Design and Optimization of Reversible BCD Adder/Subtractor Circuit for Quantum and Nanotechnology Based Systems

Majid Mohammadi, Mohammad Eshghi, Majid Haghparast and Abbas Bahrololoom

Faculty of Electrical and Computer Engineering, Shahid Beheshti University, Tehran, Iran
C.E. Department, Science and Research Branch, Islamic Azad University, Tehran, Iran
C.E. Department, Kerman University, Kerman, Iran

Abstract: Reversible logic circuits have found promising attention in nanotechnology, quantum computing and low power CMOS designs. In this paper we present a modular synthesis method to realize a reversible Binary Coded Decimal (BCD) adder/subtractor circuit. We use genetic algorithms and don’t care concept to design and optimize all parts of a BCD adder circuit in terms of number of garbage inputs/outputs and the quantum cost. We have also developed and used genetic algorithm-based synthesis software to design and optimize proper circuits for a reversible BCD adder/subtractor such as full adder, reversible 9's complement generator and reversible multiplexer. The results show improvement in the quantum cost, the number of garbage inputs and outputs.


INTRODUCTION

Decimal arithmetic has found promising uses in the financial and commercial applications. This is due to the precise calculations required in these applications as oppose to binary arithmetic where some of decimal fractions can not be represented precisely [1]. The software implementation of decimal arithmetic eliminates these conversion errors, but it is typically 100 to 1000 times slower than binary arithmetic. This attracts the attention of hardware designers to add a decimal arithmetic unit to CPUs to perform decimal calculations [2].

Energy loss on the other hand, is an important consideration in a binary arithmetic circuit. Part of the problem of energy dissipation is related to non-ideal of transistors and materials. Higher level of integration and the use of new fabrication processes have reduced the heat loss over the last decades. Another problem arises from Landauer’s principles [3] state that, logic computations that are not reversible, necessarily generate heat KT.In2 for every bits of information that is lost, where K is the Boltzmann’s constant and T is the temperature. Reversible are circuits (gates) in which there is a one-to-one mapping between vectors of inputs and outputs. The reversible designs do not lose any information.

Reversible logic circuits have found promising attention in nanotechnology [4], quantum computing and low power CMOS designs. Some reversible logic circuit synthesis methods are proposed in the literature [8, 14]. Genetic Algorithm (GA) is also used to synthesize and optimize a reversible or quantum logic circuit [15,12]. Don’t cares (DCs) can be managed in GA-based methods. Some works are also done on reversible BCD adder design and optimization [5, 6, 11, 13].

From the point of view of reversible logic circuits, there are some factors as the complexity measures of a circuit, namely, the number of the gates, the number of garbage inputs/outputs and the quantum cost. In this paper, a reversible Binary Coded Decimal (BCD) adder/subtractor has been designed and optimized by using GA and DC concept in the sense of above factors.

The structure of this paper is as follows: First we explain the background including BCD adders, reversible logic circuits, genetic algorithms and their application in the synthesis of circuits and DCs (Don’t Cares) in reversible and quantum logic circuits. Then the main contribution of paper is presented. We used the BCD adder to construct a BCD adder/subtractor. Conclusions and references are also provided.
MATERIALS AND METHODS

Background: Before that we propose the method is used in this paper some background information is needed. These are information about BCD adders, reversible logic, using genetic algorithms to synthesize a reversible circuit and the DC concept.

BCD adders: Figure 1 illustrates three parts of a BCD adder: 4bit binary adder, over 9 detection unit and correction unit. The first part is a binary adder which performs on two four-bit BCD digits and a one-bit carry input. In the second part, the over-9-detector recognizes if the result of the first part is more than 9 or not. Finally, in the third part, if the output of detector (P flag) is ‘1’, the sum is added by 6, else do nothing. A conventional BCD adder is shown in Fig. 2.

The 4bit binary adder is cascade of 4 FAs (4-bit carry-propagate adder). The detection part in Fig. 2 is constructed by using two AND gates (A1, A2) and one OR gate. The correction unit adds ‘0’ to the binary number if the binary result is less than 10 and adds 6 to the binary result if it is more than 9.

A binary full adder is a basic circuit for designing binary arithmetic units such as n-bit binary adder, subtractor and multiplier. In the same sense a BCD adder/subtractor is a basic circuit for designing BCD arithmetic units such as BCD n-digit adder/subtractor, BCD multiplier and so on.

Reversible gates and circuits: A function or a circuit is reversible if there is a one-to-one correspondence between its input and output assignments [13]. Classical reversible logic gates can be implemented in various technologies such as CMOS, optical and nanotechnology. Quantum gates, on the other hand, act on qubits. A qubit is a unit of quantum information. Generally, quantum gates cannot be implemented using conventional technologies such as CMOS.

A reversible gate has an equal number of inputs and outputs. Generally, with n inputs, there exist \(2^n\) reversible gates. The well-known 2x2 Feynman gate (Fig. 3a) operates as a controlled NOT (CNOT). If the control input of CNOT is set to ‘0’, the gate acts as a BUFFER gate; else, it acts as a NOT gate. The Feynman gate can be used as fanout gate to copy a signal. If the \(B\) input in Fig. 3a is set to ‘0’ then two outputs of the Feynman gate are \(A\) and \(C\).

