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

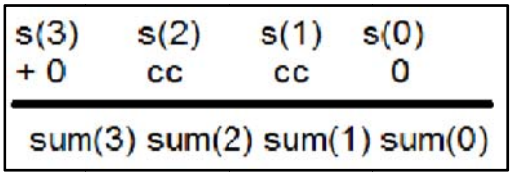

Figure 4: Logic implementation

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

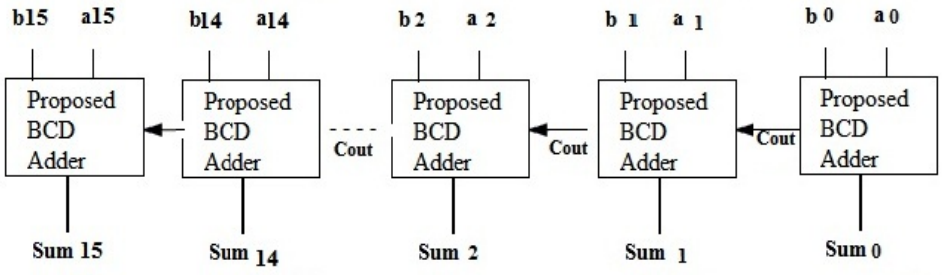

Figure 5: 64 bit BCD Adder

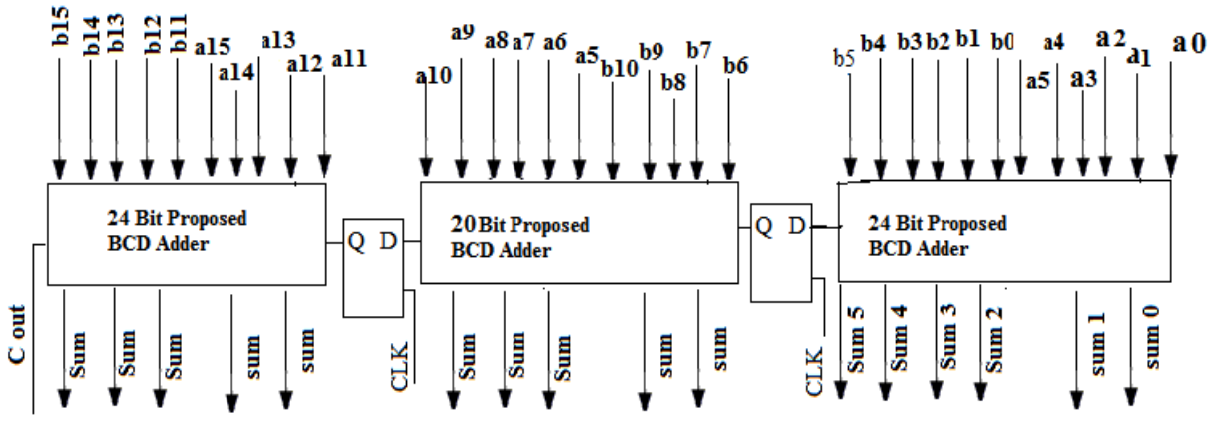

Figure 6: Proposed Pipelined 64 bit BCD Adder

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.

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.

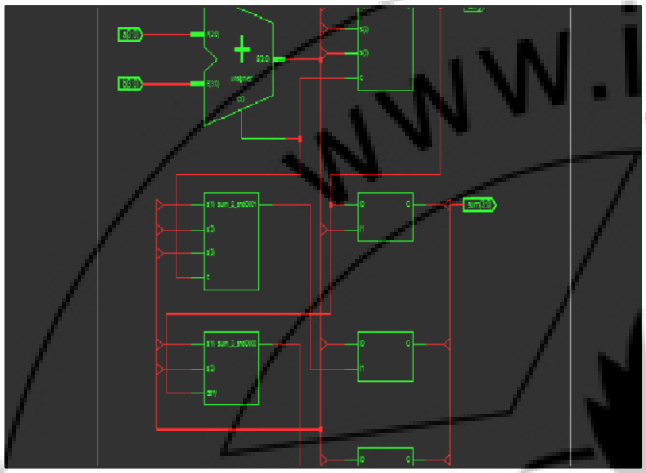

Figure 7: RTL schematic of Proposed BCD Adder

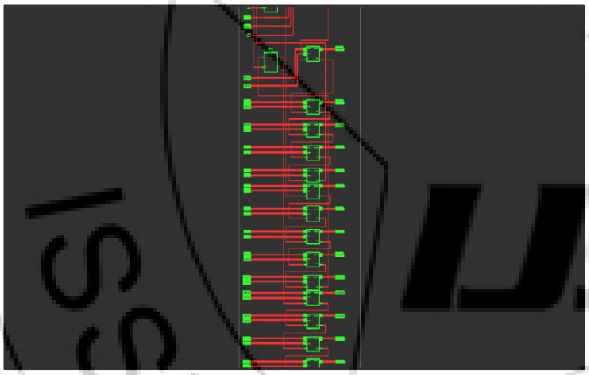

Figure 8: RTL schematic of 64 bit Proposed pipelined BCD Adder

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.

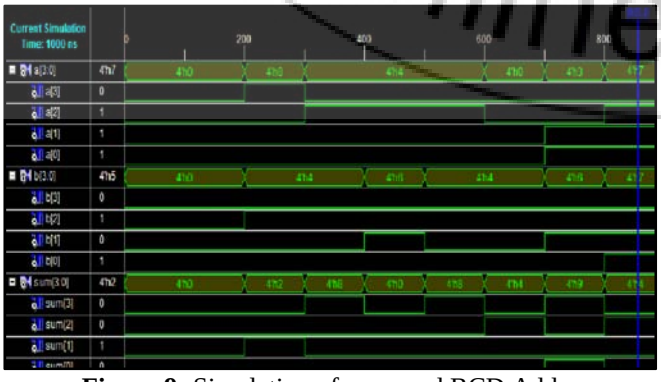

Figure 9: Simulation of proposed BCD Adder

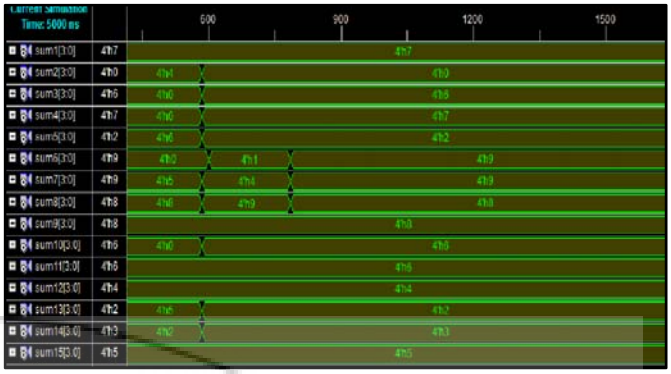

Figure 10: Simulation of proposed 64 bit pipelined BCD Adder

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.

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