International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 04 Issue: 03 | Mar -2020 ISSN: 2582-3930

# Implementation of N-bit Kogge Stone Adder for High Speed ALU

**N.Vidhya[1], Dr.B.Jaishankar[2]**

**[1]PG Scholar, Dept. of ECE, KPR Institute of Engineering and Technology, Coimbatore, India.**

**[2]Associate Professor, Dept. of ECE, KPR Institute of Engineering and Technology, Coimbatore, India.**

***Abstract -*** In Digital systems Design adder is an important component and it is used in multiple blocks of its architecture. So, speed of operation is the most important constraint. In digital adders, the speed of addition is limited by the time required to propagate a carry through the adder. The basic idea of this work is to use Kogge-stone adder cell (pre-fix adders) instead of RCA to achieve High speed, lower area and power consumption.

Kogge-Stone Adder (KSA) is a parallel prefix form carry look ahead adder (CLA). It generates carry in O (logn) 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. A high-speed n-bit Kogge-Stone adder (KSA) has been implemented. The adders are designed using Verilog and synthesized using front-end tool including analysis for performance, power, and area. Improving the speed of the n-bit (4, 8, 16, 32, 64,128bits….) Kogge-Stone adder with the single module by using the **GENERATE** concept .Optimizing the speed of an n-bit KSA without any dynamic power dissipation using the **GENERATE** concept. Thus increases overall speed of an ALU. Once detecting the particular approaches for input, output, main block and different modules, the Verilog descriptions are run through a Xilinx ISE 10.1 simulator, followed by the timing analysis for the validation, functionality and performance of the designated design.

**Keywords—** ***Serial Adder, Parallel Prefix Adder, Ripple Carry Adder, Carry Look ahead Adder, Kogge Stone Adder, High Speed ALU***

## I. INTRODUCTION

An **adder** is a digital circuit that performs addition of numbers. In many computers and other kinds of processors adders are used in the arithmetic logic units or ALU. They are also used in other parts of the processor, where they are used to calculate addresses, table indices, increment and decrement operators, and similar operations. The serial binary adder or bit-serial adder is a digital circuit that performs binary addition bit by bit. The serial full adder has three single-bit inputs for the numbers to be added and the carry in. There are two single-bit outputs for the sum and carry out. The carry-in signal is the previously calculated carry-out signal. The addition is performed by adding each bit, lowest to highest, one per clock cycle. **Example**: Ripple Carry Adder (RCA).

RCA is serial adder and it has propagation delay problem. With increase in bits, delay also increases simultaneously. Hence parallel adders (parallel prefix adders) are preferred.RCA is replaced by Kogge Stone Adder (KSA) in order to increase the speed with reduced area. Parallel Prefix adders have been one of the most notable among several designs proposed in the past. The advantage of utilizing the flexibility in implementing the three structures based upon throughput requirements. Due to continuing integrating intensity and the growing needs of portable devices, low-power and high-performance designs are of prime importance.

A Parallel Adder is a digital circuit capable of finding the arithmetic sum of two binary numbers that is greater than one bit in length by operating on corresponding pairs of bits in parallel. It consists of **full adders connected in a chain** where the output carry from each full adder is connected to the carry input of the next higher order full adder in the chain. **A n bit parallel adder requires n full adders to perform the operation.** So, for the two-bit number, two adders are needed while for four-bit number, four adders are needed and so on. Parallel adders normally incorporate carry look ahead logic to ensure that carry propagation between subsequent stages of addition does not limit addition speed.

**Example**: Carry Look ahead Adder (CLA).

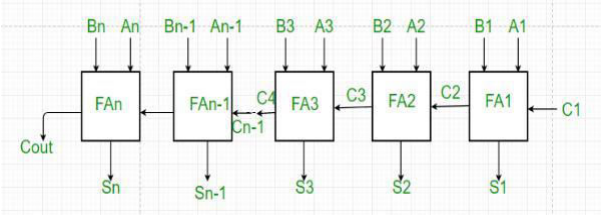


**Figure1.** Parallel Adder (CLA)



 

## II. PARALLEL PREFIX ADDER

The Parallel Prefix Adder (PPA) is one of the fastest types of adders that had been created and developed. VLSI Integer adders find applications in Arithmetic and Logic units (ALU"s), microprocessors and memory addressing units. Speed of the adder often decides the minimum clock cycle time. The need for a Parallel Prefix adder is that it is primarily fast when compared with ripple carry adder. Parallel Prefix adders have been established as the most efficient circuits for binary addition. Their regular structure and fast performance make them particularly attractive for VLSI implementation. The classical parallel prefix adder structures presented in the literature over the years optimize for logic depth, area, and fan-out and interconnect count of logic circuits. Parallel Prefix adders (PPA) are family of adders derived from the commonly known carry look ahead adders. The parallel prefix adders are KS adder (Kogge-stone), SKS adder (sparse Kogge-stone), Spanning tree and Brent-kung adder. These adders flexible, used to speed up the binary additions. The advantage of using tree structure form to increase the speed of arithmetic operation. RCA is a serial adder. RCA is used to perform any number of additions. RCA is serial adder and it has propagation delay problem. With increase in bits, delay also increases simultaneously. Hence parallel adders (parallel prefix adders) are preferred.RCA is replaced by Kogge Stone Adder (KSA) in order to increase the speed with reduced area.

