Design of Parity-Preserving Reversible Logic Signed Array Multipliers

Mojtaba Valinataj

***Abstract*—**Reversible logic as a new favorable design domain can be used for various fields especially creating quantum computers because of its speed and intangible power consumption. However, its susceptibility to a variety of environmental effects may lead to yield the incorrect results. In this paper, because of the importance of multiplication operation in various computing systems, some novel reversible logic array multipliers are proposed with error detection capability by incorporating the parity-preserving gates. The new designs are presented for two main parts of array multipliers, partial product generation and multi-operand addition, by exploiting the new arrangements of existing gates, which results in two signed parity-preserving array multipliers. The experimental results reveal that the best proposed 4×4 multiplier in this paper reaches 12%, 24%, and 26% enhancements in the number of constant inputs, number of required gates, and quantum cost, respectively, compared to previous design. Moreover, the best proposed design is generalized for *n*×*n* multipliers with general formulations to estimate the main reversible logic criteria as the functions of the multiplier size.

***Keywords*—**Array multipliers, Baugh-Wooley method, error detection, parity-preserving gates, quantum computers, reversible logic.

I. Introduction

NOWADAYS, the power consumption receives considerable attention because of its growth in different Very-Large-Scale Integration (VLSI) circuits. The reversible logic design domain is a good candidate to overcome the high power consumption because there is not any information loss in these circuits. This is based on the fact that the 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, different from ordinary logic circuits in which information is lost, the circuits made of only reversible logic gates do not dissipate this type of energy as the internal power. Therefore, reversible circuits are eligible for more research despite having the physical and implementation problems.

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 circuits. Regarding the main property of reversible logic circuits, these circuits can be thought for using in different applications such as DNA computations, nano-computing, quantum computing, and low power circuits.

Basically, the external noises and environmental effects can result in a fault and cause a reversible circuit to deviate from producing correct outputs. Moreover, in this case, the information is lost because the input vector cannot be retrieved from the output vector. Thus, similar to irreversible circuits, the fault-tolerance capability at least in the form of 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 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 [2].

The multiplication is one of the most important arithmetic operations in different computing systems. Among different types of multipliers, the array multipliers have received more attention as the fast multipliers. Therefore, in reversible logic domain, array multipliers should be considered especially respecting error detection capability. So far, different types of reversible multipliers have been designed as exemplified in [3]-[9]. Even though many of these designs 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 (QC) compared to the few previous 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. Based on the results analysis, the proposed designs show more optimized criteria compared to their previous counterparts especially regarding gate count and QC.

The rest of the paper is organized as follows. In Section II, some basic concepts and definitions as well as the parity-preserving reversible gates are described. In Section III, the related works are discussed. In Section IV, the new designs for the PPG part, and in Section V, the new designs for the MOA part are proposed. Constructing the new signed parity-preserving array multipliers by combining the proposed PPGs and MOAs is shown and evaluated in Section VI. Finally, some conclusions are drawn in Section VII.

M. Valinataj is with the Department of Electrical and Computer Engineering, Babol Noshirvani University of Technology, Babol, 47148-71167 Iran (phone: +98 11-3233-9214, e-mail: m.valinataj@nit.ac.ir).

II. Background

A. Basic Concepts and Definitions

A reversible gate 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. Since the input vector can be retrieved by the output vector, 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 $\oplus$ represents for the XOR operation:

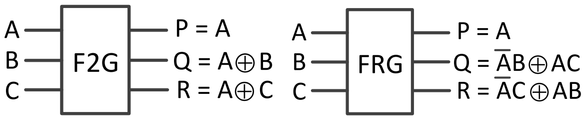
$$I_0\oplus I_1\oplus\ldots\oplus I_{n-1}= O_0\oplus O_1\oplus\ldots\oplus O_{n-1} \qquad (1)$$

The parity-preserving property for a gate makes possible all single error detection and somehow 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, to obtain an error-detecting reversible circuit, only the parity-preserving gates should be used. In a reversible gate, 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 [10]. In addition, the outputs that would not be used in the subsequent computations are called the garbage outputs. In the other words, the garbage outputs are needed just to maintain the circuit's reversibility or to make it parity-preserving [11]. Another parameter considered in reversible circuits is the hardware complexity 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 implementations. In this way, the symbols α, β, and γ may be used as the representatives for XOR, AND, and NOT operations in the outputs, respectively. As stated in [12], 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 [12] is used.

