Rover Publications
United International Journal of Engineering and Sciences (UIJES)
An International Peer-Reviewed (Refereed) Engineering and Science Journal
Impact Factor:7.984(SJIF) Volume-4, Special Issue-3; ISSN: :2582-5887

Comparative Review of Adders in VLSI

S. Ramakoteswarao, S. Lakshmi Deepika, B Rajya Lakshmi
Department of Electronics and Communication Engineering
DVR and Dr. Hima Sekhar MIC College of Technology
Vijayawada, India
koram419@gmail.com, soldierdeepika@gmail.com, rajee0833@gmail.com
DOI:10.53414/UIJES:2024.43.188

***Abstract* – In the field of VLSI, Binary Addition is the fundamental operation and the way it is performed plays a key role in reducing the delay and increasing the performance of overall architecture.**

This paper briefs about various adders in Verilog HDL with their power and time constraints. The various adders included in this paper are Half Adder, Full Adder, Ripple Carry Adder, Carry Look Ahead Adder, Carry Save Adder, Carry Skip Adder, and Carry Select Adder.

***Keywords* – Half Adder, Full Adder, Ripple Carry Adder, Carry Look Ahead Adder, Carry Save Adder, Carry Skip Adder and Carry Select Adder.**

I. INTRODUCTION

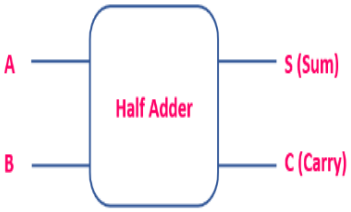
The Review of this paper is about different adders used in digital circuit, a binary adder is a digital circuit that performs binary addition, the basic arithmetic operation for binary numbers. Binary numbers consist of only two digits, 0 and 1, making the addition process simpler than in decimal arithmetic. These binary additions is used in many applications of digital circuits like Computers and calculations, Digital Communication, Digital signals processors, Memory addressing, Error Detection and Correction, Digital Encryption and Robotics.

This Binary Addition is the basic principal which is incorporated in the many digital circuits like Half Adder, Full Adder and many more circuits like Ripple Carry Adder, Carry Look Ahead Adder, Carry Save Adder, Carry Skip Adder, and Carry Select Adder, but each adder has its own process to reduce the delay and increase the speed of operation.

II. BASIC ADDER DESIGN

The Basic Adder Design consists of Half Adder, Full adder i.e.., Half Adder is a digital circuit which is used to add 2 binary bits and produces 2 outputs named Sum and Carry whereas Full Adder will also have same outputs but the main change is it can add 3 bits including carry in.

A.**Half Adder:** The Half Adder takes 2 inputs named A, B and outputs as Sum and Carry



A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$S = \overline{A}B + A\overline{B} = A \oplus B$

$C = AB$

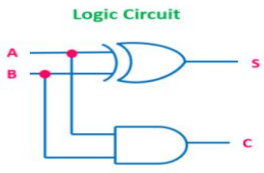
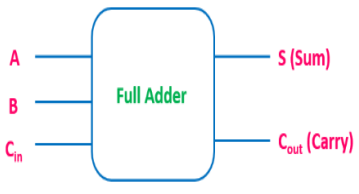


Fig. 1: Half Adder

B.**Full Adder:** The Full Adder takes 3 inputs named A, B C_{in} and and outputs as Sum and C_{out}.



A	B	C_{in}	Sum	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

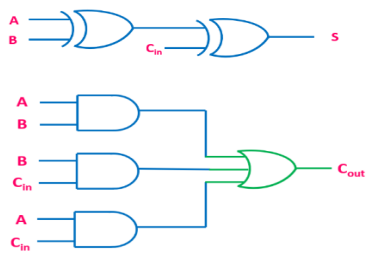


Fig. 2: Full Adder

Rover Publications
United International Journal of Engineering and Sciences (UIJES)
An International Peer-Reviewed (Refereed) Engineering and Science Journal
Impact Factor:7.984(SJIF) Volume-4, Special Issue-3; ISSN: :2582-5887

Sum = A XOR B XOR C_{in}
C_{out}= AB +BC_{in} +C_{in}A

III. ADVANCED ADDERS

The Advanced Adders are Ripple Carry Adder, Carry Look Ahead Adder, Carry Save Adder, Carry Skip Adder, Kogee - Stone Adder and Carry Select Adder and this section briefly explains about them.

