DMPedia Lecture Notes in Computer Science & Engineering
Volume 1, pp. 613-622, 2026

International Conference on Multidisciplinary Perspectives in Advanced Computing and Technology (IMPACT 2026)
G. B. Pant University of Agriculture and Technology, Uttarakhand, India. Jan. 10-11, 2026

Low Power Carry Select Adder Design in VLSI Systems

Singala Sowmya, Ch. Pallavi

Dept. of ECE, Sri Venkateswara College of Engineering, Karakambadi Road, Tirupati, India

sowmyasingala08@gmail.com, pallavi.ch@svcolleges.edu.in

Abstract:
A digital circuit that performs addition is known as an electronic adder. Various numerical representations, including arithmetic and logical operations, can be used to create adders. Most adders work with binary numbers. The carry choose adder had the highest speed among the several adder types evaluated in this study. By employing the principles of the arithmetic logic unit (ALU), gate-level optimisation reduces the power consumption and area requirements of the carry selection adder. This study employed many methodologies, including the binary-to-excess converter (BEC) and Common Boolean Logic (CBL), to build the Carry Select Adder (CSLA). The comparative analysis indicates that the proposed ALU-based design consumes less power and has a smaller footprint than the conventional CSLA.

Keywords: *Common Boolean Logic (CBL), Arithmetic Logic Unit (ALU), adder, carry-select adder (CSLA).*

1. Introduction

When designing extremely large-scale integrated (VLSI) circuits, power consumption is a key factor. The rapid advancement of VLSI technology has compelled designers to prioritise minimising silicon area to meet the increasing demand for portable devices. The primary electronic component utilised in this addition was an adder. Adders are essential components of several digital systems. Nonetheless, power consumption remains a concern across numerous adder designs. Various varieties of adders are available, including Ripple Carry Adder (RCA), Carry Skip Adder (CSKA), Carry Look Ahead Adder (CLA), and Carry Save Adder (CSLA) and more. Among these, RCA has a compact design but is the slowest owing to its large propagation delay, even though it uses less area. On the other hand, CLA provides faster results but creates a larger delay owing to the ripple-carry mechanism. Power consumption is a critical consideration in the design of very large-scale integrated (VLSI) circuits. The rapid advancement of VLSI technology has compelled designers to prioritise minimising silicon area to meet the increasing demand for portable devices. The primary electronic component utilised in this addition was an adder. Adders are essential components of several digital systems. Nonetheless, power consumption remains a concern across numerous adder designs. Various adder types are available, including Ripple-Carry Adder (RCA), Carry-Skip Adder (CSA), Carry-Look-Ahead Adder (CLA), and Carry-Save Adder (CSA).

II. Carry Select Adder

A. Carry Select Adder (CSLA)

Two ripple carry adders are used to double-check each other when adding two n-bit values in

a carry-select adder. One adder assumes a carry-in of zero and the other a carry-in of one. The multiplexer is used to find the right sum and carry once both computations have yielded their findings. See below for a picture of a 16-bit Carry Select Adder's (CSLA) layout.

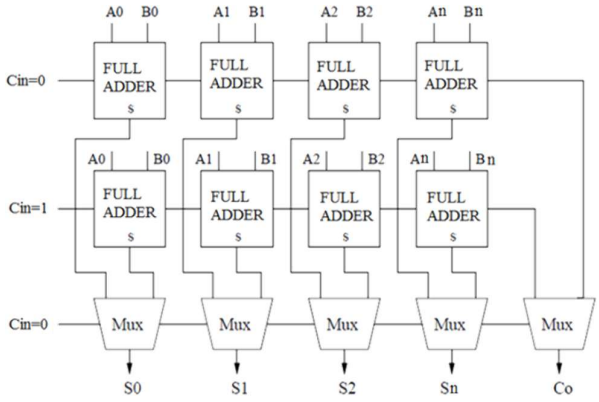

Fig.1: CARRY SELECT ADDER

The gate count of 4-bit CSLA is,

Gate Count = 136(FA+MUX)

Full adder = 104 (13*8)

MUX = 32 (4*8)

Select or press here to input text. To sum two n-bit values using a carry-select adder, two ripple carry adders were used. Two iterations of the calculation are performed: one with the carry-in assumed to be zero, and another with it assumed to be one. Upon computation of both results, the multiplexer selects the appropriate sum and carry based on the calculated carry value. There could be a fixed or variable number of bits in each carry-select block. The time it takes for the aggregate of inputs A and B to reach the carry-out should match the time it takes for the multiplexer chain to feed into the block, even if the block size changes. That way, you could be sure that the takeaway was actually there when you needed it. In a uniform block sizing implementation, the delay is often modulated by the square root of the sum of the bits being added; this is the optimal number of full-adder components for each block. By balancing the number of multiplexer delays with the number of full-adder delays, this configuration guarantees balanced performance.

