Implementation of 64 Bit Arithmetic Adders

N Aruna Kumari, M Sai Srinivas, K Aravind

Abstract: ***Adders are very essential components in integrated circuits. In the applications of Digital Signal Processing (DSP) adders are very much required. Researchers are trying to design adders which are fast, power efficient and occupies less area. Adders play vital role in modern applications. In integrated circuit designs power, area and speed are the key parameters while building a circuit. Research is going on to built adders which consume low power, less space on chip and fast or combination of these three parameters. In our survey, the implementation of different adders like Ripple Carry Adder, Carry Increment Adder, Carry Skip Adder, Carry Select Adder , Carry Look Ahead Adder, Brent Kung Adder, Sklansky Adder, Kogge-Stone Adder, Ladner-Fischer Adder, Knowles Adder, Han-Carlson Adders were discussed. We did the comparison based on the delay.***

Keywords: Ripple Carry Adder, Carry Skip Adder, Carry Increment Adder, Carry Look Ahead Adder, Carry Select Adder, BrentKung Adder, Sklansky Adder, Kogge Stone Adder, Ladner Fischer Adder, Knowles Adder, Han Carlson Adder

I. INTRODUCTION

An adder is an element which performs addition of numbers. Adders are useful in arithmetic logical units(ALU) which are widely used in computers and different types of processors. For effective and efficient implementation of arithmetic components adders play an important role. Adders can be treated as building blocks of the arithmetic component. For the operations like complementing, decoding and encoding adders are used. Generally addition involves adding of two numbers which generates sum and carry. All adder architectures either simple or complex are constructed by using fundamental blocks which are half adder and full adder.

For small number of bits, simple adders like ripple carry adder, carry look ahead adders are sufficient. However delay increases as the bits number increases because of the passing of the carry to the next stage.

So we use Parallel Prefix Adders to perform arithmetic operations on large number of bits. Parallel prefix adders are high speed adders and takes small area and gives less delay. These adders consume low power and relatively takes less area on chip. Primary concern of adders is speed and later we have chip area and power consumption of adder.

Classification of adders:

Carry Propagate Adders:
Ripple Carry Adder
Carry Look ahead Adder
Carry Increment Adder
Carry Skip Adder
Carry Select Adder
Parallel Prefix Adders(Tree Adders):
Brent Kung Adder
Sklansky Adder
Kogge-Stone Adder
Ladner-Fischer Adder
Knowles Adder
Han-Carlson Adder

Carry Propagate Adders:

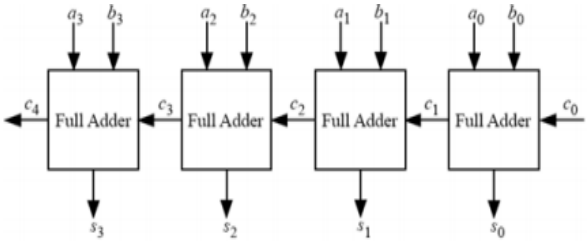

Fig 1:Ripple Carry Adder Circuit

Ripple Carry Adder (RCA) works on the basic addition principle. RCA is basic adder.

RCA consists of full adders(FA) in series as shown in above figure.

Each FA is used for addition of two bits in addition to carry bit. The generated carry will pass to next full adder and this process continues till the end.

As number of bits increases delay increases linearly.

2. Carry-lookahead adder

Carry-look ahead adder takes less amount of time to calculate carry bits. Carry look ahead adder having three levels which was shown in below figure.

To generate end carry it takes 3 levels of delay.

To generate sum it takes 4 levels of delay.

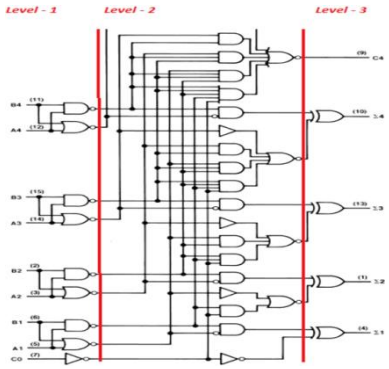

Fig 2: Carry look ahead adder

Revised Manuscript Received on February 11 , 2019.

N ArunaKumari, Electronics and Communication Engineering Department RGUKT IIIT, Nuzvid, Andhra Pradesh, India.

M Sai Srinivas, Electronics and Communication Engineering Department RGUKT IIIT, Nuzvid, Andhra Pradesh, India.

K Aravind, Electronics and Communication Engineering Department RGUKT IIIT, Nuzvid, Andhra Pradesh, India.

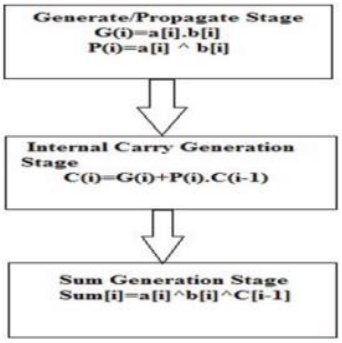

Fig 3:Algorithmof operation of Carry Lookahead Adder

$$C1 = G0 + P0 \bullet C0$$
$$C2 = G1 + G0 \bullet P1 + C0 \bullet P0 \bullet P1$$
$$C3 = G2 + G1 \bullet P2 + G + 0 \bullet P1 \bullet P2 + C0 \bullet P0 \bullet P1 \bullet P2$$
$$C4 = G3 + G1 \bullet P2 \bullet P3 + G0 \bullet P1 \bullet P2 \bullet P3 + C0 \bullet P0 \bullet P1 \bullet P2 \bullet P3$$

Fig 4: Generation of carrys

3. CARRY INCREMENT ADDER

An eight bit Carry Increment Adder (CIA) looks like as shown in figure 4. CIA contains RCA's and an incremental circuit. The incremental circuit is build by using Half Adders (HA) connected in ripple Carry fashion sequentially.

