

**ALTERA**

# Quartus II Software Design Series: Foundation

*Online Training*



# Supplemental Files to Download



- Complete presentation in PDF format
  - Lab exercise manual in PDF format
  - Lab exercise files (executable ZIP file)
  - All files contained in single .zip file
- *Click link in email or go to **Attachments** button to download (may need to hold **Ctrl** to download)*

# Objectives

- Create a new Quartus® II project
- Choose supported design entry methods
- Compile a design into an FPGA
- Locate resulting compilation information
- Assign design constraints (timing & pin)
- Perform timing analysis & obtain results
- Generate files for 3<sup>rd</sup>-party EDA simulation
- Configure an FPGA

# Class Agenda

- Projects
  - Exercise 1
- Design Entry
  - Exercise 2
- Compilation
  - Exercise 3
- Settings & Assignments
  - Exercise 4
- I/O Planning
  - Exercise 5
- Timing Analysis
  - Exercise 6
- EDA Simulation
- Programming / configuration
  - Exercise 7 (optional)

# Advanced Quartus II Courses

## ■ Quartus II Software Design Series: Verification

- Timing analysis
  - Thorough investigation of performing timing analysis on an Altera device with TimeQuest
- Power analysis
- Debugging solutions
  - SignalProbe incremental routing
  - Logic Analyzer Interface
  - In-System Memory Content Editor
  - In-System Sources & Probes
  - Chip Planner & Resource Property Editor
  - SignalTap II Embedded Logic Analyzer

## ■ Quartus II Software Design Series: Optimization

- Incremental Compilation
- Quartus II optimization features & techniques

**ALTERA**

# Quartus II Software Design Series: Foundation

*Introduction to Altera  
& Altera Devices*



# The Programmable Solutions Company®

- Programmable Logic Devices
- Tools
  - Quartus® II software
  - SOPC Builder
  - DSP Builder
  - Nios® II IDE
- Intellectual Property (IP)
  - Signal processing
  - Communications
  - Embedded processors
    - Nios II embedded processor



# Programmable Logic Families

- **Structured ASIC**
  - HardCopy® II & HardCopy® Stratix devices
- **High & medium density FPGAs**
  - Stratix® family devices
- **Low-cost FPGAs**
  - Cyclone® family devices
- **FPGAs w/ high-speed transceivers**
  - Stratix II GX, Stratix GX, & Arria® GX devices
- **CPLDs**
  - MAX® II, MAX 7000 & MAX 3000 devices
- **Configuration devices**
  - Serial (EPCS) & enhanced (EPC)



HARDCOPY™ II

**Stratix III**

**Cyclone III**

**Stratix II GX**

**Arria GX**

**MAX II**

**ALTERA.**



# Software & Development Tools



## ■ Quartus II Subscription Edition

- Stratix III, Stratix, II & Stratix devices
- Stratix II GX, Stratix GX, & Arria GX devices
- Cyclone III, Cyclone II, & Cyclone devices
- HardCopy II & HardCopy Stratix devices
- MAX II, MAX 7000S/AE/B, MAX 3000A devices
- Select older families

## ■ Quartus II Web Edition

- Free version
- Not all features & devices included
  - See [www.altera.com](http://www.altera.com) for feature comparison

**ALTERA**

# Quartus II Software Design Series: Foundation

*Quartus II Design Software Feature  
Overview*



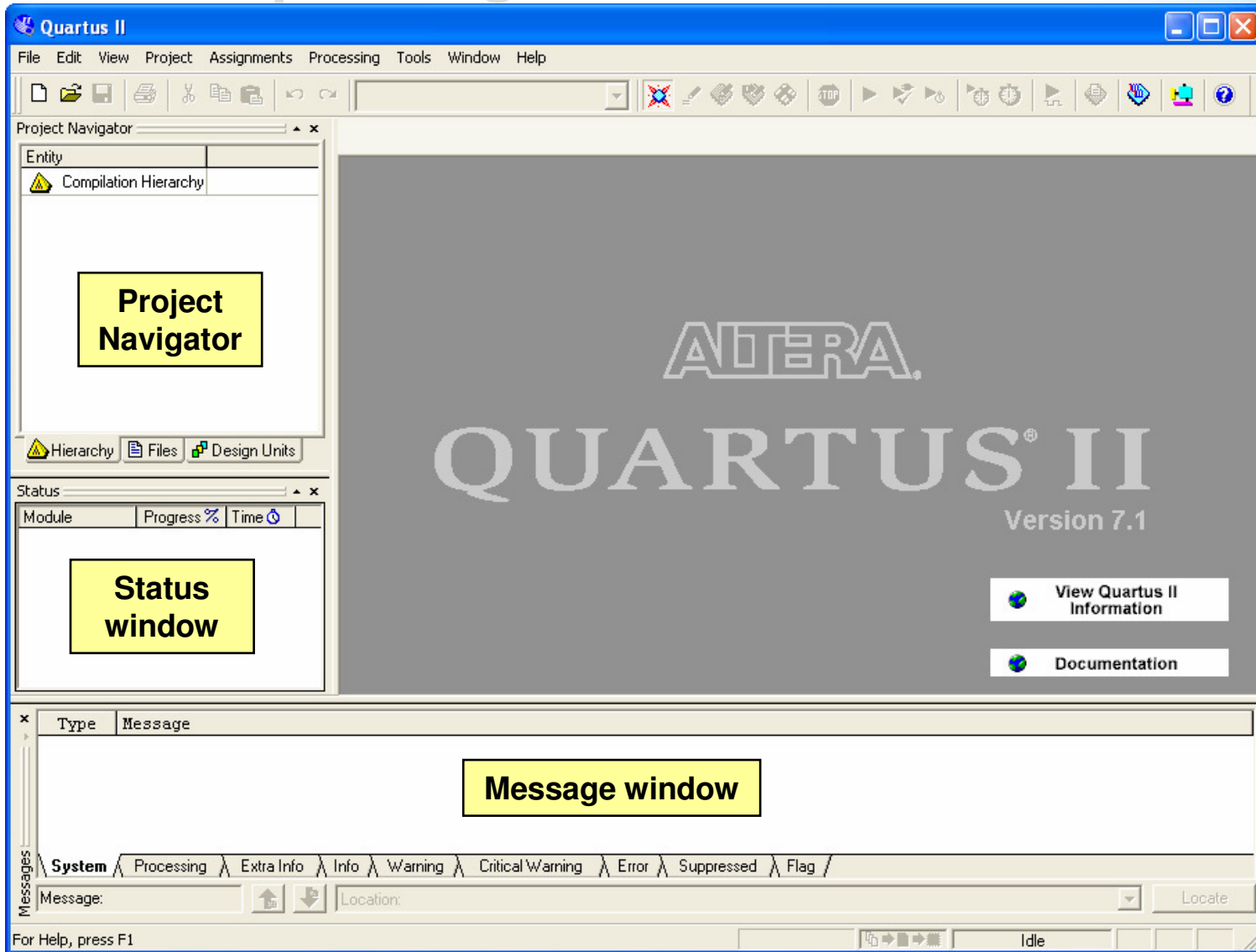
# Quartus II Design Software

- Fully-integrated development tool
  - Multiple design entry methods
  - Logic synthesis
  - Place & route
  - Simulation
  - Timing & power analysis
  - Device programming

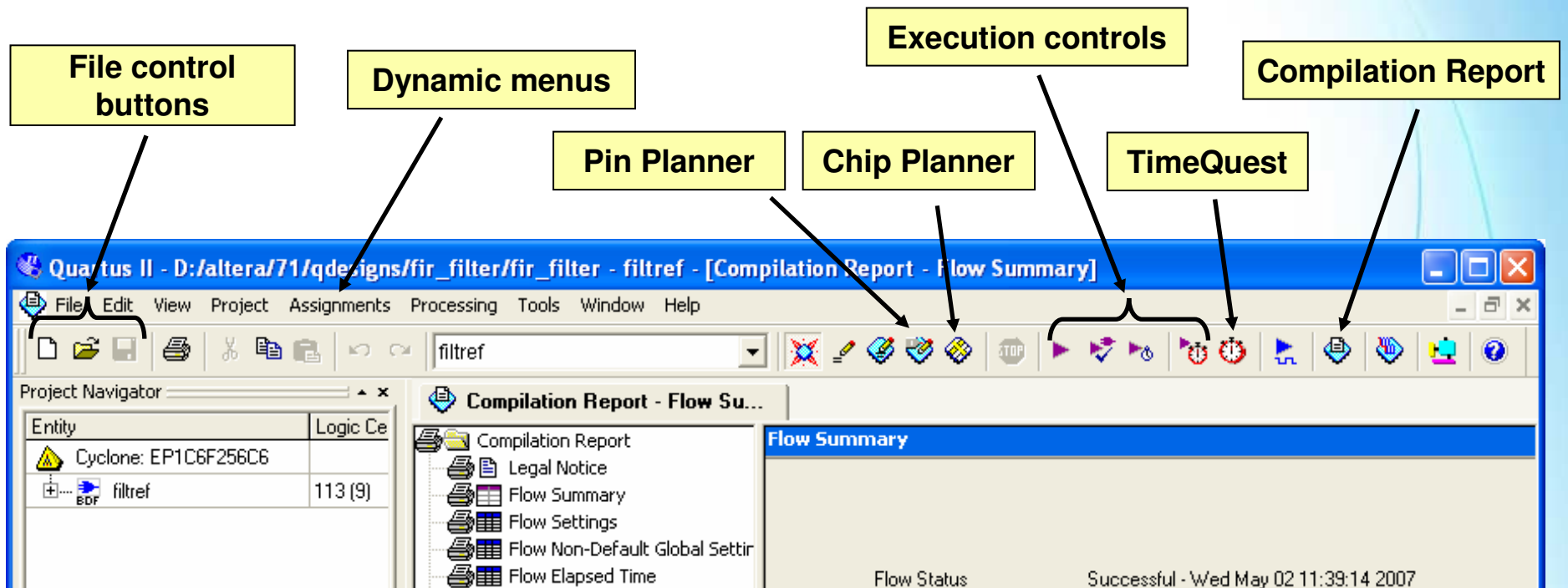
# More Features

- MegaWizard® & SOPC Builder design tools
- TimeQuest Timing Analyzer
- Incremental Compilation feature
- PowerPlay Power Analyzer tool
- NativeLink® 3<sup>rd</sup>-party EDA tool integration
- Debugging capabilities
  - From HDL to device in-system
- 32 & 64-bit Windows, Solaris, & Linux support
- Multi-processor support
- Node-locked & network licensing options

# Quartus II Operating Environment



# Main Toolbar

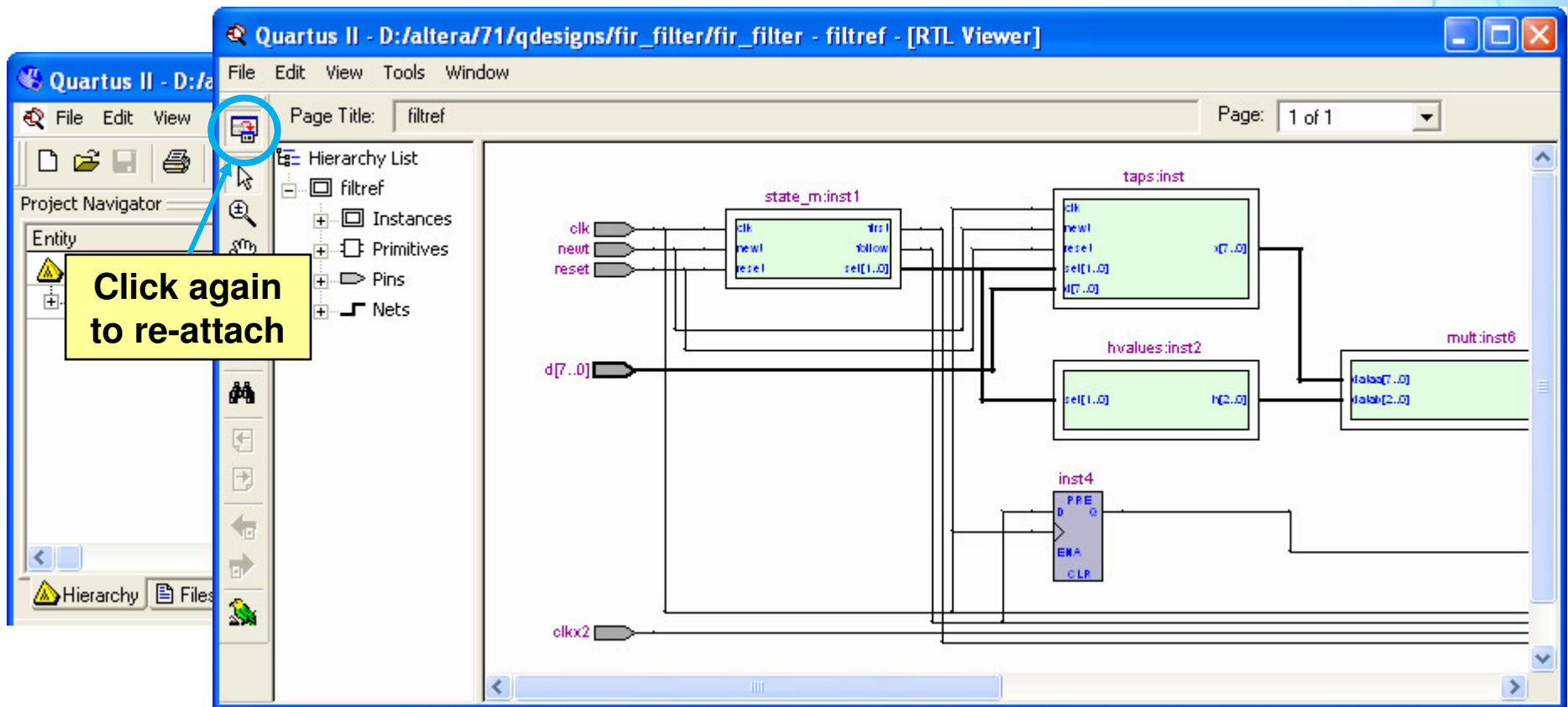


**To reset views:**

- 1. Tools ⇒ Customize ⇒ Toolbars ⇒ Reset All**
- 2. Restart Quartus II**

# Detachable Windows

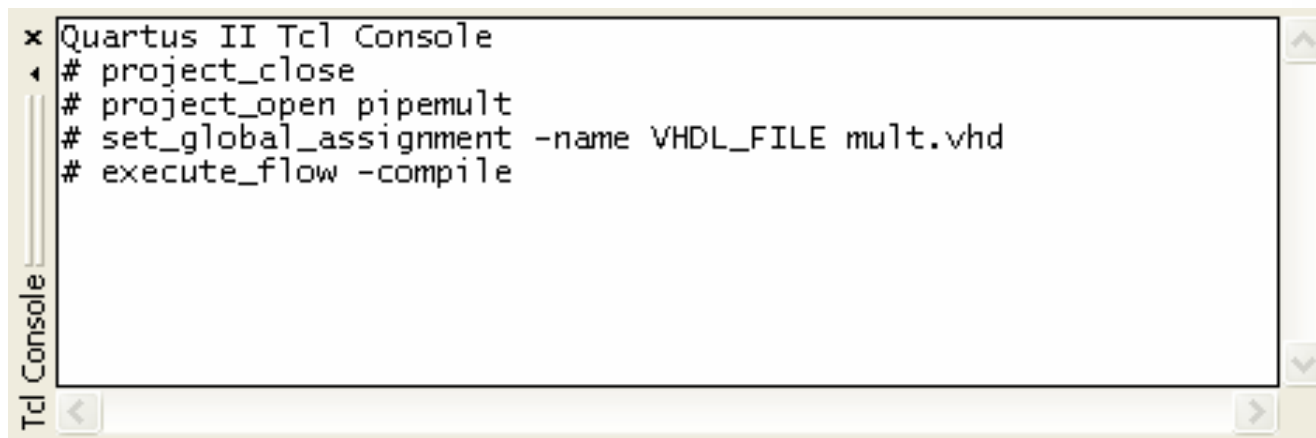
- Separate child windows from the Quartus II GUI frame (Window menu ⇒ Detach/Attach Window)



# Tcl Console Window

- Enter and execute Tcl commands directly in the GUI

View menu ⇒ Utility Windows ⇒ Tcl Console



```
Quartus II Tcl Console
# project_close
# project_open pipemult
# set_global_assignment -name VHDL_FILE mult.vhd
# execute_flow -compile
```

- Execute from command-line using Tcl shell

- `quartus_sh --shell`



# Tips & Tricks Advisor

**Help menu => Tips & Tricks**

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. <a href="#">Open Settings dialog box - Early Timing Estimate page</a>

**Provides useful instructions on using the Quartus II software. Available sections include:**

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

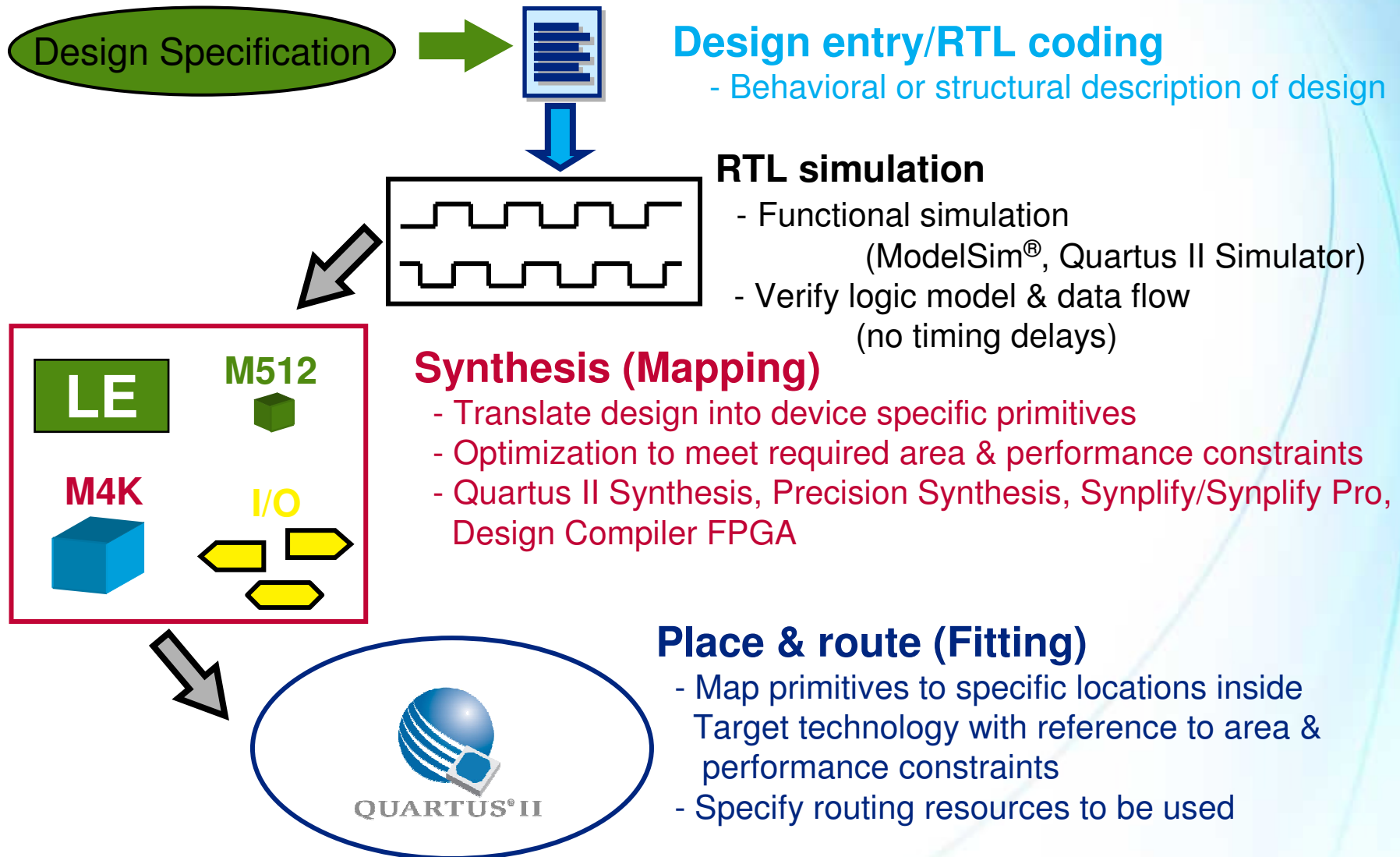
**ALTERA**

# Quartus II Software Design Series: Foundation

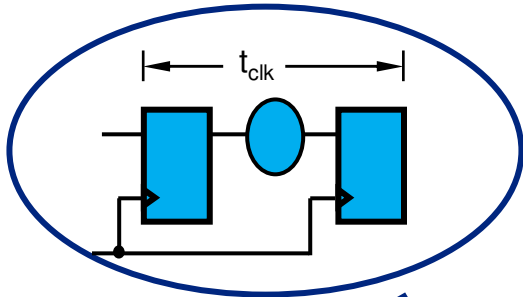
*Design Methodology*



# Typical PLD Design Flow

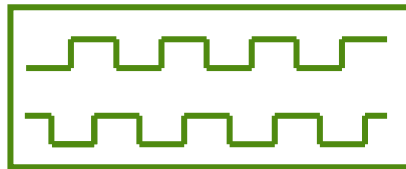


# Typical PLD Design Flow



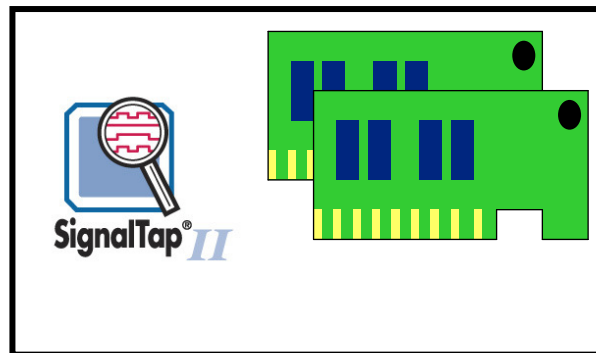
## Timing analysis (Classic Analyzer or TimeQuest)

- Verify performance specifications were met
- Static timing analysis



## Gate level simulation

- Timing simulation
- 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

**ALTERA**

# Quartus II Software Design Series: Foundation

*Quartus II 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)

## ■ Create new projects with New Project Wizard

- Can be created using Tcl scripts

# New Project Wizard

The image shows a screenshot of the 'New Project Wizard' dialog box in a software application. The dialog box is titled 'New Project Wizard: Directory, Name, Top-Level Entity [page 1 of 5]'. It contains three text input fields and a button. The first field is labeled 'What is the working directory for this project?' and contains the path 'D:\valtera\71\qdesigns\my\_new\_project'. The second field is labeled 'What is the name of this project?' and contains 'my\_project'. The third field is labeled '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.' and contains 'my\_project'. Below these fields is a button labeled 'Use Existing Project Settings ...'. At the bottom of the dialog box are four buttons: '< Back', 'Next >', 'Finish', and 'Cancel'. To the left of the dialog box is a screenshot of the application's menu bar with the 'File' menu open. The 'File' menu contains several options, with 'New Project Wizard...' highlighted. A yellow box labeled 'File menu' points to the 'File' menu. Three yellow boxes with arrows point to the three input fields in the dialog box. The first box is labeled 'Select working directory'. The second box is labeled 'Name of project can be any name; recommend using top-level file name'. The third box is labeled 'Top-level entity does not need to be the same name as top-level file name'. A yellow box with an arrow pointing to the 'Use Existing Project Settings ...' button is labeled 'Create a new project based on an existing project & settings'. At the bottom left, a white box with a black border contains the text 'Tcl: project\_new <project\_name>'. The Altera logo is visible in the bottom right corner of the slide.

**File menu**

**Select working directory**

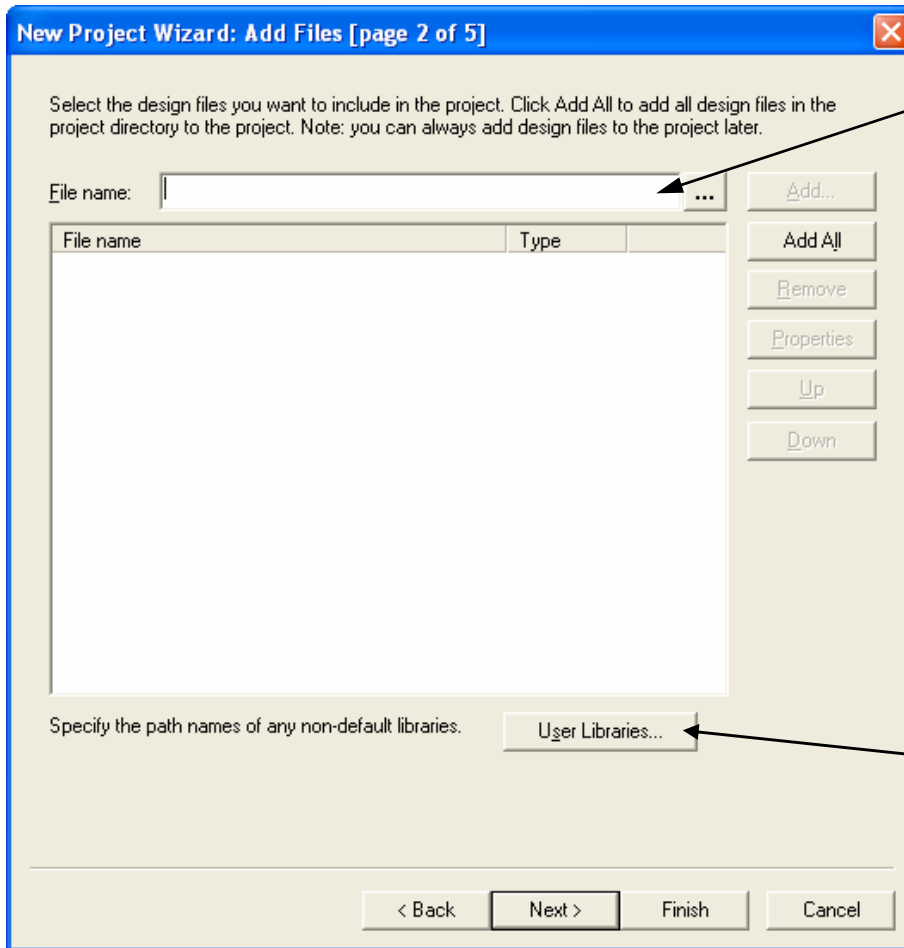
**Name of project can be any name; recommend using top-level file name**

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

**Create a new project based on an existing project & settings**

**Tcl: `project_new <project_name>`**

# Add Files



## Add design files

- Graphic (.BDF, .GDF)
- AHDL
- VHDL
- Verilog
- EDIF

### Notes:

- *Files in project directory do not need to be added*
- *Add top-level file if filename & entity name are not the same*
- *Absolute & relative paths are supported*

## Add user library pathnames

- User libraries (any directory containing files)
- MegaCore®/AMPP<sup>SM</sup> libraries
- Pre-compiled VHDL packages

```
Tcl: set_global_assignment -name VHDL_FILE* <filename.vhd>  
Tcl: set_global_assignment -name USER_LIBRARIES <library_path_name>  
* Replace with VERILOG_FILE, EDIF_FILE, AHDL_FILE or BDF_FILE
```



# Device Selection

New Project Wizard: Family & Device Settings [page 3 of 5]

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

Family: Cyclone III

Target device

Auto device selected by the Fitter

Specific device selected in 'Available devices' list

Show in 'Available device' list

Package: FBGA

Pin count: 256

Speed grade: Any

Show advanced devices

HardCopy compatible only

Available devices:

Name	Core voltage	LEs	User I/Os	Memory bits	Embe...	PLL	Global
EP3C10F256C7 ...	1.2V	10320	183	423936	46	2	10
EP3C10F256C8 ...	1.2V	10320	183	423936	46	2	10
EP3C16F256C6 ...	1.2V	15408	169	516096	112	4	20
EP3C16F256C7 ...	1.2V	15408	169	516096	112	4	20
EP3C16F256C8 ...	1.2V	15408	169	516096	112	4	20
EP3C25F256C6	1.2V	24624	157	608256	132	4	20
EP3C25F256C7	1.2V	24624	157	608256	132	4	20
EP3C25F256C8	1.2V	24624	157	608256	132	4	20

Companion device:

HardCopy II:

Limit DSP & RAM to HardCopy II device resources

< Back Next > Finish Cancel

Choose device family and filter results

Choose specific part number from list or let Quartus II choose smallest, fastest device based on filter criteria

```
Tcl: set_global_assignment -name FAMILY "device family name"  
Tcl: set_global_assignment -name DEVICE <part_number>
```



# EDA Tool Settings

**Choose EDA tools  
& file formats**

**Add or change  
settings later**

Specify the other EDA tools -- in addition to the Quartus II software -- used with the project.

EDA design entry/synthesis tool: Synplify Pro  
Format: VQM  
 Run this tool automatically to synthesize the current design

EDA simulation tool: ModelSim-Altera  
Format: Verilog  
 Run this tool automatically after compilation

EDA timing analysis tool: PrimeTime  
Format: Verilog  
 Run this tool automatically after compilation

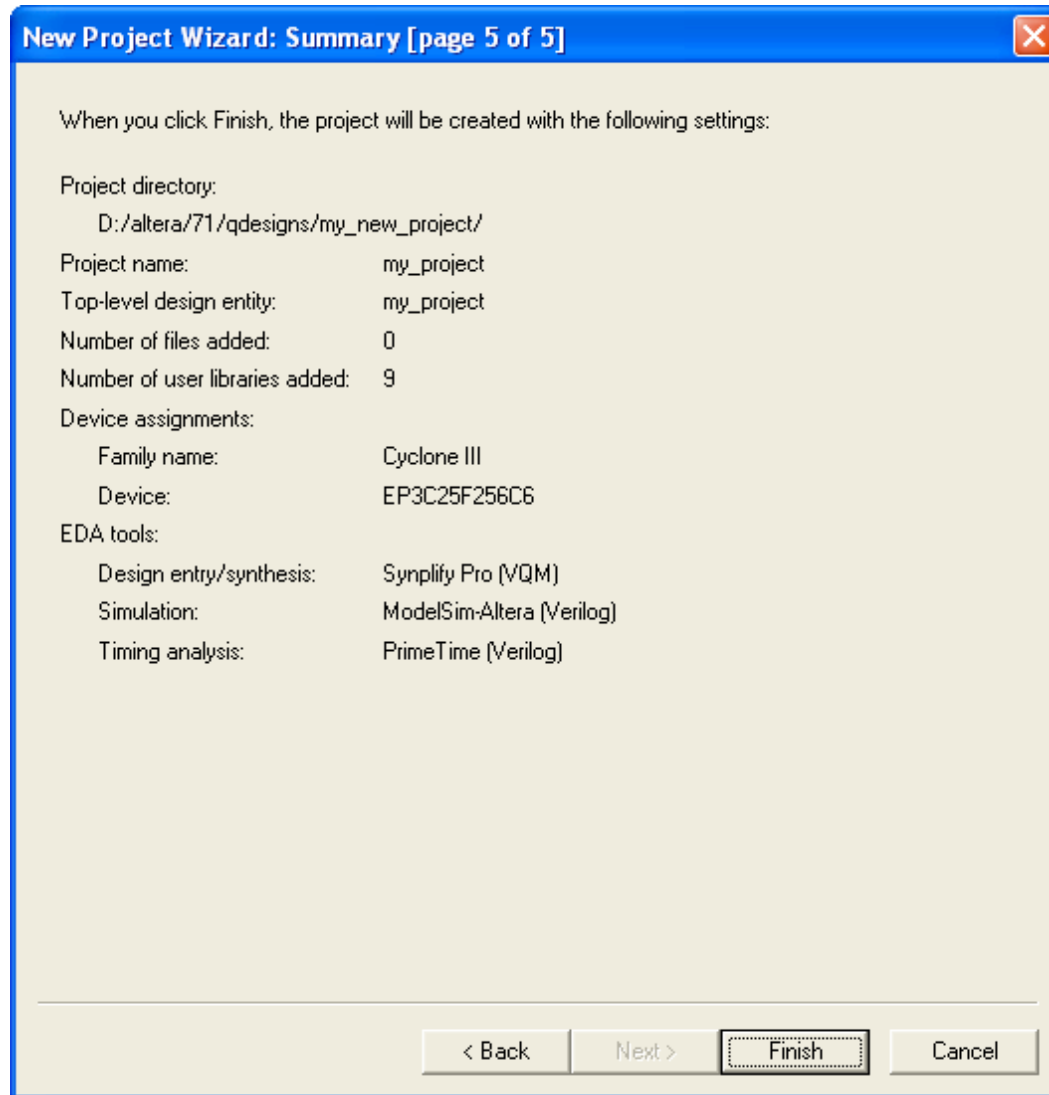
< Back   Next >   Finish   Cancel

See handbook for Tcl command format

© 2007 Altera Corporation—**Confidential**

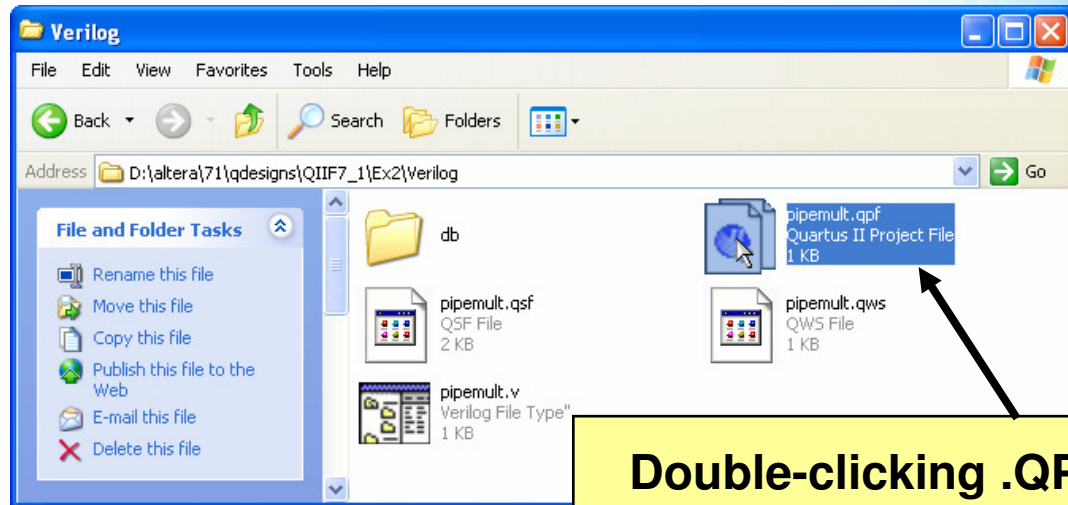
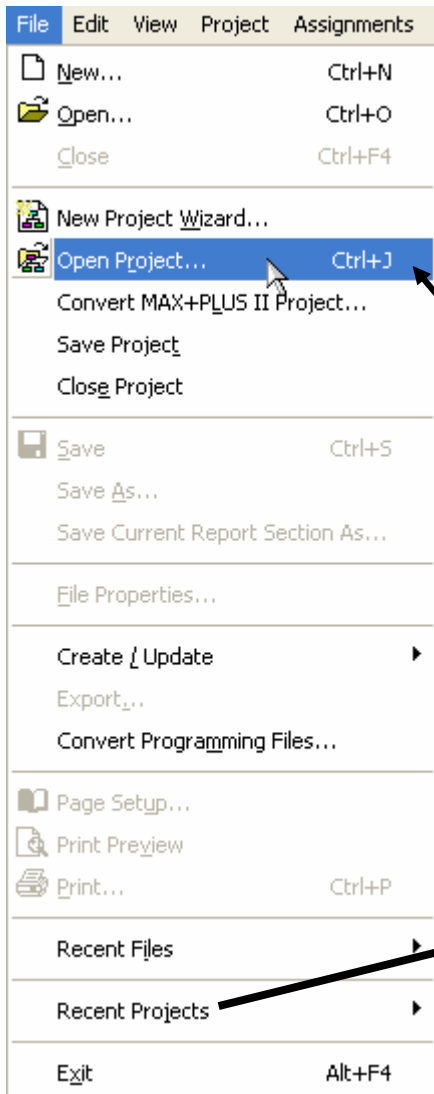
Altera, Stratix, Arria, Cyclone, MAX, HardCopy, Nios, Quartus, and MegaCore are trademarks of Altera Corporation

# Done!



**Review results & click Finish**

# Opening an Existing Project



File ⇒ Open Project...

OR

Select from most recent projects list



*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

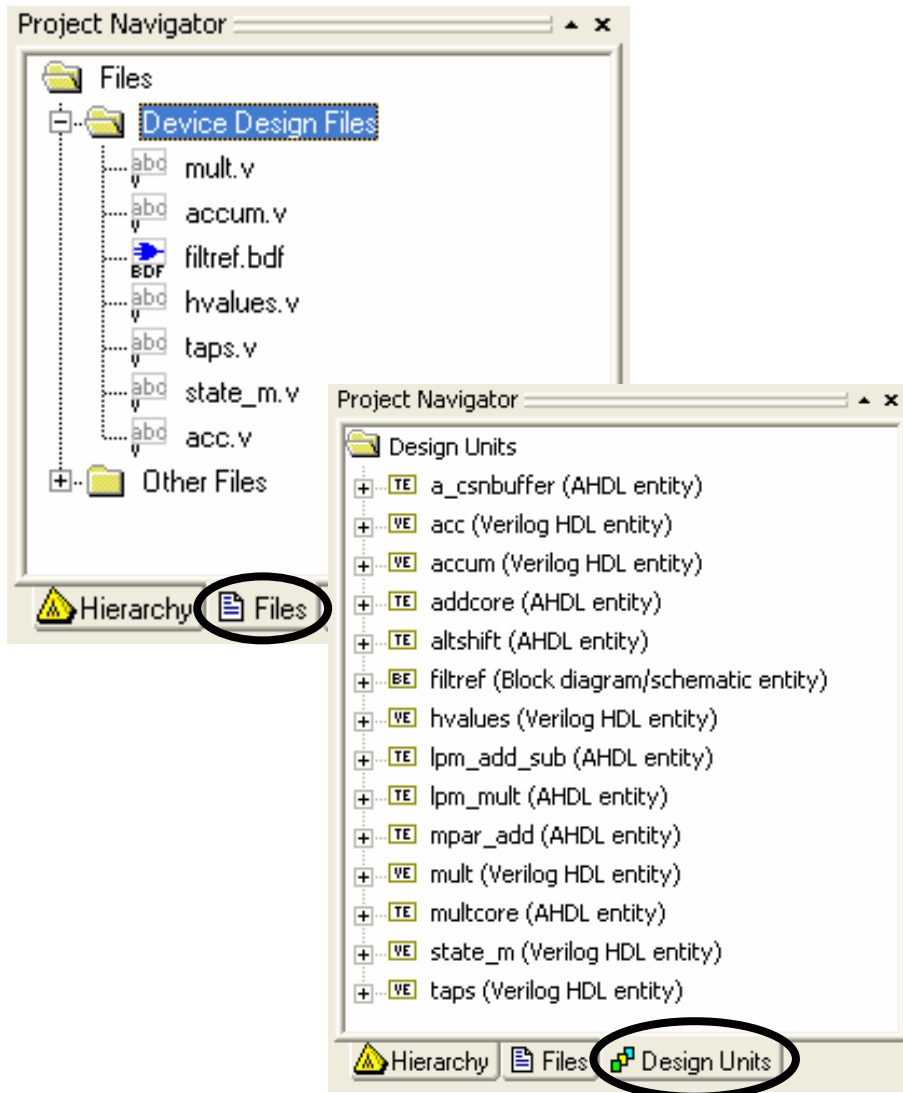
Project Navigator

Entity	Logic Cells	LC Registers	Memory Bit:
Cyclone: EP1C6F256C6			
filterf	102 (9)	58	0
taps:inst	32 (32)	32	0
state_m:inst1	5 (5)	5	0
hvalues:inst2			
acc:inst3			
mult:inst6			

Select & right-click

Full compilation or Processing menu => Start => Start Analysis & Elaboration

# Files & Design Units Tabs



## ■ Files tab

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

## ■ Design Units tab

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

# Quartus II Project Files

- Quartus II Project File (.QPF)
- Quartus II Defaults File (.QDF)
- Quartus II Settings File (.QSF)
  
- Synopsys Design Constraints (.SDC)
  - Holds timing constraints
  - Discussed later

# Project & Default Files

## ■ Quartus II Project File (QPF)

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

*fir\_filter.QPF*

```
QUARTUS_VERSION = "7.1"  
DATE = "14:31:04 May 02, 2007"  
  
# Active 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 in local project directory read before original in bin



# Quartus II Settings File (QSF)

- Stores all settings & assignments (constraints)
- Uses Tcl syntax
- Can be edited manually by user

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

Add user comments (#) & white space

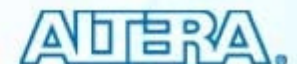
Reorganize QSF based on categories (Project menu)

Source other TCL/QSF files to organize assignments

NEW assignments added to end of file

```
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
26
27 # Project-Wide Assignments
28 # =====
29 set_global_assignment -name ORIGINAL_QUARTUS_VERSION 7.1
30 set_global_assignment -name PROJECT_CREATION_TIME_DATE "08:37:10 APRIL 10, 2007"
31 set_global_assignment -name LAST_QUARTUS_VERSION 7.1
32 set_global_assignment -name VERILOG_FILE mult.v
33 set_global_assignment -name VERILOG_FILE accum.v
34 set_global_assignment -name BDF_FILE filtref.bdf
35 set_global_assignment -name VERILOG_FILE hvalues.v
36 set_global_assignment -name VERILOG_FILE taps.v
37 set_global_assignment -name VERILOG_FILE state_m.v
38 set_global_assignment -name VERILOG_FILE acc.v
39 set_global_assignment -name SMART_RECOMPILE ON
40 set_global_assignment -name VECTOR_WAVEFORM_FILE fir.vwf
41
42 # This is where my pin assignments are located
43 # =====
44 set_location_assignment PIN_G1 -to clk
45 source "location_assignments.tcl"
46
47 # Classic Timing Assignments
48 # =====
49 set_global_assignment -name FMAX_REQUIREMENT "85 MHz"
50
51 # Analysis & Synthesis Assignments
52 # =====
53 set_global_assignment -name FAMILY Cyclone
54 set_global_assignment -name TOP_LEVEL_ENTITY filtref
55 set_global_assignment -name DEVICE_FILTER_PACKAGE FBGA
```

Note: See Appendix for more notes on using QSF file.



# Constraint File Priority

1. QSF
2. Revision-specific QDF file located in project directory
  - *<revision\_name>\_assignment\_defaults.qdf*
  - Created automatically in project directory when revision opened in new 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

# Project Management

- Project archive & restore
- Project copy
- Revisions

# Project Archive

## ■ Creates 2 files

- Compressed Quartus II Archive File (.QAR)
  - Includes design files, QPF file, & QSF file(s)
  - Option to include databases (db folder in project directory)
    - Recompile necessary if databases not included
  - Creates local QDF file for archive
- Archive activity log (.QARLOG)

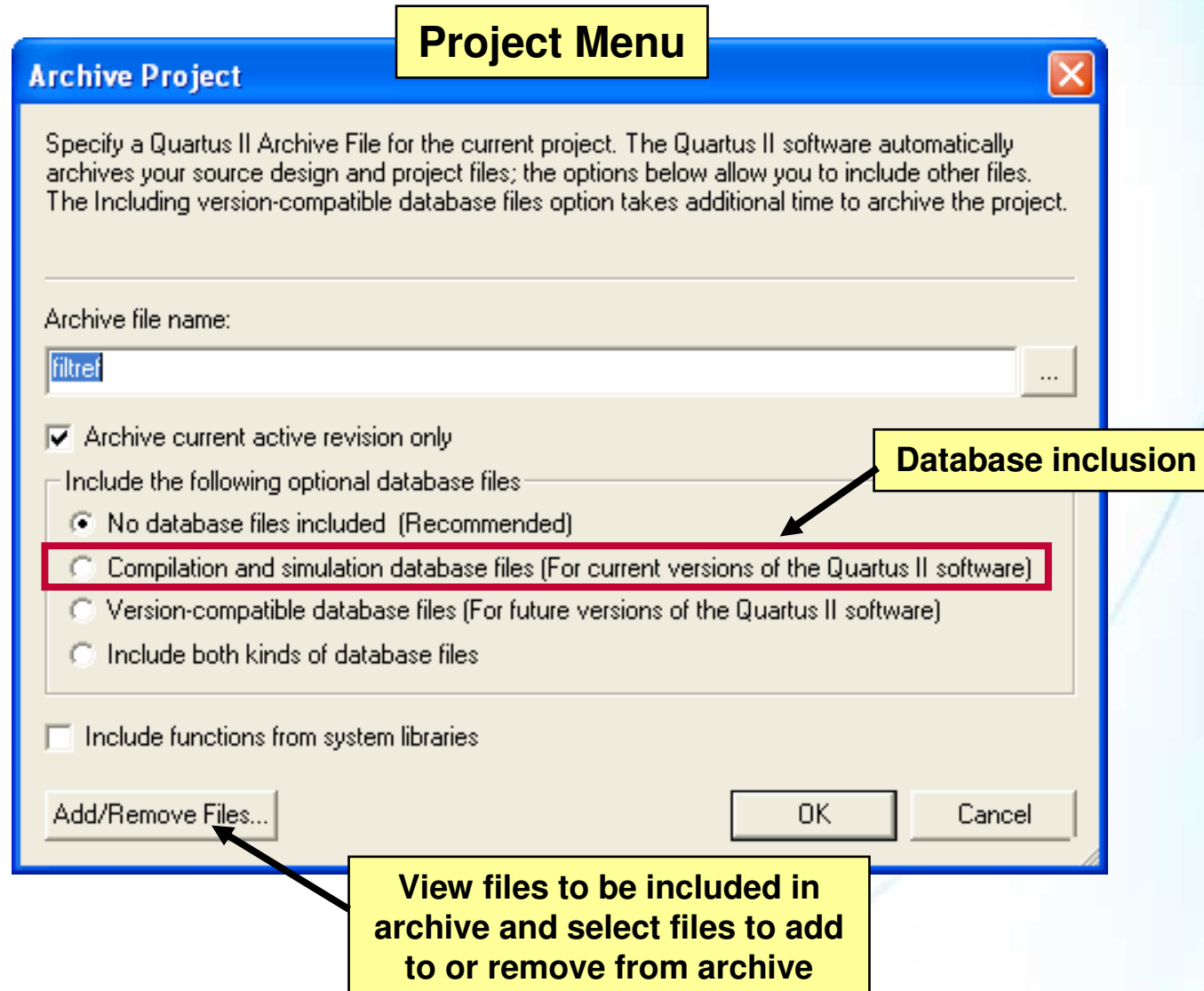
## ■ Example Uses

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

## ■ Design files referenced from user libraries are included in archive

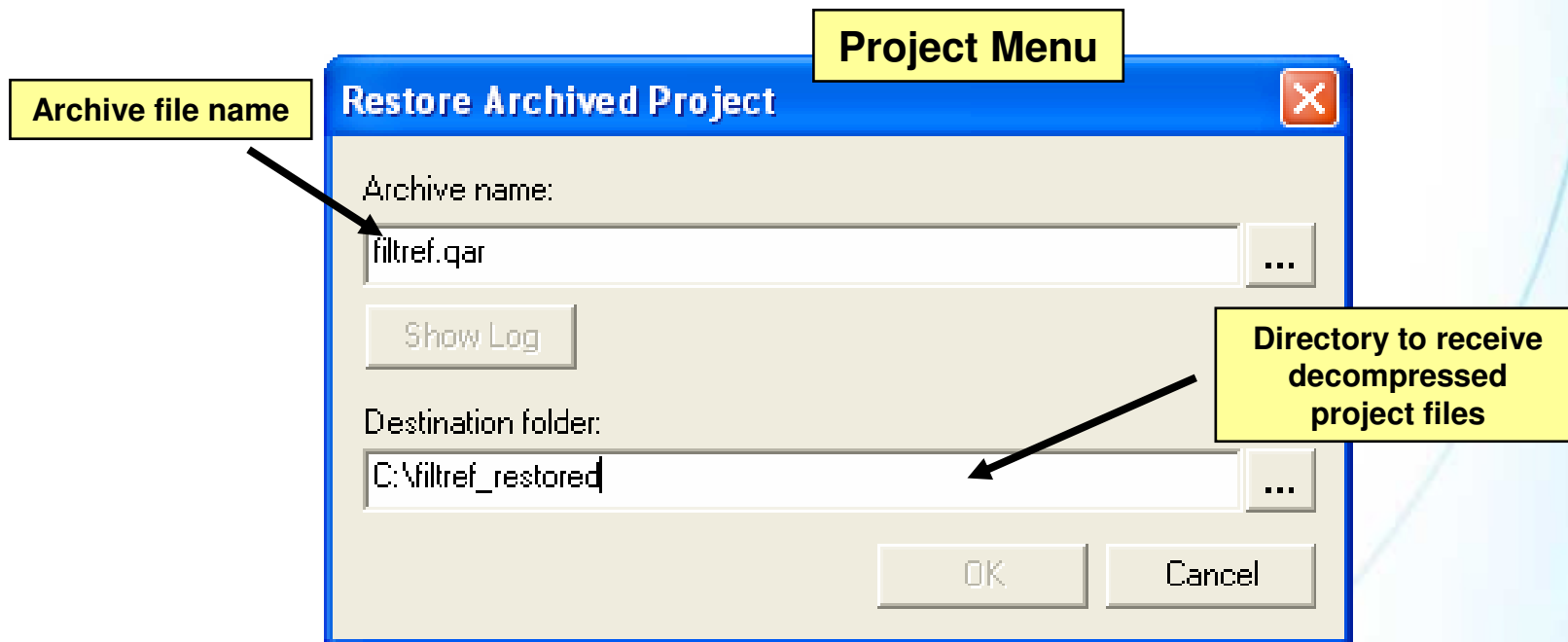
```
Tcl: project_archive <project_name>
```

# Project Archive (cont.)



# Project Restore

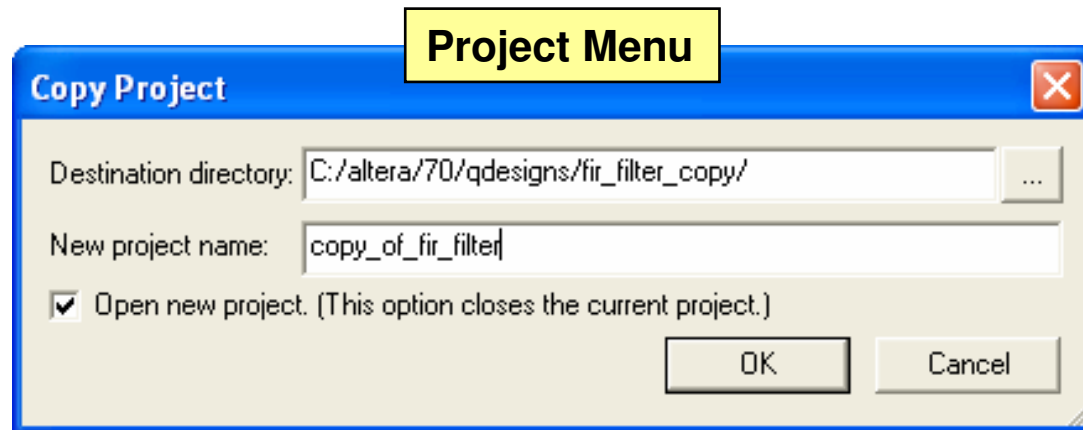
- Decompresses .QAR into specified directory



```
Tcl: project_restore <archive_file>
```

# Project Copy

- Copies & save 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
- New QDF not created; only copies QDF if it exists



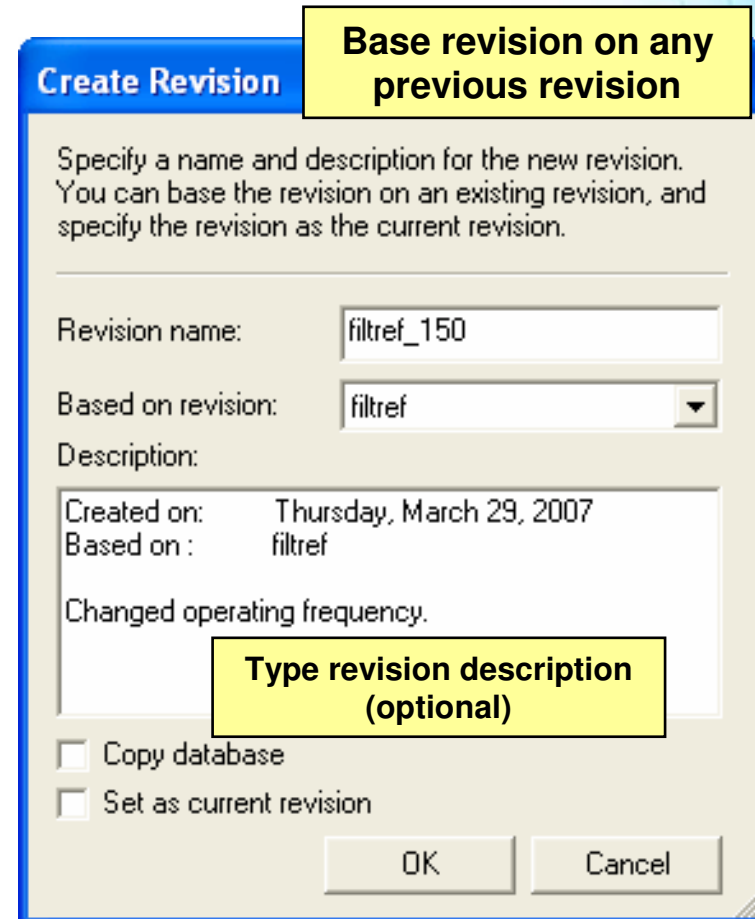
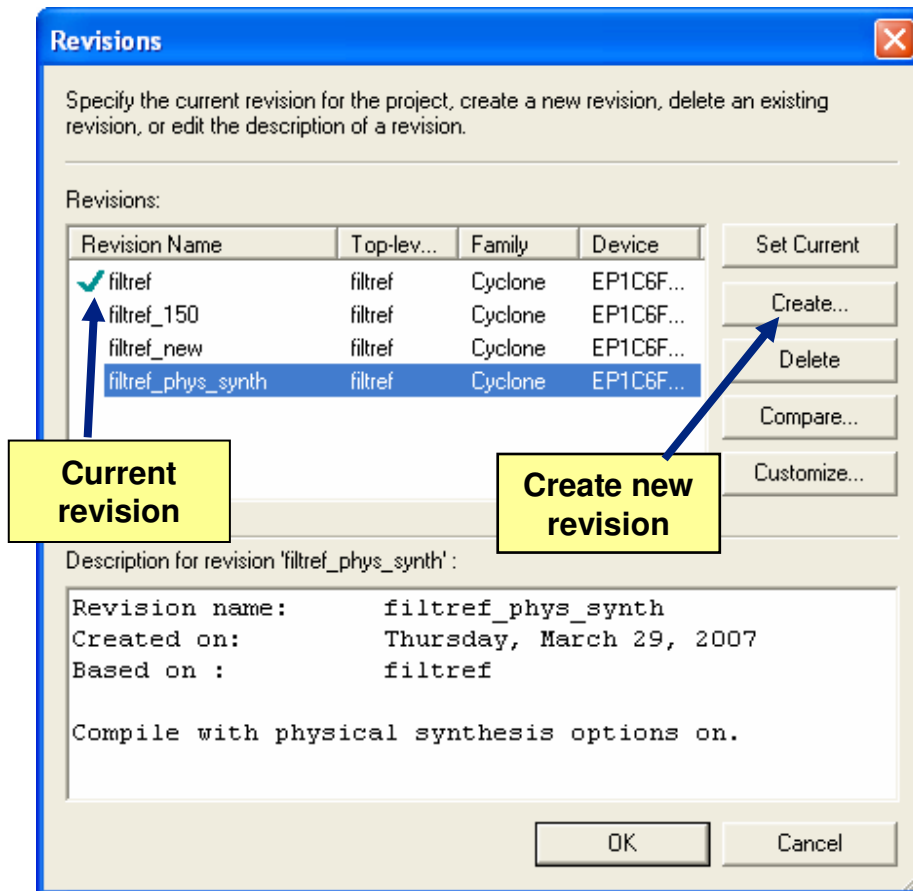
# Revisions

- Explore new sets of constraints or compile options without losing previous work
  - Allows designer to try different options on same design files
- Compare results between revisions



# Creating a Revision

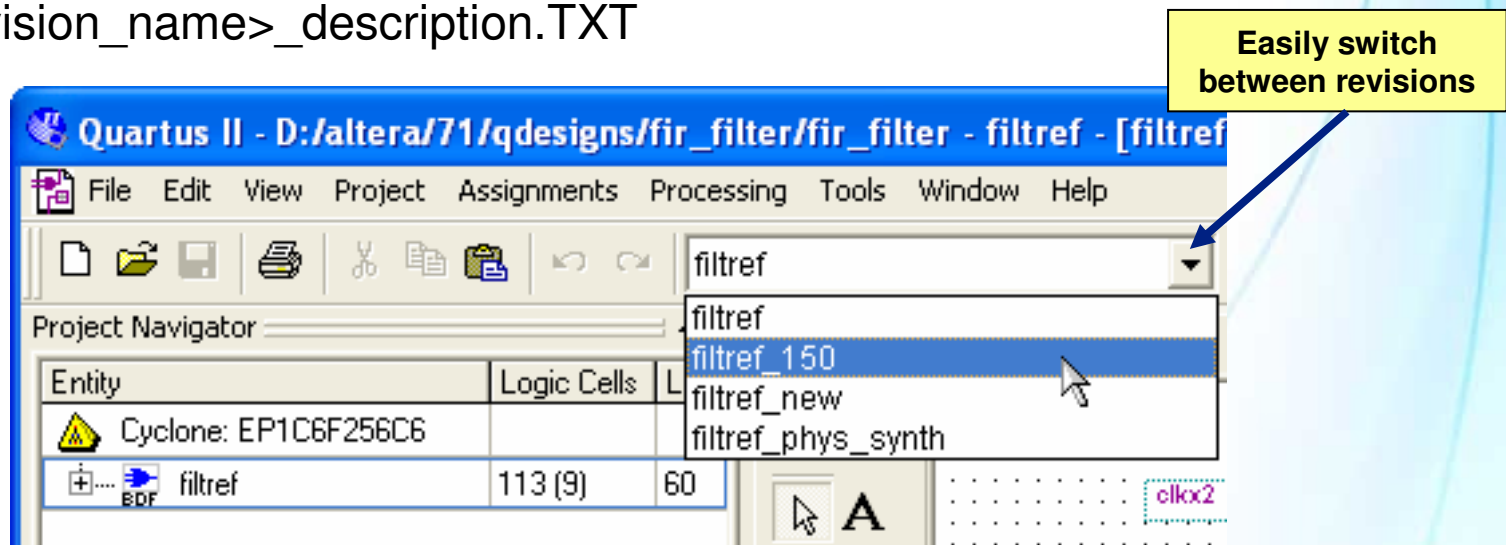
## ■ Project ⇒ Revisions



**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
  - <revision\_name>\_description.TXT



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

# Compare Revisions

**Compare Revisions**

Results | Assignments

	D:/altera/71/qdesigns/fit...	Revision filterf	Revision filterf_phys_synth
..... Total logic elements	133		133
..... Total pins	22		22
..... Total virtual pins	0		0
..... Total memory bits	0		0
..... Total PLLs	0		0
[-] Filter			
..... Fitter Status	Successful - Wed May 02 16:...		Successful - Wed May 02
..... Quartus II Version	7.1 Build 156 04/30/2007 SJ...		7.1 Build 156 04/30/2007
..... Revision Name	filterf		filterf_phys_synth
..... Top-level Entity Name	filterf		filterf
..... Family	Cyclone		Cyclone
..... Device	EP1C6F256C6		EP1C6F256C6
..... Timing Models	Final		Final
..... Total logic elements	162 / 5,980 ( 3 %)		147 / 5,980 ( 2 %)
..... Total pins	22 / 185 ( 12 %)		22 / 185 ( 12 %)
..... Total virtual pins	0		0
..... Total memory bits	0 / 92,160 ( 0 %)		0 / 92,160 ( 0 %)
..... Total PLLs	0 / 2 ( 0 %)		0 / 2 ( 0 %)
[+] Classic Timing Analyzer			
[-] TimeQuest Timing Analyzer			
[-] Setup 'clk'			
..... Slack	-7.343		-6.594
..... TNS	-71.533		-107.828
[-] Setup 'clkx2'			
..... Slack	-0.435		-0.482
..... TNS	-4.785		-3.121
[-] Hold 'clkx2'			
..... Slack	0.662		0.668
..... TNS	0.000		0.000
..... Slack	0.666		0.676
..... TNS	0.000		0.000

Customize... Close Export...

Detailed summary of revision assignments and results

To open, click Compare button in Revisions dialog box

Compare results with other projects

Export to CSV file

# *Exercise 1 Demonstration*

# Projects Entry 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, and revisions provide easy-to-use project management

# Project Support Resources

- “Managing Quartus II Projects” chapter in Volume 2 of the Quartus II Handbook

**ALTERA**

# Quartus II Software Design Series: Foundation

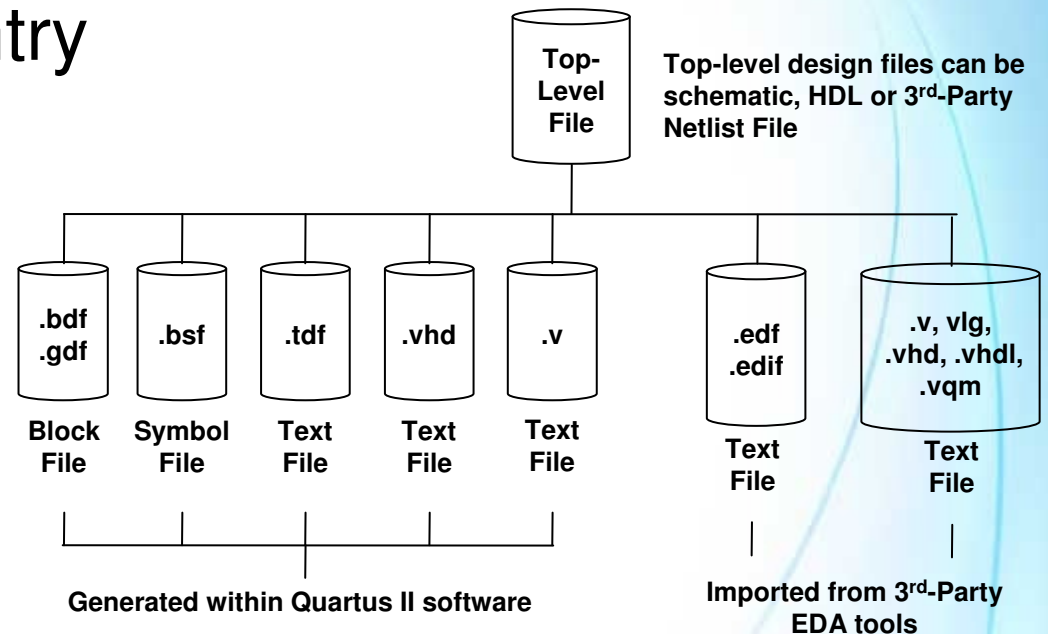
*Design Entry*



# Design Entry Methods

## ■ Quartus II design entry

- Text editor
  - AHDL
  - VHDL
  - Verilog
- Schematic editor
  - Block Diagram File
  - Graphic Design File
- Memory editor
  - HEX
  - MIF



## ■ 3<sup>rd</sup>-party EDA tools

- EDIF 2 0 0
- Verilog Quartus Mapping (.VQM)

## ■ Mixing & matching design files allowed



# Text Design Entry

## ■ Quartus II Text Editor features

- Block commenting
- Line numbering in HDL text files
- Bookmarks
- Preview/editing of full design and construct HDL templates
- Syntax coloring
- Find/replace text
- Find and highlight matching delimiters
- Function collapse/expand
- Edited but unsaved filenames appear with an asterisk (\*) next to the filename in the GUI

## ■ Enter text description

- AHDL (.tdf)
- VHDL (.vhd, .vhdl)
- Verilog (.v, .vlg, .Verilog, .vh)

# Verilog & VHDL

- **VHDL**- VHSIC hardware description language
  - IEEE Std 1076 (1987 & 1993) supported
  - IEEE Std 1076.3 (1997) synthesis packages supported
- **Verilog**
  - IEEE Std 1364 (1995 & 2001) & 1800 (SystemVerilog) supported
- Create in the Quartus II editor or any standard text editor
- Use Quartus II integrated synthesis to synthesize
- View supported commands in on-line help

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

# AHDL

- Altera hardware description language
  - High-level hardware behavior description language
  - Used in Altera megafunctions
  - Uses boolean equations, arithmetic operators, truth tables, conditional statements, etc.
  
- Create in the Quartus II editor or any standard text editor

# Text Editor Features

The image shows a screenshot of the Quartus II text editor interface with several callout boxes highlighting specific features:

- Find/highlight matching delimiters:** A yellow box with an arrow pointing to a function call in the code.
- Collapse/expand functions:** A yellow box with an arrow pointing to a function call in the code.
- Insert Template (Edit menu):** A yellow box with an arrow pointing to the 'Insert Template' menu item in the editor's menu bar.
- Bookmarks (on/off/jump to):** A yellow box with an arrow pointing to a bookmark icon in the left margin.
- Preview window: edit before inserting:** A yellow box with an arrow pointing to the preview window of the 'Insert Template' dialog.

The 'Insert Template' dialog is open, showing a tree view of language templates. The 'True Dual Port RAM (dual clock)' option is selected. The preview window shows the following Verilog code:

```
// Quartus II Verilog Template
// True Dual Port RAM with dual clocks

module true_dual_port_ram_dual_clock
(
    input [(DATA_WIDTH-1):0] data_a, data_b,
    input [(ADDR_WIDTH-1):0] addr_a, addr_b,
    input we_a, we_b, clk_a, clk_b,
    output reg [(DATA_WIDTH-1):0] q_a, q_b
);

parameter DATA_WIDTH = 8;
parameter ADDR_WIDTH = 6;

