Comparison and Simulation of Digital Adders

Krisshiv Hinduja,[1] Riya Kadulkar,[1] Prof. Uttara Bhatt[2]

[1] Student, Electronics and Telecommunication, Thadomal Shahani Engineering College, Maharashtra, India
[2] Professor, Electronics and Telecommunication, Thadomal Shahani Engineering College, Maharashtra, India

---***---

***Abstract** - In electronics, an adder is a digital logic circuit that is commonly used to add numbers. Adders are used in many computers and other types of processors to calculate addresses and associated actions, as well as table indices in the ALU and other portions of the processor. It is critical to select the appropriate adder for the task at hand. Because some adders provide higher speeds, lower power consumption, and superior performance, they must be chosen carefully and not at random.[1]. In this paper we are comparing 4 different adders on the basis of number of slices, speed, delay and power dissipation. This comparison will help to decide which adder to be used. The adders which we are going to compare are Ripple carry, Carry look ahead, Carry select, Kogge stone.*

Key Words**: **Speed, Power Dissipation, Ripple Carry, Carry Look Ahead, Carry Select and Kogge Stone.

1. INTRODUCTION

Adders are an important block in a large number of digital systems such as microprocessors, microcontrollers and various data processing units. In certain cases, adders may also be used in parts of a processor. Here addition can be explained as a process in which two input quantities are taken and added to generate two outputs, a sum and a carry.

The most important part in any digital circuit design is to try and increase the speed and decrease the power consumption and area. The usage of adders can reduce the area by a certain amount by decreasing the number of transistors used. However, a single type of adder cannot fulfil all of the criteria mentioned there are different kinds of adders that can be used depending upon the requirement and the project.

The adders discussed in the paper are:

1. Carry Look Ahead Adder
2. Carry Select Adder
3. Ripple Carry Adder
4. Kogge Stone Adder

Even though every adder has a similar function, the way the task being processed and the transistor count varies. Every logic style has its own advantages and disadvantages that influence the criteria mentioned above. To make the selection of an adder a simple task the comparison of the above-mentioned adders has been done and put forward.

The paper is organized section wise. Section II describes the working and the circuit of the adders mentioned. Section III explains the Simulation results. Section IV focuses on the comparison of the adders mentioned and the conclusion of the comparison has been put into Section V.

2. ADDERS

- ### 2.1 Carry Look Ahead

A carry-lookahead adder is also called a fast adder. It's an electronic adder that's employed in digital logic. The carry Look Ahead adder is explained with Ai, Bi as inputs and Ci as the carry input. S is the final sum and Cout is the output carry. A carry look-ahead adder minimizes propagation latency by introducing more advanced circuitry. It enhances performance by lowering the time it takes to determine carry bits.

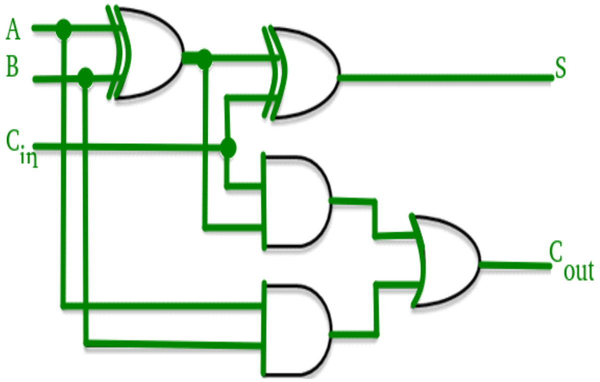


Fig-1: Carry Look Ahead Adder

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

Table-1: Conditions for Carry Look Ahead

Consider the whole adder circuit with the accompanying truth table given above. The variables 'carry generate' Pi and 'carry propagate' Gi are then defined.

Pi = Ai ⍰ Bi

Gi = AiBi

Carry generate and carry propagate can be used to express the total output and carry output.

Si = PiCi

C_{i+1} = Gi + PiCi

where Gi produces the carry when both Ai, Bi are 1 regardless of the input carry. Piis associated with the propagation of carry from Cito Ci+1.

