International Journal of Engineering Inventions
ISSN: 2278-7461, ISBN: 2319-6491
Volume 1, Issue 11 (December2012) PP: 05-12

High Performance BCD Adder-Subtractor Using Reversible Logic

G. Sree Lakshmi[1], Dinesh Alapati[2], Dr. Kaleem Fatima[3,] Dr. B.K.Madhavi[4]
[1,2]Electronics & Communication Engineering, Malla Reddy Engineering College, JNTUH, Hyderabad, India
[3] Professor and HoD, Department of ECE, MJCET,Osmania University, Hyderabad
[4] Professor, Department of ECE ,Geetanjali College of Engineering& Tech., Hyderabad

Abstract:–Applications such as address generation, encoding, decoding, data shifting, etc are of primary importance in many computing and processing applications. Design of BCD adders and Subtractors therefore demands more attention and the advent of quantum computation and reversible logic, design and implementation of all sub-systems in reversible logic has received more attention. Moore's law in VLSI designs today is no more a simple reality, the device dimensions are shrinking exponentially and the circuit complexity is growing exponentially. Various low power design techniques are proposed and successfully achieved. Device scaling is limited by the power dissipation; and demands better power optimizations methods. Techniques like Energy recovery, Reversible Logic are becoming more and more prominent special optimization techniques in Low Power VLSI designs. Reversible logic opens tremendous avenues for power optimizations in the areas such as Quantum Computing, Nanotechnology, Sprintronics and Optical Computing. Reversibility plays an important role when energy efficient computations are to be designed. The objective of this work is to design a reversible BCD Adder/Subtractor that performs combined BCD addition and subtraction through a copying circuit and 2:1 vector MUX. The performance characteristics of the proposed design are shown in the form of Quantum cost, Garbage outputs. The performance characteristics analysis is carried out in Xilinx ISE design environment.

Keywords:–BCDAdder/Subtractor; 9's complementer; Copying circuit; 2:1vector MUX.

I. INTRODUCTION

1.1. NEED FOR LOW-POWER, AREA-EFFICIENT DESIGN:

The need for low power, area-efficient and design is motivated by several factors, such as the emergence of portable systems, thermal considerations, reliability issues, and, finally, environmental concerns. The evolution of portable or mobile communication devices such as laptops, cellular phones, video games, etc. is the most important factor driving the need for low power design. The main reason behind the development of low power circuits is that many portable devices and their applications require low power dissipation and high throughput. The commercial success of portable or mobile devices depends significantly on their weight, cost, and battery life. In most cases, the cost and weight of batteries become a bottleneck that prevents the reduction of system cost and weight. Moreover, for most portable systems, the IC components consume a significant portion of the total system power. Portable devices have a strict demand for power consumption since they have limited battery capacity. Low power design also plays a significant role in high-performance integrated circuits such as microprocessors and other high-speed digital computational circuits. Due to the increase in clock frequency, there is a proportional increase in power dissipation. The power consumed by the integrated circuit is dissipated in the form of heat. This may lead to problems such as circuit degradation and operating failures. The power consumption in microprocessors is projected to grow linearly in proportion to their die size and clock frequency. Various cooling systems have been introduced to reduce the heat from power dissipation and keep the chip temperature at an admissible level. This in turn has increased the packaging cost, which results in large revenue.

1.2. REVERSIBLE COMPUTING

In recent years, reversible computing system design is attracting a lot of attention. Reversible computing is based on two concepts: logic reversibility and physical reversibility. A computational operation is said to be logically reversible if the logical state of the computational device before the operation of the device can be determined by its state after the operation i.e., the input of the system can be retrieved from the output obtained from it. Irreversible erasure of a bit in a system leads to generation of energy in the form of heat. An operation is said to be physically reversible if it converts no energy to heat and produces no entropy. Landauer[1] has shown that for every bit of information lost in logic computations that are not reversible, kTlog2 joules of heat energy is generated, where k is Boltzmann's constant and T the absolute temperature at

