

## New Approach for Implementing BCD Adder Using Flagged Logic

<sup>[1]</sup>B. Prashanthi kumari,  
Dept. of ECE, SCET,  
Narsapur.

<sup>[2]</sup>G. N. V. Ratna Kishor,  
Assistant Professor, ECE,  
SCET, Narsapur.

### Abstract

*Financial and commercial applications use decimal data and spend most of their time in decimal arithmetic. In this project I am going to propose a New Approach for implementing BCD Adder using Flagged logic. We extend the two operand ripple carry addition by one with the third input being constant. The addition technique is made fast by generating flag bits appropriate to the constant added. The third constant in case of our proposed design is 6(0110) for converting the outputs exceeding 9 to Binary Coded Decimal (BCD) number. To make area and delay comparison, the proposed adder and previously proposed adders basic BCD adder and Reduced delay BCD Adder are going to be implemented in Verilog.*

Key Words -Flagged binary adder, Carry look ahead adder, Carry skip adder, Correction circuit, flag bit computation.

### 1. INTRODUCTION AND RELATED WORK

The binary numbering system is, by far, the most common numbering system in use in computer systems today. In days long, however, there were computer systems that were based on the decimal (base 10) numbering system rather than the binary numbering system. Such computer systems were very popular in systems targeted for business/commercial systems. Although systems designers have discovered that binary arithmetic is almost always better than decimal arithmetic for general calculations, the myth still persists that decimal arithmetic is better for money calculations than binary arithmetic. Therefore, many software systems still specify the use of decimal arithmetic in their calculations.

BCD representation does offer one big advantage over binary representation: it is fairly trivial to convert between the string representation of a decimal number and its BCD representation. This feature is particularly beneficial when working with fractional values since fixed and floating point binary representations cannot exactly represent many commonly used values between zero and one

(e.g., 1/10). Therefore, BCD operations can be efficient when reading from a BCD device, doing a simple arithmetic operation (e.g., a single addition) and then writing the BCD value to some other device. Many architectures and algorithms have been proposed to date for decimal arithmetic. Behrooz shirazi et al., [2] designed an adder for redundant BCD addition. Though the design involves simple conversion of a BCD number to redundant BCD and perform addition in redundant form and again convert the result back to BCD form, it suffers from high delay. In addition the conversion circuitry used adds to the design complexity. Algorithms and adders for BCD addition are presented in [3] by Robert D.Kenny et al., in which the design using speculative addition technique have regular structure and their correction unit is independent of the number of input operands whereas the other design using non-speculative addition have lower delays.

The use of reversible logic for the design of BCD adders were presented in [4][6][17]. The reversible logic adders perform well in terms of power dissipation and logic count, however it is prone to higher delay. In a pioneer work on BCD addition Sreehari Veeramachaneni *et al.*, [7] used multiplexer for the addition of correction bits. The circuit has lower delay compared to the architectures using adders for correction bit addition.

Alp Arslan Bayrakci *et al.*, [8] proposed a BCD adder with efficient carry generation using analyzer circuit. The circuit performs well in terms of delay compared to architectures mentioned in literature and shows better area performance. Anshwal Singh *et al.*, [9] designed a novel architecture for BCD addition and subtraction. The design uses three major blocks viz., PG block, prefix block and the correction block and generates carry without any extra logic thus performing better in terms of area performance compared to the BCD adder in [10]. Sundaresan.C *et al.*, [11] in a pioneer work on design of Reduced delay BCD adder used Carry Look Ahead (CLA) adder in the initial stage being followed by carry network and correction logic in the second and third stages. Though the circuit is fast compared to the architecture in [12], the use of CLA adder in initial stage increases area cost. Chetan Kumar *et al.*, [12] presented a unified

architecture for BCD and binary addition. Though the circuit has lower delay compared to the architectures mentioned in literature the design of post correction circuitry poses problems for multi-bit operands. Osama Al-khaleel *et al.*, [13] proposed a correction free BCD adder in which the input operands are split and added in two stages. Stage 1 adds the MSB three bits of a four bit BCD number and the result out of stage 1 is passed to stage 2 and added with the LSB. The latency of the architecture is very less compared to the architectures mentioned in literature.

