

Quartus II 软件设计：基础



© 2013 Altera Corporation—Confidential



课程目的

- 建立Quartus II工程
- 设计输入文件
- 编译
- 设置和分配
- 管脚分配
- 配置/编程

Quartus II 软件设计：基础

Altera和Altera器件



© 2013 Altera Corporation—Confidential



全系列的产品解决方案



P O W E R I N G Y O U R I N N O V A T I O N

MAX®
Series

Cyclone®
Series

Arria®
Series

Stratix®
Series

ENPIRION

CPLDs

Lowest Cost,
Lowest Power

FPGAs

Cost/Power Balance
SoC & Transceivers

FPGAs

Mid-range FPGAs
SoC & Transceivers

FPGAs

Optimized for
High Bandwidth

PowerSoCs

High-efficiency
Power Management

Embedded Soft and Hard Processors

Nios® II
ARM

Design Software



DSPBuilder

Development Kits



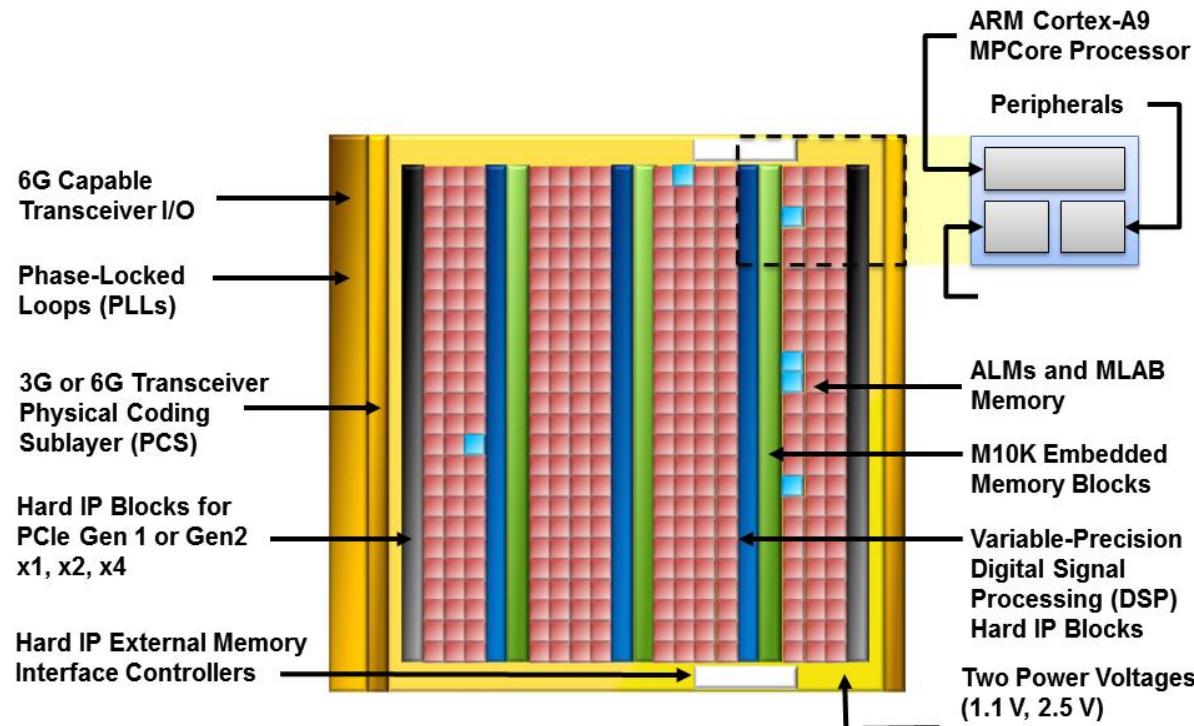
Intellectual Property (IP)

- Industrial
- Computing
- Enterprise

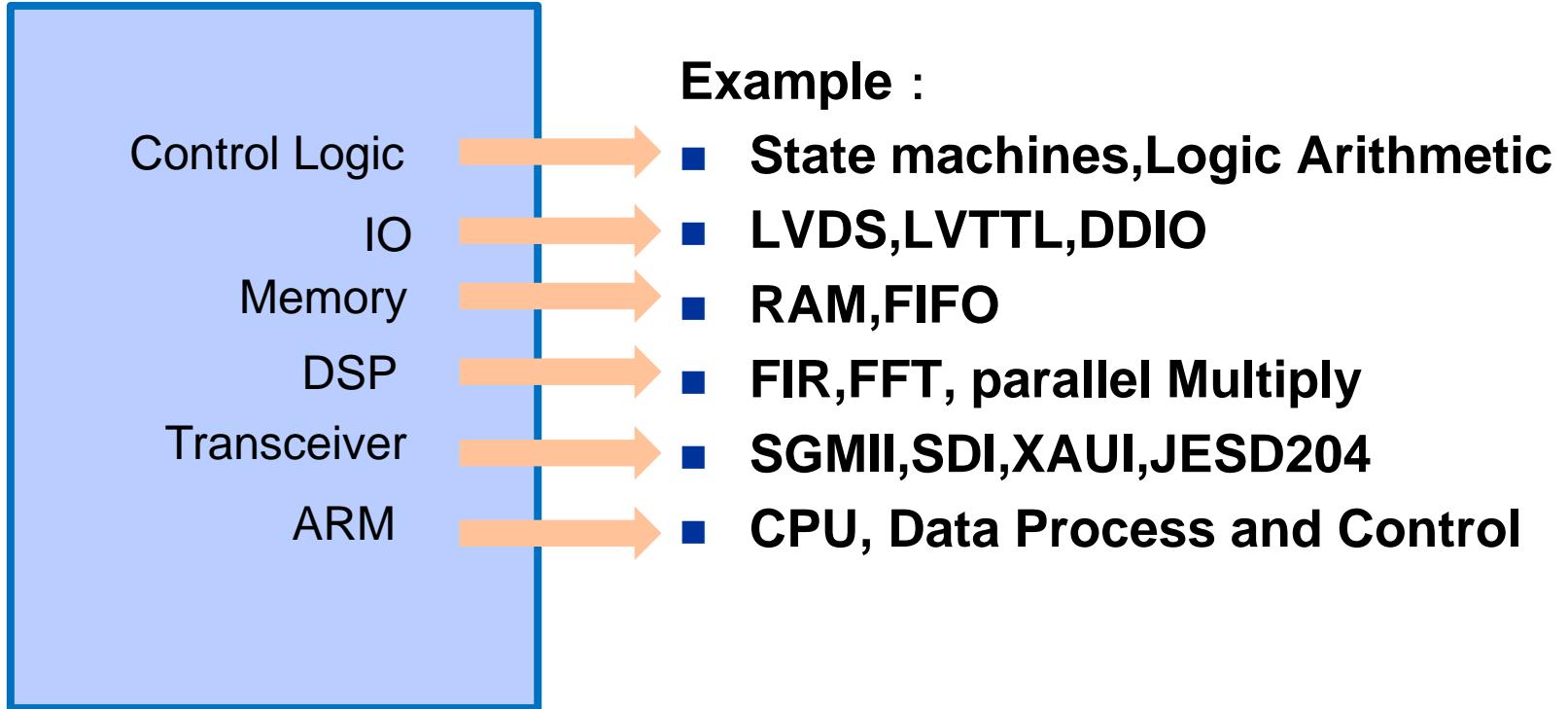


现在的PLD包含哪些东西？

- Logic
- Registers
- Multifunction I/Os
- Memory
- Transceivers
- DSP
- Clock networks
- PLLs
- ARM
- ADC



FPGA/CPLD能实现啥？



如何设计？



设计构思



设计代码



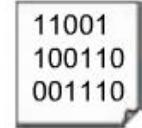
设计代码



FPGA
Design
S/W



综合 + 布局布线



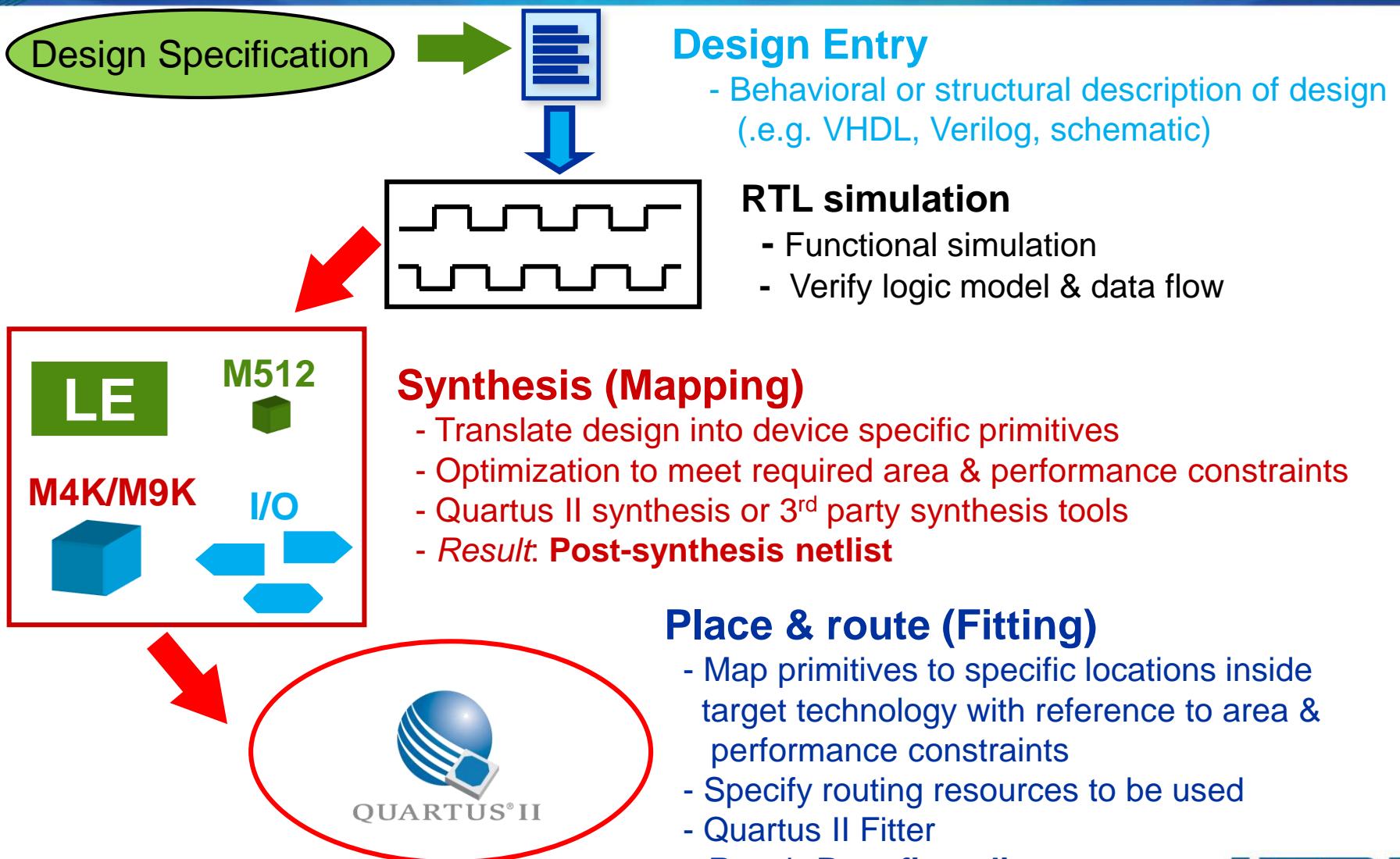
编程文件



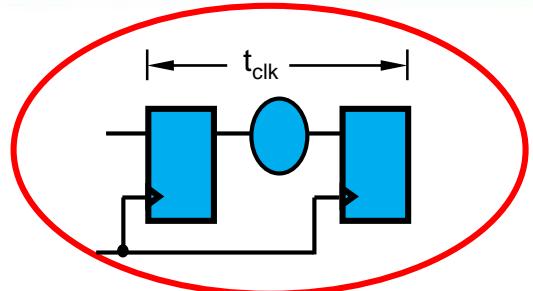
闪存



Typical PLD Design Flow

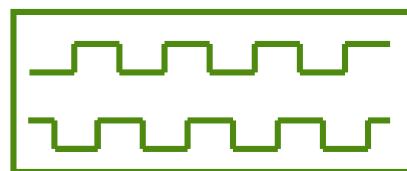


Typical PLD Design Flow



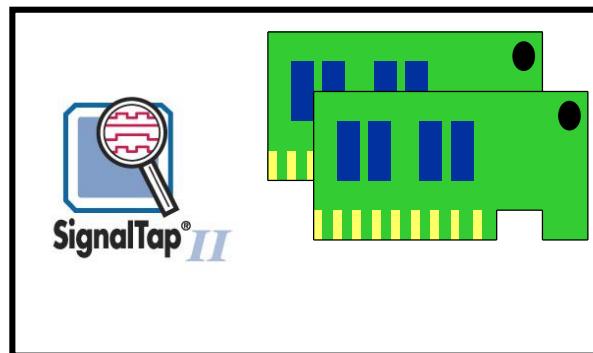
Timing analysis (TimeQuest Timing Analyzer)

- Verify performance specifications were met
- Static timing analysis



Gate level simulation (optional)*

- Simulation with timing delays taken into account
- Verify design will work in target technology



PC board simulation & test

- Simulate board design
- Program & test device on board
- Use **SignalTap II** Logic Analyzer or other on-chip tools for debugging

Quartus II 软件设计：基础

软件功能简介



© 2013 Altera Corporation—Confidential



Quartus II Design Software Features

Category	Quartus II Tool/Feature
Operating System	<ul style="list-style-type: none">▪ 32/64-bit Windows and Linux support
Licensing	<ul style="list-style-type: none">▪ Node-locked and network licensing support
Project creation	<ul style="list-style-type: none">▪ New Project Wizard
Design entry	<ul style="list-style-type: none">▪ Text Editor (HDL support)▪ Schematic Editor▪ State Machine Editor▪ MegaWizard Plug-In Manager▪ Qsys system design tool▪ DSP Builder Standard/Advanced Blockset▪ OpenCL support▪ 3rd party design entry tool support
Constraint (assignment) entry	<ul style="list-style-type: none">▪ Assignment Editor▪ Text Editor support of Synopsys Design Constraints (SDC)▪ Pin Planner▪ Scripting (Tcl) support
Design processing (synthesis and fitting)	<ul style="list-style-type: none">▪ Quartus Integrated Synthesis (QIS)▪ 3rd party EDA synthesis tool support▪ Quartus II Fitter
Design evaluation and debugging	<ul style="list-style-type: none">▪ RTL Viewer▪ Technology Map Viewers

Quartus II Design Software Features (cont.)

Category	Quartus II Tool/Feature
Power and signal analysis	<ul style="list-style-type: none">▪ PowerPlay power analyzer▪ SSN Analyzer
Static timing analysis	<ul style="list-style-type: none">▪ TimeQuest timing analyzer
Simulation	<ul style="list-style-type: none">▪ ModelSim®-Altera Starter Edition▪ ModelSim-Altera Edition▪ 3rd party EDA simulation tool support
Chip layout viewing and modification	<ul style="list-style-type: none">▪ Chip Planner▪ Resource Property Editor
Programming file generation	<ul style="list-style-type: none">▪ Quartus II Assembler
FPGA/CPLD programming	<ul style="list-style-type: none">▪ Quartus II Programmer
Hardware debugging tools	<ul style="list-style-type: none">▪ SignalTap™ II embedded logic analyzer▪ In-System Sources and Probes▪ SignalProbe incremental routing
Design optimization and productivity improvement	<ul style="list-style-type: none">▪ Design Assistant▪ Quartus II incremental compilation▪ Physical synthesis optimization▪ Design Space Explorer (DSE)

Welcome to the Quartus II Software!



The image shows the initial start screen of the Quartus II software. It features a blue and white design with a stylized globe icon. The title "QUARTUS® II" is prominently displayed. Two main sections are visible: "Start Designing" on the left and "Start Learning" on the right. A yellow callout box on the right side contains the text "Turn on or off in Tools → Options".

X

QUARTUS® II

Start Designing

Designing with Quartus II software requires a project

**Create a New Project
(New Project Wizard)**

Open Existing Project

Open Recent Project:

- C:/altera_trn/VER/lab4a/reg16.qpf
- C:/altera_trn/VER/lab4b/counter.qpf
- C:/altera_trn/VER/lab5a/mult_control.qpf
- C:/altera_trn/VER/lab5b/mult8x8.qpf

**Getting Started
With Quartus® II Software**

Start Learning

The audio/video interactive tutorial teaches you the basic features of Quartus II software

Open Interactive Tutorial

**Turn on or off in
Tools → Options**

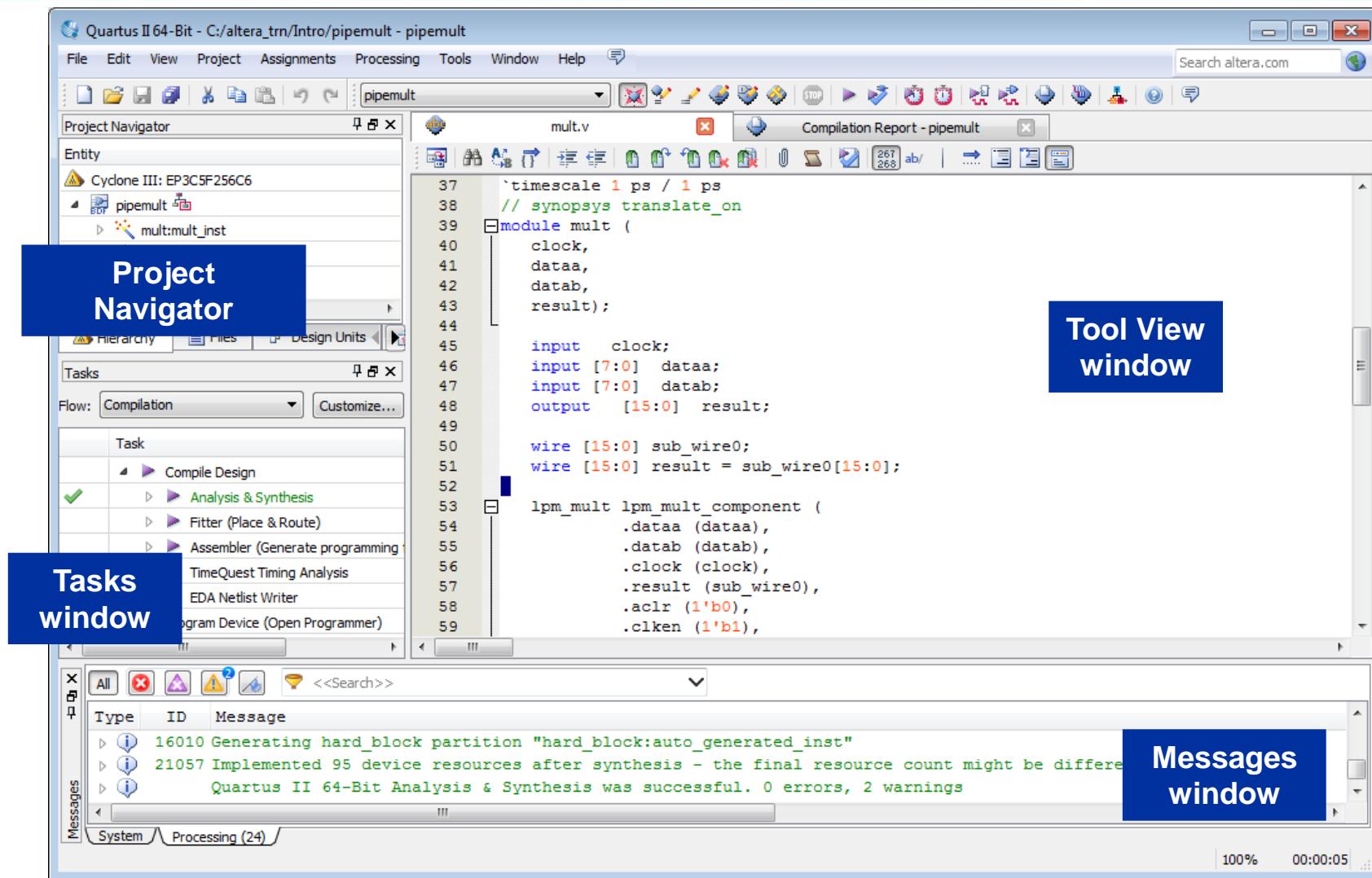
Don't show this screen again

Literature Training Online Demos Support

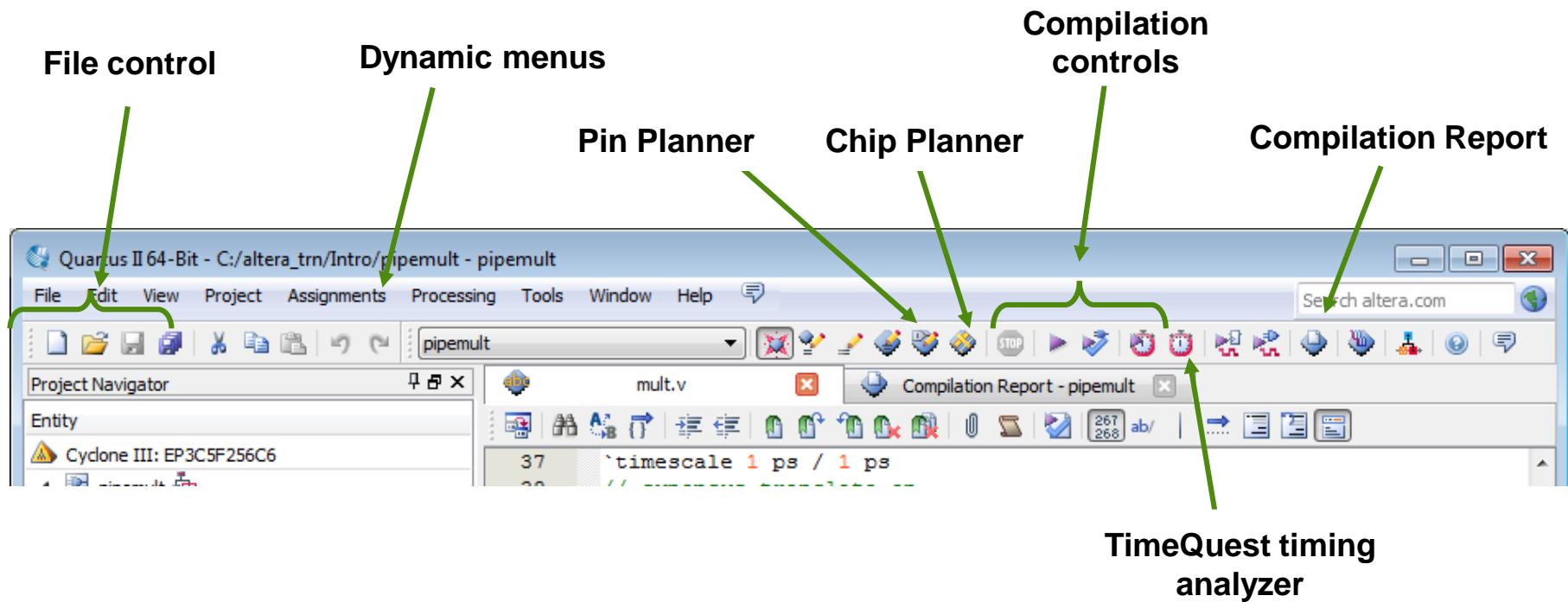
ALTERA.

ALTERA.
MEASURABLE ADVANTAGE™

Quartus II Default Operating Environment



Main Toolbar



To Customize Toolbars
Tools → Customize...

Tips & Tricks Advisor

Help menu → Tips & Tricks

The screenshot shows the 'Tips and Tricks' window from the Quartus II software. The window title is 'Tips and Tricks - C:/altera_trn/Intro/pipemult - pipemult'. The menu bar includes File, Edit, Tools, Window, Help, and a search bar for 'Search altera.com'. The left sidebar contains a tree view of tips:

- What's New in this Release
- Provide feedback to Altera
- Quartus II Features
 - Detach windows from the frame in the Quartus II software
 - Get advice on optimizing your design and the features of...
 - Get an Early Timing Estimate** (highlighted)
 - Use Incremental Compilation
 - Use SignalProbe to quickly pull out internal signals to pins
 - Use the PowerPlay Power Analyzer to check for power u...
 - Use Netlist Viewers to view your design schematic
 - Set up file type associations in Windows
- Software Options
 - Generate Compact Report Table Format
 - Additional report file options
 - Run Process at Lower Priority
 - Update assignments to disk immediately
 - Suppress Messages
 - Color messages during command-line compilation
 - Use an External Text Editor
 - Change the Tooltip Delay
- Project Settings
 - Enable Version-Compatible Database
 - Hide Entity Name
 - Specify the output directory for compilation results
 - Specify what is done during a normal compilation
 - Choose how the Fitter will process your design
 - Use Physical Synthesis to improve performance

The main content area displays a tip titled 'Get an Early Timing Estimate'.

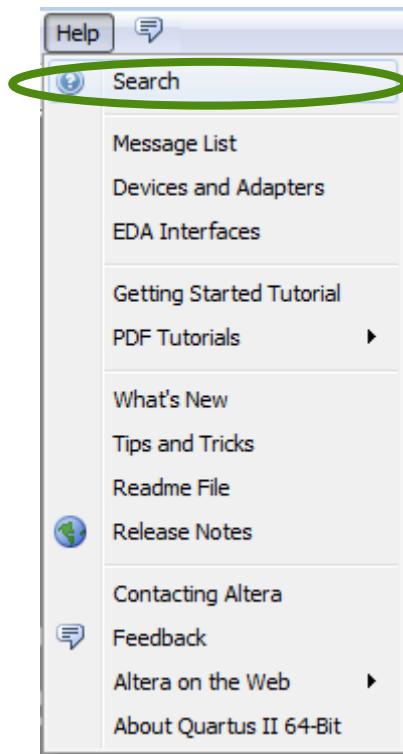
Recommendation	You can get an early timing estimate without running a full compilation.
Description	You can use the Start Early Timing Estimate command on the Processing menu to get a full timing report based on estimated delays for the design. This command can run the Fitter up to ten times faster than a full fit and produces estimated delays within 20% of what a full compilation can achieve.
Action	Use the Start Early Timing Estimate command on the Processing menu to run an early timing estimate. You can specify settings for the early timing estimate in the Settings dialog box when a project is open. Open Settings dialog box - Early Timing Estimate page

Provides useful instructions on using the Quartus II software & links to settings.

Available sections include:

- New features in current release**
- Helpful features and project settings available to designers**

Built-In Help System



The screenshot shows the Quartus II Help v13.0 web browser-based help system in a Windows Internet Explorer window. The URL in the address bar is file:///C:/altera/13.0/quartus/common/help/webhelp/master.htm#mergedProjects/optimize/lock/l.

Key features highlighted in the screenshot:

- Shortcut to Altera forums**: A green arrow points from the "Forums" link in the top navigation bar to a blue button labeled "Expand all topics".
- Show All**: A green arrow points from the "Show All" link in the search sidebar to the same "Expand all topics" button.
- Search Bar**: A green circle highlights the search bar in the top left corner of the browser window.
- Search Sidebar**: A sidebar on the left contains a search input field, checkboxes for search options (Show results that include all search words, Highlight search results), and a "Search results per page" dropdown set to 10.
- Welcome to the Quartus II Software**: The main content area displays the welcome message and highlights three topics: Quartus II Highlights, Design Capabilities, and NativeLink Integration with other EDA Tools.
- Design Flow Diagram**: A diagram shows the design flow from "Design Entry" to "RTL Simulation".
- Page Footer**: The footer includes the Altera logo, the text "MEASURABLE ADVANTAGE™", and a copyright notice: "© 2013 Altera Corporation—Confidential".

Web browser-based allows
for easy search in page

Interactive Tutorial

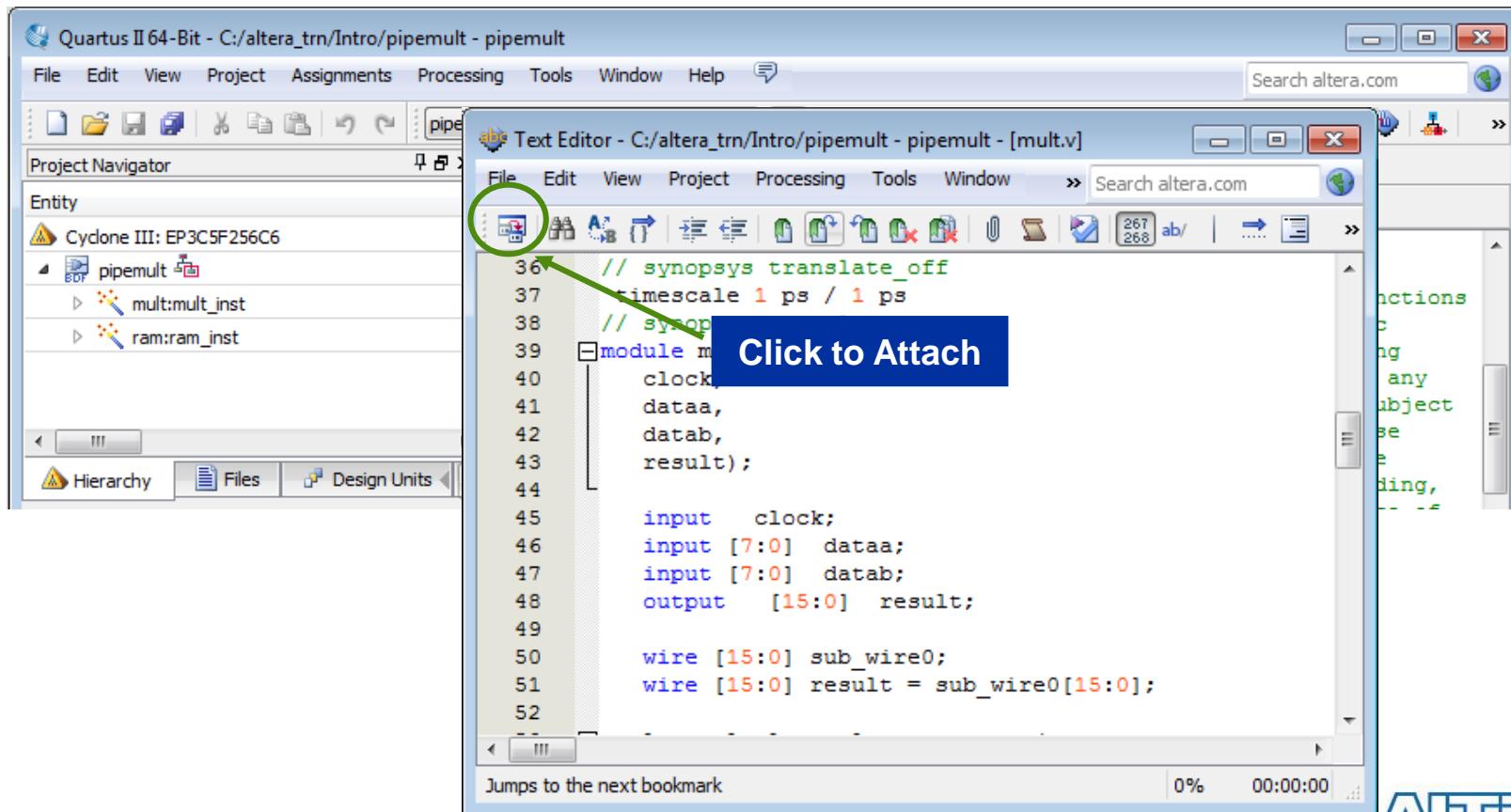
The screenshot shows a Microsoft Internet Explorer window displaying the "Quartus II Getting StartedTutorial" help page. The URL in the address bar is `C:\altera\12.1\quartus\common\help\tutorial\qtutorial.htm`. The page content is titled "Quartus II Tutorial Modules" and lists nine tutorial modules with their descriptions and estimated times:

Tutorial Module	Description	Estimated Time
Quartus II Introduction	(5 minutes)	
Create a Design	(30 minutes)	
Compile a Design	(40 minutes)	
Run Timing Analysis	(40 minutes)	
Configure a Device	(20 minutes)	
Incremental Compilation	(40 minutes)	
SignalTap II Logic Analyzer	(40 minutes)	
Create a Qsys System	(20 minutes)	

A blue callout box in the bottom left corner of the browser window contains the text: "Help menu → Getting Started Tutorial or from the Getting Started window".

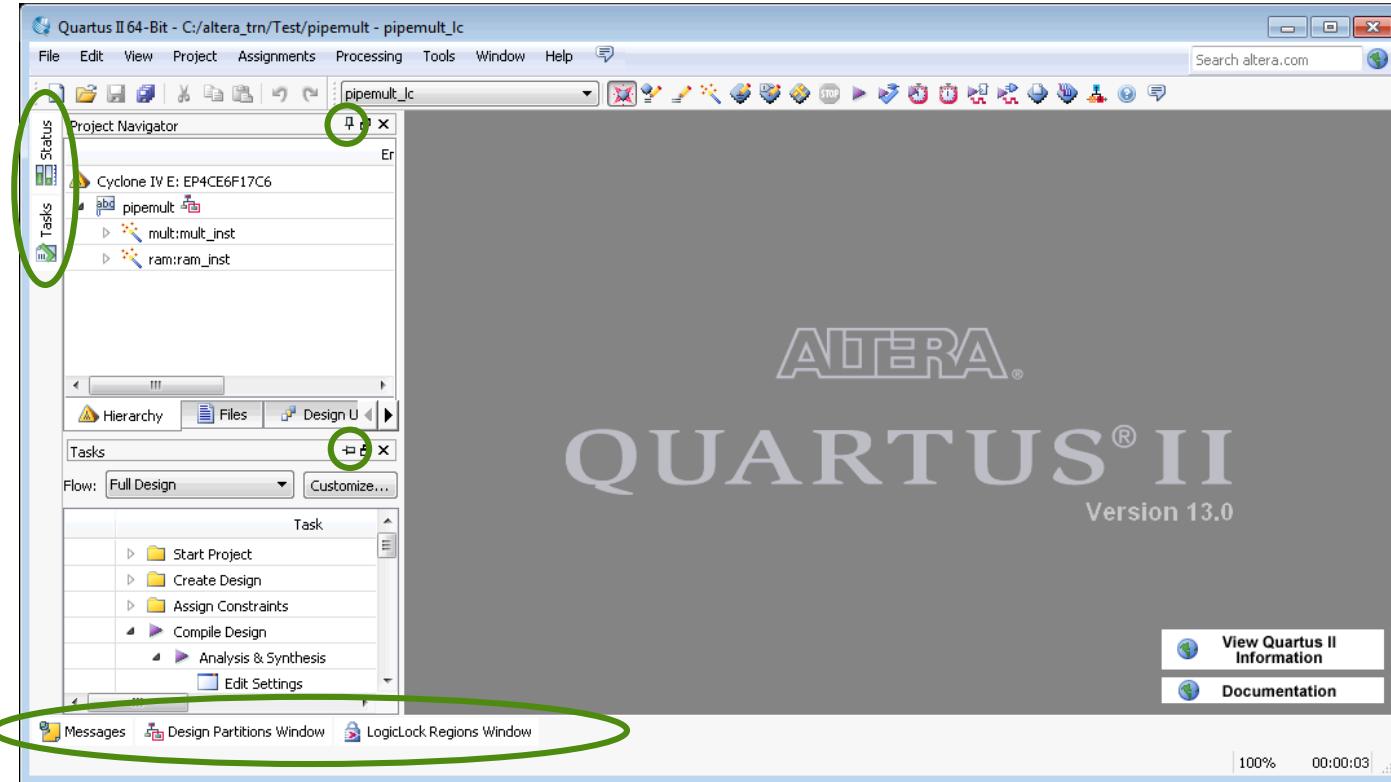
Detachable Windows

- Separate child windows from the Quartus II GUI frame
(Window menu → Detach/Attach Window)



Auto Hide/Show Dockable Windows

- Allow user to hide/unhide dockable windows
- Mouse over the window bar and it will be visible
- User can pin the window to keep it visible

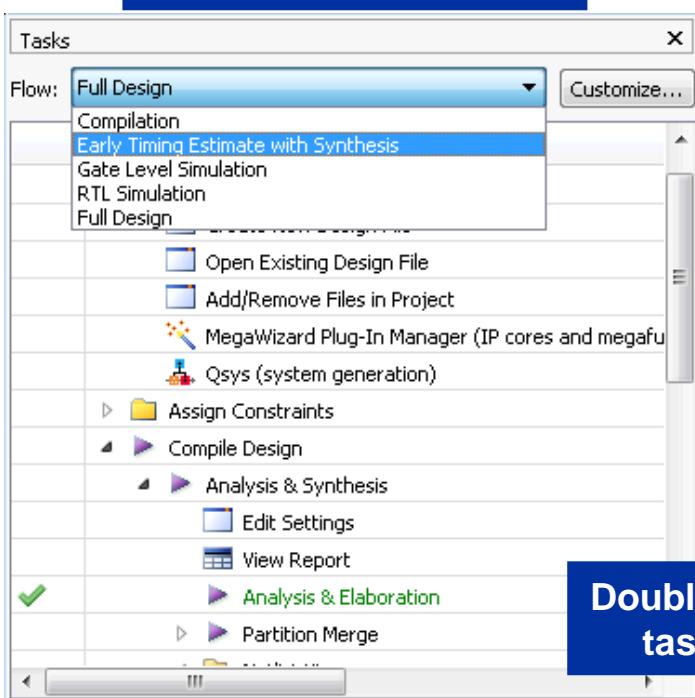


Tasks Window

- Easy access to most Quartus II functions
- Organized into related tasks within task flows

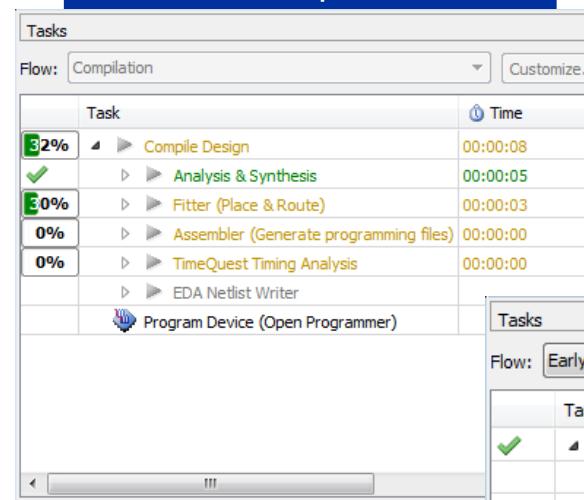
Full Design Flow

Perform all project tasks



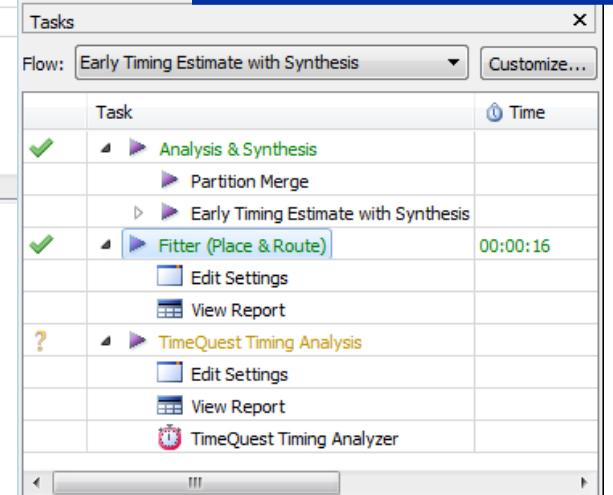
Compilation Flow

Focus on compilation tasks



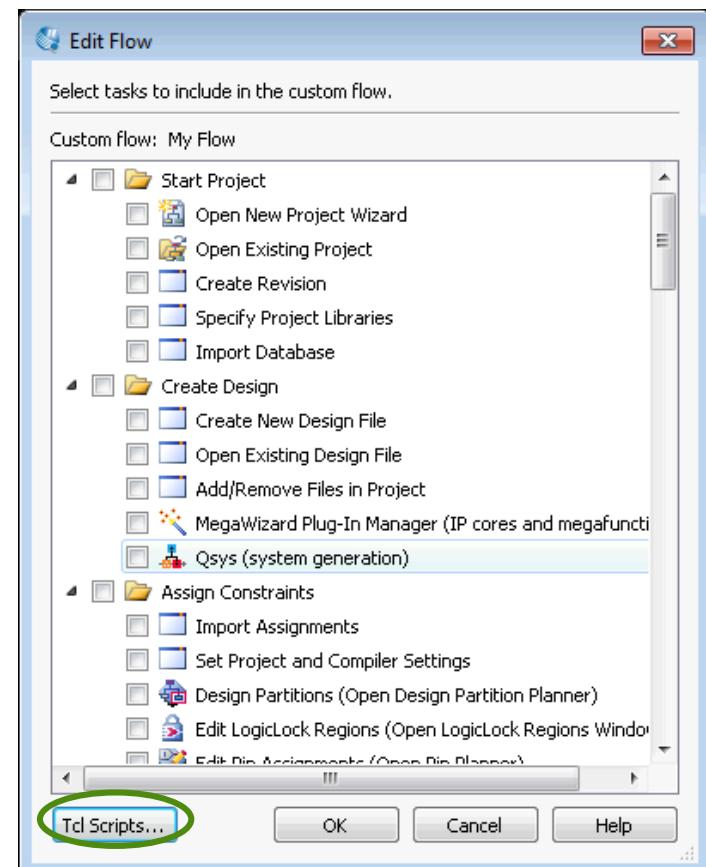
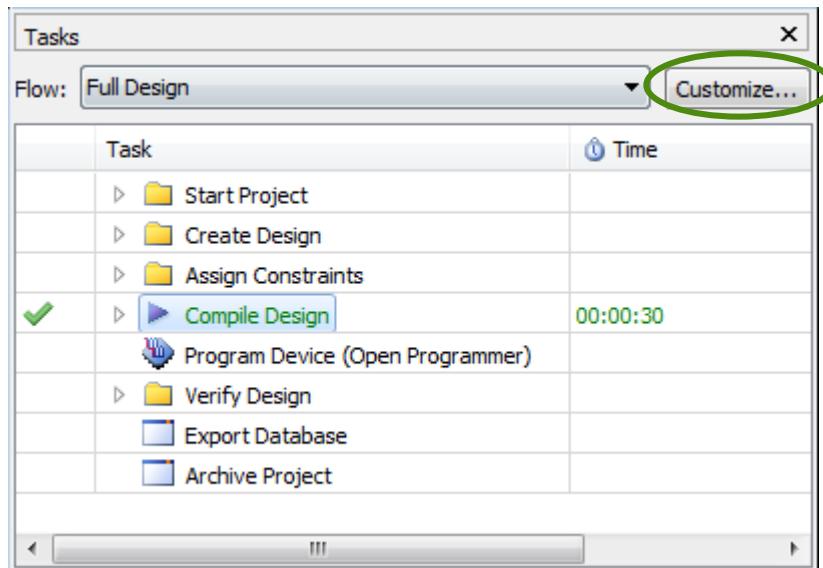
Double-click any
task to run

Early Timing Estimate
Faster compile iterations



Custom Task Flow

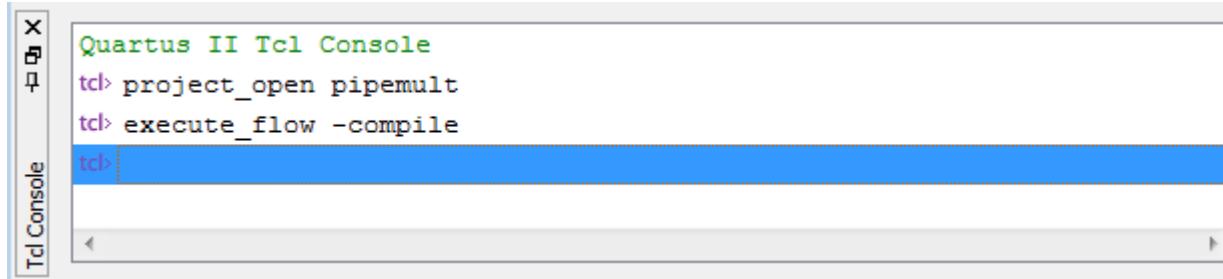
- Customize the Tasks display
- Add Tcl scripts for quick access



Tcl Console Window

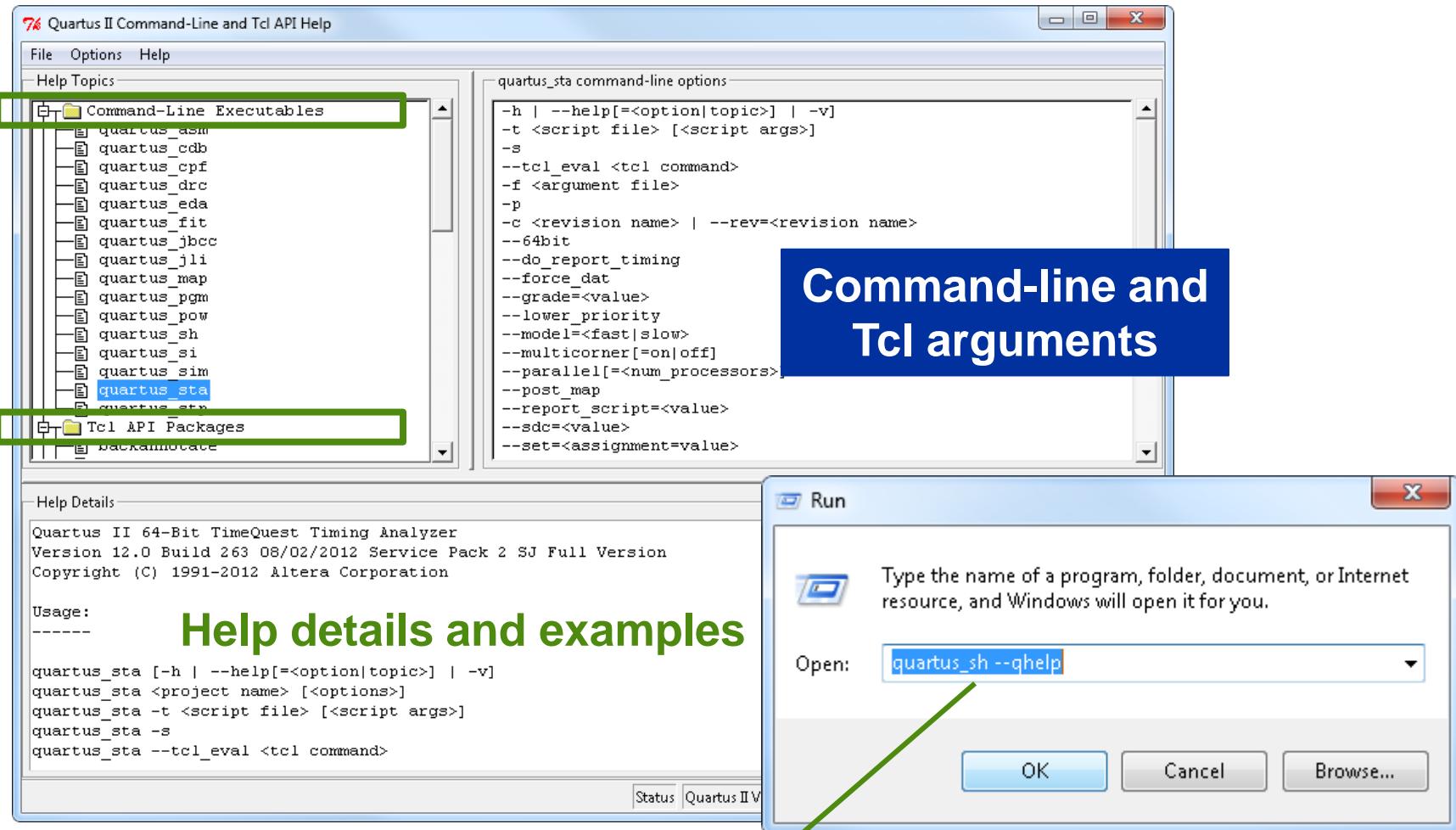
- Enter and execute Tcl commands directly in the GUI

View menu → Utility Windows → Tcl Console



- Execute from command-line using Tcl shell
 - quartus_sh -shell
- Run complete scripts from Tools menu ⇒ Tcl Scripts...

Tcl and Command Line Help



C:\<install dir>\quartus\bin64\quartus_sh --qhelp

ALTERA
MEASURABLE ADVANTAGE™

Further Information

- **On-line training classes**
 - [Introduction to Tcl](#)
 - [Quartus II Software Tcl Scripting](#)
- **Quartus II Handbook Volume 2 Chapters 2 & 3**
- **Tcl references on-line**

Class Agenda

- **Quartus II Projects**
- Design Entry
- Quartus II Compilation
- Settings & Assignments
- I/O Planning
- Programming/Configuration

Quartus II软件设计:基础

Quartus II Projects



© 2013 Altera Corporation—Confidential



Quartus II Projects Section Objectives

- Create a project
- Open an existing project
- Name the basic Quartus II project files
- Use Quartus II features to manage projects

Quartus II Projects

■ Description

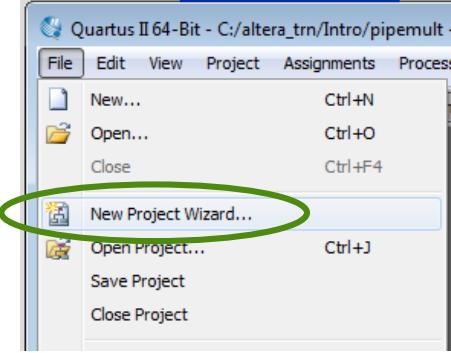
- Collection of related design files & libraries
- Must have a designated top-level entity
- Target a single device
- Store settings in Quartus II Settings File (.QSF)
- Compiled netlist information stored in **db** folder in project directory

■ Create new projects with New Project Wizard

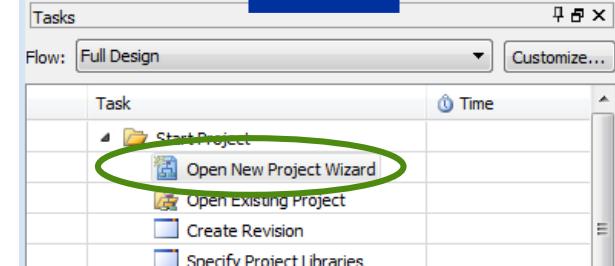
- Can be created using Tcl scripts

New Project Wizard

File menu



Tasks



Select working directory

Name of project can be any name; recommend using top-level file name

New Project Wizard

Directory, Name, Top-Level Entity [page 1 of 5]

What is the working directory for this project?
C:\altera_trn\Intro

What is the name of this project?

