Lab 1 – Preparation

Today's Tutorial

- Learning objectives for Lab 1
- Lab 1 parts
  - Warm-up exercise
  - What to hand in
- Intro to Logisim

Lab Room Assignments

- Labs take place in BA3145, BA3155 and BA3165
  - L0101: Mondays 6-9pm.
  - L0201: Wednesdays 6-9pm.

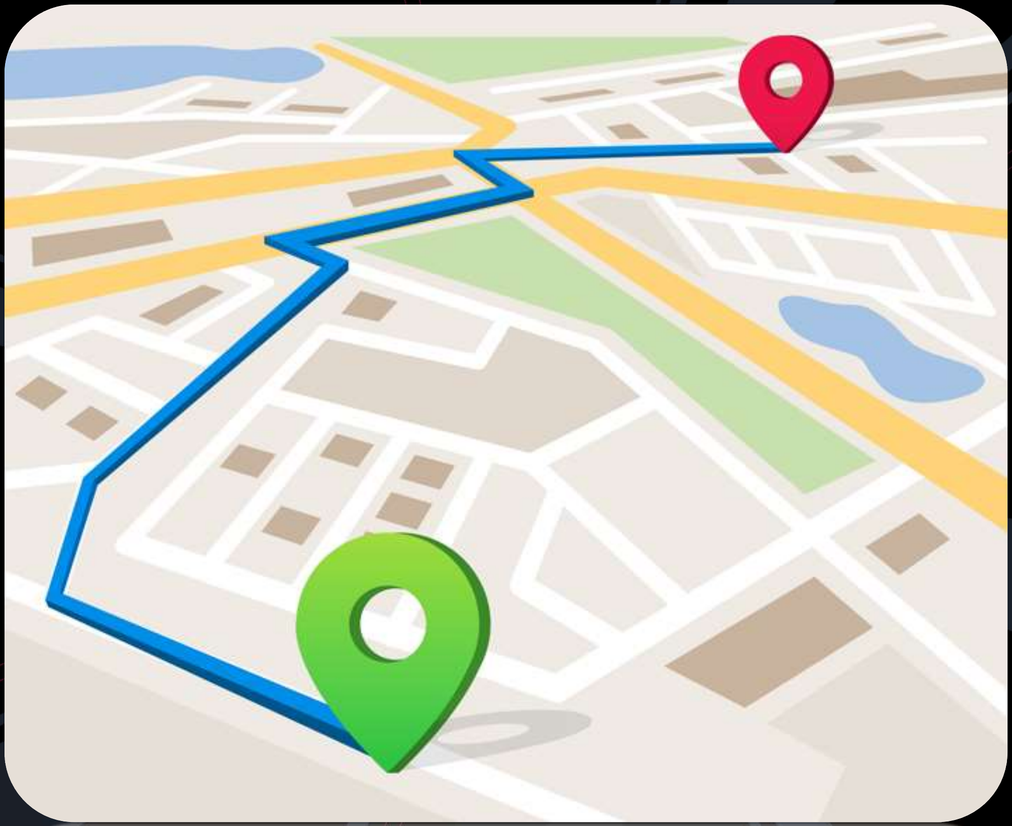
- Tentative lab assignments (boundaries might change slightly, depending on last-minute additions or drops in the class):
  - **L0101 (Mondays 6pm-9pm):**
    - BA3145: Last names Abbasi --> He
    - BA3155: Last names Henderson --> Pandey
    - BA3165: Last names Park --> Zhu
  - **L0201 (Wednesdays 6pm-9pm):**
    - BA3145: Last names Abu Arja --> Huang
    - BA3155: Last names Huynh --> Ren
    - BA3165: Last names Rizvi --> Zhou

About the Lab Rooms

- Rules about the labs:
  - These labs are only open and available during your dedicated lab section.
  - You must work in pairs for the labs, with the same partner.
  - The lab station you pick during your first lab will be yours for the entire term. You are responsible for its upkeep and maintenance.
  - **No food or drinks permitted in the labs!**
    - Please respect this to protect the lab equipment.

- A brief lab guide:
  - http://www-ug.eecg.toronto.edu/msl/handouts/labguide_DE1.html
  - Note: some parts apply mainly to engineering students.

Learning objectives



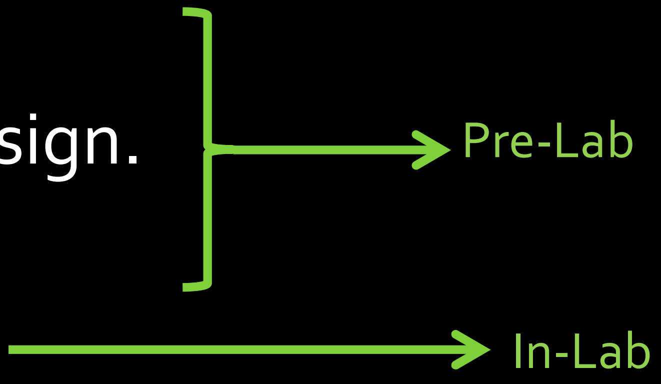
Lab 1 Learning Objectives

- What are the labs about in general?
  - Creating demo-worthy designs.

- What is Lab 1 about?
  - Learn how to build logical circuits by using logic gates and implement these in hardware.
  - Produce truth tables for a given design (starting either from a given logic function or from a description of the design’s behaviour).
  - Demonstrate familiarity with the graphic tool Logisim.

Approach to Lab 1

- Experience is the best teacher.
  - Prepare a design.
  - Implement your design.
  - Debug the circuit.
  - Demo your design



Pre-Lab

In-Lab

- The prelabs are the assignment you hand in before the beginning of the lab.
- The demos are the testing of your design and knowledge during the lab session.

