International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 3 (2015) pp. 7215-7227

http://www.ripublication.com

Design and Implementation of Time Efficient Carry Select Adder using FPGA

G.Narmadha[1], Dr. K. Balasubadra[2] , S. Deivasigamani [3]

1-Assistant Professor, Department of Electronics and Communication Engineering, SACS MAVMM Engg. College, Anna University Chennai, India. gnarmadhame@gmail.com
2-Professor, Department Computer Science and Engineering, R.M.D Engg. College, Anna University, Chennai, India. balasubadra@yahoo.com
[3] Faculty of Engineering and Computer Technology, AIMST University, Malaysia
deivasigamani@aimst.edu.my

Abstract

In general, the most basic arithmetic function of any digital system is addition. Therefore, the circuits which are used for addition covers major portion in all the digital designs and also many digital structures are available to perform addition operation. But each and every structure has its own merits and demerits. The efficiency of any digital design is determined on the basis of various parameters such as delay, power, area and time. Among these parameters, area and time play a vital role in the analysis of a digital circuit. The paper proposes a new structure to perform the addition in order to reduce the time and area consumption. The proposed adder structure is analyzed and compared with other structures using Xilinx 14.1i and implemented by using Spartan 3 Field Programmable Gate Array (FPGA) device. It results in lesser time consumption with equal area that can be produced by other structures. Any circuit with this adder will be performed well, when compared with the other adders.

Keywords: TECSA (Time Efficient Carry Select Adder), Digital design, FPGA

1 Introduction

In digital Very Large Scale Integration (VLSI) Circuits, full adder forms are the basic building blocks for all arithmetic operations. Therefore, adder has the great impact in performance of the circuits, which are based on the arithmetic operations. The

characteristics of the digital circuit are analyzed mainly based on the time and area consumption. The various existing adder structures [1-11] such as Ripple Carry Adder (RCA), Carry Look Ahead Adder (CLA), Carry Save Adder (CSA), Carry Select Adder (CSEL), Carry Bypass adder (CBY) and Area Efficient Carry Select Adder (AECSA) are analyzed based on the performance. Among all structures, some structures [1,3,6,8,9] reduce the area occupied by the circuit with the increased delay and some structures [2,4,5,7,10,11] reduce the delay with the increased consumption of area. The proposed adder structure results in optimized performance, that is, the delay is reduced with the equal consumption of area which was observed in normal adder design. So, the proposed full adder called Time Efficient Carry Select Adder (TECSA) results in faster circuit performance. The proposed work is organized as follows. **Section 2** explains the existing adder structure. **Section 3** describes the operation of the proposed TECSA. **Section 4** gives the experimental results of the existing and proposed designs. **Section 5** shows the performance analysis based on the experimental results. **Section 6** concludes the paper.

2 Existing Work

A combinational circuit that performs the addition of three bits are called full adder. Two of the input variables, denoted by a and b, represents the two significant bits to be added. The third input, c, represents the Carry from the previous least significant position. The two output variables are designated as Sum and Carry. The output variables are determined from the arithmetic Sum of the input bits. When all the input bits are zero, the output is '0'. The output Sum is equal to '1' when only one input is equal to '1' or all the three inputs are equal to '1'.

Based on the truth table given in Table1, the output Sum and Carry are described as

$$Sum = a \oplus b \oplus c \qquad (1)$$

$$Carry = a.b + b.c + c.a \qquad (2)$$

In equations (1) and (2), $\oplus$ represents Ex-Or operation and represents AND operation.

Table 1: Truth Table of Full Adder

c	a	b	Sum	Carry
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

The Conventional 1- bit adder circuit is shown in Figure1. It requires 6 gates (4 logic gates and 2 Ex-Or gates) to implement 1- bit adder circuit.