To further reduce power and latency in BCD addition we have proposed an adder using flagged binary addition[1] for the correction constant addition. The output of adders of first stage and flagged computation block are passed through a multiplexer. The control signal for the multiplexer is generated from a control circuit which produces 1 for sum values exceeding 9 and 0 else. The rest of the paper is organized as follows. Section 2 gives a brief description of BCD addition and the architecture of conventional BCD adder. Section 3 discusses about the design of proposed flagged BCD adder. In section 4 the performance of the proposed BCD adder are discussed and compared with the previous approaches. Section 5 gives brief conclusion of the work done.

## 2. OVERVIEW OF BCD ADDITION

In electronic systems, BCD is an encoding for decimal numbers in which each digit is represented by its own binary sequence. It allows easy conversion to digits and results in faster calculations. When BCD numbers are added, each sum digit should be adjusted to skip the six unused codes. For instance, the addition of two decimal digits in BCD, together with a possible carry from a previous least significant pair of digits (assuming maximum value for input digits) viz.,  $9 + 9 + 1$  would result in 19. The equivalent binary sum will be in the range 0 to 19 represented in binary as 0000 to 10011 and BCD as 0000 to 1 1001 (the first 1 being carry and next four bits being BCD digit sum). For the binary sum equal to or less than 1001 the corresponding BCD digit is correct. However when the binary sum exceeds 1001, the result is invalid BCD digit. The addition of 6 (0110) to the binary sum converts it to the correct digit and also produces carry [14]. Fig.1 shows the block diagram of a 1 digit BCD adder [14] based on the above methodology.

The input digits in binary are  $A_3A_2A_1A_0$  and  $B_3B_2B_1B_0$ .  $S_3'S_2'S_1'S_0'$  are the outputs of the first stage 4 bit adder, to which correction bits 0110 (6) is added at the second stage to produce the BCD number  $S_3S_2S_1S_0$  shown in equation (2)-(5) along with carry output  $C_N$  shown in

equation(1). The carry  $C_N$  will be one for digits exceeding 9 or else it will be 0.

$$C_N = C_{OUT} + S_3'S_2' + S_3'S_2' \text{ --- (1)}$$

$$S_0 = B_0 \text{ --- (2)}$$

$$S_1 = B_3'B_1 + B_3B_2B_1 \text{ --- (3)}$$

$$S_2 = B_3'B_2 + B_2B_1 \text{ --- (4)}$$

$$S_3 = B_3B_2'B_1' \text{ --- (5)}$$

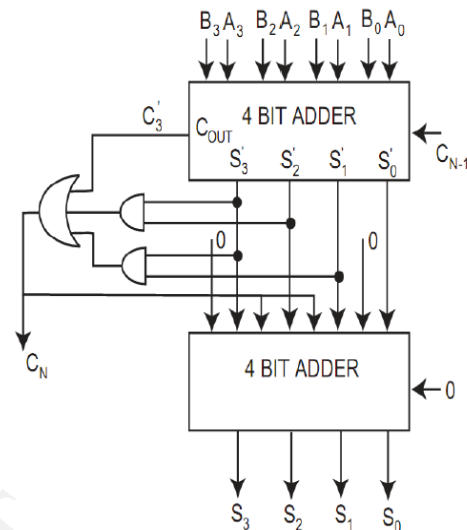


Figure 1. Block Diagram of BCD Adder

## 3. OVERVIEW OF REDUCED DELAY BCD ADDER

The conventional BCD adder is very simple, but also very slow due to the carry ripple effect. If the BCD addition is analyzed carefully, we see that there are three cases:

*Case 1:* The sum of two BCD digits is smaller than 9. In this case, it is certain that there is no carry output even if there is a carry input. Furthermore, the result for this digit does not require a correction.

*Case 2:* The sum of two BCD digits is greater than 9. In this case, a correction is required. Moreover, a carry output is produced regardless of the carry input.

*Case 3:* The sum of two BCD digits is exactly 9. In this case, the input carry determines whether a correction is required and whether a carry output is produced.

For the first two cases, the incoming carry has no effect on determining the carry output; therefore, the carry output can be determined without knowing the existence of the carry input. On the other hand, if the addition result is 9 (Case 3), then the input carry determines the existence of the carry output, which may ripple even up to the most significant digit. Therefore, Case 2 and Case 3 can be represented by a digit generate (DG) and a digit propagate (DP) signals, respectively. Figure 2

shows how the DG and DP signals of a digit are computed in our design. After having all the DG and DP signals, the output carry for each digit can be found easily by Equation 1. Due to the nature of this equation, we can form DP by ANDing only Sum [0] and Sum [3] instead of using all bits of Sum [3:0]. The DG and DP signals can be utilized similar to the generate and propagate signals used in a binary CLA circuit. Therefore, all schemes developed for a binary CLA can be used in order to speed up carry computation.