Breaking down the labs

Lab 1 breakdown

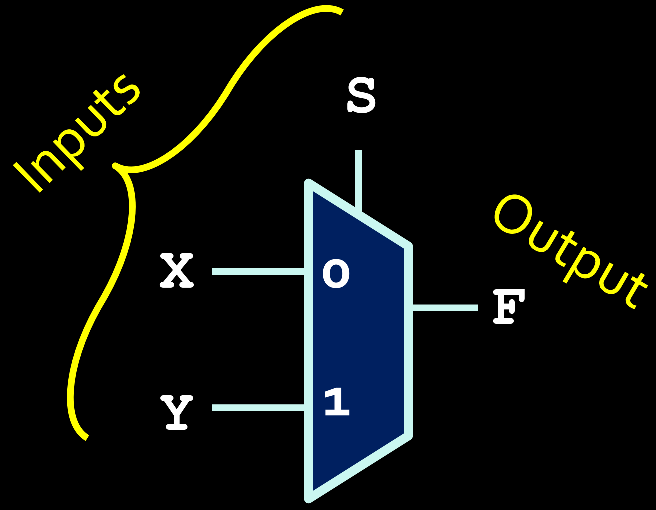
- Mark breakdown:

Pre-lab:	1 mark
Part I:	1.5 marks
Part II:	1.5 marks

Lab 1 prelab: Part 1

- Part I:
  - Design the circuit for a multiplexer:

$$F = X\overline{S} + YS$$



- This is a common short-hand expression for the output wire F.
- In this case, the value for F is the output of: $(X \wedge \neg S) \vee (Y \wedge S)$

Lab 1 prelab: Part 1 (cont’d)

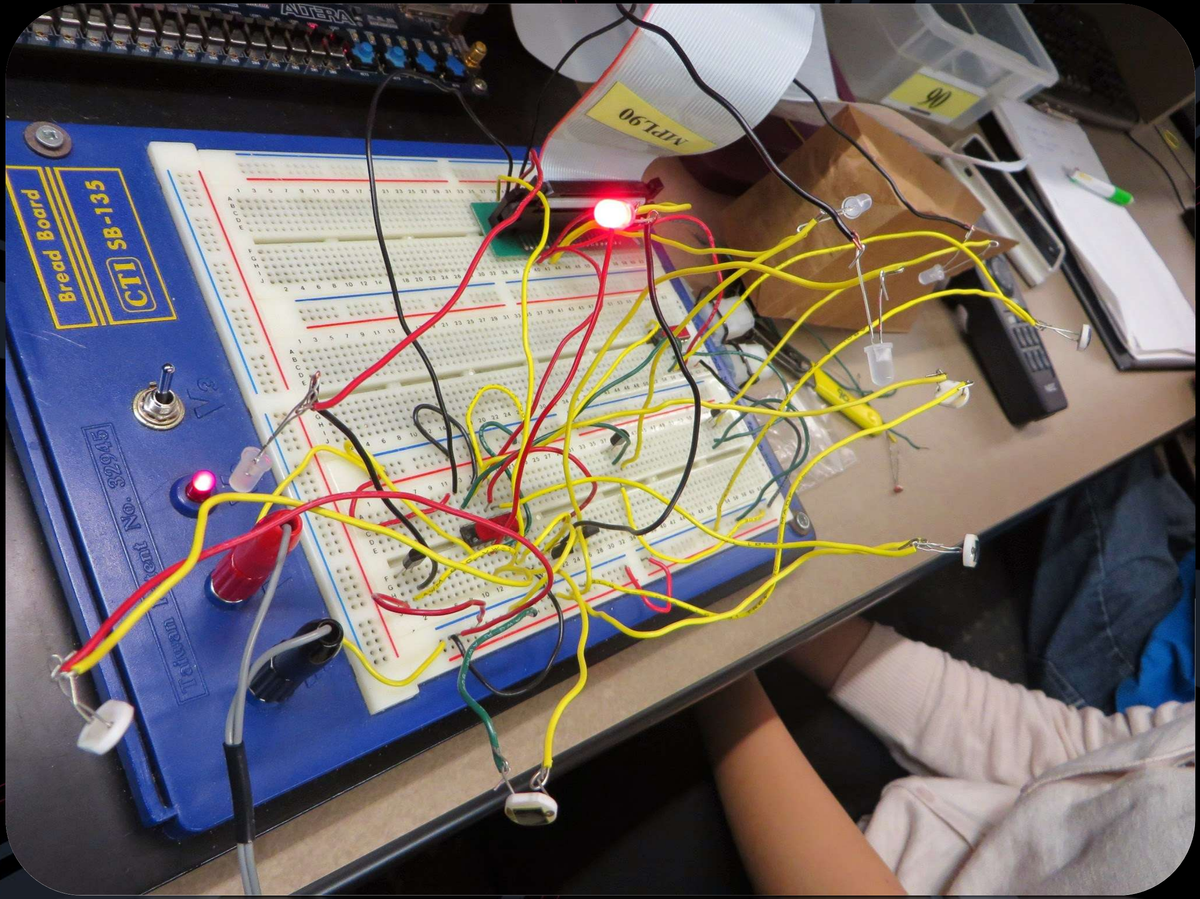
Note: The following are different ways of expressing the same thing:

- $F = X\overline{S} + YS$
- $F = X*S' + Y*S$
- F = (X and (not S)) or (Y and S)

Multiplication symbols are used to represent AND, while addition symbols represent OR.

- Demonstrate your knowledge in Part 1 by preforming the following tasks:
  - Show the truth table for the three inputs X, Y & S and the output F.
  - Represent this logical expression in gates.
  - Show how you’d build this circuit using IC chips