Parallel Prefix adders have been one of the most notable among several designs proposed in the past. The advantage of utilizing the flexibility in implementing the three structures based upon throughput requirements. Due to continuing integrating intensity and the growing needs of portable devices, low-power and high-performance designs are of prime importance. 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.

In this paper, design and implementation of optimized N- bit KSA is proposed.

## III. KOGGE-STONE ADDER (KSA)

KSA is a parallel Prefix Adder. It is considered as fastest and is widely used in industry for high performance arithematic 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.

The complete functioning of KSA can be easily comprehended by analyzing it in terms of three following distinct parts.

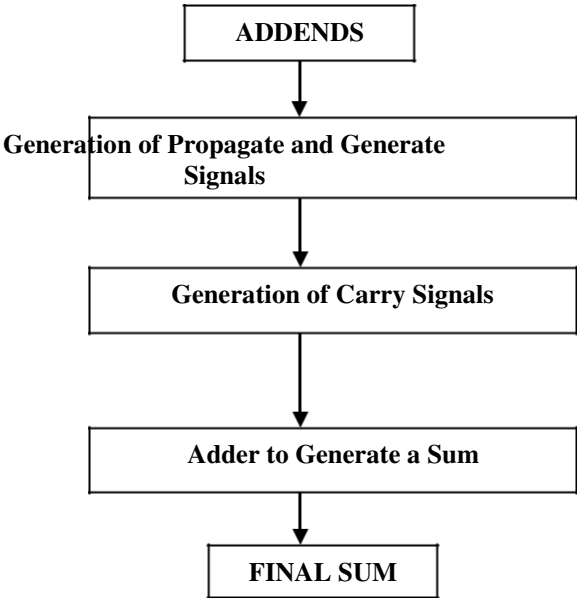


**Figure2.** Architecture of Parallel Prefix Adder

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. The advantage of utilizing the flexibility in implementing the three structures based upon throughput requirements as shown in fig2.

- Pre-processing stage
- Carry generating stage
- Post-processing stage

### PRE-PROCESSING STAGE

This step involves 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:

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

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

### CARRY GENERATING STAGE

Show the 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:

$CP_0 = P_i$ AND $P_j$ …….(3)

$CG_0 = G_i$ OR $P_i$ AND $G_j$ …….(4)

**POST PROCESSING STAGE**

This step is final step and is common to all adders of this family. It involves computations of sum bits. Sum bits are computed by the logic given below:

$S_i = P_i$ XOR $C_{i-1}$ ……..(5)

## IV. KSA ARCHITECTURE

The complete functioning of KSA can be easily comprehended by analyzing it in terms of three distinct parts as discussed in the previous section.

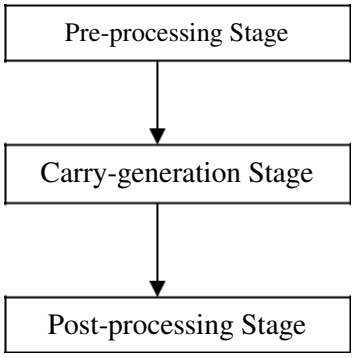


**Figure3.** Architecture of KSA

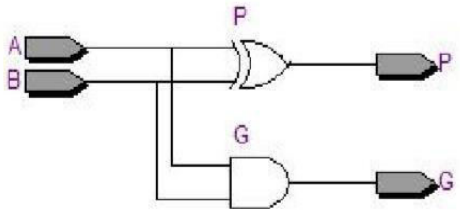


**Figure4.** Architecture of Pre-processing Stage

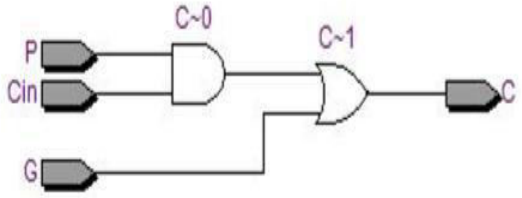


**Figure5.** Architecture of Carry-generating Stage

The above architectures are explained in the previous section.