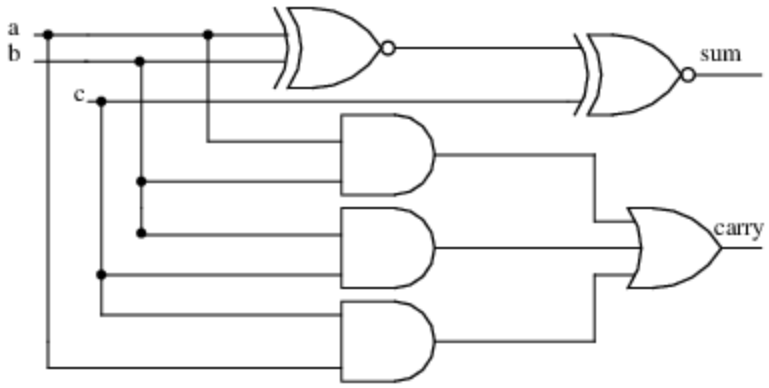

Figure 1: Conventional 1 – bit adder

2.1 Ripple Carry Adder

The addition of n- bit number can be done by connecting the n – bit full adders in cascade in which the Carry output from each full adder connected to the Carry input of the next full adder and it is shown in the Figure 2. The main drawback of this structure is that the delay increases with the number of bits.

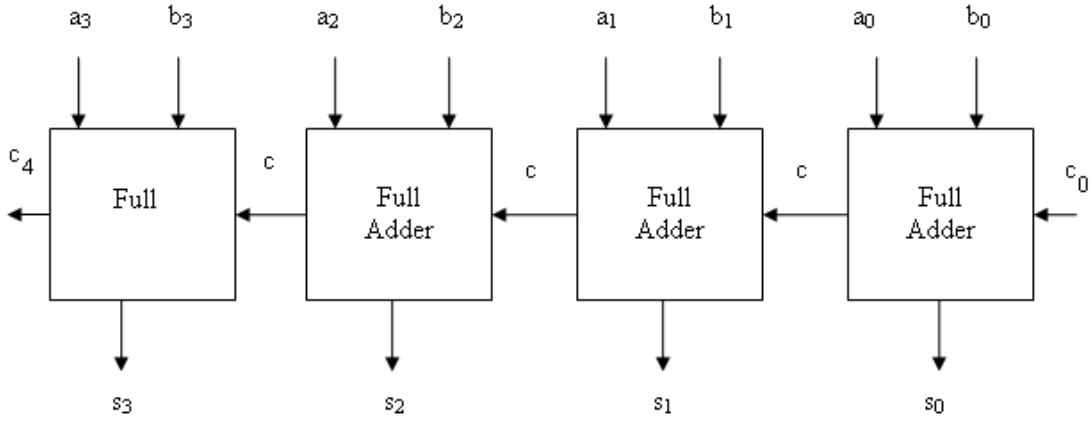

Figure 2: Ripple Carry Adder

2.2 Carry Look Ahead Adder

CLA reduces the Carry delay problem by calculating the Carry signals in advance, based on the input signals. The concept is based on the principle that a Carry signal will be generated in two cases. First, when both the bits a_i and b_i are '1', secondly, when one of the two bits is '1' and the Carry_in is '1'. The Sum and Carry equations are given by (3) and (4)

$$S_i = P_i \oplus c_i \qquad (3)$$

$$Carry = G_i + P_i.c_i \qquad (4)$$

P_i and G_i are called propagate term and generate term respectively, where $P_i = a_i \oplus b_i$ and $G_i = a_i \,.\, b_i$. Both the terms depend only on the input bits; therefore, it does not need to wait for the Carry to ripple through all the previous stages to find its proper value and is shown in the Figure 3. The only drawback of this method is that the carry logic block becomes more complicated for more than 4 bits.

