International Journal of Nanoelectronics and Materials
IJNeaM
ISSN 198-5761 | E-ISSN 2232-1535

High Efficiency Carry Save Adder using Modified–gate Diffusion Input Technique

Teoh Yong Keong[a], Siti Fatimah Abd Rahman[a,*], Mohamad Faris Mohamad Fathil[b,*], Mohamed Fauzi Packeer Mohamed[a,*], Adilah Ayoib[b,c], and Thikra S. Dhahi[d]

[a]School of Electrical and Electronic Engineering, Engineering Campus, Universiti Sains Malaysia, 14300 Nibong Tebal, Pulau Pinang, Malaysia.
[b]Institute of Nano Electronic Engineering, Universiti Malaysia Perlis, 01000 Kangar, Perlis, Malaysia.
[c]Faculty of Chemical Engineering & Technology, Universiti Malaysia Perlis, 02100 Padang Besar, Perlis, Malaysia.
[d]Electronics Technical Department, Southern Technical University, Basra, Iraq.
**Corresponding author. E-mail: fatimahrahman@usm.my; mohamadfaris@unimap.edu.my; fauzi.packeer@usm.my*

ABSTRACT

Addition is a fundamental function in the design of a digital system, necessary for applications such as signal processing, arithmetic operations, multiplexers, and control systems. Hence, the digital system's performance is considerably reliant on the efficiency of the adders. Therefore, designing a 4-bit carry save adder (CSA) that consumes less power, occupies a smaller area, and operates at a higher speed is proposed using the modified–gate diffusion input (MOD–GDI) technique. The primary focus is to reduce the area occupied by decreasing the transistor count as compared with other logic styles (i.e., conventional, and Boolean simplification) for CSA through Cadence Virtuoso simulation based on SilTerra 180 nm technology. Notably, the number of transistors is reduced from 42 in the conventional full adder to 11 in the proposed MOD–GDI design. As a result, the proposed 4-bit CSA with MOD–GDI technique is efficient in improving the speed of addition by reducing the area and power consumption.

Keywords: *Cadence, Carry save adder, Full adder, Modified-gate diffusion input, Performance*

1. INTRODUCTION

Over the past few decades, the electronics industry has been experiencing rapid and unprecedented development due to the use of integrated circuits (ICs) in computing, telecommunications, and consumer electronics [1]. Therefore, the demand for very-large-scale integration (VLSI) designs with improved power, performance, and area (PPA) is increasing daily. Furthermore, in all digital circuits, adders are the fundamental building blocks of arithmetic units such as addition, subtraction, multiplication, division, compressors, comparators, and parity checkers. Thus, an optimized design of full adder circuit can reduce power consumption, delay, and area consumption, and any improvements made in full adders will enhance the overall system's performance [2].

The ripple carry adder (RCA) is the simplest adder, however, it has time complexity due to carry-chain [3]. Carry save adder (CSA) is used to improve the delay performance of the RCA. Typically, CSA consists of a hierarchy of stand-alone full adder (FA) and performs a number of partial additions as presented in Figure 1; hence, CSA is well-known as 3:2 compressor. The n-disjoint FA in the CSA computes a single sum and carry bit depends on the corresponding bits of the three-input binary; thus, CSA is extremely fast where there is no carry propagation within each CSA cell. Moreover, it is only the final recombination of the final carry and sum requires a carry propagation addition that simply outputs the carry bits instead of propagating them to the side [4]. However, the CSA avoids carry propagation by treating the intermediate carries as outputs instead of advancing them to the next higher bit position, thus saving the carries for later propagation. Therefore, the circuit's time delay is reduced by using the above-mentioned way of adding bits.

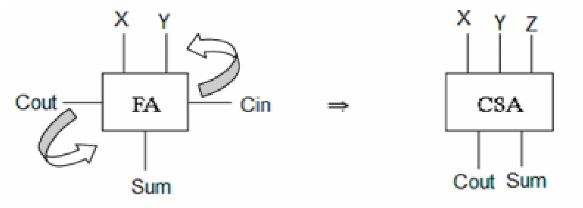


Figure 1. Block diagram of carry save adder (CSA).

