Lab 1
Introduction to FPGA Design Flow

Objective

The main objective of this lab is to introduce the trainees to the fundamentals of Intel's Quartus design flow and its tool set such as the synthesis tool, the Questa simulator, and the FPGA implementation.

Equipment

In each laboratory, there is a set of PCs with the FPGA Intel® Quartus® Prime design tools installed. You can also download it from Intel`s website at Intel® Quartus® Prime Lite Edition Design Software Version 22.1.2 for Windows. You will also be provided with the FPGA development boards and other equipment.

This lab is divided into two sections. Part 1 will guide the trainee through the steps of the FPGA Design flow. Part 2 includes an introduction to Quartus, the steps required to create a digital system using Verilog, and test the designed system on the FPGA board.

1. A typical FPGA Design Flow

The typical FPGA design flow is illustrated in Figure 1. Quartus software will be used to implement this flow through using the graphical user interface (GUI). The standard FPGA design flow involves the following steps:

1. **Design Entry** – the desired circuit is specified either by means of a schematic diagram or by using a hardware description language (HDL), such as Verilog or VHDL

2. **Synthesis** – the entered design is synthesized into a netlist that consists of the logic blocks provided in the FPGA chip.

3. **Functional Simulation** – the synthesized circuit is tested to verify its functional correctness; this kind of simulation does not consider any timing issues.

4. **Placement and Routing (Fitting)** – the Fitter tool determines the placement of the logic blocks defined in the netlist into what is actual FPGA chip has; it also chooses routing wires in the chip to make the required connections between specific blocks.

5. **Timing Analysis** – propagation delays along the various paths in the fitted circuit are analyzed to provide an indication of the expected performance of the circuit.

6. **Timing Simulation** – the fitted circuit is tested to verify both its functional correctness and timing.

7. **Programming and Configuration** – the designed circuit is implemented in a physical FPGA chip by programming the configuration switches that configure the LEs (Logic Elements) and establish the required wiring connections.

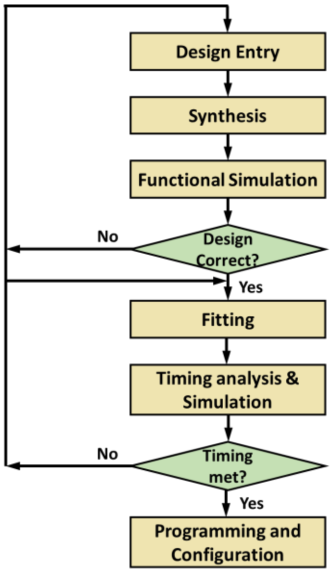
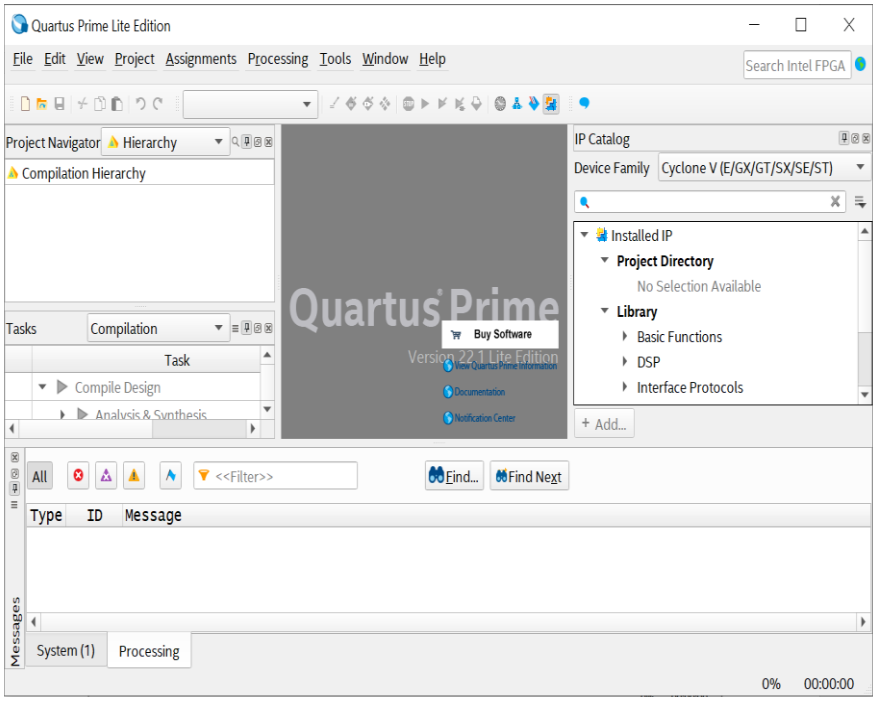


Figure 1. Typical FPGA flow.

2. Introduction to Quartus

A complete step-by-step instruction for using the Quartus software to implement a simple circuit in an Altera FPGA device is presented.

First, start the Quartus software, you should see a display similar to the one in Figure 2.



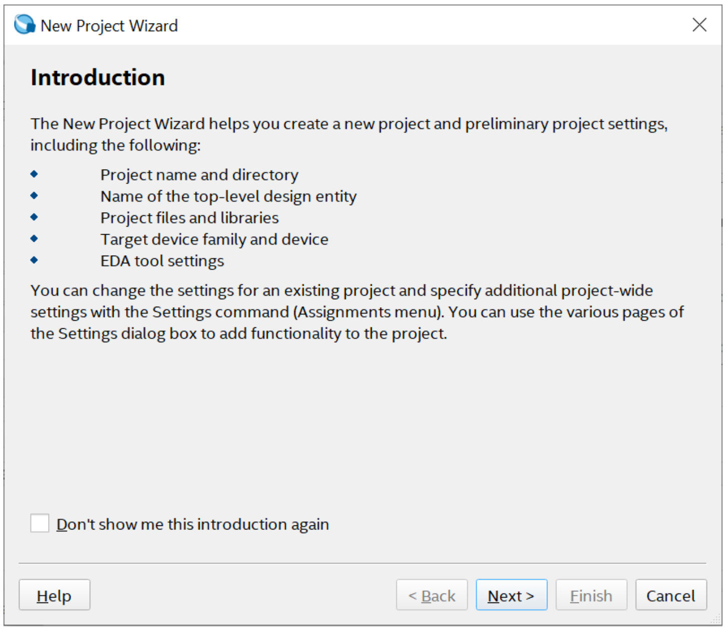
Device Family: Cyclone V (E/GX/GT/SX/SE/ST)

Figure 2. The main Quartus Prime display.

A ***project*** is a set of files that maintain information about your FPGA design. To compile a design or make pin assignments, a project should be created first.

