

Quartus Prime Introduction Using Schematic Designs

For Quartus Prime 20.1

1 Introduction

This tutorial presents an introduction to the Quartus[®] Prime CAD system. It gives a general overview of a typical CAD flow for designing circuits that are implemented by using FPGA devices, and shows how this flow is realized in the Quartus Prime software. The design process is illustrated by giving step-by-step instructions for using the Quartus Prime software to implement a very simple circuit in an FPGA device.

The Quartus Prime system includes full support for all of the popular methods of entering a description of the desired circuit into a CAD system. This tutorial makes use of the schematic design entry method, in which the user draws a graphical diagram of the circuit. Two other versions of this tutorial are also available, which use the Verilog and VHDL hardware description languages, respectively.

The last step in the design process involves configuring the designed circuit in an actual FPGA device. To show how this is done, it is assumed that the user has access to the DE-series Development and Education board connected to a computer that has Quartus Prime software installed. A reader who does not have access to the DE-series board will still find the tutorial useful to learn how the FPGA programming and configuration task is performed.

The screen captures in the tutorial were obtained using the Quartus Prime version 16.1Standard Edition; other versions of the software may be slightly different.

Contents:

- Typical CAD Flow
- Getting Started
- Starting a New Project
- Schematic Design Entry
- Compiling the Design
- Pin Assignment
- Simulating the Designed Circuit
- Programming and Configuring the FPGA Device
- Testing the Designed Circuit

2 Background

Computer Aided Design (CAD) software makes it easy to implement a desired logic circuit by using a programmable logic device, such as a field-programmable gate array (FPGA) chip. A typical FPGA CAD flow is illustrated in Figure 1.



Figure 1. Typical CAD flow.

The CAD flow involves the following steps:

- **Design Entry** the desired circuit is specified either by means of a schematic diagram, or by using a hardware description language, such as Verilog or VHDL
- Synthesis the entered design is synthesized into a circuit that consists of the logic elements (LEs) provided in the FPGA chip
- Functional Simulation the synthesized circuit is tested to verify its functional correctness; this simulation does not take into account any timing issues

- **Fitting** the CAD Fitter tool determines the placement of the LEs defined in the netlist into the LEs in an actual FPGA chip; it also chooses routing wires in the chip to make the required connections between specific LEs
- **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
- Timing Simulation the fitted circuit is tested to verify both its functional correctness and timing
- **Programming and Configuration** the designed circuit is implemented in a physical FPGA chip by programming the configuration switches that configure the LEs and establish the required wiring connections

This tutorial introduces the basic features of the Quartus Prime software. It shows how the software can be used to design and implement a circuit specified by means of a schematic diagram. It makes use of the graphical user interface to invoke the Quartus Prime commands. Doing this tutorial, the reader will learn about:

- Creating a project
- Entering a schematic diagram
- Synthesizing a circuit from the schematic diagram
- Fitting a synthesized circuit into an FPGA
- Assigning the circuit inputs and outputs to specific pins on the FPGA
- Simulating the designed circuit
- Programming and configuring the FPGA chip on 's DE-series board

3 Getting Started

Each logic circuit, or subcircuit, being designed with Quartus Prime software is called a *project*. The software works on one project at a time and keeps all information for that project in a single directory (folder) in the file system. To begin a new logic circuit design, the first step is to create a directory to hold its files. To hold the design files for this tutorial, we will use a directory *introtutorial*. The running example for this tutorial is a simple circuit for two-way light control.

Start the Quartus Prime software. You should see a display similar to the one in Figure 2. This display consists of several windows that provide access to all the features of Quartus Prime software, which the user selects with the computer mouse. Most of the commands provided by Quartus Prime software can be accessed by using a set of menus that are located below the title bar. For example, in Figure 2 clicking the left mouse button on the menu named File opens the menu shown in Figure 3. Clicking the left mouse button on the entry Exit exits from Quartus Prime software. In general, whenever the mouse is used to select something, the *left* button is used. Hence we will not normally specify which button to press. In the few cases when it is necessary to use the *right* mouse button, it will be specified explicitly.

🕥 Quartus Prime Standard Edition		
File Edit View Project Assignments Proc	assing Tools Window Help	Search altera.com
🗋 🗖 🖯 🖬	० ० 🗌 🔤 🖉 🖉 🖉 🖌 🎋 🗘) 🚫 🔒 🔹 💙
Project Navigator	B × home 🛛	
A Compilation Hierarchy	Pecent Projecto	
	Recent Projects	
	light.qpf (C:/introtutorial/light.qpf)	
	New Project Wizard	
Tasks Compilation		_
Task	Documentation Training Support What's New	Notifications
4 🕨 Compile Design		
Analysis & Synthesis		
Fitter (Place & Route)		
Assembler (Generate programming)	ig files)	
TimeQuest Timing Analysis		
EDA Netlist Writer		
Edit Settings		
Program Device (Open Programmer)	☑ Close page after project load	
	 Close page aller project road Don't show this screen again 	ADERA.
<	Dont show this screen again	now part of Intel
	💏 Find) 💏 Find Next	
Type ID Message 14565 Can't connect to	the Altera website to check for updates. Verify that you have an active interne	t connection, and that
808 ×		
System (1) Processing		0% 00:00:00
		0% 00:00:00

Figure 2. The main Quartus Prime display.

	<u>N</u> ew	Ctrl+N
~	Open	Ctrl+O
	Close	Ctrl+F4
-	New Project Wizard	
¥.	Open Project	Ctrl+J
	Save Project	
	Close Project	
	Save	Ctrl+S
	Save <u>A</u> s	
南	Save All	Ctrl+Shift+S
	<u>F</u> ile Properties	
	Create / Update	+
	Export	
	Convert Programming Files	
Ħ	Page Setup	
	Print Preview	
÷	Print	Ctrl+P
	Recent Files	+
	Recent Projects	•
	Exit	Alt+F4

Figure 3. An example of the File menu.

For some commands it is necessary to access two or more menus in sequence. We use the convention Menu1 > Menu2 > Item to indicate that to select the desired command the user should first click the left mouse button on Menu1, then within this menu click on Menu2, and then within Menu2 click on Item. For example, File > Exit uses the mouse to exit from the system. Many commands can be invoked by clicking on an icon displayed in one of the toolbars. To see the command associated with an icon, position the mouse over the icon and a tooltip will appear that displays the command name.

3.1 Quartus Prime Online Help

Quartus Prime software provides comprehensive online documentation that answers many of the questions that may arise when using the software. The documentation is accessed from the Help menu. To get some idea of the extent of documentation provided, it is worthwhile for the reader to browse through the Help menu.

The user can quickly search through the Help topics by using the dialog box in the top right corner of the main Quartus display. Another method, context-sensitive help, is provided for quickly finding documentation for specific topics. While using most applications, pressing the F1 function key on the keyboard opens a Help display that shows the commands available for the application.