The 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 QC of one unit. However, for constructing the reversible gates bigger than 2×2, different quantum primitives should be used. The reversible gates can be classified in two groups, parity-preserving 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.

B. Parity-Preserving Reversible Gates

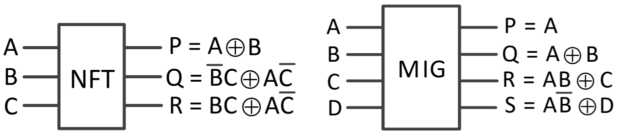
1. Double Feynman gate (F2G) [13] as a parity-preserving 3×3 reversible gate with the QC of two is shown in Fig. 1 (a). The hardware complexity of this gate equals 2α. This gate can be used as a fan-out generator in reversible circuits.
2. Fredkin gate (FRG) [14] (Fig. 1 (b)) as the oldest parity-preserving reversible gate with the QC 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 NOT operation in its outputs. This gate is a universal gate which means that all reversible logic circuits can be implemented by using only this type of gates.



(a) (b)

Fig. 1 Block diagrams of (a) F2G, and (b) FRG

3. New fault-tolerant gate (NFT) [15] as another parity-preserving reversible gate with the QC of five has the hardware complexity equal to $3\alpha + 3\beta + 2\gamma$ according to its outputs shown in Fig. 2 (a). Similar to FRG, this gate is a universal gate.
4. Modified Islam gate (MIG) [16] (Fig. 2 (b)) is a 4×4 parity-preserving reversible gate with the QC 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.



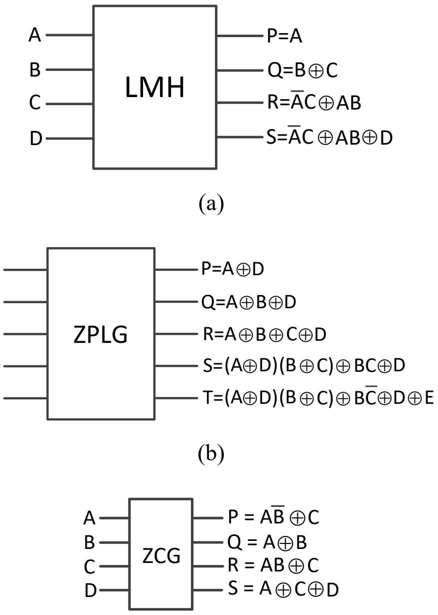
(a) (b)

Fig. 2 Block diagrams of (a) NFT, and (b) MIG

5. Lafifa-Mushfiq-Hafiz (LMH) [17] shown in Fig. 3 (a) is a 4×4 parity-preserving reversible gate with the QC 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 [12]. Thus, since two XOR operations in the output expressions operate on the same operands (in R and S outputs shown in Fig. 3 (a)), this gate includes three distinct XOR operations, which results in 3α instead of 4α. In addition, two same $\overline{A}$C operations 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. ZPLG [18] shown in Fig. 3 (b) is another 5×5 parity-preserving reversible gate with the QC 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 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 QC.

7. ZCG [18] shown in Fig. 3 (c) is a 4×4 parity-preserving reversible gate with the QC 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.



(a)

(b)

(c)

Fig. 3 Block diagrams of (a) LMH gate, (b) ZPLG, and (c) ZCG

III. Related Works

A. Parity-Preserving Reversible Full Adders

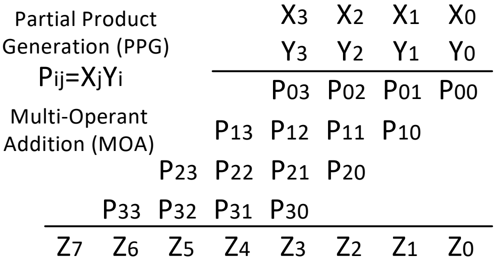
All types of multipliers, i.e. serial, parallel, and array multipliers, in some manner require the addition operation. This operation is usually performed by using full adders and half adders. There exist some parity-preserving gates that can perform the operation of a parity-preserving full adder (such as F2PG [6], LCG [12] and ZPLG [18]) or half adder (MIG [16] and ZCG [18]) 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 single NFT full adder (SNFA) [19] which includes three F2Gs and a NFT gate. This gate has the QC of 11, which is more than that of LCG and ZPLG but less than that of F2PG, and its hardware complexity is equal to $9\alpha + 3\beta + 2\delta$.

