A New Design of A 6-Bit Absolute Value Detector by Using Both VDD Scaling and Sizing to Optimize

Xuancheng Liu [1, *], Jianying Wu [2]

[1] Zhejiang University - University of Illinois Urbana-Champaign Institute, Haining, Zhejiang University, 314400, Haining, China

[2] College of Information Science and Engineering, Hohai University, Changzhou, 213022, China

* Corresponding Author Email: xuancheng.21@intl.zju.edu.cn

Abstract. The presented paper focuses on the development and optimization of a 6-bit absolute value detector, a crucial digital block in encoder and decoder applications. The primary objective is to minimize energy consumption while maintaining a delay no greater than 1.5 times the minimum delay. The optimization process combines logical effort theory with gate sizing and voltage scaling techniques. Firstly, gate sizes are adjusted while keeping VDD constant, achieving a 73% reduction in energy consumption, reaching a delay constraint of 1.5 times the minimum delay. Secondly, VDD scaling is explored without altering gate sizes, resulting in a 40% reduction in energy consumption, reaching the minimum energy consumption point. Finally, gate sizing and VDD scaling are combined, yielding a remarkable 79% reduction in total energy consumption, with a supply voltage of 0.835 volts. These optimization strategies bridge a research gap in 6-bit absolute value detectors, significantly enhancing energy efficiency and offering valuable insights for intricate digital signal processing scenarios. The findings showcase the effectiveness of combining logical effort theory with gate sizing and voltage scaling to achieve substantial energy savings while meeting stringent delay constraints.

Keywords: Absolute Value Detector, Energy Optimization, Logical Effort Theory, Voltage Scaling.

1. Introduction

The absolute value detector is a widely employed digital block within both encoder and decoder applications [1]. Contemporary research predominantly centers around 4-bit absolute value detectors, leaving a dearth of mature designs for 6-bit absolute value detectors. This paper focuses on the development of a 6-bit absolute value detector with a primary goal of minimizing energy consumption through delay optimization, aiming to maintain a delay no greater than 1.5 times the minimum delay.

The design process of this paper is divided into two distinct segments: the absolute value component and the 5-bit magnitude comparator section. In the first part, this paper employs a Karnaugh map for simplification and extends its utility by adopting a three-dimensional Karnaugh map based on a two-dimensional counterpart. This extension streamlines the output corresponding to each digit input in the absolute value processing unit. Subsequently, this paper compares the absolute value output with a threshold number, utilizing a simplified 5-bit Magnitude Comparator. This comparator yields a binary result: 1 indicating that 'a' is greater than 'b' and 0 denoting that 'b' is greater than 'a'. Throughout the design process, this paper focuses on circuit simplification to reduce energy consumption. This is achieved by substituting bulky 5-input logic gates with combinations of logic gates having no more than four inputs, thereby enhancing overall circuit performance.

Ultimately, the optimization efforts encompass VCC scaling and sizing techniques, resulting in a substantial reduction in energy consumption while adhering to the imposed delay constraint. The design and optimization of this six-bit absolute value detector bridge a notable gap in current research and offer a wider array of options for constructing intricate digital signal processing scenarios.

2. Designing circuits of absolute-value detector

2.1. Data Coding

Encoding refers to the compression of data from the original form to a smaller amount of data for convenient storage. Different encoding rules correspond to different decoding methods. If one can encode, he will also have the ability to decode [2]. In many ways of encoding and decoding, taking the absolute value and comparing the magnitude is a frequently used module, so this paper aims to design a CMOS circuit that can calculate the absolute value of the target number and compare it with the specified threshold.

2.2. Principle of 6-bit absolute-value detector

A 6-bit absolute-value detector receives 11 inputs, which consist of a 6-bit number denoted as 'A' and a 5-bit threshold labeled as 'B' (with the highest bit assumed to be zero by default). It generates a single output that signifies the outcome. When the input number 'A' is equal to or greater than the specified signal threshold 'B,' the output of the absolute-value detector is set to 1. Conversely, if the input number falls below the specified signal threshold, the output result is set to zero.

