International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 04
February 2018

Design of Proficient Adders for Multipliers using CMOS and GDI Techniques

Kanala V A Sunil Kumar Reddy; Sadula Velugonda Reddy ; Dr. P Prasanna Murali Krishna
sunilreddy.kanala@gmail.com[1] velugonda.sadula90@gmail.com[2] pprasannamurali@gmail.com[3]
[1] Pg Scholar, VLSI, Krishnachaitanya Institute of Technology & Sciences, Markapur, Andhra Pradesh.
[2] Assistant Professor, Dept Of ECE, Krishnachaitanya Institute Of Technology & Sciences, Markapur,Andhra Pradesh.
[3] Professor & HOD, Dept Of Ece, Krishnachaitanya Institute Of Technology & Sciences, Markapur,Andhra Pradesh.

Abstract: *High speed and low power adder circuits are highly demanded in VLSI design. As the days pass by, the advancement in the innovation is becoming more faster. In this paper, a new approach for high speed and low power adder design, optimization of Area, Power and Delay is proposed. The proposed work deals with the construction of high speed adder circuits. Design and modeling of various adders like Ripple Carry Adder, Kogge Stone Adder, and Brent Kung Adder is done by using CMOS and GDI logic and comparative analysis is coated.*

Keywords— *Ripple Carry Adder (RCA), Kogge Stone Adder (KSA), Brunt Kung Adder (BKA), Cadence, Multiplier using adder, Gate count, number of transistors, Power, and Delay.*

I. INTRODUCTION

ADDERS are a key building block in arithmetic and logic units (ALUs) and hence increasing their speed and reducing their power/energy consumption strongly affect the speed and power consumption of processors. There are many works on the subject of optimizing the speed and power of these units, which have been reported. Obviously, it is highly desirable to achieve higher speeds at low-power/energy consumptions, which is a challenge for the designers of general purpose processors. One of the effective techniques to lower the power consumption of digital circuits is to reduce the supply voltage due to quadratic dependence of the switching energy on the voltage. Moreover, the sub threshold current, which is the main leakage component in OFF devices, has an exponential dependence on the supply voltage level through the drain-induced barrier lowering effect. Depending on the amount of the supply voltage reduction, the operation of ON devices may reside in the super threshold, near-threshold, or sub threshold regions. Working in the super threshold region provides us with lower delay and higher switching and leakage powers compared with the near/sub threshold regions. In the sub threshold region, the logic gate delay and leakage power exhibit exponential dependences on the supply and threshold voltages. Moreover, these voltages are (potentially) subject to process and environmental variations in the nano scale technologies.

The variations increase uncertainties in the aforesaid performance parameters. In addition, the small sub threshold current causes a large delay for the circuits operating in the sub threshold region. Recently, the near-threshold region has been considered as a region that provides a more desirable tradeoff point between delay and power dissipation compared with that of the sub threshold one, because it results in lower delay compared with the sub threshold region and significantly lowers switching and leakage powers compared with the super threshold region. In addition, near-threshold operation, which uses supply voltage levels near the threshold voltage of transistors, suffers considerably less from the process and environmental variations compared with the sub threshold region. The dependence of the power (and performance) on the supply voltage has been the motivation for design of circuits with the feature of dynamic voltage and

frequency scaling. In these circuits, to reduce the energy consumption, the system may change the voltage (and frequency) of the circuit based on the workload requirement. For these systems, the circuit should be able to operate under a wide range of supply voltage levels. Of course, achieving higher speeds at lower supply voltages for the computational blocks, with the adder as one the main components, could be crucial in the design of high-speed, yet energy efficient, processors. In addition to the knob of the supply voltage, one may choose between different adder structures/families for optimizing power and speed. There are many adder families with different delays, power consumptions, and area usages. Examples include ripple carry adder (RCA), carry increment adder (CIA), carry skip adder (CSKA), carry select adder (CSLA), and parallel prefix adders (PPAs). The descriptions of each of these adder architectures along with their characteristics may be found.

