International Journal of Research
Available at
https://edupediapublications.org/journals

e-ISSN: 2348-6848 p-ISSN: 2348-795X
Volume 04 Issue 10
September 2017

Design and Analysis of Multi Precision Arithmetic Adders

[1]R. Laxman, [2]G.Viswanathreddy
[1] Assistant Professor, [2]M.Tech Student
ECE Department, Global college of Engineering &Technology,
Kadapa, A. P, India.

Abstract—Arithmetic adder is the most important basic element for many digital applications. In paper different types of adders are taken for such as Ripple Carry Adder, Carry Save adder, Carry Look ahead adder, Carry Increment adder, Carry Select adder and Carry Skip adder. Here in this paper introducing a novel technique for designing a new Carry Select adder for multi precision arithmetic circuits. By using this technique improvements has been achieved like low latency and less power consumption and along with less gate count. Experimentally synthesized and simulated by using Xilinx ISE14.7, also tested in SPARTAN3E, XC3S1600E with speed of -5.

Keywords— ASIC, DSP, RCA, CLA, CSKA, CSA, and CSLA.

I. INTRODUCTION

Design of high speed digital circuits targeted to achieve in terms of throughput, latency and power efficient [1].Arithmetic Logic Unit (ALU) is the main functional block in several areas like digital systems design, Digital Signal Processing (DSP),Data Processing units, Microprocessors, Micro controllers and cryptographic arithmetic applications and many more.

Researchers are concentrated to improve the fast addition [1, 8] operation for arithmetic designs. Addition and subtraction are the most basic elements in ALU. An adder circuit became significant hardware element for any operation like subtraction, multiplication and division including complements (1's and 2's) [18, 33], encoding, and decoding etc.

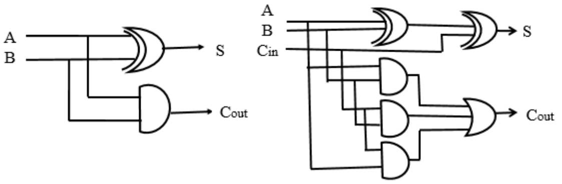


Fig 1: Half adder & Full adder

An addition operation is a process which can add two or more numbers resulting output is Sum and Carry. Any research in combinational circuit starts with the basic building blocks, namely half adder and Full adder. In adder circuit contains several logic gates like AND, OR, NAND, NOR, XOR. The following diagrams of half adder and full adder as shown in figure-1.

Half adder is one type of adder which is designed using two logic gates, namely AND and XOR. Full Adder is one type of adder which is designed using three logic gates, which are AND,OR and XOR, these circuit has 3 inputs, A, B, C in [2,25,29,33,36] and the resultant outputs are Sum and Carry.

TABLE I: GATE COUNT & GATE DELAY

comb circuit	Gate Delay	Gate Area
XOR	3	5
2:1 MUX	3	4
Half Adder	3	6
Full Adder	6	14

This paper is organized as follows: section II Discussion of different type of adders ranging from 8-bit to 1024-bit,in section III discussion of proposed carry select adder and in section IV discussion in performance analysis of different types of adders and proposed carry select adder, in section V conclusion of results and performance of this research work.

II. TYPES OF ADDERS

In this section different types of adders ranging from 8- bit to 1024-bits are discussing like Ripple Carry Adder, Carry Look ahead adder, Carry Save adder, Carry Skip adder and Carry Select adder.

A. Ripple Carry adder

Ripple Carry adders (RCAs) [1, 7, 24, 26, 30, 33] are used to perform parallel addition. These RCAs are designed by cascading of full adders and those are connected in series. Each full adder having three inputs and producing two outputs such as sum and carry.

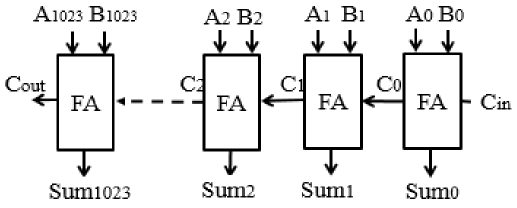


Fig 2: 1024-bit Ripple carry adder

In RCA, the output carry of the full adder is fed with the carry in of the next full adder. The carry signal ripples throughout all stages in RCA from lower significant bit to higher significant bit. The drawback of the RCA is increases the delay because each and every adder block is depend up on the previous carry input. The following RCA block as shows in figure-2.

