ISSN 2959-6157

Design and Optimization of the 4-Bit Absolute Value Logic Gate Circuits

Haojie Hu

School of Xi'An Jiaotong Liverpool University, Suzhou, China
*Corresponding author: Haojie.Hu22@student.xjtlu.edu.cn

Abstract:

Absolute value computation of binary numbers has important applications in digital signal processing and arithmetic operations, but conventional circuit designs usually face high latency and energy consumption problems. In order to improve computational efficiency, this paper proposes an optimization method based on logic gate circuits aimed at reducing the number of physical components and simplifying the circuit structure. By using a truth table for logic simplification, a more efficient 4-bit absolute value calculation circuit is designed in this study. Experimental results show that the optimised circuit performs significantly in reducing latency and power consumption and maintains the accuracy of the operation. The results of the study provide a better implementation of digital circuit design for application scenarios that require efficient computation.

Keywords: Logic gate; 4-bit absolute value calculation; Optimization

1. Introduction

In digital signal processing and arithmetic operations, because of the speed and simplicity of binary arithmetic, the calculation of binary number is so important[1]. And also the absolute value calculation of binary numbers is a basic and critical operation. Widely used in applications ranging from digital filtering and image processing to error detection and correction[2]. However, traditional computational methods often involve complex logical operations, resulting in high latency and energy consumption. To overcome these problems, this paper proposes a design method based on the optimization of logic gate circuits,in order to achieve more efficient 4-bit absolute value calculations by simplifying the logic structure. The goal of this research is to reduce the delay and power consumption of the circuit and improve the computational accuracy and efficiency.

This paper focuses on the implementation of a more efficient 4-bit absolute value calculator by optimizing the logic gate circuit design. This paper first introduces the delay and energy consumption problems existing in the traditional circuit design [3], and gives details about how to optimize the design through the logical simplification based on the truth table. In the course of the discussion, the paper analyzes the circuit behavior under different input conditions, and verifies the difference in circuit performance before and after optimization through experimental data. Ultimately, the paper concludes that the optimized circuit not only shows significant improvements in terms of energy consumption and latency, but also exhibits higher accuracy in handling negative inputs[4].

The first chapter of this paper is the introduction. Firstly, it introduces the background and significance of the research, and clarifies the goal and scope of the research. Then chapter 2 details the design of the logic gate circuit before optimization, including the working principle and design ideas of the 4-digit absolute value calculator, and analyses the logic when a negative number is input. The third chapter introduces the optimization design based on the truth table, and discusses the process of simplifying the circuit structure through logical expressions[5]. In chapter 4, the experimental data and results are presented, and the performance comparison of the circuits before and after optimization is verified by examples. Chapter 5 is a discussion section that highlights the advantages of optimal design in terms of reduced energy consumption and reduced latency. Finally, the full text is summarized.

2. Logic Gate Before Optimized

2.1 Principle 4-Bit Absolute Value Calculator

The principle of absolute value calculation is if a number is positive then the absolute value is itself, if the number is negative then the absolute value is opposite. The figure1 and definition formula(1) of the absolute value is given below:

$$|x| = \begin{cases} x, x \geq 0 \\ -x, x < 0 \end{cases} \quad (1)$$

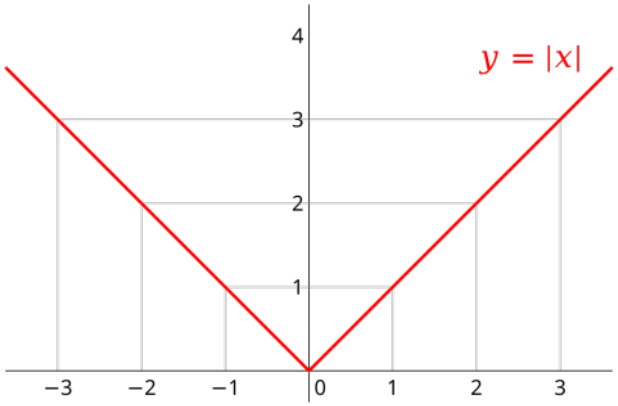

Fig.1 The definition of the absolute

In many digital signal processing and arithmetic operations, the calculation of the absolute values of binary numbers plays a crucial role. This operation is often encountered in applications ranging from digital filtering and image processing to more complex tasks like error detection and correction. The steps involved in computing the absolute value can vary depending on the specific implementation and the constraints of the system.