which computation is performed. The amount of energy dissipation in a system increases in direct proportion to the number of bits that are erased during computation. Bennett showed that kTln2 energy dissipation would not occur, if a computation were carried out in a reversible way. Reversible computation in a system can be performed if the system is composed of reversible gates. The amount of energy dissipated in a system bears a direct relationship to the number of bits erased during computation and Reversible circuits do not lose information. In reversible logic there is one-to-one mapping between the input and output vectors and vice-versa. Reversible circuits are constructed using reversible logic gates. These reversible circuits not only produce unique output vector from each input vector but also the input can be reconstructed from the outputs. A reversible circuit should be designed using a minimum number of reversible gates. Fan-out and loops are not allowed in reversible logic circuits [3]. However fan-out and feedback can be achieved by using additional gates.

A reversible logic circuit should have the following features [5]:

- Use minimum number of reversible gates.
- Use minimum number of garbage outputs.
- Use minimum constant inputs.

The output that is not used for further computations is called garbage output [9]. The input that is added to an n x k function to make it reversible is called constant input [9]. Reversible logic has applications in several technologies such as nanotechnology, DNA computing, low power design, optical computing, Spintronics and quantum computing.

II. LITERATURE SURVEY

Physical limitations placed on computation by heat dissipation were studied for many years. The usual digital computer program frequently performs operations that seem to throw away information about the computer's history, leaving the machine in a state whose immediate predecessor is ambiguous [1]. Such operations include erasure or overwriting of data, and entry into a portion of the program addressed by several different transfer instructions. In other words, the typical computer is logically irreversible - its transition function (the partial function that maps each whole-machine state onto its successor, if the state has a successor) lacks a single-valued inverse.

Landauer [1] has raised the question of whether logical irreversibility is an unavoidable feature of useful computers, arguing that it is, and has demonstrated the physical and philosophical importance of this question by showing that whenever a physical computer throws away information about its previous state it must generate a corresponding amount of entropy. Therefore, a computer must dissipate at least kTln2 of energy (about 3 X 10-21 joule at room temperature) for each bit of information it erases or otherwise throws away.

At this point of time Bennett [2] showed: An irreversible computer can always be made reversible by having it save all the information it would otherwise throw away. For example, the machine might be given an extra tape (initially blank) on which it could record each operation as it was being performed, in sufficient detail that the preceding state would be uniquely determined by the present state and the last record on the tape. However, as Landauer pointed out, this would merely postpone the problem of throwing away unwanted information, since the tape would have to be erased before it could be reused. It is therefore reasonable to demand of a useful reversible computer that, if it halts, it should have erased all its intermediate results, leaving behind only the desired output and the originally furnished input. (The machine must be allowed to save its input-otherwise it could not be reversible and still carry out computations in which the input was not uniquely determined by the output.) General-purpose reversible computers (Turing machines) satisfying these requirements indeed exist, and they need not be much more complicated than the irreversible computers on which they are patterned. Computations on a reversible computer take about twice as many steps as on an ordinary one and may require a large amount of temporary storage.

While designing reversible system, the designer has to keep track of constraints [3]: only one fan-out is allowed and loops are not permitted. A circuit is said to be reversible if the input vector can be uniquely recovered from the output vector and there is a one to one correspondence between its input and output assignments. Thus, the number of inputs and outputs in reversible logic circuits are equal. Reversible circuits allow the reproduction of the inputs from the observed [2].

A reversible conventional BCD adder was proposed in [6] using conventional reversible gates – New Gates and Peres Gates. The implementation was improved in [7] using TSG reversible gates. An improved reversible implementation of decimal adder with reduced number of garbage is proposed in [8]. An optimized reversible BCD adder using new reversible gate (SCL Gate) was proposed in [9] which uses only 8 reversible gates, 6 constant inputs and produces only 8 garbage outputs. A reversible BCD Carry Look-Ahead Subtractor was proposed in [10]. The present work proposes a combined Reversible BCD Adder-Subtractor by using a reversible 9's complementer circuit and a reversible 2:1 vector MUX.

