Realization and optimization of a new spike detection algorithm logic circuit

Yuefeng Gao1,†, Qingxing Liu2,†, Hengji Ye3,*,†

1 College of Information Sciences and Technology, Beijing University of Chemical Technology, Beijing 100000, China.
2 Changkong college, Nanjing University of Aeronautics and Astronautics, Nanjing 320100, China.
3 School of Optical and Electronic Information, Huazhong University of Science and Technology, Wuhan 430074, China
*Corresponding author email: 2019040306@mail.buct.edu.cn
†These authors contributed equally.

Abstract. At present, artificial intelligence has increasingly become the most promising and essential major. The interdisciplinary cooperation between the integrated circuit and artificial intelligence brings infinite possibilities. In the convolutional neural network area, in order to achieve the valuable output, the comparison between values is often encountered. The output is obtained by comparison between the result after the convolution calculation with the designed threshold. Therefore, our team not only design a 4-bit binary comparator hardware logic circuit to complete the task but also discuss and verify the feasibility and performance characteristics of the program from the perspective of energy and time delay. As for the overall framework, we design a convenient circuit that converts the complement code into the original code and chooses a CLA adder to accomplish this part. Using this kind of adder can ideally help us reduce the time delay at the expense of a complicated circuit schematic. In the comparator part, we design a high-quality circuit framework. The strategy of our circuit is to compare the relationship between the four-bit binary code and the threshold bit by bit from MSB to LSB, which performs better than the original 4-bit comparator, and we design two outputs that can legibly illustrate the relationship between two values. We use logic effort to discuss the normalized delay in our project. Besides, we find the connection between the energy and the delay by calculation. Finally, we design a trade-off function to make the optimization of energy and delay together with respect to voltage.

1. Introduction
Nowadays, the rapid development of electronic technology has brought tremendous changes to human life. This is an emerging technology that began to develop in the late 19th century and early 20th century [1]. In the twentieth century, electronic technology developed the fastest and most widely used, becoming an important symbol of the development of modern science and technology [2].

The first generation of electronic products is centered on electron tubes. At the end of the 1940s, the world’s first semiconductor transistor was born [3]. It was quickly used in various countries for its compactness, lightweight, power saving, and long life, replacing the electronic tube in a wide range. In the late 1950s, the world’s first integrated circuit appeared [4]. It integrated many electronic components
such as transistors on a silicon chip, making electronic products more compact. Integrated circuits have rapidly developed from small-scale integrated circuits to large-scale integrated circuits and ultra-large-scale integrated circuits so that electronic products are moving towards high-efficiency, low-consumption, high-precision, high-stability, and intelligence.

Among the integrated circuits, the most outstanding is the metal-oxide-semiconductor (MOS) integrated circuit. This is an integrated circuit composed of MOS field-effect transistors as the main components, referred to as MOSIC [5]. In 1964, the insulated gate field effect transistor was developed. It was not until 1968 that the stability of MOS devices was solved.

MOS integrated circuit is a commonly used integrated circuit. The smallest unit is an inverter, which consists of two metal-oxide-semiconductor field-effect transistors. In addition, the MOS tube is particularly easy to produce AND gates and NOT gates in digital logic circuits and has special features such as high speed and simple structure [5]. This kind of integration process is mainly used for the manufacture of digital integrated circuits, and the degree of circuit integration can be very high.

The electronic adder is a digital circuit component used to perform additional operations and is the basis of the arithmetic logic unit in the core microprocessor of an electronic computer. In these electronic systems, the adder is mainly responsible for calculating the address, index, and other data. In addition, the adder is also an important part of some other hardware, such as the binary number multiplier. Personal digital assistants and notebook computers require ultra-large-scale integrated circuits, as well as ultra-large-scale integrated designs with improved power delay characteristics [6]. The adder is one of the most basic cornerstones in all the above-mentioned circuit applications. Over the years, its optimization has been particularly important, with the main goal of reducing energy consumption and time delay and reducing structural complexity. Among them, there are generally two optimization schemes: 1. Logic layer optimization, rearranging a Boolean equation to obtain a circuit with a faster speed or a smaller area. 2. Circuit layer optimization, changing the size of the transistor or the topology of the circuit [7].

