A 93.56 FO4(1V), 141.36 Eu(1V) 4-bit Absolute-Value Detector

Qianqing Wang [1,*]

[1]School of engineering, Huazhong University of Science and Engineer, Wuhan, China

*Corresponding author: 15030240144@xs.hnit.edu.cn

Abstract: With the development of technology and The Times, the industry has higher and higher requirements for chip performance. Using better circuit design to achieve this goal has become the direction of researchers today. The 4-bit absolute value detector is one of the most important and basic circuits in computer storage and data processing. The effective improvement of its performance will greatly promote the improvement of processing efficiency in the whole chip industry. This project is to design a circuit of a 4-bit absolute value detector. The main functions it implements are it can get the absolute value of 2's complement, then compare the 3-bit absolute value with a 3-bit threshold value. This paper first divides the design into three parts: absolute module, comparison module and other small modules. Minimize the delay by exploring the critical path and changing the gate size. Finally, through theoretical calculation and optimization, the optimal scheme designed in this paper is determined. The results of this paper will be helpful to the further development of related industries.

Keywords: MOS, comparator, absolute, MATLAB.

1. Introduction

With the rapid development of the information society, our demand for chip performance is getting higher and higher [1]. Because of the limits of Moore's Law, researcher used to rely on reducing the chip size to achieve smaller energy, and the delay method is no longer effective. Therefore, using better circuit design to achieve this goal has become the direction of today's researchers [2]. The 4-bit and threshold comparator this paper studied this time is the most basic circuit in the computer, and if can simplify it, it can greatly improve the efficiency of chip operation. Although this is only a small part of the chip, the result of the painstaking efforts of all researchers can lead to a more high-speed computing chip [3].

The purpose of this paper is to build a low-power threshold comparator using CMOS transistors that use less material and better designs. As a high-precision rectifier, it may make the accuracy get improvement by absolute-Value detector and comparator [4]. This project is to design a circuit of a 4-bit absolute value detector that can get the absolute value of 2's complement, then compare the 3-bit absolute value with a 3-bit threshold value. If the absolute value is larger than the threshold value, the output will be 1 [5]. So, to realize the function, this paper should complete two parts: absolute module, comparator module.

2. Module design

Overview the project, this paper can divide it into three parts: absolute module, comparator module and other small modules. To make the circuit clearer, this paper modularize it into small parts and then use the blocks to expand the circuit from small too big. The module design is shown in figure 1 below [6].

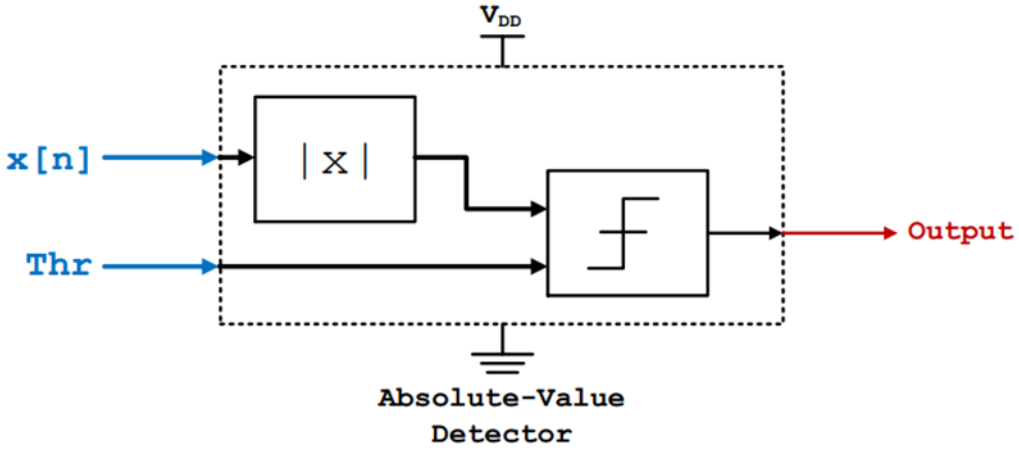

Figure 1 Module design

After this part, the absolute 3-bit value goes to the next stage, which is the comparator. The comparator will compare the absolute value and a 3-bit threshold value. If the absolute value is larger than the threshold value, the output will be positive, and vice versa [7].

