

Lab #2: Describing Sequential Circuits in VHDL

1 Introduction

In this lab you will learn how to describe sequential logic circuits in VHDL. You will design a stopwatch measuring time every 10 milliseconds. Also, you will use pushbuttons and 7-segment LEDs to control the stopwatch when running on the Altera DE1-SoC board.

2 Learning Outcomes

After completing this lab you should know how to:

- Design a counter in VHDL
- Perform functional simulation of the counter using ModelSim
- Design a stopwatch measuring time at every 10 milliseconds
- Test the stopwatch on the Altera board

3 Counters

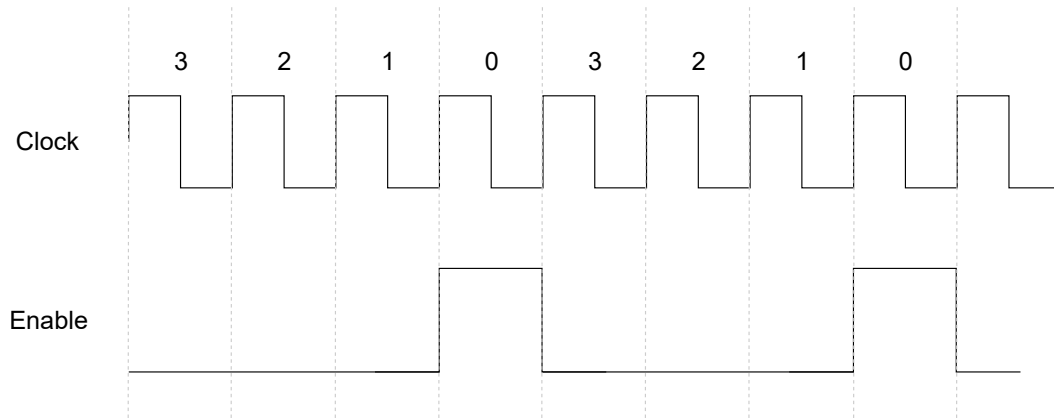
A counter is a special sequential circuit. When counting up (by one), we require a circuit capable of “remembering” the current count and adding 1 the next time we request a count. When counting down (by one), we require a circuit capable of “remembering” the current count and subtracting 1 the next time we request a count. Counters use a clock signal to keep track of time. In fact, each increment occurs (or decrement) occurs when one clock period has passed. Since counters are the main building blocks of stopwatches, we will first design an 4-bit up-counter with an asynchronous reset (which should be active low) and an enable signal. The counter counts up when the enable signal is high. Otherwise, the counter holds its previous values. Use the following entity declaration for your VHDL description of the counter:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;
entity gNN_counter is
    Port (enable      : in  std_logic;
          reset       : in  std_logic;
          clk         : in  std_logic;
          count       : out std_logic_vector (3 downto 0));
end gNN_counter;
```

Note that the up-counter that you have designed in this section will be used later in Section 5 to build a stopwatch. Once you have your circuit described in VHDL, you should simulate it. Write a testbench code and perform a functional simulation for your VHDL description of the counter.

4 Clock Divider

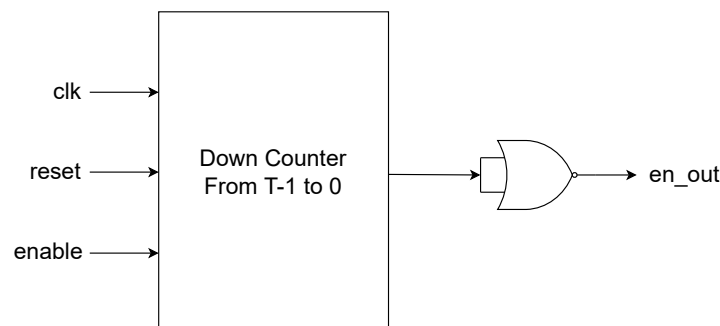
A clock divider is a circuit that generates a signal that is asserted once every T clock cycles. This signal can be used as a condition to enable the counters in the stopwatch circuit. An example of the clock and output (i.e., “enable”) waveforms for $T = 4$ is:



Implementing the clock divider circuit requires a counter counting clock periods. The counter counts down from $T - 1$ to 0. Upon reaching the count of 0, the clock divider circuit outputs/asserts 1 and the count is reset to $T - 1$. For other values of the counter, the output signal of the clock divider circuit remains 0. In this lab, we want to design a stopwatch counting in increments of 10 milliseconds. In other words, we need to assert an enable signal every 10 milliseconds. First, find the value of T for the clock divider circuit to generate an enable signal every 10 milliseconds. Note that the PLL, the device which supplies the clock for your design on the DE1-SoC board, works at a frequency of 50 MHz. Then, describe the clock divider circuit in VHDL using the following entity declaration:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;
entity gNN_clock_divider is
    Port (enable      : in  std_logic;
          reset       : in  std_logic;
          clk         : in  std_logic;
          en_out      : out std_logic);
end gNN_clock_divider;
```

Hint: the following figure shows an example of the clock divider circuit. Also, note that the down-counter inside the clock divider circuit is different from the up-counter that you designed in Section 3.

