FPGA Implementation of High Speed Hardware Efficient Carry Select Adder

Saravanakumar[1], Vijeyakumar[2], Sakthisudhan[3]
[1]Anna University, Department of ECE, Bannari Amman Institute of Technology, Tamil Nadu, India
[2]Anna University, Dr.Mahalingam College of Engg & Technology, Tamil Nadu, India
[3]Anna University, Adhi College of Engg & Technology, Tamil Nadu, India

Article Info	ABSTRACT
	This paper presents a novel architecture for high speed and hardware efficient carry select addition. We modify the two operand ripple carry addition followed in conventional Carry SeLect Adder (CSLA) with a simple and efficient gate level circuit to reduce area and delay significantly. For this, we use an increment 1 block for generating the sum outputs with carry input 1 instead of second pair ripple carry adder as in conventional CSLA. The novelty of the proposed approach is that it reduces area, and the delay due to carry propagation in second pair of adder cells. The proposed CSLA adder has been designed using structural VHDL code and synthesized using Altera Quartus II. Experimental results show that the proposed design outperform the previous approaches in terms of delay and area reduction.

Corresponding Author:

Saravanakumar,
Anna University, Department of ECE,
Bannari Amman Institute of Technology, TamilNadu, India
Email: sarapalani81@gmail.com

1. INTRODUCTION AND RELATED WORK

The basic arithmetic operation most frequently encountered in the design of digital processors is addition and design of high speed and area efficient adders is significant area of research in VLSI data path systems. A large number of adders have been designed to meet the requirements of different image and signal processing applications. Adders lie in the critical path of a processing architecture and the critical path determine the overall performance of the system. A variety of applications require certain arithmetic operations such as incrementing the sum of two numbers by unity, finding the absolute difference between two numbers, or augmenting the sum of two numbers by a constant. One approach to perform these operations is to utilize dual adders or use multi-operand adders such as the Carry-Save Adders (CSA), Carry SKip Adder (CSKA), Carry Propagate Adder (CPA) and CSLA. Also, multi operand addition forms a significant part of multiplication and certain DSP algorithms. Adder performance in a multi bit addition can be improved by reducing the delay due to carry propagation between different adder cells. This can be addressed by improving the structure of the basic adder block. A number of architectures and algorithms to improve efficiency in multi bit addition are proposed.

Kantabutra [1] proposed a novel approach for design of optimum-speed one-level CSKA. Min Cha and Swartzlander [2] proposed a modified version of carry skip logic in which to reduce delay in the first block, carry look ahead logic is used. However, the design is prone to high hardware complexity due to use of carry look ahead adder in the first stage. Youngjoon Kim and Lee-Sup Kim [3] proposed a CSLA using Ripple Carry Adder (RCA) and an add-one circuit instead of using dual RCAs. The proposed add-one circuit uses first zero finding circuit and multiplexers to reduce the area and power with no speed penalty. For bit length $n=64$, this new CSLA requires 38 percent fewer transistors than the dual ripple-carry CSLA.

Datta, et al. [4] proposed a novel approach for the design of 4-to-2 CSA using dynamic logic and the Limited Switch Dynamic Logic (LSDL) circuit. Yu Shen Lin and Radhakrishnan [5] proposed a novel design for 32 bit CSK addition. The Generate and Propagate logic used in carry look ahead addition are used to reduce delay in RCA. The number of bits in the skip blocks is decided by considering the critical path into account. The CSK adder in [5] is implemented in 25 ηm CMOS technology file. The experimental results demonstrated a critical path delay reduction of 18% compared to the best of the previous approaches. A novel technique for multi bit addition using flag bit generation was proposed by Vibhuti Dave et al. [6].

Yu pang et al. [7] proposed a novel carry skip design for reducing delay in addition due carry propagation. Yu pang design revealed better reduction in energy dissipation due to use of reversible logic. A 4 bit CSK adder was designed using 4*4 reversible TSG & Fredkin gate by Chiwande and Dakhole [8]. The proposed adder in [8] demonstrated better reduction in power dissipation compared to existing four bit CSKAs.

