

Quartus II Design Flow



© 2010 Altera Corporation



Can I still use a Processor?

YES!

Three Altera Soft Processor Choices:

- Nios II/f Fast: Optimized for Performance
- Nios II/s Standard: Faster and Smaller than Nios
- Nios II/e Economy: Smallest FPGA Footprint
- Choose peripherals you want
- SOPC Builder software builds interfaces, arbitration etc.



Altera Quartus II CAD Tools



© 2010 Altera Corporation



DE0



© 2010 Altera Corporation



DE0



© 2010 Altera Corporation



DE0 – Switches



Connections between the toggle switches and Cyclone III FPGA



DE0 – LEDs



Connections between the LEDs and Cyclone III FPGA

© 2010 Altera Corporation



DE0 – 7-segment displays

HEX0



Connections between the 7-segment displays and Cyclone III FPGA



© 2010 Altera Corporation



Developing Digital Logic courses with Altera Technology

Tutorial #1



© 2010 Altera Corporation

Outline

- Creating projects in Quartus II
- Targeting a project for a DE0 Board
- Downloading a circuit onto a DE0 board
- Compiling and debugging



Step 1: Start Quartus II



© 2010 Altera Corporation



Step 2: Create a New Project

File	Edit View Project	Assignments	Proces
D	New	Ctrl+N	
È	Open	Ctrl+O	
	Close	Ctrl+F4	
æ	New Project Wizard		
6	Open Project	Ctrl+J	
	Save Project		
	Close Project		
	Save	Ctrl+S	
	Save As		
ø	Save All	Ctrl+Shit	ft+S
	File Properties		
	Create / Update		•
	Export		
	Convert Programming File	s	
D)	Page Setup		
B.	Print Preview		
6	Print	Ctrl+P	
	Recent Files		•
	Recent Projects		•
	Exit	Alt+F4	

- Click File Menu
- Select New Project
 Wizard
- This will open a new window where project information can be specified

© 2010 Altera Corporation



Project Name and Directory

😋 New Project Wizard	x
Directory, Name, Top-Level Entity [page 1 of 5]	
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.	2
light	
Use Existing Project Settings	
< Back Next > Finish Cancel Help	,

The project must have a name, which is usually the same as the top-level design entity that will be included in the project.

© 2010 Altera Corporation



Add Source Files to Project

🔇 New Proje	ct Wiza	rd				— X
Add Fil	es [p	age 2	of 5]			
Select the d project. Note: you c	esign fik an alway	es you wa ys add dei	nt to include in the project. Click a sign files to the project later.	Add All to add all design files	in the project dire	ectory to the
File name:						Add
File Name	Туре	Library	Design Entry/Synthesis Tool	HDL Version	[Add All
					[Remove
					(Up
					(Down
					(Properties
Specify the	path na	mes of an	y non-default libraries. User Libr	aries		
			< Back	Next > Finish	Cancel	Help

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

© 2010 Altera Corporation



Specify FPGA Device

- Select the FPGA device on the board
 - Cyclone III Family
 - For DE0 EP3C16F484C6
 - Cyclone II Family
 - For DE1 EP2C20F484C7
 - For DE2 EP2C35F672C6

Board	Device Name
DE0	Cyclone III EP3C16F484C6
DE0-Nano	Cyclone IVE EP4CE22F17C6
DE1	Cyclone II EP2C20F484C7
DE2	Cyclone II EP2C35F672C6
DE2-70	Cyclone II EP2C70F896C6
DE2-115	Cyclone IVE EP4CE115F29C7

