Proceedings of the 2023 International Conference on Machine Learning and Automation
DOI: 10.54254/2755-2721/39/20230604

Design and optimization of a 16-bit spike detector based on 90nm CMOS technology

Fuchang Wen

School of Computer and Communication Engineering, Northeastern University at Qinhuangdao, Qinhuangdao, 066000, China

202012725@stu.neu.edu.cn

Abstract. To deeply understanding of brain function, the rapid and accurate extracting relevant signals during brain operation has become increasingly important. As our understanding of brain function advances, the ability to extract relevant signals quickly and accurately becomes increasingly crucial. Therefore, a 16-bit spike detector has been designed to detect neural signals. Since a 4-bit absolute value detector is a critical module of the 16-bit spike detector, this paper presents a design for a 4-bit value comparator using a chain adder and a comparator. Two methods, changing the supply voltage and changing the gate size, were utilized to reduce power consumption, and it was found that changing the supply voltage method was more effective. Additionally, the circuit's layout was designed for simulation to enhance the authenticity and reliability of the results. Currently, brain-computer interface technology remains a widely discussed topic in the field of neuroscience, and the speed and accuracy of neural signal processing continue to be critical issues that need to be addressed.

Keywords: 4-bit Absolute Value Comparator, BCI Technology, Optimization, Layout.

1. Introduction

The human brain constantly generates signals in the form of action potentials that shuttle between neurons, transmitting information known as "spikes" [1]. These spikes can be detected by specific thresholding and used to extract neural signals. However, these neural signals often contain irrelevant noise such as background noise. To extract useful neural signals, spike sorting is needed to separate the activity of each neuron, which is achieved through spike detection [2].

Spike detection holds significant importance in neural signal processing, enabling the identification and tracking of neural data channels. With the continuous advancements in neural signal acquisition technology, the precise extraction of valuable neural signals has become imperative. In this study, we propose a 4-bit comparator-based approach for spike detection, consisting of two key components: absolute value computation and numerical comparison. The absolute value computation segment employs a series of chained full adders to calculate the absolute value using 2's complement representation, subsequently comparing it with a predetermined threshold. Our circuit design optimization takes two routes: firstly, voltage supply adjustments aimed at reducing power consumption, and secondly, aperture dimension modifications to minimize power consumption. Comparative analysis revealed the superiority of the first optimization method over the latter. Finally, this paper illustrates the circuit layout and employ simulation to validate the optimization outcomes. The simulation results

DOI: 10.54254/2755-2721/39/20230604

demonstrate close alignment between the calculated and anticipated results, well within the controllable range.

2. BCI signal thresholding analysis

This chapter will present an explanation of the functioning principles of the field of Brain-Computer Interfaces (BCIs) and the techniques employed for the analysis of BCI signals, with a particular emphasis on peak detection. The discussion will lead to the design of a 4-bit comparator based on the principles of peak detection.

2.1. BCI operating principle

Research in neuroscience has found that the electrical activity of the nervous system changes in response to external stimuli before the conscious decision to move is made, or before the execution of the movement itself. This alteration can be discerned through distinctive methods and employed as a distinctive indicator of an imminent action [3]. By classifying and identifying these characteristic signals, it is possible to distinguish the intended action that caused the change in brain activity, and then program a computer to translate this intention into a command signal that can drive external devices. This metamorphosis empowers the regulation of the external milieu solely through cerebral activity, without the direct intervention of muscles or peripheral nerves. This is the basic working principle of brain-computer interfaces (BCIs).

The first international BCI conference defined BCI as a communication system between the brain and computer that does not rely on the normal output pathway of the peripheral nerves and muscles. This characteristic of BCI, which facilitates direct communication between the brain and computer without the involvement of muscles and peripheral nerves, is of great significance in the treatment of patients with complete immobility due to conditions such as stroke and cerebral palsy, as well as in the recovery of language and behavioural functions [3].

BCI systems generally consist of input, output, and signal processing and conversion components. The input component generates and detects characteristic EEG signals and describes them using parameters. The signal processing component analyses the source signals, converts continuous analogue signals into digital signals represented by certain characteristic parameters, and recognizes and classifies these characteristic signals to determine the corresponding conscious activity. The signal conversion component drives or operates the characteristic signals obtained from the processing component to produce output results and complete communication with the external environment. Figure 1 illustrates the fundamental architecture of the BCI system.

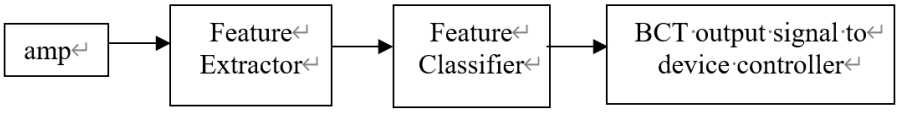

Figure 1. BCI structural representation (Photo/Picture credit: Original).

2.2. Spike detection

As a crucial component of BCI systems that bridges the gap between input and output, signal analysis and conversion can enhance classification accuracy and optimize control performance through improved algorithms [4]. Signal manipulation in BCI systems involves signal preprocessing, feature extraction, and recognition/classification. Of these stages, feature extraction and recognition/classification are the most pivotal phases in BCI signal processing. Feature extraction involves using feature signals (usually EEG brainwave signals) as source signals to identify various parameters and generate feature vectors that characterize signal features. Attributes like amplitude and frequency can be derived from either the time domain or the frequency domain, and the waveform's feature parameters extracted through this methodology find utility in tasks like classification, recognition, tracking, and transient analysis. The time-domain feature extraction technique amalgamates specialized filtering methods with sampling

DOI: 10.54254/2755-2721/39/20230604

techniques, enabling the reduction of time-domain noise within the EEG signal, and amplifying the signal-to-noise ratio. The primary focus of this paper revolves around peak detection and the development of corresponding circuits for the realization of BCI signal detection.