// Declare the RAM variable
reg [DATA_WIDTH-1:0] ram[2**ADDR_WIDTH-1:0];

always @(posedge clk_a)
begin
    // Port A
    if (we_a)
        q_a <= ram[addr_a];
end

always @(posedge clk_b)
```

# 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
    - Convert Block Diagram File (.BDF) to HDL file (VHDL/Verilog) or image file (.JPG or .BMP)

**Note:** Please see the Appendix for a more detailed discussion of the Block Diagram Editor and schematic entry.

# Altera Megafunctions

- Pre-made design blocks
- Benefits
  - Configurable settings add flexibility
  - “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
  - HDL simulation models installed in Quartus II libraries
- Two types
  - Altera-specific megafunctions (begin with “ALT”)
  - Library of parameterized modules (LPMs)
    - Industry standard logic functions
    - See [www.edif.org/lpmweb](http://www.edif.org/lpmweb) (EDIF.org archive) for more info
- Examples
  - Multiply-accumulate (ALTMULT\_ACCUM)
  - On-chip RAM/ROM (ALTSYNCRAM)
  - PLL (ALTPLL)
  - DDR/QDR memory interface (ALTMEMPHY)
  - Counter (LPM\_COUNTER)
  - Comparator (LPM\_COMPARE)

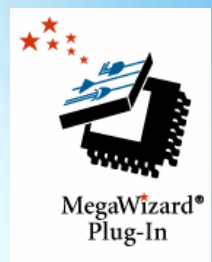
# IP Megafunctions

- Must purchase license to use in finished design
  - Logic for IP function is encrypted
- Two types
  - MegaCore® IP
    - Developed by Altera
    - Install with Quartus II software or download/install individually from [www.altera.com](http://www.altera.com)
  - Altera Megafunctions Partner Program (AMPP<sup>SM</sup>) IP
    - Developed by 3<sup>rd</sup>-Party IP vendors & certified by Altera
    - Contact vendor for evaluating and licensing function
- 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



# Example MegaCore IP

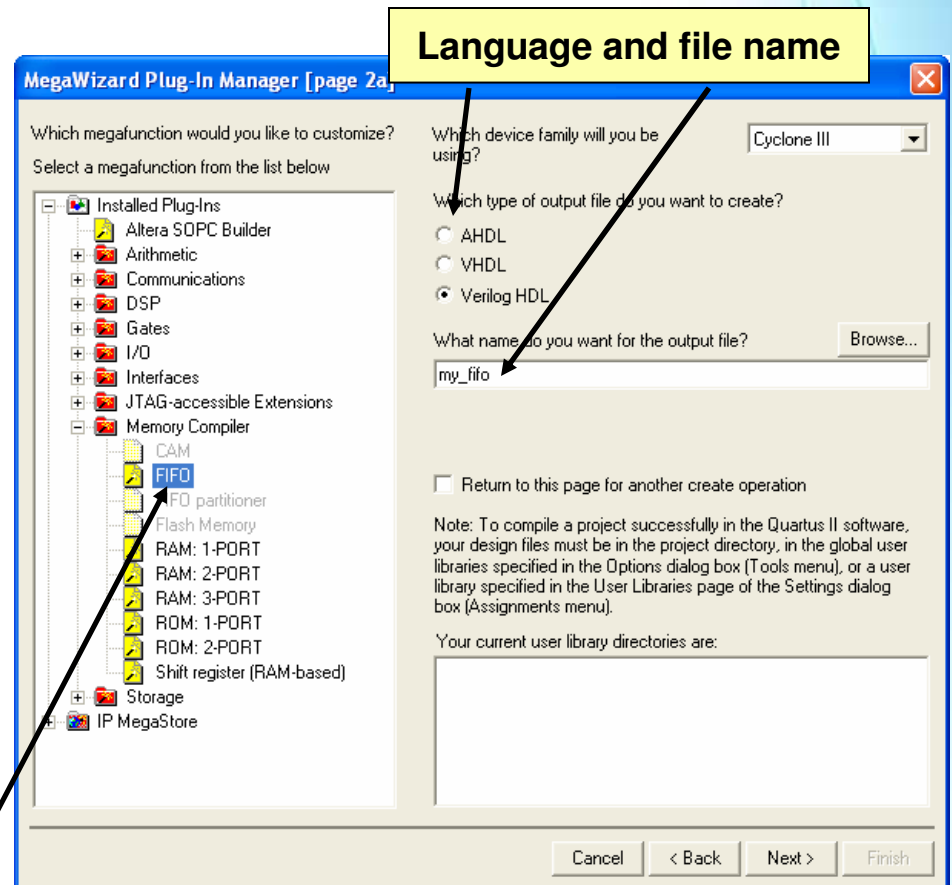
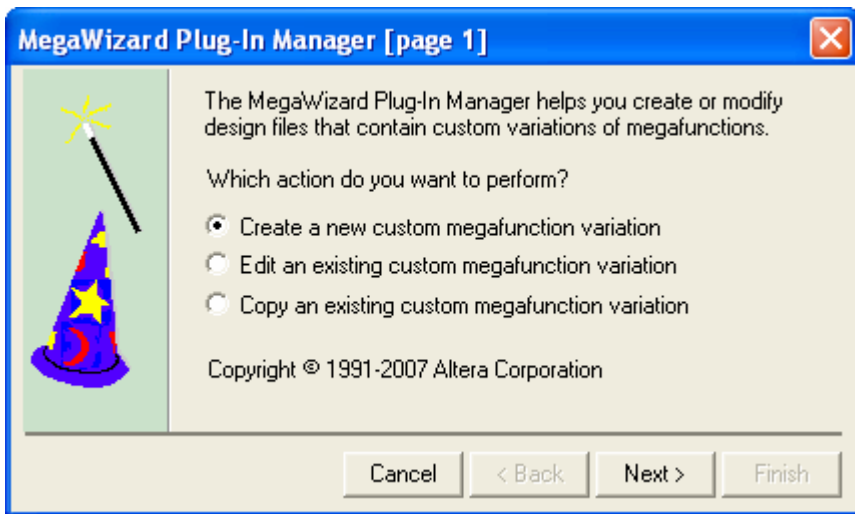
- Triple-Speed Ethernet MAC
- FIR Compiler
- Fast Fourier Transform
- DDR2 Memory Controller
- CRC Compiler
- PCI Compiler



# MegaWizard Plug-in Manager

- Eases implementation and configuration of megafunctions & IP

Tools ⇒ MegaWizard Plug-In Manager



Language and file name

Select megafunction or IP

# MegaWizard Example

**Locate documentation in Quartus II Help or the web** (points to the Documentation button)

**Multiply-Add megafunction** (points to the ALTMULT\_ADD title)

**Three step process to configure megafunction** (points to the Parameter Settings, EDA, and Summary tabs)

**Updating graphical representation** (points to the circuit diagram showing two multipliers and an adder)

**Resource usage** (points to the Resource Usage summary table):

Resource Usage
4 dsp_9bit

**Customization options** (points to the configuration panel):

- Currently selected device family: Stratix II
- General:
  - What is the number of multipliers? 2 multipliers
  - All multipliers have similar configurations
  - Add support for hardware saturation and rounding (This will force all inputs to be in Q1.15 format)
  - How wide should the A input buses be? 16 bits
  - How wide should the B input buses be? 16 bits
  - How wide should the 'result' output bus be? 33 bits
  - Create a 4th asynchronous clear input option (This forces all registers to have an associated asynchronous clear input)
  - Create an associated clock enable for each clock
- Input Representation:
  - What is the representation format for A inputs? Unsigned
  - What is the representation format for B inputs? Unsigned

Buttons: Cancel, < Back, Next >, Finish

# MegaWizard Output File Selection

MegaWizard Plug-In Manager [page 9 of 9] -- Summary

**ALTMULT\_ADD**  
Version 7.1

1 Parameter Settings 2 EDA 3 Summary

Turn on the files you wish to generate. A gray checkmark indicates a file that is automatically generated, and a red checkmark indicates an optional file. Click Finish to generate the selected files. The state of each checkbox is maintained in subsequent MegaWizard Plug-In Manager sessions.

The MegaWizard Plug-In Manager creates the selected files in the following directory:  
C:\altera\bin\Quartus II Software Design Series Foundation\QIIE7\_11Ex1\W...

File	Description
<input checked="" type="checkbox"/> my_multadd.v	Variation file
<input type="checkbox"/> my_multadd.inc	AHDL Include file
<input checked="" type="checkbox"/> my_multadd.cmp	VHDL component declaration file
<input checked="" type="checkbox"/> my_multadd.bsf	Quartus II symbol file
<input type="checkbox"/> my_multadd_inst.v	Instantiation template file
<input checked="" type="checkbox"/> my_multadd_bb.v	Verilog HDL black-box file
<input checked="" type="checkbox"/> my_multadd_syn.v	Synthesis area and timing estimation netlist

Resource Usage  
4 dsp\_9bit


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

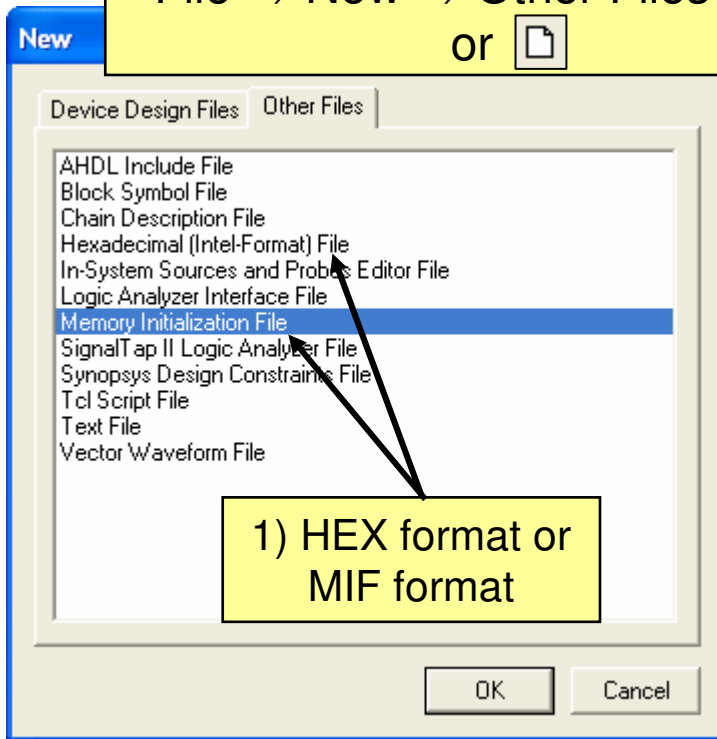
Cancel < Back Next > Finish

# Memory Editor

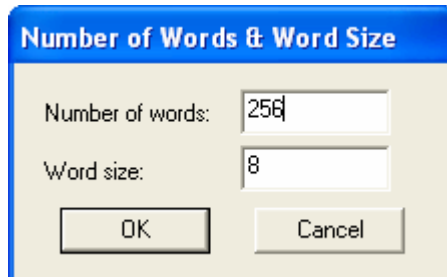
- Create or edit memory initialization files in Intel HEX (.HEX) or Altera-specific (.MIF) format
- Design entry
  - Use to initialize your memory block (ex. RAM, ROM) during power-up
- Simulation
  - Use to initialize memory blocks before simulation or after breakpoints

# Create Memory Initialization File

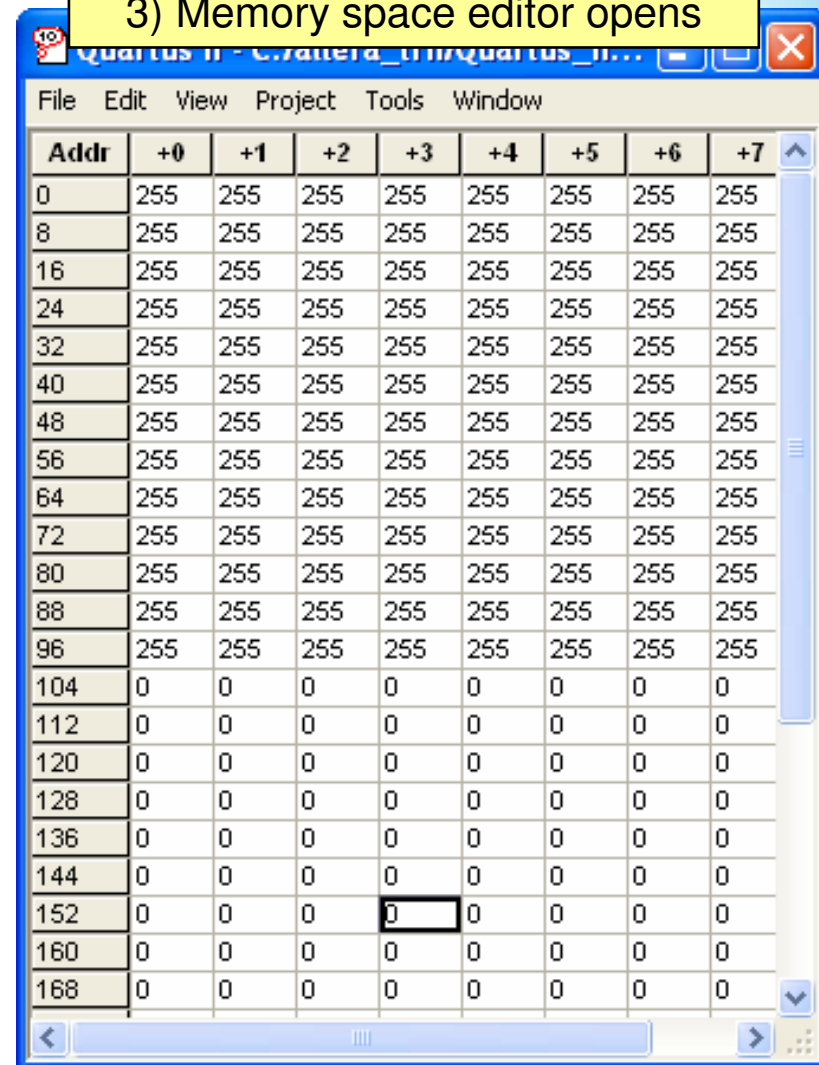
File ⇒ New ⇒ Other Files tab  
or 



2) Select Memory Size



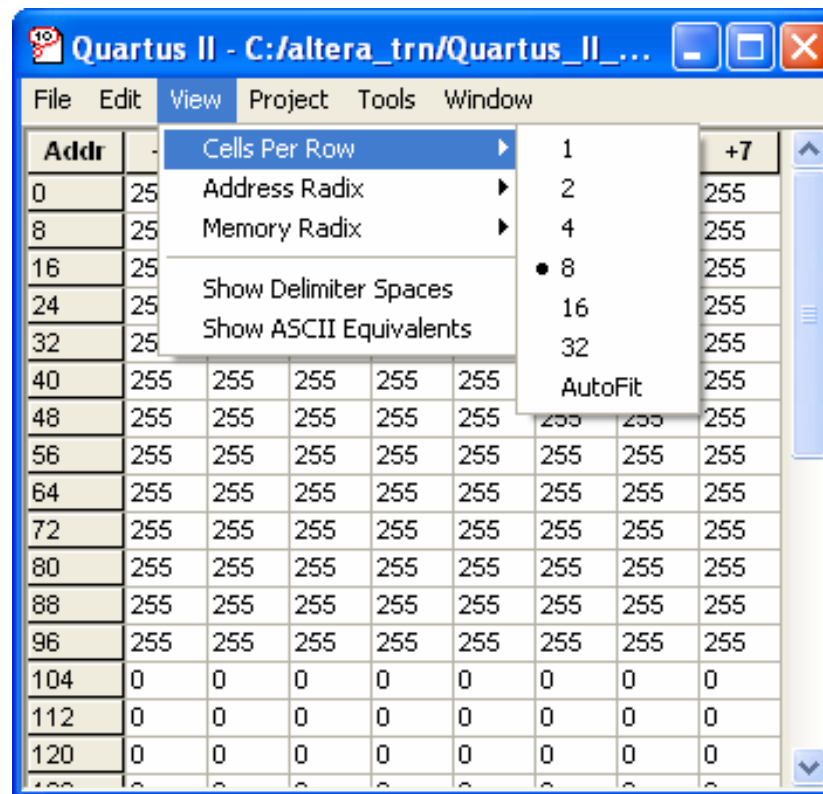
3) Memory space editor opens



Addr	+0	+1	+2	+3	+4	+5	+6	+7
0	255	255	255	255	255	255	255	255
8	255	255	255	255	255	255	255	255
16	255	255	255	255	255	255	255	255
24	255	255	255	255	255	255	255	255
32	255	255	255	255	255	255	255	255
40	255	255	255	255	255	255	255	255
48	255	255	255	255	255	255	255	255
56	255	255	255	255	255	255	255	255
64	255	255	255	255	255	255	255	255
72	255	255	255	255	255	255	255	255
80	255	255	255	255	255	255	255	255
88	255	255	255	255	255	255	255	255
96	255	255	255	255	255	255	255	255
104	0	0	0	0	0	0	0	0
112	0	0	0	0	0	0	0	0
120	0	0	0	0	0	0	0	0
128	0	0	0	0	0	0	0	0
136	0	0	0	0	0	0	0	0
144	0	0	0	0	0	0	0	0
152	0	0	0	0	0	0	0	0
160	0	0	0	0	0	0	0	0
168	0	0	0	0	0	0	0	0

# Change Options

- View options of memory editor
  - View ⇒ select from available options



# Edit Contents

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

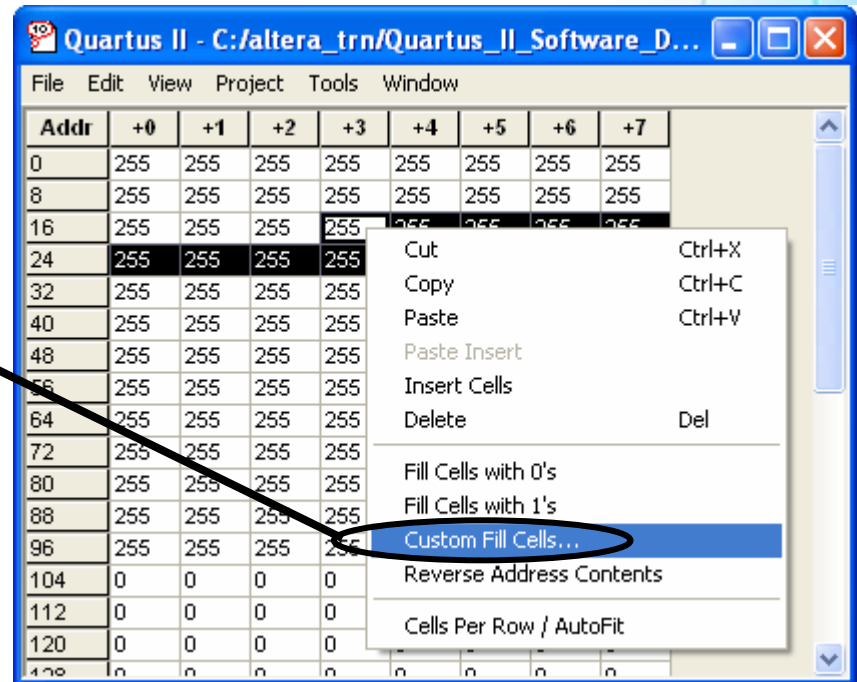
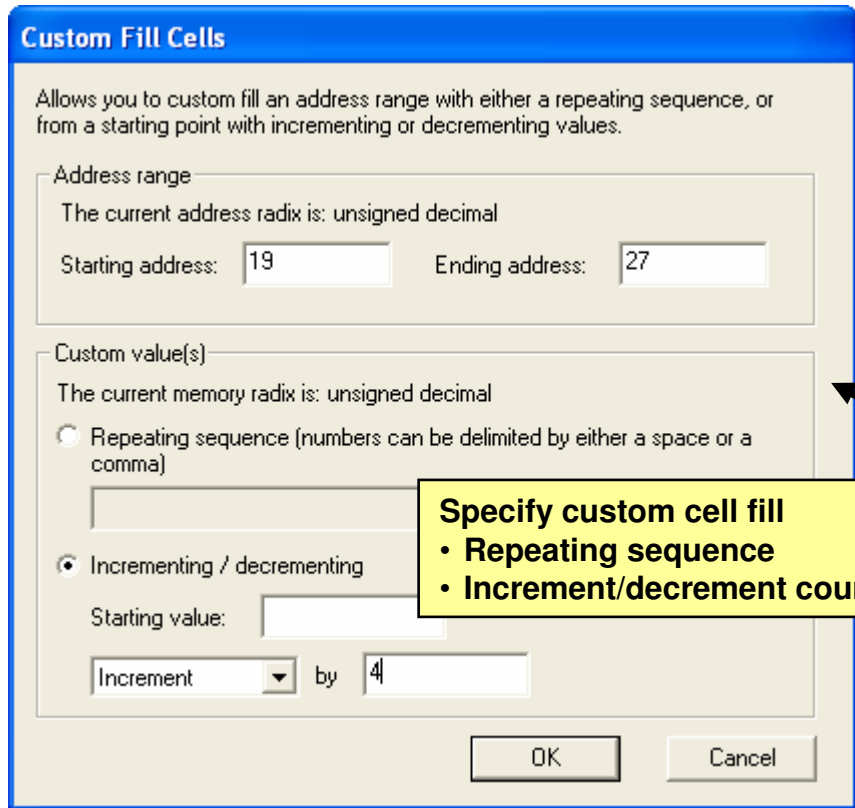
Select address location & type in a value

OR

Select the address & right-click to select fill option from menu

OR

Copy & paste from spreadsheet





# Using Memory File In Design

MegaWizard Plug-In Manager - RAM: 2-PORT [page 10 of 12]

**RAM: 2-PORT**  
Version 7.1

1 Parameter Settings | 2 EDA | 3 Summary

General > Widths/Blk Type > Clks/Rd, Byte En > Regs/Clkens/Aclrs > Output1 > Mem Init

my\_ram

data[7..0]  
wraddress[4..0]  
wren  
rdaddress[4..0]  
clock

32 Word(s) RAM

q[7..0]

Block Type: AUTO

Resource Usage  
256 ram\_bits

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

No, leave it blank  
 Initialize memory content data to XX... 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])

Browse...

File name: pipemult.hex

The initial content file should conform to which port's dimensions? PORT\_B

Cancel < Back Next > Finish

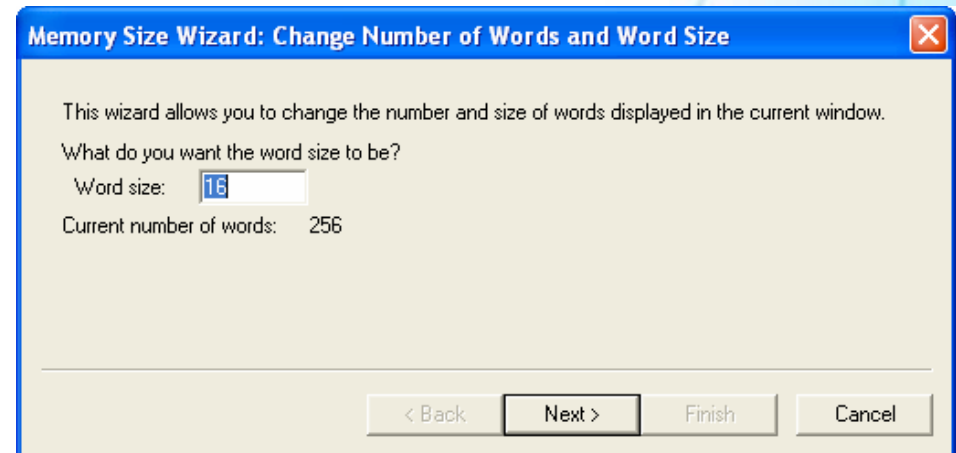
**Specify MIF or HEX file in MegaWizard**

**May also specify MIF or HEX file in HDL using the ram\_init\_file attribute**

# Memory Size Wizard

*Need to edit 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
    - Increasing word size
      - Pad words
      - Combine words
    - Decreasing word size
      - Truncate words from left
      - Truncate words from right



# EDA Interfaces Introduction

- Interface with industry-standard EDA tools that generate a netlist file
  - EDIF 2 0 0 (.EDF)
  - Verilog Quartus Mapping (.VQM)
- To import 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

# 3<sup>rd</sup>-Party Design Entry Tool Support

- Mentor Graphics®
  - LeonardoSpectrum™
  - Precision RTL Synthesis™
- Synopsys
  - Design Compiler FPGA
  - FPGA Compiler II
- Synplicity
  - Synplify
  - Synplify Pro

## *Exercise 2 Demonstration*

# Design Entry Summary

- Multiple design entry methods supported
  - Text (Verilog, VHDL, AHDL)
  - 3<sup>rd</sup>-party netlist (VQM, EDIF)
  - Schematic
- MegaWizard Plug-In Manager configures megafunctions & IP
- Memory Editor allows generation of memory initialization files
- 3<sup>rd</sup>-party EDA tools supported for design entry & synthesis

# Design Entry Support Resources

- Quartus II Handbook chapters
  - “Design Recommendations for Altera Devices” (Volume 1)
  - “Recommended HDL Coding Styles” (Volume 1)
  - 3<sup>rd</sup>-Party EDA tool chapters (Volume 1, Section 3)
- Training courses & demonstrations
  - VHDL & Verilog Basics (online courses)
  - Introduction & Advanced HDL courses

**ALTERA**

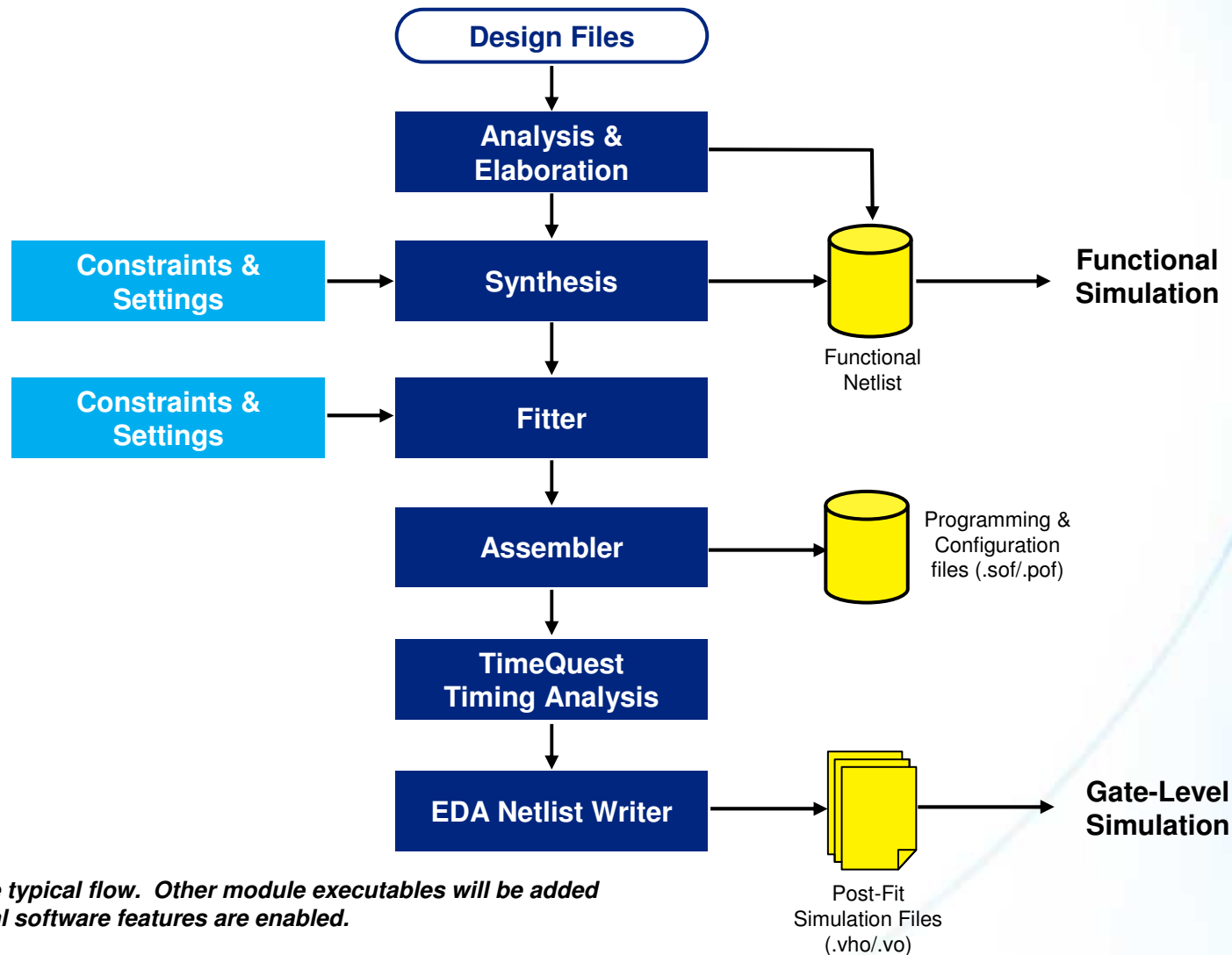
# Quartus II Software Design Series: Foundation

*Quartus II Compilation*





# Quartus II Full Compilation Flow\*



*\*This is the typical flow. Other module executables will be added if additional software features are enabled.*

© 2007 Altera Corporation—**Confidential**

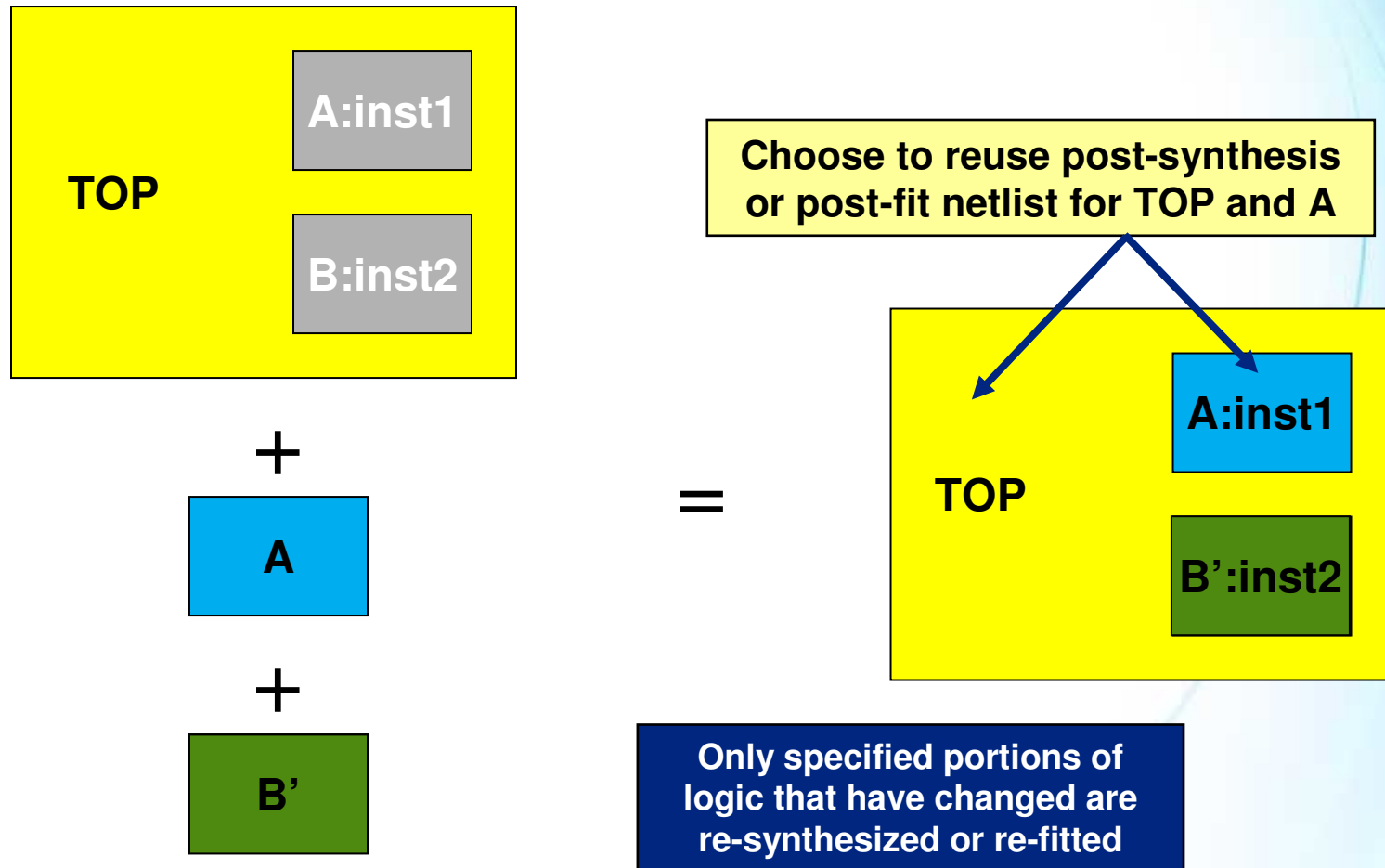
Altera, Stratix, Arria, Cyclone, MAX, HardCopy, Nios, Quartus, and MegaCore are trademarks of Altera Corporation



# Compilation Design Flows

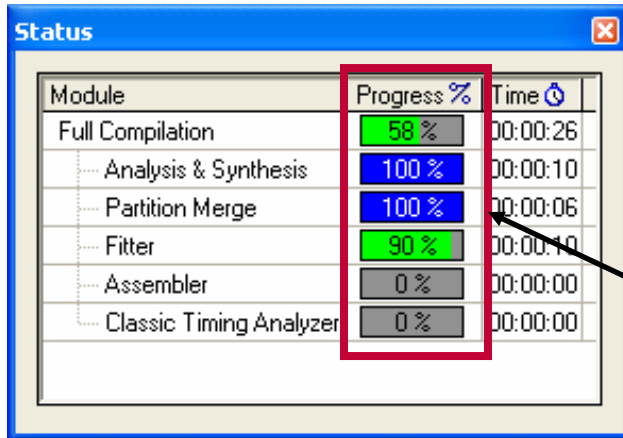
- Default “flat” compilation flow
  - Design compiled as a whole
  - Global optimizations performed
- Incremental flow (on by default for new projects)
  - User assigns design partitions
  - Each partition processed separately & results merged to form complete design
  - Netlists for partitions reused from prior successful compilation or imported from another project
    - Top-down or bottom-up flow
  - Benefits
    - Decrease compilation time
    - Preserve compilation results and timing performance
    - Enable faster timing closure

# Incremental Compilation Concept



*Note: For more details on using incremental compilation, please attend the course “Quartus II Software Design Series: Optimization” or watch the web-recording “Using Quartus II: Incremental Compilation”*

# Status & Message Windows



Module	Progress %	Time
Full Compilation	58 %	00:00:26
Analysis & Synthesis	100 %	00:00:10
Partition Merge	100 %	00:00:06
Fitter	90 %	00:00:10
Assembler	0 %	00:00:00
Classic Timing Analyzer	0 %	00:00:00



Menu Bar: View ⇒ Utility Windows

Status bars scroll to indicate progress in each stage of compilation

Message window displays informational, warning, & error messages



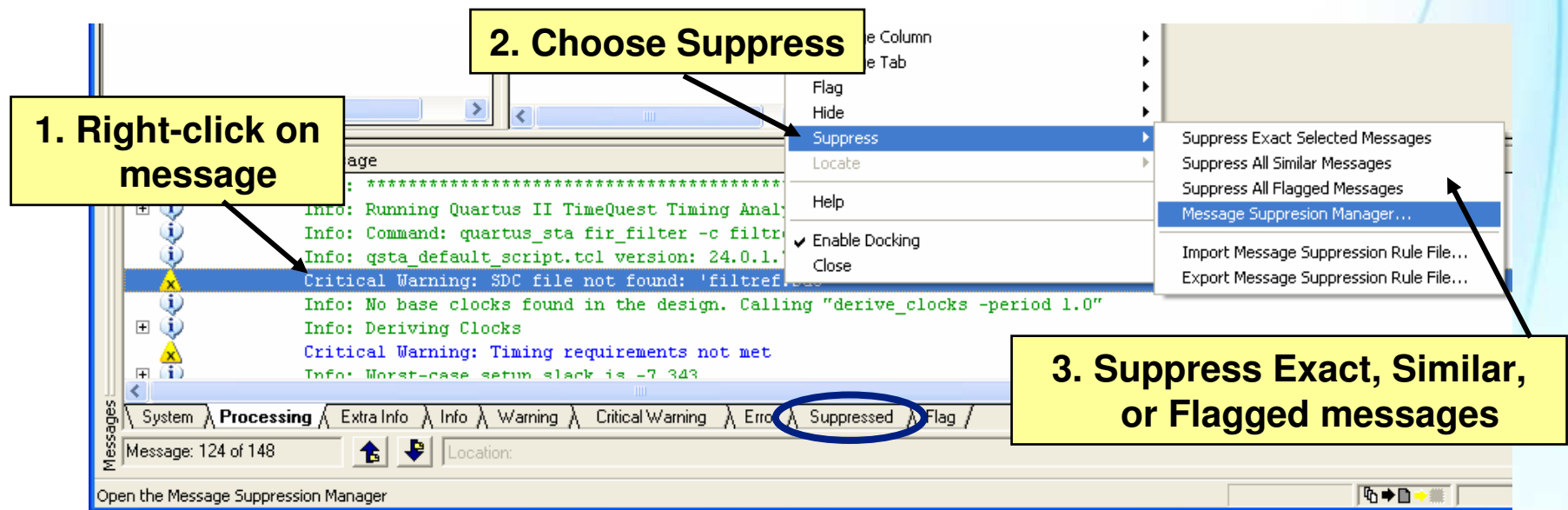
Type	Flag	Message
Info	<input type="checkbox"/>	Info: Smart recompilation skipped modul
Info	<input type="checkbox"/>	Info: *****
Info	<input type="checkbox"/>	Info: Running Quartus II Fitter
Info	<input type="checkbox"/>	Info: Command: quartus_fit --read_settings_files=on --write_settings_files=off fir_filter -c filtre
Info	<input type="checkbox"/>	Info: Selected device EP1C6F256C6 for design "filtref"



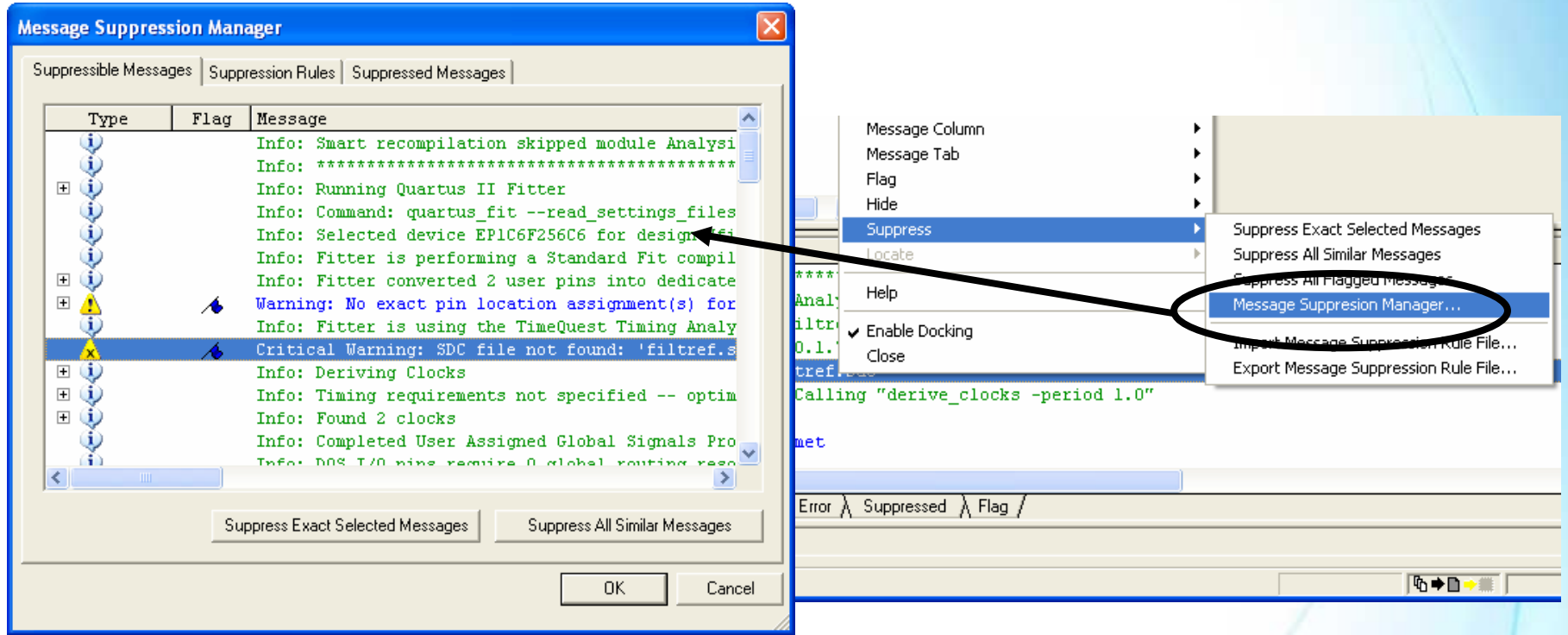
Manually flag selected messages for later review

# Message Suppression

- Hides messages from current & future compiles
  - Ex. Known synthesis warning message already investigated
- Displays suppressed messages on different tab in message window
- Stores suppression rules in <revision\_name>.SRF file



# Message Suppression Manager Tool



- Use to
  - View all suppressible messages
  - View/add/remove suppression rules
  - View messages suppressed for current & future compiles

# 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

- Graphical window containing all compilation processing information
  - Resource Usage
  - Device pin-out
  - Settings and constraints applied
  - Messages
- Opens automatically when processing begins
- *Recommendation:* Go through report for a design to get sense of information being provided
- Information also available as text files in project directory
  - Ex. *<project\_name>.fit.rpt* & *<project\_name>.map.rpt*

# Compilation Report



**Window => Detach Window to release from Quartus II workspace**

Flow Status	Successful - Thu May 03 17:30:51 2007
Quartus II Version	7.1 Build 156 04/30/2007 SJ Full Version
Revision Name	filtref
Top-level Entity Name	filtref
Family	Cyclone
Device	EP1C6F256C6
Timing Models	Final
Met timing requirements	N/A
Total logic elements	162 / 5,980 ( 3 % )
Total pins	22 / 185 ( 12 % )
Total virtual pins	0
Total memory bits	0 / 92,160 ( 0 % )
Total PLLs	0 / 2 ( 0 % )

**Each compiler executable generates separate folder**

# Example: Source Files Read

Quartus II - D:/altera/71/qdesigns/fir\_filter/fir\_filter - filtref - [Compilation Report - Analysis & Synthesis Source Files Read]

File Edit View Tools Window

Analysis & Synthesis Source Files Read

	File Name with User-Entered Path	Used in Netlist	File Type	File Name with Absolute Path
1	mult.v	yes	User Verilog HDL File	D:/altera/71/qdesigns/fir_filter/mult.v
2	accum.v	yes	User Verilog HDL File	D:/altera/71/qdesigns/fir_filter/accum.v
3	filtref.bdf	yes	User Block Diagram/Schematic File	D:/altera/71/qdesigns/fir_filter/filtref.bdf
4	hvalues.v	yes	User Verilog HDL File	D:/altera/71/qdesigns/fir_filter/hvalues.v
5	taps.v	yes	User Verilog HDL File	D:/altera/71/qdesigns/fir_filter/taps.v
6	state_m.v	yes	User Verilog HDL File	D:/altera/71/qdesigns/fir_filter/state_m.v
7	acc.v	yes	User Verilog HDL File	D:/altera/71/qdesigns/fir_filter/acc.v
8	lpm_add_sub.tdf	yes	Megafunction	d:/altera/71/quartus/libraries/megafunctions/lpm_add_sub.tdf
9	addcore.inc	yes	Megafunction	d:/altera/71/quartus/libraries/megafunctions/addcore.inc
10	look_add.inc	yes	Megafunction	d:/altera/71/quartus/libraries/megafunctions/look_add.inc
11	bypassff.inc	yes	Megafunction	d:/altera/71/quartus/libraries/megafunctions/bypassff.inc
12	altshift.inc	yes	Megafunction	d:/altera/71/quartus/libraries/megafunctions/altshift.inc
13	alt_stratix_add_sub.inc	yes	Megafunction	d:/altera/71/quartus/libraries/megafunctions/alt_stratix_add_sub.inc
14	alt_mercury_add_sub.inc	yes	Megafunction	
15	aglobal71.inc	yes	Megafunction	
16	addcore.tdf	yes	Megafunction	
17	a_csnbuffer.inc	yes	Megafunction	
18	a_csnbuffer.tdf	yes	Megafunction	
19	altshift.tdf	yes	Megafunction	
20	lpm_mult.tdf	yes	Megafunction	
21	lpm_add_sub.inc	yes	Megafunction	
22	multcore.inc	yes	Megafunction	d:/altera/71/quartus/libraries/megafunctions/multcore.inc
23	multcore.tdf	yes	Megafunction	d:/altera/71/quartus/libraries/megafunctions/multcore.tdf
24	csa_add.inc	yes	Megafunction	d:/altera/71/quartus/libraries/megafunctions/csa_add.inc
25	mpar_add.inc	yes	Megafunction	d:/altera/71/quartus/libraries/megafunctions/mpar_add.inc
26	muleabz.inc	yes	Megafunction	d:/altera/71/quartus/libraries/megafunctions/muleabz.inc
27	mul_lfrg.inc	yes	Megafunction	d:/altera/71/quartus/libraries/megafunctions/mul_lfrg.inc
28	mul_boothc.inc	yes	Megafunction	d:/altera/71/quartus/libraries/megafunctions/mul_boothc.inc
29	alt_ded_mult.inc	yes	Megafunction	d:/altera/71/quartus/libraries/megafunctions/alt_ded_mult.inc
30	alt_ded_mult_y.inc	yes	Megafunction	d:/altera/71/quartus/libraries/megafunctions/alt_ded_mult_y.inc
31	dffpipe.inc	yes	Megafunction	d:/altera/71/quartus/libraries/megafunctions/dffpipe.inc
32	mpar_add.tdf	yes	Megafunction	d:/altera/71/quartus/libraries/megafunctions/mpar_add.tdf

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

# Example: Resource Usage

Quartus II - D:/altera/71/qdesigns/fir\_filter/fir\_filter - filtref - [Compilation Report - Fitter Resou...]

File Edit View Tools Window

Compilation Report  
 Legal Notice  
 Flow Summary  
 Flow Settings  
 Flow Non-Default Global Settings  
 Flow Elapsed Time  
 Flow Log  
 Analysis & Synthesis  
 Fitter  
 Summary  
 Settings  
 Netlist Optimizations  
 Pin-Out File  
 Resource Section  
 Resource Usage Summary  
 Input Pins  
 Output Pins  
 I/O Bank Usage  
 All Package Pins  
 Output Pin Default Load For Reported  
 Resource Utilization by Entity  
 Delay Chain Summary  
 Pad To Core Delay Chain Fanout  
 Control Signals  
 Global & Other Fast Signals  
 Non-Global High Fan-Out Signals  
 Logic and Routing Section  
 Device Options  
 Advanced Fitter Data  
 Messages  
 Suppressed Messages  
 Assembler  
 TimeQuest Timing Analyzer

Fitter Resource Usage Summary		Usage
	Resource	
1	Total logic elements	162 / 5,980 ( 3 % )
2	-- Combinational with no register	77
3	-- Register only	57
4	-- Combinational with a register	28
5		
6	Logic element usage by number of inputs	
7	-- 4 input functions	
8	-- 3 input functions	
9	-- 2 input functions	
10	-- 1 input functions	
11	-- 0 input functions	43
12		
13	Logic elements by mode	
14	-- normal mode	135
15	-- arithmetic mode	27
16	-- qfbk mode	9
17	-- register cascade mode	0
18	-- synchronous clear/load mode	52
19	-- asynchronous clear/load mode	39
20		
21	Total registers	85 / 6,523 ( 1 % )
22	Total LABs	26 / 598 ( 4 % )
23	Logic elements in carry chains	30
24	User inserted logic elements	0
25	Virtual pins	0
26	I/O pins	22 / 185 ( 12 % )
27	-- Clock pins	2 / 2 ( 100 % )
28	Global signals	3

Several tables in Resource Section detail how much of FPGA resources available and used

# Netlist Viewers

## ■ RTL Viewer

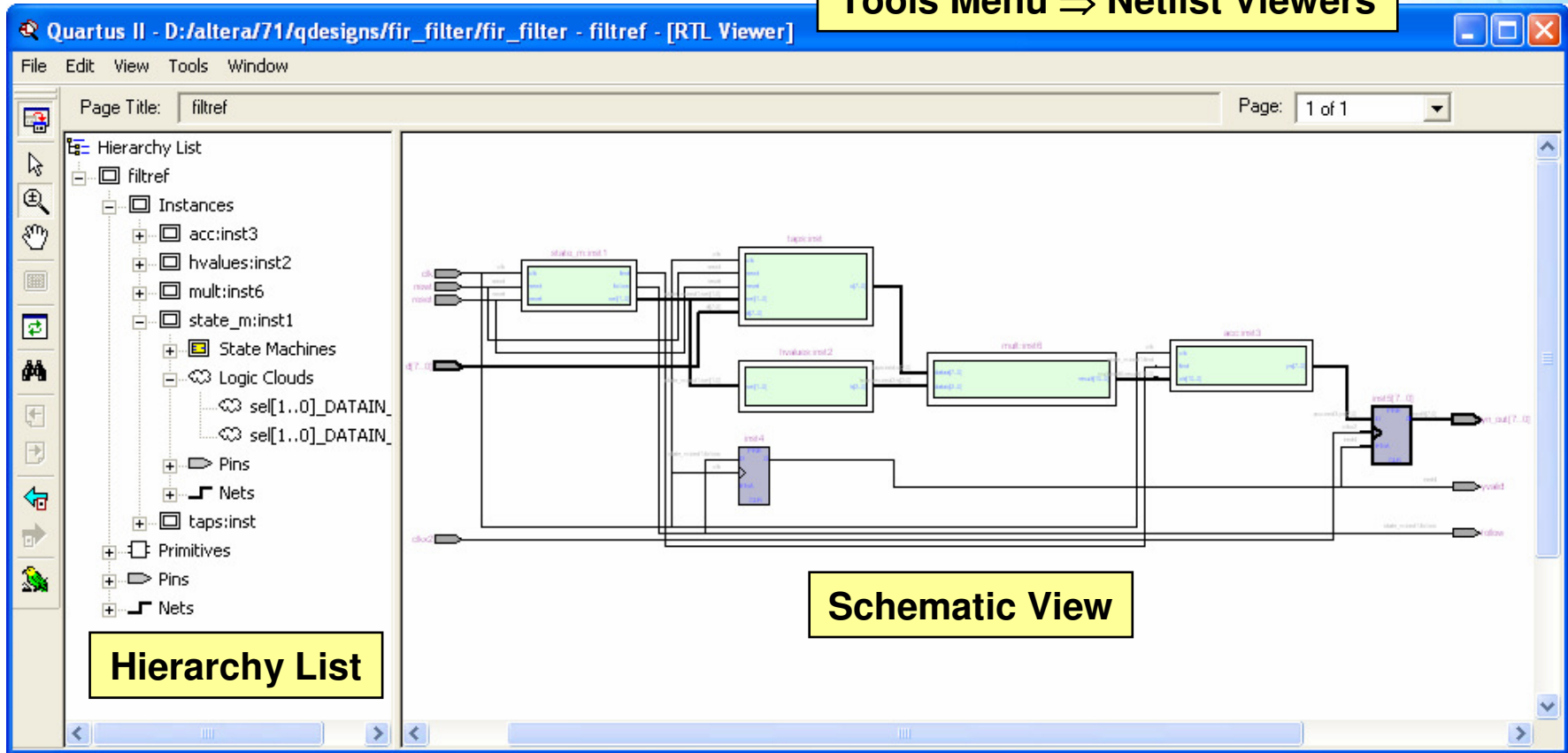
- Graphically represents results of synthesis
- Visually check initial HDL synthesis results
  - Before any Quartus II optimizations
- Locate synthesized nodes for assigning constraints
- Debug verification issues

## ■ Technology Map Viewers (Post-Mapping & regular)

- Graphically represents results of mapping (post-synthesis) & fitting
- Analyze critical timing paths graphically
  - Delay values displayed if timing
- Locate nodes & node names after optimizations
  - Assigning constraints
  - Debugging

# RTL Viewer

Tools Menu ⇒ Netlist Viewers



**Note:**

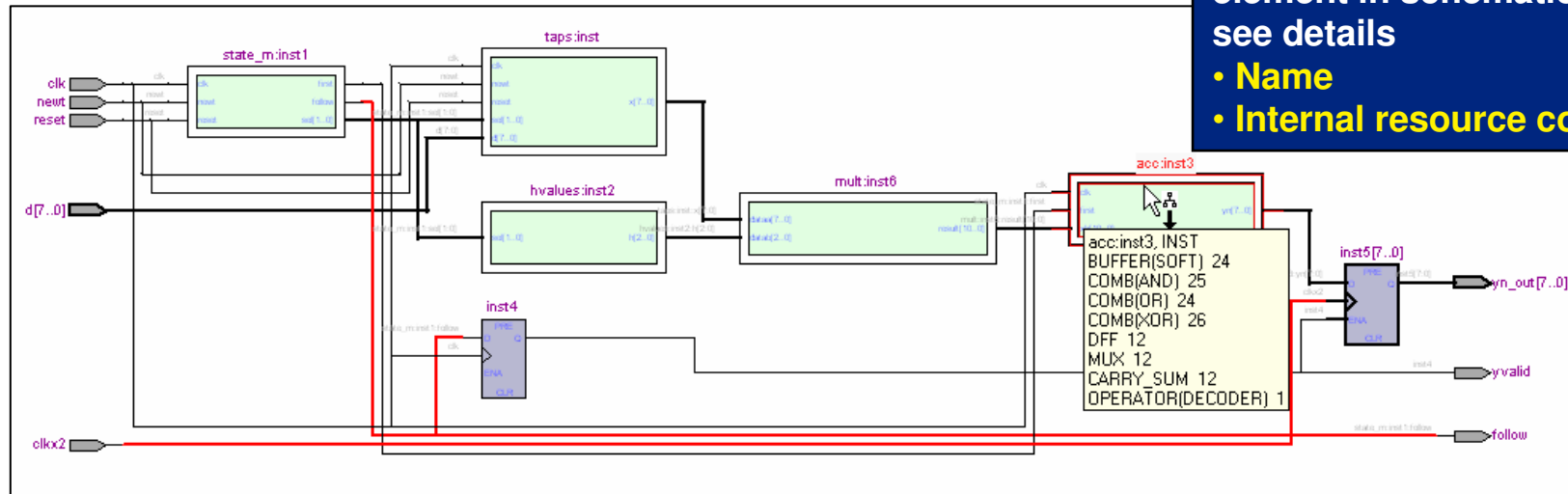
- 1) **Must Perform Elaboration First (e.g. Analysis & Elaboration OR Analysis & Synthesis)**

© 2007 Altera Corporation—**Confidential**

Altera, Stratix, Arria, Cyclone, MAX, HardCopy, Nios, Quartus, and MegaCore are trademarks of Altera Corporation



# Schematic View (RTL Viewer)



Place pointer over any element in schematic to see details

- Name
- Internal resource count

## ■ Represents design using logic blocks & nets

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

# Technology Map Viewers

Tools Menu ⇒ Netlist Viewers

The screenshot shows the Quartus II Technology Map Viewer interface. The window title is "Quartus II - D:/altera/71/qdesigns/fir\_filter/fir\_filter - filtref - [Technology Map Viewer - Post-Fitting]". The menu bar includes File, Edit, View, Tools, and Window. The page title is "Post-Fitting: acc:inst3" and the page number is "1 of 1". On the left, the Hierarchy List shows a tree structure starting with "filtref", containing "Instances" (acc:inst3, mult:inst6, state\_m:inst1, taps:inst) and "Primitives". On the right, the Schematic View displays a complex circuit diagram with numerous logic blocks and interconnecting lines. A yellow box labeled "Tools Menu ⇒ Netlist Viewers" is positioned at the top right of the window. Another yellow box labeled "Hierarchy List" is at the bottom left, and a third yellow box labeled "Schematic View" is at the bottom right.

**Note:**

1) **Must Run Synthesis and/or Fitting First**

© 2007 Altera Corporation—**Confidential**

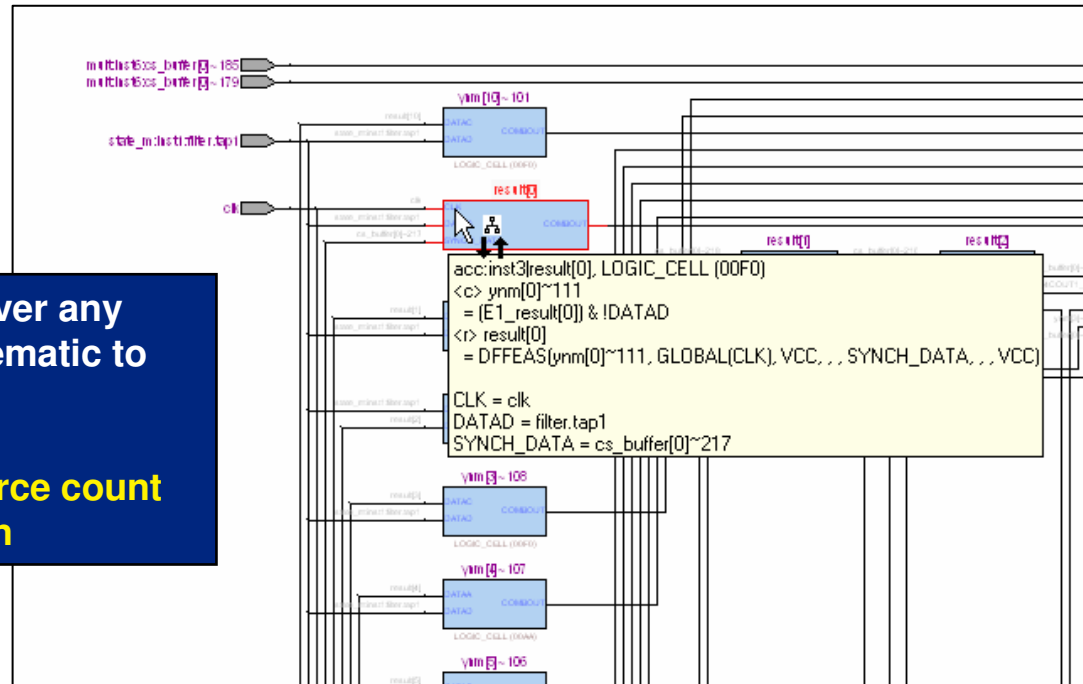
Altera, Stratix, Arria, Cyclone, MAX, HardCopy, Nios, Quartus, and MegaCore are trademarks of Altera Corporation



# Schematic View (Technology Viewer)

Place pointer over any element in schematic to see details

- Name
- Internal resource count
- Logic equation

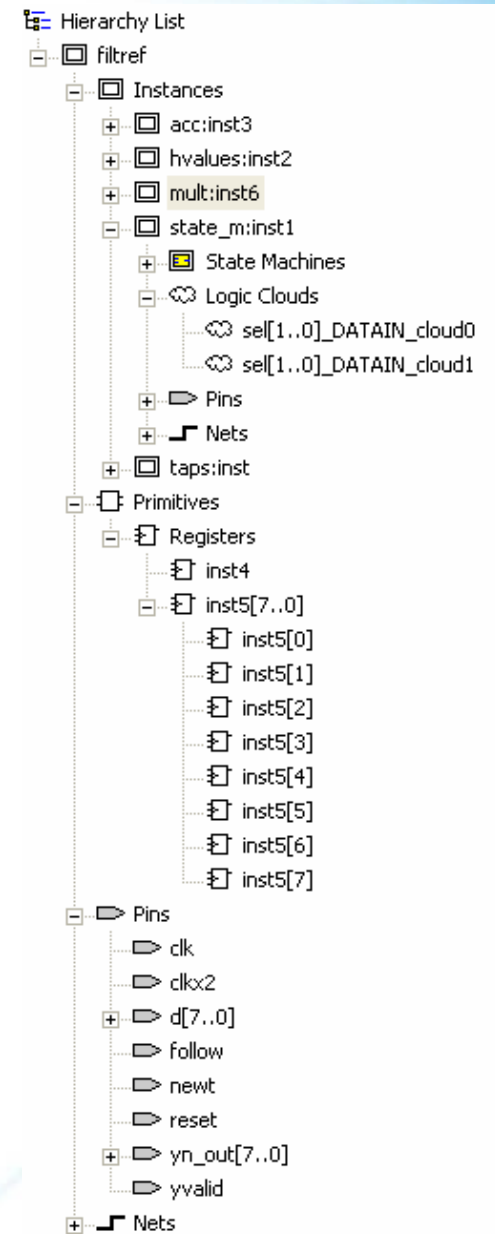


## ■ Represents design using atoms

- I/O pins & cells
- Lcells
- Memory blocks
- MAC (DSP blocks)

# 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

Quartus II - D:/altera/71/qdesigns/fir\_filter/fir\_filter - filtref - [RTL Viewer]

File Edit View Tools Window

Page Title: filtref Page: 1 of 1

**Hierarchy List**

- [-] filtref
  - [-] Instances
    - [-] acc:inst3
    - [-] hvalues:inst2
    - [-] mult:inst6
      - [-] Instances
      - [-] Pins
      - [-] Nets
      - [-] state\_m:inst1

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

**Expanding instances shows instances, pins & nets within internal modules**

# Schematic Hierarchy Navigation

- Mouse pointer indicates action

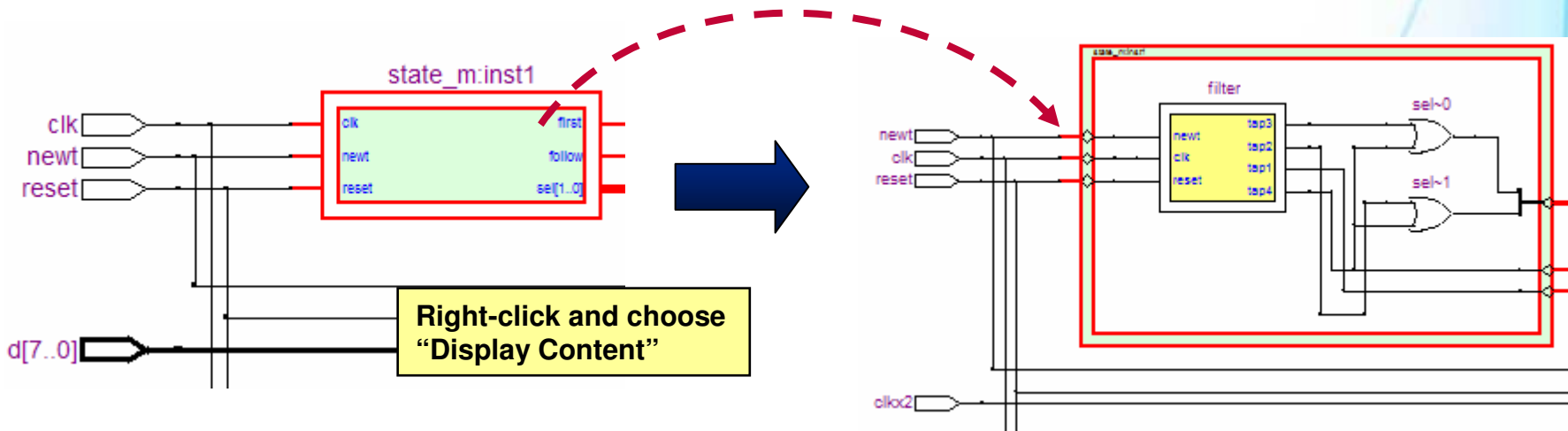


- ← Descending hierarchy
  - Double-click on instance
  - Right-click & select **hierarchy down**



- ← Ascending hierarchy
  - Double-click in empty space
  - Right-click & select **hierarchy up**

- Expand instances within schematic



# Filter Schematic

The image shows the Quartus II schematic editor interface. On the left is the Hierarchy List, showing a tree structure of components including 'taps:inst' and its sub-components like 'Operators' (Mux0-Mux7) and 'Registers'. The main workspace displays a complex schematic with many components and connections. A right-click context menu is open over a component, with the 'Filter' option selected. A secondary menu is visible, listing options: Sources, Destinations, Sources & Destinations, Selected Nodes & Nets (highlighted), Between Selected Nodes, and Bus Index. A third menu is open to the right, showing a vertical list of schematic components, with a green box highlighting a subset of them.

**Unfiltered: All components & paths shown**

**Right-click for filter menu**

**Filtered: Only selected components & related paths displayed**

**Filter options**

# Other Features

- Bird's Eye View
  - Displays overall view of design
- Page control
  - Hierarchical levels automatically partitioned
    - Control design size per page (tools ⇒ customize ⇒ options)
  - Navigate nets between page
- Go to net driver
  - Traces net back to source driver
- LUT internal detail
  - View LUT truth table, Karnaugh map, and expanded to gate logic
- Cross-probing : locate nodes from/to
  - Design files
  - Assignment Editor
  - Chip Planner
  - Chip Editor
  - Resource Property Editor
  - RTL/Technology Map Viewers

# State Machine Viewer

Tools menu ⇒ Netlist Viewers

The screenshot shows the State Machine Viewer interface. At the top, the title bar reads "Quartus II - D:/altera/71/qdesigns/fir\_filter/fir\_filter - filtref - [State M...". Below the title bar is a menu bar with "File", "Edit", "View", "Tools", and "Window". The main window area is titled "State Machine: [filtref]state\_m:inst1filter".

The top section displays a "State Flow Diagram" with five states: "idle", "tap1", "tap4", "tap2", and "tap3". The "tap4" state is highlighted with a red circle. A blue arrow points from a yellow callout box to this state. The diagram shows transitions between these states, including a "reset" input to the "idle" state.

The bottom section displays a "State Transition/Encoding Table" with the following data:

State	Condition		
1	idle	idle	(!newt)
2	idle	tap1	(newt)
3	tap4	idle	(!newt)
4	tap4	tap1	(newt)
5	tap3	tap4	
6	tap2	tap3	
7	tap1	tap2	

A blue arrow points from a yellow callout box to the "tap4" row in the table. At the bottom of the window, there are tabs for "Transitions" and "Encoding".

Highlighting state in state transition table highlights corresponding state in state flow diagram

Use drop-down to select state machine

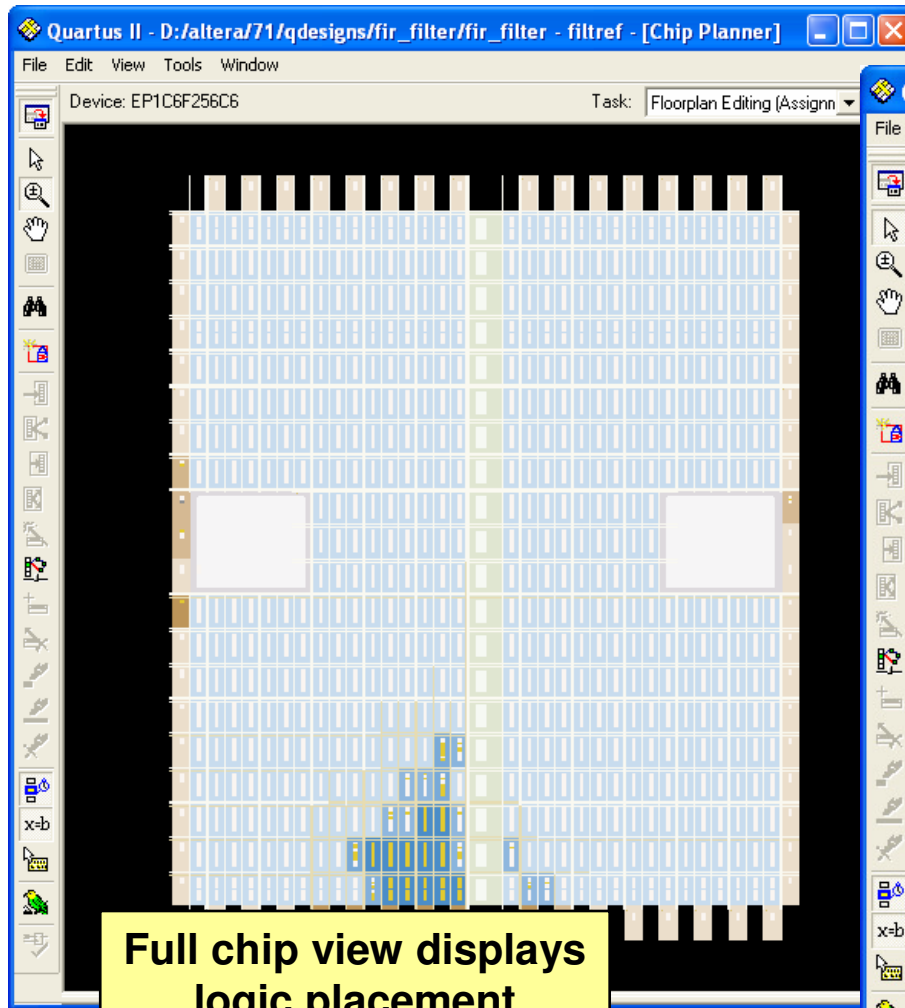
State Transition/Encoding Table

# Chip Planner

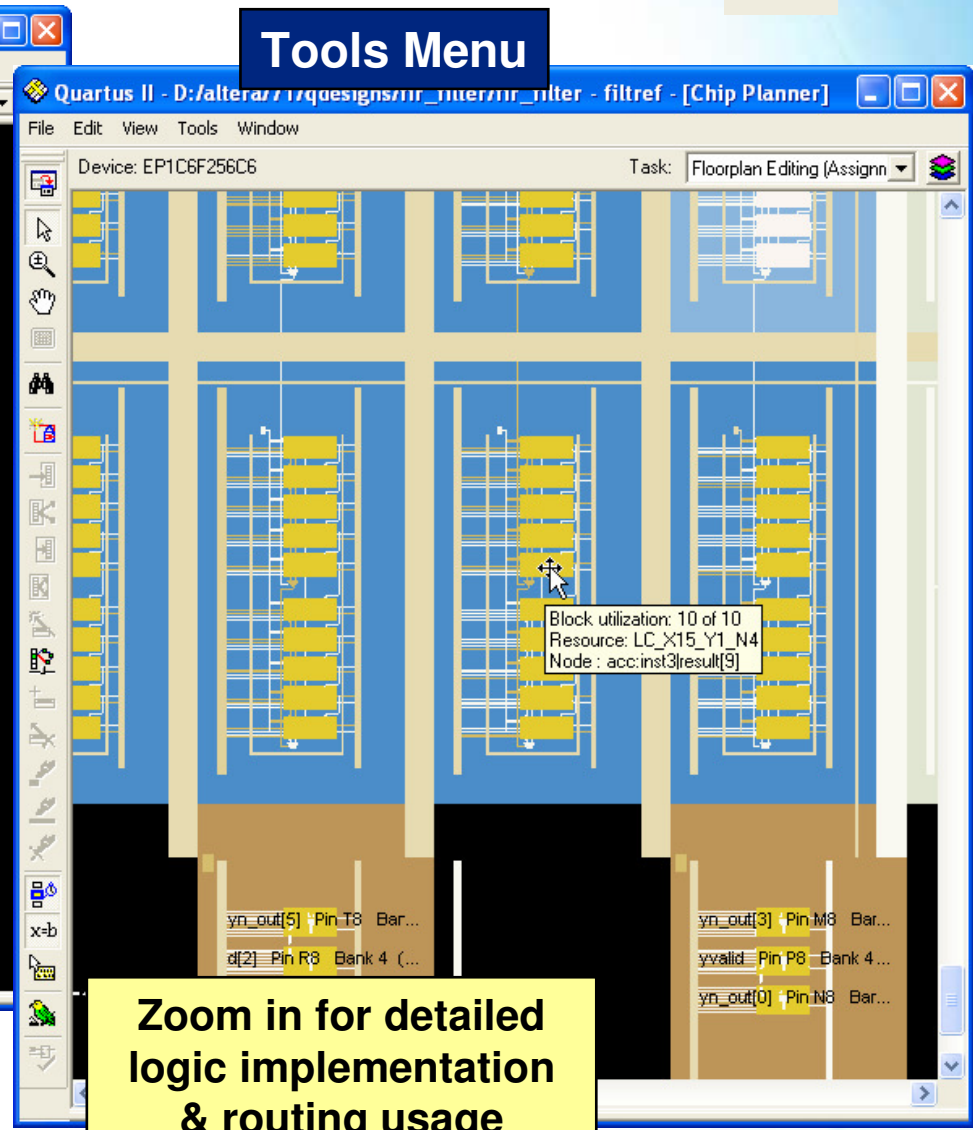
- Editable graphical view of target device
- Displays
  - Graphical layout of device resources
  - Routing channels between device resources
    - Internal routing channels within LABs
- Uses
  - View placement of design logic
  - View connectivity between resources used in design
  - Make placement assignments
  - Debugging placement related issues



# Chip Planner



Full chip view displays logic placement



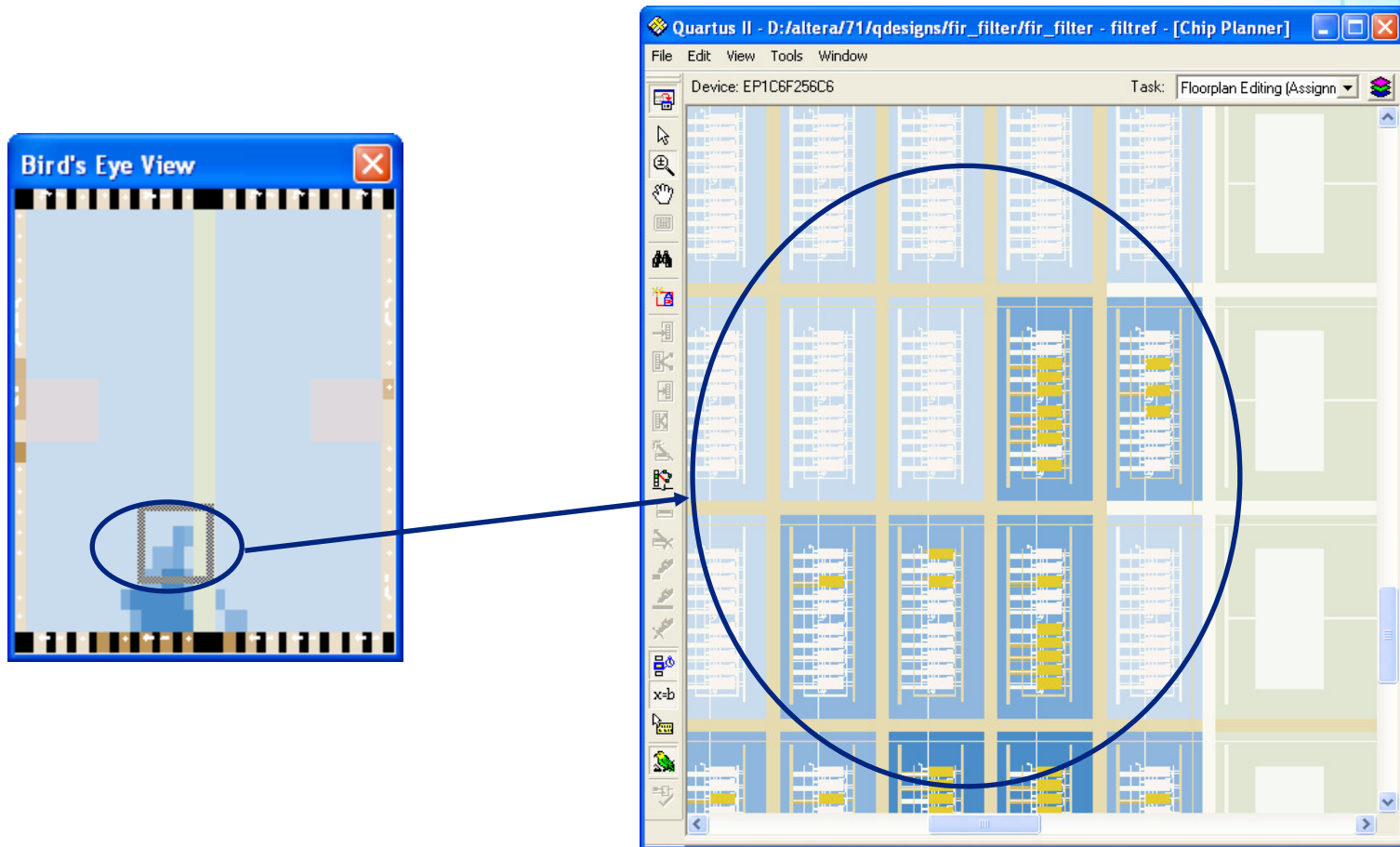
Zoom in for detailed logic implementation & routing usage

Tools Menu

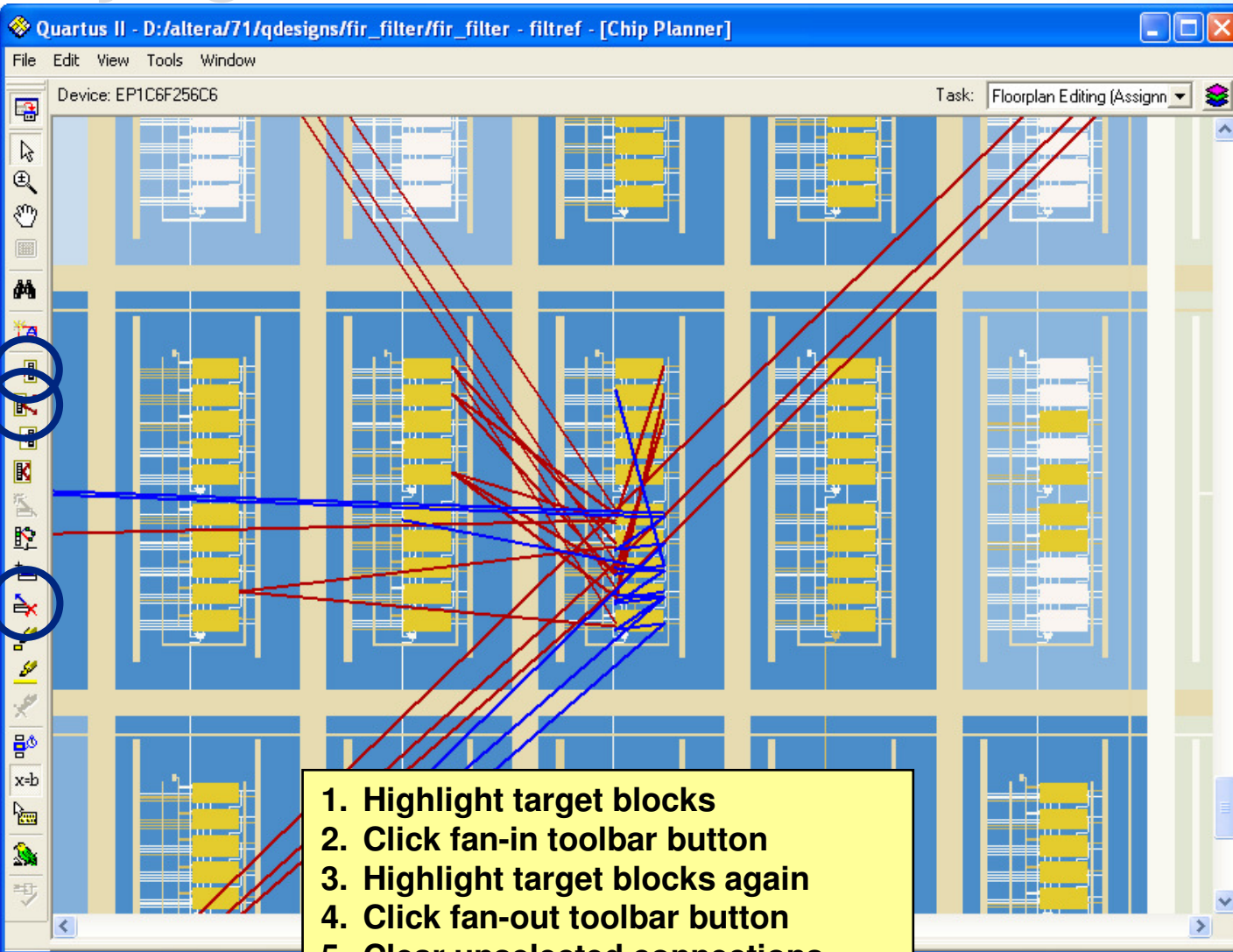
# Bird's Eye View



- Provides overall view of the entire device
- Use to navigate through the Chip Planner Floorplan



# Displaying Fan-In & Fan-Out

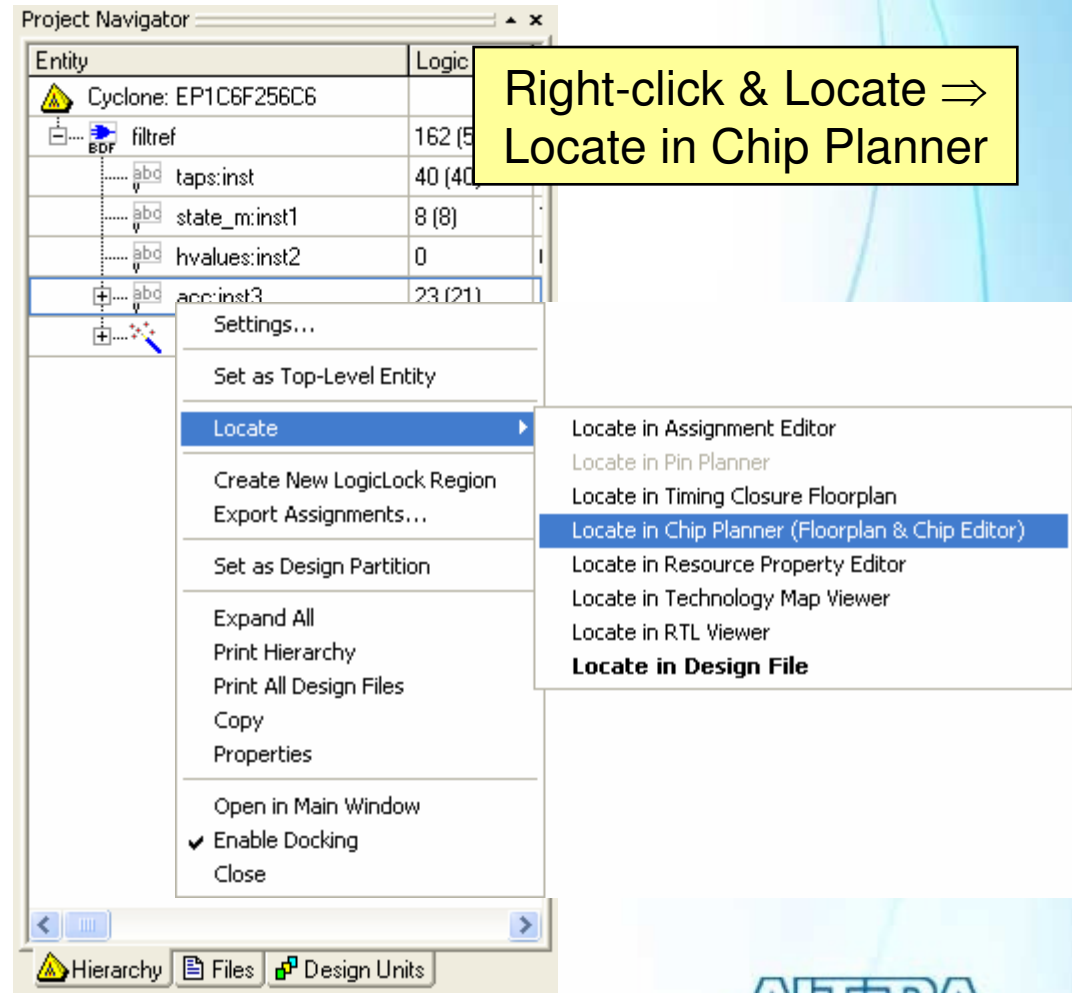


The screenshot shows the Quartus II Chip Planner interface for a device EP1C6F256C6. The main workspace displays a grid of logic blocks. A central block is highlighted in yellow, and numerous red lines radiate from it to other blocks, representing fan-out connections. Blue lines also connect various blocks, representing fan-in connections. The toolbar on the left has two buttons circled in blue, which correspond to the fan-in and fan-out tools mentioned in the instructions. A yellow text box at the bottom provides a five-step procedure for displaying these connections.

1. Highlight target blocks
2. Click fan-in toolbar button
3. Highlight target blocks again
4. Click fan-out toolbar button
5. Clear unselected connections

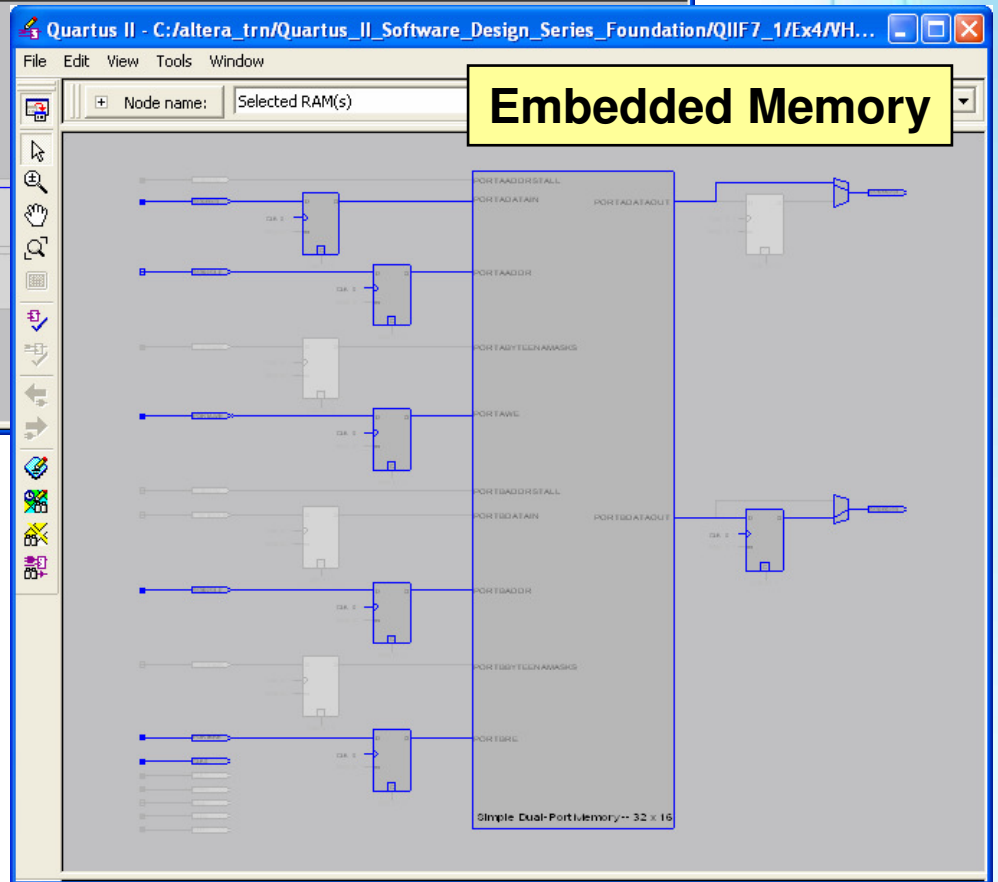
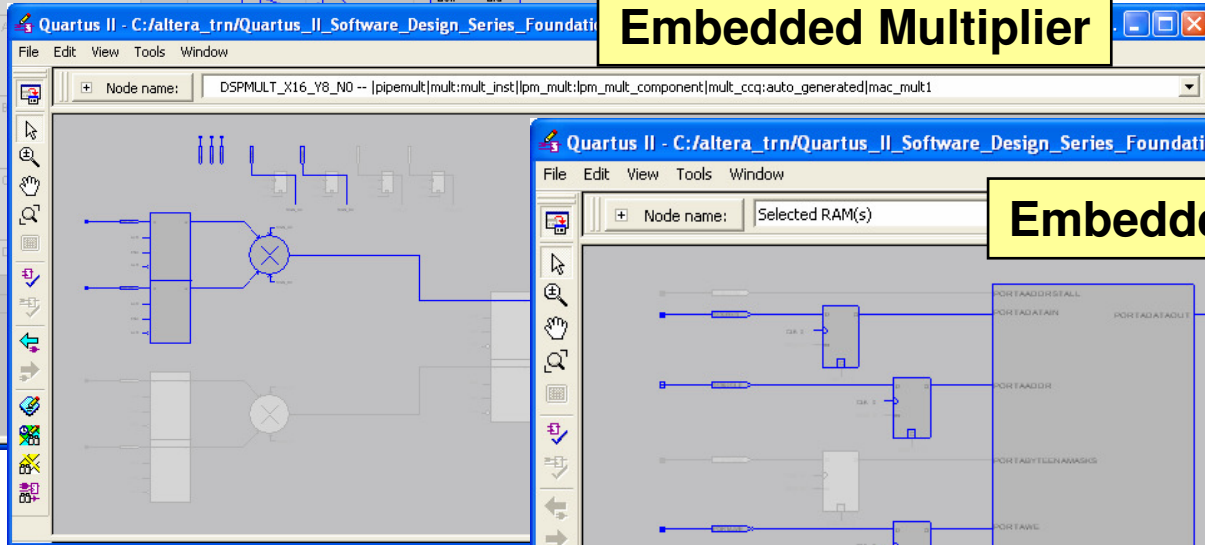
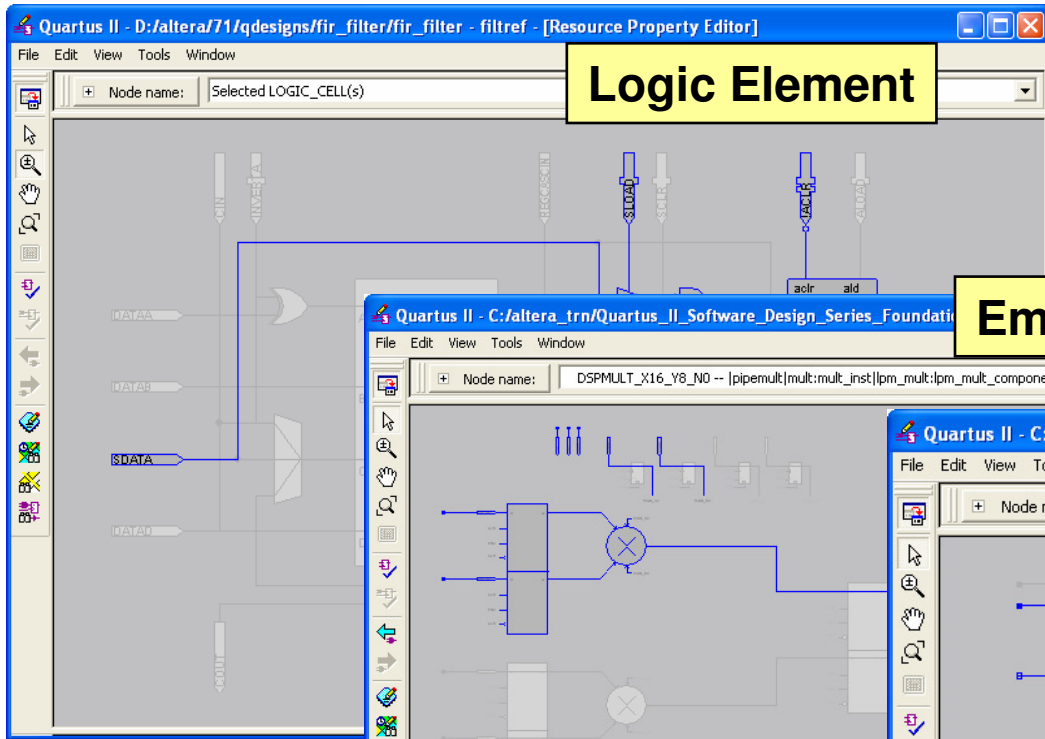
# Cross-Probing from/to Chip Planner

- Locate hierarchy blocks or specific logic from other Quartus II windows
- Project Navigator
- Compilation Report
- Design files
- RTL Viewer
- Technology Viewer
- Message window
- Pin Planner



# Resource Property Editor

- Use to view detailed logic implementation & connections
  - Cross-probe from other Quartus II windows
- Views
  - Logic cells (look-up tables & registers)
  - Embedded memory
  - Embedded multipliers
  - I/O cells
  - PLLs



# Undesired Compilation Results?

- Incorrect coding
- Non-recommended coding style
- Suboptimal synthesis & fitting settings/constraints

*Note: For more details on optimizing designs based on undesired results, please attend the course  
“Quartus II Software Design Series: Optimization”*

## *Exercise 3 Demonstration*



# Compilation Summary

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

# Compilation Support Resources

- Quartus II Handbook chapters
  - “Quartus II Incremental Compilation for Hierarchical & Team-Based Design” (Volume 1)
  - “Design Analysis & Engineering Change Management with Chip Planner” (Volume 3)

**ALTERA**

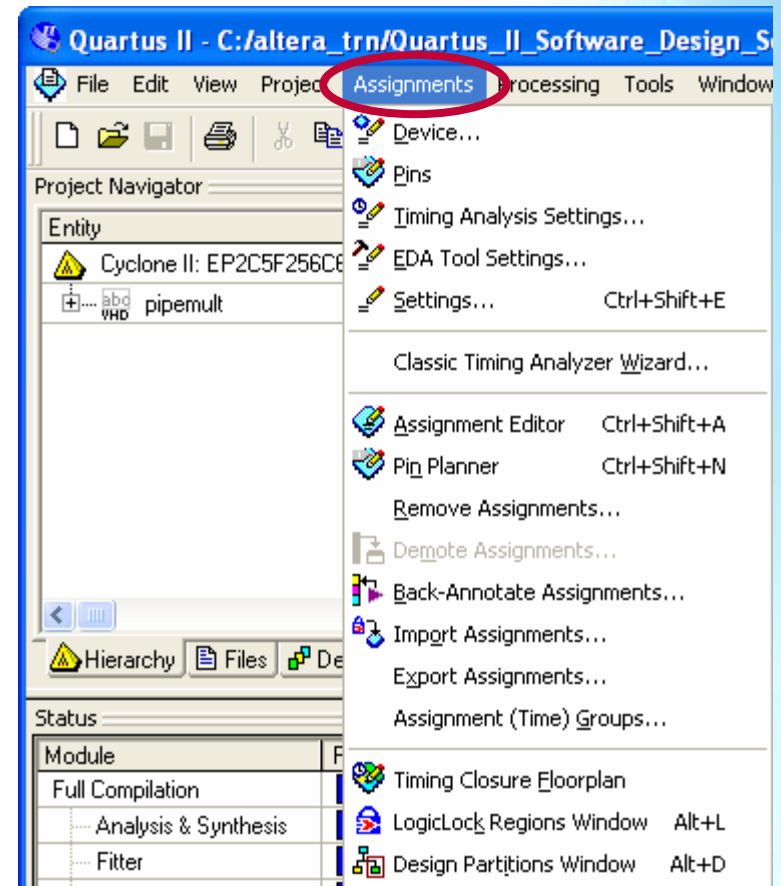
# Quartus II Software Design Series: Foundation

*Settings & Assignments*



# Synthesis & Fitting Control

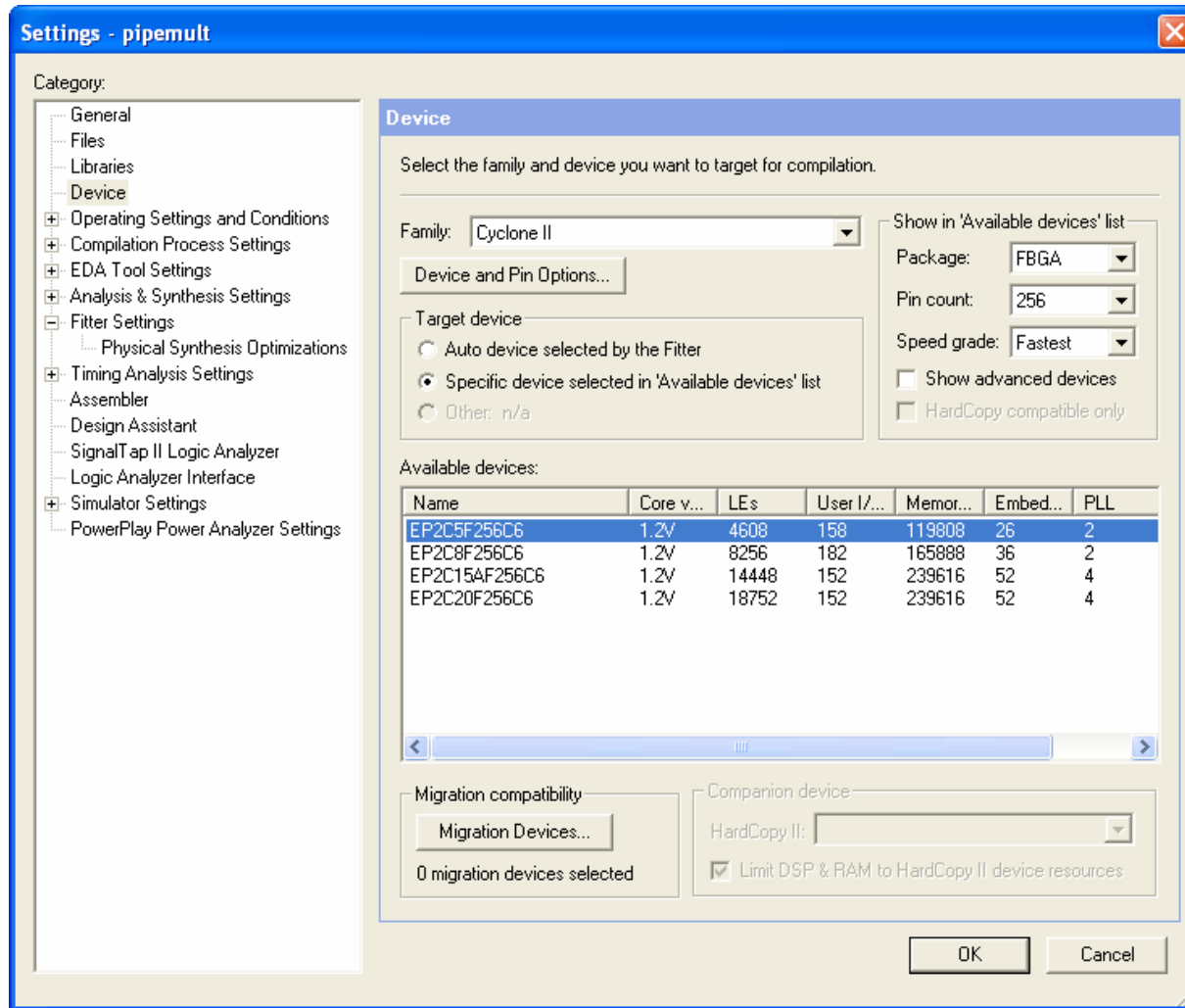
- Controlled using two methods
  - Settings
    - Project-wide switches
  - Assignments (i.e. logic options; constraints)
    - Individual entity/node controls
- Accessed using Assignments menu
- Stored in QSF file for project/revision



# Settings

- Project-wide switches
- Examples
  - Device selection
  - Synthesis optimization
  - Fitter settings
  - Physical synthesis
  - Design Assistant
- Located in Settings dialog box (Assignments menu)

# Settings Dialog Box



## Change settings

- Top-level entity
- Target device
- Add/remove files
- Libraries
- VHDL '87 or '93?
- Verilog '95, '01 or SystemVerilog?
- EDA tool settings
- Timing settings
- Compiler settings
- Synthesis settings
- Fitter settings
- Simulator settings

***Tcl: set\_global\_assignment -name <assignment\_name\*> <value>***

# Compilation Process

Settings - pipemult

Category:

- General
- Files
- Libraries
- Device
- + Operating Settings and Conditions
  - **Compilation Process Settings** (1)
  - Early Timing Estimate
  - Incremental Compilation
- + EDA Tool Settings
- + Analysis & Synthesis Settings
- Filter Settings
  - Physical Synthesis Optimizations
- + Timing Analysis Settings
  - Assembler
  - Design Assistant
  - SignalTap II Logic Analyzer
  - Logic Analyzer Interface
- + Simulator Settings
  - PowerPlay Power Analyzer Settings

**Compilation Process Settings**

Specify Compilation Process options.

Maximum processors allowed for parallel compilation:

Use smart compilation (2)

Preserve fewer node names to save disk space

Run I/O assignment analysis before compilation

Run Assembler during compilation

Run RTL Viewer preprocessing during compilation

Save a node-level netlist of the entire design into a persistent source file  
(This option specifies VQM File name for full compilation and Start VQM Writer command)

File name:  ...

Export version-compatible database (2)

Export directory:  ...

Save project output files in specified directory

Directory name:

More Settings...

Description:

- **Smart compilation<sup>(1)</sup>**
  - Skips entire compiler modules when not required (i.e. elaboration, synthesis, etc.)
  - Saves compiler time
  - Uses more disk space
- **Generate version-compatible database<sup>(2)</sup>**

