UNIVERSITI TEKNOLOGI MALAYSIA

Introduction to the EPM240 Board

Muhammad Mun'im Ahmad Zabidi, Musa Mohd Mokji, Izam Kamisian, Norhafizah Ramli

Abstract
This article introduces the EPM240 board for low-cost digital experimentation.

[1] *Department of Electronic and Computer Engineering, School of Electrical Engineering, Faculty of Engineering*
*__Corresponding author__: munim@utm.my

1. FPGA vs CPLD

Field-Programmable Devices (FPDs) are devices that are programmed by electronically connecting and disconnecting internal logic elements. This differs from microcontroller and microprocessor which are programmed by changing the list of instructions. This enable FPDs perform tasks much faster while consuming about 10x less power.

There are several types of PLDs, those can be classified under SPLD, CPLD and FPGA. The SPLD and CPLD have the same basic design but differ due the keyword 'simple' and 'complex'. SPLD has only one logic element (LE) consisting of an array of AND, OR and NOT logic gates. The simplest CPLD from Altera has 240 LEs, and the next device has 570 LEs. That makes CPLD much more powerful than SPLD.

FPGAs have different internal architecture and working mechanism compared to CPLD. Whereas CPLD a relatively small number of complex LEs, the FPGA has a huge number of simple Configurable Logic Blocks (CLBs). This allows high end FPGAs to contain millions of equivalent gates compared to just thousands for the most complex CPLD.

Despite their differences, CPLDs and FPGAs are programmed using the same design tools. For devices made by Altera, the Quartus software is used for both low-end MAX CPLD, the high end Stratix FPGA, and all devices in between. Therefore, all knowledge you gain while using the MAX will be transferable when you upgrade to more sophisticated FPGA later.

2. The EPM240 aka MAX II

The EPM240 mini board is ideal for low budget digital hardware experimentation. In the middle sits the Altera MAX II EPM240T100C5N CPLD. The board has all the necessary auxiliary circuits allow the CPLD to be programmed through Quartus II. The schematics diagram of this board is available in open sources.

The board has 100 general purpose input/output (GPIO) pins accessible through the two (2) units of 22 × 2 headers

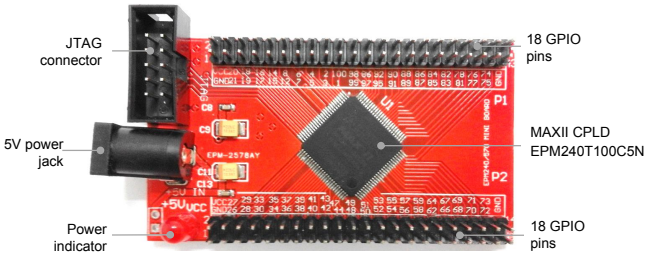


Figure 1. *EPM240 board top view.*

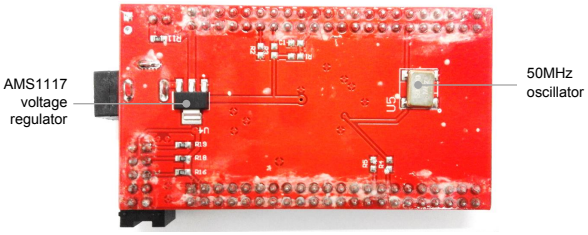


Figure 2. *EPM240 board bottom view.*

as shown Fig. 1 and Fig. 2. For experiments, a breadboard is connected to these pins.

Features:

- JTAG connector
- Can be easily programmed using the USB Blaster
- 3.3 V operating voltage by using the AMS1117 3.3V voltage regulator circuit
- Flexible I/O interface supporting 3.3 V, 2.5 V, 1.8 V, and 1.5 V logic levels
- High speed operation, up to 150 MHz
- InstantOn, non-volatile storage
- 240 logic elements
- On board power regulator (works with 5 V power adapters)
- On-board 50 MHz oscillator

3. Installing the Driver

The EPM240 board requires a USB Blaster for programming. The computer will recognize the new hardware connected to its USB port, but it will be unable to proceed if it does not have the required driver already installed.

Figure 3. *USB blaster.*

If the USB Blaster driver is not already installed, launch Device Manager.

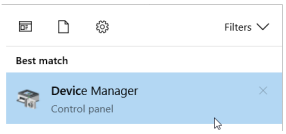


Figure 4. *Launch Device Manager.*

In Device Manager, find the new hardware under Other devices.

