Rensselaer Polytechnic Institute Computer Hardware Design – ECSE 4770

Quartus Tutorial

1. Introduction:

This tutorial session has the goal of familiarizing you with the tools necessary to complete the labs based on FPGA boards. First we will cover the basics of the software. Then you will enter a simple design the part will be simulated. Finally, there will be a brief description of downloading the design to the FPGA.

2. Quartus program basics:

2.1 Download: The Quartus tool is provided by Altera for their FPGA boards. It is available for download from the following website.

https://www.altera.com/support/software/download/altera design/quartus we/dnl-quartus we.jsp

Current version is V9.0 service pack 2 (1.31GB). Download this and install on your laptop. This version doesn't require a license. The lab workstations have a fall back version of 7.2 which is known to work in the previous academic year. In case you have problems you can get this version from the archives section of the same site.

https://www.altera.com/support/software/download/altera design/guartus archive/dnl-guartus archive.jsp

2.2 Project creation:

Start the Quartus software from your program files.



All programs→ Altera → Quartusll 9.0 web edition→Quartusll 9.0sp2 web edition (blue logo The quartus program starts up with the following screen with the options to create a new project or open an existing project.

Start Designing	Start Learning
Designing with Quartus II software requires a project	The audio/video interactive tutorial teaches you the basic features of Quartus II software
Create a New Project (New Project Wizard)	Open Interactive Tutorial
Open Existing Project	
Open Recent Project: class_tutorial proc_2pipe	

Alternatively, you could open a new project or existing project by going to the file menu.

3	Quartus II	[
File	e Edit	View	Project	Assignme	ents	Processing	Tools	Window	Help	
D	<u>N</u> ew			Ctrl+N	b	CH			- 💥	10000
2	<u>O</u> pen			Ctrl+O	E		- ×			
	<u>C</u> lose			Ctrl+F4	E					
12	New Pro	ject W	izard							
R	Open Pro	oject		Ctrl+J				and the second		
	Convert	MAX+	PLUS II Pr	oject				Purch Softwa	iase Quartus® II are Get a Nios® II	
	Save Proj	jec <u>t</u>						Pri	ocessor Free!	
	Clos <u>e</u> Pro	oject			L			Lim	ited Time Offer	▶Important Arria
	<u>S</u> ave			Ctrl+S					Buy Now	Family Changes
	Save <u>A</u> s									
	Save Cur	rent R	eport Sect	ion As						
	<u>F</u> ile Prop	erties.								
	Create /	Updat	e	•	H					
	Export				H					
	Convert	Progra	a <u>m</u> ming Fi	iles	F		- × -			
	Page Set	<u>u</u> p			H	Т	me û			
B.	Print Pre	view			E					
6	<u>P</u> rint			Ctrl+P	E					
	Recent F	iles		,	ran	nming files)				
	Recent P	rojects	s	,	╟					
	Exit			Alt+F4	mm	ner)				
-					_					

Create a new project using one of the above options. You get the following screen to set the project directory, project name and target FPGA device. Click next.

New Project Wizard: Introduction
The New Project Wizard helps you create a new project and preliminary project settings, including the following:
 Project name and directory Name of the top-level design entity Project files and libraries Target device family and device EDA tool settings
You can change the settings for an existing project and specify additional project-wide settings with the Settings command (Assignments menu). You can use the various pages of the Settings dialog box to add functionality to the project.
☐ Don't show me this introduction again
< Back Next > Finish Cancel

/hat is the working directory for this project? :/Altera\CHD //hat is the name of this project? Class_Tutoria //hat is the name of the top-level design entity for this project? This name is case sensitive and must xactly match the entity name in the design file. Class_Tutoria Use Existing Project Settings	Project Wizard: Directory, Name,	Top-Level En	tity [page 1 o	f 5]		X
Wiltera\CHD //at is the name of this project? Class_Tutoria //at is the name of the top-level design entity for this project? This name is case sensitive and must xactly match the entity name in the design file. Class_Tutorial //ass_Tutorial //ass_Tutorial //ass_Tutorial	What is the working directory for this p	oject?				
/hat is the name of this project? Class_Tutoria /hat is the name of the top-level design entity for this project? This name is case sensitive and must xactly match the entity name in the design file. Class_Tutorial Use Existing Project Settings	D:\Altera\CHD					
Class_Tutorial /hat is the name of the top-level design entity for this project? This name is case sensitive and must xactly match the entity name in the design file. Class_Tutorial Use Existing Project Settings	What is the name of this project?					
/hat is the name of the top-level design entity for this project? This name is case sensitive and must wactly match the entity name in the design file. Class_Tutorial	Class_Tutoria					
Class_Tutorial	' What is the name of the top-level desig exactly match the entity name in the de	n entity for this	s project? This	name is case s	ensitive and must	_
Use Existing Project Settings	Class_Tutorial	-				
Use Existing Project Settings						
	Use Existing Project Settings					
						_
< Back Next > Finish Cancel		< Back	Next >	Finish	Cancel	

