**Computer science department, Langara college**

**Digital Systems Design (with FPGAs)**

**Spring 2019**

# Lab 3: Clock divider

by S. Arash Sheikholeslam

In this lab, you will learn about producing slower clock out of the main clock in your FPGA. **Save and** **keep all the codes you produce because you will need them in the future labs.**

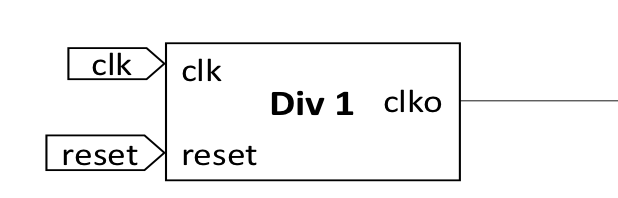
**Preparation:**

The following must be done before you come into the lab:

1. Read this handout before you start the lab.
2. Use only the pin assignment file provided by on D2L. Do not assign the pins manually.
3. Make sure you follow this manual’s instructions in terms on Vhdl file separation and naming.
4. Read a bit about clocking.
5. If you don’t get this don’t worry. I will talk about them during the class on Monday.

Part 1 [5 points]

One of the common ways of achieving the desired clock rate is to use a counter. Imagine if you have a clock at 10 Hz and you want a clock at 1 Hz, one easy way to achieve that is to use a counter such that you count from zero to 5, increasing 1 at each clock pulse. By the time, you are at the 5th pulse your output flips (from high to low or low to high.) This way you have achieve an output clock at 1 Hz.



Such a circuit is called a clock divider.

This lab is free form as in I do not care how you implement the code (behavioral structural, with or without using components.) The only requirement is to call your top-level entity **Lab3**.   
  
The above diagram takes the clock pulse and a reset as input. You have two push buttons on your board (key0 and key1), use one for reset and the other one as your clock. Define a variable to be your counter and simply count every time the clock button is pushed. Set an arbitrary value between 4-10 to toggle the output every time your counter reaches that value. Your output should connect to an LED and be initiated as on. Your reset button should set the counter to zero and the LED to on.

Part 2 [5 points]

In this part we will replace the input clock. We will no longer use the pushbutton. We use the chip’s own clock signal instead. The clock you have on board is 50 MHz.

You will now build the following circuit where you get to choose between 4 different frequencies to toggle the LED:



Question (1): Based on the above diagram, what do you think the reset signal should do?

Question (2): The counters will divide the input clock by counting up to a number n before pulsing the clock. You can calculate n by dividing the global clock frequency by your frequency of interest. We want our LED to blink at 0.5 Hz, 1 Hz, 5 Hz and 10 Hz. Find the required n and insert it in the following table.

|  |  |
| --- | --- |
| Frequency (Hz) | *n* |
| 0.5 |  |
| 1 |  |
| 5 |  |
| 10 |  |

The Multiplexer chooses the output of one of the four clock-dividers using the value on switches 1 downto 0 according to this table:

|  |  |  |
| --- | --- | --- |
| SW0 | SW1 | Div |
| 0 | 0 | 1 |
| 1 | 0 | 2 |
| 0 | 1 | 3 |
| 1 | 1 | 4 |

You will use a push button (Key0) to restart the circuit. You can implement this circuit as multiple entities and combine them structurally (that is how I would do it), or as a single entity.

Question (3): Are the pushbuttons on board active low or active high?  
Part Extra[mandatory for **those of you who finish way too early**, optional for everyone else]: Use the circuit from lab 2 and show the frequency on a pair of 7-segment. For instance, if you are blinking at 5 Hz then show 5, for 0.5 Hz show 05 .

**Performance in the lab: (10 marks)**

In this lab, you will download the circuit you designed in the Preparation on the FPGA board. The input and output pins of this FPGA are tied to the various lights and switches.

1. Read the manual carefully

1. BE SURE that you set the pin assignments before compiling your design (not doing so could damage the board!). If you have any questions about how to do this, please talk to the TA.

You must demonstrate that your circuit works to the TA or to me by the end of your lab section.

*Marking:*

Your mark for the performance part of the lab will be:

0/10: If you don’t even show up, or if you show up and don’t do anything

3/10: If you make an attempt, but really don’t get anywhere near it working

7/10: If you almost get it working, or if you get it working but can’t answer TA’s questions.

10/10: If you successfully demonstrate your design to the TA, and can answer TA’s questions.

Anything in between 0 to 3, 3 to 7 and, 7 to 10 is at instructor/TA’s discretion based on how satisfied they are with your performance and knowledge.

Note that this lab is quite easy, and you will likely finish early. Use some of the extra time to play with the software to understand what else it can do.