Lab 1 – Preparation

Reminders

- Labs take place in BA3145, BA3155, BA3165
  - L0101: Wednesdays 6-9pm.
  - L0201: Mondays 6-9pm.
  - L5101: Tuesdays 6-9pm.

- Some Rules:
  - These labs are only open and available during your dedicated lab section.
  - You must work in pairs.
  - After your first lab you will be assigned a lab station for the entire term.
  - **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.html
  - Note: some parts apply mainly to engineering students.

© Steve Engels, University of Toronto

Lab 1 Learning Objectives

- Learn how to build logic circuits by using chips that contain individual logic gates .
- Produce truth tables for a given design (starting either from a given logic function or from a description of the design's behaviour).
- Gain familiarity with the schematic builder tool of Quartus.

© Steve Engels, University of Toronto

Preparing for Lab 1

- Experience is the best teacher.
  - Preparing a design. → Pre-Lab
  - Implementing your design. → In-Lab
  - Debugging the circuit. → In-Lab
- Try to think of your prelabs as "an assignment due in the beginning of the lab".

© Steve Engels, University of Toronto

Equipment

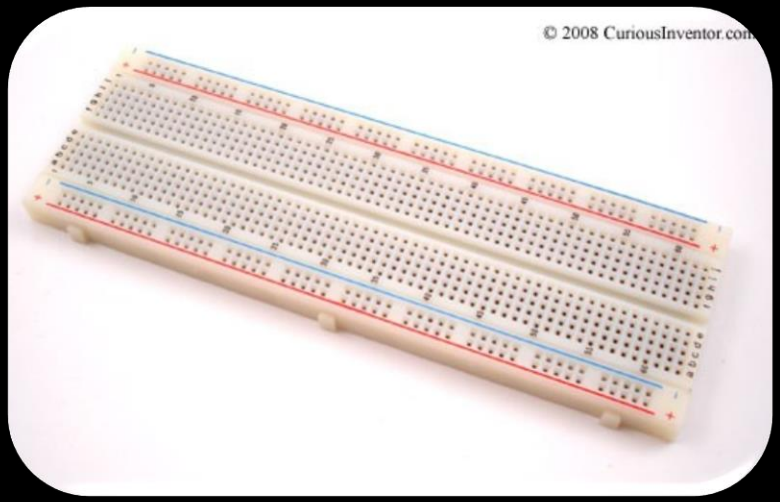
- Breadboard
- Wires
- Gates

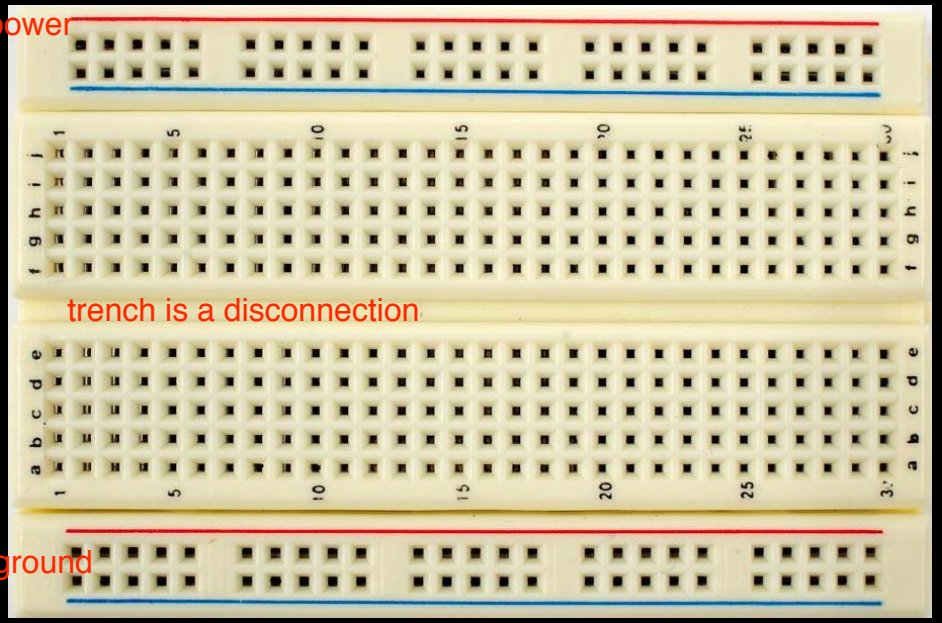
© Steve Engels, University of Toronto

A breadboard is a construction base for prototyping of electronics.

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.