2.3. Logic gate circuit design

This paper divides its circuit into two parts to reach the function of a 6-bit absolute value detector. One is an absolute value converter that has input absolutization and the other is a comparator that could compare the absolute value with the threshold number. The fundamental diagram of the circuit is shown in Figure 1.

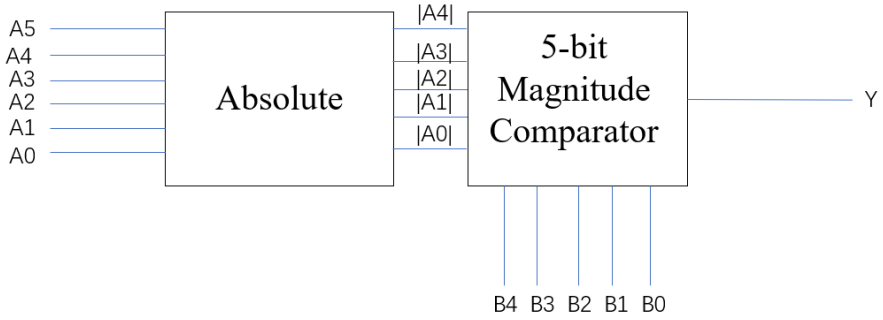

Fig. 1 Fundamental diagram of the circuit (Photo/Picture credit: Original)

2.3.1 Absolute value part

Before constructing the circuit, this paper should obtain logical expressions first. It can draw a truth table based on the principle of 6-bit absolute value detector. This paper uses 2'complement to represent a negative number. This paper chooses the Karnaugh map to simplify the logical expression of every bit and obtain the corresponding output of each absolute value bit. The six-variable Karnaugh map is represented using the three-dimensional cube in simplification which follows the same law as the two-dimensional Karnaugh map. This paper can simplify this diagram by circling the adjacent red cells, also the same applies to the squares opposite the top, bottom, left, and right [3]. The circled groups should be cubes and as large as possible. The three-dimensional Karnaugh diagram is shown in Figure 2.

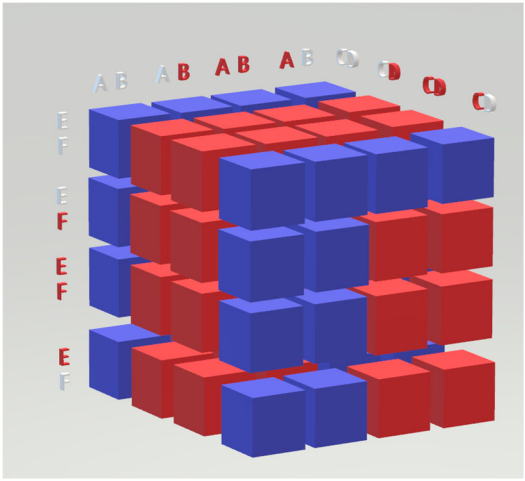

Fig. 2 Three-dimensional Karnaugh diagram (Photo/Picture credit: Original)

The picture above is a three-dimensional Karnaugh map. This paper uses the red cube to represent 1 and the blue cube to represent 0. By circling the red cubes and doing appropriate elimination, one can get a simplified logical expression of the absolute value part. For example, as shown in Figure 3, the highlighted area can be simplified as $\overline{A}\,\overline{B}\,E$.

