Evolution in Electrical and Electronic Engineering Vol. 2 No. 2 (2021) 110-119

EEEE

Homepage: http://publisher.uthm.edu.my/periodicals/index.php/eeee
e-ISSN : 2756-8458

Design and Analysis of 8-Bit Low Power ALU in 45 nm Using GDI Technique

Wong Siong Hui[1], Siti Hawa Ruslan[1]*

[1]Faculty of Electrical and Electronic Engineering,
Universiti Tun Hussein Onn Malaysia, Batu Pahat, 86400, MALAYSIA

*Corresponding Author Designation

DOI: https://doi.org/10.30880/eeee.2021.02.02.014
Received 4 July 2021; Accepted 27 July 2021; Available online 30 October 2021

Abstract: Arithmetic Logic Unit (ALU) is the main part inside a computer's central processing unit (CPU) that performs arithmetic and logic operations. Since the technology grows faster, power dissipation in ALU becomes vital to be reduced. So, Gate Diffusion Input (GDI) is introduced because less transistors are used to achieve low power dissipation. The limitation of the GDI technique is poor voltage swing output, but it can be solved by modifying the GDI cell and adding some transistors. In this paper, Mentor Graphics software is used to build the 8-bit ALU in 45 nm using the GDI technique with the power supply of 1.0 V. Results show that the GDI technique will help minimize power dissipation, time delay, and area.

Keywords: ALU, Power Dissipation, GDI

1. Introduction

Arithmetic Logic Unit (ALU) acts as a heart inside every digital computer's central processing unit (CPU). The function of ALU is to perform arithmetic and logic operations, then sent the information to the computer memory. As the technology becomes more advanced and increased in design complexity, power consumption will become a significant challenge due to increasing transistors, which required more energy and power to integrate these transistors [1]. Equation 1 clearly shows that power dissipation consists of two mainstays: static and dynamic power dissipation [2]. Not only that, more transistors usage will also cause some delay for the output execution and increase area in Very Large-Scale Integration (VLSI) design [3].

$$P_{avg\ power} = P_{dynamic\ power} + P_{static\ power} \qquad Eq.1$$

Gate-Diffusion Input (GDI) is a low power technique that only implements two transistors to design complex logic functions [4]. Figure 1 shows a basic GDI cell that consists of three inputs which are named P (input to the source/drain of pMOS), G (common gate input of both the nMOS and pMOS), and N (input to the source /drain of nMOS). Compared with the CMOS inverter, the GDI may be different in terms of the bulks, in which both nMOS and pMOS are connected to N or P, respectively.

Corresponding author: sitihawa@uthm.edu.my

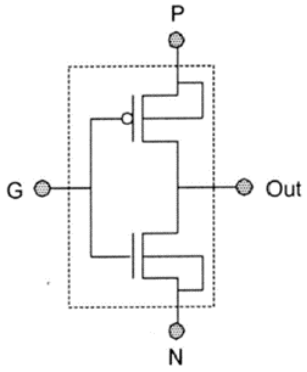

Figure 1: Basic GDI cell [5]

In the GDI technique, nMOS will produce weak logic 0, and pMOS will produce weak logic 1 because the GDI technique has disadvantages of output voltage swing degradation. In order to overcome this problem, the GDI cell is modified by permanently connecting the bulk terminal of pMOS and nMOS to the V_{dd} and ground, respectively. Not only that, by adding certain transistors, a full swing output can be achieved [6], [7].

So, in this paper, the functionality and low power 8-bit ALU circuit are designed using the GDI technique in 45 nm technology, and the power supply is set at 1 V. Besides, the operations of the ALU, whether it is logic or arithmetic operations will be selected by 3-bit opcodes. Lastly, the performance of the GDI ALU is analyzed in terms of power dissipation, time delay, and area.

2. Materials and Methods

The 8-bit ALU is designed from the front-end to the back-end using Mentor Graphics software. In the front-end phase for the GDI technique, the schematic diagram of the 2-to-1 multiplexer and basic logic gates such as NOT gate, AND gate, and OR gate designed at the transistor level are shown in Figure 2 until Figure 5.