The organization of this paper is as follows: Initially, necessary background on reversible logic gates those are used for the design is given. Then the proposed blocks are implemented using reversible gates. Then the proposed BCD Adder-Subtractor is implemented using reversible gates. Finally a comparative analysis of blocks has been given in terms of number of gates, constant inputs, number of garbage outputs and delay in terms of number of gates.

III. REVERSIBLE LOGIC GATES

At present there are many 3x3 reversible logic gates such as Fredkin gate, Toffoli gate, Double Feynman gate, Peres gate [3-8]. The quantum cost of each reversible logic gate is an important optimization parameter [7]. The quantum cost of a 1x1 reversible gate is assumed to be zero. The quantum cost of a 2x2 reversible logic gate is taken as unity. The quantum cost of other reversible gates is calculated by counting the number of V, V+ and CNOT gates present in their quantum circuit. V is the square root of NOT gate and V+ is its hermitian.

The V and V+ quantum gates have the following properties:

V * V = NOT (1)

V * V+ = V+ * V = 1 (2)

V+ * V+ = NOT (3)

This section describes the reversible gates those are used for the implementation of the proposed 2:1 vector MUX and BCD Adder / Subtractor.

Feynman Gate

Fig 1 shows a Feynman Gate [11]. Feynman Gate (FG) can be used as a copying gate. Since a fan-out greater that one is not allowed, this gate is useful for duplication of the required outputs. If the input vector Iv = (A, 0), then the output vector becomes Ov = (P = A, Q = A).

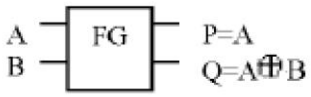

FIG. 1. 2 * 2 FEYNMAN GATE (FG)

Peres Gate

Fig 2 shows a Peres Gate (PG) [12]. It is also known as New Toffoli Gate (NTG). Functionally Peres Gate is equal with the transformation produced by a Toffoli Gate followed by a Feynman Gate.

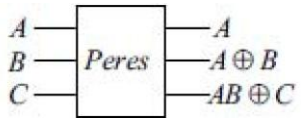

FIG. 2.3 * 3 PERES GATE (PG)

HNG Gate

Fig 3 shows a HNG Gate (HNG) [13]. The reversible HNG gate can work singly as a reversible full adder. If the input vector IV = (A, B, Cin, 0), then the output vector becomes OV = (P=A, Q=Cin, R=Sum, S=Cout). It produces only two garbage outputs and requires only one constant input. It needs only one clock cycle to perform the operation. It is better in terms of hardware complexity.

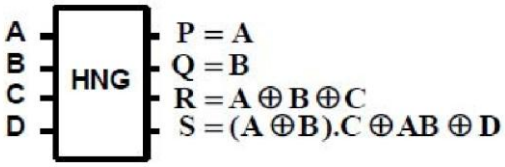

FIG. 3. 4 * 4 HNG GATE (HNG)

TKS Gate

Fig 4 shows a TKS Gate (TKS) [14].The TKS gate can be used to implement any Boolean function since two of its outputs (P & R) can function as 2:1 multiplexer.

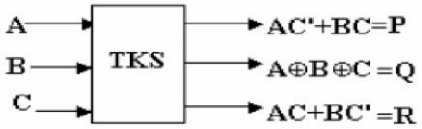

FIG. 4. 3 * 3 TKS GATE (TKS)

SCL Gate

Fig 5 shows a SCL Gate (SCL) [9]. The SCL gate (Six Correction Logic) can be used for the correction in the BCD addition.

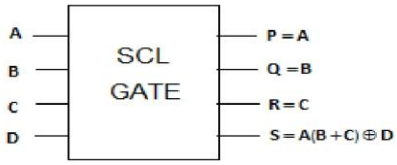

FIG. 5. 4 * 4 SCL GATE (SCL)

BVF Gate

Fig 6 shows a BVF Gate (BVF) [15].This is a reversible double XOR gate and can be used for the duplication of the required inputs to meet the fan-out requirements. This gate is used to copy the operand bits and the number of gates required to copy is reduced by 50% with same quantum cost. . If the input vector IV = (A, 0, C, 0), then the output vector becomes OV = (P=A, Q=A, R=C, S=C).