4 Starting a New Project

To start working on a new design we first have to define a new *design project*. Quartus Prime software makes the designer's task easy by providing support in the form of a *wizard*. Create a new project as follows:

- 1. Select File > New Project Wizard and click Next to reach the window in Figure 4, which asks for the name and directory of the project.
- 2. Set the working directory to be *introtutorial*; of course, you can use some other directory name of your choice if you prefer. 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. Press Next. Since we have not yet created the directory *introtutorial*, Quartus Prime software displays the pop-up box in Figure 5 asking if it should create the desired directory. Click Yes, which leads to the window in Figure 6.

New Project Wizard	×
Directory, Name, Top-Level Entity	
What is the working directory for this project?	
D:/introtutorial	
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	
Use Existing Project Settings	
< Back Next > Einish Cancel	elp

Figure 4. Creation of a new project.

🕥 Quar	tus Prime
	Directory "D:/introtutorial" does not exist. Do you want to create it?
	Yes No

Figure 5. Quartus Prime software can create a new directory for the project.

Project 1	ype							
Select the typ	e of project to create.							
Empty pro	iject							
Create n	ew project by specifyir	ng project files and I	ibraries, target	device family and	device, and EDA	tool settings.		
Project te	mplate							
	project from an existin s from the <u>Design Store</u>		'ou can choose	from design tem	olates installed wi	th the Quartus Pr	ime software, o	r download design
				< Back	Next >	Einish	Cancel	Help

Figure 6. Choosing the project type.

3. The Project Type window, shown in Figure 6, allows you to choose from the Empty project and the Project template options. For this tutorial, choose Empty project as we will be creating a project from scratch, and press Next which leads to the window in Figure 7.

					Add
٩				×	Add A <u>l</u> l
File Name	Type Library	Design Entry/Synthesis Tool	HDL Version		Remove
					<u>U</u> p
					Down
					Properties

Figure 7. The wizard can include user-specified design files.

4. The wizard makes it easy to specify which existing files (if any) should be included in the project. Assuming that we do not have any existing files, click Next, which leads to the window in Figure 8.

	and device you want ditional device suppo		tall Devices comm	and on the 10					
o determine the v	rersion of the Quartu	s Prime softw	are in which your	target device	e is supported, ref	fer to the <u>D</u>	evice Support List webpag	je.	
Device family				Sh	how in 'Available o	devices' list	t		
Eamily: Cyclone	e V (E/GX/GT/SX/SE/	/ST)		▼ Pa	ackage:	Any			7
Devices: All				_	n count:	Any			Ξ
Devices.					-	_			_
Target device				Co	ore Sp <u>e</u> ed grade:	Any			1
Auto device	selected by the Fitte	r		Na	ame filter:				٦
Specific dev Other: n/a	vice selected in 'Avai	lable devices'	list		Show advanced	d devices			
Other: n/a	:						GXB Channel PCS	PCIe Hard	
Other: n/a Available devices: Name	Core Voltage	ALMs	Total I/Os	GPIOs 314	Show advanced GXB Chanr 0		GXB Channel PCS	PCIe Hard	
Other: n/a Ayailable devices: Name 5000000000000000000000000000000000000	Core Voltage	ALMs	Total I/Os	GPIOs	GXB Chanr				
O Other: n/a Ayailable devices: Name 5CSEMA4U23A7 5CSEMA4U23C6	Core Voltage 1.1V 1.1V	ALMs 15880	Total I/Os 314	GPIOs 314	GXB Chanr		0	0	
Other: n/a Available devices: Name 5CSEMA4U23A7 5CSEMA4U23C6 5CSEMA4U23C7	Core Voltage 1.1V 1.1V 1.1V 1.1V	ALMs 15880 15880	Total I/Os 314 314	GPIOs 314 314	GXB Chanr 0 0		0	0	
Qther: n/a Available devices: Name 5CSEMA4U23A7 5CSEMA4U23C6 5CSEMA4U23C7 5CSEMA4U23C8	Core Voltage 1.1V 1.1V 1.1V 1.1V 1.1V	ALMs 15880 15880 15880	Total I/Os 314 314 314	GPIOs 314 314 314	GXB Chanr 0 0 0		0 0 0	0 0 0 0	
Other: n/a Agailable devices: Name SCSEMA4U23A7 SCSEMA4U23C6 SCSEMA4U23C7 SCSEMA4U23C8 SCSEMA4U23I7	Core Voltage 1.1V 1.1V 1.1V 1.1V 1.1V 1.1V	ALMs 15880 15880 15880 15880 15880	Total I/Os 314 314 314 314 314	GPIOs 314 314 314 314	GXB Chanr 0 0 0 0		0 0 0 0	0 0 0 0	
Other: n/a Available devices: Name SCSEMA4U23A7 SCSEMA4U23C6 SCSEMA4U23C7 SCSEMA4U23C8 SCSEMA4U2377 SCSEMA5F31A7	Core Voltage 1.1V 1.1V 1.1V 1.1V 1.1V 1.1V 1.1V 1.1V	ALMs 15880 15880 15880 15880 15880 15880	Total I/Os 314 314 314 314 314 314	GPIOs 314 314 314 314 314 314	GXB Chann 0 0 0 0 0 0		0 0 0 0 0 0	0 0 0 0 0 0	
Other: n/a	Core Voltage 1.1V 1.1V 1.1V 1.1V 1.1V 1.1V 1.1V 1.1V 1.1V	ALMs 15880 15880 15880 15880 15880 15880 15880 32070	Total I/Os 314 314 314 314 314 314 457	GPIOs 314 314 314 314 314 314 314 457	GXB Chann 0 0 0 0 0 0 0 0 0		0 0 0 0 0 0 0	0 0 0 0 0 0 0	

Figure 8. Choose the device family and a specific device.

5. We have to specify the type of device in which the designed circuit will be implemented. Choose the Cycloneseries device family for your DE-series board. We can let Quartus Prime software select a specific device in the family, or we can choose the device explicitly. We will take the latter approach. From the list of available devices, choose the appropriate device name for your DE-series board. A list of devices names on DE-series boards can be found in Table 1. Press Next, which opens the window in Figure 9.

Board	Device Name
DE0-CV	Cyclone V 5CEBA4F23C7
DE0-Nano	Cyclone IVE EP4CE22F17C6
DE0-Nano-SoC	Cyclone V SoC 5CSEMA4U23C6
DE1-SoC	Cyclone V SoC 5CSEMA5F31C6
DE2-115	Cyclone IVE EP4CE115F29C7
DE10-Lite	Max 10 10M50DAF484C7G
DE5a-NET	Arria 10 10AX115N3F45E2SG

Table 1. DE-series FPGA device names

DA tools:				
Тооі Туре	Tool Name	Format(s)		Run Tool Automatically
Design Entry/Synthesis	<none> 🔻</none>	<none></none>	-	Run this tool automatically to synthesize the current design
Simulation	<none> 🔻</none>	<none></none>	-	Run gate-level simulation automatically after compilation
Formal Verification	<none></none>]		
Board-Level	Timing	<none></none>	•	
	Symbol	<none></none>	•	
	Signal Integrity	<none></none>		
	Boundary Scan	<none></none>	~	

Figure 9. Other EDA tools can be specified.

