# My First FPGA for Altera DE2i-150 Board

### ter <mark>asIC</mark>



Chapter 1	Introduction
	1.1 Design Flow
	1.2 Before You Begin
	1.3 What You Will Learn
Chapter 2	Assign The Device
	2.1 Assign The Device
Chapter 3	Design Entry
	3.1 Add a PLL Megafunction
	3.2 Add a Multiplexer
	3.3 Assign the Pins
	3.4 Create a Default TimeQuest SDC File
Chapter 4	Compile and Verify Your Design
	4.1 Compile Your Design
	4.2 Program the FPGA Device
	4.3 Verify The Hardware
Chapter 5	<i>DE2i-150 System Builder</i>
	5.1 Headquarter & Branches

# Chapter 1



This tutorial provides comprehensive information that will help you understand how to create a FPGA design and run it on you DE2i-150 development board. The following sections provide a quick overview of the design flow, explain what you need to get started, and describe what you will learn.

# 1.1 Design Flow

Figure 1-1shows the FPGA design flow block diagram.

The standard FPGA design flow starts with design entry using schematics or a hardware description language (HDL), such as Verilog HDL or VHDL. In this step, you can create a digital circuit that is implemented inside the FPGA. The flow then proceeds through compilation, simulation, programming, and verification in the FPGA hardware.

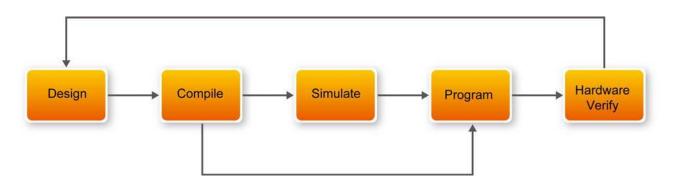


Figure 1-1 Design Flow

This tutorial guides you through all of the steps except for simulation. Although it is not covered in this document, simulation is very important to learn, and there are entire applications devoted to simulating hardware designs. There are two types of simulation, Functional and Timing Functional simulation allows you to verify that your code is manipulating the inputs and outputs appropriately. Timing (or post place-and-route) simulation verifies that the design meets timing and functions appropriately in the device.



# 1.2 Before You Begin

This tutorial assumes the following prerequisites

■ You generally know what a FPGA is. This tutorial does not explain the basic concepts of programmable logic.

■ You are somewhat familiar with digital circuit design and electronic design automation (EDA) tools.

■ You have installed the Altera Quartus II 12.0sp1 software on your computer. If you do not have the Quartus II software, you can download it from the Altera web site at www.altera.com/download.

■ You have a DE2i-150 Development Board on which you will test your project. Using a development board helps you to verify whether your design is really working.

■ You have gone through the quick start guide and/or the getting started user guide for your development kit. These documents ensure that you have:

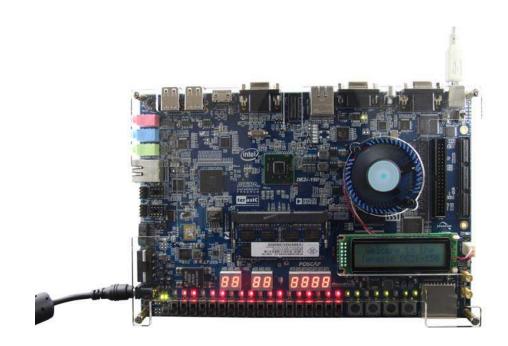
- Installed the required software.
- Determined that the development board functions properly and is connected to your computer.

Next step you should installed the USB-Blaster driver, Plug in the 12-volt adapter to provide power to the board. Use the USB cable to connect the rightmost USB connector (the one closest to the VGA FPGA) on the DE2i-150 board to a USB port on a computer that runs the Quartus II software. Turn on the power switch on the DE2i-150 board.

The computer will recognize the new hardware connected to its USB port and Power on the board as shown in **Figure 1-3**, but it will be unable to proceed if it does not have the required driver already installed. The DE2i-150 board is programmed by using Altera USB-Blaster mechanism. If the USB-Blaster driver is not already installed, the New Hardware Wizard in **Figure 1-3** will appear.







#### **Figure 1-2 Connection Setup**



Figure 1-3 Found New Hardware Wizard

Since the desired driver is not available on the Windows Update Web site, select "No, not this time" in response to the question asked and click Next. This leads to the window in **Figure 1-4**.







Figure 1-4 The driver is found in a specific location

The driver is available within the Quartus II software. Hence, select Install from a specific location and click Next to get to Figure 1-5.

Please cho	ose your search and installation options.
💿 Searc	th for the best driver in these locations.
	ne check boxes below to limit or expand the default search, which includes local and removable media. The best driver found will be installed.
	Search removable media (floppy, CD-ROM)
~	Include this location in the search:
	C:\altera\91\quartus\drivers\usb-blaster\x32
🔿 Don't	search. I will choose the driver to install.
	se this option to select the device driver from a list. Windows does not guarantee iver you choose will be the best match for your hardware.
	<pre></pre>

Figure 1-5 Specify the location of the driver

Now, choose Search for the best driver in these locations and click Browse to get to the pop-up box in **Figure 1-6** Find the desired driver, which is at location C:\altera\12.0\quartus\drivers\usb-blaster. Click OK and then upon returning to **Figure 1-5** click Next. At this point the installation will commence, but a dialog box in **Figure 1-7** will appear indicating that the driver has not passed the Windows Logo testing. Click Continue Anyway.



Browse For Folder	
Select the folder that contains drivers for your hardware	э.
🖃 🚞 quartus	^
🗉 🧰 bin	
🗉 🧰 bin64	
🗉 🛅 common	=
🗉 🚞 cusp	
🖃 🧰 drivers	_
i386	
🗉 🧰 sentinel	
🖃 🧰 usb-blaster	
🗁 x32	
🗀 x64	<u>~</u>
To view any subfolders, click a plus sign above.	
OK Cancel	

#### Figure 1-6 Browse to find the location

1	The software you are installing for this hardware:
	Altera USB-Blaster
	has not passed Wil yows Logo testing to verify its compatibility with Windows XP. (Nell me why this testing is important.)
	Continuing your installation of this software may impai or destabilize the correct operation of your system either immediately or in the future. Microsoft strongly recommends that you stop this installation now and contact the hardware vendor for software that has passed Windows Logo testing.

Figure 1-7 There is no need to test the driver

The driver will now be installed as indicated in **Figure 1-8** Click Finish and you can start using the DE2i-150 board.







Figure 1-8 The driver is installed

### 1.3 What You Will Learn

In this tutorial you will perform the following tasks:

Create a design that causes LEDs on the development board to blink at a speed that is controlled by an input key—This design is easy to create and gives you visual feedback that the design works. Of course, you can use your DE2i-150 board to run other designs as well. For the LED design, you will write Verilog HDL code for a simple 32-bit counter, add a phase-locked loop (PLL) megafunction as the clock source, and add a 2-input multiplexer megafunction. When the design is running on the board, you can press an input switch to multiplex the counter bits that drive the output LEDs.

Becoming familiar with Quartus II design tools—This tutorial will not make you an expert (Please reference DE2i-150 tut\_quartus\_intro\_verilog document), but at the end, you will understand basic concepts about Quartus II projects, such as entering a design using a schematic editor and HDL, compiling your design, and downloading it into the FPGA on your DE2i-150 development board.

Develop a foundation to learn more about FPGAs—For example, you can create and download digital signal processing (DSP) functions onto a single chip, or build a multi-processor system, or create anything else you can imagine all on the same chip. You don't have to scour data books to find the perfect logic device or create your own ASIC. All you need is your computer, your imagination, and an Altera DE2i-150 FPGA development board.







# Chapter 2

# Assign The Device

You begin this tutorial by creating a new Quartus II project. A project is a set of files that maintain information about your FPGA design. The Quartus II Settings File (.qsf) and Quartus II Project File (.qpf) files are the primary files in a Quartus II project. To compile a design or make pin assignments, you must first create a project.