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.

Top-level entity does not need to be the same name as top-level file name

Tcl: *project_new <project_name>*

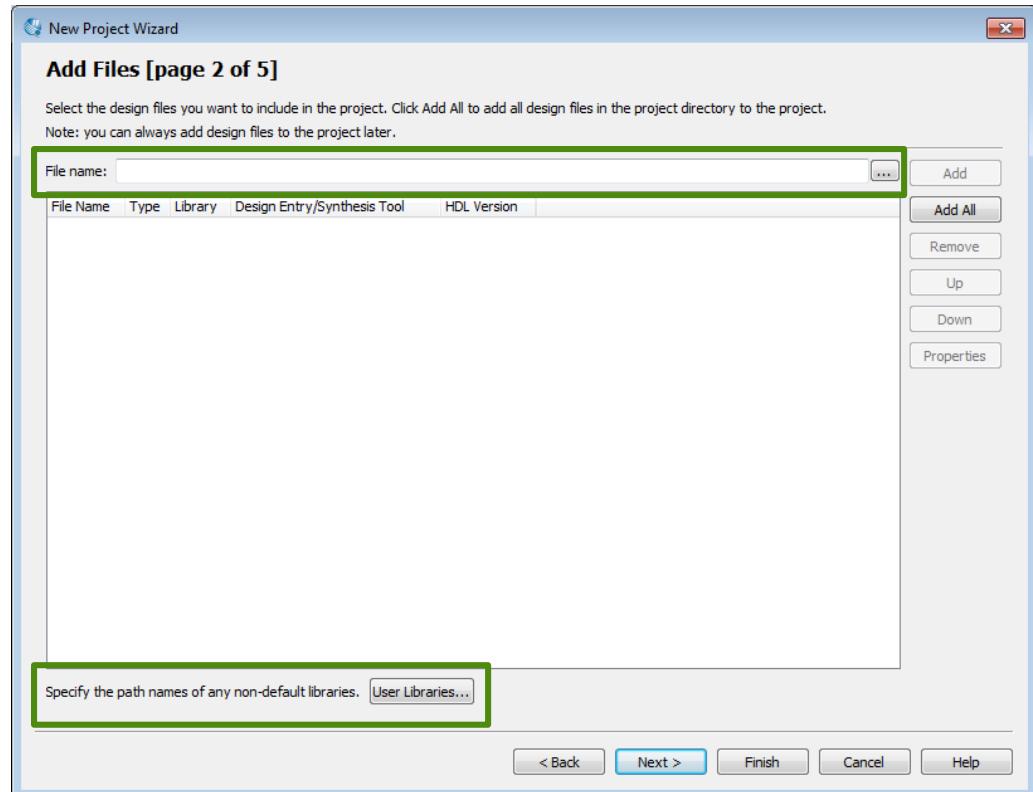
Add Files

■ Add design files

- Graphic
- VHDL
- Verilog
- SystemVerilog
- EDIF
- VQM

■ Add library paths

- User libraries
- MegaCore® library
- AMPPSM library
- Pre-compiled VHDL packages



Tcl: set_global_assignment -name VHDL_FILE <filename.vhd>

Tcl: set_global_assignment -name USER_LIBRARIES <library_path_name>

Device Selection

New Project Wizard

Family & Device Settings [page 3 of 5]

Select the family and device you want to target for compilation.

Device family

Family: Cyclone IV GX

Devices: All

Target device

Auto device selected by the Fitter

Specific device selected in 'Available devices' list

Other: n/a

Available devices:

Name	Core Voltage	Package	Pin count	Speed grade	IA	GXB Receiver Channel PMA
EP4CGX15BF14A7	1.2V	14400	81	2	2	2
EP4CGX15BF14C6	1.2V	14400	81	2	2	2
EP4CGX15BF14C7	1.2V	14400	81	2	2	2
EP4CGX15BF14C8	1.2V	14400	81	2	2	2
EP4CGX15BF14I7	1.2V	14400	81	2	2	2
EP4CGX15BN11C7	1.2V	14400	81	2	2	2
EP4CGX15BN11C8	1.2V	14400	81	2	2	2
EP4CGX15BN11I7	1.2V	14400	81	2	2	2

Companion device

HardCopy:

Limit DSP & RAM to HardCopy device resources

Choose device family

Show in 'Available devices' list

Package: Any

Pin count: Any

Speed grade: Any

Name filter:

Show advanced devices HardCopy compatible only

Filter device list

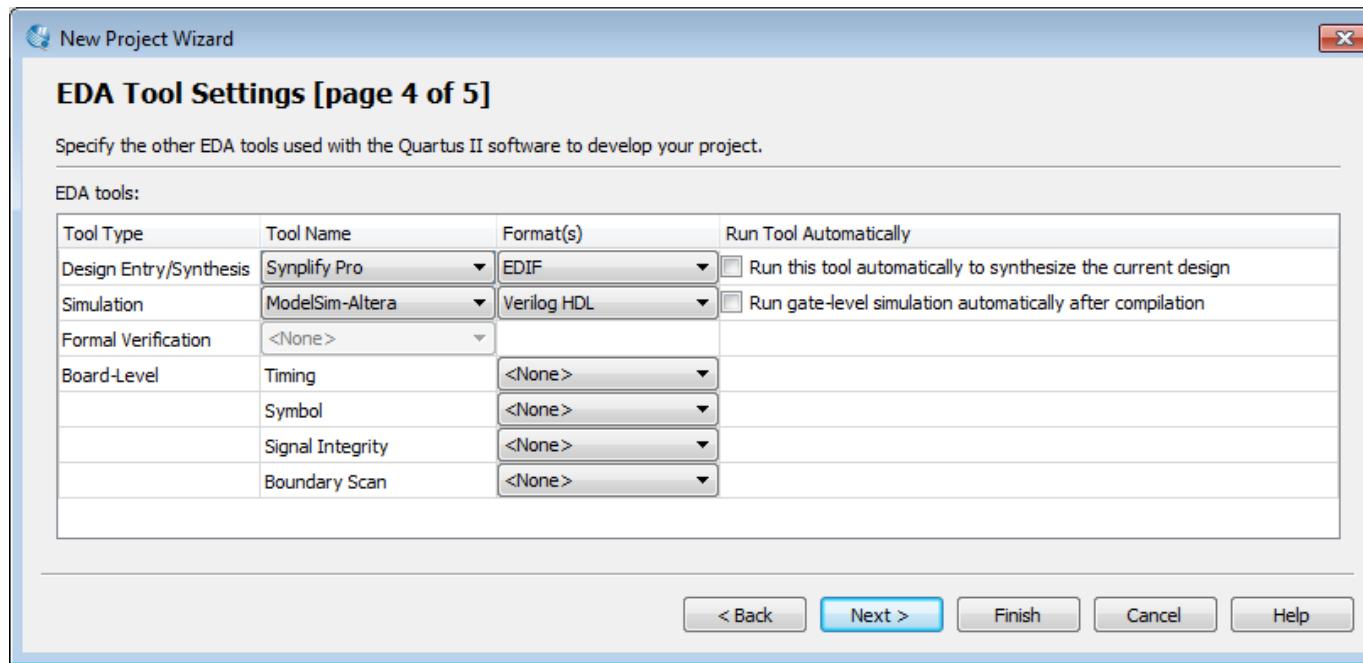
Choose specific part from list

Tcl: set_global_assignment -name FAMILY "device family name"

Tcl: set_global_assignment -name DEVICE <part_number>

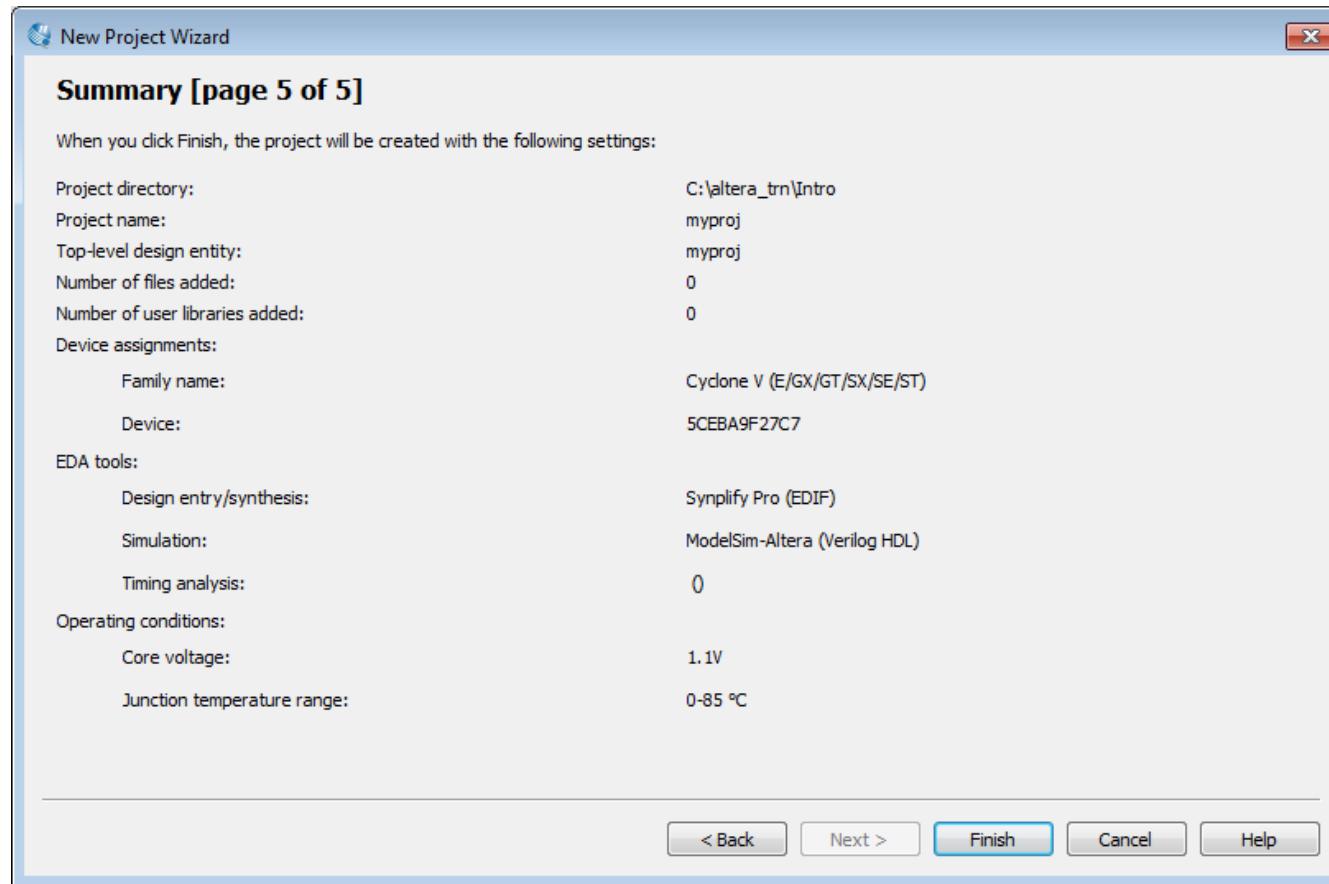
EDA Tool Settings

- Choose EDA tools and file formats
- Settings can be changed or added later

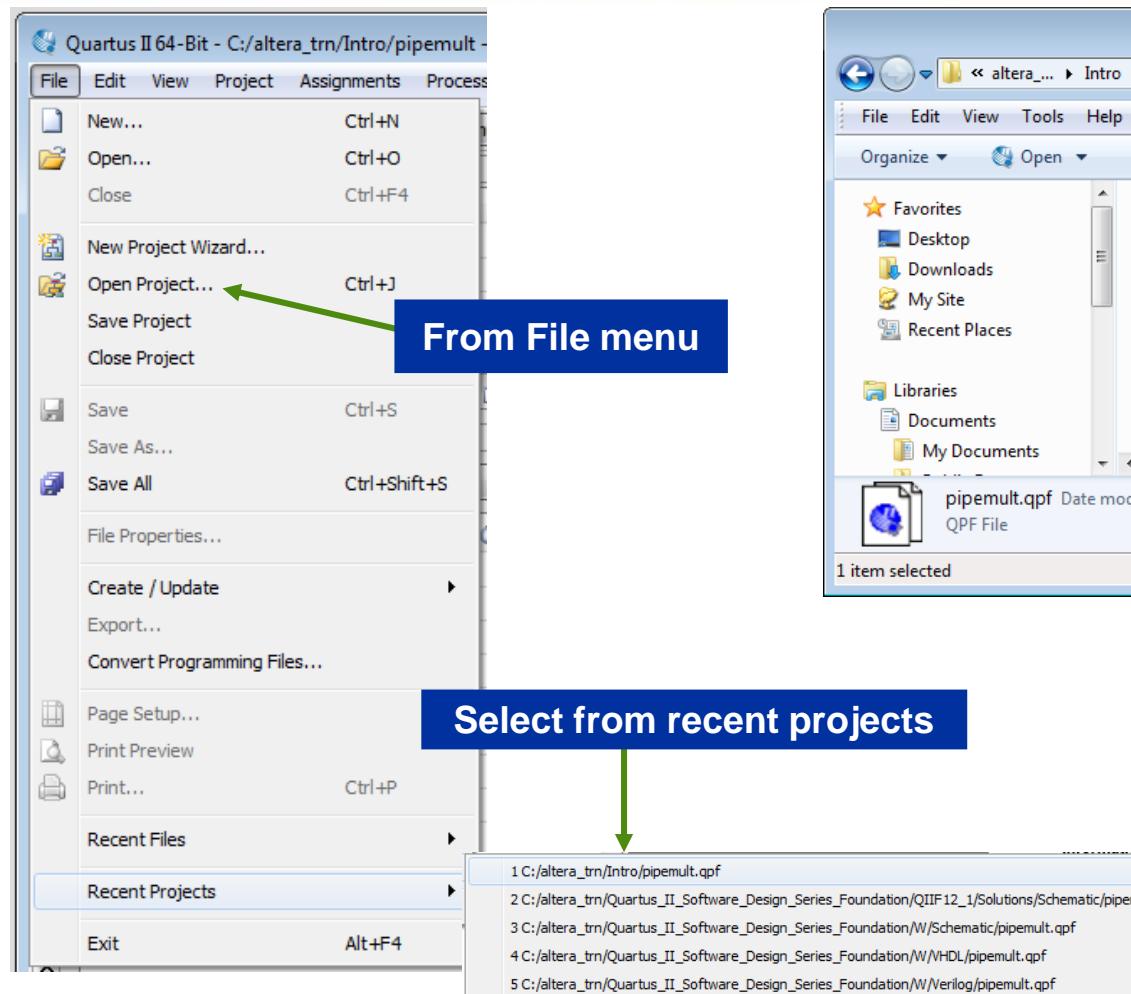


See handbook for Tcl command format

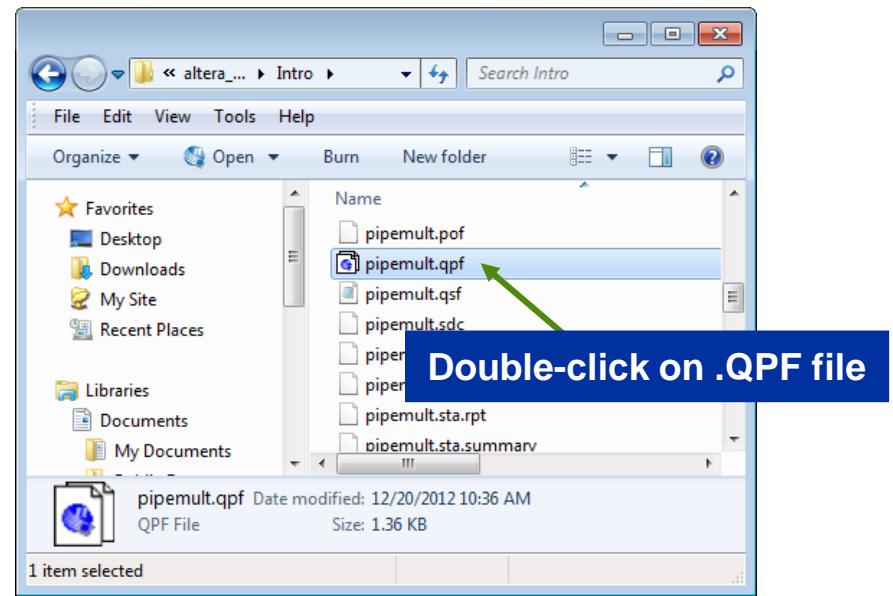
■ Review results & click Finish when done



Opening an Existing Project

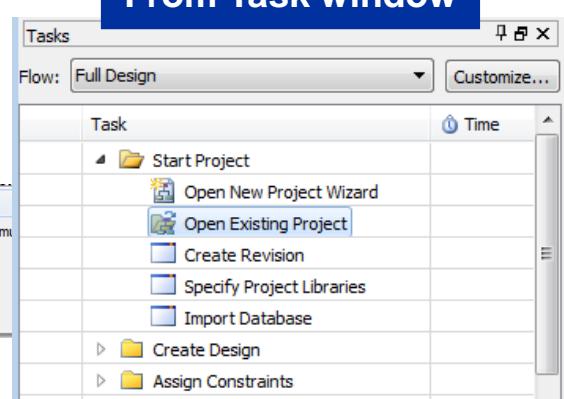


From File menu



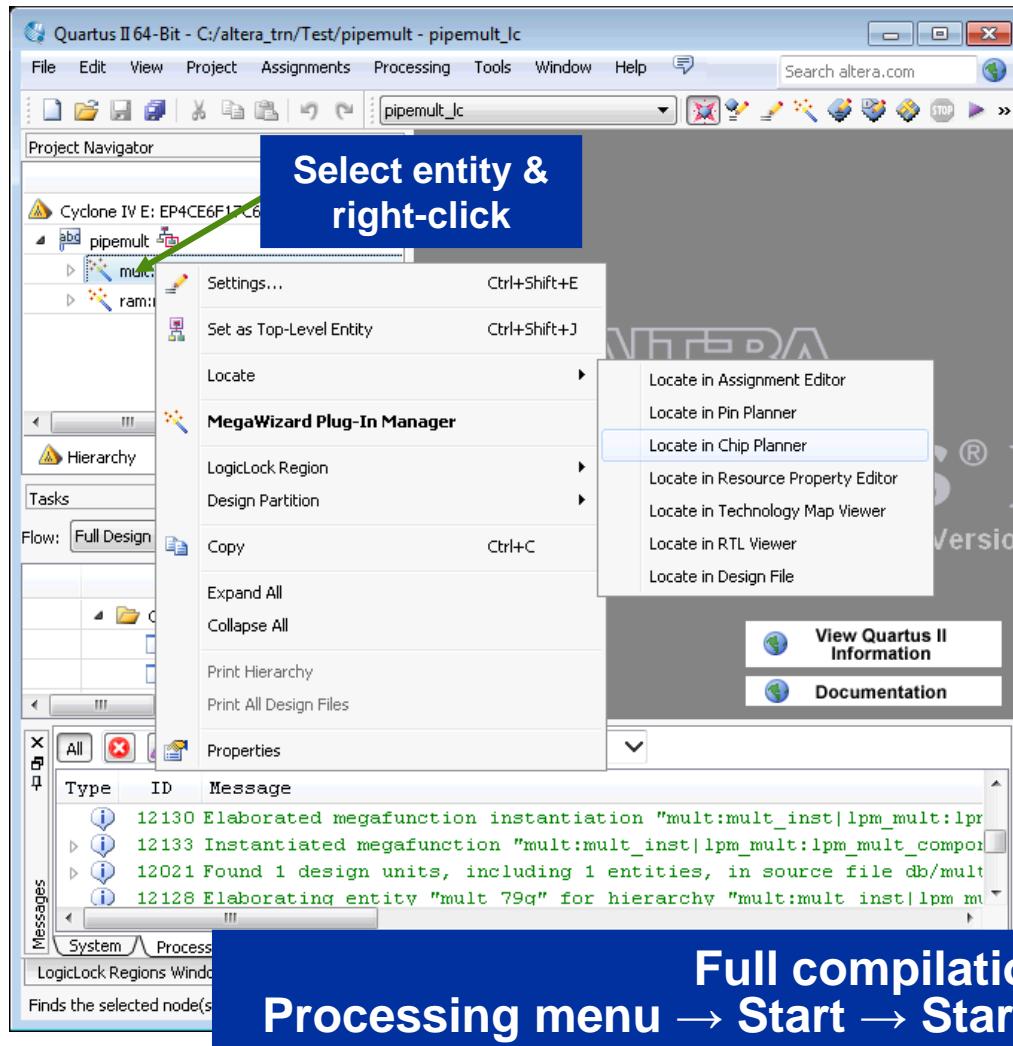
Double-click on .QPF file

From Task window



Tcl: *project_open <project_name>*

Project Navigator – Hierarchy Tab

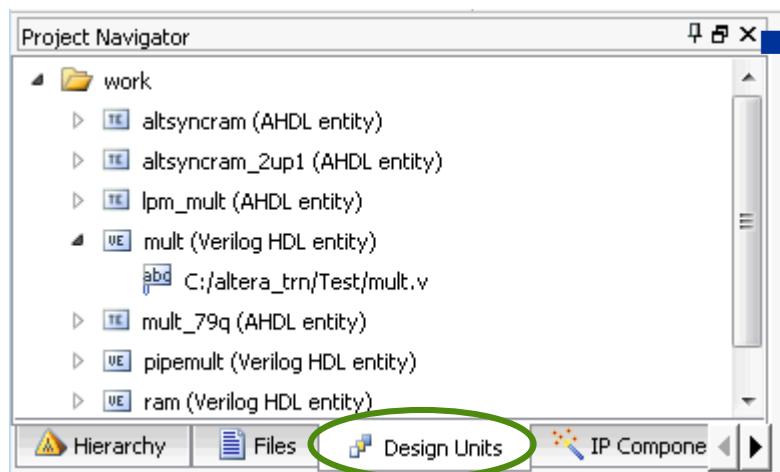
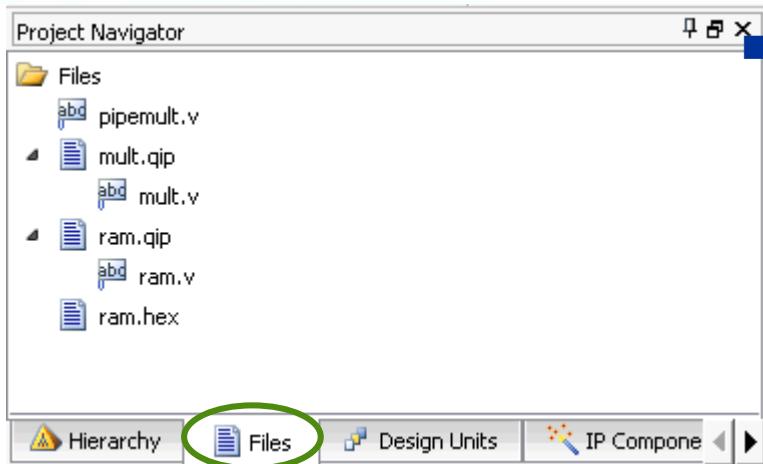


- Displays project hierarchy after project is analyzed

- **Uses**

- Set top-level entity
- Set incremental design partition
- Make entity-level assignments
- Locate in design file or viewers/floorplans
- View resource usage

Files & Design Units Tabs



Files tab

- Shows files explicitly added to project
 - Open files
 - Remove files from project
 - Set new top-level entity
 - Specify VHDL library
 - Select file-specific synthesis tool
- Can also use **Project** menu ⇒ **Add/Remove Files in Project...**

Design Units tab

- Displays design unit & type
 - VHDL entity and architecture
 - Verilog module
 - AHDL (Altera HDL) subdesign
 - Block diagram filename
- Expanded unit displays file which instantiates design unit

Quartus II Project Files & Folders

- **Quartus II Project File (.QPF)**
- **Quartus II Defaults File (.QDF)**
- **Quartus II Settings File (.QSF)**
- **db folder**
 - Contains compiled design information
 - May also see **incremental_db** for incremental compilation information
- **Synopsys® Design Constraints (.SDC)**
 - Contains timing constraints

Project & Default Files

■ Quartus II Project File (QPF)

- Quartus II version
- Time stamp
- Active revision(s)

fir_filter.QPF

```
QUARTUS_VERSION = "13.0"
DATE = "08:37:10  May 13, 2013"

# Revisions

PROJECT_REVISION = "filtref"
PROJECT_REVISION = "filtref_new"
```

■ Quartus II Defaults Files (QDF)

- Stores Quartus II project setting & assignment defaults
- Example names: *assignment_defaults.QDF* or *<revision_name>_assignment_defaults.QDF*
- Found in local project or *altera\<version>\quartus\bin* directory
 - Copy stored in local project directory read before original version in *bin*

Quartus II Settings File (QSF)

- Stores all settings & assignments *except* timing
- Uses Tcl syntax
- Can be edited manually by user

The screenshot shows a Windows-style text editor window titled "Text Editor - C:/altera_trn/Intro/pipemult - pipemult - [pipemult.qsf]*". The file contains the following Tcl code:

```
20
21 # Altera recommends that you do not modify this file. This
22 # file is updated automatically by the Quartus II software
23 # and any changes you make may be lost or overwritten.
24
25 source "location_assignments.tcl"
26
27 set_global_assignment -name DEVICE EP3C5F256C6
28 set_global_assignment -name FAMILY "Cyclone III"
29 set_global_assignment -name TOP_LEVEL_ENTITY pipemult
30 set_global_assignment -name ORIGINAL_QUARTUS_VERSION 6.1
31 set_global_assignment -name PROJECT_CREATION_TIME_DATE "22:56:"
32 set_global_assignment -name LAST_QUARTUS_VERSION 12.1
33 set_global_assignment -name DEVICE_FILTER_PACKAGE FBGA
34 set_global_assignment -name DEVICE_FILTER_PIN_COUNT 256
```

A blue callout box on the left says "User comments start with #". A green arrow points from this box to the first three lines of the file, which begin with "#". Another blue callout box on the left says "Source other Tcl/QSF files to organize assignments". A green arrow points from this box to the line "source \"location_assignments.tcl\"".

See “Quartus II Settings File Reference Manual”
for more details on QSF assignments & syntax

Constraint Files & Assignment Priority

1. QSF

- Highest priority
- Assignments always used from here first

2. Revision-specific QDF file located in project directory

- *<revision_name>_assignment_defaults.QDF*
- Created automatically in the project directory when a revision is opened in another version of the Quartus II software

3. QDF located in project directory

- *assignment_defaults.QDF*
- Created automatically in project directory when project archived & restored

4. QDF located in Quartus II *bin* directory

- Lowest priority
- Assignments only used if not found in higher priority files

Project Management

- Project archive & restore
- Project copy
- Revisions
- IP management tools
- Project clean

Project Archive

■ Creates 2 files

- Compressed Quartus II Archive File (.QAR)
 - Includes design files, QPF file, & QSF file(s)
 - Option to include databases
 - Creates local QDF file for archive
- Archive activity log (.QARLOG)

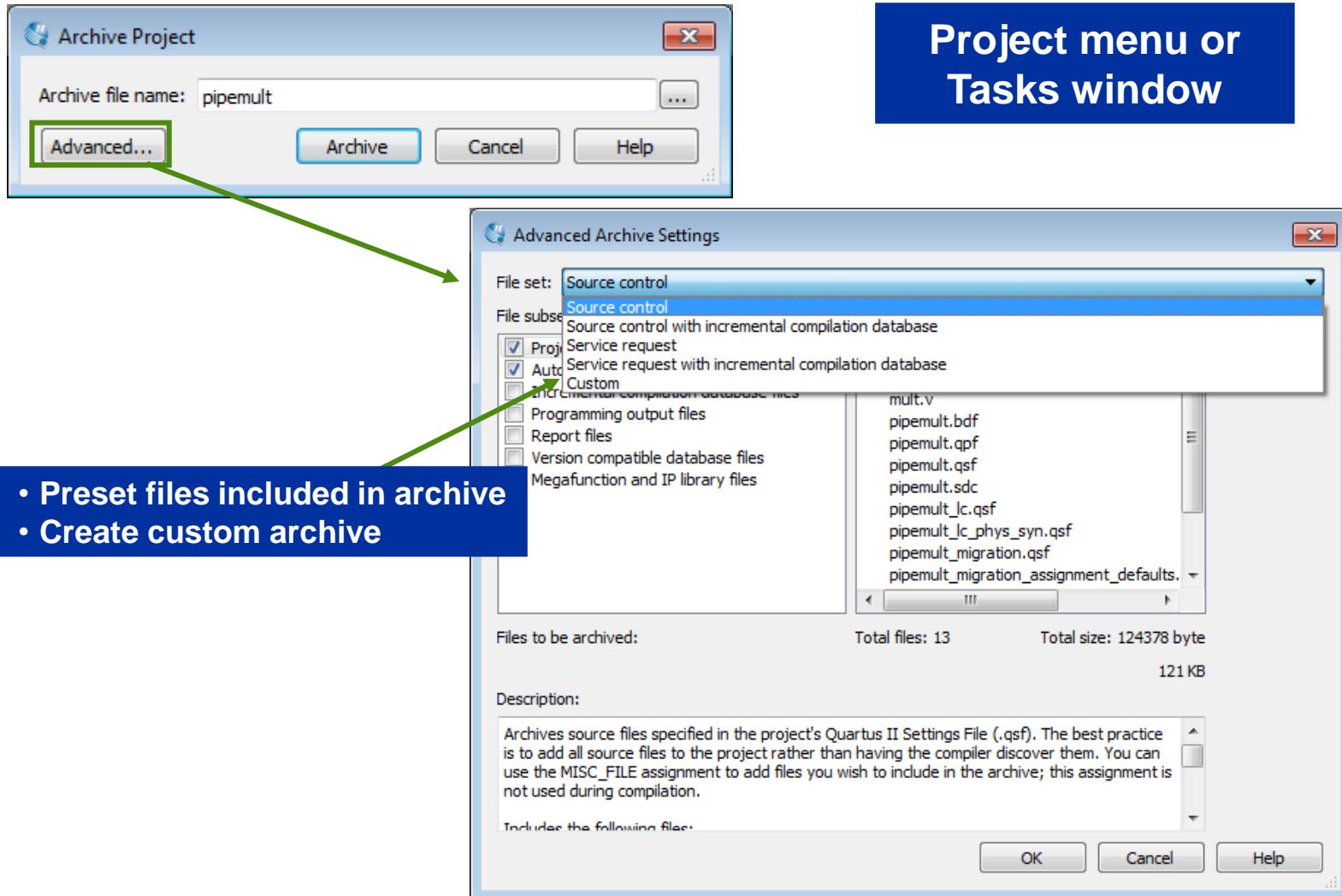
■ Example uses

- File storage (version control)
- Project handoff - Useful for sending to Altera support

■ Design files referenced from user libraries are included in archive

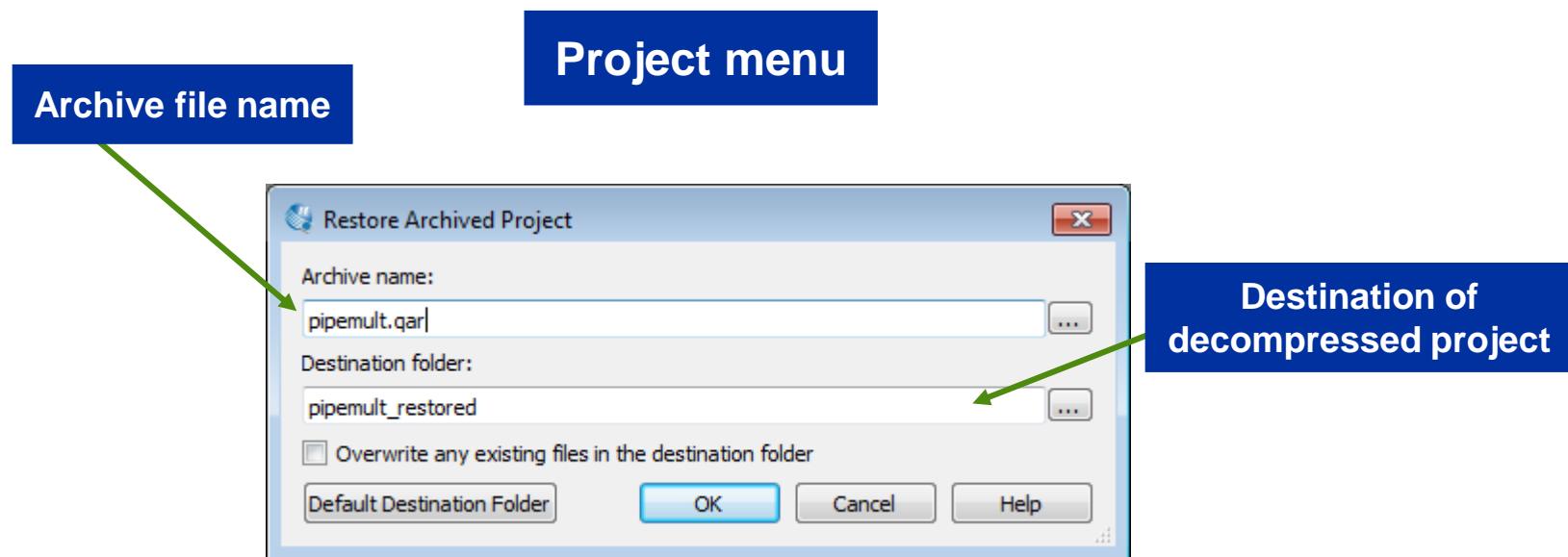
Tcl: project_archive <project_name>

Project Archive (cont.)



Restore Archived Project

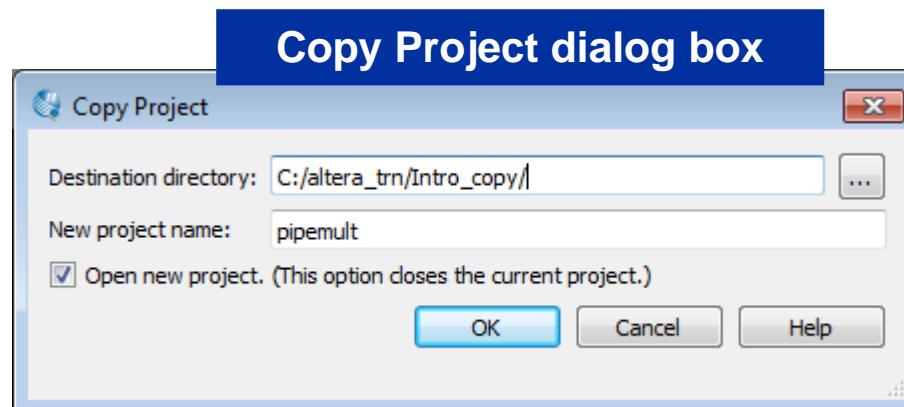
- Decompresses .QAR into specified directory



Tcl: project_restore <archive_file>

Project Copy

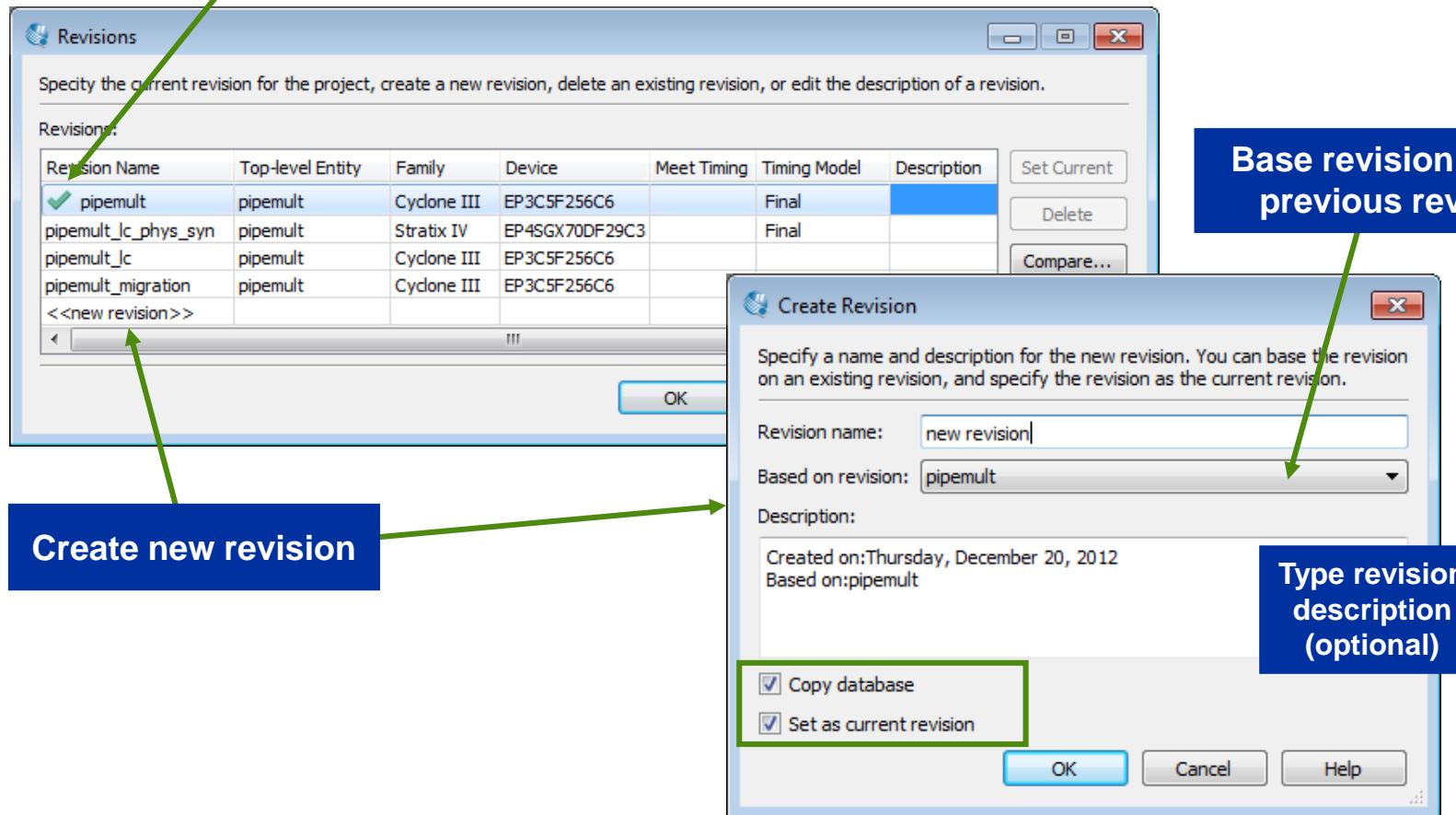
- **Copy & save exact duplicate of project in new directory**
 - Project file (.QPF)
 - Design files
 - Settings files
- **Example use - Duplicating work before editing design files**
- **User libraries are *not* copied; check paths**
- **New local QDF not created; only copies QDF if it exists**



Revisions

- **Explore new sets of constraints**
- **Compile options without losing previous work**
- **Compare results between revisions**
- **Revision-specific project files generated and stored in db directory**
 - Copy and update current revision files (no recompile required)
 - Generate new ones when new revision created and compiled

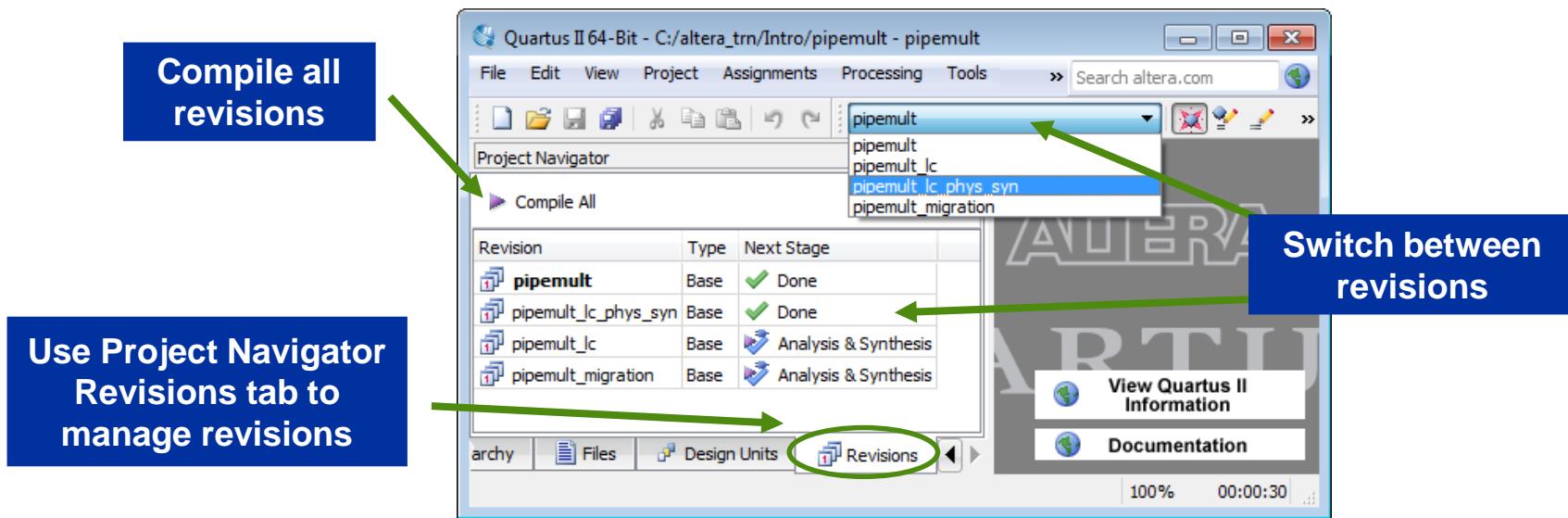
Creating a Revision



Tcl: *create_revision <revision_name>*

Project Revision Support

- QSF created for each revision
 - *<revision_name>.QSF*
- Active revision names stored in QPF
- Text file created for each revision (from description field)
 - *<revision_name>_description.TXT*



Tcl: *project_open -revision <revision_name> <project_name>*
Tcl: *set_current_revision <revision_name>*

Compare Revisions

- Click Compare... from the Revisions dialog box

Detailed summary of revision assignments and results

	pipemult_lc_phys_syn	pipemult	pipemult_lc	
Analysis & Synthesis	Successful - Mon May 13 19:39:31 2013 13.0.0 Build 156 04/24/2013 SJ Full Version	Successful - Mon May 13 19:38:31 2013 13.0.0 Build 156 04/24/2013 SJ Full Version	Successful - Mon May 13 19:39:04 2013 13.0.0 Build 156 04/24/2013 SJ Full Version	
Filter	pipemult_lc_phys_syn pipemult Cyclone IV E EP4CE6F17C6 Final	pipemult pipemult Cyclone IV E EP4CE6F17C6 Final	pipemult_lc pipemult Cyclone IV E EP4CE6F17C6 Final	
Timing Models	Total logic elements Total combinational functions Dedicated logic registers Total registers Total pins Total virtual pins Total memory bits Embedded Multiplier 9-bit elements Total PLLs	104 / 6,272 (2 %) 104 / 6,272 (2 %) 64 / 6,272 (1 %) 80 44 / 180 (24 %) 0 512 / 276,480 (< 1 %) 0 / 30 (0 %) 0 / 2 (0 %)	0 / 6,272 (0 %) 0 / 6,272 (0 %) 0 / 6,272 (0 %) 16 44 / 180 (24 %) 0 512 / 276,480 (< 1 %) 1 / 30 (3 %) 0 / 2 (0 %)	104 / 6,272 (2 %) 104 / 6,272 (2 %) 64 / 6,272 (1 %) 80 44 / 180 (24 %) 0 512 / 276,480 (< 1 %) 0 / 30 (0 %) 0 / 2 (0 %)
TimeQuest Timing Analyzer	Slow 1200mV 85C Model Setup 'clk1'	Slack TNS	3.182 0.000	4.433 0.000
	Slow 1200mV 85C Model Setup 'vir_clock'			3.182 0.000

Export...  Export to CSV file

Close Help

IP Management Tools

- Keep track of project IP cores
- Flag out-of-date IP cores
- Upgrade IP cores to latest version

Project Navigator – IP Components Tab

■ View all IP in project

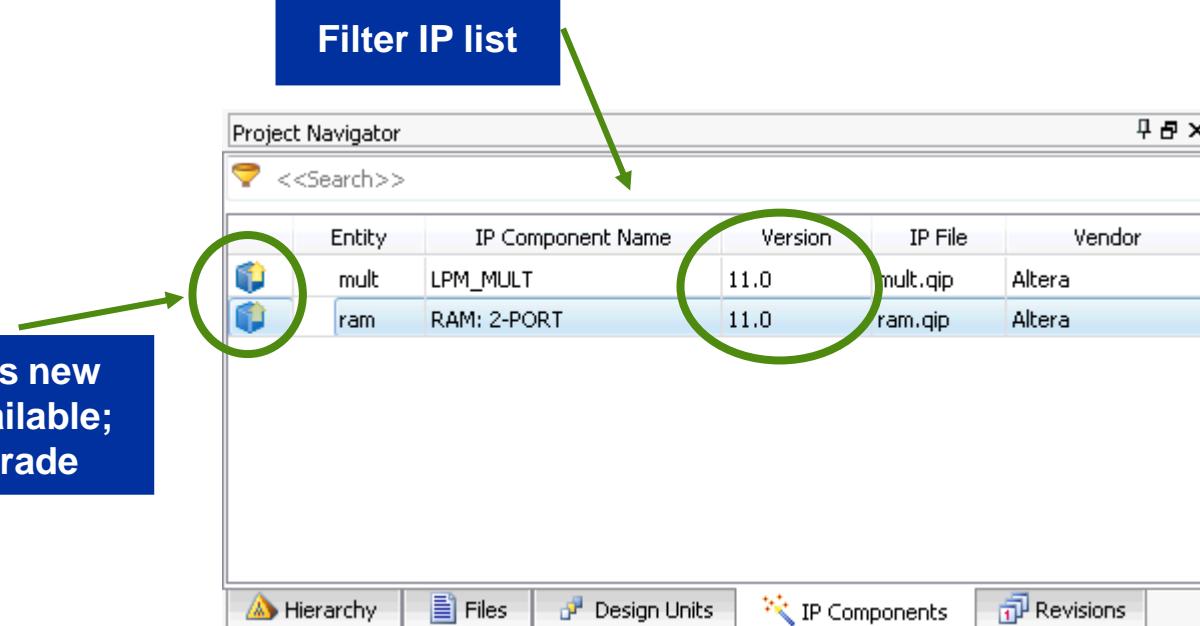
- Component name/file, Quartus II software version, vendor

■ View if new version of IP available

- Required upgrade  or optional upgrade 

■ Right-click to upgrade individual IP

Filter IP list

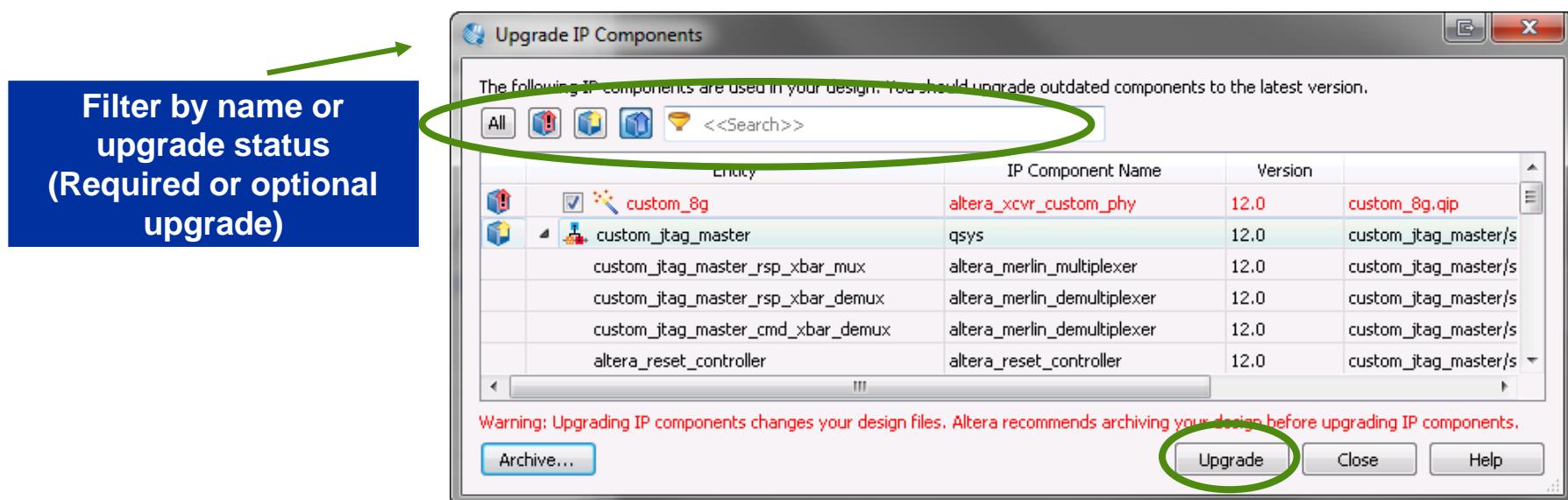


Entity	IP Component Name	Version	IP File	Vendor
mult	LPM_MULT	11.0	mult.qip	Altera
ram	RAM: 2-PORT	11.0	ram.qip	Altera

