CSCE 230 – Lab 6: Quartus and ModelSim

Due: 11:59PM, Oct 3 (Tuesday)

Objectives

- Learn how to use Altera Quartus and ModelSim to write, compile, and simulate VHDL

Useful References on Canvas

- Lecture notes for Appendix A
- Quartus Introduction Document
- ModelSim Introduction Document
- ModelSim Command Document

Lab: Quartus and ModelSim

Step 1: Installing software (not required, if you use lab computers)

- You should have already installed Quartus and ModelSim in the first lab. If not, please follow the instructions in the first lab to install it.
- Please download ModelSim (version 18.1) at the following links, and then install https://www.intel.com/content/www/us/en/software-kit/665990/intel-quartus-prime-lite-edition-design-software-version-18-1-for-windows.html

Step 2: Creating a Quartus project

- Open "Quartus Prime 18.1 Lite Edition"
- Click menu "File", and then "New Project Wizard".
- Click button "Next" to go to the "Directory, Name, and Top-Level Entity" page
  - Select a directory for the project. *If you are using the lab computers, please create a new project directory in C:\temp instead of your Z: drive (after the lab, please backup your files to your Z: drive though); otherwise, Quartus does not work correctly.*
  - Set your project name, which could be anything.
  - **Set the top-level design entity to "mux2input1bit", which is the circuit that we will design in this lab**. Note that "*This name is case sensitive and must exactly match the entity name in the design file*."
- Click button "Next" to go to the "Add Files" page
  - Here we can add files that we have already written, but for now we have none.
- Click button "Next" to go to the "Family & Device Setting" page.
  - Under "Device family", select "**Max 10**"
  - Under "Available devices", find and select "**10M50DAF484C7G**".
- Click button "Next" to go to the "EDA Tool Settings" page.
  - Here we can specify any other software tools to use, but we do not need do anything here today.
- Click button "Next" to bring you to a summary and then click "Finish".

Step 3: Writing a VHDL design file

- Create a new VHDL file
  - Click menu "File", and then "New" to open a "New" window
  - Choose "VHDL File" under "Deign Files", and then click "OK".
- Type the following VHDL design

```
library ieee;
use ieee.std_logic_1164.all;

entity mux2input1bit is                    -- circuit interface
      port(
            s,a,b   : in  std_logic;
            output  : out std_logic);
end mux2input1bit;

architecture implementation1 of mux2input1bit is -- circuit implementation
begin
      output <= (not s and a) or (s and b);
end implementation1;
```

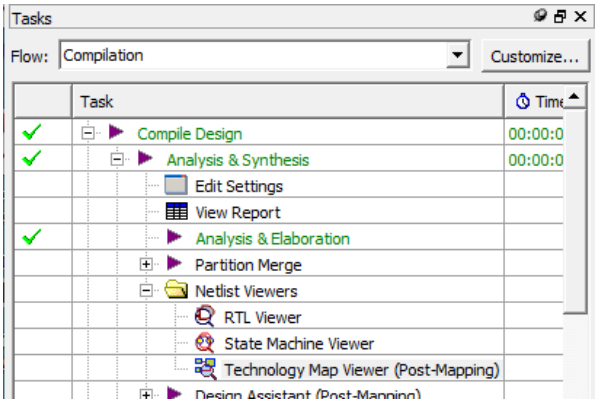
- Save your VHDL file
  - Click menu "File", and then "Save as" to open a "Save as" window
  - Set file name to "mux2input1bit.vhd".
  - **Make sure "Add file to current project" is checked**, and make sure the directory is your project directory
  - Click "Save"

Step 4: Compiling your project

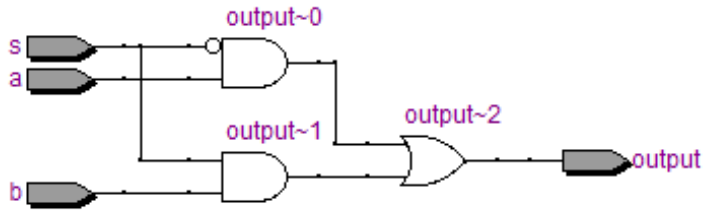
- Click menu "Processing", and then "Start Compilation". Or you can just click  on the toolbar.
  - You should see a message box "Full Compilation was successful". Ignore these warnings.
- Answer the following questions on Canvas.
  - **Question 1**: Remove the first two lines: "library ieee;" and "use ieee.std_logic_1164.all;". What is the error message when compiling the project?
    - The error message is shown in the red color in the "Messages" Window. If you could not find the "Messages" window, click menu "View", select "Utility Windows", and then "Messages".
    - Undo the change, and then continue
  - **Question 2**: Replace "output : out std_logic);" with "output : out std_logic;". What is the error message when compiling the project?
    - Undo the change, and then continue

