Quartus New Project Tutorial

This tutorial guides you through the typical steps of design, simulation, synthesis and implementation of a simple project, designed from scratch, as those that you will do at the Applications for the Digital Integrated Circuits course.

Contents

oject File Management2
eate a New Project
eate Source Files5
st Time Analysis7
t up Simulation
art Simulation12
n Assignment14
mpilation14
vice Programming15
uick Project Guide
t up Simulation

Project File Management

The normal design flow goes through all stages in Figure 1. The Quartus II software arranges these stages in the *Compilation Flow* and guides you through it.

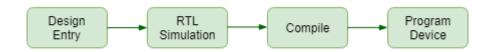


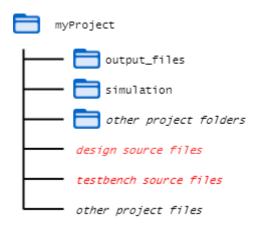
Figure 1: Quartus II Compilation Flow

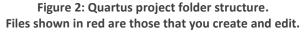
Various files are employed in each design flow stage, all of them created and managed by Quartus, except for the source files, that are written by you, the designer. All files created by Quartus reside in the project folder (directory) and its subfolders. Your source files should also be saved inside the project folder.

To keep things as neat as possible, strictly follow these two rules:

- Each project has its own folder.
- All (Verilog) source files are saved in the project folder.

If the above rules are obeyed, the project folder should have the structure shown in Figure 2.





Most of the files are taken care of by Quartus, so you need not bother about their names or where they are located. The only files you should care about are:

- design source files
- testbench source files

Remember:

• All design source files and testbench source files should be saved in the project folder.

Create a New Project

Open the New Project Wizard either by clicking on its icon in the start window, or from the menu File -> New Project Wizard ..., or, also from the menu, File -> New ... -> New Quartus II Project.

In step 1 [Directory, Name, Top-Level Entity] of the New Project Wizard (Figure 3) set the name of the project folder (the working directory), the name of the project, and the name of the top-level design module. First, select a folder where you want your project's folder to be created (for example the folder student), then type a slash (/), followed by the name you choose for your project folder (app1 for example).

New Project Wi	zard
Directory, Name, Top-Level Entity [page	1 of 5]
What is the working directory for this project?	
/home/student/app1	
What is the name of this project?	
app1	
What is the name of the top-level design entity for this project? match the entity name in the design file.	This name is case sensitive and must exactly
sum	
Use Existing Project Settings	
Help	< Back Next > Finish Cancel

Figure 3: New Project Wizard step 1 (name and folder setup)

At this stage, the project is empty and no file is added. Skip this step and click Next.

In step 3 [Family & Device Settings] of the New Project Wizard select the 5CSEMA5F31C6 device from the list (Figure 4). Then click **Next**.

New Project Wizard 🧟								
Family & D	Family & Device Settings [page 3 of 5]							
You can install ac		oort with th	e Install Devic	es co	ommand on the Tools i arget device is suppor	menu. 'ted, refer to the <u>Device Su</u>	i <u>pport List</u> webpage.	
Device family					Show in 'Available o	devices' list		
<u>F</u> amily: Cyclon	e V (E/GX/GT/SX/SE/	ST)		\$	Pac <u>k</u> age:	Any	\$	
Devices: All				\$	Pin <u>c</u> ount:	Any	+	
Target device				Core Sp <u>e</u> ed grade:	Any	\$		
○ <u>A</u> uto device	selected by the Fitte	r			Name filter:	5csema5f31c		
Specific device selected in 'Available devices' list Other: n/a					devices			
A <u>v</u> ailable devices	:							
Name	Core Voltage	ALMs	User I/Os	5	GXB Channel PMA	GXB Channel PCS	PCIe (PIPE) Hi	
5CSEMA5F31C6	1.1V	32070	457	C)	0	0	
5CSEMA5F31C7	1.1V	32070	457	C)	0	0	
5CSEMA5F31C8	1.1V	32070	457	C)	0	0	
•								
Help						< <u>B</u> ack <u>N</u> ext >	Einish Cancel	

Figure 4: Device selection

In New Project Wizard step 4 [EDA Tool Settings], change the Format for Simulation from **VHDL** to **Verilog HDL** in the drop-down list (as in Figure 5), then click **Next**.

