International Journal of Technical Innovation in Modern Engineering & Science (IJTIMES)
Impact Factor: 5.22 (SJIF-2017), e-ISSN: 2455-2585
Volume 5, Issue 05, May-2019

# Design of Efficient 32-bit Low power Variable Latency Parallel Prefix Brent-Kung Adder

Srinivasulu Gundala[1], Rajani Kumari[2]

[1]*Department of ECE, Lakireddy Bali Reddy College of Engineering, Mylavaram, Krishna Dt. Andhra Pradesh, India,*
[2]*Department of ECE, Lakireddy Bali Reddy College of Engineering, Mylavaram, Krishna Dt. Andhra Pradesh, India*

***Abstract*— A variable latency adder pays speculations in arithmetic circuits can replaced with appropriate one, which will produces faster and correct results. This paper proposes a novel variable latency adder based on Brent-Kung Parallel-Prefix topology that resulted more effective than variable latency Kogge-Stone and Han-Carlson parallel-prefix topology. The proposed adder has two stages of operations, Pre-processing stage and Generation stage. The pre-processing stage of the design has propagate and generate circuits. Generation stage focuses on the carry generation and final result and the performance of the Brent-Kung adder through black cell takes large area. So, Gray cell can be replaced in its place of black cell which gives the Efficiency in Brent-Kung Adder. Finally, a new approach to design of efficient 32 bit low power variable latency parallel prefix Brent Kung Adder (BKA) concentrates on gate levels to improve the increase and decreases the memory. The Proposed Adder which gives the addition operation offers great advantages in reducing delay. Brent-Kung adder mostly used for low-power Designs and in this paper implementation of Brent-Kung Adder synthesized using Xilinx ISE 14.7 has been modelled with VHDL.***

***Keywords*— Speculative Adder, Variable Latency Adder, Parallel-Prefix Adder, Grey Cell, Black Cell.**

## Introduction

Adders are basic functional units in computer arithmetic operations. Binary adders are used in microprocessors for addition and subtraction operations, floating point multiplication and division. Therefore adders are fundamental and basic components, improvement in performance is one of the major challenges in digital designs [1].

The addition procedure is a main process in DSP and control system applications. The speed performance and accuracy of a processor depends on the adders. Multiplexer is a combinational circuit, which consists of multiple inputs and a single output.

Brent Kung adder is used for high performance addition operation, and parallel prefix adder used to perform the addition operation [3]. It is look like a tree structure to perform the arithmetic operations. It consists of black cells and gray cells. [2] Each black cell consists of two AND gates and one OR gate [4]. Each gray cell consists of only one AND gate pi. It denotes propagate and it consists of only one AND gate [5] given in equation 1. gi denotes the generate and it consists of one AND gate and OR gate given in equation 2. [6]

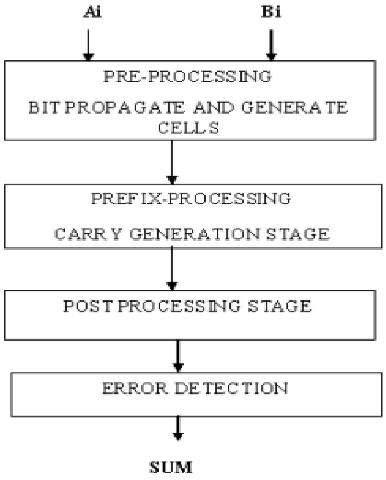


*Fig.1 Block Diagram of Parallel Prefix Adder*

pi= Ai XOR Bi -------- (1)
gi= Ai AND Bi -------------- (2)

Gi denotes the carry generate and it consists of one AND gate and one OR gate given in equation 3.It is used for first black cell. [8]

Gi=pi OR [gi AND cin] --- (3)

The Parallel Prefix adders are suitable for VLSI implementation since it differs from other adders, it can be used for large word sizes. The proposed design reduces the number of prefix operation by using more number of Brent-Kung stages and lesser number of Kogge-Stone Stages. This also reduces the complexity, silicon area and power consumption. Parallel Prefix Adder can be subdivided in the following stages: Pre-Processing, Post Processing, Error Detection and Error Correction. The Error Correction Stage is Off the critical path, as it has two clock cycles to obtain the exact sum when speculation fails. The Pre-Processing and Post-Processing Stages of a Prefix adder involve only simple operations on signals to each bit location.Fig.1. shows the block diagram of Parallel-prefix Adder. Hence, the adder performs mainly on Prefix operation. Therefore black dots represent the prefix operator, and white dots represent the simple place holders

**Literature Survey**

Parallel-Prefix adders are compute addition in two steps: one is to obtain the carry at each bit, and next one is to compute the sum bit based on the carry bit. Inappropriately, prefix trees are algorithmically slower than fast logarithmic adders, such as the carry propagate adders, however, their regular structures promote excellent results when compared to traditional CLA adders. This happens with in VLSI architectures because a carry-look ahead adder, such as the one implemented in one of Motorola's processors [9], tends to implement the carry chain in the vertical direction instead of a horizontal one, which has a tendency to increase both wire density and fan-in/out dependence. Therefore, although logarithmic adder structures are one of the fastest adders algorithmically, the speed efficiency of the carry-look ahead adder has been hampered by diminishing returns given the fan-in and fan-out dependencies as well as the heavy wire load distribution in the vertical path. In fact, a traditional carry-look ahead adder implemented in VLSI can actually be slower than traditional linear-based adders, such as the Manchester carry adder.

Sudheer Kumar Yezerla et al. [10] investigated different types of 16 bit PPA's which were implemented using Verilog Hardware Description Language. The tool used was Xilinx Integrated Software Environment (ISE) 13.2 Design Suite. The parameters considered for results were an area, power, and delay.

Anas Zainal Abidin et al. [11] investigated the performance of 4-bit BKA using Silvaco EDA tool- 0.18um Silterra Technology. Brent Kung Adder was implemented using Basic Logic Gates and Compound Gate, and then they simulation study was done by considering the design in different transistors sizes with power consumption, a number of transistors used and propagation delay as parameters.

Pappu P. Potdukhe et al. [12] proposed architecture for carrying Select Adder (CSA) using parallel prefix adder. 4 bit Brent Kung adder was used to design CSA instead of 4 bit Ripple Carry Adder (RCA). Power and delay of 4 bit RCA and 4-bit BKA architecture were calculated. Relative performances of 4 bit RCA and BKA were described using TANNER EDA tool designs.

Kostas Vitoroulis [13] designed a parallel prefix adder which employs 3-stage structure of carrying look-ahead adder. An improvement was introduced in the carry generation stage different architectures for carry generation were presented. Also, the different parallel prefix adder architectures which were developed since the 1950s were presented. PPA's are basically consists of 3 stages. They are: Pre computation, Prefix stage and Final computation .The diagram fig.2.is shown in below.

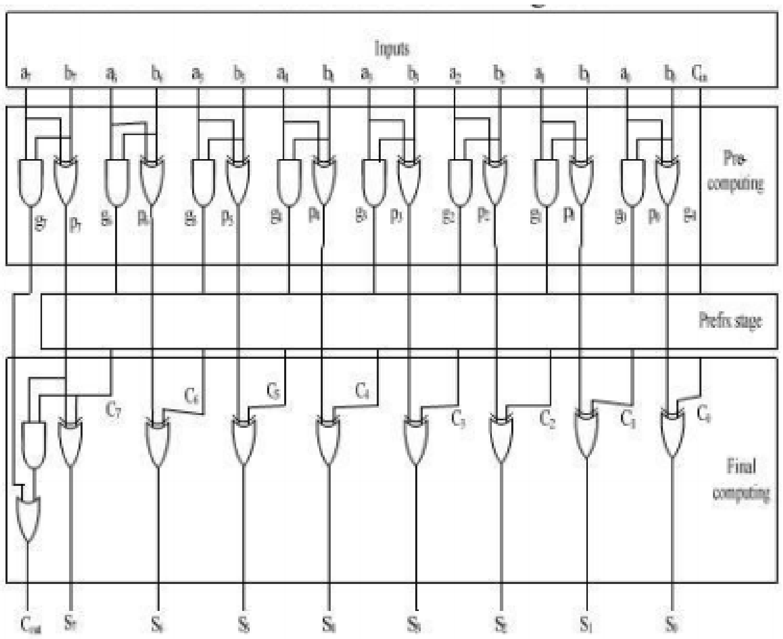

Inputs
a7 b7 a6 b6 a5 b5 a4 b4 a3 b3 a2 b2 a1 b1 a0 b0 Cin
Pre-computing
g7 p7 g6 p6 g5 p5 g4 p4 g3 p3 g2 p2 g1 p1 g0 p0 gin
Prefix stage
C7 C6 C5 C4 C3 C2 C1 C0
Final computing
Cout S7 S6 S5 S4 S3 S2 S1 S0

Fig.2.PPA Structure

In pre computation stage, propagates and generates are Computed In the prefix stage, the black cell (BC) and gray cell (GC) generates only the building prefix structures. Final computation, the sum and carryout are the final output. **Black Cell and Gray Cell** In black cell having four inputs and two outputs propagation means and operation and generation means And-or operation. The diagram is fig.3.is shown in below

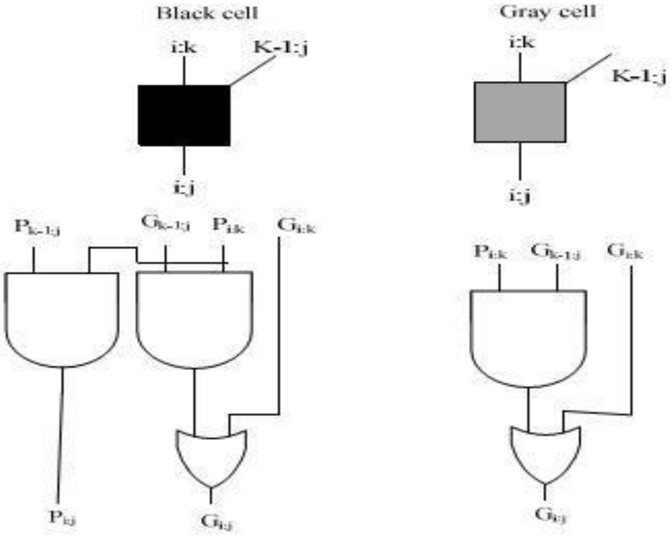


Fig.3. black cell and gray cell

In gray cell having three inputs and one output here generation means and-or operation.

**Han-Carlson Adder:** The Han-Carlson adder is a Full of Kogge-Stone adders. It employs one Kogge-Stone stage, sacking to compute the odd numbered prefixes. It enables better performance compared with Kogge-Stone for smaller adders.

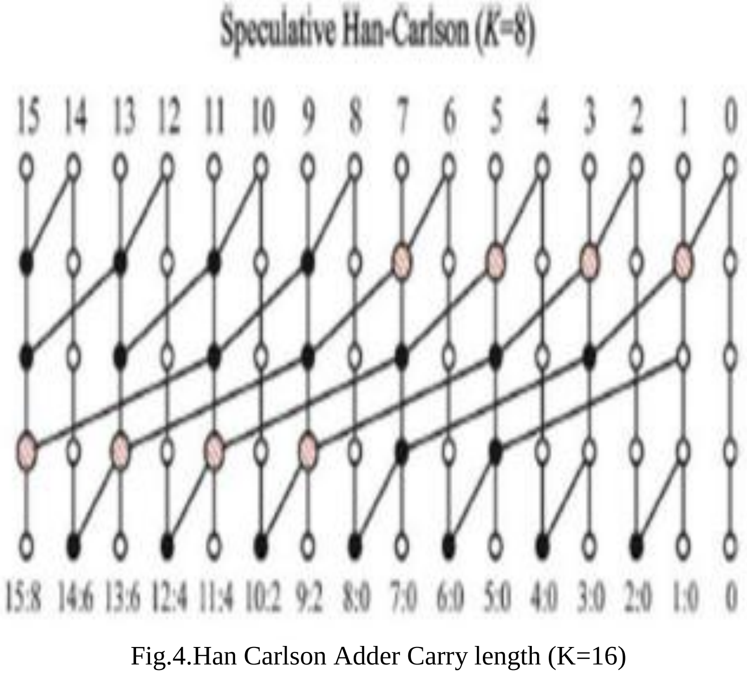


Fig.4.Han Carlson Adder Carry length (K=16)

The Han-Carlson is the family of networks among the Kogge-Stone and Brent-Kung. Han-Carlson adder can be observed of Kogge-Stone adder. This adder is different from Kogge-Stone adder in the sense of these executes carry-merge operation on even bits and generate/propagate operation on odd bits. At the end, these odd bits are recombine and even bits are carry signals to produce the true carry bits. The Han Carlson adder has five stages, wherein the middle three stages resembles with the Kogge-Stone structure. The fig.4.shows the Han Carlson adder. The advantage of the adder is that it uses much fewer cells and its shorter. Consequently there is a reduction in complexity at the cost of an further stage for carry-merge path. We have generated a Han- Carlson Speculative Prefix Processing stage by deleting the last rows of the Kogge Stone adder. That produces a speculative stage is k=8=n/2^p, here p is the number of pruned levels. Additional work is required to encompass the speculative approach to other parallel-prefix architectures, for example Brent-Kung, Ladner-Fisher, and Knowles.

## HAN-CARLSON ADDER WITH ERROR DETECTION AND ERROR CORRECTION:

The conditions wherein at least one of the estimated carries is wrong (mis-prediction) are signed by the error detection stage. In case of mis-prediction, an error signal is emphasised by the error detection stage and the output is post-processing stage is discarded. The error correction stage will give the correct sum in the next clock period. The error correction stage computes the exact carry signals, to be used in case of mis-prediction. The error correction stage is composed by the levels of the prefix- processing stage pruned to obtain the speculative adder.

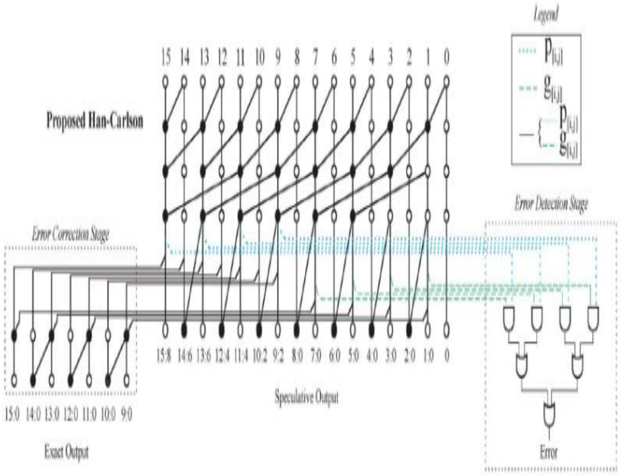


Fig.5. Error correction and detection stages for the proposed speculative Han-Carlson adder

The Fig 5. shows the error correction stage of the proposed speculative Han-Carlson adder; the error correction for Kogge-Stone topology can be obtained similarly. It can be observed that the inclusion of the error correction stage increases the fanout of some of the cells of the speculative prefix-processing stage, with adverse effect on adder speed.

**Proposed System**

Brent-Kung adder is a very popular and most widely used in adders. Generally, it gives an excellent number of stages from input to all outputs but with asymmetric loading of Intermediate stages. It is one of the parallel prefix adders.

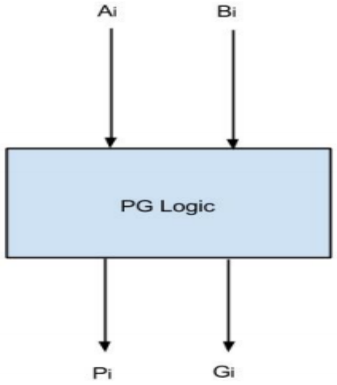


Fig.6 Block diagram of bit wise PG logic

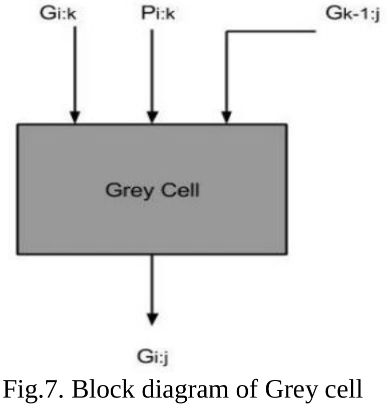


Fig.7. Block diagram of Grey cell

The inputs A and B are given to PG logic as shown in the block diagram. 32 PG logic blocks are needed for a 32-bit adder. The outputs of this block are propagate (P) and generate (G) signals. The block diagram shows the fig 6&7.These signals are given to the tree structure of Brent Kung adder. This structure contains grey cells and black cells arranged as discussed in Brent Kung adder section. A grey cell has three inputs and one output as shown in the figure. Generate and propagate signals from present stage and generate signal from previous stage are inputs. Group generate signals is the output. Each stage ends with a grey cell in any tree structure and the output of this grey cell is the group generate signal which is considered as the carry of that stage. Black cell has 4 inputs and 2 outputs. The inputs for a black cell are P and G signals of present stage and P, G signals of previous stage.

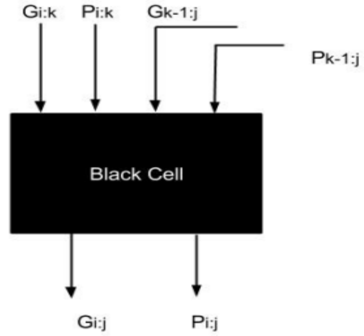


Fig.8. Block diagram of Black cell

It is one of the basic adders where these adders are the ultimate class of adders that are depends on the uses to generate and propagate logics. In case of Brent-kung adders the cost, the wiring complexity is less. But the gate level is depth of the Brent-Kung adders is 0 (log2 (n)). In ripple carry adders each bit to wait for the last bit operation. In this parallel prefix adder is instead of waiting for the carry propagation of the initially addition, the idea here is to overlap the carry propagation of the first addition the computation in the another addition, and so forth, since repetitive additions will be performed by a multi-operand adder.

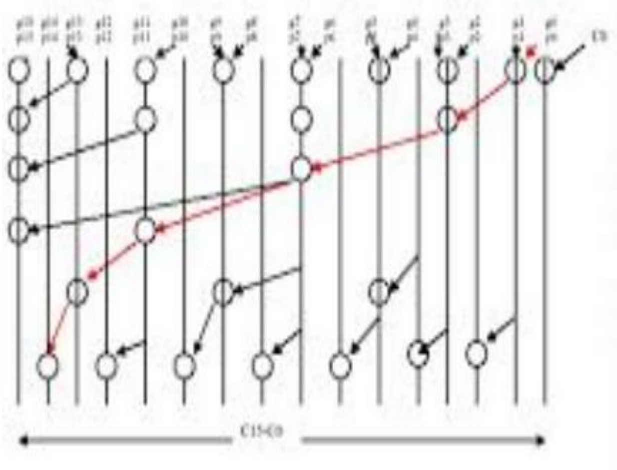


Fig.9. Proposed 32-bit Brent Kung Adder

The structure of efficient Brent kung adder involves of three stages. They are pre-processing stage, carry generation stage, post-processing stage.

**A. Pre-Processing Stage**

In this pre-processing stage, the generate and propagate are from each pair of inputs. The propagate execute “XOR” operation of input bits and generate operation “AND” operation of input bits. The propagate is (Pi) and generate is (Gi) are shown in below. 2.

$$P_i = A_i \text{ XOR } B_i ----\quad (1)$$

$$G_i = A_i \text{ AND } B_i ----\quad (2)$$

**B. Carry Generation Stage**

In this stage, the carry is generated by each bit called as carry generate (Cg). The carry propagate and generate is generated for the more operation but final cell present is the each bit of operation gives carry. The last bit carry will help to produce the sum of the next bit simultaneously to till the last bit. The carry generate and propagate are given in below equations 3 and 4.

$$C_p = P_1 \text{ AND } P_0 ------\quad (3)$$

$$C_g = G_1 \text{ OR } (P_1 \text{ AND } G_0) --\quad (4)$$

In the above carry propagation is Cp and carry generation is Cg.The above equations 3 & 4 is black cells and the below equation is carry generation in equation 5 is a gray cell. The carry propagate is generated by the further operation ,but final cell present in the each bit operation gives the carry. The last bit carry will help to produce sum of the next bit to concurrently till the last bit. In this carry is used for the next bit sum operation, the carry generate is given in below equations 5.

$$C_g = G_1 \text{OR } (P_1 \text{ AND } G_0) --\quad (5)$$

**C. Post-Processing Stage**

This is the final stage of an efficient Brent-Kung Adder, the carry of a first bit is XOR with the next bit of propagates then the output is given by the sum and it is shown in equation.6

$$S_i = P_i \text{ AND } C_{i-1} - - - - - \qquad (6)$$

It is used for two sixteen bit addition operations and each bit carry is endures to the post-processing stage with the propagate and produce the final sum. The first input bits goes to under the pre-processing stage and it will produce to the propagate and generate signals. These propagates and generates suffers carry generation stage which produces the carry generates and propagates, these undergoes post-processing stage and it provides the final sum.

In Efficient Brent Kung adder, the black cell operates three gates and the gray cell operates two gates. The gray cell will reduce to the delay and memory because it works only two gates. The proposed adder is design with both the black cells and gray cells. By using the gray cell operations at the last stage of proposed adder is gives a enormous dropping delay and memory used.

The proposed adder is shown in fig 8 which is it increases the speed and it decreases the memory for the operation of 8-bit addition. The input bits are Ai and Bi concentrates on generate and propagate by XOR and AND operations. These propagates and generates undergoes to the operations of black cell and gray cells provides the carry Ci. That carry is XOR with the propagate of next bit, that gives sum.

Simulation Results

The Efficient Brent-Kung Adder is design with an VHDL (Very High speed integration hardware description Language.). Xilinx ISE 14.7 is used and Simulation results of efficient 32-bit Low PowerVariable Latency Speculative Parallel Prefix Brent-Kung Adder are as below. Fig.10 shows the RTL schematic of the proposed design and Fig.11represents technology schematic.

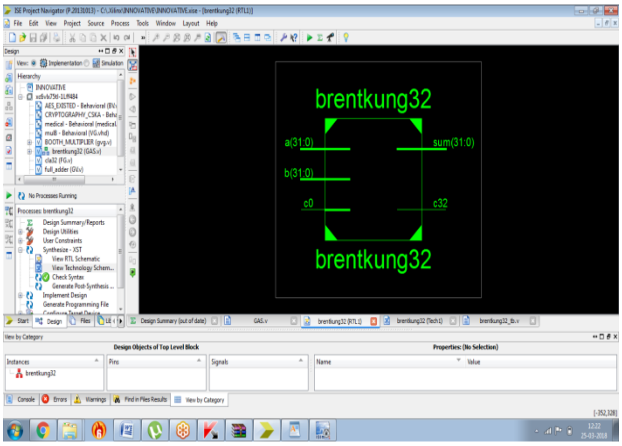


Fig.10.RTL Schematic

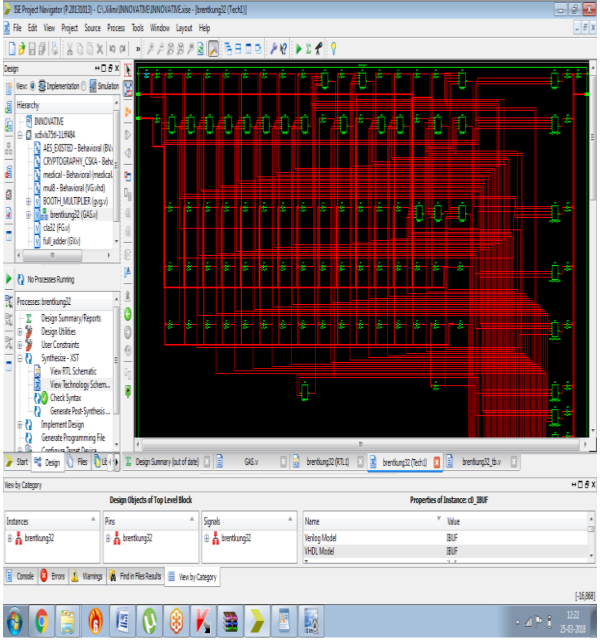


Fig.11.Technology Schematic

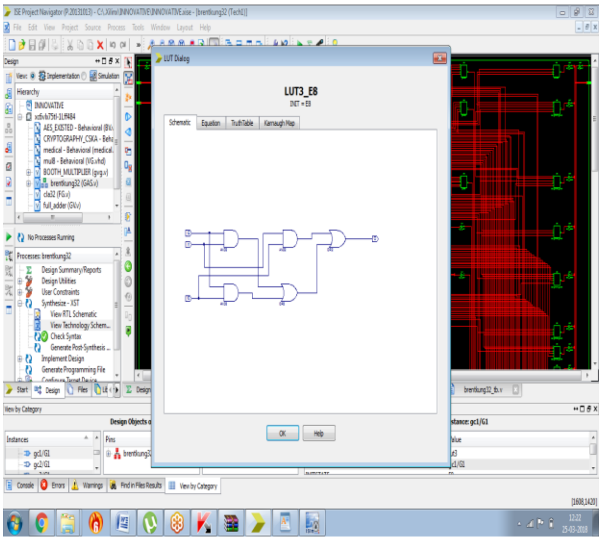
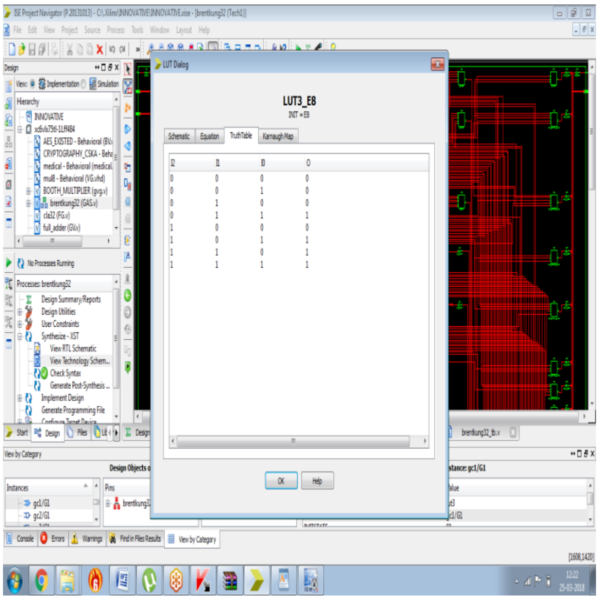


Fig.12 LOOK UP TABLE (LUT)

The design of adders is done on VHDL and the design of Look up table (LUT) is shown in Fig.12.

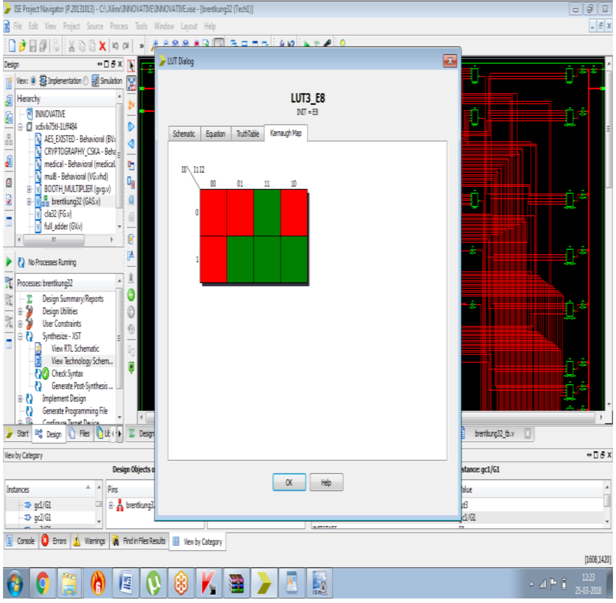


LUT3_E8

INIT = E8

| I2 | I1 | I0 | O |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

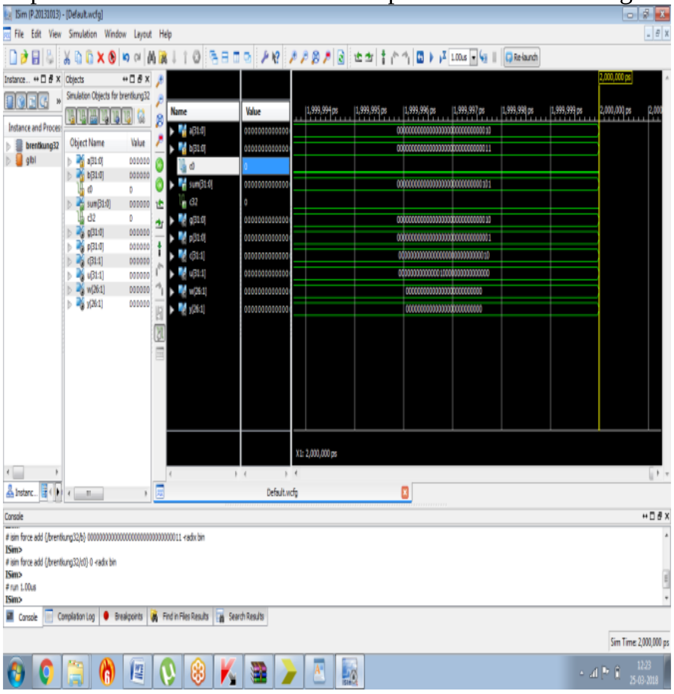
Fig.13Truth Table



| I0 \ I1I2 | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 🟥 | 🟥 | 🟩 | 🟥 |
| 1 | 🟥 | 🟩 | 🟩 | 🟩 |

Fig.14 K MAP

The Above Fig.13 and Fig.14 represents the truth table and K-Map of 32 bit Brent-Kung Adder.



```
# isim force add {/brentkung32/b} 00000000000000000000000000000011 -radix bin
ISim>
# isim force add {/brentkung32/c0} 0 -radix bin
ISim>
# run 1.00us
ISim>
```

Fig. 15Simulation Result of 32-Bit Brent-Kung Adder

The simulation result is 32-bit Low Power Variable Latency Speculative parallel prefix Brent-Kung Adder has been shown in Fig.145

## CONCLUSION

In this paper, a new approach to design an Efficient 32-bit variable Latency Speculative Brent-Kung parallel prefix Adder concentrates on gate levels to improve the speed and decreases the memory. It is like tree structure and cells in the Carry Generation Stage are decreased to speed up the binary addition. The Proposed Adder addition operation offers great advantage in reducing delay. The future scope is to design 32 Bit Proposed Adder with less number of black cells to improve the area and delay performance of this adders.

## REFERENCES

[1] Koren, Computer Arithmetic Algorithms. Natick, MA, USA: A KPeters, 2002.

[2] David h,k hoe, Chris Martinez and sri jyothsna vundavalli "Design and characterization of parallel prefix adders using FPGAs", Pages.168- 172, march2011 IEEE.

[3] K.Vitoroulis and A.J. Al-Khalili, "performance of parallel prefix adders implemented with FPGA technology," IEEE Northeast Workshop on circuits and systems, pp.498-501, Aug. 2007.

[4] Haridimos t.vergos, Member, IEEE and Giorgos Dimitrakopoulos, Member, IEEE,"On modulo 2n+1 adder design"IEEE Trans on computers, vol.61, no.2, feb 2012

[5] Giorgos Dimitrakopoulos and Dimitris Nikolos, "High-Speed Parallel-Prefix VLSI Ling Adders" IEEE Trans on computers, vol.54, no.2, Feb. 2005.

[6] S.Knowles,"Afamily ofadders," Proc.15th Symp. Comp. Arith.,pp.277-281,June2001.

[7] R.BrentandH.Kung,"Aregularlayoutfor parallel adders, "IEEETrans. Computers, vol.C-31, no.3, pp. 260-264, March1982.

[8] R.E. Brent and M.J. Kung, "Parallel Prefix Computation," J. ACM, vol. 27, no. 4, pages 831- 838, Oct. 1980.

[9] K. Du, P. Varman, and K. Mohanram, "High performance reliable variable latency carry select addition," in Proc. Design, Autom., Test Eur.Conf. Exhib. (DATE '12), Mar. 2012, pp. 1257–1262.

[10] Sudheer Kumar Yezerla, B RajendraNaik, "Design and Estimation of Delay, Power and Area. for parallel prefix adders*", proceedings of 2014 RAECS UIET Panjab University Chandigarh*, 06-08 March, 2014.

[11] Anas Zainal Abidin, Syed Abdul Mutalib Al Junid,Khairul Khaizi Mohd Sharif, Zulkifli Othman, Muhammad AdibHaron, "4-bit Brent Kung Parallel Prefix Adder Simulation Study Using Silvaco EDA Tools",DOI. 10.5013/IJSSST.a.13.3A.07.

[12] Pappu P. Potdukhe, Vishal D. Jaiswal, "Design of high speed carry select adder using Brent kung adder*", IEECOT,International conference, Chennai, India*, 03-05 March 2016.

[13] Kostas Vitoroulis, "Parallel Prefix Adders", Concordia University, 2006

[14] Pakkiraiah. Chakali, madhu Kumar. Patnala "Design of high speed Ladner - Fischer based carry select adder" IJSCE, Volume-3, Issue-1, March 2013.