2.1 Absolute module

When come to the absolute value module, this paper know that 4-bit binary numbers can represent -8 to 7 in decimal. When the highest bit is 1, it means a negative number and when the highest bit is zero, it means a positive number, which requires us to use the first bit The highest bit to determine whether this paper need to process the following 3-bit number. So, this paper chose the digital selector mux, through the highest bit 0 and 1, to select the positive number that is not processed or the negative number that is negated and added one. To get the absolute value, this paper should inverse every bit of the input and add 1, then the answer is the absolute value of the input. If the highest bit is zero, it means positive. The truth table is shown in table 1 below [8].

Table 1 Truth table

X3X2X1X0	number	X3X2X1X0	number
0111	7	1111	-1
0110	6	1110	-2
0101	5	1101	-3
0100	4	1100	-4
0011	3	1011	-5
0010	2	1010	-6
0001	1	1001	-7
0000	0	1000	-8

The first step was to design the schematic of the absolute value detector. 2:1 MUX is used to determine whether the 4-bit input is positive or negative, and output the absolute value by not changing or flipping the last 3 bits. The combination of 3 half adders will add 1 to the output from 2:1 MUX to ensure 2's complement is done. In this way this paper can know the input is a negative or positive number. And then using the X3 and half adder this paper can get the absolute number [9].

Then transform it into the comparator module. In absolute value circuit part, there are two corresponding solutions to realize it. The first solution of absolute value circuit is combinational logic style which means directly constructing the circuit through logic expressions that are based on truth table. The logical operation formulas are as follows in figure 2.

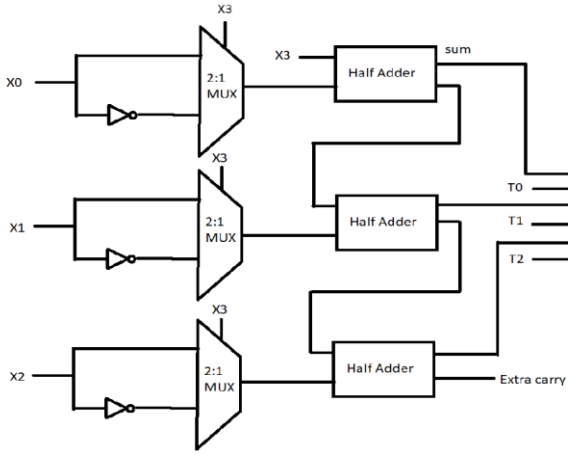

Figure 2. Absolute module

2.2 Comparator module

After this part, the absolute 3-bit value goes to the next stage, which is the comparator. The comparator will compare the absolute value and a 3-bit threshold value. If the absolute value is larger than the threshold value, the output will be 1, and vice versa [10].

For 3-bit input, this paper can get the equation below:

$$(A > B) = A_3\overline{B_3} + XA_2\overline{B_2} + X_3X_2A_1\overline{B_1} + X_3X_2X_1A_0\overline{B_0}$$

$$Where X_i = A_iB_i + \overline{A_i}\,\overline{B_i} \tag{1}$$

So, to make it come true, the comparator circuit is shown in figure 3 below.

