

# Week 9

CS110P.07

Suting Chen

# Logisim

Run the program

```
> java -jar logisim-evolution.jar -noupdates
```

Note: Don't close this terminal!

Suting Chen

# Logisim

## Basic shortcuts

- Press  $\uparrow/\downarrow/\leftarrow/\rightarrow$  keys to rotate.
- Ctrl / ⌘ + Z      Undo
- Ctrl / ⌘ + R      Redo
- Ctrl / ⌘ + S      Save      You should save **VERY** frequently.

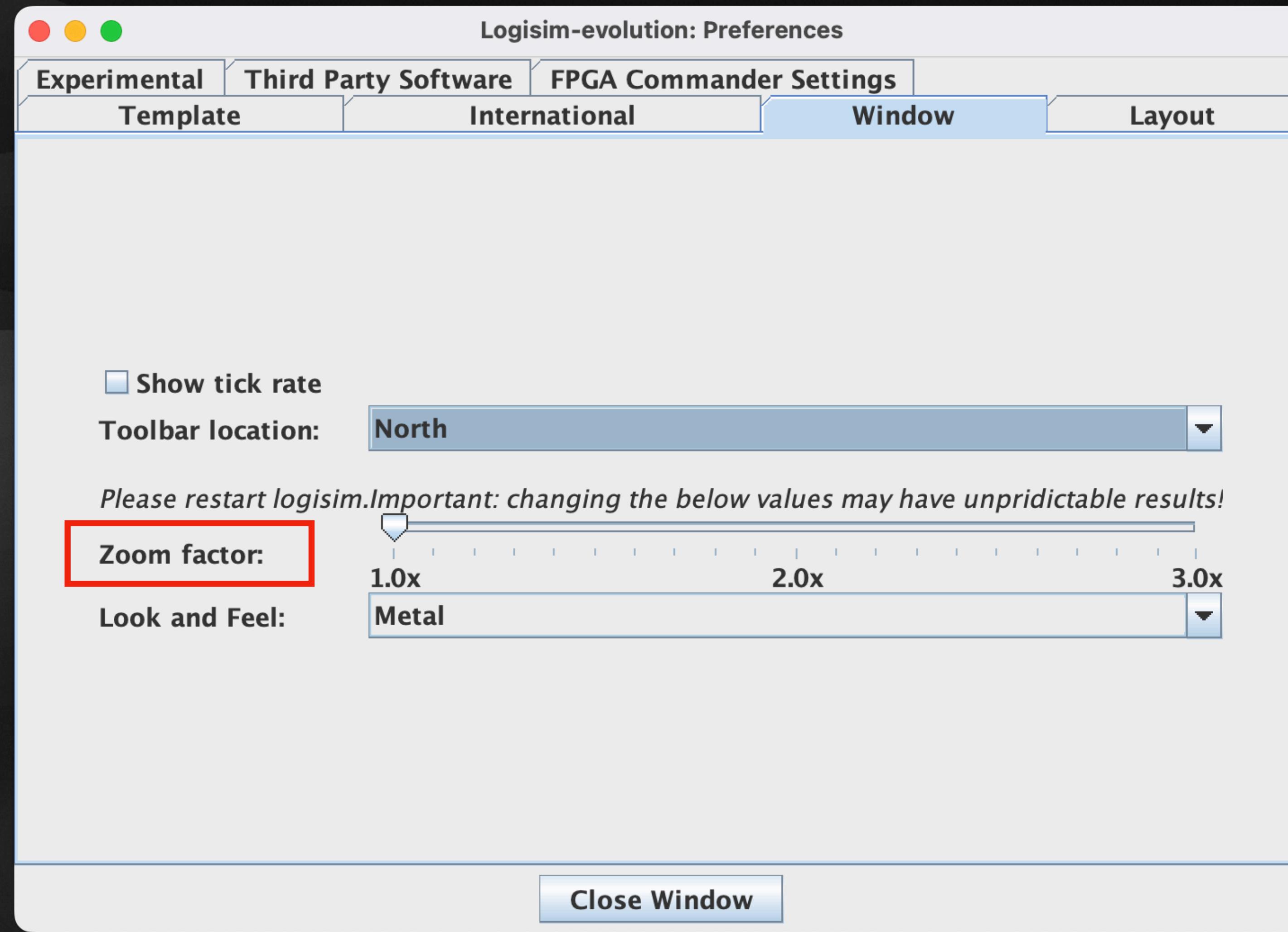
# Logisim

## Basic shortcuts

- Ctrl / ⌘ + A      Select all
- Ctrl / ⌘ + C      Copy
- Ctrl / ⌘ + V      Paste
- Ctrl / ⌘ + X      Cut
- ⌫ / ⌰      Delete

