IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

High Performance 128 Bit Ripple Carry Adder To Vlsi Design

1Dr.S.Yuvaraj, Ph. D.,(Corresponding Authors),2Dr.K. K. Senthilkumar.,Ph. D, 3Dr.K.Senthamilselvan., Ph.D.,4Dr. V. Ashok, Ph. D., 5Dr.A.R.Aravind, Ph.D.,

1, Associate Professor, Department of Electronics and Communication Engineering, Prince Shri Venkateshwara Padmavathy Engineering College, Chennai, Tamil Nadu, India,

2, Professor, Department of Electronics and Communication Engineering, Prince Shri Venkateshwara Padmavathy Engineering College, Chennai, Tamil Nadu, India,

3,Professor, Department of Electronics and Commmunication Engineering, Prince Shri Venkateswara Padmavathy Engineering College, Chennai, Tamil Nadu, India,

4, Professor, Department of Biomedical Engineering, Prince Dr. K. Vasudevan College of Engineering and Technology, Chennai,600 127, Tamil Nadu, India,

5,Associate Professor,Department of Electronics and Communication Engineering, Prince Shri Venkateshwara Padmavathy Engineering College,Chennai,Tamil Nadu, India,

Abstract: This manuscript discussed to efficient 128- bits binary adder the basic components widely used in VLSI design and it is required to be fast and support a wide bit-width in many applications. For example 128-bits are used to a common key size for symmetric ciphers and a common block size for block ciphers in cryptography. This ripple carry adder design by using HDL program is implemented in Xilinx ISE 12.2. also the result are single bit 2 input xor2 gate are 256, Basic Element of Logic (BELs) are 192, 3 input Look up table (LUTs) is 64, 5 input LUT is 128 and the input blocks (IBs) are 257, the output blocks (OBs) are 129 and the input-output blocks (IOBs) are 386. And also power consumption 487 mWatts with Ambient Temperature 50ºC. also discussed The 128-bit Ripple carry adder (RCA) maximum combinational path delay 35.448ns in logically 3.39ns with in percentage is 9.6% and routing 32.5ns with in percentage are 90.4%.

***Index Terms* -** Adder, RCA, VLSI, FPGA, Xilinx, SLICE, IOBs.

I. INTRODUCTION

Increasing the word size can speed up multiple precision mathematical libraries, with applications to cryptography, and potentially speed up algorithms used in complex mathematical processing (numerical analysis, signal processing, complex photo editing and audio and video processing).

In digital electronics adding of two-bit binary numbers can be possible by using half adder. And if the input sequence has a three-bit sequence, then the addition process can be completed by using a full adder. But if the numbers of bits are more in the input sequence then the process can be completed by using half adder. Because full adder cannot be able to complete the addition operation. So these drawbacks can be overcome by using “Ripple Carry Adder”. It’s a unique type of logic circuit used for adding the N-bit numbers in digital operations. This article describes an overview of what is ripple-carry-adder and its operation.