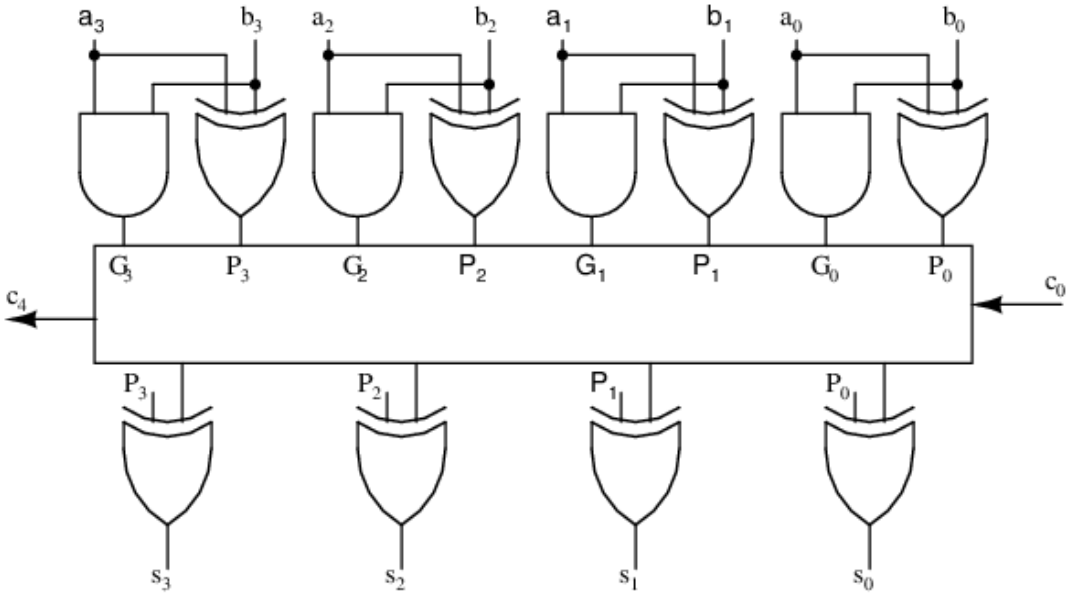

Figure 3: Carry Look ahead Adder

2.3 Carry Save Adder

The CSA consists of n full adders and every full adder computes a single Sum and Carry based on the corresponding bits of the input numbers. The entire Sum can be computed by shifting the Carry sequence left by one place and by appending a '0' to the front of the partial Sums and the result can be computed by adding these together with the use of RCA and shown in the Figure 4. It produces all of its output values in parallel and thus has the equal delay as a single full adder. The drawback of the method is that the intermediate values become unknown.

