Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 6, No.4, June : 2024

DESIGN AND COMPARISION OF MRCA WITH RIPPLE CARRY ADDER BY USING 7-NANOMETER FINFET TECHNOLOGY

D. Rebecca Florance, Research Scholar, Department of Electronics and Communication Engineering, JNTUH, University College of Engineering, Jagtial, Telangana, India.
Dr. Prabhakar, Professor of Electronics and Communication Engineering and Principal, JNTUH, University College of Engineering, Jagtial, Telangana, India.

ABSTRACT
The design and simulation of a 4-bit ripple carry adder (RCA) and a modified ripple carry adder (MRCA) utilizing 10 FinFET and 8 FinFET transistor-based full adders respectively, fabricated using 7 nm technology. Implemented in Microwind software-3.9.4, both adder architectures are analysed for performance metrics such as delay, power dissipation, and efficiency. The RCAs are constructed with cascaded full adders, where the carry output from each adder is connected to the carry input of the next. Simulation results indicate that the traditional RCA design exhibits a rise delay of 2.24 ps and power dissipation of 1.064 mW at 0.75 volts. In contrast, the MRCA design, optimized for reduced transistor count, achieves a rise delay of 1.21 ps and power dissipation of 0.184 mW at the same voltage. These findings underscore the advantages of using FinFET 7 nm technology for digital arithmetic circuits, demonstrating significant improvements in speed and energy efficiency, which are crucial for advancing microprocessor performed better than conventional methods in terms of area, power, delay metrics.
Keywords: Finfield-effect transistor, Ripple carry adder (RCA), Modified ripple carry adder (MRCA), Area, Delay, Rise delay, Power dissipation, Technology.

I. Introduction
The ripple carry adder is a fundamental digital circuit used extensively for performing the arithmetic addition of binary numbers. It operates by cascading multiple full adders, where each full adder computes a partial sum and propagates a carry to the subsequent adder. This simple yet crucial architecture forms the basis for more complex adders and is widely employed in digital systems due to its straightforward implementation [1]. The operation of an RCA is characterized by the propagation of carry signals through each stage of the adder. In an n-bit RCA, the carry must ripple through all n full adders, resulting in a delay that scales linearly with the number of bits. This inherent delay can impose significant constraints on the speed of microprocessors and other high-performance digital systems, particularly in applications demanding rapid arithmetic computations[2].With the ongoing advancements in semiconductor technology, the exploration and enhancement of adder designs using innovative transistor structures have garnered considerable attention. Notably, FinFETs (Finfield-Effect Transistors) have emerged as a promising alternative to conventional CMOS technology. FinFET's offer superior control over short-channel effects and reduced leakage currents, making them well-suited for achieving high-speed and energy-efficient digital circuits [3].
The traditional RCA design employs 10 FinFET transistors per full adder, the study provides comprehensive analyses of delay characteristics, power dissipation, and overall performance metrics for the architectures. The equations governing the operation of the RCA emphasizing the critical path delays, which are pivotal in assessing the efficiency of these adder designs. Specifically, the rise and fall delays from A1 to Vdd, along with power dissipation metrics, are examined to underscore the advantages offered by FinFET technology in enhancing the operational speed and energy efficiency of RCA's. The findings presented in this study highlight significant improvements achieved through the adoption of FinFET's, particularly showcased by reduced rise delays and lower power dissipation compared to conventional CMOS-based RCA's. These results underscore the potential of advanced transistor technologies in advancing the capabilities of digital arithmetic circuits, thereby contributing to the evolution of modern microprocessor designs and other high-speed digital applications [4]. Full

adders represent pivotal computational units within computer architecture and find ubiquitous application in digital signal processors. In numerous computational domains including computer graphics, scientific computing, and image processing, the demand for high-speed full adders is steadily escalating [5]. Designers are now emphasizing rapid speed and low power usage as they recognize the substantial impact of full adder speed on CPU performance. Full adders, serving as the beating core of arithmetic circuits, including numerous complex ones, are pivotal in meeting these design objectives [6].

II. Literature

The literature review provides a thorough examination of the progress in semiconductor technology Sicard, [7] with a particular emphasis on advancements in transistor design, power efficiency, and integration density across different technology nodes. The significant contribution of FinFET technology at 7 nm nodes is underscored, showcasing its superiority over conventional planar transistors. this section critically evaluates various conventional designs for full adders along with their respective limitations. Nikoubin et al. [8] concentrated on a full adder employing 32-nm FinFET technology, comprising 14 transistors. Additionally, they expanded their investigation to fabricate a 4-bit RCA adder leveraging the developed full adder design. Vallabhuni et al. [9] introduced utilizing 32-nm FinFET technology, devised an 8-bit Manchester carry chain adder, encountering obstacles related to constrained energy resources and an increased transistor count. Liu et al. [10] proposed a Memristor Logic (MRL)-based carry look-ahead adder, integrating hybrid-CMOS and MRL universal gate structures. nonetheless, this approach faced challenges related to higher power consumption. Hasan et al. [11] developed a full adder using hybrid pass transistor logic (HPTL) was developed with 22 transistors, incorporating XOR-XNOR and AND-OR selection logic. Implemented in generic process design kit (GPDK) 45 nm technology, the corresponding 4-bit ripple carry adder demonstrated notable path delay improvements. Malik et al. [12] explored full adder designs based on hybrid methodologies, combining CMOS and gate diffusion input, past transistor logic. However, this approach exhibited significant power consumption issues. Tyagi et al. [13] developed a full adder implemented using 10 transistors with CNTFET technology, although it encountered challenges related to increased delays and power consumption. Tyagi et al. [14] developed a full adder based on gate diffusion input technology was developed using eight transistors. Despite the reduced transistor count, accuracy-related issues were observed, resulting in decreased system precision. Srinivasu et al. [15] devised a full adder utilizing 2-to-1 multiplexers with carbon nanotube field effect transistor (CNTFET) technology, employing twelve transistors. Additionally, they introduced a hybrid adder that integrates both 4-bit ripple carry adders and parallel prefix adders. Ramkumar et al. [16] implemented one-bit full adders and using FinFET and gate diffusion input technology. Although the incorporation of multiplexer logic enabled parallel addition operations, this method exhibited increased noise levels. Mamaghani et al. [17] developed a hybrid full adder was proposed, integrating magnetic tunnel junctions and FinFET technology, utilizing 30 transistors. which are the traditional methods.

