A 64 Bit Pipeline Based Decimal Adder Using a New High Speed BCD Adder

Rahul Jain[1], Khushboo Singh[2], Ghanshyam Jangid[3]

[1]Assistant Professor GIT College Jaipur, Rajasthan, India

[2]M.Tech Student, Suresh Gyan University, Jaipur, Rajasthan, India

[3]Assistant Professor, Suresh Gyan University, Jaipur, Rajasthan, India

Abstract: ***Binary arithmetic is one of the most primitive and most commonly used applications in microprocessors, digital signal processors etc. But binary arithmetic is unable to fulfill the requirement of fractional terms thus causing inexact results. And in commercial applications fractional terms are common and efficient output is must requirement so we use Binary Coded Decimal (BCD) adders. The conventional BCD adders are slow due to use of two binary adders. In this paper, we designed and implemented new high speed BCD adders which use only one binary adder. The proposed BCD adder reduces the no. of binary adders due this reduction of adders the propagation delay of BCD adder is reduced. We also implemented 64 bit BCD adder using the pipelined technique. The proposed BCD adders are designed and implemented using verilog HDL in XILINX 9.2 version. The results of conventional BCD adders are compared with proposed BCD adders. the Experimental results demonstrate that the proposed BCD adders has 15.28% faster than conventional BCD adde . The proposed 64 bit pipelined BCD adders is 55.39 % faster than conventional 64 bit BCD adder.***

Keywords: Computer arithmetic, Decimal additions, VLSI design, flagged binary adder, Correction circuit, pipeline, FPGA.

1. Introduction

Binary addition is one of the most primitive and most commonly used applications in computer arithmetic .With the rapid growth of decimal arithmetic in many applications such as commercial, financial, or internet field. The use of simplest and easy method of decimal arithmetic becomes very important for the designers and users. In the past decades, although the binary arithmetic is widely used in processors or any other applications, but some problem occurred in performing some binary arithmetic operations. First is the fraction numbers cannot be represented by using the binary numbers. E.g., 0.710 = 0.10111…..2, it will require infinite bits for representation, so it become incorrect decimal fractions. This incorrect representation of decimal fraction causes approximation errors.

And second is ,Financial database contain decimal data if we are using binary hardware then first decimal data is converted into binary and after computation result which is in binary from it again convert in decimal data these conversion will increase the propagation delay. So, to overcome this drawback of binary arithmetic, the Binary Coded Decimal numbers is used. In BCD, each bit of decimal numbers 0 to 9 uses four bits 00002 to 10012. 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.

To further reduce power and latency in BCD addition an new BCD adder is proposed using flagged binary addition 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. But due to the use of the multiplexer the propagation delay is increased [13].

To reduce the limitation of this BCD adder we proposed a new BCD adder which speed is fast then these BCD adders .This paper is organized as follows. In section II, the normal BCD adder, in section III flagged logic BCD adder are briefly reviewed. Section IV describes the proposed high speed BCD adders. Section V describes the proposed 64 bit pipelined BCD adder .section VI describes the implementation results and the detailed comparison of all types BCD adders. Finally, section VII concludes this paper.

2. Normal BCD adder

In a BCD adder suppose we have two input **X** and Y are given to BCD adder architecture is shown in figure 1. After using these first ripple adders which composed by 4 consecutive full adders to add the values of input X and Y, the digit adder with correction which is also composed by 4 full adders is used. When result of sum is more than 9 then we add $(0110)_2$ in each nibble by using correction network. Correction values 0110, is determined by the output of $c + (S[3] \cdot S[2]) + (S[2] \cdot S[1])$, But the BCD adder is very simple, but also very slow due to the carry ripple effect. It also used two binary adders first to add input and second is used to add correction value in the output of first binary adder due to this reason it increases propagation delay and area.

Licensed Under Creative Commons Attribution CC BY

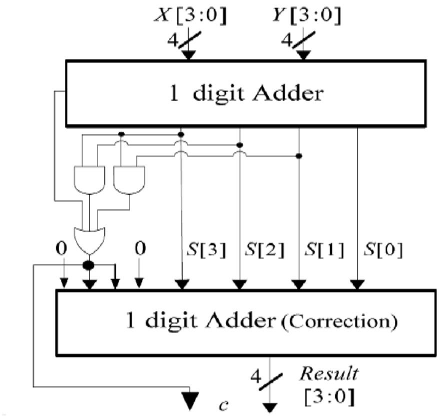


Figure 1: Normal BCD Adder

3. Flagged BCD Adder

To reduce the limitation of normal BCD adder a new flagged BCD adder was designed. 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 2. 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. 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)*

