COMPARATIVE ANALYSIS OF 4-BIT AND 8-BIT REVERSIBLE BARREL SHIFTER DESIGNS USING REVKIT

Hariprasad R.[1], Ganesh R.[1], Ravi S.[2], Marimuthu R.[1] and Balamurugan S.[1]
[1]School of Electrical Engineering, VIT University, Vellore, Tamil Nadu, India
[2]School of Electronics Engineering, VIT University, Vellore, Tamil Nadu, India
E-Mail.: hariprasad.ravi@gmail.com

ABSTRACT

In the recent years, reversible logic has emerged as a viable approach in power optimization and also has found its importance in low power CMOS, quantum computing, nanotechnology, and optical computing. The main challenge in reversible circuits is to optimize the quantum cost, time delay and the garbage outputs associated with the reversible circuit. 'RevKit' in recent years has become a popular and powerful tool for design visualization, implementation and analysis in reversible computing. In this work, we have implemented the design of reversible 4-bit and 8-bit barrel shifter circuits in RevKit and results are analyzed in terms of quantum cost, delay, garbage outputs, gate count, line count and transistor cost. Further, the simulation results have been documented and tabulated to facilitate a comparative study with conventional designs.

Keywords: reversible circuits, barrel shifters, quantum cost, time delay, garbage output, RevKit.

INTRODUCTION

In irreversible logic computations [1], each bit of information lost generates kTln2 joules of heat energy, where k is Boltzmann's constant and T is the absolute temperature at which the computation is performed. Thus, the amount of energy dissipated in a system bears a direct relationship to the number of bits erased during the computation. The kTln2 energy dissipation can be avoided [2] if a computation is carried out in a reversible manner [3]. Reversible circuits are built using reversible gates, which are logic gates that provides unique output for a unique input and there is a one to one mapping between the input and output.

Rotating and shifting data in a single cycle are required in several applications like efficient computations and arithmetic operations. Barrel shifters, more suitable for this kind of operations, since, it is capable of shifting or rotating the inputs in a single cycle and find great importance in the digital signal processing computation. In reversible system information is not erased. The number of inputs and outputs are equal in reversible gates, which means that the input stage can always be retained from the output stage. Thus, such an implementation of reversible barrel shifter will be highly efficient when compared to any conventional design in terms of time delay, garbage output or the quantum cost associated with such a structure.

The majority of the work that currently exists in literature focuses on optimizing the reversible sequential designs in terms of number of reversible gates and garbage outputs using functional verification. A few prior works have used design tools such as RevKit [15] to simulate reversible sequential circuits to determine quantum cost, the delay, and other important parameters. An improved design of reversible 8-bit barrel shifter is presented in [9] and verified by transistorized implementation of the design to show the layout, power, delay, PDP and LVS using Tanner EDA tool. However, they fall short of estimating the quantum cost, garbage outputs, gate count, line count and transistor cost.

In this work, we have implemented the design of reversible 4-bit barrel shifter and the improved 8-bit barrel shifter [9]. Both the design analyzed using 'RevKit' [15] in terms of quantum cost, delay, garbage outputs, gate count, line count and transistor cost. The results are then compared with traditional designs which can be extrapolated from simple multiplexer designs. The traditional 4-bit barrel shifter is constructed using a multiplexer [3] architecture. The multiplexer is implemented using Fredkin gates. The generated multiplexer is connected in a sequence to form the architecture of 4-bit barrel shifter. Hence a similar methodology is used in constructing the architecture of an 8-bit barrel shifter where more number of reversible multiplexers is connected to perform the 8-bit shifting operation.

This paper is organized into the following sections. Section II provides the basics of reversible logic gates and their quantum implementation. Section III briefs about the delay, quantum cost and garbage output calculation in reversible logic circuits. Section IV discusses the design methodology of implementation of reversible 4-bit and 8-bit barrel shifters. In section V, the simulation results are discussed. Conclusion and the future works are discussed in section VI.