- **Create a project.**

1. In the Quartus software, select **File > New Project Wizard**. The introduction page is as in Figure 3



New Project Wizard

Introduction

The New Project Wizard helps you create a new project and preliminary project settings, including the following:

- Project name and directory
- Name of the top-level design entity
- Project files and libraries
- Target device family and device
- EDA tool settings

You can change the settings for an existing project and specify additional project-wide settings with the Settings command (Assignments menu). You can use the various pages of the Settings dialog box to add functionality to the project.

□ Don't show me this introduction again

Figure 3. Tasks performed by the wizard

2. Click > **Next**.

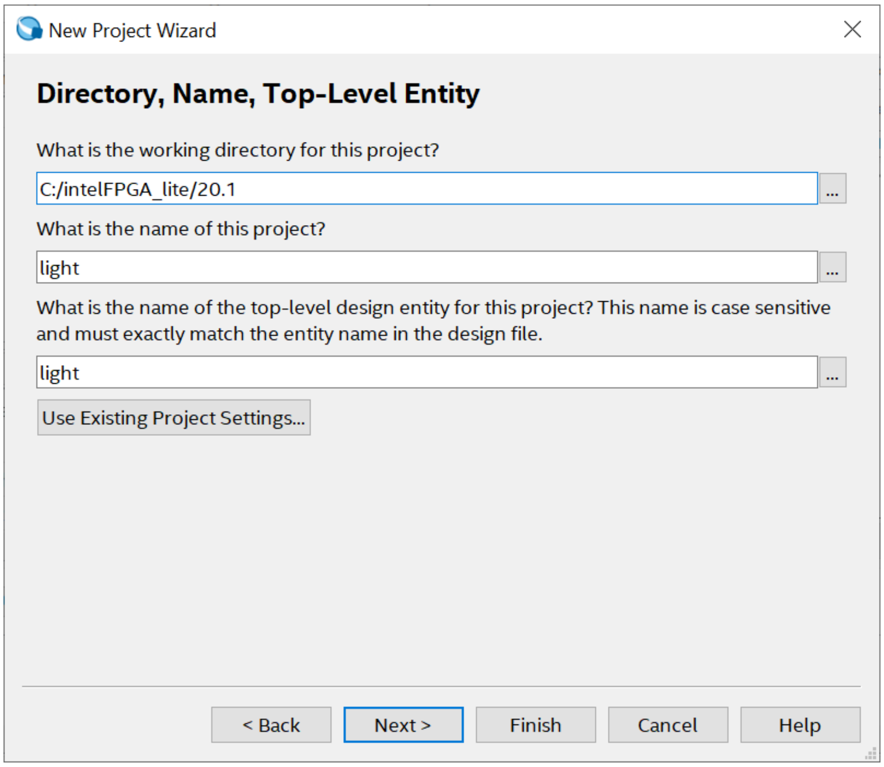
3. Enter the following information about your project: the project must have a name, which is usually the same as the top-level design entity that will be included in the project. Choose **light** as the name for both the project and the **top-level entity**, as shown in Figure 4. Click > **Next**.

4. After creating a new project wizard, the wizard makes it easy to specify which existing files (if any) should be included in the project as shown in figure 5.

5. click > **Next**, which leads to the window in Figure 6. You should specify the type of device in which the designed circuit will be implemented.

6. Choose **Cyclone V** as the target device family. A specific device can be selected by Quartus software, or we can choose the device explicitly. From the list of available devices, choose the device called 5CEBA4F23 which is the FPGA used on Altera's DE0 board. Press > **Next** to open the window in Figure 7.

7. The user can specify any third-party tools that should be used. We will not choose any other tools. Press **Next**.


New Project Wizard

Directory, Name, Top-Level Entity

What is the working directory for this project?

C:/intelFPGA_lite/20.1

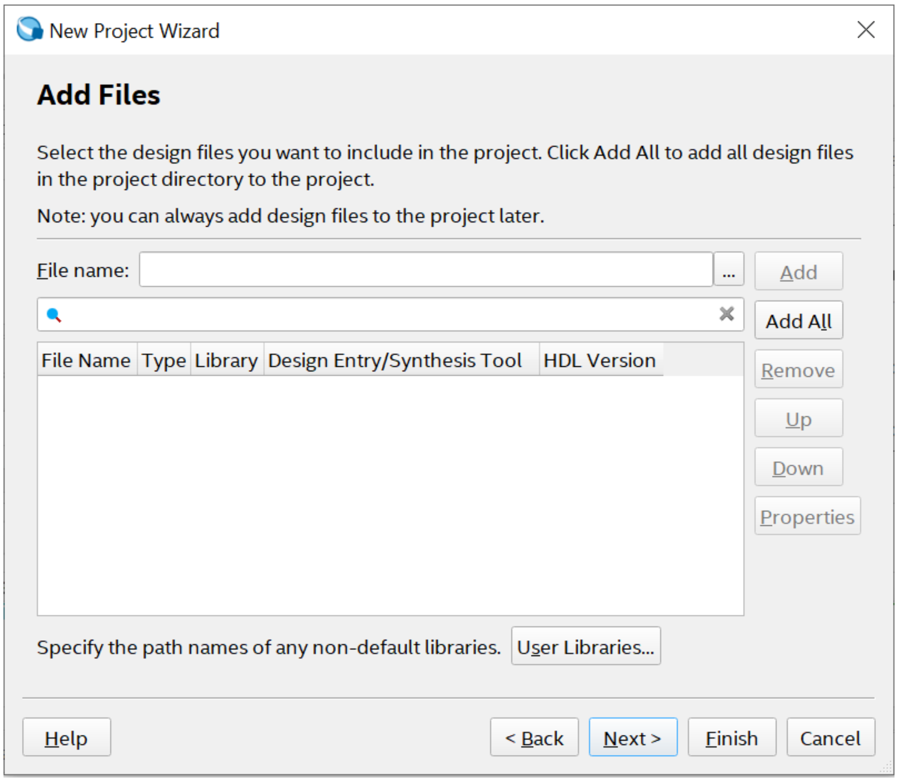
What is the name of this project?

light

What is the name of the top-level design entity for this project? This name is case sensitive and must exactly match the entity name in the design file.

light

Figure 4. Creation of a new project.


New Project Wizard

Add Files

Select the design files you want to include in the project. Click Add All to add all design files in the project directory to the project.

