ISSN:0975-9646

Design and Comparative Analysis of Various Adders through Pipelining Techniques

Aakansha[1] , Ravi Payal[2]
[1]*M.Tech Student,*
[2]*Senior Technical Officer*

[1,2]*Department of Electronics & Communication,*
Indira Gandhi Delhi Technical University for Women

Abstract- In VLSI application, adders play a very vital role. As we all know that the adders are most frequently used components in many circuits, so the designing of efficient adders is of main concern for researchers. This paper deals with the performance analysis of different adders like Ripple Carry Adder, Carry Look Ahead Adder, Carry Select Adder, Carry Save Adder, Kogge Stone Adder and Ladner Fischer Adder. The comparison is based on three performance parameters which are AREA, SPEED and MEMORY USAGE.

Keywords- Ripple Carry Adder, Carry Look Ahead Adder, Carry Select Adder, Carry Save Adder, Kogge-Stone Adder and Ladner Fischer Adder.

1. INTRODUCTION

Addition is a fundamental operation of any digital system, control system and digital signal processing. A fast and accurate operation of a digital system is influenced by the performance of an adder which the digital system uses. Adders play a very vital role in digital system because of their extensive use in many digital operations like subtraction, multiplication and division. As we know that there has been an increased growth in wireless electronics and distributed computer architecture which has pushed the need for developing innovative designs for realizing fast multi-bit adders. To increase the speed of computation, pipelining technique is used. As the number of input bits increases the delay associated with the computation of carries also increases. So our main focus is to reduce the delays associated with carry propagation. In this paper we will discuss the same.

2. RIPPLE CARRY ADDER

In Ripple Carry Adder, the Full Adder blocks are connected in cascade in such a way that output of one full adder block is connected as input to another full adder blocks. The logic circuit of Ripple Carry Adder is design in such a way that the carry-out of each full adder is the carry-in of the next full adder. As carry bits get rippled, it is called as Ripple Carry Adder.

The disadvantage with Ripple Carry Adder is that delay is more as the number of bits is increased in Ripple Carry Adder.

3. CARRY LOOK AHEAD ADDER

Carry Look Ahead Adder solves the carry delay problem of Ripple Carry Adder by calculating the carry signals in advance based on input signal. Therefore Carry Look Ahead Adder is faster than Ripple Carry Adder. Carry Look Ahead Adder decreases the carry delay by using the lesser number of gates through which a carry signal must propagate in the generation and propagation stage. Carry Look Ahead Adder operation is based on two signals called P and G for each bit position. The P and G are expressed as

Pi =Ai xor Bi Carry propagate----- (3.1)

Gi=Ai and Bi Carry generate------- (3.2)

The Si and Ci+1 represent the sum and carry-out respectively the Si and Ci+1 are expressed as.

Si = Pi xor Ci-1---------- (3.3)

Ci+1 = Gi or (Pi and Ci) = Gi+ (Pi Ci) ---------- (3.4)

The disadvantage with Carry Look Ahead Adder is that the logic block gets complex because of use of more than 4-bits.

4. CARRY SELECT ADDER

The structure of Carry Select Adder generally consists of two Ripple Carry Adders with multiplexers. In carry-select adders, both sum and carry bits are calculated for the two assumptions: input carry (i.e. carry-in or Cin) “0” and “1”. Once the carry-in is delivered as input to multiplexer, the correct calculation is chosen (using a MUX) to produce the desired output. Instead of waiting for the carry-in to calculate the sum, the sum is delivered as correct output as soon as the carry-in gets there. The time taken to compute the sum is then avoided which results in a faster speed of calculation of binary numbers.

5. CARRY SAVE ADDER

Carry Save Adders are ideal for this type of addition. The n-bit Carry Save Adder consists of n-bit full adders which are not joined together (i.e. these full adders are independent) which is used to calculate single bit sum and carry bit based on the corresponding bits of three input numbers. In Carry Save Adder, we feed n-bits input

integers to be added and Carry Save Adder produce two n-bits output which are sum and carry. Carry Save Adder is also known as (3,2) counter where 3 represent three n- bit input and 2 represents two n-bits output.

Let us take the example to understand the concept of Carry Save Adder and how Carry Save Adder performs addition.

1 0 1 0
0 0 1 1
(+) 0 1 0 1
---
1 1 0 0 ---------- SUM
0 0 1 1 ------ SAVE CARRY
---
1 0 0 1 0 ------ FINAL SUM

6. KOGGE-STONE ADDER

Kogge-Stone Adder is a type of parallel prefix form of Carry Look Ahead Adder. Kogge–Stone Adder is the fastest adder and is widely used in the industry for high performance of arithmetic circuits. For addition Parallel Prefix Adder follows these steps which are common for all the Parallel Prefix Adder.

1. Pre- processing stage

This stage is used to calculate, generate and propagate signals to each pair of inputs A and B. These signals are given by the logic equations 6.1 & 6.2 respectively.

Pi=Ai xor Bi (6.1)

Gi=Ai and Bi (6.2)

2. Carry generation stage

This stage is used to calculate carries corresponding to each bit. The operation of execution is performed in parallel due to which Kogge-Stone Adder is also known as Parallel Prefix Adder. It uses carry propagate and carry generate as intermediate signals which are shown in the logic equations 6.3 and 6.4 respectively.

CPi:j=Pi:k+1 and Pk:j (6.3)

CGi:j=Gi:k+1or(Pi:k+1andGk:j) (6.4)

3. Post processing stage

This is the final step to calculate the summation of input bits. It is common for all adders and the sum bits are computed by logic equation 6.5 & 6.6 respectively.

