



Ultra-area-efficient reversible multiplier

Mariam Zomorodi Moghadam, Keivan Navi*

Shahid Beheshti University, G.C., Tehran 1983963113, Iran

ARTICLE INFO

Article history:

Received 16 July 2011

Received in revised form

12 February 2012

Accepted 28 February 2012

Available online 17 March 2012

Keywords:

Nanotechnology

Quantum computing

Reversible logic

Reversible circuits

Reversible multiplier design

ABSTRACT

One of the most promising technologies in designing low-power circuits is reversible computing. It is used in nanotechnology, quantum computing, quantum dot cellular automata (QCA), DNA computing, optical computing and in CMOS low-power designs. Because of this broad range of applications, extensive works have been proposed in constructing reversible gates and reversible circuits, including basic universal logic gates, adders and multipliers.

In this paper we have highlighted the design of reversible multipliers and have presented two designs. Integration of adder circuit and multiplier in the design is described, in order to utilize the unused capacity of the multipliers.

We have achieved reduction in quantum cost compared to similar designs as well as appending the adder circuit to the multiplier which leads to better usage of resources. Additionally, we have described the multiplier problem for implementing $n \times n$ reversible multiplier and analyzed the required resources in terms of n . Practical implementation of this design can be achieved with the existing technologies in CMOS and nanotechnology.

Lastly, we make a tradeoff between area and time complexity to obtain two designs which can be used in different situations where different requirements are of different importances. We compare the proposed designs with each other and also to the existing ones.

© 2012 Published by Elsevier Ltd.

1. Introduction

In the nanoscale design of today's circuits, the power consumption which leads to heat dissipation in computer machinery has become one of the major challenges and attracts the attention of many researchers. This challenge represents the strongest motivation to study the reversible computing field. According to [1,2,3] any computation can be reversibly performed both logically and thermodynamically and leads to dissipating arbitrarily little energy. Quantum physics is also reversible by its nature. This is because the reverse-time evolution specified by the unitary operator $U^{-1} = U^\dagger$ always exists and several workers recognized that reversible computation could be executed within a quantum-mechanical system. First, R. Landauer in 1961 [1] demonstrated that irreversibility in the computing process which leads to loss of information requires minimum heat generation in the order of KT for each irreversible function, where K is Boltzmann's constant and T is the absolute temperature at which the computation is performed. He argued that this feature is unavoidable since the computer performs irreversible operations. Then C.H. Bennett in 1973 [2] showed that an irreversible computer can always be made

reversible. Recently due to the need for low-power design and also emerging field of nanotechnology, reversible computing has become more attractive. It plays an important role in the field of low-power circuit designs and computational nanotechnology. The role of computational nanotechnology and nanomechanics has become critically important in the cycle of growth and development of nanotechnology [21].

Computational nanotechnology is a general term for any computing which uses nanotechnology for implementing the computing hardware. It is emerging as a fundamental engineering analysis tool for the novel designs of nanodevices [21]. Quantum computing is one of the research areas in this broad field of technology. Quantum computing and quantum devices are at the heart of the computational nanotechnology.

Quantum computing is reversible by its nature and that is why reversible computing is vital for nanotechnology. In other words, because logical irreversibility implies physical irreversibility, we need reversible logic in the high-level logical computation to take advantage of quantum devices for computation. There are many implementations of reversible logic in nanotechnology such as the one in [29]. Also in the literature there are MOS implementations of reversible gates such as the one in [30].

On the other hand, one of the bottlenecks in combinational circuit design which becomes worse in reversible circuits is the extremely large amount of hardware they require. One way to

* Corresponding author. Tel.: +982129902286; fax: +982122431804.
E-mail address: navi@sbu.ac.ir (K. Navi).

reduce the hardware amount is to make them by some factor serial. Therefore based on our requirements, we may select architecture with more area consumption or one with less hardware complexity but having more delay. This is the tradeoff we make in our designs. This paper also describes the integration of adder circuit and multiplier design which can share some gates and signals with the multiplier.

In this paper, two designs of reversible multipliers, which make more efficient use of reversible gates resources as well as some of the garbage outputs as useful inputs are introduced. In the following section, we describe the popular reversible gates as a basis for designing reversible circuits. Section 3 is a survey of the works carried out in this area, especially reversible multiplier designs. Sections 4 and 5 discuss our two designs of reversible multipliers, the generalization requirements and how to append the adder circuit to it. Section 6 compares our results with other works in terms of quantum cost, number of garbage outputs, etc. Finally, in Section 7 we present our conclusions of our work and some proposals for future work.

2. Reversible gates

A logical gate is considered reversible if the number of its inputs and outputs are equal (i.e. there is a one to one correspondence between the input and output vectors and any output pattern has a unique pre-image [9]). The input vector can be restored from the output vector and as discussed, this ensures that in an ideal situation there is no internal power dissipation in the system.

Several reversible gates have been proposed over the past decades. Among them Feynman gate [16], Toffoli gate [10,11], Fredkin gate [10,11], DFG (F2G) [20] and Peres gate [12] are the most popular in reversible and quantum literature and have been studied in detail. A comparison of their functionalities and size is shown in Figs. 1–5. These are 2×2 and 3×3 reversible gates, which means they are 2 input–2 output and 3 input–3 output reversible gates respectively.

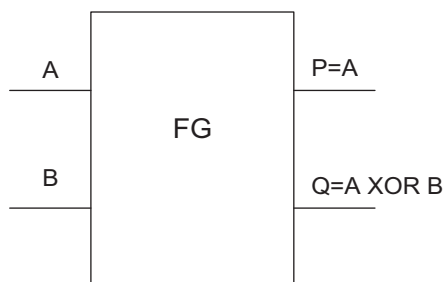


Fig. 1. Feynman gate.

