COMPUTER TECHNOLOGY
PRACTICAL SESSION P4 (LOGISIM)
SEQUENTIAL CIRCUITS: FLIP-FLOPs SIMULATION

INTRODUCTION

Flip-Flop and Clock signal in Logisim.

Figure 1 shows flip-flops D and JK in Logisim. Note that asynchronous inputs **Preset** and **Clear** are both active-high inputs. Also note that an active-high **Enable** input appears in both FFs (Enable should be 1).

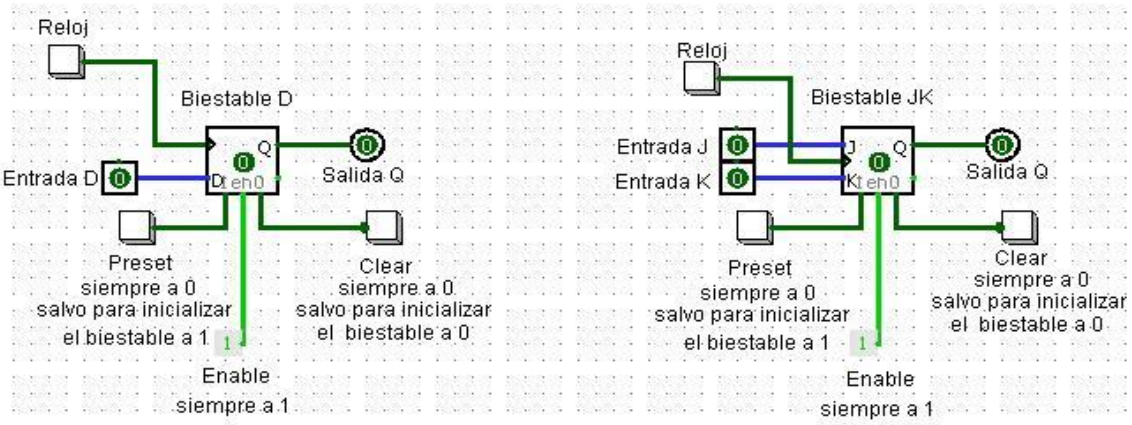

Figure 1

Important remarks:
- To get FFs you have to select ***Memory*** in the library pane.
- The flip-flops usually have one active-high **Enable** input, so you have to set Enable to 1.
- By default, FFs are positive-edge triggered ("Rising Edge"). Anyway, you can change the trigger as follows. Click in the FF in order to see the attributes panel, choose between *Rising Edge* or *Falling Edge* in the option Trigger.
- The clock signal can be generated by hand by using a single switch or a button. In Part 1, please use a ***Button*** (Button in the menu ***Input/Output)*** as a clock signal**.** By default, the initial value is 0. Value 1 is generated only at the moment the button is clicked.
- The clock signal can also be generated by means of the ***Regular Clock Signal*** (Clock). This is in the library pane as a part of the ***Wiring*** menu. In order to launch the simulation, click ***Simulate*** and activate *Simulation Enabled*. With the option *Tick Frequency* you can set the clock frequency. If *Ticks Enabled* is activated, the simulation is executed at the chosen frequency. To simulate cycle by cycle, disable *Ticks Enabled*. Now, each time you click option *Tick once* or click the clock icon, a half cycle is executed. This method will be used in Part 2 and 3.

COMPUTER TECHNOLOGY
PRACTICAL SESSION P4 (LOGISIM)
SEQUENTIAL CIRCUITS: FLIP-FLOPs SIMULATION

PART 1. INITIALIZATION AND PERFORMANCE OF FFs: D AND JK

Check the functionality of flip-flops JK and D with Logisim.

Flip flop D

INITIALIZATION

Design the circuit of Figure 1 left).

- Select ***Memory*** in the library pane and get a D FFs
- Connect D to a switch
- Connect the output (Q) to a LED.
- Connect the asynchronous input **Clear** to a ***Button*** (menu ***Input/Output).*** Do the same for **Preset.**
- Give a permanent value 1 to **Enable** (use ***1-Constante*** in menu ***Wiring*****)**
- Connect the **Clock** input to another ***Button***.

FUNCTIONALITY

Initialize to 0 the FF, by clicking the **Clear** button. After initializing the flip-flop to 0, the functionality will be checked as follows:

- Set D=1. Click the **Clock** button and check that the next state is 1. Push **Clock** again and again and check that the next state is always 1.
- Set D=0. Click the **Clock** button and check if the next state is 0.

Now, initialize the FF to 1 by pushing once the **Preset** button.

- Set D=0. Click the **Clock** button and check if the next state is 0.
- Set D=1. Click the **Clock** button and check if the next state is 1.

Flip flop JK

INITIALIZATION

Design the circuit of Figure 1 right).

- Select ***Memory*** in the library pane and get a JK FFs. Click in the FF in order to see the attributes panel and, in the option Trigger, choose *Falling Edge*.
- Connect J and K to two switches.
- Connect the output (Q) to a LED.
- Connect the asynchronous input **Clear** to a ***Button*** (menu ***Input/Output).*** Do the same for **Preset.**
- Give a permanent value 1 to **Enable**. Use ***1-Constante*** in menu ***Wiring***
- Connect the **Clock** input to another ***Button***.

