

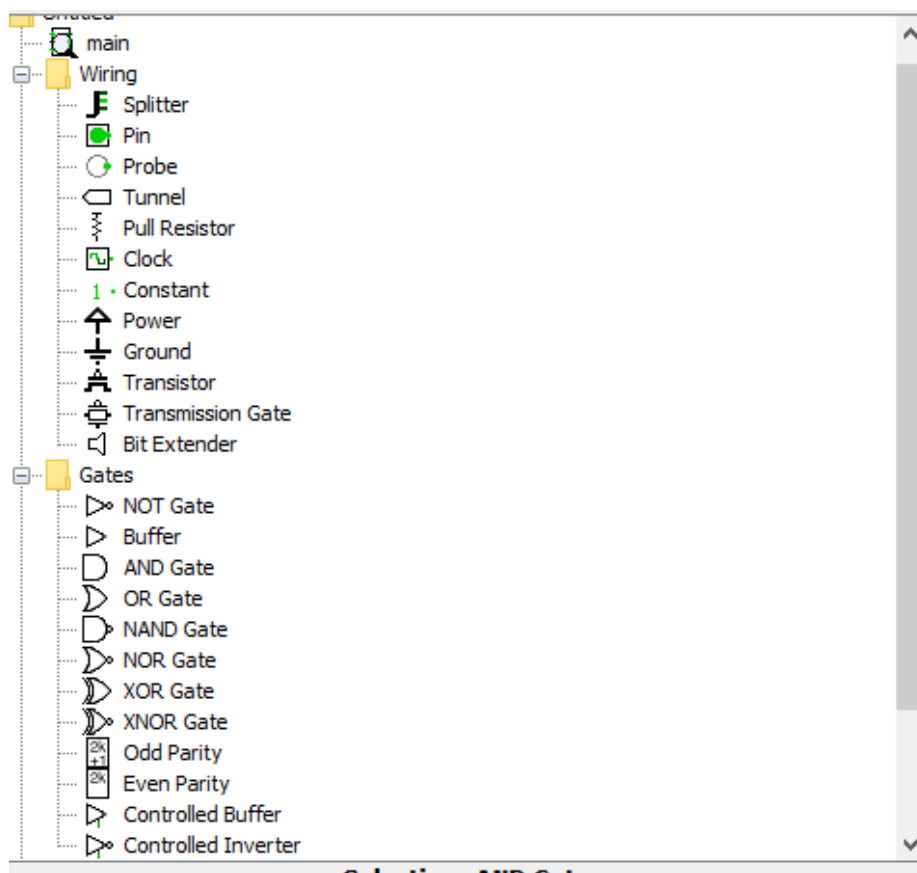
# Lab 3

Published ⋮

Required lab tools: Logisim, Arduino

## [Step 1: Familiarizing Logisim]

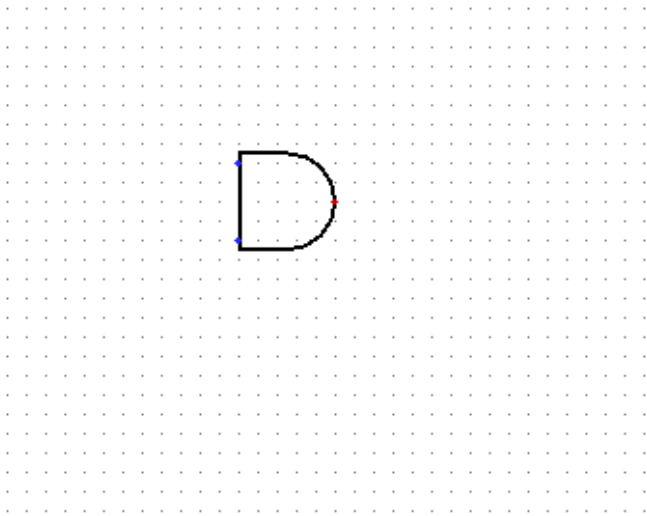
Logisim has a toolbox on the left of the screen and the "canvas" on the right of the screen. The canvas is where you will drag logic gates over.



