Realization of a Novel Fault Tolerant Reversible Full Adder Circuit in Nanotechnology

Saiful Islam, Muhammad Mahbubur Rahman, Zerina Begum, and Mohd Zulfiquar Hafiz
Institute of Information Technology, University of Dhaka, Bangladesh

Abstract: *In parity preserving reversible circuit, the parity of the input vector must match the parity of the output vector. It renders a wide class of circuit faults readily detectable at the circuit's outputs. Thus reversible logic circuits that are parity preserving will be beneficial to the development of fault tolerant systems in nanotechnology. This paper presents an efficient realization of well known Toffoli gate using only two parity preserving reversible gates. The minimum number of garbage outputs and constant inputs required to synthesize a fault tolerant reversible full adder circuit has also been given. Finally, this paper presents a novel fault tolerant reversible full adder circuit and demonstrates its superiority with the existing counterparts.*

1. Introduction

Irreversible logic circuits dissipate heat for every bit of information that is lost. Information is lost when the input vector cannot be uniquely recovered from the output vector. Theoretically reversible logic dissipates zero power since the input vector of reversible circuit can be uniquely recovered from the output vector [1]. According to Bennet [1] zero energy dissipation would be possible only if the network consists of reversible gates. Thus reversibility will become an essential property in future circuit design. Reversible computation has applications in digital signal processing, low power CMOS design, cryptography, communication, nanotechnology, and quantum computing. Synthesis of reversible logic circuits differs significantly from classical logic circuits in many ways. In reversible circuits there should be no fan-out, that is, each output will be used only once, for each input pattern there should be a unique output pattern, and the resulting circuit must be acyclic [7, 12].

Fault-tolerance is the property that enables a system to continue operating properly in the event of the failure of some its components. If the system itself made of fault tolerant components, then the detection and correction of faults become easier and simple. In communication and many other systems, detection of fault is achieved by parity. Therefore, parity-preserving reversible circuits will be the future design trends towards the development of fault tolerant reversible systems in nanotechnology. A gating network will be parity preserving if its individual gate is parity preserving [10]. This paper presents an efficient realization of well known Toffoli gates using only two parity preserving reversible gates. Theoretical minimum number of garbage outputs and constant inputs required to design a fault tolerant full adder circuit has also been established. Finally, this paper presents a novel realization of fault tolerant reversible full adder circuit and demonstrates its superiority than the existing designs.

2. Reversible Logic Gates

A gate or a circuit is called reversible if there is a one-to-one correspondence between its input and output patterns. That is, a reversible gate performs only the permutation of its input vectors. If a reversible gate has k inputs, and therefore k outputs, then we call it a $k*k$ reversible gate. A $k*k$ gate is said to have width k. There are $(2^k)!$ possible reversible gates of width k. Particular realization of an arbitrary function using gates of greater width may dramatically decrease the gate count and garbage outputs but at the same time quantum realization of these gates may be costlier. Realization of reversible function using gates with smaller width increases the gate count and garbage outputs. Therefore, there must be a trade off of using a family of reversible gates. There are many reversible gates in the literature. Among them are 2*2 Feynman Gate (FG) [3], 3*3 Peres Gate (PG) [11], 3*3 Toffoli Gate (TG) [14], 3*3 FRedkin Gate (FRG) [4], 3*3 Khan Gate (NG) [8], 3*3 double Gate (F2G) [10], and 3*3 NFT [5]. Any reversible circuit may consist of one or more reversible gates and realizes only the function that is reversible. To make an (n, k) irreversible function reversible we must add some constants at the input side and some garbages at the

output side. The relation between garbage outputs and constants has been shown in [7] as

Inputs + Constants = Outputs + Garbages

Generally, garbages are outputs that are not used as primary outputs. Any realization techniques should keep both the number of constants and garbages as low as possible.
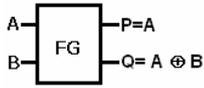

