International Journal of Engineering Research in Electronic and Communication Engineering (IJERECE) Vol 3, Issue 3, March 2016

Verilog Implementation of Reversible Logic Gate

[1] P Sravani [2] V Sai Koushik [3] D Srinivas
[1] Assistant Professor [2][3] B.E Scholor
[1][2][3]Department of ECE, Matrusri Engineering College, Hyderabad, Telangana, India

***Abstract*- Technologies day-to-day are becoming smaller, faster and more complex than its previous technologies being developed. Increase in clock frequency to achieve good speed and increase in number of transistors packed onto the chip to achieve complexity of a conventional system results in increased power consumption. All the gates used to perform Boolean algebra based computations by the use of silicon based semiconductor technology in a Conventional logic system are irreversible in nature.**

This is due to the mismatch of inputs and outputs. Reversible Logic is gaining interest in the recent past due to its less heat dissipating characteristics. This logic circuit maps to its unique input to the output and ensure one to one mapping and basis for emerging applications like DNA Computing, Bioinformatics, Nanotechnologies, Quantum Computing, Quantum Dot Cellular Data, Adiabatic CMOS, Thermodynamics, Low power Design and Optical Computing to produce zero power dissipation under ideal conditions.

This paper presents the combinational circuit and Verilog code for the basic Reversible Logic gates which are important (Feynman, Double Feynman, Fredkin, Toffoli and peres). Every Logic circuit which is combinational uses all these basic Reversible Logic Gates and can be verified through Simulation using Verilog HDL.

Keywords— Reversible Logic gates, Quantum Computing, Reversible Logic, Feynman, Fredkin, Toffoli and peres.

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[1] in the year 1960. According to Landauer's principle, the loss of one bit of information lost, will dissipate kT*ln(2) joules of energy where, k is the Boltzmann's constant and $k=1.38x10^{-23}$ J/K, T is the absolute temperature in Kelvin[1]. The primitive combinational logic circuits dissipate heat energy for every bit of information that is lost during the operation. This is because according to second law of thermodynamics, information once lost cannot be recovered by any methods.

Bennett [2] showed that if a computation is carried out in Reversible logic zero energy dissipation is possible, as the amount of energy dissipated in a system is directly related to the number of bits erased during computation. The design that does not result in information loss is irreversible. A set of reversible gates are needed to design reversible circuit. Several such gates are proposed over the past decades.

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 most prominent application of reversible logic lies in quantum computers [3]. A quantum computer will be viewed as a quantum network (or a family of quantum networks) composed of quantum logic gates; It has applications in various research areas such as Low Power CMOS design, quantum computing, nanotechnology, DNA computing etc.,

Quantum networks composed of quantum logic gates; each gate performing an elementary unitary operation on one, two or more two–state quantum systems called qubits. Each qubit represents an elementary unit of information; corresponding to the classical bit values 0 and 1. Any unitary operation is reversible and hence quantum networks effecting elementary arithmetic operations such as addition, multiplication and exponentiation cannot be directly deduced from their classical Boolean counterparts (classical logic gates such as AND or OR are clearly irreversible).Thus, quantum arithmetic must be built from reversible logical components [3]. Reversible computation in a system can be performed only when the system comprises of reversible gates. A circuit/gate 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 [4-6].

In order to achieving an optimized reversible circuit, some points should be considered:

1) Fan-out is forbidden.
2) Feedback and loop are not allowed.
3) Delay should be minimum.

4) Optimization parameters should be minimum.

The parameters such as number of reversible gates, number of constant inputs, garbage outputs, and quantum cost (QC) can be named as optimization parameters and are defined as:

1) The inputs, which equal to 0 or 1, are constant inputs.
2) Garbage outputs are output vectors which do not generate any useful function.
3) Quantum cost refers to the cost of the circuit in terms of primitive gate [7].

II. BASIC DEFINITIONS OF REVERSIBLE LOGIC

In this section some important factors in reversible logic are explained. The main object in reversible logic theory is the reversible function, which is defined as follows.

A. Reversible Function:

The Boolean function f(x1, x2 … xn) of n Boolean variables is called reversible if:

1. The number of outputs is equal to the number of inputs.
2. Any output pattern maps to a unique input pattern.

In other words, reversible functions are those that perform permutations of the set of input vectors [7-9].

For an (n, k) function, i.e. function with n-input k-output, it is necessary to add inputs and/or outputs to make it reversible. This leads to the following definition.

