MANTECH Publications

Journal of Power Electronics and Drives
Volume 7, Issue 3, September-December 2022

Wallace Tree Adders with High Speed and Power Efficiency

Rushabh Shethia[1], Roshan Rajpure[2], Harsh Shinde[3]

Assistant Professor[1], Students[2,3]

Department of EEE

Saraswati College of Engineering, Kharghar

Email: *roshanrajpure8565@yahoo.com[2]*

Abstract

In this study, Wallace Tree Adders and FIFB, FIEB, and FISB Carry Save Adders are designed, Verilog-encoded, and simulated using Cadence Software. The implementation of adders takes place on CMOS chips with a 180 nm process. Power consumption, latency, silicon area, and dynamic power dissipation simulation results are compared. Both the Carry Save Adder and Wallace Tree Adder see a rise in the number of inputs, power consumed, silicon area, and latency. The suggested Wallace Tree Adder is shown to be more cost-effective and a better solution for real-time applications than the typical CSA since it has a shorter delay, less power dissipation, and requires less silicon space.

Keywords: *Verilog, Wallace tree based adders, Power consumption, Carry save adders, Silicon area, delay, Dynamic power dissipation*

INTRODUCTION

The demand for the popularity of portable devices is pushing designers to aim for a smaller surface area, faster processing rates, longer battery life, and improved dependability. When building a system, a designer aims to conserve as much power and delay as possible. Compressors, comparators, and parity checkers are just a few of the circuits that use full adders as their most basic building blocks. These adders may also be used in ALU, digital signal processing, counters, graphic processors, address calculation, and code compression. System performance as a whole can be considerably improved by improving the performance of the full adders. The data path uses about 30% of

the system's overall power. Adders are a frequently used part of the data stream, so careful design and analysis of adders are necessary.

So far, full adders have been created using a variety of logic methods. Every design has benefits and drawbacks. Standard static CMOS full adders are one instance of such a design. Because the PMOS block has less mobility than the NMOS devices, it is the primary flaw in static CMOS circuits. As a result, seizing up PMOS devices is necessary to get the desired performance. However, the dynamic CMOS logic offers a faster operating speed. This study compares and contrasts the size, power consumption, and delay times of Wallace tree-based and carry-save adders used in 180nm dynamic CMOS technology. Cadence Software was used to simulate the Verilog code. We find that Wallace tree adders outperform conventional carry-save adders in terms of complexity and timing. The Wallace Tree Adder's biggest drawback is its atypical construction, which makes planning challenging. Furthermore, this architecture is wasteful because all adder blocks are active regardless of the size of the multiplicand. Since fewer adders are required on the critical path than with the Carry Save Adder, the Wallace Tree Adder is more economical and has a shorter delay across the critical path. The design is not only quicker, but it also takes up less space and is consequently less expensive to produce because it employs fewer adders than the traditional carry save adder design. A further advantage of using fewer transistors is that the device will use less power as fewer transistors will be required. A Wallace Tree multiplier does have one drawback, though, and that is that because of its atypical design, manufacturing it effectively is challenging. However, the advantages of employing a Wallace Tree Multiplier significantly outweigh the slight increase in manufacturing difficulty.

The study methodology for the work provided in the paper is discussed in Section II, which also looks into the drawbacks of using standard methods to develop full adders. The implementation and operation of the proposed Wallace tree adders and the four input four bit (FIFB), four input eight bit (FIEB), and four input sixteen bit (FISB) carry save adders are described in Section III. Additionally, it examines the suggested adder's power, area, and performance (delay) difficulties in comparison to the CSA adder. Section IV concludes by outlining the conclusions drawn from the outcomes of the simulation.

RESEARCH METHODOLOGY

The basic building block of complicated arithmetic circuits, including addition, multiplication, division, exponentiation, etc., is the adder. It acts as a speed limiter. In the past, to develop full-adder cells, numerous standard implementations with different logic approaches were employed. Despite having comparable functions, they all have different production methods and transistor counts. It is necessary to optimise, and this can be done at the circuit level or at the logic level. The boolean equations can be rearranged to optimise the logic level. We can create a circuit that is quicker and smaller after logic level optimization. The speed, size, power dissipation, and wiring complexity of a circuit are all impacted by the implementation design. The number of inversion levels, the number of transistors connected in series, the transistor sizes (i.e., channel widths), and the intra-cell wire capacitances all affect the circuit delay. The quantity, dimensions, and degree of wiring complexity of the transistors all affect the size of the circuit. Switching activity, node capacitances (composed of gate, diffusion, and wire capacitances), and control circuit size all affect power dissipation. In increasing order of speed performance, there are certain addition strategies that are frequently employed in numerous addition operations, such as ripple carry, carry look ahead, and carry save adders.

