A New Reversible Design of BCD Adder

Himanshu Thapliyal
Department of Computer Science and Engineering
University of South Florida,Tampa, FL, 33613
Email: hthapliy@cse.usf.edu

Nagarajan Ranganathan
Department of Computer Science and Engineering
University of South Florida,Tampa, FL, 33613
Email: ranganat@cse.usf.edu

***Abstract*—Reversible logic is one of the emerging technologies having promising applications in quantum computing. In this work, we present new design of the reversible BCD adder that has been primarily optimized for the number of ancilla input bits and the number of garbage outputs. The number of ancilla input bits and the garbage outputs is primarily considered as an optimization criteria as it is extremely difficult to realize a quantum computer with many qubits. As the optimization of ancilla input bits and the garbage outputs may degrade the design in terms of the quantum cost and the delay, thus the quantum cost and the delay parameters are also considered for optimization with primary focus towards the optimization of the number of ancilla input bits and the garbage outputs. Firstly, we propose a new design of the reversible ripple carry adder having the input carry C_0 and is designed with no ancilla input bits. The proposed reversible ripple carry adder design with no ancilla input bits has less quantum cost and the logic depth (delay) compared to its existing counterparts. The existing reversible Peres gate and a new reversible gate called the TR gate is efficiently utilized to improve the quantum cost and the delay of the reversible ripple carry adder. The improved quantum design of the TR gate is also illustrated. Finally, the reversible design of the BCD adder is presented which is based on a 4 bit reversible binary adder to add the BCD number, and finally the conversion of the binary result to the BCD format using a reversible binary to BCD converter.**

I. Introduction

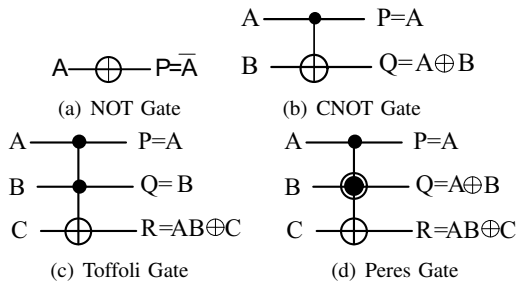
In the hardware design, binary computing is preferred over decimal computing because of ease in building hardware based on binary number system. In spite of ease in building binary hardwares, most of the fractional decimal numbers such as 0.110 cannot be exactly represented in binary, thus their approximate values are used for performing computations in binary hardware. Because the financial, commercial, and Internet-based applications cannot tolerate errors generated by conversion between decimal and binary formats, the decimal arithmetic is receiving significant attention and efforts are being accelerated to build dedicated hardware based on decimal arithmetic [1]. Among the emerging computing paradigms, reversible logic appears to be promising due to its wide applications in emerging technologies such as quantum computing, quantum dot cellular automata, optical computing, etc [2]. Reversible circuits can generate unique output vector from each input vector, and vice versa, that is, there is a one-to-one mapping between the input and the output vectors. Quantum computing needs to be build from reversible logic gates as quantum operations are reversible in nature [2]. Quantum computers of many qubits are extremely difficult to realize thus the number of qubits in the quantum circuits needs to be minimized [3], [4]. This sets the major objective of optimizing the number of ancilla input qubits and the number of the garbage outputs in the reversible logic based quantum circuits. The constant input in the reversible quantum circuit is called the ancilla input qubit (ancilla input bit), while the garbage output refers to the output which exists in the circuit just to maintain one-to-one mapping but does not work as a primary or a useful output.