Spike detection can be used in brain-machine interfaces to extract feature vectors from previously unclassified action potentials, thereby selecting meaningful channels. These feature vectors also include threshold crossing events of channels without neural data, and peak detection will discard all channels without neural data. Therefore, spike detection can automatically identify which signals are noise and which are useful. Spike detection is typically divided into two stages: pre-emphasis and thresholding, where neural signals are first converted into digital signals and then compared with a threshold [2]. When a neural signal surpasses a set threshold, a peak is detected. This paper will design an absolute value detector and a comparator based on this characteristic to achieve this function

3. A 16-bit Spike Detector

Based on the previously discussed peak detection method, this chapter will present the design principles and circuit diagram for a 16-bit spike detector. The structure of 16-bit spike detector is shown in the figure 2.

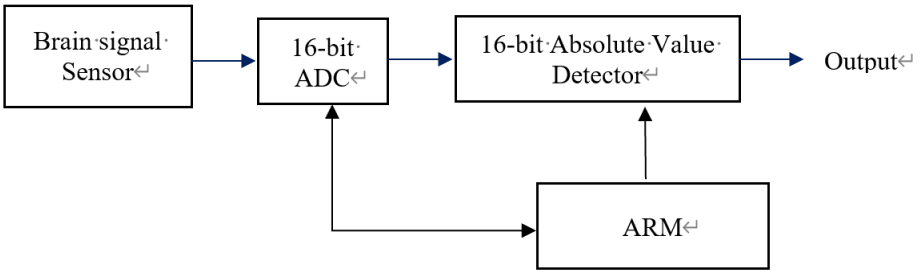

Figure 2. 16-bit absolute spike detector structure (Photo/Picture credit: Original).

There are four modules in figure 2. They are: 1. Brain signal sensor: outputs brain's spike which is analogue. 2. A 16-bit ADC is used to convert spike signal into 16-bit digital signal which is send to 16-bit absolute spike detector. 3. ARM used to: Control the operation status of 16-bit ADC. ADC status could be Power on, Power down, sleep, reset, start conversion, stop conversion, reset, and so on. Set any 15-bit digital value as the threshold for 16-bit absolute value detector. 4. 16-bit absolute detector output comparison result after it compares the value output by ADC with the threshold value set by ARM. As brain signal sensor, 16-bit ADC and ARM are nominal devices, this paper focus on the design of a 16-bit absolute value detector.

3.1. Design of a 16-bit Absolute Value Detector

3.1.1. Principles

This paper proposes the design of a 16-bit absolute value detector for preprocessing the input signal in BCI, which compares the amplitude of the input voltage with any 15-bit threshold voltage set by ARM. The principle of 16-bit absolute value detector is shown as figure 3.

DOI: 10.54254/2755-2721/39/20230604

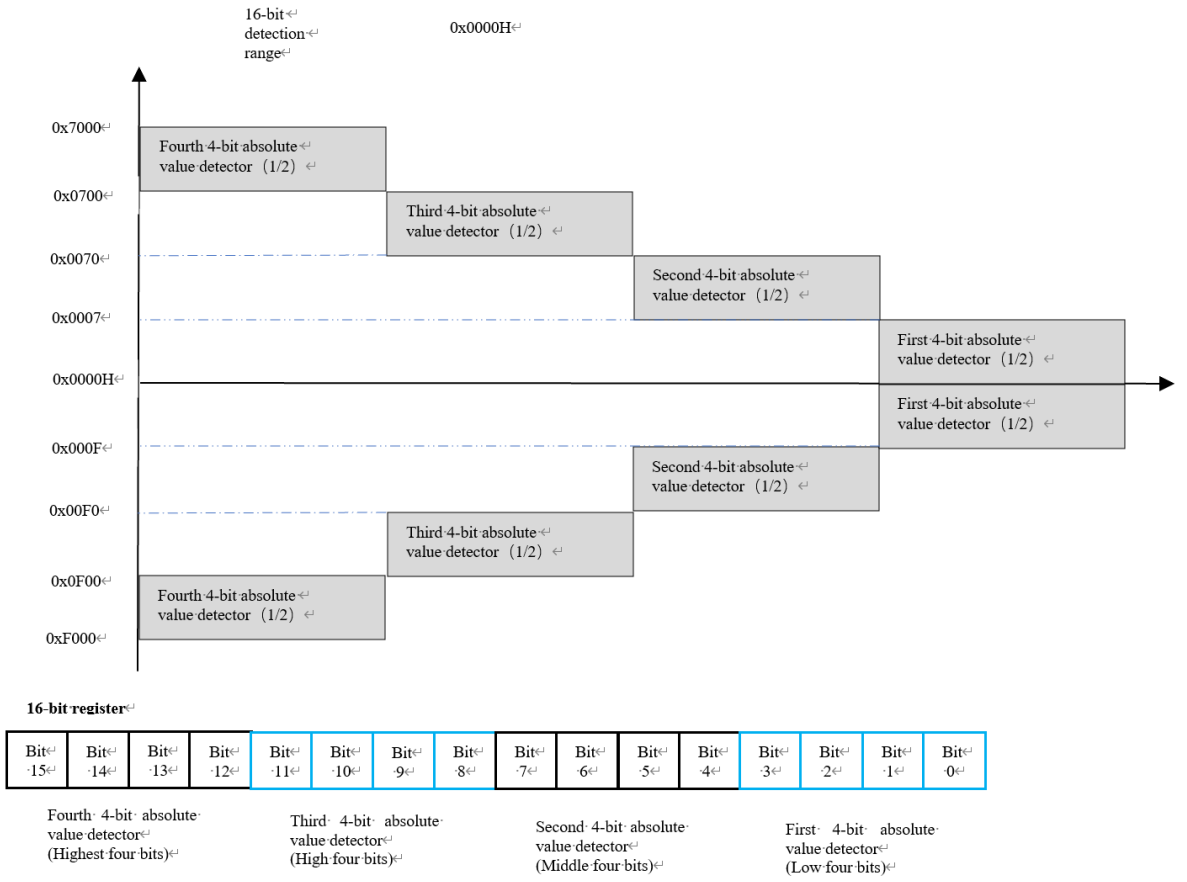

Figure 3. 16-bit absolute value detector principle (Photo/Picture credit: Original).

Figure 3 illustrates the construction of a 16-bit absolute value detector, comprising four individual 4-bit absolute value detectors and a 16-bit register. This 16-bit register serves as the repository for the detection threshold value, which is written by the ARM. Specifically, the lower 4 bits of the 16-bit register are allocated to the first 4-bit absolute value detector, the middle 4 bits to the second 4-bit absolute value detector, and so forth.

