Faculty of Science, Engineering and Technology

counters



Computer Systems

Week 3

# Overview

In this laboratory session we start using Flip Flops to build useful things like registers, and shift registers.

**Purpose:** To consolidate your knowledge of Flip Flops, and how they can be used.

**Task:**

**Time:** This lab is due by the start of your week 3 lab.

**Assessment:** This lab is worth 1% (up to a maximum of 5%) of your assessment for this unit, and only if demonstrated to your lab demonstrator in the week it is due.

**Resources:** ■ Flip Flop tutorials

* Intro to Flip Flops

***Submission Details***

You must submit the following files to Canvas:

* A document containing all required work as described below.



# Instructions

1. Start Logisim and open a new canvas

## Part 1: Storing bits with Flip Flops

Any computing hardware that seeks to perform meaningful calculations using bits requires cir- cuitry to store them - that is, circuits that can maintain a given state. In lectures we discussed Flip Flops, which are simple block circuits designed to maintain a particular binary state, and transition between binary states depending on the inputs given.

1. Review this week’s lecture slides, and if needed, also take a look at the quick video tutori- als linked under resources at the beginning of this lab sheet.
2. Create a clear canvas.
3. Using the lecture slides as a guide, wire up your own **R-S Flip Flop using a pair of 2-input NOR gates** (**do not use Logisim’s S-R Flip-Flop!**). You should have 2 input pins, one for the “*Set*” pin, and one for the “*Reset*”, and two output LEDs: *Q* and *Q’* .
4. When you’ve finished wiring it up, set both input pins to 1. The LEDs should both be dark (assuming you’ve wired it correctly).

A diagram of a set

Description automatically generated

## Export your circuit as an image and include it in your submission document.

1. Set the pins *in the following order* and record the states for Q and Q’

|  |  |  |  |
| --- | --- | --- | --- |
| **Set Reset Q Q’** | | | |
| **1** | **0** | 0 | 1 |
| **1** | 1 | 0 | 0 |
| **0** | 1 | 1 | 0 |
| **1** | 1 | 0 | 0 |

1. Describe in a sentence, the behaviour of the circuit when one of the inputs is 1 (but not both) and why this is useful for digital circuit design.

In my opinion, if SET is 0 and RESET is 1, the output Q’ is 0 and the Q is 1 (0 NOR 0). This is useful for digital circuit design because it enables the creation of memory elements that can hold a state until it is changed by an external signal.

1. What do you notice about the two times you set both inputs to 1. Briefly explain what is happening here and why this is an issue for digital circuit design?

- From my perspective, the outputs Q and Q’ are designed to complement each other, which means that when Q is 1, Q’ is 0 and in contrast. Therefore, this is a fundamental property of binary logic, where a bit can only have one of two values: 0 and 1.

- If both outputs are 1, this can lead to incorrect results or unpredicted situations. Thus, it’s essential to ensure that digital circuits produce well-defined outputs.

## Discuss 7 and 8 with your lab demonstrator and provide your answer in your submis- sion document, along with the truth table in Step 6.

1. So the unclocked R-S flip flop has issues. Lets talk about the D Flip-Flop then. Review the lectures on the D Flip-Flop, and when you feel comfortable, wire up a D Flip Flip using AND gates and NOR gates, with output LEDS labeled Q and Q’.
   * For this you will have only 1 input pin, as well as a clock input. The clock can be pulsed on and off by clicking it with the operation pointer (the finger in the top left of screen), or you can simple enable clock ticking from the menu (under “Simulate”).

## Export your circuit as an image and include it in your submission document.

1. Explore the behaviour of the D Flip Flop by filling out the following truth table

|  |  |  |  |
| --- | --- | --- | --- |
| **Clock Pin Q Q’** | | | |
| **0** | **0** | 0 | 1 |
| **0** | **1** | 0 | 1 |
| **1** | **1** | 1 | 0 |
| **1** | **0** | 0 | 1 |

1. Briefly explain the behaviour of a D Flip Flop and how it is useful for digital circuit design.

- In my view of point, D Flip Flop has two inputs: D(Data) and Clock, and two outputs (Q and Q’). When the clock signal is low, the flip flop holds its current state and ignores the D input. When the clock is high, the flip flop samples and stores the D input. The value that was previously fed into the D input is reflected at the flip flop’s Q output. The Q’ output of the flip flop is complemented by the Q output.

- The D Flip Flip is useful for digital circuit design because it can be used to create memory circuts for holding data, registers for storing data, couters for counting events, and synchronous systems.

