



iJRASET

International Journal For Research in
Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 8

Issue: III

Month of publication: March 2020

DOI:

www.ijraset.com

Call:  08813907089

E-mail ID: ijraset@gmail.com

To Design Low Power & High-Speed Adder with the Performance Analysis of different Adders

Sadanand Singh Choudhary¹, Dr. Vikash Gupta

¹M.Tech Scholar, ²HOD, Department of Electronics and Communication Engineering, Technocrats Institute of Technology

Abstract: Adders are the vital components of the CPU (central processing unit) in today's Digital world. They are used in floating point calculations, ALUs, also in order to compute addresses of memory. In other applications like microprocessors and digital signal processors (DSP) architectures full - adders plays important role. The real revolution came into existence when reduction of operating voltage and continuous scaling of the transistor size has led to a predominant enhancement of the integrated circuits(IC). The evaluation started on the premise of the following parameter performances i.e. Power consumption, area, and speed. However they'd presented a layout technique for the hybrid carry skip adders/carry look ahead. The accuracy of the digital devices is mostly determined by the operation of respective adders. The most reviewed constraints for designing the adder are area, power, speed, timing, number of cells and delay.

Keywords: Half Adder, Fast Adders, Carry select Adder, Carry save Adder, Carry Skip Adder.

I. INTRODUCTION

The real revolution came into existence when reduction of operating voltage and continuous scaling of the transistor size has led to a predominant enhancement of the integrated circuits(IC). They play important role in the packaging of FPGA device and fast performing devices with low power consumption, high speed and smaller area.

In Digital Signal Processing (DSP) and Central Processing Unit (CPU) adders are the most normally used arithmetic block, hence optimizing power is of most importance. Speed of a circuit increases rapidly with scaling technology to the depth of sub-micron and also power consumption increases significantly per chip with respect to increase in the density of the chip.

Further, High-speed and low power are the two important factors that needs to be considered in realizing modern (VLSI) Very Large Scale Integration circuits. In case of circuits' design, the low-power adders with high-performance can be given at various different levels, such as in the process technology, the logic style, architecture and layout, The adder is a circuit which performs the summation of two given inputs termed in digital electronics.

In order to perform any operation this is the basic circuit. The adder's are not only used in the different parts of the processor but also used as the part of the Adder. The one which sums the binary digits and notes the digits that are carried in and also out is known as full adder.

In case of full adder the summation operation is done for three one-digit numbers, which are written as X, Y, and Cin. X and Y can be called as operands, and input Cin is a bit which is sent from previous digit. Which actually generates a two bits as output, which are sum and carry that are given as the Cout that is carry out and sum signal output as S. Complex Adders are the one that are combination of full adders and half adders. Many complex adders are out in our digital world out of which some of them are given below:

□ Ripple Carry Adder □ Carry Look Ahead Adder □ Carry Select Adder □ Carry Skip Adder □ Carry Save Adder .

It is the one which performs summation of two single individual binary bits' x and y is an Half adder. It will generate two outputs, carry [C] and sum [S]. When two multiple bits are added the excess which goes into the next bit is the carry signal. For ex, in the above case when both inputs are entered as 1 then it resulted in 10 i.e., here 1 get's shifted to next bit the sum would be 0 and hence carry would be only 1.

The pictured below is the normal half-adder design, in which for generating the sum we use xor gate and for generating carry we use and gate.

The half adder will generate output of a carry and sum by adding two single digit input. The rest of this paper is organized as follows. Section II presents a review of the conventional adder structures. Section III presents the proposed carry select adder architecture. Section IV depicts the ASIC implementation and the results of the proposed adder circuit are analyzed. Section V concludes.