In the above screen, the working directory was chosen by user preference. The default location would have been in the installation directory. Each version upgrades of Quartus creates a separate installation

folder. Hence, it is advisable to have a separate folder for quartus projects. This choice is left to the user. Enter the project name and click Next to get the next screen. Skip the next screen on adding files and proceed to the screen on setting family and device settings.

Device faililiy				Show in 'Av	/ailable de	evice' list	
Family: Cyclone II			-	Package:	Any	,	-
Devices: All			-	Pin count:	Any	,	•
- Target device				Speed grad	ie: Any	,	-
C Auto device selecti	ed by the Fitter			🔽 Show a	idvanced	devices	
Specific device sel	ected in 'Availa	able devices	'list	🗖 HardCo	ipy compa	tible only	ŗ
Name	Core v	LEs	User I/.	Memor	Embed.	PLL	-
	1 2) (10752	User 17. 215	Memor	Embea.	PLL	- 1
EF2C20F404C0	1.27	10752	315	233616	52 52	4	
EP2C200240C8	1.2V	18752	142	239616	52	4	
EP2C35F484C6	1.2V	33216	322	483840	70	4	
EP2C35F484C7	1.2V	33216	322	483840	70	4	
EP2C35F484C8	1.2V	33216	322	483840	70	4	
EP2C35F484I8	1.2V	33216	322	483840	70	4	
EP2C35F672C6	1.2V	33216	475	483840	70	4	
	1 707	5551E III	475	nkocok	70	4	P.
- Companion device							
HardCopy							-
rididoopy.							

Altera has a long list of FPGA chips including Flex10k, Max7000, Stratix, Cyclone (I, II, III).

Our lab supports two boards for the course namely UP2 and DE2. UP2 board which has been used in the previous academic years has the Flex10k and Max7000 series chips. The DE2 board is the newer version and has the Cyclone II chip on it. We also have boards for Cyclone III. Depending on the family of FPGA, the number of logic components that can be programmed into the chip varies. For the tutorial, **choose Cyclone II** in the Family drop box. In the Available devices box, choose **EP2C35F672C6**. You can identify the right chip and number by looking at the Altera board's FPGA chip. If a different board is chosen, choose accordingly the chip used on that board.

Skip the next screen on EDA tools. This is provided if you wish to use external EDA tools from synopsis or cadence. Altera/ Quartus provides a built in tool for compiling Verilog, VHDL, AHDL and a timing analyser. This is sufficient for our course requirements.

Click finish on the last screen.

2.3 Design Entry:

From the file menu, choose New file and you would get the following screen.



There are several file types listed in the above screen. Block Diagram/ Schematic file entry is chosen for graphical entry of the schematic and is the preferred form of logic diagram entry. Verilog and VHDL file is also supported to input the circuit description in those languages. We also have a file for Memory initialization file. This file is used to store a certain pattern of hexa-decimal digits. This is useful in the advanced computer hardware design lab. The other file of interest is the Vector Waveform File. This file is used to store and to observe the output after the simulation.

Choose Block Diagram/Schematic entry to proceed with the project design. The default name would be Block#.bdf. Save this file to the file name of your choice. The first file of the project is generally saved to the default project name. Set this particular file as your top level entity in your design as shown in the following screen – **Project** \rightarrow **Set As Top level Entity**.

