Realization of Various Gate level Combinational Circuits using Reversible Fredkin Gate

Greeshma Arya, D. S. Chauhan

Abstract: ***This paper presents the digital logic gates which are reconstructed using fredkin gate [1]. The advantage of basic fredkin gate is that we could save the thermal waste which comes out due to computation that causes heat as bits just disappear into loss of energy. Such computation won't need any energy input. These assumptions make the gates sound like an energy efficient solution. However the implementation is done at level of logic gates. This can further be used in sequential circuits to increase the life time of transmitter and receiver circuitry of nodes. It will make the transmission and aggregation of information at node very energy efficient. The drawback of this application is it will cost fare amount of time to process data. These technical hurdles will increase latencies at node level. The protocols infused with energy optimization methods and reversible logic gates offered noticeable improvements in achieving performance and ensuring security of data and graphics. Since the 1980s, with work of Fredkin [1], the reversible circuits have been used in building large scale integration of circuits as elementary units of mobile computing, and recently in wireless networks, drug designing and ultra-fast computing technologies [4].***

Keywords: Entropy, Fredkin gate, Quantum Computing, Reversible Computing,

I. INTRODUCTION

Reversible computing has become a recent research trend in arena of computing where the airthmatic and logical operations upto some extent are time reversible [5]. These methods of computing use deterministic approaches for state transfers of data sets. In reversible computing we move from one state to another state maintaining energy saving but with the condition of one to one mapping of states. This approach of doing computation operations within the digital elements is very different from the conventional computing [1]

Basic Reversible Logic has the following Rules to get it implemented in any conventional digital circuit.

- Number of outputs equal that of inputs. N=M
- Unique input to unique output pattern
- Garbage outputs to be reduced to a commendable level
- Gate count should be reducing.
- Constant value for inputs [‘0’ or ‘1’] should be reduced.
- Fan-out brings down to nil.
- Loop back circuit/ Feedback is not allowed

A. Principle of thermodynamics and reversibility:

According to Neuman- Landauer[2] discovery about of the energy dissipated per bit in irreversible operation can be used in further operations by making the system reversibility operation upto some extent. So actually reversibility is aimed to upgrade the energy of systems used in performing various operations. As discussed in the work of Von Neumann-Landauer that the per bit energy dissipated in the form of heat can be calculated using kT ln 2. Although the Landauer limit was not the exact estimation of the energy consumption of by systems during operations. In last decade, it is predicated that power saving using reversible computing can cause remarkable increase in overheads which potentially enhances the impact of energy waste in digital circuit designs again, causing the potential slowdown in technology advancement and restricting research to achieve offered levels of energy efficiency if reversible computing principles are used [2]. This logic is inferred by the second law of thermodynamics which states that the amount of instability faced by system during operations always increases and the result is not reversible. Here the instability or disorder can be better understood by term Entropy of the system.

Following the powerful laws of thermodynamics and entropy we can make perfect estimates regarding energy utilization. Because entropy is used to measure energy constituted by any sort of information. So computational processes that are random in nature i.e. non deterministic require physical reversibility [3] along with the increase in set of possible initial computational states to move forwards to next state. However achieving this goal becomes difficult for the design engineering. Due to certain difficulties in devising of perfects information of states for data processing and aggregation, allowing researchers to build systems that dissipate much less than kT ln 2 powers in the form of heat becomes a hardship as arithmetic and computational functions that they perform cannot be known perfectly restricting architectural overheads. However this saving of energy comes in the form of enhanced network lifetime but with burden of latencies and bandwidth restrictions.

Physical reversibility and logical reversibility [3] are closely related types of reversibility. The physically reversible is isentropic in nature i.e. it results in no increase in actual entropy of system involved in computation [1]. There are methods of circuit design which exhibits this property. Although practically any non-stationary process can be isentropic in nature. A lot of research is going in this direction to achieve set levels for energy optimizations, so models can be developed showing perfect reversibile operations of any systems.

Revised Manuscript Received on February 15, 2020.

Correspondence Author:

Greeshma Arya, Department of ECE, UTU, Dehradun, India. Email: arya.greeshma@gmail.com

Prof D. S. Chauhan, Department of Electrical Engineering, IT BHU, Varanasi, India. Email: pdschauhan@gmail.com

However systems that is sufficiently well isolated from any other outside environments can shows good performance in attaining the required level of power. On contrary review papers on quantum computing depicts clearly that the architectural overheads in circuits are increased and optimization is restricted to small level circuits. This opens the new dimensions for goal oriented research challenges, and also signifies their need in coming era of technology based on reversible computing [6].