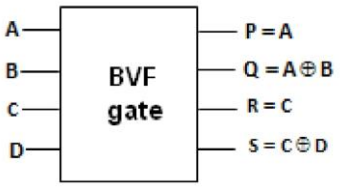

FIG. 6. 4 * 4 BVF GATE (BVF)

Nine's complementer – buffer

In the BCD subtraction, the nine's complement of the subtrahend is added to the minuend. In the BCD arithmetic, the nine's complement is computed by nine minus the number whose nine's complement is to be computed. This can be illustrated as the nine's complement of 5 will be 4 (9-5= 4), which can be represented in BCD code as 0100. In BCD subtraction using nine's complement, there can be two possible possibilities [8]:

The sum after the addition of minuend and the nine's complement of subtrahend is an invalid BCD Code (an example is when 5 is subtracted from 8) or a carryis produced from the MSB (an example is when 1 is subtracted from 9). In thiscase, add decimal 6 (binary 0110) and the end around carry (EAC) to the sum. The final result will be the positive number represented by the sum.

The sum of the minuend and the nine's complement of the subtrahend is a valid BCD code which means that the result is negative and is in the nine'scomplement form. An example is, when 8 is subtracted from 5 In BCD arithmetic, instead of subtracting the number from nine, the nine's complement of a number is determined by adding 1010 (Decimal 10) to the one's complement of the number. The nine's complementer circuit using a 4-bit adder and XOR gates is shown in Fig.7 [8]. We have realized that there is no need to use XOR gates in the nine's complementer for complementing. The use of NOT gates will better suit the purpose and will reduce the complexity of the circuit, both in CMOS as well as reversible logic implementation.

The proposed modified design of nine's complementer is shown in Fig.2; it replaces 4 XOR gates by 4 NOT gates and thus is better compared to the existing design in literature. The one-digit BCD Subtractor, using the nine's complementer circuit, is shown in Fig.3. In Fig.3, after getting the nine's complement of the subtrahend, it is added to the minuend using the BCD adder. Then the required 1010 is added by using the complement of the output carry of the BCD adder. The sign represents whether the number stored is positive or negative (for example, 5-8 will be stored as Sign=1 and Magnitude (S3...S0) = 3).

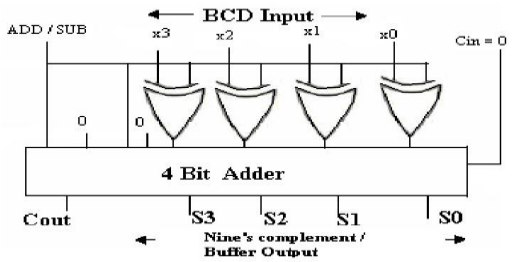

FIG. 7. CONVENTIONAL NINE'S COMPLEMENTER – BUFFER

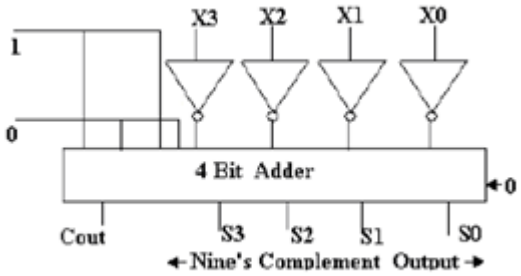

FIG. 8: OPTIMIZED NINE'S COMPLEMENTER

In the circuit, if ADD/SUB = 1, the input is complemented and added with 1010 to produce the nine's complement of the input. If ADD/SUB = 0 the circuit acts as a buffer. For the reversible implementation of the controlled NOT gate we have considered the complementer proposed in [10] which produces one garbage output. For reversible implementation of 4-bit parallel adder we have considered the circuit using HNG gates [17] since the hardware complexity is less. It has 4 constant inputs and produces 8 garbage outputs.

