ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

# EFFICIENT LOW POWER FIR FILTER DESIGN

**Rashmi K. Bhojane*, Prof. U. W. Hore.**
*M.E scholar, Department of Electronics and Telecomm, P.R Patil COE & M,SGBAU, Amravati, India
Assistant Professor, Department of Electronics and Telecomm, P.R PatilCOE &M,SGBAU, Amravati, India

## ABSTRACT

Design of area and power-efficient high-speed data path logic systems are one of the most substantial areas of research in VLSI system design. In digital adders, the speed of addition is limited by the time required to propagate a carry through the adder. Carry Select Adder (CSLA) is one of the fastest adders used in many data-processing processors to perform fast arithmetic functions. From the structure of the CSLA, it is clear that there is scope for reducing the area and power consumption in the CSLA. This work uses a simple and efficient gate-level modification to significantly reduce the area and power of the CSLA. The CSLA is used in many computational systems to alleviate the problem of carry propagation delay by independently generating multiple carries and then select a carry to generate the sum. However, the CSLA is not area efficient because it uses multiple pairs of Ripple Carry Adders (RCA) to generate partial sum and carry by considering carry input Cin = 0 and Cin = 1, then the final sum and carry are selected by the multiplexers (mux). This is modified by replacing the RCA with Cin=1 with BEC in the regular CSLA to achieve low area and power consumption. The performance of this CSLA is evaluated by implementing an FIR Filter by using the CSLA in the adder part. This work focuses on the performance of CSLA in terms of delay and power and it is found that CSLA is a high speed and low power adder.

**KEYWORDS**: **:** Multipliers, CSLA, RCA, low power.

## INTRODUCTION

Area and power reduction in data path logic systems are the main area of research in VLSI system design. High speed addition and multiplication has always been a fundamental requirement of high-performance processors and systems. In digital adders, the speed of addition is limited by the time required to propagate a carry through the adder. The sum for each bit position in an elementary adder is generated sequentially only after the previous bit position has been summed and a carry propagated into the next position. The major speed limitation in any adder is in the production of carries and many authors have considered the addition problem.

The CSLA is used in many computational systems to moderate the problem of carry propagation delay by independently generating multiple carries and then select a carry to generate the sum [2]. However, the CSLA is not area efficient because it uses multiple pairs of Ripple Carry Adders (RCA) to generate partial sum and carry by considering carry input and then the final sum and carry are selected by the multiplexers (mux). To overcome the above problem, the above CSLA is modified by using n-bit Binary to Excess-1 code converters (BEC) to improve the speed of addition. This logic can be implemented with any type of adder to further improve the speed. We use the Binary to Excess-1 Converter (BEC) instead of RCA in the regular CSLA to achieve lower area and power consumption [1], [3]-[4], [7]. The main advantage of this BEC logic comes from the lesser number of logic gates than the Full Adder (FA) structure. The modified design has reduced area and power as compared with the regular SQRT CSLA with an increase in the delay.

This is briefed as follows. First we deal with the delay, area and power evaluation methodology of the basic adder blocks. Next we present the detailed structure and the function of the BEC logic. The CSLA has been chosen for comparison

with the modified and improved design. The delay and power evaluation methodology of the regular, modified and improved SQRT CSLA are presented.

Therefore the main aim of the project is to design and implement a high speed carry select adder to enhance the speed of addition and perform fast arithmetic functions. The proposed design is applied to the FIR filter structure in the adder part to evaluate the performance of the proposed design. This work estimates the performance of the proposed design in terms of delay and power.

## PROPOSED WORK

### *A. Ripple Carry Adder*

This is the simplest type of adder but bot very efficient when large number of bits are used. Delay increases linearly with bit length.