B. Basic Fredkin Gate

The concept of basic Fredkin gate was given by Edward Frekin in 1980. Because of its principle property to work, it is also known as Conditional SWAP gate. It is one of the best gate to build a reversible computational circuit suitable for energy saving. This gate is universal, so all arithmetic and computational circuit can be developed completely using this gates (Figure 1).

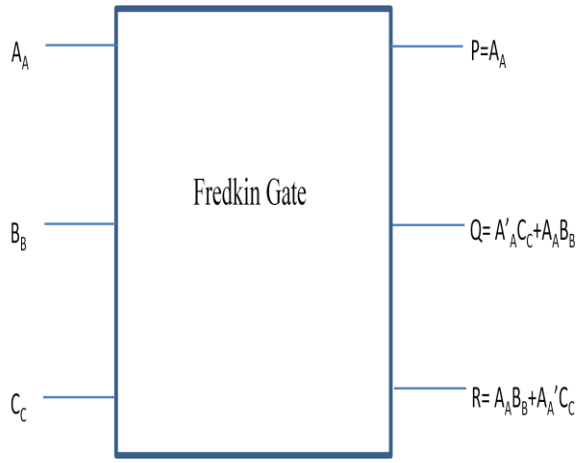

Figure1. Basic fredkin gate diagram

Originally this gate has three inputs say named as A_A, B_B, C_C then these three inputs will be mapped to three outputs say P, Q, R. Here the input C_C is always mapped directly to the output R in all cases. If for instance taking the first case as $C_C = 0$, then no swap will occurred. However for second case if $C_C = 1$, then all outputs are mapped like $A_A \leftrightarrow P$, $B_B \leftrightarrow Q$, $C_C \leftrightarrow R$.

If this gate is run in reversible direction, then it will undo itself. So we can easily fed output of previous stage fredkin gate to the input of next stage fredkin gate as $P \leftrightarrow A_A$, $Q \leftrightarrow B_B$, $R \leftrightarrow C_C$ and then the result of the this fredkin circuit is the similar as the inputs of the previous fredkin circuit. For the implementation of reversible logic we need to know exact information of arithmetic and other computational operations so that we can model the circuit with utmost certainty about the entire state of the operation, and then reconfigure it as per each arithmetic and logic operation that is occurred inside. This we can attain by keeping the precise knowledge of involved energy bits in the operations to carrying out all the arithmetic, logical operations within the node. The nodes must be configured in such a manner that the most of the energy can be retrieved in a deterministic way, and further be utilized in performing next stage operations, preferably not permitted to go in vein in the form of temperature. In the year 1973 Charles H. Bennett in his research at IBM Research centre showed that any circuit could be made both reversible both logically and thermodynamically [3], and therefore saving more per bit energy in bit manipulations , computation operations per unit of physical energy dissipated.

II. PROPOSED REVERSIBLE LOGIC GATES:

This paper presents conventional Mux and Demux circuits and their changed version after using fredkin gate to turn it into reversible MUX and DEMUX. These circuits are built, simulated on Vivado (v2016.4) and the results are analyzed with a comparative study.

A. Conventional MUX and DEMUX:

A general Multiplexer circuit works as data selector due to its inherent property that it choose among all provided input signals and sends the data to a single line on output. MUX circuit elements are mainly used to increase the amount of data transfer within a network under the constraints of a fixed amount of time period and bandwidth, with the fact that these parameters play an important role in designing an application specific protocol WSN networks. So we started our implementation with these combinational circuits which are integral part of any communication circuit.

On the opposite, a De-Multiplexer is a circuit element that takes the input from single line and forwards it to multiple outputs. A multiplexer circuit element is used in transmitter circuit while a de multiplexer element is used in the receiver circuit of a communication link. So we can say that a MUX circuit is a MISO switch, and a DEMUX is a SIMO switch [5].

B. Reversible Fredkin Mux (2- to-1):

Reversible Fredkin MUX 2-to-1 circuit diagram is shown in fig 2. Here, S is selection line, d0, d1 is input, E is enable pin, g1, and g2, g3, and g4 are Garbage outputs. Output is referred using following equation i.e. Output = $\sim S.D_0 + S.D_1$

