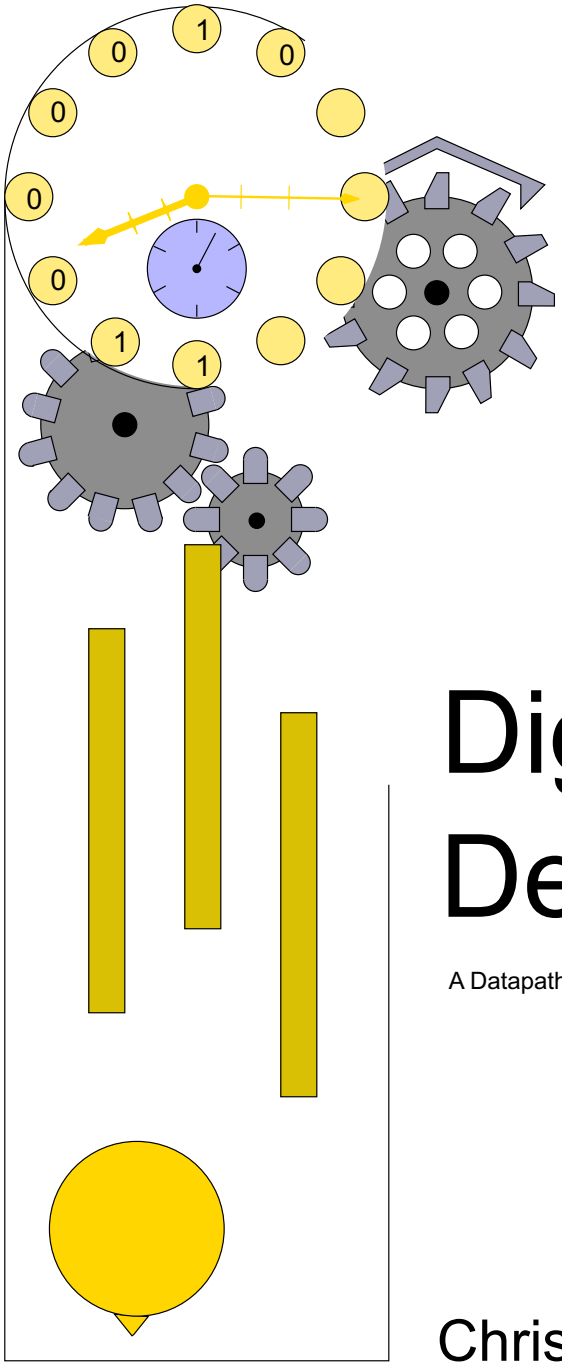


Digital Design

A Datapath and Control Approach

Chris Coulston

November 21, 2024

Contents

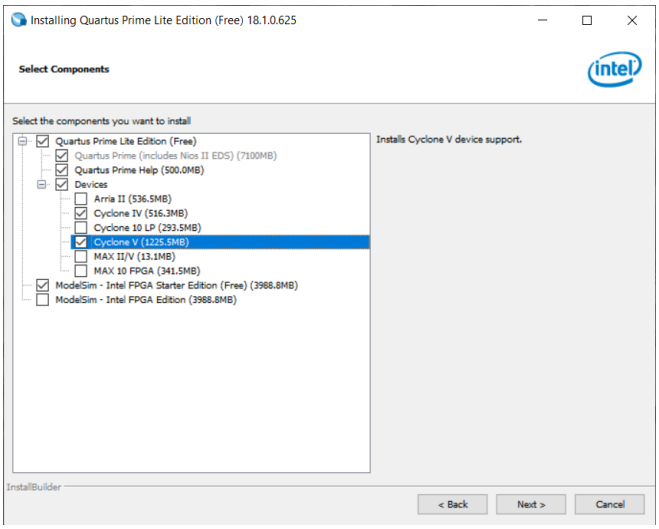
When clicking on a link in Adobe use alt+arrow left to return to where you started.