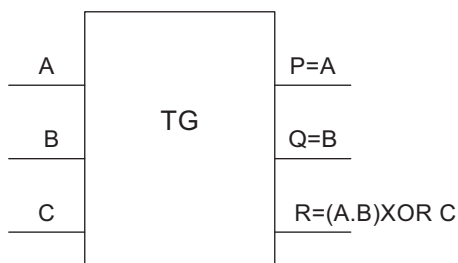


Fig. 2. Toffoli gate.

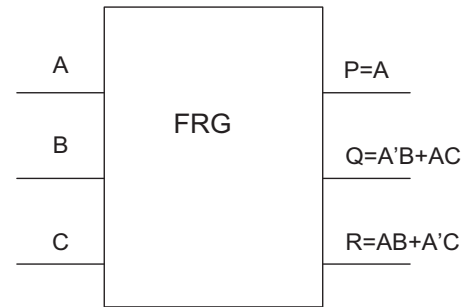


Fig. 3. Fredkin gate.

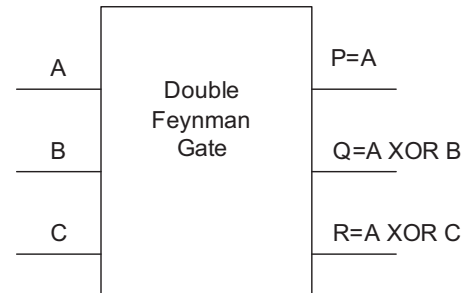


Fig. 4. DFG (F2G) gate.

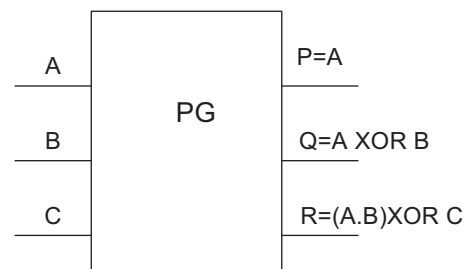


Fig. 5. Peres gate.

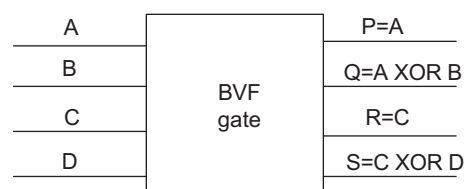


Fig. 6. BVF gate.

One of the existing reversible gates is a 4×4 BVF reversible gate. Having the input vector $I(A, B, C, D)$, this gate will result in the output vector equal to $O(P, Q, R, S) = (A, A+B, C, C+D)$. The implementation of this gate is given in Fig. 6. The DPG gate is another 4×4 reversible gate with implementation shown in Fig. 7. PFAG [14], TSG [24] and HNG [25] are other 4×4 reversible gates. Their implementations are given in Figs. 8, 9 and 10 respectively. All of them can implement Boolean functions and can also be used as a full adder.

3. Related works

In the recent years many reversible circuits as a replacement for conventional irreversible circuits have been proposed [4–8]. On the other hand, because of the extensive use of multipliers in

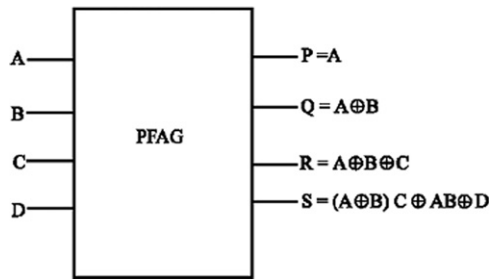


Fig. 7. DPG gate.

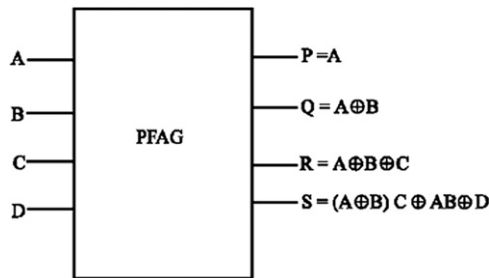


Fig. 8. PFAG gate.

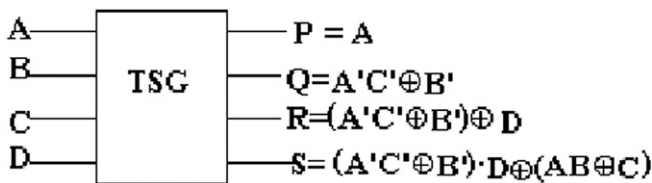


Fig. 9. TSG gate.

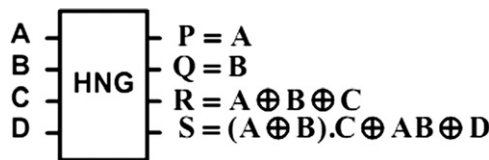


Fig. 10. HNG gate.

computer systems, several reversible circuits for implementing multipliers have been proposed [3,6,9,14–19,26]. For example, in [3], Haghparast et al. have introduced a two-part reversible multiplier circuit. The first part is for partial product generation and the second part takes the result from the first part and performs addition for producing the final result. The design uses an array of 16 PG gates for partial product generation and then addition is accomplished using a circuit which consists of PG [12] and HNG [22] gates. Other similar designs have been proposed by Islam et al. [14] and Shams et al. [15]. The only difference is in the type of gate used in the addition sub-circuit with the same partial product generation sub-circuit. Another study in reversible multiplier design has been proposed by Bhagyalakshmi and Venkatesha [6]. It consists of a fan-out generation circuit plus the partial product and additional circuits to form a 4×4 multiplier. Its fan-out circuit uses 12 BVF gates. In their design, they use three BVF gates as a building block to construct a circuit which takes its inputs as $(x_n, 0, y_n, 0)$ and makes 4 copies of x_n and 4 copies of y_n . So to have 4 copies of each of the 8 inputs in a 4×4 multiplier design, there is a need for 12 BVF gates as suggested. This design also uses a carry save adder (CSA) for reducing the four operands to two and then a carry propagating adder (CPA) is used for producing the final results for the last stage.