Traditionally, the most common approach to computing the absolute value of a signed binary number involves determining whether the number is negative[6]. If negative, convert it to its positive equivalent. This is typically done by taking the two's complement of the input number—a process that inverts all the bits of the binary representation and then adds one. For positive numbers, the absolute value remains unchanged, while for negative numbers, this method effectively converts them to their positive counterparts.

2.2 Logic Gate Input Design Ideas Before Optimization

In this design, there are 4 inputs: A0, A1, A2, A3, which involved one sign bit A0, and three value bit A1, A2, A3. When the sign input A0 is 0, which means the number is positive. When A0 equals 1, the input signal is considered to be negative. The figure 2 of 4 inputs is given below.

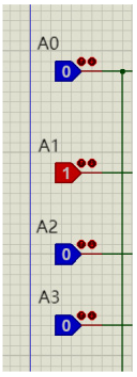

Fig.2 The four inputs

This circuit configuration ensures that when the input is positive, the original input passes through unaltered because the output of the NOT gate U8 effectively 'opens up' the input bits to bypass the inversion and addition stages. Conversely, if the input number is negative (i.e. A0 is 1), the NOT gate U8 is output to 0, thus closing the AND gate. At this point, the circuit will perform the process of minus 1 by default to generate the binary complement of the input number[7].

This design enables the circuit to determine its absolute value efficiently, based on the sign of the input by selectively by passing or participating in the subtraction and inverse steps. Instead of relying on a complex set of conditions to determine whether a number is positive or negative, use a simple NOT gate to make this decision and activate the necessary part of the circuit.

2.3 Absolute Value Logic for Negative Input

The previous section discussed how the circuit can utilize NOT gates and AND gates to pass directly through the original value of the input when the input is positive. However, when the input is negative, the circuit logic takes a different path to calculate the absolute value of the input.

When the input is negative, the output of NOT gate U8 is 0, causing the AND gate U1:A, B, C mentioned above to be non-conductive. This means that instead of passing the original value of the input directly, the circuit will perform the process of adding 1 and inverting it to generate the binary complement of that negative number. However this design takes a different way, choosing a mechanism of first reduction and then repeated reversal. This method helps reduce delay and power consumption, so that the circuit can be optimized in performance. Figure 3shown below is the subtractor part.

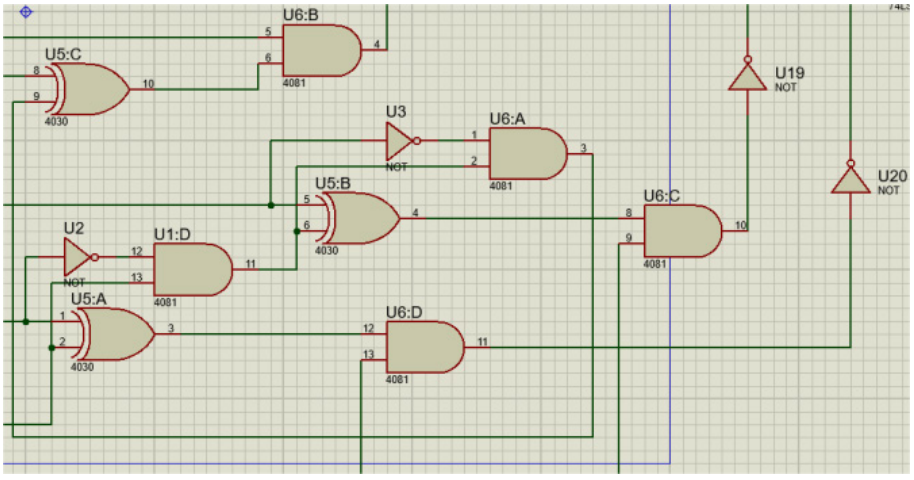

Fig.3 Subtractor consisting of XOR gates and AND gates

2.4 Principle of Subtractor

In the implementation of subtractor, XOR gate plays an important role. The three XOR gates U5: A, B, and C shown in the figure each perform the steps of a subtractor on the input bits. Specifically, the XOR gates depend on different cases of the two input signals to produce outputs when performing subtraction. The XOR gate has an output of 1 and only when the two input signals are different, which makes it possible to perform subtraction operations efficiently without adding additional logic gates.

