iJET EXPLORE International Journal of Engineering & Technology
CENTRE OF PROFESSIONAL RESEARCH PUBLICATIONS
ISSN 2049-3444

International Journal of Engineering and Technology Volume 2 No. 1, January, 2012

Design of Optimal Reversible Carry Look-Ahead Adder with Optimal Garbage and Quantum Cost

Lafifa Jamal, Md.Shamsujjoha, Hafiz Md. Hasan Babu
Department of Computer Science and Engineering, University of Dhaka, Dhaka-1000, Bangladesh.

ABSTRACT

Conventional logic dissipates more power by losing bits of information whereas reversible circuitry recovers from bit loss through unique mapping between input and output vectors. In this regard, reversible or information lossless logic has become an immensely promising technology not only in low power CMOS design and nano-technology based system, but also primal requirement for quantum computing. On the other hand, carry look-ahead adder overcomes the limitations of the ripple wave adder by computing the carry values directly from the adder input. In this paper we present compact and efficient reversible logic implementations of carry look-ahead adder. The proposed design outperforms the existing works in terms of numbers of gates, garbage outputs, quantum costs and delay. In order to show the efficiency of the proposed method lower bounds of the combinational reversible carry skip logic in terms of garbage outputs and quantum cost are proposed as well, which is first ever proposed in the literature to the best of our knowledge. This design of proposed reversible circuit is appropriate for different quantum ALU and embedded processor.

Keywords: *Delay, low power design, quantum computing, reversible gate, carry look-ahead adder.*

1. INTRODUCTION

Maxwell and Szilard first ascertained the connection between a single bit and corresponding minimum quantity of entropy [1, 2]. Summary of their observation is, "a minimum amount of energy ($KT\ ln(m)$, where K is Boltzmann's constant of 1.38×10^{-23} and T is the absolute temperature of the environment and m is an integer number proportional to number of computed bits) must be dissipated during every elemental enactment of computation". In [3] Landauer proved that energy dissipation is only avoidable if the system is made reversible. Bennett showed that a reversible computation, in which no information is destroyed, dissipate an arbitrarily small amount of energy ($KT\ ln(1)$ which is logically zero) [4]. This observation entail's that, a computing system can operate having asymptotically zero energy loss if it is made reversible. They [3, 4] also showed that, irreversible system has a fundamental lower limit to the energy dissipated during a computation, proportional to the number of bit erased ($KT\ ln(2)$, for each erased bit). An irreversible system may store information which is produced during a computation rather than erasing it but doesn't provide a unique path from each state to its previous state. Energy used to store this information is unrecoverable unless the system is made reversible. Not only this, a reversible circuit can also be viewed as a special case of quantum circuit, as quantum evolution must be reversible [5]. Over the past few years reversible circuitry gained remarkable interests in the field of nano-technology and has also found applications in optical and DNA technology [6].

A carry look-ahead adder improves speed of addition through reducing the time required to determine carry bits. It actually calculates one or more carry bits before the summation, which reduces the wait time to calculate the result of the larger value bit. For this reason it also known as no wait carry adder. The Brent-Kung and Kogge-Stone adder are examples of carry look ahead adder. An irreversible concept, review of possible optimizations and applications of carry look-ahead adder can be found at [7~10]. There are some efforts to design reversible and quantum carry look-ahead adder [11~13]. In [11] Kai at el. showed a quantum representation of carry look-ahead adder. In [13] there is a parity preserving representation of reversible carry look-ahead adder but has huge quantum cost and garbage outputs compared to [11, 12].

2. BASIC DEFINITIONS

2.1 Reversible Gate

A Reversible gate is an n-input, n-output (denoted by $n\times n$) circuit that produces a unique output pattern [14, 15] for each possible input pattern. Reversible gates are circuits in which the number of outputs is equal to the number of inputs and there is a one to one correspondence between input vector I_v and output vector O_v denoted as $I_v \leftrightarrow O_v$, *i.e.*, it can generate unique output vector from each input vector and vice versa.