3.1.2. Structure of 16-bit Absolute value detector

Figure 4 depicts the architecture of a 16-bit absolute value detector.

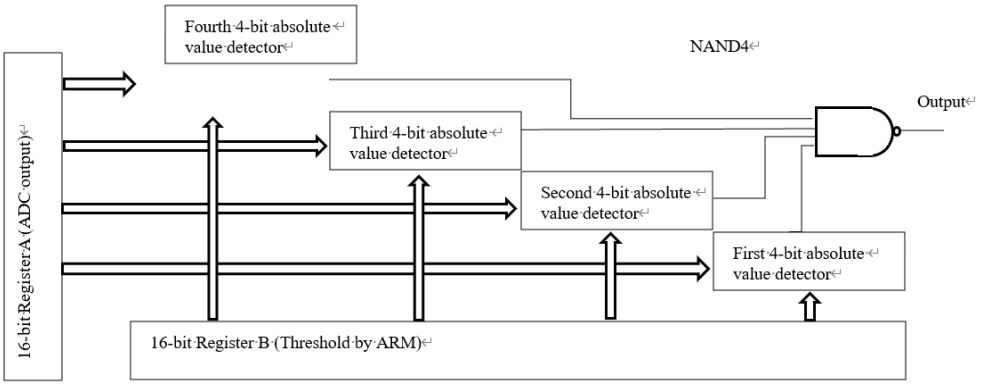

Figure 4. 16-bit absolute value detector structure (Photo/Picture credit: Original).

DOI: 10.54254/2755-2721/39/20230604

Figure 4 depicts the architecture of the 16-bit absolute value detector, which consists of two 16-bit registers, denoted as A and B, four 4-bit absolute value detectors, and a four-input NAND gate (NAN4). Register A receives the low 4 bits from the ADC, while Register B stores the threshold values. The First 4-bit absolute value detector compares the low 4 bits from Register A with the corresponding low 4 bits in Register B, and this process is replicated for the middle 4 bits in both registers, and so forth. The outputs of the four 4-bit absolute value detectors feed into the input of the NAN4 gate, which then produces the comparison result for the 16-bit absolute values held in Registers A and B.

In light of the above discussion, it is evident that the 4-bit absolute value detector serves as a crucial module within the 16-bit spike detector. Our next step involves designing the 4-bit absolute value detector using 90nm CMOS technology.

3.1.3. 4-bit absolute value detector circuit

The 4-bit absolute value detector is divided into two parts shown as figure 5: the absolute value detector and the comparator. Since the input signal is given in 4-bit two's complement form while the threshold voltage is a 3-bit value, the absolute value detector is necessary to convert negative numbers to positive ones before comparing the last three digits with the threshold voltage [5].

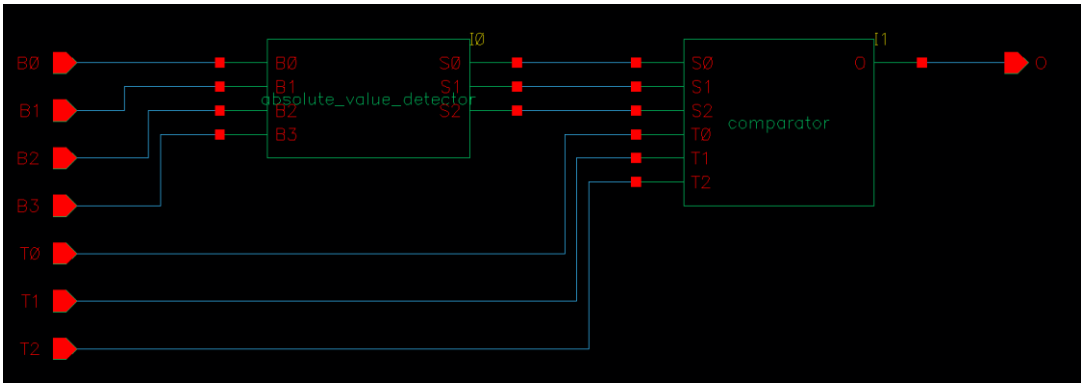

Figure 5. 16-bit absolute value detector (Photo/Picture credit: Original).

The function of 4-bit absolute value detector is shown below. Vin represents the 4-bit input voltage amplitude, which is converted to a 3-bit unsigned number by the absolute value detector. Threshold is the given 3-bit threshold voltage, which is compared with the converted 3-digit unsigned number in the comparator. If the threshold is less than |Vin|, the output is high; otherwise, the output is low. The requirement of the design is shown in figure 6.

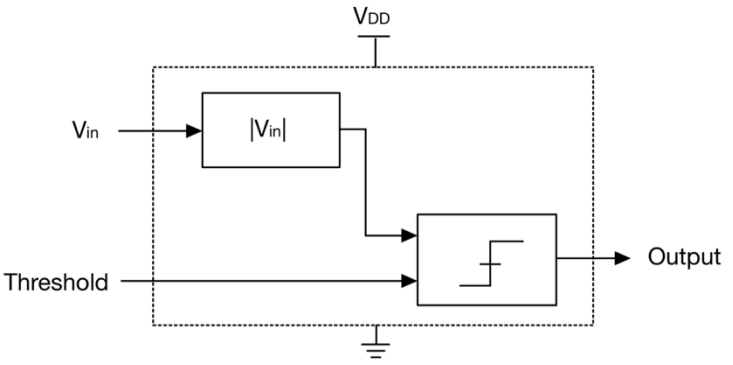

Figure 6. Requirement of the design (Photo/Picture credit: Original).

DOI: 10.54254/2755-2721/39/20230604

The input signal ranges from -7 to +7 and is represented by a 4-bit value with the highest bit, A3, indicating the sign. If A3 is 0, the input signal is positive; if A3 is 1, the input signal is negative. The input signal needs to be processed as follows: when the input signal is positive, the 3-bit value can be directly input to the comparator without any processing. When the input signal is negative, it must first be converted to a positive number using the two's complement method before comparison. The principle of two's complement is that when the highest bit is 1, each bit is inverted, changing from 0 to 1 or from 1 to 0, and then the lowest bit is added by 1. For example, if the input signal is 1011, which is -5 in decimal, after inversion, it becomes 0100, and then adding 1 to it results in 0101, which is +5 in decimal. The truth table for taking the absolute value is shown in the table 1 below [6].

