

Optimized Design of a 4-bits Absolute-Value Detector

Gengfu Yang

Department of Engineering Columbia University, New York, America

gy2299@columbia.edu

Abstract. 4-bits Absolute Value Detector circuit is a very commonly used circuit design. As to simplify the internal circuit design to make the total number of stage lesser which provides a less critical path circuit delay. Generally, this paper used combination of static CMOS design and the path transistor circuit design to design our circuit. After we got the optimized design, we applied the logic effort calculation to calculate the minimum delay of the circuit and determined the specific gate sizing value for each logic component. The energy optimization can be obtained by increasing our delay and adjusting the power supply (VDD) value. Eventually, we sacrifice our delay a little bit to get a huge amount of energy reduction.

Keywords: Adder circuit, Logic Effort, Energy Optimization.

1. Introduction

As the development of the modern technology, all the electronic products update faster and faster. Nowadays, the technology almost reaches Moore's law boundary [1]. Improvement of circuit speed and reduction of circuit energy consumption become hot topics. Throughout this acritical it introduces a way of 4-bits Absolute -Value Detector. A 4-bits Absolute-Value Detector can be used in plenty of area. Most commonly in a chip ALU unit. Since this circuit consists of adder circuit and comparator circuit, which are also broadly used in chip's ALU unit, a good optimization of 4-bits Absolute-Value Detector is very useful. This article mostly focuses on the circuit topology optimization, delay calculation, and delay and energy optimization by adjusting gate sizes and power supply value. In the following section it introduces the circuit topology optimization, delay calculation, and sizing and power supply (Vdd) optimization. Specifically, the second part is about the circuit topology, and it consists of an optimized adder circuit, two pass transistor design multiplexer, and an optimized comparator circuit. The third part is about the delay calculation. Firstly, used the circuit topology from part 2 to determine its critical path delay, then used the logic effort formula to calculate its circuit delay and the detail sizing of each logic gate. The last part is about the energy optimization. In total there are three different ways only sizing optimization, voltage optimization, and sizing plus voltage optimization to obtain an energy reduction.

2. Circuit topology and circuit style

2.1. Top-Level block diagram & 2's complementary calculations

The ultimate goal of this project is to design a 4-bits absolute value detector with a minimum energy for delay that is 50 percent longer than its minimum delay [2]. And the whole circuit is divided into two main parts, the first part is finding the magnitude value of a 4 bits random variable inputs, and the second part is comparing the calculated value with a threshold value and outputs the larger value. The top-level block diagram as shown in Figure 1.

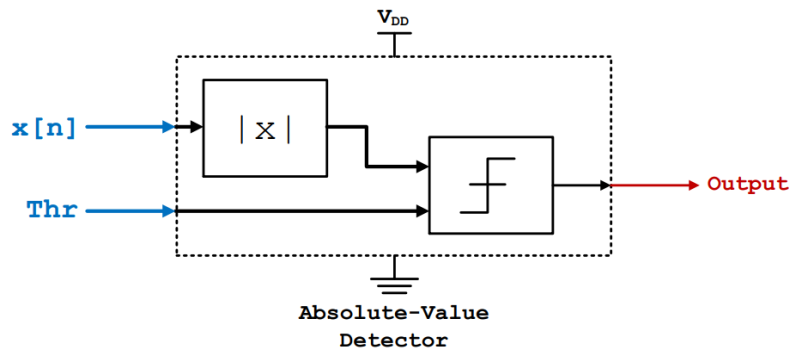


Figure 1. Top level block diagram

The Figure 1 shows the general top-level structure of the absolute value detector design. $x[n]$ is a random 4-bits value in 2’s complement format [3]. Basically, a 2’s complement value has its most significant bit (MSB) represent the value is positive or negative, when the most significant bit (MSB) value is 1 the value is negative, and when the most significant bit (MSB) value is 0 the value is positive. In our case, we only have 4-bits value, and the most significant bit (MSB) is a sign bit, so we only have 3-bits value to represent its magnitude. If the value is positive, the magnitude is just the 3-bits value itself in binary. If the value is negative, then we need to flip all bits and add 1 to it to get the magnitude value.