The idea of Carry save adder is to take 3 numbers that we want to add together, x + y + z, and convert it into 2 numbers c + s such that x + y + z = c + s. In carry save addition, we refrain from directly passing on the carry information until the very last step. Computation of these three numbers has divided into two steps, In first steps CSA is used to compute S and C, then CPA is used to compute the total sum. The delay can thus be reduced by using carry save adder.

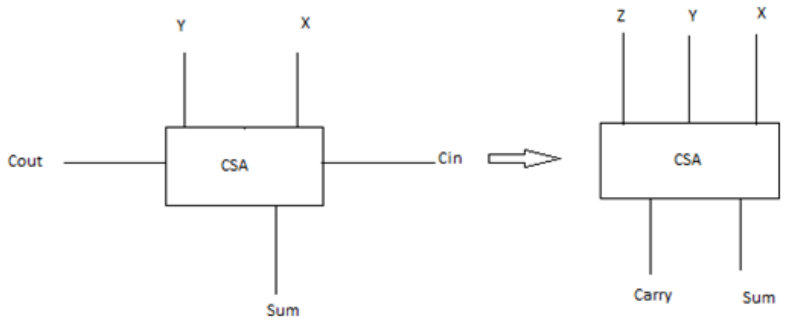

Figure 1: Carry save Adder

Proposed Work

In this section, we use the Verilog language for 180nm CMOS technology to design and construct four input four bit, four input eight bit, and four input sixteen bit carry save adders as well as proposed Wallace tree adders architectures. Cadence Software is used to simulate the Verilog codes, which are meant for the ASIC design. Adder simulation is typically used for optimization and verification.

A. Carry Save Adder:

1) Architecture of FIFB Adder

In FIFB Adder four inputs A, B, C and D each of four bits are added. The process of addition is divided into three steps. In first step A and B are added as shown in figure 2. Next C and D are added as shown in figure 3. Finally, the results of first and second steps are added to get the final output which is shown in figure 4.

