Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 3, No. 1, March : 2024

DESIGN OF 4-BIT CARRY LOOK-AHEAD ADDER USING HYBRID DESIGN METHODOLOGY

B. Sai Prakash Reddy, PG Scholar, Department of Electronics and Communication Engineering, CMR Engineering College (UGC – Autonomous), Kandlakoya, Medchal, Telangana, India
Dr. S. Poongodi, Professor, Department of Electronics and Communication Engineering, CMR Engineering College (UGC – Autonomous), Kandlakoya, Medchal, Telangana, India

Abstract

The Carry Look Ahead Adder (CLA) is a critical component in digital arithmetic circuits, facilitating high-speed and efficient binary addition operations. In conventional Gate Diffusion Input (GDI)-based Ripple Carry Adder (RCA) systems poses with power consumption and delay times pose significant challenges, limiting their performance in high-speed applications. The cascading of carry bits in GDI-RCA designs results in propagation delays and increased power dissipation, leading to suboptimal performance and energy inefficiency. Additionally, the complexity of GDI implementations often results in larger transistor counts and higher fabrication costs, further exacerbating the limitations of existing systems. Our proposed methodology for designing the 4-bit CLA using Pass Transistor Logic (PTL) techniques focuses on optimizing power consumption, reducing delay times, and enhancing overall performance. By employing PTL-based logic gates and parallel carry lookahead computation, we aim to minimize propagation delays and power dissipation in the CLA circuit. Furthermore, our approach involves exploring innovative circuit topologies and design strategies to achieve higher integration densities and lower transistor counts. Through comprehensive simulations and evaluations, we demonstrate the effectiveness of our proposed methodology in improving the efficiency and speed of 4-bit CLA designs implemented using PTL.

Keywords: Carry Look Ahead Adder, Ripple Carry Adder, Pass Transistor Logic, Gate Diffusion Input.

1. Introduction

The development of adders in VLSI technology has been a crucial aspect of advancing digital circuit design. The history of adder circuits traces back to the early days of computing, where the need for efficient arithmetic operations was paramount. Initially, adders were implemented using discrete components, such as vacuum tubes and later transistors, forming the backbone of early digital computers. As integrated circuit technology emerged, the miniaturization of components led to the integration of adders onto a single chip, paving the way for more compact and efficient designs. In the 1960s and 1970s, with the advent of medium-scale integration (MSI) and large-scale integration (LSI), adder circuits began to be realized in integrated form. Early VLSI adder designs were based on basic logic gates like AND, OR, and XOR gates, arranged in specific configurations to perform addition operations. These adders were relatively simple but served the needs of early digital systems.

However, as the demand for faster and more complex computations increased, so did the need for more efficient adder designs. This led to the development of advanced adder architectures such as carry-lookahead adders, carry-select adders, and carry-skip adders. These designs aimed to reduce the critical path delay and improve the overall speed of addition operations. The evolution of semiconductor technology further fueled advancements in VLSI adder design. With the transition to CMOS technology and the introduction of deep-submicron processes, designers could pack more transistors onto a chip, enabling the implementation of more sophisticated adder structures. In recent years, the focus has shifted towards optimizing adder designs for low power consumption and high-speed operation. Techniques such as parallel prefix adders and hybrid adder structures have been proposed to address these challenges. Moreover, the emergence of field-programmable gate arrays (FPGAs) and application-specific integrated circuits (ASICs) has provided designers with customizable platforms to implement tailored adder architectures for specific applications.

2. Literature survey

With the rapid development and wide application of information technology, signal processing algorithms are widely being used in portable wireless devices, such as smartphones, PCs, and wearable devices. Full adders and multipliers are fundamental components in digital signal processing applications [1], such as convolution, fast Fourier transform (FFT) [2,3], finite impulse response (FIR) [4,5], discrete cosine transform (DCT) [6,7], infinite impulse response (IIR) filters [8], and audio/video codecs. Conventional multipliers are becoming the bottleneck of low-power digital signal processing applications [9,10]. Generally, multipliers could be classified into various types, such as array [11,12], Booth [13,14], carry-save, and Wallace tree [15,16], according to the methods used to produce, pass, and compress the partial products. In an array multiplier, the partial product is generated by the one-bit multiplication of the multiplicand and multiplier, mostly conducted by AND gates. The partial products are directly summed up by an array of adders. The array multiplier has an explicit structure [17], which makes it easy to design and analyze. However, as the multiplier bit width increases, the critical path increases dramatically. Instead of passing the output carry to the same-level adder, carry-save array adders pass both carry and sum to the next-level adders. This reduces the carry propagation delay in all rows except the last row. Hence, it reduces both the length and the number of critical paths compared to the array multiplier. Wallace tree methods use fewer adders for compression and accumulation. The partial product bits are summed up in parallel by means of a tree of carry-save adders. They compress three or four inputs into two outputs and continue the next-level compression with fewer adders. Full adders are the most important components of multipliers, which in turn increases the demand for low-power full adders for high-performance multipliers [18]. Complementary metal–oxide–semiconductor (CMOS) full adders are most widely used, especially in the digital standard cells of many CMOS technologies. However, compared to pass-transistor logic (PTL)-based circuits, they consume more power. PTL full adders might be significant for a high-performance multiplier [19, 20, 21].