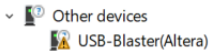


Figure 5. *Device Manager -> Other devices.*

The driver available within the Quartus II software. Select *Browse my computer for driver software.*

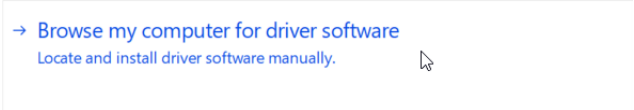


Figure 6. *The driver is not online.*

Select Update Driver from the dialog box that pops up.

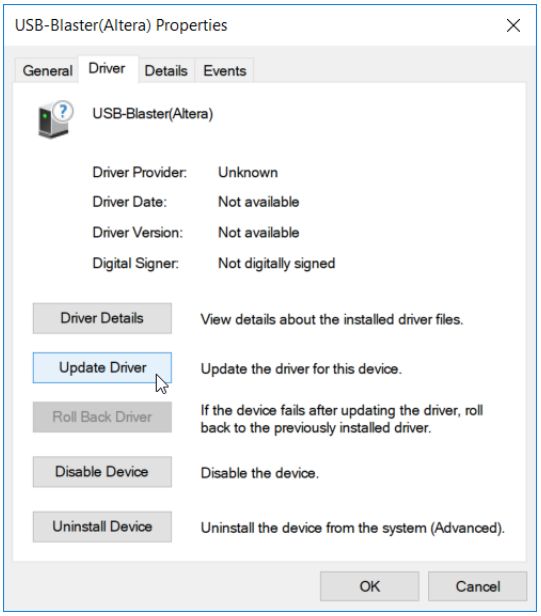


Figure 7. *Update the driver.*

Specify the driver location in the following dialog box:

```
C:\altera\version\quartus\
```

Replace `altera` with `IntelFPGA` for some of the newer Quartus versions.

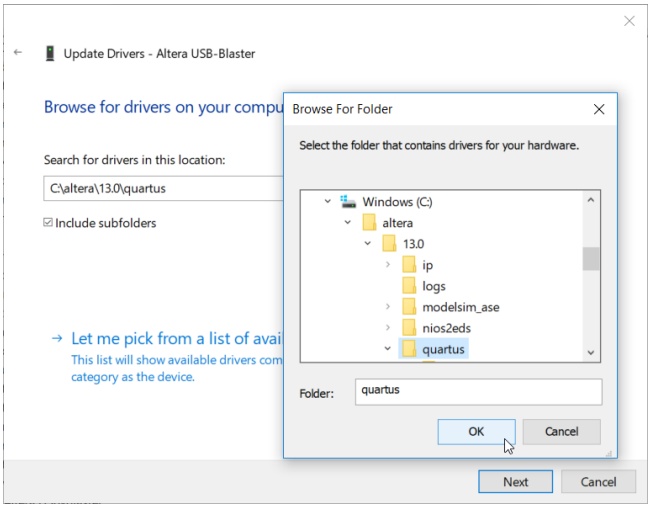


Figure 8. *Specify the driver location.*

The driver will now be installed as indicated in Figure 9. You can now use the EPM240 board.