How To 1

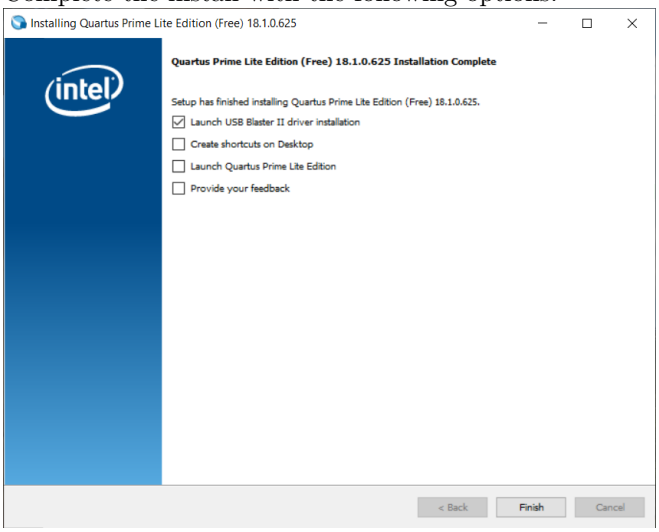
Installing Quartus software.

To download and Install Quartus II and ModelSim on your home computer, follow these instructions.

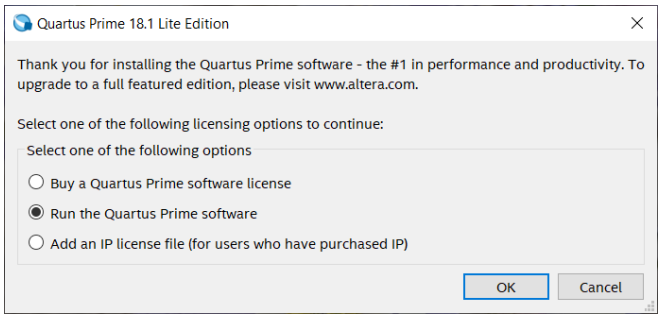
- Start at this link to download Quartus. Link: https://www.intel.com/content/www/us/en/software/programmable/quartus-prime/download.html
- Click on the person icon in the upper right and either create an account or sign in with your existing account
- Click on the "Download now" button for Lite Edition
- Click the "18.1 (2)" link on the left side of the screen
- Click the "Intel® Quartus® Prime Lite Edition Design Software Version 18.1 for Windows"
- On the redirect Download Center page click the "Download Quartus ... tar" button
- Accept the license agreement
- The approximately 5.8GB download should automatically start
- Uncompress the zip file. If needed download and install WinZip to uncompress.
- Double click on QuartusLiteSetup-18.1.0.625-windows and follow the prompts
- Select the options given below



- Install takes about 20 minutes.
- Complete the install with the following options.



- The Device Driver Wizard should auto launch after the install is finished, follow the prompt and finish.
- You need to restart your computer to complete the installation.
- When you run the Quartus software for the first time, you will be prompted to decide on a license. Select the option below.



You should be ready to start using Quartus to write Verilog and use ModelSim to check your designs.

How To 2

Creating a Project

1. Select an appropriate working directory for your project. I would recommend selecting your network drive.
   a. Create a new folder for your project *projectFolder*,
   b. Download any provided Verilog files into *projectFolder*,
2. Start Quartus II 18.1 (64-bit).
3. Select *File -> New Project Wizard.*
4. In the **Directory, Name, Top-Level Entity** page of the New Project Wizard pop-up:
   a. To the right of the "What is the working directory" box click the ... button,
   b. In the Select Folder pop-up, navigate so you can see the andgate2 directory created in step 1,
   c. Select the *projectFolder* folder, click Select Folder,
   d. In the "What is the name of this project" field type *projectName*
   e. click *Next.*
5. In the **Project Type** page of the New Project Wizard pop-up:
   a. Select the *Empty project* radio button,
   b. click *Next.*
6. If you have given Verilog files, in the **Add Files** page of the New Project Wizard pop-up:
   a. Click the ... button to the right of File name,
   b. In the Select File pop-up, navigate to, and select, *projectFiles.v*, click Open,
   c. The file should appear in the window below,
   d. Click *Next*
7. In the **Family & Device Settings** page of the New Project Wizard pop-up:
   a. Device family, Family: Cyclone V
   b. Package: FBGA
   c. Pin Count: 672
   d. Speed Grade: 7_H6
   e. Select *Specific device selected in 'Available devices' list*
   f. From the list of available devices, select: 5CGXFC5C6F27C7
   g. Click *Next*
8. In the **EDA Tool Settings** page of the New Project Wizard pop-up:
   a. In the Simulation row