The RCA has the simplest structure with the smallest area and power consumption but with the worst critical path delay. In the CSLA, the speed, power consumption, and area usages are considerably larger than those of the RCA. The PPAs, which are also called carry look-ahead adders, exploit direct parallel prefix structures to generate the carry as fast as possible. There are different types of the parallel prefix algorithms that lead to different PPA structures with different performances. As an example, the Kogge–Stone adder (KSA) is one of the fastest structures but results in large power consumption and area usage. It should be noted that the structure complexities of PPAs are more than those of other adder schemes. The CSKA, which is an efficient adder in terms of power consumption and area usage, was introduced. The critical path delay of the CSKA is much smaller than the one in the RCA, whereas its area and power consumption are similar to those of the RCA. In addition, the power-delay product (PDP) of the CSKA is smaller than those of the CSLA and PPA structures. In addition, due to the small number of transistors, the CSKA benefits from relatively short wiring lengths as well as a regular and simple layout. The comparatively lower speed of this adder structure, however, limits its use for high-speed applications.

I. RELATED WORK

Design of Fast and Efficient 1-bit Full Adder and its Performance Analysis.

The most fundamental computational process encountered in digital system is binary addition, to accomplish this process binary adders are used, half adder and full adders are most often used to carry out binary addition. This paper presents a comparative analysis of design of 1bit full adder using conventional techniques and new techniques, the design and simulation of 1-bit full adder is performed on Cadence Design Suit 6.1.5 using Virtuoso and ADE environment at GPDK 45nm technology with a unvaried width and length of PMOS and NMOS devices. The paper gives a compression of various design of 1 bit full adder with respect to number of transistors/ gate count, Delay, Power and Power Delay Product.

Comparative Analysis of 8-bit Adders for Embedded Applications

Digital computations and calculations is involved in every embedded and processing device, these devices has arithmetic logic unit or a special block to perform a desired operation, Addition can be one such operation, adders are most important and fundamental block used for Addition, Subtraction, Multiplication, Division, Address generation and so on, design and selection of adders for a embedded application plays a very important role, this paper presents a comparative survey, study and analysis of four different adders like Ripple Carry Adder, Carry Skip Adder, Carry Look ahead Adder, and Kogge Stone Adder, with precession of 8-bit using Verilog HDL coding and the tabulation of performance metrics as delay and area is performed. Given an embedded application may use adder block as its core operation for other sub operations this adder block may be chosen based on its performance and application where it is used.

I. CMOS and GDI STRUCTURE

Complementary metal–oxide semiconductor (CMOS) is a technology for constructing integrated

circuits. CMOS technology is used in microcontroller, microprocessor, static RAM and other digital logic circuits. CMOS technology is also used for several analog circuits such as image sensors (CMOS sensors), data converters, and highly integrated transceivers for many types of communication. Two CMOS is also sometimes referred to as complementary-symmetry metal–oxide–semiconductor (or COS-MOS). The words "complementary-symmetry" refer to the fact that the typical digital design style with CMOS uses complementary and symmetrical pairs of p-type and n-type metal oxide field effect transistors (MOSFETs) for logic functions. Important characteristics of CMOS devices are high noise immunity and low static power consumption. CMOS circuits are constructed in such a way that all PMOS transistors should have either an input from the voltage source or from another PMOS transistor. Similarly, all NMOS transistors should have either an input from ground or from another NMOS transistor. The composition of a PMOS transistor provides low resistance between its source and drain contacts when a low gate voltage is applied and high resistance when a high gate voltage is applied. But on the other hand, the composition of an NMOS transistor provides high resistance between source and drain when a low gate voltage is applied and low resistance when a high gate voltage is applied. CMOS achieves current reduction by complementing every NMOSFET with a PMOSFET and connecting both gates and both drains together. A high voltage on the gates will result to the condition that NMOSFET will conduct and the PMOSFET will not conduct while a low voltage on the gates causes the reverse. This technique greatly reduces power consumption and heat production. However, during the switching time both MOSFETs conduct briefly as the gate voltage goes from one state to another. GDI resembles standard CMOS inverter cell with the only difference is that CMOS inverter has only one input and two supply voltages VDD and VSS. But GDI cell can have three inputs G (common gate input of NMOS and PMOS), VDD supply is replaced by P (input to the source/drain of PMOS), VSS supply is replaced by N (input to the source/drain of NMOS).

