

COMP2611: Computer Organization

Introduction to Logisim & simple combinational circuit

❑ Today we will learn:

How to download and run Logisim;

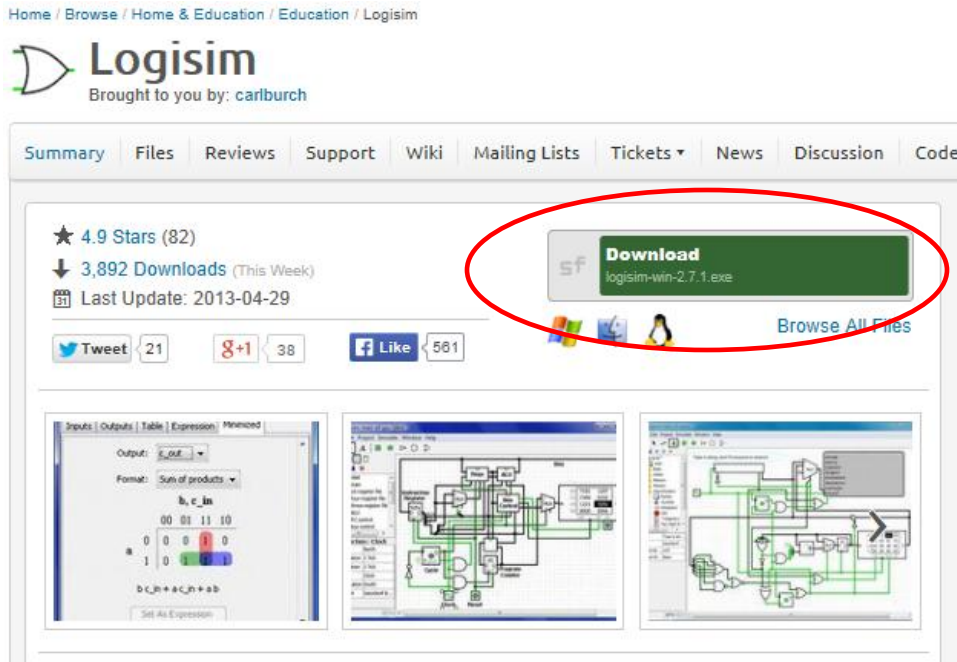
How to use basic Logisim operations;

How to build a simple combinational circuit in Logisim;

How to store a built-circuit as a library for future uses.

