

# DESIGN OF REVERSIBLE LOGIC-BASED SINGLE PRECISION FLOATING POINT MULTIPLIER

Vidya Devi V.<sup>1</sup>, Ariya S.A.<sup>2</sup>, Renganayaki G.<sup>3</sup>

KCG College of Technology, Karapakkam.  
Email: <sup>2</sup>ariyasasidharan@gmail.com, <sup>1</sup>vidyapeace@gmail.com

## ABSTRACT

Reversible circuits avoids energy loss. Hence such circuits have significant potential in futuristic technologies such as ultra low power VLSI circuits, quantum computing, optical information processing, bioinformatics, and nanotechnology. In this paper, a reversible design of a single precision floating point multiplier is proposed.  $24 \times 24$  bit multiplier is required to multiply the significand of the floating point numbers. This multiplier is constructed using operand decomposition approach. Minimization of quantum cost and garbage output is considered as optimization criteria. Quantum cost of a reversible gate represents its computational complexity. Garbage outputs are unutilized outputs that are not used as primary outputs and which cannot be used as inputs for new computation. The reversible partial product generation circuitry, the reversible half adder and full adder and reversible 4:2 compressor for use in the compression tree, the reversible conditional right-shifter for normalization of the product are carefully designed and comparison with earlier designs shows the improvement in quantum cost and garbage output.

**Index Terms:** Floating point number, Reversible logic, Reversible multiplier, Nanotechnology, Operand decomposition.

## I. INTRODUCTION

When a computational system erases a bit of information, it must dissipate  $kT \ln 2$  energy, where  $k$  is Boltzmann's constant and  $T$  is the temperature. For  $T = 300$  Kelvin (room temperature), this is about  $2.9 \times 10^{-21}$  joules. As we employ increasingly advanced techniques to design our integrated circuits, the energy dissipation per logic operation has continually been falling. If we are to continue the revolution in computer hardware performance, we must compute with better energy-efficiency methods that can beat the  $kT$  barrier.

By building computers that perform reversible logic operations, arbitrarily low levels of heat dissipation can be achieved. Reversible computing performs computation in such a way that any previous state of the computation can always be reconstructed given a description of the current state. When we begin to consider computers constructed from very tiny and fast logic gates, as in nanocomputing, reversibility becomes an essential feature for keeping energy dissipation at tolerable levels.

Floating point arithmetic is useful in applications where a large dynamic range is required or in rapid prototyping applications where the required number range has not been thoroughly investigated. Floating point multiplication is one of the major operations in

image processing, digital filters, and digital signal processing applications. This paper proposes a reversible design [1] for multiplication of single precision floating point numbers.

The remainder of the paper is organized as follows: The various related works in reversible computing are summarized in Section 2; Section 3 provides an introduction to reversible logic; Section 4 provides an introduction to floating point representation system and floating point multiplication; Section 5 discusses the architecture and components of the floating point multiplier designed in this work; Section 6 presents the simulation results; Section 7 gives the conclusion and future work.

## II. RELATED WORK

Early computer researchers were interested in the physical limits of computing operations. This chapter considers research into the physical limits of energy dissipation during computation, how this research led to consideration of reversible computing systems.

Landauer, R. [2] showed that there is a minimum heat generation involved in computation, independent of the rate of the process. He defines a device to be logically irreversible, if the output of the device does not uniquely define the inputs. He labels a machine as being logically reversible, if and only if all its individual steps are logically reversible. He has demonstrated that

every bit of information loss generates  $kT \ln 2$  joules of heat energy.

Bennett, C.H. [3] has shown that, the usual general-purpose computing machines may be made logically reversible at every step, while retaining their simplicity and their ability to do general computations.

Toffoli, T. [4] presented a general conceptual model of computation, which bridges the gap between the irreversibility of the desired abstract computing task and the reversibility of a given underlying mechanism in an explicit way within the model itself.