1. **Ripple Carry Adder:** The Full Adder takes 3 inputs named A, B C_{in} and outputs as Sum and $C_{out.}$ If those Full Adders are connected in cascade form gives the result to Ripple Carry Adder and the carry generated in each full adder C_{out} is given as $C_{in}$to the next FullAdder.

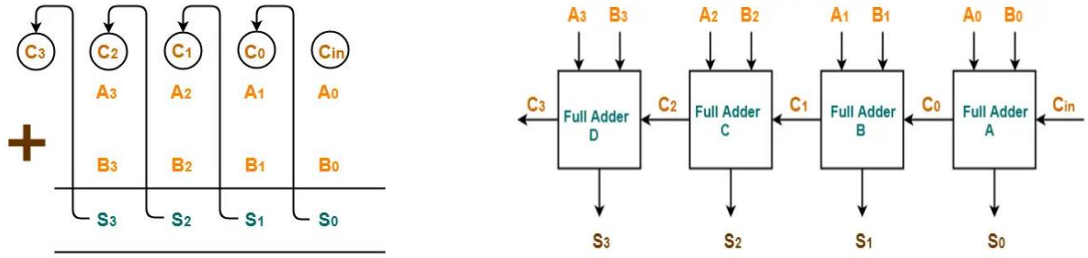


Fig. 3: Ripple Carry Adder

The above diagram shows 4 bit Ripple Carry Adder consisting of 4 full adders connected in cascade form. A0, B0 are the inputs of full adder A and S0, $C_{in}$is the initial carry and A1, B1 are the inputs of full adder B and S1 is the sum output and $C_{in}$the initial carry is fed to next Full adder as carry. Although the structure is simple and low hardware complexity, we need to wait for the propagation of carry in each stage which causes delay in operation.

2. **Carry Look-Ahead Adder:** The Carry Look-Ahead Adder is advancement over the Ripple Carry Adder, specifically designed to mitigate the propagation delay associated with carry generation. The key principle in reducing the delay in Carry Look-Ahead Adder is to pre-compute the carry generate (G) and carry propagate (P) signals for each pair of bits in parallel.

The carry output for each stage is then a function of these pre-computed signals. This approach eliminates the need for carry bits to propagate through the entire adder sequentially.

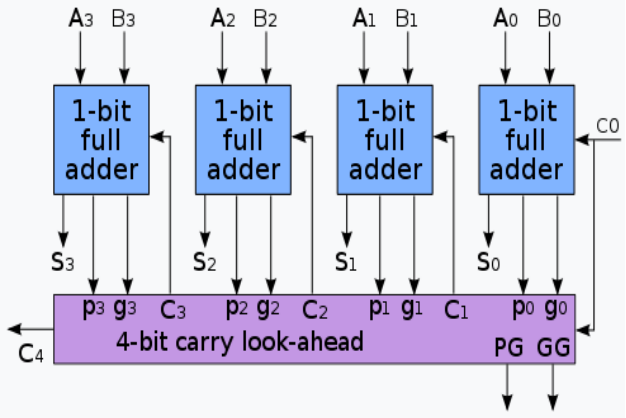


Fig. 4: Carry Look-Ahead Adder

Let's see the basic steps for computing the Carry Look-Ahead Adder:

- **Carry Generate (G) and Propagate (P) Signals :**

For each pair of input bits, calculate the carry generate (G) and carry propagate (P) signals. G indicates whether a carry is generated when both input bits are 1. P indicates whether a carry is propagated from the lower-order bit.

Gi = Ai. Bi
Pi = Ai ⊕ Bi

Rover Publications
United International Journal of Engineering and Sciences (UIJES)
An International Peer-Reviewed (Refereed) Engineering and Science Journal
Impact Factor:7.984(SJIF) Volume-4, Special Issue-3; ISSN: :2582-5887

A	B	C	C +1	Condition
0	0	0	0	
0	0	1	0	No Carry Generate
0	1	0	0	
0	1	1	1	
1	0	0	0	No Carry Propogate
1	0	1	1	
1	1	0	1	Carry Generate
1	1	1	1	

Fig. 5: Carry Look-Ahead Adder Carry Generation Table

- **Generate Sum and Carry :** Use the G and P signals to generate the sum and carry for each stage. The sum is computed using XOR gates. The carry-out is determined based on the G and P signals.

$$Si = Pi \oplus Gi., Ci+1 = Ci.Pi + Gi.$$

- **Combine Carry Bits:** The final carry-out is a combination of the pre-computed carry bits, allowing for a faster determination of the overall carry. Therefore, the carry bits C1, C2, C3, and C4 can be calculated as

C1 = C0.P0+G0.
C2 = C1.P1+G1 = (C0.P0+G0).P1+G1.
C3 = C2.P2+G2 = (C1.P1+G1).P2+G2.
C4 = C3.P3+G3 = C0.P0.P1.P2.P3 + P3.P2.P1.G0 + P3.P2.G1 + G2.P3 + G3.