REVERSIBLE GATES

A Reversible Gate is an n-input, n-output (denoted by n *n) circuit. There is a one-to-one mapping between the inputs and the outputs. Because of this corresponding mapping the input can be traced back to the output. This prevents the loss of power or logic, which is the major problem with conventional logic circuits. There are several reversible gates existing in the literature such as 1x1 NOT gate, 2x2 Feynman or CNOT gate, 3×3 reversible gates such as Fredkin gate and Peres gate. Any reversible gate can be realized using the 1×1 NOT gate, and 2×2 reversible gates such as Controlled-V and Controlled-*V*+ (V is a square-root-of NOT gate and *V*+ is its hermitian) and the Feynman gate which is also known as the Controlled NOT gate (CNOT). There are many such gates and each reversible gate having a cost associated with it called the quantum cost. The quantum cost of a reversible gate is the number of 1×1 and 2×2 reversible gates or quantum logic gates required in its design. The quantum costs of all reversible 1×1 and 2×2 gates are assumed to be unity [4-6]. Thus, to conclude the quantum cost of a reversible gate can be calculated by counting the numbers of NOT, Controlled-V, Controlled-*V*+ and CNOT gates required in its design architecture.

FEYNMAN GATE

The Feynman gate (FG) or the Controlled-NOT gate (CNOT) is a 2-input 2-output, reversible gate having the mapping (A, B) to (P = A, Q = AB) where A, B are the inputs and P, Q are the outputs, respectively. Since it is a 2×2 gate, it has a quantum cost of 1. The input and outputs of Feynman gates are represented in Figure-1(a). The Feynman gate can be used for replicating the signal in order to avoid the fan out problems in reversible logic as shown in Figure-1(d). Also, it can be also be used for generating the complement of a signal as shown in Figure-1(c). The quantum representation of Feynman gate is shown in Figure-1(b).

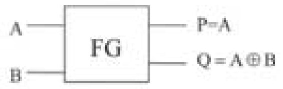

Figure-1(a). Feynman gate.

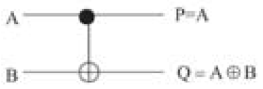

Figure-1(b). Quantum representation of Feynman gate.

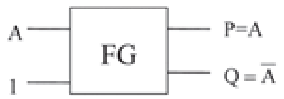

Figure-1(c). Generating a compliment using Feynman gate.

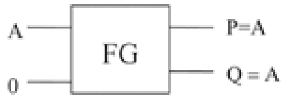

Figure-1(d). Replicating a signal using Feynman gate.

FREDKIN GATE

Fredkin gate is a reversible 3*3 gate which maps inputs (A, B, C) to outputs (P=A, Q=A'B+AC, R=AB+A'C). The Fredkin gates are widely used in the implementation of various reversible logic circuits. The quantum cost of such a gate is 5 and is equivalent to a 2*2 Feynman gate with the Quantum cost of each dotted rectangle is 1, 1 V and 2 CNOT gates [2] [14]. The input and outputs of Fredkin gates are represented in Figure-2.

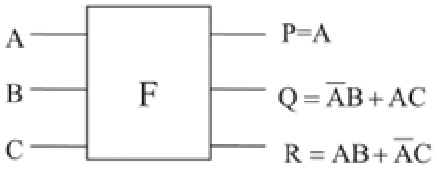

Figure-2. Fredkin gate.

QUANTUM COST, DELAY AND GARBAGE OUTPUT CALCULATION

In any reversible logic circuit design the important cost metrics involved are the quantum cost, the delay, and the number of garbage outputs. Each reversible gate has a cost associated with it called the quantum cost. The quantum cost of a reversible gate is the number of 1×1 and 2×2 reversible gates or quantum logic gates required in its design. The quantum costs of all reversible 1×1 and 2×2 gates are taken as unity. Any reversible gate can be realized using the 1×1 NOT gate, and 2×2 reversible gates such as controlled-V and controlled-V+ and the Feynman gate as discussed .Thus the quantum cost of a reversible gate can be calculated by counting the numbers of NOT, controlled-V, controlled-V+ and CNOT gates required in its implementation.

