

Design and Synthesis of Reduced Delay BCD

C. Sundaresan
School Of Information Sciences
Manipal University
Manipal, India

Chaitanya CVS
School Of Information Sciences
Manipal University
Manipal, India
Email: chaitucvs [AT] gmail.com

PR Venkateswaran
Welding Research Institute, BHEL,
Tiruchirappalli, India

Somashekara Bhat
Manipal Institute of Technology
Manipal University
Manipal India

J Mohan Kumar
School Of Information Sciences
Manipal University
Manipal, India

Abstract: Arithmetic and memory address computation are performed using adder operations. Hence, design of adders form an important subset of electronic chip design functionality. Performance of BCD adders is to be considered with gate count, area, delay, power consumption. A new BCD adder design is attempted here to reduce the delay and thereby increasing the speed of response. BCD adder design is considered with respect to high speed addition requirement including multi operand addition, multiplication and division. The new architecture supports 64 bit and 128 bit operands and reduces the delay by adding parallelism.

Keywords: Adder, BCD Adder, Carry Logic, Delay

I. INTRODUCTION

In present electronics systems, each digit in decimal number is encoded using Binary Coded Decimal (BCD) method. Decimal fractions are pervasive in human endeavors, yet most cannot be represented by binary fractions. In recent times, research has been done on designing BCD adders and different designs have been proposed [9, 10, 11, 12, 13, 14]. Enhancing the speed of the BCD arithmetic is the major consideration which need to be taken care while design the BCD arithmetic block which is being discussed in this paper. The various techniques for higher order BCD numbers high speed addition which form the core for arithmetic operations such as multi-operand addition [3, 4], multiplication [5] and division [6] is being introduced and analyzed in this paper. A new architecture for the fast decimal addition is proposed, based on which architectures for higher order adders such as ripple carry and carry look-ahead adder is derived. The rest of paper is organized as follows: BCD arithmetic's overview briefs the background for decimal adders. High speed BCD adder briefs the design of higher valence adder. In result and discussion session brief discusses about an area, timing and

power results for reduced delay BCD adder, modified reduced delay BCD adder and suggested adder high valence BCD adder has been presented.

II. BCD ARITHMETIC OVERVIEW

BCD is the digit by digit binary representation of the decimal number. For example the number $(9527)_{10} = (1001\ 0101\ 0010\ 0111)_{BCD}$. From the given example it can be observed that there is a binary code for each digit of the decimal number and then concatenated to form the BCD representation of the given decimal number. All arithmetic and logical operations should be defined to use the BCD representation. As the decimal number system contains 10 digits i.e., 0 to 9, in order to represent a BCD digit at least 4 bits are required. Considering a decimal digit A, the BCD representation is given by $A_4A_3A_2A_1$. The only point of note is that the maximum value that can be represented by a BCD digit is 9. The representation of $(10)_{10}$ in BCD is $(0001\ 0000)_2$.

Addition in BCD can be explained by considering two decimal digits A and B with BCD representations as $A_4A_3A_2A_1$ and $B_4B_3B_2B_1$ respectively. In the conventional algorithm, a 4-bit binary adder has been used to add the given input numbers. When two decimal numbers are added, an overflow condition can occur when the resultant sum exceed 9. When an overflow condition occurs, the resultant sum is added with binary equivalent of 6 to obtain the correct BCD representation. This is due binary representation of the BCD numbers. The four binary number moves in the range from 0000 to 1111 whereas BCD ranges from 0000 to 1001. In order to compensate the range 1010 to 1111 in binary into BCD addition of 6 is introduced in the BCD addition computation.

III. LITERATURE REVIEW

In this section, an overview is given for reduced delay BCD adder [13], modified reduced delay BCD adder [14]. The previously proposed BCD adders are implemented in Verilog HDL and synthesized in order to make a comparison with the proposed BCD adder. The synthesis results of the adders are presented in results and discussion session. The block diagram of the reduced delay BCD adder [13] is shown in figure 1. The reduced delay BCD adder consists of Adder, Analyzer, Carry network block and sum correction block. Carry Look Ahead adder is used to add the two BCD input. In order to determine whether an output of CLA is equal to or greater than 9, an Analyzer block is used. Analyzer block gets Sum output from CLA and generates two signals namely Digit Generate (DG) and Digit Propagate (DP). Digit Generate (DG) will be high if the Sum output of CLA is greater than 9 and Digit Propagate (DP) will be high if Sum output is exactly 9. The equations for DG and DP are given below.