Icon indicates new version is available; click to upgrade

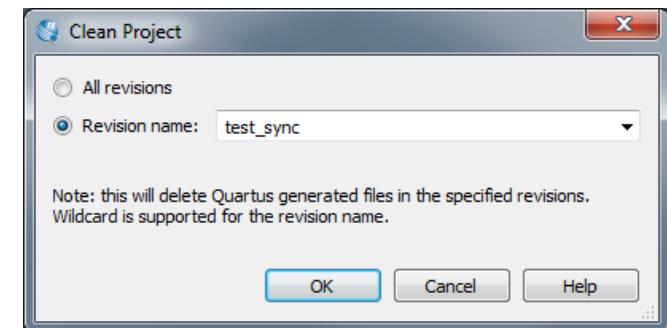
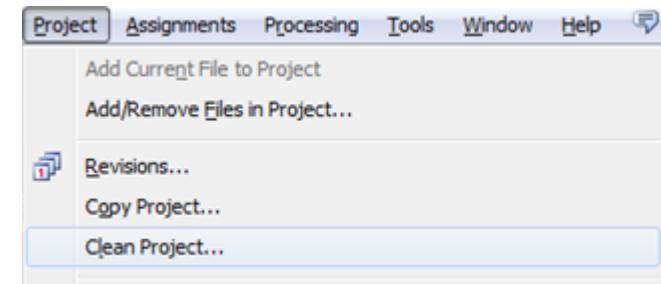
Upgrading IP Components

- Software detects out-of-date IP and displays dialog box
 - Manually open dialog box from Project menu
 - Displays required and optional upgrades
- Select IP block(s) and click Upgrade
 - Qsys components must be upgraded in Qsys



Project Clean

- **Cleans databases and output files generated by Quartus for a given revision (or revisions)**
 - Removes revision files in project subfolders
 - Removes reports, programming files, and other compiler generated output
- **Select Project Clean (Project menu)**
 - Select from available revisions
 - Clean multiple revisions using wildcards



```
Tcl: project_clean -revision <revision_name> <project_name>
```

Projects Summary

- Projects necessary for design processing
- Use New Project Wizard to create new projects
- Use Project Navigator to study file & entity relationships within project
- Project archive, copy, revisions, IP tools and clean provide easy-to-use project management

Class Agenda

- **Quartus II Projects**
- **Design Entry**
- **Quartus II Compilation**
- **Settings & Assignments**
- **I/O Planning**
- **Programming/Configuration**

Quartus II 软件设计: 基础

Design Entry



© 2013 Altera Corporation—Confidential

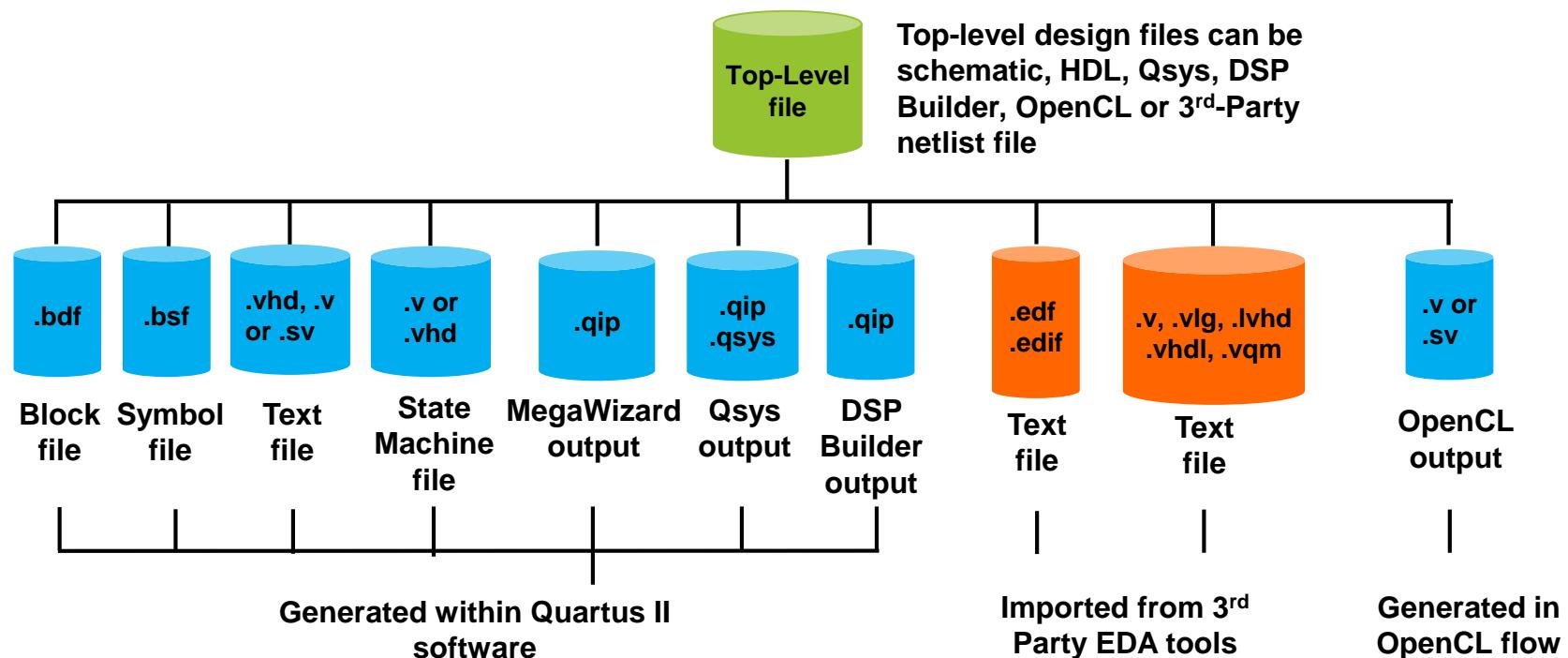


Design Entry Methods Supported

- **Text design entry**
- **Schematic design entry**
- **State Machine Editor**
- **Memory Editor**
- **3rd-party EDA tools**
- **System design entry**

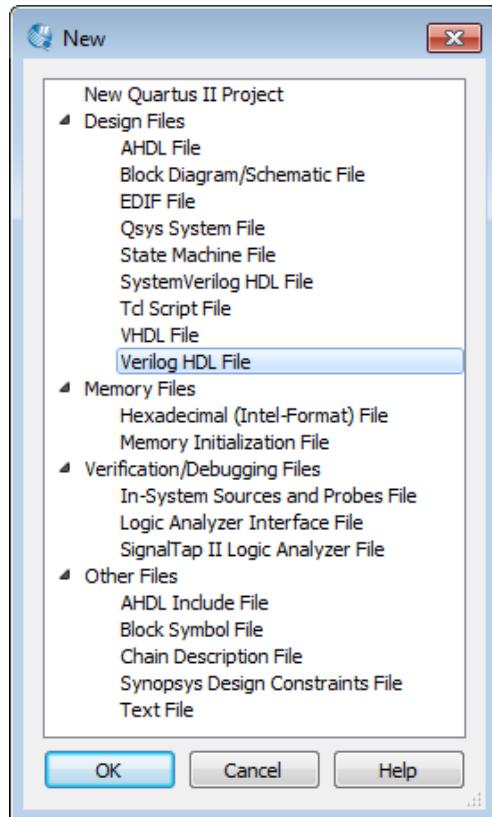
Design Entry File Types Supported

■ Mixing & matching design files allowed

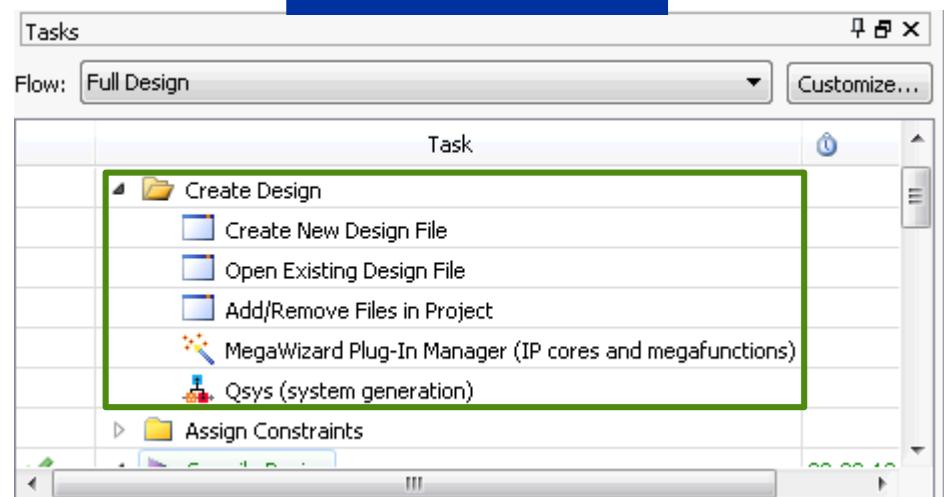


Creating New Design Files (& Others)

File → New or  in Toolbar



Tasks window



■ Quartus II Text Editor features

- Block commenting
- Line numbering in HDL text files
- Bookmarks
- Syntax coloring
- Find/replace text
- Find and highlight matching delimiters
- Function collapse/expand
- Create & edit .SDC files for TimeQuest timing analyzer
- Preview/editing of full design and construct HDL templates

■ Enter text description

- VHDL (.vhd, .vhdl)
- Verilog (.v, .vlg, .Verilog, .vh)
- SystemVerilog (.sv)

- **VHDL - VHSIC hardware description language**
 - IEEE Std 1076 (1987 & 1993) supported
 - Partial IEEE Std 1076-2008 support
 - IEEE Std 1076.3 (1997) synthesis packages supported
- **Verilog**
 - IEEE Std 1364 (1995 & 2001) & 1800 (SystemVerilog) supported
- **Use Quartus II integrated synthesis to synthesize**
- **View supported commands in built-in help**

*Learn more about HDL in Altera HDL
customer training classes*

Text Editor Features

The screenshot shows the Altera Text Editor interface with several features highlighted:

- Find/highlight matching delimiters**: Points to the magnifying glass icon in the toolbar.
- Bookmarks (on/off/jump to)**: Points to the bookmark icon in the toolbar.
- Insert Template (Edit menu)**: Points to the "Edit" menu.
- Collapse/expand functions**: Points to the collapse/expand icon in the toolbar.
- Preview window: edit before inserting & save as user template**: Points to the "Insert Template" dialog box, which displays a preview of the Verilog code for a single-port RAM module.

```
// Quartus II Verilog Template
// Single port RAM with single read/write add
module single_port_ram
#(parameter DATA_WIDTH=8, parameter ADDR_WID=1
(
    input [(DATA_WIDTH-1):0] data,
    input [(ADDR_WIDTH-1):0] addr,
    input we, clk,
    output [(DATA_WID-1):0] q
);
reg [DATA_WIDTH-1:0] ram;
reg [ADDR_WIDTH-1:0] addr_reg;
begin
    // Write
    if (we)
        ram[addr] <= data;
    addr_reg <= addr;
end
endmodule
```

Quartus II Text Editor Features

- Status bar enhancement
- Auto-complete
- Smart highlighting
- Syntax color differentiation
- Error message indicator

```
BEGIN
    rx_locked      <= sub_wire0;
    rx_out        <= sub_wire1(7 DOWNTO 0);
    rx_outclock   <= sub_wire2;

    rx
    rx_align_data_reg : ALTLVDS_RX
    rx_in
    rx_inclock
    rx_locked
    rx_out
    common_rx_tx_pll => "OFF",
    data_align_rollover => 4,
    data_rate => "UNUSED",
    deserialization_factor => 4,
    dpa_initial_phase_value => 0,
```

```
input  clock_ena;
input  signed [WIDTH-1:0] a0;
input  signed [WIDTH-1:0] b0;
input  signed [WIDTH-1:0] a1;
input  signed [WIDTH-1:0] b1;
output signed [2*WIDTH-1:0] rout;
output signed [2*WIDTH-1:0] iout;

reg  signed [WIDTH-1:0] a0_reg;
reg  signed [WIDTH-1:0] b0_reg;
reg  signed [WIDTH-1:0] a1_reg;
reg  signed [WIDTH-1:0] b1_reg;
reg  signed [WIDTH-1:0] a2_reg;
reg  signed [WIDTH-1:0] b2_reg;
reg  signed [WIDTH-1:0] a3_reg;
reg  signed [WIDTH-1:0] b3_reg;

reg  signed [2*WIDTH-1:0] rout;
reg  signed [2*WIDTH-1:0] iout;
```

```
1 //18 bit complex input, 36 bit real and imaginary outputs
2 // (a0+j*b0)*(a1+j*b1) = (a0*a1-b0*b1)+j*(a0*a1+b0*b1)
3 module half_dsp_block (clock, areset, clock_ena, a0, b0, a1, b1, rout, iout)
4     parameter WIDTH = 18;
5     int Error(10161); Verilog HDL error at half_dsp_block.v(3); object "clock" is not declared
6     input areset;
7     input clock_ena;
8     input signed [WIDTH-1:0] a0;
9     input signed [WIDTH-1:0] b0;
```

Schematic Design Entry

- **Full-featured schematic design capability**
- **Schematic Editor uses**
 - Create simple test designs to understand the functionality of an Altera megafunction - PLL, LVDS I/O, memory, etc...
 - Create top-level schematic for easy viewing & connection
 - Conversion utilities

Note: Schematic entry online training available: [Using the Quartus II Software: Schematic Design](#)

■ Pre-made design blocks

■ Benefits

- Configurable, parameterized settings add flexibility & portability
- “Drop-in” support to accelerate design entry
- Pre-optimized for Altera architecture

■ Two versions

- Quartus II megafunctions
- Intellectual Property (IP) megafunctions

Quartus II Megafunctions

■ Free & installed with Quartus II software

- Non-encrypted functions written in AHDL (Altera HDL)
- HDL simulation models installed in Quartus II libraries

■ Two types

- Altera-specific megafunctions (begin with “ALT”)
- Library of parameterized modules (LPMs)

■ Examples

- Arithmetic
- On-chip RAM/ROM
- PLLs
- DDR/QDR/RLDRAM memory controllers

IP Megafunctions

- **Must purchase license (except IP base suite)**
 - Logic for IP function is encrypted
- **Two types**
 - MegaCore IP – Developed by Altera
 - Altera Megafunctions Partner Program (AMPP) IP
- **All MegaCore functions & some AMPP functions support OpenCore® Plus feature**
 - Develop design using free version of core
 - HDL simulation models provided with IP
 - Generate time-limited configuration/programming files
 - See [AN320: OpenCore Plus Evaluation of Megafunctions](#)

MegaCore IP Examples

■ Included in IP base suite

- FIR Compiler
- Fast Fourier Transform Compiler
- DDR3 SDRAM Controller with UniPHY

■ License required

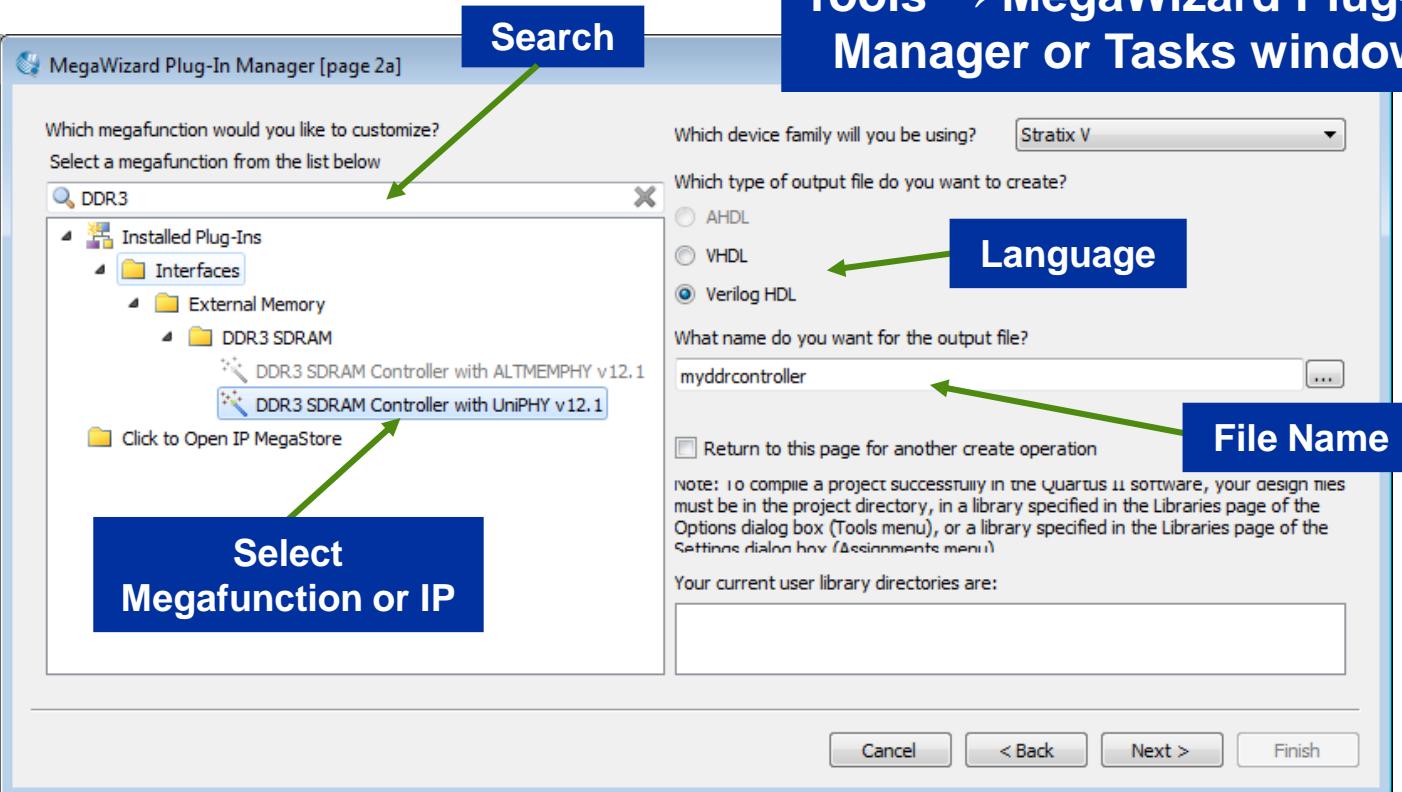
- Triple-Speed Ethernet MAC
- 10Gb Ethernet MAC
- CRC Compiler
- IP Compiler for PCI Express® soft core
 - Note: Hard IP for PCI Express does not required a license
- Video and Image Processing Suite

See <http://www.altera.com/products/ip/ipm-index.html> for a complete list of Altera IP solutions

MegaWizard Plug-in Manager

- Eases implementation and configuration of megafunctions & IP
- GUI, command line, or both

Tools → MegaWizard Plug-In Manager or Tasks window



MegaWizard Plug-In Manager [page 2a]

Search

Which megafunction would you like to customize?
Select a megafunction from the list below

Installed Plug-Ins

- DDR3
- Interfaces
- External Memory
- DDR3 SDRAM
 - DDR3 SDRAM Controller with ALTMEMPHY v12.1
 - DDR3 SDRAM Controller with UniPHY v12.1
- Click to Open IP MegaStore

Which device family will you be using? Stratix V

Which type of output file do you want to create?

AHDL

VHDL

Verilog HDL

Language

What name do you want for the output file?

myddrcontroller

File Name

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 a library specified in the Libraries page of the Options dialog box (Tools menu), or a library specified in the Libraries page of the Settings dialog box (Assignments menu).

Your current user library directories are:

Cancel < Back Next > Finish

Command line: qmegawiz <-silent> <module / wizard>=<mf_name> <ports & parameters options> file_name

MegaWizard Plug-In Example

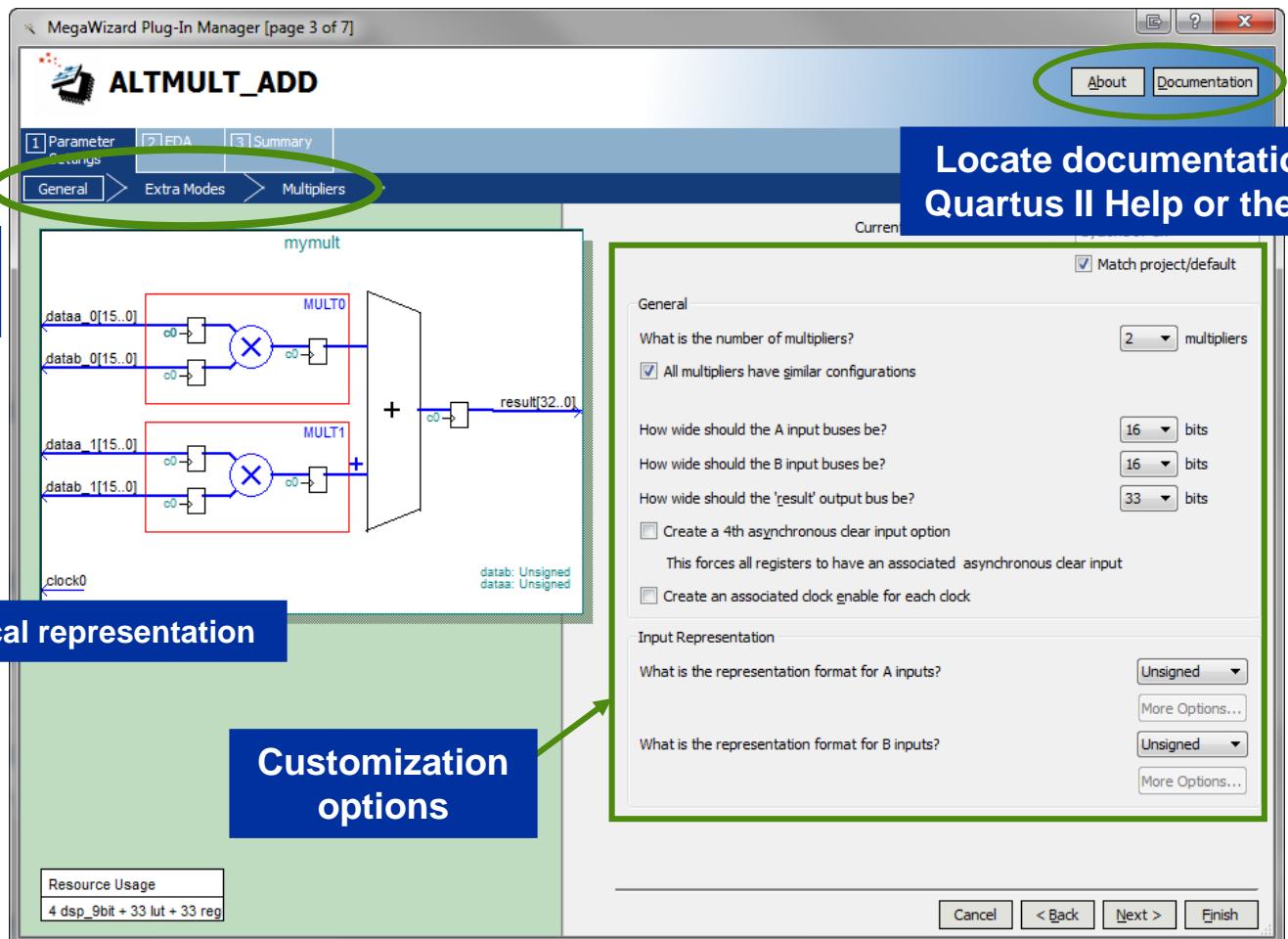
Multiply-Add Megafunction

Three step process to configure megafunction

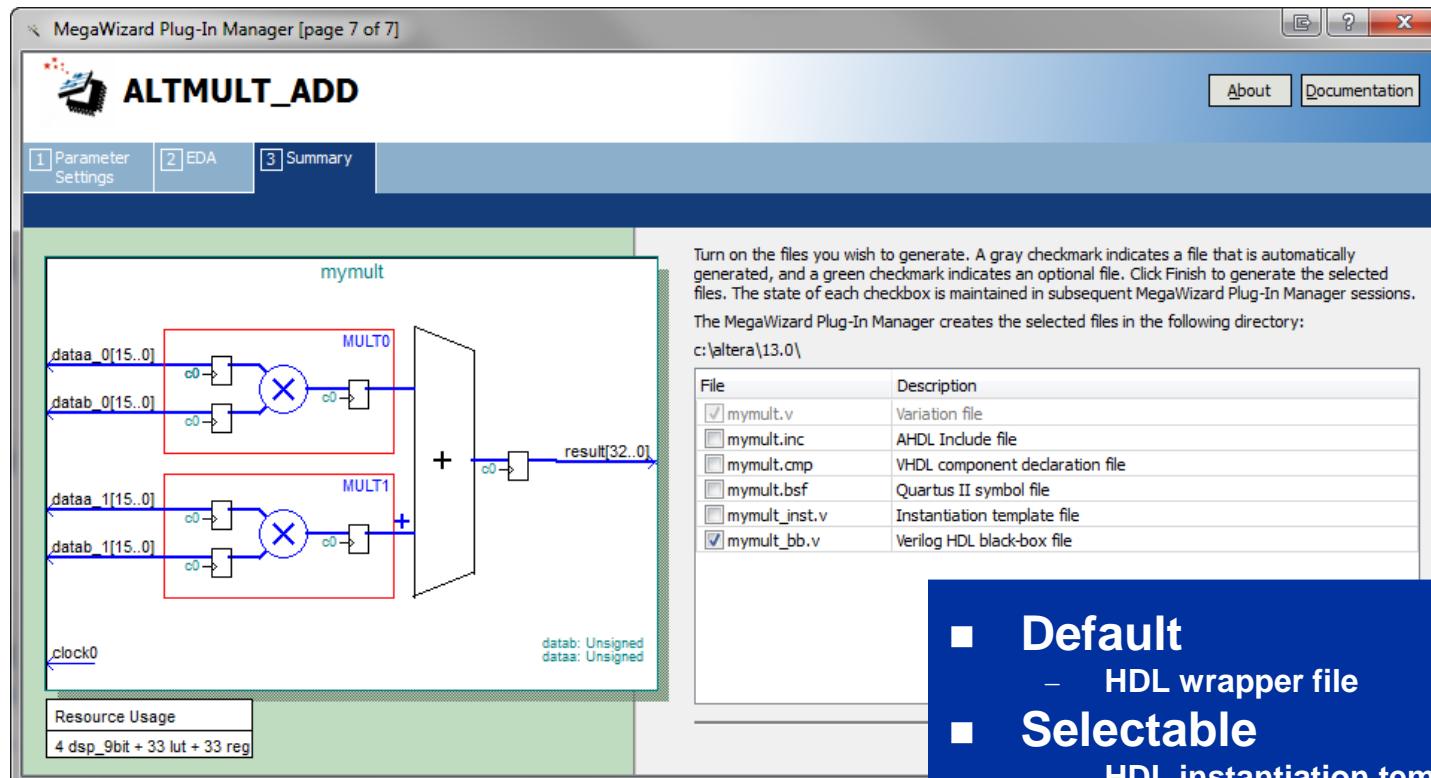
Locate documentation in Quartus II Help or the web

Updating graphical representation

Customization options



MegaWizard Plug-In Output File Selection



- **Default**
 - HDL wrapper file
- **Selectable**
 - HDL instantiation template
 - VHDL component declaration (.cmp)
 - Quartus II symbol (.bsf)
 - Verilog black box
 - Behavioral waveform (.html)

New Megafunction/IP Parameter Editor

The screenshot shows the Cyclone V Transceiver Native PHY - my_xcvr parameter editor window. On the left is a Block Diagram titled "my_xcvr" showing various signal connections. In the center is a Parameter Options panel with sections for General, Datapath Options, PMA, and Standard PCS. On the right is a Presets panel showing available project libraries and a "Select Available Presets" button. A "Create New Presets" button is also visible. At the bottom, a Messages panel displays several informational messages about the PHY's configuration.

Block Diagram with Interfaces

Parameter Options

Select Available Presets

Create New Presets

Messages

- Info: my_xcvr: PHY IP will require 2 reconfiguration interfaces for connection to the device.
- Info: my_xcvr: Reconfiguration interface offset 0 is connected to the transceiver chip.
- Info: my_xcvr: Reconfiguration interface offset 1 is connected to the transmit PLL.
- Info: my_xcvr: tx_parallel_data: For each 44 bit word the 10 active data bits are tx_parallel_data[9:0];
- Info: my_xcvr: rx_parallel_data: For each 64 bit word the 10 active data bits are rx_parallel_data[9:0];

State Machine Editor

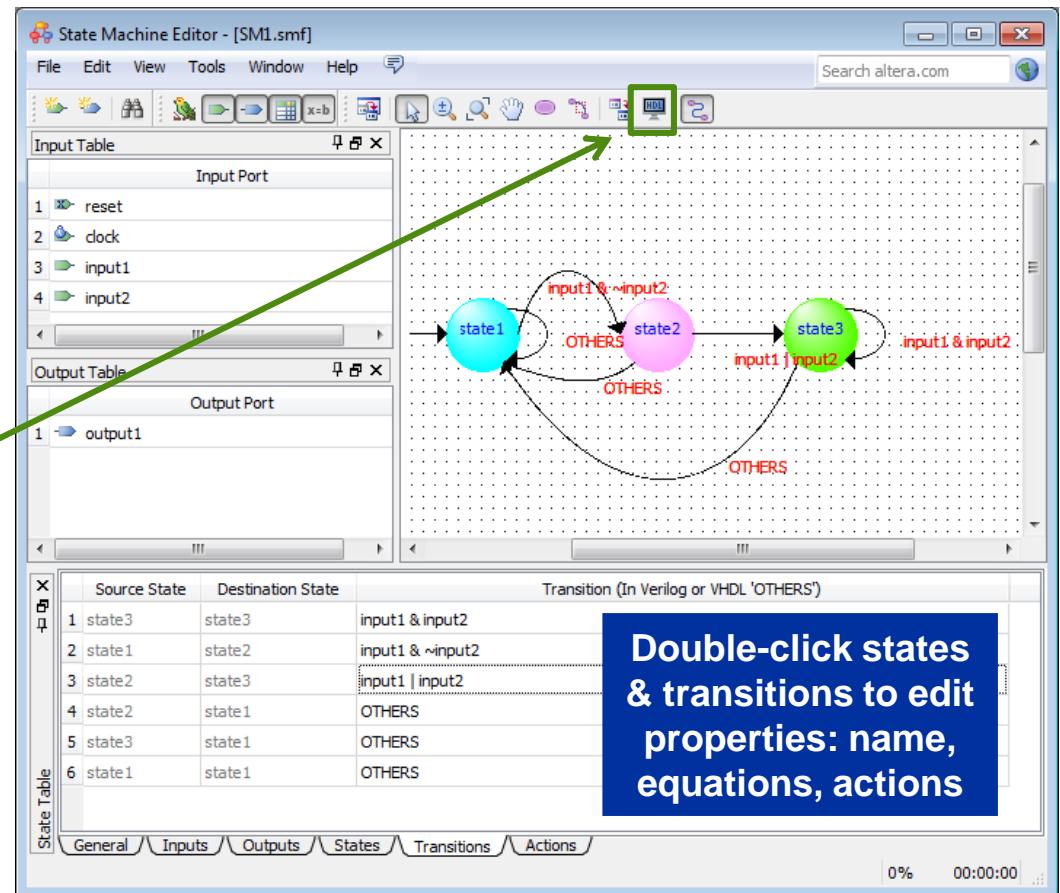
■ Create state machines in GUI

- Manually by adding individual states, transitions, and output actions
- Automatically with State Machine Wizard (Tools menu)

■ Generate state machine HDL code (required)

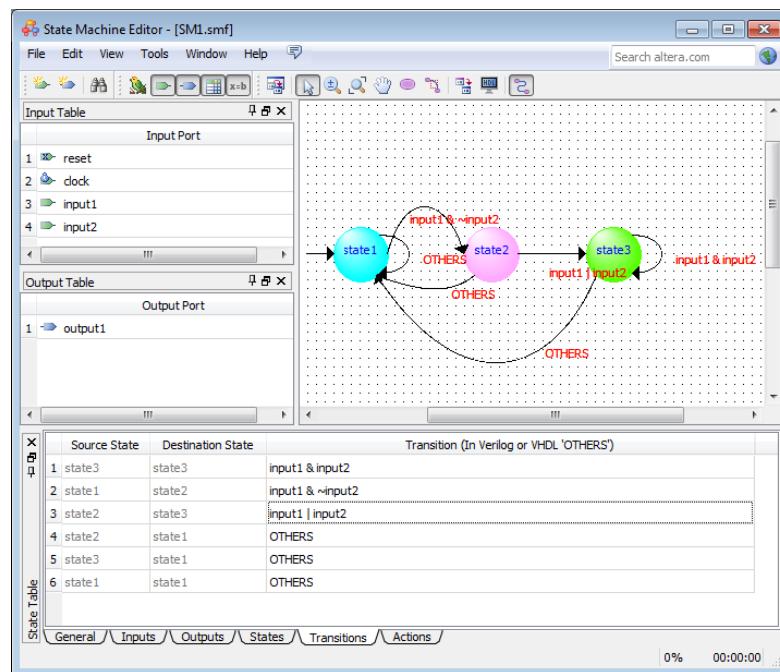
- VHDL
- Verilog
- SystemVerilog

File ⇒ New or Tasks window
Select **State Machine File (.SMF)**



From .SMF to HDL

- Generate optimized code (Verilog, SV, or VHDL)
- Automatically added to project
- Required for use



```
module SM1 (
    input reset, input clock, input input1, input input2,
    output output1);
    reg reg_output1;
    enum int unsigned { state1=0, state2=1, state3=2 }

    always_ff @(posedge clock)
    begin
        if (clock) begin
            fstate <= reg_fstate;
        end
    end

    always_comb begin
        if (reset) begin
            reg_fstate <= state1;
            reg_output1 <= 1'b0;
            output1 <= 1'b0;
        end
        else begin
            reg_output1 <= 1'b0;
            output1 <= 1'b0;
            case (fstate)
                state1: begin
                    if ((input1 & ~input2))
                        reg_fstate <= state2;
                    else
                        reg_fstate <= state1;

                    reg_output1 <= 1'b0;
                end
                state2: begin
                    if ((input1 | input2))
                        reg_fstate <= state3;
                    else
                        reg_fstate <= state1;
                end
            endcase
        end
    end
endmodule
```

■ Create or edit memory initialization files

- Intel HEX (.HEX)
 - Preferred and compatible with 3rd party tools
- Altera-specific (.MIF) format

■ Design entry

- Initialization file data sent to device during device programming to initialize memory blocks

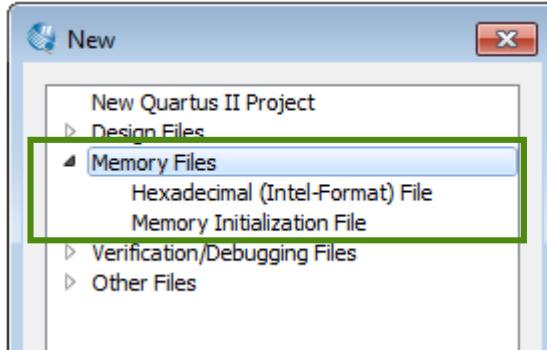
■ Simulation

- Use to initialize memory blocks before simulation or after breakpoints

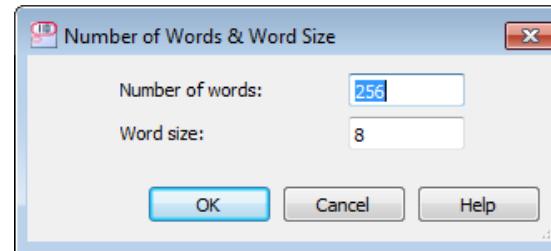
Create Memory Initialization File

File → New or Tasks window

1) HEX or MIF format



2) Select memory size



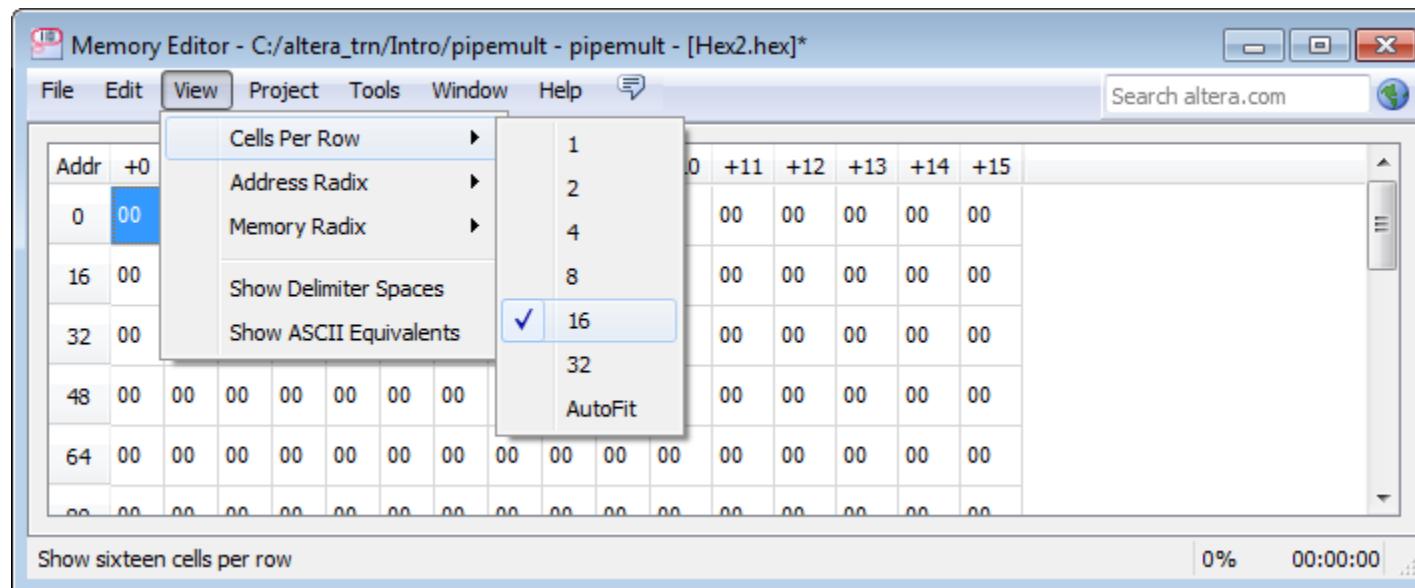
3) Memory space editor opens

A screenshot of the Quartus II Memory Space Editor. The window title is 'Hex1.hex*'. The table displays memory addresses from 0 to 48 in increments of 8, with data bytes in columns labeled +0 to +7. The data values are: Row 0: AB FF DF FF 01 00 02 00; Row 8: 03 00 04 00 00 00 00 00; Row 16: 00 67 00 00 00 05 05 00; Row 24: 00 00 00 00 00 06 00 00; Row 32: 00 00 08 00 00 00 00 00; Row 40: 00 00 00 00 00 00 00 00; Row 48: 00 00 00 00 00 00 00 00. The cell at address 48, column +6 is highlighted with a blue border.

Change Options

■ View options of memory editor

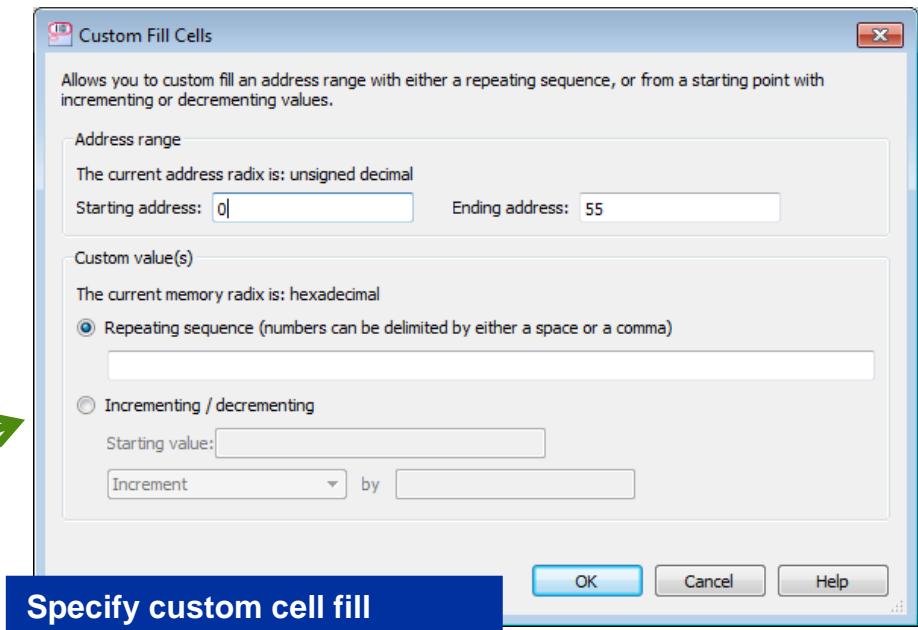
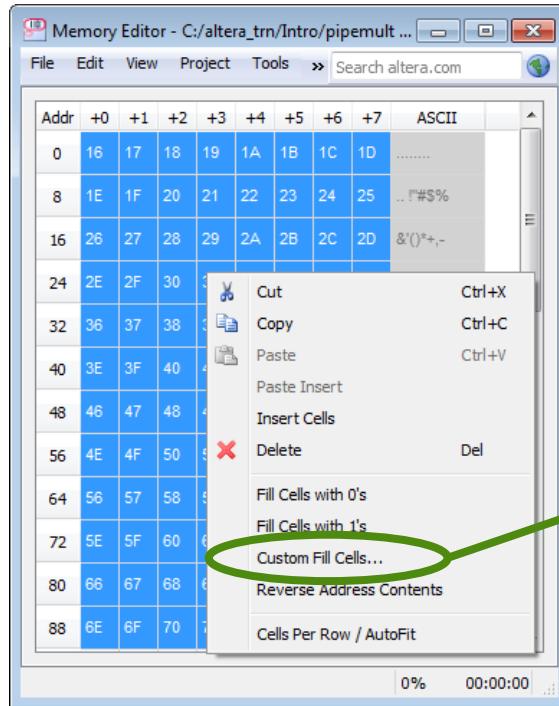
- View menu ⇒ select from available options



Edit Contents

- Edit contents of memory file
- Save memory file as .HEX or .MIF file

- Select address location & type in a value
- Select the address & right-click to select fill option from menu
- Copy & paste from spreadsheet



Specify custom cell fill
• Repeating sequence
• Increment/decrement count

Using Memory File in Design

MegaWizard Plug-In Manager [page 8 of 10]

RAM: 2-PORT

1 Parameter Settings 2 EDA 3 Summary

General > Widths/Blk Type > Clks/Rd, Byte En > Regs/Ckens/Adrs > Output1 > Mem Init >

Do you want to specify the initial content of the memory?

No, leave it blank
 Initialize memory content data to XX..X on power-up in simulation

Yes, use this file for the memory content data
(You can use a Hexadecimal (Intel-format) File [.hex] or a Memory Initialization File [.mif])

Specify MIF or HEX file in MegaWizard Plug-In

File name: ram.hex

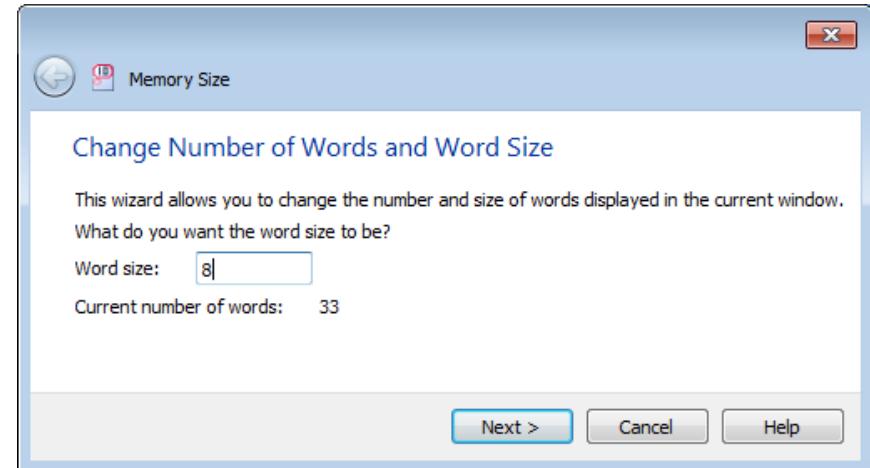
The initial content file should conform to which port's dimensions? **PORT_B**

Browse...

