Design of a Novel Fault Tolerant Reversible Montgomery Multiplier in Nanotechnology

1Ali Hatam and 2Majid Haghparast

1Department of Computer Engineering, Tabriz Branch, Islamic Azad University, Tabriz, Iran. 
2Department of Computer Engineering, Shahre-Rey Branch, Islamic Azad University, Tehran, Iran.

Abstract: Reversible logic has received great attention in recent years due to its ability to reduce the power dissipation which is the main requirement in low power VLSI design (Kocher, 1999). It has many applications in low power CMOS Design, quantum computing, DNA computing, bioinformatics, and nanotechnology. Low power consumption is a major issue in VLSI circuits today. This paper proposes fault tolerant reversible Montgomery multiplier for the first time. The proposed fault tolerant reversible multiplier circuit can multiply two n-bit binary numbers. This paper deals with fault tolerant for binary reversible circuits. The proposed circuits can be used to construct a more complex system in nanotechnology.

Key words: Reversible Logic, Fault Tolerant, Montgomery multiplier, Quantum computing, Nanotechnology based systems.

INTRODUCTION

Reversible circuits are fundamentally different from traditional irreversible ones. A gate (circuit) is reversible if and only if its input vector can be exclusively recovered from the output vector. There is a one-to-one correspondence between its input and output assignments. Therefore the number of inputs and outputs in a reversible gate or circuit are equal. Such gates allow the reproduction of the inputs from observed outputs and we can determine the inputs from the outputs. In reversible Logic if input vector cannot be recovered from its output vector, then information is lost. Therefore in reversible logics, no information is lost. A system is Fault tolerant if and only if could be enables to continue operating correctly in the event of the failure of some its components. Therefore in the fault tolerant systems, the detection and correction of failure is simple (Majid Haghparast, 2010).

There is a number of commonly used fault tolerant reversible logic gate such as FRG (Fredkin, 1982), F2G (Parhami, 2006) NFT gates as which are shown in Fig. 1.2.6 respectively. In this paper we present a new Fault Tolerant Reversible Full Adder that can work singly as a full adder unit and its quantum cost is only 14 (Keyes, 1970; Landauer, 1961; Haghparast, 2008; Hassan, 2003). Nowadays Quantum computers are constructed using reversible logic circuits. It has widespread usages in low power CMOS and Optical information processing, DNA computing, quantum computation and nanotechnology. According to Landauer’s research (Keyes, 1970; Landauer, 1961) the amount of energy dissipated for every irreversible bit operation is at least kTln2 joules, where K=1.3806505*10⁻²³m²kgs⁻²K⁻¹ (Joule/Kelvin) is the Boltzmann’s constant and T is the temperature at which operation is performed (Bennett, 1973). If a gate is irreversible, it may result in heat generation and energy waste. Bennett (Barenco, 1995; Bennett, 1973) showed that zero energy dissipation would be possible if the network consists of reversible gates only. Parity preserving is one of the main methods for error detection and correction in digital systems. We have two major problems with reversible logic synthesis: Fan-out and Feedback (Loop) are not allowed. For these reasons the design of a fault tolerant reversible circuit is more difficult than traditional circuits.

In 1985 Montgomery (1985) introduced a new method for modular multiplication. The approach of Montgomery avoids the time consuming trial division that is a common bottleneck of other algorithms. This method is proved to be very efficient and is the basis of many implementations of modular multiplication. In this design three features of improved reversible circuit were met:

Corresponding Author: Majid Haghparast, Department of Computer Engineering, Shahr-Rey Branch, Islamic Azad University, Tehran, Iran. 
E-mail: haghparast@iausr.ac.ir
1. Using minimum number of reversible logic gates.
2. Using minimum number of garbage outputs.
3. Using minimum constant inputs.

Multiplication is a heavily used arithmetic operation in many computational units. It is necessary for the processors to have high speed multipliers. Optimization of a multiplier (Haghparast, 2009) in the abovementioned design is very important. In this paper, for the first time fault tolerant reversible multiplier circuit using fault tolerant reversible gates is presented.

2. Background:
A gate, a circuit or a function is reversible if and only if there is a one-to-one mapping between its input and output. Therefore, a reversible gate has an equal number of inputs and outputs and there is a one to one correspondence between the vector of inputs and outputs (Perkowski, 2001; Thapliyal, 2005).

