Final Report of the New Design of a 4-bit Absolute-Value Comparator

Gongshang Xie

Hongxiang Gao

***Abstract*—The goal of this project is to design a 4-bit Absolute-value Detector with the minimum energy and worst-case delay of 1.25ns. A traditional method is modeling the whole program into two parts. The first one is taking the absolute-value and the second step is applying a comparator circuit bit by bit. It is both complicated and time-consuming. In this design, we will take a new kind of method that make the whole procedure more direct, using less components and therefore, more timesaving.**

***Index Terms*—Absolute-value, Comparator, Multiplexer, Mirror Adder.**

I. Introduction

This article is a report of our design——a new kind of 4-bit absolute-value detector with applying mirror adders.

Figure 1 shows the basic diagram of an absolute-value detector that needs to be designed for the project. The inputs (shown in blue) are given. The absolute-value detector (shown in black) is to be designed. The output (shown in red) will give the result that whether x[n] is larger than the Thr or not. If x[n] ¿ Thr, the output goes high, or it gives a low level output.

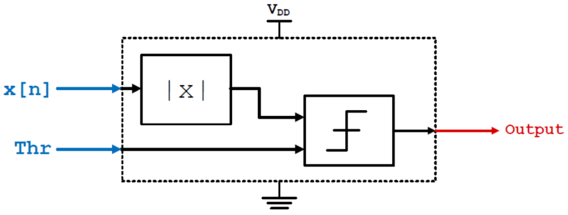


Fig. 1. Traditional Procedure of an Absolute-Value Detector

In traditional method, first thing needs to be done is to find out the absolute value of the input number——$A_1A_2A_3A_4$. The way to do that has two parts. Firstly, one need to decide whether A_1 is 1 or 0. If A_1 is 0, then it is a positive number otherwise, it is negative. Therefore, a multiplexer is needed in this part. The second way is extracting the absolute-value, so there needs at least three inverters and an 1-bit full adder.

The second thing is to construct a 4-bit comparator. Just like what we've learned in Homework7. Figure 2 shows the construction of an 1-bit comparator using complementary pass transistor logic. It is easy to build but it has its problems. It cannot pull the voltage to VDD or GND, but only at some point between them. It may not be a big problem if it only happens at the output point but it does matters in mid points, which may cause errors in consequent parts. If one wants to avoid this flaw, he will need to apply CMOS logic which will cause MOSFETs in an extremely considerable number.

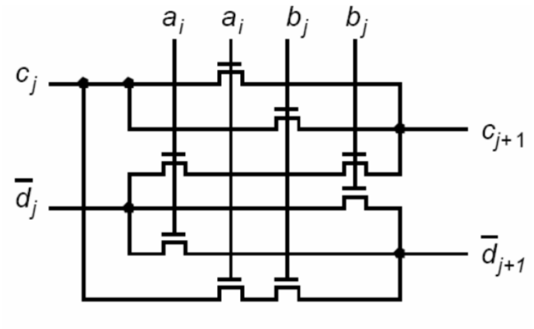


Fig. 2. Construction of an 1-bit Comparator

Since the traditional method has such a lot shortcomings, we design a new method which is both logically direct and component-saving.

II. Concept of Comparator Using Mirror Adder

The traditional method divides the whole procedure into two parts——extracting the absolute value and applying comparison, which is complex and time-consuming, whereas in our design, these two parts are applied in one procedure, the mirror adder.

For example, there is two numbers, a 4-bit number $A_1A_2A_3A_4$ in 2's complementary form and a 3-bit positive number $T_1T_2T_3$. There has two situations, A is a positive number or a negative one, different situation will have different procedure. To decide whether A is positive or negative, we need to determine the value of A_1, just like we need to do in normal procedures.

A. Situation I

First situation is that A is a negative number, which is the simpler case between the two. What we need to do is calculating T - (-A) = T + A = B. If the output B is negative, which means the magnitude of A is larger than T. In this situation, we can take the output as B_1, then B_1 should be equal to 1 if the absolute value of A is larger than T. Therefore, in this case, we just have to calculate A+T, and determine the carryout output A+T. Since the carryout is the key part, the other output of a full adder——sum is not needed, so we just need to apply a mirror adder, not the complete full adder.

While a mirror adder also has two parts, the adder part and an inverter. Figure 3 shows how they work clearly.