May also specify MIF file in HDL using the `ram_init_file` synthesis attribute

Memory Size Wizard

- Allows editing the size of memory file
- Use the Memory Size Wizard (Edit menu)
 - Edit word size
 - Edit number of words
 - Specify how to handle word size change
 - Pad words
 - Combine words
 - Truncate words
 - Add words



Importing 3rd-Party EDA Tool Files

- **Interface with industry-standard EDA tools that generate netlist files**
 - EDIF 2 0 0 (.EDF)
 - Verilog Quartus Mapping (.VQM)
- **To import and use netlist files**
 - Specify EDA tool in the Quartus II software settings
 - Instantiate block(s) in design
 - Add .EDF/.VQM file(s) to Quartus II project

3rd-Party Synthesis Tool Support



- LeonardoSpectrum™
- Precision RTL



- Design Compiler
- Synplify

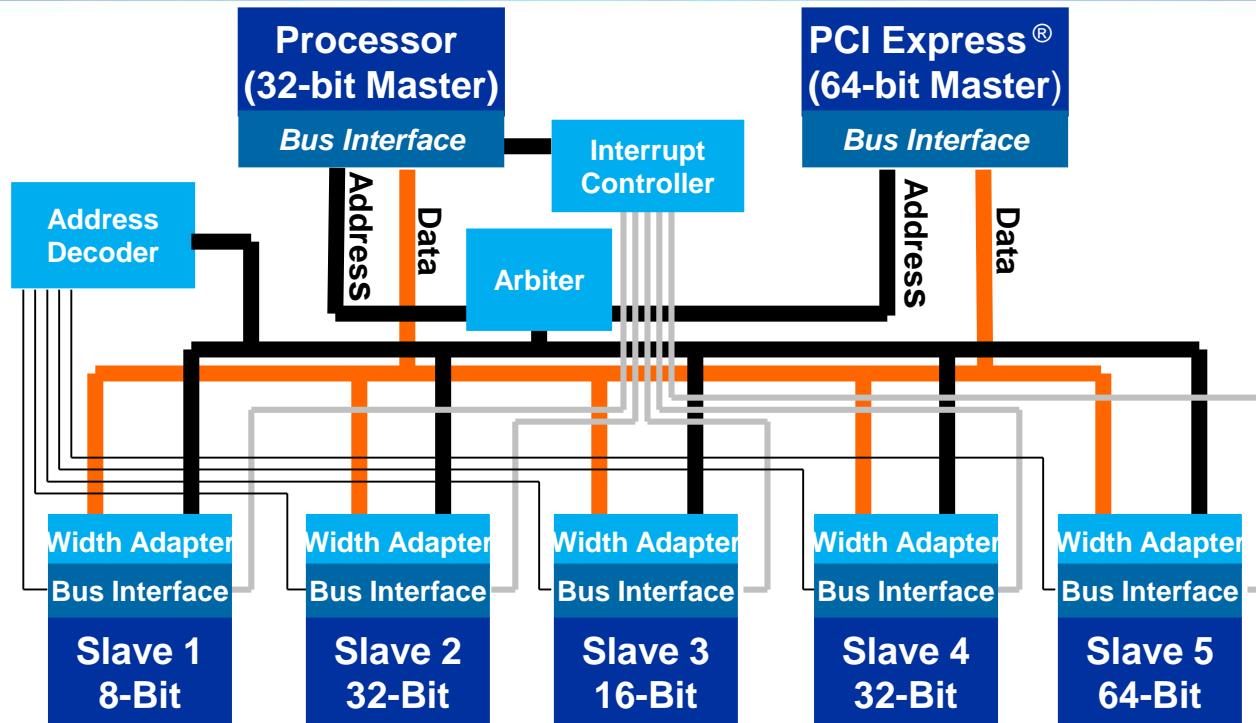
System Design Entry

- **Simplifies complex system development**
- **Raises the level of design abstraction**
 - High level design and system visualization
- **Reduces time to market**
 - Reduces design development time
 - Eases verification

System Design Entry Tools

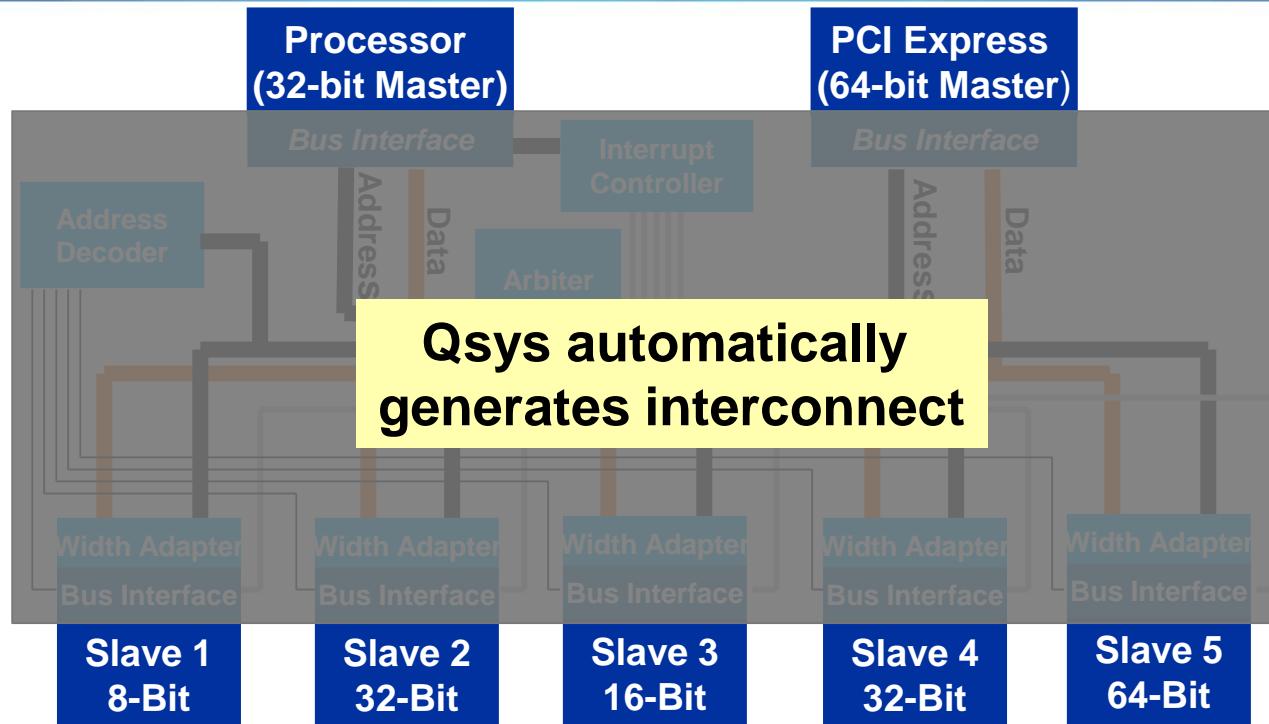
- **Qsys system development tool**
- **DSP Builder Standard/Advanced Blocksets**
- **Altera SDK for OpenCL™ Compiler**

Traditional System Design



- Components in system use different interfaces to communicate (some standard, some non-standard)
- Typical system requires significant engineering work to design custom interface logic
- Integrating design blocks and intellectual property (IP) is tedious and error-prone

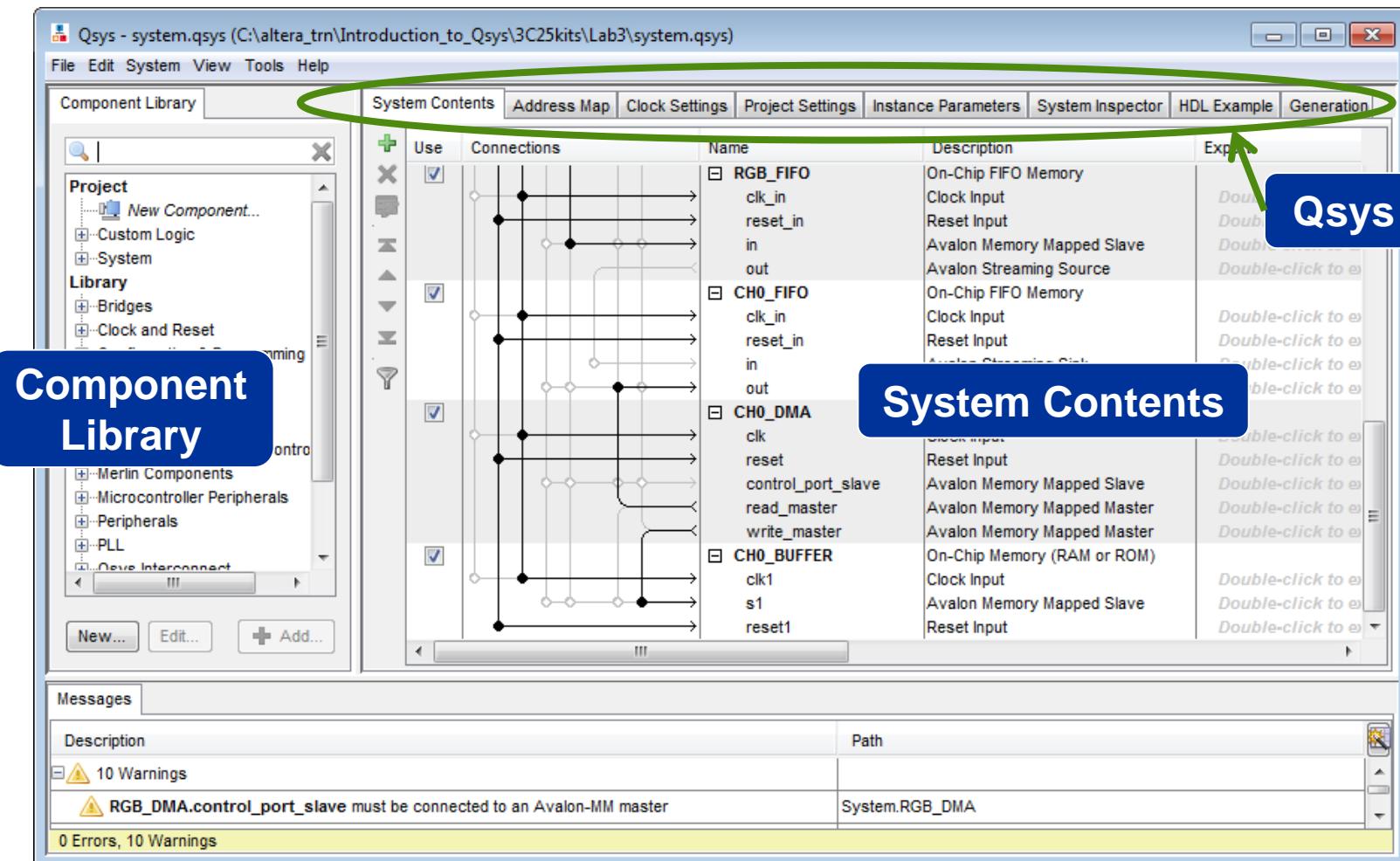
Qsys Automatic Interconnect Generation



- **Avoids error-prone integration**
- **Saves development time with automatic logic & HDL generation**
- **Enables you to focus on value-add blocks**

Qsys improves productivity by automatically generating the system interconnect logic

Qsys System-Integration Tool

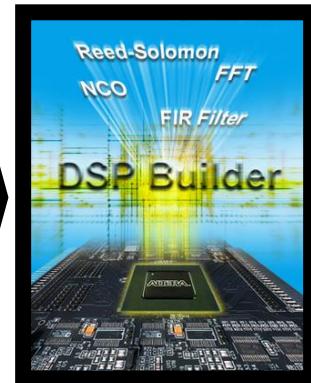
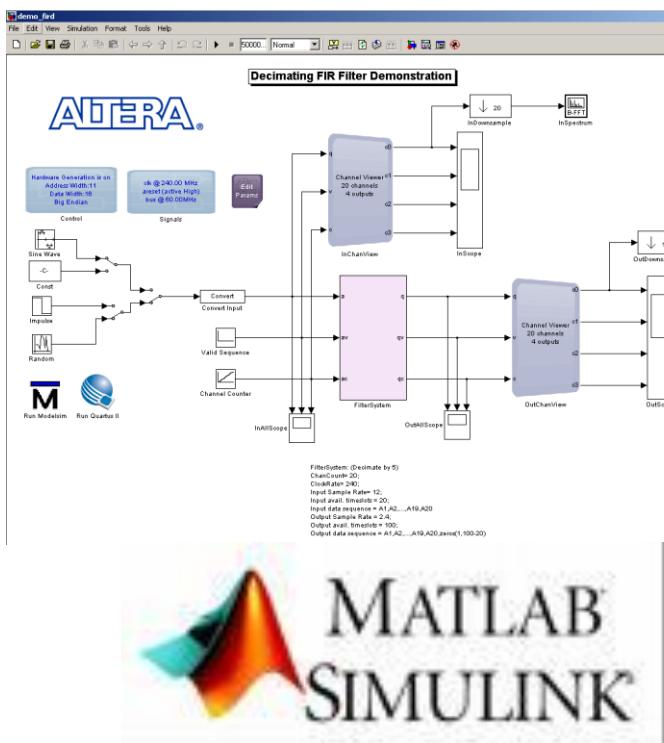


Component
Library

System Contents

Qsys tabs

DSP Builder (DSPB) Standard/Advanced Blocksets



HDL Automatically Optimized for System Clock Frequency & latency

ALTERA[®]
MEASURABLE ADVANTAGE™

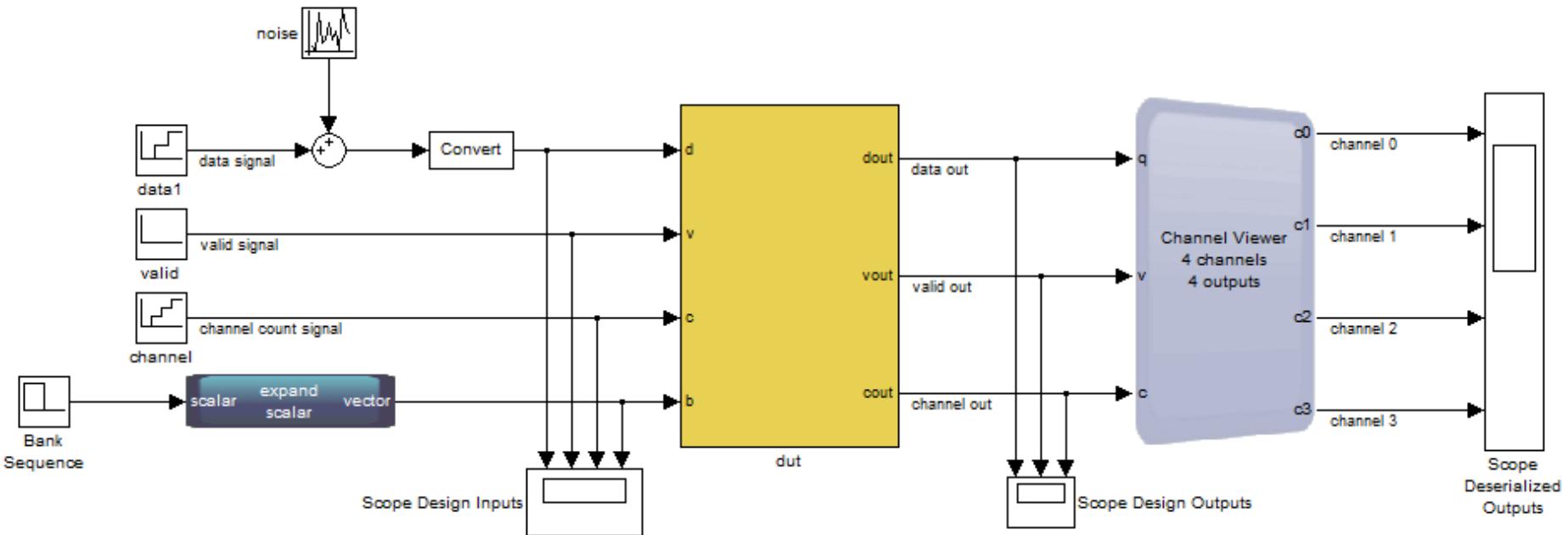
■ DSP Builder Standard Blockset

- Cycle accurate behavior (works the way you designed it)
- Multiple clock domain design
- Control rich with state machine or backpressure
- Access detailed device features
- Custom HDL code import support
- Hardware-in-the-loop simulation support

■ DSP Builder Advanced Blockset

- Specification driven design with automatic pipeline and folding
- Multichannel designs with automatic vectorized inputs
- Single system clock for the main datapath logic
- Design exploration

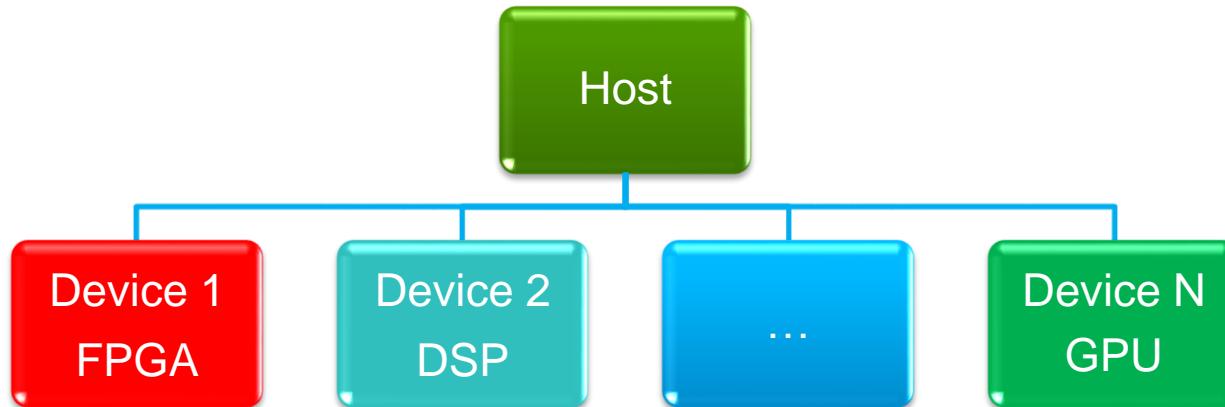
DSP Builder Advanced Blockset Example



- Top-level of a DSPB design is a testbench

Open Computing Language (OpenCL)

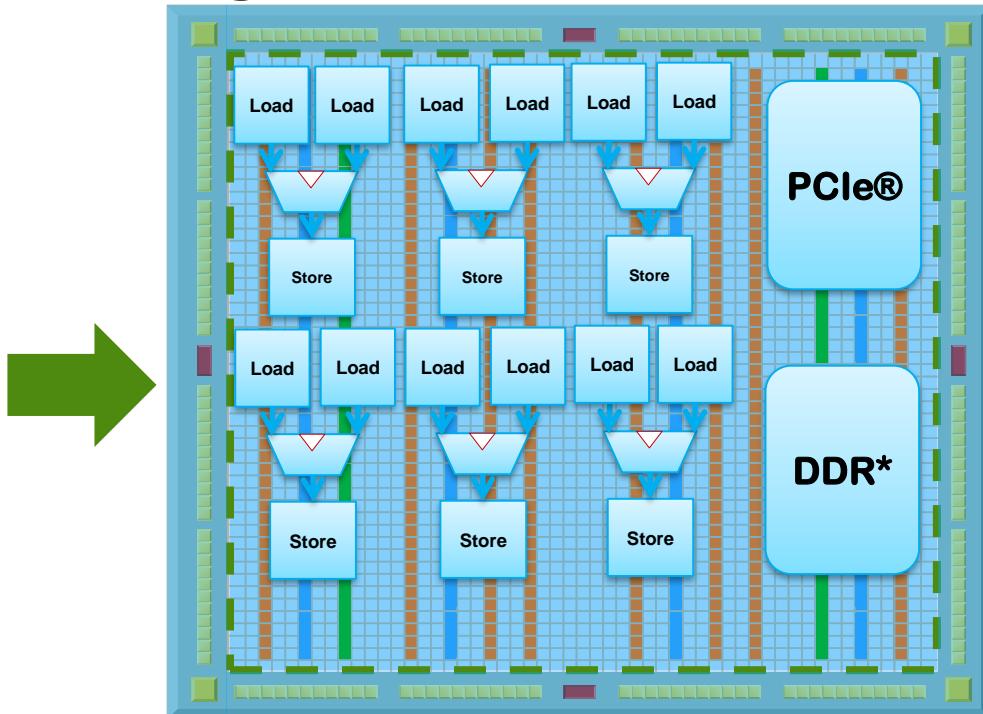
- Framework for heterogeneous programs
- Royalty-free
- Allows general purpose programs to run on multiple platforms
- Adaptable to obtain high performance



Altera SDK for OpenCL

- Translates OpenCL source file into custom-generated hardware image to be loaded onto FPGA
- Provides API for use by OpenCL host application to communicate with hardware image

```
// kernel.cl
__kernel void KernelName(...)
{
    int i = get_global_id(0);
    c[i] = a[i] + b[i];
}
```



Design Entry Summary

- **Multiple design entry methods supported**
 - Text (Verilog, VHDL, State Machine Editor output)
 - Schematic
 - 3rd-party netlist (VQM, EDIF)
 - System (Qsys, DSPB & Altera SDK for OpenCL)
- **MegaWizard™ Plug-In Manager parameterizes megafunctions & IP**
- **Memory Editor allows generation of memory initialization files**
- **3rd-party EDA tools supported for design entry & synthesis**
- **System design tools such as Qsys, DSP Builder and Altera SDK for OpenCL simplify complex design development**

Class Agenda

- **Quartus II Projects**
- **Design Entry**
- **Quartus II Compilation**
- **Settings & Assignments**
- **I/O Planning**
- **Programming/Configuration**

Quartus II软件设计:基础

Quartus II Compilation



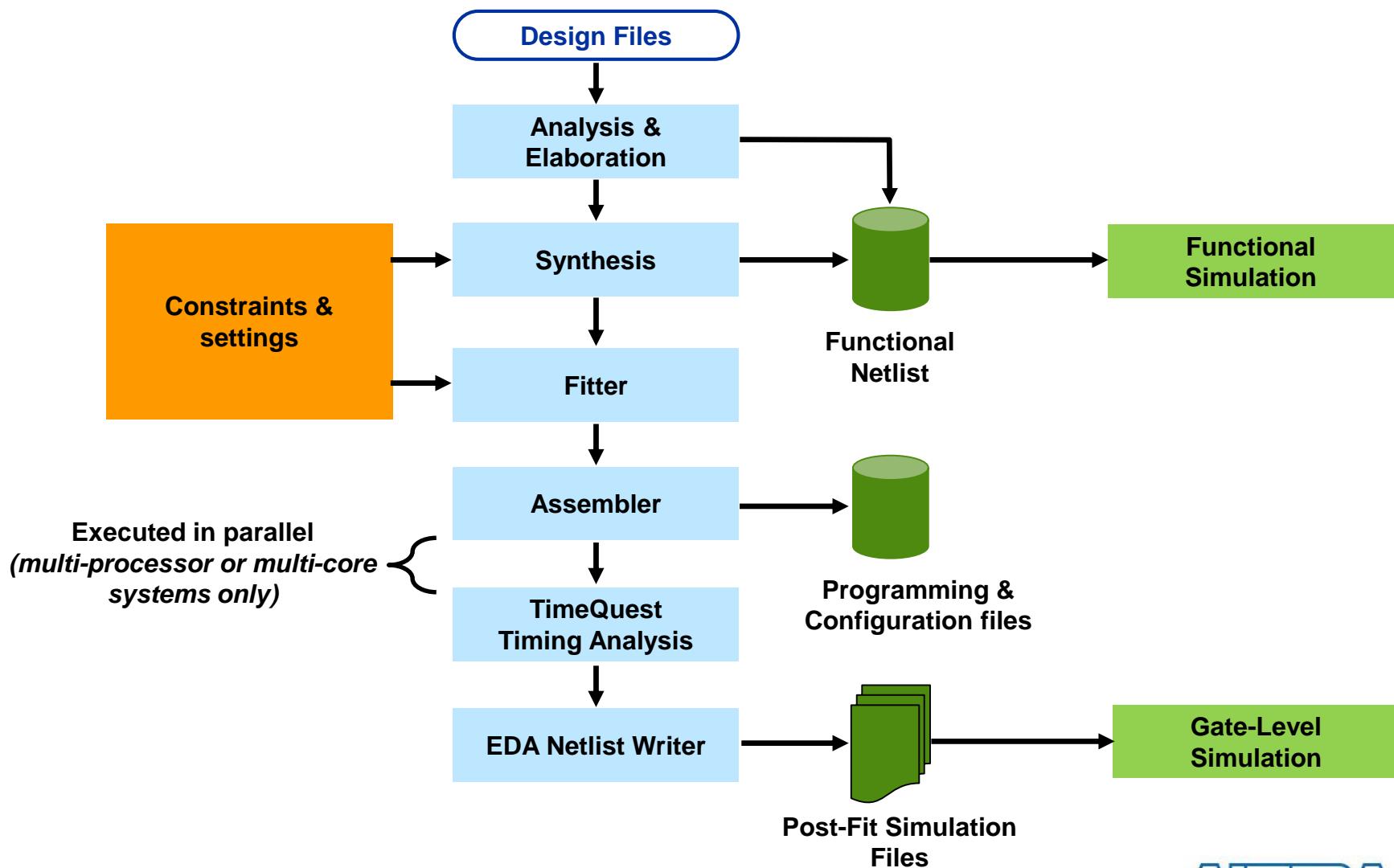
© 2013 Altera Corporation—Confidential



Compilation Section Objectives

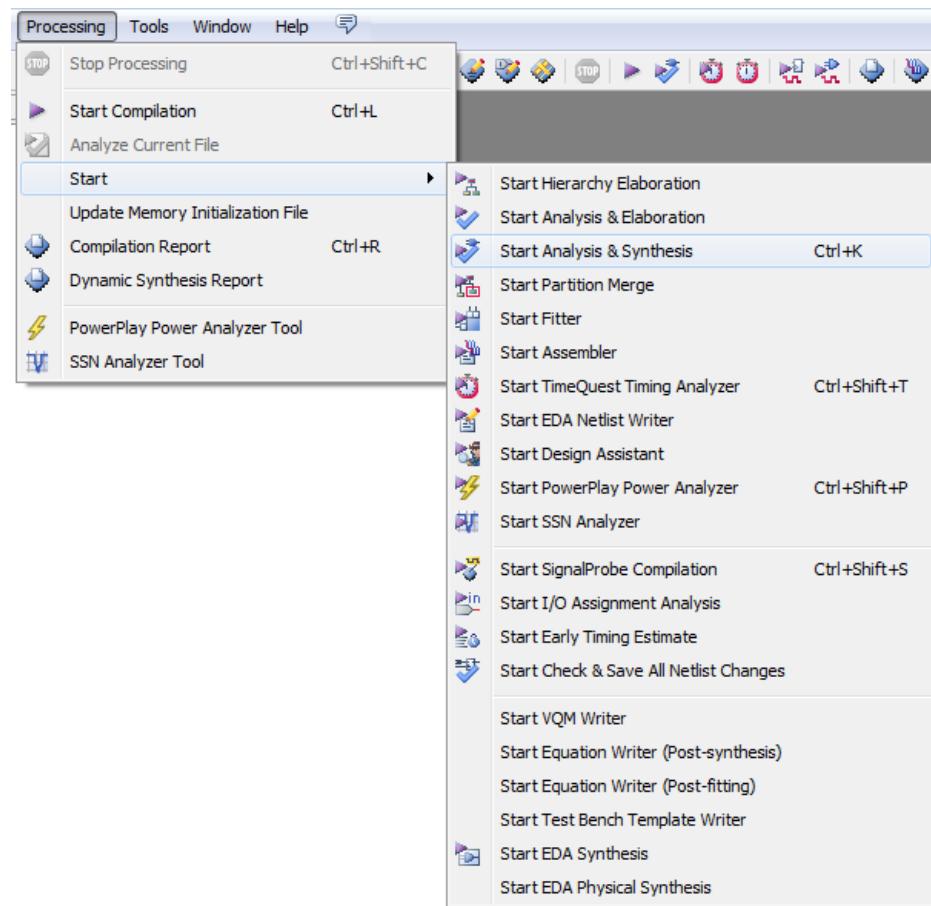
- **Describe the steps of the Quartus II compilation flow**
- **Utilize various Quartus II software tools used to understand how the design was processed**

Quartus II Full Compilation Flow



Processing Options

- **Start Compilation (Full)**
- **Start Hierarchy Elaboration**
 - Checks syntax & builds hierarchy
- **Start Analysis & Elaboration**
 - Checks syntax & builds hierarchy
 - Performs initial synthesis
- **Start Analysis & Synthesis**
 - Synthesizes & optimizes code
- **Start Fitter**
 - Places & routes design
 - Generates output netlists
- **Start Assembler**
 - Generate programming files
- **Analyzers**
 - I/O Assignment
 - PowerPlay
 - SSN (Switching Noise)
 - Design Assistant



Compilation Design Flows

■ Default “flat” compilation flow

- Design compiled as a whole
- Global optimizations performed

■ Incremental flow (on by default for new projects)

- User assigns design partitions
- Each partition processed separately; results merged
- Post-synthesis or post-fit netlists for partitions are reused
- Benefits
 - Decrease compilation time
 - Preserve compilation results and timing performance
 - Enable faster timing closure

Incremental Compilation Concept



$$\begin{array}{c} + \\ \text{A} \\ + \\ \text{B}' \end{array}$$

=



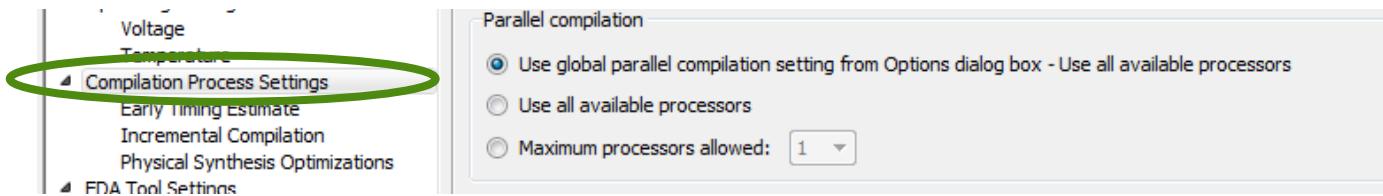
Choose to reuse post-synthesis or post-fit netlist for TOP and A

Only specified portions of logic that have changed are re-synthesized or re-fitted

Note: For more details on using incremental compilation, please attend the course [Design Optimization Using Quartus II Incremental Compilation](#), watch the online training [Introduction to Incremental Compilation](#) or view the built-in interactive tutorial (Module 7: Incremental Compilation)

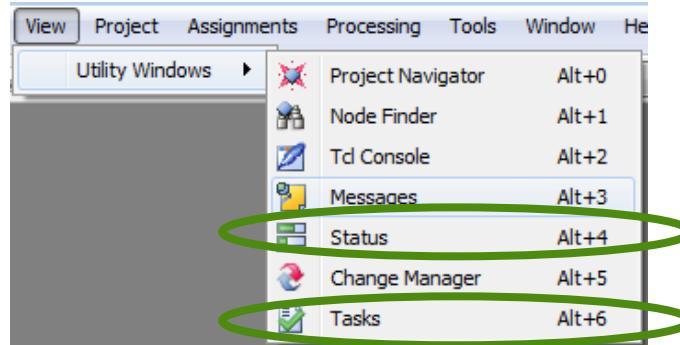
Reducing Compile Times

- Incremental compilation
- Parallel compilation
 - Take advantage of multi-processor, multi-core machines

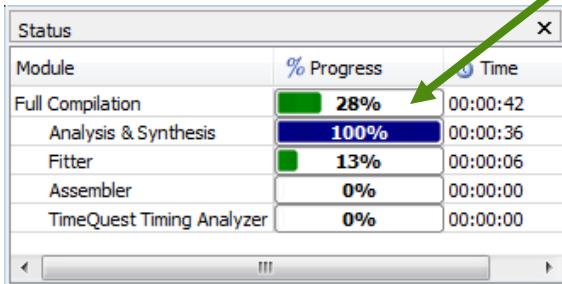


Assignments menu → Settings
(discussed in more detail later)

Status and Tasks Windows



Status bars indicate compilation progress

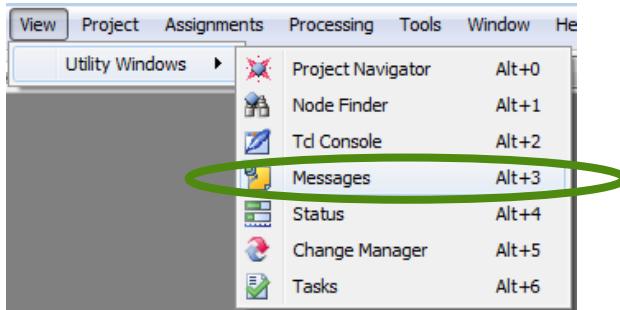


The 'Tasks' window shows the task flow for a 'Full Design' project. The tasks listed are:

Task	Time
Start Project	
Create Design	
Assign Constraints	
Compile Design	00:01:44
Analysis & Synthesis	00:00:36
Fitter (Place & Route)	00:01:08
Assembler (Generate programming files)	00:00:00
TimeQuest Timing Analysis	00:00:00
EDA Netlist Writer	
Program Device (Open Programmer)	
Verify Design	
Export Database	
Archive Project	

Message Windows

- Message window displays informational, warning, and error messages



Filter Messages by Type

All Error Critical Warning Warning Flagged

A screenshot of the 'Messages' window. At the top, there is a toolbar with icons for All, Error, Critical, Warning, and Flagged. Below the toolbar is a table with columns: Type, ID, and Message. The table contains several messages, each with a small icon indicating its type (e.g., information, warning, error). A green arrow points from the 'Error' button in the toolbar to the error message in the list. Another green arrow points from the 'Warning' button in the toolbar to the warning message in the list. A third green arrow points from the 'Flagged' button in the toolbar to the flagged message in the list.

Type	ID	Message
i	21077	High junction temperature is 85 degrees C
i	171003	Fitter is performing an Auto Fit compilation, which may decrease Fitter effort to reduce compilation time
d	169124	Fitter converted 1 user pins into dedicated programming pins
w	15714	Some pins have incomplete I/O assignments. Refer to the I/O Assignment Warnings report for details
i	176045	Design uses memory blocks. Violating setup or hold times of memory block address registers for either read or
d	169085	No exact pin location assignment(s) for 45 pins of 45 total pins
i	184020	Starting Fitter periphery placement operations
i	332104	Reading SDC File: 'pipemult.sdc'
i	332151	Clock uncertainty is not calculated until you update the timing netlist.

Right click to flag selected messages for later review

Message IDs

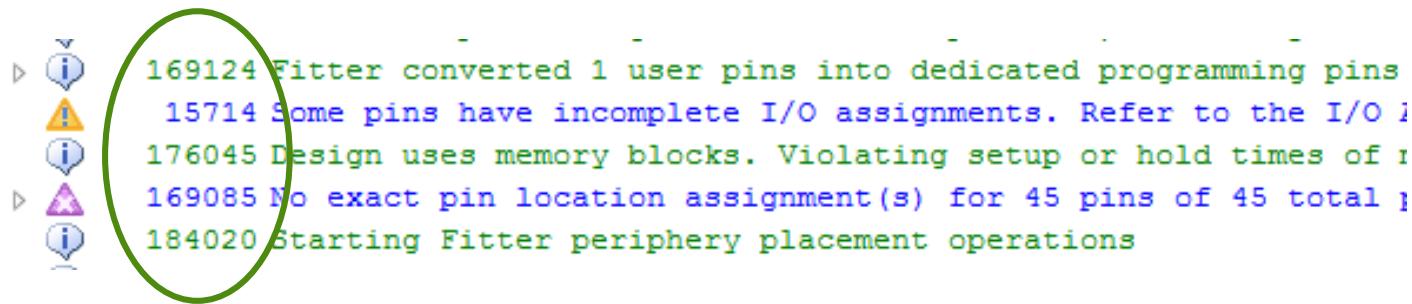
■ Most Quartus II messages have IDs

■ Benefits

- More easily reference messages in service requests
- Search for collateral documentation (Self-Help)
- Disable specific info or warning messages globally (MESSAGE_DISABLE)

■ Benefits for IP developers

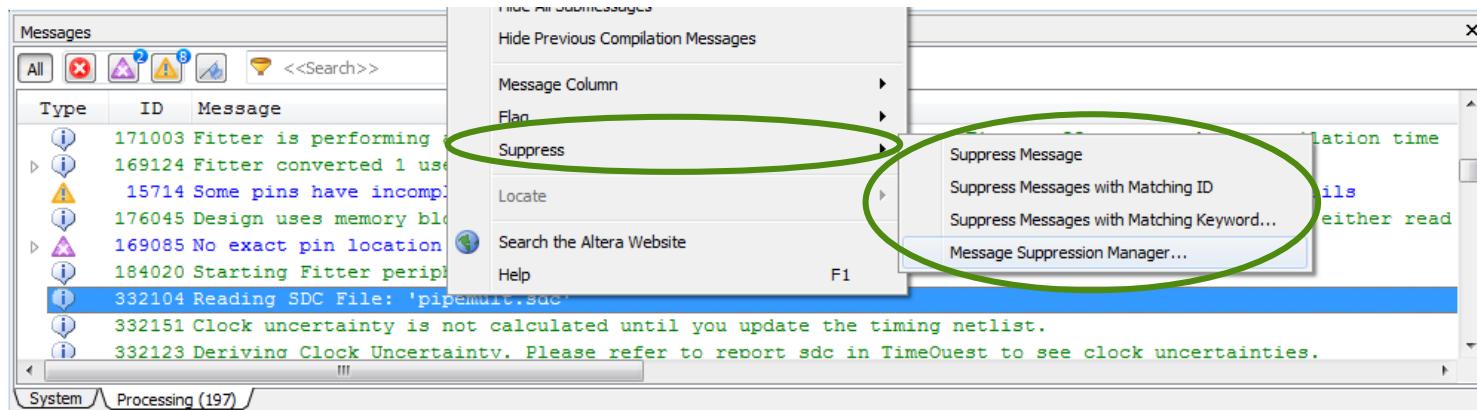
- Disable info or warning messages specifically for an IP core
- Ship “warning-free” IP



Message Suppression

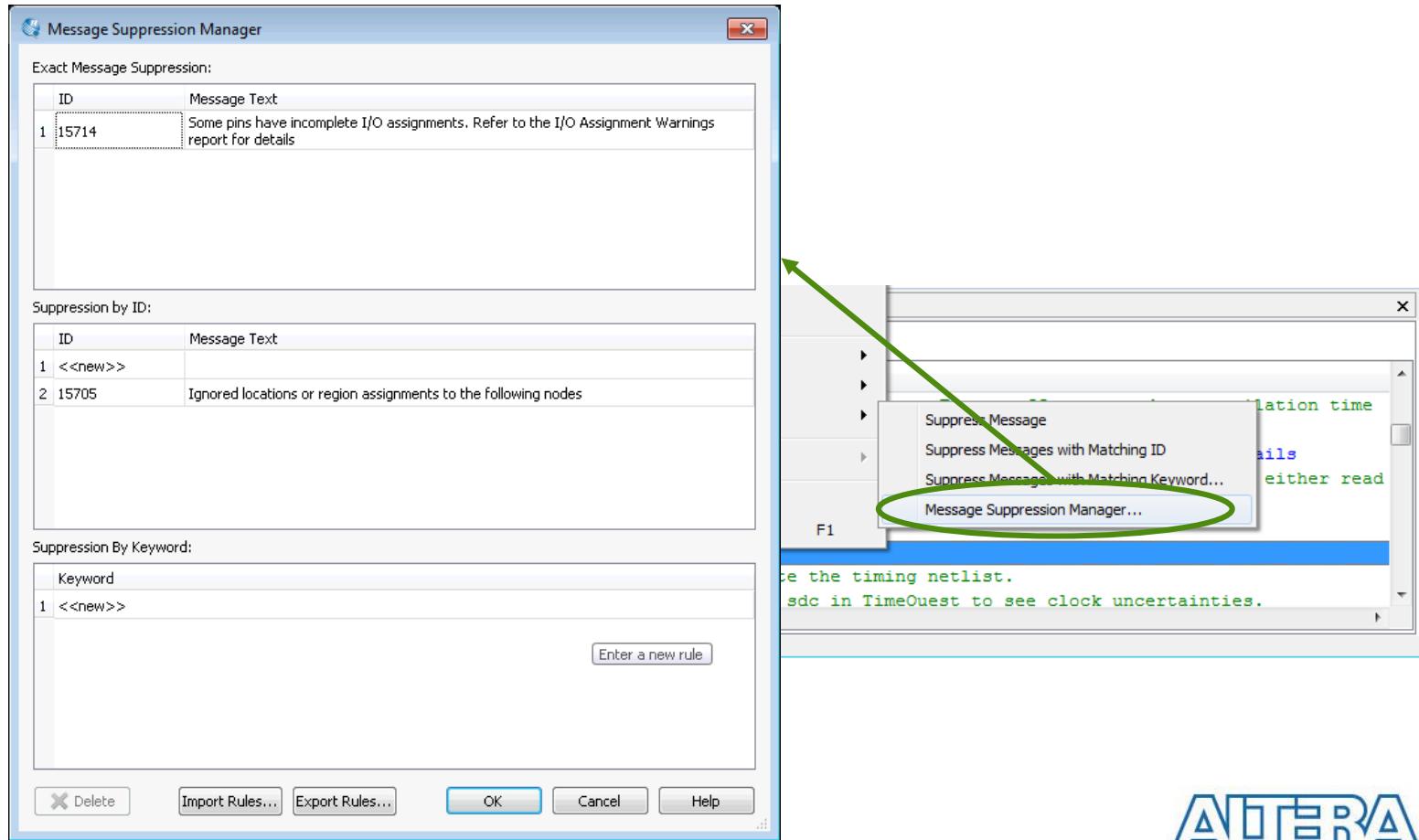
- **Hides messages from current and future compiles**
 - E.g. known synthesis warning message already investigated
- **Stores suppression rules in *<revision_name>.SRF* file**

Right click message to suppress exact message , messages with matching ID, or messages with matching keyword



Message Suppression Manager Tool

- View/edit/remove suppression rules
- Import and export message suppression rules



Viewing Compilation Results

- **Quartus II graphical tools available for**
 - Understanding design processing
 - Verifying correct design results
 - Debugging incorrect results
- **Compilation Report**
- **Viewers**
 - RTL & Technology Map
 - State Machine
- **Chip Planner**
- **Resource Property Editor**

Compilation Report

- Contains all compilation processing information
 - Resource usage
 - Device pin-out
 - Settings and constraints applied
 - Messages
- **Recommendation:** Go through report for a design to get sense of information being provided
- Information also available as text files in project directory - *<revision_name>.fit.rpt*,
<revision_name>.map.rpt, etc

Compilation Report

■ Start from Processing menu or toolbar



Compiling Report - pipemult_lc_phys_syn

Table of Contents

- Flow Settings
- Flow Non-Default Global Settings
- Flow Elapsed Time
- Flow OS Summary
- Flow Log
- Analysis & Synthesis
 - Filter
 - Summary
 - Settings
 - Parallel Compilation
 - I/O Assignment Warnings
 - Netlist Optimizations
 - Ignored Assignments
 - Incremental Compilation Section
 - Pin-Out File
 - Resource Section
 - I/O Rules Section
 - Device Options
 - Operating Settings and Condition
 - Estimated Delay Added for Hold
 - Messages
 - Suppressed Messages
 - Flow Messages
 - Flow Suppressed Messages
- Assembler
- TimeQuest Timing Analyzer

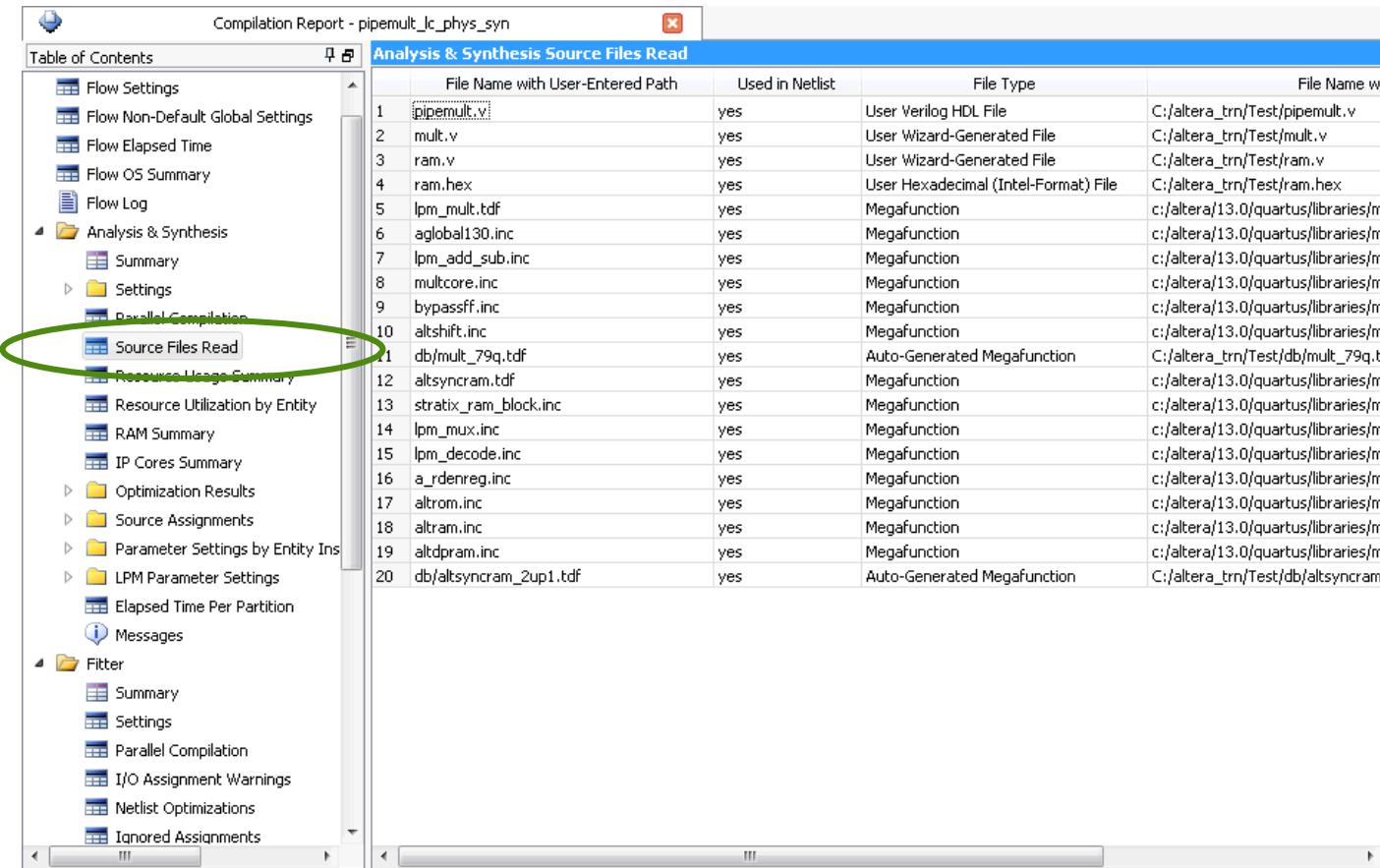
Flow Summary

Flow Status	Successful - Mon May 13 19:39:34 2013
Quartus II 64-Bit Version	13.0.0 Build 156 04/24/2013 SJ Full Version
Revision Name	pipemult_lc_phys_syn
Top-level Entity Name	pipemult
Family	Cyclone IV E
Device	EP4CE6F17C6
Timing Models	Final
Total logic elements	104 / 6,272 (2 %)
Total combinational functions	104 / 6,272 (2 %)
Dedicated logic registers	64 / 6,272 (1 %)
Total registers	80
Total pins	44 / 180 (24 %)
Total virtual pins	0
Total memory bits	512 / 276,480 (< 1 %)
Embedded Multiplier 9-bit elements	0 / 30 (0 %)
Total PLLs	0 / 2 (0 %)

Each compiler executable generates separate folder

Example: Source Files Read

- **Source Files Read table lists all design files (user-coded & library) used during last compilation along with files' type and location**



The screenshot shows the 'Compilation Report - pipemult_lc_phys_syn' window. On the left, the 'Table of Contents' pane is open, displaying various report sections like Flow Settings, Analysis & Synthesis, and Source Assignments. The 'Source Files Read' section is highlighted with a green oval. The main pane displays a table titled 'Analysis & Synthesis Source Files Read' with 20 rows of data. The columns are: File Name with User-Entered Path, Used in Netlist, File Type, and File Name with User-Entered Path (repeated). The table lists files such as pipemult.v, mult.v, ram.v, ram.hex, lpm_mult.tdf, aglobal130.inc, lpm_add_sub.inc, multcore.inc, bypassff.inc, altshift.inc, db/mult_79q.tdf, altsyncram.tdf, stratix_ram_block.inc, lpm_mux.inc, lpm_decode.inc, a_rdenreg.inc, altrom.inc, altram.inc, altdpram.inc, and db/altsyncram_2up1.tdf.

	File Name with User-Entered Path	Used in Netlist	File Type	File Name with User-Entered Path
1	pipemult.v	yes	User Verilog HDL File	C:/altera_trn/Test/pipemult.v
2	mult.v	yes	User Wizard-Generated File	C:/altera_trn/Test/mult.v
3	ram.v	yes	User Wizard-Generated File	C:/altera_trn/Test/ram.v
4	ram.hex	yes	User Hexadecimal (Intel-Format) File	C:/altera_trn/Test/ram.hex
5	lpm_mult.tdf	yes	Megafunction	c:/altera/13.0/quartus/libraries/m
6	aglobal130.inc	yes	Megafunction	c:/altera/13.0/quartus/libraries/m
7	lpm_add_sub.inc	yes	Megafunction	c:/altera/13.0/quartus/libraries/m
8	multcore.inc	yes	Megafunction	c:/altera/13.0/quartus/libraries/m
9	bypassff.inc	yes	Megafunction	c:/altera/13.0/quartus/libraries/m
10	altshift.inc	yes	Megafunction	c:/altera/13.0/quartus/libraries/m
11	db/mult_79q.tdf	yes	Auto-Generated Megafunction	C:/altera_trn/Test/db/mult_79q.t
12	altsyncram.tdf	yes	Megafunction	c:/altera/13.0/quartus/libraries/m
13	stratix_ram_block.inc	yes	Megafunction	c:/altera/13.0/quartus/libraries/m
14	lpm_mux.inc	yes	Megafunction	c:/altera/13.0/quartus/libraries/m
15	lpm_decode.inc	yes	Megafunction	c:/altera/13.0/quartus/libraries/m
16	a_rdenreg.inc	yes	Megafunction	c:/altera/13.0/quartus/libraries/m
17	altrom.inc	yes	Megafunction	c:/altera/13.0/quartus/libraries/m
18	altram.inc	yes	Megafunction	c:/altera/13.0/quartus/libraries/m
19	altdpram.inc	yes	Megafunction	c:/altera/13.0/quartus/libraries/m
20	db/altsyncram_2up1.tdf	yes	Auto-Generated Megafunction	C:/altera_trn/Test/db/altsyncram

Example: Resource Usage

- **Synthesis resource usage:** Estimates of FPGA resources required to implement design
- **Fitter resource usage:** Detailed information on all resources used by design

The screenshot shows the Altera Compilation Report interface. On the left, the Table of Contents lists various sections, with 'Resource Usage Summary' highlighted by a green box. The main pane displays the 'Analysis & Synthesis Resource Usage' report, which includes sections like 'Estimated Total logic elements', 'Total combinational functions', 'Logic element usage by number of inputs', 'Logic elements by mode', 'Total registers', 'I/O pins', and fan-out statistics. A second green box highlights the 'Resource Section' under 'Analysis & Synthesis'. To the right, the 'Fitter Resource Usage Summary' window provides detailed usage statistics for various resources. The table has columns for 'Resource' and 'Usage'. Key entries include:

- Total logic elements: 1 / 5,136 (< 1 %)
- Total registers*: 17 / 6,000 (< 1 %)
- Total LABs: partially or completely used 1 / 321 (< 1 %)
- Total block memory bits 512 / 423,936 (< 1 %)
- Total block memory implementation bits 9,216 / 423,936 (2 %)
- PLLs 0 / 2 (0 %)
- Global clocks 2 / 10 (20 %)

* Register count does not include registers inside RAM blocks or DSP blocks.

