A 42.4 FO4 (1V), 19.27 Eu (1V) 4-bit Absolute-Value Detector

Rui Gao

School of Electronic Science and Engineering (National Exemplary School of Microelectronics),
University of Electronic Science and Technology of China, Chengdu 611731, China

rgao@stu.uestc.edu.cn

Abstract. Absolute-value detector (AVD) is a fundamental arithmetic logic circuit widely applied in spike-sorting area. In this paper, a novel circuit design of a concise 4-bit AVD whose topology consists of a 2's complement circuit and a 3-bit ripple carry adder (for comparison purpose), etc., with critical path designed to be as short as possible. Since the minimum energy consumption and the minimum delay cannot be guaranteed at the same time, critical path gate sizing is conducted. Using the method of logic effort and appropriately relaxing the delay requirements to 1.5 times the minimum value, a suitable voltage ranging from 0V to 1V is found to achieve the goal of minimum power consumption.

Keywords: Absolute-value Detector, Logic Effort, Gate Sizin.

1. Introduction

Recently, spike-sorting algorithms have attracted a wide range of interest in the research community with the current progress in applications such as neural signal detecting systems, where comparison of absolute values of the signal is demanded [1, 2]. Among many methods of spike-sorting, one of the universally applied spike-detection algorithms, absolute-value detection, is chosen for further research and optimize its design.

Assume that the neural signal input is a digital signal. If analog, analog-to-digital converter should be implemented first. Based on ease of construction and ease of analysis, the circuit will be divided into the following two steps, after the signal is converted digitally, its magnitude is extracted first and compared with a threshold input, where a spike is detected when the magnitude is greater than the preset threshold. In other words, the goal of this design is to output a "1" (high logic value) when the magnitude of the input signal surpasses the given threshold value, otherwise output a "0" (low logic value).

In order to make the equipment installed with this circuit have a long battery life, its power consumption should be reduced as much as possible; at the same time, in order to make the equipment installed with this circuit sensitive to pulses, its delay should also be reduced as much as possible. For achieving these two goals, the logical effort theory is used to analyze the circuit in this paper. [3]

The paper is organized as follows. Section II describes the algorithm and the logic design of AVD, with circuit schematic attached to. Section III displays the key parameters by means of logic effort theory. Section IV demonstrates the characteristics of the circuit.

2. Absolute-value Detector (AVD)

2.1. The algorithm of AVD

In the author's opinion, a 4-bit absolute-value detector (AVD) already has all the functions of a general AVD, and it is easy to expand it to more complicated AVDs, hence the paper will design and analyze a 4-bit AVD, taking a 4-bit binary digital signal input (integers which range from -7 to 7) as an example.

Figure 1 displays the circuit structure of an absolute-value detector.

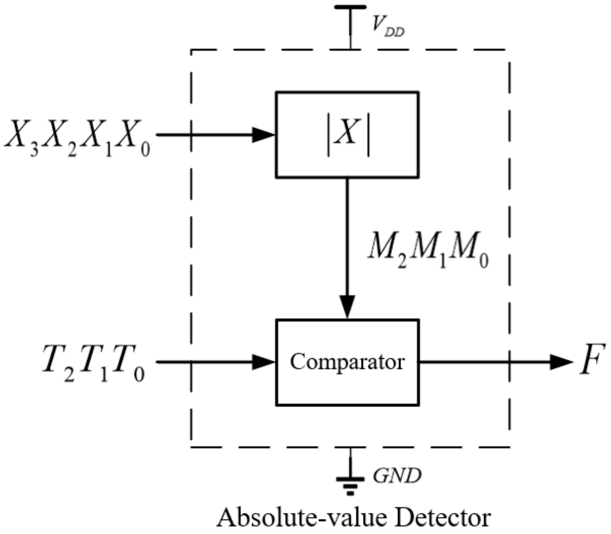

Figure 1. Structure of an absolute-value detector

In order to realize absolute value detector (AVD) mentioned in this paper, signal input should first be converted into an unsigned binary number, namely the absolute value of the signal, which should then be compared with the threshold and output 1-bit high level if it is greater than the threshold.