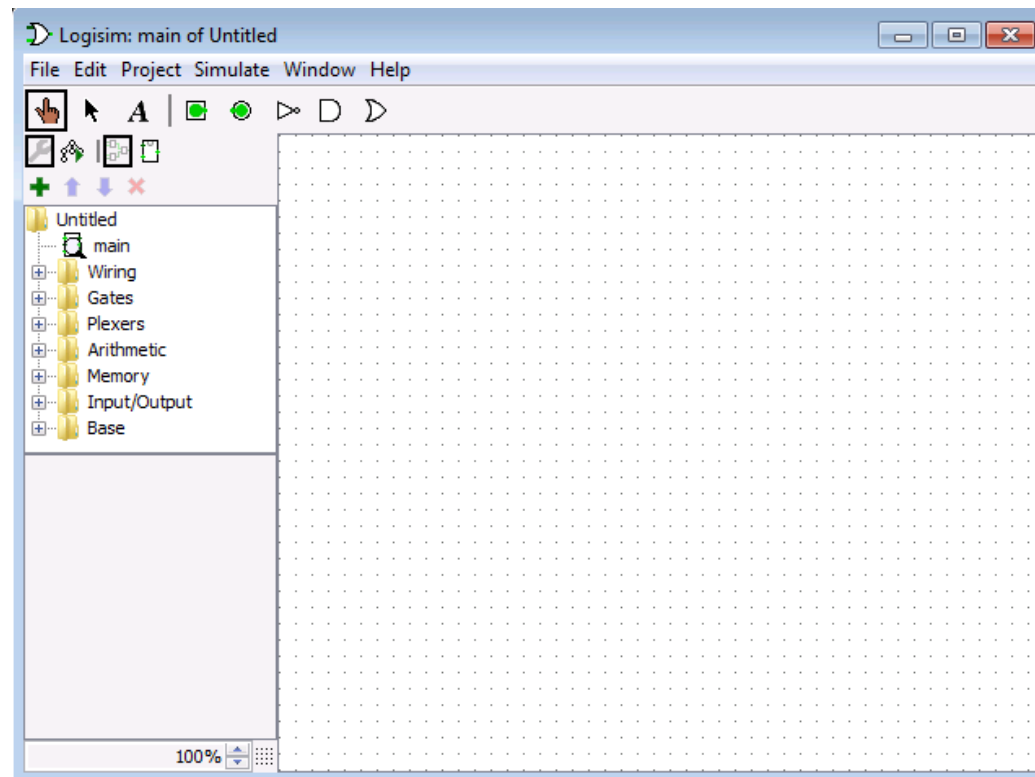
- ❑ Logisim is an interactive graphical interface for
 - ❑ designing,
 - ❑ simulating digital logic circuits.
- ❑ It is developed by Carl Burch of the Hendrix college, USA, in 2001.
- ❑ It runs on any platform that has **Java runtime** installed (version 5 or above required).
- ❑ We will illustrate in this lab how you can download the software and use it to design and test simple combinational logic circuits.

- ❑ To download Logisim you can go to the following link :
<http://sourceforge.net/projects/circuit/>
- ❑ Then you will see the following:

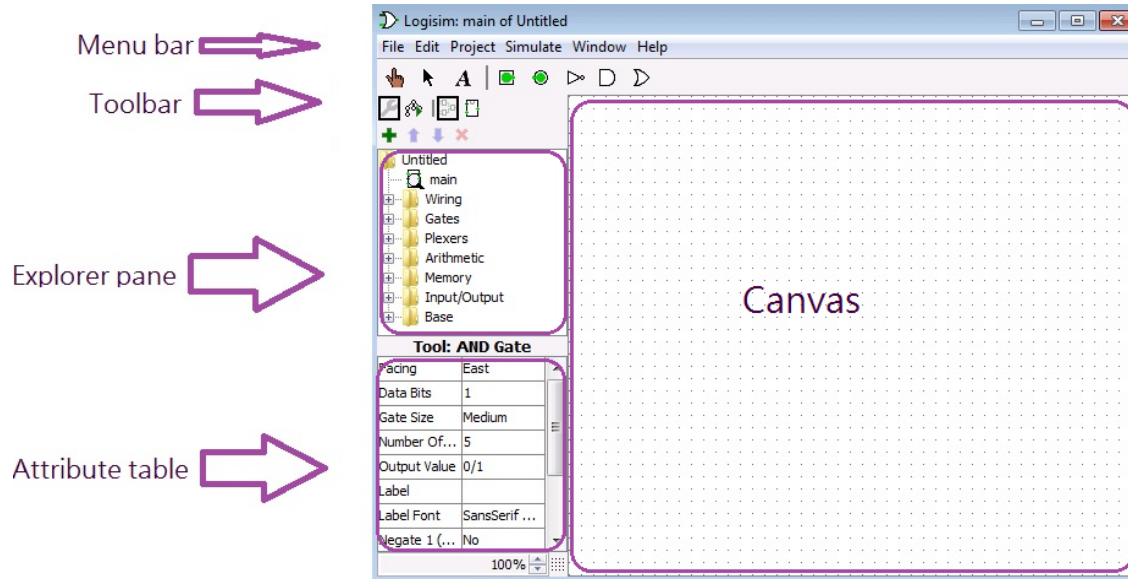


- ❑ Click on the green download button and follow the instructions to download.

