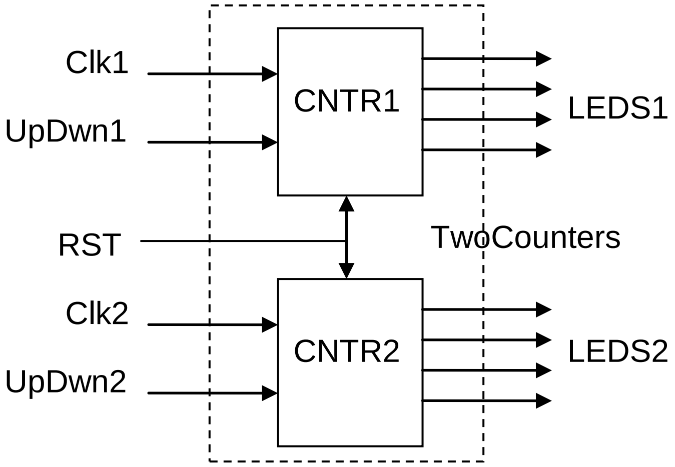
**Design Name :** Two Copies of an Up-Down Counter with Independent Inputs/Outputs

**Objective :**

Learn how to create multiple instances of a pre-described design.

**Assignment :**

1. Describe a 4-bit binary counter with `Clock`, `Up/Down` and synchronous-`Reset` inputs. Name the entity as `UpDnCtr`.
2. If you feel that it would be helpful, make your `vhd` file top-level if it is not so already and test it by connecting inputs and outputs to buttons and LEDs respectively as you have already done in the previous experiments.
3. Add a new (empty) `vhd` file with the following interfaces and make it top-level. Name the file and entity as `TwoCounters`. Interface signals of the entity would be
   `Clk1`
   `UpDwn1`
   `LEDS1` (4 bits)
   `Clk2`
   `UpDwn2`
   `LEDS2` (4 bits)
   `RST`
4. Insert the component description of the counter `UpDnCtr` you have designed in 1 into the declaration part of the architecture section of `TwoCounters`.
5. Insert two instantiations of `UpDnCtr` with names `CNTR1` and `CNTR2` in the `architecture` section. Complete `port maps`. Note that `RST` input is common and used to reset both counters.
6. Complete the constraints file by connecting `Clk` and `RST` inputs to push-buttons, `LEDS` outputs to on board LEDs. Connect `UpDwn` inputs to switches. Use pull-up, pull-down resistors when necessary.
7. Power-up and program your FPGA device through USB cable connection. Check if your design works as desired.



8. Change your `UpDnCtr` entity to a fully-synchronous design. That is, use 50 MHz clock input to store previous values of `Clk1` and `Clk2` inputs and increment/decrement counters on changes of these inputs.

**Follow Up Work :**

1. Redesign your circuit so that it works like a 2-digit BCD counter. `UpDnCtr` code should be changed to work as BCD counter. For that, you need to check the outputs of the counter with combinatorial expressions. Try to keep all inputs intact. That is, `Clk2` button should keep working to increment/decrement significant digit. It is also fed with the expression composed of `LEDS1` (`CNTR1` output).
2. What would you change if you need a 2 digit BCD counter that counts up/down between 00 and 99? Explain.

**Homework :**

Use on-board clock source (50 MHz) to feed the counter. Divide the clock signal by 50000000 by using another counter designed in `TwoCounters` architecture. Your 2 digit BCD counter will receive 1 pulse per second as a clock.

(hint: this experiment be a good exam question)