Journal of Circuits, Systems, and Computers
Vol. 18, No. 2 (2009) 311–323

World Scientific
www.worldscientific.com

OPTIMIZED REVERSIBLE MULTIPLIER CIRCUIT

MAJID HAGHPARAST*

Department of Computer Engineering,
Science and Research Branch,
Islamic Azad University, Tehran, Iran
haghparast@srbiau.ac.ir

MAJID MOHAMMADI†, KEIVAN NAVI‡
and MOHAMMAD ESHGHI§

Faculty of Electrical and Computer Engineering,
Shahid Beheshti University, Tehran, Iran,
†*m_mohamadi@sbu.ac.ir*
‡*navi@sbu.ac.ir*
§*m-eshghi@sbu.ac.ir*

Revised 8 September 2008

Reversible logic circuits have received significant attention in quantum computing, low power CMOS design, optical information processing, DNA computing, bioinformatics, and nanotechnology. This paper presents two new 4×4 bit reversible multiplier designs which have lower hardware complexity, less garbage bits, less quantum cost and less constant inputs than previous ones, and can be generalized to construct efficient reversible $n \times n$ bit multipliers. An implementation of reversible HNG is also presented. This implementation shows that the full adder design using HNG is one of the best designs in term of quantum cost. An implementation of MKG is also presented in order to have a fair comparison between our proposed reversible multiplier designs and the existing counterparts. The proposed reversible multipliers are optimized in terms of quantum cost, number of constant inputs, number of garbage outputs and hardware complexity. They can be used to construct more complex systems in nanotechnology.

Keywords: Reversible logic gate; reversible logic circuit; reversible multiplier; quantum computing; nanotechnology.

1. Introduction

Low power circuit design is one of the most interested topics in current researches of hardware designers. Landauer in the early 1960s proved that irreversible computation, regardless of its realization technique, results in energy dissipation, due to the information loss.[1] For every bit of information that is erased, K.T.ln2 joules of energy dissipates as heat, where $K = 1.3806505 \times 10^{-23}\mathrm{m}^2\mathrm{kg}^{-2}\mathrm{K}^{-1}$ (joules

*Corresponding author.

Kelvin^{-1}) is the Boltzmann's constant and T is the absolute temperature at which operation is performed.[2] Theoretically, a reversible circuit has zero internal power dissipation because it does not lose information. A circuit is reversible if and only if the input vector can be uniquely recovered from the output vector. That is, there is a one-to-one correspondence between its input and output assignments.[3–5] Consequently, a reversible logic gate or circuit has to have equal number of inputs and outputs. Such gates or circuits allow the reproduction of the inputs from observed outputs and we can determine the inputs from the outputs.[3,4,6] 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.

Quantum gates, on the other hand, are introduced based on quantum computing theory. Since a closed quantum mechanical system is inherently reversible, quantum circuits are also reversible. Technically, the 1×1 and 2×2 quantum gates are realized in some quantum techniques.[7] However, the bigger gates such as 3×3 gates cannot be directly realized in quantum technology. Therefore we use the 1×1 and 2×2 gates to implement the bigger ones. The quantum cost (QC) of a reversible or quantum circuit is defined as the number of 1×1 or 2×2 gates used to implement the circuit.

Synthesis of quantum or reversible logic circuits is significantly more complicated than traditional irreversible logic circuits. In a reversible logic circuit, fan-out and feedback are not permitted.[3] Some of the main measures in designing an efficient reversible logic circuit are: Number of gates (NOG), Number of garbage outputs (Gout), Number of constant inputs (Gin), Total quantum cost (QC), and Total logical calculations (circuit cost).