Other reversible gates are also proposed in some papers [13]. Figure 4a and 4b show the New Gate (NG) and the Peres gates, respectively. Hafiz [5] designed a full adder using one NG and one Peres gates.

A reversible circuit is usually depicted using a series of connected gates on a number of parallel lines similar to the musical staff. These lines are the inputs/outputs of the circuit. The gates are placed on these parallel lines. Using this concept, design of a reversible circuit is similar to composing a piece of music. A synthesis method using this style usually results in smaller circuits with fewer garbage outputs [8].
The Quantum Cost (QC) of a reversible circuit is defined as the number of $1 \times 1$ or $2 \times 2$ reversible quantum or logic gates that are needed to realize the circuit [9]. For instance, the Toffoli gate is realized by minimum of five $2 \times 2$ gates, therefore, its QC is five [9]. The QC of a Fredkin gate is also 5.

Note that the Peres gate has QC of 4 while it has more functionality than Toffoli gate. Actually, it has one target output as the same as the Toffoli gate (larger circle in Fig. 4b) and one additional XOR output (filled circle in Fig. 4b). Peres gate is also universal and can be used in synthesis of reversible logic circuits.

Optimizing the reversible circuits using GA and don’t cares: Genetic algorithm is an optimization algorithm. We use it in the optimal synthesis of reversible logic circuits [15]. In this algorithm, variables of search space have to be coded to a string of bits, named chromosome. Each gate is coded as shown in Fig. 5. Many gates may be used in the synthesis. Therefore, we consider a field as the gate type and assign one code to each type of gate.

The second field represents the location of main output of the gate. The third up to the last fields are location of $r$ inputs of the gate. For some types of gates such as Peres, the third field is location of the XOR output of the gate.

Each chromosome represents a complete circuit and includes the codes of $m$ gates plus R1 to Rn, the constant inputs of the circuit (Fig. 6). This will be discussed in the latter paragraphs.

After defining the chromosome structure, we define a population of chromosomes and apply three basic operators of GA to chromosomes of this population. The crossover operator selects two chromosomes randomly and exchanges corresponding segments from them. The mutation operator applies random changes to the selected chromosome with a specific probability. The selection operator selects some of good chromosomes for reproduction of the next population. In the synthesis of a reversible circuit using GA, each generated chromosome is a coded circuit.

An important advantage of GA to other synthesis methods is that various universal gates can be used in the synthesis. For example one can use the Toffoli or Peres or Fredkin gates or a combination of them for the synthesis. Another choice may be a set of universal quantum gates such as V, V+ and Feynman gates. Based on GA synthesis algorithm, a software is developed in this research which can synthesize a given function by using each of mentioned gates separately, or simultaneously.

The definition of a reversible function would possibly include inputs, outputs or conditions whose values are not important. These are named don’t cares (DCs). In [12], the DCs in a reversible function or circuit are classified into three types: DC inputs, DC conditions and DC outputs. The first type is an additional bit that we add to the input part of the truth table of a logical function to retain the reversibility of the function. Traditionally, this type of DCs is named constant input because its value is not varying in the circuit. Different values of DC inputs will result to the different synthesized circuits. To assign the optimum values to these DCs, they must be inserted in the chromosome which defines the circuit (Fig. 6).

The second type of the DCs is DC condition. When the values of some rows of the truth table are not specified or are not important, they are assumed DC conditions. These occur when some DC inputs for the function are assumed, but this is not true in all cases. In the other words, a circuit may have DC conditions while it doesn’t have any DC inputs (such as a BCD adder in which input numbers over the 9 are DCs). The third type of DCs is DC output. It is additional output whose value is not important in the rows of the truth table. Conventionally, DC outputs are called garbage outputs. If the synthesis algorithm ignores these DCs, it generates a circuit that satisfies the care conditions or outputs. This will result to obtain a smaller circuit in less synthesis time.

As we will show in the next subsection, a reversible BCD Adder has all types of DCs. These are used to obtain an optimized circuit.

Design and optimization of the reversible BCD Adder: Referring to Fig. 1 three parts of a BCD adder have to be designed with reversible gates. These parts are 4bit binary adder, detection part and correction part. The correction part is a 4-bit binary adder, too.

Hafiz used two NG gates to design a full adder. Next, four FAs are used to construct a binary ripple carry 4-bit adder. For over-9-Detector he used a circuit depicted in Fig. 7.

The detector, which is designed specifically for decimal addition, can be designed more efficient with the lower cost and less number of DC outputs. In this work, we design the detection part by using Toffoli...
Fig. 7: Detector part designed by Hafiz

Fig. 8: BCD adder circuit

gates in order to reduce its number of DC outputs, as well as the quantum cost. Detector part is a circuit which has four inputs and one output, expressed in Eq.1.

\[
P = ((S_1 + S_2)S_3) + C_4
\]

(1)

\(S_1, S_2\) and \(S_3\) are three bits of the sum and \(C_4\) is the Carry out of the 4-bit binary adder in the first part of the BCD adder (Fig. 8). The Minimum number of required DC outputs is \(\lceil \log_2 q \rceil\), where \(q\) is the maximum number of repetition in the output patterns [10]. Truth table of the \(P\) equation shows that there are 11 repeated ‘1’s in the table; therefore, it requires at least \(\lceil \log_2 11 \rceil = 4\) DC conditions. The DC conditions are due to one DC input.