COMPUTER TECHNOLOGY
PRACTICAL SESSION P4 (LOGISIM)
SEQUENTIAL CIRCUITS: FLIP-FLOPs SIMULATION

FUNCTIONALITY

Initialize the FF to 0 by clicking once the **Clear** button. After initializing the flip-flop to 0, the functionality will be checked as follows:

- Set J=1 and K=0. Click the **Clock** button and check if the next state is 1.
- Set J=0 and K=1. Click the **Clock** button and check if the next state is 0.
- Set J=0 and K=0. Click the **Clock** button and check if the next state is also 0.
- Set J=1 and K=1. Click the **Clock** button and check if the next state is 1. Push the **Clock** button again and again and check that the FF makes changes 0→1→0→1 ….

Initialize the FF to 1 by clicking once the **Preset** button. After initializing the flip-flop to 1, the functionality will be checked as follows:

- Set J=0 and K=1. Click the **Clock** button and check if the next state is 0.
- Set J=1 and K=0. Click the **Clock** button and check if the next state is 1.
- Set J=0 and K=0. Click the **Clock** button and check if the next state is also 1.
- Set J=1 and K=1. Click the **Clock** button and check if the next state is 0. Push the **Clock** button again and again and check that the FF makes changes 0→1→0→1 ….

PART 2. SIMULATION OF A 2-BITS COUNTER WITH LOGISIM

a) Figure 2 shows a 2-bits upward asynchronous counter composed of two JK flip-flops and an inverter. Implement this circuit in Logisim and use a Hex Digit Displays to see the count.

- Step-by-step simulation. First of all, initialize the circuit to 00. For that, use a common **Clear** input (C_D in the figure) as was explained in PART 1. Now, use a ***Button*** as a common clock signal (CP in the figure). In order to simulate the circuit, click the ***Button*** again and again and check if the LEDs follow the sequence 01, 10, 11, 00, 01, 10, …Don't forget to set **Enable** to 1 and choose *Rising edge* in both FFs.

- Simulation using a ***Regular Clock Signal*** (Clock). Change the ***Button*** by the ***Regular Clock Signal***, initialize the circuit to 00 and simulate the circuit at a frequency of 1Hz .

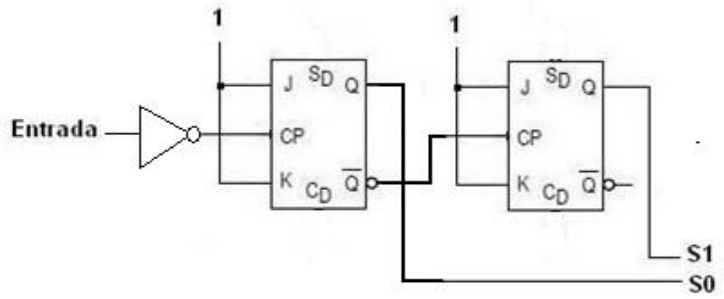

Figure 2.

b) Figure 3 shows a 2-bits downward asynchronous counter composed of two JK flip-flops. Implement this circuit in Logisim and use a Hex Digit Displays to see the count.

- Initialize now the circuit to 11 by using a common **Preset** input. Choose *rising edge* in both FFs. Simulate the circuit at a frequency of 1Hz. Which is the count now?

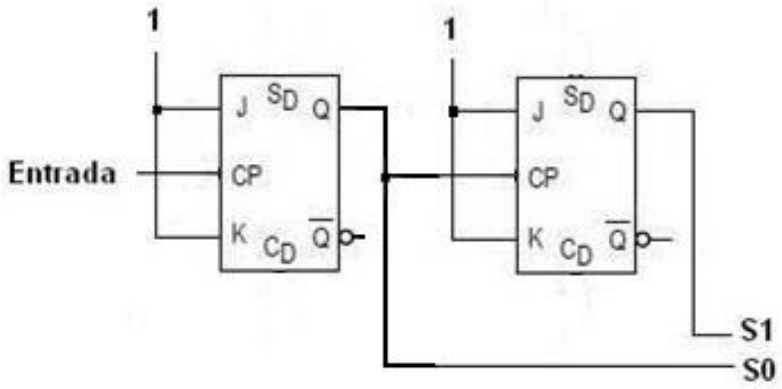

Figure 3

PART 3. IMPLEMENTING A DOUBLE COUNTER

Implement a double counter (main circuit) composed of two subcircuits C1 (2-bits upward asynchronous counter) and C2 (2-bits downward asynchronous counter). In order to synchronize both counters (simultaneous changes), it is NECESSARY to remove the inverter of Figure 2. The initial counts must be 00 for C1 and 10 for C2. When C1=C2 a yellow LED must be illuminated.

- Use two Hex Digit Displays to see both counts.
- Displays and LEDs can be found in the menu ***Input/Output***
- Simulate de circuit at a frequency of 1Hz**.**