2.5 The Operation of Inverting A Binary to Obtain the Complement of A Code.

The circuit enters the inverse phase, after the subtraction operation is completed. The NOT gates U4, U19 and U20 are responsible for inverting the output after decrementing by one to generate the final complement output. Figure 4 shows the not gate of taking inverse step.

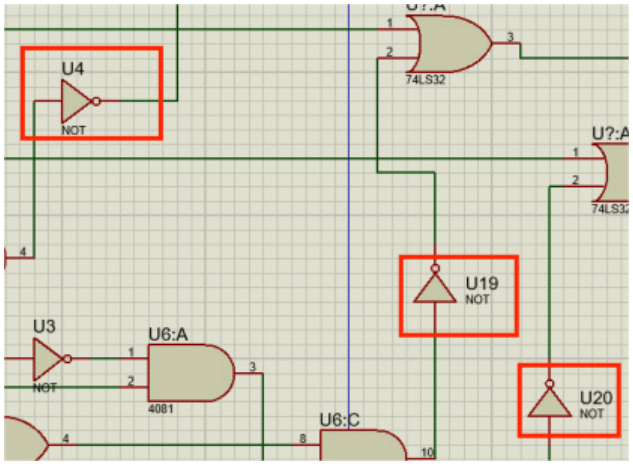

Fig.4 Not gates in taking inverse step

It is worth noting that in the process of generating the complement, the order of minus one and inverting is truly important to the performance of the circuit. By first subtracting and then inverting the design method, the circuit avoids the extra delay and power consumption that may be caused by directly inverting and adding 1. This is because in the process of directly inverting and adding 1, additional carry logic is usually required. Whereas, by first subtracting and then inverting the strategy, the circuit is able to perform the same computational task in a much shorter period of time and has a significant advantage in terms of energy consumption.

3. Optimization idea and logic gate design

3.1 Truth table and logical expression

The original design used a minus 1 and inverse operation to calculate the absolute value of the input, which was effective but resulted in significant delays and energy consumption due to the large number of MOS tubes required. In order to solve this problem, a more direct and efficient optimization method is proposed, which abandon the traditional adder and inverse operation. Instead drawing the truth table directly based on the 4-bit inputs (including the sign bit).

3.2 Generation of truth table

All possible combinations of 4-bit inputs were first listed and by analysing their sign bits and numerical parts, the required outputs for each input case were determined. This process specifies the target output value for each input condition, avoiding complex subtraction and inverse operations. The truth table 1and logic formula(2), (3), (4) are given below.

Table 1. Optimized truth table

A	B	C	D	FB	FC	FD
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	1	0
0	1	0	0	1	0	0
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	0	1	1	0
0	1	1	1	1	1	1
1	0	0	0			
1	0	0	1	1	1	1
1	0	1	0	1	1	0
1	1	0	0	1	0	0
1	0	1	1	1	0	1
1	1	0	1	0	1	1
1	1	1	0	0	1	0
1	1	1	1	0	0	1

$$B = \overline{A}B + A\overline{B} + B\overline{C}\,\overline{D} \tag{2}$$

$$E = \overline{A}C + C\overline{D} + A\overline{C}D \tag{3}$$

$$D = D \tag{4}$$

3.3 Simplification of logic gates

A more simple logic gate circuit is designed by the analysis results of the truth table. This circuit is capable of generating the target output directly from the input without relying on conventional adder and inverter circuits[8]. In addition, the simplified circuit structure not only improves the calculation speed, but also reduces the power consumption, so that the entire circuit has improved in performance and efficiency. This optimised design ensures that the circuit maintains efficient and stable operation when handling complex logic operations. Figure 5 below shows the circuit diagram designed based on the truth table.(Example: Input 0010)