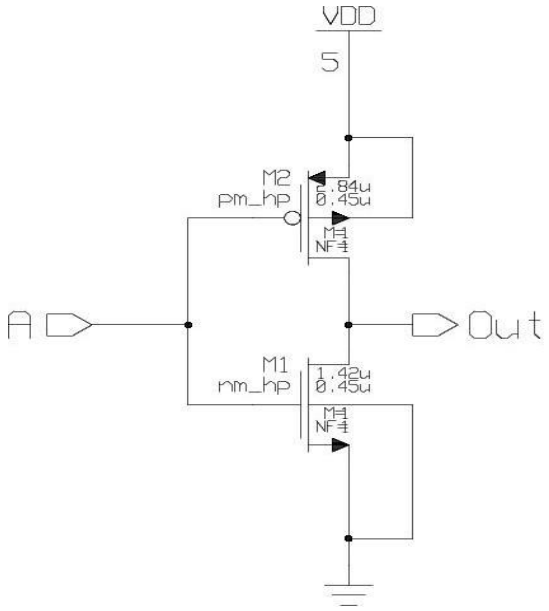

Figure 2: The schematic diagram of NOT gate based GDI

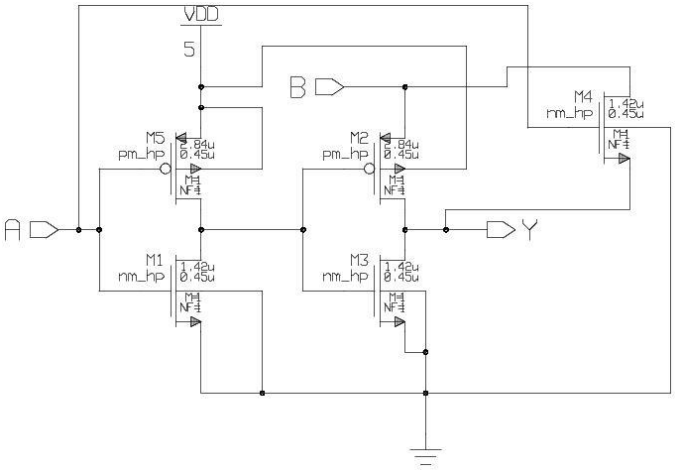

Figure 3: The schematic diagram of AND gate based GDI

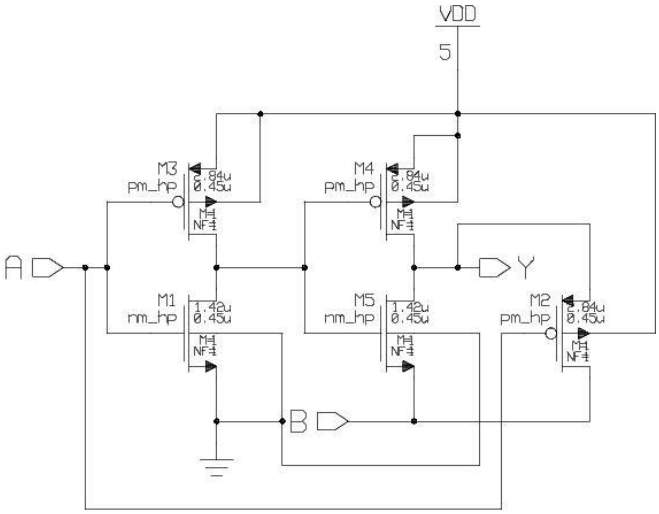

Figure 4: The schematic diagram of OR gate based GDI

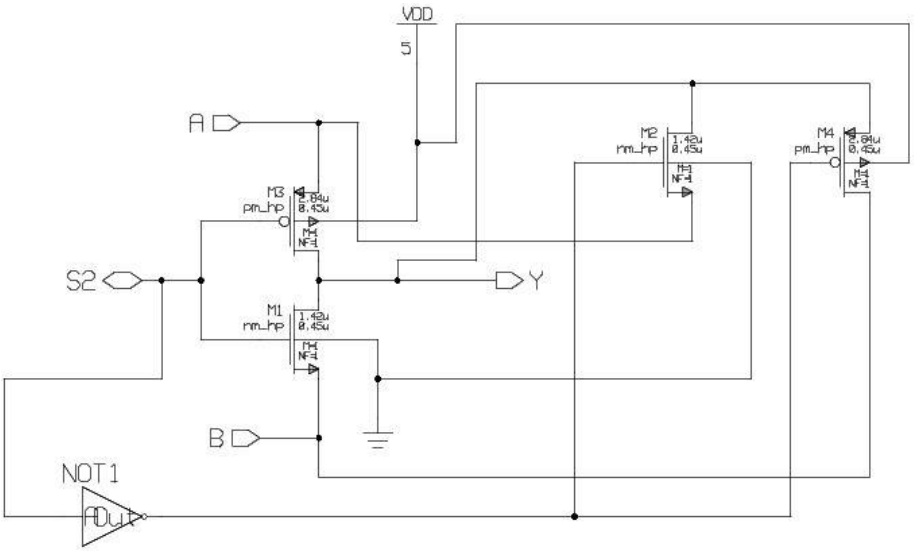

Figure 5: The schematic diagram of 2-to-1 multiplexer based GDI

After the functionality at the transistor level of the 2-to-1 multiplexer and basic logic gates have been verified to work according to the truth table, the hierarchical method is used by implementing each basic logic gate and 2-to-1 multiplexer repeatedly to form complex circuit designs. For example, XOR gate, XNOR gate, full adder, and 4-to-1 multiplexer. All the schematic designs are checked to ensure the circuits do not have any errors, and the functionality is also verified.