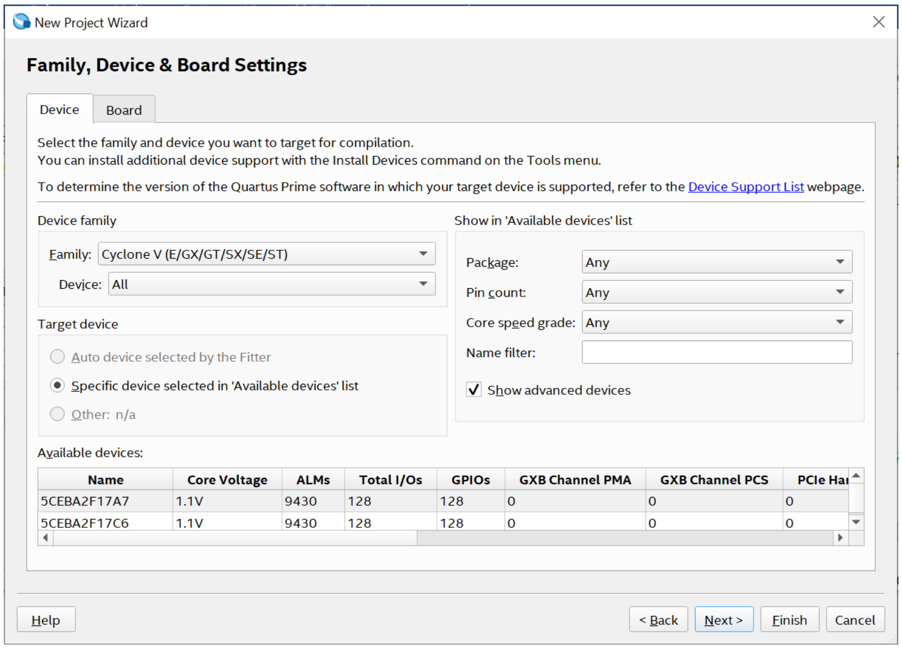
Note: you can always add design files to the project later.

File name:

File Name	Type	Library	Design Entry/Synthesis Tool	HDL Version

Specify the path names of any non-default libraries.

Figure 5. The wizard can include user-specified design files.



Family, Device & Board Settings

Select the family and device you want to target for compilation.
You can install additional device support with the Install Devices command on the Tools menu.

To determine the version of the Quartus Prime software in which your target device is supported, refer to the Device Support List webpage.

Device family

Family: Cyclone V (E/GX/GT/SX/SE/ST)

Device: All

Target device

○ Auto device selected by the Fitter
● Specific device selected in 'Available devices' list
○ Other: n/a

Show in 'Available devices' list

Package: Any

Pin count: Any

Core speed grade: Any

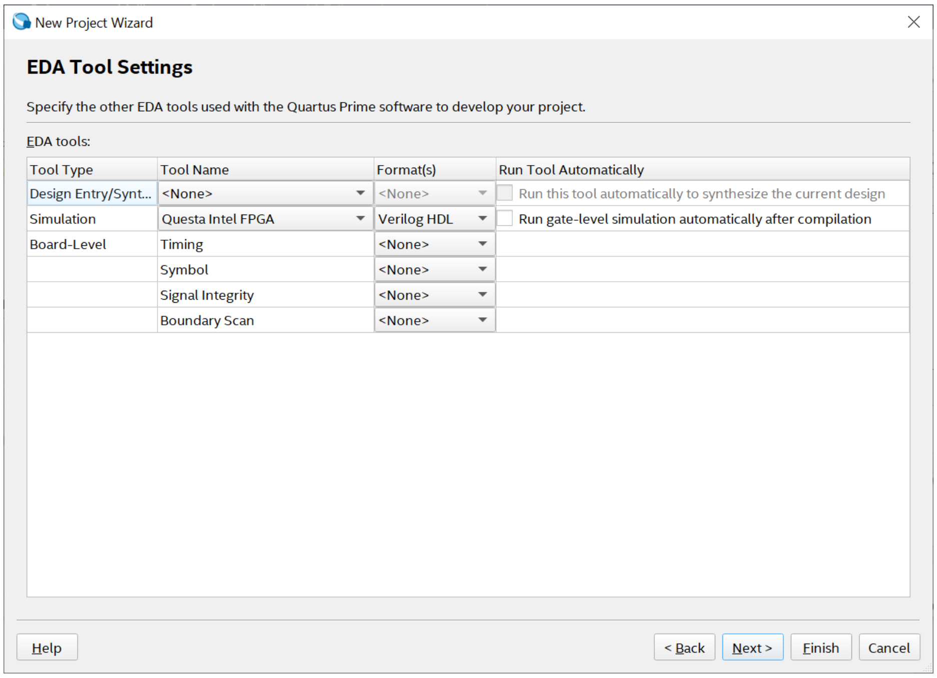
Name filter:

☑ Show advanced devices

Available devices:

Name	Core Voltage	ALMs	Total I/Os	GPIOs	GXB Channel PMA	GXB Channel PCS	PCIe Hai
5CEBA2F17A7	1.1V	9430	128	128	0	0	0
5CEBA2F17C6	1.1V	9430	128	128	0	0	0

Figure 6. Choose the device family and a specific device.



EDA Tool Settings

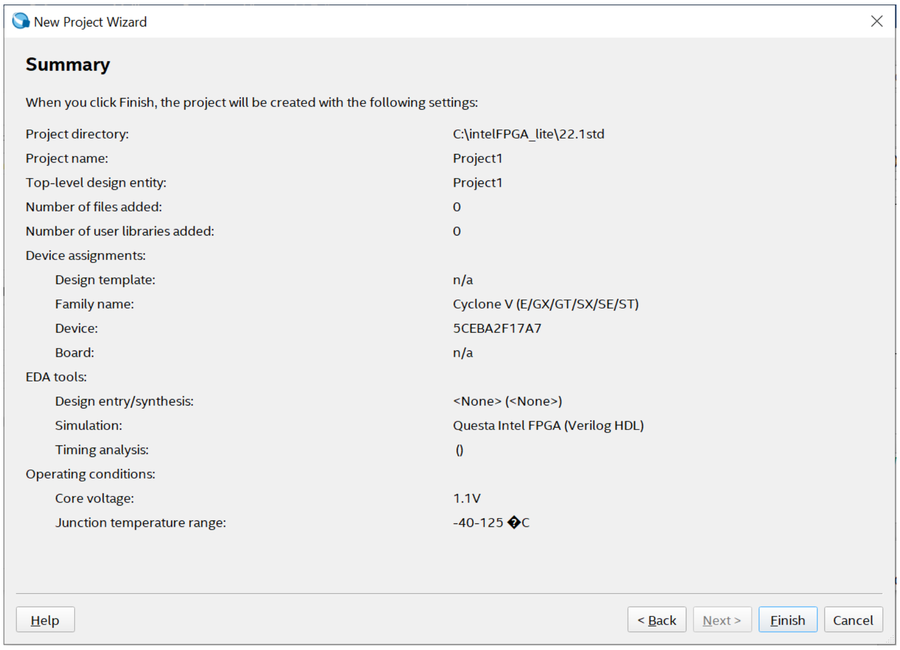
Specify the other EDA tools used with the Quartus Prime software to develop your project.