Using Lab Hardware

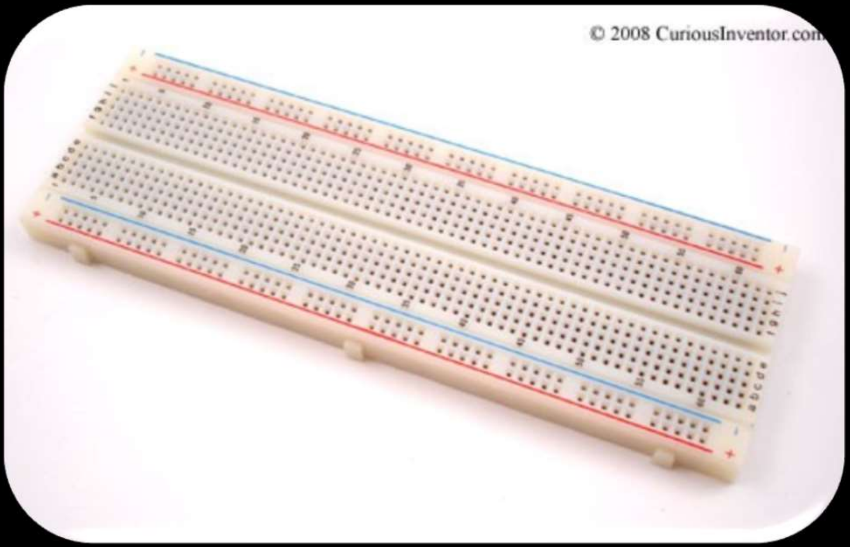


Lab Equipment

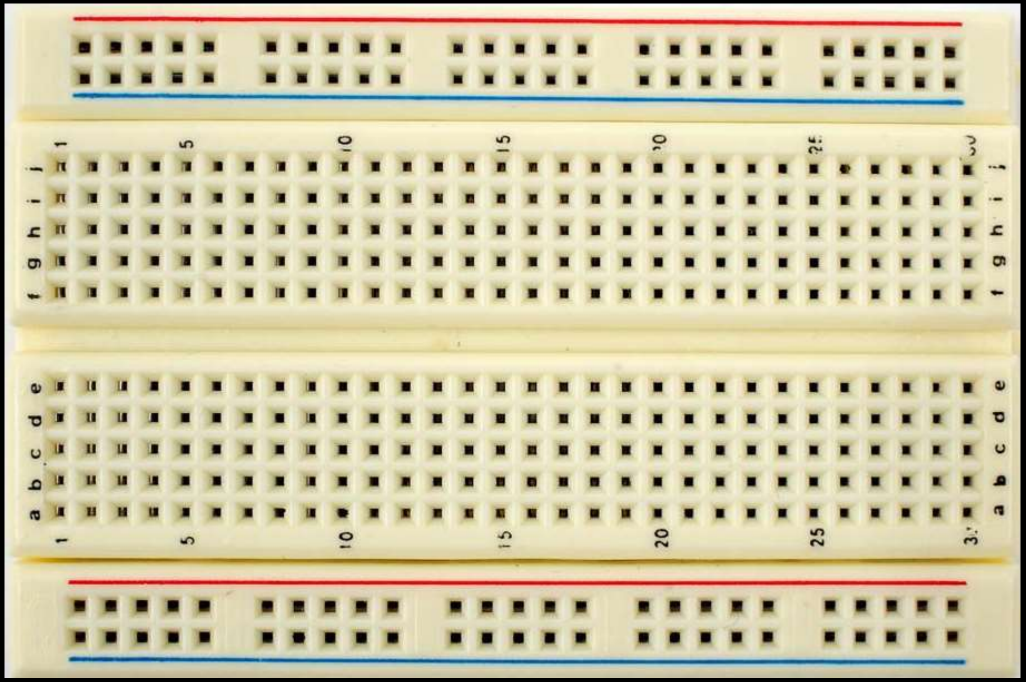
- You will only use the in-lab hardware for a couple of labs, to help you understand how it works and to get your hands dirty.
- Some of the equipment you'll see:
  - Breadboard
  - Wires
  - Gates (in chips)

Breadboard

- The standard working area for connecting digital components together.
- Red and blue horizontal rows at top are connected.
- Columns in middle sections are connected.

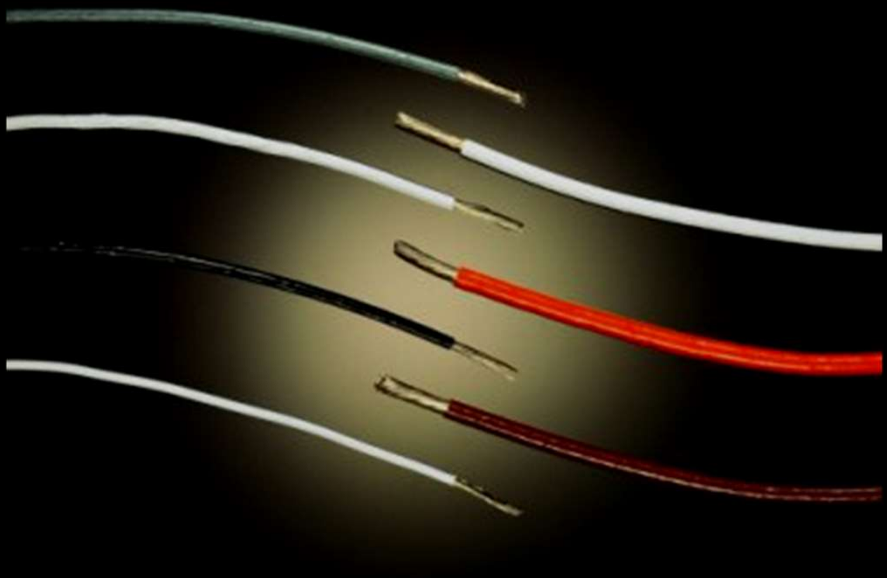


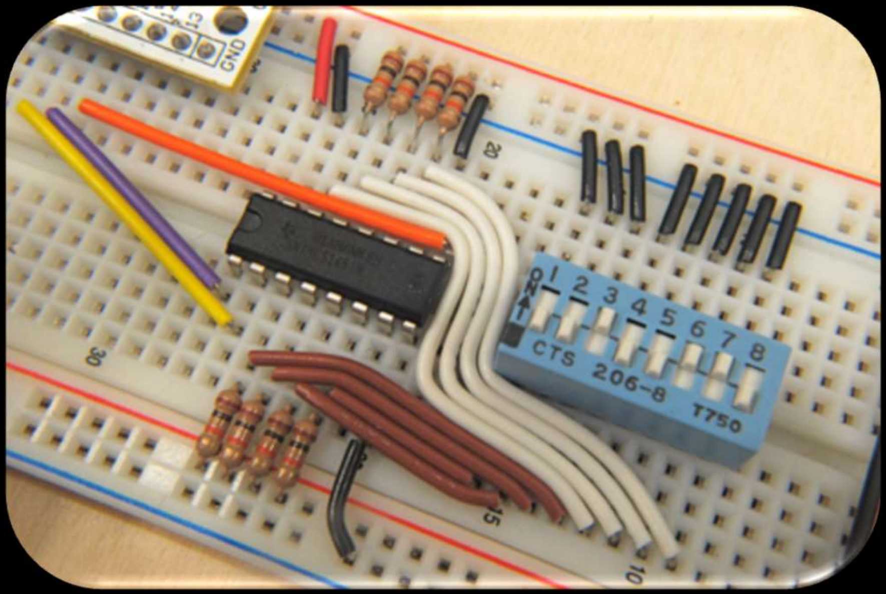
© 2008 CuriousInventor.com



Wires

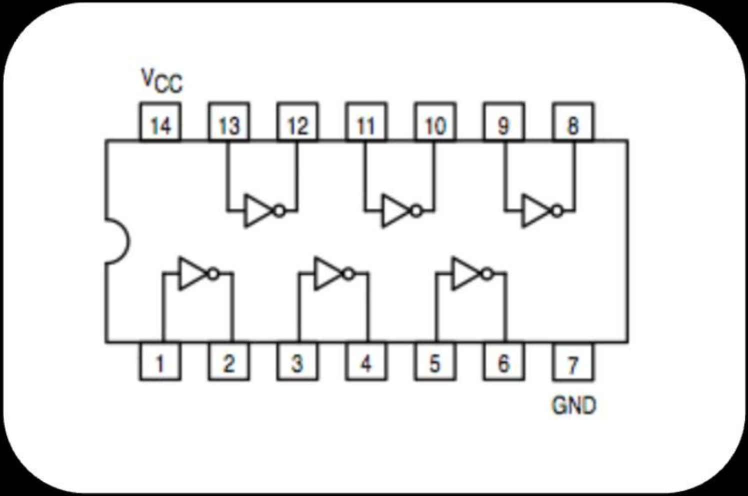
- The TAs will teach you to:
  - Use wires to connect different components together.
  - Use pre-cut wires whenever possible.
  - Use wire strippers to strip the coating off the end of a wire.





