IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

HIGH PERFORMANCE RIPPLE CARRY ADDERS IN VLSI DESIGN

[1]Dr. S. Yuvaraj, Ph. D, [2]Dr. K.K. Senthilkumar, Ph. D,[3]Name Dr. K. Senthamil Selvan, Ph. D., [4] A.R. Aravind.,M.E.,

[1]Associate Professor, [2]Professor, [3]Professor, [4]Associate Professor

[1, 2, 3, 4] Department of Electronics and Communication Engineering

[1, 2, 3, 4] Prince Shri Venkateshwara Padmavathy Engineering College, Chennai, India

Abstract: In this Manuscript discuss the High performance Ripple Carry Adders (RCAs) in digital logic circuits design with uses of Addition forms the basis for many processing operations, from ALUs to address generation to multiplication to filtering. As a result, adder circuits that add two binary numbers are of great interest to digital system designers. An extensive, almost endless, assortment of adder architectures serves different speed/power/area requirements. This section begins with half adders and full adders for single-bit addition. It then considers a of ripple carry adders (RCAs) for the addition of multibit words in digital logic design and systems. In this paper to discuss the RCAs performance of Synthesis RTL architecture namely SLICES, LUT and IOBs using Xilinx ISE Design Suite 12.2. In addition with Testing of RCAs 32, 64 bit adder's waveform generation using Mentor Graphics Tool Model Sim. And compare different digital words bit like 16, 32, and 64 bit in the performance of device area and timing analysis using Xilinx FPGA.

Index Terms **- VLSI Design, low power VLSI Design, Digital logic System design, ALU, RCAs, FPGAs.**

I. INTRODUCTION

In this section the following two additions carry through the design of digital sub-system using top-down approaches. The complete system environment is that of a 16-bit microprocessor which is readily envisaged as an interconnection of four major architectural blocks-ALU, Control Unit, IOBs Unit and Memory in particularly discuss the 16ripple carry adders.

1.1 Adder element requirements

If then $A_k = B_k$ Then $S_k = C_{k-1}$ ---- (1.1)
else S_k= bar C_{k-1} ---- (1.2)

and for the carry c_k

If $A_k = B_k$ Then $A_k = C_k = B_k$* ---- (1.3)
else $C_k = C_{k-1}$ ---- (1.4)

*This relationship could also have been stated as:

Carry $C_k = 1$ when $A_k = B_k = 1$ ---- (1.5)
$C_k = 0$ when $A_k = B_k = 0$ ---- (1.6)

1.2 A standard adder element

A 1-bit adder element may now be represented as in Figure 1. Note that any number of such elements may be cascaded to form any size of adder and that the element is quite general.

1.2.1. One-bit full adder

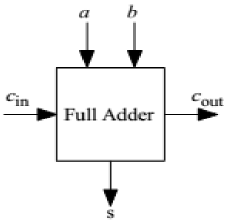


Fig. 1: One-bit full adder.

A one-bit full adder is a combinational circuit that forms the arithmetic sum of three bits. It consists of three inputs (a, b and c_{in}) and two outputs (s and c_{out}) as illustrated in Figure 1.

II. LITERATURE REVIEW

Yajuan He et al, presents an efficient reverse converter for transforming the redundant binary (RB) representation into two's complement form. The hierarchical expansion of the carry equation for the reverse conversion algorithm creates a regular multilevel structure, from which a high-speed hybrid carry-Look ahead/carry-select (CLA/CSL) architecture is proposed to fully exploit the redundancy of RB encoding for VLSI efficient implementation [1]. Vikramkumar Pudi et al explains the design of adders on quantum dot cellular automata (QCA) has been of recent interest. While few designs exist, investigations on reduction of QCA primitives (majority gates and inverters) for various adders are limited. In this paper, we present a number of new results on majority logic. We use these results to present efficient QCA designs for the ripple carry adder (RCA) and various prefix adders [2]. Three-operand binary adder is the basic functional unit to perform the modular arithmetic in various cryptography and pseudorandom bit generator (PRBG) algorithms. Carry-save adder (CS3A) is the widely used technique to perform the three-operand addition. However, the ripple-carry stage in the CS3A leads to a high propagation delay of O (n) to converse Amit Kumar Panda et al [3]. The efficient implementation of adders in differential logic can be carried out using a new generate signal (N) presented in this manuscript. This signal enables iterative shared transistor structures to be built with a better speed/area performance than a conventional implementation. It also allows an adder developed in domino logic to be easily adapted to differential logic is presents G.A. Ruiz et al [4].

