



Reversible Logic Multipliers: Novel Low-cost Parity-Preserving Designs

F. Eslami-Chalandar, M. Valinataj*, H. Jazayeri

School of Electrical and Computer Engineering, Babol Noshirvani University of Technology, Babol, Iran

PAPER INFO

Paper history:

Received 05 April 2018

Received in revised form 27 February 2019

Accepted 07 March 2019

Keywords:

Reversible Logic

Parity-Preserving Gates

Multiplication

Booth's Algorithm

Error Detection

Fault-Tolerance

ABSTRACT

Reversible logic is one of the new paradigms for power optimization that can be used instead of the current circuits. Moreover, the fault-tolerance capability in the form of error detection or error correction is a vital aspect for current processing systems. In this paper, as the multiplication is an important operation in computing systems, some novel reversible multiplier designs are proposed with the parity-preserving property which will be useful for error detection. At first, two optimal signed serial multipliers are presented based on the Booth's algorithm and its enhanced version called the K-algorithm, utilizing the new arrangements of reversible gates. Then, another low-cost serial multiplier is proposed based on the conventional Add & Shift method to be utilized in the applications in which unsigned numbers are used. Finally, a new signed parallel multiplier is proposed based on the Baugh-Wooley method that is useful for speed-critical applications. The comparative results showed that the proposed multipliers are much better than the existing designs regarding the main criteria used in reversible logic circuits including quantum cost, gate count, constant inputs, and garbage outputs.

doi: 10.5829/ije.2019.32.03c.05

1. INTRODUCTION

Generally, the VLSI circuits are built using irreversible gates and circuits that always lead to power dissipation. It is proved in literature [1] that each bit in irreversible logic consumes at least $kT \ln 2$ Joules of energy in which k is the Boltzmann's constant and T is the absolute temperature at which the computation is performed. Reversible logic is one of the best solutions to decrease energy consumption since there is no energy dissipation in this kind of circuit as the internal power consumption [2]. Reversible circuits are made of reversible gates, and it is required that a one by one mapping exists between the input vector and the output vector of each gate or circuit. This way, the number of outputs is equal to the number of inputs, and the input vector can be retrieved from the output vector. That means no information is lost in these circuits. This fact helps to decrease power consumption. Reversible circuits may have lots of applications in designing low power circuits, quantum computing and nanotechnology although nowadays there are some problems in the design of quantum circuits.

Similar to irreversible circuits, reversible circuits are fault-prone in their operations because a fault inside a reversible gate caused by an environmental effect can corrupt the resultant output vector which makes the input vector not to be recovered from the output vector, and the information is lost. Therefore, the fault-tolerance capability, at least in the form of fault or error detection, is an important aspect in reversible circuits. A well-known and low-cost method to detect errors is parity-based coding. However, in reversible gates and circuits, this coding can be used in the form of parity-preserving characteristic. A gate having this characteristic is called a parity-preserving reversible gate. In this type of reversible gate, the parity of the input vector is equal to the parity of the output vector. However, since fan-out and feedback are not allowed in reversible logic [3,4]; the implementation of this type of circuits is more difficult compared to irreversible circuits.

The multiplication is one of the important arithmetic operations in different computing systems including the quantum computers. Thus, designing a better multiplier with respect to different design aspects assists to reach a more efficient processing system. Until now, different

*Corresponding Author Email: m.valinataj@nit.ac.ir (M. Valinataj)

types of reversible multipliers have been designed [5-13]. However, many of the designs are not fault-tolerant or parity-preserving circuits. In this paper, some new low-cost parity-preserving reversible multipliers are proposed; in which more beneficial parity-preserving gates as well as better arrangements of the existing gates are exploited. The proposed multipliers include both serial and parallel architectures to be used for signed and unsigned multiplications in different applications. These designs are based on different multiplication algorithms comprising the Booth's algorithm and K-algorithm [14], Add & Shift, and Baugh-Wooley algorithm [15]. It is shown that the proposed multipliers have better design parameters compared to previous reversible multipliers especially with respect to quantum cost.

The rest of the paper is organized as follows. In section 2, some basic concepts and definitions as well as the parity-preserving reversible gates are described. In section 3 the related works are characterized. Sections 4 and 5 explain the proposed parity-preserving serial and parallel multipliers, respectively. The evaluation of the proposed reversible multipliers compared to the existing designs is presented in section 6. Finally, some conclusions are drawn in section 7.

2. BACKGROUND

In this section, at first, we discussed the basic concepts and definitions regarding reversible logic. Then, we introduced the parity-preserving reversible gates, required for the next sections of this paper.

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 $I_v = (I_0, I_1, \dots, I_{n-1})$ and $O_v = (O_0, O_1, \dots, O_{n-1})$ as 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:

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

The parity-preserving characteristic for a gate makes possible single error detection and in some cases 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. After designing a parity-preserving circuit, the error detection process can be performed using the rules stated in literature [16,17].

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 [18]. 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 [19].

In a reversible circuit, the delay is defined as the maximum number of gates on the paths from the inputs to the outputs [19]. 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 types of implementations. This way, if α , β , and γ are the representatives for XOR, AND, and NOT operations in the outputs, respectively. Then, the hardware complexity i.e. *H.C.* can be computed according to Equation (2):

$$H.C. = N(\alpha) \cdot \alpha + N(\beta) \cdot \beta + N(\gamma) \cdot \gamma \quad (2)$$

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

As stated in literature [20], in calculating the hardware complexity, it would be desired and more precise if the common operations in the output expressions would be accounted once. Therefore, in this paper, the calculation approach presented in literature [20] is used.

The most important parameter in designing the reversible circuits is the quantum cost. 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. The quantum primitives are used to build the reversible gates bigger than 2×2 . 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. In this paper, we are only dealing with the parity-preserving circuits; the main parity-preserving gates are introduced in the following section.

2. 2. Parity-Preserving Reversible Gates

1. Double Feynman gate (F2G) [21] as a parity-preserving 3×3 reversible gate with the quantum cost of two is shown in Figure 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] (Figure 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 gate.

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 $3\alpha + 3\beta + 2\gamma$. Similar to FRG, this gate is a universal gate.

4. Modified Islam gate (MIG) [24] is a 4×4 parity-preserving reversible gate with the quantum cost of 7 and the hardware complexity of $3\alpha + 2\beta + 1\gamma$. This gate is also a universal gate. In addition, this gate can be used as a parity-preserving half adder when its two last inputs are set to zero. In this case, the sum and carry are produced, accordingly.