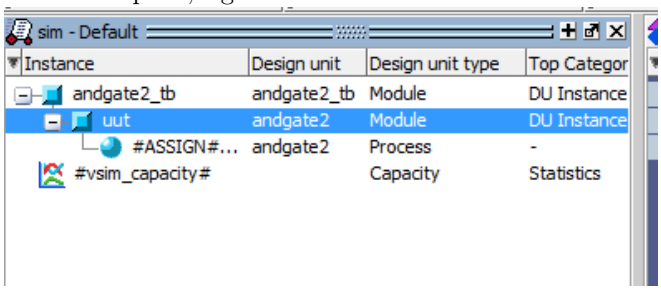
   i. Tool Name column: ModelSim-Altera
   ii. Formats column: Verilog HDL
  b. Leave other defaults alone
  c. Click Next
9. In the **Summary** page of the New Project Wizard pop-up:
  a. Review information,
  b. Click Finish.
10. Back in the main Quartus II window, Click *Tools -> Options...*
11. In the Options pop-up:
  a. Select *EDA Tool Options* from the Category menu,
  b. If the last row, "ModelSim-Altera" is blank, click on the ... button at right and navigate to the *C:\intelFPGA_lite\18.1\modelsim_ase*, select the *win32aloem* folder, the click Select Folder,
  c. Click Ok.

How To 3

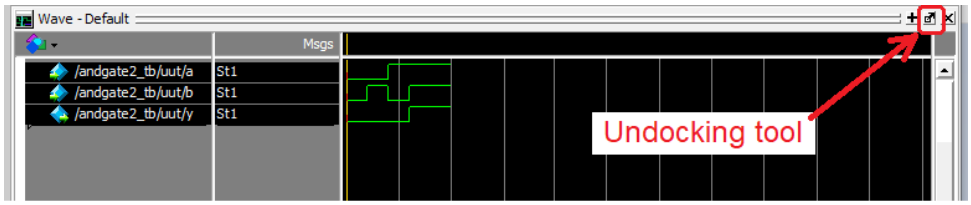
Performing a Simulation

If you are planning on performing a simulation of your design then the top level entity should be a testbench. Inside the testbench should be an instantiation of your design as the unit under test.

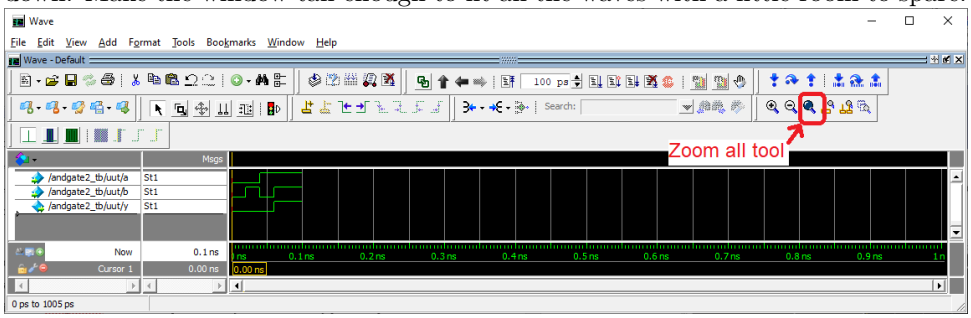
1. Click on the Files tab in the *Project Navigator* pane.
2. Right click on *topLevelProjectFile.v* in the *Project Navigator* pane and select Set as Top-Level entity.
3. Click on the Hierarchy tab in the *Project Navigator* pane.
4. In the main Quartus II window, click on *Processing -> Start -> Start Analysis & Elaboration.* This may take some time, so be patient.
5. You can close the compilation report by clicking on the x in the red box,
6. You should see *topLevelProjectFile.v* as the root entity in the Hierarchy tab in the *Project Navigator* pane.
7. In the main Quartus II window, click *Tools -> Run Simulation Tool -> RTL Simulation.* The ModelSim program will launch. This may take a few moments, be patient.
8. In ModelSim, find the *Library* pane. Expand the *work* library by clicking on the "+" at left. Right click on *topLevelProjectFile* and click *Simulate.*
9. In the sim pane, right mouse click on uut and select *Add Wave.*