$$\text{OutputCarry} = \text{DG} + \text{DP} \cdot \text{InputCarry} \text{ -----(6)}$$

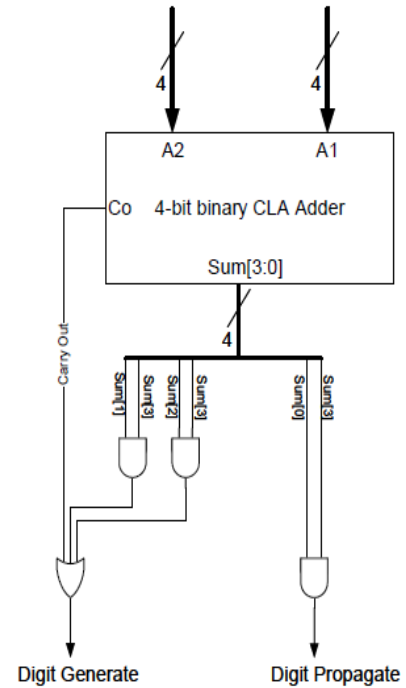


Figure2. Adder + Analyzer unit

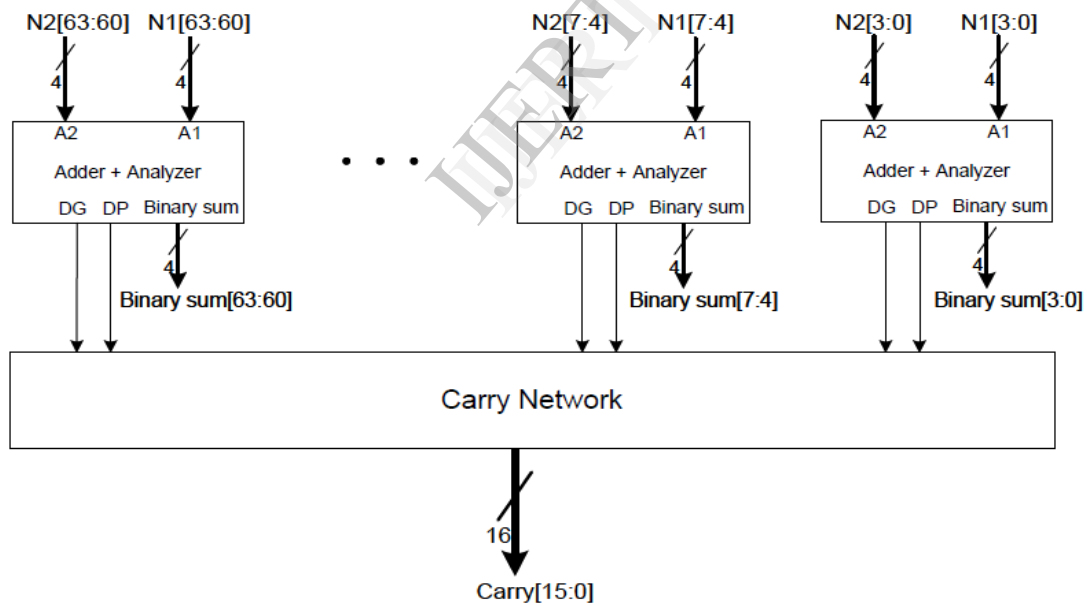


Figure 3. Adder + Analyzer + Carry Network

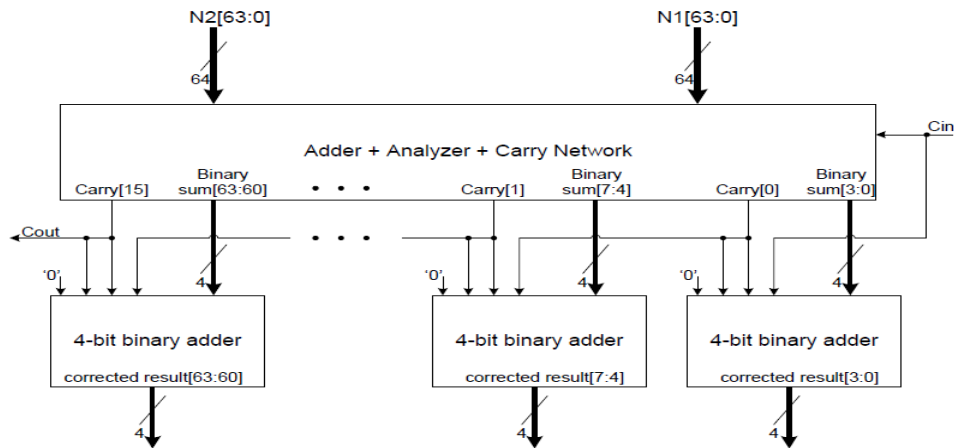


Figure4. Block Diagram of Reduced Delay BCD Adder

The combination of the first level 4-bit adders and the Carry Network is shown in Figure 3. The carry value for each digit is computed inside the Carry Network using Equation 6. The Carry Network can be any type of parallel prefix network or two level carry look-ahead logic can be used instead. The carries computed by Carry Network are used in the correction step. Figure 4 shows the complete BCD adder including the 4-bit adders used for correction.

Correction is done by adding 0, 1, 6, or 7 to the binary sum coming from the first level adder. For each digit, the existence of the output carry and the input carry determine the value to be added for correction. Table 1 shows the correction value to be added for all cases. The correction step must fulfil two requirements. First, the carry, coming from the previous digit, should increment the binary sum of the related digit by 1. Second, the output carry of the related digit should determine whether the binary sum will be corrected by adding 6 or not. The correction by adding 6 is required only when there is a decimal carry-out coming from the carry network. Figure 4 shows how the correction step fulfils these requirements. As a result of such a design, the carries are utilized only in the correction step.

The value added for correction	Possible cases	
	Input carry from prev.digit	Output carry to next.digit
0	0	0
