ISSN: 2395-1680 (ONLINE)
DOI: 10.21917/ijme.2026.0387

DESIGN OF PARALLEL ADDER USING HYBRID CLA-RCA STRUCTURE

Matcha Pavani, Poothi Venkata Lakshmi, Sabbavarapu Vivek Adithya and Reddy Giridhar
Department of Electronics and Communication Engineering, Lendi Institute of Engineering and Technology, India

Abstract

In this paper, we propose power efficient approximate carry look ahead adder (CLA) for error tolerant application. Power dissipation has become a significant issue for integrated circuit. To reduce power consumption, inexact or approximate computing has been considered as a solution where exactness is not required. Here, new approximate CLA is proposed which is based on modified equation of carry propagation. The proposed 32-bit approximate CLA is evaluated and compared with an accurate CLA and existing approximate CLA based on power and delay. The metrics of error are used to evaluate the reliability of the proposed approximate design. All circuits have been simulated in Cadence Virtuoso tool using 45-nanometer technology.

Keywords:

Approximate Computing, Carry Look-Ahead Adder, Carry Generation, Carry Propagation, Error Metrics

1. INTRODUCTION

Modern digital systems demand high-performance and energy-efficient VLSI designs, where arithmetic units play a crucial role. Among these, adders are widely used in processors and digital signal processing systems, directly influencing overall delay and power consumption. Therefore, designing adder architectures with reduced propagation delay and low power usage has become a key objective in VLSI design.

In many applications such as image and video processing, exact computational accuracy is not always required. These systems can tolerate small errors without significantly affecting output quality. This enables the use of approximate computing techniques, where simplified circuit designs are employed to achieve improvements in power efficiency and speed.

The Carry Look-Ahead Adder (CLA) is widely used for high-speed operations due to its ability to reduce delay through parallel carry computation. However, this advantage comes with increased hardware complexity and higher power consumption compared to simpler architectures such as the Ripple Carry Adder (RCA).

To overcome this limitation, this paper proposes an approximate CLA by modifying the carry propagation logic, aiming to reduce both delay and power consumption while maintaining acceptable accuracy. The proposed design achieves lower transistor count, reduced propagation delay, and significant power savings, making it suitable for error-tolerant applications.

The novelty of this research lies in the design of a power-efficient approximate Carry Look-Ahead Adder (CLA) that simplifies the carry propagation mechanism while preserving acceptable computational accuracy for error-tolerant applications. Unlike conventional CLA architectures that prioritize speed at the expense of increased hardware complexity and power consumption, the proposed architecture employs an optimized carry generation strategy to significantly reduce transistor count, propagation delay, and energy dissipation without introducing excessive computational errors. The proposed design is particularly suitable for applications such as digital signal processing, multimedia processing, and machine learning accelerators, where minor arithmetic inaccuracies have minimal impact on overall system performance.

The remainder of this paper is organized as follows: Section 2 describes the conventional CLA architecture, Section 3 presents the proposed approximate CLA design, Section 3.1 discusses the error analysis, Section 3.2 presents the simulation comparison, Section 3.3 represents the Design of inexact 2-bit adder and Section 6 concludes the paper.

2. CONVENTIONAL CLA

Adders are fundamental components in digital systems, as most arithmetic operations depend on them. The overall speed of these operations is highly influenced by the delay in adder circuits. In Ripple Carry Adders (RCA), the major limitation is the carry propagation delay, where the carry must propagate through each stage sequentially. To overcome this issue, Carry Look-Ahead Adders (CLA) are used, which reduce delay by computing carry signals in advance. In CLA, a carry will be generated in two cases. When both the input bits A and B are high i.e. '1' or when one of the two input bits is '1'. Carry-out is '1' if carry generation (CG_i) is '1' or both carry propagation (CP_i) and carry-in are '1'. So, the carry generation and propagation function are given by AND, OR operations.