# Logisim

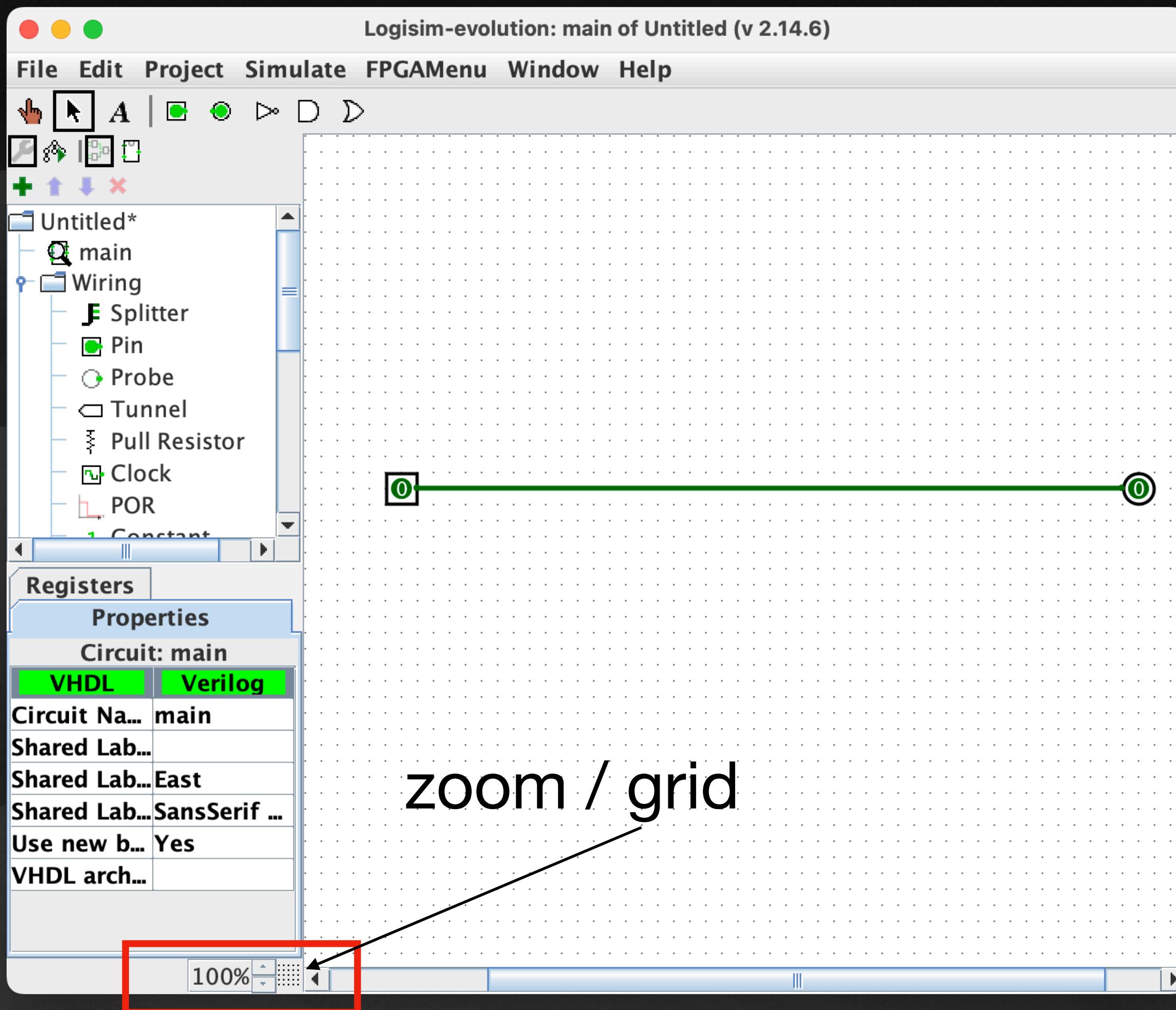
## Appearance



Suting Chen

# Logisim

## Appearance

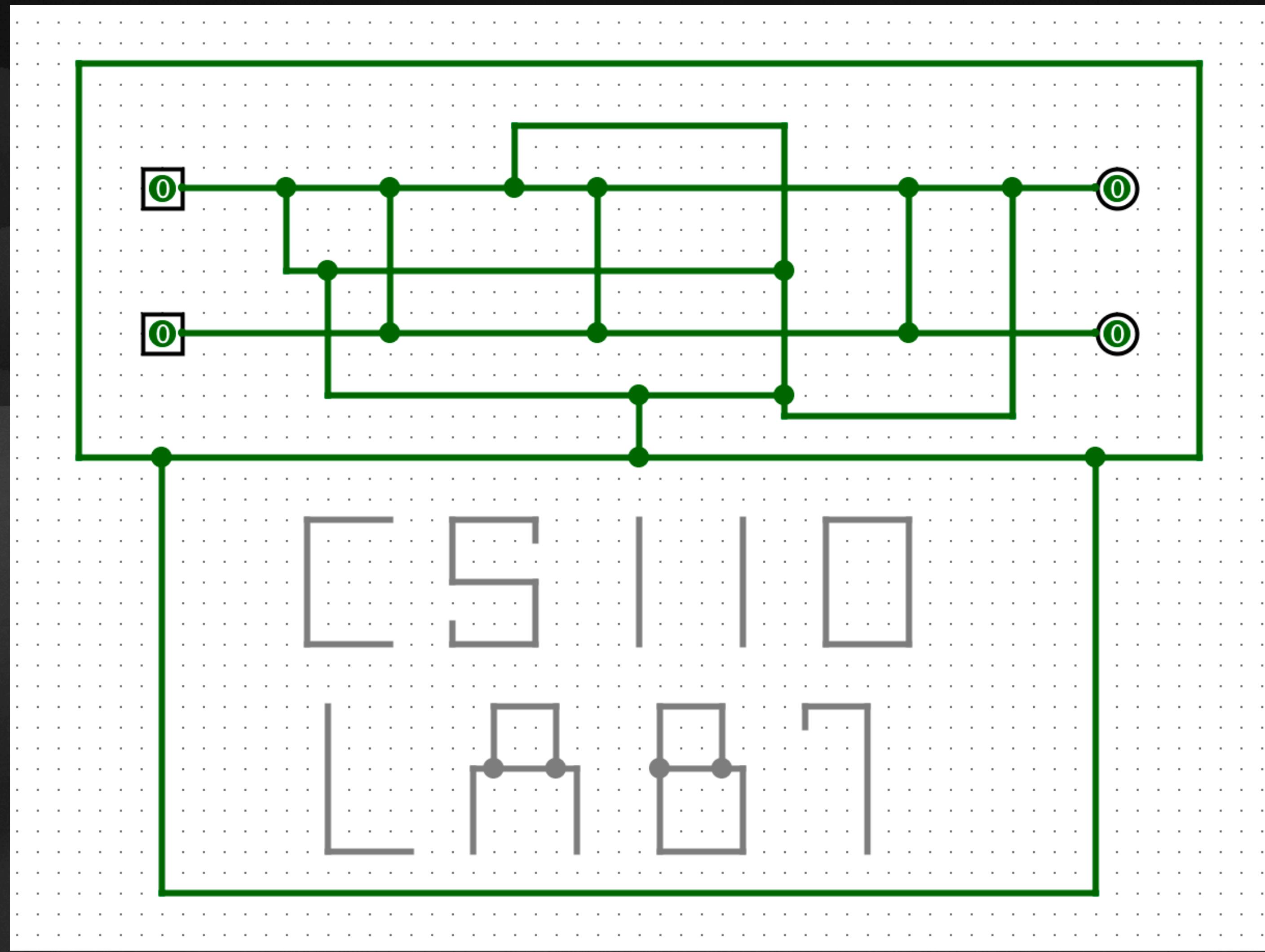