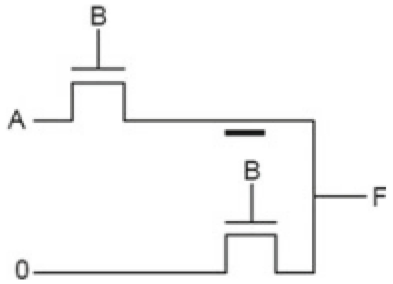

Fig. 1 The basic PTL cell

3. Proposed methodology

Pass transistor logic is a crucial logic style to design the integrated circuits because it uses less transistors, runs faster, and requires less power [7]. By eliminating redundant transistors, it reduces the count of transistors used to make different logic gates. PTL uses N number of transistors instead of 2N, and also, it has no static power consumption which makes it more powerful and successive logic style among all available logic styles [8]. PTL is bidirectional in nature. By using this technique, the combinational logic gates like AND, OR, and EX-OR are designed using four transistors, whereas six transistors are used for OR, and, AND logic gates and for EX-OR gate the transistor count is 8, using CMOS design [9]. Hence, the PTL is an effective technique to reduce the transistor count and also to achieve less power consumption with high-speed performance (Fig. 1). In other logic family's input is applied to gate terminal of transistor, but in PTL it is also applied to source/drain terminal. PTL circuits behave as switches use either NMOS transistors or parallel pair of NMOS and PMOS transistor called transmission gate. In this design the width of PMOS is taken equal to NMOS so that both the transistors can pass the signal simultaneously in parallel.

3.1 Design of Basic Gates Using PTL

The proposed design of basic gates such as inverter, two-input AND gate, two-input OR gate, two-input EX-OR gate is represented in Figs. 2, 3, 4, and 5, respectively. All the proposed designs are designed, simulated, and analyzed by using Cadence Virtuoso simulation tool at 180 nm technology with supply voltage of 1.8 V at an operating frequency of 5 MHz based on pass transistor logic.

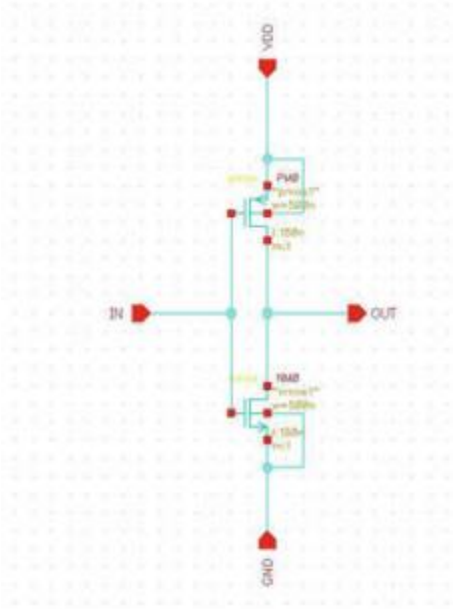

Fig. 2 Schematic diagram of Inverter

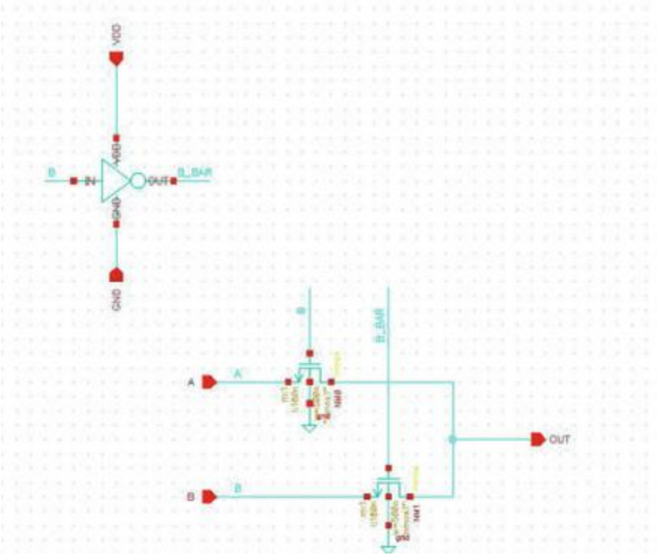

Fig. 3 Schematic diagram of two-input AND gate

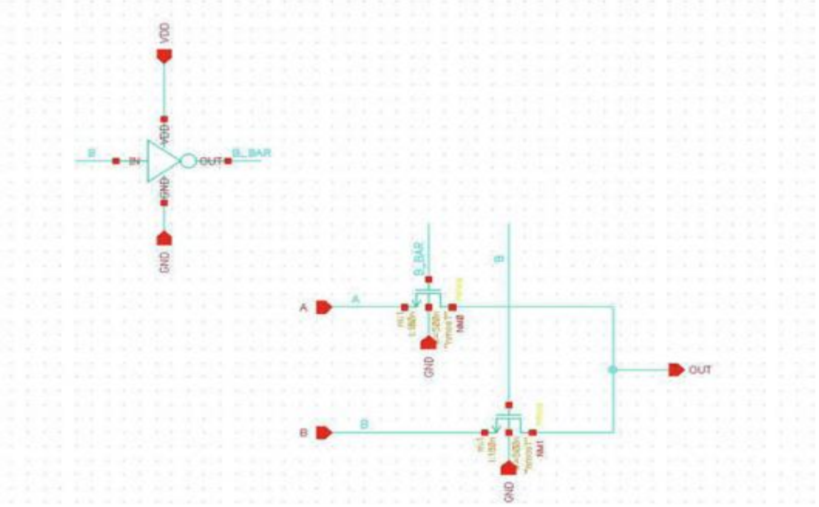

Fig. 4 Schematic diagram of two-input OR gate

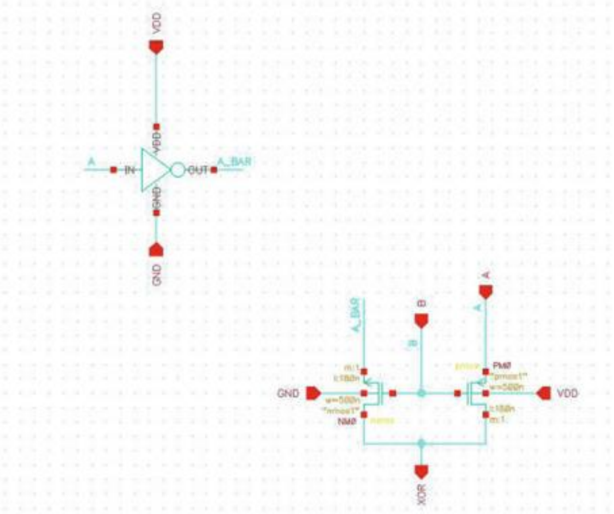

Fig. 5 Schematic diagram of two-input EX-OR gate

Design of Carry Look-Ahead Adder (CLA)

The basic adder which first comes to mind is Ripple carry adder. But it has some limitation; that is the propagation delay to calculate the carry bit. It takes too much time to propagate the carry. To overcome this type of limitation, carry look-ahead adder comes into consideration which calculates the carry bit in advance based on the input values given to the adder. To understand the working principle of Carry look-ahead adder, let us see the modified Boolean expressions for propagate P and generate G (Figs. 6 and 7).