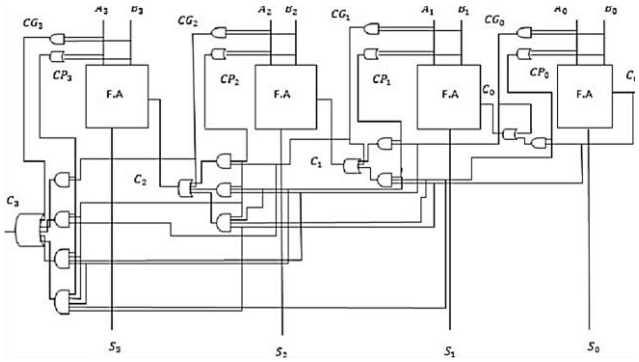

Fig.1. Diagram of 4-bit CLA

$$CP_i = A_i + B_i \tag{2}$$

The carry output at each stage can be expressed as:

$$C_i = CG_i + CP_i \cdot C_{in} \tag{3}$$

For a 4-bit CLA, the carry outputs are expanded as follows:

$$C_0 = CG_0 + CP_0 \cdot Cin \tag{4}$$

$$C_1 = CG_1 + CP_1CG_0 + CP_1CP_0C_{in} \tag{5}$$

$$C_2 = CG_2 + CP_2CG_1 + CP_2CP_1CG_0 + CP_2CP_1CP_0C_{in} \tag{6}$$

$$C_3 = CG_3 + CP_3CG_2 + CP_3CP_2CG_1 + CP_3CP_2CP_1CG_0 + CP_3CP_2CP_1CP_0C_{in} \tag{7}$$

From the above expressions, it can be observed that the carry outputs depend only on the input signals and the initial carry-in, rather than waiting for sequential propagation. This parallel computation significantly reduces delay, making CLA faster than RCA However, this improvement in speed comes at the cost of increased hardware complexity and power consumption due to additional logic required for carry computation.

Expression of C_{out} of each full-adder of CLA is dependent only on the initial carry-in, C_{in}, its CG_i and CP_i functions of the previous adder. Since CG_i and CP_i functions can be expressed in terms of inputs A and B of the full adders, carry-out of each adder are generated easily and there is no need to wait for a carry signal to propagate from each adder to next adder before a final result is achieved. Hence the CLA speeds up the addition process.

3. PROPOSED APPROXIMATE CLA

In conventional CLA, the equation of output carry of each stage for 4-bit CLA is discussed above in the Eq.(4). The function carry generation and carry propagation of fifth adder of CLA is expressed as:

$$CG_i = A_i.B_i \tag{8}$$

$$CP_i = A_i + B \tag{9}$$

The carry output of fifth adder of CLA is given by:

$$C_i = CG_i + CP_iC_{in} \tag{10}$$

By, putting the value of *CG* and *CP* in the Eq.(10), we *i* get,

$$C_i = A_i.B_i + (A_i+B_i).C_{in} \tag{11}$$

The delay and power of CLA increases with increasing the area of CLA. So, in this paper, we propose approximate adders that uses only carry propagation part. The output carry i.e. Eq.(11) can be rewritten as:

$$C_i = (A_i + B_i).C_{in} \tag{12}$$

The above equation is called approximate value where carry will be the input of next stage. As mentioned before, in the proposed CLA, we modified the equation of carry propagation with the help of truth table of OR gate.

Table.1. Truth Table of OR Gate

A	B	Y
0	0	0
0	1	1
1	0	1(0)
1	1	1

If we introduce error in the 3rd bit of the output and assume *Y*=0 (in bracket) in truth table of OR gate shown in Table.1, then the value of *Y* will be the same as *B*. Thus, again we modified the above Eq.(12) to reduce the number of transistors. So, the Eq.(12) can be rewritten as:

$$C.C_i = B_i\ C_{in} \tag{13}$$

Eq.(13) designs the Approximate CLA in this paper. By using this, the number of transistors is reduced, which in turn reduces the area of the Approximate CLA. So, the power and delay will also reduce. The diagram of proposed Approximate CLA is shown in Fig.2 below where the carry propagation part is approximated.