2. LITERATURE SURVEY

Amit Kumar Panda at al [1] are presented by a 3-operand binary adder is the basic functional unit to perform the modular arithmetic in various secure communication in the presence of malicious third-parties and pseudorandom number generator (PRNG) algorithms. Carry-save adder (CS3A) is the widely used technique to perform the 3-operand addition. However, the ripple-carry adder phase in the CS3A leads to a high propagation delay of O(n). Moreover, a parallel prefix two-operand adder such as Han-Carlson (HCA) can also be used for 3-operand addition that considerably reduces the critical path delay at the cost of supplementary hardware. Hence, a new high-speed and area-efficient adder architecture is proposed using pre-compute bitwise addition followed by carry-prefix addition logic to perform the 3-operand binary addition that consumes substantially less area, low power and radically reduces the adder delay to O(log $_2$ n). The proposed architecture is implemented on the FPGA device for practical validation and also synthesized with the commercially available 32nm CMOS technology records. The post-synthesis results of the proposed adder reported 3.12, 5.31 and 9.28 times faster than the CS3A for 32-, 64- and 128- bit architecture respectively. Moreover, it has a lesser area, lower power dissipation and smaller delay than the HC3A adder. Also, the proposed adder achieves the lowest ADP and PDP than the existing 3-operand adder techniques. Darjn Esposito et al [2], described by A variable latency adder (VLA) reduces average addition occasion by using speculation: the exact arithmetic function is replaced by an approximated one, that is faster and gives correct results most of the times. When assumption fails, an error detection and correction circuit gives the correct result in the following clock cycle. Previous papers investigate VLAs based on Kogge-Stone, Han-Carlson or carry select topologies, speculating that carry propagation involves only a few uninterrupted bits. In several applications using 2's complement demonstration, however, operands have a Gaussian distribution and a nontrivial portion of carry chains can be as long as the adder size. In this paper we propose five novel VLA architectures, based on Brent-Kung, Ladner-Fisher, Sklansky, Hybrid Han-Carlson, and Carry increment parallel-prefix topologies. Moreover, we present a new proficient error detection and correction technique, that makes proposed VLAs suitable for applications using 2's complement representation. In order to investigate VLAs performances, proposed architectures have been synthesized using the UMC 65 nm library, for operand lengths ranging from 32 to 128 bits. Obtained results show that proposed VLAs outperform previous speculative architectures and standard (non-speculative) adders when high-speed is required. Dilip P. Vasudevan et al [3] are obtained by Carry-select adders are one of the faster than other types of adders. This manuscript proposes a scheme that encodes the sum bits using two-rail codes; the encoded sum bits are then checked by self-checking checkers. The multiplexers used in the adder are also totally self-checking. The scheme is illustrated with the realization of a 2-bit carry select adder that can detect all single stuck-at faults on-line; the discovery of double faults is not guaranteed. Adders of random size can be constructed by cascading the appropriate number of such 2-bit adders. A range of adders from 4 to 128 bits is designed using this move toward employing a 0.5-mum CMOS technology. The transistor transparency in implementing these self-checking adders varies from 19.51% to 20.94%, and the area overhead varies from 16.07% to 20.67% compared to adders without built-in self-checking capability. Bader Alhazmi et al [4] is introduced by Addition is an necessary operation in all cryptographic algorithms. Higher levels of security require larger key sizes and this becomes a limiting factor in GF(p) using large integers because of the carry propagation problem. We propose a novel and efficient attribute-based large integer representation scheme suitable for large integers commonly used in cryptography such as the five NIST primes and the Pierpont primes used in supersingular isogeny Diffie-Hellman (SIDH) for post-quantum cryptography. Algorithms are proposed for this new representation to implement arithmetic operations such as two's complement, addition/subtraction, comparison, sign detection, and modular reduction. Algorithms are also developed for converting binary numbers to attribute representation and vice versa. The extensive numerical simulations were done to verify the routine of the new number representation. Results show that addition is done faster in our proposed illustration when compared with binary and residue number system (RNS)-based additions. Attribute addition outperformed RNS addition for all values of m where $128 \leq m \leq 32768$ bits for all machine word sizes w where $4 \leq w \leq 128$ bits. Attribute-based addition outperforms Kogge-Stone binary adders for a wide range of m when w is small. For increasing values of w, the speed advantages are obvious only for large values of m. This makes the proposed number representation suitable for implementing cryptographic applications in embedded processors for IoT and consumer electronic devices where w is small. Ned Bingham et al [5] are presented by A primary operator in modern computational systems has a multitude of highly optimized implementations. In general-purpose systems, the width has grown to 128 bits, putting pressure on designers to use complicated carry lookahead or tree adders to maintain throughput while sacrificing area and energy. However, the typical workload mostly exercises the lower 10-15 bits. This leaves many devices on and unused during normal operation, reducing the overall performance. We hypothesize that bit- or digit-serial implementations for arbitrary-length streams represent an opportunity to decrease the overall energy usage while increasing the throughput/area efficiency of the system and verify this hypothesis by constructing an asynchronous digit-serial adder for comparison against its bit-parallel counterparts.

In this manuscript, a new binary counter design is proposed. It uses 3-bit stacking circuits, which group all of the “1” bits together, followed by a novel symmetric method to combine pairs of 3-bit stacks into 6-bit stacks. The bit stacks are then converted to binary counts, producing 6:3 counter circuits with no xor gates on the critical path. This avoidance of xor gates results in faster designs with efficient power and area utilization. In VLSI simulations, the proposed counters are 30% faster than existing parallel counters and also consume less power than other higher order counters. Additionally, using the proposed counters in existing counter-based Wallace tree multiplier architectures reduces latency and power consumption for 64 and 128-bit multipliers are presented by Christopher Fritz et al [6]. Itamar Levi et al [7] is constructed by The recently proposed dual mode logic (DML) gates family enables a very high level of energy-delay optimization flexibility at the gate level. In this paper, this flexibility is utilized to improve energy efficiency and performance of combinatorial circuits by manipulating their critical and noncritical paths. An approach that locates the design's critical paths and operates these paths in the boosted performance mode is proposed. The noncritical paths are operated in the low energy DML mode, which does not affect the performance of the design, but allows major energy consumption reduction. The proposed approach is analyzed on a 128 bit carry skip adder. Simulations, carried out in a standard 40 nm digital CMOS process with , show that the proposed approach allows performance improvement of X2 along with reduction of energy consumption of X2.5, as compared with a standard CMOS implementation. At , improvements of 1.3X and 1.5X in performance and energy are achieved, respectively. R. Heaton et al [9] are described by An array processing chip integrating 128 bit-serial processing elements (PEs) on a single die is discussed. Each PE has a 16-function logic unit, a single-bit adder, a 32-b variable-length shift register, and 1 kb of local RAM. Logic in each PE provides the ability to mask PEs independently. A modified grid interconnection scheme allows each PE to communicate with each of its eight nearest neighbors. A 32-b bus is used to transfer data to and from the array in a single cycle. Instruction execution is pipelined, enabling all instructions to be executed in a single cycle. The 1- µm CMOS design contains over 1.1-million transistors on an 11.0-mm*11.7-mm die. Santosh Ghosh et al [8] are presented by a devoted to the design and the physical security of a parallel dual-core flexible cryptoprocessor for computing pairings over Barreto-Naehrig (BN) curves. The proposed design is specially optimized for field-programmable gate-array (FPGA) platforms. The design explores the in-built features of an FPGA device for achieving an efficient cryptoprocessor for computing 128-bit secure pairings. The work further pinpoints the susceptibility of those pairing computations against side-channel attacks and demonstrates experimentally that power consumptions of such devices can be used to attack these ciphers. Finally, we suggest a suitable countermeasure to overcome the individual weaknesses. The proposed secure cryptoprocessor needs 1,730,000, 1, 20,6000, and 8,21,000 cycles for the computation of optimal pairings, respectively. The implementation results on a Virtex-6 FPGA device hows that it consumes 23 k Slices and computes the respective pairings in 11.93ms, 8.32ms, and 5.66 ms. This manuscript are Approximate arithmetic has recently emerged as a promising paradigm for many imprecision-tolerant applications. It can offer substantial reductions in circuit complexity, delay, and energy consumption by relaxing accuracy requirements. In this paper, we propose a novel energy-efficient approximate multiplier design using a significance-driven logic compression (SDLC) approach. Fundamental to this approach is an algorithmic and configurable lossy compression of the partial product rows based on their progressive bit significance. This is followed by the commutative remapping of the resulting product terms to reduce the number of product rows are explained by Issa Qiqieh et al [10]. Jun Han et al [11] are obtained by Pairings are attractive and competitive cryptographic primitives for establishing various novel and powerful information security schemes. This manuscript presents a flexible and high-performance processor for cryptographic pairings over pairing-friendly curves at high security levels. Yujin Hyun et al [12] are elaborately A synchronous binary counter is one of the basic components widely used in VLSI design, and it is required to be fast and support a wide bit-width in many applications. However, most of the previous counters are associated with a limited counting rate due to large fan-outs and long carry chains, especially when the counter size is not small. This brief proposes a new fast structure for synchronous binary counting, which has a minimal counting period for practical counter sizes ranging from 8 to 128 bits.