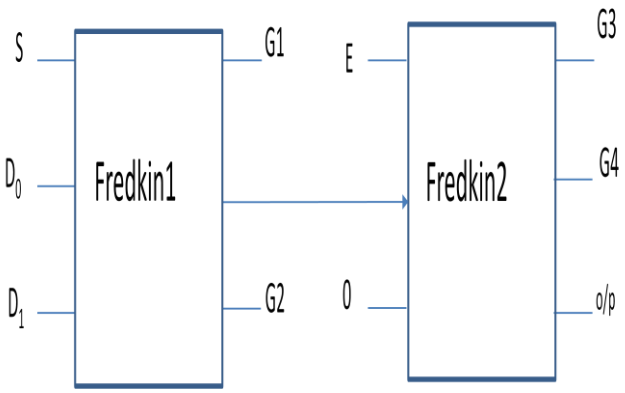

Figure 2 Reversible Fredkin MUX (2*1) circuit diagram

Verilog code for reversible Mux(2*1): module fredkin_mux(input S,D_0,D_1,E, output out);wirek1,k2,k3,k4;wire k5,k6,k7,k8;wireG1,G2,G3; wireG4,G5; buf b1(G1,S); and a1(k1,~S,D_0);and a2(c2,S,D_1); xor x1(k3,k1,k2);and a3(k4,S,D_0);and a4(k5,~S,D_1);xor x2(k6,k4,k5); buf b2(G3,E);and a5(k7,~E,k6); xor x3(G4,G7,1'b0); and a6(k8,E,k6); xor x4(out,k8,1'b0); endmodule

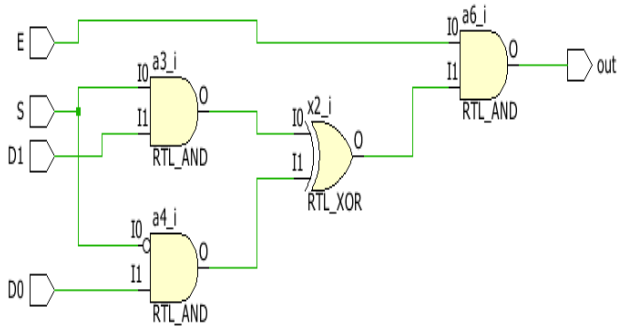

Figure 3 RTL Circuit for Reversible Fredkin Mux (2*1)

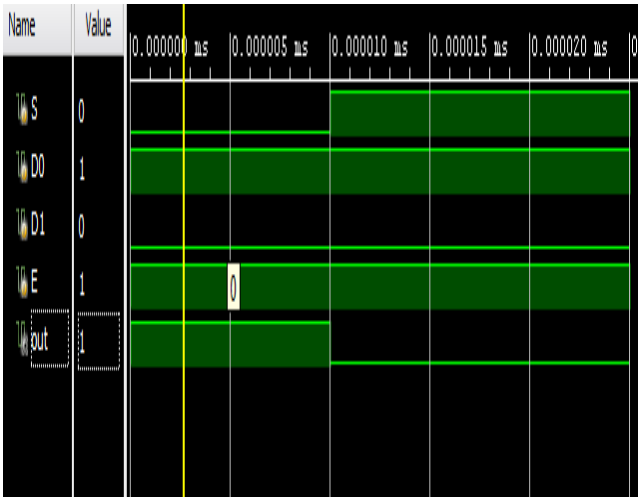

Figure 4 Results for Reversible Fredkin MUX (2*1)

C. Reversible fredkin mux (4*1):

For this Fredkin MUX inputs are S_0,S_1,E,D_0,D_1,D_2,D_3 and output is given by Y. Output statement for this MUX is shown in figure 5.

.

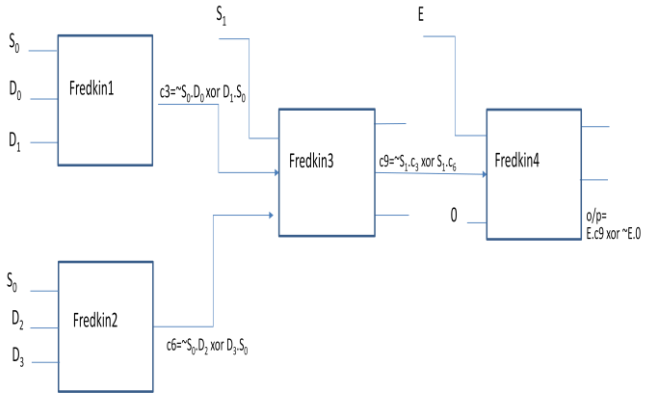

Figure 5 Reversible MUX (4*1) circuit diagram

Verilog code for Reversible MUX 4-to-1: module fredkin_mux_4x1(inputD_0,D_1,D_2,D_3, inputS_1,S_0,E,outputY); wirek1,k2,k3;wirek4,k5,k6,k7,k8,k9;and a1(k1,~S_0,D_0); and a2(k2,S_0,D_1); xor x1(k3,k1,k2); and a3(k4,~S_0,D_2);and a4(k5,S_0,D_3) xor x2(k6,k4,k5); and a5(k8,~S_1,k3); and a6(k7,S_1,k6);xor x3(k9,k7,k8);and a7(Y,k9,E); endmodule