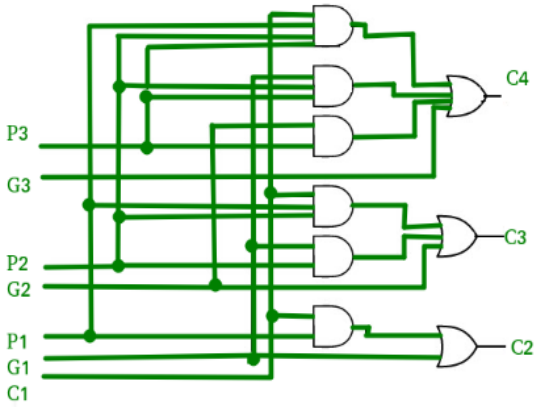


Fig. 6: Carry Look-Ahead Adder Logic Diagram

Carry Look-Ahead Adder reduces the propagation delay but the computations for generating carry are critical.

3. **Carry Select Adder:** A Carry-Select Adder is made using a two-stage Ripple Carry Adder along with a special switch called a multiplexer. When we use this Adder, it picks the sum and carry results from the first stage of the Ripple Carry Adder if the carry input is '0'. If the carry input is '1', it chooses the sum and carry results from the second stage of the Ripple Carry Adder.

To make this choice between the stages, we use a multiplexer that has N+ 1 input for N-bit addition. This multiplexer helps decide which set of results to use based on whether the carry input is '0' or '1'. In 4 bit Carry Select Adder it requires 8 Full Adders and 5 Multiplexers.

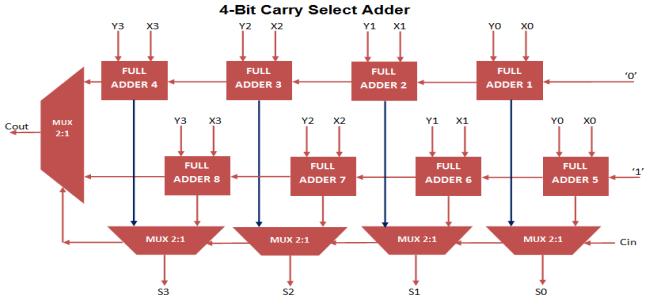


Fig. 6: Carry Select Adder

Rover Publications
United International Journal of Engineering and Sciences (UIJES)
An International Peer-Reviewed (Refereed) Engineering and Science Journal
Impact Factor:7.984(SJIF) Volume-4, Special Issue-3; ISSN: :2582-5887

The Carry-Select Adder comes with a drawback, as it demands twice the number of Full Adders needed for a regular addition operation. Additionally, it necessitates the inclusion of extra multiplexers to facilitate the selection process among the adders.

4. **Carry Skip Adder:** Carry-Select Adder is also known as Carry Bypass Adder, unlike the Carry-Select Adder, a Carry Skip Adder doesn't rely on a large number of full adders. Instead, it employs a different approach using AND gates, XOR gates, and multiplexers to implement Carry Skip Logic.

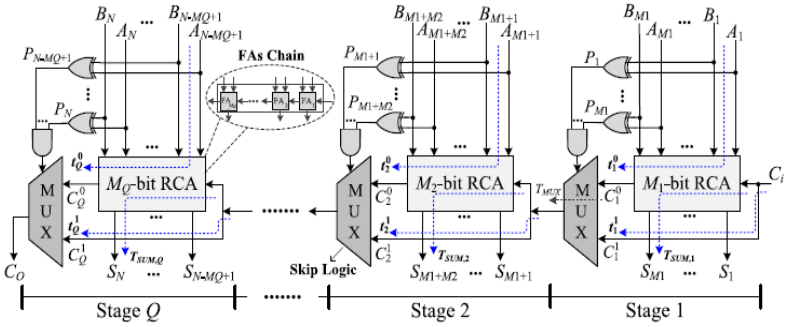


Fig. 7: Carry Skip Adder

IV. Results and Discussions

1. **Ripple Carry Adder:** The below window shows the result of 4 – bit Ripple Carry Adder, and the code was written in Verilog HDL using structural model. The below window shows the result adding 2 4 bit numbers i.e.., X = 5 and Y = 6 and the result is stored in S and C.