## V. PROPOSED KSA ARCHITECTURE

Kogge-stone adder is a parallel prefix form of Carry Look-ahead Adder. Kogge-Stone adder can be represented as a parallel prefix graph consisting of carry operator nodes. The time required to generate carry signals in this prefix adder is o (log n). It is the fastest adder with focus on design time and is the common choice for high performance adders in industry. The Kogge-Stone adder concept was developed by Peter M. Kogge and Harold S. Stone. The better performance of Kogge-Stone adder is because of its minimum logic depth and bounded fan-out. On the other side it occupies large silicon area. Hence for 16-bit adder, the height is 4 logic stages. Similarly, for 32-bit, the height is 5 logic stages. The radix of the adder refers to how many results of previous logic stages are considered to generate next stage. The design and analysis of an n-bit KSA was implemented using **GENERATE** concept within a single module. The module was written using the keyword **GENERATE** for all n-bit KSA (up to 128-bits) by changing the value of parameter as 16, 32, 64, 128.

**64 & 128-BIT KOGGE STONE ADDER:**

The processing steps of 64 and 128-bit KSA also same as above mentioned technique. The binary adder is the critical element in most digital circuit designs including digital signal processors (DSP) and microprocessor data path units. As such, extensive research continues to be focused on improving the power delay performance of the adder. In VLSI implementations, parallel prefix adders are known to have the best performance. Binary adders are one of the most essential logic elements within a digital system. Sum of the 64 & 128-bit KS adders are achieved by using only one full adder delay, because all adders are getting the carry at the same time. By using the generate concept, the optimized speed of all n-bit (16, 32, 64,128-bits) KSA was achieved without affecting the area and power. The main advantage of this concept is achieving zero dynamic power dissipation for all n-bit KSA

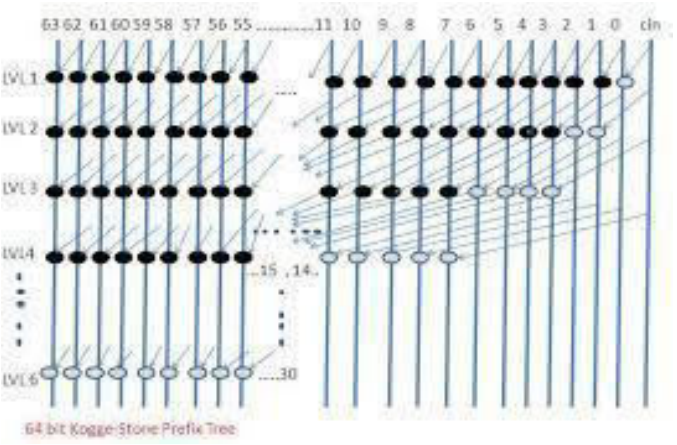


**Figure6.** Architecture of 64-bit KSA



 

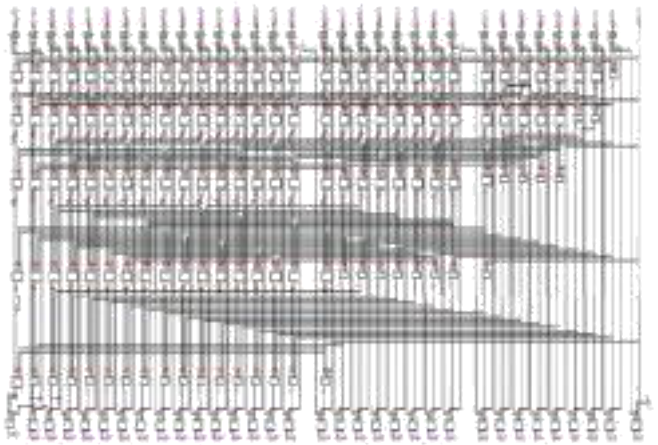

**Figure7.** Architecture of 128-bit KSA

All the above n-bit Kogge-Stone Adders are involved in similar processes. The bit width alone changed by changing the parameter within the single module using generate concept.

## VI. RESULT AND CONCLUSION

Prefix adders have been one of the most notable among several designs proposed in the past. The advantage of utilizing the flexibility in implementing the three structures based upon throughput requirements.PPA fastest adder with focus on design time and is the choice for high performance adder in industry.

The proposed design is simulated using ISE simulator and synthesized using ISE 14.2. The speed of an n-bit (16, 32, 64,128 bits) Kogge-Stone adder (KSA) was improved successfully by using **GENERATE** concept. The optimized speed of n-bit KSA was achieved without dissipating the dynamic power (i.e., the power is constant for all n-bits) for all n-bit (16, 32, 64,128 bits) by using **GENERATE** concept. The performance of the adder was improved without affecting the area and power. The power consumption for Kogge stone adder is 0.042mW for all n-bit KSA. Thus, the power dissipation is static (ideal) for all n-bit KSA and there is no dynamic power dissipation. Finally, the speed of an n-bit KSA was optimized without affecting the power utilization.