Figure 1. Feynman gate.
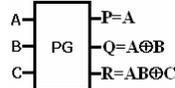
Figure 2. Peres gate.
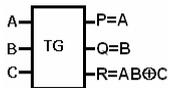
Figure 3. Toffoli gate.
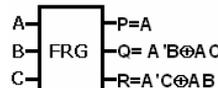
Figure 4. Fredkin gate.
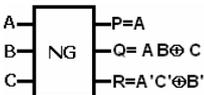
Figure 5. Khan gate.
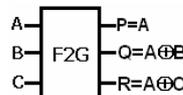
Figure 6. Feynman double gate.
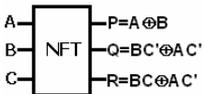
Figure 7. New fault tolerant gate.
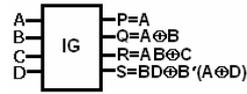

Figure 8. Islam gate.

2.1. Parity Preserving Reversible Logic Gates

Parity checking is one of the widely used error detection mechanisms in digital logic and data communication systems. This is because most of the arithmetic functions is not parity preserving. If the parity of the input data is maintained throughout the computation, no intermediate checking would be required [10]. There are some restrictions of using conventional methods of error detection in reversible circuits. Because in reversible network no fan-out is allowed and if the conventional approaches are used in reversible circuits, it may increase the number of gates and garbage outputs produced. Not only that, quantum realization of the network may be costlier and difficult. Since a reversible gates have the equal number of input and output lines, a sufficient requirement for parity preservation of a reversible gate is that each gate be parity preserving [10]. Thus, we need parity preserving reversible logic gates to construct parity preserving reversible circuits. A few parity preserving logic gates have been proposed in the literature. Among them 3*3 FRG [4] depicted in Figure 4, and 3*3 Feynman F2G [10] depicted in Figure 6 are one-through gates, which means one of the inputs is also output. Haghparast and Navi [5] have proposed a new 3*3 parity preserving reversible gate, namely NFT depicted in Figure 7. From Table 1, 2, and 3 we can see that the gates FRG, F2G and NFT are parity preserving since they satisfy

$A \oplus B \oplus C = P \oplus Q \oplus R$. And any $k*k$ reversible logic gate where the EX-OR of the inputs matches the EX-OR of the outputs will be parity preserving.

Table 1. Truth table of the parity preserving FRG.

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

Table 2. Truth table of the parity preserving F2G.

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

Table 3. Truth table of the parity preserving NFT.

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

2.2. A Proposal for a New Parity Preserving Reversible Gates

This paper presents a new 4*4 parity preserving reversible gate, Islam Gate (IG), depicted in Figure 8. The gate is one through, which means one of the input variables is also output. The corresponding truth table of the gate is shown in Table 4. It can be verified from the truth table that the input pattern corresponding to particular output pattern can be uniquely determined. The proposed reversible IG gate is parity preserving. This is readily verified by comparing the input parity $A \oplus B \oplus C \oplus D$ to the output parity $P \oplus Q \oplus R \oplus S$. The newly proposed IG gate is universal in the sense that it can be used for implementing any arbitrary Boolean function as shown in Figure 9.

a. IG as Inverter and EX-OR.   b. IG as AND and EX-OR.
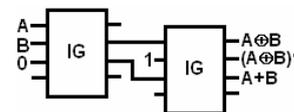
c. IG as EXOR, EX-NOR and OR

Figure 9. Proposed IG gate can implement all boolean functions.

There are other two 4*4 reversible gates in the literature, namely TSG [15] and MKG [16]. Though these two gates are reversible but not parity-preserving and therefore do not allow detection of circuit's faults.