Garbage outputs are some outputs that are not used for further computations in the circuit.[5] The inputs that are added to an $n \times k$ function to make it reversible are called constant inputs.[8]

Multiplication is a heavily used arithmetic operation in many computational units. It is necessary for the processors to have high speed multipliers. This paper presents two new 4×4 bit reversible multiplier circuits. These proposed reversible 4×4 multiplier designs has lower hardware complexity, less garbage bits, less quantum cost and less constant inputs than previous ones, and can be generalized to construct fast reversible $n \times n$ bit multipliers. The proposed reversible multiplier designs are superior in these respects to all the existing counterparts.[9–12]

An implementation of reversible HNG[13] using quantum gates is also proposed. We use reversible HNG in one of our multiplier designs. The implementation of MKG is also proposed. The MKG is not used in our designs, but included for comparison with previous reversible multiplier design that made use of it. The quantum realization of HNG shows that the full adder circuit which is designed using the HNG has minimum quantum cost among all the full adder designs in reversible logic literature.

Background of reversible and quantum logic gates are given in Sec. 2. The proposed equivalent circuit implementations of MKG and HNG are presented in Sec. 3. The proposed reversible multiplier circuit designs are described in Sec. 4. Evaluation of the proposed reversible multiplier circuits are presented in Sec. 5. Conclusions are presented in Sec. 6. A comprehensive list of references is also provided.

2. Background

This section reviews the reversible logic gates that are used in this paper. In addition, some other reversible logic gates are presented to allow for comparison with related studies. At the end of this section we will discuss about quantum gates.

An n-input n-output function F is said to be reversible if and only if there is a one-to-one correspondence between the inputs and the outputs. In the other words, the input vector can be uniquely recovered from the output vector. An $n \times n$ reversible logic gate can be represented as:

$$I_v = (I_1, I_2, I_3, \ldots, I_n),$$
$$O_v = (O_1, O_2, O_3, \ldots, O_n),$$

where I_v and O_v are input and output vectors respectively. Several reversible logic gates have been proposed in the past years[13–20] such as: Feynman gate (FG[14]), Toffoli gate (TG[15]), Fredkin gate (FRG[16]), Peres gate (PG[17]), New gate (NG[18]), MKG[19], HNG[13] and TSG.[20] In this section we review these reversible logic gates.

Feynman gate (*FG*): Feynman gate also known as controlled-not gate (CNOT), is a 2×2 reversible gate that can be described by the equations: $P = A$, and $Q = A \oplus B$, where "A" is control bit and "B" is the target bit. It is shown in Fig. 1. If control input (A) is "1", the target output (Q) is NOT of B; otherwise, it is B. The Feynman gate can be used to copy a signal. If the B input in Fig. 1 is "0" then two outputs of gate are equal to A. Since fan-out is not allowed in reversible logic circuits, the Feynman gate is used as the fan-out gate to copy a signal. The QC of this gate is one.

Toffoli gate (*TG*): Toffoli gate also known as controlled controlled-NOT (CCNOT) is a 3×3 reversible logic gate. The Toffoli gate can be represented as:

$$I_v = (A, B, C),$$
$$O_v = (P = A, Q = B, R = AB \oplus C),$$

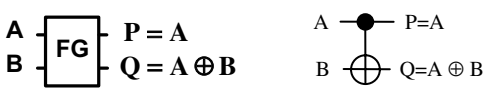


Fig. 1. Two symbols of Feynman gate.

Fig. 2. Symbol of Toffoli gate and its functionality.

where I_v and O_v are input and output vectors. The Toffoli gate is shown in Fig. 2. The Toffoli gate is universal, i.e., any reversible logic circuit can be constructed by a finite number of Toffoli gates and constant inputs. If the target input of this gate (C input) is set to "0", the target output (R) is A.B. The QC of this gate is five, i.e., it is implemented using five quantum 2×2 gates.

Fredkin gate (*FRG*): Fredkin gate also known as controlled permutation gate, is a 3×3 reversible logic gate. It can be represented as:

$$I_v = (A, B, C),$$
$$O_v = (P = A, Q = A'B \oplus AC, R = A'C \oplus AB),$$

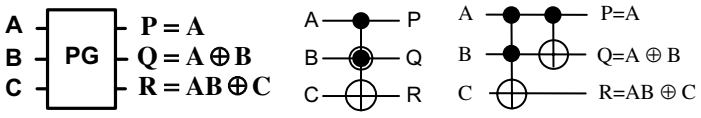
where I_v and O_v are input and output vectors. It is shown in Fig. 3. Fredkin gate is a conservative gate, that is, the Hamming weight of its input vector is the same as the Hamming weight of its output vector. This gate has QC of 5.

