



A NEW APPROACH TO DESIGN BCD ADDER AND CARRY SKIPBCD ADDER

K.Boopathi Raja¹, LavanyaS.R², Mithra.V³, Karthikeyan.N⁴

^{1,2,3,4}Department of Electronics and communication Engineering,

SNS college of technology, Coimbatore,TamilNadu, (India)

ABSTRACT

Designed and proposed Reversible logic circuit has found emerging attention in nanotechnology, quantum computing and low power CMOS design. The construction of ,low loss computational structures which are very important for the construction of arithmetic circuits used in Nano-technology, quantum computation and other low power digital circuits. the reversible BCD circuit here form the basis of the decimal ALU of a primitive quantum CPU

Keywords: Reversible logic, BCD adder, BCD carry skip adder, Transmission gate.

I. INTRODUCTION

To detect the power dissipation the reversible logic has received greater attention in certain years. Development in higher-level integration has emerged in better logic circuits and energy loss has also been reduced over the last years. The most prominent application of the reversible logic lies in quantum computers. Quantum computer will be viewed in quantum network composed of quantum logic gates. Bennett [3] showed that zero energy dissipation would be possible if the network consists of reversible gates only. Reversible logic has also found its applications in several stages such as quantum computing [4], nanotechnology [5] and DNA technology [6]

The 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 transistor and materials. Due to characteristics of floating point numbers and limitations on storing formats, not all floating-point numbers can be represented with given precision [8]. Faster hardware for decimal floating point is also imminent as it has its importance in financial, Internet based application.

A reversible logic circuit should have the following features

1. use minimum number of reversible gates.
2. use minimum number of garbage outputs.
3. use minimum constant inputs.
4. use minimum quantum costs

In this paper, advanced techniques of reversible logic implementation for BCD adder and Carry Skip BCD adder are presented. We have compared the proposed designs with the ones that already exist and found that modified designs are greater than the existing ones in terms of gates, garbage output and delay.

Basic Definitions and Literature Review

In this section, basic definitions and ideas related to reversible logic are given. **Reversible Gates** circuits in which number of inputs is always equal to the number of outputs and they will be one to one correspondence within the vector of inputs and outputs [10]. Waste output of reversible gate is said to be Garbage **Output**.

1. FG gate



Garbage output

The lag of some circuit is the maximum gates in a way from any input.

This definition is based on the following assumptions:

- I. In one unit time gates are performed
- II. Before computation begins inputs are performed.

A reversible gate is differentiated based on the complexity and input relation. Minimisation of delay is occurred.

II. OVERVIEW OF THE EXISTING DESIGNS

Combination of New gate [12] and Peres gate [13] as full-adder is designed as the reversible BCD adder. Ideas of design are follows: binary addition will be performed in 4bits. Overflow of BCD is detected. Fan-out is restricted in strict reversible circuit. Error correction value is obtained by adding 4bit parallel adder, i.e. in binary 0011, when overflow obtained. First 4-bit parallel is also obtained. The design will improve the number of gates, waste outputs and lag. Overcome the limitations of the existing designs reversible BCD adder and carry skip adder is used

III. PROPOSED REVERSIBLE BCD ADDERS

Reversible BCD and Carry Skip Reversible BCD adder with improved designs have been presented in this paper with algorithms and figures.

Basic Properties

Binary Coded Decimal (BCD) number and adder with necessary terminology are presented here.

Definition

The important measuring parameter for logic design using reversible gate includes the number of gates used for the design with the less number of unused outputs. The quantum cost of each reversible logic gate is an important optimization parameter. The quantum cost of a 1*1 reversible gate is assumed to be zero. The quantum cost of the 2*2 reversible logic gate is taken as unity. The result of the binary addition of the two BCD numbers overflow is checked by combinational circuit for BCD detection.

The resulting number is greater than 1001, and then the BCD adder overflow occurs. Let A_{0-4} and B_{0-4} be the two BCD numbers to be added and the resulting number is represented by T_{0-4} . Carry out is represented by C_5 . Resulting number is greater than 1111, and then the c_5 is set. It is easy to note $(T_2 + T_1)$. T_4 and C_5 are not set at the same time. $T_3T_2T_1T_0$ and any carry out from this addition is ignored. Carry out from the addition of two BCD numbers $A_3A_2A_1A_0$ and $B_3B_2B_1B_0$ is already computed along with F . If F is not set, no error correction is needed. The final result is the partial sum of T_{0-4} .

Proposed Reversible BCD Adder

The basic building blocks which are logically reversible should have three binary inputs and three binary outputs. There are three components of BCD binary adder: a 4-bit parallel adder, overflow detection logic and overflow correction logic.

Proposed reversible 4-bit ripple carry adder has been designed using four 1bit reversible adders. The proposed design performs better in terms of quantum cost has been reduced by 18.75%.