The delay associated with a reversible circuit is the number of such quantum gates in the critical path of the circuit where the critical path is the path connecting the input and output with maximum delay value. The total path delay can be calculated by counting the individual delay values for each quantum gate present in the critical path. It should be observed that the path delay is analogous to the logical depth of the reversible circuit. In other words, the logical depth provides the number of quantum gates in the path. Thus, the quantum cost of the circuit is the total number of quantum gates in the circuit while the delay is the number of quantum gates in the critical path of the circuit and thus both will be of different values [3].

Garbage outputs are the unutilized outputs in reversible circuits which exist just to maintain reversibility, but do not perform any useful operations. The reversible logic circuits must be constructed under two main conditions which involve avoiding fan-out problem and occurrence of loops or feedbacks. In any design these two problems are to be overcome and garbage outputs are mainly the result of such optimization [13].

DESIGN METHADOLOGY

A barrel shifter is a combinational logic circuit, which performs the operation of data shifting by a certain number of bits in one clock cycle. It can be implemented by connecting a sequence of multiplexers. As discussed earlier the Fredkin gate can serve as a 2:1 Multiplexer as its output is analogous to a 2:1 multiplexer output. Similarly a 4:1 Multiplexer and 8:1 Multiplexer can be designed [12] by connecting Fredkin gates in a sequential manner as shown in Figure-3(a) and Figure-3 (b).

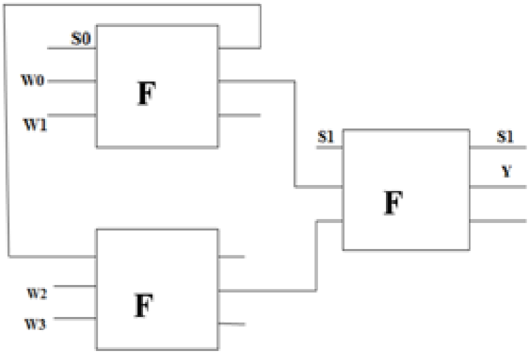

Figure-3(a). Design of 4:1 MUX using Fredkin gates [3].

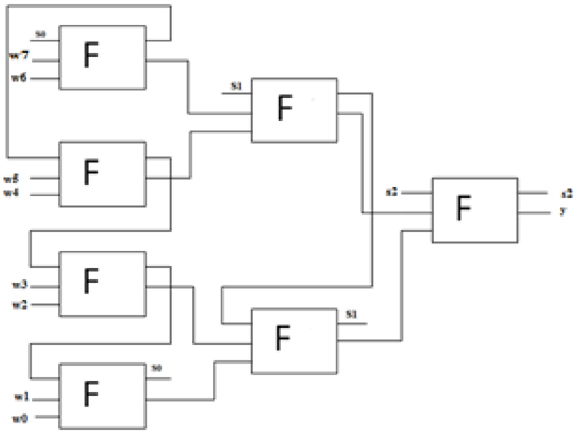

Figure-3(b). Design of 8:1 MUX using Fredkin gates.

This paper analysis two architectures, each of 4-bit and 8-bit Barrel shifter, a conventional design and an existing or quoted improved design.

Conventional (4, 2), (8, 3) Barrel Shifters

The design of (4, 2) barrel shifter is designed using 4:1 MUX [3] by successive replication of input signals to avoid fan-out as shown in Figure-3(c). Twelve Feynman gates are used for duplicating the input for the purpose of providing input lines to four 4:1 Multiplexers which are constructed as mentioned using Fredkin gates as shown in Figure-3(d).