Peres gate (*PG*): Peres gate also known as New Toffoli gate (NTG), constructed by one Toffoli and one Feynman gate is a 3×3 reversible logic gate. It can be represented as:

$$I_v = (A, B, C),$$
$$O_v = (P = A, Q = A \oplus B, R = AB \oplus C),$$

where I_v and O_v are the input and output vectors. The Peres gate is shown in Fig. 4. Peres gate is equal to the transformation produced by a Toffoli gate followed by a Feynman gate. The symbol and functionality of Peres gate are shown in Fig. 4(a).

Fig. 3. Fredkin gate.



(a) (b) (c)

Fig. 4. Peres gate: (a) symbol and functionality, (b) the symbol used in this paper (c) equivalent circuit.

Another symbol, presented in Ref. 21 is depicted in Fig. 4(b). Its equivalent circuit is shown in Fig. 4(c). The Peres gate is also universal. Though it seems more complex than the Toffoli gate, its QC is 4 which is less than QC of a Toffoli gate.

New gate (*NG*): New gate is a 3×3 reversible gate. It can be represented as:

$$I_v = (A, B, C),$$
$$O_v = (P = A, Q = AB \oplus C, R = A'C' \oplus B'),$$

where I_v and O_v are the input and output vectors. The symbol of NG is shown in Fig. 5. The implementation of this gate is reported in Ref. 21 and its QC is 7.

TSG gate: TSG gate is a 4×4 reversible logic gate. The TSG gate is shown in Fig. 6, where each output is annotated with the corresponding logic expression. The TSG is universal. Its equivalent circuit shows that the TSG has QC of 18. The optimized QC of this gate, reported in Ref. 21, is 14.

MKG gate: MKG is a 4×4 reversible logic gate. The MKG can be represented as:

$$I_v = (A, B, C, D),$$
$$O_v = (P = A, Q = C, R = (A'D' \oplus B') \oplus C,\ S = (A'D' \oplus B') \cdot C \oplus (AB \oplus D)),$$

where I_v and O_v are the input and output vectors. The MKG is universal. It is shown in Fig. 7, where each output is annotated with the corresponding logic expression. In Sec. 4 we show that the QC of the MKG is 13.

HNG: HNG is 4×4 reversible logic circuit. The HNG can be represented as:

$$I_v = (A, B, C, D),$$
$$O_v = (P = A, Q = B, R = A \oplus B \oplus C, S = (A \oplus B) \cdot C \oplus AB \oplus D),$$

where I_v and O_v are the input and output vectors. The HNG is universal. The HNG is shown in Fig. 8, where each output is annotated with the corresponding logic expression.

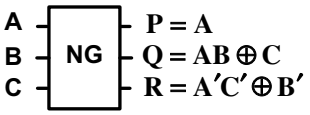
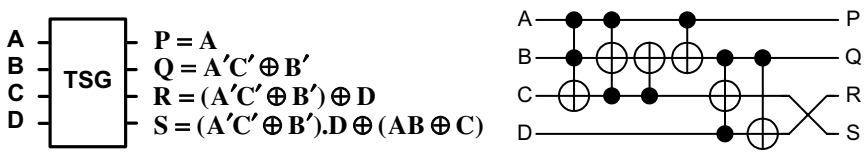


Fig. 5. New gate.



(a) (b)

Fig. 6. (a) TSG gate, (b) synthesis of TSG using Toffoli gates.

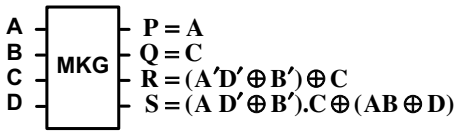


Fig. 7. MKG.