```
Tcl: set_global_assignment -name SMART_RECOMPILE ON
```

# Version-Compatible Database

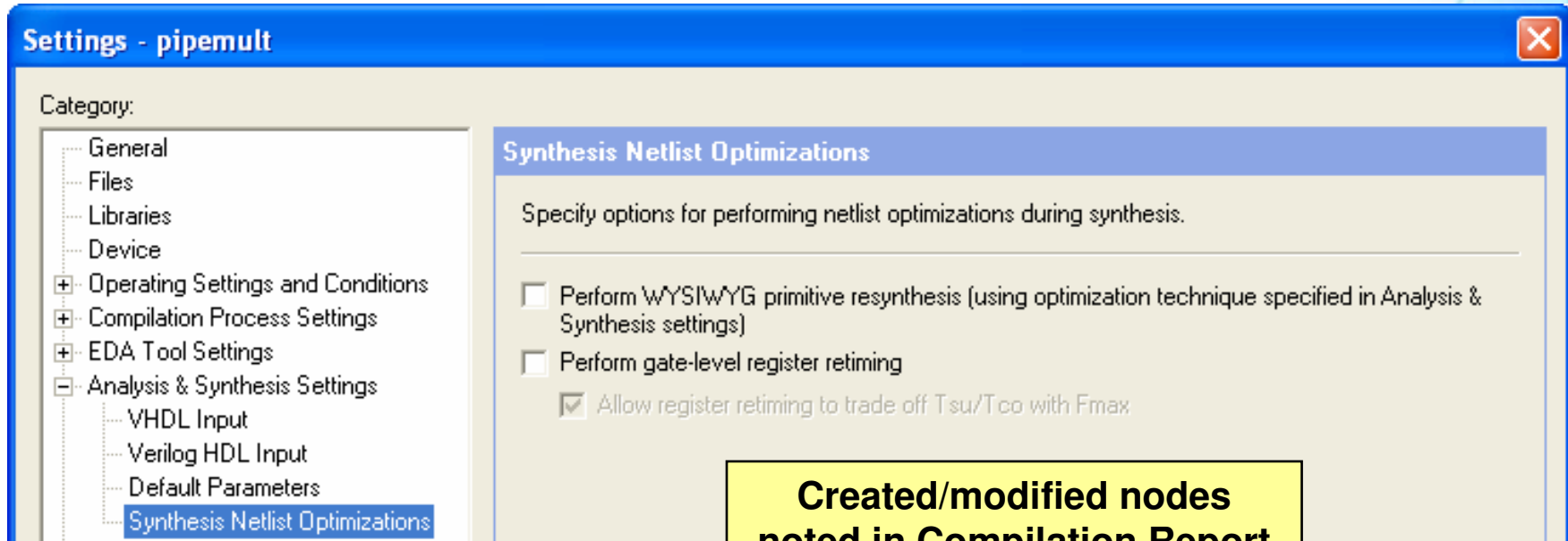
- Recommended if migrating design between versions of Quartus II software
- Exports a database from one version that can be imported directly into another version
- Use to preserve compilation results between Quartus II software versions
  - Re-running timing analysis or simulation with updated timing models
- Two methods to create
  - Settings dialog box
  - Project menu

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



# Synthesis Netlist Optimizations

- Further optimize netlists during synthesis
- Types
  - WYSIWYG primitive resynthesis
  - Gate-level register retiming



The screenshot shows a software window titled "Settings - pipemult". On the left is a tree view under "Category:" with the following items: General, Files, Libraries, Device, Operating Settings and Conditions (expanded), Compilation Process Settings (expanded), EDA Tool Settings (expanded), Analysis & Synthesis Settings (expanded), VHDL Input, Verilog HDL Input, Default Parameters, and Synthesis Netlist Optimizations (selected). The main area is titled "Synthesis Netlist Optimizations" and contains the text "Specify options for performing netlist optimizations during synthesis." Below this are three checkboxes: "Perform WYSIWYG primitive resynthesis (using optimization technique specified in Analysis & Synthesis settings)" (unchecked), "Perform gate-level register retiming" (unchecked), and "Allow register retiming to trade off T<sub>su</sub>/T<sub>co</sub> with F<sub>max</sub>" (checked).

**Created/modified nodes  
noted in Compilation Report**

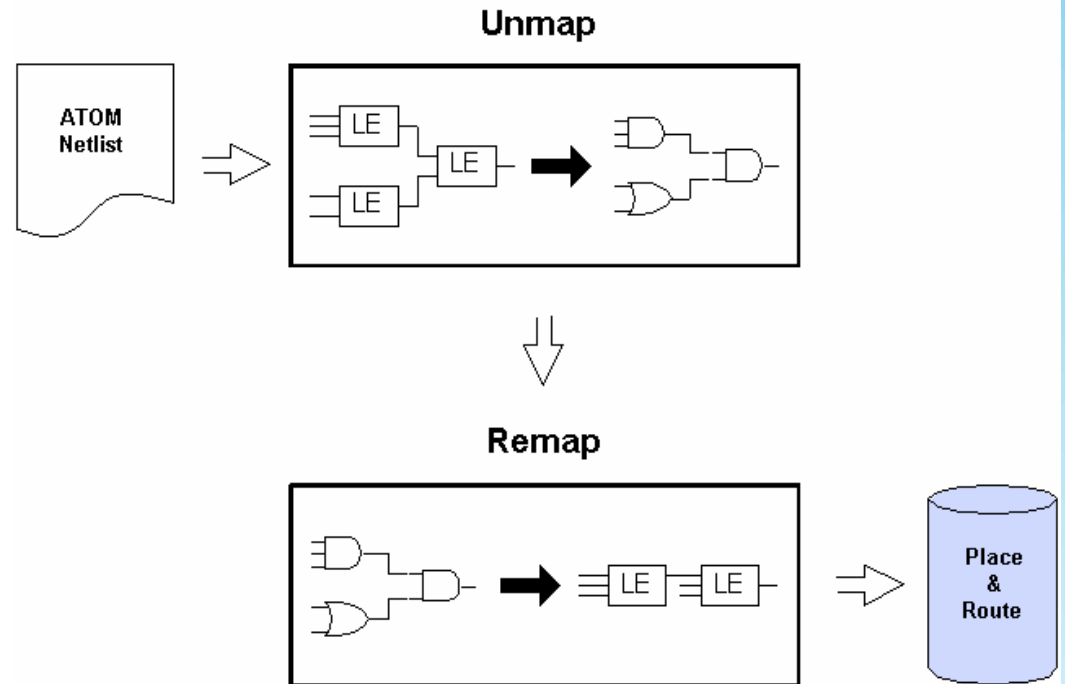
# WYSIWYG Primitive Resynthesis

- Unmaps 3<sup>rd</sup>-party atom netlist back to gates & then remaps to Altera primitives

- Unnecessary when using integrated synthesis

- Considerations

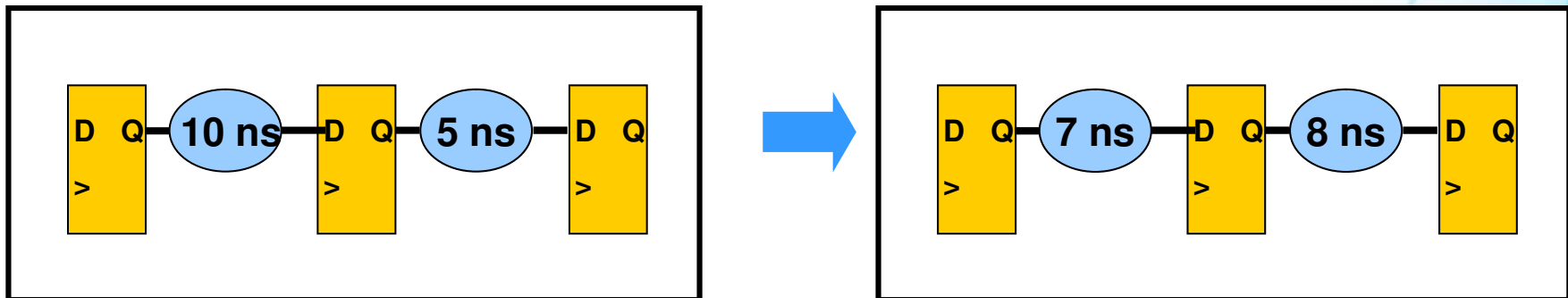
- Node names may change
- 3<sup>rd</sup>-party synthesis attributes may be lost
  - Preserve/keep
- Some registers may be synthesized away



```
Tcl: set_global_assignment -name ADV_NETLIST_OPT_SYNTN_WYSIWYG_REMAP ON
```

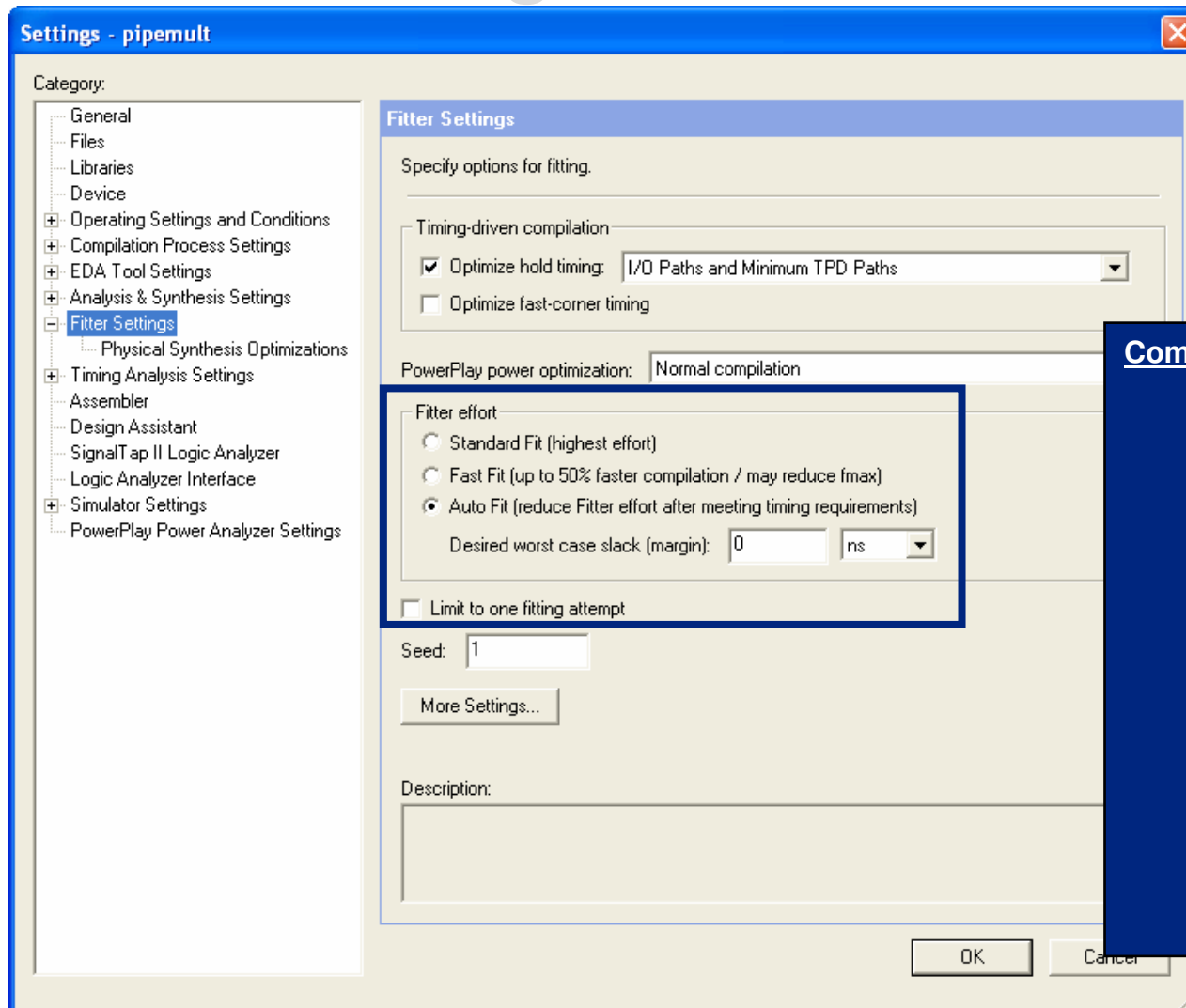
# Gate-Level Register Retiming

- Moves registers across combinatorial logic to balance timing
- Trades between critical & non-critical paths
- Makes changes at gate level



```
Tcl: set_global_assignment -name ADV_NETLIST_OPT_SYNTH_GATE_RETIME ON
```

# Fitter Settings



## Compilation speed/fitter effort

- **Standard fit**
  - Highest effort
  - Longest compile time
- **Fast fit**
  - Faster compile but possibly lesser design 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**

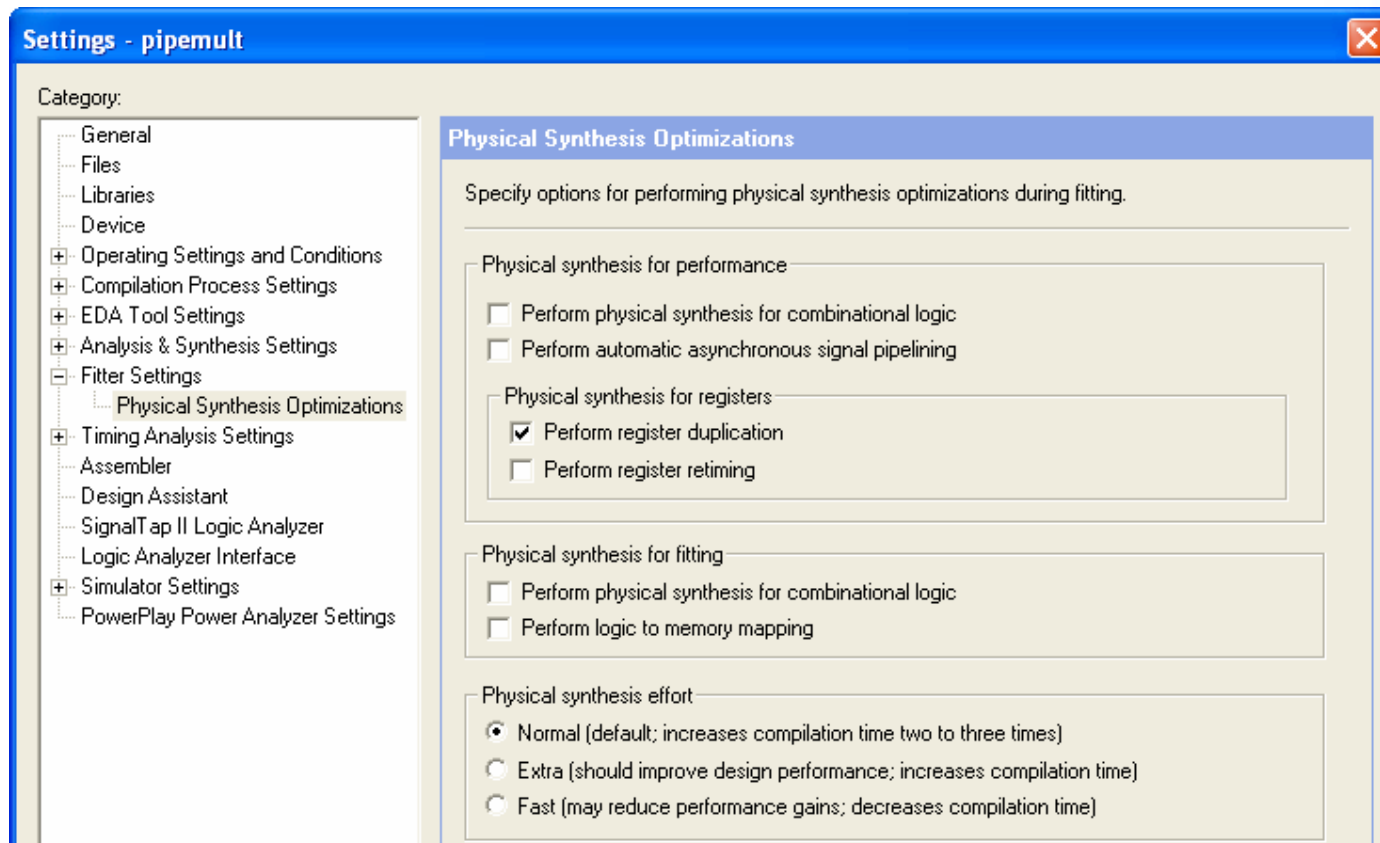
```
Tcl: set_global_assignment -name FITTER_EFFORT "<Effort Level>"
```

© 2007 Altera Corporation—**Confidential**

Altera, Stratix, Arria, Cyclone, MAX, HardCopy, Nios, Quartus, and MegaCore are trademarks of Altera Corporation

# Physical Synthesis

- Re-synthesis based on fitter output
  - Makes incremental changes that improve results for a given placement
  - Compensates for routing delays from fitter



# Physical Synthesis

## ■ Types

- Performance optimization
  - Combinational logic
  - Asynchronous signal pipelining
  - Register duplication
  - Register retiming
- Area optimization
  - Combinational logic
  - Logic to memory mapping

## ■ Effort

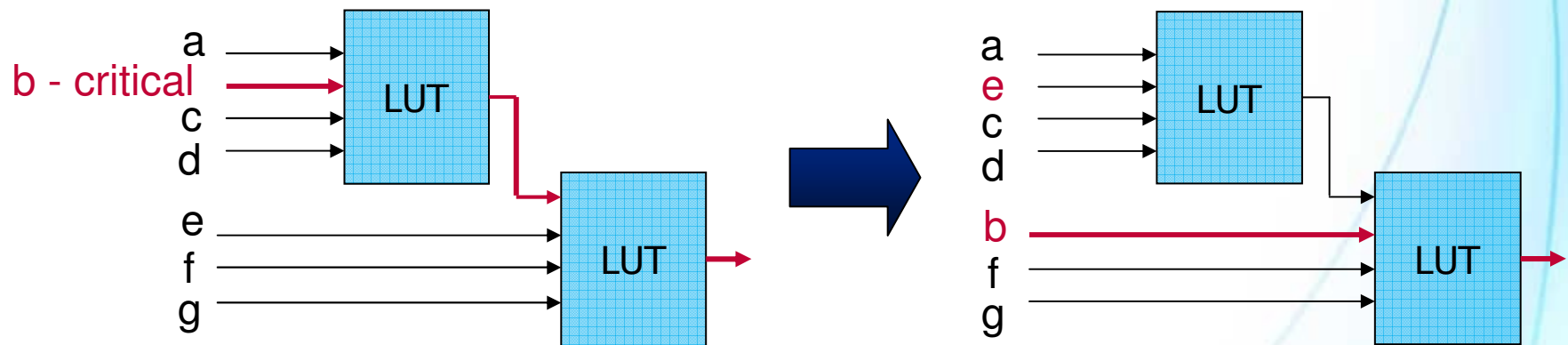
- Trades performance vs. compile time
- Normal, extra, or fast

## ■ New or modified nodes appear in Compilation Report

```
Tcl: set_global_assignment -name PHYSICAL_SYNTHESIS_EFFORT <Effort Level>
```

# Combinational Logic

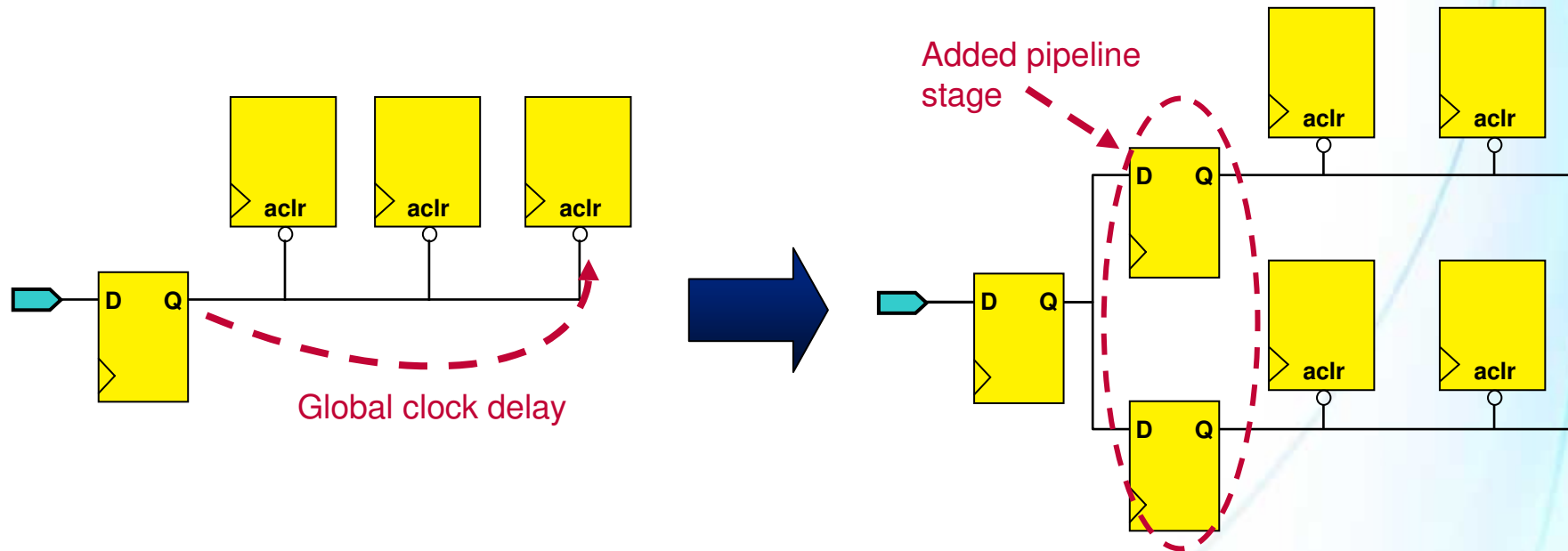
- Swaps look-up table (LUT) ports within LEs to reduce critical path LEs



```
Tcl: set_global_assignment -name PHYSICAL_SYNTHESIS_COMBO_LOGIC ON
```

# Asynchronous Signal Pipelining

- Adds pipeline registers to asynchronous clear or load signals in very fast clock domains

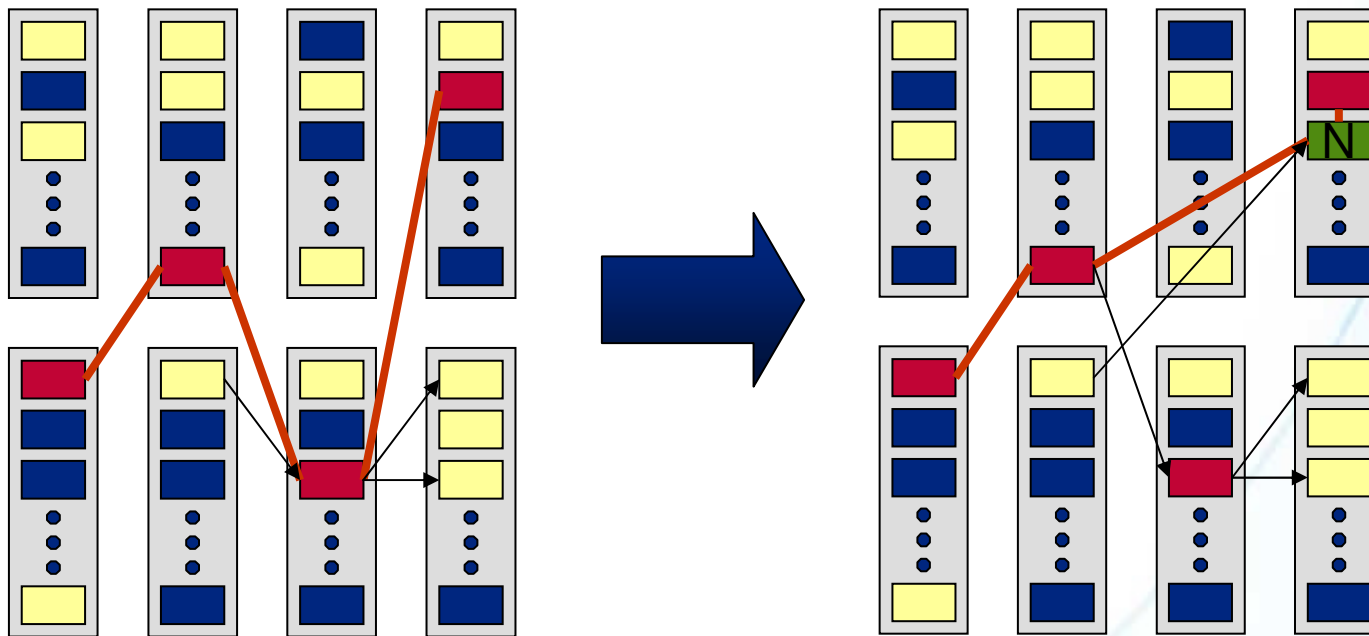


```
Tcl: set_global_assignment -name PHYSICAL_SYNTHESIS_ASYNC_SIGNAL_PIPELINING ON
```



# Duplication

- High fan-out registers or combinatorial logic duplicated & placed to reduce delay



```
Tcl: set_global_assignment -name PHYSICAL_SYNTHESIS_REGISTER_DUPLICATION ON
```

# Assignments (Logic Options)

- Individual switches applied to I/O, internal nodes or hierarchy blocks
- Use Assignment Editor to manage assignments
- Example assignments
  - Optimization Technique
  - PCI I/O
- Must perform at least analysis & elaboration to obtain hierarchy & node information

# Assignment Editor (AE)

- Provides spreadsheet assignment entry & display
  - Can copy & paste from clipboard

**Assignments Menu**

**Sort on columns**

**Assignment Editor toolbar**

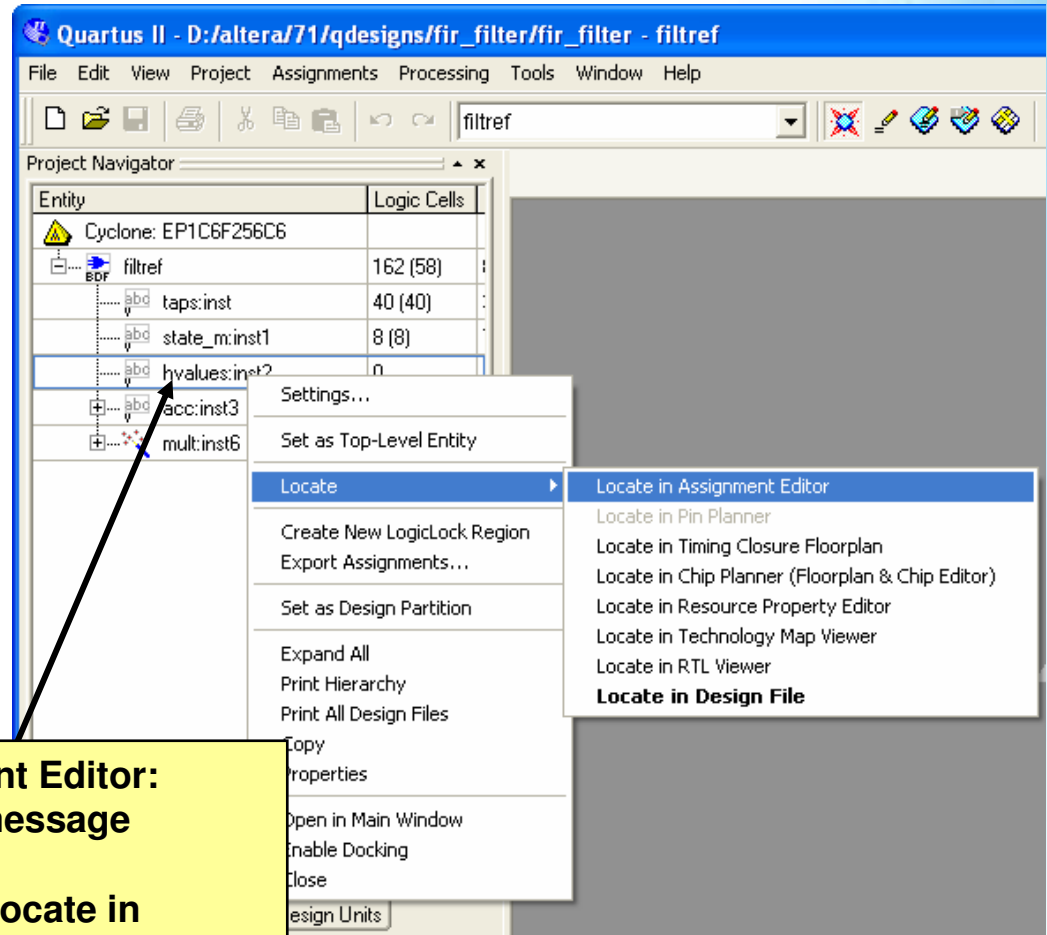
**Enable/disable individual assignments**

	From	To	Assignment Name	Value	Enabled
1		clk	Location	PIN_G1	Yes
2		clk	Clock Settings	clocka	Yes
3		clkx2	Clock Settings	clockb	Yes
4	clk	clkx2	Multicycle	2	Yes
5		d	Location	IOBANK_1	Yes
6		d	I/O Standard	SSTL-2 Class II	Yes
7		reset	I/O Standard	3.3-V LVTTTL	Yes
8		newt	I/O Standard	3.3-V LVCMOS	Yes
9		d[6]	I/O Standard	2.5 V	Yes
10		reset	Location	IOBANK_4	Yes
11		yn_out	Location	IOBANK_4	Yes
12		me	Reserve Pin	As input tri-stated	Yes
13		yvalid	Location	PIN_E14	Yes
14		clkx2	Location	PIN_C13	Yes
15		newt	Location	PIN_C15	Yes
16	<<new>>	<<new>>	<<new>>	<<new>>	<<new>>

**Customizable columns**

# Cross-Probing to Assignment Editor

- Virtually all windows & tools cross-probe (locate) to Assignment Editor
- Examples
  - Project Navigator
  - Message window
  - Compilation Report
  - Design files

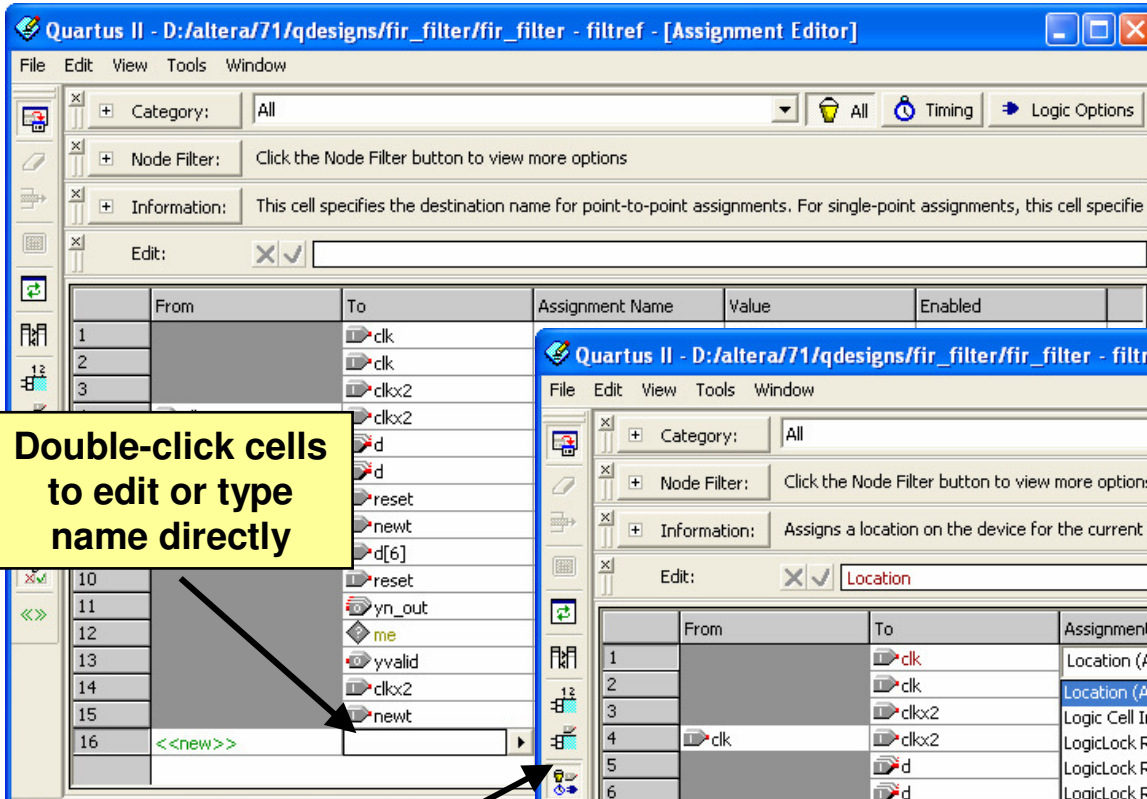


**\*Note:** Assignment Editor pre-filled with target node/pin name

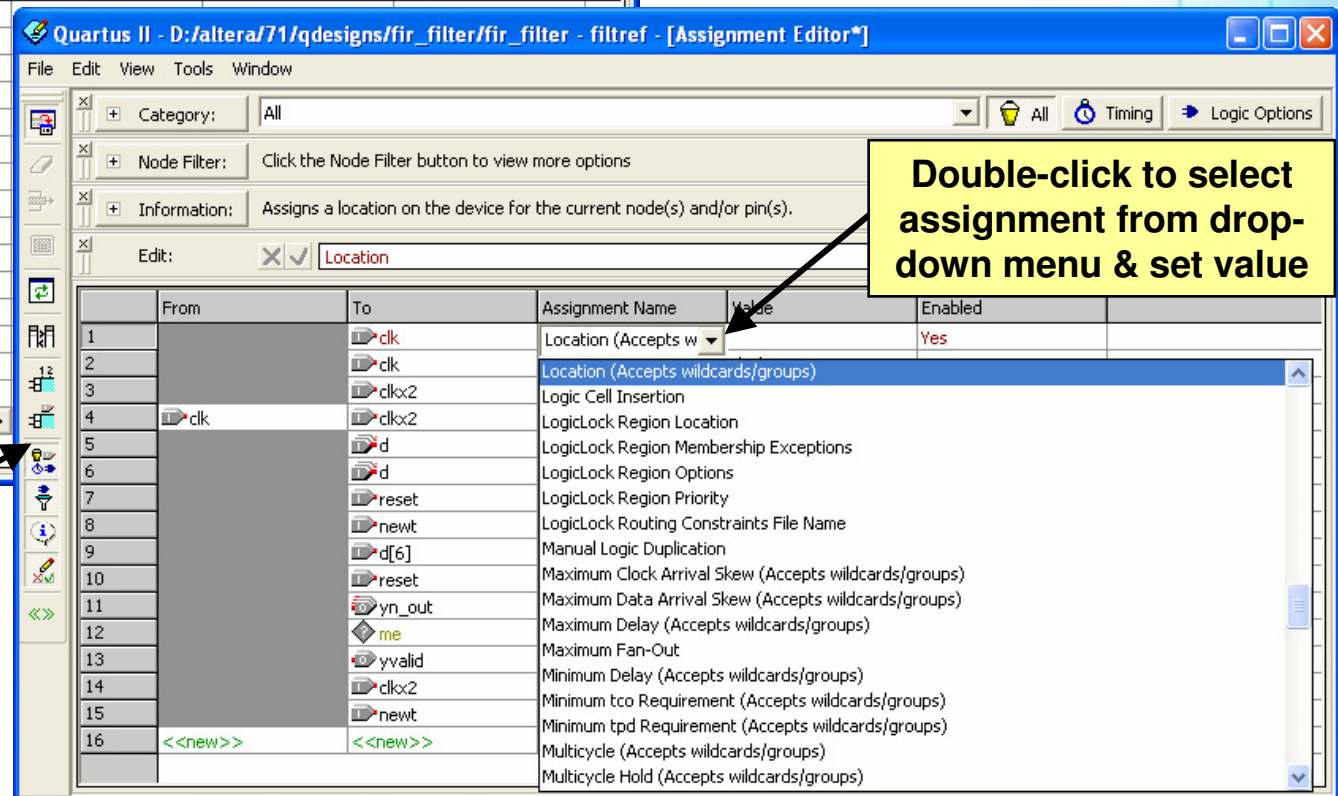
© 2007 Altera Corporation—**Confidential**

Altera, Stratix, Arria, Cyclone, MAX, HardCopy, Nios, Quartus, and MegaCore are trademarks of Altera Corporation

# Using Assignment Editor



**Double-click cells to edit or type name directly**



**Double-click to select assignment from drop-down menu & set value**

**Launch node finder or select from assignment groups**

# Editing Multiple Assignments

- Use Edit bar, auto-fill, copy & paste

	To	Location	I/O Bank	I/O Standard	General Function	Special Function	Re
1	clk			3.3-V LVTTTL			
2	d	IOBANK_1	1	SSTL-2 Class II			
3	reset	IOBANK_4	4	3.3-V LVTTTL			
4	yn_out	IOBANK_4	4	3.3-V LVTTTL			
5	yvalid	PIN_E14	3	3.3-V LVTTTL	Row I/O	LVDS38p/DQ1R3	
6	clkx2	PIN_C13	2	3.3-V LVTTTL	Column I/O	LVDS33p	
7	newt	PIN_C15	3	3.3-V LVCMOS	Row I/O	LVDS36p	
8	<<new>>	<<new>>					

# Node Finder



Search by name using wildcards (? or \*)

Use filter to select the type of nodes to be displayed

Displays nodes meeting search criteria

Locate nodes in a certain level of hierarchy

List of found nodes in selected entity & lower levels of hierarchy

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

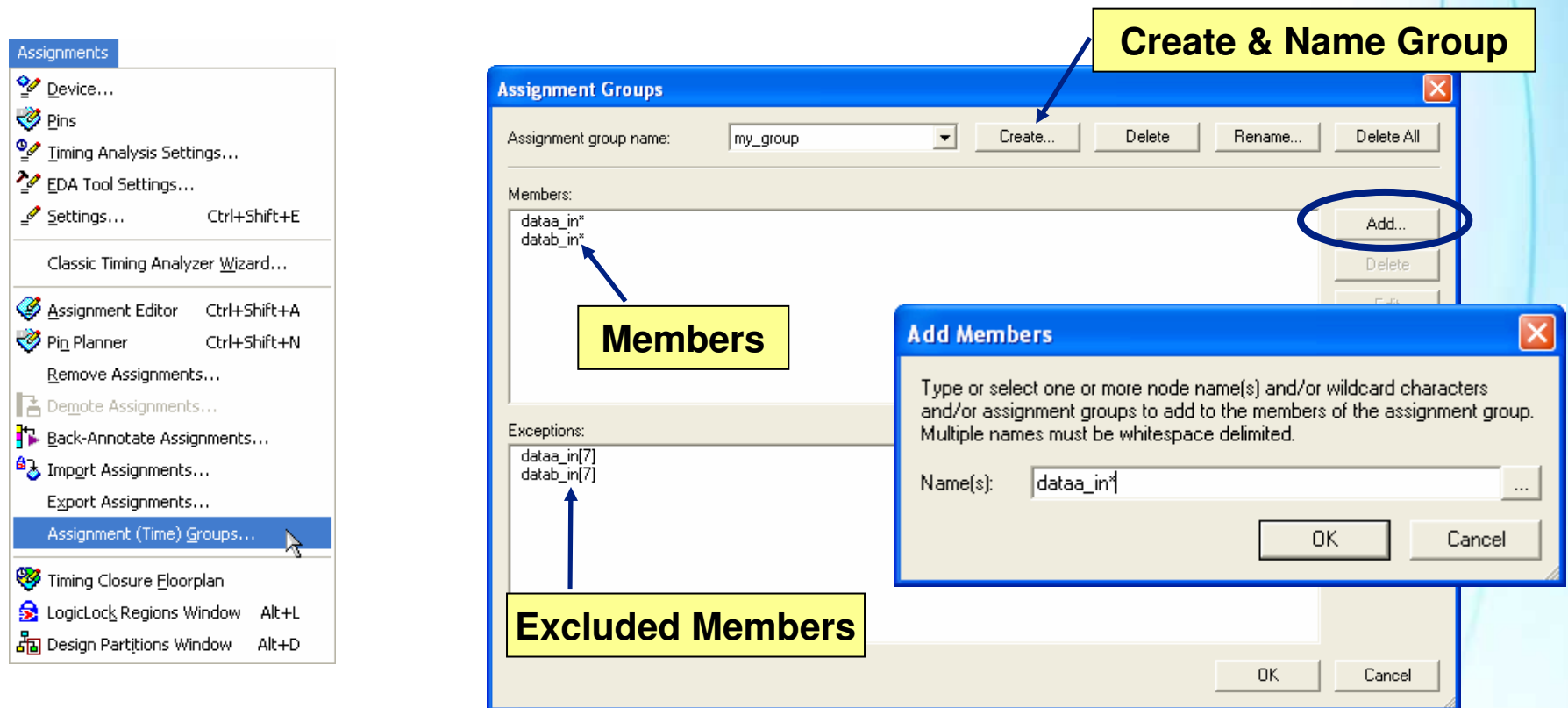
The screenshot shows the Node Finder dialog box with the following elements:

- Named:** \* (with a dropdown arrow)
- Filter:** Pins: all (with a dropdown arrow)
- Look in:** |filtref
- Nodes Found:** A table with columns for Name and Assignm.
- Filter dropdown menu:** Pinned with a checkmark, showing options: Pins: all, Pins: all & Registers: post-fitting, Registers: pre-synthesis, Registers: post-fitting, Design Entry (all names), Post-synthesis, Post-Compilation, SignalTap II: pre-synthesis, SignalTap II: post-fitting, SignalProbe.
- Buttons:** Customize..., List, Stop, OK, Cancel.
- Checkboxes:** Include subentities (checked).
- Table:**

Name	Assignm
clk	PIN_G1
clkx2	PIN_H1
d	Unassign
d[0]	PIN_C7
d[1]	PIN_B7
d[2]	PIN_G2
d[3]	PIN_H5
d[4]	PIN_A8
d[5]	PIN_D7
d[6]	PIN_N7
d[7]	PIN_D8
d[8]	PIN_A6
d[9]	PIN_B8
d[10]	PIN_G16
d[11]	Unassigned
d[12]	PIN_B6
d[13]	PIN_E1
d[14]	PIN_D5
d[15]	PIN_F7
- Navigation arrows:** A set of four arrows (>, >>, <, <<) is circled in the table area.

# Assignment (Time) Groups

- Assigns named to user-defined group of nodes
- Allows single assignment to constrain entire group








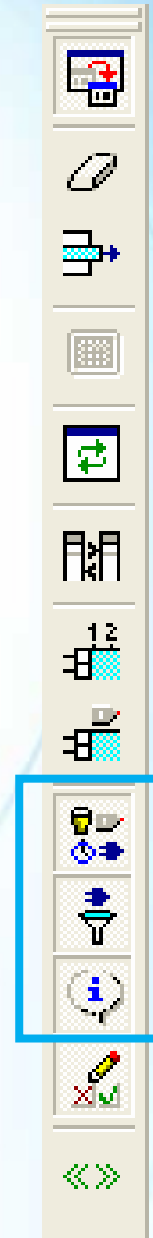
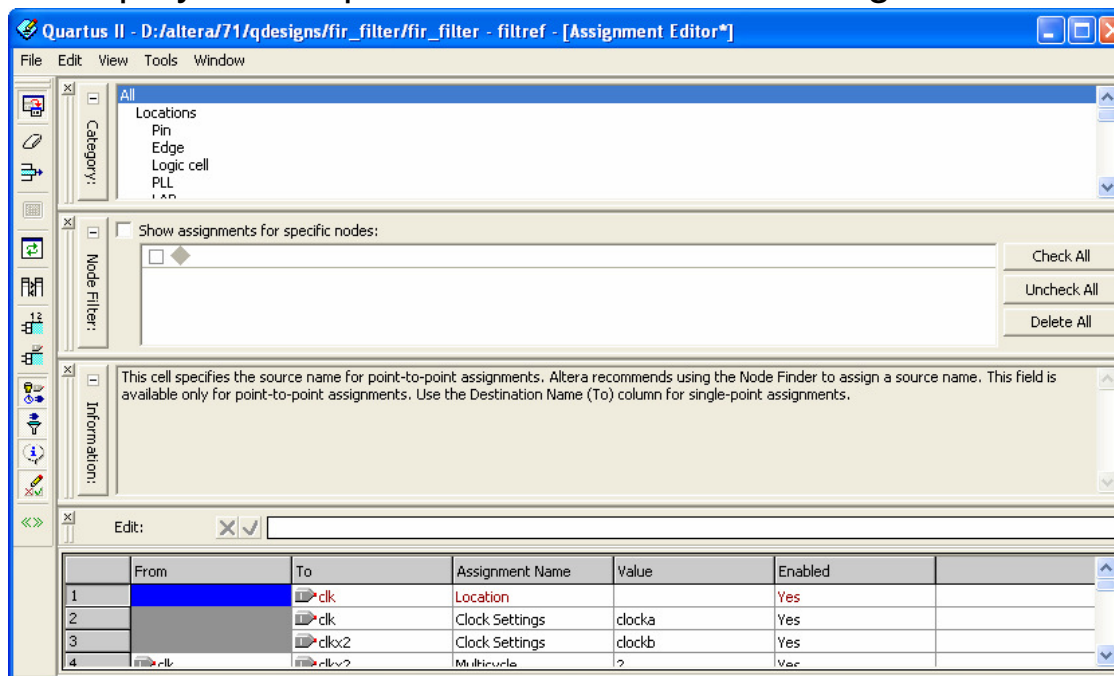
# AE Dynamic Checking

- Validity of constraint checked during entry
- Color-coded to display status
  - Grey – disabled
  - Black – applied
  - Yellow – assignment warning
  - Dark red – incomplete
  - Bright red – error/illegal value
  - Green – enter new assignment

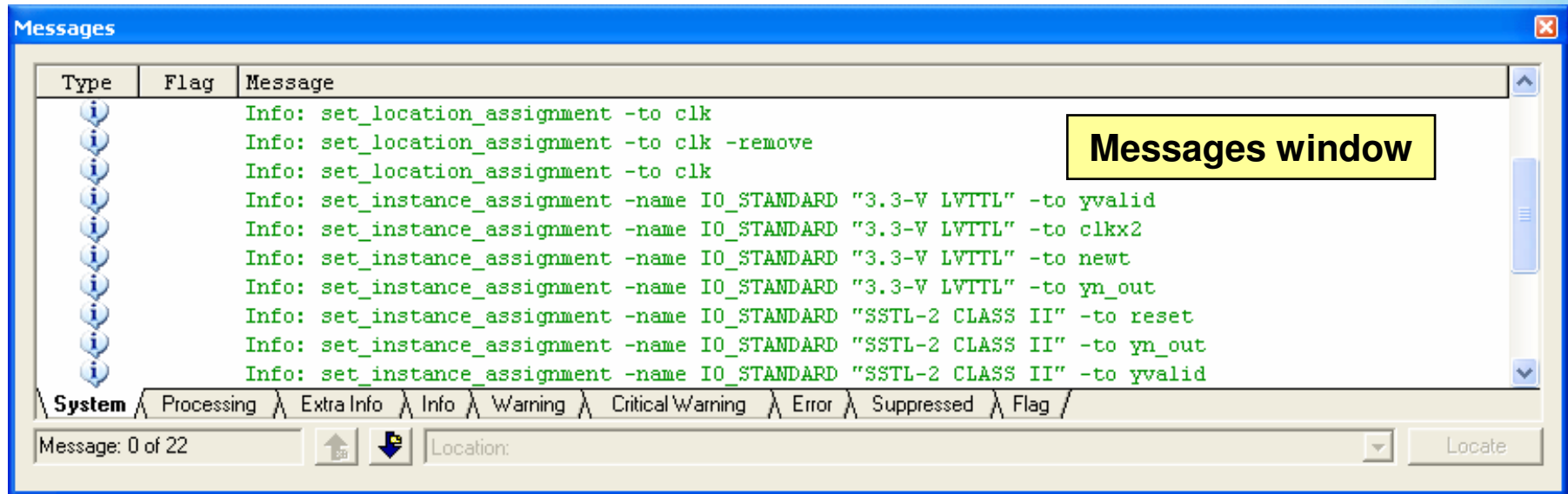
	From	To	Assignment Name	Value	Enabled
1		yn_out	Location	IOBANK_2	Yes
2		yvalid	Location	PIN_75	Yes
3		d	Location	IOBANK_1	No
4		clk	Clock Settings	clk	Yes
5		unknown_clock	Clock Settings	clk2	Yes
6		clkx2	Clock Settings	clk2	Yes
7	clk	clkx2	Multicycle	2	Yes
8		d	DQS Frequency	1MHz	Yes
9		yvalid		Minimum Current	Yes
10		d	I/O Standard	LVCMS0	Yes
11		yn_out	I/O Standard	LVCMS0	Yes
12		yvalid	I/O Standard	LVCMS0	Yes
13	<<new>>	<<new>>	<<new>>		

# Assignment Editor Features

- Category bar 
  - Filters displayed constraints based on category
    - Ex. Pin assignments, timing assignments
- Node Filter bar 
  - Filters displayed constraints based on node name
- Information bar 
  - Displays description of selected cell or assignment



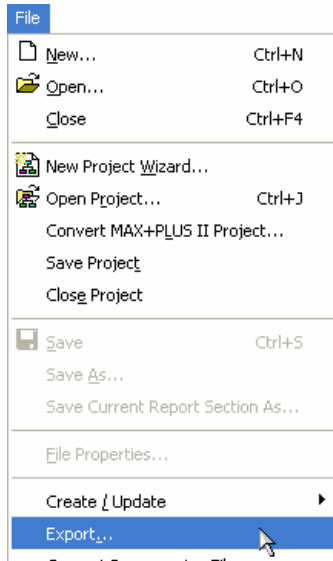
# AE Tcl Commands



- 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 CSV File Assignments (Excel)

- Export to CSV file (File menu)
  - Import data into Excel




From	To	Assignment Name	Value
	~DATA0~	Location	PIN_L8
	d[7]	Location	PIN_J4
	d[6]	Location	PIN_H4
	d[5]	Location	PIN_E6
	d[4]	Location	PIN_F1
	d[3]	Location	PIN_H3
	d[2]	Location	PIN_J6
	d[1]	Location	PIN_G4
	d[0]	Location	PIN_F2
	yn[7]	Location	PIN_H2
	yn[6]	Location	PIN_L6
	yn[5]	Location	PIN_G2
	yn[4]	Location	PIN_J2
	yn[3]	Location	PIN_G1
	yn[2]	Location	PIN_J3
	yn[1]	Location	PIN_H1
	yn[0]	Location	PIN_K2
	clk	Location	PIN_L2
	reset	Location	PIN_L3
	newt	Location	PIN_K6
	yvalid	Location	PIN_E5
	nxt	Location	PIN_A6
	yn	Output Maximum Delay	4ns
clk	input data	Input Maximum Delay	7ns
mult_mega:u4]*	acc:u5]*	Cut Timing Path	On
	clk	Clock Settings	clk

# Example Assignments

- Optimization Technique
- PCI I/O

# Optimization Technique

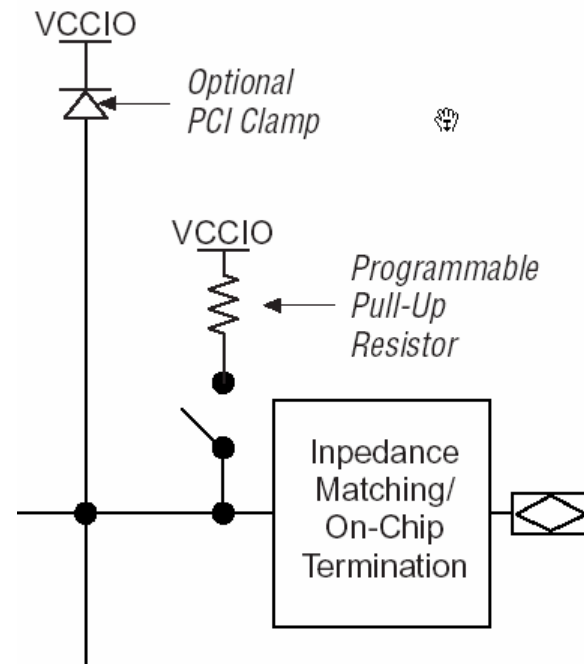
- Selects synthesis optimization goal
  - Speed
  - Balanced (default)
  - Area
- Applies only to hierarchical entities
  - Locate from Project Navigator
  - Drag and drop into Assignment Editor
- Effects synthesis & logic mapping
- Only applies to Quartus II integrated synthesis

	From	To	Assignment Name	Value	Enabled	
1		 acc:b2v_inst3	Optimization Technique -- Stratix II	Speed	Yes	
2	<<new>>	<<new>>	<<new>>			