$$DG = \text{Carry Out} + \text{Sum} [3]. (\text{Sum} [1] + \text{Sum} [2]) \quad (1)$$

$$DP = \text{Sum} [3]. \text{Sum} [0] \quad (2)$$

The carry value for each digit is computed inside the Carry network using the following equation.

$$\text{Output carry} = (DG + DP). C_{in} \quad (3)$$

The carry network can be of any parallel prefix network. The carries computed by Carry network are used in the correction step. A 4-bit binary adder is used for correction. The modified reduced delay BCD adder [14] is the modified version of the reduced delay BCD adder. There are two major differences

between reduced delay BCD adder and modified reduced delay BCD adder: First, in Adder, Analyzer block an extra input is given which is connected to C_{in} in the first stage and for subsequent stages this input is grounded. Second, C_{in} is no more used to correct the sum output of first stage adder. In existing algorithm [13] to generate the carry for each digit, a carry network has been used in addition of two BCD decimal numbers. The number levels in carry network will increase when the number of digits in given BCD digits increases. Increase in the levels of carry network will in turn increases the delay in calculating the final carry out. In the proposed adder, three signals named CG, CP, and P will be computed using input operands. CG is the carry generate signal, CP is the carry propagate signal and P is the Predict signal determines the sum of two corresponding digits is greater than or equal to 9. The final carry output C_{in}^{i+1} can be determined using the above three signals without using the carry network. In order to determine whether carry is suppressed in the current digit location or propagated to the next digit, the signal P of the current digit and signal P from previous digit can be used. The output of the carry network and the output of the carry suppression or propagation logic is ORED to generate the value which will be used in the correction step.

The equation for value (V) used in the correction step is as follows:

$$V = (C_{in}^{i+1}) + (P_i \& P_{i-1} \& C_{in}) \quad (4)$$

C_{in}^{i+1} : Output carries to next digit, P_i : Signal P to next digit, P_{i-1} : Signal P from previous digit, C_{in} : carry input from previous digit. The digital logic which implements the above algorithm is discussed in the following section.

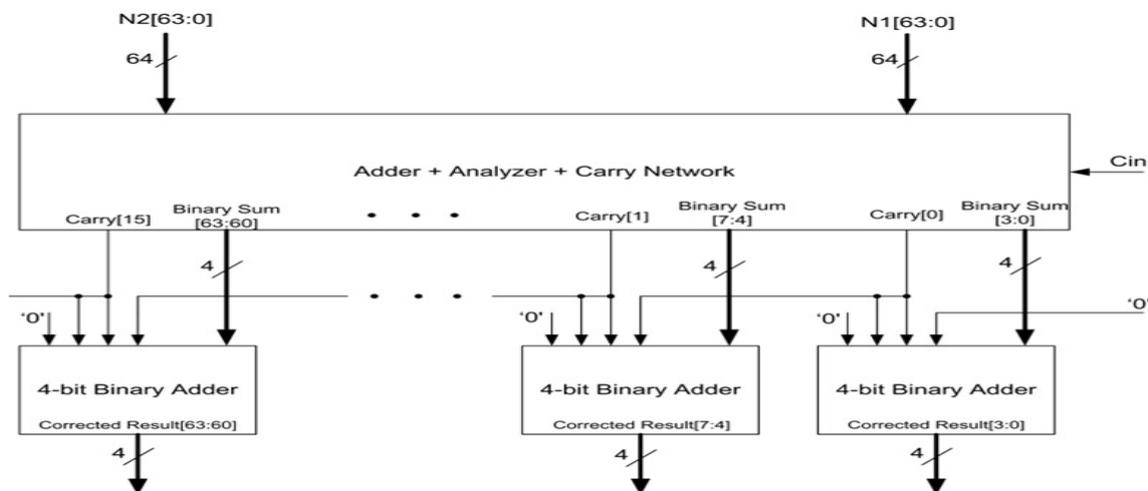


Fig. 1 Reduced Delay BCD adder

IV. HIGHER VALENCE BCD ADDITION

The existing conventional BCD adder [9] is simple in operation, but very slow due to the ripple carry effect. In BCD additions following cases are considered:

- Case 1: The sum of two BCD digits is smaller than 9.
- Case 2: The sum of two BCD digits is greater than 9.
- Case 3: The sum of two BCD digitals is exactly 9

For the first two cases, the incoming carry has no effect on determining the output carry. Therefore, the carry output will be independent of carry input. On the other hand for case 3, the output carry will depend on the input carry. Case 2 and Case 3 can be represented by a Carry Generate (CG), Carry propagates (CP) and P signals, respectively. The computation of the CG, CP and P signals of a digit are shown in Fig 2. The signals CG, CP and P are calculated using the Sum and Carry outputs of the CLA adder in first stage using following equations.

$$CP = \text{SUM}[3].\text{SUM}[0] \quad (5)$$

$$CG = \text{Cout} + (\text{SUM}[3].(\text{SUM}[2] + \text{SUM}[1])) \quad (6)$$

$$P = \text{SUM}[3].(\text{SUM}[2] + \text{SUM}[1] + \text{SUM}[0]) \quad (7)$$

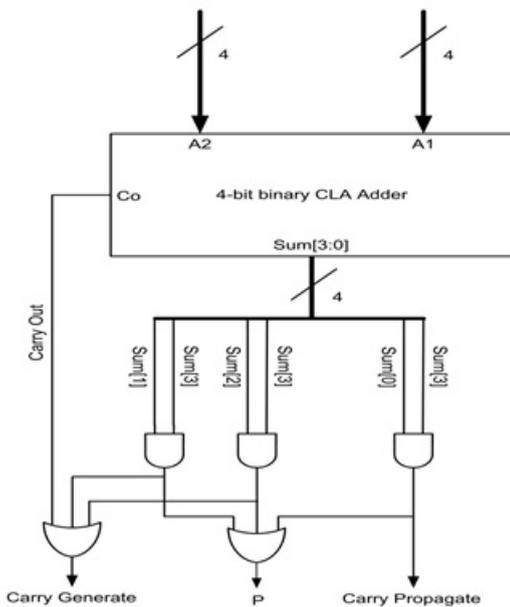


Fig. 2 Carry generate and carry propagate

In High speed BCD adder, the carry input is no more connected to the adder block as in modified reduced delay

BCD adder [14] instead; it is directly connected to the carry look ahead block which will generate carry to next stage.

Because of the above modification the first stage adders will not depend on the carry input. Also the output carry will be independent of the PG logic. Output carry from CLA block is calculated from the following equation.

$$C_{i+4} = CG_{i+4} + P_{i+4}(CG_{i+3} + P_{i+3}(CG_{i+2} + P_{i+2}(CG_{i+1} + P_{i+1}C_{in}))) \quad (8)$$

Where, $i = 0, 4, 8, \dots$

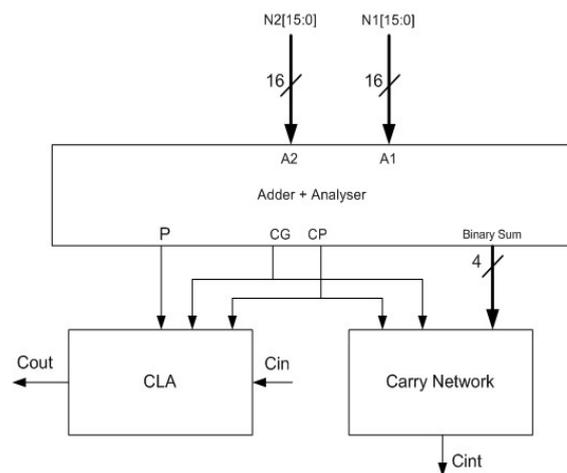


Fig. 3 Adder and analyzer block.

Carry network can be a parallel prefix network which performs their operations in a constant time irrespective of the length of inputs. In this paper Kogge-Stone [8] prefix network is used as carry network. The output of the carry network is used for correction of the output. Carry suppressor will take in P value from the present digit, P value from the previous digit and the carry input and determine whether to suppress the carry or to propagate to the next digit. Carry suppressor also generates the correction value which is to be added to the sum value to convert it back to the BCD equivalent. The integrated block diagram of Adder, Analyzer, CLA and Carry network is shown in Figure 2.

