20 YEARS *of*

ALTERA®

INNOVATION

Designing with Quartus II

QUARTUS® II

Class Agenda

- Design Methodology in Quartus® II
  - Exercise 1
- Projects
- Compilation
  - Exercise 2
- Single & Multi-Clock Timing Analysis
  - Exercise 3 &4
- Simulation
  - Exercise 5
- Programming/Configuration

20 YEARS *of*

ALTERA®

INNOVATION

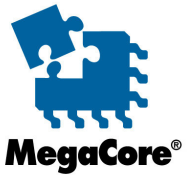
Designing with Quartus II

Introduction to Altera & Altera Devices

QUARTUS® II

The Programmable Solutions Company®

- Programmable Devices
- Design Software
- Intellectual Property (IP)



Introduction to Altera Devices

- Programmable Logic Families
  - High & Medium Density FPGAs
    - Stratix™ II, Stratix, APEX™ II, APEX 20K, & FLEX® 10K
  - Low-Cost FPGAs
    - Cyclone™ & ACEX® 1K
  - FPGAs with Clock Data Recovery
    - Stratix GX & Mercury™
  - CPLDs
    - MAX® 7000 & MAX 3000
  - Embedded Processor Solutions
    - Nios™, Excalibur™
  - Configuration Devices
    - EPC

Introduction to Altera Design Software

- Software & Development Tools:
  - Quartus II
    - Stratix II, Stratix, Stratix GX, Cyclone, APEX II, APEX 20K/E/C, Excalibur, & Mercury Devices
    - FLEX 10K/A/E, ACEX 1K, FLEX 6000, MAX 7000S/AE/B, MAX 3000A Devices
  - Quartus II Web Edition
    - Free Version
    - Not All Features & Devices Included
  - MAX+PLUS® II
    - All FLEX, ACEX, & MAX Devices

20 YEARS *of*

ALTERA®

INNOVATION

Designing with Quartus II

Quartus II Development System
Feature Overview

QUARTUS® II

Quartus II Development System

- Fully-Integrated Design Tool
  - Multiple Design Entry Methods
  - Logic Synthesis
  - Place & Route
  - Simulation
  - Timing & Power Analysis
  - Device Programming

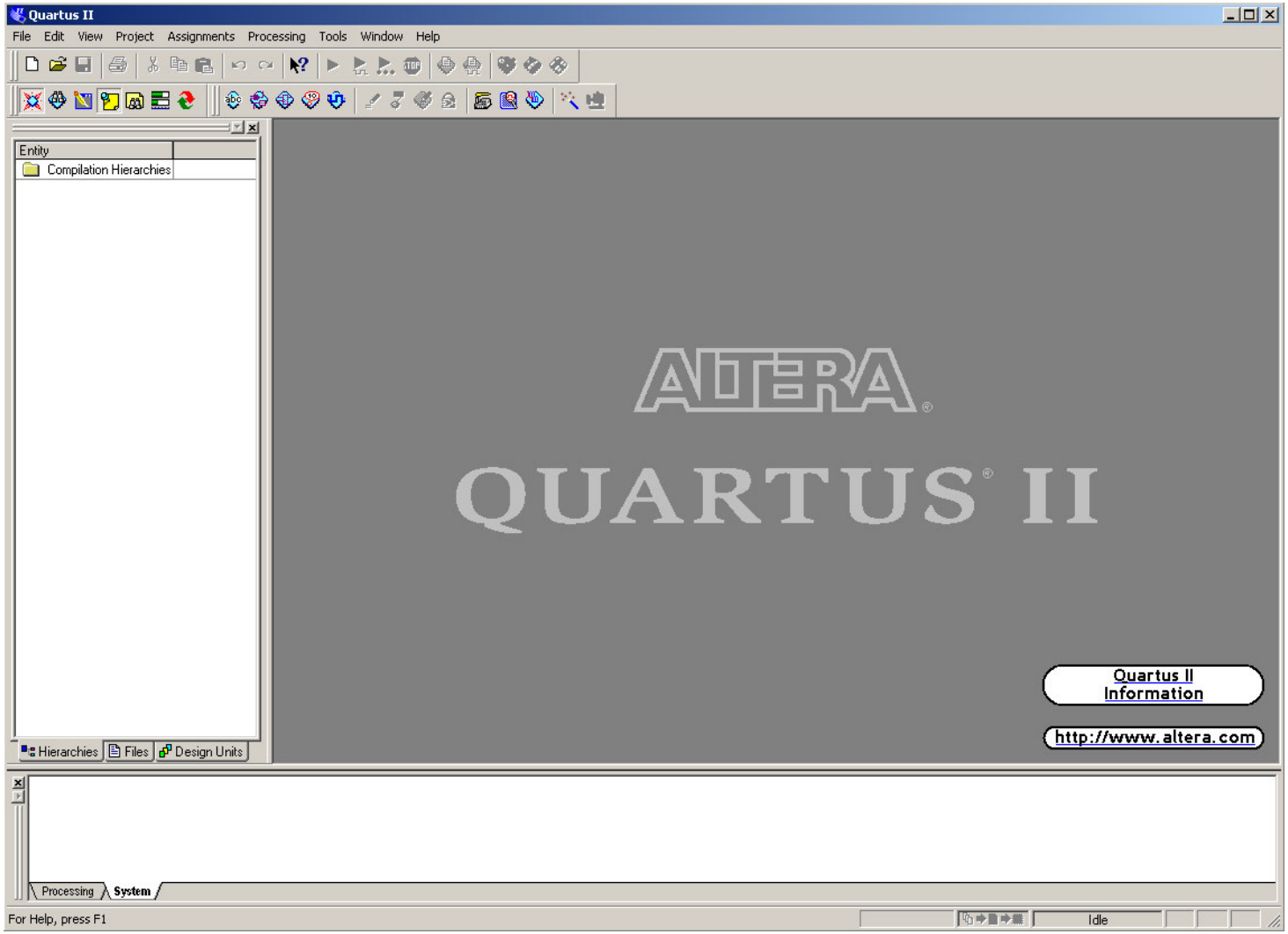
More Features

- MegaWizard® & SOPC Builder Design Tools
- LogicLock™ Optimization Tool
- NativeLink® 3rd-Party EDA Tool Integration
- Integrated Embedded Software Development
- SignalTap® II & SignalProbe™ Debug Tools
- Windows, Solaris, HPUX, & Linux Support
- Node-Locked & Network Licensing Options
- Revision Control Interface

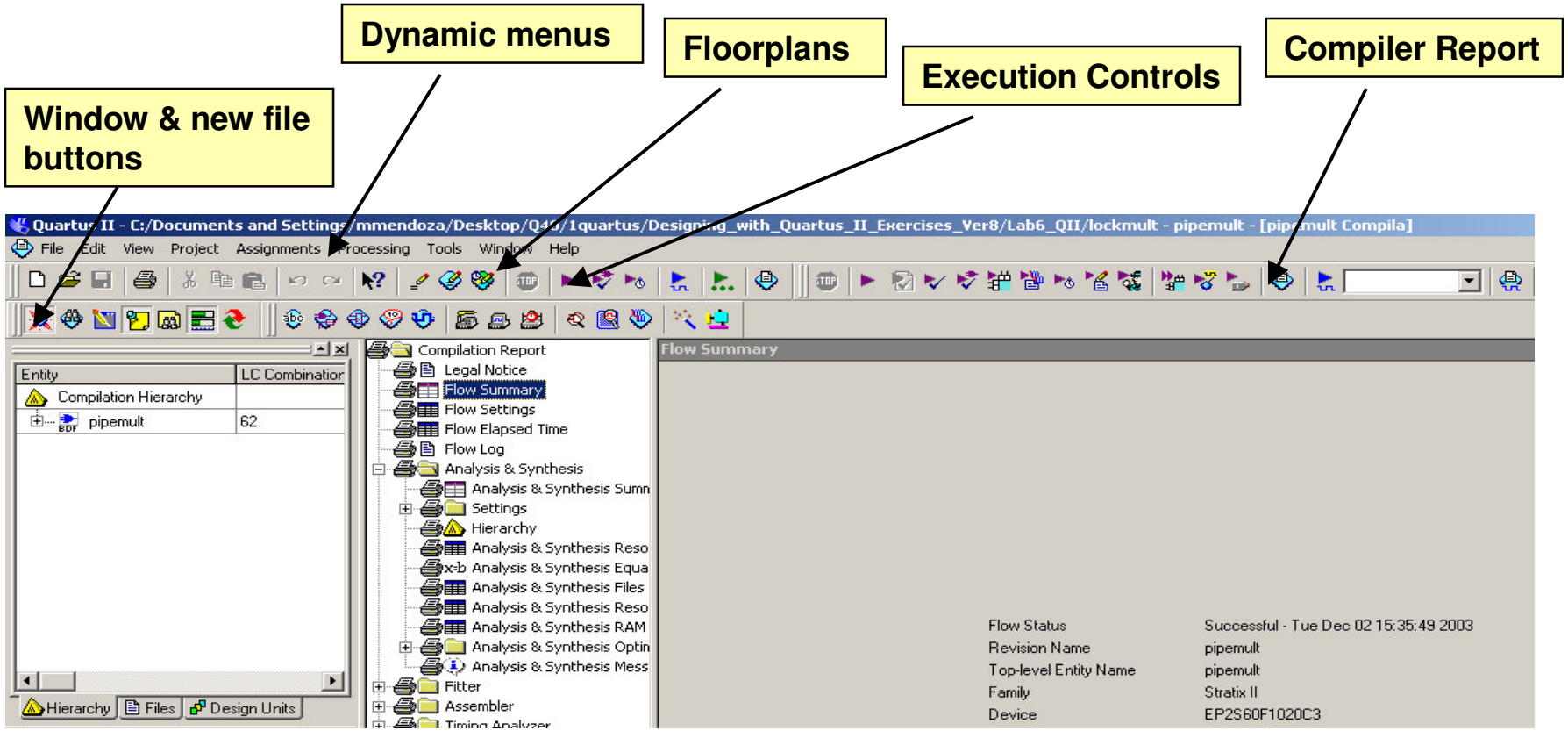
What's New in Quartus II 4.0

- SOPC Builder
- LogicLock™ Block-Based Methodology
- Assignment Editor
- Creating & Validating I/O Assignments
- IP Cores
- Integrated Synthesis
- Scripting
- Block Design Entry & Documentation
- MAX+PLUS® II Project Conversion

Quartus II Operating Environment



Main Toolbar & Modes



To Reset Views: Tools ⇒Toolbars>Reset All; Restart Quartus II

20 YEARS *of*

ALTERA®

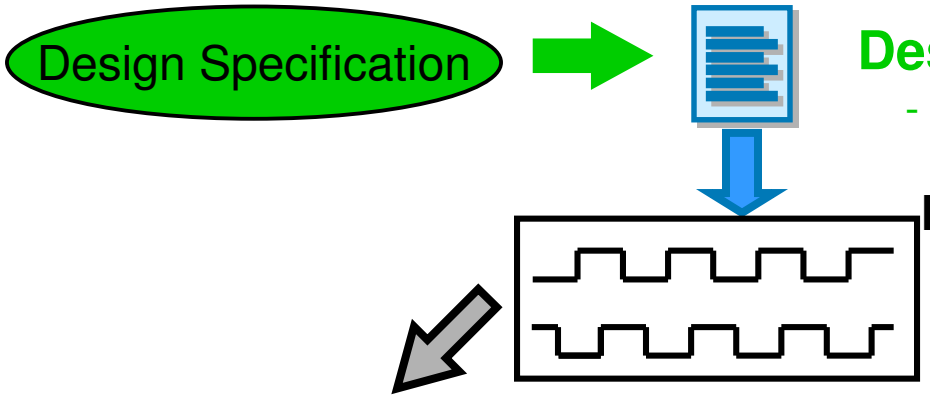
INNOVATION

Designing with Quartus II

Design Methodology

QUARTUS® II

PLD Design Flow

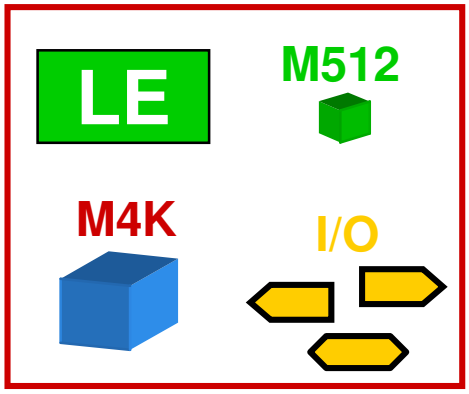


Design Entry/RTL Coding

- Behavioral or Structural Description of Design

RTL Simulation

- Functional Simulation (Modelsim®, Quartus II)
- Verify Logic Model & Data Flow (No Timing Delays)



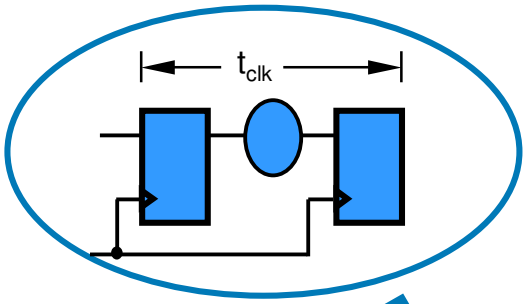
Synthesis

- Translate Design into Device Specific Primitives
- Optimization to Meet Required Area & Performance Constraints
- Precision, Synplify, Quartus II

Place & Route

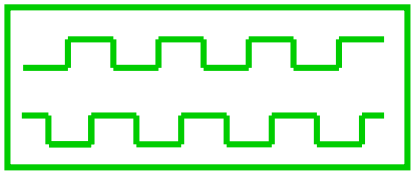
- Map Primitives to Specific Locations inside Target Technology with Reference to Area & Performance Constraints
- Specify Routing Resources to Be Used

PLD Design Flow



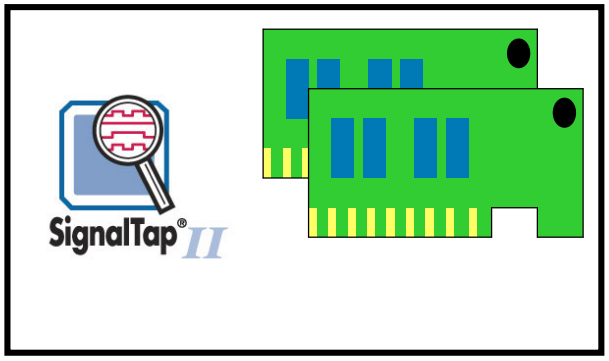
Timing Analysis

- 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** for Debugging

20 YEARS *of*

ALTERA®

INNOVATION

Designing with Quartus II

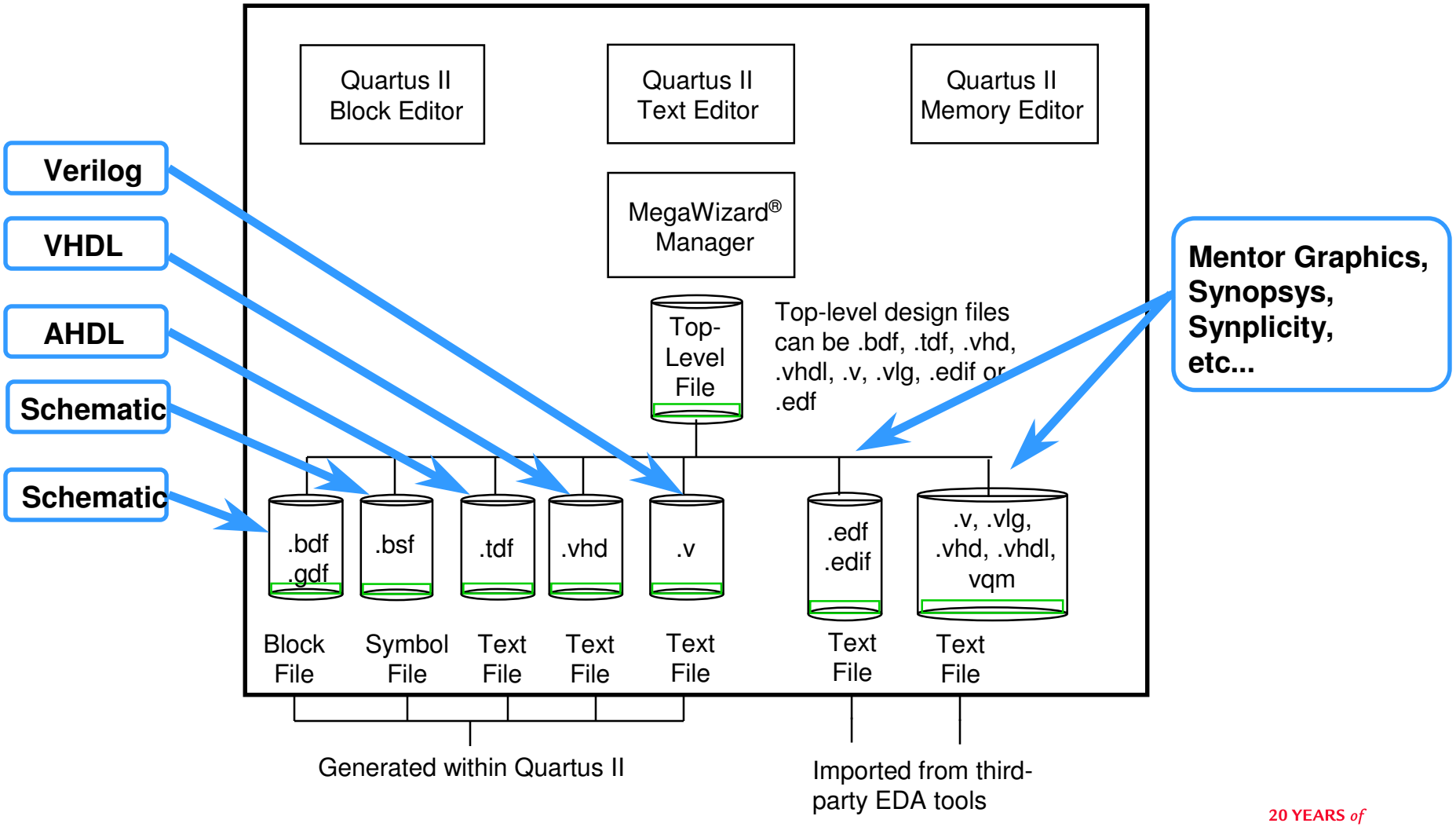
Design Entry

QUARTUS® II

Design Entry Methods

- Quartus II
  - Text Editor
    - AHDL, VHDL, Verilog
  - Memory Editor
    - HEX, MIF
  - Schematic Design Entry
- 3rd-Party EDA Tools
  - EDIF
  - HDL
  - VQM
- Mixing & Matching Design Files Allowed

Design Entry Files



Text Design Entry

- Available Features
  - Line Numbering in the HDL Text Files
  - Preview of HDL Templates
  - Syntax Coloring
  - When Editing a Text File, an Asterisk (*) Appears Next to the Filename
    - Asterisk Disappears after Saving the File
- Enter Text Description
  - AHDL (.TDF)
  - VHDL (.VHD)
  - Verilog (.V)

Verilog & VHDL

- **VHDL**- VHSIC Hardware Description Language
  - 1987 & 1993 IEEE 1074 Standards Supported
- **Verilog** - 1995 & 2001 IEEE 1364 Standard HDL

- Create in Quartus II or any Standard Text Editor
- Use Quartus II Integrated Synthesis to Synthesize
- View Supported Commands in On-Line Help

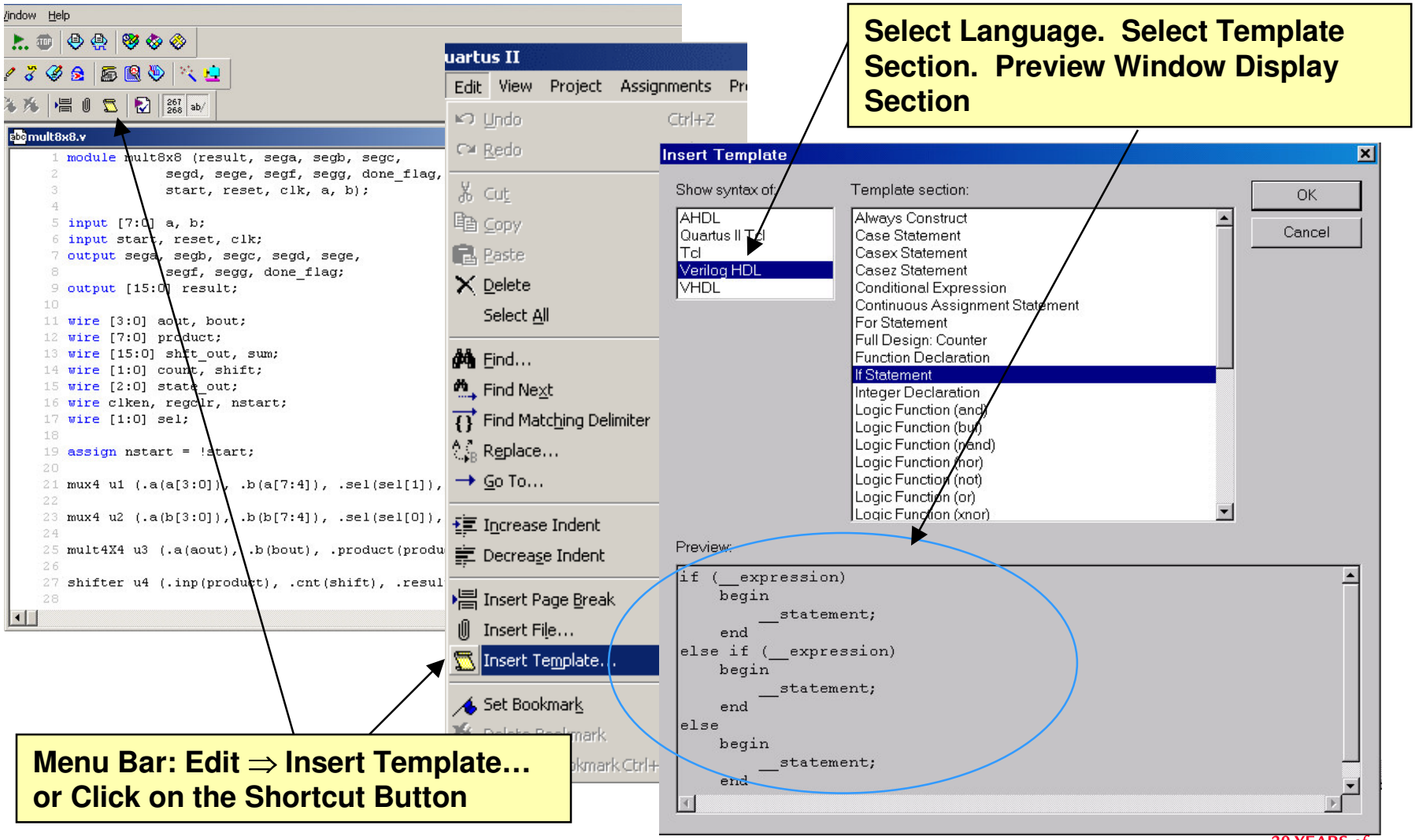
Learn more about HDL in Altera HDL Customer Training Classes

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.

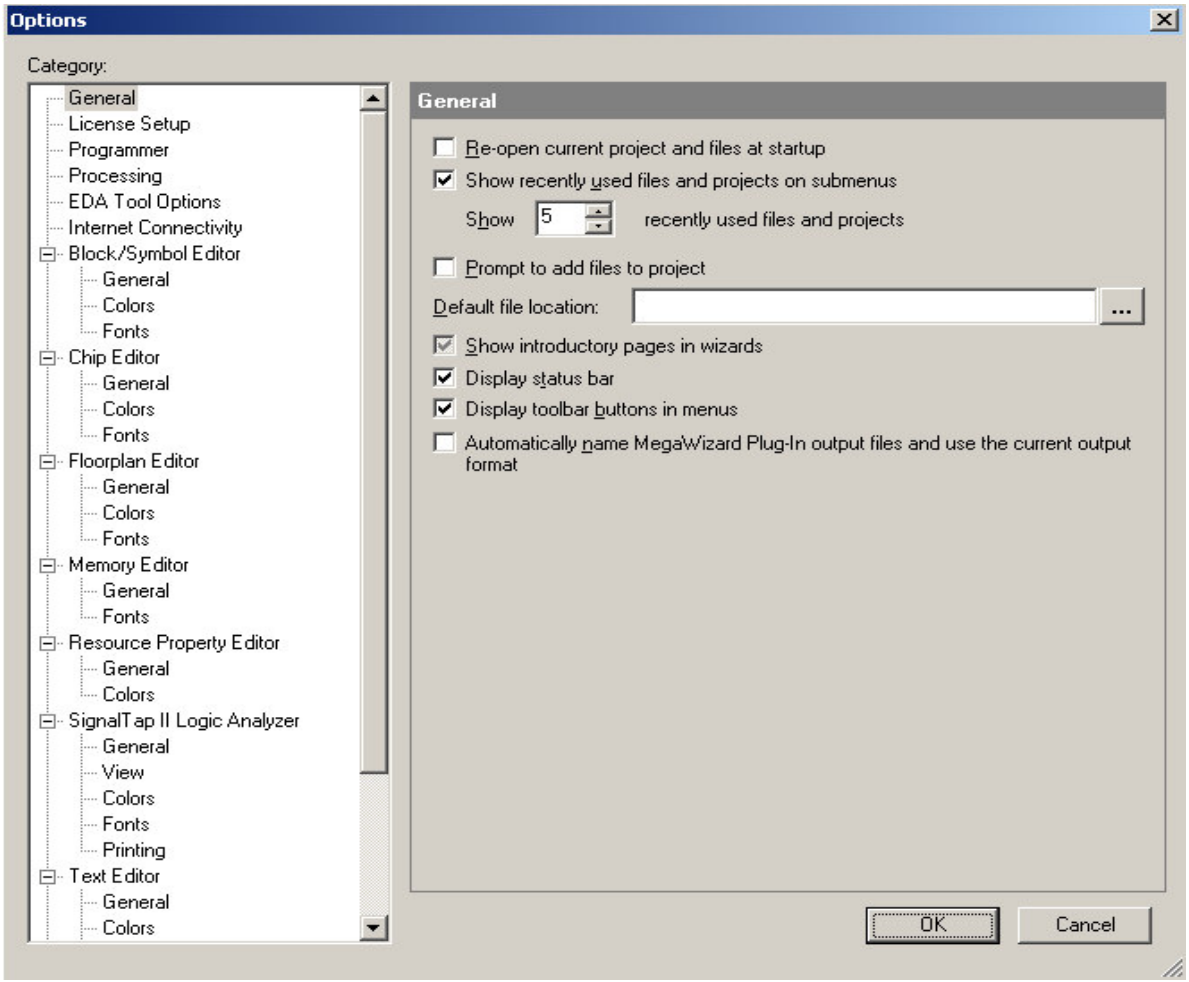
- Create in Quartus II or any Standard Text Editor

HDL Templates



Quartus II Options

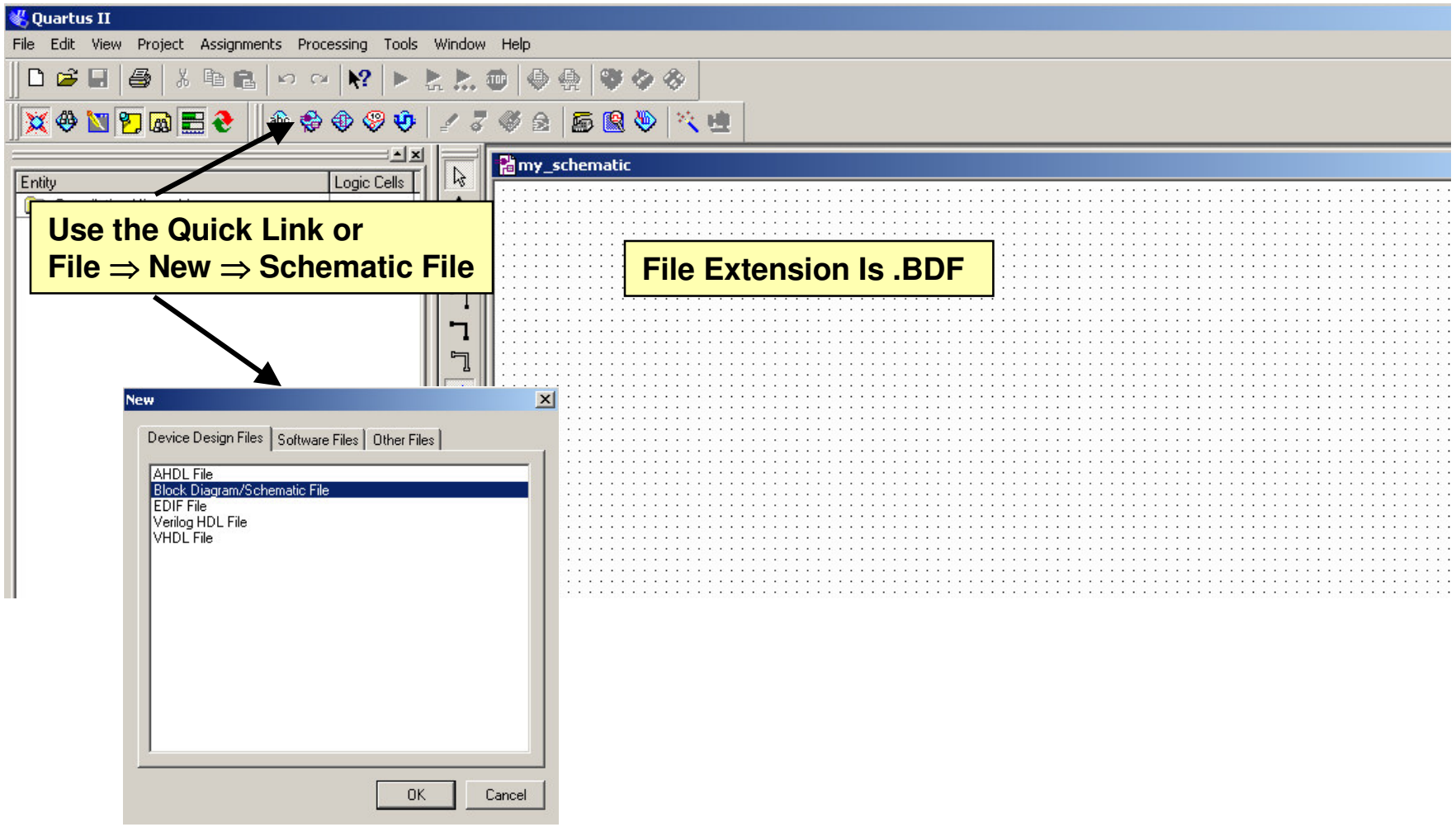
Tools ⇒ Options...