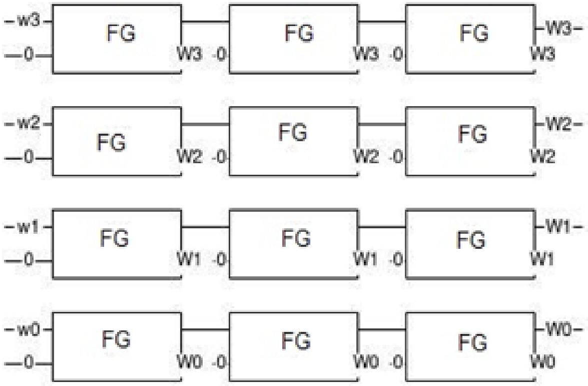

Figure-3(c). Replication of input signals to avoid fanout.

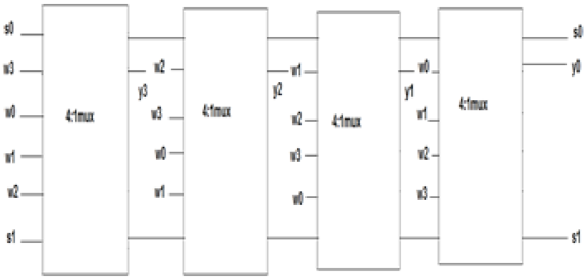

Figure-3(d). Architecture of conventional (4, 2) Barrel Shifter.

In the conventional (4, 2) and (8, 3) barrel shifter reversible design, the complexity associated with the construction is very high as the number of gates used is large in number.

Similarly an (8, 3) Barrel shifter is constructed using the same methodology. The conventional (8, 3) Barrel shifter is constructed using 56 Feynman gates and also makes use of 7, 8:1 Multiplexers using Fredkin gates [11]. The multiplexer is constructed as mentioned earlier in Figure-3(b). The architecture of conventional (8, 3) Barrel Shifter is as seen in Figure-3(g). The number of quantum gates is evidently high and the conventional (8, 3) Barrel Shifter design becomes inefficient.

Existing (4, 2) Bit Barrel Shifter

An existing improved design of a (4, 2) barrel shifter helps compensate the shortfalls of the conventional design which are high gate count, delay, complexity and efficiency. This design involves 4 Feynman gates and 6 Fredkin gates and a total of 6 Garbage output. The Quantum cost of the design is calculated to be 34. The design of this shifter is shown in Figure-3(e). It leads to a drastic reduction in quantum cost, time delay and the number of garbage outputs [10].

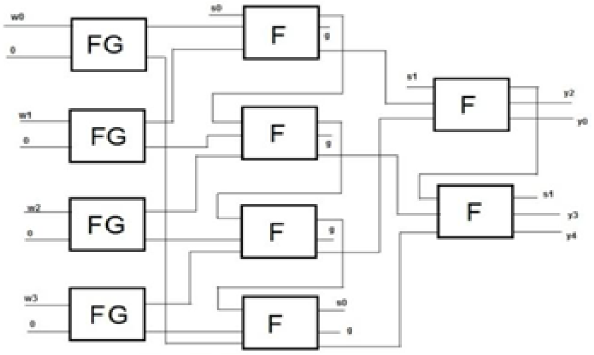

Figure-3(e). Existing design of (4, 2) Barrel Shifter.

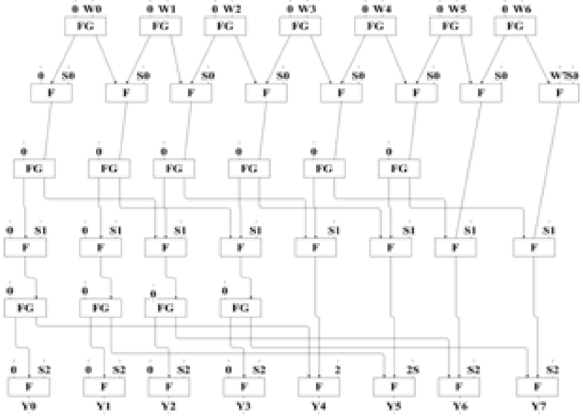

Figure-3(f). Improved design of (8, 3) Barrel Shifter [9].