```
Tcl: set_instance_assignment -name STRATIXII_OPTIMIZATION_TECHNIQUE SPEED -to <node name>
```

# PCI I/O

- 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



	From	To	Assignment Name	Value	Enabled	
1		yn_out[0]	PCI I/O	On	Yes	
2		yn_out[1]	PCI I/O	On	Yes	
3		yn_out[2]	PCI I/O	On	Yes	
4		yn_out[3]	PCI I/O	On	Yes	
5	<<new>>	<<new>>	<<new>>			

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

# Available Logic Options (Assignments)

The screenshot shows the Quartus II Help interface. The search bar contains "logic options, list of". The search results list various logic options, with "list of" selected. The main content area displays the "Logic options" page, which includes a table listing the logic options and their device family support.

**Logic options**

The following table lists the logic options that are available for all Altera devices.

Logic Option	Device Family Support
<a href="#">Add D and Q Ports of Register Node to Simulation Output Waveforms</a>	All Altera devices
<a href="#">Add Pass-Through Logic to Inferred RAMs</a>	All Altera devices
<a href="#">Add To Simulation Output Waveforms</a>	All Altera devices
<a href="#">Alias</a>	All Altera devices
<a href="#">Allow Any RAM Size For Recognition</a>	supported device families
<a href="#">Allow Any ROM Size For Recognition</a>	supported device families
<a href="#">Allow Any Shift Register Size For Recognition</a>	supported device families
<a href="#">Allow RAM blocks to use LAB location for memory</a>	Stratix III devices
<a href="#">Allow register retiming to trade off Tsu/Tco with Fmax</a>	supported device families
<a href="#">Allow Synchronous Control Signals</a>	supported device families
<a href="#">Allow XOR Gate Usage</a>	supported device families
<a href="#">Always Enable Input Buffers</a>	supported device families
<a href="#">Asynchronous Signal Pipelining Register Reach</a>	All Altera devices
<a href="#">Auto Carry Chains</a>	All Altera devices except MAX 3000, MAX 7000AE, MAX 7000B, and MAX 7000S families
<a href="#">...</a>	supported device families

- 1) Go to Quartus II Help (Index)
- 2) Type in "logic options"
- 3) Double-click "list of"

Supported devices shown for each assignment



# Updating QSF File

- QSF not updated automatically when constraint entered or Assignment Editor is saved
- QSF updated when
  - Project is saved (File menu)
  - Beginning of compilation
- Change behavior to updating assignments immediately (Tools menu ⇒ Options ⇒ General ⇒ Processing)
  - Will impact software performance due to file accesses

# Design Assistance

- Quartus II tools available to help improve designs
- Design Assistant
- Optimization Advisors
  - Resource
  - Timing
  - Power
  - Incremental Compilation
  - Compilation Time

# Design Assistant

**Settings - filtref**

Category:

- General
- Files
- Libraries
- Device
- Operating Settings and Conditions
- Compilation Process Settings
- EDA Tool Settings
- Analysis & Synthesis Settings
- Filter Settings
- Timing Analysis Settings
- Assembler
- Design Assistant**
- Signal ap II Logic Analyzer
- Logic Analyzer Interface
- Simulator Settings
- PowerPlay Power Analyzer Settings

**Design Assistant**

Specify the potential design problems that you want the Design Assistant to check for or a category of design problems.

Run Design Assistant during compilation

Select the rules you want the Design Assistant to apply to the project:

Design Assistant configuration rule names

- Clock
  - Rule C101: Gated clock should be implemented according to the Altera standard scheme
  - Rule C102: Logic cell should not be used to generate inverted clock
  - Rule C103: Gated clock is not feeding at least a pre-defined number of clock port to effectively save power : 30
  - Rule C104: Clock signal source should drive only input clock ports
  - Rule C105: Clock signal should be a global signal : 25
  - Rule C106: Clock signal source should not drive registers that are triggered by different clock edges
- Reset
- Timing closure
- Non-synchronous design structure
- Signal race
  - Rule S101: Output enable and input of the same tri-state node should not be driven by same signal source
  - Rule S102: Synchronous port and asynchronous port of the same register should not be driven by the same signal source
  - Rule S103: More than one asynchronous signal port of the same register should not be driven by the same signal source
  - Rule S104: Clock port and any other signal port of same register should not be driven by the same signal source
- Asynchronous clock domains
- HardCopy

Advanced...

Report Settings...

OK Cancel

## ■ Checks for potential design issues

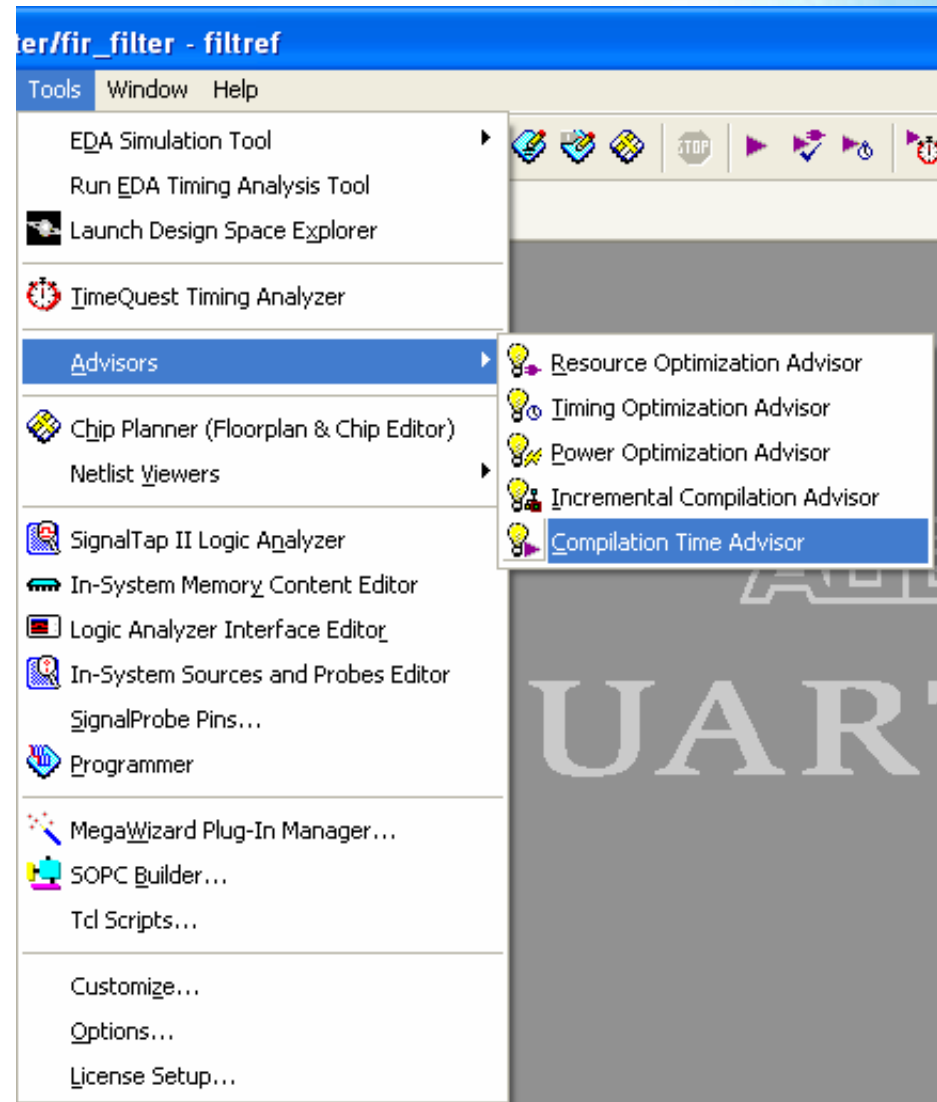
- Clocks
- Reset
- Non-synchronous design structure
- Timing closure
- Asynchronous clock domain data transfers
- Signal race conditions
- HardCopy
- FSM

© 2007 Altera Corporation—**Confidential**

Altera, Stratix, Arria, Cyclone, MAX, HardCopy, Nios, Quartus, and MegaCore are trademarks of Altera Corporation

# Optimization Advisors

- Provide design-specific recommendations (feedback) on optimizing designs
- Five types
  - Resource usage optimization
  - Timing (performance) optimization
  - Power optimization
  - Incremental compilation suggestions
  - Compilation time reduction



# Example Optimization Advisor

The screenshot shows the Quartus II Timing Optimization Advisor interface. On the left, a tree view lists various optimization recommendations, grouped into three stages. A yellow callout box labeled 'Problem area identified' points to the 'Maximum Frequency (fmax)' recommendation. Another yellow callout box labeled 'Three stages of recommendations to try in order' points to the 'Stage 1', 'Stage 2', and 'Stage 3' groupings. A third yellow callout box labeled 'Green checkmark indicates settings already in use' points to the 'Specify Fitter effort' recommendation, which has a green checkmark next to it. On the right, a detailed view of the 'Specify Fitter effort' recommendation is shown, including a description, summary, and action. A yellow callout box labeled 'Description of recommended settings' points to the description text. Another yellow callout box labeled 'Links to adjust settings' points to a blue hyperlink 'Open Settings dialog box - Fitter Settings page' at the bottom of the action section.

**Problem area identified**

**Three stages of recommendations to try in order**

**Green checkmark indicates settings already in use**

**Description of recommended settings**

**Links to adjust settings**

**Specify Fitter effort**

Recommendation	Use Auto Fit or Standard Fit to meet timing requirements.
Description	You can adjust how much effort the fitter uses to get a fit on your design. Standard fit will try to maximize the design's timing performance. Auto Fit will try to meet a design's timing requirements without spending extra effort to exceed those requirements. The fast fit feature reduces the Fitter effort so that compilation time is shorter, but timing performance may be reduced.
Summary	Recommended changes have unknown effect on logic element usage, compilation time, and maximum frequency (fmax) for the design.
Action	Set the Fitter Effort to Auto Fit if it is currently set to Fast Standard Fit if it is currently set to Auto Fit in the Fitter Settings dialog box (Assignments menu).  No action is needed for this recommendation. The recommendation is already made.  Current Global Settings: Fitter Effort = STANDARD FIT (Recommended: STANDARD FIT or Auto Fit)  <a href="#">Open Settings dialog box - Fitter Settings page</a>

# *Exercise 4 Demonstration*

# Settings & Assignments Summary

- Settings & assignments allow a designer to control how a design is synthesized, placed, & routed
- Use the Settings dialog box to adjust project-wide settings
- Use the Assignment Editor to enable/disable individual assignments targeting hierarchy blocks, internal nodes, or I/O
- The Quartus II software provides features such as the Design Assistant & Optimization Advisors to help improve design results

# Design Analysis Support Resources

- Quartus II Handbook chapters (all Volume 2)
  - “Area & Timing Optimization”
  - “Power Optimization”
  - “Assignment Editor”
  - “Netlist Optimizations & Physical Synthesis”
- Training Courses and Demonstrations
  - Optimization Advisor demonstration



**ALTERA**

# Quartus II Software Design Series: Foundation

*I/O Planning*



# I/O Planning Need

- I/O standards increasing in complexity
- FPGA/CPLD I/O structure increasing in complexity
  - Results in increased pin placement guidelines
- PCB development performed simultaneously with FPGA design
  - Sometimes before!
- Pin assignments need to be verified earlier in design cycle
- Designers need easy way to transfer pin assignments into board tools

# I/O Planning Agenda

- Assigning Device I/O Locations
- I/O Assignment Analysis
- Early I/O Planning Methodology
- PCB Tool Support

# Assigning Device I/O Locations

- Pin Planner
- Import from spreadsheet in CSV format
- Type directly into QSF file
- Scripting
- Using synthesis attributes in HDL

*Note: Other methods/tools are available in Quartus II 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
- Three main sections
  - Package view
  - All Pins list
  - Groups list

**Assignments Menu ⇒  
Pin Planner**

# Pin Planner Window

Top View - Flip Chip  
Stratix II - EP2S15F484C3

**Groups list**

**Toolbar**

**Package view (Top or Bottom)**

**All Pins list**

	Node Name	Direction	Location	I/O Bank	Vref Group	I/O Standard
1	clk	Input				3.3-V LVTTTL (default)
2	clkx2	Input				3.3-V LVTTTL (default)
3	d[7]	Input				3.3-V LVTTTL (default)
4	d[6]	Input				3.3-V LVTTTL (default)
5	d[5]	Input				3.3-V LVTTTL (default)
6	d[4]	Input				3.3-V LVTTTL (default)
7	d[3]	Input				3.3-V LVTTTL (default)
8	d[2]	Input				3.3-V LVTTTL (default)

# Pin Planner Window (cont.)

## ■ Package view

- Displays graphical representation of chip package
- Use to make or edit design I/O assignments
- Use to locate other package pins (i.e. power & configuration pins)

## ■ All pins list

- Displays pins as indicated by filter
  - Buses are auto-expanded
- Use to edit pin settings/properties directly

## ■ Groups list

- Similar to All Pins list except displays only groups & buses
- Use to make bus and group assignments
- Use to create new user-defined groups

# Assigning Pins Using Pin Planner

**Drag & drop single pin; tooltips provide pin information**

<none> @ PIN\_M2 (Dedicated Clock, CLK11p, Input)  
The pin is assignable

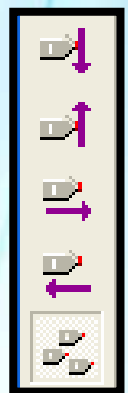
Named:	Node Name	Direction	Location
1	clk	Input	
2	clkx2	Input	
3	d[7]	Input	
4	d[6]	Input	
5	d[5]	Input	
6	d[4]	Input	
7	d[3]	Input	
8	d[2]	Input	

**Drag & drop multiple highlighted pins or buses**

<none> @ PIN\_R19 (Row I/O, DIFFIO\_TX5p)  
The pin is assignable

Named:	Node Name	Direction	Location
1	clk	Input	
2	clkx2	Input	
3	d[7]	Input	
4	d[6]	Input	
5	d[5]	Input	
6	d[4]	Input	
7	d[3]	Input	
8	d[2]	Input	

**Choose one-by-one or pin alignment direction (Pin Planner toolbar or Edit menu)**





# Assigning Pins Using Pin Planner (2)

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

**Double-click pin or I/O bank to open Properties dialog box**

**Filter nodes displayed**

**Pin Properties**

Pin number: PIN\_B16  
 Node name: d[4]  
 I/O standard: 3.3-V LVTTTL (default)  
 Reserved:  
 Properties:

Name	Value
I/O Bank	3
General Function	Column I/O
Special Function	DQSn11T
Pad ID	345
VREF Pad ID	N/A

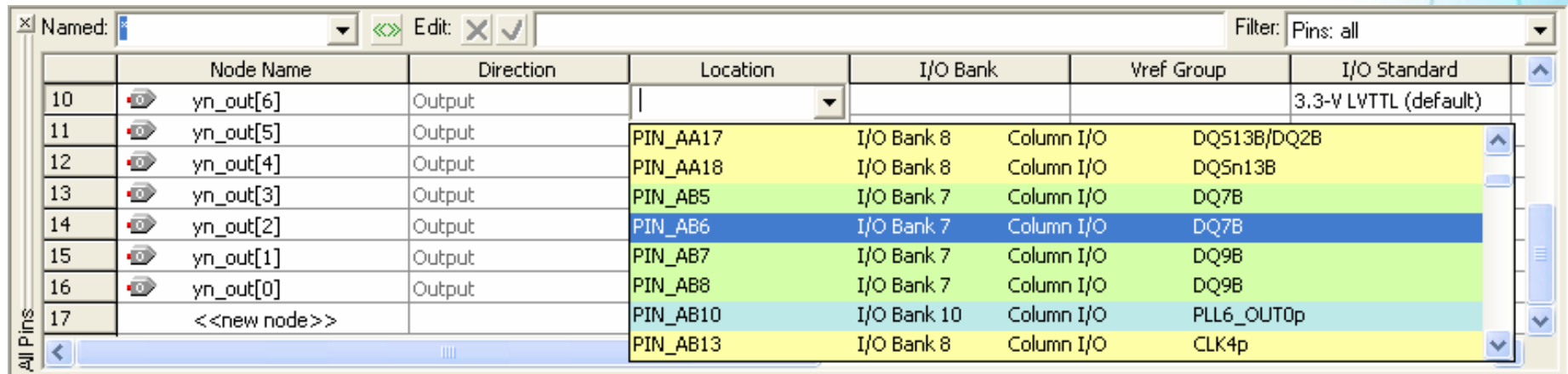
IOBANK\_3 (8 assigned/50 total)  
 (8 input/0 output/0 bidir)  
 d[4] <Input> @ PIN\_B16  
 d[5] <Input> @ PIN\_B17  
 d[6] <Input> @ PIN\_B18  
 d[7] <Input> @ PIN\_B19  
 d[3] <Input> @ IOBANK\_3  
 d[2] <Input> @ IOBANK\_3  
 d[1] <Input> @ IOBANK\_3  
 d[0] <Input> @ IOBANK\_3

Node Name	Direction	Location	I/O Bank
d[7]	Input	PIN_B19	3
d[6]	Input	PIN_B18	3
d[5]	Input	PIN_B17	3
d[4]	Input	PIN_B16	3
d[3]	Input	IOBANK_3	3
d[2]	Input	IOBANK_3	3
d[1]	Input	IOBANK_3	3
d[0]	Input	IOBANK_3	3

Filter: Pins: all  
 Pins: assigned  
 Pins: unassigned  
 Pins: input  
 Pins: output  
 Pins: bidirectional  
 Pins: all  
 customize>>

## Assigning Pins Using Pin Planner (3)

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



The screenshot shows the Pin Planner interface with a table of nodes and their assigned pins. The table has columns for Node Name, Direction, Location, I/O Bank, Vref Group, and I/O Standard. The nodes are numbered 10 through 17. Node 10 is a new node. Nodes 11 through 16 are output nodes assigned to various pins. Node 17 is a new node.

Node Name	Direction	Location	I/O Bank	Vref Group	I/O Standard
yn_out[6]	Output				3.3-V LVTTTL (default)
yn_out[5]	Output	PIN_AA17	I/O Bank 8	Column I/O	DQ513B/DQ2B
yn_out[4]	Output	PIN_AA18	I/O Bank 8	Column I/O	DQ5n13B
yn_out[3]	Output	PIN_AB5	I/O Bank 7	Column I/O	DQ7B
yn_out[2]	Output	PIN_AB6	I/O Bank 7	Column I/O	DQ7B
yn_out[1]	Output	PIN_AB7	I/O Bank 7	Column I/O	DQ9B
yn_out[0]	Output	PIN_AB8	I/O Bank 7	Column I/O	DQ9B
<<new node>>		PIN_AB10	I/O Bank 10	Column I/O	PLL6_OUT0p
		PIN_AB13	I/O Bank 8	Column I/O	CLK4p

# Reserving I/O Pins

- Type reserved I/O name directly into Pins List & select reserve configuration

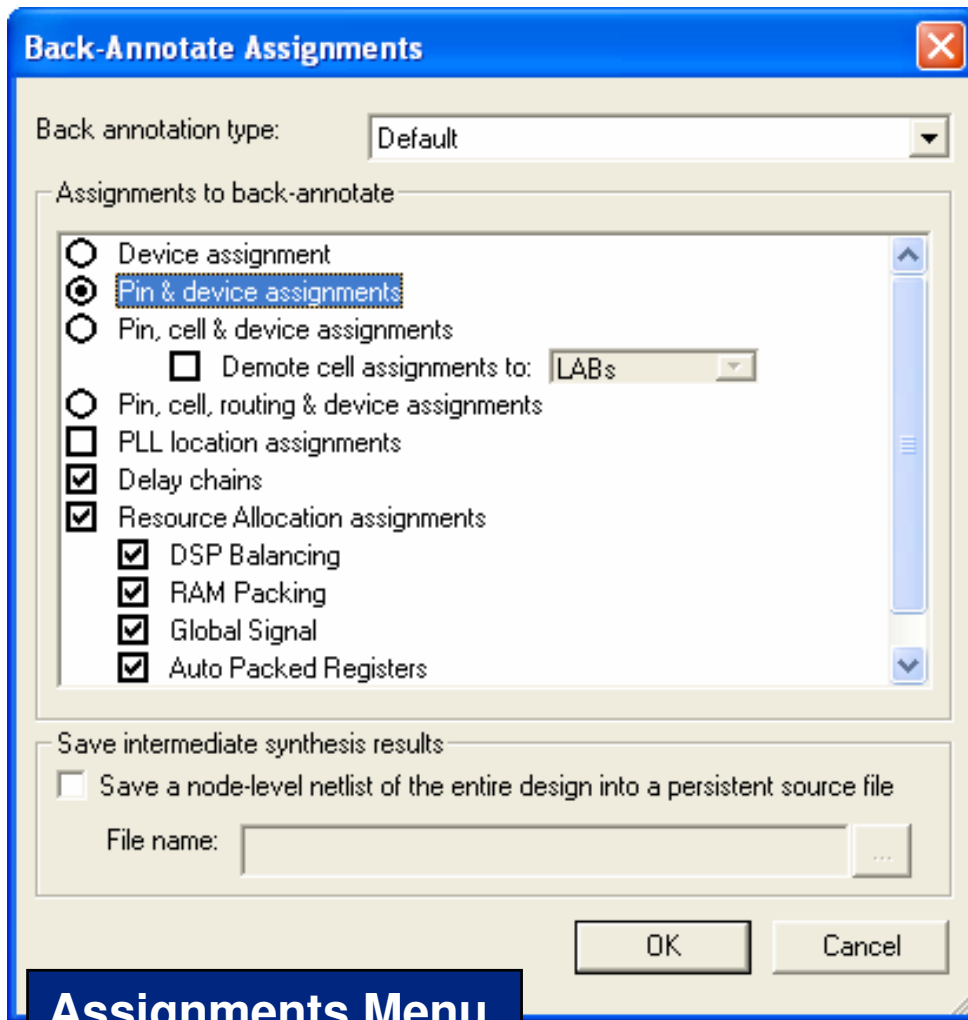
Node Name	Direction	Location	I/O Bank	Vref Group	I/O Standard	Reserved
yn_out[4]	Output				3.3-V LVTTTL (default)	
yn_out[3]	Output				3.3-V LVTTTL (default)	
yn_out[2]	Output				3.3-V LVTTTL (default)	
yn_out[1]	Output				3.3-V LVTTTL (default)	
yn_out[0]	Output				3.3-V LVTTTL (default)	
wvalid	Output				3.3-V LVTTTL (default)	
my_reserved_pin	Input				3.3-V LVTTTL (default)	
<<new node>>						

Context Menu Options:

- As bidirectional
- As input tri-stated
- As output driving an unspecified signal
- As output driving ground
- As output driving VCC

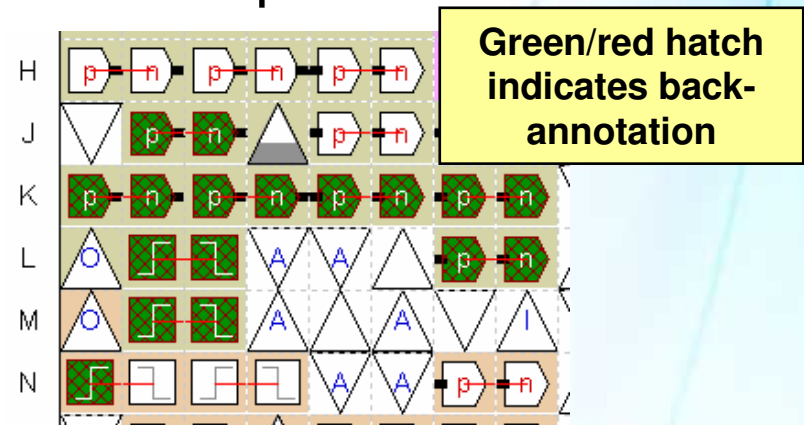
- Or right-click on pin in Package View and choose Reserve ⇒ As...
  - Pin name set to *user\_reserve\_<pin\_number>*

# Back-Annotation



## Assignments Menu

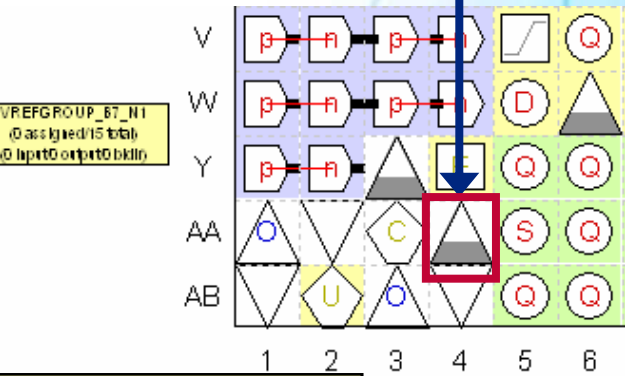
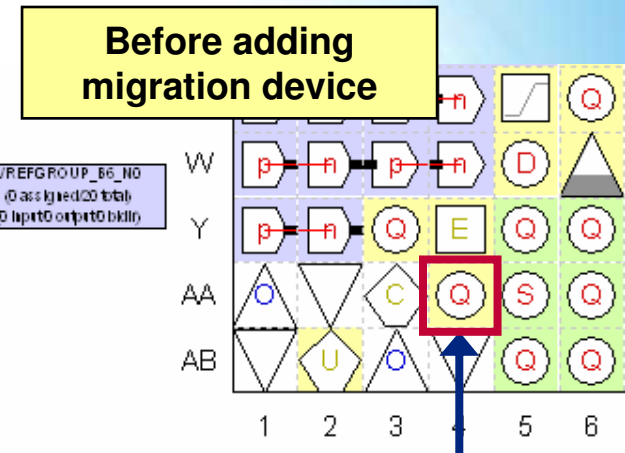
- Use to lock fitter-chosen (green) pin assignments for future compilations
  - Copies device & resource locations chosen by fitter into QSF file
    - Pins
    - Logic
    - Routing
- “Locks down” locations in floorplan





# 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



Pin Migration View

Current Device: EP2515F484C3

**View ⇒ Pin Migration View**

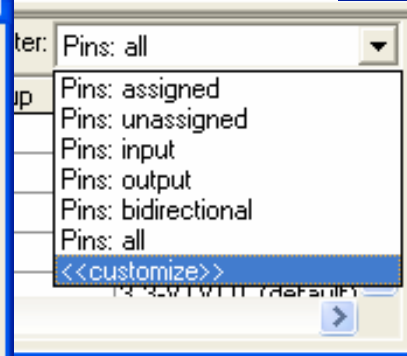
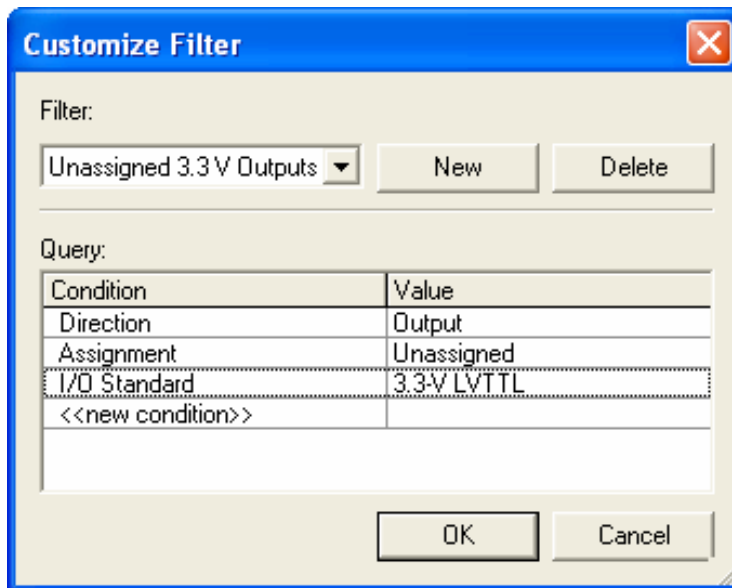
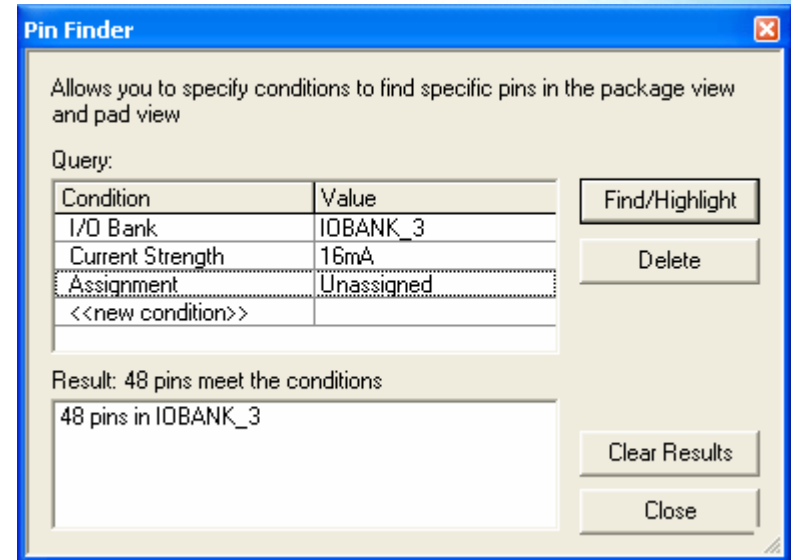
	Pin Number	Migration Result			Migration Devices					
		Pin Function	I/O Bank	VREF Group	EP2515F484C3			EP2560F484T4		
					Pin Function	I/O Bank	VREF Group	Pin Function	I/O Bank	VREF Group
1	PIN_A6	Column I/O	4	B4_N1	Column I/O	4	B4_N1	Column I/O	4	B4_N2
2	PIN_A7	Column I/O	4	B4_N1	Column I/O	4	B4_N1	Column I/O	4	B4_N2
3	PIN_A8	Column I/O	4	B4_N1	Column I/O	4	B4_N1	Column I/O	4	B4_N2
4	PIN_A10	Column I/O	9	B4_N1	Column I/O	9	B4_N1	Column I/O	9	B4_N2
5	PIN_A16	Column I/O	3	B3_N0	Column I/O	3	B3_N0	Column I/O	3	B3_N1
6	PIN_A17	Column I/O	3	B3_N0	Column I/O	3	B3_N0	Column I/O	3	B3_N1
7	PIN_A18	Column I/O	3	B3_N0	Column I/O	3	B3_N0	Column I/O	3	B3_N1
8	PIN_A19	Column I/O	3	B3_N1	Column I/O	3	B3_N1	Column I/O	3	B3_N2
9	PIN_A21	Dedicated...	3	B3_N1	Dedicated...	3	B3_N1	Dedicated...	3	B3_N2
10	PIN_AA4	VREFB7N2	7		Column I/O	7	B7_N1	VREFB7N2	7	B7_N2
11	PIN_AA5	Column I/O	7	B7_N0	Column I/O	7	B7_N0	Column I/O	7	B7_N1
12	PIN_AA12	Column I/O	8	B8_N1	Column I/O	8	B8_N1	Column I/O	8	B8_N2
13	PIN_AA13	Column I/O	8	B8_N1	Column I/O	8	B8_N1	Column I/O	8	B8_N2

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

# Additional Pin Planner Features

## ■ Pin Finder

- Locate pins meeting user-defined criteria with Pin Finder (Edit menu or right-click)
- Use to find compatible pin locations
- Found pins highlighted in Package View

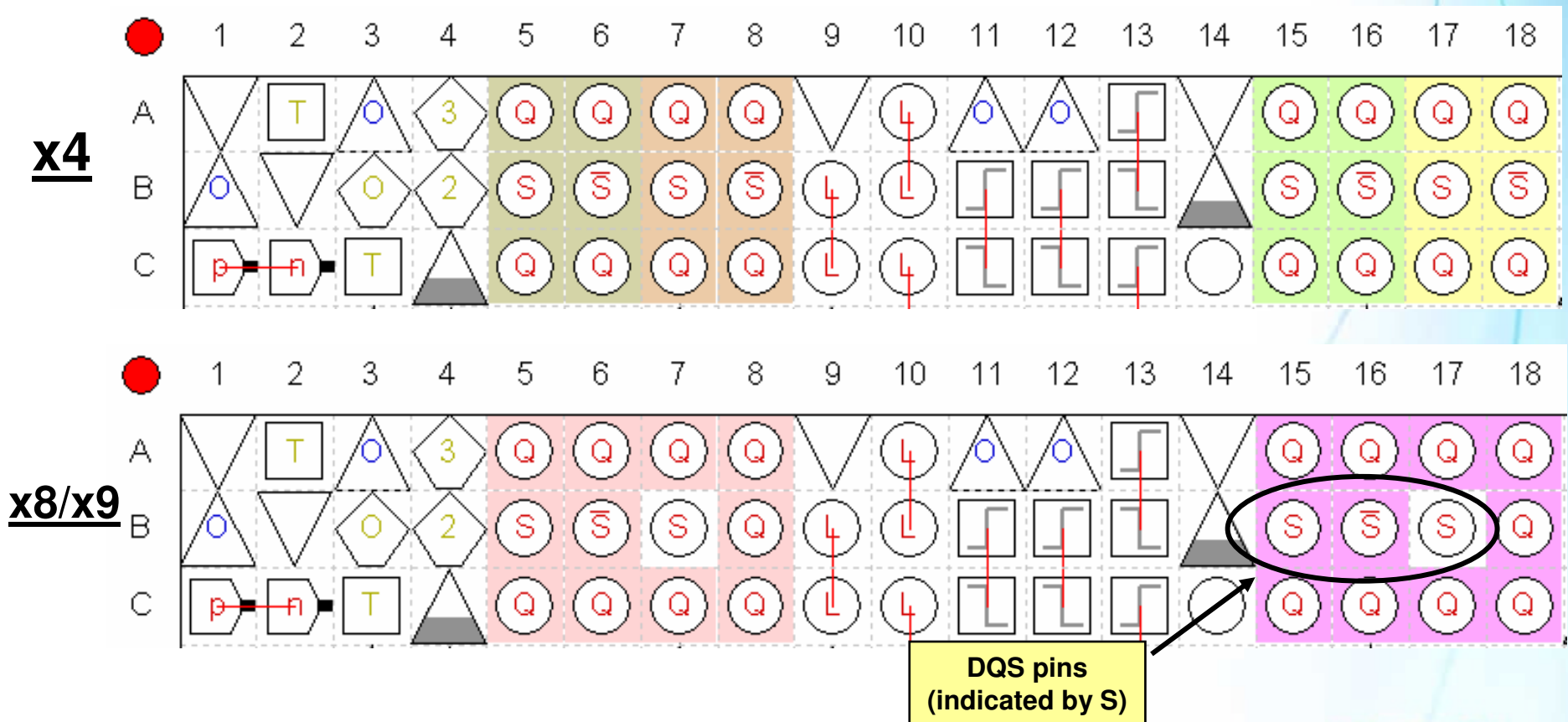


## ■ Custom Filters

- Create custom filters for All Pins list

# Show DQ/DQS Pins (View Menu)

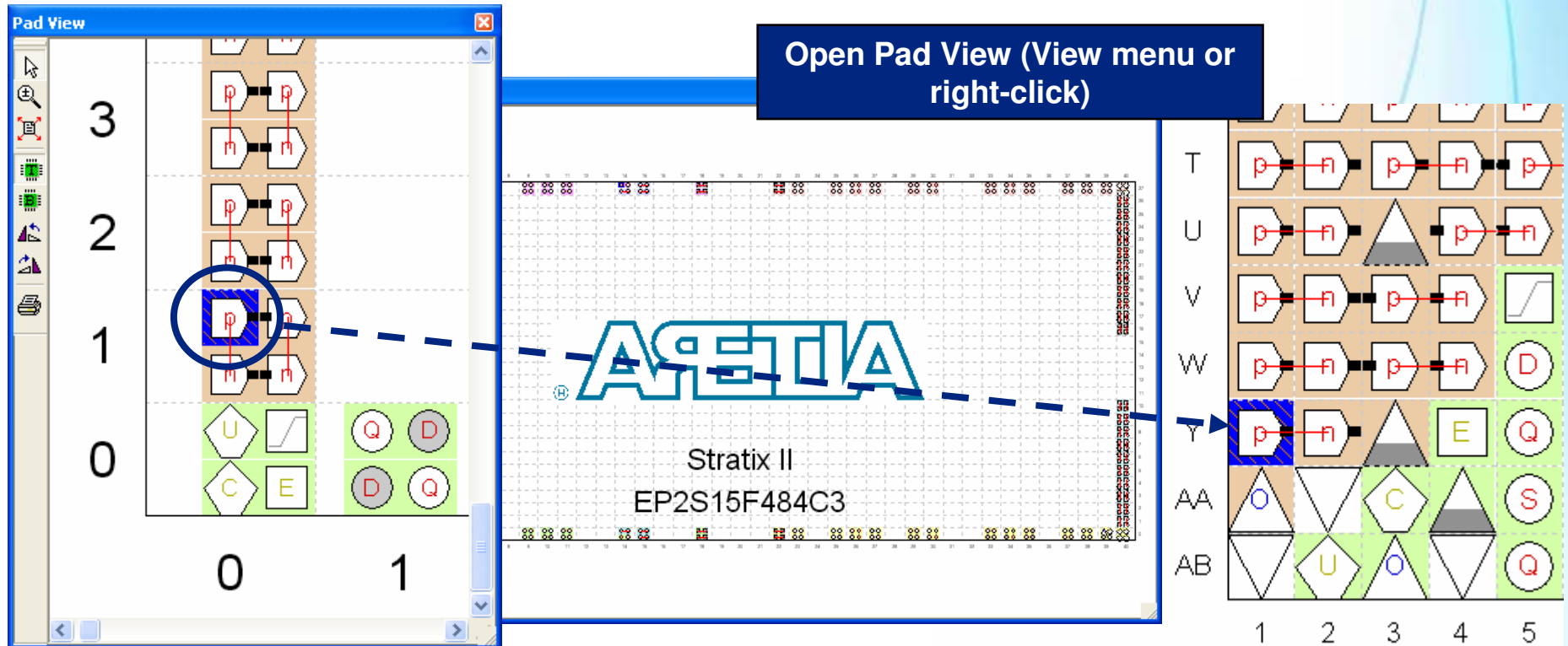
- Show color-coded DQ/DQS sets in x4, x8/x9, x16/x18, or x32/x36 modes in the Package View





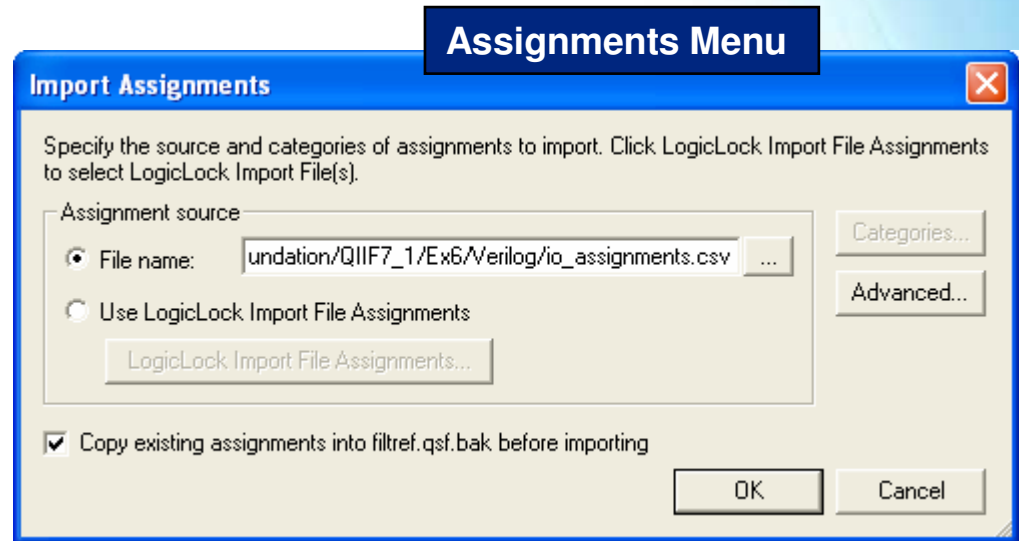
# Pad View

- Floating window to cross-reference package pin location to silicon pad location
  - Assign pins in Pad View based on pad location
- Reversed “Altera” indicates flip-chip die



# Import/Export via CSV

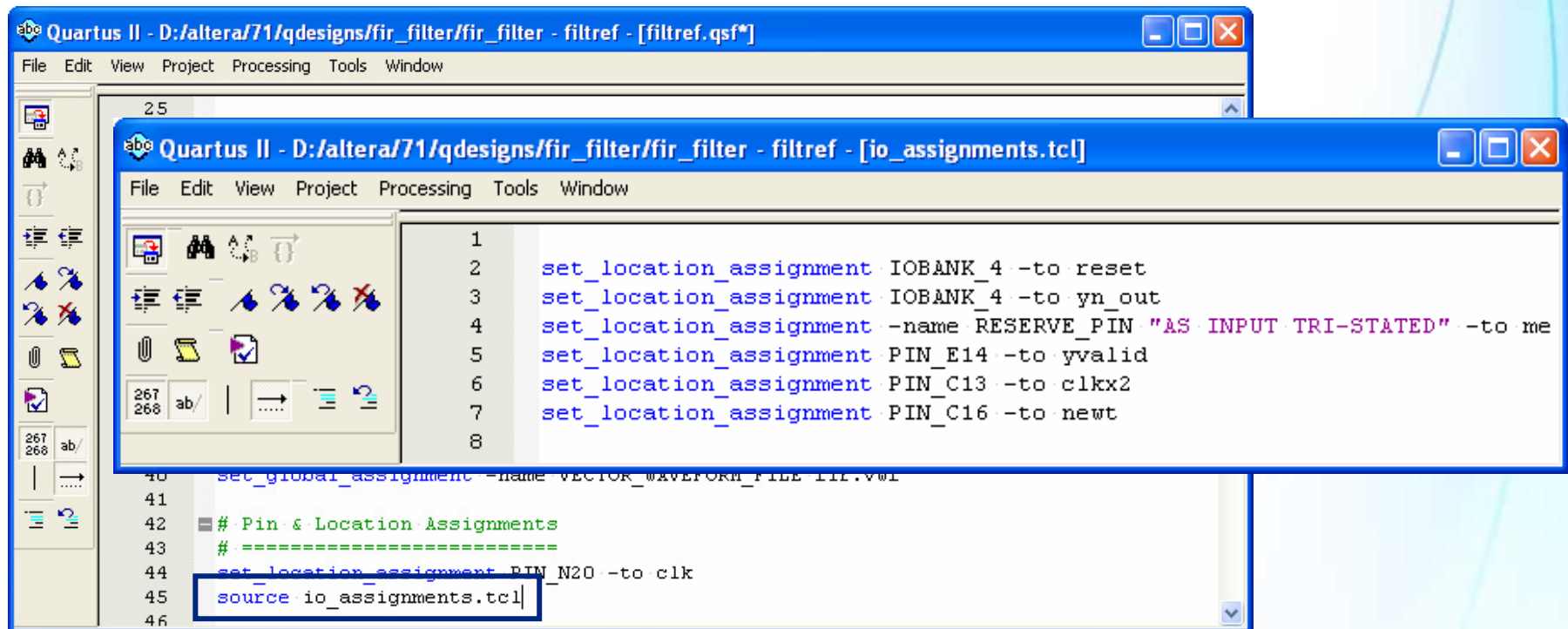
- Use spreadsheet Comma Separated Value (.CSV) file to enter or edit I/O locations
- CSV column names must match Pin Planner column headings
  - To
    - Pin name
  - Assignment Name
    - Location
  - Value
    - PIN\_<pin\_number>
  - I/O standard



	A	B	C
1	To	Assignment Name	Value
2	d[7]	Location	PIN_J4
3	d[6]	Location	PIN_H4

# Type I/O Assignments & 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



```
1
2  set_location_assignment IOBANK_4 -to reset
3  set_location_assignment IOBANK_4 -to yn_out
4  set_location_assignment -name RESERVE_PIN "AS INPUT TRI-STATED" -to me
5  set_location_assignment PIN_E14 -to yvalid
6  set_location_assignment PIN_C13 -to clkx2
7  set_location_assignment PIN_C16 -to newt
8
```

```
40  set_global_assignment -name VECTOR_WAVEFORM_FILE fil.vwf
41
42  # Pin & Location Assignments
43  # =====
44  set_location_assignment PIN_N20 -to clk
45  source io_assignments.tcl
46
```

# Using Synthesis Attributes in HDL

- chip\_pin
- altera\_chip\_pin\_lc
  - For compatibility with other synthesis tools

```
entity io_ex_vhdl is
  port (
    data_in : std_logic_vector (3 downto 0);
    data_out : std_logic);
end entity io_ex_vhdl;

attribute chip_pin : string;
attribute chip_pin of data_in : signal is "D4, D5, D6, D7";
attribute chip_pin of data_out : signal is "E9";
```

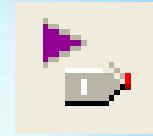
VHDL Example

```
module io_ex_ver (data_in, data_out);
  input [3:0] data_in /* synthesis chip_pin = "D4, D5, D6, D7" */;
  output data_out /* synthesis chip_pin = "E9";
```

Verilog Example

**Note:** Other I/O-related features can be specified in HDL. See *Quartus II Handbook* chapter "Quartus II Integrated Synthesis" for more details.

# I/O Assignment Analysis Command



Processing menu ⇒  
Start ⇒ Start I/O  
Assignment Analysis

Run from Pin  
Planner toolbar



- Use to check legality of all I/O assignments without full compilation
- Requirements
  - 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

# 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. A PLL that must be driven by 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
- Output pin loading
- I/O rules checking\*

**Flow Summary**

Flow Status	Successful - Tue May 08 16:00:56 2007
Quartus II Version	7.1 Build 156 04/30/2007 SJ Full Version
Revision Name	filterf
Top-level Entity Name	filterf
Family	Stratix II
Device	EP2S15F484C3
Timing Models	Final
Met timing requirements	N/A
Logic utilization	N/A
Combinational ALUTs	58
Dedicated logic registers	58
Total registers	58
Total pins	22 / 335 ( 7 % )
Total virtual pins	0
Total block memory bits	0 / 419,328 ( 0 % )
DSP block 9-bit elements	0 / 96 ( 0 % )
Total PLLs	0 / 6 ( 0 % )
Total DLLs	0 / 2 ( 0 % )

**Partial placement results also shown in floorplan**

**Detailed messages on I/O assignment issues**

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

Messages:

Type	Flag	Message
Info		Info: Selected device migration path implemented 8 pin(s) as GND
Warning		Warning: Devices selected for migration have different speed grades
Info		Info: Selected device migration path cannot use 8 pins as regular I/Os
Info		Info: Selected device migration path cannot use 8 pins as DQS I/Os
Info		Info: Selected device migration path cannot use 5 pins as nDQS I/Os
Info		Info: Selected device migration path cannot use 2 pins as DQ I/Os
Info		Info: Fitter converted 1 user pins into dedicated programming pins

\*Note: See Appendix for special reports and information generated only for Stratix II, II GX, and HardCopy II devices

© 2007 Altera Corporation—**Confidential**

Altera, Stratix, Arria, Cyclone, MAX, HardCopy, Nios, Quartus, and MegaCore are trademarks of Altera Corporation

# Question:

*How do I validate an I/O pin-out for board development?*

- Completed design
  - Run full compilation
  - Enable option to run I/O analysis before fitting
    - Checks for I/O layout problems before starting a possible time-consuming fit
    - Settings ⇒ Compilation Process Settings ⇒ Run I/O assignment analysis before compilation
- 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 (discussed next)



# Early I/O Planning Methodology

- Process to produce validated pin layout without needing design files
- Steps
  1. Create I/O assignments using Pin Planner only
  2. Implement I/O in HDL
  3. Add all I/O timing constraints
  4. Continue design flow

# 1) I/O Assignments Using Pin Planner Only

- Purpose: Verify I/O-I/O relationships
- Tasks
  - Enter pin names
  - Enter all pin-related assignments
    - I/O voltage standard
    - Current drive strength
    - PCI I/O support
    - On-chip termination
  - Reserve I/O as inputs or outputs based on direction
    - Reserve bidirectionals as outputs as typically more restrictive
  - Run I/O Assignment Analysis to check

# Notes on Step 1

- Place most restrictive pins first
  - E.g. differential pins, double-data rate, etc.
- Assign to I/O blocks or I/O region for more flexibility
  - Let fitter choose exact locations
- As alternative, use spreadsheet or script to enter I/O information

## 2) Implement I/O in HDL

- Purpose: Verify I/O in relation to core
- Tasks
  - Identify clocks & other global signals
    - Use GLOBAL SIGNAL assignment
  - Add I/O-related megafunctions & IP and make I/O assignments
    - PLLs (ALTPLL)
    - SERDES (ALTLVDS - with or without Dynamic Phase Alignment)
    - High-speed IP (e.g. RapidIO, HyperTransport)
    - DDR/QDR IP Cores
  - Set up and create top-level HDL file with all design I/O
    - Create file manually or use Quartus II commands to configure and create automatically
- Run I/O Assignment Analysis to check

# Adding I/O-Related Functions

- Import Pin Planner File (.PPF)
  - Used to import I/O names and settings
  - Created automatically by MegaWizard when creating I/O-related megafunctions
  - Creates new group in Groups List of Pin Planner
- Two methods
  - Import PPF from previously created custom megafunction
  - Create new megafunction directly from Pin Planner
- Set up and create top-level HDL file
  - Based on only I/O megafunctions and I/O assignments

***Note: See Appendix, Quartus II Handbook “I/O Management” (Volume 2), and online training “I/O Management” for information about importing/creating megafunctions through the Pin Planner and creating a top-level design file.***

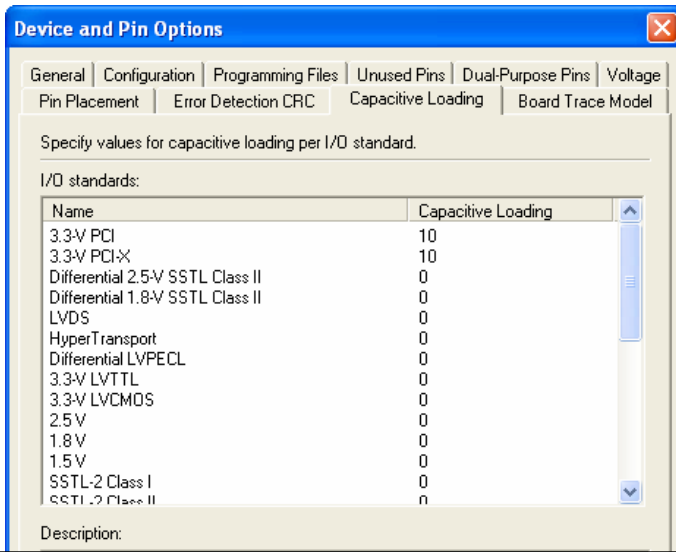
### 3) Add All I/O Timing Constraints

- Fully constrain I/O for timing analysis
- Discussed in timing analysis section

## 4) Continue design flow

- Merge validated pin assignments into another design project or revision
  - Use one project or revision to do I/O verification and another to start development of internal logic
- Use generated top-level HDL file as basis for design project
  - Remove virtual pins
  - Connect internal megafunction ports to internal logic

# “Board-Aware” Settings: Output Pin Load



**Capacitive Loading tab of Device and Pin Options button in Device Settings**

- Specifies output pin loading in picofarads (pf)
  - Changes default loading value of I/O standard
  - Changes  $t_{co}$  of output pins
- Allows designer to accurately model board conditions
- Specify for entire I/O standard in Device Settings
- Apply to individual output or bidirectional pins in Assignment Editor or Pin Planner All Pins list

Node Name	Direction	Location	I/O Standard	Output Pin Load	I/O Bank
13	reset	Input	PIN_N3	3.3-V LVTTTL (default)	6
14	yn_out[7]	Output	PIN_J6	3.3-V LVTTTL (default)	20
15	yn_out[6]	Output	PIN_L8	3.3-V LVTTTL (default)	20
16	yn_out[5]	Output	PIN_H1	3.3-V LVTTTL (default)	20
17	yn_out[4]	Output	PIN_K2	3.3-V LVTTTL (default)	20
18	yn_out[3]	Output	PIN_H2	3.3-V LVTTTL (default)	20
19	yn_out[2]	Output	PIN_J5	3.3-V LVTTTL (default)	20
20	yn_out[1]	Output	PIN_L2	3.3-V LVTTTL (default)	20
21	yn_out[0]	Output	PIN_K5	3.3-V LVTTTL (default)	20
22	yvalid	Output	PIN_L7	3.3-V LVTTTL (default)	20
23	~DATA0~	Input	PIN_E13	3.3-V LVTTTL (default)	3

**Tcl: set\_instance\_assignment -name OUTPUT\_PIN\_LOAD <value> -to <pin name>**



# Advanced I/O Timing (Stratix II & III devices only)

- Enhances analysis (over capacitive loading) by allowing user to enter board-level parameters
  - Use in lieu of or in addition to HSPICE & IBIS modeling
- View signal integrity metrics in Compilation Report (TimeQuest folder)

**Enable in TQ settings, then Device Settings ⇒ Device & Pin Options**

Name	Value
Near pull-up resistance	open
Near pull-down resistance	open
Near capacitance	open
Near series resistance	short
Transmission line distributed inductance	0
Transmission line distributed capacitance	0
Transmission line length	0
Far pull-up resistance	open
Far pull-down resistance	open
Far capacitance	open
Far series resistance	short
Termination voltage	0

**Set for all pins using I/O standard**

**Set parameters for specific I/O pin(s)**

Stratix II  
EP2S15F484C3  
pin(s): yn\_out[7]; yn\_out[6]; yn\_out[5]; yn\_out[4]; yn\_out[3]; yn\_out[2]; yn\_out[1]; yn\_out[0]  
I/O standard for selected pin(s): 3.3-V LVTTTL

**Right-click on output pin(s) in Pin Planner ⇒ Board Trace Model**

# PCB Tool Support

- Mentor Graphics DxDesigner & I/O Designer software
- Cadence Allegro Design Entry software
- OrCAD Capture

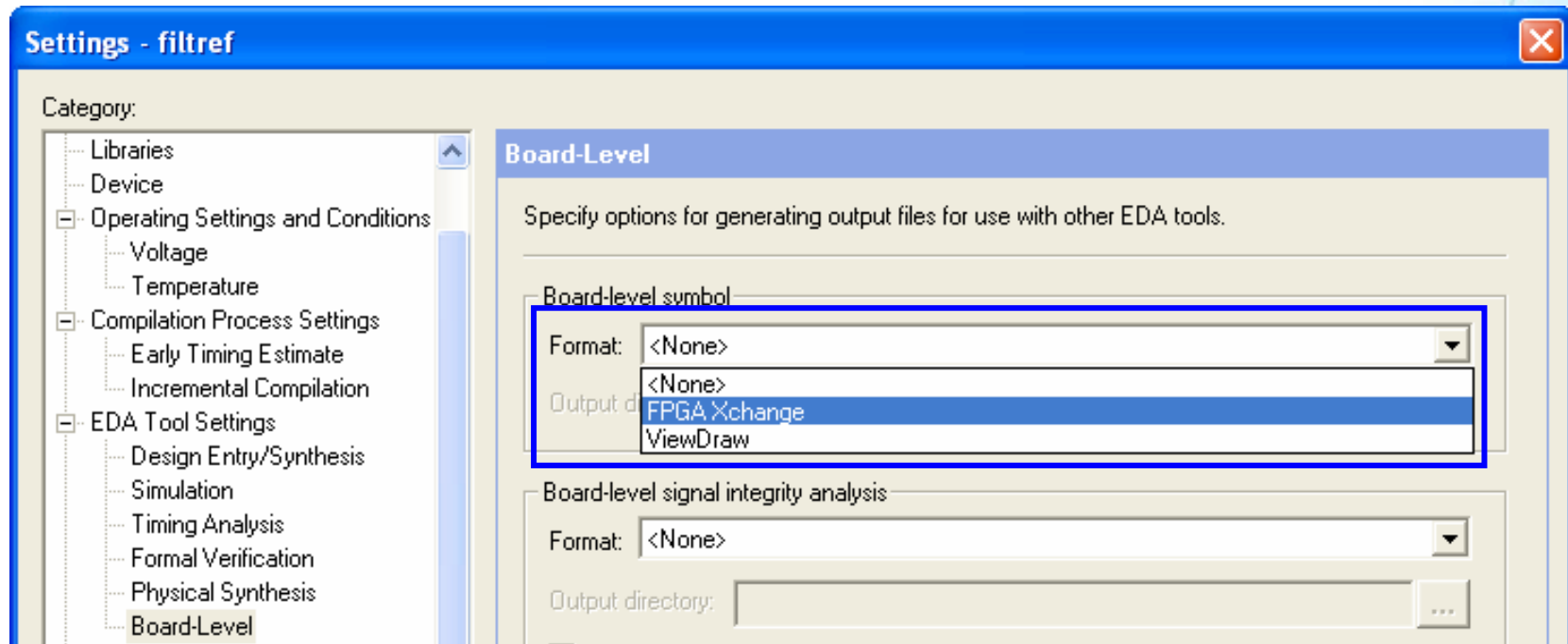
For more detail on interfacing, see Quartus II Handbook chapters “*Mentor Graphics PCB Design Tools Support*” & “*Cadence PCB Design Tools Support*” (Volume 2)

# Transferring I/O Assignments

- **FPGA Xchange file (.FX)**
  - Use to transfer pin assignments between Quartus II software & I/O Designer software
  - Contains only assigned pins
  - Not created automatically; must manually turn on .fx file generation (EDA Netlist Writer)
- **Pin-Out file (.PIN)**
  - Uses
    - Transferring pin assignments from Quartus II software to PCB tools
      - Cannot be imported back into Quartus II software
    - Generating symbols in PCB tools
  - Contains all pins, included unused
  - Automatically generated during fitting or I/O assignment analysis

# Generating .FX File

- Choose FPGA Xchange (Assignments ⇒ EDA Tools Settings ⇒ Board-Level)
- Compile or run EDA Netlist Writer (Processing ⇒ Start)



# .PIN File