Ci-1=(Pi and Cin) or Gi (6.5)

Si=Pi xor Ci-1 (6.6)

The figure of 4-Bit Kogge-Stone Adder is shown below

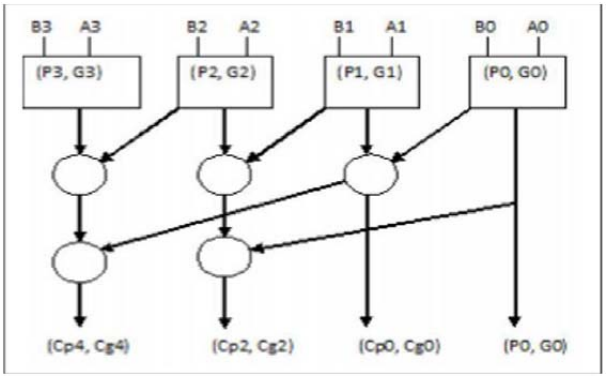

Fig 6.1: 4-Bit Kogge-Stone Adder

7. LADNER FISCHER ADDER

The Ladner-Fischer is one of the parallel prefix adders that are used to perform the addition operation. It follows the same operational steps as Kogge-Stone Adder. The figure of 4-Bit Ladner Fischer Adder is shown below.

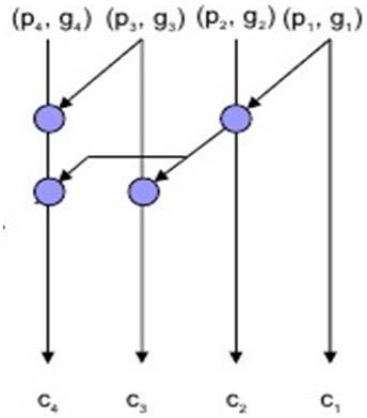

Fig 7.1: 4-Bit Ladner-Fischer Adder

8. PIPELINNING TECHNIQUES

Pipelining is a type of technique which is used to speed up the operation of a computer system. The method used for gaining significant speedup with modest hardware cost is the technique of pipelining. In pipelining technique, a task is broken down into multiple steps, and independent processing units are assigned to each step. Once the task of initial step is completed, another task may enter that step while the original task moves on to the following step. This process is similar to the one like an assembly line with a different task in progress at each stage.