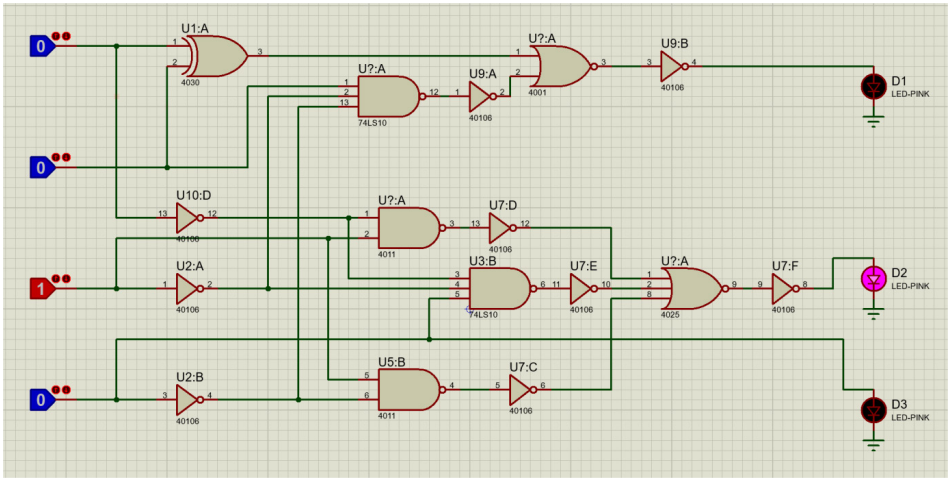

Fig.5 The circuit diagram by the truth table

4. Result

4.1 Adder design based on basic logic:

Figure 6 is the example when the behaviour of the circuit with the 1011 input if the input is negative and one bit advanced. Through the operation of the combination of the logic gates, the experimental results verify that the circuit can handle the negative input correctly and generate the expected absolute value output.

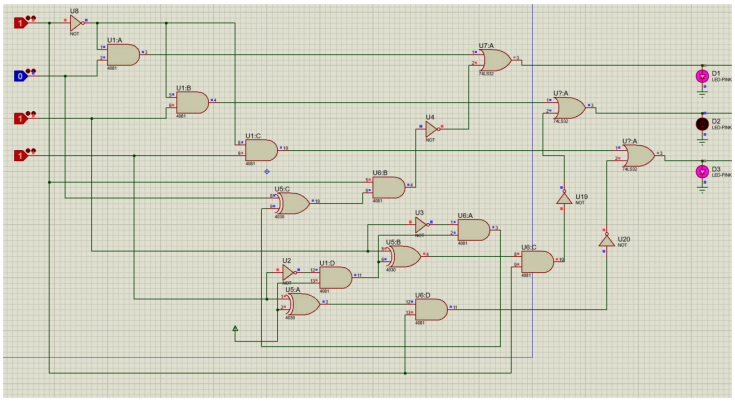

Fig.6 Input is 1011

When the input is 0101, the experimental data in Figure 7 shows that the circuit is able to correctly recognise a positive input through a logic gate. And also directly output the same absolute value, verifying the correctness of the circuit.

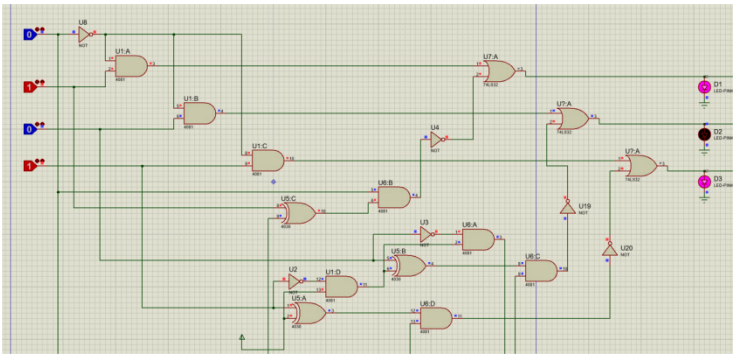

Fig.7 Input is 0101

Figure 8 is when the input is 1111, this experimental data shows the effectiveness of the circuit when dealing with negative numbers. The logic gate combination successfully converts negative numbers to their corresponding absolute values, proving the robustness of the design.