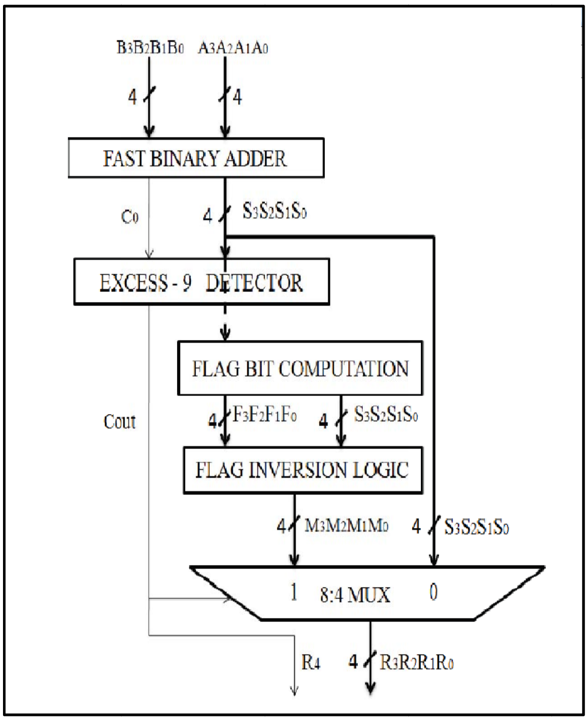


Figure 2: Flagged logic based BCD Adder

The carry bits*(d4d3d2d1)* and sum *S(S3S2S1S0)* are then used by this block to generate flag bits *(F0,F1,F2,F3)* The flag bits*(F0,F1,F2,F3)* and sum *S(S3S2S1S0)* are passed through flag inversion logic 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*.

The flagged BCD adder outperformed all other previous designs in terms of delay and area. The BCD adder also has some limitation like it used a multiplexer in final stage which increased the propagation path so propagation delay is also increased.

4. Proposed BCD Adder

We see in flagged logic BCD adder that it uses a multiplexer in final stage which increased the propagation delay. To reduce the limitation we proposed a new architecture of BCD adder which is more efficient than conventional BCD adder in the term of speed. The architecture of proposed BCD adder is given in Figure 3.

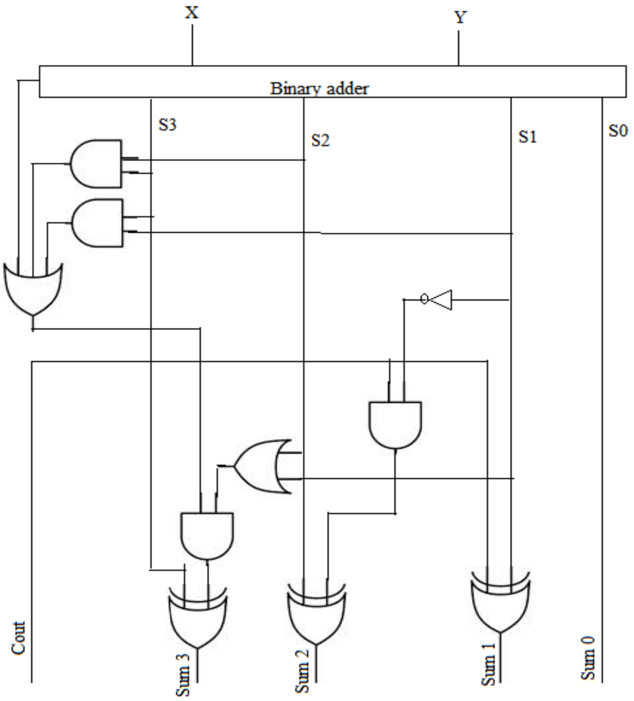
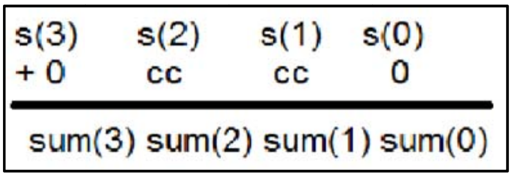


Figure 3: Proposed BCD Adders

The basic ideas for proposed new BCD adder is that in normal BCD adder second binary adder is use to add correction bit. The correction bits always become either 0(0000) or (0110).