B. 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 sub-group 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 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.

One of the popular parallel multiplier architectures is array multiplier. As stated before, the array multipliers include two parts, PPG and MOA as shown in Fig. 4. 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 [6] based on the Baugh-Wooley method [20]. In this design, the Wallace tree structure is used for the MOA part. According to [6], a 5×5 signed multiplier requires 57 reversible gates with the total QC 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 [5], 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 QC of 244 for a 4×4 multiplier.



Partial Product Generation (PPG)					X_3	X_2	X_1	X_0
					Y_3	Y_2	Y_1	Y_0
$P_{ij}=X_jY_i$					P_{03}	P_{02}	P_{01}	P_{00}
Multi-Operant Addition (MOA)				P_{13}	P_{12}	P_{11}	P_{10}	
			P_{23}	P_{22}	P_{21}	P_{20}		
		P_{33}	P_{32}	P_{31}	P_{30}			
	Z_7	Z_6	Z_5	Z_4	Z_3	Z_2	Z_1	Z_0

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

IV. Proposed PPG 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 be started. In an irreversible n×n array multiplier, n^2 AND gates are required to produce all P_{ij}s that are equal to x_jy_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. 4 despite the fact that there is not any separate AND gate in reversible logic.

Signed array multipliers may require different partial products. The best signed array multipliers are based on the Baugh-Wooley method [20]. In fact, based on the Baugh-

Wooley method, two different partial product arrangements can be used according to Figs. 5 and 6. In this paper, we call them BW1 and BW2, respectively, in which the second arrangement (Fig. 6) has lower costs. Therefore, proposing the PPGs in which all the items shown in Figs. 5 and 6 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. 5, some terms include an inverted operand. These terms can be produced by properly adjusting the inputs of FRGs. This way, the first proposed parity-preserving PPG circuit for a 4×4 array multiplier is shown in Fig. 7. Based on this figure, the first proposed PPG circuit has the QC of 100 because it includes 16 FRGs and 10 F2Gs (16×5+10×2=100).

According to Fig. 6, 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. 6, a new arrangement of LMH gates and FRGs with appropriate input adjustments is suggested as the second proposed parity-preserving PPG circuit for a 4×4 array multiplier as shown in Fig. 8. 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 QC of this PPG circuit is equal to 91 and is the lowest among the existing PPG circuits.

To compare the proposed PPG circuits in this paper with the previous designs, Table I illustrates different characteristics and reversible logic criteria for the 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 I, the bold items show the best values in each column. According to this table, the second proposed PPG is the best for the signed multiplication in all criteria.

				x_3	x_2	x_1	x_0
				y_3	y_2	y_1	y_0
				$x_3\overline{y_0}$	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	$\overline{x_2}y_3$	$\overline{x_1}y_3$	$\overline{x_0}y_3$			
	$\overline{x_3}$	0	0	x_3			
1	$\overline{y_3}$	0	0	y_3			
P_7	P_6	P_5	P_4	P_3	P_2	P_1	P_0

Fig. 5 4×4 signed multiplication based on the first Baugh-Wooley method

				x_3	x_2	x_1	x_0
				y_3	y_2	y_1	y_0
			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	$\overline{x_2y_3}$	$\overline{x_1y_3}$	$\overline{x_0y_3}$			
P_7	P_6	P_5	P_4	P_3	P_2	P_1	P_0