VLSI design engineers are constantly working to overcome the PPA trade-off issue and to satisfy the current and future developments of ICs. One such approach is to optimize the circuit design by reducing the number of transistors used and minimizing the physical area occupied by the circuit. In recent years, there has been growing interest in the use of alternative circuit design techniques, such as the modified gate–diffusion input (MOD–GDI) technique, which offers a trade-off between power consumption and circuit delay. The MOD–GDI technique is a new technique of low power digital circuit design which is adopted from the gate–diffusion input (GDI) technique [5]. It involves using a combination of transmission gates and inverters to construct logic gates, thus resulting in fewer transistors, lower power consumption, and faster operation. This technique is gaining popularity among researchers, specifically replacing the traditional complementary metal-oxide semiconductor (CMOS) technique in the field of digital circuit design due to its ability to optimize performance while minimizing area and power consumption. Akram's group [6] proposed a 4:2 compressor circuit using MOD–GDI technique and reported that the design improves the power consumption by up to 79%; moreover, it reduces the delay by up to 49%. Additionally, the MOD–GDI technique has been applied in various circuits such as adders, multipliers, and flip-flops. In 2019, Pal's group [7] designed a 16-bit adder circuit using MOD–GDI technique and achieved significant improvement in terms of power consumption, delay, and area as compared to the conventional design.

Moreover, the Boolean simplification is another technique that can be used to reduce the number of gates and hence the overall power consumption and delay. It involves the simplification of Boolean expressions using laws and theorems of Boolean algebra. Furthermore, its primary goal is to reduce the complexity of a digital circuit while maintaining its functionality. Therefore, this technique helps to minimize the number of gates and transistors used in the circuit. One common method of Boolean simplification is the Karnaugh map, which is used to identify common terms and eliminate redundant terms in a logic expression. Another technique is the Quine–McCluskey algorithm, which involves generating a prime implicant chart and minimizing the expression using the comparison and elimination process. Several studies have evaluated the effectiveness of Boolean simplification in digital circuit design. For example, Wang's group [8] proposed a method for optimizing low-power digital circuits using Boolean simplification, which resulted in a reduction in power consumption by up to 60%. In 2018, Sahu's group [9] also used Boolean simplification to optimize the design of a combinational logic circuit resulting in a 27% reduction in the number of gates and a 10% reduction in power consumption. Moreover, Malik's group [10] proposed combining Boolean simplification with transistor sizing to optimize the design of a 4-bit adder circuit which resulted in a 30% reduction in power consumption and a 38% reduction in delay.

Inspired by these advantages, we proposed both techniques, i.e., MOD–GDI and Boolean simplification, to design the FA circuits with the aim of improving their performance. A comparative analysis is conducted to overview the effect of the two optimization methods via their implementation on a 1-bit FA as a small-scale design under test (DUT), and the designed 1-bit FA is implemented on the 4-bit CSA as a small-scale DUT. Performance factors such as area, delay, and power consumption were analyzed using Cadence Virtuoso simulation based on SilTerra 180 nm technology.

2. METHODOLOGY

Design and analysis were conducted using Cadence Virtuoso®, such as Schematic Editor, Symbol Editor, Layout Editor, and Analog Design Environment (ADE).

2.1. Schematic Design

The experimental procedure began with the placing of NMOS and PMOS in corresponding order to build three different configurations of 1-bit FA, which are:

i. Basic full adder (FA_1),
ii. Boolean simplification full adder (FA_2), and
iii. MOD–GDI full adder (FA_3).