When the correction bits are fix either 0 or 6 why we use second ripple adder. We can design a new logic which automatically convert binary sum in BCD sum .we can say that the correction bits add in binary sum without using binary adder. Let we designed the logic. Suppose the output of correction logic is cc and sum bits of first binary adder is

(S (3), s (2), s (1), s (0)) Then



s(3)	s(2)	s(1)	s(0)
+ 0	cc	cc	0
sum(3)	sum(2)	sum(1)	sum(0)

Figure 4: Logic implementation

Sum [0] = s [0]; ----------------- (1)
Sum [1] = s [1] ^cc; ---------------- (2)
Sum [2] = (cc& (~s [1])) ^ s [2]; -- (3)
sum[3]=(cc&(s[1]|s[2]))^s[3]; -- (4)

where “^” means xor gate,”~” means not gate,| means or gate and “&” means and gate.

We can see in proposed BCD adder that the second binary adder is replaced my new designed logic. This logic reduces ripple effect so the propagation delay is reduced and speed of BCD adder is increased.

We also analysis that in this no. of gates are also reduced like a ripple adder uses total four three input xor gates for sum and total twelve and gates for carry generation. But proposed logic uses only three two input or gates, and two and gates, one or gate and not gate. We can say it is more efficient then previous BCD adders. Proposed 64bit pipelined BCD adder basically pipelining is a well known techniques for improving the performance of digital systems. Pipelining exploits in combinational logic in order to improve throughput. In this technique we split it into two or more than two separate block of combination logic. And blocks are connected with a register. Each block has about less delay then original block. Because each block has its own registers .all block operate independently. Working on two values at the same time. We have reduced the cycle time of the machine because we have cut the maximum delay through the combinational logic. This technique is used in our proposed 64 bit BCD adder. First we design 64bit BCD adder using proposed new high speed BCD adder by connecting them in series. But due the connect them in series the propagation delay is increased. It is due to ripple effect. The architecture of this BCD adder is given in the

64 bit BCD Adder

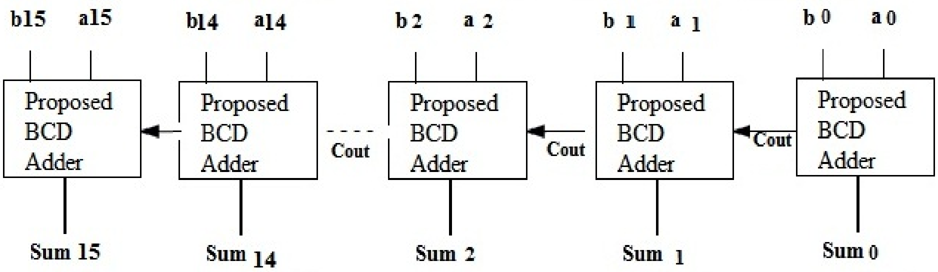


Figure 5: 64 bit BCD Adder

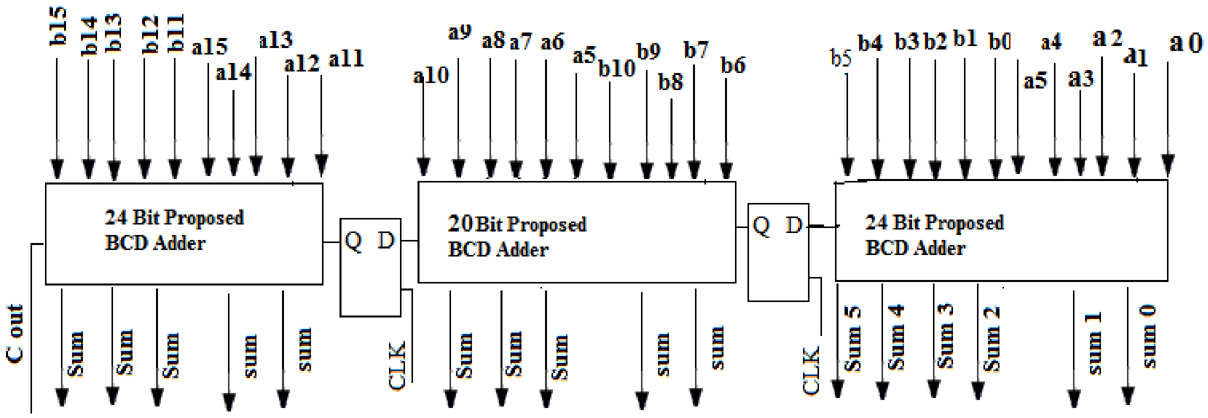


Figure 6: Proposed Pipelined 64 bit BCD Adder

To reduce the ripple effect we designed a new pipelined based 64 bit BCD adders which architecture is given in figure 6. We split 64 bit BCD adder in three blocks. They block are independently. In first combination block there are six BCD adders to compute first six inputs. The second and third block has five BCD adders to compute further inputs. Now the first block final carry signal is connected to a d flip flop which output is connected to the carry input of second block similar the third and second block connected through a d flip flop. The each block has less propagation delay then main 64 bit BCD adder. Each block operates independently. Working on three values at the same time. The right hand block would start computing for a new input while the left hand other blocks complete the function for the value started at the last cycle. Furthermore, we have reduced the cycle time of the machine because we have cut the maximum delay through the small combinational logic.