Fig. 6 4×4 signed multiplication based on the second Baugh-Wooley method

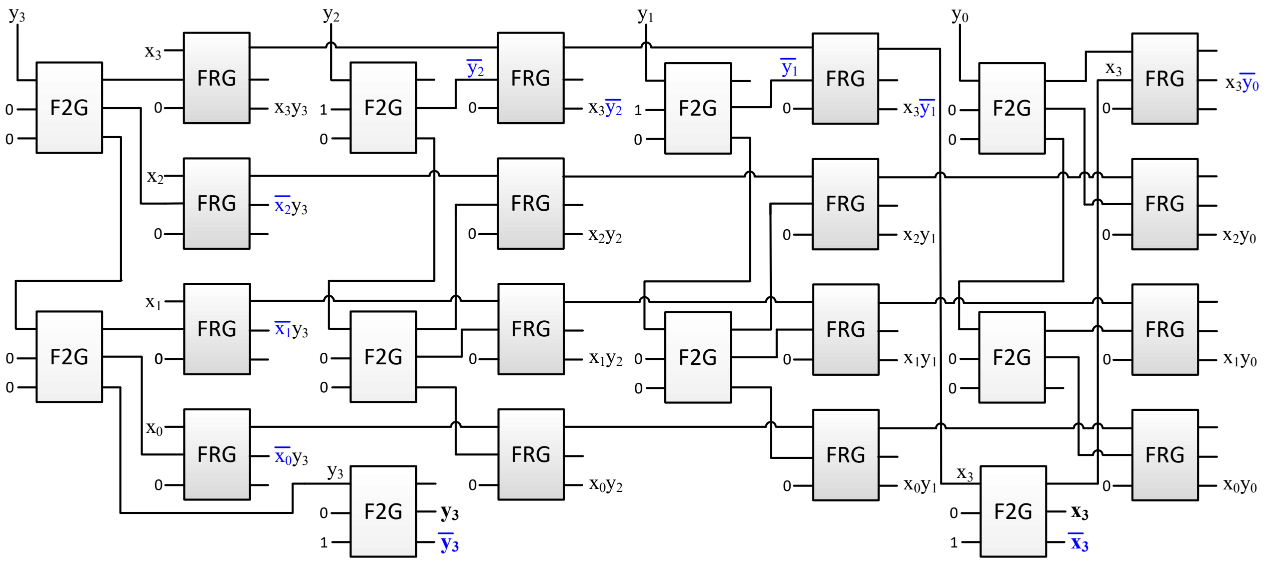


Fig. 7 First proposed 4×4 PPG based on the first Baugh-Wooley method

TABLE I
COMPARISON OF PPG PARTS OF DIFFERENT PARITY-PRESERVING ARRAY MULTIPLIERS

4×4 Multiplier	Base Algorithm	Signed	Gate Count	Constant Inputs	Garbage Outputs	QC	Hardware Complexity
[6]	BW2	Yes	26	36	23	100	58α+58β+22γ
1st proposed PPG (Fig. 7)	BW1	Yes	26	36	24	100	52α+64β+16γ
2nd proposed PPG (Fig. 8)	BW2	Yes	**16**	**27**	**19**	**91**	**43α+42β+16γ**