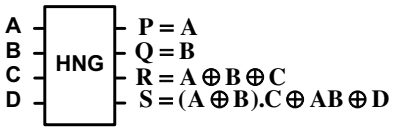


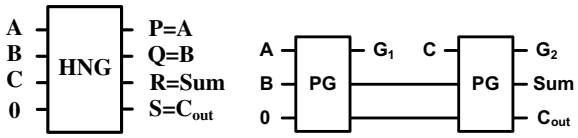
Fig. 8. Symbol of HNG.

HNG can work singly as a reversible full adder. To the best of our knowledge the minimum QC reported for FA is 6.[21] In Sec. 4 we show that the QC of the HNG is also 6. Thus, the QC of the HNG full adder is minimum possible QC for a full adder design. On the other hand Peres full adder needs two Peres gates.[14] Thus, its QC is 8. We use these two full adder circuits later to construct our multiplier designs.

The Feynman, Fredkin, NG, TSG and MKG are not used in our proposed designs, but included for comparison with existing reversible multiplier designs that make use of these gates.[9–12] A basic circuit that is used in our multiplier designs is full adder. Two different full adders which we have used in our designs are depicted in Figs. 9(a) and 9(b).

Quantum gates are defined based on quantum computing theory. Quantum gates act on small units of quantum data, named qubits (or quantum bits). A qubit is shown by a two-dimensional vector. The quantum gates are shown by unitary matrices.[7,22] A one-qubit quantum gate is shown as a 2×2 unitary matrix. For instance, the Hadamard gate operates on a single qubit. It is represented by the Hadamard matrix (Eq. (1)). Since the rows of the matrix are orthogonal, H is a unitary matrix.

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \tag{1}$$



(a) (b)

Fig. 9. Two implementations of full adder using: (a) HNG,[13] (b) Peres gate.[14]

The controlled-U gate is a gate that operates on two qubits in such a way that the first qubit serves as a control (Eq. (2)).

$$C(U) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & x_{00} & x_{01} \\ 0 & 0 & x_{10} & x_{11} \end{bmatrix}. \tag{2}$$

Two well known quantum gates are V and V^+. The V gate, named square root of NOT, have properties expressed in Eqs. (3) and (4).

$$V = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}^{1/2} = \frac{1+i}{2} \begin{bmatrix} 1 & -i \\ -i & 1 \end{bmatrix}, \tag{3}$$

$$\begin{cases} V \times V = NOT, \\ V \times V^+ = I. \end{cases} \tag{4}$$

Since the quantum gates larger than 2×2 are not directly realizable in the current quantum technology, the quantum cost (QC) is defined as the number of 1×1 or 2×2 quantum gates, needed to implement the gate or circuit.

3. Implementation of MKG and HNG

In this section we propose the implementation of MKG and HNG circuits. We show that the QC of MKG is 13, and the QC of HNG is just 6.

3.1. *Implementation of MKG*

Our proposed implementation of MKG is shown in Fig. 10. The circuit consists of a Feynman plus three Peres gates. Thus the QC of MKG is $1 + 3 * 4 = 13$.

3.2. *Implementation of HNG*

Recently, Thomsen and Gluck in Ref. 23 stated that the HNG is a complex 4×4 reversible logic gate. Because of its complex functionality, it seems the HNG has a large QC; however, our quantum implementation of the HNG (Fig. 11) shows that it has QC of 6. We can verify that this circuit (Fig. 11) is equivalent to the HNG. The P and Q outputs are equal to A and B, respectively. As shown in Fig. 8, R is $A \oplus B \oplus C$. The last output (S) cannot directly be expressed in terms

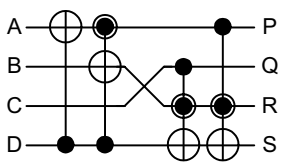


Fig. 10. Our proposed implementation of MKG.

