Efficient Approach to design Reversible Logic Blocks for Field Programmable Gate Arrays

Abu Sadat Md. Sayem
Department of Computer Science
University of Western Ontario
London, Canada
asayem@csd.uwo.ca

Sajib Kumar Mitra
Department of Computer Science and Engineering
University of Dhaka
Dhaka- 1000, Bangladesh
sajibmitra.csedu@yahoo.com

***Abstract*— Field-Programmable Gate Arrays (FPGAs) are considered as the assertive digital implementation medium as measured by design starts. The ability for designers to avoid the pitfalls of Nanoelectronic design and changing the design until last minute made FPGAs more demanding in recent years. But consumption of much area and power has been contemplated as the major drawback for FPGAs over Application Specific ICs (ASIC). In this paper, we have designed the logic block of a Plessey FPGA in reversible manner with reduced number of reversible gates, garbage outputs and quantum cost. As reversible computing reduces the power consumption of any system, this design approach will definitely resolve the power problem for FPGAs. Our proposed design uses the most cost effective reversible circuits including proposed 3*3 MG (MUX Gate) in comparison with the existing ones in literatures.**

Keywords-Reversible Logic; MUX Gate; Fredkin Gate; Plessey FPGA

I. INTRODUCTION

Computation systems which are not logically reversible typically lose information in the process of execution. According to the laws of physics information cannot really be being lost, it must be being translated into another form. That form is usually heat. So, loss of information results power dissipation. To reduce this power dissipation reversible logic was introduced. The main idea of reversible logic is to allow the construction of reversible computer by using components which preserve information content, and can thus potentially be run backwards. Hence, by implementing reversible designs of computer hardware significant amount of heat can be reduced. It has been proved that for irreversible logic computations, each bit of information lost generates *kTln2* joules of heat energy, where *k* is Boltzmann's constant and *T* the absolute temperature at which computation is performed [1], whereas the reverse term was showed by Bennett, that *kTln2* energy dissipation would not occur if the computation were carried out in a reversible manner [2]. Reversible logic is likely to be in demand in high speed power aware circuits and low-power CMOS design.

A Field-Programmable Gate Array (FPGA) is a semiconductor device that can be configured by the customer or designer after manufacturing—hence the name "field-programmable". FPGAs can be used to execute any logical function that an Application-Specific Integrated Circuit (ASIC) could perform, but the ability to update the functionality after shipping offers advantages for many applications. FPGAs contain programmable logic components called "logic blocks", and a hierarchy of reconfigurable interconnects that allow the blocks to be "wired together"—somewhat like a one-chip programmable.

In this paper we have proposed a reversible design for the logic block of Plessey FPGA, which will reduce the amount of power without any distrust. The individual elements of the design for the logic block are most optimized in terms of number of gates and garbage. We have designed reversible multiplexer (MUX) and D-Latch along with Master Slave Flip Flop by using proposed gate called MG (MUX Gate). Also we have proposed the design of Random Access Memory (RAM) followed by reversible decoder. Every component to build the logic block is proposed with the minimization of cost factors (gates, garbage and quantum cost) for getting better throughput.

The rest of the paper is presented on following perspectives: Section II discusses some background study about Reversible Logic, Quantum Cost and FPGA. Section III shows the organization of proposed MG and the different components of the logic block. In Section IV we propose the design of the logic block with the components of Section III with a brief discussion how the circuit would behave and work to perform the operation of a logic block of Plessey FPGA. Section V describes the results of the work and finally we end the paper with concluding remarks.

II. BASIC DEFINITIONS AND PROPERTIES

Here, we have presented the basic definitions and ideas related to Reversible Logic, Quantum Cost and FPGA.

Definition 1: A gate is **Reversible Gate** if and only if, the vectors of input and output domains have one-to-one mapping. Fig. 1 shows the block diagram of a 2 * 2 Reversible gate and corresponding unique input-output mapping.