- ❑ Now you have downloaded the executable file of Logisim (windows) : [logisim-win-x-y-z.exe](#)
- ❑ Logisim does not need installation, just double click on the downloaded file and you will be running Logisim!

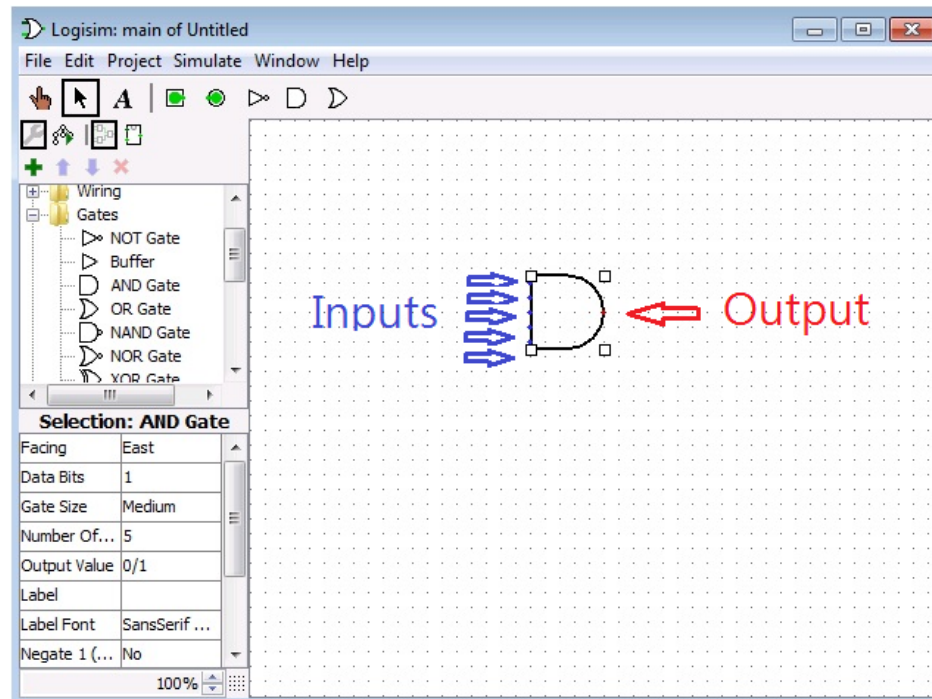


- ❑ The different components of the Logisim interface are shown below:



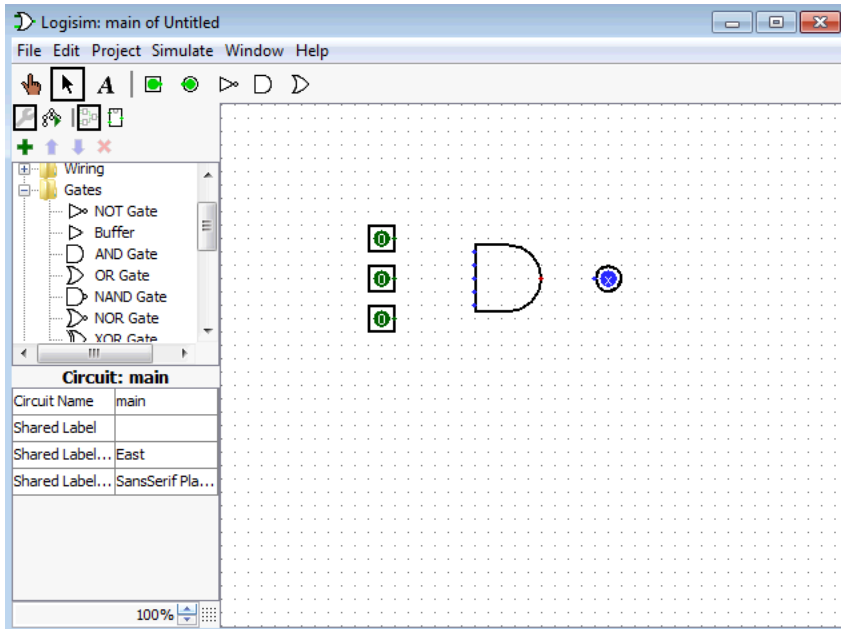
- ❑ The **Menu bar** is for accessing the different Logisim functions.
- ❑ The **Toolbar** shows icons of some of the more frequently used tools/modules.
- ❑ The **Explorer pane** is for maneuvering and accessing the available Logic components/circuits.
- ❑ The **Attribute table** shows the properties of a selected component
- ❑ The **Canvas** is the workspace for designing circuits.