As two's complement format is widely applied in computer science as the most common method of representing signed (positive, negative, and zero) integers on computers, [5] negative numbers are discussed in such format for its practicality and versatility.

The algorithm of the circuit is mainly divided into two parts:

(1) Realization of absolute value. Assume that input signal is X, the sign of the input signal should be determined first and can be determined according to the sign bit, X_3, which is the most significant bit (MSB) of X. For positive input signals, the intermediate variable output M of magnitude extraction is the same as the input. For negative input signal, perform two's complement calculation to get the intermediate variable, the corresponding positive value M.

(2) Realization of comparison the magnitude with threshold input. Compare the result M with a given threshold T and the output result F. A ripple carry adder is used for comparison, and F outputs 1 when the size of M is greater than T, and outputs 0 in other cases.

2.2. The Logic Design of AVD

Since the input is a 4-bit two's complement number, i.e., $X_3X_2X_1X_0$, its unsigned counterpart should also be a 4-bit binary number which ranges from 0 to 7 and whose most significant bit (MSB) is 0, i.e., $0M_2M_1M_0$. Since the MSB of the absolute value is always zero, 3-bit threshold input, i.e., $T_2T_1T_0$, can already meet the needs, and there is no need to add more bits.

As mentioned above, the algorithm of AVD in this paper can be divided into two steps, realization of absolute value (Step I) and comparison with the threshold (Step II).

In Step I, input signals can be discussed in positive and negative. If the input is positive (most significant bit is 0), then its absolute value is exactly identical to the least 3 bits of the input.

$$M_{0+} = X_0 \tag{1}$$

$$M_{1+} = X_1 \tag{2}$$

$$M_{2+} = X_2 \tag{3}$$

If the sign is negative (most significant bit is 1), then its absolute value is the least 3 bits of the two's complement of the input. Logical expressions are shown in detail as follows.

$$M_{0-} = X_0 \tag{4}$$

$$M_{1-} = X_1 \oplus X_0 \tag{5}$$

$$M_{2-} = X_2 \oplus (X_1 + X_0) \tag{6}$$

In order to select the correct value as the unsigned absolute value output under circumstances of the variation of the input MSB, several 2-to-1 multiplexers (MUX) controlled by the MSB are required. To realize the multiplexer, the simplest and easiest way is to implement combinational logic circuit, which is adopted in the paper. After mixing the circuits represented by equation (1) to (6), the realization of absolute value can be described as the following Boolean expressions.

$$M_0 = \overline{X_3} M_{0+} + X_3 M_{0-} = X_0 \tag{7}$$

$$M_1 = \overline{X_3} M_{1+} + X_3 M_{1-} = \overline{X_3} X_1 + X_3 (X_1 \oplus X_0) \tag{8}$$

$$M_2 = \overline{X_3} M_{2+} + X_3 M_{2-} = \overline{X_3} X_2 + X_3 (X_2 \oplus (X_1 + X_0)) \tag{9}$$

In Step II, when unsigned magnitude $M_2M_1M_0$ is obtained, it should be transferred to a comparator for comparison with the threshold. Here, a 1-bit output F is designed to indicate that the absolute value is greater than the threshold. As shown in the formula.

$$F = (M_2 > T_2) + (M_2 = T_2)(M_1 > T_1) + (M_2 = T_2)(M_1 = T_1)(M_0 > T_0) \tag{10}$$

Since both $M_2M_1M_0$ and $T_2T_1T_0$ are unsigned binary numbers, it can be further simplified as follows.

$$\begin{aligned} F &= M_2\overline{T_2} + (M_2 \otimes T_2) M_1 \overline{T_1} + (M_2 \otimes T_2)(M_1 \otimes T_1) M_0 \overline{T_0} \\ &= M_2\overline{T_2} + (M_2 \oplus \overline{T_2})(M_1\overline{T_1} + (M_1 \oplus \overline{T_1}) M_0 \overline{T_0}) \end{aligned} \tag{11}$$

Noticed that the above formula is very similar to the carry expression of a full adder, the similarity has inspired the concept that such circuit can be built by cascaded adders.