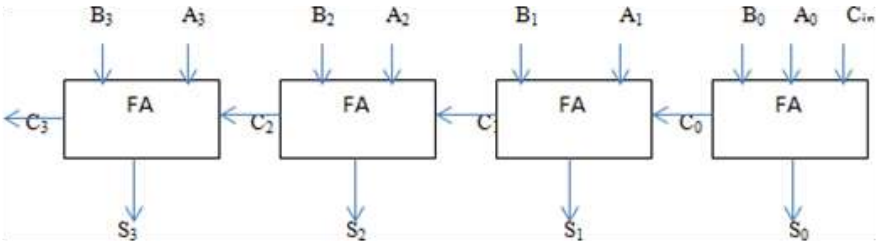

Figure 2: A and B addition Block

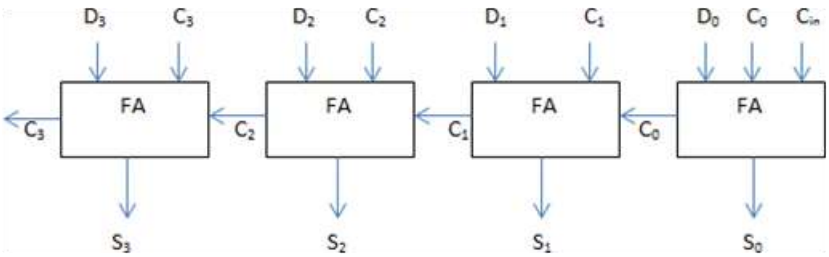

Figure 3: C and D addition Block

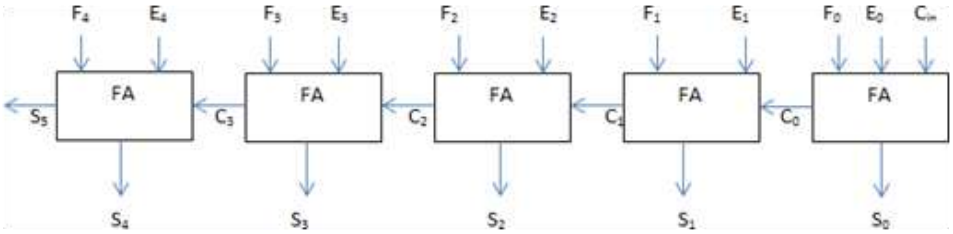

Figure 4: Addition block to add results of addition of A, B and C, D

Architecture of FIEB Adder

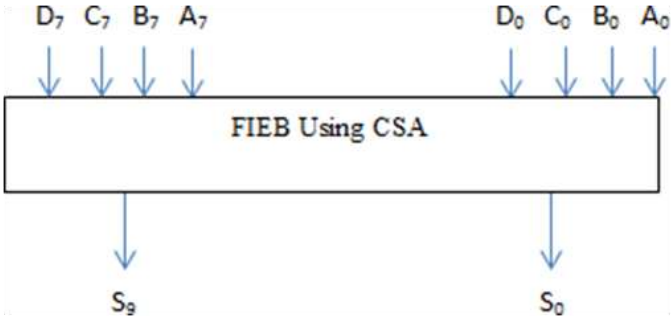

Figure 5: FIEB Adder Using CSA

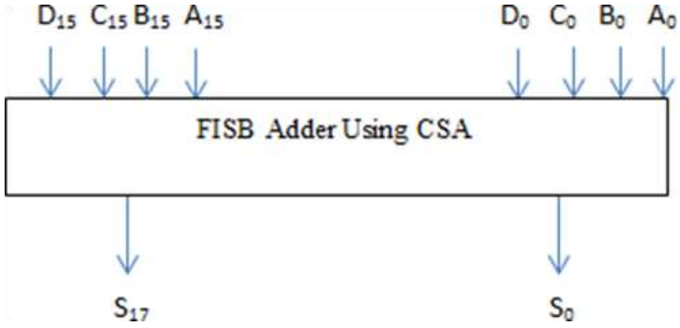

Figure 6: FISB Adder Using CSA

In FIEB Adder four inputs A, B, C and D each of eight bits are added in figure 5. The addition process is divided into three steps. In first step A and B are added, next C and D are added, and the results of first and second are added to get the output.

Architecture of FISB Adder

In FISB Adder four sixteen bit inputs A, B, C and D in figure 6. In first steps A and B are added next C and D are added, and finally the results of first and second step are added using CPA (carry propagation adder) to get the final outcome.

Wallace Tree Adders

The proposed designs for the four input 4-bit, four input 8-bit, and four input 16-bit Wallace Tree Adder are shown in Figures 8, Figure 9, and Figure 10, respectively. The addition of four inputs is handled in parallel in this system. CSA and CPA are also used to implement Wallace tree-based adders. A series of connections between half adders and full adders is used to create the CPA and CSA. In order to prepare the design for ASIC implementation, simulated cadence software and Verilog are used.