EDA Tool Se	ettings [page	e 4	of 5]		
Specify the other	EDA tools used with	n the	e Quartus II softwar	e to	develop your project.
EDA tools:					
Tool Type	Tool Name		Format(s)	1	Run Tool Automatically
Design Entry/S	<none></none>	\$	<none></none>	\$	Run this tool automatically to synthesize the o
Simulation	ModelSim-Altera	\$	VHDL		Run gate-level simulation automatically after
Formal Verific	<none></none>	\$	Verilog HDL	N	
Board-Level	Timing		SystemVerilog HI	DF4	
	Symbol		<none></none>	\$	
	Signal Integrity		<none></none>	\$	
	Boundary Scan		<none></none>	\$	

Figure 5: Choose Format for Simulation

Click Finish to close the New Project Wizard.

Create Source Files

To create a new design source file, select from the menu File -> New (Figure 6).

<u>F</u> ile	<u>E</u> dit <u>V</u> iew	<u>P</u> roject	<u>A</u> ssignments	Processing Tools
	<u>N</u> ev <mark>y</mark>		Ctrl+N	
1	Open		Ctrl+O	
	<u>C</u> lose		Ctrl+F4	
1	New Project <u>W</u>	izard		
1	Open P <u>r</u> oject		Ctrl+J	
	C D			

Figure 6: Open a new file

In the popup window that opens (Figure 7), select Verilog HDL Files and click OK.

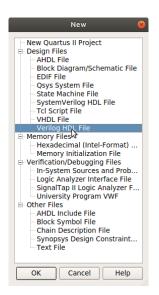


Figure 7: New file type selection

All source files that you will write during Applications, be they design source files, or testbench source files, will be of Verilog HDL type.

After you have clicked **OK**, the file opens in the Editor panel, with a default name, Verilog1.v (or Verilog2.v, or other automatically numbered default name). It is recommended that you immediately save the file with a suitable name. From the menu select **File** -> **Save As ...** (Figure 8).

<u>File Edit View</u>	Project Assignments	ProcessingToolsIndow
<u>N</u> ew	Ctrl+N	÷ 🕱 🖌 🧳 🕹 💿 🕨
Open	Ctrl+O	Veriloq1.v*
Close	Ctrl+F4	VCINOGI.V
New Project <u>Wiza</u> Open Project Save Project Clos <u>e</u> Project	rd Ctrl+J	- 人。 (7) 注 章 章 ① ① ① ① ① ② ◎ 章 dule sum(input a, input b, output [1:0] c
🛃 <u>S</u> ave	Ctrl+S	sign c = a + b;
Save <u>A</u> s		dmodule
🗿 Save All 😽	Ctrl+Shift+S	

Figure 8: First save of a source file

A file browser opens (Figure 9). If you followed strictly all previous steps as recommended, the browser will open showing the content of the project folder. You need only to type the desired name for the source file, then click on **Save**, and the file will be saved in the project folder.

Save As 😣	
Look in: 📄/home/student/app1 🔹 🔇 🛇 🖗 🔃 🗐	
Computer db	
File name: Sum.v	type a name and
Save as type: Verilog HDL Files (*.v *.vlg *.verilog) ✓ Add file to current project	save

Figure 9: Choose a filename for a source file

Note: If the browser opens in some other folder, browse through the folders and select the project folder. You should see the project folder path name (for example /home/student/app1 if you setup the project folder to be app1 in the student home directory) in the **Look in** field at the top of the Save As window.

Keep in mind these rules:

- Each module is in a separate source file.
- The name of the source file is the name of the module.
- Save any source file in the project folder.
- The top-level module name and its source file name must match the names you chose when the project was setup.

First Time Analysis

This step is necessary in order to be able to start the simulator from Quartus. When you do this first time, it's recommended that the design is at a minimum, such that Analysis quickly finishes and passes without errors, or if there are errors, they are easy to fix. Once the simulator starts, you may play with the source file code without the need to run through Analysis every time you change something.

In the Tasks panel switch the Flow to RTL Simulation (Figure 10):

Tasks	5	₽ ⊘ 🗶
Flow:	Gate Level Simulation	ze
	Compilation	
	RTL Simulation	
	Full Design	2)
	🗉 🕨 TimeQuest Timin	g Analy:
	🕀 🕨 EDA Netlist Write	r
	A Non-Array and Characterized	- - 1

Figure 10: Choose the flow.

The RTL Simulation flow has only two tasks. Double-click on Analysis & Elaboration task (Figure 11).