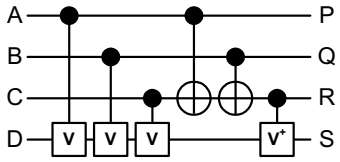


Fig. 11. Our proposed implementation for HNG. Its QC is 6.

of inputs, because of the existence of quantum V and V^+ gates. To verify the S output, we can write the truth table for it. For example if $ABCD$ inputs are 1101, two first V gates are active and act as a NOT gate (Eq. (4)). The V^+ gate is inactive because its control input is "0". Thus, in the D to S path there is a NOT operation that results to $S =$ "0". From the annotation of S we also obtain $S = (1 \oplus 1).0 \oplus 1.1 \oplus 1 = 0 \oplus 1 \oplus 1 = 0$. Other 15 states can be checked in the same manner.

4. New Reversible Multiplier Circuits

The operation of the 4×4 multiplier is depicted in Fig. 12. It consists of 16 partial product bits of the form $x_i \cdot y_i$.

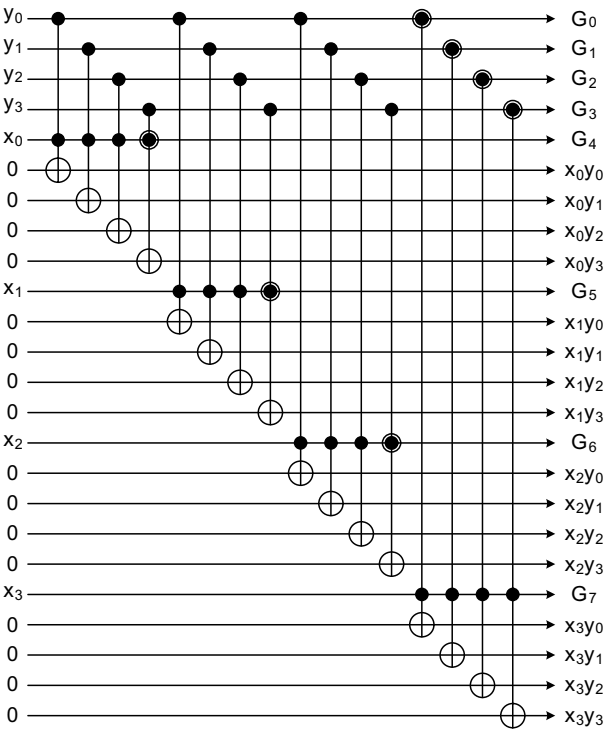
The proposed reversible 4×4 multiplier circuits have two parts. First, the partial products are generated in parallel using Toffoli and Peres gates as shown in Fig. 13(a). Then, the addition is performed as shown in Fig. 13(b).

The basic cell for such a multiplier is a full adder accepting three bits. We use HNG full adder and Peres full adder as reversible full adders which are depicted in Figs. 9(a) and 9(b). The proposed reversible multiplier circuits use eight reversible full adders. In the first design, we use HNG as full adder, and in the second approach we use Peres full adder. In addition, they need four reversible half adders. We use Peres gate as reversible half adder.

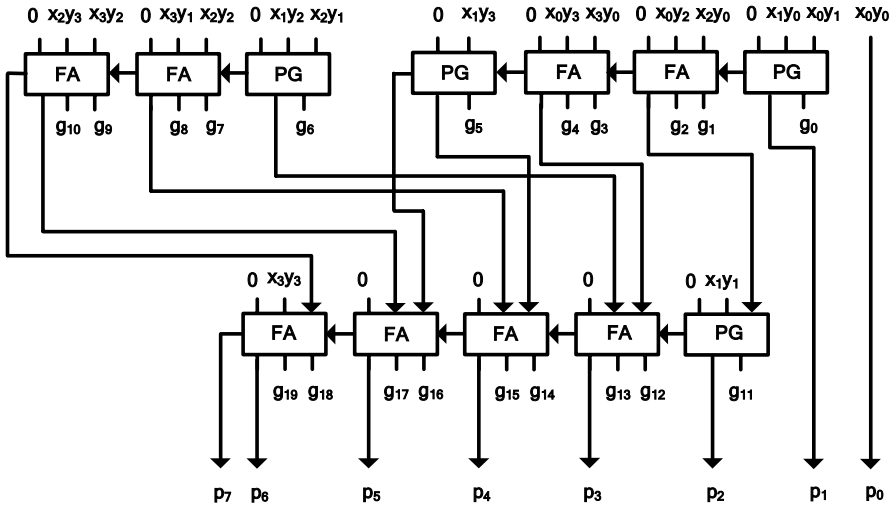
We make two choices for the full adder used in the multiplier circuit. If the design of Fig. 9(a) is used, then the QC of full adder is 6, whereas using design of Fig. 9(b) results to the QC of 8. Thus we have two designs based on which of the full adder circuits is selected. It has to be noted that if we use HNG full adder, then the summation network is the same as our proposed one in Ref. 12.

