

COMP2611: Computer Organization

Building Sequential Logics with Logisim

Overview

- You will learn the following in this lab:
 - ▣ Logisim auto-build feature,
 - ▣ building a SR latch on Logisim,
 - ▣ building a D latch on Logisim,
 - ▣ building a D flip-flop from D latches.

Quick review of Logisim operations

- If you have problem with any of the following operations, do let me know so that I can show you again.
 - ▣ select and drag a logic component onto the canvas of Logisim,
 - ▣ draw input/output wires to/from the component and connect them correctly,
 - ▣ change the data width of a component,
 - ▣ poke to change the input values.
- **NOTE: 1) To make it easier to draw the graphic and connect the wire accurately, Magnify the graph to 150-200% (change at the lower left corner of the %)**
- **2) Note the color of the wire, in running, light green (1 state), dark green (0 state). Any other color (orange, red, blue) means connection problem or other error.**

Auto build feature of Logisim

- Just for your interest, Logisim can auto-build a circuit according to the logic expression supplied by you.
- For example, to implement a circuit for the following truth table:

| Inputs | | | Output |
|--------|---|---|--------|
| A | B | C | D |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

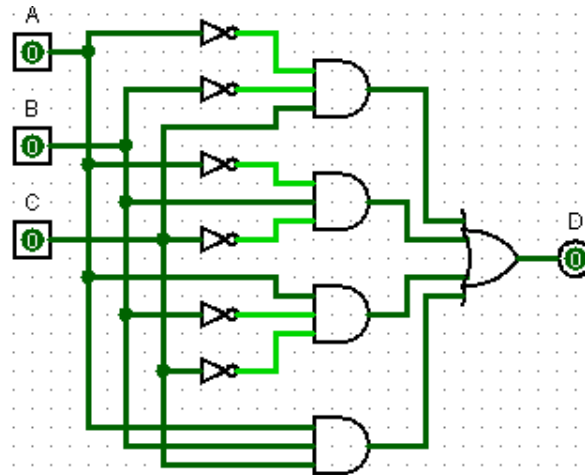
- You just need to find the sum-of-product representation for the inputs and the output:

$$D = (\sim A \cdot \sim B \cdot C) + (\sim A \cdot B \cdot \sim C) + (A \cdot \sim B \cdot \sim C) + (A \cdot B \cdot C)$$

- Enter the expression to Logisim and let Logisim build the circuit automatically.

Auto build feature of Logisim

- To do that, click “Project”, “Analyze Circuit”,
- then click the “Inputs” tab, enter “A”, “B”, “C” separately as the inputs,
- now click the “Outputs” tab, enter “D” as the output,
- click the “Expression” tab and type the expression as follows (replace every AND operator with a space):
$$(\sim A \sim B C) + (\sim A B \sim C) + (A \sim B \sim C) + (A B C)$$
- now click “Enter” tab at the lower right corner, finally click “Build Circuit” at the bottom, Logisim will build the circuit. You can poke to check its correctness.



Quick sequential logics review

- Do Let me know if you have problem to recall the following terms.
 - ▣ A clock,
 - ▣ Clock cycles,
 - ▣ Rising/falling edge of a clock cycle,
 - ▣ Edge triggered clocking,
 - ▣ Clocked/unclocked elements,
 - ▣ State elements.

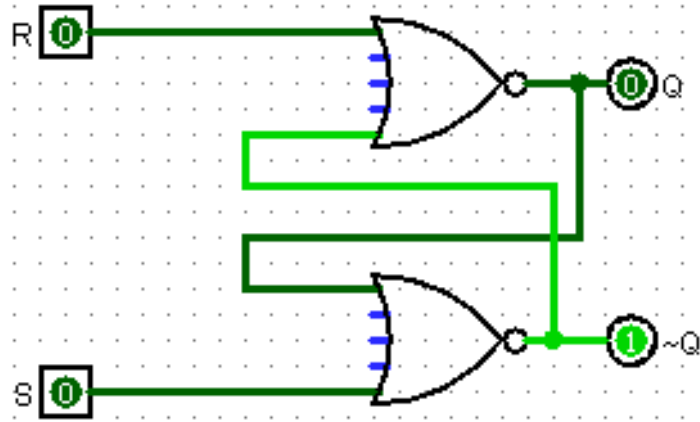
The S-R Latch

- A Set-Reset latch is the simplest form of storage element. It is an unclocked element (what does that mean?).
- Two inputs:
 - - **S** for storing the value “**1**” to the latch
 - - **R** for storing the value “**0**” to the latch
- Two outputs
 - - **Q** for outputting the value stored in the latch
 - - $\sim Q$ the complement of Q.
- The following table details the values of the two inputs and the effects on the output value Q.

| S | R | Action on Q |
|----------|----------|-----------------|
| 0 | 0 | Nothing changed |
| 0 | 1 | Q=0 |
| 1 | 0 | Q=1 |
| 1 | 1 | forbidden |