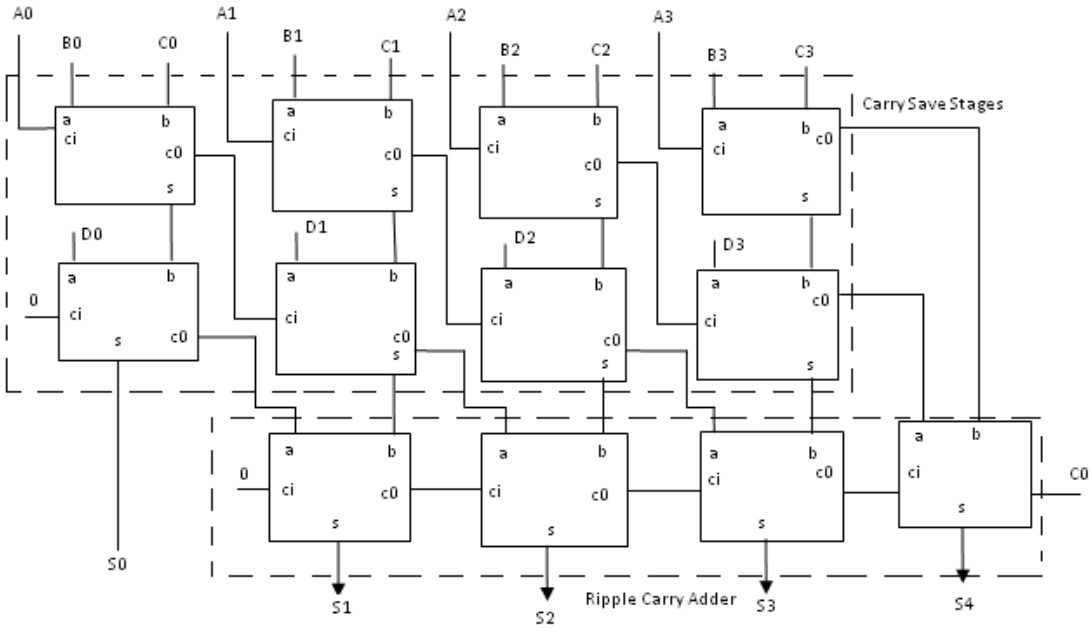

Figure 4: Carry Save Adder

2.4 Carry Select Adder

In this method, the Carry out of each block is pre-computed for both Carry_in = '0' and Carry_in = 1. The correct Carry is then selected by using multiplexer and this process can be done for all the blocks in parallel and is shown in the Figure 5. It reduces the computing delay with the increase in area.

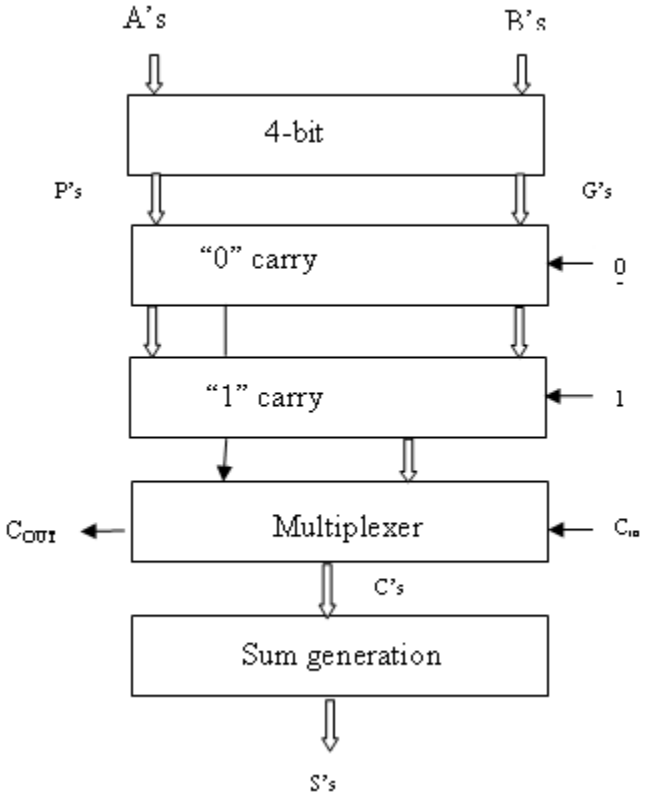

Figure 5: Carry Select Adder

2.5 Carry Bypass Adder

The design allows the Carry bit to skip over groups of n bits. It consists of the Carry skip circuit and hence the circuit calculates the Carry for the next stage rapidly and is shown in the Figure 6. So, the delay is reduced for completing the process.