```

Pin-Out File
48  -- GND+      : Unused input pin. It can also be used to report unused dual-purpose pins.
49  --          : This pin should be connected to GND. It may also be connected to a
50  --          : valid signal on the board (low, high, or toggling) if that signal
51  --          : is required for a different revision of the design.
52  -- GND*     : Unused I/O pin. This pin can either be left unconnected or
53  --          : connected to GND. Connecting this pin to GND will improve the
54  --          : device's immunity to noise.
55  -- RESERVED : Unused I/O pin, which MUST be left unconnected.
56  -- RESERVED_INPUT : Pin is tri-stated and should be connected to the board.
57  -- RESERVED_INPUT_WITH_WEAK_PULLUP : Pin is tri-stated with internal weak pull-up resistor.
58  -- RESERVED_INPUT_WITH_BUS_HOLD   : Pin is tri-stated with bus-hold circuitry.
59  -- RESERVED_OUTPUT_DRIVEN_HIGH    : Pin is output driven high.
60  -----
61
62  Quartus II Version 7.1 Build 156 04/30/2007 SJ Full Version
63  CHIP "filtref" ASSIGNED TO AN: EP2S15F484C3
64
65  Device Migration List: "EP2S60F484I4"
66  Pin Name/Usage      : Location : Dir.  : I/O Standard  : Voltage : I/O Bank : User Assignment
67  -----
68  GND                  : A1      : gnd   :                :          :          :
69  TEMPDIODEp          : A2      :       :                :          :          :
70  VCCIO4               : A3      : power :                : 3.3V    : 4        :
71  MSEL3                : A4      :       :                :          : 4        :
72  GND*                 : A5      :       :                :          : 4        :
73  GND*                 : A6      :       :                :          : 4        :
74  GND*                 : A7      :       :                :          : 4        :
75  GND*                 : A8      :       :                :          : 4        :
76  GND                  : A9      : gnd   :                :          :          :
77  GND*                 : A10     :       :                :          : 9        :
78  VCCIO4               : A11     : power :                : 3.3V    : 4        :
79  VCCIO3               : A12     : power :                : 3.3V    : 3        :
80  GND*                 : A13     :       :                :          : 3        :
81  GND                  : A14     : gnd   :                :          :          :
  
```

Description section

I/O names & settings



# *Exercise 5 Demonstration*

# 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 way or creating and managing pin assignments
- I/O Assignment Analysis helps validate a device-pin out without performing full compilations
- Pin validation can be completed during any point in design development

# I/O Planning Support Resources

- Quartus II Handbook chapters
  - *“I/O Management”* (Volume 2)
  - *“Signal Integrity Analysis with Third-Party Tools”* (Volume 3)
  - *“Mentor Graphics PCB Design Tools Support”* (Volume 2)
  - *“Cadence PCB Design Tools Support”* (Volume 2)
- Training Courses and Demonstrations
  - Online tutorial: *“FPGA to Board Design Flow Using Mentor Graphics Tools”*





# Quartus II Software Design Series: Foundation

## *Timing Analysis*



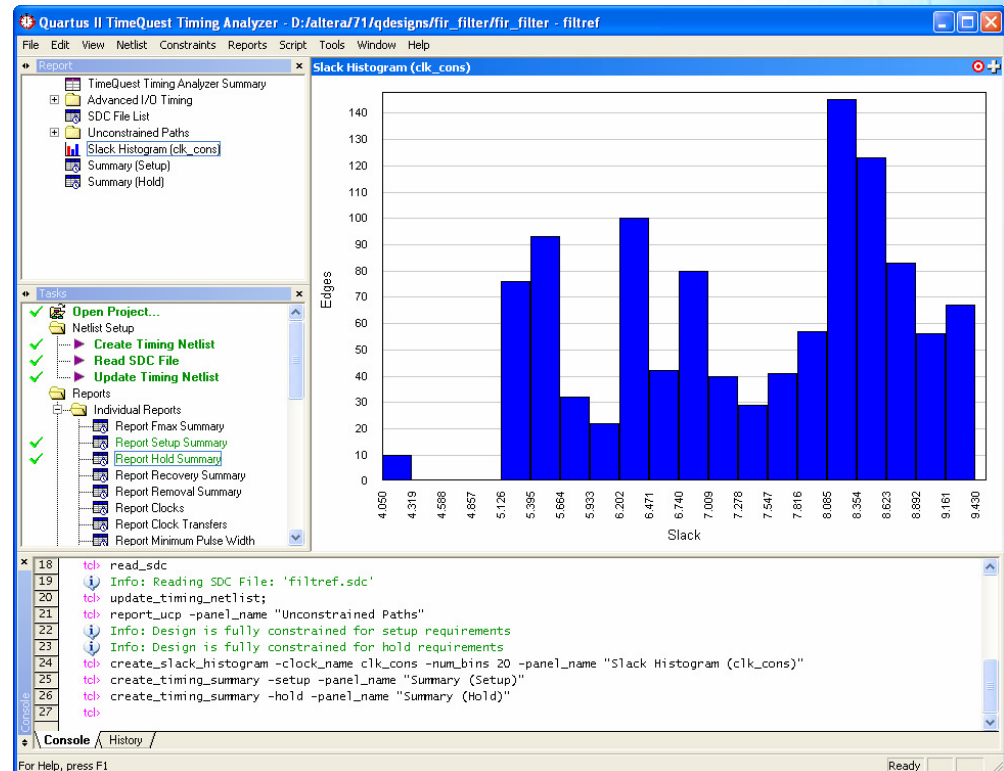
# Timing Analysis Agenda

- TimeQuest GUI
- Using TimeQuest
- Using TimeQuest in the Quartus II flow


*Note: For more details on verifying designs for timing, please attend the course “Quartus II Software Design Series: Verification”*

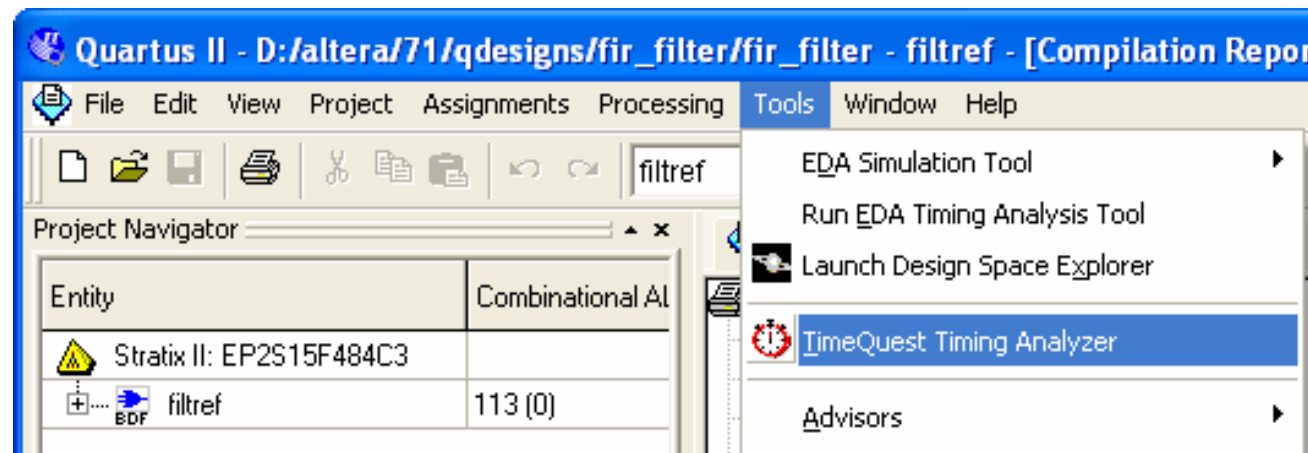
# TimeQuest Timing Analyzer

- New timing engine in Quartus II software
- Provides timing analysis solution for all levels of experience
- Features
  - Synopsys Design Constraints (SDC) support
    - Standardized constraint methodology
  - Easy-to-use interface
    - Constraint entry
    - Standard reporting
  - Scripting emphasis
    - Presentation focuses on using GUI



# Opening TimeQuest

- Toolbar button 
- Tools menu
- Stand-alone mode
  - `quartus_staw`
- Command line



# TimeQuest GUI

Menu access all TimeQuest features

Report Pane

Tasks Pane

View Pane

Console Pane

The screenshot shows the Quartus II TimeQuest Timing Analyzer interface. The main window title is "Quartus II TimeQuest Timing Analyzer - D:\altera\71\desigens\fir\_filter\fir\_filter - filtref". The menu bar includes File, Edit, View, Netlist, Constraints, Reports, Script, Tools, Window, and Help. The interface is divided into several panes:

- Report Pane:** Located on the left, it shows a tree view of reports including "TimeQuest Timing Analyzer Summary", "Advanced I/O Timing", "SDC File List", "Unconstrained Paths", "Slack Histogram (clk\_cons)", "Summary (Setup)", "Summary (Hold)", "Report Timing", "Command Info", "Path #1: Setup slack is 4.062", "Path #2: Setup slack is 4.102", "Path #3: Setup slack is 5.102", "Report Minimum Pulse Width", "Report TCCS", "Report RSKM", "Report SDC", "Report Unconstrained Paths", "Report Ignored Constraints", "Report Datasheet", "Check Timing", "Custom Reports", "Report Timing...", "Report Net Timing...", "Report Path...", "Report Minimum Pulse Width...", "Create Slack Histogram...", and "Macros".
- Tasks Pane:** Located below the Report Pane, it shows a list of tasks such as "Report Minimum Pulse Width", "Report TCCS", "Report RSKM", "Report SDC", "Report Unconstrained Paths", "Report Ignored Constraints", "Report Datasheet", "Check Timing", "Custom Reports", "Report Timing...", "Report Net Timing...", "Report Path...", "Report Minimum Pulse Width...", "Create Slack Histogram...", and "Macros".
- View Pane:** The central area displays timing analysis results for "Path #100: Setup slack is 6.459". It shows a "SLACK: 6.459 ns" and a "Path Delay Stats" pie chart. The pie chart is divided into two segments: "IC [3859] (78%)" in blue and "Cell [1072] (21%)" in red. Below the pie chart is a table for "Path Summary":
 

Property	Value
1 From Node	newt
2 To Node	taps:instkn[3]~DUPLICATE
3 Launch Clock	clk_cons
4 Latch Clock	clk_cons
5 Data Arrival Time	5.931
6 Data Required Time	12.390
7 Slack	6.459
- Console Pane:** Located at the bottom, it shows a list of Tcl commands and their outputs:
 

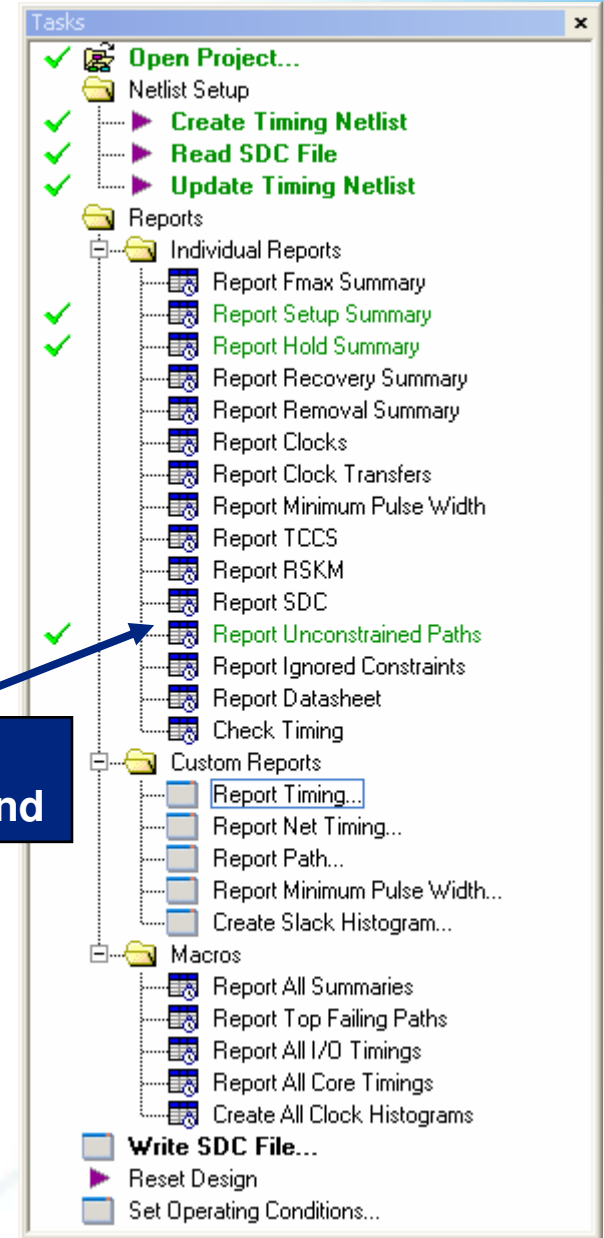
```

24 tcl> create_slack_histogram -clock_name clk_cons -num_bins 20 -panel_name "Slack Histogram (clk_cons)"
25 tcl> create_timing_summary -setup -panel_name "Summary (Setup)"
26 tcl> create_timing_summary -hold -panel_name "Summary (Hold)"
27 tcl> report_timing -from_clock clk_cons -setup -npaths 1 -detail path_only -panel_name {Report Timing}
28 Info: Report Timing: Found 1 setup paths (0 violated). Worst case slack is 4.062
33 ← 1 4.062
34 tcl> report_timing -from_clock clk_cons -setup -npaths 100 -detail path_only -panel_name {Report Timing}
35 Info: Report Timing: Found 100 setup paths (0 violated). Worst case slack is 4.062
40 ← 100 4.062
41 tcl>
      
```

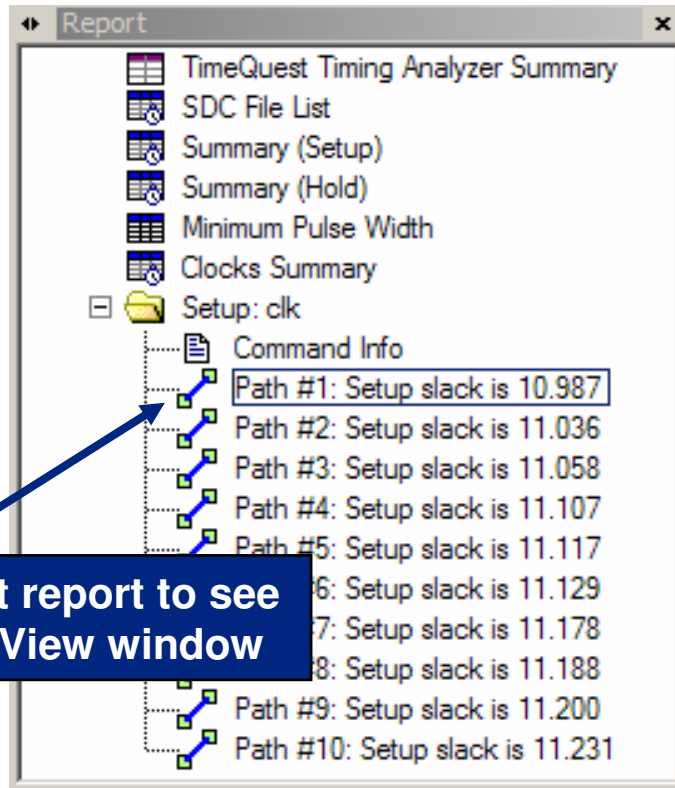
# Tasks Pane

- Provides quick access to common operations
  - Command execution
  - Report generation
- Executes most commands with default settings

Double-click to execute any command



# Report Pane



Highlight report to see detail in View window

- Displays list of reports currently available for viewing
  - Reports generated by Tasks pane
  - Reports generated using report commands

# View Pane

- Main viewing area that displays report table contents & graphical results

Minimum Pulse Width							
	Slack	Actual Width	Required Width	Pulse	Clock	Clock Edge	Target
1	4.388	5.000	0.612	High	clk_cons	Rise	inst4~ Duplicate_1
2	4.388	5.000	0.612	Low	clk		
3	4.388	5.000	0.612	High	clk		
4	4.388	5.000	0.612	Low	clk		
5	4.388	5.000	0.612	High	clk		
6	4.388	5.000	0.612	Low	clk		
7	4.388	5.000	0.612	High	clk		
8	4.388	5.000	0.612	Low	clk		
9	4.388	5.000	0.612	High	clk		
10	4.388	5.000	0.612	Low	clk		
11	4.388	5.000	0.612	High	clk		
12	4.388	5.000	0.612	Low	clk		
13	4.388	5.000	0.612	High	clk		
14	4.388	5.000	0.612	Low	clk		
15	4.388	5.000	0.612	High	clk		
16	4.388	5.000	0.612	Low	clk		
17	4.388	5.000	0.612	High	clk		
18	4.388	5.000	0.612	Low	clk		
19	4.388	5.000	0.612	High	clk		
20	4.388	5.000	0.612	Low	clk		
21	4.388	5.000	0.612	High	clk		
22	4.388	5.000	0.612	Low	clk		
23	4.388	5.000	0.612	High	clk		
24	4.388	5.000	0.612	Low	clk		
25	4.388	5.000	0.612	High	clk		

**Path Slack Report**

**Path #100: Setup slack is 6.459**

SLACK: 6.459 ns

Path Summary	
Property	Value
1 From Node	newt
2 To Node	taps:instkn[3]~DUPLICATE
3 Launch Clock	clk_cons
4 Latch Clock	clk_cons
5 Data Arrival Time	5.931
6 Data Required Time	12.390
7 Slack	6.459

Data Arrival Path						
Total	Incr	RF	Type	Fanout	Location	Element
1	0.000	0.000				launch edge time
2	0.000	0.000	R			clock network delay
3	1.000	1.000	F	iExt	1	PIN_M3
4	1.864	0.864	FF	CELL	52	IOC_X40_Y16_N1
5	5.723	3.859	FF	IC	1	LCCOMB_X39_Y18_N30
6	5.776	0.053	FF	CELL	2	LCCOMB_X39_Y18_N30
7	5.776	0.000	FF	IC	1	LCFF_X39_Y18_N31

**Path Delay Stats**

**Type [Delay (ps)]**

- IC [3859] (78%)
- Cell [1072] (21%)

Data Required Path						
Total	Incr	RF	Type	Fanout	Location	Element
1	10.000	10.000				latch edge time
2	12.480	2.480	R			clock network delay
3	12.390	-0.090		uTsu	1	LCFF_X39_Y18_N31

**Timing Table**

**Timing Histogram**



# Viewing Multiple Reports

Path #100: Setup slack is 6.459

SLACK: 6.459 ns

Path Delay Stats

Path Summary

Property	Value
1 From Node	newt
2 To Node	taps:inst xn[3]~DUPLICATE
3 Launch Clock	clk_cons
4 Latch Clock	clk_cons
5 Data Arrival Time	5.931
6 Data Required Time	12.390
7 Slack	6.459

Click & drag '+' sign to divide view pane into multiple windows

Type [Delay (ps)]

- IC [3859] (78%)
- Cell [1072] (21%)

Data Arrival Path

	Total	Incr	RF	Type	Fanout	Location	Element
1	0.000	0.000					launch edge time
2	0.000	0.000	R				clock network delay
3	1.000	1.000	F	iExt	1	PIN_M3	newt
4	1.864	0.864	FF	CELL	52	IOC_X40_Y16_N1	newt combout
5	5.723	3.859	FF	IC	1	LCCOMB_X39_Y18_N30	inst xn[3]_OTERM125~DUPLICATE dataf
6	5.776	0.053	FF	CELL	2	LCCOMB_X39_Y18_N30	inst xn[3]_OTERM125~DUPLICATE combout
7	5.776	0.000	FF	IC	1	LCFF_X39_Y18_N31	inst xn[3]~DUPLICATE datain

Data Required Path

	Total	Incr	RF	Type	Fanout	Location	Element
1	10.000	10.000					latch edge time
2	12.480	2.480	R				clock network delay
3	12.390	-0.090		uTsu	1	LCFF_X39_Y18_N31	taps:inst xn[3]~DUPLICATE

# Viewing Multiple Reports Example

View pane split into two side-by-side windows

**Slack Histogram (clk\_cons)**

Slack	Edges
4.050	0
4.319	0
4.588	0
4.857	0
5.126	75
5.395	92
5.664	0
5.933	0
6.202	100
6.471	42
6.740	80
7.009	40
7.278	40
7.547	42
7.816	57
8.085	145
8.354	122
8.623	82
8.892	56
9.161	67
9.430	0

**SLACK: 6.459 ns**

**Path Summary**

Property	Value
1 From Node	newt
2 To Node	taps:instlvr
3 Launch Clock	clk_cons
4 Latch Clock	clk_cons
5 Data Arrival Time	5.931
6 Data Required Time	12.390
7 Slack	6.459

**Path Delay Stats**

**Type [Delay (ps)]**

- IC [3859] (78%)
- Cell [1072] (21%)

**Data Arrival Path**

Total	Incr	RF	Type
1 0.000	0.000		
2 0.000	0.000	R	
3 1.000	1.000	F	iEx
4 1.864	0.864	FF	CELL 52

**Data Required Path**

Total	Incr	RF	Type	Fanout	Location
1 10.000	10.000				
2 12.480	2.480	R			
3 12.390	-0.090		uTsu	1	LCFF_X39_Y18_I

Highlight window, then highlight report in Reports pane you would like to appear there

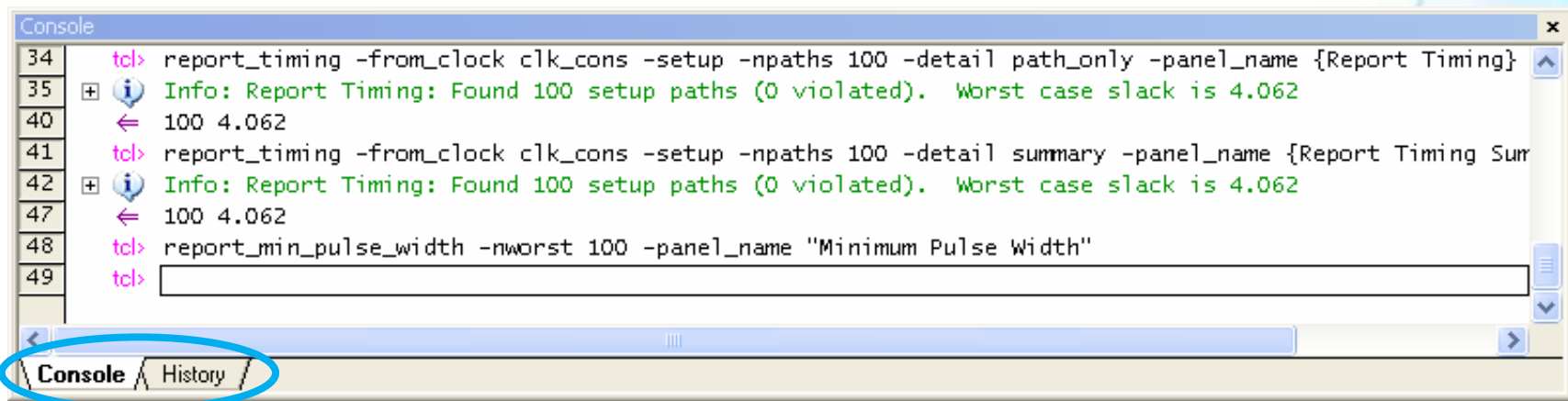
Use Target Pane button to force a selected report to appear in a pane

Drag bar to left or right to remove split



# Console pane

- Allows direct entry and execution of SDC & Tcl commands
  - Displays equivalent of command executed by GUI
- Displays TimeQuest output messages
- History tab records all executed SDC & Tcl commands



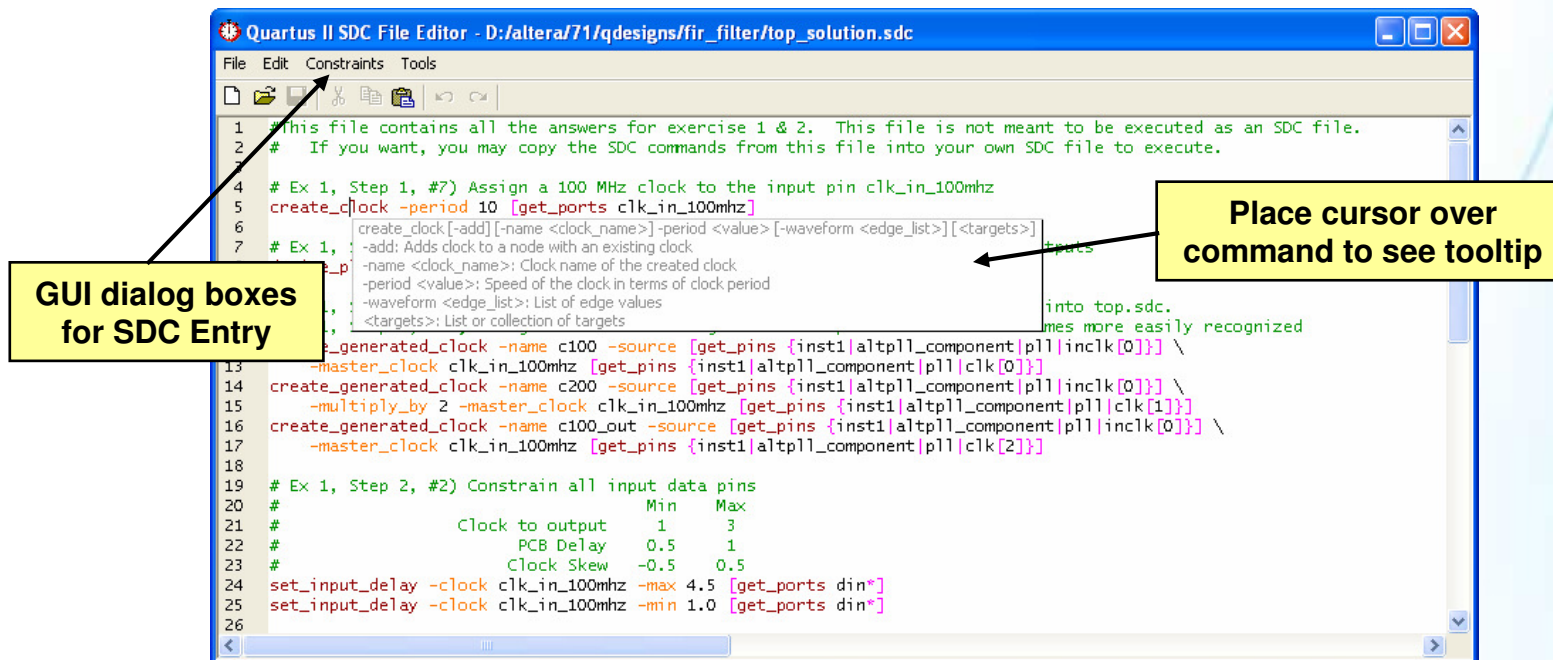
```
34 tcl> report_timing -from_clock clk_cons -setup -npaths 100 -detail path_only -panel_name {Report Timing}
35 [i] Info: Report Timing: Found 100 setup paths (0 violated). Worst case slack is 4.062
40 ← 100 4.062
41 tcl> report_timing -from_clock clk_cons -setup -npaths 100 -detail summary -panel_name {Report Timing Sumr
42 [i] Info: Report Timing: Found 100 setup paths (0 violated). Worst case slack is 4.062
47 ← 100 4.062
48 tcl> report_min_pulse_width -nworst 100 -panel_name "Minimum Pulse Width"
49 tcl>
```

# SDC File Editor

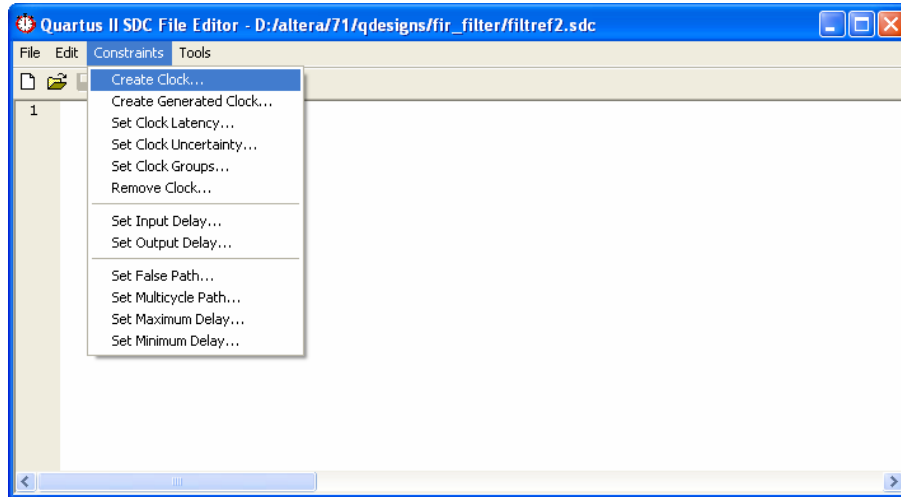
File menu ⇒ New/Open SDC File

## ■ Features

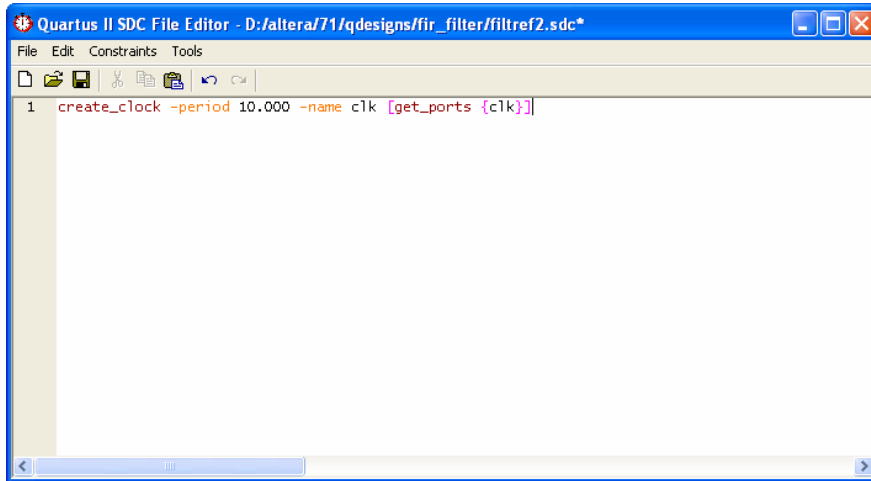
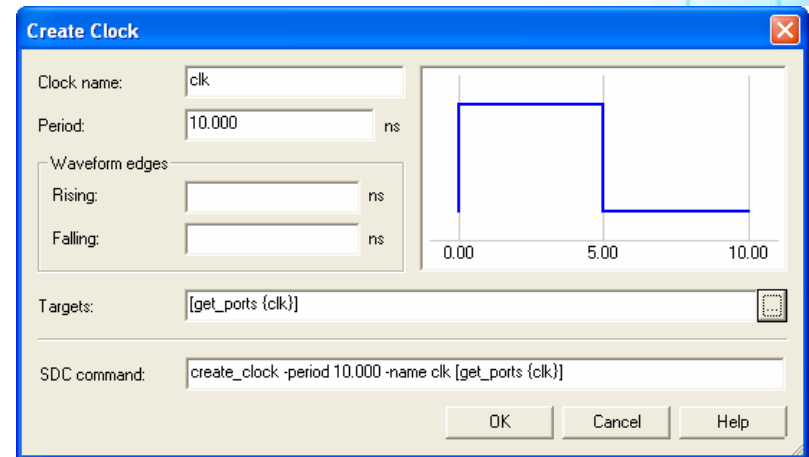
- Access to GUI dialog boxes for constraint entry
- Syntax coloring
- Tooltip syntax help



# SDC File Editor (cont.)



Construct an SDC file using TimeQuest graphical file tools

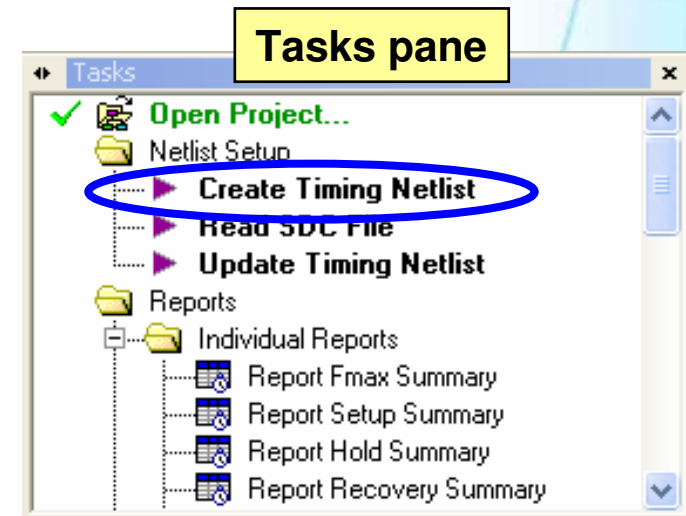
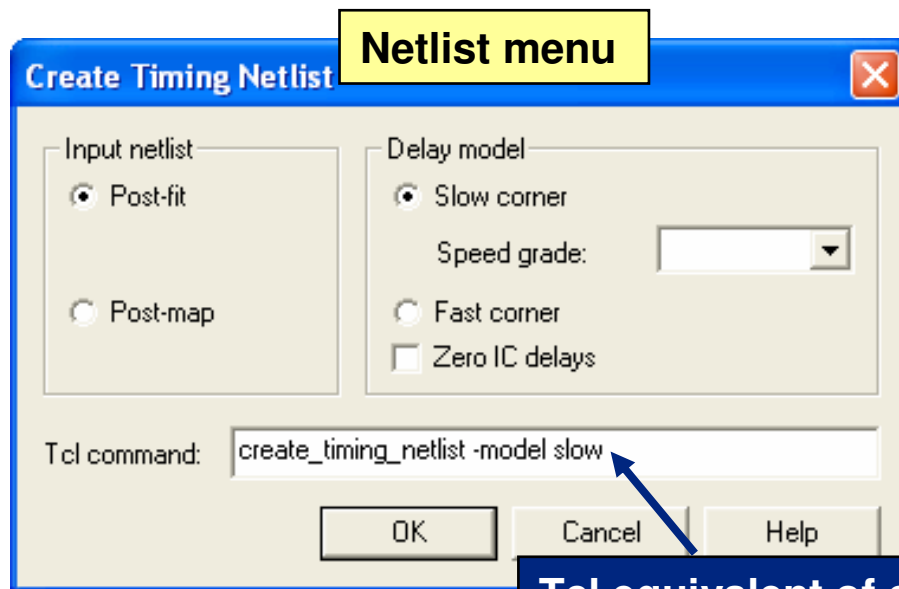


# Steps to Using TimeQuest

1. Generate timing netlist
  2. Create and read SDC file (optional)
- or*
3. Constrain design (optional)
  4. Update timing netlist
  5. Generate timing reports
  6. Save timing constraints (optional)

# 1) Generate Timing Netlist

- Creates timing netlist (i.e. database) based on compilation results
  - Post-synthesis (mapping) or post-fit
  - Worst-case (slow), best-case (fast) timing models, or set operating conditions (Stratix III and Cyclone III devices only)
- To execute:



***Tcl: create\_timing\_netlist***

© 2007 Altera Corporation—**Confidential**

Altera, Stratix, Arria, Cyclone, MAX, HardCopy, Nios, Quartus, and MegaCore are trademarks of Altera Corporation

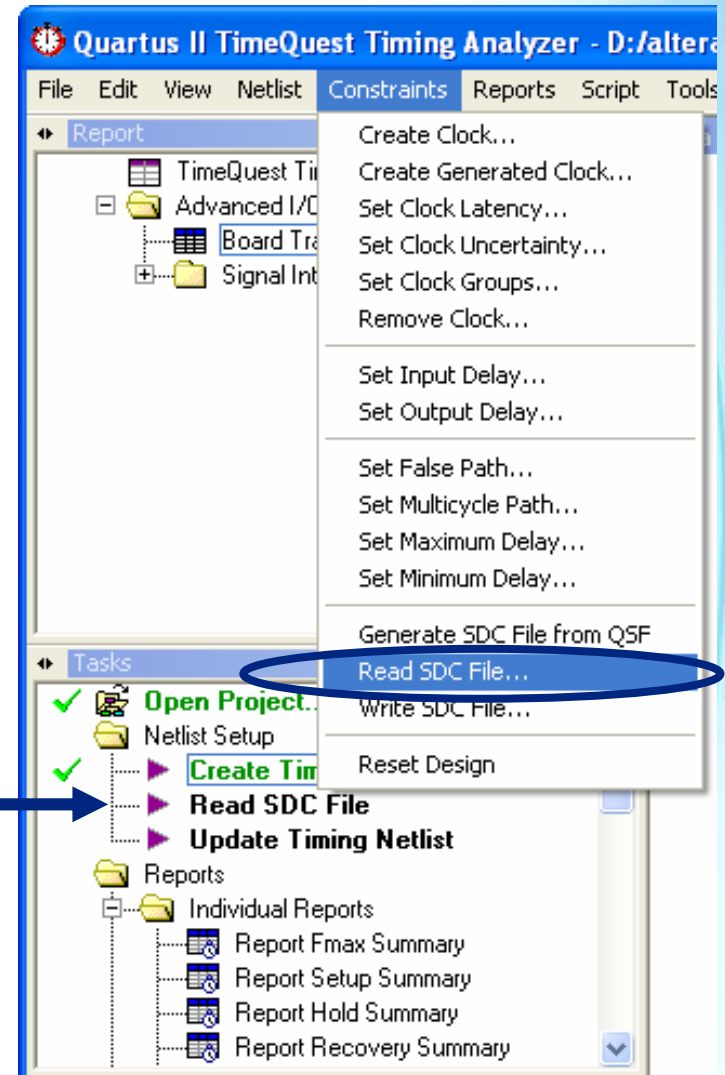
199

**ALTERA.**

## 2) Create & Read SDC File (Optional)

- Create SDC file using SDC file editor
- Read in constraints & exceptions from SDC file
  - Skip if no SDC file
- Execution
  - Read SDC File (Tasks pane or Constraints menu)
- File Precedence (if no filename specified)
  - Files specifically added to Quartus II project
  - <current\_revision>.sdc (if it exists)

