IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

# Design of efficient adder for high speed

**Sarita Shambharkar**[mtech scholar], **Prof. Archana khandait** [Guide]

Priyadarshini collage of engineering (MS), Nagpur, Maharashtra, India

## ABSTRACT

**The addition of two n bit number operations in a digital circuit is used to reduce the circuit's intricacy, and it is an operation. In this paper, we design a circuit for designing a high-speed efficient adder. The adoption of an adder with the required qualities ensures the circuit's smooth running. The adders that have been analyzed are 32-bit adders that have been generated and simulated using Xilinx synthesis tools. The results of synthesis reports and circuit simulations aid in the discovery of various attributes. Area consumed power and speed may be determined for the analyze of the device. On the basis of power and area, the ripple carry adder, carry skip adder, and kogge stone adder have been compared in this study.**

**Keyword: Ripple carry adder, carry skip adder, kogge stone adder, Xilinx index terms**

## I. INTRODUCTION

In many computers arithmetic algorithms, adders play a critical role. Not just in computers, but also in processors for various activities, one of which is the incrementing of the program counter. It is practically a need for most cognitive programmers, and it is also thought to be pliant. Adders reduce the number of transistors in any circuit. The basic building blocks of most digital systems are adders. Adders have become a critical component in the effective implementation of arithmetic units. Adders are utilized not just in the arithmetic logic unit, but also in other parts of the processor in many applications.

Although adders are critical, the choice of adder varies from programmer to programmer based on speed, power dissipation, and area use. Any digital circuit must meet the following requirements low power consumption low power dissipation small space high speed. Because a single adder cannot have all of the above characteristics, certain adders take precedence over others depending on the user's needs. The comparison of several 32-bit adders has been carried out carefully to choose adders appropriate and not difficult, various adder are used in this paper that is RCA, CSA and KSA. Area and power limits are among the elements that must be taken into account., the best adder is chosen from their operation. The primary aim of fast-growing technologies is to speed up, However, improving the device's speed alone is insufficient; the device's size is also a consideration of concern that is to be reduced to an optimum Power is another key issue, in addition to mobility. The different adders are efficient in various types of work platform they are ripple carry adder, carry skip adder, among the kogge stone adder is the more efficient adder with the high speed. In this work, the following adders have been compared:

1) Ripple carry adder.

2) Carry skip adder.

3) Kogge stone adder.

## II. METHODOLOGY

In this section, the implementation details of ripple carry adder, carry skip adder and kogge stone adder is presented to perform 32-bit operation.

### A. Ripple carry adder

The Ripple Carry Adder operates in phases. Each complete adder receives the carry-in as an input and outputs the carry-out and sum bit. A full adder's carry-out acts as the carry-in for the next most significant full adder. When carry-in is made accessible to the full adder, the full adder is activated. It starts working when the complete adder has been engaged.

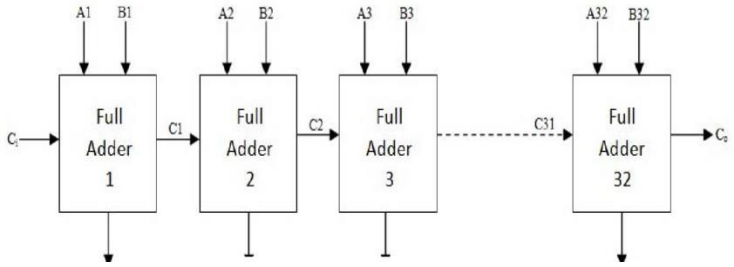



Fig:1 32 bit Ripple Carry Adder

A ripple-carry adder's layout is simple, allowing for quick design time; nevertheless, the ripple-carry adder is slow since each full adder must wait for the carry bit to be calculated from the previous full adder. So there are 32 full adders in a 32-bit ripple-carry adder, the critical path (worst case) runtime is 3 (from input to carry in first adder)+31(for carry propagation in next adder) is,

T_{{CRA}(n)=T_{HA}+(n-1).T_{c}}+T_{s}}=T_{FA}}+(n-1).T_{c}=3D+(n-1) .2D=(2n+1).D.