Suting Chen

# Logisim

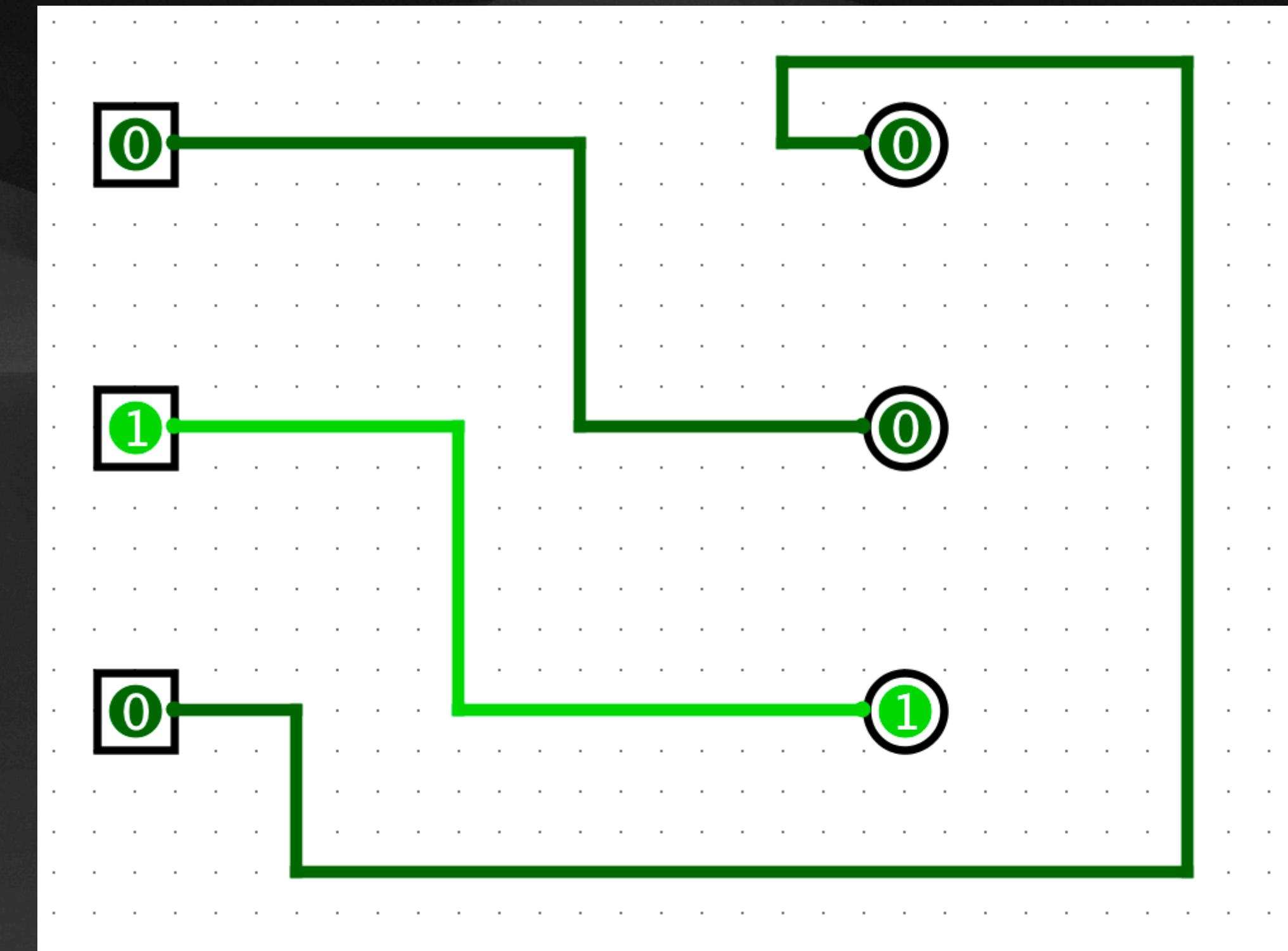
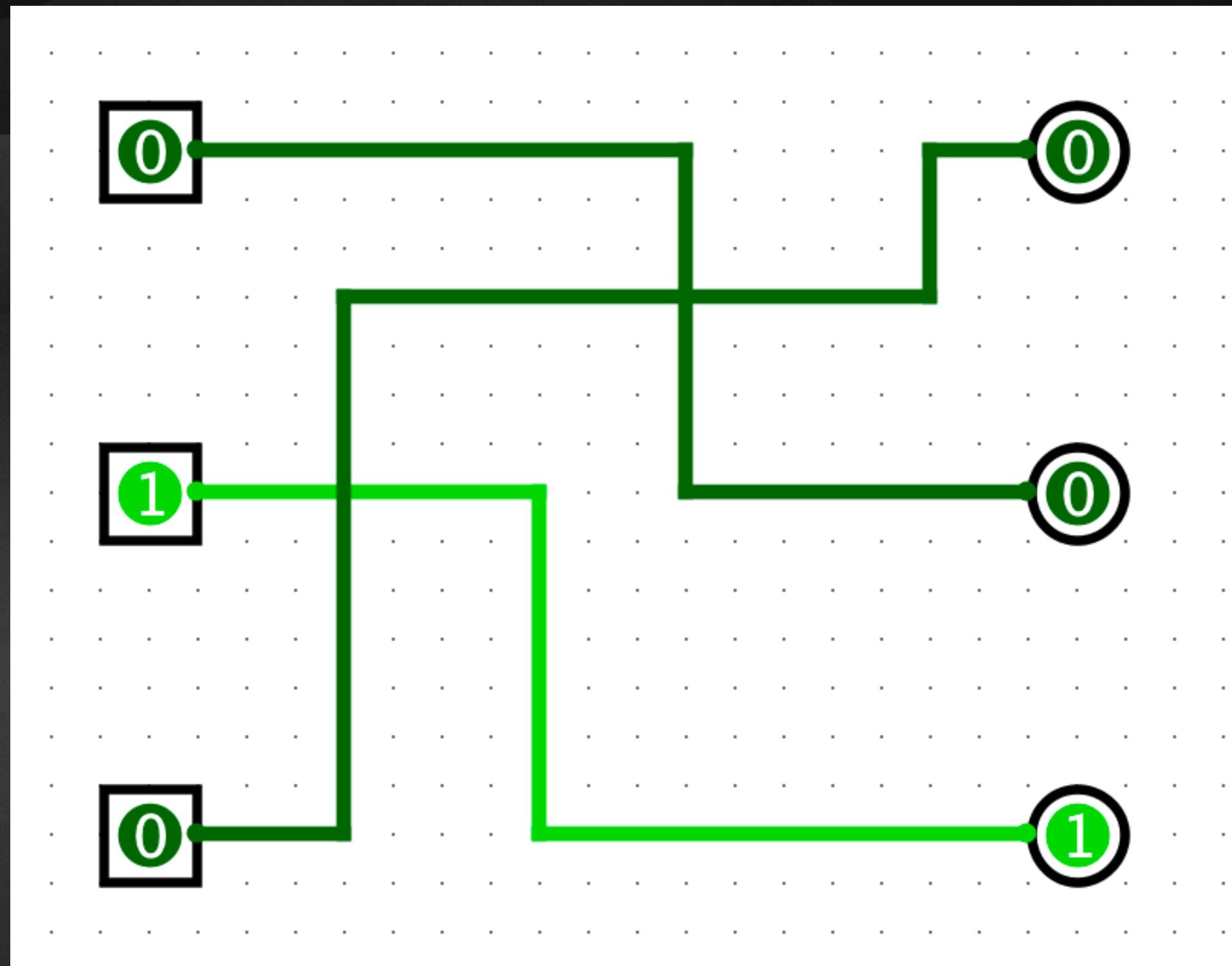
## Wiring