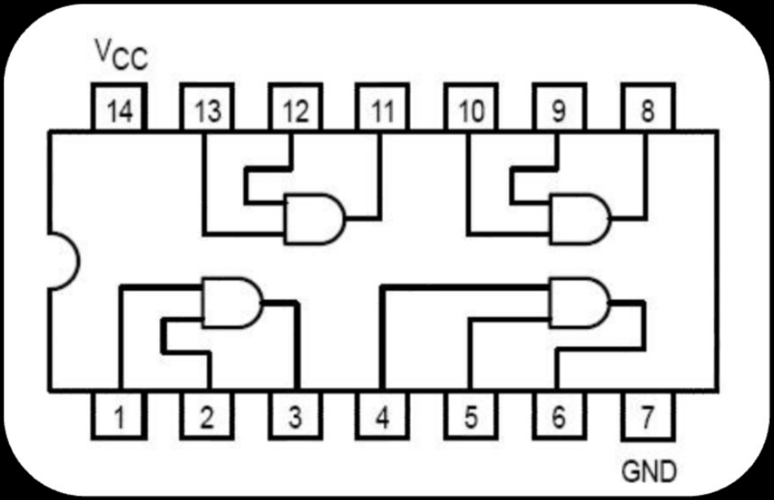
Gates

- IC chips will be supplied, which house the gates that you will use to create circuits.
- Example: 74LS04 (NOT)
  - Notch at one end helps determine alignment.
  - Usually a dot at pin #1.
  - V_{cc} and GND always have to be connected to the power source and the ground, respectively.

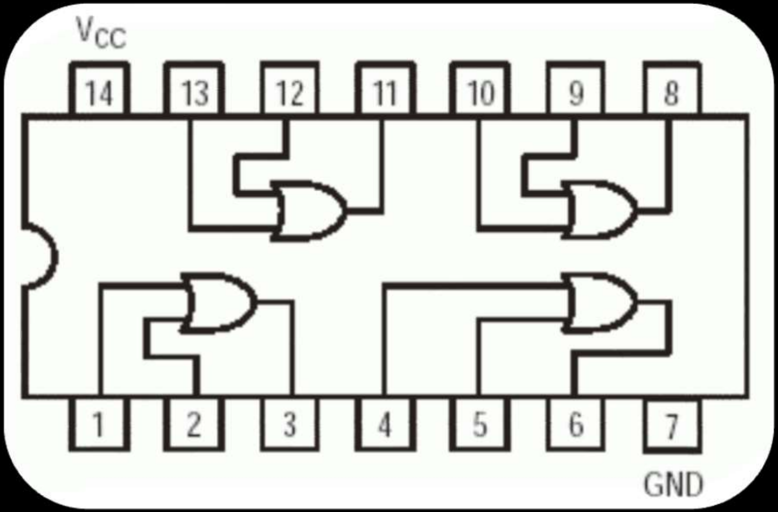


Other Gates

- 74LS08 (AND)

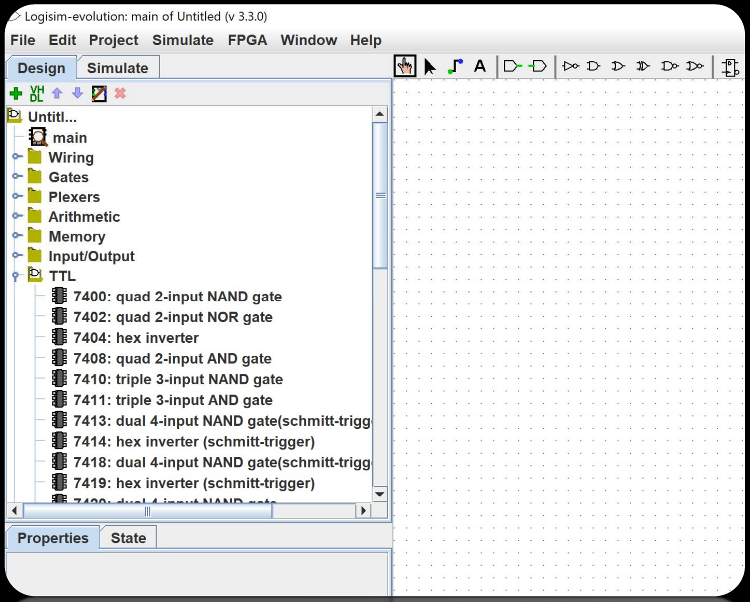


- 74LS32 (OR)



Part 1 (final step)

- Implementing your design in Logisim!
  - For later labs, you'll design circuits using the AND, OR and NOT gates in Logisim's Gates toolkit.
  - For this lab, you'll be implementing your circuits using IC chips in the lab.
  - For Part 1 and Part 2, you'll be using the chips in the TTL toolkit to design your circuit.
  - You'll use the Gates in Part 3


Logisim-evolution: main of Untitled (v 3.3.0)
File
Edit
Project
Simulate
FPGA
Window
Help
Design
Simulate
VHDL
Untitl...
main
Wiring
Gates
Plexers
Arithmetic
Memory
Input/Output
TTL
7400: quad 2-input NAND gate
7402: quad 2-input NOR gate
7404: hex inverter
7408: quad 2-input AND gate
7410: triple 3-input NAND gate
7411: triple 3-input AND gate
7413: dual 4-input NAND gate(schmitt-trigg
7414: hex inverter (schmitt-trigger)
7418: dual 4-input NAND gate(schmitt-trigg
7419: hex inverter (schmitt-trigger)
Properties
State