The proposed work focuses on the design of the reversible BCD adder primarily optimized for number of ancilla input bits and the garbage outputs. As the optimization of ancilla input bits and the garbage outputs may degrade the design in terms of the quantum cost and the delay, thus quantum cost and the delay parameters are also considered for optimization with primary focus towards the optimization of number of ancilla input bits and the garbage outputs. To the best of our knowledge this is the first attempt in the literature that designs the reversible BCD adder with the goal of optimization of the number of ancilla input bits and the garbage outputs. To achieve the desired objective, firstly we present a new design of the reversible ripple carry adder with input carry (C_0) that have no ancilla input bits and is better than the existing counterparts in terms of quantum cost and the logic depth (propagation delay). The proposed reversible ripple carry adder is able to reduce the quantum cost and the delay of reversible ripple carry adder with no ancilla input bits by efficient use of the existing reversible Peres gate and a new reversible gate called the TR gate. The quantum design of the TR gate is also illustrated. We have illustrated the design of the reversible BCD adder which is based on a 4 bit reversible binary adder to add the BCD number and finally the conversion of the result of the addition to the BCD format using the reversible binary to BCD converter. The comparison of the proposed design with the existing designs is also illustrated.

II. Basic Reversible Gates

The reversible gates used in this work are the NOT gate, the CNOT gate, the Toffoli gate [5] and the Peres gate [6] which are shown in Fig. 1. Each reversible gate has the quantum cost and the delay associated with it [7]. As discussed in [7] the NOT gate and the CNOT gate have the quantum cost of 1 and delay of 1 Δ; the Toffoli gate has the quantum cost of 5 and delay of 5Δ; the Peres gate has the quantum cost of 4 and delay of 4Δ.

978-3-9810801-7-9/DATE11/©2011 EDAA



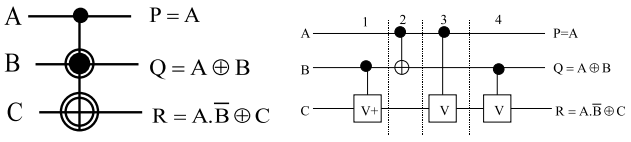
(a) NOT Gate (b) CNOT Gate

(c) Toffoli Gate (d) Peres Gate

Fig. 1. Existing Reversible Gates

A. Improved Design of the TR gate

The reversible TR gate is a 3 inputs 3 outputs gate having inputs to outputs mapping as (P=A, Q=A $\oplus$ B, $R = A \cdot \bar{B} \oplus C$ [8]. We present the graphical notation of the TR gate in Fig. 2(a) along with its new quantum implementation with 2x2 quantum gates in Fig. 2(b). The TR gate is designed from 1 Controlled V gate, 1 CNOT gate, and 2 Controlled V^+ gates resulting in its quantum cost as 4. Further, the logic depth of the quantum implementation of the TR gate is 4 resulting in its propagation delay as 4 Δ. The quantum cost and the delay of the TR gate were earlier estimated as 6 and 6 Δ, respectively [8]. The TR gate can realize the Boolean functions $A \cdot \bar{B} \oplus C$ and A $\oplus$ B with only gate. Further, it can implement the functions such as $A \cdot \bar{B}$ when its input C is tied to 0. These properties of TR gate make it very useful in designing the reversible arithmetic units.



(a) Quantum symbol of the TR Gate (b) Quantum realization

Fig. 2. TR gate and its improved quantum implementation

B. Prior Works

Arithmetic units such as adders, subtractors, multipliers form the essential component of a computing system. In [9] researchers have designed the quantum ripple carry adder having an input carry C_0 with no ancilla input bit, while the design with no input carry is designed with one ancilla input bit. In [4], [10], the researchers have investigated new designs of the quantum ripple carry adder having no input carry with no ancilla input bit and an improved delay. A comprehensive survey of quantum arithmetic circuits can be found in [3]. The researchers have also investigated the design of BCD adders and subtractors in which parameters such as the number of reversible gates, number of garbage outputs, quantum cost, number of transistors, etc are considered for optimization [11], [12], [13]. Thus to the best of our knowledge researchers have not yet addressed the design of the BCD arithmetic units primarily focusing on optimizing the number of ancilla input bits and the garbage outputs while also considering the parameters of quantum cost and the delay of the designs.