9. SIMULATION WAVEFORMS

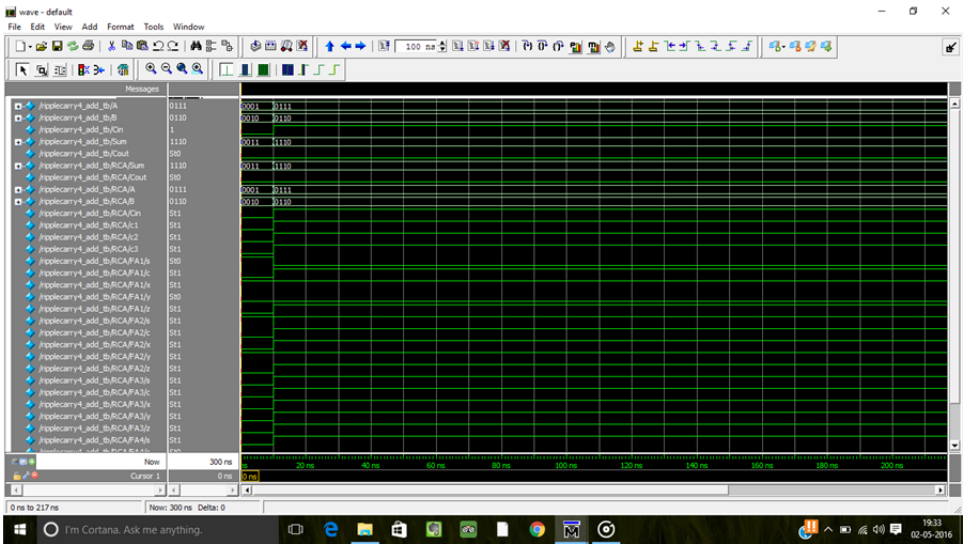

Fig 9.1: Simulation Waveform of 4-bit Ripple Carry Adder

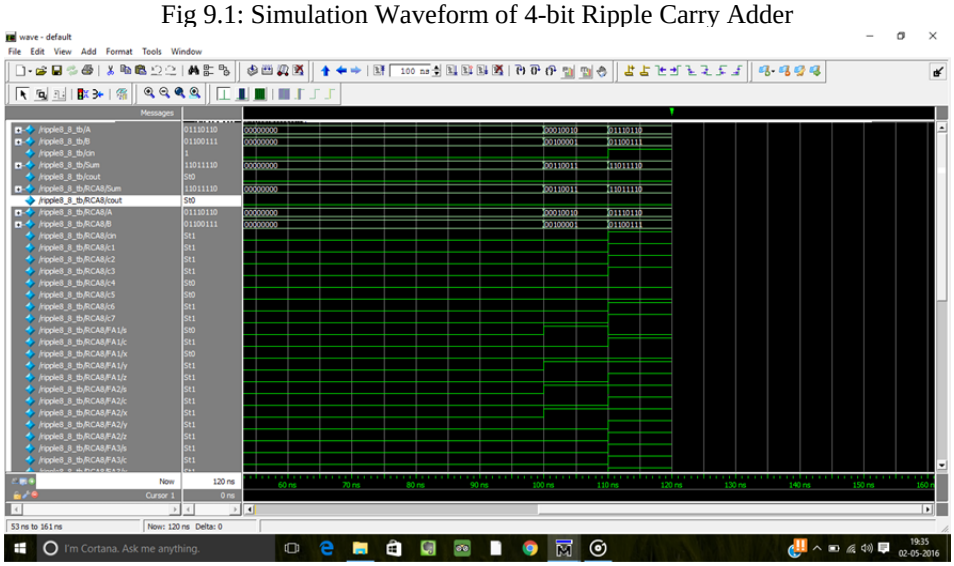

Fig 9.2: Simulation Waveform of 8-bit Ripple Carry Adder

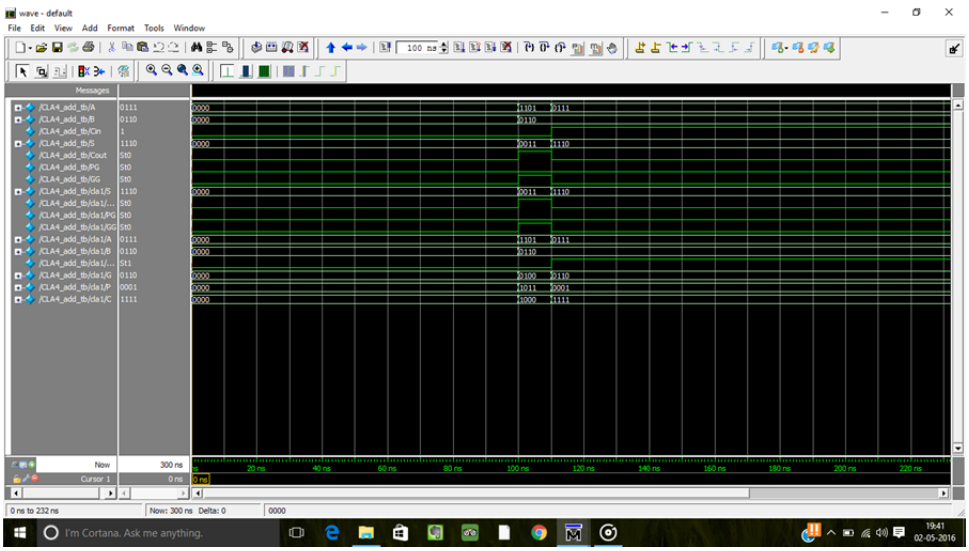

Fig 9.3: Simulation Waveform of 4-bit Carry Look Ahead Adder

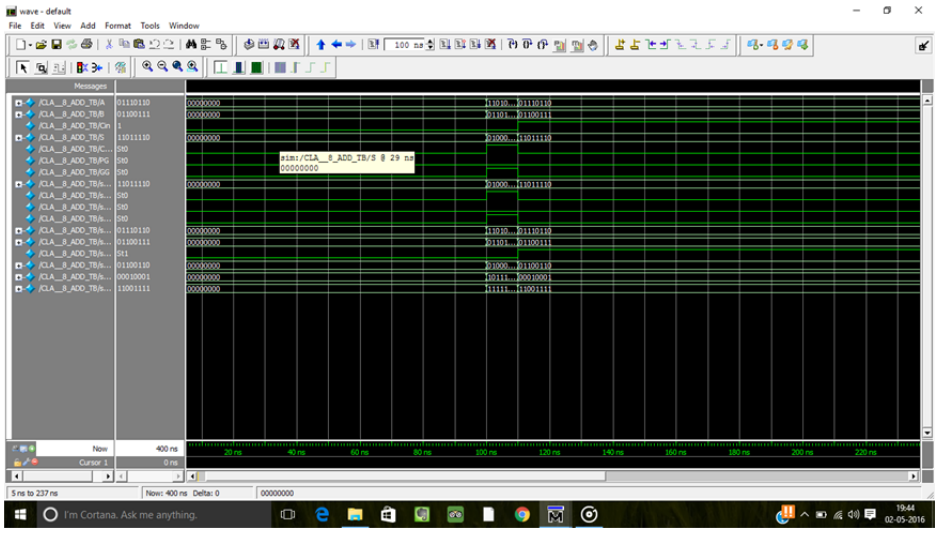

Fig 9.4: Simulation Waveform of 8-bit Carry Look Ahead Adder

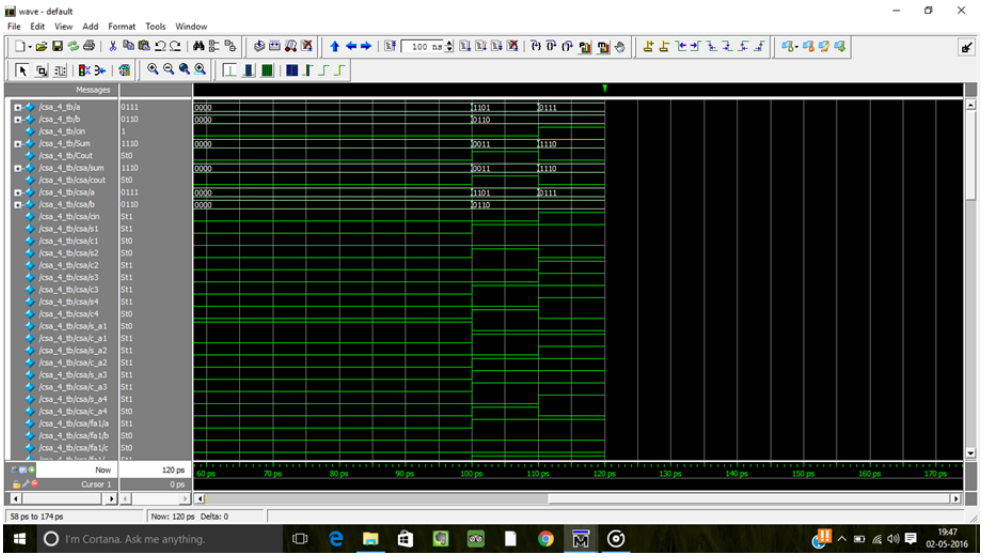

Fig 9.5: Simulation Waveform of 4-bit Carry Select Adder

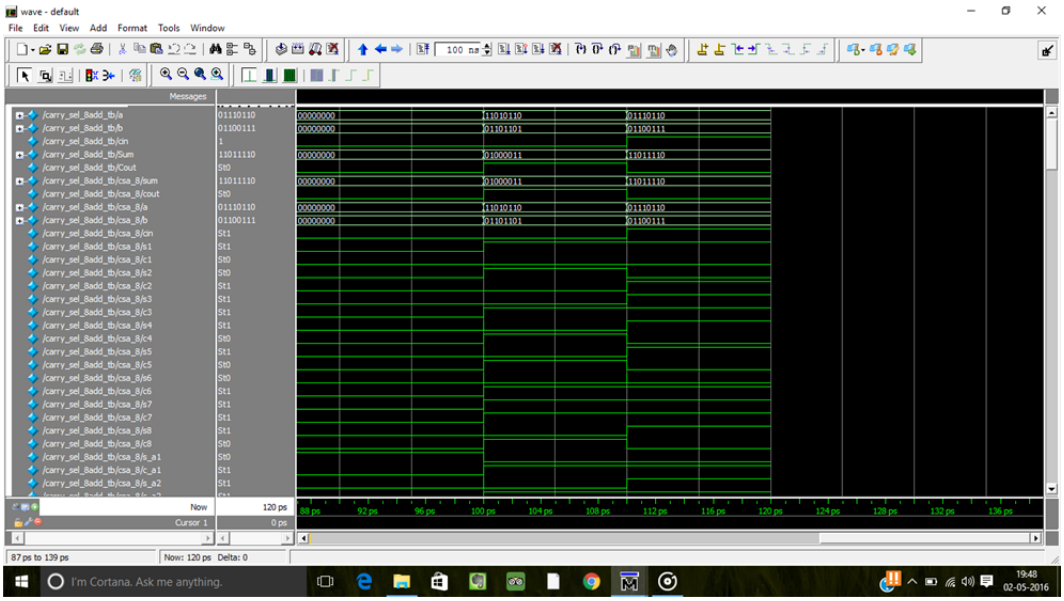

Fig 9.6: Simulation Waveform of 8-bit Carry Select Adder

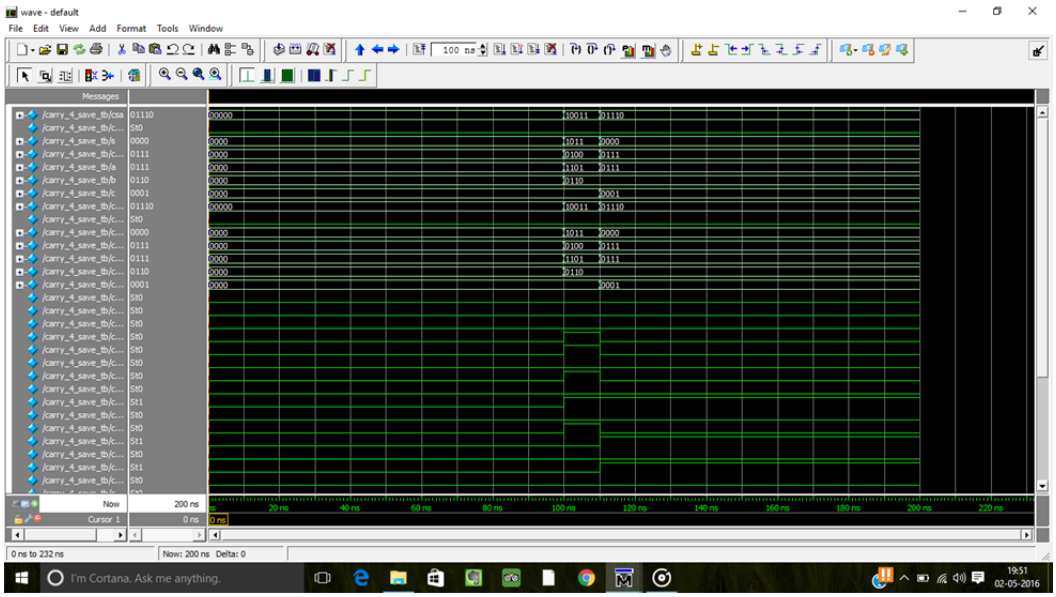

Fig 9.7: Simulation Waveform of 4-bit Carry Save Adder

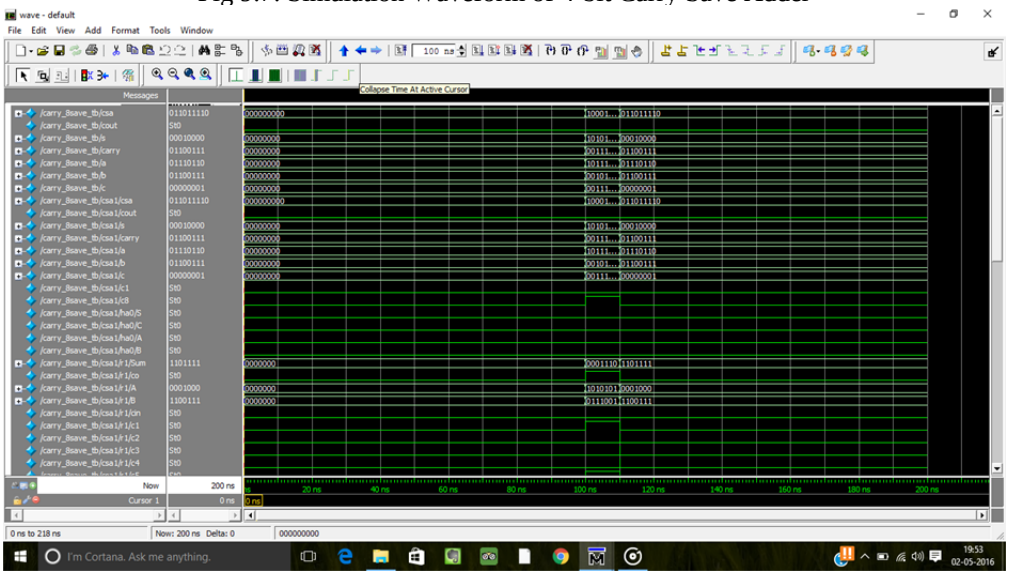

Fig 9.8: Simulation Waveform of 8-bit Carry Save Adder