III. Block diagram of four-bit ripple carry adder

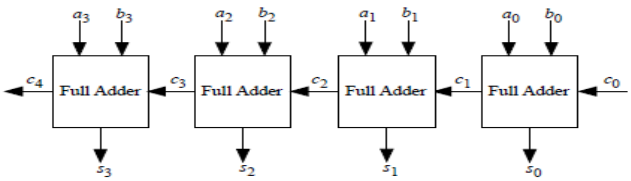


Figure 1(a): Block diagram of four-bit ripple carry adder

In Fig. 1(a) a four-bit ripple carry adder is a fundamental digital circuit designed to add two four-bit binary numbers. It consists of four full adders connected in series, with each adder processing inputs

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 6, No.4, June : 2024

from the corresponding bits of the two numbers A and B along with a carry-in, and producing a sum bit (S) and a carry-out (Cout). The carry-out from each full adder serves as the carry-in for the subsequent adder. starting with an initial carry-in of 0. This setup enables the adder to compute the sum of the least significant bits first and propagate the carry to the more significant bits sequentially. While this method is straightforward, it introduces a propagation delay as each carry-out must ripple through subsequent stages, impacting performance. Despite this limitation, ripple carry adders are widely used in arithmetic logic units (ALUs) within CPUs and microcontrollers for binary addition tasks. For improved performance, more advanced adders like carry-lookahead adders are employed to mitigate the ripple effect and reduce delay.

Equations Governing the 4-bit RCA

FA0 (Least Significant Bit): $(S0) = A0 \oplus B0 \oplus Cin$, $(C0) = (A0B0) + (Cin\ (A0 \oplus B0))$

FA1: Sum1 $(S1) = A1 \oplus B1 \oplus C0$, Carry1 $(C1) = (A1B1) + (C0(A1 \oplus B1))$

FA2: Sum2 $(S2) = A2 \oplus B2 \oplus C1$, Carry2 $(C2) = (A2B2) + (C1(A2 \oplus B2))$

FA3 (Most Significant Bit): Sum3 $(S3) = A3 \oplus B3 \oplus C2$, Carry3 (Cout) $= (A3B3) + (C2(A3 \oplus B3))$

The final output of the 4-bit RCA will be the four sum bits S0 to S3 and the final carry out Cout.

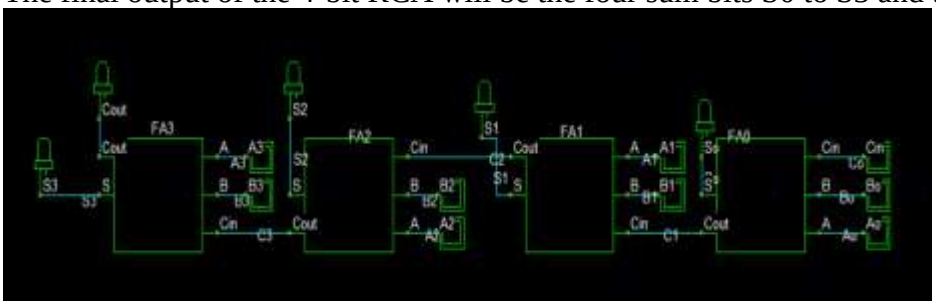


Figure 1(b): Schematic diagram of 4-bit ripple carry adder

In Fig. 1(b) a 4-bit ripple carry adder is a digital circuit designed to perform the binary addition of two 4-bit numbers. It comprises four full adders FA1 to FA4 connected sequentially, each adding corresponding bits from two input binary numbers A0 to A3 and B0 to B3 and a carry-in Cin. The carry-out Cout from each full adder feeds into the next, ensuring proper addition propagation through the chain. Each full adder generates a sum output S0 to S3 and a carry-out, with the final carry-out representing the overall carry of the addition. The RCA inherently experiences propagation delay as each stage must wait for the carry from the previous stage, affecting the timing of the final result. Despite this delay, the RCA is fundamental in constructing arithmetic units within processors and other digital systems, providing the basic mechanism for binary addition operations. Understanding the internal logic gates and propagation delays within each full adder is crucial for grasping the RCA's functionality and performance.

3.1 4-Bit ripple carry adder operation is analysed by taking truth table as reference