				x_3	x_2	x_1	x_0
			x	y_3	y_2	y_1	y_0
				x_3y_0	x_2y_0	x_1y_0	x_0y_0
			x_3y_1	x_2y_1	x_1y_1	x_0y_1	
		x_3y_2	x_2y_2	x_1y_2	x_0y_2		
	x_3y_3	x_2y_3	x_1y_3	x_0y_3			
P_7	P_6	P_5	P_4	P_3	P_2	P_1	P_0

Fig. 12. Partial products in a 4×4 multiplication.



(a) Reversible partial products generation circuit



(b) Summation Network. Design 1 uses HNG full adder, Design 2 uses Peres full adder.

Fig. 13. Proposed 4×4 reversible multiplier circuits.

Table 1. Comparative experimental results of different reversible multiplier circuits.

Reversible multiplier	No. of gates	No. of garbage outputs	Total quantum cost	No. of constant inputs	Total logical calculation
Design 1 (HNG, PG, TG)	28	28	137	28	$71\alpha + 36\beta$
Design 2 (PG, TG)	36	28	153	28	$63\alpha + 36\beta$
Ref. 12 (HNG, PG, FG)	$28 + 12\text{FG} = 40$	52	140	40	$92\alpha + 36\beta$
Ref. 11 (MKG, PG, FG)	$28 + 12\text{FG} = 40$	56	232	44	$104\alpha + 52\beta + 36\delta$
Ref. 10 (TSG, FRG, FG)	$29 + 12\text{FG} = 41$	58	274	46	$122\alpha + 103\beta + 71\delta$
Ref. 9 (NG, PG, FRG, FG)	$40 + 12\text{FG} = 52$	56	224	43	$92\alpha + 100\beta + 68\delta$
Improvement	(10–30)%	46.15%	0–2.14%	30%	(8.59–22.66)%

5. Evaluation of the Proposed Reversible Multiplier Circuits

The proposed reversible multiplier circuit is more efficient than the existing circuits presented in Refs. 9–12. Evaluation can be comprehended easily with the help of the comparative results in Table 1.

One of the main factors of a circuit is its hardware complexity. We can prove that our proposed circuits are better than the existing approaches in term of hardware complexity.

Let

α = A two input EX-OR gate calculation
β = A two input AND gate calculation
δ = A NOT calculation
T = Total logical calculation

Total logical calculation is the count of the XOR, AND, NOT logic in the output expressions. For example, PG has two XORs and one AND in the output expressions.

Therefore: $T_{(\text{PG})} = 2\alpha + 1\beta$.

It is to be noted that in reversible logic we can use Feynman gate for making fan-out. Thus, if we have two equal output expressions (or part of the output expressions), then we can copy it with Feynman. In this situation, the logical complexity is just one XOR more than the expression. For example, for MKG we need 1XOR + 1AND + 3NOT to produce $(A'D' \oplus B')$. Then we make a copy of that with just one XOR. Thus we can state that the total logical calculation for MKG is: $T_{(\text{MKG})} = (1\alpha + 1\beta + 3\delta) + (1\alpha)$ (for copy) $+ (1\alpha)$ (for last XOR of the R-expression) $+ (2\alpha + 2\beta)$ (for last part of S-expression) $= 5\alpha + 3\beta + 3\delta$.