5. LMH [25] shown in Figure 2 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 are accounted once according to the approach presented in literature [20]. Thus, as two XOR operations in the output expressions operate on the same operands (in R and S outputs shown in Figure 2), it results in 3α instead of 4α . In addition, two same A'C operations and two same AB operations exist in the output expressions which are result in a simpler term 2β instead of 4β . Finally, a distinct NOT operation (A') results in 1γ .

6. ZCG [26] shown in Figure 3a is another 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.

7. ZPLG [26] shown in Figure 3b is a 5×5 parity-preserving reversible gate with the quantum cost of eight and its hardware complexity is equal to $8\alpha + 3\beta + 1\gamma$.

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 sum and carry are produced on the R and S outputs, respectively. In addition, this gate produces the minimum cost full adder.

8. Low-cost gate (LCG) [20] is another 5×5 parity-preserving reversible gate which has the quantum cost of 10. The hardware complexity of this gate is equal to $6\alpha + 2\beta$. Similar to ZPLG, this gate can be used as a parity-preserving full adder when its two last inputs are set to zero. In this case, the sum and carry are produced, accordingly. Despite the fact that the quantum cost of LCG is higher than that of ZPLG, its hardware complexity is much less than that of ZPLG which makes it more desirable in some applications.

3. RELATED WORKS

3. 1. Parity-Preserving Reversible Full Adders

Both types of serial and parallel multiplications somehow require the addition operation. This operation is usually performed by using full adders and half adders.

As stated before, there exist some parity-preserving gates that can perform the operation of a parity-preserving full adder (LCG [20] and ZPLG [26]) or half adder (MIG [24] and ZCG [26]) after setting some of their inputs to zero. 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) [27] in which three F2Gs and a NFT gate have been used. This gate has the quantum cost of 11 which is more than that of LCG and ZPLG, and its hardware complexity is equal to $9\alpha + 3\beta + 2\delta$.

3. 2. Parity-Preserving Reversible Multipliers

Since the multiplier is one of the important elements of a computing system, many studies have been performed to design optimal multipliers. However, despite the fact that

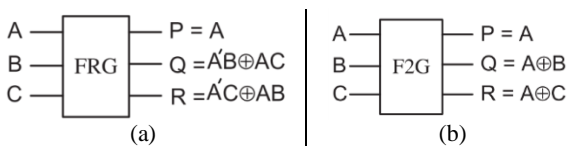


Figure 1. Block diagrams of (a) double Feynman gate, and (b) Fredkin gate

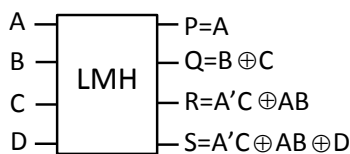


Figure 2. Block diagram of LMH gate

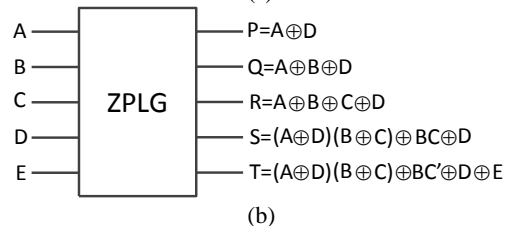
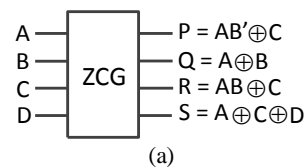


Figure 3. Block diagrams of (a) ZCG, and (b) ZPLG

there are many works [5-11] regarding reversible logic multipliers and even the recent designs reported in literature [12, 13, 28-32], there is not much work incorporating the parity-preserving multipliers. The multipliers are designed in two manners, serial or parallel. 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, parallel multipliers are better because they require a lower delay.

3. 2. 1. Serial Multipliers As stated above, there is not much work with respect to the parity-preserving serial multipliers. In fact, this type of multipliers is only proposed in literature [14] based on the well-known Booth's algorithm and its modified version called Keshuv or K-algorithm for multiplying signed numbers. The general structure of a 4-bit multiplier based on the Booth's algorithm is shown in Figure 4. As stated in literature [14], this circuit has been implemented using 46 reversible gates and with the total quantum cost of 200. It should be noted that in Figure 4, A is the first operand or multiplicand, X is the second operand or multiplier, and Z is the product.

Table 1 is used in the Booth's-based multiplier. However, the K-algorithm proposed in literature [14] (shown in Figure 5) utilizes Table 2 to perform the required operations. This table results in simpler circuit by using 2 to 1 multiplexers instead of 4 to 1 multiplexers used in the Booth's-based multiplier. In addition, it does not require copying the first operand opposed to the Booth's-based multiplier. As shown in Figure 5, in the K-algorithm the select line's value of 2 to 1 multiplexer is equal to $x_i \oplus x_{i-1}$. If this value equals '0', a 4-bit zero number is selected; otherwise the first operand (A) is selected. The two's complement of A is produced by an XOR operation between each bit of A and y_i shown in Table 2, and then, adding to the input carry equal to y_i . This method produces the two's complement of A if y_i equals '1'. The proposed multiplier in literature [14] based on the K-algorithm includes 39 reversible gates with the quantum cost of 126.

3. 2. 2. Parallel Multipliers One of the popular parallel multiplier architectures is array multiplier that includes two steps, partial product generation (PPG) and

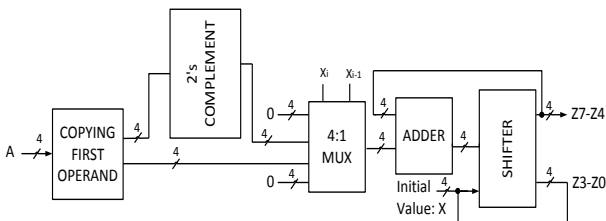


Figure 4. Block diagram of a 4-bit Booth's-based multiplier according to literature [14]

TABLE 1. Operations in the Booth's algorithm versus consecutive bits of multiplier

x_i	x_{i-1}	Required operation
0	0	Addition with zero, equivalent to no operation
0	1	Add A to partial product
1	0	Subtract A from partial product
1	1	Addition with zero, equivalent to no operation

TABLE 2. Operations in the K-algorithm versus consecutive bits of multiplier [14]

x_i	x_{i-1}	y_i	Required operation
0	0	-	No operation
0	1	0	Pass A as it is to the addition stage
1	0	1	Pass A as it is to the addition stage
1	1	-	No operation