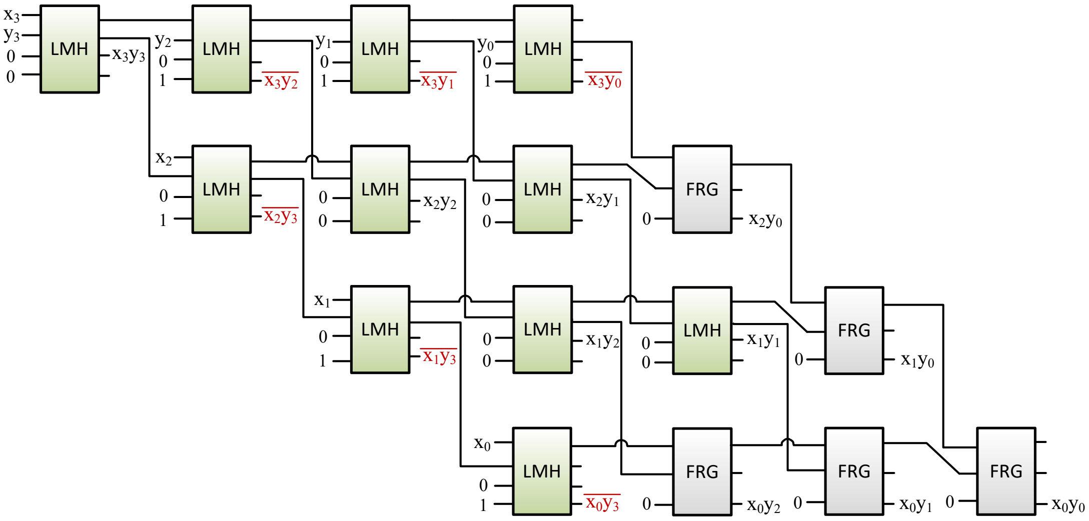


Fig. 8 Second proposed 4×4 PPG based on the second Baugh-Wooley method

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 QC of the proposed PPGs can be computed by using the following equations, for n-bit operands (n>=2) to be used in n×n array multipliers.

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

$$QC\ \text{of}\ 1st\ (n \times n)\ PPG = 5n^2 + 2(n \times \lfloor n/2 \rfloor + \ n \bmod 2) + 4 \tag{3}$$

$$Gates\ of\ 2\text{nd}\ (n \times n)\ PPG = ((n-1)^2 + 2) \times LMH + (2n-3) \times FRG \tag{4}$$

$$QC\ of\ 2nd\ (n \times n)\ PPG = 6n^2 - 2n + 3 \tag{5}$$

V. PROPOSED MOA CIRCUITS

The second part of an array multiplier is the 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 QC as the main criterion are ZPLG and ZCG, both introduced in [18] with the QC of 8 and 6, respectively. Regarding the other criteria, the numbers of constant inputs and garbage outputs are almost the same in different full adder and half adder designs.

For signed multipliers based on the Baugh-Wooley method, different MOAs should be designed for BW1 and BW2 because of different structures of Figs. 5 and 6. In fact, the new MOA circuits should add the partial products produced by the PPG circuits for BW1 and BW2. Therefore, the first parity-preserving MOA circuit for a 4×4 signed array multiplier is proposed in Fig. 9 which is beneficial for the BW1 method. 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. 9, when two operands should be added together ZCG is used as the half adder. Moreover, in Fig. 9, 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 QC of 114 because of having 12 ZPLGs and three ZCGs. In addition, the second proposed parity-preserving MOA circuit for a 4×4 signed array multiplier is depicted in Fig. 10 which is useful for the BW2 method. In Fig. 10, the only F2G is used to perform an operation equivalent to the addition by one based on the lower-left '1' shown in Fig. 6. In fact, in Fig. 10, a ZCG as a half adder should add the output carry of lower-left ZPLG by one which requires six units more QC. 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 QC by four.

To compare the proposed MOA circuits in this paper with the previous designs, Table II illustrates different characteristics and reversible logic criteria for all MOA designs. In this table, the first two 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. According to this table, the second proposed MOA has the lowest QC among the designs beneficial for signed multiplication.

TABLE II
COMPARISON OF MOA PARTS OF DIFFERENT PARITY-PRESERVING ARRAY MULTIPLIERS

4×4 Multiplier	Base Algorithm	Signed	Gate Count	Constant Inputs	Garbage Outputs	QC	Hardware Complexity
[5]	Simple array	No	20	**24**	32	140	**60α+40β+20γ**
[17]	Simple array	No	36	**24**	32	116	84α+32β+20γ
[6]	BW2	Yes	**12**	25	**33**	147	63α+51β+21γ
1st proposed MOA (Fig. 9)	BW1	Yes	15	31	43	114	111α+42β+15γ
2nd proposed MOA (Fig. 10)	BW2	Yes	13	27	35	**92**	89α+33β+12γ