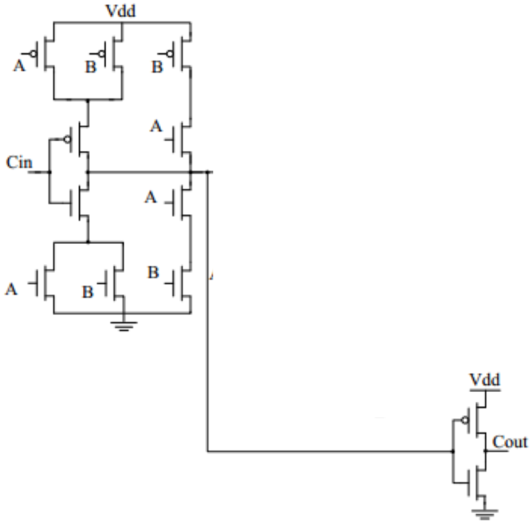


Fig. 3. Two parts of Mirror Adder

The left part we call it adder part while the output of it is $\overline{C_{out}}$. The reason we divide it into two part will be explained in latter part of this article.

B. Situation II

Second situation is that A is a positive number, which is a little bit more complicated but also more tricky and interesting. If A is larger than T, T-A should be a negative number, which means that the first bit of T - A should be 1. Let's say B = T - A. If A>T, then B<0, so B_1=1. If we take the output as B_1, then B_1 should be equal to 1 if the absolute value of A is larger than T.

In this situation, the key point is to calculate T - A, a subtraction. Since we've never learned about how to apply an subtractor, we can apply this into two parts. The first part is take A's negative, and then, add them together.

Let's show it in a formula. B = $T_1T_2T_3$ - $A_2A_3A_4$, while -$A_2A_3A_4$ = $\overline{A_2}\ \overline{A_3}\ \overline{A_4}$ + 1. Then we should add it with T. One thing tricky is that there is two additions. First is the adding 1 when calculating -$A_2A_3A_4$, and the second one is getting the final result. We can combine the two procedure into one. What we need to do is setting C_i of the first mirror adder as 1, which may helps us avoiding a lot of messy things.

C. Combine Them Together

There is two differences between the two situation. First one is in situation2, we need to take the negative for each bit of A while situation1 doesn't, and the second one is in the adder part. In situation2, the value of C_i is 1 while in the other case it should be equal to 0.

III. Schematic Simulation

We constructed 5 circuits in all, from the simplest inverter to to final big picture.

- Inverter

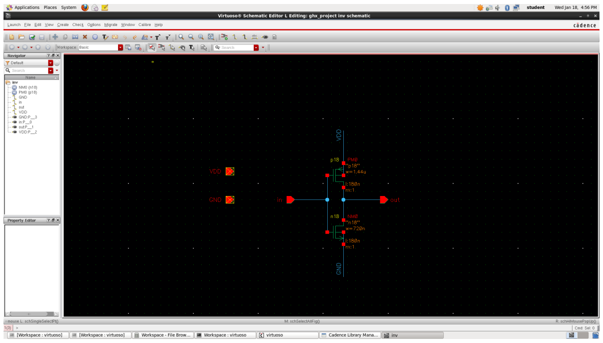


Fig. 4. Schematic of Inverter

Figure 4 is the schematic of our inverter, the size of it is 8 units for upper PMOS and 4 units for lower NMOS.

- Multiplexer

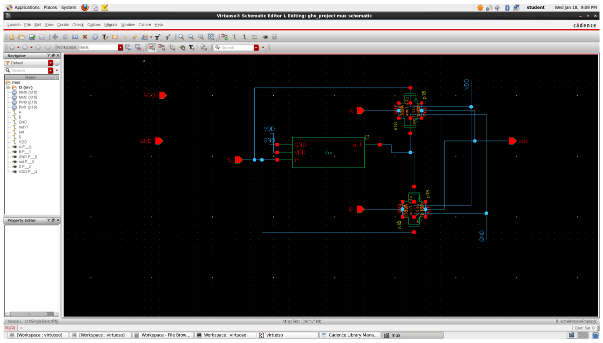


Fig. 5. Schematic of Multiplexer

Figure 5 is the schematic of a multiplexer. The output signal depends on the input of A_1. It works just like we've mentioned, A_1 determines which procedure the circuit executes. Size of the circuit is shown in Figure 6.