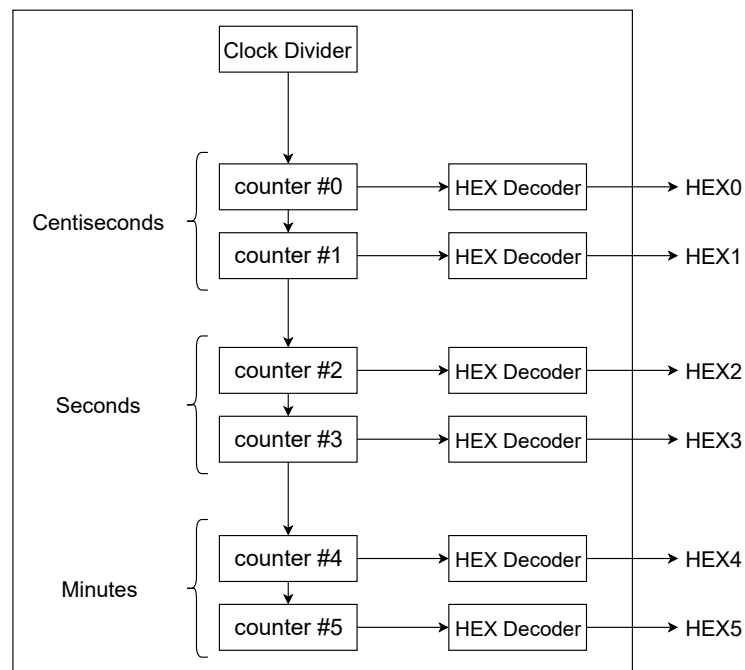


Once you have your circuit described in VHDL, write a testbench code and perform a functional simulation for your VHDL description of the clock divider.

5 Stopwatch

In this part, you will design a simple stopwatch using the counter and clock divider circuits. You will use the pushbuttons to control the stopwatch and 7-segment displays to display the elapsed time in decimal. Pushbuttons PB0, PB1 and PB2 are used to start (or resume), pause and reset the stopwatch, respectively. When these buttons are released, the circuit has to remain at the new state denoted by their corresponding function. For example, when PB1 is pushed and then released, the stopwatch circuit pauses the count until told otherwise by pushing one of the other pushbuttons. Therefore, you need a memory element to hold the operating state (e.g., running, paused) of the stopwatch. Note that the output of a pushbutton is high when the button is not being pushed, and is low when the button is being pushed. The first two 7-segment displays (i.e., HEX1-0), the second two 7-segment displays (i.e., HEX3-2) and the last two 7-segment displays (i.e., HEX5-4) are used to show time in centiseconds, seconds and minutes, respectively.

You will need to create six instances of your gNN_counter and gNN_7_segment_decoder you created in Lab Assignment #1 for each decimal digit in the stopwatch. Since we measure time in increments of 10 milliseconds, the counter measuring time in centiseconds increments only when the output signal of the clock divider circuit becomes high. The following figure shows the high-level architecture of the stopwatch circuit.



Note that counters #0, #1, #2, and #4 count from 0 to 9, while counters #3 and #5 count from 0 to 5.

Describe the stopwatch circuit in VHDL using the following entity declaration:

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;
entity gNN_stopwatch is
    Port (start      : in  std_logic;
          stop       : in  std_logic;
          reset      : in  std_logic;
          clk        : in  std_logic;
          HEX0       : out std_logic_vector (6 downto 0);
          HEX1       : out std_logic_vector (6 downto 0);
          HEX2       : out std_logic_vector (6 downto 0);
          HEX3       : out std_logic_vector (6 downto 0);
          HEX4       : out std_logic_vector (6 downto 0);
          HEX5       : out std_logic_vector (6 downto 0));
end gNN_stopwatch;
  
```

You will now test your stopwatch circuit using the DE1-SoC board. Compile the circuit in the Quartus software. Once you have compiled the stopwatch circuit, it is time to map it on the Altera DE1-SoC board. Perform the pin assignment for both HEX displays and pushbuttons according to the aforementioned instruction. Make sure that you connect the clock signal of your design to 50 MHz clock frequency (see the DE1 user's manual for the pin location of 50 MHz clock frequency). Program the board and demonstrate your stopwatch to the TA. You should be able stop, start and reset your stopwatch circuit using the pushbuttons.

6 Deliverables and Grading

6.1 Demo

Once completed, you will demo your project to the TA. You will be expected to:

- fully explain how the HDL code works,
- perform functional simulation using ModelSim, and
- demonstrate that the stopwatch circuit is functioning properly using the pushbuttons and 7-segment LEDs on the DE1-SoC board.

6.2 Written report

You are also required to submit a written report and your code on myCourses. Your report must include:

- A description of the counter and clock divider circuits. Explain why these two circuits are considered as sequential designs.
- Explain why even though we could build a clock divider using an up-counter it is easier to build the divider using a down-counter.
- A discussion of how the counter and clock divider circuits were tested, showing representative simulation plots. How do you know that these circuits work correctly?
- A description of the stopwatch circuit. Explain why you created six instances of the counter circuit in your design and why?
- A discussion of how the stopwatch circuit was tested.
- A summary of the FPGA resource utilization (from the Compilation Report's Flow Summary) and the RTL schematic diagram for the stopwatch circuit. Clearly specify which part of your code maps to which part of the schematic diagram.

Finally, when you prepare your report have in mind the following:

- The title page must include the lab number, name and student ID of the students, as well as the group number.
- All figures and tables must be clearly visible.
- The report should be submitted in PDF format.
- It should document every design choice clearly.
- The grader should not have to struggle to understand your design. That is,
 - Everything should be organized for the grader to easily reproduce your results by running your code through the tools.
 - The code should be well-documented and easy to read.

Grading Sheet

Group Number:

Name 1:

Name 2:

Task	Grade	/Total	TA Signature
VHDL code for the counter circuit		/15	
Creating testbench code for the counter circuit		/5	
Functional simulation of the counter circuit		/5	
VHDL code for the clock divider circuit		/15	
Creating testbench code for the clock divider circuit		/5	
Functional simulation of the clock divider circuit		/5	
VHDL code for the stopwatch circuit		/25	
Testing the adder circuit on the DE1-SoC board		/25	
Total		/100	