Let the input vector be \( I \), output vector be \( O \) and they are called as follows, \( I = (I_1, I_2 \ldots I_{n-1}, I_n) \) and \( O = (O_1, O_2 \ldots O_{n-1}, O_n) \). And there is a one to one correspondence from \( I_n \) to \( O_n \).

Reversible Fault Tolerant and Quantum Gates, and Circuits:
Reversible logic gates can be implemented in various technologies such as CMOS circuits, optical circuits, and Nanotechnology. Some parity preserving reversible gates are also implemented in QCA (Majid Haghparast, 2008) recently and are used for fault testing (Haghparast, 2008).

FRG gate is depicted in Fig.1. FRG gate is 3-input, 3-output reversible gates. Furthermore, it is a one-through gate, which means that one input variable is also outputs.

Each FRG gate can be used as “Multiplexer” gate. A 2-input, n-bit reversible multiplexer requires n FRG gates and produces n garbage outputs. Its quantum cost (QC) is 5. The corresponding truth tables of FRG gate is depicted in the tables 1.

A3*3 Feynman Double Gate (F2G) is depicted in Fig.2. The F2G gate is a Feynman Gate with an additional input and one more output. The extra input and output, along with the control input “A” define a second controlled NOT operation (Vasudevan, 2004). It is a one-through gate and its quantum cost (QC) is only 2.

F2G can implement COPY functions. The implementation of F2G gate as COPY function is shown in Fig. 3. Fan-out and Feedback (Loop) are not allowed. The corresponding truth tables of F2G gate is depicted in table 2.

Our Proposed full adder (Haghparast, 2009) circuit can be realized three garbage outputs and only two constant inputs. The \( V^+ \) gate is the Hermitian conjugate (Barenco, 1995; Haghparast, 2009) of \( V \). The \( V \) and \( V^+ \) quantum gates function are shown in follow:

\[
V \times V = \text{NOT} \\
V^+ \times V^+ = \text{NOT} \\
V \times V^+ = V^+ \times V = I
\]

This function shows that \((V \times V)\) or \((V^+ \times V^+)\) gates in series are equivalent to a NOT gate. Also Hermitian conjugate “\(V^+\)” & “\(V\)” in series, the result is equivalent to a BUFFER gate (Haghparast, 2008; Kaye, 2007). The full adder circuit output equations \(S=A\oplus B\oplus Cin\) and \(Cout= (A\oplus B)\oplus AB\) produce the same output \(S=1\) and \(Cout = 0\), for the three distinct input combinations \(A = 0, B = 0, Cin = 1\); \(A = 0, B = 1, Cin = 0\); and \(A = 1, B = 0, Cin = 0\). Therefore, to separate all repeated values of outputs \(S\) and \(Cout\) we need three garbage outputs, as shown in Table 2. Thus the total number of outputs is \(2+3=5\). Since in a reversible circuit the number of inputs must be equal to the number of outputs and there are three inputs in a full adder circuit, A, B and Cin. Hence two inputs need to be constant input.

To implement an n-bit Full Adder, we must use N number of Fault Tolerant Reversible Full Adder gates. A 3*3 NFT Gate is depicted in Fig. 6. It can be defined as \(Iv= (A, B, C)\) and \(Ov= (P=A\oplus B, Q'=B\oplus C\oplus A\oplus C, R'=BC\oplus AC)\). Where \(Iv\) and \(Ov\) are the input and output vectors respectively. NFT can implement all Boolean functions. The implementation of NFT gate as the AND gate is shown in Fig 6 (a). We used this property in our proposed Fault Tolerant Multiplier gate. The corresponding truth tables of NFT gate is depicted in the tables 3. Its quantum cost (QC) is only 5.
Fault Tolerant reversible register: Figure 7(a) shows the implementation of reversible D Latch (Majid Haghparast, 2010). The parity preserving D Latch requires a FRG and an F2G gate. The reversible D Latch can be used to implement a reversible clocked D flip-flop.

Reversible D Flip-Flop (Majid Haghparast, 2010) using master slave Method. Each D Flip Flop contains one F2G gate and two Fault Tolerant reversible D Latch gate. Thus the proposed Fault Tolerant reversible D Flip Flop requires three F2G and two Fredkin gates and its QC is \((5+2)+2+(5+2)=16\). The D Flip–Flop structure and block diagram is shown in Fig 8(a) and 8(b).

The reversible D flip-flops can be used to implement a fault Tolerant reversible register. The n-bit reversible shift register is shown in Fig 9.