Table 1. Truth value table of absolute value logic.

Input($A_3A_2A_1A_0$)	Output($S_2S_1S_0$)	Input($A_3A_2A_1A_0$)	Output($S_2S_1S_0$)
0000	000	1000	000
0001	001	1001	111
0010	010	1010	110
0011	011	1011	101
0100	100	1100	100
0101	101	1101	011
0110	110	1110	010
0111	111	1111	001

3.2. Circuit Composition

Since the two's complement method involves addition for each bit and the carry must also be considered, a 3-bit ripple carry adder is selected to complete the absolute value detector [7]. The figure 7 below shows the binary addition logic structure of the two's complement operation. As can be seen from the figure 7, three 1-bit full adders can be cascaded into a 3-bit ripple carry adder. A3 is used as the mode control input line, and when A3=0, addition is performed; when A3=1, subtraction is performed, which is equivalent to adding 1 to the lowest bit of the adder to complete the two's complement operation.

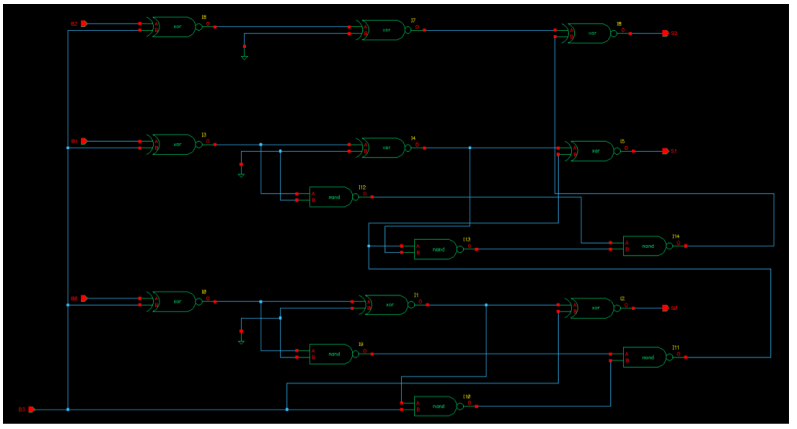

Figure 7. A 3-bit full adder (Photo/Picture credit: Original).

This paper applying the two's complement operation to the input signal amplitude, which is shown in figure 8 and figure 9. It obtains a 3-digit value, which is then subjected to direct comparison with a provided threshold value using comparator [8].

DOI: 10.54254/2755-2721/39/20230604

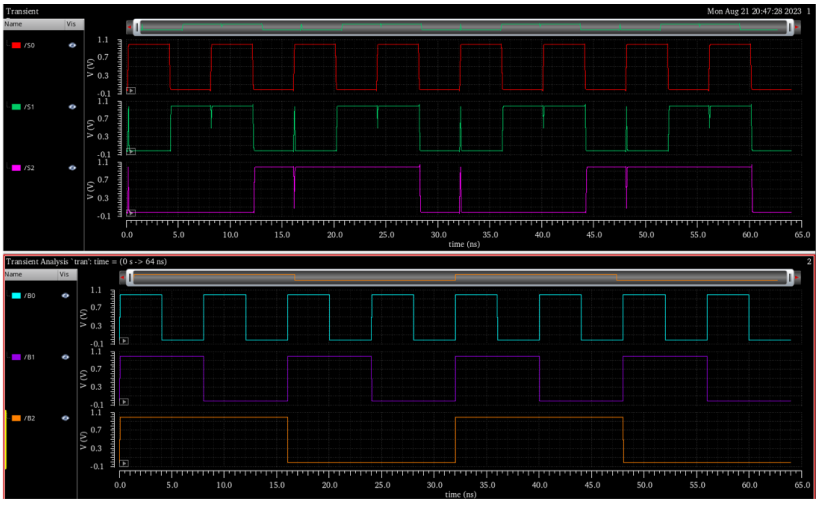

Figure 8. Absolute value operation when the input is negative (Photo/Picture credit: Original).

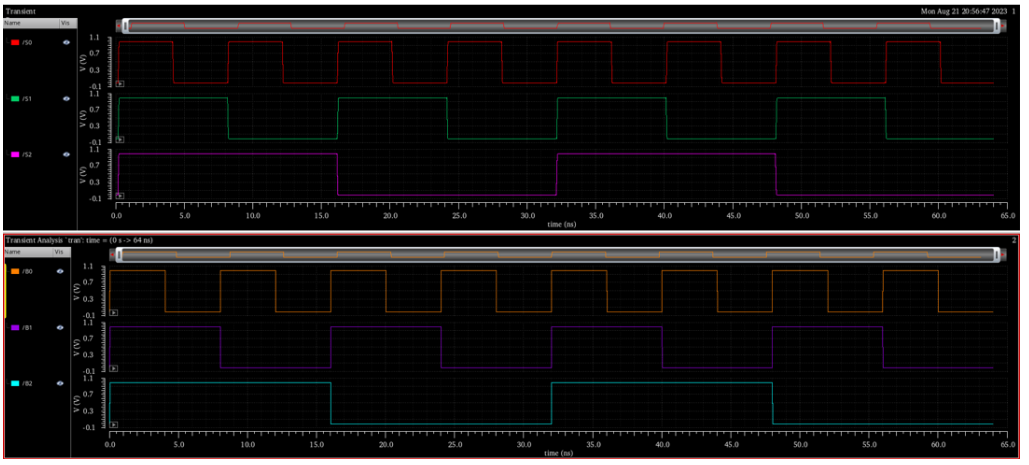

Figure 9. Absolute value operation when the input is positive (Photo/Picture credit: Original).

The diagram below in figure 10 illustrates the logical circuitry of a 3-bit comparator. In this setup, the highest bit, S2, from the input is juxtaposed with the highest bit, T2, from the threshold voltage. When S2 exceeds T2, the output assumes a high state; if S2 is less than T2, the output registers a low state. In cases where S2 equals T2, the subsequent bit is brought into comparison, continuing until the relative magnitudes of the signal amplitude and the threshold value are resolved. When the signal amplitude aligns with the threshold value, the output is maintained at a low state [9].