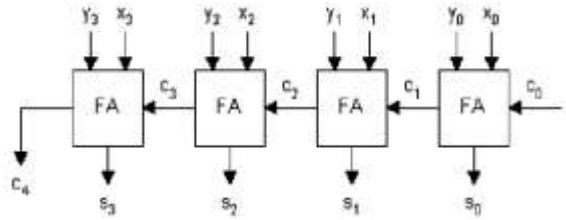


**Figure 1: Block diagram of Ripple carry adder**

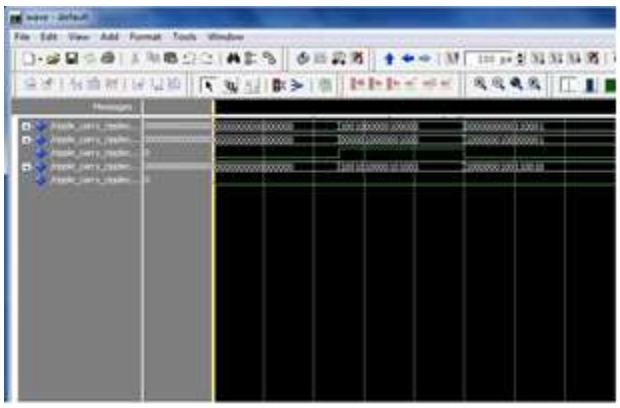

**Figure 2: Simulation Output**

### *B. 16-BIT REGULAR CARRY SELECT ADDER*

A Carry Select Adder is a particular way to implement an adder, which is a logic element that computes the (n+1) bit sum of two n-bit numbers. The carry-select adder is simple but rather fast. The carry-select adder generally consists of two ripple carry adders and a multiplexer. Adding two n-bit numbers with a carry-select adder is done with two adders (therefore two ripple carry adders) in order to perform the calculation twice, one time with the assumption of the carry being zero and the other assuming one. After the two results are calculated, the correct sum, as well as the correct carry, is then selected with the multiplexer once the correct carry is known. It is just simply a 16- bit full adder in which we have two 16 bit input with one carry in and a 16 bit sum output with a single bit carry out.

The structure of a 16 bit CSLA is shown below:

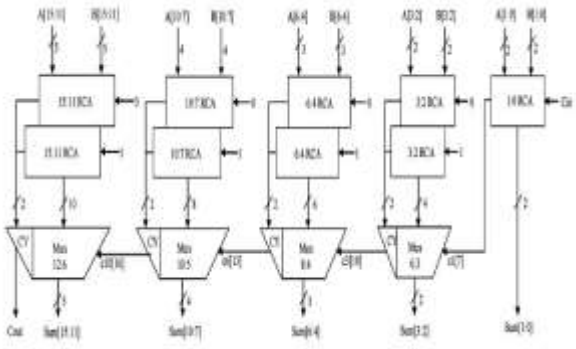

**Figure 3 -bit Regular CSLA**

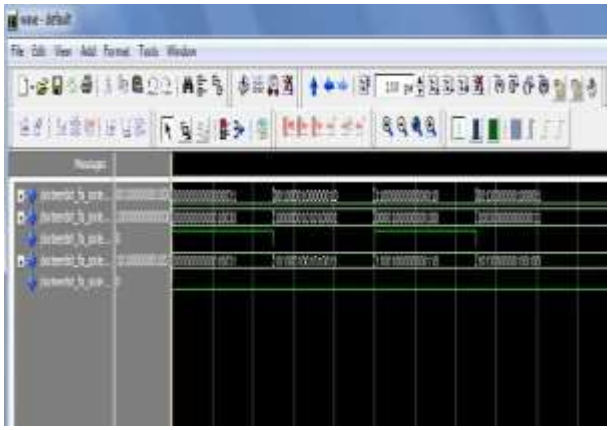

**Figure 4: Simulation Output**

### *C. Wallace Tree Multiplier*

A Wallace tree is an efficient hardware implementation of a digital circuit that multiplies two integers. The WT multiplier sums up all the bits of the same weights in a merged

tree rather than completely adding the partial products in pairs. Full adder (FA) and Half adder (HA) cells are used to add three or two equally weighted bits respectively to produce two bits: the sum bit with a weight equal to that of the operands and the carry bit with a weight equal to one more than that of the operands. The height of the WT is reduced by a factor of 3:2, whenever a FA is used. The final tree is composed of as many levels of FA and HA cells as are necessary to reduce the height of the tree to 2. The hardware synthesis process for a WT multiplier mainly consists of two steps. The first step is to arrange the partial product bits as the initial WT structure, as shown in Fig. 2 for the case of a 4x4 multiplier with operands (a3; a2; a1; a0) and (b3; b2; b1; b0). Secondly, a series of FA and HA transformations are applied on the WT structure until the tree height is reduced to 2. At this point, any n-bit conventional adder may be used to add the remaining two n-bit rows of the tree to get the final multiplication result.

ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785

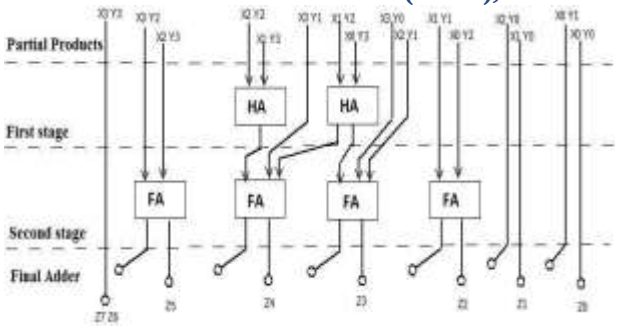


*Figure 5: Block Diagram of Wallace Tree Multiplier*

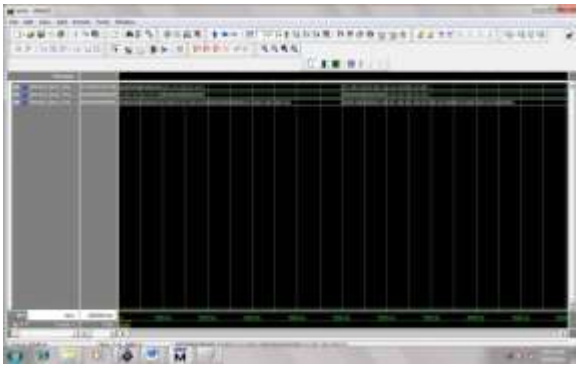

*Figure 6: Simulation Output*

*D. **Array Multipliers***

Array multiplier is well known due to its regular structure. Multiplier circuit is based on add and shift algorithm. Each partial product is generated by the multiplication of the multiplicand with one multiplier bit. The partial product are shifted according to their bit orders and then added.

The addition can be performed with normal carry propagate adder. N-1 adders are required where N is the multiplier length.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | A3 | A2 | A1 | A0 | Inputs |
| | | | x | B3 | B2 | B1 | B0 | |
| | | | C | B0 x A3 | B0 x A2 | B0 x A1 | B0 x A0 | |
| | | + | B1 x A3 | B1 x A2 | B1 x A1 | B1 x A0 | | |
| | | C | sum | sum | sum | sum | | |
| | + | B2 x A3 | B2 x A2 | B2 x A1 | B2 x A0 | | | Internal Signals |
| | C | sum | sum | sum | sum | | | |
| + | B3 x A3 | B3 x A2 | B3 x A1 | B3 x A0 | | | | |
| C | sum | sum | sum | sum | | | | |
| Y7 | Y6 | Y5 | Y4 | Y3 | Y2 | Y1 | Y0 | Outputs |

ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785

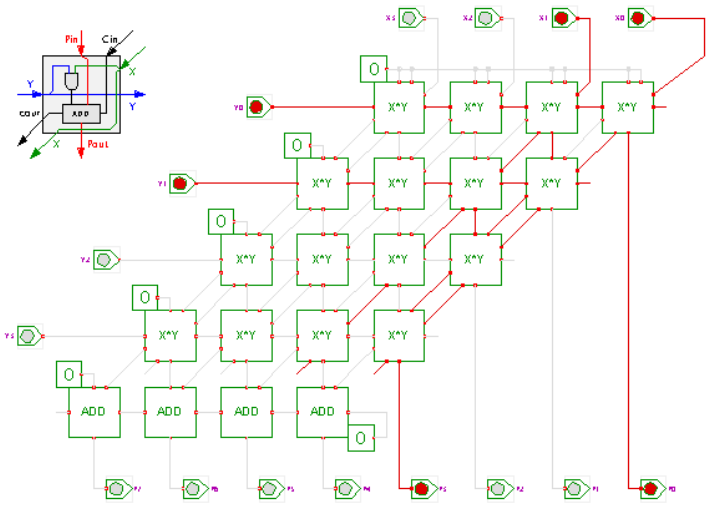


***Figure 7: Simulation Output***

Now as both multiplicand and multiplier may be positive or negative, 2's complement number system is used to represent them. ***If the multiplier operand is positive*** then essentially the same technique can be used but care must be taken for sign bit extension

The reason for dealing with signed number incorrectly is the absence of sign bit expansion in this multiplier.