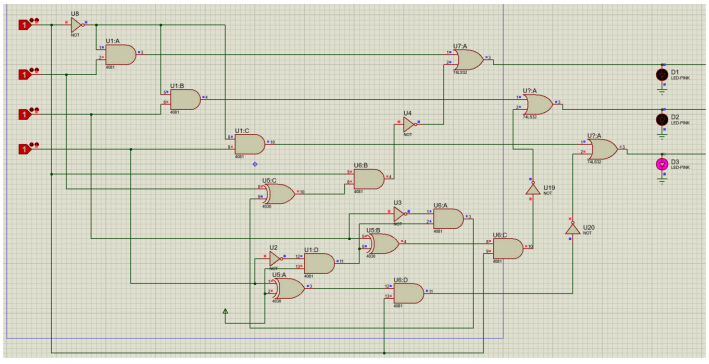

Fig.8 Input is 1111

Figure 6-8 show the examples of circuit design before optimization. It can be concluded that more MOS tubes are used in this design to implement the logic operations of minus one and inverse. Its complex logical structure re-

sults in high latency and energy consumption, though this design is functionally complete. Specifically, due to the higher number of logic gate layers in the circuit, the signal has a longer path to propagate through the circuit, and thus a higher delay.Further more, frequent signal switching results in high dynamic power consumption, which reduces the energy efficiency ratio of the circuit.

4.2 The optimised logic circuit design based the truth table:

Figure 9 shows when the input is 1011, the circuit can still accurately recognize negative input after optimization, and generates the correct absolute value through a simplified combination of logic gates, demonstrating the performance gains of the optimised design.

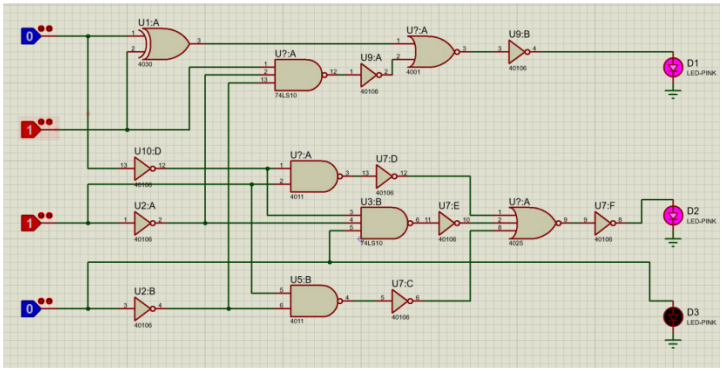

Fig.9 Input is 0110

Figure 10 For the 0101 input, the circuit outputs the absolute value directly through the optimised logic path, further verifying the correctness of this optimised design when dealing with positive numbers.

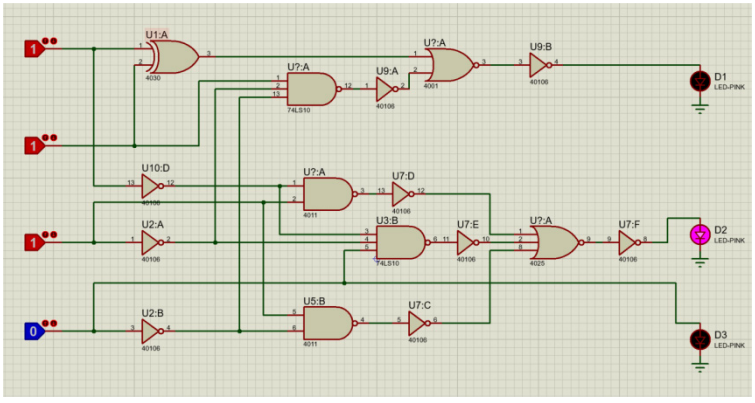

Fig.10 Input is 1110

Figure. 11 shows the accuracy of negative carry processing and the low energy consumption of absolute value when negative carry processing compared with that before optimization in the case of input 1111.