B. Carry LookAhead Adder (CLA)

CLA [2] is to overcome the rippling effect and Propagation delay occurred in RCA. This CLA reduces the gate delay and it makes fast operation. Propagation and generation are carried out at first stage, internal carry is generated at second stage and the final stage sum can be calculated [3,21,22,23,27,30,33]. The following CLA structure and computation as shown in below.

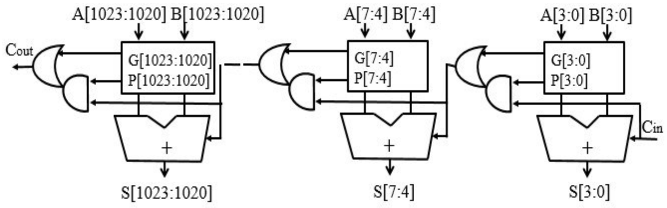


Fig 3: 1024-bit Carry look ahead adder

Step-1: Generation & propagation

G[i] =A[i] & B[i]. P [i] = A[i] xor B[i]

Step-2: internal carry generation

C[i] = G[i] + P [i].C[i − 1]

Step-3: Sum generatio n

s[i] = A[i]XOR B[i] xor C[i − 1]

C. Carry Save adder

The Carry Save adder (CSA) consist of n-full adders and it performs addition operation. In these carry save adder technique, where carry is not propagated, instead carry will be stored at current stage. This carry is connecting to the next stage as addend value, So that delays can be reduced [3, 6, 12, 14-17].

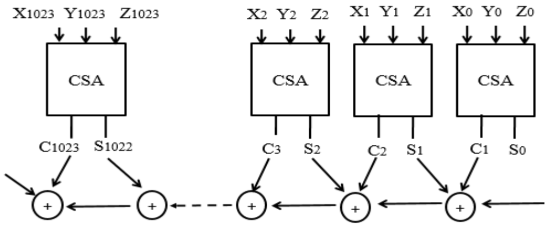


Fig 4: 1024-bit Carry save adder

The main use of the carry save adder is utilized in multipliers for high speed Digital Signal processing appli cations[12,27,30,33], also used as subtraction operation in division circuits to get the exact remainder. CSA block diagram as shows in figure-4.

D. Carry Skip adder

Carry Skip adder (CSKA) [1,3,27,30,33] is a process of skipping the carry and makes it fast operation. Carry Skip adder consist of RCAs with a special speed-up carry chain called Carry skip. The CSKA computation steps and block diagram as shown in below.

$$P [i, i+3] = (P\ i+3) * (P\ i+2) * (P\ i+1) * (P\ i),$$

$$\text{Carry} = C_{i+4} + (P\ [i, i+3]) * C_{in}.$$

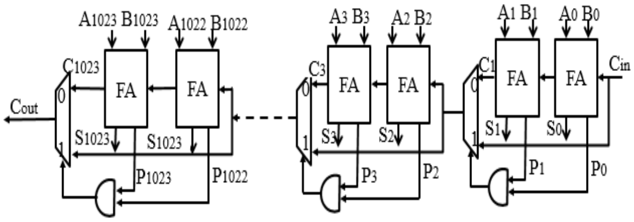


Fig 5: 1024-bit Carry skip adder

E. Carry Select adder

Carry Select adder (CSLA) is to [9-11,19,20,28,31, and 32] perform the parallel computation by selecting carry input either '0' or '1'. Depending upon the carry input, multiplexers selects the carry input to produce the sum output and to propagate the carry input for the next stage. Therefore this CSLA mechanism delays can be reduced. The amount of speed in Cary Select adder is 40%-90% faster than RCA's. The block diagram of CSLA as shown in below.