Integer addition is one of the most important operations in digital computer systems because the performance of processors is significantly influenced by the speed of their adders. This paper proposes a self-timed carry-Lookahead adder in which the logic complexity is a linear function of n, the number of inputs, and the average computation time is proportional to the logarithm of the logarithm of n is discuss Fu-Chiung Cheng et al [5]. Heumpil Cho et al [6] knowingly Quantum-dot cellular automata (QCA) is an emerging nanotechnology for electronic circuits. Its advantages such as faster speed, smaller size, and lower power consumption are very attractive. The fundamental device, a quantum-dot cell, can be used to make gates, wires, and memories. As such it is the basic building block of nanotechnology circuits. While the physical nature of the nanoscale materials is complicated, the circuit designer can concentrate on the logical and structural design, so the design effort is reduced. Because of its novelty, the current literature shows only simple circuit structures. C. Nagendra et al [7] several classes of parallel, synchronous adders are surveyed based on their power, delay and area characteristics. The adders studied include the linear time ripple carry and Manchester carry chain adders, the square-root time carry skip and carry select adders, the logarithmic time carry Look ahead adder and its variations, and the constant time signed-digit and carry-save adders. Most of the research in the last few decades has concentrated on reducing the delay of addition. With the rising popularity of portable computers, however, the emphasis is on both high speed and low power operation. The ripple-carry adder (RCA) has the simplest circuit structure but the longest delay among all adders. Thus, it is often realized with the dynamic circuits when speed is the major concern. In this paper, we propose circuit-level and architecture-level innovations over the dynamic RCA (DRCA) that lead to high operation speed and low hardware overhead. Circuit-wise, we propose a cost-effective way to eliminate the race problem of DRCA. Architecture-wise, we propose a new carry-forwarding scheme that combines a diagonal forwarding with the multilevel folding for dramatic speed improvement of the DRCA. Finally, a new multilevel carry-forwarding scheme is proposed to reduce the circuit complexity while keeping the speed is presents Chung-Hsun Huang et al [8]. K.K. Parhi et al [9] presents novel hybrid carry-select modified-tree (CSMT) adder architectures for binary carry generators and adders using multiplexers only. These architectures not only require the fewest number of multiplexers, but also consume the least energy for a specified latency. These architectures are based on a carry-select configuration where each block can be a carry-select or tree or modified-tree block. The modified-tree blocks permit ripple in the carry-generation process; which leads to dramatic reduction in the number of multiplexers as well as power consumption.

To design a power-efficient digital signal processor, this study develops a fundamental arithmetic unit of a low-power adder that operates on effective dynamic data ranges. Before performing an addition operation, the effective dynamic ranges of two input data are determined. Based on a larger effective dynamic range, only selected functional blocks of the adder are activated to generate the desired result while the input bits of the unused functional blocks remain in their previous states. The added result is then recovered to match the required word length. Using this approach to reduce switching operations of non-effective bits allows input data in 2's complement and sign magnitude representations to have similar switching activities is explain O.T.-C. Chen et al [11]. A. Tyagiet el [10] presents the carry-select or conditional-sum adders require carry-chain evaluations for each block for both the values of block-carry-in, 0 and 1. The author introduces a scheme to generate carry bits with block-carry-in 1 from the carries of a block with block-carry-in 0. This scheme is then applied to carry-select and parallel-prefix adders to derive a more area-efficient implementation for both the cases. The proposed carry-select scheme is assessed relative to carry-ripple, classical carry-select, and carry-skip adders. The analytic evaluation is done with respect to the gate-count model for area and gate-delay units for time. Hung Chi Lai et el explain the logic networks of carry–save adders such as high-speed multipliers, multioperand adders, and double-rail input parallel adders are designed based on the parallel adders with a minimum number of NOR gates discussed in [12]. W. F. Wallace et al to explain a new type of simple adder, suitable for asynchronous digital circuits and implementation in VLSI technology, which has either speed and/or area advantages over existing designs. It is based on the concept of predicting the carry from least to most significant halves of a 16 bit adder in such a way that it has a high probability of being correct, while introducing only a low area overhead from the required early completion control circuitry [13]. Ahmed H.Fouadm et al presents the possibilities of generalizing a memristor based ternary adder circuit, to a memristor based multi-valued logic adder. The proposition tries to achieve the theoretical advantages of processing different numbering systems, increasing the density, and decreasing the processing time, by utilizing the memristor properties and dynamics [14] the ripple carry adders as shown in figure 5 is given below.