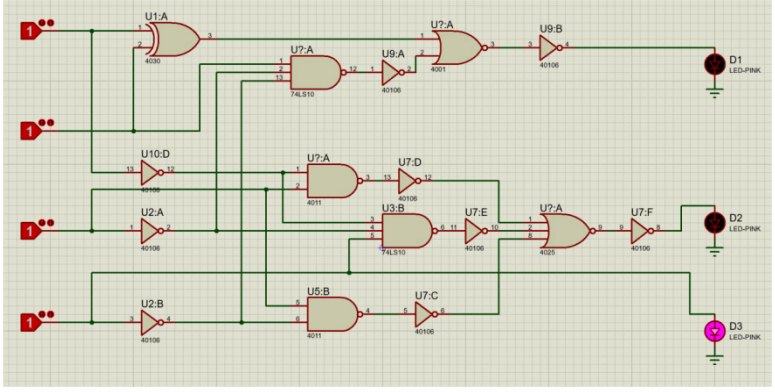

Fig.11 Input is 1111

Figure 9-11 lists the examples of optimized circuit design. The circuit structure has been significantly simplified by reducing the number of redundant logic gate layers and optimising the use of MOS tubes. This optimised design performs the logic mapping directly through the truth table, eliminating the operations of minus 1 and inverse, thus greatly reducing the delay and energy consumption of the circuit.

5. Discussion

The core advantage of the optimised design is that by reducing the number of MOS tubes used in the circuit, it significantly reduces the energy consumption and arithmetic delay of the circuit [9]. Traditional subtractor and inverse circuit designs rely on complex arrays of logic gates. Although this structure can perform the computation accurately, it consumes a lot of energy in the hardware implementation and produces high latency. By simplifying these complex operations into a direct mapping which based on the truth table, the optimised circuit drastically reduces the number of redundant logic gate layers and connection paths. Thus reducing the power consumption in the circuit. This optimisation method is not only use circuit resources more efficiently, but also improves the dynamic performance of the circuit. Especially for applications that need to process large-scale input data. The optimized circuit can achieve a faster speed when it calculate the absolute value with a lower energy consumption. This increase of efficiency makes optimal design not only applicable to simple logic operations, but also has the potential to be generalised in the design of VLSI [10]. This optimization fully reflects the necessity and effectiveness of this design optimization.

6. Conclusion

This paper successfully proposes a optimized logic gate method of the design of 4-bit absolute value circuits, aiming at reducing latency and energy consumption. The circuit performance has been significantly improved, by changing the traditional subtracter and inverse circuit design method to a simplified design which is based on the direct logic of truth table. Because of reducing number of MOS fet and logic gate layers, the optimised circuit shows lower latency and energy consumption, in order to improves the overall efficiency and stability of the circuit. These results prove the necessity and effective of the proposed optimization strategy, especially in applications of requiring efficient and fast processing of input big data. Future work could explore the optimisation and scalability of more complex arithmetic operations further, in order to make progress with digital signal processing and VLSI design.

References

[1] Low-Voltage/Low-Power Integrated Circuits and Systems: Low-Voltage Mixed-Signal Circuits | IEEE eBooks | IEEE Xplore. (n.d.).

[2] AVR223: Digital Filters with AVR. (2008). In Microcontrollers Application Note.

[3] Dong, X., Jing, B., & Yang, X. (2021). Improved Design of a 4-bit Absolute-Value Detector Using Simplified Chain Carry Adder. Journal of Physics Conference Series, 2113(1), 012043.

[4] Exploration of energy-delay tradeoffs in digital circuit design. (2008, March 1). IEEE Conference Publication | IEEE Xplore.

[5] Lin Shaoling. (2014). Optimal design of multi-cylinder sequential electronic control system based on the truth table. In Journal of Mudanjiang University (p. Eleven).

[6] Wei, Y. (2013). Study on complement codes of the negative. ELECTRONIC TEST, No12, 40–41.

[7] Wang, F. (2019). Step input auxiliary control for a class of nonlinear systems. Journal of Wuhan Polytechnic University, 38(4), 1973–1995.

[8] Shi Zhanhua, He Lijuan, Liang Xiaoqing, & Zhang Xiaojuan. (2022). Analysis and simulation of combinational logic circuits based on Multisim10.0. ELECTRONICS WORLD.

[9] Yang, G. (2023). Optimized design of a 4-bits absolute-value detector. Highlights in Science, Engineering and Technology, 31, 224–231.

[10] Rawat, A., & Kumar, A. (2013). Designing issues associated with very large scale integrated circuits designing issues associated with very large scale integrated circuits. International Journal of Computer Science and Technology, 4.