B. BEC-BASED CSLA

In this Carry Select Adder (CSLA), the Ripple Carry Adder (RCA) that presumes a carry-in value of '1' is substituted with a Binary To Excess-1 Converter (BEC). In this configuration, each full adder cell in the RCA awaits the carry-in signal prior to producing the carry-out signal. To attain linear dependency, both potential carry-in values were considered, and the outcomes were pre-assessed for each scenario. Upon ascertaining the accurate carry-in value, the appropriate outcome was chosen with a multiplexer.

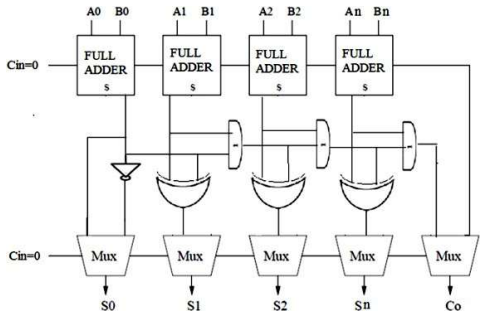

Fig. 2: Binary-to-excess-1 converter

The gate count = 92(FA+MUX+NOT+OR)

FA = 52 (13*4)

MUX = 32 (8*4)

NOT = 4 (4*1)

OR = 4 (1*4)

The conventional CSLA's space and power usage were decreased by employing this approach as opposed to an RCA with a carry-in ratio of 1. It takes a 4-bit BEC to swap out a 3-bit RCA. The diagram depicts the 4-bit BEC's component table, which details the parts and how they work. A 4-bit BEC and a multiplexer are used to accomplish the CSLA's basic operation. One input of the 8:4 multiplexer accepts (B3, B2, B1, and B0) as its input, while the other input is derived from the BEC output. This generates two potential partial outcomes concurrently, with the multiplexer employed to choose either the BEC output or the direct inputs according to the control signal Cin. The significance of BEC logic lies in its capacity to markedly decrease the silicon area in the construction of CSLAs with a substantial bit count.

C.CBL-BASED CSLA

The Mathematical Analysis of Logic, George Boole's debut work, published in 1847, introduced Boolean logic. An Investigation of the Laws of Thought (1854), his second work, provided further detail on the subject. According to Huntington, Sheffer first suggested the

term "Boolean Logic" in 1913. Nowadays, every computer language has Boolean logic, which is crucial for digital electronics development.

Boolean logic is a subset of algebra in mathematics and logic where variables in the truth table can only take on the values true or false, usually denoted as 1 and 0, respectively. In contrast to elementary algebra, which utilises numerical values for variables and emphasises addition and multiplication, Boolean logic primarily involves the operations of conjunction (AND), represented by ∧, disjunction (OR), represented by ∨, and negation (NOT), represented by ¬. Consequently, it serves as a formalism to delineate logical relations analogous to how conventional logic articulates numeric relations.

One INVERTER gate and one XOR gate can be used to form a summation signal pair, which can then be used to construct a shared Boolean logic term. A carry propagation channel with one AND gate and one OR gate pre-built to handle likely carry input values allows for logic-based selection of an appropriate summation output upon carry-in signal readiness.

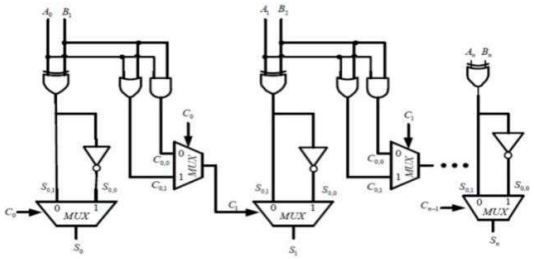

Fig. 3: COMMON BOOLEAN LOGIC

The gate count of 4-bit CBL is,

GATECOUNT = 88(FA+MUX+AND+NOT+EXOR)

FA = 52 (13*4)

MUX = 20 (5*4)

AND = 3 (3*1)

NOT = 1 (1*1)

EXOR = 12 (3*4)

After the carry-in signal is ready, it is feasible to select the correct carry-out by considering its logical condition. This solution guarantees that the summation generation circuit, which employs

XOR and INVERTER gates, and the carry-out generation circuit, which employs OR and AND gates, may work together.