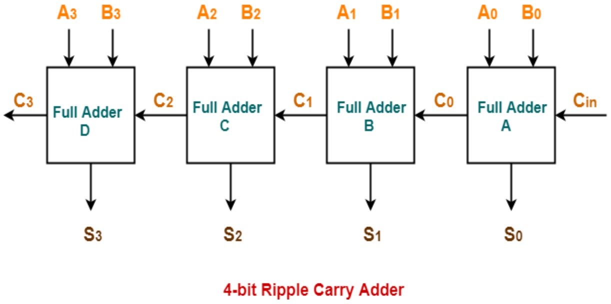


Fig.2: RCAs

III. RIPPLE CARRY ADDERS (RCAs)

A ripple carry adder is a digital circuit that produces the arithmetic sum of two binary numbers. It can be constructed with full adders connected in cascaded, with the carry output from each full adder connected to the carry input of the next full adder in the chain. Figure 2 shows the interconnection of four full adder (FA) circuits to provide a 4-bit ripple carry adder. Notice from Figure 3 that the input is from the right side because the first cell traditionally represents the least significant bit (LSB). Bits and   in the figure represent the least significant bits of the numbers to be added. The sum output is represented by the bits $S_{3,}$ S_2, S_1 and S_0.

Using your full-adder create a 4-bit ripple-carry adder (schematic only) with pipelined inputs and outputs. The ripple-carry architecture is shown in figure 3. In the ripple carry adder, the output is known after the carry generated by the previous stage is produced. Thus, the sum of the most significant bit is only available after the carry signal has rippled through the adder from the least significant stage to the most significant stage. As a result, the final sum and carry bits will be valid after a considerable delay.

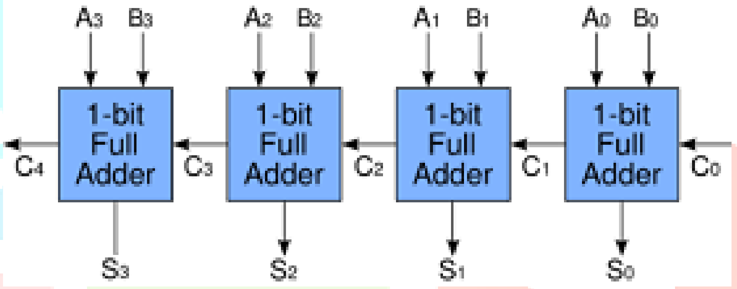


Fig. 3: 4-bit ripple carry adder.

IV. 16, 32, 64-bit RIPPLE CARRY ADDERS

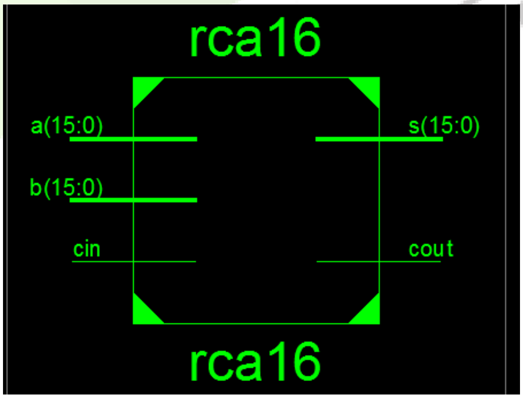


Fig. 4 (a): 16-bit Ripple Carry Adder (RCA) RTL Diagram

As shown In Figure 4(a). 16-bit ripple carry adder diagram using VHDL/Verilog Coding is implemented in Xilinx Software ISE 12.2i. Full adder vhdl/Verilog coding synthesized with Register Transfer Level (RTL) diagram and Technology RTL is synthesized, post and route generated as shown in Figure 4(a), 4(b) and simulated full adder output waveform a[15:0]=16'hABCD, b[15:0]=16'hABCD and cin=1'b0 then sum[15:0]=16'h579a, cout=1'b1 else a[15:0]=16'h1A2D, b[15:0]=16'hBC66 and cin = 1'b1 then sum[15:0]=16'hD694, cout=1'b0 as shown in Figure 4(c) and generate bit file to fused in Xilinx PROM XCF32p and implemented in FPGA Family as show in figure 4(d).