Existing improved design of (8, 3) Barrel Shifter

The authors of [9] propose a new design of (8,3) Barrel Shifter that is composed of 24 Muxes which can be simplified into a circuit consisting of 17 Feynman gates and 24 Fredkin gates as shown in the Figure-3(f). The reduction in the gate count helps reduce the time delay and the associated quantum cost.

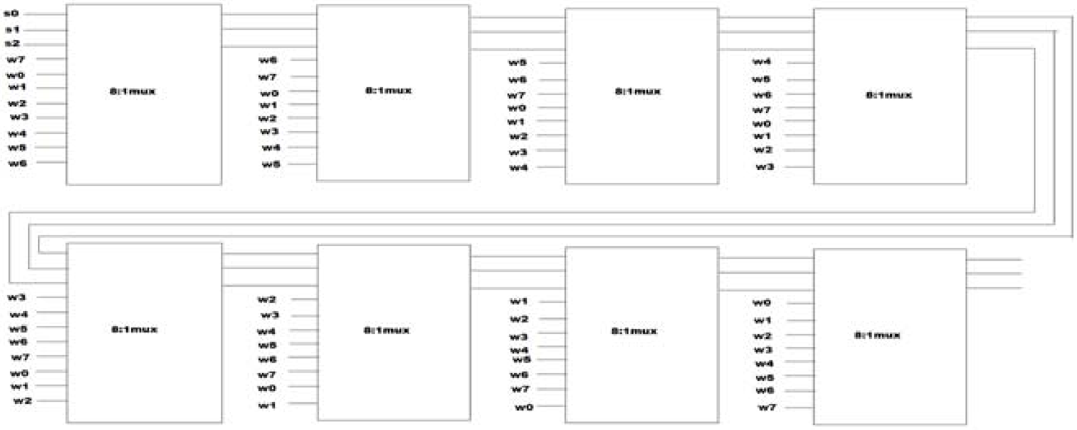

Figure-3(g). Architecture of conventional (8, 3) Barrel shifter.

RevKit implementation

The RevKit [15] implementation involves simulating the above mentioned design piece by piece by breaking down into simple reversible logic gates such as the Fredkin and Feynman gates. The libraries available in RevKit [15] help build complex reversible logic circuits and visualize its quantum level representation. It also helps in functional verification of the circuits, along with determining parameters such as quantum cost, delay, gate count, line count, garbage outputs and transistor cost. The quantum representation of the existing and reported [9] (4, 2) and (8, 3) barrel shifters are shown in Figures 4 (a) and 4 (b).

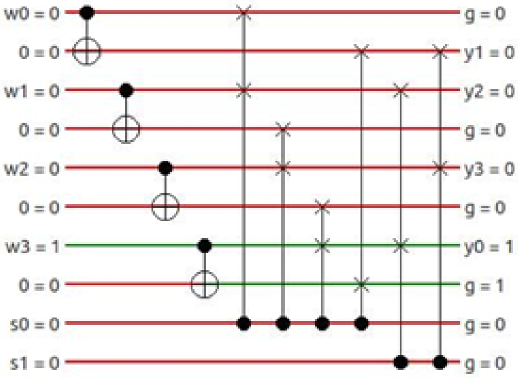

Figure-4(a). Quantum representation of improved (4, 2) Barrel Shifter.