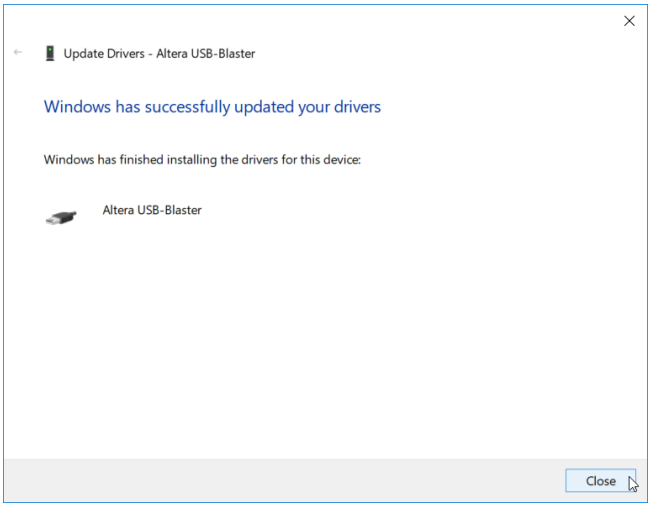


Figure 9. *The driver is installed.*

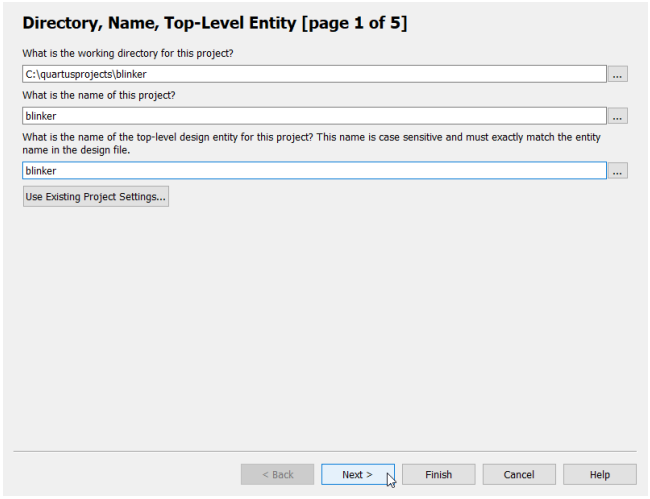
Next, we check out the software and hardware installation whether they are working properly. To do this, we will create a project, enter a short Verilog code, compile it, then download the code to the CPLD.

4. Project Settings

Run Quartus-II Web Edition and select the **File ➥ New Project Wizard** menu.

Page 1

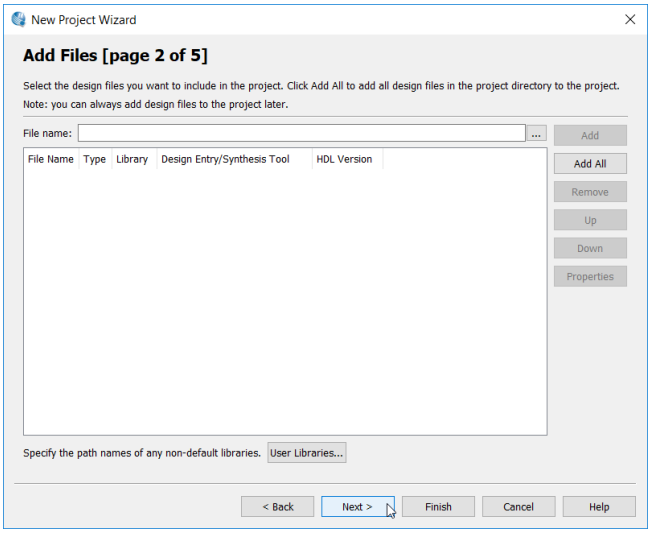
This page sets the location of your project and the name of your project. It is highly recommended to change the directory outside the Quartus software directory. This will cause less problems later if you have to reinstall Quartus. For example, you can create a folder in your home directory called *Documents/QuartusProjects*. Inside this folder, create your project folder for example *Documents/QuartusProjects/blinker*.



After setting the directory, project name and top-level entity, click Next >.

Page 2

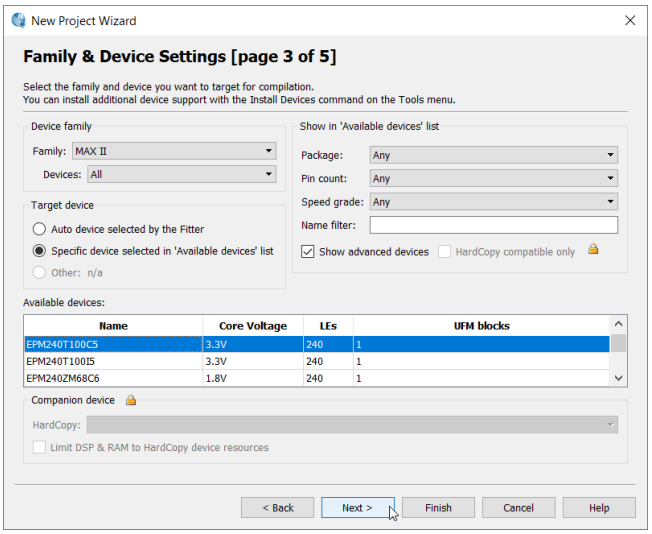
This page lets you add files from another project.



Since this is your first project, there is nothing to add. Just click Next >.

Page 3

This page selects the device we are using.

