IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

CARRY SKIP ADDER USING REVERSIBLE LOGIC GATES

[1]Shivunoori Gayathri, [2]Shuvanka Maji

[1]Student, [2]Assistant Professor

[1]Electronics and Communication Engineering,

[1]Vidya Jyothi Institute of Technology. Hyderabad, India

Abstract: The adder is a digital circuit that can multiply two numbers. The half adder, sometimes known as the full adder, is a device that adds single-bit integers together. To increase the bit size of a number, we utilize parallel adders, which are made up of a series of complete adders, such as the carry save adder (CSA), carry select adder (CSLA), carry skip adder (CSKA), and so on.

The speed of addition is limited by the time it takes for a carry to propagate through a digital adder. In a simple adder, the total for each bit position is calculated sequentially after the previous bit position has been summed and a carry has been transferred to the next place.

The binary adder is used in almost all digital circuit designs, including digital signal processors (DSP) and microprocessor data path units. Power consumption, area, and speed are all important considerations in VLSI design. There are various methods for reducing power use and saving space.

A carry skip adder (CSKA) is employed in this project, as well as a square root carry select adder (CSLA). The suggested design consumes less energy and occupies less space than the current one. Furthermore, as compared to earlier designs, the hardware complexity is minimized. This project uses the Xilinx-ISE 14.7 tool for simulation, logical verification, and further synthesizing, using VERILOG as the HDL language.

Index Terms - Carry skip adder (CSKA), square root carry select adder, OAI and AOI compound circuits, Verilog HDL.

I.INTRODUCTION

The adders are used in multipliers and DSP to run different algorithms including FFT, FIR, and IIR. Every time the idea of multiplication is mentioned, the adders are brought into the picture. As microprocessors handle millions of instructions each second, speed is the most crucial factor to take into account when constructing multipliers. Gadget miniaturization should be high and power consumption should be minimal due to the mobility of the device.

More battery backup is needed for devices like mobile phones and laptops. These three characteristics must thus be optimized by a VLSI designer. It is exceedingly challenging to adhere to these restrictions. Therefore, a compromise between the limits must be struck based on the need or application. Carry ripple adders have the smallest size but the slowest speed, while carry look ahead adders are faster but take up more space.

A middle ground between the two adders is provided by carry choose adders. Wang et al. introduced a novel hybrid adder idea in 2002 that uses hybrid carry look-ahead/carry select adders to speed up the adding process. Low power multipliers built on brand-new hybrid full adders were introduced in 2008. The goal of recent research in the field of digital electronics has mostly been to increase the speed of digital systems. Power dissipation is one of the most important design characteristics, as shown recently by the need for mobility and the modest increase in battery performance.

Area, latency, and power dissipation are the three most often used metrics to gauge a circuit's performance or evaluate different circuit designs. Power dissipation is strictly constrained by portability, yet fast computing speeds are still necessary. The power-delay product therefore becomes the most important performance parameter in modern VLSI systems. At various stages of the design process, optimizations are necessary to decrease power dissipation and increase speed.

We investigate the most efficient ways to construct adders in order to achieve low power consumption and fast speed since the majority of digital circuitry is built of simple and/or complicated gates. One of the most important areas of study in VLSI system design is the construction of high-speed data route logic systems that are both space and power efficient. The time it takes a carry to go through an adder limits the pace of addition in digital adders. In a basic adder, each bit location's sum is created sequentially only after the preceding bit position's sum and a carry have been transmitted to the subsequent position. In many computing systems, the CSLA is used to reduce the issue of carry propagation latency by independently producing a number of carries before choosing one of them to produce the total.

However, the CSLA is not space-efficient since it employs several pairs of ripple carry adders (RCA) to produce partial sum and carry by taking into account carry input Cin=0 and Cin=1, and then the multiplexers pick the final sum and carry (mux). The topic of Adder is a digital circuit. A digital circuit that adds numbers is known as an adder or summer in electronics. Adders are employed not just in the arithmetic logic units of many computers and other types of processors, but also in other areas of the processor where they are used to calculate addresses, table indexes, and other things comparable. Although adders may be built for a variety of numerical representations, such excess-3 or binary-coded decimal, the most often used adders work with binary integers. It is simple to convert an adder into an adder-subtractor when two's complement or ones' complement is being utilized to represent negative values. A more complicated adder is needed for other signed number formats.

II. METHOD

1. HALF ADDER:

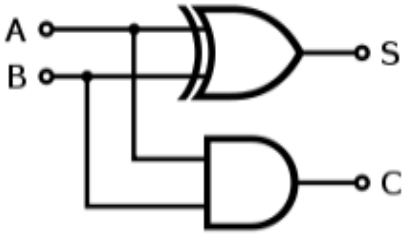

Figure 1 Half Adder

The half adder adds two one-bit binary numbers A and B. It has two outputs, Sum and Carry. The simplest half adder design, shown in figure 3.1, incorporates an XOR gate for Sum and an AND gate for Carry.