A. Conventional Adder Circuits

1) *Carry Ripple Adder*: As word specifies this adder is the one that ripples the carry to next stage which is sent from the previous bit. It is combination of n full adders. So, the full adder output carry would be provided to the preceding full adder as input. Drawback of this adder is that it must wait for all the previous carry to generate the final carry. The circuit of the carry ripple adder as given below

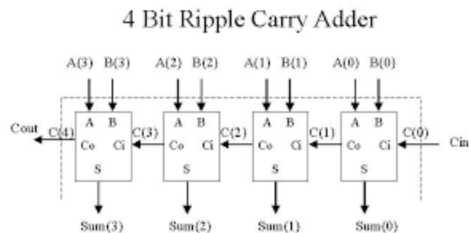


Fig 1 Ripple Carry Adder

2) *Carry look Ahead Adder*: It is another complex adder. Which uses concept of generate the carry and propagate the carry. However, the concept of operation of generating carry and propagating carry is as the following. For carry generate we use $G = A * B$. For carry propagate we use $P = A \text{ xor } B$. For sample given, logic of carry generates (g) and carry propagates (p) equations are provided following. Out of which numbers in below equations are the input signals, initiating from 0 least significant bit to 3 most significant bit: $C1 = G0 + (P0 * C0)$ $C2 = G1 + (P1 * C1)$ $C3 = G2 + (P2 * C2)$ $C4 = G3 + (P3 * C3)$ Further keeping C1 in C2, then C2 in C3 and C3 in C4 we get $C1 = G0 + (P0 * C0)$ $C2 = G1 + (P1 * G0 + (P1 * P0 * C0))$ $C3 = G2 + (P2 * G1 + (P2 * P1 * G0 + (P2 * P1 * P0 * C0)))$ $C4 = G3 + (P3 * G2 + (P3 * P2 * G1 + (P3 * P2 * P1 * G0 + (P3 * P2 * P1 * P0 * C0))))$

a) *Carry Select Adder*: It is another complex adder which is a conditional sum adder. It is constructed with sharing of logic values commonly for sum generation. For sharing them commonly, a inverter with a xor gate are used to perform addition outputs which are given below figure. When input carry-in is be ready, then it can generate accurate sum depending on the input carry-in signal. Further for the output carry, It should use one and gate & one or gate to find out possible input carry values before generating

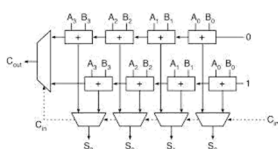


Fig 2 Carry look ahead adder:

3) *Carry Skip Adder*: It is also known as Carry Bypass Adder which is another type of complex adder. The operation of this adder has reduced the delay over ripple carry adder than the other adder. The enhancement of least delay is done using many carry skip adders combine to form a single carry skip-adder block. It has created to increase the operation through summing produced carry digit to the whole adder. The resultant circuit [1] is given in following figure for a 4bit adder. The input carry-in digit is provided as Ci and further the adder produces the output carry-out digit (Ci+4) by itself. Carry skip circuit utilizes a pair of logic gates. The (and) gate is utilized to use the carry-in then finally compares with the propagate signal group.

$$P(i,i+3) = P_{i+3} * P_{i+2} * P_{i+1} * P_i$$

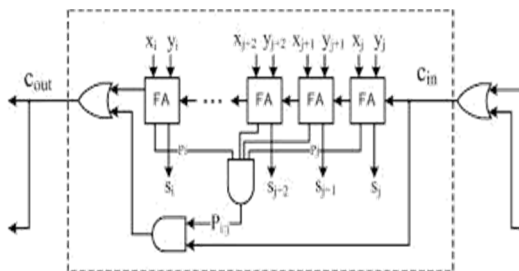


Fig 3 Carry Skip adder:

With help of above results, the generated output of (and) gate provided as input to (or) gate with C_{i+4} to generate final Carry out.
 $Carry = C_{i+4} + P_{(i,i+3)} * C_i$

Carry Save adder:

It is the one with low spread delay [4] (basic way), however it either of summing both input bits to an individual output sum, it sums the three input bits to a two output bits. At last, then its outputs are added to a conventional carry ripple adder or carry look ahead, which will generate the entire inputs output as sum.