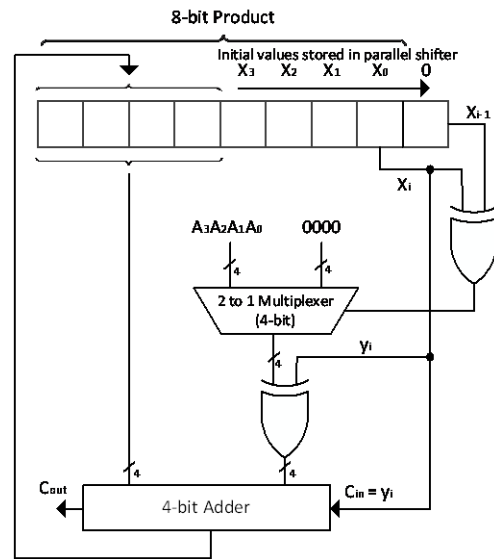


Figure 5. Block diagram of a 4-bit multiplier based on the K-algorithm [14]

multi-operand addition (MOA) in which the partial products will be added together. Despite the fact that various reversible array multipliers exist in the literature, few designs are parity-preserving, as well. The first parity-preserving signed array multiplier is proposed in literature [9] based on the Baugh-Wooley method [15]. As stated in literature [9], this multiplier includes 57 gates with the quantum cost of 401 for 5-bit input operands. In this design, two new parity-preserving gates called MNFT (modified NFT) and F2PG are used in addition to the well-known parity-preserving gates including F2G, FRG and MIG.

In literature [8] a parity-preserving unsigned array multiplier is proposed utilizing F2Gs and FRGs to implement the PPG part, and MIGs to construct half adders and full adders of the MOA part. This multiplier requires a quantum cost of 244 for 4-bit input operands. In literature [25] another parity-preserving unsigned array multiplier is presented which reduces the required quantum cost to 205. This design utilizes FRG and a new gate called LMH (Lafifa-Mushfiq-Hafiz) to implement the PPG part, and incorporates MIG and SNFA to construct half adders and full adders, respectively, for the MOA part.

4. PROPOSED PARITY-PRESERVING SERIAL MULTIPLIERS

In this section, the new parity-preserving serial multipliers with better criteria compared to the existing designs are introduced in details. The preliminary work regarding serial multipliers is proposed in literature [33]. Similar to previous parity-preserving designs, the proposed multipliers help to detect at least single errors. As stated in section 2, after designing a parity-preserving multiplier, the error detection process can be performed using the methods illustrated in literature [16,17].

4. 1. Signed Multipliers The Booth's algorithm which is the base of signed serial multipliers has five stages according to Figure 4:

Stage (1): copying the first operand's bits (multiplicand's bits)

Stage (2): computing the two's complement of first operand

Stage (3): using a multiplexer to select among the first operand, its two's complement, and zero (based on Table 1)

Stage (4): using an adder to perform the required additions

Stage (5): shifting the result to right arithmetically using a parallel shifter

Stages 1 and 2 are performed once. However, the next stages should be run more times dependent to the size of operands.

As stated before, a reversible circuit should only include the parity-preserving gates if the error detection capability is intended. Therefore, the proposed multipliers comprise only the parity-preserving gates. The first proposed signed serial multiplier is based on the Booth's algorithm. In this multiplier, different stages mentioned above are implemented as follows for 4-bit operands that will be extended to n -bit operands ($n \times n$ multiplier):

- (1) In literature [14] for copying multiplicand's bits, four F2Gs are used. One of the copies is sent to the stage (2) to compute the two's complement of multiplicand,

and the other copy is sent to the multiplexer in stage (3). However, in the first proposed multiplier, the one's complement of multiplicand is produced as well, by the same number of F2Gs and is sent to the next stage to compute the two's complement. In other words, different from literature [14], half of the stage (2) is performed along with the stage (1). The quantum cost of this section with the operation shown in Figure 6 is equal to eight because of using four F2Gs.

- (2) For generating the two's complement of multiplicand, the received one's complement from stage (1) is added to one. However, in literature [14] four NOT gates have been used for inverting four bits of multiplicand before adding to one. Thus, due to the fact that the NOT gate is not parity-preserving, two F2Gs should be used for this purpose according to Figure 7. By this modification, the quantum cost of the multiplier proposed in literature [14] does not change. However, according to Figure 7, both the number of constant inputs and the number of garbage outputs are increased by one while the number of gates is decreased by two.

To perform the required addition and prepare the last two's complement result four half adders are required. In the proposed multiplier, four ZCGs are used instead of MIGs in literature [14] with the total quantum cost of 24.

- (3) Each one-bit 4 to 1 parity-preserving multiplexer is made by three one-bit 2 to 1 multiplexers. Since each one-bit 2 to 1 multiplexer can be constructed by a FRG, the stage (3) requires 12 FRGs with the total quantum cost of 60 for a 4×4 multiplier. The outcome of this section is shown in Figure 8.
- (4) The main adder of the proposed multiplier requires a 4-bit adder which includes a half adder in the least significant bit and three full adders. In the proposed multiplier, ZCG and ZPLG are utilized to construct the only half adder and three full adders, respectively, instead of using MIGs. Therefore, this stage, shown in Figure 9, has the quantum cost of $3 \times 8 + 6 = 30$.
- (5) Different from literature [14] in which seven FRGs and one F2G are used to implement the parallel shifter, eight F2Gs are utilized in the first proposed multiplier to realize the parallel right shift which leads to lower quantum cost. However, some extra F2Gs are required to feedback some bits to the parallel input in the manner that a direct feedback is not produced from a gate's output to its input to prevent unallowable feedbacks. In the proposed Booth's-based multiplier, four F2Gs are used for this purpose instead of seven F2Gs in literature [14]. Therefore this section that is depicted in Figure 10 has the quantum cost of 24.

According to the explanation above, the first proposed signed serial multiplier which is a Booth's-

based design includes 36 gates with the quantum cost of 146 by exploiting new arrangements of the well-known gates to rebuild the different parts in addition to utilizing some newer gates. This multiplier is shown in Figure 11.