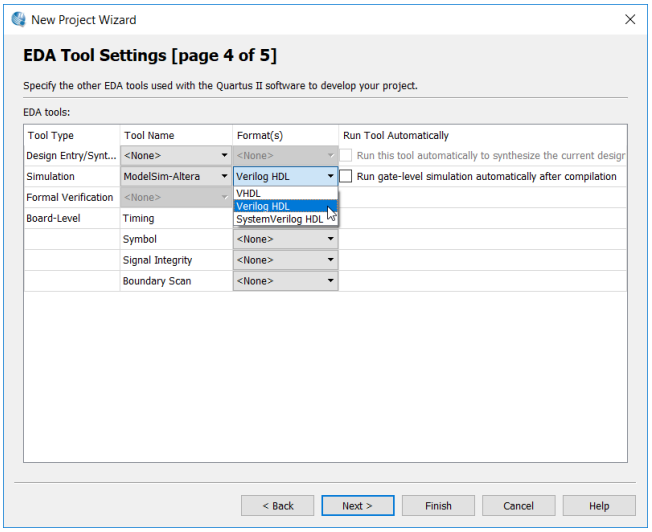


- For device family, select **MAX II**.
- To narrow down the device list, select Pin count **100**. This will shorten the list so that you can find the exact device faster. Or you can simply scroll through the long list.
- In the device list, select **EPM240T100C5**.

When you have selected the correct device, click Finish. Optionally, you can click Next > to go to Page 4.

Page 4 (optional)

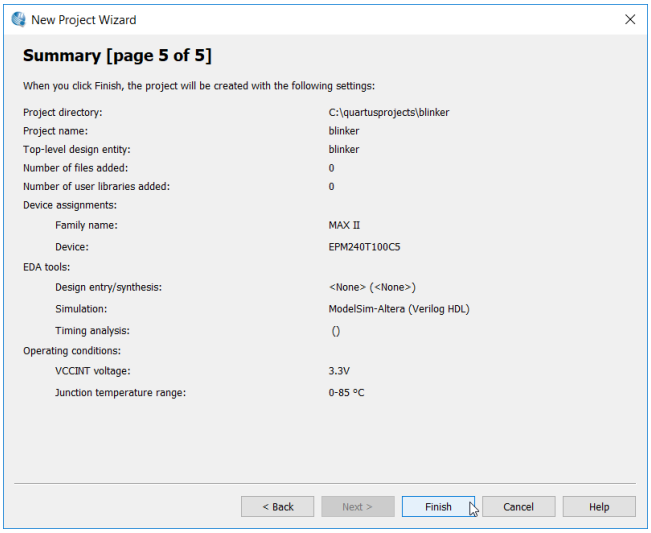
This page sets the EDA Settings.



Simply make sure the check box is OFF then click Next >.

Page 5

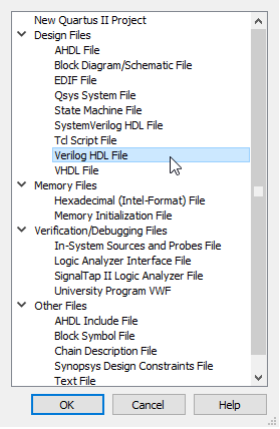
This page summarizes all settings.



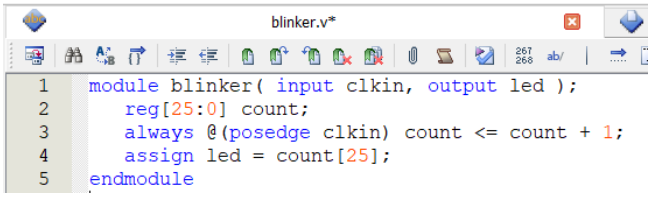
Verify the device assignment, then click Finish.

5. Blinker Source Code

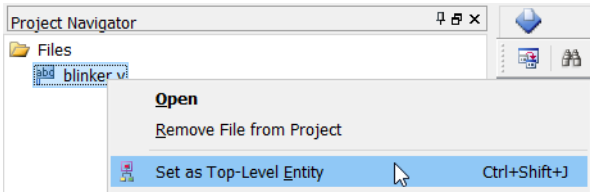
From the top menu, choose **File ➥ New**. In the pop-up dialog box, select **Verilog HDL File**:



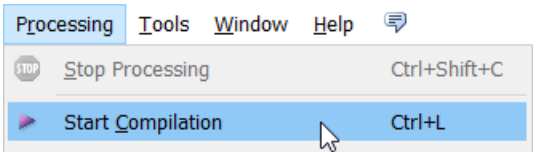
Then key in this code as is (case sensitive):



Save it as **blinker.v** then at the Project Navigator set the code as Top level entity.