$$C_{out} = (A \oplus B) C_{in} + AB \tag{12}$$

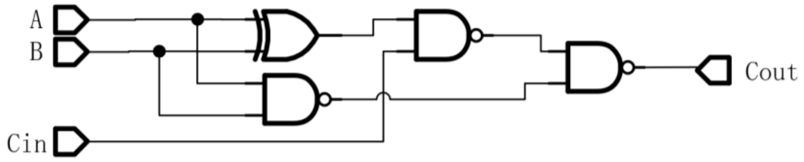

Figure 2. Diagram of a Full Adder (only the carry-out function is considered)

The 3-bit ripple carry adder referring to the design uses carry to implement size comparisons. For design parameters, three full adders are used with carry inputs connected to carry outputs. The input is the signal magnitude M and the bitwise negated threshold T, and the output is a binary number F, which represents whether M is greater than T.

2.3. Design of Circuit Schematic

This is the schematic of the proposed circuit:

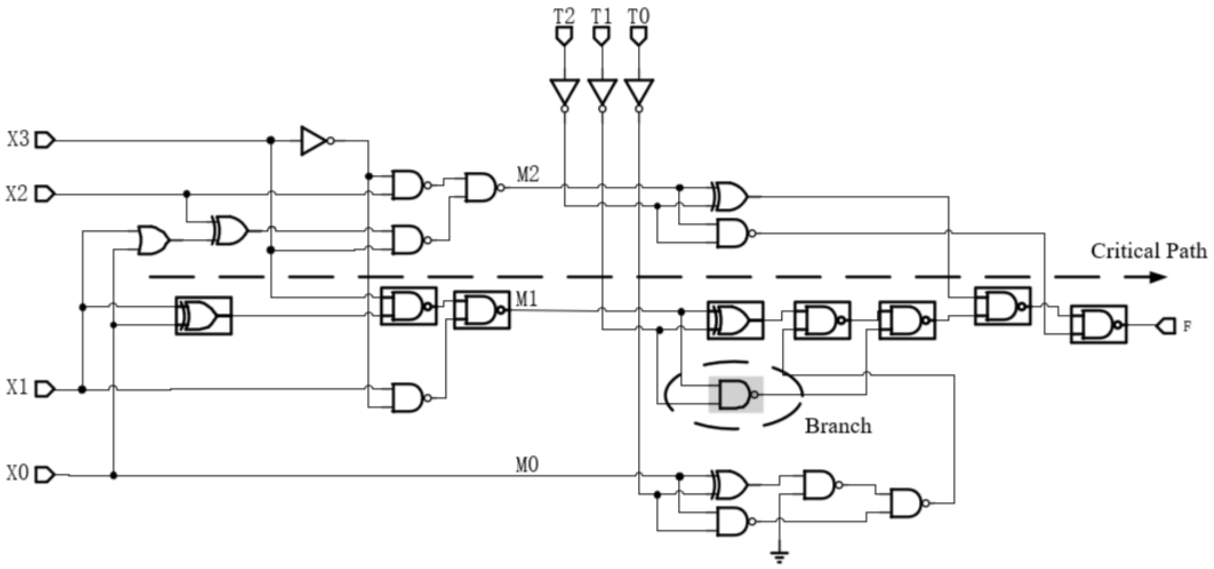

Figure 3. Circuit Schematic with Critical Path and Branch Marked

Within the schematic, $X_3X_2X_1X_0$ represents input signal, $T_2T_1T_0$ represents input threshold, $M_2M_1M_0$ represents absolute value, and F represents the output, where F=1 indicates that $M_2M_1M_0>T_2T_1T_0$ and F=0 otherwise. Critical path has been marked for calculating the delay in the subsequent optimization procedure.