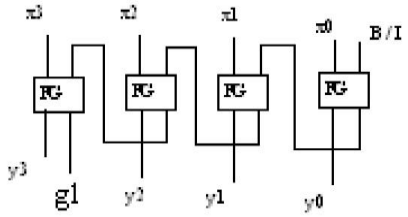

FIG. 9. REVERSIBLE CONTROLLED COMPLEMENTER-BUFFER

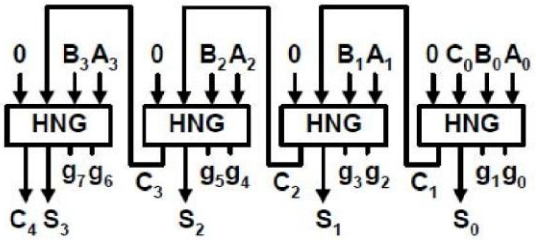

FIG. 10. REVERSIBLE 4-BIT PARALLEL ADDER

PROPOSED COPYING CIRCUIT

Since fan-out of the outputs is not allowed in the reversible circuits, we propose a reversible circuit to copy 5-bit data. The proposed copying circuit is shown in Fig 10. It uses 2 BVF gates and one FG gate. It has 5 constant inputs and has no garbage output.

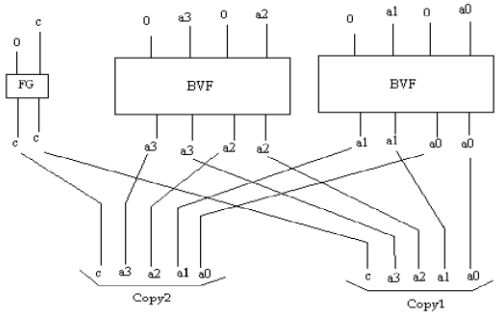

FIG. 11. PROPOSED COPYING CIRCUIT

PROPOSED 2:1 VECTOR MUX

In TKS gate, using C input as a select pin, we can produce multiplexer output in the outputs P and R. In this proposed vector MUX, P output is considered as the primary output and R is considered as garbage output. This proposed vector MUX uses no constant input and produces 10 garbage outputs.

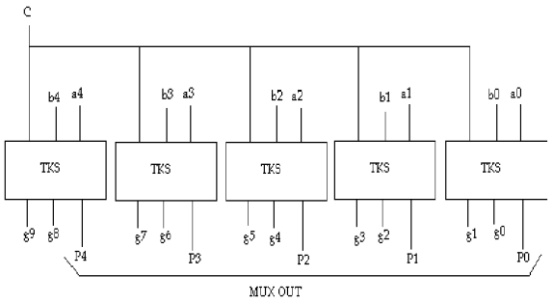

FIG.12: 2:1 VECTOR MUX

IV. PROPOSED BCD ADDER/SUBTRACTOR

Fig 13 shows the circuit of the proposed single digit BCD Adder-Subtractor. This circuit is a modified version of the one proposed in [10]. Instead of the Carry Look-Ahead logic, parallel logic is used in this proposed method. This single circuit is capable of functioning as BCD Adder or BCD Subtractor depending on the ADD/SUB control input.

