Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Systems Architecture 52 (2006) 272–282
JOURNAL OF SYSTEMS ARCHITECTURE
www.elsevier.com/locate/sysarc

Design of a compact reversible binary coded decimal adder circuit

Hafiz Md. Hasan Babu *, Ahsan Raja Chowdhury

Department of Computer Science and Engineering, University of Dhaka, Dhaka-1000, Bangladesh

Received 20 September 2004; received in revised form 25 April 2005; accepted 10 May 2005
Available online 11 July 2005

Abstract

Reversible logic is an emerging research area and getting remarkable interests over the past few years. Interest is sparked in reversible logic by its applications in several technologies, such as quantum, optical, thermodynamics and adiabatic CMOS. This paper represents a synthesis method to realize reversible binary coded decimal adder circuit. Firstly, a reversible full-adder circuit has been proposed that shows the improvement over the two existing circuits. A lower bound is also proposed for the reversible full-adder circuit on the number of garbage outputs (bits needed for reversibility, but not required for the output of the circuit). After that, a final improvement is presented for the reversible full-adder circuit. Finally, a new reversible circuit has been proposed, namely reversible binary coded decimal (BCD) adder, which is the first ever proposed in reversible logic synthesis. In the way to propose reversible BCD adder, a reversible *n*-bits parallel adder circuit is also shown. Lower bounds for the reversible BCD adder in terms of number of garbage outputs and number of reversible gates are also shown. Delay has also been calculated for each circuit.

Keywords: Reversible logic; Garbage output; Full adder; Parallel adder; Binary coded decimal

1. Introduction

Energy loss is an important consideration in digital design. Part of the problem of energy dissipation is related to non-identity of switches and materials. Higher level of integration and the use of new fabrication processes have dramatically reduced the heat loss over the last decades. Another problem arises from Landauer's principles [1,6] state that, logic computations that are not reversible, necessarily generate heat $kT * \ln 2$ for every bits of information that is lost, where k is the Boltzmann's constant and T is the temperature.

* Corresponding author. Tel.: +880 2 9670734; fax: +880 2 8615583.
E-mail addresses: hafizbabu@hotmail.com (H.Md. Hasan Babu), farhan717@yahoo.com (A.R. Chowdhury).

1383-7621/$ - see front matter
doi:10.1016/j.sysarc.2005.05.005

For room temperature T, the amount of dissipating heat is small (i.e. 2.9×10^{-21} J) but not negligible. The design that does not lose any information is called reversible. *Reversible* are circuits (gates) in which the number of inputs is equal to the number of outputs and there is a one-to-one mapping between vectors of inputs and outputs. More formally, a *reversible* logic gate is a k-input, k-output (denoted $k * k$) device that maps each possible input pattern into a unique output pattern [2–4].

While constructing reversible circuits, some restrictions should be strictly maintained [2,3]:

- Fan-out is not permitted.
- Loops are not permitted.

Traditional design methods use, among other criteria, the number of gates as complexity measure (sometimes taken with some specific weights reflecting are of the gate). From the point of view of reversible logic we have one more factor, which is more important than the number of gates used, namely the number of garbage outputs. The unutilized outputs from a reversible gate/circuit are called "garbage". Good synthesis methods always produce less number of garbage outputs, but sometimes garbage outputs are unavoidable. For example, a single output function of n variables will require at least $n - 1$ garbage outputs, since the reversibility necessitates an equal number of outputs and inputs.

In this paper, we have proposed a reversible circuit named as reversible binary coded decimal (BCD) adder that poses all the good features of reversible logic synthesis. We have succeeded to restrict the number of garbage outputs and number of reversible gates as few as possible. Through quantum cost is realizable for almost every reversible gates [17], it is avoided in this paper. There exist several synthesis methods for reversible functions realization and minimization [16,18,19], but all of them realize the straightforward function (from truth table). But our proposed circuit is the composition of two logic: 4-bit (n-bit) reversible adder and combinational circuit.

This paper is organized as follows: Section 2 provides the necessary background on reversible logic along with the examples of popular reversible logic gates. Section 3 provides the synthesis of reversible circuit which covers our first proposed full-adder circuit with two existing full-adder circuits [7,14], theory on the lower bound (minimum requirement) of garbage outputs for reversible full-adder circuit, another new full-adder design approach, a generalized algorithm to construct reversible full-adder circuit and comparative study among the reversible full-adder circuits. Proposed design technique for reversible BCD adder is thoroughly discussed in Section 4. The paper concludes with observations and suggestions for further study in Section 5.

2. Background of the reversible logic gates

In this section, we will describe all about reversible logic and different types of gates. Though it is briefly described about garbage outputs, this section presents more appropriate examples and figures.

Definition 2.1. *Garbage* is the number of outputs added to make an n-input k-output Boolean function ((n,k) function) reversible. In other sense, a reversible logic gate has an equal number of inputs and outputs ($k * k$) and all the outputs are not expected. Some of the outputs should be considered to make the circuit reversible and those unwanted outputs are known as *garbage* outputs.