The more traditional CMOS full adders now have better indicators for voltage scaling and can provide full swing output. But the disadvantage is that this method will have a larger area and slower speed due to the larger input capacitance [8]. In order to better understand the advantages and disadvantages of different types of adders, this article compares the power and delay between different types of full adder circuits so that we can have a more comprehensive understanding of some problems of adders [9].

At present, a more popular optimization method is to carry out the comprehensive and detailed design for each subsystem, which helps to improve the overall performance of complex circuits not only directly but also easily realizes the arithmetic operations in the ultra-large integrated-scale integrated circuit system [10]. In addition, size optimization is also the focus of optimization of the MOS adder. This paper proposes a new way to reduce power consumption and demonstrates that reducing area is an effective way to reduce power consumption [11].

Integrated circuits also play an important role in artificial intelligence, and one of the more commonly used applications is to design a circuit to detect spikes from the nervous system. The algorithm is a commonly used spike detection algorithm, also known as absolute value detection. Its basic diagram is shown in Fig.1. Here x[n], Thr, and Output respectively represent the detected value, the preset value (threshold), and whether it is recognized as valid (1 means valid, 0 means invalid).

The simple idea is that after the signal is converted into a digital signal, it is first corrected by absolute value and compared with a preset threshold. If it is greater than the threshold, it means that a spike signal is detected. And in order to enable the device to work for a long time to detect the nerve impulse of patients who have a mental illness such as violent fever, we need to reduce its power consumption as much as possible; to make the device sensitive to nerve impulse, we also hope to reduce its detection time (also known as delay time). We plan to use an adder and a comparator to accomplish this goal, reduce energy consumption, and delay time through a series of measures.
2. Mechanism

2.1 Optimization of 2’s Complementary Calculation

Because the input is a 4-bit binary number in complement form, in order to make the comparison between our input and the threshold number, we need to convert the complement code into the original form.

There are two cases of the input. For convenience, we use $A_3 \ A_2 \ A_1 \ A_0$ (4-bit, from MSB to LSB) to describe our information:

If $A_3 = 0$, which means that the input itself is positive), then the original form of the input remains the same.

If $A_3 = 1$, which means the input itself is negative), then the original form of the input is to make the bitwise negation and plus 1.

To this end, our team designs a practical strategy that can effectively avoid the discussion about whether $A_3$ is positive or negative. Our team decided to make XOR between $A_3$ and $A_2$, $A_1$, $A_0$ in turn, and then add $000A_3$. We can use the following formula to express 1’s complement:

$$A'_i = A_3 \oplus A_i \quad (i = 0,1,2,3)$$

(1)

As we can see, if $A_3 = 1$, it means that the 4-bit binary number is less than zero. We assume that the input's complement is 1001, and the number obtained after the complement conversion operation is 0110. According to our strategy, XOR $A_3$ and $A_3A_2A_1A_0$ respectively, the number obtained is 0110, plus $000A_3$, which is 0001, and the final result is also 0110. If $A_3 = 0$, that is to say, this 4-bit binary number is greater than zero. After the operation of converting the complement to the original code, its complement form is the same as itself. As shown in the figure, we take 0011 as an example, and the number obtained after the operation is still 0011. Using our strategy, we perform exclusive OR operations separately, plus $000A_3$, which is 0000, and the final result are also 0011. Our strategy avoids the discussion of $A_3$ value and saves hardware resources. The 2’complement can be expressed as follows:

$$A''_3A''_2A''_1A''_0 = A'_3A'_2A'_1A'_0 + 000A_3$$

(2)

The final result is shown in Tab.2 below:

| Cases | $A_3A_2A_1A_0$ | $A'_3A'_2A'_1A'_0$ |
|-------|----------------|--------------------|
| $A_3 = 1$ | 1001 | 0110                   |
| $A_3 = 0$ | 0011 | 0011                   |
2.2 **Optimization of carrying Lookahead Adder**

In order to achieve great performance in the time delay calculation, we choose the Carry Lookahead Adder (CLA) rather than the Ripple Carry Adder (RCA). We inspect these two adders from various perspectives.