3. Key Parameters of AVD

In the theory of logic effort, the delay of a CMOS gate can be represented as follows:

$$t_p = t_{p0}(p + gh/\gamma) \tag{13}$$

Where t_{p0} represents the intrinsic delay of an inverter;

g is a constant that equals to the ratio between parasitic and gate capacitance;

h is equivalent fanout, defined as the ratio between the gate's external load and internal capacitance. Here h is also called electrical effort;

p describes the intrinsic delay ratio of a gate and an inverter, which depends on not only the circuit topology but also the layout style, and is also called parasitic delay;

g is called logical effort. For a given load, a complex gate must work "harder" than the inverter to get a similar response. Logic effort represents how much more input capacitance a gate exhibits than an inverter when supplying the same output current as an inverter. Since it is only related to the topology of the circuit, the parameter g is significantly useful.

The total delay of path passing through a combinational logic block can be expressed as

$$t_p = \sum_{j=1}^{N} t_{p,j} = t_{p0}\sum_{j=1}^{N}(p_j + g_j h_j/\gamma) \tag{14}$$

In order to optimize the delay to its minimum, the N-1 partial derivatives should be calculated and set to zero, revealing that each stage should have the same gate effort (h times g):

$$g_1h_1 = g_2h_2 = \cdots = g_Nh_N \tag{15}$$

The total logical effort along a path (G) in the circuit can be found by multiplying the logical effort of all the gates on this path:

$$G = g_1g_2\cdots g_N \tag{16}$$

The effective fanout (or electrical effort) H of the path can be also defined as follows, which establishes the relationship between the load capacitance of the last stage in the path and the input capacitance of the first stage:

$$H = h_1 h_2 \cdots h_N = C_{out} / C_{in} \tag{17}$$

In order to relate F to the equivalent fanout of the individual gates, another coefficient must be introduced to account for the logic fanout inside the circuit. When there is fanout on the output of a node, some of the total drive current flows along the path and some leaves the path. Branching effort b of a logic gate on a given path is proposed to describe the influence of the leaving current:

$$b_j = \frac{C_{on-path,j} + C_{off-path,j}}{C_{on-path,j}} \tag{18}$$

The path branching effort is defined as the product of all the branching effort on the path, that is,

$$B = b_1 b_2 \cdots b_N \tag{19}$$

Total path effort (PE) can be determined as

$$PE = GHB \tag{20}$$

Taking branching effort into consideration, the stage effort to minimize path delay is:

$$f^* = \sqrt[N]{PE} = g_1 h_1 b_1 = g_2 h_2 b_2 = \cdots = g_N h_N b_N \tag{21}$$

In this case, the minimum delay through the circuit is:

$$t_p = t_{p0} \sum_{j=1}^{N} (p_j + g_j h_j b_j / \gamma) = t_{p0} (\sum_{j=1}^{N} p_j + N f^* / \gamma) \tag{22}$$

3.1. Critical Path Analysis

3.1.1 Critical Path Parameters

From the input to the output, there is definitely a path whose delay is the maximum among all possible paths within the circuit, which is called the critical path. As shown in Figure 6, the critical path is marked, however, the rest of the circuit has hindered further analysis, and in order to overcome it, we need to extract the critical path first, which is shown in Figure 7.

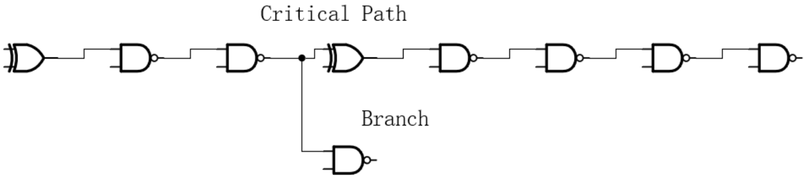

Figure 4. Diagram of Critical Path Only

As shown above, the critical path consists of six NAND2 gates and two XOR2 gates. Besides, the path has a branch that affects the energy consumption of the circuit, which will be discussed later.

Here, the concept of logic effort is applied for its advantages in simplifying circuit analysis and quickly analyzing alternative circuit designs. [4] According to logic effort theory, structures of involved CMOS gates are shown below, followed by their parameters.