- 6. The user can specify any third-party tools that should be used. A commonly used term for CAD software for electronic circuits is *EDA tools*, where the acronym stands for Electronic Design Automation. This term is used in Quartus Prime messages that refer to third-party tools, which are the tools developed and marketed by companies other than . Since we will rely solely on Quartus Prime tools, we will not choose any other tools. Press Next.
- 7. A summary of the chosen settings appears in the screen shown in Figure 10. Press Finish, which returns to the main Quartus Prime window, but with *light* specified as the new project, in the display title bar, as indicated in Figure 11.

Summary	
When you click Finish, the project will be created with the fo	ollowing settings:
Project directory:	D:/introtutorial
Project name:	light
Fop-level design entity:	light
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:	5CSEMA5F31C6
EDA tools:	
Design entry/synthesis:	<none> (<none>)</none></none>
Simulation:	<none> (<none>)</none></none>
Timing analysis:	0
Operating conditions:	
Core voltage:	1.1V
Junction temperature range:	0-85 °C

Figure 10. Summary of project settings.



Figure 11. The Quartus Prime display for a created project.

5 Design Entry Using the Graphic Editor

As a design example, we will use the two-way light controller circuit shown in Figure 12. The circuit can be used to control a single light from either of the two switches, x_1 and x_2 , where a closed switch corresponds to the logic value 1. The truth table for the circuit is also given in the figure. Note that this is just the Exclusive-OR function of the inputs x_1 and x_2 , but we will implement it using the gates shown.



Figure 12. The light controller circuit.

The Quartus Prime Graphic Editor can be used to specify a circuit in the form of a block diagram. Select File > New to get the window in Figure 13, choose Block Diagram/Schematic File, and click OK. This opens the Graphic Editor window. The first step is to specify a name for the file that will be created. Select File > Save As to open the pop-up box depicted in Figure 14. In the box labeled Save as type choose Block Diagram/Schematic File (*.bdf). In the box labeled File name type *light*, to match the name given in Figure 4, which was specified when the project was created. Put a checkmark in the box Add file to current project. Click Save, which puts the file into the directory *introtutorial* and leads to the Graphic Editor window displayed in Figure 15.



Figure 13. Choose to prepare a block diagram.

Organize 🔻 🛛 Nev	v folder			•
*	Name		Date modified	Туре
🗟 Libraries	🔰 db		10/23/2015 3:26 PM	File folder
Docume Music Pictures Videos Computer Local Dis	incremental_db		10/21/2015 2:06 PM	File folder
New Vol				
🕐 DVD Driv 👻 🤞		m		
File name:	light.bdf			

Figure 14. Name the file.

File Edit View Project Assignments Processing Project Navigator Image: Comparison of the image: Co	Tools Window Hep	Search altera.com
Project Navigator	iight.bdf	
Entity:Instance	<mark>音 📐 🔍 🕊</mark> A 🗇 📽 • 🖸 🗅 🗅 🦷 🔪 🔍 🗆	
A Cyclone V: 5CGXFC7C7F23C8		
tight /m t		
Tasks Compilation ▼ ≡ 🗍 🗗 ×		
Task ^		
▲ ► Compile Design		
▲ ► Analysis & Synthesis		
Edit Settings		
View Report		
Analysis & Elaboration		
Partition Merge		
> Netlist Viewers		
Design Assistant (Post-Mapping)		
▷ VO Assignment Analysis		
< III +	< III	••••••
Image: System (1) Processing	tera website to check for updates. Verify that you have an active internet	connection, and that



5.1 Importing Logic-Gate Symbols

The Graphic Editor provides a number of libraries which include circuit elements that can be imported into a schematic. Double-click on the blank space in the Graphic Editor window, or click on the $\boxed{10}$ icon in the toolbar that looks like an AND gate. A pop-up box in Figure 16 will appear. Expand the hierarchy in the Libraries box as shown in the figure. First expand *libraries*, then expand the library *primitives*, followed by expanding the library *logic* which comprises the logic gates. Select *and2*, which is a two-input AND gate, and click OK. Now, the AND gate symbol will appear in the Graphic Editor window. Using the mouse, move the symbol to a desirable location and click to place it there. Import a second AND gate by repeating the previous steps. Note that you can enable the Repeat-insert mode in the Symbol window to quickly add multiple copies of a gate. A symbol in the Graphic Editor window can be moved by clicking the $\boxed{10}$ icon in the toolbar that looks like a mouse cursor, then clicking the symbol you want to move and dragging it to a new location with the mouse button pressed. Next, select *or2* from the library and import the OR gate into the diagram. Then, select *not* and import two instances of the NOT gate. Rotate the NOT gates into proper position by right clicking on the desired gate and selecting Rotate by Degrees > Rotate Left 90, or by using the "Rotate left 90" icon $\boxed{10}$ Arrange the gates as shown in Figure 17.



Figure 16. Choose a symbol from the library.

Quartus Prime Standard Edition - D:/introtutoria	/light - light	
File Edit View Project Assignments Proces	sing Tools Window Help	Search altera.com
🗋 🗖 🖯 🤟 🚺) (* light	🔶 🚫 🖧 🔹 Ϙ
Project Navigator 🔥 Hierarchy 🔻 🚍 🖪 t	P x 🔀 light.bdf* 🗵	
Entity:Instance	☐ ☐ [] A D ≤ · □]	
A Cyclone V: 5CGXFC7C7F23C8		
> light nื่≞		ź
Tasks Compilation	P x	
Task		· · · · · · · · · · · · · · · · · · ·
4 🕨 Compile Design		
a 🕨 Analysis & Synthesis		
Edit Settings	X	
View Report	=	
Analysis & Elaboration	-	
Partition Merge		
Netlist Viewers		
 Design Assistant (Post-Mapping 		
	2	
VO Assignment Analysis		
	III	Þ
	e Altera website to check for updates. Verify that you have an active intern	net connection, and tha
		4
System (1) Processing		
		739, 326 0% 00:00:00

Figure 17. Import the gate symbols into the Graphic Editor window.

5.2 Importing Input and Output Symbols

Having entered the logic-gate symbols, it is now necessary to enter the symbols that represent the input and output ports of the circuit. Use the same procedure as for importing the gates, but choose the port symbols from the library *primitives/pin*. Import two instances of the input port and one instance of the output port, to obtain the image in Figure 18.



Figure 18. Import the input and output pins.

Assign names to the input and output symbols as follows. Make sure nothing is selected by clicking on an empty spot in the Graphic Editor window. Point to the top input symbol and double-click the mouse. The dialog box in Figure 19 will appear. Type the pin name, x1, and click OK. Similarly, assign the name x2 to the other input and f to the output. Alternatively, it is possible to change the name of an element by double-clicking on the name and typing a new one directly.