For[9] the Total logical calculation is: $T = 92\alpha + 100\beta + 68\delta$, for[10] the Total logical calculation is: $T = 122\alpha + 103\beta + 71\delta$, for[11] the Total logical calculation is: $T = 104\alpha + 52\beta + 36\delta$, for[12] the Total logical calculation is: $T = 92\alpha + 36\beta$, for

proposed Design 1, the Total logical calculation is: $T = 9 * (1\alpha + 1\beta)$(for TG) + $11 * (2\alpha + 1\beta)$ (for PG) $+ 8 * (5\alpha + 2\beta)$ (for HNG) $= 71\alpha + 36\beta$, and for proposed Design 2, the Total logical calculation is: $T = 9 * (1\alpha + 1\beta)$ (for TG) $+ 27 * (2\alpha +$ $1\beta)$ (for PG) $= 63\alpha + 36\beta$. Therefore, our proposed designs are better than the existing circuits in term of complexity. Design 2 is better than Design 1 in term of total logical calculations.

Garbage output refers to the output of the reversible gate that is not used as a primary output or as input to other gates. Our proposed reversible multiplier circuits produce 28 garbage outputs, but the design in Ref. 9 produces 56 garbage outputs; the design in Ref. 10 produces 58 garbage outputs; the design in Ref. 11 produces 56 garbage outputs, and the design in Ref. 12 produces 52 garbage outputs. So, we can state that our designs are also better than all the existing counterparts in term of number of garbage outputs.

Number of constant inputs is one of the other main factors in designing a reversible logic circuit. The input that is added to an $n \times k$ function to make it reversible is called constant input.[8] Our proposed reversible multiplier designs require 28 constant inputs, but the design in Ref. 9 requires 43 constant inputs; the design in Ref. 10 requires 46 constant inputs; the design in Ref. 11 requires 44 constant inputs; and the design in Ref. 12 requires 40 constant inputs. So, we can state that our proposed designs are also better than all the existing designs in term of number of constant inputs.

Comparing our proposed reversible multiplier designs with the existing circuits in Refs. 9–12, it is found that the proposed Design 1 requires 28 reversible logic gates and the proposed Design 2 requires 36 reversible logic gates but the existing design in Ref. 9 requires 52 reversible gates; the existing design in Ref. 10 requires 41 reversible gates; the existing design in Ref. 11 requires 40 reversible gates; and the existing design in Ref. 12 requires 40 reversible gates. So, we can state that our design approach is also better than all the existing designs in term of number of reversible logic gates.

Total quantum cost is one of the other main factors in designing a reversible logic circuit. Total quantum cost of our proposed Design 1 is 137, but the total quantum cost of the existing design in Ref. 9 is 224; the total quantum cost of the existing design in Ref. 10 is 274; the total quantum cost of the existing design in Ref. 11 is 232; and the total quantum cost of the existing design in Ref. 12 is 140. It is to be noted that the total quantum cost of our proposed Design 2 is 153. So, we can state that our approach of Design 1 is also better than all the existing designs in term of total quantum cost, which is an important measure of merit to evaluate a reversible logic design.

From the above discussion we can conclude that the number of constant inputs and garbage output are drastically reduced. Now, if hardware complexity is the most important parameter, then it is better to use Design 2, and if the QC of the circuit is the most important parameter, then the propounded reversible multiplier circuit Design 1 is suggested.

6. Conclusion and Future Work

In this paper, we present two novel 4×4 bit reversible multiplier circuit designs. Table 1 demonstrates that the proposed reversible multipliers are better than the existing designs in term of hardware complexity, number of gates, garbage outputs, constant inputs, and total quantum cost. Furthermore, realizations of MKG and HNG are presented. Our proposed reversible multiplier circuits can be applied to the design of complex systems in nanotechnology.

As future work, some other techniques to reduce the garbage outputs and constant inputs (as reversible update used in Ref. 24) might be possible. In addition, some other optimization techniques such as genetic algorithm may be used to reduce the quantum cost of the circuit.