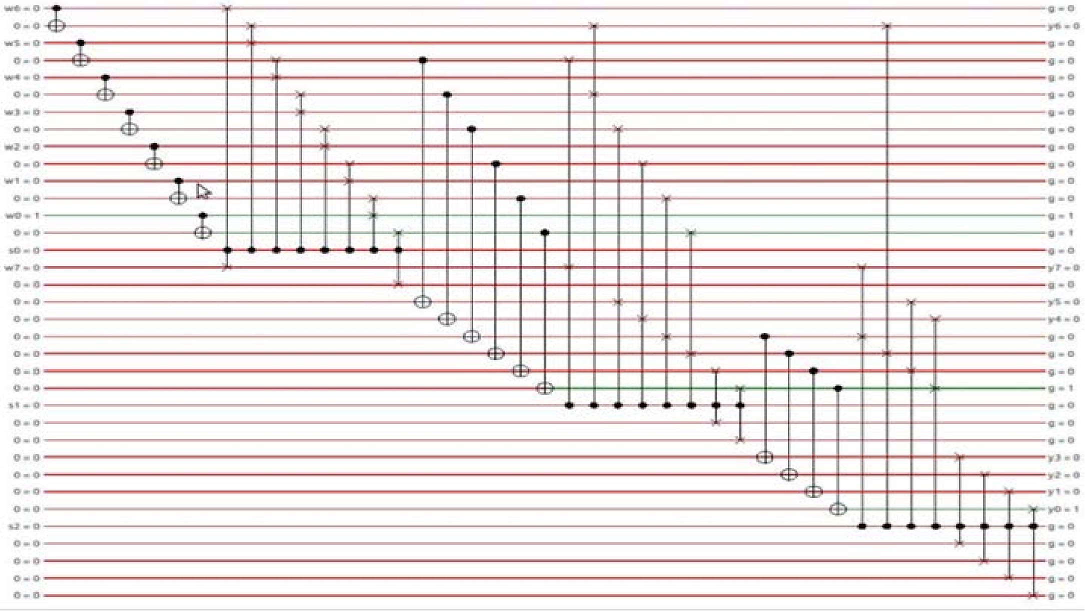

Figure-4(b). Quantum representation of improved (8, 3) Barrel Shifter.

Hence, using the designs cited in the paper, a successful verification of the barrel shifter was done using RevKit [15] software and the simulation results are tabulated in the succeeding section. The results also compare parameters such as quantum cost, time delay and garbage output with previous work and presents on how such parameters are reduced using the modified design.

RESULTS

The design was implemented in RevKit [15] software. The simulation results for 8-bit and 4-bit barrel shifters are as shown in Figures 4 (a) and 4 (b). The RevKit [15] software had built in library for reversible gates and the architecture was coded in Ubuntu platform and quantum level implementation of the architecture was designed.

The parameters such as Quantum cost, Time Delay, Garbage output, Gate count, Line count and Transistor cost were calculated using RevKit [15] and the obtained values are tabulated as shown in Table-1. The tabulations show a drastic reduction in all three vital parameters. Hence the optimization of quantum costs, time delay and garbage output for reversible gates are in consonance with the expected results.

Table-1. Comparative analysis.

Criteria	**Reversible (4,2) Barrel Shifter using 4:1 Mux** [3]	**Modified existing reversible (4,2) Barrel Shifter**	**Reversible (8,3) Barrel Shifter using 8:1 Mux**[3]	**Improved reversible (8,3) Barrel Shifter** [9]
Quantum Cost	72	34	336	137
Garbage Output	14	6	59	27
Delay	40	20	160	40
Gate Count	24	10	112	41
Line Count	18	10	67	35
Transistor Cost	192	80	896	328

CONCLUSIONS

In this work, 4-bit and 8-bit Reversible barrel shifters have been successfully implemented in RevKit [15]. The reversible barrel shifter was also contrasted and analyzed for parameters such as quantum cost, time delay and garbage outputs. The architecture in this paper was simulated and quantum level representation of both 4-bit and 8-bit reversible architecture was constructed. The paper also comprehensively compares the existing 4-bit and 8-bit architectures in terms of quantum cost, time delay garbage output, transistor cost, gate count, line count with that of the improvised designs [9] in this paper. The following are the important conclusions inferred in this paper from the simulation results

- The Quantum cost of 8-bit barrel shifter and 4-bit barrel shifter was approximately reduced by 59% and 52% respectively
- The Garbage output of 8-bit barrel shifter and 4-bit barrel shifter was approximately reduced by 54% and 57% respectively
- The Time delay for 8-bit barrel shifter and 4-bit barrel shifter was approximately reduced by 75% and 50% respectively.
- Similarly, as seen from Table I, there was a reduction in Gate count, Line Count and Transistor cost as well. Hence this improvised method can be highly useful in terms of energy consumption, efficiency and in terms of speed.