While summing at least three bits together, the sequence of carry-save adders [2] is been ended by a individual carry-look ahead adder that gives much preferable propagation delays over the sequence of carry look a-head adder. Specifically, it's spread delay is not influenced by the width of the bits being included

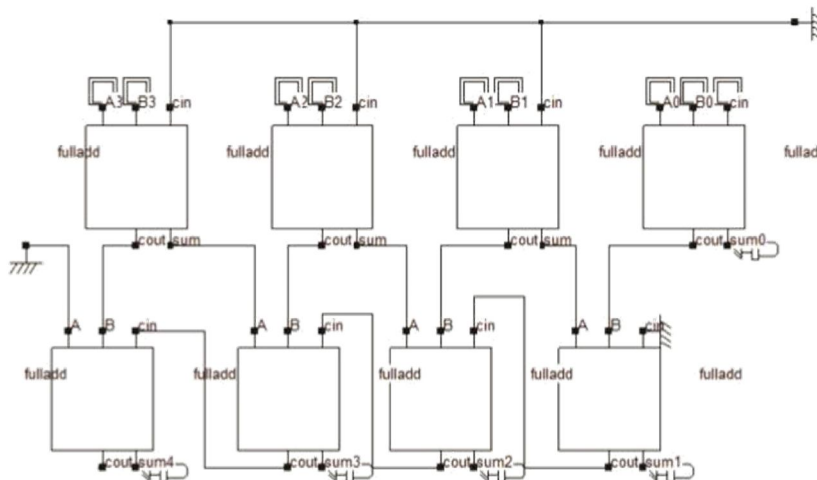


Fig 4 Carry Save Adder

II. LITERATURE REVIEW

Pooja Kansliwal, Mahendra vucha, Rashmi Solanki, Prashant Gurjar (2011)[1]: This paper tells about the equipment usage of the different high speed adders. That are like full adder, carry-look a-head adder, carry_skip adder, carry ripple adder, carry select adder, these are integrated & recreated in the Xilinx-ISE 9.2i stage, whose output parameters caught like region and speed are thought for 16-bit and 8-bit adders

Reena Rani, Laxmi Kanth Singh, Neelam Sharma(2009)[2]: In this paper numeric operations are performed with the help of a greater radix system, for example, Quaternary Signed Digit (QSD). They rely of Quaternary stamped digit structure. In QSD, every bit is addressed by a bit between 3 to -3. Pass on development, intensive operation on broad number of bits, for instance, 64, 128, and higher shall completed using unflinching deferral, less diserse quality. FPGA instruments are used for Hardware implementation of these circuits. The arrangements mirrored using modalism programming and joined with the help of Leonardo Spectrum

Jasbir Kaur, Lalith Sood (2015)[4]: In this paper, the execution of the different adders, for example, the Carry skip adder, the Carry increment adder, Ripple carry adder, the Carry look a-head adder, the Carry select adder , the Carry save adder are talked about, they are analyzed in the premise of their execution parameter's, for example speed, power circulation.

R. P. P Singh, Praveen Kumar, Balwinder Singh (2009)[7]: In this paper, the overall performance analysis of the exceptional fast adders has been accomplished. The evaluation started on the premise of the following parameter performances i.e. Power consumption, area, and speed. However they'd presented a layout technique for the hybrid carry skip adders/carry look ahead. The usage of each fix and the variable block length has changed carry bypass adder.

A. Design Approach

To design these fast adders, we have studied various papers. To implement them we need Verilog code i.e., main module and test bench. After getting the module for respective adders we tried to simulate the module in Xilinx and cadence NCSim tool, at this time we get the simulation result for different input combination. For synthesis of delay and power we use fast.lib and slow.lib where we check the delay and power at 45nm, 90nm, 180nm technologies. So, finally our main aim is to compare these techniques at the different technologies.