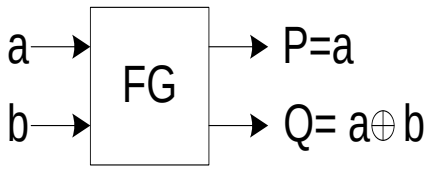

Figure 1. Block diagram of reversible Feynman gate

2.2 Quantum Computing and Reversibility

First proposed in 1970s, quantum computing relies on quantum physics, taking advantage of certain quantum physics properties of atoms or nuclei that allow them to work together as quantum bits, or qubits. Qubits can perform certain calculations exponentially faster than conventional bits [16]. Quantum computers encode information as a series of quantum-mechanical states such as spin directions of electrons or polarized orientations of a photon that might represent as ***|0>*** or ***|1>***, or might represent a superposition of the two values. Quantum computer demand underneath circuitry be reversible. This imposes extra costs when doing classical computations on a quantum computer, which can be necessary in subroutines of quantum computations [17]. Every reversible or quantum gate/circuit is built from 1×1 and 2×2 quantum primitives and its cost calculated as a total sum of 2×2 gates used since 1×1 gate cost nothing i.e. zero [18].

2.3 Garbage Output

Unwanted or unused output(s) of a reversible gate (or circuit) is known as **garbage output(s)**. More formally, the output(s) which is(are) needed only to maintain the reversibility are called garbage output(s). For example, to perform Exclusive-OR between two inputs, we can use 2×2 reversible Feynman gate [19] (**FG**) which produces an extra output along with desire output signal to preserve reversibility. This extra output is the garbage output, denoted by P in Fig. 1. Heavy price is paid off for each garbage output.

2.4 Restrictions on Reversible Logic Synthesis

In terms of state mapping reversible circuitry is the only solution which can regain every computed stage through backward order. But designing reversible circuit using reversible gate need to maintain some restrictions [18] such as, more than one fan-out is not allowed and loop or feedback is strictly restricted.

2.5 Delay of Reversible Circuit

The slowest logic path (path which consists maximum number of gate(s) for any input to any output) in the circuit is known as critical path. Delay of a reversible circuit is the delay of critical path. This definition is based on following two assumptions [18]:

- Every gate in the circuit will take same amount of time for internal logic operations.
- All inputs to the circuit are known before the computation begins i.e. internal structure and operations of each gate of the circuit are known in advance.

Another option to calculate the delay of the circuit is to assume some parameter for each logic operation (α for Ex-OR, β for OR, γ for AND etc) of the circuit and calculate the delay [11, 12]. In this case we actually calculate total number of logic operations needed to realize the circuit in terms of the assume parameters.

2.6 Reversible Feynman Gate

The input vector, $\boldsymbol{I_v}$ and output vector, $\boldsymbol{O_v}$ for 2×2 Feynman gate (***FG***) [19] is defined as follows: $I_v = (a, b)$ and $O_v = (a, a \oplus b)$. Block diagram of ***FG*** is shown in Fig.1. Normally ***FG*** is used as copying gate. For example, if input vector is $I_v = (a, 0)$ then output vector of ***FG*** will be $O_v = (a, a)$.

2.7 Reversible Toffoli Gate

Another commonly used reversible gate is Toffoli gate (***TG***). Input and output vectors for 3×3 ***TG*** are defined as follows [20]: $I_v = (a, b, c)$ and $O_v = (P = a, Q = b, R = ab \oplus c)$. Block diagram of reversible Toffoli gate is shown in Fig. 2. Using Toffoli gate one can implement logic operation such as, AND, NAND, NOT, Ex-OR etc.