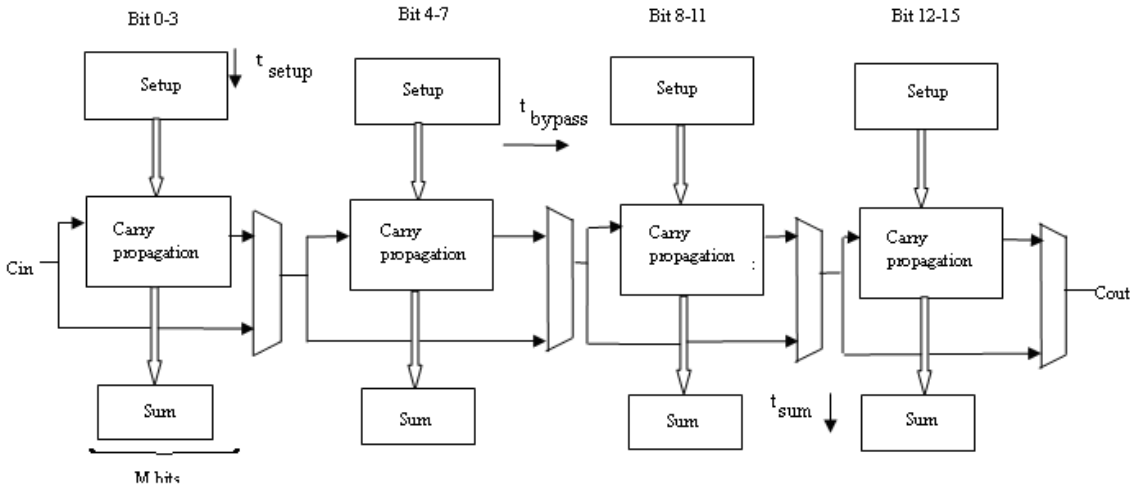

Figure 6: Carry By-pass Adder

2.6 Area Efficient Carry Select Adder

The operation of this adder [11] is based on the normal carry select adder. But the difference is that the one Ex-Or gate and one inverter are used to generate the Sum output and one OR gate and one AND gate are used to generate the carry. Finally, two multiplexers are used each for Sum and Carry to get the final output. That is, if $C_{in} = 0$, the Ex-Or gate output is selected for the Sum output by the corresponding multiplexer and the AND gate output is selected for the Carry output by the other multiplexer which is used for generating the Carry output. If $C_{in} = 1$, the output of the inverter is selected for the Sum and the output of OR gate is selected for the Carry.

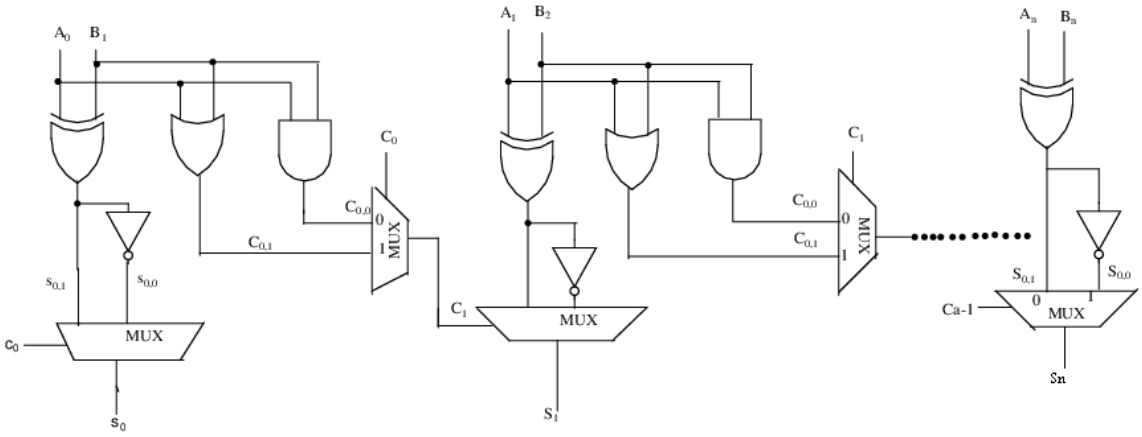

Figure 7: Area Efficient Carry Select Adder

3 Proposed Time Efficient Carry Select Adder

In TECSA, the truth table is mentioned in Table 1 viewed as two halves. First half has the inputs with the third input c is equal to '0' and the other half has the inputs with 'c' is equal to '1'.The interpretations can be made only with the two inputs a and b. In the first half, when c = '0', the sum can be determined by using Ex-Or gate and carry can be calculated using AND gate which is similar to the conventional adder. Whereas, in the second half, when c = '1', the Sum can be determined by inverting the Sum value which was obtained in the first half (Ex-Nor operation) and are given by