Lab 1 prelab: Part 2

- Part II:
  - Given the function:

$$f = (a+b)' + cb'$$

  - How would you implement this in gates?
  - What is the minimal number of gates you need?
- Part III:
  - Implement these circuits in Logisim.
  - Test your designs (using Poke tool and test files)

Warm Up Example

- Design a circuit that implements the following logic function, using only 2-input AND and 2-input OR gates.

```
f = a*b + (c + b)
```

- Write down the truth table for this design.
  - Note: This expression is common shorthand for:

```
f = a AND b OR (c OR b)
```

Warm Up Example cont’d

- Is there a cheaper implementation (i.e., with fewer gates)?
  - `f = a*b + (c + b)`

What you're going to do

1. Determine the Boolean logic equation that you need to implement.
   - Might require you to create a truth table first.
2. Convert this equation into an equivalent circuit of AND, OR and NOT gates (using order of operations when necessary).
3. Determine how many chips you'll need to implement all the gates for this circuit.
4. For each gate in your diagram, indicate the IC pin number for each input and output.

What to hand in

Prelab vs Lab Demo

- Prelab exercises are due before 6pm on your lab day.
  - Written/hand-drawn elements in PDF files.
  - Logisim circuits as `*.circ` files.
  - ~~Logisim tests as `*.txt` files.~~
- TAs will ask to look at your Logisim designs, so have those ready to share with them.
  - Also be ready to share the hand-written elements in case a question arises about your design process.

Pre-lab reports

- The written report should include the following:
  - Lab number and title
  - Student info (last name, first name, student #)
  - Exercise parts
    - Each in its own clearly-labeled section.
    - Restate the question (summarized).
    - Provide the calculations (if applicable).
    - Illustrate the solution (including pin labels).
  - PLEASE BE NEAT.
- The Logisim files you submit should be named to reflect the lab number and part number.
  - e.g. `lab1_part2.circ`

In-Lab Tasks

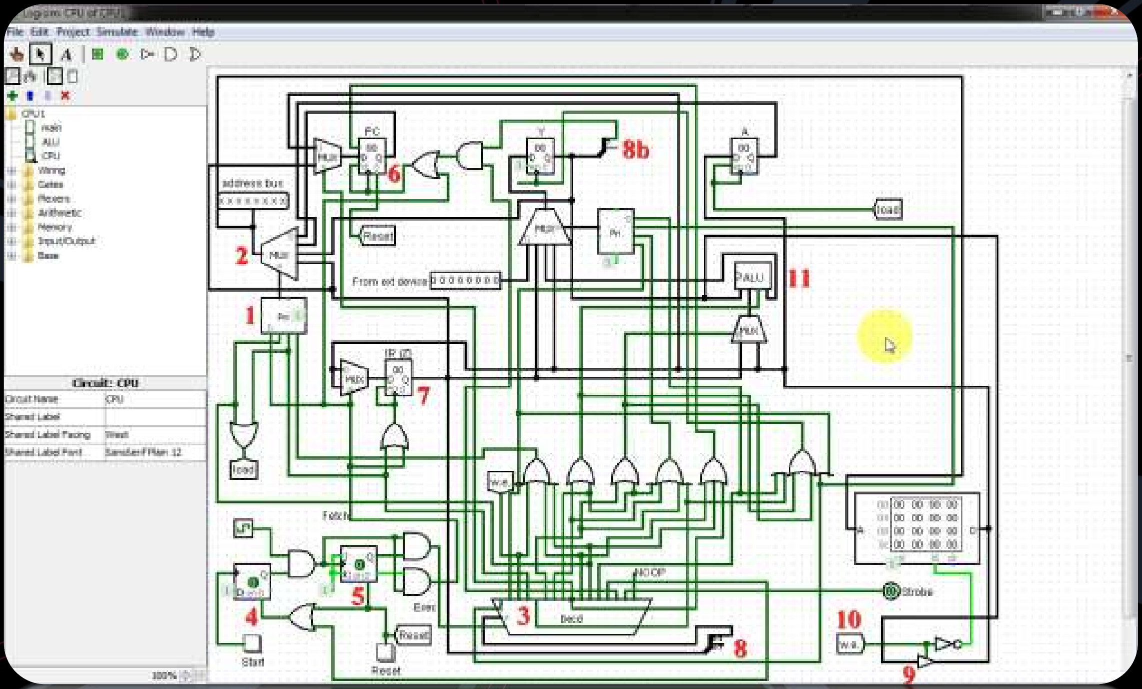
- The TAs will demonstrate how to plug the chips and wires into the breadboard, and how to connect the breadboard to the lights and switches on the side.
  - Don’t be late to the first lab!!

- Helpful tips:
  - Remember to turn off the power supply when connecting/plugging in components in the breadboard.
  - Using some sort of colour convention for your wires can be helpful.
    - e.g. have all wires connected to the ground be one colour and use another colour for all `Vcc` wires.
  - Use the logic probe (provided in the lab kit) to test each connection when debugging!

Things to note

- This will be the easiest lab you do in the course.
- Whenever possible, use the tools and submit a printed pre-lab report.
- Try to come up with the smallest circuits possible.
  - How do you reduce a complex circuit?
  - For now, think back to boolean algebra axioms!
  - Simple reasoning helps as well ☺