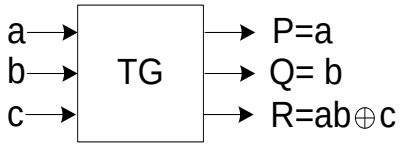

Figure 2. Block diagram of Toffoli gate

2.8 Reversible Peres Gate

Input and output vectors for reversible 3×3 Peres gate (***PG***) are defined as follows [21]: Input vector $I_v = (a, b, c)$ and output vector $O_v = (P = a, Q = a \oplus b, R = ab \oplus c)$. Block diagram of reversible Peres gate is shown in Fig. 3. Using Peres gate one can implement logic operation such as, AND, Ex-OR, NAND etc. In later section we will show how Peres gate can be used to develop carry generator and propagator for compact reversible carry look-ahead adder. There are some other common reversible gates such as Fredkin gate [20], TSG gate [18], NEW gate [24] etc. For proposed compact reversible carry look-ahead adder we use only reversible ***FG***, ***TG*** and ***PG***.

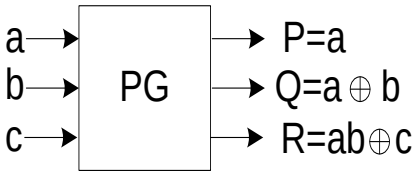

Figure 3. Block diagram of Peres gate

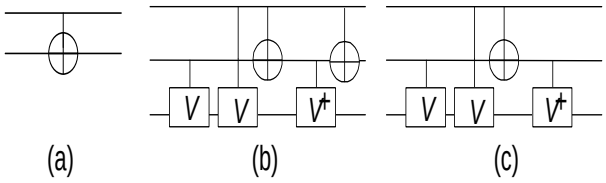

Figure 4. Quantum equivalent realization of reversible (a) Feynman gate (b) Toffoli gate (c) Peres gate

2.9 Quantum Equivalent Realization of Reversible Gates

Fig. 4(a), 4(b) and 4(c) show the quantum equivalent realization of reversible ***FG***, ***TG*** and ***FRG*** respectively. In this figure ***V*** is a square root-of-NOT (SRN) gate and V^+ is its hermitian. Thus ***VV*** creates a unitary matrix of NOT gate. And $VV^+ = I$, an identity matrix describing just a quantum wire. Quantum cost of each ***V*** and V^+ is 1 but ***VV*** and VV^+ has cost of zero [12~24]. Therefore, quantum cost ***FG***, ***TG*** and ***PG*** are 1, 5 and 4 as there are 1, 5 and 4, 2×2 quantum primitives respectively.

3. CARRY LOOK-AHEAD ADDER

Block diagram of a classical full adder (***FA***) is shown in Fig. 5, which is composed of two parts, a sum module and a carry module.

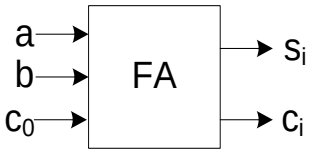

Figure 5. Block diagram of classical full adder

If input vector for classical full adder is, $I_v = (a,b,c_0)$ then, Sum (s_i) and Carry (c_{i+1}) can be computed from the Eq.1 and Eq.2, where c_0 is the initial carry whose value is zero [13].

$$s_i = a_i \oplus b_i \oplus c_i \qquad (1)$$

$$c_{i+1} = a_i b_i \oplus b_i c_i \oplus c_i a_i \qquad (2)$$

For the addition of two *n*-bit numbers ripple carry adder or carry look-ahead adder is a straight forward choice. In ripple carry adder (or serial carry and parallel output full adder) we need to compute the previous carry (c_{i-1}) in order to compute the next one (c_i). This delay disadvantage can be prevented if we forefend this dependencies among carry bits [6~10]. Carry look-ahead adder actually skips this dependency among carry bits by two modules, namely, carry generation (G_i) and carry propagation (P_i) where,

$$G_i = a_i b_i \qquad (3)$$

$$P_i = a_i \oplus b_i \qquad (4)$$

Therefore, we can rewrite the Eq. 1 and Eq. 2 as,

$$s_i = P_i \oplus c_i \qquad (5)$$

$$c_{i+1} = G_i \oplus P_i c_i \qquad (6)$$

Then the series of iterative operations to expand Eq.5 and Eq.6 for creating carry look-ahead adder can be carried out as follows:

$$c_1 = G_0 \oplus P_0 c_0$$

$$c_2 = G_1 \oplus P_1 c_1 = G_1 \oplus P_1 (G_0 \oplus P_0 c_0)$$

$$c_3 = G_2 \oplus P_2 c_2 = G_2 \oplus P_2 (G_1 \oplus P_1 c_1)$$
$$= G_2 \oplus P_2 (G_1 \oplus P_1 G_0 \oplus P_1 P_0 c_0)$$

$$Therefore\ ,\ c_n = G_{n-1} \oplus P_{n-1} (G_{n-2} \oplus \ldots \oplus P_{n-2} P_{n-3} \ldots P_0 c_0)$$

$$s_0 = P_0 \oplus c_0$$

$$s_1 = P_1 \oplus c_1 = P_1 \oplus G_0 \oplus P_0 c_0$$

$$s_2 = P_2 \oplus c_2 = P_2 \oplus G_1 \oplus P_1 c_1 = P_2 \oplus G_1 \oplus P_1 (G_0 \oplus P_0 c_0)$$

$$s_3 = P_3 \oplus c_3 = P_3 \oplus G_2 \oplus P_2 c_2$$
$$= P_3 \oplus G_2 \oplus P_2 (G_1 \oplus P_1 c_1)$$
$$= P_3 \oplus G_2 \oplus P_2 (G_1 \oplus P_1 G_0 \oplus P_1 P_0 c_0)$$

$$Therefore\ ,\ s_n = P_n \oplus G_{n-1} \oplus P_{n-1} (G_{n-2} \oplus \ldots \oplus P_{n-2} P_{n-3} \ldots P_0 c_0)$$

4. PROPOSED REVERSIBLE CARRY LOOK-AHEAD ADDER

4.1 Architecture of Proposed Method

From the iteration mentioned earlier, it is clear that the sum bit has a close relationship with the carry propagation function (P_i), carry generation function (G_i) and the initial carry (c_0). To generate $P_i = a_i \oplus b_i$ and $G_i = a_i b_i$ we use reversible Peres gate shown in Fig. 6.

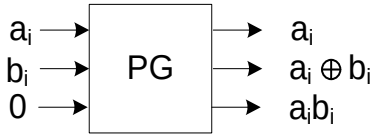

Figure 6. Reversible Peres gate as carry generator and propagator

From Fig. 6 we find that, A Peres gate with third input set to zero can be used for skipping the carry, thus the reversible full adder will not wait for the carry from the lower digit. Form Fig. 6 we also find that there are one garbage output and its quantum cost is 4 according to our previous discussion. Architecture of proposed 4-bit reversible carry look-ahead adder is shown in Fig.7. Algorithm for proposed reversible *n*-bits carry look-ahead adder is given in Algorithm 1. Idea behind the Algorithm 1 comes from the key property of carry look-ahead adder, and the key property is to remove the dependencies among carry bits with the help of P_i and G_i. To generate P_i and G_i we use reversible ***PG*** (Shown in Fig. 6).Here Feynman gate is used as a copying gate to generate sufficient number of P_i and G_i. Finally with all this P_i, G_i and initial carry we calculate s_i.