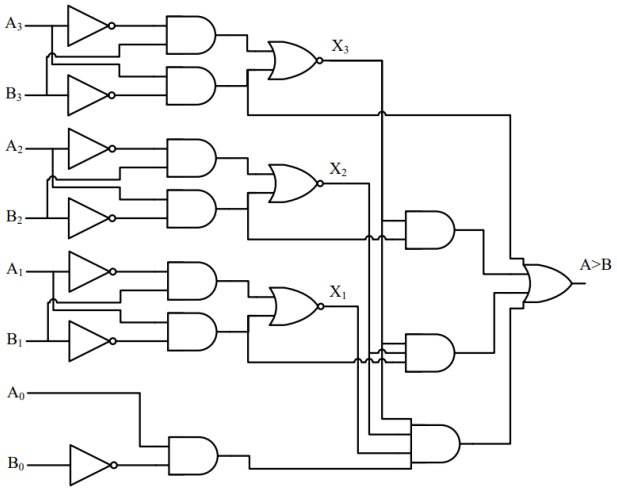

Figure 3 Comparator circuit

2.3 Total circuits

It is easy for us to get the total circuits by linking each output from absolute-value module to the comparator module's input. Also, the output from comparator module is what this paper want to get which is above or below the threshold number in figure 4.

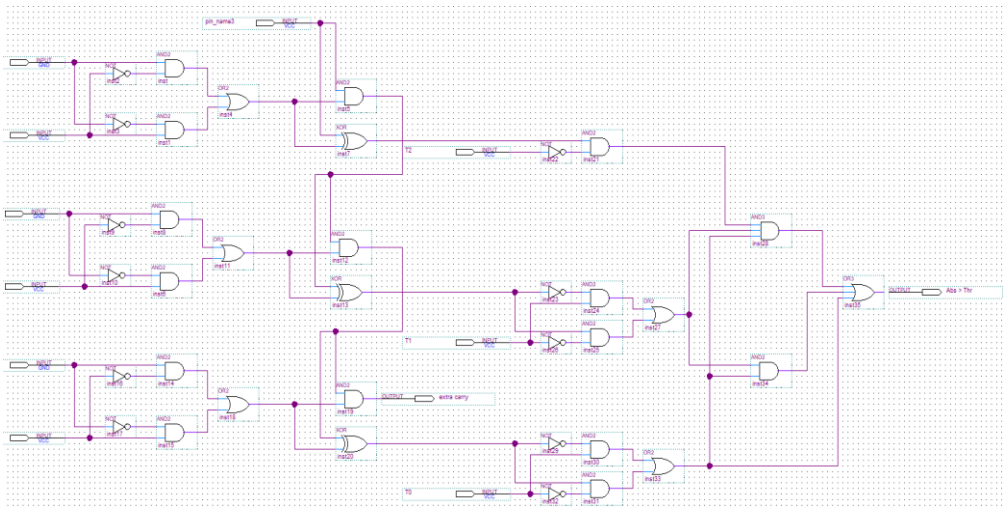

Figure 4 Total circuits

3. Calculate and optimize

3.1 Calculate

Firstly, this paper finds the main path of this circuit. And it been known that AND, OR and XOR are not easy to be generate, so this paper transforms them into NAND, NOR which are easy to get. Moreover, this paper simplifies this circuit by using the theory below. the transform therapy is shown in figure 5 below

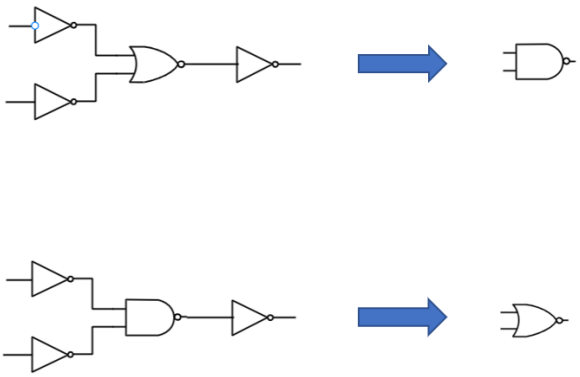

Figure 5 Therapy

It is a good way to say the main stage shorter than before, which is not only beneficial for calculating but also a good way to get shorter delay. That is the reason why this paper must transform them into that way. So, it is easy to get the optimized circuit below. the conclusion is shown in figure 6 below.