### B. Carry skip Adder

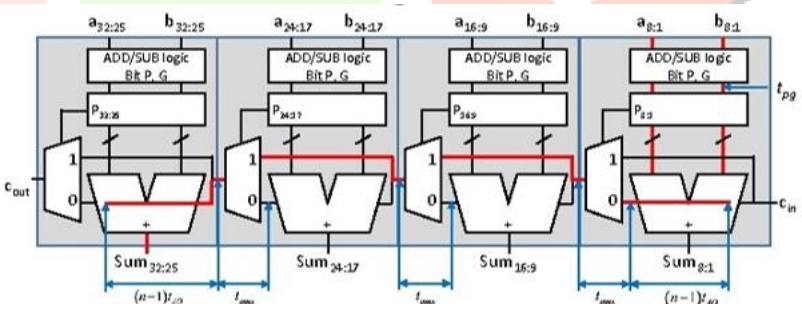



Fig:2 32 bit Carry skip adder

Figure 2 shows how the 32-bit adder is separated into four adder blocks. In the final CS4 block, carry-select adders were used, which substantially increased the hardware. The 32-bit carry-skip adder design presented by uses a combination of RCAs together with carry-skip group generate-propagate logic (SKIP), carry-generate logic (CG), and SKIP logic (PG). The total adder is divided into a number of blocks with varying widths. Both the carry generation and Each block is further divided into sub blocks. In a recursive manner, a sub block can contain additional levels of sub blocks. A number of variable width RCAs makes up the lowest-level sub block.

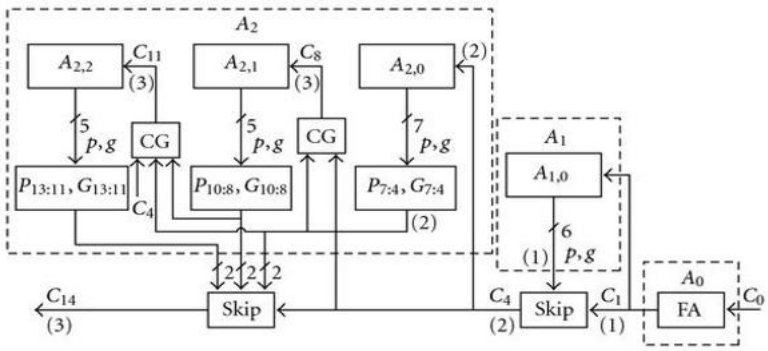



Fig:3 Block schematics for first three blocks of adder

The 32-bit adder is divided into four blocks. A block diagram of the first three blocks (Ao, A1, and A2) is shown in Figure 2. The first block Ao (LSB) is a full adder by itself. The carry from the first block C1 is fed into the second block A1 and the skip logic though too. Each complete adder's generate and propagate functions (p,g) are generated in one unit time, where one unit time has determined as the delay of a complex CMOS gate connects the output node to any supply rail, with at most three transistors linked in series. In Figure 2 indicate the number of unit delays in signal arrival times at the relevant signal leads. Since the delay of a complex CMOS gate is proportional to its stack height, the stack height in our design is limited to three. This means that any series connected route can only have a maximum of three transistors (NMOS or PMOS). This means that the maximum number of inputs to the carry-skip logic is limited to (7). When the generate-propagate outputs are used for group generation and propagation (3) outputs, although, a stack height of 3 in the CMOS implementation allows for a 4-bit RCA.[3] In any series connected path, the maximum number of transistors (NMOS or PMOS) is (3).

C. Kogge stone adder

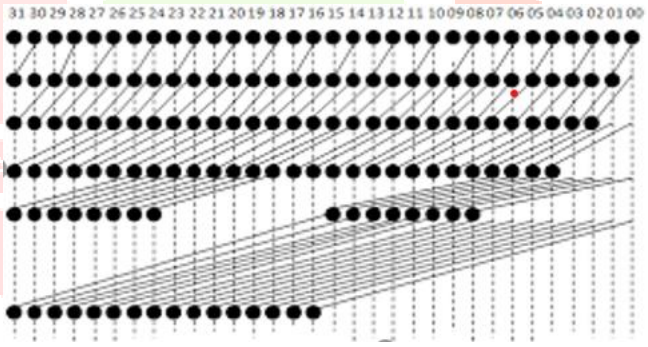



Fig:4 32-bit kogge stone adder

The design of a 32-bit KSA design is shown in the figure above. There are five stages to this design. The calculation of Propagate and Generate signals using full adders with carry input that process included in first and second stage. The third and fourth stages included the generation of carry signals by using values of Propagate and Generate. The fifth stage includes the computation of sum bits based on P and carry generation values with efficient addition operations, KSA plays a significant role and it reflects the prefix form of Carry Look further. Adder is a type of adder (CLA)Also, this type of PPA entirely decreases the delay time in design to generate the carry signals. As an outcome, for efficient arithmetic operations, this KSA is extensively implemented in DSP (Digital Signal Processing) laboratories and control system companies.