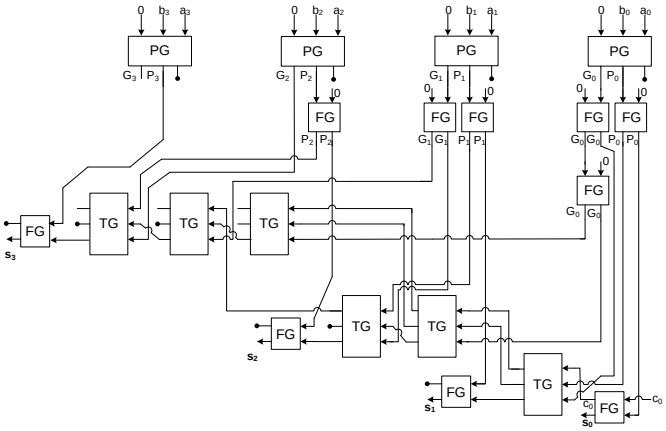

Figure 7. Architecture of proposed 4-bit reversible carry look-ahead adder

Algorithm 1: Algorithm fort *n*-bit reversible carry look-ahead adder.
Input: Initial carry $c_0 = 0$, two *n*-bit numbers $(a_{n-1}, \ldots, a_2, a_1, a_0)$ and $(b_{n-1}, \ldots, b_2, b_1, b_0)$
Output: An *n*-bit sum $(s_{n-1}, \ldots, s_2, s_1, s_0)$
1. **begin**
2. **for** $i \leftarrow 0$ **to** n-1 **do**
3. Generate P_i and G_i with help of ***PG***.
4. With the help of ***FG*** generate necessary amount of P_i and G_i to calculate s_i.
5. From generated P_i, G_i and with the supplied initial carry c_0 calculate s_i, for this purpose use reversible ***TG*** and ***FG***.
6. **end**

4.2 Evaluation of the Proposed Method

Theorem 1: A combinational circuit for reversible carry skip logic can be realized by at least one garbage output.

Proof: Carry generation block, carry propagation block and initial carry are needed in order to realize the reversible carry skip logic. Circuit shown in Fig. 8 can be used to realize combinational carry skip logic. It can be verified that the circuit of Fig. 8 is reversible by constructing its truth table, there is no need to give more detail. Fig. 8 is the extension of reversible Peres gate (***PG***) with an extra input and output. Therefore, it is named as ***EPG***. 2nd output of ***EPG*** will not be used but it is indispensable key for reversible logic, so 2nd output is the garbage output, *i.e.* we can realize circuit for carry skip logic with one garbage.

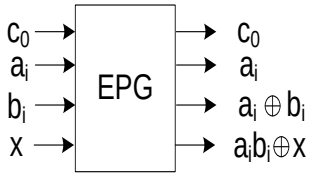

Figure 8. Circuits for realizing carry skip logic with one garbage output

Now, let assume combinational circuit which can realize reversible carry skip logic without garbage output does exists, then the structure of the combinational circuit will certainly be like the Fig. 9. Corresponding truth table for Fig. 9 is shown in Table I.

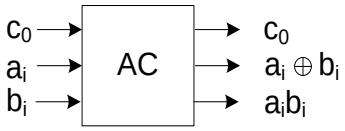

Figure 9. The assumptive circuit (*AC*)

Table 1: Truth table for fig. 9

c_0	a_i	b_i	c_0	P_i	G_i
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	1	0
0	1	1	0	0	1
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	1	1	0
1	1	1	1	0	1

According to the above truth table (Table 1) it is clear that input combination (0-0-1) and (0-1-0) will lead to the same output combination which will not satisfy the requirement of reversible logic synthesis. Therefore the combinational circuit which can realize the reversible carry skip logic without garbage output doesn't exist.

Theorem 2: A combinational circuit for reversible carry skip logic can be realized by at least 4 quantum cost.

Proof: A combinational carry skip logic is composed of initial carry bit (c_0), carry generation (G_i) and carry propagation (P_i) block where, $G_i = a_i b_i$ $P_i = a_i \oplus b_i$. A P_i can be realized by one 2×2 quantum primitive, which quantum cost is one. On the other hand G_i can't be realized by using two 2×2 quantum primitive gates (with maintaining the reversible property). However, a c_0, G_i and P_i can be realized by the combination of 4 2×2 quantum primitive gates as shown in Fig. 10. Therefore, a combinational circuit for reversible carry skip logic can be realized by at least 4 quantum cost.