Family: Cydone II Devices: All Target device Auto device selected by the Fitter Specific device selected in 'Available devices' list Other: Other: n/a Name Core Voltage LEs User I/Os Mame Core Voltage LEs User I/Os Mane Core Voltage Les User I/Os Mane	Family: Cyclone II Devices: All Target device Auto device selected by the Fitter Specific device selected in 'Available devices' list Other: n/a Package: Any Wallable devices: Package: Any Wallable devices: Specific device selected in 'Available devices' list Specific devices celected in 'Available devices' list Value filter: Show advanced devices HardCopy compatible Wallable devices: EP2C35F484C6 1.2V 33216 322 483840 70 EP2C35F484C8 1.2V 33216 322 483840 70 EP2C35F484C8 1.2V 33216 322 483840 70 EP2C35F484E8 1.2V 33216 322 483840 70 EP2C35F672C6 1.2V 33216 322 483840 70 EP2C35F672C7 1.2V 33216 475 483840 70 EP2C35F672C7 1.2V 33216 475 483840 70
Pamiry: Cyclone II Package: Any Package: Any Pin count: Any Speed grade: Any Name filter: Speed grade: Speed grade: Speed grade: Speed grade: Speed grade: Ange filter: Speed grade:	Panniy: Cyclone II ▼ Devices: All ▼ Target device Pin count: Any O Auto device selected by the Fitter Speed grade: Any O Specific device selected in 'Available devices' list Valiable devices: Wame filter: Vailable devices: Valiable devices: Valiable devices: HardCopy compatible Valiable devices: Valiable devices: Valiable devices: Embedded multiplice Specify 484C6 1.2V 33216 322 483840 70 Specify 484C8 1.2V 33216 322 483840 70 Specify 5484C8 1.2V 33216 322 483840 70 Specify 5484C8 1.2V 33216 322 483840 70 Specify 672C6 1.2V 33216 322 483840 70 Specify 672C6 1.2V 33216 475 483840 70 Specify 672C7 1.2V 33216 475 483840 70 Specify 672C8 1.2V 33216 475 483840 70 <
Devices: All Target device Auto device selected by the Fitter Specific device selected in 'Available devices' list Other: n/a vailable devices: Name Core Voltage LEs User I/Os Mame Core Voltage LEs User I/Os Mamery Bits Embedded multiplier S Sep2C35F484C6 1.2V 33216 322 483840 70 Sp2C35F484C8 1.2V 33216 322 483840 70 Sp2C35F48418 1.2V 33216 322 483840 70 Sp2C35F672C6 1.2V 33216 322 483840 70 Sp2C35F672C7 1.2V 33216 322 483840 70 Sp2C35F672C8	Devices: All Pin count: Any Target device Specific device selected by the Fitter Specific device selected in 'Available devices' list Specific device selected in 'Available devices' list Specific device selected in 'Available devices' list Name filter: Name filter: • Other: n/a Valiable devices: Valiable devices HardCopy compatible • Other: n/a Valiable devices: Valiable devices HardCopy compatible • Other: n/a Valiable devices: Valiable devices HardCopy compatible • Other: n/a Valiable devices: Valiable devices: HardCopy compatible • Other: n/a Valiable devices: Valiable devices: HardCopy compatible • Other: n/a Valiable devices: Valiable devices: HardCopy compatible • Other: n/a Valiable devices: Valiable devices: HardCopy compatible • Other: n/a Valiable devices: Valiable devices: Valiable devices: • Other: n/a Yaliable devices: Valiable devices: Valiable devices: Valiable devices: • Other: n/a Yaliable devices: Yaliable devices: Yaliable devices: Valiable devices: HardCopy compatible devices:
Target device Speed grade: Any Auto device selected by the Fitter Specific device selected in 'Available devices' list Other: n/a Show advanced devices HardCopy compatible on Name Core Voltage LEs User I/Os Hemory Bits Embedded multiplier 9 P2C35F484C6 1.2V 33216 322 483840 70 P2C35F484C7 1.2V 33216 322 483840 70 P2C35F484C8 1.2V 33216 322 483840 70 P2C35F484C8 1.2V 33216 322 483840 70 P2C35F672C6 1.2V 33216 322 483840 70 P2C35F672C7 1.2V 33216 475 483840 70 P2C35F672C8 1.2V 33216 475 483840 70 P2C35F672C8 1.2V 33216 475 483840 70 P2C35F672C8 1.2V 33216 475 483840 <th>Target device Speed grade: Any • Auto device selected by the Fitter • Specific device selected in 'Available devices' list • Specific device selected in 'Available devices' list • Specific devices selected in 'Available devices' list • Show advanced devices • HardCopy compatible • Other: n/a • Vision devices • HardCopy compatible • HardCopy compatible • P2C35F484C6 • Core Voltage • LEs • User I/Os • Hemory Bits • Embedded multiplie • P2C35F484C7 • L2V • 3216 • 22 • 483840 • 70 • P2C35F494C8 • L2V • 3216 • 22 • 483840 • 70 • P2C35F672C6 • L2V • 3216 • 322 • 483840 • 70 • P2C35F672C7 • L2V • 3216 • 475 • 483840 • 70</th>	Target device Speed grade: Any • Auto device selected by the Fitter • Specific device selected in 'Available devices' list • Specific device selected in 'Available devices' list • Specific devices selected in 'Available devices' list • Show advanced devices • HardCopy compatible • Other: n/a • Vision devices • HardCopy compatible • HardCopy compatible • P2C35F484C6 • Core Voltage • LEs • User I/Os • Hemory Bits • Embedded multiplie • P2C35F484C7 • L2V • 3216 • 22 • 483840 • 70 • P2C35F494C8 • L2V • 3216 • 22 • 483840 • 70 • P2C35F672C6 • L2V • 3216 • 322 • 483840 • 70 • P2C35F672C7 • L2V • 3216 • 475 • 483840 • 70