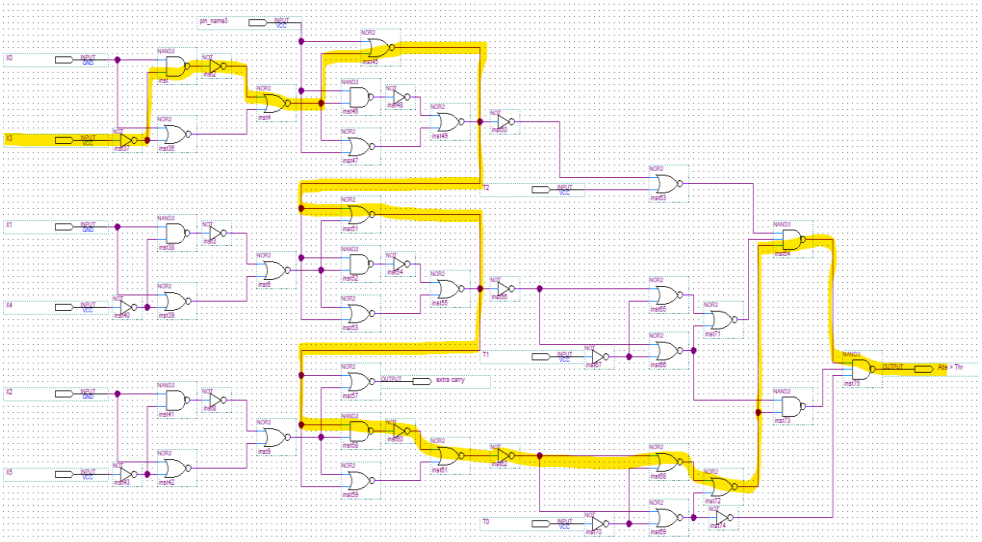

Figure 6 Conclusion

In this project this paper assume the unit-sized inverter is Wp equal to 650nm and equal to 430nm. And then this paper can find the WP: WN is equal to 1.5. After drawing the graph of electronic components and analyzing, it is not difficult to get the parasitic and logical effort of them. the Path parasitic delay is shown in table 2 below.

Table 2 Path parasitic delay

	INV	NAND	NOR
P	1	2	2
G	1	1.4	1.6

And then, when this paper come to this critical map, it is easy to find the main path and the branch effort of each component and know that there are 16 stages of our main path. This paper consider each path and select this which not only have the shortest stage but also it has the maximum effort from the branches. The main path is shown in figure 7 below.

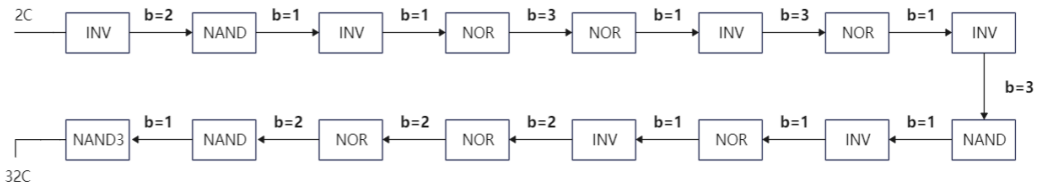

Figure 7 Main path

3.1.1 Calculating critical path delay (VDD=1V)

When comes to the main path, this paper calculate the logic effort in the multi-level path, which is shown below.

$$G = \Pi g_i = 82.87 \tag{2}$$

Path electrical effort can be generated:

$$H = \frac{C_{out(path)}}{C_{in(path)}} = 32 \tag{3}$$

Path branch effort can be generated: $B = 432$

Path effort can be generated.

$$F = GBH = 572769.97 \tag{4}$$

$$\hat{f} = \sqrt[N]{F} = 2.29 \tag{5}$$

Path parasitic delay can be generated

$$P = 26 \tag{6}$$

Total path delay can be generated

$$D = N\hat{f} + P = 63.04 \tag{7}$$

3.1.2 Gate sizing of the critical path

As we known the C_l of the output bit of the critical path is a 32-bit inverter. This paper calculate the size of the critical path by the formula below, it can be worked out.

this paper use the formula below in order to calculate the path effort (F).

$$F = \prod (g \times h) \tag{8}$$

Also, get the stage effort by formula below:

$$f = g \times h \tag{9}$$

This paper can gate sizing by this formula:

$$C_{in} = g \times \frac{C_{out}}{f} \tag{10}$$

And then, this paper get the sizing of the critical path gates from left to right. the Cin is shown in table 3 below.