DOI: 10.54254/2755-2721/39/20230604

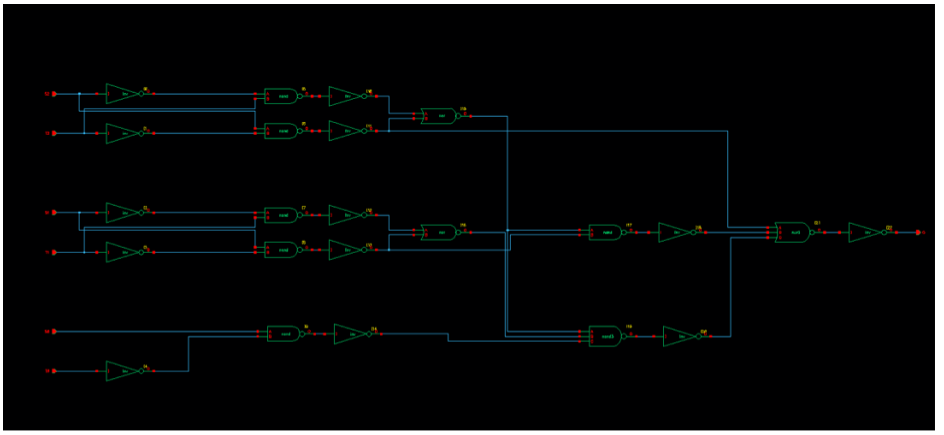

Figure 10. A 3-bit comparator (Photo/Picture credit: Original).

After connecting the absolute value circuit and the comparator circuit, the final 4-bit absolute value detector is obtained as shown in the figure 11 below.

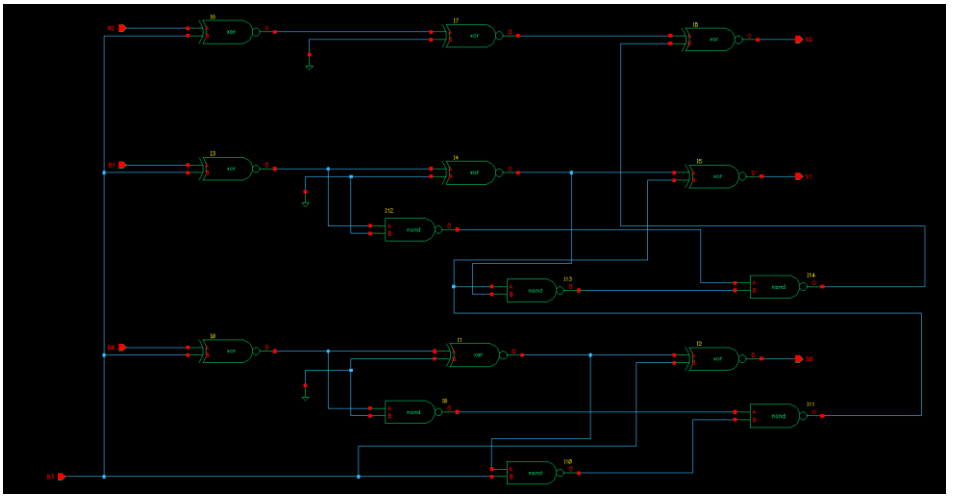

Figure 11. 4-bit absolute value detector (Photo/Picture credit: Original).

Based on the given threshold voltage of 4 (T2T1T0=100), the result of comparing the input signal amplitude with the threshold voltage is shown in the figure 12 below.

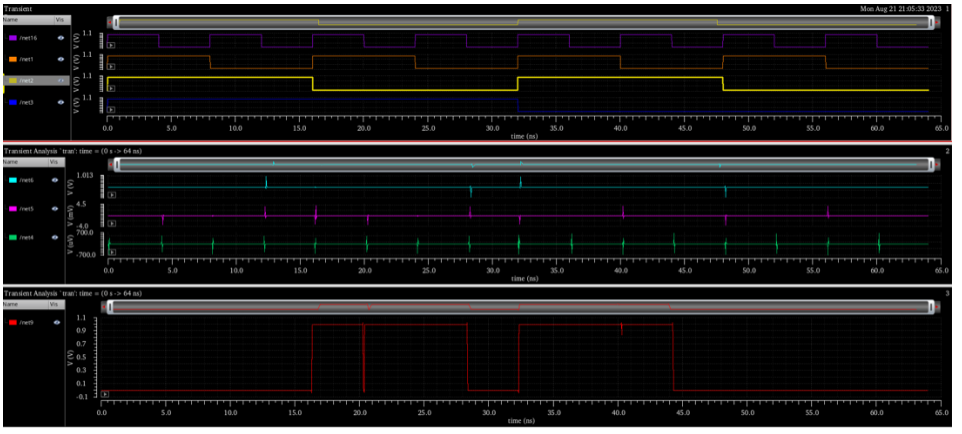

Figure 12. The input signal is compared with a voltage threshold of 4 (Photo/Picture credit: Original).

DOI: 10.54254/2755-2721/39/20230604

4. Key Metrics

Within this section, this paper will enhance the efficiency of the previously devised circuit diagram through two distinct approaches: firstly, by lowering the VDD (power supply voltage), and secondly, by adjusting the gate dimensions to introduce a 1.5-fold increase in delay. Furthermore, a corresponding layout has been devised based on the circuit diagram to validate the outcomes.

4.1. Critical Path Analysis

After designing the circuit, this article will reduce the power consumption of the circuit by increasing the minimum delay of the minimum energy delay ratio topology by 50%. "Delay" refers to the propagation delay caused by the critical path of the circuit, and "energy" refers to the total energy extracted from VDD, including the output [10]. Therefore, to optimize the circuit, it is necessary to first find the critical path of the circuit. The figure 13 below shows the critical path of the designed circuit in this article.