The central result of the paper is that it is ideally possible to build sequential circuits with zero internal power dissipation. Even when these circuits are interfaced with conventional ones, power dissipation at the interface would be at most proportional to the number of input/output lines, rather than to the number logic gates as in conventional computers. This paper provides the foundations for the theory of reversible computing. It concludes that, the choice to use reversible mechanism in describing computing processes is a viable one.

The paper by Fredkin, E. and Toffoli, T. [5] deals with "conservative logic", a new mathematical model of computation which explicitly reflects in its axioms certain fundamental principles of physics, such as the reversibility of the dynamical laws and the conservation of certain additive quantities. The line of approach offered by conservative logic shows that it is ideally possible to build sequential circuits with zero internal power dissipation.

Feynman, R. [6] presented a Hamiltonian for a system of interacting parts, which will behave in the same way as a large system in serving as a universal computer. The sources of imperfections of the machine are also discussed. The Hamiltonian describes in detail all the internal computing actions, but not those interactions with the exterior involved in entering the input and reading the output.

Thapliyal, H. and Srinivas, M.B. [7] utilize a new  $4 \times 4$  reversible gate called TSG gate in this paper to build the components of a primitive reversible/quantum ALU. The most significant aspect of the TSG gate is that it can work singly as a reversible full adder, that is reversible full adder can now be implemented with a single gate only. A Novel reversible 4:2 compressor

is also designed from the TSG gate which is later used to design a novel  $8 \times 8$  reversible Wallace tree multiplier.

Thapliyal, H. and Ranganathan, N. [8] present novel designs of reversible sequential circuits that are optimized in terms of quantum cost, delay and the garbage outputs. The optimized designs of several reversible sequential circuits are presented.

### III. BACKGROUND

Reversible computation can be performed through circuits that do not lose information and are reversible in nature. Reversible circuits are designed using reversible gates which are logic gates that can generate unique output vector from each input vector, and vice versa.

Reversible logic has received significant and considerable interest in quantum computation because time evolution of a closed quantum mechanical system is inherently reversible. It has applications in low power CMOS design, optical information processing, DNA computing, bioinformatics, quantum computing and nanotechnology.

#### A. Reversible Gate

A Reversible gate is a  $k$ -input,  $k$ -output circuit ( $k \times k$ ) that produces a unique output pattern for each possible input pattern. There is a one to one correspondence between the vector of inputs and outputs.

Thus, an  $N \times N$  reversible gate can be represented as

$$I_v = (I_1, I_2, I_3, \dots, I_N) \quad (1)$$

$$O_v = (O_1, O_2, O_3, \dots, O_N) \quad (2)$$

Where  $I_v$  and  $O_v$  represent the input and output vectors respectively.

Classical logic gates are irreversible since input vector states cannot be uniquely reconstructed from the output vector states. There is a number of existing reversible gates such as Fredkin gate, Toffoli Gate, etc.

#### B. Quantum Cost

Each reversible gate has a cost associated with it called the quantum cost. The quantum cost of a reversible gate is the number of  $1 \times 1$  and  $2 \times 2$

reversible gates or quantum logic gates required in its design. The quantum costs of all reversible  $1 \times 1$  and  $2 \times 2$  gates are taken as unity.

### C. Garbage Outputs

Garbage outputs are the outputs of the reversible circuit that serve no useful function except to preserve its reversibility.

Given  $n$  Boolean inputs, any multiple output Boolean function on such  $n$  Boolean inputs must have exactly  $n$  Boolean outputs so that it is reversible. But most Boolean functions have lesser number of outputs than inputs. Hence, additional outputs are added to the circuit in order to get an  $n \times n$  reversible function.

### D. Feynman Gate

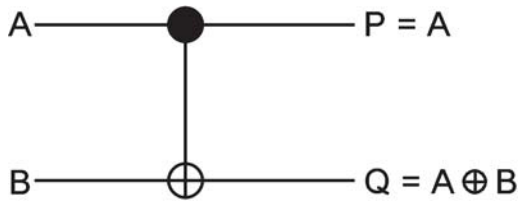


Fig. 1. Feynman Gate

The Feynman gate [6] is a 2-input 2-output reversible gate. It is also known as controlled-not gate (CNOT). It can be represented as:

$$Iv = (A, B) \quad (3)$$

$$Ov = (P = A, Q = A \oplus B) \quad (4)$$

Where  $A, B$  are the inputs and  $P, Q$  are the outputs, respectively. It has a quantum cost of 1. Fig. 1 shows the quantum representation of the gate. It can be used for copying the signal, generating the complement of a signal, etc.

