Realization of a Novel Reversible SCG Gate and its Application for Designing Parallel Adder/Subtractor and Match Logic

Diganta Sengupta Mahamuda Sultana Atal Chaudhuri

Department of Computer Science and Engineering
Jadavpur University
Kolkata – 700032, West Bengal, India

ABSTRACT
In recent years, Quantum Electronics and Reversible Logic have emerged as a major area of research having applications in low power CMOS circuits, cryptography, optical computing and nanotechnology. The fact that classical logic gates such as AND, OR, XOR etc., barring the NOT gate, cannot predict the input given the output and hence generate heat due to information loss, has given rise to the concept of reversible logic. In this paper, a new reversible 4 * 4 "SCG" gate has been proposed which is being used to realize the classical set of logic gates in the reversible domain. The most promising fact of the proposed gate is that a single SCG gate can be used to realize a reversible Full Adder/Subtractor circuit or a single bit reversible Comparator. It has been shown that the Full Adder/Subtractor and the single bit Comparator using the proposed gate is much better and optimized in terms of number of garbage outputs and the number of reversible gates used in comparison to the existing counterparts in literature. Further efficient Reversible Parallel Adder/Subtractor circuits and Match Logic have been designed using the proposed SCG gate. Also a 4-bit digital comparator has been designed by cascading a series of single bit comparators using SCG gate.

General Terms
Architecture, Logic Design, Reversible Logic Gate.

Keywords
Reversible Logic, Reversible Gate, Reversible Full Adder/Subtractor, Reversible Comparator, Garbage Output, Constant Input, Match Logic.

1. INTRODUCTION
According to Landauer [1], a single bit of information loss amounts to KT ln(2) joules of energy dissipation, where 'K' is the Boltzmann's constant and 'T' is the absolute temperature at which the bit loss operation has been performed. Hence there is a direct relationship between the amounts of energy lost to the number of bits erased during computation involving classical logic gates. In conventional logic, a single switching of voltage-coded logic signals dissipates the stored energy in those signals which amounts to $E_{sig} = {}^{1}/_{2}\, CV^2$. This energy can be saved by use of reversible logic [2]. Bennett [3] showed that circuits built using reversible logic gates dissipate no energy as they have theoretically zero internal power. According to Gordon Moore, the performance of integrated circuit continues to improve at an exponential rate and doubles every 18 months, hence generating a lot of heat and reducing the life of the circuit. Therefore it is impossible to design Quantum circuits without reversible gates. According to Asher Peres [4], logical bits (0 and 1) can be represented by spin components, that is $|\uparrow>$ stands for logical 1 (true) and $|\downarrow>$ for logical 0 (false). Therefore a change in bit from 0 to 1 or vice versa, or a spin change can be accomplished by a magnetic field thereby dissipating no heat. Thus the problem remains in specifically remembering an input given a certain output. Reversible gates serve as the solution to the aforesaid problem.
Classical computers engage conventional logic gates for processing large amount of data. Apart from the NOT gate, no other boolean gate is reversible, i.e. in no other gate the input vector can be deterministically ascertained from the output vector of that gate. A Reversible Gate comes with the following requirements:

a. The input vector can be uniquely determined by the output vector

b. There is a one-to-one correspondence between the input and the output assignments

c. There should not be a fan-out of more than one.

d. Feedback is not allowed in reversible logic circuits.

Therefore, a $m \times n$ function F, where m = number of inputs, and n = number of outputs, is said to be reversible if and only if m = n and F has a one-to-one correspondence. In reversible logic gates, the output which is not used as a primary output or is not an input to another gate is known as a garbage output [5]. The inputs on the other hand having fixed values are known as constant inputs [6]. Hence for an optimized reversible circuit, the number of garbage outputs, the number of constant inputs and the number of reversible gates used should be a minimum. Reversible circuits find use in the domain of low power CMOS design [7], optical computing [8], quantum computing [9] and nanotechnology [10].

A $n \times n$ reversible gate is represented using two vectors:

$$I_V = (I_1, I_2, I_3 \ldots\ldots I_n) \text{ and } O_V = (O_1, O_2, O_3 \ldots\ldots O_n)$$

Where I_V and O_V are the input and output vectors respectively. Several reversible logic gates have been proposed in the last decade. Among them a 2 * 2 gate is the Feynman Gate [11], the 3 * 3 gates are the Toffoli Gate [12], Fredkin Gate [13], Peres Gate [4], New Gate [14], the 4 * 4 gates are the MKG Gate [15], HNG Gate [16], TSG Gate [17] and IG Gate [6].