I. DESIGN OF ADDERS

Ripple Carry Adder: The RCA block can be constructed by cascading full adder blocks in series. It is possible to create a logical circuit using multiple full adder to add N-bit numbers. Each full adder inputs a Cin, which is the Cout of the previous adder. This kind of adder is RCA, since each carry bit "ripples" to the next full adder. For an n bit adder it requires n 1 bit full adder. Drawbacks of Ripple Carry Adder: The RCA is relatively slow since each full adder must wait for the carry bit to be calculated from the previous full adder.

Equation of Ripple Carry Adder

The corresponding boolean expressions are given here to construct a ripple carry adder. In the half adder circuit the sum and carry bits are defined as

$$\text{Sum} = A \oplus B$$
$$\text{Carry} = AB$$

In the full adder circuit the Sum and Carry output is defined by inputs A, B and Carry'in as

$$\text{Sum} = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$$
$$\text{Carry} = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$$

Having these we could design the circuit. But, we first check to see if there are any logically equivalent statements that would lead to a more structured equivalent circuit. With a little algebraic manipulation, one can see that

$$\text{Sum} = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$$
$$= (\bar{A}B + A\bar{B})\bar{C} + (AB + \bar{A}\bar{B})C$$
$$C = (A \oplus B)\bar{C} + (\bar{A} \oplus \bar{B})C$$
$$= A \oplus B \oplus C$$
$$\text{Carry} = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$$
$$= AB + (A\bar{B} + \bar{A}B)C = AB + (A \oplus B)C$$

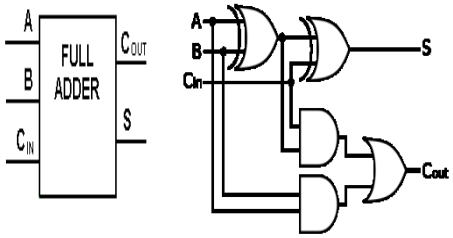


Fig 1: Block diagram and Schematic of 1-bit Full Adder

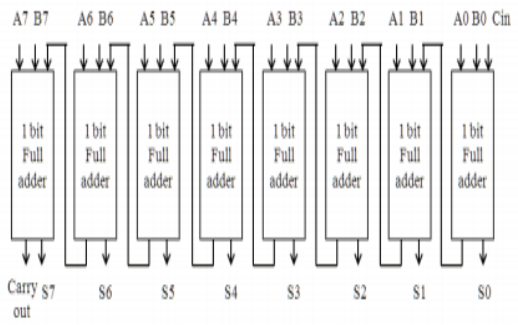


Figure 2 : Block diagram of 8-bit Ripple Carry Adder

Kogge Stone Adder

These adders involve the execution of an operation in parallel. This is done by segmentation the operation in smaller pieces which are computed in parallel. The outcome of the operation depends on the initial inputs. Parallel Prefix Adder (PPA) is equivalent to carry look ahead adder (CLA).A Carry look ahead adder is a type of adder used in digital logic. CLA is designed to overcome the latency introduced by repelling effect of carry bits in RCA.A CLA improves speed by reducing carry bits. It calculates one or more carry bits before the sum, which reduces the wait time to calculate the result of larger bit value.

CLA uses the concept of generating (G) and propagating (P) carries. The two differ in the way their carry generation block is implemented. Equations used to generate carry in CLA are given by:

$$Ci = Gi{-}1\ OR\ (Pi{-}1\ AND\ Ci{-}1\ ) \qquad (1)$$

The main advantage of PPA is the carry reduces the number of logic levels by essentially generating the carries in parallel.PPA fastest adder with focus on design time and is the choice for high performance adder in industry.