$Pi = ai \oplus bi$ (1)

$Gi = ai \cdot bi$ (2)

For 4-bit CLA Summations are given from Eqs. 3–6 as follows:

$Sum0 = a0 \oplus b0 \oplus c0$ (3)

$Sum1 = a1 \oplus b1 \oplus c1$ (4)

$Sum2 = a2 \oplus b2 \oplus c2$ (5)

$Sum3 = a3 \oplus b3 \oplus c3$ (6)

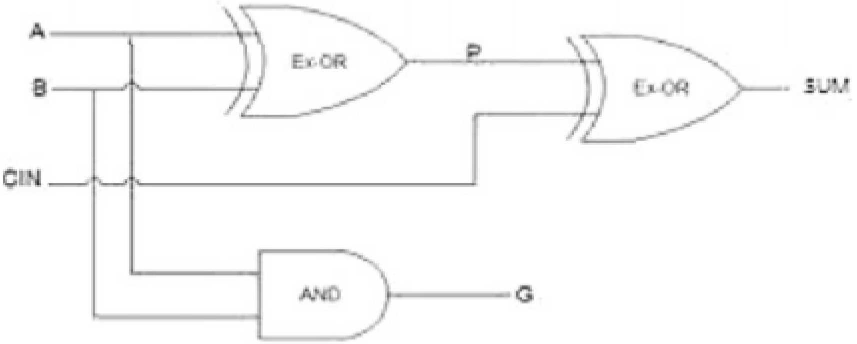

Fig. 6 Gate-level representation of B-cell

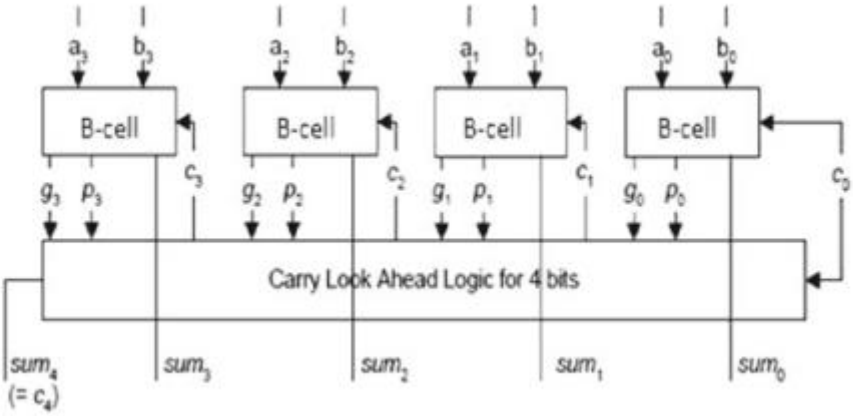

Fig. 7 Block diagram of 4-bit CLA

Carry-out are given from Eqs. 7–10 as follows

$c1 = g0 + p0 \cdot c0$ (7)

$c2 = g1 + p1 \cdot g0 + p1 \cdot p0 \cdot c0 = g1 + p1\ g0 + p0 \cdot c0 = g1 + p1 \cdot c1$ (8)

$c3 = g2 + p2 \cdot g1 + p2 \cdot p1 \cdot g0 + p2 \cdot p1 \cdot p0 \cdot c0 = g2 + p2\ g1 + p1 \cdot g0 + p1 \cdot p0 \cdot c0 = g2 + p2 \cdot c2$ (9)

$c4 = g3 + p3 \cdot g2 + p3 \cdot p2 \cdot g1 + p3 \cdot p2 \cdot p1 \cdot g0 + p3 \cdot p2 \cdot p1 \cdot p0 \cdot c0 = g3 + p3\ g2 + p2 \cdot g1 + p2 \cdot p1 \cdot g0 + p2 \cdot p1 \cdot p0 \cdot c0 = g3 + p3 \cdot c3$ (10)

By using the basic gates such as Inverter, AND gate, OR gate, and EX-OR gate which have been designed earlier based on pass transistor logic (PTL), let us design the B-cell first and then by using B-cells we can easily construct our proposed 4-bit CLA circuit.

4. Results and discussion

Figure 8 depicts the schematic diagram of a PTL AND gate. In this schematic, the PTL technique is employed to implement the logical AND operation between two input signals. The AND gate functionality is achieved by utilizing pass transistors to control the flow of signals based on the input conditions. Figure 9 presents the simulation waveform of the PTL-AND gate schematic depicted in Figure 8. The waveform illustrates the behavior of the PTL-AND gate under different input conditions and provides insights into its logical operation. Specifically, the waveform shows the input signals (ai and bi) and the corresponding output signal (OUT) as a function of time or input transitions. When both input signals are high (logic 1), the output signal transitions to a high state, indicating a logical AND operation. Conversely, when any of the input signals is low (logic 0), the output signal remains low, reflecting the behavior of the AND gate. The simulation waveform confirms the correct functionality of the PTL-AND gate and validates its performance in producing the expected output based on the input conditions.