General Forr	nat	
	le pins, enter a name in AHDL bus notation	
(For example: "	name[30]"), or enter a comma-seperated list of nan	nes.
Pin name(s):	x1	
Default value:	VCC	•

Figure 19. Naming of a pin.

5.3 Connecting Nodes with Wires

The symbols in the diagram have to be connected by drawing lines (wires). Click on the icon in the toolbar to activate the Orthogonal Node Tool. Position the mouse pointer over the right edge of the x1 input pin. Click and hold the mouse button and drag the mouse to the right until the drawn line reaches the pinstub on the top input of the AND gate. Release the mouse button when you see a box appear, which leaves the line connecting the two pinstubs. Next, draw a wire from the input pinstub of the leftmost NOT gate to touch the wire that was drawn above it. Note that a dot will appear indicating a connection between the two wires.

Use the same procedure to draw the remaining wires in the circuit. If a mistake is made, a wire can be selected by clicking on it, and removed by pressing the Delete key on the keyboard. Upon completing the diagram, click on the licon, to activate the Selection Tool. Now, changes in the appearance of the diagram can be made by selecting a particular symbol or wire and either moving it to a different location or deleting it. The final diagram is shown in Figure 20; save it.



Figure 20. The completed schematic diagram.

6 Compiling the Designed Circuit

The entered schematic diagram file, *light.bdf*, is processed by several Quartus Prime tools that analyze the file, synthesize the circuit, and generate an implementation of it for the target chip. These tools are controlled by the application program called the *Compiler*.

Run the Compiler by selecting Processing > Start Compilation, or by clicking on the toolbar icon that looks like a blue triangle. Your project must be saved before compiling. As the compilation moves through various stages, its progress is reported in a window on the left side of the Quartus Prime display. In the message window, at the bottom of the figure, various messages are displayed throughout the compilation process. In case of errors, there will be appropriate messages given.

When the compilation is finished, a compilation report is produced. A tab showing this report is opened automatically, as seen in Figure 21. The tab can be closed in the normal way, and it can be opened at any time either by selecting Processing > Compilation Report or by clicking on the icon \bigcirc . The report includes a number of sections listed on the left side. Figure 21 displays the Compiler Flow Summary section, which indicates that only one

🕥 Quartus Prime Standard Edition - D:/introtutorial/ligi				
File Edit View Project Assignments Processing	Tools Window Help			Search altera.com
	Clight		🔕 👓 🕨 🗲 🌾	👌 🕒 🚠 🔹 🦻
Project Navigator 🔥 Hierarchy 🔻 🖃 🖪 🏾 🛪	ight.bdf	· ·	n Report - light 🛛 🔀	
Entity:Instance		Flow Summary Flow Status	Successful - Tue May 10 09:33:37 2016	
 Cyclone V: 5CGXFC7C7F23C8 ight n ight n Task Task Task Compilation Task Compile Design Analysis & Synthesis Edt Settings Edt Settings View Report Analysis & Elaboration Partition Merge Netist Viewers Design Assistant (Post-Mapping) VO Assignment Analysis 	 Flow Settings Flow Non-Default Global Setting Flow CS Summary Flow Lagsed Time Flow Log Analysis & Synthesis Fitter Flow Messages Flow Suppressed Messages Assembler TimeQuest Timing Analyzer 	Quartus Prime Version Revision Name Top-level Entty Name Family Device Timing Models Logic utilization (in ALMs) Total registers Total pins Total virtual pins Total virtual pins Total block memory bits Total DSP Blocks Total DSP Blocks Total DSS PMA RX Deserializers Total HSSI PMA RX Deserializers Total HSSI PMA TX Serializers Total HSSI PMA TX Serializers Total PLLs	16.0.0 Build 211 04/27/2016 SJ Standard light light Cyclone V 5CGXFC7C7F23C8 Final 1/56,480 (<1%) 0 3/288 (1%) 0 0/7,024,640 (0%) 0/7,024,640 (0%) 0/6 (0%) 0/6 (0%) 0/6 (0%) 0/6 (0%) 0/6 (0%)	Edition
* All 🔕 🔊 🚺 📐 🔻 <		Find 😽 Find Next		
Type ID Message 332102 Design is not fully co 332102 Design is not fully co Quartus Prime Timeque: 293000 Quartus Prime Full co System (1) Processing (124)	onstrained for hold require st Timing Analyzer was succ npilation was successful. O	ments essful. O errors, 6 w	varnings	F
System (1) Processing (124)				238, 6 100% 00:01:13

logic element and three pins are needed to implement this tiny circuit on the selected FPGA chip.

Figure 21. Display after a successful compilation.

6.1 Errors

Quartus Prime software displays messages produced during compilation in the Messages window. If the block diagram design file is correct, one of the messages will state that the compilation was successful and that there are no errors.

If the Compiler does not report zero errors, then there is at least one mistake in the schematic entry. In this case a message corresponding to each error found will be displayed in the Messages window. Double-clicking on an error message will highlight the offending part of the circuit in the Graphic Editor window. Similarly, the Compiler may display some warning messages. Their details can be explored in the same way as in the case of error messages. The user can obtain more information about a specific error or warning message by selecting the message and pressing the F1 function key.

To see the effect of an error, open the file *light.bdf*. Remove the wire connecting the output of the top AND gate to the OR gate. To do this, click on the $\boxed{\]}$ icon, click the mouse on the wire to be removed (to select it) and press Delete. Compile the erroneous design by clicking on the $\boxed{\]}$ icon. A pop-up box will ask if the changes made to the *light.bdf* file should be saved; click Yes. After trying to compile the circuit, Quartus Prime software will display error messages in the Messages window, and show that the compilation failed in the Analysis & Synthesis Intel Corporation - FPGA University Program

stage of the compilation process. The compilation report summary, given in Figure 22, confirms the failed result. In the Table of Contents panel, expand the Analysis & Synthesis part of the report and then select Messages to have the messages displayed as shown in Figure 23. The Compilation Report can be displayed as a separate window as in Figure 23 by right-clicking its tab and selecting Detach Window, and can be reattached by clicking Window > Attach Window. Double-click on the first error message, which states that one of the nodes is missing a source. Quartus Prime software responds by displaying the *light.bdf* schematic and highlighting the OR gate which is affected by the error, as shown in Figure 24. Correct the error and recompile the design.



Figure 22. Compilation report for the failed design.