2.3. Conservative Reversible Logic Gates

A $k*k$ reversible gate is conservative if the Hamming weight (number of logical ones) of its input equals the Hamming weight of its output [2]. A consequence of a gate's reversibility and conservability is that conservative reversible gates are zero-preserving and one-preserving. A gate is zero-preserving if all zeros input produces all zeros outputs. A gate is one-preserving if all ones input produce all ones outputs. There may have gates that are conservative but not reversible. A conservative reversible gate is a gate that is both conservative and reversible. From Table 1 we can see that FRG is zero-preserving and therefore conservative.

Table 4. Truth table of the proposed reversible IG.

A	B	C	D	P	Q	R	S
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	1
0	1	0	0	0	1	0	0
0	1	0	1	0	1	0	1
0	1	1	0	0	1	1	0
0	1	1	1	0	1	1	1
1	0	0	0	1	1	0	1
1	0	0	1	1	1	0	0
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	0
1	1	1	1	1	0	0	1

The conservativeness property divides the input vectors of $k*k$ reversible gates into $k+1$ groups having 0 to k ones (or k to 0 zeros). Any conservative reversible gate permutes the vectors of each group separately. For example, if k=3 we will have
$$N_c = (3_{C_0})!+(3_{C_1})!+(3_{C_2}) + (3_{C_3})!= 36$$
parity preserving reversible gates. From the above definition of parity-preserving and conservative reversible gates, we can deduce the following two lemmas.

Lemma 1: The necessary and sufficient condition of a gate to be conservative is that it must be zero-preserving or one-preserving.

Lemma 2: A conservative reversible gate is parity-preserving.

F2G and NFT are parity preserving but not conservative. The only conservative reversible logic gate in the literature is Fredkin gate, FRG. Fredkin gate is also parity-preserving since it satisfies the equation $A \oplus B \oplus C = P \oplus Q \oplus R$. In this study, we have tested all $k*k$ reversible logic gates for k= 2, 3, 4 and 5 to see whether they are parity-preserving (and conservative) or not and the statistics is given in Table 5. To generate parity-preserving and conservative reversible gates the following algorithm is used.

2.4. Design of Parity Preserving TG Circuit

Toffoli gate [14], depicted in Figure 3, is one of the most important and useful gate in reversible logic circuit synthesis. A complete reversible logic circuit synthesis algorithm based on Toffoli gate has been presented in [9]. Therefore, implementation of parity preserving Toffoli gate is essential. A parity preserving reversible TG circuit is presented in [10], which is shown in Figure 10. The circuit requires three reversible gates (one is FG and two are F2G gates) and produces two garbage outputs. The quantum cost of this implementation is 9. Another parity preserving reversible TG circuit is presented in [5], which is shown in Figure 11. The circuit requires two reversible gates (one is NFT and one is F2G gate) and produces one garbage output. The quantum realization of this implementation is not provided and is therefore unknown.

*Algorithm: Generate parity preserving k*k reversible gates*
Input: Gate width k.
Output: N_p: Number of parity preserving reversible gates.
* N_c: Number of conservative reversible gates.*
Initialization: I_v= {0, 1, 2, ..., 2^k-1}
* N_p=0; N_c=0;*

Run time complexity: $O(2^k! \times 2^k)$

1. *for (i = 0; i <(2^k)!; ++ i) {*
2. *O_v= Next_Permuation_of(I_v);*
3. *parity_flag=1; con_flag=1;*
4. *for(j = 0; j < 2^k; ++j) {*
5. *if (!(Number_of_1s(Binary_String[I_v[j]] %2 = Number_of_1s(Binary_String[O_v[j]]%2)){*
6. *parity_flag=0;*
7. *break; }*
8. *}*
9. *for(j = 0; j < 2^k; ++j) {*
10. *if (!(Number_of_1s(Binary_String[Iv[j]] = Number_of_1s(Binary_String[Iv[j]])){*
11. *con_flag=0;*
12. *break; }*
13. *}*
14. *if (parity_flag =1) { N_p++; print "Parity_Preserving: (I_v, O_v)"}*
15. *if (con_flag = 1) {N_c++; print "Conservative: (I_v, O_v)"}*
16. *}*