III. ASIC IMPLEMENTATION AND RESULT

A. Carry Select Adder

Carry select adder has been implemented in Xilinx and Ncsim using gate level modeling for verification we have taken the results and verified with industry standard cadence tools.

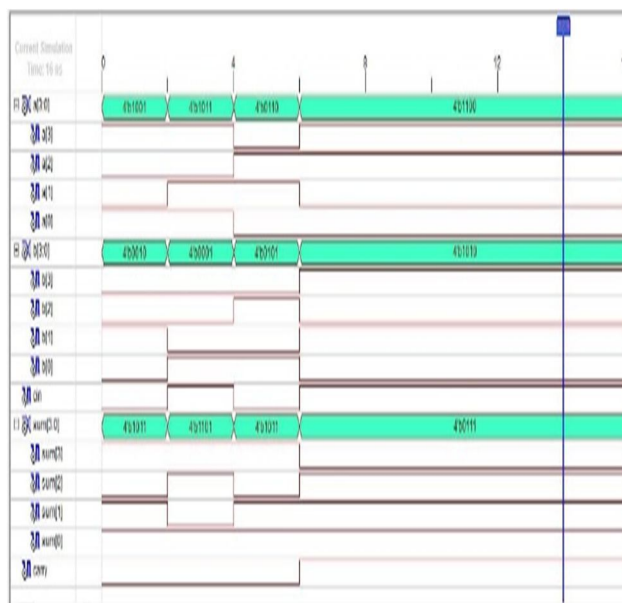


Fig 5 Output waveform of Carry select adder

B. Carry Skip Adder

Carry skip adder has been implemented in Xilinx and Ncsim using gate level modeling for verification we have taken the results and verified with industry standard cadence tools.



Fig 6 Output waveform of Carry Skip Adder

C. Carry Save Adder

Carry save adder has been implemented in Xilinx and Ncsim using gate level modeling for verification we have taken the results and verified with industry standard cadence tools.

IV. PERFORMANCE EVALUATION

Table 1 Comparison between the parameters of Carry Select adder, Carry Skip adder, Carry Save adder.

| Fast Adders | Technology used | Type | Cells | Total Power (uW) | Total Delay(ps) |
|--------------------|-----------------|------|-------|------------------|-----------------|
| Carry Select Adder | 45nm | Fast | 4 | 1654.587 | 279 |
| | | Slow | 4 | 1056.492 | 892 |
| | 90nm | Fast | 4 | 4627.775 | 291 |
| | | Slow | 4 | 2976.080 | 1057 |
| | 180nm | Fast | 4 | 35393.882 | 597 |
| | | Slow | 4 | 21659.847 | 1484 |
| Carry Skip Adder | 45nm | Fast | 6 | 3398.523 | 150 |
| | | Slow | 6 | 2240.899 | 1480 |
| | 90nm | Fast | 6 | 5145.281 | 106 |
| | | Slow | 6 | 3318.709 | 444 |
| | 180nm | Fast | 6 | 36795.242 | 362 |
| | | Slow | 6 | 19339.202 | 829 |
| Carry Save Adder | 45nm | Fast | 8 | 4197.663 | 298 |
| | | Slow | 8 | 2677.958 | 979 |
| | 90nm | Fast | 9 | 8958.592 | 306 |
| | | Slow | 9 | 5832.694 | 1170 |
| | 180nm | Fast | 9 | 57347.523 | 682 |
| | | Slow | 9 | 34730.669 | 1612 |