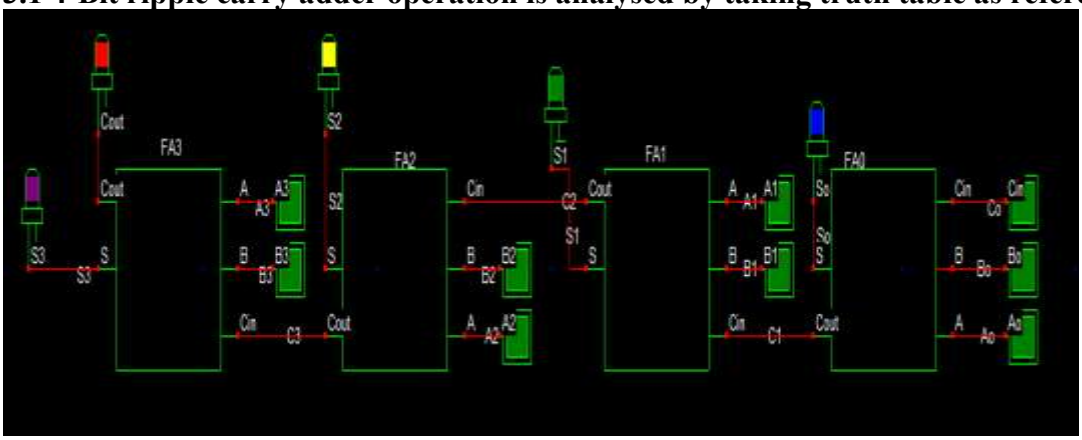
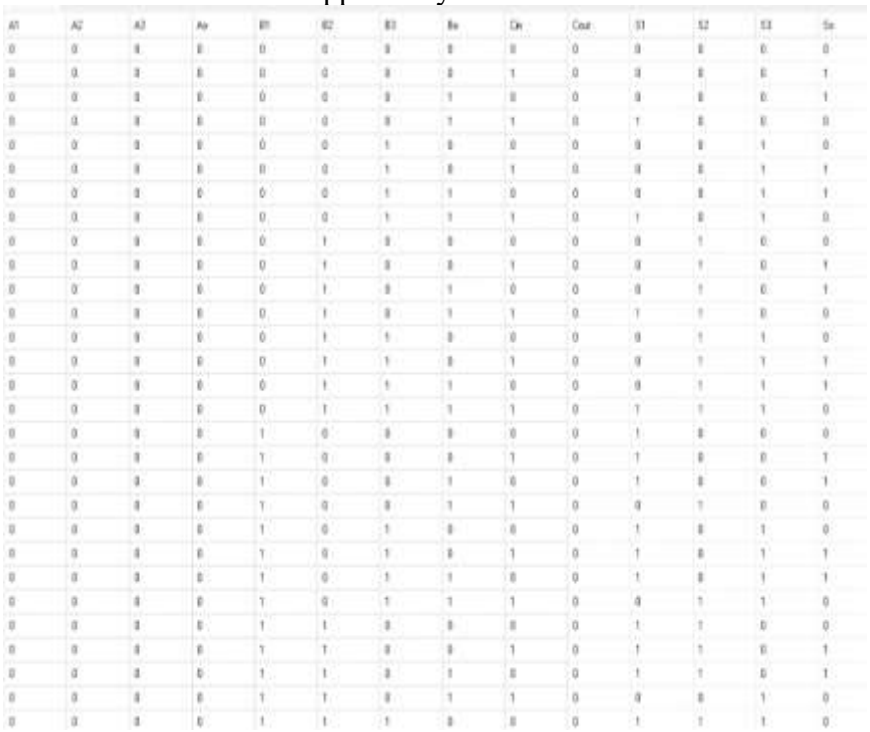


Figure1(c): 4-Bit ripple carry adder with all inputs as 1

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 6, No.4, June : 2024

Table 1: Operational truth table of 4-Bit ripple carry adder



In Fig. 1(c) a 4-bit ripple carry adder is a digital circuit designed for performing binary addition of two 4-bit numbers. composed of four sequentially connected full adders (FA0 to FA3). Each full adder processes two input bits (A0 to A3 and B0 to B3) and a carry-in (Cin), generating a sum output (S0 to S3) and a carry-out (Cout). The carry-out from one full adder serves as the carry-in for the subsequent adder. with the first full adder receiving an external carry-in labelled as C0, when all inputs are one output is also one. Internally, each full adder uses XOR gates for sum calculation and AND/OR gates for carry propagation. Due to propagation delays within the logic circuitry, the RCA's final sum and carry-out are valid only after the cumulative delay of all stages. Despite its simplicity and inherent delay, the RCA is foundational for building arithmetic units in processors and digital systems, performing essential binary addition operations.

3.2 Ripple carry adder delay

In a ripple carry adder, the output for each bit is determined only after the carry from the preceding stage has been generated and propagated. Consequently, the sum of the most significant bit (MSB) becomes available only after the carry signal has rippled through all preceding stages, leading to a significant delay. For an n-bit RCA, the delay for obtaining the sum of the MSB is calculated as $4n$+2-time units, while the carry delay is $4n$+3-time units. For a 4-bit RCA, this translates to a sum delay of 18 nanoseconds and a carry delay of 19 nanoseconds. As the number of bits increases, the RCA becomes progressively slower, with the carry chain propagation delay being a critical factor influencing overall microprocessor speeds. This inherent delay underscores the limitations of RCAs in high-speed digital systems and highlights the need for more efficient adder architectures, such as carry-lookahead adders, to enhance performance.

3.3 Critical path delay

The critical path delay in a 4-bit ripple carry adder is established by the duration needed for the carry signal to propagate through all the full adders within the circuit. The worst-case scenario arises when a carry bit must traverse from the least significant bit to the most significant bit, given that each full adder relies on the carry-in from its predecessor.

The critical path delay is calculated by summing the carry propagation delays of each full adder and adding the sum propagation delay of the last full adder. For a 4-bit RCA, this delay is given by the formula critical path delay = 3 x (Carry Propagation Delay) + Sum Propagation Delay. This calculation

assumes uniform carry propagation delay across all full adders and accounts for the sum delay only at the final stage, reflecting the sequential nature of the carry propagation process. Consequently, this delay underscores the speed limitations of RCAs, as the propagation of the carry signal through numerous stages introduces notable latency.