3. RIPPLE CARRY ADDERS

A structure of multiple full adders is cascaded in a manner to gives the results of the addition of an n bit binary sequence. This adder includes cascaded full adders in its structure so, the carry will be generated at every full adder stage in a ripple-carry adder circuit. These carry output at each full adder stage is forwarded to its next full adder and there applied as a carry input to it. This process continues up to its last full adder stage. So, each carry output bit is rippled to the next stage of a full adder. By this reason, it is named as “RIPPLE CARRY ADDER”. The most important feature of it is to add the input bit sequences whether the sequence is 4 bit or 5 bit or any.

“One of the most important point to be considered in this carry adder is the final output is known only after the carry outputs are generated by each full adder stage and forwarded to its next stage. So there will be a delay to get the result with using of this carry adder”. There are various types in ripple-carry adders. like 4, 8, 16, 32, 64 and 128-bit are respectively.

3.1. Ripple Carry Adder Truth Table

The Below truth table shows the output values for the possible combinations of all inputs for 4-bit ripple-carry-adder. As shown in Table 1.

Table 1: Ripple Carry Adder Truth Table

S. No.	**Input A Sequence**				**Input B Sequence**				**Sum output**				**Cout**
Combination	**A_3**	**A_2**	**A_1**	**A_0**	**B_3**	**B_2**	**B_1**	**B_0**	**S_3**	**S_2**	**S_1**	**S_0**	**Carry**
1	0	0	0	0	0	0	0	0	0	0	0	0	$C_0=0$
2	0	0	0	1	0	0	0	1	0	0	1	0	$C_1=0$
3	0	0	1	0	0	0	1	0	0	1	0	0	$C_2=0$
4	0	0	1	1	0	0	1	1	0	1	1	0	$C_3=0$
5	0	1	0	0	0	1	0	0	1	0	0	0	$C_4=0$
6	0	1	0	1	0	1	0	1	1	0	1	0	$C_5=0$
7	0	1	1	0	0	1	1	0	1	1	0	0	$C_6=0$
8	0	1	1	1	0	1	1	1	1	1	1	0	$C_7=0$
9	1	0	0	0	1	0	0	0	0	0	0	0	$C_8=1$
10	1	0	0	1	1	0	0	1	0	0	1	0	$C_9=1$
11	1	0	1	0	1	0	1	0	0	1	0	0	$C_{10}=1$
12	1	0	1	1	1	0	1	1	0	1	1	0	$C_{11}=1$
13	1	1	0	0	1	1	0	0	1	0	0	0	$C_{12}=1$
14	1	1	0	1	1	1	0	1	1	0	1	0	$C_{13}=1$
15	1	1	1	0	1	1	1	0	1	1	0	0	$C_{14}=1$
16	1	1	1	1	1	1	1	1	1	1	1	0	$C_{15}=1$

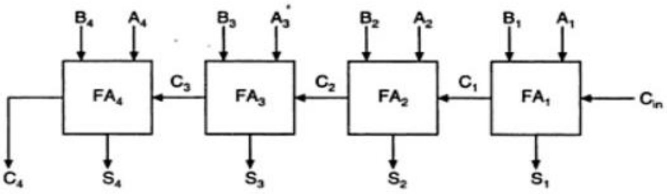


Figure 1. 4-bit Ripple Carry Adders.