Definition 2: A garbage bit is an additional output that makes an *n*-input *m*-output function reversible. That is, unwanted or unused output of reversible gate (or circuit) is known as **Garbage Output**. In the following figure (Fig. 1), reversible EX-OR operation requires another dummy output (*p*) which never be used except reversibility.

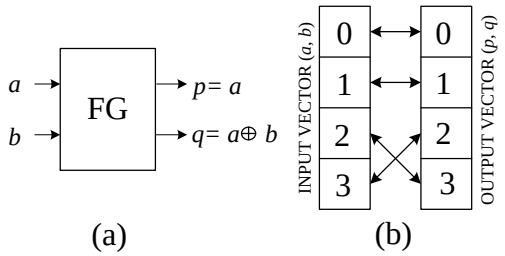

Fig. 1. Reversible Gate: (a) Feynman Gate and (b) Input-output mapping of Feynman Gate

Definition 3: The input vector, I_v and output vector, O_v for 2 * 2 **Feynman Gate (FG)** [3] is defined as: $I_v = (a, b)$ and $O_v = (a, a \oplus b)$. Fig. 1 shows the design of Feynman gate and corresponding mapping.

Definition 5: The input vector, I_v and output vector, O_v for 3 * 3 **Toffoli Gate (TG)** [5] is defined as: $I_v = (a, b, c)$ and $O_v = (a, b, ab \oplus c)$. Fig. 2(a) shows the design of Toffoli Gate.

Definition 6: Input vector, I_v and output vector, O_v of 3 * 3 **Fredkin Gate (FRG)** [6] is defined as (shown in Fig. 2(b)): $I_v = (a, b, c)$ and $O_v = (a, a'b \oplus ac, a'c \oplus ab)$.

Definition 7: Input vector, I_v and output vector, O_v for 3 * 3 **Peres Gate (PG)** [7] is defined as (shown in Fig. 2(c)): $I_v = (a, b, c)$ and $O_v = (a, a \oplus b, ab \oplus c)$.

Definition 8: The **Field Programmable Gate Array (FPGA)** is a device that contains a matrix of reconfigurable gate array logic circuitry [8]. A single FPGA can replace thousands of discrete components by incorporating millions of logic gates in a single integrated circuit (IC) chip. Fig. 3 shows the generalize structure of FPGA with Logic block and Input-Output block.

For the quantum analysis the basic algorithm and the template matching technique [9] is used. According to [10] each of the 2-qubit gates (Quantum Exclusive-OR, Controlled-*V*, Controlled-*V*+) has a quantum implementation cost of 1. In addition, when both quantum XOR and controlled-*V* (*V*+) are operated on the same two qubits in a symmetric pattern, their total cost is considered as 1 as well.

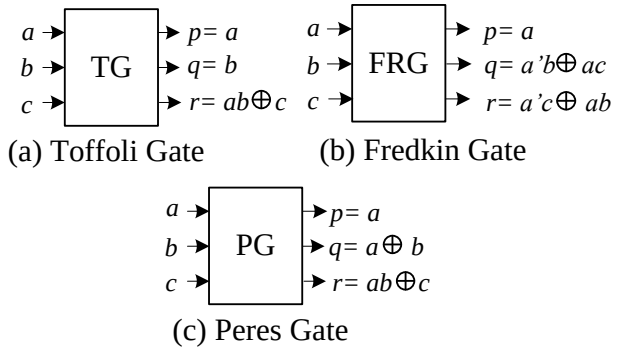

Fig. 2. Popular Reversible gates are used in Reversible Computing

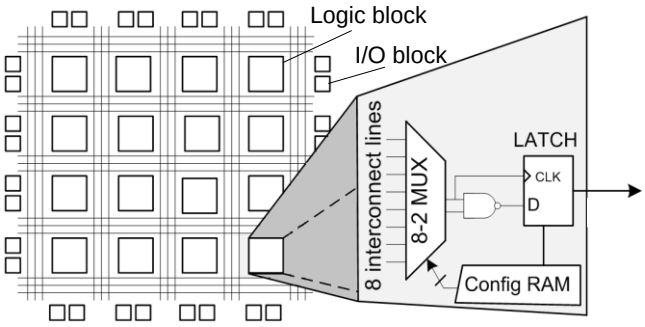

Fig. 3. Simplified Internal Structure of FPGA and internal structure of Logic block