3.4 Backend layout of 4-bit ripple carry adder using 7-nm technology

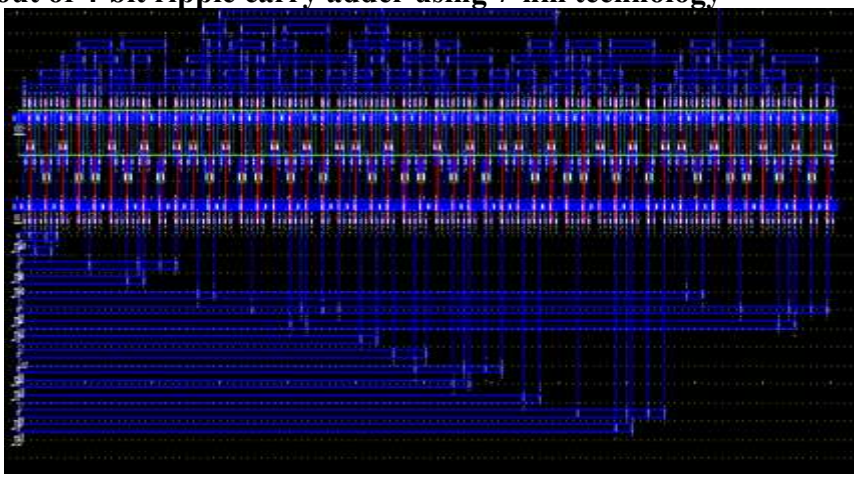


Figure1(d): Backend layout of 4-bit ripple carry adder using 7-nm technology

In Fig.1(d) the ripple carry adder is a fundamental digital circuit used to perform binary addition by chaining together multiple full adders. Each full adder processes three inputs two binary digits (A and B) and a Carry-in (Cin), producing a Sum (S) and a Carry-out (Cout). Within the RCA architecture, the carry-out from one full adder seamlessly transitions to become the carry-in for the subsequent higher bit's adder, thereby instigating a ripple effect coursing through the chain from the least significant bit to the most significant bit. The layout of an RCA typically features rows and columns representing bit positions and stages of the adder, with interconnections illustrating the carry propagation. Different colours in the layout might indicate various connection types, such as metal traces, polysilicon, or diffusion layers. The entire circuit operates with a supply voltage (Vdd) and ground (GND), essential for the transistors' functionality. Transistor channel width and length influence speed and power characteristics, while the RCA embodies the equations for sum (Si = Ai ⊕ Bi ⊕ Ci-1) and carry-out, fundamental to its operation.

4.0 Modified ripple carry adder design

A modified ripple carry adder is a digital circuit designed to calculate the arithmetic sum of two binary numbers by employing modified full adders interconnected in a cascade configuration. Each full adder's carry output is linked to the carry input of the next adder in the chain, facilitating the sequential propagation of carry bits. the interconnection of four modified full adders, forming a 4-bit MRCA, with inputs entering from the right to signify the least significant bit. This 4-bit MRCA, designed using 7-nm technology and 8 FinFET transistors per full adder, is implemented in Microwind software-3.9.4, highlighting advanced semiconductor technology for enhanced performance and reduced power consumption in digital circuits.

4.1 Block diagram 4-bit ripple carry adder

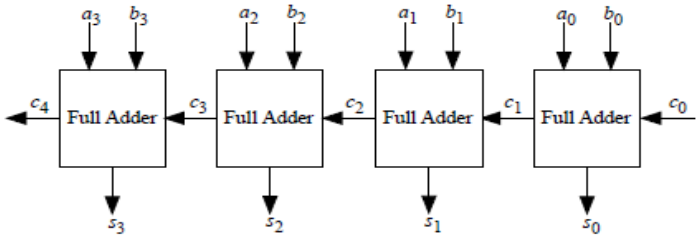


Figure 2(a): Block diagram of 4-bit ripple carry adder

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 6, No.4, June : 2024

In Fig. 2(a) the 4-bit ripple carry adder block diagram showcases a digital circuit capable of adding two 4-bit binary numbers, with inputs labelled A0 to A3 and B0 to B3. Comprising four sequentially connected full adders, each full adder takes three inputs: a bit from each number A and B and Cin, and produces a sum and carry-out. The rightmost full adder, handling the least significant bit, starts with an initial carry-in of 0. The carry-out from each full adder ripples to the next adder's carry-in, creating a sequential propagation of carry bits through all stages. This ripple effect results in a slight propagation delay, as each stage must wait for the previous carry to stabilize. The sum outputs (S0 to S3) and final carry-out provide the complete binary addition result, making the 4-bit RCA a fundamental component for arithmetic operations in digital circuits.

Equations Governing the 4-bit RCA.

FA0 (Least Significant Bit): (S0) = A0 ⊕ B0 ⊕ Cin, (C0) = (A0B0) + (Cin (A0 ⊕ B0))

FA1: Sum1 (S1) = A1 ⊕ B1 ⊕ C0, Carry1 (C1) = (A1B1) + (C0(A1 ⊕ B1))

FA2: Sum2 (S2) = A2 ⊕ B2 ⊕ C1, Carry2 (C2) = (A2B2) + (C1(A2 ⊕ B2))

FA3 (Most Significant Bit): Sum3 (S3) = A3 ⊕ B3 ⊕ C2, Carry3 (Cout) = (A3B3) + (C2(A3 ⊕ B3)).

The final output of the 4-bit MRCA will be the four sum bits (S0 to S3) and the final carry out (Cout).

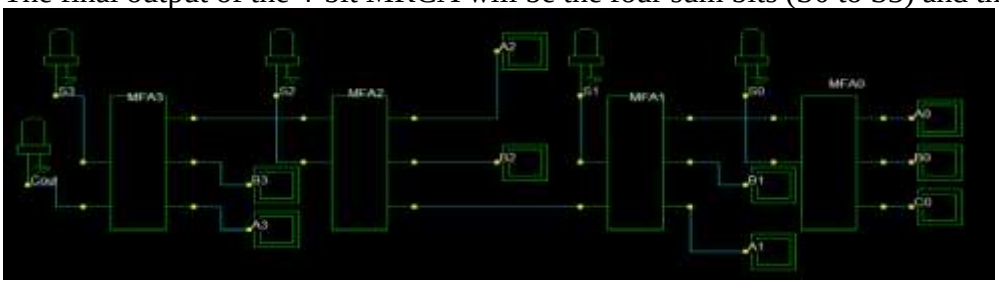


Figure 2(b): Schematic diagram of modified ripple carry adder design

In Fig. 2(b) the schematic diagram of modified ripple carry adder block illustrates a circuit designed to add two 4-bit binary numbers, with inputs labelled A0 to A3 and B0 to B3. The circuit consists of four full adders connected sequentially from right to left. Each full adder receives three inputs: a bit from each number A and B and a carry-in, and it produces a sum and a carry-out. The rightmost full adder, handling the least significant bit , starts with an initial carry-in of 0. The carry out from each adder ripples into the carry in of the next, ensuring proper propagation through all stages. The sum outputs (S0 to S3) represent the calculated sum for each bit position. Although specific modification details are not provided in the diagram, such modifications typically aim at enhancing performance, reducing power consumption, or other optimizations. This ripple carry configuration efficiently propagates the carry through each stage, ensuring accurate addition of the binary numbers.