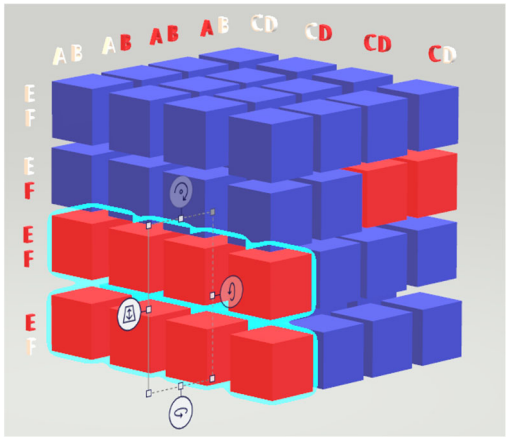

Fig. 3 An example of obtaining a logical expression in 3-D Karnaugh diagram (Photo/Picture credit: Original)

Then repeat the process repeatedly till get the simplest expression. For example, the absolute value of A_3 can be expressed as the following formula.

$$|A_3| = \overline{A_5}A_3 + A_3\overline{A_2}\,\overline{A_1}\,\overline{A_0} + A_5\overline{A_3}A_2 + A_5\overline{A_3}A_0 + A_5\overline{A_3}A_1 \quad (1)$$

By these methods, this paper gets the initial logical expression of every absolute value bit as follows.

$$|A_4| = A_5\overline{A_4} + \overline{A_5}A_4 + A_5\overline{A_3}\,\overline{A_2}\,\overline{A_1}\,\overline{A_0} \quad (2)$$

$$|A_2| = \overline{A_5}\,A_2 + A_2\overline{A_1}\,\overline{A_0} + A_5\overline{A_4}\,\overline{A_3}\,\overline{A_2} + A_5\overline{A_2}A_0 + A_5\overline{A_2}A_1 \quad (3)$$

$$|A_1| = A_1\overline{A_0} + \overline{A_5}A_1 + A_5\overline{A_1}A_0 \quad (4)$$

$$|A_0| = A_0 \quad (5)$$

From above expressions, the circuit of this part can be constructed shown as Figure 4. The circuit in Figure 4 is improved by replacing the 4 input and 5 input gates, which will be discussed later.

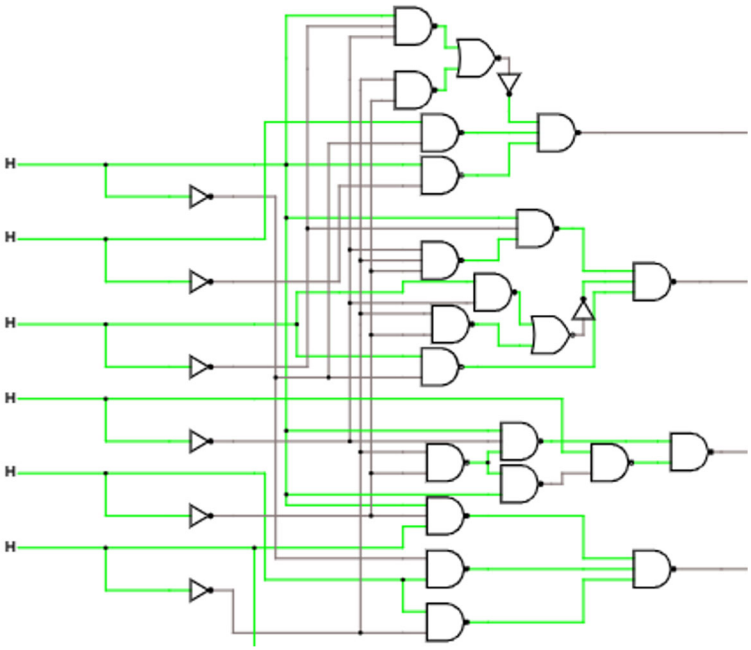

Fig. 4 The circuit of absolute value part (Photo/Picture credit: Original)

2.3.2 Magnitude Comparator part

The absolute value and the threshold number will be compared and if $|A_4|$ is the same as B_4, the second layer, a NOR gate will pass 1 to the third AND gate layer. Every bit will be compared at the same time. Once different comparison results are produced, the NOR gate will pass 0 to the AND gate layer, which will cause every bit lower than that cannot influence the result, and the result(A>B) will be determined by that different bit. If $|A_n| \geq B_n$, the result is 1, if $|A_n| \leq B_n$, the result is 0. Based on this idea, build a simple and clear logic gate circuit as Figure 5. It is clear that for the last layer of the OR gate and for the lowest bit there is an AND gate with five inputs.