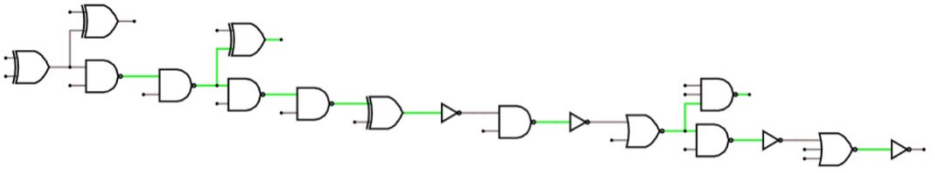

Figure 13. Critical path (Photo/Picture credit: Original).

The critical path in the above figure consists of 14 stages, including 2 XOR gates, 6 NAND gates, 2 NOR gates, 4 inverters, and 3 branches. To facilitate the calculation of delay, this article makes the following assumptions: (1) The unit size inverter is Wp=650nm, Wn=470nm. (2) the input capacitance of each input should not exceed the capacitance of two unit-sized inverters connected in series. (3) The output is an inverter with a size of CL=32 units. (4) VDD is set to 1V.

Based on the above assumptions, the logical effect and parasitic delay of each basic logic unit can be calculated, as shown in the table 2 below. The parameter g represents the logical effect, and the parameter p represents the parasitic delay.

Table 2. Logic and parasitic effort of common gates.

gate type	number of input					
	1		2		3	
	g	p	g	p	g	p
inverter	1	1				
Nand			1.398	2	1.796	3
Nor			1.6	2	2.204	3
Xor			4	4		

4.2. Minimum delay and power consumption

The total logical effect G and parasitic delay P can be calculated from Table 2, and minimum delay can be calculated by the following formular:

$$G = \prod g_i \tag{1}$$

$$H = \frac{C_{out}}{C_{in}} \tag{2}$$

$$B = \sum \frac{C_{on-path} + C_{off-path}}{C_{off-path}} \tag{3}$$

$$F = GHB \tag{4}$$

DOI: 10.54254/2755-2721/39/20230604

$$f = \sqrt[N]{F} \tag{5}$$

$$P = \sum p \tag{6}$$

$$D = Nf + P \tag{7}$$

$$f = g \times h \tag{8}$$

$$C_{in} = g \times \frac{C_{out}}{f} \tag{9}$$

$$E = C \times {V_{DD}}^2 \tag{10}$$

Where H is the Path Electrical Effort, B is the Path Branch Effort, F is the fanout of the entire circuit, D is the delay of the entire circuit, and N represents the number of stages in the critical path, which is 14 in this circuit. After calculation, the fanout f of each stage is 2.36, the total delay D is 62.04 ps, and the power consumption E is 39.61.

Using the formula for Cin, the size of each stage in the critical circuit can be calculated starting from the last stage [7]. The results are shown in the table 3 below.

Table 3. Calculation result of size of each stage on critical path.

stage	1	2	3	4	5	6	7
C_{in}	0.08	0.05	0.08	0.14	0.23	0.39	0.23
stage	8	9	10	11	12	13	14
C_{in}	0.54	0.91	2.15	3.18	5.37	12.66	13.6

4.2.1. Reducing Power Consumption by Adjusting VDD

After calculating the minimum delay of the critical path under VDD=1, this section aims to satisfy the 1.5x delay requirement by adjusting the VDD value within the range of 0-1V, in order to reduce power consumption. Assuming a VT value of 0.2V and a Dopt value of 1.5 times the minimum delay D, the following formula is used for calculation:

$$D = \frac{K \times V_{DD}}{(V_{DD} - V_T)} \tag{11}$$

$$D_{opt} = \frac{K \times V_{DD-opt}}{(V_{DD-opt} - V_T)} \tag{12}$$

After calculation, it can get:

$$V_{DD\text{-}opt} = 0.78V \tag{13}$$

Therefore, to meet the 1.5x delay requirement, we can lower the power supply voltage to 0.78V to satisfy the required delay while effectively reducing power consumption.

The power consumption in this case is:

$$E = E_0 \cdot {V_{DD}}^2 = 24.1 \tag{14}$$

4.2.2. Lowering Power Consumption by Adjusting Gate Size

Adjusting the gate size can lead to changes in the total capacitance and power consumption of the critical path. In this section, we aim to determine the minimum power consumption by changing the size of each gate on the critical path while keeping VDD=1 and Dopt=1.5D constant. The modified gate sizes are shown in the following table 4.

DOI: 10.54254/2755-2721/39/20230604

Table 4. When the size equal L.

Sizing	1	1	1	1	1	1	1
g	4	1.398	1.398	1.398	1.398	4	1
h	1	1	1	1	1	1	1
Sizing	1	1	1	1	1	1	L
g	1.398	1	1.6	1.398	1	2.204	1
h	1	1	1	1	1	L	32/L

$$D_{opt} = \sum g_i h_i + P = 20.998 + 2.204L + \frac{32}{L} + 29 = 1.5D \Rightarrow L = 18.77 \quad (15)$$

$$E = C \cdot {V_{DD}}^2 = (14 + 18.77) \times 1 = 32.77 \quad (16)$$

4.2.3. Reducing Power Consumption by Adjusting Gate Size and VDD

In this section, we aim to further reduce power consumption by combining the two approaches mentioned above. The calculation process involves first determining the delay for different gate sizes, then calculating the optimal VDD required to satisfy this delay, and finally calculating the power consumption. When L equals to 20:

$$D_{opt} = \sum g_i h_i + P = 20.998 + 2.204L + \frac{32}{L} + 29 = 95.668 \quad (17)$$

$$\frac{\frac{1}{(1-0.2)^2}}{\frac{V_{DD}}{(V_{DD}-0.2)^2}} = \frac{1.5D}{D_{opt}} \Rightarrow V = 0.98 \quad (18)$$

$$E = C \cdot {V_{DD}}^2 = (14 + 20) \times 0.98^2 = 32.65 \quad (19)$$

Change the size of L, we can get different power consumption, the result are as follows in table 5.

Table 5. Scaling V_{DD} and gate sizing for energy reduction.

L	20	25	30	31	32	33	34
V	0.982	0.916	0.862	0.852	0.842	0.834	0.825
Delay	32.78	32.72	32.69	32.67	32.61	32.69	32.61
L	35	36	37	38	39	40	41
V	0.816	0.808	0.8	0.792	0.784	0.777	0.77
Delay	32.62	32.64	32.64	32.61	32.57	32.6	32.61

It can be observed that changing the gate size has a minimal effect on power consumption. By only adjusting the VDD value without changing the gate size, we were able to achieve the minimum power consumption of E=24.1.