😲 Quartus II - D:/Altera	a/CHD/Class_Tutorial - Class_Tutorial - [Class_Tutorial.bo	df]
📲 File Edit View	Project Assignments Processing Tools Window	/ 1
🛛 D 🚅 🗟 🗿 🎒	Add Current File to Project	
Project Navigator	Add/Remove <u>F</u> iles in Project	Futo
Entity	<u>R</u> evisions	
Cyclone II: EP2C35	Copy Project	
	Archive <u>P</u> roject	
	Restore Archived Project	
	Import Database	
	Export Database	::
	Import Design Partition	
	Export <u>D</u> esign Partition	11
	Generate <u>B</u> ottom-Up Design Partition Scripts	
	Generate Tcl File for Project	::
A Hierarchy Files	Generate PowerPlay Early Power Estimator File	
Tasks Flow: Compilation	Organi <u>z</u> e Quartus II Settings File	
Task 🖌	HardCopy <u>U</u> tilities	
E Compile D	Locate	:::
Fitter	Set as Top-Level Entity Ctrl+Shift+J	
Elassi	Hierarchy	

If you now go to the **Project** \rightarrow **Add/Remove Files**, you will encounter a screen that we had encountered during project set up. And you will notice the current file listed in the box.

In the schematic window, you will notice a graphical menu on the left. Hovering your mouse over these options will list Selection tool, Text tool, Symbol tool, Block tool, Orthogonal node tool, Orthogonal bus tool and some options to rotate/ flip your symbols. To start, **double click in the schematic window** to get the following screen listing the various libraries/ symbols listed. Expand the selection to see various symbol lists.

Symbol



Altera provides prebuilt library entities for mega functions (custom built to your preferences), maxplus libraries that include the 74 series chips commonly used in the lab protoboards, as well as some primitives. The name box allows you to search or a particular symbol (such as input, and2, output, nor2 etc.). The Mega wizard plug-in manager gives you more options to build custom designs. This will be shown later.

Select the logic symbol you wish to enter and press OK to insert it into the schematic window. Place them anywhere in the schematic. They can be moved around by clicking on them and dragging to the new position. For the tutorial, we will have a simple logic with NAND, AND and an OR gate.

Choose primitives \rightarrow Logic \rightarrow nand2; similarly for nor2 and and2.

Insert the symbols for Input and output by following the same procedure. These symbols are also listed in the primitives section (or enter input or output in the name box).

You may also enter the actual chip (7400 for nand and 7402 for nor) from the maxplus library which is available in the "others" \rightarrow maxplus2 \rightarrow 7400; similarly for 7402. Do check datasheets available on websites such as alldatasheet.com for these chip numbers and pin numbers for more details.

If you wish to delete any incorrect entry, select the symbol and press delete on your keyboard.

Connect the blocks using the orthogonal node tool (thin wire) available on the left menu. This allows drawing vertical/horizontal lines. You will need to click and hold as you draw. Ensure that there are no gaps to ensure proper connection between the two terminals. Moving one of the symbols should automatically stretch the connected wire if there are no breaks in the wires.

Rename the input and output pins by double clicking them. You will get the following pop up screen.

Pin Properties		x
General Format To create mult ''name[30]''),	ple pins, enter a name in AHDL bus notation (for example, or enter a comma-separated list of names.	
Pin name(s):	pin_name	
Default value:	VCC	
	ОК	Cancel

Enter the corresponding name for the pin name.

Create the following schematic.



The above logic can be reused as a new design entity in a different schematic. To do this, you would need to create a symbol file for the given schematic.

Choose File → **Create / Update** → **Create Symbol File** for current file. You can further edit the symbol file by opening it and making changes to the symbol file.

A new file is created with the default name as the same existing file but the extension is .bsf (block symbol file). Alternatively, you can manually create a new symbol file from the **File new file Other Files Block symbol file** and Enter the input and output pins. This is not recommended for the class as you may miss pins that could cause compilation errors later.

To use this schematic in a different schematic, open a new block schematic file.

File \rightarrow New \rightarrow Block Schematic file

Since we are instantiating the other schematic symbol in this new file, we will need to set the new file as the top level entity after it has been saved to a file name. With the new file window selected, choose **Project**->Set as top-level entity.

In the new file, double click to insert a symbol. In the menu, under Project, you will see the previously created symbol file listed. Select this and insert it in the schematic.

Symbol	×
Libraries:	Class_Tutorial A Z B C D
Class_Tutorial	inst
Repeat-insert mode Insert symbol as block Launch MegaWizard Plug-In	
MegaWizard Plug-In Manager	
OK Cancel	[]

You may set a new set of input /output pins in the new schematic.

