

Lab 6: Chronometer

The purpose of this lab is to build a chronometer. By doing so, you will familiarize yourself with using the buttons on the FPGA, and hopefully have a better understanding of how your VHDL code translates into memory.

In this lab, you are allowed to reuse codes that you wrote in the Arithmetic Logic Unit and Seven Segment Display labs. If you have lost your code or do not want to adapt, you can also use ALU and SSD codes from other resources **by properly citing them**. In either case, **please do not include recycled/borrowed codes in your report**.

Implement a chronometer on the FPGA. It should display the time on the **seven segment display** in **MM:SS** format (where M stands for minutes, and S for seconds). This chronometer should be able to **count** both **forwards** and **backwards**. It should have **five button inputs** with the following functions:

- **Pause/Resume:** Pause and resume the counter
- **Reset:** Reset the count to 00:00
- **Switch Direction:** Change between forwards and backwards counting modes
- **Add 10:** Add 10 seconds to the current time
- **Subtract 10:** *Simile*

Use the five buttons on your FPGA (do not use switches!) These buttons should work as one would expect; they should trigger the relevant function once per press (most of the time). For example, pressing the “Add 10” button once should not result in a 20 or 30 second increase. Also, it should not miss a button press (most of the time).

Program your FPGA. Explain your RTL schematics and show your working chronometer to your TA to get their approval. Do not forget to include screenshots of schematics and photographs of the FPGA in your report.

When you press a button, the resulting input is not always a single clear pulse. It might generate a noisy waveform (with many rising and falling edges) as the buttons are “bouncy”. You should find a way to overcome this problem and make the buttons work properly. You may search “button debouncing”. Simply checking positive changes (now 1, previous 0) with a relatively slow clock should also work most of the time. Be reminded that you should not directly write “if rising_edge(button)”, and write equivalent code inside an “if rising_edge(clock)”.

In your report try to answer the following questions:

- Do you use any type of memory when taking inputs from buttons? If so, please explain.
- What type of memory do you think your chronometer uses (latches, flip-flops etc.)? Can you show which part of your code results in this kind of memory?