We made the comparison between the CLA adder and the RCA adder in Tab.3.

| Advantages and Disadvantage | RCA                                       | CLA                                      |
|-----------------------------|-------------------------------------------|------------------------------------------|
| Structural Features         | $C_{\text{out}}$ of the low full adder is connected to the high full adder $C_{\text{in}}$ | The $C_{\text{in}}$ of each full adder comes from the logic of the advanced bit |
| Advantage                   | Simple circuit layout and convenient design | The delay is fixed as a three-level gate delay, has nothing to do with the number of bits |
| Disadvantage                | The high-order must wait for the low-order operation to complete, and the delay time is long | If the number of bits of the adder is further widened, the circuit becomes complicated |

And this is the schematic of our logic circuit, and we add two intermediate variables to accelerate the calculation. Finally, we design this adder based on the Boolean expression optimization of the complement calculation and the concept of carrying lookahead adder as shown in Fig.2. Equation (3) is the Boolean expression of the adder.

![Fig.2 Schematic of Adder Logic Circuit](image-url)
\[ C_1 = G_0 + P_0 \cdot C_0 \]
\[ C_2 = G_1 + P_1 \cdot C_1 \]
\[ = G_1 + P_1 \cdot (G_0 + P_0 \cdot C_0) \]
\[ C_3 = G_2 + P_2 \cdot C_2 \]
\[ = G_2 + P_2 \cdot (G_1 + P_1 \cdot G_0 + P_1 \cdot P_0 \cdot C_0) \]
\[ C_4 = G_3 + P_3 \cdot C_3 \]
\[ = G_3 + P_3 \cdot (G_2 + P_2 \cdot G_1 + P_2 \cdot P_1 \cdot G_0 + P_2 \cdot P_1 \cdot P_0 \cdot C_0) \]

In these formulas, \( P_i = A_i'' + B_i \), \( G = A_i'' \cdot B_i \).

2.3 Optimization of Comparator

The traditional 4bit comparator is obtained by simplifying the design based on the truth table. The strategy of the 4bit comparator designed by our team is to compare the MSB to the LSB. If the high-value bit can be judged, the subsequent low Numerical bit comparison does not need to be carried out, which maximizes the speed of operation and reduces the time delay. According to our ideas, we can get the following logic circuit diagram as shown in Fig.3.

![Fig.3 Schematic of Comparator Logic Circuit](image_url)
We can see that there are two outputs in the entire circuit diagram. We call them EQ (equal) and GR (greater than). When EQ=1, the size of the 4-bit binary input representing the entire circuit is equal to the threshold voltage. At this time. In this case, GR must be 0. When EQ=0, which means that the 4-bit input signal and the threshold voltage of the entire circuit are not equal. At this time, when GR=1, it means that the 4-bit input signal is greater than the threshold voltage. When GR=0, it means that the 4-bit input signal is less than the threshold voltage. It can be seen from the circuit schematic that the entire system is similar in form from top to bottom so that we can observe from $A_2B_2$ to examine the feasibility and logic of the entire circuit. The figure below shows the different results represented by different values of $A_i$ and $B_i$. For the GT signal, when $A_3B_3 = 10$, through multiple logic gates, the GT signal can be directly determined to be 1, and it has nothing to do with $A_2B_2$, $A_1B_1$, and $A_0B_0$. When $A_3B_3 = 10$ or $10$, the judgment of this layer of the circuit is equivalent to invalid, continue to look at the relationship between $A_2$ and $B_2$ in the next layer. When $A_3B_3 = 01$, the GT signal can be determined to be 0 after passing through the logic gate, and the EQ signal is also 0. The whole circuit system is compared from the high position, which saves the circuit resources and reduces the circuit delay. In the end, we can get four situations, as shown in Tab.4.

| $A_iB_i$ | Case     |
|---------|----------|
| 00      | Equal    |
| 01      | Less than|
| 10      | Greater than|
| 11      | Equal    |

### 2.4 Logic effort

We use the concept of logic effort to get the normalized delay of our Absolute-Value Detector. Compared with the traditional way of delay calculating, the logic effort can normalize the delay to a fictitious time constant, which is more practical and convenient for examining the performance of the circuit.