Example 2.1. If we want to find the Exclusive-OR between two variables in reversible computation, the circuit will look like Fig. 1.

One extra output should be produced to make the circuit reversible and that unwanted output ($P = A$, marked as $*$) is known as garbage. Now we will define some popular reversible gates where most of them are used in our proposed design.

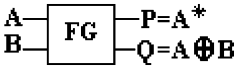

Fig. 1. Calculating exclusive-OR in reversible mode using Feynman gate (FG).

Definition 2.2. $1 * 1$ *NOT gate* is the simplest among all the reversible gates where the gate has only one input (A) and one output (B) such that $B = A'$.

Example 2.2. The truth table for the gate is given in Table 1.

Definition 2.3. Let I_v and O_v be the input and output vector of a $2 * 2$ *Feynman gate* (*FG*) [9,10] respectively, where $I_v = (A, B)$ and $O_v = (P = A, Q = A \oplus B)$.

Example 2.3. The block diagram for $2 * 2$ Feynman gate is shown in Fig. 1.

Definition 2.4. Let I_v and O_v be the input and output vector of a $3 * 3$ *Toffoli gate* (*TG*) [8,11] respectively, where $I_v = (A, B, C)$ and $O_v = (P = A, Q = B, R = AB \oplus C)$.

Example 2.4. Fig. 2 shows the $3 * 3$ Toffoli gate.

Definition 2.5. Let I_v and O_v be the input and output vector of a $3 * 3$ *Fredkin gate* [8,12] respectively, where $I_v = (A, B, C)$ and $O_v = (P = A, Q = A'B \oplus AC, R = A'C \oplus AB)$.

Example 2.5. Fig. 3 shows the block diagram of $3 * 3$ Fredkin gate.

Table 1
Truth table for $1 * 1$ NOT gate

A	B
0	1
1	0

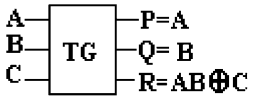

Fig. 2. $3 * 3$ Toffoli gate.

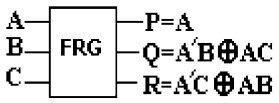

Fig. 3. $3 * 3$ Fredkin gate.

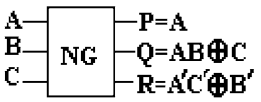

Fig. 4. $3 * 3$ New gate.

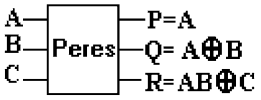

Fig. 5. $3 * 3$ Peres gate.

Definition 2.6. A $3 * 3$ *new gate* (*NG*) [7] can be defined as $I_v = (A, B, C)$ and $O_v = (P = A, Q = AB \oplus C, R = A'C' \oplus B')$, where I_v and O_v are the input and output vector respectively.

Example 2.6. The block diagram of a $3 * 3$ New gate is shown in Fig. 4.

Definition 2.7. A $3 * 3$ *Peres gate* (*Peres*) [15] can be defined as $I_v = (A, B, C)$ and $O_v = (P = A, Q = A \oplus B, R = AB \oplus C)$ where I_v and O_v are the input and output vector respectively.

Example 2.7. Fig. 5 shows the block diagram of $3 * 3$ Pares gate.

3. Synthesis of reversible circuits

The main differences of synthesizing a circuit with reversible gates, as compared to synthesizing a standard binary circuit [13] are the following:

- The number of outputs of a logic gate is equal to the number of inputs. It is easy to find solutions sacrificing one or more gate outputs for garbage, but such solutions are of less value.
- Every gate output that is not used as input to other gate or as a primary output is called garbage. A heavy price is paid for every garbage bit, if the garbage bit is left unattended, or if the mirror circuit and spy gates are added.
- In reversible logic, fan-out of any gate output is not allowed; every output can be used only once. Feynman gates can be used as "copying

circuits'', the same way as in the ''spy circuits'', to increase the fan-out. However, for every fan-out of two Feynman gate is used. Obviously, this increases the cost and delay.

3.1. Reversible full-adder circuit

We proposed a design scheme of a reversible full-adder circuit in [2], which is shown in Fig. 6. This circuit requires three reversible gates (one $3*3$ new gate, one $3*3$ Toffoli gate and one $2*2$ Feynman gate) and produces two garbage outputs. This design requires less number of gate and garbage outputs than [13,14,7] consecutively.

In this section, we will also show the improvement of our proposed previous reversible full-adder circuit. We construct the reversible full-adder circuit with the help of one New gate and one Peres gate [15]. The proposed circuit was shown in Fig. 7.

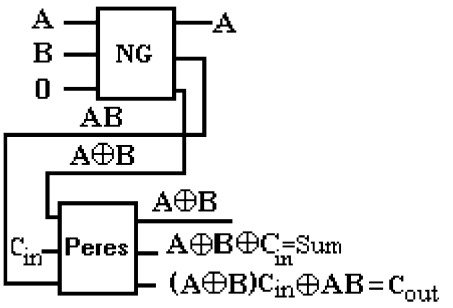

Fig. 7. Full-adder implementation using a $3*3$ new gate (NG) and a $3*3$ Peres gate (Peres).