EDA tools:

Tool Type	Tool Name	Format(s)	Run Tool Automatically
Design Entry/Synt...	<None>	<None>	□ Run this tool automatically to synthesize the current design
Simulation	Questa Intel FPGA	Verilog HDL	□ Run gate-level simulation automatically after compilation
Board-Level	Timing	<None>	
	Symbol	<None>	
	Signal Integrity	<None>	
	Boundary Scan	<None>	

Figure 7. Other EDA tools can be specified.

8. The screen in Figure 8 shows a summary of the chosen setting. Press > **Finish**, which returns to the main Quartus window, but with ***light*** specified as the new project, in the display title bar, as indicated in Figure 9.



New Project Wizard

Summary

When you click Finish, the project will be created with the following settings:

Project directory:	C:\intelFPGA_lite\22.1std
Project name:	Project1
Top-level design entity:	Project1
Number of files added:	0
Number of user libraries added:	0
Device assignments:	
Design template:	n/a
Family name:	Cyclone V (E/GX/GT/SX/SE/ST)
Device:	5CEBA2F17A7
Board:	n/a
EDA tools:	
Design entry/synthesis:	<None> (<None>)
Simulation:	Questa Intel FPGA (Verilog HDL)
Timing analysis:	0
Operating conditions:	
Core voltage:	1.1V
Junction temperature range:	-40-125 �C

Figure 8. Summary of the project settings.

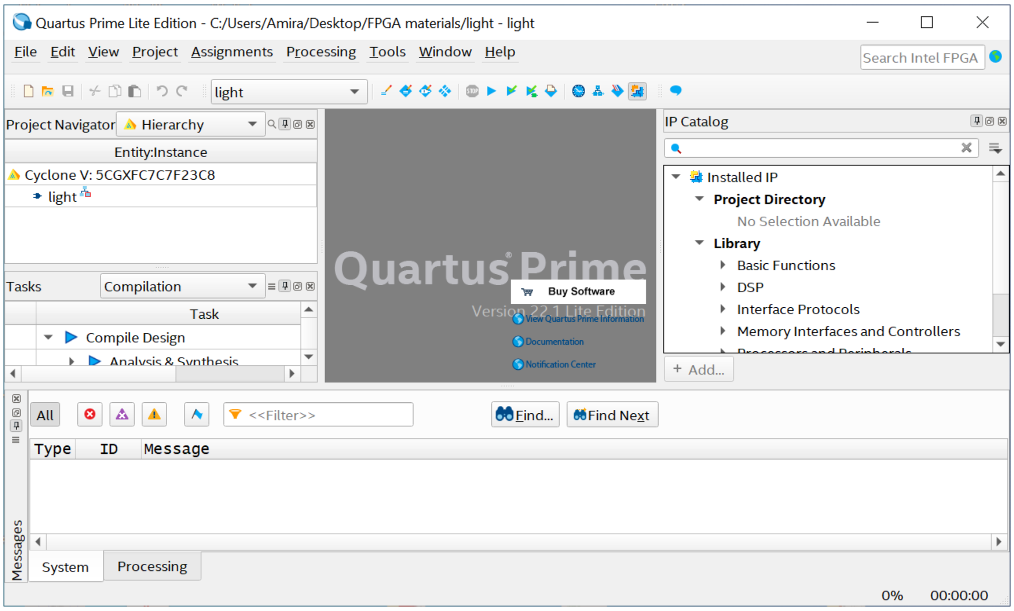
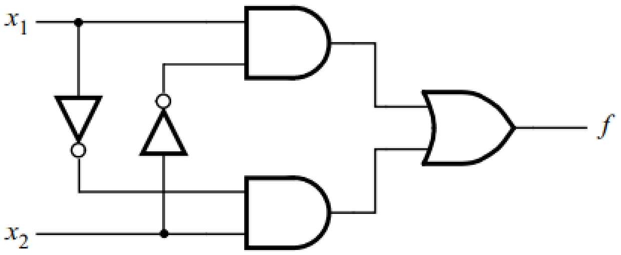


Figure 9. The Quartus Prime display for the created project.

➢ Design Entry Using Verilog Code

A simple design example is presented. A signal light can be controlled from either of the two switches, x_1 and x_2, where a closed switch corresponds to the logic value 1. The circuit and the truth table are given in Figure10. The circuit is an Xor function of the inputs x_1 and , x_2, but we will specify it using the gates shown.



x_1	x_2	f
0	0	0
0	1	1
1	0	1
1	1	0

Figure 10. The light controller circuit and truth table.

Code 1: Verilog code for the circuit in Figure 10

```
module light(x1, x2, f);
 input x1, x2;
 output f;
 assign f = (x1 & ~x2)| (~x1 & x2);
endmodule
```

- **Creating Verilog file Using the Quartus Text Editor**

1. Select **File** > **New** to get the window in Figure 11, choose **Verilog HDL File**, and click > **OK**. This opens the Text Editor window.