# 2.1 Assign The Device

 In the Quartus II software, select File > New Project Wizard. The Introduction page opens. See Figure 2-1

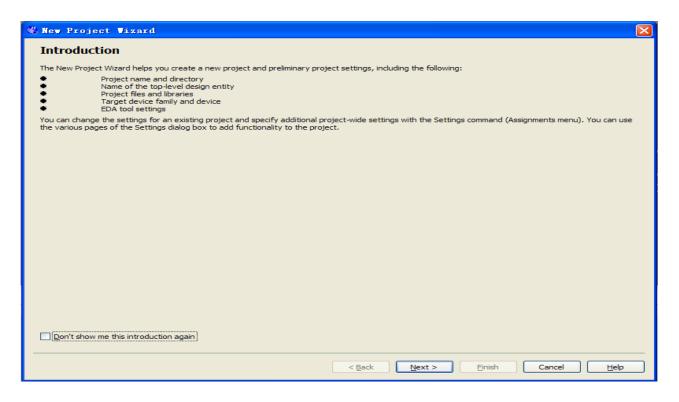


Figure 2-1 New Project Wizard introduction

- 2. Click Next.
- 3. Enter the following information about your project:



#### ter <mark>asIC</mark>

a. What is the working directory for this project? Enter a directory in which you will store your Quartus II project files for this design.

- b. For example, E:\My\_design\my\_first\_fpga.
- c. File names, project names, and directories in the Quartus II software cannot contain spaces.
- d. What is the name of this project? Type my\_first\_fpga.

e. What is the name of the top-level design entity for this project? Type my\_first\_fpga. See Figure 2-2.

New Project Vizard	
Directory, Name, Top-Level Entity [page 1 of 5]	
What is the working directory for this project?	
E:/My_design/my_first_fpga	
What is the name of this project?	
my_first_fpga	
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.	
my_first_fpga	
< Back Next > Einish Cancel	Help

Figure 2-2 Project information

f. Click Next.

g. You will assign a specific FPGA device to the design and make pin assignments. See **Figure 2-3**.



4 4 1	Devices:       All         Target device       Pin gount:         Auto device selected by the Fitter       Speed grade:         Specific device selected in 'Available devices' list       Name filter:         Other:       n/a         Ayailable devices:       Show advanced devices         Name       Core Voltage       LEs       User I/Os         GXB Transmitter Channel PMA       GXB Receiver Channel PMA		count: Any					Devices: All		
Image and     Image and       Speed grade:     Any       Speed grade:     Any       Name filter:     Image and the second devices       Image and the second devices     HardCopy compatible only	Target device       Specific device selected by the Fitter            • Auto device selected in 'Available devices' list        Specific device selected in 'Available devices' list             • Other: n/a        Mame filter:             • Auto device selected in 'Available devices' list        Show advanced devices             • Other: n/a           • State of the selected in 'Available devices' list             • Other: n/a           • Core Voltage           • LEs             • PEOGATISOC 2000           • Core Voltage           • LEs             • PEOGATISOC 2000           • LES           • User I/Os           • GXB Transmitter Channel PMA           • GXB Receiver Channel PM/             • PEOGATISOC 2000           • LES           • User I/Os           • GXB           • GXB Receiver Channel PM/             • PEOGATISOC 2000           • LES           • User I/Os           • GXB           • GXB Receiver Channel PM/             • PEOGATISOC 2007           • LES           • LES           • A             • PEOGATISOC 2007           • LES             • PE		ed grade: Any				ected by the Fitter	Target device		
Imagediate     Imagediate       Speed grade:     Any       Speed grade:     Any       Name filter:     Imagediate       Imagediate     Imagediate       Imagediate     Any       Imagediato     Any	Target device       Auto device selected by the Fitter <ul> <li>Auto device selected in 'Available devices' list</li> <li>Other: n/a</li> </ul> vgailable devices:               Show advanced devices                 Vgailable devices:               Solution:                 Vgailable devices:               Solution:                 Vgailable devices:               Solution:                 Vgailable devices:               Solution:                 Pe4CGX150CF2317               1.2V                 149760               287                 EP4CGX150DF27C7               12V                 149760             426               8                 EP4CGX150DF27C7               12V                 149760             426               8		ed grade: Any				ected by the Fitter	Target device		
Name filter:         Show advanced devices         HardCopy compatible only         rr I/Os         GXB Transmitter Channel PMA         GXB Receiver Channel PMA         4	O Auto device selected by the Fitter       Name filter:         Image: Specific device selected in 'Available devices' list       Image: Show advanced devices         Image: Other: n/a       Show advanced devices         Image: Specific device selected in 'Available devices' list       Image: Show advanced devices         Image: Specific device selected in 'Available devices' list       Image: Show advanced devices         Image: Specific device selected in 'Available devices' list       Image: Show advanced devices         Image: Specific device selected in 'Available devices' list       Image: Show advanced devices         Image: Specific device selected in 'Available devices' list       Image: Show advanced devices         Image: Specific device selected in 'Available devices' list       Image: Show advanced devices         Image: Specific device selected in 'Available devices' list       Image: Show advanced devices         Image: Specific device selected in 'Available devices' list       Image: Show advanced devices         Image: Specific device selected in 'Available devices' list       Image: Show advanced devices         Image: Specific device selected in 'Available devices' list       Image: Show advanced devices         Image: Specific device selected in 'Available' selected tevices       Image: Show advanced devices         Specific device selected tevice selected tevices       Image: Show advanced devices         Specific device selected tevice selecte		ne filter:				lected by the Fitter	and the second s		
Show advanced devices HardCopy compatible only      I/Os GXB Transmitter Channel PMA GXB Receiver Channel PMA	Or Auto device selected by the Fitter            • Specific device selected in 'Available devices' list             • Other: n/a          vailable devices:                • Core Voltage                 • If Yoo                • Provision: 2000                 • Provision: 2000                 • Provision: 2000                 • Provision: 2000                 • Provision: 2000          • Provision: 2000          • Provision: 2000          • Provision: 2000          • Provision: 2000          • Provision: 2000          • Provision: 2000          • Provision: 2000          • Provision: 2000          • Provision: 2000          • Provision: 2000	B HardCopy compatible only					lected by the Fitter	Q		
r I/Os GXB Transmitter Channel PMA GXB Receiver Channel PMA	• Specific device selected in 'Available devices' list         • Other: n/a         vailable devices:         • Show advanced devices         • HardCopy compatible only         • Show advanced devices         • Show advanced devices         • HardCopy compatible only         • Show advanced devices         • Show advanced devices         • HardCopy compatible only         • Show advanced devices         • Show advanced devices	s HardCopy compatible only	Show advanced devices				O Auto device selected by the Fitter			
r I/Os GXB Transmitter Channel PMA GXB Receiver Channel PMA	O Other: n/a           Valiable devices:           Name         Core Voltage         LEs         User I/Os         GXB Transmitter Channel PMA         GXB Receiver Channel PMA           FP4CGX150CF23I7         1.2V         149760         287         4         4           FP4CGX150DF27C7         1.2V         149760         426         8         8	s HardCopy compatible only	Show advanced devices		Specific device selected in 'Available devices' list					
4 4 1	Name         Core Voltage         LEs         User I/Os         GXB Transmitter Channel PMA         GXB Receiver Channel PMA           EP4CGX150CF23I7         1.2V         149760         287         4         4           EP4CGX150DF27C7         1.2V         149760         426         8         8           EP4CGX150DF27C8         1.2V         149760         426         8         8				it.	devices lis	selected in Available	<ul> <li>Specific device</li> </ul>		
	EP4CGX150DF27C7     1.2V     149760     426     8     8       EP4CGX150DF27C8     1.2V     149760     426     8     8		smitter Channel PMA	GXB			Lore Voltage			
	EP4CGX150DF27C8 1.2V 149760 426 8 8			4	1000					
8 8 1		8 1		8	426	149760	1.2V	EP4CGX150DF27C7		
8 8 1	EP4CGX150DF27I7 1.2V 149760 426 8 8	8 1		8	426	149760	1.2V	EP4CGX150DF27C8		
8 8 1		8 1		8	426	149760	1.2V	P4CGX150DF2717		
8 8 1	P4CGX150DF31C7 1.2V 149760 508 8 8	8 1		8	508	149760	1.2V	P4CGX150DF31C7		
	EP4CGX150DF31C8 1.2V 149760 508 8 8	8 1		8	508	149760	1.2V	P4CGX150DF31C8		
	a particular a la construction de la constru			8	508	149760	1.2V			
8 8 1 8 8 1 ×		1.5%								
8 8	EP4CGX150DF31C8 1.2V 149760 508 8 8	8 8 8		8 8 8	426 508 508	149760 149760 149760	1.2V 1.2V 1.2V	EP4CGX150DF27I7 EP4CGX150DF31C7 EP4CGX150DF31C8		
8 8		1995 - 1996 - 1996 - 1996 - 1996 - 1996 - 1996 - 1996 - 1996 - 1996 - 1996 - 1996 - 1996 - 1996 - 1996 - 1996 -			(Britished)	and and the of the second	A STATUTE TO A STATUTE OF A STA			
					177770		45.0 FB.0			
8 8 1	P4CGX150DF3117 1.2V 149760 508 8 8			8	508	149760	1.2V	P4CGX150DF31I7		
8 8 1		1.52								

Figure 2-3 Specify the Device Example

h. Click Finish.