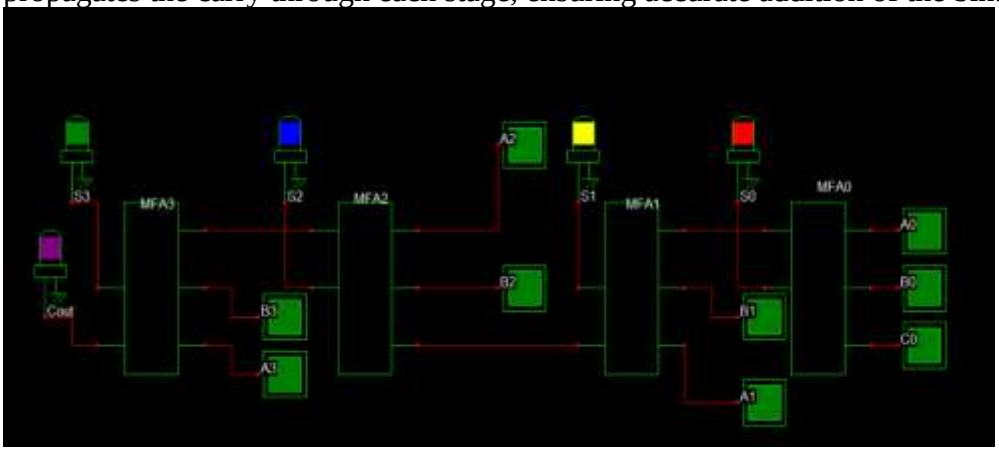


Figure 2(c): Modified 4-bit ripple carry adder with all inputs as one

In Fig. 2(c) the modified ripple carry adder diagram details a digital circuit designed to add two 4-bit binary numbers, with inputs explicitly labelled as A0 to A3 and B0 to B3 for the individual bits of the two numbers, and C_0 for the initial carry-in, typically set to 0. The circuit consists of four modified full adders (MFA0 to MFA3), each representing a 1-bit full adder. These adders are connected

sequentially from right to left, starting with MFA0, which receives inputs A_0, B_0, and C_0. Each full adder produces a sum (S) and carry-out (Cout), with the carry out from each adder propagating as the carry in to the next adder. The sum outputs (S0 to S3) correspond to the sum bits of each respective bit position, calculated using the formula $Si=Ai \oplus Bi \oplus Ci-1$Si=Ai⊕Bi⊕Ci−1. The leftmost full adder (MFA3) produces the final carry-out, which may be used for further stages if needed. the configuration maintains the fundamental ripple carry approach, ensuring accurate binary addition through sequential carry propagation and gives all the outputs as 1.

4.2 Modified ripple carry adder delay

In a modified ripple carry adder, the output depends on the carry generated by the previous stages, leading to significant delays in obtaining the final sum and carry bits. Specifically, for an n-bit modified RCA, the delay for the sum of the most significant bit (Sn-1) is 4n+2, and for the carry (Cn) is 4n+3. For a 4-bit modified RCA, this translates to a sum delay of 18 nanoseconds and a carry delay of 19 nanoseconds. These delays occur because the carry signal must ripple through all stages, from the least significant bit to the most significant bit, before the final outputs are valid. This ripple effect results in substantial delays, making the RCA relatively slow, particularly as the number of bits increases. Consequently, the carry chain propagation delay is a critical factor influencing the speed of microprocessors, often becoming the limiting factor in their performance.

4.3 Critical Path Delay

The critical path delay in a modified 4-bit ripple carry adder is a pivotal factor in assessing its operational efficiency. This delay is primarily influenced by the cumulative propagation of the carry signal through each stage of the adder. In the MRCA architecture, each modified full adder must await the arrival of the carry-in signal from the preceding stage before executing its own summation and generating a carry-out. The most severe delay scenario, known as the critical path delay, arises when a carry bit traverses through all full adders from the least significant bit to the most significant bit. This dependency, where the carry-out of one adder becomes the carry-in of the next, underscores the inherent latency in the system. The critical path delay is computed as the summation of the carry propagation delays across all full adders, augmented by the summation propagation delay of the last full adder. For a modified 4-bit RCA, a concise formula, 3× (Carry Propagation Delay) + Sum Propagation Delay, encapsulates the critical path delay calculation. This formulation assumes uniform carry propagation delay across all adders while incorporating the final sum propagation delay once, reflecting the time required for determining the final sum of the Most Significant Bit.

