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).

## 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.

-> From the above table, when there is 1 at any inputs, the Q and Q’ have the opposite value to each other, based on which input is acting. This is a usefulness for circuit design as circuit could be used for storing, maintaining and toggling states

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 ?

-> This is a problem when both inputs are set to 1. Due to its racing condition, it’s breaking rule by having both of inputs at 1, resulting in their complement which can lead to an undefined, unstable and unpredictable state.

## Discuss 7 and 8 with your lab demonstrator and provide your answer in your submission 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** | 1 | 0 |  |
| **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.

-> A D Flip-Flop’s behaviour: If clock signal transits from low to high state, the output Q is updated to the value. And if the state change from high to low, the output Q’ is the complement of the Q one. There are three main advantages using it: synchronizing data, storing and transferring clock-controlled data and providing edge-triggered behaviour.

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

-> Clock acts as a trigger which controls when the flip-flop can have a change its state, it also provides timing reference regularly as well. It has a crucial impact on the outputs: If it’s rising or falling, Flip-flop will sample and transfer the data to the output Q. Q’ get inverse value of Q. If clock is in nonactive edge , there is no change in the result of both outputs, regardless D input’s alter.

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

-> There are several reasons:

+ Simplified Design / Simplified Input

+ Less Prone To Error / Avoid Invalid States

+ Easier To Use

+ More Flexible

+ Clock-Controlled / Clock Synchronization

+ Wider Applicability

+ Reliable Data Storage

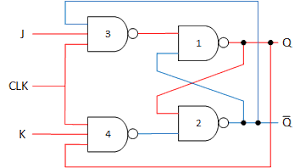
## Discuss 11 -13 with your lab demonstrator and provide your answer in your submission document, along with the truth table above.

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.

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** | Q | Q’ |
| **1** | **0** | 1 | 0 |
| **0** | **1** | 0 | 1 |
| **1** | **1** | Q (toggled) | Q’(toggled) |



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

- Connect J input to K input together

- Put a connection between above inputs and D inputs

Thus, when there is a signal from the clock, both J-K inputs and D input can set their values

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

-> there are 2 steps to make a J-K Flip-Flop to perform as a toggle (T Flip-Flop)

- Make a connection with J input and K input

- This combination input will act as a T input

-> When T=1, the Flip-Flop toggles

When T=0, it remains stable

## 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 dif ferent 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.