The correction to the sum output of first stage adder is done by adding 0, 1, 6 or 7 to it. As in [13], the generation of correction value will depend on the present digit carry output and previous digit carry output.

Figure 3 shows the block diagram for High Speed BCD adder which includes CLA adder in the first stage, combination of PG logic, CLA to generate next stage carry and Carry

suppressor in the second stage and Correction logic in third stage.

Figure 4 shows the block diagram of the proposed reduced delay BCD adder.

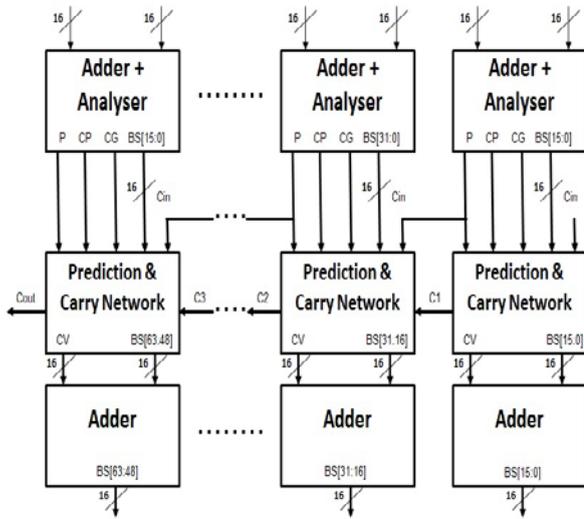


Fig. 4 Higher valence BCD adder

V. RESULTS AND DISCUSSION

The reduced delay BCD adder [13], the modified reduced delay BCD adder [14] and High speed BCD adder has been designed to support 64-bit and 128-bit decimal addition with BCD operands. All the adders are designed using Verilog HDL and the designs are optimized in synopsys design compiler using TSMC 65nm library. The results of 64-bit and 128 bit presented in table 1 and table 2 respectively.

When Reduced Delay BCD (RBCD) adder is compared for delay with Modified Reduced Delay BCD adder (MRBCD) and High Speed BCD adder (HSBCD), delay decreases by 2.2% and 58.5% respectively.

When Modified Reduced Delay BCD adder is compared with High Speed BCD adder delay decreases by 57.9%. In terms of area, when Reduced Delay BCD (RBCD) adder is compared, Modified Reduced Delay BCD adder (MRBCD) and High Speed BCD adder (HSBCD), area increases by 2.6% and 13.1% respectively.

When Modified Reduced Delay BCD adder is compared with High Speed BCD adder area increases by 10.3%. Power calculations show that Reduced Delay BCD (RBCD) adder shows an increase of 3.3% and 9.7% respectively over Reduced Delay BCD adder (MRBCD) and High Speed BCD adder (HSBCD). When Modified Reduced Delay BCD adder is compared with High Speed BCD adder, power increases by 6.1%.

When simulations are repeated for 128 bit design, the results are as in Table 2. In terms of delay, Modified Reduced Delay BCD adder (MRBCD) and High Speed BCD adder (HSBCD), delay decreases by 5% and 39.1% respectively over reduced delay BCD adder (RBCD). When Modified Reduced Delay BCD adder is compared with High Speed BCD adder delay decreases by 35.9%. For area calculations, when Reduced Delay BCD (RBCD) adder is compared, Modified Reduced Delay BCD adder (MRBCD) and High Speed BCD adder (HSBCD), area increases by 0.9% and 7.8% respectively. However, for the same comparison, Modified Reduced Delay BCD adder is compared with High Speed BCD adder area increases by 6.7%. Power calculations show, Modified Reduced Delay BCD adder (MRBCD) and High Speed BCD adder (HSBCD), power increases by 2.3% and 6.4% respectively over RBCD. When Modified Reduced Delay BCD adder is compared with High Speed BCD adder, power increases by 4%.

Table 1 Synthesis results of 64 bit high speed BCD adder and 64 bit Modified Reduced delay BCD adder.

Parameters	Reduced Delay BCD adder	Modified Reduced Delay	High Speed
Gate Count	445	452	556
Area	1187 μ m	1218 μ m	1343.52 μ m
Delay	0.9ns	0.88ns	0.37ns
Dynamic Power	355.41 μ w	365.83 μ w	389.14 μ w

leakage Power	33.2 μ w	35.9 μ w	37.2 μ w
Total Power	388.61 μ w	401.73 μ w	426.34 μ w

Table 2 Synthesis results of 128 bit high speed BCD adder and 128 bit Modified Reduced delay BCD adder.