6	0	1
1	1	0
7	1	1

Table1. The Selection of the Value to be Added for Correction

### 4. PROPOSED FLAGGED BCD ADDER

We have proposed a hardware efficient BCD adder using flagged binary addition. The various blocks of the proposed BCD adder are 4 bit Ripple Carry Adder(RCA), Excess 9 detector, flag bit computation block, flag inversion block and four 2:1 multiplexers whose schematic is shown in figure 6. The input  $A(a3a2a1a0)$  and  $B(b3b2b1b0)$  are fed to the first stage binary adder. The sum output  $S(S3S2S1S0)$  and carry out  $Co$  of this stage is fed to Excess 9 detector shown in figure 6(a). If the sum  $S(S3S2S1S0)$  is less than or equal to 9 the  $Cout$  of Excess 9 detector will be zero and the sum  $S(S3S2S1S0)$  will be passed out through the multiplexer. If the sum  $S(S3S2S1S0)$  exceeds 9, the  $Cout$  of Excess 9 detector will be 1 and the sum bits will be passed through the flag bit computation block to generate intermediate carry bits ( $d4d3d2d1$ ) shown in equation (7)-(10).

$$d1=d0 \& s0 \text{ ---(7)}$$

$$d2=d1 + s1 \text{ ---(8)}$$

$$d3=d2 + s2 \text{ ---(9)}$$

$$d4=d3 + s3 \text{ ---(10)}$$

The carry bits ( $d4d3d2d1$ ) and sum  $S(S3S2S1S0)$  are then used by this block to generate flag bits ( $F0,F1,F2,F3$ ) shown in equation (11) -(14).

$$F1=d1 \text{ ---(11)}$$

$$F2=d2 \text{ ---(12)}$$

$$F3=d3 \text{ ---(13)}$$

$$F4=d4 \text{ ---(14)}$$

The flag bits ( $F0,F1,F2,F3$ ) and sum  $S(S3S2S1S0)$  are passed through flag inversion logic shown in fig.6(c) to generate the BCD output  $M3M2M1M0$  for  $S(CoS3S2S1S0)$  which exceeds 9. The  $M3M2M1M0$  of the flagged inversion block forms the other input to the multiplexer which is passed out for 1 value of  $Cout$ .