•	•	•	•	•	•	•	•	•	•	• •	• •	• •	• •	•		• •	•	•	•	·	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	·	•	•	•	•	•	•	•	•	•		• •		•	·	•	·	·	•	•	•	•	• •	• •	•	•	•	•	• •	•
•		•	•	•	•	•	•	•	•	• •	• •	• •	• •	•		• •	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	·	·	•	·	•	•	•	•	•	•	•	•	•	•				•	•	•	•	•	•	•	•	•	• •		•	•	•	•	• •	•
						•	•	•	•	• •	• •		• •	•			•			·	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•								·	·		•	•	•	• •					•	• •	•
							•	•	•	• •				•								•	•	•	•	er.													•	•	•	•	•	•														• •						• •	•
									•														•	•	•	1	CI	as	ss	1	TI	ut	or	ria				£.,																											•
																														-	_	_	_			•		£.,																											•
																												L								L		÷.,																											
		. 1	11	<u>ا</u>	Α.								Г		_	∍	_	-Þ	P	Л		-				-		١.	Λ.						7	⊢		-						0	U	Р	UT				_	_	2	- Ç	JU	Л	_2								
		. 5	~~~)		×		ŝ.				1		11	~					. 1	-	L		£.,						ļ	••••			···•							·		·	,						ι.	
		. 1	11	<u>ا_</u> ۱	в								Г			\neg	_	-P	P	Л		-				-		4.1	R							L		£.,																											
		. 5)		×		ŝ.				1		Ľ	υ.							L		Ļ.,																											
			11		С								Г			\neg	_	1	P	Л						-		4,	0							L		Ļ.,					• .																						
		- 5)		÷.,		ŝ.				1		Ľ	6							L		Ļ.,					.																						
			11	۱.	D								Г	_	_		_	1	P	Л						-		4.	n.							L		Ļ.,																											
																		,N	19	÷		λ.				1		Ľ	υ.							L		Ļ.,																											
																												L								L		ι.																											
																												L								L		ι.																											
																												L	_		_	_	_			1		ι.																											
																											in	st	ŧ.									ι.																											
																										I												λ.																											

We have completed schematic entry and symbol creation.

3. Compilation and Simulation:

To compile the project, click on the **Processing** \rightarrow **Start Compilation**. This would carry out a series of steps from Analysis and Synthesis, Fitter (Place & Route), Assembler and Timing Analysis. Alternatively, you could do this one step at a time by going to Processing \rightarrow start \rightarrow and choose each step in sequence one at a time.

The window below shows the compilation messages. If there are any messages in red, they need to be fixed.

To carry out simulations, we need to setup the simulation input and output vectors. Create a new file (Vector Waveform File).

You should be getting a blank waveform file as shown in the following screen capture.

📸 Tuti	orial_to	p.bdf			🕀	Compilation Report	- Flow Summary		🛛 🗹 🖓 Wavel	iorm1.vwf	
	Mast	er Time Bar:	16.2	5 ns	Pointer:	1.55 ns	Interval:	-14.7 ns	Start:	End:	
A			Value at	0 ps		10.0 ns		20.0 ns		30.0 ns	40.0 ns
₩ 🔍		Name	16.25 ns				16.25 ns				
86 2.5.											
VII ×											
0 <u>1</u>											
Z XW											
XI XH											
$\sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{i=1}^{n} \sum_{i$											
$\forall \overline{c} \ \forall \overline{\diamond}$											
$\underline{\underline{\times 5}} \times \underline{B}$											
₽ Ż↓											
	1										
			+	<							•

The left partition (Name) would be where the nodes are added. The right partition is where the input and output waveforms are observed. To select the input and output pins that are being observed for simulations, **right click in the Name pane** \rightarrow **Insert** \rightarrow **Node or bus.** This should give you the following screen

Insert Node o	r Bus			×
Name:	I			ОК
Туре:	INPUT	•]	Cancel
Value type:	9-Level	-]	Node Finder
Radix:	ASCII	-]	
Bus width:	1			
Start index:	0			
🔲 Display g	ray code count as binary count			

Click on Node finder to get to the next screen shot

Named: Filter: Look in: Look in: Interview Nodes Found: Selected Nodes: Name Assignments >>> >>> < Interview Value Value <th>Node Finder</th> <th></th> <th></th> <th></th> <th>×</th>	Node Finder				×
Look in: Iutoria_top Cancel Nodes Found: Selected Nodes: Name Assignments	Named:	▼ Filter: Pins: all	Customize	List	OK
Nodes Found: Name Assignments >>>> >>>> Vodes Found: Selected Nodes: Name Assignments >>>> Vodes Found: Selected Nodes: Name Assignments Vodes Found: <td>Look in: Tutorial_top</td> <td>ĸ</td> <td>💌 🔽 Include subentities</td> <td>Stop</td> <td>Cancel</td>	Look in: Tutorial_top	ĸ	💌 🔽 Include subentities	Stop	Cancel
Name Assignments >	Nodes Found:		Selected Nodes:		
	Name	Assignments -	Name	Assignments	
		>			
		>>			
		Þ	•	۴	

The "Look in" box identifies the schematic file from which the node is listed. Click List. The left pane identifies the nodes found. Select the required nodes and send them to the right pane by clicking the right arrow. Click ok to return to the insert node/bus window. Click ok to return to the vector waveform screen.

