J Supercomput (2017) 73:4843–4867
DOI 10.1007/s11227-017-2057-z

Novel parity-preserving reversible logic array multipliers

Mojtaba Valinataj[1]

Published online: 25 April 2017

Abstract Reversible logic as a new promising design domain can be used for DNA computations, nanocomputing, and especially constructing quantum computers. However, the vulnerability to different external effects may lead to deviation from producing correct results. The multiplication is one of the most important operations because of its huge usage in different computing systems. Thus, in this paper, some novel reversible logic array multipliers are proposed with error detection capability through the usage of parity-preserving gates. By utilizing the new arrangements of existing reversible gates, some new circuits are presented for partial product generation and multi-operand addition required in array multipliers which results in two unsigned and three signed parity-preserving array multipliers. The experimental results show that the best of signed and unsigned proposed multipliers have the lowest values among the existing designs regarding the main reversible logic criteria including quantum cost, gate count, constant inputs, and garbage outputs. For 4 × 4 multipliers, the proposed designs achieve up to 28 and 46% reduction in the quantum cost and gate count, respectively, compared to the existing designs. Moreover, the proposed unsigned multipliers can reach up to 58% gate count reduction in 16 × 16 multipliers.

Keywords Array multipliers · Baugh–Wooley method · Error detection · Reversible logic · Parity-preserving gates · Quantum computers

✉ Mojtaba Valinataj
m.valinataj@nit.ac.ir

[1] School of Electrical and Computer Engineering, Babol Noshirvani University of Technology, Shariati Ave., Babol 47148-71167, Iran

1 Introduction

In recent years, the high power consumption of VLSI circuits has decelerated feature size reduction of nanometer-scale technologies as was expected from the Moore's law. Therefore, some researches have been conducted to examine novel design areas in order to resolve the new challenges. The reversible logic design paradigm is one of the candidates to overcome the high power dissipation due to the fact that there is no information loss in these circuits. This is based on the fact that one bit information loss results in *kTln2* joules of energy dissipation in which k is the Boltzmann's constant and T is the absolute temperature at which the computation is performed [1]. Therefore, opposed to ordinary or irreversible logic circuits in which information loss is common, the circuits comprising only reversible logic gates do not dissipate this type of energy [2] as the internal power consumption. As a result, reversible logic circuits are worthy to be investigated despite having some physical and implementation difficulties.

Each gate or circuit requires having a one-to-one mapping between its input vector and output vector to be accounted as reversible. In this manner, the number of outputs is equal to the number of inputs. In addition, the input vector can be recovered from the output vector, which means no information is lost in these gates or circuits. Regarding the main characteristic of reversible logic circuits, these circuits can be thought for using in different applications such as DNA computations, nanocomputing, optical computing, quantum computing, and low-power circuits.

Different external noises and environmental effects can result in a fault and cause a reversible circuit to deviate from producing intended outputs. This way, the information is also lost because the input vector will not be recovered from the output vector, anymore. Therefore, similar to irreversible circuits, the fault-tolerance capability at least in the form of fault or error detection should be considered in reversible circuits. A cost-effective approach to detect errors in reversible circuits is the use of parity-preserving gates. This approach is based on the parity-based coding which is a well-known and low-cost method to detect errors in irreversible circuits. A gate having this characteristic is called a parity-preserving reversible gate. In this paper, this characteristic is the main property which will be focused on. However, it should be considered that the implementation of reversible circuits is more complicated in comparison with irreversible circuits because two simple concepts including fan-out and feedback are not allowed in reversible logic [3].

One of the most important arithmetic operations in different computing systems is the multiplication. Among different types of multipliers, the array multipliers have received more attention as the fast multipliers. Therefore, in reversible logic design paradigm, array multipliers should be considered especially respecting fault-tolerance and error detection capabilities. Until now, different types of reversible multipliers have been designed such as [4–13]. Even though many of these designs such as [4,6–8,10,12] are related to array multipliers, in most of them the parity-preserving gates required for error detection capability have not been used. Therefore, in this paper, some novel parity-preserving reversible logic array multipliers are proposed with the emphasis on requiring lower costs especially the quantum cost compared to the few existing parity-preserving designs. In this manner, respecting the main parts of array multipliers, the new designs for partial product generation (PPG) and multi-operand addition (MOA) are presented by exploiting better and newer parity-preserving gates as well as some new arrangements of the existing gates. The proposed multipliers comprise both signed and unsigned multipliers to be used in different applications. Based on the results analysis, the proposed array multipliers contain more optimized design criteria compared to previous parity-preserving reversible array multipliers especially regarding gate count and quantum cost.

The rest of the paper is organized as follows. In Sect. 2, some basic concepts and definitions as well as the parity-preserving reversible gates are described. In Sect. 3 the related works are discussed. In Sect. 4 the new designs for the PPG part, and in Sect. 5 the new designs for the MOA part are proposed. Constructing the new signed and unsigned parity-preserving array multipliers by combining the proposed PPGs and MOAs are shown and evaluated in Sect. 6. Finally, some conclusions are drawn in Sect. 7.

2 Preliminaries

2.1 Basic concepts and definitions

A reversible gate or circuit is an $n \times n$ circuit so that for any n-tuple input vector, a unique n-tuple output vector will appear at the circuit's output. Due to the fact that the input vector can be retrieved by the output vector, as well, we can write $I_v \leftrightarrow O_v$ in which $I_v = (I_0, I_1, \ldots, I_{n-1})$ and $O_v = (O_0, O_1, \ldots, O_{n-1})$ are the input and output vectors, respectively.

A parity-preserving reversible gate is a gate in which the parity of the inputs is equal to the parity of the outputs according to the following equation in which represents for the XOR operation:

$$I_0 \bigoplus I_1 \bigoplus \cdots \bigoplus I_{n-1} = O_0 \bigoplus O_1 \bigoplus \cdots \bigoplus O_{n-1} \tag{1}$$

The parity-preserving characteristic for a gate makes possible all single-error detection and some of multiple-error detection at its outputs. It is worth mentioning that a reversible circuit containing only the parity-preserving gates has itself the parity-preserving property. Therefore, if a reversible circuit with error detection capability is intended, it should only include the parity-preserving gates.

In a reversible gate or circuit, the constant inputs are the inputs whose values do not change in a gate and are maintained at either 0 or 1 in order to perform the intended functions. These inputs are also added to a gate to make it reversible [14]. In addition, the outputs that would not be used in the subsequent computations are called the garbage outputs. In other words, the garbage outputs are needed just to maintain the circuit's reversibility or to make it parity-preserving [15,16]. Another parameter considered in reversible circuits is the hardware complexity or total logical calculation which is the number of AND, XOR, and NOT operations, separately, appeared in the output expressions. In other words, the hardware complexity shows the computational complexity of a reversible circuit that can be important in some types of implementations. This way, if α, β, and γ are the representatives for XOR,

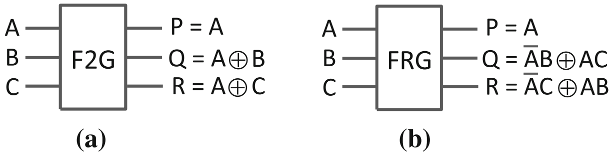


Fig. 1 Block diagrams of **a** double Feynman gate, and **b** Fredkin gate

AND, and NOT operations in the outputs, respectively, then the hardware complexity can be computed according to the following equation:

$$\text{Hardware complexity} = N(\alpha).\alpha + N(\beta).\beta + N(\gamma).\gamma \qquad (2)$$

In the equation above, $N(^*)$ is the number of *-type operations in the output expressions.

As stated in [17], in calculating the hardware complexity, it will be better and more precise if the common operations in the output expressions would be accounted once. Therefore, in this paper, the calculation approach presented in [17] is used.

The quantum cost (QC) is the most important parameter in designing the reversible circuits. This criterion is defined as the number of 1 × 1 and 2 × 2 quantum primitives required for implementing a reversible circuit. The NOT gate is the only 1 × 1 quantum primitive which has the quantum cost of one unit. However, for constructing the reversible gates bigger than 2 × 2, different quantum primitives should be used. In a point of view, the reversible gates can be classified in two general groups, parity-preserving reversible gates and non-parity-preserving reversible gates. As in this paper we are only dealing with the parity-preserving circuits, the main parity-preserving gates are introduced in the following.

2.2 Parity-preserving reversible gates

Many of the basic reversible gates such Feynman gate (FG) [18], Toffoli gate (TG) [19], and Peres gate (PG) [20] are not parity-preserving. However, in the parity-preserving reversible gates that will be described in the following, the parity of the input vector is equal to the parity of the output vector.