The basic FA (FA_1) is built up by placing CMOS logic gates, such as two Exclusive-OR (XOR) gates, two AND gates, and one OR gate. The configuration is connected according to the Boolean expression of the FA referred to as (1) and (2): where A, B, and Carry in (C_{in}) are the inputs given to the FA, while Sum and Carry out (C_{out}) are the outputs obtained. The total number of transistors used in this configuration is 42.

$$\text{Sum}= C_{in} \oplus A \oplus B \tag{1}$$

$$C_{out} = A \cdot B + C_{in} \cdot (A \oplus B) \tag{2}$$

The Boolean simplification full adder (FA_2) is created by simplifying the Boolean expression, which represents the combinational logic circuits of the basic FA using Boolean algebra laws. This is to reduce the original expression to an equivalent expression that has fewer terms and, hence, directly reduce the number of logic gates as well as the number of transistors needed to implement the same combinational logic circuit. The schematic diagram for FA_2 can be drawn with a total transistor count of 30 by applying the CMOS gate circuitry, which consists of a PMOS pull-up network and a NMOS pull-down network, as shown in Figure 2 (a). Considering Equation (1) and (2), the condition for FA_2 generation can be represented as follows:

$$\text{Sum} = (A \cdot B \cdot C_{in}) + \overline{C_{out}} \cdot (A + B + C_{in}) \tag{3}$$

$$\overline{C_{out}} = (A \cdot \overline{B} \cdot \overline{C_{in}}) + (B + \overline{A} + \overline{C_{in}}) + (C_{in} \cdot \overline{A} \cdot \overline{B}) \tag{4}$$

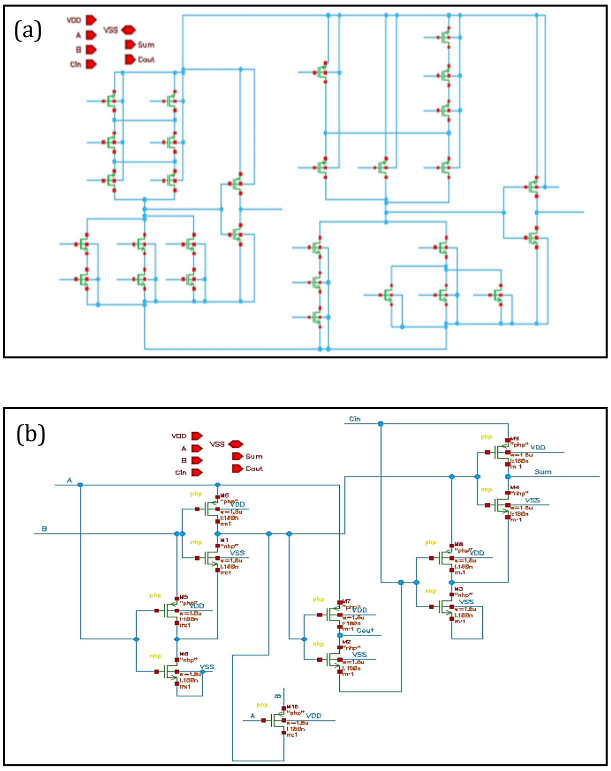
In the MOD–GDI cell, the bulk node of every PMOS transistor is connected to VDD and the bulk node of every NMOS transistor is connected to GND. This technique uses conventional four-terminal NMOS and PMOS transistors, which makes it easy for implementation in various standard CMOS technologies. Therefore, the MOD–GDI full adder (FA_3) is designed by replacing the CMOS logic gate with the MOD–GDI cells to carry out the logic function, which will allow the formation of the Boolean expression where the Sum expression is referred to as (1) while the Carry out expression is changed. Equation (5) expresses the algorithm of obtaining C_{out}, which is in the form of S'A+SB; this is the logic function for the 2×1 MUX, as the operations of MUX can be implemented using MOD–GDI with a minimal transistor count of two. However, there is a feedback transistor implemented into the circuit to improve the full swing. The overall schematic diagram drawn for FA_3 consumed a total of 11 transistor counts, as shown in Figure 2 (b).