To extend the size of first proposed multiplier to be used for larger operands, Equation (3) can be used to compute the number of different gates and total quantum cost. The generalized circuit of proposed Booth's-based multiplier is shown in Figure 12 for n -bit operands.

$$\begin{aligned} \text{Required gates of } (n \times n) \text{ multiplier} \\ = 4n \times F2G + 3n \times FRG + (n + 1) \times ZCG + \\ (n - 1) \times ZPLG \end{aligned} \quad (3)$$

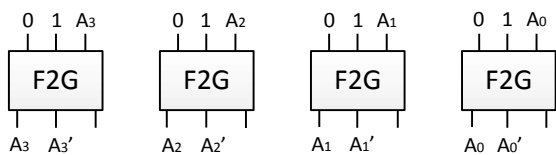


Figure 6. Generation of one's complement of a 4-bit multiplicand along with its replication in the first proposed multiplier

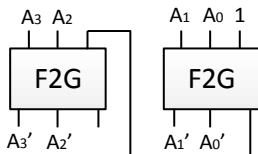


Figure 7. Proposed circuit for inverting four bits of multiplicand instead of using four NOT gates

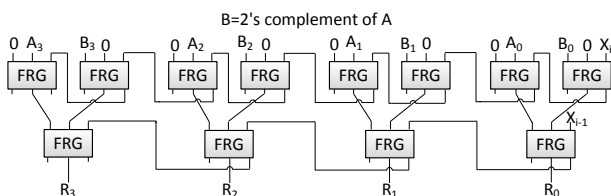


Figure 8. 4-bit 4 to 1 multiplexer based on [14]

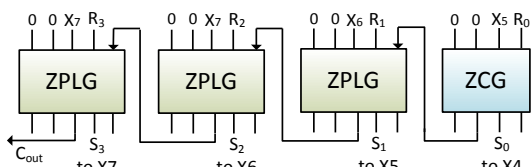


Figure 9. 4-bit adder in the first proposed multiplier

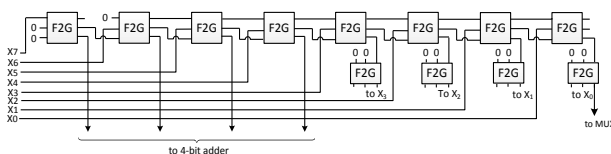


Figure 10. Parallel shifter in the first proposed multiplier

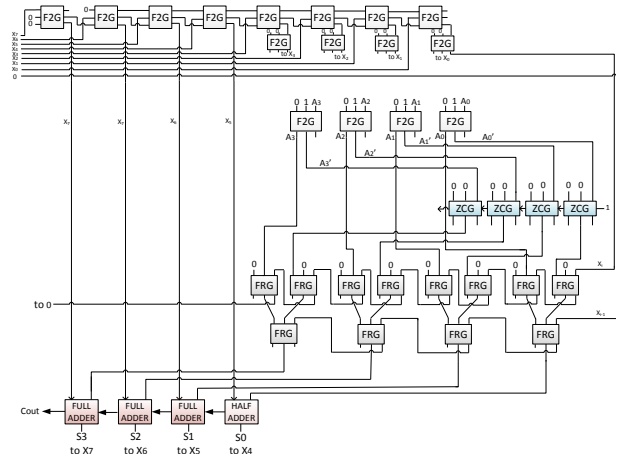


Figure 11. Proposed 4-bit parity-preserving Booth's-based multiplier

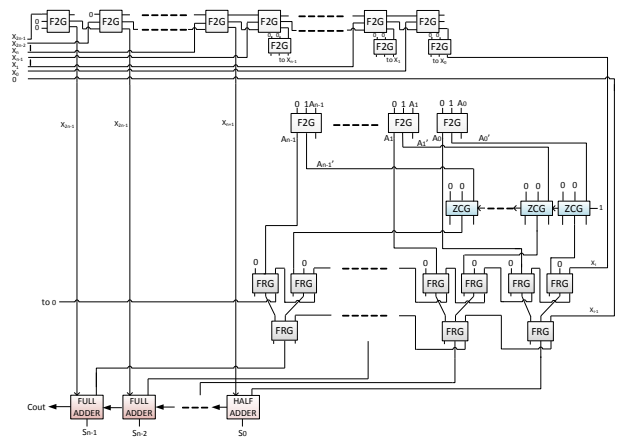


Figure 12. Generalized structure of proposed $n \times n$ Booth's-based multiplier

The second proposed signed serial multiplier is based on the K-algorithm reported in literature [14] which is an improved version of the Booth's algorithm, as stated in section 3.2.1. This multiplier, depicted in Figure 13, is implemented as follows for 4-bit operands:

- (1) Similar to that of the first proposed multiplier, 12 F2Gs are used for parallel shifter. However, this section has been implemented by eight FRGs and five F2Gs in literature [14].
- (2) The implementation of 4-bit parity-preserving 2 to 1 multiplexer requires four FRGs since each one-bit 2 to 1 multiplexer can be realized by a FRG.
- (3) According to Figure 5, to obtain the one's complement of multiplicand using the XOR gates between the multiplexer and the adder, only two F2Gs are required similar to Figure 7 instead of four F2Gs used in literature [14]. It should be noted that the one's complement will be sent to the adder only when y_i equals '1'.

(4) The 4-bit adder of the second proposed multiplier includes four full adders implemented by four ZPLGs instead of using MIGs. The input carry of this adder is y_i so that the two's complement of multiplicand is finally used when y_i equals '1' according to Table 2 and Figure 5.

In addition to the sections described above, a F2G is required to produce $x_i \oplus x_{i-1}$ and two copies of x_i to be moved to the y_i by default. This gate is placed on top-right of Figure 13. Therefore, the second proposed signed serial multiplier includes 23 gates with the quantum cost of only 82 by utilizing new arrangements of some basic parity-preserving gates to realize the different parts of the multiplier.

To extend the size of second proposed multiplier to be used for larger operands, Equation (4) can be used to compute the number of different gates and total quantum cost. The generalized circuit of proposed multiplier based on the K-algorithm is shown in Figure 14 for n -bit operands.

Required gates of $(n \times n)$ multiplier

$$= (3n + 1 + [n/2]) \times F2G + n \times FRG + n \times ZPLG \quad (4)$$

4. 2. Unsigned Multiplier

The third proposed multiplier in this paper is an unsigned serial multiplier based on the Add & Shift method. Due to the fact that this multiplier is unsigned, it naturally has less complexity compared to the first and second proposed multipliers. In this method, according to the least significant bit of the second operand, only two situations may occur. If this bit equals zero, "0000" will be sent to the adder, otherwise if it equals one, the multiplicand (A) will be sent to the adder. This multiplier, depicted in Figure 15, is implemented as follows for 4-bit operands:

- (1) Similar to previous proposed multipliers in this paper, 12 F2Gs are used for parallel shifter.
- (2) Similar to the second proposed multiplier, the implementation of 4-bit 2 to 1 multiplexer requires four FRGs. The selection is made between the multiplicand and the 4-bit zero number.