KSA's operation can be easily appreciated by splitting into three main parts:
1.Pre processing
This step includes computing the signals to generate and propagate with each pair of bits in A and B. The logic equations below provide these signals:
$p_i = A_i$ *xor* $B_i$ $g_i = A_i$ *and* $B_i$ [5]
2.Carry look ahead network

This block differentiates KSA apart from other adders and is responsible for its excellent performance. This step involves computing the carries that correspond to each bit. It utilizes group propagate and generate as intermediary signals, as defined by the following logic equations:

$P_{i:j} = P_{i:k+1}$ *and* $P_{k:j}$

$G_{i:j} = G_{i:k+1}$ *or* ($P_{i:k+1}$ *and* $G_{k:j}$ )[5]

3.Post processing

This is the last stage, and it is the same for all adders in this family (carry look ahead). It includes the addition of bits. The logic below is used to compute the sum bits:

$S_i = p_i$ xor $C_{i-1}$ [5].

## III. EXPERIMENTAL RESULT

The design is developed using VHDL. Simulation is done in Modelsim and the result obtained is again verified in the Xilinx. Fig 5 shows the simulation result of Ripple Carry Adder. Fig 6 shows the simulation result of the carry skip adder and Fig 7 shows the kogge stone adder.

Fig:5 32 bit Ripple Carry Adder

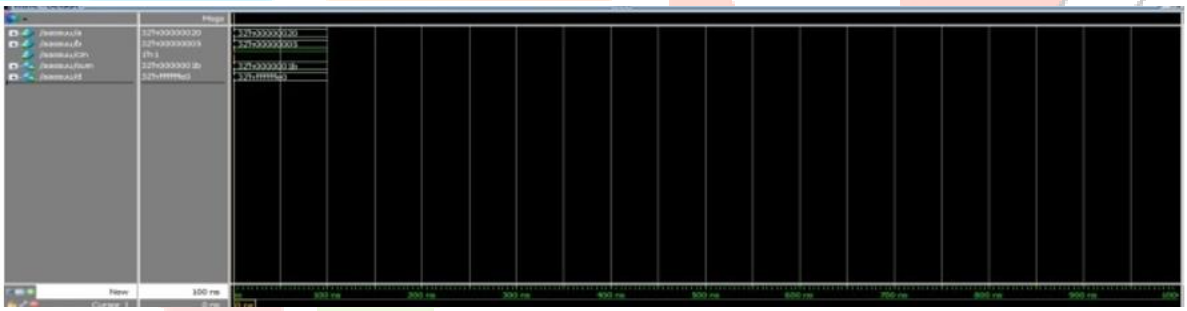


Fig:6 32 bit Caary Skip Adder

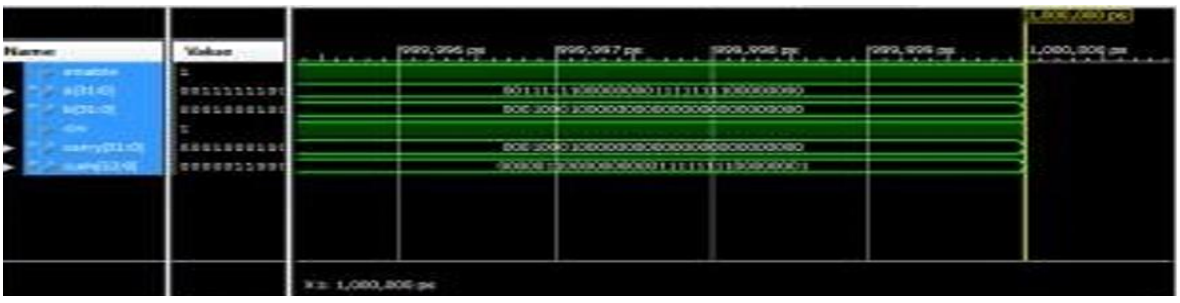


Fig:7 32 bit Kogge Stone Adder

## IV. COMPARISION

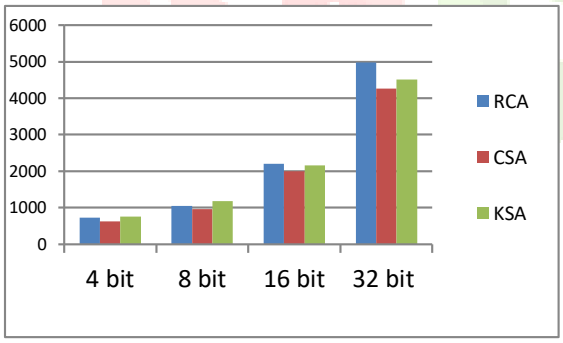
The comparison table shown below explains the comparison of all the three adders mentioned earlier. The comparison is done in two categories i.e., area or power of operation. the ripple carry adder operates with the lowest speed of power and also occupies less area, Kogge stone adder occupies less area and operates with high power. The comparisons have been made under the family of spartan 3E.

| No. | List of 32-bit adder | Area (Cell Area) | Power(mW) |
|---|---|---|---|
| 1. | Ripple Carry Adder | 4986 | 1.524 |
| 2. | Carry Skip Adder | 4258 | 1.497 |
| 3. | Kogge stone Adder | 4515 | 1.580 |

Fig: 8 comparison table of Adders