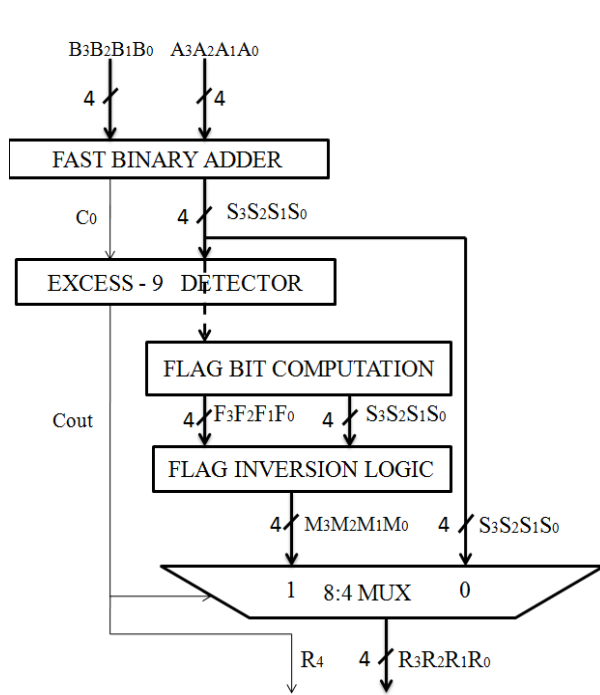
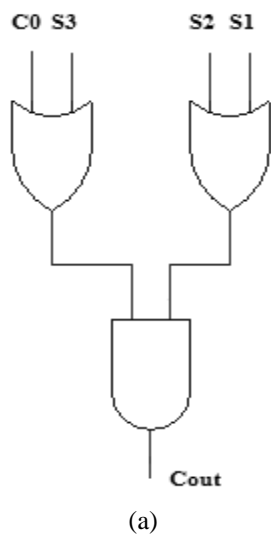
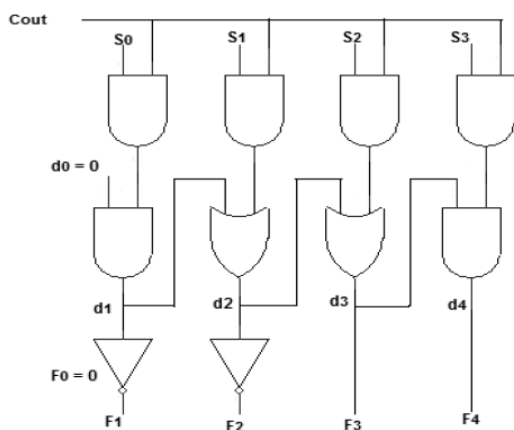


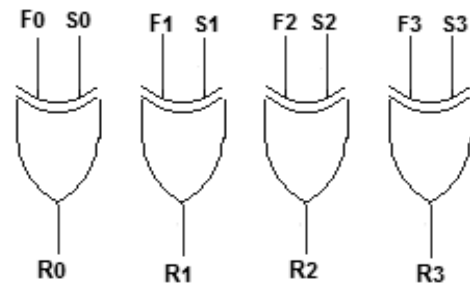
Figure 5. Block diagram of proposed flagged BCD adder



(a)



(b)



(c)

Figure 6. Schematic of (a) Excess-9 Detector, (b) Carry computation and Flag bit computation block (c) Flag inversion logic

## 5. RESULTS AND DISCUSSION

The proposed flagged BCD adder is described using Verilog to produce gate level net list. We evaluated the proposed BCD adder design using Carry Skip Adder(CSA) for the first stage addition. Conventional BCD adder [14], Reduced Delay BCD Adder are used for comparison. The area and delay results are shown in table 2. From the reports, it is seen that our proposed flagged BCD adder design using CSK adder has lower logic cell count compared to all other architectures used for comparison, thanks to the low area CSK adder, flagged bit computation logic and flag inversion logic which realizes constant(0110) addition with fewer gates. The delay of the proposed flagged BCD adder designs is less when compared with all other BCD adder designs used for comparison. This is due to the long carry generation(Cout) path which is 2 OR and 1 OR excess of 4 xor delays in CSA based BCD adder[15] and Conventional BCD adder [14] respectively.

## 6. CONCLUSION

We have proposed a new BCD adder design by using the concept of high speed addition for three operands (two input operand and a constant) by generating flag bits. Extensive comparison using synthesis results shows that the proposed flagged BCD adder outperformed all other previous designs in terms of delay and area. The potential benefits of reduced logic cell count of our proposed flagged BCD adder can be realized in fair ADP performance. This suggests the suitability of our proposed flagged BCD adder for portable VLSI implementation. Moreover, the proposed BCD adder can be easily extended to multi digit addition.



BCD Adders	LUT's	Delay (ns)
Conventional BCD Adder	192	122.530
Reduced Delay BCD Adder	208	22.103
Proposed Flagged BCD Adder	171	14.794

Table 2.Comparison of area and delay of proposed BCD Adder.