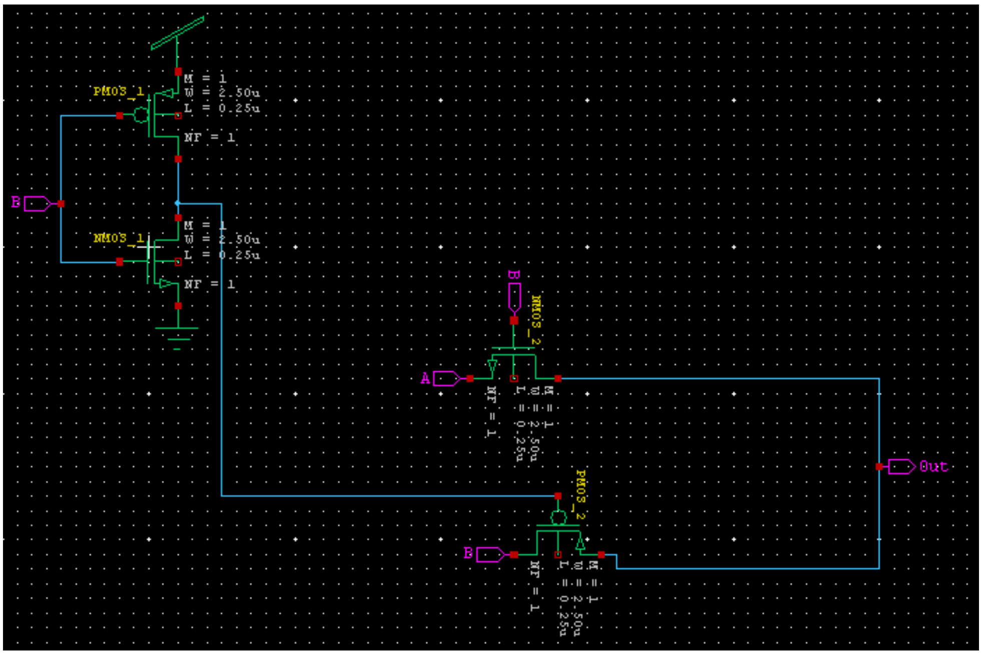

Figure 8: PTL-AND Schematic.

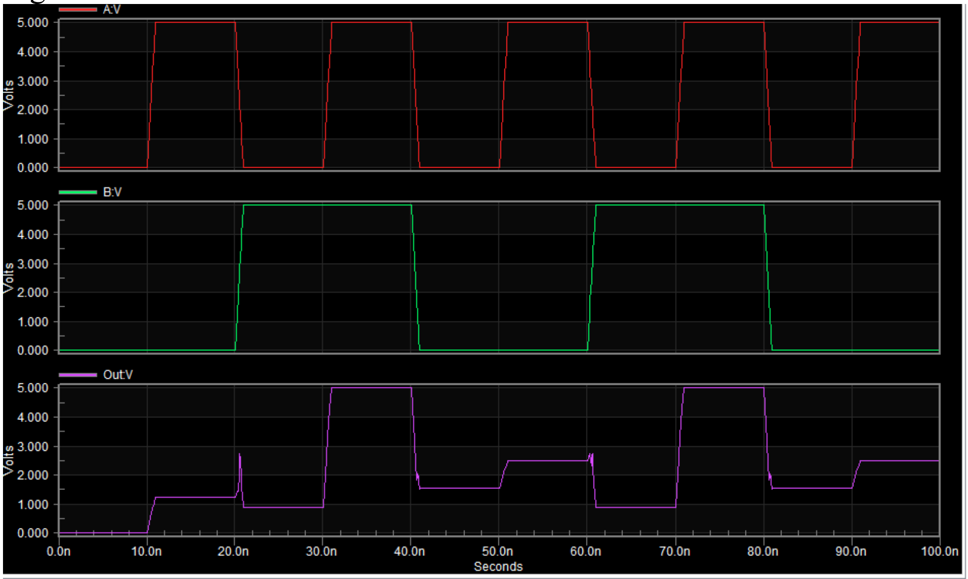

Figure 9: Simulation Waveform of PTL-AND Schematic.

Figure 10 illustrates the schematic diagram of a PTL OR gate. Similar to the PTL-AND gate, the PTL-OR gate utilizes pass transistors to implement the logical OR operation between two input signals. Figure 11 displays the simulation waveform of the PTL-OR gate schematic depicted in Figure 10. The waveform showcases the behavior of the PTL-OR gate under various input conditions and demonstrates its logical operation. Specifically, the waveform depicts the input signals (ai and bi) and the resulting output signal (OUT) as a function of time or input transitions. When either of the input signals is high (logic 1), the output signal transitions to a high state, indicating a logical OR operation. Conversely, when both input signals are low (logic 0), the output signal remains low, reflecting the behavior of the OR gate. The simulation waveform validates the correct operation of the PTL-OR gate and confirms its ability to produce the expected output based on the input conditions.