4.4 Backend layout of modified ripple carry adder using 7-nm technology

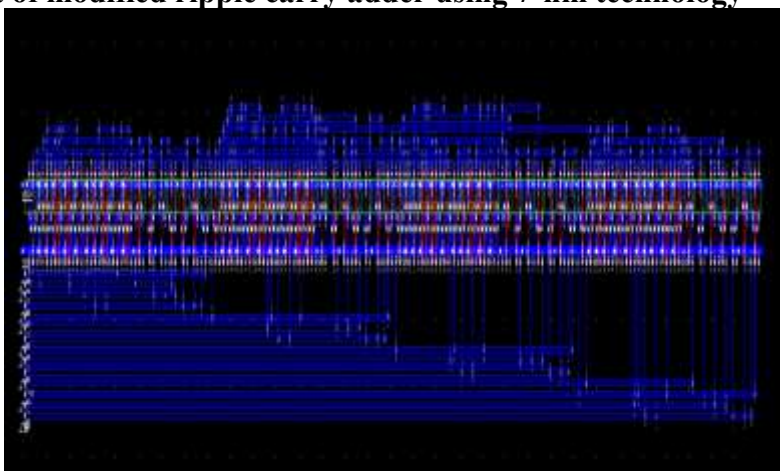


Figure 2(d): Backend layout of modified ripple carry adder using 7-nm technology

In Fig. 2(d) the modified ripple carry adder layout design optimized for adding binary numbers. It builds on the traditional ripple carry adder by enhancing performance and reducing propagation delays. The layout features multiple interconnected blocks, each representing a 1-bit full adder, arranged sequentially to process individual bits of the 4-bit binary numbers labelled A0 to A3 and B0 to B3. The

initial carry-in, usually set to 0, enters the rightmost full adder, and the carry-out propagates leftward through each subsequent stage. Sum outputs from each adder are connected to form the final result, while the leftmost full adder provides the final carry-out. This design ensures efficient carry propagation and accurate sum calculation, making it an essential component in arithmetic-logic units (ALUs) and various computational tasks within microprocessors.

IV. Results

A comprehensive simulation performance study of suggested adders is presented utilizing a wide variety of parameters. In addition, to perform the proposed MRCA is compared with the performance of traditional adders using a variety of parameters. The Microwind 3. 9.4 software tool is used in the process of designing and simulating the different designs.

4.1 Timing diagram of 4-bit ripple carry adder

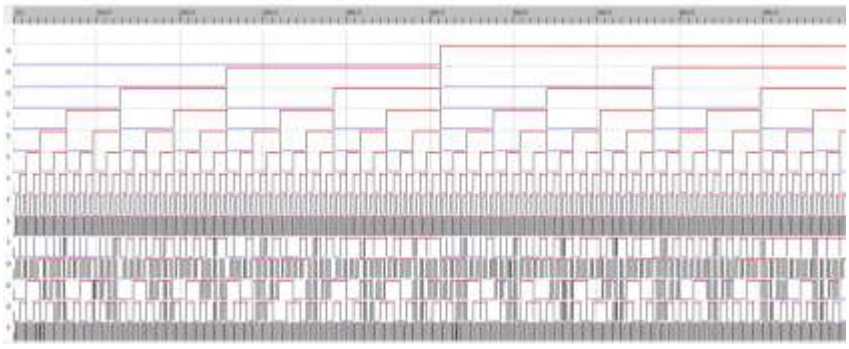


Figure 3(a): Timing diagram of 4-bit ripple carry adder

In Fig.3(a) the timing diagram of a 4-bit ripple carry adder illustrates the sequential propagation of sum and carry signals through each adder stage. In a typical operation, the RCA adds two 4-bit binary numbers, with inputs labelled A0 to A3 and B0 to B3. The carry-out (Cout) from each stage becomes the carry-in (Cin) for the next stage, creating a ripple effect. For instance, given inputs A = 0101 and B = 1010, the sum outputs (S0 to S3) and the final carry-out (Cout) are calculated sequentially: S0=1, S1=1, S2=0, S3=1, and Cout=1. This step-by-step calculation highlights the inherent delay due to carry propagation, as each stage must wait for the previous carry to be resolved before producing a valid output. Consequently, the final result is only available after all carry signals have propagated through the adder stages, underscoring the impact of propagation delay in RCA's.

4.2 Timing diagram of 4-Bit Ripple carry adder layout

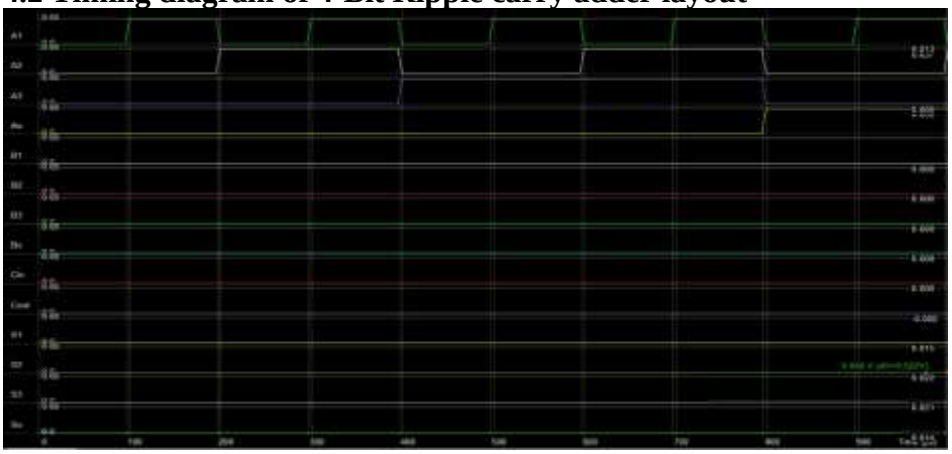
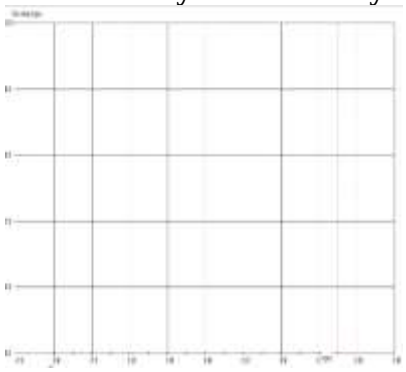