Inputs		Outputs	
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Table 1 Truth table of half adder

2. FULL ADDER:

With the addition of an OR gate to combine their carry outputs, two half adders can be combined to make a full adder.

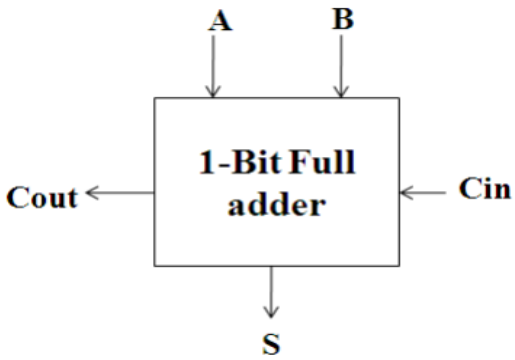

Figure 2 Full adder

The logic diagram of Full adder is shown in the figure 3.3. The full adder uses 2 XOR gates for the calculation of its sum and 1XOR, 1 OR and 2 AND gates for the calculation of carry out.

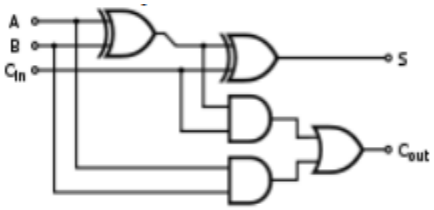

Figure 3 Logic diagram of Full adder

The full adder is usually a component in a cascade of adders, which add 8, 16, 32etc binary numbers. The circuit produces a two bit output sum represented by the signals C_{out} and S. The one-bit full adder's truth table is:

Inputs			Outputs	
A	B	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Table 2 Truth table of 1-bit full adder

A full adder can be implemented in many different ways such as with a custom transistor-level circuit or composed of other gates.

One example implementation is with

$S=ABC_{in}$

Cout= (A.B) + (Cin. (A B))

In this implementation, the final OR gate before the carry-out output may be replaced by an XOR gate without altering the resulting logic. Using only two types of gates is convenient if the circuit is being implemented using simple IC chips which contain only one gate type per chip.

In this, Cout can be implemented as Cout= (A.B) + (Cin. (A B)).

A full adder can be constructed from two half adders by connecting A and B to the input of one-half adder, connecting the sum from that to an input to the second adder, connecting C_i to the other input and OR the two carry outputs. Equivalently, S could be

Made the three-bit XOR of A, B and C_i and C_{out} could be made the three-bit majority function of A, B and C.

III. FAST ADDERS:

3. RIPPLE CARRY ADDER

Concatenating the N full adders forms N bit Ripple carry adder. In this carry out of previous full adder becomes the input carry for the next full adder. It calculates sum and carry according to the following equations. As carry ripples from one full adder to

The other, it traverses longest critical path and exhibits worst-case delay.

Si = Ai xor Bi xorCi

Ci+1 = Ai Bi + (Ai + Bi) Ci; where i = 0, 1, n-1

RCA is the slowest in all adders (O (n) time) but it is very compact in size (O (n) area). If the ripple carry adder is implemented by concatenating N full adders, the delay of such an adder is 2N gate delays from C_{in}

to C_{out}. The delay of adder increases linearly with increase in number of bits. The block diagram of RCA is shown in figure 3.4.

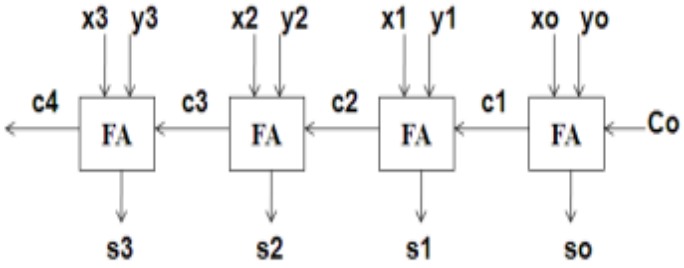

Figure 4 Block diagram of RCA

It is possible to create a logical circuit using multiple full adders to add N-bit numbers. Each full adder inputs a C_{in}, which is the C_{out} of the previous adder. This kind of adder is a ripple carry adder, since each carry bit "ripples" to the next full adder. It can be noted that the first (and only the first) full adder may be replaced by a half adder.

The layout of a ripple carry adder is simple, which allows for fast design time. However, the ripple carry adder is relatively slow, since each full adder must wait for the carry bit to be calculated from the previous full adder. The gate delay can easily be calculated by inspection of the full adder circuit. Each full adder requires three levels of logic. In a 32-bit ripple carry adder, there are 32 full adders, so the critical path (worst case) delay is 3 (from input to carry in first adder) + 31 * 2 (for carry propagation in later adders) = 65 gate delays. A design with alternating carry polarities and optimized AND-OR-Invert gates can be about twice as fast.