We usually use formula (4) to calculate delay.

$$t_p = 0.69 R_g C_{in,gate} \left( \frac{C_{par,gate}}{C_{in,gate}} + \frac{C_{out}}{C_{in,gate}} \right)$$

We make the following definition

$$\tau_{gate} = 0.69 R_g C_{in,gate}$$

$$\gamma_{gate} = \frac{C_{par,gate}}{C_{in,gate}}$$

So that we can have formula (7):

$$\frac{\text{Delay}}{\tau_{INV}} = \frac{\tau_{gate}}{\tau_{INV}} (\gamma_{gate} + \frac{C_{out}}{C_{in,gate}})$$

To simplify this formula, we have

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

$$h = \frac{C_{out}}{C_{in,gate}}$$

$$p = \frac{C_{par,gate}}{C_{par,INV}} \cdot \gamma_{INV}$$

Where $g$ means logic effort, $h$ means electrical fanout, $p$ means parasitic delay, $d$ means normalized delay.

Then we have the normalized delay in formula (11):
\[ d = gh + p \gamma_{INV} \]  
\[ (11) \]

Here, we assume that \( \gamma_{INV} \) is equal to 1, then we get:
\[ d = gh + p \]  
\[ (12) \]

To calculate the logic effort of each gate, first, we need to choose an input of the gate, then find the total device width driven by that input, Find \( W_P \), the pull-up device width of a single device that has equivalent drive strength as a gate’s pull-up of that input. For a reference inverter with Equal Rise/Fall, we determine the total gate widths of the inverter devices with \( W_P \), finally divide the total device width by total gate widths to get \( g \).

When we applied logic effort to inverter chain as shown in Fig.4:

\[ t_{p,j} = R_{gate,unit} C_{in,gate,unit} \left( \gamma + \frac{C_{in,j+1}}{C_{in,j}} \right) \]  
\[ (13) \]

\[ t_p = \sum_{j=1}^{N} t_{p,j} = \tau \sum_{j=1}^{N} \left( \gamma + \frac{C_{in,j+1}}{C_{in,j}} \right), C_{in,N+1} = C_L \]  
\[ (14) \]

To find minimize delay, we find N-1 partial derivatives, we have
\[ \frac{C_{in,j+1}}{C_{in,j}} = \frac{C_{in,j}}{C_{in,j-1}} \]  
\[ (16) \]

When each stage is sized by \( f \) and has same fanout \( f \),
\[ f^N = F = \frac{C_L}{C_{in,1}} \]  
\[ (17) \]

Effective fanout of each stage:
\[ f = \frac{N}{\sqrt{F}} \]  
\[ (18) \]

And the minimum path delay is:
\[ t_p = N \tau (\gamma + \sqrt{\frac{N}{F}}) \]  
\[ (19) \]

So in the generalization for multistage networks, we have minimum path delay:
\[ D_{min} = \sum (g_i \cdot h_i + p_i) = N \cdot f + P \]  
\[ (20) \]

Path Effort Delay:
\[ D_F = \sum_i f_i \]  
\[ (21) \]

Path Parasitic Delay:
\[ P = \sum_i p_i \]  
\[ (22) \]

Total Path Delay:
\[ D = \sum_i d_i = D_F + P \]  
\[ (23) \]

If the circuit has a branch, we need to take branching effort into consideration:
\[ b = \frac{C_{on-path} + C_{off-path}}{C_{on-path}} \]  
\[ (24) \]
And we have path branching effort:

$$B = \prod b_i$$  \hspace{1cm} (25)