1. Double Feynman gate (F2G) [21] as a parity-preserving 3 × 3 reversible gate with the quantum cost of two is shown in Fig. 1a. The hardware complexity of this gate is equal to 2α. This gate can be used as a fan-out generator in reversible circuit synthesis.
2. Fredkin gate (FRG) [22] (Fig. 1b) as the oldest parity-preserving reversible gate with the quantum cost of five has the hardware complexity equal to $2\alpha + 4\beta + 1\gamma$ due to the fact that there exist two distinct XOR operations, four distinct AND operations, and only a distinct NOT operation in its output expressions. This gate is a universal gate that means all logic operations or reversible logic circuits can be implemented only by using this type of gates.

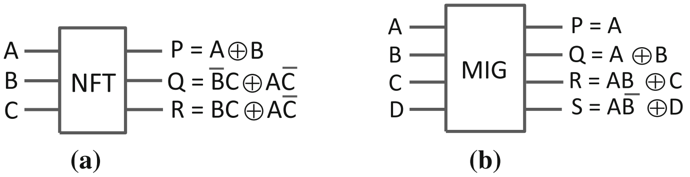

A
B
C
NFT
$P = A \oplus B$
$Q = \overline{B}C \oplus A\overline{C}$
$R = BC \oplus A\overline{C}$
(a)
A
B
C
D
MIG
$P = A$
$Q = A \oplus B$
$R = AB \oplus C$
$S = A\overline{B} \oplus D$
(b)

Fig. 2 Block diagrams of **a** new fault-tolerant gate, and **b** modified Islam gate

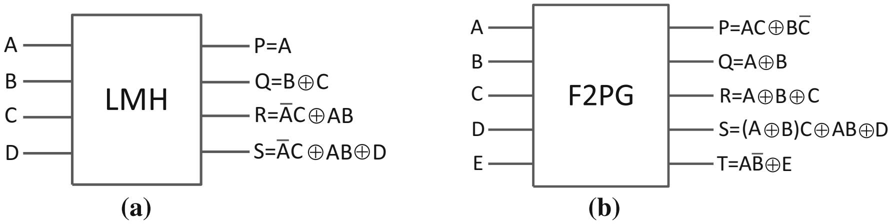

A
B
C
D
LMH
$P=A$
$Q=B \oplus C$
$R=\overline{A}C \oplus AB$
$S=\overline{A}C \oplus AB \oplus D$
(a)
A
B
C
D
E
F2PG
$P=AC \oplus B\overline{C}$
$Q=A \oplus B$
$R=A \oplus B \oplus C$
$S=(A \oplus B)C \oplus AB \oplus D$
$T=A\overline{B} \oplus E$
(b)

Fig. 3 Block diagram of **a** LMH gate, and **b** F2PG

3. New fault-tolerant gate (NFT) [23] as another parity-preserving reversible gate with the quantum cost of five has the hardware complexity equal to $3\alpha + 3\beta + 2\gamma$ according to its outputs shown in Fig. 2a. Similar to FRG, this gate is a universal gate.
4. Modified Islam gate (MIG) [24] (Fig. 2b) is a 4 × 4 parity-preserving reversible gate with the quantum cost of 7 and the hardware complexity equal to $3\alpha+2\beta+1\gamma$. This gate is a universal gate, as well. In addition, this gate can be used as a parity-preserving half adder when its C and D inputs are set to zero. In this case, the output sum and carry are produced on Q and R outputs, respectively.
5. LMH (Lafifa–Mushfiq–Hafiz) [25] shown in Fig. 3a is a 4 × 4 parity-preserving reversible gate with the quantum cost of six and the hardware complexity equal to $3\alpha + 2\beta + 1\gamma$. The obtained hardware complexity is based on the fact that the common or the same operations in the outputs are accounted once according to the approach presented in [17]. Thus, since two XOR operations in the output expressions operate on the same operands (in R and S outputs shown in Fig. 3a), this gate includes three distinct XOR operations which results in 3α instead of 4α. In addition, two same operations $\overline{A}C$ and two same AB operations exist in the output expressions which result in a simpler term 2β instead of 4β. Finally, a distinct NOT operation ($\overline{A}$) results in 1γ.
6. F2PG [8] shown in Fig. 3b is a 5 × 5 parity-preserving reversible gate which has the quantum cost of 14. The hardware complexity of this gate is equals to $6\alpha + 5\beta + 2\gamma$. This gate can be used as a parity-preserving full adder when the D and E inputs are set to zero. In this case, the output sum and carry will be equal to $Sum = A \bigoplus B \bigoplus C$ and $Cout = (A \bigoplus B)C \bigoplus AB$ that are produced on the R and S outputs, respectively.
7. ZPLG [26] shown in Fig. 4a is another 5 × 5 parity-preserving reversible gate with the quantum cost of eight and its hardware complexity is equal to $8\alpha + 3\beta + 1\gamma$. Similar to F2PG, this gate can be used as a parity-preserving full adder when the

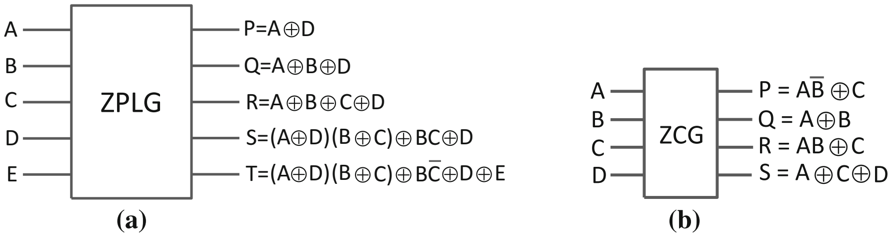

A
B
C
D
E
ZPLG
$P=A\oplus D$
$Q=A\oplus B\oplus D$
$R=A\oplus B\oplus C\oplus D$
$S=(A\oplus D)(B\oplus C)\oplus BC\oplus D$
$T=(A\oplus D)(B\oplus C)\oplus B\overline{C}\oplus D\oplus E$
(a)
A
B
C
D
ZCG
$P = A\overline{B} \oplus C$
$Q = A\oplus B$
$R = AB \oplus C$
$S = A \oplus C\oplus D$
(b)

Fig. 4 Block diagrams of **a** ZPLG, and **b** ZCG

D and E inputs are set to zero. In this case, the output sum and carry are produced on the R and S outputs, respectively. In addition, this gate produces the full adder with minimum quantum cost.

8. ZCG [26] shown in Fig. 4b is a 4 × 4 parity-preserving reversible gate with the quantum cost of six. The hardware complexity of this gate is equal to $5\alpha+2\beta+1\gamma$. Similar to MIG, this gate can be used as a parity-preserving half adder when its C and D inputs are set to zero. In addition, this gate produces the minimum cost half adder.

3 Related works

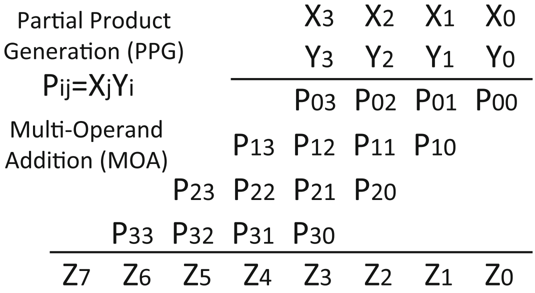
3.1 Parity-preserving reversible full adders

All types of multipliers, i.e., serial, parallel, and array multipliers, in some manner require the addition operation. This operation is usually performed by using full adders and half adders. Although there are many designs for non-parity-preserving adders such as [15,27,28], only the parity-preserving adders can be helpful for the parity-preserving multipliers. There exist some parity-preserving gates that can perform the operation of a parity-preserving full adder (such as F2PG [8], LCG [17] and ZPLG [26]) or half adder (MIG [24] and ZCG [26]) after setting some of their inputs to zero as the constant inputs. However, a full adder can be constructed by connecting two half adders, as well. In addition, a parity-preserving full adder may be constructed by using a few parity-preserving gates similar to SNFA (single NFT full adder) [29] which includes three F2Gs and a NFT gate. This gate has the quantum cost of 11 more than that of LCG and ZPLG but less than that of F2PG, and its hardware complexity is equal to $9\alpha + 3\beta + 2\gamma$.

