Laboratory Exercise 4

Counters

The purpose of this exercise is to build and use counters. The designed circuits are to be implemented on an Altera DE2-series Board.

Students are expected to have a basic understanding of counters and sufficient familiarity with VHDL language to implement various types of flip-flops.

Part I

Consider the circuit in Figure 1. It is a 4-bit synchronous counter which uses four T-type flip-flops. The counter increments its value on each positive edge of the clock if the *Enable* signal is asserted. The counter is reset to 0 by setting the *Clear* signal low. You are to implement a 8-bit counter of this type.

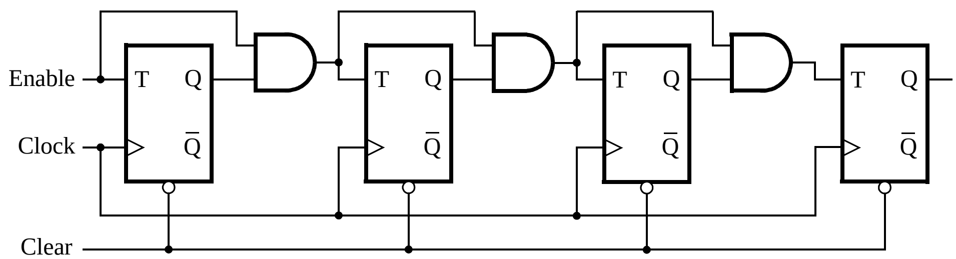

Figure 1: A 4-bit counter.

1. Write a VHDL file that defines a 8-bit counter by using the structure depicted in Figure 1. Your code should include a T flip-flop module that is instantiated 8 times to create the counter. Compile the circuit. How many logic elements (LEs) are used to implement your circuit? What is the maximum frequency, F_{max}, at which your circuit can be operated?
2. Simulate your circuit to verify its correctness.
3. Augment your VHDL file to use the pushbutton KEY_0 as the *Clock* input, switches SW_1 and SW_0 as *Enable* and *Clear* inputs, and 7-segment displays *HEX1-0* to display the hexadecimal count as your circuit operates. Make the necessary pin assignments needed to implement the circuit on the DE2-series board, and compile the circuit.
4. Download your circuit into the FPGA chip and test its functionality by operating the implemented switches.
5. Implement a 4-bit version of your circuit and use the Quartus II RTL Viewer to see how Quartus II software synthesized your circuit. What are the differences in comparison with Figure 1?

Part II

Another way to specify a counter is by using a register and adding 1 to its value. This can be accomplished using the following VHDL statement:

$$Q <= Q + 1;$$

Compile a 16-bit version of this counter and determine the number of LEs needed and the F_{max} that is attainable. Use the RTL Viewer to see the structure of this implementation and comment on the differences with the design from Part I.

Part III

Use an LPM from the Library of Parameterized modules to implement a 16-bit counter. Choose the LPM options to be consistent with the above design, i.e. with enable and synchronous clear. How does this version compare with the previous designs?

Note: The tutorial *Using the Library of Parameterized Modules (LPM)* explains the use of LPMs. It can be found on the Altera University Program website.

Part IV

Design and implement a circuit that successively flashes digits 0 through 9 on the 7-segment display *HEX0*. Each digit should be displayed for about one second. Use a counter to determine the one second intervals. The counter should be incremented by the 50-MHz clock signal provided on the DE2-series board. Do not derive any other clock signals in your design–make sure that all flip-flops in your circuit are clocked directly by the 50-MHz clock signal.

Part V

Design and implement a circuit that displays the word HELLO, in ticker-tape fashion, on the eight 7-segment displays *HEX7-0*. Make the letters move from right to left in intervals of about one second. The patterns that should be displayed in successive clock intervals are given in Table 1.

Clock cycle	Displayed pattern							
0				H	E	L	L	O
1			H	E	L	L	O	
2		H	E	L	L	O		
3	H	E	L	L	O			
4	E	L	L	O				H
5	L	L	O				H	E
6	L	O				H	E	L
7	O				H	E	L	L
8				H	E	L	L	O
...	and so on							

Table 1. Scrolling the word HELLO in ticker-tape fashion.

Preparation

The recommended preparation for this laboratory exercise includes:

1. VHDL code for **Part I**
2. Simulation of the VHDL code for **Part I**
3. VHDL code for **Part II**
4. VHDL code for **Part III**

In addition, a module that displays a hex digit on seven segment display the students designed in a previous lab would be an asset.