RESEARCH ARTICLE OPEN ACCESS

Design & Analysis of Low Power, Area-Efficient Carry Select Adder

Shuchi Verma, Sampath Kumar V.
JSS Academy of Technical Education, Noida – 201301, India

ABSTRACT
This paper deals with the design & analysis of Carry Select Adder (CSLA) & Carry lookahead adder (CLA). Adders are designed using 0.18µm CMOS process technology & simulated with Modelsim6.3f. The adder designs, Regular CSLA, modified CSLA using BEC, modified CSLA without using multiplexer, modified CSLA using D-Latch & Carry lookahead adders in 4-bit, 16-bit, 32-bit, are compared with the simulated results on the basis area.

***Keywords**:* Area, BEC, CSLA, D-Latch & power consumption.

I. INTRODUCTION

Carry Select Adder (CSLA) is one of the fastest adders used in much computational system to perform fast arithmetic functions [1]. On comparing CSLA with Ripple carry adder (RCA) & Carry lookahead adder (CLA), CSLA compromises between these two. If there is N-bit RCA, the delay is linearly proportional to N. Thus for large values of N the RCA gives highest delay of all adders. The Carry Look-Ahead Adder (CLA) gives fast results but consumes large area. If there is N-bit adder, CLA is fast for N≤4, but for large values of N its delay increases more than other adders. So for higher number of bits, CLA gives higher delay than other adders due to presence of large number of fan-in and a large number of logic gates. The Carry Select Adder (CSLA) provides a compromise between small area but longer delay RCA and a large area with shorter delay CLA [2]

In this research, 4-bit, 16-bit, 32-bit Regular CSLA, modified CSLA using BEC, modified CSLA without using mux, modified CSLA using D-Latch & Carry lookahead adder have been simulated. Comparison of various adders is done on the basis of area with help of simulated results.

This paper is summarized as follows:

Section II gives brief introduction of structure of Regular CSLA & modified CSLA. Section III gives the performance analysis of various adders & comparison of simulated results. Finally, the work is concluded in Section IV.

II. ADDER STRUCTURE

Regular Carry select adder is designed by using RCA & multiplexer. It is cascading of RCAs (RCA is cascading of full adders). One RCA with carry input $C_{in} = 0$, and other RCA with $C_{in} = 1$. Design of 16-bit Regular CSLA is done by using this technique [1]. It has five groups of different bit size RCA. The replicated input is given to both the RCAs to generate partial sum and carry, then the final sum and carry are selected by the multiplexers. Area count (gate count) is done by AND, OR, INVERTER (AOI) implementation [1].

Gate count = FA + HA + Mux

(Number of AND, OR, INVERTERs present in FA (full adder), HA (half adder) & multiplexer). Due to multiple pair of RCAs, the Regular CSLA consumes larger area.

In the modified CSLA, the RCA with $C_{in} = 1$, is replaced by Binary to excess one converter (BEC). To replace n-bit full adder, n+1 bit BEC is required. The main advantage of this BEC logic comes from the lesser number of logic gates than the n-bit Full Adder (FA) structure [1]. The partial sum & carry are generated by RCA & BEC, then the final sum & carry is selected by multiplexer. If we have 4-bit inputs $(B_0, B_{1,}B_2, B_3)$ & the outputs are $(X_0, X_{1,}X_2, X_3)$ then the Boolean expressions of the 4-bit BEC is listed as:

$$X_0 = \sim B_0$$
$$X_1 = B_0 \wedge B_1$$
$$X_2 = B_2 \wedge (B_0 \& B_1)$$
$$X_3 = B_3 \wedge (B_0 \& B_1 \& B_2)$$

(Functional symbols are ~NOT,& AND, ^XOR [1]).

Using this technique 16-bit modified CSLA using BEC is designed. It has five groups of different bit sizes. Area count is done by AOI implementation.

Gate count = FA + HA + Mux + BEC

This modified CSLA obtains reduced area with slight increase in delay [1].

In the next modification of Regular CSLA, the RCA with $C_{in} = 1$ & multiplexer, is replaced by combinational circuit, which consist of AND & XOR gate. Design of 16-bit modified CSLA without using mux is done [2]. It has five groups of different bit sizes. Initialy RCA structure calculates sum & carry for $C_{in} = 0$, the output of full adder is given to the combinational circuit, one of the input of combinational circuit is previous stage carry. Then the final sum & carry is selected by combinational circuit to generate proper output [2].

But the disadvantage of this modified CSLA without using mux is that the delay is increased then Regular CSLA & modified CSLA using BEC [2] .

In the final modification of Regular CSLA, the RCA with $C_{in} = 1$, is replaced by D-Latch. For n bit RCA structure it required n D-latches with enable pin as a clk. Design of 16-bit modified CSLA using D-Latch is done [3]. Latches are used to store one bit information. In one clock cycle each of the two addition is performed. Addition operation for carry input $C_{in} = 1$ is performed when clock pulse is high, and when the clock pulse is low, assuming carry input $C_{in} = 0$. D-Latches are enabled and store the sum and carry for carry is equal to one. According to the value of C_{in} whether it is 0 or 1, the multiplexer selected the actual sum and carry [3]. The modified CSLA using D-Latch has lowest delay as compared to regular & modified CSLAs.