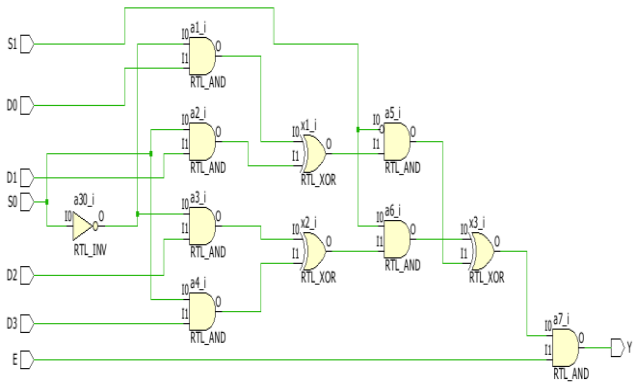

Figure 6 Reversible Fredkin MUX (4*1) RTL circuit

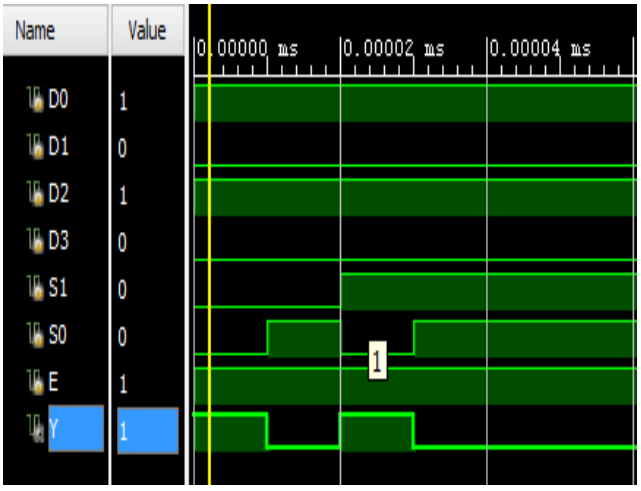

Figure 7 Results for Reversible Fredkin MUX (4*1)

D. Reversible Fredkin MUX (8*1):

This reversible Fredkin MUX circuit is shown in figure 8. Input connections are defined as S_0,D_0,D_1,D_2,D_3,D_4,D_5,D_6,D_7 and output is defined by Y as shown in figure 8.

Verilog code of MUX (8*1): module fredkin_mux_8x1 (input D_0,D_1,D_2,D_3,D_4,inputD_5,D_6,D_7,inputS_2,S_1,S_0,E,outputY);wirek1,k2,k3,k4;wirek5,k6,k7,k8,k9;wirek10,k11,k12;wirek13,k14,k15,k16,k17,k18;wirek19,k20,k21;wirek22,k23and a1(k1,~S_0,D_0); and a2(k2,S_0,D_1); xor x1(k3,k1,k2); and a3(k4,~S_0,D_2); and a4(k5,S_0,D_3);xor x2(k6,k4,k5); and a5(k7,~S_0,D_4); and a6(k8,S_0,D_5); xor x3(k9,k7,k8); and a7(k10,~S_0,D_6);

and a8(k11,S_0,D_7); xor x4(k12,k10,k11); and a9(k13,~S_1,k3);and a10(k14,S_1,c6); xor x5(k15,k13,k14); and a11(k16,~S_1,k9);and a12(k17,S_1,k12);

xor x6(k18,k16,k17);and a13(k19,~S_2,k15);and a14(k20,S_2,k18); xor x7(k21,k19,k20);and a15(Y,E,k21) endmodule