The S-R Latch

- Start Logisim and build the S-R latch as shown below. (make sure you label the inputs and the outputs)

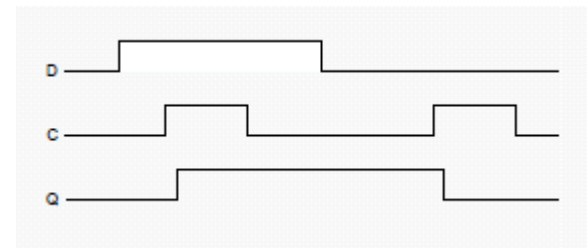


- Poke the inputs and verify your circuit is correct by referring to the table on the previous slide. Save the latch as “sr-latch.circ” for future use.

The D-Latch

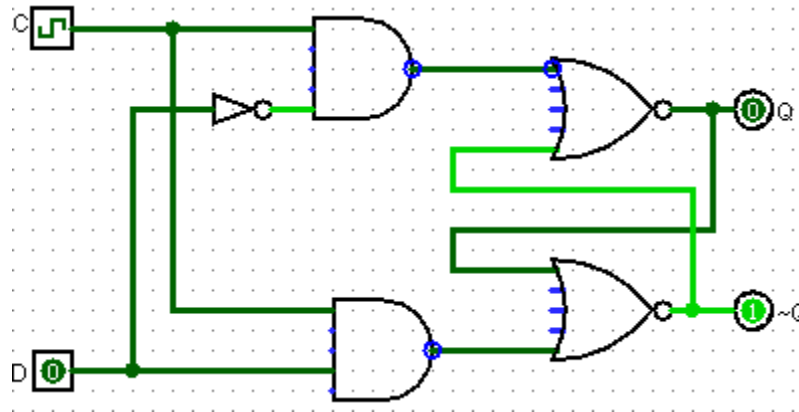
- A D-latch is a clocked storage element.
- The value stored in a D-latch can be updated iff the clock is asserted (i.e. Hi).
- Two inputs
 - - **D**, the value to be stored
 - - a **clock signal** (that alternates between Hi and Lo regularly)
- Two outputs
 - - **Q** for outputting the value stored in the latch
 - - **$\sim Q$** the complement of Q.
- The following table details the values of the two inputs and the effects on the output value Q.

| C (Clock) | D | Action on Q |
|-----------|---|-----------------|
| 0 | 0 | Nothing changed |
| 0 | 1 | Nothing changed |
| 1 | 0 | Q=0 |
| 1 | 1 | Q=1 |

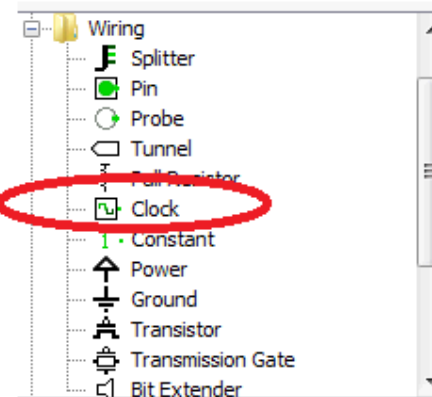


The D-Latch

- Build the D-latch as shown below. (make sure you label the inputs and the outputs)



- You can find the clock element in the wiring folder of the “Explorer pane”



The D-Latch

- Click “Simulate”, “Tick Frequency”, “1 Hz” so that the clock will alternate at the frequency of 1Hz.
- Now click “Show simulation Hierachy” button on the toolbar, and then click the “Enable clock ticks” button to start the clock.

Show simulation hierarchy



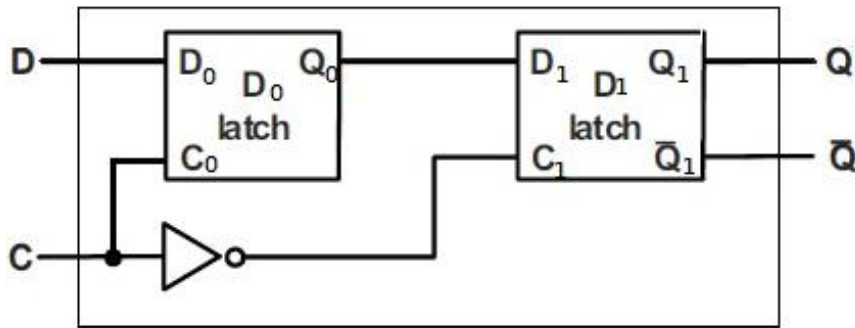
Enable clock ticks

Stop clock ticks

- Once the clock is running you can poke the input **D** to store value to the latch. Take some time to observe the way the input is being stored.
- Store the latch as “d-latch.circ”.