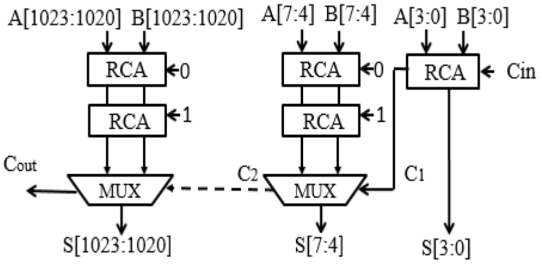


Fig 6: 1024-bit Carry select adder

III. PROPOSED CARRY SELECT ADDER

There are several types of adders which are discussed in previous section, among all the adders, Carry select adder is one of the simplest and fastest adder, but still there is scope to reduce the latency in the carry select adder. The main objective is to optimize the gate delay and gate count in proposed carry select adder [1, 2, 4, 5,9,10, and 13].

The proposed carry select adder (CSLA) is modifies the regular CSLA architecture. The regular CSLA architecture consists of RCAs and multiplexers, by the same way in proposed system, instead of RCAs replaced by the two methods, namely carry look ahead adder (CLA) and carry increment adder (CIA). These two methods are design with NAND gates. By that proposed design can reduces the gate

Count and gate delay when compared to regular CSLA. 4-Bit CIA and 4-bit (NAND) CLA block diagram as shown in figure-7 & figure-8.

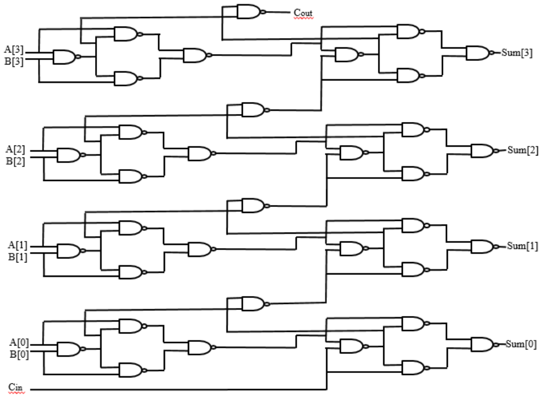


Fig 7: 4-bit NAND CLA

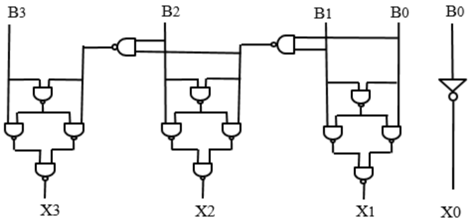


Fig 8: 4-bit Carry increment adder

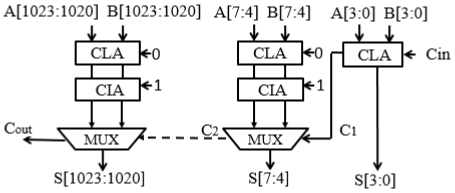


Fig 9: 1024-bit Carry select adder

Theoretical calculations are evaluated in terms of Gate delay (GD) and Gate count (GC) for 4-bit RCA and 4-bit NAND-CLAs. Gate count and Gate delay calculations as shown in table-I.GD & GC for 4-bit RCA is 24 & 56, similarly GD & GC for 4-bit CLA is 12 & 36 and GD and GC for 4-bit CIA is 9 & 16 . Therefore, gate delay and gate count is reduced in proposed CSLA when compared to existing CSLA. Similarly these same calculations are used to compute for 1024-bit CSLA.

IV. PERFORMANCE ANALYSIS

To evaluate the performance analysis of different type of adders ranging from 8-Bit to 1024-Bit. It has been experimentally synthesized and simulated by XILINX ISE14.7 and targeted in to SPARTAN3E, XC3S1600E with speed of -5. The synthesis results are tabled and illustrated as shown in terms of graphs. The performance of comparison has been measured based on synthesis report in terms of device utilization, computational delay and memory utilization.

TABLE II: SYNTHESIS REPORT

Bit size	RCA				CLA			
	Slices	LUTs	Delay (ns)	Memory (MB)	Slices	LUTs	Delay (ns)	Memory (MB)
8	9	16	13	149	6	10	9	149
16	18	32	22	149	23	40	19	149
32	37	64	39	149	35	60	26	150
64	74	128	73	150	69	120	45	151
128	147	256	141	152	138	240	85	152
256	294	512	277	156	276	480	163	157
512	589	1024	548	166	552	960	320	169
1024	1178	2048	1091	185	1104	1920	634	193

The above tabulated values are taken from synthesis report for RCA and CLA generated by the XILINX 14.7 cad tool