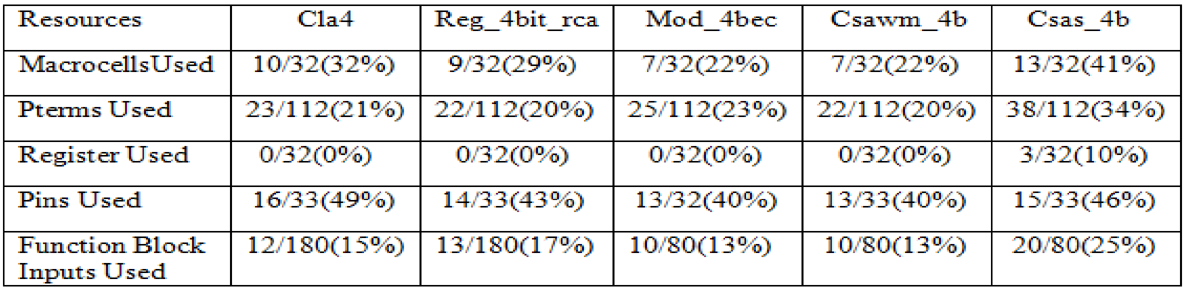
Carry lookahead adder (CLA) is also one of the fastest adder used in many data-processing processor. It takes lowest carry propogation delay, to generate final sum & carry output, but the disadvantage of CLA is this that it consumes larger area as compared to other adders.

III. PERFORMANCE ANALYSIS

Here we have simulated various 4-bit, 16-bit, 32-bit Regular CSLA, modified CSLAs & CLA. The Xilinx 9.1i software is used for synthesising the adders, & Modelsim6.3f is used to compile & simulate to verify the VHDL code. Table1, Table2, Table3 gives the component comparison of various 4-bit, 16-bit, 32-bit adders.

COMPARISON OF REGULAR AND MODIFIED CSLA:

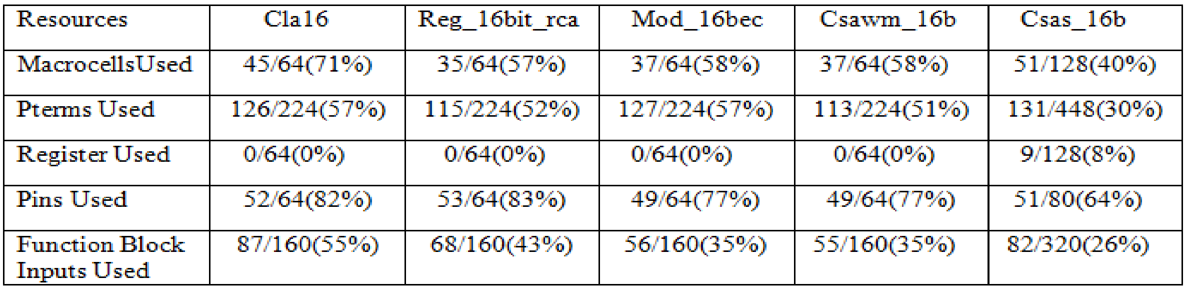
Table 1: Component comparison of various 4-bit adders



Resources	Cla4	Reg_4bit_rca	Mod_4bec	Csawm_4b	Csas_4b
MacrocellsUsed	10/32(32%)	9/32(29%)	7/32(22%)	7/32(22%)	13/32(41%)
Pterms Used	23/112(21%)	22/112(20%)	25/112(23%)	22/112(20%)	38/112(34%)
Register Used	0/32(0%)	0/32(0%)	0/32(0%)	0/32(0%)	3/32(10%)
Pins Used	16/33(49%)	14/33(43%)	13/32(40%)	13/33(40%)	15/33(46%)
Function Block Inputs Used	12/180(15%)	13/180(17%)	10/80(13%)	10/80(13%)	20/80(25%)

Cla4: 4-bit Carry lookahead adder, **Reg_4bit_rca:** Regular 4-bit Carry select adder, **Mod_4bec:** Modified 4-bit Carry select adder using BEC, **Csawm_4b:** Modified 4-bit Carry select adder without using mux, **Csas_4b:** Modified Carry select adder using D-Latch (it is also called Carry select adder with sharing).

