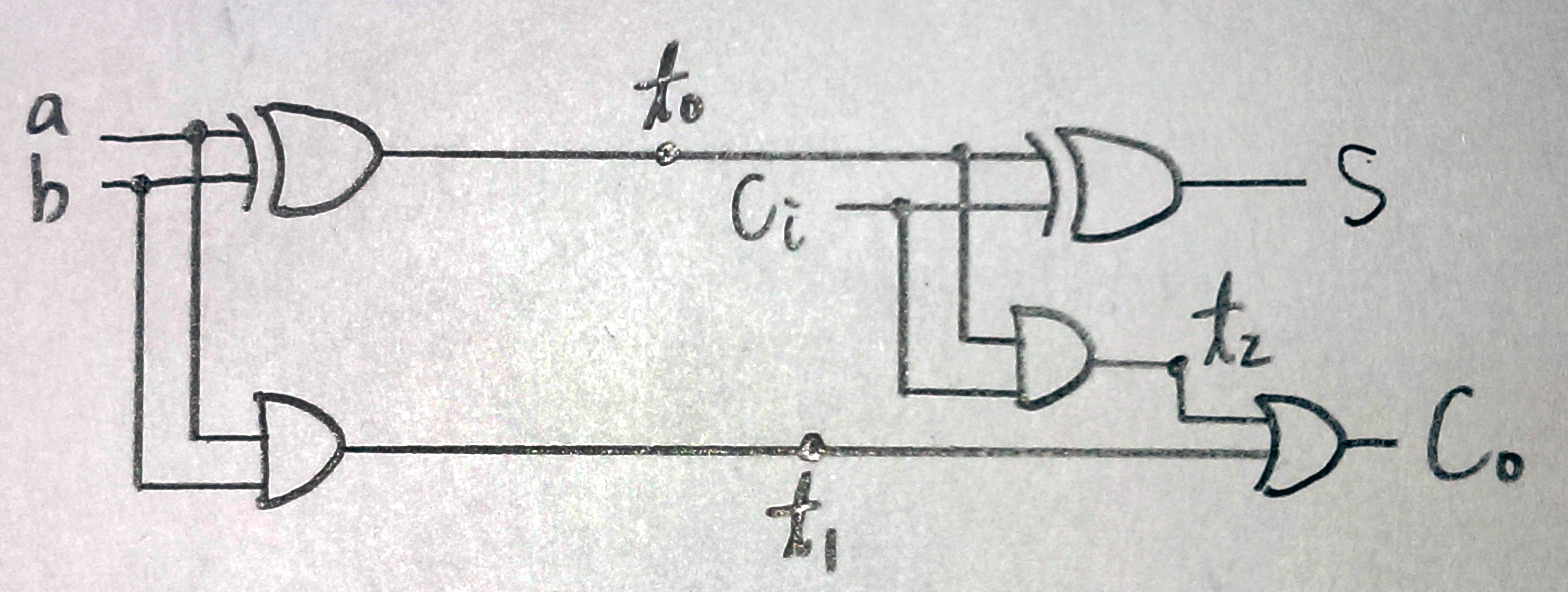