4. CARRY LOOK AHEAD ADDERS:

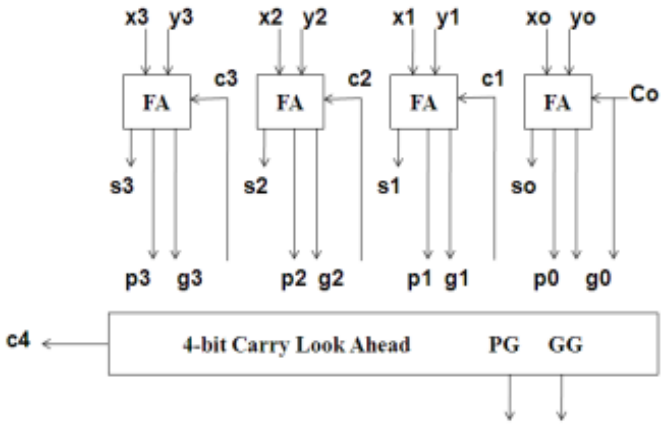

Figure 5 4-bit adder with carry look ahead

To reduce the computation time, engineers devised faster ways to add two binary numbers by using carry-look ahead adders.

They work by creating two signals (P and G) for each bit position, based on if a carry is propagated through from a less significant bit position (at least one input is a '1'), a carry is generated in that bit position (both inputs are '1'), or if a carry is killed

In that bit position (both inputs are '0').

In most cases, P is simply the sum output of a half-adder and G is the carry output of the same adder. After P and G are generated, the carries for every bit position are created. Some advanced carry-look ahead architectures are the Manchester carry chain, Brent– Kung adder and the Kogge–Stone adder.

Some other multi-bit adder architectures break the adder into blocks. It is possible to vary the length of these blocks based on the propagation delay of the circuits to optimize computation time. These block-based adders include the carry bypass adder which will determine P and G values for each block rather than each bit and the carry select adder which pre-generates sum and carry values for either possible carry input to the block.

Other adder designs include the carry save adder, carry-select adder, and conditional-sum adder, carry skip adder and carry-complete adder.

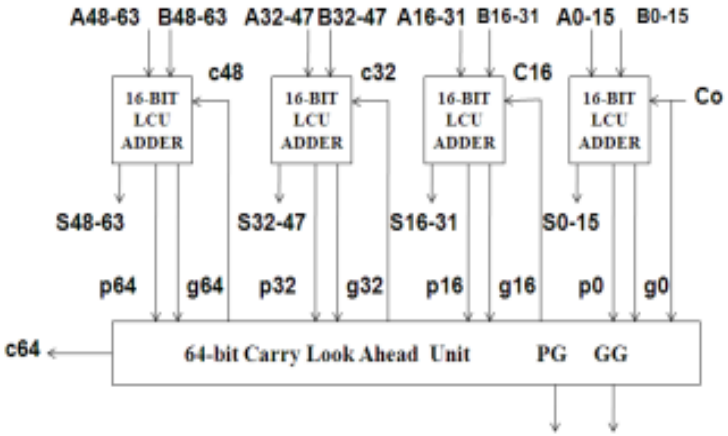

Figure 6 A 64-bit carry look ahead unit

By combining multiple carry look ahead adders even larger adders can be created. This can be used at multiple levels to make even larger adders. For example, the following adder is a 64-bit adder that uses four 16-bit CLAs with two levels of LCUs.

5. CARRY SKIP ADDER (CSKA):

The carry-skip adder reduces the time needed to propagate the carry by skipping over groups of consecutive adder stages, is

Known to be comparable in speed to the carry look-ahead technique while it uses less logic area and less power.

IV. UNIFORM SIZED ADDER:

A carry skip adder divides the words to be added into groups of equal size of k-bits. Carry Propagate pi signals may be used within a group of bits to accelerate the carry propagation. If all the pi signals within the group are pi=1, carry bypasses the entire

Group as shown in figure 3.7.