```
Tcl: read_sdc [<filename>]
```

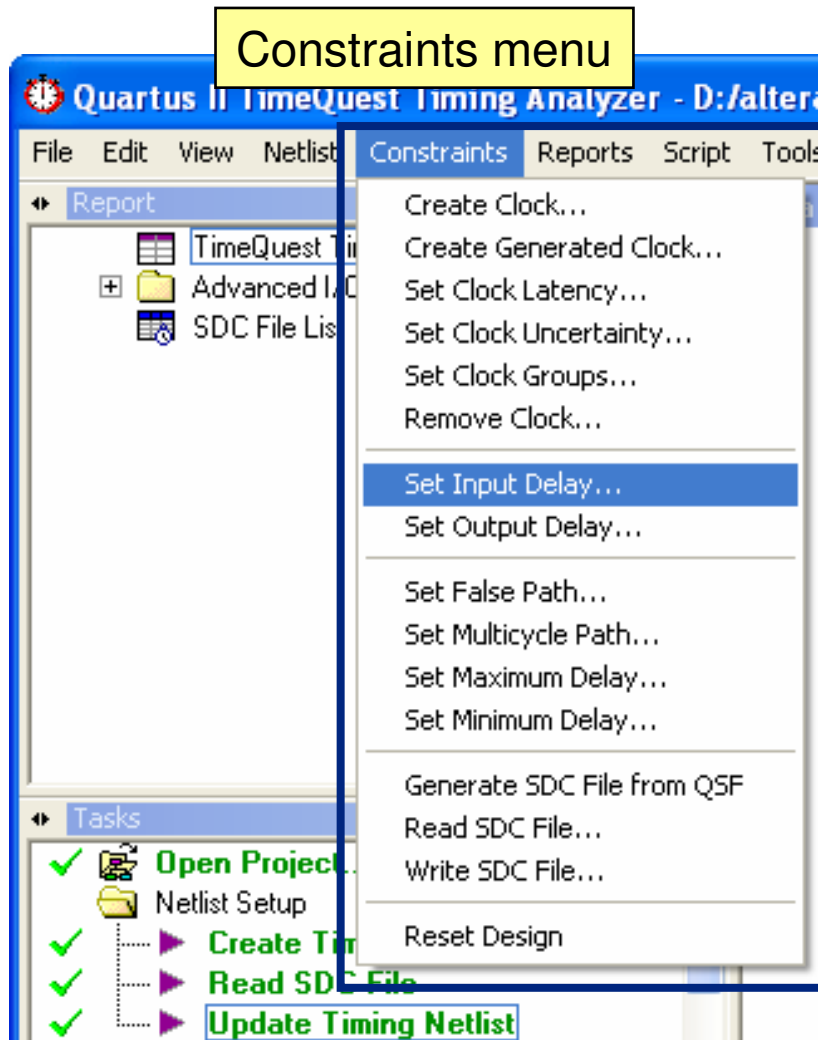




## 3) Constrain Design (Optional)

- Add new constraints directly
  - Not added to SDC file
  - Use GUI or Console pane
  - Not needed if all constraints in SDC file
- Examples
  - `create_clock`
  - `create_generated_clock`
  - `set_input_delay`
  - `set_output_delay`

# Using GUI to Enter Constraints



- Most common constraints can be accessed from the Constraints menu
- Use if unfamiliar with SDC syntax

# Constraining

- User **MUST** enter constraints to fully analyze design
  - TimeQuest only performs slack analysis on constrained design paths
  - Recommendation: Constrain all paths (clocks & I/O)
- May create and read SDC file (Step 2), enter constraints directly in TimeQuest (Step 3) or both

## 4) Update Timing Netlist

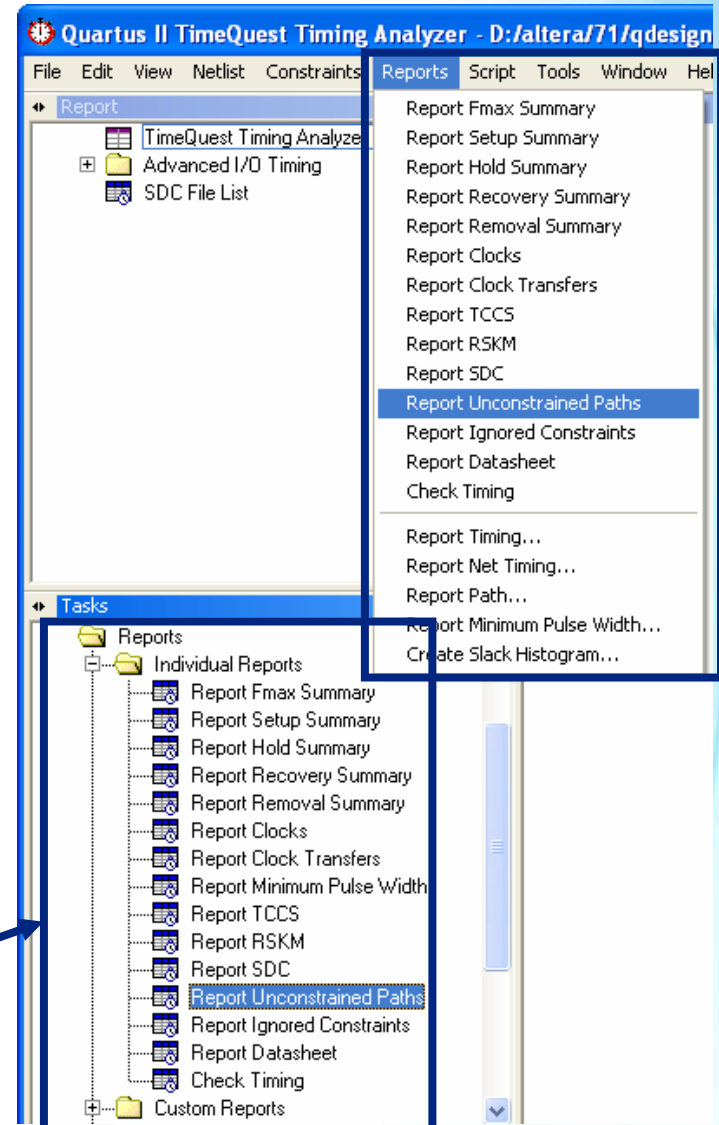
- Applies SDC constraints/exceptions to current timing netlist
- Generates warnings
  - Undefined clocks
  - Partially defined I/O delays
  - Combinatorial loops
- Update timing netlist after adding any new constraint
- Execution
  - Update Timing Netlist (Tasks pane or Netlist menu)

***Tcl: update\_timing\_netlist***

# 5) Generate Timing Reports

- Verify timing requirements and locate violations
- Check for fully constrained design or ignored timing constraints
- Two Methods
  - Tasks pane
    - Automatically creates/updates netlist & reads default SDC file if needed
  - Reports menu
    - Must have valid netlist to access
    - Tasks pane or Reports menu

**Double-click on individual report**



# “Out of Date” Reports

- Adding new constraints causes current reports to be “out of date”
- Update timing netlist & regenerate reports (report right-click menu)

The screenshot displays the Quartus II software interface. On the left, a 'Report' window shows a tree view of reports including 'TimeQuest Timing Analyzer Summary', 'Advanced I/O Timing', 'Board Trace Model Assignments', 'Signal Integrity Metrics', 'SDC File List', and 'Clocks Summary'. A right-click context menu is open over the 'Clocks Summary' report, showing options: 'Export...', 'Regenerate', 'Regenerate All Out of Date', 'Delete', 'Delete All Out of Date', and 'Delete All'. The 'Regenerate All Out of Date' option is highlighted. On the right, the 'Clocks Summary' report is displayed as a table with the following data:

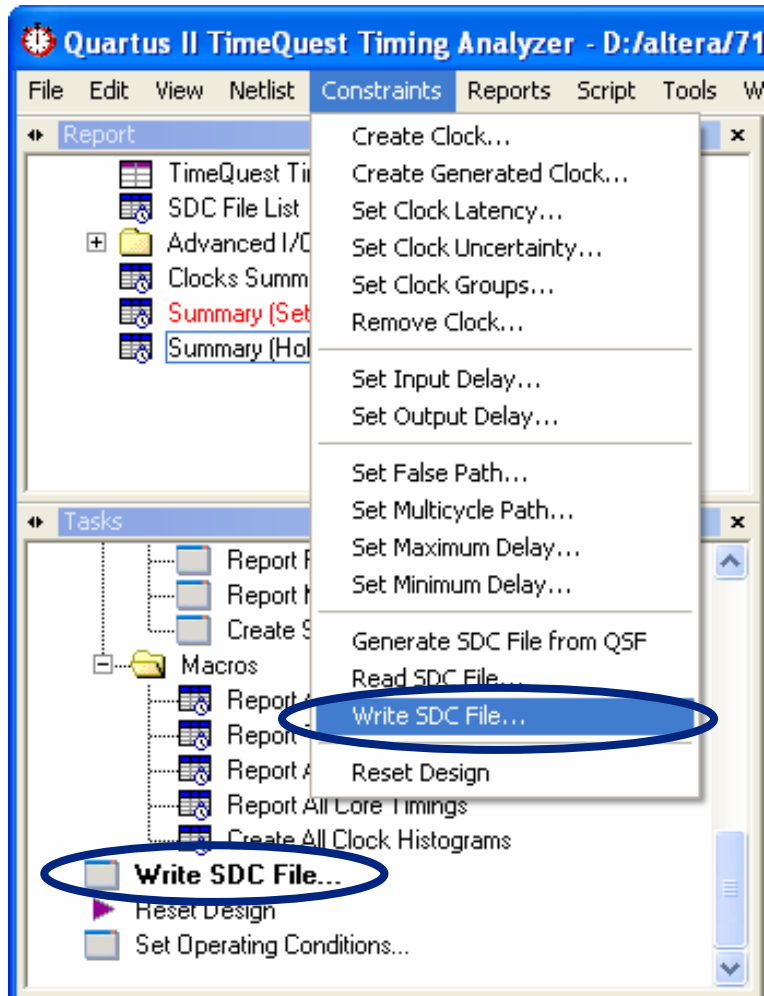
	Clock Name	Type	Period	Rise	Fall	Duty Cycle	Divide by	Multiply by	Phase
1	clk_cons	Base	10.000	0.000	5.000				
2	clkx2_cons	Base	10.000	0.000	5.000				

The background of the report area is yellow with a repeating 'OUT OF DATE' watermark.

# Reset Design Command

- Located in Tasks pane
- Flushes all timing constraints from current timing netlist
  - Functional Tcl equivalent: `delete_timing_netlist` command followed by `create_timing_netlist`
- Uses
  - “Re-starting” timing analysis on same timing netlist applying different constraints or SDC file
  - Starting analysis over if results seem to be unexpected

## 6) Save Timing Constraints (Optional)



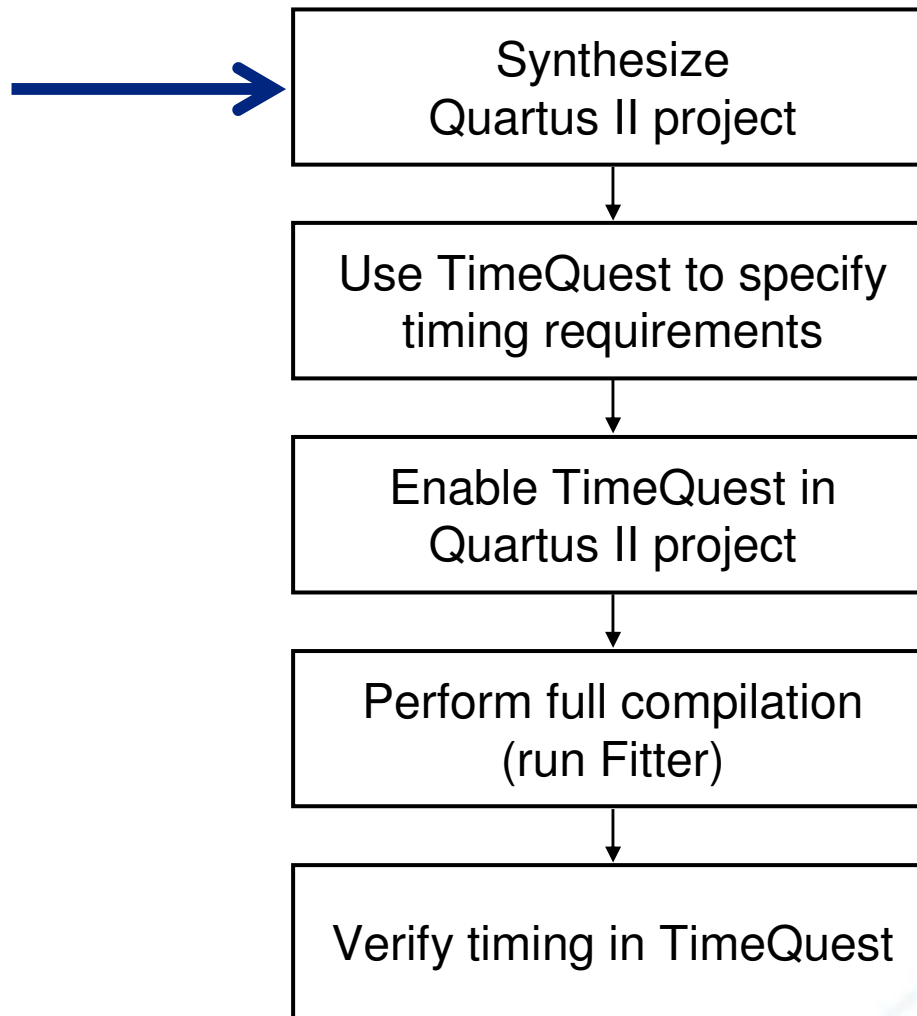
- `write_sdc` command
  - Saves all constraints & exceptions applied to current netlist into SDC file
  - Use if constraints added during TimeQuest session outside of input SDC file
- Notes
  - SDC files generated by TimeQuest only if requested
  - Converts Altera-specific SDC commands into standard SDC
  - Run `report_sdc` command to see what will get written to SDC file



# Steps to Using TimeQuest (Review)

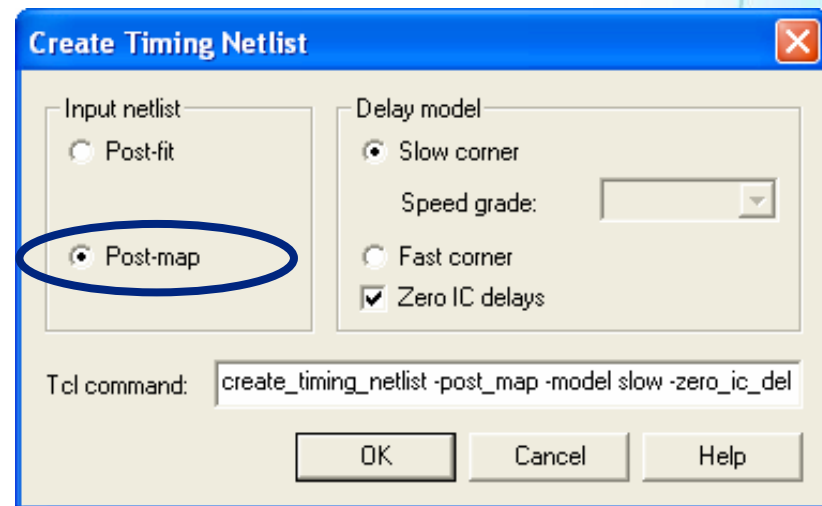
1. Generate timing netlist
  2. Read SDC file (optional)
- or*
3. Constrain design (optional)
  4. Update timing netlist
  5. Generate timing reports
  6. Save timing constraints (optional)

# Using TimeQuest in Quartus II Flow



# Specifying Timing Requirements

- Follow TimeQuest steps
- Use `-post_map` argument for synthesis (mapping) only netlist
  - If the design is already fully compiled, then choose `-post_fit` (default)

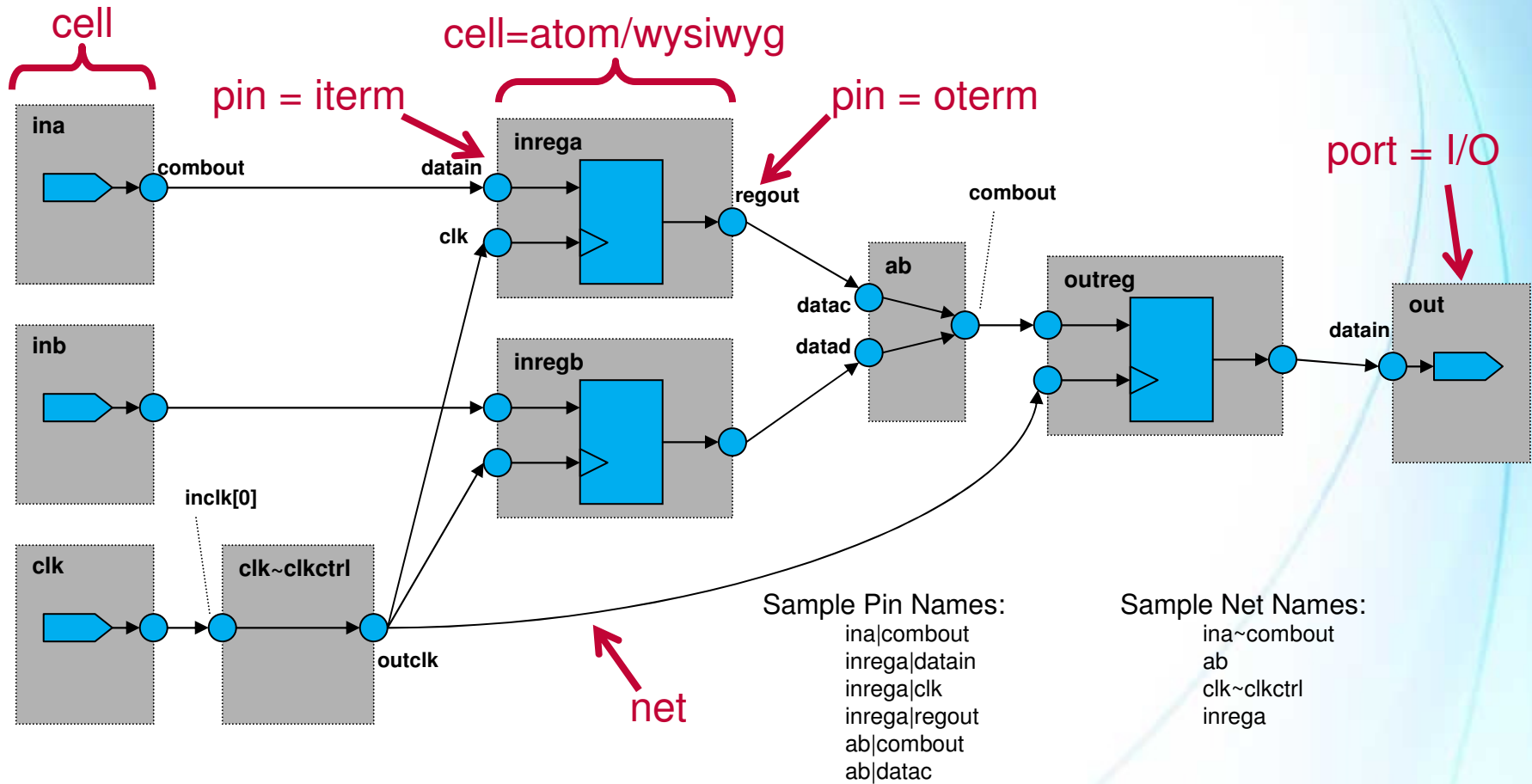


- Enter constraints via Constraints menu or Console pane

# SDC Netlist Terminology

Term	Definition
Cell	Device building blocks (e.g. look-up tables, registers, embedded multipliers, memory blocks, I/O elements, PLLs, etc.)
Pin	Input or outputs of cells
Net	Connections between pins
Port	Top-level inputs and outputs (e.g. device pins)

# SDC Netlist Example

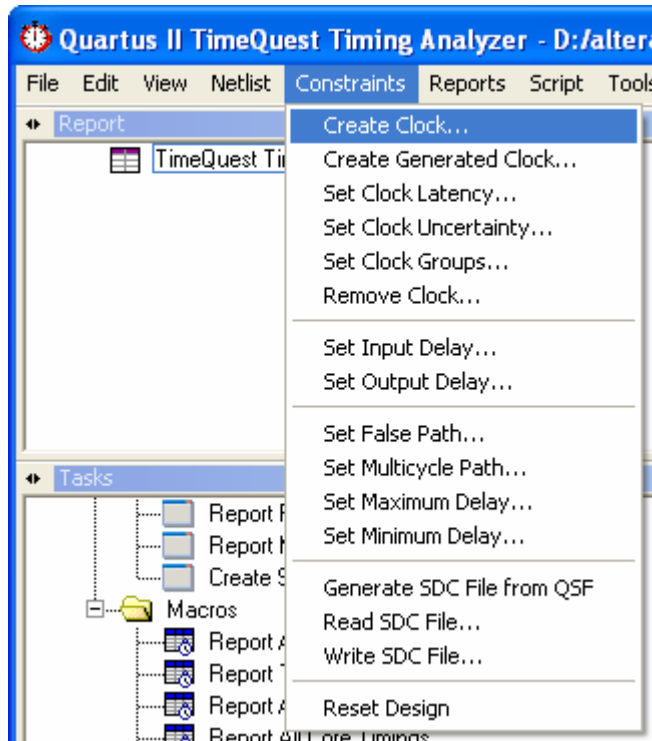


# Collections

- Searches and returns from the design netlist with a list of names meeting criteria
- Used in SDC commands
  - Some collections searched automatically during commands usage and may not need to be specified
- Examples
  - `get_ports`
  - `get_pins`
  - `get_clocks`
  - `all_clocks`
  - `all_registers`
  - `all_inputs`
  - `all_outputs`

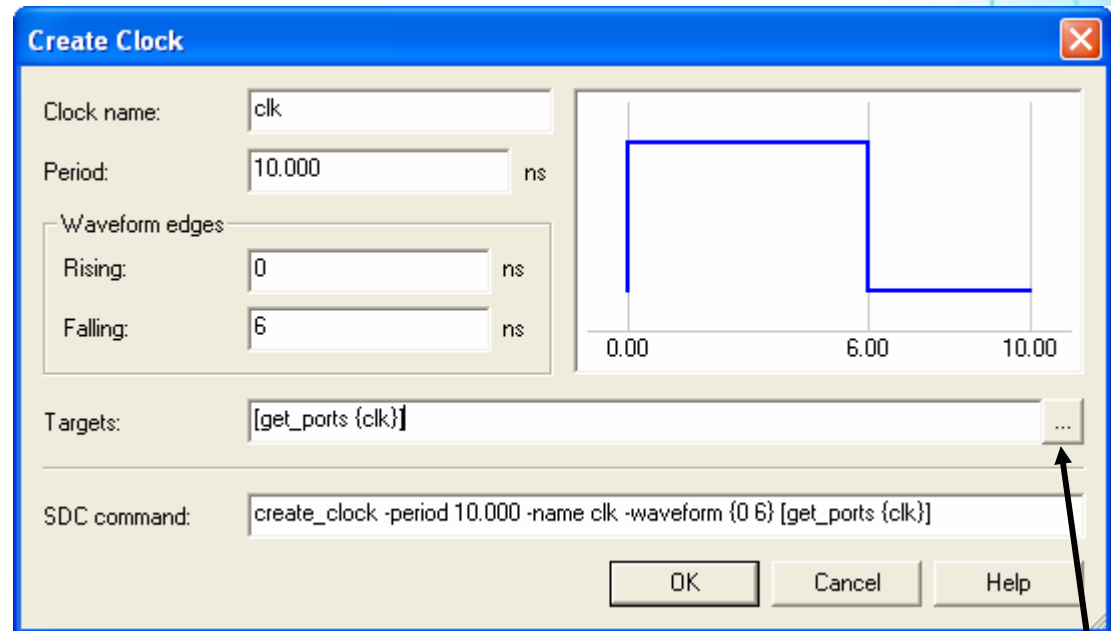
*See “TimeQuest Timing Analyzer” chapter of the Quartus II Software Handbook for a complete list & description of each*

# Create Clock



## Create Clock fields:

- **Clock Name** – Assign name to clock setting; defaults to target node name
- **Period** – Clock period in nanoseconds
- **Waveform edges** – Use for non-50% duty cycle clocks
- **Targets** – Port or pin to which clock setting is being applied

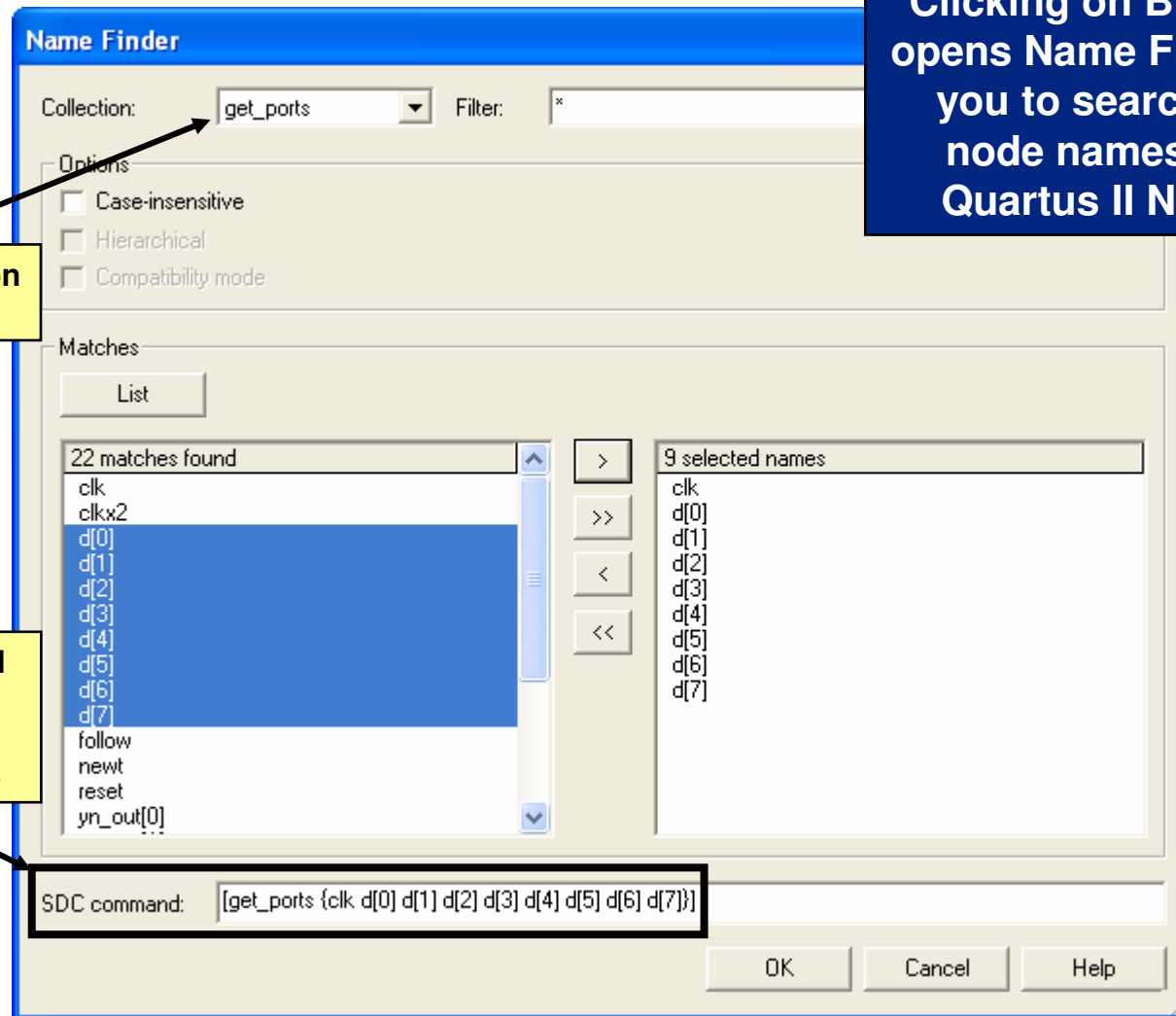


***Important Note:*** Unlike the Classic Timing Analyzer, all design clocks are related by default. This means TimeQuest will analyzer paths between clock domains whether you have specifically related them or not.

Name Finder (next slide)

***SDC: create\_clock***

# Name Finder





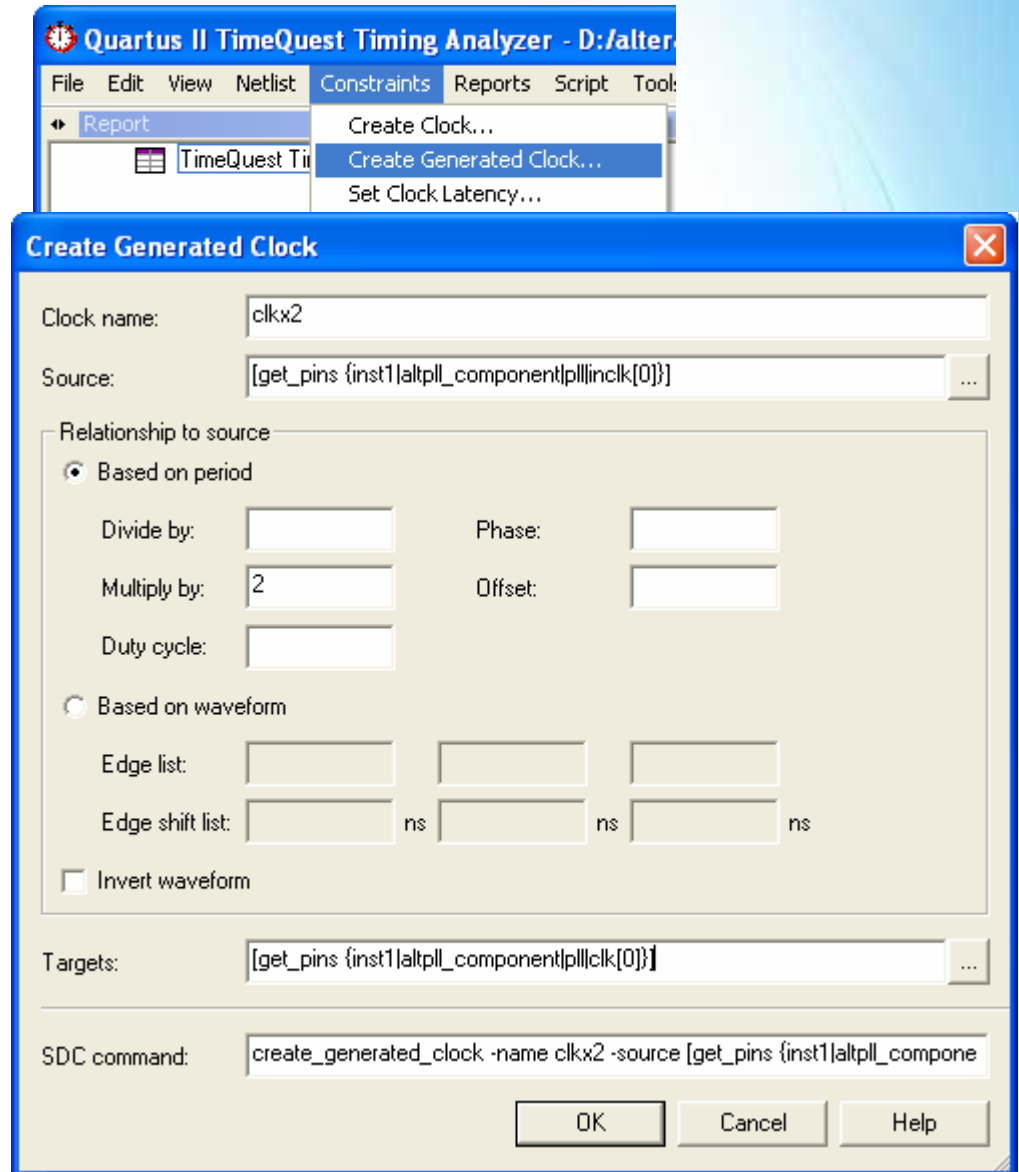
# Generated Clocks

- Generated clocks are clock signals derived from a previously created clock
  - E.g. clock dividers, ripple clocks, PLLs
  - Must be defined by a constraint

## Create Generated Clock fields:

- **Clock Name** – Assign name to clock setting
- **Relationship to source** – Specify how generated clock is related to base clock. The **Based on waveform** section allows for more complexity in the relationship to the base clock (not discussed)
- **Targets** – Port or pin to which clock setting is being applied

***SDC: create\_generated\_clock***

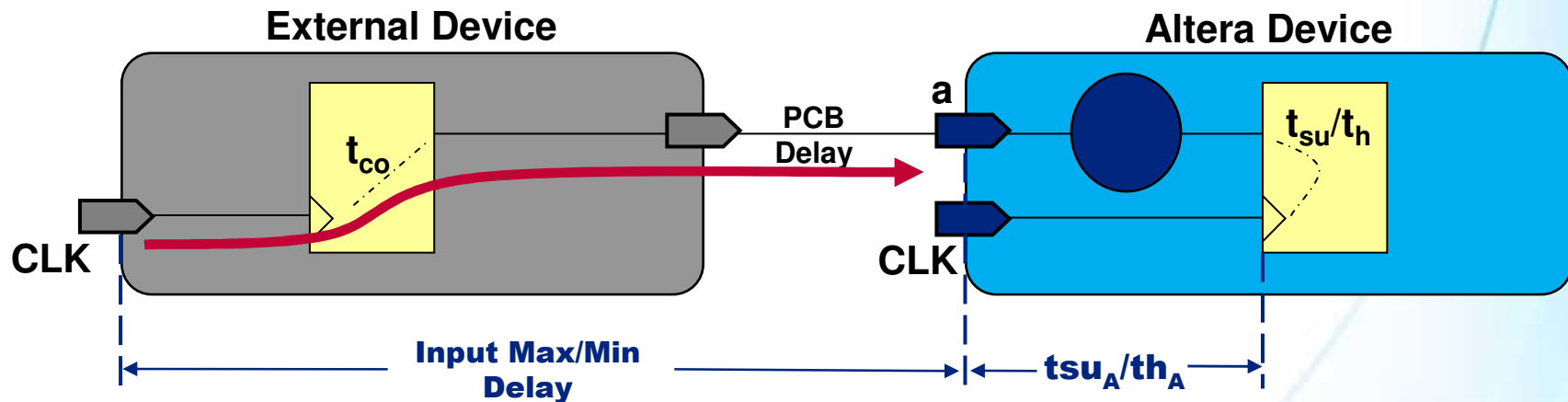


# I/O Constraining

- Specify system-level timing constraints
- Settings
  - Input maximum delay
  - Input minimum delay
  - Output maximum delay
  - Output minimum delay

# Input Maximum/Minimum Delay

- Maximum/minimum delay from external device to Altera I/O
  - Represents external device max & min  $t_{co}$  + PCB delay - PCB clock skew
- Constrains registered input path ( $t_{su}/t_h$ )

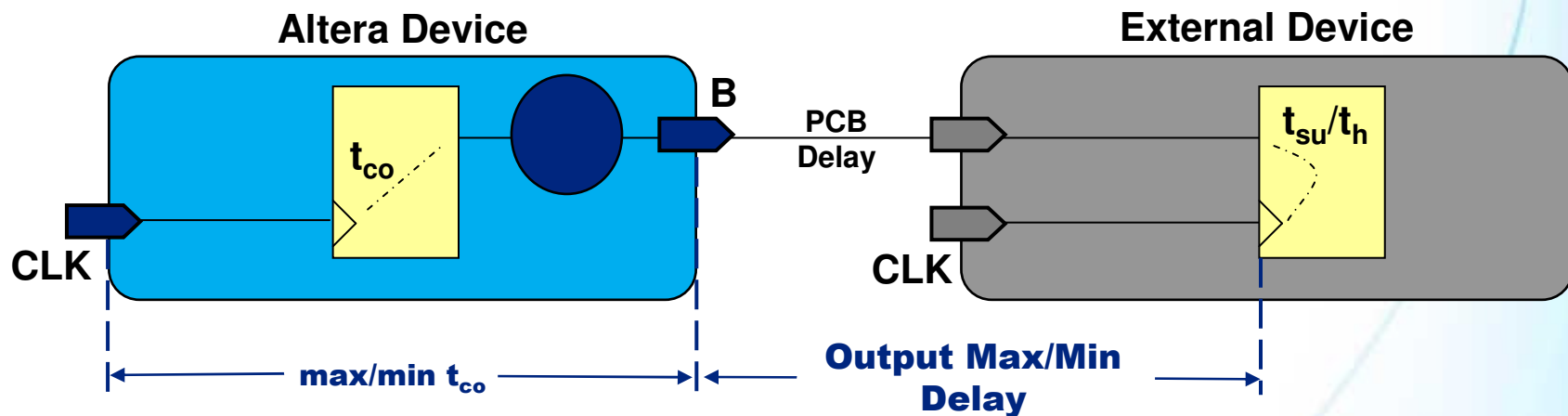


$$tsu_A \leq t_{CLK} - \text{Input Maximum Delay}$$

$$th_A \leq \text{Input Minimum Delay}$$

# Output Maximum/Minimum Delay

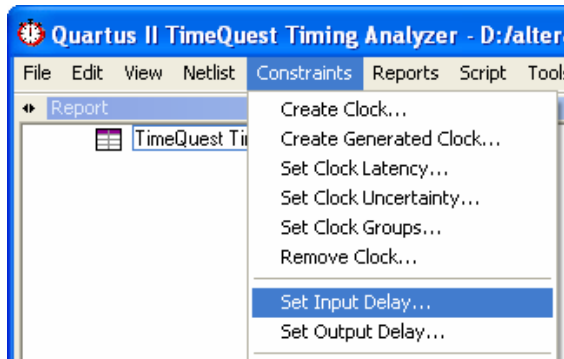
- Maximum/minimum delay from Altera I/O to external device
  - Represents external device  $t_{su}/t_h$  + PCB delay - PCB clock skew
- Constrains registered output path (max & min  $t_{co}$ )



$$t_{co_B} \leq t_{CLK} - \text{Output Maximum Delay}$$

$$t_{co_B} \geq \text{Output Minimum Delay}$$

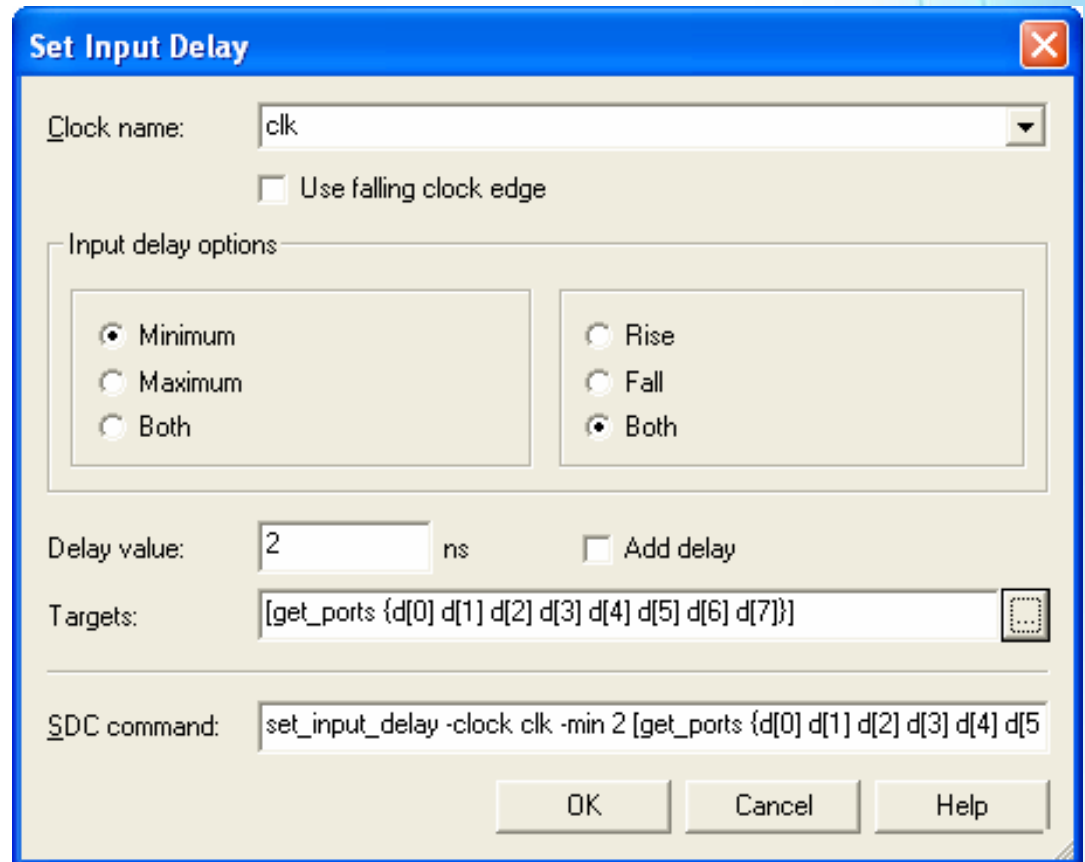
# Set Input/Output Delay



## Set Input/Output Delay fields:

- **Clock Name** – Specify source clock
- **Input delay options** – Choose max or min constraint. Rise/Fall indicate if the constraint applies particularly to a rising or falling edge transition (advanced).
- **Delay value** – Total off chip delay
- **Add delay** – Must use if applying multiple sets of input/output delays to the same port (e.g. input ports feeding multiple internal registers)
- **Targets** – Port to which setting is being applied

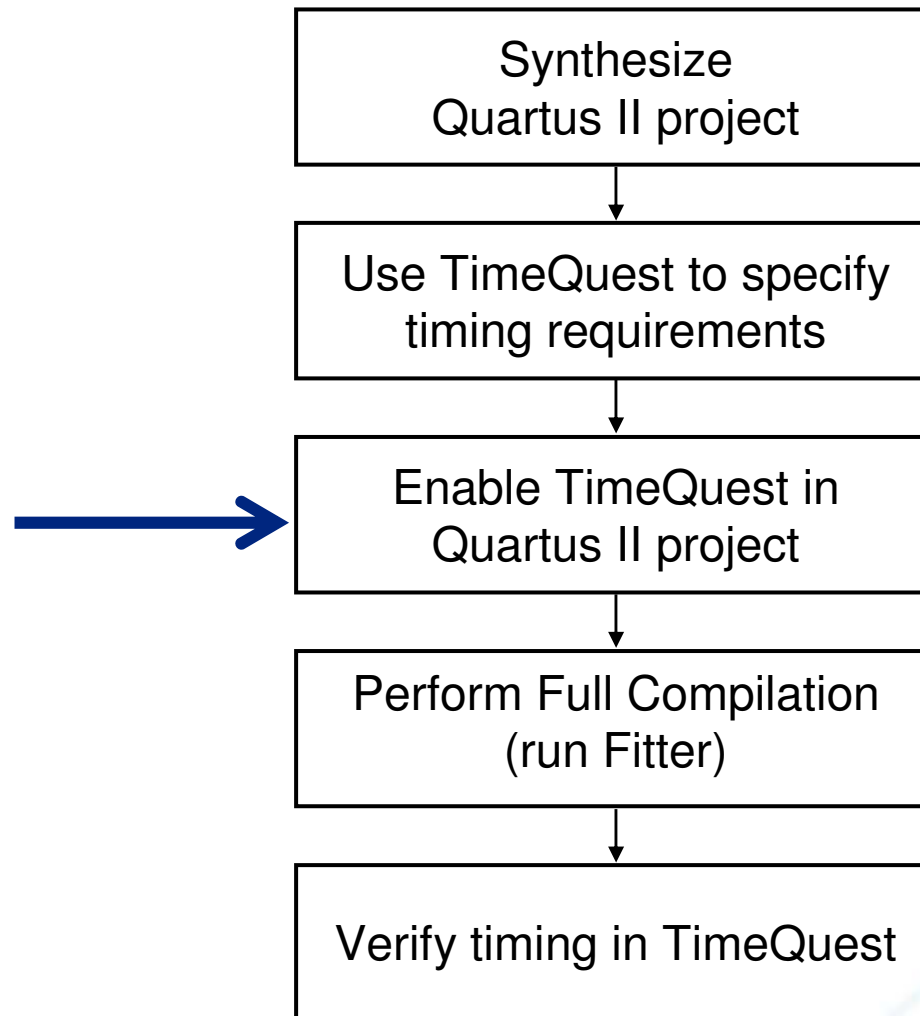
***SDC: set\_input\_delay***  
***SDC: set\_output\_delay***



# Useful Reports for Design Constraining

- Report Clocks
  - Use to ensure all clocks have been defined correctly
- Report Unconstrained Paths
  - Use to determine if any constraints are missing
- Report SDC
  - Use to review what constraints have currently been applied to the netlist
- Check Timing
  - Use to check issues with design or applied constraints
- Report Clock Transfers
  - Use to determines nets crossing clock domains
  - Remember, by default, all clock domains are related and analyzed with respect to one another
    - Paths between domains might need to be set as false paths
- Report Ignored Constraints
  - Use to determine if any constraints being ignored

# Using TimeQuest in Quartus II Flow

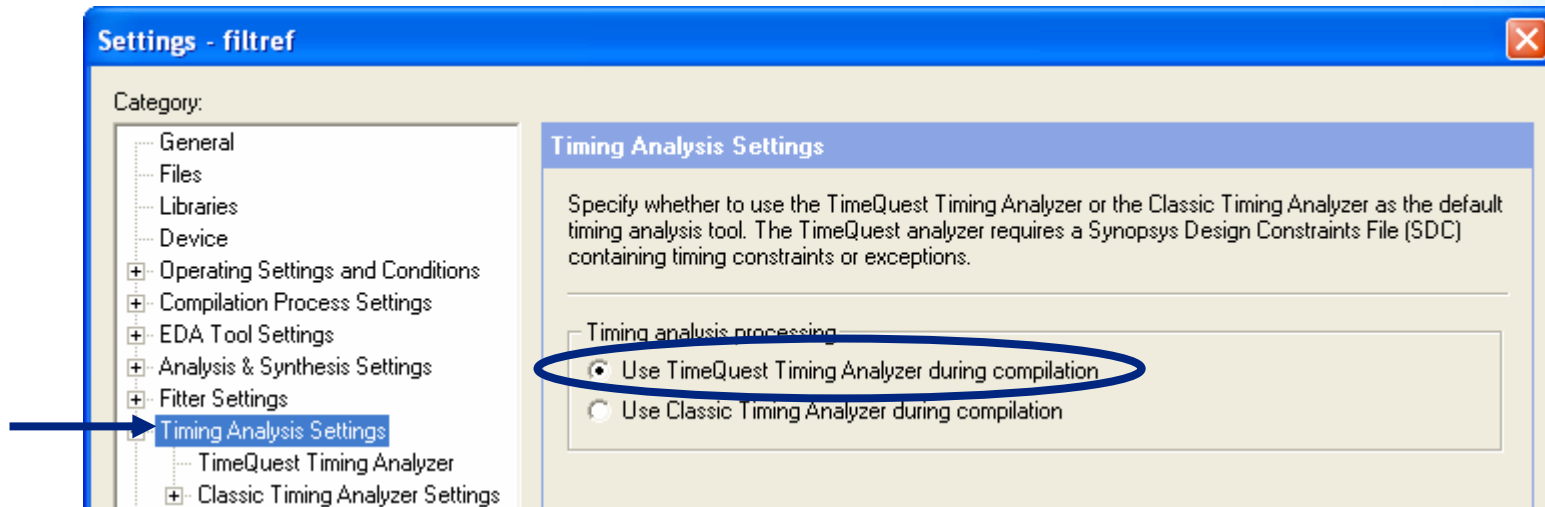


# Enabling TimeQuest in Quartus II

- Tells Quartus II to use SDC constraints during fitting
- File order precedence
  1. Any SDC files added to Quartus II project (in order)
  2. *<current\_revision>.SDC*



# Enabling TimeQuest in Quartus II Software



## Notes:

- Arria GX only supports Timequest.
- TimeQuest is enabled by default for new Stratix III and Cyclone III designs.

# Adding SDC File to Quartus II Project

- Add SDC files to TimeQuest Timing Analyzer page of Settings dialog box

Settings - filtref

Category:

- General
- Files
- Libraries
- Device
- Operating Settings and Conditions
- Compilation Process Settings
- EDA Tool Settings
- Analysis & Synthesis Settings
- Fitter Settings
- Timing Analysis Settings
  - TimeQuest Timing Analyzer**
  - Classic Timing Analyzer Settings
- Assembler
- Design Assistant
- SignalTap II Logic Analyzer
- Logic Analyzer Interface
- Simulator Settings

**TimeQuest Timing Analyzer**

Specify TimeQuest Timing Analyzer options.

SDC files to include in the project

SDC filename:  ...

File name	Type
filtref.sdc	Synopsys Desi...
top_solution.sdc	Synopsys Desi...

Buttons: Add, Remove, Up, Down

Enable Advanced I/O Timing

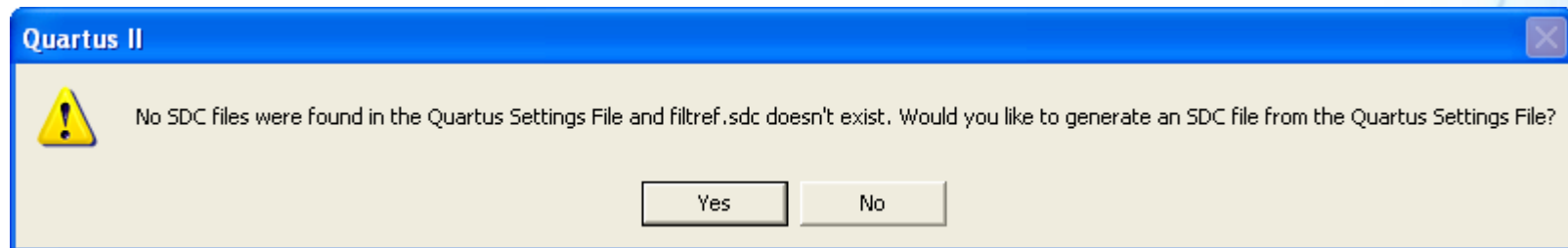
Enable multicorner timing analysis

Click Add to add SDC to list

Analyze fast and slow corners at the same time

# Quartus Settings File (QSF)

- SDC constraints are not stored in QSF
- TimeQuest uses script to convert QSF timing assignments to SDC
  - Constraints menu
  - Done automatically if no SDC file exists when first opening TimeQuest

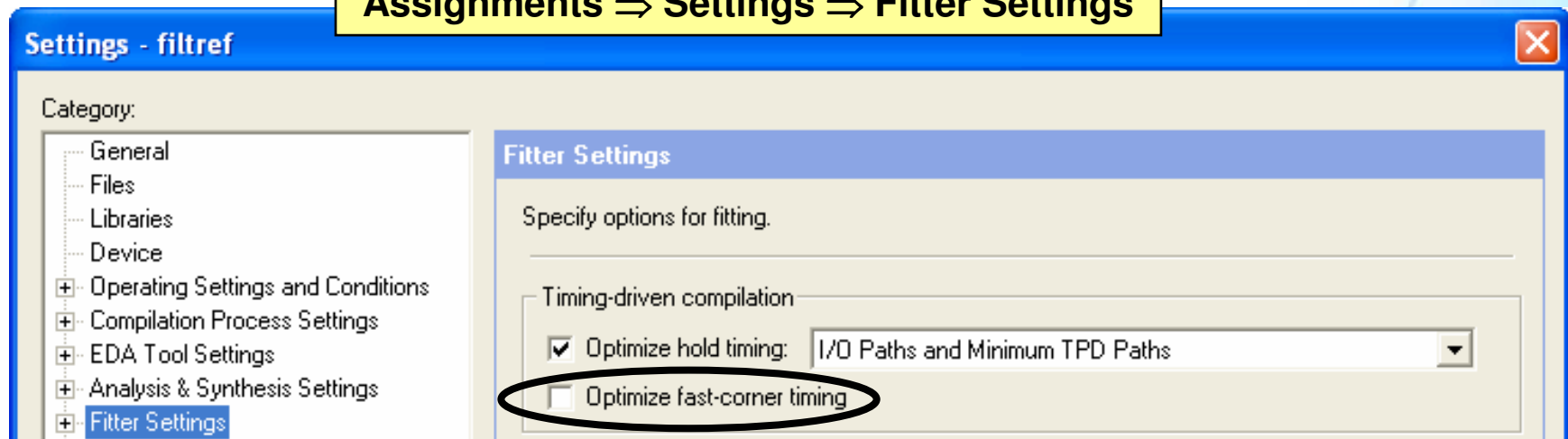


- See Quartus II Handbook Chapter, “*Switching to the TimeQuest Timing Analyzer*” for details
  - Differences between Classic Timing Analyzer and TimeQuest
  - Details on conversion utility

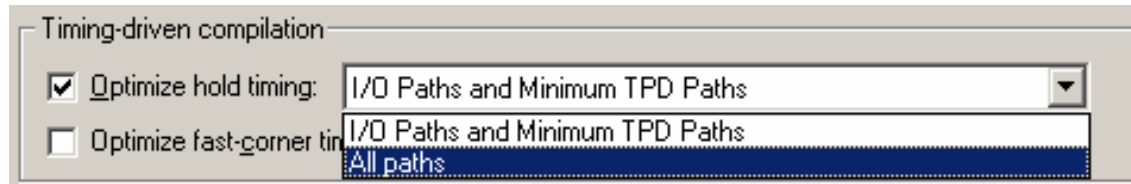
# Timing Driven Compilation (TDC)

- Directs fitter to place & route logic to meet timing assignments
  - Optimize timing (on by default)
    - Placing nodes in critical paths closer together
    - Located in “more settings” box
  - Optimize fast-corner timing
    - Optimizing for fast process (minimum timing models)
    - Can add up to 10% to compile time

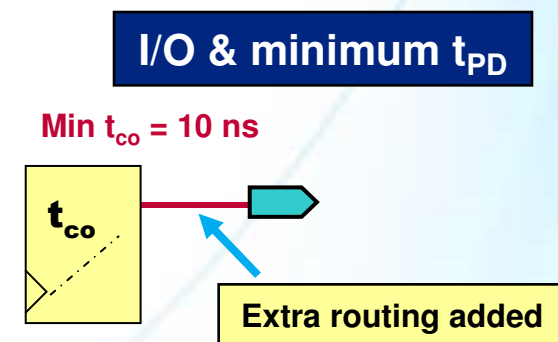
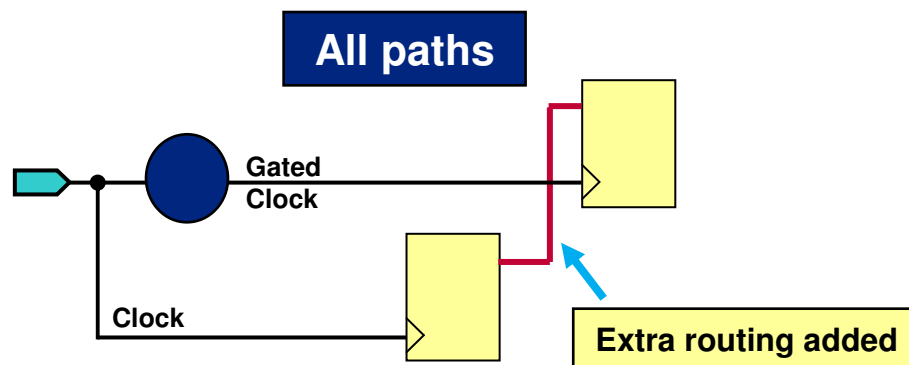
Assignments ⇒ Settings ⇒ Fitter Settings



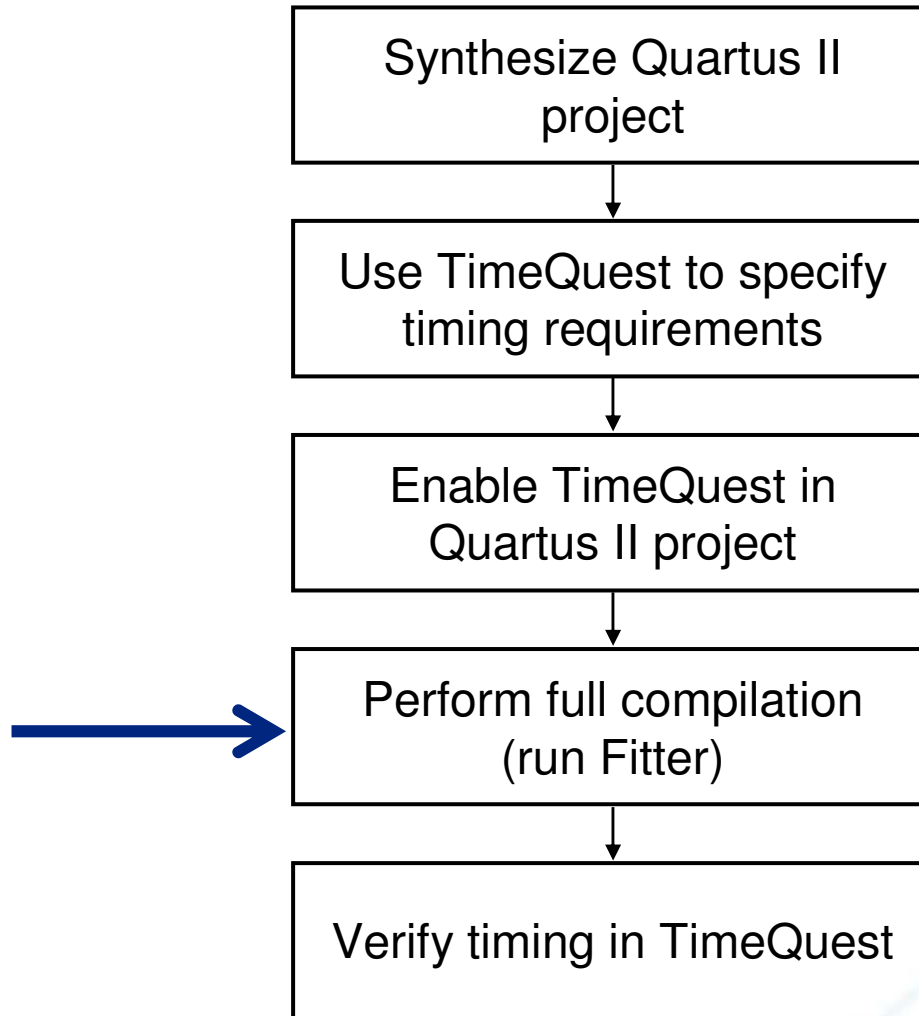
# Optimize Hold Timing



- Modifies place & route to meet hold or minimum timing requirements
  - May add additional routing in path
- Settings
  - Any I/O & minimum  $t_{pd}$  paths (default)
  - All paths (I/O & internal)



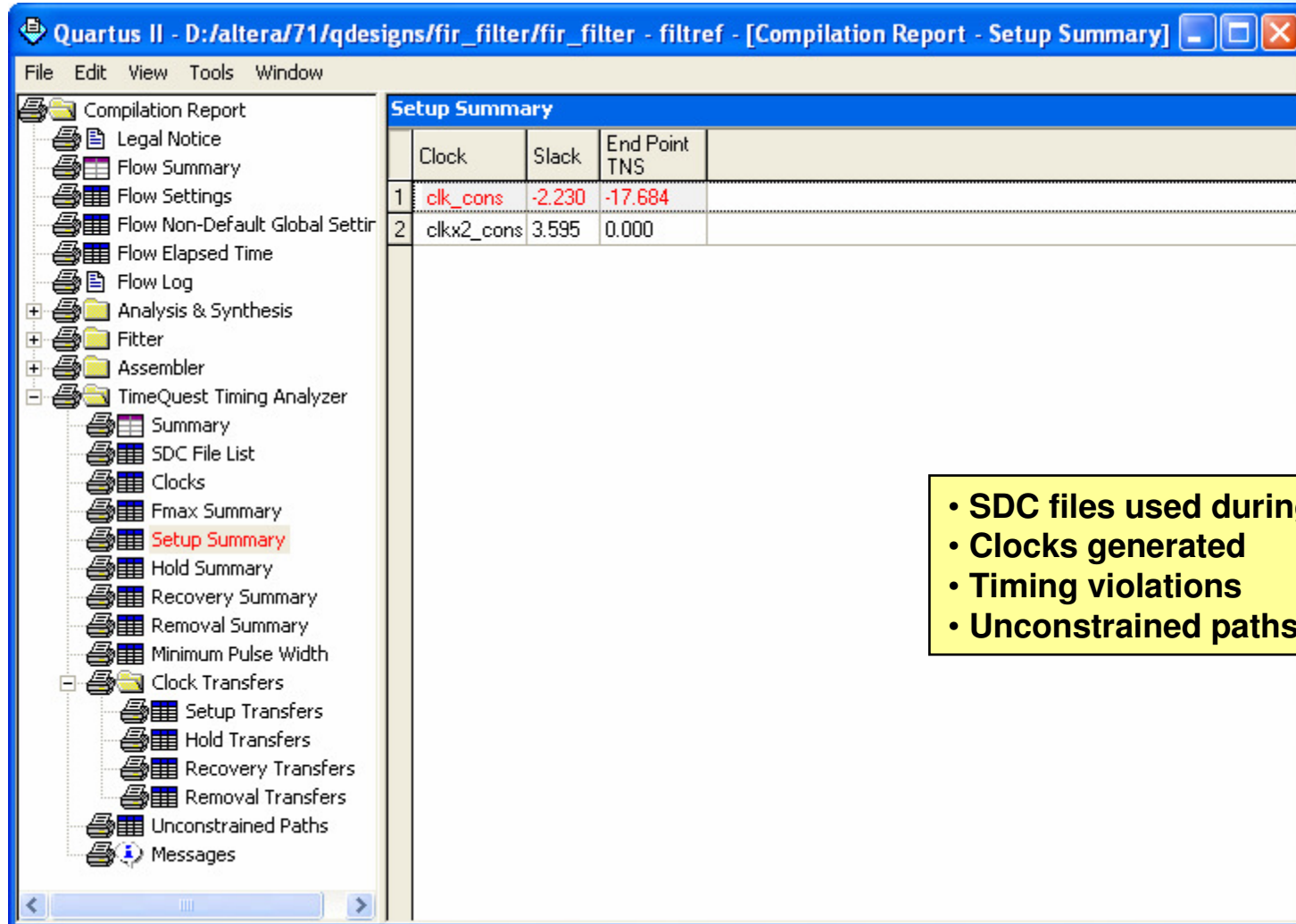
# Using TimeQuest in Quartus II Flow



# Verifying Timing Requirements

- View TimeQuest summary information directly in Quartus II Compilation Report
- Open TimeQuest for more thorough analysis
  - Follow TimeQuest flow
  - Run TimeQuest easy-to-use reporting capabilities (Tasks pane)
  - Place Tcl reporting commands into script file
    - Easy repetition

# TimeQuest Summary Reports



The screenshot shows the Quartus II interface with the TimeQuest Timing Analyzer. The left pane shows a tree view of reports, with 'Setup Summary' selected. The main pane displays a table with the following data:

	Clock	Slack	End Point TNS	
1	clk_cons	-2.230	-17.684	
2	clkx2_cons	3.595	0.000	

- SDC files used during fitting
- Clocks generated
- Timing violations
- Unconstrained paths



# Generating Detailed Reports

The image shows the Quartus II TimeQuest Timing Analyzer interface. On the left, the Reports menu is open, listing various report types. A blue callout box points to the 'Report Timing...' option in the menu. Below the menu, a list of reports is shown, with 'Report Timing...' circled in blue. A blue callout box points to this circled option. At the bottom left, a white box contains the text 'Tcl: report\_timing'. On the right, the 'Report Timing' dialog box is open. A blue callout box points to the 'Detail level' dropdown menu, which is set to 'Path Only'. Another blue callout box points to the 'Report panel name' and 'File name' options, which are set to 'Summary' and 'Full Path' respectively. A third blue callout box points to the 'Open' button at the bottom of the dialog. A large blue arrow points from the 'Report Timing...' menu option to the dialog box.

**Choose Report Timing (Reports menu) or double-click on Report Timing (Tasks pane)**

**Select level of detail**

**Select where to send output report**

**Tcl: report\_timing**

# 3<sup>rd</sup>-Party Timing Analysis Tool Support

- Synopsys
  - PrimeTime
- Mentor Graphics
  - TAU

## *Exercise 6 Demonstration*

© 2007 Altera Corporation—**Confidential**

Altera, Stratix, Arria, Cyclone, MAX, HardCopy, Nios, Quartus, and MegaCore are trademarks of Altera Corporation

235

The Altera logo is located in the bottom right corner of the slide. It consists of the word "ALTERA" in a bold, blue, sans-serif font. The letters are outlined, giving it a three-dimensional appearance. The background of the slide features a light blue gradient with several curved, glowing lines that sweep across the right side.

# Summary

- TimeQuest provides an easy-to-use tool to verify timing
  - Entering timing constraints
  - Run various timing reports

# TimeQuest Support Resources

- Quartus II Handbook Chapters
  - “The TimeQuest Timing Analyzer” (Volume 3)
  - “Switching to the TimeQuest Timing Analyzer” (Volume 3)
- Training & Demonstrations
  - “Validating Performance with the TimeQuest Static Timing Analyzer” (online recording)

**ALTERA**

# Quartus II Software Design Series: Foundation

*EDA Simulation*



# Simulation

- 3<sup>rd</sup>-party EDA tool simulation
  - RTL (functional)
  - Post-synthesis (functional)
    - Optional
  - Gate-level (timing)
  
- Quartus II simulation covered in Appendix

# 3<sup>rd</sup>-Party Simulation Support

- Mentor Graphics
    - ModelSim
  - Cadence
    - NCSim
  - Synopsys
    - VCS/MX
  - Aldec
    - Active-HDL
- 
- All support NativeLink tool flow



# RTL Simulation Files (VHDL)

- Design files
- RTL (functional) models
  - LPM megafunction models
    - 220model.vhd & 220\_pack.vhd
    - Compile into lpm library
  - Altera-specific megafunction models
    - altera\_mf.vhd & altera\_mf\_components.vhd
    - Compile into altera\_mf library
  - Altera primitive models (LCELL, OPNDRN, etc.)
    - altera\_primitives.vhd & altera\_primitives\_components.vhd
    - Compile into altera library
- HEX files
  - Memory initialization

➤ All model files located in “eda\sim\_lib” directory in Quartus II installation path  
➤ Pre-compiled in ModelSim-Altera

# RTL Simulation Files (Verilog)

- Design Files
- RTL (functional) Models
  - LPM megafunction models
    - 220model.v
    - Compile into `lpm` library
  - Altera-specific megafunction models
    - `altera_mf.v`
    - Compile into `altera_mf` library
  - Altera primitive models (LCELL, OPNDRN, etc.)
    - `altera_primitives.v`
    - Compile into `altera` library
- HEX files
  - Memory initialization

➤ All model files located in “eda\sim\_lib” directory in Quartus II installation path  
➤ Pre-compiled in ModelSim-Altera

# Megafunction Exceptions

- RTL simulation requires using device-specific simulation models (similar to gate-level)
  - altclkbuf
  - alkclkctrl
  - DDR megafunctions (not IP)
  - MAX II UFM megafunctions
  - altmemmult
  - altremote\_update
- Stratix II GX and Stratix GX designs require additional libraries for RTL simulation
  - See Quartus II Handbook, Volume 3, Section 1: “Simulation”

# MegaWizard Simulation Libraries

MegaWizard Plug-In Manager - RAM: 2-PORT [page 11 of 12] -- EDA

**RAM: 2-PORT**  
Version 7.1

Parameter Settings | **EDA** | Summary

my\_ram

data[7..0]  
wraddress[4..0]  
wren  
rdaddress[4..0]  
clock

32 Word(s) RAM

q[7..0]

Block Type: AUTO

Resource Usage

256 ram_bits (AUTO)
---------------------

Simulation Libraries

To properly simulate the generated design files, the following simulation model file(s) are needed

File	Description
altera_mf	Altera megafunction simulation library

**EDA page of MegaWizard specifies which model libraries are needed for simulation**

Synthesis

Some third-party synthesis tools may benefit from reading the details of this megafunction. Using this netlist, the synthesis tool is able to estimate timing and resource usage for the megafunction

Generate a netlist for synthesis area and timing estimation

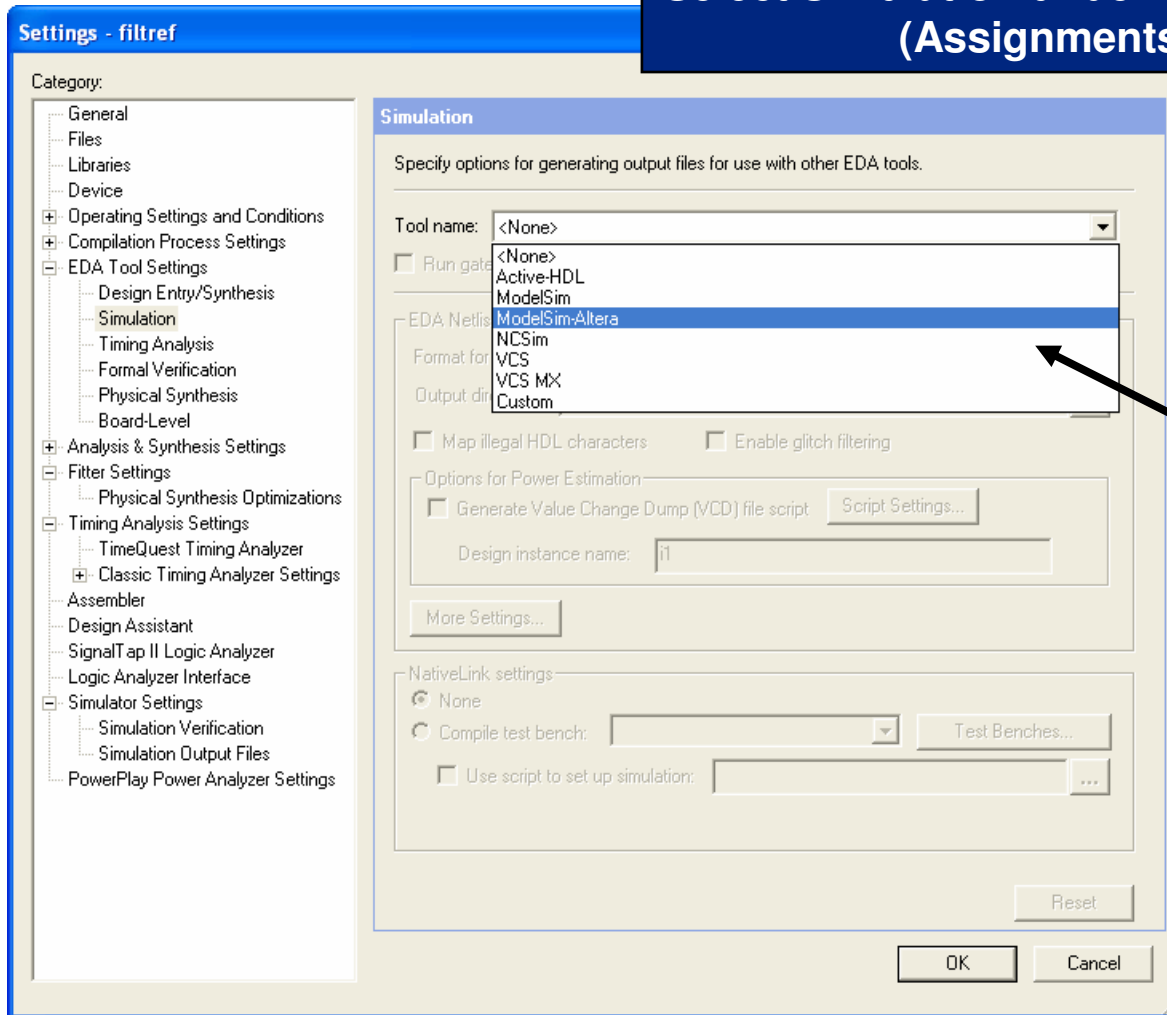
Cancel < Back Next > Finish

# Post-Processing Simulations

- Two types
  - Post-synthesis
  - Gate-level (timing)
- Quartus II software must generate output simulation netlist files for specific simulator
  - EDA Netlist Writer

# Specify Simulator

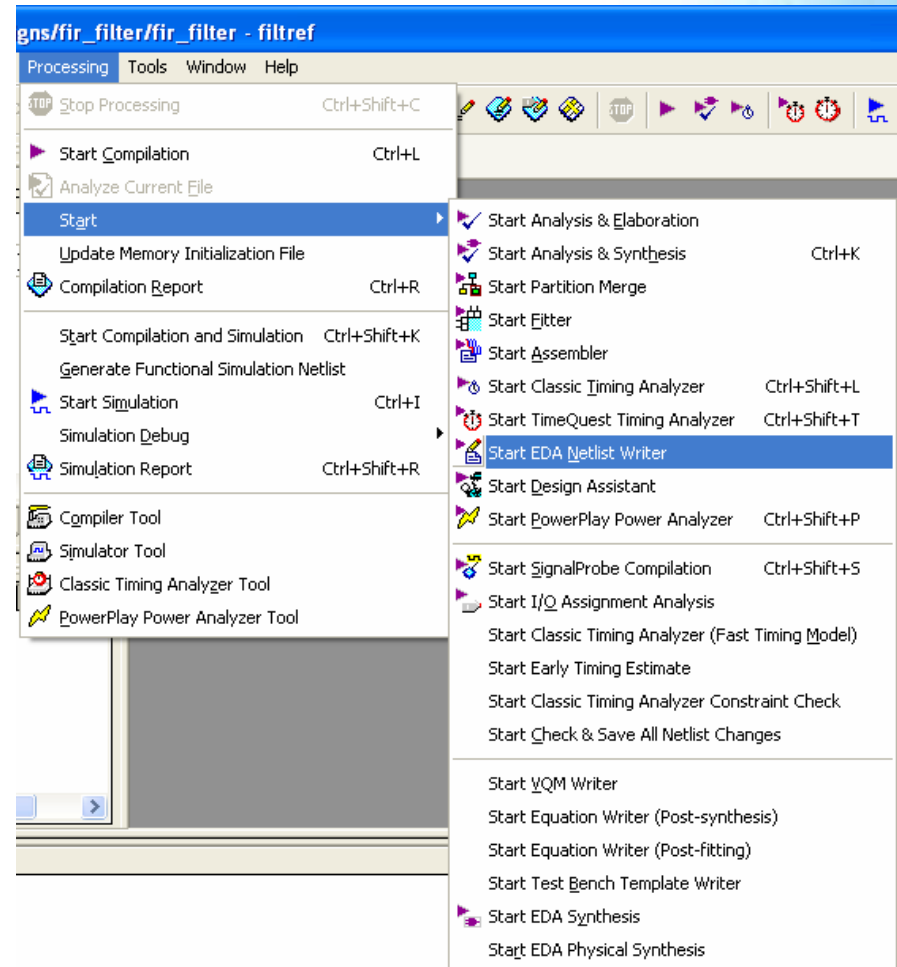
Select Simulation under EDA Tools Settings  
(Assignments menu)



Select simulation  
tool

# Generating 3<sup>rd</sup>-Party Netlists

- Full compilation
- Execute process individually
  - Processing menu ⇒ Start ⇒ Start EDA Netlist Writer
  - Generates files without full compilation
- Scripting



# Specifying Simulation Options

**Output netlist language**

**Output directory**

**Generate VCD for power analysis**

**NativeLink settings (discussed later)**



# Gate-Level Simulation Files (VHDL)

- All model files located in “eda\sim\_lib” directory in Quartus II installation path
- Pre-compiled in ModelSim-Altera

- Use Quartus II-generated files
  - VHDL output file (.VHO)
  - Standard delay format output file (.SDO)
  - Located in “*simulation*\<*simulator\_tool*>” subdirectory of project (default)
- Device-specific simulation models
  - Use <device\_name>\_atoms.vhd & <device\_name>\_atoms\_components.vhd
  - Compile into <device\_name> library
- HEX files for memory initialization

# Gate-Level Simulation Files (Verilog)

- All model files located in “eda\sim\_lib” directory in Quartus II installation path
- Pre-compiled in ModelSim-Altera

- Use Quartus II-generated files
  - Verilog output file (.VO)
  - Standard delay format output file (.SDO)
  - Located in “*simulation*\<*simulator\_tool*>” subdirectory of project (default)
- Device-specific simulation models
  - <device\_name>\_atoms.vo
  - Compile into <device\_name> library
- HEX files for memory initialization

# Device Library Names

- Use these library names for compiling device-specific simulation models
  - arriagx
  - arriagx\_hssi
  - stratixiii
  - stratixii
  - stratixiigx
  - stratixiigx\_hssi
  - stratix
  - stratixgx
  - stratixgx\_gxb
  - cycloneiii
  - cycloneii
  - cyclone
  - maxii

➤ For older device families, see Quartus II Handbook, Volume 3, Section 1: “Simulation”

# Post-Synthesis Simulation

- Enable functional simulation netlist
  - No SDO is generated
- Synthesize design
- Run EDA Netlist Writer
- Use gate-level (device-specific) simulations models

# Enabling Functional Simulation Netlist

Settings - filtref

Category:

- General
- Files
- Libraries
- Device
- Operating Settings and Conditions
- Compilation Process Settings
- EDA Tool Settings
  - Design Entry/Synthesis
  - Simulation
  - Timing Analysis
  - Formal Verification
  - Physical Synthesis
  - Board-Level
- Analysis & Synthesis Settings
- Fitter Settings
  - Physical Synthesis Optimizations
- Timing Analysis Settings
  - TimeQuest Timing Analyzer
  - Classic Timing Analyzer Settings
- Assembler
- Design Assistant
- SignalTap II Logic Analyzer

Simulation

Specify options for generating simulation netlist:

Tool name: ModelSim-Altera

Run gate-level simulation

EDA Netlist Writer options:

Format for output netlist: Verilog

Output directory: simulation

Map illegal HDL characters

Options for Power Estimation:

Generate Value Change Report

Design instance name: sim\_1

More Settings...

More EDA Tools Simulation Settings

Specify the settings for the EDA third party simulation options in your project.

Option

Name: Generate netlist for functional simulation only

Setting: On

Description:  
Generate Verilog or VHDL netlist for functional simulation with EDA simulation tools. The SDF Timing file (.sdf) is not generated for the project. This option is not available for the VCS MX simulation tool.

Reset

Reset All

Existing option settings:

Name:	Setting:
Architecture name in VHDL output ne...	structure
Bring out device-wide set/reset signal...	Off
Disable setup and hold time violations...	Off
Do not write top level VHDL entity	Off
Flatten bases into individual nodes	Off
Generate netlist for functional simulati...	On
Maintain hierarchy	Off
Truncate long hierarchy paths	Off

OK

Cancel

Click on "More Settings" button in Simulation category

# Additional Simulation Library

## ■ sgate library

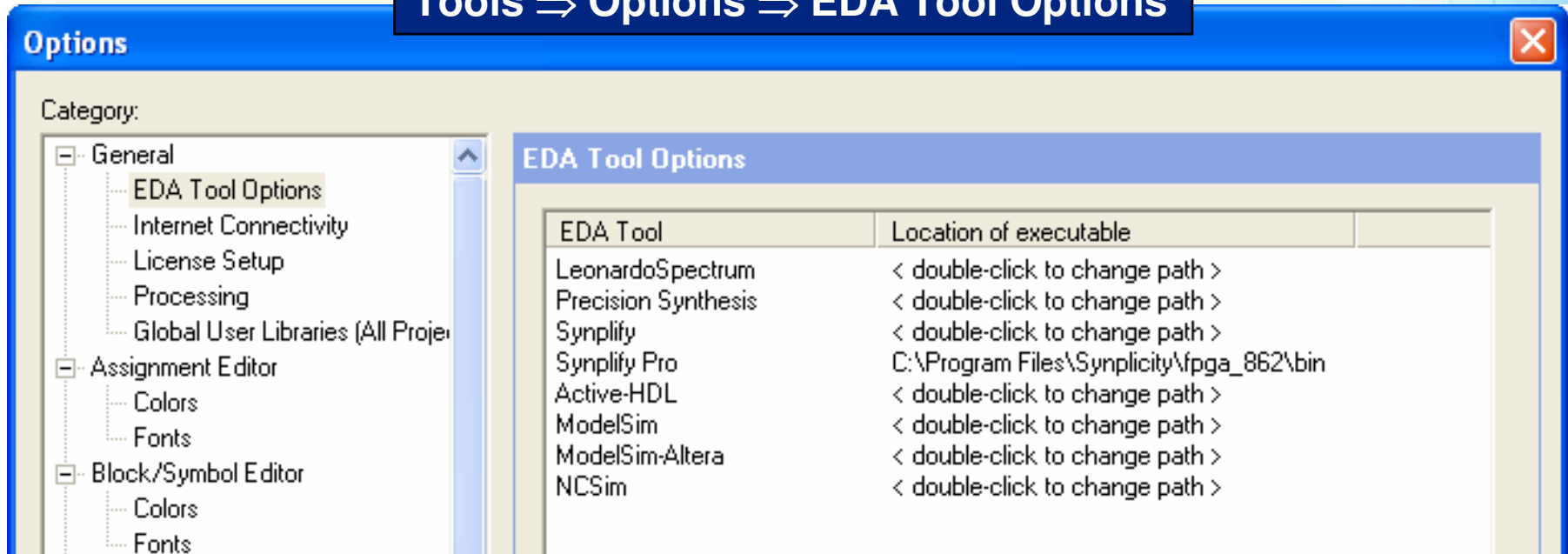
- Needed for functional simulation of IP
- Needed for any GX device simulation
- VHDL
  - sgate.vhd & sgate\_pack.vhd
- Verilog
  - sgate.v

# Using NativeLink for Simulation

- Specify path to simulation tool
- Enable NativeLink settings

# Specify Path to Simulator

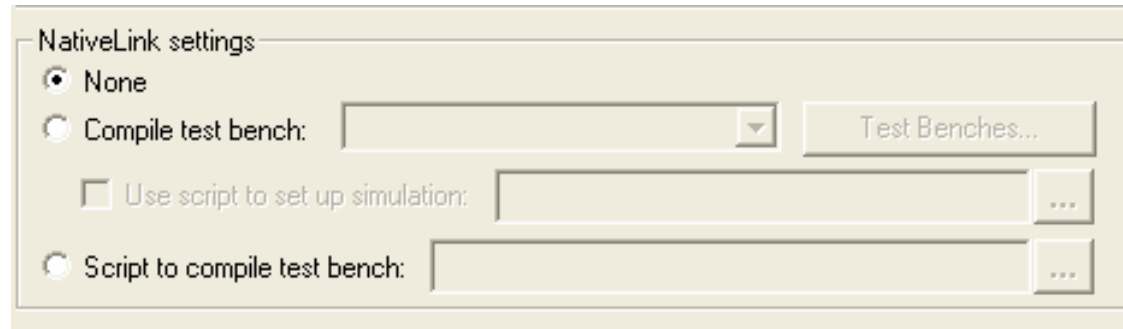
Tools ⇒ Options ⇒ EDA Tool Options



**Double-click to specify path to simulation tool executable**



# Enable NativeLink Settings



- None
  - NativeLink compiles simulation models & design files
- Compile test bench
  - NativeLink compiles all files (including test bench) and starts simulation
- Use script to compile test bench
  - NativeLink compiles simulation models & design files
  - User specifies script to compile test bench and start simulator

# Setting Up Test Benches

**Test Benches**

Specify settings for each test bench.

Existing test bench settings:

Name	Top level module	Design Instance	Run for	Test bench file(s)
testbench1	pipemult_tb	pipemult_u1	2us	pipemult_tb.vhd
testbench2	pipemult_tb	pipemult_u1	1us	pipemult_tb_new.vhd

**Create test bench settings for each test bench to be simulated**

OK

**New Test Bench Settings**

Create new test bench settings.

Test bench name:

Top level module in test bench:

Design instance name in test bench:

Simulation period

Run simulation until all vector stimuli are used

End simulation at:

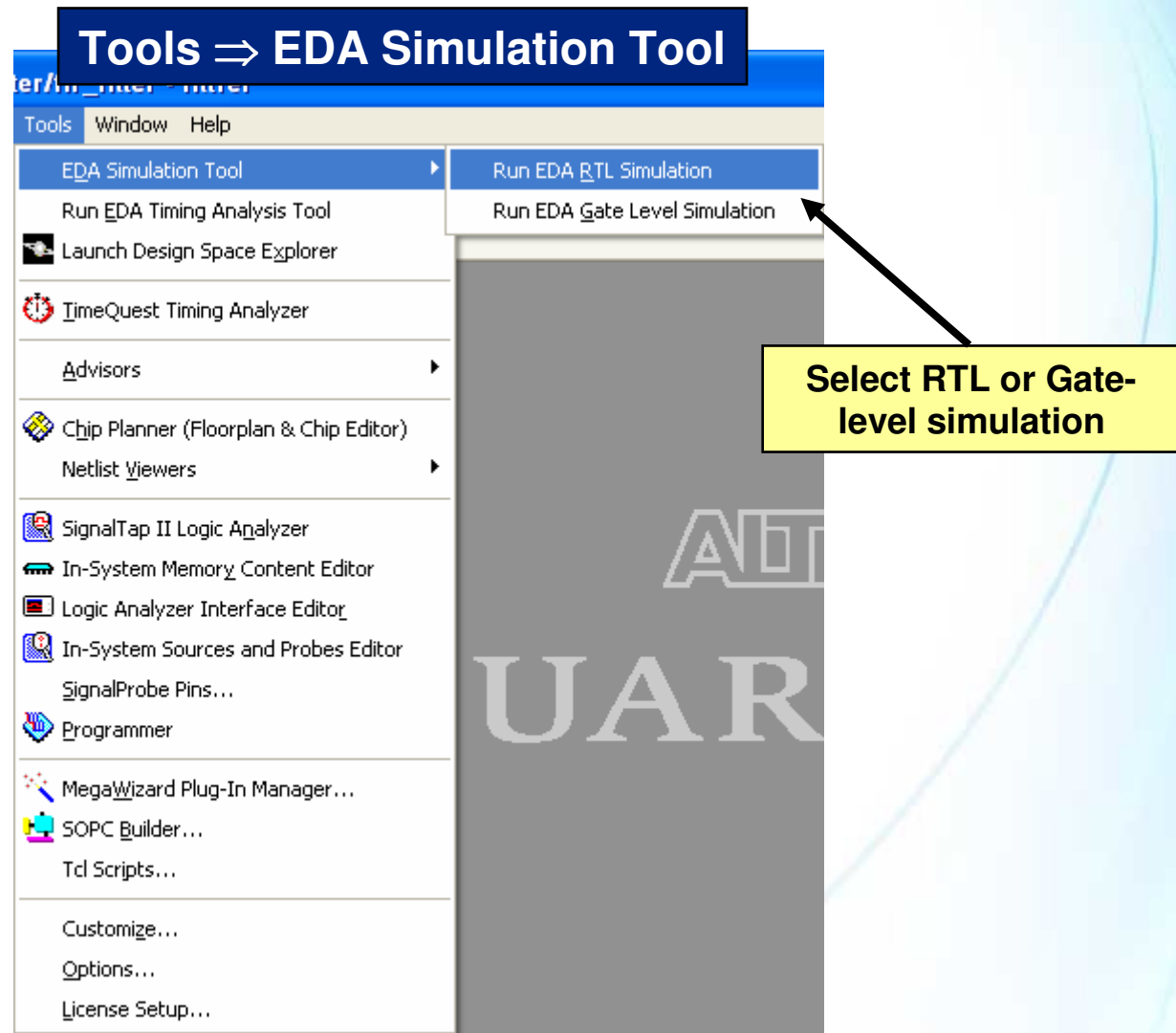
Test bench files

File name:  ...

File name	Library	HDL Version
pipemult_tb.vhd		Default

OK Cancel

# Running NativeLink Simulation



# Simulation Summary

- Simulating with 3<sup>rd</sup>-party EDA tools

# Simulation Support Resources

- Quartus II Handbook chapters (Volume 3)
  - “Mentor Graphics ModelSim Support”
  - “Synopsys VCS Support”
  - “Cadence NC-Sim Support”
  - “Simulating Altera IP in Third-Party Simulator Tools”

**ALTERA**

# Quartus II Software Design Series: Foundation

*Programming/Configuration*



# Programming/Configuration

- Setting device options
- Assembler module
- Programmer & chain description file
  - Programming directly with the Quartus II Programmer
- File conversion
  - Creating multi-device programming files

# Setting Device Options

- Assignments ⇒ Device ⇒ Device & Pin Options

Settings - filtref

Category:

- General
- Files
- Libraries
- Device
- Operating Settings and Conditions
- Compilation Process Settings
- EDA Tool Settings
  - Design Entry/Synthesis
  - Simulation
  - Timing Analysis
  - Formal Verification
  - Physical Synthesis
  - Board-Level

**Device**

Select the family and device you want to target for compilation

Family: Stratix II

**Device and Pin Options...**

Target device

- Auto device selected by the Fitter
- Specific device selected in 'Available devices' list
- Other: n/a

Package: Any

Pin count: Any

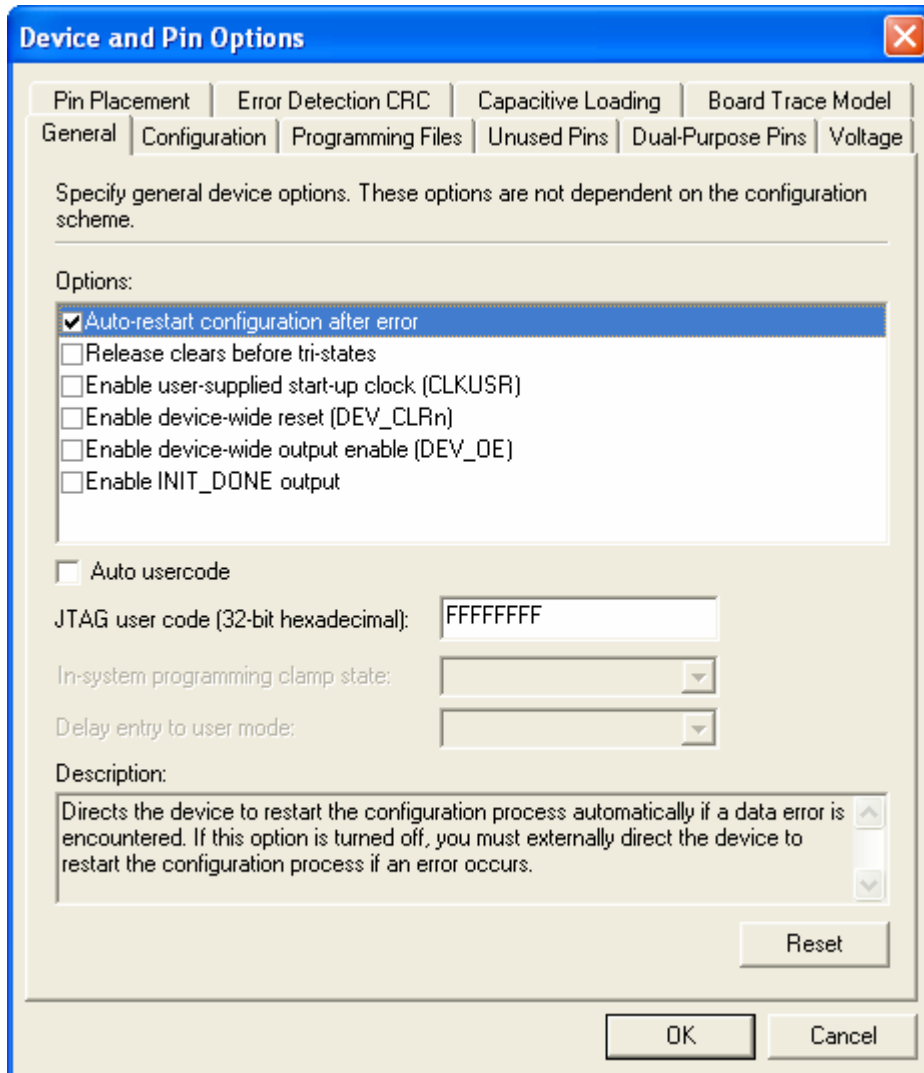
Speed grade: Any

- Show advanced devices
- HardCopy compatible only

**Device options control configuration & initialization of device**

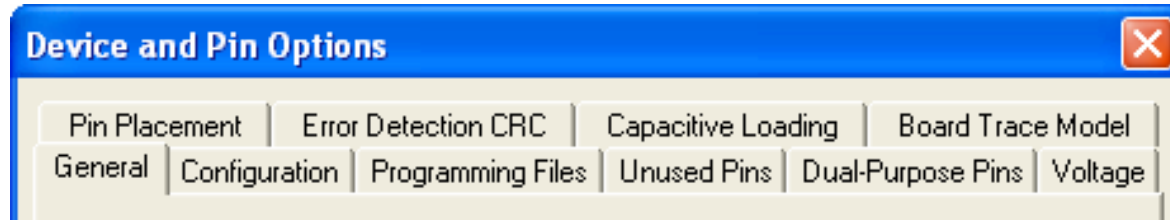


# General Tab



- Device options not dependent on configuration scheme (off by default)
  - Enable device-wide clear
  - Enable device-wide output enable
  - Enable initialization done output pin

# Other Device & Pin Option Tabs

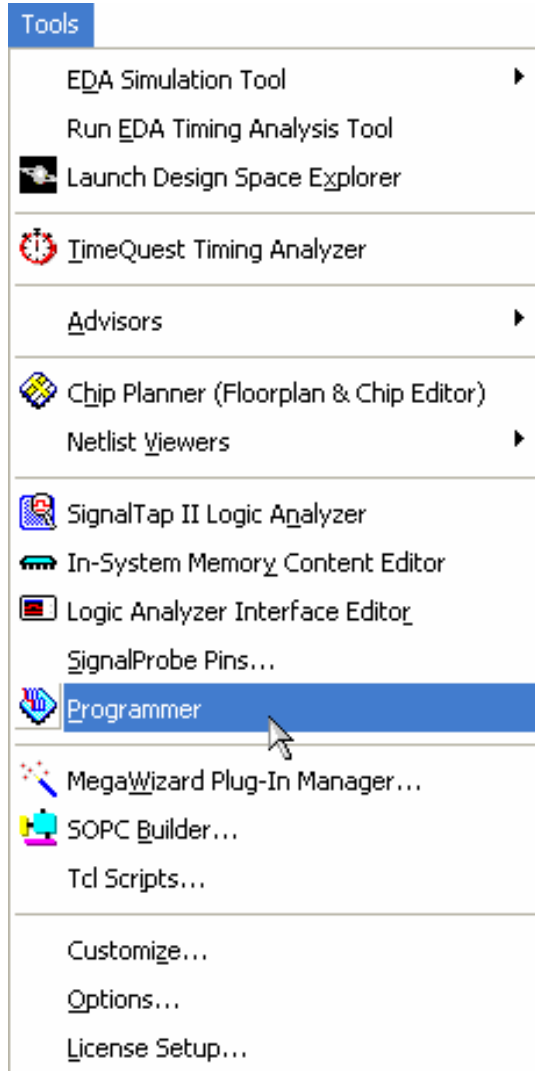


- **Configuration**
  - Generates correct configuration & programming files every compilation
  - Enables special features of configuration devices
    - Enable programming file compression
    - Set configuration clock frequency
- **Programming Files**
  - Output files always created
    - POF (programming object file)
    - SOF (SRAM object file)
  - Other selectable output files
    - Jam (jedec stapl)
    - JBC (JAM byte-code)
    - RBF (raw binary file)
    - HEXOUT (intel hex format)
- **Unused pins**
  - Indicates state of all unused I/O pins after configuration is complete
- **Dual-purpose pins**
  - Selects usage of dual-purpose pins after configuration is complete
- **Error detection CRC**
  - Enables internal CRC circuitry & frequency
- **Capacitive Loading**
  - Sets default capacitive loading for each I/O standard
- **Board Trace Model**
  - Sets default board trace model characteristics for each I/O standard (Stratix II & III devices only)

# Quartus II Assembler Module

- Generates all configuration/programming files
  - As selected in device & pin options dialog box
- Ways to run assembler
  - Full compilation
  - Execute assembler individually
    - Processing menu ⇒ Start ⇒ Start Assembler
    - Generates files without full compilation
      - Switching configuration devices
      - Enabling/disabling configuration device feature
  - Scripting

# Open Programmer



- Enables device programming
  - USB-Blaster™ cable
  - ByteBlaster™ II or ByteBlasterMV™ cables
  - Masterblaster™ cable
  - APU (Altera programming unit)
- Opens chain description file (.CDF)
  - Stores device programming chain information

# CDF File

- Lists devices & files for programming or configuration
- Programs/configures in top-to-bottom order

Quartus II - C:/altera/70/qdesigns/fir\_filter/fir\_filter - filtref - [filtref.cdf\*]

File Edit Processing Tools Window

Hardware Setup... No Hardware Mode: JTAG Progress: 0%

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

File	Device	Checksum	Usercode	Program/Configure	Verify	Blank-Check	Examine	Security Bit	Erase	ISP CLAMP
filtref.sof	EP1C6F256	000A1428	FFFFFFFF	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
[-] filtref.pof	EPCS4	075B6DC1	00000000	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
[-] Page_0				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
.../.../61/qdesigns...	EPC16	1C661491	FFFFFFFF	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
.../.../61/qdesigns...	EP2S60F672C5ES	00A99868	FFFFFFFF	<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**

# Example CDF Files

Single device chain

The screenshot shows the Quartus II Hardware Setup window for a single device chain. The hardware is identified as USB-Blaster [USB-0] in JTAG mode. The progress bar shows 0%. A table lists the files to be programmed:

File	Device	Checksum	Usercode	Program/Configure	Verify	Blank-Check	Examine	Security Bit	Erase	ISP CLAMP
filtref.sof	EP1C6F256	000A0A84	FFFFFFF	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

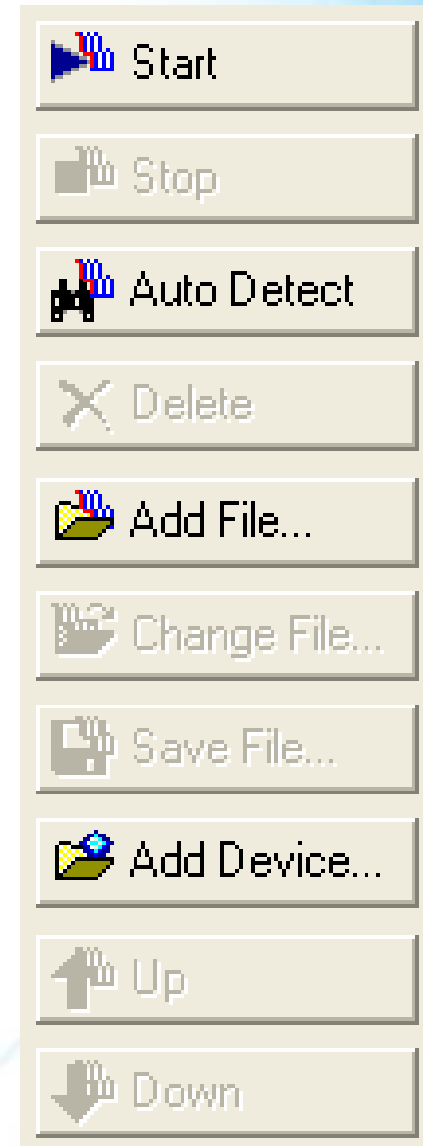
Multiple device chain

The screenshot shows the Quartus II Hardware Setup window for a multiple device chain. The hardware is identified as No Hardware in JTAG mode. The progress bar shows 0%. A table lists the files to be programmed:

File	Device	Checksum	Usercode	Program/Configure	Verify	Blank-Check	Examine	Security Bit	Erase	ISP CLAMP
filtref.sof	EP1C6F256	000A1428	FFFFFFF	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
[-] filtref.pof	EPCS4	075B6DC1	0000000	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
[-] Page_0				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
[-] ../././61/qdesigns...	EPC16	1C661491	FFFFFFF	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
[-] ../././61/qdesigns...	EP2S60F672C5ES	00A99868	FFFFFFF	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

# Programmer Toolbar

- Start programming
- Auto detect devices in JTAG chain
- Add/remove/change devices in chain
- Add/remove/changes files in chain
- Change order of files in chain
- Setup programming hardware



# Setting Up Programming Hardware

Click on the Hardware Setup button

Hardware Setup

Hardware Settings | JTAG Settings

Select a programming hardware setup to use when programming devices. This programming hardware setup applies only to the current programmer window.

Currently selected hardware: USB-Blaster [USB-0]

Available hardware items:

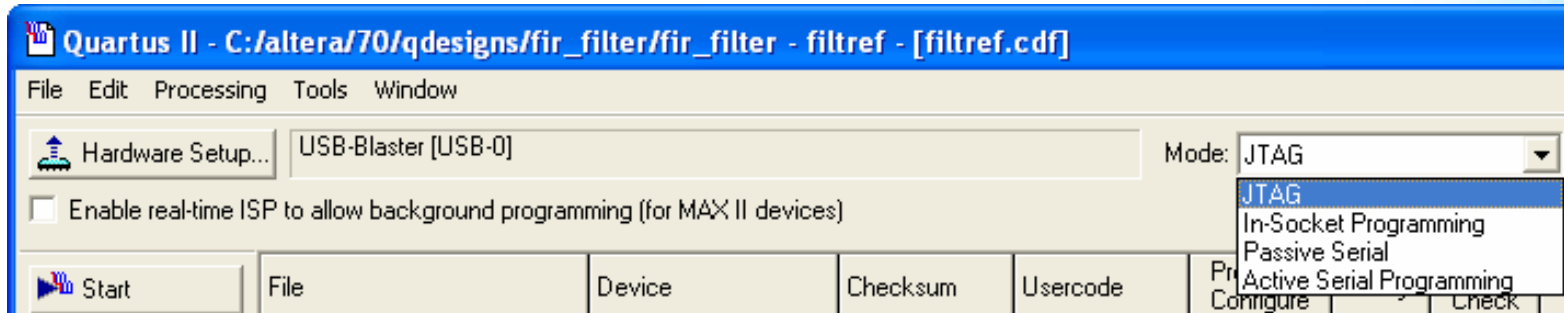
Hardware	Server	Port
USB-Blaster	Local	USB-0

Buttons: Add Hardware..., Remove Hardware, Close

Use drop-down to select programming hardware



# Chain Programming Modes

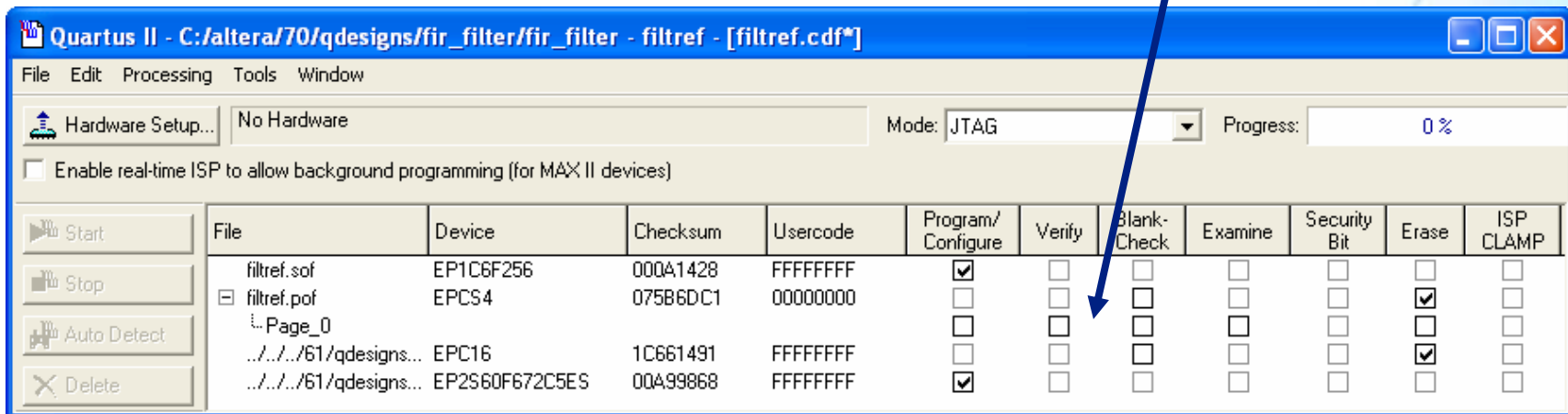


- JTAG
  - JTAG chain consisting of Altera & non-Altera devices
- Passive serial
  - Altera FPGAs only
- Active serial
  - Altera serial configuration devices
- In-socket programming
  - CPLDs & configuration devices in APU

# Programming Options

- Program/Configure
  - Applies to all devices
- Verify, Blank-Check, Examine & Erase
  - Configuration devices
  - MAX II, MAX 7000 & MAX 3000
- Security Bit & ISP Clamp
  - MAX II, MAX 7000 & MAX 3000

Check the appropriate boxes to perform actions when programming starts



The screenshot shows the Quartus II programming interface. The window title is "Quartus II - C:/altera/70/qdesigns/fir\_filter/fir\_filter - filtref - [filtref.cdf\*]". The menu bar includes File, Edit, Processing, Tools, and Window. The hardware setup is "No Hardware" and the mode is "JTAG". The progress is at 0%. There is a checkbox for "Enable real-time ISP to allow background programming (for MAX II devices)".

Start	File	Device	Checksum	Usercode	Program/Configure	Verify	Blank-Check	Examine	Security Bit	Erase	ISP CLAMP
▶ Start	filtref.sof	EP1C6F256	000A1428	FFFFFFFF	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
■ Stop	[-] filtref.pof	EPCS4	075B6DC1	00000000	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
🔍 Auto Detect	Page_0				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
✖ Delete	.././../61/qdesigns...	EPC16	1C661491	FFFFFFFF	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	.././../61/qdesigns...	EP2S60F672C5ES	00A99868	FFFFFFFF	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

# Bypassing Devices In JTAG Chain (1)

Quartus II - C:/altera/70/qdesigns/fir\_filter/fir\_filter - filtref - [filtref.cdf\*]

File Edit Processing Tools Window

Hardware Setup... No Hardware Mode: JTAG Progress: 0%

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

File	Device	Checksum	Usercode	Program/Configure	Verify	Blank-Check	Examine	Security Bit	Erase	ISP CLAMP
filtref.sof	EP1C6F256	000A1428	FFFFFFFF	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
[-] filtref.pof	EPCS4	075B6DC1	00000000	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
[-] Page_0				<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
../.../61/qdesigns...	EPC16	1C661491	FFFFFFFF	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
../.../61/qdesigns...	EP2S60F672C5ES	00A99868	FFFFFFFF	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

**Method 1: Add programming file & leave Program/Configure box unchecked**

# Bypassing Devices In JTAG Chain (2)

Hardware Setup... No Hardware Mode: JTAG Progress: 0%

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

File	Device	Checksum	Usercode	Program/Configure	Verify	Blank-Check	Examine	Security Bit	Erase	ISP CLAMP
filtrf.sof	EP1C6F256	000A1428	FFFFFFFF	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
filtrf.pof	EPCS4	075B6DC1	00000000	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Page_0				<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
../70/qdesigns...	EPC16	1C661491	FFFFFFFF	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
../70/qdesigns...	EP2S60F672C5ES	00A99868	FFFFFFFF	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<none>	EP2C35U484	00000000	<none>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

**Select Devices**

Device family:  Cyclone II

Device name:  EP2C35U484

**Method 2: Click Add Device & select device leaving programming file field blank**

# Adding Non-Altera Device to Chain

Hardware Setup... No Hardware Mode: JTAG Progress: 0%

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

File	Device	Checksum	Usercode	Program/Configure	Verify	Blank-Check	Examine	Security Bit	Erase	ISP CLAMP
filtrf.sof	EP1C6F256	000A1428	FFFFFFFF	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
filtrf.pof	EPCS4	075B6DC1	00000000	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
...Page_0				<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
.../61/qdesigns...	EPC16	1C661491	FFFFFFFF	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
.../61/qdesigns...	EP2K10K100-10	00A99868	FFFFFFFF	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input 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"/>

**Select Devices**

Device family:  Cyclone II,  Enhanced Configuration Devices,  EPC1,  EPC2,  FLEX10K,  FLEX10KA,  FLEX10KE,  FLEX6000,  FLEX8000,  HardCopy II,  MAX II,  MAX3000A,  MAX7000AE,  MAX7000B,  MAX7000S,  MAX9000,  Stratix,  Stratix GX,  Stratix II,  Stratix II GX,  User Defined,  Virtual JTAG TAP

Device name: MY\_DEVICE

**New Device**

Device Name: MY\_DEVICE

Instruction register Length: 8

JTAG ID: ID Mask

Allow none

**Click new & create user-defined devices to add non-Altera devices to chain**

# Starting the Programmer

Click Start

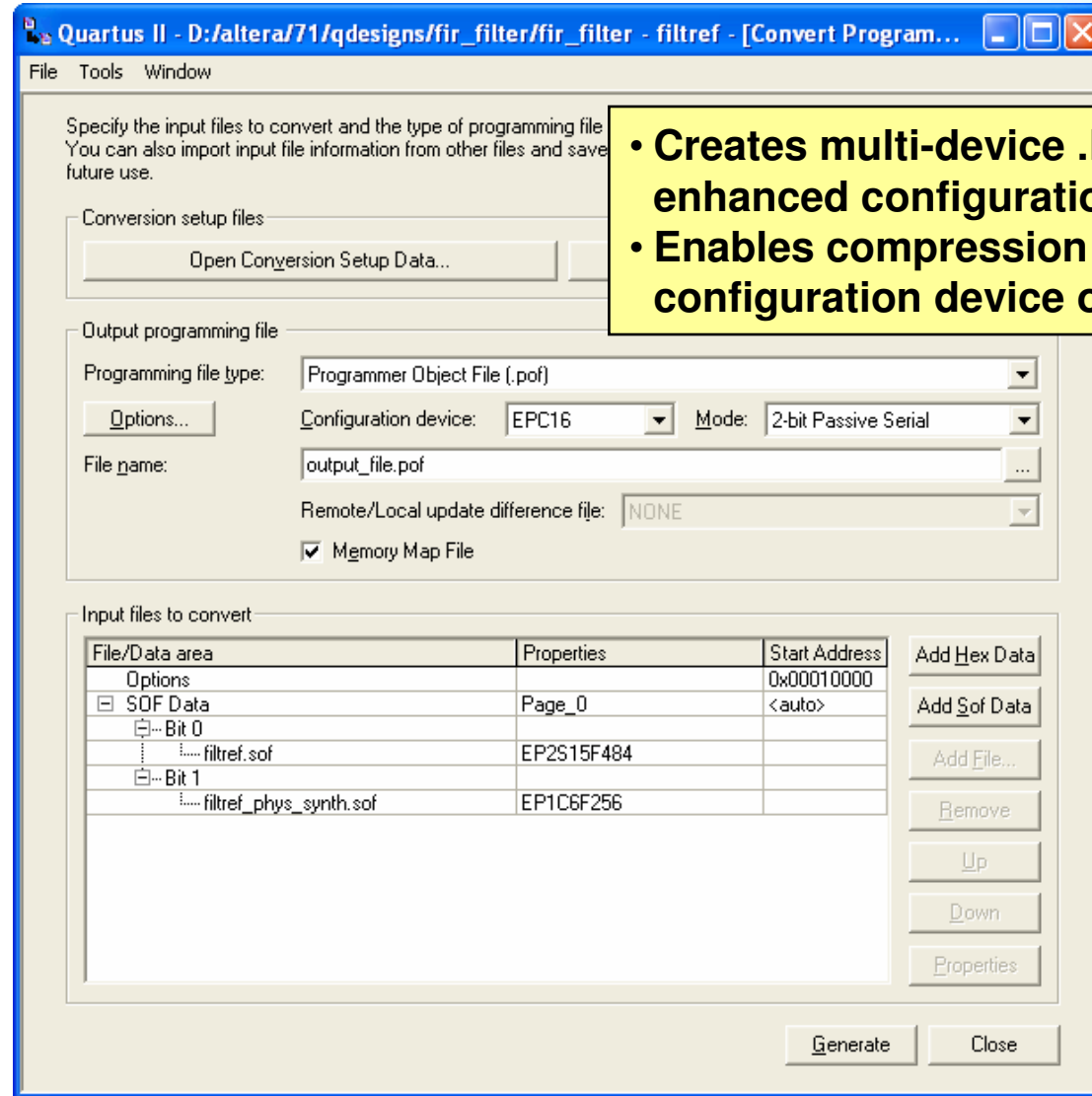
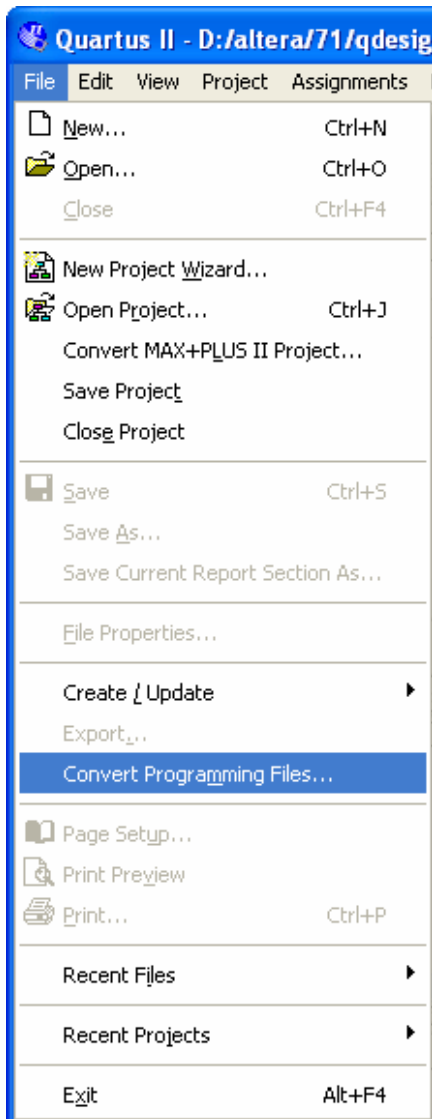
Progress field shows percentage of programming completion

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

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

	File	Device	Checksum	Usercode	Program/Configure	Verify	Blank-Check	Examine	Security Bit	Erase	ISP CLAMP
<input type="checkbox"/>	filtref.sof	EP1C6F256	000A1428	FFFFFFFF	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	[-] filtref.pof	EPCS4	075B6DC1	00000000	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	[-] Page_0				<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	../..../61/qdesigns...	EPC16	1C661491	FFFFFFFF	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	../..../61/qdesigns...	EP2S60F672C5ES	00A99868	FFFFFFFF	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input 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"/>

# Converting SOF Programming Files



- Creates multi-device .POF for enhanced configuration devices
- Enables compression & other configuration device options

## *Exercise 7 Demonstration*



# Programming/Configuration Summary

- Setting device options
- Generating programming files
- Programming device or devices in chain
- Converting programming files

# Programming Support Resources

- Programming Center
- Configuration Center

# Class Summary

- Design entry techniques
- Project creation
- Compiler settings & assignment editor
- Timing analysis
- Simulation
- Programming/configuration

# Learn More Through Technical Training

## Instructor-Led Training



With Altera's instructor-led training courses, you can:

- Listen to a lecture from an Altera technical training engineer (instructor)
- Complete hands-on exercises with guidance from an Altera instructor
- Ask questions & receive real-time answers from an Altera instructor
- Each instructor-led class is one or two days in length (8 working hours per day).

## Online Training



With Altera's online training courses, you can:

- Take a course at any time that is convenient for you
- Take a course from the comfort of your home or office (no need to travel as with instructor-led courses)

Each online course will take approximate one to three hours to complete.

<http://www.altera.com/training>

View training class schedule & register for a class

# Advanced Quartus II Courses

## ■ Quartus II Software Design Series: Verification

- Timing analysis
  - Thorough investigation of performing timing analysis on an Altera device with TimeQuest
- Power analysis
- Debugging solutions
  - SignalProbe incremental routing
  - Logic Analyzer Interface
  - In-System Memory Content Editor
  - In-System Sources & Probes
  - Chip Planner & Resource Property Editor
  - SignalTap II Embedded Logic Analyzer

## ■ Quartus II Software Design Series: Optimization

- Incremental Compilation
- Quartus II optimization features & techniques

# Altera Technical Support

- Reference Quartus II software on-line help
- [Quartus II Handbook](#)
- Consult Altera applications (factory applications engineers)
  - MySupport: <http://www.altera.com/mysupport>
  - Hotline: (800) 800-EPLD (7:00 a.m. - 5:00 p.m. PST)
- Field applications engineers: contact your local Altera sales office
- Receive literature by mail: (888) 3-ALTERA
- FTP: <ftp.altera.com>
- World-wide web: <http://www.altera.com>
  - Use solutions to search for answers to technical problems
  - View design examples

# Give us your feedback

- When you registered for this training you received a confirmation email
- Please click on the link in the email to complete a short survey
- Your feedback is important to help us improve future trainings!

Thank you!

**ALTERA**

# Quartus II Software Design Series: Foundation

*Appendix*



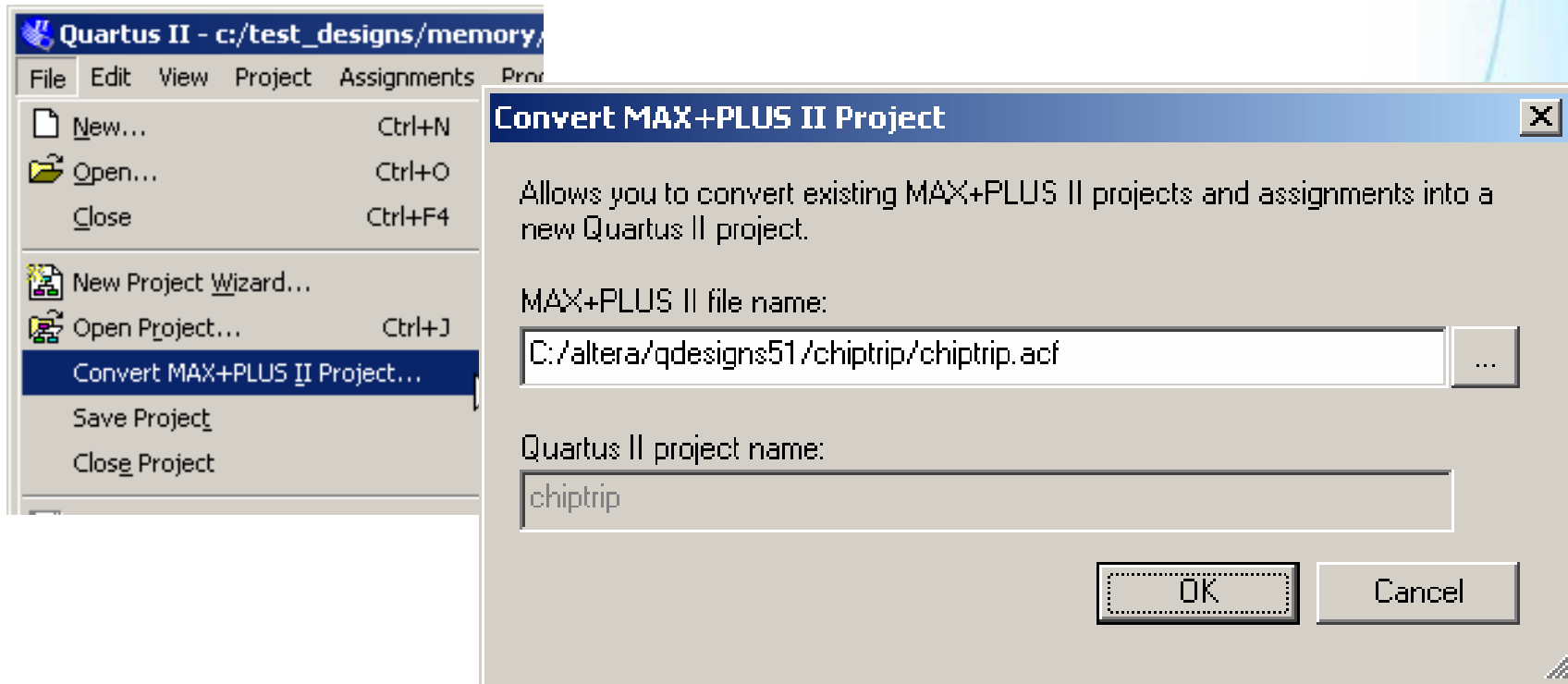


# Appendix

- Converting MAX+PLUS II designs to Quartus II
- Schematic design entry
- QSF Notes
- More fitter settings
- Power Optimizations
- Early I/O planning tasks
- TimeQuest “golden” SDC file tip
- Simulating in Quartus II

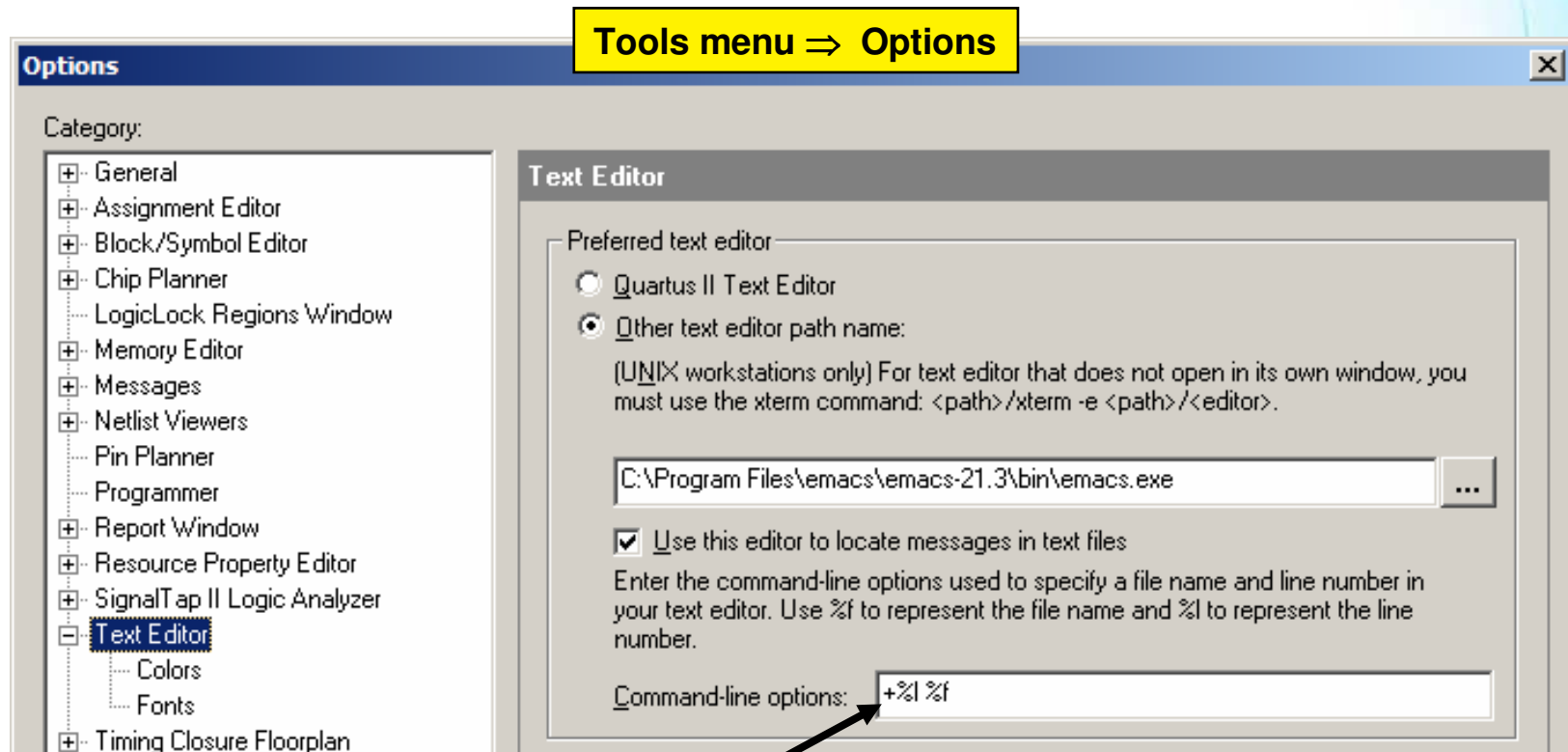
# MAX+PLUS II to Quartus II

- Convert MAX+PLUS II projects into Quartus II projects
- Assignments automatically translated



# Using Own Text Editor

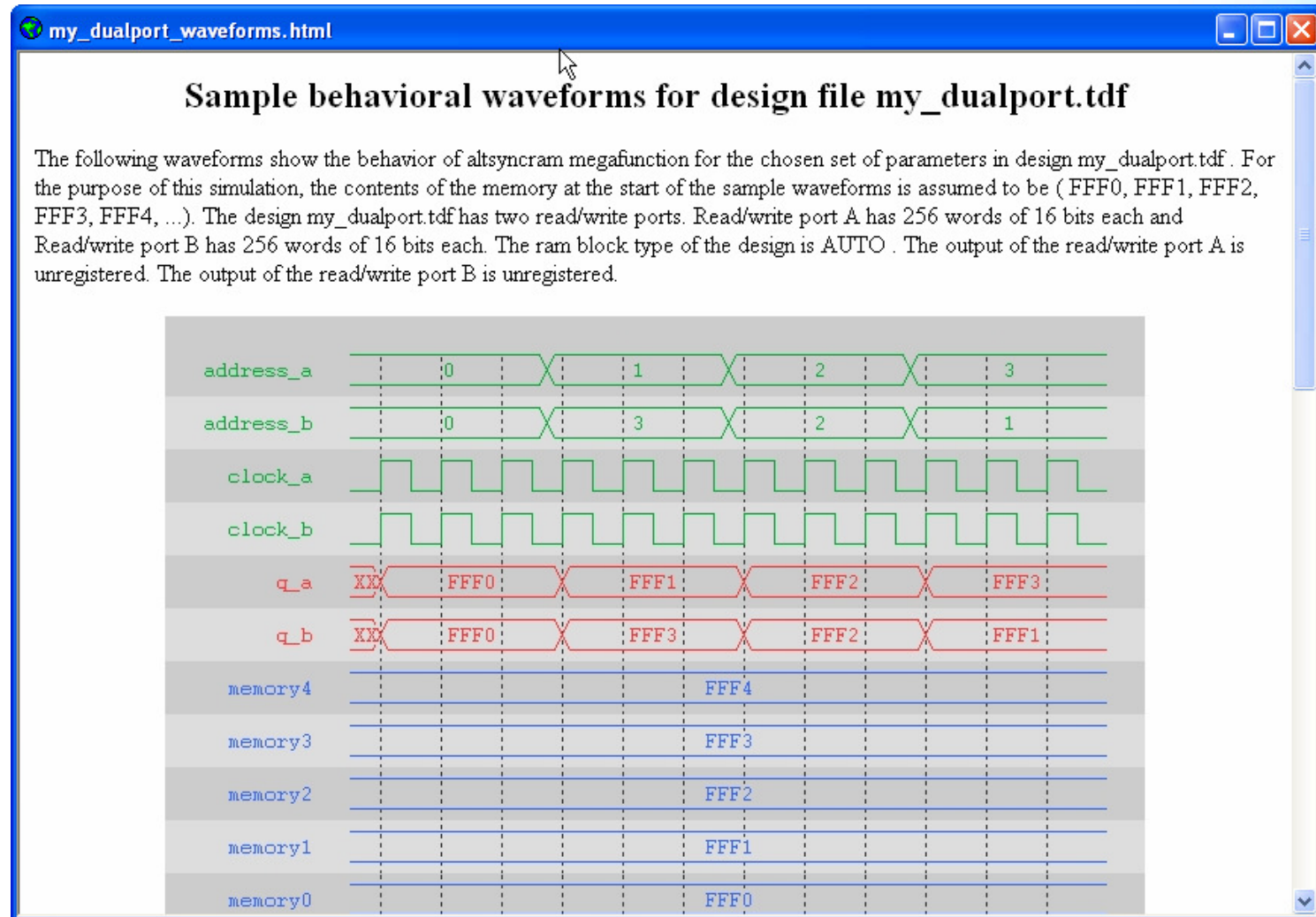
- Enter path to preferred text editor executable



# Behavioral Waveforms

- HTML file generated by MegaWizard
- Description of megafunction functionality
  - Reviews selected parameters
  - Describes read & write operations
- Supported megafunctions
  - Subset of memory
  - Subset of arithmetic
  - PLL

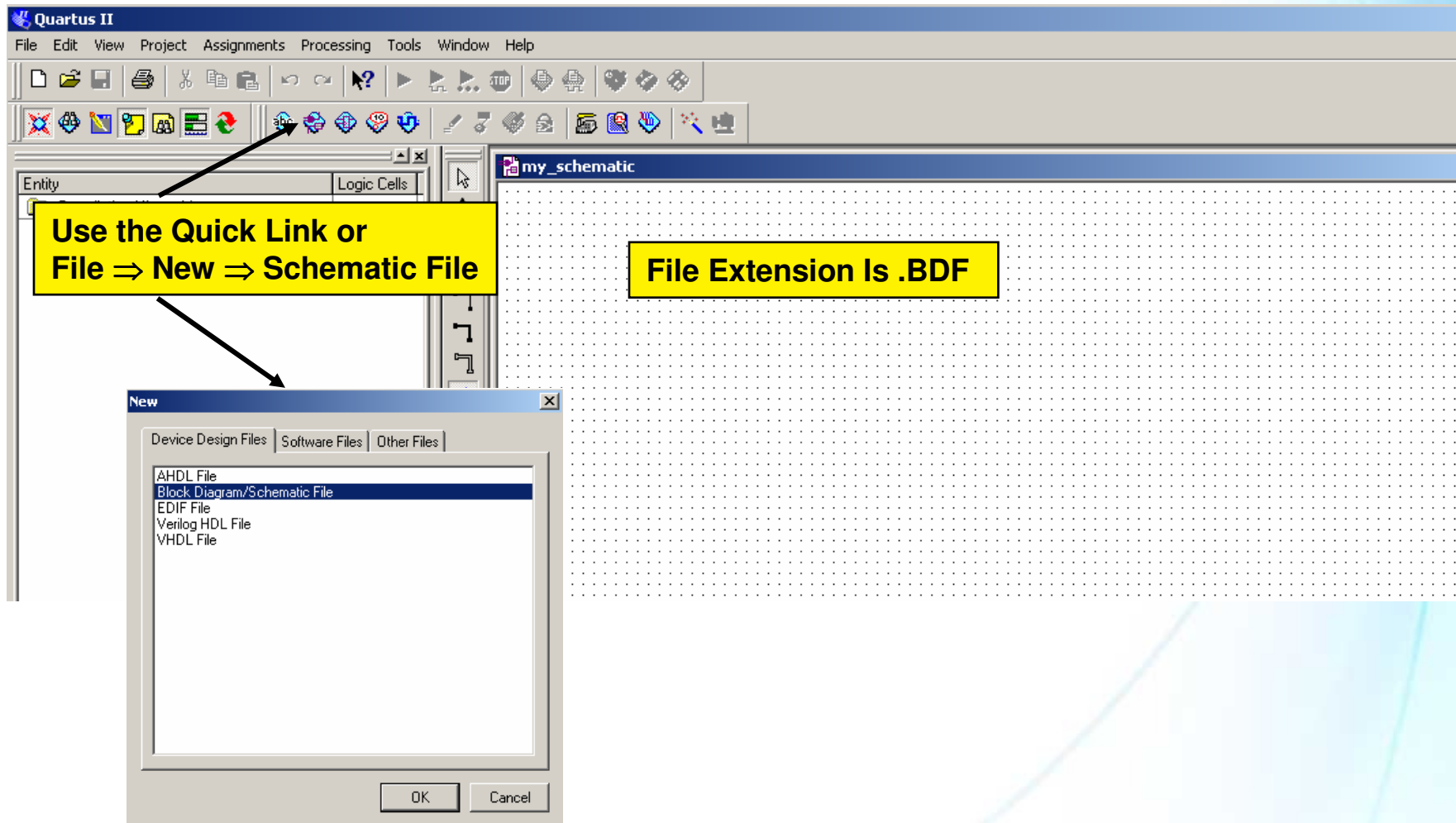
# Example Waveform



# Schematic Design Entry

- Full-featured schematic design capability
- Schematic design creation
  - Draw schematics using library functions (blocks)
    - Gates, flip-flops, pins & other primitives
    - Altera megafunctions & LPMs
  - Create symbols for Verilog, VHDL, or AHDL design files
  - Connect all blocks using wires & busses

# Create Schematic



# Insert Symbols

**Open the Symbol Window:  
Use the Toolbar or  
Double Click Schematic  
Background**

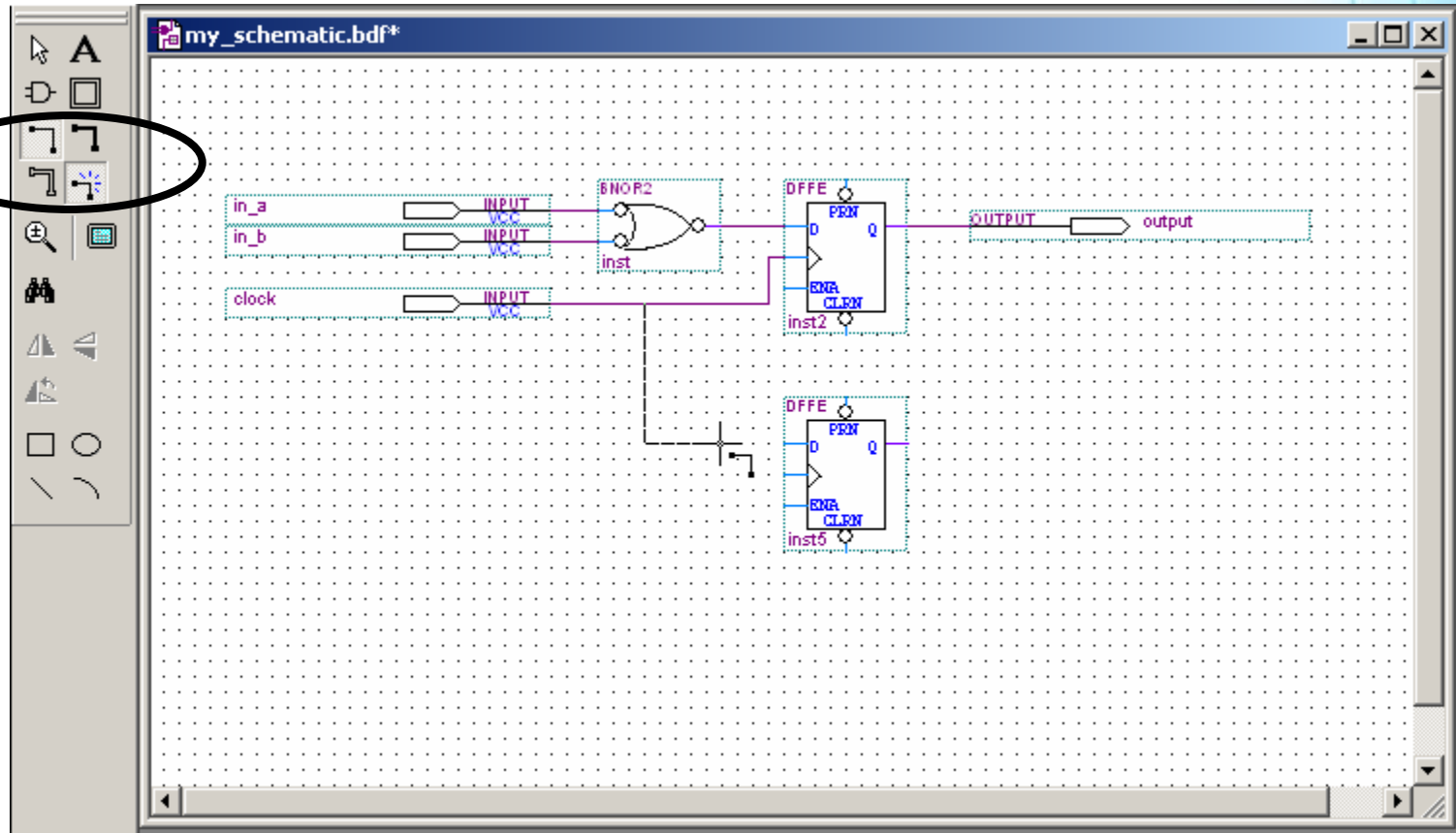
**Local Symbols  
Created from  
MegaWizard or  
Design Files**

**Library Symbols**



# Connect Wires & Buses

Draw Wires,  
Buses, or Conduit



# Change Names & Properties

The screenshot shows the Quartus II interface. On the left, the Project Navigator displays a hierarchy for 'Stratix II: EP2S15F484C3' with components like 'filtref', 'taps:inst', 'state\_m:inst1', 'hvalues:inst2', 'acc:inst3', and 'mult:inst6'. The main window shows a schematic diagram for 'filtref.bdf' with inputs 'clkx2', 'clk', 'reset', and 'newt'. Two callout boxes provide instructions: one pointing to the 'clk' pin name and another pointing to the 'taps' block.

**Double-Click on Pin Name to Change; Hit Enter to Advance to Next Pin**

**Right-Click on any Block to Change Properties (Ex. Instance Name)**

I/O	Type
clk	INPUT
reset	INPUT
sel[1..0]	INPUT
newt	INPUT
clk[2..0]	INPUT
x[1..0]	OUTPUT

I/O	Type
sel[1..0]	INPUT
h[2..0]	OUTPUT

Parameter	Value

# Create Symbols

Converted schematic to a Symbol to be used in other schematic files

The screenshot shows the Quartus II IDE with a project named 'mult8x8'. The main window displays the HDL code for a module named 'mult8x8'. The code is as follows:

```
1 module mult8x8 (a, b, start, reset, clk,  
2                 sega, segb, segc, segd, sege, segf, segg,  
3                 done_flag, result);  
4  
5 input a[7:0];  
6 input b[7:0];  
7 output sega[7:0];  
8 output segb[7:0];  
9  
10 wire re;  
11 wire [3:0] s;  
12 wire [7:0] r;
```

The 'File' menu is open, and the path 'File → Create/Update → Create Symbol...' is highlighted. A yellow callout box points to this path with the text: 'File ⇒ Create/Update ⇒ Create Symbol...'. The 'Create Symbol' dialog box is open, showing the 'Project' directory selected in the 'Libraries' list. A yellow callout box points to this selection with the text: 'Symbol Created in Project Directory'. The dialog box also shows the 'Name' field set to 'mult8x8' and several checkboxes for options like 'Repeat-insert mode', 'Insert symbol as block', and 'Launch MegaWizard Plug-In'. In the background, a schematic diagram of the 'mult8x8' symbol is visible, showing its inputs (a, b, start, reset, clk) and outputs (sega, segb, segc, segd, sege, segf, segg, done\_flag, result).

# Convert BDF for HDL

**File** ⇒ **Create/Update** ⇒ **Create Symbol...**

**Choose VHDL or Verilog**

**Create HDL Design File for Current File**

File type

VHDL

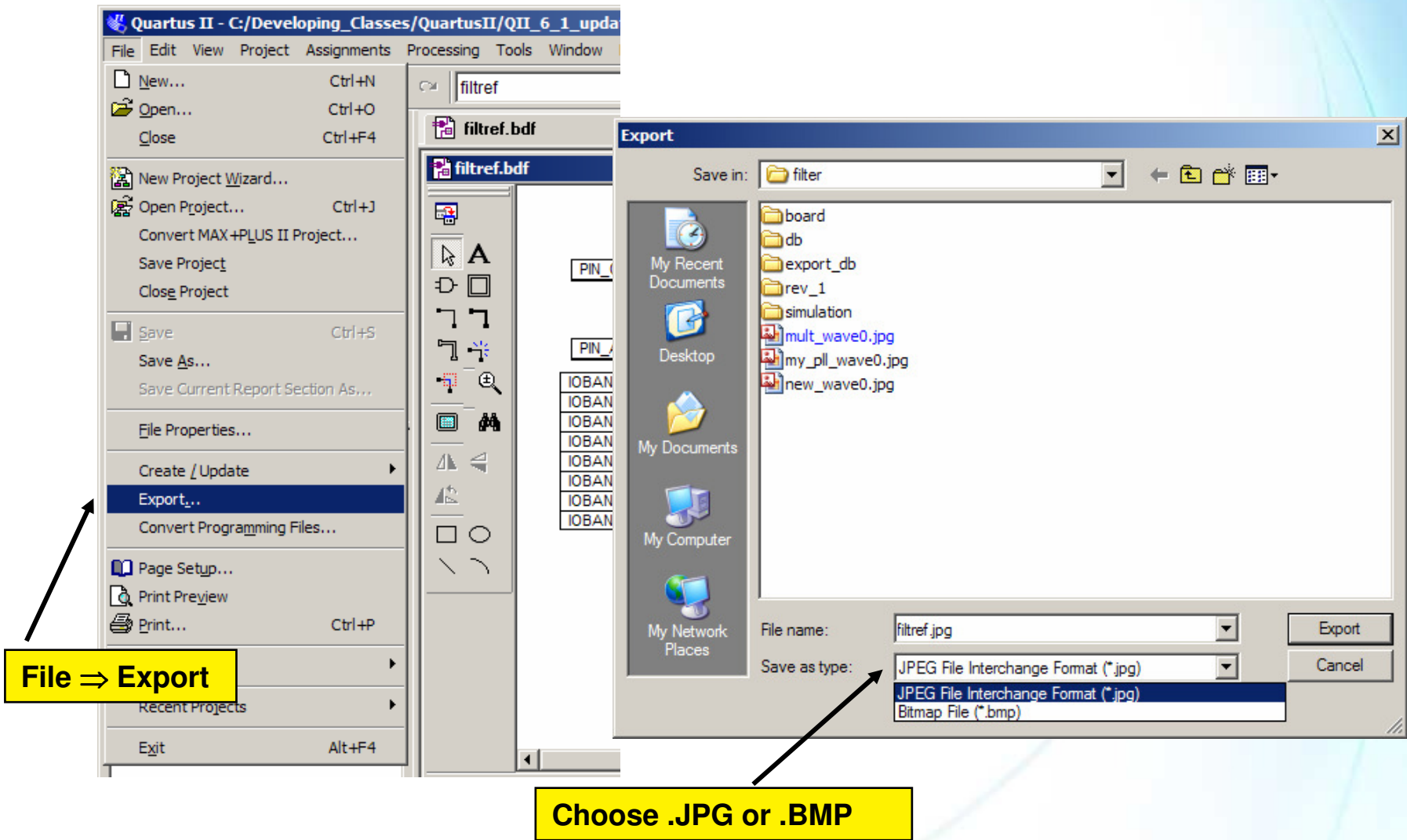
Verilog HDL

File name: C:/Developing\_Classes/QuartusII/QII\_6\_1\_u

I/O	Type
sel[1..0]	INPUT
h[2..0]	OUTPUT

I/O	Type
clk	INPUT

# Convert BDF to Image File



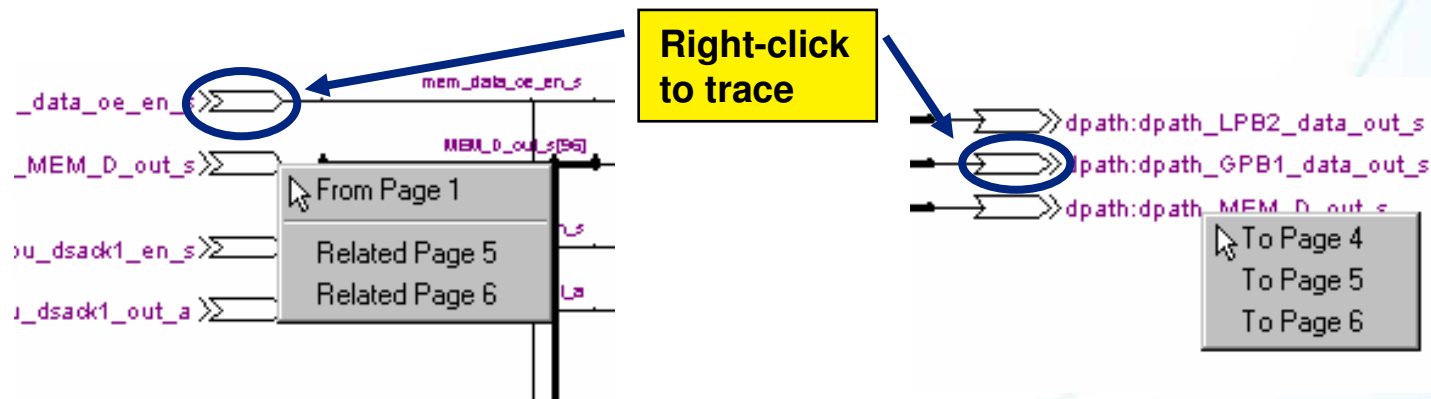
# QSF Notes

- Changes to existing assignments updates (i.e. Not moved to end of file)
  - Exceptions (always at end of file)
    - Adding/removing source files
    - Editing members in assignment group
- Sourcing rules
  - Source statements always preserved (not overwritten)
  - Changes to assignments read from source file written back to original file
  - New assignments always written to end of main QSF
  - All assignments copied into new QSF file when new revision created

# Page Control



- Hierarchical levels automatically partitioned
  - Control design size per page (tools ⇒ customize ⇒ options)
- Use toolbar to move between pages
- Navigate nets between pages

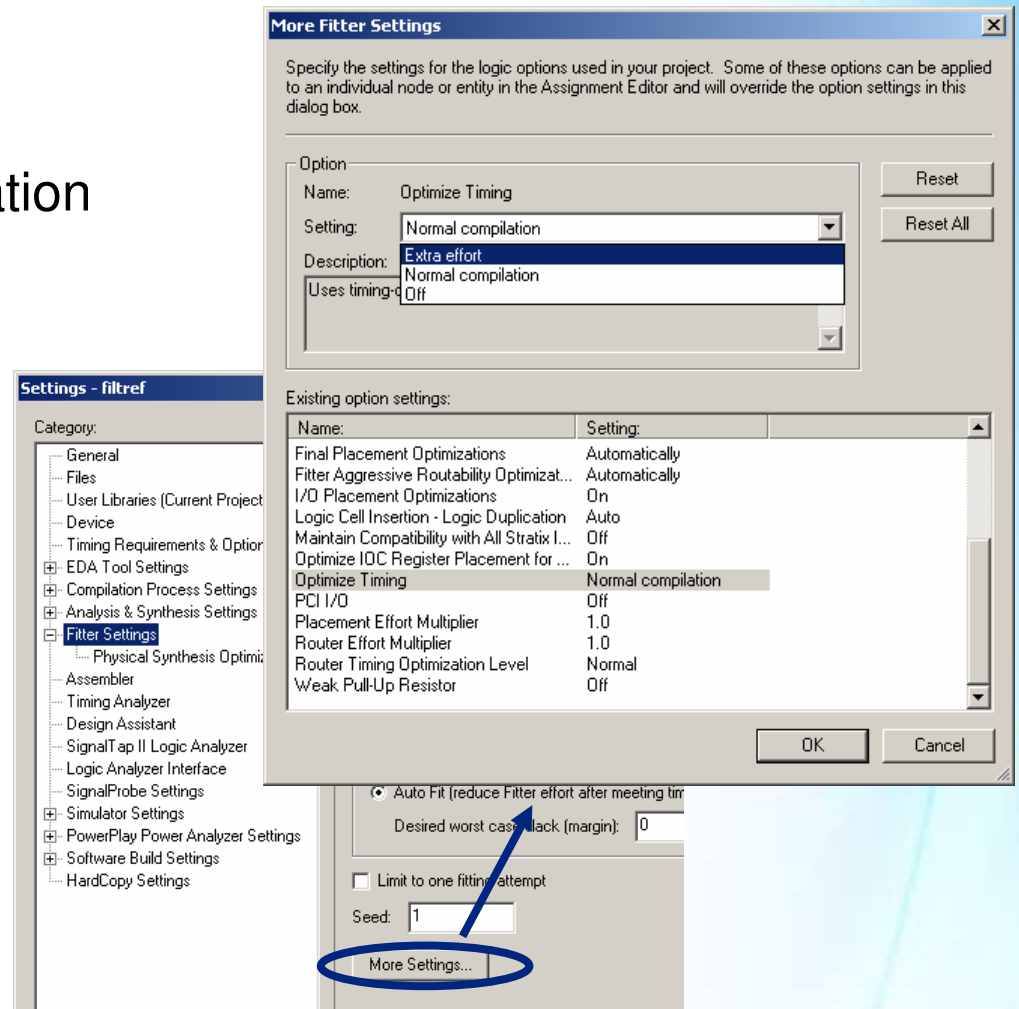


# Fitter Settings – More Settings

- Adjust project-wide defaults
- Enable/disable increased/decreased optimization routines
  - May affect compile time

## Examples

- **Optimize IOC Register Placement for Timing**
  - Automatically Uses I/O Cell Registers to Improve I/O Pin Timing
- **Auto Global Signals**
  - Auto-Promotes High-Fanout Control Signals to Globals
- **Fitter Aggressive Routability Optimization**
  - Automatically Performed during Increased Fitting Attempts





# Importing I/O Megafunction

- Select PPF file created when running MegaWizard Plug-In Manager

MegaWizard Plug-In Manager - ALTLVDS [page 21 of 21] -- Summary

**ALTLVDS**  
Version 7.1

1 Parameter Settings | 2 EDA | 3 Summary

Turn on the files you wish to generate. A gray checkmark indicates automatically generated, and a red checkmark indicates an option. Finish to generate the selected files. The state of each checkbox is maintained in subsequent MegaWizard Plug-In Manager sessions.

The MegaWizard Plug-In Manager creates the selected files in the directory: D:\altera\71\qdesigns\fir\_filter\

File	Description
<input checked="" type="checkbox"/> my_interface.v	VHDL component file
<input checked="" type="checkbox"/> my_interface.ppf	PinPlanner ports PPF file
<input type="checkbox"/> my_interface.ins	AMBL include file
<input type="checkbox"/> my_interface.cmp	VHDL component declaration file
<input checked="" type="checkbox"/> my_interface.bsf	Quartus II symbol file
<input type="checkbox"/> my_interface_inst.v	Instantiation template file
<input checked="" type="checkbox"/> my_interface_bb.v	Verilog HDL black-box file

Resource Usage

1 clkctrl + 32 reg + 5 stratixii\_lvds\_transmitter + 1 stratixii\_pll

**Create/Import Megafunction**

Create a new custom megafunction

Import an existing custom megafunction

:\altera\71\qdesigns\fir\_filter\my\_interface.ppf ...









Instance name: my\_interface

OK Cancel

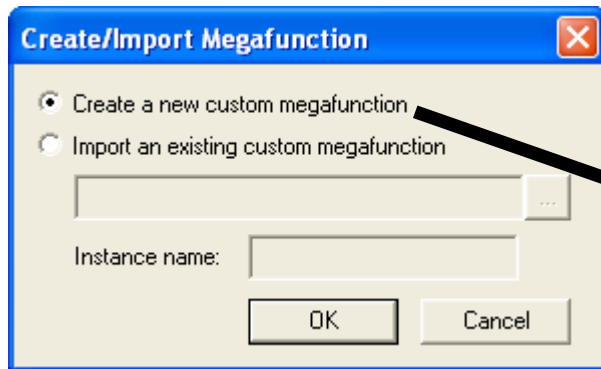
**Pin Planner Edit menu => Create/Import Megafunction**

# Imported Megafunction

- Megafunction pins appear as a new bus in Groups List
  - Names prefixed with user-defined instance name

	Node Name	Direction
 +	d[7..0]	Input Group
 +	yn_out[7..0]	Output Group
 -	my_interface	Bidir Group
 +	my_interface_tx_in[31..0]	Input Group
	my_interface_tx_inclock	Input
	my_interface_tx_corecl...	Output
 +	my_interface_tx_out[3....	Output Group
	my_interface_tx_outclock	Output
	<<new node>>	

# Creating New I/O Megafunction



Right-click in Groups List or All Pins List  
or  
Pin Planner Edit menu ⇒ Create/Import Megafunction  
or  
Click on toolbar button

I/O-related megafunctions only

Create and/or import pins and pin properties from I/O related megafunctions & IP (i.e. DDR, LVDS)

# Configure & Generate Top-Level HDL (1)

- Choose Set Up Top-Level Design File
  - Right-click, Edit menu, or toolbar
- Configure megafunctions & IP cores by changing values in Type and Node Name columns

**Set Up Top-Level Design File**

Megafunctions: my\_ddioin [ALTDDIO\_IN], my\_ddioout [ALTDDIO\_OUT]

User Nodes

Megafunction: ALTDDIO\_IN Instance: my\_ddioin

Port Name	Direction	Type	Node Name
aclr	Input	Internal	reset
datain[7..0]	Input	External	datain[7..0]
inclock	Input	External	clock
dataout_h[7..0]	Output	Internal	my_ddioin_dataout_h[7..0]
dataout_l[7..0]	Output	Internal	my_ddioin_dataout_l[7..0]

Instance Nodes:

- d[7..0]
- datain[7..0]
- my\_ddioin\_dataout\_h[7..0]
- my\_ddioin\_dataout\_l[7..0]
- my\_ddioout\_datain[7..0]
- my\_ddioout\_dataout\_h[7..0]
- my\_ddioout\_dataout\_l[7..0]
- yn\_out[7..0]

**Port Name & Direction comes from megafunction definition & cannot be changed**

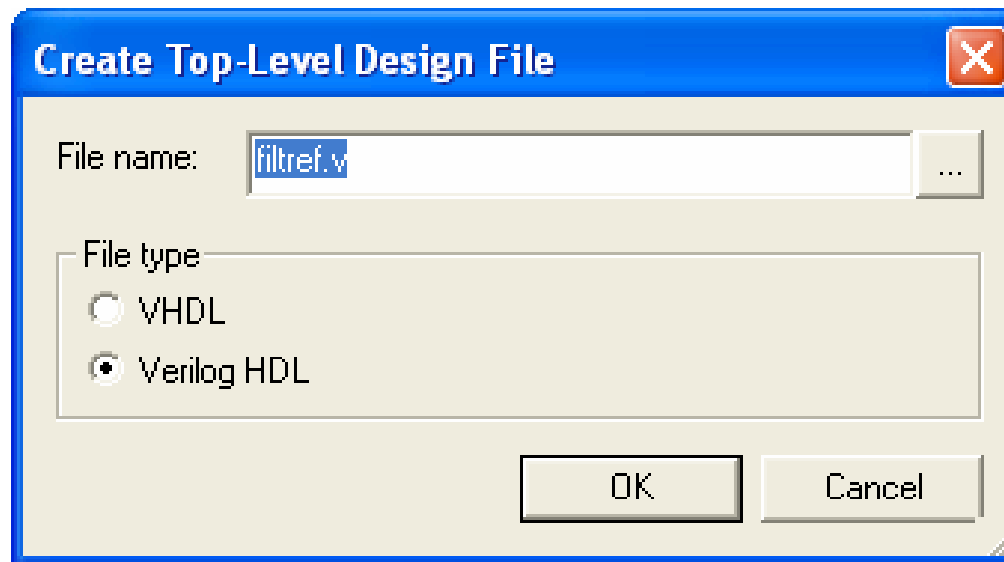
**Choose Type:**  
*External* type connect to device pins and *will* appear in Pin Planner pin lists if left unconnected  
*Internal* type connect to or become internal nodes and *will not* appear in Pin Planner pin lists

**Use Node Name to connect megafunction ports to device I/O and internal node names**

- Connect to other megafunction ports by reusing port or node names (double-click & select from drop-down)
- Rename device I/O & create new node names by typing in new names

# Configure & Generate Top-Level HDL (2)

- Choose Create Top-Level Design File
  - Right-click, File menu (Create/Update submenu), or toolbar
- Select a file name and type

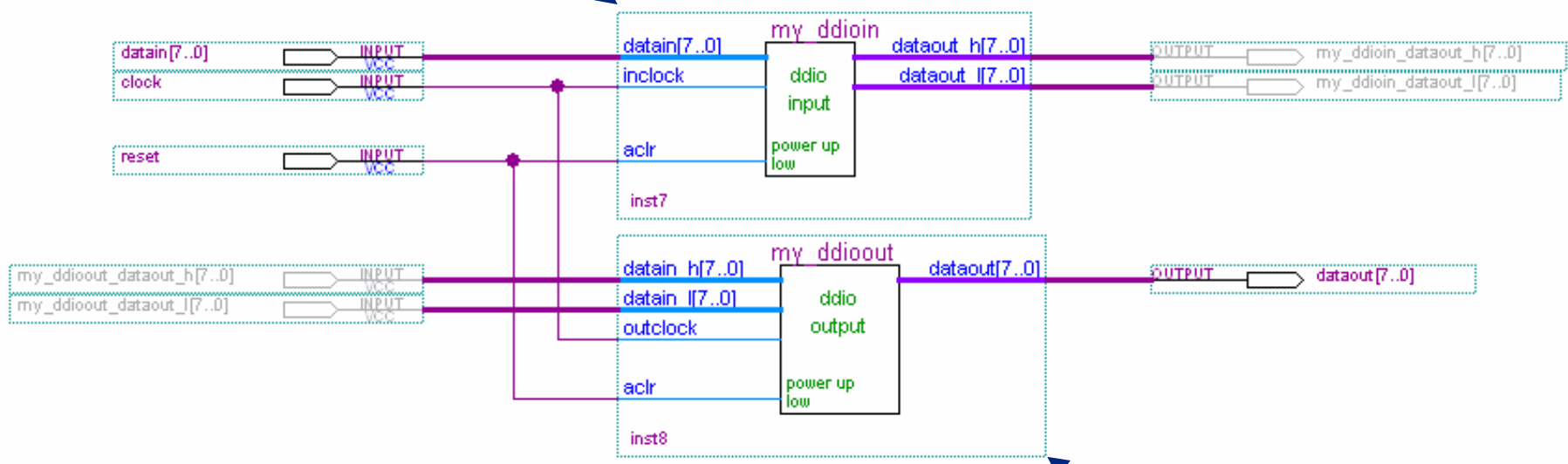


**Set Up Top-Level Design File**

Megafunctions  
 my\_ddioin (ALTDIO\_IN)  
 my\_ddioout (ALTDIO\_OUT)  
 User Nodes

Megafunction: ALTDIO\_IN Instance: my\_ddioin

Port Name	Direction	Type	Node Name
aclr	Input	External	reset
datain[7..0]	Input	External	datain[7..0]
inclock	Input	External	clock
dataout_h[7..0]	Output	Internal	
dataout_l[7..0]	Output	Internal	



**Schematic representation of top level file created (schematic not actually created)**

**Set Up Top-Level Design File**

Megafunctions  
 my\_ddioin (ALTDIO\_IN)  
 my\_ddioout (ALTDIO\_OUT)  
 User Nodes

Megafunction: ALTDIO\_OUT Instance: my\_ddioout

Port Name	Direction	Type	Node Name
aclr	Input	External	reset
datain_h[7..0]	Input	Internal	
datain_l[7..0]	Input	Internal	
outclock	Input	External	clock
dataout[7..0]	Output	External	dataout[7..0]



## Notes on Step 2 (Early I/O Planning Methodology)

- Internal nodes will be commented and set as virtual pins in created top-level file (represented as gray pins on previous slide)
- Make changes to megafunction or IP through the Pin Planner
  - Highlight in Groups List and choose Edit Megafunction (Edit menu or right-click)
  - Must re-import megafunction or IP when changes are made
  - Must update top-level file when changes are made
  - Run I/O Assignment Analysis after any change to validate I/O

# Pin Planner – Error Checking

	Error Checking Description (Examples)
1	Resource availability in I/O bank or Vref group
2	Once half of differential pair is assigned, complement pin not available for assignment
3	No output/bidirectional pins assigned to dedicated input pins (e.g. clock pins)
4	Pin I/O standard must be compatible with the node's I/O standard
5	All nodes in same Vref group must have the same Vref voltage
6	A node assigned to the I/O bank must have an I/O standard compatible with the bank VCCIO

*Recommendation: Run I/O Assignment Analysis (Discussed Later)  
for Complete I/O & I/O-Core Checking*



# Stratix II I/O Rules Section

- Folder in Compilation Report containing I/O Rules tables
  - Summary with number of I/O Rules checked, passed & failed
  - List of I/O Rules & whether passed or failed
  - Matrix of I/O Rules checked on a per pin basis

**I/O Rules Summary**

I/O Rules Statistic	Total
1 Total I/O Rules	31

**I/O Rules Details**

Status	ID	Category	Rule Description
Pass	IO_000001	IO_000001	IO_000001

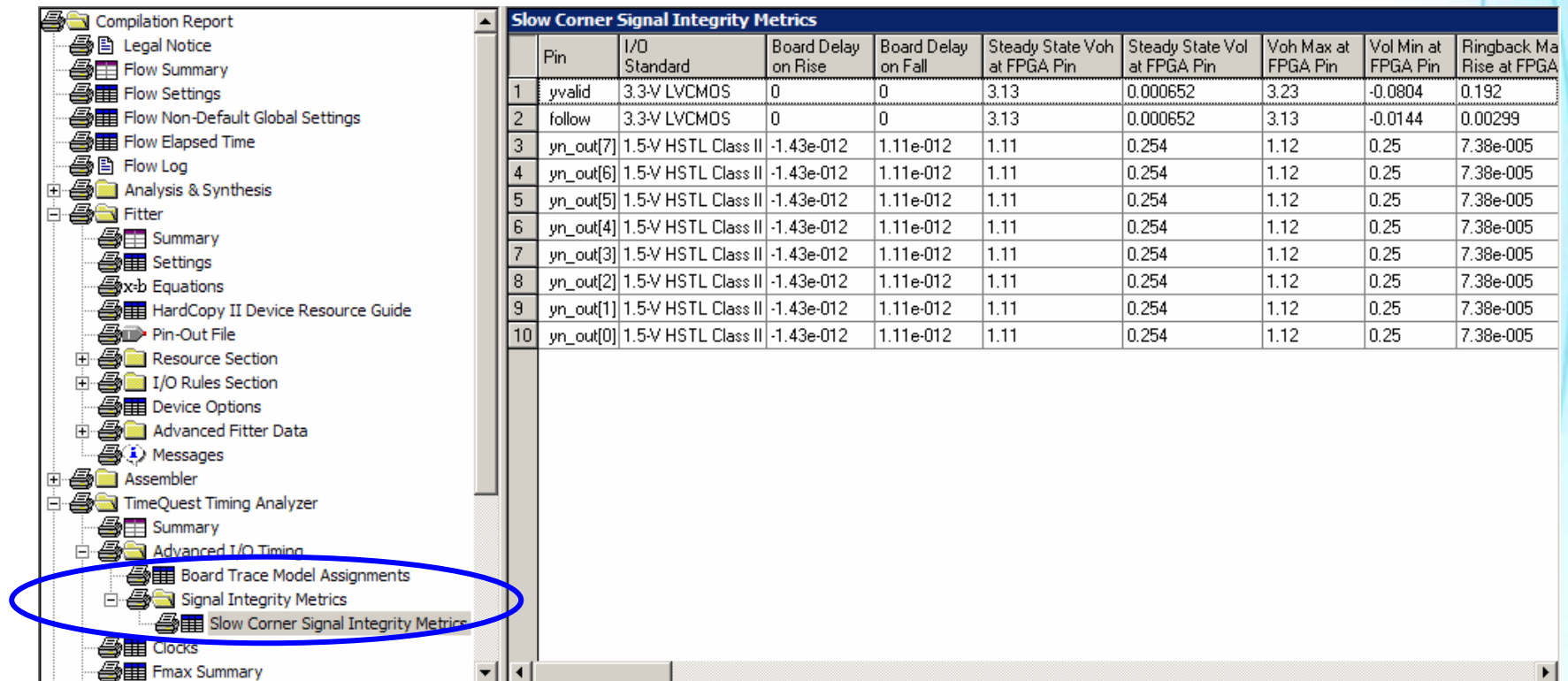
**I/O Rules Matrix**

Pin/Rules	IO_000001	IO_000002	IO_000003	IO_000004	IO_000005	IO_000006	IO_000007	IO_000008
1 Total Pass	23	0	23	0	13	23	23	0
2 Total Unchecked	0	0	0	0	0	0	0	0
3 Total Inapplicable	0	23	0	23	10	0	0	23
4 Total Fail	0	0	0	0	0	0	0	0
5 yvalid	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pass	Pass	Inapplicable
6 follow	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pass	Pass	Inapplicable
7 yn_out[7]	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pass	Pass	Inapplicable
8 yn_out[6]	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pass	Pass	Inapplicable
9 yn_out[5]	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pass	Pass	Inapplicable
10 yn_out[4]	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pass	Pass	Inapplicable
11 yn_out[3]	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pass	Pass	Inapplicable
12 yn_out[2]	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pass	Pass	Inapplicable
13 yn_out[1]	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pass	Pass	Inapplicable
14 yn_out[0]	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pass	Pass	Inapplicable
15 clk	Pass	Inapplicable	Pass	Inapplicable	Pass	Pass	Pass	Inapplicable
16 reset	Pass	Inapplicable	Pass	Inapplicable	Pass	Pass	Pass	Inapplicable
17 clkx2	Pass	Inapplicable	Pass	Inapplicable	Pass	Pass	Pass	Inapplicable
18 newt	Pass	Inapplicable	Pass	Inapplicable	Pass	Pass	Pass	Inapplicable
19 d[0]	Pass	Inapplicable	Pass	Inapplicable	Pass	Pass	Pass	Inapplicable
20 d[1]	Pass	Inapplicable	Pass	Inapplicable	Pass	Pass	Pass	Inapplicable
21 d[2]	Pass	Inapplicable	Pass	Inapplicable	Pass	Pass	Pass	Inapplicable
22 df31	Pass	Inapplicable	Pass	Inapplicable	Pass	Pass	Pass	Inapplicable



# Advanced I/O Analysis Output

- Detailed Signal Integrity Metrics appears as in table in timing analysis report



The screenshot shows the Quartus II software interface. On the left is a project tree with 'Advanced I/O Timing' expanded, and 'Slow Corner Signal Integrity Metrics' highlighted with a blue oval. On the right is a table titled 'Slow Corner Signal Integrity Metrics' with the following data:

Pin	I/O Standard	Board Delay on Rise	Board Delay on Fall	Steady State Voh at FPGA Pin	Steady State Vol at FPGA Pin	Voh Max at FPGA Pin	Vol Min at FPGA Pin	Ringback Ma Rise at FPGA	
1	yvalid	3.3-V LVCMOS	0	0	3.13	0.000652	3.23	-0.0804	0.192
2	follow	3.3-V LVCMOS	0	0	3.13	0.000652	3.13	-0.0144	0.00299
3	yn_out[7]	1.5-V HSTL Class II	-1.43e-012	1.11e-012	1.11	0.254	1.12	0.25	7.38e-005
4	yn_out[6]	1.5-V HSTL Class II	-1.43e-012	1.11e-012	1.11	0.254	1.12	0.25	7.38e-005
5	yn_out[5]	1.5-V HSTL Class II	-1.43e-012	1.11e-012	1.11	0.254	1.12	0.25	7.38e-005
6	yn_out[4]	1.5-V HSTL Class II	-1.43e-012	1.11e-012	1.11	0.254	1.12	0.25	7.38e-005
7	yn_out[3]	1.5-V HSTL Class II	-1.43e-012	1.11e-012	1.11	0.254	1.12	0.25	7.38e-005
8	yn_out[2]	1.5-V HSTL Class II	-1.43e-012	1.11e-012	1.11	0.254	1.12	0.25	7.38e-005
9	yn_out[1]	1.5-V HSTL Class II	-1.43e-012	1.11e-012	1.11	0.254	1.12	0.25	7.38e-005
10	yn_out[0]	1.5-V HSTL Class II	-1.43e-012	1.11e-012	1.11	0.254	1.12	0.25	7.38e-005

# Alternative Method to Writing SDC File

- Maintain “golden” SDC input file in SDC File Editor
  - File holds user-entered constraints
  - Add design-specific comments
  - Arrange SDC commands according to design
- Edit “golden” SDC file
  - Edit input SDC file directly using file editor features
  - Copy SDC commands from Console pane (History tab)
  - Copy commands from TimeQuest-generated output SDC file

# Quartus II Simulation

- Simulator method & features overview
- Simulator settings
- VWF file creation
- Simulation output

# Supported Simulation Methods

## ■ Quartus II

- VWF (vector waveform file)
  - Primary graphical waveform file
- CVWF (compressed vector waveform file)
  - Compressed binary version of VWF
- VCD (value change dump file)
  - Standardized text-based input file (IEEE-1364)
- VEC (vector file)
  - Text-based input file
- Tcl/TK scripting

## ■ 3<sup>rd</sup>-party simulators

- Verilog/VHDL testbench

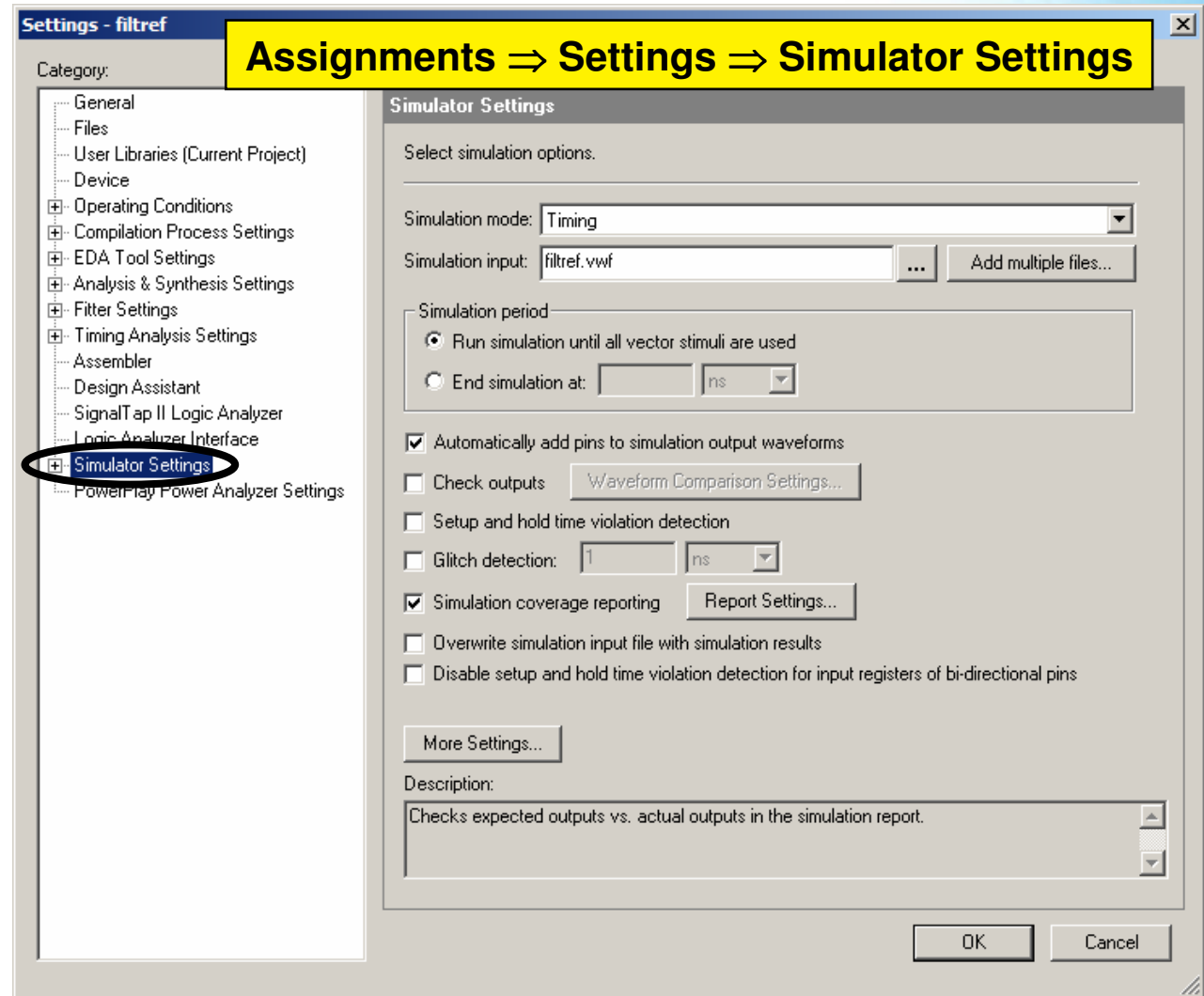
**Note:** The simulator channel file (.SCF) and table file (.TBL) are also supported for backwards compatibility with MAX+PLUS II

# Simulator Features

- Converts VWF into HDL testbench
- Generates HDL testbench template
- Supports breakpoints
- Performs automatically
  - Adding output pins to output waveform file
  - Checking outputs at end of simulation

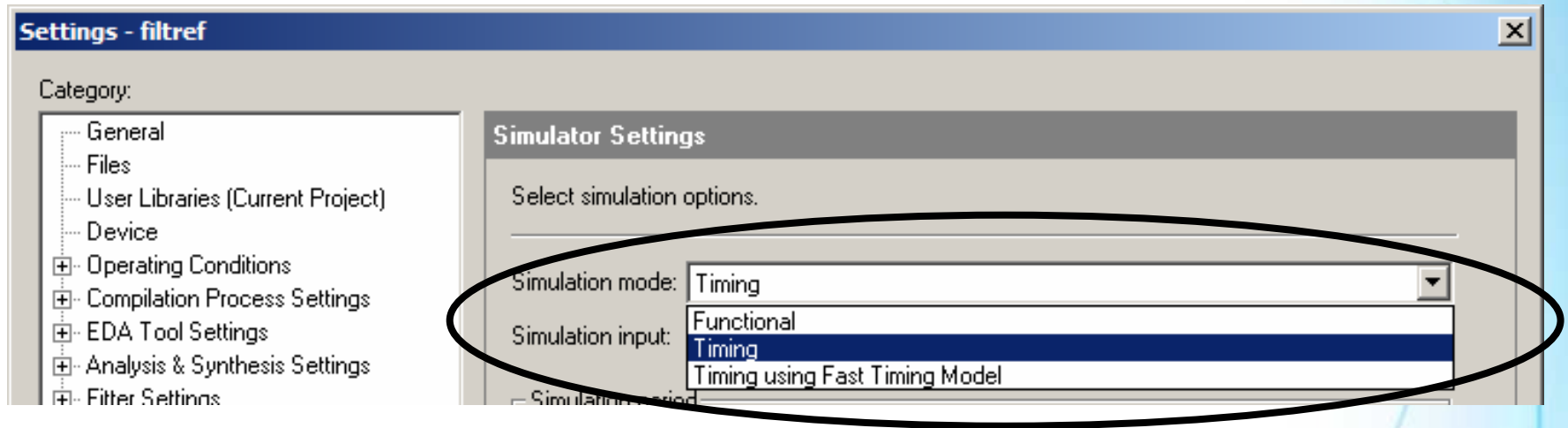
# Simulator Settings

- Mode
- Input file
- Period
- Options





# Simulator Modes



- **Functional**

- Type: RTL
- Uses pre-synthesis netlist

- **Timing**

- Type: gate-level or post-place & route
- Uses fully compiled netlist
- Uses worst-case timing model

- **Timing Using Fast Timing Model**

- Similar to Timing
- Uses Best-Case Timing Model

# Simulator Input & Period

- Specifies stimulus & length of simulation period

Select simulation options.

Mode: Timing

Stimulus file: filtref.clwvf ... Add multiple files...

Simulation period

Run simulation until all vector stimuli are used

End simulation at: [ ] ns

Annotations:

- Run Simulation until End of Stimulus File (points to the selected radio button)
- Specify Stimulus File(s) (points to the Add multiple files... button)
- Enter End Time (points to the empty input field)

# Simulator Options

The image shows a screenshot of the 'Simulator Options' dialog box. The dialog has a light gray background and contains several options with checkboxes. To the right of the dialog, five yellow callout boxes with black text point to specific settings:

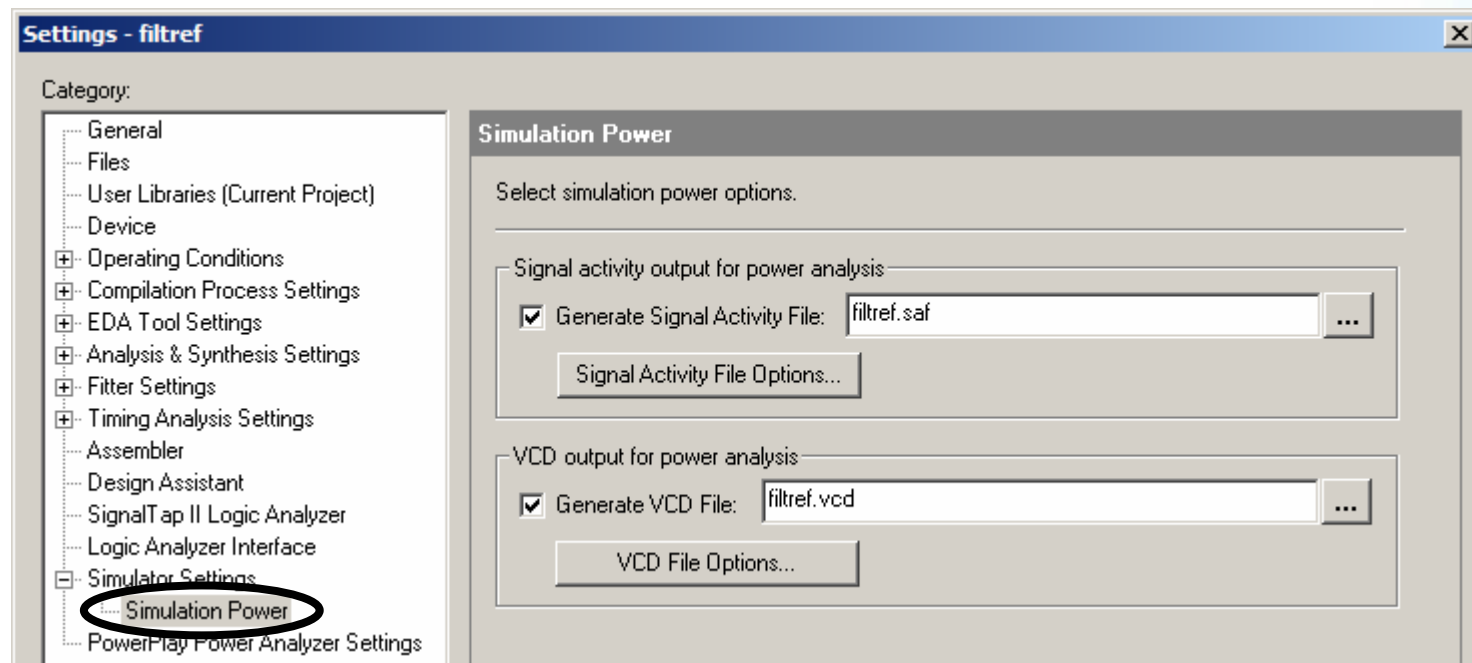
- Automatically Add Output Pins to Simulation:** Points to the checked option 'Automatically add pins to simulation output waveforms'.
- Compares Simulation Outputs to Outputs in Stimulus File:** Points to the 'Waveform Comparison Settings...' button.
- Monitors & Reports Simulation for Glitches:** Points to the checked option 'Simulation coverage reporting' and its 'Report Settings...' button.
- Reports Setup & Hold Violations:** Points to the 'More Settings...' button.
- Reports Toggle Ratio:** Points to the 'Description' text area.

The dialog options include:

- Automatically add pins to simulation output waveforms
- Check outputs [Waveform Comparison Settings...](#)
- Setup and hold time violation detection
- Glitch detection:
- Simulation coverage reporting [Report Settings...](#)
- Overwrite simulation input file with simulation results
- Disable setup and hold time violation detection for input registers of bi-directional pins
- [More Settings...](#)
- Description:**  
Specifies the source of input vectors to be used for incremental input simulation.

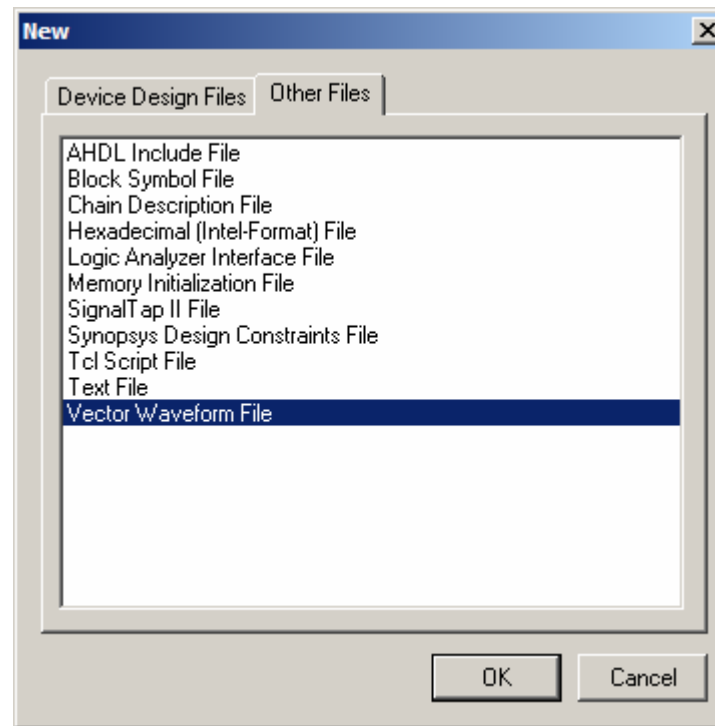
# Generate SAF

- Generate SAF or VCD file for power calculation

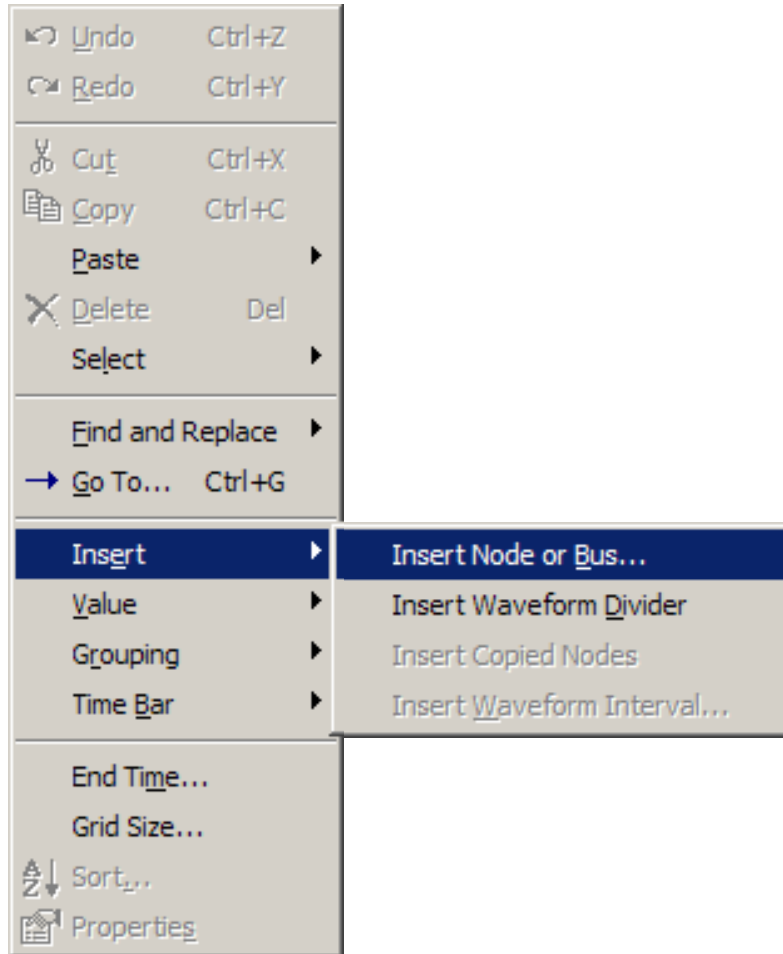


# Create New Vector Waveform File

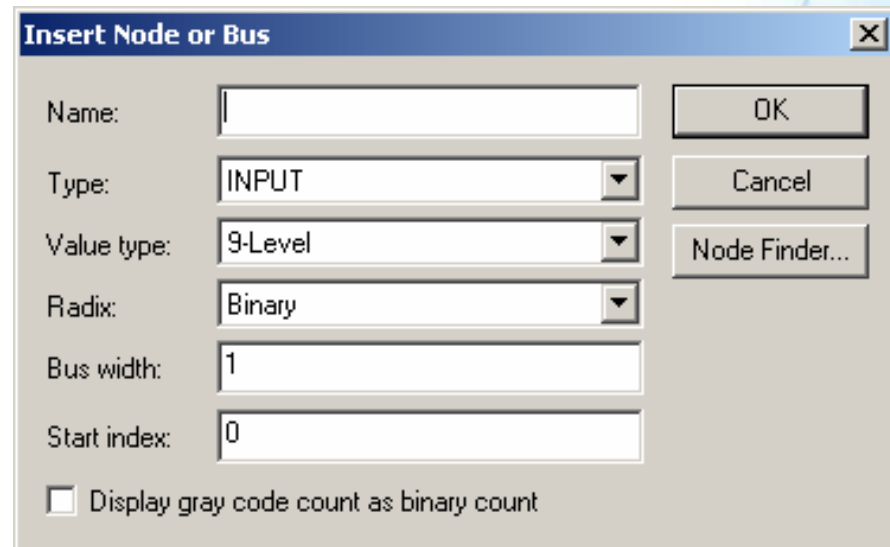
- Select **File** ⇒ **New** ⇒ **Vector Waveform File** (Other Files tab)



# Insert Nodes

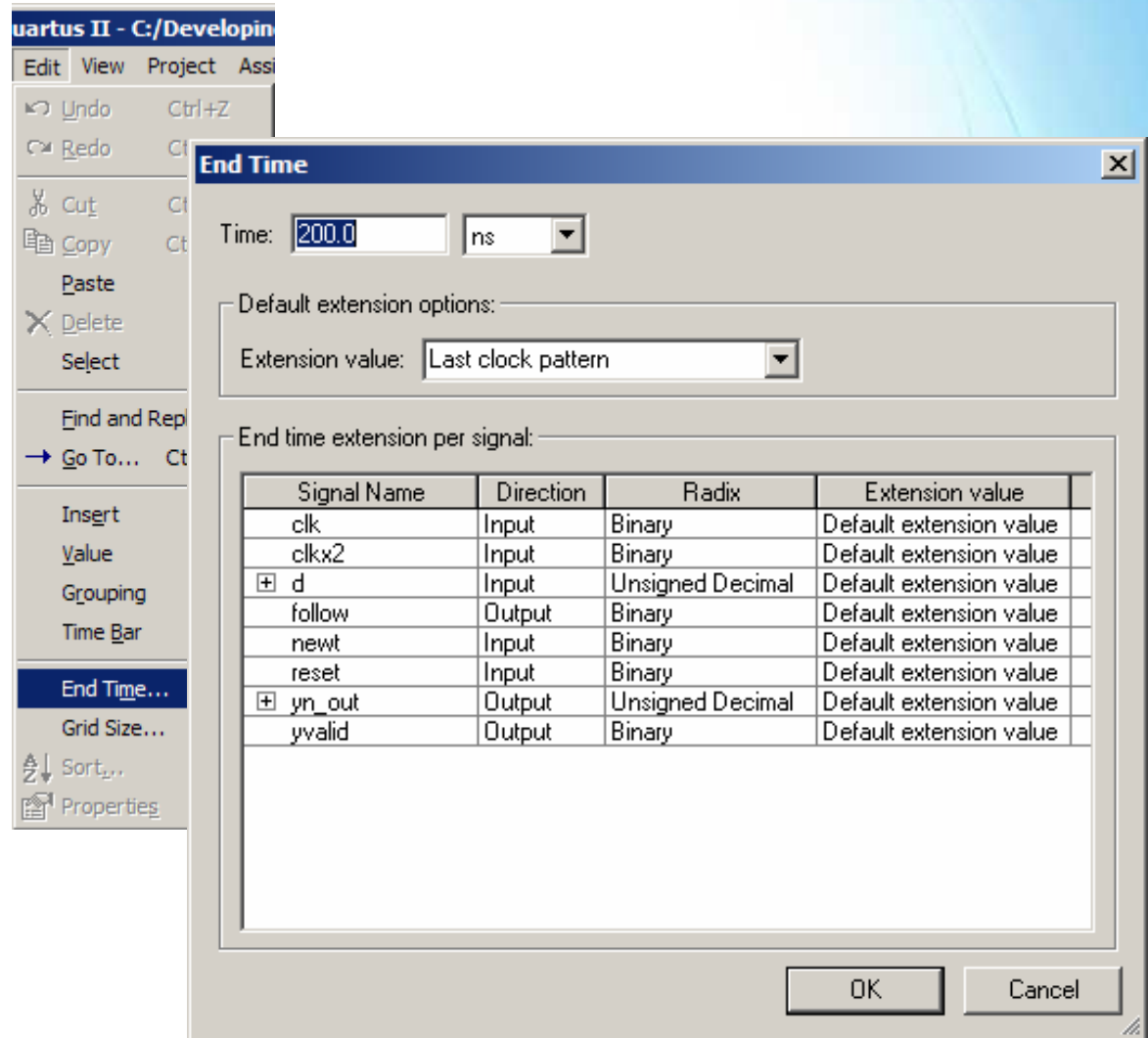


- Select **Insert Node or Bus** (Edit menu)
  - VWF must be open
  - Use node finder



# Specify End Time

- Maximum length of simulation time
  - Edit menu
- Specify how to extend signal values



# Insert Time Bars

- Set one time bar as master
- Insert other time bars
  - Relative to master
  - Absolute

Specify Time Bar

Time Bar Organizer

Time bar

Time: 90 ns

Interpret time as

Absolute time

Relative to master time bar

Existing time bars:

Absolute time	Relative time
M 20.0 ns	0 ps
50.0 ns	30.0 ns
90.0 ns	70.0 ns

Set as Master

Time Bar

Set Master Time Bar



# Draw Stimulus Waveform

- Highlight portion of waveform to change
- Overwrite value with desired value

The screenshot shows the Quartus II waveform editor. The 'Value' menu is open, showing various options for setting signal values. The 'High Impedance (Z)' option is highlighted. A portion of the waveform is selected, and a toolbar on the right contains various editing tools and shortcuts.

Value	Shortcut
Uninitialized (U)	Ctrl+Alt+U
Forcing Unknown (X)	Ctrl+Alt+X
Forcing Low (0)	Ctrl+Alt+0
Forcing High (1)	Ctrl+Alt+1
High Impedance (Z)	Ctrl+Alt+Z
Weak Unknown (W)	Ctrl+Alt+W
Weak Low (L)	Ctrl+Alt+L
Weak High (H)	Ctrl+Alt+H
Don't Care (DC)	Ctrl+Alt+D
Invert	Ctrl+Alt+I
Count Value...	Ctrl+Alt+V
Clock...	Ctrl+Alt+K
Arbitrary Value...	Ctrl+Alt+B
Random Values...	Ctrl+Alt+R

# Overwrite Waveform Signal Values

- 1 = forcing '1'
- 0 = forcing '0'
- X = forcing unknown
- U = uninitialized
- Z = high impedance
- H = weak '1'
- L = weak '0'
- W = weak unknown
- DC = don't care

# Overwrite Waveform Patterns

## ■ Clock

- Enter period & duty cycle

**Time range**

Start time: 0 ps

End time: 10.0 ns

**Base waveform on**

Clock settings:

clk

Time period:

Period: 10.0 ns

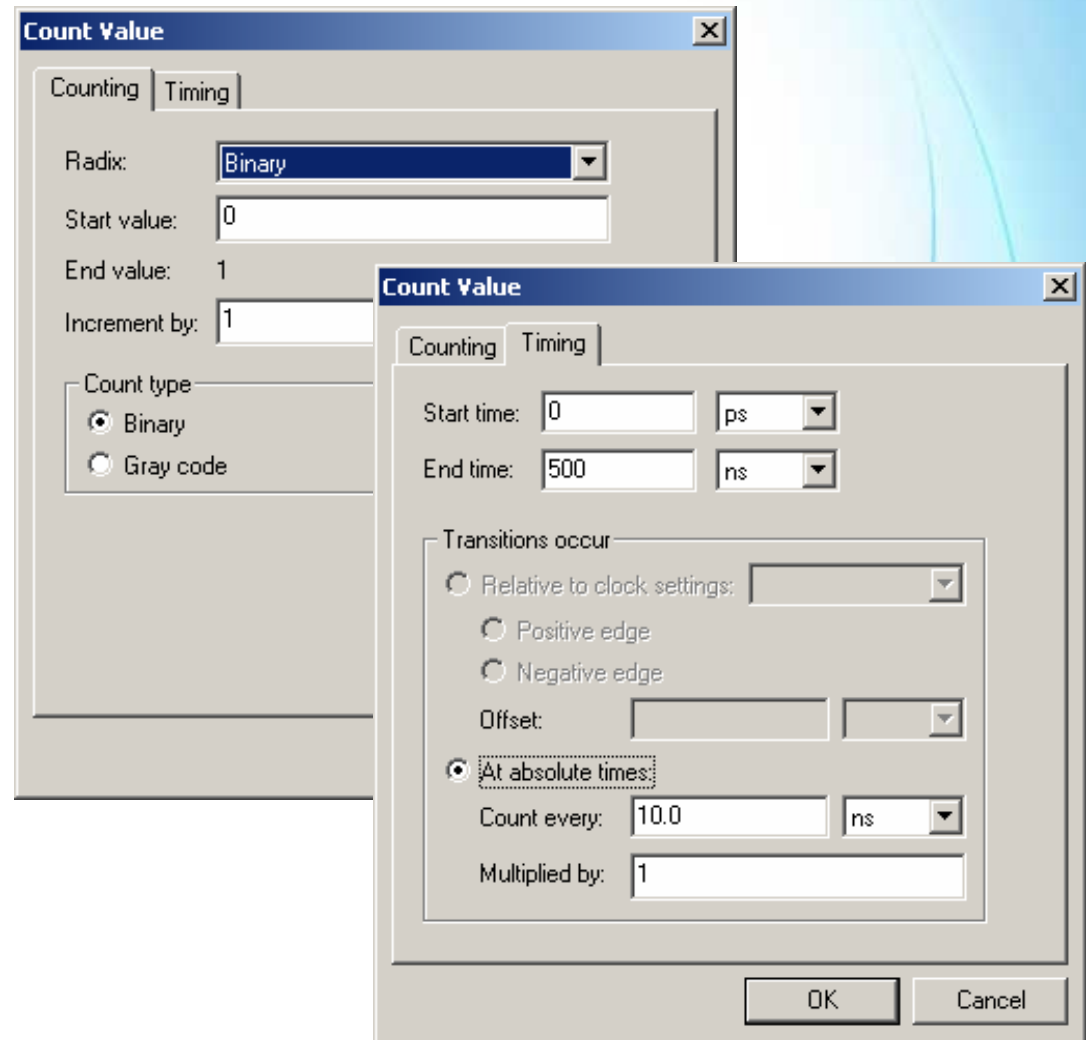
Offset: 0.0 ns

Duty cycle (%): 50

OK Cancel

# Overwrite Waveform Patterns (cont.)

- Counting pattern
  - Enter count timing
  - Enter start value & increment

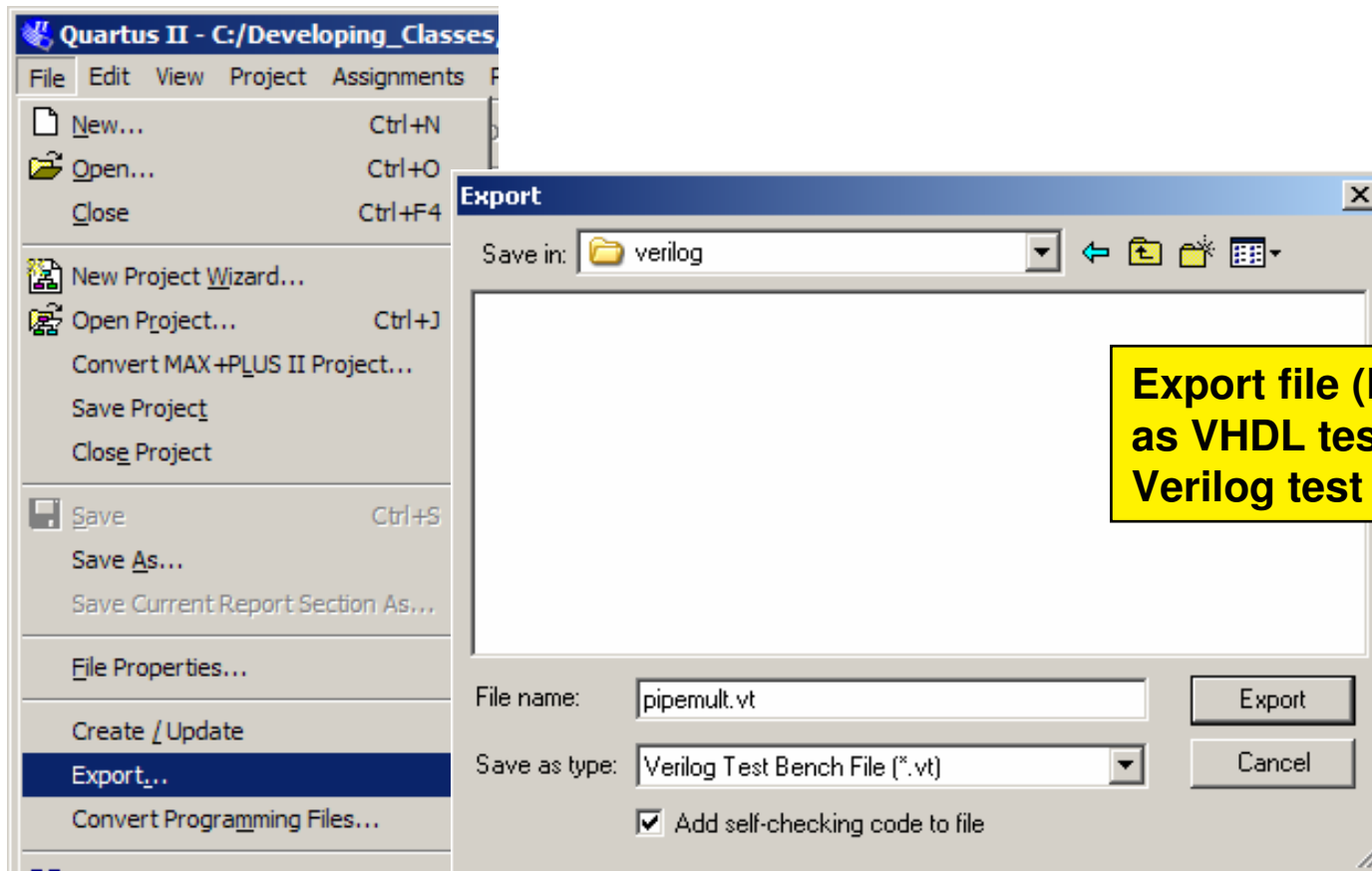


# Other Waveform Patterns

- Arbitrary Value
  - Group value for busses
- Random Value

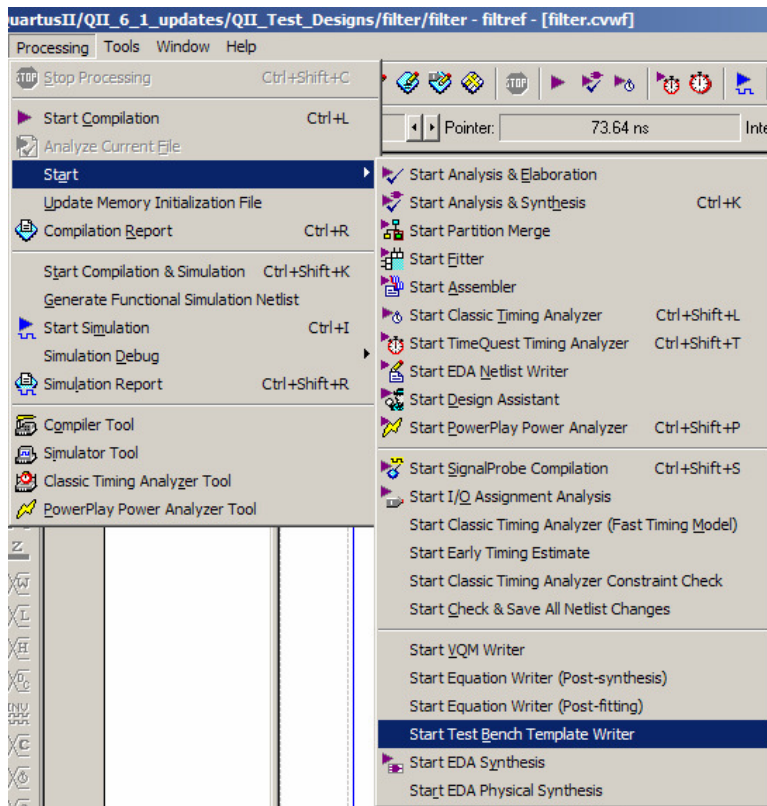
# Waveform to Testbench Generator

- Converts VWF into HDL testbench



# Testbench Template Generator

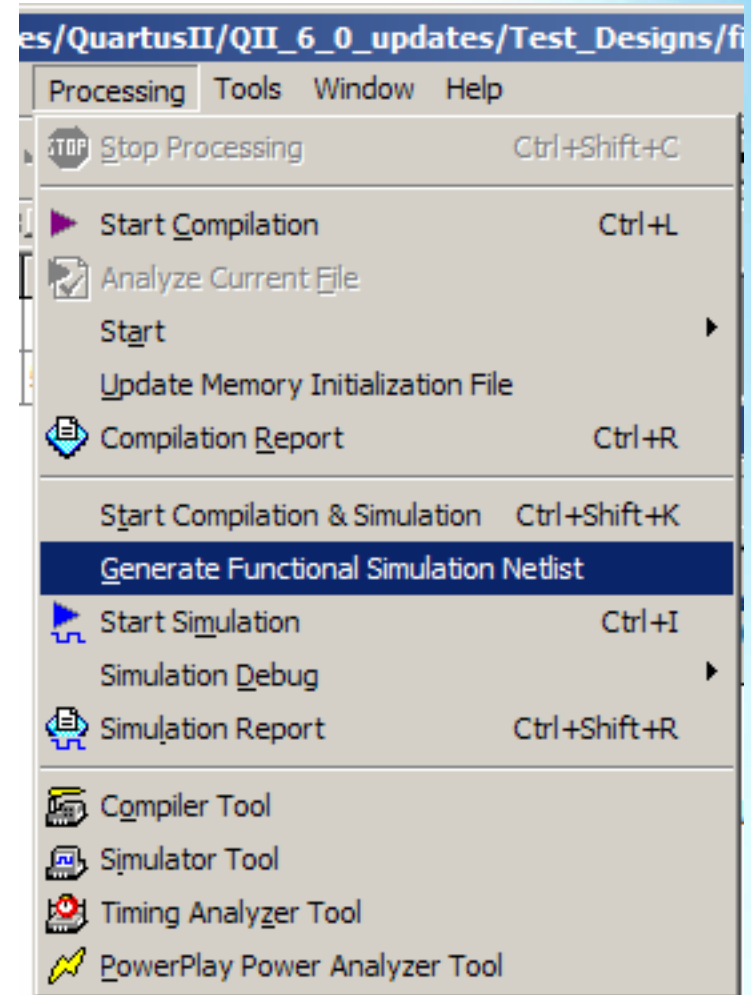
- Generates HDL testbench template
  - User inserts test stimulus