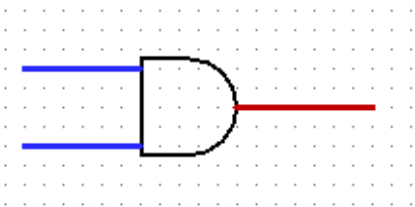
Toolbox shows the input/output pins available and logic gates available.

Selection: AND Gate	
Facing	East
Data Bits	1
Gate Size	Medium
Number Of Inputs	5
Output Value	0/1
Label	
Label Font	SansSerif Plain 12
Negate 1 (Top)	No
Negate 2	No
Negate 3	No
Negate 4	No
Negate 5 (Bottom)	No

The options box (below the toolbox when a gate is selected) has a number of options that affect the behavior of the logic gate you're drawing on the canvas.

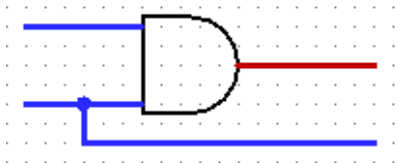


This is an AND gate dragged from the toolbox into the canvas. Using the options section, notice how there are now only two input wires, rather than the default five.



To connect "wires" to the input and output pins of the logic gates, simply click the pin and drag out a wire. You don't have to connect the wire to anything initially, but you can click and draw wires to any number of input pins.

To connect a wire to another wire, click on the wire to connect from and then drag out a wire. A bullet will be formed, informing you that the wires are bonded.



Notice the dot that "bonds" two wires together. You are not limited to bonding only two wires. Theoretically, an unlimited number of wires can be bonded together.

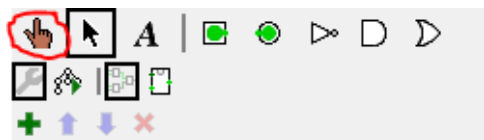
## [Step 2: Building Elementary Logic]

1. Using logisim's toolbox, options box, and canvas, develop a logic circuit that mimics the picture below:



The circuit above uses: (2) input pins, (1) inverter, (1) AND gate, (1) XOR gate, and (1) output pin.

2. Using the "finger" tool, flip the input pins from 0 to 1 and see what the output pin changes to with different selections of the input pins:



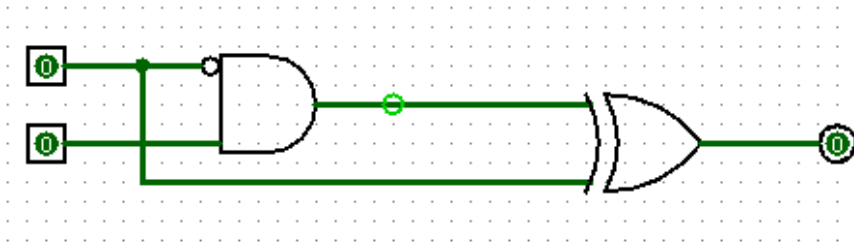
The finger tool toggles the input pins:



While this circuit has an explicit inverter, you can create inverter "bubbles" by using the options box:

Selection: AND Gate	
Facing	East
Data Bits	1
Gate Size	Medium
Number Of Inputs	2
Output Value	0/1
Label	
Label Font	SansSerif Plain 12
Negate 1 (Top)	Yes
Negate 2 (Bottom)	No

Which will then create an AND gate with a NOT bubble on the top input:



Notice that there are no options for an inverter for the output of the AND gate. Instead, you have to use the explicitly inverted gate, such as the NAND gate.

### [Step 3: Building Block Logic]

1. Develop a circuit that implements the following truth table. Use the following logic gates: (2) input pins (A and B), (1) XOR gate, (1) AND gate, (1) inverter (or inverter bubble), (2) output pins (Q and S).

A (i)	B (i)	Q (o)	S (o)
0	0	1	0
0	1	0	0
1	0	0	0
1	1	1	1

2. Test your circuit by using the "finger tool" to flip the input pins A and B. Make sure that Q and S represent the values given by the truth table.

3. In Logisim, click "Export Image..." under the "File" menu. Export your circuit as a PNG file, with a scale factor of 100% and "Printer View" checked.