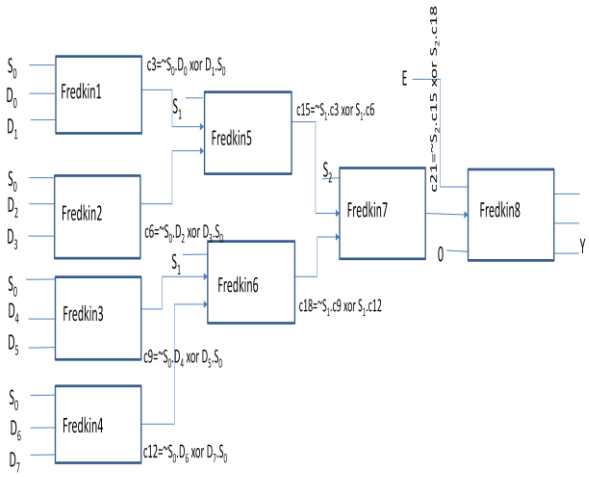

Figure 8 Reversible Fredkin MUX(8-to-1) circuit diagram

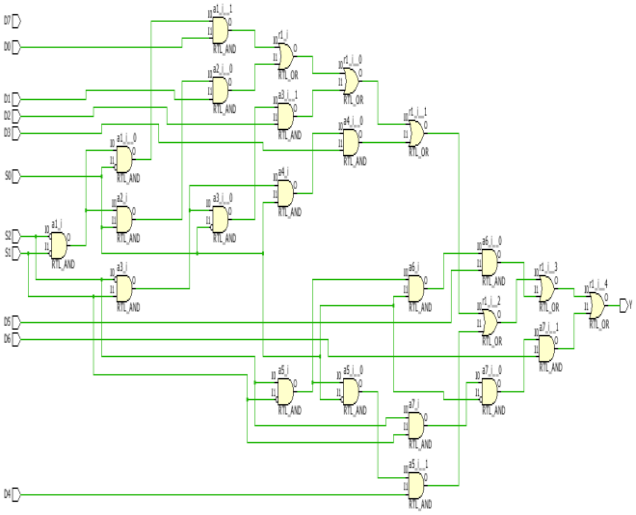

Figure 9 Reversible Fredkin MUX (8-to-1) RTL circuit

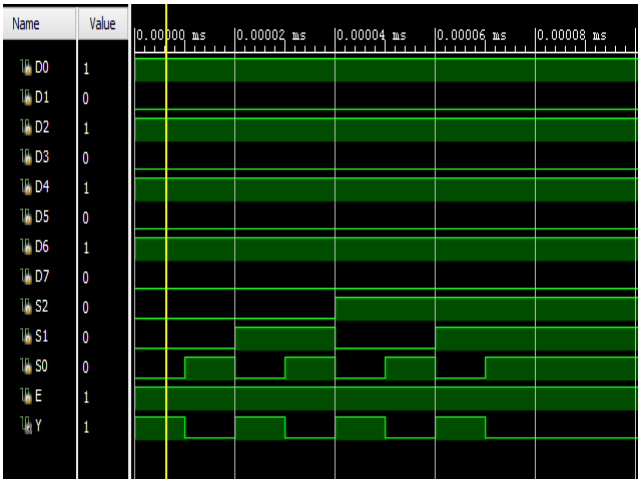

Figure 10 Results for Reversible Fredkin MUX (8-to-1)

E. Reversible demux (1-to-4):

For reversible fredkin DEMUX the ouput statement is achieved using the basic working principle of DEMUX. Here the Output statement is given as: **OUTPUT** Y_0, Y_1, Y_2, and Y_3, as shown in figure 11.

Verilog code for Fredkin Demux : module fredkin_Demux(input E,S_1,S_0,I,outputY_0,Y_1,Y_2,Y_3); wirek1,k2,k3;wirek4,k5,k6,k7,k8;wirek9,k10;wirek11,k12,k13;wirek14,k15,k16;wire k17,k18,19,k20;wirek21,k22,k23;wirek24,k25,k26;wire k27,k28,k29;wirek30,k31,k32;wirek33,k34;not n1(k33,S_1); not n2(k34,S_0);anda1(k1,k33,k34);anda2(k2,S_0,1'b0); xor x1(k3,k1,k2);and a3(k4,k3,I);and a4(k5,~k3,1'b0); xor x2(k6,k4,k5);and a5(k7,E,k6); and a6(k8,~k6,1'b0); xor x3(Y_0,k7,k8); and a7(k9,k33,S_0);and a8(k10,k34,1'b0); xor x4(k11,k9,k10); and a9(k12,k11,I); and a10(k13,~k11,1'b0); xor x5(k14,k12,k13); and a11(k15,E,k14);and a12(k16,~k14,1'b0);xor x6(Y_1,k15,k16);and a13(k17,S_1,k34); and a14(k18,S_0,1'b0); xor x7(k19,k17,k18); and a15(k20,k19,I); and a16(k21,~k19,1'b0); xor x8(k22,k20,k21); and a17(k23,E,k22); and a18(k24,~k22,1'b0); xor x9(Y_2,k23,k24); and a19(k25,S_1,S_0);and a20(k26,k34,1'b0); xor x10(k27,k25,k26); and a21(k28,k27,I); and a22(k29,~k27,1'b0); xor x22(k30,k28,k5);and a23(k31,E,k30); and a24(k32,~k30,1'b0); xor x23(Y_3,k31,k32); endmodule