3.2 Parity-preserving reversible multipliers

Due to the fact that the multiplication is a vital operation in most of processing system, many studies have been performed to design optimal multipliers including reversible designs. The multipliers are designed in two manners, serial or parallel in which the array multipliers can be considered as the most important subgroup of parallel multipliers. These different types of multipliers can be utilized according to different requirements of the variety of applications. Therefore, when a low-cost design is very

Fig. 5 A 4 × 4 unsigned multiplier that can be implemented as an array multiplier with two parts



				X3	X2	X1	X0
				Y3	Y2	Y1	Y0
				P03	P02	P01	P00
			P13	P12	P11	P10	
		P23	P22	P21	P20		
	P33	P32	P31	P30			
Z7	Z6	Z5	Z4	Z3	Z2	Z1	Z0

important, serial multipliers are better because of having a lower cost. On the other hand, if a high-speed design is intended, array or parallel multipliers are better because they have more speed.

The first parity-preserving serial multipliers are proposed in [9] based on the well-known Booth's algorithm and its modified version called Keshuv or K-algorithm for multiplying signed numbers. Even though these multipliers require low costs, their delay is high because of their nature and base algorithms. One of the popular parallel multiplier architectures is array multiplier. As stated before, the array multipliers include two parts, partial product generation (PPG) and multi-operand addition (MOA) as shown in Fig. 5. In the PPG part only partial products are produced by a simple parallel circuit, and in the MOA part the produced partial products will be added together. Despite the fact that various reversible array multipliers exist in the literature, few designs are also parity-preserving. The first parity-preserving signed array multiplier is proposed in [8] based on the Baugh–Wooley method [30]. In this design, the Wallace tree structure is used for the MOA part. According to [8], a 5 × 5 signed multiplier requires 57 reversible gates with the total quantum cost of 401 in which the parity-preserving gates including F2G, FRG, MIG, NFT, and F2PG are utilized. The undesirable property of this design is that it cannot simply be extended for larger designs. In fact, the MOA part should be designed and optimized for each multiplier size.

In [7] a parity-preserving unsigned array multiplier is proposed utilizing F2Gs and FRGs to implement the PPG part, and only MIGs to construct half adders and full adders of MOA part. This multiplier requires a quantum cost of 244 for a 4 × 4 multiplier. In [25] another unsigned parity-preserving array multiplier is presented that in comparison with [7] reduces the required quantum cost of a 4 × 4 multiplier to 205. This design utilizes FRG and LMH to implement the PPG part according to Fig. 6, and incorporates MIG and SNFA to construct half adders and full adders, respectively, for the MOA part. In Fig. 6, vectors x and y are 4-bit input operands, and G at the output of the gates stands for garbage outputs.

4 Proposed partial product generation circuits

The first part or stage of an array multiplier is the PPG. Normally, after finishing the PPG, the next stage (MOA) in which the partial products should be added can

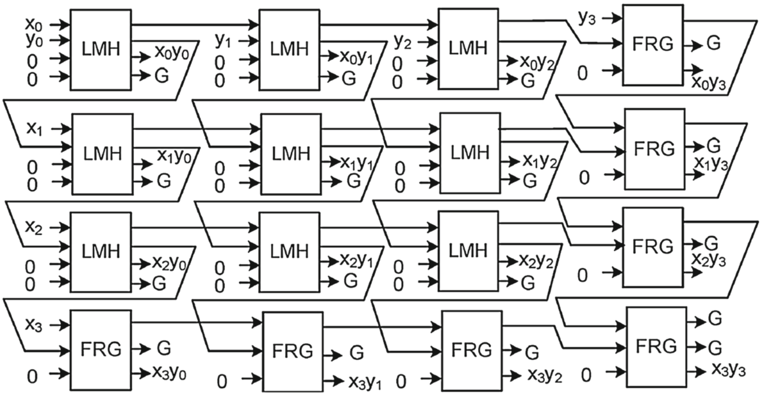


Fig. 6 Partial product generation for the 4 × 4 unsigned array multiplier [25]

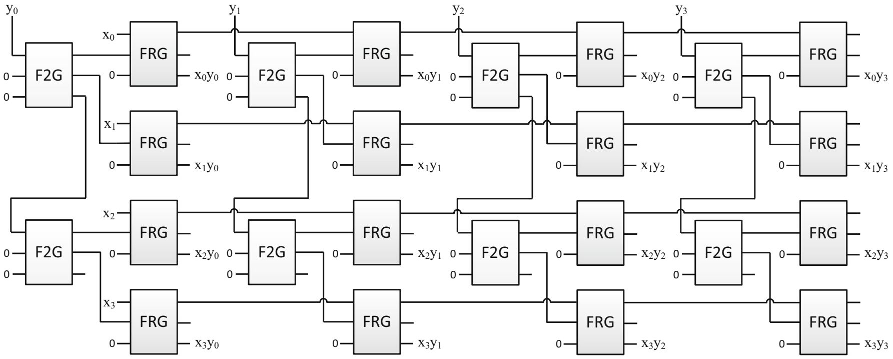


Fig. 7 First proposed partial product generation for the 4 × 4 array multiplier

be started. In an irreversible $n \times n$ array multiplier, n^2 AND gates are required to produce all P_{ij}s that are equal to $x_j y_i$ in which x and y are n-bit input operands, and i and j are the indices from 0 to $n - 1$. Therefore, in the 4 × 4 reversible counterpart, reversible gates should produce all $P_{ij}s$ according to Fig. 5 despite the fact that there is not any separate AND gate in reversible logic. Thus, in the first proposed parity-preserving partial product generation circuit depicted in Fig. 7, FRGs are adjusted for AND operations. In this figure, F2Gs are only utilized for fan-out production required in this circuit. This circuit, in spite of having simple structure, had not been proposed in the literature. This PPG which includes 16 FRGs and 8 F2Gs for a 4 × 4 array multiplier has the quantum cost of 96.

One of the benefits of array multipliers is that they can simply be extended for larger designs based on the smaller designs. Thus, the number of different required gates and the quantum cost of the first proposed PPG can be computed by using Eqs. (3) and (4), respectively, for n-bit operands ($n \geq 2$) to be used in $n \times n$ array multipliers.

Fig. 8 4 × 4 signed multiplication based on the first Baugh–Wooley method

$$\begin{array}{cccccccc}
 & & & & x_3 & x_2 & x_1 & x_0 \\
 & & & & y_3 & y_2 & y_1 & y_0 \\
\cline{4-8}
 & & & & x_3\overline{y}_0 & x_2y_0 & x_1y_0 & x_0y_0 \\
 & & & x_3\overline{y}_1 & x_2y_1 & x_1y_1 & x_0y_1 & \\
 & & x_3\overline{y}_2 & x_2y_2 & x_1y_2 & x_0y_2 & & \\
 & x_3y_3 & \overline{x}_2y_3 & \overline{x}_1y_3 & \overline{x}_0y_3 & & & \\
 & \overline{x}_3 & 0 & 0 & x_3 & & & \\
1 & \overline{y}_3 & 0 & 0 & y_3 & & & \\
\hline
P_7 & P_6 & P_5 & P_4 & P_3 & P_2 & P_1 & P_0
\end{array}$$

Fig. 9 4 × 4 signed multiplication based on the second Baugh–Wooley method

$$\begin{array}{cccccccc}
 & & & & x_3 & x_2 & x_1 & x_0 \\
 & & & & y_3 & y_2 & y_1 & y_0 \\
\cline{4-8}
 & & & 1 & \overline{x_3y_0} & x_2y_0 & x_1y_0 & x_0y_0 \\
 & & & \overline{x_3y_1} & x_2y_1 & x_1y_1 & x_0y_1 & \\
 & & \overline{x_3y_2} & x_2y_2 & x_1y_2 & x_0y_2 & & \\
1 & x_3y_3 & \overline{x_2y_3} & \overline{x_1y_3} & \overline{x_0y_3} & & & \\
\hline
P_7 & P_6 & P_5 & P_4 & P_3 & P_2 & P_1 & P_0
\end{array}$$

$$\text{Required gates of 1st } (n \times n) \text{ PPG} = n^2 \times \text{FRG} + n \times \lfloor n/2 \rfloor \times \text{F2G} \tag{3}$$

$$\begin{aligned}\text{QC of 1st } (n \times n) \text{ PPG} &= n^2 \times \text{QC}_{\text{FRG}} + n \times \lfloor n/2 \rfloor \times \text{QC}_{\text{F2G}} \\ &= 5n^2 + 2n \times \lfloor n/2 \rfloor \end{aligned} \tag{4}$$

Signed array multipliers may require different partial products. The best signed array multipliers are based on the Baugh–Wooley method [30]. In fact, based on the Baugh–Wooley method, two different partial product arrangements can be used according to Figs. 8 and 9. In this paper, we call them BW1 and BW2, respectively, in which the second arrangement (Fig. 9) has lower costs. On the other hand, the first PPG proposed in Fig. 7 cannot be used for BW1 even though it can be used for BW2. Therefore, proposing the PPGs in which all the items shown in Figs. 8 and 9 required for the MOA part are produced can be very advantageous because otherwise more gates should be used to produce the inverted operands, separately.