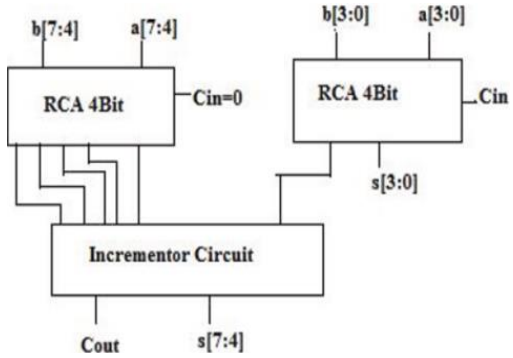

Fig 4:Carry Increment Adder

Algorithm for operation of Carry Increment Adder:

For a 8 - bit Carry Increment Adder as shown above contains two 4 – bit Ripple Carry Adders.

step 1 : Calculate values of sums[3:0]S , [4:7]s , *C*3 and *C7* in parallel by using these RCA'S.

step 2 : Take carry from first RCA (*C*3) , LSB sum bit (*s*4) from the second RCA and give them as inputs for the first half adder in the incremental circuit. The resultant sum bit is *S*5(overall sum bit).

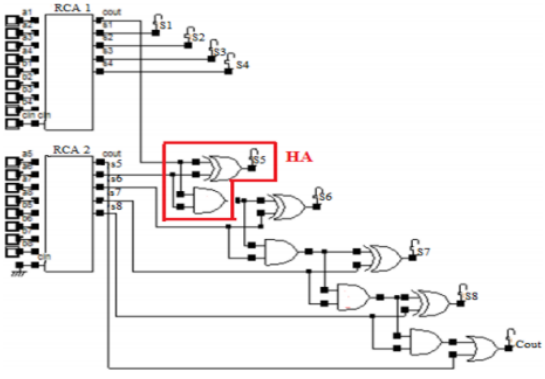

Fig 5:Carry Increment Adder operation

step 3 : Take carry form the previous half adder , sum bit (*s*5) from the second RCA and give them as inputs for the second half adder in the incremental circuit. The resultant sum bit is S6.

step 4 : Like this we can get the sum bits and carry out. Total delay = 4 bit RCA(using full adders)delay + 4 bit RCA (using half adders)delay.

4. CARRY SKIP ADDER

We can improve the delay in carry skip adder compared to ripple carry adder by doing minor modifications to the ripple carry adder. The n-bit CSA contains a n-bit carry ripple-chain, an and gate of n-input and a multiplexer component. For the n-input AND-gate we connect propagate bit(Pi) which is generated by carry-ripple-chain. The resulting bit from and gate acts as the selection bit for the multiplexer which give outputs either last carry bit (Cn) or the input carry(C0) which is nothing but carry-out signal (C_{out}).

$$S = P_{n-1} \wedge P_{n-2} \wedge P_{n-3} \ldots\ldots P1 \wedge P_0$$
$$= P_{[0:\, n-1]}$$

The operation of Carry skip adder

Case 1: All bits generate propagate (i,e., *p*0= *p*1 = *p*2 = *p*3 = 1) Ex : A = 1010, B = 0101

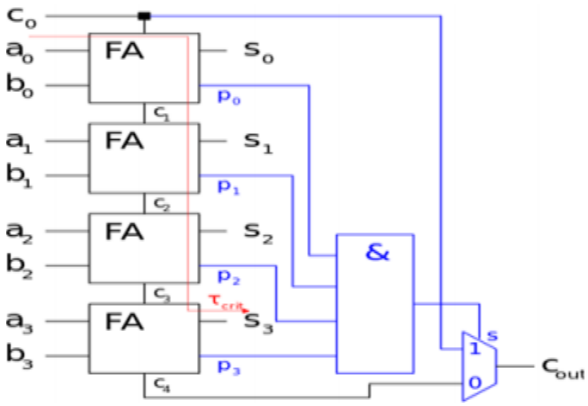

Fig 6:carry-skip adder

Here we do not wait for generation of last carry and we can equate final carry (C3) to input carry (C0). So we will get delay of RCA only.

Case 2: Here we can not generate propagates for all bits. Ex : A = 1010,B = 0111. Here we need to wait for final carry (C3). So the delay is greater than the RCA delay.

5. CARRY SELECT ADDER

The carry select adder contains two Ripple-carry adders and a multiplexer. It calculates addition of 2 n-bit groups twice by involving two adders one with the assuming carry-in as zero and the other time as one. At next level with multiplexer which is given with correct carry-in known as selector the result sum bit signals and carry out bit signals will be selected.