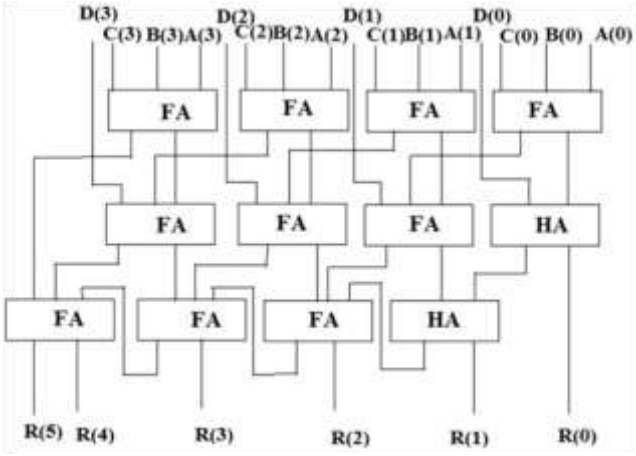

Figure 8: Four input four bit Wallace Tree Adder

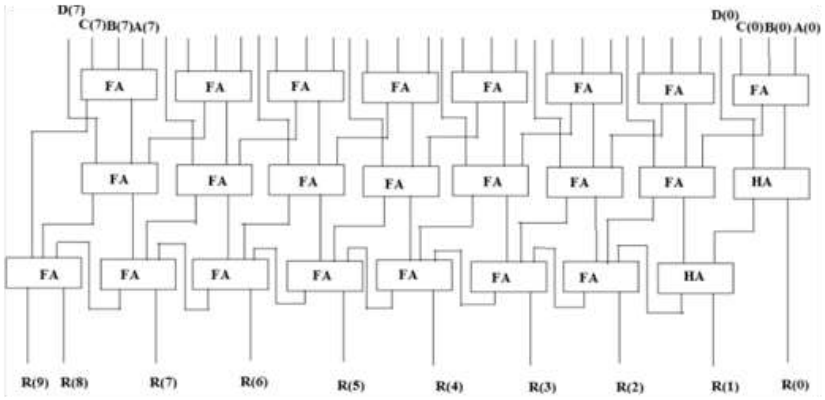

Figure 9: Four input sixteen bit Wallace Tree Adder

Simulation Results and Discussion

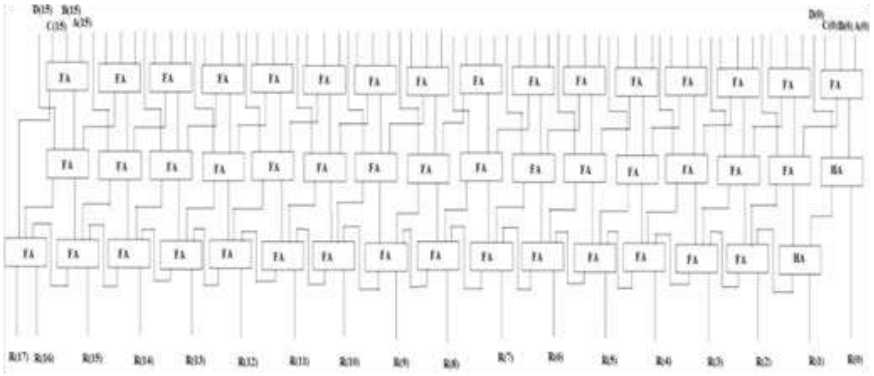

Figure 10: Four input eight bit Wallace Tree Adder

The ASIC implemented Verilog codes for the designed architecture are simulated using Cadence Software and the simulation results obtained are analysed in this section.

Carry Save Adders

Table 1: Result Analysis Carry Save Adder

Results Analysis CSA				
Adders	**Area(μm2)**	**Delay(ps)**	**Power(μw)**	**Dynamic Power(μw)**
FIFB	866	1025	171.54	170.61
FIEB	1645	2066	323.15	322.46
FISB	3311	2987	691.11	692.33

Wallace Tree Adders

Table 2: Results Analysis Wallace Tree Based Adder

Results Analysis Wallace tree based adder				
Adders	**Area(μm2)**	**Delay(ps)**	**Power(μw)**	**Dynamic Power(μw)**
FIFB	772	951	133.68	132.98
FIEB	1620	1053	315.23	315.10
FISB	3213	2290	689.30	689.20

Analysis: The power dissipation in the three Carry Save Adders is depicted in Table 1. As length of inputs increases, the power dissipated also increases. For the case of delay in the three Carry Save Adders, it's also directly proportional to the length of the inputs. The silicon area and dynamic power also shows the same tendency, with the increase in length of three Carry Save Adders they also get increased. The result also suggests that the silicon area, delay, power dissipation and dynamic power are getting nearing to double as the number of bits increasing from 4 bits to 8bits, and 8 bits to 16 bits.

Comparison of Carry Save Adders And Wallace Tree Adder

Figures 8, Figure 9, Figure 10, and Figure 11 show the relative power dissipation, silicon area, delay induced, and dynamic power dissipation for the FIFB, FIEB, and FISB Carry Save and the Four Input, Eight Input, and Sixteen Input Wallace Tree Adders, respectively. The comparison results imply that with shorter input lengths to Wallace Tree Adders, the power dissipation and dynamic power decrease in higher order. Even when the number of inputs is increased, the Wallace Tree Adder consistently reduces power dissipation and dynamic power. The results of the comparison of silicon area usage between the CSA and WTA likewise show a considerable reduction in the area used for the WTA. The relative delay results for CSA and WTA highlight the Wallace Tree Adders' observable high- speed nature. [3] The comparison results demonstrate that there will be a trade-off

between the carry saving adder and the Wallace tree adder's area and power. According to comparisons with other literature, the current proposed additive outperforms others. The delays demonstrate how long it takes for the procedure to complete. WTA performs worse than CSA at the same frequency. Depending on how quickly they process information, different adder architecture types have varying delays.