4. When prompted, choose Yes to create the my\_first\_fpga project directory. You just created your first Quartus II FPGA project. See Figure 2-4.



🖑 Quartus II 32-bit - E:/My_design/my_first_fpga	/my_first_fpga - my_first_fpga
Eile Edit <u>V</u> iew Project <u>A</u> ssignments P <u>r</u> ocessing <u>T</u> ools <u>W</u> indow	Help 🖗 Search altera.com
🖸 🗃 🗐 🎒 🔏 🕅 🛍 🗠 🖓 [my_first_foga	
Project Navigator 양 문 ×	
Entity	
Cyclone IV GX: EP4CGX150DF31C7     → my_first_fpga	
Kierarchy B Files d <sup>g</sup> Design Units	ADERA.
Tasks @ & X	OTTA DETTO TT
Flow: Compilation	<b>OUARTUS</b> <sup>®</sup> II
Task Ö Time 🛆	Version 12.0
Compile Design	
Analysis & Synthesis     Edit Settings	
View Report	March 1
Analysis & Elaboration	View Quartus II Information
Partition Merge	2 Documentation
× × V < <search>&gt; V</search>	
× V < <search>&gt; V 7 9 Type Message</search>	
System / Processing / Extra Info / Info / Warning / Critical W Location:	Varning / Error / Suppressed / Flag /
	0% 00:00:0

Figure 2-4 my\_first\_fpga project

# Chapter 3



# 3.1 Add a PLL Megafunction

This section describes How to Add a PLL Megafunction

In the design entry step you create a schematic or Block Design File (.bdf) that is the top-level design. You will add library of parameterized modules (LPM) functions and use Verilog HDL code to add a logic block. When creating your own designs, you can choose any of these methods or a combination of them.

1. Choose File > New > Block Diagram/Schematic File (see **Figure 3-1** to create a new file, Block1.bdf, which you will save as the top-level design.

🐝 New	
<ul> <li>New Quartus II Project</li> <li>Design Files <ul> <li>AHDL File</li> <li>Block Diagram/Schematic File</li> <li>EDIF File</li> <li>Qsys System File</li> <li>State Machine File</li> <li>SystemVerilog HDL File</li> <li>Td Script File</li> <li>VHDL File</li> <li>Verlog HDL File</li> <li>Hexadecimal (Intel-Format) File</li> <li>Memory Initialization File</li> <li>Verification/Debugging Files</li> <li>In-System Sources and Probes File</li> <li>Logic Analyzer Interface File</li> <li>SignalTap II Logic Analyzer File</li> <li>Other Files</li> <li>AHDL Include File</li> <li>Block Symbol File</li> <li>Chain Description File</li> <li>Synopsys Design Constraints File</li> <li>Text File</li> </ul> </li> </ul>	
OK Cancel Help	

Figure 3-1 New BDF

2. Click OK.



ter asIC

- 3. Choose File > Save As and enter the following information.
  - File name: my\_first\_fpga
  - Save as type: Block Diagram/Schematic File (\*.bdf)
- 4. Click Save. The new design file appears in the Block Editor (see Figure 3-2).

🕊 Quartus II 32-bit - E:/My_design/my_first_fp;		
File Edit View Project Assignments Processing Tools Window		Search altera.com
🗋 🕞 🖬 🎒 🎒 👗 🖻 🛍 🗠 📿 🕅 my_first_foga	<u>▼ ¥ 2 2 8 8 8 0 ► 8 10 0 18 1</u>	🔍 🕘 👗 🧶 🗣
Project Navigator	📸 my_first_fpga.bdf 🗵	
Entity	□□□□□\\\<	> ➡ > ■ ■ > ■ >
Cyclone IV GX: EP4CGX150DF31C7 → my_first_fpga ຢ້ອ		
✓     ✓       ✓     Hierarchy       E     Files       Ø     Ø       Ø     Ø       ✓     Customize		=
Task     Time       Image: Compile Design     Image: Compile Design       Image: Compile Design </td <td></td> <td></td>		
Image: Search >>       Imag	< <u>1</u>	
System / Processing / Extra Info / Info / Warning / Critica	Warning / Error / Suppressed / Flag /	
S Location:		Locate
		104, 201 0% 00:00:00

#### Figure 3-2 Bank BDF

- 5. Add HDL code to the blank block diagram by choosing File > New > Verilog HDL File.
- 6. Click OK to create a new file Verilog1.v, which you will save as simple\_counter.v.
- 7. Select File > Save As and enter the following information (see Figure 3-3).
  - File name: simple\_counter.v
  - Save as type: Verilog HDL File (\*.v, \*.vlg, \*.verilog)





另存为					? 🔀
保存在 (L): 我最近的文档 「」 東面 文档 一 文 我的文档 一 一 文 和 一 一 、 和 一 一 、 和 一 一 、 和 一 一 、 和 一 一 、 和 一 一 、 和 一 一 、 和 一 一 、 和 一 一 、 和 一 一 一 和 一 一 和 一 一 和 一 一 和 一 一 和 一 一 和 一 一 和 一 一 和 一 一 和 一 一 和 一 一 和 一 一 和 一 一 和 一 一 和 一 一 和 一 一 和 一 一 和 一 一 和 一 一 和 一 一 和 一 一 和 一 一 和 一 一 和 一 一 和 一 一 和 一 一 和 一 一 和 一 一 和 一 一 和 一 一 和 一 一 一 和 一 一 和 一 一 一 一 一 一 一 一 一 一 一 一 一	my_first_1	Ēpga	-	È d'	
	文件名 (2): 保存类型 (2):	simple counter v       Verilog HDL Files (       I Add file to cument pro		▼ [	保存( <u>S</u> ) 取消

Figure 3-3 Saving the Verilog HDL file

The resulting empty file is ready for you to enter the Verilog HDL code.

8. Type the following Verilog HDL code into the blank simple\_counter.v file (see **Figure 3-4** The Verilog File of simple\_counter.v).

//It has a single clock input and a 32-bit output port

module simple\_counter (

CLOCK\_50,

counter\_out

);

input CLOCK\_50;

output [31:0] counter\_out;

reg [31:0] counter\_out;





```
always @ (posedge CLOCK_50)
```

// on positive clock edge

begin

counter\_out <= #1 counter\_out + 1;// increment counter</pre>

end

endmodule

// end of module counter

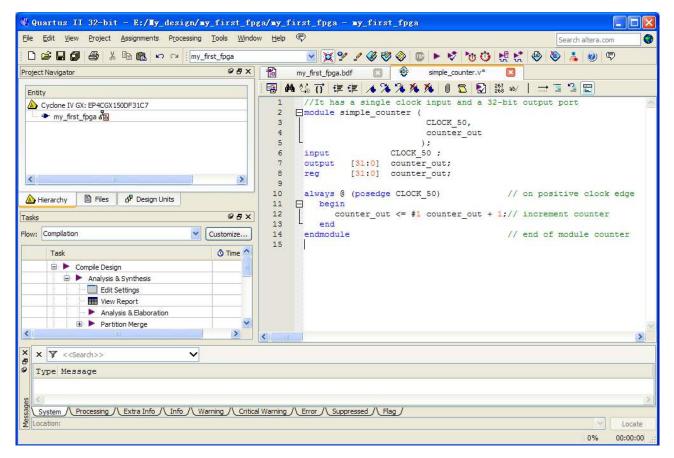


Figure 3-4 The Verilog File of simple\_counter.v

9. Save the file by choosing File > Save, pressing Ctrl + s, or by clicking the floppy disk icon.

10. Choose File > Create/Update > Create Symbol Files for Current File to convert the simple\_counter.v file to a Symbol File (.sym).You use this Symbol File to add the HDL code to your BDF schematic.