4.3. Circuit layout

The circuit layout is a graphical representation of the physical information related to the type, size, relative position, and connectivity of devices in an integrated circuit. The implementation of circuit functionality on the layout can greatly reflect the performance, cost, and power consumption of the circuit [11]. After designing the circuit diagram, the corresponding layout can be drawn based on the design of the CMOS transistor in each gate circuit [12]. Figure 14 shows the schematic diagram of an inverter.

DOI: 10.54254/2755-2721/39/20230604

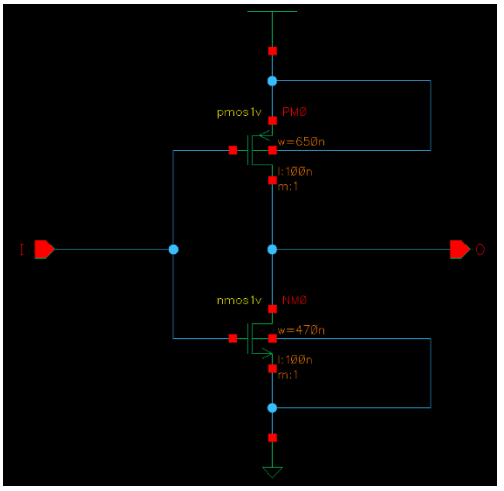

Figure 14. Schematic diagram of the inverter (Photo/Picture credit: Original).

The corresponding layout can be drawn based on the connecting method of the CMOS transistor in the figure, as shown in Figure 15.

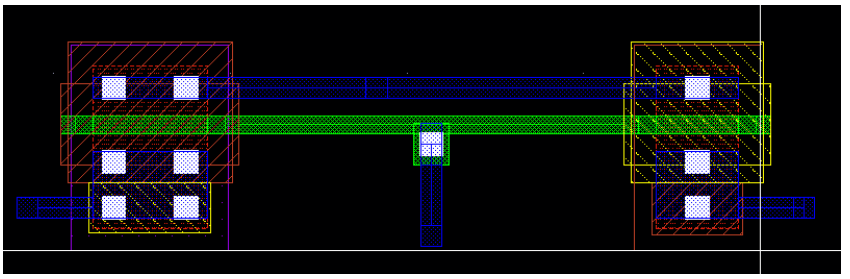

Figure 15. Layout of inverter (Photo/Picture credit: Original).

The left side shows the PMOS transistor, with the yellow part as the drain, connected to VDD through the blue metal wire, and the red part as the source, connected to the source of the NMOS transistor through a metal wire. The green part is the gate, connected to the input through the middle metal wire. The right side shows the NMOS transistor, with the red part as the drain, connected to ground through the blue metal wire, and the yellow part as the source, connected to the source of the PMOS transistor through the blue metal wire. The white area is a contact hole used to connect the metal wire. Other logic gates are connected in the same way, and the entire map can be drawn in figure 16.

DOI: 10.54254/2755-2721/39/20230604

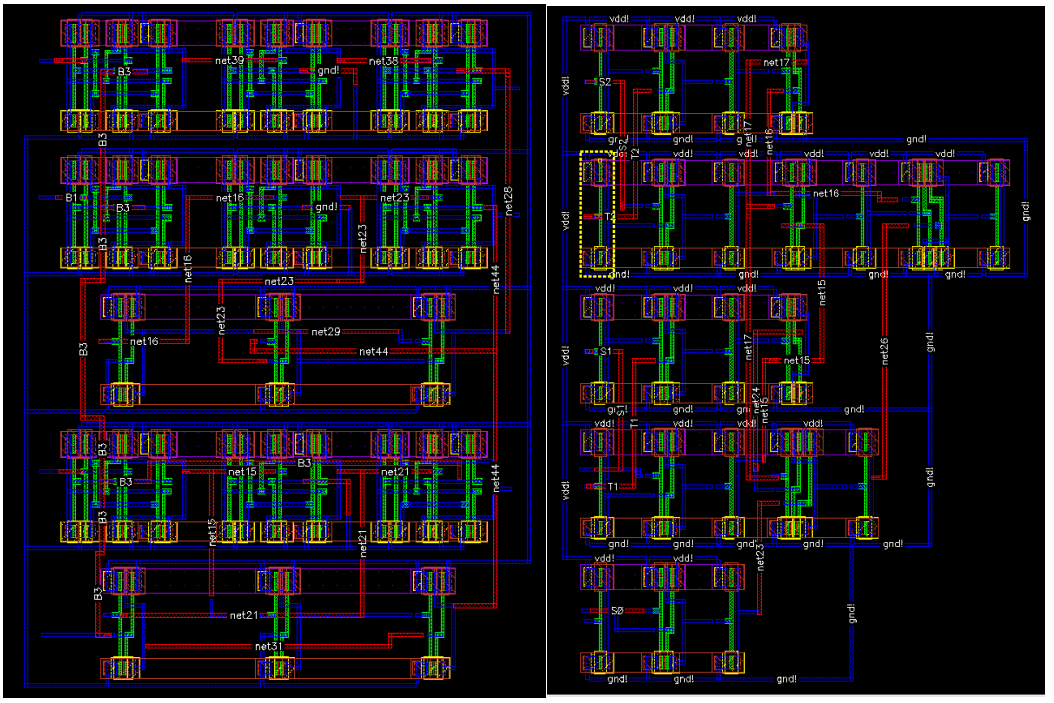

Figure 16. Layout of 3-bit full adder and comparator (Photo/Picture credit: Original).

The layout of 4-bit absolute value detector is shown as figure 17.