2. PROPOSED 4 * 4 REVERSIBLE GATE

A 4 * 4 reversible gate has been proposed in this paper (See Figure 1). The Truth table for the corresponding gate is shown in Table 1. A closer look at the Truth Table reveals that the input pattern corresponding to a specific output pattern can be uniquely determined and thereby maintaining that there is a one-to-one correspondence between the input vector and the output vector. In this gate the input vector is given by $I_V = (D, C, B, A)$ and the corresponding output vector is $O_V = (P, Q, R, S)$. The inputs D,C,B and A are termed as the input terminals 1,2,3 and 4 respectively and the outputs P,Q,R and S are termed as the output terminals 1,2,3 and 4 respectively throughout the rest of the paper.

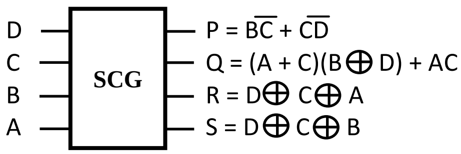

Fig 1: Proposed 4 * 4 Reversible SCG Gate

Table 1. Truth table of Reversible SCG Gate.

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

$$P = \sum m(2,3,4,5,6,7,10,11)$$

$$Q = \sum m(3,5,6,7,9,12,13,15)$$

$$R = \sum m(1,3,4,6,8,10,13,15)$$

$$S = \sum m(2,3,4,5,8,9,14,15)$$

3. REALIZATION OF THE CLASSICAL OPERATIONS

The proposed SCG gate can implement all the conventional boolean functions. Realization of AND operation, OR operation, XOR operation (See Figure 2), NAND operation and XNOR operation (See Figure 3), NOT operation, COPY operation (See Figure 4) are shown.

The fact that the proposed gate can implement NAND operation signifies that any boolean function can be implemented using the gate as NAND gate is a universal gate. Also since AND, OR and NOT operation can be implemented justifies the aforesaid because any boolean function can be materialized in product – of – sum or sum – of – products form. Also the COPY operation is an important operation which can be realized using the proposed reversible SCG gate.

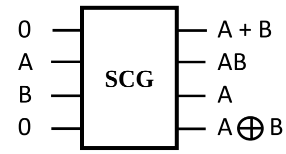

Fig 2: SCG gate implementing reversible AND, OR and XOR operation.

The configuration shown above can also be used in Carry Look Ahead Addition technique as $G_i = A_i B_i$ and $P_i = A_i + B_i$ [18].

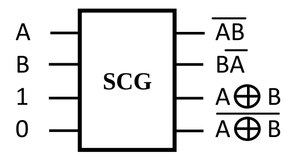

Fig 3: SCG Gate implementing reversible NAND and XNOR operation.

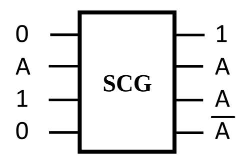

Fig 4: SCG Gate implementing reversible NOT and COPY operation.

In Figure 4, it can be seen that a signal 'A' is being produced at two output terminals, hence it can be said that the signal has been copied. Also the fourth output terminal produces the compliment of 'A' thereby justifying the NOT operation.

4. REALIZATION OF PARALLEL ADDER/SUBTRACTOR

The most promising fact of the proposed SCG gate is that this gate can singly be used as a Full Adder as well as a Full Subtractor circuit with minimum amount of garbage outputs and constant inputs. Figure 5 shows the implementation of the SCG gate as a Full Adder and Figure 6 shows the implementation as a Full Subtractor. In Table 2, a detailed description of the number of reversible gates required and garbage outputs is given together with a comparison with other Reversible Full Adder/Subtractor designs in literature. Also in Table 2, a comparison has been done in the Reversible Comparator domain.

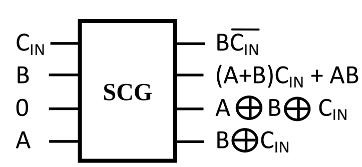

Fig 5: Full Adder using a single Reversible SCG Gate

The second output terminal in Figure 5 corresponds to the Carry-Out and the third terminal the Sum output of a Full Adder. In Figure 6, again the second terminal on the output side of the reversible gate corresponds to the Borrow-Out and the third terminal the Difference output of a Full Subtractor.