Tasks	5	₽ © ¥
Flow:	RTL Simulati	Customize
	Task	
	🛛 🕨 Analysis &	Elaboration
	🗄 🕨 RTL Simula	ation

Figure 11: Start Analysis and Elaboration

If there are no errors in your design, the analysis should finish with 0 errors reported in its last message:

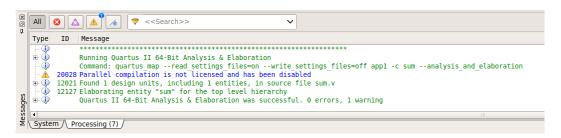


Figure 12: Successful Analysis & Elaboration

If, instead, there are errors, you must fix them and run Analysis & Elaboration. Repeat this procedure until no errors reported, otherwise the simulator cannot be launched from Quartus.

Set up Simulation

Prior to start the simulator, you need to specify what is the testbench file to be run.

From the menu select **Assignments** -> **Settings...** (a) or click right on any item in the Project Navigator panel and select **Settings...** (b).

	elp <u>File Edit View Project Assignments Processing Too</u>
ر الله <mark>Y Device</mark> ✓ Settings Ctrl+Shift+E	sum Project Navigator P @ Sum
Assignment Editor Ctrl+Shift+A A5F31C6 Pin Planner Ctrl+Shift+N Remove Assignments Back-Annotate Assignments Import Assignments	Entity Cyclone V: 5CSEMA5F31C6 Settings Locate Locate Locate Entity A & T F F A A & T F F A A A A A A A A A A A A A
a)	b)

In the Settings Simulation window (Figure 13), change **NativeLink settings** from **None** to **Compile test bench**. If the testbench is not yet set, click **TestBenches...** to select a testbench.

	Settings - sum	
Category:		Device
Category: General Files Libraries P Settings P Settings P Catalog Search Locations Operating Settings and Conditi Voltage Temperature Compilation Process Settings Incremental Compilation Physical Synthesis Optimizal EDA Tool Settings Design Entry/Synthesis Simulation Formal Verification Board-Level Analysis & Synthesis Settings Verilog HDL Input Default Parameters Fitter Settings TimeQuest Timing Analyzer Assembler Design Assistant SignalTap II Logic Analyzer Logic Analyzer Interface PowerPlay Power Analyzer Sett SSN Analyzer	Simulation Specify options for generating output files for use with other EDA tools. Tool name: ModelSim-Altera Run_gate-level simulation automatically after compilation EDA Netlist Writer settings Format for output netlist: Verilog HDL Output girectory: simulation/modelsim Map illegal HDL characters Enable glitch filtering Options for Power Estimation	Device
	Generate Value Change Dump (VCD) file script Script Settings Design instance name: More EDA Netlist Writer Settings NativeLink settings Ngne Compile test bench: Use script to set up simulation: Script to compile test bench: More NativeLink Settings	Benches
▲	Weight Buy Software OK Cancel Apply	Help

Figure 13: Change NativeLink settings from None to Compile test bench

A popup window appear that shows testbenches of the project (Figure 14).

	۲ ۱	Test Benches		<u>×</u>
Specify setting	s for each test bench.			
Existing test b	ench settings:			New
Name	p Level Modu Jesign Instanc	Run For	Test Bench File(s)	Edit
				Delete
				Delete
			OK Cancel	Help

Figure 14: Testbench settings

The project being totally new, there is no testbench in the list yet. If you have already created and saved a source file for the testbench, it needs to be added to the project in this step. Click on **New...**, to add it to the list of testbenches. A new window opens that enables you to set this testbench (Figure 15).

New Test Bench Settings 🛛 😣	
Create new test bench settings.	
Test bench name:	
Top level module in test bench:	
Use test bench to perform VHDL timing simulation	
Design instance name in test bench: NA	
Simulation period	
$\ensuremath{}$ Run simulation until all $\underline{v}ector$ stimuli are used	
○ <u>E</u> nd simulation at:	
Test bench and simulation files	Click to open the
File name: Add	File Selection
File Name Library HDL Version Remove	dialog window
Цр	
Down	
Properties	
OK Cancel Help	

Figure 15: Add a new testbench to the project

First of all specify the file name of the testbench source file. Click on search button to open the file selection dialog box. The file selection dialog box will open in the project folder (Figure 16). Select the testbench source file and click **Open**.

		Select File				8
Look in:	home/student/app1		\$ G	Э	0	🎉 📰 🗏
Computer	db incremental_db output_files app1.qpf sum.qsf sum.v sum_tb.v					
File <u>n</u> ame:	sum_tb.v					Open 2
Files of type:	All Files (*)		 		¢	Cancel

Figure 16: Select a testbench file in File selection dialog window

The File Selection dialog closes and the selected testbench file is listed in the File name field. Click **Add** to add the testbench file to the list. Then type the filename (without extension!) in the **Test bench name** field at the top of the New Test bench Settings window. Supposing you saved the testbench file with the same name as that of the testbench module, the **Test bench name** will be mirrored in the text field below. After that, the New Test bench Settings window should look as in Figure 17.