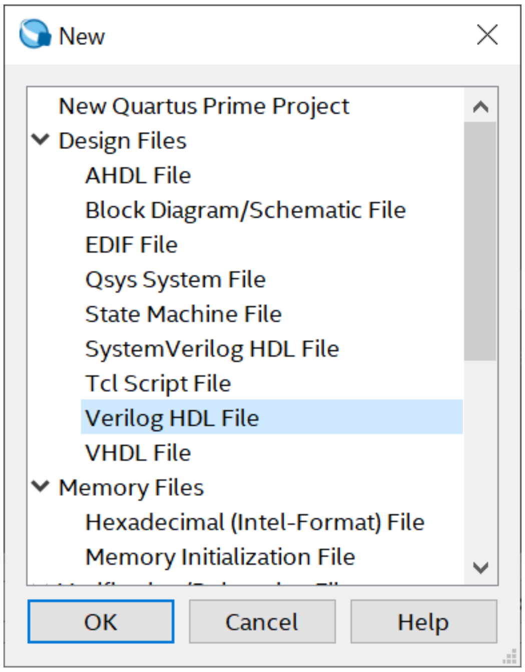
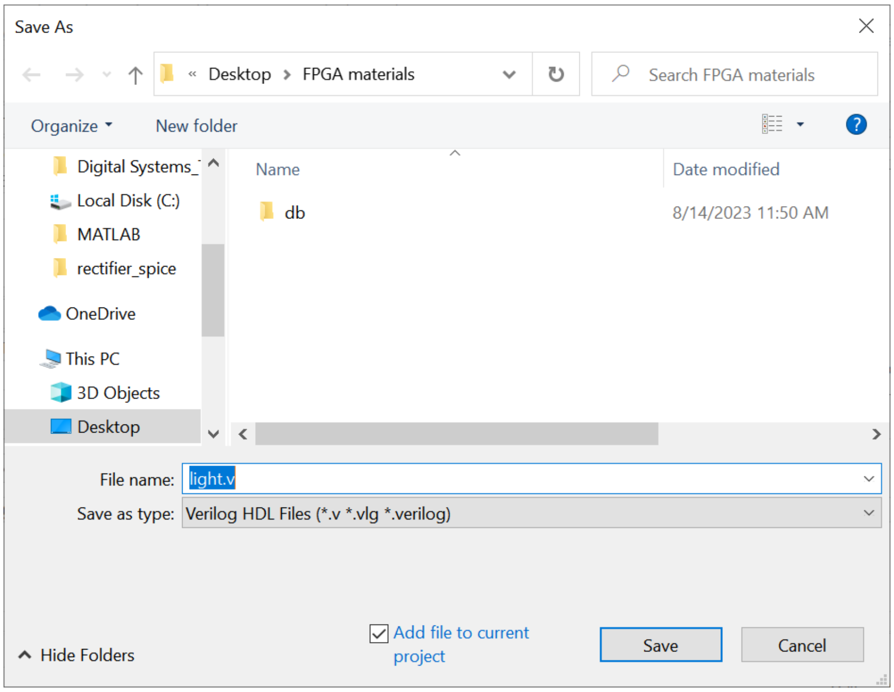


Figure 11. Choose to prepare a Verilog file.

- **To specify a name for the file that will be created.**

  - Select **File** > **Save As** to open the pop-up box depicted in Figure 12.
  - In the box labeled Save as type choose **Verilog HDL File**.
  - In the box labeled File name type light.
  - Put a checkmark in the box Add file to current project. Click **Save**.
  - Maximize the Text Editor window and enter the Verilog code in Figure 13 into it.
  - Save the file by typing **File** > **Save**, or by typing the shortcut Ctrl-s.



File name: light.v

Save as type: Verilog HDL Files (*.v *.vlg *.verilog)

☑ Add file to current project

Figure 12. Name the file.

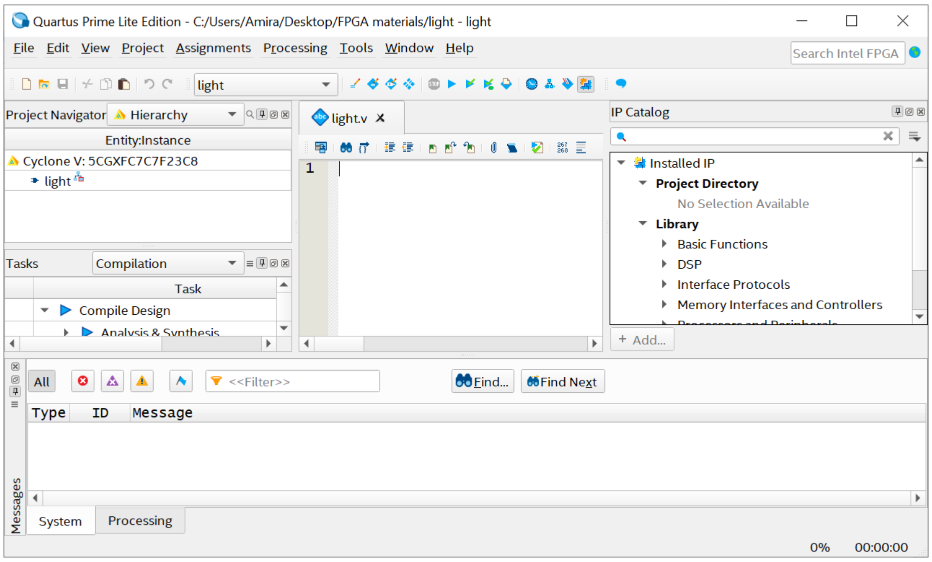
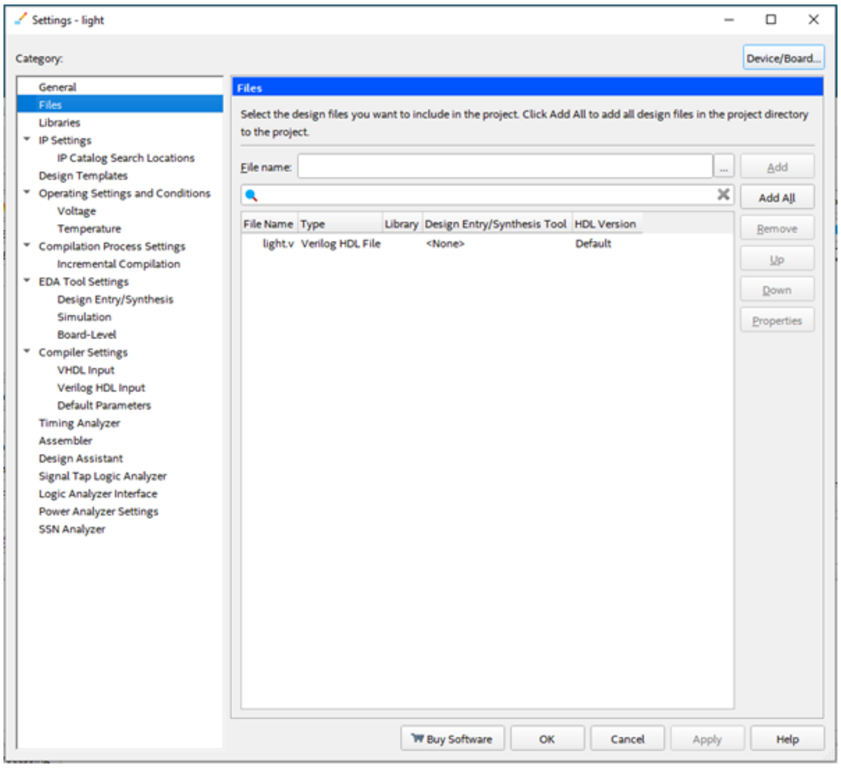


Figure 13. Text Editor window.

➢ Adding Design Files to a Project

As explained before, you can tell Quartus software which design files it should use as part of the current project. To see the list of files already included in the light project,

- select **Assignments** > **Settings**, which leads to the window in Figure 14. As indicated on the left side of the figure, click on the item Files.
- Another way of making this selection is to choose **Project** > **Add/Remove Files** in Project.