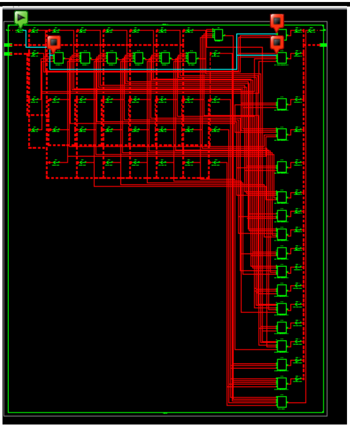
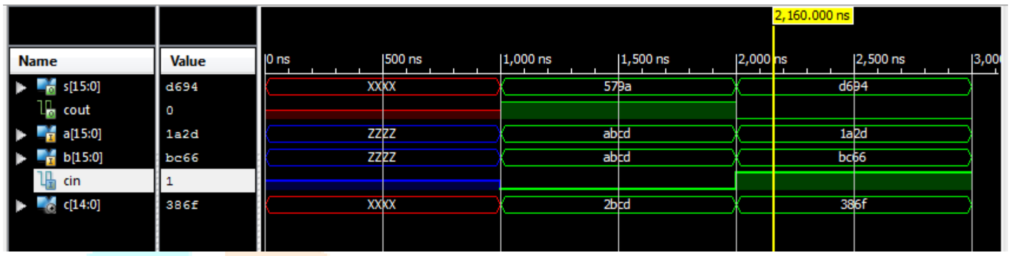


Fig. 4(b): 16-bit Ripple Carry Adder (RCA) Technology RTL Diagram



Name	Value
s[15:0]	d694
cout	0
a[15:0]	1a2d
b[15:0]	bc66
cin	1
c[14:0]	386f

Fig. 4(c). 16-bit Ripple Carry Adder (RCA) Timing Diagram

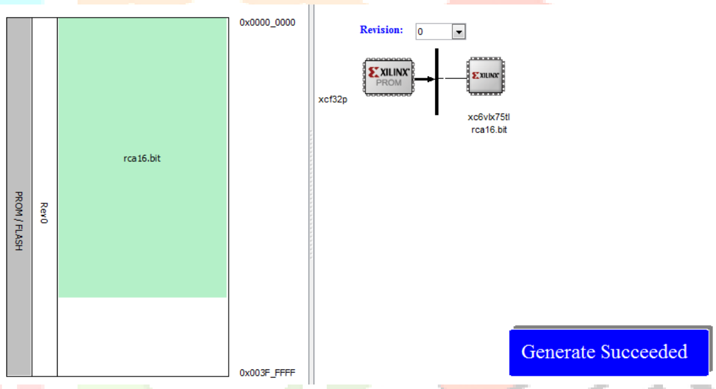


Fig. 4(d): 16-bit Ripple Carry Adder (RCA) implement in FPGA Diagram

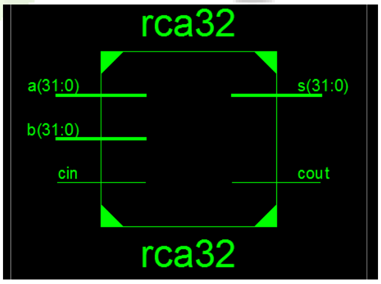


Figure 5(a). 32-bit Ripple Carry Adder (RCA) RTL

As shown In Figure 5(a). 32-bit ripple carry adder diagram using VHDL/Verilog Coding is implemented in Xilinx Software ISE 12.2i. Full adder vhdl/Verilog coding synthesized with Register Transfer Level (RTL) diagram and Technology RTL is synthesized, post and route generated as shown in Figure 5(a), 5(b) and simulated full adder output waveform a[31:0]=16’h12ad34eb, b[31:0]=16’hbca2489c and cin=1’b0 then sum[31:0]=16’hcf4f7d82, cout=1’b0 else a[31:0]=16’h12ad34eb, b[31:0]=16’h1234abcc and cin = 1’b0 then sum[31:0]=16’h45ab5dcb, cout=1’b0 as shown in Figure 5(c) and generate bit file to fused in Xilinx PROM XCF32p and implemented in FPGA as show in fig. 5(d).