- ❑ Let's test an simple logic component - the AND gate on Logisim.
- ❑ To incorporate an AND gate to the Logisim canvas, select "AND gate" from the "Gates" folder in the "Explorer pane", then move the mouse cursor to the canvas, click the left mouse button to fix the position of the gate.



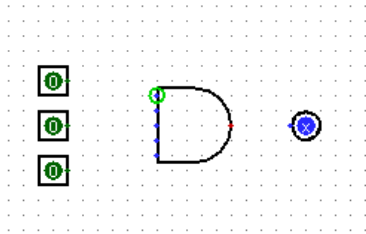
- ❑ Note the five small blue dots and the single red dot. These are the spots where input or output wires can be attached.

- ❑ Add three inputs and one output to the AND gate (assume this AND gate accepts three inputs).
- ❑ The inputs are added by selecting the **input pin** (solid circle in a square) from the toolbar, and then dragging it to the canvas
- ❑ The output is added in a similar manner, except that you have to select the **output pin** (solid circle in a circle) from the toolbar.

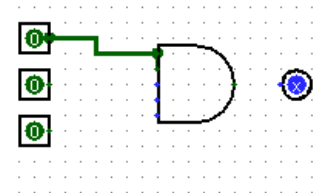
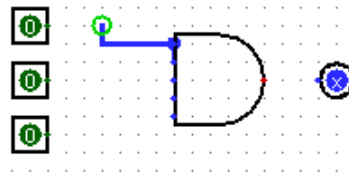
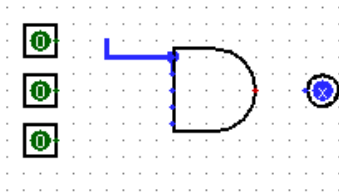


- ❑ If you want to move any component on the canvas to another location, click the arrow icon on the toolbar, select the component you want to move, and then move it.

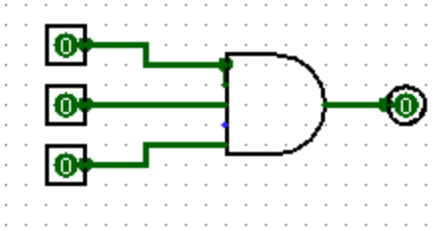
- ❑ Now we have added the input and output pins, but we have not yet connected them to the AND gate.
- ❑ To connect the inputs to the AND gate, first click the arrow icon on the toolbar and then point the mouse cursor to the possible input spots of the gate, a small green circle will appear:



- ❑ Now click the left mouse button continuously and you will be able to draw a wire from the input spot.
- ❑ Whenever necessary, you can draw the wire for multiple times as shown below:



- ❑ Connect the remaining wires as follows:

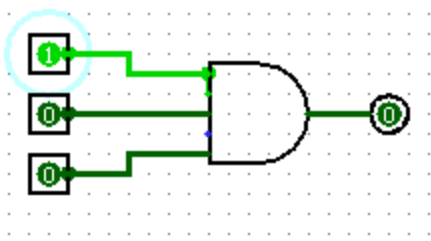


- ❑ Note once you have completely connected the input/output wires, the output changes from the blue "x" to "0". This indicates the AND gate has already computed a zero for the zero inputs.
- ❑ To change the input values, select the poke tool from the toolbar (make sure "**simulation enabled**" is **ticked** under the "**simulate**" option of the menu bar)