Intro to Logisim

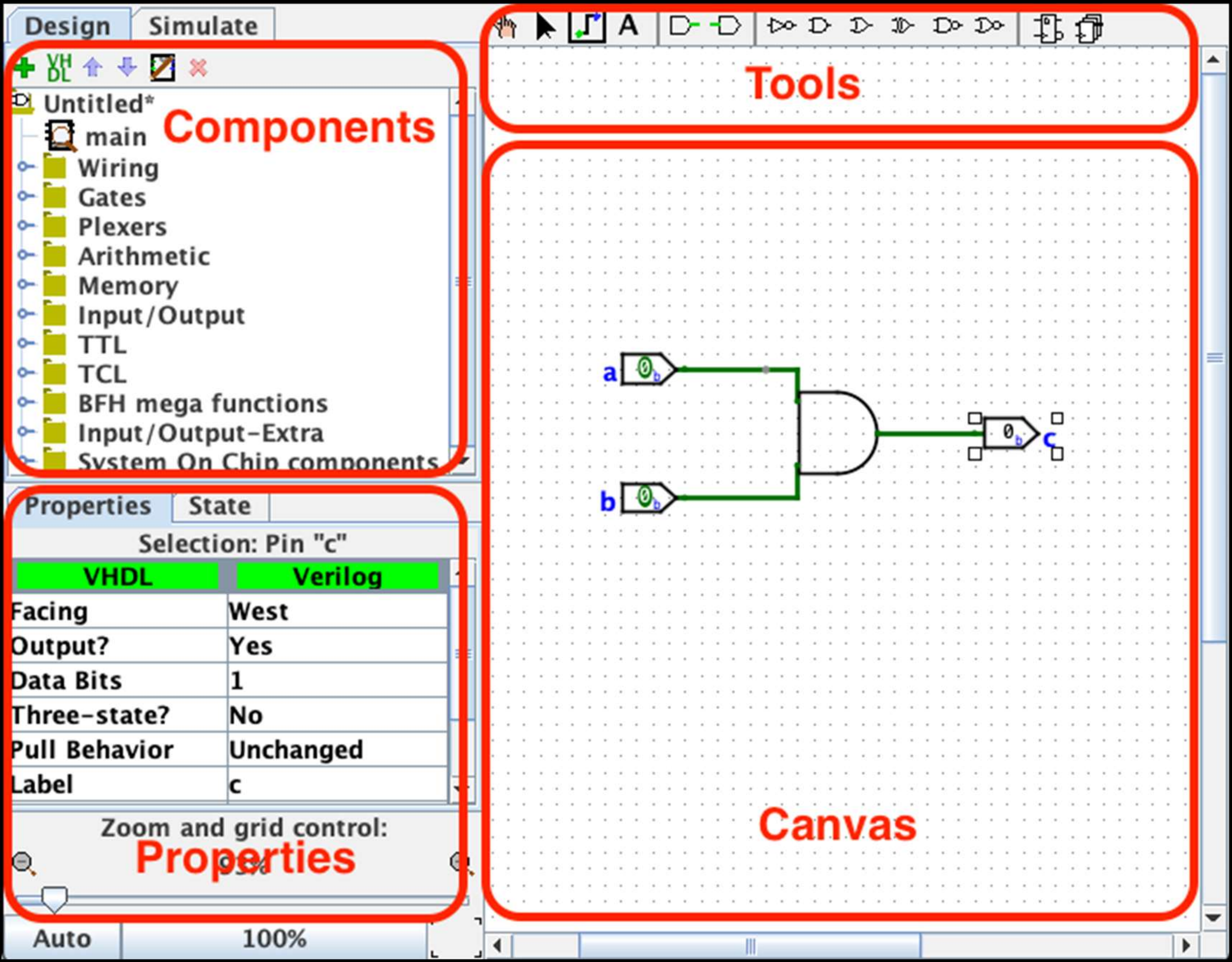


Circuit: CPU	
Circuit Name	CPU
Shared Label	
Shared Label Facing	West
Shared Label Font	SansSerif Plain 12

Logisim installation

- Logisim is a powerful logic circuit simulation environment.
  - In this course, we will be using version 3.6.1 (or later):
    - https://github.com/reds-heig/logisim-evolution/releases/tag/v3.6.1
    - Just the `jar` file is needed.
- Note:
  - Make sure to use Logisim-Evolution downloaded at the above link. Do NOT use the original Logisim or any other variations or versions of it.
  - You will need Java 13 installed to launch this.

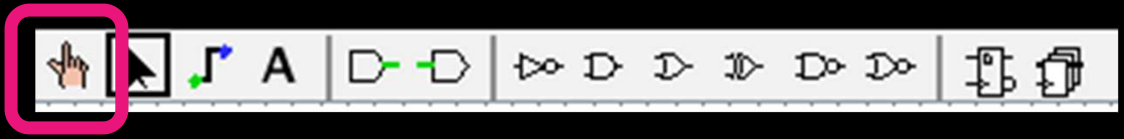
Logisim walkthrough



Selection: Pin "c"

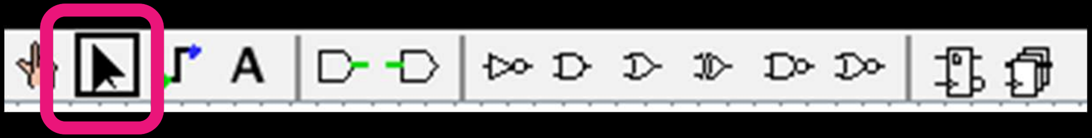
VHDL	Verilog
Facing	West
Output?	Yes
Data Bits	1
Three-state?	No
Pull Behavior	Unchanged
Label	c

Tools/Views



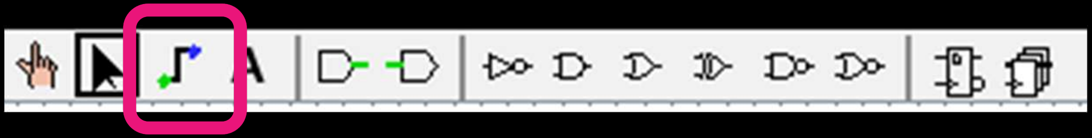
Poke: Click on wires to inspect their state, click on most components to change their state.

Tools/Views



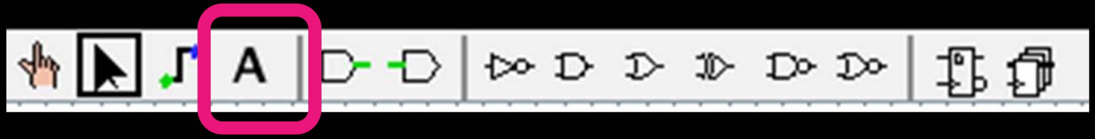
Select: Selects and moves things in the canvas, and manipulates wires/buses. Click and drag from inputs/outputs to create wires/buses.

Tools/Views



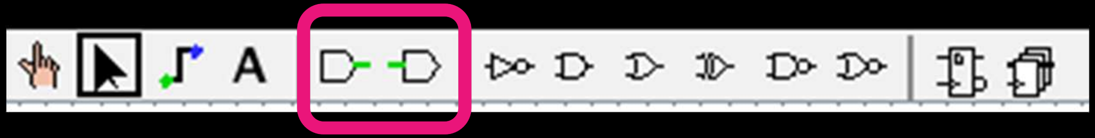
Wire: Creates wires on the canvas.

Tools/Views



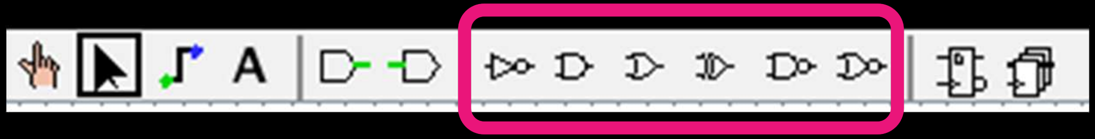
Text: Add text on the Canvas.