3.2 8-bit Ripple Carry Adder

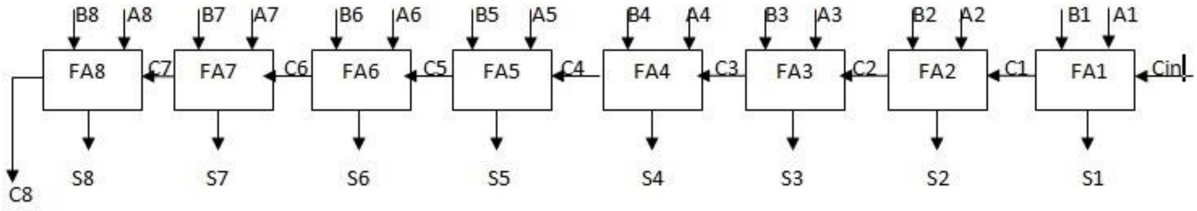


Figure 2. 8-bit Ripple Carry Adder

3.3 16-bit Ripple Carry Adder

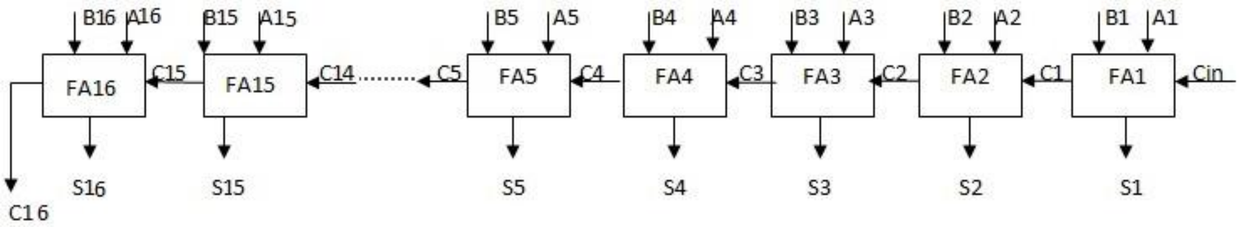


Figure 3. 16-bit Ripple Carry Adder

4. ARCHITECTURE OF RIPPLE CARRY ADDERS

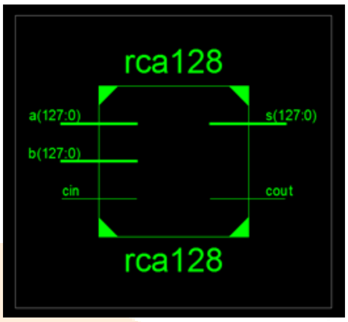


Figure 4. 128-bit Ripple Carry Adder (RCA) RTL Diagram

The above Figure 4 are 128-bit ripple carry adder RTL diagram designed by HDL code for 128-bit is synthesized by Xilinx ISE 12.2 version. It shows input a[127:0] sequence from 127 to 0 binary inputs, input b[127:0] sequence from 127 to 0 binary inputs with initial carry $c_{in.}$ and similarly output s[127:0] [127:0] sequence from 127 to 0 binary outputs with Carry out C_{out}.

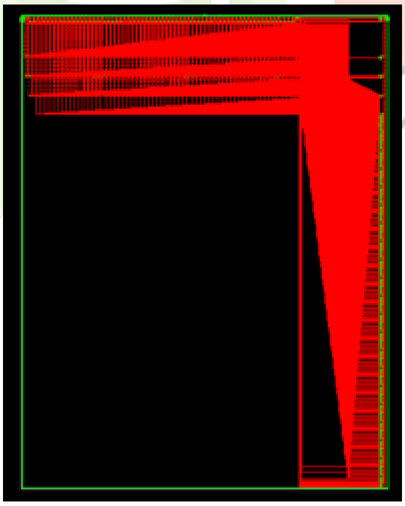


Figure 5. 128-bit Ripple Carry Adder (RCA) Technology RTL Diagram

The 128-bit Ripple Carry Adder (RCA) Technology View RTL Diagram as shown in Figure 5. It shows single bit 2 input xor2 gate are 256, Basic Element of Logic (BELs) are 192, 3 input Look up table (LUTs) is 64, 5 input LUT is 128 and the input blocks (IBs) are 257, the output blocks (OBs) are 129 and the input-output blocks (IOBs) are 386 as in Table 2.