The Quartus II software creates a Symbol File and displays a message (see Figure 3-5).







Figure 3-5 Create Symbol File was Successful

- 11. Click OK.
- 12. To add the simple\_counter.v symbol to the top-level design, click the my\_first\_fpga.bdf tab.
- 13. Choose Edit > Insert Symbol.
- 14. Double-click the Project directory to expand it.
- 15. Select the newly created simple\_counter symbol by clicking it's icon.

You can also double-click in a blank area of the BDF to open the Symbol dialog box

🔓 Symbol	
Libraries:	simple_counter CLOCK_50 counter_out[31.0]
Name:	inst
simple_counter	
Repeat-insert mode         Insert symbol as block         Launch MegaWizard Plug-In	
MegaWizard Plug-In Manager	
	OK Cancel

Figure 3-6 Adding the Symbol to the BDF

16. Click OK.



#### ter <mark>asIC</mark>

17. Move the cursor to the BDF grid; the symbol image moves with the cursor. Click to place the simple\_counter symbol onto the BDF. You can move the block after placing it by simply clicking and dragging it to where you want it and releasing the mouse button to place it. See **Figure 3-7**.

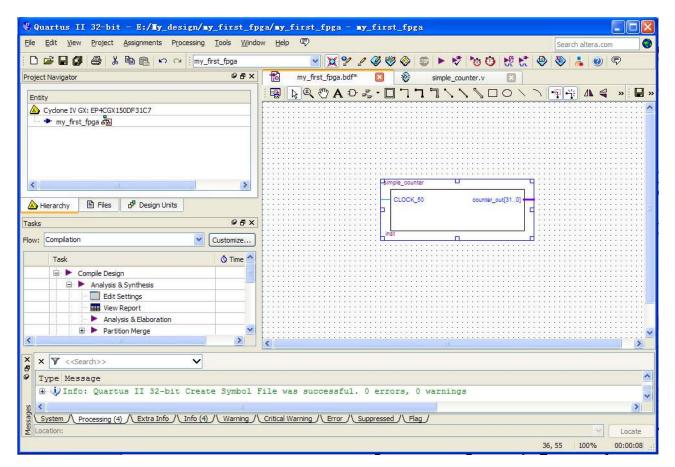


Figure 3-7 Placing the simple\_counter symbol

18. Press the Esc key or click an empty place on the schematic grid to cancel placing further instances of this symbol.

19. Save your project regularly.

Using Quartus Add a PLL Megafunction

Megafunctions, such as the ones available in the LPM, are pre-designed modules that you can use in FPGA designs. These Altera-provided megafunctions are optimized for speed, area, and device family. You can increase

Efficiency by using a megafunction instead of writing the function yourself. Altera also provides more complex functions, called MegaCore functions, which you can evaluate for free but require a license file for use in production designs. This tutorial design uses a PLL clock source to drive a simple counter. A PLL uses the on-board oscillator (DE2i-150 Board is 50 MHz) to create a constant clock frequency as the input to the counter. To create the clock source, you will add a



### ter asIC

pre-built LPM megafunction named ALTPLL.

1. Choose Edit > Insert Symbol or click Add Symbol on the toolbar---

Click Megawizard Plug-in Manager. The MegaWizard® Plug-In Manager appears (see Figure 3-8).

🐇 IegaViza	ard Plug-In Manager [page 1]	×
*	The MegaWizard Plug-In Manager helps you create or modify design files that contain custom variations of megafunctions. Which action do you want to perform? () Create a new custom megafunction variation () Edit an existing custom megafunction variation () Cogy an existing custom megafunction variation () Cogy an existing custom megafunction variation () Copyright (C) 1991-2012 Altera Corporation	
	Cancel < Back Next > Einish	

Figure 3-8 Mega Wizard Plug-In Manager

3. Click Next.

4. In MegaWizard Plug-In Manager [page 2a], specify the following selections (see Figure 3-9):

a. Choose I/O > ALTPLL.

b. Under Which device family will you be using? Choose the Cyclone IV GX for DE2i-150 development board.

c. Under Which type of output file do you want to create? Choose Verilog HDL.

d. Under What name do you want for the output file? Type pll at the end of the already created directory name.

e. Click Next.





	Select a megafunction from the list below	Which device family will you be using? Which type of output file do you want to AHDL YHDL Verilog HDL What name do you want for the output E:/My_design/my_first_fpga/pll.v Output files will be generated using the Return to this page for another crea Note: To compile a project successfully in	file?  classic file structure te operation n the Quartus II software, your design a library specified in the Libraries page of a library specified in the Libraries page
Altera PLL Reconfig v12.0	ALTREMOTE_UPDATE ALTTEMP_SENSE Altera CRCERROR Verify v12 Altera PLL Reconfig v12.0	of the Settings dialog box (Assignments	menu).

Figure 3-9 MegaWizard Plug-In Manager [page 2a] Selections

5. In the MegaWizard Plug-In Manager [page 3 of 14] window, make the following selections (see Figure 3-10).

a. Confirm that the Currently selected device family option shows the device family that corresponds to the development board you are using.

- b. The device speed grade choose 6 for DE2i-150.
- c. Set the frequency of the inclock0 input 50 MHz.
- d. Click Next.



### ter asIC

TegaTizard Plug-In Tanager [page	3 of 14]
Parameter PLL 3Output Clocks	4 EDA 5 Summary
General/Modes > Inputs/Lock > Bandwidth	/SS > Clock switchover > Currently selected device family:
pll	Match project/default
Cik Ratio Ph (dg) DC (%) o0 1/1 0.00 50.00 Cyclone IV GX	General         Which device speed grade will you be using?         0         Use military temperature range devices only         What is the frequency of the indk0 input?         Set up PLL in LVDS mode         Data rate:         Not Available         Mbps
	PLL Type         Which PLL type will you be using?         Fast PLL         Operation Mode         How will the PLL outputs be generated?         Output the feedback path inside the PLL
	Ouse the recodack path inside the PLL     Output for an external feedback (External Feedback Mode)     Which output clock will be compensated for?
	Cancel < <u>B</u> ack <u>N</u> ext > <u>F</u> inish

Figure 3-10 MegaWizard Plug-In Manager [page 3 of 14] Selections

6. Turn off all options on MegaWizard page 4. As you turn them off, pins disappear from the PLL block's graphical preview. See **Figure 3-11** for an example.



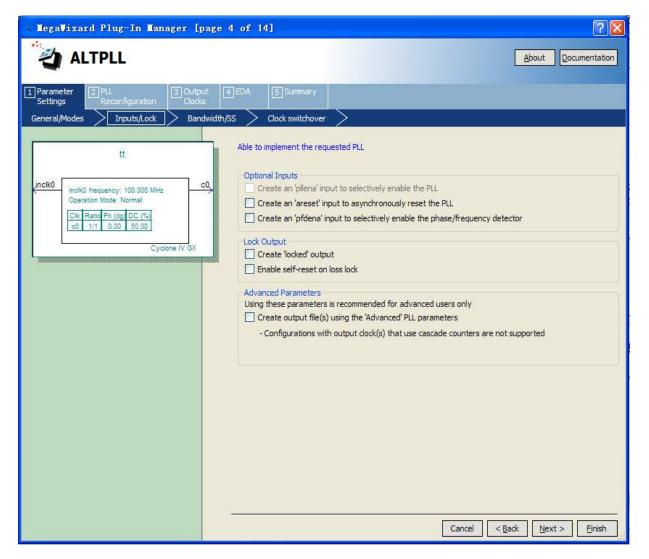


Figure 3-11 MegaWizard Plug-In Manager [page 4 of 14] Selections

7. Click Next four times, In the MegaWizard Plug-In Manager [page 8 of 14] window, make the following selections (see Figure 3-12).