Compilation Report - D:/introtu File Edit Tools Window Hel	
Table of Contents Image: Contents Image: Contents Image: Contents <th><pre>Running Quartus Prime Analysis & Synthesis Command: quartus.mapread.settings_files=onwrite_settings_files=off light 20030 Parallel compilation is enabled and will use 6 of the 6 processors detected 12021 Found 1 design units, including 1 entities, in source file light.bdf 12127 Elaborating entity "light" for the top level hierarchy 275008 Primitive "AND2" of instance "inst1" not used 275008 Primitive "NOT" of instance "inst1" not used 12009 Node "inst2" is missing source Quartus Prime Analysis & Synthesis was unsuccessful. 1 error, 2 warnings</pre></th>	<pre>Running Quartus Prime Analysis & Synthesis Command: quartus.mapread.settings_files=onwrite_settings_files=off light 20030 Parallel compilation is enabled and will use 6 of the 6 processors detected 12021 Found 1 design units, including 1 entities, in source file light.bdf 12127 Elaborating entity "light" for the top level hierarchy 275008 Primitive "AND2" of instance "inst1" not used 275008 Primitive "NOT" of instance "inst1" not used 12009 Node "inst2" is missing source Quartus Prime Analysis & Synthesis was unsuccessful. 1 error, 2 warnings</pre>
	۲ الله الله الله الله الله الله الله الل
4	Processing (10)

Figure 23. Error messages.



Figure 24. Identifying the location of the error.

7 Simulating the Designed Circuit

Before implementing the designed circuit in the FPGA chip on the DE-series board, it is prudent to simulate it to ascertain its correctness. Quartus Prime's Simulation Waveform Editor tool can be used to simulate the behavior of a designed circuit. Before the circuit can be simulated, it is necessary to create the desired waveforms, called *test vectors*, to represent the input signals. It is also necessary to specify which outputs, as well as possible internal points in the circuit, the designer wishes to observe. The simulator applies the test vectors to a model of the implemented circuit and determines the expected response. We will use the Simulation Waveform Editor to draw the test vectors, as follows:

- 1. In the main Quartus Prime window, select File > New > Verification/Debugging Files > University Program VWF to open the Simulation Waveform Editor.
- 2. The Simulation Waveform Editor window is depicted in Figure 31. Save the file under the name *light.vwf*; note that this changes the name in the displayed window. Set the desired simulation to run from 0 to 200 ns by selecting Edit > Set End Time and entering 200 ns in the dialog box that pops up. Selecting View > Fit in Window displays the entire simulation range of 0 to 200 ns in the window, as shown in Figure 32. You may wish to resize the window to its maximum size.



Figure 31. The Waveform Editor window.

Simulation Waveform Editor	- D:/introtutorial,	/light - light - [light.vwf]			
File Edit View Simulation	Help 🤜				Search altera	com 🚯
💽 🍳 👗 & 라 👗 🗶	$\overline{\mathbf{L}} \setminus \overline{\mathbf{H}} $ we have $\overline{\mathbf{C}} \times \overline{\mathbf{C}}$	∑ X? XB ₽ ₽ ₽ ₽ ₽	■■號			
Master Time Bar: 0 ps		Pointer: 38.64 ns	Interval: 38.64 ns	Start:	End:	
Name Value a		40.0 ns	80.0 ns	120.0 ns	160.0 ns	200.0 ns 🔺
0 ps	0 ps					
•	* •		III			- 4
					0%	00:00:00

Figure 32. The augmented Waveform Editor window.

3. Next, we want to include the input and output nodes of the circuit to be simulated. Click Edit > Insert > Insert Node or Bus to open the window in Figure 33. It is possible to type the name of a signal (pin) into the Name box, or use the Node Finder to search your project for the signals. Click on the button labeled Node Finder to open the window in Figure 34. The Node Finder utility has a filter used to indicate what type of nodes are to be found. Since we are interested in input and output pins, set the filter to Pins: all. Click the List button to find the input and output nodes as indicated on the left side of the figure.

de or Bus	×
	ОК
NPUT -	Cancel
9-Level 🔻	
Binary	Node Finder
1	
0	
ray code count as binary cou	int
	9-Level Binary 1

Figure 33. The Insert Node or Bus dialogue.

Node Finder						×
Named: *		Filter:	ins: all		•	ОК
Look in: *					List	Cancel
Nodes Found:			Sele	cted Nodes:		
Name	Туре			Name		Туре
out f	Output		i	x1	Input	
in x1	Input		2 📮	x2	Input	
in x2	Input		>> 🛀	f	Output	
			<			
		6	<<			

Figure 34. Selecting nodes to insert into the Waveform Editor.

Click on the x1 signal in the Nodes Found box in Figure 34, and then click the > sign to add it to the Selected Nodes box on the right side of the figure. Do the same for x2 and f. Click OK to close the Node Finder window, and then click OK in the window of Figure 33. This leaves a fully displayed Waveform Editor window, as shown in Figure 35. If you did not select the nodes in the same order as displayed in Figure 35, it is possible to rearrange them. To move a waveform up or down in the Waveform Editor window, click within the node's row (i.e. on its name, icon, or value) and drag it up or down in the Waveform Editor.

	<u>E</u> dit <u>V</u> iew	<u>S</u> imulation <u>H</u> e					Search alter	ra.com
	🔍 💥 🖞	, , , , , , , , , , , , , , , , , , ,	H WY XC X&	X2 XB 🗞 🗞 3	🖕 📠 账			
laster	Time Bar: 0	ps	• • F	ointer: 672 ps	Interval: 672 ps	Start:	End:	
	Name	Value at 0 ps	0 ps 0 ps	40.0 ns	80.0 ns	120.0 ns	160.0 ns	200.0 ns
in	x1	в 0						
in	x2	В 0						
eut	f	BX	*****	*******	*****	*******	*******	******
					<u> </u>			



4. We will now specify the logic values to be used for the input signals x1 and x2 during simulation. The logic values at the output f will be generated automatically by the simulator. To make it easy to draw the desired waveforms, the Waveform Editor displays (by default) vertical guidelines and provides a drawing feature that snaps on these lines (which can otherwise be invoked by choosing Edit > Snap to Grid). Observe also a solid vertical line, which can be moved by pointing to its top and dragging it horizontally. This reference line is used in analyzing the timing of a circuit; move it to the *time* = 0 position. The waveforms can be drawn using the Selection Tool, which is activated by selecting the icon **k** in the toolbar.

To simulate the behavior of a large circuit, it is necessary to apply a sufficient number of input valuations and observe the expected values of the outputs. In a large circuit the number of possible input valuations may be huge, so in practice we choose a relatively small (but representative) sample of these input valuations. However, for our tiny circuit we can simulate all four input valuations given in Figure 12. We will use four 50-ns time intervals to apply the four test vectors.

We can generate the desired input waveforms as follows. Click on the waveform for the x1 node. Once a waveform is selected, the editing commands in the Waveform Editor can be used to draw the desired waveforms. Commands are available for setting a selected signal to 0, 1, unknown (X), high impedance (Z), weak low (L), weak high (H), a count value (C), an arbitrary value, a random value (R), inverting its existing value (INV), or defining a clock waveform. Each command can be activated by using the Edit > Value command, or via the toolbar for the Waveform Editor. The Value menu can also be opened by right-clicking on a selected waveform.