Suting Chen



# Logisim

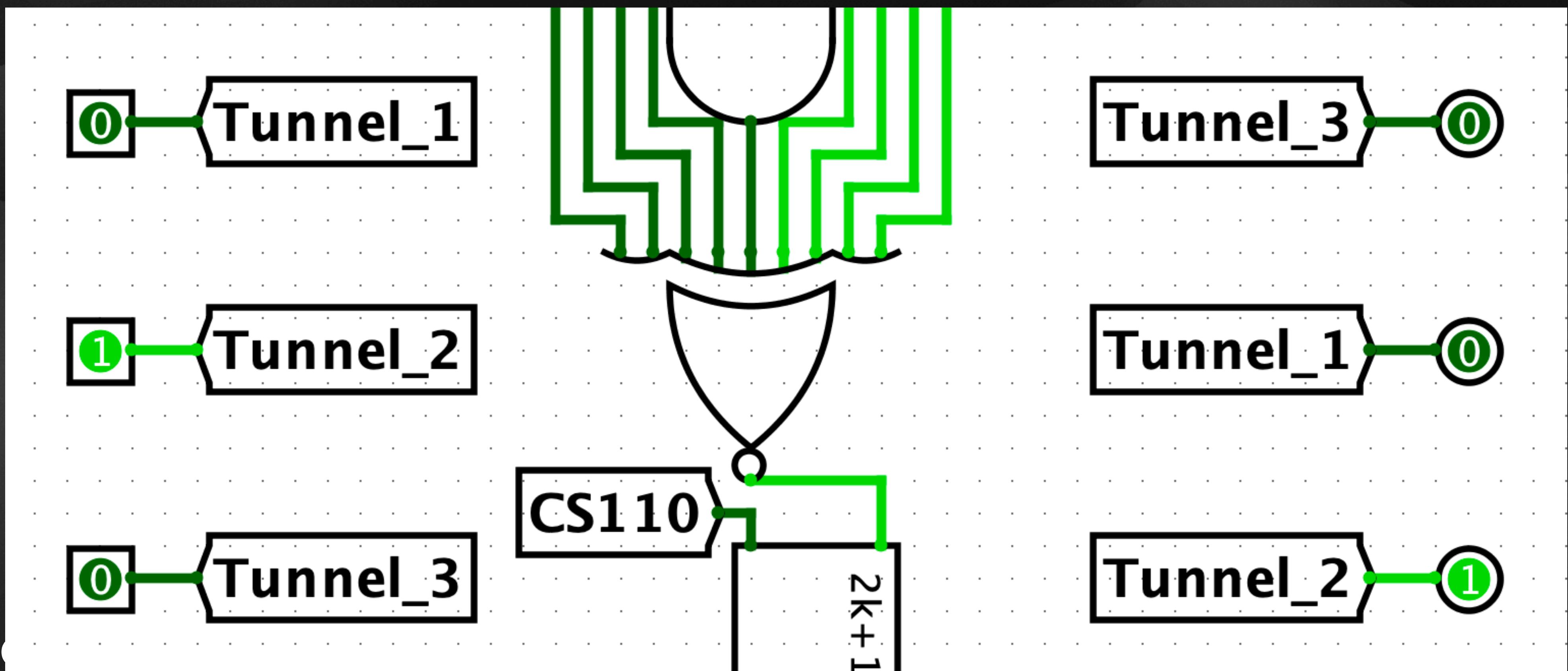
## Wiring



Suting Chen

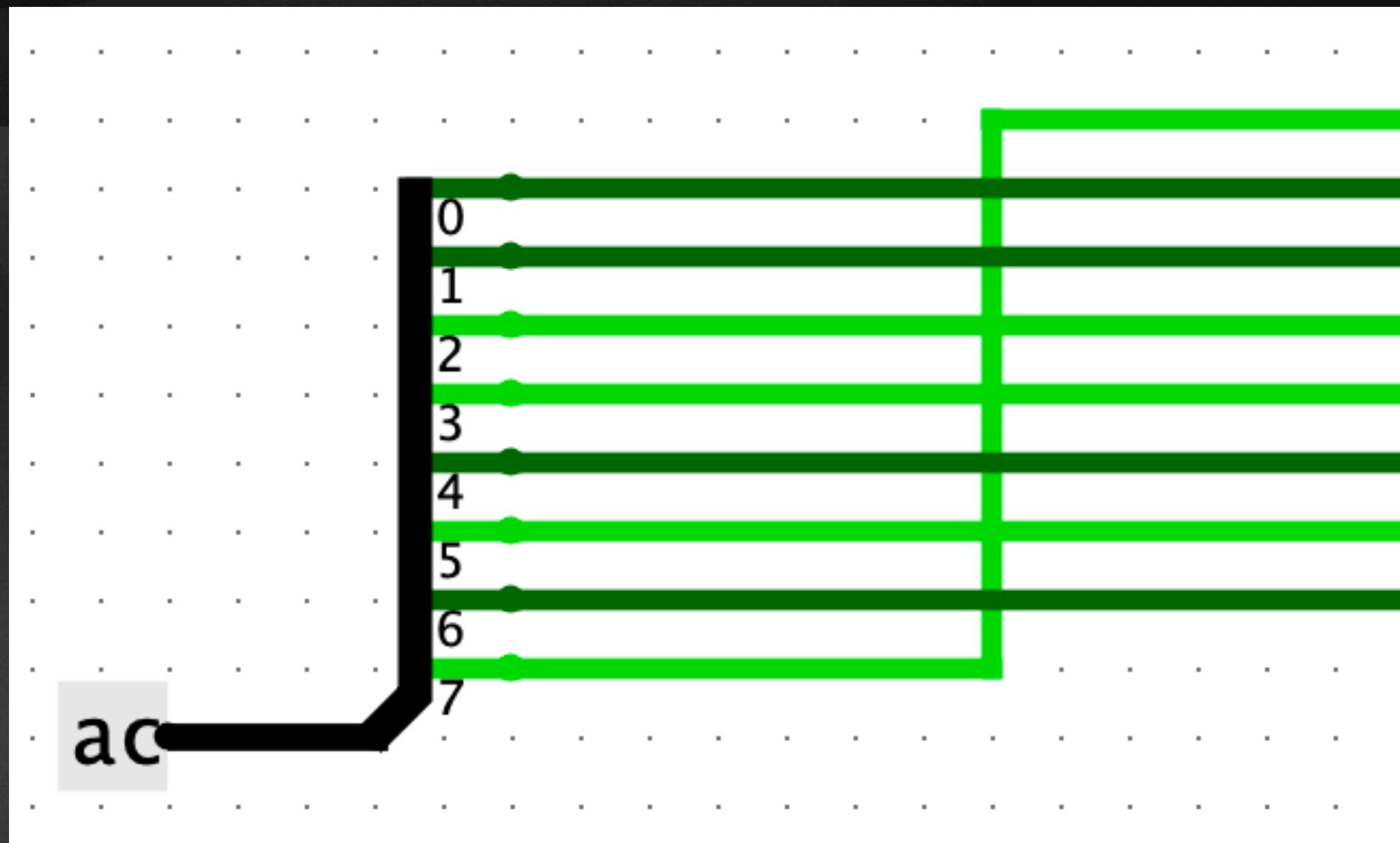
# Logisim

## Wiring

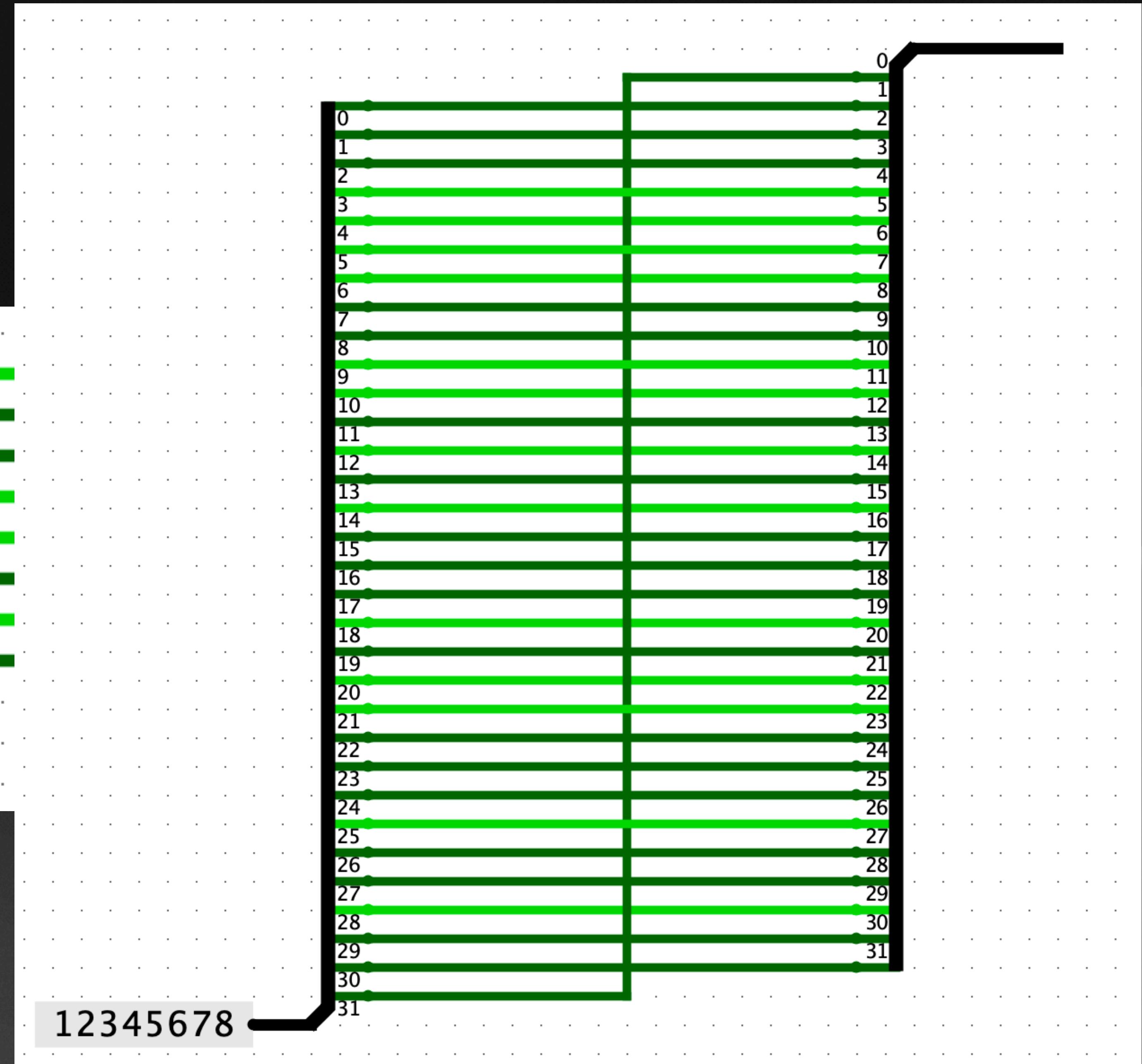


# Logisim

Use “LEGO pieces” wisely

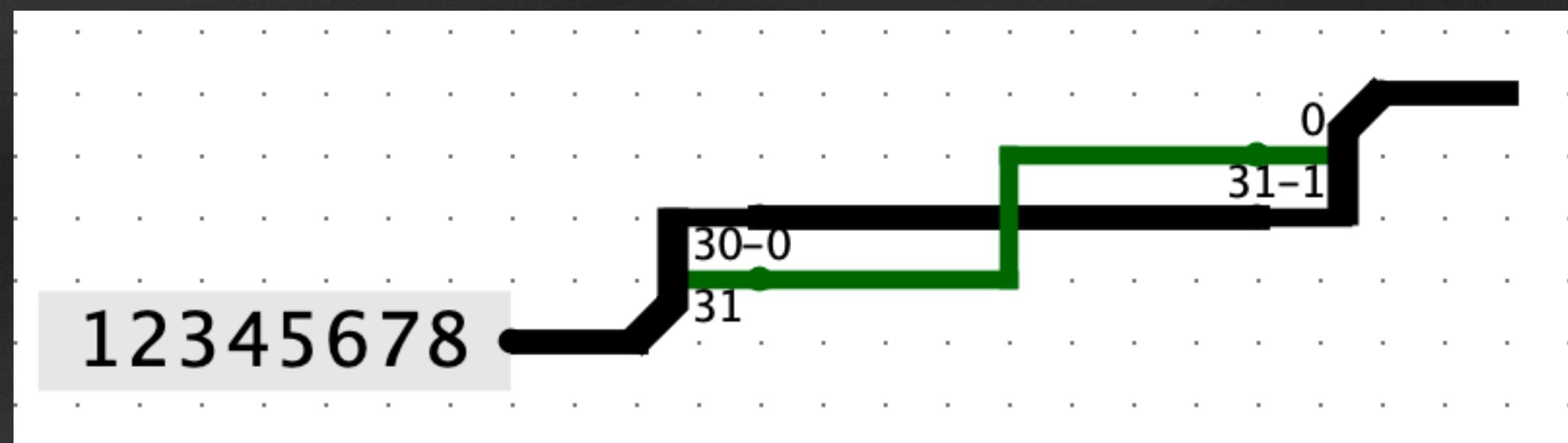