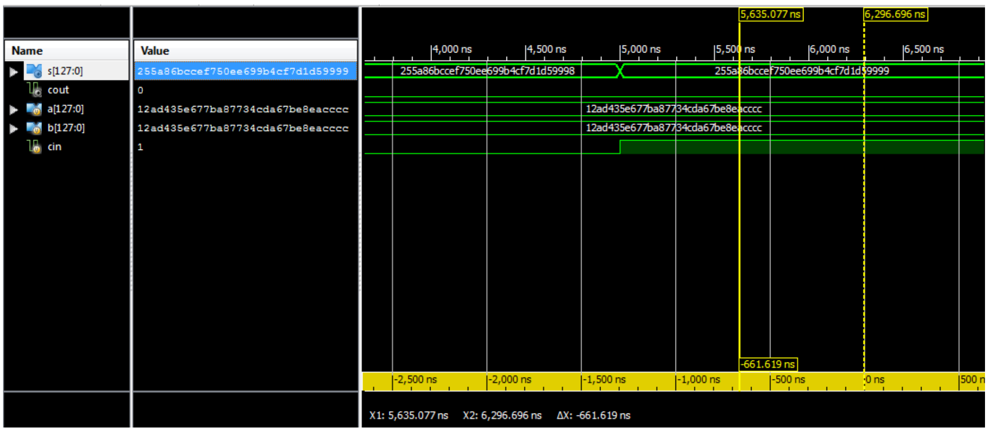
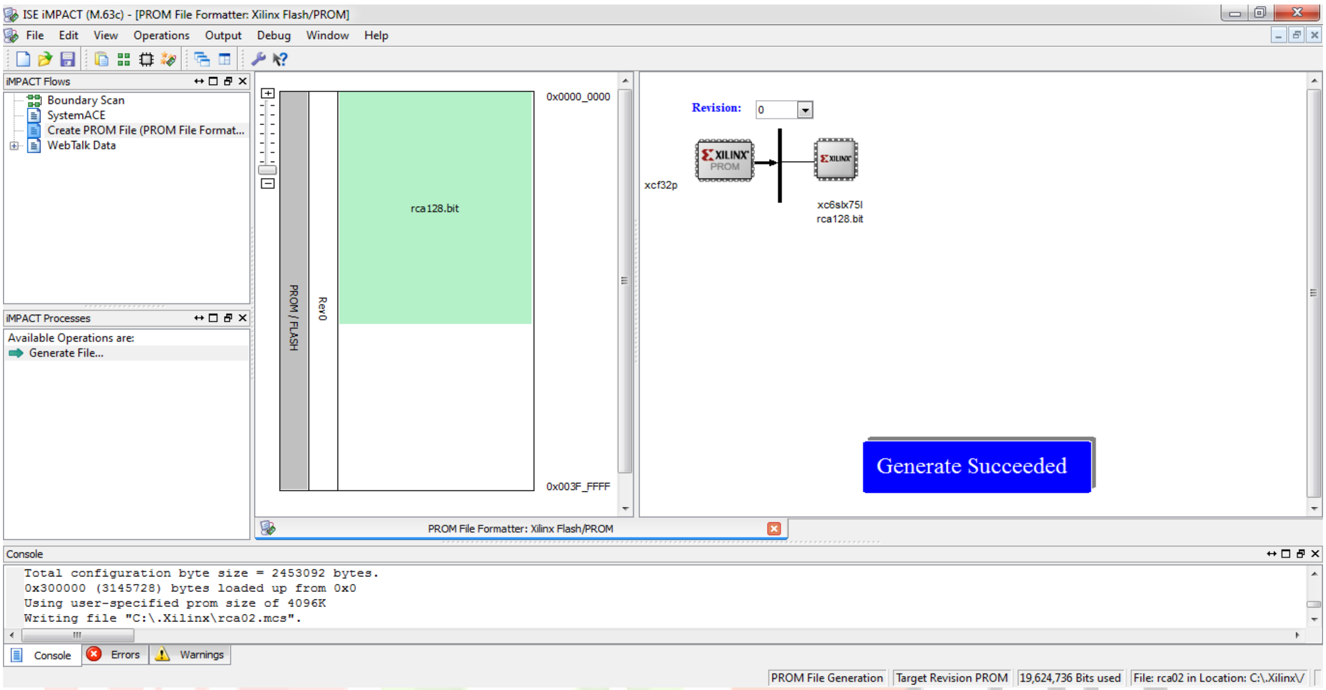

Name
Value
s[127:0]
255a86bccef750ee699b4cf7d1d59999
cout
0
a[127:0]
12ad435e677ba87734cda67be8eacccc
b[127:0]
12ad435e677ba87734cda67be8eacccc
cin
1
5,635.077 ns
6,296.696 ns
4,000 ns
4,500 ns
5,000 ns
5,500 ns
6,000 ns
6,500 ns
255a86bccef750ee699b4cf7d1d59998
255a86bccef750ee699b4cf7d1d59999
12ad435e677ba87734cda67be8eacccc
12ad435e677ba87734cda67be8eacccc
-661.619 ns
-2,500 ns
-2,000 ns
-1,500 ns
-1,000 ns
-500 ns
0 ns
500 n
X1: 5,635.077 ns X2: 6,296.696 ns ΔX: -661.619 ns

Figure 6. 128-bit Ripple Carry Adder (RCA) Timing Diagram



```
Total configuration byte size = 2453092 bytes.
0x300000 (3145728) bytes loaded up from 0x0
Using user-specified prom size of 4096K
Writing file "C:\.Xilinx\rca02.mcs".
```