3.2. Lower bound for the reversible full-adder circuit

In this section, we will present a property of reversible full-adder circuit.

Theorem 3.2.1. *A reversible full-adder circuit can be realized with at least two garbage outputs.*

Proof. In the full-adder circuit, there are three distinct combination $(0,0,1)$, $(0,1,0)$ and $(1,0,0)$ for the same output $(1,0)$. When we add the first garbage bit, two of the distinct input combinations produce the same output (1–0–0) or (1–0–1). Because, a garbage bit (0/1) can generate only two distinct combinations. Therefore, it is clear that the second bit is essential to produce the unique output combinations corresponding to its input combination. So, at least two garbage bits are required for a reversible full-adder circuit. □

3.3. Existing full-adder circuits

A full-adder implementation using two $3*3$ Toffoli gates and two $2*2$ Feynman gates is presented in [13,14], which is shown in Fig. 8. The circuit requires four reversible gates and produces two garbage outputs.

Another full-adder implementation using two $3*3$ new gates and one $2*2$ Feynman gate is presented in [7], which is shown in Fig. 9. The circuit requires three reversible gates and produces three garbage outputs. This design requires one less gate

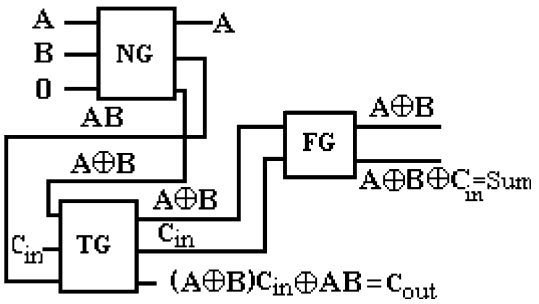

Fig. 6. Proposed full-adder implementation using one $3*3$ new gate (NG), one $3*3$ Toffoli gate (TG) and one $2*2$ Feynman gate (FG).

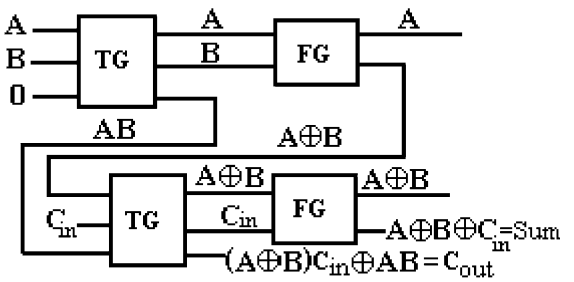

Fig. 8. Existing full-adder implementation using two $3*3$ Toffoli gates (TG) and two $2*2$ Feynman gates (FG) [13,14].

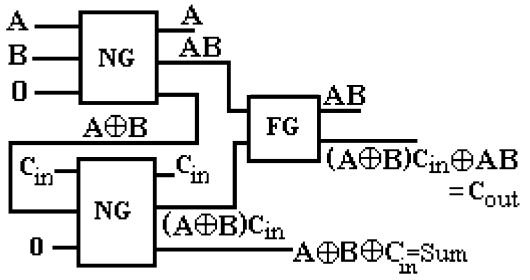

Fig. 9. Another full-adder implementation using two 3 * 3 new gates (NG) and one 2 * 2 Feynman gate (FG) [7].

but produces one more garbage output than that of previous design [13,14] (shown in Fig. 8).

3.4. Comparative study for the reversible full-adder circuits

Comparing with the existing circuit 1 (discussed in Section 3.3, Fig. 8) with our most recent proposed reversible circuit (Fig. 7), it is found that the proposed circuit produces same number of garbage outputs but requires two less number of gates. Unit delay is also improved, as the proposed circuit's unit delay is two (considering the output of a single gate as one unit delay). So, the proposed circuit is more efficient than the existing circuit [14].

Similarly, if we consider the existing circuit 2 (discussed in Section 3.3, Fig. 9), it is found that the proposed circuit requires less number of gates as well as produces less number of garbage outputs. Evaluation of the newly proposed circuit can be comprehended easily with the help of the comparative results presented in Table 2.

The following procedure (Algorithm 1) was considered to construct the reversible full-adder circuits in Figs. 6 and 7.

Table 2
Comparison of different reversible full-adder circuits

Referenced circuit	No. of gates	No. of garbage outputs	Unit delay
Proposed circuit 1 (Fig. 6)	3	2	3
Proposed circuit 2 (Fig. 7)	2	2	2
Existing circuit 1 [14]	4	2	4
Existing circuit 2 [7]	3	3	3

Algorithm 1. Construct_RFAC (Current_Solution, Best_Solution)

```
{
  if(Current_Solution is Better than Best Solution) {
        if(Current_Solution is Optimal) END;
        else
    Best_Solution = Current_Solution;
  }
  for count1 = 1 to Total_Gates                    {
    for count2 = 1 to Total_Combinations {
      Comb = Find_Best_Combination
      (Gate [count1], count2);
      Solution = Current_Solution      (Gate
      [count1], Comb);
      Construct_RFAC                   (Solution,
      Best_Solution);   }}
```

3.5. The n-bit reversible parallel adder

We can easily construct an n-bit parallel adder by cascading n reversible full-adder. Number of gates and number of garbage outputs will increase than 1-bit adder (full-adder) with proportion to n. Fig. 10 shows the n-bit parallel reversible adder.