Some examples about how to calculate the magnitude of the 4-bits values from input $A_3A_2A_1A_0$ to the magnitude output value $S_3S_2S_1S_0$. After the magnitude calculation the output value will go to the next step which is a comparator. In the figure1 the small box on the right represents the comparator. In this step it will compare the calculated result from the first step and compare it with a threshold value. The final output is 1 if the value is larger than the threshold value, and it is 0 if the value is less than or equal to the threshold value. The detail information as shown in the table 1 below.

Table 1. 2’s complement value calculation example

A_3 (Sign bit)	$A_3A_2A_1A_0$ (Input value)	$A_3A_2A_1A_0$ (Flipped all bits)	$A_3A_2A_1A_0$ (Add one)	$S_3S_2S_1S_0$ (Magnitude value)
1 (negative)	1011 (-5)	0100	0101	0101 (+5)
0 (positive)	0111 (+7)	None	None	0111 (+7)

2.2. Circuit topology of magnitude value calculator

The magnitude value calculator consists of two parts. Generally, when the value is positive, we don’t need to do any calculation, and the directly outputs the lower 3-bits value. If the value is negative, we need to follow the steps as described in table1 to get the magnitude values. Since we only have two values ($A_3A_2A_1A_0$ and 1) add together in the negative case, we only need half adder to do the calculation [4]. The figure2 shows the half adder circuit topology.

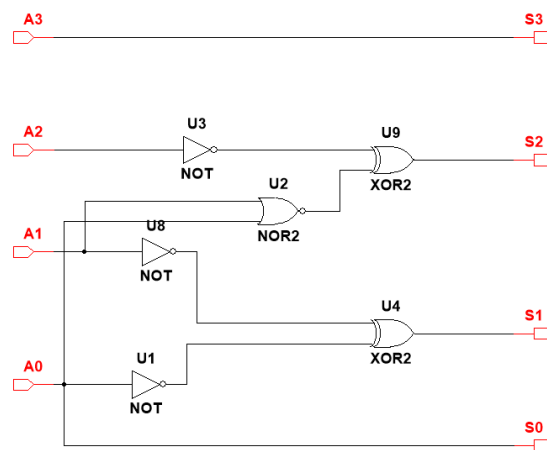


Figure 2. Half adder topology

It consists of two simplified half adders. A normal half adder has XOR gate to calculate its sum (1) and an AND gate to calculate its carry (2).

$$Sum = A \oplus B \tag{1}$$

$$C_{out} = AB \tag{2}$$

In our case, the less significant bit value (LSB) A0 always pluses 1 in the negative case. According to the equation (1) $A0' \oplus 1$ is just A0 itself. Therefore, we delete the XOR gate for A0. As shown in the figure2 the A0 directly outputs to S0. According to the equation (2) the A0 carry out calculation is $C_{out} = A0' 1$ there for the carry out value is just $A0'$ itself. So, the bottom AND gate is also removed and the $A0'$ is directly connected to the next stage. For the A1 carry out calculation, it also follows the equation (2) $C_{out} = A0' A1'$. Because the AND gate has two stages in a complementary CMOS design which has more delay and more power consumption than a single nor gate. According to the formula (3), we use the De Morgan's law to change the AND gate into a two input nor gate [5].

$$A1' A0' = (A1 + A0) \tag{3}$$

Generally, by comparing our simplified half adder circuit in figure 2 and the original half adder design in figure 3. Our simplified version only has two stages for its critical path delay and has a smaller number of gates than the traditional half adder design. A traditional half adder topology as shown in the Figure 3 below.