CSLA is preferred in Digital signal processors and application specific ICs designed to execute dedicated algorithms such as convolution, correlation and filtering to alleviate the problem of carry propagation delay in addition [9]. However, the hardware complexity of CSLA is high due to use of pair of RCAs to generate partial sum and carry corresponding to carry input 1 and 0. Then the final sum and carry are selected from the partial results by using multiplexers (mux) [10]. However, Ramkumar and Kittur [11] in a novel approach proposed a CSLA using Binary to Excess-1 Converter (BEC) instead of RCA for second stage addition to alleviate the problem of high hardware complexity. The proposed adder in [11] revealed lower area and reduced power dissipation due to minimal switching. To further reduce area and latency in CSLA addition we have proposed an increment 1 block for second stage addition. The novelty of the increment 1 block is that it reduces area and delay significantly.

The rest of the paper is organized as follows. Section 2 gives an overview of carry select addition. Section 3 discusses about the design of proposed hardware efficient CSLA. In section 4 the performance of the proposed design are discussed and compared with the previous approaches. Section 5 gives brief conclusion of the work done.

2. OVERVIEW OF CARRY SELECT ADDITION

Based on the carry selection addition approach, for the case of 16 bit input, the input bits (A_{15-0} & B_{15-0}) can be grouped into four pairs of 4 bits(A_{15-12}, A_{11-8}, A_{7-4}, A_{3-0} and B_{15-12}, B_{11-8}, B_{7-4}, B_{3-0}). The corresponding most significant and least significant digits(grouped bits) can be added in parallel using separate 4 bit RCA with initial carry input as zero for first stage and 1 for second stage. The sum outputs from pair of adder cells at the corresponding bit position are adder by one. The various blocks of the proposed hardware efficient CSLA are first stage RCA, Increment 1 block and 2:1 passed through 2:1 mux, with the select signal for the mux being the carry out of the most significant adder in previous group. Thus, the total delay for a 16 bit addition in CSLA is 4 carry and 4 mux delay with the expense of little increase in hardware. However, for a 16 bit addition conventional RCA require 16 carry propagation delay to realise the final sum. The schematic of the basic CSLA for *n*=16 is shown in Figure 1.
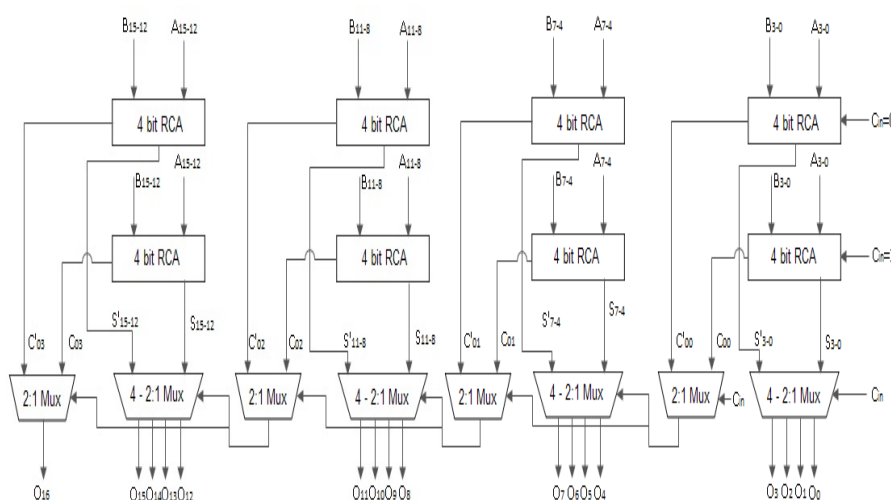