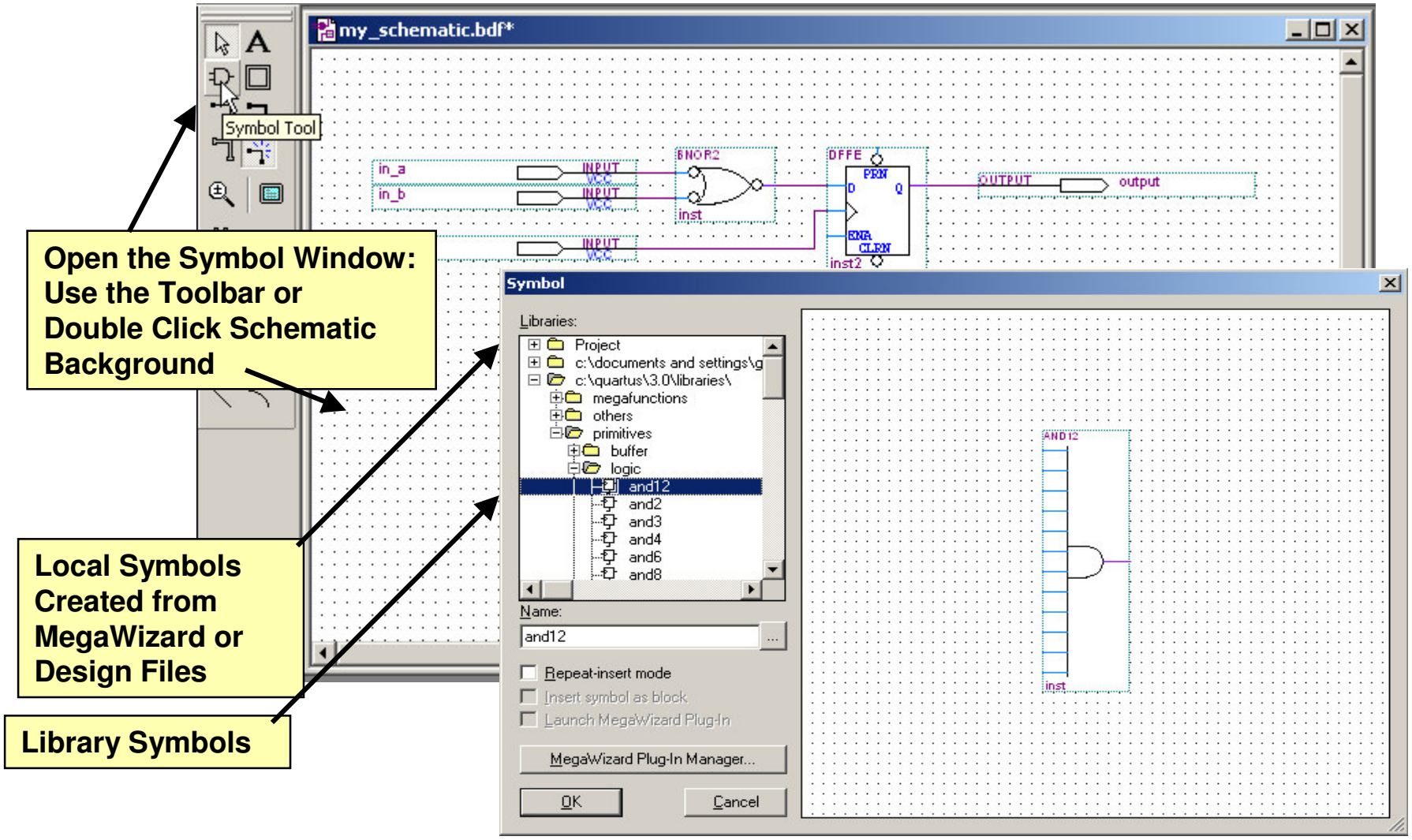
Schematic Design Entry

- 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 out of Verilog, VHDL, or AHDL Design Files
  - Connect All Blocks Using Wires & Busses
- 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

Create Schematic



Insert Symbols



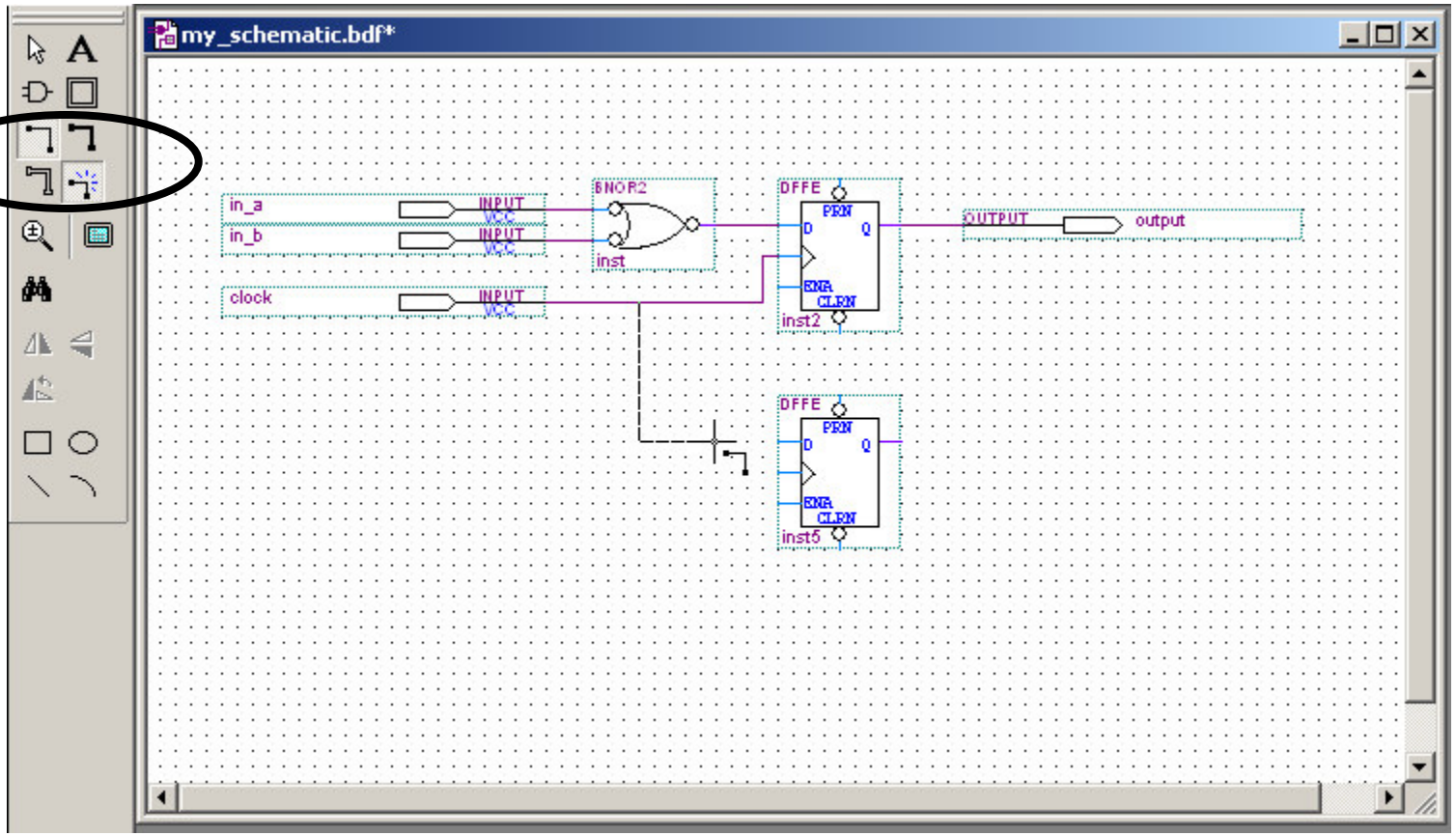
Open the Symbol Window: Use the Toolbar or Double Click Schematic Background

Local Symbols Created from MegaWizard or Design Files

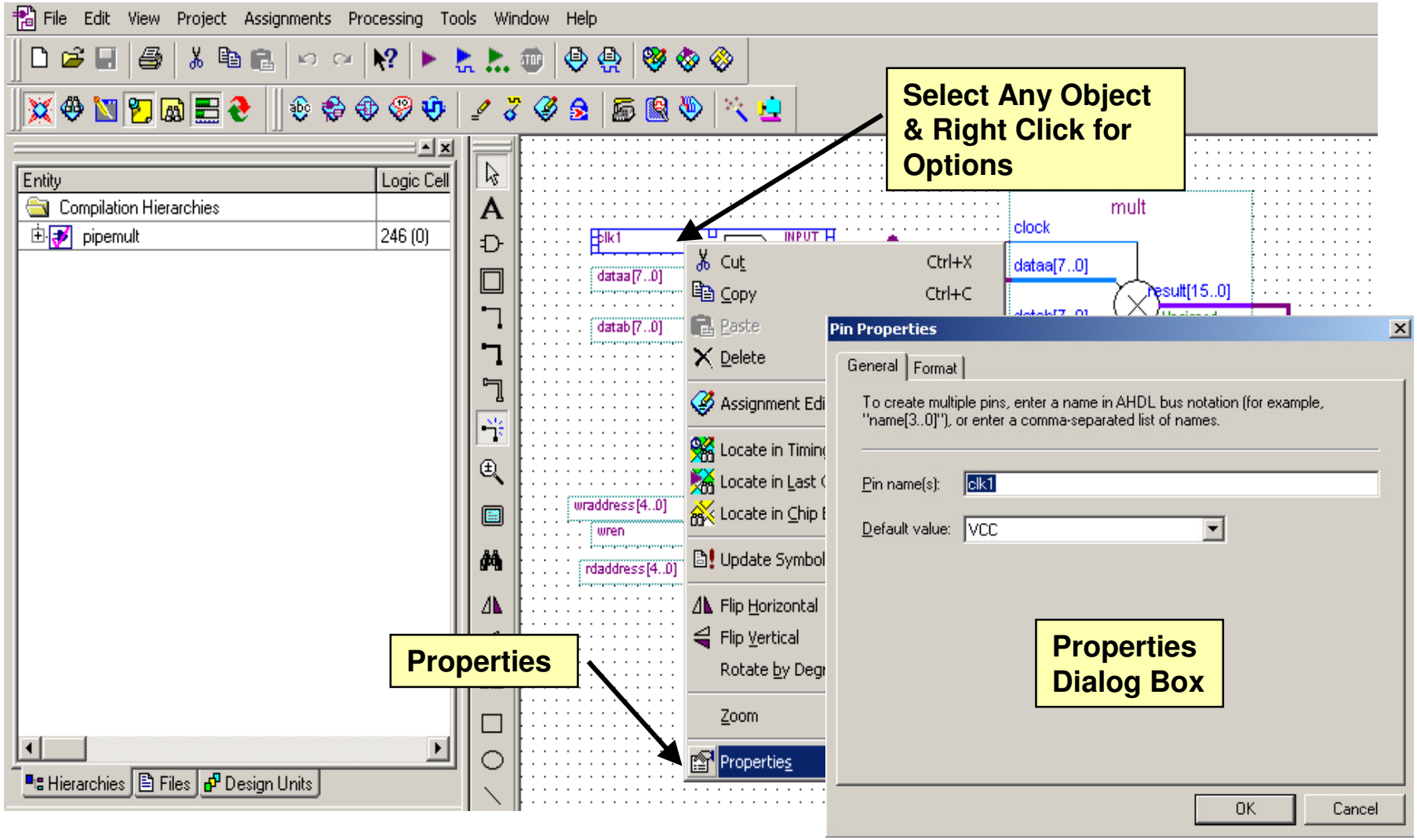
Library Symbols

Connect Wires & Buses

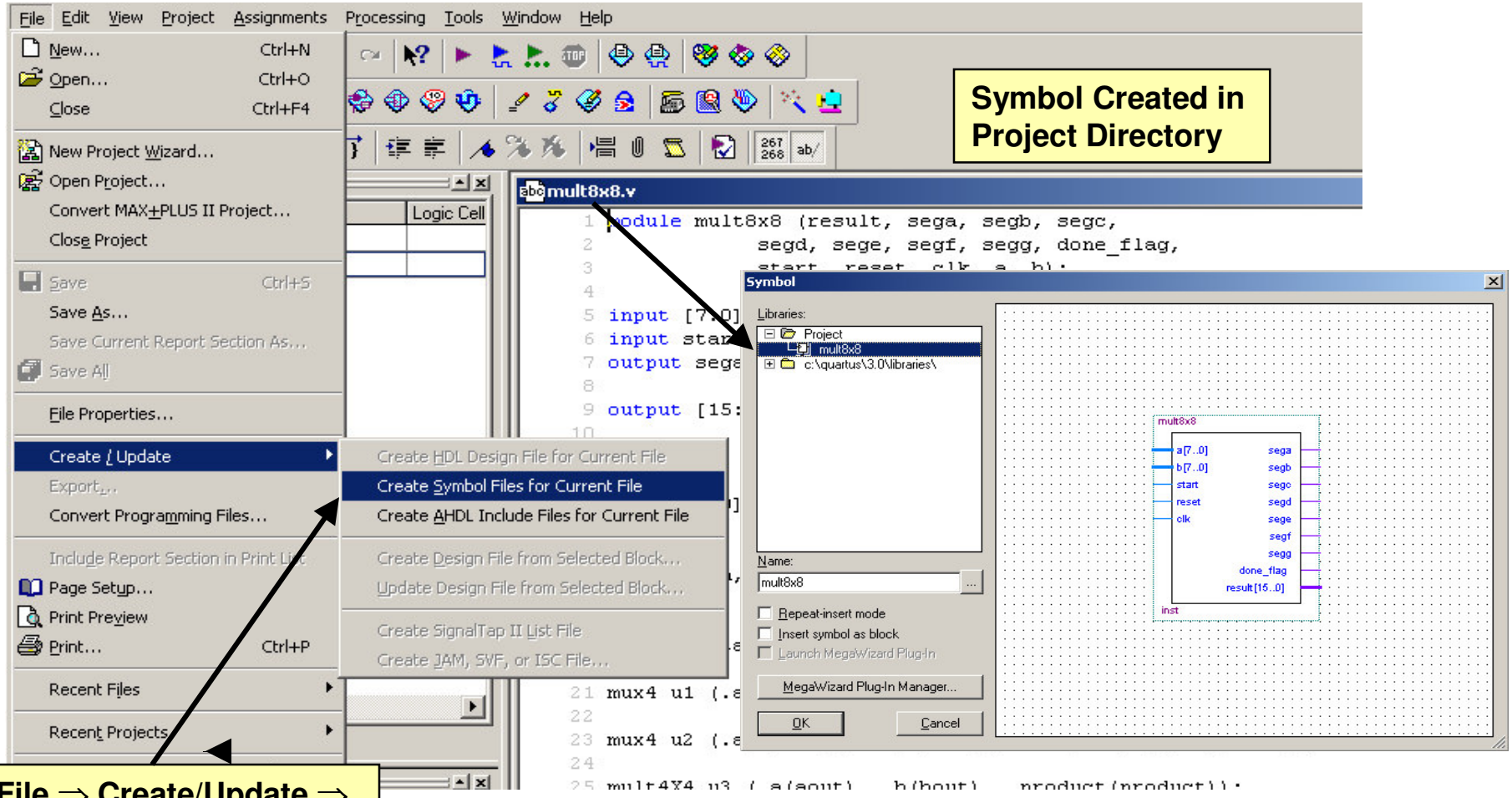
Draw Wires, Buses, or Conduit



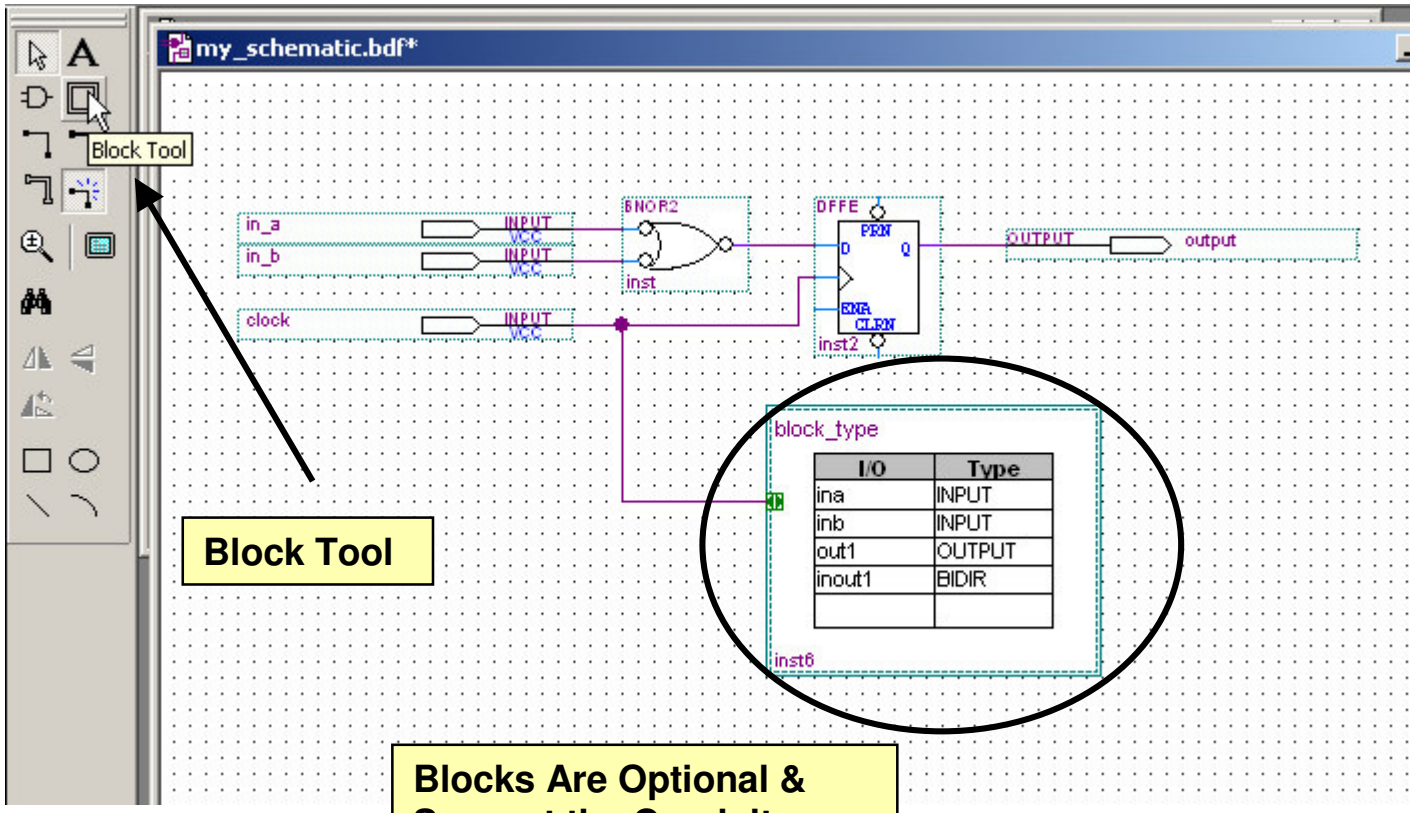
Change Names & Properties



Create Symbols



Optional Block Design Entity

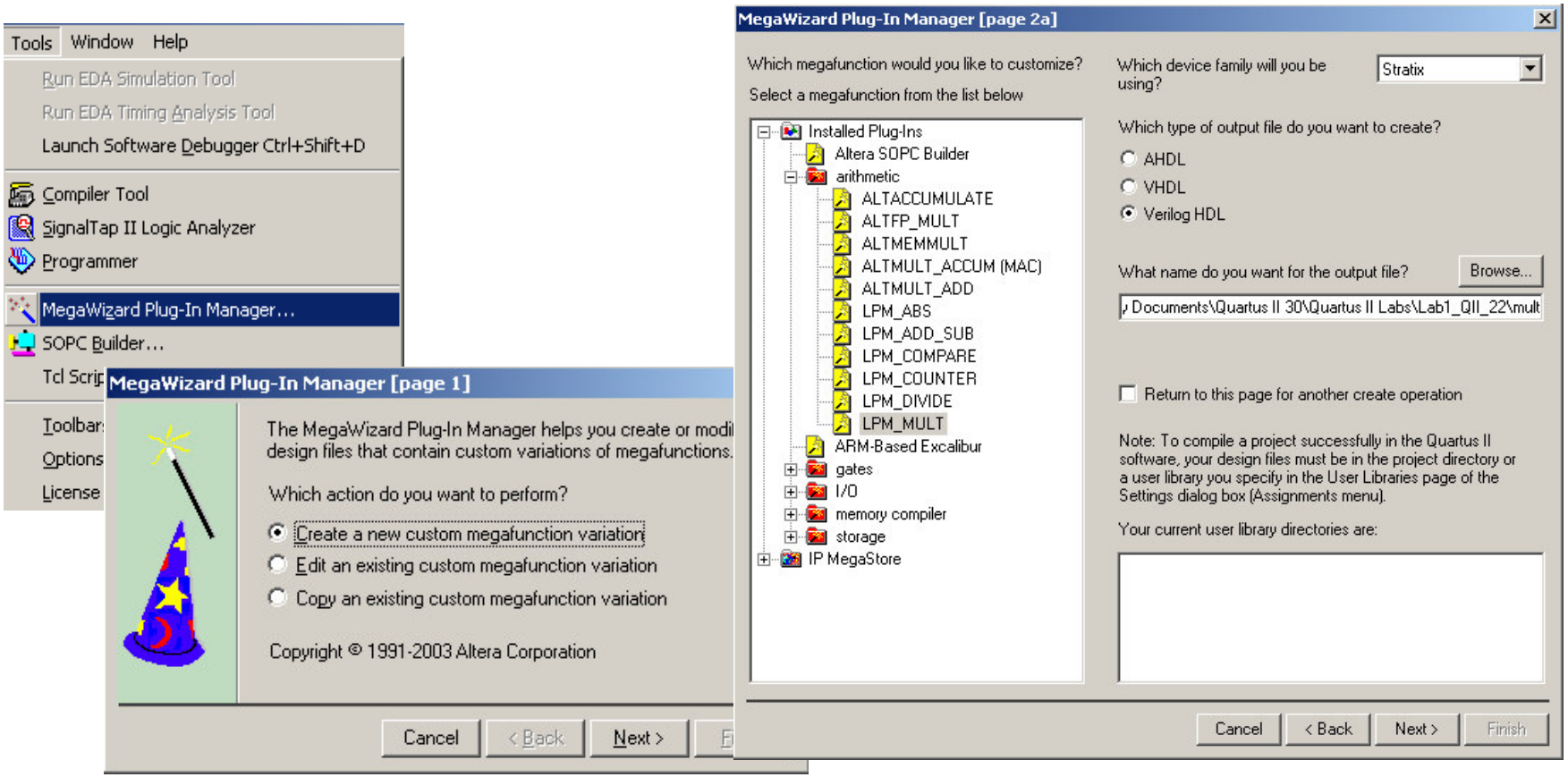


Megafunctions

- Pre-Made Design Blocks
  - Ex. Multiply-Accumulate, PLL, Double-Data Rate
- Benefits
  - Accelerate Design Entry
  - Pre-Optimized for Altera Architecture
  - Add Flexibility
- Two Types
  - Altera-Specific Megafunctions
  - Library of Paramerterized Modules (LPMs)
    - Industry Standard Logic Functions