$$C_{out} = A \cdot \overline{(A \oplus B)} + C_{in} \cdot (A \oplus B) \quad (5)$$

The four operands of 4-bit CSA are intended for the exploitation of twelve full adders in three blocks during which within initial two blocks consist of eight full adders of CSA. Therefore, the last block consists of four full adders which are RCA cascaded in a manner to produce a result of its preceding state. The methodology to design the 4-bit CSA circuit is similar to that of the 1-bit FA, where the schematic of 4-bit CSA is drawn based on the instance created for the 1-bit FA previously. In other words, the schematic design of 4-bit CSA circuit also consists of three different types of schematics, which is as follows:

i. Basic 4-bit CSA (CSA_1),
ii. Boolean simplification 4-bit CSA (CSA_2), and
iii. MOD-GDI 4-bit CSA (CSA_3).

However, all three schematic diagrams share the same configuration, as demonstrated in Figure 2 (c), but different in the FA used to implement the circuit design. The 4-bit CSA has 22 pins in total, 16 of which are input pins and the remaining six are output pins labelled as S0 – S4, and C_{out}.



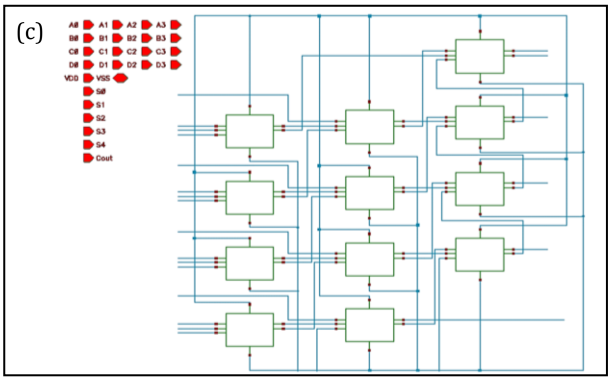


Figure 2. Schematic level of (a) 1-bit Boolean simplification FA, (b) 1-bit modified–gate diffusion input FA, and (c) 4-bit carry save adder.

2.2. Layout Design

In the Layout XL workspace, MOSFET layout is generated from source based on corresponding schematic design, and each of the MOSFET is placed according to the floor plan. The connection is routed by different metal layers, such as Metal 1, Metal 2, etc. Respective vias are placed in between the contact of different metal layers to link the different layers and form a connection between them. The physical verification flows include the use of Design Rule Check (DRC) and Layout versus Schematic (LVS) to verify the layout interconnection of each schematic design that was previously done. The software calculates the overall area and the optimized layout of the proposed 1-bit MOD–GDI full adder, as shown in Figure 3.

Consequently, parasitic extraction (PEX) is performed to extract physical layout information from the designed layout, which is parasitic parameters, such as resistance, capacitance, and parasitic effects. Then, it generates a corresponding netlist after the execution. PEX is commonly performed after LVS and DRC to ensure the accurate representation of the circuit for further analysis and verification. PEX helps designers to validate and optimize their designs by capturing accurate electrical behavior of the circuit. The same procedures are done for the design of four operands of 4-bit CSA via the formation of a hierarchical design, where the layout drawn for 1-bit FA is implemented 12 times.