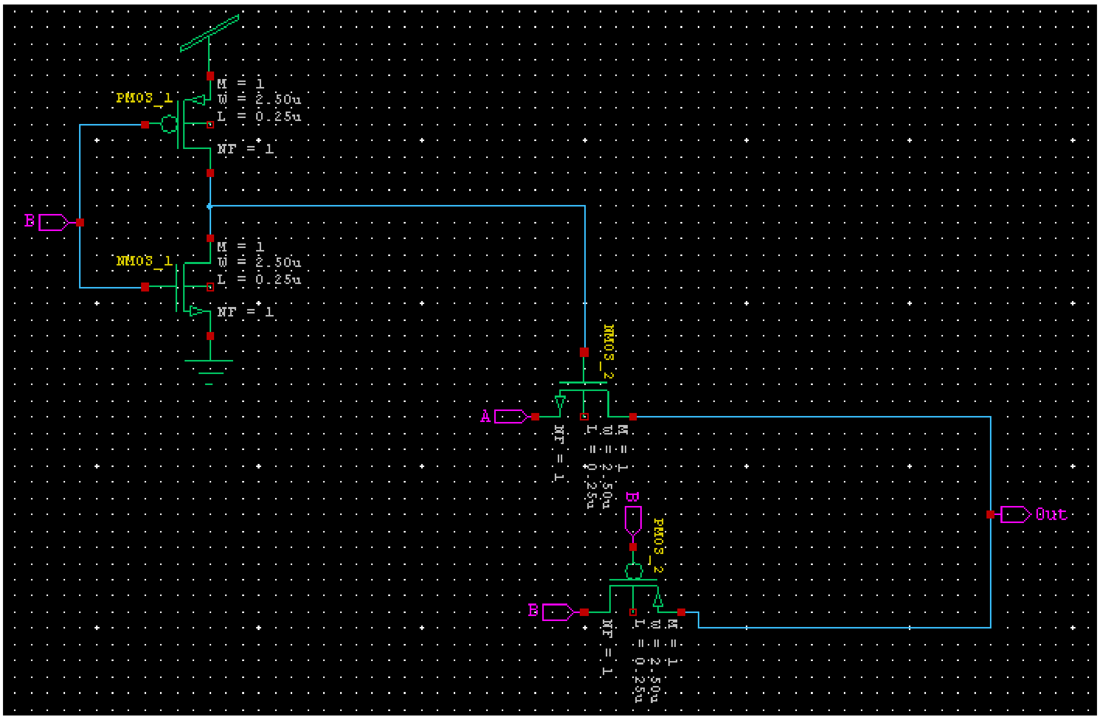

Figure 10: PTL-OR Schematic.

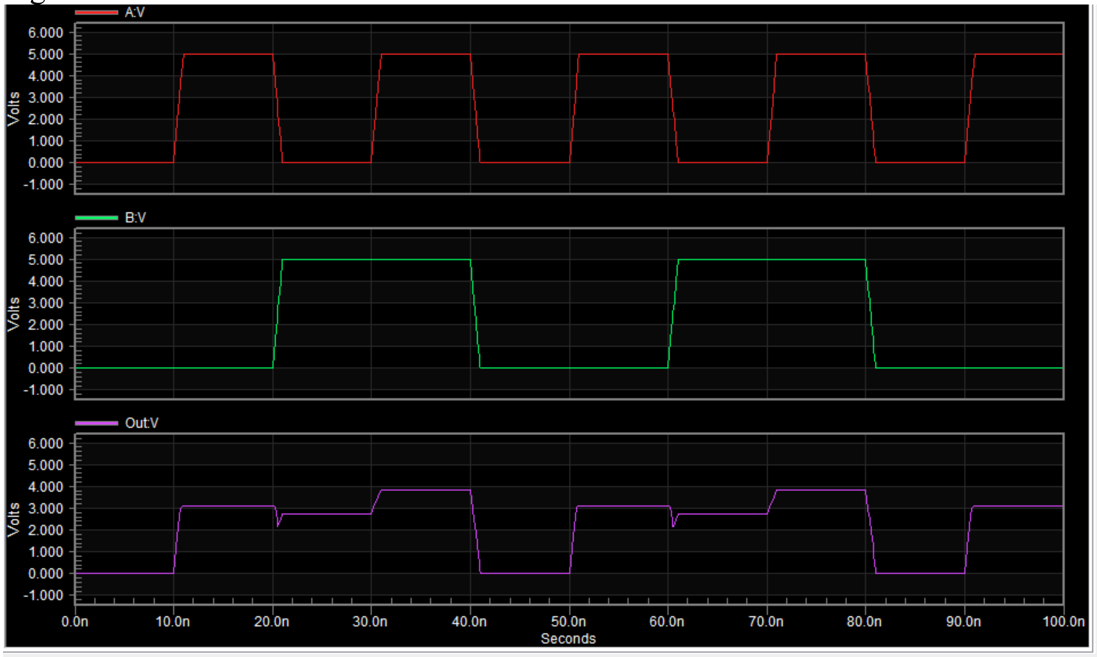

Figure 11: Simulation Waveform of PTL-OR Schematic.

Figure 12 showcases the schematic diagram of a PTL XOR gate. The PTL-XOR gate utilizes pass transistors to implement the logical exclusive OR (XOR) operation between two input signals. Figure 13 presents the simulation outcome of the PTL-XOR gate schematic depicted in Figure 12. The simulation outcome depicts the behavior of the PTL-XOR gate under different input conditions and provides a visual representation of its logical operation. Specifically, the outcome illustrates the input signals (ai and bi) and the resulting output signal (OUT) as a function of time or input transitions. When the input signals are different (one high and one low), the output signal transitions to a high state, indicating a logical XOR operation. Conversely, when the input signals are the same (both high or both low), the output signal remains low, reflecting the behavior of the XOR gate. The simulation outcome validates the correct functionality of the PTL-XOR gate and confirms its ability to produce the expected output based on the input conditions.

