EE342 - Digital System Design

Laboratory Note

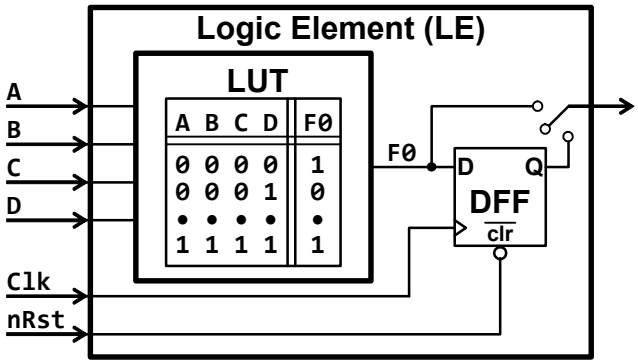
Quartus Introduction

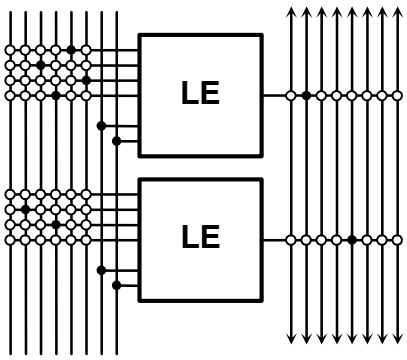
Contents:

1 FPGA Structure

Digital circuits designed in Quartus II are compiled to be used in programmable logic devices manufactured by Altera. Field programmable gate array (**FPGA**) is the most common type of these devices. As in any other circuit construction process, an FPGA handles two problems; **1)** define function of circuit elements, and **2)** make connections between the circuit elements.

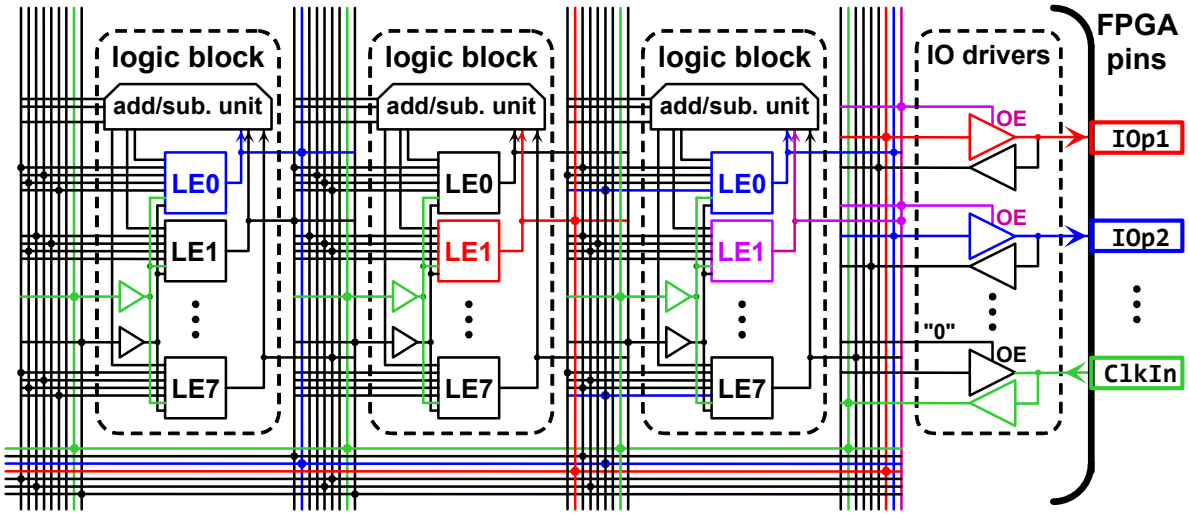
The first problem is solved by ***logic elements*** (**LE**) that are the basic building blocks of programmable logic circuits. A typical logic element shown below has a ***look-up table*** (**LUT**) that can store all possible output combinations as a function of the inputs. A 4-input LUT is a tiny memory that has 16 addressable locations. LUT inputs are the memory address, and stored data bits are the output function just like the truth table given for any logic function. Function of each logic element is stored into the LUT memory when the FPGA is programmed. Each logic element can be used as a combinational logic block, or its output can be stored in a flip-flop to implement sequential logic circuits.