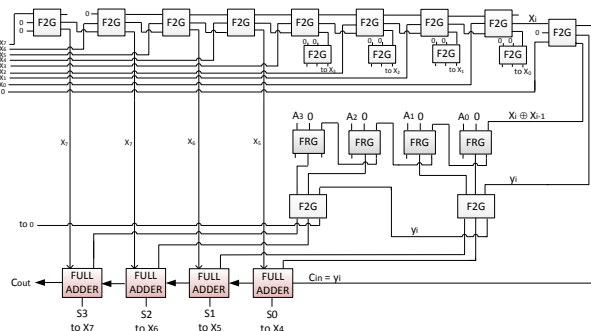


Figure 13. Proposed 4-bit parity-preserving multiplier based on the K-algorithm

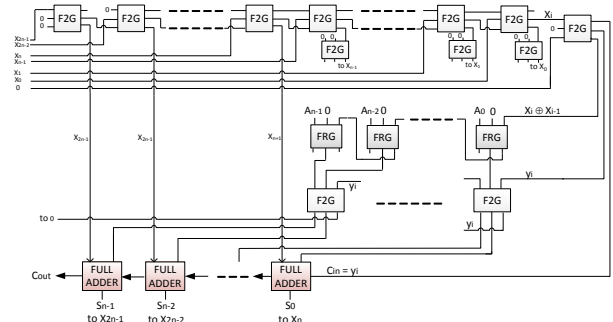


Figure 14. Generalized structure of proposed $n \times n$ multiplier based on the K-algorithm

(3) The 4-bit adder of the third proposed multiplier includes a half adder in the least significant bit and three other full adders. Thus, it is constructed by a ZCG as a half adder and three ZPLGs as full adders. This multiplier that includes 20 gates requires the quantum cost of 74 which is lower than that of the previous multipliers. To extend the size of third proposed multiplier for larger operands, Equation (5) is useful to compute the number of different gates and total quantum cost. The generalized circuit of proposed multiplier based on the Add & Shift method is shown in Figure 16 for n -bit operands.

Required gates of $(n \times n)$ multiplier

$$= 3n \times F2G + n \times FRG + 1 \times ZCG + (n - 1) \times ZPLG \quad (5)$$

5. PROPOSED PARITY-PRESERVING PARALLEL MULTIPLIER

The proposed parity-preserving parallel multiplier in this paper is a signed array multiplier based on the Baugh-Wooley method. A sample 4-bit multiplication regarding this method is shown in Figure 17. In this figure, P_{ij} is the complement of P_{ij} , and X_3 , Y_3 and Z_7 are the sign bits of two input operands and output product, respectively.

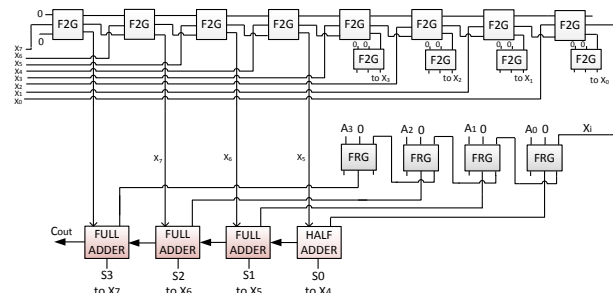


Figure 15. Proposed 4-bit parity-preserving multiplier based on the Add & Shift method

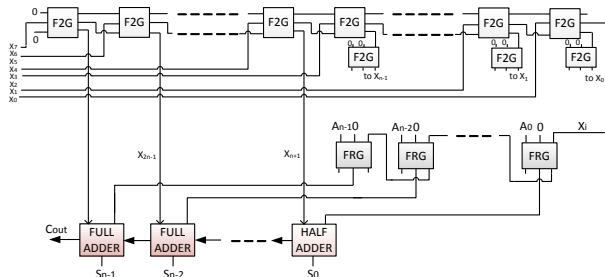


Figure 16. Generalized structure of proposed $n \times n$ multiplier based on the Add & Shift method

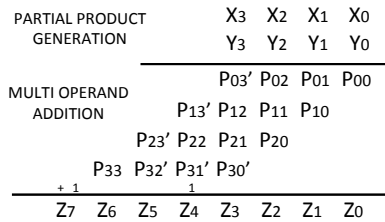


Figure 17. A 4×4 signed multiplication based on the Baugh-Wooley method

In addition, P_{ij} stands for $X_j Y_i$ that can be produced by an AND gate in the partial product generation part. According to literature [25], Figure 18 can be used for a low-cost PPG part of a 4×4 multiplier which includes seven FRGs and nine LMH gates with the total quantum cost of 89. In this figure, LMH gates receive two operands as inputs, and generate a copy of both input operands and their corresponding one-bit partial product P_{ij} , as well. However, based on Figure 17, some P_{ij} signals should be inverted to comply with the signed multiplication. Therefore, some F2Gs should be utilized similar to Figure 7 to produce the required inverted values. For a 4×4 multiplier, three F2Gs are enough to yield six P_{ij}' signals shown in Figure 17, as depicted in Figure 19.

The second part of an array multiplier is the multi-operand addition. To implement this circuit in the proposed design, ZCG is used as half adder and ZPLG is used as full adder, according to Figure 20. Since these gates have the quantum cost of six and eight, respectively, the quantum cost of multi-operand addition circuit shown in Figure 20 is equal to 92 including the single F2G. This F2G is responsible to produce the correct MSB of the product based on the Baugh-Wooley method. Therefore, the total quantum cost of this 4×4 multiplier including the circuits shown in Figs. 18 to 20 is equal to 187.

6. RESULTS AND DISCUSSION

In this section, some comparisons will be performed between the proposed parity-preserving multipliers and

their previous counterparts. To perform precise comparisons, the main criterions are used including gate count, number of constant inputs, number of garbage outputs, quantum cost, and hardware complexity. The gate count is the number of required gates to realize a circuit. In addition, the number of constant inputs in each circuit is the number of gates' inputs whose values should be constant at either '0' or '1' to perform the intended functions. However, the number of garbage outputs is the number of gates' outputs in the whole design that are not connected to the other gates or are not used as the outputs of the circuit.

The proposed parity-preserving serial multipliers are characterized in Table 3 along with the previous designs. The only existing parity-preserving serial multipliers were proposed in literature [14], so the comparisons are made with these circuits in Table 3. According to this table, the first and second proposed serial multipliers which are based on the Booth's algorithm and the K-algorithm, respectively, are better than their previous equivalent designs presented in literature [14] in all criteria.