Table 5. Statistics of the parity preserving and conservativeness of different k*k reversible gates.

| Family of Reversible Gates	Total No of Gates	No of Parity Preserving Gates	No of Conservative Gates
2*2	24	4	2
3*3	40320	576	36
4*4	20922789888000	1625702400	414720
5*5	2.631308369336935301672180 1216e+35	4.377631366 9739505254 4e+26	1.8962192793 6e+17
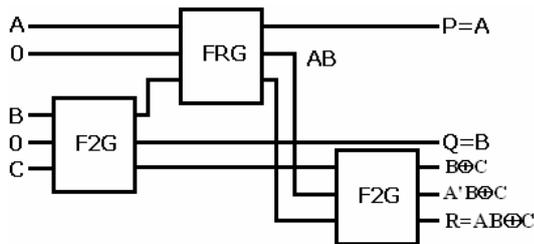
Figure 10. Existing TG with parity preservation, presented in [10].

In this study we have presented a new parity preserving reversible TG circuit, shown in Figure 12. The circuit requires only two reversible gates (one is FRG and one is F2G) and produces only one garbage output. Not only that, the quantum realization of the proposed design in NMR technology [13] is also given, as depicted in Figure 13. The quantum cost of the proposed design is 7 only.
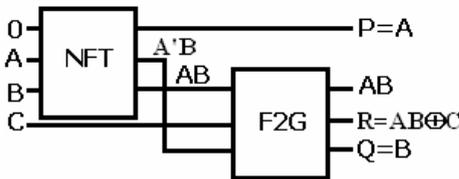
Figure 11. Existing TG with parity preservation, presented in [5].
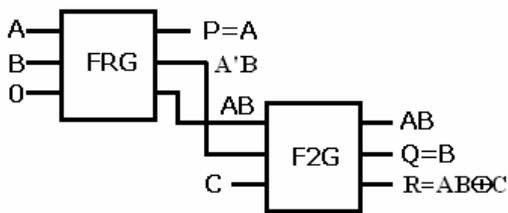
Figure 12. Proposed parity preserving TG implementation.
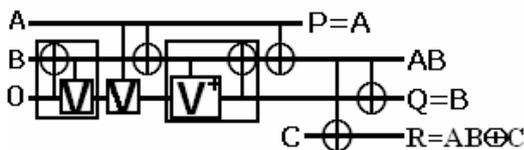
Figure 13. Quantum realization of the proposed parity preserving TG.

3. Synthesis of Fault Tolerant Reversible Full Adder Circuit

This section first establishes the minimum number of garbage outputs and constant inputs required to design a fault tolerant reversible full adder circuit; and then proposes a new realization of fault tolerant reversible full adder circuit using the newly proposed parity preserving 4*4 reversible gate IG and that is efficient than the existing designs.

Theorem 1: A full adder circuit can be realized with at least two garbage outputs and only one constant input [7].

Proof: The full adder circuit output equations $S=A\oplus B\oplus C_{in}$ and $C_{out}=(A\oplus B)Cin\oplus AB$ produce the same output S=1 and C_{out}=0, for the three distinct input combinations A=0, B=0, C_{in}=1; A=0, B=1, C_{in}=0; and A=1, B=0, C_{in}=0. Therefore, to separate all repeated values of outputs S and C_{out} we need at least two garbage outputs, as shown in Table 6. Thus the total number of outputs is 2+2=4. Now since in a reversible circuit the number of inputs must be equal to the number of outputs and there are three inputs in a full adder circuit A, B and C_{in}. Hence the fourth input needs to be constant input.

Table 6. Input combinations that produce the same output combinations in full adder circuit (shown shaded).

Input				Output			
A	B	Cin	C1	S	Cout	G1	G2
0	0	1	0	1	0	0	0
0	1	0	0	1	0	0	1
1	0	0	0	1	0	1	0

Theorem 2: A fault tolerant reversible full adder circuit can be realized with at least three garbage outputs and two constant inputs.