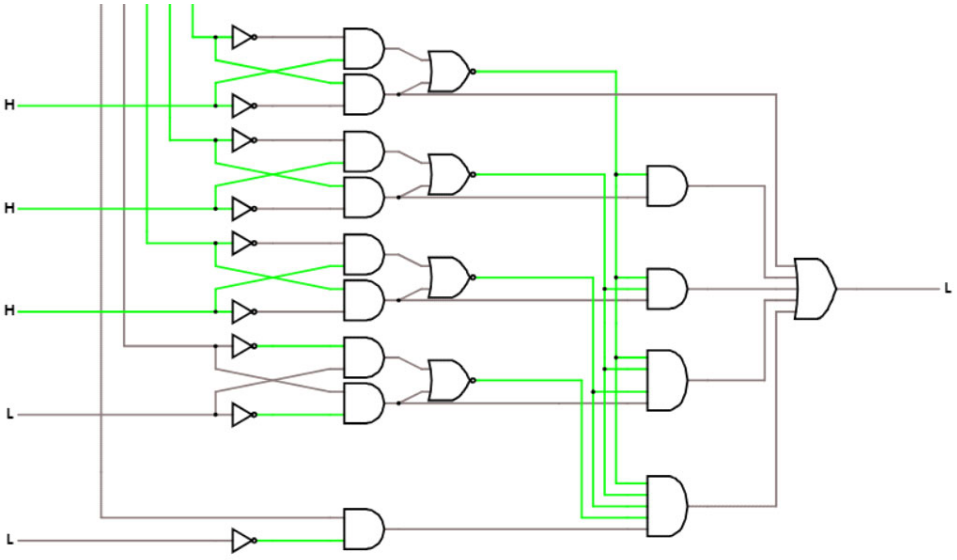

Fig. 5 The original circuit of the comparator part (Photo/Picture credit: Original)

The above circuit is based on the following logical expressions.

$$\text{Output}(A > B) = A_4\overline{B_4} + X_4A_3\overline{B_3} + X_4X_3A_2\overline{B_2} + X_4X_3X_2A_1\overline{B_1} + X_4X_3X_2X_1A_0\overline{B_0} \quad (6)$$

$$X_i = A_iB_i + \overline{A_i}\,\overline{B_i} \quad (7)$$

2.4. Circuit simplification

Firstly, when building the circuit, this paper simplifies the individual expressions to make every gate have inputs less than four and use only the basic logic gates such as the NAND gate, NOR gate, and inverter in circuit design. In this way, the paper can reduce the number of layers of transistors, because the AND gate and OR gate cost more layers of transistors in the CMOS technique contributing to a larger delay.

Secondly, after building the original circuits, it is found that there are some 4-input and 5-input logic gates. These logic gates will contribute to a much larger delay than 2 or 3 input gates because they will cost larger logic effects. Therefore, this paper replaces 4-input and 5-input gates with 2 or 3-input gates, which is shown in Figure 6.

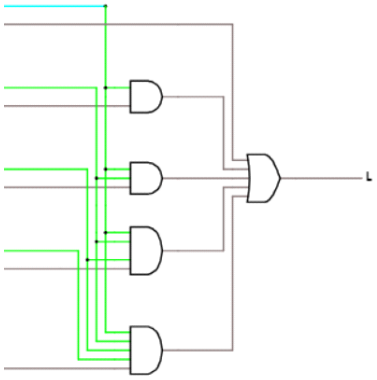

Fig. 6 A part of circuit in the comparator part (Photo/Picture credit: Original)

This paper breaks 4 and 5-input gates into several 2 and 3-input gates and utilizes De Morgan's laws to replace Figure 7 with a NAND gate. In the end, this paper obtains the result as shown in Figure 8.