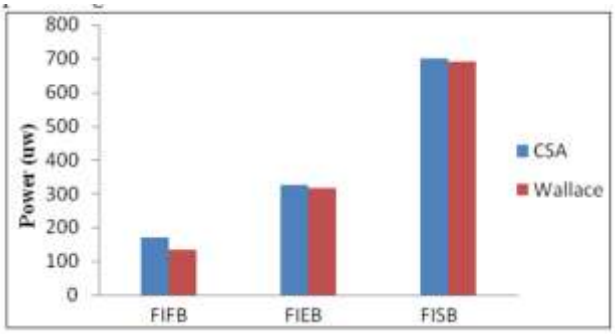

Figure 11: Power Dissipation in CSA and WTA

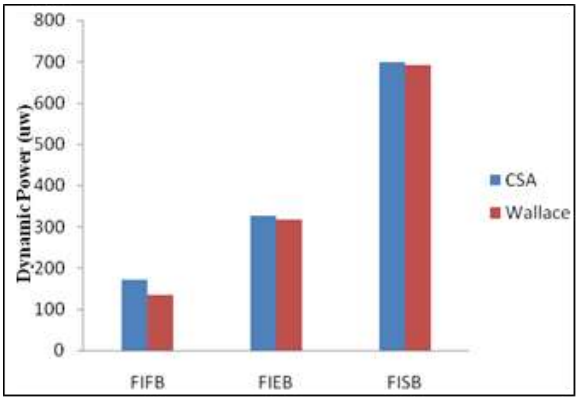

Figure 12: Dynamic Power Dissipation in CSA and WTA

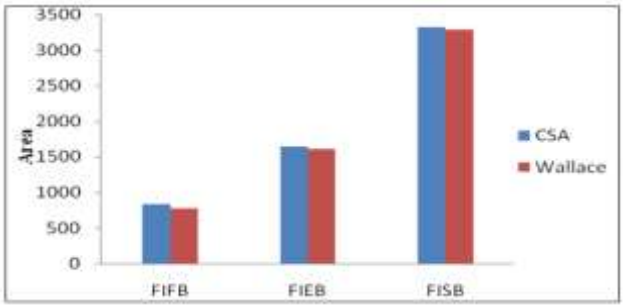

Figure 13: Silicon Area in CSA and WTA

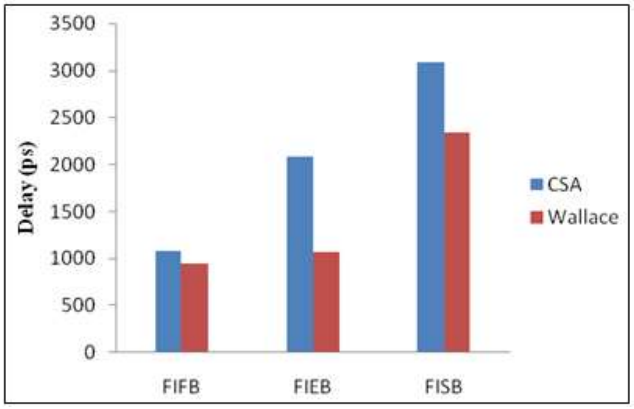

Figure 14: Delay in CSA and WTA

CONCLUSION

It is discovered that Wallace tree adders significantly outperform typical carry-save adders in terms of complexity and timing. Wallace tree adder's uneven structure, which makes planning challenging, is its biggest drawback. The Wallace Tree Adder is more economical and has a shorter delay across the crucial path than the Carry Save Adder because fewer adders are required on the critical path. The design is faster, takes up less space, and costs less to produce since it employs fewer adders than the traditional carry save adder design. Another advantage of utilising fewer adders is that the device requires fewer transistors and uses less power as a result. However, a Wallace Tree Adder has one drawback: due of its atypical design, it is challenging to construct effectively. However, the advantages of using a Wallace Tree Adder outweigh the slight increase in production difficulty. Wallace tree adders are not just swift, but also less energy and space-hungry.