Table 1
QC of reversible gates.

Gate name	QC
Fredkin gate	5
Feynman gate	1
Toffoli gate	5
Peres gate	4

We have some criteria to compare the results of our proposed approach with the existing literatures. The cost of a reversible circuit is determined by four parameters: number of garbage outputs, number of constant inputs, number of gates and quantum cost. Number of garbage outputs refers to the number of outputs that are neither used as primary outputs nor for further computations, but they are added to make the circuit reversible [13,26,9]. Number of constant inputs refers to the number of inputs whose values are not to be changed in a given circuit and have to be maintained at either 0 or 1 in order for the circuit to work. They are also added to make the circuit reversible [13,26,27]. Number of gates refers to the total number of reversible gates in a given circuit. The quantum cost (QC) is defined as the number of 1×1 or 2×2 reversible quantum or logic gates that are required to realize the circuit [23,4]. Table 1 shows the quantum cost of some of the different reversible gates used in literatures.

4. Multiplier design

We applied a straightforward method for implementing a multiplier consisting of AND gates to form the partial products and then added the partial products using a set of carry-save adders (CSAs). Finally a standard ripple-carry adder is used to add the outputs of the CSA in the last stage, producing the final sum of the four original inputs. Using this scheme for adding 4 or more operands requires carry propagation as shown in Fig. 11. For $n \times n$ multipliers we have to continue this procedure, adding an input for each stage of full adders, without any intermediate carry propagation.

Although there are many other designs which are widely used in irreversible logic, we cannot say with certainty that they can be used in the same way and result in the same achievements in reversible logic, e.g. sequential schemes. For future work, we intend to study some of these alternatives.

For the sake of simplicity, we assumed that we have a 4×4 multiplier, but it could be easily extended to any size of n for $n \times n$ multiplication. In the next section we describe the requirements for designing n -bit multiplier with respect to n .

This design has two segments which work sequentially according to the method explained above. The output of the first stage is transferred to the second stage in order to produce the final result. These two stages include (1) partial product generation which will be carried out by a separate circuit and (2) addition of the partial products generated in the first stage. We explain each part in the following subsections.

4.1. Partial product generation

Partial products of an $n \times n$ multiplier require $n \times n$ 2-input AND operations. We used PG and TG gates respectively in consecutive designs for constructing these partial products as shown in Figs. 13 and 14. As can be seen from figures, in these designs, for each AND operation we have a constant value of logical zero for input C of third input of TG and PG gates to

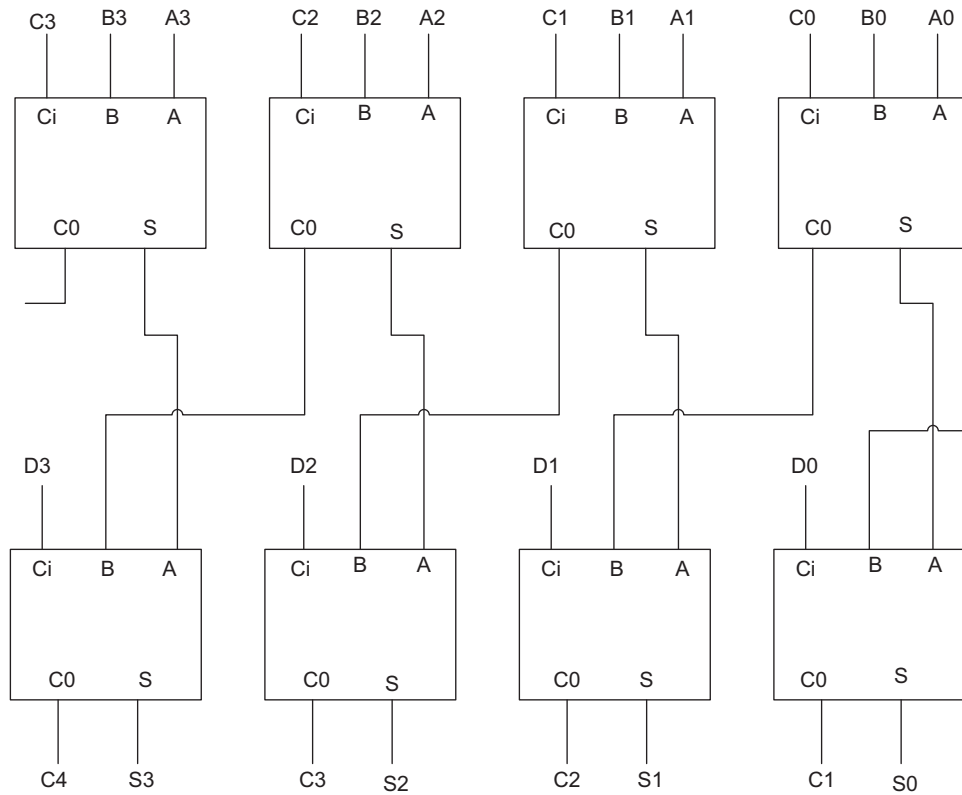


Fig. 11. Carry-save adder design.

produce AND from those reversible gates. These constant inputs are some sort of cost in the designs.