### E. Toffoli Gate

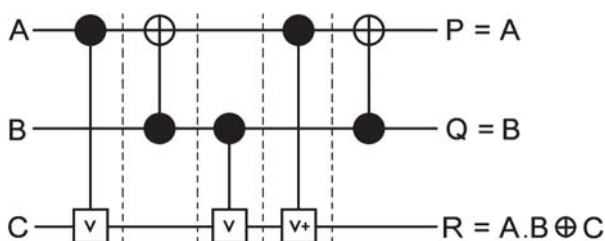


Fig. 2. Toffoli Gate

A Toffoli gate [4] is a  $3 \times 3$  two-through reversible gate as shown in Fig. 2. Two-through means two of its outputs are the same as inputs. It can be represented as:

$$Iv = (A, B, C) \quad (5)$$

$$Ov = (P = A, Q = B, R = A \cdot B \oplus B, R = A \cdot B \oplus C) \quad (6)$$

Where  $A, B, C$  are inputs and  $P, Q, R$  are outputs, respectively. Toffoli gate also known as controlled controlled-NOT (CCNOT). It has quantum cost of 5.

### F. Peres Gate

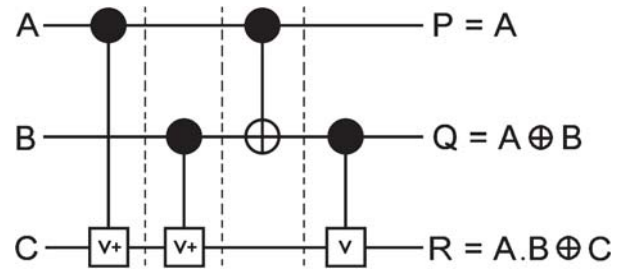


Fig. 3. Peres Gate

A Peres gate [9] is a 3 inputs-3 outputs ( $3 \times 3$ ) reversible gate. It is also known as New Toffoli gate (NTG), constructed by one Toffoli and one Feynman gate. It can be represented as:

$$Iv = (A, B, C) \quad (7)$$

$$Ov = (P = A, Q = A \oplus B, R = A \cdot B \oplus C) \quad (8)$$

Where  $A, B, C$  are inputs and  $P, Q, R$  are outputs, respectively. Fig. 3 shows the quantum implementation of the Peres gate with quantum cost of 4.

### G. Fredkin Gate

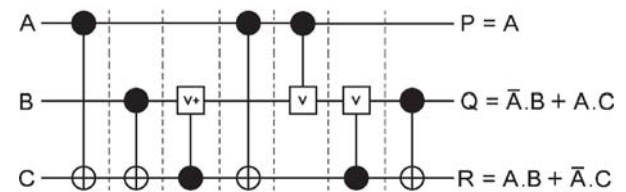


Fig. 4 Fredkin Gate

A Fredkin gate [5] is a  $(3 \times 3)$  conservative reversible gate, also known as controlled permutation gate. It can be represented as:

$$Iv = (A, B, C) \quad \dots(9)$$

$$Ov = (P = A, Q = \bar{A} \bullet B + A \bullet C, R = A \bullet B + \bar{A} \bullet C) \dots(10)$$

Where A, B, C are inputs and P, Q, R are outputs, respectively. Fig. 4 shows its quantum implementation with quantum cost of 4.

#### H. HNG Gate

A HNG gate [10, 11] is a 4 inputs- 4 outputs  $(4 \times 4)$  reversible gate. It can be represented as:

$$Iv = (A, B, C, D) \quad (11)$$

$$Qv = (P = A, Q = B, R = A \oplus B \oplus C, \quad (12)$$

$$S = (A \oplus B) \bullet C \oplus A \bullet B \dots D)$$

Where A, B, C are inputs and P, Q, R are outputs, respectively.