Auto device Auto device selected by the Fitter Specific device selected in 'Available devices' list Other: n/a Name filter: Show advanced devices HardCopy compatible on vailable devices: Value filter: Image: Core Voltage LEs User I/Os Memory Bits Embedded multiplier 9 P2C3SF484C6 1.2V 33216 322 483840 70 P2C3SF484C7 1.2V 33216 322 483840 70 P2C3SF484C8 1.2V 33216 322 483840 70 P2C3SF484C8 1.2V 33216 322 483840 70 P2C3SF672C6 1.2V 33216 322 483840 70 P2C3SF672C7 1.2V 33216 475 483840 70 P2C3SF672C8 1.2V 33216	Auto device selected by the Fitter Name filter: N
Auto device selected by the Fitter Specific device selected in 'Available devices' list Other: n/a Show advanced devices HardCopy compatible on railable devices: State of the selected in 'Available devices' list Other: n/a Show advanced devices HardCopy compatible on *ailable devices: State of the selected in 'Available devices' list Show advanced devices HardCopy compatible on *ailable devices: State of the selected in 'Available devices' list Show advanced devices HardCopy compatible on *2C3SF484C6 State of the selected in 'Available devices' list State of the selected in 'Available devices' list Show advanced devices Embedded multiplier for the selected in 'Available devices' list State of the selected in 'Available devices' list State of the selected in 'Available devices' list State of the selected in 'Available devices' list * 2023FF484C7 State of the selected in 'Available devices' list State of the selected in 'Available devices' list	Auto device selected by the Fitter Name hitter: Image: Specific device selected in 'Available devices' list Image: Specific device selected in 'Available devices' list Other: n/a Show advanced devices Image: HardCopy compatible valiable devices: Image: HardCopy compatible Image: HardCopy compatible valiable devices: Vser I/Os Memory Bits Embedded multiplie v2C3SF484C6 1.2V 33216 322 483840 70 v2C3SF484C8 1.2V 33216 322 483840 70 v2C3SF48418 1.2V 33216 322 483840 70 v2C3SF672C6 1.2V 33216 475 483840 70 v2C3SF672C7 1.2V 33216 475 483840 70
Image: Specific device selected in 'Available devices' list Image: Specific devices Image: Show advanced devices HardCopy compatible on the selected in 'Available devices' list *allable devices: ************************************	Name Core Voltage LEs User I/Os Memory Bits Embedded multiplie 22C3SF484C6 1.2V 33216 322 483840 70 22C3SF484C7 1.2V 33216 322 483840 70 22C3SF484C8 1.2V 33216 322 483840 70 22C3SF484C8 1.2V 33216 322 483840 70 22C3SF48418 1.2V 33216 322 483840 70 22C3SF672C6 1.2V 33216 322 483840 70 22C3SF672C7 1.2V 33216 475 483840 70 22C3SF672C7 1.2V 33216 475 483840 70 22C3SF672C7 1.2V 33216 475 483840 70 22C3SF672C8 1.2V 33216 475 483840 70 22C3SF672C8 1.2V 33216 475 483840 70
Other: n/a Other: n/a ailable devices: Name Core Voltage LEs User I/Os Memory Bits Embedded multiplier 9 2C35F484C6 1.2V 33216 322 483840 70 2C35F484C7 1.2V 33216 322 483840 70 2C35F484C8 1.2V 33216 322 483840 70 2C35F484C8 1.2V 33216 322 483840 70 2C35F748418 1.2V 33216 322 483840 70 2C35F672C6 1.2V 33216 475 483840 70 2C35F672C7 1.2V 33216 475 483840 70 2C35F672C8 1.2V 33216 475 483840 70 2C35F672C8 1.2V 33216 475 483840 70 mmmode of the total colspan="4">total colspan="4"total colspan="4">total colspan="4"total col	Name Core Voltage LEs User I/Os Memory Bits Embedded multiplie 2C35F484C6 1.2V 33216 322 483840 70 2C35F484C8 1.2V 33216 322 483840 70 2C35F484C8 1.2V 33216 322 483840 70 2C35F672C6 1.2V 33216 475 483840 70 2C35F672C7 1.2V 33216 475 483840 70 2C35F672C8 1.2V 33216 475 483840 70 2C35F672C8 1.2V 33216 475 483840 70
Other: n/a ailable devices: User I/Os Memory Bits Embedded multiplier 9 2C35F484C6 1.2V 33216 322 483840 70 2C35F484C7 1.2V 33216 322 483840 70 2C35F484C8 1.2V 33216 322 483840 70 2C35F484R8 1.2V 33216 322 483840 70 2C35F484R8 1.2V 33216 322 483840 70 2C35F672C6 1.2V 33216 475 483840 70 2C35F672C7 1.2V 33216 475 483840 70 2C35F672C8 1.2V 33216 475 483840 70 2C35F672C8 1.2V 33216 475 483840 70 Immondevice	Other: n/a Subset Vision Core Voltage LEs User I/Os Memory Bits Embedded multiplie 22C35F484C6 1.2V 33216 322 483840 70 22C35F484C7 1.2V 33216 322 483840 70 22C35F484C8 1.2V 33216 322 483840 70 22C35F484B1 1.2V 33216 322 483840 70 22C35F484B1 1.2V 33216 322 483840 70 22C35F672C6 1.2V 33216 475 483840 70 22C35F672C7 1.2V 33216 475 483840 70 22C35F672C8 1.2V 33216 475 483840 70 22C35F672C7 1.2V 33216 475 483840 70
Name Core Voltage LEs User I/Os Memory Bits Embedded multiplier *2C35F484C6 1.2V 33216 322 483840 70 *2C35F484C6 1.2V 33216 322 483840 70 *2C35F484C8 1.2V 33216 322 483840 70 *2C35F484C8 1.2V 33216 322 483840 70 *2C35F484E8 1.2V 33216 322 483840 70 *2C35F782C6 1.2V 33216 475 483840 70 *2C35F672C7 1.2V 33216 475 483840 70 *2C35F672C8 1.2V 33216 475 48384	Name Core Voltage LEs User I/Os Memory Bits Embedded multiplie 22C35F484C6 1.2V 33216 322 483840 70 22C35F484C7 1.2V 33216 322 483840 70 22C35F484C8 1.2V 33216 322 483840 70 22C35F484818 1.2V 33216 322 483840 70 22C35F672C6 1.2V 33216 322 483840 70 22C35F672C7 1.2V 33216 475 483840 70 22C35F672C8 1.2V 33216 475 483840 70 22C35F672C8 1.2V 33216 475 483840 70 22C35F672C8 1.2V 33216 475 483840 70