10. Choose *Simulate -> Run -> Run 100.* You should see inputs and output from *topLevelProjectFile.*
11. If you are asked to save the waveform. Perform the following steps:
    a. Undock the Wave pane by clicking the undocking tool icon.



b. Resize the undocked Wave window vertically by grabbing its top edge and dragging down. Make the window tall enough to fit all the waves with a little room to spare.



c. Click the Zoom all tool to file the available horizontal space with the waveform.
d. Re-order the waves so that the inputs are highest and outputs are lowest. Do this by grabbing their name and dragging it to the correct location.
e. Color the intermediate signals (p1, p2, p4, p7) yellow by right clicking on them, selecting properties. In the View tab of the Wave Properties pop-up, click the Colors... button for Wave Color and choose Yellow, click Close, then OK.
f. Color the output signals red. Leave the input signals green.
g. Click File -> Export -> Image
h. Navigate to your project directory, provide a File name, then click Save

12. Close ModelSim. Do not save wave commands.

How To 4

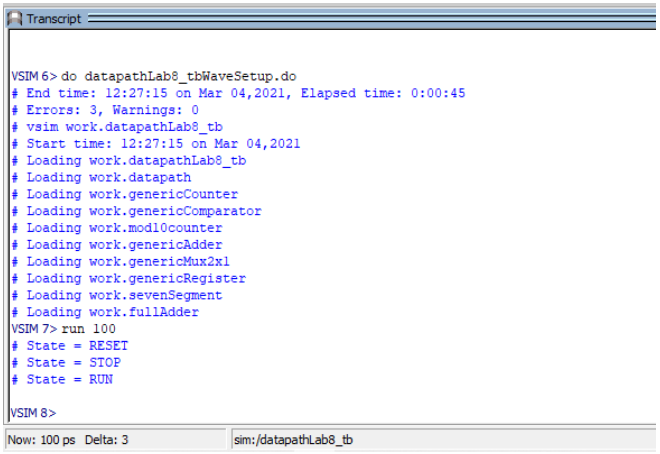
Using a Do file

I find setting up the testbench waves to be a pain, especially when you are making a lot of mistakes and need to rerun your simulation multiple times; each time setting up the waveforms. In order to simplify the process of setting up the waveforms, you can write a script files that performs the waveform setup and then call the script file inside ModelSim. The script file is called a "do" file. They are very easy to make and will save you time. If a do file is provided to you, you will most likely need to edit it because your signal names may be different.

In the discussion below I have used two placeholders: `<labName>` is the name of your testbench module. `<projectDirectory>` is the system path to your Verilog files corresponding to your project.

- If provided, download `<labName>_tbWaveSetup.do` into the: `<projectDirectory>\simulation\modelsim` directory
- If a do file is not created, you can use the template provided in Listing 4.1 as a starting point to make one for yourself. Make sure to put the do file in the: `<projectDirectory>\simulation\modelsim` directory
- Open `<labName>_tbWaveSetup.do` file using Notepad. The syntax is pretty straight forward and corresponds to the text displayed in the ModelSim console window when you add or modify waveforms.
- From Quartus, you need to:
  - Make sure that your testbench is the top-level. Do this in the Project Navigator, select File view and then right click on the file testbench and select "Set As Top Level Entity"
  - Launch the simulation. Do this by selecting Tools - Run Simulation Tool - RTL Simulation
  - This will launch Model Sim for your testbench
- From Model Sim, you need to:
  - Maximize the Model Sim window – this makes it easier to see all the subwindows.
  - In the library subwindow, open the **work** library
  - Right click on your testbench and select Simulate
  - In the console area of ModelSim (shown in the image below) type:

```
VSIM 3> do  <projectName>\_tbWaveSetup.do
```



- You can type "run <time>" in this area (as shown) to simulate some amount of time. I found this VERY handy when debugging my Verilog code.
- Also note that the console has tab completion. This allows you to type the first few characters of a command/filename and press Tab to fill in the rest of the command/filename. If there is more than one choice, the command/filename will be completed up to the ambiguity.