## REFERENCES

[1] Vibhuti Dave, Erdal Oruklu and Jafar Saniie, "Constant addition with flagged binary adder architectures", *Integration the VLSI journal* Vol.43 pp. 258–267 2010.

[2] Behrooz Shirazi, David Y.Y.Yun , and Chang N.Zhang, "Vlsi Designs for Redundant Binary-Coded Decimal Addition", *IEEE* pp.52-56 1988.

[3] Robert D. Kenney and Michael J. Schulte, "High-Speed Multi-operand Decimal Adders", *IEEE Transactions On Computers*, Vol.54, No.8, pp.953-963, August 2005.

[4] Hafiz Mohammed Hasan Babu and Ahsan Raja Chowdhury, "Design of a Reversible Binary Coded Decimal Adder by Using Reversible 4-bit Parallel Adder", *IEEE, VLSID* 2005.

[5] Himanshu Thapliyal and Nagarajan Ranganathan, "A New Reversible Design of BCD Adder" *EDAA* 2011.

[6] Ashis Kumer Biswas, Md. Mahmudul Hasan, Moshaddek Hasan, Ahsan Raja Chowdhury and Hafi Md. Hasan Babu, "A Novel Approach to Design BCD Adder an carry Skip BCD Adder" *21st International Conference on VLSI Design* pp. 566-571 2008.

[7] Sreehari Veeramachaneni, M.Kirithi Krishna, Lingamneni Avinash, Sreekanth Reddy P,M.B.Srinivas, "Novel, High-Speed 16-Digit BCD Adders Conforming to IEEE 754r Format" *IEEE Computer Society Annual Symposium on VLSI (ISVLSI'07)*, 2007.

[8] Alp Arslan Bayrakci and Ahmet Akkas, "Reduced Delay BCD Adder", *IEEE* pp.266-271, 2007.

[9] Anshul Singh, Aman Gupta, Sreehari Veeramachaneni and M.B. Srinivas, " A High Performance Unified BCD and Binary Adder / Subtractor", *IEEE Computer Society Annual Symposium on VLSI*, pp.211-216 2009.

[10] Sreehari Veeramachaneni.M, Kirithi Krishna.V, Prateek G.S, Subroto.S, Bharat, M.B.Srinivas, "A Novel Carry- Look Ahead Approach to a Unified BCD and Binary Adder/Subtractor", *21st International Conference on VLSI Design* 2008, pages 547-552, January 2008.

[11] SundaresanC, Chaitanya CVS, PR Venkateswaran, Somashekara Bhat and Mohan Kumar J, "Modified Reduced Delay BCD Adder" *IEEE4th International Conference on Biomedical Engineering and Informatics (BMEI) 2011*, pages 2148-2151.

[12] Chetan Kumar V, Sai Phaneendra P, Syed Ershad Ahmed, Sreehari Veeramachaneni, N, M.B Srinivas, "A Unified Architecture for BCD and Binary Adder/Subtractor" , *14th Euromicro Conference on Digital System Design* pp.426-429 2011.

[13] Osama Al-Khaleel, Mohammad Al-Khaleel, Zakaria Al-Qudah,Christos A. Papachristou and Khaldoon Mhaidat, and Francis G. Wolff," Fast Binary/Decimal Adder / Subtractor with a Novel Correction-Free BCD Addition" *IEEE* pp.455-459 2011.

[14] Moris Mano - *Digital design* 3rd edition Prentice Hall Puclication 2001.

[15] Himanshu Thapliyal, Saurabh Kotiyal and M.B Srinivas, "Novel BCD Adders and their Reversible Logic Implementation for IEEE 754r Format", *Proceedings of the 19th International Conference on VLSI Design (VLSID'06)* 2006.

[16] [http:// webster.cs.ucr.edu / AoA / Windows / HTML / Advanced Arithmetica6.html](http://webster.cs.ucr.edu/AoA/Windows/HTML/AdvancedArithmetica6.html) – Chapter 4 More data representation – subsection 4.3 - Basics of Binary Coded Decimal (BCD) Representation

[17] B. Ramkumar and Harish M Kittur "Low-Power and Area-Efficient Carry Select Adder", *IEEE transactions on Very Large Scale Integration (VLSI) systems*, Vol. 20, No. 2, pp.371-375, February 2012.

[18] Min Cha Earl.E.Swartzlander, Jr. "Modified carry-skip adder for reducing first block delay", *proceedings of 43rd Midwest symposium on circuits and systems*, August 2000.