Clock Division Settings input 10



### ter <mark>asIC</mark>

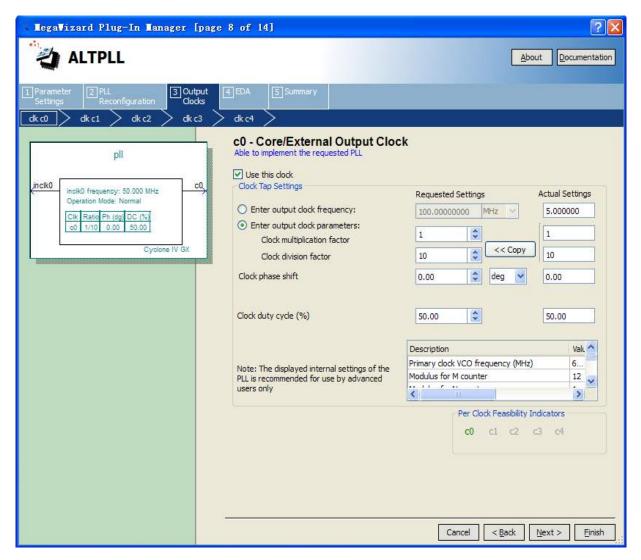


Figure 3-12 MegaWizard Plug-In Manager [page 8 of 14] Selections

- 9. Click Next.
- 10. The wizard displays a summary of the files it creates (see Figure 3-13). Click Finish .

#### ter asIC

• TegaVizard Plug-In Manager [pag	e 14 of 14]	?×
		<u>About</u> <u>Documentation</u>
Parameter 2 PLL Settings Reconfiguration Clocks	4 EDA 5 Sum	mary
pll jncik0 incik0 frequency: 50.000 MHz C0	green checkmark indi checkbox is maintaine	wish to generate. A gray checkmark indicates a file that is automatically generated, and a cates an optional file. Click Finish to generate the selected files. The state of each ed in subsequent MegaWizard Plug-In Manager sessions. g-In Manager creates the selected files in the following directory: st. foga\
Operation Mode: Normal		
Clk Ratio Ph (dg) DC (%)	File	Description
oD 1/10 0.00 50.00	⊻ pll,v	Variation file
	⊻ pll.ppf	PinPlanner ports PPF file
Cyclone IV GX	pll.inc	AHDL Include file VHDL component declaration file
	pll.cmp	Quartus II symbol file
	pll_inst.v	Instantiation template file
	pll bb.v	Verilog HDL black-box file
		Cancel < Back Next > Finish

Figure 3-13 Wizard-Created Files

The Symbol window opens, showing the newly created PLL megafunction. See Figure 3-14.



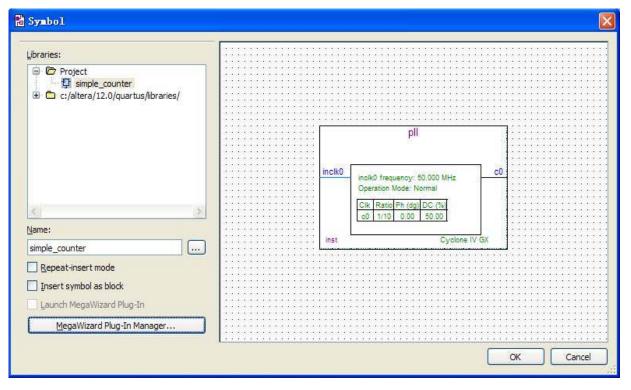


Figure 3-14 PLL Symbol

11. Click OK and place the pll symbol onto the BDF to the left of the simple\_counter symbol. You can move the symbols around by holding down the left mouse button, helping you ensure that they line up properly. See **Figure 3-15**.



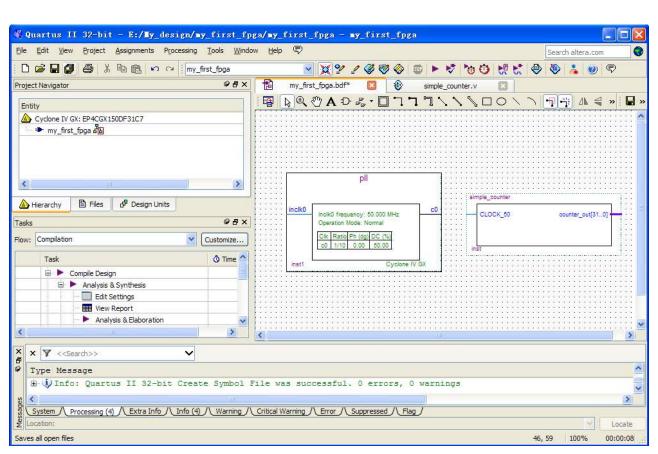


Figure 3-15 Place the PLL Symbol

12. Move the mouse so that the cursor (also called the selection tool) is over the pll symbol's c0 output pin. The orthogonal node tool (cross-hair) icon appears.

13. Click and drag a bus line from the c0 output to the simple\_counter clock input. This action ties the pll output to the simple\_counter input (see Figure 3-16).

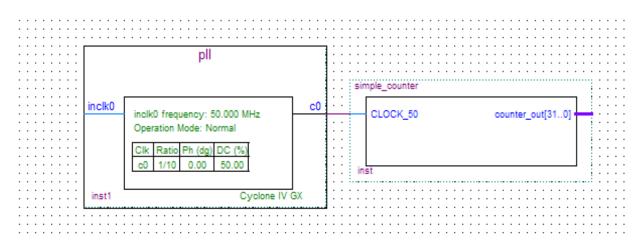


Figure 3-16 Draw a Bus Line connect pll c0 port to simple\_counter CLOCK\_50 port

14. Add an input pin and an output bus with the following steps:



terasic



- a. Choose Edit > Insert Symbol.
- b. Under Libraries, select quartus/libraries > primitives > pin >input. See Figure 3-17

#### c. Click OK

If you need more room to place symbols, you can use the vertical and horizontal scroll bars at the edges of the BDF window to view more drawing space.

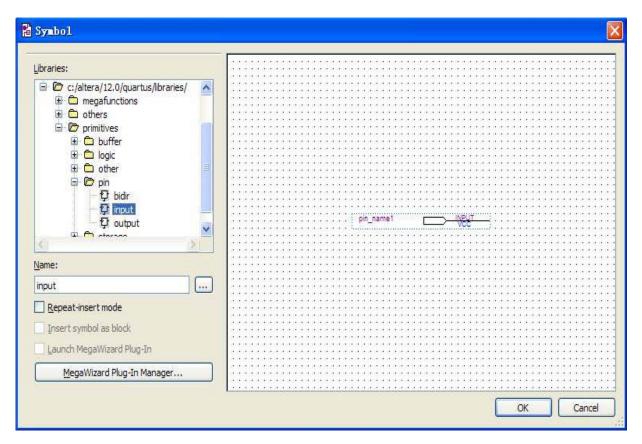
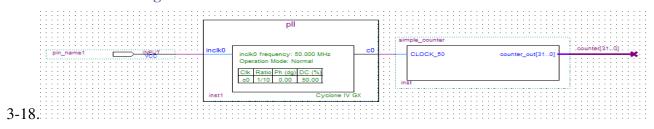


Figure 3-17 Input pin symbol

d. Place the new pin onto the BDF so that it is touching the input to the pll symbol.

e. Use the mouse to click and drag the new input pin to the left; notice that the ports remain connectedasshowninFigure









f. Change the pin name by double-clicking pin\_name and typing CLOCK\_50 (see **Figure 3-19**). This name correlates to the oscillator clock that is connected to the FPGA.

g. Using the Orthogonal Bus tool, draw a bus line connected on one side to the simple\_counter output port, and leave the other end unconnected at about 4 to 8 grid spaces to the right of the simple\_counter.