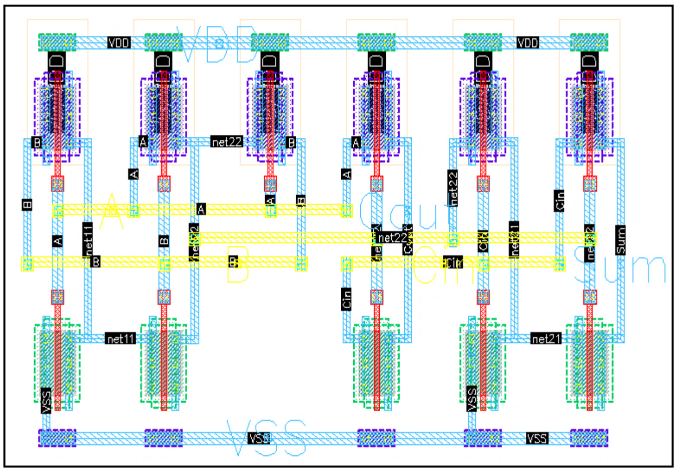


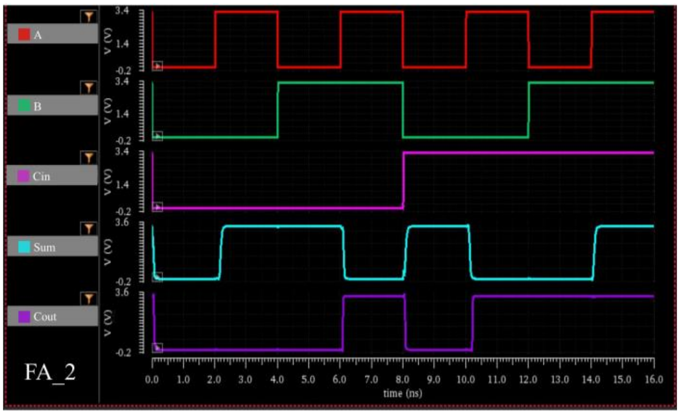
Figure 3. Layout design of 1-bit MOD–GDI full adder (FA_3).

3. RESULTS AND DISCUSSION

The designed circuit is evaluated in terms of propagational delay, power dissipation, and area of consumption; where propagational delay and power dissipation will be simulated from the testbench circuit for both 1-bit FA and 4-bit CSA circuit with their respective technique implemented. On the other hand, the area of consumption is calculated from the layout diagram. For the testbench circuit, each of the input terminals are connected to a pulse voltage source model (VPULSE). The input signal is set to test all possibility patterns to verify the correct functional of a full adder.

Figure 4 shows the waveform of the simulation; the operation of a FA is verified through results that show the circuit had performed the sum operation as intended. To conduct simulation of the four operands of 4-bit CSA, a total of 16 VPULSE are implemented in the testbench circuit for the input terminals. Since there is multiple number of inputs for the CSA, only two patterns are considered in this simulation, where all output terminals experienced both rising and falling transitions. Figure 5 shows the simulation done for the four operand 4-bit CSA and outputs are obtained exactly as expected. Hence, the designed schematic is verified to be of correct design. In digital circuits, the correct '1' or '0' logic levels are typically interpreted by voltages. When a voltage level at a node exceeds a specific voltage threshold, it is recognized as a logic high '1'. Conversely, if the voltage level falls below another threshold, it is considered a logic low '0'.

The evaluation for area of consumption of the designed circuit is taken in Layout XL. Table 1 shows the area consumption obtained from the EDA's calculator and total number of transistors counts utilized in the designs. For example, Figure 3 shows the layout of the proposed 1-bit MOD–GDI full adder (FA_3) in 180 nm technology with an area of 274.4832 μm². It is worth mentioning that the FA_3 design requires only 11 transistors, whereas FA_1 and FA_2 require 42 and 30 transistors, respectively. Hence, FA_1 and FA_2 have larger corresponding area of consumptions than FA_3, which are 1398.4299 μm² and 596.7000 μm², respectively. A similar observation trend can be found for 4-bit CSA, which revealed that the CSA_3 design consumed the smallest area of 3492.4688 μm² with the fewest transistors used of 132.



