International Journal of Advance Research in Science and Engineering
Vol. No.6, Issue No. 01 , January 2017
www.ijarse.com

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

STUDY OF DIFFERENT ADDERS AND ANALYZE THE DELAY

Gitanjali Sharma

Department of Electronics & Communication,
ABES Engineering College, Ghaziabad (India)

ABSTRACT

In this paper, comparison of different adders, different implementation techniques and architecture of various kind of adders. Here we will be finding most relevant adder for different purpose of digital signal processing, microprocessors for arithmetic operations. Adders used in all logical functions such as subtraction multiplication and division. A comparative study of different kinds of adder such as, ripple carry adder, carry select adder, carry skip adder, carry lookahead adder, , parallel prefix adder suggest different logic pertaining to different profile consideration. Arithmetic logical unit of any processor used for scientific calculations and need fast adders for the computation and reduce delay. The performance of adders in term of speed and power is crucial for most of the digital signal processing application. The speed of operation of adder is limited by carry propagation and generation from input to output. Speeding the binary addition investigate the adders by all prospective such as delay, power consumption, bit size and area.

Keywords*: Adder, Parallel Prefix Adder, Carry Lookahead Adder, Carry Skip Adder, Carry Select Adder, Ripple Carry Adder, Carry, Delay.***

I INTRODUCTION

Addition of two binary digits is a vitally necessary operation in any Digital, Analog, or Control system. Fast and precise operation of digital system is the demand of the new trends of technology. The maximum operating speed of adders depends on the computation of carries. Specialised speed optimisation of adder architecture are required high performance generate carries. Normally carries generate in two different ways such as serially as conventional methods and other is in parallel form. Some architecture like ripple carry adder[1], carry select adder[6], carry skip adder[7] ,carry lookahead adder[5],parallel prefix adder[3].

The parallel adder is the most important element in the calculation of binary addition in digital system. with the rising trends of mobiles ,high performance integrated circuits, low power consumption digital circuits which has minimum delays in comparison with other devices. CLA proposed almost 20 years ago and its main purpose was accelerating n-bit addition in VLSI technology. It accepted as fasted adder used for high performance arithematic circuits in industries.

II RIPPLE CARRY ADDER

A ripple carry adder is a example of cascade addition. it contains full adder in the circuit to compute the addition of two different binary numbers. In ripple carry adder carry out of each stage full adder used as carry in next significant full adder. This is called ripple carry adder because each carry get ripple to next stage**[5]**.

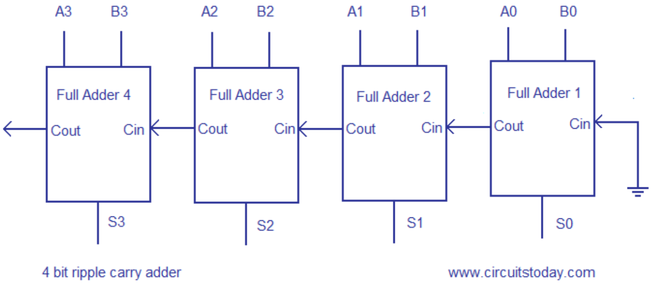


Fig. 1. Ripple Carry Adder[1]

III CARRY SELECT ADDER

A carry select adder architecture has dual ripple carry adder .Performance of this adder has two modules as per RCAs both RCA calculate twice , initially with the assumption of carry in as zero and other assuming it as one. fig. 2. Is explain the function of carry select adder using two RCA. As the bit size increase complexity also increased and as gate are used its effect the delay for example in case of 32 bit carry select adder 95 gate delay will occur due to 31*3 from input to last full adder and 2 for final sum and carry. The delay of adder increase as the size of adder increase in term of bits.

Carry select adder is used in many computational digital system to alleviate the problem of carry propagation delay by generation of multiple carries and then select particulate carry to compute the final sum .carry select adder is complex because its use dual RCA and it is constructed by single-bit full adder and each adder compute still previous carry out signal is ready. Figure 2. Shows the structure of CSA.