B. Reversible logic gate:

Reversible Gates are circuits in which number of outputs is equal to the number of inputs and there is a one to one correspondence between the vector of inputs and outputs [10- 12]. It not only helps us to determine the outputs from the inputs but also helps us to uniquely recover the inputs from the outputs.

C. Ancilla inputs/ constant inputs:

Anicilla inputs are used to denote the present value inputs that were added to an (n, k) function to make it reversible. The constant inputs are known as ancilla inputs. [13].

D. Garbage outputs:

Garbage is the number of outputs added to make an n-input k-output function ((n; k) function) reversible. The relation between garbage outputs and constant inputs is [7]

Input + constant input = output + garbage. [7]

As with reversible gates, a reversible circuit has the same number of input and output wires; the reversible circuit with n inputs is called an n X n circuit or a circuit on n wires.

E. Quantum cost:

Quantum cost refers to the cost of the circuit in terms of the cost of a primitive gate. It is calculated knowing the number of primitive reversible logic gates (1*1 or 2*2) required to realize the circuit.

F. Flexibility:

Flexibility refers to the universality of a reversible logic gate in realizing more functions [14].

G. Gate Level:

This refers to the number of levels in the circuit which are required to realize the given logic functions.

H. Hardware Complexity:

This refers to the total number of logic operation in a circuit. Means the total number of AND, OR and EXOR operation in a circuit [11] and [15].

I. Design Constraints for Reversible Logic Circuits:

Reversible logic imposes many design constraints that need to be either ensured or optimized for implementing any particular Boolean functions.

1) In reversible logic circuit the number of inputs must be equal to the number of outputs.
2) For each input pattern there must be a unique output pattern.
3) Each output will be used only once, that is, no fan out is allowed.
4) The resulting circuit must be acyclic.

III. REVERSIBLE LOGIC GATES

In this section, we describe all about reversible logic and reversible logic gates. Though it is already briefly described about garbage outputs, in this section we will define these with more appropriate Reversible logic gates.

(i) NOT Gate: 1*1 NOT gate is the simplest among all the reversible gates where the gate has only one input (A) and one output (B) such that B = A'. The block diagram for 1*1 NOT gate is shown in Fig.3.1. (Quantum Cost = 0)

Figure 3.1: 1x1 NOT GATE

Code :

```
module NOTGATE( input A, output P );
assign P = ~A;
endmodule
```

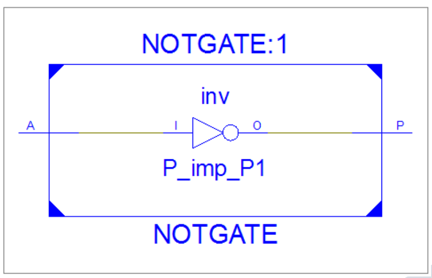

Figure 3.2: combinational circuit diagram of 1x1 NOT GATE

***(ii) Feynman Gate*: **Let I_v and O_v be the input and output vector of a 2*2 Feynman gate (FG) [16,17] respectively, where I_v= (A,B) and O_v = (P=A, Q=A⨁B). The block diagram for 2*2 Feynman gate is shown in Fig.3.3. (Quantum Cost = 1)

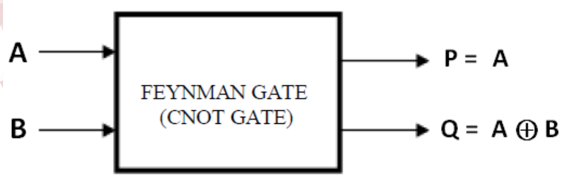

Figure 3.3: 2x2 FEYNMAN GATE (CNOT GATE)

Code:

```
module Feynman( input A, B,   output P, Q );
assign P = A;
assign Q = A^B;
endmodule
```

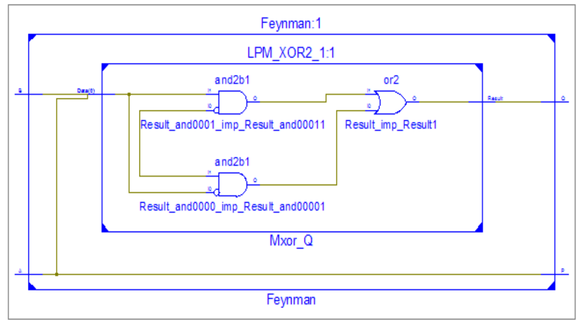

Figure 3.4: combinational circuit diagram of 1x1 CNOT GATE

***(iii) Double Feynman Gate*: **Let I_v and O_v be the input and output vector of a 3*3 Double Feynman Gate respectively, where I_v = (A, B, C) and O_v = (P=A, Q=A⨁B, R=A⨁C). Fig.3.5 shows the 3*3 Double Feynman gate. (Quantum Cost = 2)