Proof: The full adder circuit output equations $S=A\oplus B\oplus C_{in}$ and $C_{out}=(A\oplus B)Cin\oplus AB$ produce the same output S=1 and C_{out}=0, for the three distinct input combinations A=0, B=0, C_{in}=1; A=0, B=1, C_{in}=0; and A=1, B=0, C_{in}=0. The parity of the input vector matches the parity of the corresponding output vector. To separate all repeated values of outputs S and C_{out} as well as keeping their parity unchanged we need at least three garbage outputs, as shown in Table 7. Thus the total number of outputs is 2+3=5.Now since in a reversible circuit the number of inputs must be equal to the number of outputs and there are three inputs in a full adder circuit A, B and C_{in}. Hence other two inputs need to be constant inputs.

Table 7. Input combinations that produce the same output combinations in full adder circuit (shown shaded).

Input					Output				
A	B	Cin	C₁	C₂	S	Cout	G₁	G₂	G₃
0	0	1	0	0	1	0	0	0	0
0	1	0	0	0	1	0	0	1	1
1	0	0	0	0	1	0	1	0	1

There are two fault tolerant reversible full adder circuits in the literature [2, 6]. The fault tolerant full adder circuit in [2] uses four FRGs and the fault tolerant full adder circuit in [6] requires six parity preserving reversible gates (two FRGs and four F2Gs). This paper presents a new design of fault tolerant reversible full adder circuit that uses only two IGs, depicted in Figure 14. It requires only two clock cycles.
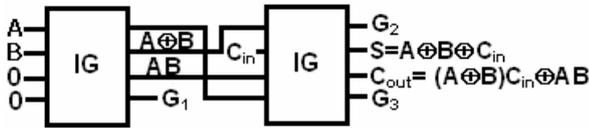

Figure 14. Proposed fault tolerant reversible full adder circuit.

4. Results and Discussion

The proposed parity preserving reversible TG and fault tolerant full adder circuits are more efficient than the existing circuits presented in [5, 10] and [2, 6] respectively. Evaluation of the proposed circuits can be comprehended easily with the help of the comparative results given in Tables 8 and 9.

4.1. Hardware Complexity

One of the main factors of a circuit is its hardware complexity. We can prove that our proposed circuit is better than the existing approaches in term of hardware complexity. Let α = A two input EXOR gate calculation, β = A two input AND gate calculation, δ = A NOT gate calculation, and T = Total logical calculation.

For [10] the Total logical calculation is: $T=6\alpha+4\beta+2\delta$, for [5] the Total logical calculation is: $5\alpha+3\beta+2\delta$, and for our proposed parity preserving reversible TG circuit, the Total logical calculation is: $4\alpha+4\beta+\delta$. Therefore, the hardware complexity of the proposed parity preserving reversible TG circuit is less than the existing counterparts. For [2] the Total logical calculation is: $T=8\alpha+16\beta+4\delta$, for [6] the Total logical calculation is: $12\alpha+8\beta+2\delta$, and for our proposed parity preserving reversible full adder circuit, the Total logical calculation is: $8\alpha+6\beta+2\delta$. Therefore, the hardware complexity of the proposed parity preserving reversible full adder circuit is less than the existing counterparts.

4.2. Garbage Outputs

Garbage output refers to the output of the reversible gate that is not used as a primary output or as input to other gates [7-9]. One of the other major constraints in designing a reversible logic circuit is to lessen number of garbage outputs. Our proposed parity preserving reversible TG circuit produces only one garbage output which is equal to the design in [5] and minimum, but the design in [10] produces two garbage outputs. So, we can state that our design approach is better than the

design given in [10] in term of number of garbage outputs. The proposed parity preserving reversible full adder circuit produces only three garbage outputs which are equal to the design in [6] and this is the theoretical minimum proved earlier in this paper, but the design in [2] produces six garbage outputs. So, we can state that our design approach is better than the design given in [2] in term of number of garbage outputs.