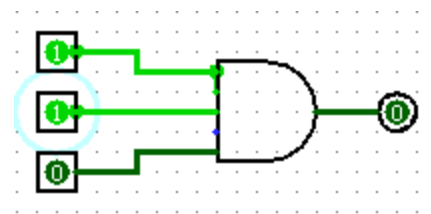


- ❑ Move the cursor to the input pin and change it by clicking the left mouse button at the pin.

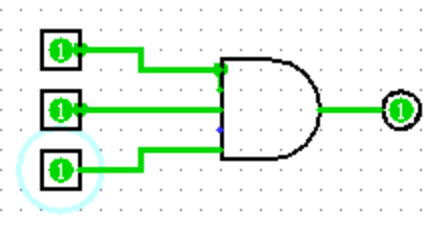
- ❑ After poking the first input, it changes from "0" to "1", the output from the AND gate is now "0".



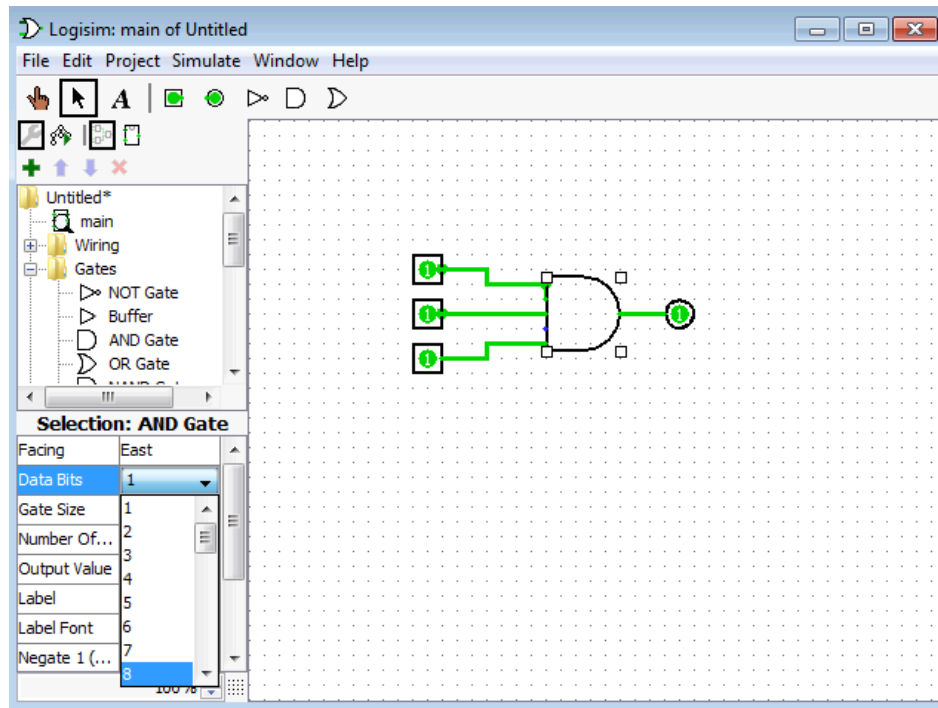
- ❑ The second input also changes from "0" to "1" after being poked, the output from the AND gate is still "0", which is to be expected.



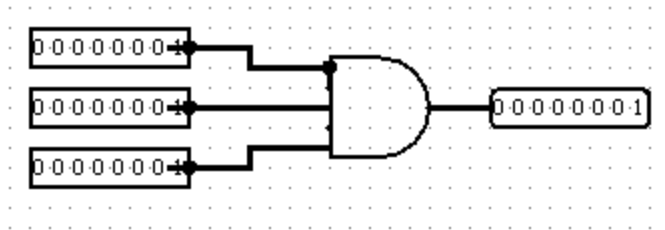
- ❑ Now the third input has been changed to "1" and note that the output becomes "1".