Licensed Under Creative Commons Attribution CC BY

5. Results and Comparison

We evaluate the performance of conventional and modified BCD adders and implement them on virtex – 5 FPGA families. For Design Entry and delay report we synthesize these adders using Xilinx ISE 9.2i. We use verilog as hardware description language.

The technology schematic of proposed high speed BCD is shown in figure 7. The technology schematic of proposed 64 bit pipelined BCD adder is shown in figure 8.

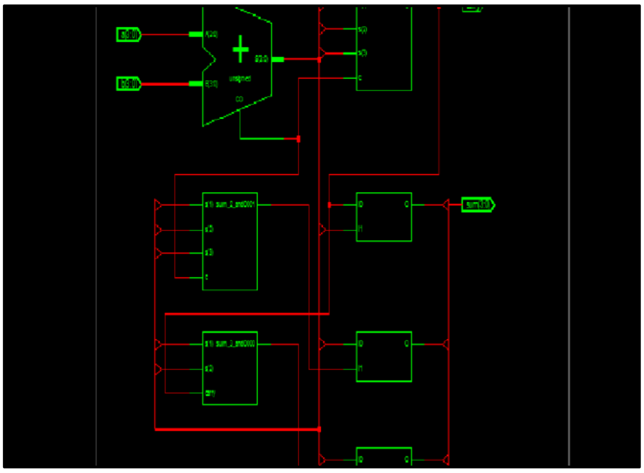


Figure 7: RTL schematic of Proposed BCD Adder

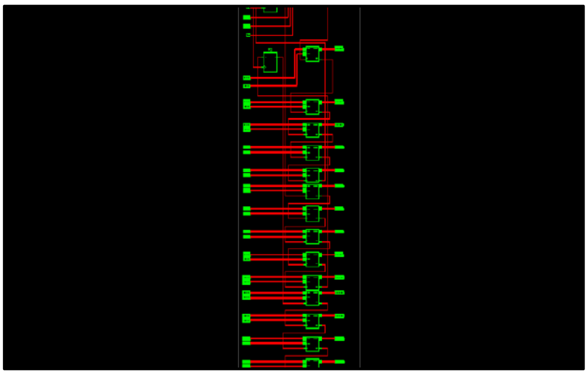


Figure 8: RTL schematic of 64 bit Proposed pipelined BCD Adder

The simulation waveform is also shown in figure 9 and 10. Where A and B is input and sum is BCD sum. We use behavior, data flow and gate level modeling in designing in verilog. We also design and implemented all conventional BCD adders.

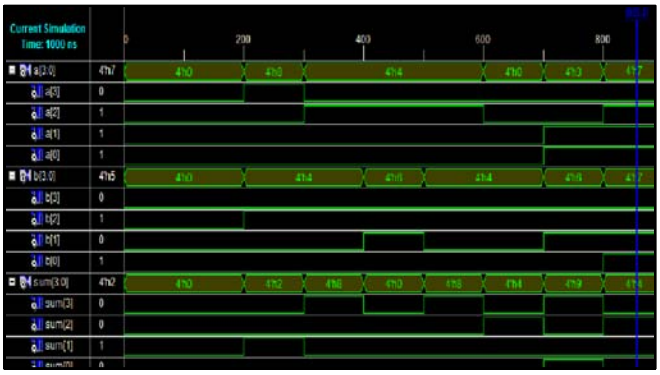


Figure 9: Simulation of proposed BCD Adder

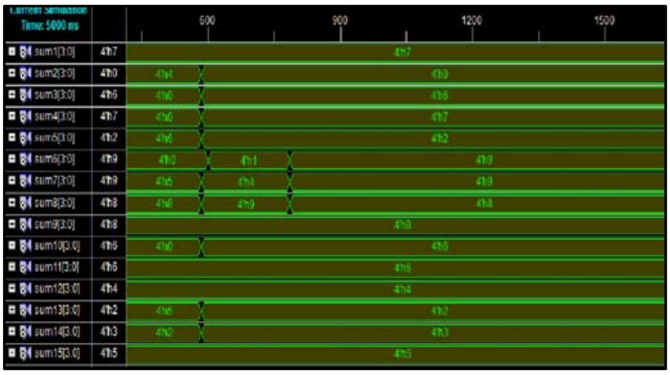


Figure 10: Simulation of proposed 64 bit pipelined BCD Adder

The proposed high speed BCD adder has propagation delay is 4.430ns and conventional normal BCD adder has delay 5.229ns and flagged logic based BCD adder has delay 5.040ns. The proposed is 15.28% faster than conventional normal BCD adder.