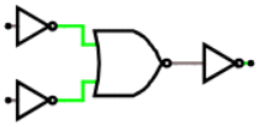

Fig. 7 A small part can be replaced by a NAND gate (Photo/Picture credit: Original)

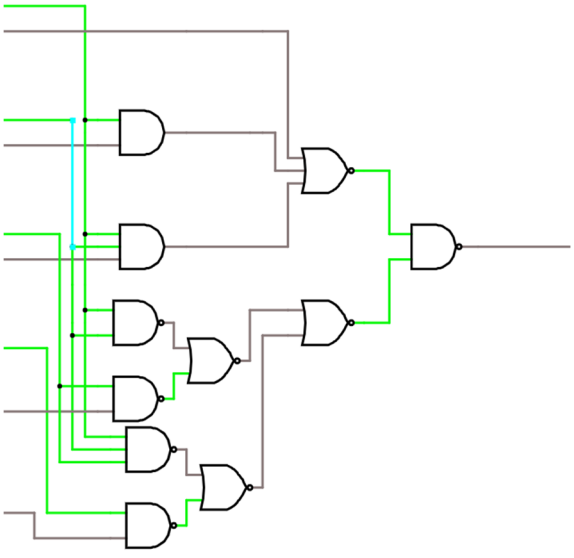

Fig. 8 A part of circuit in the comparator part after simplification (Photo/Picture credit: Original)

Therefore, it gets the whole circuit of 6-bit absolute value detector as shown in Figure 9.

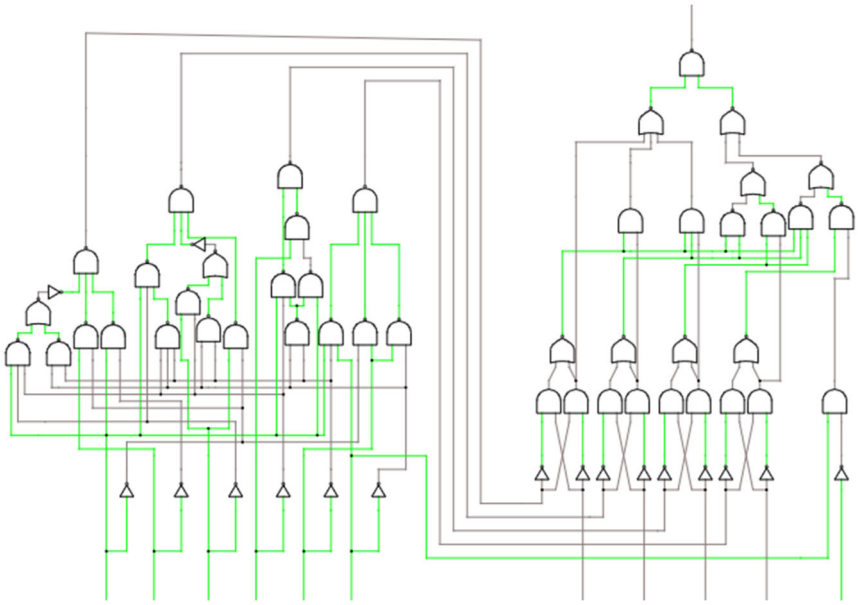

Fig. 9 The complete circuit of 6-bit absolute value detector (Photo/Picture credit: Original)

3. Calculation and optimization

3.1. The critical path of design

The path having the most delay throughout the entire design is called the critical path [4]. In design, the path with most logic gates costs maximum delay as shown in Figure 10.

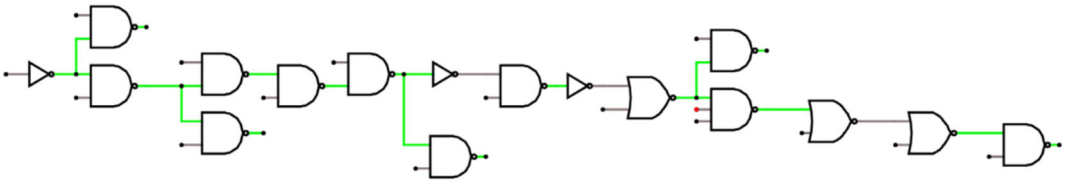

Fig. 10 Critical path (Photo/Picture credit: Original)

There are 13 stages in the critical path with 4 branches.