Select the design files you want to include in the project. Click Add All to add all design files in the project directory to the project.

File Name	Type	Library	Design Entry/Synthesis Tool	HDL Version
light.v	Verilog HDL File		<None>	Default

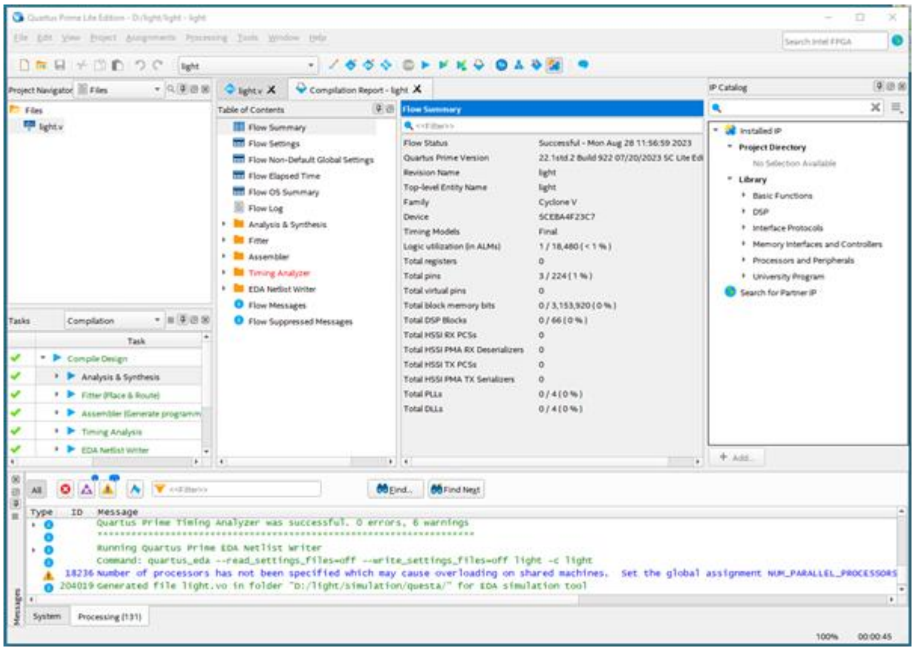
Figure 14. Settings window.

➢ Compiling the Designed Circuit

Several Quartus tools can analyze the Verilog code in the file **light.v**. These tools are controlled by the application program called the Compiler.

To run the Compiler

1. Select **Processing > Start Compilation**, or by clicking on the toolbar icon that looks like a blue triangle.
2. As the compilation moves through various stages, its progress is reported in a window on the left side of the Quartus display.
3. Successful (or unsuccessful) compilation is indicated in a pop-up box. Acknowledge it by clicking OK, which leads to the Quartus display in Figure 15. In the message window, at the bottom of the figure, various messages are displayed. In case of errors, there will be appropriate messages given. When the compilation is finished, a compilation report is produced. A window showing this report opens automatically, as seen in Figure 15.



Flow Summary	
Flow Status	Successful - Mon Aug 28 11:56:59 2023
Quartus Prime Version	22.1std.2 Build 922 07/20/2023 SC Lite Edi
Revision Name	light
Top-level Entity Name	light
Family	Cyclone V
Device	5CEBA4F23C7
Timing Models	Final
Logic utilization (in ALMs)	1 / 18,480 (< 1 %)
Total registers	0
Total pins	3 / 224 (1 %)
Total virtual pins	0
Total block memory bits	0 / 3,153,920 (0 %)
Total DSP Blocks	0 / 66 (0 %)
Total HSSI RX PCSs	0
Total HSSI PMA RX Deserializers	0
Total HSSI TX PCSs	0
Total HSSI PMA TX Serializers	0
Total PLLs	0 / 4 (0 %)
Total DLLs	0 / 4 (0 %)

```
Type   ID      Message
               Quartus Prime Timing Analyzer was successful. 0 errors, 6 warnings
               ******************************************************************
               Running Quartus Prime EDA Netlist Writer
               Command: quartus_eda --read_settings_files=off --write_settings_files=off light -c light
       18236   Number of processors has not been specified which may cause overloading on shared machines.  Set the global assignment NUM_PARALLEL_PROCESSORS
       204019  Generated file light.vo in folder "D:/light/simulation/questa/" for EDA simulation tool
```

100%   00:00:45

Figure 15. Display after a successful compilation.

The compilation report can be opened at any time either by selecting Processing > Compilation Report or by clicking on the icon. The report includes several sections listed on the left side of its window. Figure 16 displays the Compiler Flow Summary section, which indicates that only one logic element and three pins are needed to implement this tiny circuit on the selected FPGA chip.

➢ Pin Assignment

In the compilation steps above, the Quartus Prime Compiler was free to choose any pins on the selected FPGA to serve as inputs and outputs. However, the DE0 board has hardwired connections between the FPGA pins and the other components on the board. We will use two push buttons, labeled KEY0 and KEY1, to provide the external inputs, x1, and x2, to our example circuit. These buttons are connected to the FPGA pins PIN_U7 and PIN_W9, respectively. We will connect the output f to the red light-emitting diode labeled LEDR0, which is hardwired to the FPGA PIN_ AA2.

➢ Pin assignments are made by using the Assignment Editor as follow:

1. Select **Assignments > Pin Planner** to reach the window in Figure 16.

2. In Figure 17, Click on x1 as the first pin to be assigned; this will enter x1 in the displayed table. Follow this by double-clicking on the box to the right of this new x1 entry, in the column labeled Location.
3. Now, you can just type the name of the pin in the Location box. Use the same procedure to assign input x2 to PIN_W9 and output f to pin AA2, which results in the image in Figure 18. To save the assignments made, choose **File > Save**.
4. You can also close the Assignment Editor window, in which case a pop-up box will ask if you want to save the changes to assignments; click **Yes**. Recompile the circuit so that it will be compiled with the correct pin assignments.