In Fig. 8, some terms include an inverted operand. These terms can be produced by properly adjusting the inputs of FRGs and changing the bit locations of both input operands compared to Fig. 7. This way, the second proposed parity-preserving PPG circuit for a 4 × 4 array multiplier is shown in Fig. 10. This circuit which is useful for the BW1 method utilizes two more F2Gs compared to Fig. 7 to produce x_3, y_3, and the inverted x_3 and y_3 (bold items in Fig. 10) required in Fig. 8. As a result, the second proposed PPG circuit has the quantum cost of 100 which is only four units higher than that of the first proposed PPG circuit, and on the other hand, it can be used in a signed array multiplier without any modification.

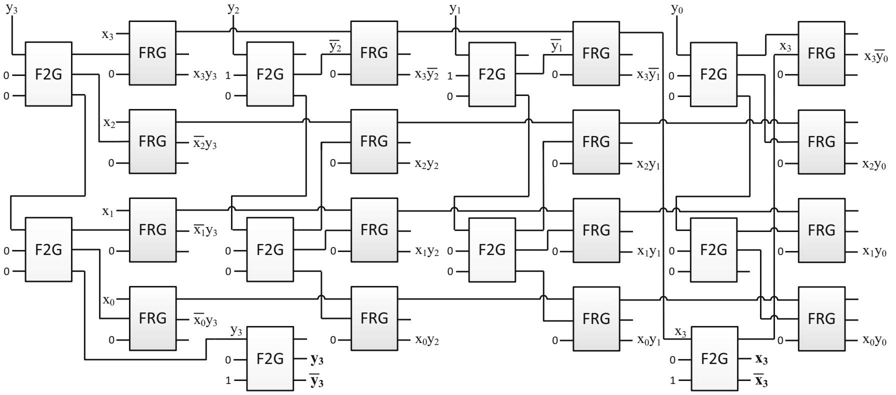


Fig. 10 Second proposed partial product generation for 4 × 4 array multiplier based on the first Baugh–Wooley method

By more investigation, it is obtained that the LMH gate introduced in [25] can be used instead of an FRG and an F2G, while it can produce the terms required in the BW1 method. In fact, an LMH gate with the quantum cost of six which is one unit more than that of FRG, can be used to propagate two input signals compared to one input in the FRG, and thus helps to eliminate some F2Gs by this fan-out production. Therefore, the third proposed parity-preserving PPG circuit for a 4 × 4 array multiplier is obtained according to Fig. 11 and can be used in the BW1 method-based signed multiplier with the quantum cost lower than that of the first and second proposed PPG circuits. In Fig. 11, some FRGs are used in the locations in which only one signal propagation is required, and thus there is no need to LMH gates. In fact, three FRGs in the right-most locations (except the lower-right FRG) propagate y_0 input signal, and three other FRGs in the last row propagate x_0 input signal to their right-hand gates. In addition, two F2Gs are required similar to Fig. 10 to produce x_3, y_3, and the inverted x_3 and y_3 (bold items in Fig. 11) required for the BW1 method.

To extend the size of second and third proposed PPG circuits to be used in larger signed multipliers, the following equations can be used to predict the number of different gates and total quantum cost for n-bit operands:

$$\begin{aligned}\text{Required gates of 2nd } (n \times n) \text{ PPG} &= n^2 \times \text{FRG} + (n \times \lfloor n/2 \rfloor \\ &\quad + n \bmod 2 + 2) \times \text{F2G} \qquad (5)\\ \text{QC of 2nd } (n \times n) \text{ PPG} &= n^2 \times \text{QC}_{\text{FRG}} + (n \times \lfloor n/2 \rfloor + n \bmod 2 + 2) \\ &\quad \times \text{QC}_{\text{F2G}} = 5n^2 + 2\,(n \times \lfloor n/2 \rfloor \\ &\quad + n \bmod 2) + 4 \qquad (6)\\ \text{Required gates of 3rd } (n \times n) \text{ PPG} &= (n-1)^2 \times \text{LMH} \\ &\quad + (2n-1) \times \text{FRG} + 2 \times \text{F2G} \qquad (7)\\ \text{QC of 3rd } (n \times n) \text{ PPG} &= (n-1)^2 \times \text{QC}_{\text{LMH}} + (2n-1) \times \text{QC}_{\text{FRG}} \\ &\quad +2 \times \text{QC}_{\text{F2G}} = 6n^2 - 2n + 5 \qquad (8)\end{aligned}$$

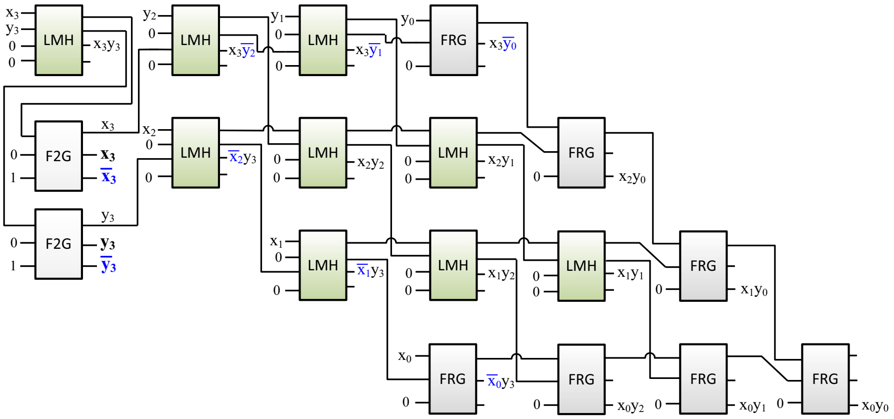


Fig. 11 Third proposed partial product generation for 4 × 4 array multiplier based on the first Baugh–Wooley method

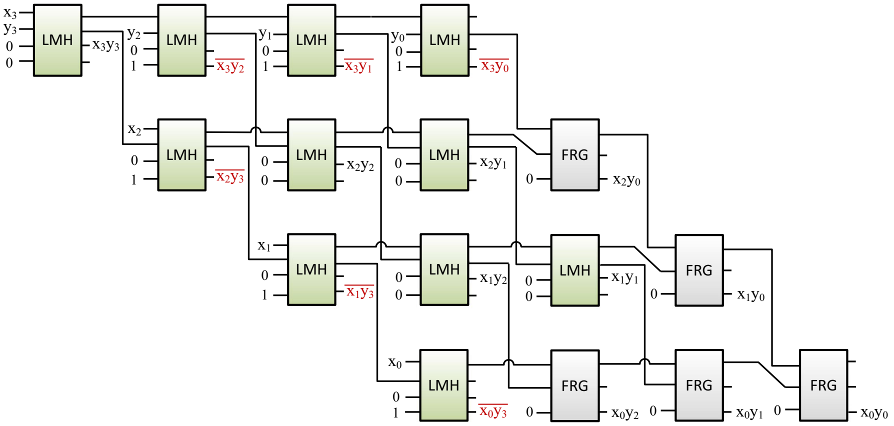


Fig. 12 Fourth proposed partial product generation for 4 × 4 array multiplier based on the second Baugh–Wooley method

According to Fig. 9, some terms of partial products should be inverted to be used in a signed array multiplier based on the BW2 method. The gate with the lowest cost that can produce an inverted product term is LMH. Thus, to produce the required terms shown in Fig. 9, a new arrangement of LMH gates and FRGs with appropriate input adjustments is suggested as the forth proposed parity-preserving PPG circuit for a 4 × 4 array multiplier as shown in Fig. 12. In this figure, some LMH gates are adjusted with the constant inputs of zero and one for their third and fourth inputs to produce the inverted product terms, and other LMH gates are adjusted with two zero constant inputs to produce normal product terms. Moreover, FRGs are used as far as possible to produce the remaining product terms in the locations that there is no need to LMH gates. The quantum cost of this PPG circuit is equal to 91 and is the lowest among four proposed PPG circuits.