If we consider a 64- bit carry select adder then the structure is like as shown

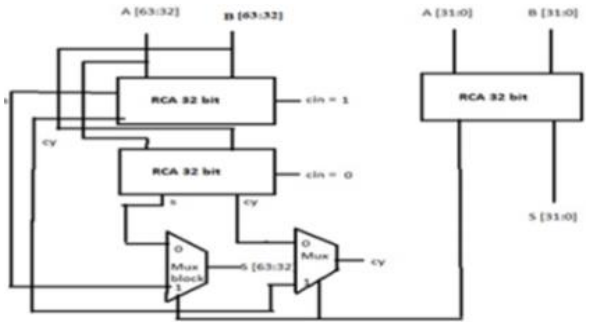

Fig 7:64- bit carry select adder

II. PARALLEL PREFIX ADDERS

Ripple carry adders (RCA) offers linear delay and are uncomplicated adders but the slowest ones. Carry look Ahead adder (CLA) has less time delay and developed to parallel-prefix structures called tree adders. Carry Look Ahead adders are similar to Parallel-Prefix adders in terminology.

These adders perform parallel addition which is important aspect in microprocessors, Digital signal processing and other high speed applications. Parallel Prefix adder lowers logic complexity and time delay thereby improve performance. So Parallel Prefix adders are required elements in the high speed arithmetic circuits.

Tree adders generates carries in parallel so that they compute fast but with increased area and power. The carry tree decreases total number of logic levels (N) by calculating carries in parallel. This is the advantage of prefix adders. In comparison with carry path of other adders the parallel-prefix adders are more reliable in terms of speed since the delay complexity is of the order O(log2N).

Parallel prefix computation has the following important steps:

Computing carry generation(G) and carry propagation(P) bits using input bits. This stage consists of computation of propagate and generate bits respecting to each pair of bits in A and B. According to prefix computation *Gi* (generate) and *Pi* (propagate) signals are defined by the equations(1) and (2) respectively.

Pi=Ai^Bi.....(1)

Gi= Ai • Bi.....(2)

prefix computation is calculating all carry signals in parallel. This step consists of computation of generate and propagate carries respecting to each bit. These computations execute in parallel. After the computation of carries they are divided into smaller segments. carry propagate and generate bits are used as intermediate signals. Carry generating operation block is important in parallel prefix adders, and consists three types of components called

1. Black Cell
2. Grey Cell
3. Buffer

Black cells compute both carry generate and propagate bits.

Similarly in the calculation of carrys done by which are used in post-processing stage to calculate sum.

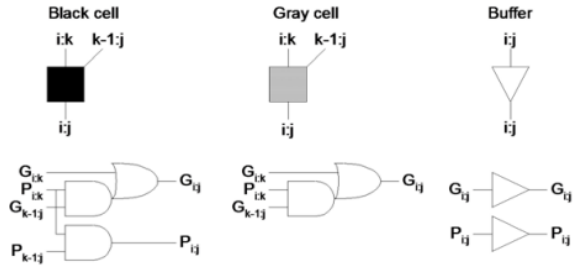

Fig 8: Carry generation and propagation cells architecture in Parallel Prefix Adders

Buffers are used to balance the loading effect.

Final stage consists of a simple adder circuit to generate final sum bit.

Below computations represent the three stages of parallel prefix adder in their respective structure:

Pre − computation :

Gm:n= An•Bn; Pm:n =An^Bn, P0 = 0

Prefix − computation :

Gm:n= Gn:k+ Pn:k• Gk-1:1 and

P m:n = Pn:k•Pk−1: j

Post − computation :

Sn= Pn ^Gn-1:0

CARRY LOOKAHEAD TREES

*C*1:0 = *G*1 + *G*0*P*1 + *C*0*P*0*P*1

*C*2:0 = *G*2 + *P*2*G*1:0

*C*2:0 = *G*2 + *G*1*P*2 + *G*0*P*1*P*2 + *C*0*P*0*P*1*P*2

=*G*2:1 + *P*2:1*C*0

With this basic concept we can implement the different types of tree adders

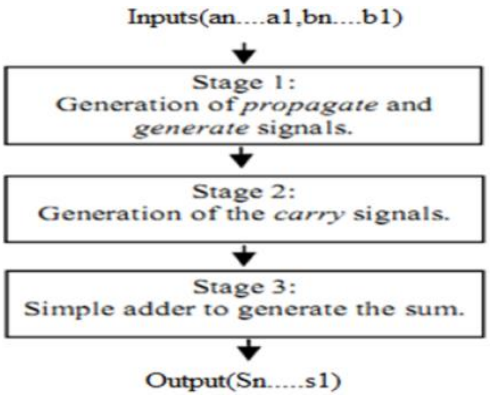

Fig 9: Operation of Parallel Prefix Adders

1. KOGGE STONE ADDER

KoggeStone adder can be treated as the parallel-prefix type carry look ahead adder. Kogg-Stone adder was implemented by Peter M.Kogge and HarodS.Stone in 1973.

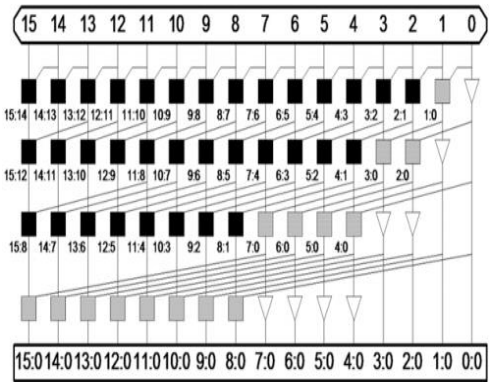

Fig 10: Kogge Stone Adder

Kogge Stone Adder carry generation stage

Every column stage produces both propagate as well as generate signals. Generate signals which are calculated in the final stage are made XOR with initially produced propagate and generate signals to produce sum. The advantage in Kogge-Stone adder is that it generates carry bits in O (log2 n) time delay complexity which made itself gives best performance in VLSI implemented circuits.