III. Reversible Component of the Proposed Logic Block

In this section, we discussed the design of the different components that are required to build the logic block of Plessey FPGA. We need an 8-to-2 Multiplexer as the two input of NAND of the logic blocks are connected to two 4-to-1 MUXs; so in total we need two 4-to-1 MUXs, D-latch, and Reversible Random Access Memory (RRAM) for the proposed design.

A. Proposed Reversible MUX Gate (MG)

We have proposed a new reversible gate which can be efficiently used to design reversible MUX with minimum cost than the existing designs. The gate is named as MUX Gate (MG). Fig. 4(a) shows the pictorial representation and Fig. 4(b) overviews the quantum realization of the proposed MG. The resulting Quantum Cost is 4. The verification of reversibility of MG can be defined as Fig. 4(c) where the value of input or output vector is represented as equivalent decimal number.

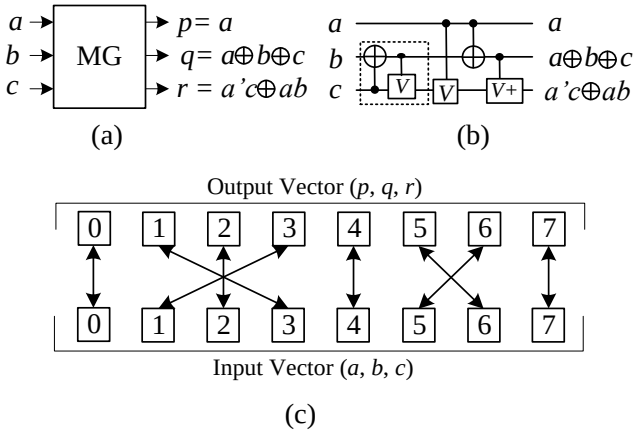

Fig. 4. Proposed design of 3 * 3 Reversible MUX Gate (MG): (a) 3 * 3 MUX Gate, (b) Quantum representation of MG (Quantum cost is 4) and (c) One to One mapping between input-output vectors of MUX Gate

B. Proposed 4-to-1 Reversible MUX

A Multiplexer or MUX is a device that performs multiplexing; it selects one of many analog or digital input signals and forwards the selected input into a single line. A multiplexer of 2^n inputs has n select lines, which are used to select which input line to send to the output. For the proposed design of reversible logic block we need an 8-to-2 MUX.

To design the 8-to-2 reversible MUX we used two 4-to-1 MUXs. The proposed design of the 4-to-1 reversible MUX can be realized using only three MGs (shown in Fig. 5) which produces three garbage outputs and the quantum cost is 12. Any *n*: 1 reversible MUX can be designed with this design approach with n-1 MGs. In [11] *n*: 1 reversible MUX was realized using (*n*-1) FRGs. But FRG has quantum cost of 5 where MG has 4. So the cost of this design is better than the existing design of MUX in [11].

C. Design of Proposed Reversible D-Latch

Equation for a reversible D-Latch is $Q^* = D.C + C'Q$. The D-Latch is used to capture, or ``latch" the logic level which is present on the Data line when the clock input is HIGH. If the data on the *D* line changes state while the clock pulse is high, then the output, *Q* follows the input *D*. We have proposed a reversible D-Latch with one MG and one FG depicted in Fig. 6 which has quantum cost of 5. In the design of [12] D-Latch with only output *Q* requires 2 reversible gates which produces 2 garbage outputs and has total quantum cost of 6. In [13] D-latch was proposed by 1 gate but has quantum cost of 7 with 2 garbage outputs. As quantum cost and garbage output are the two important cost measurements in reversible logic circuit design ,our proposed design for the D-Latch is better than the design of [12] and [13].

