# Laboratory 5
Counters

This is an exercise in using counters.

**Part I**

Consider the circuit in Figure 1. It is a 4-bit synchronous counter which uses four Toggle flip-flops (see lecture notes). The counter increments its count on each positive edge of the clock if the Enable signal is asserted. The counter is reset to 0 by using the (asynchronous) Reset (called CLEAR in diagram below) signal. You are to implement a 16-bit counter of this type.

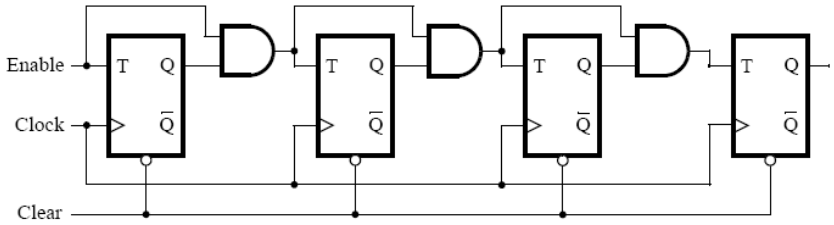


Figure 1. A 4-bit counter.

1. Write a VHDL file that defines a 16-bit counter by using the structure depicted in Figure 1 (you should have done this in your pre-lab), and compile the circuit. Look at your Compilation Report. How many logic elements (LEs) are used to implement your circuit? Under Timing Analyzer, what is the maximum frequency, Fmax, at which your circuit can be operated?
2. Simulate your circuit to verify its correctness (you should have done this in your pre-lab).
3. Augment your VHDL file to use the pushbutton KEY0 as the Clock input, switches SW1 and SW0 as Enable and Reset inputs, and use the 7-segment displays HEX3-0 to display the hexadecimal count as your circuit operates. Use the hexadecimal decoder from Lab 4. Implement your circuit on the DE2 board and test its functionality by operating the implemented switches.

**Part II**

Simplify your VHDL code so that the counter specification is based on the VHDL process:

```
PROCESS(clk, R)
BEGIN
 IF R = '1' then
 Q <= (others => '0');
 ELSIF (clk'event AND clk='1' AND enable='1') then
 Q <= Q+1;
 END IF;
 count <= Q;
END PROCESS;
```

You will need to include the following library: use ieee.std_logic_signed.all;
Compile a 16-bit version of this counter and compare the Fmax that is attainable from this design and Part I

**Part III**

Next, use the on-board 50-MHz clock signal, CLOCK_50 as your clock signal. You will need to use this clock signal to clock ALL flip-flops in your circuit. Design a clock timer that changes every second. Use a counter to determine the onesecond intervals. Hint: $2^{26}$ = 67,108,864. Create a second counter that will update every second, triggered by the first counter. Use an enable signal to update the second counter at the appropriate times. This second counter should be 16 bits wide. Display the counter results on HEX3-0