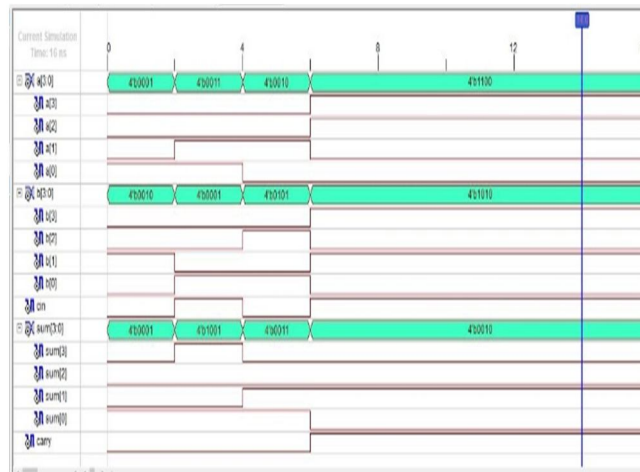


Fig 7. Output waveform of Carry Save Adder

V. CONCLUSION

Carry Select adder, Carry Skip Adder and Carry Save Adder each one of 4-bit has been implemented using Verilog in CADENCE. For simulation we have used Xilinx, NCSim and for synthesis we have used RTL compiler v14.10 have been used. As a result, it has been concluded that 4-bit Carry Select Adder gives the optimized result as compared to Carry Skip Adder and Carry Save Adder in all the technologies 45nm, 90nm, 180nm. When coming to the No. of cells Carry Select Adder uses less number of Cells when compared to Carry Skip and Carry Save Adder. While coming to total power consumption Carry Select Adder consumes less power compared to Carry Skip and Carry Save Adder. Delay is reduced in an efficient manner in case of 4-bit Adders.

VI. ACKNOWLEDGMENT

We are thankful to the Department of Electronics and Communication Engineering, Technocrats Institute of Technology ,Rajiv Gndhi Proudhyogiki Vishwavidyalaya Bhopal (M.P) .for the support rendered in carrying out this work.

REFERENCES

- [1] Prashant Gurjar, Rashmi Solanki, Pooja Kansliwal, Mahendra Vucha, "VLSI Implementation of Adders for High Speed ALU" International Journal of Computer Applications, September 2011.
- [2] Reena Rani, Laxmi Kanth Singh, Neelam Sharma, "FPGA Implementation of Fast Adders using Quaternary Signed Digit Number System" 2009 International Conference on Emerging Trends in Electronic and Photonic Devices & Systems (ELECTRO-2009)
- [3] Nuno Roma, Tiago Dias, Leonel Sousa, "Fast Adder Architectures: Modeling and Experimental Evaluation" Portuguese Foundation for Science and for Technology under the research project Configurable and Optimized Processing Structures for Motion Estimation (COSME) POSI/CHS/40877/2001.
- [4] Jasbir Kaur, Lalith Sood, "Comparison Between Various Types of Adder Topologies" International Journal of Computer SCienCe and technology IJCST Vol. 6, ISSue 1, Jan - MarCh 2015
- [5] Akash Kumar, Deepika Sharma, "Performance Analysis of Different Types of Adder for High Speed 32 Bit Multiply And Accumulate Unit" International Journal of Engineering Research and Applications (IJERA) Vol. 3, Issue 4, Jul-Aug 2013, pp.1460-1462
- [6] R. P. P Singh, Praveen Kumar, Balwinder Singh, "Performance Analysis Of Fast Adders Using VHDL" 2009 International Conference on Advances in Recent Technologies in Communication and Computing
- [7] Jucemar Monteiro, José Luís Güntzel, Luciano Agostini, "A1CSA: An Energy-Efficient Fast Adder Architecture for Cell-Based VLSI Design" supported by the Brazilian Council for Scien-tific and Technological Development (CNPq).
- [8] Pavan Kumar.M.O.V, Kiran.M, "Design Of Optimal Fast Adder" International Conference on Advanced Computing and Communication Systems (ICACCS-2013)



10.22214/IJRASET



45.98



IMPACT FACTOR:
7.129



IMPACT FACTOR:
7.429



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089  (24*7 Support on Whatsapp)