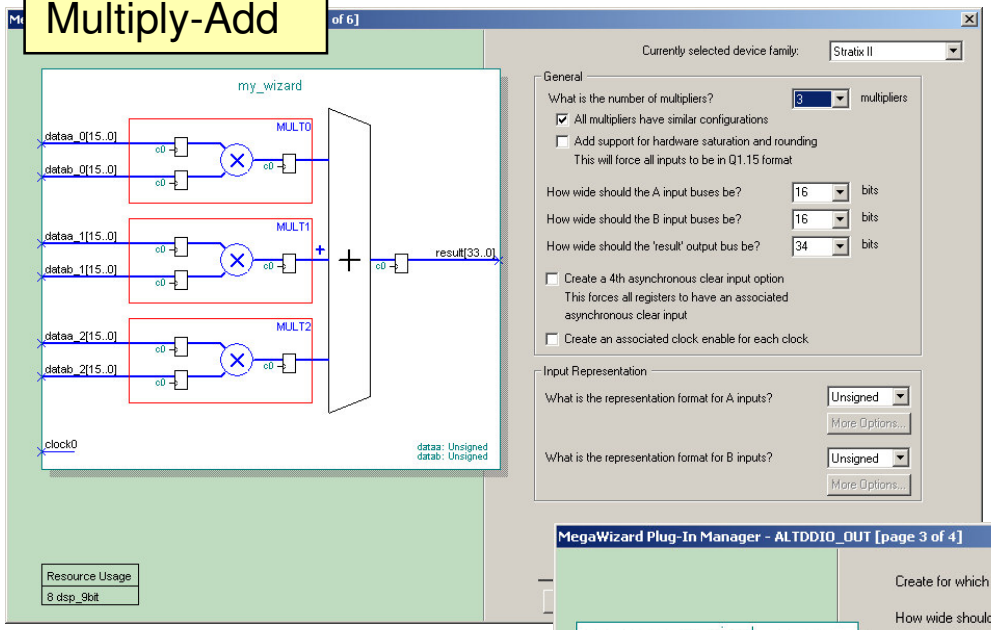
MegaWizard Plug-In Manager

- Eases Implementation of Megafunctions & IP

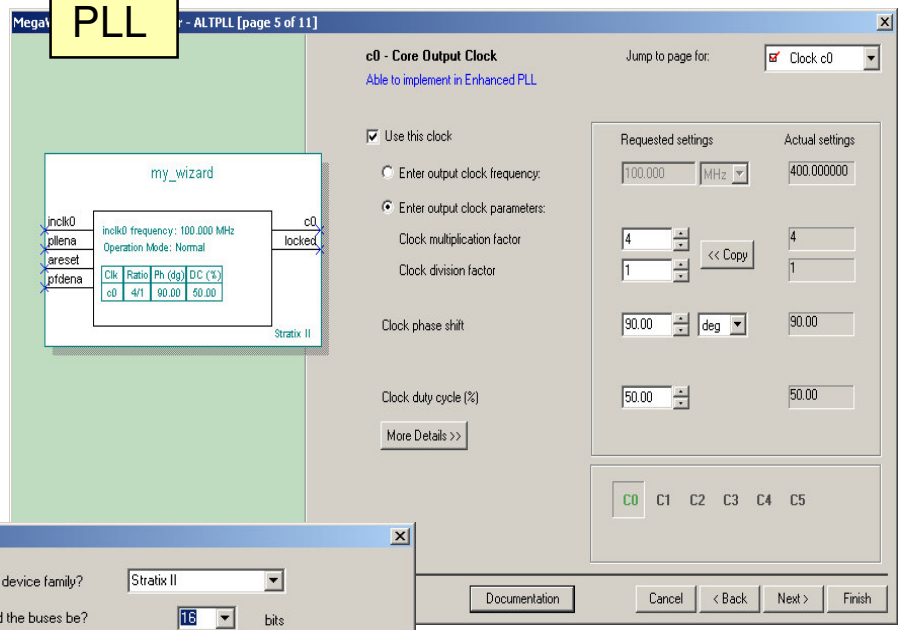


MegaWizard Examples

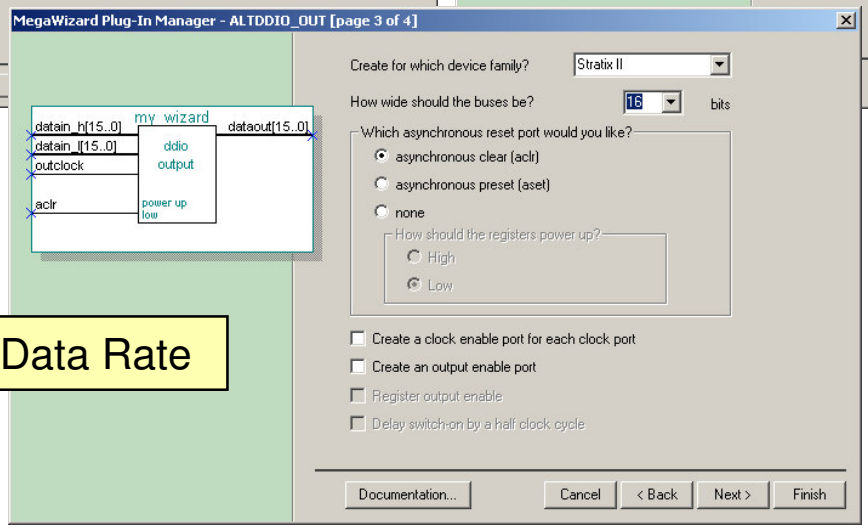
Multiply-Add



PLL

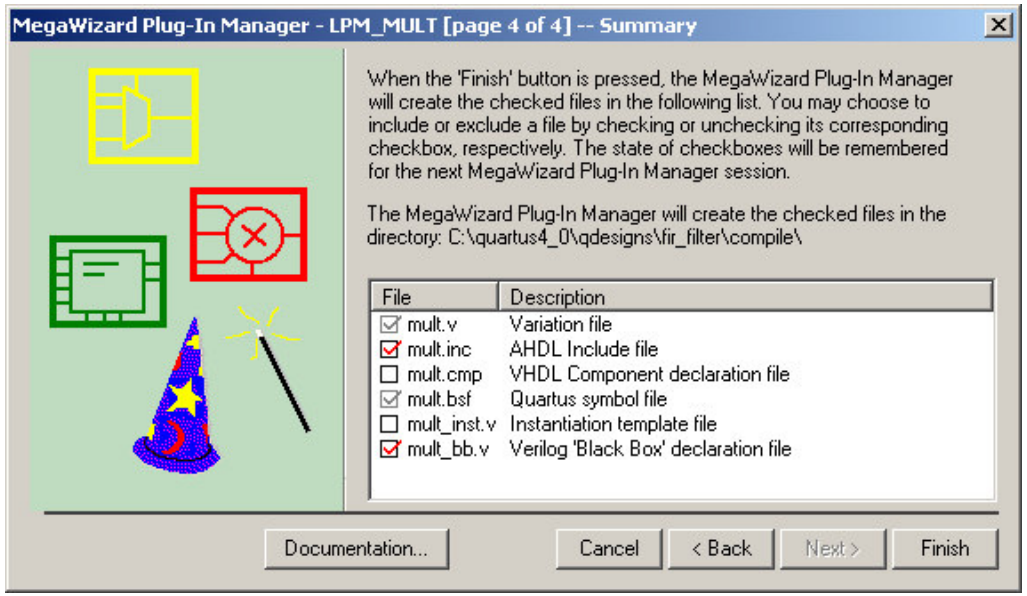


Double-Data Rate



MegaWizard Output File Selection

- Default
  - HDL Wrapper File
- Selectable
  - HDL Instantiation Template
  - VHDL Component Declaration (CMP)
  - Quartus II Symbol (BSF)
  - Verilog Black Box



Memory Editor

- Create or Edit Memory Initialization Files in Intel Hex (.HEX) or Altera-Specific (.MIF) Format

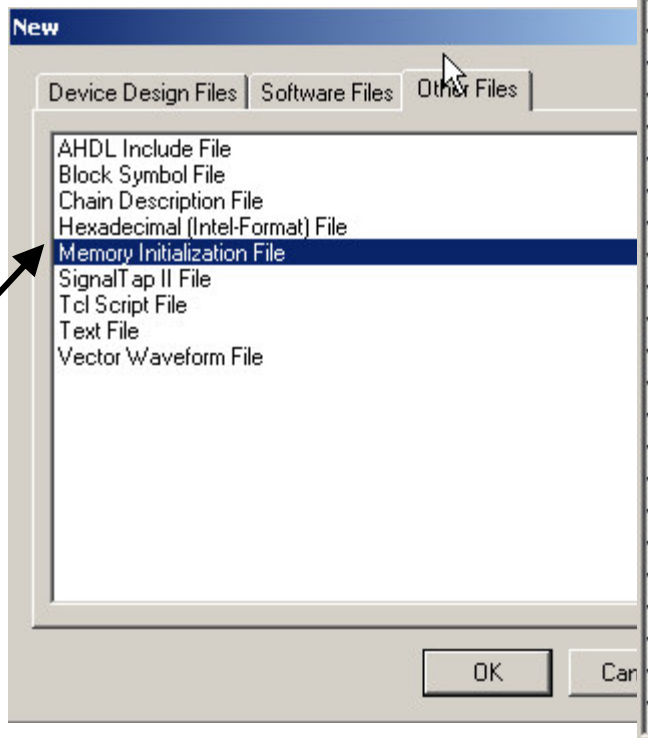
- Design Entry
  - Use to Initialize Your Memory Block (Ex. RAM, ROM) during Power-Up

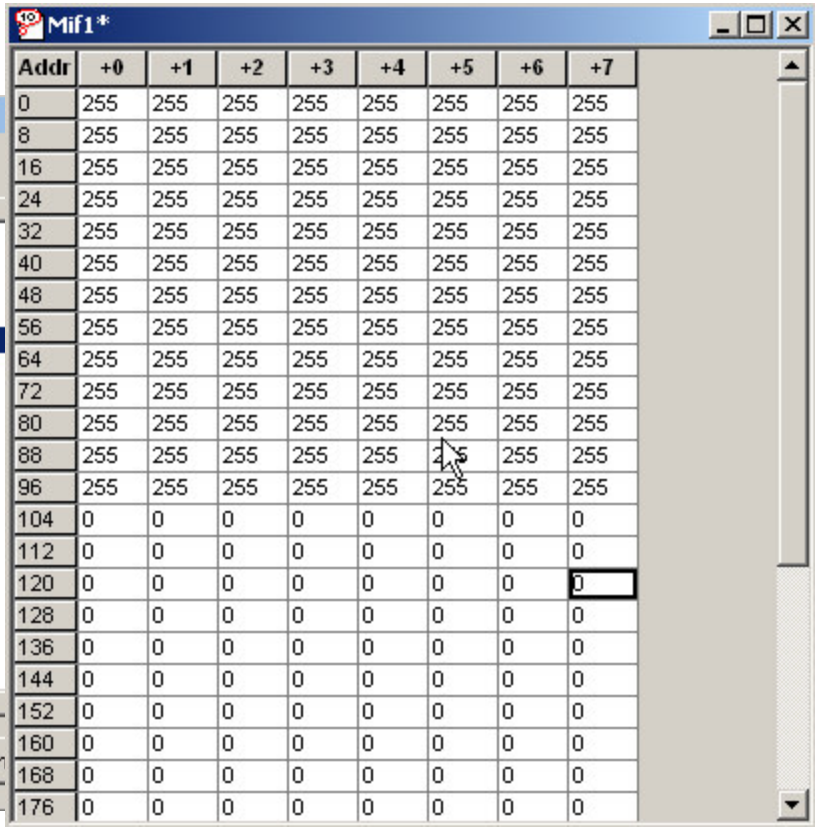
- Simulation
  - Use to Initialize Memory Blocks before Simulation

Create Memory Initialization File

File ⇒ New ⇒ Other Files tab

HEX format or MIF format

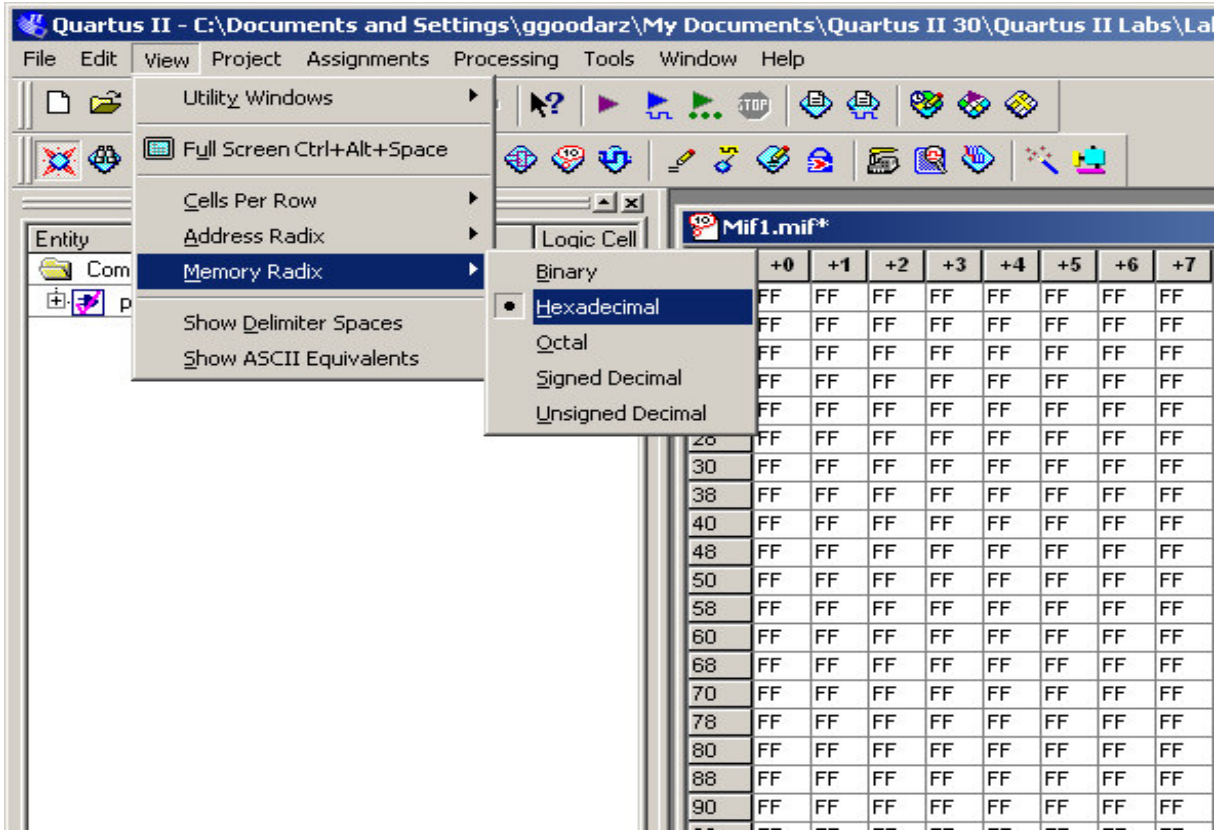




Addr	+0	+1	+2	+3	+4	+5	+6	+7
0	255	255	255	255	255	255	255	255
8	255	255	255	255	255	255	255	255
16	255	255	255	255	255	255	255	255
24	255	255	255	255	255	255	255	255
32	255	255	255	255	255	255	255	255
40	255	255	255	255	255	255	255	255
48	255	255	255	255	255	255	255	255
56	255	255	255	255	255	255	255	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
176	0	0	0	0	0	0	0	0

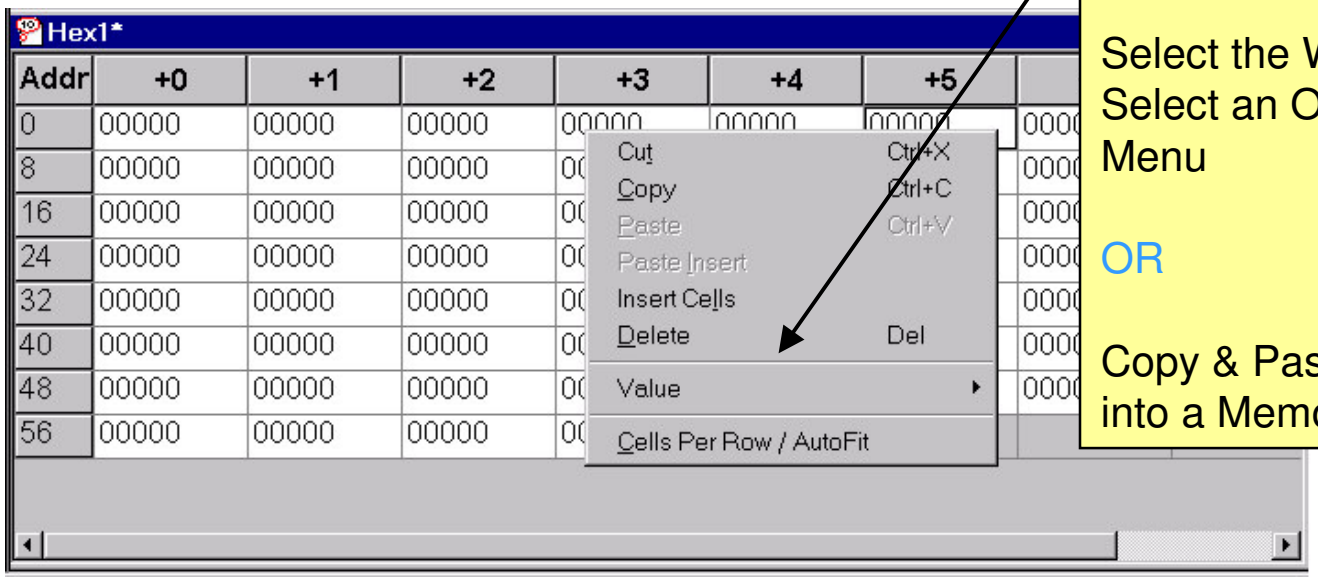
Change Options

- View Options of Memory Editor
  - View ⇒ Select from Available Options



Edit Contents

- Edit Contents of the Memory File
- Save the Memory File as .HEX or .MIF File



Select the Word & Type in a Value

OR

Select the Word & Right Click to Select an Option from the Pop-Up Menu

OR

Copy & Paste from a Spreadsheet into a Memory File

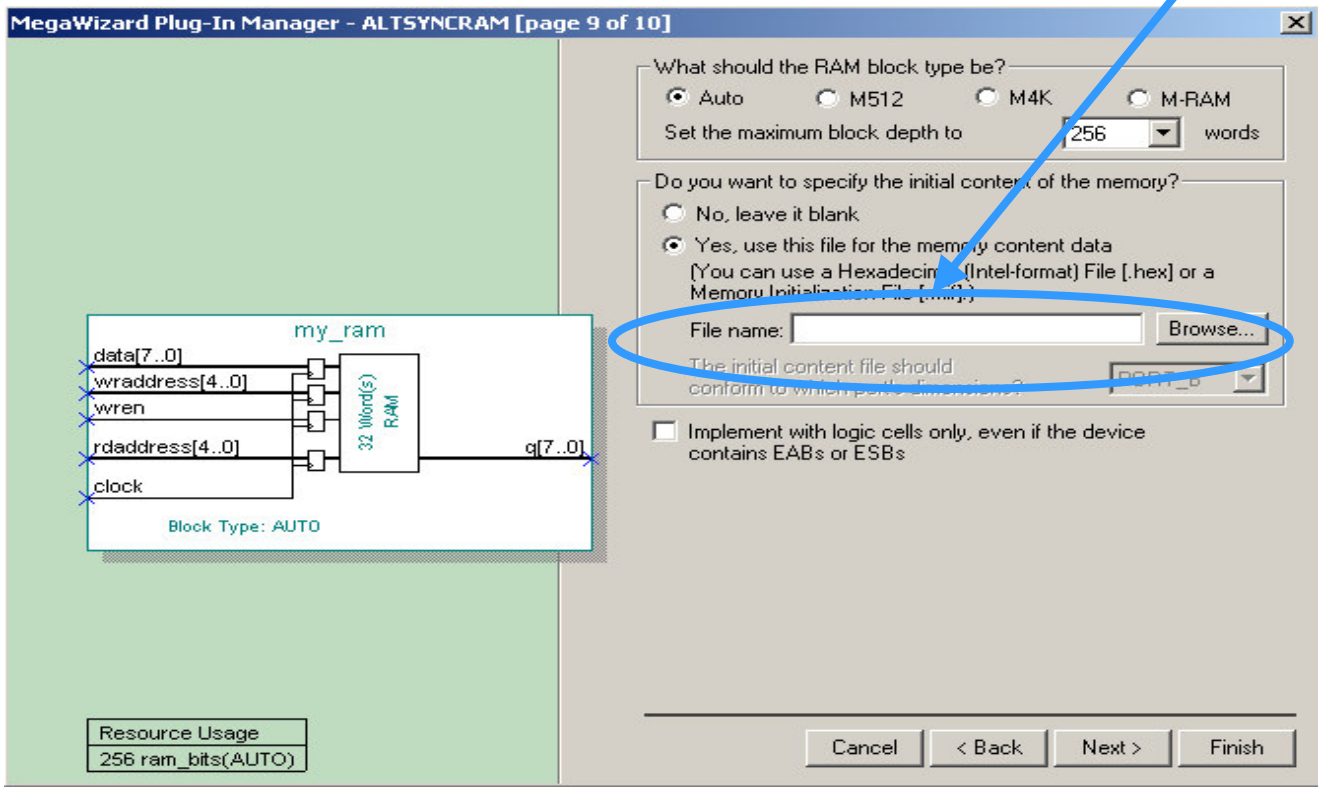
Memory Size Wizard

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

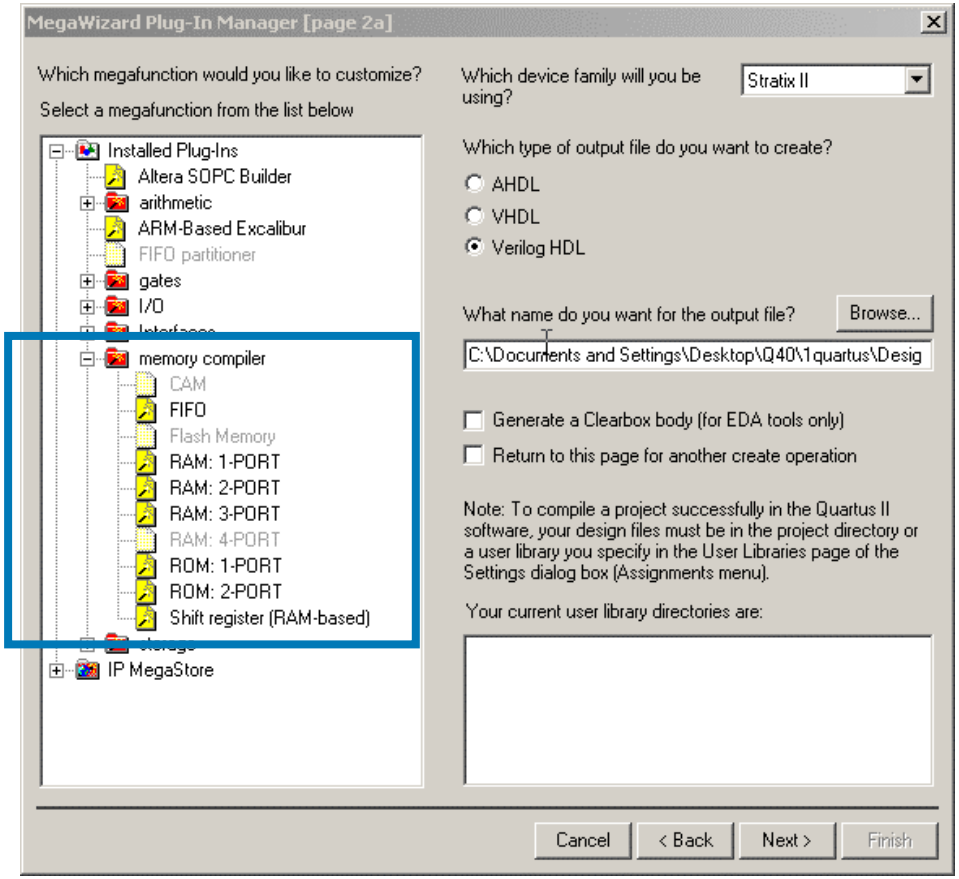
Using Memory File in Design

Specify MIF or HEX file in MegaWizard Interface



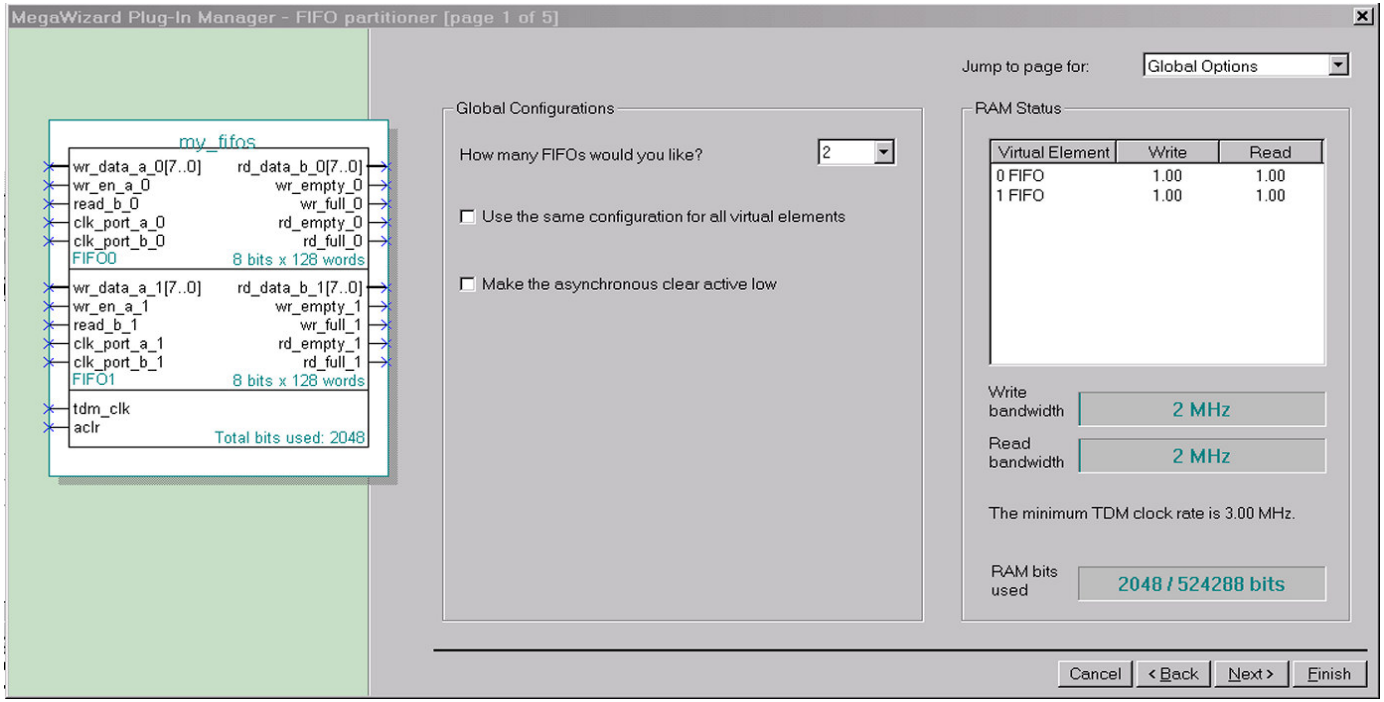
Memory Compiler

- Memory Functions are Organized by Type
- MegaWizard is Family-Aware
  - Select Memory Type & Parameters
  - Appropriate Megafunction Selected Automatically



FIFO Partitioner

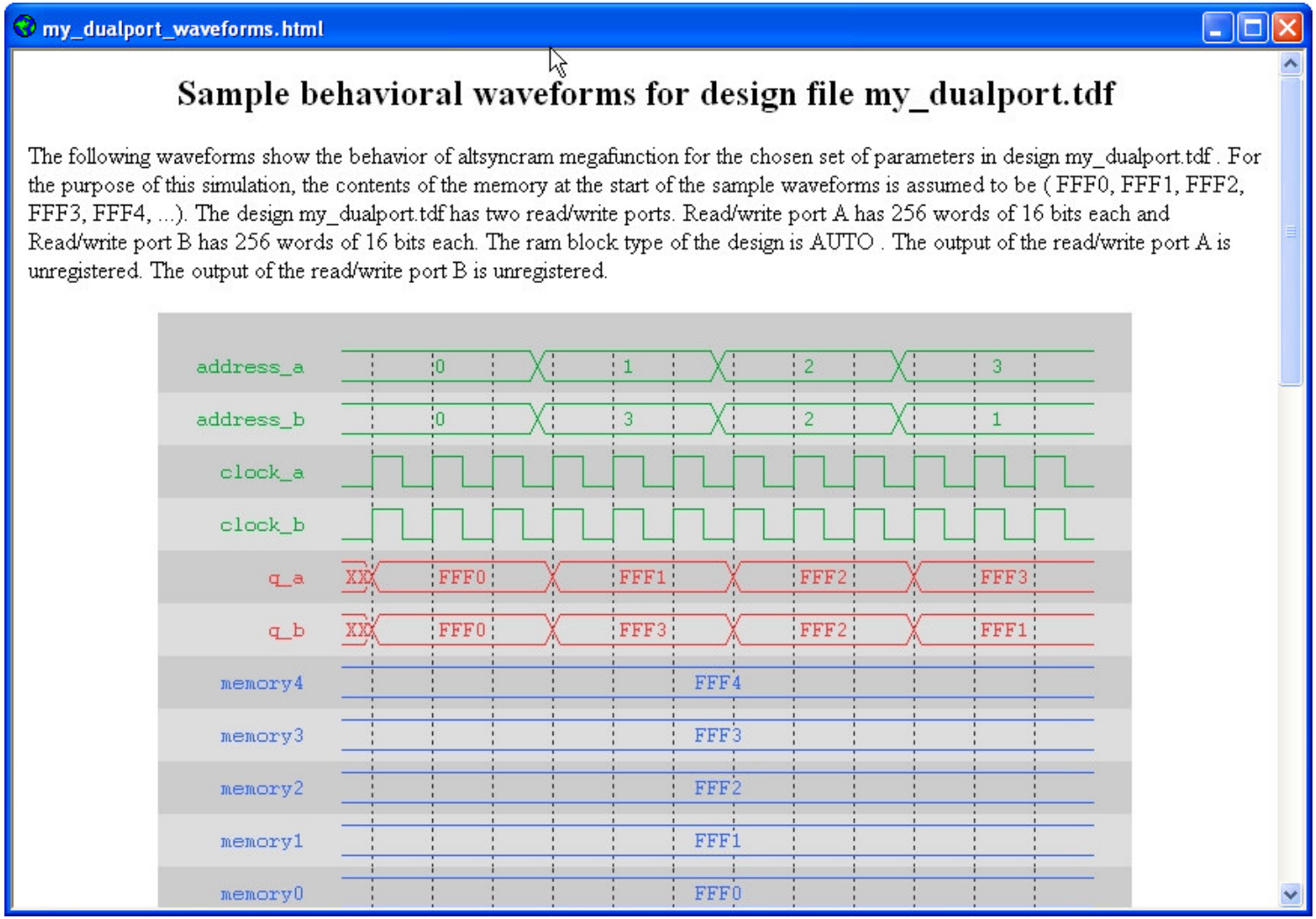
- Map Multiple FIFOs into a Single Physical Memory Block
- Partition M-RAM Blocks into Multiple FIFOs