III. Design Methodology of Proposed Reversible ripple carry adder with input carry

Considering the addition of two numbers a_i and b_i stored at memory locations A_i and B_i, respectively, where 0≤i≤n-1. The input carry C_0 is stored at memory location A_{i-1}. Further, consider that z $\in \{0, 1\}$ is stored at memory location A_n. At the end of the computation, the memory location B_i will have s_i, while location A_i keep the value a_i. There is an additional location that initially stores the value z, and at the end of the computation will have the value $z \oplus s_n$. Thus at the end of the computation, location A_n will have the value of s_n when z=0. The proposed method improves the delay and the quantum cost by selectively using the Peres gate and the TR gate at the appropriate places. The proposed method is explained below with an example of addition of two 4 bit numbers a=$a_0...a_3$ and b=$b_0...b_3$:

1) For i=0 to n-1: At pair of locations A_i and B_i apply the CNOT gate such that the location A_i will maintain the same value of a_i, while location B_i transforms to the value $a_i \oplus b_i$. The step 1 is shown for addition of 4 bit numbers in Fig.3. At this step 1, the input state is transformed to:
$|c_0\rangle \left(\bigotimes_{i=0}^{n-1} |b_i \oplus a_i\rangle |a_i\rangle\right) |z\rangle$

2) For i= -1 to n-2: At pair of locations A_{i+1} and A_i apply the CNOT gate such that the location A_{i+1} will maintain the same value, while the location A_i transforms to $a_{i+1} \oplus a_i$. Further, apply a CNOT gate at pair of locations A_{n-1} and A_n such that the location A_{n-1} will maintain the same value, while the location A_n transforms to $a_{n-1} \oplus a_n$. The step 2 is illustrated in Fig. 3. At this step, the state is transformed to:

$$|c_0 \oplus a_0\rangle \left(\bigotimes_{i=0}^{n-2} |b_i \oplus a_i\rangle |a_i \oplus a_{i+1}\rangle\right) |b_{n-1} \oplus a_{n-1}\rangle |a_{n-1}\rangle |z \oplus a_{n-1}\rangle$$

3) For i=0 to n-2: At locations A_{i-1}, B_i and A_i apply the Toffoli gate such that A_{i-1}, B_i and A_i are passed to the inputs A, B, C, respectively, of the Toffoli gate. The step 3 is illustrated in Fig.3. Apply a Peres gate at location A_{n-2}, B_{n-1} and A_n such that A_{n-2}, B_{n-1} and A_n are passed to the inputs A, B, C, respectively, of the Peres gate. After this step, the state is transformed to:

$$|c_0 \oplus a_0\rangle \left(\bigotimes_{i=0}^{n-2} |b_i \oplus a_i\rangle |c_{i+1} \oplus a_{i+1}\rangle\right) |b_{n-1} \oplus c_{n-1}\rangle |a_{n-1}\rangle |z \oplus s_n\rangle$$

Further, for i=0 to n-2: Apply a NOT gate at location B_i as illustrated in Fig.3. This step will transform the state to:

$$|c_0 \oplus a_0\rangle \left(\bigotimes_{i=0}^{n-2} \overline{|b_i \oplus a_i\rangle} \, |c_{i+1} \oplus a_{i+1}\rangle \right) |b_{n-1} \oplus c_{n-1}\rangle \, |a_{n-1}\rangle \, |z \oplus s_n\rangle$$

4) For i=n-2 to 0: At locations A_{i-1}, B_i and A_i apply the TR gate such that A_{i-1}, B_i and A_i are passed to the inputs A, B, C, respectively, of the TR gate. The step is illustrated in Fig.3. Further, for i=0 to n-2: Apply a NOT gate at location B_i as illustrated in Fig.3. After this step, the state is transformed to:

$$|c_0 \oplus a_0\rangle \left(\bigotimes_{i=0}^{n-2} |b_i \oplus c_i\rangle \, |a_i \oplus a_{i+1}\rangle \right) |b_{n-1} \oplus c_{n-1}\rangle \, |a_{n-1}\rangle \, |z \oplus s_n\rangle$$