Resource	Usage
Total logic elements	1 / 5,136 (< 1 %)
-- Combinational with no register	0
-- Register only	1
-- Combinational with a register	0
Total registers*	17 / 6,000 (< 1 %)
-- Dedicated logic registers	1 / 5,136 (< 1 %)
- I/O registers	16 / 864 (2 %)
Total LABs: partially or completely used	1 / 321 (< 1 %)
Virtual pins	0
I/O pins	45 / 183 (25 %)
-- Clock pins	2 / 4 (50 %)
-- Dedicated input pins	0 / 9 (0 %)
Global signals	2
MSKs	1 / 46 (2 %)
Total block memory bits	512 / 423,936 (< 1 %)
Total block memory implementation bits	9,216 / 423,936 (2 %)
Embedded Multiplier 9-bit elements	1 / 46 (2 %)
PLLs	0 / 2 (0 %)
Global clocks	2 / 10 (20 %)

■ RTL Viewer

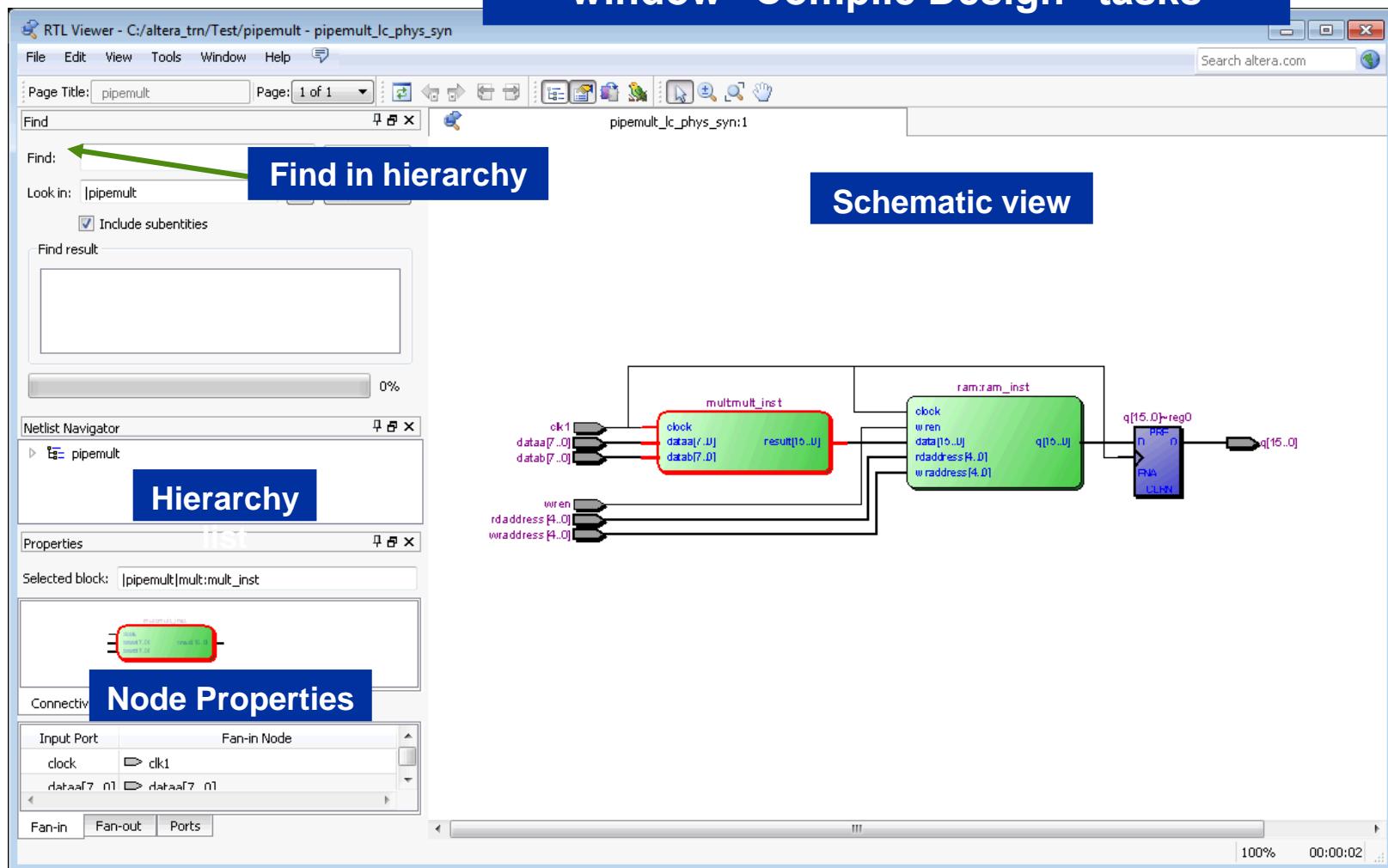
- Schematic of design after Analysis and Elaboration
- Visually check initial HDL before synthesis optimizations
- Locate synthesized nodes for assigning constraints
- Debug verification issues

■ Technology Map Viewers (Post-Mapping or Post-Fitting)

- Graphically represents results of mapping (post-synthesis) & fitting
- Analyze critical timing paths graphically
- Locate nodes & node names after optimizations (cross-probing)

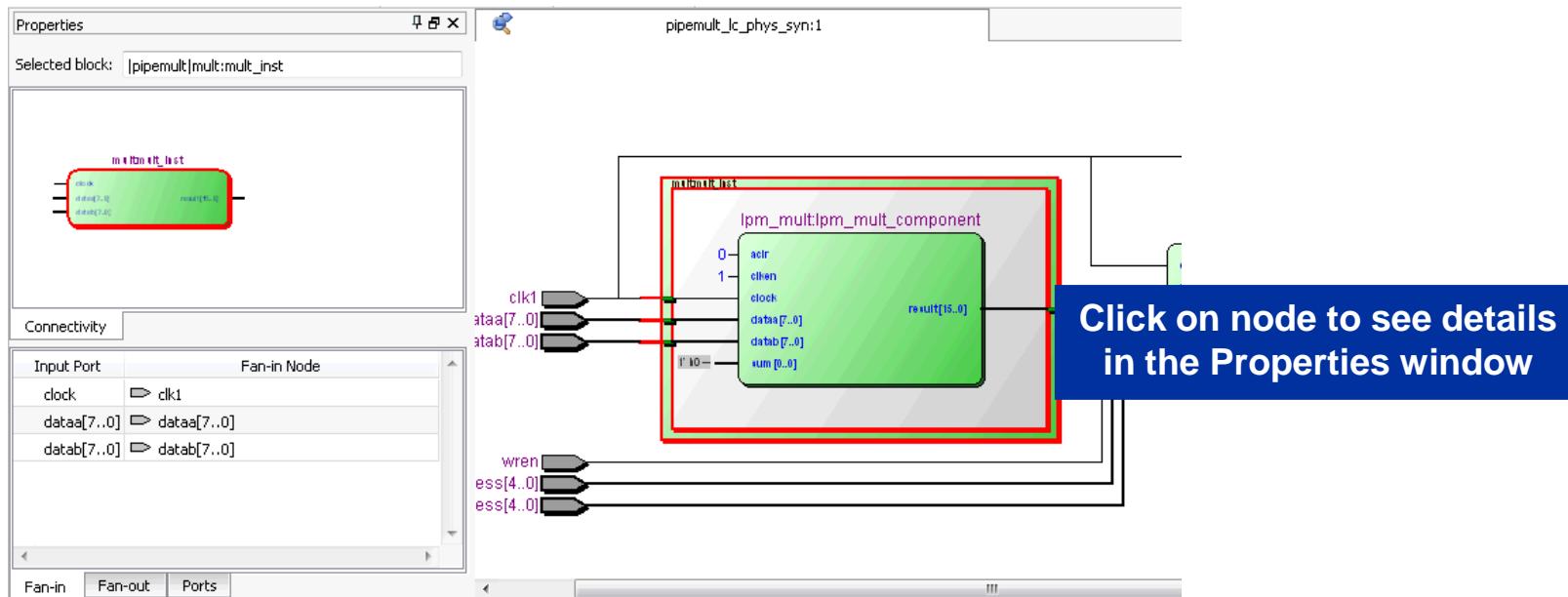
RTL Viewer

Tools menu → Netlist Viewers or Tasks
window “Compile Design” tasks



Note: Must perform elaboration first (e.g. Analysis & Elaboration OR Analysis & Synthesis)

Schematic View (RTL Viewer)



■ Represents design using logic blocks & nets

- I/O pins
- Registers
- Muxes
- Gates (AND, OR, etc.)
- Operators (adders, multipliers, etc.)

Schematic Hierarchy Navigation

■ Descend hierarchy

- Double-click on instance
- Right-click & select **Hierarchy Down**



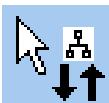
■ Ascend hierarchy

- Double-click in white space
- Right-click & select **Hierarchy Up**



■ Middle hierarchy

- Double-click on instance descends
- Double-click in white space ascends



Technology Map Viewers

Tools Menu → Netlist Viewers or
Tasks window “Compile Design” tasks

Hierarchy list

Schematic view

The screenshot shows the Altera Technology Map Viewer interface. On the left, there is a 'Hierarchy list' window titled 'Selected block: mult_component[mult_79q:auto_generated]dff16'. It displays a schematic symbol for a D flip-flop with its inputs and outputs labeled. Below it is a 'Connectivity' section with a table:

Input Port	Fan-in Node
CLK	clk1~inputclkctrl
D	dffe16~0

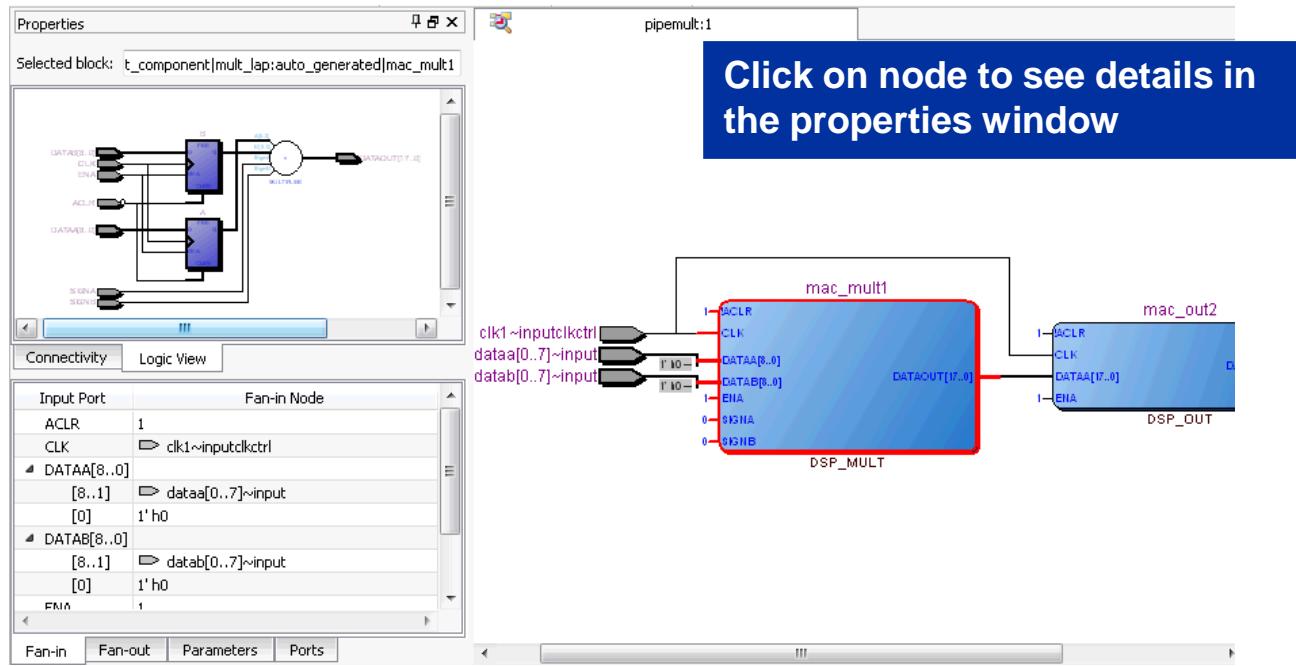
At the bottom of the hierarchy list window are tabs for 'Fan-in', 'Fan-out', and 'Ports'.

The main area of the window is a 'Schematic view' showing a complex logic network. The network consists of various logic cells, including DFFs, AND gates, OR gates, and XNOR gates, connected by wires. Some cells are highlighted in red, indicating selected components. Labels such as 'op_3-0', 'op_2-0', 'w101x[5]-5', 'w204x[1]-0', 'w204x[5]-1', 'op_2-2', 'dfe16', 'dfe17', 'dfe18', 'os2a[5]-2', 'dfe23', and 'dfe25' are visible above the corresponding cells. The schematic is highly detailed, showing internal connections and pin assignments for each component.

Note: Must run synthesis and/or fitting first

Schematic View (Technology Viewer)

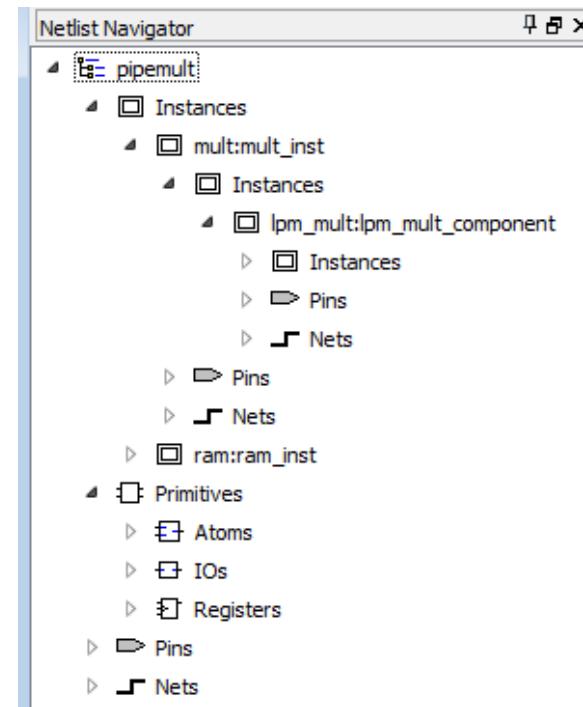
- **Represents design using atoms**
 - I/O pins & cells
 - Lcells
 - Memory blocks
 - MAC (DSP blocks)



© 2013 Altera Corporation—Confidential

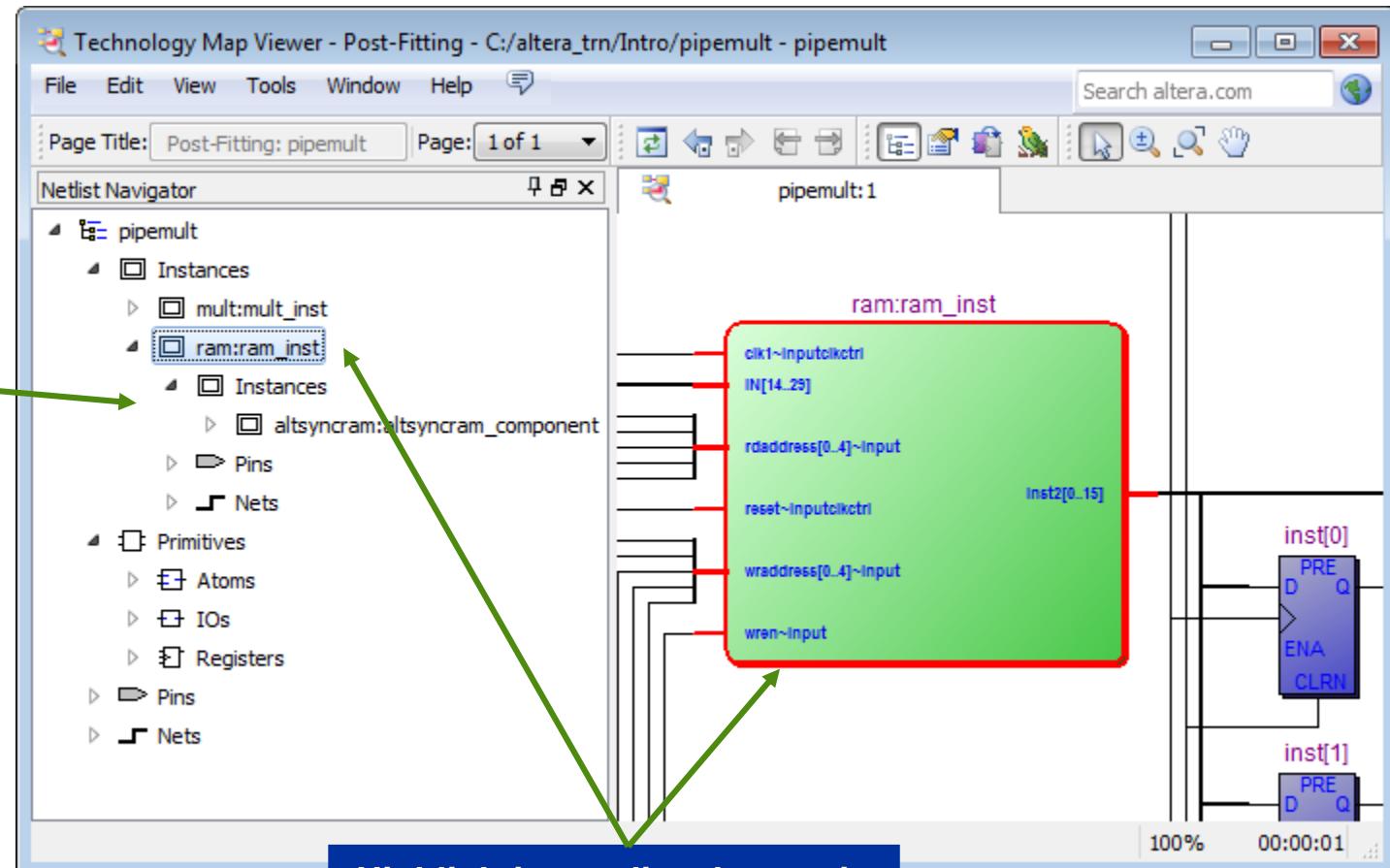
Hierarchy List

- Traverse between levels of design hierarchy
- View logic schematic for each hierarchical level
- Break down each hierarchical level into netlist elements or atoms
 - Instances
 - Primitives
 - Pins
 - Nets
 - State machines
 - Logic clouds (if enabled)



Using Hierarchy List

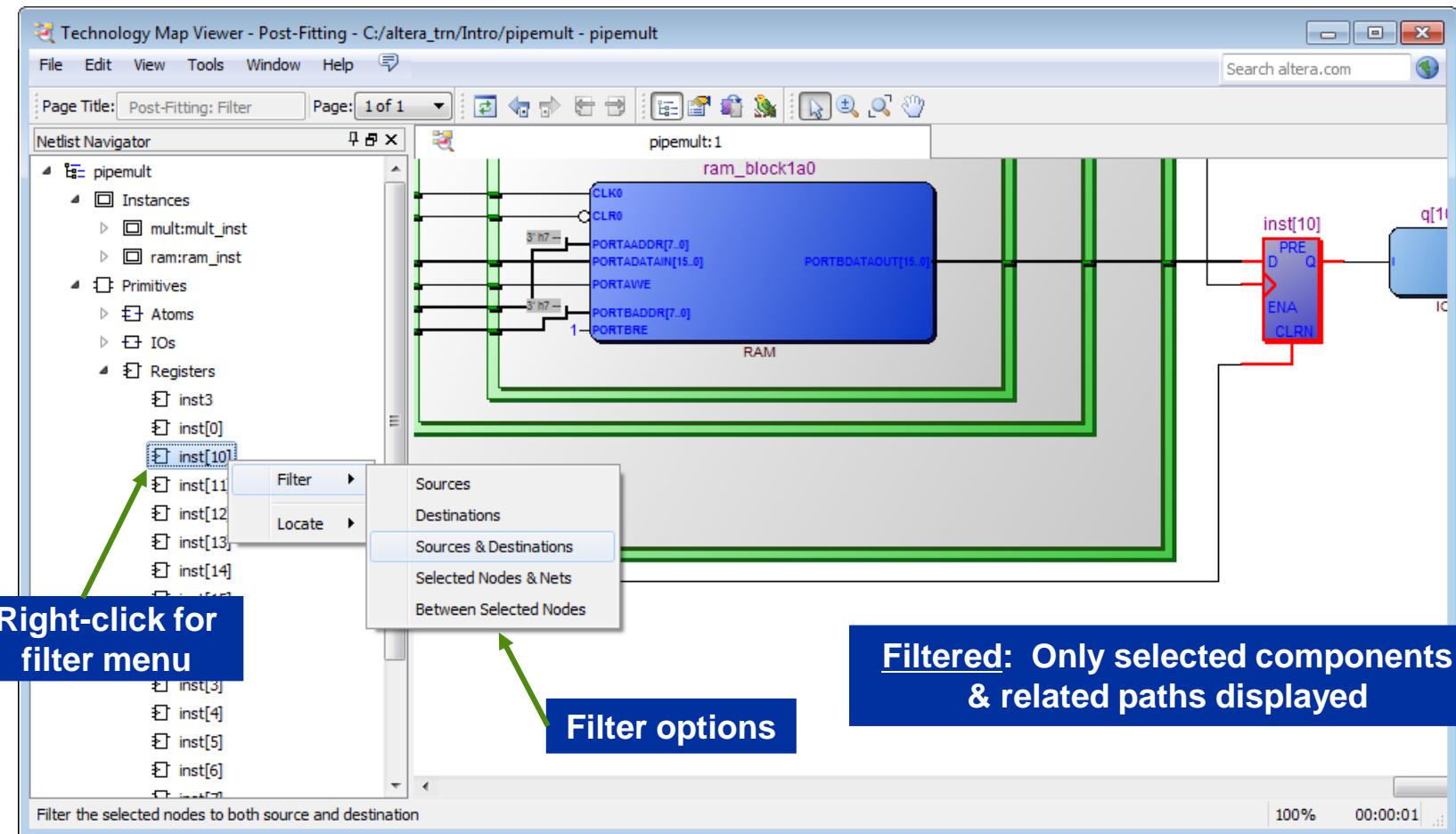
Expanding instances shows
• Instances
• Pins
• Nets



Highlighting netlist element in hierarchy list highlights/views that element in schematic view

Filter Schematic

Unfiltered: All components & paths shown

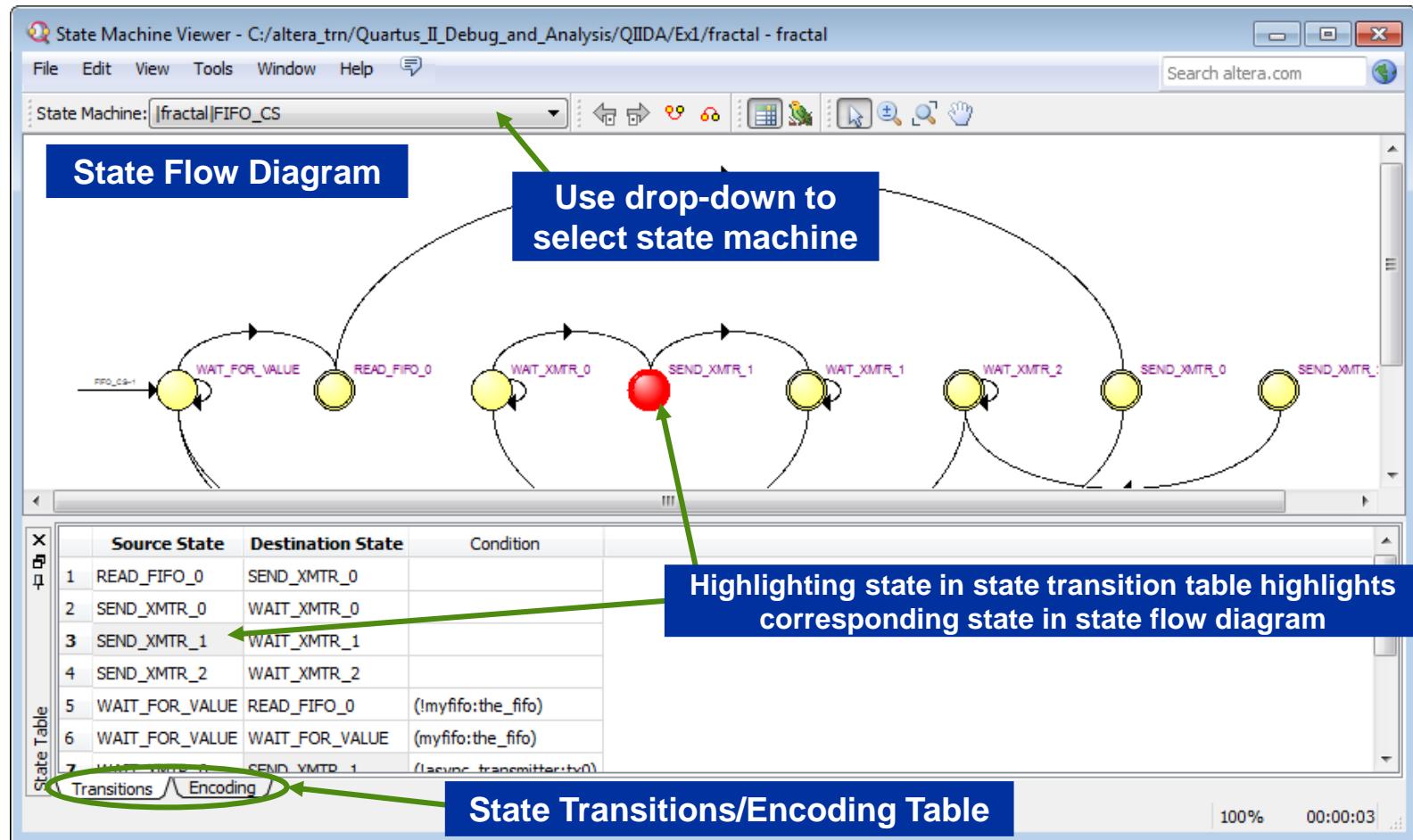


Other Netlist Viewer Features

- **Bird's Eye View:** overall panning view of design
- **Magnifying glass tool**
- **Page control**
- **Go to net driver:** traces net back to source driver
- **Lookup table (LUT) internal detail**
- **Cross-probing:** locate nodes from/to
 - Design files
 - Assignment Editor
 - Chip Planner
 - Resource Property Editor
 - RTL/Technology Map Viewers
 - Pin Planner
 - TimeQuest timing reports

State Machine Viewer

- Tools Menu → Netlist Viewers or Tasks window “Compile Design” tasks



- **Graphical view of design resource usage in target device**
- **Displays**
 - Graphical layout of device resources
 - Routing channels between device resources
 - Global clock regions
- **Uses**
 - View placement of design logic including timing paths
 - View connectivity between resources used in design
 - Make placement assignments
 - Debugging placement-related issues



Chip Planner

Tools Menu or
Tasks window “Compile Design” tasks

Report Window

Device Floorplan

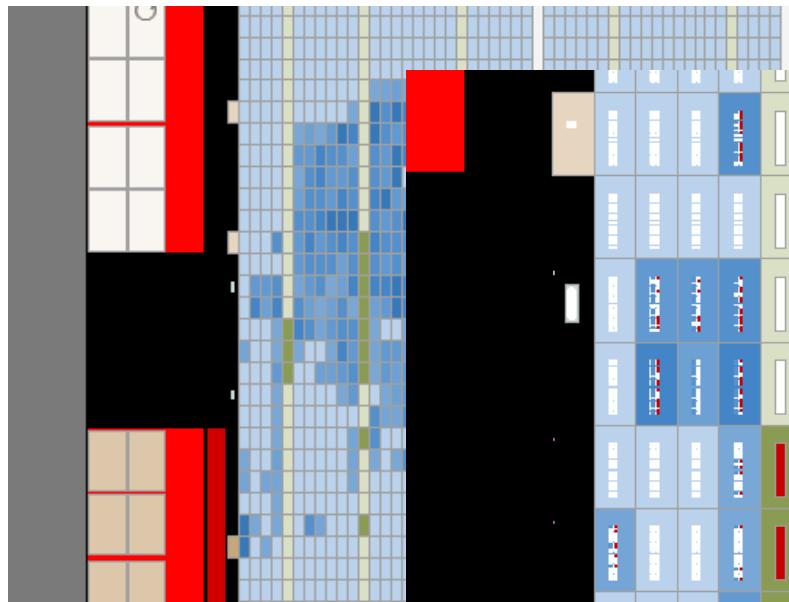
Layer Settings

Selected Node Properties

The screenshot shows the Chip Planner application window. The main area displays a 'Device Floorplan' with various colored regions (blue, green, red) representing different logic blocks and resources. A 'Report Window' is open on the left, showing a tree view of reports like 'Spine Clock Utilization Report'. A 'Tasks window' is also open, listing tasks such as 'Report Resources...', 'Report Compilation Metrics', and 'Mark Selection'. On the right side, there are two panels: 'Layer Settings' which shows options for 'Background', 'Block Utilization', and 'Design Partition Planner'; and 'Selected Node Properties' which shows details for a selected node named 'fractal|Add321~11' at coordinates (112, 59), with tabs for 'Properties/Modes' and 'Values'.

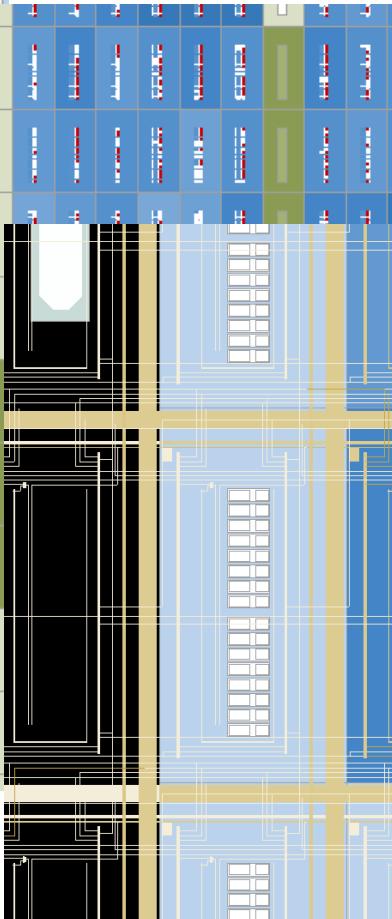
Floorplan Views

Overall device resource usage

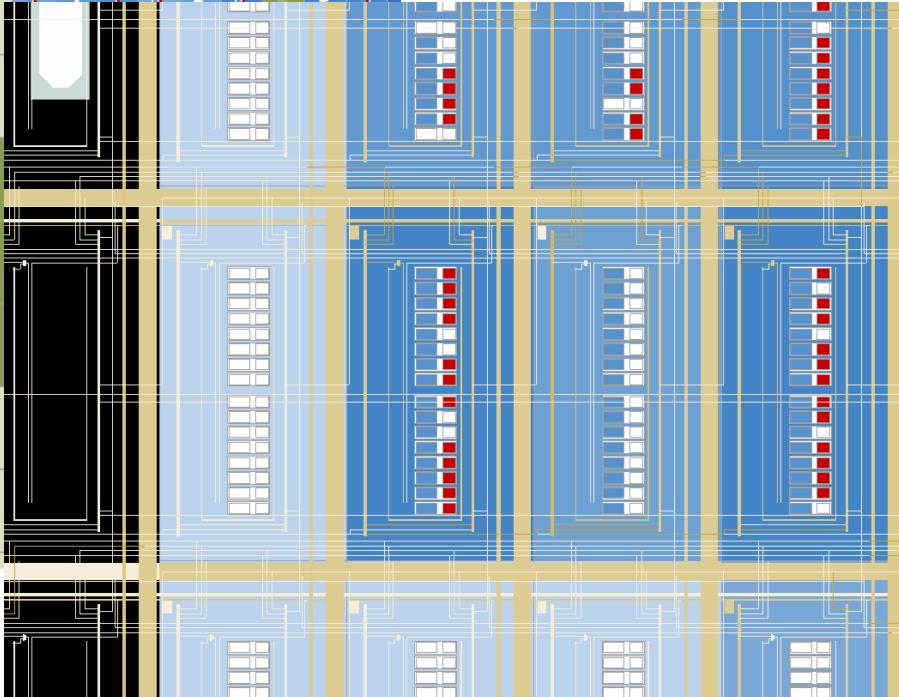


Zoom in for detailed
logic implementation
& routing usage

Lower level block usage

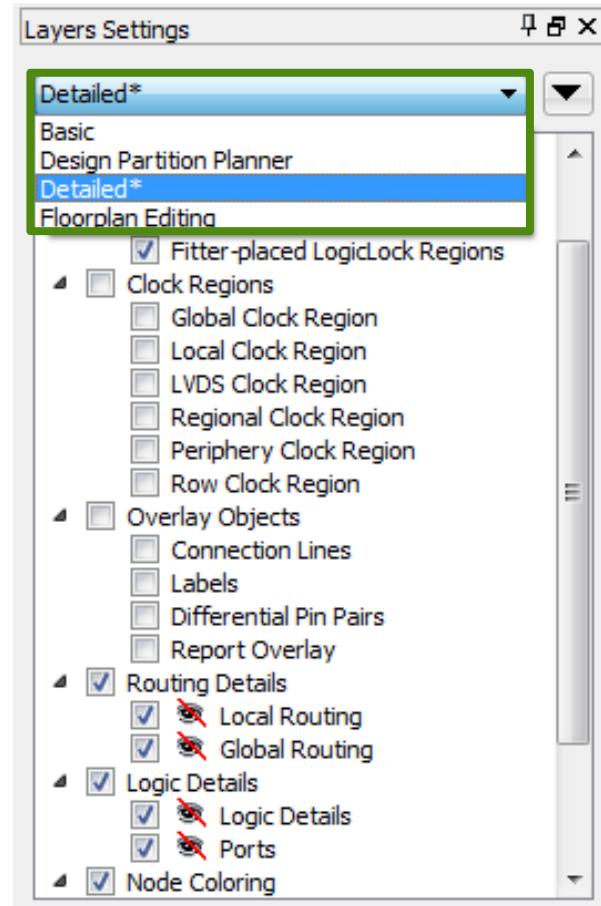


Lowest level routing detail

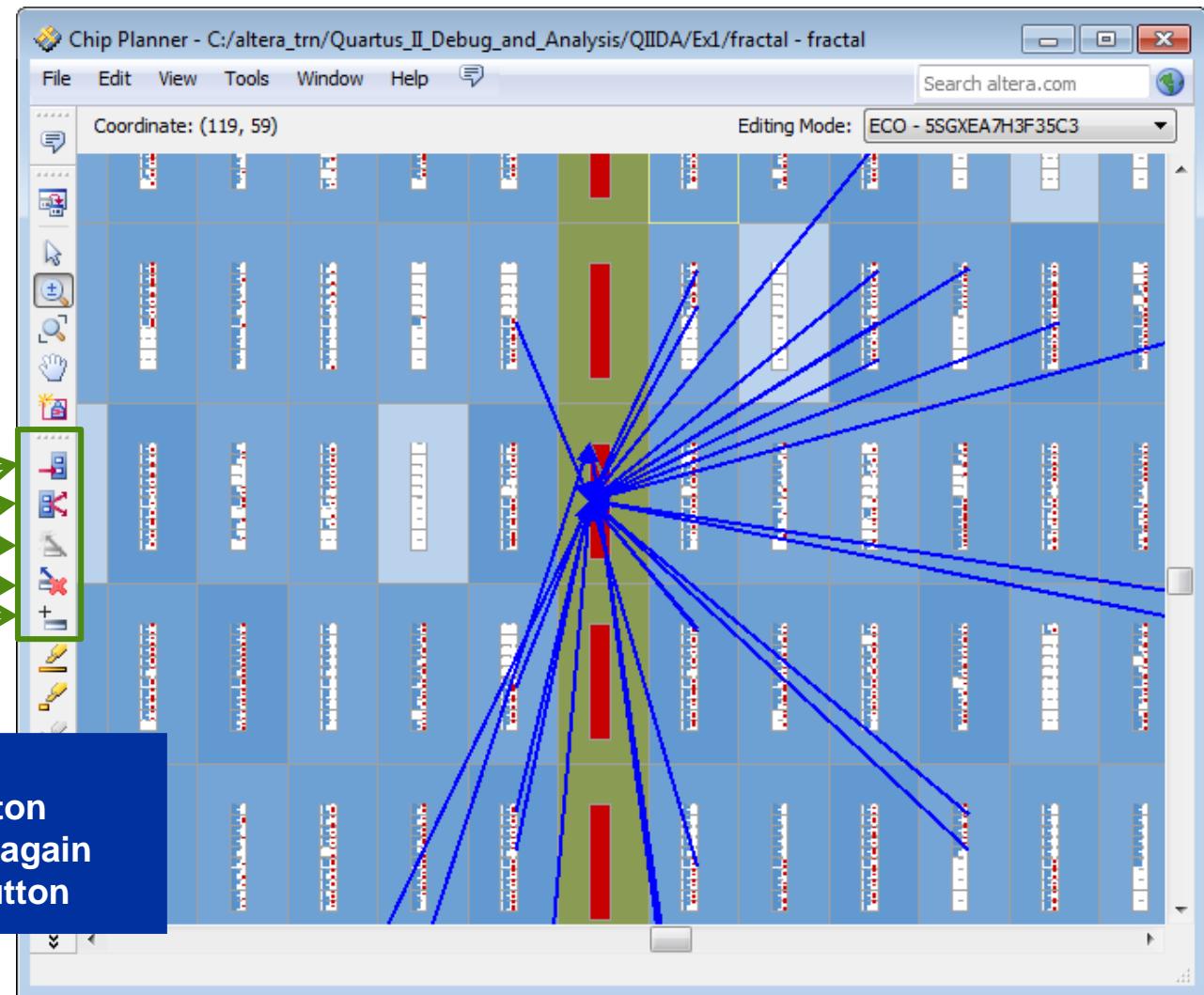


Layers Settings

- Use to quickly change which device resources are displayed
 - e.g. logic details, routing channels
- Default layers
 - Basic
 - Design Partition Planner
 - Detailed
 - Floorplan Editing
- User-defined layers

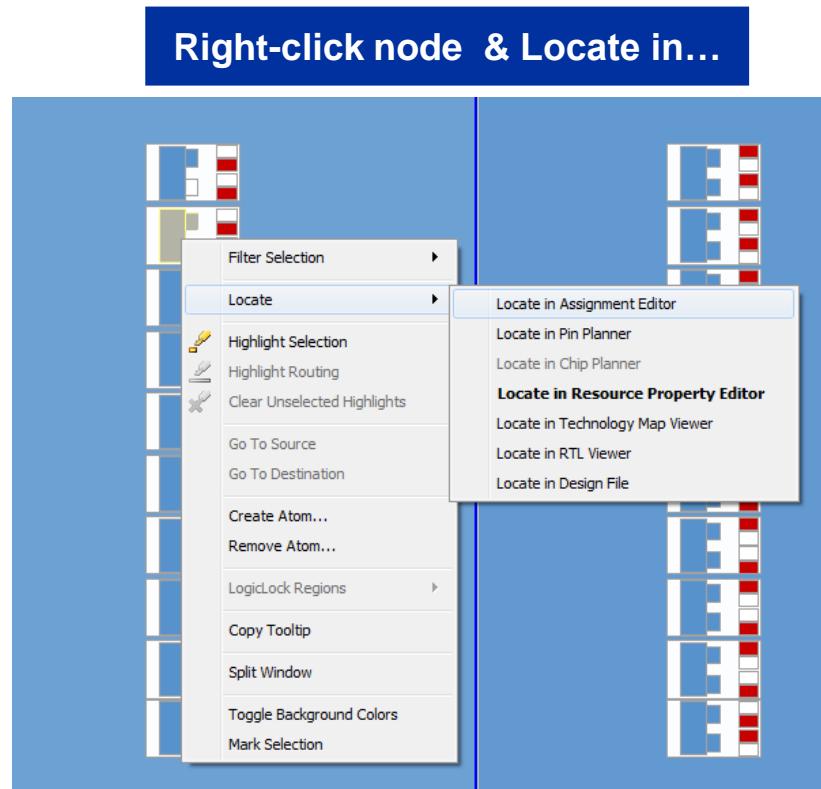


Displaying Fan-In & Fan-Out



Cross-Probing from/to Chip Planner

- Locate hierarchy blocks or specific logic from/to other Quartus II tools
- Project Navigator
- Compilation Report
- Design files
- RTL Viewer
- Technology Viewer
- Message window
- Pin Planner
- TimeQuest reports
- Resource Property Editor



Other Chip Planner Features

- **Bird's Eye view**
- **Magnifying glass tool**
- **Window splitting**
 - Split floorplan into two individually controlled views
- **Routing congestion display**
- **Node Properties window**
 - Displays architectural details and configuration information on highlighted node/cell
- **Location assignments tab**
 - Make location assignments directly in Chip Planner

Reasons for Undesired Results

- Incorrect coding
- Non-recommended coding style
- Suboptimal synthesis & fitting settings/constraints (discussed next)
- Use tools described to find problems and help fix them

*Note: For more details on optimizing designs based on undesired results, please attend the course
[“Timing Closure with the Quartus II Software”](#)*

Compilation Summary

- **Compilation includes synthesis & fitting**
- **Compilation Report contains detailed information on compilation results**
- **Use Quartus II software tools to understand how design was processed**
 - RTL Viewer
 - Technology Map Viewers
 - State Machine Viewer
 - Chip Planner

Class Agenda

- **Quartus II Projects**
- **Design Entry**
- **Quartus II Compilation**
- **Settings & Assignments**
- **I/O Planning**
- **Programming/Configuration**

Quartus II软件设计:基础

Settings & Assignments



© 2013 Altera Corporation—Confidential

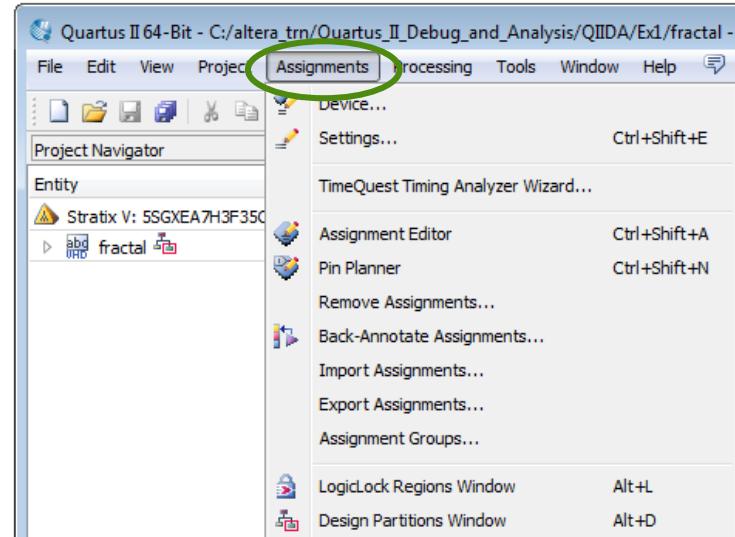
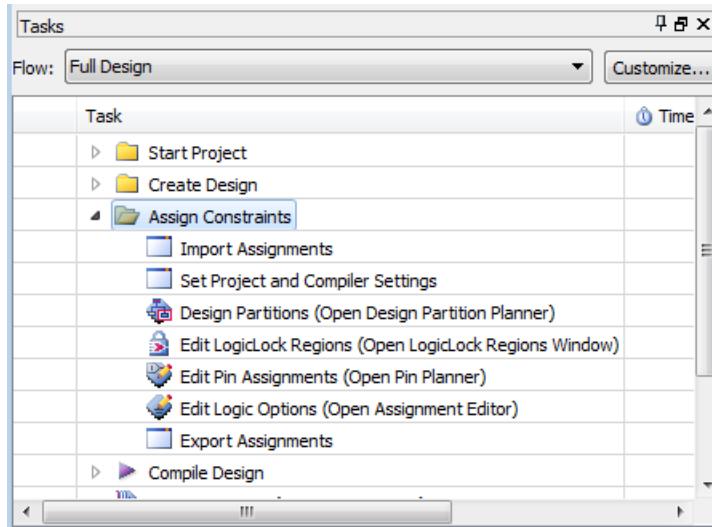


Setting & Assignments Objectives

- Define the difference between settings and assignments
- List some examples of setting assignments that can appear in a Quartus II project
- Create settings and assignments using various methods in the Quartus II software

Synthesis & Fitting Control

- Controlled using two methods
 - Settings - Project-wide switches
 - Assignments - Individual entity/node controls
- Both accessed in Assignments menu or Tasks window
- Stored in .QSF file for project/revision

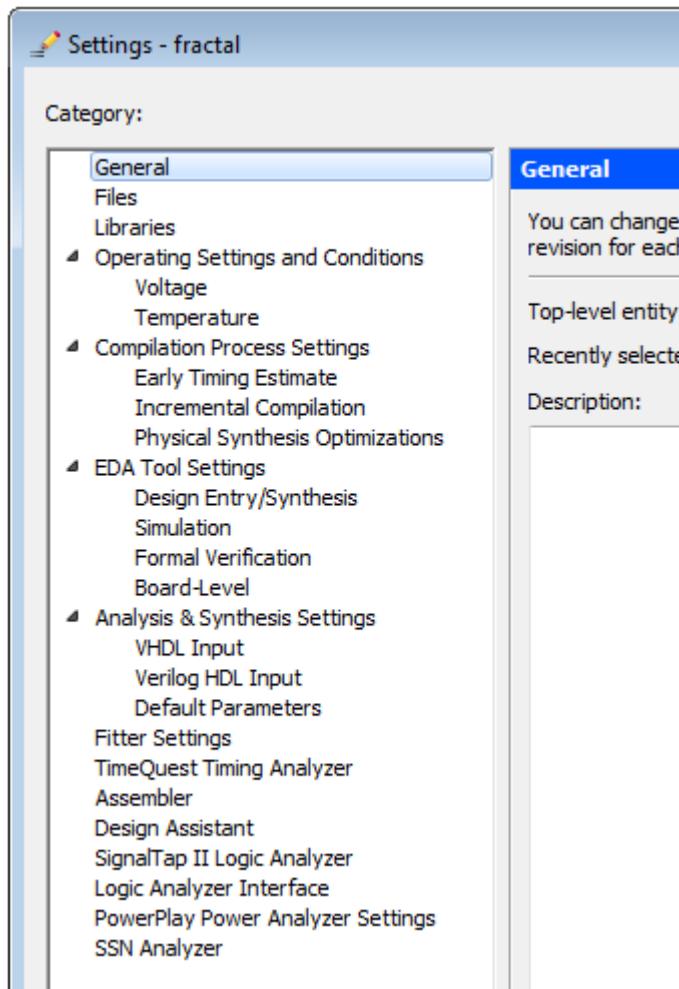


- **Project-wide switches that affect entire design**
- **Examples**
 - Device selection
 - Synthesis optimization
 - Fitter settings
 - Physical synthesis
 - Design Assistant
- **Located in Device and Settings dialog boxes**
 - **Assignments** menu
 - **Set Project and Compiler Settings** task in Tasks window

Settings Dialog Box

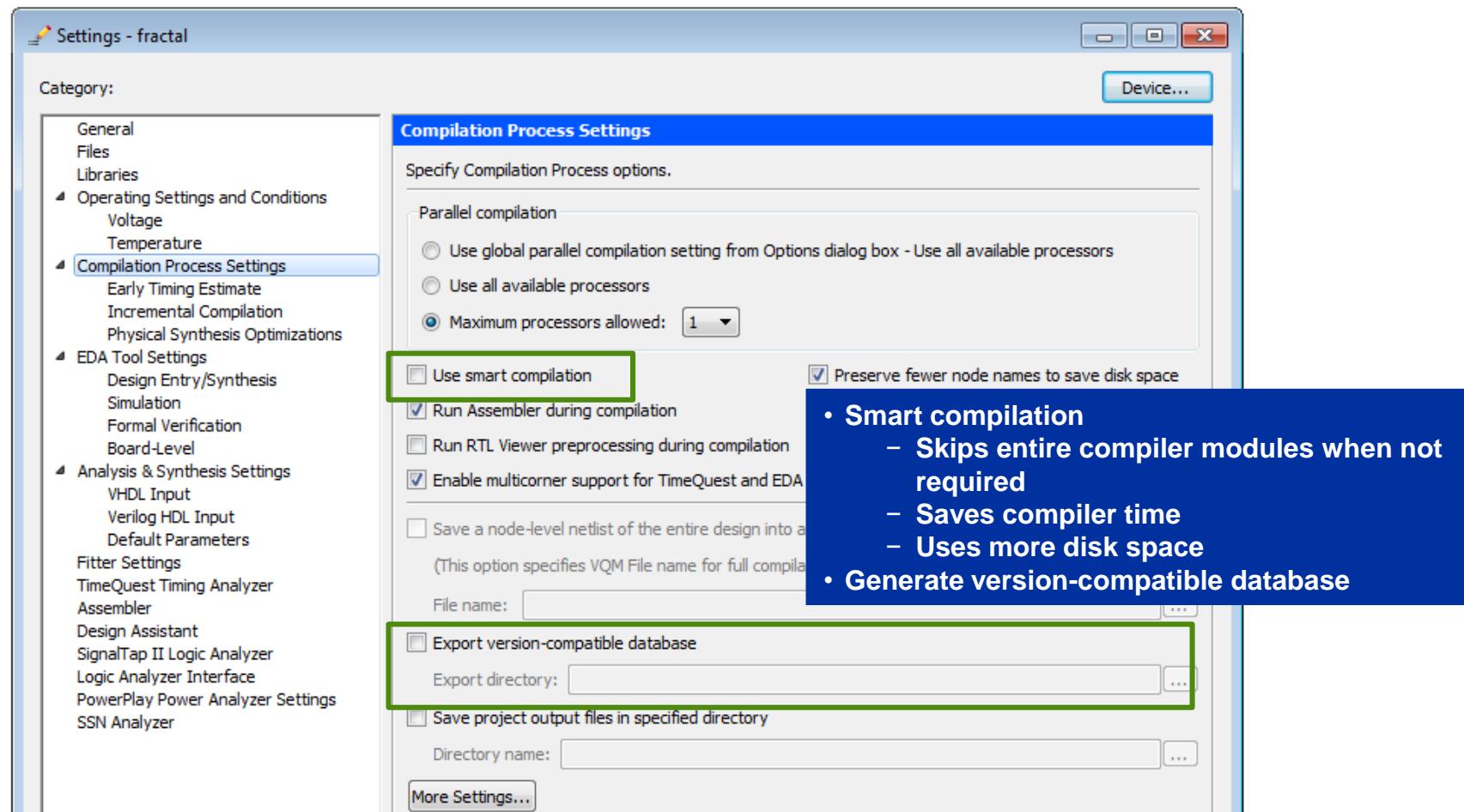
Change settings

- Top-level entity
- Add/remove files
- Libraries
- Compiler settings
- EDA tool settings
- Synthesis settings
- Fitter settings
- Timing analyzer settings
- Power analysis settings



Tcl: `set_global_assignment -name <assignment_name> <value>`

Compilation Process Setting Examples



Tcl: *set_global_assignment -name SMART_RECOMPILE ON*

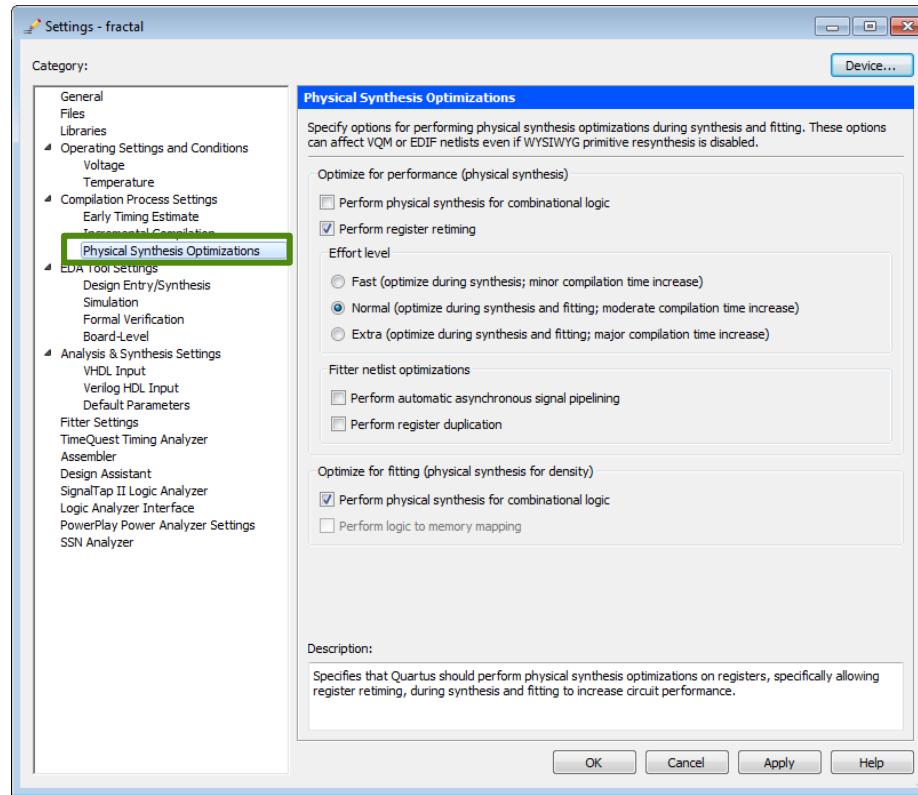
Version-Compatible Database

- **Migrating design between versions of Quartus II software**
- **Exports special database (export_db)**
- **Preserve compilation results between Quartus II software versions**
 - Re-run timing analysis or simulation with updated timing models
 - No need to fully recompile
- **Two methods to create**
 - Settings dialog box
 - Project menu (**Export Database**)

```
Tcl: set_global_assignment -name AUTO_EXPORT_VER_COMPATIBLE ON  
Tcl: set_global_assignment -name VER_COMPATIBLE_DB_DIR <directory_name>
```