Memory Behavioral Waveforms

- HTML file Generated by MegaWizard
- Description of Memory Functionality
  - Reviews Selected Parameters
  - Describes Read & Write Operations

Example Memory Waveform

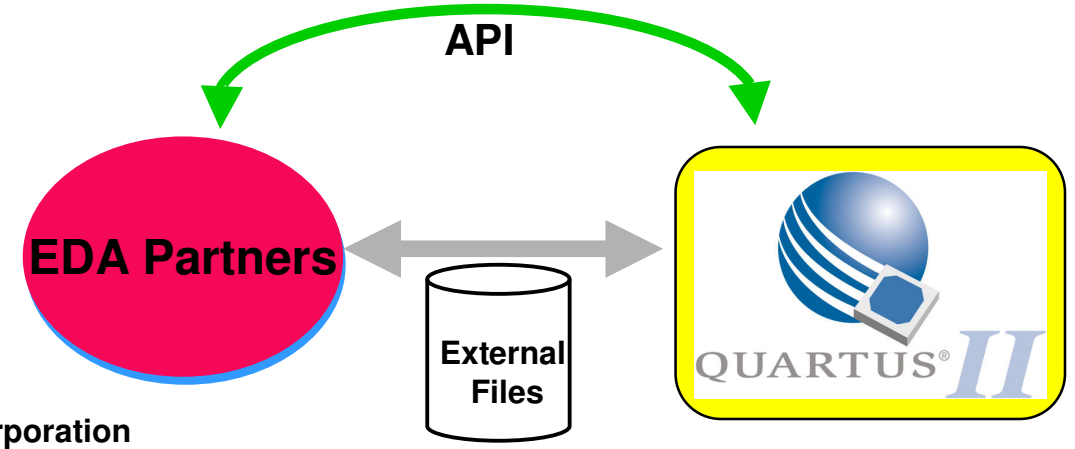


EDA Interfaces Introduction

- Interface with Industry-Standard EDA Tools that Generate a Netlist File
  - EDIF 200
  - VHDL ’87 or ’93
  - Verilog
- NativeLink Interface Provides Seamless Integration with 3rd-party EDA Software Tools
  - Tools Pass Information/Commands in Background
  - Designers Can Complete Entire Design in One Tool

NativeLink

- Comprised of Two Components
  - External Files
    - WYSIWYG (What You See Is What You Get) ATOM Netlist Files (EDIF, Verilog, VHDL)
    - Cross Reference Files (Ex. XRF)
    - Timing Files (Ex. SDO)
  - Application Programming Interface (API)
    - Pre-Defined Interface of Commands/Functions



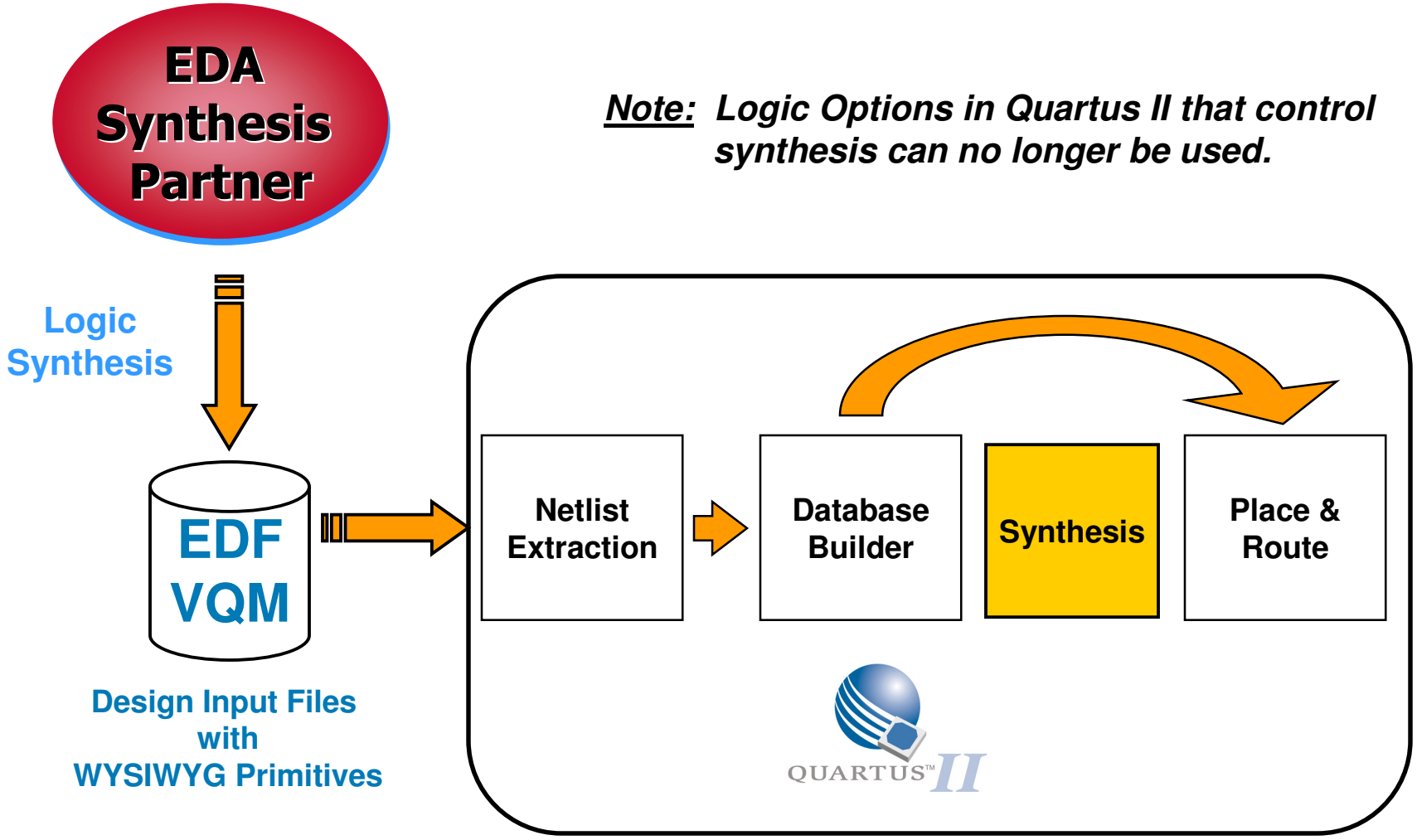
WYSIWYG ATOM Primitives

- Set of Design Primitives that Support WYSIWYG Compilation
- Provide Direct Control of How a Design Is Technology-Mapped to a Specific Target Device
- Allow Synthesis Vendors to Provide an Optimal Realization of a Design for Each Architecture

```
00104   stratix_lcell par_err_high_0_and2_62_Z (
00105     .combout(par_err_high_0_and2_62),
00106     .dataa(ram_out_c_11),
00107     .datab(ram_out_c_10),
00108     .datac(ram_out_c_9),
00109     .datad(ram_out_c_8)
00110 );
00111 defparam par_err_high_0_and2_62_Z.operation_mode="normal";
00112 defparam par_err_high_0_and2_62_Z.lut_mask="9669";
00113 defparam par_err_high_0_and2_62_Z.synch_mode="off";
00114 defparam par_err_high_0_and2_62_Z.sum_lutc_input="datac";
```

Synplify .vqm

WYSIWYG Compilation Flow



Note: *Logic Options in Quartus II that control synthesis can no longer be used.*

Three EDA Design Flows

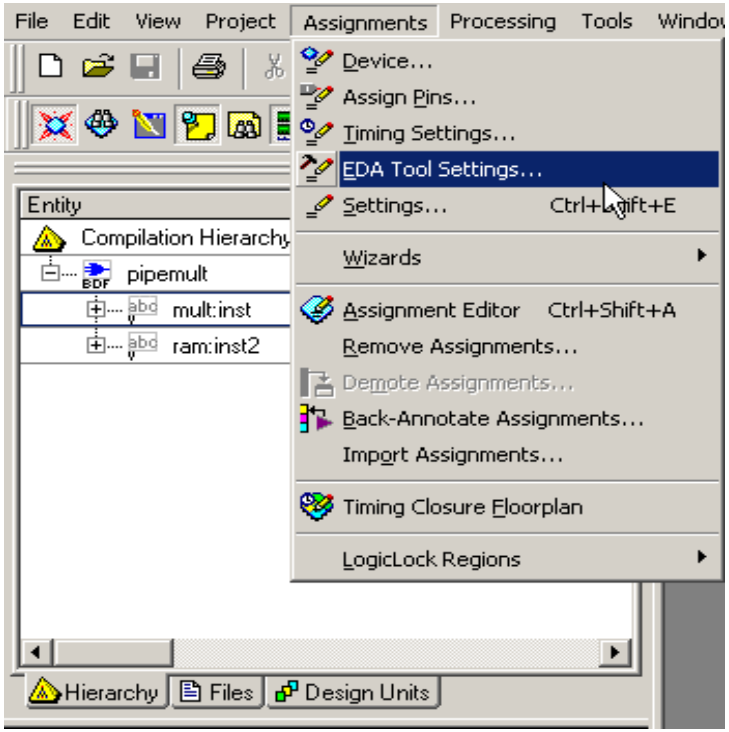
- Quartus II Driven Flow
  - User Launches other EDA Tools from Quartus II in the Background
  - Messages Appear in Quartus II Message Window
- Vendor Driven Flow
  - User Runs Quartus II in the Background from the 3rd-Party EDA Tool
- File Based Flow
  - Each Tool Ran Separately
  - Files Are Manually Transferred between Tools

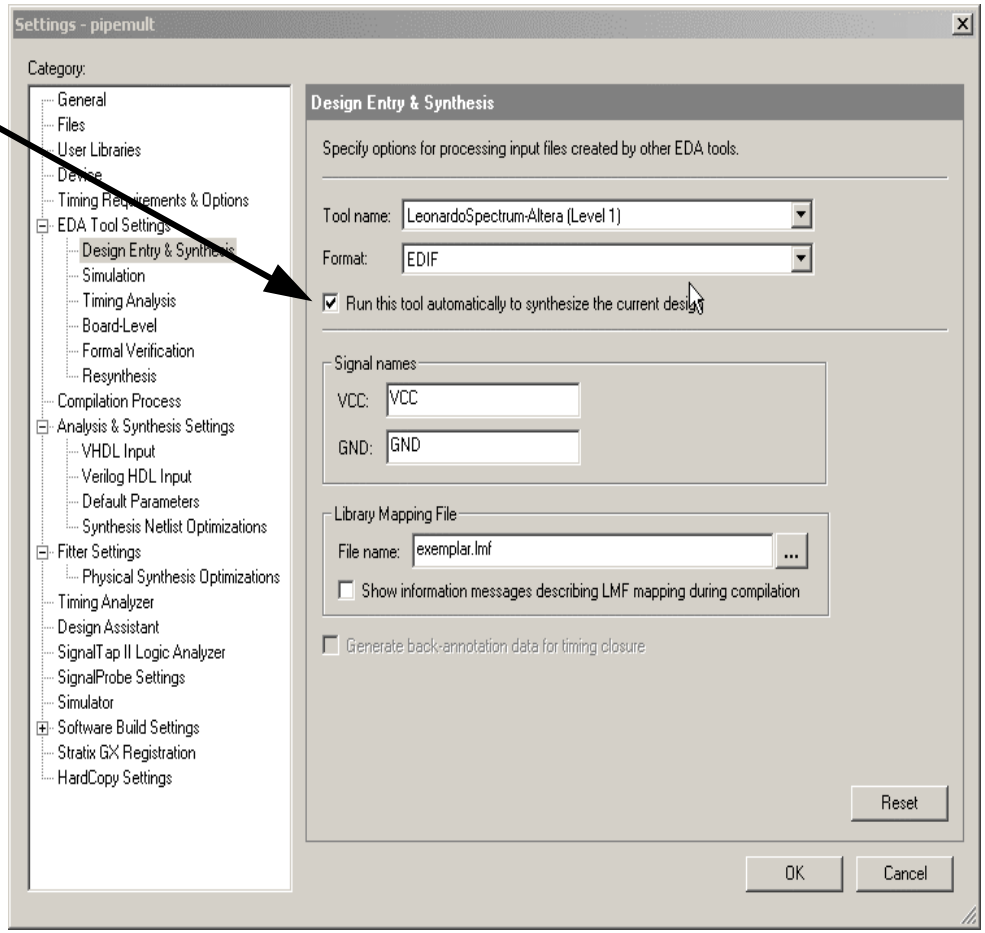
Quartus II Driven Flow

Assignments ⇒ EDA Tool Settings

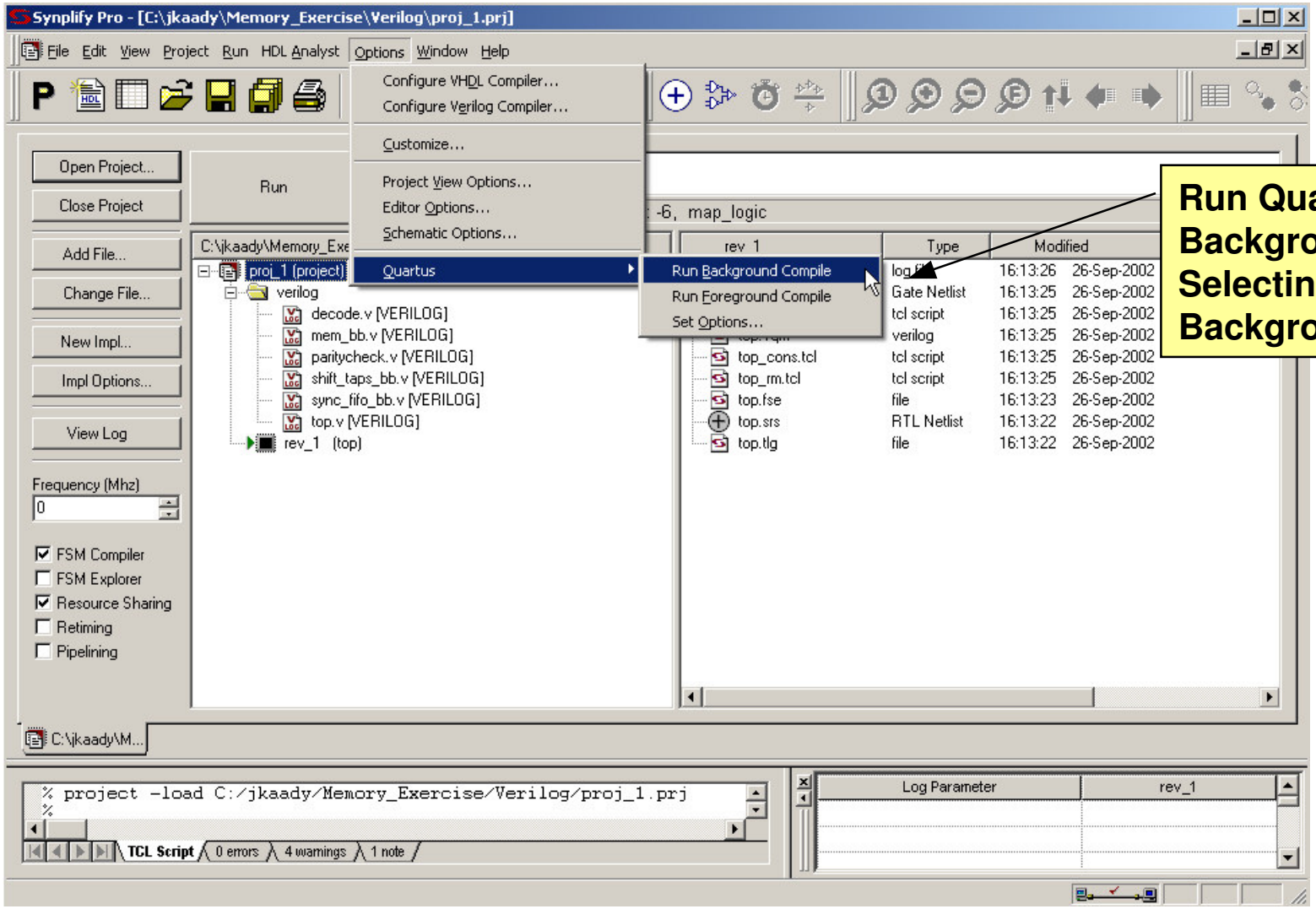
Check the "Run …" Button to Launch the EDA Tool in the Background

Leave Unchecked for File Based Flow



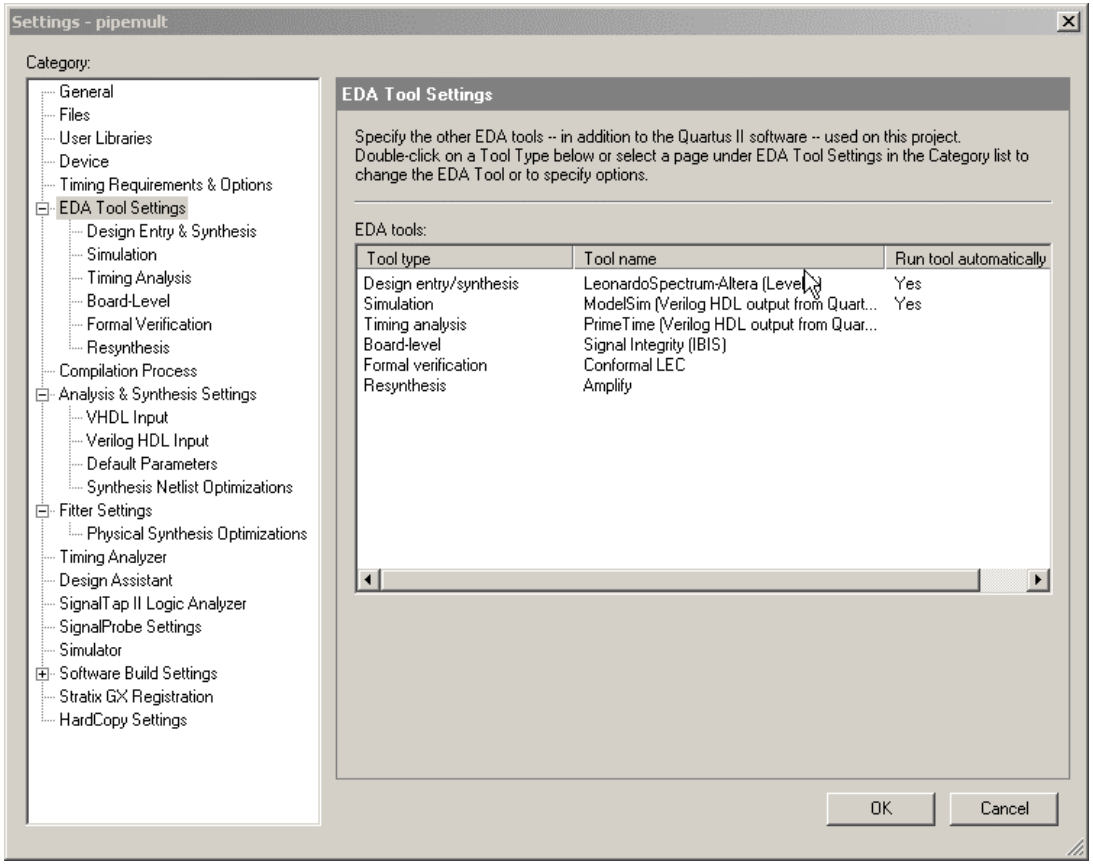


EDA Driven Flow



File Based Flow

- Work Separately in Individual Tools
  - Synthesis
  - Simulation
  - Timing Analysis
  - Board-Level
  - Formal Verification
  - Resynthesis



Third Party Tool Support

Synthesis Tools

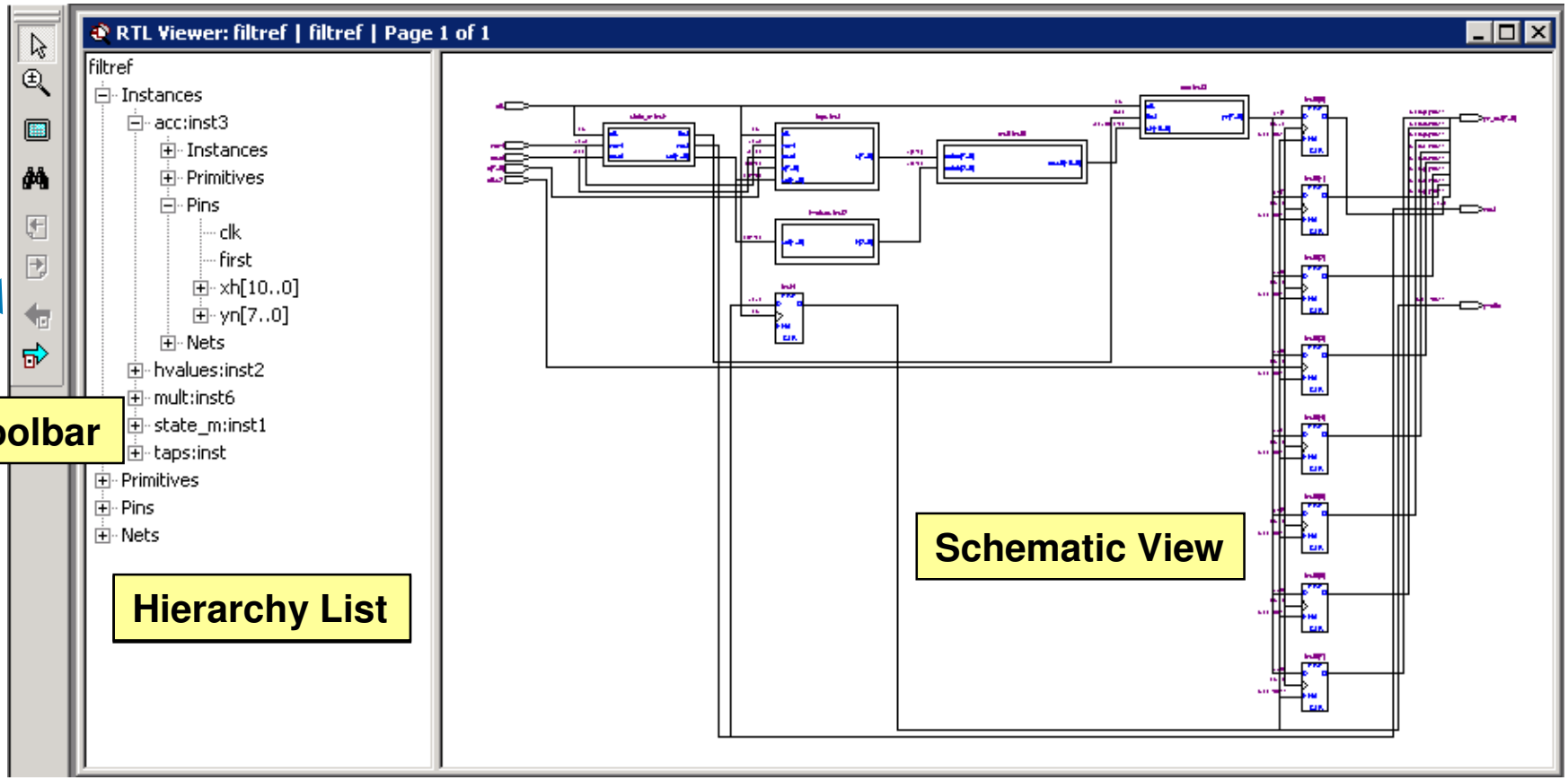
- LeonardoSpectrum™
- Precision
- DesignCompiler
- FPGA Compiler II™
- FPGA Express™
- Synplify
- Synplify Pro
- Amplify

Verification Tools

- ModelSim®
- ModelSim-Altera
- Cadence Verilog-XL
- Cadence NC-Verilog
- Cadence NC-VHDL
- Innoveda BLAST
- PrimeTime®
- Synopsys® VCS & VSS
- Mentor Graphics® Tau
- Synopsys Scirocco

RTL Viewer

- Graphically Represents Results of Synthesis



RTL Viewer Uses

- Visually Checking Initial HDL Synthesis Results
  - Before Any Quartus II Optimizations
- Locating Synthesized Nodes for Assigning Constraints
- Debugging Verification Issues
- Reading VQM/EDIF Netlist Files

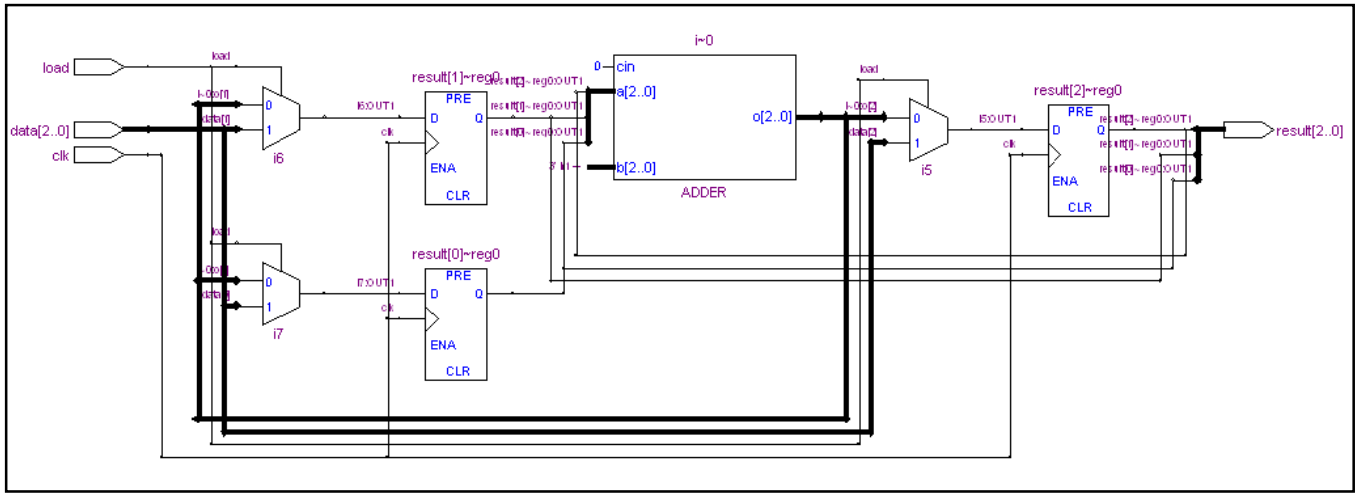
Starting RTL Viewer

1. Run Analysis & Elaboration (Processing Menu or )
   - Any Processing that Performs Elaboration
2. Open RTL Viewer
   - Tools Menu or 
   - Displays Last Successful Compilation