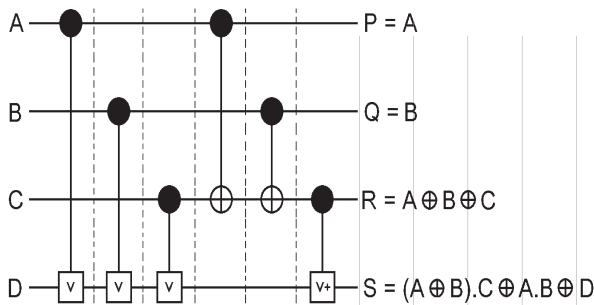


Fig. 5. HNG Gate

Fig. 5 shows the quantum implementation of the HNG gate. It has a quantum cost of 6.

## IV. FLOATING-POINT MULTIPLICATION

### A. Floating-Point Number

The floating-point representation system is a radix numeration system in which the location of the radix point is indicated by an exponent of the radix. The precision of the number is independent of its magnitude. Hence, there is no fixed number of digits before and after the radix point, i.e., the radix point "floats". Floating-point numbers whose radix equals 2 are called Binary Floating-point Numbers.

IEEE binary normal floating-point numbers are represented as a product of three components in the following form [12]:

$$(-1)^S \times (1 + 2^{1-p} \times T) \times 2^{E-bias}$$

Where,

- S is sign
- E is biased exponent
- T is trailing significand

### B. Single Precision Floating-Point Format

The IEEE floating-point standard defines a 32 bit floating-point format officially referred to as binary 32. It is commonly called Single precision Floating-Point Format.

Single precision format is characterized by precision of 24 bits and an exponent range from  $-126$  to  $+127$ . A bias of value 127 is used to obtain a biased exponent range from 1 to 255. Representation of the single precision format is shown in Fig. 6.

The three fields representing the number are:

- 1-Bit Sign, S
- 8-Bit Biased Exponent, E
- 23-Bits Trailing Significand, T

1	8	23
S	E	T

Fig. 6. Representation of number

The value of the single precision number is given by the product of the function of the three components as shown below:

$$(-1)^S \times (1 + 2^{-23} \times T) \times 2^{E-127}$$

### C. Multiplication

Multiplication of two floating point numbers involves pair-wise multiplication of the three factors S, E, T of the two numbers.

Given two normal single precision floating point numbers, X and Y, defined as:

$$X = (-1)^{S_x} \times (1 + 2^{-23} \times T_x) \times 2^{E_x-127}$$

$$Y = (-1)^{S_y} \times (1 + 2^{-23} \times T_y) \times 2^{E_y-127}$$

The product of the two floating point numbers is expressed as pair-wise multiplication of their respective components:

$$\{XY\} = \{(-1)_X^S 165 S_Y\} \times \{(1 + 2^{-23} \times T_S) (1 + 2^{-23} \times T_Y)\} \times \{2_X^E + E_Y^{-254}\}$$

## V. ARCHITECTURE

Multiplication of floating point numbers involves pair-wise multiplication of the sign, exponent, and significand of the multiplicand and multipliers. The computations involved are:

- Sign-bit computation
- Significand multiplication
- Exponent computation
- Normalization of product

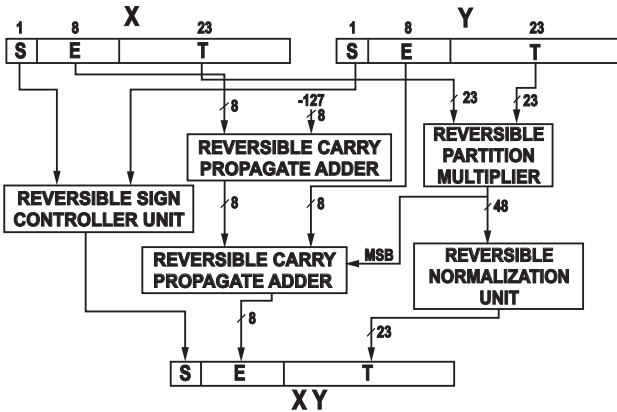


Fig. 7. Single Precision Floating Point Format

The overall structure of the reversible floating point multiplier is shown in Fig. 7. The major components of the design are:

- Reversible sign controller unit
- Reversible partition multiplier
- Reversible carry propagate adder
- Reversible normalization unit

The components are designed using reversible gates to minimize power consumption. Each part of the design is optimized in terms of quantum cost and garbage outputs.

### A. Reversible Sign Controller Unit



Fig. 8. Sign controller unit

An exclusive OR function is used to compute the sign bit of the product. The exclusive OR function is realized by a single Feynman gate. Hence the sign controller shown in Fig. 8 has a quantum cost 1 and the number of garbage output of the sign controller unit is 1.

### B. Reversible Partition Multiplier

The reversible partition multiplier is designed to perform the multiplication of 24-bit significands of the single precision floating point numbers [13]. It generates a 48-bit number which is normalized and rounded by the reversible normalization unit to obtain the trailing significand field of the product.

It utilizes a technique called operand decomposition to realize the  $24 \times 24$  bit reversible partition multiplier [14]. 24 bit numbers, A and B, to be multiplied are logically partitioned into 3 vectors of 8 bits each.

A is divided as:

- $AH = A_{23}-A_{16}$
- $AM = A_{15}-A_8$
- $AL = A_7-A_0$

B is divided as:

- $BH = B_{23}-B_{16}$
- $BM = B_{15}-B_8$
- $BL = B_7-B_0$

Thus the  $24 \times 24$  bit reversible multiplication is carried out using nine  $8 \times 8$  bit reversible. It generates nine 16-bit partial products. The  $8 \times 8$  bit reversible multipliers are designed using Wallace tree structure. A Wallace tree is an efficient implementation of a digital circuit that multiplies two integers, primarily used for fast multiplication. The 16-bit partial products are logically aligned as shown in Fig. 9 and compressed using reversible half adders and reversible full adders to generate the 48-bit product.



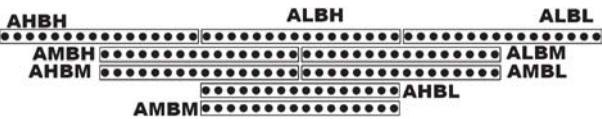


Fig. 9. Logical Alignment of 16-Bit Partial Products

C. Reversible  $8 \times 8$  Wallace Tree Multiplier

An  $8 \times 8$  bit reversible Wallace tree multiplier is used as a module in the design of reversible partition multiplier. Wallace tree multiplication consists of three conceptual stages:

- Partial product generation
- Partial product compression
- Final summation

The first step of generating all 64 one-bit partial products is performed in a reversible manner with a design that utilizes a cascade of Toffoli and Peres gates. The connection in series reduces garbage outputs. Fig. 10 illustrates the partial products generation for multiplication of two 8-bit numbers, X and Y.