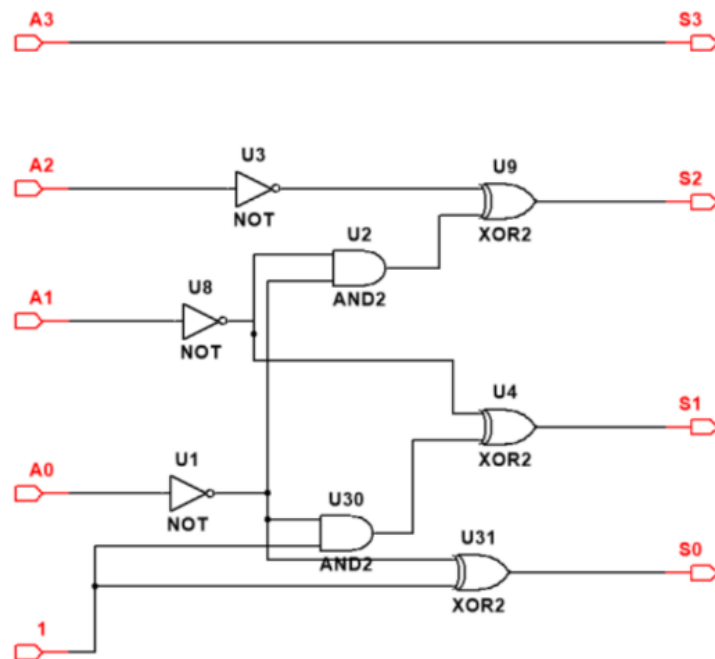


Figure 3. Traditional half adder topology

2.3. MUX design in magnitude calculator circuit

A multiplexer (MUX) is needed to select which values will feed into the next comparator stage. If the A3 is 0 which means the value is positive the multiplexer (MUX) will output A2A1A0, and if the A3 is 1 which means the value is negative the multiplexer (MUX) will output S2S1S0. Following the figure 2 schematic we add the two multiplexer (MUX) circuit to do the selection. The overall magnitude calculator shcematicas shown in Figure 4 below.

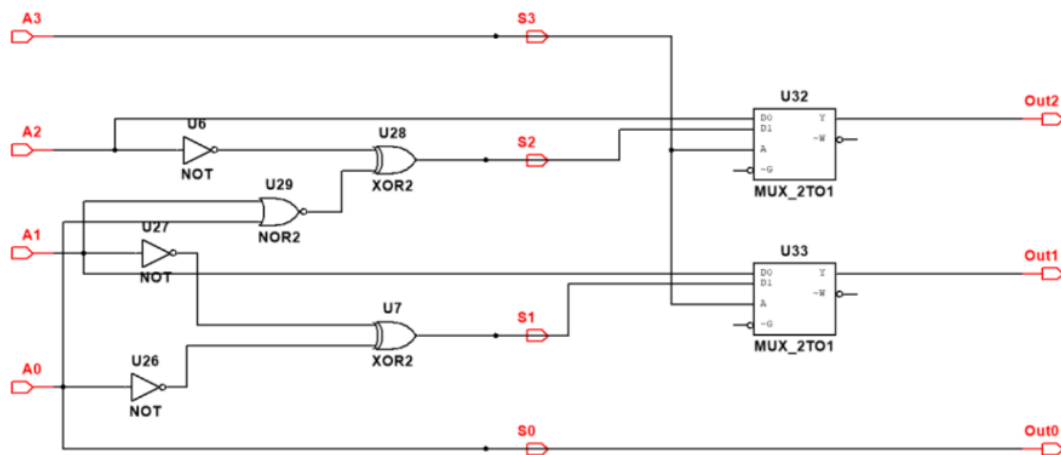


Figure 4. Overall magnitude calculator schematic

According to the figure 4 schematic the multiplexer (MUX) is just a black box. Figure 5 shows the detail inner structure of our multiplexer (MUX) design. We used the pass transistor structure design, and it also has an inner inverter in it. According to figure 5 C is the selection bit, which connects with A3, and it will select between A and B. The reason why we used pass transistor design instead of complementary CMOS design is because pass transistor has fewer number of gates and less stages than the complementary CMOS for the multiplexer (MUX) design [6]. The detail schematic as shown in Figure 5 below.