In Figure 6, a 2-bit ALU is designed, and it has consisted of four 4-to-1 multiplexers, two full adders, two 2-to-1 multiplexers, and two simple logic gates such as OR gate, AND gate, NOT gate, XNOR gate, and XOR gate. In order to build the proposed 8-bit ALU shown in Figure 7, the hierarchical method of four 2-bit ALUs is used.

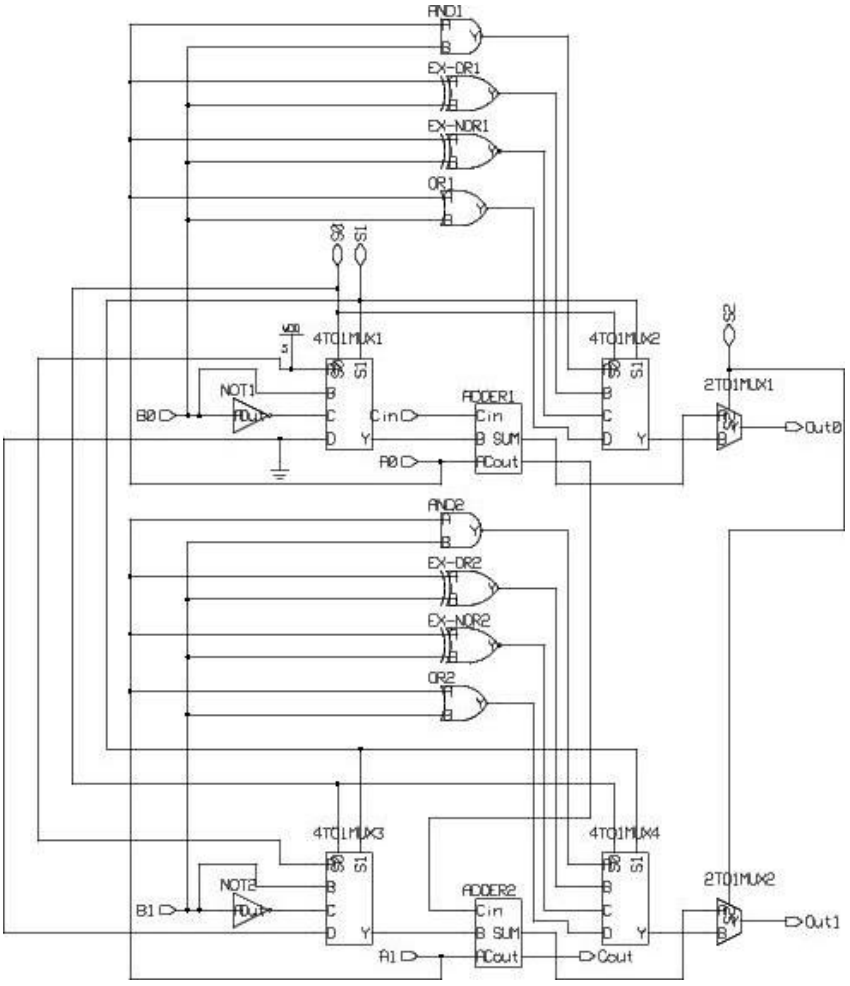

Figure 6: The logic block diagram of 2-bit ALU

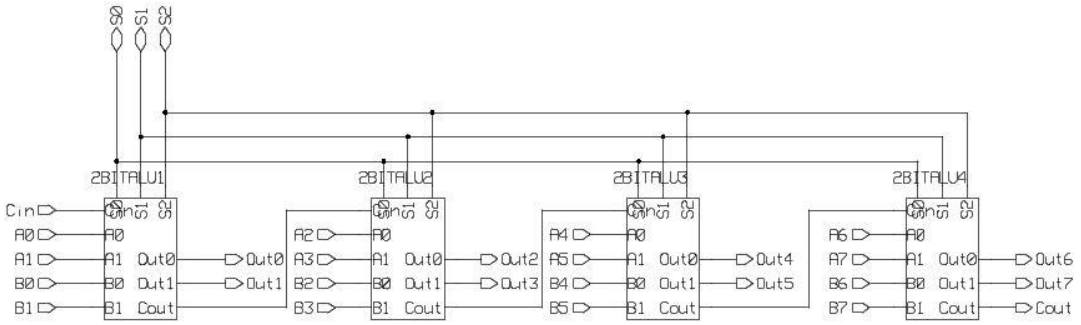

Figure 7: The logic block diagram of 8-bit ALU

For the operation of the 8-bit ALU, it will perform arithmetic (decrement, addition, subtraction, increment) and logical (logical AND, logical XOR, logical OR, logical XNOR) operations that are shown in Table 1, in which the three select signals will responsible to select the operation.