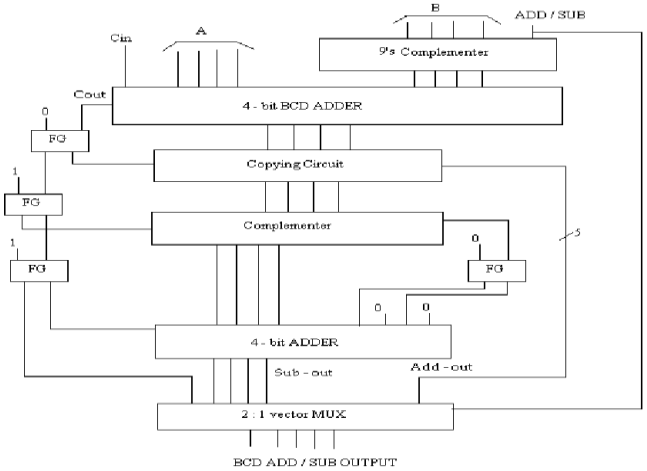

FIG. 13.PROPOSED BCD ADDER-SUBTRACTOR CIRCUIT

When the control input ADD/SUB = 1, the 9's complementer generates the nine's complement of the input digit B. This complemented data is added with the input digit A using 4-bit BCD adder [9] which uses 6 constant inputs and produces 10 garbage outputs. Two copies this output is generated using the copying circuit for the further process. One copy is given as the input to the complementer to produce the BCD subtracted output. This BCD Subtractor output is given as one of the input to the 2:1 5-bit vector MUX.When the control input ADD/SUB = 0, the 9's complementer acts as a buffer. This buffered input is added with input digit A using 4-bit BCD adder. This output is copied and given as one of the input to the 2:1 5-bit vector MUX.

The proposed 2:1 5-bit vector MUX is used to produce the final output based on the control input ADD/SUB. When the control input ADD/SUB = 0, the BCD Adder output is routed to the output of the circuit. When the control input ADD/SUB = 1, the BCD Subtractor output is routed to the output of the circuit.

V. RESULTS AND DISCUSSIONS

The proposed circuit uses a total number of 36 reversible gates consisting 14 Feynman gates, 13 HNG gates, one Peres gate, one SCL gate, two BVF gates and five TKS gates.

The total delay of the proposed design is calculated in terms of gate delays.

The total delay $\tau_{total} = \tau_{9'sC} + \tau_{bcd\text{-}ad} + \tau_{copy} + \tau_{comp} + \tau_{4\text{-}bit\ ad} + \tau_{vect\text{-}mux}$ + 4FG

where,

$\tau_{9'sC}$ = total delay in the 9's complementer-buffer circuit = 4 FG + 4 HNG

$\tau_{bcd\text{-}ad}$ = total delay in 4-bit BCD Adder

= 1 FG + 5 HNG +1 PG + 1 SCL

τ_{copy} = total delay in copying circuit= 1 FG + 2 BVF

τ_{comp} = 4 FG

$\tau_{4\text{-}bit\ ad}$ = total delay in the 4-bit adder = 4 HNG

$\tau_{vect\text{-}mux}$ = 5 TKS

Therefore τ_{total} = 36 gate delays

VI. CONCLUSION

This paper proposes a unified BCD Adder-Subtractor circuit along with its reversible logic implementation. The logical verification was done using Xilinx ISE. The simulation result is shown in Fig 14. The proposed system can be used for designing large reversible systems. The analyses of various blocks discussed in the design are tabulated in Table-1.

TABLE I: ANALYSES OF VARIOUS BLOCKS

Parameters /Various Blocks	No. of Gates	No. of garbage outputs	No. of constant inputs	Delay in terms of gates
9's Complementer	**8**	**9**	**4**	**8**
4 – bit BCD Adder	**8**	**10**	**6**	**8**
Copying Circuit	**3**	**0**	**5**	**3**
Complementer	**4**	**0**	**0**	**4**
4 – bit Adder	**4**	**8**	**4**	**4**
2:1 vector MUX	**5**	**10**	**0**	**5**
Total design	**32**	**37**	**19**	**32**

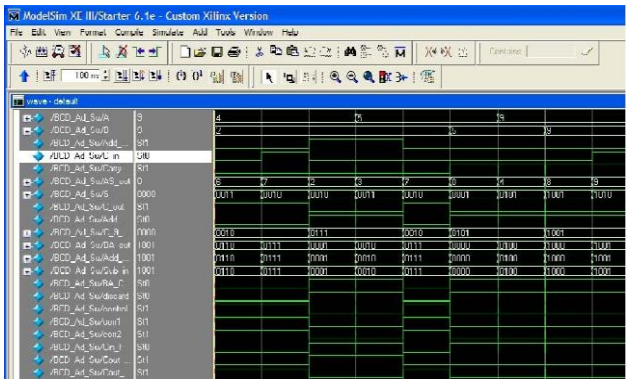

FIG. 14.SIMULATION RESULTS OF BCD ADDER-SUBTRACTOR

TABLE 2: COMPARISION OF VARIOUS DESIGNS

Design	Reversible gates	Garbage outputs	Garbage inputs	Delay in terms of gates
Design [1]	60	38	23	159
Design [2]	31	23	08	106
Design [3]	31	23	08	76
Proposed	32	37	19	32

