A Detailed Scrutiny and Reasoning on VLSI Binary Adder Circuits and Architectures

K Mariya Priyadarshini, R. S. Ernest Ravindran, P. Ratna Bhaskar

Abstract: ***In this document a survey on recent developments in the design of binary adders is done. Adders are the core cells of any arithmetic unit which define the speed of any processor. The motivation of this paper is to focus on different kinds of architectures of higher order binary adders that provide high speed, less power to increase the level of integration on any integrated circuits (IC). Though there are many algorithms proposed for improving the speed of an adder the challenges still remain in designing fast and accurate adders. At the schematic level we scrutiny six different adders for high speed and low power applications.***

Key words: ***carry propagation delay, fast adder principles, carry selection, carry skip, prefix adders.***

I. INTRODUCTION

In communications and portable multimedia applications emerge, there is always a need for more prominent designs with low voltage, very thin size and high frequency of operation. Adders are the leaf cells in any DSP systems. Hence design of fast adders has become crucial as it affects the execution time of a digital system in variables of voltage and latency.

Every binary adder takes full adder as a basic cell which adds three single bits and its expressions of sum and carry are

SUM= A XOR B XOR C and
COUT= A.B + A.C+ B.C

The basic multi bit binary adder is the Ripple Carry Adder (RCA). RCA utilizes full adders for multi bit addition. The carry out after each full adder addition is sent to next satge. If for an n-bit RCA C_{out} need to be evaluated it has to wait until full adder addition for n-1 stages is performed. Accordingly, the last sum and carry bits will be legitimate after a significant deferral [1].To overcome the problem of carry propagation adders ike carry look ahead adder, carry increment adder, carry select adder, carry look ahead adders and parallel prefix adders are presented in this paper[2] .A carry-look ahead adder (CLA) generates the sum bit irrespective of carry input it receives from previous stages. Carry is generated and propagated using look-ahead logic. The following figure 1.1 shows a basic 4-bit CLA [3].From the figure we can see a look-ahead structure logic is used for generating carry. Due to this more number of MOS transistors are required to implement the circuit which increases area of the chip [3, 4]. Carry Look-ahead Adder (CLA) is a kind of optimized adder when compared with conventional RCA.

Revised Manuscript Received on May 07, 2019.

K Mariya Priyadarshini, PhD in Koneru Lakshmaiah Educational Foundation, Deemed to be University, Andhra Pradesh, India.

Dr. R.S. Ernest Ravindran PhD in 2016 from Anna University, Chennai, India.

P. Ratna Bhaskar, PhD in Koneru Lakshmaiah Educational Foundation, Deemed to be University, Andhra Pradesh, India.

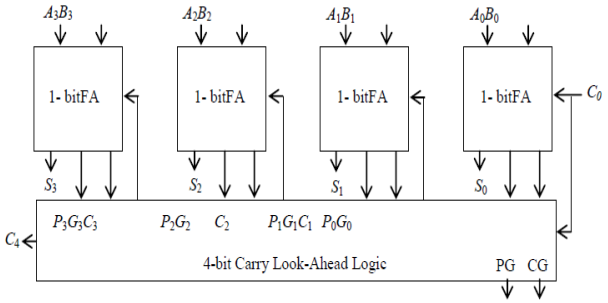


Fig 1.1 Block diagram of Carry Look Ahead Adder

Among the fast adders Carry Skip Adder (CSKA) is one option with reduced carry propagation delay. CSKA doesn't need any separate logic for generating and carry bits which reduces critical path length, but increases area and power dissipation same as RCA. Energy efficient product is very much low when compared to RCA and CLA [6, 7]. Layout for CSKA can be easily implemented with less wire lengths and regular structures. The slow compilation time of this adder structure to generate and propagate carry bits, limits its use in high frequency applications. However CSKA adder's performance degrades for higher order bits and in few combinations of bits, computational complexity equals RCA [8].

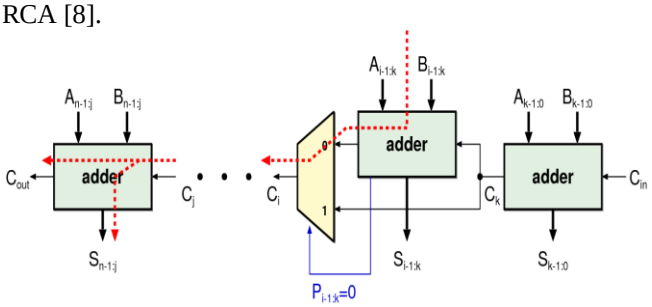


Fig 1.2 General architecture of Carry Skip Adder(CSKA)

$P_{i-1:k}$=0(not all propagate signals from bit i--1 to bit k are 1), the result of the carry is generated within this block. $P_{i-1:k}$=1,the carry of the previous block is propagated.

Carry Increment Adder (CIA) is one more efficient adder. By using clock phase techniques CIA increases the speed of carry propagation and sum generation. It also has smaller chip area compared with RCA, CLA and CSKA adder topologies but if the bit width is increased speed will be decreased and chip area.[9].

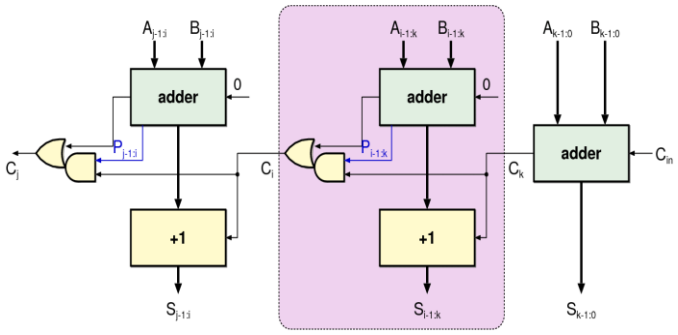


Fig 1.3Block diagram of Carry Increment Adder

In CIA adders c_{in} is assumed to be zero. In worst case when carry is known output is incremented.

One more high speed and low power adder used in digital signal processors and ALU is Carry Select Adder (CSA). CSA uses multiplexing method to generate final sum and carry out. The architecture of CSA consists of two simple blocks RCA and a multiplexer. But the usage of RCA in CSA degrades speed and increases power for higher order bits. The second stage of RCA with carry in '1' is replaced with binary to excess-1 code converter [11]. A carry-select adder generates sum bits for c_{in}='0' and c_{in}='1' at every stage. Based on the previous stage c_{in} multiplexers outputs the exact sum[12].