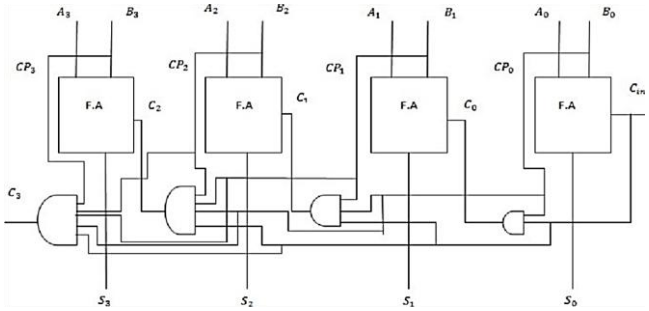

Fig.2. Proposed Approximate CLA

3.1 ERROR ANALYSIS OF PROPOSED APPROXIMATE CLA

In this section, the error metric has given to evaluate the accuracy of the proposed Approximate CLA. The comparative study includes the proposed Approximate CLA and existing Approximate CLA with the conventional CLA. Here, the function of the proposed design, the existing approximate design and the conventional design of CLA are simulated in MATLAB.

To reduce power consumption, inexact or approximate computing has been considered as a solution where exactness is not required. Here, new approximate CLA is proposed which is based on modified equation of carry propagation. The proposed 32-bit approximate CLA is evaluated and compared with an accurate CLA and existing approximate CLA based on power and delay. The metrics of error are used to evaluate the reliability of the proposed approximate design. All circuits have been simulated in Cadence Virtuoso tool using 45-nm technology.

A Carry Look-Ahead Adder (CLA) consumes significantly more power than a Ripple Carry Adder (RCA) because the CLA is designed for high speed, which is achieved by pre-calculating carries, a process requiring more complex logic and thus more power. In contrast, the RCA is the slowest but the most power-efficient adder design, as it generates carries serially, "rippling" them from one bit to the next, which requires less complex circuitry. There is a fundamental trade-off between speed and power in digital circuit design. When designing adders for specific applications, designers must select a topology like a CLA or RCA based on the performance priorities of that system.

The error metrics considered for the accuracy evaluation are as following: Error Distance (ED): It is defined to be the arithmetic. Difference between the accurate product (M) and the approximate product, (M_0), i.e., ED = $|M_0-M|$. Mean Error Distance (MED): It is the average of Error Distance. Normalized Mean Error Distance (NMED): It is the normalization of MED by the maximum output value, $NMED=MED/P_{max}$, where P_{max} is the maximum magnitude of the output of an accurate design.

In Table.2, the error metrics for the approximate CLA under all possible input combinations of and 4-bit adder are reported. In Table.2, we did comparison between proposed Approximate CLA and existing Approximate CLA. The total Error Distance, MED, NMED are less in the proposed Approximate CLA as compared to the existing Approximate CLA.

Table.2. Error Metrics of the Proposed CLA Compared to the Existing Approximate CLA

Design	Total Error Distance	MED	NMED
Existing Approximate CLA	9.96	3.39	2.59
Proposed Approximate CLA	5.15	2.01	1.34

3.2 SIMULATION COMPARISON

In this section, the performance summary of the proposed 32-bit Approximate CLA and existing Approximate CLA are studied. The circuits of 32-bit proposed Approximate CLA and existing Approximate CLA as well as conventional CLA have been simulated in Cadence Virtuoso tool using 45-nm technology at 1V power supply. The proposed 32-bit CLA is designed using eight blocks of 4-bit Approximate CLA. The results based on some input samples for the power, delay of the 32-bit Approximate CLA, existing Approximate CLA and conventional CLA have been given in Table II. The maximum delay among all outputs has been considered here. The power consumption, delay and number of transistors have reduced as compared to the conventional CLA and existing Approximate CLA. The proposed approximate CLA has power savings 61% and 25% less delay as compared to the existing approximate design and power savings 63% and 42% less delay as compared to the conventional CLA. The proposed design is evaluated using power, delay, and error metrics.

First method uses less space, less power, and causes minimum delay, but at the expense of precision. The researchers can approximate a piece of an n-bit adder using the second design process, which does not rely on an approximation of a single bit. Compared to the adders developed using the first process, this yields a substantially lower error [13]. The carry speculation method calculates the sum output of such a block (multi-bit) [15]. The foundation of this design is the observation that a long carry chain is infrequently produced by adding two random inputs. As a result, multi-bit approximation adders are the preferred option for study in the modern research.