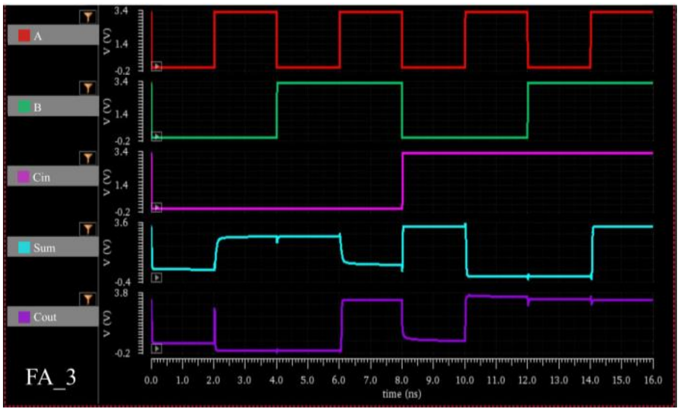


Figure 4. Waveform simulated for 1 bit-full adder of the proposed Boolean simplification (FA_2) and modified–gate diffusion input (FA_3) technique.

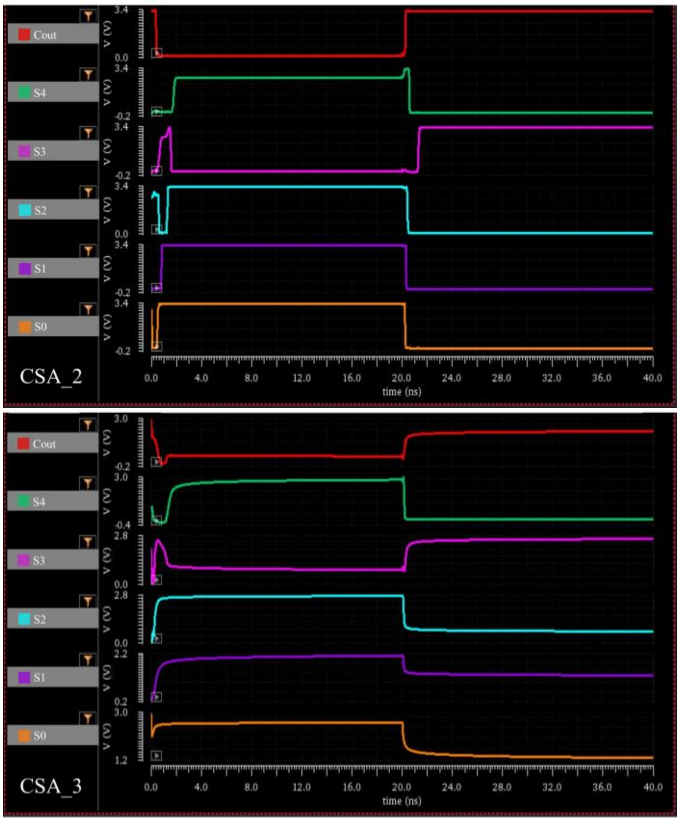


Figure 5. Waveform simulated for 4-bit carry save adder of the proposed Boolean simplification (CSA_2) and modified–gate diffusion input (CSA_3) technique.

With an aim to optimize the circuit's performance in terms of delay and power, transient response is implemented as the analysis method for this evaluation process. The propagation delay is evaluated by determining whether the edge type had been correctly selected within the rising edge and falling edge, as well as the order of the edge, as there is still possibility for glitch in signal waveform. A detailed comparison of the proposed 1-bit FA and 4-bit CSA (i.e., Boolean simplification and MOD–GDI technique) along with the basic designs are given in Table 2 and Table 3 for pre- and post-layout simulation, respectively.