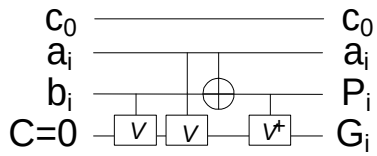

Figure 10. Quantum equivalent realizations of reversible combinational carry skip logic

Theorem 3: Let F be the required number of the gates for n-bit reversible carry look-ahead adder and ***PG***, ***FG*** and ***TG*** represent Press, Feynman and Toffoli gate respectively. Then,

$$F = n\,PG + \frac{n(n+1)}{2}FG + \frac{n(n-1)}{2}TG$$

Proof: For an n-bit carry look-ahead adder there are total of n carry propagator (P_i) and n carry generator (G_i). According to our design procedure, to generate nP_i and nG_i we need n Peres gate (***PG***). To generate sufficient amount of P_i and G_i, we need (n-i-1) Feynman gate (***FG***) for each i, where i = 0 to n-1. Finally to calculate each s_i except the 1st one, we need i ***TG*** and one ***FG***. For the 1st s_i (*i.e.* s_0) we need only one ***FG.*** Therefore, total number of gates for n-bit reversible carry look-ahead adder is,

$$\sum_{0}^{n-1} PG + ((n-1)+\ldots+2+1)\,FG + (1+2+\ldots+(n-1))\,TG + \sum_{0}^{n-1} FG$$

$$= n\,PG + \frac{n(n+1)}{2}FG + \frac{n(n-1)}{2}TG$$

Lemma 1: Let Q be the quantum cost for n-bit reversible carry look-ahead adder. Then,

$$Q = 3n^2 + 2n$$

Proof: In Theorem 3 we proved that, we need total of n ***PG*** $+ n(n+1)/2$ ***FG*** $+ n(n-1)/2$ ***TG*** gates for n-bit reversible carry look-ahead adder. According to our previous discussion we resolve that, quantum cost of each ***PG***, ***FG*** and ***TG*** are 4, 1 and 5 respectively. Therefore, total quantum cost for n-bit reversible carry look-ahead adder is,

$$4n + \frac{n(n+1)}{2} + 5\frac{n(n-1)}{2} = 3n^2 + 2n$$

4.3 Performance Analysis of the Proposed Method

Table 2 shows the performance of the proposed 4-bit reversible carry look-ahead adder. In Table 2 we consider the following notations for calculating the delay of the circuit,

α is the time for two-input Ex-OR calculation
β is the time for two-input AND calculation
γ is the time for a NOT calculation

From Table 2, we conclude that the proposed design based on reversible Peres, Toffoli and Feynman gates is much better than the existing ones, in terms of quantum cost, garbage outputs and delay (if we consider the unit delay for each gate, as we discuss in earlier section then the delay of the Fig. 7 is 9). We also want to mention that the proposed scheme also requires fewer number of gates than the all existing deigns [11~13]. In these comparisons we don't consider the work of [13] which requires more gates, garbage outputs, delay and quantum cost than the works of [11] and [12]. In the proposed design and in [11, 12] carry generator (G_i), carry propagator (P_i), where i= 0 to n and the initial carry (c_0) is not consider as garbage outputs.

Table 2: Performance Analysis of Proposed Design

	Kinds of gate	Quantum Cost	Garbage Output	Delay
Proposed Design	3	56	10	20α+10β
Existing Design[11]	5	99	19	53α +37β +61γ
Existing Design[12]	5	88	11	22α +24β

5. CONCLUSION

In this paper, we carried out reversible logic synthesis for reversible carry look-ahead adder. Several theoretical

explanations are given to prove the efficiency of the proposed scheme. Lower bounds on the garbage output and quantum cost for reversible combinational carry skip logic is proposed as well, which is first ever proposed in the literature to the best of our knowledge. The proposed circuit can be used in embedded processors such as, digital signal processors [22], high performance processors [23], high-speed and low-power applications [24], quantum ALU [25] etc.

REFERENCES

[1] Maxwell, J. C., 1902. Theory of heat, Longmans Green and co, 2nd ed.

[2] Szilard, L., 2003. On the decrease of entropy in a thermodynamic system by the intervention of intelligent beings maxwell's demon 2 entropy. Classical and quantum information computing, pp.110

[3] Landauer, R., 1961. Irreversibility and Heat Generation in the Computing Process. IBM Journal of Research and Development, vol. 5, pp. 183-191.