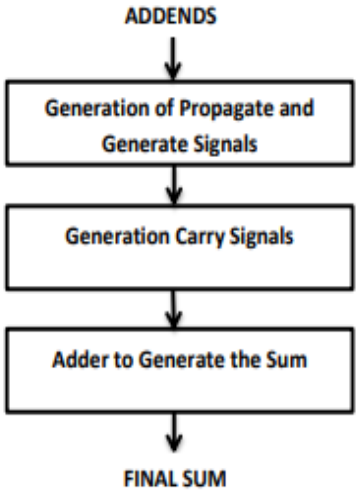


Fig 3 Architecture of Parallel Prefix Adder

Fig.3. shows the architecture of Parallel prefix adder. KSA is a parallel Prefix Adder. It is considered as fastest and is widely used in industry for high performance arithmetic circuits. KSA employs the 3-stage structure of the CLA adder, the improvement is in the carry generation stage which is the most intensive one. In KSA carries are computed fast by computing the carries in parallel. This is often desirable to use an adder with good timing, area and efficiency trade off. The carry computation method leads to speed up the overall operation significantly. This reduces the area and increase the speed.

KSA is a parallel prefix form carry look ahead adder. It generates carry in O (log n) time and is widely considered as the fastest adder and is widely used in the industry for high performance arithmetic circuits.

In KSA, carries are computed fast by computing them in parallel at the cost of increased area. It has three processing stages for calculating the sum bits. Working of KSA:

i. Pre-processing: In this step the computation of generate and propagate signals corresponding too each pair of bits in A and B. These signals are given by the logic equations below:

Pi = Ai xor Bi (3)

Gi = Ai and Bi (4)

ii. Carry look ahead network: This block differentiates KSA from other adders and is the main force behind its high performance.

This step involves computation of carries corresponding to each bit. It uses group propagate and generate as intermediate signals which are given by the logic equations below:

Black Cell. The black cell takes two pairs of generate and propagate signals (Gi, Pi) and (Gj, Pj) as input and computes a pair of generate and propagate signals (G, P) as output

$$G = Gi \text{ OR } (Pi \text{ AND } Pj) \quad (5)$$

$$P = Pi \text{ AND } Pj \quad (6)$$

Grey Cell. The gray cell takes two pairs of generate and propagate signals (Gi, Pi) and (Gj, Pj) as inputs and Computes a generate signal G as output

$$G = Gi \text{ OR } (Pi \text{ AND } Pj) \quad (7)$$

iii. Post processing: This is the final step and is common to all adders of this family (carry look ahead). It involves computation of sum bits. Sum bits are computed by the logic given below:

$$Si = pi \text{ xor } Ci\text{-}1.$$

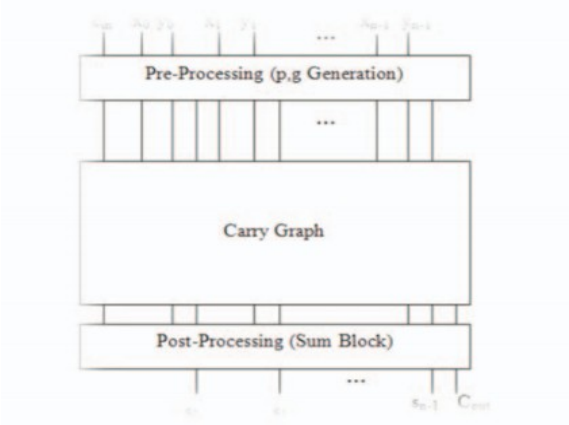


Figure 4: Computational stages of Parallel Prefix Adder.