Figure 7. 128-bit Ripple Carry Adder (RCA) implement in FPGA

As shown In Figure 4. 128-bit ripple carry adder diagram using HDL Code is implemented in Xilinx Software ISE 12.2i. Full adder HDL code synthesized with Register Transfer Level (RTL) diagram and Technology RTL is synthesized, post and route generated as shown in Figure 5 and simulated 128 bit full adder output waveform a[127:0]=16’H12AD435E677BA87734CDA67BE8EACCCC, b[127:0]=16’H AD435E677BA87734CDA67BE8EACCCC and cin=1’b0 then sum[127:0]=16’H, 255A86BCCEF750EE699B4CF7D1D59998 cout=1’b0 else a[127:0]=16’H12AD435E677BA87734CDA67BE8EACCCC, b[127:0]=16’H12AD435E677BA87734CDA67BE8EACCCC and cin = 1’b1 then sum[127:0]=16’H255A86BCCEF750EE699B4CF7D1D59999, cout=1’b1 as shown in Figure 6. Also to implemented in FPGA Vertex6 Low power Device as shown in Figure 7.

5. RESULTS AND DISCUSSIONS

We use the Xilinx ISE 12.2i FPGA design and a review of Ripple carry adders shows that the RCA Architecture 128-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 4. Total estimated power consumption is 7mW, Quiescent Voltage is Vcco33 3.30V, Quiescent Current is 2mA.

TABLE 2. RCAs performance Parameters utilization summary:

No. of bits	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
128-bit	256	192	64	128	257	129	386

The 128-bit Ripple carry adder wrote in HDL code are implemented in Xilinx ISE software to produce the results in Table 1 and as shown in figure 8 have been discussed here single bit 2 input xor2 gate are 256, Basic Element of Logic (BELs) are 192, 3 input Look up table (LUTs) is 64, 5 input LUT is 128 and the input blocks (IBs) are 257, the output blocks (OBs) are 129 and the input-output blocks (IOBs) are 386. And also power consumption 487 mWatts with Ambient Temperature 50ºC.

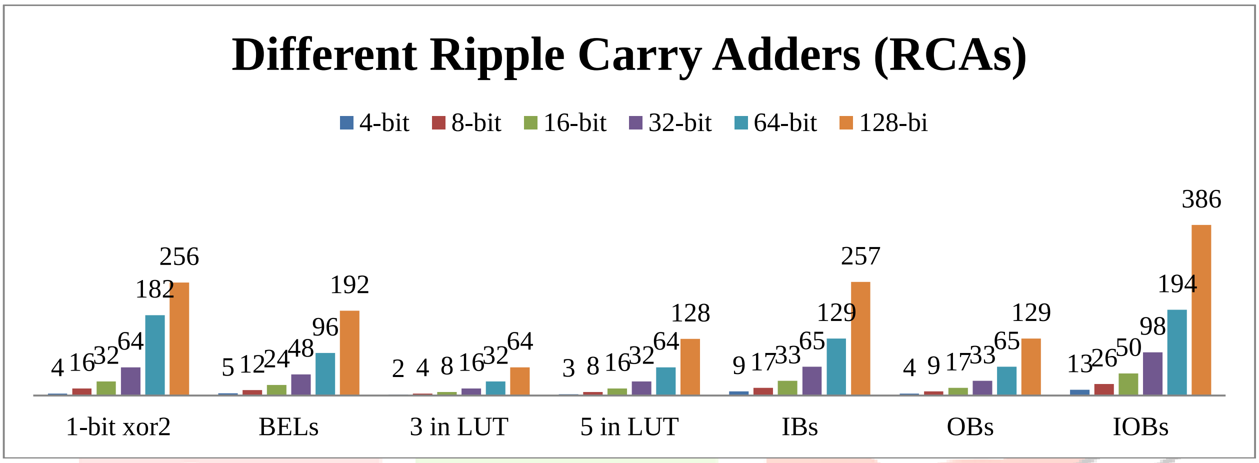


Figure 8. RCAs performance Parameters utilization summary

TABLE 2. RCAs performance Parameters Timing summary:

No. of bits	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
128-bit	35.448	3.398	32.050	9.6	90.4

The 128-bit Ripple carry adder (RCA) maximum combinational path delay 35.448ns in logically 3.39ns with in percentage is 9.6% and routing 32.5ns with in percentage are 90.4%. discussed as on Table 2 and as shown in Figure 9.