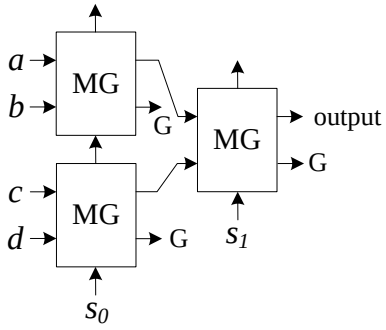

Fig. 5. Design of the proposed 4-to-1 reversible MUX

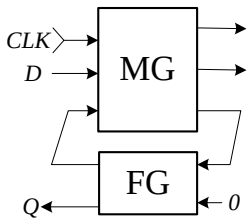

Fig. 6. Proposed D-Latch with two gates

D. Design of Proposed Reversible D-Latch

The design of the RRAM consists of three parts; *n*-to-2^n decoder, WRITE enabled Master Slave Flip Flop, and the D-Latch. We have already shown the design of D-Latch in prior section. We will use the same D-Latch here too.

1) Design of Reversible n-to-2^n Decoder: The design steps of the *n-to-*2^n decoder are shown in Fig. 7(a-c). Only one FG can be realized as a 1-*to*-2 decoder. 2-*to*-4 reversible decoder can be designed with one FG and two FRGs with only one garbage output. With one FG and 6 FRGs 3-to-8 decoder can be realized. So, with 1 FG and increasing the number of FRGS we can design any *n-to-*2^n reversible decoder. When $n > 1$ we can design any *n*-to-2^n decoder by the design approach of Fig. 8 (b-c) with 1 FG and (2^n-2) FRGS which will produce (*n*-1) garbage outputs. With the common characteristics we can state Theorem 1.

Theorem 1: *To design an n-to-2^n reversible decoder with FG and FRG if the number of required FG is N_{FG}, number of FRG is N_{FRG}, garbage output is GB and the total delay is D, then,*

$$N_{FG} = 1;\ N_{FRG} = (2^n - 2);\ GB = (n-1);\ D = (2^{n-1} + n - 1)$$

Proof: According to Fig. 8 for a reversible *n-to-2^n* decoder if we design with FG and FRG total number of FG will be 1, for all the steps, when n>1; so N_{FG} is 1.for *n*= 2, 3, 4, 5, …… Number of required FRG will be 2, 6, 14, 30, …. Thus for any number *n* the N_{FRG} will be (2^n-2). A 2-to-4 decoder produces (2-1) =1 garbage output; 3-to-8 produces (3-1)= 2 garbage outputs, so the total number of garbage output for any *n-to-*2^n reversible decoder is (*n*-1). Now delay for 1-to-2 decoder for the above design is 1, 2-to-4 is 3, 3-to-8 is 6, 4-to-16 is 11. So Delay, *D*= $2^{(n-1)}$ + *n*-1. Finally we can say that;