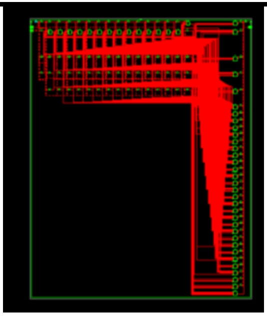
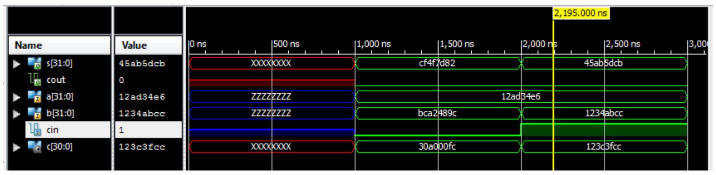
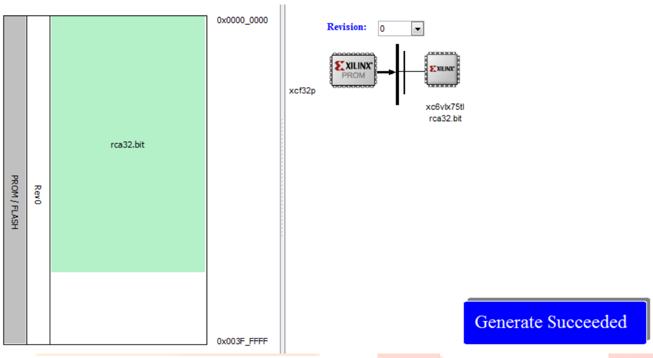


Figure 5(b). 32-bit Ripple Carry Adder (RCA) RTL Technology



```
Name        Value
s[31:0]     45ab5dcb
cout        0
a[31:0]     12ad34e6
b[31:0]     1234abcc
cin         1
c[30:0]     123c3fcc
```

Figure 5(c). 32-bit Ripple Carry Adder (RCA) Timing



Revision: 0

Generate Succeeded

Figure 5(d). 32-bit Ripple Carry Adder (RCA) implement in FPGA

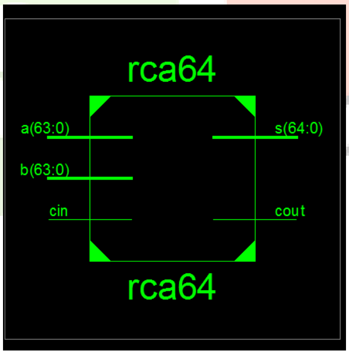


Figure 6(a). 64-bit Ripple Carry Adder (RCA) RTL Diagram

As shown In Figure 6(a). 64-bit ripple carry adder diagram using VHDL/Verilog Coding is implemented in Xilinx Software ISE 12.2i. Full adder vhdl/Verilog coding synthesized with Register Transfer Level (RTL) diagram and Technology RTL is synthesized, post and route generated as shown in Figure 6(a), 6(b) and simulated full adder output waveform a[63:0]=16’h12ad435e677ba877, b[63:0]=16’h34cda67be8eacccc and cin=1’b0 then sum[63:0]=16’h,477A E9DA 5066 7543 cout=1’b0 else a[63:0]=16’h12ad435e677ba877, b[63:0]=16’h34cda67be8eacccc and cin = 1’b1 then sum[63:0]=16’h477A E9DA 5066 7544, cout=1’b1 as shown in Figure 6(c) and generate bit file to fused in Xilinx PROM XCF32p and implemented in FPGA Family as show in figure 6(d).

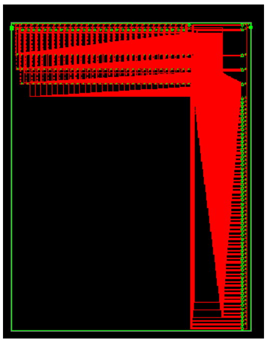
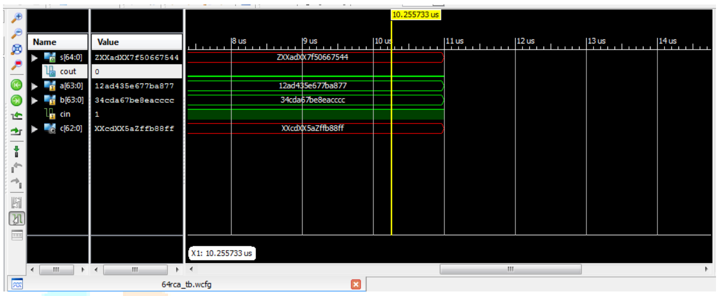


Figure 6(b). 64-bit Ripple Carry Adder (RCA) Technology RTL Diagram



Name	Value
s[64:0]	ZXXadXX7f50667544
cout	0
a[63:0]	12ad435e677ba877
b[63:0]	34cda67be8eacccc
cin	1
c[62:0]	XXcdXX5aZffb88ff