TABLE III: SYNTHESIS REPORT

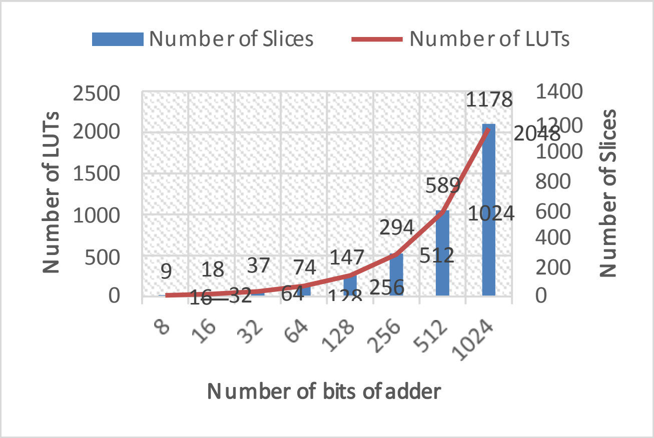
Bit size	CSA				CSKA			
	Slices	LUTs	Delay (ns)	Memory (MB)	Slices	LUTs	Delay (ns)	Memory (MB)
8	12	21	12	149	15	26	11	150
16	23	41	19	149	30	52	15	150
32	46	81	32	150	60	104	24	151
64	91	161	57	152	120	208	43	151
128	182	321	109	154	240	416	80	153
256	364	641	211	162	479	832	153	157
512	545	961	314	168	957	1664	300	165
1024	1453	2561	827	202	1914	3328	596	181

The above tabulated values are taken from synthesis report for CSA and CSKA generated by the XILINX 14.7 tool.

TABLE IV: SYNTHESIS REPORT

Bit size	CSLA				Proposed CSLA			
	Slices	LUTs	Delay (ns)	Memory (MB)	Slices	LUTs	Delay (ns)	Memory (MB)
8	15	26	12	149	11	20	10	49
16	30	52	17	150	23	42	15	150
32	59	104	29	151	46	86	24	151
64	118	208	51	153	93	174	41	153
128	237	416	96	157	187	350	76	158
256	478	832	153	167	375	702	148	169
512	957	1664	300	186	751	1406	289	189
1024	1914	3328	596	225	1503	2814	572	229

The above tabulated values are taken from synthesis report for CSLA and proposed CSLA generated by the XILINX 14.7 tool.



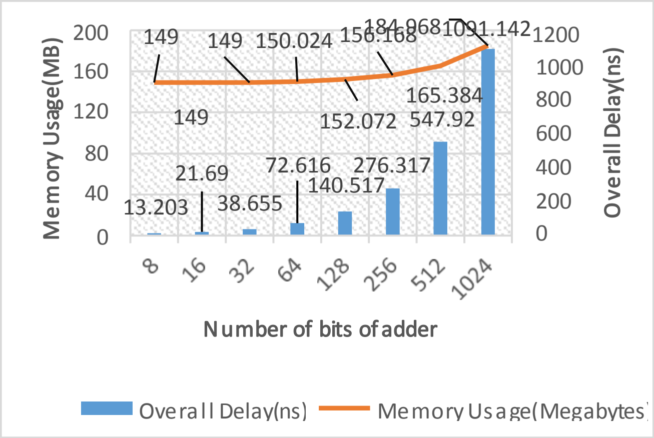
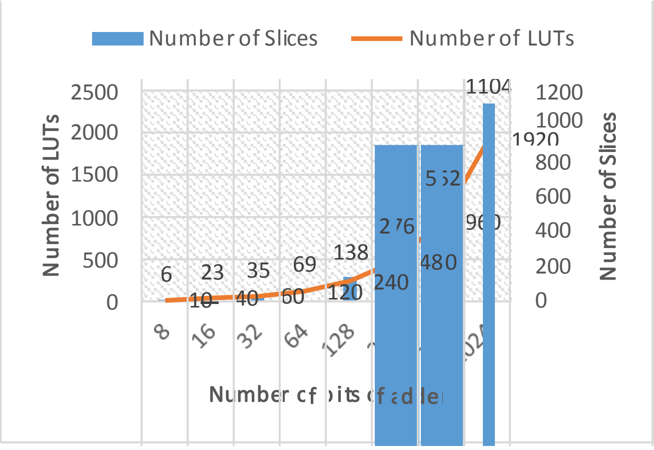


Fig 10(a): Device utilization of RCA

Fig (10 (b): Delay and Memory usage of RCA

The above 10(a) & 10(b) graphs represents how much amount of device utilization, delay and memory usage occupied for 8 to 1024- bit RCA.