Step 5: Viewing the generated circuit

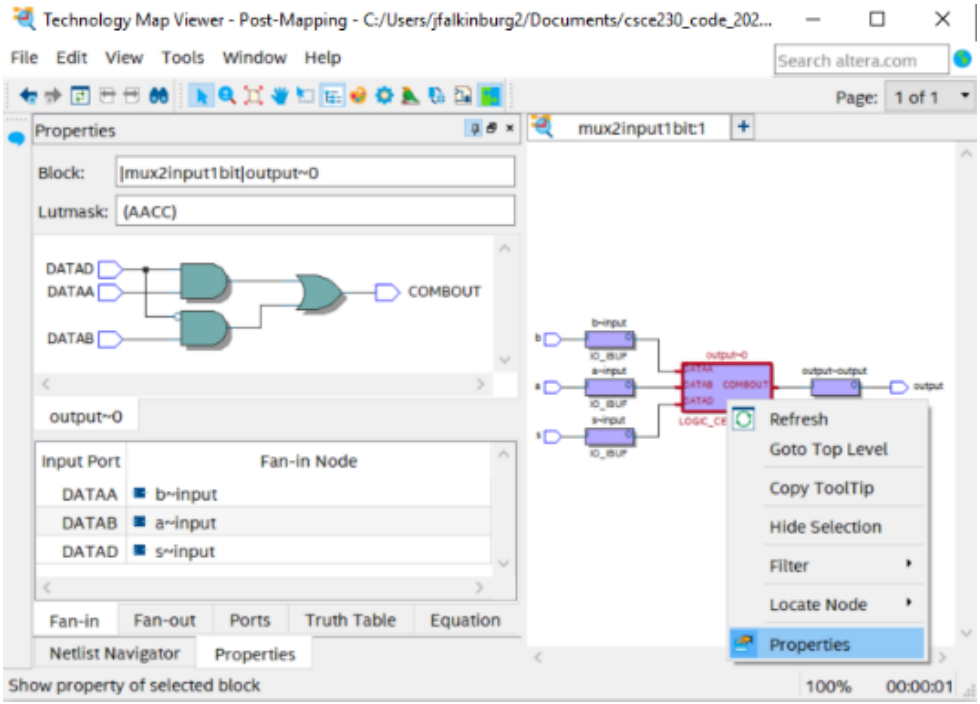
- Find the "Tasks" panel on the left-hand side. If you could not find it, click menu "View", select "Utility Windows", and then "Tasks"
- Expand "Compile Design", expand "Analysis & Synthesis", and expand "Netlist Viewers" as shown below



- Double-click "RTL viewer" to view the original logic circuit as shown below.



- Double-click "Technology Map Viewer (Post-Mapping)" to view the minimized logic circuit as shown below. Note that you need to right click and view properties on the main block in the "Technology Map Viewer" window to show the detailed gates used in that block.



- We can see that RTL Viewer shows the exactly same logic circuits as the Technology Map Viewer, because the original logic circuit is already the minimized logic circuit.
- Answer the following questions on Canvas
  - Please replace "output <= (not s and a) or (s and b);" with "output <= (not s and a and not b) or (not s and a and b) or (s and not a and b) or (s and a and b);"**, re-compile your project,** and then answer the following questions.
  - **Question 3**: What is the implementation cost of the original logic circuit shown in RTL viewer? Recall that the implementation cost is the sum of the total of number of AND and OR gates plus the total number of gate inputs.
  - **Question 4**: What is the implementation cost of the minimized logic circuit shown in Technology Map Viewer? Again please double click on the main block in the "Technology Map Viewer" window to show the detailed gates.
  - Please undo the change, and then continue.

Step 6: Writing the simulation script

- Create a new simulation file
  - Click menu "File", and then "New" to open a "New" window
  - Choose "Text File" under "Other Files", and then click "OK".
- Type the following simulation script