Lemma 3.5.1. *The n-bit reversible parallel adder can be realized with at least* $2n$ *garbage output*, $2n$ 3×3 *reversible gates and* $2n$ *unit delay.*

4. Proposed reversible BCD adder

In this section, we will show all about our proposed reversible BCD adder with the details specification of each part of this adder. Algorithm 2 is the summarization of the design technique of reversible BCD adder.

4.1. Basic definitions and terminologies

In this section, we will discuss about the properties of a binary coded decimal (BCD) number and BCD adder.

Definition 4.1.1. Binary coded decimal or BCD representation uses four bits for each decimal digit,

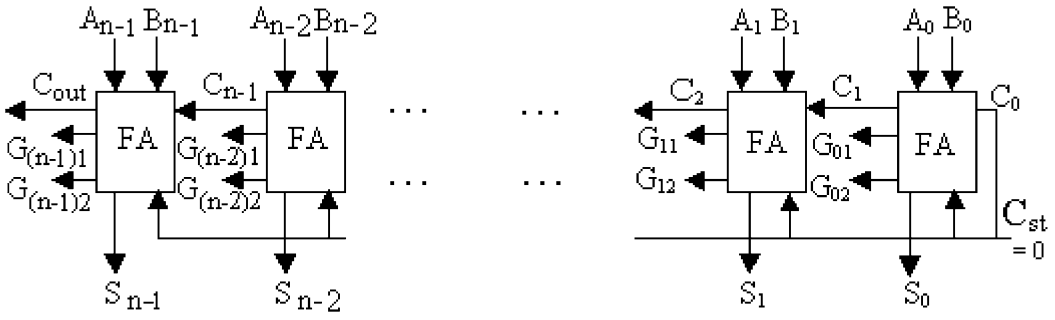

Fig. 10. The n-bit reversible parallel adder.

that is, the BCD code converts each decimal digit to binary digits individually [5]. Though 16 distinct digits can be represented in 4-bit binary numbers, only the first 10 numbers (0000 to 1001) are valid in BCD system.

Example 4.1.1. 0010 1001 is a valid BCD number that is equivalent to 29 in decimal system, but 0101 1010 is not valid as 1010 cannot be represented with the help of one decimal digit.

Definition 4.1.2. A BCD adder is a special type of adder that adds two BCD numbers and converts the result into its equivalent BCD number [5]. The conversion is required because of overflow of the addition.

Definition 4.1.3. A BCD adder is special type of adder that can only add two BCD numbers successfully.

Example 4.1.2. A BCD adder will produce 1001 after adding 0101 and 0100. In another case, the addition of 0101 and 0111 will be 1100, that is converted into 0001 0010 by BCD adder, as 1100 (12 in decimal) cannot be represented with one decimal digit.

For error correction purpose, a combinational logic should be used in a BCD adder that will control the overflow of the result of the addition of two BCD numbers. Let $A = A_3A_2A_1A_0$ and $B = B_3B_2B_1B_0$ be the two BCD numbers, $S_3S_2S_1S_0$ is the addition of A and B, and C_4 is carry of the addition. In the combinational logic, S_3, S_2, S_1 and C_4 should be checked for error correcting purpose. The equation, which will directly control the error-correcting module is $X = C_4 + (S_1 + S_2) \cdot S_3$. If X is 1, 0110 (decimal 6) should be added with the previous result ($S_3S_2S_1S_0$) to calculate the final result. In this case, a carry will be generated that directs the overflow of the addition. In another case, if X is 0, no modification will occur in the previous addition.

4.2. Synthesis algorithm of proposed circuit

The construction of reversible BCD adder consists of several steps. The steps to construct a reversible BCD adder are shown in Algorithm 2.

Algorithm 2. Construct_Reversible_BCD_Adder

1. Construct a reversible full-adder circuit.
2. Construct a reversible 4 bits parallel adder from n-bit reversible parallel adder.
3. Construct necessary combinational logic to control overflow of the result of addition by selecting appropriate reversible gates.
4. Apply two 4 bits operands (A_3, A_2, A_1, A_0 and B_3, B_2, B_1, B_0) into the first 4 bits reversible adder and generate the initial sum (S_3, S_2, S_1, S_0) with carry C_4.
5. Apply the result into the combinational logic for error correction.
6. Apply the initial sum (S_3, S_2, S_1, S_0) into the first four pins of the second reversible 4-bit parallel adder and error correcting code into the last four pins to get the final addition.