It was observed that in the MOD–GDI technique, the power consumption and speed is significantly improved than that of other full adders. The average power was reduced dramatically from 885.5260 to 0.5888 μW for FA_1 and FA_3, respectively. This is possibly due to the use of a smaller number of transistors that could minimize the power consumed by the proposed FA_3 in comparison with FA_1 and FA_2. Accordingly, the delay was found to be 108.8549, 396.0005, and 45.5206 ps for FA-1, FA_2, and FA_3, respectively. The design technique using MOD–GDI was further extended for implementing 4-bit CSA (Table 3) as well, and it was found to be working efficiently with a power of 0.0031 μW and a delay of 500.7865 ps at a 3.3 V supply voltage. It was observed that the MOD–GDI CSA reduced the overall propagation delay by 28% while significantly improving average power consumption from 1.6134 to 0.0031 mW. The findings indicate that the proposed MOD–GDI technique improved CSA performance compared to the conventional CMOS technology design (CSA_1). The overall percentage improvements for pre- and post-layout simulation are approximately 20% and 28%, respectively. This might be due to the presence of parasitic effects including capacitance, inductance, and resistance, which cause current delays and slower signal transitions. This slower switching behavior can result in reduced switching activity and consequently lower the average power consumption. Nonetheless, the CSA implemented with the Boolean Simplification full adder (CSA_2) had the highest propagation delay, despite having fewer transistors compared to CSA_1. This is due to the Sum terminal from each of the 1-bit full adders being required to wait for the result of the C_{out} part to be obtained first since the modified Boolean expression for the Sum consisted of the use of C_{out}.

Among the designs developed, the MOD–GDI technique demonstrated the best optimization in PPA, implying that the performance evaluation in terms of propagation delay, power dissipation, and area consumption is correlated with the transistor counts, making it the most compact design with 11 transistors used in an area of 274.5 μm^2. The performance parameter of the proposed MOD–GDI were compared with the existing FA designs such as transistor hybrids [11] and new hybrid [12] techniques in 180 nm technology (Table 4). Furthermore, the design used in this study offered improved PPA by 80% and 82% in delay and power consumption, respectively, with respect to the best reported design by Bhattacharyya's group [12].

Table 1 Comparison between total number of transistors count and area consumption

Design Circuit	Total Number of Transistor Count	Area Consumption (μm^2)
FA_1	42	1398.4299
FA_2	30	596.7000
FA_3	11	274.4832
CSA_1	504	17386.3200
CSA_2	360	7726.1600
CSA_3	132	3492.4688

Table 2 Comparison between propagation delay and power consumption for pre- and post-layout simulation (1-bit FA)

Parameter	Layout	Design Circuit		
		FA_1	FA_2	FA_3
Rising time (ps)	Pre	83.9927	346.7070	14.4579
	Post	98.4118	574.9710	15.0354
Falling time (ps)	Pre	96.5160	140.8460	57.9827
	Post	119.2980	217.0300	76.0058
Propagation delay (ps)	Pre	90.2544	243.7765	36.2203
	Post	108.8549	396.0005	45.5206
Average power (μW)	Pre	771.8550	793.5140	1.3392
	Post	885.5260	824.9440	0.5888

FA_1 = basic full adder, FA_2 = Boolean simplification full adder, FA_3 = modified–gate diffusion full adder.

Table 3 Comparison between propagation delay and power consumption for pre- and post-layout simulation (4-bit CSA)

Parameter	Layout	Design Circuit		
		CSA_1	CSA_2	CSA_3
Rising time (ps)	Pre	517.6250	797.2050	351.9460
	Post	739.3270	1297.9100	462.1640
Falling time (ps)	Pre	446.8390	349.7330	416.9970
	Post	653.5750	581.3660	539.4090
Propagation delay (ps)	Pre	482.2320	573.4690	384.4715
	Post	696.4510	939.6380	500.7865
Average power (mW)	Pre	1.8292	7.6422	0.0035
	Post	1.6134	7.5031	0.0031

CSA = carry save adder, CSA_1 = basic 4-bit CSA, CSA_2 = Boolean simplification 4-bit CSA, CSA_3 = modified–gate diffusion input 4-bit CSA.