We can change formula (18) to:

$$\text{Path effort} : F_{\text{path}} = G_{\text{path}} \cdot H_{\text{path}} \cdot B_{\text{path}}$$  \hspace{1cm} (26)

2.5 Trade-off between Energy Consumption and Delay time

According to the previous analysis, we know that when $V_{DD}$ increases, the energy consumption will increase, and the delay time will decrease simultaneously. In order to explore the optimal trade-off relationship between delay and energy, I intend to construct an optimization function (also known as a trade-off function) from the perspective of mathematical analysis to find the optimal voltage.

The steps are as follows:

- Construct an appropriate trade-off function;
- Normalize the variables that need to be measured;
- Find the domain of the independent variables (possibly multiple) of the trade-off function;
- Using the optimization method to solve the maximum value of the trade-off function (the function of one variable is to find the extreme point).

We use the average trade-off function:

$$y = \sqrt{(E' - E)^2 + (D' - D)^2}$$  \hspace{1cm} (27)

Where $E'$ and $D'$ are respectively the energy and delay after normalization. The normalization formula is as follows:

$$x' = \frac{x}{\max(x) - \min(x)}$$  \hspace{1cm} (28)

The energy expression is:

$$E = CV_{DD}^2$$  \hspace{1cm} (29)

The delay expression is:

$$D = \frac{C}{I_D} = \frac{CV_{DD}}{I_D} = \frac{CV_{DD}}{k'\frac{W}{L}(V_{DD} - V_T)^2} = \frac{kV_{DD}}{(V_{DD} - V_T)^2}$$  \hspace{1cm} (30)

When $V_{DD} = 0.78V$, $\min(E) = 0.6084C$; When $V_{DD} = 1V$, $\max(E) = C$.

The normalized energy is:

$$E' = \frac{E}{\max(E) - \min(E)} = \frac{CV_{DD}^2}{C - 0.6084C} \approx 2.5V_{DD}^2$$  \hspace{1cm} (31)

The delay is a single valued function of the voltage, so we can get the derivative as follows:

$$\frac{dD}{dV_{DD}} = -k\frac{V_{DD} + V_T}{(V_{DD} - V_T)^3} < 0$$  \hspace{1cm} (32)

When $V_{DD} = 1V$, $\max(D) = 1.5625k$; When $V_{DD} = 0.78V$, $\min(E) = 3k$.

The normalized delay is:

$$D' = \frac{D}{\max(D) - \min(D)} = \frac{kV_{DD}}{(V_{DD} - V_T)^2} \approx \frac{0.7V_{DD}}{(V_{DD} - V_T)^2}$$  \hspace{1cm} (33)

The average values of energy and delay are as follows:

$$\bar{E} = \int_{V_{DD} = 0.78}^{1} V_{DD}^2 dV_{DD} \approx 0.175 \div 0.22 = 0.795$$  \hspace{1cm} (34)

$$\bar{D} = \int_{V_{DD} = 0.78}^{1} \frac{0.7V_{DD}}{(V_{DD} - V_T)^2} dV_{DD} \approx 0.2914 \div 0.22 = 1.32$$  \hspace{1cm} (35)

The final trade-off function is as follows:
\[ y = \sqrt{(2.5V_{DD}^2 - 0.795)^2 + \left(\frac{0.7V_{DD}}{(V_{DD} - V_T)^2} - 1.32\right)^2} \]  

(36)

3. Result

3.1 Energy and Delay

We have the following results through the optimized calculation of logical efforts, as shown in Tab.5.

| Variable                                      | Result                  |
|-----------------------------------------------|-------------------------|
| Final Delay [FO4(1V)]                        | 7031.25\(C_L\)         |
| Final Energy [E_unit(1V)]                     | 19.47\(C\) (inverter)  |
| Critical Path Delay at 1V                     | 93.5                    |
| The energy at 1V and Min Delay                | 32\(C\)(inverter) 7031.25\(C_L\) |

3.2 Calculation of \(V_{PD}\)

We use the delay of the basic RC network to characterize the time delay model, and the equivalent resistance \(R_{eq}\) can be calculated like this.

\[ R_{eq} = \frac{1}{2} \frac{V_{DD}}{V_{PD}} \int \frac{V}{(1 + yV)^2} dV \approx \frac{3}{4} \frac{V_{DD}}{I} \]

(37)

In this formula, we assume \(y = 0\).