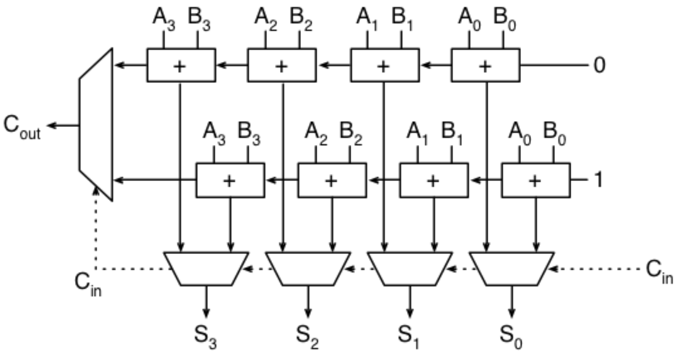


Fig. 2. Stages For Carry Select Adder

IV CARRY SKIP ADDER

The carry skip adder is also known as carry bypass adder. This adder improves the delay of ripple carry adder. It contains ripple carry adder chain. No. Of ripple chain increased with bit size id the adder. Carry skip adder has three sub parts as single Carry skip adder, block Carry skip adder and multilevel Carry skip adder. In Carry skip adder to get the speed-up operation of adder, propagated carry is skipped to i without waiting for ripple. Carry skip adder requires less chip area[]. In carry skip adder modules are divided into r-bit block in simple carry scheme. Carry skip logic is added to each block to force the carry in next block. Each block of stage contains a ripple carry adder to compute the sum and carry out.The block contain propagate and generate signal.[5]

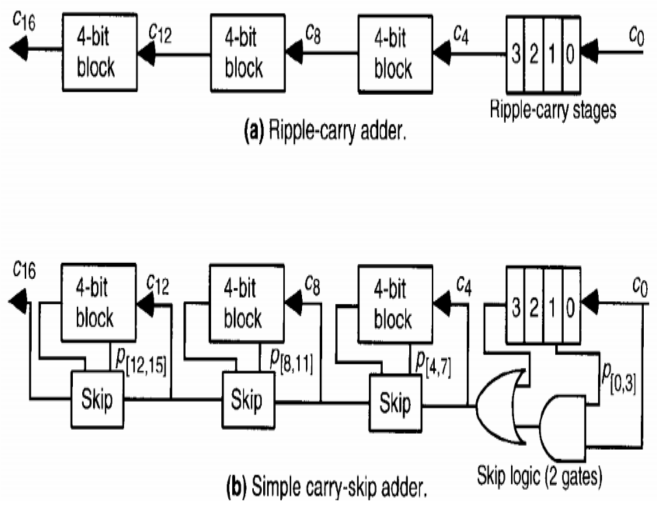


Fig. 3. Stages For Carry Skip Adder[7]

In figure 3, mention two adders ripple carry and simple carry adder. According to figure.3. block 1 using skip logic to propagate carry in next stage. and propagated carry added to the potion of entire adders. Carry skip circuit consist two different logic gates i.e .AND and OR. AND gate compute carry-in and propagated carry with n-bits[7].

V CARRY LOOK-A-HEAD ADDER

This is advanced approaches to calculate the carry before calculating the sum of bits. In previous adders carry propagation delay is of great importance. Different adders design approaches have been employed to overcome the carry propagation delay. Carry look-ahead approach calculates carry in advance based on input signal. CLA carry will be generated in two case.

1) when both input are 1.

2) if either input is 1.

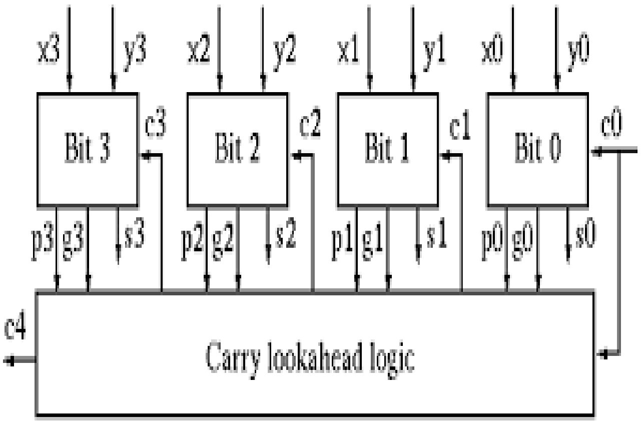


Fig. 4. Stages For Carry Look-Ahead Adder.[5]

In CLA two carry will be calculated internally i.e. propagated carry and generated carry. Equation for P_i and $G_{i\ .}$

P_i =A_i xor B_i

G_i = A_i and B_i

G_i is known as generated carry signal.