Because in reversible logic, fan-out is not considered reversible, it is not permissible to have a fan-out of a gate and for producing replications of signals we have to use a reversible gate. In the first proposed design, the fan-out generation circuit is reduced and in the second one it is eliminated. This part is responsible for generating fan-out signals for the operands of the partial products and at the same time reducing the number of garbage outputs in the circuit. In the first design we do not eliminate the fan-out generation circuit completely, but reduce the number of gates from 12 in the multiplier mentioned in [6] to 4 in our design and then in the design II we eliminate the fan-out generation circuit.

Every TG gate with inputs A, B and C and with constant value of logical zero in its C input, produces three outputs which are $P=A$, $Q=B$ and $R=A \cdot B$. The last output R is used for partial products. The number of AND operations needed is $n \times n$ therefore the same number of TG gates is required. In our design we used the P and Q outputs for fan-outs as discussed in Section 4.1.2.

4.1.1. Design I

In design I, we use a fan-out generator as shown in Fig. 12. In addition to other measures, the delay of the circuit is taken into account. Except in the newly published paper [28], there is no investigation of the delay of various reversible gates.

First, we make some assumption about the delay of several gates that we used:

$$\Delta_F : \text{Feynman gate delay} \quad (1)$$

$$\Delta_{DF} : \text{Double Feynman gate delay} \quad (2)$$

$$\Delta_P : \text{Peres gate delay} \quad (3)$$

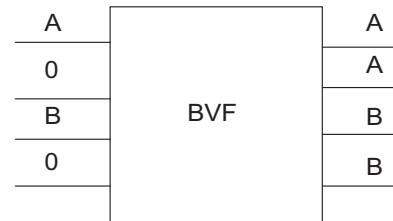


Fig. 12. Fan-out generator (design I).

$$\Delta_{BVF} : \text{BVF gate delay} \quad (4)$$

$$\Delta_{DPG} : \text{DPG gate delay} \quad (5)$$

$$\Delta_{TG} : \text{TG gate delay} \quad (6)$$

The circuit used in Fig. 12 is composed of 1 double Feynman gate and 1 Feynman gate. The first has a quantum cost of 2 and the second has a quantum cost of 1 with a total quantum cost of 3 for generating 4 fan-out signals.

For the consecutive stages of the partial product circuit shown in Fig. 13, the fan-out generator circuit produces inputs with a delay equal to Δ_{BVF} . The generated outputs are: four y_0 , four y_1 , four y_2 and four y_3 signals. Every BVF gate generates four output signals. The first two BVF gates generate four y_0 and four y_1 and the two others generate four y_2 and four y_3 signals.

In this model of 4×4 multipliers, we have the following equations of the overall delay of partial product sub-circuit:

$$\Delta_{\text{overall}} : 4\Delta_P + \Delta_{BVF} \quad (7)$$

where Δ_{overall} is the overall delay of the partial product sub-circuit.

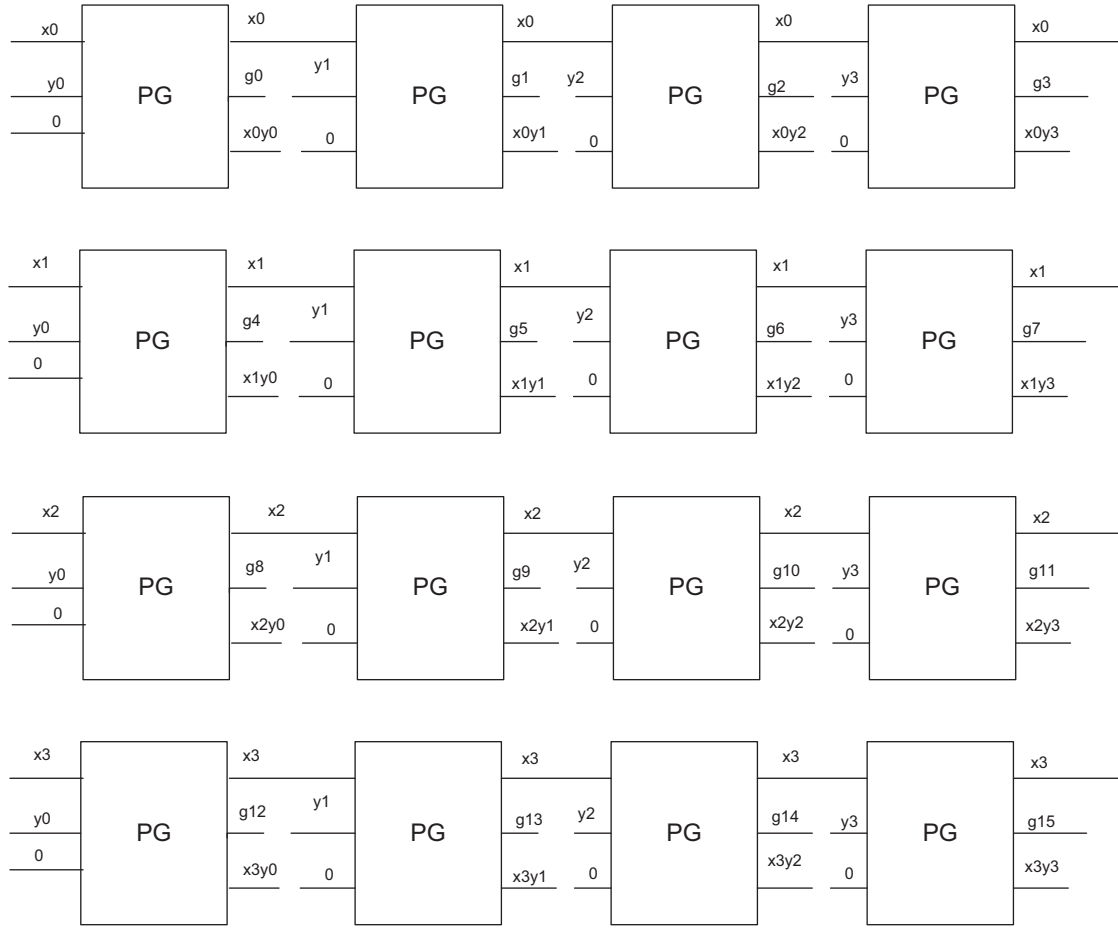


Fig. 13. Partial product generation—design I.