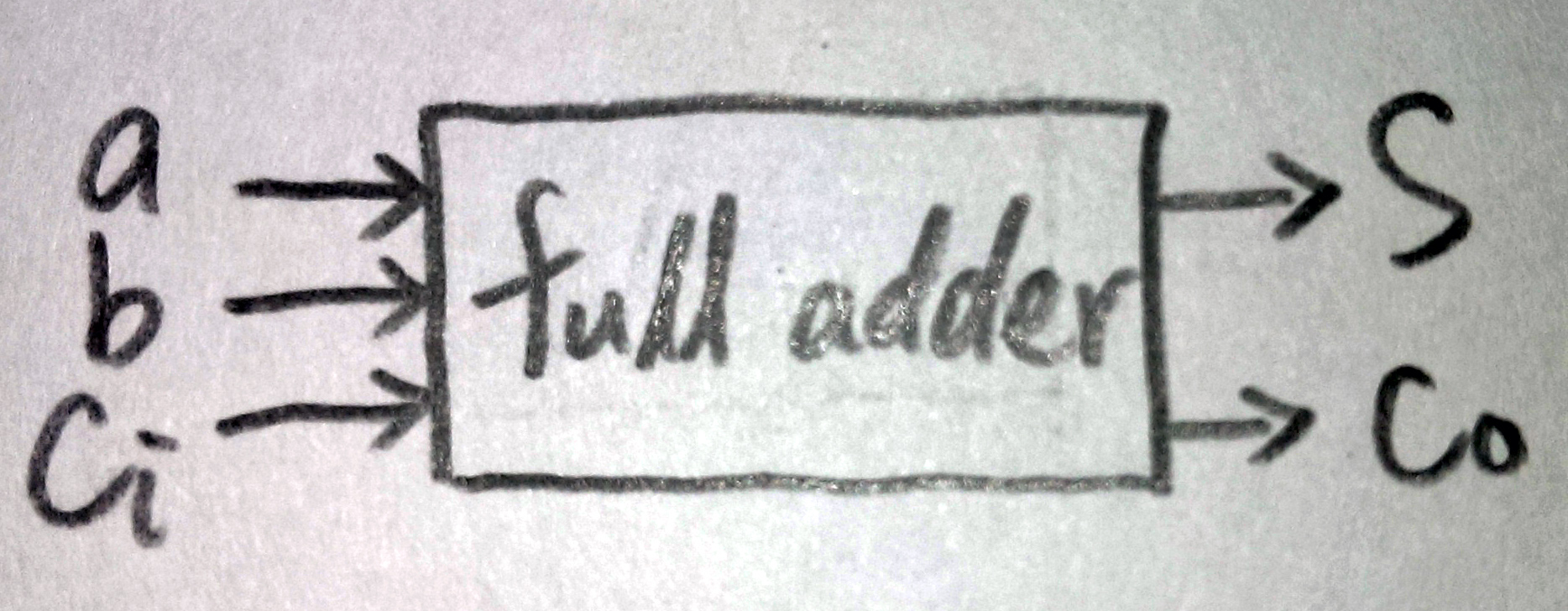
**Lab 1: Introduction to Verilog HDL**