In a four-stage carry look-ahead adder, the carry output Boolean function of each stage can be represented as

C1 = G0 + P0Cin

C2 = G1 + P1C1 = G1 + P1G0 + P1P0Cin

C3 = G2+P2C2 =G2+P2G1+P2P1G0+P2P1P0Cin

C4=G3+P3C3=G3+P3G2+P3P2G2+P3P2P1G0 P3P2P1P0Cin

From the above Boolean equations, we can observe that c4 does not have to wait for c3 and c2 to propagate but actually c4 is propagated at the same time as c3 and c2. Since the Boolean expression for each carry output is the sum of products so these can be implemented with one level of AND gates followed by an OR gate.

The implementation of three Boolean functions for each carry output (c4, c3 and c2) for a carry look-ahead carry generator shown in below figure.

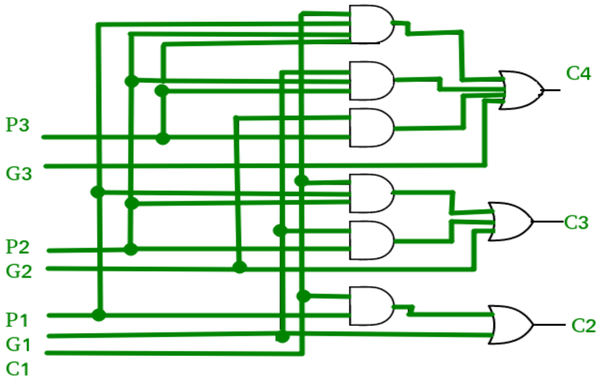


Fig-2: Boolean Representation for Carry Look Ahead

Time Complexity Analysis:

A carry look-ahead adder can be thought of as having two parts

- This section calculates the carry for each bit.
- For each bit position, this is the component that adds the input bits and the carry.

The log(n) complexity arises from the part that generates the carry, not the circuit that adds the bits. Now, for the generation of the nth carry bit, we need to perform a AND between (n+1) inputs. The complexity of the adder comes down to how we perform this AND operation. If we have AND gates, each with a fan-in (number of inputs accepted) of k, then we can find the AND of all the bits in $\log_k(n+1)$ time. This is represented in asymptotic notation as $\ominus$(logn).

Advantages and Disadvantages of Carry Look-Ahead Adder:

Advantages –

- The propagation delay is reduced.
- It gives the quickest addition logic.

Disadvantages –

- As the number of variables grows, the Carry Look-ahead adder circuit becomes more difficult.
- The circuit is more expensive since it uses more hardware.

2.2 Carry Select Adder

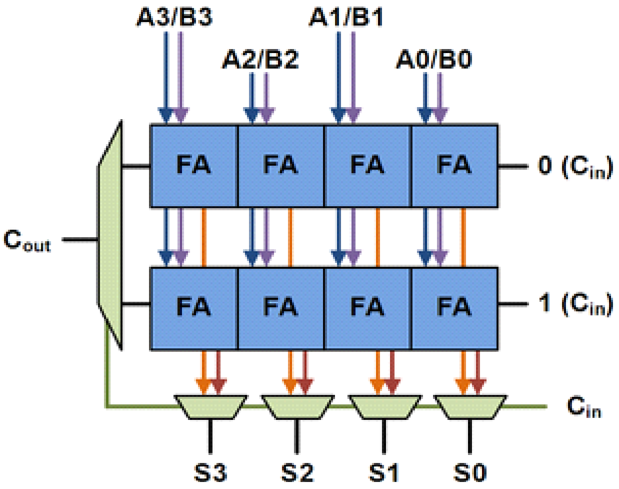


Fig-3: Carry Select Adder

The basic building block of a carry-select adder is shown above, with a block size of 4. Two 4-bit ripple-carry adders are multiplexed together, with the carry-in selecting the resultant carry and sum bits. Because one ripple-carry adder assumes a carry-in of 0 while the other assumes a carry-in of 1, using the actual carry-in to determine which adder had the proper assumption produces the desired result.