ALGORITHM

Input: T : The partial sum received from the binary adder which was mentioned by 4bit vector discussed.

Output: From this algorithm the vector would be the output, overflow detection bit is F . After overflow detection it is noted that the vector is required for correction, but detection circuit was featured. There are numerous ways of generating copies are in numerous way of T vector at any level, but the detection circuits preferred to produce T vector

Overflow detection bit, the resulting circuit of the expression may contain at least 2 blocks. The approach might be same for the following -

- 1) The first block will take T_1 and T_2 and output will be $T_2 + T_1$
- 2) The second block will take the T_4 , C_4 and output from first block will be $T_2 + T_1$ and compute the result $F = (T_2 + T_1) \cdot T_3 \oplus C_4$.

Return R : = $T \cup F$;

end

Example Figure 4.2.1 shows a direct implementation of Algorithm 4.2.1 where $T = (C_4, T_3, T_2,$

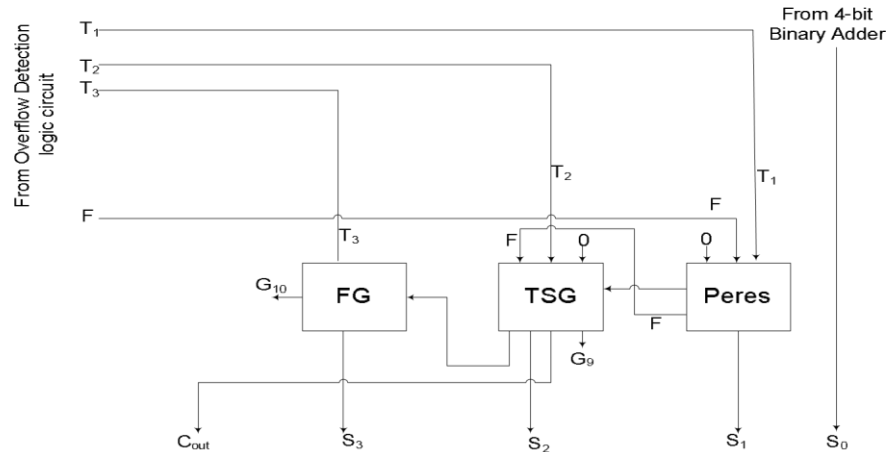


Figure 4.2.2. Designing a 1 bit BCD adder's correction logic circuit

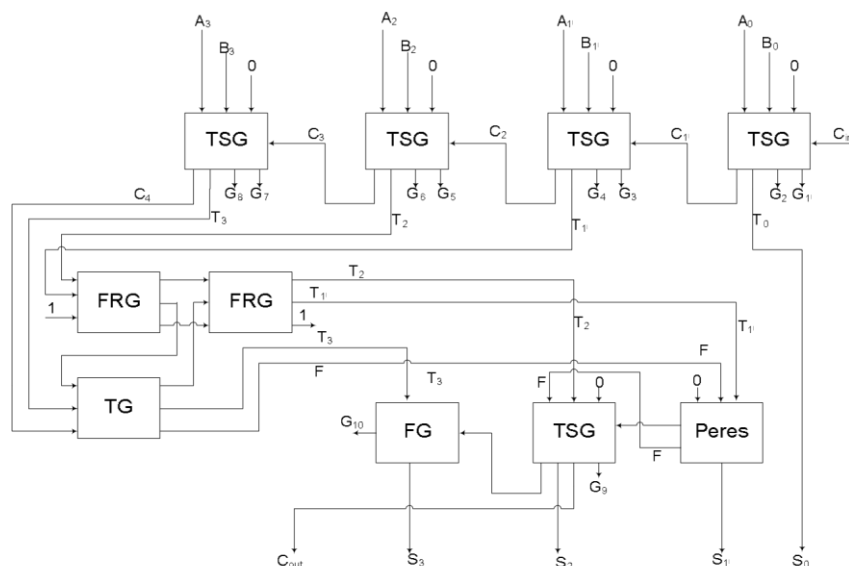
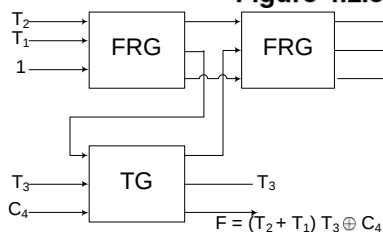


Figure 4.2.3. Design of a 1-digit BCD adder



T_1). **Figure digit BCD adder's overflow**

detection logic

Algorithm OCA (R)

Input: R: From the overflow detection logic circuit a 4bit vector is received.