The simulation results clearly demonstrate that the proposed approximate CLA achieves an effective trade-off between computational accuracy and hardware efficiency. The reduction in transistor count contributes directly to lower switching activity, thereby minimizing dynamic power consumption and improving overall energy efficiency. Furthermore, the simplified carry propagation logic shortens the critical path, resulting in reduced propagation delay without significantly affecting the correctness of arithmetic operations. Error analysis confirms that the generated inaccuracies remain within acceptable limits for error-resilient applications, making the proposed architecture suitable for multimedia processing, digital signal processing, and machine learning accelerators. Compared with existing approximate CLA designs, the proposed architecture offers superior power-delay performance while maintaining reliable operational characteristics. These results validate the effectiveness of the proposed approximation strategy for developing next-generation low-power VLSI arithmetic circuits.

Table.3. Performance Comparison of CLA, existing CLA and Proposed Approximate CLA

Design	Power (µw)	Propagation Delay(ns)	Number of Transistors
Conventional CLA	7.0	0.1508	1968
Existing Approximate CLA	6.68	0.1169	1056
Proposed Approximate CLA	2.55	0.0870	480

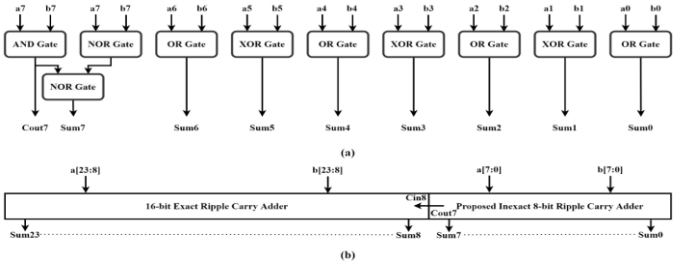

Fig.3.(a) Proposed inexact 8-bit ripple carry adder, (b) Block diagram of 24-bit inexact mantissa adder

3.3 DESIGN OF INEXACT 2-BIT ADDER

The mantissa adder consists of 24 bits [6]. It is a bit tricky while approximating such carry merger adder. The weight of the sum bits is increasing with the increase in higher order bits. Approximating higher significant bits could affect the error metrics significantly. Therefore, in this research work, eight lower significant bits are approximated and the rest 16 higher significant bits are added using exact full adders. The inexact multi-bit adder proposed in [16] is used in this work as the proposed inexact mantissa adder. The proposed inexact adder is designed by considering the addition of 2-bits. The 2-bit adder takes five bits (a0, b0, a1, b1, and Cin) as input and gives three bits (Cout1, Sum1, and Sum0) as the output. The block diagram of the proposed 2-bit adder is shown in Fig.3 and the Boolean expressions of the 2-bit adder is given as

$$Sum_0 = a_0 \oplus b_0 \oplus C_{in} \quad (14)$$

$$Sum_1 = a_1 \oplus b_1 \oplus Cout_0 \quad (15)$$

$$Cout_1 = a_1.b_1 + (a_1+b_1)\, Cout_0 \quad (16)$$

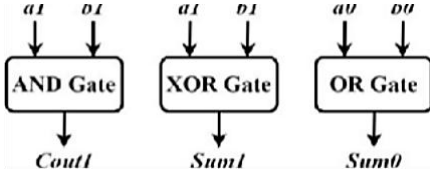

Fig.4. Proposed inexact 2-bit adder

The approximation of exact 2-bit adder is carried out by modifying the truth table. The probability of occurrence of the input pattern having maximum number zeros is the maximum. Thereby, we try to locate the errors in the low probable input patterns. In addition, we try to locate the errors of Cout1, Sum1, and Sum 0 outputs for the same input pattern. The two lower bits (a0 and b0) are added with an OR gate and the output is Sum0. Similarly, the next two bits (a1 and b1) are given to an XOR gate to generate Sum1. Meanwhile, a1 and b1 are also given to an AND

gate to generate the Cout1, which is the carry in bit to the next subsequent exact adder. The Sum 1 and Sum 0 of the 2-bit inexact adder offers 4 errors each for the input pattern of 0101, 0111, 1101, and 1111. Meanwhile, Cout1 offers only two errors for the input pattern of 0111 and 1101. Thereby, it is observed that for these 4 input patterns the output is erroneous. It is noteworthy that locating errors for different outputs to the same input pattern could help in reducing the error metrics. To expand this adder to add number of bits, XOR-OR architecture could be connected in the more significant side and the AND gate should be connected in the last to generate the carry bit respectively.