3.2. Minimum delay and Maximum energy consumption

The concept of logical effort serves as a fundamental guideline in design, aimed at optimizing both delay and energy consumption [5]. This theory's effectiveness lies in its allocation of logical effort values to different types of logic gates, indicating their respective drive capabilities relative to a reference inverter. For each logic gate, two critical parameters are considered: logical effort and parasitic delay [6]. Specifically, the logical effort of a given logic gate is quantified as the ratio between its input capacitance and that of an inverter capable of delivering an equivalent output current [7]. By changing the width of transistors, this paper can achieve both equal resistance and output currents of the gate and reference inverter.

$$g = \frac{R_{gate}C_{in,gate}}{R_{INV}C_{in,INV}} = \frac{C_{in,gate}}{C_{in,INV}} \tag{8}$$

The parasitic delay is the ratio of the parasitic capacitance to the input capacitance of the inverter.

$$p = \frac{C_{par,gate}}{C_{par,INV}} \tag{9}$$

In this critical path, it has 4 kinds of logic gates. Their logical effort and parasitic effort are shown in Table 1.

Table 1. Logical effort and parasitic effort

Logic gate	Inverter	2 input NAND	2 input NOR	3 input NAND
g	1	1.398	1.602	1.796
p	1	2	2	3

With the above logical and parasitic effect, the paper moves on to calculate path logical effort and path electrical effort thereby obtaining path effort. The also considers the contribution of the branch effect to path effort. Note the output bit is loaded with C_L = 32 unit-sized inverters.

$$G = \sum g \tag{10}$$

$$H = \frac{C_{load}}{C_{in}} \tag{11}$$

$$b = \frac{C_{onpath} + C_{offpath}}{C_{onpath}} \tag{12}$$

$$B = \sum b \tag{13}$$

$$F = GHB \tag{14}$$

G is path logical effort and H is path electrical effort, b is the branching effect of a single gate and B is the path branching effort [8]. The path effort F is 30090.169. Finally, this paper calculates the minimum delay using the following formulas [9].

$$N = 13 \tag{15}$$

$$f = \sqrt[N]{F} \tag{16}$$

$$D = Nf + \sum p \tag{17}$$

N indicates the number of stages; Stage effort f is 2.211; Minimum delay D is 52.743. To compute energy consumption, this paper needs to calculate the sizing of each gate in the critical path.

$$C_{in} = \frac{gC_{out}}{f} \tag{18}$$

Results are shown in Table 2.

Table 2. Size of each gate

Stage	1	2	3	4	5	6	7	8	9	10	11	12	13
Size	1.000	2.205	1.743	1.379	2.180	3.448	3.179	5.028	11.118	15.345	10.622	14.660	20.233

When delay reaches its minimum, energy consumption reaches its maximum. The paper now can compute maximum energy consumption using the following formulas [10]. This paper ignores switching probability here to simplify computation and assume V_{DD}. Assume gamma (Cparasitic / Cgate) = 1 for delay modeling.

$$E_i = (\gamma s_i + s_{i+1})V_{DD}^2 \tag{19}$$

$$E = \sum E_i \tag{20}$$

E_i indicates the energy consumption of the ith logic gate and E is the total energy consumption, which is 215.28.

3.3. Optimization by sizing with fixed V_{DD}

For optimization in this paper, this paper tries to minimize energy consumption and keep the delay of the circuit smaller than 1.5 times the minimum delay. With the size of gates changing, the capacitance of gates will also change causing the energy consumption of the critical path to change.

To calculate the size of each gate and the energy consumption, the paper utilizes a powerful tool—Excel Solver—to solve this nonlinear programming problem. This paper sets the size of each gate as a variable and limits delay to greater than minimum delay and no more than 1.5 times minimum delay. V_{DD} is fixed as 1v. Table 3 shows the result of the computation.

$$d_i = g_i \frac{s_{i+1}}{s_i} + p_i \tag{21}$$

d_i indicates the delay of the ith logic gate.