指 Tuto	rial_top.l	bdf			1	Compilation Rep	ort - Flow Summary		🖸 W	aveform1.vwf*			
	Master	Time Bar:	16.2	!5 ns	Image: Pointer:	200 ps	Interval:	-16.05 ns	Start:	80.0 ns	End:	90.0 ns	
∖ A ₩ Q		Name	Value at 16.25 ns	0 ps	20.0 ns 16.25 ns	40.0 ns	60.0 ns	80.0 ns	100,0 ns	120 _, 0 ns	140 _, 0 ns	160 _, 0 ns	
	₽0	IN_A	A 0										_
86 0 C		IN_B	A O						,				_
	DP 2	IN_C	A0										+
^w ô	@4	T_Z	AX						*****			****	***
VI VH													
X2 XB													
<u>™</u> , 21													
			Þ	•									•

The waveform pattern for the input initially set to default zero. You can change the pattern to the desired value of high or low by drawing a rectangle for each segment you wish to change with the mouse and choosing the high or low option.

Save the vector waveform file.

Click simulate icon shown in the figure below to get the output results of the simulation

							(\frown							
💥 🖉 🏈	V	8	•		\$	M 👌	∍ ©(. (۵ 🗳	2					
		1.2	s		-		. \		<u>.</u>	• • •					
📸 Tutorial_top.bdf			🕘	Compilation	Report	Flow Summary	🛛 🖸 Class	_Tutorial.vwf		i 🄄 🥹 Simulatio	on Report - Sim	ulatio [指 Class_Tutorial	.bdf	
Simulation Re	Simu	ulation \	Waveforms												
🛛 🚑 🖹 Legal Not	Simu	ulation m	ode: Timing												~
Flow Sum															
Flow Setti															Ŧ
		Master	Time Bar:	16.2	5 ns	 Point 	er: 2	5.49 ns	Interval:	9.24 ns	Start:		End:		
Setting Simula Simula	A ⊛		Name	Value at 16.25 ns	0 ps	20.0 ns 16.25 ns	40.0 ns	60.0 ns	80.0 ns	100,0 ns	120,0 ns	140,0 ns	160 _, 0 ns	180 _, 0 ns	200.0 ns
- 🚑 🎹 INI Usi	±,	₽0	IN_A	A 0											
- 🚑 🏵 Messa	Đ	⊡ •1	IN_B	A 0											
	ÅÅ.	⊡ >2	IN_C	A 0											
	ara.	₽ 3	IN_D	A 1											
	•	- 1 ● 4	T_Z	A 0											
	-														
	鄂														
	₽↓														

Take a moment to validate the results for each set of inputs. The output should correspond to the logic implemented in the schematic.

Save your project (File → Save Project)

Quartus provides a means to archive the files into a small quartus archival which is convenient for transfer to a flashdrive or uploading to your email. You can restore the files from this archival on a different machine or simply keep these archives as a backup for your designs. To do this, first check whether all the required files are present in the project (**Project** \rightarrow **Add/Remove files** in Project).

To archive: click **Project** Archive Project resulting in the following pop up window.

Archive Project			×
Archive file name:	Class_Tutorial		
Advanced		Archive	Cancel

All the relevant files for the project will be automatically included and archived. You can locate the archived file as your project name.qar (Quartus II Archive File) in the directory where the project was initially created. To restore, open Quartus and click Project \rightarrow Restore Archived file. Browse to locate the .qar file and follow the onscreen instructions regarding the directory where the project needs to be restored.

4. Transferring design to FPGA board

4.1 Initial setup for DE2 board:

The board details can be found on the DE2 user manual found in the following link.

ftp://ftp.altera.com/up/pub/Webdocs/DE2_UserManual.pdf



Before transferring the compiled binary to the FPGA board, you need to assign the pin number/ names to the input /output pins in the schematic. Go to Assignments – pins to get the following screen shot.