Suting Chen



# Logisim

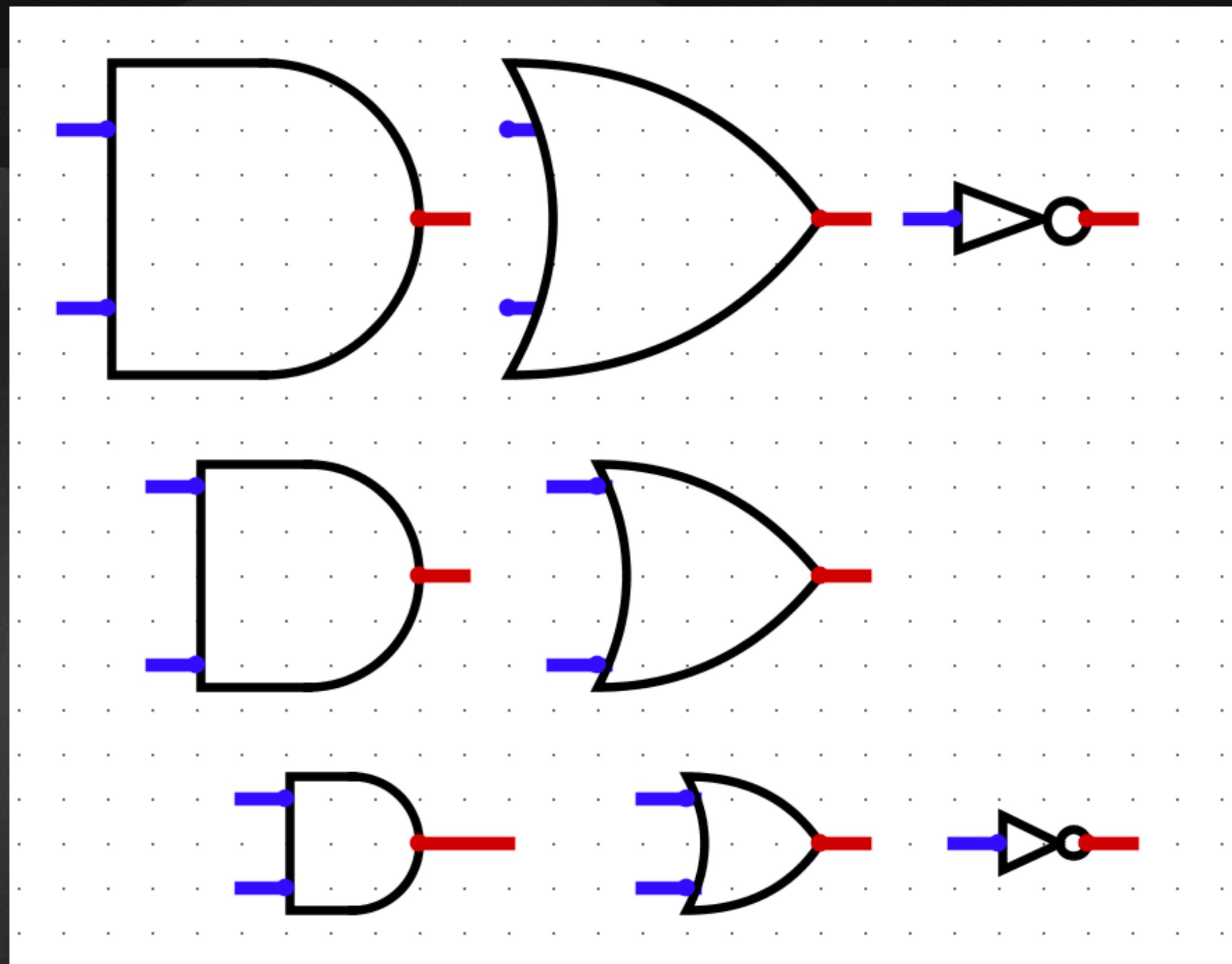
Use “LEGO pieces” wisely



Selection: Splitter	
VHDL	Verilog
Facing	East
<b>Fan Out</b>	<b>2</b>
Bit Width In	32
Appearance	Left-handed
Bit 0	0 (Top)
Bit 1	0 (Top)
Bit 2	0 (Top)
Bit 3	0 (Top)

# Logisim

Use “LEGO pieces” wisely