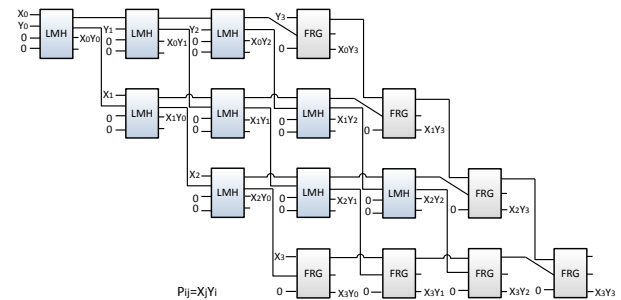


Figure 18. Partial product generation for the 4×4 array multiplier

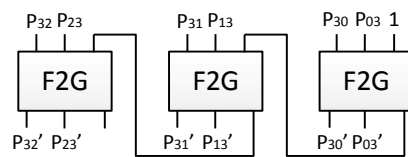


Figure 19. Inverted one-bit partial products as required in Figure 17

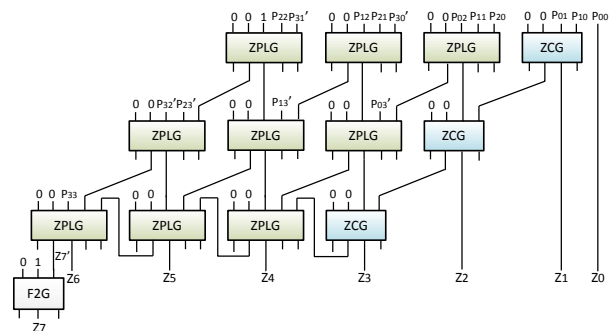


Figure 20. Proposed multi-operand addition for the 4×4 signed array multiplier

In addition, the third proposed serial multiplier which is based on the Add & Shift method is the only unsigned multiplier in Table 3, and has the best values based on the mentioned criteria.

It should be noted that the number of constant inputs and garbage outputs in the proposed multipliers can simply be obtained regarding their corresponding figures shown before. However, the C_{out} signal of the adder part in the first and second proposed multipliers (Figures 11 and 13) is accounted as a garbage output due to the fact that it is not used for the multiplication process. Furthermore, the number of constant inputs and garbage outputs of half adders and full adders are apparent since ZCG and ZPLG are used in the adder parts of proposed serial multipliers similar to the adder shown in Figure 9. Regarding the hardware complexity, this criterion for each circuit is calculated by summing the hardware complexity of all the gates constructing the circuit.

To illustrate the precise amounts of improvements attained by the new proposed multipliers, Figure 21 depicts the percentages of reduction in four different criterions for the first and second 4-bit proposed multipliers compared to their older Booth's-based and K-based counterparts proposed in literature [14], respectively. According to this figure, the maximum improvements are obtained for the second proposed multiplier compared to its K-based counterpart in

literature [14] based on the gate count and quantum cost that are 41% and 34.9%, respectively.

The amounts of required gate count and quantum cost for larger multipliers are illustrated in Table 4 according to their general formula. Similar to that of Table 3, the third proposed serial multiplier which is based on the Add & shift method requires the least gate count and quantum cost for all the adder sizes. Furthermore, Figure 22 depicts the percentages of reduction in the gate count and quantum cost of the first and second proposed multipliers compared to the Booth's-based and K-based counterparts proposed in literature [14], respectively, for 8x8 and 16x16 multipliers. According to this figure, the amounts of improvements are almost the same for a specific proposed multiplier with different sizes. However, the amounts of improvements are slightly increasing for the first proposed multiplier while the size of multiplier is increasing. The reverse of this characteristic is true for the second proposed multiplier.

Table 5 demonstrates comparative results of different parallel multipliers including the fourth proposed parity-preserving multiplier which is based on the Baugh-Wooley method along with the previous parity-preserving designs. In this table, the designs from [8] and [25] are unsigned array multipliers while the design from [9] is the only existing parity-preserving signed multiplier.

TABLE 3. Comparison of different parity-preserving serial multipliers

4x4 multiplier	Base algorithm	Signed	Gate count	Constant inputs	Garbage outputs	Quantum cost	Hardware complexity
[14] (based on Figure 4)	Booth's	Yes	44	52	61	200	$99\alpha+98\beta+30\gamma$
1 st proposed circuit (Figure 11)	Booth's	Yes	36	44	48	146	$105\alpha+67\beta+20\gamma$
[14] (based on Figure 5)	K-alg.	Yes	39	30	34	126	$82\alpha+60\beta+20\gamma$
2 nd proposed circuit (Figure 13)	K-alg.	Yes	23	24	28	82	$70\alpha+28\beta+8\gamma$
3 rd proposed circuit (Figure 15)	Add & Shift	No	20	21	25	74	$61\alpha+27\beta+8\gamma$

TABLE 4. Comparison of larger serial multipliers and their general formula

Multiplier design	Main formula	8x8 multiplier		16x16 multiplier	
		Gate count	Quantum cost	Gate count	Quantum cost
[14] (based on Figure 4)	$(3n + \lceil n/2 \rceil) \times F2G + (5n - 1) \times FRG + (3n - 1) \times MIG$	90	412	182	836
1 st proposed circuit (Figure 11)	$4n \times F2G + 3n \times FRG + (n + 1) \times ZCG + (n - 1) \times ZPLG$	72	294	144	590
[14] (based on Figure 5)	$(5n + 3) \times F2G + 3n \times FRG + n \times NFT$	75	246	147	486
2 nd proposed circuit (Figure 13)	$(3n + 1 + \lceil n/2 \rceil) \times F2G + n \times FRG + n \times ZPLG$	45	162	89	322
3 rd proposed circuit (Figure 15)	$3n \times F2G + n \times FRG + 1 \times ZCG + (n - 1) \times ZPLG$	40	150	80	302

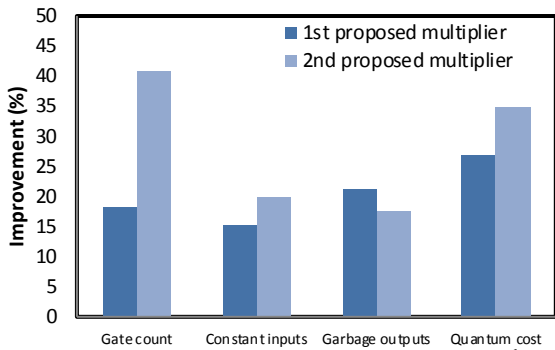


Figure 21. Improvements obtained in the 1st and 2nd 4-bit proposed multipliers compared to the designs in [14]

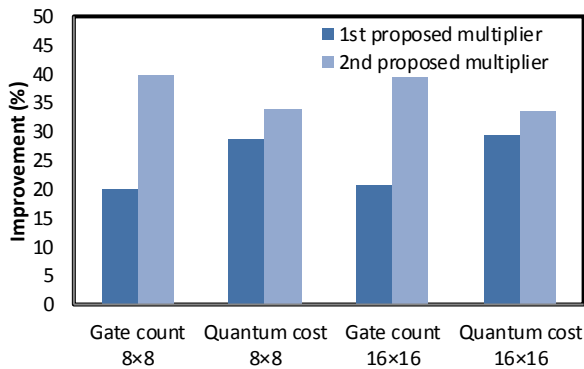


Figure 22. Improvements of the larger 1st and 2nd proposed multipliers compared to the designs in [14]