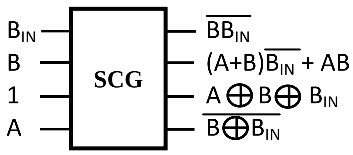

Fig 6: Full Subtractor using a single Reversible SCG Gate

Table 2. Comparative results on different Reversible Full Adder/Subtractor circuits

Name of the gate	Full Adder / Subtractor		Number of gates required for Single Bit Comparator	Universal Gate NAND/NOR realization
	No. of gates	No. of garbage outputs		
SCG	1	2	1	1 gate
TSG [17]	1	2	More than 1	1 gate
HNG [16]	1	2	More than 1	1 gate
IG [19], [22]	2	3	More than 1	1 gate
Fredkin Gate [23]	5	4	More than 1	More than 1 gate
Feynman gate and New gate [20]	3	3	More than 1	Not Applicable
Feynman gate and Toffoli gate [20]	4	2	More than 1	Not Applicable
Feynman gate, New gate and Toffoli gate[20], [21]	3	2	More than 1	Not Applicable

As a single SCG gate is enough for the implementation of a full Adder/Subtractor, hence an n – bit parallel Adder/Subtractor can be implemented using n SCG gates only. As has been shown that in the design of a full Adder/Subtractor itself the proposed gates gains above its existing counterparts in the count of gates and garbage outputs, hence the parallel Adder/Subtractor circuit designed by the proposed gate subsequently utilizes the minimum number of SCG gates and has the minimum number of garbage outputs. A circuit for a 4 bit parallel Adder/Subtractor has been shown having a control input (CT) which controls whether the circuits performs parallel addition or parallel subtraction.

Let the two 4 bit numbers to be parallel added or subtracted be $A = A_3A_2A_1A_0$ and $B = B_3B_2B_1B_0$.

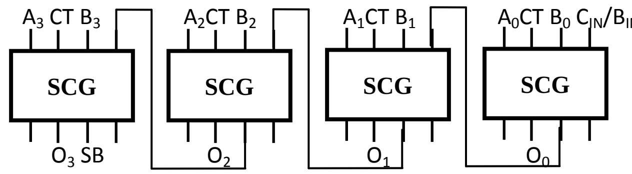

Fig 7: 4-Bit Parallel Adder/Subtractor using Array of Reversible SCG Gates.

In Figure 7, $O_0 - O_3$ are the four Sum or Difference outputs depending upon whether addition or subtraction is done. The CT control signal controls whether the circuit is operated in addition or subtraction mode. A 0 value at CT input makes the circuit a Parallel Adder (See Figure 8) and a 1 at the CT makes it a Parallel Subtractor (See Figure 9).

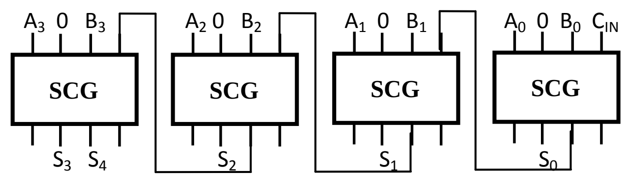

Fig 8: 4-Bit Reversible Parallel Adder making CT = 0.

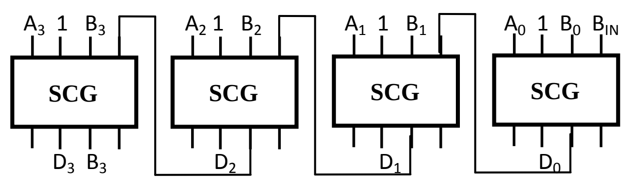

Fig 9: 4-Bit Reversible Parallel Subtractor making CT = 1.

The four outputs S_0, S_1, S_2 and S_3 in Figure 8 correspond to the four Sum Outputs having S_4 as the fifth output for the adder, and the four outputs D_0, D_1, D_2 and D_3 correspond to the four Difference outputs in Figure 9. The B_3 output in Figure 9 is the sign bit for the subtractor and the result is in 2's compliment format. The circuit shown in Figure 8 can also be said to be that of a Ripple Carry Adder.