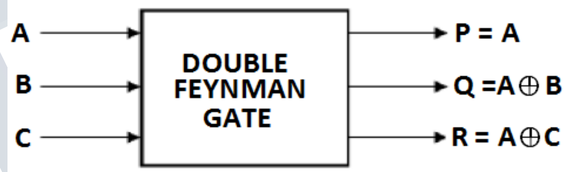

Figure 3.5: 3x3 DOUBLE FEYNMAN GATE

Code:

```
Module Double_Feynman(   input A, B, C,  output P, Q, R );
assign P = A;
assign Q = A^B;
assign R = A^C;
endmodule
```

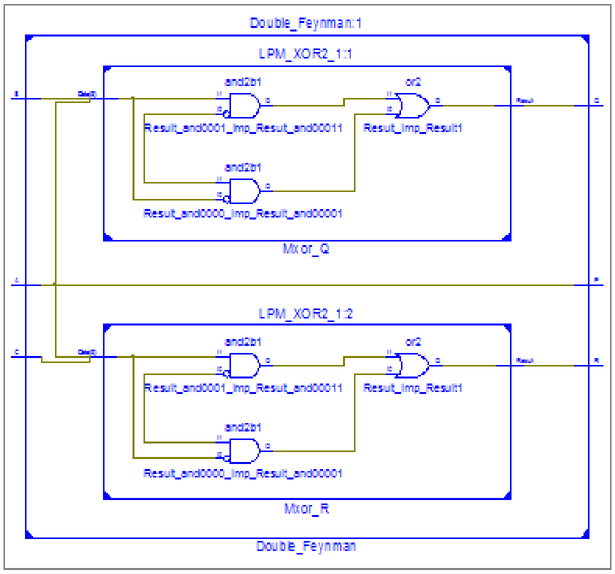

Figure 3.6: combinational circuit diagram of 3x3 DOUBLE FEYNMAN GATE

(iv) Toffoli Gate: Let I_v and O_v be the input and output vector of a 3*3 Toffoli Gate (TG) [18,19] respectively, where I_v =(A, B, C) and O_v=(P=A, Q=B, R=AB⊕ C). Fig.3.7 shows the 3*3 Toffoli gate. (Quantum Cost = 5)

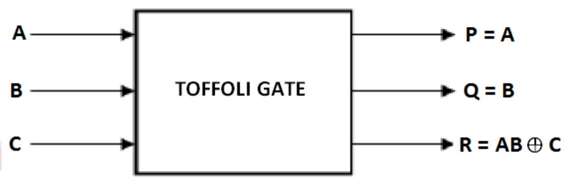

Figure 3.7: 3x3 TOFFOLI GATE

Code:

```
module Toffoli(  input A,B,C, output P,Q,R  );
assign P = A;
assign Q = B;
assign R = (A&B)^C;
endmodule
```

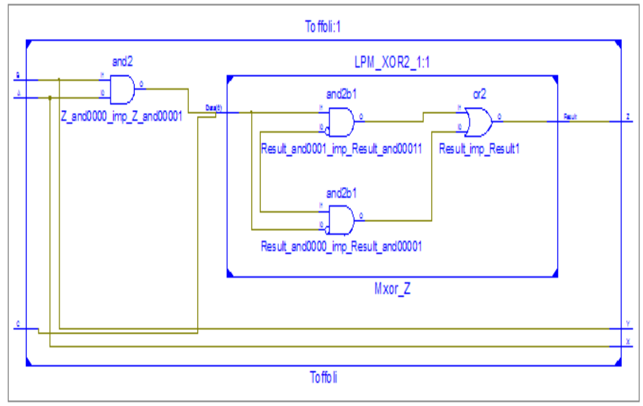

Figure 3.8: combinational circuit diagram of 3x3 TOFFOLI GATE

(v) Fredkin Gate: Let I_v and O_v be the input and output vector of a 3*3 Fredkin Gate [18,20] respectively, where I_v=(A,B,C) and O_v=(X=A,Y=A’B⊕AC , Z=A’C⊕AB). Fig. 3.9 shows the block diagram of 3*3 Fredkin gate. (Quantum Cost = 5)