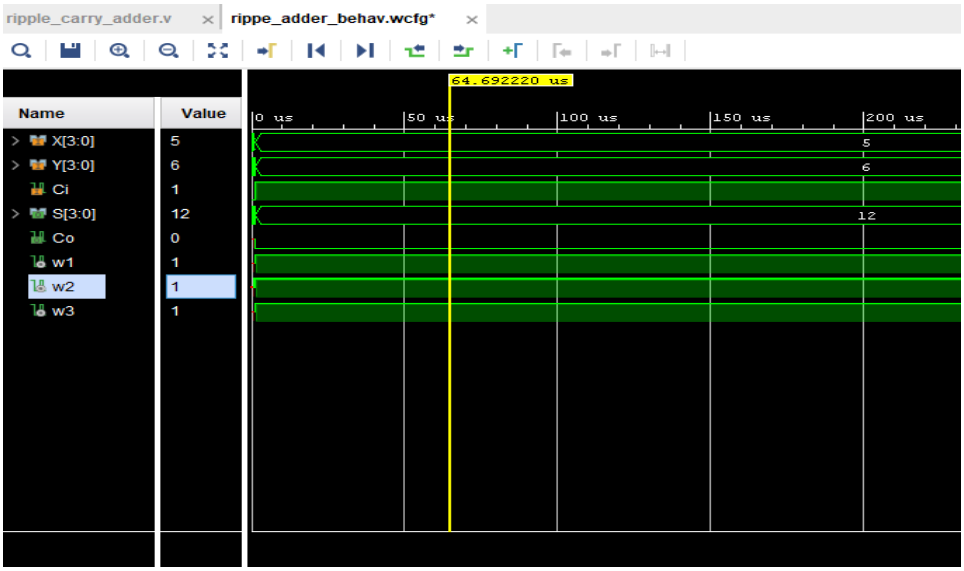


Fig. 8 Ripple Carry Adder

2. **Carry Look-Ahead Adder:** The below window shows the result of 4 – bit Carry Look-Ahead Adder, and the code was written in Verilog HDL using structural model. The below window shows the result adding 2 4 bit numbers i.e.., X = 5 and Y = 6 and the result is stored in Sum and C.

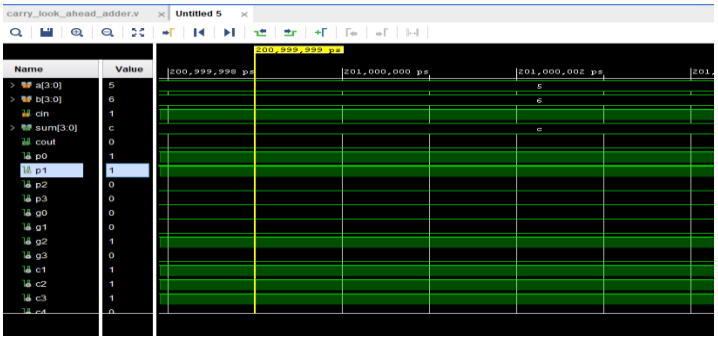


Fig. 9: Carry Look-Ahead Adder

3. **Carry Select Adder:** The below window shows the result of 4 – bit Carry Select Adder, and the code was written in Verilog HDL using structural model. The below window shows the result adding 2 4 bit numbers i.e.., A = 5 and B = 4 and the result is stored in S and C_{out}.

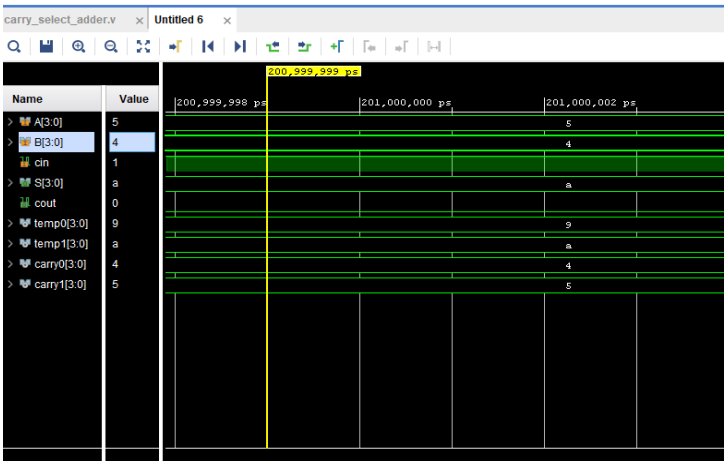


Fig. 10: Carry Select Adder

4. **Carry Skip Adder:** The below window shows the result of 4 – bit Carry Skip Adder, and the code was written in Verilog HDL using structural model. The below window shows the result adding 2 4 bit numbers i.e.., A = 5 and B = 7 and the result is stored in Sum and C_{out}.