At c = '0', $S_{out} = a \oplus b$, sum = S_{out} and Carry = a.b

At c = '1', $\overline{Sum} = S_{out}$ and Carry = a.b

Instead of using one Ex-Nor gate, the output is obtained only by using one NOT gate. Because, the Ex-Or functionality between two inputs is already obtained in the first half section. So, the output is simply obtained by inverting the Ex-Or gate output. Therefore the propagation delay is not reduced, because of using single input NOT gate in the place of two inputs Ex-Nor gate.

In the case of carry calculation, the conventional adder needs four logic gates (3 AND gates and 1 OR gate), whereas, TECSA needs either one OR gate or one AND gate. That is, when c = '0', one 2-input AND gate and when c = '1', one 2- input OR gate are enough to obtain Carry. The operation is similar to the area efficient carry select adder but the difference is that, instead of using two multiplexers, only one multiplexer is used to get the desired output. Therefore, TECSA needs four gates (1 Ex-Or, 1 AND, 1 OR, and 1 NOT) and one 2-input Multiplexer to perform 1-bit addition and is shown in the Figure 8.

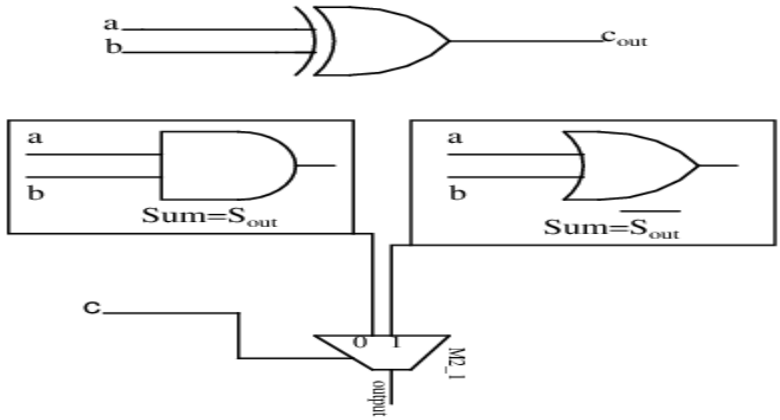

Figure 8: Proposed Time Efficient Carry Select Adder

4 Experimental Results

The number of gates required for conventional method and proposed adder are given in the Table 2. The time and area consumption based on the number of slices and numbers of Look Up Tables (LUTs) are given in Table 3, Table 4 and Table 5 respectively. The various types of existing adders and proposed TECSA are simulated and synthesized using Xilinx 14.1i with the FPGA device Spartan 3. The analysis has been made for various bit sizes such as 4-bits, 8-bits, 16-bits, 32-bits and 64-bits.

Table 2: Requirement of Gates

Bit Size	Conventional Adder					Time Efficient Carry Select Adder(TECSA)						
	XOR	AND	OR	NOT	Total	C	XOR	AND	OR	NOT	MUX	Total
1-Bit	2	3	1	-	6	C=0	1	1	-	-	4	6
						C=1	1	-	1	1		7
4-Bit	8	12	4	-	24	C=0	4	4	-	-	16	24
						C=1	4	-	4	4		28
8-Bit	16	24	8	-	48	C=0	8	8	-	-	32	48
						C=1	8	-	8	8		56
16-Bit	32	48	16	-	96	C=0	16	16	-	-	64	96
						C=1	16	-	16	16		112
32-Bit	64	96	32	-	192	C=0	32	32	-	-	128	192
						C=1	32	-	32	32		224

(1 two input multiplexer requires 4 gates and all the gates represented in the table are two input gates)