III. **Proposed CSLA**

The Carry Select Adder (CSLA) is included into an Arithmetic and Logic Unit (ALU) in the suggested adder. ALU was first proposed by John von Neumann. Arithmetic and logic operations like addition and subtraction are carried out using an ALU, a digital circuit. One essential component of the CPU is the ALU. An ALU is a combinational circuit used in digital electronics that uses integer binary numbers to carry out bitwise logical and arithmetic operations. Numerous actions carried out on sets of bits can be decomposed into repetitive operations on single bits.

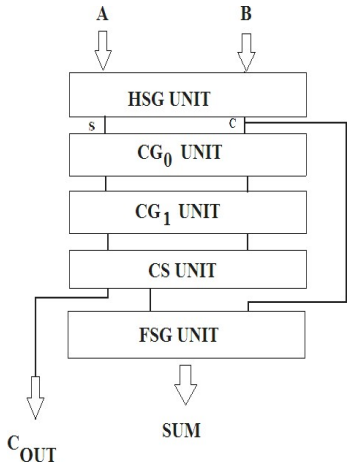

FIG. 4: ARITHMETIC AND LOGIC UNIT

The gates that are employed in the suggested CSLA include

- The Half Sum Generator (HSG) is implemented using the AND and XOR gates.
- The Carry Generator (CG) and Carry Selector (CS) both use the OR and AND gates.
- The Full Sum Generator (FSG) is the XOR gate.

Part of the ALU in this architecture are the Half Sum Generator (HSG), Carry Generator for Cin=0 (CG0), Carry Generator for Cin=1 (CG1), Carry Selector (CS), and Full Sum Generator (FSG).

The total number of gates in a 4-bit ALU is calculated as follows:

GATECOUNT = 60(EXOR+AND+OR)

EXOR = 8 (8*2)

AND = 16 (16*1)

OR = 12 (12*1)

A Half Adder (HSG) that produces Sum (S0) and Carry (C0) makes up the suggested CSLA. After that, the carrying generators receive the C0 and S0 values. The carry-in is assumed to be "0" (CG0) by one generator and to be "1" (CG1) by the other. The carry-in signal is used by the Carry Selector (CS) to choose the carry. Ultimately, the carry-out (Cout), half adder sum (S0), and half adder's sum (S0) were used to form the Full Sum (FSG).

ALGORITHM TO DESIGN ALU

1.The first step is to figure out what the ALU will do.

2. Start by building a 1-bit ALU, applying the "divide and conquer" principle.

3. There are n units that can be used to split an n-bit ALU. A name for this approach is the "bit sliced method."

4. This method of bit-slicing involves separating an ALU into logic and arithmetic units.

IV. **Simulation Tool**

A simulation program called Microwind is used to build integrated circuits at the level of physical description. The IC was built using a stick diagram in microwind circumstances. A stick diagram is a graphic depiction that uses colour coding to communicate layer information. This gave the circuit partner a top view. Tracing the signals' passage is made easier by the colour code.

The Microwind and DSCH were used to simulate different adder architectures. The complete adder module utilised in this investigation underwent timing studies and functional testing. Microwind

was created especially for submicron CMOS VLSI design training. It comes with a fast online analogue simulator, an electrical circuit simulator, and a layout editor. With lambda adjusted to 0.06 μ (60 nm), the technology library utilised in this study is based on 7-metal-layer 0.12 μm CMOS technology.

Microwind provides two environments: a simulator and a logic editor. The logic design simulations, which included delay analysis and physical circuit extraction, were validated using these environments, specifically DSCH and Microwind. A schematic entry was used to simulate each of the adders employed in this investigation. The functioning of the corresponding test patterns was confirmed after they were created. The schematic file was transformed into a Verilog file following verification. The Verilog file was then imported into the Microwind environment using the command "COMPILE VERILOG FILE," which turned the logic design schematic into a physical layout.

Layout design rules are a set of geometric limitations or guidelines that must be followed by the physical mask layout of any circuit that is to be fabricated using a certain process.

For each circuit made using a specific process, the primary goal of design guidelines is to use the least amount of silicon while achieving a high yield and reliability. The layout window was scaled according to lambda and presented in grid format. The stick diagram in the microwind was utilised to generate patterns utilising the lambda design rule.

The lambda unit was reinforced using almost equal sections of the smallest quantity that could be used for lithography. It is built upon the reference metric "λ," which serves as the unit of measurement.

Value = mλ, where m is the scaling multiple.

V. Simulation Result

MICROWIND 2 and DSCH were used to successfully test and synthesise the suggested CSLA design. Microwind is a tool for layout-level circuit design and simulation. The performance of the suggested CSLA was assessed in this study using the microwind as a simulation tool. You may see the simulation's outcomes in a table. Data on area, power, and delay are all included in the table. A design's total cell area is equal to its total power, which is the sum of its internal, switching, and leakage powers.