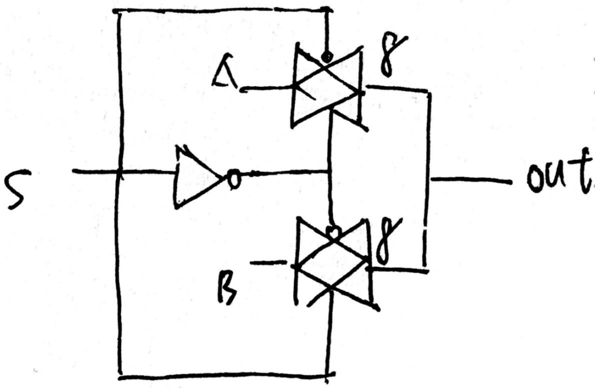


Fig. 6. Sizing the Multiplexer

- Mirror Adder

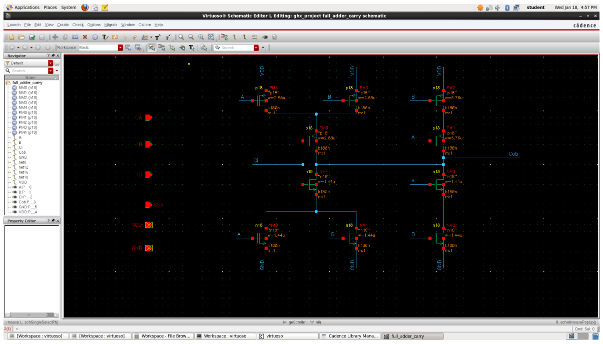


Fig. 7. Schematic of Inverter

Figure 7 is the schematic of a mirror adder, actually, part of it. It is the addition part of a mirror adder. Figure 8 shows how we size it.

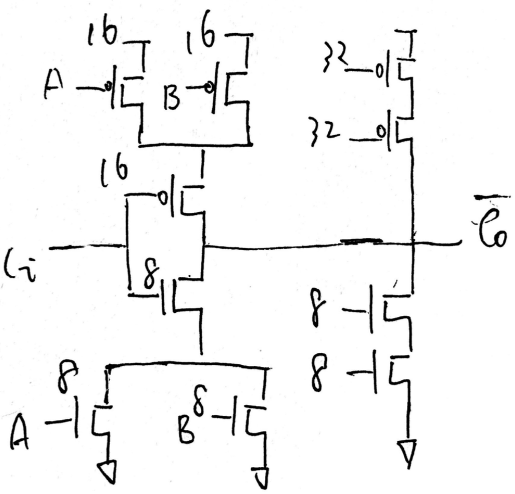


Fig. 8. Sizing the Mirror Adder

- Simulation Circuit

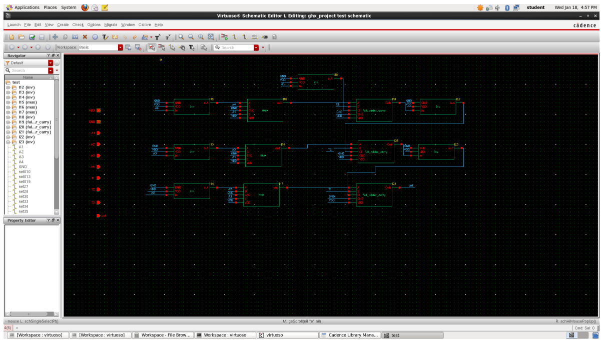


Fig. 9. Schematic of Whole Circuit

Figure 9 is the schematic of the whole circuit and how to connect these components.

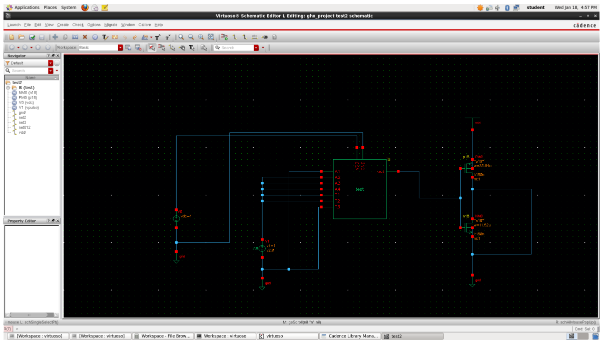


Fig. 10. Test Circuit

Figure 10 is the test circuit. The critical input is A = 0111 and T = 110.