Example 4.2.1. To comprehend Algorithm 2 clearly we have presented each step in different figures. According to Algorithm 2, the design of the reversible BCD adder has been shown as follows: Fig. 11 (Step 1, which was also shown in Fig. 7), Fig. 12 (Step 2), Fig. 13 (Step 3), Fig. 14 (Step 4), Fig. 15 (Step 5), and Fig. 16 (Step 6). The final circuit, reversible BCD adder, is presented in Fig. 17.

In our proposed circuit, we used four reversible full-adder circuits of [4] to construct a reversible 4-bit parallel adder, and each of the full-adder circuit produces two garbage outputs. So, the total number of garbage outputs generated from the reversible 4-bit adder is 8. Combinational logic $(((S_1 + S_2) * S_3) + C_4)$ generates six garbage outputs in our proposed reversible BCD adder after using three Toffoli gates. Feynman gates, which are used for copying bits, do not produce any garbage outputs while one copy is needed for a bit and require only one unit delay.

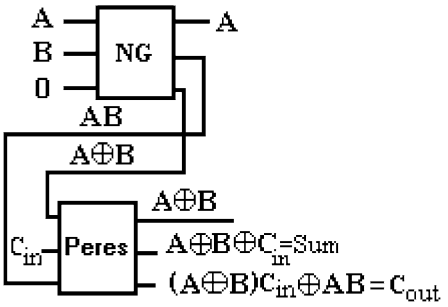

Fig. 11. A reversible full-adder circuit with only two reversible gates, i.e. one new gate and a Peres gate that generates only two garbage outputs [4].

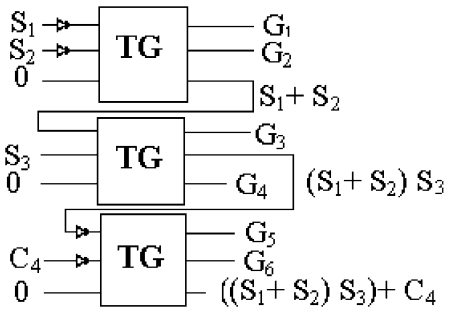

Fig. 13. Combinational logic for reversible BCD adder.

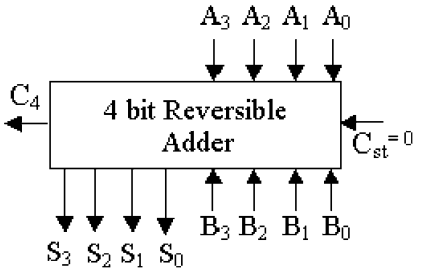

Fig. 14. 4-bit reversible parallel adder with two 4-bits variable *A* and *B*.

As a result, the total number of garbage outputs required to construct a reversible BCD adder is $2 * 8 + 6 = 22$, which satisfies Theorem 4.3.1 in Section 4.3. In our proposed circuit, we use four reversible full-adder circuits to construct a reversible 4-bit parallel adder, where the total 8 reversible gates are required to construct a 4-bit

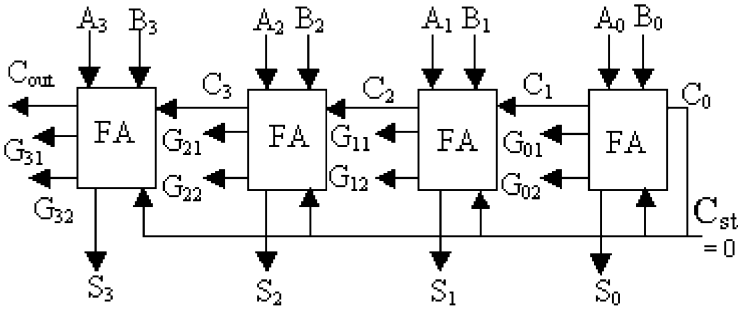

Fig. 12. 4-bit reversible parallel adder with the cascade of four reversible full-adders.

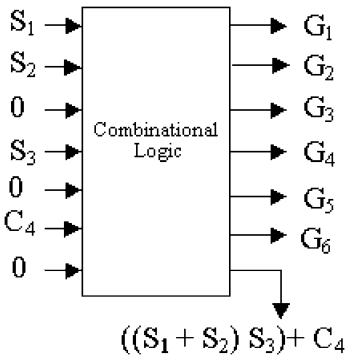

Fig. 15. Combinational logic for BCD adder with appropriate inputs that produces six garbage outputs.

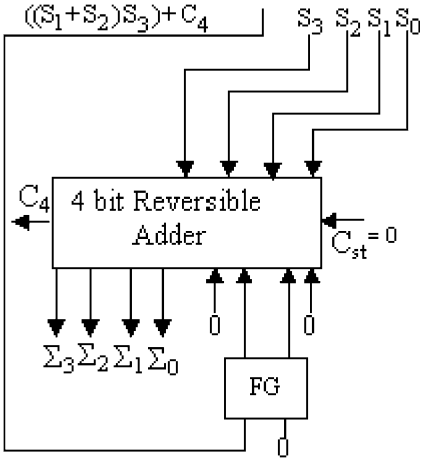

Fig. 16. Second 4-bit parallel reversible adder with the output of the combinational logic.

reversible adder and the total unit delay for a 4-bit reversible parallel adder is $2 * 4 = 8$. Combinational logic $(((S_1 + S_2) * S_3) + C_4)$ requires three 3×3 Toffoli gates (TG) in our proposed design that requires 3 unit delay.