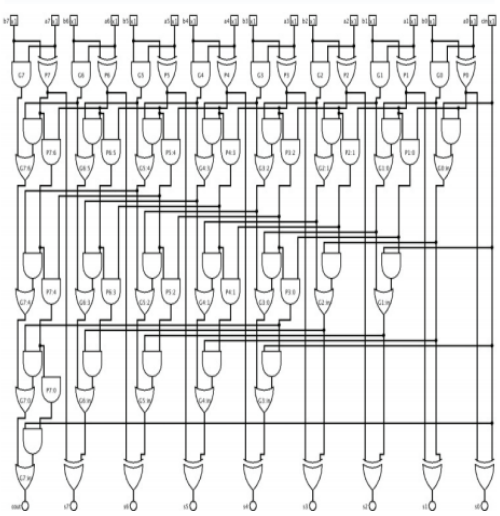


Figure 5: Schematic of 8-bit Kogge Stone Adder

Brent Kung Adder

The type of structure of any adder greatly affects the speed of the circuit. The logarithmic structure is considered to be one of the fastest structures. The logarithmic concept is used to combine its operands in a tree- like fashion. The logarithmic delay is obtained by restructuring the look-ahead adder. The restructuring is dependant on the associative property, and the delay is obtained to be equal to (log2N) t, where 'N' is the number of input bits to the adder and t is the propagation delay time. Hence, for a 16-bit structure, the logarithmic adder has a delay equal to '4t', while for a simple ripple carry adder the delay is given by (N-1)t and is equal to '15t' for 'N' and 't' being the number of input bits and the delay time, respectively. Hence it is seen that this structure greatly reduces the delay, and would be especially beneficial for a structure with large number of inputs. This advantage is, however, obtained at the expense of large area and a complex structure.

In the following section, a structure known as the Brent Kung Structure, which was first proposed by Brent and Kung in 1982 and which uses the logarithmic concept, is discussed. This structure used an operator known as the dot (·) operator, which is explained in the architecture, for its basic blocks.

Brent Kung Architecture

In order to approach the structure known as the Brent Kung Structure, which uses the logarithmic concept, the entire architecture is easily understood by dividing the system into three separate stages:

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 04
February 2018

1. Generate/Propagate Generation
2. The Dot (·) Operation
3. Sum generation

Generate/Propagate Generation

If the inputs to the adder are given by the signals A and B, then the generate and propagate signals are obtained according to the following equations.

G = A.B (3.1) P = AÅ. B (3.2)

The Dot (·) Operation =

The most important building block in the Brent Kung Structure is the dot (·) operator. The basic inputs to this structure are the generate and propagate signals generated in the previous stage. The · operator is a function that takes in two sets of inputs-- (g, p) and (g', p')-- and generates a set of outputs-- (g + pg', pp').

These building blocks are used for the generation of the carry signals in the structure. For the generation of the carry signals, the carry for the kth bit from the carry look-ahead concept is given by

Co,k = Gk +Pk(Gk-1 +Pk-1 +P k-1 (…+P1(G0 +P0 Ci,0))) (3.3)

Using the dot operator explained above the Equation 3.3 can be written for the different carry signals as

Co,0 = G0 +P Ci,0 = a (G0,P0) Co,1 = G1 + G0 P1 = a ((G1 , P1)·(G0, P0)) …… C0,k = a ((Gk,Pk)·(Gk-1,Pk-1)·…·(G0,P0)) (3.4)

where a is a function defined in order to access all the tuples. All the carry signals generated at different stages in the structure. In the structure, two binary tree structure are represented -- the forward and the reverse trees. The forward binary tree alone is not sufficient for the generation of all the carry signals. It can only generate the signals shown as Co,0,Co,1, Co,3 and Co,7. The remaining carry signals are generated by the reverse binary tree.

Sum Generation.

The final stage in this architecture is the sum generation stage. The sum is given by

S= AÅ BÅ C (3.5)

where A and B are the input signals, and C is the carry signal. The carry is obtained from the dot operator stage discussed earlier, and the exclusive of A and B is actually the propagate signal itself. Hence the sum 'S' can finally be represented and realized as

S = P Å C (3.6)

Brent Kung Parallel Prefix Adder has a low fan-out from each prefix cell but has a long critical path and is not capable of extremely high speed addition. In spite of that, this adder proposed as an optimized and regular design of a parallel adder that addresses the problems of connecting gates in a way to minimize chip area. Accordingly, it considered as one of the better tree adders for minimizing wiring tracks, fan out and gate count and used as a basis for many other networks.