- Simulation Result

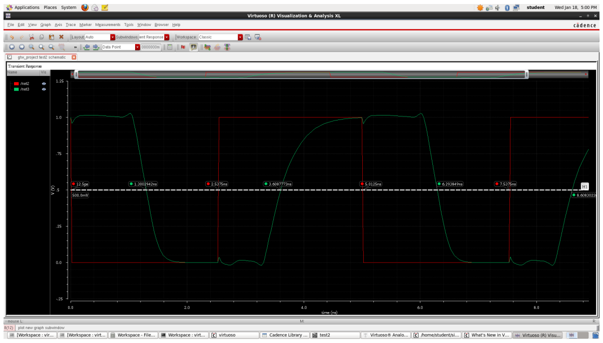


Fig. 11. Test Circuit

Figure 11 is the simulation result. There are two kind of delay in the circuit. Rising delay is equal to 1.0712ns and falling delay is 1.2813ns. The average delay is 1.17ns.

The critical path we chose in the test procedure is shown in Figure12.

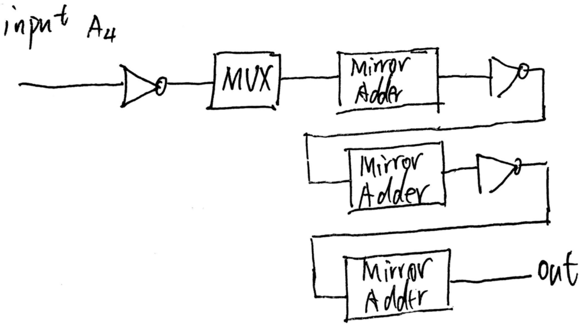


Fig. 12. Test Circuit

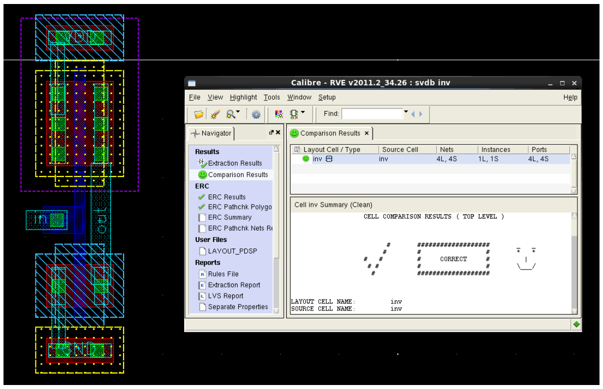
Now we can answer the question mentioned before. The question is why we decide to divide the mirror into two parts, the adder part and the inverter part. The reason is it can help us to size the circuit easier. In traditional mirror adders, they are combined together, so it will be hard to size them. In addition, in the scheme, what we need is the output of $\overline{C_{out}}$. It will be less messy if we divide the mirror adder apart.

The logic effort of the whole circuit is G = 64. Then the average logic effort of each part is 1.8. In real design procedure, we need to size circuits so that the average logic effort between 3 and 4. Actually in schematic simulation part, we did not pay a lot of emphasis on sizing the circuit. But thanks to the tricky design, the delay is quite acceptable.

IV. Layout Simulation

Then we are going to step to the layout part.

- Inverter



Layout Cell / Type	Source Cell	Nets	Instances	Ports
inv	inv	4L, 4S	1L, 1S	4L, 4S

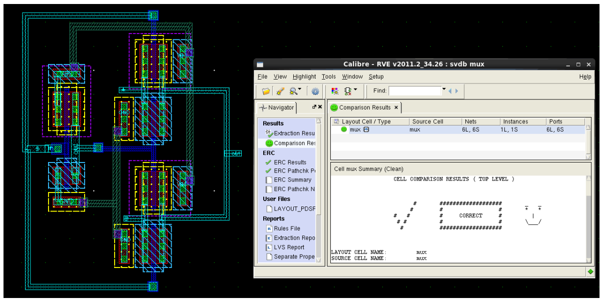
Cell inv Summary (Clean)

```
                CELL COMPARISON RESULTS ( TOP LEVEL )


                   #       ###################
                  #        #                 #      _   _
             #   #         #     CORRECT     #      *   *
              # #          #                 #        |
               #           ###################      \___/


LAYOUT CELL NAME:         inv
SOURCE CELL NAME:         inv
```

Fig. 13. Layout of Inverter

Figure 13 shows the layout of the inverter. The righter block is the proof of our design is adopted both drc and lvs test.