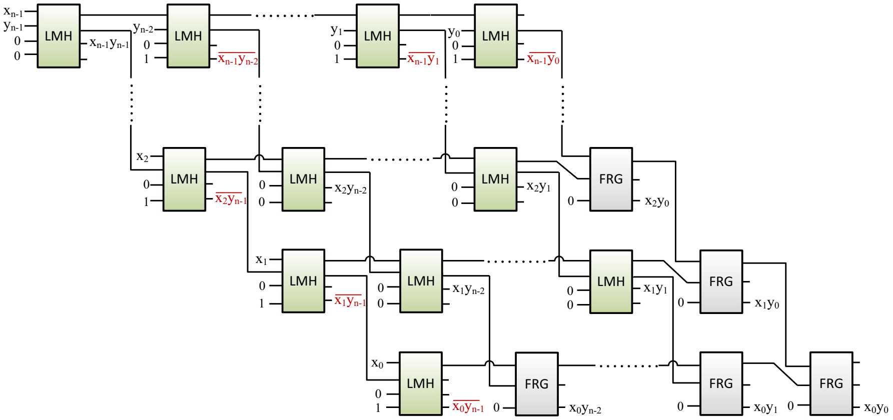


Fig. 13 Generalized structure of fourth proposed partial product generation for $n \times n$ array multiplier

To extend the size of fourth proposed PPG circuit to be used in larger signed multipliers, Eqs. (9) and (10) are used to predict the number of different required gates and total quantum cost for n-bit operands. In addition, to demonstrate the generalized structure of the PPG circuits for n-bit operands, it is shown in Fig. 13 for the best proposed PPG in this paper to be used in $n \times n$ signed array multipliers.

$$\text{Required gates of 4th } (n \times n) \text{ PPG} = ((n-1)^2 + 2) \times \text{LMH} + (2n-3) \times \text{FRG} \tag{9}$$

$$\text{QC of 4th } (n \times n) \text{ PPG} = ((n-1)^2 + 2) \times \text{QC}_{\text{LMH}} + (2n-3) \times \text{QC}_{\text{FRG}} = 6n^2 - 2n + 3 \tag{10}$$

To compare the proposed PPG circuits in this paper with the previous designs, Table 1 illustrates different characteristics and reversible logic criteria for all PPG designs. It is worth mentioning that the calculation of values for the criteria in each circuit is straightforward. In fact, the number of required gates, constant inputs and garbage outputs are obtained based on the figure drawn for each proposed circuit. The number of constant inputs in each figure is the number of gates' inputs whose values are either '0' or '1'. In addition, the number of garbage outputs is the number of gates' outputs that are not connected to the other gates or are not used as the outputs of the circuit. The values for other criteria are obtained by summing the values for all the gates. In Table 1, the first three rows are mainly for unsigned multipliers, and the remaining rows are the designs based on the first or second Baugh–Wooley method for signed multiplication. In addition, the bold items show the best values in each column, separately for signed and unsigned PPG circuits. According to this table, the fourth proposed PPG is the best for the signed multiplication in all criteria except the number of garbage outputs. Moreover, this PPG can be assumed as the best in general because its criteria are very close to that of the unsigned design in [25] while a signed multiplier can be used for unsigned operands and it is not true, reversely.

Table 1 Comparison of PPG parts of different parity-preserving array multipliers

4 × 4 multiplier	Base algorithm	Signed	Gate count	Constant inputs	Garbage outputs	Quantum cost	Hardware complexity
[7]	Simple array	No	28	40	32	104	$56\alpha + 64\beta + 16\gamma$
[25]	Simple array	No	**16**	**25**	**17**	**89**	$\boldsymbol{41\alpha + 46\beta + 16\gamma}$
First proposed PPG (Fig. 7)	Simple array	No	24	32	24	96	$48\alpha + 64\beta + 16\gamma$
[8]	BW2	Yes	26	36	23	100	$58\alpha + 58\beta + 22\gamma$
Second proposed PPG (Fig. 10)	BW1	Yes	26	36	24	100	$52\alpha + 64\beta + 16\gamma$
Third proposed PPG (Fig. 11)	BW1	Yes	19	29	**17**	93	$45\alpha + 46\beta + 16\gamma$
Fourth proposed PPG (Fig. 12)	BW2	Yes	**16**	**27**	19	**91**	$\boldsymbol{43\alpha + 42\beta + 16\gamma}$

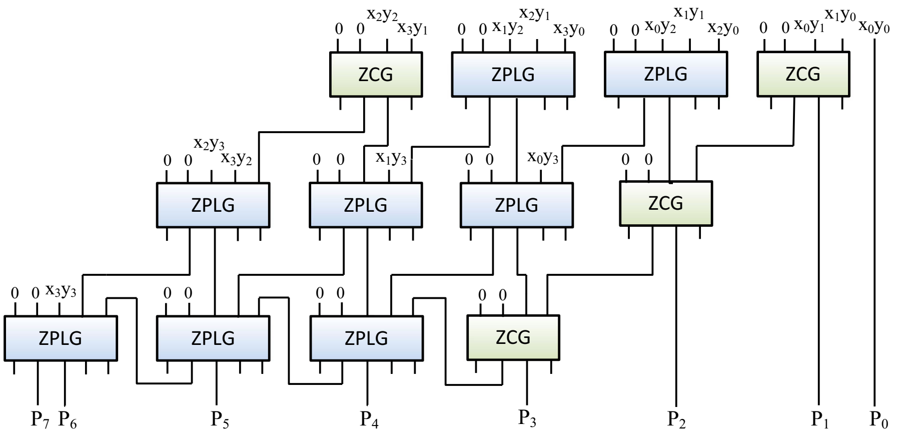


Fig. 14 First proposed multi-operand addition for 4 × 4 unsigned array multiplier

5 Proposed multi-operand addition circuits

The second part of an array multiplier is the multi-operand addition or MOA. Because of the nature of this part which is the addition operation, the main building blocks will be full adders and half adders. As stated before, the full adder and half adder with the lowest quantum cost as the main criterion are ZPLG and ZCG both introduced in [26] with the quantum cost of 8 and 6, respectively. Regarding other criteria, the number of constant inputs and garbage outputs are almost the same in different full adder and half adder designs. In this section similar to previous section, the designs for signed and unsigned multiplications are separately proposed. Thus, the first proposed parity-preserving MOA circuit for a 4 × 4 unsigned array multiplier is shown in Fig. 14. In this figure, all the terms of partial products produced in the PPG part are added based on their weight in the addition process to produce the result of multiplication which is an eight-bit output P. In Fig. 14, when two operands should be added together ZCG is used as the half adder. Moreover, in Fig. 14 the output carries of full adders and half adders are passed to the next column diagonally as much as possible to reduce the overall delay. This circuit has the quantum cost of 88 because of having eight ZPLGs and four ZCGs. To extend the size of first proposed MOA circuit to be used in larger multipliers, Eqs. (11) and (12) can be used to predict the number of different required gates and total quantum cost for n-bit operands.

$$\begin{aligned} \text{Required gates of 1st } (n \times n) \text{ MOA} &= n\,(n-2) \times \text{ZPLG} \\ &\quad + n \times \text{ZCG} \end{aligned} \tag{11}$$

$$\begin{aligned} \text{QC of 1st } (n \times n) \text{ MOA} &= n\,(n-2) \times \text{QC}_{\text{ZPLG}} \\ &\quad + n \times \text{QC}_{\text{ZCG}} = 8n^2 - 10n \end{aligned} \tag{12}$$

For signed multipliers based on the Baugh–Wooley method, different MOAs should be designed for BW1 and BW2 because of different structures of Figs. 8 and 9. In fact,