ŧ	Pin Prop	erti	ies	×
	General F	Forma		
	<u>P</u> in name(s):	:	CLOCK_50	
	<u>D</u> efault valu	e:	VCC VC	
			OK Cancel Help	

Figure 3-19 Change the input port name

h. Right-click the new output bus line and choose Properties.

i. Using the Orthogonal Bus tool, draw a bus line connected on one side to the simple\_counter output port, and leave the other end unconnected at about 6 to 8 grid spaces to the right of the simple\_counter.

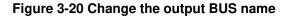
j. Type counter [31..0] as the bus name (see **Figure 3-20**). The notation [X ..Y] is the Quartus II method for specifying the bus width in BDF schematics, where X is the most significant bit (MSB) and Y is the least significant bit (LSB).

k. Click OK. Figure 3-21 shows the BDF.



ter <mark>asIC</mark>

Bus Prope	erties	×
General	ont Format	
<u>N</u> ame:	counter[310] name in block design file	
	确定	取消



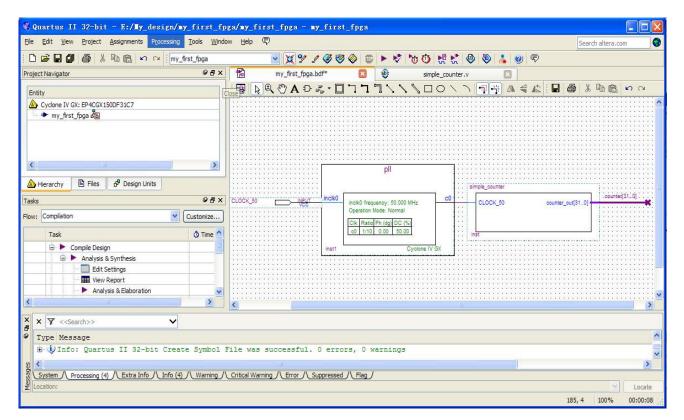


Figure 3-21 BDF

# 3.2 Add a Multiplexer

This design uses a multiplexer to route the simple\_counter output to the LED pins on the DE2i-150 development board. You will use the MegaWizard Plug-In Manager to add the multiplexer, lpm\_mux. The design multiplexes two variations of the counter bus to four LEDs on the DE2i-150 development board.

- 1. Choose Edit > Insert Symbol.
- 2. Click Megawizard Plug-in Manager.
- 3. Click Next.
- 4. Choose Installed Plug-Ins > Gates > LPM\_MUX.

5. Choose the device family that corresponds to the device on the development board you are using, choose Verilog HDL as the output file type, and name the output file counter\_bus\_mux.v (see Figure 3-22).

6. Click Next.





Which megafunction would you like to customi Select a megafunction from the list below	<ul> <li>Which device family will you be using?</li> <li>Which type of output file do you want t</li> <li>AHDL</li> <li>VHDL</li> <li>Verilog HDL</li> <li>Verilog HDL.</li> <li>What name do you want for the output</li> <li>E:/My_design/my_first_fpga/counter_I</li> <li>Output files will be generated using the</li> <li>Return to this page for another creations of the Settings dialog box (Tools menu), or of the Settings dialog box (Assignments</li> <li>Your current user library directories are</li> </ul>	: file? bus_mux.v dassic file structure ate operation in the Quartus II software, your a library specified in the Libraries a library specified in the Libraries s menu).	s page of
-------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------

Figure 3-22 Selecting Ipm\_mux

- 7. Under How many 'data' inputs do you want? Select 2 inputs (default).
- 8. Under How 'wide' should the data input and result output be? Select 4 (see Figure 3-23).

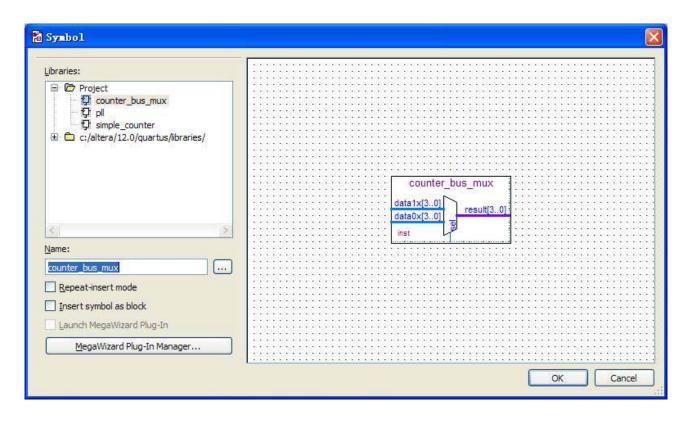
ter asIC

▲ TegaVizard Plug-I	n Manager [page 3 of 5]	? 🛛
👌 LPM_MU	x	About Documentation
1 Parameter Settings	3 Summary	
counter_bus_mux data1x[30] data0x[30]	Currently selected device fa	mily: Cyclone IV GX
¥	How many 'data' inputs do you want? 2 How wide should the 'data' input and the 'result' output buses be? 4 Do you want to pipeline the multiplexer?	✓ bits
	No     Yes, I want an output latency of     Create an asynchronous Clear input	dock cycles
	Create a Clock Enable input	
Resource Usage 1 lpm_mux	Cancel	< <u>Back</u>

Figure 3-23 Ipm\_mux settings

- 9. Click Next.
- 10. Click Finish twice. The Symbol window appears (see Figure 3-24 for an example).







#### 11. Click OK

12. Place the counter\_bus\_mux symbol below the existing symbols on the BDF. See Figure 3-25

	pll	]	
	no#0	simple_counter	.countet[31.0]
CLOCK_50	Inclk0         inclk0 frequency: 50.000 MHz         c0           Operation Mode: Normal         Clk         Ratio         Ph (dg)         DC (%)         c0         1/10         0.00         50.00         Inclk0         Inclk0	CLOCK_50 counter_out[310]	
	inst1 Cyclone IV GX		
	COU data1x[3	nter_bus_mux 30] result[30]	
	data0x[3 inst3	30]	· · · · · · · · · · · · · · · · · · ·

Figure 3-25 Place the lpm\_mux symbol

13. Add input buses and output pins to the counter\_bus\_mux symbol as follows:



### ter <mark>asIC</mark>

a. Using the Orthogonal Bus tool, draw bus lines from the data1x[3..0] and data0x[3..0]

Input ports to about 8 to 12 grid spaces to the left of counter\_bus\_mux.

b. Draw a bus line from the result [3..0] output port to about 4 to 8 grid spaces to the right of counter\_bus\_mux.

- c. Right-click the bus line connected to data1x[3..0] and choose Properties.
- d. Name the bus counter[26..23], which selects only those counter output bits to connect to

the four bits of the data1x input.

Because the input busses to counter\_bus\_mux have the same names as the output bus from simple\_counter, (counter[x .. y]) the Quartus II software knows to connect these busses.

- e. Click OK.
- f. Right-click the bus line connected to data0x[3..0] and choose Properties.

g. Name the bus counter [24..21], which selects only those counter output bits to connect to the four bits of the data1x input.

h. Click OK. Figure 3-26 shows the renamed buses.

.counter[28.	counter_bus_mux data1x[30] result[30]	.result[30].
.counted[24.	data0x[30]	
	Figure 3-26 Renamed counter_bus_mux I	Bus Lines

If you have not done so already, save your project file before continuing.

- 14. Choose Edit > Insert Symbol.
- 15. Under Libraries, double-click quartus/libraries/ > primitives > pin > output (see Figure 3-27).





Libraries:	
	OUTPUT
8	
Name:	
output	
Repeat-insert mode     Insert symbol as block     Launch MegaWizard Plug-In	
MegaWizard Plug-In Manager	

Figure 3-27 choose an output pin

- 16. Click OK.
- 17. Place this output pin so that it connects to the counter\_bus\_mux result [3..0] bus output line.
- 18. Rename the output pin as LEDG [3..0] as described in steps 13 c and d. (see Figure 3-28).

counter_bus_mux           data1x[30]           result[30]           .result[30]           .result[30]           .inst3			
data0x[30] inst3	counter_bus_mux	]	
data0x[30]	data1x[30] result[30]		
	data0x[30]		

Figure 3-28 Rename the output pin

- 19. Attach an input pin to the multiplexer select line using an input pin:
- a. Choose Edit > Insert Symbol.
- b. Under Libraries, double-click quartus/libraries/ > primitives > pin > input.
- c. Click OK.



ter asIC

- 20. Place this input pin below counter\_bus\_mux.
- 21. Connect the input pin to the counter\_bus\_mux sel pin.
- 22. Rename the input pin as KEY [0] (see Figure 3-29).