(1) Diagrams of Logic Circuit Blocks

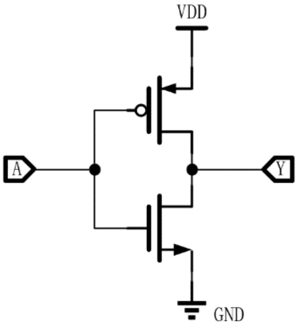

Figure 5. Diagram of CMOS Inverters

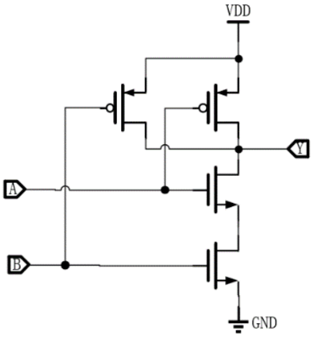

Figure 6. Diagram of CMOS NAND2 Gates

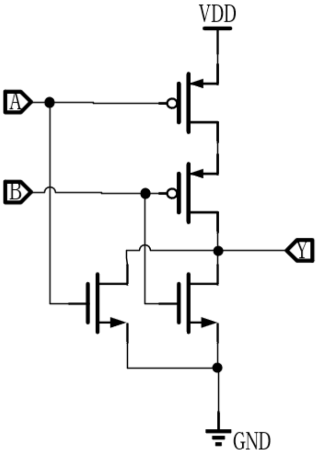

Figure 7. Diagram of CMOS NOR2 Gates

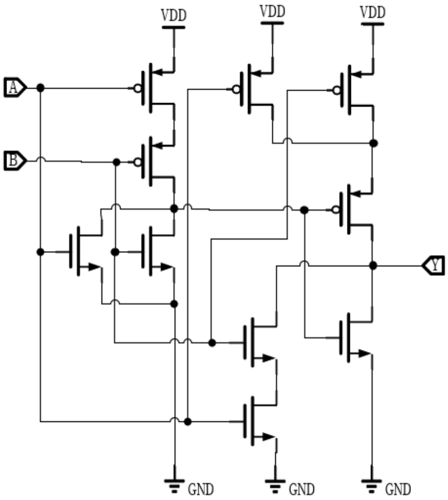

Figure 8. Diagram of CMOS XOR2 Gates

(2) Parameters

According to logic effort theory, correlative parameters, which are logic effort and parasitic delays of applied gates, are shown in Table.1 as follows.

Table 1. Logic Effort and Parasitic Delay of the Logic Gates

Gate Type	Logic Effort (g)	Parasitic Delay (p)
Inverter	1	P_{inv}
NAND2	4/3	$2P_{inv}$
NOR2	5/3	$2P_{inv}$
XOR2	4	$4P_{inv}$

3.1.2 Minimum Path Delay

Referring to the theory of logic effort, with the assumption that V_{DD}=1V, the detailed calculation for the logic effort in the multi-level path is shown below. [7, 8]

Since the logical effort of XOR2 gate and NAND2 gate are:

$$g_{XOR2} = 4, g_{NAND2} = 4/3 \tag{23}$$

Path logical effort (G) is calculated:

$$G=\prod g_i = 4^2 \times (4/3)^6 = 89.9 \tag{24}$$

Path electrical effort (H) is calculated:

$$H = \frac{C_{out}}{C_{in}} = 32 \tag{25}$$

Since there is only one branch (Figure 7), and its branch effort can be expressed as:

$$b_1 = \frac{4+4/3}{4} = 1.33 \tag{26}$$

Path branch effort (B) is calculated:

$$B = \prod b_n = b_1 = 1.33 \tag{27}$$

Path effort (PE) is calculated:

$$PE = GHB = 3826.1 \tag{28}$$

Best stage effort (f*) is calculated:

$$f^* = \sqrt[N]{PE} = \sqrt[8]{3826.1} = 2.80 \tag{29}$$

Since the parasitic delay of XOR2 gate and NAND2 gate are:

$$p_{XOR2} = 4, p_{NAND2} = 2 \tag{30}$$

Path parasitic delay (P) is calculated:

$$P=\sum p_n = 4\times 2+2\times 6=20 \tag{31}$$

Under the assumption that

$$\gamma = \frac{C_{parasitic}}{C_{gate}} = 1 \tag{32}$$

Path delay can be expressed as:

$$d_n = g_n h_n + p_n \gamma = g_n h_n + p_n \tag{33}$$

Minimum path delay (D) is calculated:

$$D = \sum (g_n h_n + p_n) = Nf^* + P = 8 \times 2.80 + 20 = 42.40 \tag{34}$$

3.1.3 Critical Path Gate Sizing

After the best stage effort f* is determined, each gate within the critical path can be sized using the following formula: [9]

$$C_{in,n} = g_n b_n \frac{C_{out,n}}{f^*} \tag{35}$$

Here, calculation results are shown here, numbered from left to right.