## CONCLUSION

All the models of CSLA are designed and are implemented in vhdl using Xilinx tool and the results are compared in terms of delay and power. There are different adders among compared them by different criteria like Area, Time and then Area-Delay Product etc. so that we can judge to know which adder was best suited for situation. After comparing all we came to a conclusion that Carry Select Adders are best suited for situations where Speed is the only criteria. Similarly Ripple Carry Adders are best suited for Low Power Applications. But Among all the Carry Select Adder Adder had the least Area-Delay product that tells us that, it is suitable for situations where both low power and fastness are a criteria such that we need a proper balance between both as is the case with our Project.

## REFERENCES

1. Zhijun Huang,High level optimization techniques for low power multiplier design University of California, los angels, 2003.
2. L. Ciminiera, P. Montuschi, "Carry-Save Multiplication Schemes Without Final Addition", IEEE Transaction on Computer, vol. 45, no. 9, Sep. 1996.
3. A. D. Booth, A signed binary multiplication technique", Quart. J. Mechanical and Applied Math. vol.4, pp. 235240, 1951,
4. W.-C. Yeh and C.-W. Jen, High-speed booth encoded parallel multiplier design, IEEE Transactions on Computers, vol. 49, pp. 692701, 2000.
5. B. S. Cherkauer, E. G. Friedman, "A Hybrid Radix- 4/Radix-8 Low Power Signed Multiplier Architecture",IEEE Transaction on Circuits and Systems, vol.44, no. 8,Aug. 1997.

6. H. Lee, ”A power-aware scalable pipelined Booth multiplier,” in Proc. IEEE Int. SOC Conf., 2004, pp.123-126.
7. Tisserand, ”Low-power arithmetic operators,” in Low Power Electronics Design, C. Piguet, Ed. CRC Press,Nov. 2004, ch. 9
8. Nagendra, M. J. Irwin, and R. M. Owens, ”Area-timepower tradeoffs in parallel adders,” IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing, vol. 43, no. 10, pp. 689-702,oct 1996.
9. Z. Huang and M. D. Ercegovac, ”High-performance low- power left- toright array multiplier design”, IEEE Transactions on Computers, vol. 54, no. 3, pp. 272- 283,Mar. 2005.
10. P. F. Stelling, C. U. Mattel, V. G. Oklobdzija, and R. Ravi, Optimal Circuits for Parallel Multipliers, IEEE Transactions on Computers, vol. 47(3): 273 - 285, 1998.
11. W. C. Yeh, and C. W. Jen, High-Speed Booth Encoded Parallel Multiplier Design, IEEE Transactions on Computers, vol. 49 (7), pp. 692-701, 2000.
12. J. A. Gibson and R. W. Gibbard, Synthesis and Comparison of Twos Complement Parallel Multipliers, IEEE Transactions on Computers, vol. C-24, pp 10201027, October 1975.
13. C. S. Wallace, Suggestion for a Fast Multiplier, IEEE Transactions on Electronic Computers, vol. EC-13, pp. 1417, 1964.
14. W. Gallagher and E. Swartzlander. High Radix Booth Multipliers Using Reduced Area Adder Trees. In Twenty-Eighth Asilomar Conference on Signals, Systems and Computers, volume 1, pages 545549, 199
15. E. Costa, J. Monteiro, and S. Bampi. A New Architecture for Signed Radix 2 Pure Array Multipliers. In IEEE International Conference on Computer Design, pages 112117, 2002.
16. K.H. Tsoi, P.H.W. Leong, "Mullet - a parallel multiplier generator," fpl, pp.691-694, International Conference on Field Programmable Logic and Applications, 2005., 2005.
17. S. Tahmasbi Oskuii, P. G. Kjeldsberg, and O. Gustafsson, "Transition activity aware design of reduction-stages for parallel multipliers," in Proc. 17th Great Lakes Symp. On VLSI, March 2007, pp. 120-125.
18. Ayman A. Fayed, Magdy A. Bayoumi, "A Novel Architecture for Low- Power Design of Parallel Multipliers," vlsi, pp.0149, IEEE Computer Society Workshop on VLSI 2001, 2001

## AUTHOR BIBLIOGRAPHY

| | |
|---|---|
|  | **Rashmi K. Bhojane** received Bachelor of Engineering degree in Electronics Engineering from RTMNU Nagpur & Pursuing Master of Engineering in Electronic and Telecommunication from P.R.Patil ,College of Engineering and management , Amravati. |
|  | **Prof. U. W. Hore** has received M.E degree from SGB, Amravati University. Currently he is working as Assisnt Professor EXTC Dept.,inP.R.Patil, College of Engineering and Management, Amravati. |