Figure 1. Schematic of conventional CSLA

3.   PROPOSED HARDWARE EFFICIENT CSLA

The proposed high speed hardware efficient CSLA uses a simple gate level increment 1 block for incrementing the outputs of first stage multiplexers as shown in Figure 2. For a 16 bit input A_{15-0} and B_{15-0}, we divide bits as $A_{15-12}, A_{11-8}, A_{7-4}, A_{3-0}$ and $B_{15-12}, B_{11-8}, B_{7-4}, B_{3-0}$ and are fed to the first stage adder in parallel. The sum outputs $S'(S'_{15}S'_{14}S'_{13}S'_{12})$ $(S'_{11}S'_{10}S'_{9}S'_{8})$ $(S'_{7}S'_{6}S'_{5}S'_{4})$ & $(S'_{3}S'_{2}S'_{1}S'_{0})$ of the first stage adder are fed to Increment 1 block to produce the equivalent sum output $S(S_{15}S_{14}S_{13}S_{12})$ $(S_{11}S_{10}S_{9}S_{8})$ $(S_{7}S_{6}S_{5}S_{4})$ & $(S_{3}S_{2}S_{1}S_{0})$ and a carry output C_{out}, corresponding to the carry input 1. The first stage RCA output(S') and Increment 1 block output(S) are fed to the multiplexer with the control signal for the multiplexer being the carry output of previous group of adder cells . The schematic of the increment 1 block is shown in Figure 3.The logic that defines the output of Increment 1 block are shown in Equation (1) to Equation(5). We use 'i' to represent the position of the adder block, with i=0 being the least adder block. (Note that symbol Θ represents XOR logic, ~ represents NOT logic, ^ represents AND logic).
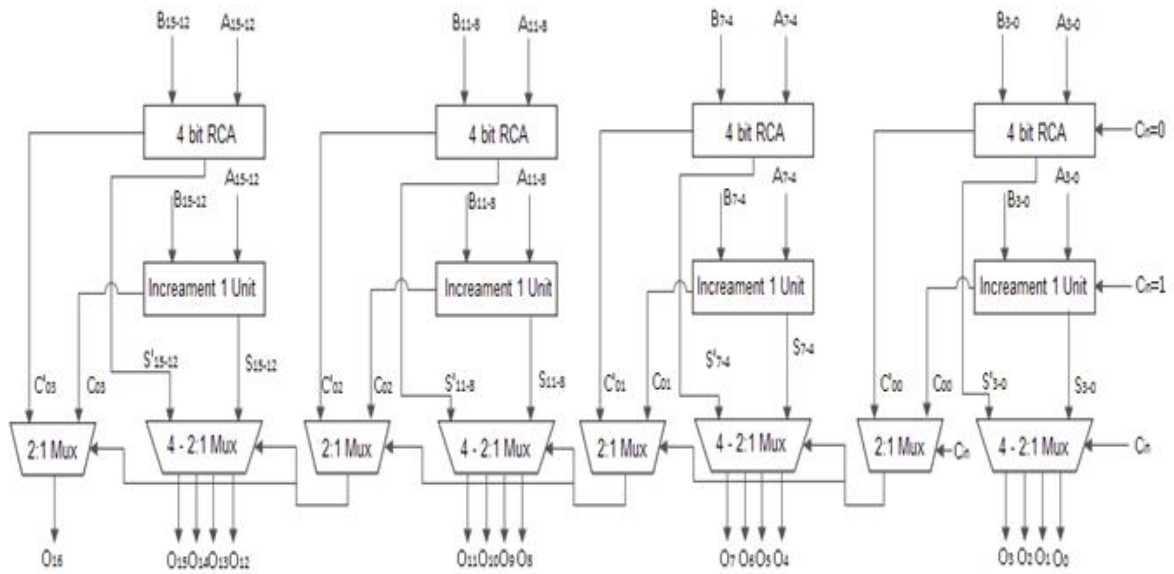