KS has minimum fan-out with large area. It reduces the critical path to great extent so that increases it's performance in implementing to higher bit adders like 32-bit, 64-bit, and 128-bitComparing with Brent-Kung adder.

KS takes greater area for implementing but it has lesser fan-out. Which makes wiring congestion of this adder is a problem.

Maximum fan-out: 2.

2. BRENT KUNG ADDER

Brent-Kung adder was implemented by Brent and Kung which was published in the year of 1982.

Brent Kung Adder carry generation stage

The Brent-Kung adder computes odd prefix bits first and then even like in figure calculated carries c1, c3, c7, c15. It computes prefix bits for 2 input bit groups. These act as inputs for next cell groups to compute prefixes for 4-bit groups, which in turn computes prefixes for 8-bit groups, and so on.

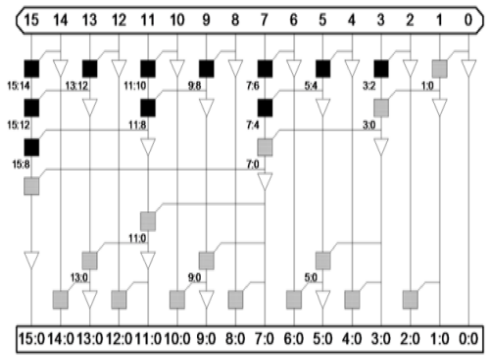

Fig 11: Brent Kung Adder

These prefix bits used to compute the carries for each bit in the next level. Generated carries will be XOR with the group propagate bits of the next stage for computing the Sum bit.

The advantage of this adder is that it takes less area which makes the implemented adder with less wiring congestion compared to the other prefix adders like KoggeStone adder. This adder has logic depth in maximum(gives longer time delay) and number of nodes in minimum(gives lesser area).This will also reduce the delay when implemented higher bit adders without compromising the power performance of the adder.

Maximum fan-out: 2.

3. LADNER-FISCHERADDER

R.Ladner and M. Fischer developed Ladner-Fischer parallel prefix adder in 1980. This tree structure represents mediated structure of Brent-Kung prefix tree and Sklansky prefix tree.

Ladner fischer adder carry generation stage

In the initial stage it computes prefixes for odd number bits and it uses an another additional stage to propagate the odd number bits to the even positions. This adder has minimum logic depth but it is having more fan-out. Fig 2.18 shows the 16-bit LF adder.

Maximum fan-out: n/2.

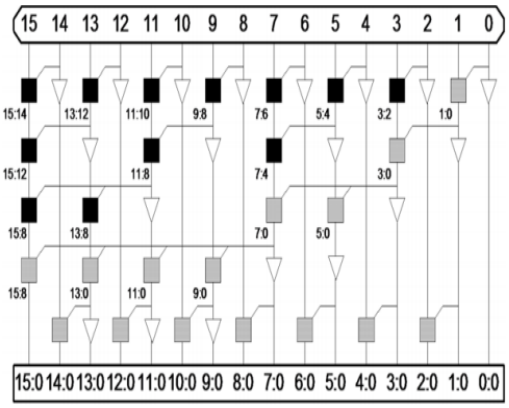

Fig 12: Ladner Fischer adder

4. HAN-CARLSON ADDER

The Han Carlson adder can be treated as represents a hybrid tree structure of the Brent Kung and Kogge Stone structures. Han-Carlson adder can also be represented as a scattered version of Kogge Stone adder.

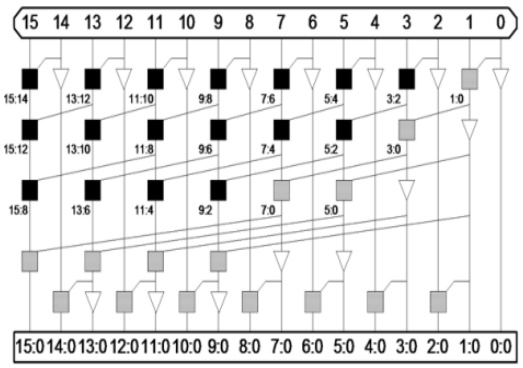

Fig 13:Han-Carlson adder

Han carlson adder carry generation stage

Carry generation of HC involves five stages. Central three stages be like the Kogge-Stone tree structure. This structure is different KS tree structure as it performs carry merge operations on even bits like in figure2.19.by calculating c2, c4, c6, c8, c10, c12, c14 and generate, propagate operations on odd bits. At last, these odd group propagated signals merged with former odd bit carry bits and produce the final carry bits. This adder uses less Black cells and it has shorter wire complexity than KS adder .The complexity can be reduced at the cost of an extral stage for carry merge path. [1]

Maximum fan-out: 2.

5. KNOWLES ADDER

A family of prefix tree structures is proposed by S.Knowles with flexible architectures. Knowles tree structure is a combination of tree networks Kogge Stone and Sklansky adders.[1]

Maximum fan-out: 3.

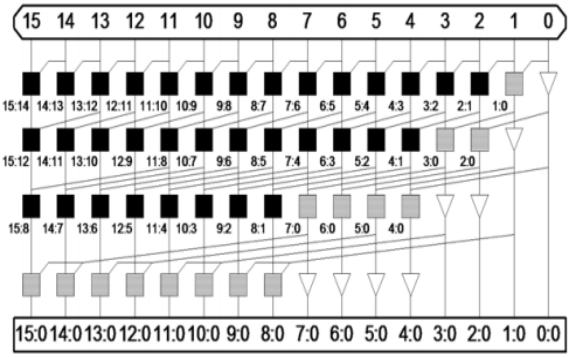

Fig 14: Knowles adder

6. SKLANSKY ADDER

Sklansky adder has simple prefix tree structure.