For the INCREMENT operation, the input will be the logic '1' and is performed by adding '1' to the addend. Besides, a complement of B will be implemented in the ALU for the SUBTRACTION operation. The input will be the logic '0' for the DECREMENT operation, and the operation will be similar to the subtraction operation [8][9].

Since the 8-bit ALU consists of arithmetic and logical operations, the first 4-to-1 multiplexer and the full adder will be considered the arithmetic part. In contrast, the second 4-to-1 multiplexer and the simple logic gates will be considered as the logical part. For the arithmetic part, the first 4-to-1 multiplexer will be responsible for selecting the input according to the condition of the selection lines and sending it to the full adder to compute the result. For the logical part, the second 4-to-1 multiplexer is responsible for selecting either OR, AND, XNOR, or XOR operations according to the condition of the selection lines. Lastly, the 2-to-1 multiplexer will decide whether to perform arithmetic or logical operations and sends it out.

Table 1: Truth table ALU

Selection Lines			Operations
S2	S1	S0	
0	0	0	DECREMENT
0	0	1	ADDITION
0	1	0	SUBTRACTION
0	1	1	INCREMENT
1	0	0	AND
1	0	1	XOR
1	1	0	XNOR
1	1	1	OR

3. Results and Discussion

In this section, the results obtained from the front-end to the back-end design for the 8-bit ALU in the GDI technique are illustrated. First and foremost, the functionality of the 8-bit ALU is checked by obtaining the desired simulation waveform as shown in Figure 8 and 9, where the result of the output waveform will perform similar to the proposed operation in Table 1. For example, the 8-bit ALU will operate AND operation when 100 has been chosen by the selector of S2S1S0 respectively, as shown from period 40-50 ns in cursor 2. So, the output waveform will only perform logic 1 when both of the input A and B is at the logic level high. Although the output waveform for the 8-bit ALU is correct, some spikes occurred in the waveform, which may be due to the switching issue in the transistor. In order to solve this issue, the circuits are needed to do some modifications, which includes changing the width and length of the transistors being used in the circuit.

After verifying the waveform for the 8-bit ALU is correct, the power dissipation and time delay of the 8-bit ALU are analyzed. Figures 10 and 11 show that the 8-bit ALU needs around 2.2229 ns to perform the output and consumed around 510.08 μW for the power. Since the 8-bit ALU is more complex and requires more transistors, the 8-bit ALU will consume more power and need more time to execute the output waveform.