P_i is known as propagated ccarry signal.

$$C_1 = G_0+P_0C_0$$

$$C_2 = G_1+P_1C_1$$
$$C_2 = G_1+P_1(G_0+P_0C_0)$$
$$C_2 = G_1+P_1G_0+P_1P_0C_0$$
$$C_3 = G_2+P_2C_2$$
$$C_3 = G_2+P_2(G_1+P_1G_0+P_1P_0C_0)$$
$$C_3 = G_2+P_2G_1+P_2P_1G_0+P_2P_1P_0C_0$$
$$C_4 = G_3+P_3C_3$$
$$C_4 = G_3+P_3(G_2+P_2G_1+P_2P_1G_0+P_2P_1P_0C_0)$$
$$C_4 = G_3+P_3G_2+P_3P_2G_1+P_3P_2P_1G_0+P_3P_2P_1P_0C_0$$

The sun bit and final carry bit will be calculate with the help of propagated carry and generated carry. mathematical equation to add the bits and find out carry bit areas given below:

S_i =P_i xor G_i

C_{i+1}= G_i or(P_i and C_i)

above equation are general equation of CLA. It consist 3 levels of logic. first level generate propagated carry and generated carry of all input bitsit contains two gates XOR and AND gate. second level calculate all carry bit with the help of the previous propagated carry and generated carry as explained in above equation of C_{i+1}.third level use four XOR gates to calculate the sum bit.

VI PARALLEL PREFIX ADDER

The parallel prefix adder is a family of adder derived from CLA. It is further expansion of CLA to reduce time delay. Parallel indicate the execution of small segments of operation parallel.

The only real flaw in KSA is the number of wires gets enormously high. The 16-bit KSA is already filled with cross- connections and get so much worse in 64 bit or 128 bit adders. KSA has best performance in VLSI implementation and it has large area with minimum fan-out.

The Prefix arrangement network describes the type of the Parallel Prefix Adder. The arrangement of the prefix was described by Chung-Kuan Cheng, Haiku Zhu and Ronald Graham; it has the minimal depth for a known 'n' bit adder. Structures of the best logarithmic adder with a fan-out of two for minimizing the area-delay product are proposed by Matthew Ziegler and Mircea Stan. The Sklansky adder proposes a prefix network with a least depth at the cost of enhanced fan-out for specific computational nodes. The algorithm invented by Kogge-Stone has both low fan-out and optimal depth but it introduces problems like large number of interconnections and also account for conjunction circuit realizations. Brent-Kung adder includes the advantage i.e. number of computational nodes are minimum, which results in reduction of area but due to maximum depth of structure, there is slight increase in latency when compared with other models.

The adder proposed by Han-Carlson combines the structures of Kogge-Stone and Brent-Kung adders to achieve a balance between logic depths and interconnections' count. Knowles proposed a category of logarithmic adders with minimum logic depths by keeping the fan-out to maximum. Fischer and Ladner invented a general method to construct a prefix network by keeping the depths slightly higher in comparison with Sklansky topology but achieved some positive results by reduction of the maximum fan-out in critical path for the computation nodes. Yan Sun and et al. presented sparse tree binary adder, it combines the merits of prefix adder and carry look ahead adder. Jainhau Liu and et al. Proposed Integer Linear Programming method to construct parallel prefix adders The proposed Parallel Prefix adder for 64-bits has been proposed.

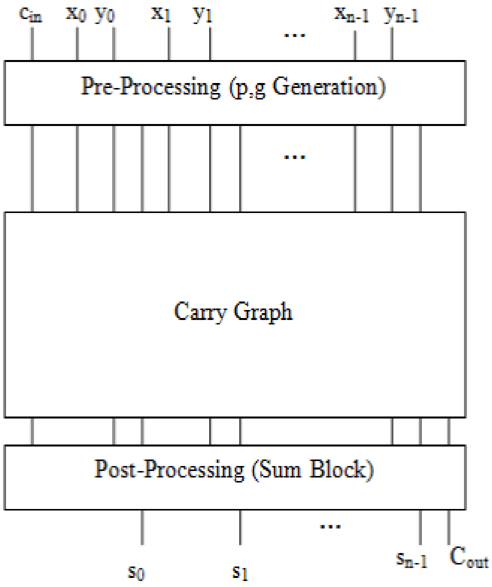


Fig.5 Stages for Parallel Prefix Adder [2]