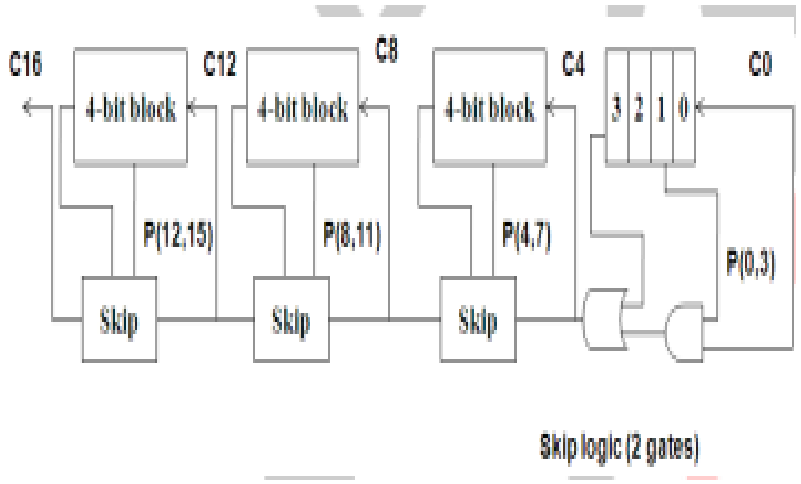

Figure 7 Carry skip adder

6. VARIABLE BLOCK ADDER:

The idea behind Variable Block Adder (VBA) is to minimize the critical path delay in the carry chain of a carry skip adder, while allowing the groups to take different sizes. In case of carry skip adder, such condition will result in a greater number of skips between stages.

Such an adder design is called variable block design, which is tremendously used to fasten the speed of adder. In the variable block carry skip adder design, we divided a 32-bit adder into 4 blocks or groups. The bit widths of groups are taken as: First block is of 4 bits, second is of 6 bits, third is 18 bits wide and the last group consist of most significant 4 bits.