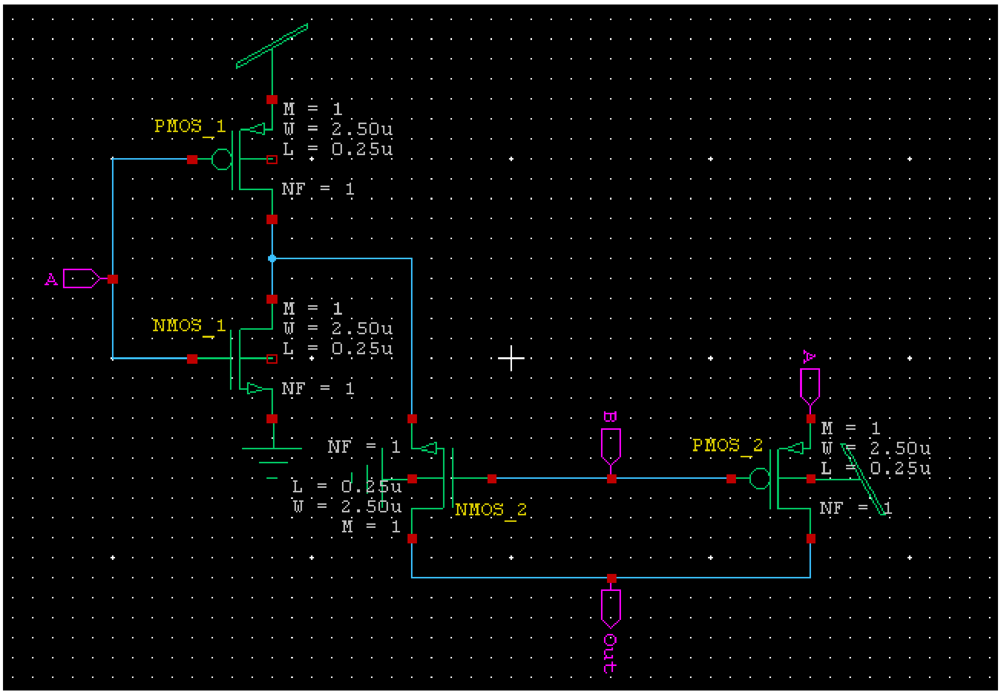

Figure 12: PTL-XOR Schematic.

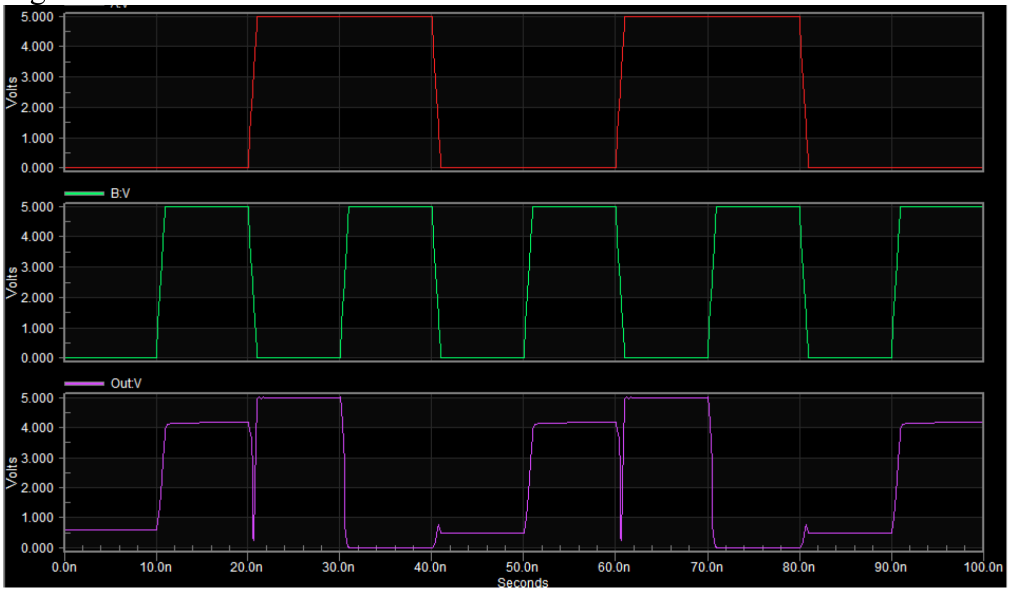

Figure 13: Simulation outcome of PTL-XOR Schematic.

Figure 14 depicts the simulation circuit of a 4-bit CLA. This circuit represents the hierarchical structure of the CLA, which comprises multiple full-adder cells interconnected to perform binary addition operations. The simulation circuit includes input signals (a0 to a3, b0 to b3, and cin) representing the binary numbers to be added and an output signal representing the sum output of the addition operation. The CLA simulation circuit facilitates the analysis and verification of the CLA's performance under different input conditions and provides insights into its speed and efficiency in performing binary addition. Figure 15 presents the waveform generated from the simulation of the 4-bit CLA circuit depicted in Figure 14. The waveform showcases the behavior of the CLA under various input conditions and provides a visual representation of its performance in binary addition operations. Specifically, the waveform illustrates the input signals (a0 to a3, b0 to b3, and cin) and the resulting output signal (sum) as a function of time or input transitions. The waveform confirms the correct functionality of the CLA and validates its ability to produce the expected sum output based on the input numbers and carry-in bit. Additionally, the waveform may reveal insights into the speed and efficiency of the CLA in performing binary addition operations, facilitating further analysis and optimization if necessary.

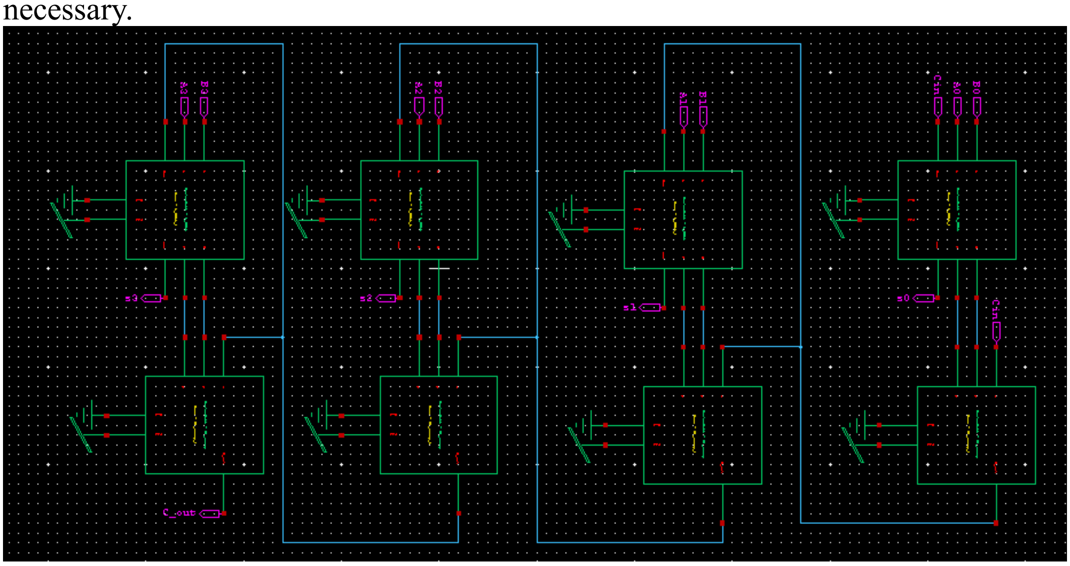

Figure 14: 4-Bit CLA Simulation Circuit.