### [Step 4: Binary Arithmetic]

1. Develop a calculator by writing the mul, add, sub, and two functions in the following Arduino code:

```
//COS130 Calculator Lab
const int SERIAL_BAUD = 115200;
```

```

int Mul(short left, short right);
int Add(int left, int right);
int Sub(int left, int right);
short Div(short dividend, short divisor, int &remainder);
int Two(int left);

void setup() {
  // put your setup code here, to run once:
  Serial.begin(SERIAL_BAUD);
}

void loop() {
  // put your main code here, to run repeatedly:
  int opl;
  char op;
  int opr;
  int result;
  int remainder;

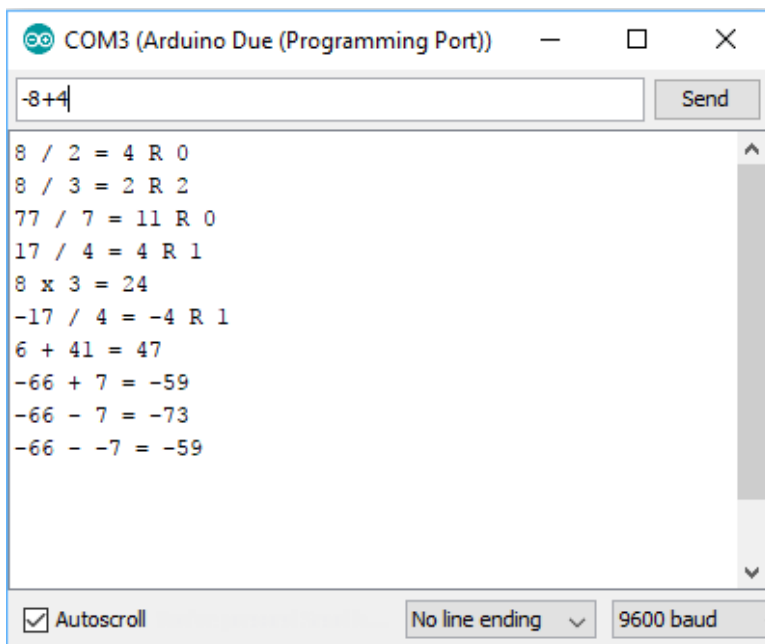
  if (Serial.available() > 0) {
    String input = Serial.readString();
    sscanf(input.c_str(), "%d %c %d", &opl, &op, &opr);
    switch (op) {
      case 'x':
        result = Mul(opl, opr);
        break;
      case '+':
        result = Add(opl, opr);
        break;
      case '-':
        result = Sub(opl, opr);
        break;
      case '/':
        Serial.print(opl);
        Serial.print(" ");
        Serial.print(op);
        Serial.print(" ");
        Serial.print(opr);
        Serial.print(" = ");
        Serial.print(Div(opl, opr, remainder));
        Serial.print(" R ");
        Serial.println(remainder);
        return;
      default:
        Serial.println("Usage: left op right\n");
        Serial.println("Where op is:");
        Serial.println(" + - Add\n - - Subtract\n x - Multiply\n / - Divide");
        return;
    }
    Serial.print(opl);
    Serial.print(" ");
    Serial.print(op);
    Serial.print(" ");
    Serial.print(opr);
    Serial.print(" = ");
  }
}

```

```
Serial.println(result);
}
}

//Write the functions Mul, Add, Sub, Div, and Two below
```

2. Your functions must **not** have any arithmetic operators: +, -, \*, /, %, or !. You may **ONLY** use logical operators: &, |, ^, ~, <<, >>.
3. **Add** will take two parameters (left and right) and return them added together. For example, Add(17, 9) will return 26. You may only use logical operators. Furthermore, **you may not use any if statements** in this function!
4. **Two** will return the twos complement of the parameter. You may use only logical operators here! If you need to add, use your add function.
5. **Sub** will take two parameters (left and right) and return left - right. This function will simply call your add and two functions. Remember that subtracting is the same as adding the twos complement.
6. **Mul** will take two parameters (left and right) and return left \* right. You may not use any arithmetic operators. You may only use if statements to determine if the result is negative. The rest of the code must only use logical operators.
7. **Div** will take three parameters (dividend, divisor, and a reference remainder) and return dividend / divisor and remainder will = dividend % divisor. You may not use any arithmetic operators (use your functions as necessary). You may use any logic and / or bitwise operations. Finally, you must use the restoring division algorithm covered in class.
8. Test your program by using the "serial monitor".



```
COM3 (Arduino Due (Programming Port))
-8+4 Send
8 / 2 = 4 R 0
8 / 3 = 2 R 2
77 / 7 = 11 R 0
17 / 4 = 4 R 1
8 x 3 = 24
-17 / 4 = -4 R 1
6 + 41 = 47
-66 + 7 = -59
-66 - 7 = -73
-66 - -7 = -59
Autoscroll No line ending 9600 baud
```