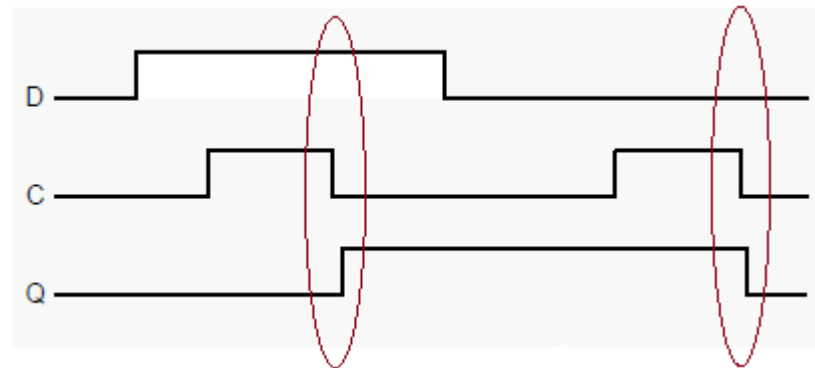
The D Flip-Flop

- A D flip-flop is an edge triggered device. Its value can only be changed on a clock edge (a much shorter duration than that of the D latch).
- The D flip-flop can be made from two D latches:



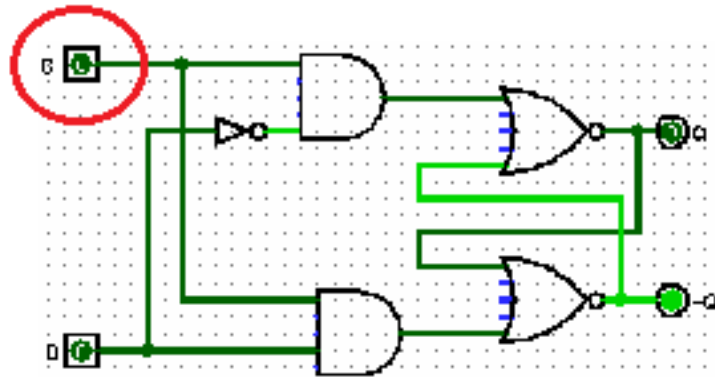
- The timing diagram for the D flip-flop is shown below, note that changes of the flip-flop value occur only at the falling edge of the clock (circled by the two red ovals)

| C (Clock) | D | Action on Q |
|------------------|-------------------|-----------------|
| Falling edge | 0 | 0 |
| Falling edge | 1 | 1 |
| Non-falling edge | X (don't care) | Nothing changed |

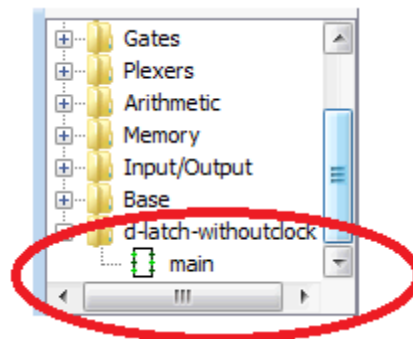


The D Flip-Flop

- Load “d-latch.circ” and replace the clock with an input named “C”. Store it as “d-latch-withoutclock.circ”. (by deleting the clk from the D-latch, and replacing it by C input)

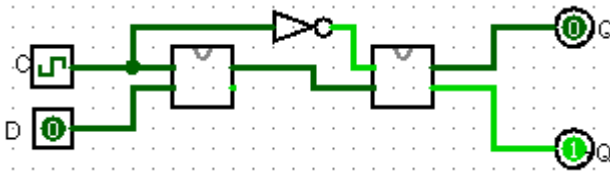


- Start a new project (by pressing “Ctrl+N”) in Logisim. Load the D latch built by clicking “Project”, “Load Library”, “Logisim library...” and select “d-latch-withoutclock.circ”. Now you should be able to find the d-latch in the “Explore pane”.



The D Flip-Flop

- Click on “main” and drag the D latch to the canvas to draw the D Flip flop.
- Mind that locations of the C and D on the D latch (i.e. they are different from that of slide 12).
- Your finished flip-flop should be similar to the following:



- Enable the clock as discussed on slide 11, and poke the input “D” to see the changes of the output at Q and ~Q.
- Save the flip-flop as “D-flip-flop.circ”.
- Now remove the clock from the D flip-flop and replace it by an input “C” just like before. Save this version as “D-flip-flop-withoutclock.circ”. We will use it to build a register.

Building a 2-bit register with the D Flip-Flops

- We can build a simple register that holds 2 bits with the D flip-flop (you can use the D flip-flop to build register of any size, we build a 2-bit register here for simplicity).
- Basically, the idea is to use two D flip-flops in the circuit, each of which holds a single bit.
- The input will be 2-bit in width. We need to separate the individual bits by using a “splitter” and direct the bits to the corresponding D flip-flops.
- It will be covered in the next lab.

Conclusion

- We covered the following topics today:
 - ▣ Logisim auto-build feature,
 - ▣ building a SR latch on Logisim,
 - ▣ building a D latch on Logisim,
 - ▣ building a D flip-flop from D latches.