Tools/Views



Default Input/Output: default type of input and output for the circuits. You will be using them a lot throughout the course.

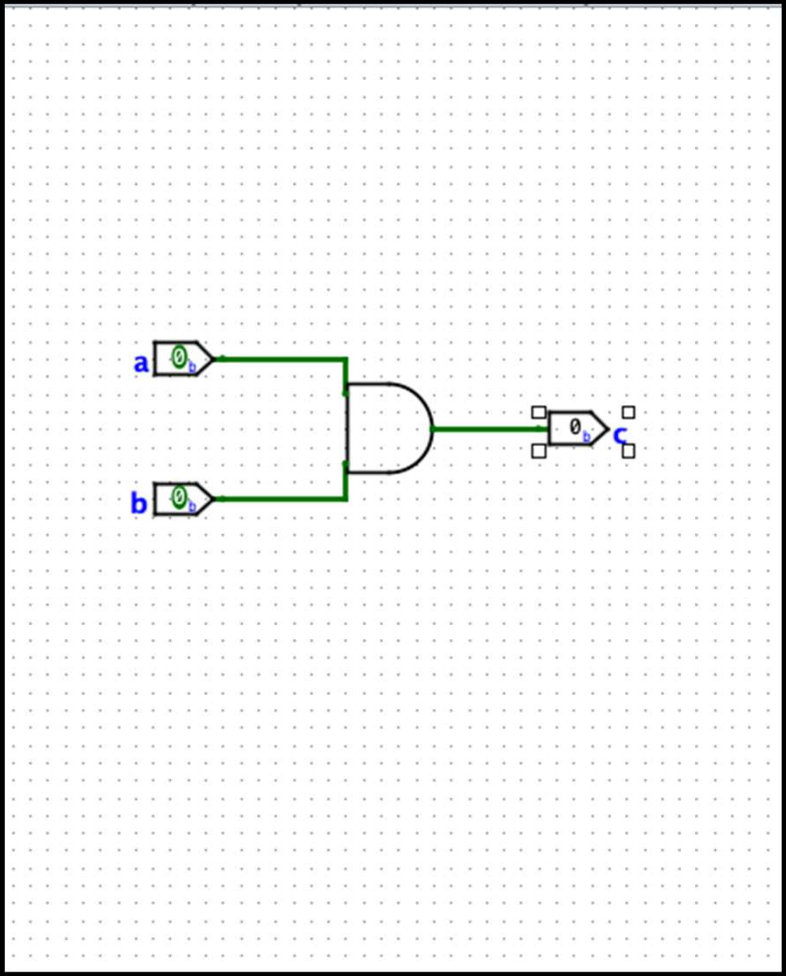
Tools/Views



Logic gates: some commonly used logic gates that you can click and drop on the canvas to build your circuits.

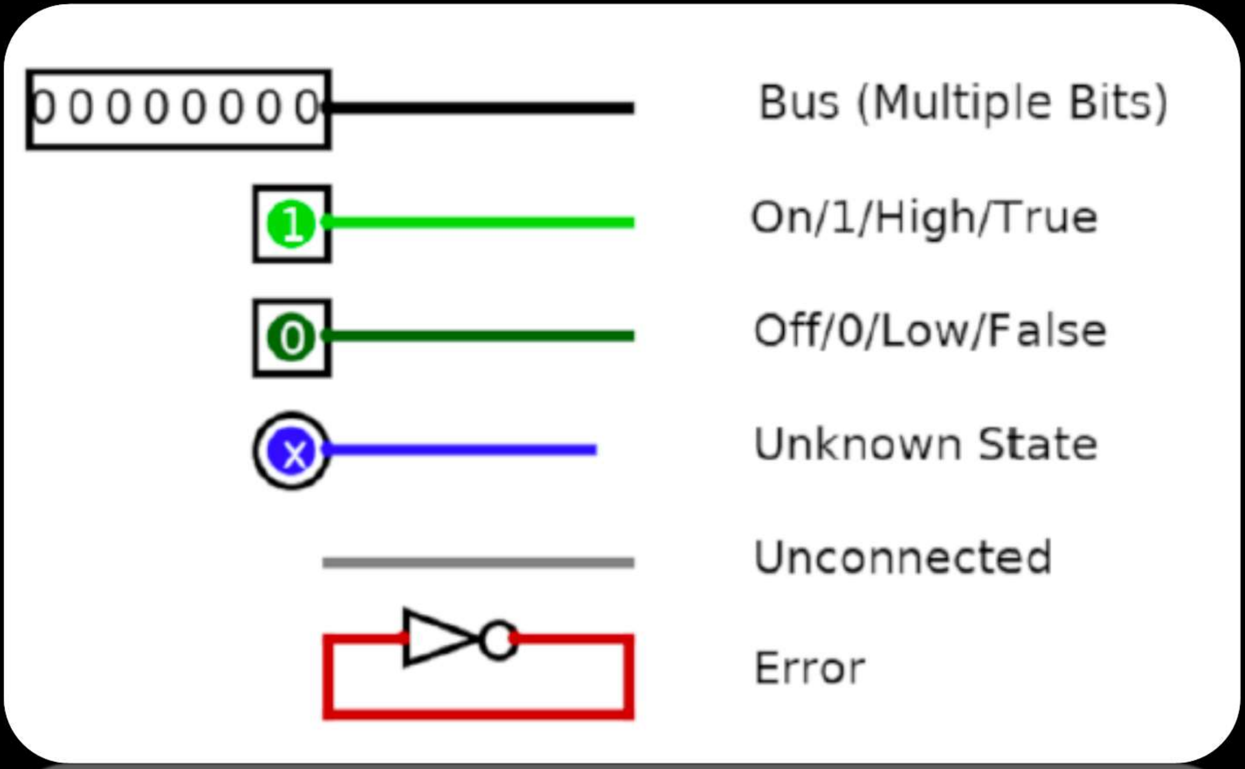
Canvas

- Canvas is where you will be building your circuits by dropping components on the canvas and then connect them with wires.