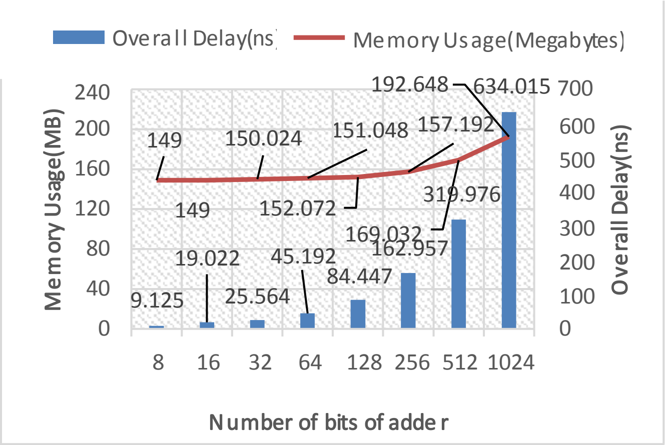
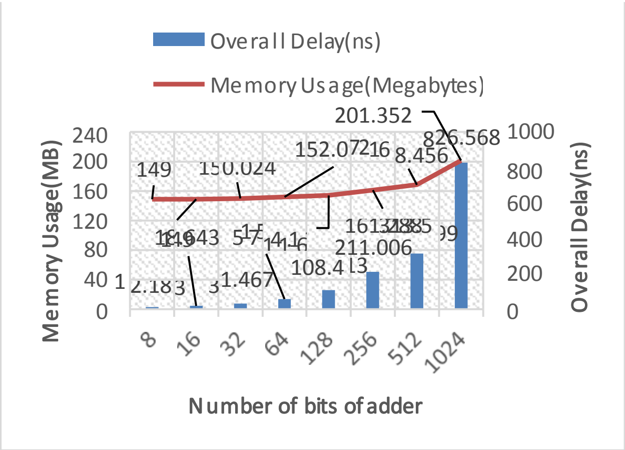


Fig 11 (a): Device utilization of CLA

Fig 11 (b): Delay and Memory usage of CLA

The above 11(a) &11(b) graphs represents how much amount of device utilization, delay and memory usage occupied for 8 to 1024- bit CLA.



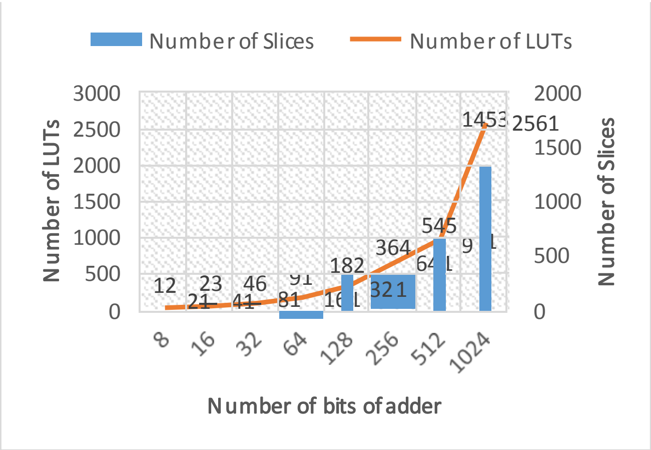
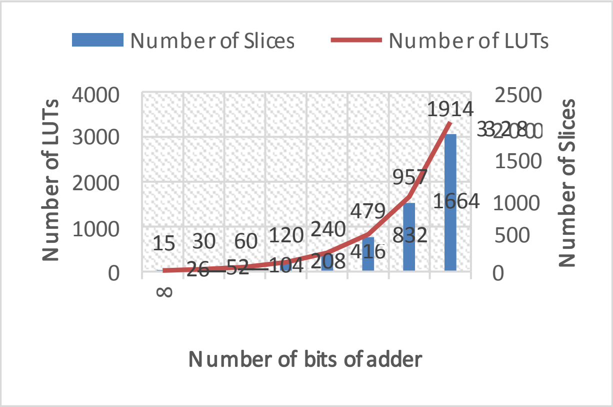


Fig 12 (a): Device utilization of CSA

Fig 12 (b): Delay and Memory usage of CSA

The above 12(a) &12(b) graphs represents how much amount of device utilization, delay and memory usage occupied for 8to1024-bitCSA.