Three Feynman gates are required for copying S_1, S_2, S_3 in order to avoid the fan-out generated from those bits. Finally, another Feynman gate is required for copying the result of combinational logic $(((S_1 + S_2) * S_3) + C_4)$. So, four Feynman gates are required in our proposed BCD adder. As a result, the total number of gates (f_{ngt}) required to construct a reversible BCD adder is $2 * 8 + 3 + 4 = 23$, which satisfies the Theorem 4.3.2 in Section 4.3.

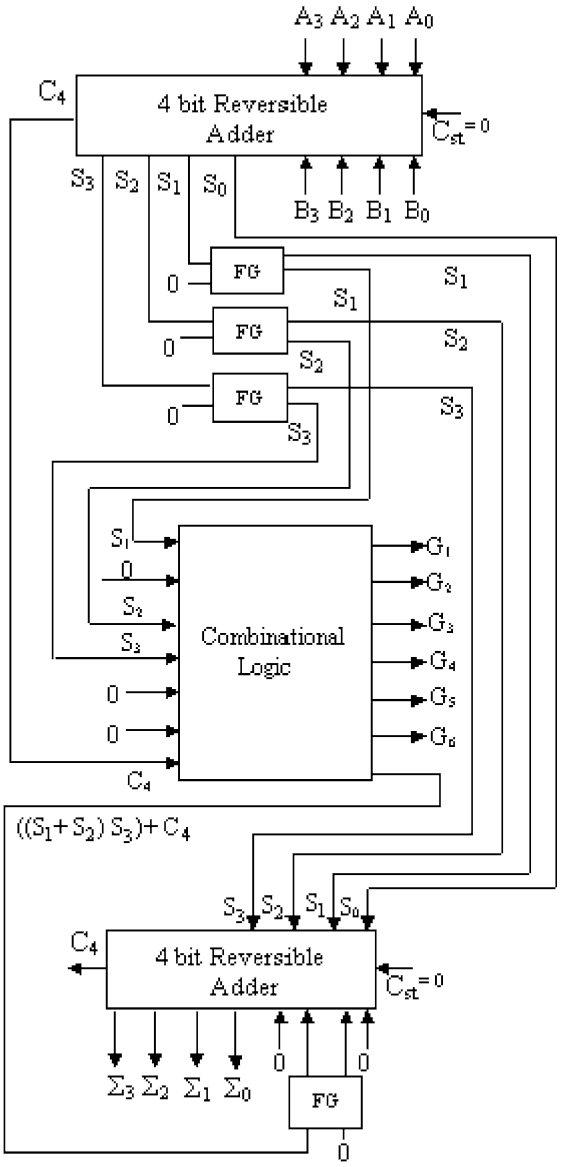

Fig. 17. Full-design of the proposed reversible BCD adder.

4.3. Lower bounds for the reversible BCD adder

Some key points are considered in the design time of reversible BCD adder. First one is the number of gates that we have tried to keep as few as possible. Second key point is the number of garbage outputs. A 4 bits reversible parallel adder should be constructed with the help of four reversible full-adders. We have shown the 4 bits

reversible parallel adder in Fig. 12. But there are some lower bounds on both the number of gates and number of garbage outputs.

Lemma 4.3.1. *A reversible* 4*-bit adder parallel adder can be realized by at least* 8 *garbage outputs.*

Proof. It has been proved that, a reversible full-adder circuit can be realized by at least 2 garbage outputs [2,3]. As a reversible 4-bit parallel adder consists of four reversible full-adder, a reversible 4-bit parallel adder can be realized by at least $(2 * 4 =) 8$ garbage outputs. □

Lemma 4.3.2. *A reversible* 4*-bit adder parallel adder can be realized by at least* 8 *reversible gates.*

Proof. It has been proved that, a reversible full-adder circuit can be realized by at least 2 reversible gates [4]. As a reversible 4-bit parallel adder consists of four reversible full-adder, a reversible 4-bit parallel adder can be realized by at least $(2 * 4 =) 8$ reversible gates. □

On the design of a reversible circuit, copies of some of the bits are greatly needed for computation purposes, so that fan-out is needed. But fan-out is strictly prohibited in reversible computation. In that case, a reversible gate should be used for copying. But, inappropriate selection of gate leads to generate a garbage output with the expected copy. But we will select Feynman gate as an appropriate gate.