The Sklansky represents divide and conquer structure. This Structure computes prefixes recursively for 2-bit groups, 4-bit groups, 8-bit groupsthen16-bit groups and so on by adding two smaller adders in each level.[2]

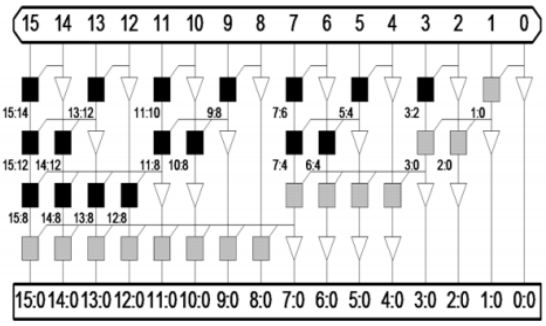

Fig 15: Sklansky Adder

Sklansky adder carry generation stage

By divide and conquer concept it gives the delay of log2n stages by calculating intermediate prefix bits along with the large group prefixes. The advantage of this adder is architecture is simple and regular but fanouts double at each stages it has fan out problem.

Maximum fan-out: (n/2)+1.

III. CRITICAL PATH:

DEFINITION: The critical path can be defined as the maximum delay path between the input and output.

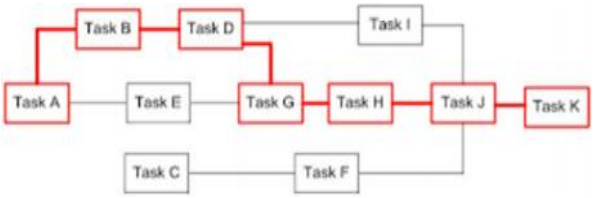

Fig 16: Critical path

CRITICAL PATHS FOR DIFFERENT ADDERS:

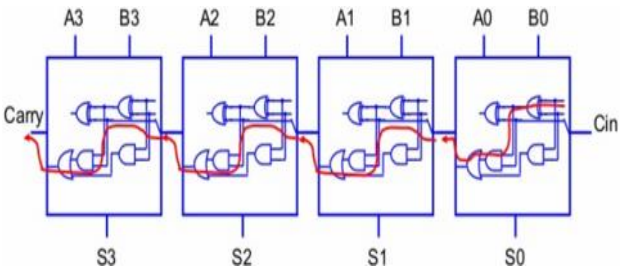

Fig 17: Critical path for 4-bit ripple carry adder

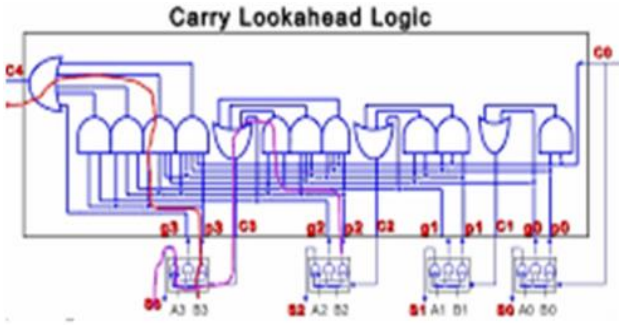

Fig 18: Critical path for 4-bit carry look ahead adder

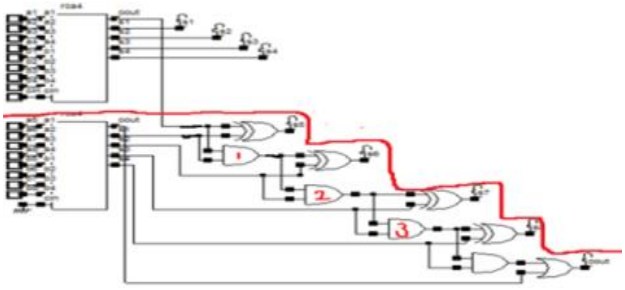

Fig 19: Critical path for 8- bit carry increment adder

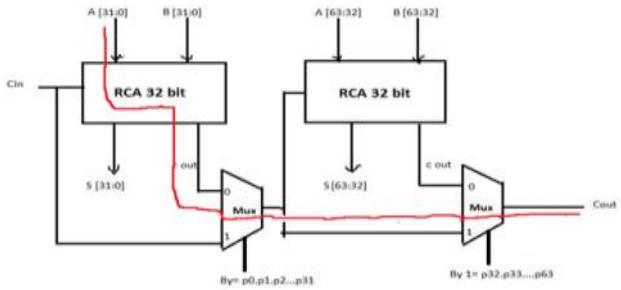

Fig 20: Critical path for 64-bit carry bypass adder

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

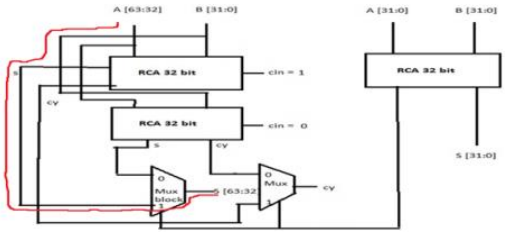

Fig 21: Critical path for 64- bit carry select adder

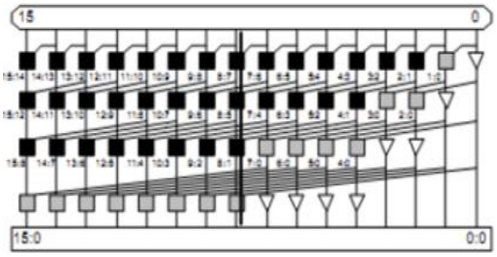

Fig 22: Critical path for 16-bit Kogge Stone adder

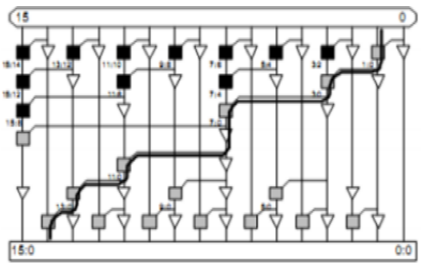

Fig 23: Critical path for 16- bit Brent Kung adder

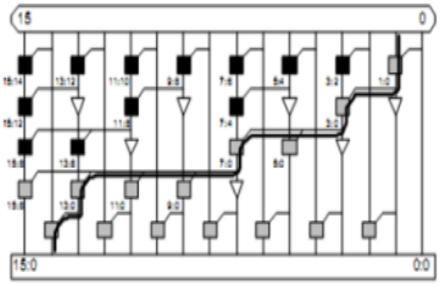

Fig 24: Critical path for 16-bit Ladner Fischer adder

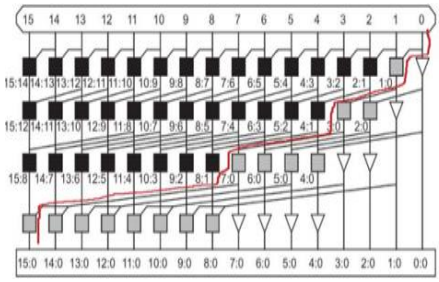

Fig 25: Critical path for 16 -bit Knowles adder

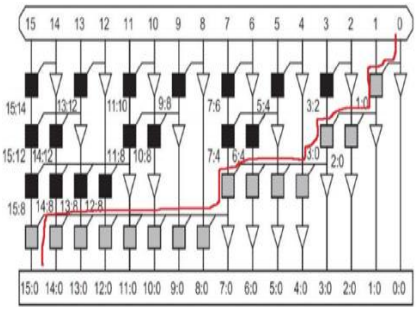

Fig 26: Critical path for 16-bit Sklansky adder

N-BIT ADDER	MAXIMUM COMINATIONAL PATH DELAY
RIPPLE CARRY ADDER	=N*(Delay of Full adder) =N*[Delay of (XOR gate+ AND gate +OR gate]
CARRY INCREMENT ADDER	=[N/2*(Delay of Full adder)+N/2(Delay of Full adder)+ Delay of OR GATE]
CARRY SKIP ADDER	Case 1(p0 = p1 = p2 = p3 = 1): Delay = N*(Delay of Full adder) Case 2(any Pi is not equal to 1)Delay = [N* (Delay of Full adder) + multiplexer delay]
CARRY BYPASS ADDER	Case 1(p0 = p1 = p2 = p63 = 1):Delay = N/2*(Delay of Full adder) Case 2(any Pi is not equal to 1)Delay = [N* (Delay of Full adder) + 2*(Multiplexer delay)]
CARRY SELECT ADDER	=[N/2*(Full adder delay)+ Delay of one multiplexer]
CARRY LOOKAHEAD ADDER	sum = 4 units delay , carry= 3 units delay
BRENT KUNG ADDER	$2\log_2^n - 1$
KOGGE STONE ADDER	$\log_2^n$
LADNER FISCHER ADDER	$\log_2^n + 1$
KNOWLES ADDER	$\log_2^n$
HANCARLSON ADDER	$\log_2^n$
SKLANSKY ADDER	$\log_2^n$

Fig 27: Maximum combinational Path for Different adders(N-BIT)

ADDER	DELAY(ns)
RIPPLE CARRY ADDER	95.75
CARRY LOOKAHEAD ADDER	37.12
CARRY INCREMENT ADDER	70.79
CARRY SKIP ADDER	92.12
CARRY BYPASS ADDER	57.85
BRENT KUNG ADDER	13.971
SKLANSKY ADDER	18.185
KOGGE STONE ADDER	28.526
LADNER FISCHER ADDER	21.029
KNOWLES ADDER	31.057
HANCARLSON ADDER	9.406
ADDER USING XST	10.049

Fig 28: DELAY COMPARISON OF ADDERS(64-BIT)

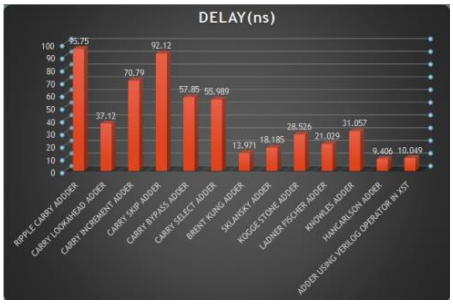

Fig 29: DELAY COMPARISON

IV. RESULTS(64-BIT)

USING ISE DESIGN SUIT 14.4
Synthesis Tool : XST(VERILOG)
Simulator : ISim Simulator