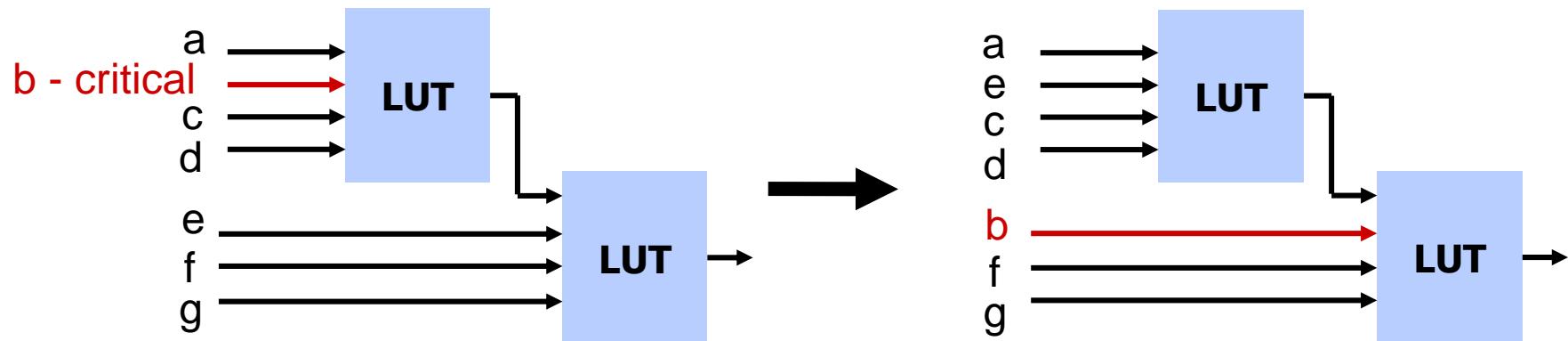
Physical Synthesis

- Optimize during synthesis or re-synthesize based on Fitter output
 - Makes incremental changes that improve results for a given placement
 - Compensates for routing delays from Fitter
 - Apply globally (Settings) or only to specific design entities (Assignment Editor)



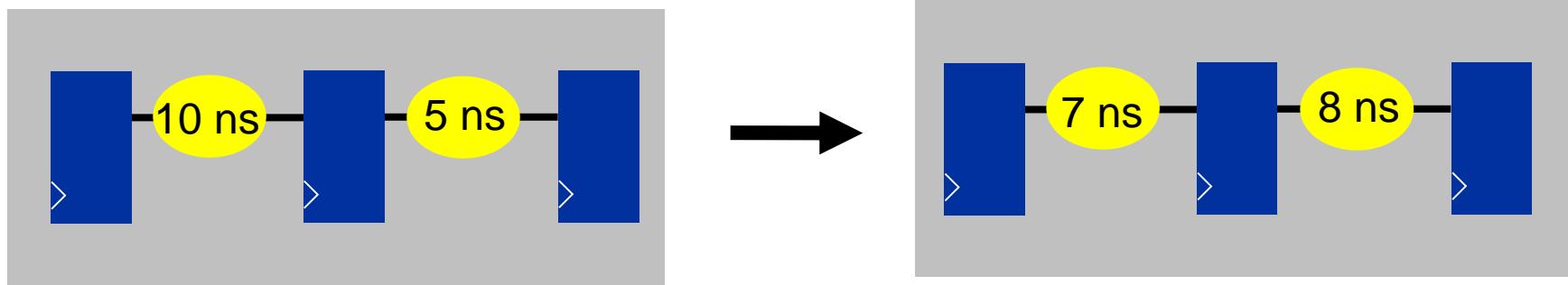
Physical Synthesis Example (Combinational Logic)

- Swaps look-up table (LUT) ports within logic elements (LEs) to reduce critical path delay



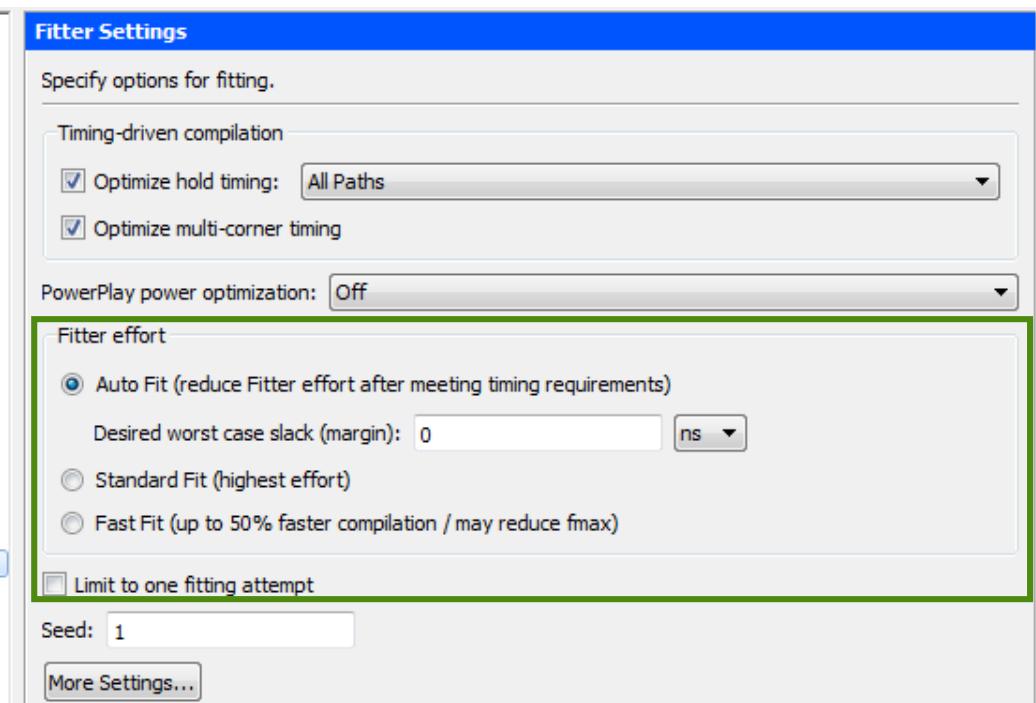
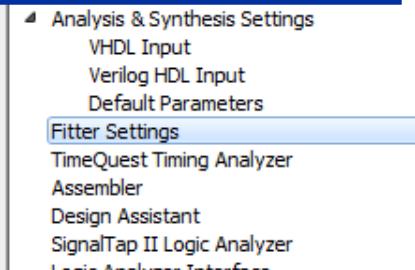
Physical Synthesis Example (Gate-Level Register Retiming)

- Moves registers across combinatorial logic to balance timing
- Trades between critical & non-critical paths
- Makes changes at gate level
- Does not change logic functionality



Fitter Settings – Fitter Effort

- Standard Fit
 - Highest effort
 - Longest compile time
- Fast Fit
 - Faster compile
 - Possibly lesser performance
- Auto Fit
 - Compile stops after meeting timing
 - Conserves CPU time
 - Will mimic standard fit for hard-to-fit designs
 - Default for new designs
- One fitting attempt



Tcl: set_global_assignment -name FITTER_EFFORT "<Effort Level>"

Assignments - Logic Options, Constraints

- Individual switches applied
 - I/O
 - Internal nodes
 - Hierarchical blocks (design entities)
- Assignment Editor manages assignments
- Must perform analysis & elaboration
- Example assignments
 - Optimization Technique
 - PCI I/O

Assignment Editor

■ Provides spreadsheet assignment entry & display

- Copy & paste support
- Multi-cell editing

Assignments menu or Tasks window

The screenshot shows the Assignment Editor interface with the following features highlighted:

- Filter nodes**: A blue callout points to the "Filter on node names" input field at the top of the grid.
- Sort on customizable columns**: A blue callout points to the column headers, specifically the "Assignment Name" header which is currently sorted.
- Enable/disable individual assignments**: A blue callout points to the "Enabled" column, which contains checkboxes for each row.

Status	From	To	Assignment Name	Value	Enabled	Entity
1 Ok		mult:mult_inst	DSP Block Balancing	Logic Elements	Yes	pipemult
2 Ok		mult:mult_inst	Optimization Technique	Speed	Yes	pipemult
3	<<new>>	<<new>>	<<new>>			

Information panel (bottom right):

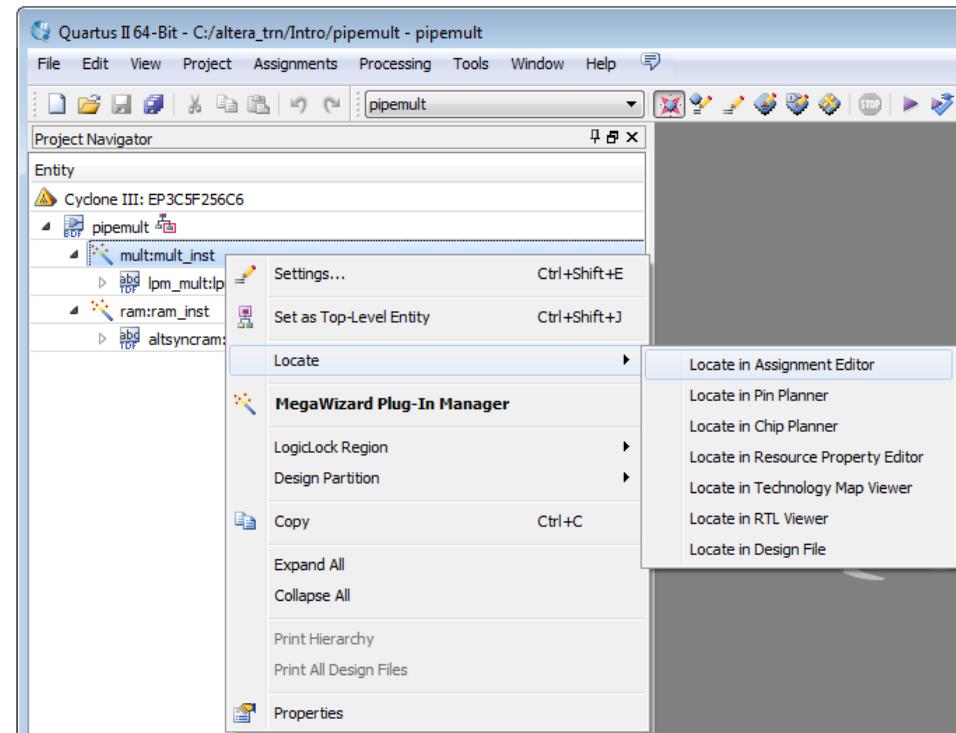
Specifies the overall optimization goal for Analysis & Synthesis: attempt to maximize performance, minimize logic usage, or balance high performance with minimal logic usage.

Creating Assignments: Cross-Probing

■ Cross-probe (Locate) to Assignment Editor

- Project Navigator
- Message window
- Compilation Report
- Design files

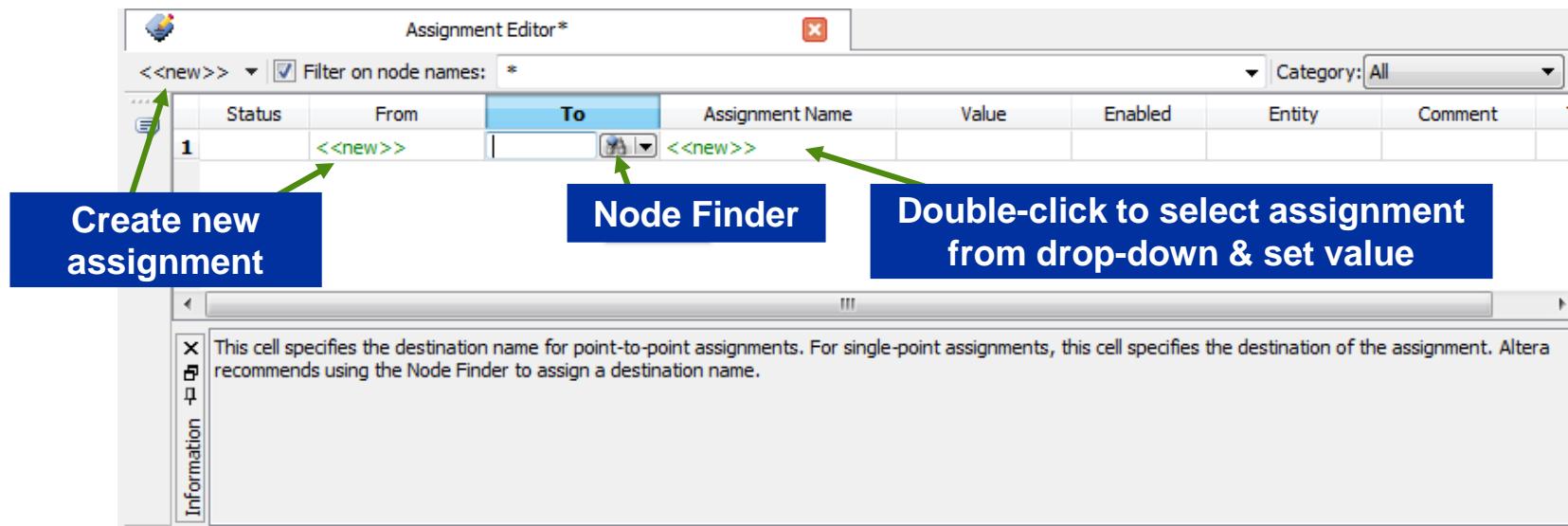
■ Assignment Editor Node Filter automatically filled in with cross-probed node(s)



■ Easiest method for creating assignments

Creating Assignments: Assignment Editor

- Locate from other tools
- Double-click on <<new>> button or <<new>> in the From or To columns
 - Type in object name
 - Click on Node Finder icon to search



Node Finder



Search by name using wildcards (? Or *)

Use filter to select the type of nodes to be selected

Show/hide more filters

Node Finder

Named: *clk*

Options

Filter: SignalTap II: pre-synthesis

Look in: [pipemult]

Nodes Found:

Name	Assignments
pipemult	
in clk1	PIN_E15
mult:mult_inst	
lpm_mult:lpm_mult_component	
clk1	Unassigned
mult_lap:auto_generated	
clk1	Unassigned

Selected Nodes:

Name	Assignments
mult:mult_inst lpm_mult...lap:auto_generated clk1	Unassigned
clk1	
mult_lap:auto_generated	
clk1	

List of found nodes arranged by hierarchy

Select nodes on left & use arrows to move to the right

Tooltip with hierarchy path to target node

Use filter to select the type of nodes to be selected

Show/hide more filters

Locate nodes in a certain level of hierarchy

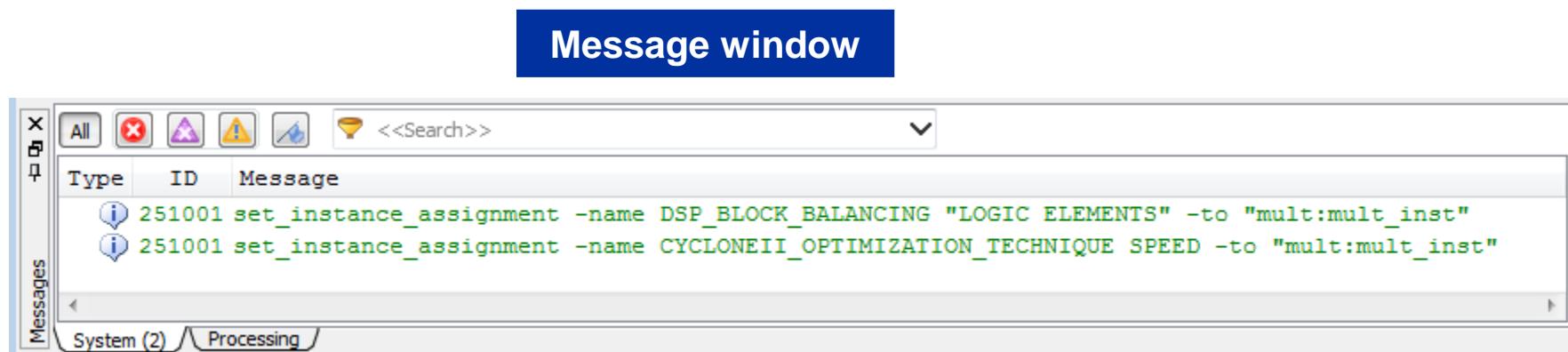
Selected Nodes:

Name	Assignments
mult:mult_inst lpm_mult...lap:auto_generated clk1	Unassigned
clk1	
mult_lap:auto_generated	
clk1	

Assignment Tcl Commands

■ Equivalent Tcl commands displayed as assignments are entered

- Manually copy to create Tcl scripts
- Export command (**File** menu) writes all assignments to a Tcl file



Export Assignments

■ Export to CSV file (File menu → Export)

- Import and edit assignments in Excel

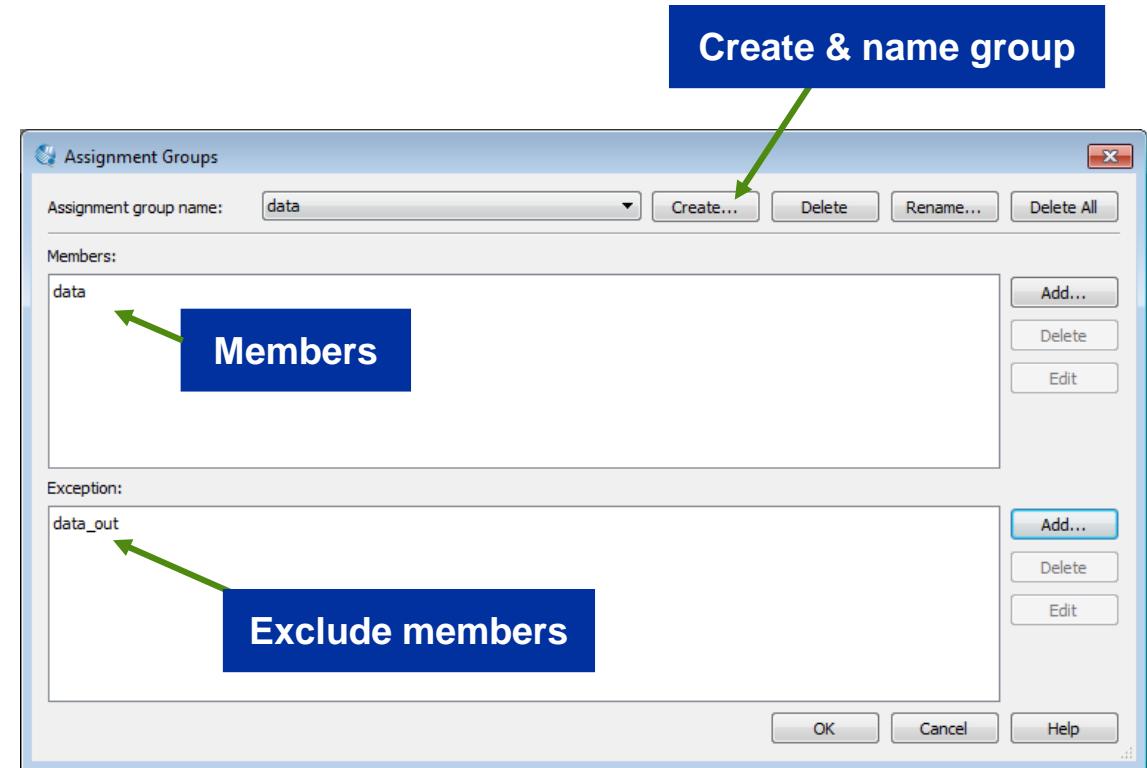
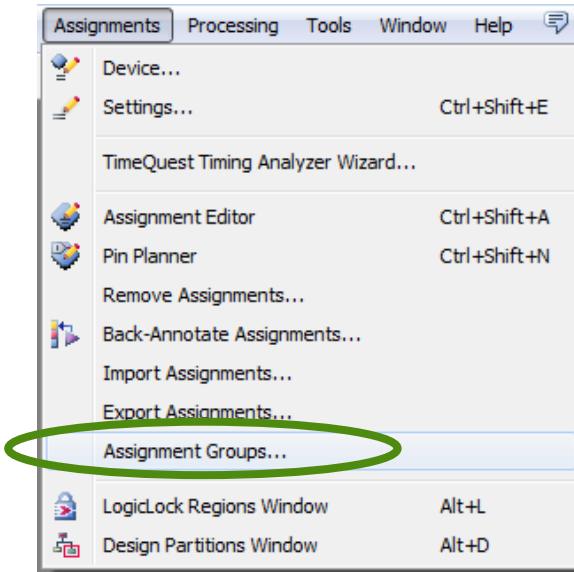
15	# Quartus II 64-Bit Version 12.1 Build 177 11/07/2012 SJ Full Version							
16	# File: C:\altera_trn\Intro\pipemult_assignments.csv							
17	# Generated on: Fri Dec 21 15:58:44 2012							
18								
19	Status	From	To	Assignment Name	Value	Enabled	Entity	Comment Tag
20	Ok		mult:mult_inst	DSP Block Balancing	Logic Elements	Yes	pipemult	
21	Ok		mult:mult_inst	Optimization Technique	Speed	Yes	pipemult	
22		<<new>>	<<new>>	<<new>>				
23								

■ Export QSF (Assignments menu → Export Assignments)

- Migrate assignments to another project

Assignment Groups

- Assign names to user-defined groups of nodes
- Allows single assignment to constrain entire group

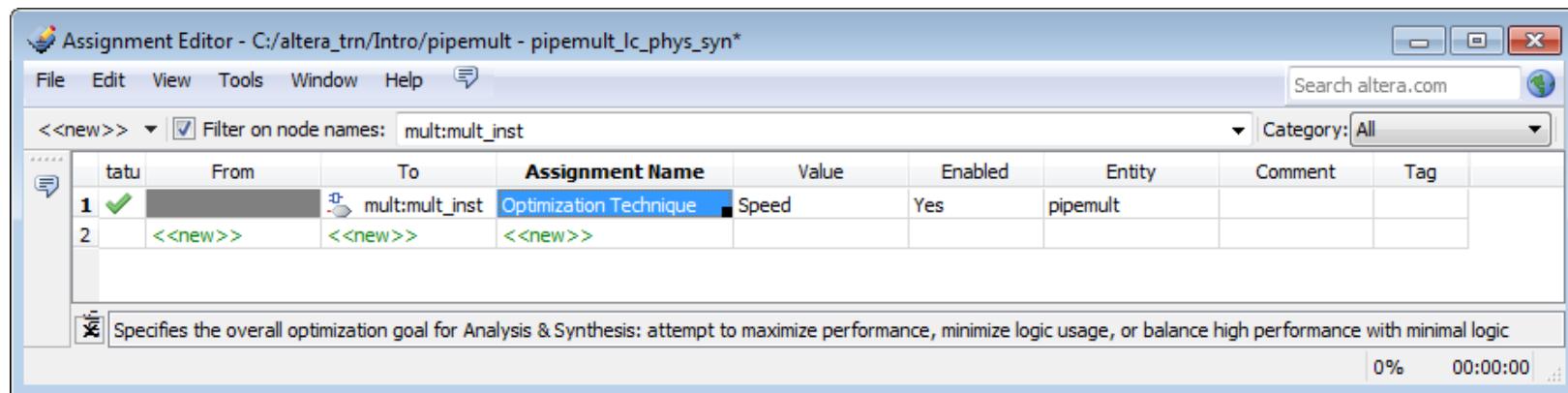


Example Assignments

- Optimization Technique
- PCI I/O

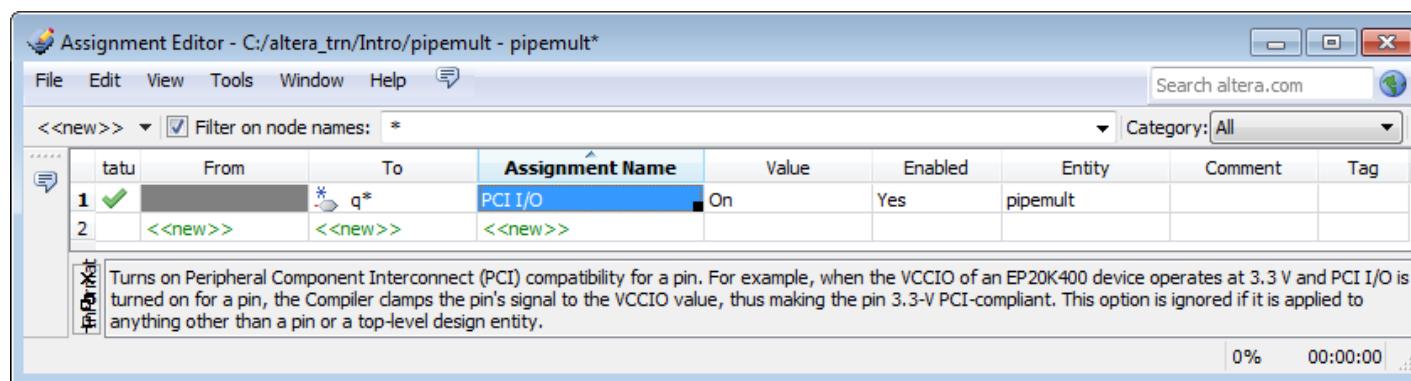
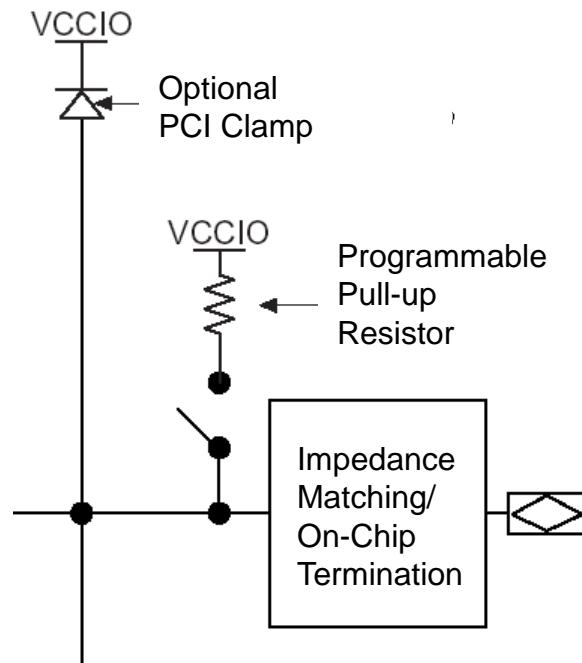
Optimization Technique

- Selects synthesis optimization goal (*Speed, Balanced, Area*)
- Apply to hierarchical entities
 - Locate (cross-probe) from Project Navigator
 - Or drag and drop into Assignment Editor
- May also apply project-wide in Analysis & Synthesis Settings
- Affects synthesis & logic mapping - applies to Quartus II synthesis



Tcl: `set_instance_assignment -name OPTIMIZATION_TECHNIQUE SPEED -to <entity_name>`

- Turns on PCI™ compatibility for pins
 - Ignored if applied to anything other than a pin or a top-level design entity
- Controls clamping diode located in the I/O elements
 - For applicable families



Tcl: `set_instance_assignment -name PCI_IO ON -to <pin name>`

Available Logic Options (Assignments)

The screenshot shows a Windows Internet Explorer window displaying the Quartus II Help v12.1 "Logic options" page. The browser title bar reads "Quartus II Help v12.1 - Windows Internet Explorer". The address bar shows the URL "file:///C:/altera/12.1/quartus/common/help/webt...". The page content is titled "Quartus II Help v12.1 > Logic Options". On the left, there is a navigation pane with a tree view of help topics, including "Using the Netlist Viewer", "Using the State Machine Editor", "Integrating Other EDA Tools", "Devices and Adapters" (which is expanded), "Logic Options" (which is highlighted with a green box and has a green arrow pointing to it from a callout box), "API Functions for Tcl", "Quartus II Scripting Support", "Shortcuts", "Glossary", "I/O Standards", and "Message List". The main content area is titled "Logic options" and contains the text: "The following logic options are available in the Quartus II software." followed by a list of categories: "Advanced logic options:", "Global Signals logic options:", "I/O Features logic options:", "I/O Timing Logic Options:", "Synthesis logic options:", "Simulation logic options:", "Fitter optimization:", and "Other:". A blue callout box on the right side of the page contains the text: "Links to all available assignments organized by category with lists of supported devices".

Select “Logic Options” in Quartus II Help “Contents”

Links to all available assignments organized by category with lists of supported devices

Updating QSF File

- **QSF not updated automatically when constraint entered or Assignment Editor saved**
- **QSF updated only when**
 - Project is saved (**File** menu)
 - Project is closed
 - Beginning of any processing task (e.g. compilation)
 - Assignment Editor is closed or saved
 - MegaWizard manager is launched
 - Changing revisions
- **Change behavior to updating assignments immediately
(Tools menu → Options → General → Processing)**
 - May impact software performance slightly due to file accesses

Design Assistance

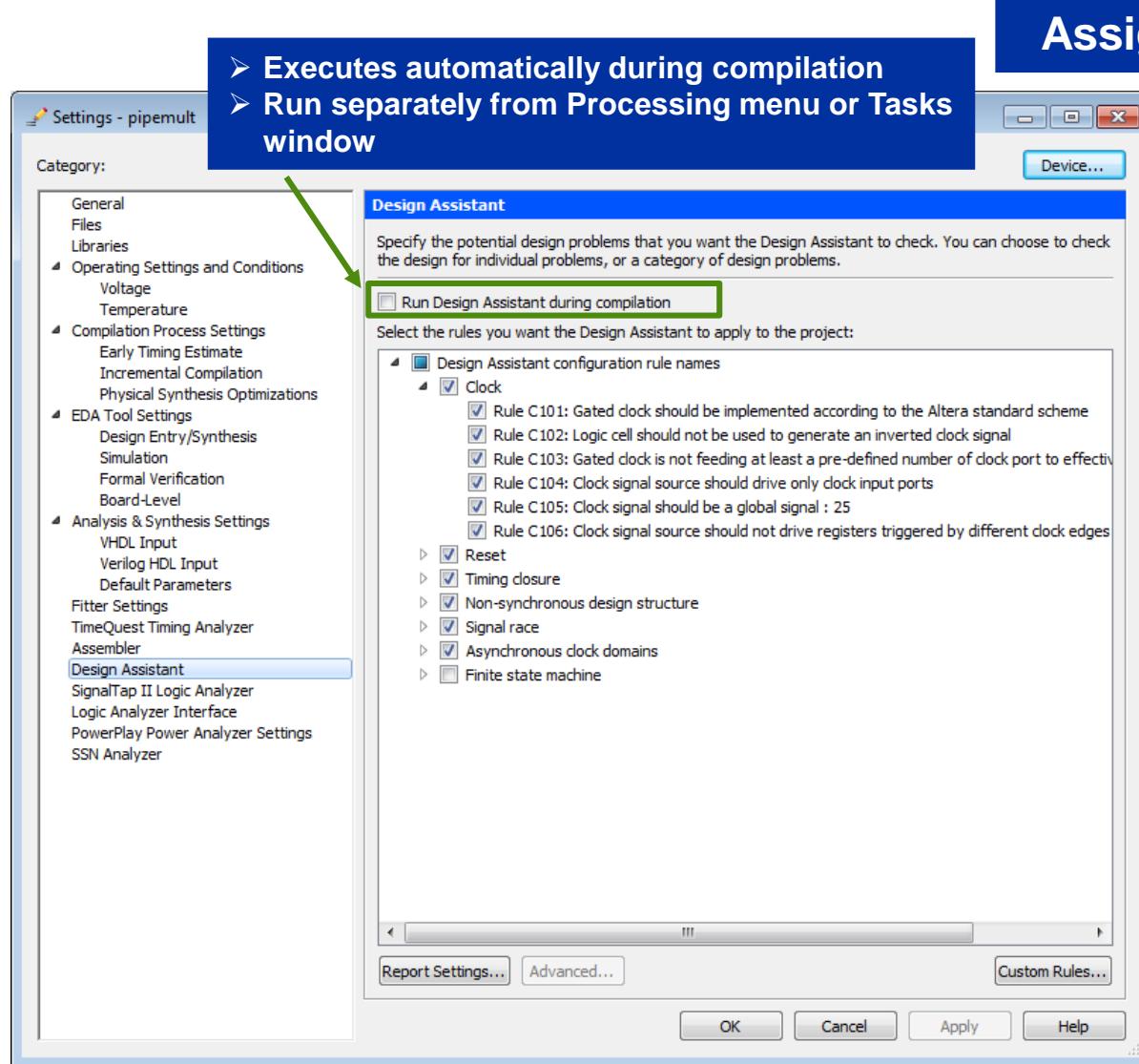
■ Design Assistant

- Check design against sets of good design practice rules

■ Advisors

- Check current settings and assignments
- Make recommendations for different optimization goals

Design Assistant



Assignments → Settings

Potential design issues

- Clocks
- Reset
- Non-synchronous design structure
- Timing closure
- Asynchronous clock domain data transfers
- Signal race conditions
- FSM

Settings & Assignments Summary

- **Settings & assignments allow a designer to control how a design is synthesized and placed & routed**
- **Use the Settings dialog box to adjust project-wide settings**
- **Use the Assignment Editor to enable/disable individual assignments targeting hierarchical blocks, internal nodes, or I/O**
- **Design Assistant & Optimization Advisors help improve design results through design rule checking and settings recommendations**

Class Agenda

- **Quartus II Projects**
- **Design Entry**
- **Quartus II Compilation**
- **Settings & Assignments**
- **I/O Planning**
- **Programming/Configuration**

Quartus II软件设计:基础

I/O Planning



© 2013 Altera Corporation—Confidential



I/O Planning Agenda

- **Creating device I/O location and related assignments**
- **Analysis**
 - Compiler process
 - Live I/O checking

Creating I/O-Related Assignments

- Pin Planner
- Import from spreadsheet in CSV format
- Type directly into QSF file
- Scripting

Note: Other methods/tools are available in the Quartus II software to make I/O assignments. The above are the most common or recommended.

Pin Planner

■ Interactive graphical tool for assigning pins

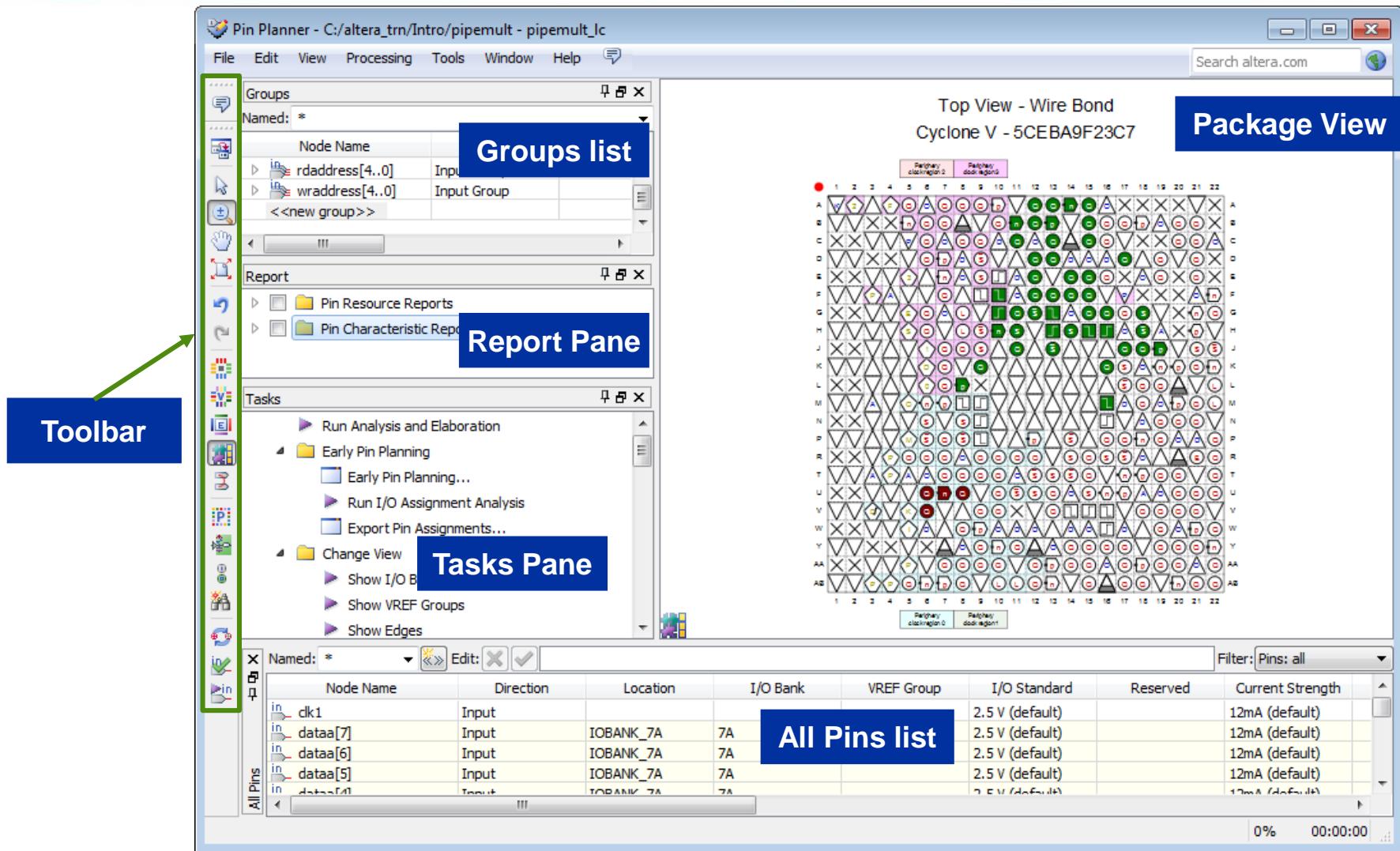
- Drag & drop pin assignments
- Set pin I/O standards
- Reserve future I/O locations

■ Default window panes

- Package View
- All Pins list
- Groups list
- Tasks window
- Report window

Assignments menu →
Pin Planner or
“Assign Constraints”
folder in Tasks window

Pin Planner Window



Pin Planner Window Panes

■ Package View

- Displays graphical representation of chip package
- Locate, make, or edit I/O assignments

■ All Pins list

- Displays I/O pins (signals) in design
- Edit pin assignments

■ Groups list

- Similar to All Pins list displaying only groups & buses
- Make bus and group assignments
- Create new user-defined groups

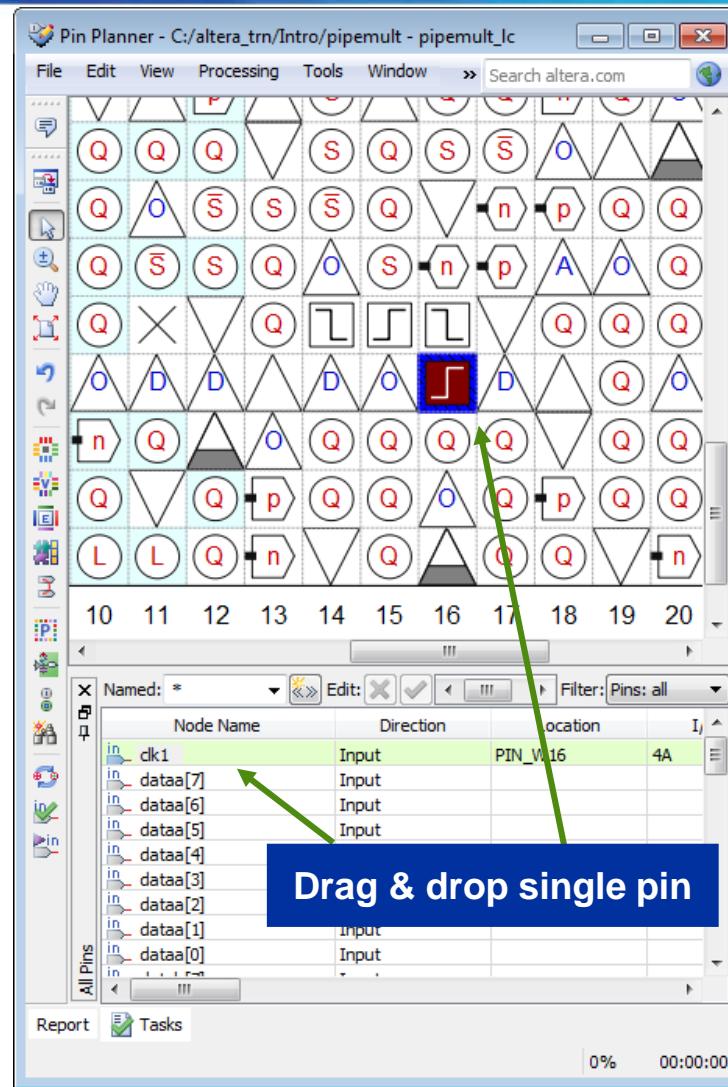
■ Tasks Pane

- Perform tasks such as Early Pin Planning and view changes

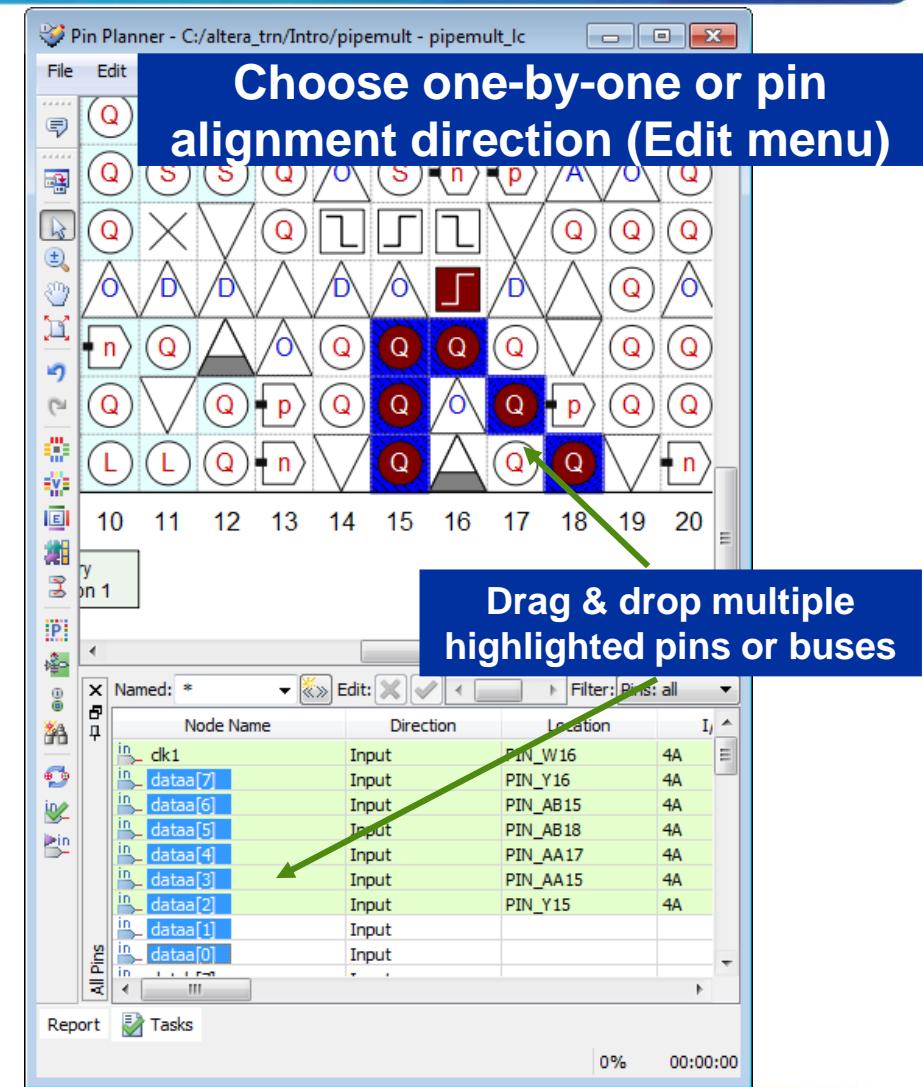
■ Report Pane

- Quickly enable/disable views generated by some view change tasks

Assigning Pin Locations Using Pin Planner



Drag & drop single pin



Choose one-by-one or pin alignment direction (Edit menu)

Drag & drop multiple highlighted pins or buses

Assigning Pin Locations Using Pin Planner (2)

The screenshot shows the Pin Planner interface for an Altera device. The main workspace displays a grid of pins numbered 3 to 19. A specific pin, PIN_B17, is highlighted with a purple dashed box and a callout pointing to the 'Pin Properties' panel on the right. The 'Pin Properties' panel shows details for PIN_B17, including its I/O bank (7A), VREF group (B7A_N0), and edge (TOP). The 'All Pins' table at the bottom lists various nodes and their assigned pins, with a filter set to 'Pins: all'. A blue callout box with the text 'Drag & drop to I/O bank, VREF block or device edge' is positioned over the pin grid, indicating the placement of pins. Another blue callout box with the text 'Filter nodes displayed in All Pins list' is positioned over the 'All Pins' table.