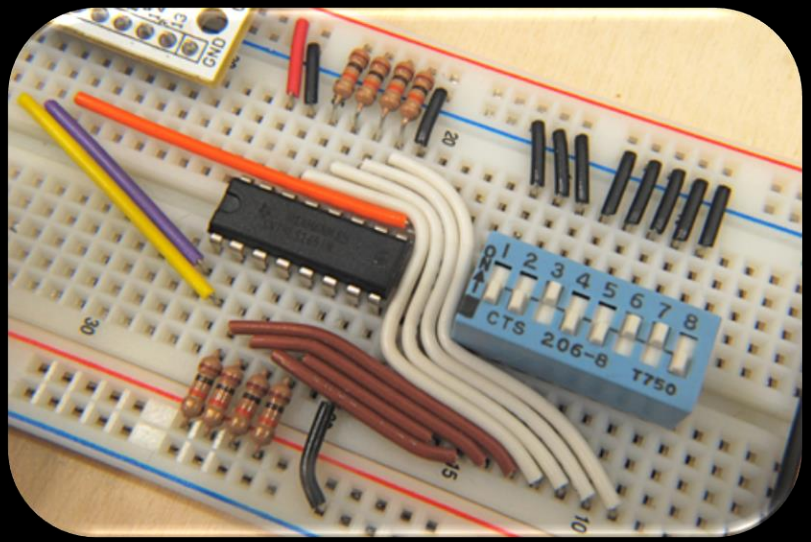




© Steve Engels, University of Toronto

Wires

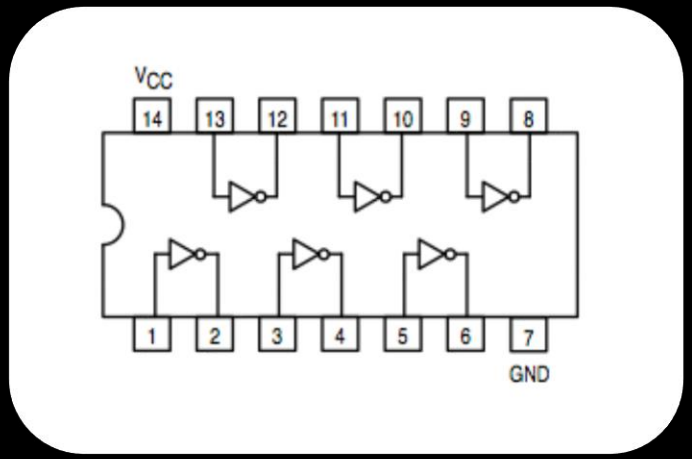
- Use this to connect different components together.
- Use the pre-cut wires whenever possible.
- Learn how to strip the coating off the end of a wire.



© Steve Engels, University of Toronto

Gates

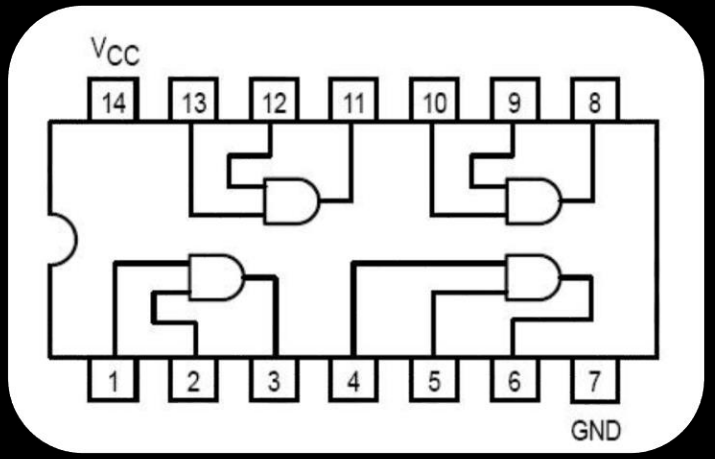
- IC chips will be supplied, which house the gates that you will use to create circuits.
- Example: 74LS04 (NOT)
  - Notch at one end helps determine alignment.
  - Usually a dot at pin #1.
  - V_{cc} and GND always have to be connected to the power source and the ground, respectively.



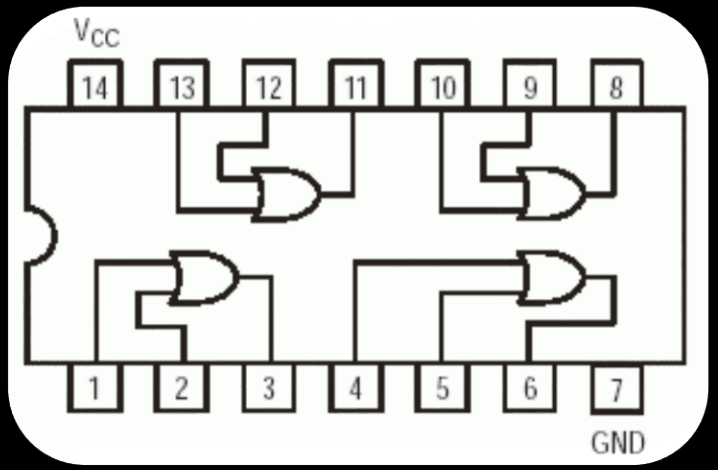
© Steve Engels, University of Toronto

Other Gates

- 74LS08 (AND)



- 74LS32 (OR)



© Steve Engels, University of Toronto

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!

© Steve Engels, University of Toronto

Pre-lab 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).
  - BE NEAT.

© Steve Engels, University of Toronto

Warm Up Example

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

$$f = ab + (c + b)$$

- Write down the truth table for this design.

© Steve Engels, University of Toronto

Warm Up Example cont'd

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

use absorption law b + ba = b; so f = ab + c + b = b + c

© Steve Engels, University of Toronto

Things to note

- This will be the easiest lab you do in the course.
- Whenever possible, use the tools and bring in 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 ☺

© Steve Engels, University of Toronto

After the lab reflect on

..how long it took you to implement even simple logic circuits on the breadboard.

- Can you imagine doing this for more complex circuits?!
  - Hardware Description Languages (HDL), like Verilog, and design tools to the rescue! ☺

© Steve Engels, University of Toronto