Table 3. Results after sizing

Stage	Size	Delay	Energy
1	1	2	2
2	1	3.398	2
3	1	3.398	2
4	1	3.398	2
5	1	3.398	2
6	1	2	2
7	1	3.398	2
8	1	2	2
9	1	3.602	2
10	1	4.796	2
11	1	3.602	2
12	1	4.304	2.438
13	1.438	33.110	33.438

By only sizing, the energy consumption is reduced to 57.876 and the delay reaches its constraint, which is 1.5 times the minimum delay. Compared to the case without sizing, the energy consumption is reduced by 73%.

3.4. Optimization by scaling V_{DD} without sizing

From formula 19, the energy consumption is directly affected by the magnitude of V_{DD}. Therefore, the energy consumption can be minimized by reducing V_{DD}. However, reducing V_{DD} will contribute to a larger delay. In this case, this paper assumes that delay is proportional to VDD as VDD / $(VDD - VT)^2$ in order to find optimal VDD. Assume VT = 0.2V. This paper gradually reduces the voltage from 1v and calculates the delay and energy then plots a graph to show their tendency in Figure 11.

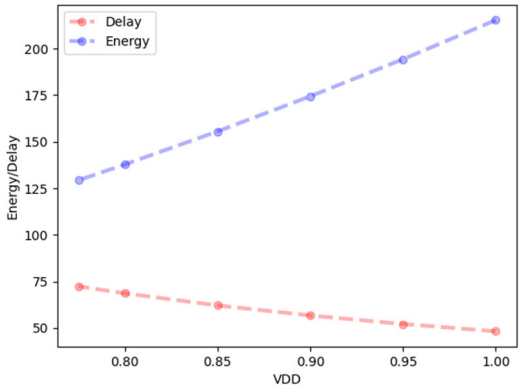

Fig. 11 Results of scaling VDD (Photo/Picture credit: Original)

According to Figure. 11, when V_{DD} is reduced to 0.775, the delay reaches 1.5 times the minimum delay. At this time, the energy consumption is 129.3, which is reduced by 40%.

3.5. Optimization by both sizing and scaling V_{DD}

Based on previous computations, it is known that both sizing and scaling V_{DD} can contribute to lower energy consumption. Therefore, this paper moves on to minimizing energy consumption by using both methods. Combining the relationship between supply voltage and delay and the relationship between sizing and delay, the paper derives a new expression for delay.

$$D = \sum \frac{V_{DD}}{1.5625(V_{DD} - V_T)^2} \left(g_i \frac{s_{i+1}}{s_i} + p_i \right) \tag{22}$$

From the above formula, when V_{DD} is 1v, delay exactly equals $g_i/s_{i+1}+p_i$, which conforms to the formula in 3.3. Additionally, the expression of the delay satisfies the previous assumption that delay is proportional to VDD as VDD / (VDD – VT)^2. Using a similar method in Excel Solver, the data is shown in the following table 4.

Table. 4 Results after both sizing and scaling V_{DD}

Stage	Size	Delay	Energy
1	1.207	2.425	1.538
2	1	4.506	1.393
3	1	4.506	1.393
4	1	4.506	1.393
5	1	4.506	1.393
6	1	2.652	1.393
7	1	4.506	1.393
8	1	2.652	1.393
9	1	4.777	1.393
10	1	6.361	1.393
11	1	5.883	1.756
12	1.520	8.653	4.051
13	4.294	16.468	25.284

This paper treats the whole process of optimization as scaling V_{DD} before sizing because they are independent of each other. The graph reflecting the impact of these two processes is shown in Figure 12.