RTL Viewer Features

- Schematic View
- Hierarchy List
- Hierarchy Navigation
- Filter Schematic
- Page Control

Schematic View



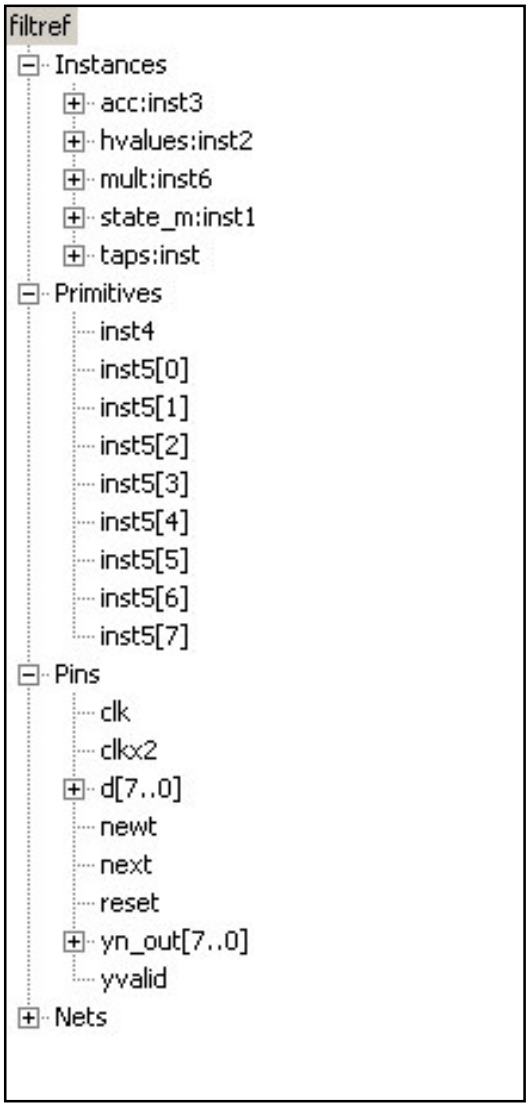
- Represents Design Using Logic Blocks & Nets
  - Registers
  - Muxes
  - Gates
  - Adders

Hierarchy List

- Traverses between Design Hierarchy
- Views Logic Schematic for Each Hierarchical Level
- Breaks down Design into Netlist Elements

Netlist Elements

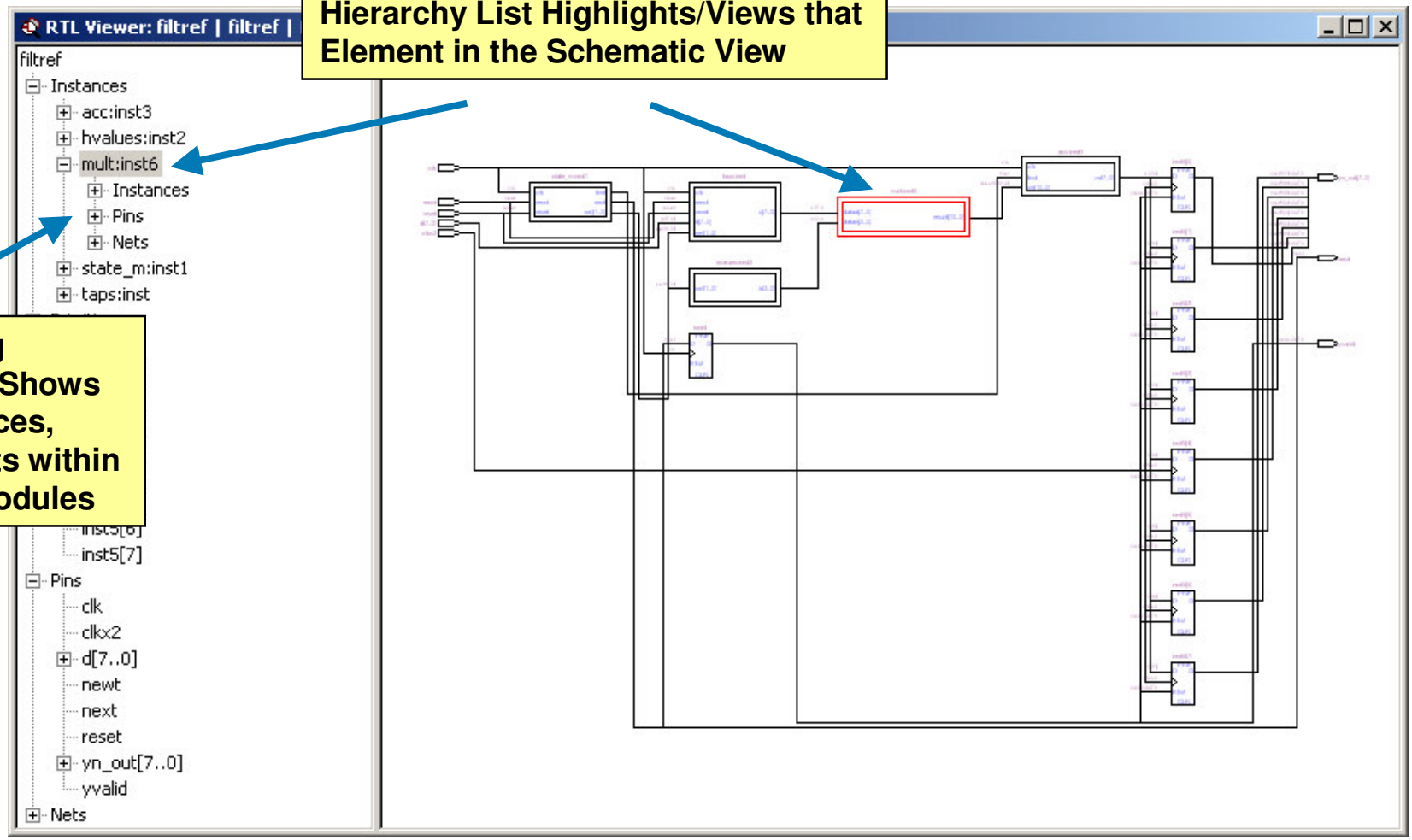
- **Instances**
  - Lower Hierarchy Levels that Can Be Expanded (Unless Encrypted)
- **Primitives**
  - Low-level Nodes/Blocks that Cannot Be Expanded
- **Pins**
  - I/O Ports in the Current Level of Hierarchy
- **Nets**
  - Nets or Wires that Connect Nodes



Using Hierarchy List

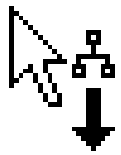
Highlighting a Netlist Element in Hierarchy List Highlights/Views that Element in the Schematic View

Expanding Instances Shows the Instances, Pins & Nets within Internal Modules

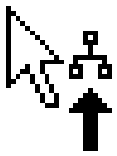


Hierarchy Navigation

- Hierarchy List
  1. Click on Netlist Element (Ex. Instance)
  2. Schematic View Automatically Adjusts Hierarchical Level to Show Element
- Schematic View
  - Mouse Pointer Indicates Action



←Descending Hierarchy
- Double-Click on Instance
- Right-Click & Select **Hierarchy Down**

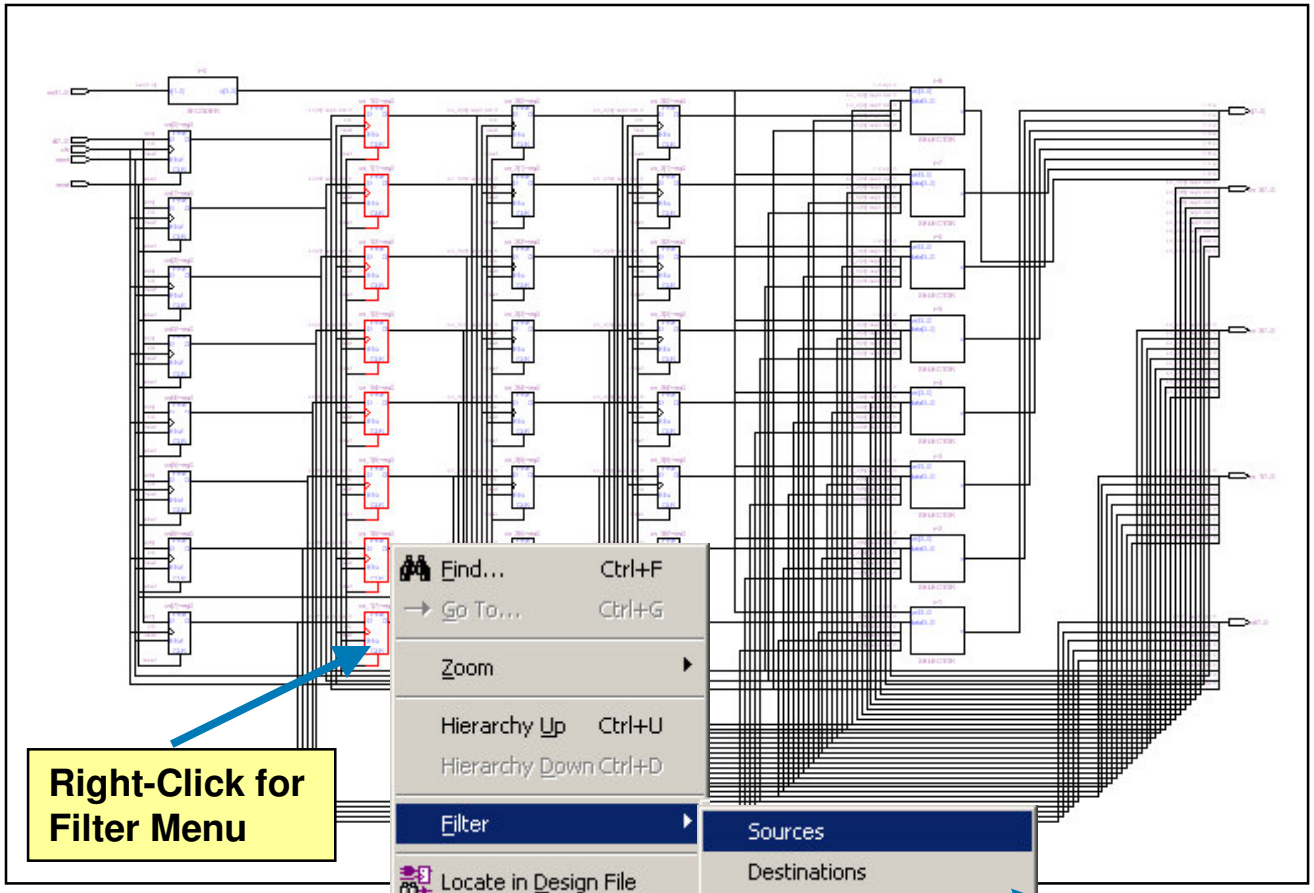


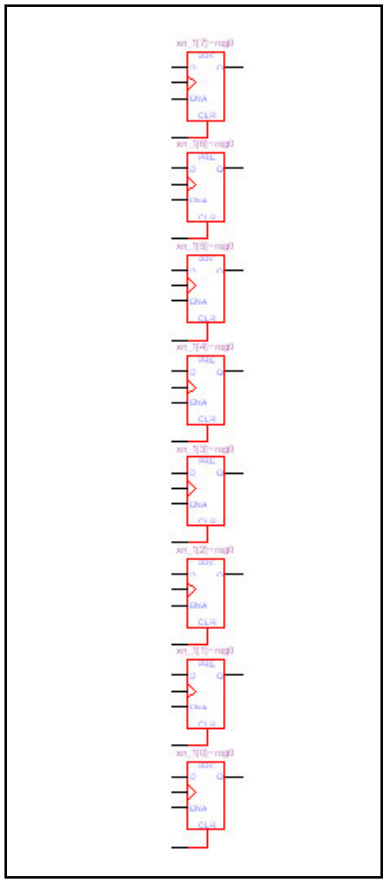
←Ascending Hierarchy
- Double-Click in Empty Space
- Right-Click & Select **Hierarchy Up**

Filter Schematic

Unfiltered: All Components & Paths Shown

Filtered: Only Selected Components & Related Paths Displayed



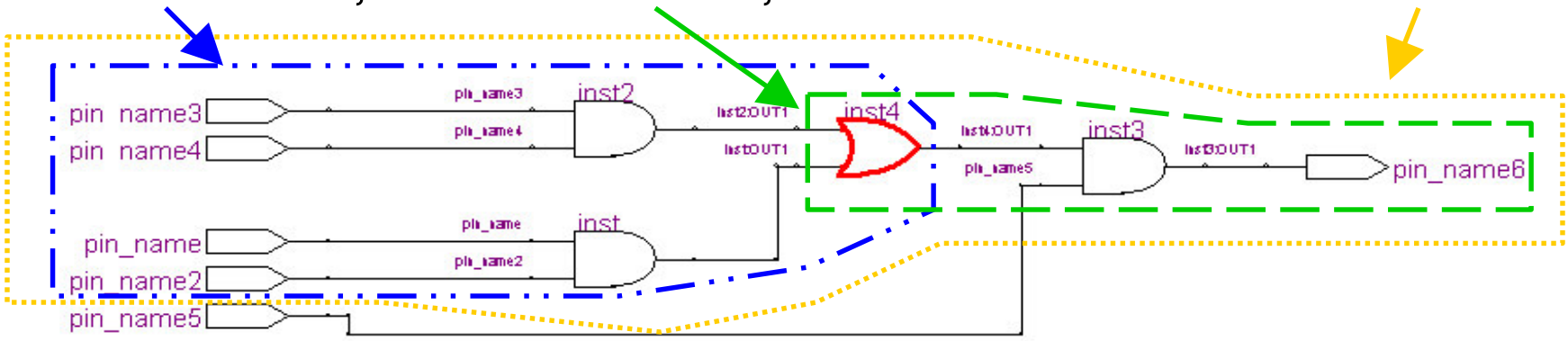


Right-Click for Filter Menu

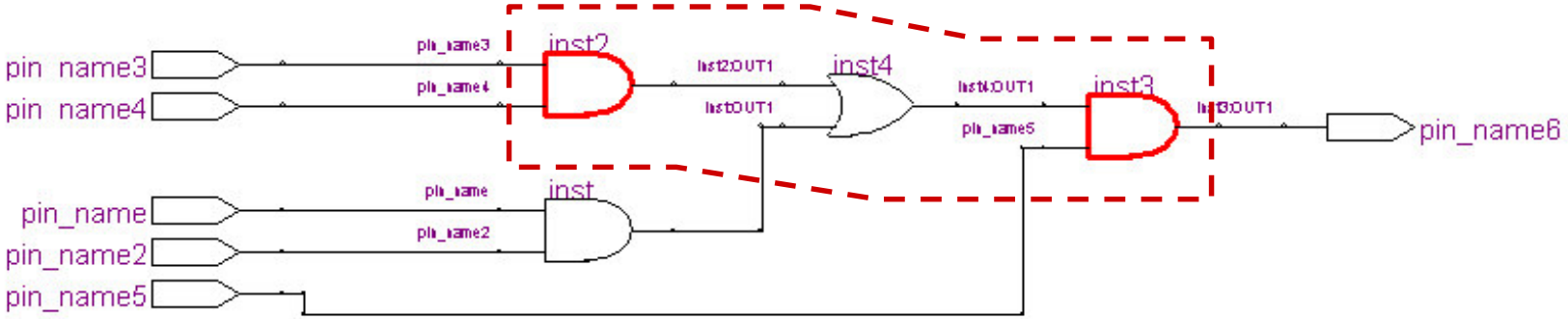
Filter Options

Filter Options

- Sources, Destinations, Sources & Destinations



- Between Selected Nodes



More on Filtering

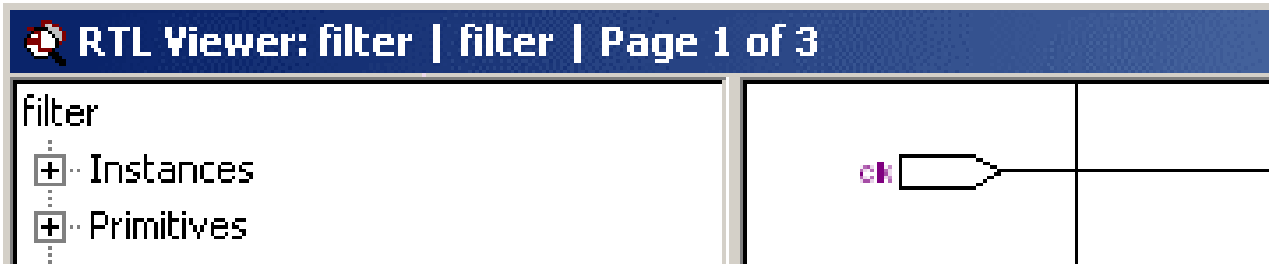
- Displays Only Nodes on Current Hierarchy Level
- Stops Tracing Paths at
  - Hierarchical Ports
  - Registers
    - Option to Disable (Tools Menu ⇒ Options)
  - Specified Number of Logic Levels
    - 10 by Default (Tools Menu ⇒ Options)

Page Control

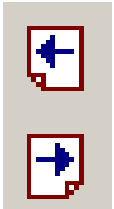
- Schematic Automatically Partitioned
- Option to Control Design Size Per Page (Tools Menu ⇒ Options)
  - Nodes Per Page
    - 1 – 1000 (Default: 50)
  - Ports Per Page
    - 1 – 2000 (Default: 1000)

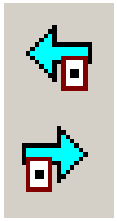
Managing Pages

- Title Bar Indicates Number of Pages



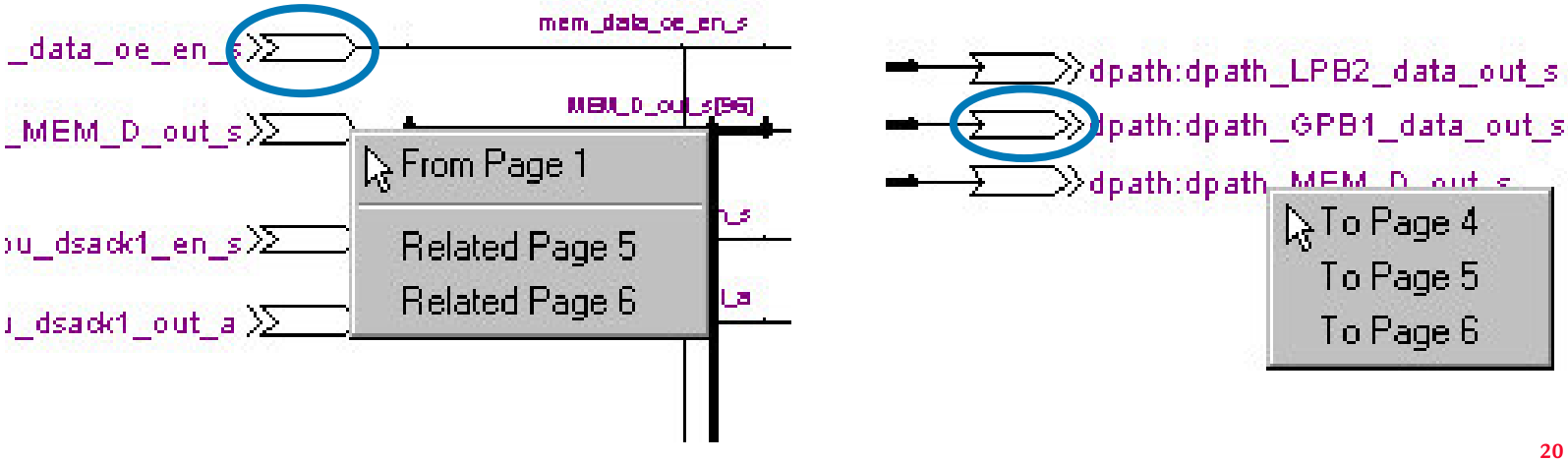
- Moving between Pages
  - Next Page/Previous Page
    - Moves between Partitioned Pages
  - Back/Forward
    - Displays Previously Viewed Design Views





Nets between Pages

- Indicated by Special I/O Connectors
- Right-Click to Trace within Hierarchy
  - From Page (Source)
  - To Page (Destination)
  - Related (Other Parallel Connections)



Other Features

- Go to Net Driver
  - Traces Net Back to Source Driver
  - Changes Schematic View if Necessary
- Locate in Design File
  - Finds Node Definition in Source File

Exercise 1

Please go to Exercise 1 in the Exercise Manual

Exercise Summary

- Schematic Design Entry is Simple
- MegaWizard Makes It Easy
  - Implements Many Design Entities in AHDL, Verilog, or VHDL
  - Makes Best Use of the Device Architecture
  - Has Direct Link to Make IP Downloads Even Easier
- Analysis & Elaboration Option Checks the File

Summary

- Multiple Design Entry Methods
  - Text (Verilog, VHDL, AHDL)
  - Third Party Netlist (VQM, EDF)
  - Schematic
- Memory Editor
- MegaWizard
- EDA Tool Flows
- RTL Viewer

20 YEARS *of*

ALTERA®

INNOVATION

Designing with Quartus II

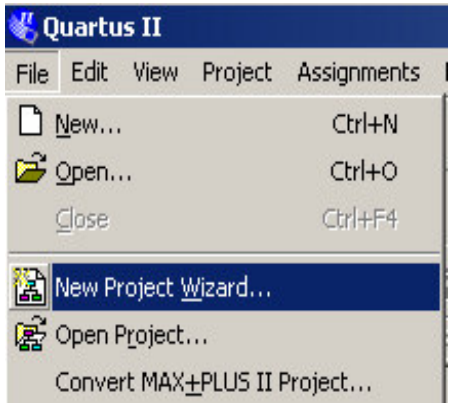
Quartus II Projects

QUARTUS® II

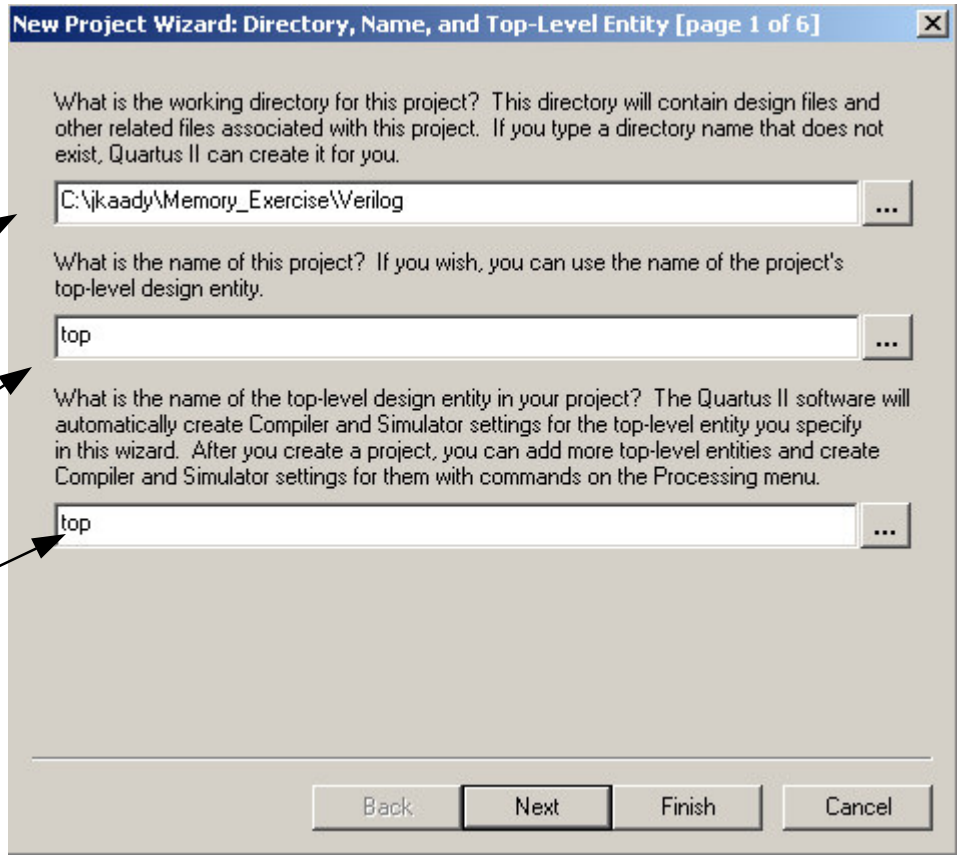
Quartus II Projects

- Description
  - Collection of Related Design Files & Libraries
  - Must Have at Least One Designated Top-Level Entity
  - Target a Single Device or Can Be Partitioned into Multiple Devices
  - Store Settings in Quartus Settings File (.QSF)
- Create New Projects with New Project Wizard
  - Can Be Created Using Tcl Scripts

New Project Wizard



Invoke New Project Wizard

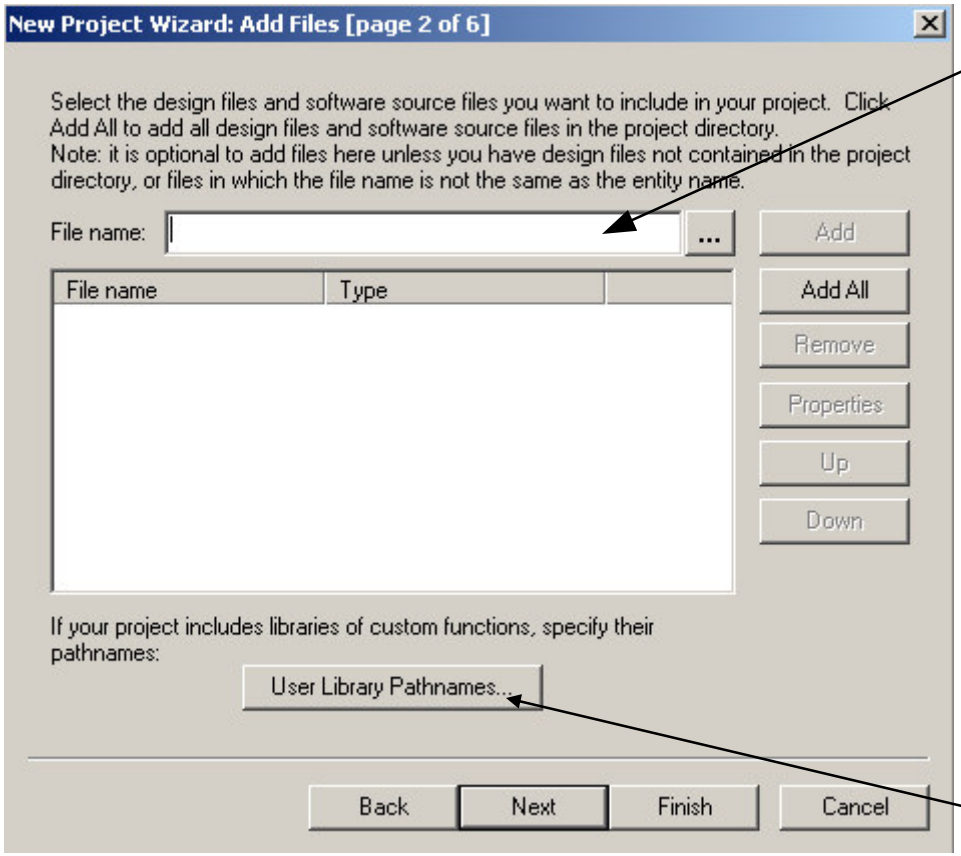


Select Working Directory

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

Add Files



Add Design Files

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

Notes:

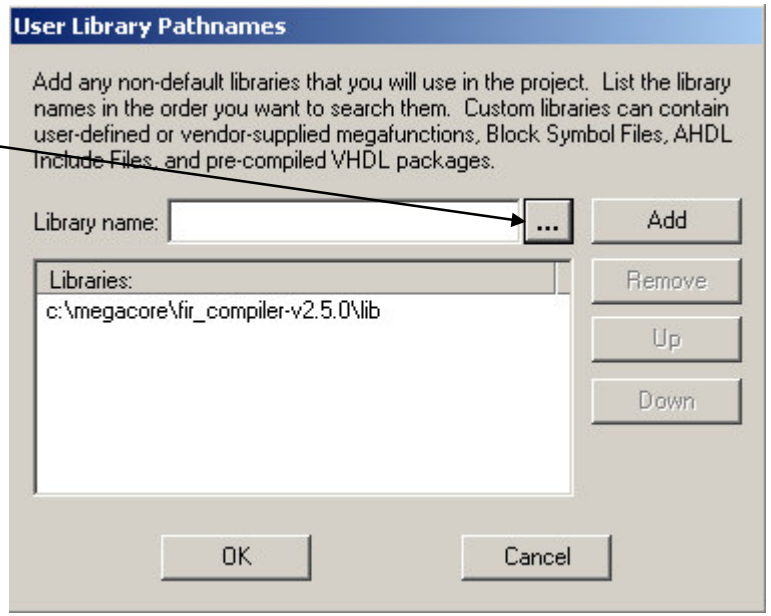
- *Files in project directory do not need to be added*
- *Add top level file if filename & entity name are not the same*

Add User Library Pathnames & Files

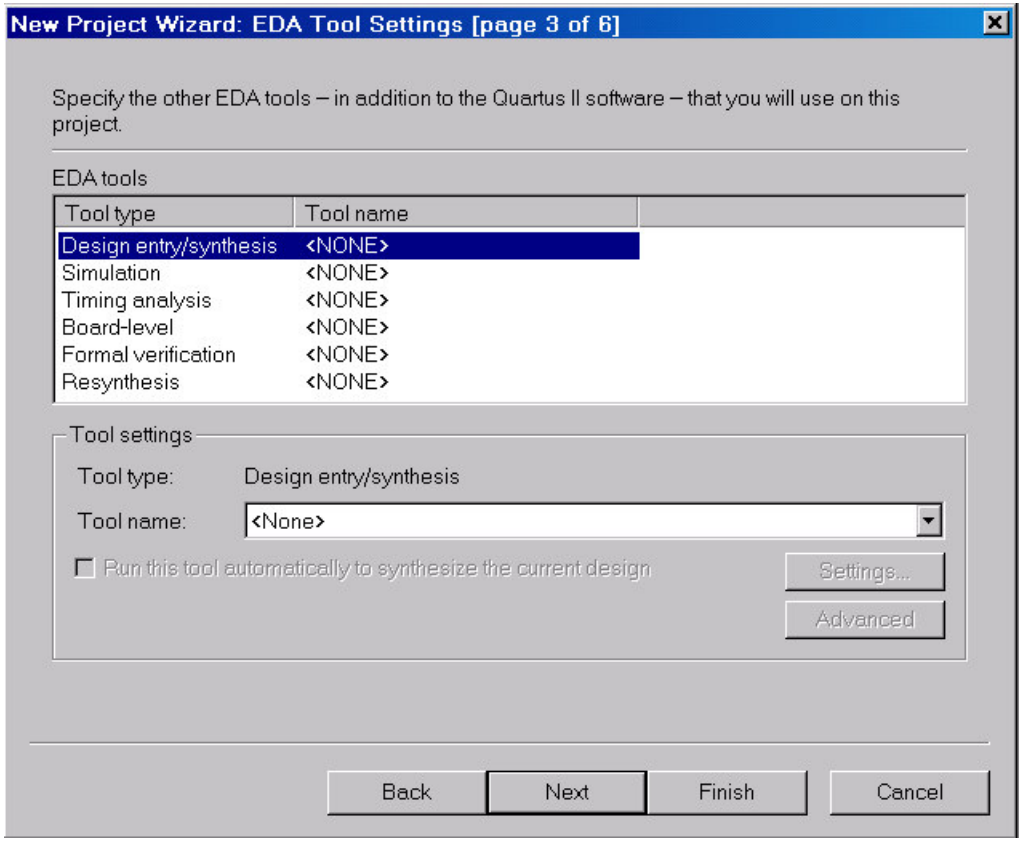
User Libraries

Add User Library Paths

- **User Libraries**
- **MegaCore®/AMPP Sm Libraries**
- **Pre-Compiled VHDL Packages**



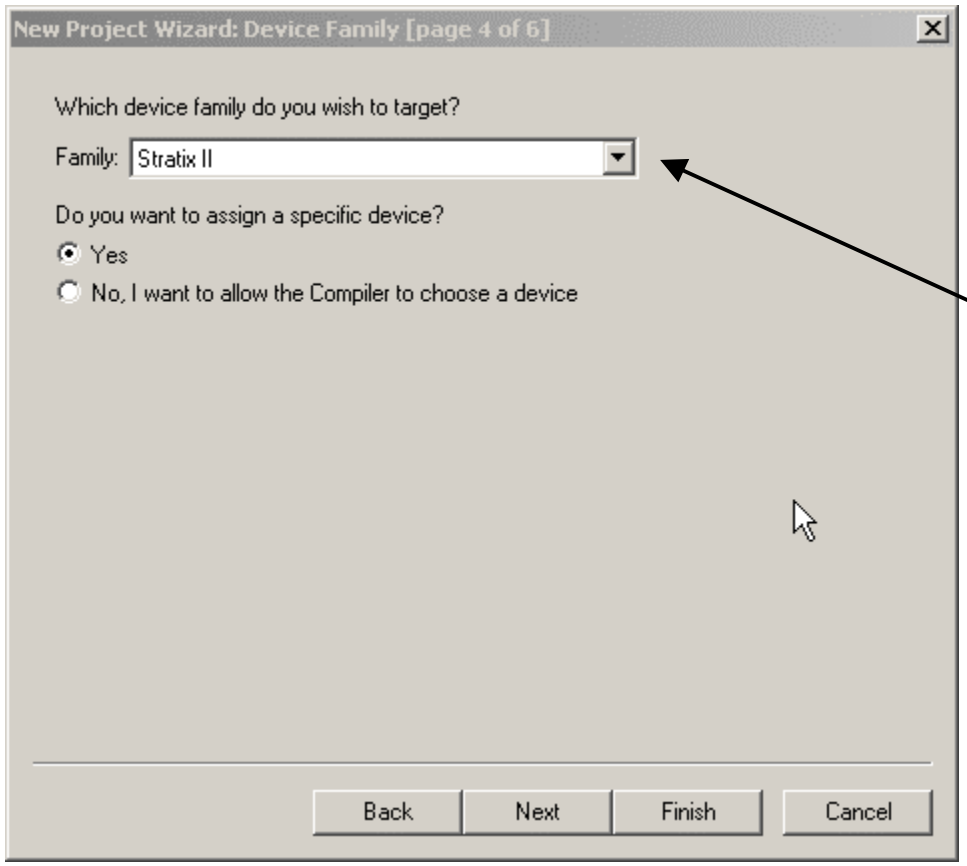
EDA Tool Settings



Choose EDA Tools

Add or Change Settings Later

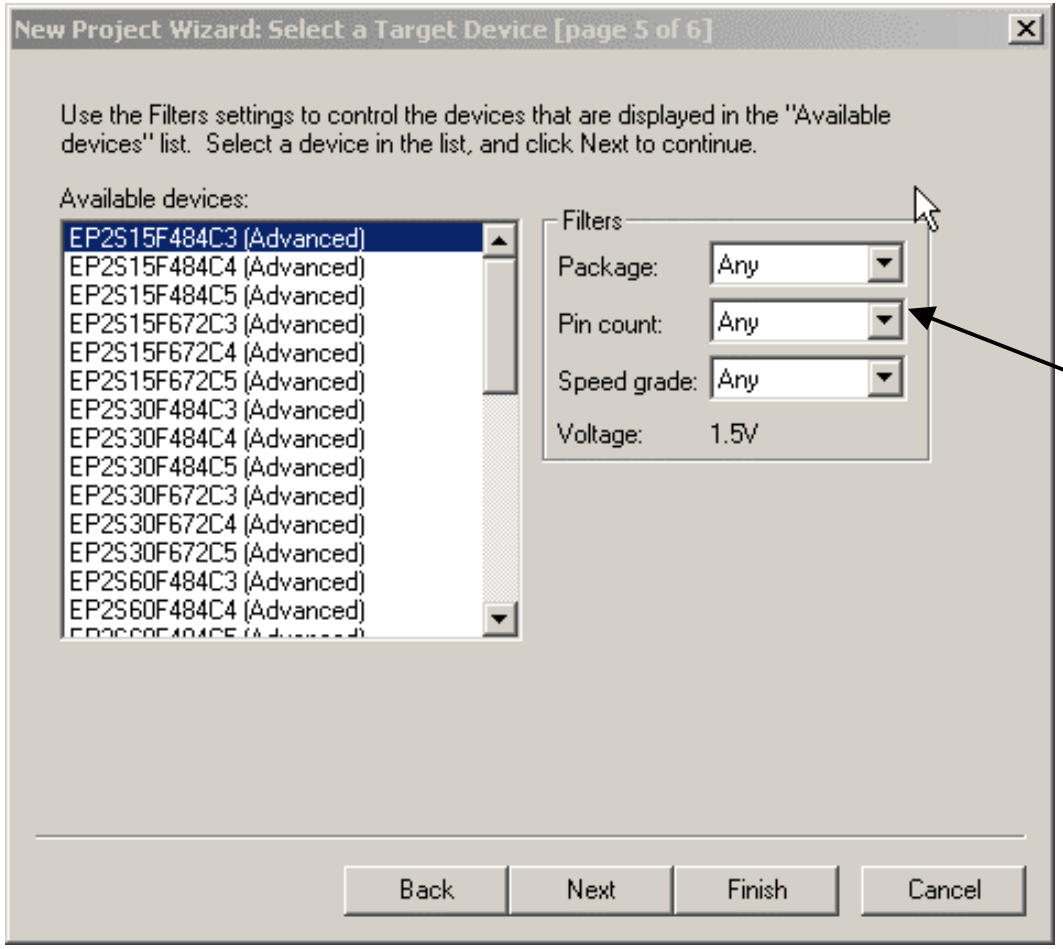
Device Selection



Choose the Device Family

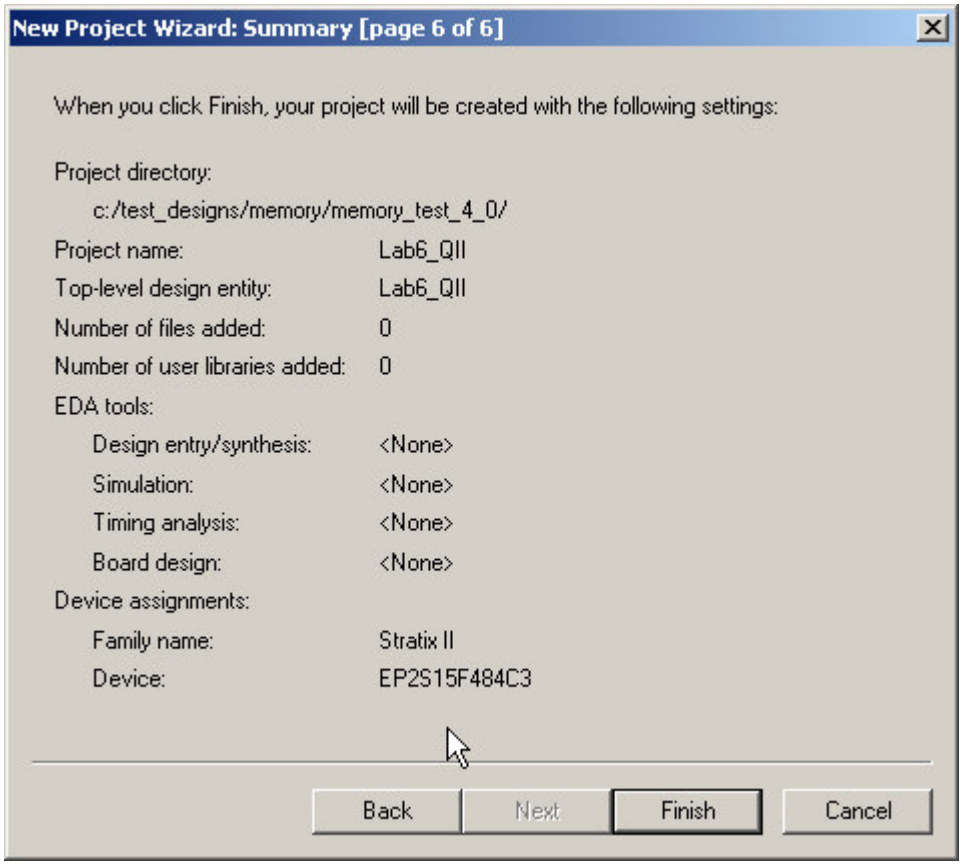
Auto Device or Choose a Specific Device

Device Selection



Use Filters for Auto Selection or Picking a Specific Device

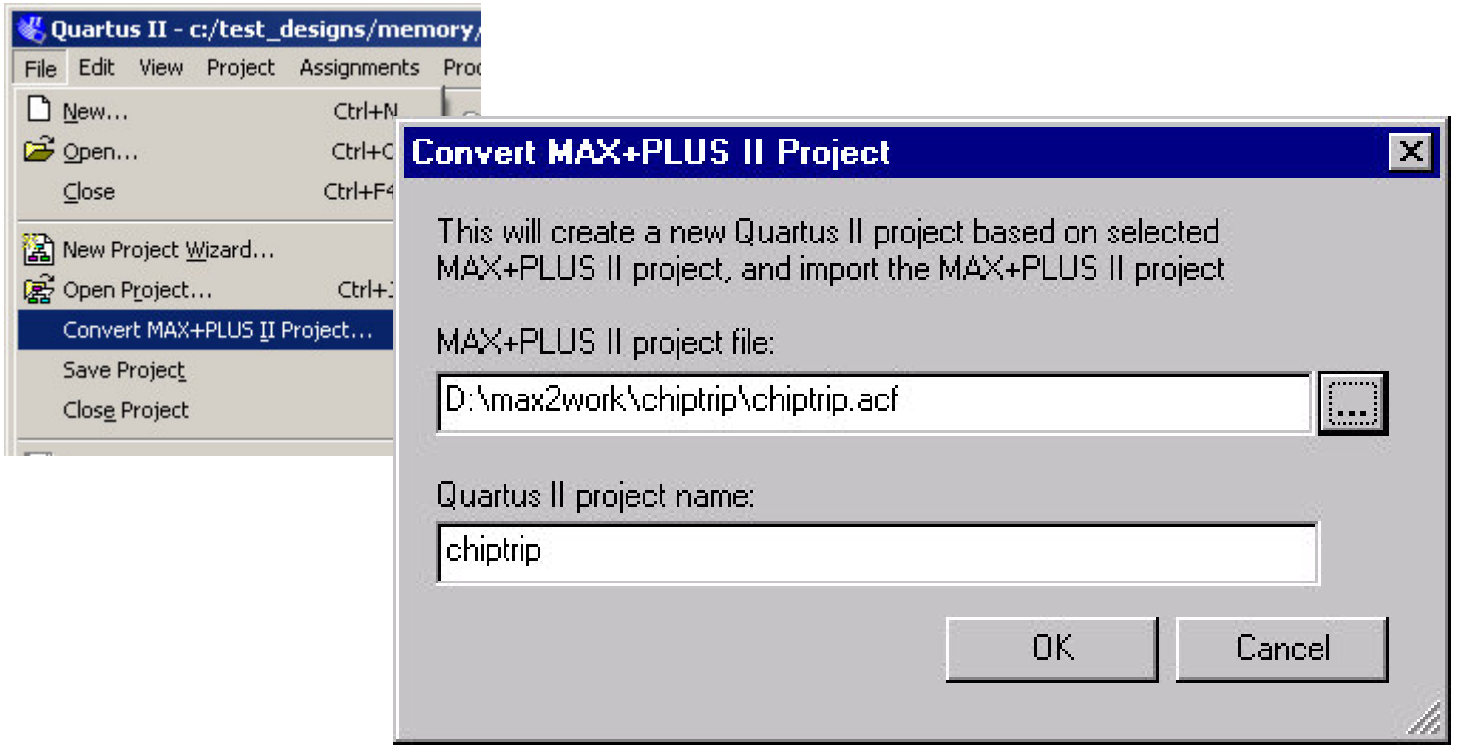
Done!



Review Results & Click on Finish

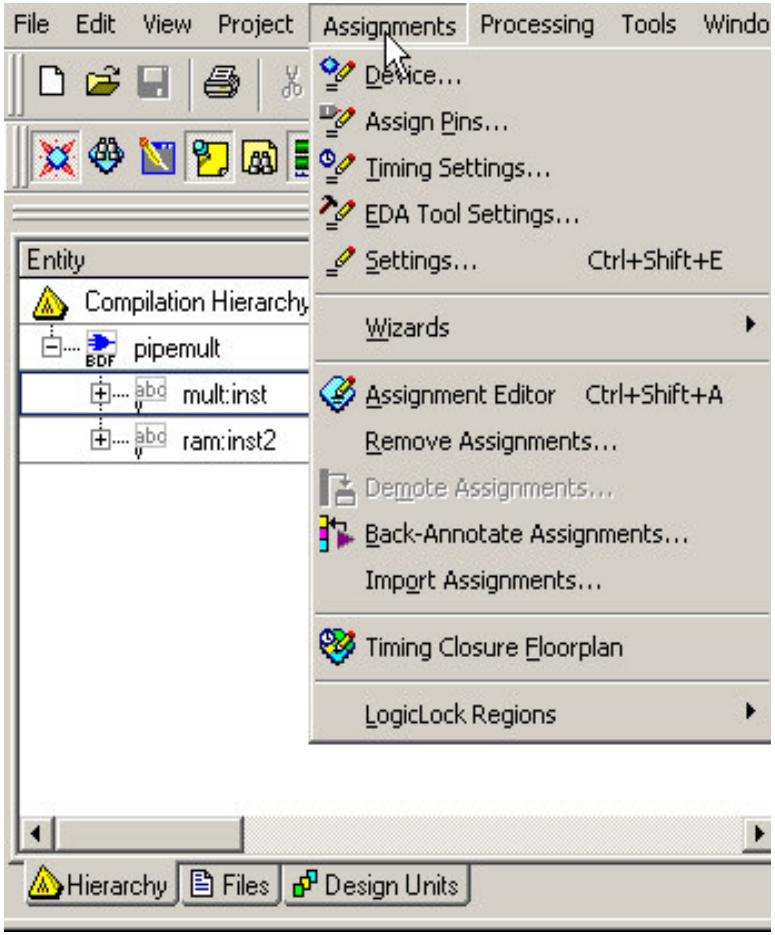
MAX+PLUS II to Quartus II

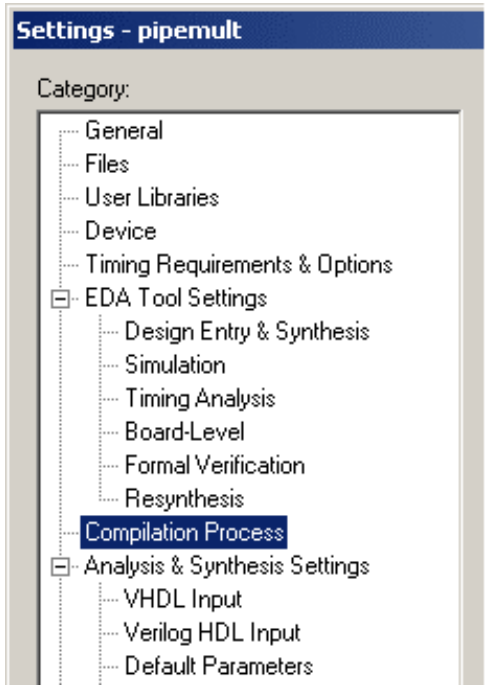
- Convert MAX+PLUS II Projects into Quartus II Projects
- Assignments Automatically Translated



Assignments Menu

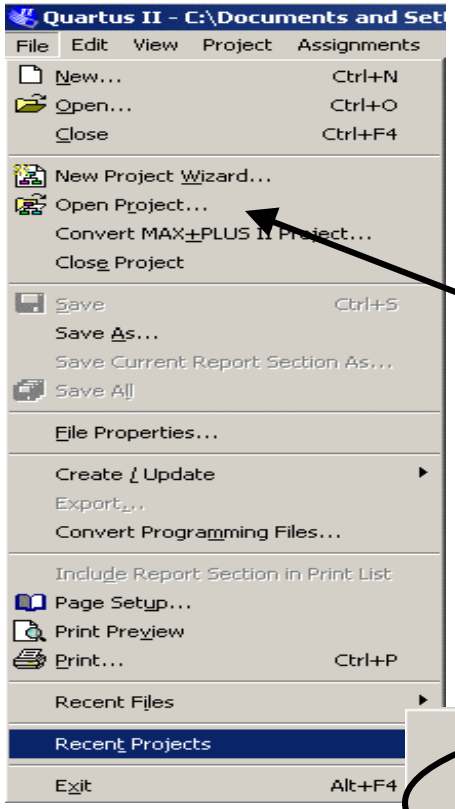
- Accesses All Assignments & Settings
- Opens Settings Dialog Box

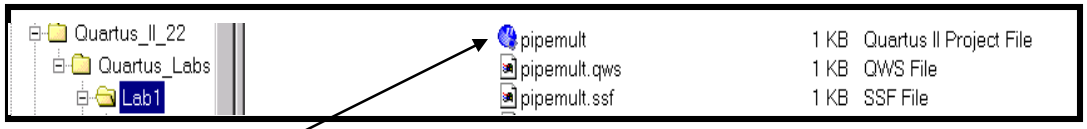




Editing Project Settings

- Open Existing Project First to Edit Settings





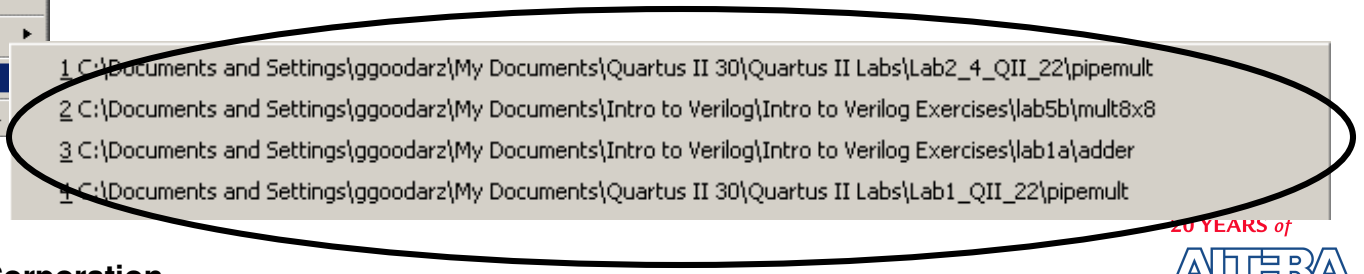
Double-Clicking the .Quartus File Will Auto Launch Quartus II

OR

File ⇒ Open Project

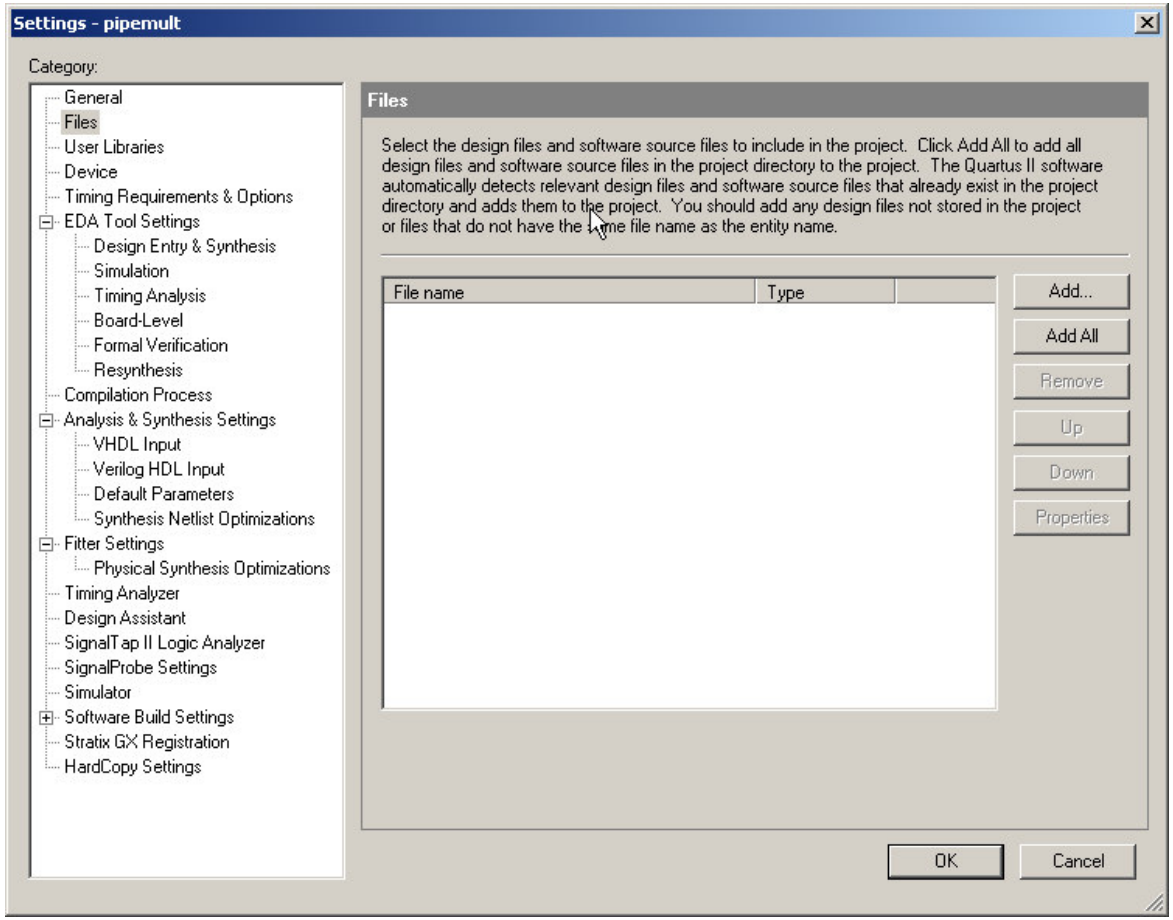
OR

Select from Most Recent Projects List



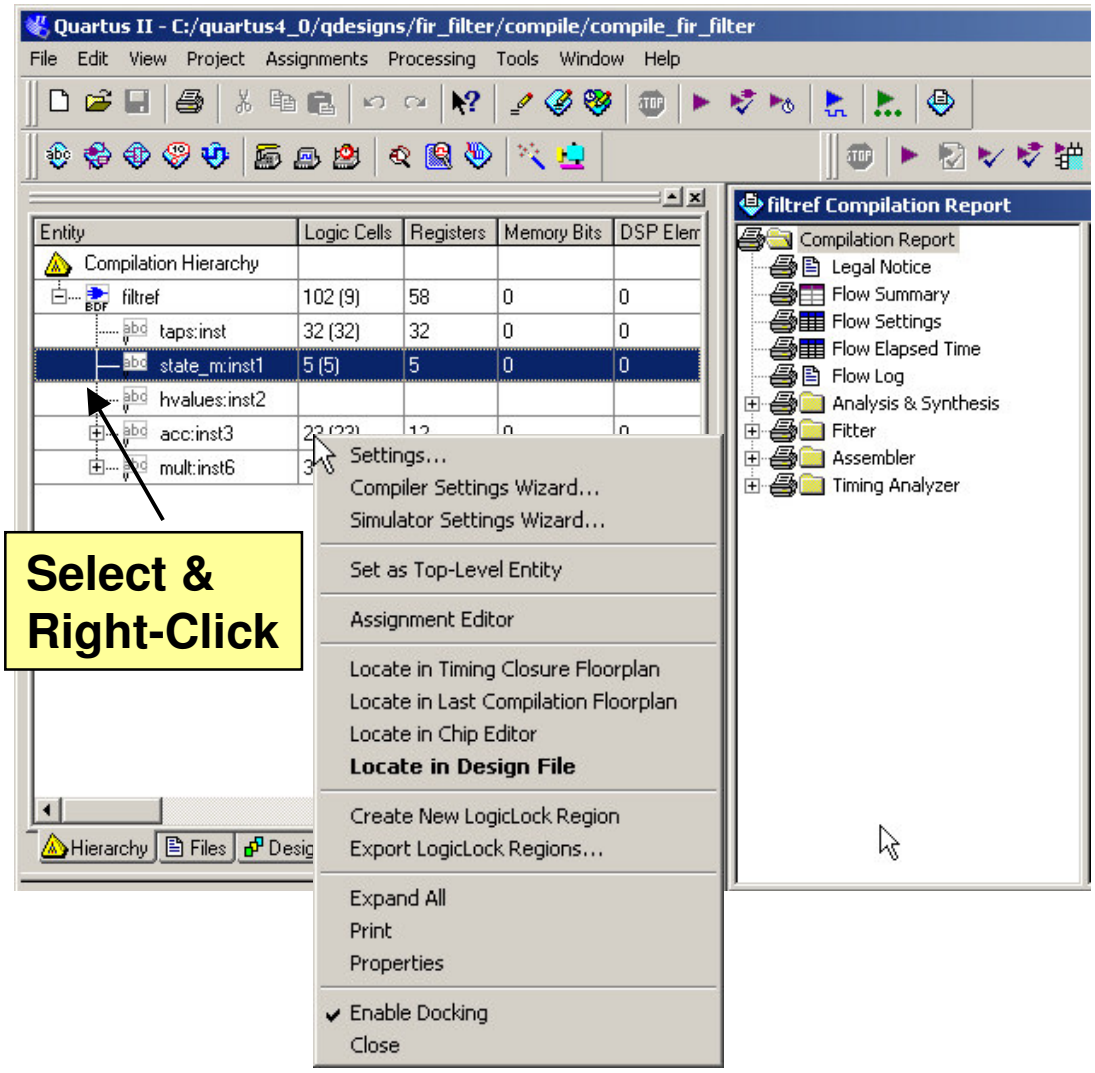
Editing Project Settings (cont.)