The logic elements are surrounded by a connection grid that carries input and output signals as shown on the right. Signal paths are routed between logic elements and input/output pins through programmable switches on the connection grid. Circles in the figure represent the programmable switches. Rings indicate the switches that are left open, and the solid black circles indicate the switches that connect inputs or outputs to the grid.

Eight or more logic elements are combined to form **logic blocks** as in the following figure. Each logic block has a dedicated addition/subtraction unit that performs simple arithmetic operations very efficiently. High-speed counters and accumulator functions are implemented easily with these addition/subtraction units. Signal routing examples are shown with colored lines in the figure. The green clock input pin is connected to all logic blocks. The driver of **`ClkIn`** pin is disabled permanently, since this pin is always used as an input. Blue logic element in the first block is wired to the third block, and then to a bidirectional FPGA pin. Similarly, red logic element in the second block is wired to another bidirectional pin. Output enable (OE) signals of these bidirectional pins are controlled by another logic element to change the input/output functions dynamically.



An FPGA may contain other types of functional blocks, such as multipliers, addressable memory blocks, and frequency multipliers with phase-lock-loops (PLL) to generate clock signals with variable frequency. The term, "***field programmable,***" implies that the circuit program is downloaded into the FPGA in the ***field***, away from the manufacturing site. The FPGA program is stored in temporary memory and it is lost when the FPGA power is turned off. Therefore, a microprocessor or some other type of device should download the FPGA program after the FPGA power is turned on again. Other types of programmable logic devices have permanent program memory that remains intact when power is turned off. FPGA manufacturers also provide FLASH memory chips that can download the FPGA program automatically after power is turned on. These memory chips eliminate the need for an external programming source in case an FPGA is required to operate as a stand-alone device.

2 Quartus II Installation

Quartus II is a design software provided by Altera (www.altera.com) to support digital design applications for programmable logic devices. Quartus II combines Verilog and VHDL compilers with the place-and-route, timing analysis and simulation tools. It is possible to compile a Verilog project and observe the behavior of the synthesized circuit on a single development tool.

The latest software version is Quartus II Web Edition v14.1 for Windows. The installation files for this newest version take more than 2 GByte, and it requires Modelsim as an additional program for simulation (+1.1 GByte). An earlier version,

Quartus II Web Edition v9.0, is sufficient for laboratory applications and project assignments in this course. The setup file, **90_quartus_free.exe**, is 1.4 GByte and it is less demanding on the computer resources. This setup file is available on a DVD with the course assistants.

If you can copy **90_quartus_free.exe** to your hard disk then installation will be faster compared to running the same program directly from DVD. Disabling your anti-virus program can speed up the installation significantly. You should first **1)** physically disconnect your PC from the local Ethernet network, **2)** turn off your wireless modem, and **3)** disconnect the PC from any other ISP connection before you disable the anti-virus program.

These are the features you need to be aware of during the installation:

MAX+Plus II - like Interface:
MAX+Plus II was the old support software distributed by Altera before Quartus II became available. Selecting this option adds a pull-down menu list for convenient access to frequently used tools and utilities in Quartus II.

Full or Custom Installation:
You may choose Custom installation to save disk space, and delete the support options for unnecessary Altera FPGA and PLD families. Support options for some device families, such as Cyclone-III and Stratix series of devices can take several hundred megabytes of disk space. We will only need Quartus II support for **Cyclone** and possibly **Cyclone-II** families for this course. You can always start the Quartus II setup again and add the necessary options in the future.

Setting Options

Following are the useful options that can make your first HDL experience less painful. Some of these options should be selected as default already, and some of them may be moved into a different category in future Quartus II releases.

1. Select the **Tools->Options** menu item.

In **General** category:
- Check Re-open current project at startup option.
- Select "**Verilog**" as HDL preference.
- Check Show full file path in window titles option. This will make sure that you are using the correct files, not the files with same name in an old project folder.

In **General/Internet Connectivity** category:
- **Un-check** Check the Altera web site for new Quartus information option.

In **General/Processing** category:
- Check the "Overwrite simulation input file with simulation results" box.
- Select "**Vector Waveform File (.vwf)**" as Simulation results format.

2. Select the **Assignments->Settings** menu item.

In **Compilation Process Settings** category:
- Un-check Preserve fewer node names to save disk space option.

3 Making and Compiling a Project

Create a new working directory with a short path information that will allow you to keep track of your project files easily. It doesn't had to be in the default project directory of Quartus II that was determined during installation. You should have a separate directory for each homework or project. It is better if you organize the project files directly under one of the hard drives in your PC.