Table 3: Experimental Results based on the Time Consumption in Ns

BIT SIZE	RCA	CLA	CSA	CBY	CSEL	ASCSA	TECSA
4 bit	12.008	13.902	15.818	15.225	10.419	15.225	11.911
8 bit	17.585	20.394	15.455	21.717	17.339	21.717	17.044
16 bit	28.741	33.378	35.279	34.701	24.138	34.701	24.8
32 bit	50.967	59.346	61.383	60.669	28.274	60.669	45.335
64 bit	95.825	111.282	113.591	112.552	66.546	103.552	86.406

Table 4: Experimental Results based on the number of Slices

BIT SIZE	RCA	CLA	CSA	CBY	CSEL	ASCSA	TECSA
4 bit	4	4	11	5	5	4	4
8 bit	9	9	15	10	15	9	9
16 bit	18	18	45	19	32	13	19
32 bit	37	37	91	37	68	37	37
64 bit	74	74	183	74	139	74	74

Table 5: Experimental Results based on the number of LUTs

BIT SIZE	RCA	CLA	CSA	CBY	CSEL	ASCSA	TECSA
4 bit	8	8	19	9	9	8	8
8 bit	16	16	27	17	27	16	16
16 bit	32	32	79	33	59	23	33
32 bit	64	64	159	65	123	64	65
64 bit	128	128	319	129	251	128	129

5 Performance Analysis

Based on the results given in Table 3 and Table 4, the graphs have been drawn for various bit sizes and it is shown in Figure 9 and Figure 10 respectively. The performance of existing adder designs and TECSA design are compared on the basis of time and area consumption. When comparing all the designs, CSA requires less time when compared to the other adder structures including TECSA design but the area utilization is very high. The AECSA requires less area but the time consumption is high. The proposed TECSA requires less time as well as with the equal consumption of area which is required by other structures. Therefore, TECSA produces the optimized results which balance both time and area consumption leads to the better performance compared to the other designs.

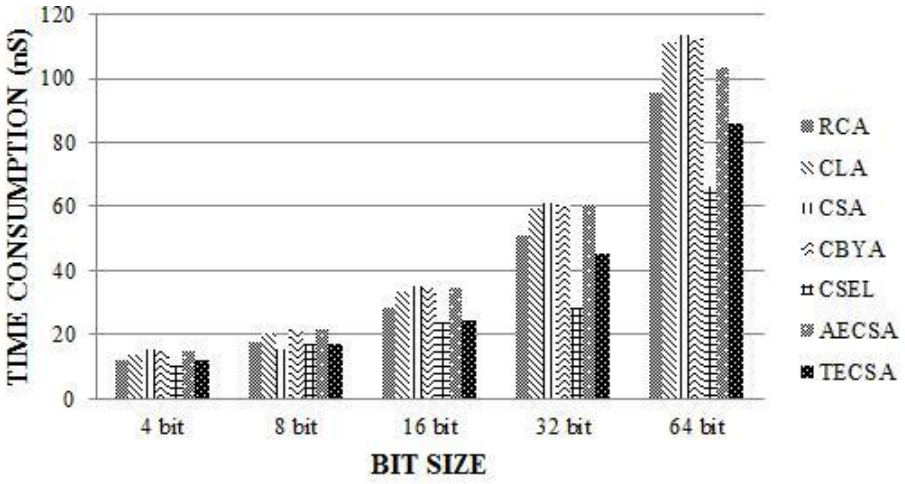

Figure 9: Comparison between various Adders and TECSA based on the Time Consumption