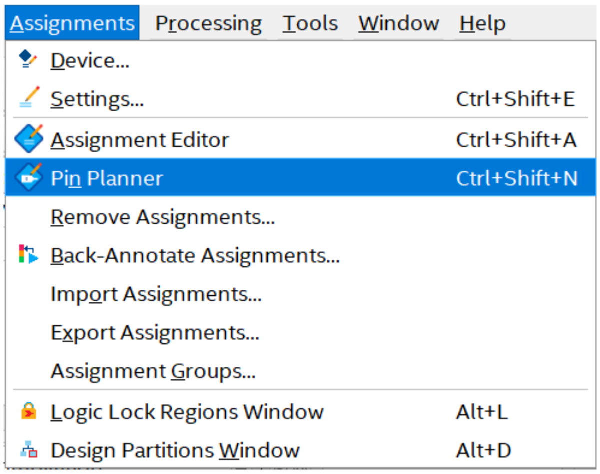
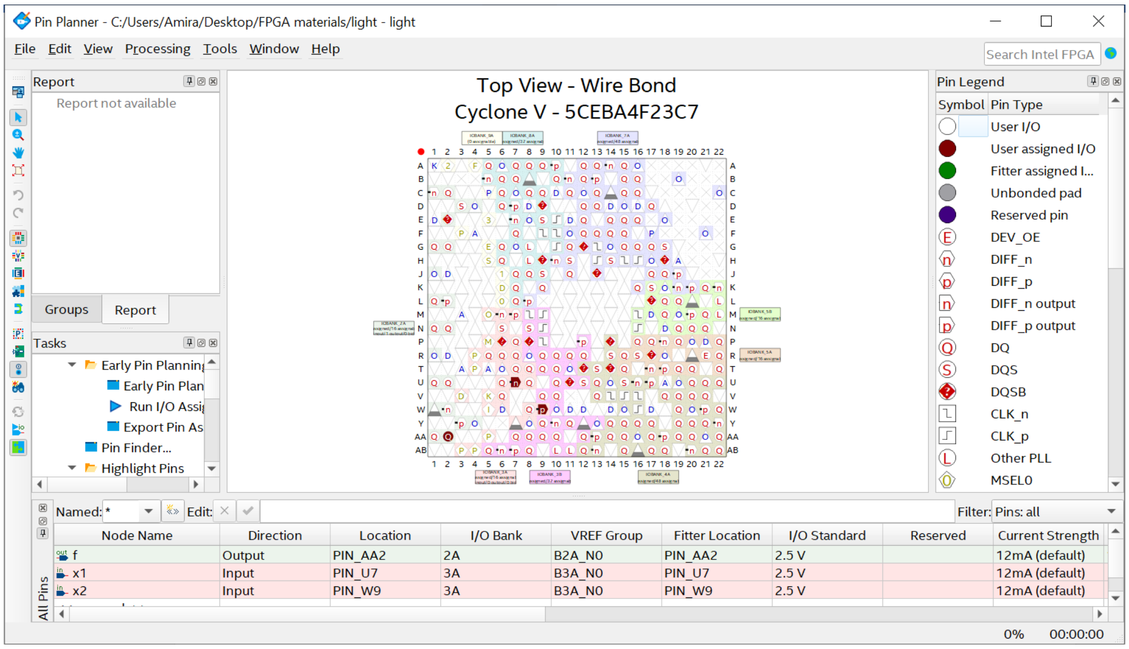


Figure 16. The Assignment Editor window.



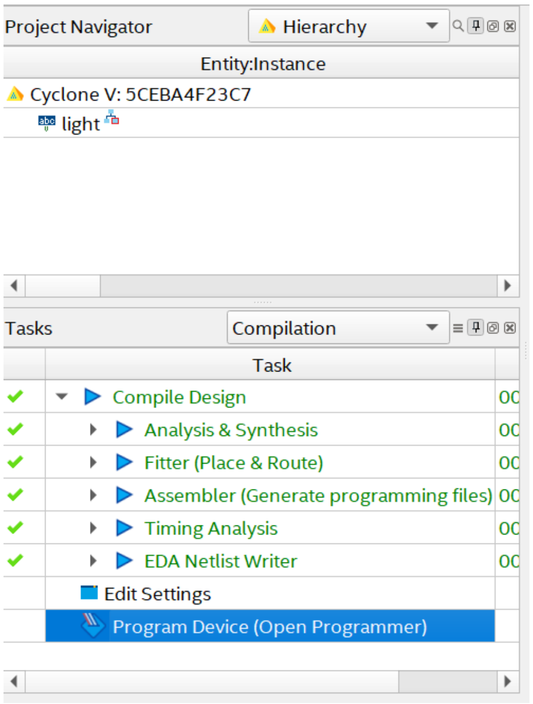
Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard	Reserved	Current Strength
f	Output	PIN_AA2	2A	B2A_N0	PIN_AA2	2.5 V		12mA (default)
x1	Input	PIN_U7	3A	B3A_N0	PIN_U7	2.5 V		12mA (default)
x2	Input	PIN_W9	3A	B3A_N0	PIN_W9	2.5 V		12mA (default)

Figure 17. The complete assignment.

Programming the FPGA

The following steps illustrate the configuration of the FPGA development board.

1. Select **Program Device** from **Task Menu**, which leads to the window in Figure 19.



Project Navigator — Hierarchy

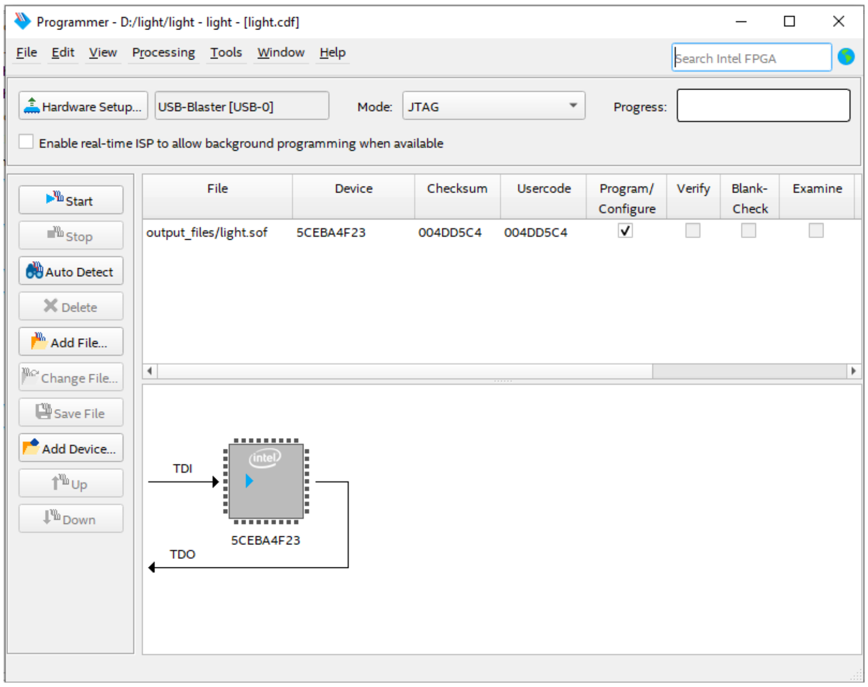
Entity:Instance

- Cyclone V: 5CEBA4F23C7
  - light

Tasks — Compilation

	Task	
✓	Compile Design	00
✓	Analysis & Synthesis	00
✓	Fitter (Place & Route)	00
✓	Assembler (Generate programming files)	00
✓	Timing Analysis	00
✓	EDA Netlist Writer	00
	Edit Settings	
	Program Device (Open Programmer)	

Figure 18. The device programming.