Do the pin assignments for the given input / output pins to the corresponding IObanks on the FPGA chip. You have the freedom to choose any IOBank and pin number. Later, we will show how to assign the pins to a particular switch or led.

Now that the assignments have been done, compile the project again. Click **Tools** \rightarrow **Programmer** to open a new window which would allow us to transfer the binary to the FPGA.

Ensure that your computer is connected to the FPGA board. In the case of DE2 boards (Cyclone II), you can connect directly through the USB port on your computer. For the older boards with MAX/FLEX chips, you would need to connect through the printer port. Installation of a byte blaster program is also required for the printer port method. These have been set up on the lab computers for the older boards.

U Quartus II - D:/Altera/CHD/Class_Tutorial - Class_Tutorial.cdf]													
File Edit Proces	sing Tools Window												
🔔 Hardware Setup	No Hardware						М	lode: JTAG			 Progre 	:\$\$:	0%
Enable real-time I	SP to allow background prog	gramming (for MAX II devi	ices)										
Mu Start	File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank• Check	Examine	Security Bit	Erase	ISP CLAMP		
🖬 Stop	Class_Tutorial.sof	EP2C35F672	002F82F3	FFFFFFF	2								
Auto Detect													
🗙 Delete													
Add File													
🞬 Change File													
Save File													
🗳 Add Device													
🜓 Up													
🔑 Down													
For Help, press F1													NUM

When the hardware is connected, the start button is enabled. Click it to transfer the sof file to the FPGA. Your logic circuit is now programmed into the FPGA. Connecting external wires to the corresponding pins on the IOBanks to the breadboards, one can simulate the logic with external signal waveforms.

4.2 Altera USB blaster driver:

This should be already come with Quartus installation. Check the drivers directory in the quartus installation folder. C:\altera\90sp2\quartus\drivers

- 1. Plug the USB-Blaster download cable into the PC. The **Found New Hardware** dialog box appears.
- 2. Select Locate and install driver software (recommended).
- 3. Select Don't search online.
- 4. When you are prompted to **Insert the disc that came with your USB-Blaster**, select **I don't** have the disc. Show me other options.
- 5. Select Browse my computer for driver software (advanced) when you see the Windows couldn't find driver software for your device.
- 6. Click **Browse...** and browse to the *<Path to Quartus II installation>\drivers\usb-blaster* directory. Click **OK**. {C:\altera\90sp2\quartus\drivers\x32}
- 7. Select the **Include subfolders** option and click **Next**.
- 8. If you are prompted Windows can't verify the publisher of this driver software, select Install this driver software anyway in the Window Security dialog box.
- 9. The installation begins.
- 10. When the software for this device has been successfully installed appears, click Close.

In the previous programmer window, you may need to set up the Altera hardware setup (left top corner). Change the currently selected hardware to USB-Blaster and close the window.

dware Setup ardware Settings JTAG Settings Select a programming hardware se hardware setup applies only to the) etup to use whe current progra	en programming d mmer window.	evices. This programming
Currently selected hardware: Available hardware items:	USB-Blaster (U	JSB-0]	.
Hardware	Server	Port	Add Hardware
USB-Blaster	Local	USB-0	Remove Hardware
			Close

4.3 Quartus II Version 4.2 and later

- 1. Start the Quartus® II software.
- 2. Choose Programmer from the Tools menu. The Programmer window will open.
- 3. Click the Hardware Setup... button to open the Hardware Setup window.
 - a. The selected programming hardware is identified as Currently Selected Hardware.
 - b. Programming hardware that is already set up appears in the Available hardware items window.
- 4. Click the Add Hardware button to open the Add Hardware window if the programming hardware you would like to use is not listed in the Available hardware items window.
 - a. Select the appropriate programming cable or programming hardware from the Hardware Type list.
 - b. Select the appropriate port and baud rate if necessary.
 - c. Click OK.
- 5. Select the programming hardware you would like to use by choosing it in the Available hardware items list.
- 6. Click Close.
- 7. Your programming hardware has been set up.

4.4 Configuration to use Switches and Leds:

Rename the input/output pins to SW# (#: 0 to 17), LEDG# (0 to 7) for green LEDs and LEDR#(0 to 17) for red LEDs.