- Assignments ⇒ Settings



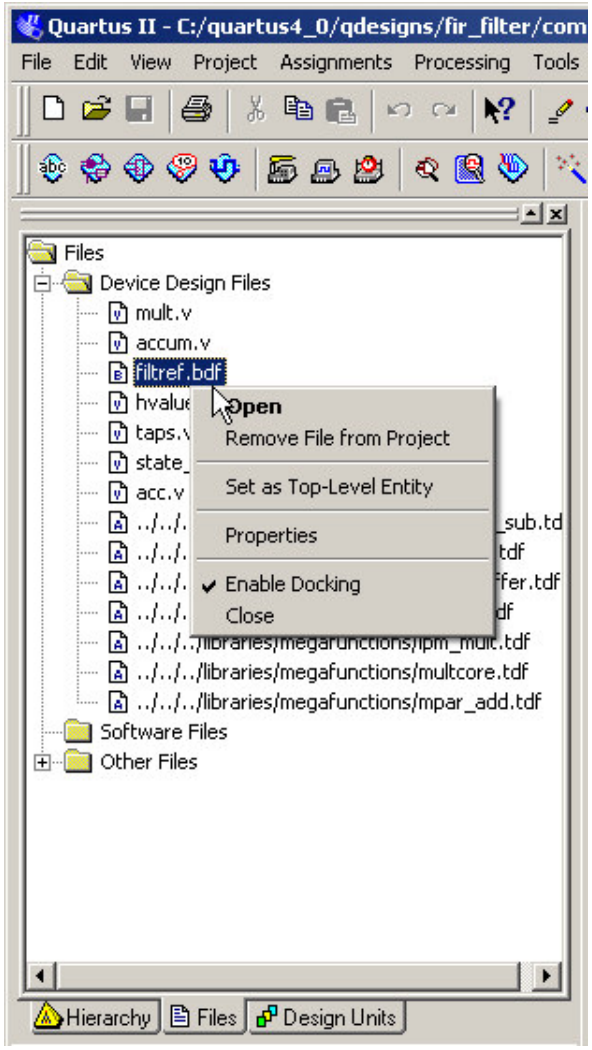
Change Settings
- **Rename**
- **Add/Remove Files**
- **Libraries**
- **VHDL '87, '93?**
- **Verilog '95, '01?**
- **EDA Tool Settings**
- **Timing Settings**
- **Compiler Settings**
- **Simulator Settings**

Project Navigator – Hierarchy Tab



- Displays Project Hierarchy after Project Is Analyzed
- Uses
  - Set Top-Level Entity
  - Make Assignments
  - Locate in Design Files, Floorplan or Chip Editor
  - View Resource Usage
  - Drag & Drop Hierarchical Blocks

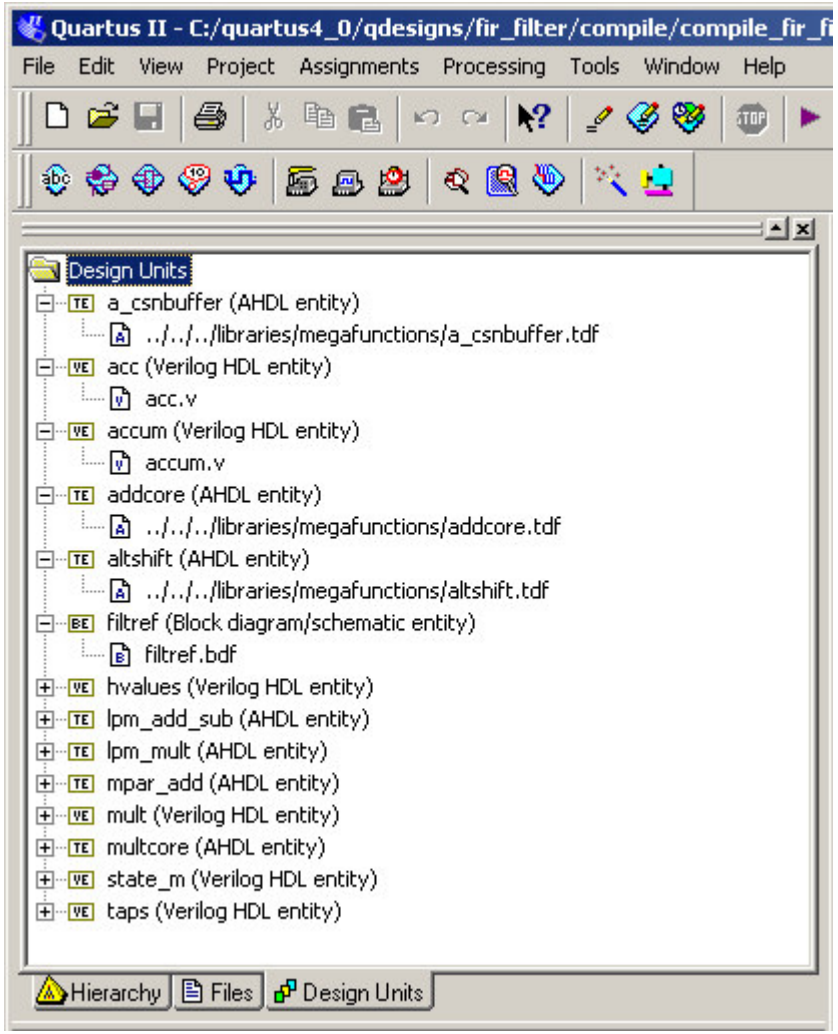
Project Navigator – Files Tab



- Shows All Files in Project
  - All Source Files Appear under Design Files
- Uses
  - Open Files
  - Remove Files from Project
  - Set New Top-Level Entity

Project Navigator - Design Units Tab

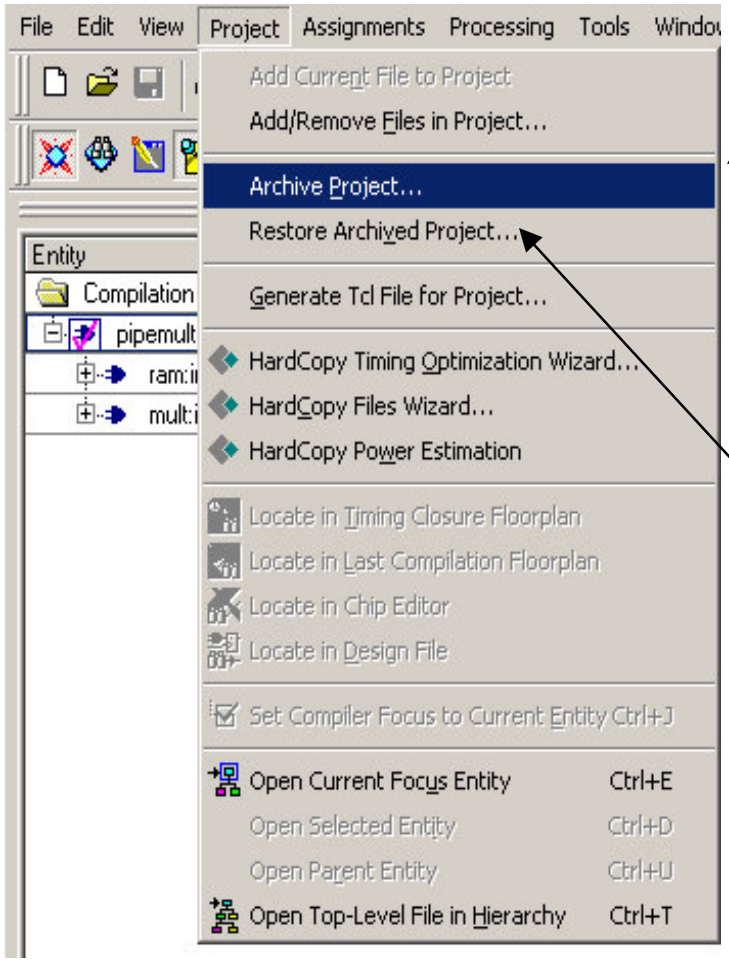
- Displays each Design Unit & Type
  - VHDL Entity
  - VHDL Architecture
  - Verilog Module
  - AHDL Subdesign
  - Block Diagram Filename
- Details the File which Instantiates Design Unit



Project Files

- Quartus Project File (QPF)
  - Quartus II Version
  - Time Stamp
  - Active Revision
- Quartus Default File (QDF)
  - Project Defaults
  - Name: assignment_defaults.qdf
  - Local or Bin Directory
    - Local Read First

Project Archive & Restore



Archive Project Creates:

1. **<project>.QAR**
   - Compressed Archive File
   - Options to Choose which Files Get Archived
2. **<project>.QARLOG**
   - Log of Archive Activity

Restore Archived Project:

- Restores Complete Project & Library Function Files from **<project>.QAR** File

Summary

- Use Project Wizard to Create New Projects
- Use Assignments Menu Dialog Boxes to
  - Edit Existing Project Settings
  - Edit Third-Party Tools
- Use Project Navigator to Study File & Entity Relationships within Project

20 YEARS *of*

ALTERA®

INNOVATION

Designing with Quartus II

Quartus II Compilation

QUARTUS® II

Quartus II Compilation

- Synthesis
- Fitting
- Generating Output
  - Timing Analysis Output Netlist
  - Simulation Output Netlists
  - Programming/Configuration Output Files

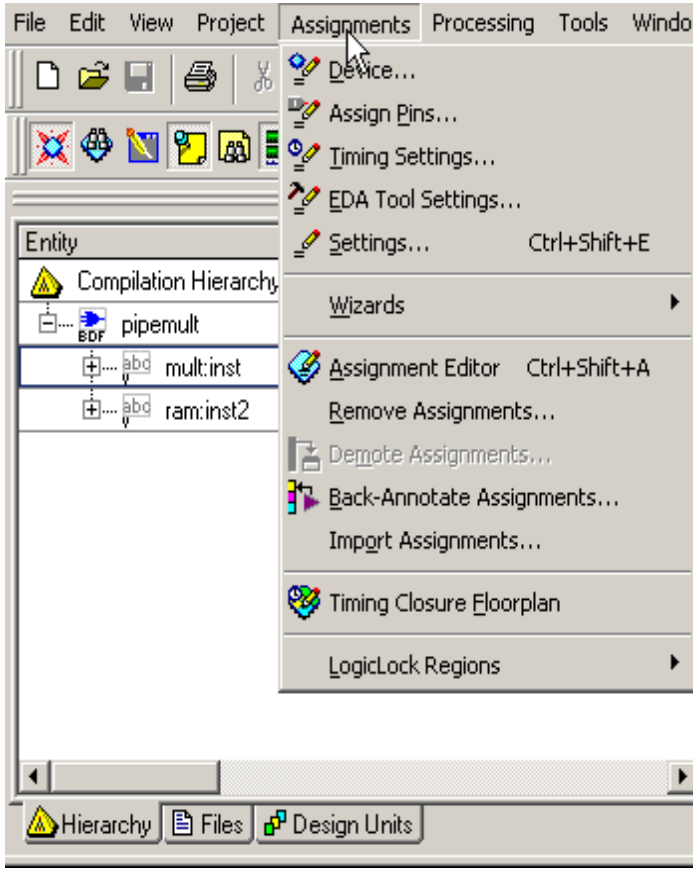
Controlling Synthesis & Fitting

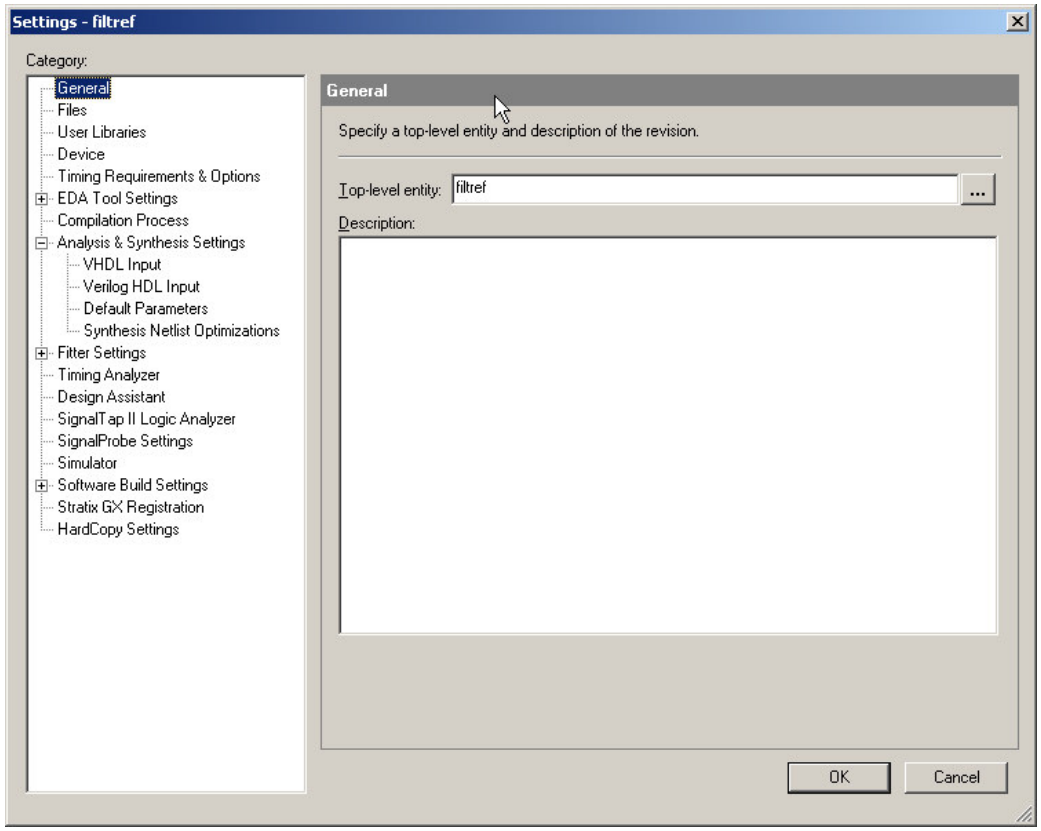
- Two Methods to Control Logic Synthesis & Fitting Operations
  - Settings
    - Project-Wide Switches
  - Assignments (i.e. Logic Options; Constraints)
    - Individual Entity/Node Controls

Settings

- Examples
  - Device Selection
  - Pin Assignments
  - Synthesis Optimization
  - Fitter Optimization
  - Physical Synthesis
- Located in Settings Dialog Box (Assignments Menu)

Editing Settings

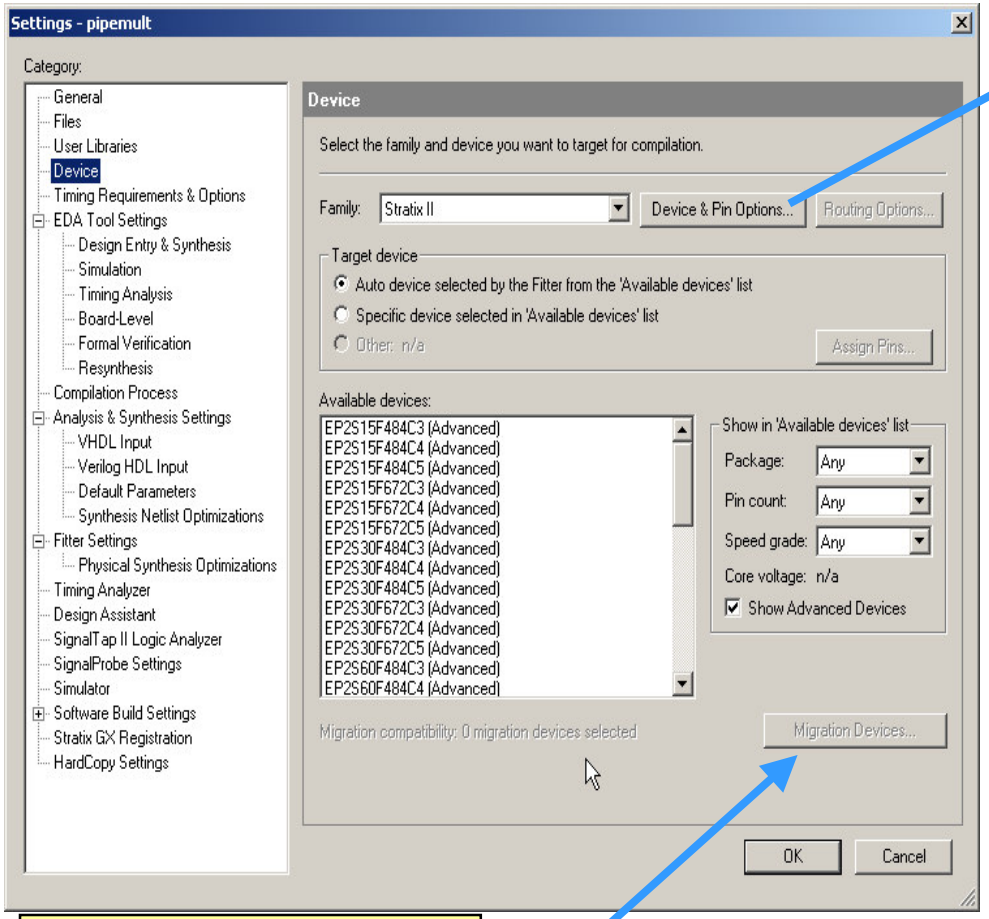


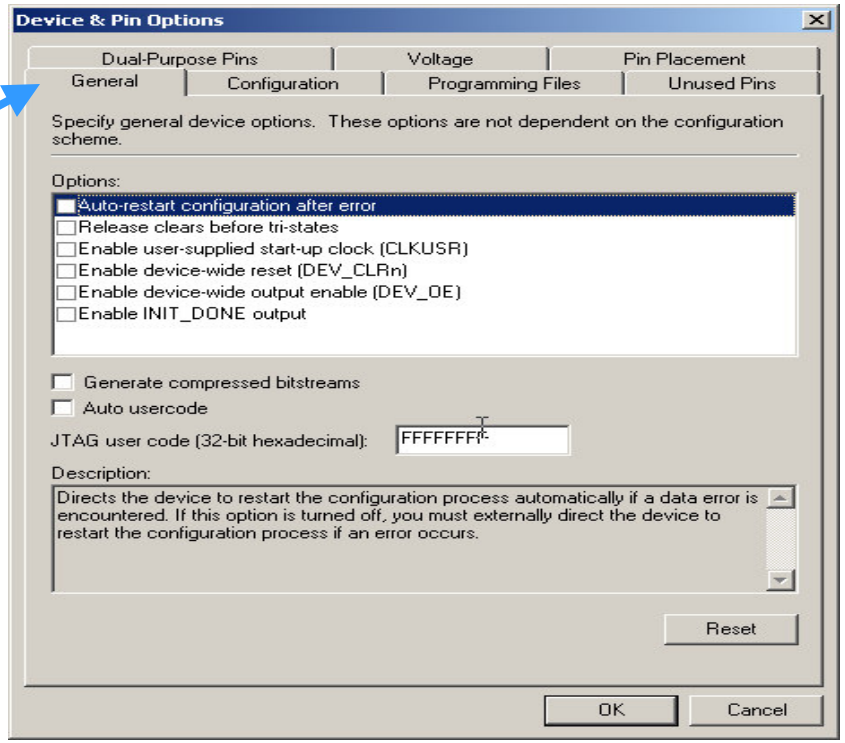


Assignments ⇒ Settings

Several Options Control Compilation & Device Options

Device Selection





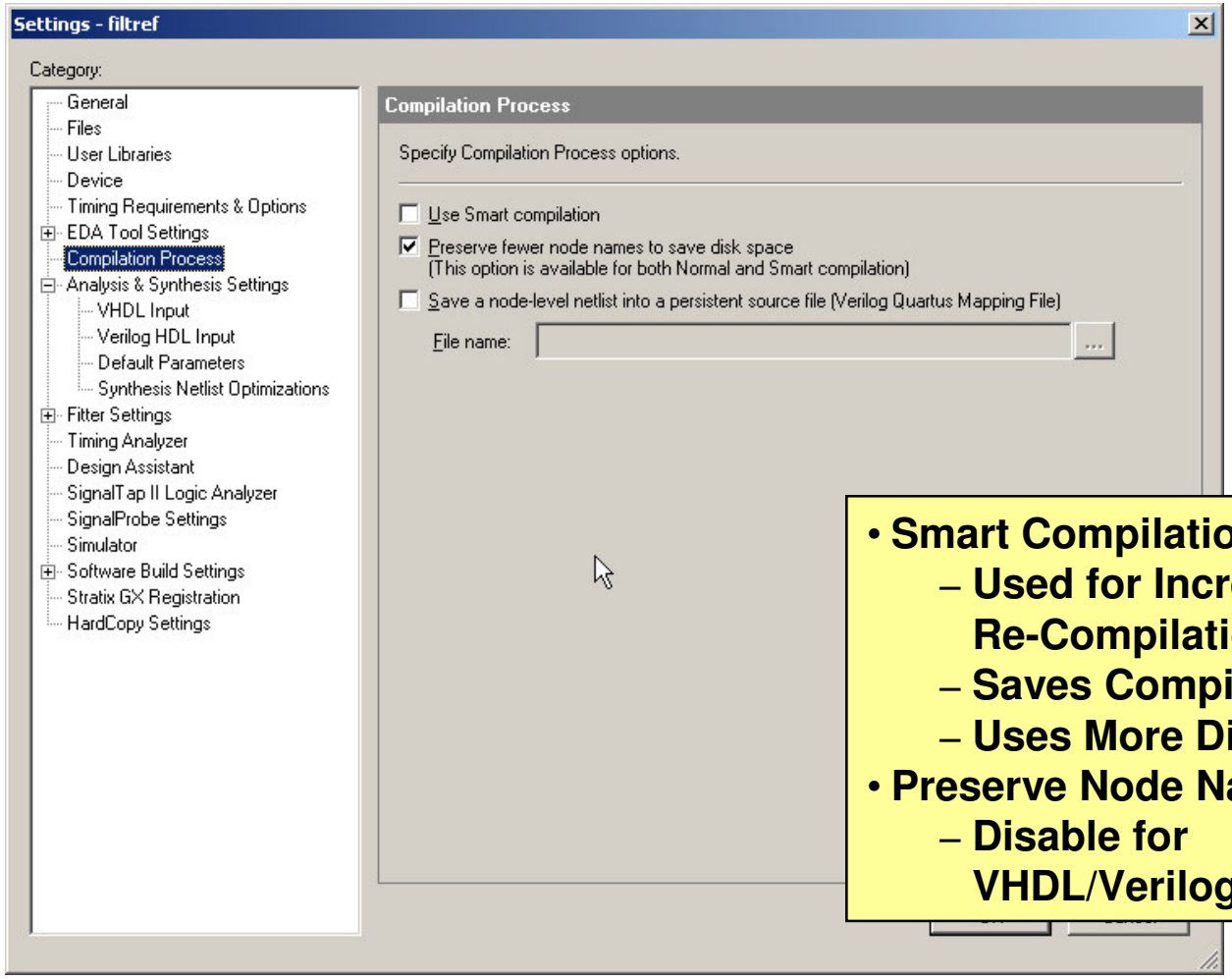
Change Device Type

Migration Devices

Specify Device & Pin Options

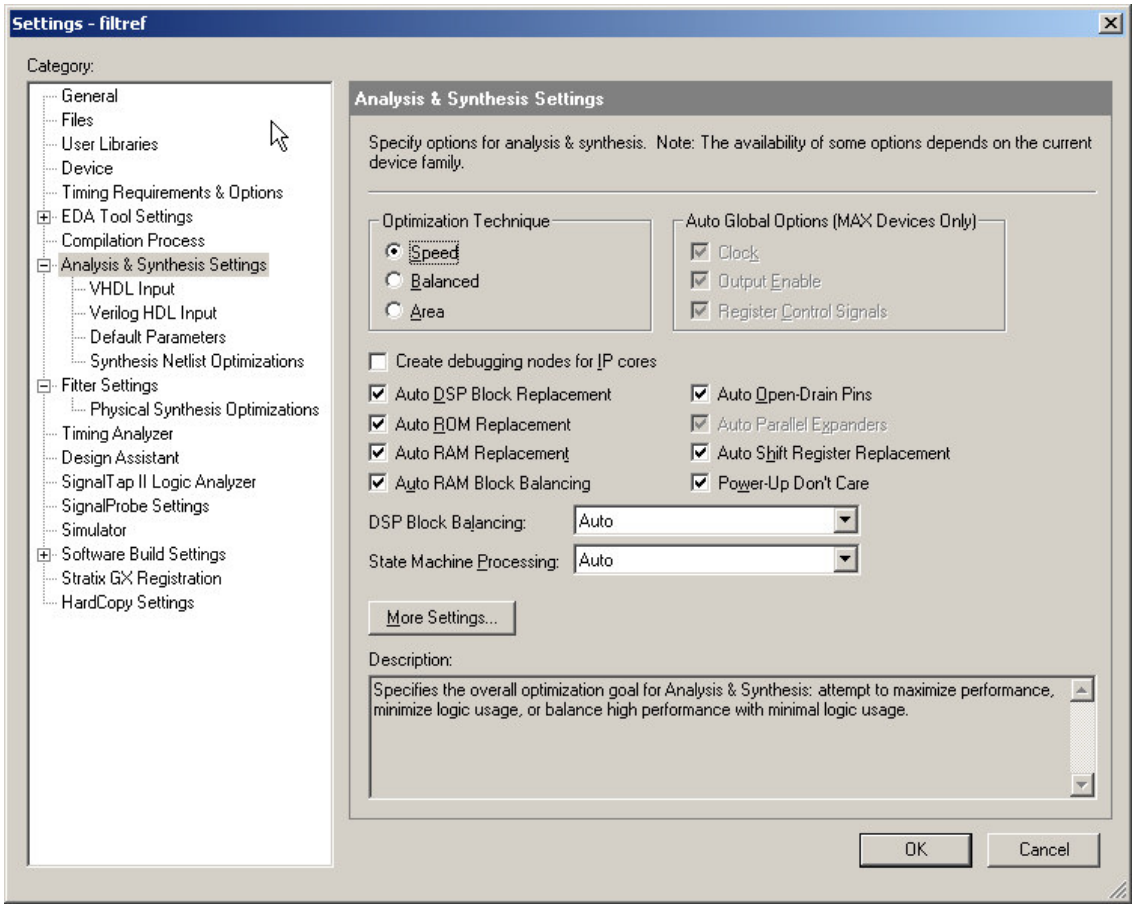
- General
- Configuration
- Programming Files
- Unused Pins
- Dual Purpose Pins
- Voltage