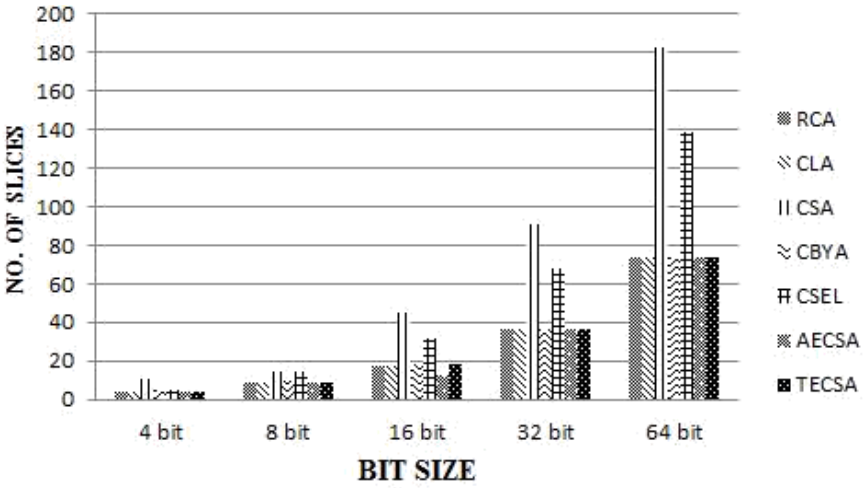

Figure 10: Comparison between various Adders and TECSA based on the number of slices

6 Conclusion

Due to the enormous impact of adder structure in arithmetic applications, a new TECSA has been proposed. The performance of TECSA was compared against conventional adder. TECSA yields the better results in time and area consumption. Hence, the faster arithmetic designs will be realized if TECSA is used for addition purpose.

References

[1] Daniela Elena Popescu, "The Performability analysis for different kinds of adders", University of Oradea, Romania.

[2] M. Hosseinghadiry, H. Mohammadi, M. Nadisenejani, "Two new low power high performance full adder with minimum gates", International Journal of Electrical and Computer Engineering, 2009.

[3] Chetana Nagendra, Mary Jane Irwin, Robert Michael Owens, "Area-Time-Power Trade - offs in Parallel Adders", IEEE Transactions on Circuits and Systems, Vol.43, No. 10,October 1996, pp. 689- 702.

[4] Reza Hashemian, "A New Design for High Speed and High – Density Carry Select Adders".

[5] Shanzhen Xing, William W.H. Yu, "FPGA Adders: Performance Evaluation and Optimal Design", IEEE Design and Test of Computers, 1998, pp. 24 – 29.

[6] Paolo Zilari, Stefania Perri, "A Fast Carry Chain Adder for Virtex – 5 FPGAs", IEEE Conference Proceedings, 2010, pp. 304 – 308.

[7] Manjula B.B, Venkatesh S. Sanganal, Hemalatha K.N, Ravichandra V, "FPGA Implementation of Optimized 4 – bit BCD and Carry skip adders using Reversible gates", International Journal of Soft Computing and Engineering, ISSN: 2231 – 2307, Vol.2, Issue - 3, July 2012.

[8] S.V. Padmajarani, Dr. M. Muralidhar,"A new approach to implement Parallel Prefix adders in an FPGA", International Journal of Engineering Research and applications, ISSN: 2248 – 9622, Vol.2, Issue 4, July 2012, pp. 1524 – 1528.

[9] Hadi Parandesh Afshar, Ajay Kumar Verma, Philip Brisk, Paolo Jenne, "Improving FPGA Performance for Carry – Save Arithmetic", IEEE Transactions on VLSI, Vol.18, No.4, April 2010, pp. 578 – 590.

[10] Taeko Matsunaga, Shini Kimura, Yusuke Matsunaga, " Multi – operand Adder Synthesis on FPGAs using Generalized Parallel Counter".

[11] I – Chyn Wey, Cheng – Chen Ho, Yi – Sheng Lin, Chien – Chang Peng, "An Area – Efficient Carry Select Adder Design by sharing the Common Boolean Logic Term", Proceedings of the International Multiconference of Engineers and Computer Scientists, Vol. II, 2012.