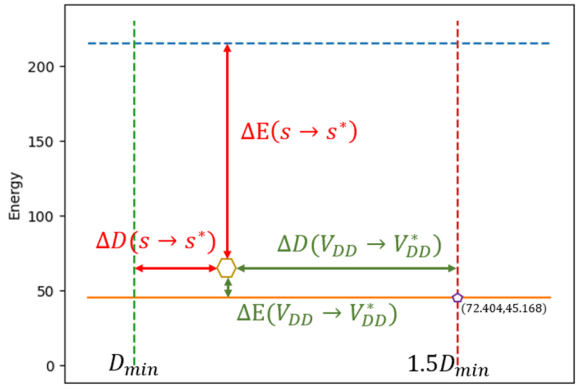

Fig. 12 Picture of both sizing and scaling V_{DD} (Photo/Picture credit: Original)

It can be found sizing exerts a larger impact on reducing energy consumption but less impact on increasing delay. Scaling V_{DD} exerts a larger impact on increasing delay but less impact on reducing energy consumption. The total energy consumption is 45.168 achieving a reduction of 79% when the supply voltage V_{DD} is 0.835v.

4. Conclusion

For the process of optimization, the paper utilizes the logical effort theory to simplify the calculation of delay and energy consumption. This paper first changes the size of each gate without changing VDD to find the minimum energy consumption when the delay is not larger than its upper bound. In this case, the energy consumption is reduced to 57.876 and the delay reaches its constraint, which is 1.5 times the minimum delay. Compared to the worst case without using any optimized method, the energy consumption is reduced by 73%. Secondly, the paper tries to scale VDD without changing the size of each gate. For this case, the minimum energy consumption is reduced to 129.3 achieving a reduction of 40%. In the end, this paper combines sizing and scaling VDD together to chase a lower energy consumption. The total energy consumption is 45.168 achieving a reduction of 79% successfully when the supply voltage VDD is 0.835v.

Though this paper optimizes the energy consumption of the circuit, there are still some flaws and limitations in this paper. Firstly, all of the calculation is based on the logical effort model. However, there is a deviation between the model and the reality. The property of a transistor with different widths over lengths may be eminently different than the theoretical values. In the actual optimization, the size of each gate must depend on measuring results. Secondly, this paper does not consider the impact of the material of the device on energy consumption and delay. In real life, choosing to use new kinds of MOSFETs rather than the traditional ones is more optimal. For instance, utilizing SiC-Mosfet will be beneficial to chasing Low on-off loss.

Authors Contribution

All the authors contributed equally, and their names were listed in alphabetical order.

References

[1] Lloyd R. Cognitive maps: Encoding and decoding information. Annals of the Association of American Geographers, 1989, 79(1): 101-124.

[2] Blair E. A reflexive exploration of two qualitative data coding techniques. Journal of Methods and Measurement in the Social Sciences, 2015, 6(1): 14-29.

[3] Rushdi A M. Improved variable entered Karnaugh map procedures. Computers & electrical engineering, 1987, 13(1): 41-52.

[4] Bowman K A, Tang X, Eble J C, et al. Impact of extrinsic and intrinsic parameter fluctuations on CMOS circuit performance. IEEE Journal of solid-state circuits, 2000, 35(8): 1186-1193.

[5] Pappu A M, Apsel A B. Analysis of intrachip electrical and optical fanout. Applied Optics, 2005, 44(30): 6361-6372.

[6] Oklobdzija V G, Zeydel B R, Dao H, et al. Energy-delay estimation technique for high-performance microprocessor VLSI adders. Proceedings 2003 16th IEEE Symposium on Computer Arithmetic. IEEE, 2003: 272-279.

[7] Sutherland I, Sproull R F, Harris D. Logical effort: designing fast CMOS circuits. Morgan Kaufmann, 1999.

[8] Shihab E, Bird C, Zimmermann T. The effect of branching strategies on software quality. Proceedings of the ACM-IEEE international symposium on Empirical software engineering and measurement. 2012: 301-310.

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

[10] Kinget P R. Scaling analog circuits into deep nanoscale CMOS: Obstacles and ways to overcome them. 2015 IEEE Custom Integrated Circuits Conference (CICC). IEEE, 2015: 1-8.