Parameters	Reduced Delay BCD adder	Modified Reduced Delay	High Speed
Gate Count	942	950	1112
Area	2493 μ m	2517.11 μ m	2687.04 μ m
Delay	1.2ns	1.14ns	0.73ns
Dynamic Power	735.32pW	748.65pW	780.99pW
leakage Power	68.1pW	73.3pW	74.48pW
Total Power	803.42pW	821.95pW	855.47pW

VI. CONCLUSION

This paper describes the design high speed BCD adder to perform decimal addition. The adder has been synthesized with 65nm CMOS library. The new decimal adder improves the delay of BCD addition by increasing parallelism. The adder has been designed using the Verilog HDL and verified for different corner case inputs. The new proposed adder has the shortest delay among the decimal adders examined in this paper.

ACKNOWLEDGEMENT

We would like to thank Vidyand, system administrator for his extended support during our laboratory. We also thank Tanmay M Rao and Arjun Ashok Post Graduate students for their assistance in VLSI Lab.

REFERENCES

- [1] W. Buchholz. Fingers or Fists? (The Choice of Decimal or Binary Representation). Communications of the ACM, 2(12):3–11, December 1959.
- [2] M. F. Cowlshaw. Decimal Floating-Point: Algorithm for Computers. In Proceedings of 16th IEEE Symposium on Computer Arithmetic, pages 104–111, June 2003.
- [3] R.D. Kenney and M.J. Schulte, “Multioperand Decimal Addition,” Proc. IEEE CS Ann. Symp. VLSI, pp. 251-253, Feb. 2004.
- [4] R.D. Kenney and M.J. Schulte, “High-speed multioperand decimal adders,” IEEE Transactions on Computers, Page(s):953 – 963, Volume 54, Issue 8, Aug. 2005.
- [5] M.A. Erle and M.J. Schulte, “Decimal Multiplication via Carry-Save Addition,” Proc. IEEE 14th Int’l Conf. Application-Specific Systems, Architectures, and Processors, pp. 348-358, June 2003.
- [6] P. Parhami, Computer Arithmetic: Algorithms and Hardware Designs. New York: Oxford Univ. Press, 2000.
- [7] I. S. Hwang. High Speed Binary and Decimal Arithmetic Unit. United States Patent, (4,866,656), September 1989.
- [8] Neil H.E. Weste, David Harris and Ayan Banerjee, Cmos Vlsi Design: A Circuits And Systems Perspective, 3ed, Pearson Education, 2006.
- [9] M. M. Mano. Digital Design, pages 129–131. Prentice Hall, third edition, 2002.
- [10] M. S. Schmookler and A.W. Weinberger. High Speed Decimal Addition. IEEE Transactions on Computers, C-20:862– 867, Aug. 1971.
- [11] B. Shirazi, D. Y. Y. Young, and C. N. Zhang. RBCD: Redundant Binary Coded Decimal Adder. In IEEE Proceedings, Part E, No. 2, volume 136, pages 156–160, March 1989.
- [12] J. D. Thompson, N. Karra, and M. J. Schulte. A 64-Bit Decimal Floating-Point Adder. In Proceedings of the IEEE Computer Society Annual Symposium on VLSI, pages 297– 298, February 2004.
- [13] Sundaresan C, Chaitanya CVS, Dr PR Venkateswaran, Dr Somashekara Bhat, Mohan Kumar J, “High Speed BCD Adder”, Springer Journal of

Advances in Intelligent and Soft Computing ISSN: 1867-5662, ISBN #: 978-3-642-28307-9

- [14] Sundaresan C, Chaitanya CVS, Dr PR Venkateswaran, Dr Somashekara Bhat, Mohan Kumar J, “Modified Reduced Delay BCD Adder”,

Proceedings of 4th IEEE International Conference on Biomedical Engineering and Informatics(BMEI'11), 15th – 17th October 2011, organized by Donghua University, Shanghai, China. BMEI Catalog #: CFP1193D-CDR, ISBN #: 978-1-4244-9350-0