Output: $S(C_{out}, S_3, S_2, S_1, S_0)$ final corrected BCD sum. The detection logic As T vector that was fed does not include T_0 , it is free and intact to use as S_0 . waiting for the for the final carry outis not mandatory, because if F is 1, it is sure that the $C_{out}=1$, so we need not propagate further to compute this carry. **begin**

1)The first block will take T_1 and F from the overflow detection logic circuit and generate $S_1 = T_1 \oplus F$ and $carry_out_1 = T_1.F$.

2)The second block will take carry out of the first block, T_2 from the overflow detection circuit and F (this F can be duplicated using numerous techniques, in our circuit first block generates F again) and generate $S_2 = T_2 \oplus F \oplus carry_out_1$. It will also generate $carry_out_2 = (T_2 \oplus F). carry_out_1 + T_2.F$.

3) The third block will take carry out of the second block, T_3 from the overflow detection circuit and generate $S_3 = T_3 \oplus carry_out_2$.

return S ;

end

Example 4.2.2. Figure 4.2.2 shows a direct implementation of Algorithm 4.2.2.

Algorithm BCD_ADDER_CONSTRUCTION _ ALGORITHM (A, B)

Input: $A = (A_3, A_2, A_1, A_0)$ and $B = (B_3, B_2, B_1, B_0)$ are two 4-bit input BCD vectors.

Output: Final corrected BCD sum $S (C_{out}, S_3, S_2, S_1, S_0)$. **begin**

$T :=$ Binary Adder output(A, B);

$R :=$ ODA(T); $S :=$ OCA(R);

Return S ; **end**

Proposed Carry Skip Reversible BCD Adder

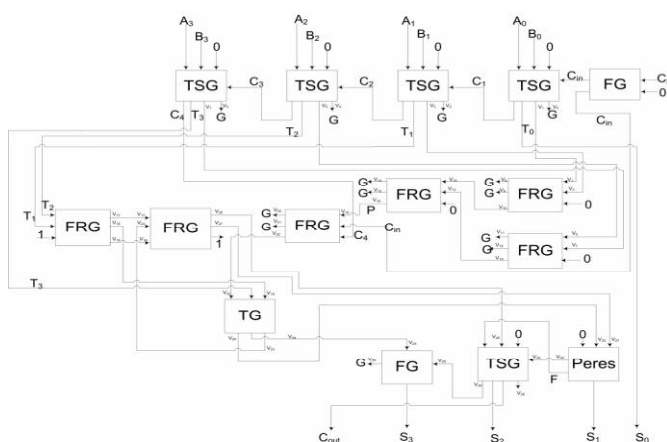


figure:Carry skip 1digit BCD adder

It uses two four bit adders, carry skip logic circuits and a correction logic circuit. Carry skip logic may generate the carry out, C_{out} instantaneously. We will present these components with proper algorithms and appropriate figures. The proposed design is found to be much better than the existing one [11] in terms of number of gates, number of garbage and delay. The quantum cost of the 2*2 reversible logic gate is taken as unity. the result of the binary addition of the two BCD numbers overflow is checked by combinational circuit for BCD detection.

Proper expression for this condition is: $P_i = A_i \oplus B_i$ and C_{in} to the block will propagate to the carry output of the block if the entire P_i 's are set. In this way, we can generate C_{out} without waiting for it to be generated in ripple carry fashion. Let, the propagation signal for the block is denoted by P . Then, $P = P_3 \cdot P_2 \cdot P_1 \cdot P_0$. If P is set, C_{in} will be propagated to the C_{out} . However, in the other case, C_{out} will be generated in the ripple carry fashion. So, carry skip logic bit of the block is $K = P \cdot C_{in} + C_4$ where C_4 is the carry generated in the ripple carry fashion. The overall overflow detection bit, $F = (T_1 + T_2) \cdot T_3 \oplus K$ is generated in the same way with Reversible BCD adder presented earlier in this paper. Overflow correction logic incorporated is the same as the Reversible BCD adder.

The following procedure (Algorithm 4.3.1) is used for the design of Carry Skip 1-digit BCD adder. This procedure is presented along with appropriate figure.

Algorithm CARRY_SKIP_BCD_ADDER_

ALGORITHM (A, B, C_i)

Input: A (A_3, A_2, A_1, A_0) and B (B_3, B_2, B_1, B_0)

are two input vectors and C_{in} is the carry in.

Output: A BCD adder capable of performing the sum = $A + B$. The buffer vector S ($C_{out}, S_3, S_2, S_1, S_0$) will store the result.

begin

Step 1: Compute P (propagate bit). Initially $P := \text{true}$ for all i in $\{0, 1 \dots 3\}$ **do**

$P := P \text{ AND } (A_i \oplus B_i)$.

Step 2: Compute $T := \{C_4, T_3, T_2, T_1, T_0\}$, where

$T_i := A_i \oplus B_i \oplus C_i$ and C_i 's are generated from each adder block.