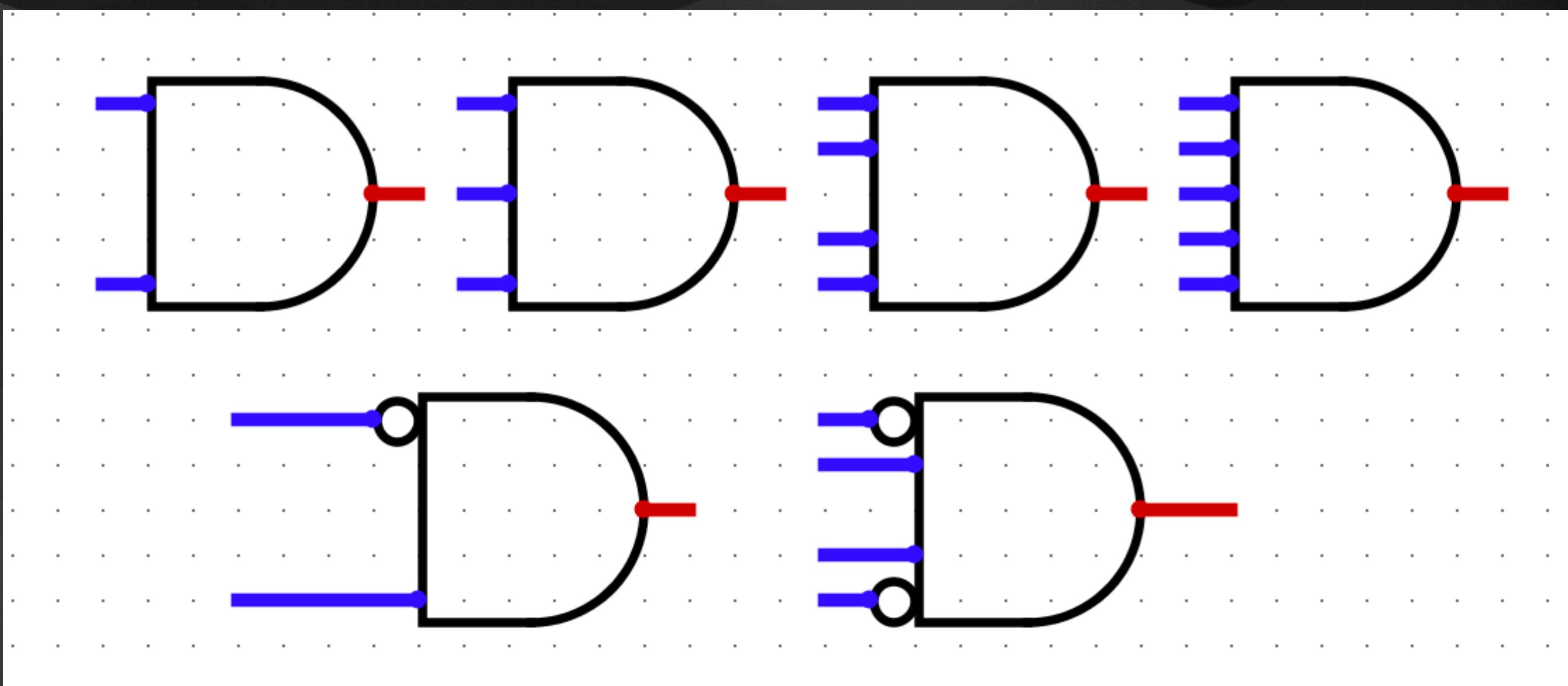


Properties		Registers
Selection: AND Gate		
VHDL	Verilog	
Facing	East	
Data Bits	1	
Gate Size	Wide	
Number Of Inputs	Narrow	
Output Value	Medium	
Label	Wide	
Label Font	SansSerif Bold 16	
Negate 1 (Top)	No	
Negate 2 (Bottom)	No	