Table 1: Results of the simulation measured in terms of power and area

BIT Size	**Adder**	**Gate Count**	**Power (Mw)**
4-	CSLA	136	0.867

Bit	BEC	95	0.417
	CBL	88	0.388
	PROPOSED	60	0.147
	CSLA (ALU)		
8-Bit	CSLA		
	BEC	272	3.114
	CBL	188	2.892
	PROPOSED	176	1.177
	CSLA (ALU)	120	0.797

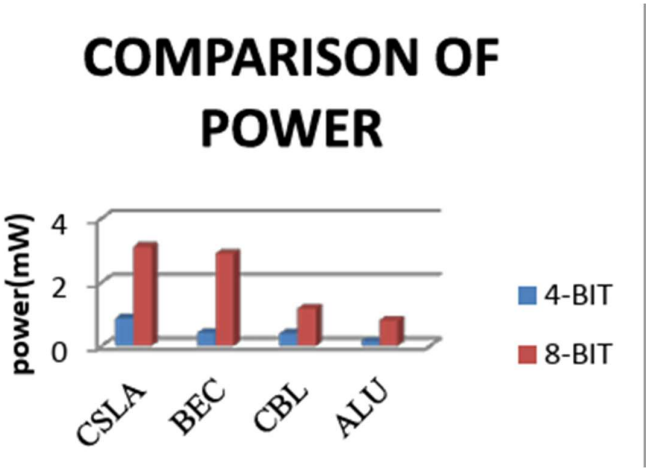

Fig. 5: Comparison of 8-Bit and 4-Bit power

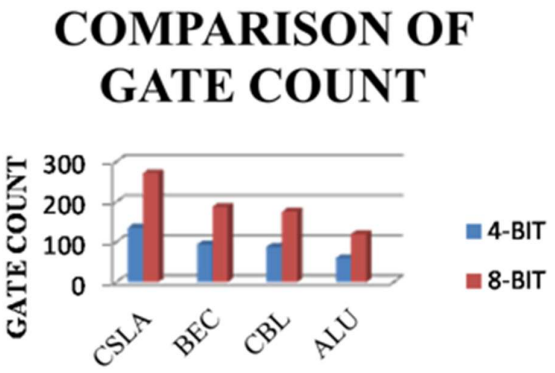

Fig. 6: Count of 8-bit and 4-bit gates compared

VI. **Conclusion**

This research presents an effective technique for reducing the CSLA architecture's area and power consumption. There are major advantages to reducing the number of gates, including reduced area and power consumption. The comparison found that the suggested CSLA design uses less energy and occupies less space. The low-area and low-power characteristics of the modified CSLA architecture make it well-suited for implementation in VLSI technology.

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, 2011.

[2]. Sharma, V. K. Enabling Mission-Critical Communication via VoLTE for Public Safety Networks. IJAIDR-Journal of Advances in Developmental Research, 10(1).

[3]. Ceiang, T. Y. and Hsiao, M. J. (Oct. 1998), “Carry-select adder using single ripple carry adder,” Electron. Lett., vol. 34, no. 22, pp. 2101-2103.

[4]. Ramkumar, B., Kittur, I.H.M., and Kannan, P.M. (2010), “ASIC implementation of modified faster carry save adder,” Eur. J. Sci. Res., vol. 42, no. I, pp. 53-58, 2010.

[5]. J.M. Rabaey, Digital Integrated Circuits: A Design Perspective. Upper Saddle River, NJ: Prentice-Hall, 2001.

[6]. Sharma, V. K. Enabling Mission-Critical Communication via VoLTE for Public Safety Networks. IJAIDR-Journal of Advances in Developmental Research, 10(1).

[7]. Y. Kim and L.-S. Kim, “64-bit carry-select adder with reduced area,” Electron. Lett., vol. 37, no. 10, pp. 614-615, May 2001.

[8]. "An area efficient 6+-bit square root carry-select adder for low power applications," from Y. He, C. H. Chang, and J. Gu, in Proc. IEEE Int. Symp. Circuits Syst. 2005, vol. 4, pp. 4082-4085.

[9]. Y. Kim and L.-S. Kim, “64-bit carry-select adder with reduced area,” Electron. Lett., vol. 37, no. 10, pp. 614-615, May 2001.

[10]. "An area efficient 6+-bit square root carry-select adder for low power applications," from Y. He, C. H. Chang, and J. Gu, in Proc. IEEE Int. Symp. Circuits Syst. 2005, vol. 4, pp. 4082-4085.