The PG gate prepares the signals x_0, x_1, x_2, x_3 for use in consecutive stages without any additional overhead. This also leads to reduction in garbage outputs in the PG gate.

4.1.2. Design II

We further use the garbage outputs to completely eliminate the need for fan-out generation circuit. In this design, we used a multi-stage process for partial product generation.

In each TG gate, there are two outputs that are not used for product generation. So we can use them for generating all of the required fan-out signals sequentially as shown in Fig. 14. Partial product generation uses TG gate in this new design. In the first stage only one TG gate works, the one which produces x_0y_0 , but this is not the only order and we can use a different order but it is of no importance. This gate produces two signals x_0 and y_0 required by gates 1.2 and 2.1. The other two required signals x_1 and x_0 are from the original output and gate 1.1 respectively. Generally, in the horizontal line in Fig. 14, there are the input signals of vector $\langle y \rangle$ and in the vertical line there are the input signals of vector $\langle x \rangle$ which come from the original input system and are not duplicated. In the second stage, since their inputs are ready, the gate that produces x_0y_1 and the gate that produces x_1y_0 work together. Gate 1.2 produces signals x_0 and y_1 and gate 2.1 produces signals x_1 and y_0 . In the next stage gate 1.3 uses signals x_0 and y_2 , the first is prepared by gate 1.2 and the second is prepared by original signal coming from outside. Gate 2.2 uses signals x_1 and y_1 , the first is prepared by gate 2.1 and the second is prepared by gate 1.2. Gate 3.1

uses signals x_2 and y_0 , the first is prepared by the original signal coming from outside, the second is prepared by gate 2.1 and so on.

This sort of output generation is to some extent similar to the way instruction pipelining works in processors. Calculating the delay of this scheme is straightforward:

$$\Delta_{overall} = 7\Delta_{TG} \quad (8)$$

In this design we have a reduction of 24 garbage outputs compared to one of [6] and there are only eight garbage outputs in the partial product circuit, which is a remarkable reduction.

The same parts in the reversible multiplier given in [6] have the following delay for execution:

$$\Delta_{overall} = \Delta_{PG} + 2\Delta_{BVF} \quad (9)$$

Consequently, we have an increase of Δ_t in this new design which is

$$\Delta_t = 7\Delta_{TG} - (\Delta_{TG} + 2\Delta_{BVF}) \quad (10)$$

If the delay of all reversible gates is the same and we take it Δ then Δ_t becomes

$$\Delta_t = 7\Delta - (\Delta + 2\Delta) = 4\Delta \quad (11)$$

Design I has a delay reduction of 2Δ of a reversible gate compared to design II and because of usability of TG in partial product generation against using PG in design I, the cost of design II is higher than that of design I. If we could use another reversible gate with quantum cost equal to or less than 4, then the design would be more efficient. Obviously this is the limitation

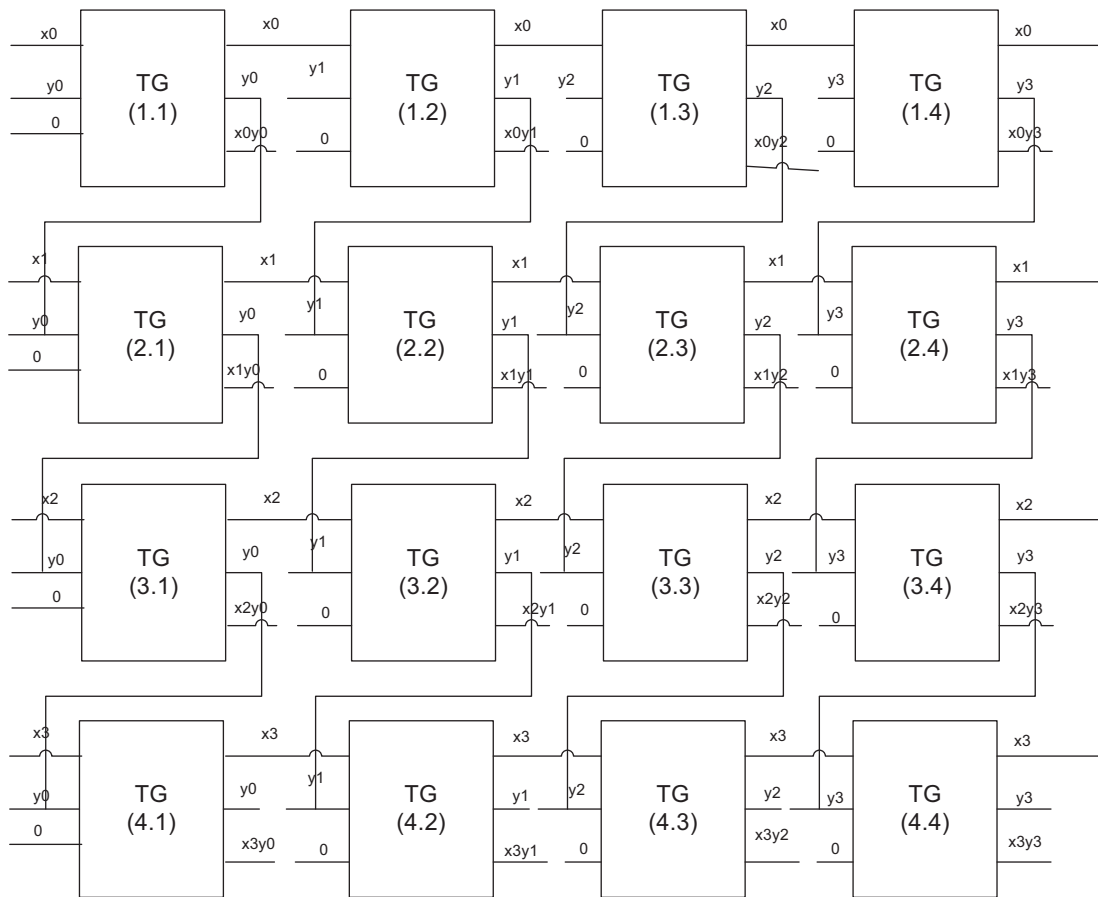


Fig. 14. Partial product generation—design II.