## V. COMPARISION GRAPH

In comparison graph area and the power can easily analyze. In this comparison graph the kogge stone adder is more efficient adder than the other two adders RCA & CSA. All the 4-bit, 8-bit, 16-bit and 32-bit are shown in comparison graph.



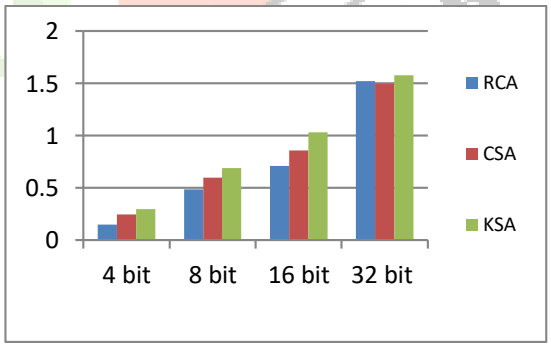




Fig: 9 comparison graph for Area

Fig:10 comparison graph for power

## VI. CONCLUSION

In this comparison study, the comparison is done among all the above mentioned 4-bit and 8 bit and 32-bit adders using VHDL codes. As we can observe the differences in the above-mentioned comparison table, Ripple Carry Adder consumes less area but takes more time for execution. Carry Skip adder consumes less area. The Kogge Stone Adder, is better as compared to RCA & CSA, as it requires less area and also requires lesser time. The efficient performance can be obtained using the Kogge Stone Adder.

## VII. REFERENCE

1)B. Koyada, N. Meghana, M. O. Jaleel and P. R. Jeripotula, "A comparative study on adders," *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, Chennai, 2017,pp.2226-2230,doi: 10.1109/WiSPNET.2017.8300155.

2) H. Naseri and S. Timarchi, "Low-Power and Fast Full Adder by Exploring New XOR and XNOR Gates," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 8, pp. 1481-1493, Aug. 2018, doi: 10.1109/TVLSI.2018.2820999.

3)T. Sultana, R. Bardhan, T. F. Bithee, Z. Tabassum and N. J. Lisa, "A compact design of n-bit ripple carry adder circuit using QCA architecture," *2015 IEEE/ACIS 14th International Conference on Computer and Information Science (ICIS)*, Las Vegas, NV, 2015, pp. 155-160, doi: 10.1109/ICIS.2015.7166586.

4)Manjunath K M, Abdul Lateef Haroon P S, A. Pagi and Ulaganathan J, "Analysis of various full-adder circuits in cadence," *2015 International Conference on Emerging Research in Electronics, Computer Science and Technology (ICERECT)*, Mandya, 2015, pp. 90-97, doi: 10.1109/ERECT.2015.7498994.

5) Daphni, S., & Grace, K. V. (2019). Design and Analysis of 32-bit Parallel Prefix Adders for Low Power VLSI Applications. *Advances in Science, Technology and Engineering Systems Journal*, *4*(2), 102-106.