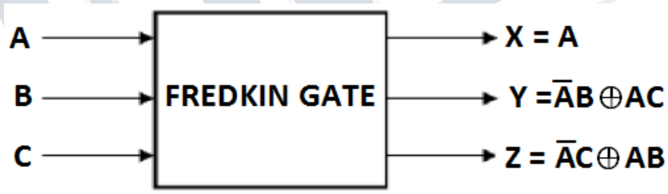

Figure 3.9: 3x3 FREDKIN GATE

Code:

```
module Fredkin(  input A, B, C,    output X, Y, Z   );
assign X = A;
assign Y =((~A)&B) ^ (A&C);
assign Z =((~A)&C) ^( A&B);
endmodule
```

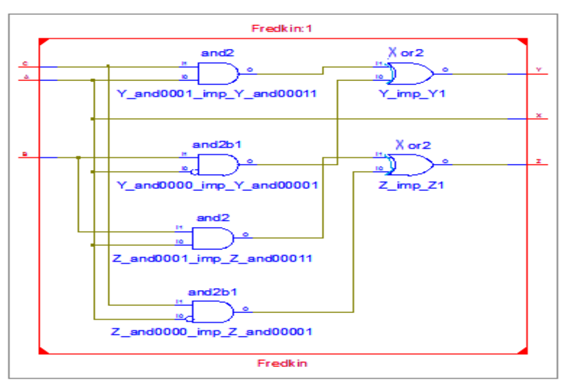

Figure 3.10: combinational circuit diagram of 3x3 FREDKIN GATE

(vi) Peres Gate: Let I_v and O_v be the input and output vector of a 3*3 Peres Gate [18,20,21] respectively, where I_v=(A,B,C) and O_v=(X=A,Y=A⊕B , Z=AB⊕C). Fig. 3.11 shows the block diagram of 3*3 Peres gate. (Quantum Cost = 4)

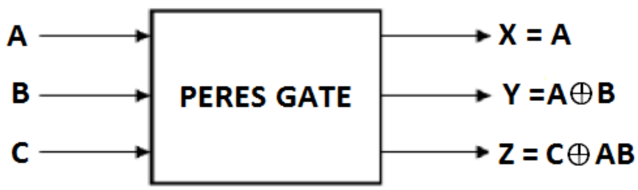

Figure 3.11: 3x3 PERES GATE

Code:

```
module Peres(input A,B,C, output X,Y,Z     );
assign  X = A;
assign Y =A^B;
assign Z =(A&B)^C;
endmodule
```

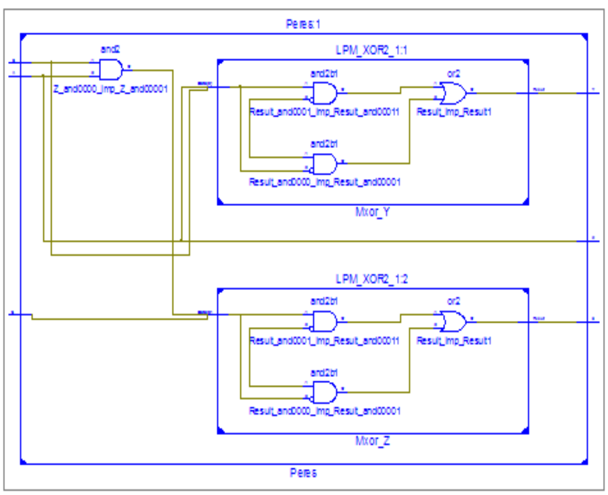

Figure 3.12: combinational circuit diagram of 3x3 PERES GATE

Peres Gate [11] is an important gate which has a low quantum cost as compared to other gates. A single Peres gate can give generate and propagate outputs when the third input C = 0. Two Peres gates can be combined to form a full adder.

IV. 4-BIT ADDER USING PFA (PERES FULL ADDER) BLOCK:

Some of the most used universal quantum gates[21] and their quantum cost are shown in figures 3.1,3.3,3.5,3.7 and 3.9. All these gates are used for implementing any logical function therefore they can also implement the full adder functions sum and carry.