Set x1 to 0 in the time interval 0 to 100 ns, which is probably already set by default. Next, set x1 to 1 in the time interval 100 to 200 ns. Do this by pressing the mouse at the start of the interval and dragging it to its end, which highlights the selected interval, and choosing the logic value 1 in the toolbar. Make x2 = 1 from 50 to 100 ns and also from 150 to 200 ns, which corresponds to the truth table in Figure 12. This should produce the image in Figure 36. Observe that the output f is displayed as having an unknown value at this time, which is indicated by a hashed pattern; its value will be determined during simulation. Save the file.

🕤 Sim	nulation Wav	eform Editor - D:	/introtutorial/lig	ght - light - [light.vw]*		
<u>F</u> ile	<u>E</u> dit <u>V</u> iew	Simulation <u>H</u> el	p 🤿				Search altera.com
	9 🔍 🔆 🕰	ይ ሊ 👗 🗡 🛛	<u>₩ ₩ \C \&</u>	X2 XB 🔩 👯 🦓	· 🖬 🎼		
Maste	r Time Bar:) ps		pinter: 110.54 ns	Interval: 110.54 ns	Start: 150.0 ns	End: 200.0 ns
	Name	Value at 0 ps	0 ps 0 ps	40.0 ns	80.0 ns	120.0 ns	160.0 ns 200.0 ns
in	x 1	в 0					
in_	x2	B 0					
eut	f	ВΧ	×****	\times	****	*******	×××××××××××××
		•	•		III		
							0% 00:00:00

Figure 36. Setting of test values.

8.1 Performing the Simulation

Intel Corporation - FPGA University Program November 2016 A designed circuit can be simulated in two ways. The simplest way is to assume that logic elements and interconnection wires in the FPGA are perfect, thus causing no delay in propagation of signals through the circuit. This is called *functional simulation*. A more complex alternative is to take all propagation delays into account, which leads to *timing simulation*. Typically, functional simulation is used to verify the functional correctness of a circuit as it is being designed.

8.1.1 Functional Simulation

Before running a functional simulation it is necessary to run Analysis and Synthesis on your design by selecting the \leq icon in the main Quartus Prime window. Note that Analysis and Synthesis gets run as a part of the main compilation flow. If you compiled your design in Section 6, then it is not necessary to run Analysis and Synthesis again.

To perform the functional simulation, select Simulation > Run Functional Simulation or select the icon in the Simulation Waveform Editor window. A pop-up window will show the progress of the simulation then automatically close when it is complete. At the end of the simulation, a second Waveform Editor window will open the results of the simulation as illustrated in Figure 37. Observe that the output f is as specified in the truth table of Figure 12.

ile	<u>E</u> dit <u>V</u> iew	Simulation <u>H</u> e	lp 🤜				Search alter	a.com
	9 💥 🖞	ያ <mark>ተ </mark>	H INV XC X	X2 XB 🗞 🗞 🌴				
laste	r Time Bar: () ps		Pointer: 167.84 ns	Interval: 167.84 ns	Start:	End:	
	Name	Value at 0 ps	0 ps 0 ps	40.0 ns	80.0 ns	120.0 ns	160.0 ns	200.0 ns 🕯
in	x1	В 0						
in_	x2	В 0						
out	f	В 0						
٠.		1						- + -

Figure 37. The result of functional simulation.

8.1.2 Timing Simulation

Having ascertained that the designed circuit is functionally correct, we should now perform the timing simulation to see how it will behave when it is actually implemented in the chosen FPGA device. Before running a timing simulation, it is necessary to compile your design by selecting the \triangleright icon in the main Quartus Prime window. Unlike functional simulations, timing simulations require the full compilation of your design, not just Analysis and Synthesis.

To perform the timing simulation, select Simulation > Run Timing Simulation or select the icon in the Simulation Waveform Editor window. The simulation should produce the waveforms in Figure 38. Observe that there is a delay of about 5 ns in producing a change in the signal f from the time when the input signals, x_1 and x_2 , change their values. This delay is due to the propagation delays in the logic element and the wires in the FPGA device.

Note: timing simulations are only supported by Cyclone IV and Stratix IV FPGAs. If your DE-series board does not have a Cyclone IV or Stratix IV FPGA, the result of a timing simulation will be identical to the functional simulation.

ile I	Edit View	Simulation	Help 🤜				Search alte	ra.com
		-					Search alle	
	2 💥 🖞	r <u>1</u> Z	XI XH WY X	C X3 X7 XR ₽ ₽ ₽ ₽	🐴 🔜 昵			
laster	Time Bar: 0	ps	•	Pointer: 56.67 ns	Interval: 5	6.67 ns Start:	End:	
	Name	Value a 0 ps	0 ps 0 ps	40.0 ns	80.0 ns	120.0 ns	160.0 ns	200.0 ns 🔺
in	x1	в 0						
in	x2	В 0			L			
out	f	ВX	XXX	Г				
4		÷.				: : : : 		

Figure 38. The result of timing simulation.

8 Pin Assignment



Figure 1-4 Board Block Diagram

During the compilation above, the Quartus Prime Compiler was free to choose any pins on the selected FPGA to serve as inputs and outputs. However, the DE-series board has hardwired connections between the FPGA pins and

the other components on the board. We will use two toggle switches, labeled SW_0 and SW_1 , to provide the external inputs, x_1 and x_2 , to our example circuit. These switches are connected to the FPGA pins listed in Table 2.

Signal Name	FPGA Pin No.	Description
SW0	PIN_C10	Slide Switch[0]
SW1	PIN_C11	Slide Switch[1]
SW2	PIN_D12	Slide Switch[2]
SW3	PIN_C12	Slide Switch[3]
SW4	PIN_A12	Slide Switch[4]
SW5	PIN_B12	Slide Switch[5]
SW6	PIN_A13	Slide Switch[6]
SW7	PIN_A14	Slide Switch[7]
SW8	PIN_B14	Slide Switch[8]
SW9	PIN_F15	Slide Switch[9]

Table 2. DE-Series Pin Assignments

Signal Name	FPGA Pin No.	Description
LEDR0	PIN_A8	LED [0]
LEDR1	PIN_A9	LED [1]
LEDR2	PIN_A10	LED [2]
LEDR3	PIN_B10	LED [3]
LEDR4	PIN_D13	LED [4]
LEDR5	PIN_C13	LED [5]
LEDR6	PIN_E14	LED [6]
LEDR7	PIN_D14	LED [7]
LEDR8	PIN_A11	LED [8]
LEDR9	PIN_B11	LED [9]

Signal Name	FPGA Pin No.	Description
KEY0	PIN_B8	Push-button[0]
KEY1	PIN_A7	Push-button[1]

Signal Name	FPGA Pin No.	Description
ADC_CLK_10	PIN_N5	10 MHz clock input for ADC (Bank 3B)
MAX10_CLK1_50	PIN_P11	50 MHz clock input(Bank 3B)
MAX10_CLK2_50	PIN_N14	50 MHz clock input(Bank 3B)



Figure 3-12 Clock circuit of the FPGA Board

Pin assignments are made by using the Assignment Editor. Select Assignments > Assignment Editor to reach the window in Figure 25 (shown here as a detached window). In the Category drop-down menu select All. Click on the <<new>> button located near the top left corner to make a new item appear in the table. Double click the box under the column labeled To so that the Node Finder button appears. Click on the button (not the drop down arrow) to reach the window in Figure 26. Click on \leq and \leq to show or hide more search options. In the Filter drop-down menu select Pins: all. Then click the List button to display the input and output pins to be assigned: f, x1, and x2. Click on x1 as the first pin to be assigned and click the > button; this will enter x1 in the Selected Nodes box. Click OK. x1 will now appear in the box under the column labeled To. Alternatively, the node name can be entered directly by double-clicking the box under the To column and typing in the node name.