REFERENCES

1) Landauer, R., "Irreversibility and heat generation in the computing process", IBM J. Research and Development, 5 (3): pp. 183-191, 1961.

2) Bennett, C.H., "Logical reversibility of Computation", IBM J. Research and Development, 17: pp. 525-532, 1973..

3) Kerntopf, P., M.A. Perkowski and M.H.A. Khan," On universality of general reversible multiple valued logic gates", IEEE Proceeding of the 34th international symposium on multiple valued logic (ISMVL'04), pp: 68-73, 2004..

4) Perkowski, M., A. Al-Rabadi, P. Kerntopf, A. Buller, M. Chrzanowska-Jeske, A. Mishchenko, M. Azad Khan, A. Coppola, S. Yanushkevich, V. Shmerko and L. Jozwiak, "A general decomposition for reversible logic", Proc. RM'2001, Starkville, pp: 119-138, 2001..

5) Perkowski, M. and P. Kerntopf, "Reversible Logic. Invited tutorial", Proc. EURO-MICRO, Sept 2001, Warsaw, Poland.

6) Hafiz Md. Hasan and A.R. Chowdhury, "Design of Reversible Binary Coded decimal Adder by using Reversible 4 – bit Parallel Adder", VLSI Design 2005, pp. 255 – 260, Kolkata, India, Jan 2005.

7) Himanshu Thapliyal. S. Kotiyal and M.B.Srinivas, "Novel BCD Adders and their Reversible Logic Implementation for IEEE 754r Format", VLSI Design 2006, Hyderabad, India, Jan 4 – 7, 2006, pp. 387 – 392.

8) R. James, T.K. Shahana, K.P. Jacob and S. Sasi, "Improved Reversible logic Implementation of Decimal Adder", IEEE 11th VDAT Symposium Aug 8 – 11, 2007.

9) H.R.Bhagyalakshmi, M.K.Venkatesha, "Optimized reversible BCD adder using new reversible logic gates", Journal of Computing, Volume2, Issue 2, pp. 28 – 32, February 2010.

10) Himanshu Thapliyal, Sumedha K. Gupta, "Design of Novel Reversible Carry Look-Ahead BCD Subtractor", 9th International Conference on Information Technology (ICIT'06), 2006.

11) R. Feynman, "Quantum Mechanical Computers", Optical News, 1985, pp. 11 – 20.

12) Peres, A., 1985. "Reversible logic and quantum computers" , Physical Review: A, 32 (6): 3266 - 3276.

13) M. Haghparast, S. J. Jassbi, K. Navi and O. Hashemipour, "Design of a Novel Reversible Multiplier Circuit using HNG Gate in Nanotechnology ", World Applied Sci. J., Vol. 3, pp. 974-978, 2008.

14) Himanshu Thapliyal, M.B Srinivas, "Novel Design and Reversible Logic Synthesis of Multiplexer Based Full Adder and Multipliers", pp. 1593 – 1596, 2005.

15) H.R.Bhagyalakshmi, M.K.Venkatesha, "An Improved Design of a Multiplier using Reversible Logic Gates", International Journal of Engineering Science and Technology, Volume2, Issue 8, pp. 3838 – 3845, February 2010.

16) R P Jain," Modern Digital Electronics", Third Edition, Tata McGraw Hill, pp.206-207.

17) Haghparast M. and K. Navi, 2008. "A Novel reversible BCD adder for nanotechnology based systems" American Journal of Applied Sciences, 5(3):282-288.