Name Core Voltage LEs User I/Os Memory Bits Embedded multiplier 22C35F484C6 1.2V 33216 322 483840 70 22C35F484C7 1.2V 33216 322 483840 70 22C35F484C8 1.2V 33216 322 483840 70 22C35F484C8 1.2V 33216 322 483840 70 22C35F48418 1.2V 33216 322 483840 70 22C35F672C6 1.2V 33216 475 483840 70 22C35F672C7 1.2V 33216 475 483840 70 22C35F672C8 1.2V 33216 475 48384	Name Core Voltage LEs User I/Os Memory Bits Embedded multiplie 22C35F484C6 1.2V 33216 322 483840 70 22C35F484C7 1.2V 33216 322 483840 70 22C35F484C8 1.2V 33216 322 483840 70 22C35F484B8 1.2V 33216 322 483840 70 22C35F484B8 1.2V 33216 322 483840 70 22C35F672C6 1.2V 33216 475 483840 70 22C35F672C7 1.2V 33216 475 483840 70 22C35F672C8 1.2V 33216 475 483840 70
Name Core Voltage LEs User I/Os Memory Bits Embedded multiplier 9 22C35F484C6 1.2V 33216 322 483840 70 22C35F484C7 1.2V 33216 322 483840 70 22C35F484C8 1.2V 33216 322 483840 70 22C35F484C8 1.2V 33216 322 483840 70 22C35F48418 1.2V 33216 322 483840 70 22C35F48476 1.2V 33216 475 483840 70 22C35F672C7 1.2V 33216 475 483840 70 22C35F672C7 1.2V 33216 475 483840 70 22C35F672C8 1.2V 33216 475 483840 70 22C35F672C8 1.2V 33216 475 483840 70 22C35F672C8 1.2V 33216 475 483840 70	Name Core Voltage LEs User I/Os Memory Bits Embedded multiplie 22C35F484C6 1.2V 33216 322 483840 70 22C35F484C7 1.2V 33216 322 483840 70 22C35F484C8 1.2V 33216 322 483840 70 22C35F484C8 1.2V 33216 322 483840 70 22C35F484B8 1.2V 33216 322 483840 70 22C35F672C6 1.2V 33216 475 483840 70 22C35F672C7 1.2V 33216 475 483840 70 22C35F672C8 1.2V 33216 475 483840 70 22C35F672C8 1.2V 33216 475 483840 70 22C35F672C8 1.2V 33216 475 483840 70
Name Core Voltage LEs User I/Os Memory Bits Embedded multiplier 9 22C35F484C6 1.2V 33216 322 483840 70 22C35F484C7 1.2V 33216 322 483840 70 22C35F484C8 1.2V 33216 322 483840 70 22C35F48418 1.2V 33216 322 483840 70 22C35F48418 1.2V 33216 475 483840 70 22C35F672C7 1.2V 33216 475 483840 70 22C35F672C8 1.2V 33216 475 483840 70 22C35F672C8 1.2V 33216 475 483840 70 22C35F672C8 1.2V 33216 475 483840 70	Name Core Voltage LEs User I/Os Memory Bits Embedded multiplie 22C35F484C6 1.2V 3216 322 483840 70 22C35F484C7 1.2V 3216 322 483840 70 22C35F484C8 1.2V 3216 322 483840 70 22C35F484C8 1.2V 3216 322 483840 70 22C35F672C6 1.2V 3216 3216 453 70 22C35F672C7 1.2V 3216 475 483840 70 22C35F672C8 1.2V 3216 475 483840 70
22235F484C6 1.2V 33216 322 483840 70 22C35F484C7 1.2V 33216 322 483840 70 22C35F484C8 1.2V 33216 322 483840 70 22C35F48418 1.2V 33216 322 483840 70 22C35F48418 1.2V 33216 322 483840 70 22C35F672C6 1.2V 33216 475 483840 70 22C35F672C7 1.2V 33216 475 483840 70 22C35F672C8 1.2V 33216 475 483840 70	22C35F484C6 1.2V 33216 322 483840 70 22C35F484C7 1.2V 33216 322 483840 70 22C35F484C7 1.2V 33216 322 483840 70 22C35F484C8 1.2V 33216 322 483840 70 22C35F4848 1.2V 33216 322 483840 70 22C35F672C6 1.2V 33216 475 483840 70 22C35F672C7 1.2V 33216 475 483840 70 22C35F672C8 1.2V 33216 475 483840 70 22C35F672C8 1.2V 33216 475 483840 70 22C35F672C8 1.2V 33216 475 483840 70
P2C35F484C7 1.2V 33216 322 483840 70 P2C35F484C8 1.2V 33216 322 483840 70 P2C35F48418 1.2V 33216 322 483840 70 P2C35F672C6 1.2V 33216 475 483840 70 P2C35F672C7 1.2V 33216 475 483840 70 P2C35F672C8 1.2V 33216 475 483840 70	P2C35F484C7 1.2V 33216 322 483840 70 P2C35F484C8 1.2V 33216 322 483840 70 P2C35F484C8 1.2V 33216 322 483840 70 P2C35F48418 1.2V 33216 322 483840 70 P2C35F672C6 1.2V 33216 475 483840 70 P2C35F672C7 1.2V 33216 475 483840 70 P2C35F672C8 1.2V 33216 475 483840 70
P2C35F484C8 1.2V 33216 322 483840 70 P2C35F48418 1.2V 33216 322 483840 70 P2C35F672C6 1.2V 33216 475 483840 70 P2C35F672C7 1.2V 33216 475 483840 70 P2C35F672C8 1.2V 33216 475 483840 70	P2C3SF484C8 1.2V 33216 322 483840 70 P2C3SF48418 1.2V 33216 322 483840 70 P2C3SF672C6 1.2V 33216 475 483840 70 P2C3SF672C7 1.2V 33216 475 483840 70 P2C3SF672C8 1.2V 33216 475 483840 70 P2C3SF672C8 1.2V 33216 475 483840 70
P2C35F48418 1.2V 33216 322 483840 70 P2C35F672C6 1.2V 33216 475 483840 70 P2C35F672C7 1.2V 33216 475 483840 70 P2C35F672C8 1.2V 33216 475 483840 70	P2C3SF484I8 1.2V 33216 322 483840 70 P2C3SF672C6 1.2V 33216 475 483840 70 P2C3SF672C7 1.2V 33216 475 483840 70 P2C3SF672C8 1.2V 33216 475 483840 70 P2C3SF672C8 1.2V 33216 475 483840 70
22C3SF672C6 1.2V 33216 475 483840 70 P2C3SF672C7 1.2V 33216 475 483840 70 P2C3SF672C8 1.2V 100 100 100 100 HardCopy: 100 100	P2C3SF672C6 1.2V 33216 475 483840 70 P2C3SF672C7 1.2V 33216 475 483840 70 P2C3SF672C8 1.2V 33216 475 483840 70 P2C3SF672C8 1.2V 33216 475 483840 70
22C35F672C7 1.2V 33216 475 483840 70 22C35F672C8 1.2V 33216 475 483840 70 Companion device Image: Companion device	P2C35F672C7 1.2V 33216 475 483840 70 P2C35F672C8 1.2V 33216 475 483840 70
2C35F672C8 1.2V 33216 475 483840 70 Companion device HardCopy:	P2C35F672C8 1.2V 33216 475 483840 70
Companion device HardCopy:	
Companion device	III
Companion device HardCopy:	
HardCopy:	Companion device
hardcopy:	Unad Canada
	tardcopy:
Limit DSP & RAM to HardCopy device resources	Limit DSP & RAM to HardCopy device resources
Limit DSP & RAM to HardCopy device resources	Limit DSP & RAM to HardCopy device resources
	LITTLE LATE A DAPPEND FROM AND A DEVICE LEADER FOR