References

1. R. Landauer, Irreversibility and heat generation in the computing process, *IBM J. Res. Dev.* **5** (1961) 183–191.
2. C. H. Bennett, Logical reversibility of computation, *IBM J. Res. Dev.* **17** (1973) 525–532.
3. M. Perkowski, 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'01, Starkville* (2001), pp. 119–138.
4. M. Perkowski and P. Kerntopf, Reversible logic, *Proc. EURO-MICRO*, Warsaw, Poland (2001).
5. H. Thapliyal and M. B. Srinivas, Novel reversible TSG gate and its application for designing reversible carry look ahead adder and other adder architectures, *Proc. 10th Asia-Pacific Computer Systems Architecture Conference (ACSAC'05)*, Lecture Notes in Computer Science, Vol. 3740 (Springer-Verlag 2005), pp. 775–786.
6. P. Kerntopf, M. A. Perkowski and M. H. A. Khan, On universality of general reversible multiple valued logic gates, *IEEE Proc. 34th Int. Symp. Multiple Valued Logic (ISMVL'04)* (2004), pp. 68–73.
7. P. Kaye, R. Laflamme and M. Mosca, *An Introduction to Quantum Computing* (Oxford University Press, 2007).
8. S. Islam and R. Islam, Minimization of reversible adder circuits, *Asian J. Inform. Tech.* **4** (2005) 1146–1151.
9. H. Thaplyal, M. B. Srinivas and H. R. Arabnia, A reversible version of 4×4 bit array multiplier with minimum gates and garbage outputs, *Int. Conf. Embedded System, Applications (ESA'05)*, Las Vegas, USA (2005), pp. 106–114.
10. H. Thaplyal and M. B. Srinivas, Novel reversible multiplier architecture using reversible TSG gate, *IEEE Int. Conf. Computer Systems and Applications* (2006), pp. 100–103.
11. M. Shams, M. Haghparast and K. Navi, Novel reversible multiplier circuit in nanotechnology, *World Appl. Sci. J.* **3** (2008) 806–810.
12. M. Haghparast, S. J. Jassbi, K. Navi and O. Hashemipour, Design of a novel reversible multiplier circuit using HNG gate in nanotechnology, *World Appl. Sci. J.* **3** (2008) 974–978.
13. M. Haghparast and K. Navi, A novel reversible BCD adder for nanotechnology based systems, *Am. J. Appl. Sci.* **5** (2008) 282–288.

14. R. Feynman, Quantum mechanical computers, *Optics News* **11** (1985) 11–20.
15. T. Toffoli, Reversible computing, Tech Memo MIT/LCS/TM-151, MIT Lab for Computer Science (1980).
16. E. Fredkin and T. Toffoli, Conservative logic, *Int. J. Theor. Phys.* **21** (1982) 219–253.
17. A. Peres, Reversible logic and quantum computers, *Phys. Rev. A* **32** (1985) 3266–3276.
18. M. H. A. Khan, Design of full adder with reversible gates, *Int. Conf. Computer and Information Technology*, Dhaka, Bangladesh (2002), pp. 515–519.
19. M. Haghparast and K. Navi, A novel reversible full adder circuit for nanotechnology based systems, *J. Appl. Sci.* **7** (2007) 3995–4000.
20. M. Haghparast and K. Navi, A novel fault tolerant reversible gate for nanotechnology based systems, *Am. J. Appl. Sci.* **5** (2008) 519–523.
21. M. Mohamadi, M. Eshghi and K. Navi, Optimizing the reversible full adder circuit, *IEEE EWDTS*, Yerevan (2007), pp. 312–315.
22. A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin and H. Weinfurter, Elementary gates for quantum computation, *Phys. Rev. A* **52** (1995) 3457–3467.
23. M. K. Thomsen and R. Gluck, Optimized reversible binary-coded decimal adders, *J. Syst. Archi.* (2007).
24. S. A. Cuccaro, T. G. Draper, S. A. Kutin and D. P. Moulton, A new quantum ripple-carry addition circuit, *8th Workshop Quantum Information Processing*, 2005.
25. M. Mohammadi and M. Eshghi, Heuristic methods to use don't cares in automated design of reversible and quantum logic circuits, *Quantum Inform. Process. J.* (2008).

March 27, 2009 19:43 WSPC/123-JCSC 00508