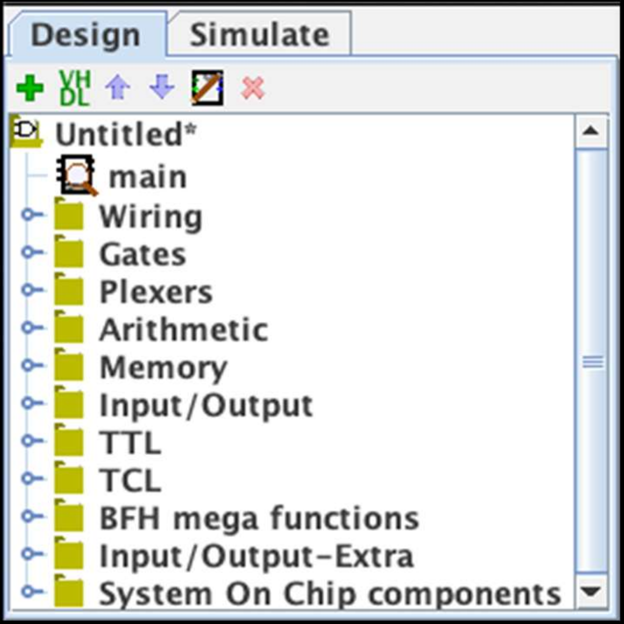
Wires

- Wires and buses can have many states. You can inspect the state of a wire/bus using **poke** in the toolbar.



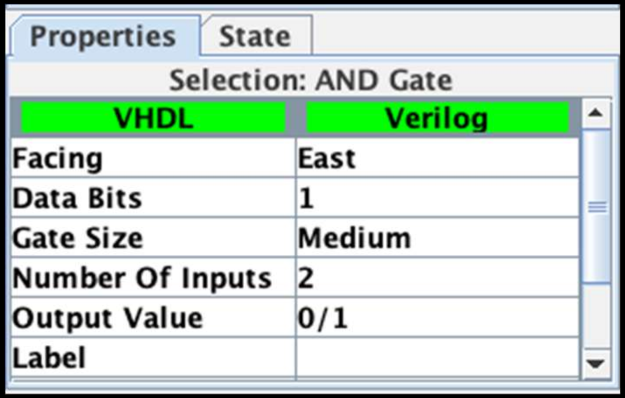
Components

- This contains all the circuits in this lab as well as all the built-in components.
- Double-click on each circuit to view it. To place a component from this list, select it, and then click somewhere in the canvas.
- You can add/delete a circuit using the green + sign and the red x sign on the top.



Properties

- If you click on a component on the canvas, you would be able to view and edit its properties.
- For example, this is the properties for an AND gate. You can change the number of input bits and number of inputs here. This will be useful in the future.

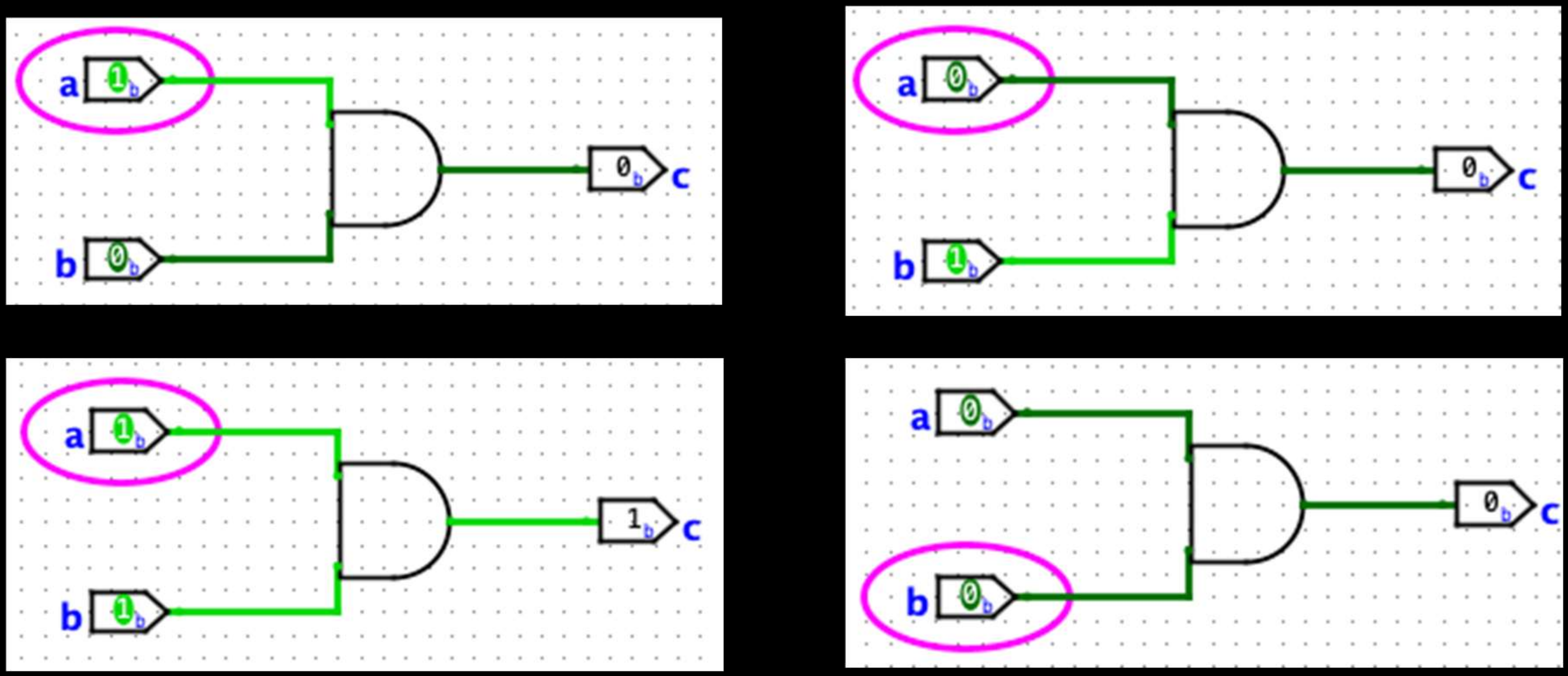


Selection: AND Gate

VHDL	Verilog
Facing	East
Data Bits	1
Gate Size	Medium
Number Of Inputs	2
Output Value	0/1
Label	

Testing in Logisim

- The easiest and most visual way of testing is using the Poke in the tool bar and click on the components to change the state. This will be very useful throughout the course so make sure you try this out.



Testing in Logisim

- Another way is through test vector files. (details can be found in the lab handout)
- Steps involved:
  1. List the truth table for your circuit, the values for the inputs and the expected values for the outputs
  2. Logisim will be able to run the tests according to your truth table to test the functionality of your circuit.