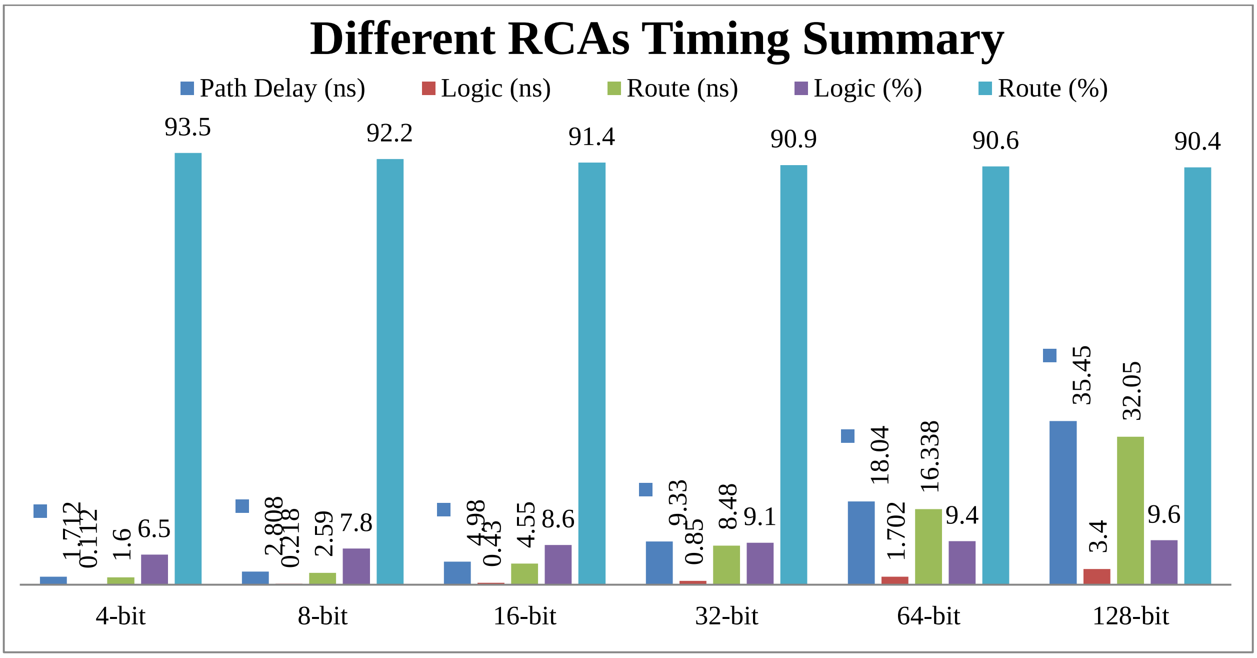


Figure 9. RCAs performance Parameters Timing summary

TABLE 3. RCAs performance Parameters Timing constraint: Default path analysis

No. of bits	destination ports (y)	Levels of Logic
4-bit	**4**	**4**
8-bit	**9**	**6**
16-bit	**17**	**10**
32-bit	**33**	**18**
64-bit	**65**	**34**
128-bit	**129**	**66**

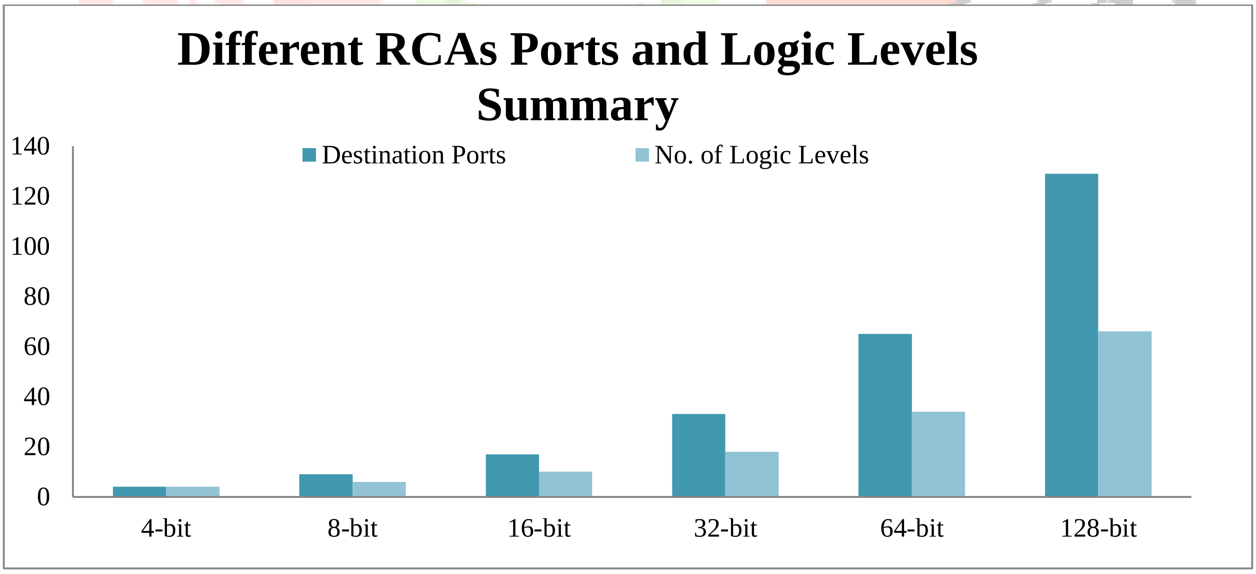


Figure 10. RCAs performance Parameters Different RCAs Ports and Logic Levels Summary

The 128-bit Timing constraints are Default path analysis, in Xilinx ISE 12.2i **Total number of paths is 1687** and destination ports are 129. And ration between Total number of paths and destination ports are 16897 / 129 = 13.08, with in percentage 7.65 (%) and Logic level are 66 discussed on Table 3 and as shown in Figure 10.