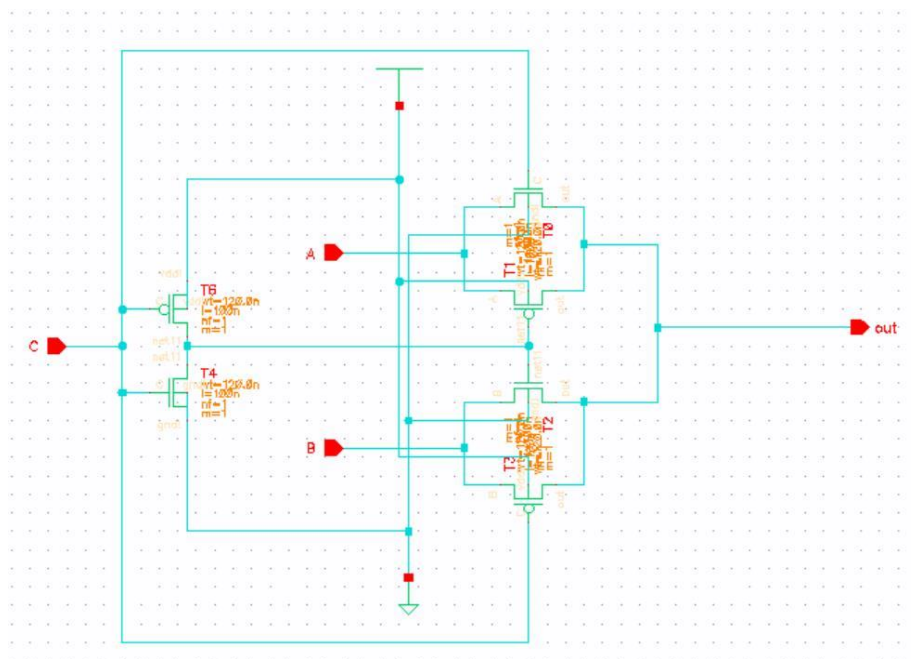


Figure 5. Pass transistor design of the MUX

2.4. Comparator circuit topology

The last step is the comparator circuit. Basically, we compare the output value from the multiplexer (MUX) with a threshold value (B2B1B0). If the threshold value is larger, the output is 0, and if the threshold value is smaller, the output is 1. Figure 6 below shows the comparator circuit schematic [7].

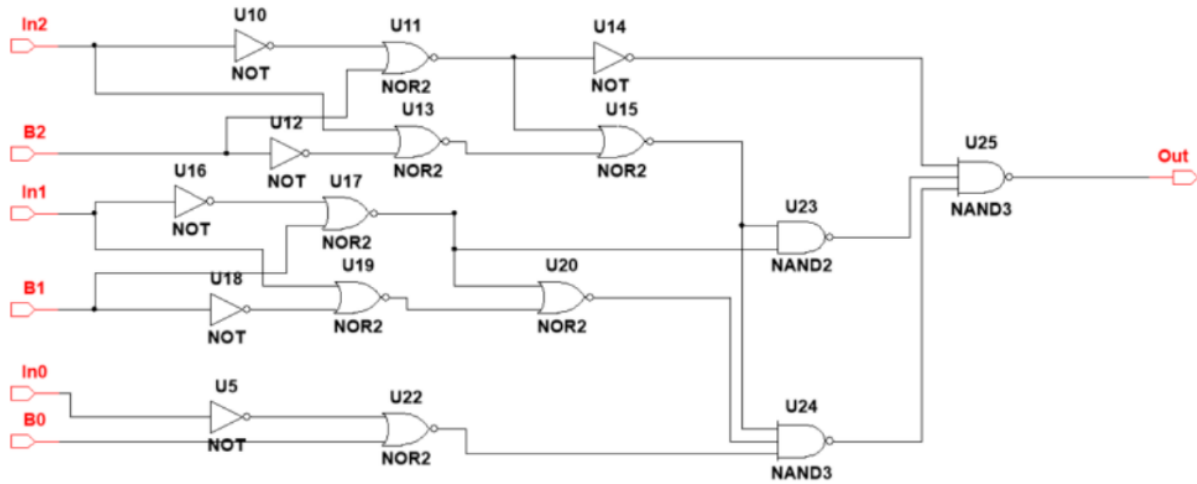


Figure 6. Comparator circuit schematic

This is a 3-bit comparator circuit with a critical path number of 5 stages from In2 to Out.

2.5. Overall topology and the total circuit critical path

We connect all the circuit topology above to get the final 4-bits absolute value Detector. The total circuit schematic as shown in Figure 7 below.