Table 4 Performance of full adder in 180 nm technology

Full Adder	Delay (ps)	Average Power (µW)	Transistor Count	Ref.
Transistor Hybrid	231	3.35	18	[11]
New Hybrid	224	4.15	16	[12]
Proposed MOD-GDI	36	1.34	11	This work

4. CONCLUSION

This study compares the performance of the proposed MOD–GDI carry save adder with that of other adder configurations (conventional CMOS, Boolean simplification, etc.) with respect to the number of transistors, area, propagation delay, and power consumption. All 1- and 4-bit adder architectures are simulated using Cadence Virtuoso based on the SilTerra 180 nm technology. Among the full adders, the 1-bit MOD–GDI designed using 11 transistors showed the best performance with the minimum area of 274.4832 µm^2 and lowest propagation delay of 45.5206 ps compared to conventional CMOS and Boolean simplification. The proposed 1-bit FA was then implemented in the four operand 4-bit CSA and the performance was compared with those implemented using the conventional design and Boolean simplification methods. The proposed MOD–GDI design of the CSA provided the most optimum performance for propagation delay, average power consumption, and area, with values of 500.78665 ps, 0.0031 mW, and 3492.4688 µm^2, respectively. The proposed low-power, high-speed adder can be used for the efficient implementation of multipliers and other complex circuits.

ACKNOWLEDGMENTS

The author would like to thank the School of Electrical and Electronic Engineering, Universiti Sains Malaysia (USM) and Collaborative Microelectronic Design Excellence Centre (CEDEC), USM as well as Institute of Nano Electronic Engineering (INEE), Universiti Malaysia Perlis for the continuous assistance and support.

REFERENCES

[1] A. Pandey and T. Pratyush. (2017). Implementation of full adder at one terabyte per second speed using cadence. International Journal of Engineering and Technical Research, Vol. 7, pp. 1420-145.

[2] A. Raghunandan and H. V. R. Aradhya. (2019). Area and timing analysis of advanced adders under changing technologies. International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT), pp. 33-38.

[3] V. Vijay, M. Sreevani and E. Mani Rekha. (2022). A review on n-bit ripple carry adder, carry select adder and carry skip adder. Journal of VLSI Circuit and Systems, Vol. 4, No. 1.

[4] A. Pandey and T. Pratyush. (2017). Implementation of full adder at one terabyte per second speed using cadence. International Journal of Engineering and Technical Research, Vol. 7, No. 5, pp. 1420145.

[5] O. A. Badry and M. A. Abdelghani. (2018). Low power 1-bit full adder using full-swing diffusion input technique. International Conference on Innovative Trends in Computer Engineering, pp. 205.

[6] U. Akram and J. Akram. (2018). Design of 4:2 compressor using modified-gate diffusion input technique. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Vol. 7, No. 3, pp. 2172-2179.

[7] S. Pal, S. Maity, and S. Sarkar. (2019). Design of a 16-bit adder circuit using modified gate diffusion input (MGDI) technique. International Journal of Advanced Computer Science and Applications, Vol. 10, pp. 89-94.

[8] H. Wang, Q. Dong, Y. Liu, and X. Lu. (2017). Low-power digital circuit optimization based on Boolean simplification. Journal of Electronic Science and Technology, Vol. 15, No. 4, pp. 337-341.

[9] K. K. Sahu, S. V. Kulkarni and M. P. Joshi. (2018). Combinational logic circuit optimization using Boolean Simplification. Journal of Advanced Research in Dynamical and Control Systems, Vol. 10, No. 14, pp. 253-259.

[10] V. Malik, N. Singh, and A. Jain. (2016). Optimization of 4-bit adder circuit using Boolean simplification and transistor sizing. International Journal of Computer Applications, Vol. 140, No. 5, pp. 20-23.

[11] S. Ancy Simon and S. Sujithamol. (2020). Implementation of carry save adder using novel eighteen transistor hybrid full adder. 2020 International Conference on Power Electronics and Renewable Energy Applications (PEREA), pp 1-4.

[12] P. Bhattacharyya and V. Kumar. (2015). Performance analysis of a low-power high-speed hybrid 1-bit full adder circuit. IEEE Trans Syst., Vol. 23, pp. 1-8.