Click pin or I/O bank to display pin properties

Drag & drop to I/O bank, VREF block or device edge

Filter nodes displayed in All Pins list

Node Name	Direction	Location	I/O Bank	VREF Group	I/O Standard	Reserved	Current Strength	Slew Rate	Differential
in_clk1	Input	PIN_W16	4A	B4A_N0	2.5 V (default)		12mA (default)		
in_dataa[7]	Input		7A		2.5 V (default)		12mA (default)		
in_dataa[6]	Input		7A		2.5 V (default)		12mA (default)		
in_dataa[5]	Input		7A		2.5 V (default)		12mA (default)		
in_dataa[4]	Input		7A		2.5 V (default)		12mA (default)		
in_dataa[3]	Input		7A		2.5 V (default)		12mA (default)		
in_dataa[2]	Input		7A		2.5 V (default)		12mA (default)		
in_dataa[1]	Input		7A		2.5 V (default)		12mA (default)		
in_dataa[0]	Input		7A		2.5 V (default)		12mA (default)		

Assigning Pin Locations Using Pin Planner (3)

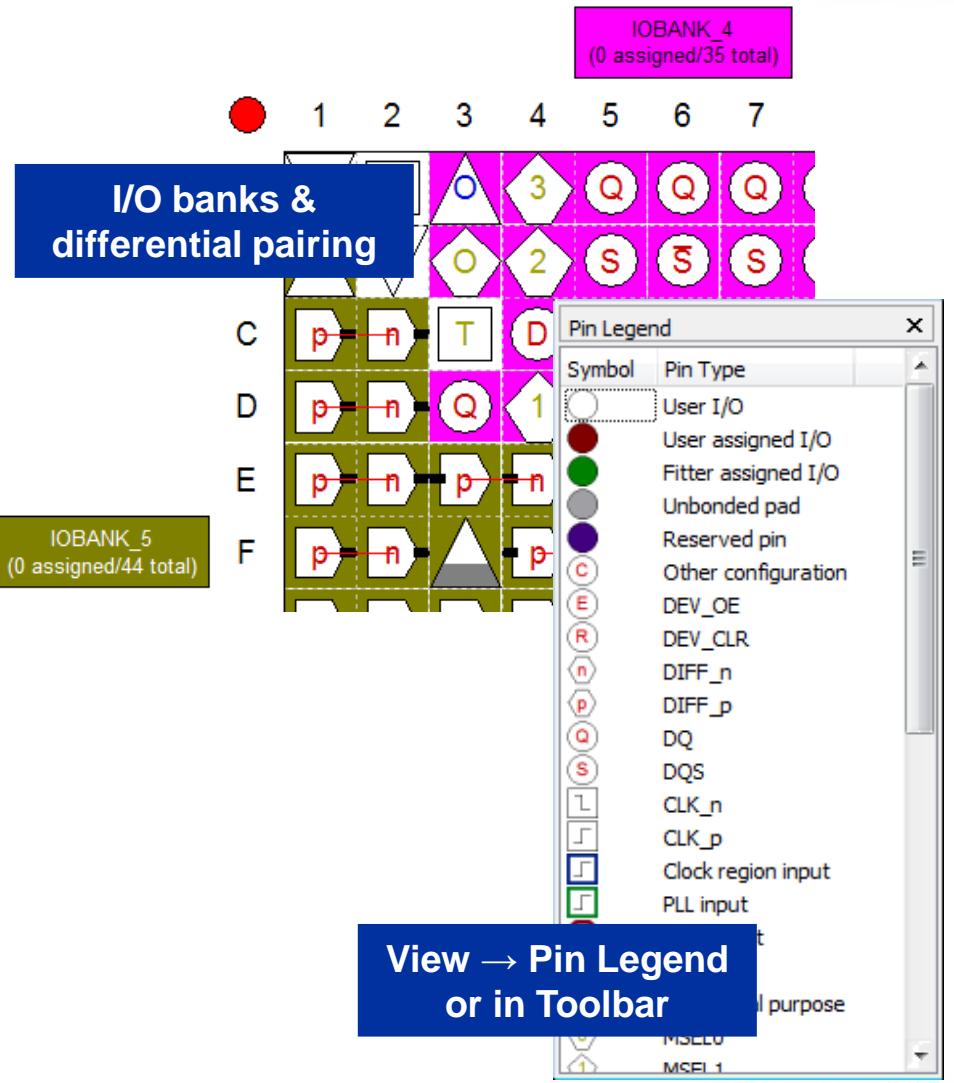
- Select available locations from list of pins color-coded by I/O bank

Node Name	Direction	Location	I/O Bank	VREF Group	I/O Standard	Reserved	Current Strength	Slew Rate	Differential
in_dk1	Input	PIN_W16	4A	B4A_N0	2.5 V (default)		12mA (default)		
in_dataa[7]	Input	PIN_AA13							
in_dataa[6]	Input	PIN_AA20	IOBANK_4A	Column I/O	DIFFIO_TX_B61p, DIFFOUT_B61p, DQ8B				
in_dataa[5]	Input	PIN_AA22	IOBANK_4A	Column I/O	DIFFIO_TX_B64p, DIFFOUT_B64p, DQ8B				
in_dataa[4]	Input	PIN_AB5	IOBANK_3B	Column I/O	DIFFIO_TX_B33p, DIFFOUT_B33p, DQ5B				
in_dataa[3]	Input	PIN_AB6	IOBANK_3B	Column I/O	DIFFIO_TX_B33n, DIFFOUT_B33n				
in_dataa[2]	Input	PIN_AB7	IOBANK_3B	Column I/O	DIFFIO_TX_B36p, DIFFOUT_B36p				
in_dataa[1]	Input	PIN_AB8	IOBANK_3B	Column I/O	DIFFIO_TX_B37p, DIFFOUT_B37p, DQ5B				
in_dataa[0]	Input	PIN_AB10	IOBANK_3B	Column I/O	FPLL_BL_CLKOUT1, FPLL_BL_CLKOUTn, DIFFIO_TX_B45n, DIFFOUT_B45n, DQ6B				
		PIN_AB11	IOBANK_3B	Column I/O	FPLL_BL_CLKOUT0, FPLL_BL_CLKOUTp, FPLL_BL_FB, DIFFIO_TX_B45p, DIFFOUT_B45p, DQ6B				
		PIN_AB12	IOBANK_4A	Column I/O	DIFFIO_TX_B49p, DIFFOUT_B49p, DQ7B				
		PIN_AB13	IOBANK_4A	Column I/O	RZQ_0, DIFFIO_TX_B49n, DIFFOUT_B49n				

Other Pin Planner Features

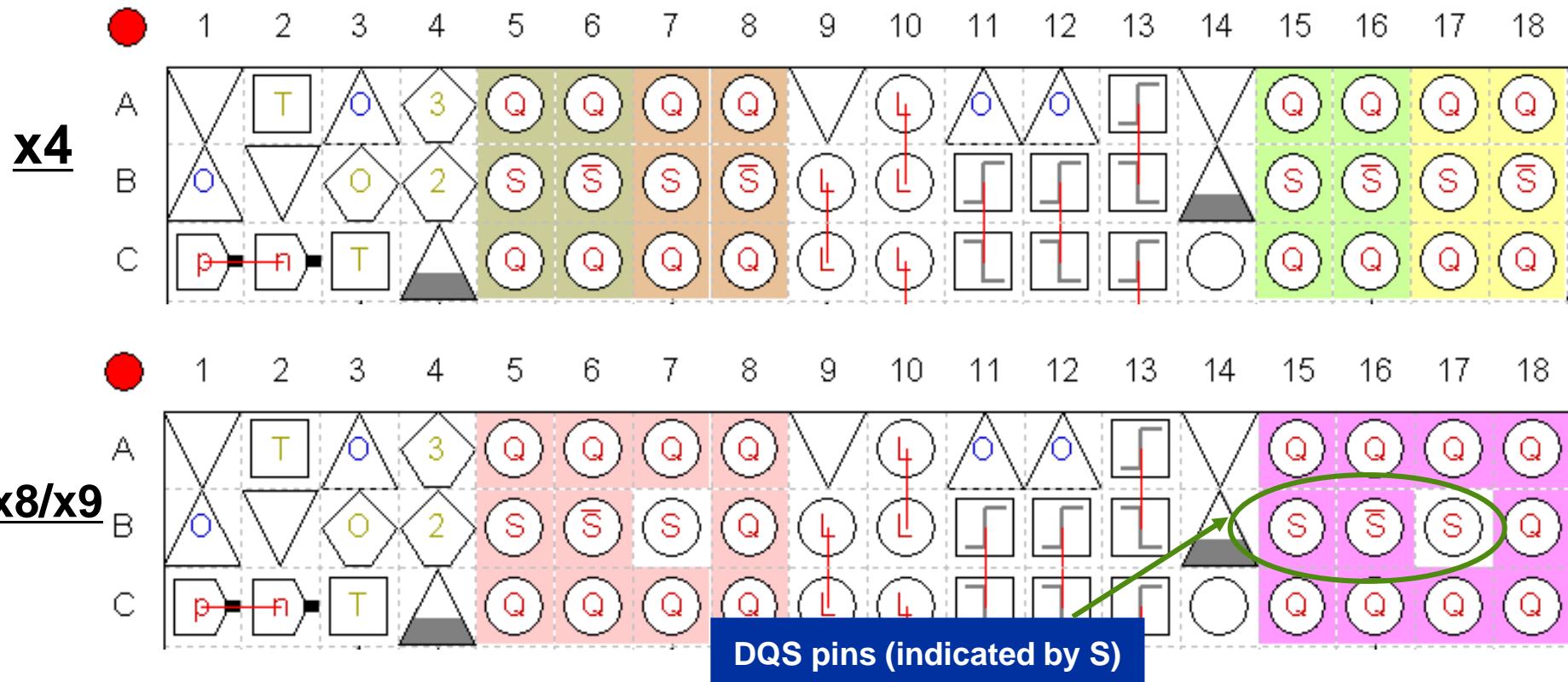
- **Displays (View ⇒ Show, Toolbar buttons, or right-click in Package View)**

- Device edges
- I/O banks
- VREF groups
- Differential pin pairing
- DQ/DQS pins



Additional View: Show DQ/DQS Pins

- Show color-coded DQ/DQS sets in x4, x8/x9, x16/x18, or x32/x36 modes in the Package View for DDR interfaces



Pin Migration View

- Select migration devices in Device Settings
- View & compare pin function differences between migration devices
- Package View adjusts to prevent non-migratable assignments

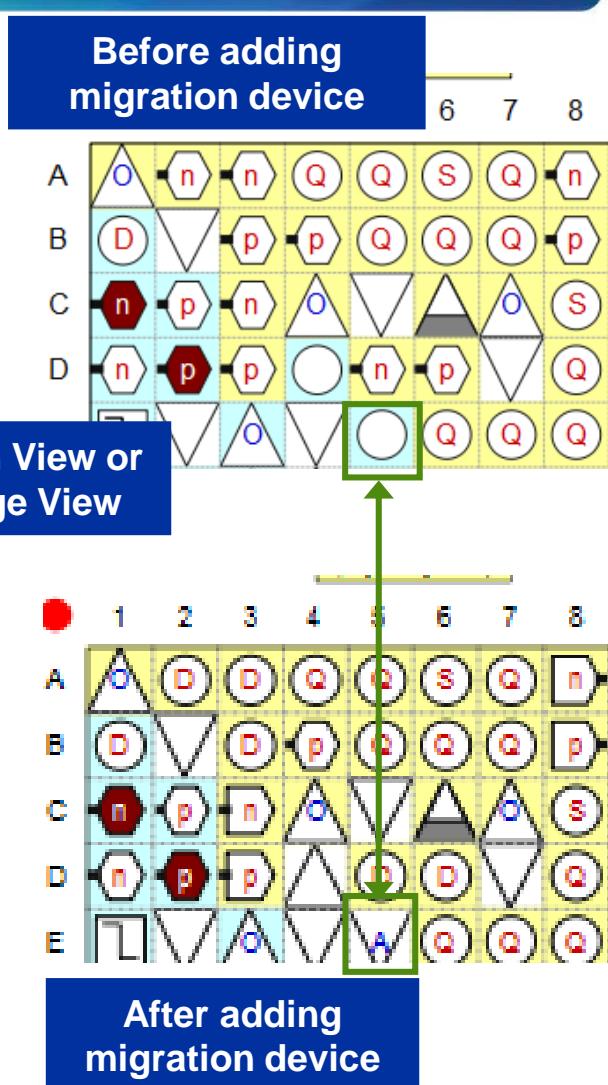
View → Pin Migration View or right-click in Package View

Pin Migration View

Current Device: EP4CE6F17C6

Pin Number	Migration Result			Migration Devices					
	Pin Function	I/O Bank	VREF Group	EP4CE6F17C6			EP4CE22F17C6		
				Pin Function	I/O Bank	VREF Group	Pin Function	I/O Bank	VREF Group
PIN_F6	GND			Column I/O	8	B8_N0	GND		
PIN_E5	VCCA3			Row I/O	1	B1_N0	VCCA3		
PIN_E5	GNDA3			Row I/O	1	B1_N0	GNDA3		
PIN_D4	VCCD_PLL3			Row I/O	1	B1_N0	VCCD_PLL3		
PIN_T8	Column I/O	3	B3_N0	Column I/O	3	B3_N0	Dedicated Clock	3	B3_N0
PIN_R8	Column I/O	3	B3_N0	Column I/O	3	B3_N0	Dedicated Clock	3	B3_N0
PIN_T9	Column I/O	4	B4_N0	Column I/O	4	B4_N0	Dedicated Clock	4	B4_N0
PIN_R9	Column I/O	4	B4_N0	Column I/O	4	B4_N0	Dedicated Clock	4	B4_N0
PIN_B9	Column I/O	7	B7_N0	Column I/O	7	B7_N0	Dedicated Clock	7	B7_N0
PIN_A9	Column I/O	7	B7_N0	Column I/O	7	B7_N0	Dedicated Clock	7	B7_N0
PIN_B8	Column I/O	8	B8_N0	Column I/O	8	B8_N0	Dedicated Clock	8	B8_N0
PIN_A8	Column I/O	8	B8_N0	Column I/O	8	B8_N0	Dedicated Clock	8	B8_N0

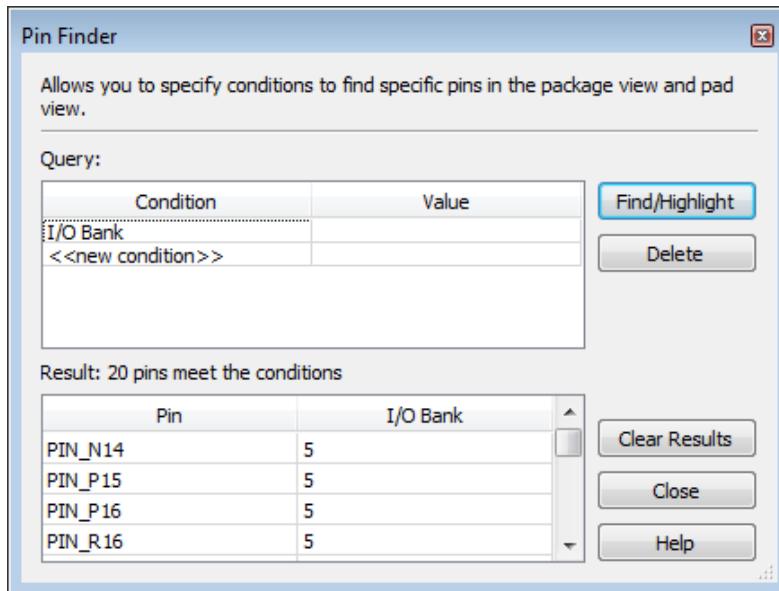
Device... Pin Finder... Show only highlighted pins Show migration differences Export... Help



More Pin Planner Features (1)

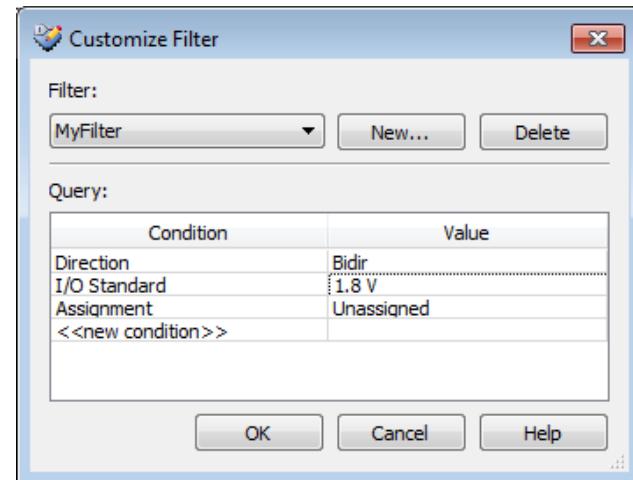
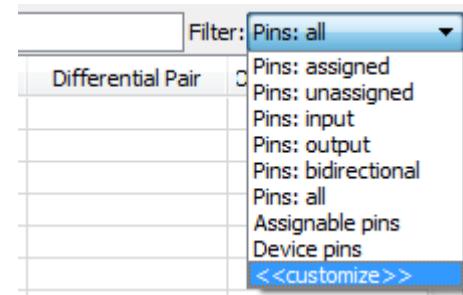
■ Pin Finder

- Locate pins meeting user-defined criteria with Pin Finder
- Use to find compatible pin locations
- Pins highlighted in Package View



■ Custom Filters

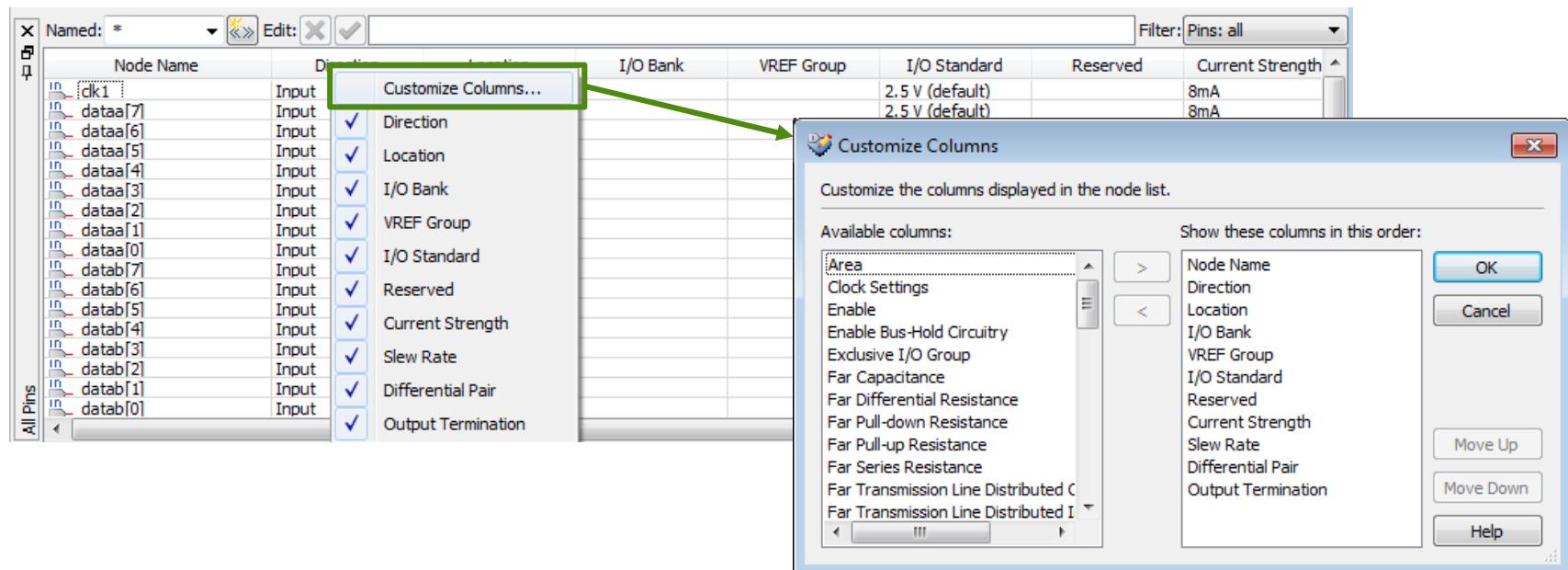
- Create custom filters for All Pins list



More Pin Planner Features (2)

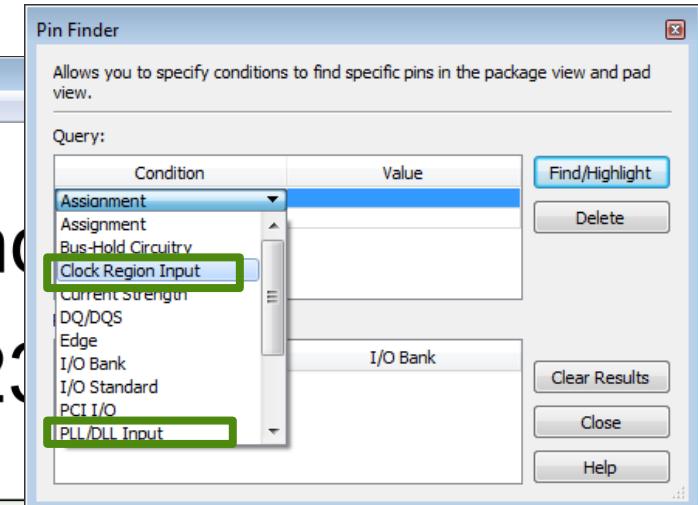
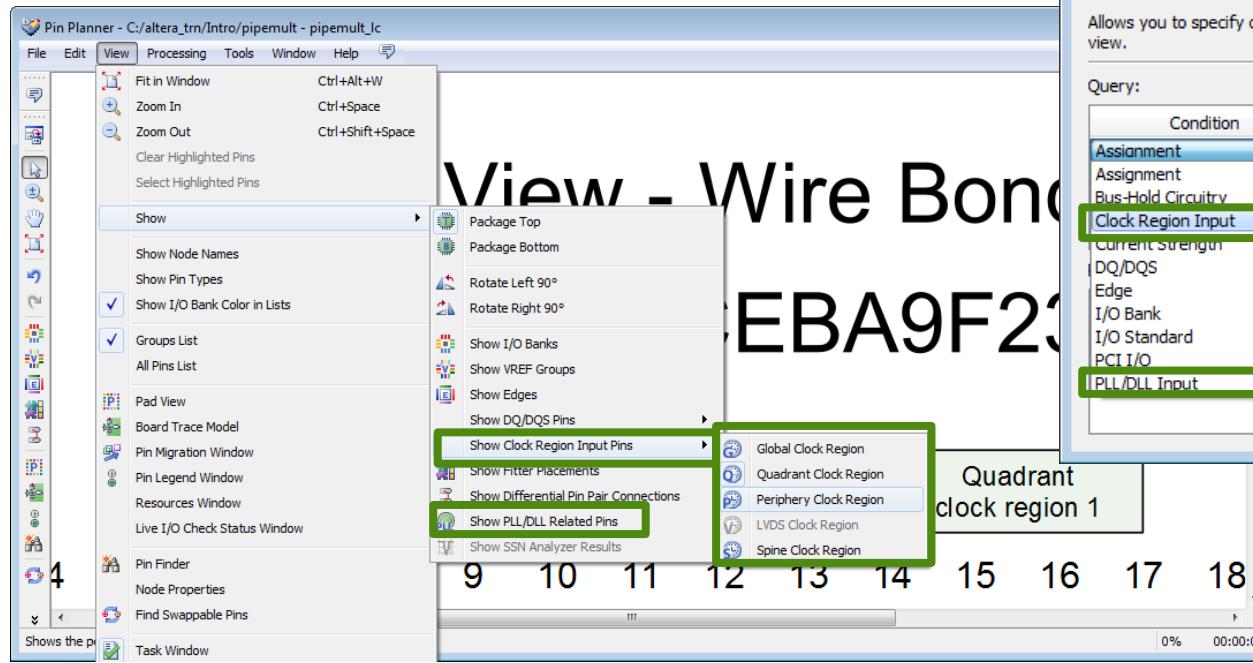
■ Customizable columns

- Select the I/O-related assignment columns to be shown in All Pins list for easy management
- Right click on column heading to select which columns will be displayed



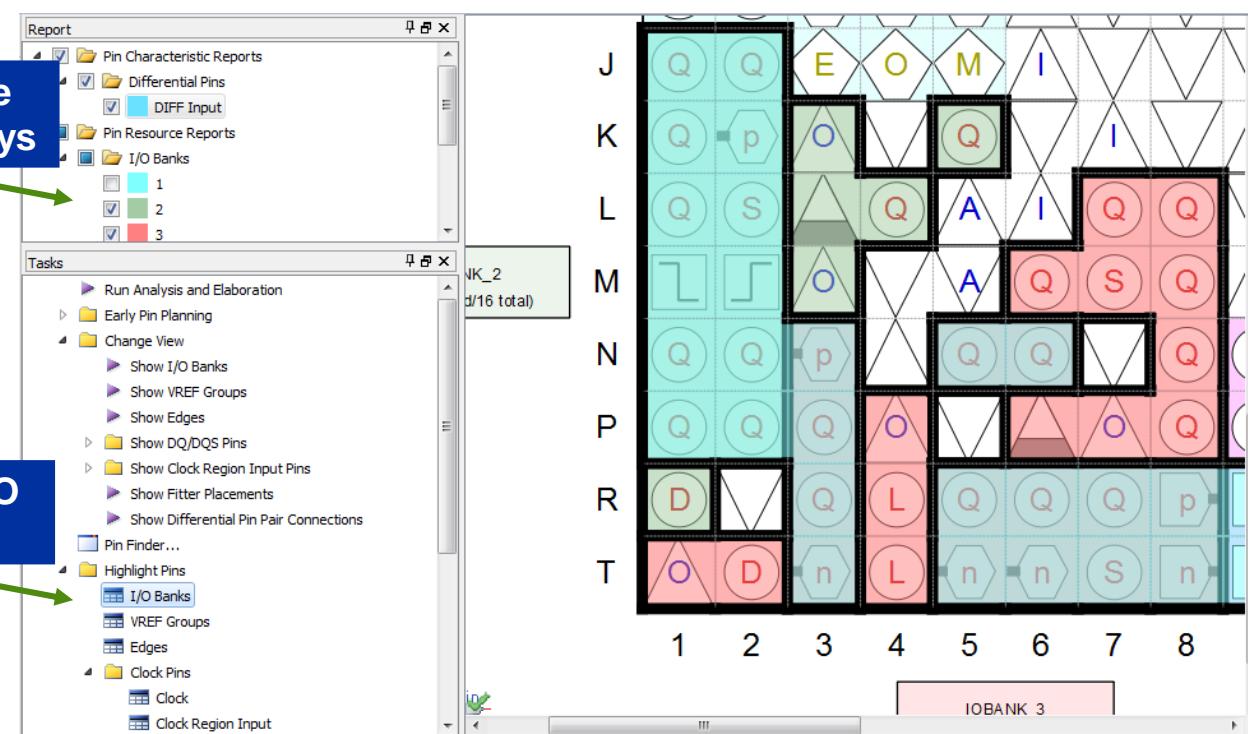
Pin Planner – Clock/PLL Support

- Show Clock Region Input Pins view
 - Shows the input pins in each clock region
- Show PLL/DLL related pins
- Updated Property windows to show clock region information
- Pin Finder allows searching of clock region input pins, PLL input/output pins



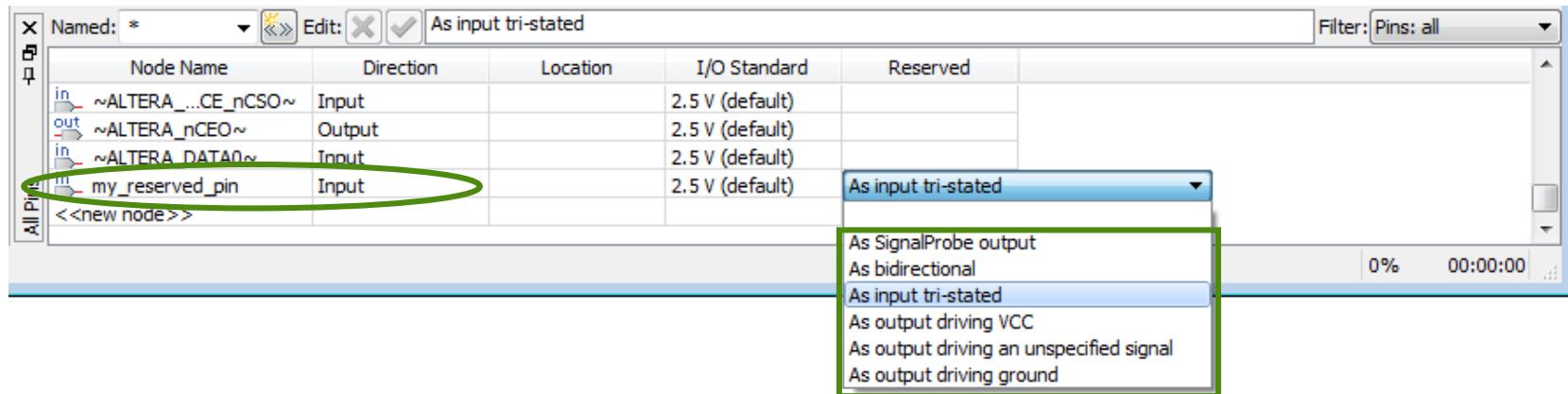
Pin Planner Tasks & Report Windows

- Use Tasks window to access common operations
- Change View folder of tasks changes display (same as View menu)
- Highlight Pins folder of tasks generates report overlays similar to Chip Planner
 - Access reports in Report pane



Reserved and Unused I/O Pins

- Type reserved I/O name into All Pins list & select reserve configuration
- Prevents Fitter from placing unassigned signal on pin (*discussed next*)

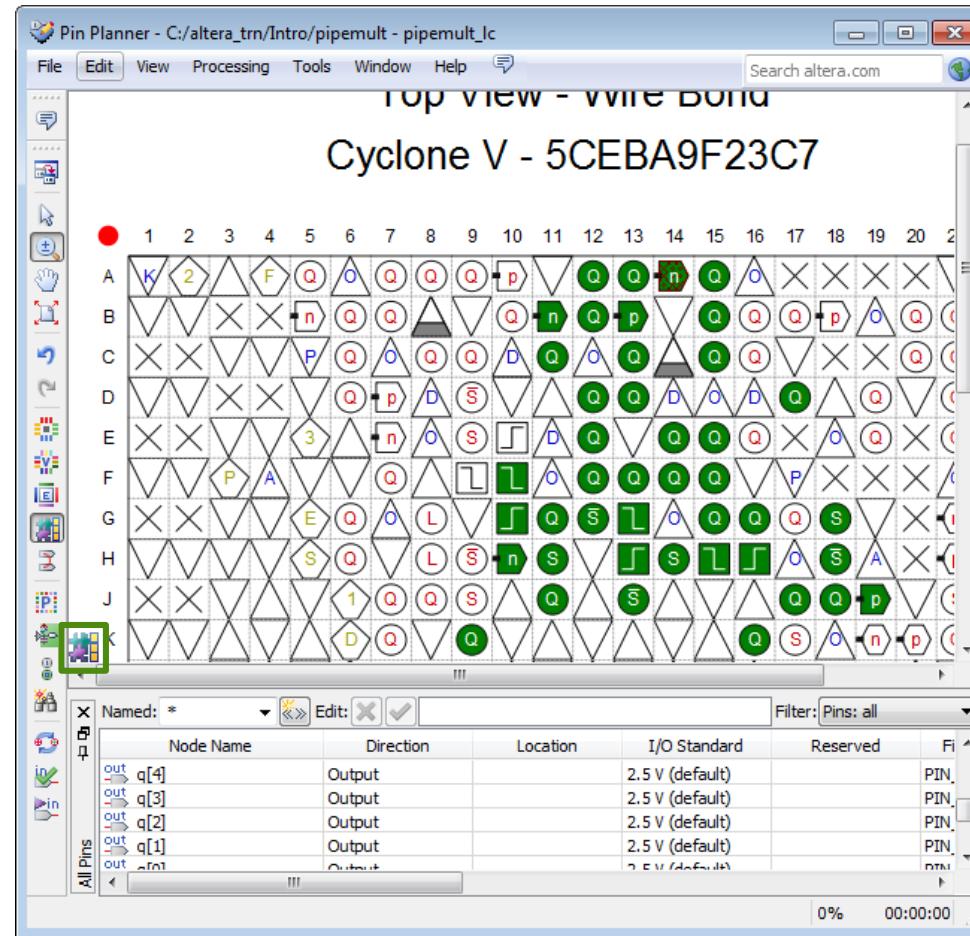


- Or right-click on pin in Package View and click Reserve Pins→ As...
 - Pin name set to *user_reserve_<pin_number>*
- Set initial state of other unused pins in Device settings in Settings dialog box (see *previous slide*)

Show Fitter Placements

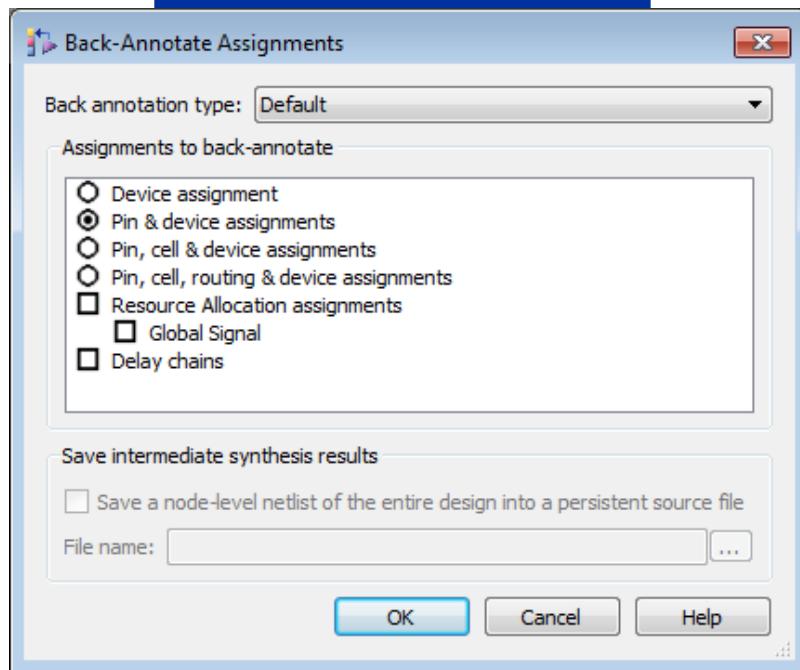
- View I/O locations selected by the Fitter

View → Show
or in Toolbar



Back-Annotation

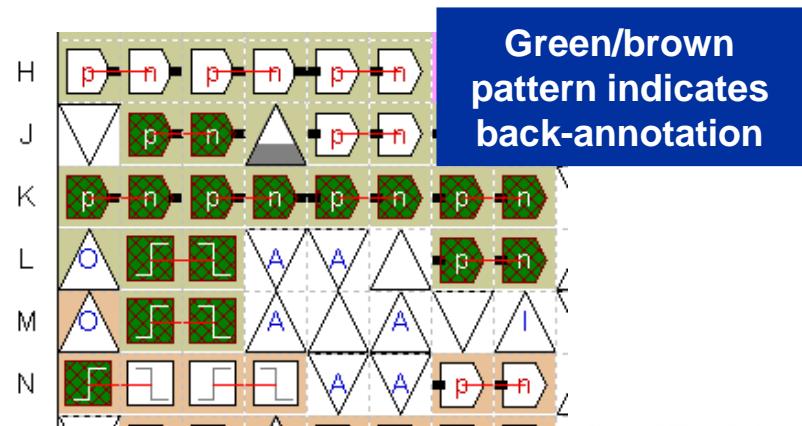
Assignments menu



■ Use to lock Fitter-chosen pin assignments for future compilations

- Copies device & resource locations chosen by fitter into QSF file
 - Pins
 - Logic
 - Routing

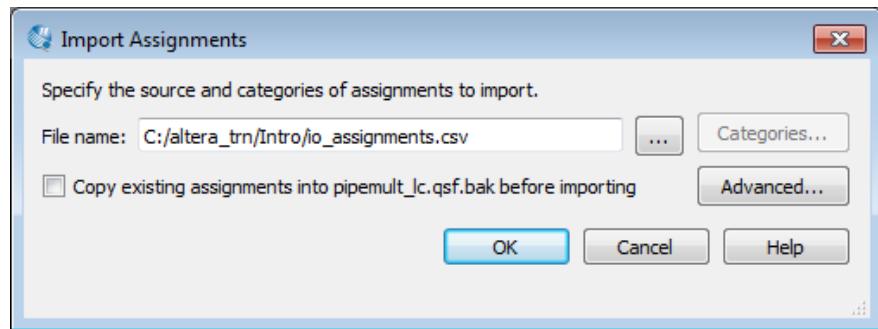
■ “Locks down” locations in Pin Planner



Other Methods: Import/Export via CSV

- Use spreadsheet Comma Separated Value (.CSV) file to enter or edit I/O locations
- Convenient for transferring assignments between project revisions
- CSV column names must match Pin Planner column headings

Pin Planner File menu or
Quartus II assignments menu



	A	B	C	D	E	F
1	To	Direction	Location	I/O Bank	VREF Group	I/O Standard
2	clk1	Input	PIN_E15	6	B6_N0	1.8 V
3	dataa[7]	Input	PIN_B14	7	B7_N0	2.5 V
4	dataa[6]	Input	PIN_A14	7	B7_N0	2.5 V
5	dataa[5]	Input	PIN_B13	7	B7_N0	2.5 V
6	dataa[4]	Input	PIN_A12	7	B7_N0	2.5 V

Other Methods: QSF Editing & Scripting

- Type pin-related assignments directly into QSF
- Type pin-related assignments into separate Tcl
 - Source Tcl file in project QSF
 - Execute Tcl file to write assignments into QSF

```
abc Text Editor - C:/altera_trn/Intro/pipemult - pipemult - [pipemult.qsf]*
```

```
File Edit View Project Processing Tools Window Help Search altera.com
```

```
20
21 # Altera recommends that you do not modify this file.
22 # file is updated automatically by the Quartus II soft
23 # and any changes you make may be lost or overwritten.
24
25 source "location_assignments.tcl"
26
27 set_global_assignment -name DEVICE EP3C5F256C6
28 set_global_assignment -name FAMILY "Cyclone III"
29 set_global_assignment -name TOP_LEVEL_ENTITY pipemult
30 set_global_assignment -name ORIGINAL_QUARTUS_VERSION 6
31 set_global_assignment -name PROJECT_CREATION_TIME_DATE
32 set_global_assignment -name LAST_QUARTUS_VERSION 12.1
33 set_global_assignment -name DEVICE_FILTER_PACKAGE FBGA
34 set_global_assignment -name DEVICE_FILTER_PIN_COUNT 25
```

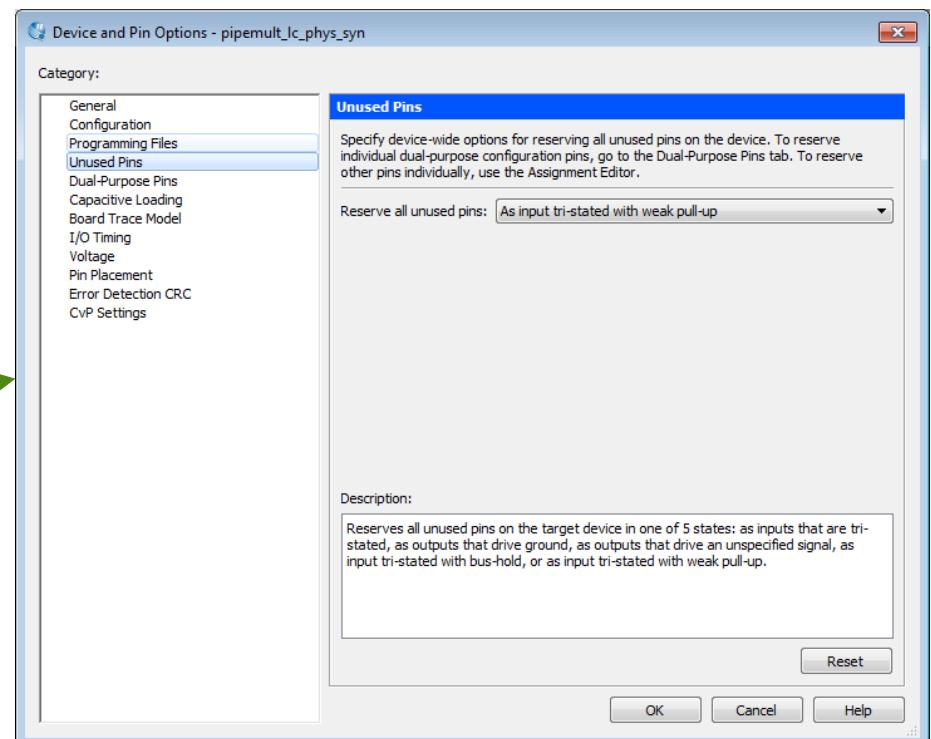
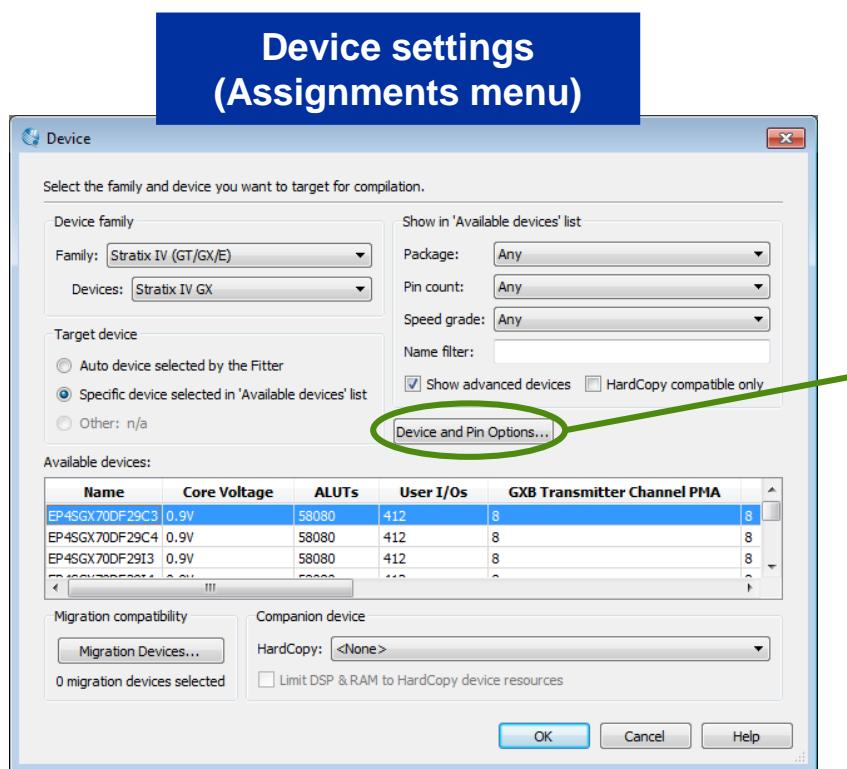
```
abc Text Editor - C:/altera_trn/Intro/pipemult - pipemult_lc - [location_assignments.tcl]
```

```
File Edit View Project Processing Tools Window Help Search altera.com
```

```
22 set_location_assignment PIN_A15 -to datab[6]
23 set_location_assignment PIN_D17 -to datab[5]
24 set_location_assignment PIN_J13 -to datab[4]
25 set_location_assignment PIN_M16 -to clk1
26 set_location_assignment PIN_H15 -to dataa[7]
27 set_location_assignment PIN_J17 -to dataa[6]
28 set_location_assignment PIN_C15 -to dataa[5]
29 set_location_assignment PIN_F14 -to dataa[4]
30 set_location_assignment PIN_H16 -to dataa[3]
31 set_location_assignment PIN_B15 -to dataa[2]
32 set_location_assignment PIN_B13 -to dataa[1]
33 set_location_assignment PIN_G11 -to q[15]
34 set_instance_assignment -name IO_STANDARD "2.5 V" -to q[15]
35 set_location_assignment PIN_E15 -to q[14]
36 set_instance_assignment -name IO_STANDARD "2.5 V" -to q[14]
37 ----- PIN_E12 -- q[13]
```

Global Pin Settings

- Select global settings for pins
 - e.g. unused I/O configuration, dual-purpose configuration pins, device CRC checking
- Assignments in Pin Planner or other tools have precedence over global settings



Verifying I/O Assignments

■ I/O Assignment Analysis

- Checks legality of all I/O assignments without full compilation

■ Minimal requirements for running

- I/O declaration
 - HDL port declaration
 - Reserved pin
- Pin-related assignments
 - I/O standard
 - Current strength
 - Pin location (pin, bank, edge)
 - PCI clamping diode
 - Toggle rate



Processing menu →
Start → Start I/O
Assignment Analysis
or Tasks window

I/O Rules Checked

■ No internal logic

- Checks I/O locations & constraints with respect to other I/O & I/O banks
- e.g. Each I/O bank supports a single V_{CCIO}

■ I/O connected to logic

- Checks I/O locations & constraints with respect to other I/O, I/O banks, & internal resources
- e.g. PLL must be driven by a dedicated clock input pin

Note: When working with design files, synthesize design before running I/O Assignment Analysis

I/O Assignment Analysis Output

Compilation Report (Fitter section)

- Pin-out file
- I/O pin tables
- I/O rules checking

The screenshot shows the Quartus II 64-Bit interface with the project 'C:/altera_trn/Intro/pipemult - pipemult_lc'. The 'Compilation Report - pipemult_lc' window is open. On the left, the 'Table of Contents' sidebar has a green box around the 'I/O Rules Section' under 'Incremental Compilation Section'. The main area shows the 'I/O Rules Matrix' table with 26 rows (q[0] to q[25]) and 7 columns (Pin/Rules, IO_000001 to IO_000005, IC). Most entries are 'Pass' or 'Inapplicable'. A blue box on the right contains text about messages on I/O assignment issues.

Pin/Rules	IO_000001	IO_000002	IO_000003	IO_000004	IO_000005	IC
1 Total Pass	45	0	45	0	0	45
2 Total Unchecked	0	0	0	0	0	0
3 Total Inapplicable	0	45	0	45	45	0
4 Total Fail	0	0	0	0	0	0
5 q[15]	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pa
6 q[14]	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pa
7 q[13]	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pa
8 q[12]	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pa
9 q[11]	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pa
10 q[10]	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pa
11 q[9]	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pa
12 q[8]	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pa
13 q[7]	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pa
14 q[6]	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pa
15 q[5]	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pa
16 q[4]	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pa
17 q[3]	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pa
18 q[2]	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pa
19 q[1]	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pa
20 q[0]	Pass					
21 clk1	Pass					
22 reset	Pass					
23 wren	Pass					
24 wraddress[0]	Pass					
25 wraddress[1]	Pass					
26 wraddress[2]	Pass					

Messages on I/O assignment Issues

- Compiler assumptions
- Device & pin migration issues
- I/O bank voltages & standards

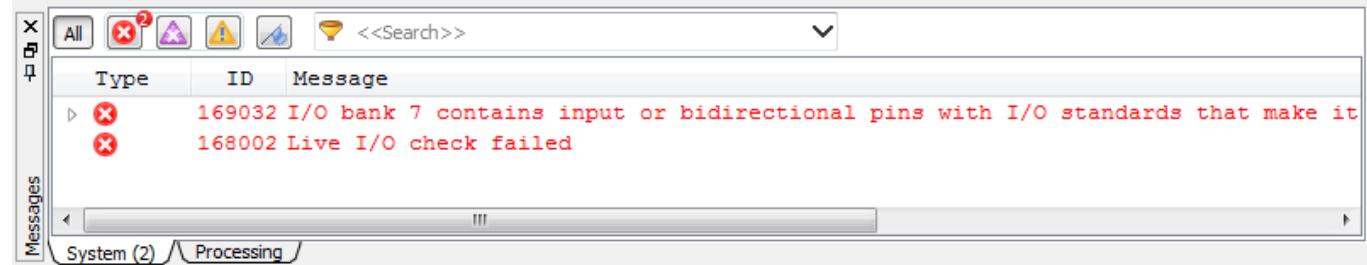
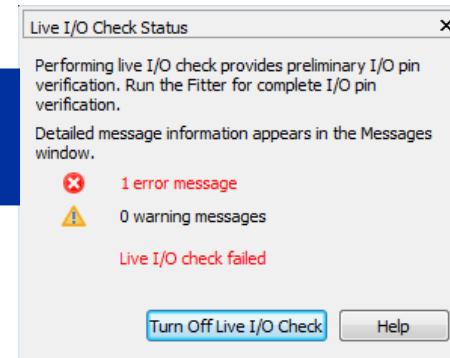
Live I/O Checking

- Perform limited I/O checks as assignments are made
- Status window alerts to failing assignments
 - Errors detailed in Messages window and Live I/O Check Status window
- Full I/O Assignment Analysis still required

Turn on or off
in Pin Planner
toolbar



In Pin Planner, View
menu → Live I/O
Check Status Window

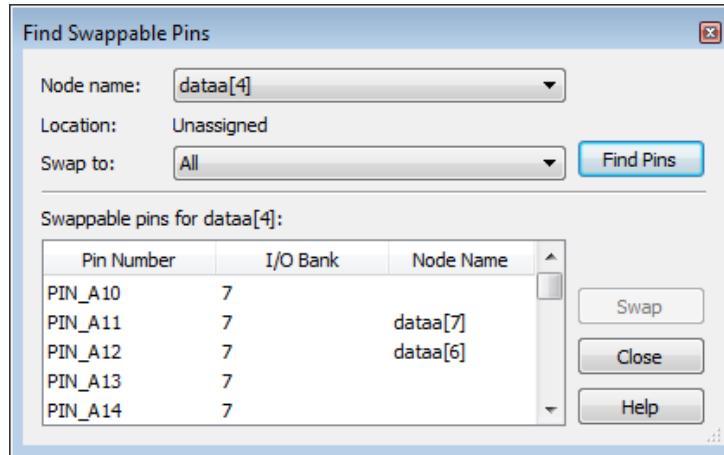


Swappable Pins



- Find compatible pins for swapping locations
- Works directly with Live I/O Check
 - Live I/O Check must be turned on
- Useful for last minute I/O location changes

View menu → Find Swappable Pins, Toolbar, or right-click menu



Selected pin
highlighted in
Package view

Validating I/O Pin-out

■ Completed design

- Run full compilation
- Enable option to **Run I/O Assignment Analysis before compilation** (**Settings** dialog box → **Compilation Process** category)

■ Incomplete design with completed top-level design file

- Run I/O Assignment Analysis on design

■ Incomplete or no design files

- Use early I/O planning methodology (see [I/O System Design](#) online training)

I/O Planning Summary

- Pin assignments can be performed in many ways, graphically & by means of text files
- The Pin Planner provides an easy-to-use graphical method of creating and managing pin assignments
- I/O Assignment Analysis helps validate a device pin-out without performing a full compilation
- Live I/O checking runs a subset of checks as assignments are made
- Pin validation can be completed during any point in design development

Class Agenda

- **Quartus II Projects**
- **Design Entry**
- **Quartus II Compilation**
- **Settings & Assignments**
- **I/O Management**
- **Programming/Configuration**

Quartus II软件设计:基础

Programming & Configuration



© 2013 Altera Corporation—Confidential

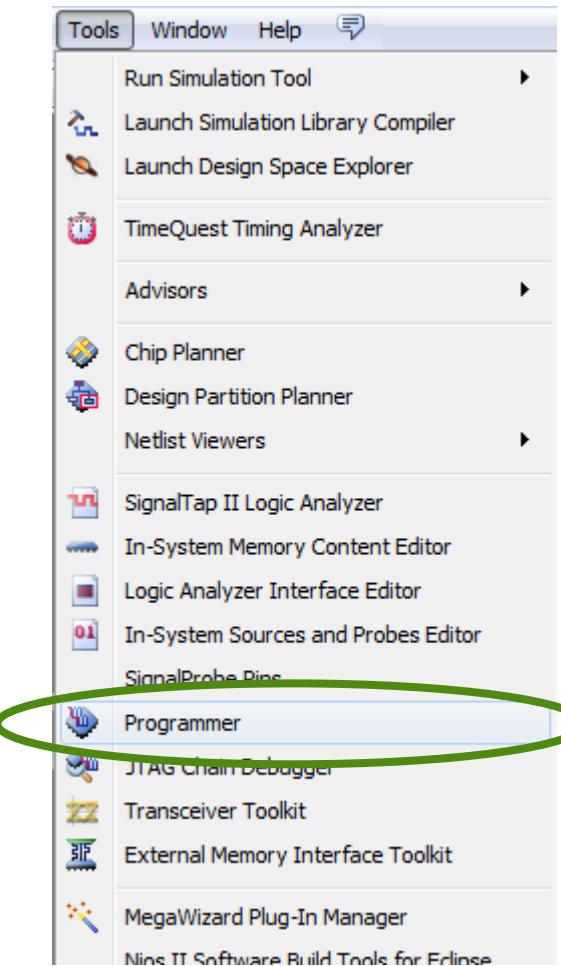


Programming Section Objectives

- Define the difference between the types of programming files generated by the Quartus II software
- Program a device
- Convert from one type of programming file to another