Step 3: Compute carry skip logic bit, $K := P \cdot C_{in} + C_4$.

Step 4: The overall overflow detection bit $F :=$

$(T_1 + T_2) \cdot T_3 \oplus K$, which is **true** whenever a BCD overflow is detected.

Step 5: Add binary 0110 to T if overflow detection bit F is **true**.

Step 6: Compute $S := (C_{out}, S_3, S_2, S_1, S_0)$, the final sum of the addition process.

Return S ;

end

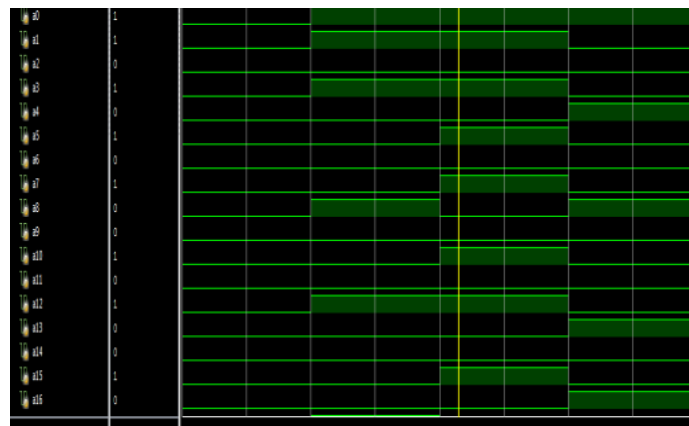
shows a direct implementation of Algorithm

The Fredkin gates in the middle of the Figure 4.3.1 generate the block propagation, P and carry skip logic bit, K . Fredkin gates and Toffoli gate on the left side performs the BCD overflow detection same as for reversible BCD adder. BCD overflow correction logic is also like the reversible BCD adder. Table 2 shows the comparative analysis of the improved Carry Skip Reversible BCD adder with the one presented in [11] and it clearly shows that the proposed design outperforms the existing one in every metrics. Circuit presented in [11] allows multiple fan-outs that are prohibited in strict reversible sense.

Output waveform of BCD adder



Output waveform of carry skip 1digit BCD adder



IV. CONCLUSIONS

In this paper, reversible logic syntheses were carried out for both BCD adder and carry skip BCD adder. The designs have been done for ease of reversible logic implementation and it has been found that the proposed designs are far better than the existing ones [9,11] in terms of number of gates needed, number of garbage outputs produced and delay. Improved Carry Skip BCD adder can perform much faster than the BCD adder. If multiple BCD blocks are used in the carry skip adder, i.e. m -digit BCD numbers, then carry skip BCD adder

has the potential to perform the desired operation much faster. BCD adders can be an important part of some other larger and more complex reversible circuits. Fast and improved BCD adders may also find its use in future quantum computers [4].

REFERENCES

- [1] Keyes R, Landauer R. Minimal Energy Dissipation in Logic. *IBM Journal of Research and Development* 1970; 14: 153-7.
- [2] Landauer R. Irreversibility and heat generation in the computational process's. *IBM Journal of Research Development* 1961; 5: 183-91.
- [3] Bennett CH. Logical reversibility of computation. *IBM Journal of Research and Development* 1973; 17: 525-32.
- [4] Shende VV, Prasad AK, Markov IL, Hayes JP. Synthesis of reversible logic circuits. *IEEE Transaction on CAD* 2003; 22.(6): 723-9.
- [5] Moore GE. Cramming more components onto integrated circuits. *Journal of Electronics* 1965; 38(8).
- [6] Frank M. Physical Limits of Computing. CIS 4930.1194X/6930.1078X, 2000.
- [7] Perkowski M. Reversible Computation for Beginners. Lecture Series 2000. Portland State University. <http://www.ee.pdx.edu/~mperkows>.
- [8] Hayes JP. *Computer Architecture and Organization*, 3rd ed. McGraw-Hill; 1998.
- [9] Babu HMH, Chowdhury AR. Design of a Compact Reversible Binary Coded Decimal Adder Circuit.
- [10] Babu HMH, Islam MR, Chowdhury AR, Chowdhury SMA. Synthesis of full-adder circuit using reversible logic. 17th International Conference on VLSI Design 2004; 757-60.
- [11] Thapliyal H, Kotiyal S, Srinivas MB. Novel BCD Adders and their Reversible Logic Implementation for IEEE 754r Format. 19th International Conference on VLSI Design 2006; 387-92.
- [12] Khan M. H. A and Perkowski M. Multi-output ESOP synthesis with cascades of new reversible families. 6th International Symposium on Representations and Methodology of Future Computing Technologies, March 2003, 144-153.
- [13] Peres A. Reversible Logic and Quantum Computers. *Physical Review* 1985; 3266-76.