4.3. Constant Inputs

Number of constant inputs is one of the other main factors in designing a reversible logic circuit. The input that is added to an $n*k$ function to make it reversible is called constant input [7]. Our proposed parity preserving reversible TG circuit requires only one constant input which is equal to the design in [5], but the design in [10] requires 2 constant inputs. So, we can state that our design approach is better than the design given in [10] in term of number of constant inputs. The proposed parity preserving reversible full adder circuit requires only two constant inputs which are equal to the design in [2] and this is the minimum theoretically, but the design in [6] requires 5 constant inputs. So, we can state that our design approach is better than the design given in [6] in term of number of constant inputs.

4.4. Quantum Costs

The detailed cost of a reversible gate depends on any particular realization of quantum logic. Generally, the cost is calculated as a total sum of 2*2 quantum primitives used [13]. The cost of FG, depicted in Figure 1, is only 1. The cost of TG is exactly the same as the cost of FRG, and is 5. According to [13], the only cheapest quantum realization of a complete (universal) reversible gate is PG, depicted in Figure 2 and its cost is 4. But PG is neither parity preserving nor conservative. The quantum realization of NG and NFT is not available at this stage and is therefore unknown. The proposed parity preserving reversible TG circuit has been designed using one FRG gate and one F2G gate. The total quantum cost of our design is 7. The existing design in [10] requires one FRG gate and two F2G gates and quantum cost is 9. The design in [5] uses one NFT gate and one F2G gate. The quantum realization of NFT is not available and the realization cost of their design in any particular technology is unknown.

From the above discussion we can conclude that the proposed parity preserving reversible TG and full adder circuit is better than all the existing counterparts in term of hardware complexity. And one of the remarkable features of the proposed TG circuit is that its quantum realization is given in NMR technology [13].

Table 8. Comparative results of different parity preserving reversible toffoli gate circuits.

	No of Gates	No of Garbage Outputs	No of Constant Inputs	Total logical Calculation	Quantum Cost
This study	2	1	1	$4\alpha+4\beta+\delta$	5+2=7
Existing work[5]	2	1	1	$5\alpha+3\beta+2\delta$	Unknown
Existing work[10]	3	2	2	$6\alpha+4\beta+2\delta$	5+2+2=9

Table 9. Comparative results of different fault tolerant reversible full adder circuits.

	No of Gates	No of Clock Cycles	No of Garbage Outputs	No of Constant Inputs	Total logical Calculation
This study	2 IGs=2	2	3	2	$8\alpha+6\beta+2\delta$
Existing work[2]	4 FRGs=4	4	3	2	$8\alpha+16\beta+4\delta$
Existing work[6]	2FRgs+4 F2Gs=6	6	6	5	$12\alpha+8\beta+2\delta$

5. Conclusions

In this study, we have given an overview of the $k*k$ parity preserving reversible gates. We have found that a conservative reversible logic gate is parity preserving too. There are fewer conservative gates than the parity preserving reversible logic gates in a $k*k$ reversible gate families. An efficient parity preserving reversible TG circuit has been presented. It has been demonstrated that the proposed parity preserving reversible TG circuit is better than the existing counterparts in terms of hardware complexity, gate count, garbage outputs and constant inputs. Finally, this paper presents a novel realization of fault tolerant reversible full adder circuit and demonstrates its superiority than the existing designs.

References

[1] Bennet C., "Logical Reversibility of Computation," *IBM Journal of Research and Development*, vol. 17, no. 6, pp. 525-532, 1973.

[2] Bruce J., "Efficient Adder Circuits Based on a Conservative Reversible Logic Gates," *in Proceedings of IEEE Computer Society Annual Symposium on VLSI*, Pittsburg, pp. 83-88, 2002.