	New Test Bench Settings 🛛 😵								
Create new test	Create new test bench settings.								
Test bench name	Test bench name: sum_tb								
Top level <u>m</u> odule	Top level <u>m</u> odule in test bench: sum_tb								
Use test benc	h to perform V	HDL timing simulati	on						
<u>D</u> esign instan	ce name in tes	t bench: NA							
Simulation perio	bd								
Run simulati	ion until all <u>v</u> eo	ctor stimuli are used							
O <u>E</u> nd simulation	ion at:	s	\$						
Test bench and	simulation file	S							
<u>F</u> ile name:				Add					
File Name	Library	HDL Version	۱	<u>R</u> emove					
sum_tb.v		Default		Up					
				Down					
Properties									
		ОК	Cancel	Help					

Figure 17: New Test Bench Settings after completion

Click **OK**. The New Test Bench Settings closes and the newly selected testbench appear in the list of Test Benches window (Figure 18).

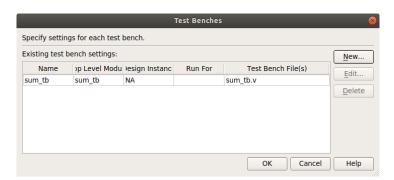


Figure 18: Test Benches list after the testbench was added

Click **OK** to close the Test Benches dialog box. The Simulation settings should now show the testbench in the **Compile test bench** field, as in Figure 19.

	Settings - sum	● 🖲 😣
Category:		Device
 General Files Libraries IP Settings -IP Catalog Search Locations Operating Settings and Conditi - Voltage - Temperature Compilation Process Settings EDA Tool Settings - Design Entry/Synthesis Simulation - Formal Verification - Board-Level Analysis & Synthesis Settings - VHDL Input - Verigg HDL Input - Verigg HDL Input - TimeQuest Timing Analyzer - SignalTap II Logic Analyzer Logic Analyzer Interface Power Analyzer Sett - SSN Analyzer 	Use script to set up simulation: Script to compile test bench: More NativeLink Settings	
	w Buy Software OK Cancel Ap	нер

Figure 19: Simulation settings completed

Leave everything else as default and click **OK** to finalize simulation setting. You are now ready to start simulation.

Start Simulation

From the menu select Tools -> Run Simulation Tool -> RTL Simulation (Figure 20).

		Ma 22:10
		Quartus II 64-Bit - /home/stuc
P <u>r</u> ocessing	<u>T</u> ools <u>W</u> indow <u>H</u> elp 🤜	
ım	Run Sim <u>u</u> lation Tool	🕐 🛃 <u>R</u> TL Simulation
	🗞 Launch Simulation Library Compiler	👷 Gate Level Simulation
🏊 Launch Design Space Explorer		
🗚 🕼 7	TimeQuest Timing Analyzer) 267 ab/
modulo oum_	Advisors	

Figure 20: Start RTL Simulation from menu

You may, instead, click on RTL Simulation button in the toolbar (Figure 21):

珍	Ó Ó	🛃 🤹 I 🧶 I 🕭 I 👗
	sun	n_tt RTL Simulation
	• 🔳	

Figure 21: RTL Simulation button

The ModelSim-Altera simulator is launched (Figure 22). It takes a couple of seconds for it to open all its windows. If everything is correctly set up and the testbench file has no errors, the Objects panel is populated with testbench signals and the Wave panel will show their waveform.

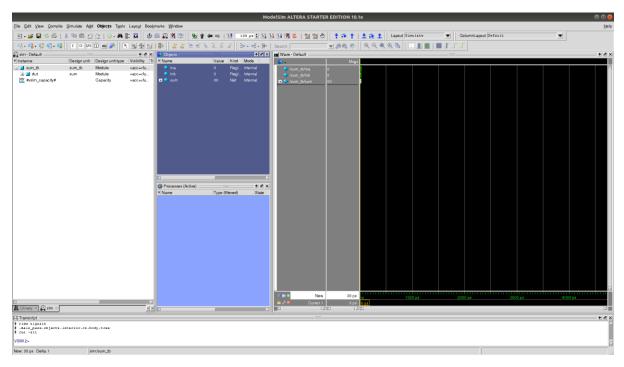


Figure 22: ModelSim-Altera simulator window right after launch

Note: If there are no signals in the Wave and Object windows, that means the testbench has not compiled because of compiling errors, or it has not loaded because the module it instantiates does not match the compiled top-level design module. In the Browser panel at the left of the simulator window you will see some instances, but not the testbench instance. In that case you must look for the errors in the Transcript panel at the bottom of the simulation window, close the simulator, fix the errors in the testbench and then relaunch the simulator.