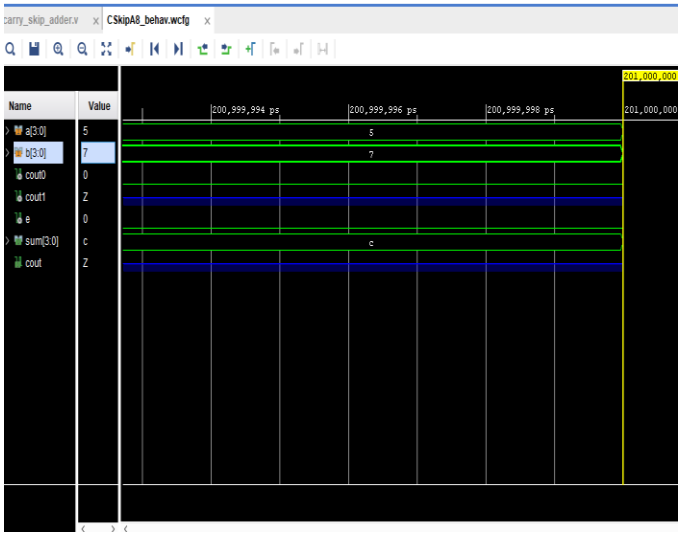


Fig. 11: Carry Skip Adder

VI. CONCLUSION

In this paper the 4 types of adders named Ripple Carry Adder, Carry Look Ahead Adder, Carry Select Adder and Carry Skip Adder was compared for 4 bit and shown in the table.

While comparing all the adders the power consumed by Carry Skip Adder was less and Setup and Hold time was also less i.e.., power is 2.353W, Setup time is 4.8154 and Hold time is 1.6734.

We can extend the design by using some other techniques in verilog HDL code so that we can reduce some more power and utilities and in advanced to Carry Skip Adder there are some other adders available.

A conclusion section is not required. Although a conclusion may review the main points of the paper, do not replicate the abstract as the conclusion. A conclusion might elaborate on the importance of the work or suggest applications and extensions.

Table 1: ????????

Name	Slice LUTs(41000)	Slice (10250)	LUT as Logic(41000)	Bonded IOB(3000)
Ripple Carry Adder	4	1	4	13

Carry Look-Ahead Adder	6	2	6	14
Carry Select Adder	4	2	4	14
Carry Skip Adder	4	1	4	13

Table 2: ????????

Name	Power	Set up Time	Hold Time
Ripple Carry Adder	2.719W	6.1058	1.39
Carry Look-Ahead Adder	3 W	5.6796	2.1374
Carry Select Adder	2.96W	6.174	2.138
Carry Skip Adder	2.353W	4.8154	1.6734

REFERENCES

[1] V. G. Oklobdzija, B. R. Zeydel, and H. Q. Dao, "Comparison of high-performance VLSI adders in the energy-delay space," in IEEE. transactions on Very Large Scale Integration (VLSI) Systems (vol. 13,no. 6, June 2005).
[2] SaradinduPandu, A. Benerjee, B. Maji, and Dr. A. K. Mukhopadhyay, "Power and delay comparison in between different types of full adder circuits," in International Journal of Advanced Rsearch in Electrical,Electroncis and Instrumentation Engineering, vol. 1, no. 3, Sep. 2012.
[3] JasbirKaur and LalitSood, "Comparison between various types of adder topologies," in IJCST, vol. 6, no. 1, Jan.–Mar. 2015.
[4] SwaroopGhosh, Patrick Ndai, and Kaushik Roy, "A novel low overhead fault tolerant Kogge-Stone adder using adaptive clocking," DATE 2008.
[5] MarojuSaiKumar and Dr. P. Samundiswary, "Design and performance analysis of various adders using verilog," in International Journal of Computer Science and Mobile Computing, IJCSMC, vol. 2, no. 9,pp. 128–138, Sep. 2013.
[6] BhavaniKoyada, N. Meghana, Md. Omar Jaleel and Praneet RajJeripotula, " A Comparative Study on Adders", in IEEE WiSPNET 2017 conference.