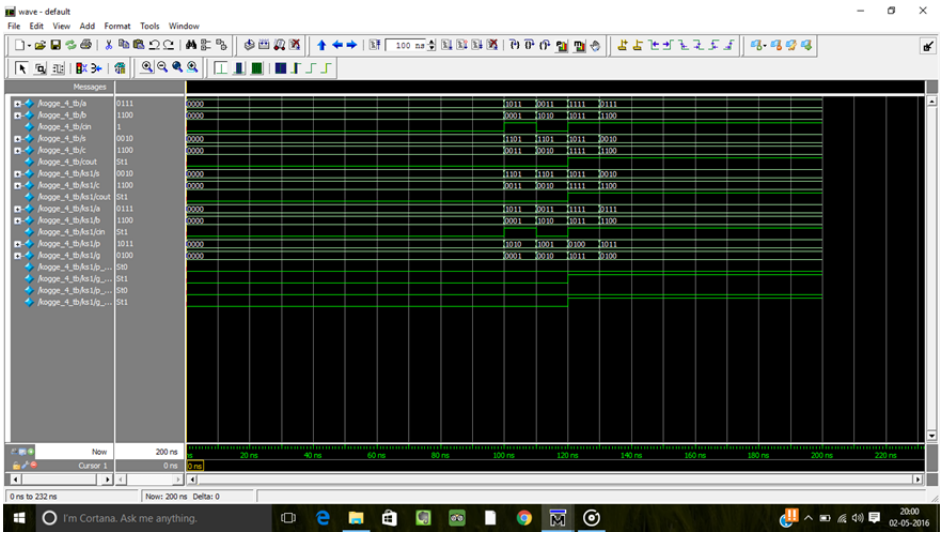

Fig 9.9: Simulation Waveform of 4-bit Kogge-Stone Adder

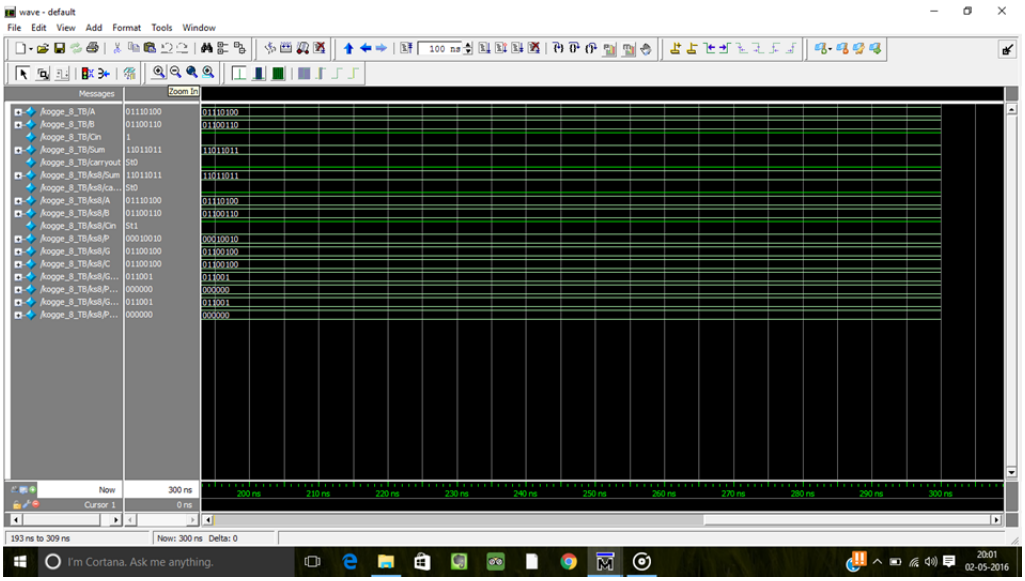

Fig 9.10: Simulation Waveform of 8-bit Kogge-Stone Adder

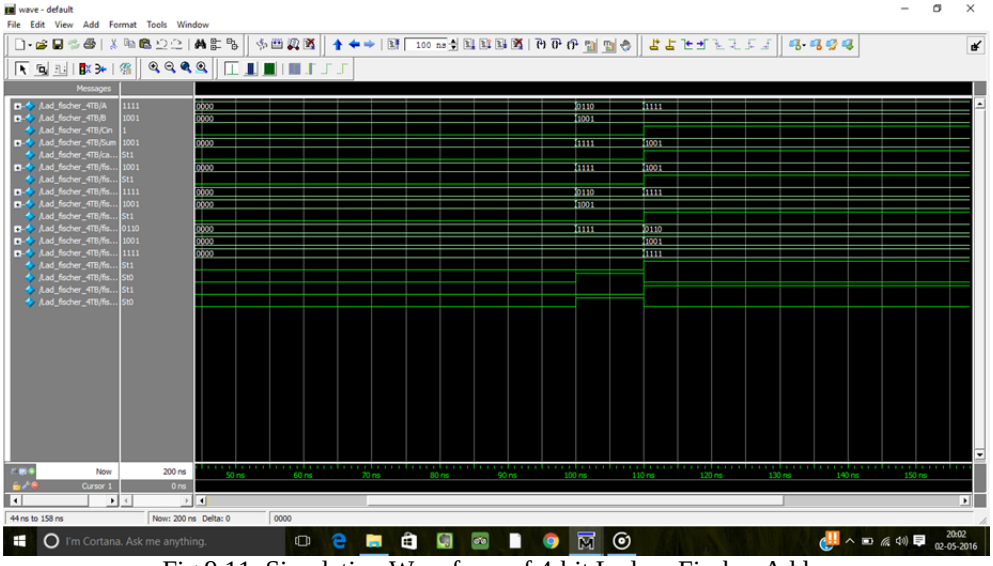

Fig 9.11: Simulation Waveform of 4-bit Ladner Fischer Adder

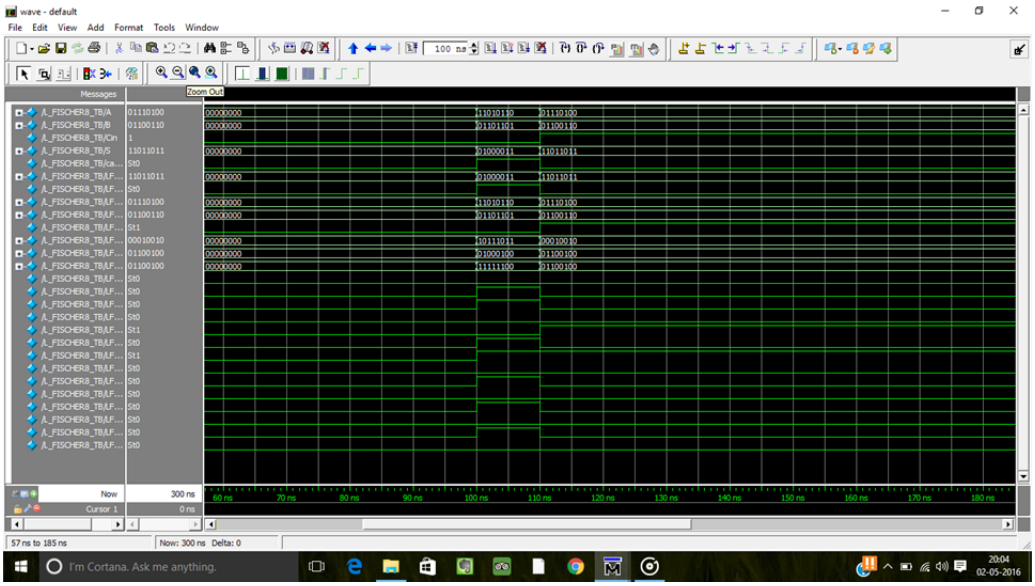

Fig 9.12: Simulation Waveform of 8-bit Ladner Fischer Adder

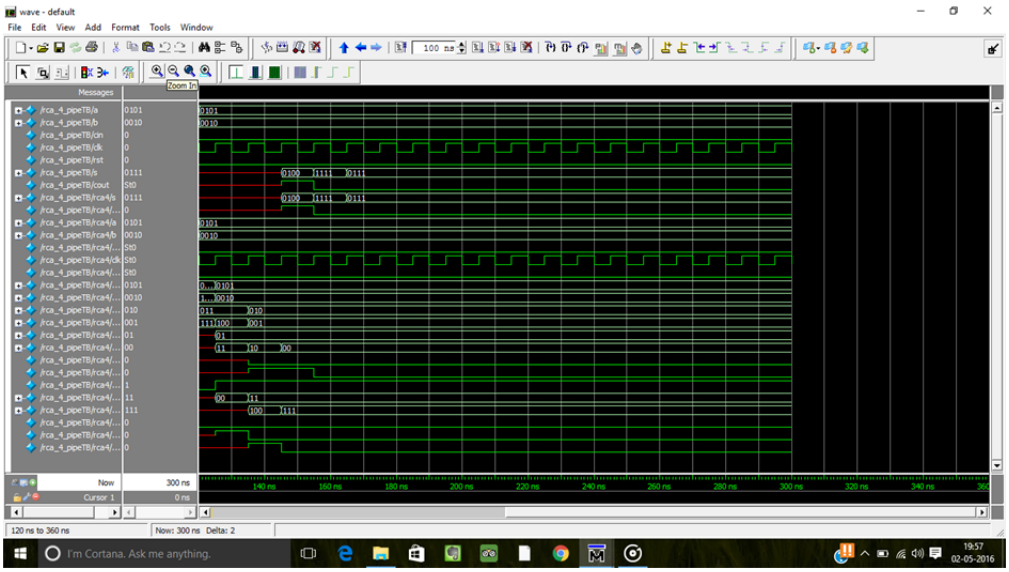

Fig 9.13: Simulation Waveform of pipelined 4-bit Ripple Carry Adder

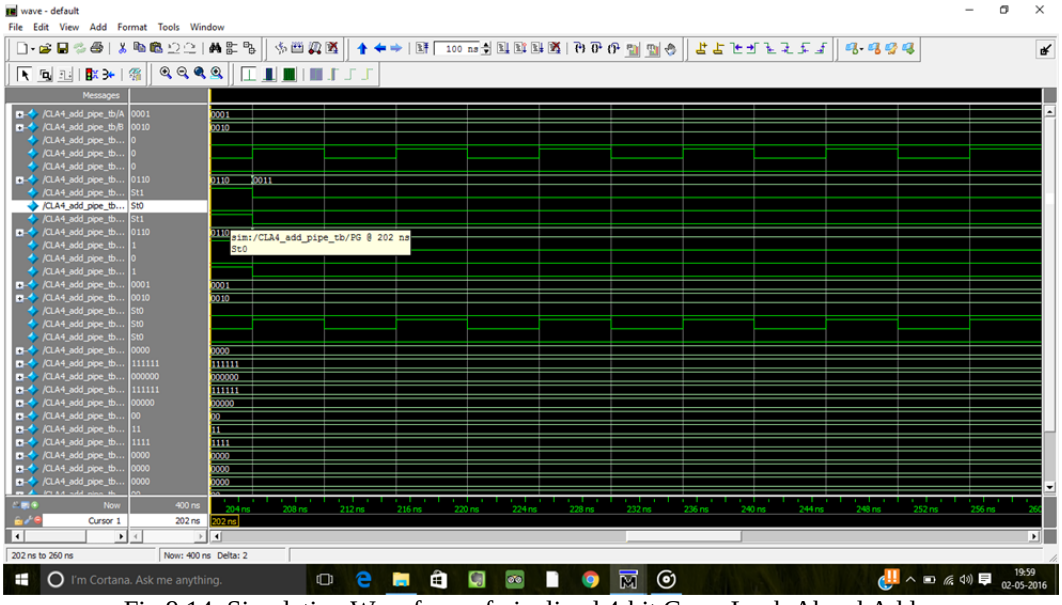

Fig 9.14: Simulation Waveform of pipelined 4-bit Carry Look Ahead Adder

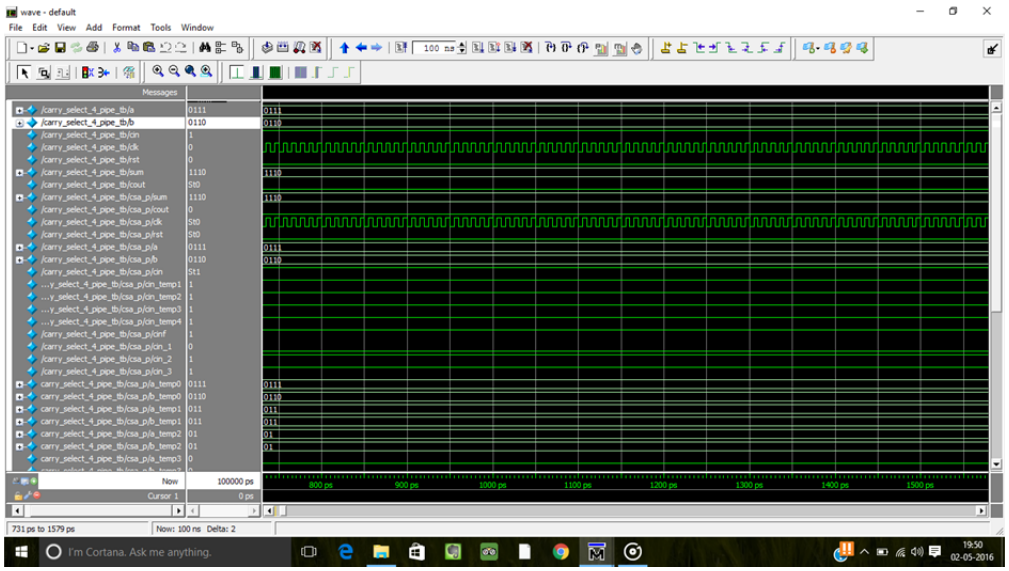

Fig 9.15: Simulation Waveform of pipelined 4-bit Carry Select Adder

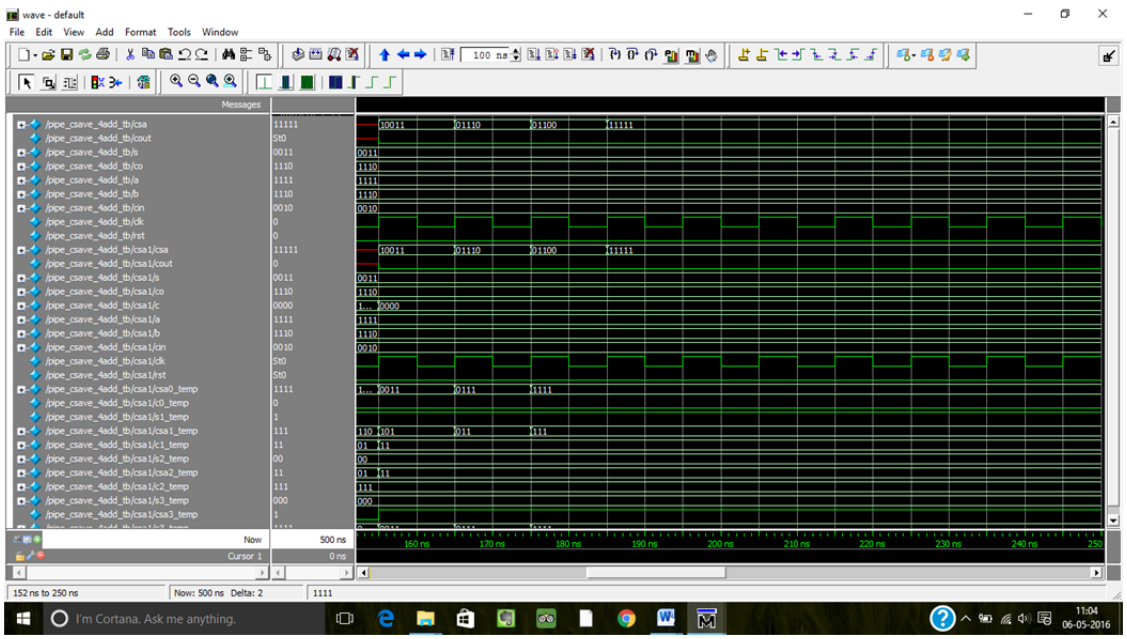

Fig 9.16: Simulation Waveform of pipelined 4-bit Carry Save Adder

10. COMPARISION TABLE