- Multiplexer



Layout Cell / Type	Source Cell	Nets	Instances	Ports
mux	mux	6L, 6S	1L, 1S	6L, 6S

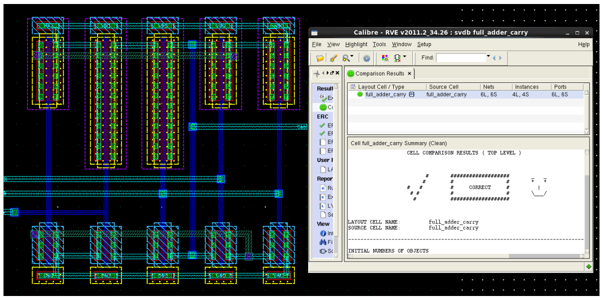
Cell mux Summary (Clean)

```
                CELL COMPARISON RESULTS ( TOP LEVEL )


                   #       ###################
                  #        #                 #      _   _
             #   #         #     CORRECT     #      *   *
              # #          #                 #        |
               #           ###################      \___/


LAYOUT CELL NAME:         mux
SOURCE CELL NAME:         mux
```

Fig. 14. Layout of Multiplexer

Figure 14 shows the layout of the multiplexer. The righter block is the proof of our design is adopted both drc and lvs test.

- Mirror Adder


Calibre - RVE v2011.2_34.26 : svdb full_adder_carry
File View Highlight Tools Window Setup Help
Find:
Result
Ex
Co
ERC
User F
Repor
View
Comparison Results

Layout Cell / Type	Source Cell	Nets	Instances	Ports
full_adder_carry	full_adder_carry	6L, 6S	4L, 4S	6L, 6S

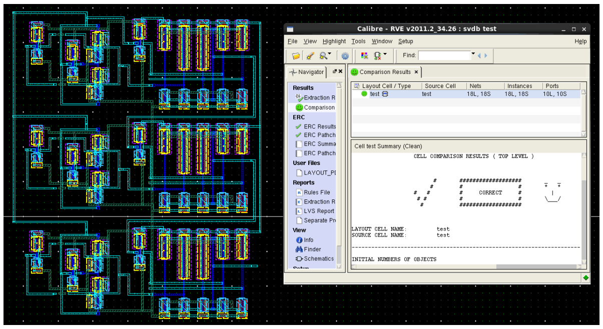
Cell full_adder_carry Summary (Clean)

```
                CELL COMPARISON RESULTS ( TOP LEVEL )


                   #       ###################
                  #        #                 #      _   _
             #   #         #     CORRECT     #      *   *
              # #          #                 #        |
               #           ###################      \___/


LAYOUT CELL NAME:         full_adder_carry
SOURCE CELL NAME:         full_adder_carry

------------------------------------------------------------------------

INITIAL NUMBERS OF OBJECTS
```

Fig. 15. Layout of Adder

Figure 15 shows the layout of the mirror adder. The righter block is the proof of our design is adopted both drc and lvs test.

- Simulation Circuit



Layout Cell / Type	Source Cell	Nets	Instances	Ports
test	test	18L, 18S	18L, 18S	10L, 10S

Cell test Summary (Clean)

```
                CELL COMPARISON RESULTS ( TOP LEVEL )


                   #       ###################
                  #        #                 #      _   _
             #   #         #     CORRECT     #      *   *
              # #          #                 #        |
               #           ###################      \___/


LAYOUT CELL NAME:         test
SOURCE CELL NAME:         test

------------------------------------------------------------------------

INITIAL NUMBERS OF OBJECTS
```

Fig. 16. Layout of Big Picture

Figure 16 shows the layout of the whole circuit. The righter block is the proof of our design is adopted both drc and lvs test.

- Simulation Result

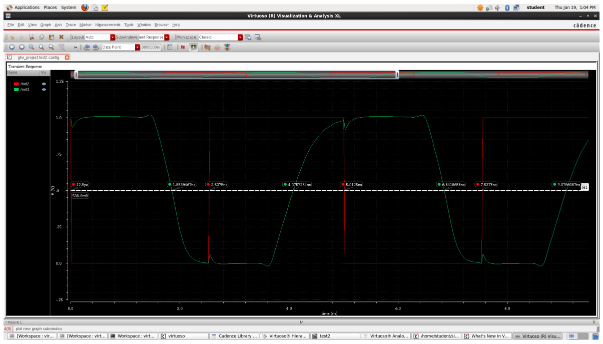


Fig. 17. Test of Simulation

With VDD as 1V, the simulation result is shown in Figure 17 the rising delay is 1.5382ns and the falling is 1.8411ns. To achieve the time requirement, one method is to boost the high level voltage VDD.