[4] Bennett, C. H., 1973. Logical Reversibility of computation IBM Journal of Research and Development, vol. 17, no. 6, pp. 525-532.

[5] Nielsen, M. A., Chuang, I., and Grover, L. K., 2002. Quantum computation and quantum information. American Journal of Physics, vol. 70, no. 558.

[6] Perkowski, M., 2003. A hierarchical approach to computer-aided design of quantum circuits. 6th International Symposium on Representations and Methodology of Future Computing Technology, pp. 201–209.

[7] Koren, I., 2001. Computer Arithmetic Algorithms. A K Peters, 2001.

[8] Parhami, B., 2000. Computer Arithmetic: Algorithms and Hardware Designs. Oxford University Press.

[9] Wallace, C. S., 1964. A suggestion for a fast multiplier. IEEE Trans. Computers, Vol. EC-13, pp. 14-17.

[10] Dadda, L., 1965. Some schemes for parallel multipliers. Alta Frequenza, Vol: 34, No. 5, pp. 349-356.

[11] Kai, C. W., and Tseng, C. C., 2002. Quantum plain and carry look-ahead adders. National Symposium on telecommunications, vol: 12.

[12] Thapliyal, H., and Srinivas, M., 2005. A novel reversible TSG gate and its application for designing reversible carry look-ahead and other adder architectures. Lecture Notes in Computer Science, vol:3740 pp. 805- 817.

[13] Islam, M. S., Rahman, M. M., Begum. Z., and Hafiz, M. Z., 2009. Fault tolerant reversible logic synthesis: Carry look-ahead and carry-skip adders. Advances in Computational Tools for Engineering Applications, pp. 396-401, July 2009.

[14] Hasan Babu. H. M., Islam. M.R., Chowdhury. A. R., and Chowdhury. S.M.A., 2003. Reversible logic synthesis for minimization of full-adder circuit, IEEE Conf. Digital Syst. pp.50–54.

[15] Hasan Babu. H. M., Islam. M.R., Chowdhury. A. R., and Chowdhury. S.M.A., 2004. Synthesis of full-adder circuit using reversible logic, 17th International Conference on VLSI Design, pp. 757–760.

[16] Altepeter. J. B., A tale of two qubits: how quantum computers work. Available at, http://arstechnica.com/science/guides/2010/01/a-tale-of-two-qubits-how-quantum-computers-work.ars; (last visited Aug 10, 2010).

[17] Shor. P. W., 1999. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. Society for Industrial and Applied Mathematics, SIAM, vol. 41, no. 2, pp. 303–332.

[18] Biswas. A. K., Chowdhury. A. R., and Hasan Babu. H. M., 2008. Efficient approaches for designing reversible binary coded decimal adders, Microelectronics Journal, vol. 39, pp. 1693–1703.

[19] Feynman. R., 1985. Quantum mechanical computers, Opt. News.

[20] Fredkin. E., and Toffoli. T., 1982. Conservative logic. Int'l J. Theoretical Physics,vol. 21, pp. 219–253.

[21] Peres. A., 1985. Reversible logic and quantum computers. Physical Review, vol. 32, no. 558.

[22] Voyiatzis. D.G., and Paschalis. A., 2005. Accumulator-based test generation for robust sequential fault testing in DSP cores in near-optimal time. IEEE Transactions on VLSI Systems, vol. 14, pp. 1079–1086.

[23] Metzgen. P., 2004. Optimizing a high performance 32- bit processor for programmable logic. Proceedings of International Symposium on System-on-Chip.

[24] Conway. T., 2004. Galois field arithmetic over gf(pm) for high-speed/low power error-control applications. IEEE Transactions on Circuits and Systems, vol. 51.

[25] Thapliyal. H., and Srinivas. M.,2006. Design of Wallace Tree Multiplier and Other Components of A Quantum ALU Using Reversible TSG Gate. SPIE Quantum Informatics, vol. 6264.

