目录

abstract：Cadence is a leading provider of EDA and semiconductor IP. In this project, I try to use the Cadence to realize a **4-bit “Absolute-value Detector”**, which is combined by 2 modules: absolute module and comparator module.

Keywords

Cadence, **4-bit “Absolute-value Detector”**, schematic, layout

Introduction:

第一章：项目简介

1.1功能介绍

A 4-bit ”Absolute-value Detector” is used to compare the Absolute-value of a signed number with a threshold value. If the absolute-value is greater than the threshold value, then output one. Otherwise, it will output zero.

So, to realize the function, we should complete two parts: absolute module and comparator module.

1.2实现工具介绍

Cadence is a leading provider of EDA and semiconductor IP. Its custom/analog tools help engineers design the transistors, standard cells, and IP blocks that make up SoCs. Its digital tools automate the design and verification of giga-scale, giga-hertz SoCs at the latest semiconductor processing nodes. Its IC packaging and PCB tools permit the design of complete boards and subsystems. In short, Cadence® technology helps customers build great products that connect the world.

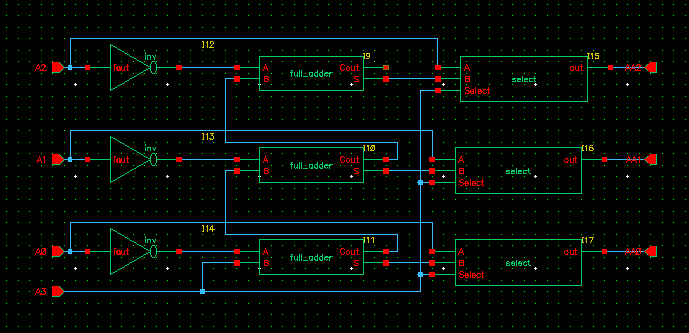
第二章：模块设计

Overview the project, we can divide it into three parts: absolute module, comparator module and other small modules. To make the circuit clearer, we modularize it into small parts and then use the blocks to expand the circuit from small to big. After that, we also do the layout to do the concrete realization.

2.1 absolute module

When doing the absolute module, we should consider two cases, whether the input is positive or negative. If the highest bit is one, it means negative. To get the absolute value, we 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. And the absolute value is the lower 3-bit. So we should use a mux to get the answer we need.

原理图：



abs

2.2 comparator module

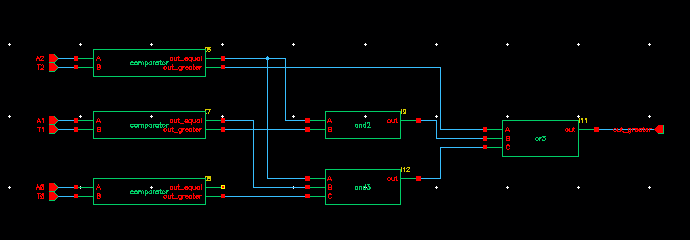
After getting the absolute value, now it’s time to compare it with the threshold value. To do this, we need to compare each bit one by one.

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

A = A2A1A0 T = T2T1T0

F(A > T) = (A2 > T2) + (A2 = T2)(A1 > T1) + (A2 = T2)(A1 = T1)(A0 > T0)

So, to make it come true, we can get the circuit below:

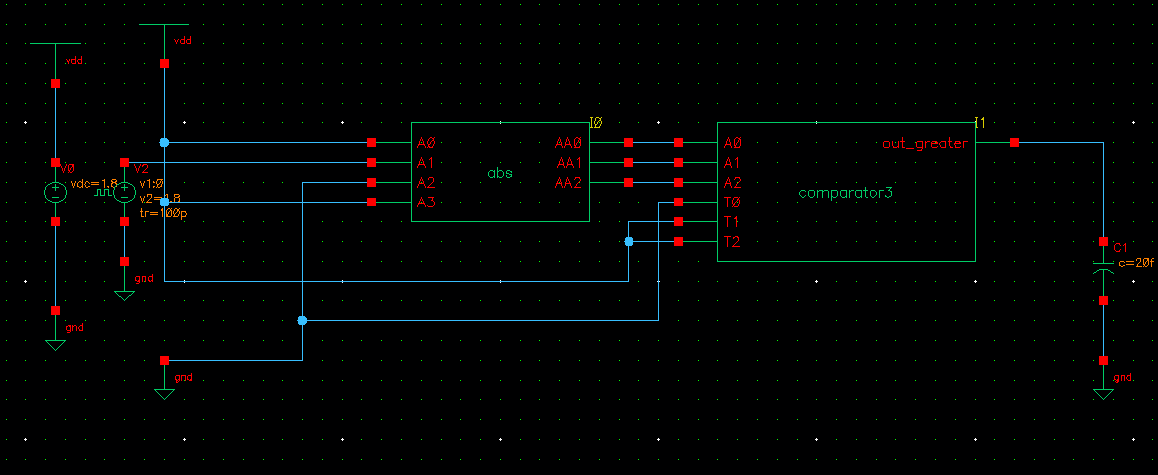


3-bit comparator

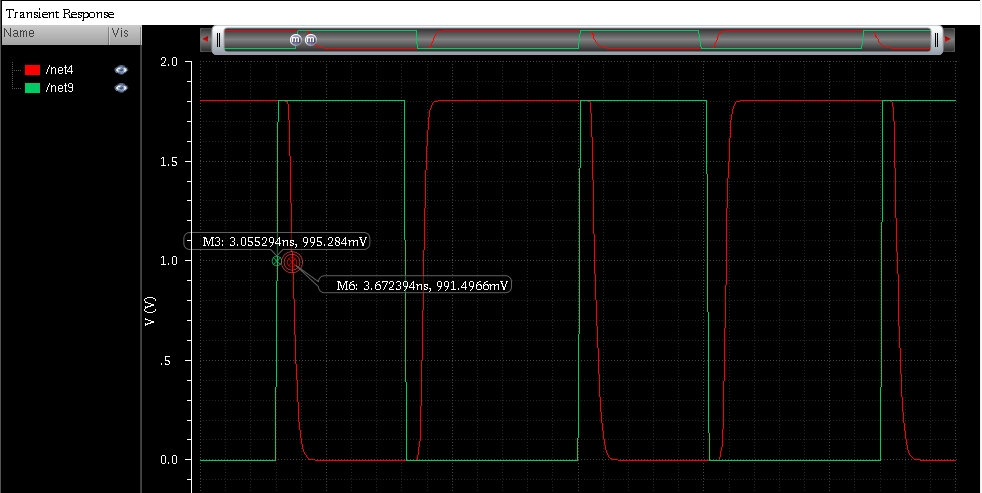
2.3 other small modules

To make the circuit more efficient, we choose to use the transmission gate (TG) to build the circuit. So we should build each gate by TG instead of using the symbol in the library.

第三章：总体电路实现



Result:



Perspective

The perspective of our project is good. The absolute value detector contains many small modules such as adder, mux, comparator and so on. These are all core of the any other circuits. Although, this project is not that hard, it is the region of others.

Discussion

When doing the project, we met a big problem: TG. At first, we use only one TG as the output in one sub-circuit. However, it will lead to wrong output. When TG is closed, the output will be still high. After researching for a long time, we didn’t choose to add pulldown directly in the TG. Instead, we choose to use TG in pairs. And then, the pulldown problem is solved perfectly.

Conclusion

After finishing the project, I clearly know how difficult it is to realize such simple function by circuit, which just need only 3 lines if using C++.

To optimize the result, we use TG instead of CMOS. And it needs much less MOSFETs than using CMOS, which will reduce the delay. So we need to spend more time on this project. We have to refer to many data in the website and do many attempts. It is therefore that we can learn more knowledge than those who use CMOS.It must be worth the effort.

reference

<http://blog.csdn.net/a8039974/article/details/51685520> 数值比较器

<https://en.wikipedia.org/wiki/Transmission_gate> WIKIPEDIA for TG