File Ed	lit View Tool	s Window Help	•			5	Search altera.com	
< <new></new>	> 🔻 🔽 Filter on	node names: *					I	
Status	From	То	Assignment Name	Value	Enabled	Entity	Comment	Та
	< <new>></new>	< <new>></new>	< <new>></new>					
×								
<				m				

Figure 25. The Assignment Editor window

Follow this by double-clicking on the box to the right of this new x1 entry, in the column labeled Assignment Name. Now, the drop-down menu in Figure 27 appears. Scroll down and select Location (Accepts wildcards/groups). Instead of scrolling down the menu to find the desired item, you can just type the first letter of the item in the Assignment Name box. In this case the desired item happens to be the first item beginning with L. Finally, double-click the box in the column labeled Value. Type the pin assignment corresponding to SW_0 for your DE-series board, as listed in Table 2.

Use the same procedure to assign input x^2 and output f to the appropriate pins listed in Table 2. An example using a DE1-SoC board is shown in Figure 28. To save the assignments made, choose File > Save. You can also simply 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.

Named:	*				•	List 🔗
Options						
Filter:	Pins: all					▼ Customize
Look in:	light	-		🗸 Include su	bentities	Hierarchy view
Matching	Nodes:	+: -:		Nodes Found	l:	
	Name	Assignments	₽ ∥	Nam	e	Assignments
	f x1 x2	Unassigned Unassigned Unassigned	∧ ⊗ <			
•		-		•	III	•

Figure 26. The Node Finder displays the input and output names.

Assignment Name	Value	Enabled	Entity	Comment	Tag
ocation (Accepts wildcards	(aroups)				
ILAB Add Timing Constraints		eed-Through Mode	Setting Don't Car	e (Accepts wildcard	ds/groups)
Manual Logic Duplication (Ac	cepts wildcards/	groups)			
Match PLL Compensation Clo	ck (Accepts wild	cards/groups)			



¢	Assignme	nt Editor - D:/intro	otutorial/light - light	ť				
E	ile <u>E</u> dit <u>)</u>	<u>V</u> iew <u>T</u> ools <u>W</u> i	indow <u>H</u> elp 🤜				Search	altera.com
<	<new>> 💌</new>	Filter on node r	names: *			•	Category: All	-
	Status	From	То	Assignment Name	Value	Enabled	Entity	Comment
1	🖌 Ok		in_ x1	Location	PIN_AB12	Yes		
2	🖌 Ok		in_ x2	Location	PIN_AC12	Yes		
3	🖌 Ok		out f	Location	PIN_V16	Yes		
4		< <new>></new>	< <new>></new>	< <new>></new>				
•				III				
× P P		hows the status of	the assignment in the	current row.				
								0% 00:00:00

Figure 28. The complete assignment.

The DE-series board has fixed pin assignments. Having finished one design, the user will want to use the same pin assignment for subsequent designs. Going through the procedure described above becomes tedious if there are many pins used in the design. A useful Quartus Prime feature allows the user to both export and import the pin assignments from a special file format, rather than creating them manually using the Assignment Editor. A simple file format that can be used for this purpose is the *Quartus Settings File (QSF)* format. The format for the file for our simple project (on a DE1-SoC board) is

set_location_assignment PIN_AB12 -to x1 set_location_assignment PIN_AC12 -to x2 set_location_assignment PIN_V16 -to f

By adding lines to the file, any number of pin assignments can be created. Such *qsf* files can be imported into any design project.

If you created a pin assignment for a particular project, you can export it for use in a different project. To see how this is done, open again the Assignment Editor to reach the window in Figure 28. Select Assignments > Export Assignment which leads to the window in Figure 29. Here, the file *light.qsf* is available for export. Click on OK. If you now look in the directory, you will see that the file *light.qsf* has been created.

<mark>}</mark>	Export Assignments
	Assignments to export
	File name:
	D:/introtutorial/atom_netlists/light.qsf
	Export assignments hierarchy path:
	light
	Save intermediate synthesis results
	Save a node-level netlist of the entire design into a persistent source file
	File name:
	OK Cancel Help
	Cancer Help

Figure 29. Exporting the pin assignment.

You can import a pin assignment by choosing Assignments > Import Assignments. This opens the dialogue in Figure 30 to select the file to import. Type the name of the file, including the *qsf* extension and the full path to the directory that holds the file, in the File Name box and press OK. Of course, you can also browse to find the desired file.

ſ	S Import Assignments	×
	Specify the source and categories of assignments to import.	
	File name:	Categories
	Copy existing assignments into light.qsf.bak before importing	Advanced
	OK Cancel	Help
		.4

Figure 30. Importing the pin assignment.

For convenience when using large designs, all relevant pin assignments for the DE-series board are given in individual files. For example, the DE1-SoC pin assignments can be found in the $DE1_SoC.qsf$ file, which is available from Intel's FPGA University Program website. This file uses the names found in the $DE1_SoC$ User Manual. If we wanted to make the pin assignments for our example circuit by importing this file, then we would have to use the same names in our Block Diagram/Schematic design file; namely, SW[0], SW[1] and LEDG[0] for x1, x2 and f, respectively. Since these signals are specified in the $DE1_SoC.qsf$ file as elements of vectors SW and LEDG, we must refer to them in the same way in our design file. For example, in the $DE1_SoC.qsf$ file the 10 toggle switches are called SW[9] to SW[0]. In a design file they can also be referred to as a vector SW[9..0].

Creating a block diagram of your circuit:

Follow the following steps to create a block symbol of your circuit:

- Save the file by choosing File > Save, pressing Ctrl + s, or by clicking the floppy disk icon.
- Choose File > Create/Update > Create Symbol Files for Current File to convert the *.bfd file to a Symbol File (*.sym). You use this Symbol File to add the HDL code to your BDF schematic.



Figure 31. Steps to create a symbol

The Quartus software creates a Symbol File and displays a message

Using a predefined block symbol in your design

To start a new higher level design which incorporates the created block symbol:

- Open a new file
- Add the block symbol under components just like adding other gates, except this would be under your project
- Use other gates to complete your design
- Mark your design as a top-level design so that your new pin assignments can be implemented.