REFERENCES

1. Maroju Sai Kumar, Dr. P. Samundiswary, “Design and Performance Analysis of Various Adders using Verilog”, International Journal of Computer Science and Mobile Computing, pp. 128-138, 2013.

2. R.P.P Singh, Praveen Kumar, Balwindar Singh, “Performance Analysis Of Fast Adders Using VHDL”, International Conference on ARTCC, IEEE, pp-189-193, 2009.

3. Manuel Qrtiz, franciscoquiles, Javier Hormigo, “Efficient Implementation of Carry-Save

Adders in FPGAs", International Conference on ASAP, IEEE, pp-207-210, 2009.

4. Shivani Parmar, Kirat Pal Singh, "Design of high speed hybrid carry select adder", Advance Computing Conference (IACC) 2013 IEEE 3rd international, pp-1653-1663, 2013.

5. Shruti Murgai, Ashutosh Gupta, Gayathri Muthukrishnan, "Energy Efficient and High Performance 64-bit Arithmetic Logic Unit using 28nm Technology", International Conference on Advances in computing, Communications and Informatics, IEEE, pp. 453-456, 2015.

6. Alvaro Vazquez, Elisardo Antelo, "Multi-Operand Decimal Addition by Efficient Reuse of a Binary Carry-Save Adder Tree", International Conference, IEEE, pp-1685-1689, 2010.

7. Vladimir V. Shubin, "Analysis and Comparison of Ripple Carry Full Adders by Speed", International Conference and Seminar on EDM, IEEE, pp-132-135, 2010.

8. Pallavi Saxena, Urvashi Purohit, Priyanka Joshi, "Analysis of Low Power, Area- Efficient and High Speed Fast Adder", International Journal of Advanced Research in Computer and Communication Engineering, pp. 3705-3710, 2013.

9. Arun Kumar P., "Reusable High Speed Architecture Design for Video Transmission in Satellite Applications", Recent Researches in Electrical Engineering, pp. 316-324, 2014.

10. Gopal Paul, Rohit Reddy, C. R. Mandal Bhargab, B. Bhattacharya, "A BDD-based Design of an Area-Power Efficient Asynchronous Adder", International Symposium on VLSI, IEEE, pp-29-34, 2010.

11. Neeraj Jain, Puran Gour, Brahmi Shrman, "A High Speed Low Power Adder in Multi Output Domino Logic", International Journal of Computer Applications, pp.30-33, 2014.

12. Kirat Pal Singh, Shivani Parmar, Dilip Kumar, "Design of High Performance MIPS Cryptography Processor", Proc. 9th International

Conference Heterogeneous Networking for Quality, Reliability, Security and Robustness, Springer LNICST, pp. 778-793, January 2013.

13. Atef Ibrahim, Fayez Gebali, "Optimized structures of hybrid ripple carry and hierarchical carry lookahead adders", Microelectronics Journal, Elsevier, pp. 783-794, 2015.

14. Kirat Pal Singh, Shivani Parmar, "VHDL Implementation of a MIPS-32 bit Pipeline Processor", International Journal of Applied Engineering Research, pp. 1952-1956, 2012.

15. Mark S. K. Lau, Keck Voon Ling, Yun Chung Chu, and Arun Bhanu, "Modeling of Probabilistic Ripple- Carry Adders", International Symposium on Electronic Design, Test & Applications, IEEE, pp- 201- 206, 2010

16. Priya Meshram, Mithilesh Mahendra, Parag Jawarkar, "Designed Implementation of Modified Area Efficient Enhanced Square Root Carry Select Adder",International Journal for Research in Emerging Science and Technology, pp. 96-99, 2015.

17. Shiwani Dod, "Modified Booth Dadda Multiplier using Carry Look ahead adder Design and Implementation", International Journal of Computer Science & Engineering Technology, pp. 86- 93, 2016.

18. Pallavi Saxena, "Design of Low Power and High Speed Carry Select Adder Using Brent Kung Adder", International Conference on VLSI Systems, Architecture, Technology and Applications, IEEE, pp. 1-6, 2015.