The diagram shows an n-bit shift register made from Fault Tolerant reversible master-slave D flip-flops. During each clock pulse, one bit is transmitted from left to right. In order to get the data out of the register, they must be shifted out serially. This shift register can be implemented either Serially (SISO) or Parallel (PIPO). Therefore, an n-bit reversible Fault Tolerant shift register can be realized by 5n gates, n+1 garbage outputs and its QC is at least 16n.

**Fault Tolerant Reversible Register (Majid Haghparast, 2010):**

Register is a quick storage available in our circuits. Each flip-flop can store one bit of information, a register with n D flip-flops can store n bits of information. Figure 10 shows the implementation of Fault Tolerant reversible clocked D flip-flop. The fault tolerant reversible D flip-flops can be used to implement a Fault tolerant reversible register. In reversible register, clock output of a Fredkin gate is connected to the clock input of the Fredkin gate of next D flip-flop. Each Fault tolerant reversible D flip-flop contains one FRG gate and one F2G gate. Therefore, an n-bit reversible Fault Tolerant register can be realized by 2n gates, 2n garbage outputs and its QC is at least 7n.

3. **Proposed Architecture of Fault Tolerant Reversible Montgomery Multiplier:**

Figure 11 shows the implementation of the Fault Tolerant reversible Multiplier using the Fault Tolerant reversible components. In the Figure, darker lines present multiple bits and dotted lines represent single bit. In this Section the Multiplication Algorithm is presented and an analysis of improvements on the algorithm is done. The algorithm of the Fault Tolerant Reversible Multiplier (FTRM) on two integers X and Y , with required parameters for N bits of precision, will result in the number \( Z = \text{FTRM}(X,Y) = XY \cdot r^{-1} \mod M \), where \( r=2^n \) and M is an integer in the range \( 2^{n-1} < M < 2^n \) such that \( \gcd(r,M) = 1 \) and \( 0 < X, Y < 2M \). Since \( r=2^n \), it is sufficient that the modulus M be an odd integer. For this application, M is usually a prime number or a product of primes, thus this condition is easily satisfied. Given two operands Y (multiplicand) and X (multiplier) and the modulus M, the algorithm presented in this section executes a series of operations to generate \( XY \cdot r^{-1} \mod M \), scanning Y and M word-by-word and scanning X bit-by-bit. This characteristic enables us to derive a hardware implementation that is very regular and based on simple operations. At the end of the execution of algorithm (or algorithm execution), the SUM Register stores the result \( XY \cdot 2^{-(n+2)} \mod M \).

**Fig. 1:** Fault Tolerant FRG Gate.

**Fig. 2:** Fault Tolerant F2G Gate.
Fig. 3: Fault Tolerant COPY (Fan-out) Gate.

Fig. 4: Our Proposed Quantum Fault Tolerant Reversible Full Adder. Its QC is only 14.

Fig. 5(a): Symbol of three-to-three Fault Tolerant Reversible Full Adder gate and its functionality.

Fig. 5(b): Schematic of n-bit parallel Fault Tolerant Reversible Full Adder.

Fig. 6(a): Fault Tolerant NFT Gate.
Fig. 7(a): Reversible Fault Tolerant D Latch structure.

Fig. 7(b): D Latch block diagram.

Fig. 8 (a): Reversible Fault Tolerant D Flip-Flop structure.

Fig. 8 (b): Reversible Fault Tolerant D Flip-Flop block diagram.

Fig. 9: Proposed Fault Tolerant Reversible Fault Tolerant n-bit shift register.

Fig. 10: N-bit Fault Tolerant Reversible registers.
Fig. 11: Proposed architecture of Fault Tolerant Reversible Montgomery Multiplier.

Table 1: Truth table of the parity preserving FRG.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>P</th>
<th>Q</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 2: Truth table of the parity preserving F2G.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>P</th>
<th>Q</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0*</td>
<td>0*</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1*</td>
<td>1*</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 2: Truth table of the Fault Tolerant Reversible Full Adder.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Cin</th>
<th>Sum</th>
<th>C_{out}</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 3: Truth table of the parity preserving NFT.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>P</th>
<th>Q</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0*</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0*</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0*</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1*</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Inputs: X, Y, M
Output: \(X \times Y \times 2^{(n-2)} \mod M\)
Xi: \(i^{th}\) bit of X