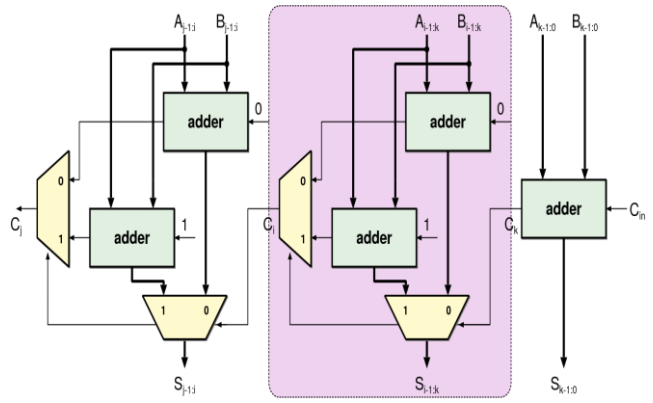
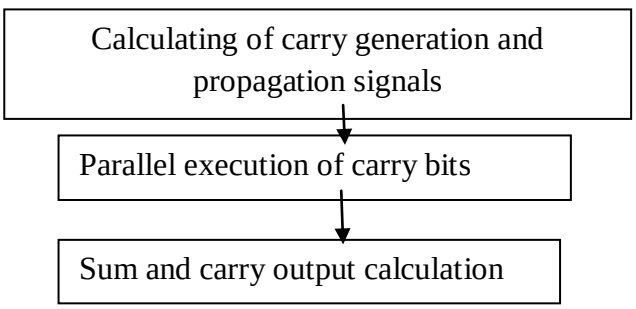


Fig 1.4Block diagram of Carry Select Adder

Parallel Prefix Adders (PPA) have an optimized carry tree structure with a systematic approach to generate sum and propagate carry bits [13]. The word Parallel means addition of two numbers is done parallel which is a dominant factor for high frequency microprocessor, DSP's and mobile communications. Delay and power overhead problems are far less when compared with above discussed adders. PPA have three steps of computation: 1) By using number of bits to be added carry is generated and propagated. 2)In prefix computation carry is computed parallel. 3)Total sum of given inputs is then calculated.



There are different kind of parallel prefix adders like PPA using RCA, PPA using sklansky, PPA using CIA, PPA using Brent kung Adder, Kogg stone and Huan Carlson.

The remaining sections of the paper consist of detailed qualitative analysis of above-mentioned binary adders. Section II describes architectures and functionality of all adders. Section III describes area power and delay comparisons and section IV and V consists of conclusion and references.

II. II BINARY ADDERS

1. Ripple Carry Adders (RCA)

RCA is the basic multi bit binary adder. Many fast RCA are proposed and some fast adders like radix-2 and higher order radix adders are discussed in this paper.

1.1 Ling Addition: in this technique [14] the c_{out} of $i+1^{th}$ stage of two binary words denote can be calculated as

$ci+1 = Gi:0 = gi + nki.gi-1 + nki.nki-1.gi-2 + \ldots$

$= nki.gi + nki.gi-1 + nki.nki-1.gi-2 + \ldots$

$= nki.(gi + gi-1 + nki-1.gi-2 + \ldots)$

$= nki.Hi:0$ (1)

Here nk stands for "not kill" and nki = ai + bi, where a,b are two binary words gi denotes "generate", defined as ai.bi, and Gi: 0 is group generate as a component of all the critical bits from '0' up to'1'. Hi:0, is obtained from Gi:0 by computing one nk bit at a time, exclusively known as a "Ling carry". Ling also proved that the sum bits, si+1, is a function of:

$$S_{i+1} = p(i+1) \oplus c(i+1) \quad (2)$$
$$= p(i+1) \oplus G(i:0)$$
$$= p(i+1) \oplus (nki.Hi:0)$$
$$= Hi:0.(p(i+1) \oplus nki) + (\ !Hi:0.pi+1)$$

Here p is propagate signal, defined as $pi = ai \oplus bi$. It is examined that when ling addition is implemented on parallel prefix adders it has led to a fan-out of 4 inverter delays for any word length of adder or its architecture by replacing two logic levels with complex gates like Exclusive-or gates or more.

1.2 Base-2 Ling Addition: Base-2 Ling adder [14] is implemented NAND gates with fan-in of 2 per full adder on the carry path by using Ling's method. The following expressions show how carry is propagated for two bit adder:

$C(i+1) = G(i:0)$

$= g_i + (\ nki.gi-1) + (\ nKi:i-1.Gi-2:0)$ (3)

The following equations are derived using lings factorials:

$Gi:0 = (\ nki.gi) + (nki.gi-1) + (nKi:i-1.nki-2.Gi-2:0)$

$= nki.(Hi:i-1 + nKi-1:i-2.Gi-2:0)$ (4)

The resultant Base-2 adder design is shown below, here H*x:c0 = nkx.Gx-1:c0 signifies carry appended not kill bit nkx. It is observed that carry input, c0, is Added with a0 and b0 in combination with the 2-input And Or Invert and 2-input Or And Invert logic cells generating !(Hi:i-1) and !(nKi+1:1) respectively. This optimizes critical path by one binary value.

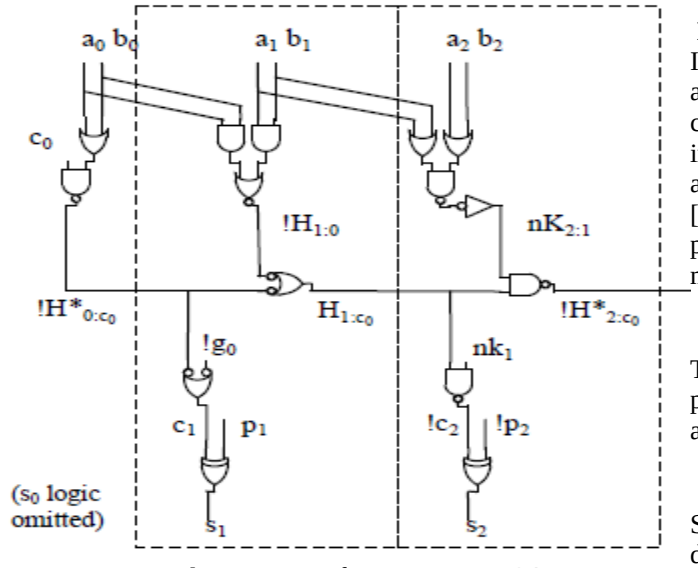


Fig 2.1 logic circuit of Base-2 Ling RCA

1.3 Base-4 Ling Adder: considering base-2 adder design as baseline base-4 adder is implemented, with AOI(fan-in-2 and fan-out-1)/OAI(fan-in-2 and fan-out-1) gates for each full adders along the critical path of carry out ;AOI22/OAI22 cells propagate carry to next stage. Additional gates are required to carry even numbered bits of sum which increases layout area. In order to reduce load on AOI/OAI gates half inverters are used to spin H (or !H) to the sum logic. The count of gates in c_n AOI21/OAI21 gates is determined by deciding the load capacitance and setting the load/drive proportion to 4, as previously:

[2.0(wire) + 0.5(inv) + 0.85·f]/0.55·f = 4 → f = 1.85.

The following figure illustrates a base-4 Ling Adder with ripple carry logic [14].

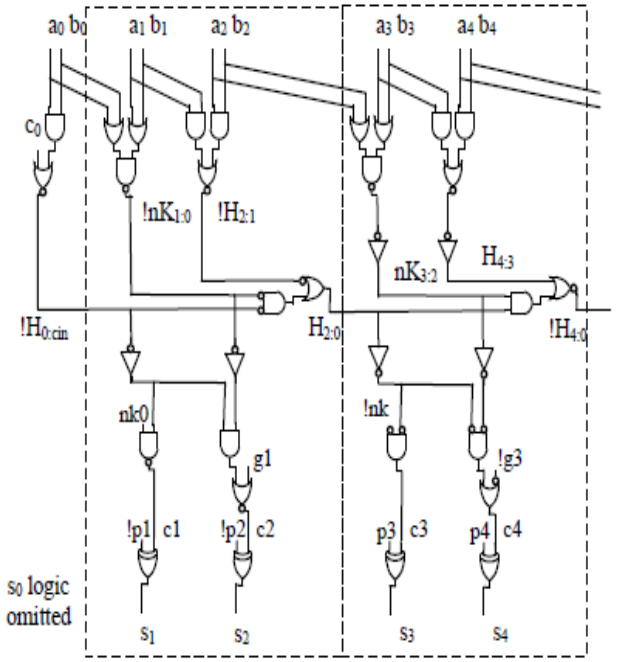


Fig 2.2 logic circuit of Base-4 Ling RCA

By cascading Base-4, base-8 and base-16 ripple adder can be constructed. At base-16 and above however increases carry propagation delay which is not feasible for high speed circuits.

Look Ahead Adder (CLA)

In n-bit RCA n-bits will have n no. of full adders. The addition is said to be computed only when (n-1) th stage carry is generated. As a result the delay of the carry chain increases which cannot be applicable for high frequency applications like DSP and Digital communications. CLA [14,15] calculates carry in two intermediate steps one is propagation (Pi) and two is generation (Gi) can be defined mathematically as follows.

$$P_i = A_i(Xor)B_i$$
$$G_i = A_i \,.\, B_i$$

The above equations uses only input bits to compute propagate and generate bits with just on gate delay. The sum and carry outputs would be composed below:

$$C_{i+1} = G_i + P_i\, C_i$$
$$S_i = C_i\, Xor\, A_i(Xor)B_i$$

So, C_{i+1} is a function of inputs and C_i. These conditions demonstrate that a carry value would be produced in two cases:

1) If bits A_i and B_i switch to '1' at same intervals.

2) If any one of A_i or B_i equals to '1' and carry-in C_i is '1'.

the basic cell a CLA adder is shown below for generating and propagating logic implementation. The truth table of a CLA adder for generating carry bit is shown in the table below.

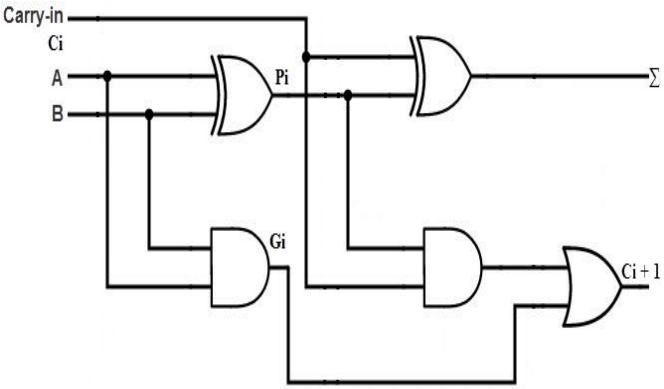


Fig 2.3logic circuit of CLA basic cell

Table 2.1 condition for generating carry in CLA

A	B	Ci	Ci+1	Condition
0	0	0	0	No Carry Generation
0	0	1	0	
0	1	1	1	
1	0	0	0	No Carry Propagate
1	0	1	1	
1	1	0	1	
1	1	1	1	Carry Generate
1	1	0	1	

2.1 Section-Carry Based Carry Look Ahead Generator Adder (SCBCLA)

The SCBCLA [15] is from CLA, where only one CLA output, sets carry input for the successive SCBCLA stages. Depending on how the carry is rippled within SCBCLA the sum bits are generated.

These kind of adders sums up output based on RCA technique and carries output defined by CLA architecture. The generic SCBCLA architecture is illustrated in figure shown below.

The SCBCLA architecture shown in the above figure 2.4 is divided into three blocks propagate-generate block, m-bit carry look ahead generator block and sum calculation block. There are two types of SBCLA one homogenous and the other hybrid or. Moreover, heterogeneous SCBCLAs produce streamlined plan measurements contrasted with homogeneous SCBCLAs. For the basic version, the carry propagation critical path delay equals gate delays of a XOR gate with fan-in 2, AND gate of fan-in 5 and OR gate of fan-in 2 as seen in Fig.2.5.

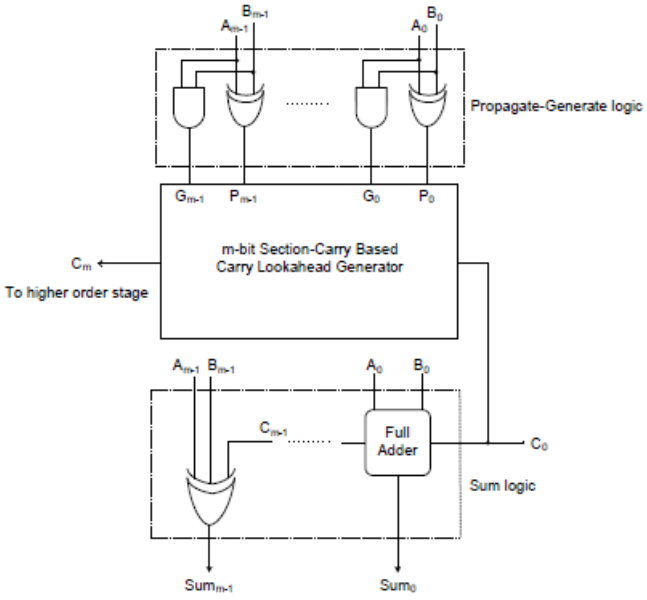


Fig 2.4 Micro Architecture of m-bit SBCLA

The optimized SBCLA is shown in Fig2.6 where the critical path delay equals to gate delays of XOR gate with fan-in 2 ,AND gate with fan-in 4,OR gate with fan-in 4 and fan-in 2. In figure 2.6 Section Carry Based Carry Look Ahead Adder with fan-in 4 is constructed using 5-input AND gate and OR gates, which are not bolstered in present day CMOS process advancements [16, 17].In this manner a superior substitute is to opt for the fragmented 4-bit SCBACLA.

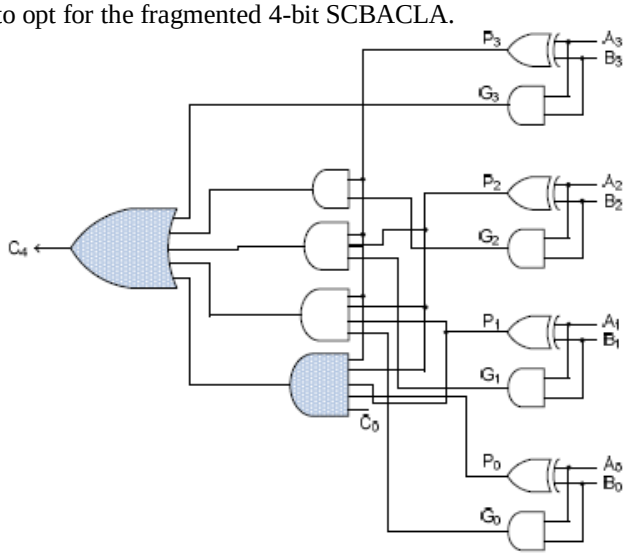


Fig 2.5 logic circuit for 4 bit carry generator of CLA

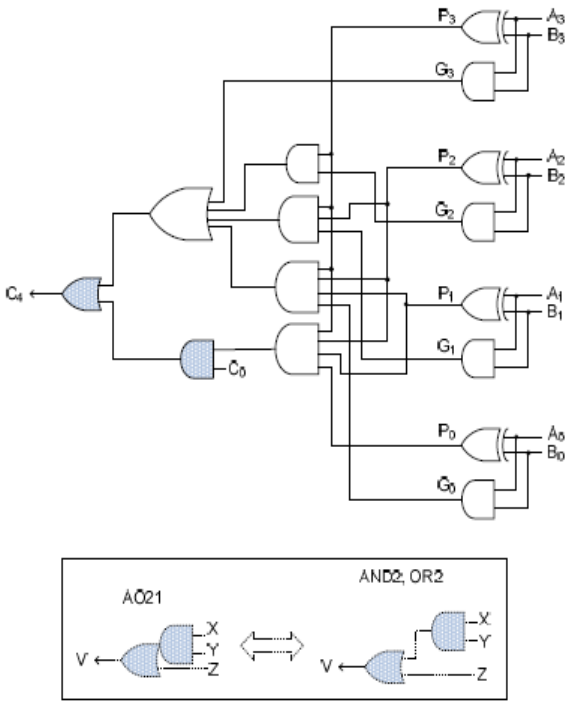


Fig 2.6logic circuit for 4 bit carry generator of SCB CLA

3. Carry Skip Adder

CSKA is one of the energy saving adders in area occupancy and power usage. When compared to RCA CSKA will have shorter critical path. The energy efficiency product of CSKA is too less when compared with CSA and parallel prefix adders. CSKA provides relatively less wiring length along with well ordered and uncomplicated layout. The conventional CSKA is constructed with chain of full adders and multiplexers. A 2x1 multiplexer interconnects the RCA blocks. The logic diagram of conventional CSKA is shown below.

As the number of RCA chains increases the design of static CMOS CSKA becomes complex, which increases the parasitic effects of circuit. Many methods have been proposed to revamp the execution of CSKA at critical voltages one among it is adaptive clock stretching operation [18] to [20]. By using this method the critical paths in the CSKA adders are rarely activated. When data paths with large gate delay in the adder carry the signal are on, the structure makes use of two clock cycles to finish the operation. In this method we observe a trade-off between power consumption and throughput. Two more techniques are proposed in [21] one is CI-CSKA (Carry Increment CSKA) the structure depends on link and implicationplans.

The 2x1 multiplexer in conventional CSKA are replaced by AOI/OAI compound gates because power consumption of AOI (OAI) is less when compared with multiplexers [22] and the logic circuit is shown below.

The adder is constructed with 2- *N* bits inputs, *A* and *B*, consisting of *Q* stages. *Mj (j* = 1, . . . , *Q)* is the size of each RCA block, with Carry input zero except the first block.). Therefore, all the blocks complete their jobs concurrently. In these kind of adders, when the first slab calculates the sum and carry bits (i.e., *SM*1, . . . , *S*1), and *C*1, the remaining blocks

parallely generates the middle of the path results [i.e., {*ZK j+Mj , . . . , ZK j*+2, *ZK j*+1} for *K j* =_*j*−1 *r*=1 *Mr (j* = 2, . . . , *Q)*], and also *Cj* signals.

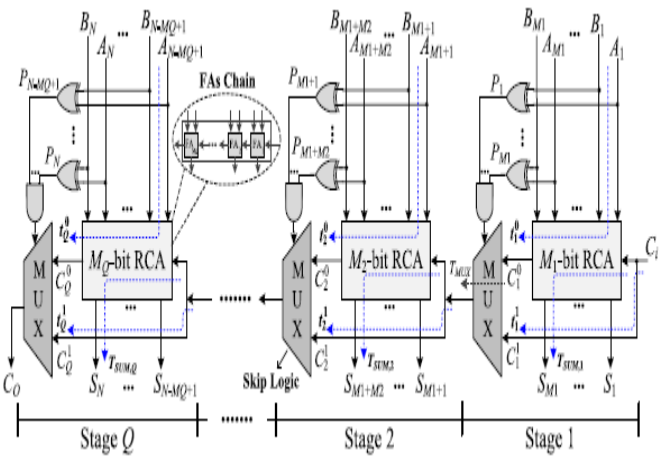


Fig 2.7 Logic circuit of Conventional CSKA

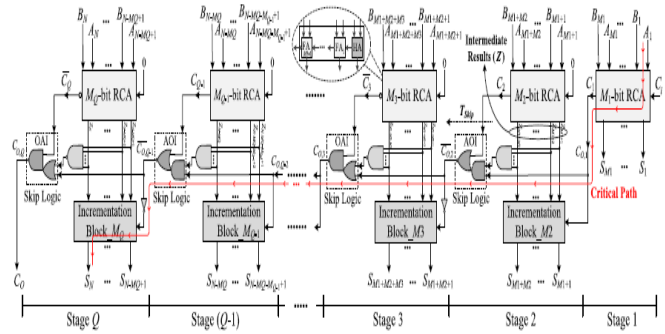


Fig 2.8 Logic circuit of Carry Increment CSKA

The internal structure of carry increment block is shown below.

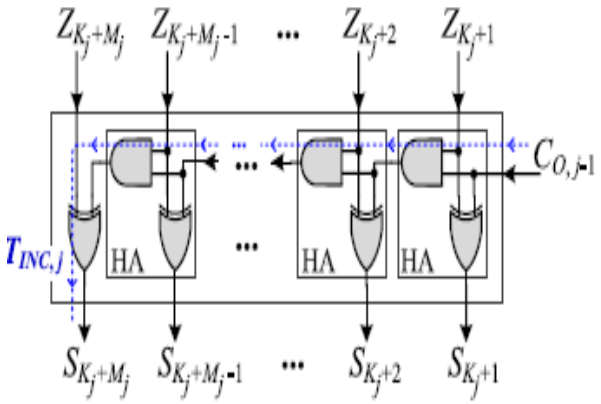


Fig 2.9 Carry Incremental Block internal structure

Variable delay length CI CSKA is proposed to improve logic delay [23]. In order to compute sum and carry bits in just one clock period variable supply voltage are inbuilt. Hence clock frequency need not be decreased as longest delay paths are occasionally triggered. The following figure explains the concept of adaptive variable delay clock stretching and also supplies voltage scaling. In Fig.2.10, the input bits $j + 1)^{th}$ and$(j + m)^{th}$ are used for estimating the carry propagation of the j^{th} stage (FA) to the final carry output of $(j + m)^{th}$ stage. In this design, the Longest Latency Chain(LLC) is from stage-1 to stage-N. While the carry proliferation way from first stage to the (j+m)th organize and the convey spread way from (j +1)th stage to the Nth stage are the off length longest ways [23].The slack time in this structure determines voltage scaling parameters, which is characterized by the delay contrast between LLC. As the critical path power ups with a probability of<1/2*m*, the clock stretching has no affect upon efficiency (e.g., for a 64-bit adder, *m* = 16–20 may be taken into account) [23]. But in some cases the predictor may not predict when the critical path is exactly driven. But as m increases mis predictions decrease but increases the critical path which restricts threshold voltage reduction. Keeping in consideration of all these tradeoffs predictor radix size need to be fixed.

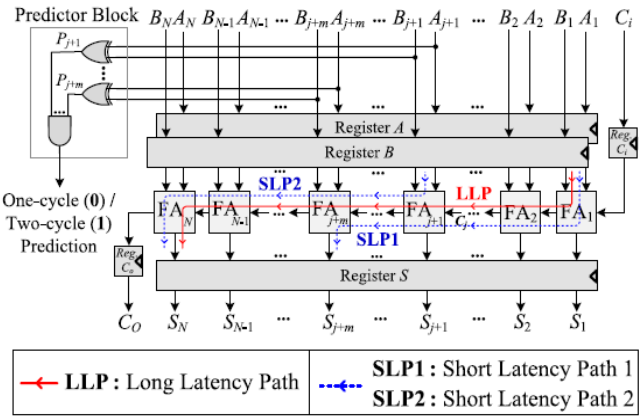


Fig 2.10 Generic structure of variable latency CSKA

4. Carry Select Adder (CSA)

The structure of Carry Select Adder is built using dual RCA which increases power and delay of the logic circuit. [24,25] Low power and high-speed CSA can be executed by replacing RCA with CLA, CSKA and CIA. To further improve the performance second ripple carry adders with Carry input 1 can be replaced with Binary to Excess one converter (BEC) as shown below. The following topology shows the basic CSA using RCA.

An example of 16-bit CSA using RCA is explained below

```
1010 1010 1011 1111
0000 0011 1111 1010 (Cin=0)
(o) 1010 1110 1011 1001

1010 1010 1011
0000 0011 1111 (Cin =1)
(o) 1010 1110 1011
```

Fig 2.11(a) Example for CSA

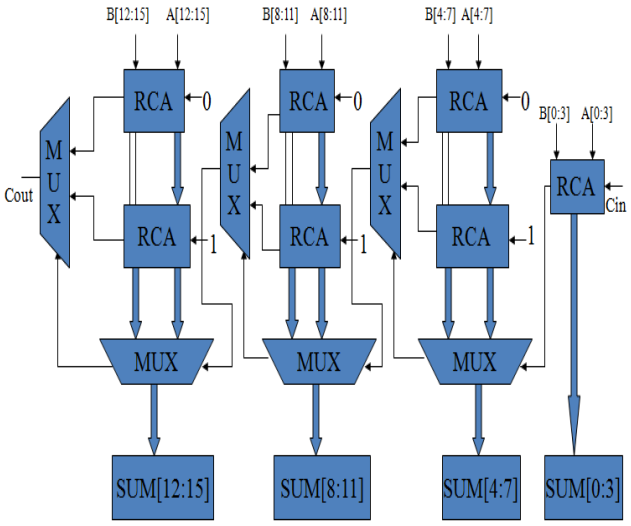


Fig 2.11(b) logic diagram of CSA using RCA

4.1 Modified CSA with BEC

when two RCA are used in CSA delay doubles at every carry propagation. So as to cut down the carry propagation delay RCA with carry input 1 is replaced with BEC [26]. The input to the BEC is SUM output of corresponding RCA of that stage. Bit one is added to LSB bit of sum bits generated when carry input is zero. BEC is implemented using Exclusive OR gates.

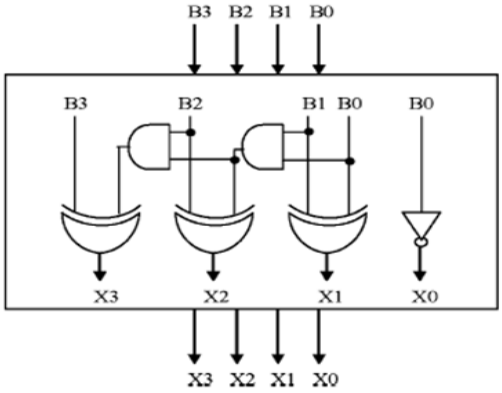


Fig 2.12 Logic circuit for BEC converter

4.2 Modified CSA with CLA, CSKA

In Fig 2.11 we observe that conventional CSA [27] reduces carry propagation delay to a major extent by breaking 16-bit carry propagation to 4 different 4-bit RCA connected via 2x1 multiplexer. But the performance can be further improved by replacing RCA with carry input zero by CLA and RCA with carry input 1 by BEC. The topology of modified CSA is shown in the figure below.

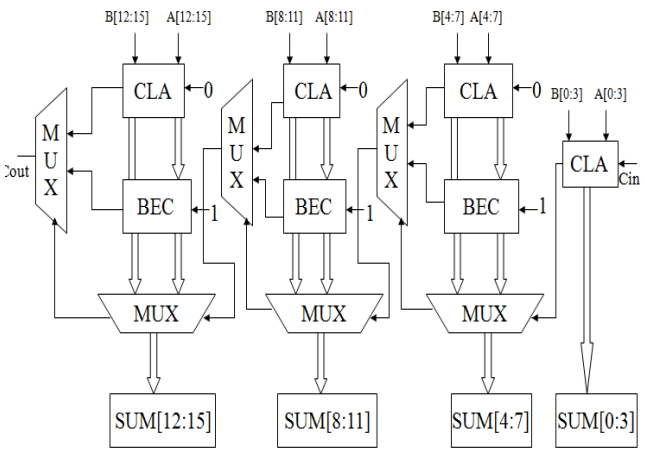


Fig 2.13 Modified CSA with CLA

From the above discussions as we have already discussed the carry generate and propagate signals reduce the carry propagation delay unlike in RCA. Similarly, in place of CLA we can replace CSKA which further reduces area and power consumption along with delay [28,29]. Thus, Carry Select Adders have more flexibility to modify the topology depending upon consumer's choice.

5. Parallel Prefix Adders(PPA)

A variety of proposed prefix adders are discussed in the following passage which are area and power efficient. Contingent aggregate expansion for prefix 94 adder suggested by Sklansky with less complexity but fan-out increases for few computations. The innovation proposed by Kogge and Stone has both excellent logical intensity with fewer outputs but constructed with huge number of wires to connect gates. Brent and Kung proposed an adder which has less computation nodes but with paramount prefix extent which increases intermission. Ladner and Fischer proposed a affix structure work with more profundity when contrasted with sklansky adder. This technique decreases the most extreme fan-out for calculation hubs in the basic way. The prefix adder proposed by Han and Carlson couples Brent-Kung and Kogge-Stone adder's which overcomes trade-off between depth of logic, wiring count and number of intermediate nodes. Reto Zimmermann proposed a optimized adder using depth controlled compression and expansion.

Matthew Ziegler and Stan proposed prefix adder archtecture with a maximum two outputs for reducing the yield factor. Knowles displayed a class of logarithmic adders with least profundity by permitting the fan-out to develop. Algorithm for generating prefix carry trees proposed by A.B Smith and C.C Lim uses higher valency prefix cells in the initial stages of the architecture to accommodate less number of prefix cells in the critical path and to achieve less interconnect length. An algorithmic methodology to devolep irregular PPA proposed by Jianhua Liu et al achieves nominal delay for a given set of input signals. The use of higher valency prefix cells for standard prefix architectures such as Brent-Kung, Sklansky, Ladner Fischer, Kogge-Stone and Han-Carlson was proposed by Harris. This leads to reduced number of rationale levels to the detriment of more noteworthy fan-in at each level.

A zero deficient PPA with minimal depth for a given width was proposed by Haikun Zhu et al. PPA takes place in three steps as explained below

1. Generating and Propagating Carry signals: Past carry is determined and engendered to the following piece and create thecarry bit below are the mathematically defined equations:

$Gi = Ai \cdot Bi$ ……..(1)

$Pi = Ai \oplus Bi$ …… (2)

2. Computing of all carry signals:

$Gi:j = Gi:k + Pi:k \cdot Gk-1:j$ …….(3)

$Pi:j = Pi:k \cdot Pk-1:j$ …… (4)

3. Evaluation of Final Sum:

$Si = Pi \oplus Gi-1:0$ …… (5)

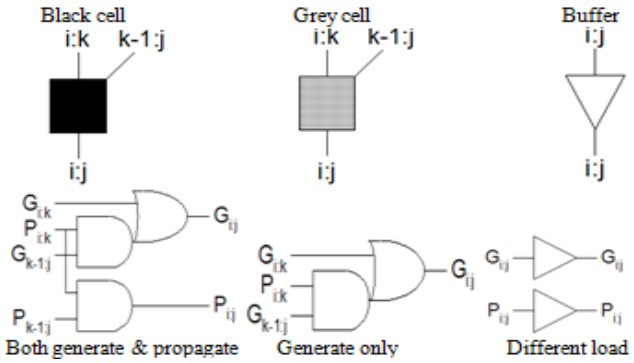


Fig 2.14 schematic representation of generate and propagate signals of PPA

5.1 Kogge-Stone(K.S.) adder

PPA using Kogge-Stone adder is suggested by P.M. Kogge and H.S.Stone[31]. When compared with Brent–Kung adder the Kogge–Stone adder occupies more area but shows low fan-out which can increase the performance of CMOS .nonetheless, interconnection has reached the bottle neck and is regularly an issue for Kogge–Stone adders. Kogge-Stone adder has a disadvantage of wiring congestion when expanded to 2^5-bit, 2^6-bit, and 2^7 -bit adders. The main motto is to reduce longest computational paths to considerable length of bits. In fig.2.6 all vertical blocks are used for propagation and generation of carry bits. Sum bits are generated by performing XOR operation in between the last stage generate bits and first stage propagate bits.

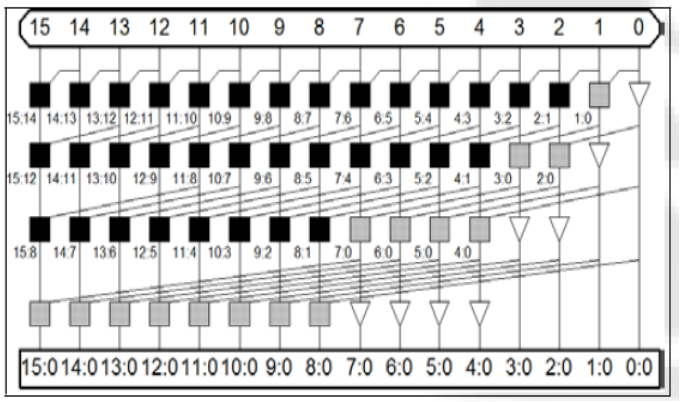


Fig 2.16 16-bit Kogge-Stone PPA

5.2 Brent Kung(B.K.) Adder

Brent –Kung PPA was introduced by Brent . Brent-Kung adders has maximum logic depth but has less interconnects and minimal area. Generate sum for odd number bits and then for even bit positions.[33]. It calculates prefix for 4-bit groups and used for calculating 8-bit group prefix, and in turn used for a group of 16-bits and higher order words. The affixes then track backwards to calculate the carry-in bits at every stage. The tree like structured adders require *2log2n-1* stages. In general, black and grey cells are buffers used to reduce fan-out. Using CMOS logic and Transmission gates this adder can be developed for 2^3, 2^4, and 2^5-bit using CMOS logic [36]. The following figure displays the structure of a 2^4-bit Brent-Kung PPA.

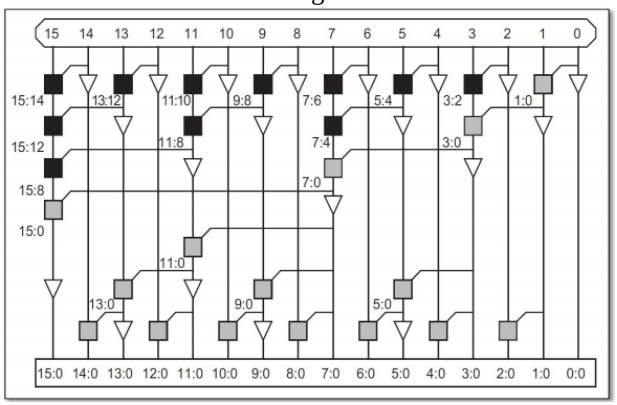


Fig 2.16 16-bit Brent-Kung PPA

5.3 Han-Carlson(H.C) Adder

The performance of *Han-Carlson* trees lies within Kogge-Stone and Brent-Kung [34]. H.C adder has sparse version of K.S adder. These Adders follow a novel technique in calculating sum and carry bits, first it will fuse carry bits on divide by 2 bits and works on complement of divide by 2 bits. The true carry bits are generated by combining odd bits with even carry bit signals. Out of the five stages of Carlson adder 2^{nd} 3^{rd} and 4^{th} stages are replica of Kogge-Stone structure. When compared to kogge-stone adders these adders show less logic complexity as it uses short length wires which in turn reduces logic multiplicity at a cost of added stages for carry-merge path. Han-Carlson adder can be easily constructed by modifying pseudo-code of Kogg-Stone adder [35]. The construction of 16-bit Han-Carlson PPA is exhibited below.

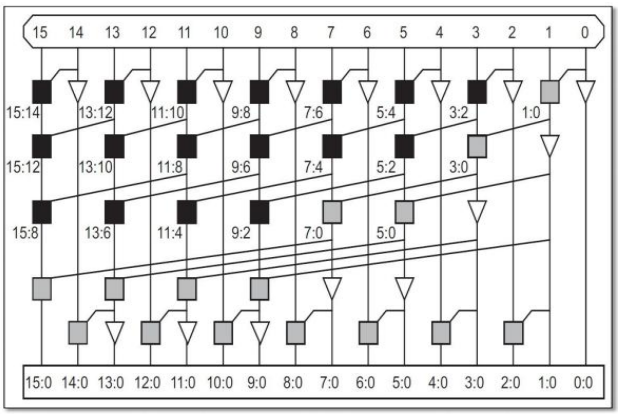


Fig 2.17 16-bit Han-Carlson PPA

5.4 S. Knowels(S.K.) adder

A family of prefix trees with malleable architectures are proposed by S.Knowles[37]. Knowles adders are a combination of. S.K. adder utilizes the fan-out at every logic level.

Fig.2.17 displays a 2-word Knowles adder. S.K. prefix adders can be designed to be hybrid by facilitating distinctive fan-out in Similar rational level.

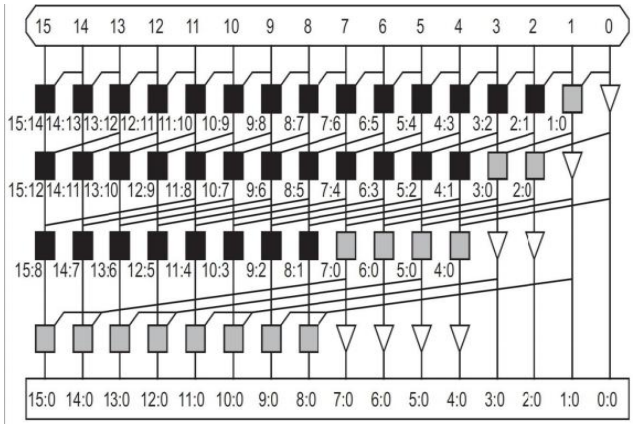


Fig 2.18 16-bit S.Knowels PPA

5.5 Parallel Prefix Adder- Ladner- Fischer

R. Ladner and M. Fischer in 1980 R. Ladner and M. Fischer a PPA. L.Fischer prefix tree construction lies between Brent.K and Sklansky prefix tree. The LF adder [38] has least rationale profundity however it has expansive fan-out. Ladner-Fischer adder has conveyed administrator hubs. The entry way retardation for L.Fischer prefix tree is log2n+1. Fig.2.18 demonstrates the 2-word LF adder

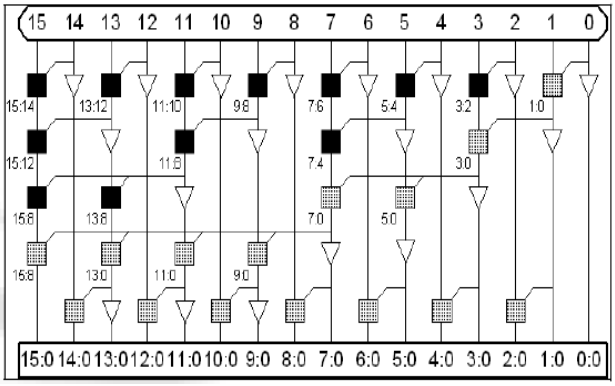


Fig 2.18 Ladner- Fischer parallel prefix adder

III. PERFORMANCE COMPARISON OF ADDERS

Kogge-Stone, Han-Carlson and Knowles adders requires big count of parallel interconnects for lengthy bit adders, which increases the coupling capacitance on each wire. Because, of high fanout Sklansky architecture becomes slow. When interconnect is considered Han-Carlson its topology. even though *Kogge-Stone* occupy small area but tough to Propagate and Generate carry and sum signals. *Brent-Kung* is of course a first adder but is very bad at propagation of carry signal. L.Fischer structure there are slightly added rationale levels and increase in number of outputs. H.Carlson got huge rationale levels however less cells. S. Knowles has various cells and wires and fewer outputs. Sklansky has least rationale levels and most elevated fan-out. Kogge-Stone snake was the best among the others when wiring capacitance is disregarded. Correlations are organized in the table3.1 between adders that swell convey and that figure parallel talked about so far .

Table: 3.1 Performance comparison of PPA

Adder	Area	Speed	FO	≈A.	≈T.	$\approx FO_{max}$.
RCA	Small	Slow	Min	n-1	n-1	2
SK	Large	Fast	high	(n/2)*logn	Logn	(n/2)
BK	Med	Med	med	2n-logn	2logn-2	Logn
KS	Huge	Fast	min	nlogn	Logn	2
HC	Large	Fast	min	(n/2)logn	Logn+1	2
CIA	Med	Med	med	2n-2	2n	2n

IV. COCLUSION

We have surveyed different architectures of adder circuits, which focuses more on high capacity Parallel Prefix Adders. Our discussion is inclined on how the topologies are modified making them applicable for high speed low power applications. At first we discussed about RCA ling adders performance and how can a RCA be used for DSP applications. CIA, CLA, CSA, CSKA comes under the family of high-speed binary adders where we observe a tradeoff between area power and delay. In order to enhance the speed performance of adders for high frequency applications Parallel Prefix Adders were discussed, and it is observed that Kogge-Stone PPA shows good results. we have reviewed all the architecture levels of binary adders.

REFERENCES

1. Nimisha A Deshmukh1, Prof. Mrs. Arachana Khandait "A Review Paper on Comparative Evaluation of Carry Select Adder and Modified Carry Select Adder" International Journal of Emerging Trends in Engineering and Basic Sciences (IJEEBS) *Volume 3, Issue 2 (March-April 2016), PP.42-49.*
2. Srdjan Capkun, Frank K. Gürkaynak "Combinational Circuits: Theory" Adapted from Digital Design and Computer Architecture, David Money Harris & Sarah L. Harris ©2007 Elsevier
3. Amita1, Mrs. Nitin Sachdeva2 "Design and analysis of Carry Look Ahead Adder using CMOS technique" OSR Journal of Electronics and Communication Engineering (IOSR-JECE) Volume 9, Issue 2, Ver. VII (Mar - Apr. 2014).
4. A. Prabhu, K. Pradeepagiri, R. Ranjitha Implementation of multiple bit carry look ahead adder using verilog platform, International Journal of Advanced Information Science and Technology, Vol.3, No.7, July 2014.
5. Poornima Shrivastava , Prof. Balram Yadav and Prof Bharti Chourasia "Survey on Carry Look Ahead Adder" International Journal of Electrical, Electronics
6. Ubaidulla, Prof. Asia Hazareena, Implementation of High Speed and Energy Efficient Carry Skip Adder using Verilog HDL, International Journal of Science Technology & Engineering, Volume 3, Issue 02 August 2016.
7. Milad Bahadori, Mehdi Kamal, Ali Afzali-Kusha, Senior Member, IEEE, and Massoud Pedram, Fellow, IEEE "High-Speed and Energy-Efficient Carry Skip Adder Operating Under a Wide Range of Supply Voltage Levels", Ieee Transactions On Very Large Scale Integration (VLSI) Systems.
8. T.Sowmiya, S.Jeya Anusuya "High- Performance And Energy efficient CSKA Operating Under A Wide Range Of Supply Voltages"IJARIIE- Vol-3 Issue-3 2017.
9. A.Sai Ramya , Mounica ACN , BSSV Ramesh babu "Performance analysis of different 8-bit full adders" IOSR Journal of VLSI and Signal Processing (IOSR-JVSP) Volume 5, Issue 4, Ver. II (Jul - Aug. 2015),

10. K.Mariya Priyadarshini, N.V.N. Ravi Kiran, N. Tejasri, T.C. Venkat Anish "Design Of Area And Speed Efficient Square Root Carry Select Adder Using Fast Adders", International Journal Of Scientific & Technology Research Volume 3, Issue 6, June 2014.
11. Pappu P. Potdukhe, Vishal D. Jaiswal "Review Of Carry Select Adder By Using Brent Kung Adder", International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE), Volume 4, Issue 10, October 2015.
12. Geeta Rani, Sachin Kumar, "Delay Analysis of Parallel-Prefix Adders" International Journal of Science and Research (IJSR), Volume 3 Issue 6, June 2014.
13. Neil Burgess ARM Inc. Austin, TX, USA "Fast Ripple-Carry Adders in Standard-Cell CMOS VLSI" 2011 20th IEEE Symposium on Computer Arithmetic.
14. H. Ling, "High-Speed Binary Adder", IBM J. Research & Dev., vol 25, p.156-166 (May 1981).
15. P.Balasubramanian,N.E.Mastorakis "Design of Synchronous Section-Carry Based Carry Lookahead Adders with Improved Figure of Merit" WSEAS TRANSACTIONS on CIRCUITS and SYSTEMS,Volume15, 2016.
16. Synopsys Digital Standard Cell Library SAED_EDK32/28_COREDatabook, revision 1.0.0, 2012.
17. G. Yang, S.-O. Jung, K.-H. Baek, S.H. Kim, S. Kim, S.-M. Kang, "A 32-bit carry lookahead adder using dual-path all-N logic," IEEE Trans. VLSI Systems, vol. 13, no. 8, pp. 992-996, 2005.
18. H. Suzuki, W. Jeong, and K. Roy, "Low power adder with adaptive supply voltage," in *Proc. 21st Int. Conf. Comput. Design*, Oct. 2003,pp. 103–106.
19. H. Suzuki, W. Jeong, and K. Roy, "Low-power carry-select adder using adaptive supply voltage based on input vector patterns," in Proc. *Int. Symp. Low Power Electron. Design (ISLPED)*, Aug. 2004.
20. Y. Chen, H. Li, K. Roy, and C.-K. Koh, "Cascaded carry-select adder (C2SA): A new structure for low-power CSA design," in *Proc. Int. Symp.Low Power Electron. Design (ISLPED)*, Aug. 2005.
21. Milad Bahadori, Mehdi Kamal, Ali Afzali-Kusha, *Senior Member, IEEE*, and Massoud Pedram, *Fellow, IEEE* "High-Speed and Energy-Efficient Carry Skip Adder Operating Under a Wide Range of Supply Voltage Levels", IEEE Transactions On Very Large Scale Integration (Vlsi) Systems, Vol. 24, No. 2, February 2016.
22. J. M. Rabaey, A. Chandrakasa, and B. Nikolic, *Digital Integrated Circuits: A Design Perspective*, 2nd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 2003.
23. S. Ghosh, D. Mohapatra, G. Karakonstantis, and K. Roy, "Voltage scalable high-speed robust hybrid arithmetic units using adaptive clocking," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 18, no. 9.
24. Kuldeep rawat,Tarek Darwish, and Magdy Bayoumi, "A low power and reduced area Carry Select Adder", 45th Midwest Symposium and circuits and systems, vol.1,March 2002.
25. B.Ramkumar, Harish M Kittur and P. Mahesh Kannan, "ASIC implementation of Modified Faster Carry Save Adder", European journal of scientific research, vol.42,2010.
26. M.Moris Mano, "Digital Design", Pearson Education, 3rd edition 2002.
27. Singh, R.P.P.; Kumar, P.; Singh, B., "Performance Analysis of Fast Adders Using VHDL", Advances in Recent Technologies in Communication and Computing, 2009.
28. Hemima.R, Chrisjin Gnana Suji.C "Design Of 4 Bit Low Power Carry Select Adder" Proceedings of 2011 International Conference on Signal Processing, Communication, Computing and Networking Technologies (ICSCCN 2011).
29. Basant Kumar Mohanty and Sujit Kumar Patel "Area–Delay–Power Efficient Carry-Select Adder" IEEE transaction on circuits and systems-II: Express briefs, Vol. No. 6,June 2014.
30. 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.
31. B Pullarao, J.Parveen Kumar, "Design of High Speed Based On Parallel Prefix Adders Using In FPGA", International Journal of Engineering Sciences & Research Technology (IJESRT) Dec, 2013
32. Deepa Yagain, Vijaya Krishna A, Akansha Baliga,"Design of High-Speed Adders for Efficient DigitalDesign Blocks", International Scholarly ResearchNetwork ISRN electronics, July2012.
33. M. Mahaboob Basha, Dr. K. Venkata Ramanaiah, Dr. P. Ramana Reddy, B. Lokeshwar Reddy, " An efficient model for design of 64-bit High Speed Parallel Prefix VLSI adder", International Journal of Modern Engineering Research (IJMER) Vol. 3, Issue 5, Sep-Oct 2013.
34. Han, Carlson, "Fast Area-Efficient VLSI Adders, IEEE" 1987.
35. Vishal R. Naik, Sonia Kuwelkar," DESIGN OF A CARRY TREE ADDER", nternational Journal of Pure and Applied Research in Engineering and Technology (IJPRET) Vol. 2(9), 413-424, 2014.
36. Noel Daniel Gundi "Implementation Of 32 Bit Brent Kung Adder Using Complementary Pass Transistor Logic". Visvesvaraya Technological University, Belgaum, Karnataka, India June 2008.
37. S. Knowles, "A family of adders", in Proc. 15th IEEE Symp. Comp. Arith., June 2001.
38. R. Ladner and M. Fischer,"Parallel prefix computation", Journal of ACM. La. Jolla CA, Vol.27,October 1980.
39. Megha Talsania and Eugene John"A Comparative Analysis of Parallel Prefix Adders",worldcomp-proceedings.com/proc/p2013/CDE4118.pdf.

AUTHORS PROFILE

K Mariya Priyadarshini doing her PhD in Koneru Lakshmaiah Educational Foundation, Deemed to be University, Andhra Pradesh, India.she did her Masters in VLSI Design from Karunya University, where she worked on multi domain clock skewing for reducing clock jitters and increase clock efficiency. Her field of research include Two fold Clock Domains, designing of logic circuits for IOT applications.

Dr. R.S. Ernest Ravindran received his PhD in 2016 from Anna University, Chennai, India. He is currently working as assistant professor at K L University Vaddeswaram, Guntur, India. He did his Masters in Nanoscience and Technology from Anna University, Trichy, India where he established surface coated CdS quantum dots as a luminescent probe for silver ion detection. His field of research involves nanocomposites (ceramic-polymer) for high-energy storage application, green synthesis of metal nanoparticles in biomedicine applications.
.

Author-3 Photo

Third Author personal profile which contains their education details, their publications, research work, membership, achievements, with photo that will be maximum 200-400 words.