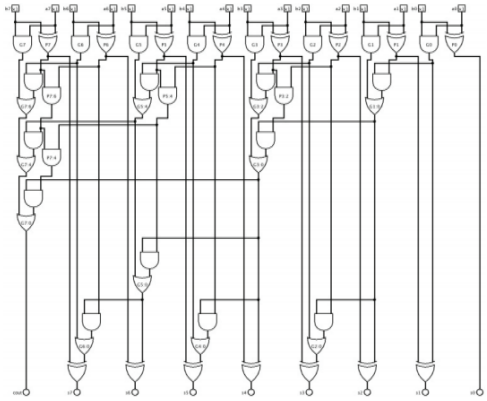
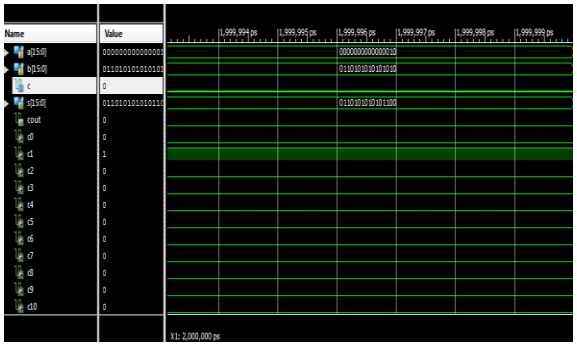


Figure 6: Schematic of 8-bit Brent Kung Adder

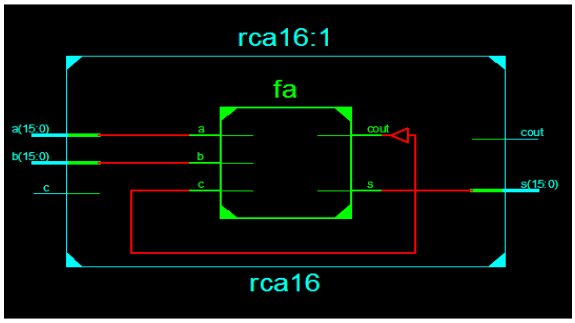
IV. EXPERIMENTAL RESULTS

Ripple Carry Adder

Simulation:

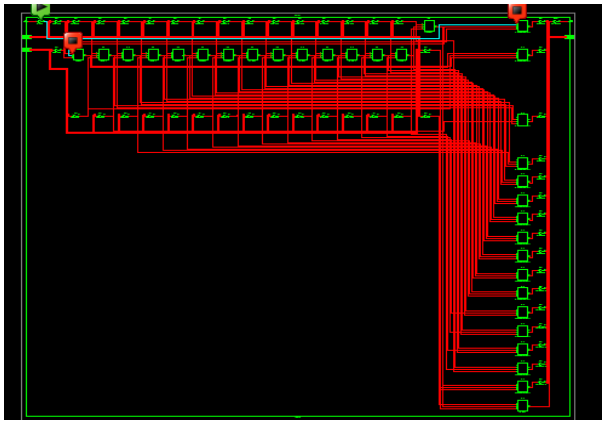


RTL Schematic:



International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 04
February 2018

Technology Schematic:



Design Summary:

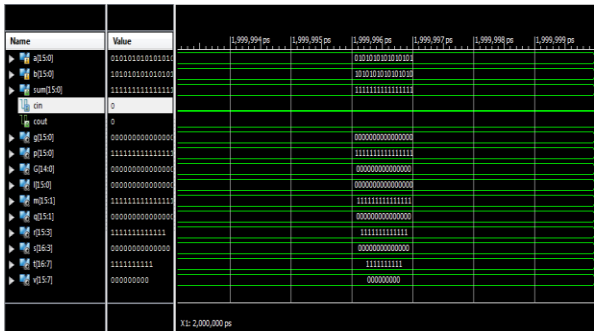
Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	18	4656	0%
Number of 4 input LUTs	32	9312	0%
Number of bonded IOBs	50	232	21%