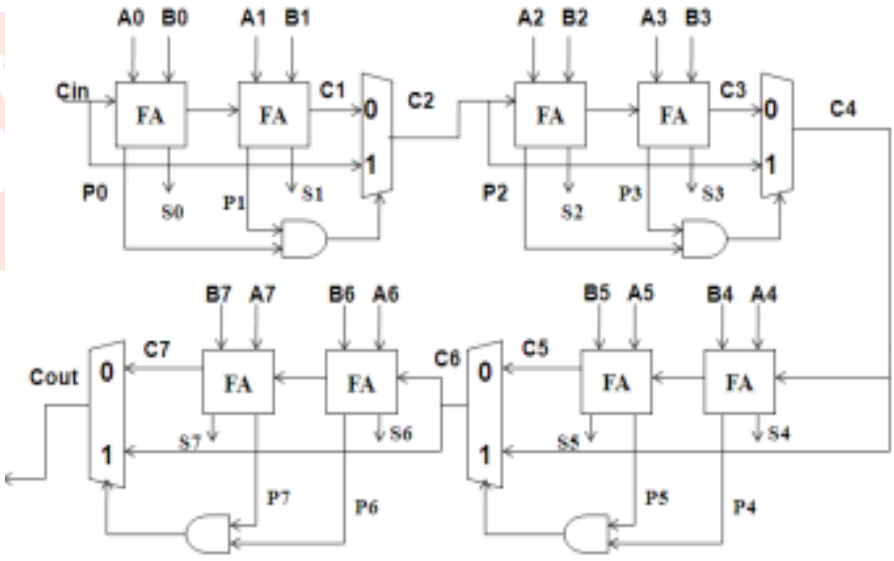

Figure 8 Architectural block of 8-bit Carry skip adder

V.RESULTS

- **RTL SCHEMATIC: -**

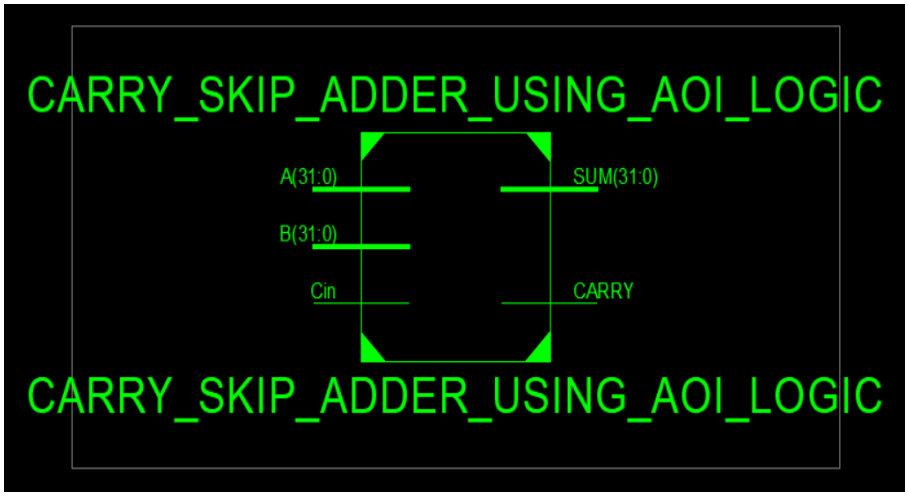

Figure 9 RTL Schematic of existed adder

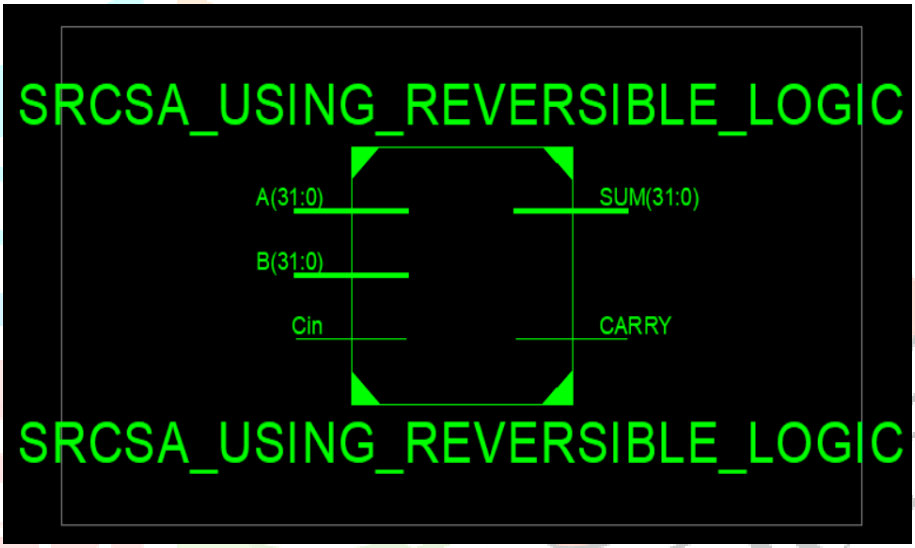

Figure 10 RTL Schematic of Proposed adder

- **TECHNOLOGY SCHEMATIC: -**