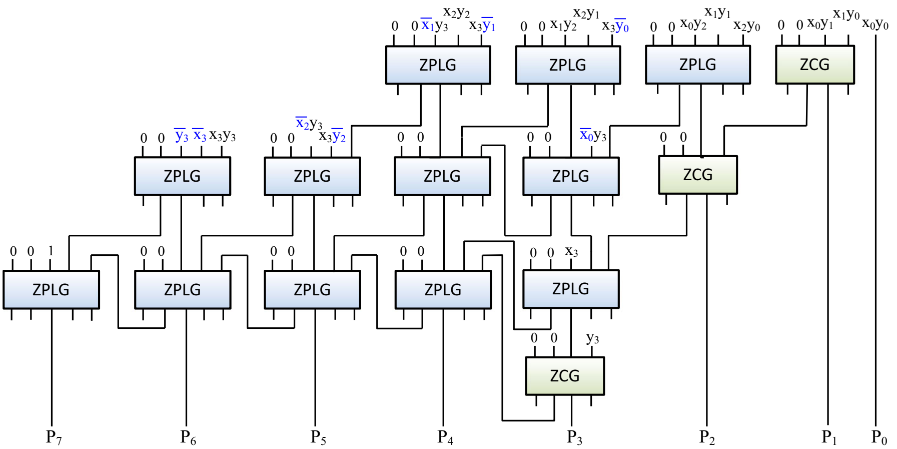


Fig. 15 Second proposed multi-operand addition for 4 × 4 signed array multiplier

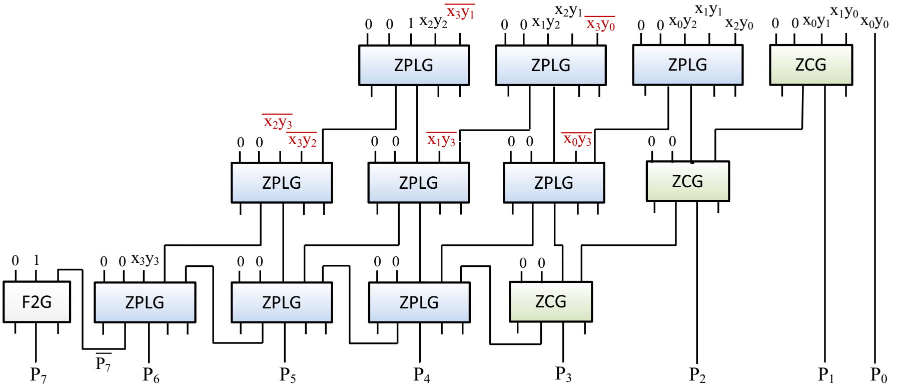


Fig. 16 Third proposed multi-operand addition for 4 × 4 signed array multiplier

the new MOA circuits should add the partial products produced by the PPG circuits for BW1 and BW2. Therefore, the second parity-preserving MOA circuit for a 4 × 4 signed array multiplier is proposed in Fig. 15 which is beneficial for the BW1 method. In addition, the third proposed parity-preserving MOA circuit for a 4 × 4 signed array multiplier is depicted in Fig. 16 which is useful for the BW2 method. In Fig. 16, the only F2G is used to perform an operation equivalent to the addition by one based on the lower-left '1' shown in Fig. 9. In fact, in Fig. 16 a ZCG as a half adder should add the output carry of lower-left ZPLG by one which requires six units more quantum cost. In this special case, this addition operation is equivalent to inverting the output carry of lower-left ZPLG that can be performed by using a F2G after an appropriate adjustment of its inputs which results in a reduction of quantum cost by four.

To extend the size of third proposed MOA circuit to be used in larger signed multipliers, Eqs. (13) and (14) can be used to predict the number of different required gates and total quantum cost for n-bit operands. Moreover, Fig. 17 demonstrates the

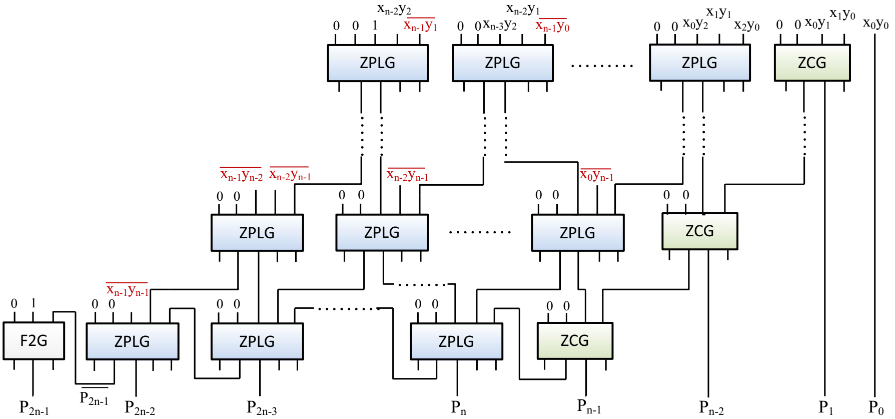


Fig. 17 Generalized structure of third proposed multi-operand addition for $n \times n$ signed array multiplier

generalized structure of this MOA as the best proposed MOA in this paper respecting the quantum cost for signed multiplication of n-bit operands.

$$\text{Required gates of 3rd } (n \times n) \text{ MOA} = (n-1)^2 \times \text{ZPLG} + (n-1) \times \text{ZCG} + 1 \times \text{F2G} \tag{13}$$

$$\text{QC of 3rd } (n \times n) \text{ MOA} = (n-1)^2 \times \text{QC}_{\text{ZPLG}} + (n-1) \times \text{QC}_{\text{ZCG}} + 1 \times \text{QC}_{\text{F2G}} = 8n^2 - 10n + 4 \tag{14}$$

To compare the proposed MOA circuits in this paper with the previous designs, Table 2 illustrates different characteristics and reversible logic criteria for all MOA designs. In this table, the first three rows are only applicable to unsigned multipliers, and the remaining three rows are the designs dedicated for signed multiplications based on the first or second Baugh–Wooley method. In addition, the bold items show the best values in each column, separately for signed and unsigned MOA circuits. However, the best values have not been bold in some cases where all the appropriate designs have the same value or the best value cannot be selected. According to this table, the first proposed MOA has the lowest quantum cost among the designs applicable to unsigned multiplication, and the third proposed MOA has the lowest quantum cost among the designs beneficial for signed multiplication.

6 Results and discussion

In this section, the proposed parity-preserving signed and unsigned array multipliers will be illustrated by combining the appropriate proposed PPG circuits and MOA circuits. After constructing different multipliers, some comparisons will be performed between the proposed multipliers and their previous signed or unsigned counterparts. In the comparisons, similar to Tables 1 and 2, five main criterions are used including gate count, number of constant inputs, number of garbage outputs, quantum cost, and

Table 2 Comparison of MOA parts of different parity-preserving array multipliers

4 × 4 multiplier	Base algorithm	Signed	Gate count	Constant inputs	Garbage outputs	Quantum cost	Hardware complexity
[7]	Simple array	No	20	24	32	140	$\boldsymbol{60\alpha + 40\beta + 20\gamma}$
[25]	Simple array	No	36	24	32	116	$84\alpha + 32\beta + 20\gamma$
First proposed MOA (Fig. 14)	Simple array	No	**12**	24	32	**88**	$84\alpha + 32\beta + 12\gamma$
[8]	BW2	Yes	**12**	**25**	**33**	147	$63\alpha + 51\beta + 21\gamma$
Second proposed MOA (Fig. 15)	BW1	Yes	15	31	43	114	$111\alpha + 42\beta + 15\gamma$
Third proposed MOA (Fig. 16)	BW2	Yes	13	27	35	**92**	$89\alpha + 33\beta + 12\gamma$

Table 3 Comparison of different parity-preserving unsigned array multipliers

4 × 4 multiplier	Gate count	Constant inputs	Garbage outputs	Quantum cost	Hardware complexity
[7]	48	64	64	244	$116\alpha + 104\beta + 36\gamma$
[25]	52	**49**	**49**	205	$125\alpha + 78\beta + 36\gamma$
First proposed circuit (combination of Figs. 7 and 14)	36	56	56	184	$132\alpha + 96\beta + 28\gamma$
Second proposed circuit (combination of Figs. 6 and 14)	**28**	**49**	**49**	**177**	$125\alpha + 78\beta + 28\gamma$

hardware complexity even though the quantum cost is the most important criterion. In the following, at first unsigned array multipliers and then, signed array multipliers are characterized and evaluated.

6.1 Unsigned array multipliers

Based on the previously proposed PPG and MOA circuits in this paper for unsigned array multiplication, we propose two parity-preserving unsigned array multipliers. For the 4 × 4 multiplication, the first proposed multiplier is constructed by combining the first PPG and the first MOA proposed before depicted in Figs. 7 and 14, respectively. This multiplier has the quantum cost of 184 that is the sum of 96 and 88. The second proposed multiplier is constructed by combining the PPG shown in Fig. 6 from [25] as the best unsigned PPG and the first proposed MOA depicted in Fig. 14. This combination leads to the best unsigned multiplier with the quantum cost of 177 which is the minimum value among all designs. Therefore, two proposed parity-preserving unsigned array multipliers are characterized in Table 3 along with all previous designs that are from [7] and [25]. In this table, the bold items show the best values in each column. According to this table, the second proposed multiplier is the best with respect to all design criteria except the hardware complexity where one of two designs, the design in [7] or the second proposed multiplier in this paper, can be judged as the best. Moreover, to illustrate the precise amounts of improvements attained by two proposed unsigned multipliers, Fig. 18 depicts the percentages of reduction in four criterions for the first and second 4-bit designs compared to the designs proposed in [7] and [25]. Based on this figure, the improvements of the second proposed multiplier in this paper compared to [7] is 41.7, 23.4, 23.4, and 27.5% for the gate count, constant inputs, garbage outputs, and quantum cost, respectively. Moreover, the improvements of the second proposed multiplier compared to [25] as the best of previous designs is 46.2 and 13.7% for the gate count and quantum cost, respectively, while their constant inputs and garbage outputs are equal.