Sum = A XOR b XOR C

Carry = ((A XOR B) C) XOR AB

For this we go with peres gate as it has low quantum cost as compared with the discussed above basic Reversible Logic gates. The Peres implemented Full Adder with its corresponding quantum cost is shown in the figure 4.1

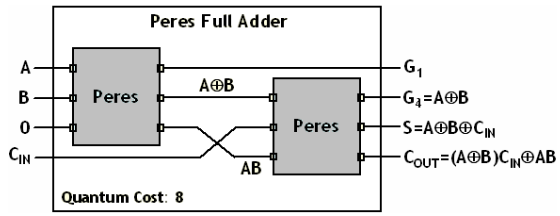

Figure 4.1:Full Adder using Peres Gates

Code:

```
module FULLADDER( input A,B,Cin, output SUM,Cout
);
Peres P1(A,B,0,G1,G2,G3);
Peres P2(G2,Cin,G3,G4,SUM,Cout);
endmodule
```

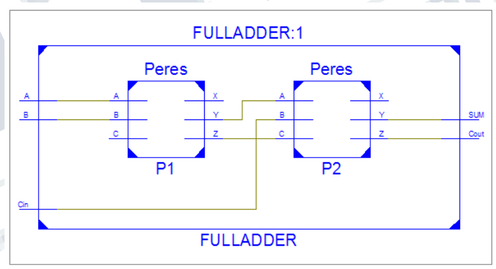

Figure 4.2: Full Adder using Peres Gates

This PFA (Peres Full Adder) can be taken as a block as shown in figure 4.3 in order to facilitate the notation of its expansion with a Quantum Cost equal to 8. The inputs order was also changed to better fit in an expansion diagram and the logic diagram is given in figure 4.4.

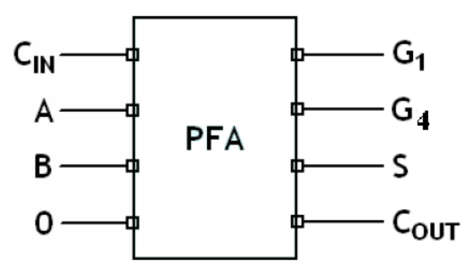

Figure 4.3: PFA as a block

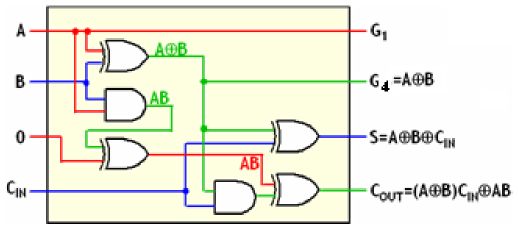

Figure 4.4: PFA traditional logic implementation.

Once we take the PFA as a block, we can derive the algorithm to implement an n-bits adder. This algorithm was implemented in this design for a 4-bit adder and can be seen in figure 4.5.

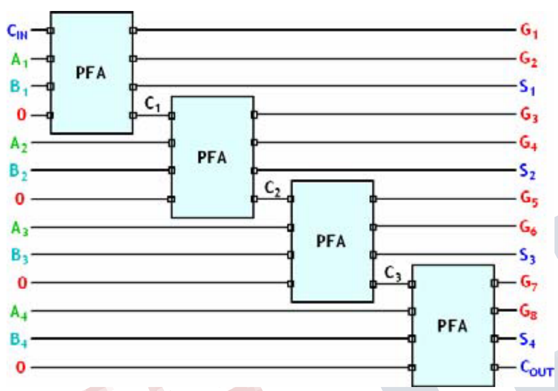

Figure 4.5: 4-bits adder implementation

V. DISADVANTAGES

1. However, in order to attain the supposed benefits of reversible computation, the reversible machine must *actually* be run backwards to attain its original state. If this is not happening then typically the machine is *heated up* and thus it stops its working.

2. You must make sure weather your computation was performed with no errors when reversible machine actually be run backwards - otherwise chaos (and not the original starting condition) may result when the machine is run backwards.

So: do you think is the reversible logic a waste of time? No. Reversible logic is of substantial significance.

VI. ADVANTAGES

What do digital power management and digital heat management even mean?

1. Digital power refers to ordered bit patterns, which can be used to do digital work.
2. Management of digital power involves moving it to where it is needed.
3. Digital heat refers to disordered bit patterns that are no good to anyone.
4. Management of digital heat involves moving it to where it can be dumped.