Starting a new Quartus II project involves opening a new project file and performing all required steps in the **Assignments** menu, such as selecting a device, setting all device and synthesis options, and assigning all device pin functions and labels. In your case, you just need to follow a few steps to set the minimum required options. Quartus II software will provide the default settings for the remaining options and assign the pins automatically when you compile your project.

You will start with creating a simple Quartus II project. Making copies of the first project for other experiments is not suggested. Links to all project files should be checked after a project is moved into another directory. You can always make copies of your Verilog files from the first experiment and include them in the other projects.

Setting Up a New Project:

1. Start Quartus II and select the **File->New Project Wizard...** menu item.

2. Click on the **[. . .]** (browse) button and select the working directory for your project.

3. Type in the project name that will be copied as the top-level entity name of the project. The top-level entity name is case-sensitive and it should exactly match the top-level module name in your Verilog files. Please spend a few seconds to give your project a meaningful name instead of using a previously defined module name. Click on the **[Next]** button.

4. Click on the **[. . .]** (browse) button and add the Verilog file(s) you prepared for the project. If you don't have a Verilog file readily available, you can open a new file and add it to the project later. Click on the **[Next]** button.

5. Select "**Cyclone**" as the device family and then select "**EP1C3T144C8**" among the available devices. The device selection is not critical, but it is better if all projects in the class are based on the same device to avoid compatibility problems in the future. Click on the **[Next]** button.

6. You don't need to choose any other EDA tool. You will use the synthesis, simulation, and analysis tools available in the Quartus II software. Click on the **[Next]** button and then click on the **[Finish]** button.

Writing Verilog Code:

Select the **File->New...** menu item and then the "**Verilog HDL File**" among the Device Design File options. Write the Verilog code and make sure that "**Add file to current project**" box is checked when you save the file. You can use the **Edit->Insert Template...** menu item to start with the basic syntax of common Verilog constructs.

Verilog HDL file has the "**.v**" extension and it is a plain text file. You may prefer to setup your Windows operating system to open Verilog files using a simple text

editor such as WordPad. If you do so, you can quickly see the contents of Verilog files without starting Quartus every time.

Adding/Removing Project Files:

You need to complete two steps to include additional Verilog modules from other files in the project:

1. Instantiate the new module in the top-level or some other module that is defined in the existing project files.

2. Tell Quartus where to find your module description. If the new Verilog module is in a separate file, then you need to include this file in the project so that the Quartus II Verilog compiler can locate it.

To add a Verilog file to the project:
1. Select the **Project->Add/Remove Files in Project...** menu item.
2. Click on **[. . .]** button to select the file(s) you want to add, and then click on the **[Open]** button.
3. Click on the **[Add]** button and then the **[OK]** button in the "Settings" window.

To remove a file from the project:
1. Select the **Project->Add/Remove File in Project...** menu item.
2. Click on the file you want to remove in the list of project files.
3. Click on the **[Remove]** button and then the **[OK]** button.

Compiling:

Select the **Processing->Start Compilation...** menu item. A new window will appear and display information about the compilation process. The tabs below the message window can select the compiler messages at a certain severity level (error, critical warning, warning, information) and hide all others for easy viewing. You will receive the following four warning messages during compilation mostly because of the unused pins and synthesis options.

Warning: Feature LogicLock is not available with your current license
You need to pay money to use this feature.

Warning: Found pins functioning as undefined clocks and/or memory enables
Your design has a clock input. Normally, it should be assigned to a dedicated clock input.

Warning: No exact pin location assignment(s) for 4 pins of 4 total pins
You did not make any input/output assignments to the actual FPGA pins. The compiler made the assignments randomly.

Warning: The Reserve All Unused Pins setting has not been specified, and will default to 'As output driving ground'.
The selected FPGA, **EP1C3T144C8**, has nearly 100 I/O pins and the project used only a few pins. Unused pins are set to "**0**" outputs by default.

Any additional warning message besides those listed above may indicate some kind of trouble in the Verilog code or in the project settings. You should become familiar with the type of warning messages (at least with the total number of warnings) you get after compiling a project successfully. Sometimes your mistakes that cause incorrect results may appear as a simple warning, not as a fatal compiler error.

Viewing Project Hierarchy:

Quartus II has a "**Project Navigator**" window to provide easy access to project files. If the Project Navigator window is closed, you can activate it selecting **View->Utility Widows->Project Navigator** menu item. One of the navigation options (located at the bottom of the Project navigator) displays a list of all the files included in a project. Double-click on a file name to open the project file.