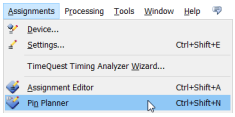


Click on the **Processing ➥ Start Compilation** menu. If you found errors, correct the mistake and recompile. You can ignore warnings.

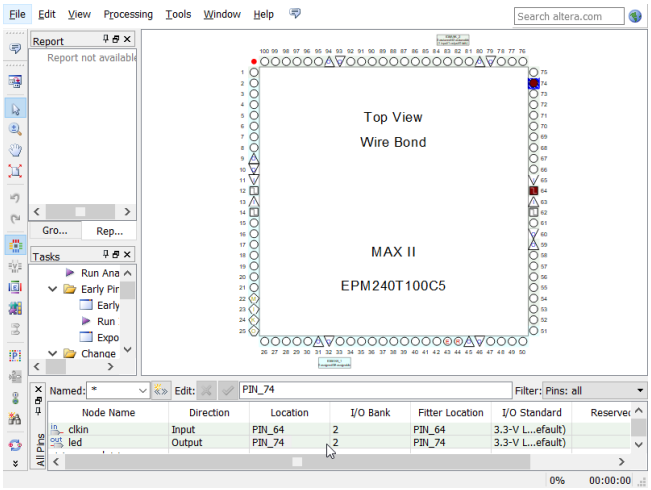


6. Pin Assignment

From the top menu, choose **Assignments ➥ Pin Planner** menu:



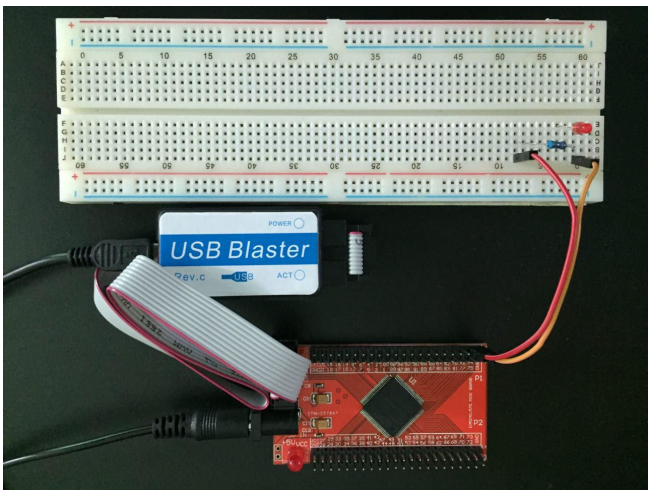
In the dialog box, assign `led` to Pin 74 and `clkin` to Pin 64.



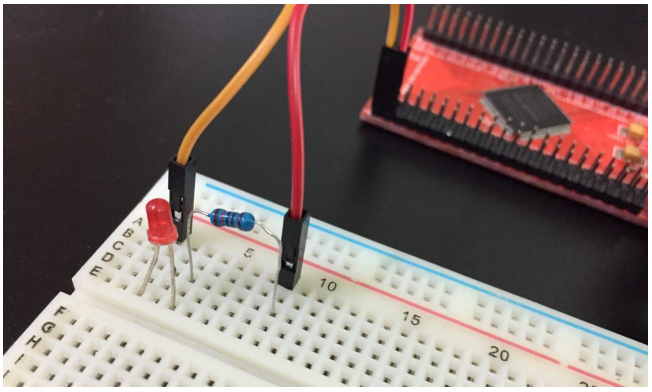
After setting the pins, you must recompile to generate the FPGA programming file. The design is ready to be downloaded into the CPLD.

7. Wiring

Connect pin 74 on EPM240 board to a resistor. Use a resistor with a value anywhere from 100 Ω to 470 Ω. Connect the resistor to the positive terminal of an LED. Connect ground pin to the ground of the LED board.

