

## CS303 - Logic and System Design Laboratory Assignment #4

**Deadline: 30.12.2022 23.55 Fall 2022/2023**

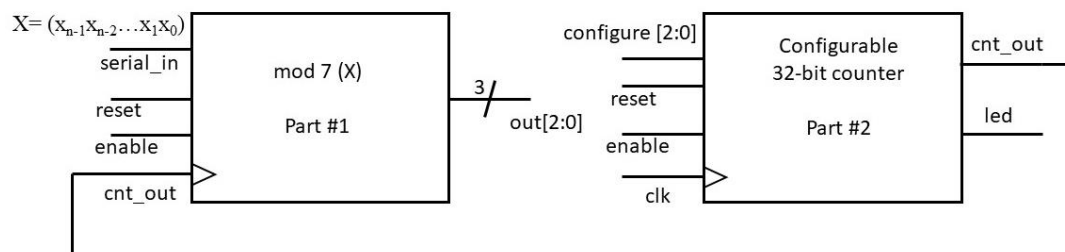
In Lab #4, you are asked to design a circuit that calculates the remainder of an arbitrary-length binary number divided by 7, using Verilog HDL. You are also asked to implement the circuit on FPGA board.

### 1. Pre-lab Assignment

The design will have two parts:

- Part #1 will calculate the mod 7 (X) of an arbitrary-length binary input.
- Part #2 will generate the clock signal (cnt\_out) for Part #1. It will be a configurable counter which is configured by a 3-bit input (configure [2:0]).

The following figure shows two parts of the design and how they interact with each other.



**Figure 1 :** Block diagram of Lab #4

#### Part #1: Mod 7 (X) Circuit

The circuit will have four inputs:

**Clock input (cnt\_out):** At every posedge of the clock signal the circuit will sample the serial input. The clock input (cnt\_out) comes from the output of the configurable counter.

**Reset input (reset):** Required to reset the state machine.

**Enable input (enable):** When enable is active high, circuit will sample the serial input. When enable is low, the circuit does not accept serial input and preserves its current state.

**A serial input (serial\_in):** You will feed the serial input through this input starting from the least significant bit (LSB) one by one. In  $n$  cycles you can feed an  $n$ -bit binary number  $X$ .

$$X = (x_{n-1}x_{n-2} \dots x_2x_1x_0)$$

The circuit will have one output:

**3-bit output (out [2:0]):** The output will show the remainder of an arbitrary-length binary number divided by 7 in each clock cycle

**Example:** Assume  $X = (1101011)_2$

$\text{out}[2:0] (0) = (000 + 1 \cdot 2^0) \bmod 7 = 001$

$\text{out}[2:0] (1) = (001 + 1 \cdot 2^1) \bmod 7 = 011$

$\text{out}[2:0] (2) = (011 + 0 \cdot 2^2) \bmod 7 = 011$

$\text{out}[2:0] (3) = (011 + 1 \cdot 2^3) \bmod 7 = 100$

$\text{out}[2:0] (4) = (100 + 0 \cdot 2^4) \bmod 7 = 100$

$\text{out}[2:0] (5) = (100 + 1 \cdot 2^5) \bmod 7 = 001$

$\text{out}[2:0] (6) = (001 + 1 \cdot 2^6) \bmod 7 = 010$

## Part #2: Configurable Clock Generator

The configurable clock counter has four inputs:

**Clock input (clk):** At every posedge of the clock signal the circuit will increment by one. This clock will be the internal clock of the FPGA.

**Reset input (reset):** Required to reset the counter. Please note that “reset” input is common with Part#1.

**Enable input (enable):** When enable is active high, counter works and LED can be turned on/off. When enable is low, counter stops, and LED preserves its state. Please note that “enable” input is common with Part #1.

**3-bit configuration input (configure [2:0]):** The configure [2:0] will determine how many cycles the counter will count with the following configurations:

If (configure = 111), the counter counts to 32'h FFFFFFFF

If (configure = 110), the counter counts to 32'h 80000000

If (configure = 101), the counter counts to 32'h 40000000

If (configure = 100), the counter counts to 32'h 20000000

If (configure = 011), the counter counts to 32'h 10000000

If (configure = 010), the counter counts to 32'h 08000000

If (configure = 001), the counter counts to 32'h 04000000

If (configure = 000), the counter counts to 32'h 02000000

**For the simulation purposes on Vivado, you can use the below configuration. Otherwise, simulation will take a long time. For the real implementation, use the previous configuration.**

If (configure = 111), the counter counts to 32'h 0000FFFF

If (configure = 110), the counter counts to 32'h 00008000

If (configure = 101), the counter counts to 32'h 00004000

If (configure = 100), the counter counts to 32'h 00002000

If (configure = 011), the counter counts to 32'h 00001000

If (configure = 010), the counter counts to 32'h 00000800

If (configure = 001), the counter counts to 32'h 00000400

If (configure = 000), the counter counts to 32'h 00000200

The configurable clock counter has two outputs:

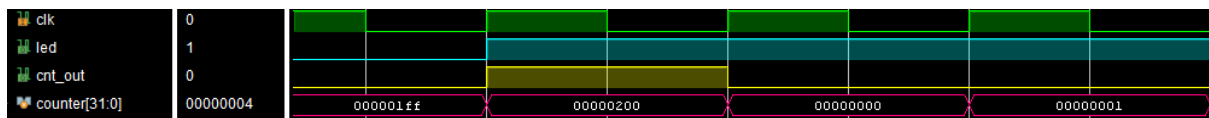
**Output showing that counting is done (cnt\_out):** Every time the counter value reaches the configured value, the cnt\_out will be asserted. In the next clock cycle, it will be deasserted.

**LED output (led):** Everytime the cnt\_out is asserted (posedge cnt\_out), the led output will change state meaning that the LED will turn ON and OFF.

Timing diagram of clk, cnt\_out, 32bits counter, and LED should be as the Figure 2, 3, 4.



**Figure 2:** Zoom out Version, LED toggles @ positive edge of cnt\_out



**Figure 3:** Zoom in Version, LED turns on transition @ positive edge of cnt\_out



**Figure 4:** Zoom in Version, LED turns off transition @ next positive edge of cnt\_out

For the constraint file, 1 push button, 5 switches and 4 LEDs will be needed. **Please use the given constraint file in the lab assignment “LAB4.xdc”.** If your design is ready, generate a bitstream file for the FPGA before coming to lab to makes it faster to test your design on the FPGA.

**Your pre-lab submission should include the following:**

1. All Verilog files used in your assignment
2. A report explaining all your work in detail: You need to show that your design works correctly by putting simulation results in your lab report

## 2. In-Lab Assignment

In the lab you will show your simulation results and implement your designs on the FPGA board.