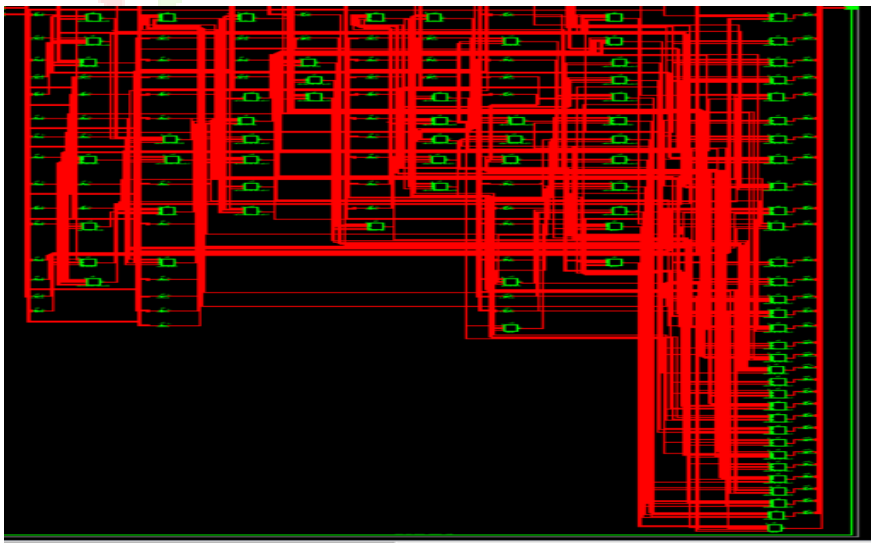

Figure 11 View Technology Schematic of existed adder

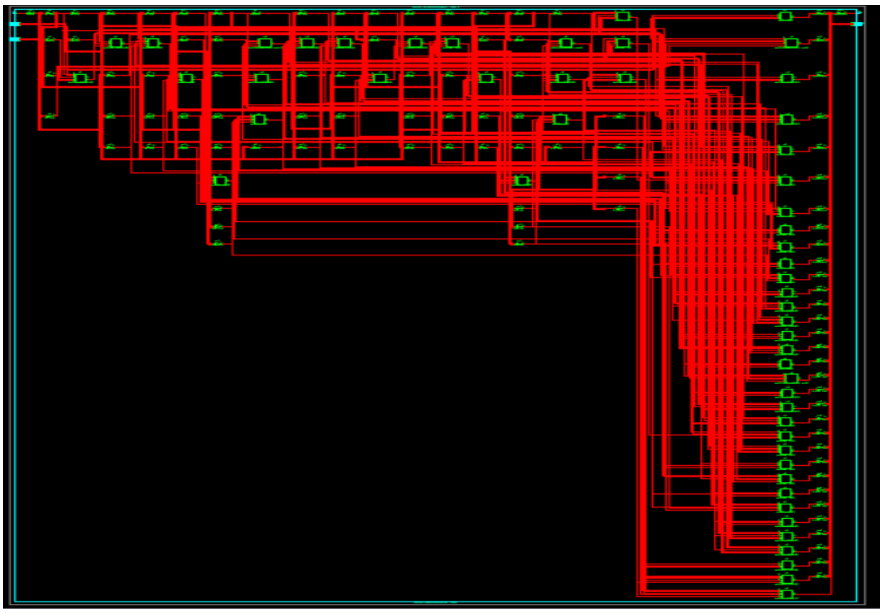

Figure 12 View Technology Schematic of proposed adder

- **SIMULATION: -**

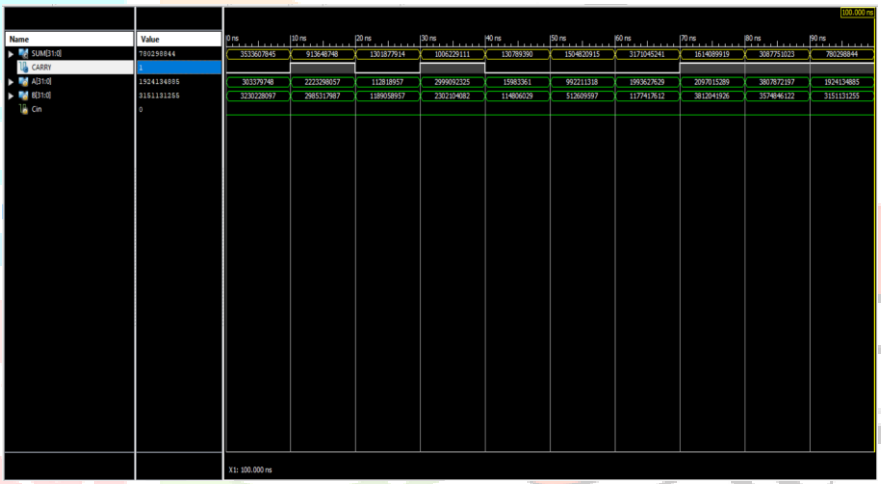

Figure 13 Simulated Waveforms of existed adder

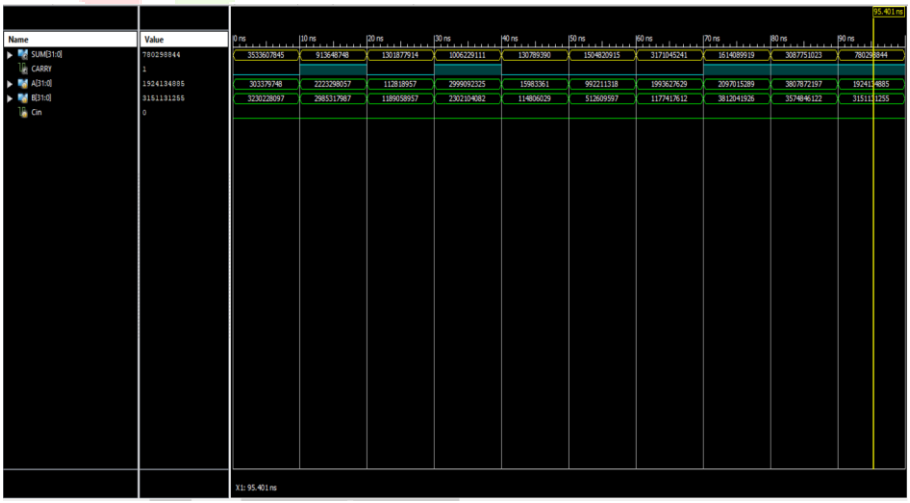