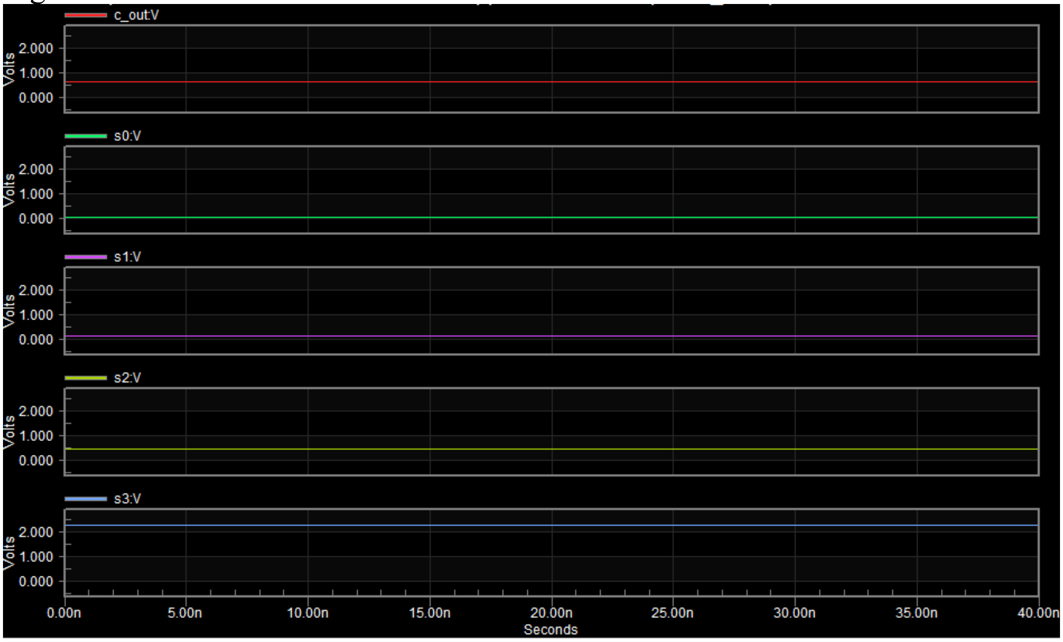

Figure 15. 4-Bit CLA waveform.

5. Conclusion

In conclusion, the 4-bit CLA represents a significant advancement in digital arithmetic circuit design, offering high-speed operation, reduced power consumption, and improved integration density compared to conventional RCA. The CLA utilizes a parallel carry lookahead technique to compute carry bits in advance, enabling faster carry propagation and facilitating high-speed addition operations. By leveraging PTL techniques, the CLA achieves simplified circuit topologies and reduced transistor counts, leading to improved performance and efficiency in digital arithmetic circuits. Moreover, the CLA's hierarchical structure allows for modular and scalable design, making it suitable for applications requiring multi-bit addition with minimal delay. The simulation results further confirm the superior performance of the CLA, with lower average power consumption, maximum power consumption, time delay, and transistor count compared to traditional RCA implementations. Overall, the 4-bit CLA represents a versatile and efficient solution for implementing binary addition operations in digital systems, driving innovation and advancement in the field of digital circuit design.

References

[1]. Di Meo, Gennaro, Gerardo Saggese, Antonio GM Strollo, and Davide De Caro. "Approximate MAC unit using Static Segmentation." *IEEE Transactions on Emerging Topics in Computing* (2023).

[2]. Zhang, Xuan, Zhuoran Song, Xing Li, Zhezhi He, Li Jiang, Naifeng Jing, and Xiaoyao Liang. "HyAcc: A Hybrid CAM-MAC RRAM-based Accelerator for Recommendation Model." In *2023 IEEE 41st International Conference on Computer Design (ICCD)*, pp. 375-382. IEEE, 2023.

[3]. Kim, Eunhwan, Hyunmyung Oh, Nameun Kang, Jihoon Park, and Jae-Joon Kim. "A Capacitive Computing-In-Memory Circuit with Low Input Loading SRAM Bitcell and Adjustable ADC Input Range." *IEEE Transactions on Circuits and Systems II: Express Briefs* (2023).

[4]. Subin Ki, Juntae Park, and Hyun Kim. "Dedicated FPGA Implementation of the Gaussian TinyYOLOv3 Accelerator." *IEEE Transactions on Circuits and Systems II: Express Briefs* (2023).

[5]. Yao, Chun-Yen, Tsung-Yen Wu, Han-Chung Liang, Yu-Kai Chen, and Tsung-Te Liu. "A Fully Bit-Flexible Computation in Memory Macro Using Multi-Functional Computing Bit Cell and Embedded Input Sparsity Sensing." *IEEE Journal of Solid-State Circuits* (2023).

[6]. Shubham Kumar, Paul R. Genssler, Somaya Mansour, Yogesh Singh Chauhan, and Hussam Amrouch. "Frontiers in AI Acceleration: From Approximate Computing to FeFET Monolithic 3D Integration." In *2023 IFIP/IEEE 31st International Conference on Very Large-Scale Integration (VLSI-SoC)*, pp. 1-6. IEEE, 2023.

[7]. Cheon, Sungsoo, Kyeongho Lee, and Jongsun Park. "A 2941-TOPS/W Charge-Domain 10T SRAM Compute-in-Memory for Ternary Neural Network." *IEEE Transactions on Circuits and Systems I: Regular Papers* (2023).