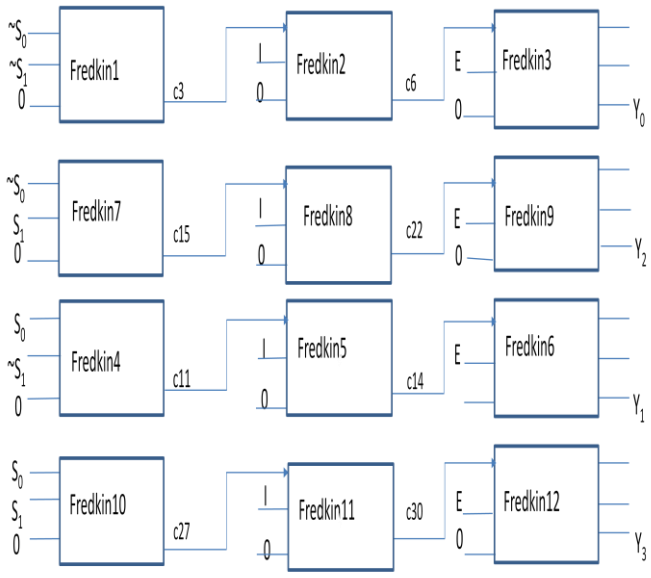

Figure 11 Reversible Fredkin DEMUX (1*4) circuit diagram

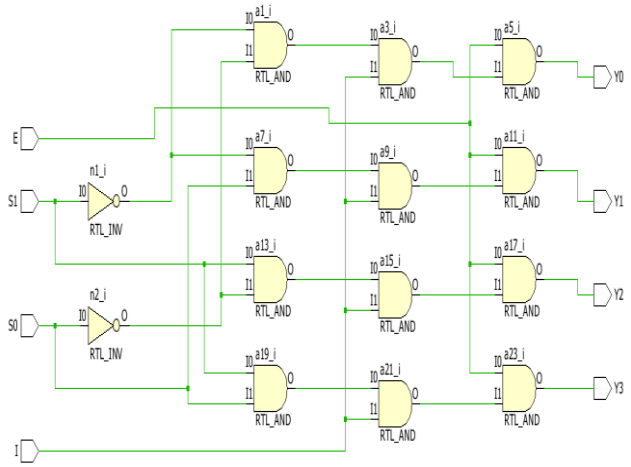

Figure 12 Reversible Fredkin DEMUX (1*4) RTL Circuit

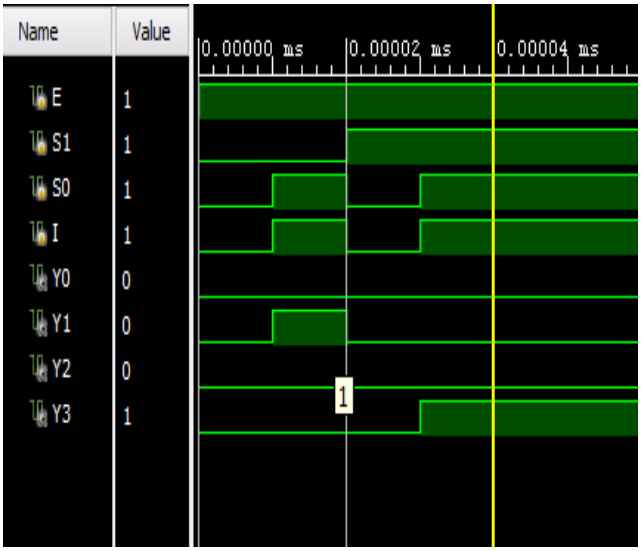

Figure 13 Results for Reversible Fredkin DEMUX (1-to-4)

F. Reversible Fredkin Encoder (8-to-3):

Output statements for Encoder are provided be AA, BB, and CC. The statements are shown in figure 14. Here inputs are given as E, d_1, d_2, d_3, d_4, d_5, d_6, d_7.