© 2010 Altera Corporation



Additional EDA Tools

- Specify Tools, in addition to Quartus II, that you will use
- These are unnecessary for small student designs
 - Leave all entries as <None>
 - Press Next

DA tools:			
Tool Type	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
Timing Analysis	<none></none>	<none> *</none>	Run this tool automatically after compilation
Formal Verification	<none> •</none>		
Board-Level	Timing	<none></none>	
	Symbol	<none></none>	
	Signal Integrity	<none> -</none>	
	Boundary Scan	<none></none>	

© 2010 Altera Corporation



- A summary of the chosen settings appears in the screen shown in Figure.
- Press Finish, which returns to the main Quartus II window, but with *light* specified as the new project, in the display title bar

© 2010 Altera Corporation



Simple Project



LIBRARY ieee ; USE ieee.std_logic_1164.all ; ENTITY light IS PORT (x1, x2 : IN STD_LOGIC ; f : OUT STD_LOGIC) ; END light ; ARCHITECTURE LogicFunction OF light IS BEGIN f <= (x1 AND NOT x2) OR (NOT x1 AND x2); END L = is Exerction of the second s

END LogicFunction;

© 2010 Altera Corporation



Step 3a: Create Source File

- Select File > New to get the window in Figure, choose VHDL File, and click OK. This opens the Text Editor window.
- Specify a name for the file that will be created and select File > Save to open a pop-up box and in the box labeled Save as type choose VHDL File



© 2010 Altera Corporation



Step 3b: Add Source File

Select: •Assignments > Settings and looks for File item or •Project > Add/Remove Files in Project

_ℓ Settings - light	
Category:	Device
General Files	
Files Libraries Joperating Settings and Conditions Select the design files you want to include in the project. Click Add All to add al design directory to the project.	n files in the project
Voltage Temperature File name:	Add
Compilation Process Settings Early Timing Estimate File Name Type Library Design Entry/Synthesis Tool HDL Version	Add All
Incremental Compilation Physical Synthesis Optimizations 4 EDA Tool Settings Design Entry/Synthesis Simulation Formal Verification Boord Level 4 Analysis & Synthesis Settings VHDL Input Default Parameters Fitter Settings TimeQuest Timing Analyzer Assembler Design Assistant SignalTap II Logic Analyzer Logic Analyzer Interface PowerPlay Power Analyzer Settings SSN Analyzer	Remove Up Down Properties
OK Cancel Ap	Apply Help

© 2010 Altera Corporation



Step 4: Assign Pins to connect switches/lights to inputs and outputs of your circuit

- Click Assignments, then Import Assignments...
- Import file
 - DE0_pin_assignments.qsf
- Imports locations for predefined port names, such as SW, LEDG, KEY, and others
 - Can be done manually for custom port names

Component	DE0
SW ₀	PIN_J6
SW1	PIN_H5
LEDG ₀	PIN_J1

© 2010 Altera Corporation



Step 4: Assign Pins to connect switches/lights to inputs and outputs of your circuit

Pin & Location Assignments

set location assignment PIN B1 -to LEDG[9] set location assignment PIN B2 -to LEDG[8] set location assignment PIN C2 -to LEDG[7] set location assignment PIN C1 -to LEDG[6] set location assignment PIN E1 -to LEDG[5] set location assignment PIN F2 -to LEDG[4] set location assignment PIN H1 -to LEDG[3] set location assignment PIN J3 -to LEDG[2] set location assignment PIN J2 -to LEDG[1] set location assignment PIN J1 -to LEDG[0] set location assignment PIN D2 -to SW[9] set location assignment PIN E4 -to SW[8] set location assignment PIN E3 -to SW[7] set location assignment PIN H7 -to SW[6] set location assignment PIN J7 -to SW[5] set location assignment PIN G5 -to SW[4] set location assignment PIN G4 -to SW[3] set location assignment PIN H6 -to SW[2] set location assignment PIN H5 -to SW[1] set location assignment PIN J6 -to SW[0] set location assignment PIN F1 -to KEY[2] set location assignment PIN G3 -to KEY[1] set location assignment PIN_H2 -to KEY[0]

Pin & Location Assignments

set location assignment PIN E11 -to HEX0[0] set location assignment PIN F11 -to HEX0[1] set location assignment PIN H12 -to HEX0[2] set location assignment PIN H13 -to HEX0[3] set location assignment PIN G12 -to HEX0[4] set location assignment PIN F12 -to HEX0[5] set location assignment PIN F13 -to HEX0[6] set location assignment PIN D13 -to HEX0[7] set location assignment PIN A15 -to HEX1[6] set location assignment PIN E14 -to HEX1[5] set location assignment PIN B14 -to HEX1[4] set location assignment PIN A14 -to HEX1[3] set location assignment PIN C13 -to HEX1[2] set location assignment PIN B13 -to HEX1[1] set location assignment PIN A13 -to HEX1[0] set location assignment PIN B15 -to HEX1[7] # Pin & Location Assignments

_____ set location assignment PIN F14 -to HEX2[6] set location assignment PIN B17 -to HEX2[5] set location assignment PIN A17 -to HEX2[4] set location assignment PIN E15 -to HEX2[3] set location assignment PIN B16 -to HEX2[2] set location assignment PIN A16 -to HEX2[1] set location assignment PIN D15 -to HEX2[0] set location assignment PIN A18 -to HEX2[7] set location assignment PIN G15 -to HEX3[6] set location assignment PIN D19 -to HEX3[5] set location assignment PIN C19 -to HEX3[4] set location assignment PIN B19 -to HEX3[3] set location assignment PIN A19 -to HEX3[2] set location assignment PIN F15 -to HEX3[1] set location assignment PIN B18 -to HEX3[0] set location assignment PIN G16 -to HEX3[7]

SW: IN STD_LOGIC_VECTOR(N DOWNTO 0); LEDG: OUT STD_LOGIC_VECTOR(N DOWNTO 0); HEX0: OUT STD_LOGIC_VECTOR(0 TO N)

x1, x2, f ?????

© 2010 Altera Corporation



Step 5: Compile Design



© 2010 Altera Corporation

Step 6: Examine Compilation Report

Quartus II 64-Bit - D:/introtutorial/light - light								
File Edit View Project Assignments Processing Tools Window Help 🖓 Search altera.com								
<u>□ </u>								
Project Navigator 🖉 🗗 🗙 light.vhd	Compilation Report							
Entity Image: Second secon	Flow Summary Flow Status Successful - Tue May 08 12:07:57 2012 Quartus II 64-Bit Version 12:0 Build 173 05/02/2012 SJ Full Version Revision Name light Top-level Entity Name light Family Cyclone II Device EP2C35F672C6 Tming Models Final Total logic elements 1 / 33,216 (< 1 %) Total combinational functions 1 / 33,216 (< 1 %) Total registers 0 Total pins 3 / 475 (< 1 %) Total virtual pins 0 Total memory bits 0 / 483,840 (0 %)							
Assembler (Generate pr Embedded Multiplier 9-bit elements 0 / 70 (0 %) Total PLLs 0 / 4 (0 %)								
Type Message Info (332102): Design is not fully constrained f Info: Quartus II 64-Bit TimeQuest Timing Analyze Info (293026): Skipped module PowerPlay Power An Info (293000): Quartus II Full Compilation was s System (Processing (89) (Extra Info (Info (81) (Warning (5) (Critical V Location:	Tor hold requirements er was successful. 0 errors, 3 warnings halyzer due to the assignment FLOW_ENABLE_POWER_ANALYZER successful. 0 errors, 8 warnings Warning (3) / Error / Suppressed (6) / Flag / Uccate 100% 00:00:53							