In the pin assignment window (Assignments \rightarrow Pins)

Rename the nodes and pin numbers to the following entries

		Node Name	Direction	Location	I/O Bank	VREF Group	I/O Standard	Reserved	Group	
1	٢	LEDG4	Output	PIN_AE23	7	B7_N0	3.3-V LVTTL (default)			
2		SW0	Input	PIN_N25	5	B5_N1	3.3-V LVTTL (default)			
3		SW1	Input	PIN_N26	5	B5_N1	3.3-V LVTTL (default)			
4		SW2	Input	PIN_P25	6	B6_N0	3.3-V LVTTL (default)			
5		SW3	Input	PIN_AE14	7	B7_N1	3.3-V LVTTL (default)			-
						·	1			-

The values for Switch location and Led location can be found from the DE2 manual- Table 4.1 (Page 28) and Table 4.3 (Page 4.3).

Compile again and download program to test your setup. Do remember to rename the pin names to the corresponding LED/SW names in the schematic as well as simulation window too.

			•	• •					·	·	•	•				•				·	·		•	• •	•			·	·	·	·	·	·	·				•	·		•	 •	·	·	·	•			·	·			·	·	•	•	·			•	• •	·	·					·		
:	:	:					:	:	:	:	:	:	:	:	:				:	:	:	:					:	:	:	:	ſ	CI	a	ss	2	Тι	ito	ri	al				:	:	:				:	:	:		:	:			:	:	:			:	:	:	:			:		:
:	:	:				:	:	ģ				·	·	·	·							·					÷	:	:	:			Γ							٦			1	:	:													÷						÷	:		:	:	: :	:
:		·	•		•	•		4	5	/w	, 											<u> </u>		V.	č		t				t	_	t	A					Ζ	F						-		<u></u>			-[Ż.		EU	G	4						L	·	• •	•	·	• •	
·	L	PI	N_	N	2	5	ŀ		S	N	1								С			\succ	_	VE	Y	<u> </u>	1				ł		ł	в									÷	÷	·		• •	÷	·	÷	÷	• •	÷	÷		•	·	·	·			÷	÷	L	PI	N_	A	Ε2	3	ŀ
		PI	N_	N	2	6	ŀ	Τ	S	N.	2								С			\succ	-	VE	Y	Γ.	-	-	-	·	+		ł	с										÷						÷	÷				•					•				÷						
:	Γ	PI	N_	Ρ	25	5	ŀ		S	N	3	·····	·····	·····		·····	·····					\succ	1	¥.	ų	T.	7		·	·	+			D									:	:	:			:	:	:	:		:	:			:	:	:			:	:	:	:		:	:		:
:		ΡI	N	А	Æ	14	1	1																			1	:	:	:													:	:	:		: :		:	:	:	: :	:	:			:	:	:			:	:	:	:	: :	:	:	: :	:
:	5							1	:	:	:	:	:	:	:	:	: :	:	:	:	:	:		: :		:	:	:	:	:			L	_									1	:	:	:	: :	1	:	:	:	: :	:	:	:		:	:	:	:	: :	:	:	:	:	: :	:	:	: :	:
:	÷	:				:	:	:	÷	÷	:	:	:	:	:	:		:	:	:	:	:				:	:	:	:	÷	L	in	151	t 								:	÷	÷	÷	:	: :	:	÷	÷	:	: :	÷	:	:	:	÷	:	:	:	: :	÷	÷	÷	:	: :	:	÷	: :	:
	•	•	•						•	•	•	•	•	•	•	•				•	•	•	•					•	•	•	•	•	•	•	•	•			•		•		•	•	•	•			•	·	•	• •		•	•		•	•	•	•		•	•	•	•	• •		•	• •	
:	:	:				:	:	:	:	:	:	:	:	:	:				:	:	:	:				:	:	:	:	÷	:	:	:	:	:	:	: :		:	:			:	:	:		: :		:	:	:		:	:			:	:	:			1	:	:	:	: :	:	:	: :	:

Other pins frequently used are the 7 segment displays (Table 4.4) and the expansion headers (Table 4.7).

5. Additional details regarding using Mega functions:

When handling larger designs, this feature of quartus will be quite useful. It provides customized library of bigger functional blocks such as adder, Flipflop array, memory etc. Open the symbol insert window and choose Mega Wizard Plugin manager as shown in the following screen.