PARAMETERS	RIPPLE CARRY ADDER	CARYY LOOK AHEAD ADDER	CARRY SELECT ADDER	CARRY SAVE ADDER	KOGGE STONE ADDER	LADNER FISCHER ADDER
LUTs	16	17	29	30	25	16
SLICE	12	9	15	18	13	10
IOBs	26	16	26	50	26	26
DELAYS(ns)	13.20	12.344	12.679	12.699	9.123	11.073
TOTAL MEMORY USSAGE(Kb)	294436	294408	310592	294448	294344	294388

Table 10.1: Comparison Table of 4-Bit Adders

PARAMETERS	RIPPLE CARRY ADDER	CARYY LOOK AHEAD ADDER	CARRY SELECT ADDER	CARRY SAVE ADDER	KOGGE STONE ADDER	LADNER FISCHER ADDER
LUTs	8	40	29	30	9	7
SLICE	6	22	15	18	5	4
IOBs	14	28	26	50	15	14
DELAYS(ns)	8.959	12.344	8.217	12.699	7.963	7.859
TOTAL MEMORY USSAGE(Kb)	294412	290240	294420	294408	294380	294384

Table 10.2: Comparison Table of 8-Bit Adders

PARAMETERS	RIPPLE CARRY ADDER	CARYY LOOK AHEAD ADDER	CARRY SELECT ADDER	CARRY SAVE ADDER
LUTs	15	47	32	25
SLICE	16	32	23	19