Verilog code for Fredkin Encoder 8-to-3: module reverse_encoder_3_8(AA,BB,CC,d_0,d_1,d_2,d_3,d_4,d_5,d_6,d_7,E);output AA,BB,CC; input E,d_0,d_1,d_2,d_3,d_4,d_5,d_6,d_7; wire k1,k2,k3,k4,k5,k6,k7,k8;wire g1,g2,g3,g4,g5,g6,g7,g8;buf buf1(c1,d_3); or frd1(c2,d_1,c1);and and1(g1,c2,1'b1);or frd2(k3,d_2,k1);and and2(g2,k3,1'b1);buf buf2(k4,d_5);or frd3(k5,d_4,k4);and and3(g3,k5,1'b1);buf buf3(k6,d_7);or frd4(k7,d_6,k6);and and4(g4,k7,1'b1);or frd5(k8,k4,k6);and and5(g5,k8,1'b1);or frd6(g6,g1,g5);and and6(AA,g6,E);or frd7(g7,g2,g4);and and7(BB,g7,E);or frd8(g8,g3,g4);and and8(CC,g8,E);endmodule

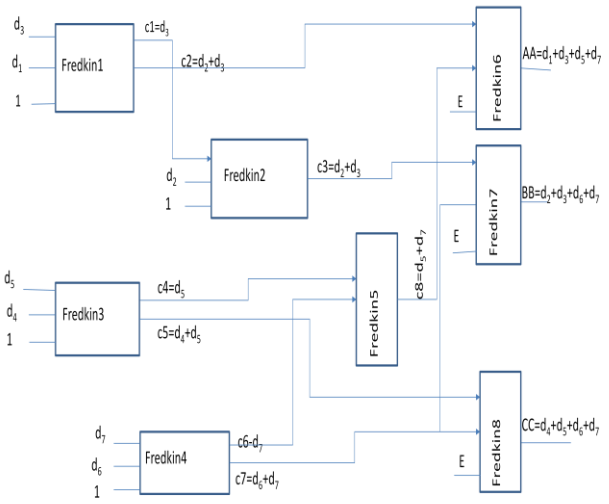

Figure 14 Reversible Fredkin Encoder (8*3) circuit diagram

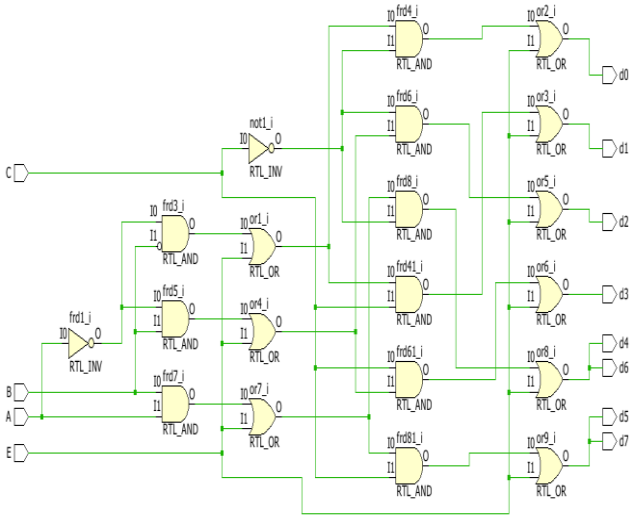

Figure15. Fredkin ENCODER (8*3) RTL Circuit

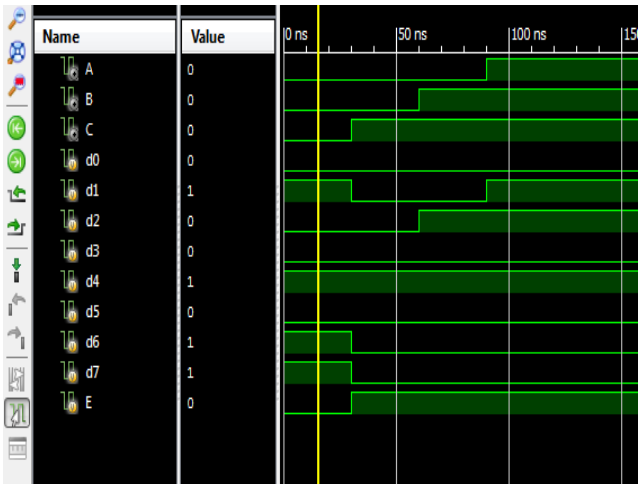

Figure 16 Results for Reversible ENCODER (8*3)

III. RESULTS

After simulation of reversible fredkin circuits for various configurations on vivado (v2016.4) using FPGA Board Nexsys4ddr (xc7a100tcsg324-1). These results are shown for single circuit elements and modeling is done at gate level, which shows fair amount for power saving. Further if they are simulated for large scale digital circuits they will show adequate amount of energy saving as the circuits are sequential.