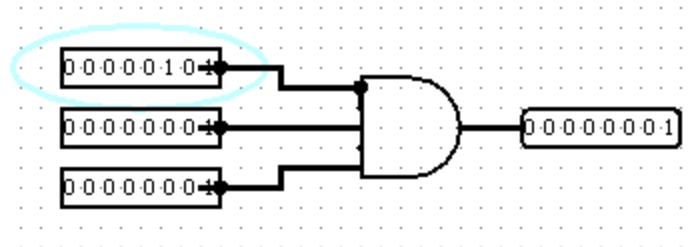
- ❑ Each input pin of the previous AND gate is 1-bit wide. To modify the width of the data, one can use the **attribute table**.
- ❑ To increase the data width of the AND gate, first point the cursor to the gate in the canvas to select the gate.
- ❑ Then find the item "Data Bits" in the **attribute table**, and adjust it as wishes.



- ❑ Continue to adjust the data widths for the remaining components (input/output), until all the wires are in black and none of the wires is orange in color.
- ❑ You don't have to adjust the data widths for the wires as they will adjust automatically to match the components.



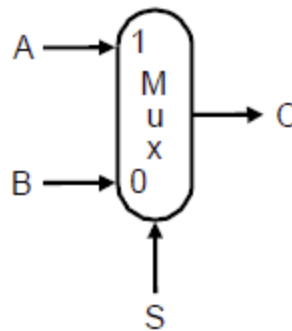
- ❑ Now the three 8-bit inputs are 0000 0001 and the output is 0000 0001, a **bitwise** AND operation has been performed on the three inputs.
- ❑ You can change the value of the individual bits by poking them just like before:



- ❑ At this point can you
 - ❑ drag logic components onto the canvas of Logisim?
 - ❑ identify the input/output spot of a component once it is on the canvas?
 - ❑ draw input/output wires to/from the component and connect them correctly?
 - ❑ poke to change the input values?
 - ❑ specify data widths of the components?

- ❑ If there is any question, it is good to be sorted out at this point, before we go on to discuss more complicated operations in Logisim.

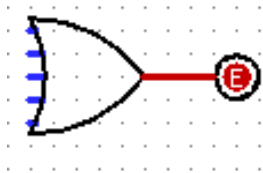
- ❑ Now you already know the basics of Logisim, so we can build a circuit using it.
- ❑ We are going to build a 1-bit (2-input) multiplexer on Logisim.
- ❑ The following is the figure of a 1-bit (2-input) multiplexer, **A** and **B** are two inputs, **S** is the control input that selects data either from A or B to be forwarded to the output **C**.



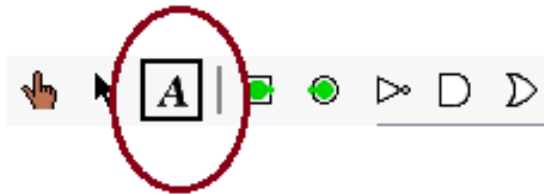
- ❑ The Boolean expression for C in terms of A, B and S is:

$$C = A \cdot S + B \cdot (\sim S), \text{ where } \sim S \text{ is the negation of } S$$

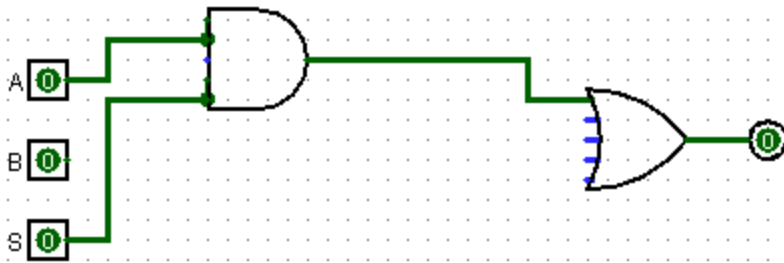
- ❑ To build such a multiplexer, first we need to draw an OR gate in Logisim, and connect it with an output