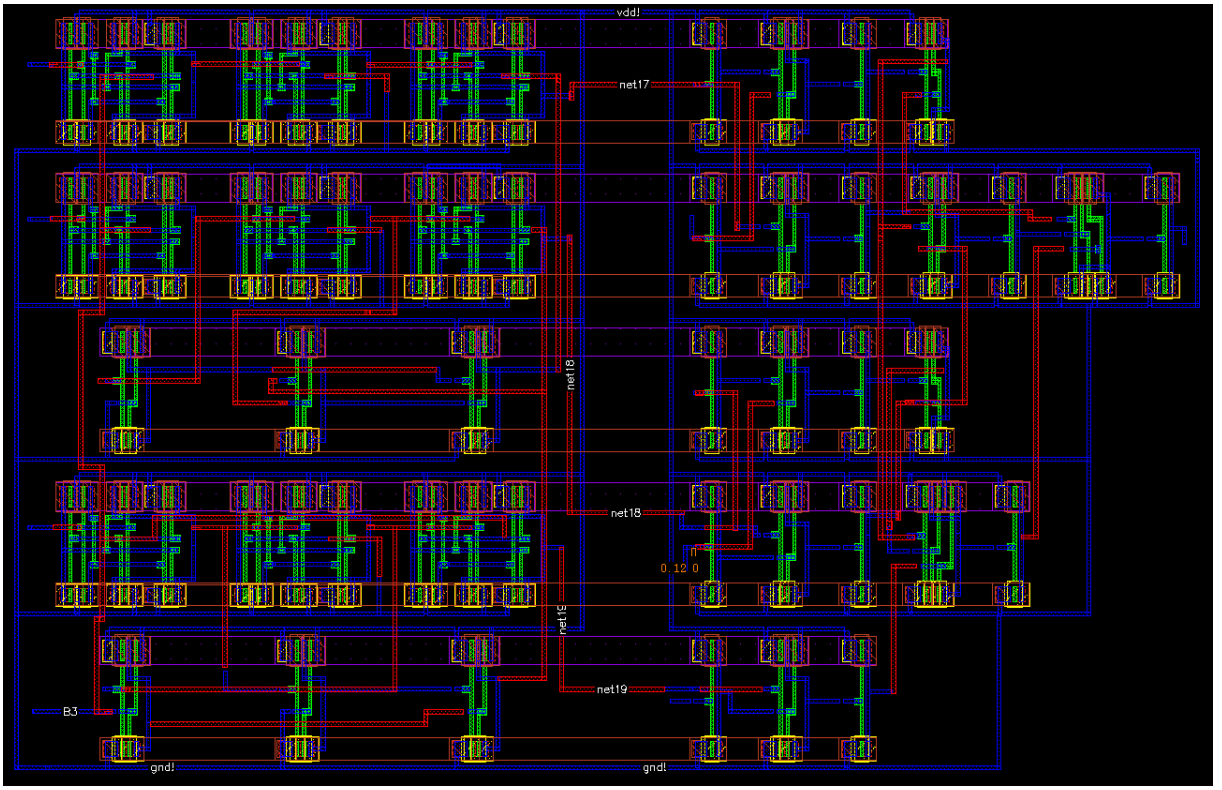

Figure 17. Layout of 4-bit absolute value detector (Photo/Picture credit: Original).

DOI: 10.54254/2755-2721/39/20230604

After layout simulation, the calculated delay is 65.32 and power consumption is 26.37. Due to the layout wiring, device orientation, parasitic capacitance, and other influencing factors in the layout, the calculated values may differ from the values obtained from the schematic diagram calculation, which makes them more accurate.

5. Conclusion

This article presents the design of a 16-bit spike detector, relying on a pivotal 16-bit absolute value detector that originates from the foundation of a 4-bit absolute value detector. The operability of the 16-bit spike detector is contingent upon a 16-bit ADC, which can be seamlessly extended to accommodate 32-bit or higher resolutions when employing 32-bit or greater ADCs. Given the paramount importance of the 4-bit absolute value detector within the framework of a 16-bit spike detector, we have developed this module using 90nm CMOS technology. The 4-bit absolute value detector incorporates a chained adder and comparator, strategically mitigating power consumption through adjustments in power supply voltage and gate size. The results show that changing the power supply voltage is a better method, while changing the gate size mainly affects the VT of the CMOS transistor and has little effect on delay and power consumption. To improve the accuracy and reliability of the content, the circuit layout is also drawn to make the designed circuit more practical. The key path of the circuit has a large number of stages and branches, and the performance can be further optimized by changing the circuit structure. Currently, BCI technology is still a widely discussed topic, and it has significant implications in various fields involving the human brain. At the same time, the speed and accuracy of neural signal processing are still issues that need to be addressed.

References

[1] Lewicki M S. A review of methods for spike sorting: the detection and classification of neural action potentials. Network, 1998.
[2] Obeid I, Wolf P D. Evaluation of spike-detection algorithms for a brain-machine interface application. Biomedical Engineering IEEE Transactions on, 2004, 51(6):905-911.
[3] Noninvasive Brain–Machine Interfaces for Robotic Devices. Annual Review of Control Robotics and Autonomous Systems, 2021, 4(1).
[4] D Valencia, Alimohammad A. Neural Spike Sorting Using Binarized Neural Networks. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 2020, (99):1-1.
[5] Yuan, M. An absolute-value detector with threshold comparing for spike detection in the brain-machine interface. Journal of Physics: Conference Series, 2021, 2113(1): 012038.
[6] Charles S. De Morgan's law and theorems. 2013.
[7] S. Prema. Comparative Analysis of Different Full Adder Circuits. Programmable Device Circuits and Systems,2013,5(2).
[8] Efstathiou Constantinos,Kitsos Paris. Efficient majority logic magnitude comparator design. Microprocessors and Microsystems,2021,82.
[9] Efstathiou, C., & Kitsos, P . Efficient majority logic magnitude comparator design. Microprocessors and Microsystems, 2021, 82: 103-112.
[10] Sharroush, S. M., & Abdalla, Y. S. Parameter extraction and modelling of the MOS transistor by an equivalent resistance. Mathematical and Computer Modeling of Dynamical Systems, 2021, 27(1): 50–86.
[11] Kim, K., & Park, S. Delay-time modeling and critical-path verification for CMOS digital designs. Computer-Aided Design, 1991, 23(9): 604–614.
[12] Sharma, V., & Rajawat, A. Review of Approaches for Radiation Hardened Combinational Logic in CMOS Silicon Technology. IETE Technical Review, 2018, 35(6): 562–573.