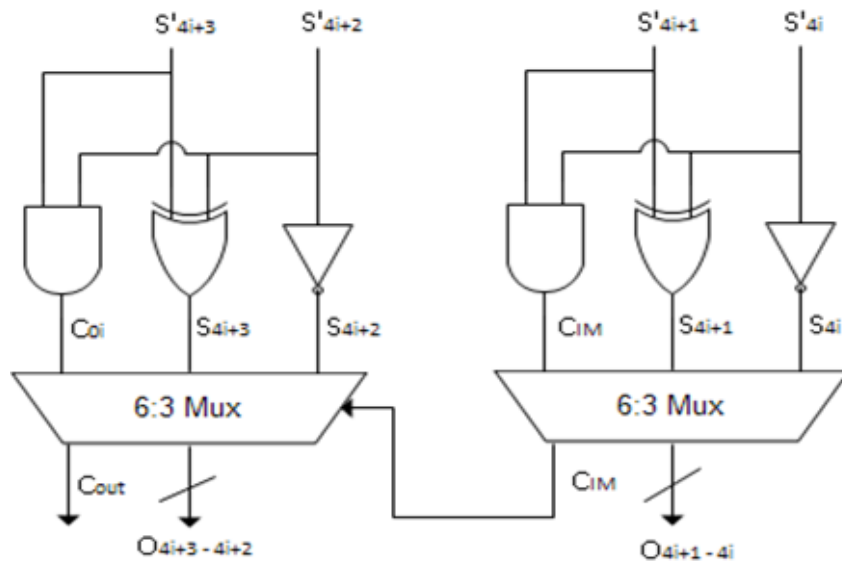Figure 2. Schematic of proposed hardware efficient CSLA for 16 bit addition

Figure 3. Schematic of increment 1 unit

$$S4i = \sim S'4i \tag{1}$$

$$S4i + 1 = S'4i \ominus S'4i + 1 \tag{2}$$

$$S4i + 2 = \sim S'4i + 2 \tag{3}$$

$$S4i + 3 = S'4i + 2 \ominus S'4i + 3 \tag{4}$$

$$Cout = S'4i + 2 \wedge S'4i + 3 \tag{5}$$

4. AREA AND DELAY EVALUATION METHODOLOGY OF PROPOSED CSLA

To evaluate the area and delay of the proposed hardware efficient CSLA, we used BEC-CSLA [11] and Conventional designs [10] for comparison. The area and delay evaluation of the proposed and conventional designs are done based on the NAND equivalent implementation of the basic elements that make up the proposed and conventional designs. The NAND implementation of basic elements viz., Full Adder (FA), NOT, AND and OR gates used in the proposed and conventional BCD adder designs are shown in Figure 4. To calculate the area count and critical delay of the proposed CSLA and previous CSLA designs we assume the delay of NAND gate to be 1 unit and area equal to 1 count. The gates in parallel between dotted lines perform parallel operation and we use only one gate delay in case of parallel gates for calculation of worst case delay of circuit/element. Based on the above approach, the worst case delay is found by counting the number of NAND gates in the critical path and the area is evaluated by counting the total number of NAND gates that make up the circuit. The area and delay values of the proposed CSLA and previous approaches are shown in Table 1. From the reports in Table 1, it is seen that the proposed CSLA design demonstrates significant reduction in area and delay.

Table 1. Area Count (in NAND equivalent) and Delay of the Proposed CSLA and Previous Approaches

Basic Designs	Parameter	1st stage adder	2nd stage adder/Add 1 circuit	2:1 mux	Total
Conventional	Area	48	48	20	116
	Delay	12	12	3	27
CSLA-BEC	Area	48	19	20	87
	Delay	12	7	3	22
Proposed	Area	48	14	20	82
	Delay	12	3	6	21