Above figure shows the stages of parallel prefix adder .it contains three different stages and compute the carry in parallel form. Stages are mention in following steps:

1. Pre-processing stage
2. Carry look-a-head stage
3. post processing stage

VII KOGGE STONE ADDER

The Kogge stone adder is a circuit, combined of carry look ahead adder and parallel prefix adder as it uses functioning of both. KSA is designed to work faster as compared to other adders in its analogy because of it shows the less delay among other architectures. The reason behind lesser delay is functioning of parallel prefix adder, which uses calculations at all stages together rather than calculating each step one by one as that in serial adders. In a circuit of normal adder, it adds 2 gate delays, but KSA can be used to combine any set of P and G signals that are next to each other, and even to combine some P and G signals that are already combined. The Kogge stone adder is fastest possible layout as it scales on logarithmic basis. Every time system adds a combining step, KSA doubles the number of bits that can be added[3].

The only real flaw in KSA is the number of wires gets enormously high. The 16-bit KSA is already filled with cross- connections and get so much worse in 64 bit or 128 bit adders. KSA has best performance in VLSI implementation and it has large area with minimum fan-out. Three modules of Kogge stone adder are given below

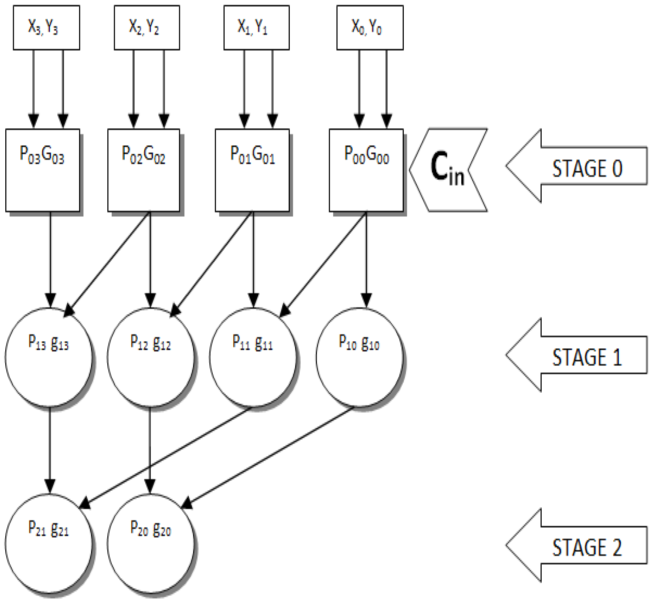


Fig.5 Modules for Kogge Stone Adder

VIII MODULES

A. Computation of Signals

In this module, the generated and propagated signals i.e. Input signals are used to generate carry Input of each adder. 'x_i' and 'y_i' are the input signals, whereas 'P_{0i}' and 'G_{0i}' are the carry input signals. Here 'i' represents bit position. This stage is referred as Stage-'0'. These signals can be calculated by following equations (1) & (2)-

$P_{0i}= x_i$ XOR y_i …………… (1)

$G_{0i}= x_i$ AND y_i …………... (2)

B. Carry Look Ahead Processing

In this stage, carries corresponds to each bit is computed. The carries so generated are called Propagated carry (p_{ji}) and Generated carry (g_{ji}), here 'j' represents stage number. Number of stages can be calculated by the formula-

$N= 2^n$

Where 'N'= Number of bits that are provided as input for addition, 'n'= Number of stages. For example, if we want to add two 4-bit values then number of stages will be two.

This carry look ahead stages are referred as stage-1, 2...j. The execution is conducted in Parallel form and after execution, both of the carries are divided in smaller pieces, a carry operator contains two AND gates and one OR gate. Operator uses propagate and generate as interconnected signals and given by equations (3) & (4)-

$p_{ji}= P_{(j-1)i}$ AND$P_{(j-1)(i-1)}$……………...(3)

$g_{ji}= G_{(j-1)i}$ OR ($P_{(j-1)i}$ AND $G_{(j-1)(i-1)}$)…..(4)

This calculation is for stage-1. As we move from lower number of bits addition to higher additions and number of stages increase, equations (3) and (4) will change according to previous stage. For example for 4-bit addition equations for stage-2 carries will be-

$p_{ji}= p_{(j-1)i}$ AND $p_{(j-1)(i-2)}$……………...(5)

$g_{ji}= g_{(j-1)i}$ OR ($p_{(j-1)i}$ AND $g_{(j-1)(i-2)}$)…....(6)

Above equations show that at a time multiple processing takes place in parallel hence Kogge Stone Adder is normally referred as Parallel Prefix Adder.