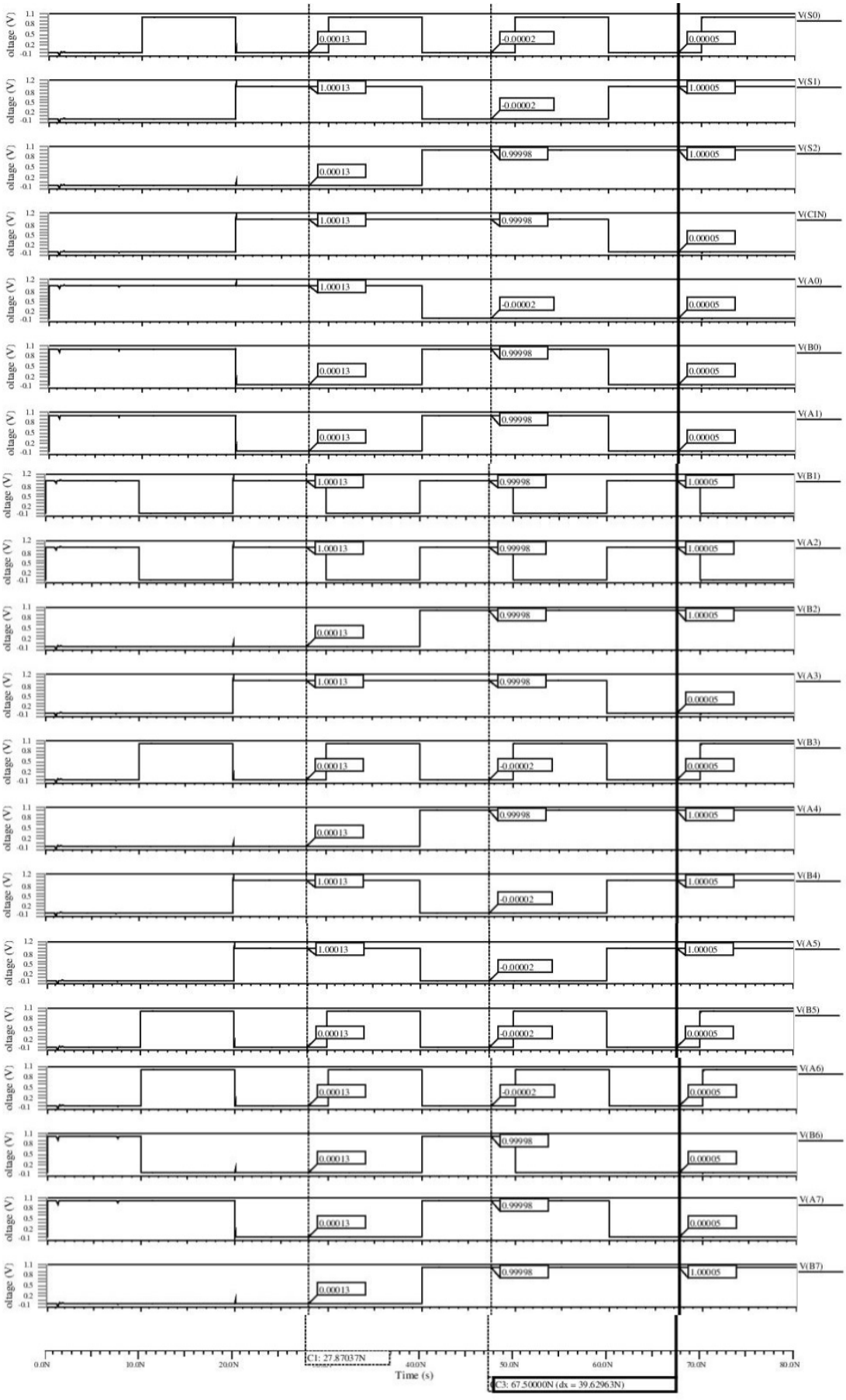

Figure 8: (Continued on next page)