Project Navigator 🛛 🕫 × 👔	Adder.bdf* 🛛 👫 Block4.bdf* 🖸	IP Catalog # 6
9, × 🙀 k 🔍		9 (* X
Entity Cyclone V: 5CEBA4F23C7 Adder	Symbol	
< → Hierarchy ■ Files ◀ ►	Adder Ad	DSP Interface Protocols Memory Interfaces and Controllers Processors and Periphenal University Program Search for Partner IP
Tasks ##×		
łow: Compilati • Customize	A Z Name:	
Tasl ^	nst ✓ Repeat-insert mode	
✓ ► Compile Design	Threat symbol at black	
> Analysis & Synth		
> 🕨 Fitter (Place & Rc		OK Cancel
> Assembler (Gene	inst	
< > <		> Add
×		
🛱 All 🔉 🛆 🔺 🦽 💎 << <filter>></filter>	와 Find 커 Find Next	
'vor TD Message		
sačes		

Figure 32. Using a created block symbol

9 Programming and Configuring the FPGA Device

The FPGA device must be programmed and configured to implement the designed circuit. The required configuration file is generated by the Quartus Prime Compiler's Assembler module. Intel's DE-series board allows the configuration to be done in two different ways, known as JTAG and AS modes. The configuration data is transferred from the host computer (which runs the Quartus Prime software) to the board by means of a cable that connects a USB port on the host computer to the USB-Blaster connector on the board. To use this connection, it is necessary to have the USB-Blaster driver installed. If this driver is not already installed, consult the tutorial *Getting Started with Intel's DE-Series Boards* for information about installing the driver. Before using the board, make sure that the USB cable is properly connected and turn on the power supply switch on the board.

In the JTAG mode, the configuration data is loaded directly into the FPGA device. The acronym JTAG stands for Joint Test Action Group. This group defined a simple way for testing digital circuits and loading data into them, which became an IEEE standard. If the FPGA is configured in this manner, it will retain its configuration as long as the power remains turned on. The configuration information is lost when the power is turned off. The second possibility is to use the Active Serial (AS) mode. In this case, a configuration device that includes some flash memory is used to store the configuration data. Quartus Prime software places the configuration data into the configuration device on the DE-series board. Then, this data is loaded into the FPGA upon power-up or reconfiguration. Thus, the

FPGA need not be configured by the Quartus Prime software if the power is turned off and on. The choice between the two modes is made by switches on the DE-series board. Consult your manual for the location of this switch on your DE-series board. The boards should be set to JTAG mode by default. This tutorial discusses only the JTAG programming mode.

9.1 JTAG Programming for the DE0-CV, DE0-Nano and DE2-115 Boards

For the DE0-CV, DE0-Nano and DE2-115 Boards, the programming and configuration task is performed as follows. If using the DE1-SoC board, then the instructions in the following section should be followed. To program the FPGA chip, the RUN/PROG switch on the board must be in the RUN position. Select **Tools** > **Programmer** to reach the window in Figure 39. Here it is necessary to specify the programming hardware and the mode that should be used. If not already chosen by default, select JTAG in the Mode box. Also, if the USB-Blaster is not chosen by default, press the **Hardware Setup**... button and select the USB-Blaster in the window that pops up, as shown in Figure 40.



Figure 39. The Programmer window.

Observe that the configuration file *light.sof* is listed in the window in Figure 39. If the file is not already listed, then click Add File and select it. This is a binary file produced by the Compiler's Assembler module, which contains the data needed to configure the FPGA device. The extension *.sof* stands for SRAM Object File. Ensure the Program/Configure check box is ticked, as shown in Figure 40.

lardware Settings	JTAG S	Settings			
elect a programmi ardware setup ap					s. This programming
urrently selected I	hardware:	USB-Blas	ster [USB-0]		
Available hardwa	re items				
Hardware			Server	Port	Add Hardware
USB-Blaster			Local	USB-0	Remove Hardware

Figure 40. The Hardware Setup window.

Now, press Start in the window in Figure 39. An LED on the board will light up corresponding to the programming operation. If you see an error reported by Quartus Prime software indicating that programming failed, then check to ensure that the board is properly powered on.

9.2 JTAG Programming for the DE0-Nano-SoC and DE1-SoC Board

For the DE1-SoC board, the following steps should be used for programming. Select Tools > Programmer to reach the window in Figure 41. Here it is necessary to specify the programming hardware and the mode that should be used. If not already chosen by default, select JTAG in the Mode box. Also, if *DE-SoC* is not chosen by default as the programming hardware, then press the Hardware Setup... button and select the *DE-SoC* in the window that pops up, as shown in Figure 42.



Intel Corporation - FPGA University Program November 2016

Figure 41. The Programmer window.

Observe that the configuration file *light.sof* in directory *output_files* is listed in the window in Figure 41. If the file is not already listed, then click Add File and select it. This is a binary file produced by the Compiler's Assembler module, which contains the data needed to configure the FPGA device. The extension *.sof* stands for SRAM Object File. Ensure the Program/Configure box is checked. This setting is used to select the FPGA in the Cyclone V SoC chip for programming. If the *SOCVHPS* device is not shown as in Figure 41, click Add Device > SoC Series V > SOCVHPS then click OK. Ensure that your device order is consistent with Figure 41 by clicking on a device and then clicking **1**th up or **1**th Down.

ardware Settings	JTAG	Settings			
elect a programming ardware setup appli					s. This programming
urrently selected ha	rdware:	DE-SoC	[USB-1]		
Available hardware	items				
Hardware			Server	Port	Add Hardware
DE-SoC			Local	USB-1	Remove Hardware

Figure 42. The Hardware Setup window.

Now, press Start in the Programmer. An LED on the DE1-SoC board will light up while the FPGA device is being programmed. If you see an error reported by Quartus Prime software indicating that programming failed, then check to ensure that the board is properly powered on.

10 Testing the Designed Circuit

Having downloaded the configuration data into the FPGA device, you can now test the implemented circuit. Try all four valuations of the input variables x_1 and x_2 , by setting the corresponding states of the switches SW_1 and SW_0 . Verify that the circuit implements the truth table in Figure 12.

If you want to make changes in the designed circuit, first close the Programmer window. Then make the desired changes in the Block Diagram/Schematic file, compile the circuit, and program the board as explained above.

Copyright © 1991-2016 Intel Corporation. All rights reserved. Intel, The Programmable Solutions Company, the stylized Intel logo, specific device designations, and all other words and logos that are identified as trademarks and/or service marks are, unless noted otherwise, the trademarks and service marks of Intel Corporation in the U.S. and other countries. All other product or service names are the property of their respective holders. Intel products are protected under numerous U.S. and foreign patents and pending applications, mask work rights, and copyrights. Intel warrants performance of its semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel Corporation. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

This document is being provided on an "as-is" basis and as an accommodation and therefore all warranties, representations or guarantees of any kind (whether express, implied or statutory) including, without limitation, warranties of merchantability, non-infringement, or fitness for a particular purpose, are specifically disclaimed.