C. Summation Process

This is the final stage to calculate summation of the input bits. This stage is same for all kind of carry look ahead adders and parallel prefix adders. Generate bits which are produced in the last stage are XORed with the initial propagated carries after the input to produce the sum bits. Generated carries are treated as final carries in the summation process. The sum bits (S_i) and carry bits (C_i) are given by following equations-

$S_i= P_{0i}$ XOR C_{i-1}………… (7)

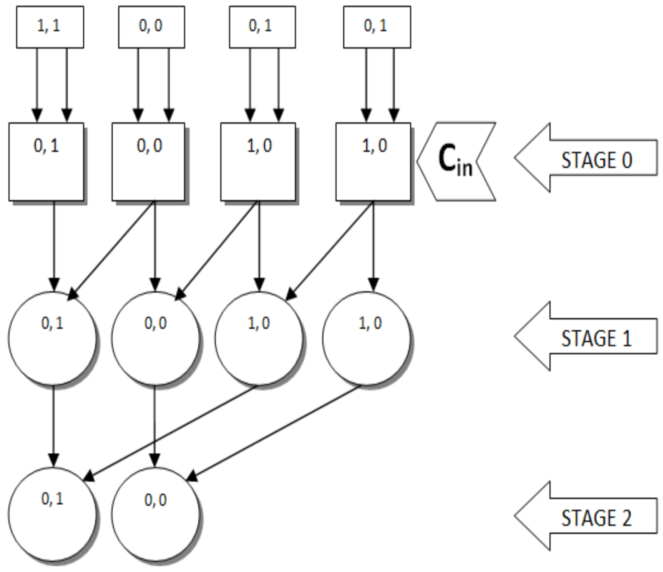
$C_i = g_i$ …………………… (8)

While calculating carries bits, the circuit requires an initial carry i.e. 'C_{in}'. Hence C_{in} is kept '0' initially and therefore the initial carry bit i.e. $C_0 = C_{in}$. C_{in} is also used as a carry while executing initial summation bit i.e. 'S_0'. **[4]**

IX BLOCK DIAGRAM

The figure shows network and stages for a 4-bit Kogge stone adder (see fig. 3). The functioning of higher bits adders like 16 or 32 bits could be understood from this basic diagram as the equations for various stages remains the same.

Further, explanations can be done by an example illustrating simple 4-bit addition. In fig. 4, two 4-bit binary numbers i.e. '1011' and '1000' are added using a Kogge stone adder. From equations for each stage as stated earlier, the flow diagram could be prepared as of network fig. 3. In stage '0' carry input signals are generated using equation (1) and (2). Thereafter propagated carry and generated carry bits are executed using equation (3), (4), (5) and (6) in stage '1' and '2'. Finally, from a simple adder principle sum bits and carry bits are generated from equation (7) and (8). The main advantage of Kogge stone adder is its mechanism as all the stages are working in parallel. After all the calculations, proper sequencing of bits is done to get proper summation. As shown in fig. 4, after arranging sum bits from S_0 to $S_3$and then 'C_3'i.e. final carry sum is obtained. From fig. 4, the sum of '1011' and '1000' comes out to be '10011'. **[5]**



Now, $C_i = g_{i,} C_{in} = o$ → $C_0 = 0$, $C_1 = 0$, $C_2 = 0$, $C_3 = 1$

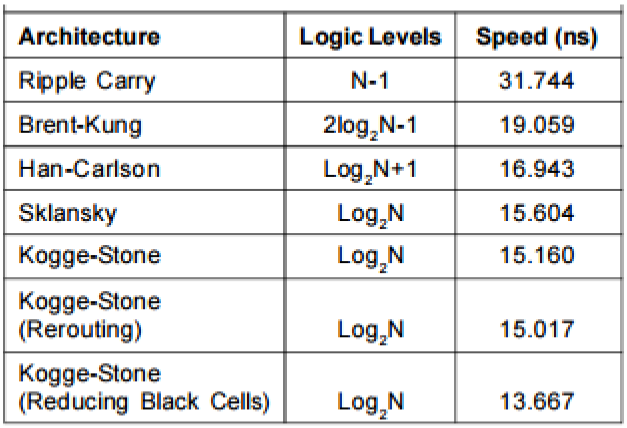
And, $S_i = C_{i-1}$ XOR P_{0i} → $S_0 = 1$, $S_1 = 1$, $S_2 = 0$, $S_3 = 0$

Fig.6. Addition of two 4-bit binary numbers KSA logic

X SUMMARY

On the basis of different paper on different types of adders comparison of delay, power and area for adders. For ripple carry adder(16 bit) delay is 3.875ns.carry look-ahead adder delay is 8.106.carry skip adder(16 bit) delay is 9.168ns.

Parallel prefix adder or Kogge stone adder delay is 6.700 ns. Sparse Kogge stone adder delay is 8.015 ns. Brent kung adder has 8.094ns delay.



Architecture	Logic Levels	Speed (ns)
Ripple Carry	N-1	31.744
Brent-Kung	2log$_2$N-1	19.059
Han-Carlson	Log$_2$N+1	16.943
Sklansky	Log$_2$N	15.604
Kogge-Stone	Log$_2$N	15.160
Kogge-Stone (Rerouting)	Log$_2$N	15.017
Kogge-Stone (Reducing Black Cells)	Log$_2$N	13.667

Fig.7. Comparison of Various Adder Architecture [2]

Above figure shows that Kogge stone adder has higher speed among parallel prefix adder family. Result show There is reduction in delay as well as logic levels and 9.84% faster than other adders.**[2].** and as the number of bits increase Kogge stone adder delay also decrease with respect to bit size.

XI. FUTURE SCOPE

In today’s era, addition is one of the most important and basic function required by electronics industry at a very large scale. Basic designs of Kogge Stone adder i.e. from 4 bit to 32 bit circuits are available to work on. But the demand of such reliable and efficient circuits is increasing in market for much bigger calculations. Now most of the systems that are being used i.e. from a computer or laptop microprocessor to a chip required in missile guiding system are 64 bit and the future systems shall surely be 128 bit and many firms are working on it. Kogge stone adder is approximately 12 times faster than a standard ripple carry adder. **[7]** Hence, working on 128 bit Kogge Stone Adder will surely help in designing systems with abilities and advantages of it as stated earlier.We propose a methodology to construct a 64-bit or higher (128-bit preferably) Kogge- Stone adder (KSA) that is typically used in multimedia applications in order to minimize its power-delay product. The goal of the design is to enable the designer to provide the best configuration to achieve specifications in terms of power consumption, delay, and area. The parallel prefix formulation of binary addition in Kogge stone adder is a very easy way to explain a family of parallel binary adders. The Proposed low power prefix structure of 64-bit or higher bits KSA makes it more suited for arithmetical units and multiplication units for complex ranges of input data computation.

REFERENCES

[1] " Performance analysis of different 8-bit full adders" *Volume 5, Issue 4, Ver. II (Jul - Aug. 2015),*

[2] "design and implementation of faster parallel prefix kogge stone adder" ISSN 2319 – 2518 www.ijeetc.com Vol. 3, No. 1, January 2014 © 2014 IJEETC

[3] "Design and estimation of delay, power and area for parallel prefix adder", RACES UIET Punjab university Chandigarh,06-08-2014, IEEE.

[4] Implementation of 64-Bit Kogge Stone Carry Select Adder with ZFC for Efficient Area, http://www.syslog.co.in/vlsi-projects/implementation-of-64-bit-kogge-stone-carry-select-adder-with-zfc-for-efficient-area.pdf

[5] "Fpga implementation of different adder architectures" (ISSN 2250-2459, Volume 2, Issue 8, August 2012

[6] "Area Efficient Carry Select Adder with Low Power" *SSRG International Journal of Electrical and Electronics Engineering (SSRG-IJEEE) – volume 2 Issue 2 Feb 201*

[7] "Design and Performance Analysis of Various Adders using Verilog" *IJCSMC, Vol. 2, Issue. 9, September 2013,*