[You are finished with this lab!]

Upload your PNG file directly. You will need to rename your .ino file to a .txt file and upload it. For example, mylab.ino will need to be uploaded as mylab.txt. The reason is that Canvas doesn't recognize .ino files, so it makes it difficult for the TAs to grade.

Points 100

Submitting a file upload

File Types png and txt

<b>Due</b>	<b>For</b>	<b>Available from</b>	<b>Until</b>
Sep 12, 2017	Everyone	Sep 8, 2017 at 7:58am	Sep 12, 2017 at 11:59pm

### Lab 3

<b>Criteria</b>	<b>Ratings</b>					<b>Pts</b>
NOT Gate Properly Wired	<b>5.0 pts Full Marks</b>		<b>0.0 pts No Marks</b>			5.0 pts
A=0,B=0 Gives Proper Result	<b>5.0 pts Full Marks</b>		<b>0.0 pts No Marks</b>			5.0 pts
A=0,B=1 Gives Proper Result	<b>5.0 pts Full Marks</b>		<b>0.0 pts No Marks</b>			5.0 pts
A=1,B=0 Gives Proper Result	<b>5.0 pts Full Marks</b>		<b>0.0 pts No Marks</b>			5.0 pts
A=1,B=1 Gives Proper Result	<b>5.0 pts Full Marks</b>		<b>0.0 pts No Marks</b>			5.0 pts
Input Pins Properly Connected	<b>5.0 pts Both</b>	<b>2.5 pts One</b>		<b>0.0 pts No Marks</b>		5.0 pts
Output Pins Properly Connected	<b>5.0 pts Both</b>	<b>2.5 pts One</b>		<b>0.0 pts No Marks</b>		5.0 pts
Add function -No arithmetic operations -No if statements - Logical operations only	<b>20.0 pts Full Marks</b>	<b>17.0 pts Mostly Correct</b>	<b>11.0 pts Somewhat Correct</b>	<b>5.0 pts Marginally Correct</b>	<b>0.0 pts No Marks</b>	20.0 pts
Sub function -No arithmetic operations -No if statements - Logical operations only -Must call Add and Two	<b>10.0 pts Full Marks</b>	<b>6.0 pts Mostly Correct</b>	<b>4.0 pts Somewhat Correct</b>	<b>2.0 pts Marginally Correct</b>	<b>0.0 pts No Marks</b>	10.0 pts
Two function -No arithmetic operations -No if statements - Logical operations only	<b>10.0 pts Full Marks</b>	<b>6.0 pts Mostly Correct</b>	<b>4.0 pts Somewhat Correct</b>	<b>2.0 pts Marginally Correct</b>	<b>0.0 pts No Marks</b>	10.0 pts
Mul function -No arithmetic operations -If statements only for determining negative operands -Logical operations only	<b>20.0 pts Full Marks</b>	<b>17.0 pts Mostly Correct</b>	<b>11.0 pts Somewhat Correct</b>	<b>4.0 pts Marginally Correct</b>	<b>0.0 pts No Marks</b>	20.0 pts
Div function -No arithmetic operations -Logical operations only -Must use restoring division algorithm on lecture slides (same as in your textbook).	<b>15.0 pts Full Marks</b>	<b>11.0 pts Mostly Correct</b>		<b>6.0 pts Somewhat Correct</b>	<b>0.0 pts No Marks</b>	15.0 pts



<b>Criteria</b>	<b>Ratings</b>	<b>Pts</b>
Total Points: 110.0		