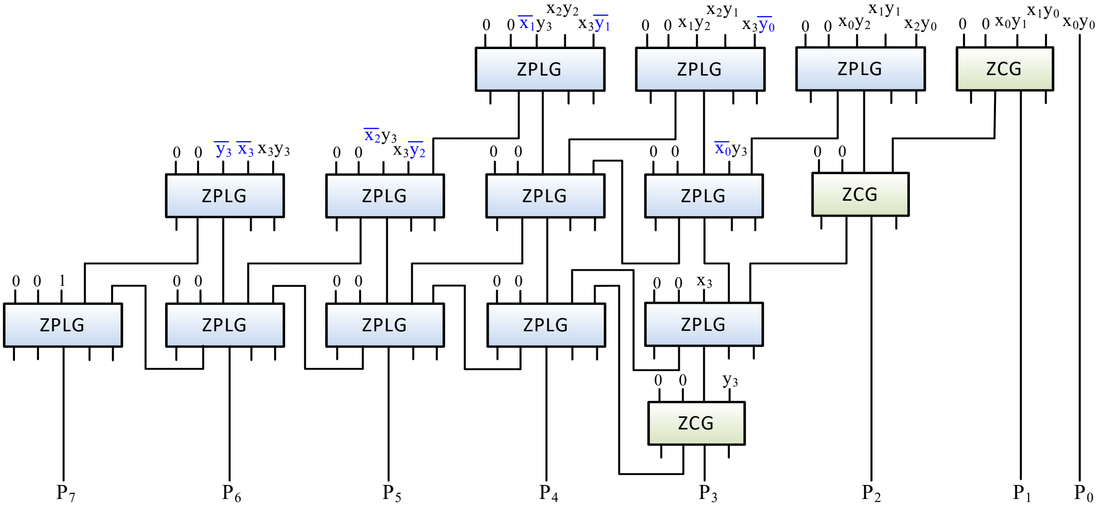


Fig. 9 First proposed MOA for 4×4 signed array multiplier

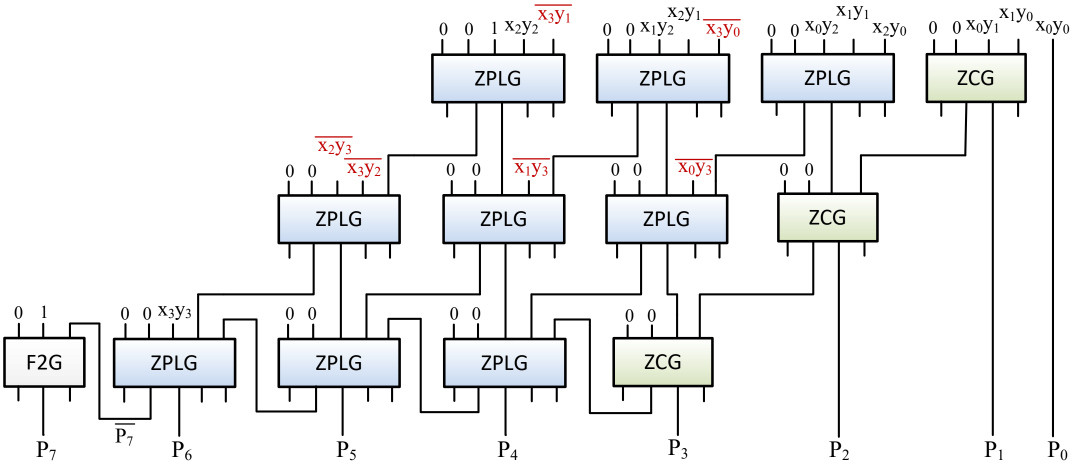


Fig. 10 Second proposed MOA for 4×4 signed array multiplier

To extend the size of second proposed MOA circuit to be used in larger multipliers, the following equations can be used to predict the number of different required gates and total QC for n-bit operands.

$$Gates\ of\ 2nd\ (n \times n)\ MOA = (n-1)^2 \times ZPLG + (n-1) \times ZCG + 1 \times F2G \quad (6)$$

$$QC\ of\ 2nd(n \times n)\ MOA = 8n^2 - 10n + 4 \quad (7)$$

VI. RESULTS AND DISCUSSION

In this section, the proposed parity-preserving signed 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 previous designs. In the comparisons, similar to Tables I and II, five main criteria are used including gate count, number of constant inputs, number of garbage outputs, QC, and hardware complexity even though the QC is the most important criterion.