Step 7a: Program the DE0 Board

- **FPGA JTAG mode**: it will retain its configuration as long as the power remains turned on.
- Active Serial (AS) mode: a configuration device that includes some flash memory is used to store the configuration data
- FPGA JTAG mode: flip the RUN/PROG switch into the RUN position and select Tools > Programmer
- Observe that the configuration file light.sof is listed in the programmer window. If the file is not already listed, then click Add File and select it

🕦 Programmer - D:/in	trotutorial/light - light -	[light.cdf]						
File Edit View Pro	ocessing Tools Windo	w Help 🛡				Sea	arch altera.c	om 📀
📩 Hardware Setup	USB-Blaster [USB-0]	Mode:	JTAG	•	Progress:			
Enable real-time ISP	o to allow background prog	ramming (for MAX II and M	IAX V devices)					
📕 Start	File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank- Check	Examine
Stop	light.sof	EP2C35F672	002F836D	FFFFFFF	\checkmark			
Auto Detect								
X Delete								
Add File								
🖾 Change File	4							
Save File								A
Add Device								=
🚹 Up	ADTS							
Down								-
		P						

© 2010 Altera Corporation



Step 7a: Program the DE0 Board



© 2010 Altera Corporation



Step 7b: Program the DE0 Board

- Here it is also 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

lardware Settings JTAG Settings elect a programming hardware set ardware setup applies only to the	ngs tup to use when prog current programmer	ramming device window.	s. This programming
urrently selected hardware: U	SB-Blaster [USB-0]		×
Hardware	Server	Port	Add Hardware
USB-Blaster	Local	USB-0	Remove Hardware
			Close

© 2010 Altera Corporation



Step 7c: Program the DE0 Board

- Click on the Program/Configure check box
- Press Start in the Programmer window; an LED on the board will light up when the configuration data has been downloaded successfully

🔖 Programmer - D:/	introtutorial/light - ligh	t - [light.cdf]					_	
File Edit View P	Processing Tools Win	dow Help 💎				Sea	arch altera.	om 📀
🔔 Hardware Setup.	USB-Blaster [USB-0]	Mode	: JTAG	•	Progress:			
Enable real-time I	SP to allow background pro	ogramming (for MAX II and	MAX V devices)					
Start	File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank- Check	Examine
Stop.	light.sof	EP2C35F672	002F836D	FFFFFFF	₹			
Add Fie Change Fie Save File	4		III					Þ
Add Device								=

© 2010 Altera Corporation



Step 8: See your design work on the board

- Having downloaded the configuration data into the FPGA device, you can now test the implemented circuit.
- Flip the RUN/PROG switch to RUN position.
- Try all four valuations of the input variables x1 and x2, by setting the corresponding states of the switches SW1 and SW0.

© 2010 Altera Corporation



Step 9a: Simulate your design

- Select Start > All Programs > Altera > University Program > Simulation Tools > QSim to open the Qsim tools.
- Select File > Open Project to display a popup window in which you can browse your directories and choose a project file (.qpf file).
- Select the project you wish to simulate and click OK.
- Generate the node finder files by selecting Processing > Generate Node Finder Files
- From QSim, open the Waveform Editor window by selecting File > New Simulation Input File.



© 2010 Altera Corporation



Step 9b: Simulate your design

- 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.
- Select View > Fit in Window



© 2010 Altera Corporation



Step 9c: Simulate your design

- Include the input and output nodes of the circuit to be simulated.
- Click Edit > Insert > Insert Node or Bus: 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.

😲 Insert Noc	🖞 Insert Node or Bus					
Name:	1	ОК				
Type:	INPUT 💌	Cancel				
Value type:	9-Level 🔻					
Radix:	Binary 💌	Node Finder				
Bus width:	1					
Start index:	0					
🗌 Display gr	ray code count as binary count	.d				

Named:	×	Filter: Pir	ns: all	-	ОК
look in:	•			List	Cancel
Nodes Fou	und:		Selected Node	S:	
Name	Туре	>	Name	Туре	
🕑 f 👘	Output		🕪 x1	Input	
🕩 x1	Input		🕪 x2	Input	
🕩 x2	Input	<	🗇 f	Output	
MP XC	npor	<<		ουφοι	

© 2010 Altera Corporation



Step 9d: Simulate your design

- We will now specify the logic values to be used for the input signals x1 and x2 during simulation
- Click on the waveform for the x1 node and using the Edit > Value command, or via the toolbar, 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.



© 2010 Altera Corporation



Step 9e: Simulate your design

- E.g. set x2 to 1 in the time interval 50 to 100 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.
- 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.



© 2010 Altera Corporation



Step 9f: Simulate your design

- To perform the functional simulation, return to the QSim Window and select Assign > Simulation Settings...
- Click the Browse button and select the light.vwf file you created.
- Choose Functional as the simulation type, and click OK.
- Select Processing > Generate Simulation Netlist.
- A simulation run is started by Processing > Start Simulation,

76 Simulation Settings			×
Simulation Settings Specify VWF File			
D:/introtutorial/light.vv/f			Browse
Simulation Type			
Functional	C Timing		
		ОК	Cancel

© 2010 Altera Corporation



Step 9g: Simulate your design

- To perform timing simulation choose Select Assign > Simulation Settings...
- Choose Timing as the simulation type, and click OK.
- Run the simulator
- Observe that there is a delay in producing a change in the signal f from the time when the input signals, x1 and x2, change their values.
- This delay is due to the propagation delays in the logic element and the wires in the FPGA device