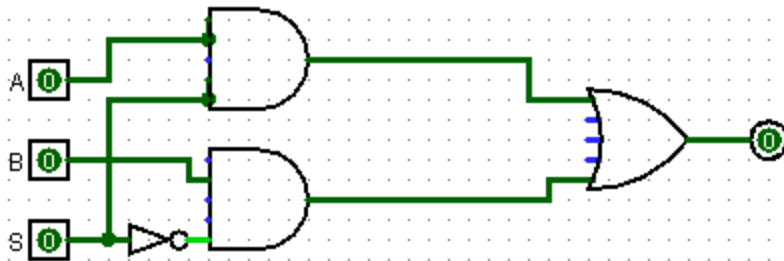
- ❑ Then we need to build the part that does $A \cdot S$, we need to give the names A, B, S for the inputs/control, to do that just click on the “**text tool**” in the toolbar, and then click the component to enter the desired name.



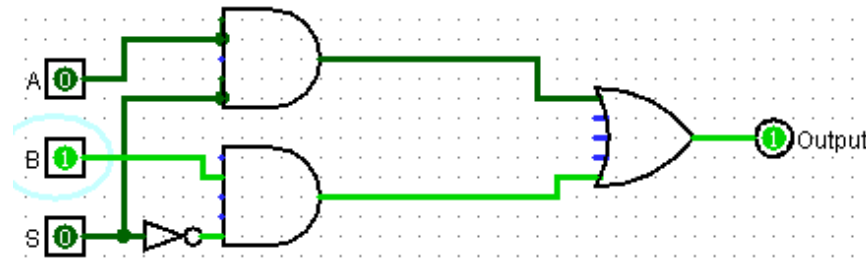
- Now connect A and S to an AND gate, and connect the output of the AND gate to the OR gate.



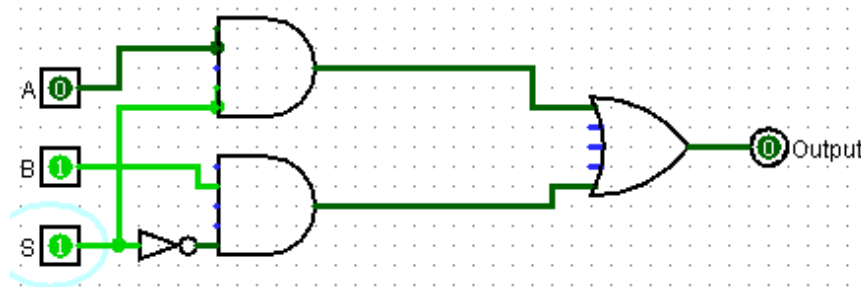
- Do the same thing for B and ($\sim S$) with the help of a NOT gate, we have the completed multiplexer.



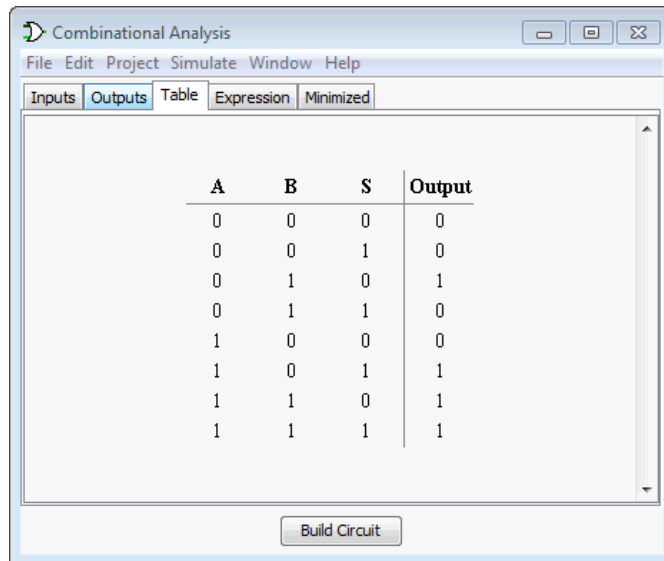
- ❑ Let's poke the circuit to see if it works as expected. When $S=0$, the value of B is forwarded to the output.