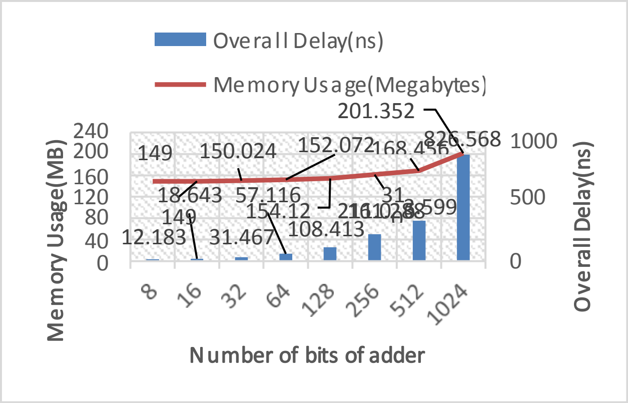
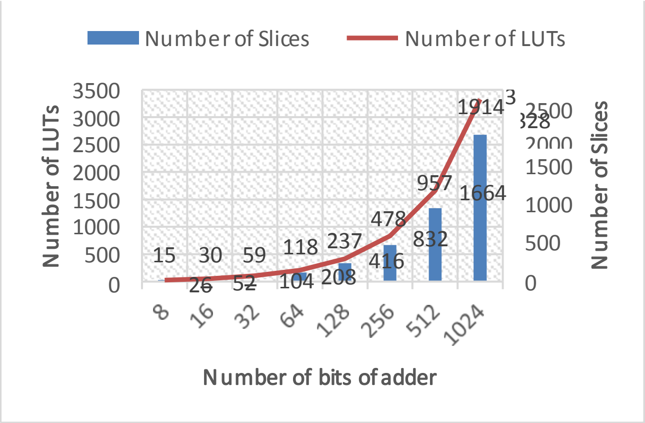


Fig 13(a): Device utilization of CSKA

Fig 13 (b): Delay and Memory Usage of CSKA

The above 13(a) & 13(b) graphs represents how much amount of device utilization, delay and memory usage occupied for 8 to 1024- bit CSKA.



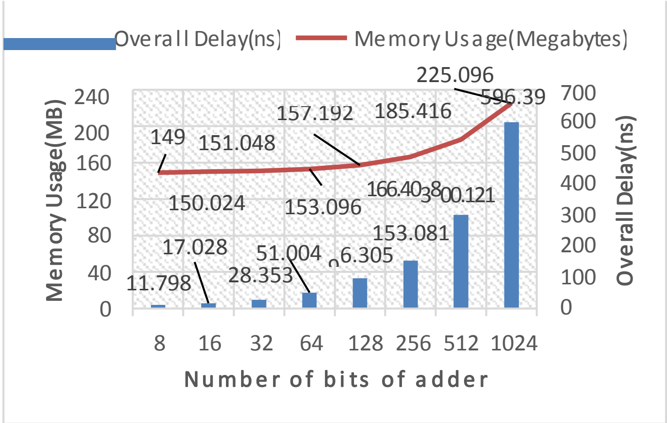
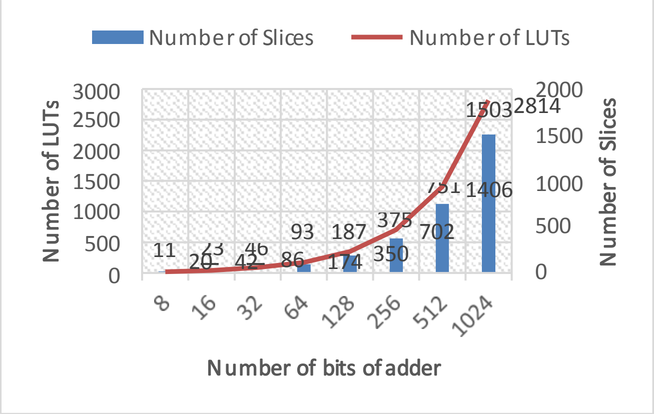


Fig 14 (a): Device utilization of CSLA

Fig 14 (b): Delay and Memory usage of CSLA

The above figure-14(a) & 14(b) shows how much amount of delay and Memory are occupied for 8-bit to 1024 CSLA.