IOBs	16	18	16	28
DELAYS(s)	7	8	8	9
TOTAL MEMORY USSAGE(Kb)	294400	310616	294372	294380

Table 10.3: Comparison Table of 4-Bit Pipeline Adders

11. CONCLUSION

From the above performance analysis table 10.1, we can observe that area wise (LUTs, SLICE, IOBs) 4-bit Carry Look Ahead Adder is best suited whereas 4-bit Kogge-Stone Adder is best suited in terms of delays and memory usage.

By looking into the performance analysis table 10.2, we get to know that 8-bit Ladner Fischer Adder is best suited in terms of area (LUTs, SLICE, IOBs) and delay whereas memory usage of 8-bit Carry Look Ahead Adder is better than 8-bit Ladner Fischer Adder.

By going through the performance analysis table 10.3, we find that area (LUTs, SLICE, IOBs) and delay wise 4-bit pipeline Ripple Carry Adder is best suited whereas 4-bit pipeline Carry Select Adder is best suited in terms of memory usage wise.

REFERENCES

1. Vaibhav Gupta, Debabrata Mohapatra, Anand Raghunathan, and Kaushik Roy, *Low-Power Digital Signal Processing using Approximate Adders*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 32, no. 1, pp.124-137,Jan 2013.
2. Singh, R.P.P.; Kumar, P.; Singh, B., "Performance Analysis of Fast Adders Using VHDL", Advances in Recent Technologies in Communication and Computing, 2009.
3. O. Kwon, E. Swartzlander, and K. Nowka, "A fast hybrid carry-lookahead/carry-select adder design", Proc. of the 11th Great Lakes symposium on VLSI, pp.149-152, March 2001.
4. ` N. M. Chore, and R. N. Mandavgane, "A Survey of Low Power High Speed 1 Bit Full Adder", Proceeding of the 12th International Conference on Networking, VLSI and Signal Processing, pp. 302-307, 2010.
5. Weste Neil, Harris David and Banerjee Ayan (2009), "CMOS VLSI Design: A Circuits and System Perspective",Pearson Education.
6. P. Kogge and H. Stone, .A parallel algorithm for the efficient solution of a general class of recurrence relations,. *IEEE Trans. Computers*, vol. C-22, no. 8, pp. 786.793, Aug. 1973.
7. R. Ladner and M. Fischer,"Parallel prefix computation",Journal of ACM. La. Jolla CA, Vol.27, pp.831-838,October 1980.
8. Kogge P and Stone H, "A Parallel Algorithm for the Efficient Solutions of a General Class of Recurrence Relations", IEEE Transactions on Computers, Vol. C-22,No.8, 1973.
9. P. M. Kogge and H. S. Stone, "A Parallel Algorithm for the Efficient Solution of a General Class of Recurrence Equations," IEEE Trans. on Computers, Vol. C-22, No 8, August 1973.
10. J. M. Rabaey, "Digital Integrated Circuits- A Design Perspective", New Jersey, Prentice-Hall, 2001..