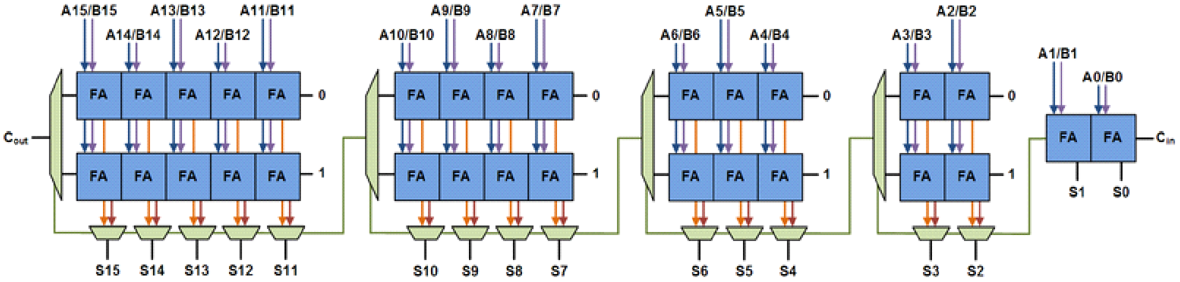


Fig-4: 16-bit Carry Select Adder

Similarly, a variable-size 16-bit carry-select adder can be created. An adder with block sizes of 2-2-3-4-5 is shown below. When the full-adder delay equals the MUX delay, which is unusual, this break-up is excellent. Two complete adder delays and four mux delays make up the total delay. We strive to keep the delay between the two carry chains and the preceding stage's carry chain equal.

Advantages –

- It is faster.

Disadvantages –

- It requires a larger area.

2.3 Ripple Carry Adder

A ripple carry adder is a logic circuit that is made up of multiple full adders arranged in a cascade form. In this each carry bit gets rippled into the next stage. The carry out bit of the full adder becomes the input carry bit for the next full adder. There are three types of ripple carry adders - 4-bit, 8-bit and 16-bit. The circuit for the 4-bit ripple carry adder is shown in Figure below.

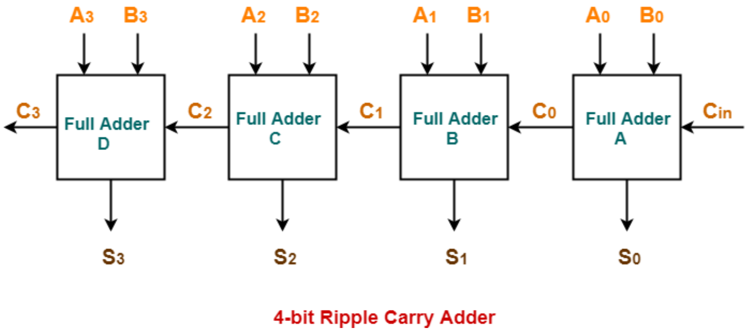


Fig-5: 4-bit Ripple Carry Adder

The ripple carry adder has a high propagation delay, the time that occurs between the input and output. This propagation delay affects the speed of the circuit and also the power consumption. The circuit has low area consumption but high delay time, slow in speed and also high power consumption. The equations for the ripple carry adder are shown below.

Sum = A1⨁B1⨁Cin

Carry = A1B1⨁B1Cin⨁CinA1

The inputs used for the full adder are A and B and after addition we get the Sum and Carry. When the inputs Ao and Bo are given to the first full adder along with the starting carry Cin as 0 the sum and carry are obtained using the above-mentioned equations. The carry will be rippled to the next full adder and the input bits A1 and B1 are given along with the rippled carry. This process goes on until the final full adder which provides us with the output sum and carry bits.

The ripple carry adder is used when the input bits are large and the operation cannot be performed by the full adders and half adders. The advantage obtained here is that addition for n-bit sequences can be performed but at the cost of the speed of the circuit as the delay time is high.

2.4 Kogge Stone Adder

The Kogge Stone adder is an example of a parallel prefix adder. It is flexible as compared to the other adders and also performs the operations with high speed. It is considered as a high-performance adder due to minimum logic depth which in turns increases the speed. The circuit diagram for the 4-bit kogge stone adder is shown in Figure.