239
SUM0: LSB of SUM FTRM (X, Y, M)

**Step 1:**
Initialize SUM and CARRY to 0

**Step 2:**
Set i: = 0

**Step 3:**
Set CNS: = 0

**Step 4:**
Repeat Steps 5 to 9 while i ≤ n + 1

**Step 5:**
Perform carry save addition on SUM, CARRY and X_i·Y using reversible FTFA_1

**Step 6:**
The outputs (SUM, CARRY) from FTFA_1 are fed into FTFA_2. Add these outputs with SUM0·M using FTFA_2.

**Step 7:**
Store the results of above Step into FTR registers (SUM and CARRY).

**Step 8:**
Perform right shift operation on SUM and CARRY by 1 bit using F.T.R shift registers.

The results (SUM and CARRY) of these operations are feedback into the inputs of FTFA_1

**Step 9:**
Set i: = i + 1

**Step 10:**
Set CNS: = 1

**Step 11:**
Repeat Steps 12 and 13 while CARRY ≠ 0.

**Step 12:**
Perform carry save addition on SUM and CARRY using reversible FTFA_1 and FTFA_2.

**Step 13:**
Store the results of above Step into F.T.R registers (SUM and CARRY)

**Step 14:**
Return SUM = X·Y·2^{n+2} mod M.

5. **Theorem:**
The proposed n-bit Fault Tolerant Reversible Montgomery Multiplier can be realized by at least 9n+5 gates and 8n+8 garbage outputs with quantum cost of 104n+12.

**Proof:**
The proposed n-bit Fault Tolerant Reversible Montgomery Multiplier contains:

- Two NFT gates to compute AND-1 and AND-2 blocks, producing 2n garbage outputs, 5n quantum cost
• Two n-bit reversible mux, requiring 2n Fredkin gates, 2n garbage outputs, 10n quantum cost
• Two n-bit reversible PIPO shift registers, requiring 10 gates, 2(3n) garbage outputs, 16(n) quantum cost
• Two (n)-bit registers, requiring 2(2n) gates, 2(n) garbage outputs, 2(6n) quantum cost
• Two Fredkin gates for selections, producing 4garbage outputs with 10 quantum cost.
• 2n F2G gates to construct two COPY blocks, requiring quantum cost of 2(n)
• One F2G gate to get another copy of Sum0 with 2 quantum cost
• Two FTFA gates, producing 3(n) garbage outputs with 14(n) quantum cost

Therefore, the proposed architecture needs: 2(n)+5(n)+2(n)+2+2+1= 9n+5 Fault Tolerant Reversible gates
and produces 2(n)+5(n)+2(n) +3(n)+8 =8n+8 garbage outputs with quantum cost of 
2(5n)+2(10n)+ 2(16n)+2(6n)+2(14n)+2+2(n)+10 = 104n+12.

Conclusion:
In this paper, we present optimized hardware architecture using Fault Tolerant Reversible circuits to
implement Montgomery multiplication. Also this paper presents a new Fault Tolerant Reversible Full Adder
and presents its hardware complexity and quantum implementation. This circuit can work signily as a full adder
unit and it is efficient in terms of gate count, garbage outputs, constant input and quantum costs. A novel nxn
Fault Tolerant Reversible Full Adder circuit is using this gate and other Reversible gate has also been
presented. It has been demonstrated that the proposed nxn multiplier is efficient in terms of hardware
complexity, gate count, garbage output and constant input.

REFERENCES
Haghparast, M., M. Mohammadi, K. Navi and M. Eshghi, 2009. Optimized Reversible Multiplier Circuit,
Haghparast, M., K. Navi, 2008. A Novel Fault Tolerant Reversible Gate For Nanotechnology Based
Haghparast, M., K. Navi, 2008. Design of a Novel Fault Tolerant Reversible Full Adder For
Hassan Babu Hafiz Md and Md. Rafiqul Islam and Ahsan Raja Chowdhury and Syed Mostahed Ali
Chowdhury, On the realization of reversible full adder circuit -2003 -International Conference on Computer
and Information Technology- 2: 880-883.
Hassan Babu Hafiz Md and Md. Rafiqul Islam and Syed Mostahed Ali Chowdhury and Ahsan Raja
Hassan Babu Hafiz Md and Md. Rafiqul Islam and Syed Mostahed Ali Chowdhury and Ahsan Raja
5: 183-191.
44: 519-521.
Majid Haghparast, 2008. A New Low Power Dynamic Adder Cell Based on Majority Function •World