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 two new parity-preserving signed array multipliers. For the 4×4 multiplication, the first proposed multiplier is constructed by combining the first PPG (Fig. 7) and the first MOA (Fig. 9) proposed before based on the BW1 method. This multiplier has the QC of 214 that is the sum of 100 and 114. The second proposed signed multiplier is constructed by combining the second PPG (Fig. 8) and the second MOA (Fig. 10) based on the BW2 method. This combination leads to the best signed array multiplier with the QC of 183 which is the minimum value among all designs.

The proposed parity-preserving signed array multipliers are characterized in Table III along with previous designs. 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 [6] or the second proposed multiplier in this paper, can be judged as the best. Moreover, Table IV depicts the best proposed signed array multiplier in this paper compared to the only existing 5×5 design from [6] 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 [6].

To illustrate the precise amounts of improvements attained by the best proposed signed multiplier, Fig. 11 depicts the percentages of reduction in four criterions for the second proposed design compared to the design in [6] for 4×4 and 5×5 multiplier sizes. Based on this figure, all criteria have some enhancements in the second proposed multiplier. The best improvements are obtained for the QC 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 the second proposed multiplier for larger input operands, Table V demonstrates the formulae for the major reversible logic criteria for *n*×*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.

VII. Conclusion

In this paper, two novel parity-preserving reversible signed array multipliers were proposed by designing some new PPG and MOA circuits required in array multipliers. To attain better designs, the new arrangements of existing parity-preserving reversible gates were utilized as well as exploiting newer gates. The proposed signed array multipliers are based on two types of the Baugh-Wooley method. The best proposed signed array multiplier in this paper has achieved considerable improvement in the QC and gate count compared to previous designs. In addition to the basic 4×4 multipliers, the proposed multipliers have been investigated for *n*×*n* multipliers by exploiting some general formulations. The experimental results have revealed the prominence of the proposed multipliers with different sizes compared to previous designs respecting the reversible logic criteria.

TABLE III
Comparison of Different Parity-Preserving Signed Array Multipliers

4×4 multiplier	Base algorithm	Gate count	Constant inputs	Garbage outputs	QC	Hardware complexity
[5]	Simple array, unsigned	48	64	64	244	116α+104β+36γ
[17]	Simple array, unsigned	52	**49**	**49**	205	125α+78β+36γ
[6]	BW2, signed	38	61	56	247	121α+109β+43γ
1st proposed circuit (combination of Figs. 7 and 9)	BW1, signed	41	67	67	214	163α+106β+31γ
2nd proposed circuit (combination of Figs. 8 and 10)	BW2, signed	**29**	54	54	**183**	132α+75β+28γ

TABLE IV
Best Proposed Signed Array Multiplier Compared to the Only Existing 5×5 Baugh-Wooley Multiplier

5×5 multiplier	Gate count	Constant inputs	Garbage outputs	QC	Hardware complexity
[6]	57	90	90	401	190α+180β+69γ
2nd proposed circuit (combination of Figs. 8 and 10)	49	86	86	297	218α+120β+45γ

TABLE V
Evaluation of the Best Proposed Signed Array Multiplier with Different Sizes Based on Its General Formulae

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

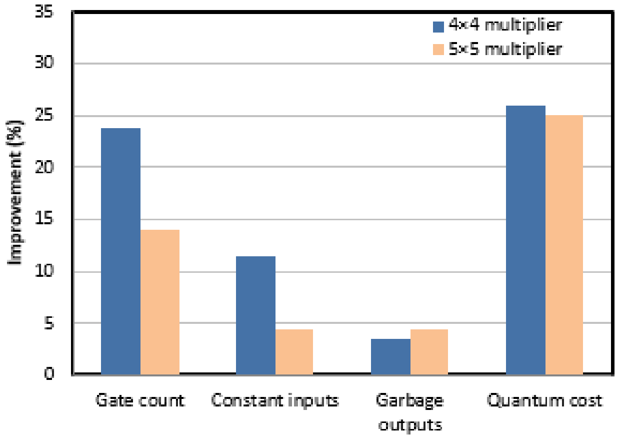


Fig. 11 Improvements of the 5th proposed array multiplier compared to the design in [6] for different sizes

REFERENCES

[1] R. Landauer, "Irreversibility and heat generation in the computing process," IBM Journal of Research and Development, vol. 5, no. 3, pp. 183–191, 1961.