Timing Summary:

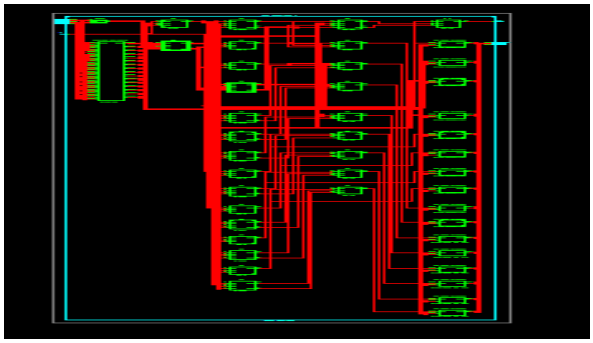
```
Timing Summary:
---------------
Speed Grade: -5

   Minimum period: No path found
   Minimum input arrival time before clock: No path found
   Maximum output required time after clock: No path found
   Maximum combinational path delay: 21.690ns
```

Kogge Stone Adder

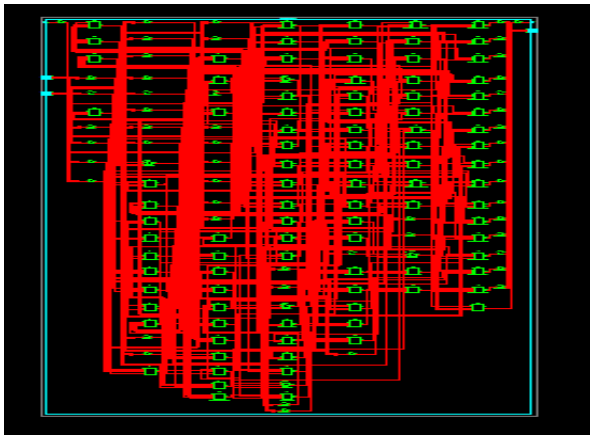
Simulation:



RTL Schematic:



Technology Schematic:



Design Summary:

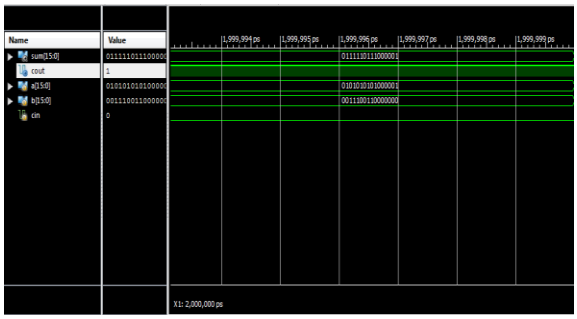
Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	54	4656	1%
Number of 4 input LUTs	94	9312	1%
Number of bonded IOBs	50	232	21%

Timing Summary:

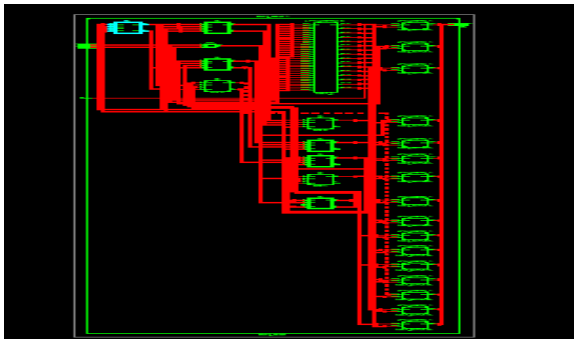
```
Timing Summary:
---------------
Speed Grade: -5

   Minimum period: No path found
   Minimum input arrival time before clock: No path found
   Maximum output required time after clock: No path found
   Maximum combinational path delay: 12.499ns
```