$$N_{FG} = 1;\ \ N_{FRG} = (2^n - 2);$$
$$GB = (n-1);\ \ D = (2^{n-1} + n - 1)$$

□

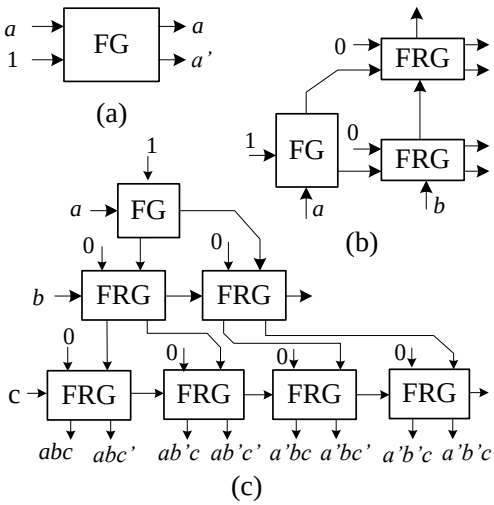

Fig. 7. Steps of designing reversible *n-to-2^n* decoder: (a) FG as 1-to-2 Decoder, (b) DSG as 2-to-4 Decoder and (c) 3-to-8 Decoder with DSG and FRGs

2) Design of Reversible WRITE Enabled Master Slave Flip Flop: WRITE enabled Master Slave FFs are the most substantial part of a Reversible Random Access Memory (RRAM). As data is both read from and written into RAM, each Flip-flop should work on two modes: read and write. We proposed a design of reversible WRITE enable Master Slave FF, which is shown in Fig. 8. The design was implemented by two MGs, two FGs and one TG with a delay of 5 which produces only 2 garbage outputs.

Finally the design of the proposed Reversible RAM is shown in Fig. 9. A RAM is a two dimensional array of Flip-flops. There are 2^n rows while each row contains m Flip-flops. Each time only one of the 2^n output lines of the decoder is active which selects one row of FFs of the RAM. Whether a read or a write operation is performed depends on the W input. When W is HIGH, m FFs of the selected row of the RAM is written with the inputs D_1 to D_m. When W is LOW, Q_1 to Q_m contains stored bits in the Flip Flops of the selected row and simultaneously the Flip Flops are refreshed with the stored bits.

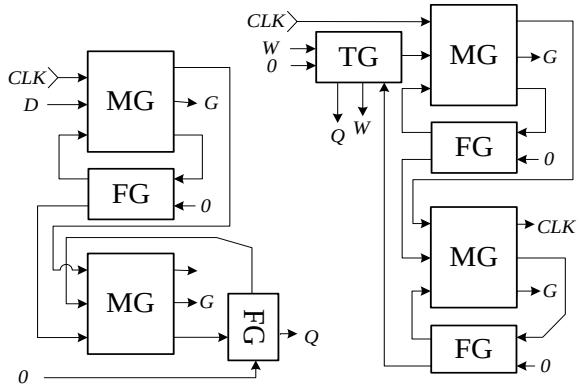

Fig. 8. (a) MS configuration of D-Flip Flop and (b) proposed WRITE enabled MS FF

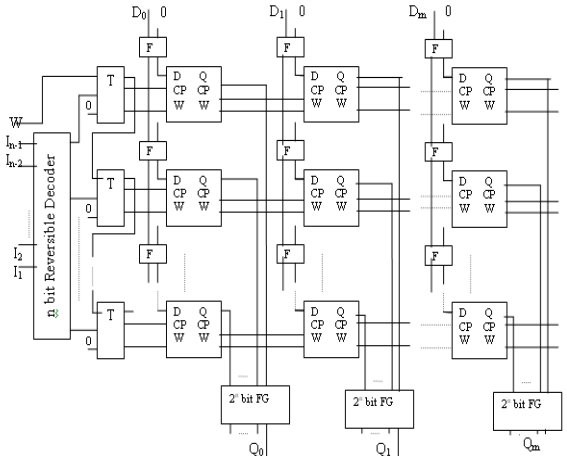

Fig. 9. The $m*2^n$ reversible Random Access Memory

IV. Design of Reversible Logic Block of Plessey FPGA

Now we have all the components that are required to realize the logic block for Plessey FPGA. Fig. 10 shows the block diagram of proposed design of the reversible logic block. The actual operation details of logic blocks of FPGA are beyond the scope of this paper. We tried to give a brief discussion here.