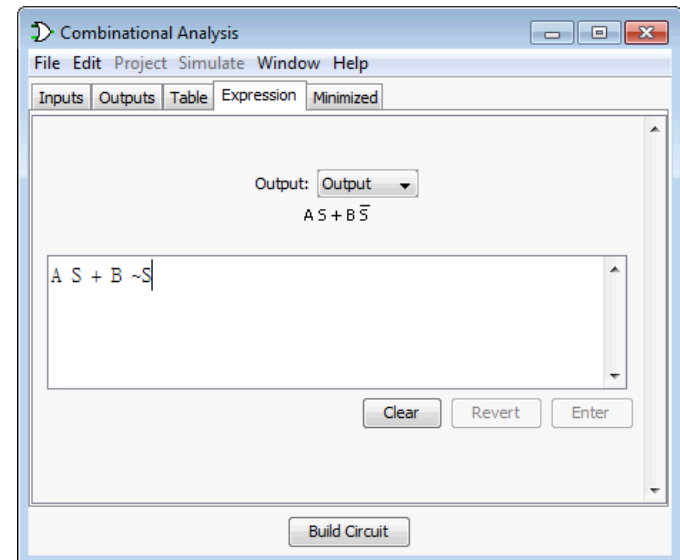
- ❑ When $S=1$, the value of A is forwarded to the output.



- By clicking “Project”, “Analyze circuit”, “Table”, we can see the input/output relationship. And if you click “Expression” instead of “Table”, you will see the corresponding Boolean expression of the circuit.



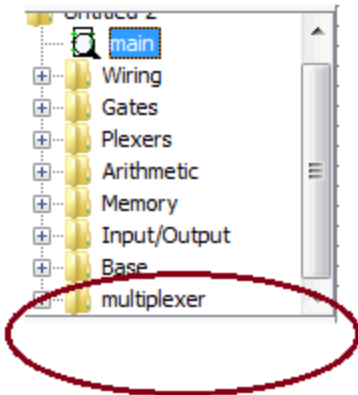
A	B	S	Output
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



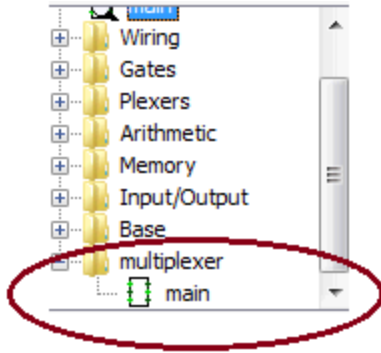
Output:

$A S + B \bar{S}$

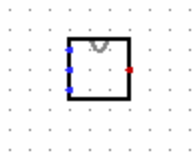
- ❑ From time to time, you may want to store your work for future use. To do that, save your work as a **.circ** file under Logisim (i.e. multiplexer.circ).
- ❑ To load the work for use, click "**Project**", "**Load Library**", "**Logisim Library**" and choose the .circ file you saved.
- ❑ The circuit will appear in the explorer pane. One just need to click the folder to access the real circuit.



- ❑ To use the circuit, just drag the circuit from the explorer pane to canvas.



- ❑ The circuit will be characterized by the inputs and the output only (i.e. like a black-box). You can use it to build more complicated circuits.



❑ Today we have learnt:

How to download and run Logisim;

How to use basic Logisim operations;

How to build a simple combinational circuit in Logisim;

How to store a built-circuit as a library for future uses.