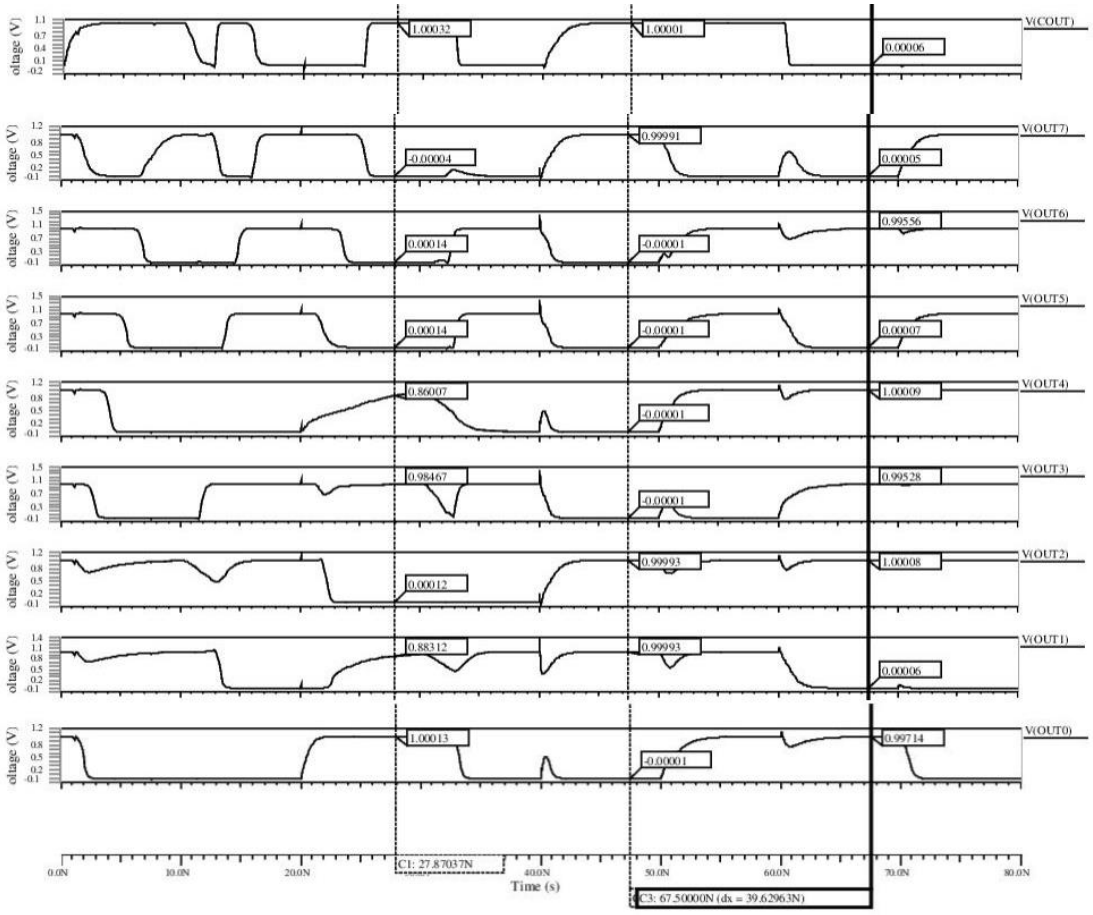

Figure 9: Waveform of 8-bit ALU in GDI technique

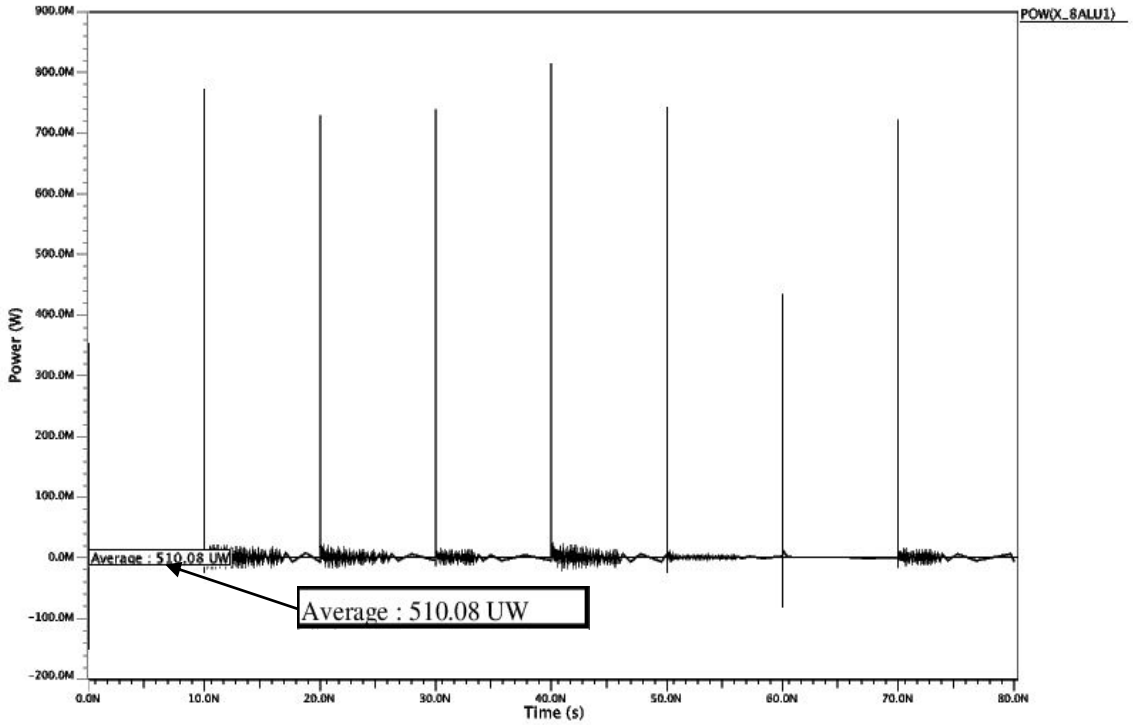

Figure 10: Average power dissipation of 8-bit ALU based GDI technique

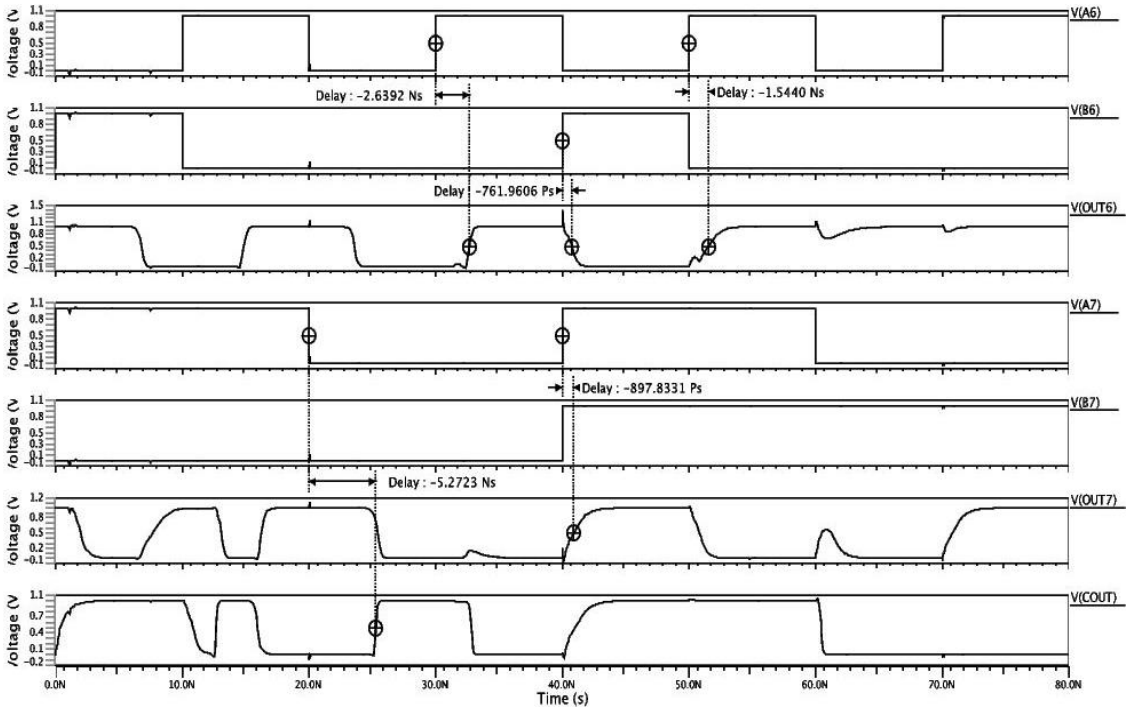

Figure 11: Different time delay of 8-bit ALU based GDI technique

After finish designing the front-end, the design continues with the physical layout for each schematic diagram. The same hierarchical method is used to design the complex physical layout. Figure 12 shows that the physical layout of the 8-bit ALU for the GDI technique, and the area can be measured by using a ruler in Mentor Graphics to determine the height and width of the layout. Using a simple calculation of 109 × 302 μm, the 8-bit ALU occupies around 35124 μm^2.

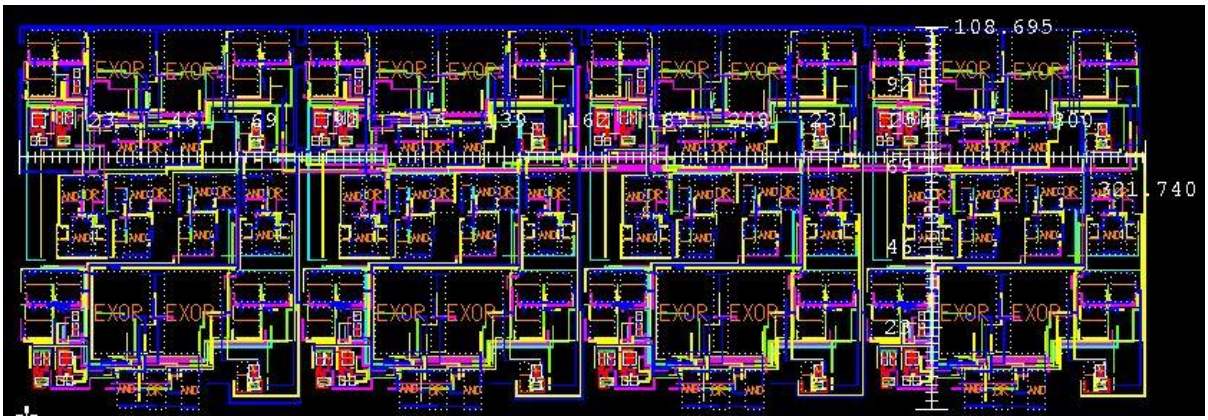

Figure 12: The physical layout of 8-bit ALU for GDI technique

3.1 The performance of the ALU between the CMOS and GDI techniques

The performance of the ALU using CMOS and GDI techniques are analyzed, and the results are summarized in Table 2. Then, some calculations are needed by using equation 2 in order to obtain the percentage of the reduction for the performance analysis between the GDI and CMOS technique. This percentage information is important because researchers or designers will know whether this proposed design of the GDI technique can increase the ALU's performance or not compared to the CMOS technique. Through the research of the existing technology, this proposed design in the GDI technique is expected to reduce around 40% of the power

dissipation, shorten the time delay, and taken less area using a 1.0 V power supply when compared to standard CMOS ALU design.