Another navigation option (first option at the bottom) displays the "entities" or Verilog modules used in the project starting with the top level. Note that Quartus will not be able to show all modules in the project until it completes the Analysis & Synthesis part of the compilation successfully after you add a new module file. Once you have the hierarchy display updated, then you can easily access the individual module definitions just as you open the project files.

4 Simulation

You are ready to test your Verilog code and see the simulation results after you successfully compile a project. The first step is to create a waveform file where you can define the input waveforms and save the simulation results. You can do the following using the Quartus **Simulator Tool** and **Waveform Editor**:

1. Generate input signals for input pins and bi-directional (tri-state) pins.
2. See simulated output signals at the output pins and bi-directional pins.
3. See most intermediary signals inside the modules.

Creating a Waveform File:

Select the **File->New...** menu item and then select "**Vector Waveform File**" type listed under Other Files or Simulation/Test Files categories. Quartus will open a blank waveform file. Quartus will suggest the project name as the default file name when you save the waveform file for the first time. It is better to give a different name (i.e. HW2sim1.vwf or HW2test3.vwf) so you can easily distinguish them from the other simulation or project files.

Adding Waveforms:

Select **Edit->Insert->Insert Node or Bus...** menu item. You can type in a I/O pin label or a complete reference to an internal signal node. It is easier to use the **[Node Finder...]** where you can select the signals to be displayed from a list of available input/output pins or internal nodes. Node Finder allows listing of available node names based on two criteria:

1. Node Name: Restricts the list of node names to include a certain text. Asterisk (*) can be used as a wild-card character to extend the search criteria.

2. Filter: Selects an option to determine the type of nodes to be displayed. The most useful options are:
Pins: all: Lists all input/output pins of the top-level module.
Pins: all & Registers: post fitting: Lists all internal storage element outputs in addition to the input/output pins of the top-level module.

Design Entry (all names): All node names defined in the design entry files.
Post Compilation: Node names that are still available after compilation.

You can see the output of storage elements that are assigned to a logic element after the compilation. If you cannot find a signal node in the **Node Finder** these can be the possible reasons:

- Synthesis tool eliminated the node since it did not have any effect on an output pin directly or indirectly. You should check if your code has the correct output connections.
- Node is eliminated or it is merged with another node with the same functionality after the optimization.
- A particular node name disappeared after the synthesis because there was another name assigned to the same node.
- Node is defined as output of a combinational logic that is embedded in a look-up table. Remember that look-up-tables were used in an FPGA to implement combinational logic gates. Quartus may combine several logic gates in a single look-up table, and the node you are looking for disappears in that look-up table.
- Node list is not up-to-date. You need to compile again.

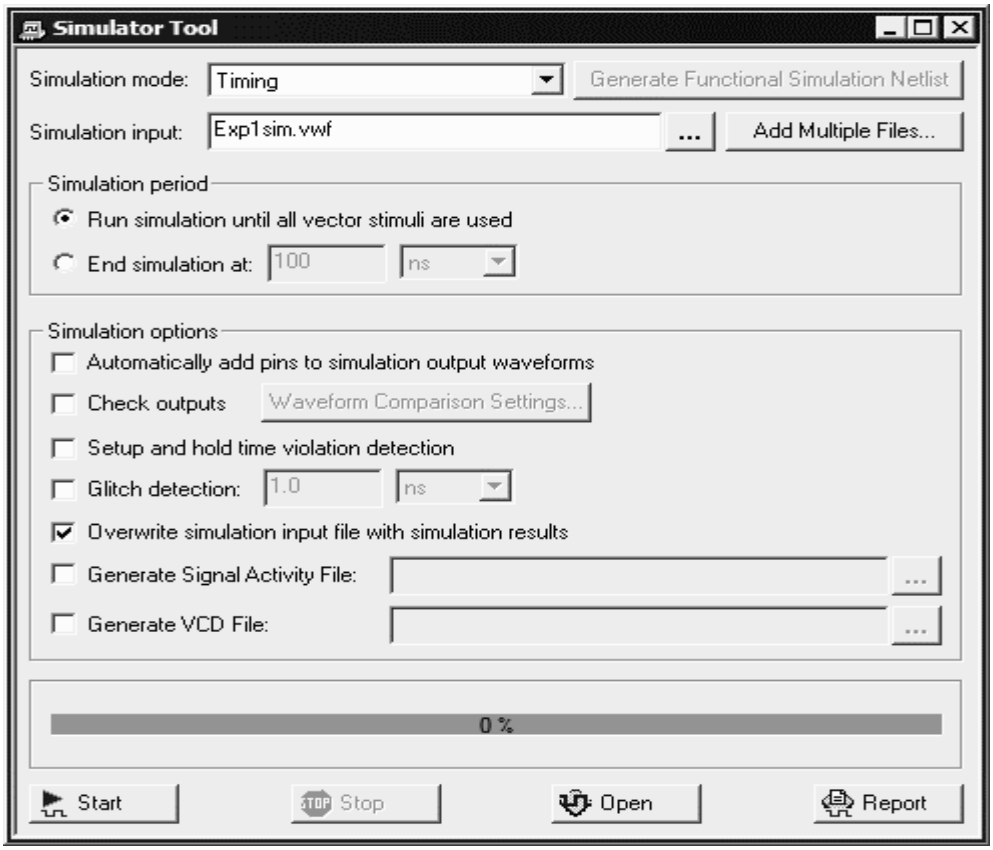
Editing Waveforms:

You can use the following commands to edit waveforms. The **Waveform Editor** of Quartus should be active (a waveform file should be selected) to enable these options.

- Activate "Waveform Editor" toolbar if you closed it: Tools->Customize Waveform Editor.../Toolbars.
- Change the end time of time axis: Edit->End Time...
- Click on a waveform label to select a complete waveform.
- Select a complete waveform and drag it up or down to reorder waveforms.
- Select a waveform or part of it and use **[0]**, **[1]**, **[Z]** buttons on the Waveform Editor toolbar to set signal level.
- Select a waveform or part of it and use **[clock]** button to generate a periodic clock signal.
- Select a bus (array of nodes such as Pdata[7:0]) or part of it and use **[?]** button to set values.
- Select a bus and use the Edit->Grouping options to see the individual bits.

Start the Simulator:

Select the **Processing->Simulator Tool** menu item, or select the **Simulator** option under **MAX+PLUS II** menu to open the Quartus simulator tool. First you need to select a waveform file as the "**Simulation input**" using the **[...]** (browse) button. Following picture shows the Simulator Tool settings:



Running Simulation:

Click on the **[Open]** button to activate the Waveform Editor and to see the contents of the simulation input file if it is not already active. Click on the **[Start]** button to run a simulation. Quartus will ask for your confirmation to display the simulation results when you select the waveform display again. Click on **[Yes]** to display the updated simulation results.

5 Tips to Avoid Trouble

- Always make sure that the Verilog files you edit are the files in the project and the waveform file you open is the simulation input file. Every person can make this mistake while working with Quartus or another development environment. You edit a file but you cannot see any changes at the output, because the file you are editing is not the file you compile or simulate.
- If you move the project files in a different directory, then make sure that all project files refer to the correct folder. You can see the full path of a file if you click on the **[Properties]** button after selecting one of the files in the project files list.
- Avoid creating a project directory on your desktop (masaüstü), because the actual path information will be too long. A directory path, such as "**`C:\Documents and Settings\<user name>\Desktop\EE342\Projects\Homework1`**" is not completely visible in window titles or project file listings.
- Save the simulation waveform file after you edit inputs and update the waveform display after simulation.

Tips to Avoid Virus Infections for Windows Users:

Tip 1: Make at least two partitions when you initialize your hard disk next time. One partition is for installation of operating system and other executable programs and the second partition is for your stuff (pictures, music, homeworks, projects, etc.). 99% of infections invade the system partition. Next time you get an infection, you can format the system drive only, and leave your data drive intact. You may need more partitions for 1) storing setup/installation disks of system, drivers and other software, 2) installation of other operating systems.

Tip 2: Consider your external USB drive (FLASH or HD) as infected, if it touches a public computer, such as 1) a computer in an Internet cafe, 2) presentation system at a conference or meeting, 3) printing facility in a bookstore or a stationary store.

Tip 3: Turn off the unnecessary networking protocols. You don't need "Client for Microsoft Networks" and "File and Printer Sharing for Microsoft Networks" to have Internet access and POP/SMTP e-mail access. Open "Network and Dialup Connections" window and check the properties of individual network connections.

Tip 4: Avoid using poorly maintained or unknown wireless network access points.