Table 1 Comparison of results with conventional circuits

S.no.	Combinational Circuit Element	Irreversible Circuit (in Watt)	Reversible Circuit (in Watt)
1	MUX (2-to-1)	0.444	0.385
2	MUX (4-to-1)	0.545	0.446
3	MUX (8-to-1)	0.629	0.521
4	DEMUX (1-to-4)	0.757	0.530
5	ENCODER (8-to-3)	0.969	0.850

IV. CONLUSION AND FUTURE SCOPE

The proposed circuits has shown better results for energy saving during operations, they have also reduced number of gates used. A comparison of energy level has been done in Table 1 with the conventional gates. These proposed circuits can be efficiently used in node level circuitry, which further help in saving energy during data aggregation and transmission. This proposed work is step towards building large and energy efficient networks with its use in large integration of sequential circuits.

In next segment of work fredkin will be used in other digital gate circuits e. g. counters and microcontroller circuits. Somehow, any newly developed circuit using reversible logic gates will still suffer from heat dissipation because all error detection and correction methods are irreversible operations in nature. In near future the applications of this kind of energy saving mechanism will be used in development of ultra efficient long lived communication networks along with their vast use in quantum computing.

APPLICATIONS

Other areas of applications of reversible computing are aerospace physics, chemistry modelling and material design and modelling, drug design and discovery, computer security and cryptography, and financial processing. Somehow areas that require high energy efficiency, speed and performance will always looking for quantum computing as a promising solution. Few are mentioned here as,

- Wireless Communication.
- Big data computing.
- Mobile Computing.
- Molecular computing.
- Computer graphics.
- Image and media processing using digital signal processing (DSP).

REFERENCES

1. E. Fredkin and T. Toffoli, “Conservative logic,” *International Journal of Theor. Physics*, vol. 21, pp. 219-253, 1982.
2. Landauer .R, “Irreversibility and heat generation in the computing process”, IBM J. Research and Development,Vol.5, Issue 3, July 1961.
3. Bennett C.H., “Logical reversibility of Computation”, IBM J.Research and Development,Vol.14, Issuse 6, Nov 1973.
4. P. Shor, “Algorithms for quantum computation: discrete log and factoring”, Proc. 35th Annual Symp. On Found. Of Computer Science, IEEE Computer Society, Los Alamitos, 1994.
5. Priyanka P, Rajesh K M,”Analysis of reversible logic gates”, *International Journal of Innovative Research in Technology*, ISSN: 2349-6002, Volume 4, Issue 7, Page(s):398-401, December 2017
6. Allen Lang Macdonald,“ A new statement of the second law of thermodynamics” American journal of physics 63(12):1122-1127.
7. Simon Hykin,”Communication Sysytem”, 4th edition, John Wiley Publications.
8. T. Toffoli., “Reversible Computing‖, Tech memo MIT/LCS/TM-151”, MIT Lab for Computer Science . 1980.
9. A. Peres,”Reversible Logic and Quantum Computers”, Physical Review A, vol. 32, 1985.
10. P. Singla and N. K. Malik, “A cost effective design of reversible programmable logic array,” *IJCA*, vol. 41, no. 15, Mar. 2012.

AUTHORS PROFILE

Ms Greeshma Arya received her B Tech in Electronics and Instrumentation Engineering from Lucknow University, and M Tech in Digital Communication from AKTU, Lucknow. She has more than 10 years of teaching experience. Her area of research are electronic circuits and wireless communication. She has authored 8 research papers in refereed international Journals and another two are accepted for publication in scopus journals. Presently she is working as assistant professor in ECE department with IGDTUW, Delhi.

Prof. D S Chauhan, born on 18th September, 1949 at Dholpur (Rajasthan), was Vice-Chancellor, U.P. Technical University (2000-06), Lovely Professional University (2006-07), Jaypee University of Information Technology (2007-09), Uttarakhand Technical University (2009-13), Chairman, BOAT(NR), Kanpur had his post doctoral work at Goddard Space Flight Centre, Greenbelt Maryland, USA and B.Tech. (1972) IIT. B.H.U., M.Tech (1978) N.I.T., Tiruchirapalli, and Ph.D (1986) IIT, Delhi. He has served at Banaras Hindu University since 1979 till April, 2013 began as Assistant Professor then Associate Professor and the Professor. He has served Director, KNIT (State Govt. Engg. College), Sultanpur (1999-2000). He was Member of AICTE (2001-04) and member, University Grants Commission (2006-09). Lead delegation NAFSA 2009 Indian team to USA. He has produced 41 Ph.D. and 2 D.Sc. and published 267 research papers in International and National Journals and Conferences.