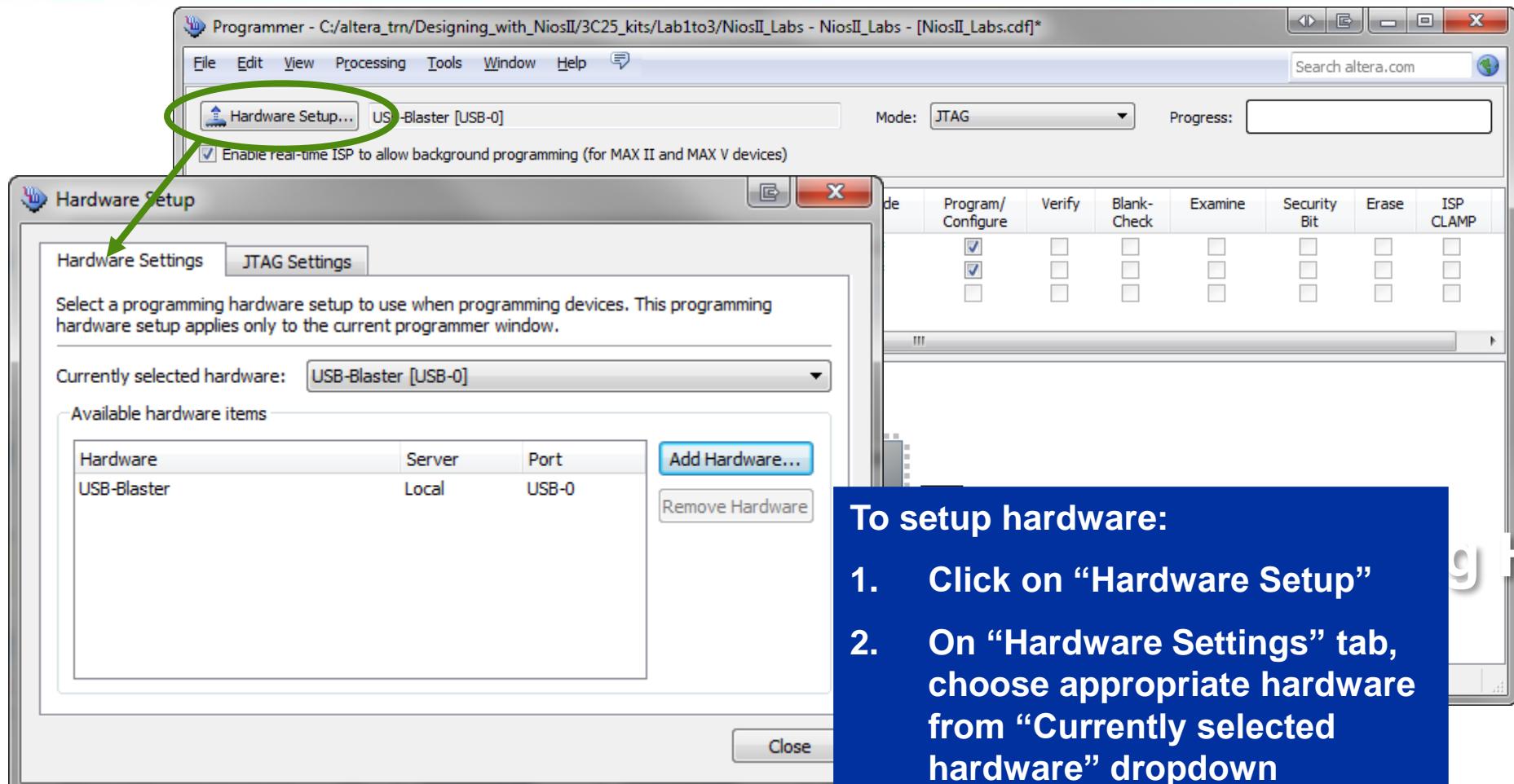
Quartus II Programmer

■ Use to configure Altera devices in-system



- Click on Programmer Icon on the toolbar
or
- Tools menu→ Programmer





To setup hardware:

1. Click on “Hardware Setup”
2. On “Hardware Settings” tab, choose appropriate hardware from “Currently selected hardware” dropdown

Note: Add hardware as needed

Supported Programming Files

■ **.SOF (SRAM Object File)**

- Used to configure FPGAs directly from Quartus II software through download cable

■ **.POF (Programming Object File)**

- Used to program CPLDs and configuration devices

■ **.JIC (JTAG Indirect Configuration File)**

- Used to program EPICS (Altera serial configuration) devices through their dedicated configuration interface with FPGAs

JTAG Chain

Programmer - C:/altera_trn/Designing_with_NiosII/3C25_kits/Lab1to3/NiosII_Labs - NiosII_Labs - [NiosII_Labs.cdf]*

File Edit View Processing Tools Window Help

Search altera.com

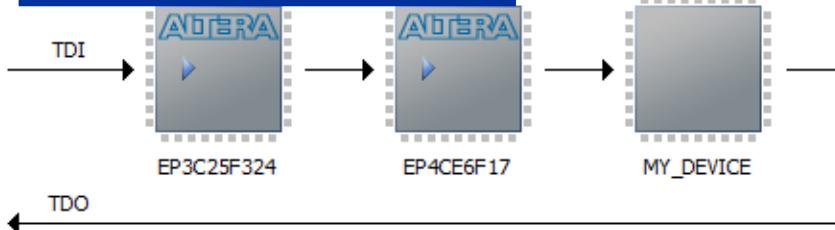
Hardware Setup... USB-Blaster [USB-0] Mode: JTAG Progress:

Enable real-time ISP to allow background programming (for MAX II and MAX V devices)

File	Device	Checksum	Usercode	Program/Configure	Verify	Blank-Check	Examine	Security Bit	Erase	ISP CLAMP
NiosII_Labs.sof	EP3C25F324	0053074A	FFFFFFFF	<input checked="" type="checkbox"/>	<input type="checkbox"/>					
.../Quartus_II_Softw...	EP4CE6F17	0008DCE8	FFFFFFFF	<input checked="" type="checkbox"/>	<input type="checkbox"/>					
<none>	MY_DEVICE	00000000	<none>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

When adding files, the device for that file is automatically chosen

Enable programming



Change the order of the device chain

JTAG Chain:

- Can consist of combination of Altera FPGAs, CPLDs and configuration devices as well as non-Altera (or user) devices*
- Devices in the chain will be programmed from top down

Programming Options

Programmer - C:/altera_trn/Designing_with_NiosII/3C25_kits/Lab1to3/NiosII_Labs - NiosII_Labs - [NiosII_Labs.cdf]*

File Edit View Processing Tools Window Help

Hardware Setup... USB-Blaster [USB-0] Mode: JTAG Progress:

Enable real-time ISP to allow background programming (for MAX II and MAX V devices)

File	Device	Checksum	Usercode	Program/Configure	Verify	Blank-Check	Examine	Security Bit	Erase	ISP CLAMP
NiosII_Labs.sof	EP3C25F324	0053074A	FFFFFFFF	<input checked="" type="checkbox"/>	<input type="checkbox"/>					
.../Quartus_II_Softw...	EP4CE6F17	0008DCE8	FFFFFFFF	<input checked="" type="checkbox"/>	<input type="checkbox"/>					
<none>	MY_DEVICE	00000000	<none>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

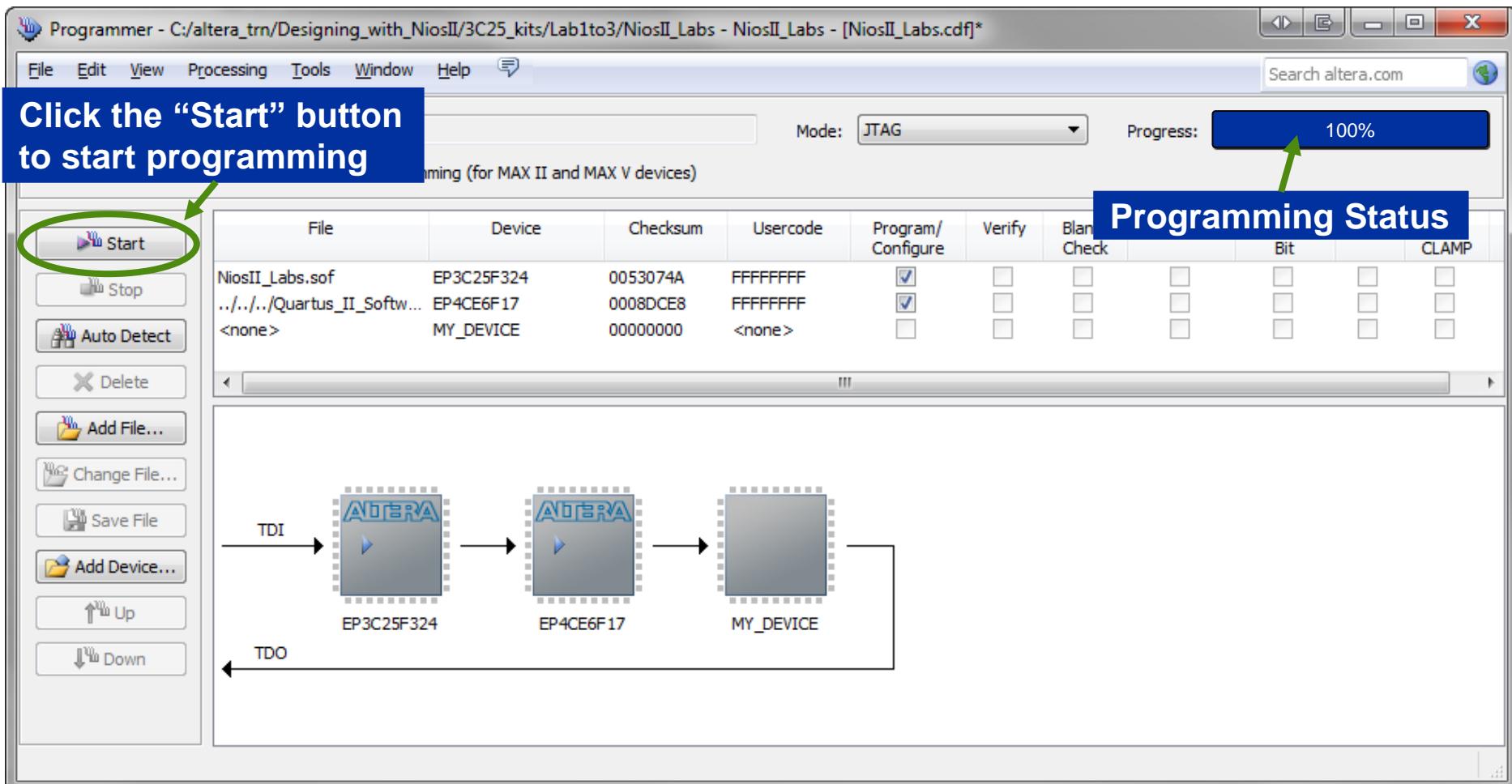
Start Stop Auto Detect Delete Add File... Change File... Save File

TDI →  →  EP4CE6F17

These options only apply to CPLDs or configuration devices.

Option	Description
Verify	Verifies contents of against respective programming files
Blank-Check	Check whether the device is blank
Examine	Reads back contents. Can save examine data as .pof file.
Security Bit	Protect CPLD from being examined.
Erase	Erase the contents of devices

Programming



Programming .jic File

Programmer - C:/Current_Training_Classes/Timing_Analysis/COS_062612/filter - filtref - [output_files/filtref.cdf]*

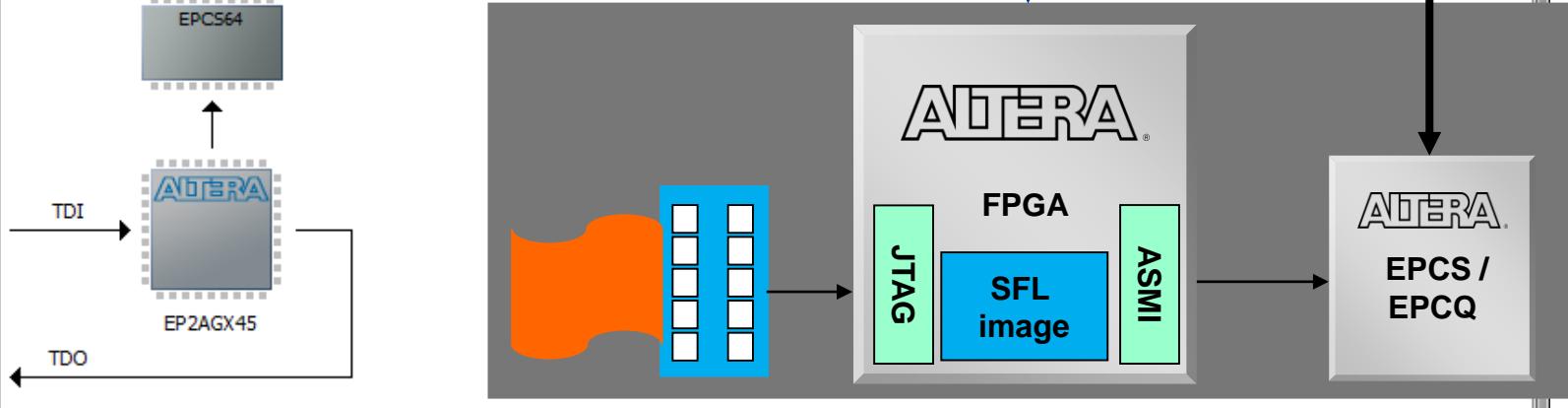
File Edit View Processing Tools Window Help Search altera.com

Hardware Setup... USB-Blaster [USB-0] Mode: JTAG Progress:

Enable real-time ISP to allow background programming (for MAX II and MAX V devices)

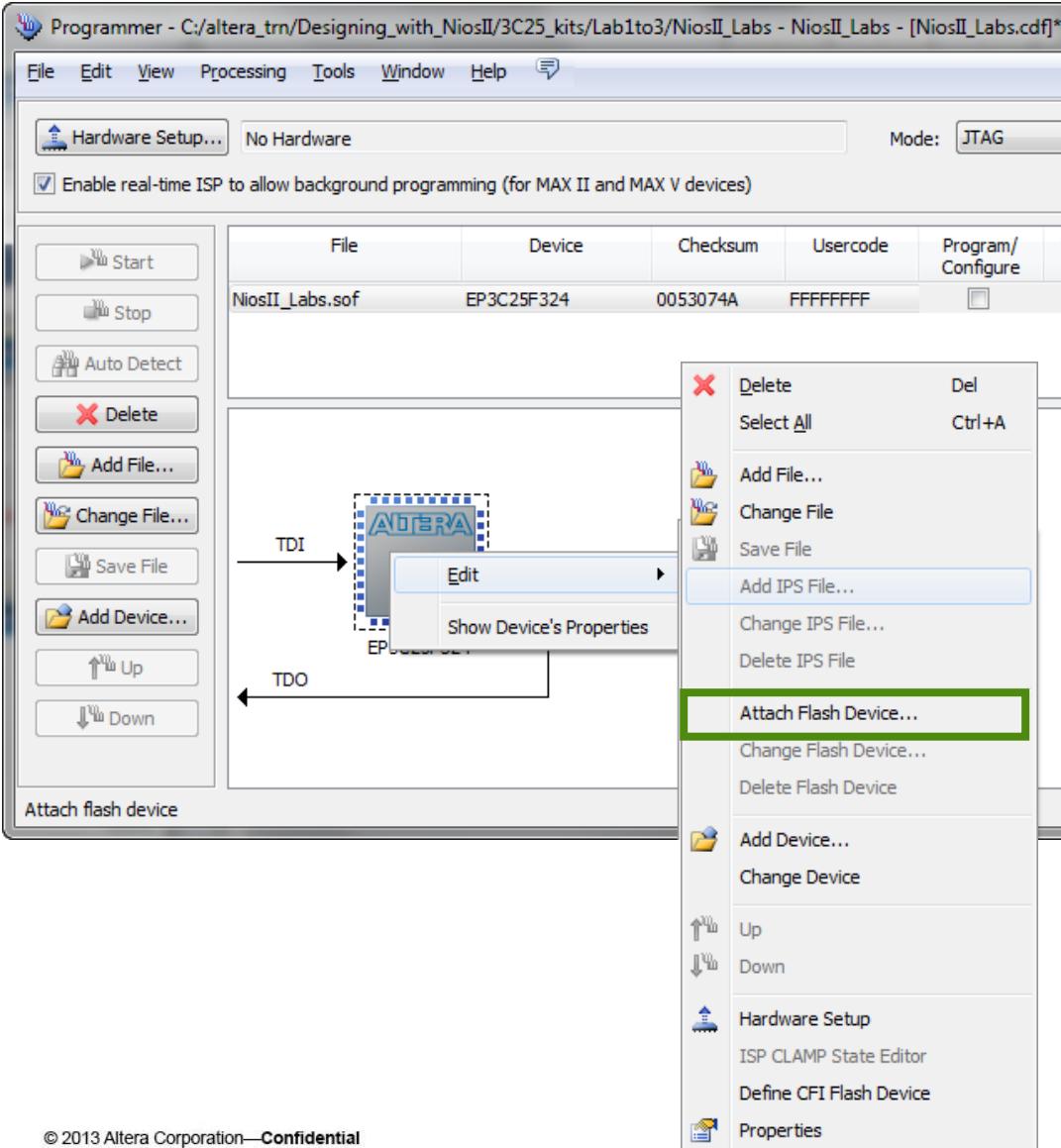
File	Device	Checksum	Usercode	Program/Configure	Verify	Blank-Check	Examine	Security Bit	Erase	ISP CLAMP
<none>	EP2AGX45	00000000	FFFFFF	<input type="checkbox"/>						
output_files/filtref.jic	EPCS64	49538A36		<input type="checkbox"/>						

Start Stop Auto Detect Delete Add File... Change File... Save File Add Device... Up Down



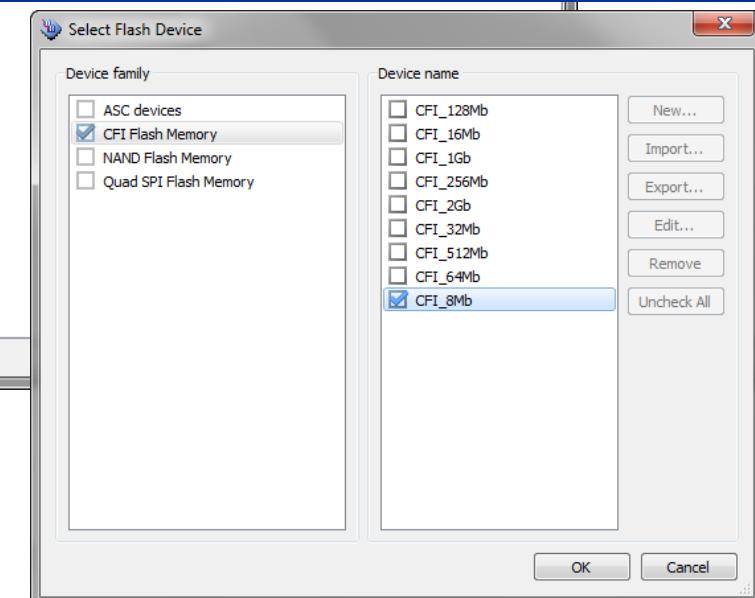
JIC files used to program EPCS configuration devices (which do not have JTAG pins)

Quartus II Programmer – Flash Programming



**Program flash using SFL, PFL or
Nios® II system:**

1. **Highlight FPGA/CPLD**
2. **Choose “Attach Flash Device”
(Right-click → Edit) or Edit
menu)**
3. **Select flash and OK**



ALTERA
MEASURABLE ADVANTAGE™

Quartus II Programmer – Flash Programming Example

The screenshot shows the Quartus II Programmer software interface. On the left, there's a toolbar with buttons for Start, Stop, Auto Detect, Delete, Add File..., Change File..., Save File, Add Device..., Up, and Down. The main window has a menu bar with File, Edit, View, Processing, Tools, Window, Help, and a search bar for altera.com. A status bar at the bottom shows 'C:/altera_trn/Designing_with_NiosII/3C25_kits/Lab1to3/NiosII_Labs - NiosII_Labs - [NiosII_Labs.cdf]*'. The central area displays a table with columns: File, Device, Checksum, Usercode, Program/Configure, Verify, Blank-Check, Examine, Security Bit, Erase, and ISP CLAI. Two rows are visible: 'NiosII_Labs.sof' (Device: EP3C25F324, Checksum: 0053074A, Usercode: FFFFFFFF) and 'NiosII_Labs.flash' (Device: CFI_8Mb, Checksum: 00000000). Below the table is a schematic diagram showing an EP3C25F324 device connected to a CFI_8Mb flash chip via TDI and TDO lines. A 'Flash Programming Properties...' dialog box is open in the foreground, showing 'Instance ID: 0' and 'Base address: 0x600000' with OK and Cancel buttons.

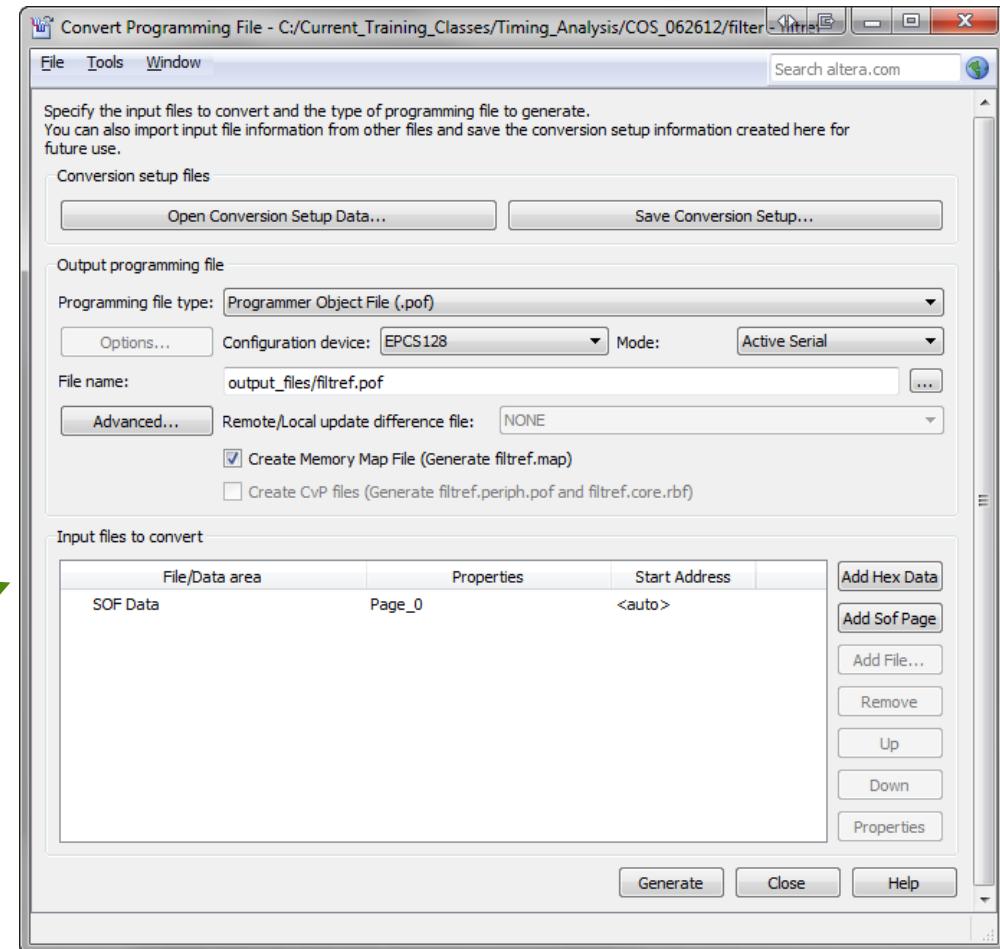
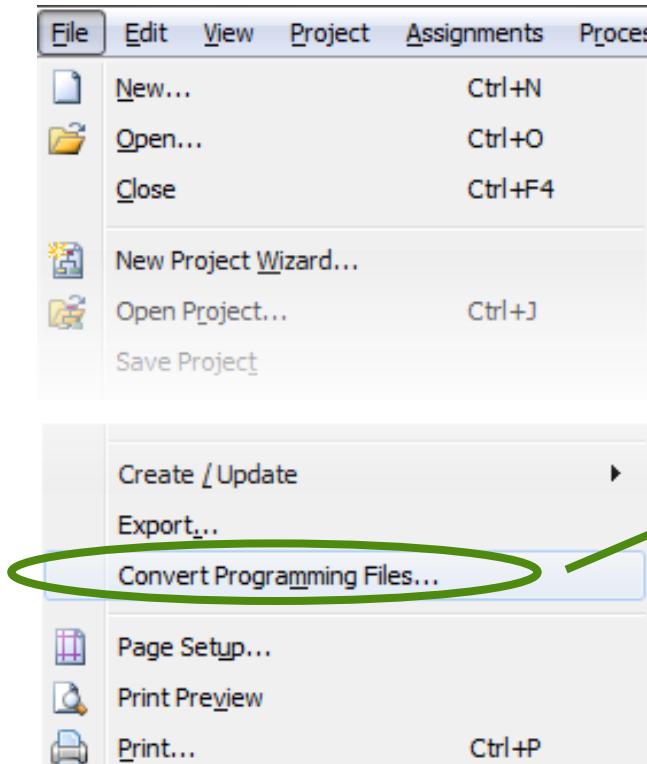
1. Program FPGA with Nios II system
2. Select flash file to target flash device
3. Program flash

Programming File Conversion

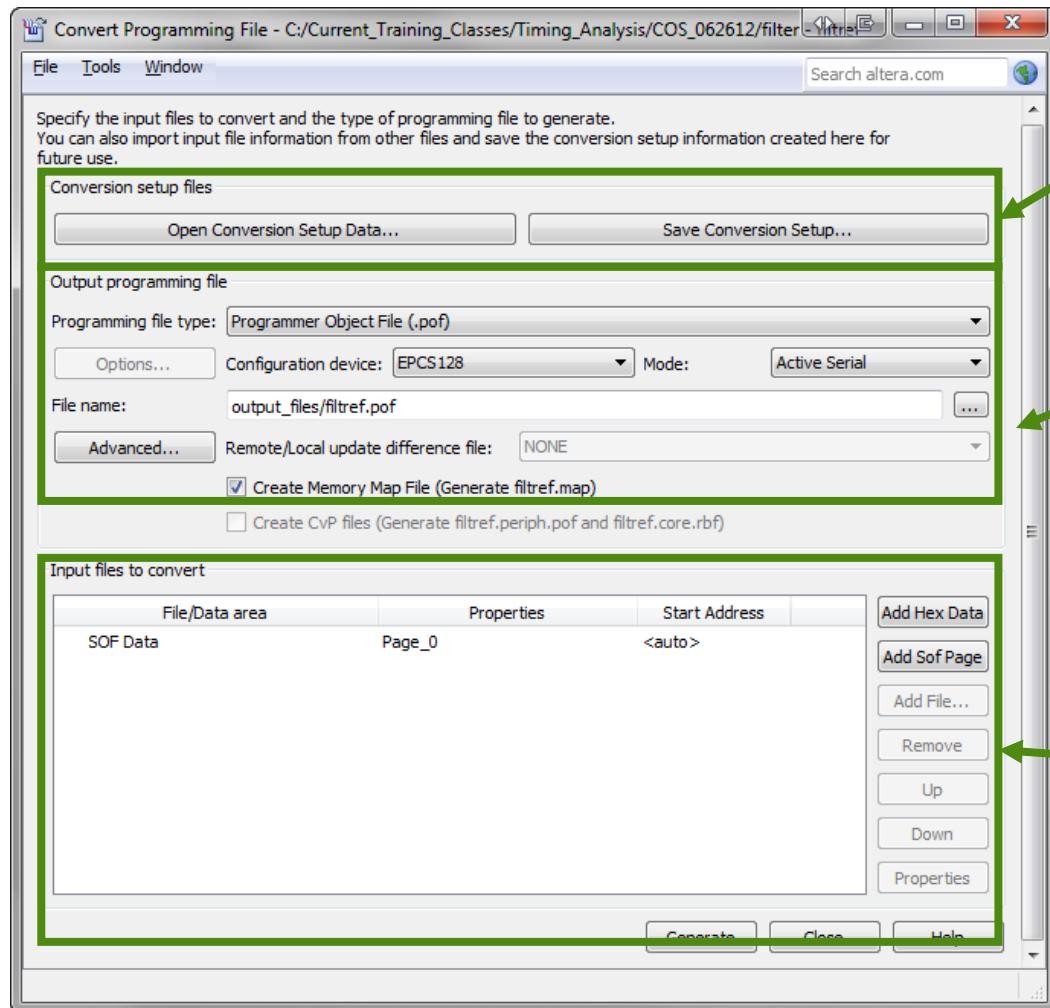
- **Quartus II Assembler automatically generates .SOF files for any FPGA**
 - May generate additional single-device programming file types from Device Settings dialog box (Assignments menu → Device → Device and Pin Options → Programming Files)
- **.SOF files can only be used by the Quartus II Programmer to directly configure FPGA**
- **Other configuration solutions require converting the .SOF file(s) into other file formats**
 - .JIC files
 - Multi-FPGA programming files

Converting Programming Files

Convert .sof and .pof files to other supported programming files



Convert Programming Files GUI



Load or save file conversion setup

Select an optional programming file format

Select a configuration device and its supported mode

Select more options for the configuration device

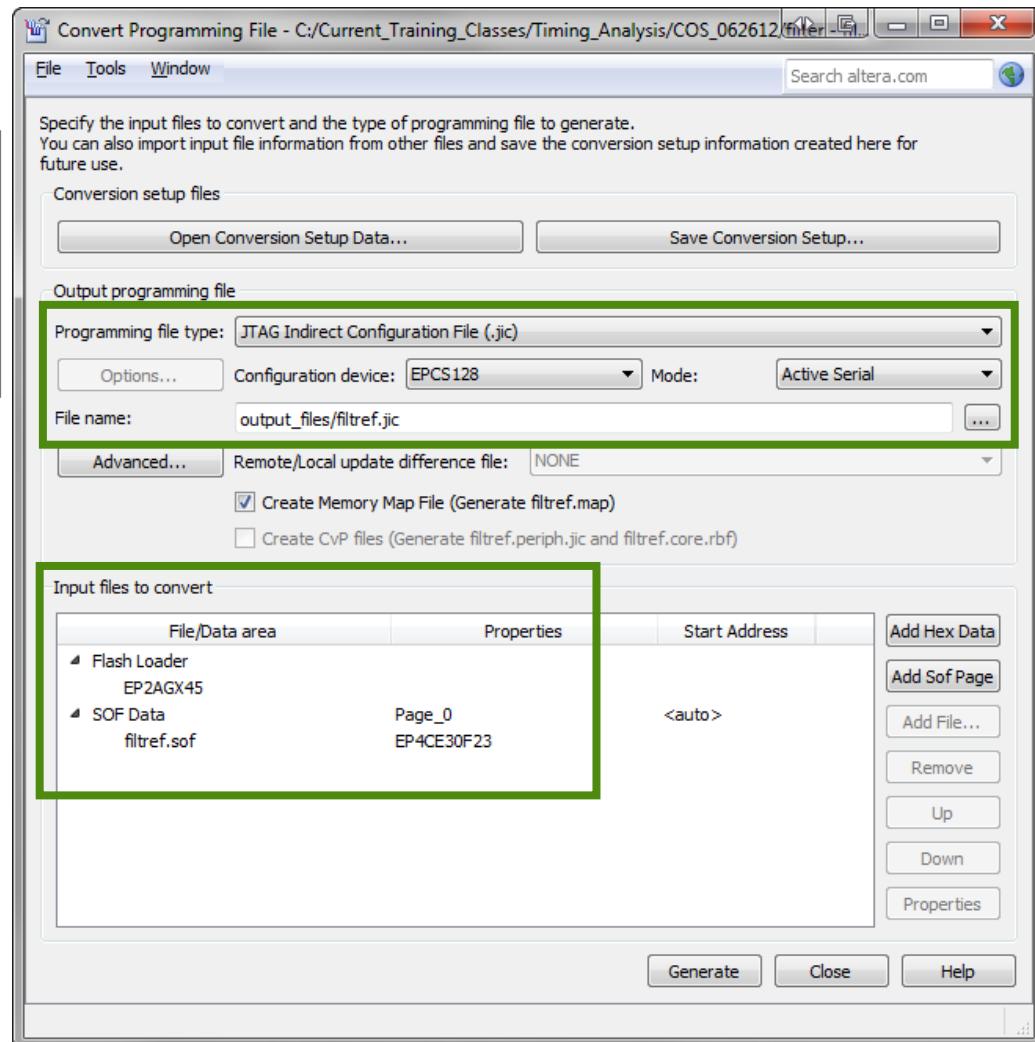
Add input configuring/programming file(s)

EPCS Programming - .jic generation



To create the .jic file:

1. Select .jic as output file type
2. Highlight Flash Loader
 - Click “Add Device”
 - Browse for FPGA
3. Click “Generate”



Summary

- Use the Quartus II Programmer to program and configure Altera devices for in-system testing
- Use the file conversion utility to generate additional programming file types and multi-FPGA programming files

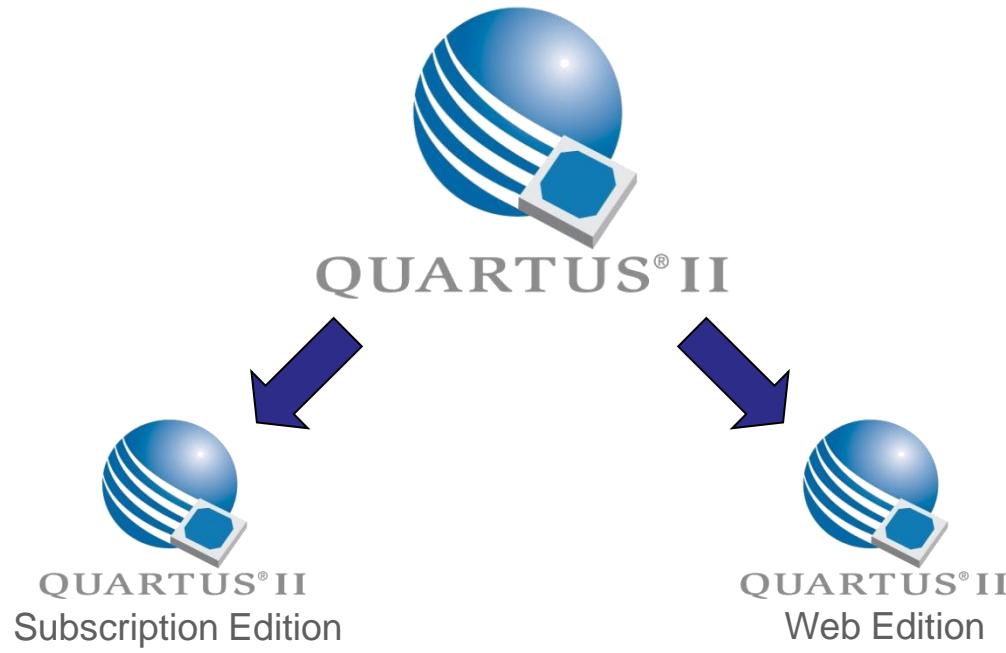
Appendix



© 2013 Altera Corporation—Confidential



Quartus II 软件 – 两个版本



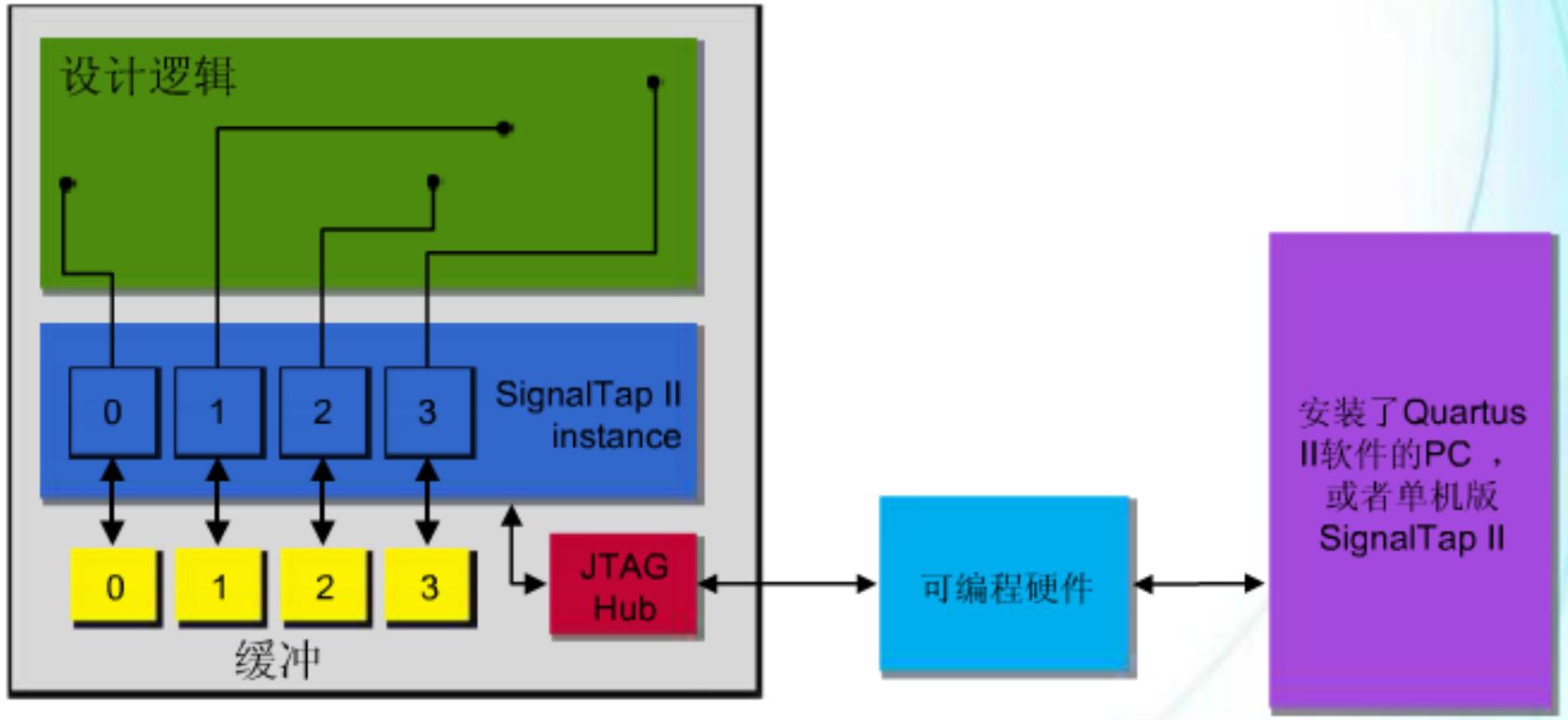
Devices Supported	All	Selected Devices
Features	100%	95%
Distribution	Internet & DVD	Internet & DVD
Price	Paid	Free

[Feature Comparison available on Altera web site](#)

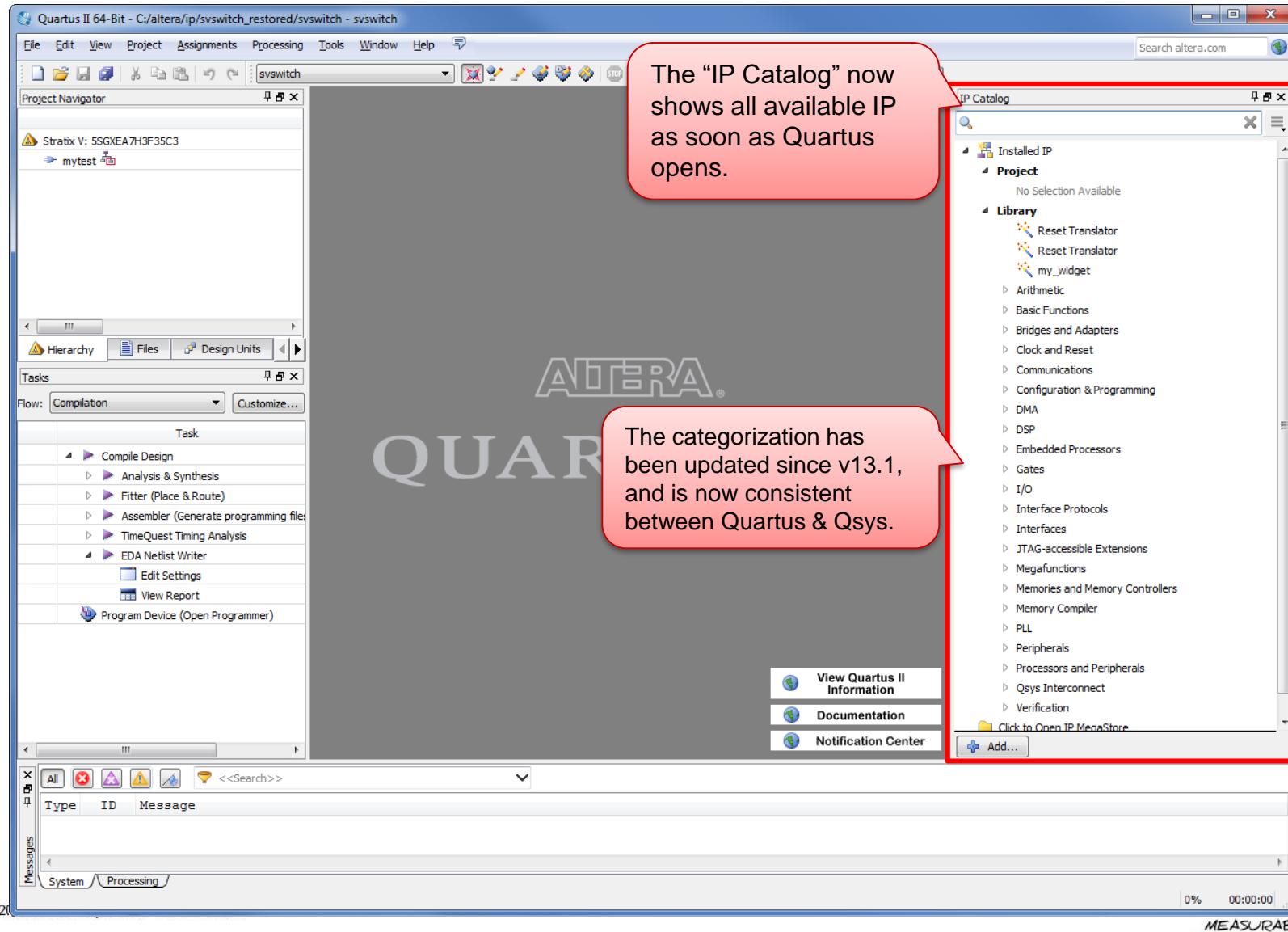
- Starting in 13.1, the following devices are not supported in Subscription Edition or Web Edition software
 - Stratix I / II / II GX
 - Arria I GX
 - Cyclone I / II
 - MAX7000/3000
 - All HardCopy ASIC device family
- Starting in 14.0, the following devices are not supported in Subscription Edition or Web Edition software
 - Stratix III
 - Cyclone III

Signal Tap

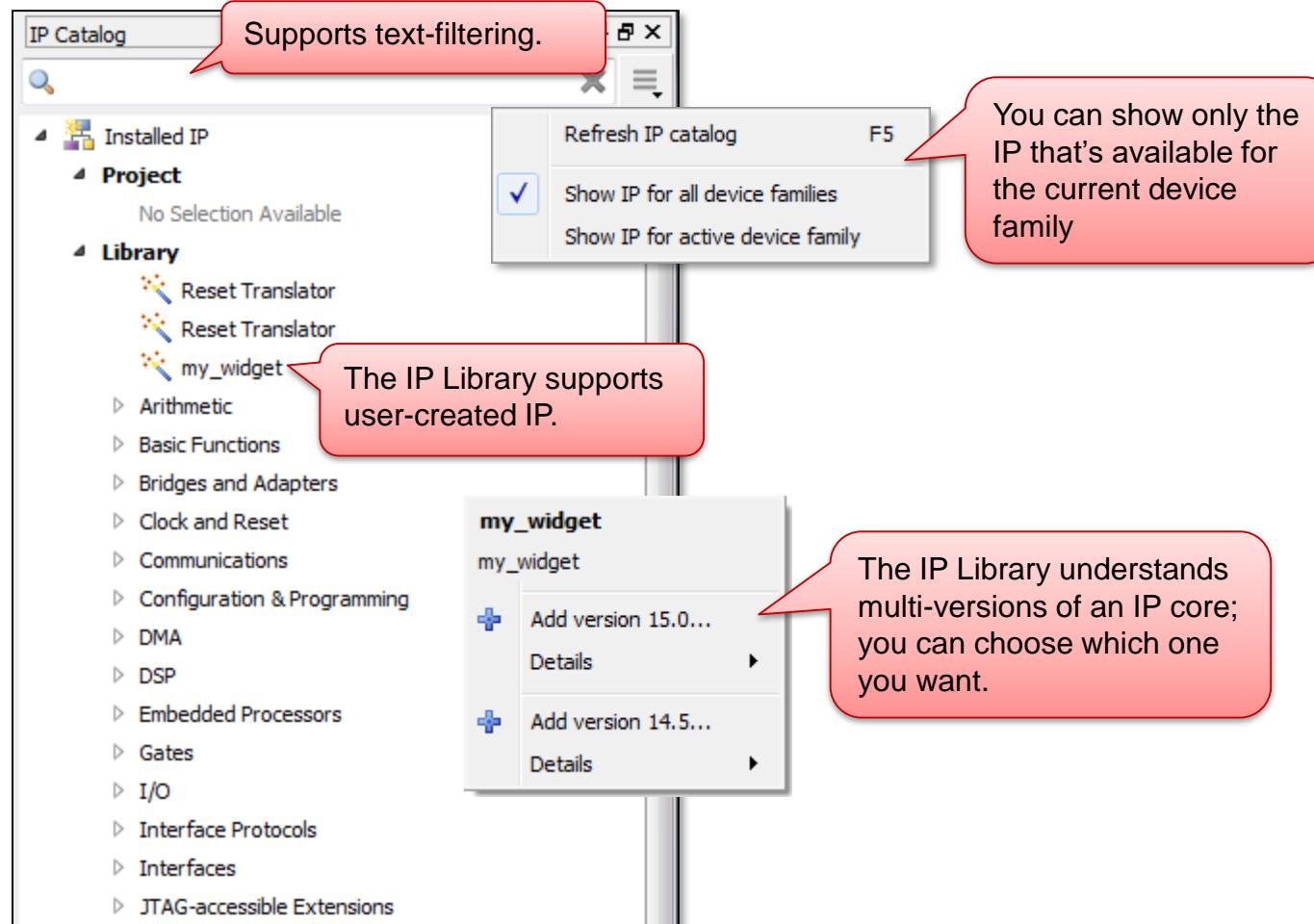
Altera FPGA器件



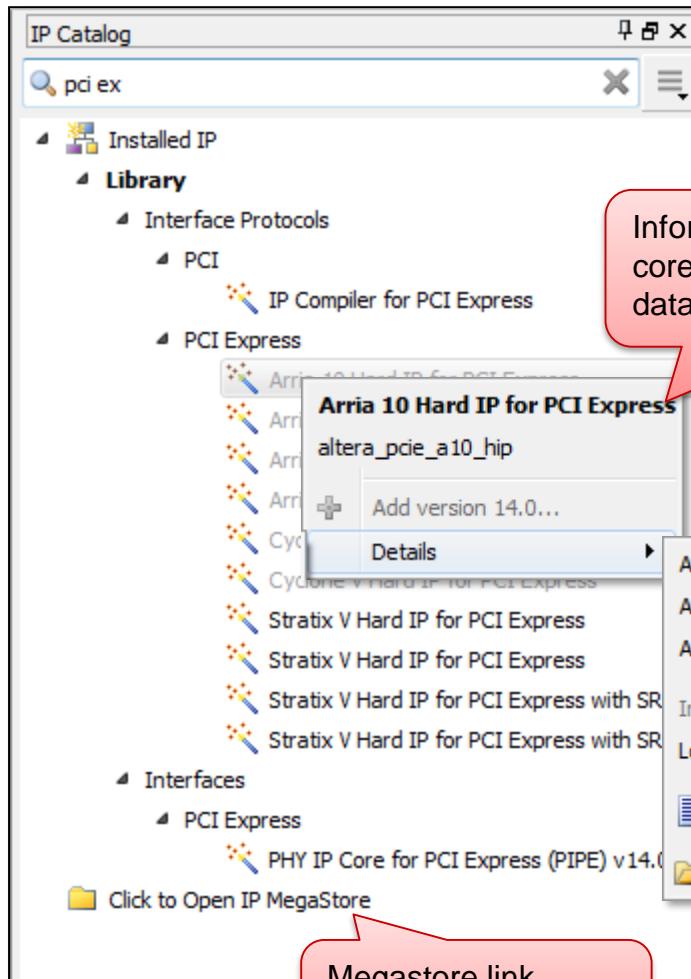
Quartus II v14.0 - IP Catalog



Quartus II v14.0 - IP Catalog



Quartus II v14.0 - IP Catalog



Information about each IP core, including a link to the datasheet, when available.

Arria 10 Hard IP for PCI Express (altera_pcie_a10_1ip)

Arria 10 Hard IP for PCI Express

Altera Corporation

Installed version: 14.0

Location: C:\altera\14.0_B164\ip\altera\altera_pcie\altera_pcie_a10_1ip\altera_pcie_a10_1ip_hw.td

DATASHEET

Open Component Folder..

Megastore link
opens to a page on
www.altera.com.

Arria 10® Hard IP for PCI Express® IP Core in the Quartus® II 13.1 Software Release

December 2013

[Subscribe](#) [Send Feedback](#)

The Arria 10 Hard IP for PCI Express has separate user guides for the **Interface Type** that you select in the component GUI. When you select the Avalon® Memory-Mapped (Avalon-MM) interface for the Gen2 x8, Gen3 x4, and Gen3 x8 variants, the IP core automatically includes high-performance DMA Read and DMA Write engines. Click on the link below for the appropriate user guide.

- [Arria 10 Hard IP for PCI Express User Guide for the Avalon Streaming Interface](#)
- [Arria 10 Hard IP for PCI Express User Guide for the Avalon Memory-Mapped Interface](#)
- [Arria 10 Hard IP for PCI Express User Guide for the Avalon Memory-Mapped Interface with DMA](#)

© 2013 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, HARDCORE, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders, as described at [www.altera.com/legal.html](#). Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty; Altera reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

ISO
9001:2008
Registered