[3] Feynman R., "Quantum Mechanical Computers," *Optical News*, vol. 11, pp. 11-20, 1985.

[4] Fredkin E., "Conservative Logic," *International Journal of Theoretical Physics*, pp. 219-253, 1982.

[5] Haghparast M., "A Novel Fault Tolerant Reversible Gate for Nanotechnology Based Systems," *American Journal of Applied Sciences*, vol. 5, no. 5, pp. 519-523, 2008.

[6] Haghparast M., "Design of a Novel Fault Tolerant Reversible Full Adder for Nanotechnology Based Systems," *World Applied Sciences Journal*, vol. 3, no. 1, pp. 114-118, 2008.

[7] Islam M. and Md. Rafiqul Islam, "Minimization of Reversible Adder Circuits," *Asian Journal of Information Technology*, vol. 4, no. 12, pp. 1146-1151, 2005.

[8] Khan M., "Design of Full-Adder with Reversible Gates," *in Proceedings of 5th International Conference on Computer and Information Technology*, Dhaka, Bangladesh, pp. 515-519, 2002.

[9] Miller D., "A Transformation Based Algorithm for Reversible Logic Synthesis," *in Proceedings of Design Automation Conference*, USA, pp. 318-323, 2003.

[10] Parhami B., "Fault Tolerant Reversible Circuits," *in Proceedings of 40th of Asimolar Conference Signals, Systems, and Computers*, Pacific Grove, pp. 1726-1729, 2006.

[11] Peres A., "Reversible Logic and Quantum Computers," *Physical Review*, vol. 32, no. 6, pp. 3266-3276, 1985.

[12] Perkowski M., "A General Decomposition for Reversible Logic," *in Proceedings of RM 2001*, Starkville, pp. 119-138, 2001.

[13] Perkowski M., "A Hierarchical Approach to Computer-Aided Design of Quantum Circuits," *in Proceedings of 6th International Symposium on Representations and Methodology of Future Computing Technologies*, Germany, pp. 201-209, 2003.

[14] Toffoli T., "Reversible Computing," Automata*, Languages and Programming*, Springer-Verlag, pp. 632-644, 1980.

[15] Thapliyal H. and Srinivas M., "A New Reversible TSG Gate and Its Application For Designing Efficient Adder Circuits." *in Proceedings of 7th International Symposium on Representations and Methodology of Future Computing Technologies*, Tokyo, Japan, 2005.

[16] Haghparast M. and Navi K., "A Novel Revisable Full Adder Circuit for Nanotechnology Based Systems," *Journal of Applied Sciences*, vol. 7, no. 24, pp. 3995-4000, 2007.

Saiful Islam is a lecturer in the Institute of Information Technology at the University of Dhaka, Bangladesh. He holds MS degree in computer science and engineering from the University of Dhaka, 2007. His research interests include reversible logic design, multimedia information retrieval, machine learning, evolutionary computing, and pattern recognition.

Muhammad Mahbubur Rahman received his master degree in information technology from Institute of Information Technology at University of Dhaka. Currently, he is working as a lecturer in the Department of Computer Science at American International University, Bangladesh. His research interests include data mining, machine learning, bioinformatics, game theory, artificial intelligence, econometrics, and logic circuit minimization.

Zerina Begum obtained her BSc (Hons) and MSc in physics from the University of Dhaka in1986 and 1987, respectively. She was awarded M Phil for her research in semiconductor technology and computerized data acquisition system in 1995. She is still continuing to serve University of Dhaka as a professor in the Institute of Information Technology. Her current research interest areas include semiconductor technology, computerized data acquisition system, advanced digital electronics, software engineering, and computational biology.

Mohd Zulfiquar Hafiz obtained his BSc (Hons) and MSc in mathematics from the University of Dhaka in 1986 and 1987, respectively. He is working in University of Dhaka as an associate professor in the Institute of Information Technology. His current research interest areas include computer networks, embedded systems, and computational fluid dynamics.