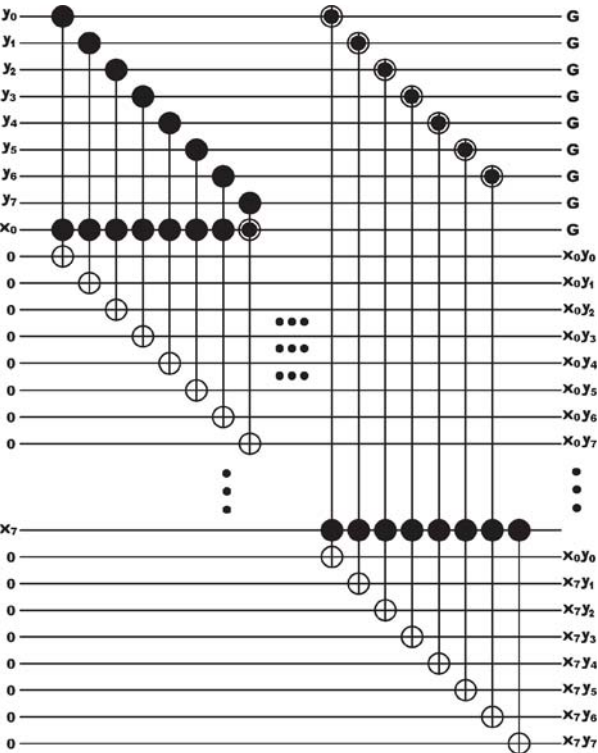


Fig. 10. Partial Product Generation

Partial product compression is done in two stages. The first stage shown in Fig. 11 uses four Reversible half adders (1, 9, 10 and 18), four Reversible full adders (2, 8, 11 and 17), and ten Reversible 4:2 compressors (3-7, 12-16).

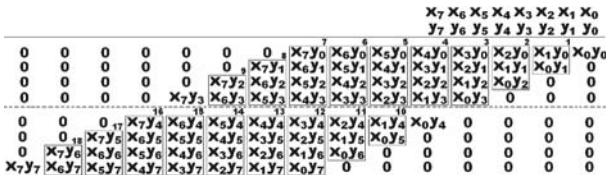


Fig. 11. Partial Product Compression - First Stage

The second stage uses six Reversible half adders (19, 20, 28-31),

Table 1  
Result of Reversible Partial Product Generation

	Quantum Cost	Garbage Outputs
Existing work [7]	320	64
Proposed design	305	16

Table 2  
Result of Reversible Full Adder

	Quantum Cost	Garbage Outputs
Existing work [7]	13	2
Existing work [15]	8	2
Proposed design	6	2

Table 3  
Result of Reversible 4:2 Compressor

	Quantum Cost	Garbage Outputs
Existing work [7]	26	4
Proposed design	12	4

**Table 4**  
**Result of Reversible  $8 \times 8$  Wallace Tree Multiplier**

	Quantum Cost	Garbage Outputs
Existing work [7]	113	186
Proposed design	635	106

**Table 5**  
**Result of Reversible Single Precision Floating Point Multiplier**

	Quantum Cost	Garbage Outputs
Sign Controller	1	1
$24 \times 24$	6429	1265
Carry Propagate Adder	48	16
Normalization Unit	283	72

Two Reversible full adders (21, 22), and five Reversible 4:2 compressors (23-27) as shown in Fig. 12.

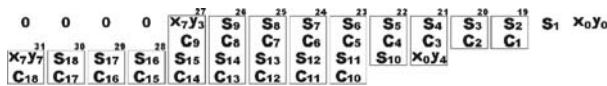


Fig. 12. Partial Product Compression - Second Stage

The Final summation is shown in Fig. 13. It consists of two half adders and eleven full adders.

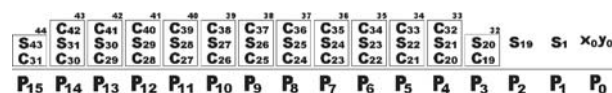


Fig. 13. Final Summation

#### D. Reversible Normalization Unit

The reversible partition multiplier generates a 48-bit number which is normalized and rounded by the reversible normalization unit to obtain the trailing significand field of the product. The Fredkin gate

provides the function of a reversible multiplexer in this circuit. This circuit also performs round-to-zero rounding of the significand by discarding the unwanted bits as garbage outputs.

## VI. RESULTS

Each unit of the proposed design is functionally verified. Also, the quantum cost, delay, and number of garbage outputs are calculated. Existing designs from earlier papers are also functionally verified, and quantum cost and number of garbage outputs calculated. Table 1 to Table 5 presents the results of reversible partial product generation unit, reversible full adder, reversible 4:2 compressor and reversible  $8 \times 8$  wallace tree multiplier. The results of the major blocks are presented in Table 5.

## VII. CONCLUSION

Reversible logic circuits seek to minimize information loss and hence offer a promising design approach for low-power computing. In this work, a reversible multiplier is designed to multiply single precision floating point numbers with a criteria to optimize Quantum cost and Garbage outputs.