Pin Assignment

TBD

Compilation

TBD

Device Programming

In the Tasks panel double click on **Program Device** (or, alternatively, select **Tools -> Programmer** from the Menu bar). The Programmer window will open (Figure 23).

	Programme	r - /home/stude	nt/app1/app	o1 - sum - [0	[hain1.cdf]			
<u>F</u> ile <u>E</u> dit <u>V</u> ie	w P <u>r</u> ocessing <u>T</u> oo	ls <u>W</u> indow <u>H</u>	elp 🔊					
Hardware Se	http No Hardware me ISP to allow backg			II and MAX	¢ V devices)	Progr	ess:	
■ Start	File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank- Check	Examine
Stop								
Auto Detect								
× Delete								
Add File	•			11111111				(
Change File.								
Save File								
Add Device								
¶ [™] ⊔Up								
J [™] Down								

Figure 23: Programmer window

If there is nothing in the panels of the Programmer window, the first step is to connect to the physical board. Click on **Hardware Setup** button as shown above in Figure 23. The Hardware Setup dialog window shows up (Figure 24).

	l	Hardware Setup	
ardware Settings J	TAG Settings		
elect a programming l ardware setup applies			nming devices. This programming Idow.
urrently selected hard	ware: No Hard	ware	:
Available hardware ite	ems		
Hardware	Server	Port	Add Hardware
DE-SoC	Local	3-6	Remove Hardware
			· · · · · · · · · · · · · · · · · · ·
			Close

Figure 24: Hardware Setup window

There is no hardware shown in **Currently selected hardware** field, but there should be one in the **Available hardware items** list, as shown in the caption above.

Attention! If the device doesn't appear in the list or the list is empty, make sure that you powered up the development board (otherwise the Quartus II could not connect to it), or that it is properly connected to your PC.

In **Currently selected hardware** field select from the drop-down list the De-SoC device, as shown in Figure 25 below:

Hardware Setup 🛛 🚳							
Hardware Settings JTAG Settings 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 Available hardware items DE-SoC [3-6]							
Hardware DE-SoC	Server Port Local 3-6		dd Hardware				

Figure 25: Hardware selection

Click on **Close** button. Now the Programmer Window shows the De-Soc device and also has the *Auto-Detect* button enabled (Figure 26).

	Programme	r - /home/stude	nt/app1/app	o1 - sum - [0	[hain2.cdf]			008
<u>F</u> ile <u>E</u> dit <u>V</u> ie	ew P <u>r</u> ocessing <u>T</u> oo	ls <u>W</u> indow <u>H</u>	elp 💎			Searc	:h altera	.com 🚯
Hardware So Enable real-ti	etup DE-SoC [3-6] ime ISP to allow backg		<u> </u>	II and MAX	¢ V devices)	Progr	ess:	
▶ [™] Start	File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank- Check	Examine
Jan Stop								
Auto Detect								
💥 Delete								
Add File				11111111				
Change File.								
Save File								
Add Device								
¶ [™] Up								
J [™] Down								

Figure 26: Programmer Window ready to configure the device

Click on the **Auto-Detect** button as shown above. A small pop-up window (Figure 27) appears where you should check the right device.

Select Device	8
Found devices with shared JTAG ID for device 2. Please select yo \odot 5CSEBA5	ur device.
SCSEMA5 SCSTFD5D5	
 SCSXFC5C6 SCSXFC5D6 	
	ОК

Figure 27: Select Device pop-up window

Check **5CSEMA5** device, as it is the type of Cyclone V FPGA on the development board, and then click **OK**.

Now the Programmer window panels are filled up with a chain of two devices detected on board, and two entries that allow you to program them (Figure 28).