Table 3 Each Cin

	1	2	3	4	5	6	7	8
Cin	2.00	2.29	3.75	8.59	4.10	5.87	4.48	6.41
9	10	11	12	13	14	15	16	17
4.89	8.00	18.32	26.22	30.03	21.49	15.38	25.15	32

Table 3: each Cin

3.1.3 Calculate the consulting VDD

The VDD is optimal voltage for the VDD of 0v -1v. this paper optimize the circuits by using the maximum delay which is 1.5x minimum delays.

Using expression shown below, this paper can generate the VDD by using tp:

$$t_p = \frac{C_{VDD}}{K'(VDD - V_T)^2}, K' = \frac{W}{L}K \tag{11}$$

the maximum delay is 1.5 times of minimum delays, this paper assume VT= 0.2v:

$$D = \frac{V_{DD}}{(V_{DD} - V_T)^2}. \tag{12}$$

$$V_{DD} = 0.7758V \tag{13}$$

This paper can get VDD equal to 0.776V, the critical path delay can be generated:

$$D_1 = 1.5D = 94.56 \tag{14}$$

3.2 The total energy

$$E = C_l V_{DD}{}^2 = 186.97 \tag{15}$$

This paper get those consulted results by excel, which is easy for us to get the right data step by step. There more calculation detail is shown in table 4 below. This paper sign each important data in the table and calculation result.

Table 4. Calculation details

	type	G	B	H	P	Delay	size	Energy	VDD scaling
MUX	INV	1	2	2.29	1	3.29	2.00	2.00	1.20
	NAND	1.4	1	1.64	2	4.29	2.29	2.29	1.38
	INV	1	1	2.29	1	3.29	3.75	3.75	2.25
	NOR	1.6	3	1.43	2	4.29	8.59	8.59	5.16
HF1	NOR	1.6	1	1.43	2	4.29	4.10	4.10	2.46
HF2	INV	1	3	2.29	1	3.29	5.87	5.87	3.52
	NOR	1.6	1	1.43	2	4.29	4.48	4.48	2.69
HF3	INV	1	3	2.29	1	3.29	6.41	6.41	3.85
COM	NAND	1.4	1	1.64	2	4.29	4.89	4.89	2.94
	INV	1	1	2.29	1	3.29	8.00	8.00	4.81
	NOR	1.6	1	1.43	2	4.29	18.32	18.32	11.01
	INV	1	2	2.29	1	3.29	26.22	26.22	15.75
	NOR	1.6	2	1.43	2	4.29	30.03	30.03	18.03
	NOR	1.6	2	1.43	2	4.29	21.49	21.49	12.91
	NAND	1.4	1	1.64	2	4.29	15.38	15.38	9.24
	NAND3	1.8	1	1.27	2.4	4.69	25.15	25.15	15.11
		82.87	432	16	26	63.04	32	186.97	112.30

3.2 Optimize

After calculating the energy under minimum delay, Excel Solver is used to find the energy under 1.5x delay. When gate sizing is the only variable that increases delay to 1.5x, the energy is 112.30 Eu(1V). When VDD is the only variable that increases delay, VDD will be 0.776V, and energy is 112.30Eu(1V).

Moreover, we know that if increase the delay reach to 1.5 times of minimum delay (93.56), the total energy will decrease in 141.36. it is a good way to decline the energy. However, it is not enough. This paper also try to decrease the VDD scaling and gate sizing in order to decline that energy which is a beneficial for delay and energy. And for combinations of VDD scaling and gate sizing, the percentage of each of their contribution to delay varies, and the energy will be calculated by Excel Solver. Below is the chart of energy results generated from Excel Solver.