5. REALIZATION OF DIGITAL COMPARATOR AND MATCH LOGIC

A single SCG gated has the potential to implement a single bit comparator as shown in Figure 10. If $B > A$, then the first output terminal gives a high logic, if $B < A$, then the second terminal gives a high logic. Equal bits are recognized by the last output terminal when it respectively goes high.

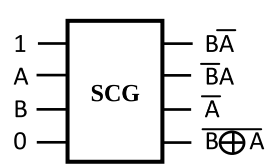

Fig 10: Reversible SCG implementing single bit Comparator

Hence intelligent cascading of such gates can generate efficient Match Logics as illustrated in Figure 11. Two four bit numbers have been checked for a bit-by-bit matching and consequently the logic gives a "True" (Logic-1) output for a match and "False" (Logic-0) output for at least one bit position change in the two four bit numbers.

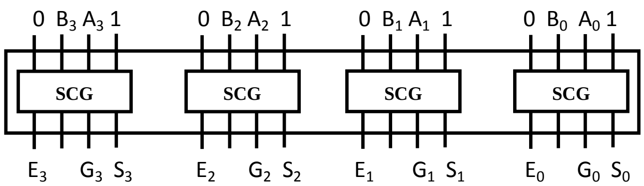

Fig 11: Cascaded Reversible SCG gates in 4-bit comparator configuration.

Let the two 4 – bit numbers to be compared for match be $A = A_3A_2A_1A_0$ and $B = B_3B_2B_1B_0$. Each pair of bits (A_iB_i) is fed to each comparator as shown in Figure 11. Consequently each comparator block compares the two input bits and generates the respective outputs. The naming of the outputs is as follows:

E_i shows True logic when $A_i = B_i$,

G_i shows True logic when $A_i > B_i$,

S_i shows True logic if $A_i < B_i$

For the two 4-bit numbers A and B to be equal, all the E_i outputs should be equal to Logic 1. Logically, it can be represented as:

$$\prod_{i=0}^{3} Ei = 1$$

Or $\quad E_3E_2E_1E_0 = 1 \quad (1)$

The implementation of such an expression is done by the aid of the Comparator Block of Figure 11 and three more SCG gates as shown in Figure 12. Two of the three additional SCG gates are being used as Full Adders and the third one implements the classical AND operation.

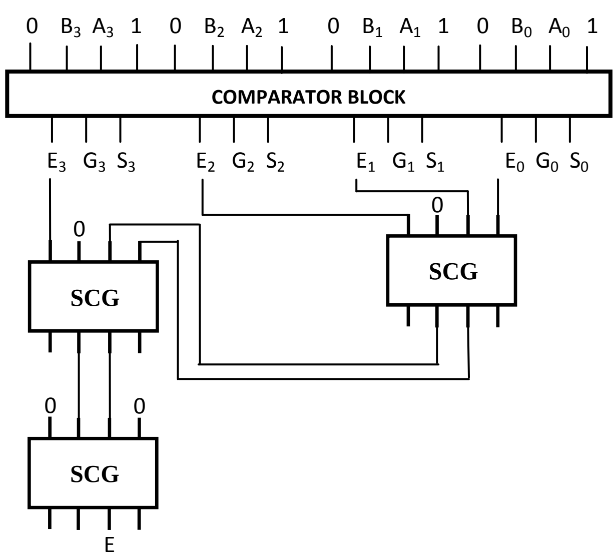

Fig 12: Match Logic using Reversible SCG Gates.

Figure 12 shows the complete Match Logic for matching two 4-bit numbers. The final "E" output displays a Logic 1 on a successful Match. The whole design produces a total of 11 garbage outputs out of which 4 are from the comparator block. Instead of employing three additional SCG gates after the comparator block, three Toffoli gates or three Peres gates can also be used for generating the "E" output. The Toffoli or Peres gates are to be used as AND gates where the first AND gate implements E_1E_0, the second implements E_3E_2, and the third does an AND operation on the outputs of the previous two AND gates.

Use of Toffoli or Peres gate instead of SCG gate as discussed before keeps the gate count same, only there is a mere reduction of 1 garbage output, i.e. the number of garbage outputs reduces from 11 to 10.