Figure 14 Simulated Waveforms of proposed adder

- **PARAMETERS: -**

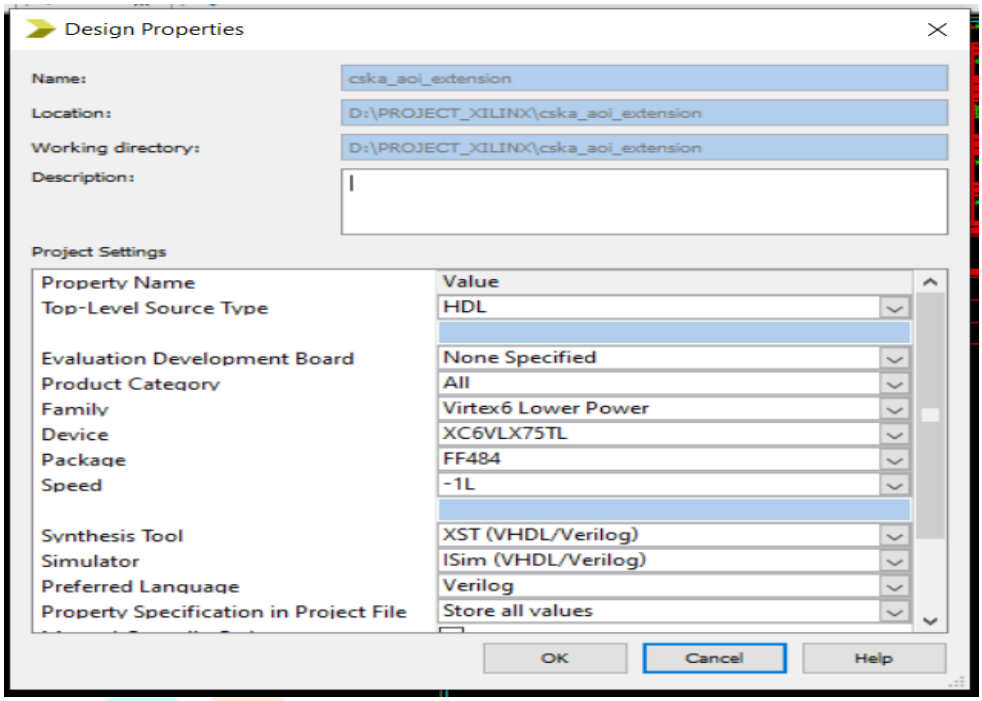

Figure 15 Device Used For Synthesis

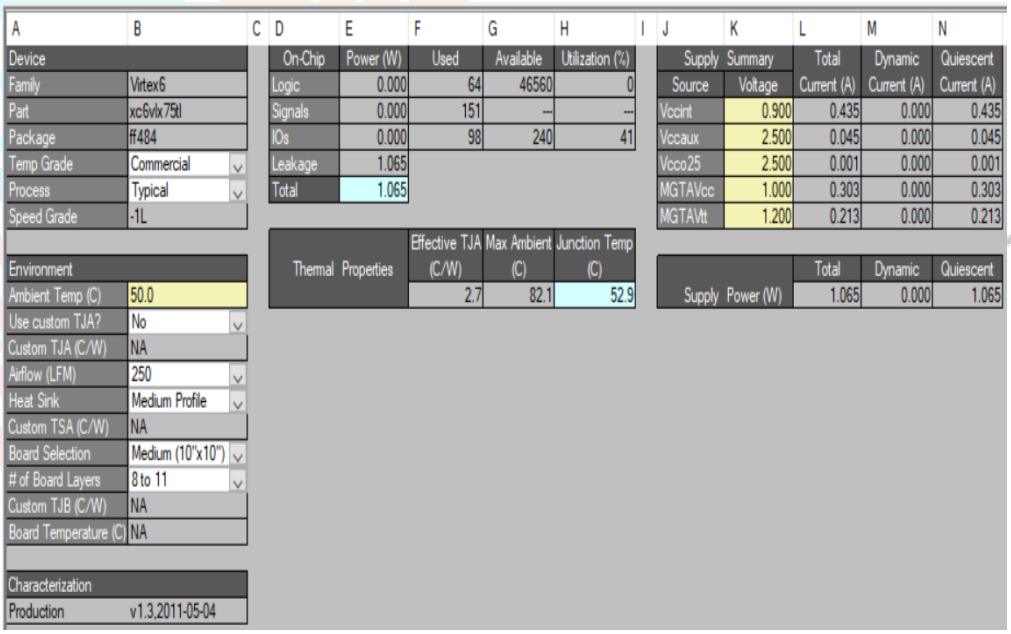

Figure 16 Power Analysis

Parameter	CI-CSKA	SRCSLA USING REVERSIBLE LOGIC
Power (m. Watt)	1.4639	0.7319
Number of LUT's	64	32