Our proposed design has very simple and efficient working method to operate as logic block for FPGA. In FPGA All the logic blocks are connected with programmable wires. Although not shown here, A Wire Segment is a wire unbroken by programmable switches. One or two switches may attach to the wire segment. Each end of a wire segment typically has a switch attached. A Track is a sequence of one or more wire segments in a line. A Routing channel is a group of parallel tracks. There are two kinds of blocks one master and another slave illustrated as 'M' and 'S' respectively. Also, the rows of logic blocks alternate in their direction of flow [14].

In Fig. 10, PG acts as the NAND gate. The multiplexers are controlled by RRAM cells. Each input of the NAND gate comes from a 4-to-1 multiplexer. The inputs to each multiplexer are connected to either the output of the previous NAND gate in the row or the output of the NAND gate above or below this logic block, whichever is closer, the other blocks are connected similarly, which are not shown here.

There are three vertical long tracks per column of logic block and two horizontal long tracks per row. There are two types of long tracks in each direction: short range tracks which travel 10 blocks and long range tracks which travel the width or length of the entire chip [14].

In summary, two inputs of the NAND come from the two reversible 4-to-1 MUXs which are connected to prior logic blocks. The MUX operation is signaled and controlled by the configuration RRAM. Then the NAND does the logic operation and stores in the D-Latch which will be send to next logic block again. With this design approach all the logic blocks can be designed in reversible manner.

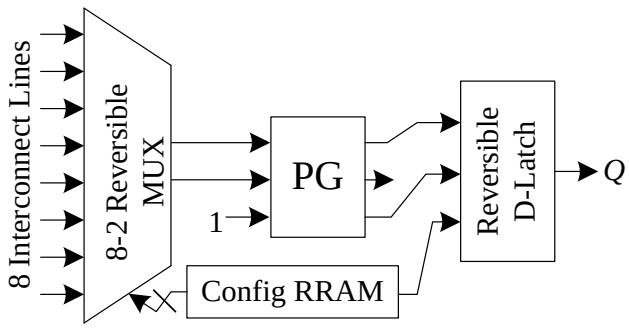

Fig. 10. Design of the proposed reversible logic block of Plessey FPGA

V. Result and Discussion

A. Evaluation of Proposed new Reversible Gate

Our proposed MUX gate (MG) can be used to design any *n*:1 reversible MUX. This gate can map any Boolean functions which were done by FRG before in various reversible circuits. But it has less quantum cost than FRG. So FRG can be effectively replaced by this gate for any design with reduced quantum cost. With the operation ability of FRG and less quantum cost this gate can significantly contribute in reversible community.

B. Evaluation of the Reversible Component of Logic Block of FPGA

With careful literature review, we can show that the reversible components used to design the final logic block are so far the most optimized design with compared with the existing ones. The reversible 4-to-1 MUX requires total 3 gates, which has total quantum cost of 12, delay 3 and produces 3 garbage outputs. We can design any *n*:1 reversible MUX with *n*-1 MGs. The comparison with the existing design of n: 1 MUX is shown in Table I.

TABLE I. Comparison of different n:1 MUXs

***n*:1 MUX**	**Component and cost**		
	No of gates	***Garbage Output***	***Quantum Cost***
Existing [11]	*n*-1	3	5*(*n*-1)
Proposed	*n*-1	3	4*(*n*-1)

The proposed D-latch is designed with only two reversible gates which have two garbage outputs, delay of 2 and quantum cost of 5. The most cost effective design for reversible D-latch for only output *Q* is proposed in [12] uses 1 reversible gate with 2 garbage outputs and quantum cost of 7. So, our design of D-latch is better than [12] in terms of the number of garbage outputs and quantum cost and Table II shows the comparison.