The proposed design of Reversible sign controller unit is functionally verified and its quantum cost and garbage output found. A design is proposed for an  $8 \times 8$  Reversible Wallace tree multiplier that is used to construct the reversible partition multiplier. It is shown that the proposed design of reversible partial product generation unit, reversible full adder, reversible 4:2 compressor and reversible  $8 \times 8$  wallace tree multiplier are better than their existing designs in terms of quantum cost and garbage output.

Reversible Partition multiplier is designed using nine  $8 \times 8$  reversible Wallace tree multipliers. Reversible carry propagate adder is designed for computing Exponent of product. Reversible normalization unit is designed to normalize the product. The results show that the proposed design is optimized in terms of Quantum cost and garbage output.

## REFERENCES

- [1] M. Nachtigal, H. Thapliyal, N. Ranganathan, "Design of a Reversible Single Precision Floating Point Multiplier Based on Operand Decomposition," Proc. the 10th IEEE Intl. Conf. on nanotechnology Joint Symposium with Nano Korea 2010, Seoul, Korea, Aug 17-20, 2010, pp. 233-237.

- [2] R. Landauer, "Irreversibility and heat generation in the computational process," IBM J. Research and Development, 5, pp. 183-191, 1961.
- [3] H. Bennett, "Logical reversibility of computation," IBM J. Research and Development, pp. 525-532, November 1973.
- [4] T. Toffoli, "Reversible computing," Technical Memo MIT/ LCS/TM-151, MIT Laboratory for Computer Science, 1980.
- [5] E. Fredkin, T. Toffoli, "Conservative logic," Int. J. Theory. Phys, vol. 21, no. 3-4, pp. 219-253, 1982.
- [6] Feynman, R. "Quantum Mechanical Computers," Optics News, Vol.11, pp. 11-20, 1985.
- [7] H. Thapliyal and M.B. Srinivas, "Novel reversible 'TSG' gate and its application for designing components of primitive/reversible quantum ALU," Proc. the 5th IEEE Intl. Conf. on Information, Communications and Signal Processing, Bangkok, Thailand, Dec 6-9, 2005, pp. 1425-1429.
- [8] H. Thapliyal and N. Ranganathan, "Design of reversible sequential circuits optimizing quantum cost, delay and garbage outputs," To Appear ACM Journal of Emerging Technologies in Computing, 2010.
- [9] A. Peres, "Reversible logic and quantum computers," Physical Review A, vol.32, issue 6, pp. 3266-3276, Dec 1985.
- [10] M. Haghparast and K. Navi, "A Novel reversible BCD adder for nanotechnology based systems", American Journal of Applied Sciences, vol. 5, issue 3, pp. 282-288, 2008.
- [11] M. Mohammadi, M. Eshghi, M. Haghparast and A. Bahrololoom, "Design and Optimization of Reversible BCD Adder/Subtractor Circuit for Quantum and Nanotechnology Based Systems", World Applied Sciences Journal, vol. 4, no. 6, pp. 787-792, 2008.
- [12] "IEEE Standard for Floating-Point Arithmetic," IEEE Std 754-2008, pp.1-58, Aug. 29 2008 doi: 10.1109/IEEESTD.2008.4610935.
- [13] Shah, F.A.; Jamal, H.; Khan, M.A.; "Reconfigurable Low Power FIR Filter based on Partitioned Multipliers," Microelectronics, 2006. ICM '06. International Conference on, pp.87-90, 16-19 Dec. 2006.
- [14] V. Nahalingam and N. Ranganathan, "Improving accuracy in Mitchell's logarithmic multiplication using operand decomposition," IEEE Transactions on Computers, vol. 25, no. 12, pp. 1523-1534, December 2006.
- [15] M. Haghparast, M. Mohammadi, K. Navi and M. Eshghi, "Optimized reversible multiplier circuit," Journal of Circuits, Systems, and Computers, vol. 18, no. 2, 1-13, Feb. 2009.