Lemma 4.3.3. *Feynman gate is the most suitable gate for a single copy of a bit.*

Proof. As there are exactly two outputs corresponding to the inputs of a Feynman gate, a '0' in the second input will copy the first input in both the outputs of that gate. So, Feynman gate is the most suitable gate for single copy of bit since it is not producing any garbage output. □

Now we will present two theorems related to the number of gates and the number of garbage outputs for the reversible BCD adder.

Theorem 4.3.1. *Let* f_{gb} *be the minimum number of garbage outputs for a reversible* 4*-bit parallel adder and* t_{gb} *be the minimum number of garbage outputs produced by the combinational logic in a reversible BCD adder. Let* f_{ngb} *be the number of garbage outputs for a reversible BCD adder, then*

$$f_{ngb} \geqslant 2 * f_{gb} + t_{gb},$$

where $f_{gb} \geqslant 8$, $t_{gb} \geqslant L - 1$ *and L be the no. of bits required for combinational logic.*

Proof. A reversible BCD adder consists of two reversible 4-bit parallel adders and a combinational logic. As, two reversible 4-bit parallel adders are the components of the reversible BCD adder, no less than $2 * f_{gb}$ garbage outputs will be generated from these two adders. According to Lemma 4.3.1, $2 * f_{gb} = 2 * 8 = 16$.

For L number of bits in the combinational logic, there are two steps to calculate garbage outputs.

In the best case, there is only one garbage output for the first two bits for L; and in the best case, there will be at least one garbage output for each of the rest of the bits of L except the first two bits to represent simple AND–OR functions. Fewer gates and garbage outputs may be required in case of complex transformation of the function. So, in the best case, the total number of garbage outputs for the combination logic (t_{gb}) is

$$t_{gb} = 1 + L - 2 = L - 1.$$

As a result, the total number of garbage outputs of a reversible BCD adder (f_{ngb}) is

$$f_{ngb} \geqslant 2 * f_{gb} + t_{gb},$$

where $f_{gb} \geqslant 8$, $t_{gb} \geqslant L - 1$. □

Theorem 4.3.2. *Let* f_{gt} *be the minimum number of gates in a reversible* 4*-bit parallel adder,* t_{gt} *be the minimum number of gates required to calculate combinational logic and* Δ *be the number of bits to be copied for interfacing purposes among the* 4*-bit parallel adders and combinational logic. Let* f_{ngt} *be the number of gates required to construct a reversible BCD adder, then*

$$f_{ngt} \geqslant 2 * f_{gt} + t_{gt} + \Delta,$$

where $f_{gt} = 8$, $t_{gt} = L - 1$, $\Delta = 4$.

Proof. $f_{gt} = 8$ and $t_{gt} = L - 1$ can be proved easily in the same ways as Theorem 4.3.1, but with the help of Lemma 4.3.2. Finally, at least a few numbers of bits (Δ) are to be copied in almost every reversible circuit, as fan-out is strictly prohibited in reversible circuit. For interfacing purposes among the 4-bit parallel adders and combinational logic in reversible BCD adder, there is exactly four ($\Delta = 4$) bits to be copied [5]. According to Lemma 4.3.3, for Δ bits, the Δ reversible gates (Feynman gates) are required exactly. As a result total number of reversible gates of a reversible BCD adder (f_{ngt}) is

$$f_{ngt} \geqslant 2 * f_{gt} + t_{gt} + \Delta,$$

where $f_{gt} = 8$, $f_{tg} = L - 1$, $\Delta = 4$. □

4.4. Delays for reversible BCD adder

We have already shown the lower bounds for reversible BCD adder in terms of number of gates and number of garbage outputs. We can easily calculate the unit delay of our proposed circuit, which is expandable for n-bit BCD adder. From Example 4.2.1, we find that, the unit delay for 4-bit parallel reversible is 8 and for combinational logic it is 3. But, an extra unit delay is required for copying the bits S_1, S_2 and S_3 through Feynman gate (as they are performed parallel, the total delay for three bits copy is one). Another unit delay is required for copying the error detecting bit $(((S_1 + S_2) * S_3) + C_4)$. So, the total delay for the proposed BCD adder is $(8 * 2 + 3 + 2) = 21$ units.

Like an n-bit parallel adder, we can create n-bit reversible BCD adder by cascading the proposed circuit. For an n-bit reversible BCD adder, the unit delay will be $21n$, as 21-unit delays are required for a single BCD adder.

5. Conclusions and future works

In this paper, we presented a reversible design technique for binary coded decimal (BCD) adder circuit. We have also established the lower bounds on the total number of gates and garbage outputs for reversible BCD adder. Besides proposing two new reversible adder circuits, we have also shown the synthesis of reversible full-adder circuits extensively with the comparison result of different reversible full-adder circuits. Unit delay is also measured for both BCD and full-adder circuits. There exists many significant applications of reversible logics such as low power CMOS, quantum computing and optical computing [1,6,8–11]. The proposed BCD adder circuit is one of the contributions of reversible logic. This circuit can further be used in a large reversible system as a module of reversible logic. Besides proposing two new reversible adder circuits, we have also shown the synthesis of reversible full-adder circuits extensively with the comparison result of different reversible full-adder circuits. In future, we have the plan to construct a large reversible system that executes more than one reversible operations concurrently. Moreover, we also have the plan to synthesize multiple-valued expressions using reversible logic [19].