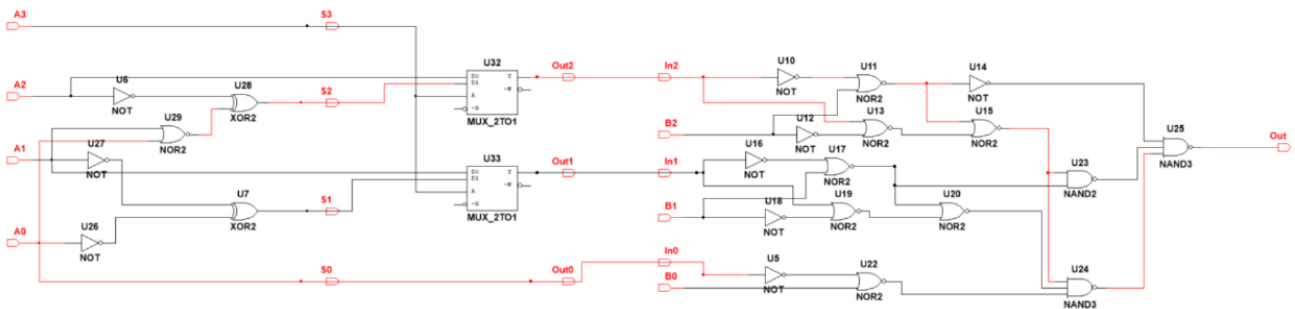


Figure 7. Total circuit schematic

Figure 7 shows the overall schematic design of the 4-bits absolute value comparator with its critical path marketed in red. The total number of stages in the critical path are 8 stages. We will calculate the critical path delay at the next section.

3. Critical path delay calculation

According to the previous section figure 6, there are total of 8 stages of critical path from input port to the output port. Because we assume there is an inverter chain at each input as C_{in} , and we assume that the output load is $32C_{in}$ load. The critical path circuit as shown in Figure 8 below.

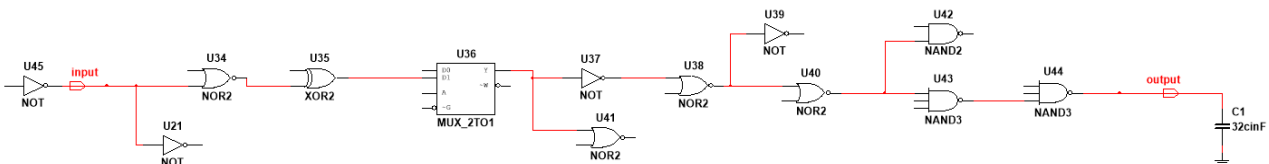


Figure 8. Critical path circuit schematic

Before input port there is a not gate as input load with capacitance as C_{in} , and after the output port there is a 32 times C_{in} capacitance load. Based on this we have our critical path in total of 9 stages with output load of 32 times bigger as input load. Also, there are 3 branches in this critical path needed to be considered during the delay calculation. For all of the NAND and NOR gates we used the

complementary CMOS design, and we assume a unit-sized inverter is W_p equal to 650nm, W_n equal to 430nm, and L_p equal to L_n equal to 100nm.

$$g = \frac{R_{gate} \times C_{in, gate}}{R_{Inv} \times C_{in, Inv}} \quad (4)$$

$$h = \frac{C_{out}}{C_{in, gate}} \quad (5)$$

$$p = \frac{C_{par, gate}}{C_{par, Inv}} \quad (6)$$

According to the formula (4) and (5), we recalculated the logic effort (4) and parasitic effort (6) for all the gates we used in our design [8]. For the XOR gate and the multiplexer (MUX), since we used the transmission gate design for the multiplexer, we need to consider these two gates together during the logic effort and parasitic effort calculation. Basically, these two gates become an inverter follow up with a black box circuit with its own logic effort and parasitic effort values. The detail calculation value as shown in the Table 2 below.

Table 2. Logic effort and parasitic effort for each gate

Gate	Logic effort (g)	Parasitic effort (p)
Not	1	1
Nand2	1.398	2
Nor2	1.602	2
Nand3	1.296	3
Xor2 with MUX	8	5.398

Then the path effort can be calculated based on the formula (7), and we can get all the input capacitance value for each stage to do the sizing.

$$F_{path} = G_{path} \times H_{path} \times B_{path} \quad (7)$$

All the input capacitance value are shown in Table 3.

Table 3. Input capacitance value for each stage

Stages	C_{in}
1	0.9995
2	2.0056
3	4.0789
4	13.2952
5	2.0814
6	6.7827
7	8.4958
8	9.7199
9	17.6362

4. Sizing and Vdd optimization

Generally, based on our calculations above we got the minimum delay for our circuit design, but our energy has not been optimized yet. We decided to sacrifice our delay to decrease the total energy consumption, and the most delay we can get is 1.5 times the minimum delay we have right now [9]. We assume the delay is proportional to the Vdd as shown in formula (8), and the maximum supply voltage is 1.

$$Delay = \frac{k \times v_{dd}}{(v_{dd} - v_T)^2} \quad (8)$$

Then we plug in the value we have we can get the minimum voltage value.