Based on this table, the fourth proposed multiplier in this paper requires the least quantum cost and gate count compared to previous designs while it is better than its

nearest counterpart proposed in literature [9] in all criteria. Figure 23 illustrates the percentages of improvements attained by the fourth proposed multiplier in four different criteria compared to the only existing signed multiplier proposed in literature [9] and the best unsigned multiplier proposed in literature [25] for the 4x4 size. According to this figure, the fourth proposed multiplier is better than previous designs except in the number of constant inputs and garbage outputs compared to reported data in literature [25]. Regarding this fact, it should be noted that signed multipliers naturally require more cost in comparison with unsigned multipliers. However, the fourth proposed design in this paper only requires more constant inputs and garbage outputs compared to reported data in literature [25], and it is better in the other criteria especially in the quantum cost that is a more important criterion.

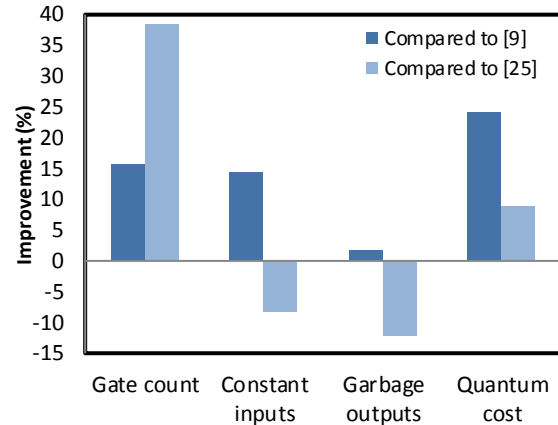


Figure 23. Improvements of the 4th proposed multiplier compared to previous designs

TABLE 5. Comparison of different parity-preserving parallel multipliers

4x4 multiplier	Base algorithm	Signed	Gate count	Constant inputs	Garbage outputs	Quantum cost	Hardware complexity
[8]	Array	No	48	64	64	244	$116\alpha+104\beta+36\gamma$
[25]	Array	No	52	49	49	205	$125\alpha+78\beta+36\gamma$
[9]	Baugh-Wooley	Yes	38	61	56	247	$121\alpha+109\beta+43\gamma$
4 th proposed circuit (Figures 18 to 20)	Baugh-Wooley	Yes	32	53	55	187	$136\alpha+79\beta+28\gamma$

7. CONCLUSIONS

In this paper, three novel low-cost reversible serial multipliers were proposed along with a new parallel multiplier with the parity-preserving capability. Since attaining the low-cost designs useful for error detection was the main goal of this paper, some techniques were

used including new arrangements of parity-preserving reversible gates, better utilization of existing reversible gates, and exploiting newer gates. This way, the low-cost signed and unsigned serial multipliers were proposed for cost-critical applications in which if only unsigned numbers exist, the third proposed multiplier can be used as the best design. On the other hand, the fourth proposed

design as a signed parallel multiplier which is based on Baugh-Wooley method can be used in the applications in which the speed is more unsigned numbers exist, the third proposed multiplier important. In addition to the basic 4-bit designs, larger serial multipliers were designed and investigated respecting the main criteria used in reversible logic circuits. The proposed multipliers with different sizes are evidently superior in comparison with the existing designs with respect to different criteria especially the quantum cost and gate count. For example, the second proposed multiplier with different sizes achieved to around 35 and 40% improvements in the quantum cost and gate count, respectively, compared to the existing counterpart.