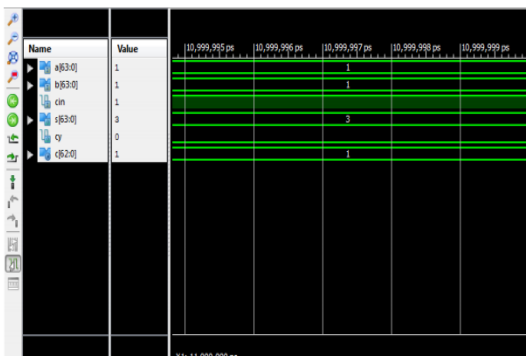

Fig30: 64-bit ripple carry adder result

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

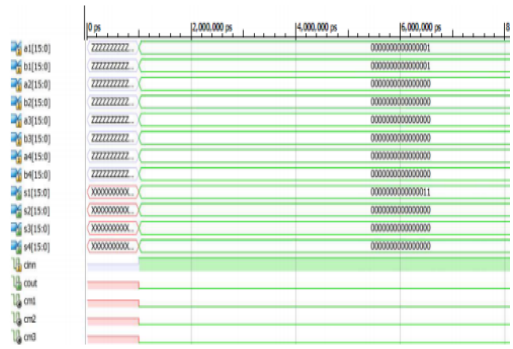

Fig 31: 64-bit carry lookahead adder result

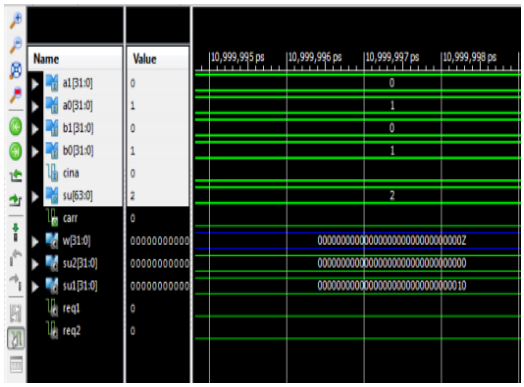

Fig 32: 64-bit carry increment adder result

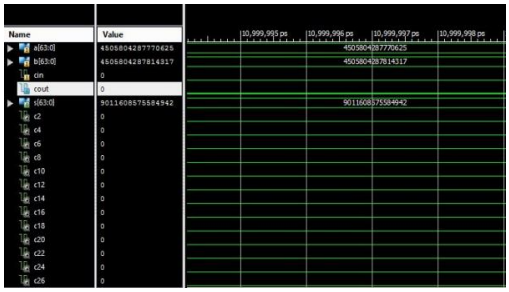

Fig 32: 64-bit carry skip adder result

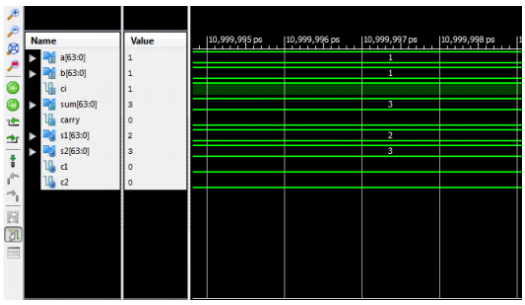

Fig 33: 64-bit carry select adder result

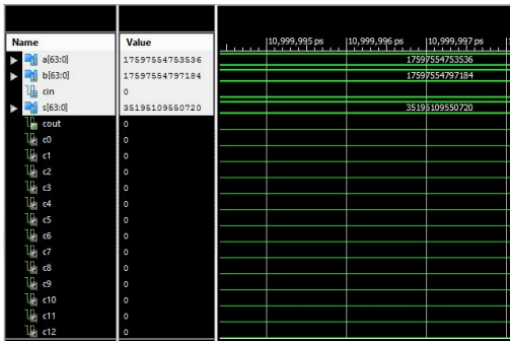

Fig 34: 64-bit Koggge stone adder result

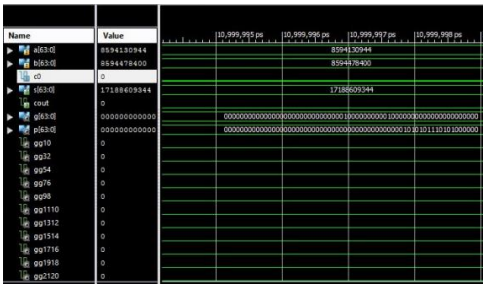

Fig 35: 64-bit Brent kung adder result

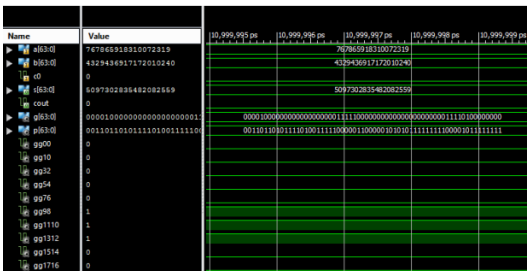

Fig 36: 64-bit Ladner fischer adder result

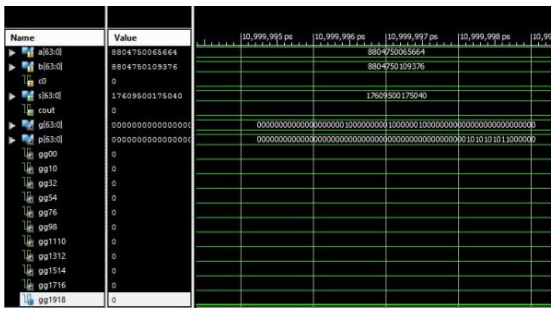

Fig 37: 64-bit Han carlson adder result

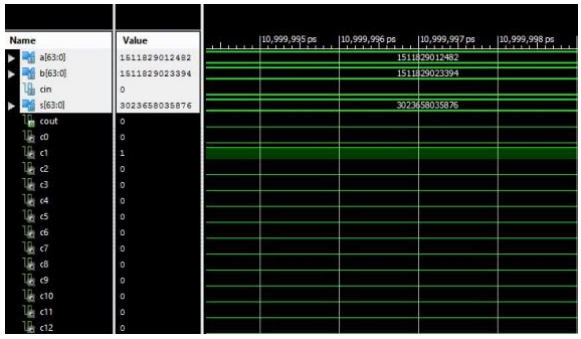

Fig 38: 64-bit Knowles adder result

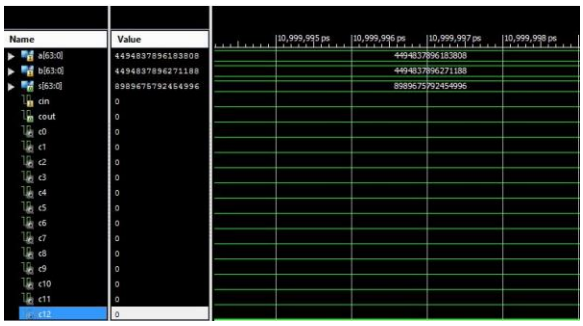

Fig 39: 64-bit Sklansky adder result

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

SUM OF TWO NUMBERS USING XILINX TOOL (64-BIT)

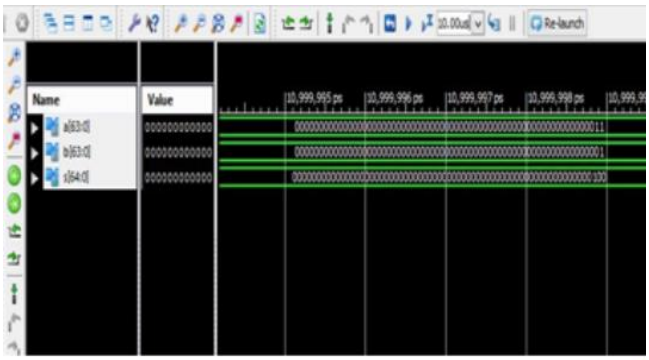

Fig 40: sum of two 64-bit numbers using XST

V. CONCLUSION ANDFUTURE SCOPE

- From the delay comparison table Han Carlson Adder has least delay.
- From the gate count comparison table Knowles adder has high circuit complexity.
- It's circuit complexity is in comparable with carry look ahead adder as we go for 64 bit addition.
- By using carry propagate adders we can get less delay for lower bit addition.
- As we go beyond that carry look ahead adder becomes more complex and requires high area.So we prefer parallel prefix adders at higher bit addition.
- For the addition upto 64 bits we can get least delay by using HANCARLSON Adder.
- By taking gate count also into consideration Brent Kung adder is preferred.
- Han Carlson Adder got the comparable delay with the Adder using verilog operator in XST.

VI. FUTURE WORK