Suting Chen

# Logisim

Use “LEGO pieces” wisely

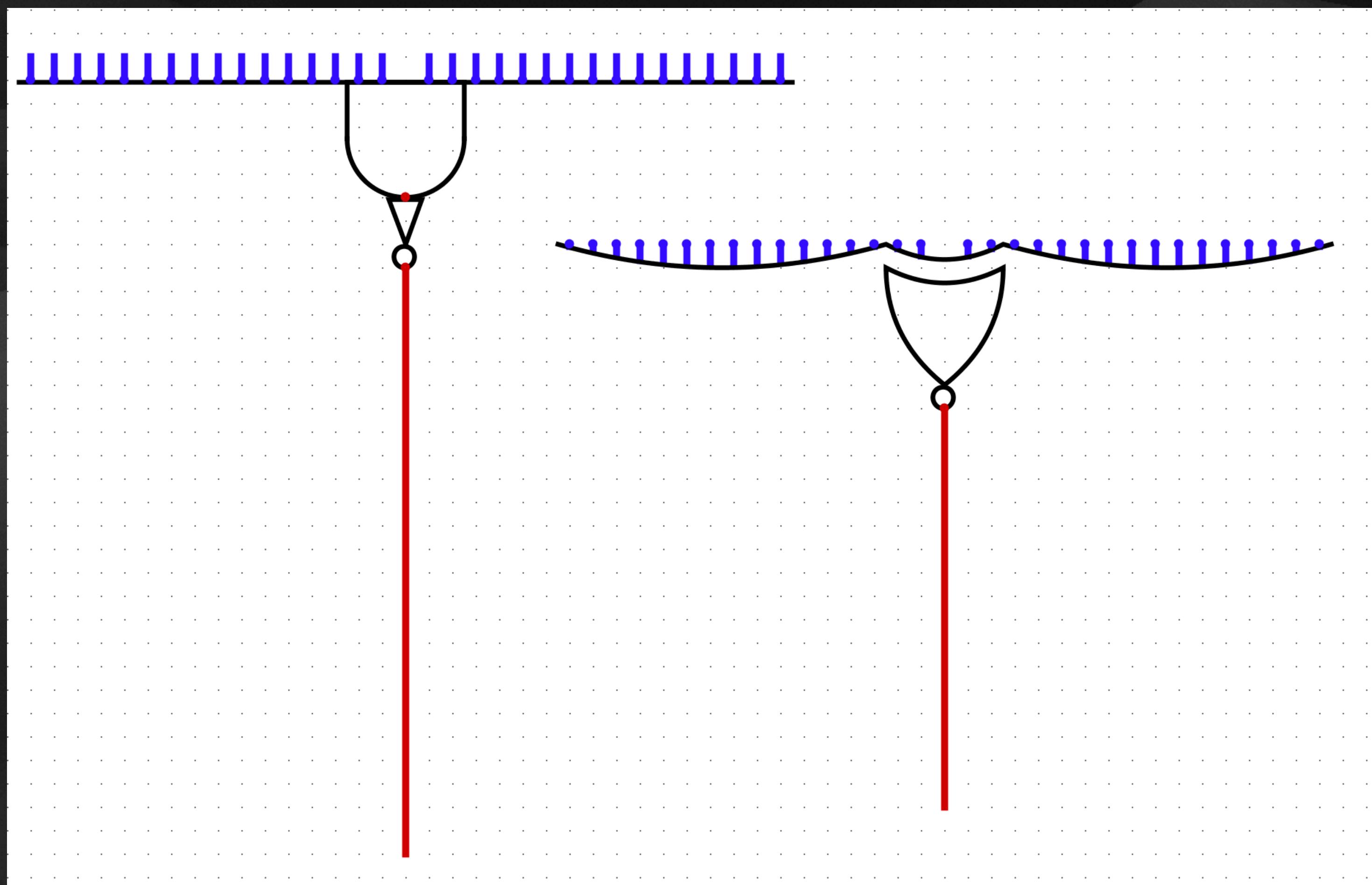


Properties		Registers
Selection: AND Gate		
VHDL	Verilog	
Facing	East	
Data Bits	1	
Gate Size	Medium	
Number Of Inputs	4	
Output Value	0/1	
Label		
Label Font	SansSerif Bold 16	
Negate 1 (Top)	Yes	
Negate 2	No	
Negate 3	No	
Negate 4 (Bottom)	Yes	