Table 3 Parameter Comparison

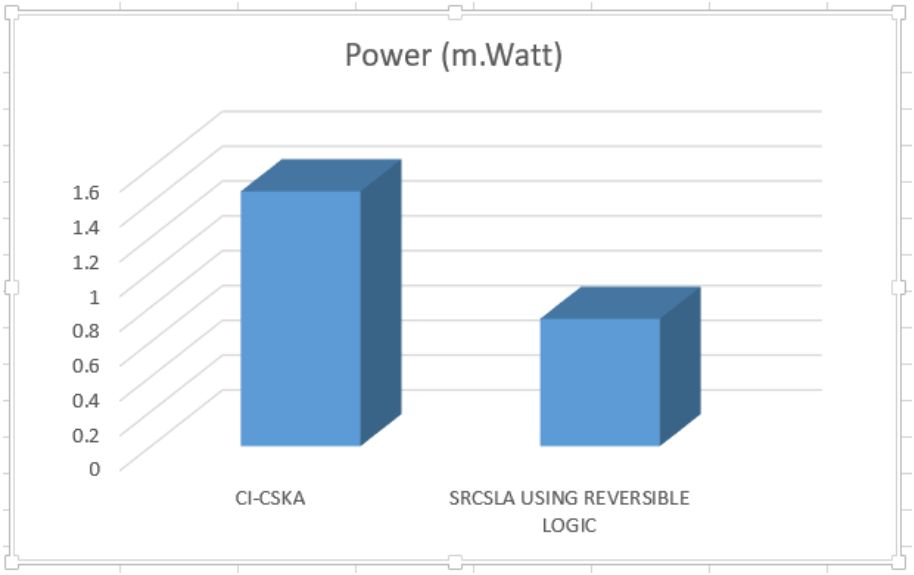

Figure 17 Power comparison bar graph

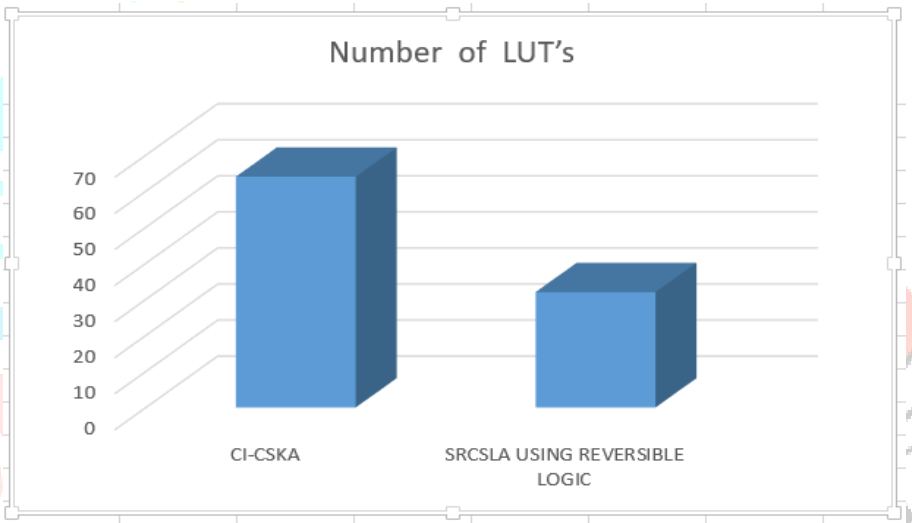

Figure 18 LUT comparison bar graph

VI.CONCLUSION

In this paper, we introduced square root CSLA, using reversible logic gates was proposed, which exhibits a less power consumption and less area of the conventional one. In addition, AOI and OAI compound gates were exploited for the carry skip logics and the suggested adder was designed using reversible concept. The proposed adder consumed 32 LUTs with 0.7319 m.

Watt power consumption and conventional design consumed 64 LUTs with 1.4639 m. Watt power consumption. The results suggested the proposed design is best (its area and power usage) compared to IC-CSKA using AOI. The results also suggested the proposed-adder structure as a very good adder for the applications where both the power and area.

The implementation, synthesis and simulation are performed in XILINX-ISE tool in Verilog HDL language. In future the implementation of this can enhance the performance in dsp applications, image processing, filters and cryptographic applications. Area and power-based applications.

VII.REFERENCES

[1] I. Koren, *Computer Arithmetic Algorithms*, 2nd ed. Natick, MA, USA:A K Peters, Ltd., 2002.

[2] R. Zlatanovici, S. Kao, and B. Nikolic, "Energy–delay optimization of 64-bit carry-look ahead adders with a 240 ps 90 nm CMOS design example," *IEEE J. Solid-State Circuits*, vol. 44, no. 2, pp. 569–583, Feb. 2009.

[3] S. K. Mathew, M. A. Anders, B. Bloechel, T. Nguyen, R. K. Krishnamurthy, and S. Borkar, "A 4-GHz 300-mW 64-bit

integer execution ALU with dual supply voltages in 90-nm CMOS," *IEEE J. Solid-State Circuits*, vol. 40, no. 1, pp. 44–51, Jan. 2005.

[4] V. G. Oklobdzija, B. R. Zeydel, H. Q. Dao, S. Mathew, and R. Krishnamurthy, "Comparison of high-performance VLSI adders in the energy-delay space," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 13, no. 6, pp. 754–758, Jun. 2005.

[5] B. Ramkumar and H. M. Kittur, "Low-power and area-efficient carry select adder," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 20, no. 2, pp. 371–375, Feb. 2012.

[6] M. Vratonjic, B. R. Zeydel, and V. G. Oklobdzija, "Low- and ultra-low-power arithmetic units: Design and comparison," in *Proc. IEEE Int. Conf. Comput. Design, VLSI Comput. Process. (ICCD)*, Oct. 2005, pp. 249–252.

[7] C. Nagendra, M. J. Irwin, and R. M. Owens, "Area-time-power tradeoffs in parallel adders," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 43, no. 10, pp. 689–702, Oct. 1996.

[8] Y. He and C.-H. Chang, "A power-delay efficient hybrid carrylookahead/ carry-select based redundant binary to two's complement converter," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 55, no. 1, pp. 336–346, Feb. 2008.