Hence a powerful approach for designing, simulating visualizing and implementing reversible logic circuits and specifically reversible barrel shifters using a tool called RevKit [15] has been shown in this paper. Additional parameters that are not typically evaluated during circuit designs have been tabulated and compared.

REFERENCES

[1] R. Landauer. 1961. Irreversibility and heat generation in the computational process. IBM J. Research and Development. 5: 183-191.

[2] C.H. Bennett. 1973. Logical reversibility of computation. IBM J. Research and Development. 17: 525-532.

[3] S. Kotiyal, H. Thapliyal and N. Ranganathan. 2011. Design of a reversible bidirectional barrel shifter. In Proceedings of the 11th IEEE International Conference on Nanotechnology, Portland, Oregon, USA. pp. 463-468.

[4] W. N. Hung, X. Song, G.Yang, J.Yang, and M. Perkowski. 2006. Optimal synthesis of multiple output boolean functions using a set of quantum gates by symbolic reachability analysis. IEEE Trans. Computer-Aided Design. 25(9): 1652-1663.

[5] DiVincenzo D.P., Shor P.W. 1996. Fault-tolerant error correction with efficient quantum codes. Phys. Rev. Lett. 77: 3260-3263.

[6] D.M. Miller, D. Maslov and G.W. Dueck. 2003. A transformation based algorithm for reversible logic synthesis. Proceedings of the IEEE Design Automation Conference. pp. 318-323.

[7] Himanshuthapliyal and Nagarajan ranganathan. 2010. Design of Reversible Sequential Circuits Optimizing Quantum Cost, Delay and Garbage Outputs, University of South Florida, Tampa ACM Journal on Emerging Technologies in Computer Systems. 6(4): Article 14, Pub. Date.

[8] O. Anjaneyulu, T. Pradeep and C.V. Krishna Reddy2. 2012. Design of An Efficient Reversible Logic Based Bidirectional Barrel Shifter. International Conference on Electronics and Communication Engineering, 20th, May Bangalore, ISBN: 978-93-81693-29-2.

[9] Ch. Srigowri, M. Srihari. 2013. VLSI Design of Low Power Reversible 8-bit Barrel Shifter. IJTEL, ISSN: 2319-2135. 2(6).

[10] Thumma Pradeep. 2012. Logarithmic Reversible Barrel Shifter. IEEE International Conference on Advanced Communication Control and Computing Technologies (ICACCCT).

[11] Hashmi and H. Babu. 2010. An efficient design of a reversible barrel shifter. In VLSI Design, 2010. VLSlD '10. 23rd International Conference on. pp. 93-98.

[12] S. Gorgin and A. Kaivani. 2007. Reversible barrel shifters. In Proc. 2007 Int. I. Conf. On Computer Systems and Applications, Amman. pp. 479-483.

[13] D.P. Vasudevan, P.K. Lala, J. Di, and J.P Parkerson. 2006. Reversible-logic design with online testability. IEEE Transaction on Instrumentation and Measurement. 55(2): 406-414.

[14] H. Wood and D.J. Chen. 2004. Fredkin gate circuits via recombination enzymes. Proceedings of Congress on Evolutionary Computation (CEC). 2: 1896-1900.

[15] Mathias Soeken, Stefan Frehse, Robert Wille, Rolf Drechsler. RevKit: 2012. An Open Source Toolkit for the Design of Reversible Circuits. Reversible Computation 2011, Lecture Notes in Computer Science. 7165.