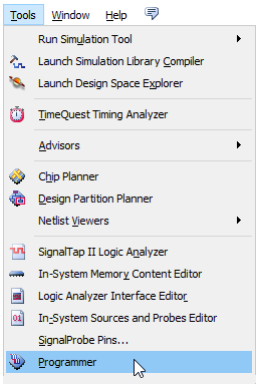


Connect USB Blaster to PC and EPM240 board. Connect power.



8. Programming

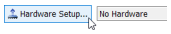
From the top menu, select **Tools ➥ Programmer**.



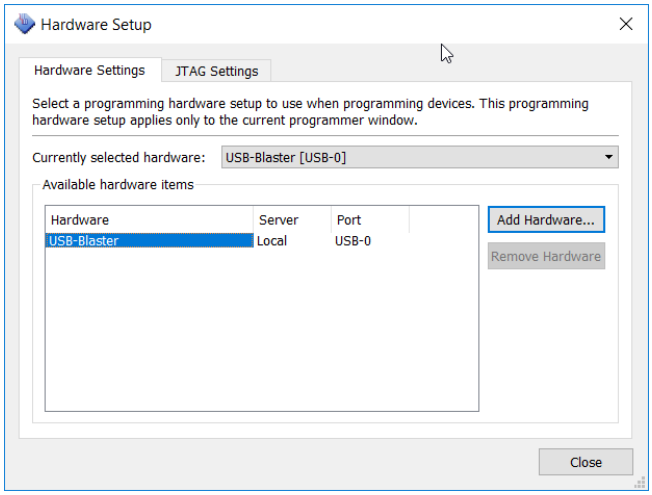
Hardware Setup

Do this step once to setup USB Blaster for programming.

Select Hardware Setup....

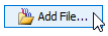


In the dialog box that follows, select USB Blaster.

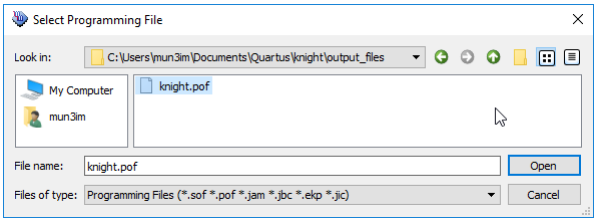


Proceed to Programming

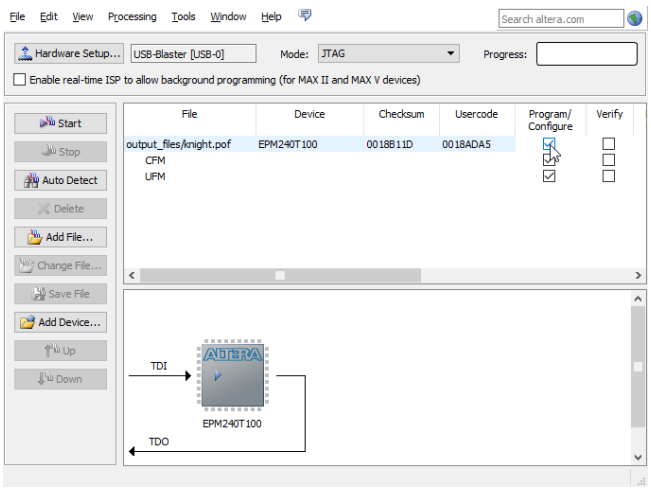
Click Add File...



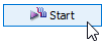
Navigate to **output_files** directory under your project. Select the POF file.



Check the Program/Configure buttons.



Then click Start to program the chip.



If everything goes well, the LED should blink once about every second.

Acknowledgments

Thanks to Siti Nursyuhada binti Mahsahirun and Zulkifli Md. Yusof, both of Faculty of Manufacturing, Universiti Malaysia Pahang for the original idea.

References

[1] Munim Zabidi, Izam Kamisian, and Ismahani Ismail. *The Art of Digital Design*. 2019.

[2] *MAX II Device Handbook*. Altera, Oct. 2008. URL: https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/hb/max2/max2_mii5v1.pdf.

[3] *Introduction to the Quartus® II Software*. Version 10. Altera. 2010. URL: https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/manual/intro_to_quartus2.pdf.

[4] *Intel FPGA USB Download Cable User Guide*. Intel. Oct. 2016. URL: https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/ug/ug_usb_blstr.pdf.

[5] *My First FPGA Design Tutorial*. Altera. July 2008. URL: https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/tt/tt_my_first_fpga.pdf.