To figure out the proposed multipliers for larger input operands, Table 4 demonstrates the number of different required gates and quantum cost for $n \times n$ multipliers

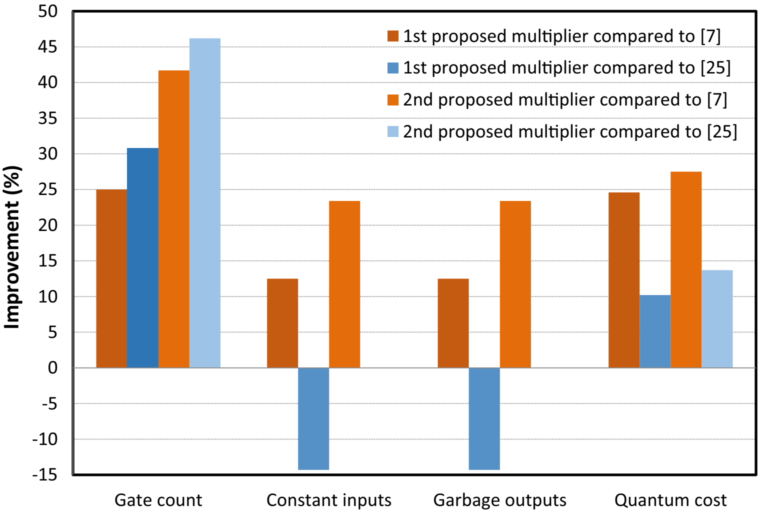


Fig. 18 Improvements of the first and second proposed 4-bit unsigned array multipliers compared to previous designs

as the functions of operands' size n together with the results for two samples 8×8 and 16×16 multipliers. In this table, two proposed unsigned array multipliers are only compared to [25] as the best of previous designs. Furthermore, Fig. 19 depicts the percentages of reduction in the gate count and quantum cost of the proposed unsigned array multipliers compared to [25], for 8×8 and 16×16 multipliers. The important result based on this figure is that the amounts of improvements attained by both proposed multipliers for 16×16 size is higher than that of 8×8 size which itself has a higher improvement compared to the 4×4 multiplier. Therefore, larger multipliers such as 32×32 size benefit more from the proposed designs.

In addition to the possibility for calculating the gate count and quantum cost of larger multipliers according to Table 4, other criteria can be estimated for $n \times n$ multipliers, as well. In other words, it is possible to estimate the number of constant inputs and the number of garbage outputs for $n \times n$ multipliers by considering the number of constant inputs or garbage outputs in each gate type of the PPG and MOA parts of the proposed multipliers, and considering the main formula for the required gates of PPG and MOA circuits, as well. For example, Eqs. (15) to (18) can be used to estimate these criteria for the first and second proposed array multipliers. In addition, the design proposed in [25] follows Eqs. (17) and (18) for its constant inputs and garbage outputs, respectively, which are compatible with the results presented in Table 3.

$$\text{Constant inputs of 1st } (n \times n) \text{ multiplier} = 3n^2 + 2n \times (\lfloor n/2 \rfloor - 1) \tag{15}$$

$$\text{Garbage outputs of 1st } (n \times n) \text{ multiplier} = 4n^2 - 3n + n \times ((n+1) \bmod 2) \tag{16}$$

$$\text{Constant inputs of 2nd } (n \times n) \text{ multiplier} = 4n^2 - 4n + 1 \tag{17}$$

$$\text{Garbage outputs of 2nd } (n \times n) \text{ multiplier} = 4n^2 - 4n + 1 \tag{18}$$

Table 4 Comparison of larger unsigned array multipliers and their general formulae

Multiplier design	General formulae for $n \times n$ multiplier		8×8 multiplier		16 × 16 multiplier	
	Required gates	Quantum cost	Gate count	Quantum cost	Gate count	Quantum cost
[25]	$(n-1)^2 \times \text{LMH} + (2n-1) \times \text{FRG} + n \times \text{MIG} + n(n-2) \times \text{NFT} + 3n(n-2) \times \text{F2G}$	$17n^2 - 17n + 1$	264	953	1168	4081
First proposed circuit (combination of Figs. 7 and 14)	$n^2 \times \text{FRG} + n \times \lfloor n/2 \rfloor \times \text{F2G} + n(n-2) \times \text{ZPLG} + n \times \text{ZCG}$	$13n^2 + 2n \times \lfloor n/2 \rfloor - 10n$	152	816	624	3424
Second proposed circuit (combination of Figs. 6 and 14)	$(n-1)^2 \times \text{LMH} + (2n-1) \times \text{FRG} + n(n-2) \times \text{ZPLG} + n \times \text{ZCG}$	$14n^2 - 12n + 1$	120	801	496	3393

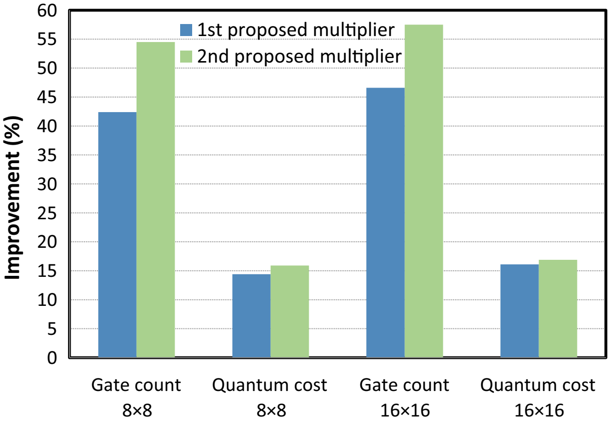


Fig. 19 Improvements of the first and second proposed unsigned array multipliers with different sizes compared to the design in [25]

6.2 Signed array multipliers

The proposed signed array multipliers are based on the previously proposed PPG and MOA circuits in this paper for signed array multiplication. This way, we propose three new parity-preserving signed array multipliers. For the 4 × 4 multiplication, the first proposed signed multiplier as the third proposed multiplier in this paper is constructed by combining the second PPG (Fig. 10) and the second MOA (Fig. 15) proposed before based on the BW1 method. This multiplier has the quantum cost of 214 that is the sum of 100 and 114. The second proposed signed multiplier (the fourth proposed multiplier in this paper) is constructed by combining the third PPG (Fig. 11) and the second MOA (Fig. 15) based on the BW1 method. This multiplier with the quantum cost of 207 is totally better that the first proposed signed multiplier regarding different criteria. The third proposed signed multiplier (the fifth proposed multiplier in this paper) is constructed by combining the fourth PPG (Fig. 12) and the third MOA (Fig. 16) based on the BW2 method. This combination leads to the best signed array multiplier with the quantum cost of 183 which is the minimum value among all signed designs.

Three proposed parity-preserving signed array multipliers are characterized in Table 5 along with the only existing design in [8]. In this table, the bold items show the best values in each column. According to this table, the fifth proposed multiplier is the best with respect to all design criteria except the hardware complexity where one of two designs, the design in [8] or the fifth proposed multiplier in this paper, can be judged as the best. Moreover, Table 6 depicts the best proposed signed array multiplier in this paper compared to the only existing 5 × 5 design from [8] that its MOA part is based on the Wallace tree structure. This table reveals the superiority of the proposed design that uses a simple array for its MOA, in comparison with the design in [8].