The designs can be further developed for higher bits. These designs can be implemented on FPGA and ASIC also. Also by combining the different tree adders as well as the technology used to implement them, a suitable adder with significant less time delay may be achieved.

REFERENCES

1. Pudi. V, Sridhara., K,"Low Complexity Design of RippleCarry and Brent KungAddersinQCA", Nanotechnology, IEEE transactions on, Vol. 11, Issue 1, pp. 105-119, 2012.
2. Vibhuti Dave, ErdaOruklu and JafarSaniie, "Performance Evaluation ofFlagged Prefix Adders for Constant Addition", Department of Electrical andComputer Engineering, Illilois Institute of technology, Chicago, 200630
3. Comparison Between Various Types of Adder Topologies 1Jasbir Kaur, 2Lalit Sood,IJCST Vol. 6, Issue 1, Jan - March 2015 ISSN : 0976-8491 (Online) | ISSN : 2229-4333 (Print)
4. Yousuf, Romana& , Najeeb-ud-din. (2008). Synthesis of carry select adder in 65 nm FPGA.IEEE Region 10 Annual International Conference, Proceedings/TENCON.1-6. 10.1109/TENCON.2008.4766397.
5. O. J. Bedrij, "Carry-select adder," IRE Trans. Electron.Comput.,vol. EC-11, no. 3, pp. 340–344, Jun. 1962.
6. Sarabdeep Singh, DilipKumar,"Design of Area and PowerEfficient Modified Carry Select Adder", International Journalof Computer Applications, Vol. 33, No. 3, pp. 14-18, Nov2011.
7. Animulislam, M.W. Akram, S.D. pable, Mohd. Hasan,"Design and Analysis of Robust Dual Threshold CMOS FullAdder Circuit in 32 nm Technology", International Conferenceon Advanced in Recent Technologies in Communication andComputing, 2010.
8. DeepaSinha, Tripti Sharma, K.G.Sharma, Prof.B.P.Singh,"Design and Analysis of low Power 1-bit Full AdderCell",IEEE, 2011.
9. Nabihah Ahmad, RezaulHasan,"A new Design of XORXNORgates for Low Power application", InternationalConference on Electronic Devices, Systems and Applications(ICEDSA), 2011.
10. Padma Devi, AshimaGirdher, Balwinder Singh, "ImprovedCarry Select Adder with Reduced Area and Low PowerConsumption", International Journal of Computer Application, Vol. 3, No. 4, June 2010 .

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication