INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 4.295
(Volume 4, Issue 3)
Available online at: www.ijariit.com

Simulation study of brent kung adder using cadence tool

T. Vamshi Krishna
vamshi27496@gmail.com
School of Engineering and Technology, Jain University, Bangalore Rural, Karnataka

Niveditha S
niharikads118@gmail.com
School of Engineering and Technology, Jain University, Bangalore Rural, Karnataka

Mamatha. G. N
mamathagn50@gmail.com
School of Engineering and Technology, Jain University, Bangalore Rural, Karnataka

Sunil. M. P
sunilmp93@gmail.com
School of Engineering and Technology, Jain University, Bangalore Rural, Karnataka

ABSTRACT

Adders are the most fundamental piece of any computerized framework. In order to perform the addition of two numbers, adders are used. They also form the requisite part of Arithmetic and Logic Unit. Besides this application, they are also used in computers to calculate address, indices and operation codes. Adders are also used to employ different algorithms in Digital Signal Processing.

There is a prerequisite to provide an efficient adder design which fulfills the trade-off amongst speed and space consideration to increase the performance of the system. In the modern age, in addition to the trade-off between speed and space, power consumption assumes an imperative. Gadgets with low power utilization and good performance are favored in real-time applications. Parallel Prefix adders are the ones generally utilized as a part of Digital Designs due to the adaptability associated with outlining these Adders.

Brent Kung Adder7 (BKA) is a low power parallel prefix adder, as it uses minimum circuitry to obtain the result. A simulation study of this adder is carried out using cadence tool. The 4 bit, 8 bit, 16 bit and 32-bit BKAs were designed and simulated using CMOS logic- 45nm Technology. A comparative study2 was made by comparing the obtained results with Ripple Carry adder and Carry Look-ahead adders10. Obtained results show that the power consumption and propagation delay for the BKA implementation are reduced compared to RCA and CLA.

Keywords: *Brent Kung Adder, Delay, Parallel Prefix Adder, and Power.*

1. INTRODUCTION

Computation is a type of calculation that includes both arithmetical and non-arithmetical steps and follows a well-defined model like an algorithm11. The arithmetical steps include addition, subtraction, multiplication, division etc. The process of calculating the total of two or more numbers is called as addition and the circuits which perform this operation are called adders. The fundamental block of any digital design is an adder. Apart from addition, adders also perform other functions like subtraction, multiplication, and division. Very Large Scale Integrated adders find applications in Arithmetic and Logic Unit (ALU), microprocessors and memory addressing units. Any adder must satisfy the trade-off between power consumption, speed, and area.

Basic types of adders are a half adder and a Full adder which can add two bits and three bits respectively and gives the sum and carry as outputs12. To add a larger series of numbers, logic schemes such as carry look ahead, carry skip or carry select are used. But, as the width of the adder increases, the propagation delay of carrying passing through the stages becomes dominant. Therefore, in current technology, Parallel Prefix Adders7 (PPA) is the best among the existing adders, with respect to the area and delay, and are particularly good for high-speed addition of large numbers. Parallel prefix adder, as the name suggests, it describes a prefix as an outcome of the execution of the operation depending on the

initial inputs. Parallel in the name defines that the process involves the execution of the operation in parallel. This is done by segmentation into smaller pieces that are then computed in parallel4. Then all the bits of the sum will be processed simultaneously which leads to the faster execution of operation with reduced delay.

Richard P. Brent and H.T. Kung designed Brent Kung Adder (BKA) in the year 1982. It is a very well-known parallel prefix adder which gives an optimal number of stages from input to all outputs but with asymmetric loading on all intermediate stages3. BKA occupies less area than the other 3 adders called Sparse Kogge Stone Adder (SKA), Kogge-Stone adder (KSA) and Spanning tree adder1. This adder uses a limited number of propagating and generate cells than the other 3 adders. The cost and wiring complexity is less in Brent Kung adders.
Brent Kung adder usually computes the sum in 3 stages2.

- The initial stage consists of a Pre-processing unit where Group Generate and Group Propagate signals are obtained from inputs
- The intermediate stage is the Carry generation stage to which the outputs of the pre-processing stage are fed as inputs and carry signals are generated.
- The last stage is post-processing where the final result is obtained using the Carry signal from the intermediate stage and propagate a signal from the initial stage.

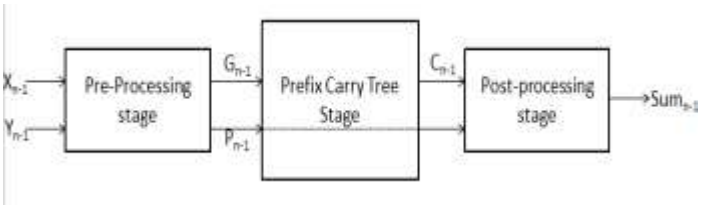


Fig-1 Block Diagram of Brent Kung adder [1]

2. LITERATURE SURVEY

Sudheer Kumar Yezerla et al. [1] 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. [2] 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. [3] proposed an 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 [4] 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.

Noel Daniel Gundi [5] extended a sixteen-bit BKA style to thirty-two bit exploitation complementary pass semiconductor unit logic and enforced. The parameters thought for results were space and delay. it had been shown that the CMOS style has lesser propagation delay compared to the CPL style. The transistors employed by CPL style were more in number.

3. PRELIMINARY BACKGROUND

R.Brent and H.Kung designed Brent Kung Adder in the year 1982.
Brent Kung adder has 3 stages namely pre-processing stage, prefix carry tree stage and post-processing stage. The function of each stage, their circuits along with necessary equations are shown below.

3.1 Pre-processing Stage:

This stage consumes two inputs Ai and Bi and produces two outputs- generate signal Gi and Pi2. The outputs are computed using the following equations.

$$Gi = Ai + Bi \text{---(1)}$$
$$Pi = Ai \text{ xor } Bi \text{ ---(2)}$$

The circuit can be obtained by referring to the above equations (1) and (2).

Fig-2 Pre-processing circuit

The block of this stage will be used at every single input bits of the adder.

3.2 Carry Generation stage:

The signal from the pre-processing stage can proceed with a consequent stage so as to get all carry bit signals. This stage contains 3 main complicated logic cells referred to as-as Black cell, gray cell, and buffer cell. Black cell works out each Gi:j and Pi:j as outlined in equation (3) and (4), whereas grey cell solely executes Gi:j2 (3).

$$Gi:j = Gi:k + Pi:k\ Gk\text{-}1:j \text{ ---(3)}$$
$$Pi:j = Pi:kPk\text{-}1:j \text{ ---(4)}$$

The content of all the 3 cells is shown below.

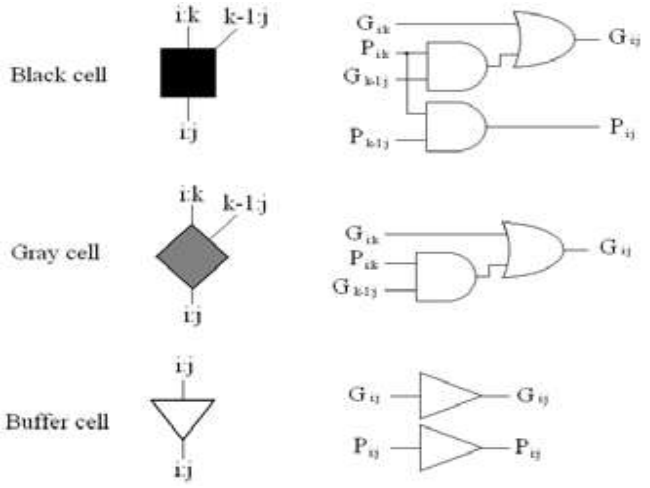


Fig-3 Complex logic cells in carry generation stage [1]

A Brent Kung prefix tree is designed using these cells and the tree will differ as the number of input bits differ.

3.3 Post-processing stage:

This stage is the final one where the exclusive-OR operation is done between the propagate signal, Pi and a lower bit carry signal output from the carry generation stage, Ci-1. The final adder result can be obtained by following equations2 (5) and (6).

$$Si = Pi \text{ xor } Ci\text{-}1 \quad \text{---(5)}$$
$$Ci = Gi + Pi\ Ci\text{-}1 \text{ or } Ci = Gi \quad \text{---(6)}$$

The circuit is as shown.

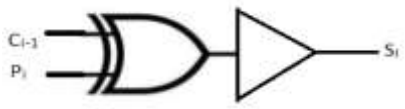


Fig-4 Post-processing circuit

After designing circuits of all the stages, they are combined to obtain the Brent Kung adders for any number of input bits.

4. METHODOLOGY

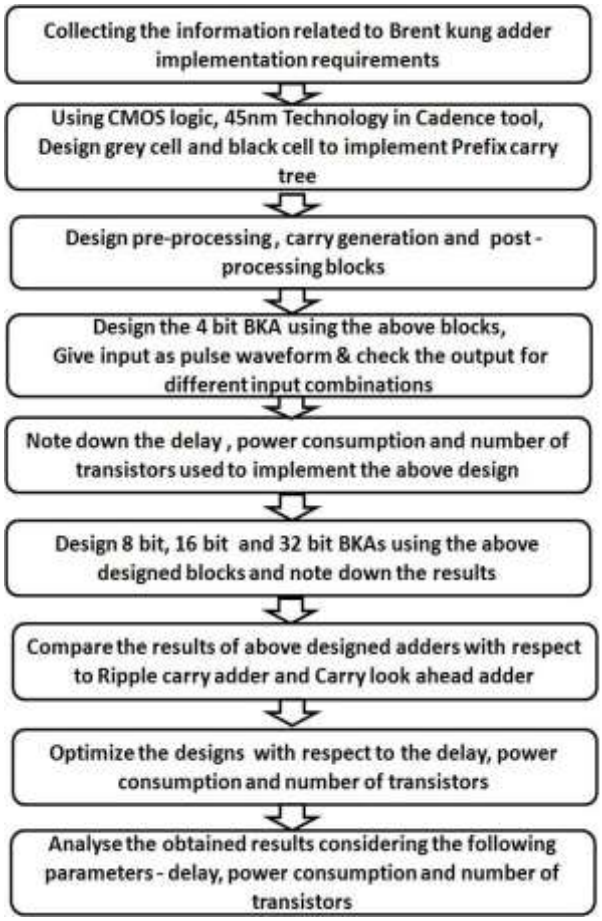


Fig-5 Methodology-Flow chart

4.1 Implementation:

By using 3 Logic Gates- AND, OR and XOR, the other transistor level circuits are designed and implemented. The Schematic of AND, OR, XOR and Buffer gates is as shown below.

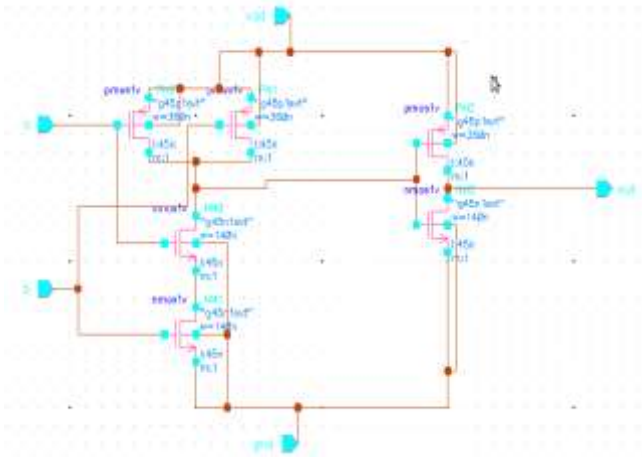


Fig-6 Schematic of AND gate

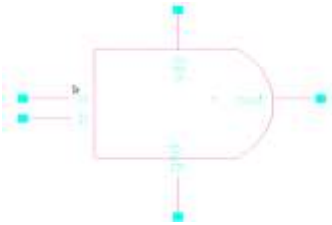


Fig-7 Symbol of AND gate

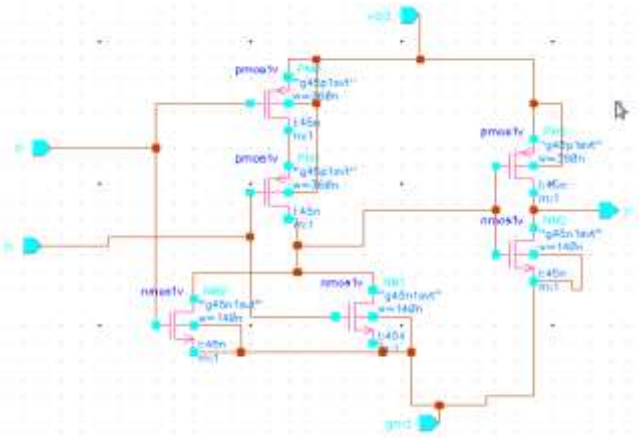


Fig-8 Schematic of OR gate

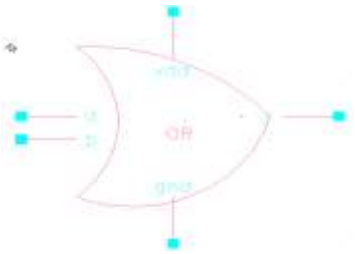


Fig-9 Symbol of OR gate

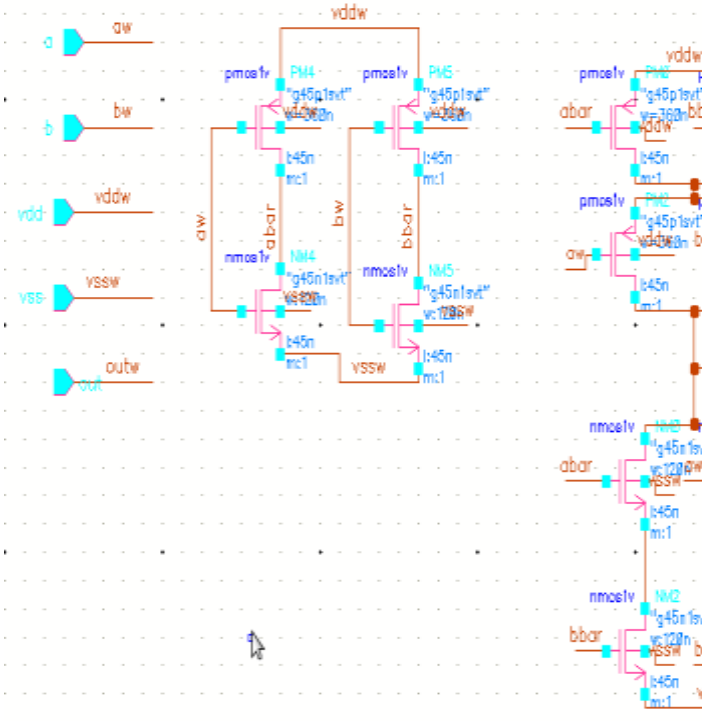


Fig-10 Schematic of XOR gate

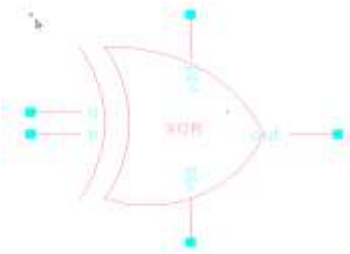


Fig-11 Symbol of XOR gate

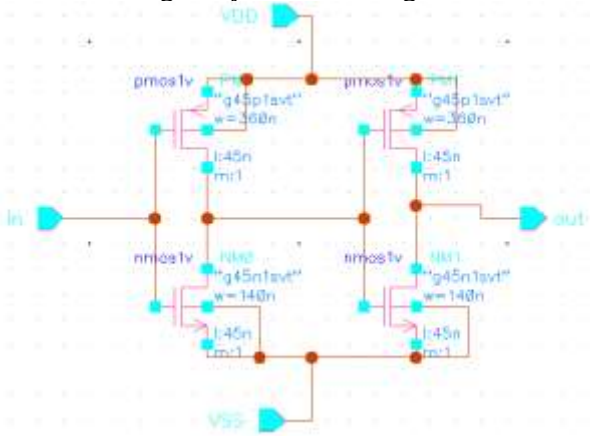


Fig-12 Schematic of Buffer

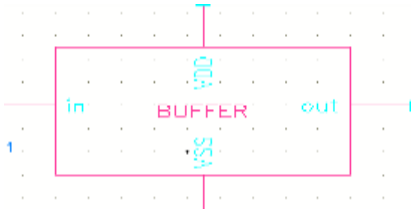


Fig-13 Symbol of Buffer

Using the above gates, Black cell, and Gray cell were designed from the equations 3 and 4. Their schematics are as shown below.

Black Cells gives Group Generate and the Group Propagate bits as its output whereas gray cell gives out only Group Generate bit

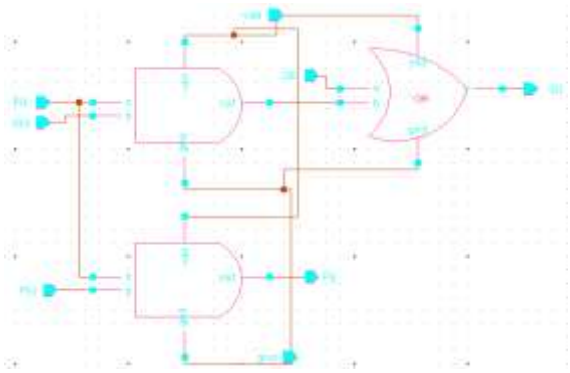


Fig-14 Black Cell

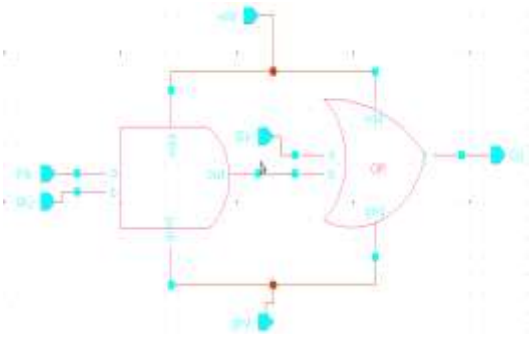


Fig-15 Gray Cell

Carry generation stage is designed using black cell and gray cell.
Pre-Processing stage is nothing but a half-adder which gives sum and carry bits for the respective inputs which are used as generate(Gi) and propagate(Pi) signals for further stages.

$$Gi = Ai + Bi$$
$$Pi = Ai \text{ xor } Bi$$

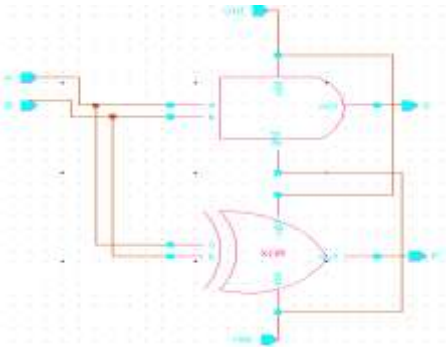
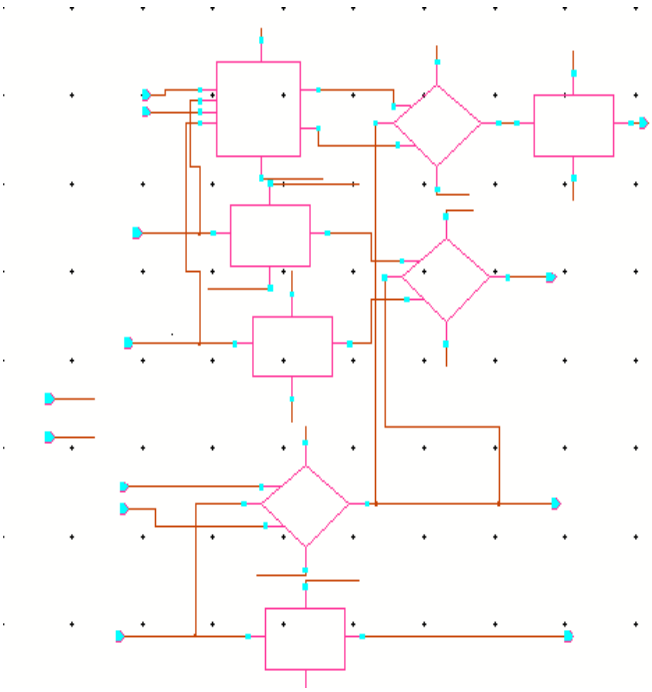


Fig-16 Pre-Processing Stage

The signal from the pre-processing stage will act as inputs to this stage. The carry bit signals are computed here. This stage contains three main cells which are a black cell, gray cell and buffer as described above.



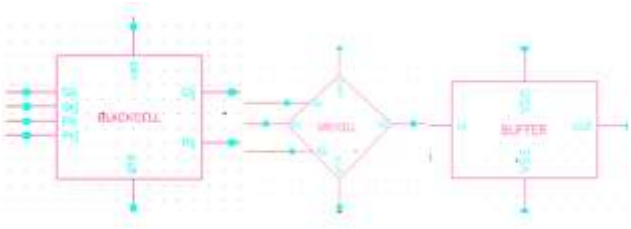


Fig-17 Carry generation Stage

Post-processing is the final stage from which we obtain the end result.

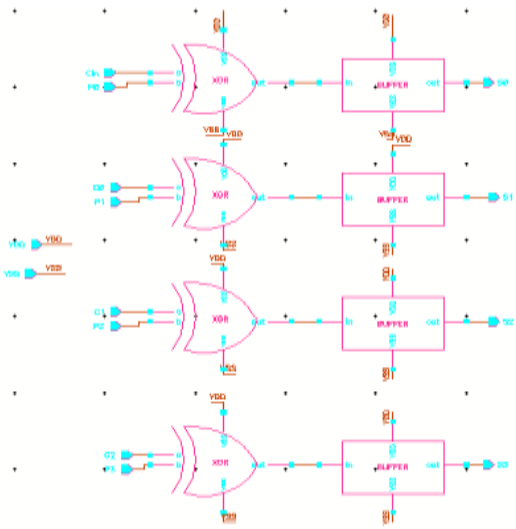


Fig-18 Post-Processing Stage

The 4-bit BKA is designed using 3 stages. The schematic of 4-bit BKA is as shown below

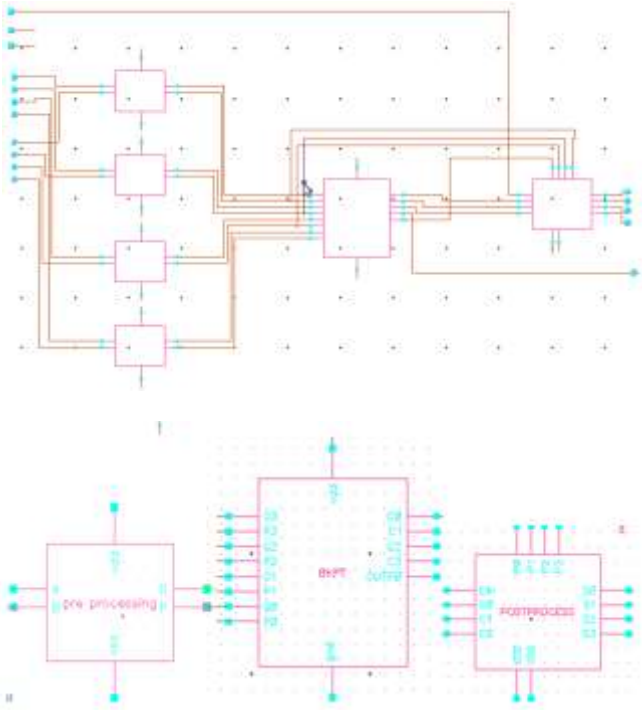


Fig-19 4-bit Brent Kung Adder schematic

Ripple carry adder can be constructed by cascading full adders in series. The carry-out of the present stage is fed as carry-in to the succeeding stage. It is called as a ripple carry adder because each carry bit gets rippled into the next stage.

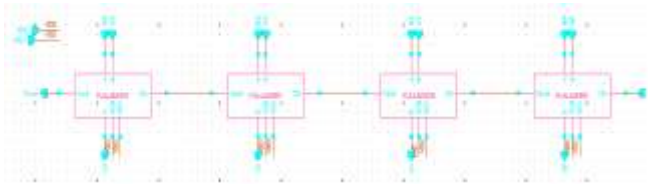


Fig-20 4-bit Ripple Carry Adder

Carry Look-ahead Adder works similar to that of RCA but it uses a logic called carry look-ahead logic which makes it different from the other adders (RCA).

The equations are given below.

Cout= AB+ [A xor B] Ci and Ci=Gi + Pi Ci-1

Where G = AB = Carry Generate- carry is generated irrespective of carry from previous stage Ci.

P = A xor B = Carry Propagate- carry from the previous stage Ci is propagated to next stage if A xor B is 1.

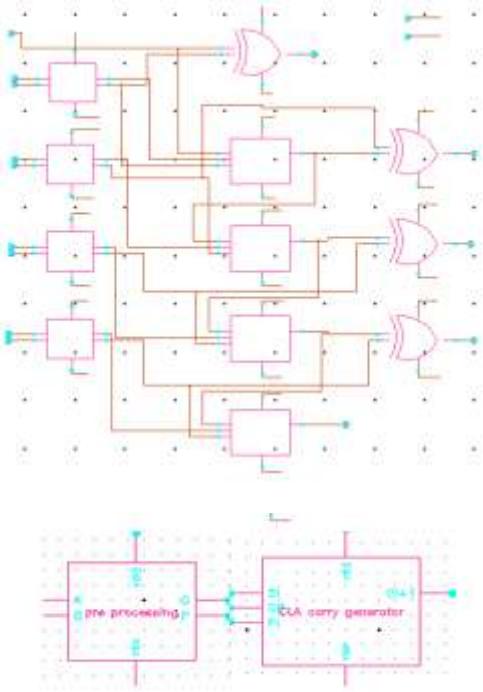
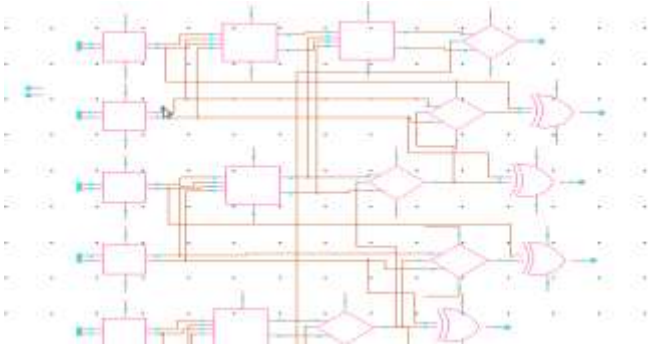


Fig-21 4-bit Carry Look Ahead Adder

An 8 bit, 16 bit and 32-bit BKA are also designed similar to 4-bit BKA. The schematics of 8 bit, 16 bit and 32 BKA, RCA and CLA are shown below respectively.



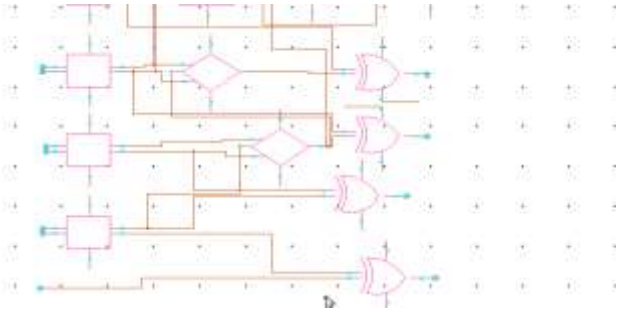


Fig-22 8-bit Brent Kung Adder

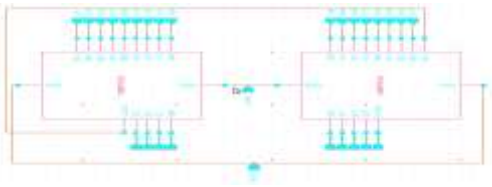


Fig-23 8-bit Ripple Carry Adder

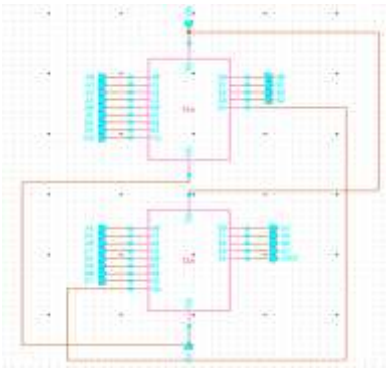


Fig-24 8-bit Carry Look Ahead Adder

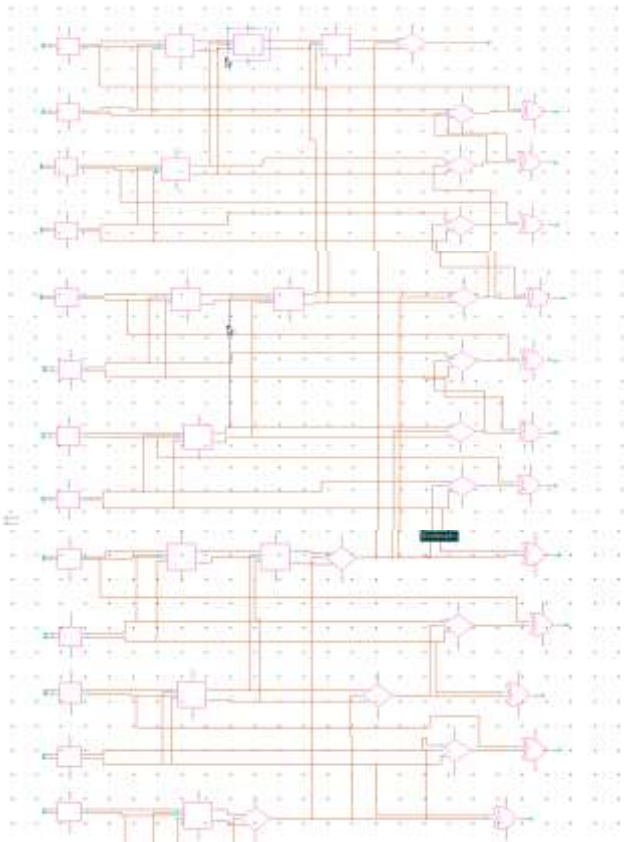


Fig-25 16-bit Brent Kung Adder

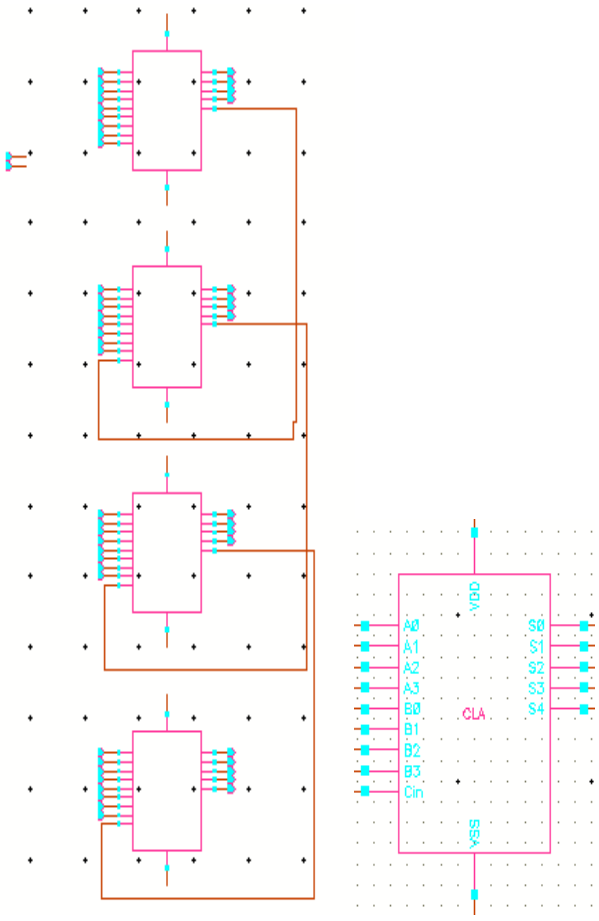


Fig-26 16-bit Carry Look Ahead Adder

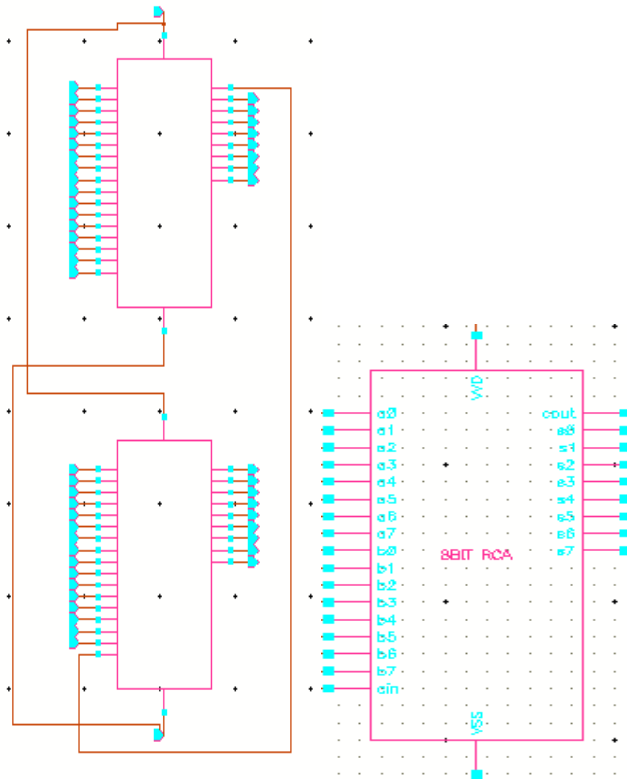
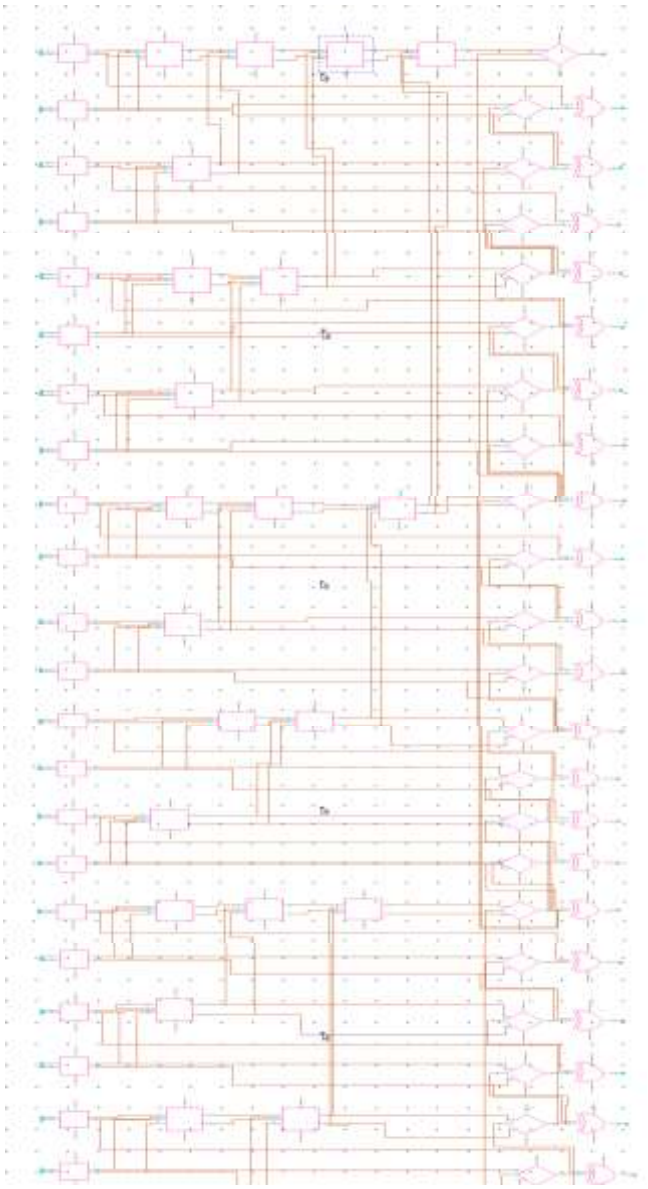


Fig-27 16-bit Ripple Carry Adder



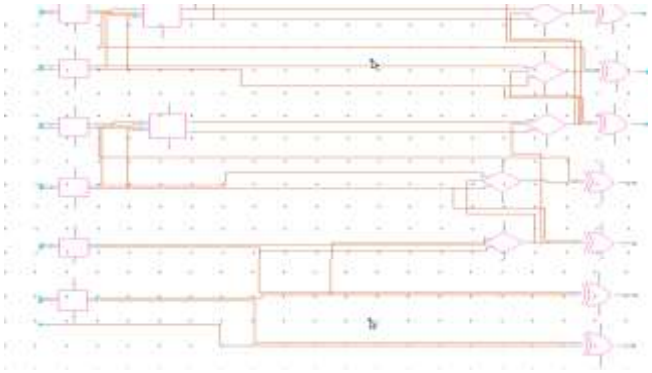


Fig-28 32-bit Brent Kung Adder

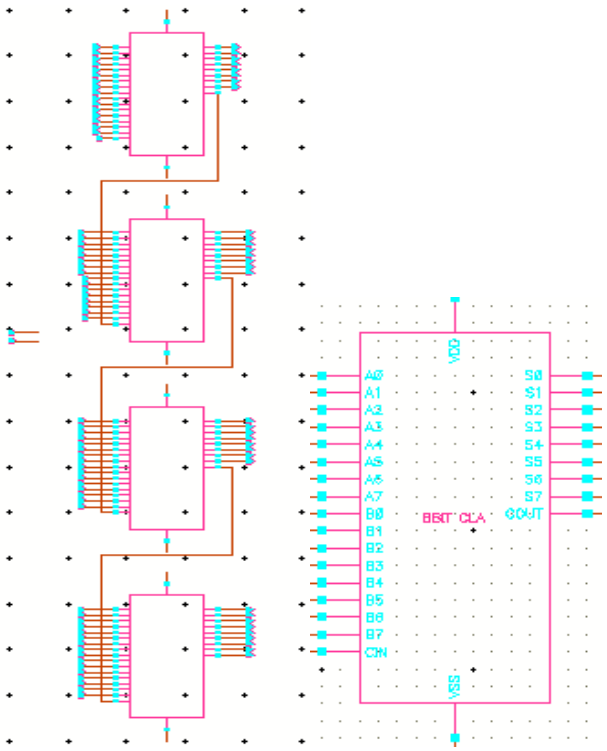


Fig-29 32-bit Carry Look-ahead Adder

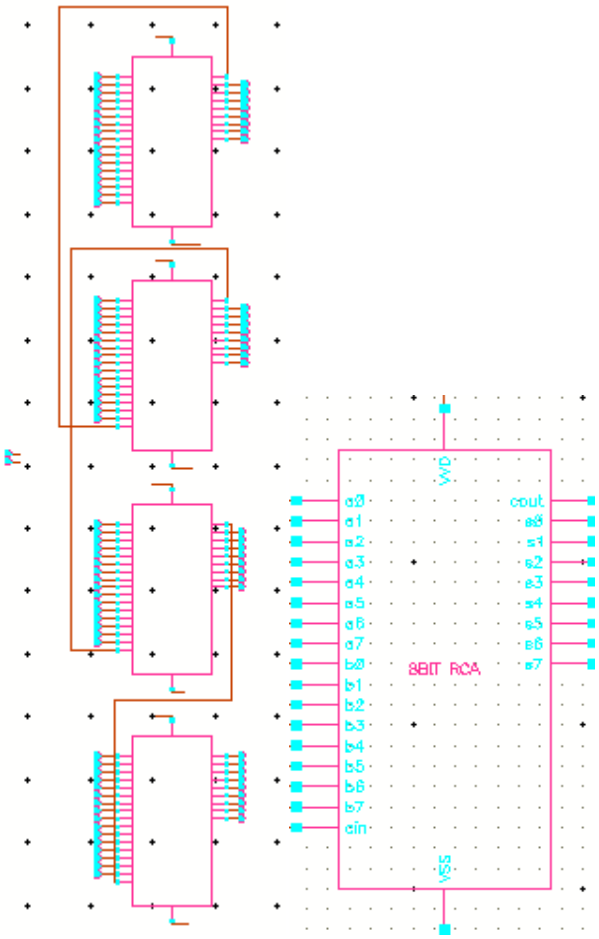


Fig-30 32-bit Ripple Carry Adder

5. RESULTS AND DISCUSSION

The circuits are designed in cadence virtuoso environment using 45nm technology GPDK tool kit with a voltage supply of 1V and threshold voltage of 0.5V. The output waveforms and results of 4 bit, 8 bit, 16 bit and 32 bit Brent Kung adder, ripple carry adder and carry look-ahead adder are shown below respectively.

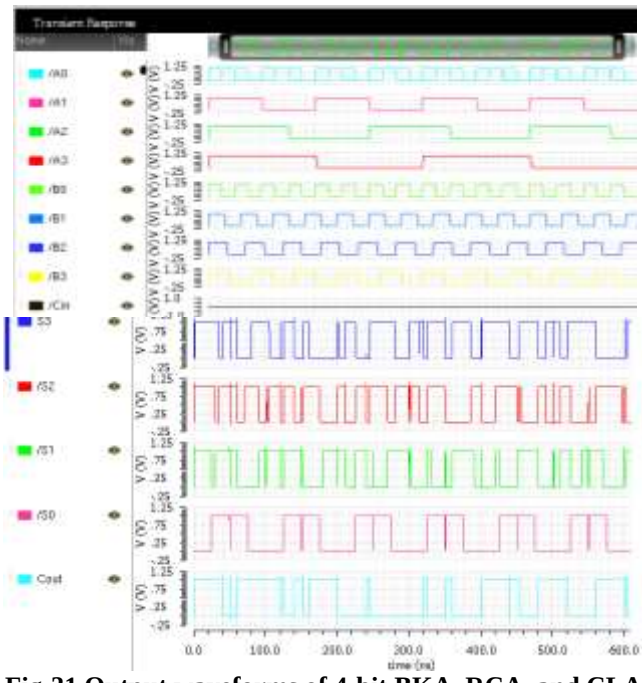


Fig-31 Output waveforms of 4-bit BKA, RCA, and CLA

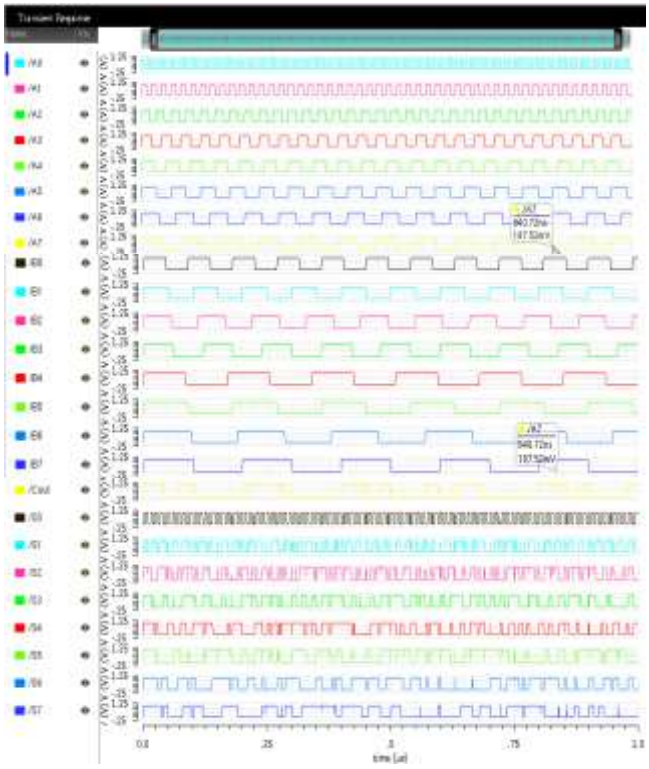
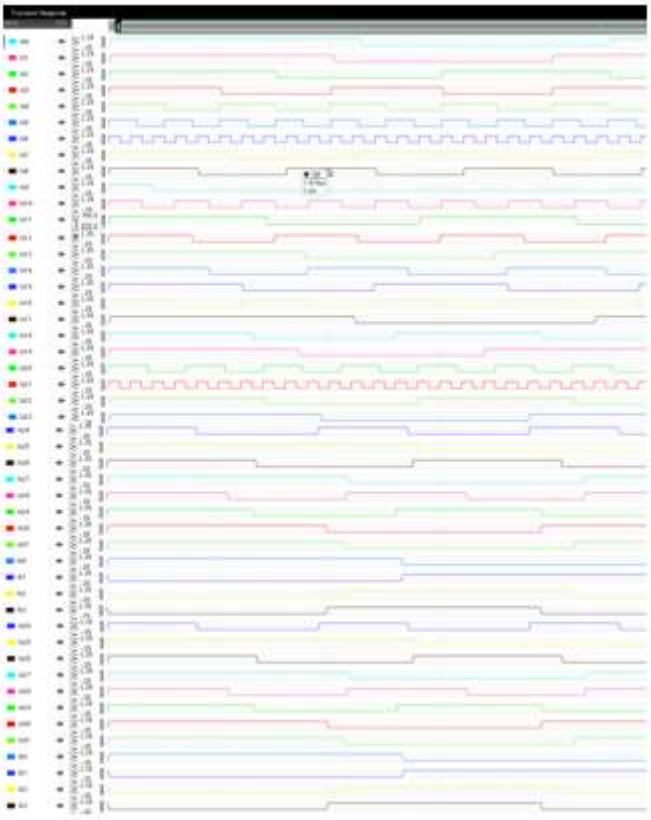


Fig-32 Output waveforms of 8-bit BKA, RCA, and CLA



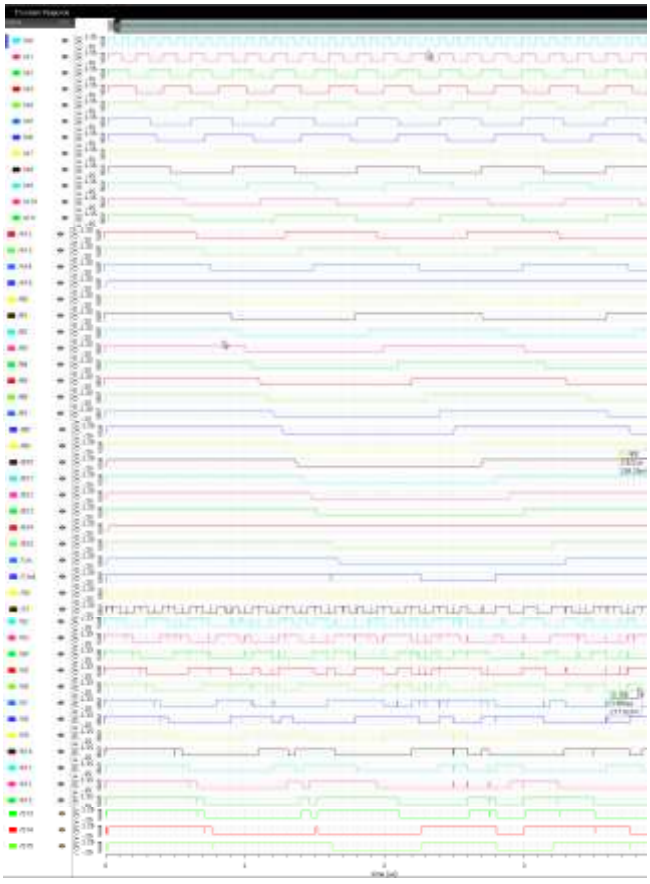


Fig-33 Output waveforms of 16-bit BKA, RCA, and CLA

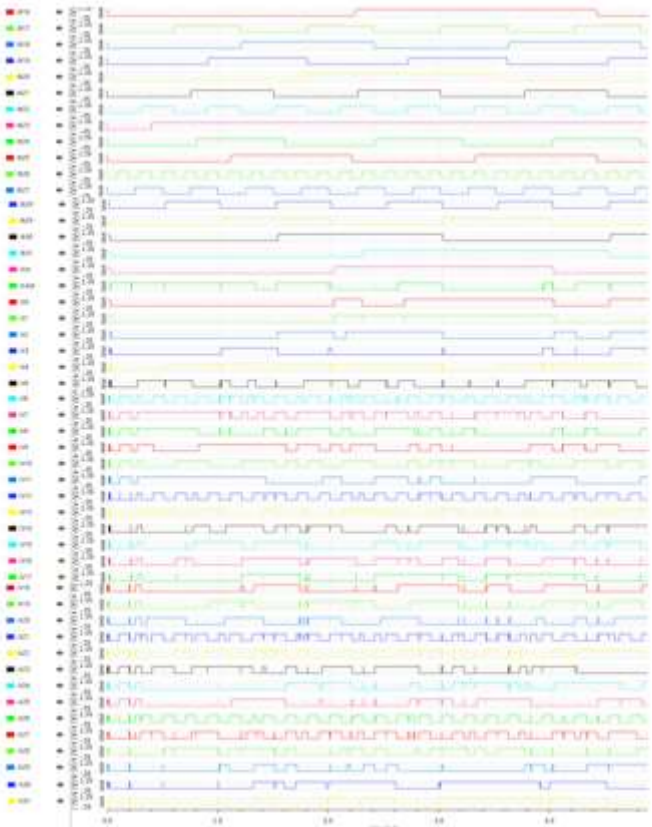


Fig-34 Output waveforms of 32-bit BKA, RCA, and CLA

Table-1: Table of results

Circuit	Delay(sec)	Power Consumption (watts)	No. of transistors used
4 bit BKA	19.396e-9	936.5e-9	160
4 bit RCA	20.347e-9	885.4e-9	168
4 bit CLA	31.224e-9	340.1e-9	192
8 bit BKA	14.927e-9	3.661e-6	384
8 bit RCA	68.235e-9	1.017e-6	336
8 bit CLA	78.688e-9	0.561e-6	384
16 bit BKA	265.04e-9	649.4e-9	876
16 bit RCA	235.41e-9	583.5e-9	672
16 bit CLA	180.03e-6	572.6e-9	768
32 bit BKA	187.59e-9	1.375e-6	1782
32 bit RCA	213.31e-9	1.891e-6	1344
32 bit CLA	209.84e-9	1.164e-6	1536

From the simulation study, it can be observed that the propagation delay of BKA is 63.297% lesser compared to CLA and 26. 425% compared to RCA. Also, the power consumption for 32 bit BKA is 27.3% lesser compared to 32 bit RCA.

From the obtained results, it can be concluded that BKA is the fastest adder compared to the other two adders i.e., RCA and CLA even though it uses number of transistors.

Further, the number of transistors required for designing BKA can be optimized. Also, the power consumption by BKA can be reduced by making proper transistor sizing.

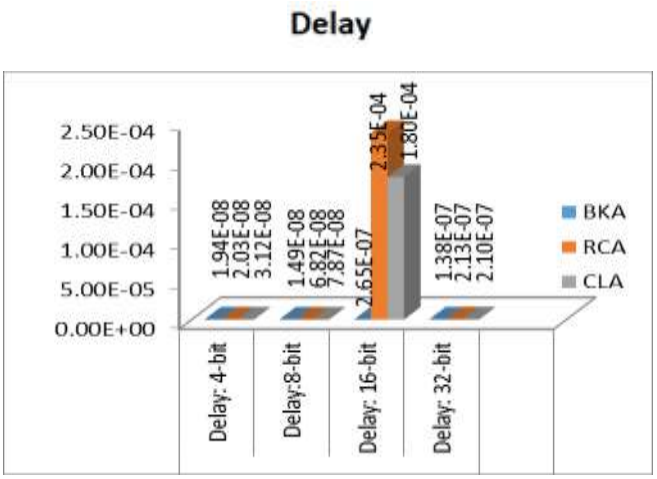


Fig-35 Comparison of propagation delay for 4,8,16 and 32-bit BKA, RCA and CLA respectively

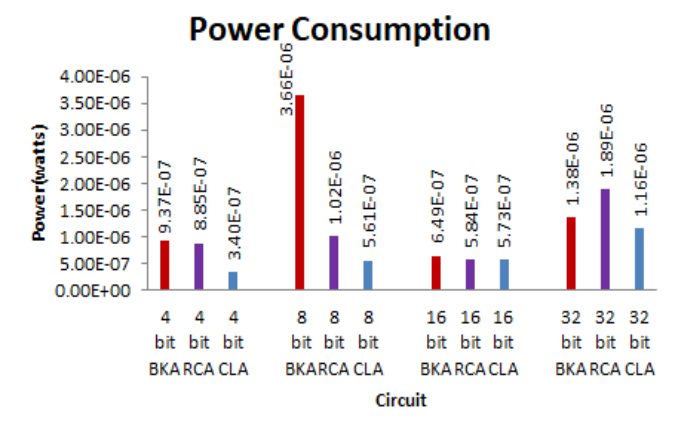


Fig-36 Comparison of Power Consumption for 4,8,16 and 32-bit BKA, RCA and CLA respectively

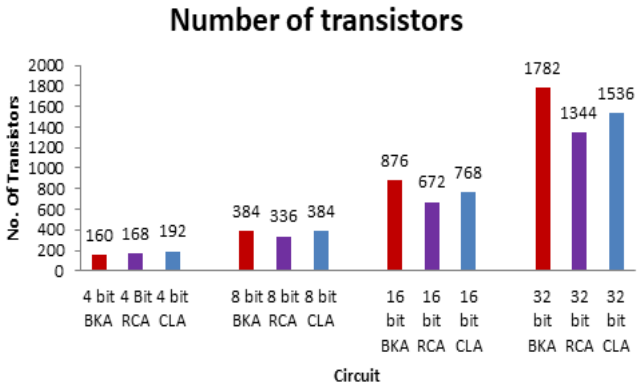


Fig-37 Comparison of number of transistors used for 4,8,16 and 32 bit BKA, RCA and CLA respectively

6. REFERENCES

[1] 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

[2] 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.

[3] 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.

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

[5] Noel Daniel Gundi, "Implementation of 32 bit Brent Kung Adder using complementary pass transistor logic", June 2008. [6] Vibhuti Dave, ErdaOruklu and JafarSaniie, "Performance Evaluation of Flagged Prefix Adders for Constant Addition", Department of Electrical and Computer Engineering, Illilois Institute of technology, Chicago, 2006.

[7] Richard P. Brent and H.T. Kung, "A Regular Layout for Parallel Adders", *IEEE Transactions on Computers* Volume 31 Issue 3, March 1982, Pages 260-264.

[8] Rashmi D.S, SadiyaRukhsar. R, Shilpa H.R, Vidyashree C.R, Kunjan D Shinde, Nithin H.V,"Modeling of Adders using CMOS and GDI Logic for Multiplier Applications-A VLSI Based Approach", *International Conference on Circuit, Power and Computing Technologies*, 2016.

[9] Amita, NitinSachdeva, "Design and Analysis of Carry Look Ahead Adder Using CMOS Technique", *IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)* e-ISSN: 2278-2834,p- ISSN: 2278-8735.Volume 9, Issue 2, Ver. VII, Mar - Apr. 2014.

[10] Chetana Nagendra, Mary Jane Irwin and Robert Michael Owens, "Area-Time-Power Tradeoffs in Parallel Adders", *IEEE transactions on circuits and systems-11: analog and digital signal processing*, vol. 43, no. 10, October 1996.

[11]"Computation: Definition and Synonyms from Answers.com". *Answers.com*. Archived from the original on 22 February 2009 and retrieved on 26 April 2017.

[12] Mano, M. Morris n, *Digital Logic, and Computer Design*. Prentice-Hall. pp. 119–123. ISBN 0-13-214510-3.

[13] www.circuitstoday.com/ripple-carry-adder

[14] Rosenberger, Gerald B, "Simultaneous Carry Adder". *U.S. Patent 2,966,30.*