🍕 Sir	nulation Wav	eform Editor -	[light.sim.vwf] (i	Read-Only)				_ 0 <mark>_ X</mark>
File	Edit View	Help 💎					Search alter	a. com ; 😯
]€, ‰ -9	▶ 슈 <u>폰</u> Xī	Xe Xi Xe Xi	5 X7 X8 🔜 🎀				
Maste	er Time Bar: () ps		Pointer: 27.06 ns	Interval: 27,06 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 🔺
₽	×1	BO	3					
P	x2	BO						
ø	F	B 0				I		
			•		· · · · · ·			• •
							0%	. 00:00:00

© 2010 Altera Corporation



FSM Viewer

- Open the FSM Viewer
 - Click **Tools**
 - Expand Netlist
 Viewers
 - Click State
 Machine Viewer





Examine State Machine

Quartus II - D:/Course/Digital Logic/stopper/	stoppe	er - stopper - [S	itate Machine V	/iewer pulse_FSM:FSM	l y_Q]	
& File Edit View Project Assignments Processing	Tools	Window Help				_ 8 ×
🛛 🗅 🖨 🖬 🎒 😹 🕺 🏝 📾 🕞 🗠 🕞 sto	pper		• 💢 =	/ 🥝 🤣 🛞 🖿 🕨	🕏 Þo 🏷 🗘 😓 🕹 🙋	
Project Navigator 🔤 🔺 🗙	ábo	stopper v	1 660	nulse FSM v	🕒 Compilation Benort - Flow Sum 🗌 🔍 St	ate Machine Viewer
🔄 Files			Y	paloo_r office		
abd pulse_FSM.v	-	State Machine:	stopper[pulse_F9	SM:FSMly_Q		<u> </u>
stopper.v						
327 984	13					
	€,					
	G					
				DATE: NO.		
	99					
	60			S0	S1 S2	
	-			- A	\forall \forall	
	-					
					- /	
		Source State	Destination State	Condition		
Hierarchy El Files Prosign Units		1 S0	SO	(!i_pulse) + (i_pulse).(!Resetn)		
Tasks 🛛 🗛 🗙		2 S0	S1	(i_pulse).(Resetn)		
Flow: Full Design		3 S1	SO	(!Resetn)		
T-UP		4 S1	S2	(Resetn)		
		5 S2	SO	(!i_pulse) + (i_pulse).(!Resetn)		
		6 S2	S2	(i_pulse).(Resetn)		
T Create Design						
E C Assign Constraints						
🗸 🗆 🕨 Compile Design						
🗸 🕀 Analysis & Synthesis						
🖌 🕂 Fitter (Place & Route)						
🖌 🖌 🖈 Assembler (Generate programming fi						
🖌 🕸 🕨 Classic Timing Analysis						
🗄 🕨 EDA Netlist Writer						
Program Device (Open Programmer)						
🗆 🔄 Verify Design						
Simulate Design		Transitions &	Encoding /			
× Message: 0 of 614		JA . remonitione A				- Locate
For Hole proce E1						
rur neip, press F1					j*u ≠ ⊠ — ≣ j idie	

© 2010 Altera Corporation



See the Circuit in RTL Viewer

Start the RTL Viewer

- Click Tools
- Expand the
 Netlist Viewers
 list
- Click RTLViewer

& Quartus II - D:/Course/Digital Logic/seg_	shift/seg_shift - seg_shift	
Project Navigator	Run EDA Simulation Tool Run EDA Timing Analysis Tool	*************
	ImeQuest Timing Analyzer Advisors	•
	 Chip Planner (Floorplan and Chip Editor) Design Partition Planner 	
	Netlist Viewers	🕨 🕄 RTL Viewer
	 SignalTap II Logic Analyzer In-System Memory Content Editor Logic Analyzer Interface Editor In-System Sources and Probes Editor SignalProbe Pins Programmer 	State Machine Viewer Etate Machine Viewer (Post-Mapping) Etate Inchnology Map Viewer
Hierarchy Files P Design Units	 Mega<u>Wi</u>zard Plug-In Manager SOPC <u>B</u>uilder Tcl Scripts 	OIIAR
Task 🗗 🔂 Start Project	Customize Options License Setup	



Examine the Circuit



© 2010 Altera Corporation



SignalTap II Embedded Logic Analyzer

- A soft logic analyzer
 - Instantiate as a module in your design
- Connects to the board on which a design is running
- Collects data when a trigger event occurs
- Displays data on your computer
- How does it work?

File Edit View Project Assignmen	s Processing	Tools Window Help				
D 😅 🖬 🗿 🍜 🖌 🖻 🕻 Project Navigator	3 M M	Run E <u>D</u> A Simulation Tool Run <u>E</u> DA Timing Analysis Tool	• • @			
Entity		Launch Design Space Explorer				
Cyclone II: EP2C35F672C6		🚺 TimeQuest Timing Analyzer				
0 oldunate		<u>A</u> dvisors	•			
		 Chip Planner (Floorplan and Chip Editor) Design Partition Planner Netlist Viewers 	,			
		📓 SignalTap II Logic A <u>n</u> alyzer				
AHierarchy E Files P Design Un	ts	In-System Memory Content Editor Logic Analyzer Interface Editor In-System Sources and Probes Editor SignalProbe Pins				
Tasks		Programmer				
Task 🗹		SOPC Builder	2			
🗄 🚞 Start Project		Tel Scripts				
🖃 🛄 Advisors		Customize				
🛨 🛄 Create Design		Customizer.				
E Compile Design		Options				
		Ficeuse bernh	_			

© 2010 Altera Corporation



SignalTap II Operation





Altera U. P. Teaching Materials for Digital Logic

Available for download: <u>http://university.altera.com</u>

Tutorials

- Teach students how to use Quartus II, etc.

Lab Exercises

- 10 digital logic exercises
- Complete with documentation and solutions (both in Verilog and VHDL)

IP Cores for DE boards

- VGA, Audio I/O, and others
- Great for projects



© 2010 Altera Corporation

Altera U.P. Web Site



University Program

→Edit Page →Downloads

🛛 Educational Materials 🛸 Announcements 🛸 Innovate Contests 🛸 Members 🛸 Support

Altera Home > University Program

Altera[®] University Program provides complete support for introducing students to digital technology. The support includes hardware, software, and teaching materials. Hardware support is in the form of Development and Education boards (DE1, DE2, and DE3), specially designed for use in teaching and research laboratories. Software support consists of the Quartus[®] II Computer Aided-Design software, Nios II soft processor, and simulation tools. Teaching materials comprise tutorials and ready-to-teach laboratory exercises for use in digital logic and computer organization courses.