6. CONCULUSIONS

In this Manuscript discuss the Efficient 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 128-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 manuscript to discuss the RCAs performance of Synthesis RTL architecture namely for 128-bit are 192 SLICE, and Maximum Combinational logic path delay for are 35.45nsec using Xilinx ISE Design Suite 12.2. Here we single bit 2 input xor2 gate are 256, Basic Element of Logic (BELs) are 192, 3 input Look up table (LUTs) is 64, 5 input LUT is 128 and the input blocks (IBs) are 257, the output blocks (OBs) are 129 and the input-output blocks (IOBs) are 386. And also power consumption 487 mWatts with Ambient Temperature 50ºC. and also discussed The 128-bit Ripple carry adder (RCA) maximum combinational path delay 35.448ns in logically 3.39ns with in percentage is 9.6% and routing 32.5ns with in percentage are 90.4%.

CONFLICT OF INTERESTS

The authors are declare that there is no conflicts of the interests are regarding the publication of this manuscript.

ACKNOWLEDGEMENT

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

REFERENCES

[1]. Amit Kumar Panda; Rakesh Palisetty; Kailash Chandra Ray, “High-Speed Area-Efficient VLSI Architecture of Three-Operand Binary Adder”, IEEE Transactions on Circuits and Systems I, Vol. 67, Issue. 11, pp: 3944 – 3953. Nov. 2020, DOI: 10.1109/TCSI.2020.3016275.

[2]. Darjn Esposito; Davide De Caro; Antonio Giuseppe Maria Strollo, “Variable Latency Speculative Parallel Prefix Adders for Unsigned and Signed Operands”, IEEE Transactions on Circuits and Systems I, Vol. 63, Issue. 8, pp.1200 – 1209. Aug. 2016, DOI: 10.1109/TCSI.2016.2564699.

[3]. Dilip P. Vasudevan; Parag K. Lala; James Patrick Parkerson, “Self-Checking Carry-Select Adder Design Based on Two-Rail Encoding” IEEE Transactions on Circuits and Systems I, Vol. 54, Issue. 12, pp.2696 – 2705. Dec. 2007, DOI: 10.1109/TCSI.2007.910537.

[4]. Bader Alhazmi; Fayez Gebali, “Fast Large Integer Modular Addition in GF(p) Using Novel Attribute-Based Representation” IEEE Access, Vol. 7, pp. 58704 – 58719. May 2019. DOI: 10.1109/ACCESS.2019.2914641.

[5]. Ned Bingham; Rajit Manohar, “Self-Timed Adaptive Digit-Serial Addition”, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol.27, Issue.9, pp.2131 – 214. Sep. 2019.

[6]. Christopher Fritz; Adly T. Fam, “Fast Binary Counters Based on Symmetric Stacking”, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 25, Issue. 10, pp.2971 – 2975, Oct 2017. DOI: 10.1109/TVLSI.2017.2723475.

[7]. Itamar Levi; Alexander Fish, “Dual Mode Logic—Design for Energy Efficiency and High Performance”, IEEE Access, Vol.1. pp.258 – 265, may 2013. DOI: 10.1109/ACCESS.2013.2262015.

[8]. Santosh Ghosh; Debdeep Mukhopadhyay; Dipanwita Roychowdhury, “Secure Dual-Core Cryptoprocessor for Pairings Over Barreto-Naehrig Curves on FPGA Platform”, IEEE Transactions on Very Large Scale Integration (VLSI) Systems,Vol.21, Issue. 3, pp. 434 – 442. Mar 2013.

[9]. R. Heaton; D. Blevins; E. Davis, “A bit-serial VLSI array processing chip for image processing”, IEEE Journal of Solid-State Circuits, Vol. 25, Issue. 2, pp.364 – 368. Apr. 1990. DOI: 10.1109/4.52157.

[10]. Issa Qiqieh; Rishad Shafik; Ghaith Tarawneh; Danil Sokolov; Shidhartha Das; Alexandre Yakov, “Significance-Driven Logic Compression for Energy-Efficient Multiplier Design”, IEEE Journal on Emerging and Selected Topics in Circuits and Systems, Vol. 8, Issue. 3, pp.417 – 430. Sep. 2018. DOI: 10.1109/JETCAS.2018.2846410.

[11]. Jun Han; Yang Li; Zhiyi Yu; Xiaoyang Zeng, “A 65 nm Cryptographic Processor for High Speed Pairing Computation”, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 23, Issue. 4, pp. 692 – 701. Apr 2015. DOI: 10.1109/TVLSI.2014.2316514.

[12]. Yujin Hyun; In-Cheol Park, “Constant-Time Synchronous Binary Counter With Minimal Clock Period”, IEEE Transactions on Circuits and Systems II: Express Briefs, Vol. 68, Issue, 7, pp. 2645 – 2649. Jul 2017. DOI: 10.1109/TCSII.2021.3054014.