Suting Chen

# Logism

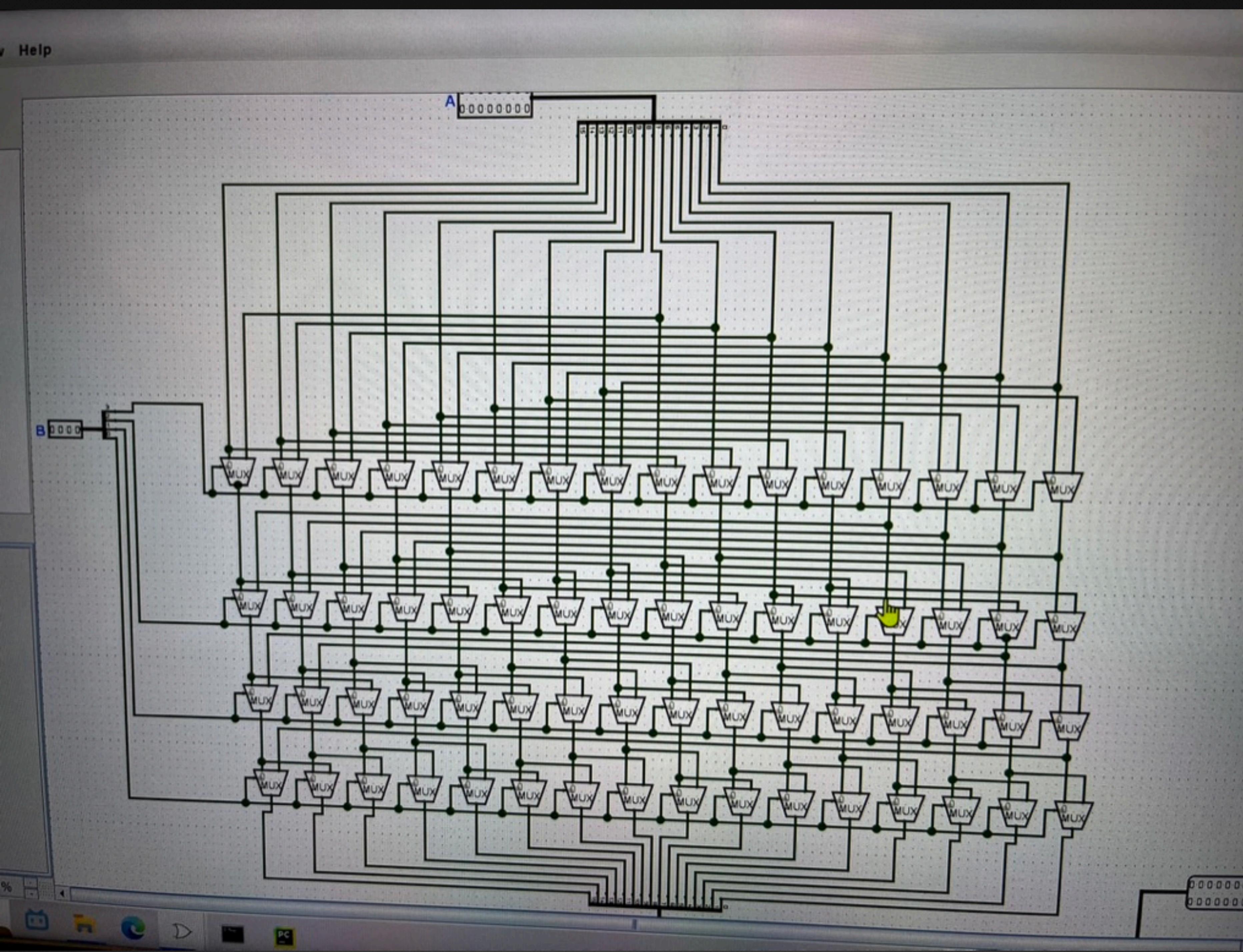
# Things you should avoid



# Suting Chen

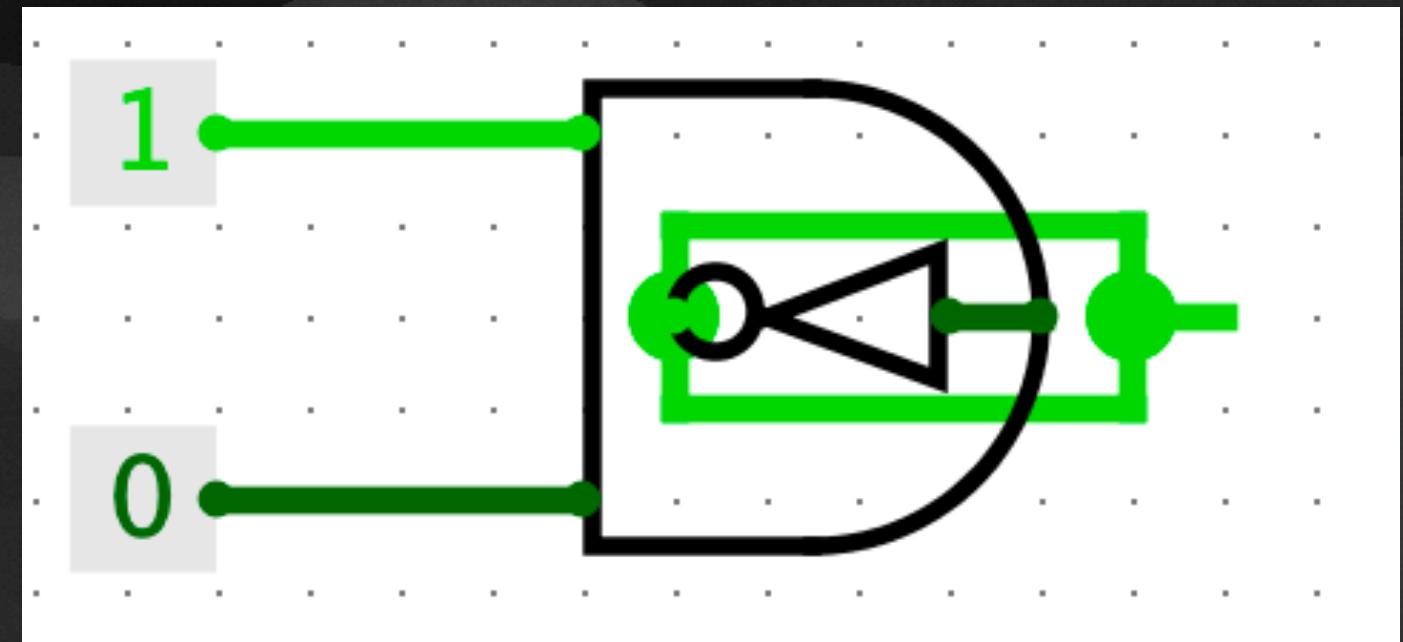
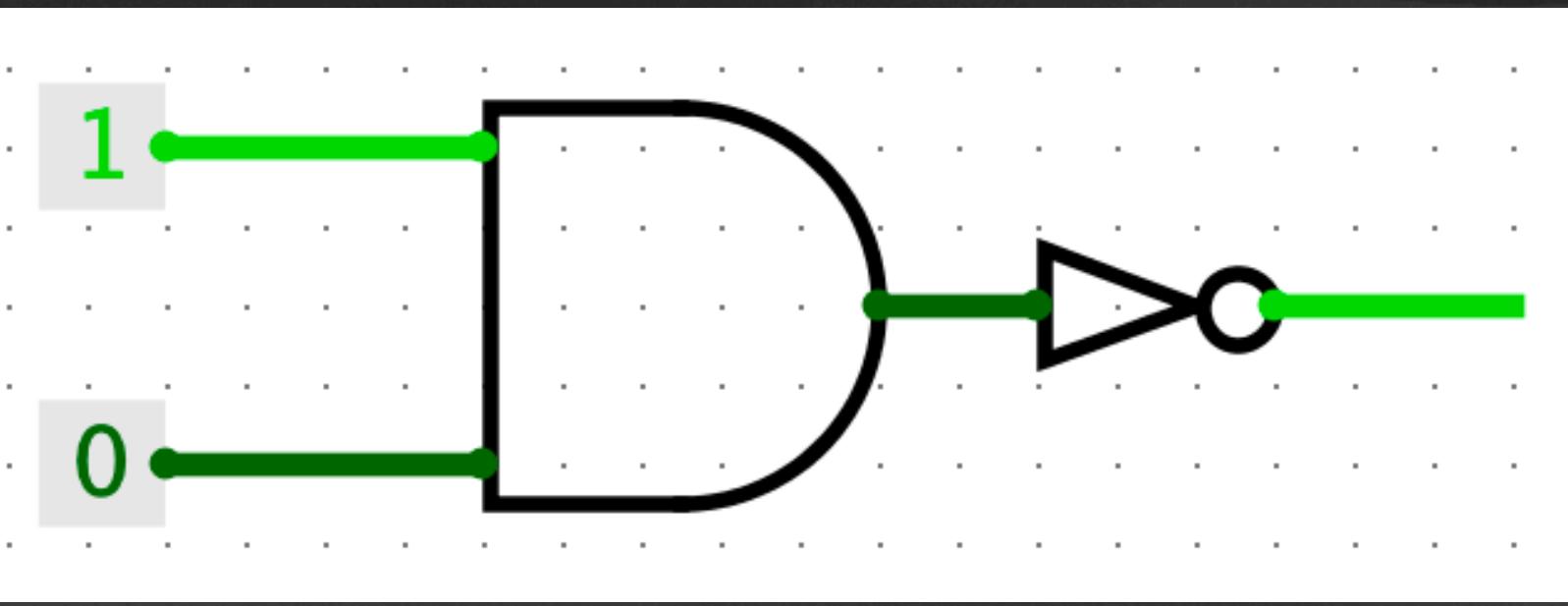
# Logis Things

Suting Che



# Logisim

## Things you should avoid



Suting Chen

Project 2.1 is easy, **BUT**

**Start early!**

... or **fail the course**

# Recommended readings

1. [https://robotics.shanghaitech.edu.cn/static/ca2020/CA2020\\_LogisimTutorial1.mp4](https://robotics.shanghaitech.edu.cn/static/ca2020/CA2020_LogisimTutorial1.mp4)
2. [https://robotics.shanghaitech.edu.cn/static/ca2020/CA2020\\_LogisimTutorial2.mp4](https://robotics.shanghaitech.edu.cn/static/ca2020/CA2020_LogisimTutorial2.mp4)
3. [https://robotics.shanghaitech.edu.cn/static/ca2020/CA2020\\_LogisimTutorial3.mp4](https://robotics.shanghaitech.edu.cn/static/ca2020/CA2020_LogisimTutorial3.mp4)

Suting Chen

# Recommended readings

## 4. Help -> Tutorial

Suting Chen

Logisim-evolution Documentation

### Beginner's tutorial

Next: [Step 0: Orienting yourself](#)

Welcome to Logisim!

Logisim allows you to design and simulate digital circuits. It is intended as an educational tool, to help you learn how circuits work.

To practice using Logisim, let's build a XOR circuit – that is, a circuit that takes two inputs (which we'll call  $x$  and  $y$ ) and outputs 0 if the inputs are the same and 1 if they are different. The following truth table illustrates.

$x$	$y$	$x \text{ XOR } y$
0	0	0
0	1	1
1	0	1
1	1	0

We might design such a circuit on paper.

But just because it's on paper doesn't mean it's right. To verify our work, we'll draw it in Logisim and test it. As an added bonus, we'll get a circuit that's looks nicer than what you probably would draw by hand.

[Step 0: Orienting yourself](#)

Logisim References

- Guide to Being a Logisim User
- Beginner's tutorial**
  - Step 0: Orienting yourself
  - Step 1: Adding gates
  - Step 2: Adding wires
  - Step 3: Adding text
  - Step 4: Testing your circuit
- Libraries and attributes
- Subcircuits
- Wire bundles
- Combinational analysis
- Menu reference
- Memory components
- Logging
- Command-line verification
- Test Vectors
- Application preferences
- Project options
- Value propagation
- HDL IP
- Users
- JAR libraries
- About the program
- Library Reference