The DE boards provide a high-quality, yet low-cost platform. To mitigate the cost of equiping teaching and research laboratories, Altera University Program boards are provided at a price which is at, or below, the manufacturing cost. Altera also offers a hardware donation program for qualified schools. Professors and lecturers can request support under this program through the <u>Members</u> section of this University Program website.

Also, Altera sponsors design contests for students around the world. These contests have been highly popular and have resulted in many innovative designs.

For more information about the Altera University Program and its offerings, click on the links below or read the <u>Altera</u> <u>University Program Overview</u> document.

- Educational Materials
- <u>University Program Members</u>
- Innovate Design Contests
- <u>Support</u>

🗹 Rate This Page

Products | End Markets | Technology | Training | Support | About Altera | Buy Online Jobs | Investor Relations | Contact Us | Site Map | Privacy | Legal Notice

Copyright © 1995-2009 Altera Corporation. All Rights Reserved.

RSS Feeds Email Updates

© 2010 Altera Corporation



Tutorials

- Tutorials
 - Getting Started with Altera's DE0 Lab Board
 - Introduction to Quartus II
 - With Verilog, or VHDL, or Schematic
 - Using library modules (LPMs)
 - With Verilog or VHDL
 - Quartus II Simulation
 - With Verilog or VHDL
 - Timing Considerations
 - With Verilog or VHDL
 - Signal Tap II
 - Hardware Debugging
 - Debugging with ModelSim

Introduction to Quartus II

Compose Aidel Design (CAD) software nodes it easy to implement a design logic circuit by using a pro- primative sign circuit, each as a ded programmable para surge (PEOA) clap. A styring PEOA CAD dow it Distributed Tepera I.
Torigin Tany Signoria State Signoria Signo
The CAD flow involves the following resp: • Design Eastry – the desired circuit is specified either by means of a schematic dispram, or by using a horizontal methods and the state of the state as specified with the state of the state in the FGA day • Functional Simulation – the synthesized circuit is word to wordly in functional convertises; this samelation does not take into account any transpinose.



Acrobat Document

© 2010 Altera Corporation



Digital Logic Lab Exercises

Ten lab exercises released

- From basic logic gates to simple processors:

• Lab 1: Switches, Lights, and Multiplexers

- Lab 2: Numbers and Displays
- Lab 3: Latches, Flip-flops, and Registers
- Lab 4: Counters
- Lab 5: Real-time Clock and Timers
- Lab 6: Adders, Subtractors, and Multipliers
- Lab 7: Finite State Machines
- Lab 8: *Memory Blocks*
- Lab 9: A Simple Processor
- Lab 10: An Enhanced Processor







Lab 1: Switches and Lights



© 2010 Altera Corporation





00

01

10

2

M

© 2010 Altera Corporation

ALTERA, ARRIA, CYCLONE, HARDCOPY, MAX, MEGACORE, NIOS, QUARTUS & STRATIX are Reg. U.S. Pat. & Tm. Off. and Altera marks in and outside the U.S.

W







© 2010 Altera Corporation



... Lab 1

Select and display one of three characters

Part 5



Rotate the word dE0 on three displays

Part 5

$SW_9 SW_8$	Character pattern					
00	d	E	0			
01	E	0	d			
10	0	d	E			

© 2010 Altera Corporation



... Lab 1



Extend your design from Part V so that is uses all four 7-segment displays on the DE0 board

$SW_9 SW_8$	Character pattern						
00		d	E	0			
01	d	E	0				
10	E	0		d			
11	0		d	E			

© 2010 Altera Corporation



Lab 7

FSM – Sequence recognizer

- There is an input w and an output z.
- Whenever w = 1 or w = 0 for four consecutive clock pulses the value of z has to be 1; otherwise, z = 0.
- Overlapping sequences are allowed, so that if w = 1 for five consecutive clock pulses the output z will be equal to 1 after the fourth and fifth pulses





© 2010 Altera Corporation



Lab 7

Part 1

Current S	State W	/ Ne	ext state	Ζ	Cu	rrent State	W	Ne	xt state	Ζ
A: 00000	0001 0	в:	000000010	0	A:	00000001	1	F:	000100000	0
B: 00000	0010 0	C:	00000100	0	B:	00000010	1	F:	000100000	0
C: 00000	0100 0	D:	000001000	0	C:	00000100	1	F:	000100000	0
D: 00000	1000 0	Е:	000010000	0	D:	000001000	1	F:	000100000	0
E: 00001	0000 0	Е:	000010000	1	Е:	000010000	1	F:	000100000	1
F: 00010	0000 0	в:	0000000 1 0	0	F:	000100000	1	G:	001000000	0
G: 00100	0000 0	в:	000000010	0	G:	001000000	1	н:	01000000	0
н: 01000	0000 0	в:	0000000 1 0	0	Н:	010000000	1	Ι:	100000000	0
I: 10000	0000 0	в:	000000010	1	I:	10000000	1	Ι:	10000000	1



Y_D(1) <= (y_Q(0) OR y_Q(5) OR y_Q(6) OR y_Q(7) OR y_Q(8)) AND NOT (w);



Lab 7

FSM – Sequence recognizer

 Describe the state table for the FSM by using a VHDL CASE statement in a PROCESS block, and use another PROCESS block to instantiate the state flip-flops.

ARCHITECTURE Behavior OF part2 IS SIGNAL Clock, Resetn, w, z : STD_LOGIC; TYPE State_type IS (A, B, C, D, E, F, G, H, I); SIGNAL y_Q, Y_D : State_type; Part 2



© 2010 Altera Corporation

ALTERA, ARRIA, CYCLONE, HARDCOPY, MAX, MEGACORE, NIOS, QUARTUS & STRATIX are Reg. U.S. Pat. & Tm. Off. and Altera marks in and outside the U.S.



•••