The comparator block can also be used efficiently to find the greater or the smaller of the two 4-bit numbers A and B. For efficient realization of the greater of the two numbers, the G_i outputs are employed using the following logical equation.

$$\begin{aligned} G &= G_3 + G_2E_3 + G_1E_3E_2 + G_0E_3E_2E_1 \\ &= G_3 + E_3[G_2 + E_2\{G_1 + G_0E_1\}] \end{aligned} \quad (2)$$

Close observation of (2) reveals that only six SCG gates are sufficient to implement the logic, three working as AND gates and the rest three as OR gates. It has already been seen in Figure 2 that an SCG gate is capable of implementing the AND operation as well as the OR operation. Figure 14 shows the implementation of (2) for finding out the greater of the two numbers A and B.

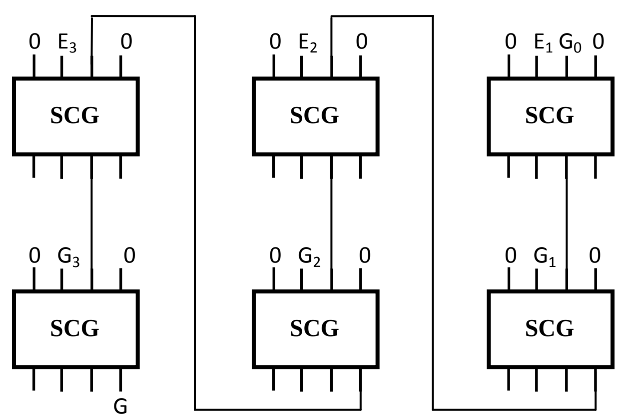

Fig 13: Logic diagram for finding the greater of two 4-bit numbers.

E_1, E_2, E_3, G_0, G_1, G_2, G_3 are the outputs of the comparator block in Figure 11. G is the final output in Figure 13 which when high denotes that the 4-bit number A is greater than B.

Since it can be detected whether two 4-bit numbers are equal or a number is greater than the other, it can be inferred that the architecture of whether a number is smaller than the other can also be designed simply by the logical expression:

$$S = (E + G)' = E'G'$$

Thus two outputs of Figure 12 and Figure 13 are fed into a set of two SCG gates as shown in Figure 14.

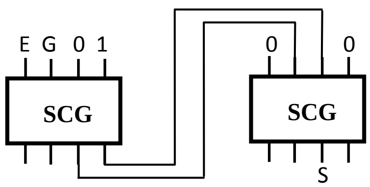

Fig 14: Logic diagram for finding the smaller of two 4-bit numbers.

In Figure 14, the final output is S which when goes high shows that the 4-bit number A is smaller than the 4-bit number B. Thus Figure 12, 13 and 14 taken together can be efficiently assembled to design a 4-bit digital comparator, the block representation of which is shown in Figure 15.

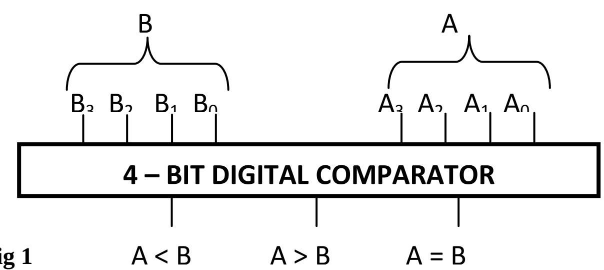

Fig 1

6. CONCLUSION

The proposed reversible SCG gate has been used to implement all the classical set of logical operations including three combinational circuits – Full Adder, Full Subtractor and Comparator using a single gate. Also it has been seen in section 4 that reversible Ripple Carry Adder and parallel Subtractor configurations can be optimally implemented using the said reversible gate. Hence novel designs of various reversible adders can be realized using the SCG gate in an efficient manner as it can implement all the boolean function operations including product – of – sum or sum – of – products forms for the same. Since reversible adders can be implemented, hence careful designs of reversible multipliers can also be achieved using the proposed reversible SCG gate. The Match Logic illustrated in section 5 can be used further for effective designing of the Match Logic in Associative Memory architectures, Code Matching and various other similar architectures.

7. ACKNOWLEDGMENTS

We would like to thank the Department of Computer Science and Engineering, Jadavpur University, Kolkata, India for providing us with all the possible assistance in pursuing the required research work.