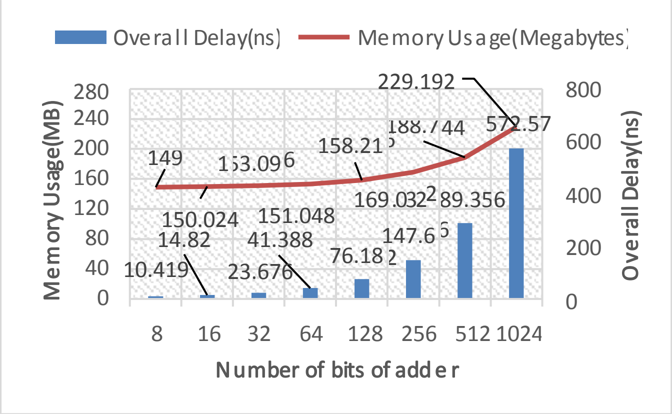


Fig 15 (a): Device utilization of proposed CSLA

Fig 15 (b): Delay and Memory usage of proposed CSLA

The above 15(a) & 15(b) graphs represents how much amount of device utilization, delay and memory usage occupied for 8 to 1024-Bit proposed CSLA. Finally, proposed CSLA shows better results when compared to exist CSLA.

V. CONCLUSION

In this paper, study of different types of adders ranging from 8-bit to 1024-bits, namely RCA, CLA, CSA, CSKA, CSLA and proposed CSLA and these different types of adders are used for multi precision arithmetic circuits. A new type of carry select adder is proposed and it reduces the area and delay when compared to regular CSLA. The performance of these designs are compared and evaluated based on synthesis report generated by XILINX 14.7. Hence after comparison of among different types of adders, proposed CSLA shows better performance in terms of area and latency.

ACKNOWLEDGMENT

The authors would like to thank Principal and the management of Global college of Engineering &Technology, for their kind support to carry out the research work and encouragement for the publication.

REFERENCES

[1] R.Uma, Vidya Vijayan, M.Mohanapriya, Sharon Paul, “Area, Delay and Power Comparison of Adder Topologies”, International Journal of VLSI Design & Communication Systems, Vol. 3, No. 1, pp. 153-168, Feb 2012.

[2] M.lavanya “Gate Count Comparison of Different 16-Bit Carry Select Adders” DOI: 10.15662/ijareeie.2014.0307034, Vol. 3, Issue 7, July 2014.

[3] Jasbirkaur, lalitsood, “Comparison between Various Types of Adder Topologies” IJCST Vol. 6, Issue 1, Jan - March 2015.

[4] Padma Devi, Ashima Girdher, Balwinder Singh, “Improved Carry Select Adder with Reduced Area and Low Power Consumption”, International Journal of Computer Application, Vol. 3, No. 4, June 2010.

[5] I-Chyn Wey, Cheng-Chen Ho, Yi-Sheng Lin, and Chien-Chang Peng., “An Area-Efficient Carry Select Adder Design by Sharing the Common Boolean Logic Term”, IMECS, vol II ,pp. 1091-1094,March 2012.

[6] Deepa Sinha, Tripti Sharma, K.G.Sharma, Prof.B.P.Singh, “Design and Analysis of low Power 1-bit Full Adder Cell”,IEEE, 2011.

[7] Shubin.V.V,“Analysis and Comparison of Ripple Carry Full Adders by Speed”, Micro/Nano Technologies and Electron Devices (EDM), 2010, International Conference and Seminar on, pp. 132- 135, 2010.

[8] R.Uma, “4-Bit Fast Adder Design: Topology and Layout with Self- Resetting Logic for Low Power VLSI Circuits”, International Journal of Advanced Engineering Sciences and Technology, Vol No.7,Issue No. 2, 197 – 205.