4. CONCLUSION

In this paper, a power-efficient approximate CLA has been presented for error-tolerant applications. The proposed design simplifies the carry propagation logic, resulting in a significant reduction in transistor count, power consumption, and propagation delay compared to conventional and existing approximate CLA architectures. Simulation results demonstrate that the proposed 32-bit approximate CLA achieves approximately 63% reduction in power consumption and 42% improvement in delay compared to conventional CLA, while maintaining acceptable error levels. Due to its improved efficiency and controlled accuracy, the proposed design is well suited for applications such as digital signal processing and multimedia systems, where minor computational errors are tolerable. Future work can focus on extending the design to higher bit-width architectures and optimizing accuracy-performance trade-offs further.

Future research will focus on enhancing the proposed approximate Carry Lookahead Adder (CLA) by developing adaptive approximation techniques that dynamically adjust the level of approximation based on application-specific accuracy requirements. The architecture can be extended to 64-bit and 128-bit designs to evaluate its scalability for high-performance computing systems. Further optimization of the carry generation logic using emerging technologies, such as FinFET, Gate-All-Around (GAA) transistors, and carbon nanotube field-effect transistors (CNTFETs), can improve energy efficiency and reduce leakage power. Additionally, integrating the proposed approximate CLA into arithmetic units such as multipliers, multiply-accumulate (MAC) units, and digital signal processors will enable comprehensive evaluation in real-world applications, including image processing, machine learning accelerators, and edge computing devices. Future studies may also investigate hybrid approximation techniques that balance power savings with computational accuracy, along with FPGA and ASIC implementations to validate hardware performance under practical operating conditions. These advancements will further enhance the applicability of the proposed design in next-generation low-power and error-resilient computing systems.

REFERENCES

[1] B.K. Mohanty and S.K. Patel, "Area-Delay-Power Efficient Carry Select Adder", *IEEE Transactions on Circuits and Systems II: Express Briefs*, Vol. 61, No. 6, pp. 418-422, 2014.

[2] R. Hegde and N.R. Shanbhag, "Soft Digital Signal Processing", *IEEE Transactions on Very Large-Scale Integration (VLSI) Systems*, Vol. 9, No. 6, pp. 813-823, 2001.

[3] B. Shim, S. Sridhara and N. Shanbhag, "Reliable Low-Power Digital Signal Processing via Reduced Precision Redundancy", *IEEE Transactions on Very Large-Scale Integration (VLSI) Systems*, Vol. 12, No. 5, pp. 477-488, 2004.

[4] D. Shin and S.K. Gupta, "Approximate Logic Synthesis for Error Tolerant Applications", *Proceedings of International Conference on Design, Automation and Test in Europe*, pp. 957-960, 2010.

[5] H. Jiang, J. Han, and F. Lombardi, "A Comparative Review and Evaluation of Approximate Adders", *Proceedings of IEEE/ACM Great Lakes Symposium on VLSI*, pp. 343-348, 2015.

[6] A.K. Verma, P. Brisk and P. Ienne, "Variable Latency Speculative Addition: A New Paradigm for Arithmetic Circuit Design", *Proceedings of International Conference on Design, Automation and Test in Europe*, pp. 1250-1255, 2008.

[7] V.K. Chippa, S.T. Chakradhar, K. Roy and A. Raghunathan, "Analysis and Characterization of Approximate Computing: Resilience for Approximate Computing", *Proceedings of 50th Annual Design Automation Conference*, pp. 1-9, 2013.

[8] A. Raha, H. Jayakumar and A. Raghunathan, "Input-Based Dynamic Reconfiguration of Approximate Arithmetic Units for Video Encoding", *IEEE Transactions on Very Large-Scale Integration (VLSI) Systems*, Vol. 24, No. 3, pp. 1111-1124, 2016.

[9] R. Ye, T. Wang, F. Yuan, R. Kumar and Q. Xu, "On Reconfiguration-Oriented Approximate Adder Design and its Application", *Proceedings of IEEE/ACM International Conference on Computer-Aided Design*, pp. 48-54, 2013.

[10] J. Hu and W. Qian, "A New Approximate Adder with Low Relative Error and Correct Sign Calculation", *Proceedings of International Conference on Design, Automation and Test in Europe*, pp. 1449-1454, 2013.