Symbol		x
Libraries:	· · · · · · · · · · · · · · · · · · ·	
E C Project		
C:/altera/Josp2/quartus/libra		· · ·
megarunctions		
🖽 🛄 gates		
E 🕂 🖽 10		
🗄 🖾 storage 📃 🗏		• • •
🗄 🗀 others		. : :
🗄 🗁 primitives		
🕀 🙆 buffer		
		•••
• III P		
Name:		· · ·
		• • •
		. : :
Repeat-insert mode		
Insert sumbol as block		
Launch MegaWizard Plug-In		· • •
MegaWizard Plug-In Manager	↓	
	N	
	[
OK Cancel		•••
	L	<u>''''</u> '

You get a new pop up window where you can create a new custom design or edit an existing design. Choose New and click Next.

MegaWizard P	lug-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 Copy an existing custom megafunction variation Copyright (C) 1991-2009 Altera Corporation												
	Cancel < Back Next > Finish												

You now get a new screen which has many library functions in the left pane. Choose the type of function you wish to build. For the tutorial example, we are showing how to build a Read Only Memory custom block. In the right pane, you have the option to generate a verilog or vhdl or ahdl file.

Which megafunction would you like to customize? Select a megafunction from the list below	Which device family will you be Cyclone II
	Which type of output file do you want to create? AHDL Verilog HDL Verilog HDL What name do you want for the output file? Browse D:\Altera\CHD\ROM_test Return to this page for another create operation Note: To compile a project successfully in the Quartus II software, your design files must be in the project directory, in the global user libraries specified in the Options dialog box (Tools menu), or a user library specified in the User Libraries page of the Settings dialog box (Assignments menu).
EUM: 1-FUR I ROM: 2-PORT Shift register (RAM-based) ⊕- Storage ⊡- IP MegaStore	Your current user library directories are: Project User Libraries: working_model\mem\ memmodel\ c:\altera\projs\proc_id_dp\memmodel\
, , , , , , , , , , , , , , , , , , , ,	, Cancel < Back Next > Finisi

It would be advisable to stay consistent with a particular language throughout the project. Since the class uses VHDL in the advanced CHD course, we are choosing VHDL. Enter a file name for the vhdl custom block. The next screen shows you how to customize the mega function block such as number of address bits, clock enabled, bus width, size of memory. Click Next and proceed through the following screens till the end.

MegaWizard Plug-In Manager - ROM: 1-PORT	[page 3 of 7]
ROM: 1-PORT	About Documentation
1 Parameter 2 EDA 3 Summary Settings	
General Regs/Clkens/Adrs Mem Init	
address[40]	Currently selected device family: Cyclone II Match project/default
Clock Block type: AUTO	How wide should the 'q' output bus be? How many 8-bit words of memory? What should the memory block type be? Matching Matching Matchin
	○ M-RAM ○ LCs Options Set the maximum block depth to Auto ✓ words
Resource Usage 1 M4K	What docking method would you like to use?
MegaWizard Plug-In Manager - ROM: 1-PO	DRT [page 4 of 7]
1 Parameter 2 EDA 3 Summary	<u>About</u> <u>D</u> ocumentation
Settings	
ROM_test address[40] clock	Which ports should be registered? □ 'data' input port ☑ 'address' input port ☑ 'g' output port ☑ Create one dock enable signal for each dock signal. All registered ports are controlled by the enable signal (a)
Block type: AUTO	Create a byte enable port
	What is the width of a byte for byte enable? 8 v bits
Resource Usage	Create an 'adr' asynchronous dear for the registered ports
	Cancel < Back Next > Finish

MegaWizard Plug-In Manager - ROM: 1-PORT [pa	age 5 of 7]	and well, its		
ROM: 1-PORT			About	Documentation
Parameter I Parameter Settings I EDA I Summary			Togar	Eccamentation
General > Regs/Clkens/Adrs > Mem Init				
ROM_test address[40] g g g g g g g g g g g g g g g g g g g	Do you want to spect No, leave it bla Initialize me Yes, use this fil (You can use a Initialization File File name: The initial contr to which port's	cify the initial content on nk mory content data to e for the memory content Hexadecimal (Intel-for e [.mif]) memory_init.mif ent file should conform dimensions?	f the memory (XX on powe ent data mat) File [.he	? er-up in simulation ex] or a Memory Browse
	 Allow In-System M independently of t The 'Instance ID' of 	emory Content Editor t he system clock If this ROM is:	NONE	d update content
Resource Usage				
1 M4K		Cancel	< <u>B</u> ack	Next > Finish





You can now insert this new symbol and proceed with your design.

Summary: This tutorial covered all the introductory aspects of the Quartus Altera tool. Students should be in a position to implement new design and execute them on the FPGA boards.