Figure 3(b): Timing diagram of 4-Bit ripple carry adder layout

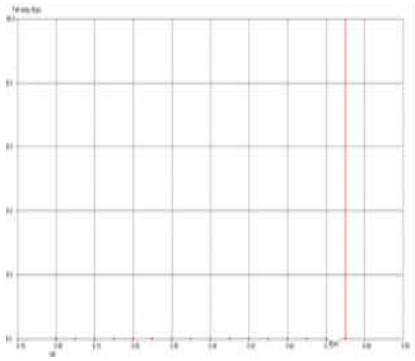
In Fig.3(b) the timing diagram for a ripple carry adder (RCA) illustrates the sequential processing of binary inputs (A0 to A3 and B0 to B3) and the propagation of carry bits through the adder. Starting with the least significant bit, the sum outputs (S0 to S3) and carry-out signals (Cout) show a slight delay due to gate propagation. Each sum output (Si) depends on the inputs (Ai and Bi) and the carry-in from the previous stage (Cin), while each carry-out (Ci) influences the subsequent stage. The

diagram as shown in Fig.3(b) highlights the ripple effect, where the carry-out from one bit position becomes the carry-in for the next, causing a cumulative delay that is critical to understanding the overall timing constraints and performance of the adder. This visual representation is essential for designing efficient digital systems, as it underscores the importance of managing propagation delays to ensure accurate and timely calculations.

4.3 Rise delay and fall delay from a1 to vdd



Graph: 1(a): Rise delay

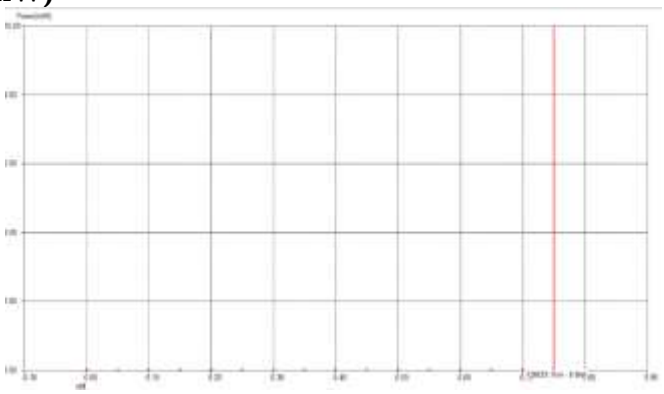


Graph: 1(b): Fall delay

In graph 1(a) rise delay in digital circuits denotes the duration required for a signal to transition from a low voltage level (A1) to a high voltage level (Vdd) during its rising edge. This process involves charging the capacitive load of the signal path, which is influenced by factors such as transistor characteristics, load capacitance, supply voltage (Vdd), and threshold voltage (Vth). The provided graph, with the x-axis representing time and the y-axis representing rise delay, illustrates the progression of this transition. The shaded red region between approximately 7.8 ns and 8.2 ns highlights a specific range of interest, indicating significant observations or variations in the rise delay within this timeframe. As the signal rises towards vdd, the graph visualizes how the rise delay evolves over time, emphasizing critical moments in the signal's transition.

In graph1(b) fall delay in digital circuits characterizes the duration necessary for a signal to transition from a high voltage level (A1) to a low voltage level (typically 0V) during its falling edge. This process involves discharging the capacitive load of the signal path, influenced by factors such as transistor characteristics, load capacitance, supply voltage (Vdd), and threshold voltage (Vth). The provided graph, with the x-axis representing time and the y-axis representing fall delay, highlights the critical moment of this transition. The vertical red line at approximately 0.8 ns indicates the precise point where the fall delay is observed, marking the key moment of the signal's transition from high to low. The rest of the graph remains empty, underscoring the significance of this specific delay in the overall signal propagation.

4.4 Power dissipation (nW)



Graph: 1(c) Power dissipation (nW)

The graph 1(c) of power dissipation in nanowatts (nW) versus supply voltage (vdd) illustrates the critical relationship between voltage and energy loss in a CMOS device. As the supply voltage increases, power dissipation rises sharply, reflecting higher energy loss as heat. The red vertical band

highlights a specific vdd range of interest, while the annotation cmos tnm = 80 mV indicates the threshold noise margin, where the circuit transitions between logic states. This graph underscores the importance of balancing performance and power consumption in circuit design lower vdd reduces power dissipation but may affect speed, while higher vdd enhances performance but increases power loss. Managing this trade-off is essential in the development of efficient, high-performance electronic devices.

5.0 Timing diagram of modified 4-bit ripple carry adder

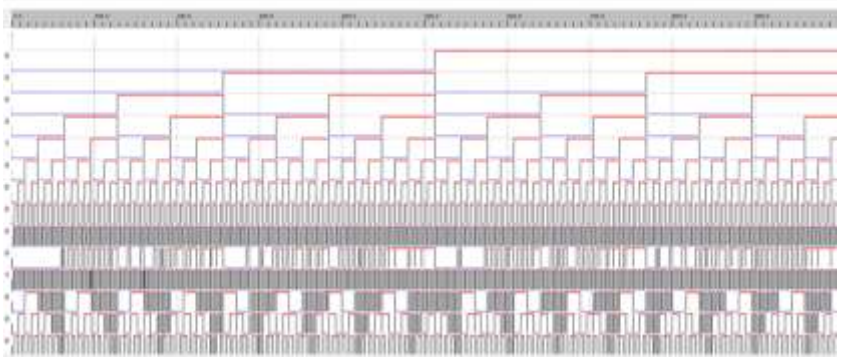


Figure 4(a): Timing diagram of modified 4-bit ripple carry adder

In Fig.4(a) the timing diagram for a modified ripple carry adder visually represents the dynamic behaviour of signals within the circuit as it processes two 4-bit binary numbers. It includes inputs labelled A0 to A3 and B0 to B3, which change over time as the binary numbers are added. The initial carry-in (Cin), usually set to 0, impacts the first stage of addition. The diagram illustrates how each full adder (MFA0 to MFA3) processes its respective bits and how the carry propagates sequentially from one stage to the next. Sum outputs (S0 to S3) and the final carry-out (Cout) are also depicted, showing their changes in response to evolving inputs and carry values. This visual representation helps engineers understand propagation delays, synchronization, and overall performance of the modified RCA, ensuring accurate and efficient binary addition.