[2] M. Perkowski, et al., "A general decomposition for reversible logic," Proc. RM, pp. 119–138, 2001.

[3] E. Pouraliakbar, M. Haghparast, and K. Navi, "Novel design of a fast reversible Wallace sign multiplier circuit in nanotechnology," Microelectronics Journal, vol. 42, pp. 973–981, 2011.

[4] M. Z. Moghadam and K. Navi, "Ultra-area-efficient reversible multiplier," Microelectronics Journal, vol. 43, pp. 377–385, 2012.

[5] S. Babazadeh and M. Haghparast, "Design of a nanometric fault tolerant reversible multiplier circuit," Journal of basic and applied scientific research, vol. 2, no. 2, pp. 1355–1361, 2012.

[6] X. Qi and F. Chen, "Design of fast fault tolerant reversible signed multiplier," Intl. Journal of the Physical Sciences, vol. 7, no. 17, pp. 2506–2514, 2012.

[7] S. Kotiyal, H. Thapliyal, and N. Ranganathan, "Circuit for reversible quantum multiplier based on binary tree optimizing ancilla and garbage bits," Proc. of 27th Intl. Conf. on VLSI Design (VLSID), pp. 545–550, 2014.

[8] V. G. Moshnyaga, "Design of minimum complexity reversible multiplier," Proc. of IEEE Region 10 Conf. (TENCON), pp. 1–4, 2015.

[9] S. Kotiyal, H. Thapliyal, and N. Ranganathan, "Reversible logic based multiplication computing unit using binary tree data structure," Journal of Supercomputing, vol. 71, pp. 2668–2693, 2015.

[10] D. Maslov and G. W. Dueck, "Reversible cascades with minimal garbage," IEEE Trans. CAD Integr. Circuits Syst., vol. 23, no. 11, pp. 1497–1509, 2004.

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

[12] M. Valinataj, M. Mirshekar, and H. Jazayeri, "Novel low-cost and fault-tolerant reversible logic adders," Computers and Electrical Engineering, vol. 53, pp. 56–72, 2016.

[13] B. Parhami, "Fault-tolerant reversible circuits," 40th Asilomar Conference on Signals, Systems and Computers (ACSSC), pp. 1726–1729, 2006.

[14] E. Fredkin and T. Toffoli, "Conservative logic," Intl. Journal of Theoretical Physics, vol. 21, pp. 219–253, 1982.

[15] M. Hagparast and K. Navi, "A novel fault tolerant reversible gate for nanotechnology based system," American Journal of Applied Sciences, vol. 5, no. 5, pp. 519–523, 2008.

[16] M. S. Islam, M. M. Rahman, Z. Begum, and M. Z. Hafiz, "Fault tolerant reversible logic synthesis: carry look-ahead and carry skip adders," Intl. Conf. on Advances in Computational Tools for Engineering Applications (ACTEA), pp. 396–401, 2009.

[17] L. Jamal, M. M. Rahman, and H. M. H. Babu, "An optimal design of a fault tolerant reversible multiplier," IEEE 26th Intl. SOC Conf. (SOCC), pp. 37–42, 2013.

[18] R. G. Zhou, Y.-C. Li, and M.-Q. Zhang, "Novel design for fault tolerant reversible binary coded decimal adders," Intl. Journal of Electronics, vol. 101, no. 10, pp.1336–1356, 2014.

[19] S. K. Mitra and A. R. Chowdhury, "Minimum cost fault tolerant adder circuits in reversible logic synthesis," 25th IEEE Intl. Conf. VLSI Design (VLSID), pp. 334–339, 2012.

[20] C. R. Baugh and B. A. Wooley, "A two's complement parallel array multiplication algorithm," IEEE Trans. Comput., vol. 22, no. 12, pp. 1045–1047, 1973.