Table 2
Comparative results for fan-out generation sub-circuit.

Fan-out generator	Quantum cost	No. of constant inputs	No. of garbage outputs	No. of gates	Total cost
Design I	8	8	0	4	12
Design II	NA	NA	NA	NA	NA
Bhagyalakshmi and Venkatesha [6]	24	24	0	12	36

Table 3
Comparative results for partial product sub-circuit.

Partial product generator	Quantum cost	No. of constant inputs	No. of garbage outputs	No. of gates	Total cost
Design I	64	16	16	16	96
Design II	80	16	8	16	104
Bhagyalakshmi and Venkatesha [6]	64	16	32	16	112
Haghpars et al. [26]	76	16	8	16	100
Haghpars et al. [3]	64	16	32	16	112
Banerjee and Pathak [9]	80	16	8	16	104
Islam et al. [14]	64	16	32	16	112
Naderpour and Vafaei [18]	64	16	32	16	112
Thapliyal and Srinivas [19]	80	16	16	16	112

of existing reversible gates which could be solved in the future. We have identified this problem for future investigation by researchers.

The tradeoff in these approaches is the increased latency by Δ_t units due to the sequential execution in order to reduce number of garbage outputs and constant inputs in design II.

One can select from these two designs which represent the tradeoff between two choices, the design which improves the delay and another design which improves the gate count and area consumption. Our achievements have been highlighted in Tables 2–4. The other remarkable point in these two designs is their homogeneous structure which is a good criterion in

Table 4
Comparative results of different reversible multipliers.

Multiplier	Quantum cost	No. of constant inputs	No. of garbage outputs	No. of gates	Total cost
Design I	136	36	28	32	196
Design II	144	28	24	28	196
Bhagyalakshmi and Venkatesha [6]	152	52	52	40	244
Haghparast et al. [26]	140	28	28	28	196
Haghparast et al. [3]	152	52	52	52	256
Banerjee and Pathak [9]	168	76	50	80	298
Islam et al. [14]	144	28	52	52	290
Naderpour and Vafaei [18]	152	42	52	42	246
Thapliyal and Srinivas [19]	234	58	58	53	345

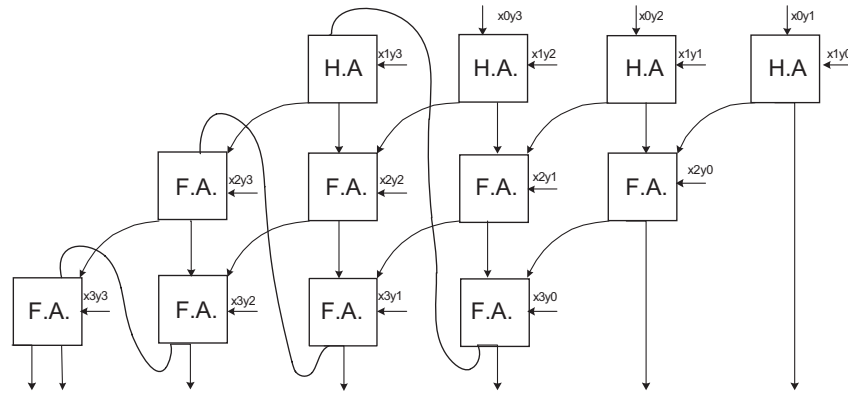


Fig. 15. Addition sub-circuit.

reversible implementations. They use the same gates in all parts of the design.

4.2. Addition sub-circuit

Addition of partial products is done with the carry save addition method. In this method the first row of addition will be half adders. The carries of each full-adder is diagonally forwarded to the next row of the addition sub-circuit. Use of carry save adder when performing a large number of addition operations needs to be performed quickly in order to be beneficial. In the fourth row, the carries of the circuit is added to the inputs of the addition as shown in Fig. 15. Only one standard ripple carry is required in order to add the outputs of the CSA in the last stage. Fig. 15 shows the structure of the addition sub-circuit of the proposed design. The introduced delay can be reduced using the approaches in other designs like the one in [3] which uses ripple carry adders.

Due to regular structure, this design can be easily extended to larger multipliers by placing one extra row in the design with each one-bit increase in the numbers and also one block of full-adder in each row.