Since the DC input must be constant in the final circuit, this additional input is also named constant input in the literature.

Figure 9a illustrates the resulted circuit of the detector by means of Toffoli gates. The output \(P\) is the expected output of the detector. It uses three 3×3 and two 2×2 Toffoli gates. The QC of this circuit is 17.

Using Fredkin gates, a simpler circuit can be obtained. This circuit which is shown in Fig.9.b, uses only three Fredkin gates with quantum cost of 15.

To optimize the other parts of BCD adder, some optimized FAs can be used. For instance, in [8] an FA with four Toffoli and Feynman gates is proposed. In [12] various optimized designs of FA with and without quantum gates are proposed. An FA can also be designed by Peres gates. In this research, we have designed and optimized FA using GA and using quantum and Peres gates, shown in Fig. 10.

The first design uses 6 quantum 2×2 gates having total QC of 6. Another design uses only two Peres gates but its QC is 8.

New BCD subtractor: To calculate the BCD operation \(a-b\) we have to add \(a\) to 9’s complement of \(b\) with carry.
A₀ A₁ A₂ A₃
Q₀ Q₁ Q₂ Q₃

Fig. 11: 9’s complement circuit: A₃A₂A₁A₀ is input digit and Q₃Q₂Q₁Q₀ is output digit.

A₀ A₁ A₂ A₃
A₄
Q₀ Q₁ Q₂ Q₃
Gar

Gar

Gar

A₄

A₀

A₁

A₂

A₃

Q₀ Q₁ Q₂ Q₃
Gar

Gar

Gar

Fig. 12: Implementation of a multiplexer using Fredkin gates. A₃A₂A₁A₀ is input digit and A₇A₆A₅A₄ is 9’s complement of input digit and Q₃Q₂Q₁Q₀ is output digit. A₈ is control input.

We obtain 93 or 109 values for QC of BCD adder/subtractor according to the FA design is used. Table 1 shows the summary of parameters of our two designs. This table also shows the Hafiz’s design for comparison. Note that Hafiz had only designed the BCD adder part of this circuit. We have added the other parts of our subtractor design to this design to show a comparison.

As the Table 1 shows, the number of garbage inputs and outputs is decreased. Our designs also show a good improvement in the Quantum Cost (QC) of circuit.

CONCLUSION

In this paper, the Genetic Algorithm (GA) and DCs (DCs) are used to optimize a binary coded decimal adder and subtractor. This circuit can further be used in a large reversible system as a module of BCD reversible adder/subtractor. We have also developed and used genetic algorithm-based synthesis software to design and optimize proper circuits for a reversible BCD adder/subtractor such as full adder, reversible 9’s complement generator and reversible multiplexer. The GA-based synthesis algorithm used in this paper has many advantages to the other synthesis methods. For instance, optimum or near optimum results can be obtained, many universal gates can be used for synthesis, quantum gates can be included and don’t cares can be handled.

Table 1 shows the summary of parameters of designs which we proposed in this paper. This table also shows the Hafiz’s design. Note that Hafiz had only designed the BCD adder part of this circuit. This table shows that all important parameters such as the quantum cost and the number of garbage inputs and outputs are reduced in the proposed circuits. In addition, our circuits can also do subtraction.

The reversible BCD adder/subtractor can be used as a basic circuit for constructing the other reversible BCD arithmetic circuits. For example, by cascading \( n \) blocks of this circuit, an \( n \) digit BCD adder and subtractor can be assembled. Some works are also done on BCD multiplication which can be extended to reversible form by using the proposed BCD adder.

All the proposed circuits are quantum and reversible circuits that can be used in nanotechnology based systems.

<table>
<thead>
<tr>
<th>Design</th>
<th>QC</th>
<th>G-in</th>
<th>G-out</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Hafiz [5]</td>
<td>136</td>
<td>21</td>
<td>26</td>
</tr>
<tr>
<td>2 FA in Fig. 10b</td>
<td>109</td>
<td>19</td>
<td>24</td>
</tr>
<tr>
<td>3 FA in Fig. 10a</td>
<td>93</td>
<td>19</td>
<td>24</td>
</tr>
</tbody>
</table>

Table 1: Summary parameters of designs

RESULTS AND DISCUSSION

We can calculate the QC of the overall design of BCD adder/subtractor. It is QC of BCD adder plus QC of the 9’s complement circuit plus QC of multiplexer. To implement the BCD adder we can use the FA designs of Fig. 10a or Fig. 10b. Then the QC of BCD adder is 8xFA+15+4. Thus the QC of BCD adder is 67 for design Fig. 10a and 83 for design Fig. 10b. Thus,
REFERENCES