```
simulation\modelsim\dac_demo_ver.vt
77
78 // assign statements (if any)
79 assign {t_wire_indata622,t_wire_in_deskew,t_wire_in_clock,
80 dac_demo_ver tb (
81 // port map - connection between master ports and signals/regist
82 .indata622(t_wire_indata622),.in_deskew(t_wire_in_deskew),.i
83 initial
84 // code that executes only once
85 // insert code here --> begin
86
87 // --> end
88 $display("Running testbench");
89 end
90 always(
91 // optional sensitivity list
92 // @(event1 or event2 or .... eventn)
93 begin
94 // code executes for every event on sensitivity list
95 // insert code here --> begin
96
97 @eachvec;
98 // --> end
99 end
100 endmodule
101
```

# Before Functional Simulation

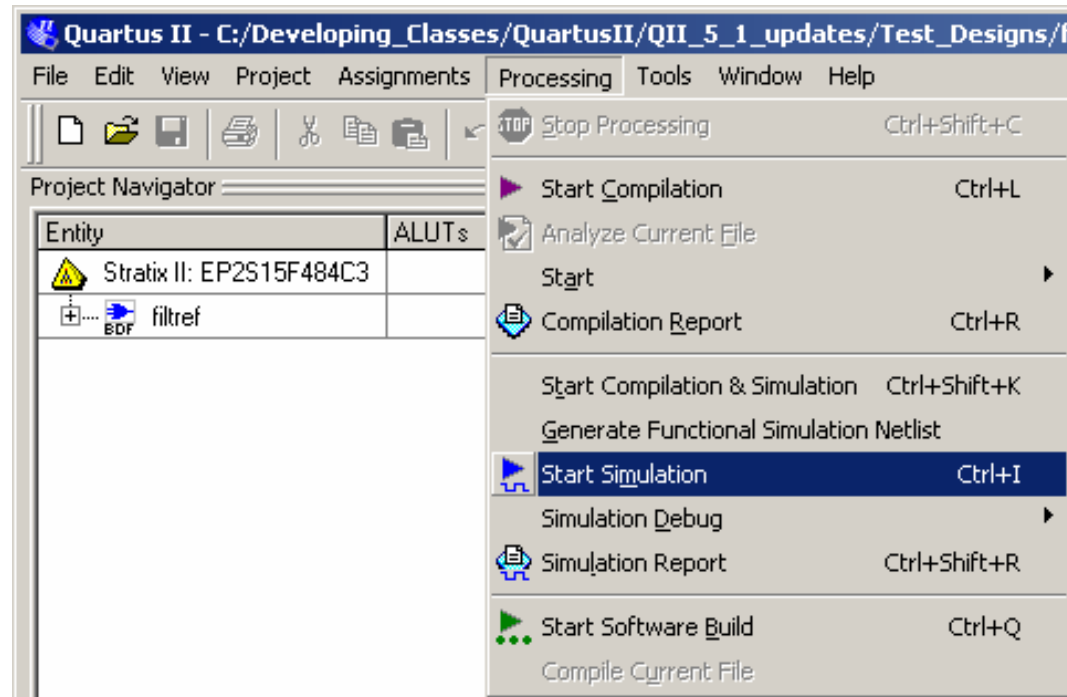
- Perform **Generate Functional Simulation Netlist** (Processing menu)
  - Creates pre-synthesis netlist
  - Fails simulation if not performed





# Starting Simulation

- Processing menu ⇒ Start Simulation

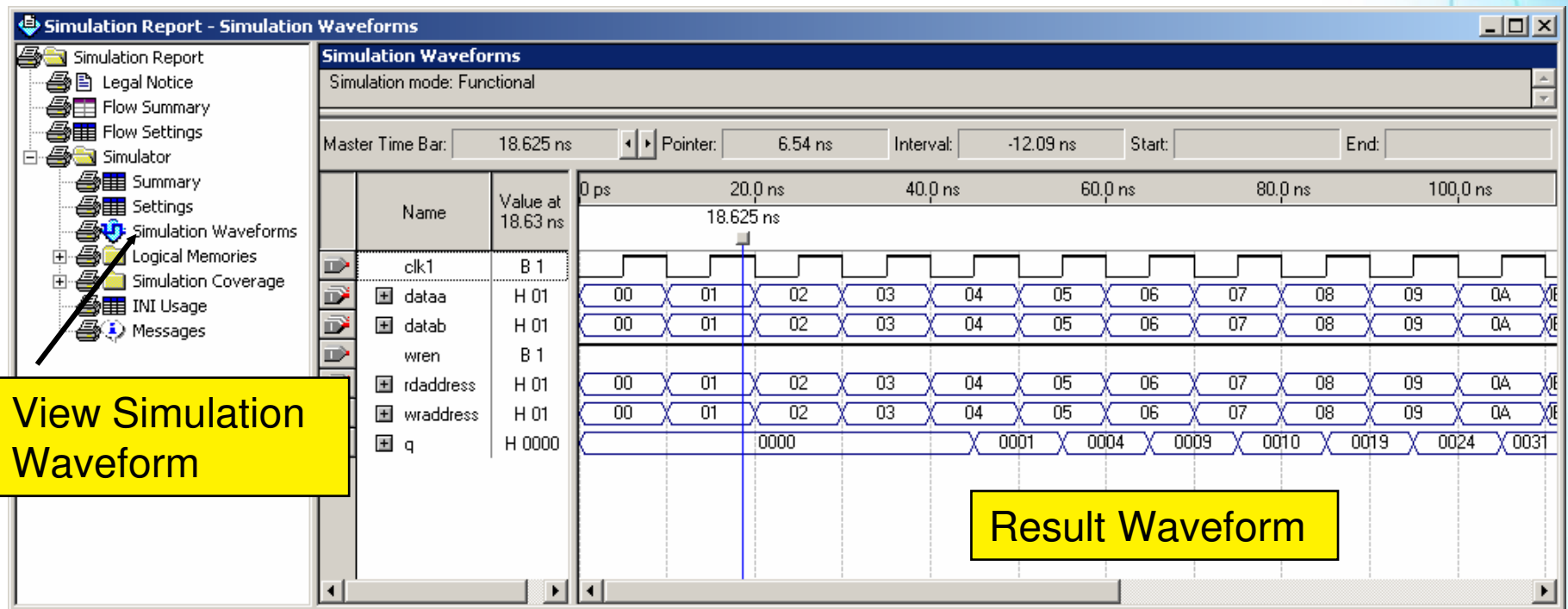


- Scripting

# Simulation Report



- Displays simulation result waveform

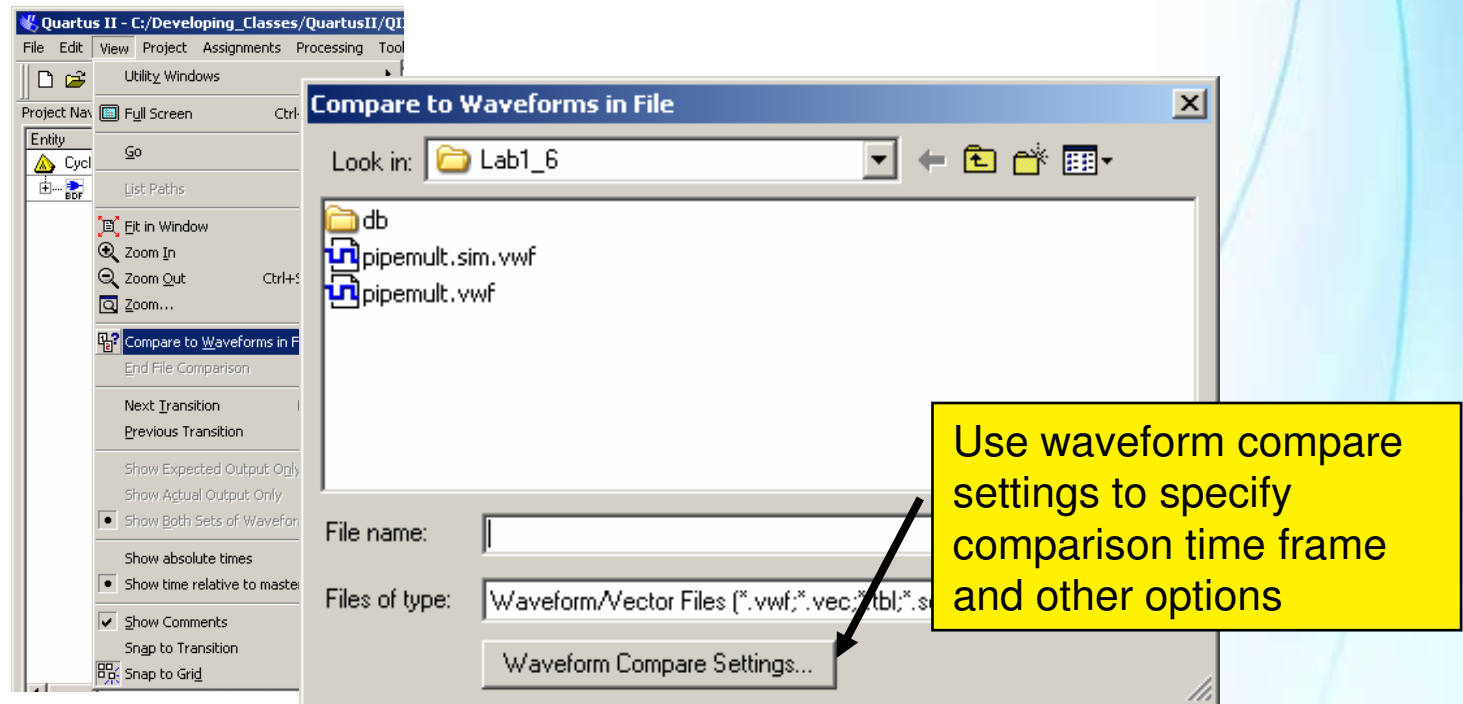


View Simulation Waveform

Result Waveform

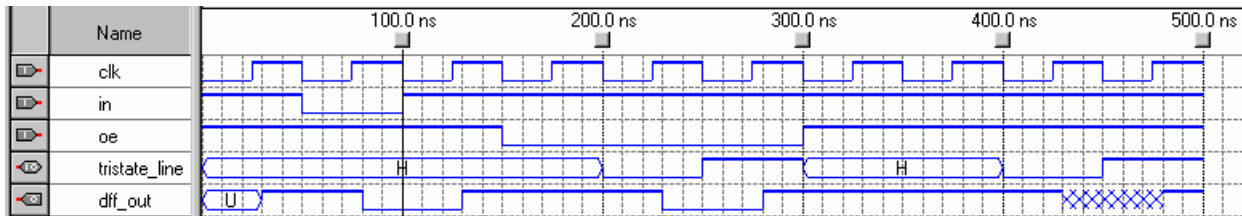
# Comparing Waveforms

- Select Compare to Waveforms (View menu)
  - Simulation waveform must be open
- Select VWF comparison file

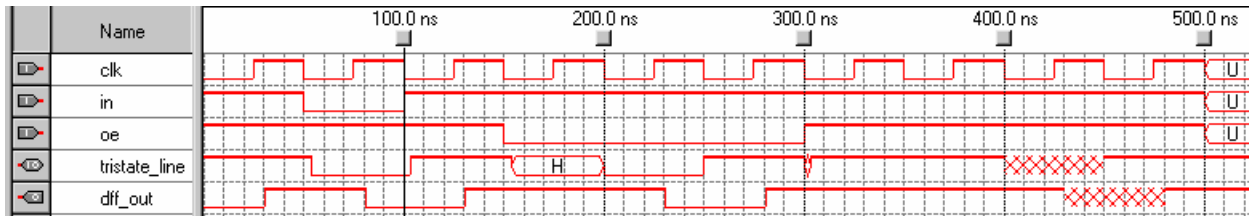


# Compared Waveforms (Simulator Report)

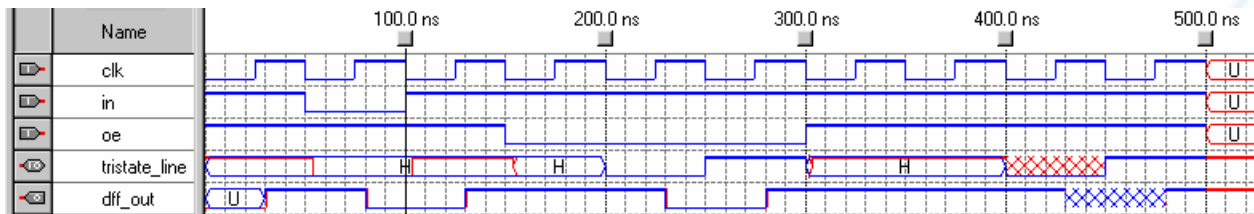
## ■ Original waveforms (ctrl+1)



## ■ Compared file waveforms (ctrl+2)

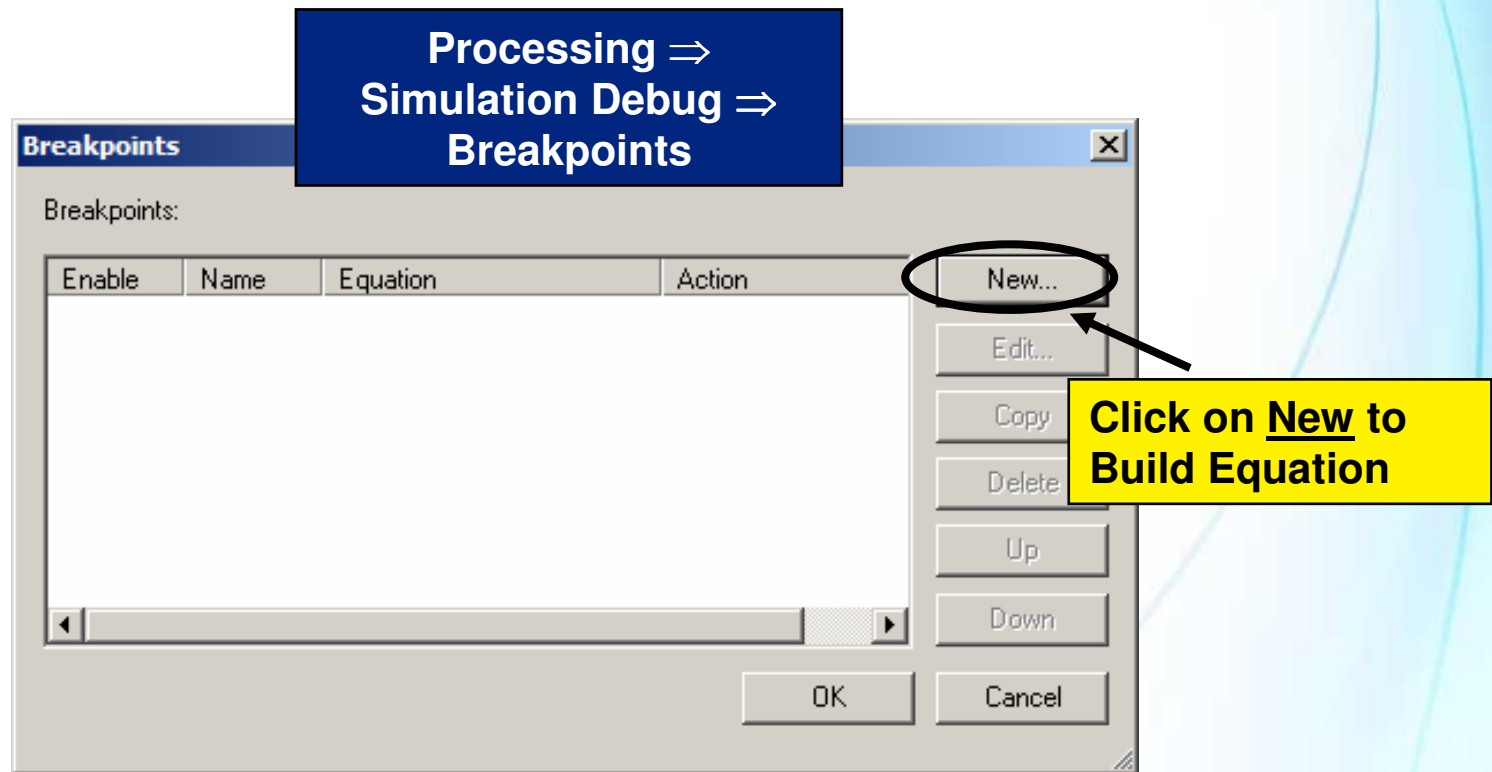


## ■ Both sets of waveforms (ctrl+3)



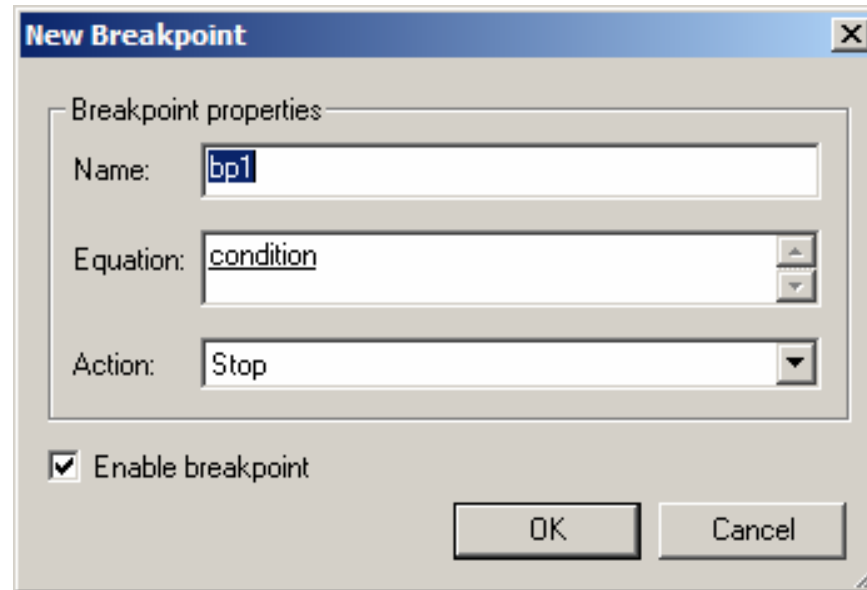
# Breakpoints

- Interrupts simulation at specified points



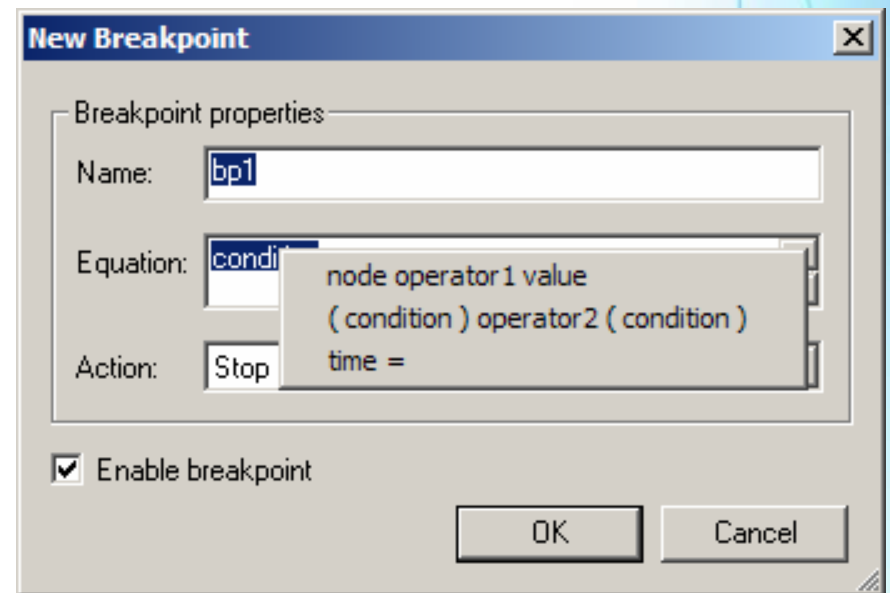
# Breakpoints

- Consists of 3 parts
  - Name
  - Equation (condition)
  - Action

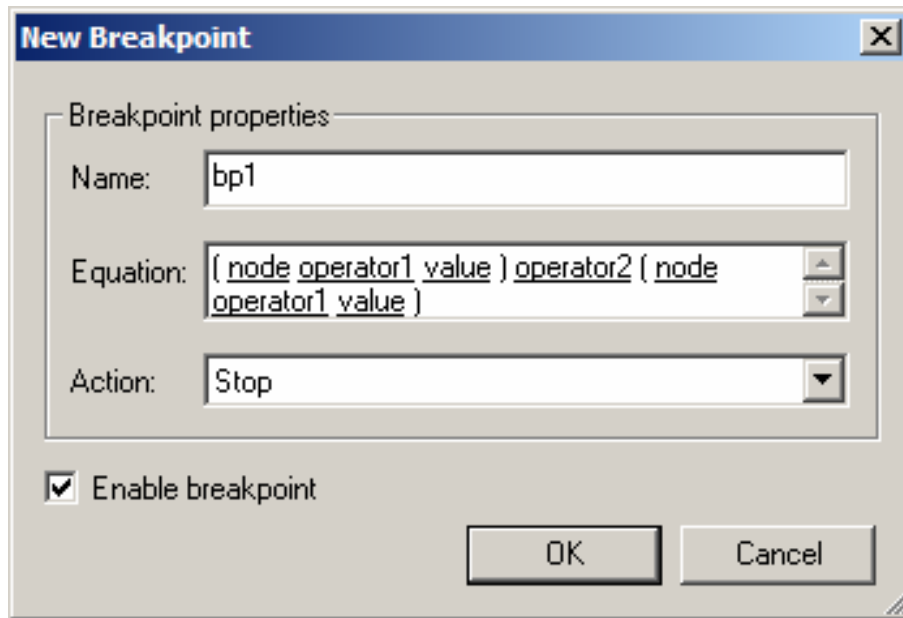


# Breakpoint Conditions

- `<Node> <operator1> <value>`
  - Single condition
  - Ex. `Ena = 1`
- `Time = <value>`
  - Single condition
  - Time = 500ns
- `<Condition> <operator2>`  
`<condition>`
  - Complex tests
  - `Ena = 1 && time > 500ns`



# Breakpoint Equations (Cont.)



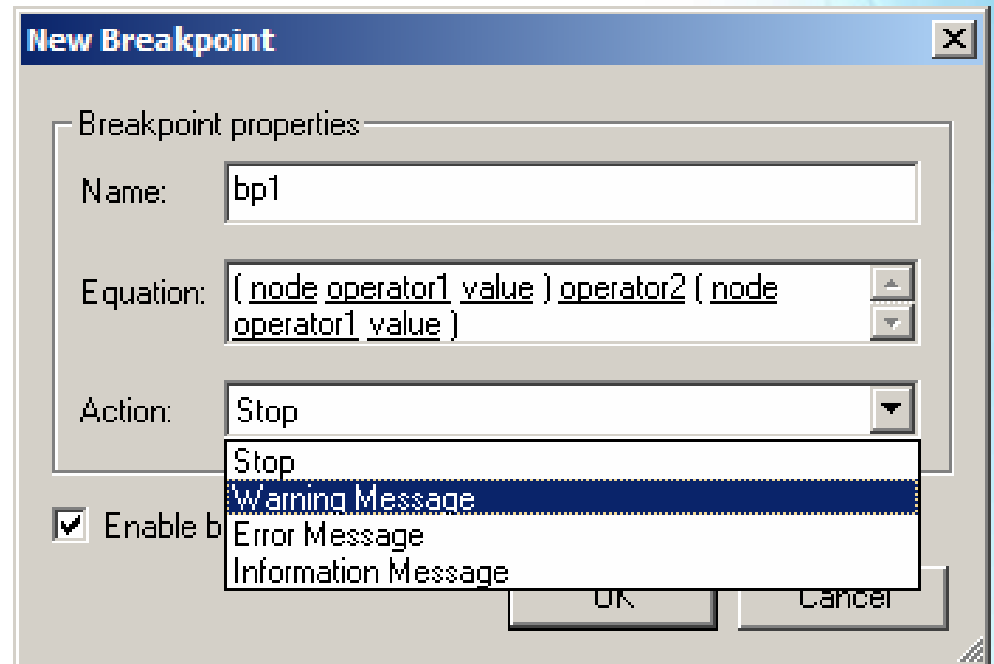
## ■ Clicking on:

- Node
  - Opens node finder
- Operator1
  - Allows selection of <, >, =
- Operator2
  - Allows selection of && (AND) and || (OR) operators

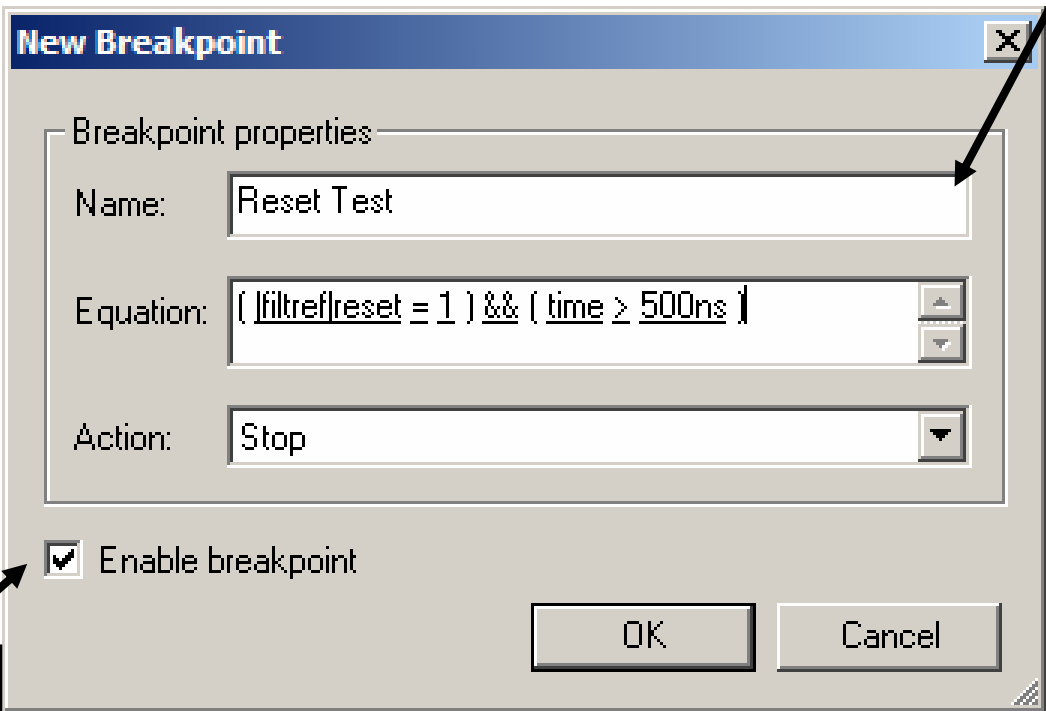


# Breakpoint Actions

- Stop
- Give error message
- Give warning message
- Give informational message



# Example Breakpoint



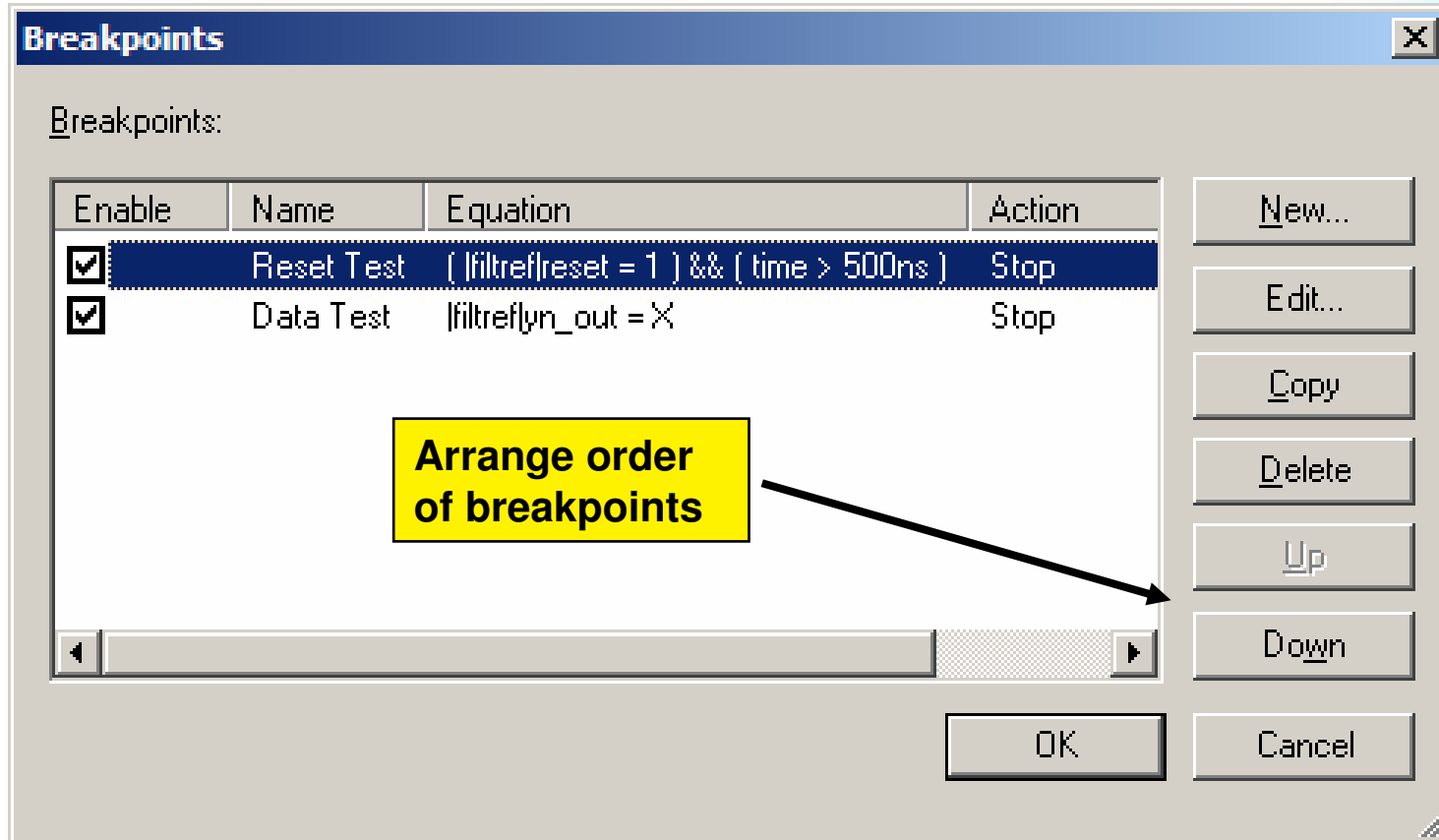
The image shows a 'New Breakpoint' dialog box with the following fields and controls:

- Name:** Reset Test
- Equation:** ( !filref|reset = 1 ) && ( time > 500ns )
- Action:** Stop
- Enable breakpoint
- Buttons: OK, Cancel

Annotations:

- A yellow box labeled 'Name breakpoint' with an arrow pointing to the 'Name' field.
- A yellow box labeled 'Enable/disable breakpoints' with an arrow pointing to the 'Enable breakpoint' checkbox.

# Organizing Breakpoints



*Please go to optional exercise in the  
Exercise Manual*