8. REFERENCES

- Landauer, R., "Irreversibility and heat generation in the computing process", *IBM Journal of Research and Development*, Vol. 5, No. 3, (1961), 183–191.
- Bennet, C., "Logical reversibility of computation", *IBM Journal of Research and Development*, Vol. 17, No. 6, (1973), 525–532.
- Toffoli, T., "Reversible Computing", *Tech. memo MIT/LCS/TM-151, MIT Lab. for Computer Science*, (1980).
- Perkowski, M., Al-Rabadi, A., Kerntopf, P., Buller, A., Chrzanowska-Jeske, M., Mishchenko, A., et al., "A general decomposition for reversible logic", *Proc. RM*, (2001), 119–138.
- Zhou, R., Shi, Y., Wanga, H. and Cao, J., "Transistor realization of reversible 'ZS' series gates and reversible array multiplier", *Microelectronics Journal*, Vol. 42, (2011), 305–315.
- Pouraliakbar, E., Haghparast, M. and Navi, K., "Novel design of a fast reversible Wallace sign multiplier circuit in nanotechnology", *Microelectronics Journal*, Vol. 42, (2011), 973–981.
- Moghadam, M. Z. and Navi, K., "Ultra-area-efficient reversible multiplier", *Microelectronics Journal*, Vol. 43, (2012), 377–385.
- Babazadeh, S. and Haghparast, M., "Design of a nanometric fault tolerant reversible multiplier circuit", *Journal of basic and applied scientific research*, Vol. 2, No. 2, (2012), 1355–1361.
- Qi, X. and Chen, F., "Design of fast fault tolerant reversible signed multiplier", *International Journal of the Physical Sciences*, Vol. 7, No. 17, (2012), 2506–2514.
- Hatkar, A. P., Hatkar, A. A. and Narkhede, N. P., "ASIC Design of Reversible Multiplier Circuit", *Proc. of Intl. Conf. on Electronic Systems, Signal Processing and Computing Technologies*, (2014), 47–52.
- Kotiyal, S., Thapliyal, H. and Ranganathan, N., "Circuit for Reversible Quantum Multiplier Based on Binary Tree Optimizing Ancilla and Garbage Bits", *Proc. of 27th Intl. Conf. on VLSI Design (VLSID)*, (2014), 545–550.
- Nagamani, A. N., Nikhil, R., Nagaraj, M. and Agrawal, V. K., "Reversible radix-4 booth multiplier for DSP applications", *Intl. Conf. on Signal Processing and Communication (SPCOM)*, (2016), 1–5.
- Gowthami, P. and Satyanarayana, R. V. S., "Design of an efficient multiplier using Vedic mathematics and reversible logic", *IEEE Intl. Conf. on Computational Intelligence and Computing Research (ICIC)*, (2016), 1–4.
- Bhardwaj, K. and Deshpande, M., "K-Algorithm: An Improved Booth's Recoding for Optimal Fault-Tolerant Reversible Multiplier", *26th Intl. Conf. on VLSI Design*, (2013), 362–367.
- Baugh, C. R. and Wooley, B. A., "A two's complement parallel array multiplication algorithm", *IEEE Transactions Computers*, Vol. 22, No. 12, (1973), 1045–1047.
- Przigoda, N., Dueck, G., Wille, R. and Drechsler, R., "Fault detection in parity preserving reversible circuits", *IEEE 46th Intl. Symp. on Multiple-Valued Logic (ISMVL)*, (2016), 44–49.
- Gaur, H. M., Singh, A. K. and Ghanekar, U., "Testable Design of Reversible Circuits using Parity Preserving Gates", *IEEE Design & Test*, In Press, Publication date: 09 Nov. 2017.
- Maslov, D. and Dueck, G. W., "Reversible cascades with minimal garbage", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 23, No. 11, (2004), 1497–1509.
- Biswas, A. K., Hasan, M. M., Chowdhury, A. R. and Babu, H. M. H., "Efficient approaches for designing reversible Binary Coded Decimal adders", *Microelectronics Journal*, Vol. 39, (2008), 1693–1703.
- Valinataj, M., Mirshekar, M. and Jazayeri, H., "Novel low-cost and fault-tolerant reversible logic adders", *Computers and Electrical Engineering*, Vol. 53, (2016), 56–72.
- Parhami, B., "Fault-tolerant reversible circuits", *40th Asilomar Conference on Signals, Systems and Computers (ACSSC)*, (2006), 1726–1729.
- Fredkin, E. and Toffoli, T., "Conservative logic", *International Journal of Theoretical Physics*, Vol. 21, (1982), 219–253.
- Haggarast, M. and Navi, K., "A Novel Fault Tolerant Reversible Gate for Nanotechnology Based System", *American Journal of Applied Sciences*, Vol. 5, No. 5, (2008), 519–523.
- Islam, M. S., Rahman, M. M., Begum, Z. and Hafiz, M. Z., "Fault tolerant reversible logic synthesis: carry look-ahead and carry skip adders", *Intl. Conf. on Advances in Computational Tools for Engineering Applications (ACTEA)*, (2009), 396–401.
- Jamal, L., Rahman, M. M. and Babu, H. M. H., "An optimal design of a fault tolerant reversible multiplier", *IEEE 26th Intl. SOC Conf. (SOCC)*, (2013), 37–42.
- Zhou, R. G., Li, Y.-C. and Zhang, M.-Q., "Novel design for fault tolerant reversible binary coded decimal adders", *International Journal of Electronics*, Vol. 101, No. 10, (2014), 1336–1356.
- Mitra, S. K. and Chowdhury, A. R., "Minimum cost fault tolerant adder circuits in reversible logic synthesis", *25th IEEE Intl. Conf. VLSI Design (VLSID)*, (2012), 334–339.
- Moallem, P. and Ehsanpour, M., "A novel design of reversible multiplier circuit", *International Journal of Engineering-Transactions C: Aspects*, Vol. 26, No. 6, (2013), 577–586.
- Madhulika, C., Nayak, V. S. P., Prasanth, C. and Praveen, T. H. S., "Design of systolic array multiplier circuit using reversible logic", *IEEE Intl. Conf. on Recent Trends in Electronics Information & Communication Technology*, (2017), 1670–1673.
- Nagamani, A. N., Kumar, S. S. and Agrawal, V. K., "Design of garbage free reversible multiplier for low power applications", *4th Intl. Conf. on Power, Control & Embedded Systems*, (2017), 1–4.
- Zoka, S. and Gholami, M., "Two novel D-flip flops with level triggered reset in quantum dot cellular automata technology", *International Journal of Engineering-Transactions C: Aspects*, Vol. 31, No. 3, (2018), 415–421.
- Valinataj, M., "Novel parity-preserving reversible logic array multipliers", *Journal of Supercomputing*, Vol. 73, No. 11, (2017), 4843–4867.
- Eslami-Chalandar, F., Valinataj, M. and Jazayeri, H., "Design of error detecting serial multipliers in reversible logic", *Electronics Industries Quarterly (EIQ)*, Vol. 8, No. 1, (2017), 99–110.

Reversible Logic Multipliers: Novel Low-cost Parity-Preserving Designs

F. Eslami-Chalandar, M. Valinataj, H. Jazayeri

School of Electrical and Computer Engineering, Babol Noshirvani University of Technology, Babol, Iran

P A P E R I N F O

چکیده

Paper history:

Received 05 April 2018

Received in revised form 27 February 2019

Accepted 07 March 2019

Keywords:

Reversible Logic

Parity-Preserving Gates

Multiplication

Booth's Algorithm

Error Detection

Fault-Tolerance

منطق برگشت‌پذیر یکی از نمونه‌های نوظهور برای بهینه‌سازی توان مصرفی است که می‌تواند به جای مدارهای فعلی مورد استفاده قرار گیرد. همچنین، تحمل‌پذیری اشکال به صورت تشخیص یا تصحیح خطا جنبه‌ای ضروری برای سیستم‌های پردازشی امروزی است. در این مقاله، به خاطر اهمیت عملیات ضرب در سیستم‌های محاسباتی، چندین طراحی جدید برای ضرب‌کننده برگشت‌پذیر با ویژگی حفظ پاریتی پیشنهاد می‌شوند که برای تشخیص خطا مناسب خواهند بود. در ابتدا، دو ضرب‌کننده سری علامت‌دار بهینه بر مبنای الگوریتم بوث و نسخه بهبودیافته آن به نام الگوریتم K، با استفاده از چینش‌هایی جدید برای گیت‌های برگشت‌پذیر ارائه می‌گردند. سپس، ضرب‌کننده سری کم‌هزینه دیگری بر پایه روش مرسوم جمع-انتقال پیشنهاد می‌شود که برای کاربردهایی شامل ضرب اعداد بدون علامت مناسب است. در انتها، یک ضرب‌کننده موازی علامت‌دار جدید بر پایه روش با-وولی پیشنهاد می‌گردد که برای کاربردهای نیازمند به سرعت بالا مناسب است. نتایج مقایسه‌ها نشان می‌دهد که ضرب‌کننده‌های پیشنهادی با توجه به معیارهای اصلی مورد استفاده در مدارهای با منطق برگشت‌پذیر شامل هزینه کوانتومی، تعداد گیت، تعداد ورودی‌های ثابت و تعداد خروجی‌های بی‌استفاده، بسیار بهتر از طراحی‌های موجود هستند.

doi: 10.5829/ije.2019.32.03c.05