Table 2: Component comparison of various 16-bit adders



Resources	Cla16	Reg_16bit_rca	Mod_16bec	Csawm_16b	Csas_16b
MacrocellsUsed	45/64(71%)	35/64(57%)	37/64(58%)	37/64(58%)	51/128(40%)
Pterms Used	126/224(57%)	115/224(52%)	127/224(57%)	113/224(51%)	131/448(30%)
Register Used	0/64(0%)	0/64(0%)	0/64(0%)	0/64(0%)	9/128(8%)
Pins Used	52/64(82%)	53/64(83%)	49/64(77%)	49/64(77%)	51/80(64%)
Function Block Inputs Used	87/160(55%)	68/160(43%)	56/160(35%)	55/160(35%)	82/320(26%)

Cla16: 16-bit Carry lookahead adder, **Reg_16bit_rca:** Regular 16-bit Carry select adder, **Mod_16bec:** Modified 16-bit Carry select adder using BEC, **Csawm_16b:** Modified 16-bit Carry select adder without using mux, **Csas_16b:** Modified 16-bit Carry select adder using D-Latch (it is also called Carry select adder with sharing).

Table 3: Component comparison of various 32-bit adders

Resources	Cla32	Reg_32bit_rca	Mod_32bec	Csawm_32b	Csas_32b
MacrocellsUsed	218/384(57%)	76/256(30%)	87/128(68%)	85/128(67%)	103/256(41%)
Pterms Used	425/1344(32%)	244/896(28%)	240/448(54%)	238/448(54%)	260/896(30%)
Register Used	0/384(0%)	0/256(0%)	0/128(0%)	0/128(0%)	18/256(8%)
Pins Used	100/118(85%)	103/118(88%)	97/0(1%)	97/0(1%)	99/118(84%)
Function Block Inputs Used	549/964(58%)	134/640(21%)	143/320(45%)	131/320(41%)	171/640(27%)

Cla32: 32-bit Carry lookahead adder, **Reg_32bit_rca:** Regular 32-bit Carry select adder, **Mod_32bec:** Modified 32-bit Carry select adder using BEC, **Csawm_32b:** Modified 32-bit Carry select adder without using mux, **Csas_32b:** Modified 32-bit Carry select adder using D-Latch (it is also called Carry select adder with sharing).

Area is estimated by considering percentage use of macrocells. Percentage area reduction of various adders is plotted in Fig.1.

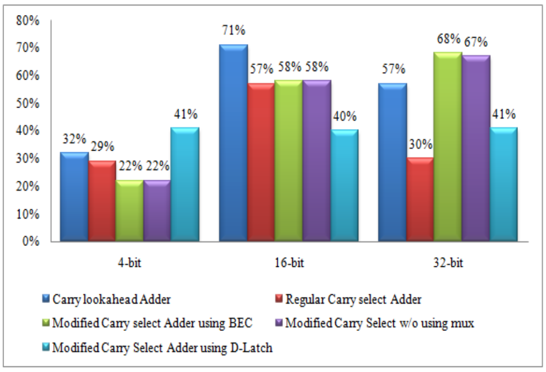


Fig.1 Percentage area reduction of various 4-bit, 16-bit, 32-bit adders.

From the Fig.1, & Table 1, Table 2, Table 3 it can be concluded that Regular CSLA & Modified Carry select adder using D-Latch, achieve lower area than Carry lookahead adder. Modified CSLA using BEC & modified CSLA without using multiplexer obtain almost equal area.

IV. CONCLUSION

In this paper, simulation results of various Regular Carry Select Adders, modified CSLA using BEC, modified CSLA without using mux, modified CSLA using D-Latch & Carry Look Ahead Adder, in 4-bit, 16-bit, 32-bits is estimated. The Xilinx 9.1i software is used for synthesising the project, & Modelsim6.3f is used to compile & simulate to verify the VHDL code. Device family, Spartan 3A, into which the adder design is implemented, is used. XC3S50A device is used to implement the design.

REFERENCES

[1]. B. Ramkumar and Harish M. Kittur "Low-Power and Area-Efficient Carry Select Adder", IEEE transactions on very large Scale integration (VLSI) systems, vol. 20, pp.371-375, February 2012.

[2]. Sarabdeep Singh and Dilip Kumar "Design of Area and Power Efficient Modified Carry Select Adder", International Journal of Computer Applications, vol.33, pp. 14-18, November 2011.

[3]. Laxman Shanigarapu, Bhavana P.Shrivastava "Low-Power and High speed Carry Select Adder", International Journal of Scientific and Research Publication, vol.3, pp. 1-9, August 2013.

[4]. O. Bedrij, "Carry Select Adder," IRE Trans. on Electronic Computers, vol. EC-11, pp. 340- 346, 1962.

[5]. Mr. C. S. Manikandababu " An Efficient Carry Select Adder architecture for VLSI hardware implementation", IJMIE, vol.2, pp.610-622, May 2012.

[6]. T. Y. Ceiang and M. J. Hsiao, "Carry-select adder using single ripple carry adder", vol. 34, No. 22, pp. 2101–2103, Oct. 1998.

[7]. He, Y. Chang, C. H. and Gu, J. "An Area Efficient 64-Bit Square Root Carry-Select Adder for Low Power Applications", in Proc. IEEE Int. Symp. Circuits Syst., vol.4, pp. 4082–4085, 2005.