5. RESULTS AND DISCUSSION

The proposed hardware efficient CSLA and designs used for comparison are described using structural VHDL to produce gate level net list and synthesized using Altera Quartus II with EP2C35F672C6 device. The area, delay and total power dissipation results of proposed CSLA and previous approaches are shown in Table 2. From the reports, it is seen that the proposed CSLA design has lower logic cell count compared to all other architectures used for comparison, thanks to the increment 1 block which realizes constant (0001) addition with fewer gates. The delay of the proposed increment 1 -CSLA design is less when compared with all other adder designs used for comparison. This is due to the elimination of delay due to carry propagation in second pair of adders with carry input 1. However, the proposed CSLA exhibit little higher total power dissipation compared to the conventional design. The better delay performance of the proposed CSLA reveals the best Area – Delay Product (ADP) and Power-Delay Product(PDP) performances compared to the previous approaches.

Table 2. Comparison of Area, Delay and Power Dissipation of Proposed CSLA and state-of the Art Designs for n=16

Name of the Design	Area (Number of Logic Elements)	Delay (ns)	Total power dissipation (mW)	ADP	PDP (mW-ns)
Conventional CSLA	51	18.153	176.74	925.8	3208.36
CSLA with BEC	44	17.019	177.50	748.84	3020.87
Proposed CSLA	42	16.816	177.36	706.27	2982.485

5. CONCLUSION

A novel approach for the design of carry select adder which reduces area and delay significantly is proposed in this brief. Extensive comparison using synthesis results shows that the proposed CSLA outperformed all other previous designs in terms of delay and area reduction. The potential benefits of reduced logic cell count and delay of the proposed carry select adder realizes an ADP and PDP reduction of 23.7% and 7.4% respectively compared to the conventional design and 5.6% and 1.2% compared to BEC-CSLA. The proposed CSLA is thus hardware and energy efficient, and suitable for portable VLSI implementation.

REFERENCES

[1] Kantabutra, V. "Designing optimum carry-skip adders" *10th IEEE Symposium on Computer Arithmetic*, 1991, Grenoble.
[2] Min Cha and Swartzlander, E.E. "Modified carry skip adder for reducing first block delay" *Proceedings of the 43rd IEEE Midwest Symposium on Circuits and Systems*, 2000.
[3] Youngjoon Kim and Lee-Sup Kim "A low power carry select adder with reduced area" *The 2001 IEEE International Symposium on Circuits and Systems*, 2001.ISCAS 2001, Sydney, NSW.
[4] Datta, R. Abraham, J.A. ; Montoye, R. ; Belluomini, W. ; Hung Ngo ; McDowell, C. ; Kuang, J.B. ; Nowka, K." A low latency and low power dynamic Carry Save Adder" *Proceedings of the 2004 International Symposium on Circuits and Systems,* 2004.ISCAS '04.
[5] Yu Shen Lin and Radhakrishnan, D. "Delay Efficient 32-bit Carry-Skip Adder" *13th IEEE International Conference on Electronics, Circuits and Systems*, 2006, Nice
[6] Vibhuti Dave, Erdal Oruklu and Jafar Saniie,"Constant addition with flagged binary adder arhitectures", *Integration the VLSI journal* Vol.43 pp. 258–267 2010.
[7] Yu Pang, Junchao Wang; Shaoquan Wang " A 16-bit carry skip adder designed by reversible logic" *5th IEEE International Conference on Biomedical Engineering and Informatics (BMEI)*, 2012, Chongqing.
[8] Chiwande, S.S. and Dakhole, P.K. "VLSI design of power efficient Carry Skip Adder using TSG & Fredkin reversible gate" *IEEE International Conference on Devices, Circuits and Systems (ICDCS)*, 2012, Coimbatore.
[9] Vijeyakumar,K. N; Sumathy,V; Nithya, M ; Venkatanarayanan, C; Thiruchitrabala,V. "Design of low power full adder using active level driving circuit" *WSEAS transactions on circuits and systems*,Vol.11, No.8, 2012.
[10] Moris Mano, M and Michael D.Ciletti "Digital Design" Pearson Education, 2009.
[11] 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, No. 2, February 2012