References

[1] C. Bennett, Logical reversibility of computation, IBM Journal of Research and Development 17 (1973) 525–532.
[2] Hafiz Md. Hasan Babu, Md. Rafiqul Islam, Ahsan Raja Chowdhury, Syed Mostahed Ali Chowdhury, Reversible logic synthesis for minimization of full-adder circuit, in: IEEE Conference on Digital System Design 2003, EuroMicro'03, Belek, Antalya, Turkey, 2003, pp. 50–54.
[3] Hafiz Md. Hasan Babu, Md. Rafiqul Islam, Ahsan Raja Chowdhury, Syed Mostahed Ali Chowdhury, Synthesis of full-adder circuit using reversible logic, in: 17th International Conference on VLSI design 2004, Mumbai, India, 2004, pp. 757–760.
[4] Hafiz Md. Hasan Babu, Md. Rafiqul Islam, Ahsan Raja Chowdhury, Syed Mostahed Ali Chowdhury, On the realization of reversible full-adder circuit, in: International Conference on Computer and Information Technology, vol. 2, Dhaka, Bangladesh, 2003, pp. 880–883.
[5] Ronald J. Tocci, Digital Systems—Principles and Application, New Delhi, India, 1996.
[6] R. Landauer, Irreversibility and heat generation in the computational process, IBM Journal of Research Development 5 (1961) 183–191.
[7] Md. M.H Azad Khan, Design of full-adder with reversible gates, in: International Conference on Computer and Information Technology, Dhaka, Bangladesh, 2002, pp. 515–519.
[8] E. Fredkin, T. Toffoli, Conservative logic, International Journal of Theoretical Physics 21 (1982) 219–253.

[9] G.J. Milburn, The Feynman Processor, Perseus Books, 1998.
[10] R. Feynman, Quantum mechanical computers, Optical News (1985) 11–20.
[11] T. Toffoli, Reversible Computing, Tech memo MIT/LCS/TM-151, MIT Lab for Computer Science, 1980.
[12] P.D. Picton, Fredkin Gates as the Basic for Comparison of Different Logic Designs, Synthesis and Optimization of Logic Systems, London, UK, 1994.
[13] M. Perkowski, L. Jozwiak, P. Kerntopf, A. Mishchenko, A. Al-Rabadi, A. Coppola, A. Buller, X. Song, M.M.H.A. Khan, S. Yanushkevich, V.S Shmerko, M. Chzazowska-Jeske, A general decomposition for reversible logic, in: Proc. RM'01.
[14] Andrel B. Khlopotine, M. Perkowski, Pawel Kerntopf, Reversible logic synthesis by iterative composition, in: Proc. IWLS'02, pp. 261–266.
[15] A. Peres, Reversible logic and quantum computers, Physical Review A 32 (1985) 3266–3276.
[16] D. Maslov, G. Dueck, Reversible cascades with minimal garbage, IEEE Transactions on CAD 23 (11) (2004) 1497–1509.
[17] D. Maslov, G. Dueck, Improved quantum cost for *n*-bit Toffoli gates, IEE Electronics Letters 39 (25) (2003) 1790–1791.
[18] D. Maslov, G. Dueck, Garbage in reversible designs of multiple-output functions, in: 6th International Symposium on Representations and Methodology of Future Computing Technologies (RM), Trier, Germany, March 2003, pp. 162–170.
[19] M. Miller, G. Dueck, D. Maslov, A synthesis method for MVL reversible logic, in: International Symposium on Multiple-Valued Logic (ISMVL), Toronto, Canada, May 2004, pp. 74–80.

Hafiz Md. Hasan Babu was born in Bangladesh. He received his M.Sc. degree in Computer Science and Engineering from the Technical University of Brno, Czech Republic in 1992 under the Czech Government Scholarship. He obtained his Ph.D. in VLSI CAD in 2000 from the Kyushu Institute of Technology, Japan under the Japanese Government Scholarship. He worked with the Department of Computer Science and Engineering, Khulna University, Bangladesh from 1992 to March 2001. Now, he is the Chairman and Associate Professor of the Department of Computer Science and Engineering, University of Dhaka, Dhaka, Bangladesh. In 1995, he was at the Asian Institute of Technology (AIT), Thailand under the DAAD Fellowship from the Federal Republic of Germany. His research interests include Logic Design and Switching Theory, Representation of Logic Functions and Multiple-Valued Logic. He is a member of IEEE.

Ahsan Raja Chowdhury received his B.Sc. degree in Computer Science and Engineering from University of Dhaka, Bangladesh, in 2003. Now he is a student of MS program in Computer Science and Engineering at the University of Dhaka, Bangladesh. His research interests include Logic Synthesis and Design, Reversible Logic, Fuzzy Logic, Multiple-valued Logic and Genetic Algorithm.