The proposed 64 bit pipeline based BCD adder has propagation delay has 12.801ns and without pipelined concept the propagation delay is 28.701ns. Experimental results also demonstrate that the proposed 64 bit pipelined BCD adders is 55.39 % faster than conventional 64 bit BCD adder. The comparison of conventional and modified BCD adders is shown in following table I. It shows that proposed BCD adders are more efficient in the term of speed then conventional BCD adders

Table 1: Comparison of all BCD Adder

S. No	Type of BCD adder	Propagation delay
1	Normal BCD adder	5.229ns
2	Flagged logic based BCD adder	5.040ns
3	Proposed BCD adder	4.430ns
4	64 bit BCD adder	28.701ns
5	64 bit pipelined BCD adder	12.801ns

6. Conclusion

In this paper, the conventional and modified proposed BCD adders are designed using Verilog. The delay of modified BCD adders is less as compared to the conventional BCD adders. We use a new logic to add the correction bits in binary sum which is faster than conventional adder .it increase the speed of BCD adder. Pipeline technique also reduces the propagation delay by increasing throughout. When implemented on FPGA, the result proved that the proposed booth BCD adder is 15.28% faster than conventional normal BCD adder and pipeline based 64 bit BCD adder is 55.39 % faster than conventional 64 bit BCD adder. The proposed approach also applies with minor modifications to three input decimal addition.

7. Future Scope

The future scope of the paper is that we can designed for three input or further inputs which will be faster than conventional decimal adders . We can also use state machine approach to increase the speed. For power reduction we can use clock gating technique.

Licensed Under Creative Commons Attribution CC BY

References

[1] Tso-Bing Juang,Hsin-Hao Peng,Chao-Tsung Kuo."Area efficient BCD Adder,"19th IEEE/IFIP International Conference on VLSI Design,2011

[2] Draft IEEE Standard for Floating-Point Arithmetic. New York: IEEE, Inc., 2004,

[3] Alp Arslan Bayrakci and Ahmet Akkas. Reduced Delay BCD Adder. IEEE, 2007.

[4] T. Lang, and A. Nannarelli, "Division Unit for Binary Integer Decimals," Prof. 20th IEEE International Conference on Application-specific Systems, Architectures and Processors (ASAP), pp. 1-7, 2009.

[5] T. Lang and A. Nannarelli, "A Radix-10 Digit-Recurrence Division Unit: Algorithm and Architecture," IEEE Transactions on Computers,

[6] Vol. 56, No. 6, pp. 727-739, June 2007.

[7] R. K. James, T. K. Shahana, K. P. Jacob, and S. Sasi, "Decimal multiplication using compact BCD ultiplier," Proc. International Conference on Electronic Design (ICED), pp. 1-6, 2008

[8] P. M. Kogge and H. S. Stone. A Parallel Algorithm for The Efficient Solution of a General Class of Recurrence Equations.IEEE Trans. on Computers, C-22(8), Aug. 1973.

[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. SchulteB. A 64-Bit Decimal Floating-Point Adder. In Proceedings of the IEEE Computer Society Annual Symposium on VLSI, pages 297– 298, February 2004.

[13] +B. Prashanthi kumari, G. N. V. Ratna Kishor, “New Approach for Implementing BCD Adder Using Flagged Logic” ternational Journal of Engineering Research & Technology IJERT) Vol. 2 Issue 12, December - 2013

Author Profile

Khushboo Singh is a final year student of M.Tech (DD) EC+VLSI FROM Gyan Vihar School of engineering and Technology Jaipur, Rajasthan, India.

Ghanshyam Jangid received the B.E. in electronics and communication from Global Institute of Technology Jaipur, Rajasthan, India in 2008 and M.Tech in VLSI Design from MNIT Jaipur in 2013.currently Assistant Professor at Gyan Vihar in EC department.

Rahul Jain received the B.E. in Electronics and Communication from Arya Engineering College Jaipur in 2009. Presently working as Asst. Professor in Global Institute of Technology, Jaipur, Rajasthan, India

Licensed Under Creative Commons Attribution CC BY