5. Additional capabilities

Now we consider an $n \times n$ multiplier design. This process can be easily extended in order to have a word-width multiplier. In the addition sub-circuit of the multiplier, each extra bit in the multiplied numbers requires one extra row of full adders and also one full adder is added to each row, except for row one which requires a half adder. Therefore it can be easily computed that for an addition sub-circuit an $n \times n$ multiplier, " $n(n-2)$ " full adders and " n " half adders are required. Moreover, in the partial product

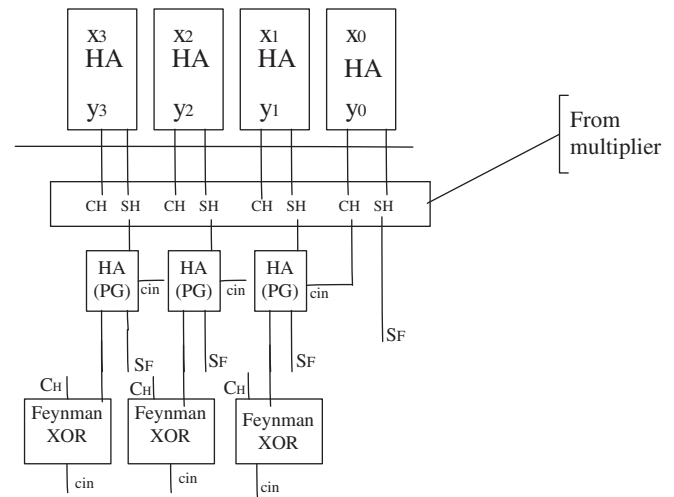


Fig. 16. Adder design with multiplier resources.

sub-circuit, the number of required TG or PG gates (depending on the design used) is $n \times n$.

Next, considering another part of the design and referring to design I of suggested reversible multiplier, we used a Peres gate for producing partial products in the first stage. Whereas the goal of designing each reversible arithmetic unit is to exploit it in a reversible ALU, we consider having an addition unit, so we can use the unused capacity of multiplier for designing adder circuit. In Design I, PG gate produces XOR of bits in multiplier and multiplicand. Also AND operation is prepared in the addition sub-circuit. Therefore by having another PG gate and one Feynman for each bit, this design can include two units in one package. Fig. 16 shows the implementation of adder based on the existing resources of multiplier circuit. PG gates are prepared with half

adders of every two corresponding bits in the numbers X and Y , i.e. a multiplier and a multiplicand. Another half adder plus one XOR are sufficient to implement each stage of adder circuit. The corresponding formulas are as follows:

$$\begin{aligned} S_H &= x_i \oplus y_i \\ C_H &= x_i y_i \end{aligned} \quad (12)$$

These two computations are ready due to the outputs of the multiplier circuit and they result in

$$\begin{aligned} S_F &= S_H \oplus C_{in} \\ C_F &= C_H \oplus S_H C_{in} \end{aligned} \quad (13)$$

where S_H and C_F are sum and carry outputs of the half adder respectively. Also S_F and C_F are the final sum and carry outputs of each level respectively. Finally C_{in} is the carry input which enters each column of the adder circuit from the previous level, i.e. C_F of the previous level.

So using the outputs of partial product stage for adder circuit we can reduce the number of garbage outputs. This reduction is 8 bits if we use design I and 4 if we use design II.

The next section presents comparison of various design approaches. In order to compare our results with other designs for multipliers that did not consider fan-out generation circuit, we added a circuit with 24 Feynman gates for fan-out generation which is a typical scheme as applied in [6]. Without it, there is no real comparison between our design and theirs.

6. Results

In this section we present some comparisons of proposed designs in the literatures with the suggested design in this paper. In multiplier design suggested in [6] by H.R. Bhagyalakshmi and M.K. Venkatesha there is a complete fan-out generation circuit which uses 12 reversible BVF gates for its implementation. In [26] a similar scheme which seems to use garbage outputs as fan-out signals has been considered. Results reported in other papers have not considered the fan-out generation circuit. The method used in [31] has considered the fan-out for multiplier design similar to our work, but it is for signed multiplier and its results cannot be compared to unsigned multipliers. In the result tables, except for [6] and [26], we considered a circuit with 24 Feynman gate for fan-out generation which is a typical scheme as applied in [6] for comparing our results with other designs. Therefore for correct comparison we are required to add the cost of this circuit to the results reported in other papers. The estimated cost would not be valid without considering the cost of this part of the design, since the fan-out circuit has a considerable impact on the cost of the design.

In the measurements, the sub-circuit of addition requires 12 reversible gates composed of 4 PG [12] and 8 DPG gates, having a quantum cost of 64 for the whole sub-circuit with 4 out of 20 garbage outputs used for the adder circuit.

Tables 2–4 give the comparative result of different reversible multipliers including our designs I and II. The total cost parameter as introduced in [9] is the sum of gate count, number of garbage bits and quantum cost of reversible circuit. Results of other factors are as follows:

- For the number of garbage outputs our designs have 28 and 24 garbage outputs respectively compared to 28 for the best existing design.
- For the number of gates our designs have 32 and 28 gates respectively compared to 28 for the best existing design.
- For the number of constant inputs: our designs have 36 and 28 constant inputs respectively compared to 28 for the best existing design.

- For the quantum cost: our designs have quantum cost of 136 and 144 gates respectively compared to 140 for the best existing design.

From the above discussion we can conclude that the proposed reversible multiplier circuit is better than all of the existing counterparts with an augmented capability of integrating the adder circuit to it.

7. Conclusions

In this paper, we proposed two designs for an $n \times n$ multiplier in reversible logic with lower hardware complexity in terms of quantum cost compared to other designs. The design also includes the integration of adder circuit and multiplier in order to construct a two unit processing element which has not been considered in other works. Another contribution of this work is sequential execution of different gates and making tradeoff between performance and area of the circuit in reversible logic. We have shown that this design has less quantum cost compared to other multipliers designed to date. For future work, we suggest designing of reversible multipliers with different logical designs proposed in conventional combinational and sequential logic with the aim to improve the performance and/or total cost.