5) For i=n-1 to 0: At pair of locations A_i and A_{i-1} apply the CNOT gate such that the location A_i will maintain the same value of a_i, while location A_{i-1} transforms to the value $a_i \oplus a_{i-1}$. This step is shown in Fig.3, and after this step the state is transformed to:

$$|c_0\rangle \left(\bigotimes_{i=0}^{n-2} |b_i \oplus c_i\rangle \, |a_i\rangle \right) |b_{n-1} \oplus c_{n-1}\rangle \, |a_{n-1}\rangle \, |z \oplus s_n\rangle$$

6) For i=0 to n-1: At pair of locations A_i and B_i apply the CNOT gate such that the location A_i will maintain the same value of a_i, while location B_i transforms to the value $a_i \oplus b_i$. This step is shown for addition of 4 bit numbers in Fig.3 and after this step the state is transformed to:

$$|c_0\rangle \left(\bigotimes_{i=0}^{n-1} |s_i\rangle \, |a_i\rangle \right) |z \oplus s_n\rangle$$

Thus, the proposed methodology is able to design the reversible ripple carry adder with an input carry without any ancilla input bit.

Delay and Quantum Cost

- The step 1 of the proposed methodology needs n CNOT gates working in parallel thus this step has the quantum cost of n and delay of 1 Δ.
- The step 2 of the proposed methodology needs n+1 CNOT gates working in series thus this step has the quantum cost of n+1. The delay of this stage will be only 2 Δ as here n-1 CNOT gates work in parallel with the Toffoli gates of the next stage thus only 2 CNOT gates contributes to the delay..
- The step 3 needs n-1 Toffoli gates working in series thus contributing to the quantum cost of 5(n-1) and delay of 5(n-1) Δ. There is a Peres gate contributing to quantum cost of 4 and delay of 4 Δ. Further, there are n-1 NOT gates working in parallel with the Peres gates thus contributing to quantum cost of n-1 and zero delay. The total quantum cost of this stage is 5(n-1)+4+n-1 while the delay contribution of this stage is 5(n-1) Δ +4 Δ.
- The step 4 needs n-1 TR gates working in series thus contributing to the quantum cost by 4(n-1) and delay of 4(n-1) Δ. Further, there are n-1 NOT gates which all work in parallel with the TR gates except the last NOT gate. Thus, it contributes to quantum cost of n-1 and delay of 1 Δ. Thus this step has the quantum cost of 4(n-1)+n-1 and the delay of 4(n-1) Δ +1 Δ.
- The step 5 needs n CNOT gates working in parallel with the TR gates and the NOT gates, except the last one. Thus this step has the quantum cost of n and delay of 1 Δ.
- The step 6 needs n CNOT gates working in parallel thus this step has the quantum cost of n and delay of 1 Δ.

Thus the total quantum cost of n bit reversible ripple carry adder is n+n+1+5(n-1)+4+n-1+4(n-1)+n-1+n+n=15n-6. The propagation delay will be 1 Δ+2 Δ+5(n-1) Δ+4 Δ+4(n-1) Δ+1 Δ+1 Δ+1 Δ=(9n+1) Δ. A comparison of the proposed design with the existing designs is illustrated in Table I which shows that the proposed design of reversible ripple carry adder with input carry is designed with no ancilla input bit and has the less quantum cost and delay compared to its existing counterparts. In Table I AIs, GOs, QC stand for ancilla inputs, garbage outputs and quantum cost, respectively. The comparison is not illustrated with [4], [10] as they have designed the reversible ripple carry adder with no input carry.