And in the saturation region, we can calculate the current in the formula,

\[ I = \frac{1}{2} \mu n C_{ox} \frac{W}{L} (V_{DD} - V_T)^2 \]

(38)

So, we bring this equation in the last equation, and we can achieve the time delay \(t_p\)

\[ t_p = 0.69R_{eq}C_L = 0.69 \times \frac{3}{4} \frac{C_L V_{DD}}{V_{PD}^2 V_T} \sim \frac{kV_{DD}}{(V_{DD} - V_T)^2} \]

(39)

3.3 Calculation of Sizing

According to the calculation method of intrinsic delay, we can get the following formulas:

\[ t_{p0} = 0.69R_{eq}C_{int} \]

(40)

\[ t_p = 0.69R_{eq}(C_{int} + C_{ext}) = t_{p0} \left(1 + \frac{C_{ext}}{C_{int}}\right) \]

(41)

We define sizing parameter \(S\), which is a constant proportional to the size of the MOS tube. Generally, we think that the equivalent capacitance of a MOS tube is proportional to its size area, and the equivalent resistance is inversely proportional to its area, so we can get equations as follows:

\[ C_{int} = SC_{iref} \]

(42)

\[ R_{eq} = \frac{R_{eq}}{S} \]

(43)

\[ t_p = 0.69(R_{eq}/S)(SC_{iref}) \left(1 + \frac{C_{ext}}{SC_{iref}}\right) = 0.69R_{eq}C_{iref} \left(1 + \frac{C_{ext}}{SC_{iref}}\right) = t_{p0} \left(1 + \frac{C_{ext}}{SC_{iref}}\right) \]

(44)
As a result, when $S$ increases, delay decreases. When $V_{DD}$ decreases, delay increases. In addition, the intrinsic delay is:

$$t_{po} = \frac{Q}{I} = \frac{C V_{DD}}{I} = \frac{C V_{DD}}{K \frac{w}{L} (V_{DD} - VT)^2} \approx k V_{DD}$$

(45)

Next, we get the following formulas in the calculation process:

$$k \left( \frac{1}{1 - 0.2} \right)^2 = k \frac{25}{16}$$

(46)

$$\frac{\text{delay'}}{\text{delay}} = \frac{3}{2} = \frac{(V' - 0.2)^2}{25}$$

(47)

After calculation, because the equation is a quadratic equation with one variable, we get two results: $V' = 0.78V$, $V'' = 0.05V$. We need to make sure the MOS is working in the appropriate region, so $V' = 0.78V$.

Our final energy is:

$$E = C_L V^2$$

(48)

3.4 Logic Effort

We use static complementary CMOS to structure different gates. The inputs to our adder are driven by a unit-sized buffer (chain of two-unit sized inverters), and the output bit is loaded with $C_L = 32$ unit-sized inverters. $W_p/W_N$ of the unit-sized inverter is 2:1.

As we all know, it’s more convenient to structure the gates that have NOT logic with static complementary CMOS, such as NAND gate and NOR gate. In order to simplify the inner COMS construction of each gate and the calculation of Normalized delay, in our design, we use a NAND gate plus a NOT gate to replace a AND gate, and we use a NOR gate and a NOT gate to replace an OR gate.

In order to size the gate for the minimum delay, we need to identify the input vectors that will exercise critical path, the critical path is the path that goes through most gates and has the most branches, and here we’ve labeled the critical path whose color is red of the adders and comparators and marked the $g, h, b$ corresponding to each logic gate in Fig.5.