1. What is the role of the clock ? How does it impact the changing of state of Q and Q’ ?

The clock signal plays a crucial role in the operation of a D Flip Flop. It synchronizes the changing of state of the outputs Q and Q’. Here’s how the clock signal impacts:

- Edge-triggered behavior: The output (Q and Q’) changes state only on the rising or falling edge of the clock signal. This means that the output (Q and Q’) changes state only when the clock signal transitions from low to high (rising edge) or high to low (falling edge)

- Sampling and storing data: When the clock signal is high, the D Flip Flop samples the input data (D) and stores it. The value of D is then reflected at the Q output, and its complement is reflected at the Q’ output.

- Holding the current state: When the clock signal is low, the D Flip Flop holds its current state and ignores the D input. This means that the output (Q and Q’) remains unchanged, even if the input data (D) changes.

1. Why is it generally preferred over the R-S Flip Flop?

In digital electronics, flip-flops are basic memory elements used to store binary data. While both types of flip-flops have their own benefits and drawbacks, D flip-flop is generally preferred over the RS Flip Flop due to its simplified input logic, reduced complexity, improved reliability, and easier synchronization.

## Discuss 11 -13 with your lab demonstrator and provide your answer in your submission document, along with the truth table above.

1. J-K Flip Flops are like your general purpose Flip Flop because they are programmable. Review the video on JK Flip Flops, and when you’re feeling comfortable, wire up a J-K FF using NAND gates. Two of your NAND gates will need to deal with three inputs.

## Logisim will not be able to simulate this circuit but export your completed circuit as an image and include it in your submission document.

## A diagram of a circuit Description automatically generated

1. **Complete and include this truth table for JK Flip Flops in your submission document.**

|  |  |  |  |
| --- | --- | --- | --- |
| **J K Q (when clocked) Q’ (when clocked)** | | | |
| **0** | **0** | 0  1 | 0  1 |
| **1** | **0** | 0  1 | 1  1 |
| **0** | **1** | 0  1 | 0  0 |
| **1** | **1** | 0  1 | 1  0 |

1. How can a J-K Flip Flop be made to behave like a D Flip Flop?

A J-K Flip Flop can be made to behave like a D Flip Flop by connecting the J and K inputs toghether with a NOT gate and typing them to the D input. This is because a D Flip Flop has a single input (D) that determines the output, whereas a J-K Flip Flop has two separate inputs (J and K) that control the output.

* Step 1: Connect J and K inputs toghether
* Step 2: Tie the connected J-K node to the D input.

A diagram of a circuit

Description automatically generated

1. How can a J-K Flip Flop be made to behave like a toggle (T Flip Flop)?

A J-K Flip Flop can be made to behave like a toggle (T Flip Flop) by connecting two inputs (J and K) toghether and typing them to T input. This configuration allows the J-K Flip Flop to toggle its output on each other pulse, like a T Flip Flop.

A diagram of a circuit

Description automatically generated

## Discuss these questions with your lab demonstrator and provide your answer in your submission document, along with the truth table in Step 15.

**Part 2 - Register this !**

1. Registers are just adjacent Flip-Flops that store collections of bits. You’re about to wire up a register in Logisim, but first review the lecture slides, and if needed, take a look at the resources above to remind yourself how Flip Flops work. We’re going to work with D Flip Flops here.
2. We’re not going to wire our own Flip Flops anymore. We’re going to use Logisism’s. Familiarise yourself with Logisim’s D Flip Flop. Bring one into your canvas, and connect up an input pin, and a clock, and connect an LED to the output “Q”. Have a play and verify it works as you expect (ask your lab demonstrator for assistance if needed).
3. Now wire-up a 4-bit big-endian register with D Flip Flops in Logisim. Do this by using 4 pins for each input, and connect 4 LEDS to the output.
4. When complete, demonstrate your register to your lab demonstrator by showing them different combinations of input bits, and how this changes the output when the clock pulses.

## Export your circuit as an image and include it in your submission document.

1. Use your register to fill out the following test schedule:

|  |  |  |
| --- | --- | --- |
| **Ox Input Binary Output Binary** | | |
| **0** | **0000** | 0000 |
| **1** | 0001 | 0001 |
| **2** | 0010 | 0010 |
| **3** | 0011 | 0011 |
| **5** | 0101 | 0101 |
| **A** | 1010 | 1010 |
| **B** | 1011 | 1011 |
| **C** | 1100 | 1100 |
| **D** | 1101 | 1101 |
| **E** | 1110 | 1110 |
| **F** | 1111 | 1111 |

**Complete this table and place a copy of it in your submission document**

**Provide your answers in your submission document**

**When complete:**

* + Submit your answers (screen shots, etc) in a single document using **Canvas**
  + Show your lab demonstrator your working circuits in class (you must do this to get the cred- it). Your lab demonstrator may request you to resubmit if issues exist.