Figure 6(c). 64-bit Ripple Carry Adder (RCA) Timing Diagram

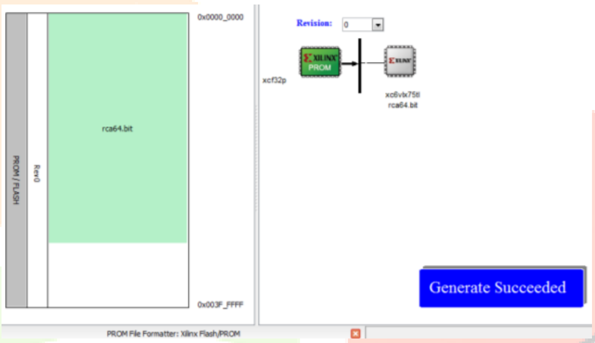


Figure 6(d). 64-bit Ripple Carry Adder (RCA) implement in FPGA Diagram

V. RESULTS AND DISCUSSIONS

We use the Xilinx ISE 12.2i FPGA design and a review of Ripple carry adders shows that the RCA Architecture 64, 32, 16 bit is still a popular choice for its stable performance and high speed capability. The other advantage of the RCA over other exiting algorithm is its high convergence rate. The high- speed capability and register rich architecture of the FPGA is ideal for implementing RCA. The proposed model is programmed downloaded on XILINX FPGA board. It is implemented on FPGA Device Vertex Low power. After synthesizing in Xilinx project navigator we got RTL schematic diagram of our proposed design which is shown in Figure 6(a), 5(a) and 4(a). Total estimated power consumption is 7mW, Quiescent Voltage is Vcco33 3.30V, Quiescent Current is 2mA.

TABLE 1. RCAs performance Parameters utilization summary:

SL. No	1-bit xor2	BELs	3 in LUT	5 in LUT	IBs	OBs	IOBs
4-bit	4	5	2	3	9	4	13
8-bit	16	12	4	8	17	9	26
16-bit	32	24	8	16	33	17	50
32-bit	64	48	16	32	65	33	98
64-bit	128	96	32	64	129	65	194

The 4-bit Ripple carry adder wrote in VHDL/Verilog code are implemented in Xilinx ISE software to produce the results in Table 1 have been discussed here single bit 2 input xor2 gate are 4, Basic Element of Logic (BELs) are 5, 3 input Look up table (LUTs) is 2, 5 input LUT is 3 and the input blocks (IBs) are 9, the output blocks (OBs) are 4 and the input-output blocks (IOBs) are 13. Similarly 8-bit ripple carry adder have been discussed single bit 2 input xor2 gate are 16, Basic Element of Logic (BELs) are 12, 3 input Look up table (LUTs) is 4, 5 input LUT is 8 and the input blocks (IBs) are 17, the output blocks (OBs) are 9 and the input-output blocks (IOBs) are 26. Similarly 16-bit ripple carry adder have been discussed single bit 2 input xor2 gate are 32, Basic Element of Logic (BELs) are 24, 3 input Look up table (LUTs) is 8, 5 input LUT is 16 and the input blocks (IBs) are 33, the output blocks (OBs) are 17 and the input-output blocks (IOBs) are 50. Similarly 32-bit ripple carry adder have been discussed single bit 2 input xor2 gate are 64, Basic Element of Logic (BELs) are 48, 3 input Look up table (LUTs) is 16, 5 input LUT is 32 and the input blocks (IBs) are 65, the output blocks (OBs) are 33 and the input-output blocks (IOBs) are 98. Similarly 64-bit ripple carry adder have been discussed single bit 2 input xor2 gate are 128, Basic Element of Logic (BELs) are 96, 3 input Look up table (LUTs) is 32, 5 input LUT is 32 and the input blocks (IBs) are 129, the output blocks (OBs) are 65 and the input-output blocks (IOBs) are 194 and also power consumption 487 mWatts with Ambient Temperature 50ºC are discussed as on Table 1 and as shown in Figure 7.