$$\begin{gathered} C_{in,1} = 1.02, C_{in,2} = 0.71, C_{in,3} = 1.49, C_{in,4} = 3.13 \\ C_{in,5} = 1.65, C_{in,6} = 3.46, C_{in,7} = 7.26, C_{in,8} = 15.24 \end{gathered} \tag{36}$$

3.2. Delay Comparison Between Different Paths

The above derivations are based on the correct selection of the critical path. Once the critical path is incorrectly selected, the calculated results have no effect. In order to verify the correctness of the chosen critical path, and to reinforce the notion that the critical path has the longest delay, two possible and typical potential critical paths (actually non-critical) will be discussed as follows.

(1) Non-critical path 1

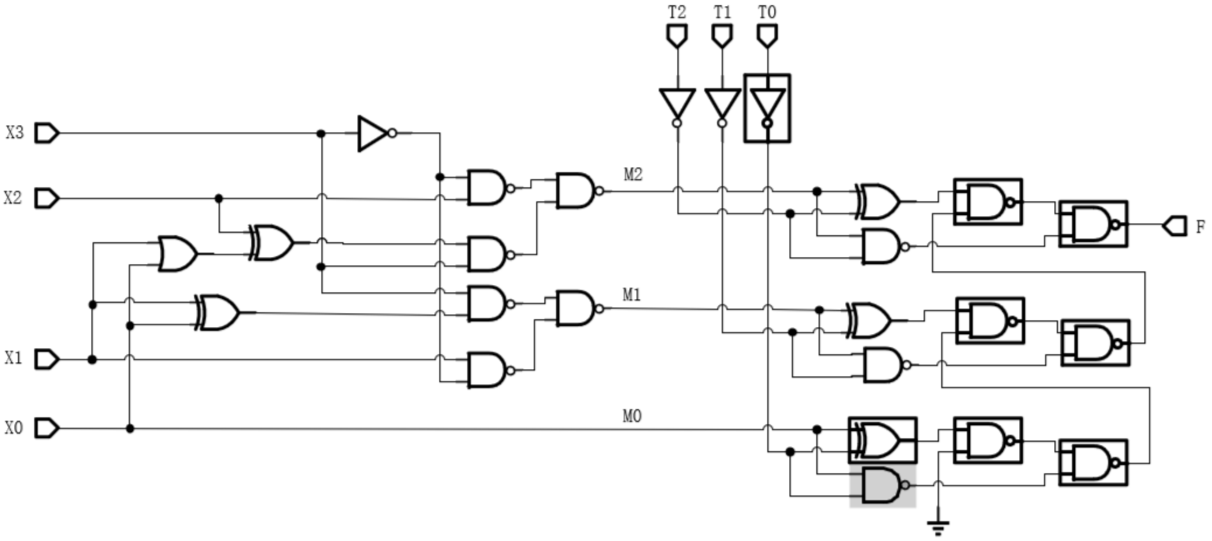

Figure 9. Detailed Circuit Schematic with Non-critical Path 1 and Branch Marked

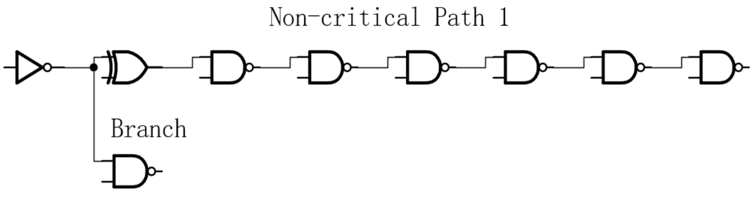

Figure 10. Diagram of Non-critical Path 1 Only

The minimum path delay (D) can be calculated in the same way as the critical path.

$$D = 35.88 \tag{37}$$

Compared to the critical path, replacing an XOR gate with an inverter reduces total path delay to 35.88, which is 15.4% less than the delay of the critical path.

(2) Non-critical path 2.

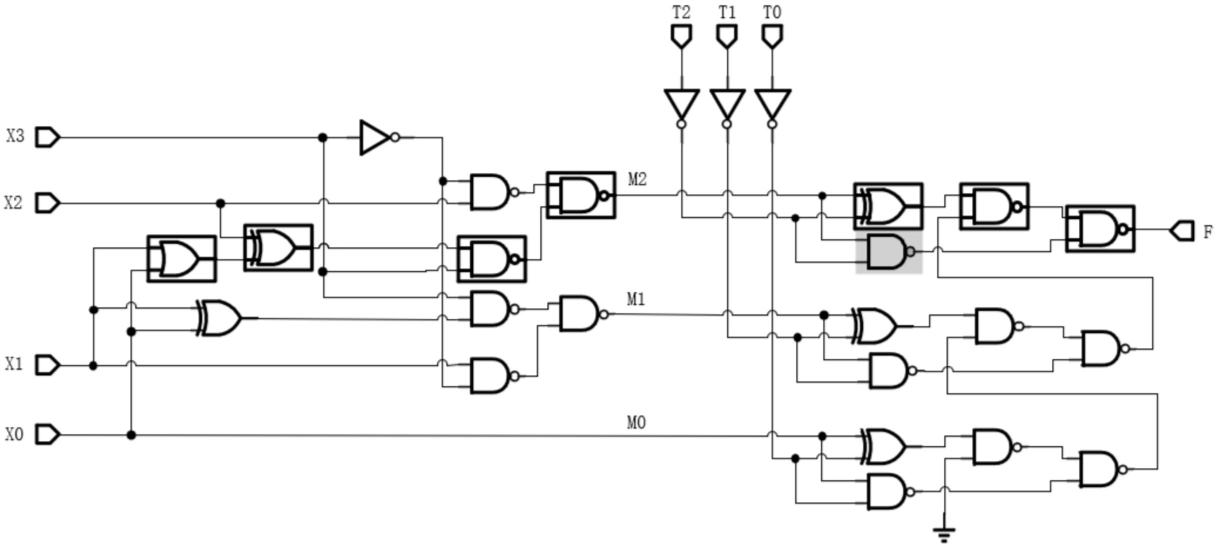

Figure 11. Detailed Circuit Schematic with Non-critical Path 2 and Branch Marked

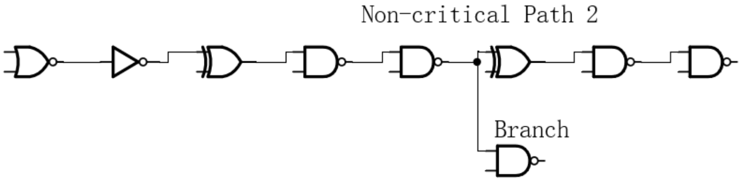

Figure 12. Diagram of Non-critical Path 2 Only

The minimum path delay (D) can be calculated in the same way as the critical path.

$$D = 41.24 \tag{38}$$

Compared to the critical path, replacing a NAND gate with an inverter and replacing another NAND gate with an NOR gate reduce total path delay to 41.24, which is 2.7% less than the delay of the critical path. The differences can be clearly distinguished in the following table (Table.2).