$$Delay_{min} = \frac{k \times 1}{(1-0.2)^2} \quad (9)$$

$$1.5Delay_{min} = \frac{k \times v_{dd,min}}{(v_{dd,min} - v_T)^2} \quad (10)$$

Then by solving the formula (9) and (10), the minimum voltage we can get is 0.7751V. In total we did three different approaches to do the optimization such as Vdd optimization only, sizing only, and sizing and Vdd optimization. Basically, if the voltage is the minimum value which is 0.7751V and keep the sizing as the original minimum delay value. The critical path delay will increase 50% from 79.52 to 119.3 fanout 4 delay, and the energy reduce 39.92%. Also, if we keep voltage as the original and only do the sizing optimization, the delay will also increase 50%, but there will be 74.28% energy reduction. And if we do two optimization both, there will be 78% energy reduction which is the lowest energy we can get. All the detail value are shown in the Table 4 [10].

Table 4. Optimization

Optimization type	Critical path delay	Total energy
minimum delay (1V)	$t_{p_A0_IN \rightarrow OUT} = [79.52] \text{ FO4}_{(1V)}$	E = [386.74] Eu _(1V)
vdd only (0.7751V)	$t_{p_A0_IN \rightarrow OUT} = [119.3] \text{ FO4}_{(1V)}$	E = [232.35] Eu _(1V)
sizing only (1V)	$t_{p_A0_IN \rightarrow OUT} = [119.3] \text{ FO4}_{(1V)}$	E = [99.48] Eu _(1V)
sizing & vdd (0.86V)	$t_{p_A0_IN \rightarrow OUT} = [119.3] \text{ FO4}_{(1V)}$	E = [85.1] Eu _(1V)

5. Conclusion

In summary the section 2 introduces a better half adder circuit design, a more static transmission gate multiplexer (MUX) design, and a comparator design, and all of them are best fits to the 4-bit Absolute-Value Detector. Because of the optimization circuit design introduced in section 2, in section 3 the total critical path delay is smaller than a traditional design. Then, in section 4 we sacrifice the delay a little bit to get a minimum energy consumption. We tried different ways of optimization and eventually got a minimum energy consumption which is 78% reduction. According to all the optimization as mentioned above the 4-bit Absolute-Value Detector has less delay and lower energy consumption.

References

- [1] Dreslinski R G, Wieckowski M, Blaauw D, et al. Near-Threshold Computing: Reclaiming Moore's Law Through Energy Efficient Integrated Circuits [J]. Proceedings of the IEEE, 2010, 98(2): 253-266.
- [2] Yuan Mingzheng. An Absolute-value Detector with Threshold Comparing for Spike Detection in Brain-machine Interface [J]. Journal of Physics: Conference Series, 2021, 2113(1).
- [3] Heranmoy Maity, Arijit Kumar Barik, Arindam Biswas, Anup Kumar Bhattacharjee, Anita Pal. Design of Quantum Cost, Garbage Output and Delay Optimized BCD to Excess-3 and 2's Complement Code Converter [J]. Journal of Circuits, Systems and Computers, 2018, 27(12).
- [4] Holden K N. N-bit carry select adder circuit having only one full adder per bit: US, US4525797 A [P]. 1985.
- [5] Charles S. De Morgan's law and theorems. 2013.
- [6] Ohkubo N, Suzuki M, Shinbo T, et al. A 4.4-ns CMOS 54/spl times/54-b multiplier using pass-transistor multiplexer [C] // Proceedings of IEEE Custom Integrated Circuits Conference - CICC'94. IEEE, 2002.
- [7] Yoshioka M. Comparator circuit and analog digital converter having the same: US, US20100039305 A1 [P]. 2011.
- [8] Uma R, Dhavachelvan P. Performance evaluation of full adders in ASIC using logical effort calculation [C] // 2013 Third International Conference on Recent Trends in Information Technology (ICRTIT). IEEE, 2014.

- [9] Dao H Q, Zeydel B R, Oklobdzija V G. Energy optimization of pipelined digital systems using circuit sizing and supply scaling [J]. IEEE Transactions on Very Large-Scale Integration Systems, 2006, 14(2): 122-134.
- [10] Aizik Y, Kolodny A. Exploration of energy-delay tradeoffs in digital circuit design [C] // Electrical and Electronics Engineers in Israel, 2008. IEEEI 2008. IEEE 25th Convention of. IEEE, 2009.