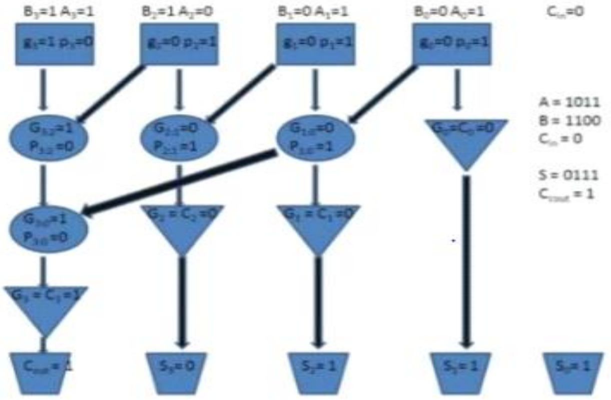


Fig-6: 4-bit Kogge Stone Adder

This adder is the most highly used adder in industries that require high speed performance and as 32-bit, 64-bit and 128-bit adders. The kogge stone adder reduces the delay time by a large amount to obtain the carry signals. The delay is represented by the equation log2n which is the number of stages present in the operator. The equations for the kogge stone adder are shown below.

Pi:j = Pi:k+1 AND Pk:j

Gi:j = Gi:k+1 OR (Pi:k+1 AND Gk:j)

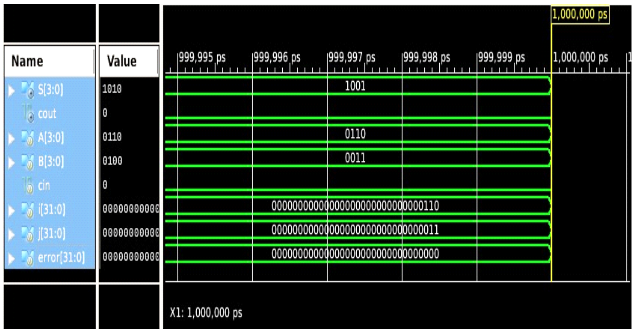
The circuit is made up of three stages. The first stage makes use of full adders provided with the input carry to obtain the propagate (P) and generate (G). The second stage uses the above equations to generate the bits P1 and G1. The third stage similarly generates the term P2 and G2. The sum bits are generated depending on the output of the first stage and the generated terms.

3. SIMULATION RESULT

In this section we'll be discussing simulation of the adders. The Xilinx synthesis tool is used for synthesis and Xilinx simulation tool is used for simulation. Thesimulation results provide us with data on the operands, operation, and outcomes.[1]

The operands used here are A and B that include 4 bits along with an input carry Cin. The values of operands taken: -

For carry look ahead adder are A = "0110", B = "0100", cin = 0 which results in the sum as S = "1010" and output carry as cout = '0'.



Name	Value
S[3:0]	1010
cout	0
A[3:0]	0110
B[3:0]	0100
cin	0
i[31:0]	00000000000
j[31:0]	00000000000
error[31:0]	00000000000

Fig-7: Simulated Waveform of CLA

For carry select adder are A = “0101”, B = “0110”, cin = 1 which results in the sum as S = “1100” and output carry as cout = ‘0’.

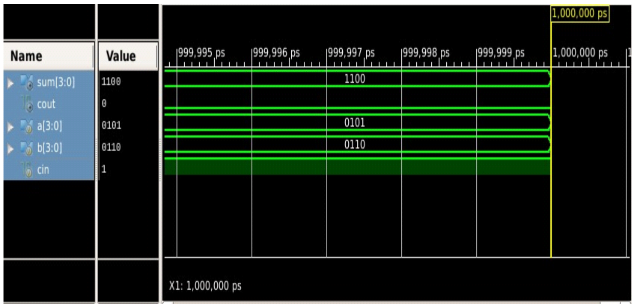


Fig-8: Simulated Waveform of CSA

For ripple carry adder are A = “1111”, B = “1111”, cin = 0 which results in the sum as S = “1110” and output carry as cout = ‘1’.

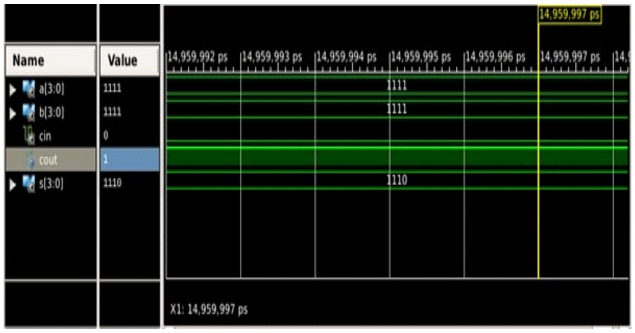


Fig-9: Simulated Waveform of RCA

For Kogge stone adder are A = “1101”, B = “1100”, cin = 0 which results in the sum as S = “1001” and output carry as cout = ‘1’.