**Figure8.** Simulation output of 64-bit KSA

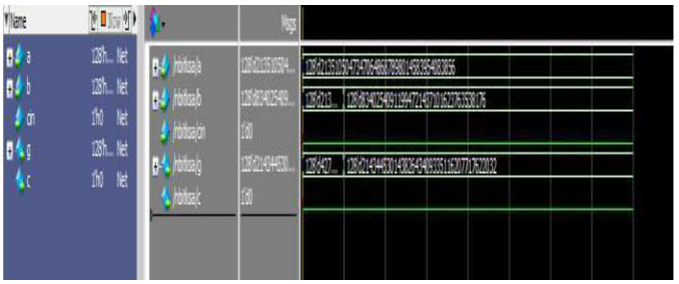

**Figure9.** Simulation output of 128-bit KSA

**TABLE1.** CELL AND AREA UTILIZATION

| ADDER | CELL | CELL AREA |
|---|---|---|
| 32-bit RCA | 50 | 838 |
| 32-bit KSA | 98 | 1024 |

**TABLE2.** POWER UTILIZATION

| ADDER | Dynamic Power(mW) | Static Power(mW) | Total Power(mW) |
|---|---|---|---|
| RCA | 0.15 | 0.32 | 0.47 |
| KSA | 0.00 | 0.042 | 0.042 |

The main advantage of this implementation is, that we have achieved same power dissipation for all number of bits in a KSA and there is no dynamic power dissipation.

**TABLE3.** TIME UTILIZATION

| ADDER | DELAY (ns) |
|---|---|
| 32-bit RCA | 31.219 |
| 32-bit KSA | 6.543 |

The comparison of 32-bit RCA and KSA were done, and the speed of the 32-bit KSA is achieved 42% more than the previously implemented paper and the power dissipation also concluded as very low.

**TABLE4.** TIME UTILIZATION OF KSA

| BITS | DELAY(ns) |
|---|---|
| 16-bit KSA | 3.204 |
| 32-bit KSA | 6.543 |
| 64-bit KSA | 11.414 |
| 128-bit KSA | 11.678 |



 

## VII. REFERENCES

[1] Darjn Esposito, Davide De Caro, "Variable Latency Speculative Han- Carlson Adder", Senior Member, IEEE, Ettore Napoli, Nicola Petra, Member, IEEE, Antonio Giuseppe Maria Strollo, Senior Member, IEEE Transaction circuits and systems—I:Regularpapers,Vol.62,No.5,May2015.

[2] DavidHKHoe,ChrixMatinezandSri JyothsnaVandavalli, "Design and Characterization of Parallel Prefix Adders Using FPGA", System theory (SSST), 2011 IEEE Hard South system.

[3] C.N. Shilpa, Kunjan D shinde, H.V.Nithin , "Design, Implementation and Comparative Analysis of Kogge

[4] Kunjan D. Shinde, 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.

[5] Kunjan D. Shinde, Jayashree C. N. "Modeling, Design and Performance Analysis of Various 8-bit Adders for EmbeddedApplications",Proceedingsofthe 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.

[6] Lee Mei Xiang, Muhammad Mun'im Ahmad Zabidi, AinyHaziyahAwab, Ab Al- Hadi Ab Rahman "VLSI Implementation of a Fast Kogge-Stone Parallel-Prefix Adder" International Postgraduate Conference on Applied Science & Physics 2017

[7] M PrasannaKumar, V Sidhartan, Dr K Gopalkrishnan, "Comparative Analysis of Brent Kung and Kogge Stone Parallel Prefix Adders for their Area, Delay and Power Consumption", Research Paper Engineering, Volume 5, Issue 10, October 2015 ISSN 2249555.

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

[9] P. Chaitanya kumari, R.Nagendra"Design of 32-bit Parallel Prefix Adders" IOSR Journal of Electronics and Communication Engineering (IOSR-JECE) e-ISSN: 2278- 2834,p- ISSN: 2278-8735.Volume 6, Issue 1 (May. - Jun. 2013)

[10] R. P. Brent and H. T. Kung, "A regular layout for parallel adders," IEEETrans.Computer,vol.C-31, no.3, pp.260–264, Mar.1982.

## AUTHOR'S PROFILE

**N.Vidhya** is presently pursuing Master's degree in VLSI Design at KPR Institute of Engineering and Technology, Coimbatore. She has interested in Digital Design and VLSI System Design.

**Dr.B.Jaishankar** is presently working as an Associate Professor in Dept. of ECE at KPR Institute of Engineering and Technology, Coimbatore. His research areas are VLSI system design.