4.1 Example do file for hiLow Module

- Run the testbench for the hiLow module provided on Canvas. Produce a timing diagram with the following characteristics. Zoom to fill the available horizontal space with the waveform. Color inputs green and outputs red. Order the traces from top to bottom as

signal	radix	color trace
t_seedSwitch	unsigned	green
t_guessSwitch	unsigned	green
t_playSwitch	unsigned	green
t_randBut	default	green
t_hiLowBut	default	green
LFSR	unsigned	yellow
t_randNum	hex	red
t_randDisp	hex	red
t_hiLowSeg	hex	red
t_greenLEDs	default	red

- The do file for this testbench is shown in Listing 4.1. From top to bottom the sections are as follows.
  - Any line that starts with a "#" is a comment. The URL is a complete reference for do file syntax.
  - The restart command resets the simulation. I included this because I sometimes like to rerun the same simulation multiple times. This isn't particularly useful for combinational logic circuits.
  - The delete wave command removes any waveforms that may have been added previously. Again, I included this because I sometimes like to rerun the same simulation multiple times
  - The add wave command puts a signal into the waveform viewing area. There are

two parameters included which you will find helpful.
  * Radix changes what base the waveform value is displayed.
  * Color changes the color that the waveform is displayed.
- Once you have created the do file, you call it by running it from the console area using the do command discussed previously.
- You can advance the simulation time using the run command discussed previously.

Listing 4.1: do file for hiLow_tb.

```
# ------------------------------------------------------------------
# File:  hiLow_tbWaveSetup.do
# ------------------------------------------------------------------
restart -f
delete wave *

add wave -position end  -radix unsigned -color green sim:/hiLow_tb/t_seedSwitch
add wave -position end  -radix unsigned -color green sim:/hiLow_tb/t_guessSwitch
add wave -position end  -radix unsigned -color green  sim:/hiLow_tb/t_playSwitch
add wave -position end  -color green sim:/hiLow_tb/t_randBut
add wave -position end  -color green sim:/hiLow_tb/t_hiLowBut
add wave -position end  -radix unsigned -color yellow sim:/hiLow_tb/uut/randNum
add wave -position end  -radix hex -color red sim:/hiLow_tb/t_randDisp
add wave -position end  -radix hex -color red sim:/hiLow_tb/t_hiLowSeg
add wave -position end  -color red sim:/hiLow_tb/t_greenLEDs
```

4.2 Model Sim commands

ModelSim macros (also called DO files) are scripts that contain ModelSim and, optionally, Tcl commands.

You run the commands by typing tyhem in the console window at the bottom of the ModelSim window. What follows are the most popular commands that we use in the class. This is not an exhaustive list. You can find the complete list of command in the `ModelSim® Command Reference Manual` pdf. You will have to search the internet for this file as its location changes.

do `<file>` The do command executes commands contained in a macro `<file>`. If you provide no argument, the simulation runs for the default time (100 ns).

run ¡time¿ This command advances the simulation by the specified ¡time¿.

restart This command retarts the simuloation and resets the simulation time to zero. The -f option specifies that the simulation will be restarted without requiring confirmation in a popup window.

delete wave This command removes waveforms from the Wave window. The * option removes all the waves.

add wave This command can add signals, waves and busses to the Wave window:
- -position Specifies where the command adds the signals.
- -radix Specifies a user-defined radix.
- -color Specifies the color used to display a waveform.
-

radix define This command is used to create a user-defined radix used to map bit patterns to a set of enumeration labels. This command is used to map the binary codes for the states in a finite state machine (FSM) into symbolic names that are displayed on the timing diagram. This makes it much easier to understand what state your FSM is in. This bit patterns may contain "?" as a don't care specifier.

vsim `<testbench>` This eliminates the need to open the word library and simulate the testbench. This is a legacy command that is no longer used in the class, until you see it.