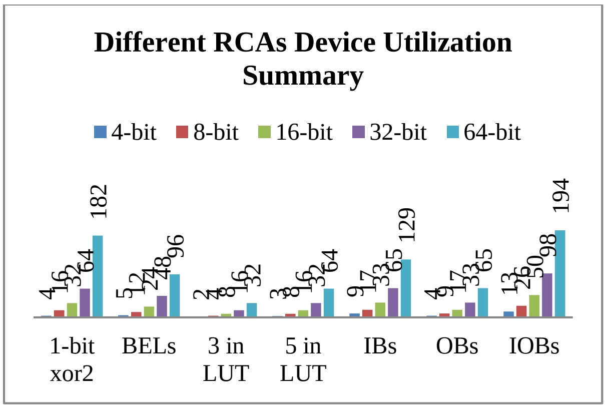


Figure 7. RCAs performance Parameters utilization summary

TABLE 2. RCAs performance Parameters Timing summary:

SL. No	Max. Comb. Path Delay (ns)	Logic (ns)	Route (ns)	Logic (%)	Route (%)
4-bit	1.71	0.11	1.60	6.5	93.5
8-bit	2.80	0.21	2.59	7.8	92.2
16-bit	4.98	0.43	4.55	8.6	91.4
32-bit	9.33	0.85	8.48	9.1	90.9
64-bit	18.04	1.70	16.33	9.4	90.6

The 4-bit Ripple carry adder (RCA) maximum combinational path delay 1.71ns in logically 0.11ns within percentile is 6.5% and routing 1.60 ns within percentile are 93.5%. Similarly The 8-bit Ripple carry adder (RCA) maximum combinational path delay 2.80ns in logically 0.21ns within percentile is 7.8% and routing 2.59 ns within percentile are 92.2%. Similarly The 16-bit Ripple carry adder (RCA) maximum combinational path delay 4.98ns in logically 0.43ns within percentile is 8.6% and routing 4.55ns within percentile are 91.4%. Similarly The 32-bit Ripple carry adder (RCA) maximum combinational path delay 9.33ns in logically 0.85ns within percentile is 9.1% and routing 8.48ns within percentile are 90.9%. Similarly The 64-bit Ripple carry adder (RCA) maximum combinational path delay 18.04ns in logically 1.7ns within percentile is 16.33% and routing 9.4ns within percentile are 90.6% are discussed as on Table 2 and as shown in Figure 8.

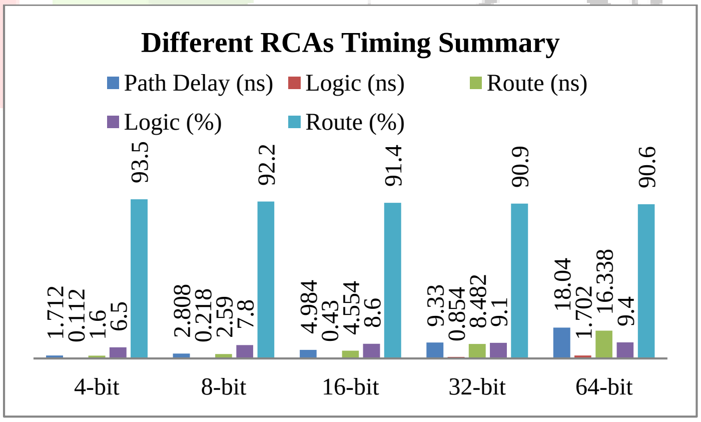


Figure 8. RCAs performance Parameters Timing summary

VI. CONCULUSIONS

In this Manuscript discuss the High performance Ripple Carry Adders (RCAs) in digital logic circuits design with uses of Addition forms the basis for many processing operations, from ALUs to address generation to multiplication to filtering. As a result, adder circuits that add two binary numbers like 32-bit and 64-bit are of great interest to digital system designers.It then considers a plethora of ripple carry adders (RCAs) for the addition of multibit words in digital logic design and systems. In this paper to discuss the RCAs performance of Synthesis RTL architecture namely SLICE for 32, 64-bit are 48 SLICE, 96 SLICE is respectively, LUT for 32, 64-bit are 32 LUT, 64 LUT is respectively and Maximum Combinational logic path delay for 32, 64-bit are 90.90nsec, 90.60nsec is respectively using Xilinx ISE Design Suite 12.2. Here we discuss max. Combinational path delays for 32- bit are 90.90nsec for whereas 64-bit are 90.60nsec in difference 0.30nsec with add another 32-bit binary numbers. In this results reduced another 0.30nsec In addition with Testing of RCAs 32, 64 bit adder's waveform generation using Mentor Graphics Tool Model Sim. And compare different digital words bit like 16, 32, and 64-bit in the performance of device area and timing analysis for 32, 64-bit ripple carry adder logically 0.85nsec , 1.70nsec and place and routing (PAR) time are 8.49 nsec , 16.33nsec is respectively using Xilinx FPGA.