![Fig.5 The overall logic circuit of the spike detection algorithm](image-url)
To obtain the total normalized delay of the critical path, we need to calculate the logic effort ($g$) and Parasitic delay ($p$) of each gate. Through the method we mentioned in the Mechanism part, we obtain the $g$ and $p$ of each gate as shown in Tab.6:

| Gate            | $g$ | $p$ |
|-----------------|-----|-----|
| XOR gate        | 2   | 4   |
| NAND gate       | 4/3 | 2   |
| NOR gate        | 5/3 | 2   |
| Four-input NAND gate | 2   | 4   |
| Four-input NOR gate | 3   | 4   |

The branch effort also needs to be considered when calculating logic effort. To construct an XOR gate by static COMS construction, we have to get a negative logic of the input, that lead to a branch effort in the inner construction of the XOR gate, and it can be calculated to be 1.5. To sum up, there are ten branches in the critical path as shown in Tab.7.

| $b_1$ | $b_2$ | $b_3$ | $b_4$ | $b_5$ | $b_6$ | $b_7$ | $b_8$ | $b_9$ | $b_{10}$ |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 1.5   | 9/5   | 11/6  | 1.5   | 1.5   | 7/3   | 1.5   | 2     | 2     | 2       |

and

Finally, we have:
Path effort:

$$F = \prod g \cdot h \cdot b = 1868989.63$$

(49)

Fanout:

$$f = g \cdot h \cdot b = \sqrt[25]{F} = 1.78$$

(50)

Normalized delay:

$$D = 25 \cdot f + \Sigma P = 93.5$$

(51)

4. Conclusion

Based on the results and discussions presented above, the conclusions are obtained as below:

According to formula (36), when VDD is equal to 0.81, the y has a minimum value, which means that the optimal trade-off between the two is reached at this time. The combination of Carry Lookahead Adder and comparator has a good performance in both delay and energy consumption.

Acknowledgments

This work was financially supported by the Applied Research Project of Public Welfare Technology of Zhejiang Province (2017C33013).

References

[1] Wang Jianping. The development history of electronic technology application system[J]. Electronic Technology and Software Engineering, 2014(05): 125.
[2] Zhou Zixin. Briefly describe the development process of electronic technology[J]. Science Education Journal: Electronic Edition, 2016(16):141-141.
[3] Ye Wenhao. Deep love and rapid development of China's lamp triode semiconductor industry [J]. China Lighting, 2005, 000(006): 72-75.
[4] Chen Biao. The development of integrated circuit technology[J]. Microprocessor, 2011, 32(3):1
[5] Pan Guizhong. MOS integrated circuit structure and manufacturing technology [M]. Shanghai Science and Technology Press, 2010.
[6] C.-K. Tung, Y.-C. Hung, S.-H. Shieh, and G.-S. Huang, “A low-power high-speed hybrid CMOS full adder for embedded system,” in Proc. IEEE Conf. Design Diagnostics Electron. Circuits Syst., vol. 13, Apr. 2007, pp. 1–4.

[7] Zhou Dapeng, Long Anwen, Xu Guorong, et al. Structural parameter constraints for the optimal design of the advance position adder[J]. Journal of South-Central University for Nationalities (Natural Science Edition), 2006, 25(001): 58-61.

[8] Waste, N. H. E., & Harris, D. (2010). CMOS VLSI design: A circuits and systems perspective. (4th ed.). Boston: Addison Wesley Publishing Company.

[9] Panda S, Maji A, Mukhopadhyay A. Power and Delay Comparison in between Different types of Full Adder Circuits[J]. International Journal of Advanced Research in Electrical Electronics & Instrumentation Engineering, 2013, 1(3).

[10] Goal, S., Kumar, A., & Bayoumi, M. A. (2006). Design of robust, energy efficient full adders for deep-submicrometer design using hybrid-CMOS logic style. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 14(12), 1309–1321.

[11] Rabaey, J. M., Chandrakasan, A., & Nikolic, B. (2003). Digital integrated circuits: A design perspective. (2nd ed.). New Delhi: Pearson Education.