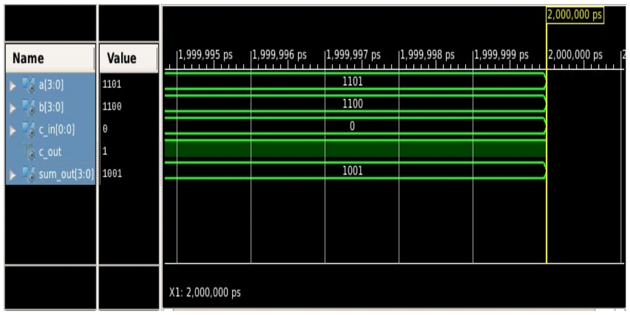


Fig-10: Simulated Waveform of KSA

4. COMPARISON ANALYSIS

From the study we can analyze the area occupied and the speed of the operation for each adder using the Xilinx software and corresponding VHDL and Verilog Codes. This gives a brief idea about the adder selection and shows that out of the four adders mentioned the Kogge Stone Adder is the most efficient.[3]

The Adders can also be compared based on the number of gates used which in turn again compensate for the area. The Kogge stone adder makes use of the greatest number of gates. Then comes the Carry Select Adder with a gate count of around 600.

The Carry Look Ahead Adder has the least number of gates that is 272 and the Ripple Carry Adder has the gate count approximately equal to 288.

SR. NO	Adder Topology	Area / No of Slices	Speed (ns)	LUT Used
1.	Ripple Carry	4	8.96	9
2.	Carry Look Ahead	4	8.92	11
3.	Carry Select	5	8.35	9
4.	Kogge Stone	4	7.82	5

Table-2: Comparison based of Area and Speed

The Adders mentioned can also be compared using a few different parameters such as the power dissipation and the delay time that takes place in a particular operation. The table below provides a comparison for the parameters mentioned and one can accordingly select an Adder that best suits the purpose or the operation to be executed.

SR. NO	Adder Topology	Delay (ns)	Power Dissipation
1.	Ripple Carry	6.445	0.206 mW
2.	Carry Look Ahead	13.463	1.082 mW
3.	Carry Select	12.425	1.109 mW
4.	Kogge Stone	9.077	3.022 mW

Table-3: Comparison based on Power Dissipation and Delay

5. CONCLUSION

For the comparative study, the above-mentioned different adders have been compared using the VHDL or Verilog Codes. The comparison based on the different parameters have been mentioned in the table shown below. The Kogge Stone Adder proves to be one of the best among all the adders that are mentioned. The Ripple Carry Adder consumes less area but takes a comparatively larger time for execution. The Carry Look Ahead Adder requires almost the same area as that of the Ripple Carry Adder but compared to the Ripple Carry Adder it executes the operation in a lesser time as compared to it. The Carry Select Adder has a lesser speed as compared to the CLA Adder but the area occupied by it is larger than the other adders. The Kogge Stone Adder has the same area as the Ripple Carry and Carry Look Ahead but the speed of operation is the fastest among all the adders. The most efficient performance among all the adders can be obtained using the Kogge Stone Adder.[3]

REFERENCES

1. Bhavani Koyada, N. Meghana, Md. Omair Jaleel and Praneet Raj Jeripotula, "A Comparative Study On Adders," in IEEE WiSPNET 2017 conference.

2. R.UMA,Vidya Vijayan , M. Mohanapriya ,Sharon Paul, "Area, Delay And Power Comparison Of Adder Topologies," in International Journal of VLSI design & Communication Systems (VLSICS) Vol.3, No.1, February 2012.

3. R. Jacob Baker, CMOS Circuit Design, Layout and Simulation, IEEE Press Series on Microelectronic Systems Stuart K. Tewksbury and Joe E. Brewer, Series Editors, Wiley Student Edition

4. M. Morris Mano, Digital Logic and Computer Design, Third Edition.