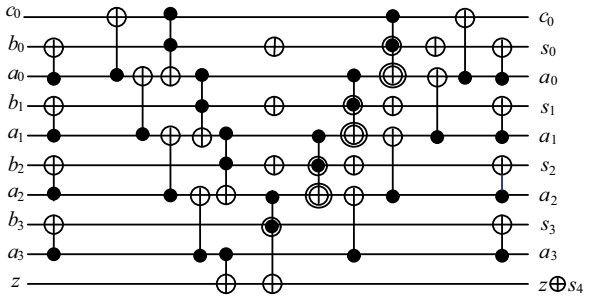


Fig. 3. Proposed reversible 4 bit adder with input carry

TABLE I
A COMPARISON OF RIPPLE CARRY ADDER WITH INPUT CARRY

	AIs	GOs	QC	Delay Δ
[9]	0	n+1	17n-6	10n+2
[9]	0	n+1	17n-22	10n-8
Proposed	0	n	15n-6	9n+5

IV. DESIGN OF REVERSIBLE BCD ADDER

In the BCD addition, the two decimal digits A and B, together with the input carry Cin, are first added in the top 4-bit binary adder to produce the 4 bit binary sum (K_3 to K_0) and the carry out (Cout). The outputs of the binary adder can be passed to a binary to BCD converter to have the result of the binary addition in the BCD format. This approach is illustrated in the Fig. 4(a) where a 5 bit binary to BCD

converter produces the desired output in the BCD format. In this work, we have proposed the equivalent reversible design of the approach shown in Fig.4(a) to design the reversible BCD adder optimized for the number of ancilla input bits and the number of garbage outputs. The 4 bit reversible ripple carry adder with input carry can be designed based on the methodology illustrated for ripple carry adder with input carry as shown in Fig.3. For binary to BCD conversion, we have used the reversible binary to BCD converter proposed in [13] that has the quantum cost of 16 and delay of 16 Δ. The proposed design of the reversible BCD adder is shown in Fig.4(b) in which the input carry Cin is represented by c_0, Cout represents the carry out of the 4 bit reversible binary adder, and OC represents the output carry of the 1 digit BCD adder.

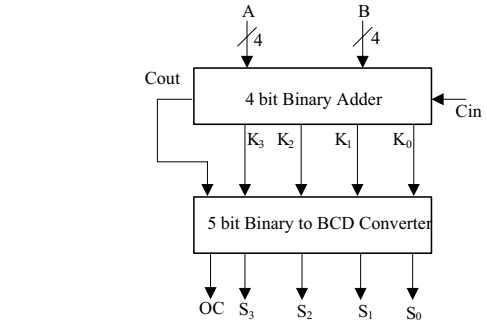
The proposed reversible BCD adder design has 1 ancilla input bit and 5 garbage outputs. The quantum cost of the proposed reversible BCD adder with input carry is 70 while the delay is 57 Δ. A comparison with the existing designs of the reversible BCD adder is illustrated in Table II in which AIs, GOs, QC stand for ancilla inputs, garbage outputs and quantum cost, respectively. The design 3 in [13] is the best existing design in literature considering the number of ancilla input bits and the garbage outputs. It needs 2 ancilla inputs bits, 6 garbage outputs and has the quantum cost of 103. The delay of [13] is not known. It can be observed from the comparison table that the proposed design is better than the design 3 in [13] in terms of number of ancilla input bits and the number of garbage outputs. It also improves the quantum cost compared to design 3 in [13] by 47%. Thus the proposed design in this work is efficient compared to existing designs in literature.

TABLE II
A COMPARISON OF REVERSIBLE BCD ADDERS

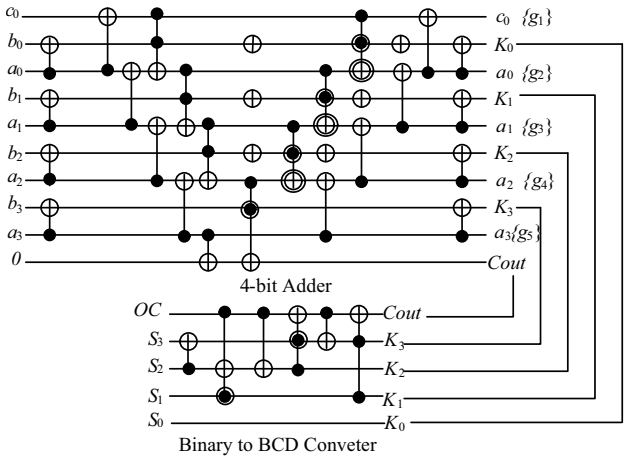
	AIs	GOs	QC	Delay Δ
[11]	7	10	55	-
[12]	4	8	169	-
Design 3[13]	2	6	103	-
Proposed design	1	5	70	57