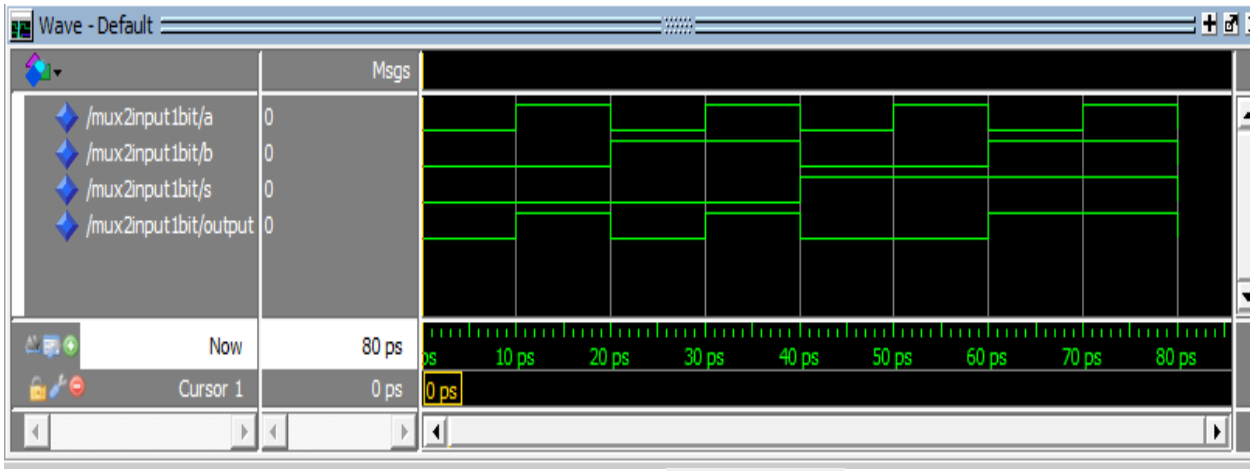
```
#the name must be the same as your vhdl filename
vsim mux2input1bit

#view waveform
view wave

#view input signals
add wave a
add wave b
add wave s

#view output signal
add wave output

#set input signal values
force a 0 0, 1 10 -repeat 20
force b 0 0, 1 20 -repeat 40
force s 0 0, 1 40 -repeat 80

#run simulation
run 80
```

  - This simulation script will
    - simulate the mux2input1bit.vhd file
      - **vsim** `filename`
    - show the waveform of signals a, b, s, and output
      - **view wave**
      - **add wave** `signal`
    - test different input signal values
      - **force** `signal value1 time1, value2 time2 -repeat time3`
      - This command sets `signal` to `value1` at `time1`, to `value2` at `time2`, and then repeats this pattern every `time3`
      - The values are binary by default.
      - The unit of times are the default time resolution (ps=10^{-12} second).
      - This simulation script tests all eight possible combinations of signals a, b, and s in 80 ps (one combination for 10 ps).
    - run the simulation for 80 ps
      - **run** `simulation_time`
  - You can find more information about the simulation commands from the ModelSim Commands document on Canvas.
- Save your simulation script
  - Click menu "File", and then "Save as" to open a "Save as" window

- Set file name to "task.do".
- **Make sure "Add file to current project" is checked**, and make sure the directory is your project directory
- Click "Save"

Step 7: Simulating the circuit

- Setup your Quartus project to use ModelSim
  - Click menu "Tools", and then click "Options" to open the "Options" window.
    - Click "EDA Tool Options", then specify the path to the executable ModelSim program (for example, C:\intelFPGA_lite\18.1\modelsim_ase\win32aloem).
    - Click "OK"
  - Click menu "Assignments", and then click "Settings" to open the "Setting" window.
    - Click "Simulation" under "EDA Tool Setting"
    - Set "Tool name" to "ModelSim-Altera"
    - Set "Format for output netlist" to "VHDL"
    - Select "Script to compile test bench", and then select the task.do file.
- Simulate your circuit
  - Before simulating your circuit, check whether the task panel has any yellow ?. If so, you need to compile your project again until you get green ✓.
  - Click menu "Tools", click "Run Simulation Tool", and then click "RTL Simulation " or "Gate Level Simulation" (no difference for us) to start ModelSim. Or you can directly click  or  on the toolbar.
  - The "Wave" window of ModelSim should looks as follows



  - If waves are too small or large to view, please click  to zoom in or zoom out the wave window.
  - You can find the value of a signal at a time by clicking the wave of that signal at that time.

- Answer the following questions on Canvas
  - **Question 5**: At simulation time 55 ps, what is the value of signal a?
  - **Question 6**: At simulation time 55 ps, what is the value of signal b?
  - **Question 7**: At simulation time 55 ps, what is the value of signal s?
  - **Question 8**: At simulation time 55 ps, what is the value of signal output?
  - **Question 9**: Run a simulation and leave ModelSim open. Now go back to Quartus and run another simulation. What error do you encounter?
  - **Question 10**: Close ModelSim, change the first line of task.do from "vsim mux2input1bit" to "vsim mux2input" and save it. Now run a simulation. What is the error message of ModelSim?
    - The error message is shown in the red color in the bottom "Transcript" window of ModelSim
    - You may need to scroll up the "Transcript" window to find the red error message
    - Undo the change, and then continue to the next question
  - **Question 11**: Close ModelSim, change "add wave output" to "add wave outputt" and save it. Now run a simulation. What is the error message of ModelSim?