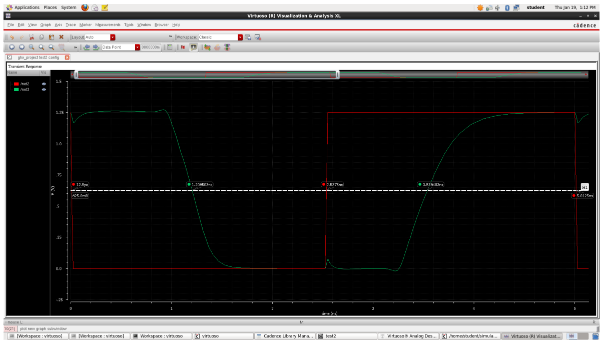


Fig. 18. Test of Simulation with Voltage Boost

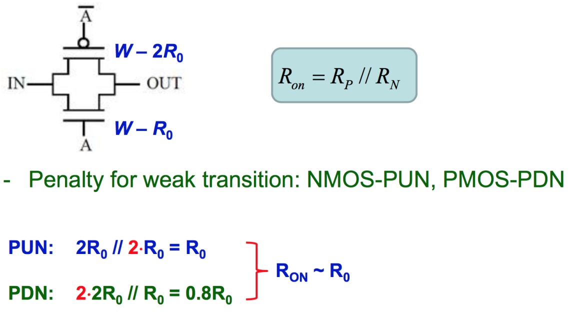
Figure 18 is the result with a voltage boost. In this case, we set the high level voltage at 1.25V, then it leads to the delay time at 0.9989ns and 1.194ns, which achieve the requirement of 1.25ns.

V. Conclusion

There are two points needs to be figured out. First one is about the design, the next big thing is about the delay.

For our design, we have confidence that this design is one of the best. The whole circuit needs only about 56 MOSFETs, which is far less than that in traditional case. As a common knowledge, less components will lead to less time delay.

Then let's talk about the delay. From Figure 11, there is a result that the rising delay and the falling delay is difference and the difference is about 0.2ns. Considering the whole delay is in 1ns level, this difference is quite large. The reason we guess is the effective pull strength is different for pulling up and pulling down. The key difference we think it is in the multiplexer part. We set the size of the transmission gate as 8, and the size of PMOS and NMOS are the same, which they should be equal. In class analysis, we take the pulling strength as equal, as shown below. It is a big approximation, and the larger size of the transmission gate, the bigger difference of the pulling strength will be.



$R_{on} = R_P \,//\, R_N$

- Penalty for weak transition: NMOS-PUN, PMOS-PDN

PUN: $2R_0 \,//\, 2 \cdot R_0 = R_0$

PDN: $2 \cdot 2R_0 \,//\, R_0 = 0.8R_0$

$R_{ON} \sim R_0$

Fig. 19. TG Analysis in Class

Just as shown in Figure 11 and Figure 17, the difference between two result is quite large. We can only enhance the high level voltage VDD. We found that if we boost the input voltage from 1V to 1.25V, the time-consumption is reduced about 50 percents, so it is also a proof that the time consumption is proportional with the square of of input voltage level.

VI. Flaws and Prospects

There is a big difference of the result between schematic and layout. We think it could be attribute to two points. First one is in layout analysis, it takes parasitic parameters into account while in schematic's case, all of the components are lumped elements. Second reason is the wires. In real circuit analysis, resistance and capacitance could not be ignored. Since the circuits works at high frequency, 800MHz is relatively a big number, phase on lines could be different, the electric length of the wires is also a big problem. We think in practical analysis, one can apply transmission line theorem to analyze the circuit. In addition, the components will radiate EM wave when it's working, which is also a variable.

Second thing need to be emphasized is the prospect of this design. As we mentioned before, we didn't pay a lot attention to sizing the circuit, which cause the average logic effort of each part is about 1.8, much lower than 3. The purpose we extract the inverter from mirror adder is to size it but we didn't execute it practically. We took all the inverters at a same size, so the first thing to do is sizing these inverters, and then, we can step to further procedures.