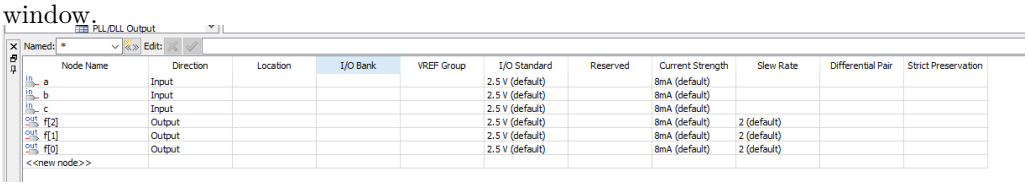
How To 5

Synthesizing a Verilog Module

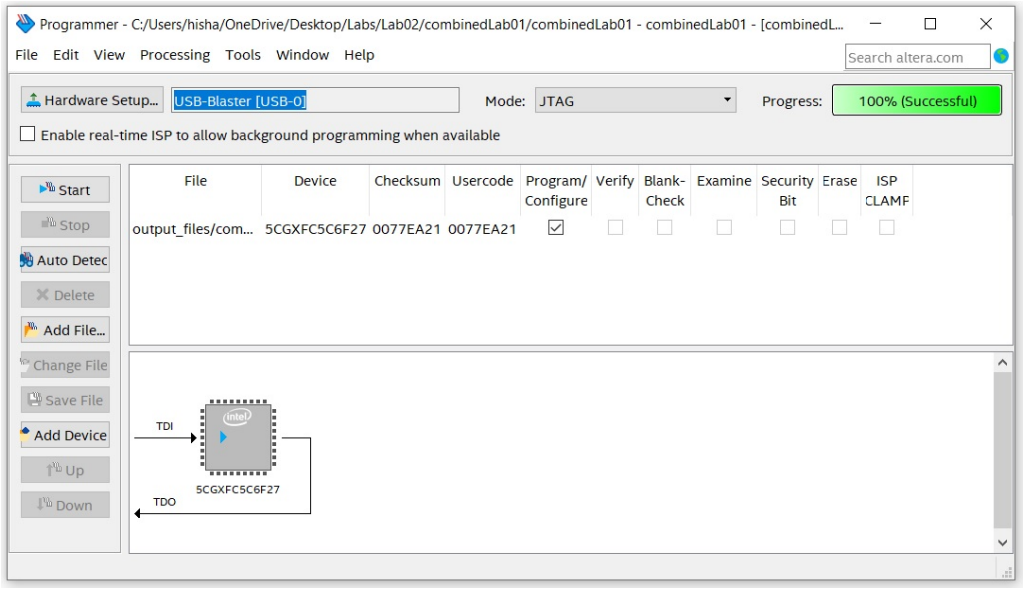
The combinedLab01 verilog file used in this example has three inputs and three outputs that are mapped to slide switches and LEDs.

It's time to realize the *combinedLab01* Verilog file to FPGA. To do this follow these steps:

1. In Project Navigator pane, select the File tab
2. Right mouse click *combinedLab01.v* and select Set As Top Level Entity.
3. Processing -> Start -> Start Analysis and Elaboration
4. Assignments -> Pin Planner
5. In the Pin Planner pop-up you should see the pin assignment pane at the bottom of the window.



6. Double click in the Location cell for row c
7. Scroll down the list of pins to PIN_AC9
8. Complete the pin assignment for the other 5 inputs and outputs using the information contained in pin assignment table completed earlier.
9. Double check your pin assignments.
10. File -> Close. Note closing your file incorporates this assignment into the project.
11. Back in the Quartus window, Processing -> Start Compilation <Ctrl-L>
12. Tools -> Programmer
13. In the Programmer pop-up window click Add File...
14. In the Select Programming File pop-up, navigate to your project directory, then into the output files folder, the select combinedLab01.sof, click Open. You should see something like the following.



15. Connect the Altera Cyclone V GX FPGA to your computer through the USB port, connect the power supply, and push the red power-on button. Try not to be annoyed by the infernal blinking LEDs.
16. In the Programmer pop-up
    a. Click Hardware Setup....
    b. In the Hardware Setup select USB-Blaster [USB=0] from the Currently selected hardware pull-down
    c. Click Close
17. Back in the Programmer window, the box next to Hardware Setup... should reflect your choice. Click Start,
18. The Development board should stop its infernal blinking and run your program. Your design is now running on the FPGA.