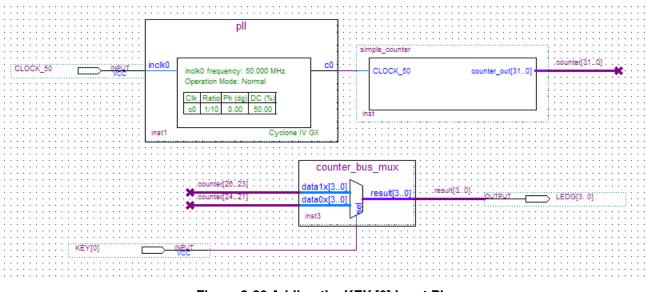


Figure 3-29 Adding the KEY [0] Input Pin

You have finished adding symbols to your design. You can add notes or information to the project as text using the Text tool on the toolbar (indicated with the A symbol). For example, you can add the label "OFF = SLOW, ON = FAST" to the KEY [0] input pin and add a project description, such as "My First FPGA Project."

#### 3.3 Assign the Pins

In this section, you will make pin assignments. Before making pin assignments, perform the following steps:

1. Choose Processing > Start > Start Analysis & Elaboration in preparation for assigning pin locations.

2. Click OK in the message window that appears after analysis and elaboration completes.

To make pin assignments that correlate to the KEY [0] and CLOCK\_50 input pins and LEDG[3..0] output pin, perform the following steps:

 Choose Assignments > Pins, which opens the Pin Planner, a spreadsheet-like table of specific pin assignments. The Pin Planner shows the design's six pins. See Figure 3-30





	Edit View Processing	Tools <u>W</u> indow <u>H</u> elp	Ţ			Search alter	a.com	C
	asks P D X							
×	K Named: *	» Edit: 🗶 🗸				Filter: Pi	ns: all	1
) 6 9	Node Name	Direction	Location	I/O Bank	VREF Group	I/O Standard	Reserved	1
	CLOCK_50	Input				2.5 V (default)		1
	KEY[0]	Input				2.5 V (default)		
	DEDG[3]	Output				2.5 V (default)		
	DEDG[2]	Output				2.5 V (default)		
	LEDG[1]	Output				2.5 V (default)		
		Output				2.5 V (default)		
All Dine	E 🐵 LEDG[0]	output						

#### Figure 3-30 Pin Planner Example

2. In the Location column next to each of the six node names, add the coordinates (pin numbers) as shown in Table 3-1 for the actual values to use with your DE2i-150 board.

Pin Name	FPGA Pin Location
KEY[0]	AA26
LEDG[3]	F26
LEDG[2]	F27
LEDG [1]	AB25
LEDG [0]	AA25
CLOCK_50	AJ16

#### Table 3-1 Pin Information Setting

Double-click in the Location column for any of the six pins to open a drop-down list and type the location shown in the table alternatively, you can select the pin from a drop-down list. For example, if you type F1 and press the Enter key, the Quartus II software fills in the full PIN\_F1 location name for you. The software also keeps track of corresponding FPGA data such as the I/O bank and VREF Group. Each bank has a distinct color, which corresponds to the top-view wire bond drawing in the upper right window. See **Figure 3-31**.





e	Edit View Processing	Tools Window He	sp 🖓			Search alte	ra.com
	Tasks						
>		🔊 Edit: 🗶 🗸 PIN	_AA25			Filter: P	ins: all
	Node Name	Direction	Location	I/O Bank	VREF Group	I/O Standard	Reserved
7 5	CLOCK_50	Input	PIN_AJ16	4	B4_N2	2.5 V (default)	
8	KEY[0]	Input	PIN_AA26	5	B5_N2	2.5 V (default)	
	LEDG[3]	Output	PIN_F26	6	B6_N0	2.5 V (default)	
	LEDG[2]	Output	PIN_F27	6	B6_N0	2.5 V (default)	
	и 🖾 LEDG[1]	Output	PIN_AB25	5	B5_N2	2.5 V (default)	
i		Output	PIN_AA25	5	B5_N2	2.5 V (default)	
	₹ <						>

Figure 3-31 Completed Pin Planning Example

Now, you are finished creating your Quartus II design!

## 3.4 Create a Default TimeQuest SDC File

Timing settings are critically important for a successful design. For this tutorial you will create a basic Synopsys Design Constraints File (.sdc) that the Quartus II TimeQuest Timing Analyzer uses during design compilation. For more complex designs, you will need to consider the timing requirements more carefully.

To create an SDC, perform the following steps:

- 1. Open the TimeQuest Timing Analyzer by choosing Tools > TimeQuest Timing Analyzer.
- 2. Choose File > New SDC file. The SDC editor opens.
- 3. Type the following code into the editor:

create\_clock -name "CLOCK\_50" -period 20.000ns [get\_ports {CLOCK\_50}]

derive\_pll\_clocks

derive\_clock\_uncertainty

4. Save this file as my\_first\_fpga.sdc (see Figure 3-32)



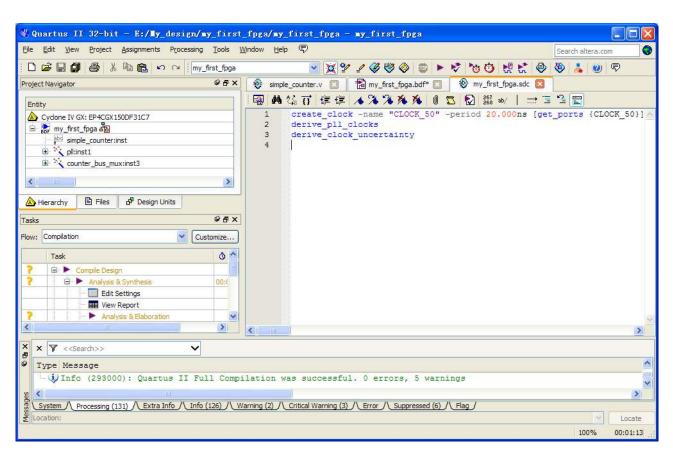


Figure 3-32 Default SDC

Naming the SDC with the same name as the top-level file except for the .sdc extension causes the Quartus II software to using this timing analysis file automatically by default. If you used another name, you would need to add the SDC to the assignments file list.



ter as IC



#### **Chapter 4**

# Compile and Verify Your Design

After creating your design you must compile it. Compilation converts the design into a bitstream that can be downloaded into the FPGA. The most important output of compilation is an SRAM Object File (.sof), which you use to program the device. The software also generates other report files that provide information about your code as it compiles.

### 4.1 Compile Your Design

If you want to store .SOF in memory device (such as flash or EEPROMs), you must first convert the SOF to a file type specifically for the targeted memory device.

Now that you have created a complete Quartus II project and entered all assignments, you can compile the design.

In the Processing menu, choose Start Compilation or click the Play button on the toolbar.

If you are asked to save changes to your BDF, click Yes.

While compiling your design, the Quartus II software provides useful information about the compilation (see Figure 4-1).



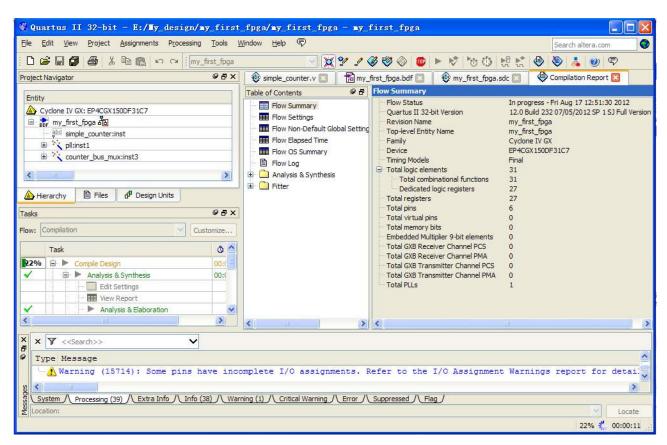


Figure 4-1 Compilation Message for project

When compilation is complete, the Quartus II software displays a message. Click OK to close the message box.