Brent Kung Adder

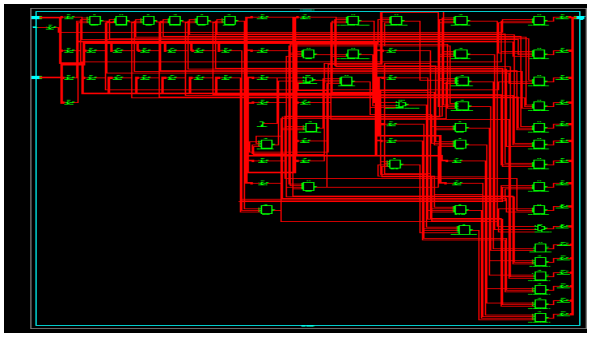
Simulation:



RTL Schematic.:



Technology Schematic:



Design Summary:

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	23	4656	0%
Number of 4 input LUTs	39	9312	0%
Number of bonded IOBs	49	232	21%

Timing Summary:

```
Timing Summary:
---------------
Speed Grade: -5

   Minimum period: No path found
   Minimum input arrival time before clock: No path found
   Maximum output required time after clock: No path found
   Maximum combinational path delay: 17.328ns
```

Comparison Table

TYPE OF ADDER	AREA			DELAY(ns)
	SLICES	LUTS	IOBS	
RCA	18	32	50	21.690ns
Kogge Stone Adder	54	94	50	12.499ns
Brent Kung Adder	23	39	48	17.328ns

CONCLUSION

Adders are core and essential block in many modules which involves computation and adders play a vital role in the design of multipliers using adder based logic, hence the design and implementation of the adders is a prime concern, In this paper we have designed, modeled the different adders like RCA, KSA, BKA using different design style and coated comparative results obtained. From the results it is clear that the adders designed using GDI design style give less delay and consumes less number of gate count, CMOS design style give less power consumption, as these the performance parameters are prime concerned while designing a module. Hence the choice has to be made and as per the desire of the designer and the prime concern of performance measure needed at that point in time.

Future Scope

These adders can be used in different multipliers in the partial product reduction which helps in fast addition and increases speed of the multipliers.

REFERENCES

[1] Mr. Kunjan D. Shinde, Mrs. Jayashree C. N."Modeling, Design and Performance Analysis of Various 8-bit Adders for Embedded Applications", Proceedings of the ELSEVIER in International Conference on Emerging Research in Computing, Information, Communication and Applications (ERCICA-14)" at Nitte Meenakshi Institute of Technology, Bangalore, Karnataka, India. 01st and

02nd August 2014, ISBN 9789351072621, pp. 823 to 831.

[2] Mr. Kunjan D. Shinde, Mrs. Jayashree C. N. "Design of Fast and Efficient 1-bit Full Adder and its Performance Analysis",Proceedings of the IEEE International Conference on "Control, Instrumentation, Communication and Computational Technologies (ICCICCT-2014) " at Noorul Islam University, Kumaracoil, Kanyakumari, Tamilnadu, India. 10th and 11th July 2014, IEEE Digital Xplore ISBN 978-1-4799-4191-9, pp. 1275 to 1279.

[3] Mr. Kunjan D. Shinde, Mrs. Jayashree C. N. "Comparative Analysis of 8-bit Adders for Embedded Applications", Proceedings of the IJERT in National Conference on "Real Time System (NCRTS-14)" at City Engineering College Bangalore, Karnataka, India. 9th and 10th May 2014, ISSN- 2278-0181 pp. 682 to 687, NCRTS'14 Conference Proceedings, IJERT publication

[4] Mr.Nurdiani Zamhari, Mr. Peter Voon, Miss/Mrs.KuryatiKipli, Mr.Kho Lee Chin, Mr.MaimunHujaHusin, "Comparison of Parallel Prefix Adder (PPA)",proceedings of the World Congress on Engineering 2012 Vol IIWCE 2012, July 4 - 6, 2012, London, U.K.

[5] Anas Zainal Abidin, Syed Abdul Mutalid Al Junid, KhairulKhaiziMohd Sharif,"A survey on 4 bit Brent kung PPA simulation study usingsilvaco EDA Tools".

[6] Avinash shrivastava,Chandrahas Sahu, "Performance Analysis of PPA based on FPGA, International Journal of engineering trends and technology (IJETT)-volume 21,number 6-march 2015.