**103061207 徐安廷**

**Design Specification**

1. **Full Adder**

* Logic Equation: **S=a⊕b⊕Ci , Co=a’b+aci+bci**
* Logic Diagram:
* Block Diagram:
* I/O:

Half Adder:

Inputs: a, b.

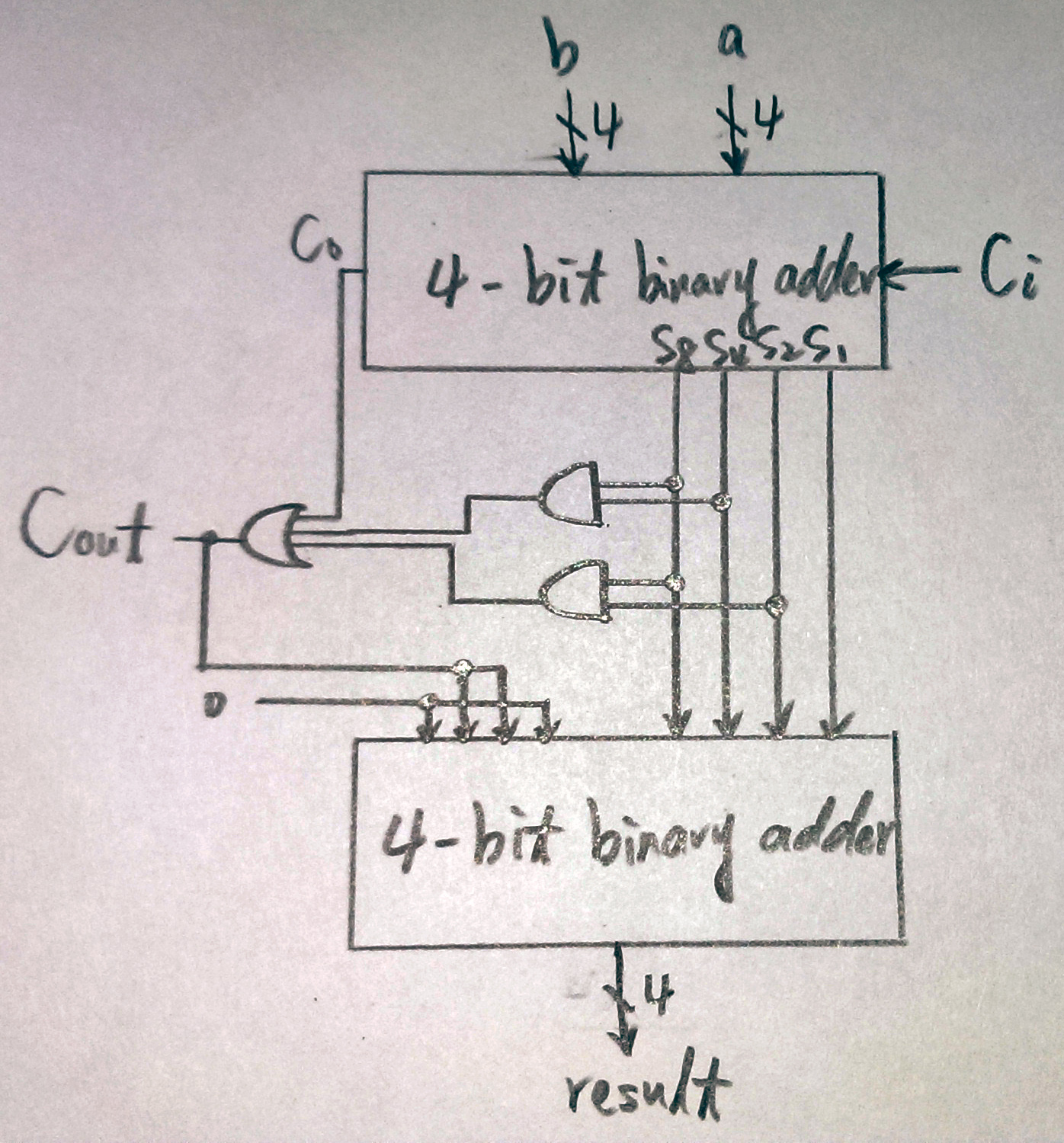
Outputs: s, co.

Full Adder:

Inputs: a, b, ci.

Outputs: co, s.

1. **Single Digit Decimal Adder**

* Block Diagram:
* I/O:

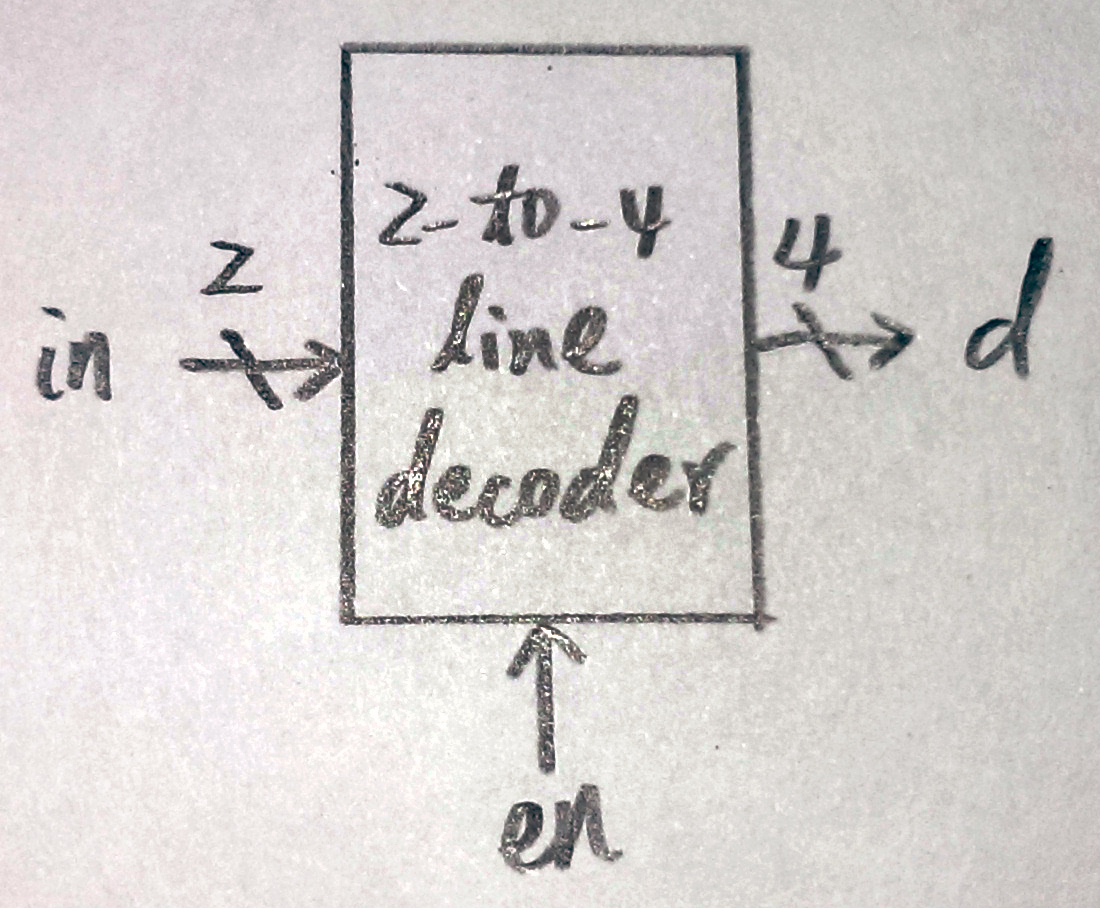
Inputs: a[3:0], b[3:0], ci.

Outputs: result[3:0], s[3:0], co, cout.

1. **2-to-4-Line Decoder**

* Logic Equation: **d[0]=E．in[1]’．in[0]’ , d[1]=E．in[1]’．in[0] ,**

**d[2]=E．in[1]．in[0]’ , d[3]=E．in[1]．in[0].**

* Block Diagram:
* I/O:

Inputs: E, in[1:0].

Output: d[3:0].

**Design Implementation**

1. **Full Adder**

* a and b is the bit we like to add, and ci is an one bit carry in.

s is the result, and co is an one bit carry out.

* The logic function should be: s=a^b^Ci , co=a’b|ac|bc

1. **Single Digit Decimal Adder**

* Logic function:

When adding input a and b, if the result is larger than 9, it should add 6 to the result. So, we will get a one bit carry out and a four-bit binary number result[3:0].

Cout=K+Z8Z4+Z8Z2

1. **2-to-4-Line Decoder**

* Logic function

E is enable. When E=0, the decoder will output 0000.

D[0] = E & (~in[1]) & (~in[0])

D[1] = E & (~in[1]) & (in[0])

D[2] = E & (in[1]) & (~in[0])

D[3] = E & (in[1]) & (in[0])

**Discussion**

* It was really a happy experiment, only when debugging made me disappointed. The only wrong thing a met is I forgot to link the full adders in the second 4-bit adder. So, it is not weird that the simulation turned out lots of red Xs. The experiment results went all correct, since I follow the references in the hangout, so it ran as I expected.

**Conclusion**

* I have got familiar with the use of Verilog, and know how to declare I/Os variables and write a testbench to test my circuit. During the debugging, I tried to understand the error messages. Hope it will help me to finish future projects.

**References**

* Teaching handouts <Introduction> p.52:

It helps making a decimal adder without using any if statements, but by only using the simple logic gates to get the goal.