[8]. Wang, Shuyu, and Hao Cai. "Computing-in-Memory with Enhanced STT-MRAM Readout Margin." *IEEE Transactions on Magnetics* (2023).

[9]. Jing, Naifeng, Zihan Zhang, Yongshuai Sun, Pengyu Liu, Liyan Chen, Qin Wang, and Jianfei Jiang. "Exploiting bit sparsity in both activation and weight in neural networks accelerators." *Integration* 88 (2023): 400-409.

[10]. Antolini, Alessio, Carmine Paolino, Francesco Zavalloni, Andrea Lico, Eleonora Franchi Scarselli, Mauro Mangia, Fabio Pareschi et al. "Combined HW/SW Drift and Variability Mitigation for PCM-based Analog In-memory Computing for Neural Network Applications." *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 13, no. 1 (2023): 395-407.

[11]. Noh, Seock-Hwan, Jahyun Koo, Seunghyun Lee, Jongse Park, and Jaeha Kung. "FlexBlock: A flexible DNN training accelerator with multi-mode block floating point support." *IEEE Transactions on Computers* (2023).

[12]. Kushwaha, Dinesh, Rajat Kohli, Jwalant Mishra, Rajiv V. Joshi, S. Dasgupta, and Anand Bulusu. "A Fully Differential 4-Bit Analog Compute-In-Memory Architecture for Inference Application." In *2023 IEEE 5th International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, pp. 1-5. IEEE, 2023.

[13]. Wang, Chia-Chun, Yun-Chen Lo, Jun-Shen Wu, Yu-Chih Tsai, Chia-Cheng Chang, Tsen-Wei Hsu, Min-Wei Chu, Chuan-Yao Lai, and Ren-Shuo Liu. "Exploiting and Enhancing Computation Latency Variability for High-Performance Time-Domain Computing-in-Memory Neural Network Accelerators." In *2023 IEEE 41st International Conference on Computer Design (ICCD)*, pp. 515-522. IEEE, 2023.

[14]. Laxman, Amgoth, N. Siva Sankara Reddy, and B. Rajendra Naik. "Design and implementation of hybrid logic-based MAC unit using 45 nm technology." *e-Prime-Advances in Electrical Engineering, Electronics and Energy* 6 (2023): 100317.

[15]. Vaithiyanathan, Dhandapani, Britto Pari James, and Karuthapandian Mariammal. "Comparative Study of Single MAC FIR Filter Architectures with Different Multiplication Techniques." In *2023 Second International Conference on Electrical, Electronics, Information and Communication Technologies (ICEEICT)*, pp. 01-10. IEEE, 2023.

[16]. Tang, Song-Nien. "Area-Efficient Parallel Multiplication Units for CNN Accelerators with Output Channel Parallelization." *IEEE Transactions on Very Large-Scale Integration (VLSI) Systems* 31, no. 3 (2023): 406-410

[17]. Mishra, Ravi Shankar, Puran Gour, Sandeep Dhariwal, Gaurav Kumar, and Anubhav Anand. "Design and Analysis of Low Power MAC for DSP Processor." In 2023 International Conference on Artificial Intelligence and Applications (ICAIA) Alliance Technology Conference (ATCON-1), pp. 1-3. IEEE, 2023.

[18]. Locatelli, Pedro Sartori, Dalton Martini Colombo, and Kamal El-Sankary. "Time-Domain Multiply–Accumulate Unit." *IEEE Transactions on Very Large-Scale Integration (VLSI) Systems* (2023).

[19]. Edavoor, Pranose J., Aneesh Raveendran, David Selvakumar, Vivian Desalphine, and Gopal Raut. "Design and Analysis of Posit Quire Processing Engine for Neural Network Applications." In *2023 36th International Conference on VLSI Design and 2023 22nd International Conference on Embedded Systems (VLSID)*, pp. 252-257. IEEE, 2023.

[20]. Parmar, Vivek, Franz Müller, Jing-Hua Hsuen, Sandeep Kaur Kingra, Yannick Raffel, Maximillian Lederer, Tarek Ali et al. "Demonstration of Differential Mode FeFET-Array for multi-precision storage and IMC applications." In *2023 International VLSI Symposium on Technology, Systems and Applications (VLSI-TSA/VLSI-DAT)*, pp. 1-2. IEEE, 2023.

[21]. Wang, Hechen, Renzhi Liu, Richard Dorrance, Deepak Dasalukunte, Dan Lake, and Brent Carlton. "A Charge Domain SRAM Compute-in-Memory Macro With C-2C Ladder-Based 8-Bit MAC Unit in 22-nm FinFET Process for Edge Inference." *IEEE Journal of Solid-State Circuits* 58, no. 4 (2023): 1037-1050.

[22]. Deny, J., M. Dinesh Ram, C. Satheesh, B. Suryakumar, and Omkar Rushikesh Mekala. "10 GB MAC Core Verification Monitor Module." In *2023 7th International Conference on Intelligent Computing and Control Systems (ICICCS)*, pp. 1722-1725. IEEE, 2023.

[23]. Xu, Xingyu, Qingwen Wei, Yang Zhang, Hao Cai, and Bo Liu. "Work-in-Process: Error-Compensation-Based Energy-Efficient MAC Unit for CNNs." In *Proceedings of the International Conference on Compilers, Architecture, and Synthesis for Embedded Systems*, pp. 3-4. 2023.

[24]. Ponraj, Jeyakumar, R. Jeyabharath, P. Veena, and Tharumar Srihari. "High-performance multiply-accumulate unit by integrating binary carry select adder and counter-based modular Wallace tree multiplier for embedding system." *Integration* 93 (2023): 102055.

[25]. Vinoth, R., and M. V. R. Kasyap. "Design and Implementation of High Speed 32-bit MAC Unit." In *Journal of Physics: Conference Series*, vol. 2571, no. 1, p. 012027. IOP Publishing, 2023.