Programmer - D:/light/light - light - [light.cdf]

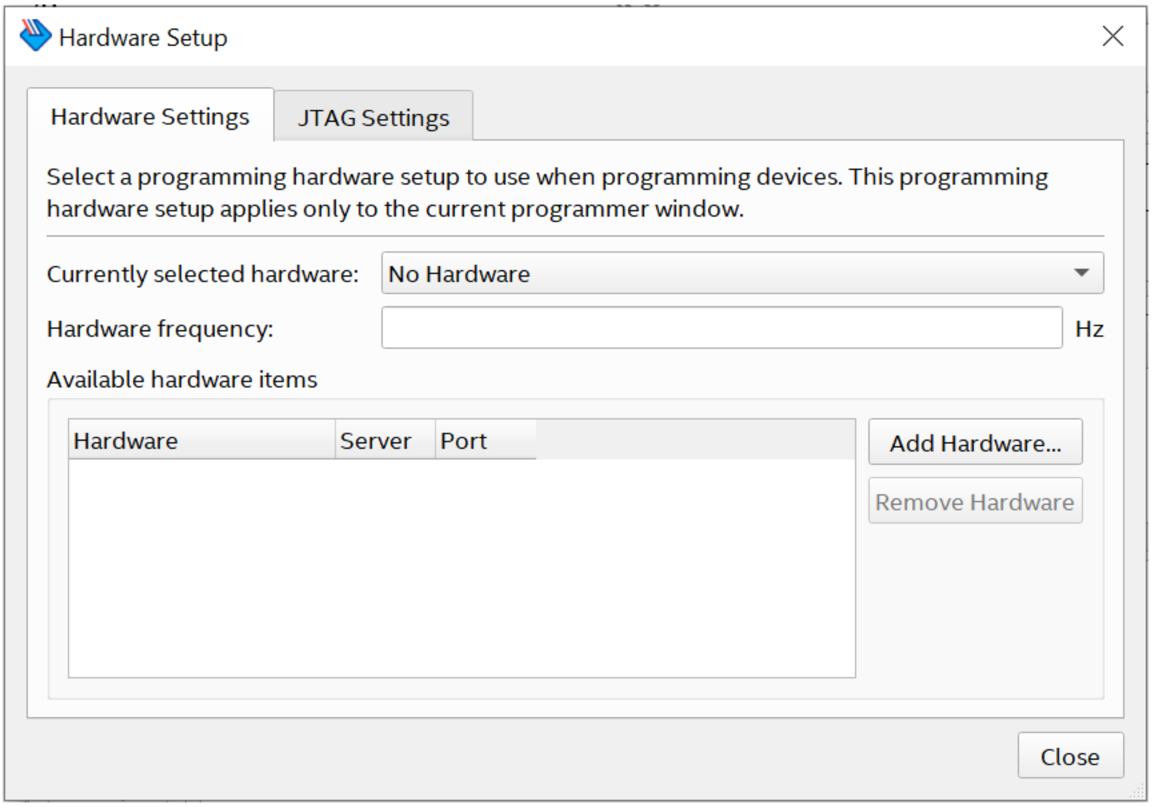
USB-Blaster [USB-0] Mode: JTAG Progress:

□ Enable real-time ISP to allow background programming when available

File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank- Check	Examine
output_files/light.sof	5CEBA4F23	004DD5C4	004DD5C4	☑	□	□	□

Figure 19. FPGA detected in Quartus programmer

2. Select from the **Programmer Window** > **Hardware Setup**, which leads you to the window in Figure 20.



Select a programming hardware setup to use when programming devices. This programming hardware setup applies only to the current programmer window.

Currently selected hardware: No Hardware

Hardware frequency: Hz

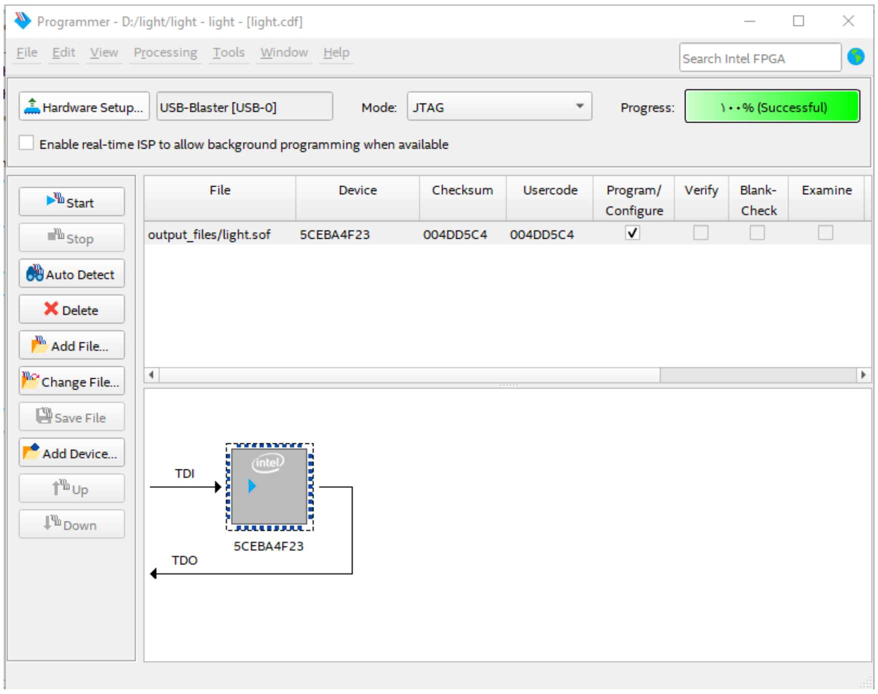
Available hardware items

Hardware	Server	Port

Figure 20. Hardware setup.

3. Select from **Currently selected hardware** > **USB-Blaster** to allow the Quartus Prime programmer to the detect FPGA device.

4. Then, Select **Start** to program the FPGA device.



Programmer - D:/light/light - light - [light.cdf]

Hardware Setup... USB-Blaster [USB-0]

Mode: JTAG

Progress: ١٠٠% (Successful)

□ Enable real-time ISP to allow background programming when available

File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank-Check	Examine
output_files/light.sof	5CEBA4F23	004DD5C4	004DD5C4	☑	□	□	□

Figure 21. Programming success.