EE342 - Digital System Design

Laboratory Experiment - 1
Module Instantiation

Preliminary Work

1. Read the Verilog Introduction lecture notes, **EE342_L04_VerilogIntro.pdf**.

2. Read the section, "**5.3.2 Module Instantiation**", in **EE342_L05_VerilogCont.pdf**. If you have trouble in understanding module instantiation then read EE342_L04_VerilogIntro.pdf again.

3. Install Quartus II web edition software on your computer following the instructions given in EE342_Lab_QuartusIntro.pdf.

4. Write a Verilog module that implements a 3-bit counter with the following inputs and outputs:

Clock input: Triggers at rising edge

Reset input: Active-low asynchronous reset input

Counter outputs: 3-bit counter outputs incremented at every rising clock edge as long as the reset input is high.

5. Write a Verilog module that performs either AND or EX-OR operation between two operands. The module has three inputs and an output:

Input-1: First operand of AND or EX-OR operation.

Input-2: Second operand of AND or EX-OR operation.

Function selector input: Selects operation type; 0=>AND, 1=>EX-OR.

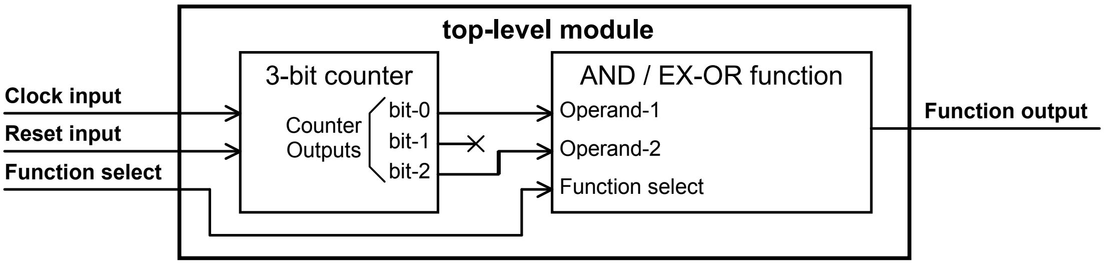
Function output: Result of selected operation.

6. Save the source code of the modules as plain text, in a file with "**.v**" (Verilog source) extension.

Procedure

1. Create a project directory and set up a new project in Quartus II using the **File->New Project Wizard...** menu item following the instructions given in EE342_L06_QuartusIntro.pdf.

2. Create a top-level Verilog module for the project that instantiates the two modules you wrote in the preliminary work. The top-level module should organize all inputs and outputs as shown in the figure given below:



3. Save the top-level module source code. Add the source code of the instantiated modules into project file list. Save your project.

4. Compile and debug the project.

5. Run the simulation tool of Quartus II and check if your code produces the correct output.

5.a) Create a new waveform file and make it simulation input/output file.

5.b) Set "End Time" of the waveform file to 10 μs.

5.c) Include all input/output pins in the waveform display.

5.d) Generate a 10 MHz clock as the clock input.

5.e) Run a simulation and check if the output waveform is as expected.

5.f) Display the counter module output waveforms on the screen. Run the simulation again and verify that your code generates the correct output waveform for all possible input combinations.