To illustrate the precise amounts of improvements attained by the best proposed signed multiplier, Fig. 20 depicts the percentages of reduction in four criterions for

Table 5 Comparison of different parity-preserving signed array multipliers

4 × 4 multiplier	Base algorithm	Gate count	Constant inputs	Garbage outputs	Quantum cost	Hardware complexity
[8]	BW2	38	61	56	247	$121\alpha + 109\beta + 43\gamma$
Third proposed circuit (combination of Figs. 10 and 15)	BW1	41	67	67	214	$163\alpha + 106\beta + 31\gamma$
Fourth proposed circuit (combination of Figs. 11 and 15)	BW1	34	60	60	207	$156\alpha + 88\beta + 31\gamma$
Fifth proposed circuit (combination of Figs. 12 and 16)	BW2	**29**	**54**	**54**	**183**	$132\alpha + 75\beta + 28\gamma$

Table 6 Best proposed signed array multiplier compared to the only existing 5×5 Baugh–Wooley multiplier

5×5 multiplier	Gate count	Constant inputs	Garbage outputs	Quantum cost	Hardware complexity
[8]	57	90	90	401	$190\alpha + 180\beta + 69\gamma$
Fifth proposed circuit (combination of Figs. 12 and 16)	49	86	86	297	$218\alpha + 120\beta + 45\gamma$

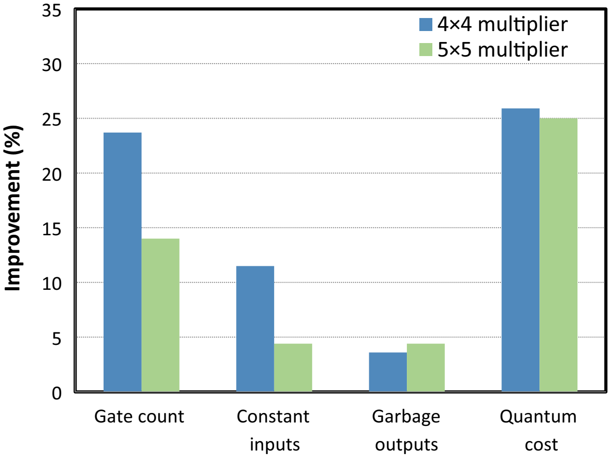


Fig. 20 Improvements in the fifth proposed array multiplier compared to the design in [8] for different sizes

the fifth proposed design compared to the design in [8] for 4 × 4 and 5 × 5 multiplier sizes. Based on this figure, all criteria have some enhancements in the fifth proposed multiplier. The best improvements are obtained for the quantum cost as the most important criterion, that are equal to 25.9 and 25% for 4 × 4 and 5 × 5 multipliers, respectively. Moreover, the improvements in the gate count are equal to 23.7 and 14% for 4 × 4 and 5 × 5 multipliers, respectively.

To figure out the performance of fifth proposed multiplier for larger input operands, Table 7 demonstrates the formulae for the major reversible logic criteria for $n \times n$ multipliers as the functions of operands' size n together with the results for two samples 8 × 8 and 16 × 16 multipliers. Based on this table, the number of constant inputs and the number of garbage outputs will be equal in each specific multiplier size.

7 Conclusion

In this paper, five novel parity-preserving reversible array multipliers were proposed by designing some new partial product generation and multi-operand addition circuits required in array multipliers, for both signed and unsigned multiplications. To attain better designs, the new arrangements of existing parity-preserving reversible

Table 7 Evaluation of the best proposed signed array multiplier with different sizes based on its general formulae

Size of fifth proposed multiplier	Gate count	Constant inputs	Garbage outputs	Quantum cost
$n \times n$	$((n-1)^2+2) \times \text{LMH} + (2n-3) \times \text{FRG} + (n-1)^2 \times \text{ZPLG} + (n-1) \times \text{ZCG} + 1 \times \text{F2G}$	$4n^2-4n+6$	$4n^2-4n+6$	$14n^2-12n+7$
8×8	121	230	230	807
16×16	497	966	966	3399

gates were utilized as well as exploiting newer gates. Therefore, the proposed signed and unsigned array multipliers can be used in different reversible logic applications especially in the speed-critical applications. The proposed signed array multipliers are based on the first and second Baugh–Wooley method. The last proposed signed array multiplier in this paper as the best design has achieved 26% improvement in the quantum cost compared to the best existing design. In addition to the basic 4×4 multipliers, the proposed multipliers have been generalized to $n \times n$ multipliers, and some general formulae were exploited to investigate larger multipliers such as 8×8 and 16×16 designs. The experimental results have shown the superiority of the proposed designs with different sizes compared to the existing designs respecting the main reversible logic criteria especially the quantum cost and gate count.

References

1. Landauer R (1961) Irreversibility and heat generation in the computing process. IBM J Res Dev 5(3):183–191
2. Bennet C (1973) Logical reversibility of computation. IBM J Res Dev 17(6):525–532
3. Perkowski M, Al-Rabadi A, Kerntopf P, Buller A, Chrzanowska-Jeske M, Mishchenko A et al (2001) A general decomposition for reversible logic. In: Proceedings of RM, pp 119–138
4. Zhou R, Shi Y, Wanga H, Cao J (2011) Transistor realization of reversible "ZS" series gates and reversible array multiplier. Microelectron J 42:305–315
5. Pouraliakbar E, Haghparast M, Navi K (2011) Novel design of a fast reversible Wallace sign multiplier circuit in nanotechnology. Microelectron J 42:973–981
6. Moghadam MZ, Navi K (2012) Ultra-area-efficient reversible multiplier. Microelectron J 43:377–385
7. Babazadeh S, Haghparast M (2012) Design of a nanometric fault tolerant reversible multiplier circuit. J Basic Appl Sci Res 2(2):1355–1361
8. Qi X, Chen F (2012) Design of fast fault tolerant reversible signed multiplier. Int J Phys Sci 7(17):2506–2514
9. Bhardwaj K, Deshpande M (2013) K-Algorithm: an improved Booth's recoding for optimal fault-tolerant reversible multiplier. In: 26th International Conference on VLSI Design, pp 362–367
10. Hatkar AP, Hatkar AA, Narkhede NP (2014) ASIC design of reversible multiplier circuit. In: Proceedings of International Conference on Electronic Systems, Signal Processing and Computing Technologies, pp 47–52
11. Kotiyal S, Thapliyal H, Ranganathan N (2014) Circuit for reversible quantum multiplier based on binary tree optimizing Ancilla and Garbage bits. In: Proceedings of 27th International Conference on VLSI Design (VLSID), pp 545–550

12. Moshnyaga VG (2015) Design of minimum complexity reversible multiplier. In: Proceedings of IEEE Region 10 Conference (TENCON), pp 1–4
13. Kotiyal S, Thapliyal H, Ranganathan N (2015) Reversible logic based multiplication computing unit using binary tree data structure. J Supercomput 71:2668–2693
14. Maslov D, Dueck GW (2004) Reversible cascades with minimal garbage. IEEE Trans CAD Integr Circuits Syst 23(11):1497–1509
15. Biswas AK, Hasan MM, Chowdhury AR, Babu HMH (2008) Efficient approaches for designing reversible binary coded decimal adders. Microelectron J 39:1693–1703
16. Maslov D (1980) Reversible logic synthesis. Doctoral Dissertation, University of New Brunswick
17. Valinataj M, Mirshekar M, Jazayeri H (2016) Novel low-cost and fault-tolerant reversible logic adders. Comput Electr Eng 53:56–72
18. Feynman R (1985) Quantum mechanical computers. Opt News 11:11–20
19. Toffoli T (1980) Reversible computing, Tech. memo MIT/LCS/TM-151, MIT Lab. for Computer Science
20. Peres A (1985) Reversible logic and quantum computers. Phys Rev 32:3266–3276
21. Parhami B (2006) Fault-tolerant reversible circuits. In: 40th Asilomar Conference on Signals, Systems and Computers, (ACSSC), pp 1726–1729
22. Fredkin E, Toffoli T (1982) Conservative logic. Int J Theor Phys 21:219–253
23. Hagparast M, Navi K (2008) A novel fault tolerant reversible gate for nanotechnology based system. Am J Appl Sci 5(5):519–523
24. Islam MS, Rahman MM, Begum Z, Hafiz MZ (2009) Fault tolerant reversible logic synthesis: carry look-ahead and carry skip adders. In: International Conference on Advances in Computational Tools for Engineering Applications (ACTEA), pp 396–401
25. Jamal L, Rahman MM, Babu HMH (2013) An optimal design of a fault tolerant reversible multiplier. In: IEEE 26th International SOC Conference (SOCC), pp 37–42
26. Zhou RG, Li Y-C, Zhang M-Q (2014) Novel design for fault tolerant reversible binary coded decimal adders. Int J Electron 101(10):1336–1356
27. Thapliyal H, Arabnia HR, Srinivas MB (2009) Efficient reversible logic design of BCD subtractors. Trans Comput Sci III LNCS 5300:99–121
28. Thapliyal H, Jayashree HV, Nagamani AN, Arabnia HR (2013) Progress in reversible processor design: a novel methodology for reversible carry look-ahead adder. Trans Comput Sci XVII LNCS 7420:73–97
29. Mitra SK, Chowdhury AR (2012) Minimum cost fault tolerant adder circuits in reversible logic synthesis. In: 25th IEEE International Conference VLSI Design (VLSID), pp 334–339
30. Baugh CR, Wooley BA (1973) A two's complement parallel array multiplication algorithm. IEEE Trans Comput 22(12):1045–1047