5.1 Timing diagram of modified ripple carry adder layout

Figure 4(b): Layout timing diagram of modified ripple carry adder

In Fig.4(b) a layout timing diagram for a modified ripple carry adder visually depicts how signals within the circuit evolve over time as binary numbers are processed. Key components include the inputs (A0 to A3 and B0 to B3), representing bits of two 4-bit binary numbers. These inputs change over time, reflecting the data being processed. The initial carry-in (Cin), typically set to 0, is also shown, affecting the first adder stage. The diagram includes each modified full adder (MFA0 to MFA3), illustrating how the carry signal propagates sequentially from one stage to the next. Sum outputs (S0 to S3) and the final carry-out (Cout) are displayed, showing how they react to changes in inputs and intermediate carry values. This detailed representation helps engineers analyse the circuit's

performance, including propagation delays and synchronization, ensuring efficient and accurate binary addition.

5.2 Comparison between RCA and MRCA

RCA	**MRCA**
Rise delay at 0.75 volts = 2.24PS	Rise delay at 0.75 volts = 1.21 PS
Power dissipation at 0.75 volts = 1.064 mW	Power dissipation at 0.75 volts = 0.184 mW

V. Conclusion

In conclusion, our comprehensive review of full adder designs underscores the diverse implementations and inherent trade-offs essential for optimizing digital arithmetic circuits. Technologies such as FinFET, CNTFET, and hybrid CMOS offer notable advantages in transistor count, power consumption, and propagation delay, yet they also pose unique challenges. The 4-bit Ripple Carry Adder, fundamental to digital systems for its simplicity, is hindered by significant propagation delays, particularly impacting high-speed applications. Advanced implementations like the 7 nm FinFET-based Modified Ripple Carry Adder demonstrate enhanced performance, with reduced rise delay (1.21 ps) and lower power dissipation (0.184 mW) compared to RCA (2.24 ps rise delay, 1.064 mW power dissipation), yet still face challenges with carry propagation delays. Detailed timing diagrams and power dissipation analyses provide critical insights into the dynamic behaviour and efficiency limitations of these designs. To meet the increasing demands for speed and efficiency in modern computing, future research should prioritize the development of innovative adder architectures, which have the potential to overcome delay issues and drive significant advancements in microprocessor technology.

References

- [1] Morris Mano, M.; Kime, C.R. Logic and Computer Design Fundamentals; Pearson Education, 2015.
- [2] Weste, N.H.E.; Harris, D.M. CMOS VLSI Design: A Circuits and Systems Perspective; Pearson Education, 2011.
- [3] Yu, B., et al. A 22nm high performance and low-power CMOS technology featuring fully depleted tri-gate transistors. In Proceedings of the IEEE International Electron Devices Meeting, 2012, pp. 3-1.
- [4] Kang, S.M.; Leblebici, Y. CMOS Digital Integrated Circuits: Analysis and Design; McGraw-Hill, 2003.
- [5] Vallabhuni, R.R., et al. Comparative Analysis of 8-Bit Manchester Carry Chain Adder Using FinFET at 18nm Technology. In Proceedings of the 2020 3rd International Conference on Intelligent Sustainable Systems (ICISS); IEEE, 2020.
- [6] Vendhan, M.A. Analysis on Circuit Metrics of 1-Bit FinFET Adders Realized using Distinct Logic Structures. Indian J. Sci. Technol. 2019, 12, 1–4.
- [7] Sicard, E. Advances in Semiconductor Technology; IEEE, 2017.
- [8] Nikoubin, T. Hybrid logical effort for hybrid logic style full adders in multistage structures.
- IEEE Trans. Very Large-Scale Integr. (VLSI) Syst. 2019, 27, 1138–1147.
- [9] Vallabhuni, R.R., et al. Comparative Analysis of 8-Bit Manchester Carry Chain Adder Using FinFET at 18nm Technology. In Proceedings of the 2020 3rd International Conference on Intelligent Sustainable Systems (ICISS); IEEE, 2020.
- [10] Liu, G., et al. A carry lookahead adder based on hybrid CMOS-memristor logic circuit. IEEE Access 2019, 7, 43691–43696.
- [11] Hasan, M., et al. Design of a scalable low-power 1-bit hybrid full adder for fast computation. IEEE Trans. Circuits Syst. II Express Briefs 2019, 67, 1464–1468.
- [12] Malik, G.F.A., et al. Modelling for triple gate spin-FET and design of triple gate spin FET-based binary adder. IET Circuits Devices Syst. 2020, 14, 464–470.

- [13] Tyagi, P.; Singh, S.K.; Dua, P. Gate Diffusion Input Based 10-T CNTFET Power Efficient
- Full Adder. Recent Adv. Electr. Electron. Eng. 2021, 14, 415–427.
- [14] Tyagi, P.; Singh, S.K.; Dua, P. Ultra-Low Power 8-Transistor Modified Gate Diffusion
- Input Carbon Nano-Tube Field Effect Transistor Full Adder. IETE J. Res. 2021, 1–14.
- [15] Srinivasu, B.; Sridharan, K. A transistor-level probabilistic approach for reliability analysis of arithmetic circuits with applications to emerging technologies. IEEE Trans. Reliab. 2017, 66, 440–457.
- [16] Ramkumar, E., et al. Design and Analysis of Low Power and High Speed FinFET based Hybrid Full Adder/Subtractor Circuit (FHAS). In Proceedings of the 2020 IEEE International Symposium on Smart Electronic Systems (iSES) (Formerly iNiS); IEEE, 2020.
- [17] Mamaghani, S.B.; Moaiyeri, M.H.; Jaberipur, G. Design of an efficient fully nonvolatile and radiation-hardened majority-based magnetic full adder using FinFET/MTJ. Microelectron. J. 2020, 103, 104864.