International Journal of Advance Research in Science and Engineering
Volume No.06, Issue No. 10, October 2017
www.ijarse.com
IJARSE
ISSN: 2319-8354

# SURVEY OF VLSI ARCHITECTURE

**S.Mythili[1], S.Naveena[2], A.Nandini[3], V.Manikandan[4]**

*[1]AP/ECE, SNS College of Technology, Coimbatore (India)*

*[2,3,4]Student/ECE, SNS College of Technology, Coimbatore (India)*

## ABSTRACT

*Digital computation depends upon the efficiency of the rate of computations carried out. Addition is one of the most highly used computation process of any digital firm. Addition is a fundamental operation for any digital system, digital signal processing or control system.. A fast and accurate operation of a digital system is greatly influenced by the performance of the resident adders. These high speed addition computations are performed by highly efficient adders. In this paper we are going to discuss about various adders and their working principle.*

## I. INTRODUCTION

The basic building blocks of all arithmetic circuits are Adders: Adders add two binary numbers and give out carry and sum as output. Adders are commonly found in the critical path of many building blocks of microprocessors and digital signal processing chips. Not only for addition, but also for subtraction, multiplication, and division; Adders are very essential. Addition is one of the fundamental arithmetic operations. A fast and accurate operation of a digital system is greatly influenced by the performance of the resident adders. The most important for measuring the quality of adder designs in the past were propagation delay and area. Improvement in speed of adder indirectly improves speed of system

## II.PARALLEL PREFIX ADDER

Parallel-Prefix adders perform parallel addition i.e. most important in microprocessors, DSPs, mobile devices and other high speed applications. Parallel-Prefix adder reduces logic complexity and delay thereby enhancing performance with factors like area and power. The parallel prefix adders are the most efficient and fast adders till date. There are many types of parallel prefix adders that perform addition operation with different logics and every type uses prefix operation. The prefix operation is the logic of computing the output depending on the previous input. The different type of parallel prefix adders are namely, Brent Kung Adder, Kogge-Stone Adder, Ladner FischerAdder, SQRT CSLA, Han-Carlson Adder, CSA, Linear CSLA. Therefore the requisite element in the high speed arithmetic circuits is parallel prefix adders since twenty years. Parallel prefix computation carries out three necessary or vital steps:

1) Computation of carry generation & carry propagation signals by using no. of input bits.

2) Calculating all the carry signals in parallel that is called prefix computation.

3) Evaluating total sum of given inputs

## III.CARRY LOOK AHEAD ADDER

The carry propagation time is the main constraint of the ripple carry adder family. Other arithmetic operations such as multiplication are performed on the basis of addition and subtraction operation.

Thus by default the speed of the addition operation can be increased by increasing the speed of the adders, which can be achieved by reducing the carry propagation delay of the adders.

The carry look ahead adder employs the principle of carry look ahead which overcomes the issue by calculating the carry at the time of computation in each stage. The carry look ahead adder calculates the carry of each stage computation in advance based on the inputs of the corresponding stages. The propagate and the generate signals of each stage are calculated in advance and are compiled under two cases namely,

1. When both the A and B bits are 1

(or)

2. When one of the two bits is 1 and the carry signal of the previous stage is one.

The carry and propagate signals if the adder with the input signals A and B are given by,

$$Pi = Ai + Bi$$

$$Gi = Ai\ Bi$$

The output of the adder sum and carry signals are given by,

$$Si = Pi + Ci$$

$$Ci+1 = Gi + PiCi$$

The carry signals of various stage of the computation process of the adders are calculated from the expression,

$$C1 = G0 + P0C0$$

$$C2 = G1 + P1C1$$

$$= G1 + P1 (G0 + P0C0)$$

$$= G1 + P1G0 + P1P0C0$$

$$C3 = G2 + P2C2$$

$$= G2 + P2G1 + P2P1G0 + P2P1P0C0$$

$$C4 = G3 + P3C3$$

$$= G3 + P3G2 + P3P2G1 + P3P2P1G0 + P3P2P1P0C0$$

## IV.CARRY SKIP ADDER

The main constraint of computing the arithmetic functions of the adders and their delay intervention during computation is rather high.

To improve the delay intervention and the functionality of the adder chip carry skip adder principle can be used. The delay intervention of the adder chip can be improved depending upon the number of CSA used in coherence. The relation between the number of Carry Skip Adders used to the delay interference of the adder system is inversely proportional. The propagate signal of the Cary Skip Adder can be expressed as $Pi = Ai + Bi$

The main advantage of the Carry Skip Adder is to reduce the latency. The number of input AND gates is equal to the width of the adder. This relation is not applicable if the bandwidth of the adder is large and if additional delays are present in the system.

The performance characteristics of the Carry Skip Adder can be explained as,

- It is chained.
- It reduces the critical path.
- No real speed benefit.

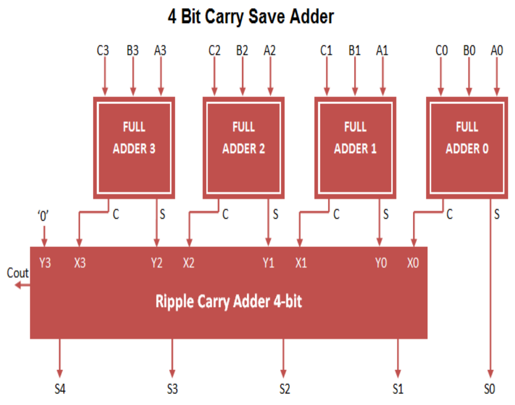
There are two types of Carry Skip Adder namely,

- Variable Carry Skip Adder
- Multi-level Carry Skip Adder

The structure of the carry skip adder consists of six full adders, Cin and Cout.

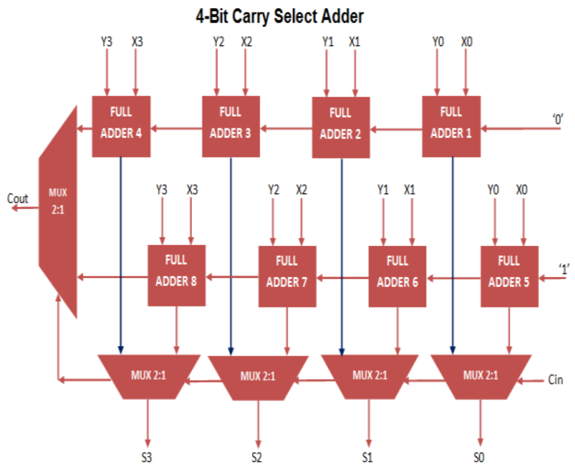
## V.CARRY SAVE ADDER

Another Algorithm for faster calculation of Addition is Carry Save Adder. It is also same as full Adder. From the two inputs we first produce two temporary Outputs named as Sum and Carry. For getting sum bit we first perform bitwise XOR and for the Carry bit we execute bitwise AND for the two input numbers, then at last add them by moving Carry bit left by unit place to Sum it up to produce final answer.



## IV.LINEAR CARRY SELECT ADDER

The main function of the Linear Carry Select Adder is that it computes (n+1) at two bit of numbers and the depth of the Linear Carry Select Adder is (0√n).

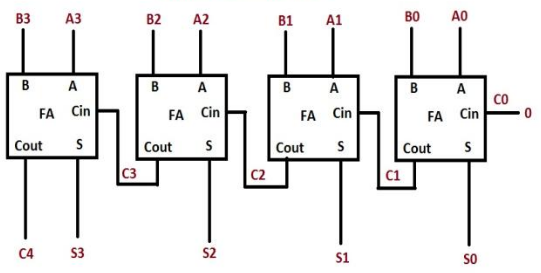


The linear carry select adder implies a particular way to implement the adder function. There are two results that can be calculated from the correct sum Cs which is taken and they are correct carry in, Cin and correct carry out Cout signals. So the delay the block size is calculated as (√n).

## VII.RIPPLE CARRY ADDER

A logic circuit in which carry-out of each full adder is the carry in of succeeding next most significant full adder is a ripple carry adder. In ripple carry adder, each carry bit gets rippled into next stage hence it is called a ripple carry adder. The sum and carry out bits of any half adder stage in a ripple carry adder is not valid until the carry in of that stage occurs.

A 4 BIT RIPPLE CARRY ADDER



The most widely used logic style is standard Complementary metal oxide semiconductor. It is a technology for constructing integrated circuits. CMOS circuitry dissipates less power than other logic families with resistive loads.

## VIII.KOGGE-STONE ADDER

The most popular carry look ahead adder amongst the high speed parallel prefix adders is the Kogge-Stone adder. It has the fastest adder design; the application of the fast computation is achieved at the cost of increased area. The Kogge-stone adder has three segments in which the addition operation is performed, they are

- Pre-Processing
- Carry look ahead network
- Post-Processing

**A) Pre-Processing**

The function of this stage is to generate the carry propagate signal Pi and the carry generate signal Gi from the input signal bits A and B. The carry propagate bit is obtained by performing the exclusive operation of the input bits Ai and Bi. The carry generate bit is obtained by performing the and operation of the input bits Ai and Bi. The above steps are illustrated as the equations given below,

Pi = Ai xor Bi

Gi = Ai and Bi

**B) Carry look ahead network**

The function of this stage is to compute the carry propagate and generate signal bits of the next stage based on the inputs of the previous stage. The generate and propagate bits namely, Gij, Pij, are computed by this stage

International Journal of Advance Research in Science and Engineering
Volume No.06, Issue No. 10, October 2017
www.ijarse.com
IJARSE
ISSN: 2319-8354

form the previous inputs (Gi, Pi) and (Gj, Pj). The bits Pi:j and Gi:j are also calculate by this stage. The sequence of equation required to calculate the above mentioned bits are given below,

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

Gi:j = Gi:k+1 or (Pi:k+1 and Gk:j)

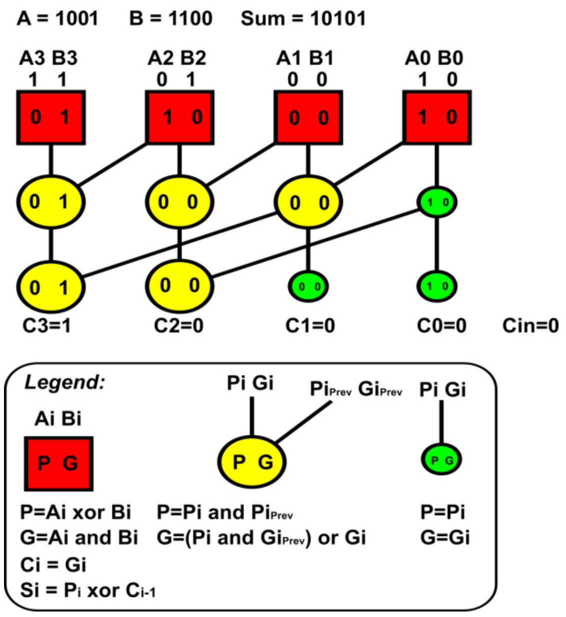
**C) Post-Processing**

The sum and carry bits of the entire addition operation is performed at this stage and this is responsible for the generation of the sum Si and Carry Ci bit of the addition operation, this can be achieved by computing the following equations,

Si = Pi xor Ci-16

Ci = Gi:0 or (Cin and Pi:0)

This is the stage where the overall addition operation is full filed by the adder after the generation of various bits based on various logics of the previous stages.



## IX. KUNG ADDER

The Brent Kung adder is a familiar type of the parallel prefix adder. Practically, the Brent kung adder has low fan-out from each prefix which in turn does reduce the delay significantly but the adder has long circuit path, which is ideally not suitable for high speed arithmetic computation. The Brent Kung adder proposes an optimized high speed added that addresses the problems of gate count, delay and gate connection in a way to reduce the gate area noticeably, in spite of having long critical paths.

The Brent kung adder is considered as one of the better high speed adders for minimizing the wiring tracks, fan-out and gate counts used as a basis for many other networks.

In parallel prefix adders, binary addition is usually expressed in terms of generate signal, propagate signal, carry signal and the sum signal at each bit position ($1 \leq i \leq n$), all the above listed signals can be calculated by the equations,

Gi =ai + bi

Pi = ai + bi

Ci = (gi, gi + pi, ci-1)

Si = pi + ci-1

The Brent Kung adder computed the sum in three stages namely,

- Pre-Processing
- Prefix Carry Tree
- Post- Processing

A) **Pre-Processing Stage**

The n bit parallel prefix adder operation begins with the Pre-Processing stage for the generation of Pi and Gi.

**B) Prefix Carry Tree Stage**

The carry bits signal are obtained from the signal from the first stage that will proceed with the next stage.

This stage contains three main complex logic categories namely; Black Cell, Grey Cell and the Buffer Cell. The black cell computes the Gi1j and Pj1i, whereas the grey cell can only execute the Gi1j. The prefix carry tree stage is a part that differentiates or determines the adder that is used.

Gi1j= Gi1k + Pi1, k x Gk-i1j

Pi1j= Pi1k x Pk-i1j

**C) Post-Processing Stage**

This stage is where the overall adder operation is completed, the carry signals of the second stage will pass through to the final stage (i.e.,) Post-Processing Stage. The final result of the entire adder operation is obtained from the equation.

## X. LADNER FISCHNER ADDER

Ladner Fischner adder is a form of parallel prefix adder. It also falls under the category of carry look ahead adder that can be represented as a parallel prefix graph containing the operator nodes. It is also the fastest adder with the focus on design time and the most common choice of the high speed performance adder amongst the industries;As the time required to generate the carry signals is calculated using the expression (log n). The better performance of the Ladner Fischner adder is because of its logic depth being minimum and bounded fan-out. The only downside to the Ladner Fischner adder is that it occupies a large silicon area.

The Ladner Fischner adders are more flexible and are used to speed up the binary additions and are obtained from Carry Look Ahead (CLA) structure. Tree structure form is used to increase the speed of arithmetic operation. The Ladener Fischner adder consists of black cells and grey cells. Each black cell consists of two AND gates and one OR gate. Multiplexer is a combinational circuit which consist of a single output obtained by multiplexing multiple inputs. Each grey cell consists of one AND gate only. The Pi denotes the propagate bit and it consists of only one AND gate. Gi denotes the generate bit and it consists of one AND gate and one OR gate. The computation equations for the propagate and the generate bits are given below,

Pi = Bi and Bi-1

Gi = Ai or (BI and Ai-1)

The generate bit can also be calculated by the equation,

Gi-2 = Ai-2 or (Bi-2 and Ai-1)

The addition operation in the Ladner Fischner Adder is carried out in three stages namely,

- Pre-Processing Stage
- Carry Generation Stage
- Post-Processing stage

**A) Pre-Processing Stage**

Just like any other parallel prefix adder the first stage is to compute the carry propagate Pi and the carry generate Gi. The equations for the computations of the propagate and the generate signals are,

Pi = Ai xor BI

GI = Ai and Bi

**B) Carry Generation Stage**

As the name suggests, this stage is for generation, of carry bits. The carry propagate and generate are generated at each cell, but the final cell present in each bit is responsible for generating the carry. The last bit carry of each stage will help to produce the sum of the next bit simultaneously until the computation of the final bit of each stage.

**C) Post- Processing Stage**

The final stage of the Ladner Fischner adder is effect as the carry of the first bit of every cell is xored with the next bit if the propagates, and then the output is given as the final sum.

Si = Pi and Ci-1

## XI. SQUARE ROOT CARRY SELECT ADDER

The construction of the adder is by equalizing the delay through the dual carry chains and multiplier block signal from the previous stage. The Square Root Carry Select Adder is also known as the Non-Linear Carry Select Adder. The SQRT CSLA uses Binary to Excess-1 Converter in the place of Ripple Carry Adder to achieve lower delay with a slightly increase in area.

In the case of a regular SRQT CLSA has 2 ripple carry adder with 2:1 multiplexer, it comparatively has a larger area due to multiple adder parts which is considered as the main disadvantage of each SQRT CSLA. The adder is divided into 5 groups containing different bit sized carry adders, the carry out are calculated from the last stage by the multiplier 5.

## XII. HAN-CARLSON ADDER

Adders are more commonly used in DSP lattice to serve the purpose of arithmetic calculation with the maximum efficiency. The efficiency of an adder depends on the parameters like the reliability, processing speed, gain and delay performance of the adder and chip size which is proportional to the functionality of the adder chip. The adders can be made more effective and efficient in the case of parallel prefix adders as the chip area size in minimized without the sacrifice of the functionality if the adder.The Han- Carlson adder is a combination of different stages of the Brent Kung Adder and the Kogge-Stone Adder.The processing of Han-Carlson adder is similar to that of Brent Kung and Kogge-Stone adder containing the same number of stages ; The process of stages are similar too. The main difference is that the adder offers trade-offs between the number of stages of computation of logic manipulation and number of logic gate;

Brent Kung uses minimal number of computational nodes, which yield the maximum depth. Kogge-Stone achieves high speed in the computation process and low fan out with the downside of having complex circuitry which requires more number of wiring tracks.

The Han-Carlson adder uses comparatively fewer number of prefix operations by making changes to the number of computational stages of the adder unlike the Kogge-Stone and the Brent Kung adder, it also reduces the area of the adder chip which provides the flexibility of the adder with greater functionality and lesser chip size.

G= A and B

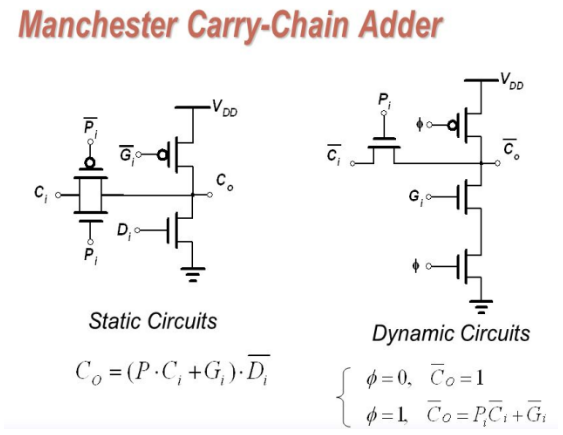
P= A xor B

The computation of the prefix operation is given by the expression,

(gi. pi)o(gj. pj) = (gi + pi . gj, pi . pj)

## XIII. MANCHESTER CARRY CHAIN ADDER

A variation of the carry-look ahead adder that uses shared logic to lower the transistor count is The Manchester carry chain; the logic for generating each carry contains all of the logic used to generate the previous carries.


Manchester Carry-Chain Adder
$V_{DD}$
$\overline{P_i}$
$\overline{G_i}$
$C_i$
$C_o$
$D_i$
$P_i$
Static Circuits
$C_O = (P \cdot C_i + G_i) \cdot \overline{D_i}$
$P_i$
$\phi$
$\overline{C_i}$
$\overline{C_o}$
$G_i$
Dynamic Circuits
$\phi = 0, \ \overline{C_O} = 1$
$\phi = 1, \ \overline{C_O} = P_i\overline{C_i} + \overline{G_i}$

The intermediate carries are generated by tapping off nodes in the gate that calculates the most significant carry value. Not all logic families has these internal nodes; however, CMOS being a major example. Dynamic logic can support shared logic, as can transmission gate logic. One of the major downsides of the Manchester carry chain is that the capacitive load of all of these outputs, together with the resistance of the transistors causes the propagation delay time, to increase much more quickly than a regular carry look ahead. A Manchester-carry-chain section generally won't exceed 4 bits.

## XIV.PARALLEL PREFIX LING ADDER

Ling adder was first proposed by Huey Ling. It is an adder which improves the speed of traditional parallel prefix signal calculation. A parallel prefix Ling adder can be considered as three bocks:

- The pre-processing block
- The propagation block
- The sum generation block

## XV. CONCLUSION

The primary purpose of the paper is to make a survey of all details and parameters of carry skip adder, carry select adder, ripple carry adder, Han Carlson adder, kogge stone adder, Manchester carry chain, linear carry select to identify the efficient adder by calculating the parameters like power gain and delay.

## REFERENCE

[1] N. H. E. Weste and D. Harris, CMOS Pearson–Addison-Wesley, 2011.

[2] Saradindu panda, A.Banerjee, B.Maj, DR.A.K.Makhopadhyay "international journal of advanced research" in electrical, electronic instrumentation engineering vol.1, issuse3, September 2012.

[3] P. M. Kogge and H. S. Stone, "A parallel algorithm for the efficient solution of a general class of recurrence equations," Computers, IEEE Transactions on , vol. C-22, no. 8, pp. 786–793, 1973.

[4] Sreenivaas Muthyala Sudhakar, Kumar P. Chidambaram and Earl E. Swartzlander Jr. "Hybrid Han-Carlson Adder "The University of Texas at Austin, 2012.

[5] M. Nesenbergs and V. O. Mowery, "Logic synthesis of high speed digital comparators," Bell System Technical Journal, vol.38, pp. 19–44, 1959.

[6] A. K. Verma, P. Brisk, and P. Ienne, "Variable LatencySpeculativeAddition: A New Paradigm for Arithmetic Circuit Design,"inProc.Design, Auto. Test Eur. (DATE '08), Mar. 2008, pp. 1250–1255

[7] DeepaYagain, Vijaya Krishna A and Akansha Baliga "Design of High-Speed Adders for Efficient Digital Design Blocks", 2012.

[8] D. Harris, "A Taxonomy of Parallel Prefix Networks,"inProc. 37thAsilomar Conf. Signals Systems andComputers, pp. 2213–7, 2003.

[9] Neil H.E. Weste, David Harris, Ayan Banerjee, "CMOS VLSI Design ," Third Edition.

[10] DeepaYagain, Vijaya Krishna A, " High Speed Digital Filter Design using register Minimization Timing & Parallel Prefix Adders.",2011

[11] R.P Brent and H.T.Kung ".A regular layout for parallel adders,"IEEE Trans.Computers, volume. C-31, no. 3, pp. 260.264, Mar. 1982.

[12] P. M. Kogge and H. S. Stone, "A parallel algorithm for the efficient solution of a general class of recurrence equations,"IEEE Trans. Computer., vol. C-22, no. 8, pp. 786–793, Aug. 1973.

[13] T. Han and D. Carlson, .Fast area-efficient VLSI Adders, In Proc. 8th Symp.Comp. Arithmetic, Sept. 1987, pp. 49.56.

[14] S. M. Nowick, "Design of a low-latency asynchronous adder using speculative completion," IEE Proc. Computer. Digit. Tech., vol.143, no.5, pp. 301–307, Sep. 1996.

[15] Behrooz Parhami, *"Computer Arithmetic"*, Oxford Press, 2000, pp131.

[16] Osorio, M., Sampaio, C., Reis, A., Ribas, R.: Enhanced 32-bit Carry look-ahead adder using multiple output enable-disable CMOS differential logic. In: Proceedings of 17th Symposium on Integrated Circuits and System Design, pp. 181–1

[17] Neil H. E Weste, Principles of CMOS VLSI Design --- A systems Perspective 2nd, Kamran Eshraghian, 1994.

[18] Jan M. Rabaey, Digital Integrated Circuits --- A Design Perspective, Prentice- Hall, 1996.

[19] John P. uyemura, CMOS Logic Circuit Design, Kluwer Academic Publishers,85 (2004)