



Alexandria University
Faculty of Engineering
Electrical Engineering Department

EE432 VLSI Modeling and Design
Lab#5: System-Level Design of Sequential Circuits

Objectives

The purpose of this exercise is to build and use sequential circuits such as counters. The designed circuits are to be implemented on an Altera FPGA board. Upon the completion of this Lab, you should be able to:

1. Design sequential circuits using the behavioral description style.
2. Use Quartus II software to capture, synthesis, and implement digital FPGA systems and verify your design with Modelsim.
3. Download the design bitstream and test your implementation on the CIC-310 CPLD-FPGA kit.

Requirements

Lab 5 will expose students to the design of some sequential circuits such as counters. An 8-bit counter will be developed in Verilog using behavioral and structural modeling. A parametric counter will be developed in Verilog

You are required to write testbenches in Verilog to verify the functionality of the developed circuits. Use Modelsim to verify the functionality of the circuits and Quartus to configure the Altera FPGA. Test the implemented circuit using a set of on/off switches, pushbuttons, LEDs, and 7-segment displays offered by the CIC-310 CPLD-FPGA kit.

Counter Description:

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 an 8-bit counter of this type.

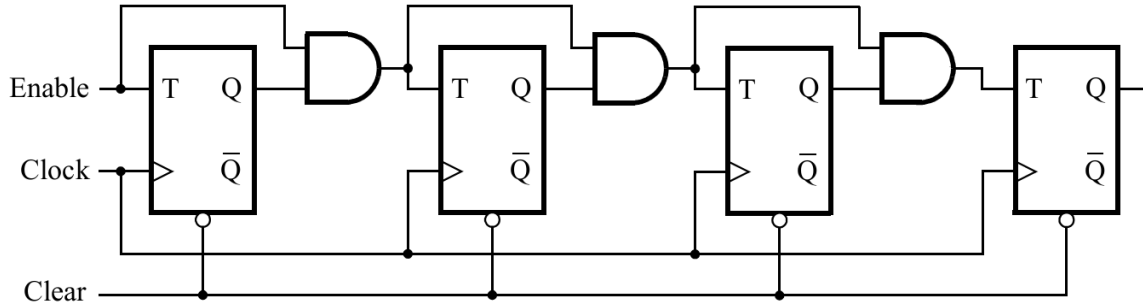


Figure 1: A 4-bit Counter

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

$$Q \leq Q + 1;$$

Compile a 16-bit version of this counter and determine the number of look-up tables (LUTs) 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 previous design.

In Verilog hardware description language we can describe a variable-size counter by using a parameter declaration. The parameter n specifies the number of bits in the counter. A particular value of this parameter is defined by using a defparam statement. By using parameters we can instantiate counters of different sizes in a logic circuit, without having to create a new module for each counter.

Create a modulo- k counter by modifying the design of an 8-bit counter to contain an additional parameter. The counter should count from 0 to $k-1$. When the counter reaches the value $k-1$ the value that follows should be 0.

Your circuit should use a pushbutton as an asynchronous reset, and another pushbutton as a manual clock input. The contents of the counter should be displayed on seven-segment displays. Compile your design with Quartus II software, download your design onto the FPGA board, and test its operation.

Procedures:

1. Write a Verilog file that defines an 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 LUTs and registers 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 Verilog file to use a pushbutton as the Clock input, two switches as Enable and Clear inputs, and 7-segment displays to display the hexadecimal count as your circuit operates. Make the necessary pin assignments needed to implement the circuit on the CIC-310 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?
6. Write a Verilog file that defines an 8-bit counter and describe its operation using behavioral modeling as described above. How many LUTs and registers are used to implement your circuit? What is the maximum frequency, f_{max} , at which your circuit can be operated?
7. Can you describe the counter using for loop? If so, compare between the RTL structures generated for both behavioral descriptions in steps 6 and 7.
8. Repeat steps from 2-6 for the new top-level counter.
9. Write a Verilog file that defines a generic n-bit counter and describe its operation using behavioral modeling.
10. Repeat steps from 2-6 for the parametric counter with $n=16$ and 32 .