Table 4 Excel Solver

VDD%	10%	25%	40%
VDD	0.939	0.866	0.808
Gate sizing%	40%	25%	10%
Energy	116.88	116.51	108.86

It is found that as VDD scaling contributes more to delay, energy decreases more, but the actual relationship between energy and VDD scaling can be different.

4. Conclusions

4.1 Summary

Overview the project, this paper can divide it into three parts: absolute module, comparator module and other small modules. Our design generates minimum delay by gate sizing, and this paper used half adders only for the adder section. Also, for our design, this paper only used NOR, NAND and NOT gates, which means that this paper converts the original AND, OR and XOR gates to those gates that can be implemented by static CMOS directly. This paper discuss and design the sizing on non-critical paths to reduce energy consumption. Because when this paper calculates the delay of its entire circuit, this paper only calculate the delay of its critical paths. The delay of non-critical path is not so important. So, without affecting the delay of critical paths (in another word, this paper are going to

ensure that Delay of non-critical paths should be not greater than critical), Reduce devices' size on unimportant path is OK.

4.2 Discussion

First is that this paper can change the topology for getting shorter critical path. This paper notice that the delay and energy are much lower in the presentation of other groups, this may be caused by the topology difference. This paper can try carry look-ahead adders for shorter critical path, and may get a better value. Second, this paper can find the best combination of gate sizing and VDD scaling. Different contribution of VDD scaling and gate sizing to delay can vary, so a good combination of VDD and gate sizing can lead to the best delay and energy. At last, this paper can also do gate sizing partially instead of globally. And this paper can try MATLAB instead of Excel Solver in optimization part. When this paper was doing the optimization section with Excel Solver, there were some glitches and either cannot get results, or got strange results. MATLAB may be able to solve those problems that met.

References

[1] Maheshwari S., Raghav H.S., Gupta A. Characterization of Logical Effort for Improved Delay.VLSI D-esign and Test. Communications in Computer and Information Science, vol 382, P108-117,2013.

[2] J. Yuan, C. Svensson. High-speed CMOS circuit technique. IEEE Journal of Solid-State Circuits, Volume: 24, P62-70,1989.

[3] Mehmet A. Chitt. TRANSISTOR SIZING IN CMOS CIRCUITS. Proceedings of the 24th ACM/IEEE, Design Automation Conference, P 121-124,1987.

[4] T. Sakurai, A.R. Newton. Delay analysis of series-connected MOSFET circuits. IEEE Journal of Solid-State Circuits, Volume: 26, Issue: 2, P122-131,1991.

[5] Sakurai, Takayasu and A. Richard Newton. "A simple MOSFET model for circuit analysis." IEEE Transactions on Electron Devices 38 (1991): 887-894.

[6] Sakurai, Takayasu and A. Richard Newton. "Delay analysis of series-connected MOSFET circuits." IEEE Journal of Solid-state Circuits 26 (1991): 122-131.

[7] Shur, Michael S., Tor A. Fjeldly, Trond Ytterdal and Kie Young Lee. "Unified MOSFET model." Solid-state Electronics 35 (1992): 1795-1802.

[8] Elzakker, M.V., Tuijl, E.V., Geraedts, P.F., Schinkel, D., Klumperink, E.A., & Nauta, B. (2010). A 10-bit Charge-Redistribution ADC Consuming 1.9 μW at 1 MS/s. IEEE Journal of Solid-State Circuits, 45, 1007-1015.

[9] Hedenstierna, N., & Jeppson, K.O. (1987). CMOS Circuit Speed and Buffer Optimization. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 6, 270-281.

[10] T. Sakurai; A.R. Newton. Alpha-power law MOSFET model and its applications to CMOS inverter delay and other formulas. IEEE Journal of Solid-State Circuits ,Volume: 25, Issue: 2,pages 584-594,1990.