Table 2. Delay Comparison of Different Paths

Paths	Delay	Ratio to Critical Path Delay
Critical Path	42.40	1
Non-critical Path No.1	35.88	0.8462
Non-critical Path No.2	41.24	0.9726

3.3. Optimal Voltage of VDD (VDD^{opt}) Calculation

Assume that the target optimal voltage for VDD ranges from 0V to 1V, which provides the maximum delay that is 1.5 times of the minimum path delay. [10]

Assuming that

$$V_{DD} = 1V \tag{39}$$

Energy consumption at non-optimal VDD is calculated:

$$E = C_L V_{DD}^2 = 32.00 \tag{40}$$

Assuming that VT=0.2V, following equations can be used in order to find the optimal VDD:

$$D = \frac{kV_{DD}}{(V_{DD} - V_T)^2} \tag{41}$$

$$1.5D = \frac{kV_{DD}^{opt}}{(V_{DD}^{opt} - V_T)^2} \tag{42}$$

$$V_{DD}^{opt} = 0.776V \tag{43}$$

Energy at optimal VDD is calculated as

$$E_{opt} = C_L V_{DD}^{opt\,2} = 19.27 \tag{44}$$

Which is 39.8% lower than that of non-optimal VDD.

4. Summary of Key Characteristics of the Circuit

When VDD changes from 1V to 0.776V, although the critical path delay increases by 50%, the power consumption decreases by 39.8%.

Table 3. Voltage and Energy Characteristics

VDD	Critical Path Delay	Energy Consumption
1V	$tp_{\_IN \to OUT}$=42.40 FO4(1V)	E=32.00$E_{u(1V)}$
0.776V	$tp_{\_IN \to OUT}$=63.60 FO4(1V)	E=19.27$E_{u(1V)}$

From the above table, such change can reduce the energy consumption considerably without significantly increasing the delay. At the same time, smaller energy consumption also means lower operating temperature.

5. Conclusion

In this paper, a design that suits both the delay goal and the energy goal has been designed and optimized, achieving the proposed 4-bit absolute-value detector with the least energy consumption for a tolerable delay that is 150% of the minimum delay, with circuit style figured out, block-level schematic of AVD implemented, critical path identified and sizing optimized. Since the design consists of relatively few gates and short critical path, it performs well in terms of power consumption and delay.

Aside from the advantages it has, yet there are still some flaws. As there are a plenty of XOR gates implemented in the design, which contribute to a large part of the delay and energy consumption, further improvement could include using methods like mirror adders to reduce the quantity. Also, since the delays of different circuits are different, the input signals arrive at the same node asynchronously, which may lead to race hazards. For this reason, simulation software can be used to optimize the circuit, such as substituting CMOS gates with transmission gates or even simply adding storage units as buffers.

References

[1] Casson A, Rodriguez-Villegas E. Toward online data reduction for portable electroen-cephalography systems in epilepsy [J]. IEEE Transactions on Biomedical Engineering, 2009, 56(12): 2816-2825.

[2] Iranmanesh S, Raikos G, et al. CMOS implementation of a low power absolute value comparator circuit [C]. 2016 14th IEEE International New Circuits and Systems Conference (NEWCAS), 2016.

[3] Goel S, Kumar A and Bayoumi M. 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, 2006, 14(12): 1309-1321.

[4] Yu Zhan, Yu Menglin, et al. The use of reduced two's-complement representation in low-power DSP design [C]. 2002 IEEE International Symposium on Circuits and Systems (ISCAS), 2002.

[5] Sutherland I, Sproull R F, Sproull B, et al. Logical effort: designing fast CMOS circuits [M]. Morgan Kaufmann, 1999.

[6] Hirata M, Nagashima S, et al. A novel electrostatic ion-current absolute-value detector under circumstances with simultaneous electron incidence by the use of a proposed “Self-Collection” method for secondary electron emission [J]. Fusion science and technology, 2003, 43(1): 262-264.

[7] Sutherland I, Sproull R. Logical effort: designing for speed on the back of an envelope [J]. IEEE Advanced Research in VLSI, 1991.

[8] Levi I, Belenky A, Fish A. Logical effort for CMOS-based dual mode logic gates [J]. IEEE Transactions on very large scale integration (VLSI) systems, 2013, 22(5): 1042-1053.

[9] Wang C, Markovic, D. Delay estimation and sizing of CMOS logic using logical effort with slope correction [J]. IEEE Transactions on Circuits and Systems II: Express Briefs, 2009, 56(8): 634-638.

[10] Markovic D, Nikolic B, Brodersen R. Analysis and design of low-energy flip-flops [C]. In Proceedings of the 2001 international symposium on Low power electronics and design. 2001: 52-55.