VII. Acknowledgment

The authors would like to thank to Chairman and Vice Chairman, Principal of Prince Shri Venkateshwara Padmavathy Engineering College, Chennai, India, Faculty of Electrical Engineering, Department of Electronics and Communication Engineering, the anonymous reviewers and associate editor for their comments that greatly improved the manuscript.

References

[1] Zhoufeng Ying, ShounakDhar, Zheng Zhao, ChenghaoFeng, RohanMital, Chi-Jui Chung. 2018. Electro-Optic Ripple-Carry Adder in Integrated Silicon Photonics for Optical Computing. IEEE Journal of Selected Topics in Quantum Electronics, 24(6).

[2]. Amit Kumar Panda, Rakesh Palisetty, Kailash Chandra Ray. 2020. High-Speed Area-Efficient VLSI Architecture of Three-Operand Binary Adder. IEEE Transactions on Circuits and Systems I: Regular Papers, 67(11) : 3944 – 3953.

[3] Vikramkumar Pudi, K. Sridharan. 2012. Low Complexity Design of Ripple Carry and Brent–Kung Adders in QCA. IEEE Transactions on Nanotechnology, 11(1):105 – 119.

[4] Yajuan He, Chip-Hong Chang. 2008. A Power-Delay Efficient Hybrid Carry-Look ahead/Carry-Select Based Redundant Binary to Two's Complement Converter. IEEE Transactions on Circuits and Systems I: Regular Papers, 55(1) : 336 – 346.

[5] G.A. Ruiz. 1998. Evaluation of three 32-bit CMOS adders in DCVS logic for self-timed circuits. Published in: IEEE Journal of Solid-State Circuits, 33(4): 604 – 613.

[6] Fu-Chiung Cheng; S.H. Unger; M. Theobald, "Self-timed carry-Lookahead adders", Published in: IEEE Transactions on Computers (Volume: 49, Issue: 7, Jul 2000), Page(s): 659 – 672, DOI: 10.1109/12.863035.

[7] Heumpil Cho; Earl E. Swartzlander, "Adder Designs and Analyses for Quantum-Dot Cellular Automata", IEEE Transactions on Nanotechnology (Volume: 6, Issue: 3, May 2007), Page(s): 374 – 383, DOI: 10.1109/TNANO.2007.894839.

[8] Chung-Hsun Huang, Jinn-Shyan Wang, ChingweiYeh, Chih-Jen Fang. 2004. The CMOS carry-forward adders. IEEE Journal of Solid-State Circuits, 39(2): 327 – 336.

[9] K.K. Parhi. 1999. Low-energy CSMT carry generators and binary adders. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 7(4): 450 – 462.

[10] O.T.-C. Chen, R.R.-B. Sheen, S. Wang. 2002. A low-power adder operating on effective dynamic data ranges. Published in: IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 10(4): 435 – 453.

[11] A. Tyagi. 1993. A reduced-area scheme for carry-select adder. Published in: IEEE Transactions on Computers, 42(10): 1163 – 1170.

[12] H. C. Lai and S. Muroga. 1979. Minimum parallel binary adders with NOR (NAND) gates. IEEE Trans. Comput., C-28: 648-659.

[13] W.F. Wallace, S.S. Dlay, O.R. Hinton. 2001. Probabilistic carry state estimate for improved asynchronous adder performance. 148(6): 221 – 226.

[14] C. Nagendra, M.J. Irwin, R.M. Owens. 1996. Area-time-power tradeoffs in parallel adders. Published in: IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing. 43(10) : 689 – 702.

[15]. Ahmed H.Fouadm, Ahmed G.Radwan. 2019. Memristor-based quinary half adder. AEU - International Journal of Electronics and Communications, 98: 123-130.

[16] L.A. Montalvo, K.K. Parhi, J.H. Satyanarayana. 2002. Estimation of average energy consumption of ripple-carry adder based on average length carry chains. VLSI Signal Processing, IX.

[17] Wu-Tung Cheng. 1987. A Minimum Test Set for Multiple Fault Detection in Ripple Carry Adders. IEEE Transactions on Computers, C-36(7): 891 – 895.

[18] Kleanthis Papachatzopoulos, Vassilis Paliouras. 2019. Static Delay Variation Models for Ripple-Carry and Borrow-Save Adders. IEEE Transactions on Circuits and Systems I: Regular Papers, 66(7) : 2546 –2559.