$$\text{Percentage of the Reduction} = \frac{\text{CMOS} - \text{GDI}}{\text{CMOS}} \times 100\% \quad Eq.2$$

Table 2: Performance analysis of ALU between the CMOS and GDI techniques

Design		Average power (µW)	Average Time Delay (µs)	Area (µm^2)	Transistor Count
2-bit ALU	CMOS	246.16	3.3539	10887.6040	520
	GDI	108.63	1.8767	8637.8292	290
4-bit ALU	CMOS	614.58	3.5065	22033.2580	1040
	GDI	273.49	1.9987	17424.3520	580
8-bit ALU	CMOS	1139.84	3.9399	44174.8587	2080
	GDI	510.08	2.2229	32797.6293	1160

According to Table 2, less transistors are used to develop the ALU using the GDI technique. The number of transistors in the GDI technique is reduced around 44% compared to the CMOS technique. The reduction in the number of transistors in the GDI technique has improved the ALU's performance in terms of power dissipation, time delay, and area.

For the average power dissipation and time delay, the GDI technique consumed less power and shorten the time by executing the output waveform based on the operation, which reduced 55% and 44%, respectively when it is compared with the standard CMOS technique. Besides, when the number of transistors used in the GDI technique is less, surely the physical layout area is also reduced. From the analysis, compared with the standard CMOS technique, it is shown that the ALU using the GDI technique managed to reduce almost 26% of the area.

4. Conclusion

Overall, this project proved that the GDI technique is an efficient low power technique by implementing less transistors into the complex circuits. Since the number of transistors in designing the 8-bit ALU for the GDI technique is less, it will reduce the power dissipation, time delay, and area. Also, when the technology is shrinking down, the performance analysis results such as power dissipation, time delay, and area are improved. The improvement of the ALU's performance is very useful because it made people's life become better. Although this proposed design can already improve the ALU's performance, further research is needed and find a new idea to redesign the ALU circuit by implementing fewer transistors to improve the ALU's performance again to make people more convenient and save power in using digital electronics. This is because the number of transistors is the crucial element to improve the performance of the ALU, so it is necessary to find a new idea to reduce the transistor usage in the ALU.

Acknowledgement

Special thanks to the Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia for its support.

References

[1] S. Kshirsagar and M. B. Mali, "A review of clock gating techniques in low power applications," *Int. J. Innov. Res. Sci. Eng. Technol.*, vol. 4, no. 6, pp. 4324–4328, 2015, doi: 10.15680/IJIRSET.2015.0406103

[2] C. Venkatesan, S. M. Thabsera, M. G. Sumithra, and M. Suriya, "Analysis of 1-bit full adder using different techniques in Cadence 45 nm technology," *2019 5th Int. Conf. Adv. Comput. Commun. Syst. ICACCS 2019*, pp. 179–184, 2019, doi: 10.1109/ICACCS.2019.8728449

[3] Y. Zhou and H. Guo, "Application specific low power ALU design," *Proc. 5th Int. Conf. Embed. Ubiquitous Comput. EUC 2008*, vol. 1, pp. 214–220, 2008, doi: 10.1109/EUC.2008.81

[4] H. Pasha, "Design and analysis of low power high speed full adder cell using modified GDI technique at 90 nm technology," *Int. J. Innov. Res. Electr. Electron. Instrum. Control Eng.*, vol. 4, no. 8, pp. 43–48, 2016, doi: 10.17148/IJIREEICE.2016.4810

[5] A. Morgenshtein, A. Fish, and I. Wagner, "Gate-diffusion input (GDI): a power-efficient method for digital combinatorial circuits," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 10, no. 5, pp. 566–581, 2002, doi: 10.1109/TVLSI.2002.801578

[6] E. V Nagalakshmi, K. Kavya, "Design of 4 bit ALU using modified GDI technology for power reduction," *Int. J. Eng. Sci. Invent.*, vol. 7, no. 2, pp. 38–45, 2018

[7] S. Hiremath and D. Koppad, "Low Power Multiplier and Divider Circuit Using Full Swing Gate Diffusion Input," *J. Eng. Appl. Sci. 12*, no. 12, pp. 9425–9429, 2017

[8] K. Priyanka, A. Prasad, and P. P. M. Krishna, "Low power 16 bit ALU design using full adder and multiplexer," *Int. J. Res.*, vol. 05, no. 01, pp. 2783–2788, 2018

[9] V. Dubey and R. Sairam, "An arithmetic and logic unit optimized for area and power," *Int. Conf. Adv. Comput. Commun. Technol. (ACCT)*, pp. 330–334, 2014, doi: 10.1109/ACCT.2014.70