VII. CONCLUSION

This paper presents Verilog CODE for all Reversible Logic Gates, which provide us to design Verilog CODE of any complex combinational circuit. Here we have tried to make the Verilog code as much as possible. We can simulate and synthesis it using Xilinx 15.1 software and verified using Z series board and also calculate the power consumption and compare it with the irreversible Combinational Circuits.

REFERENCES

1. R.Landauer, “Irreversibility and Heat Generation in the Computational Process”, IBM Journal of Research and Development, 5,pp. 183-191,1961.

2. C.H.Bennett, “Logical Reversibility of Computation”, IBM J.Research and Development, pp.525-532, November1973.

3. Vlatko Vedral, Adriano Bareno and Artur Ekert, “QUANTUM Networks for Elementary Arithmetic Operations”, arXiv:quantph/ 9511018 v1, nov 1995.

4. Perkowski, M., A. Al-Rabadi, P. Kerntopf, A. Buller, M. Chrzanowska-Jeske, A. Mishchenko, M. Azad Khan, A. Coppola, S.Yanushkevich, V. Shmerko and L. Jozwiak, —”A general decomposition for reversible logic”, Proc. RM’2001, Starkville, pp: 119-138, 2001..

5. Perkowski, M. and P. Kerntopf, —”Reversible Logic. Invited tutorial”, Proc. EURO-MICRO, Sept 2001, Warsaw, Poland.

6. Hafiz Md. Hasan and A.R. Chowdhury, —”Design of Reversible Binary Coded decimal Adder by using Reversible 4 – bit Parallel Adder”, VLSI Design 2005, pp. 255 – 260, Kolkata, India, Jan 2005.

7. B.Raghu kanth, B.Murali Krishna, M. Sridhar, V.G. Santhi Swaroop —”A DISTINGUISH

BETWEEN REVERSIBLE AND CONVENTIONAL LOGIC GATES", International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com Vol. 2, Issue 2,Mar-Apr 2012, pp.148-151

8. D. Maslov, Reversible logic synthesis, Ph.D. thesis, The Faculty of Computer Science, The University of New Brunswick, Canada, 2003

9. D. Maslov, G.W. Dueck, Garbage in reversible designs of multiple output functions, in Proceedings of the 6th International Symposium on Representations and Methodology of Future Computing Technologies (RM 2003), Trier, Germany, pp. 162–170, March 2003

10. Ashis Kumer Biswas, Md. Mahmudul Hasan, Moshaddek Hasan, Ahsan Raja Chowdhury and Hafiz Md. Hasan Babu. ―"A Novel Approach to Design BCD Adder and Carry Skip BCD Adder". 21st International Conference on VLSI Design, 1063-9667/08 $25.00 © 2008 IEEE DOI 10.1109/VLSI.2008.37.

11. Abu Sadat Md. Sayem, Masashi Ueda." Optimization of reversible sequential circuits". Journal of Computing, Volume 2, Issue 6, June 2010, ISSN 2151-9617.

12. H.Thapliyal and N. Ranganathan, ―"Design of reversible sequentialcircuits optimizing quantum cost, delay and garbage outputs", ACMJournal of Emerging Technologies in Computing Systems, vol. 6, no.4, Article 14, pp. 14:1–14:35, Dec. 2010.

13. J.Smoline and David P.DiVincenzo, ―"Five two-qubit gates are sufficient to implement the quantum fredkin gate", Physics Review A, vol. 53, no.4, pp. 2855-2856,1996.

14. M. Haghparast,K. Navi, "A Novel Reversible BCD Adder For Nanotechnology Based Systems", American Journal of Applied Sciences 5 (3): 282-288, 2008,ISSN 1546-9239.

15. Abu Sadat Md. Sayem, Masashi Ueda.‖ Optimization of reversible sequential circuits‖. Journal ofComputing, Volume 2, Issue 6, June 2010, ISSN 2151-9617.

16. Milburn, Gerard.j., The Feynman processor perseus books 1998

17. Feynman R., 1985. Quantum mechanical computers, Optics News, 11: 11-20.

18. E. fredkin, T. Toffoli, "Conservative Logic", International Journal of Theory of Physics,21, 1982, pp 219-253

19. Toffoli T., 1980. Reversible computing, Tech Memo MIT/LCS/TM-151, MIT Lab for Computer Science.

20. P.D. Picton, " Fredkin gates as the basic for comparison of different logic designs", Synthesis and optimization of logic system, London, VK, 1994.

21. Peres, A. 1985. Reversible logic and quantum computers. Physical Review A, 32: 3266-3276.