8. REFERENCES

[1] Landauer, R., “Irreversibility and heat generation in the computing process”, *IBM J. Research and Development*, vol. 5 (3): pp. 183-191, 1961.

[2] M. P. Frank, “Introduction to reversible computing: motivation, progress and challenges”, *In Proceedings of the 2nd Conference on Computing Frontiers,* 2005, pp 385-390.

[3] Bennett, C.H., “Logical reversibility of computation”, *IBM J. Research and Development*, vol. 17: pp. 525-532, 1973

[4] Peres, A., “Reversible logic and quantum computers”, Physical Review: A, vol. 32 (6): pp. 3266-3276, 1985.

[5] H Thapliyal and M. Srinivas, “Novel reversible TSG gate and its application for designing carry look ahead adder and other adder architectures”, *Proceedings of the 10th Asia-Pacific Computer Systems Architecture Conference (ACSAC 05)*, 2005, pp 775-786.

[6] Saiful Islam M. and Md. Rafiqul Islam, “Minimization of reversible adder circuits”. *Asian J. Inform. Tech.,* vol. 4 (12): pp. 1146-1151, 2005.

[7] G Schrom, “Ultra Low Power CMOS Technology”, PhD Thesis, Technischen UniversitatWien, June 1998.

[8] E. Knil, R. Laflamme, and G.J Milburn, “ A Scheme for Efficient Quantum Computation With Linear Optics”, *Nature*, pp 46-52, Jan 2001.

[9] M. Nielsen and I. Chaung, “Quantum Computation and Quantum Information”, Cambridge University Press, 2000.

[10] R.C. Merkle, “Two Types of Mechanical Reversible Logic”, *Nanotechnology*, vol. 4:pp. 114-131, 1993.

[11] Feynman, R., “Quantum mechanical computers”, Optics News, vol. 11: pp. 11-20, 1985.

[12] Toffoli T., Reversible Computing, Tech Memo MIT/LCS/TM-151. MIT Lab for Computer Science, 1980

[13] Fredkin, E. and T. Toffoli, “Conservative Logic”. Int’l J. Theoretical Physics, vol. 21: pp. 219-253, 1982.

[14] Md. M. H. A. Khan, “Design of Full Adder with reversible gate”, *International Conference on Computer and Information Technology,* 2002, pp 515-519.

[15] Haghparast, M. and K. Navi, “A Novel Reversible Full Adder Circuit for Nanotechnology Based Systems”. *J. Applied Sci.,* vol. 7 (24) pp. 3995-4000, 2007

[16] Haghparast M. and K. Navi, “A Novel reversible BCD adder for nanotechnology based systems”. *Am. J. Applied Sci.*, vol. 5 (3), pp. 282-288, 2008.

[17] H. Thapliyal and M.B Srinivas, “A New Reversible TSG Gate and Its Application For Designing Efficient Adder Circuits”, Proceedings of the 7th International Symposium on Representations and Methodology of Future Computing Technologies (RM 2005), Tokyo, Japan, 2005.

[18] Lala P. K., Parkerson J. P., Chakraborty P., “Adder Designs using Reversible Logic Gates” *WSEAS Transactions. on Circuits and Systems*, 2010

[19] Islam, M.S. et a!.,"Low cost quantum realization of reversible multiplier circuit", *Information technology journal*,vol. 8, pp 208, 2009

[20] H. Md. H Babu, Md. R. Islam, S. M. A Chowdhury and A. R. Chowdhury, "Reversible Logic synthesis for Minimization of Full-Adder Circuit", *Proceedings of the EuroMicro Symposium on Digital System Design (DSD '03),* 2003, pp 50-54.

[21] H. Md. H Babu, Md. R. Islam, S. M. A Chowdhury and A. R. Chowdhury, "Synthesis of Full Adder Circuit using Reversible Logic", *Proceedings 17th International Conference on VLSI Design (VLSI Design 2004),* 2004, pp 757-760.

[22] S. Islam, Md. M Rahman, Z. Begum and Md. Z Hafiz, "Realization of a novel Fault Tolerant Reversible Full Adder Circuit in Nanotechnology", *The Int'l Arab J if Information Technology,* vol. 7 (3), pp. 317-322, 2010.

[23] 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.