V. CONCLUSIONS

In conclusions, we have presented efficient design of reversible BCD adder primarily optimizing the parameters of number of ancilla input bits and garbage outputs. The optimization of the quantum cost and the delay are also considered. The efficient design of the BCD adder depends on the design methodology used for designing the reversible ripple carry adder and the reversible binary to BCD converter. Thus for future research, efficient design schemes for reversible ripple carry adder and the reversible binary to BCD converter is an interesting area to investigate.



(a) Design of conventional BCD adder



(b) Proposed design of reversible BCD adder with input carry

Fig. 4. Proposed reversible BCD adder

REFERENCES

[1] L. Wang, M. Erle, C. Tsen, E. M. Schwarz, and M. J.Schulte, "A survey of hardware designs for decimal arithmetic," *IBM J. Research and Development*, vol. 54, no. 2, pp. 8:1 – 8:15, 2010.

[2] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. New York: Cambridge Univ. Press, 2000.

[3] Y. Takahashi, "Quantum arithmetic circuits: a survey," *IEICE Trans. Fundamentals*, vol. E92-A, no. 5, pp. 276–1283, 2010.

[4] Y. Takahashi and N. Kunihiro, "A linear-size quantum circuit for addition with no ancillary qubits," *Quantum Information and Computation*, vol. 5, no. 6, p. 440448, 2005.

[5] T. Toffoli, "Reversible computing," MIT Lab for Computer Science, Tech. Rep. Tech memo MIT/LCS/TM-151, 1980.

[6] A. Peres, "Reversible logic and quantum computers," *Phys. Rev. A, Gen. Phys.*, vol. 32, no. 6, pp. 3266–3276, Dec. 1985.

[7] H. Thapliyal and N. Ranganathan, "Design of reversible sequential circuits optimizing quantum cost, delay and garbage outputs," *ACM Journal of Emerging Technologies in Computing Systems*, vol. 6, no. 4, Article 14, pp. 14:1–14:35, Dec. 2010.

[8] H. Thapliyal and N. Ranganathan, "Design of efficient reversible binary subtractors based on a new reversible gate," in *Proc. the IEEE Computer Society Annual Symposium on VLSI*, Tampa, Florida, May 2009, pp. 229–234.

[9] S. A. Cuccaro, T. G. Draper, S. A. Kutin, and D. P. Moulton, "A new quantum ripple-carry addition circuit," *http://arXiv.org/quant-ph/0410184*, Oct 2004.

[10] Y. Takahashi, S. Tani, and N. Kunihiro, "Quantum addition circuits and unbounded fan-out," *http://arxiv.org/abs/0910.2530*, Oct 2009.

[11] A. K. Biswas, M. M. Hasan, A. R. Chowdhury, and H. M. Hasan Babu, "Efficient approaches for designing reversible binary coded decimal adders," *Microelectron. J.*, vol. 39, no. 12, pp. 1693–1703, 2008.

[12] M. Thomsen and R.Glück, "Optimized reversible binary-coded decimal adders," *J. Syst. Archit.*, vol. 54, no. 7, pp. 697–706, 2008.

[13] M. Mohammadi, M. Haghparast, M. Eshghi, and K. Navi, "Minimization optimization of reversible bcd-full adder/subtractor using genetic algorithm and don't care concept," *International J. Quantum Information*, vol. 7, no. 5, pp. 969–989, 2009.