References

- [1] R. Landauer, Irreversibility and heat generation in the computational process, *IBM J. Res. Dev.* 5 (1961) 183–191.
- [2] C.H. Bennett, Logical reversibility of computation, *IBM J. Res. Dev.* (1973) 525–532.
- [3] M. Haghparast, S. Jafaralijassbi, K. Navi, O. Hashemipour, Design of a novel reversible multiplier circuit using HNG gate in nanotechnology, *World Appl. Sci. J.* 3 (6) (2008) 974–978.
- [4] M. Mohammadi, M. Eshghi, M. Haghparast, A. Bahrololoom, Design and optimization of reversible BCD adder/subtractor circuit for quantum and nanotechnology based systems, *World Appl. Sci. J.* 4 (6) (2008) 787–792.
- [5] P.K. Lala, Adder designs using reversible logic gates, *WSEAS Trans. Circuits Syst.* 9 (6) (2010) 369–378.
- [6] H.R. Bhagyalakshmi, M.K. Venkatesha, An improved design of a multiplier using reversible logic gates, *Int. J. Eng. Sci. Technol.* 2 (8) (2010) 3838–3845.
- [7] H.G. Rangaraju, U. Venugopal, K.N. Muralidhara, K.B. RAJA, Low power reversible parallel binary adder/subtractor, *Int. J. VLSI Des. Commun. Syst. (VLSICS)* 1 (3) (2010) 23–34.
- [8] G. Yang, W.N.N. Hung, X. Song, M. Perkowski, Majority-based reversible logic gates, *Theor. Comput. Sci.* 334 (1–3) (2005) 259–274.
- [9] Banerjee, And A. Pathak 2009. An analysis of reversible multiplier circuits. <arXiv:0907.3357>, 1–10.
- [10] T. Toffoli, Reversible Computing, Technical Memo MIT/LCS/TM-151, MIT Lab. for Computer Science, 1980 (unpublished).
- [11] E. Fredkin, T. Toffoli, Conservative logic, *Int. J. Theor. Phys.* 21 (1982) 219–253.
- [12] A. Peres, Reversible logic and quantum computers, *Phys. Rev. A* 32 (1985) 3266–3276.
- [13] D. Maslov, Reversible Logic Synthesis, Doctoral Dissertation, The University of New Brunswick, 1980.
- [14] M.S. Islam, et al., Low cost quantum realization of reversible multiplier circuit, *Inf. Technol. J.* 8 (2009) 208.
- [15] M. Shams, M. Haghparast, K. Navi, Novel reversible multiplier circuit in nanotechnology, *World Appl. Sci. J.* 3 (5) (2008) 806–810.
- [16] R.P. Feynman, Quantum mechanical computers, *Opt. News* 11 (1985) 11–20.
- [17] R. Zhou, Y. Shi, J. Cao, H. Wang, Comment on “design of a novel reversible multiplier circuit using HNG gate in nanotechnology”, *World Appl. Sci. J.* 10 (2) (2010) 161–165.
- [18] F. Naderpour, A. Vafaei, Reversible multipliers: decreasing the depth of the circuit, in: *Proceedings of the fifth International Conference on Electrical and Computer Engineering*, Dhaka, Bangladesh, 2008, pp. 306–310.
- [19] H. Thapliyal, M.B. Srinivas, Novel reversible multiplier architecture using reversible TSG gate, in: *Proceedings of the IEEE International Conference on Computer Systems and Applications*, 2006, pp. 100–103.
- [20] B. Parhami, Fault tolerant reversible circuits, in: *Proceedings of the 40th Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, October 2006, pp. 1726–1729.
- [21] D. Srivastava, S.N. Atluri, Computational nanotechnology: a current perspective, *Comput. Modeling Eng. Sci.* 3 (5) (2002) 531–538.

- [22] M. Haghparast, K. Navi, A novel reversible full adder circuit for nanotechnology based systems, *J. Appl. Sci.* 7 (2007) 3995–4000.
- [23] A. Barenco, C.H. Bennett, R. Cleve, D.P. Divincenzo, N. Margolus, P. Shor, T. Sleator, J.A. Smolin, H. Weinfurter, Elementary gates for quantum computation, *Phys. Rev. A* 52 (5) (1995) 3457–3467.
- [24] H. Thapliyal, M.B. Srinivas, Novel reversible TSG gate and its application for designing reversible carry look ahead adder and other adder architectures, in: *Proceedings of the 10th Asia-Pacific Computer Systems Architecture Conference (ACSAC)*, Lecture Notes of Computer Science, vol. 3740, Springer-Verlag, 2005, pp. 775–786.
- [25] M. Haghparast, K. Navi, A novel reversible bcd adder for nanotechnology based systems, *Am. J. Appl. Sci.* 5 (3) (2008) 282–288.
- [26] M. Haghparast, M. Mohammadi, K. Navi, M. Eshghi, Optimized reversible multiplier circuit, *J. Circuits Syst. Comput.* 18 (2) (2009) 311–323.
- [27] D. Maslov, G.W. Dueck, Reversible cascades with minimal garbage, *IEEE Trans. CAD Integr. Circuits Syst.* 23 (11) (2004) 1497–1509.
- [28] M.S. Islam, M.M. Rahman, Z. Begum, M.Z. Hafiz, Fault tolerant variable block carry skip logic (vbcscl) using parity preserving reversible gates, *Int. J. Comput. Elect. Eng.* 3 (1) (2011) 1793–8163.
- [29] J. Cervera, S. Mafe, Multivalued and reversible logic gates implemented with metallic nanoparticles and organic ligands, *Chem. Phys. Chem.* 11 (2010) 1654–1658.
- [30] Y.V. Rentergem, A. DE VOS, Optimal design of a reversible full adder, *Int. J. Unconv. Comput.* 1 (2005) 339–355.
- [31] E. Pour Ali Akbar, M. Haghparast, K. Navi, Novel design of a fast reversible Wallace sign multiplier circuit in nanotechnology, *Microelectr. J.* (2011). doi:10.1016/j.mejo.2011.05.007.