TABLE II. Comparison of different Reversible D-Latch with onlt output Q

D-Latches	**Components and cost**		
	No of gates	***Garbage Output***	***Quantum Cost***
Existing [12]	2	2	6
Existing [13]	1	2	7
Proposed	2	2	5

Our proposed *n-to-*2^n decoder can be designed with FG and FRGs with generalize equation. Any *n-to-*2^n reversible decoder can be designed with FG and FRG with theorem 1. The proposed master Slave WRITE enabled FF is designed with total 5 reversible gates, which has delay of 5 produces only 2 garbage outputs. The total quantum cost of the design is 15. The reversible RAM of Fig. 9 is an $m*2^n$ RAM, where *m* is the number of selection bits and *n* is the number of input bits. This design requires total $2^n*(m+2)-3$ reversible gates. So all the individual components we used to design the proposed logic block for the FPGA are optimized with the cost parameters.

C. Evaluation of the Reversible FPGA

With all the reversible elements of section III, we designed the final logic block for Plessey FPGA in section IV. It can operate as a logic block and obviously in reversible manner. The operation is briefly described in section IV with the figure.

Conclusions

FPGAs are better than ASICs in all aspects but power and area. In this paper we proposed an efficient reversible design for the logic blocks of FPGA. For designing low power digital circuit reversible logic is widely used in recent years. All the reversible components those were used in this paper are cost effective than the existing ones in literature. In future all other parts of the FPGA like I/O blocks, routing wires need to be designed in reversible way to make the whole FPGA reversible.

References

[1] R. Landauer, "Irreversibility and heat generation in the computing process", IBM J. Research and Development, vol- 5, pp. 183-191, 1961.

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

[3] R. Feynman, "Quantum mechanical computers", Optical News, vol- 11, no- 2, pp. 11–20, 1985.

[4] M. Haghparast and K. Navi, "A Novel Fault Tolerant Reversible Gate for Nanotechnology Based Systems", Am. J. Applied Sci., vol- 5, no- 5, pp. 519-523, 2008.

[5] T. Toffoli, "Reversible computing", MIT Lab for Comp. Sci., vol- 85, pp. 632-644, 1980.

[6] P. D. Picton, "Fredkin Gates as the Basic for Comparison of Different Logic Designs", Synthesis and optimization of Logic Systems, IEE Colloquium on, London, 1994.

[7] A. Peres, "Reversible logic and quantum computers", Physical Review, vol. 32, pp. 3266–3276, 1985.

[8] J. Rose, A. Gamal and etal, "Architecture of Field-programmable gate arrays", Proceedings of the IEEE, vol. 81, Issue 7, pp. 1013 – 1029, Jul, 1993.

[9] D. M. Miller, D. Maslo and G. D. Dueck, "A Transformation Based Algorithm for Reversible Logic Synthesis", Annual ACM IEEE Design Automation Conference, Proceedings of the 40th annual Design Automation Conference, Anaheim, CA, USA, pp. 318 – 323, 2003.

[10] W. N. N. Hung, X. Song, G. Yang, J. Yang and P. Mark, "Optimal synthesis of multiple output Boolean functions using a set of quantum gates by symbolic reachability analysis", IEEE transactions on Computer-Aided Design of integrated circuits and Systems, vol. 25, No. 9, pp. 1652-1663, September 2006.

[11] H. Thapliyal and N. Ranganathan, "Concurrently Testable FPGA Design for Molecular QCA Using Conservative Reversible Logic Gate", Proc. ISCAS 2009, Taipei, pp. 1815 - 1818, May 2009.

[12] H. Thapliyal and N. Ranganathan, "Design of Reversible Latches Optimized for Quantum Cost, Delay and Garbage Outputs", 23rd International Conference on VLSI Design, pp.235-240, 2010.

[13] A. S. M. Sayem and M. Ueda, "Optimization of reversible sequential circuits", Journal of Computing, vol. 2, No. 6, NY, USA, ISSN 2151-9617, June 2010.

[14] J. Rose and S. Brown, "Flexibility of interconnection structures for field-programmable gate arrays", vol. 26, Issue 3, pp. 277 – 282, March 1991.