	Programme	r - /home/stude	nt/app1/app	1 - sum - [C	hain2.cdf]*			
<u>F</u> ile <u>E</u> dit <u>V</u> ie	w Processing <u>T</u> o	ols <u>W</u> indow <u>H</u>	elp 🐬			Search	n altera.	com 🚯
Hardware Se	etup DE-SoC [3-6 me ISP to allow back		e: JTAG ing (for MAX	II and MAX	¢ V devices)	Progre	ss:	
Jan Start	File	Device	Checksum	Usercode	Program/ Configure		Blank- Check	Examine
Stop	<none></none>	SOCVHPS	00000000	<none></none>				
ww Stop	<none></none>	5CSEMA5	00000000	<none></none>				
Auto Detect								
× Delete								
Add File	•							Þ
Change File								
Save File								
Add Device		\rightarrow						
Add Device								
1[™]Up	SOCVH	0200000						
	TDO							
J [™] Down								

Figure 28: Programmer window with detected chain

You will always program the FPGA, therefore double click on the 5CSEMA5 row (second row) under the *File* column as indicated in Figure 28. A file selection dialog box (Figure 29) allows you to select the FPGA programming bit file:

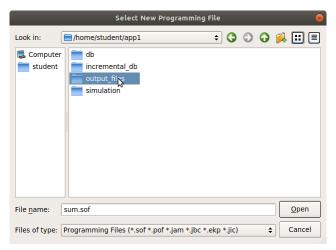


Figure 29: Select programming file window

In the project folder (in which the explorer opens by default) select the output_files subfolder. There you will find the .sof file of the project. It has, by default, the name of the top-level module.

	Select New Programming File				8
Look in:	<pre>/home/student/app1/output_files</pre>	0	Э	0	🎉 📰 🗏
Scomputer	sum.sof				
File <u>n</u> ame:	sum.sof				Open
Files of type:	Programming Files (*.sof *.pof *.jam *.jbc *.ekp *	•.jic)		\$	Cancel

Figure 30: Select programming file window

Select the desired . sof file and click **Open** (Figure 30).

The file selection dialog box closes and the Programmer window (see Figure 31) has now the row of the FPGA device filled with all needed details (don't bother, if you followed all steps as in this tutorial, they are all OK).

One last thing you need to do before programming the FPGA. Click the **Program/Configure** checkbox in the device row (5CSEMA5) as shown below (Figure 31).

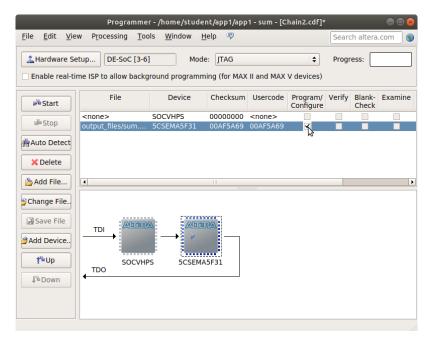


Figure 31: Programmer window ready to program the FPGA

Now cross your fingers and click the **Start** button in the Programmer window. A couple of seconds are needed to transfer the .sof file to the FPGA on De-Soc board. If the Progress field on the top right of the Programmer window stops at 100%, the board is programmed and ready to play with. **Congratulations!**

Quick Project Guide

- New Project Wizard
 - Step 1 [Directory, Name, Top-Level Entity]
 - working directory: *myProject*
 - name of this project: myProject
 - name of the top-level design entity: myModule
 - Step 3 [Family & Device Settings]
 - Select device (5CSEMA5F31C6)
 - Step 4 [EDA Tool Settings]
 - Simulation -> ModelSim Altera -> Verilog HDL
- Design source files
 - File -> New ... -> Verilog HDL File
 - Save as *myModule.v* (if it's the file for the top-level module) or *mySubmodule.v* (for other source files)
 - o Edit & Save
- RTL Simulation Flow -> Analysis & Elaboration (fix errors and rerun if needed)
- Testbench source files
 - File -> New ... -> Verilog HDL File
 - Save as myModule_tb.v
 - o Edit & Save
- Settings -> Simulation Settings
 - o check Compile test bench
 - Test Benches... -> New... -> New Test Bench Settings
 - select myModule_tb.v file and Add it
 - type myModule_tb in the Test bench name textfield
- Tools -> Run Simulation Tool -> RTL Simulation
 - o Analyse, debug, edit, recompile, restart, run
- Assignments -> Pin Planner
 - \circ choose Location for each pin
 - Compilation Flow -> Compile Design
- Compilation Flow -> Program device
 - Power on the physical board (its LEDs start flashing)
 - Programmer -> Hardware Setup... -> select De-Soc device
 - Auto Detect -> select 5CSEMA5
 - Select *myModule.sof* file for 5CSEMA5 device and check **Program/Configure** checkbox
 - o Start