The Quartus II Messages window displays many messages during compilation. It should not display any critical warnings; it may display a few warnings that indicate that the device timing information is preliminary or that some parameters on the I/O pins used for the LEDs were not set. The software provides the compilation results in the Compilation Report tab as shown in **Figure 4-2**.



ter as IC



Flow Status	Successful - Fri Aug 17 12:52:25 2012
Quartus II 32-bit Version	12.0 Build 232 07/05/2012 SP 1 SJ Full Version
Revision Name	my_first_fpga
Fop-level Entity Name	my_first_fpga
Family	Cyclone IV GX
Device	EP4CGX150DF31C7
Fiming Models	Final
Fotal logic elements	31 / 149,760 ( < 1 % )
Total combinational functions	31 / 149,760 ( < 1 % )
Dedicated logic registers	27 / 149,760 ( < 1 % )
Total registers	27
Total pins	6 / 508 (1%)
Total virtual pins	0
Total memory bits	0 / 6,635,520 ( 0 % )
Embedded Multiplier 9-bit elements	0 / 720 ( 0 % )
otal GXB Receiver Channel PCS	0/8(0%)
otal GXB Receiver Channel PMA	0/8(0%)
otal GXB Transmitter Channel PCS	0/8(0%)
Total GXB Transmitter Channel PMA	0/8(0%)
Total PLLs	1/8(13%)

Figure 4-2 Compilation Report Example

## 4.2 Program the FPGA Device

After compiling and verifying your design you are ready to program the FPGA on the development board. You download the SOF you just created into the FPGA using the USB-Blaster circuitry on the board. Set up your hardware for programming using the following steps:

a) Connect the power supply cable to your board and to a power outlet.

b) For the DE2i-150 board, connect the USB-Blaster (included in your development kit) to J9 and the USB cable to the USB-Blaster. Connect the other end of the USB cable to the host computer.

Refer to the getting started user guide for detailed instructions on how to connect the cables.

c) Turn the DE2i-150 board on using the on/off switch.

Program the FPGA using the following steps.

1. Choose Tools > Programmer. The Programmer window opens. See Figure 4-3.





Hardware Setup	USB-Blaster [USB-0]	Mode:	JTAG	*	Progress:			
Enable real-time ISP	to allow background progra	amming (for MAX II and M	1AX V devices)					
M Start	File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank- Check	Examine
Stop	my_first_fpga.sof	EP4CGX150DF31	00724DE5	FFFFFFF				
Auto Detect								
Adio Delect								
X Delete								
	1							
Add File	1							
Change File	3							
Change File	<		100					
Change File	5							
Add File Change File Save File Add Device	<u></u>		. un					

Figure 4-3 Programmer Window

2. Click Hardware Setup.

3. If it is not already turned on, turn on the USB-Blaster [USB-0] option under currently selected hardware. See Figure 4-4.

Ð	Hardware Setup			
	Hardware Settings JTAG Setting Select a programming hardware setu hardware setup applies only to the co Currently selected hardware: USB Available hardware items	p to use when progr		s. This programming
	Hardware	Server	Port	Add Hardware
	USB-Blaster	Local	USB-0	Remove Hardware
				Close

Figure 4-4 Hardware Setting

4. Click Close.



#### ter <mark>asIC</mark>

- 5. If the file name in the Programmer does not show my\_first\_fpga.sof, click Add File.
- 6. Select the my\_first\_fpga.sof file from the project directory (see Figure 4-5).

Hardware Setup	USB-Blaster [USB-0]	Mode:	JTAG	~	Progress:	10	0% (Succes	isful)
] Enable real-time ISP	to allow background prog	ramming (for MAX II and	MAX V devices)					
M Start	File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank- Check	Examine
Stop	my_first_fpga.sof	EP4CGX150DF31	00724DE5	FFFFFFF	<b>V</b>			
Auto Detect								
Auto Detect Change File	<		- Think					
Auto Detect  Delete  Add File	<		pilm:					

Figure 4-5 Downloading Complete

Congratulations, you have created, compiled, and programmed your first FPGA design! The compiled SRAM Object File (.sof) is loaded onto the FPGA on the development board and the design should be running.

### 4.3 Verify The Hardware

When you verify the design in hardware, you observe the runtime behavior of the FPGA hardware design and ensure that it is functioning appropriately.

Verify the design by performing the following steps:

1. Observe that the four development board LEDs appear to be advancing slowly in a binary count pattern, which is driven by the simple\_counter bits [26..23].

The LEDs are active low, therefore, when counting begins all LEDs are turned on (the 0000 state).

2. Press and hold KEY [0] on the development board and observe that the LEDs advance more quickly. Pressing this KEY causes the design to multiplex using the faster advancing part of the



ter <mark>asIC</mark>

counter (bits [24..21]).

3. If other LEDs emit faintness light, Choose Assignments > Device. Click Device and Options. See Figure 4-6.

Device family				Show in 'Available devices' list	
Eamily: Cyclone IV	/ GX		~	Package: Any	~
Devices: All			1	Pin count: Any	~
				Canad available Annu	
Target device				Speed grade: Any	
CONTRACTOR OF THE				Name filter:	
<u>Auto device sel</u>	ected by the Fitter			Show advanced devices HardCo	any compatible only
Specific device	selected in 'Available	devices' lis	it 🛛	Silow advanced devices I ha⊑det	эрү сотрацые опту
O Other: n/a					
				Device and Pin Options	
vailable devices:					
Name	Core Voltage	LEs	User I/Os	GXB Transmitter Channel PMA	GXB Receiver Chan
P4CGX110DF31C8	1.2V	109424	508	8	8
P4CGX110DF3117	1.2V	109424	508	8	8
P4CGX150CF23C7	1.2V	149760	287	4	4
P4CGX150CF23C8	1.2V	149760	287	4	4
P4CGX150CF23I7	1.2V	149760	287	4	4
P4CGX150DF27C7	1.2V	149760	426	8	8
P4CGX150DF27C8	1.2V	149760	426	8	8
P4CGX150DF27I7	1.2V	149760	426	8	8
P4CGX150DF31C7	1.2V	149760	508	8	8
P4CGX150DF31C8	1.2V	149760	508	8	8
P4CGX150DF3117	1.2V	149760	508	8	8
	tif				>
Migration compatibi	lity Compa	anion devio	e		
and elegate exceptions		1			
Migration Devic	es HardC	CODAL!			

Figure 4-6 Device and Options

Choose unused pins. Reserve all unused pins: Choose the As input tri-stated option. See Figure 4-7.



🐇 Device and Pin Options	- my_first_fpga
Category:	
General	Unused Pins
Programming Files Unused Pins Dual-Purpose Pins Capacitive Loading Board Trace Model I/O Timing Voltage Pin Placement Error Detection CRC CvP Settings	Specify device-wide options for reserving all unused pins on the device. To reserve individual dual-purpose configuration pins, go to the Dual-Purpose Pins tab. To reserve other pins individually, use the Assignment Editor.         Reserve all unused pins:       As input tri-stated         Description:       Description:
	Reserves all unused pins on the target device in one of 5 states: as inputs that are tri-stated, as outputs that drive ground, as outputs that drive an unspecified signal, as input tri-stated with bushold, or as input tri-stated with weak pull-up.
	OK Cancel Help

Figure 4-7 Setting unused pins

Click twice OK.

4. In the Processing menu, choose Start Compilation. After the compile, Choose Tools > Programmer. Select the my\_first\_fpga.sof file from the project directory. Click Start. At this time you could find the other LEDs are unlighted.



## **Chapter 5**

## DE2i-150 System Builder

#### 5.1 Headquarter & Branches

- Tel : +886-3-5508800
- Fax : +886-3-5502860
- Add : No.356, Sec. 1, Fusing E. Rd., JhuBei City , HsinChu County Taiwan 302-64
- Email : sales@terasic.com / support@terasic.com