Compilation Process



- **Smart Compilation**
  - **Used for Incremental Re-Compilation**
  - **Saves Compiler Time**
  - **Uses More Disk Space**
- **Preserve Node Names**
  - **Disable for VHDL/Verilog Synthesis**

Synthesis Options



Global Optimization Goal

- Select Speed vs. Area or Balanced

Logic Replacement

- Replace Logic with Equivalent Megafunction

State Machine Processing

- Auto, One-Hot or Minimal Bit

Seed

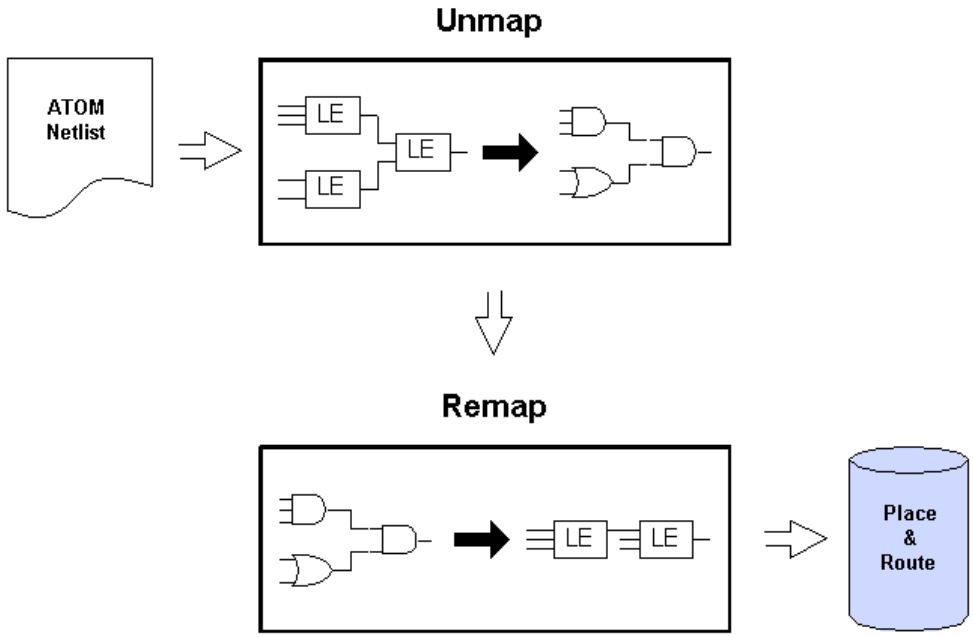
- Controls Initial Placement Configuration
- Different Seeds Give Different Fmax

Synthesis Netlist Optimizations

- Further Optimize Netlists during Synthesis
- Types
  - WYSIWYG Primitive Resynthesis
  - Gate-Level Register Retiming

WYSIWYG Primitive Resynthesis

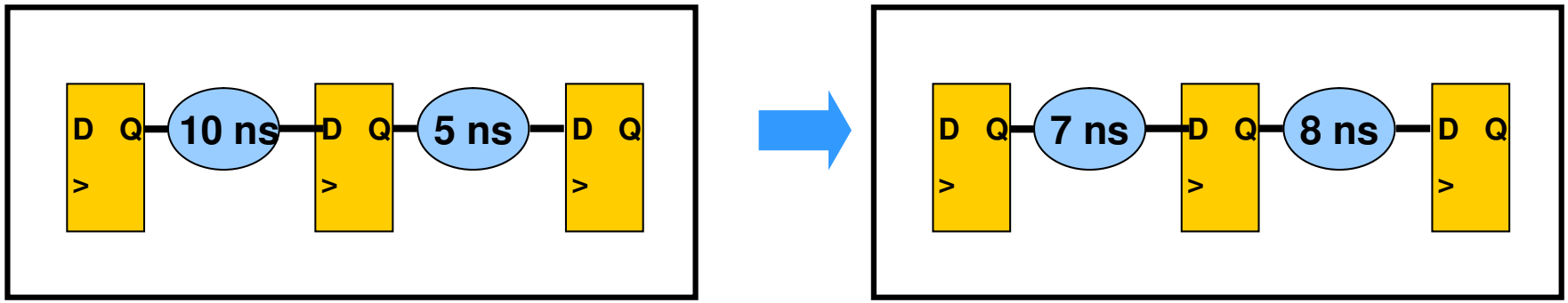
- Unmaps 3rd-Party Atom Netlist Back to Gates & then Remaps to Altera Primitives
  - Unavailable when Using Integrated Synthesis
- Considerations
  - Node Names May Change
  - 3rd-Party Synthesis Attributes May Be Lost
    - Preserve/Keep
  - Some Registers May Be Synthesized Away



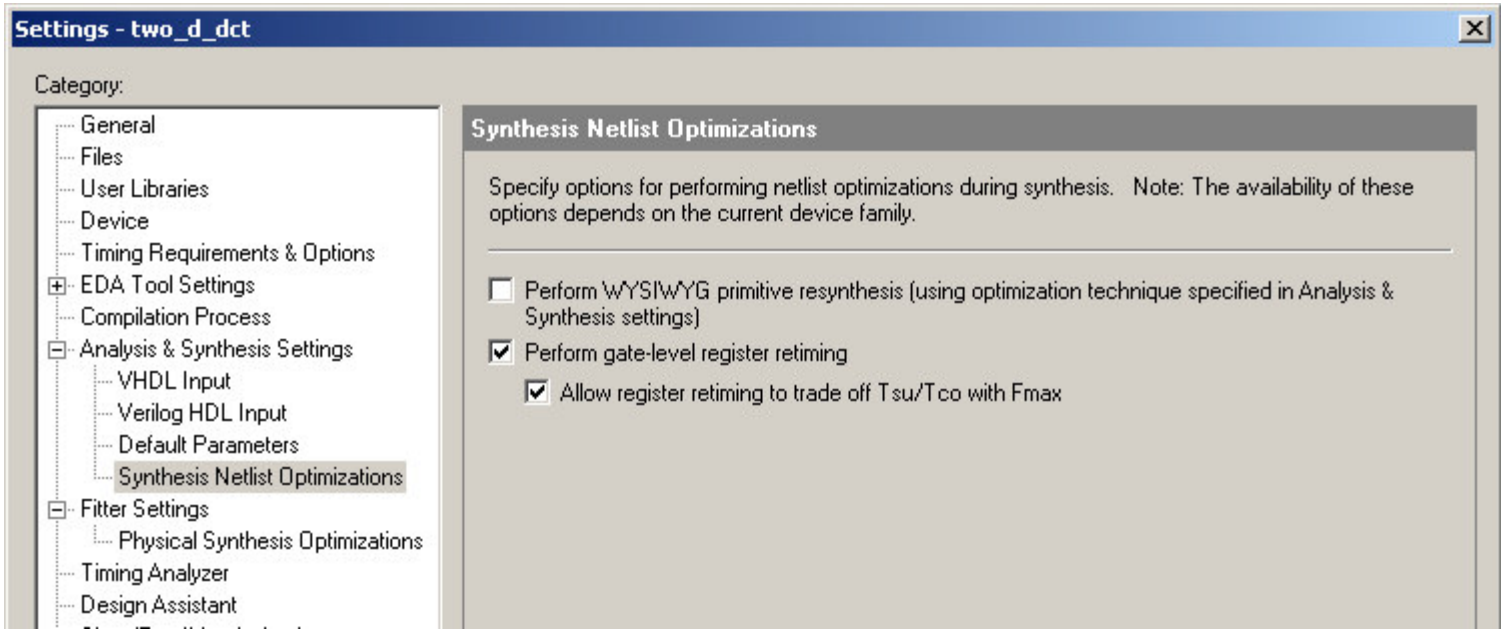
4-5 % Improvement

Gate-Level Register Retiming

- Moves Registers across Combinatorial Logic to Balance Timing
- Trades between Critical & Non-Critical Paths
- Makes Changes at Gate Level



Enabling Synthesis Optimizations



Created/Modified Nodes Noted in Compilation Report

Fitter Optimizations

Timing Driven Compilation

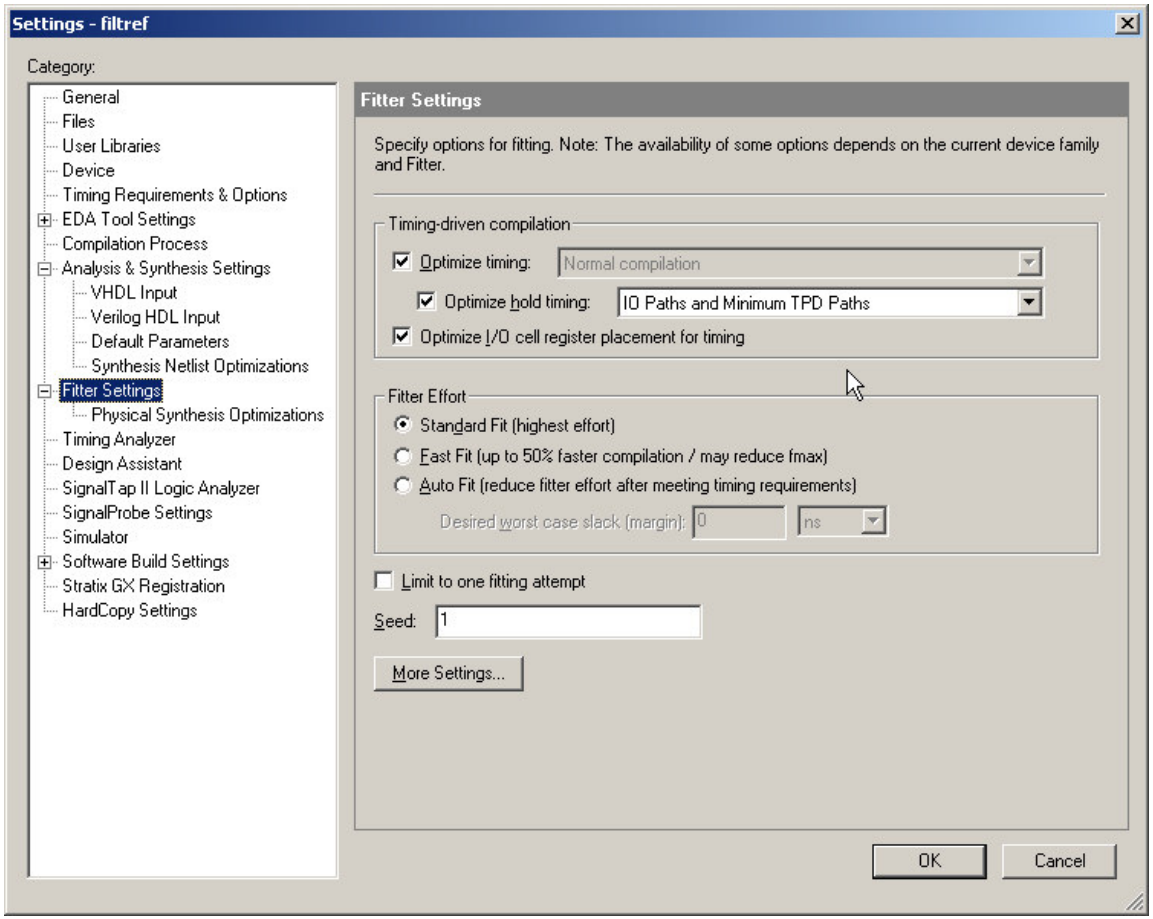
- Optimize Internal Timing
- Optimize I/O Register Placement

Compilation Speed/Fitter Effort

- Standard Fit (Highest Effort)
- Fast Fit (Faster Compile but Possibly Lesser Design Performance)
- Auto Fit (Compile Stops after Meeting Timing)
- One Fitting Attempt

Seed

- Controls Initial Placement Configuration

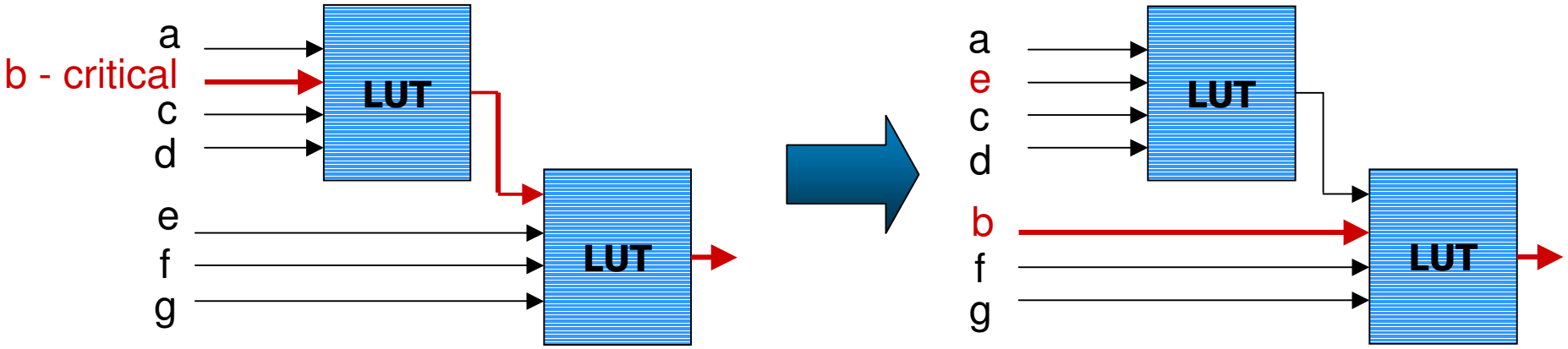


Physical Synthesis

- Re-Synthesis Based on Fitter Output
  - Makes Incremental Changes that Improve Results for a Given Placement in Altera Device
  - Compensates for Routing Delays from Fitter
    - Routing Delays Large Part of Typical Critical Path Delay
- Types
  - Combinational Logic
  - Registers
    - Register Duplication
    - Register Retiming

Combinational Logic

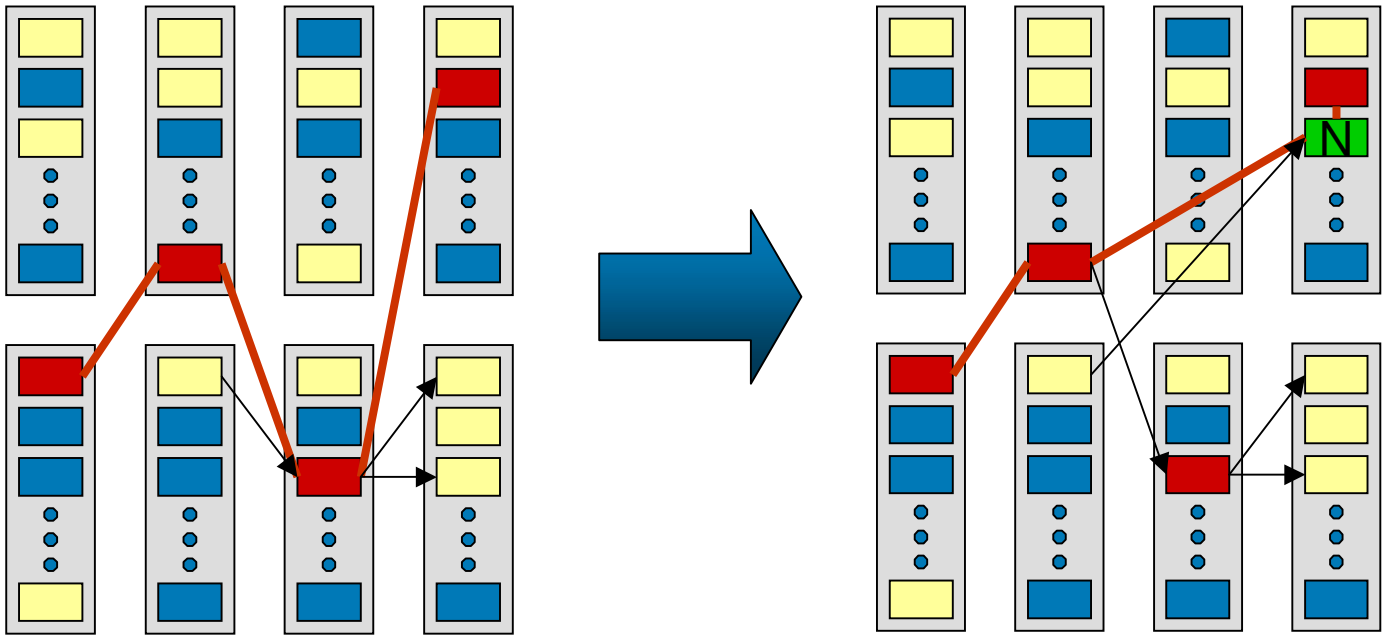
- Swaps Look-Up Table (LUT) Ports within LEs to Reduce Critical Path LEs



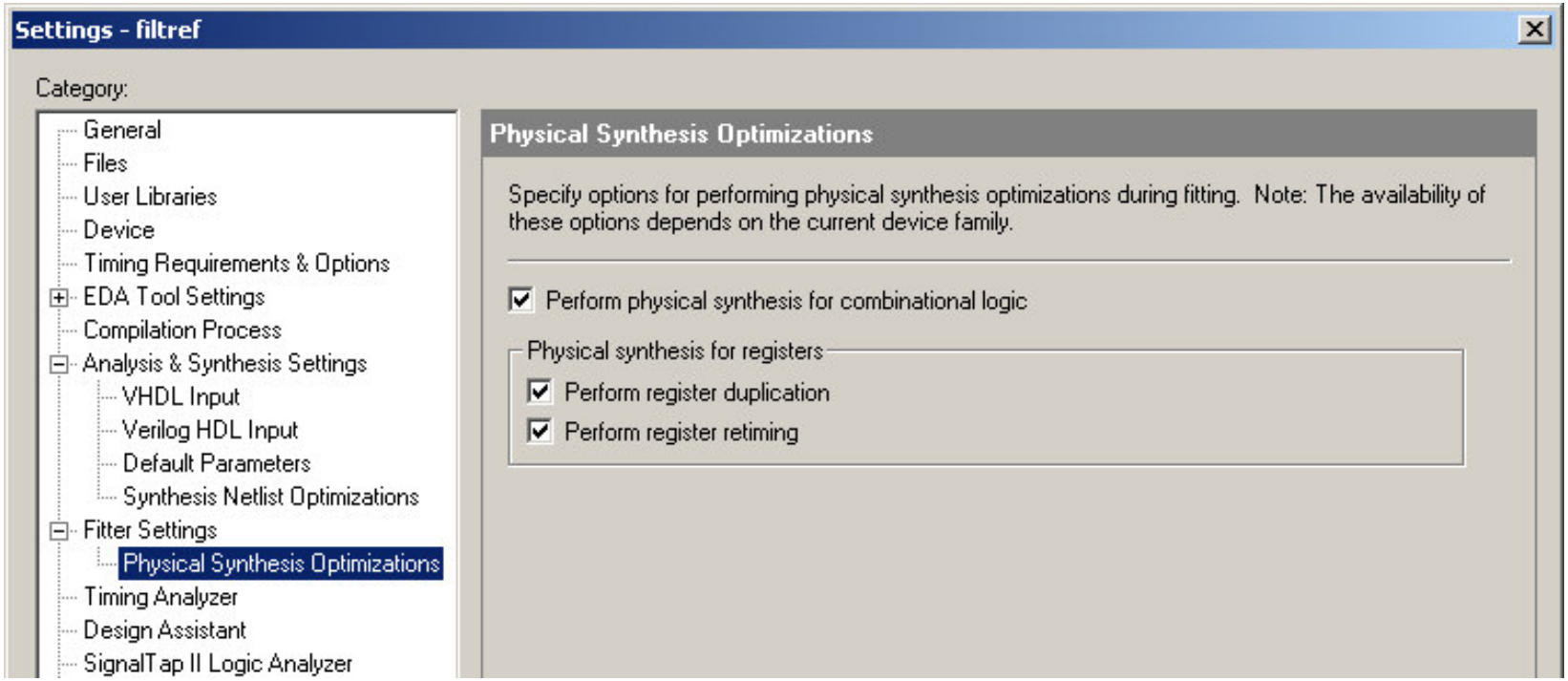
- Allows LUT Duplication to Enable Further Optimizations on the Critical Path

Register Duplication

- High Fan-Out Register Is Duplicated & Placed to Reduce Delay
  - Combinational Logic May Also Be Duplicated



Enabling Physical Synthesis



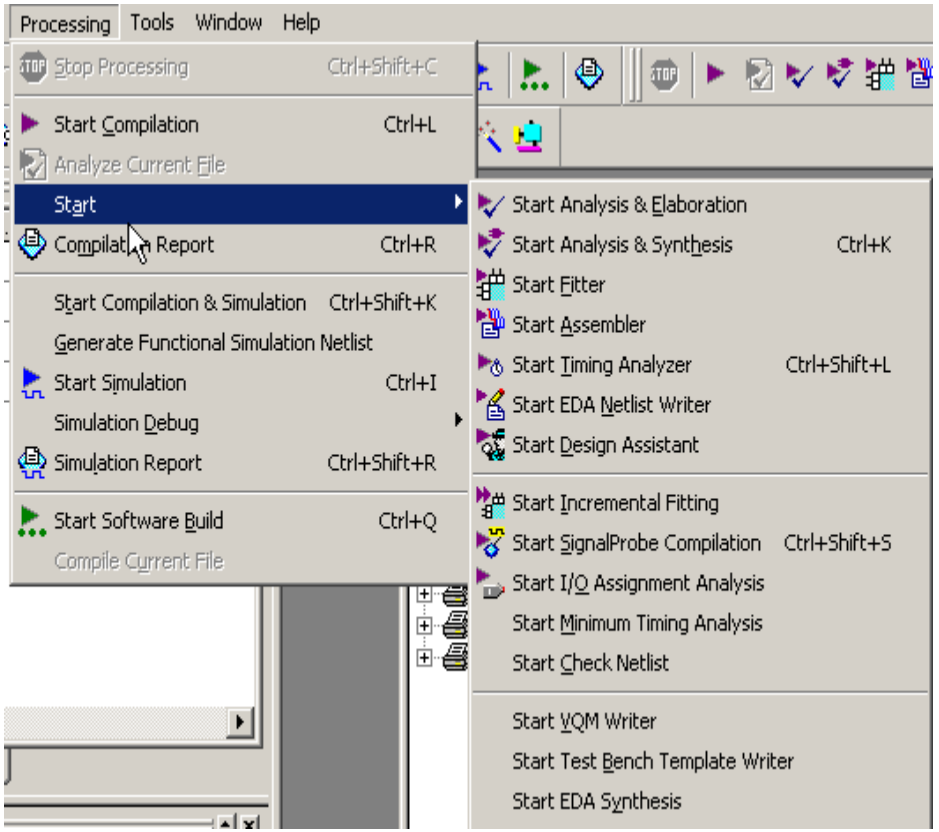
Created/Modified Nodes Noted in Compilation Report

More Netlist Optimizations Info

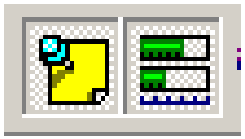
- Disable Optimizations on Specific Entities/Nodes
  - Set **Netlist Optimizations** Logic Option to **Never Allow**
  - Entities/Nodes Cannot Be Altered
- Prevent Nodes from Being Optimized Away
  - HDL Attributes
    - Use **PRESERVE** (Registers) or **KEEP** (Combinational Logic)
  - **Preserve** Logic Option
- Unsafe Registers & Logic Remain Unaffected
  - Registers with Certain Timing Constraints
  - Registers that Feed Clocks, Asynchronous Controls
  - Registers Fed by Registers in Other Clock Domains

Processing Options

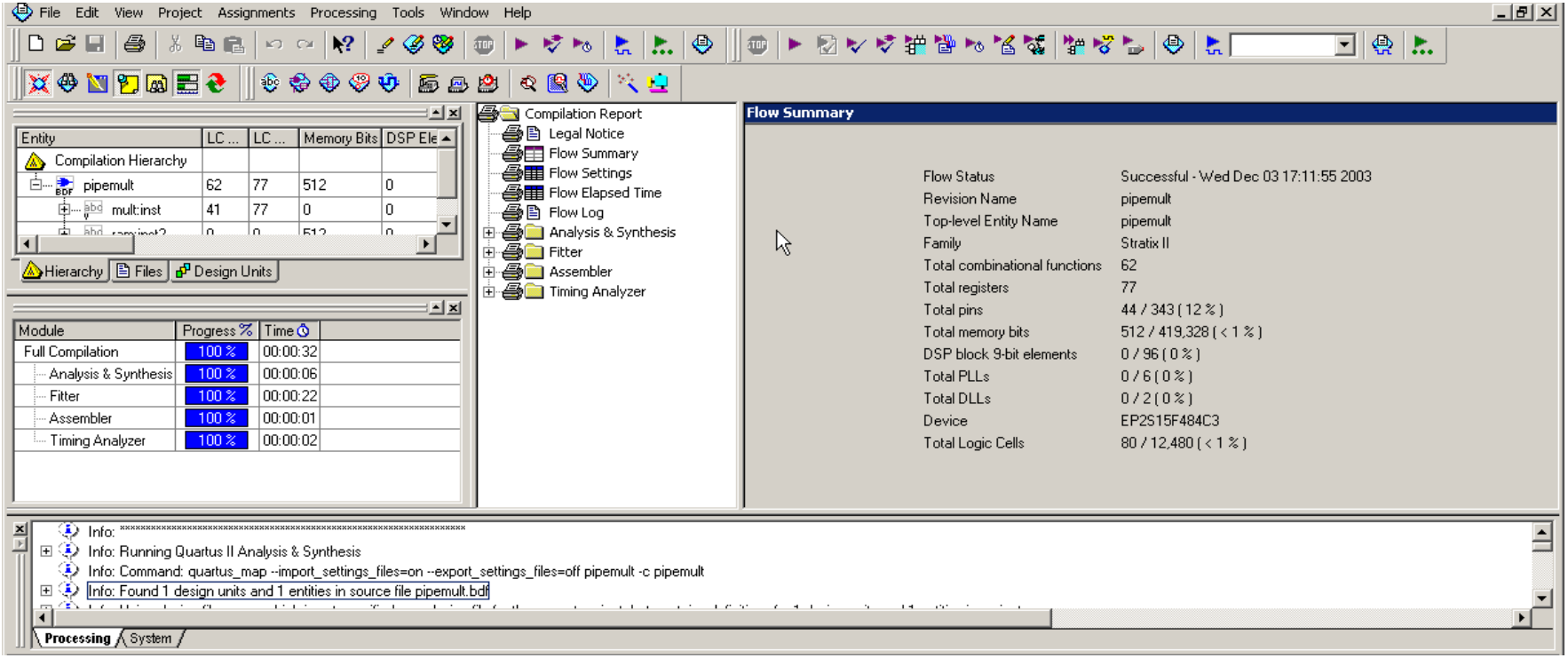
- Start Compilation
  - Perform Full Compilation
- Start Analysis & Elaboration
  - Check Syntax & Build Database Only
- Start Analysis & Synthesis
  - Synthesize Code & Estimate Timing
- Start Fitter
- Start Assembler
- Start Timing Analysis
- Start I/O Assignment Analysis
- Start Design Assistant



Status & Message Windows



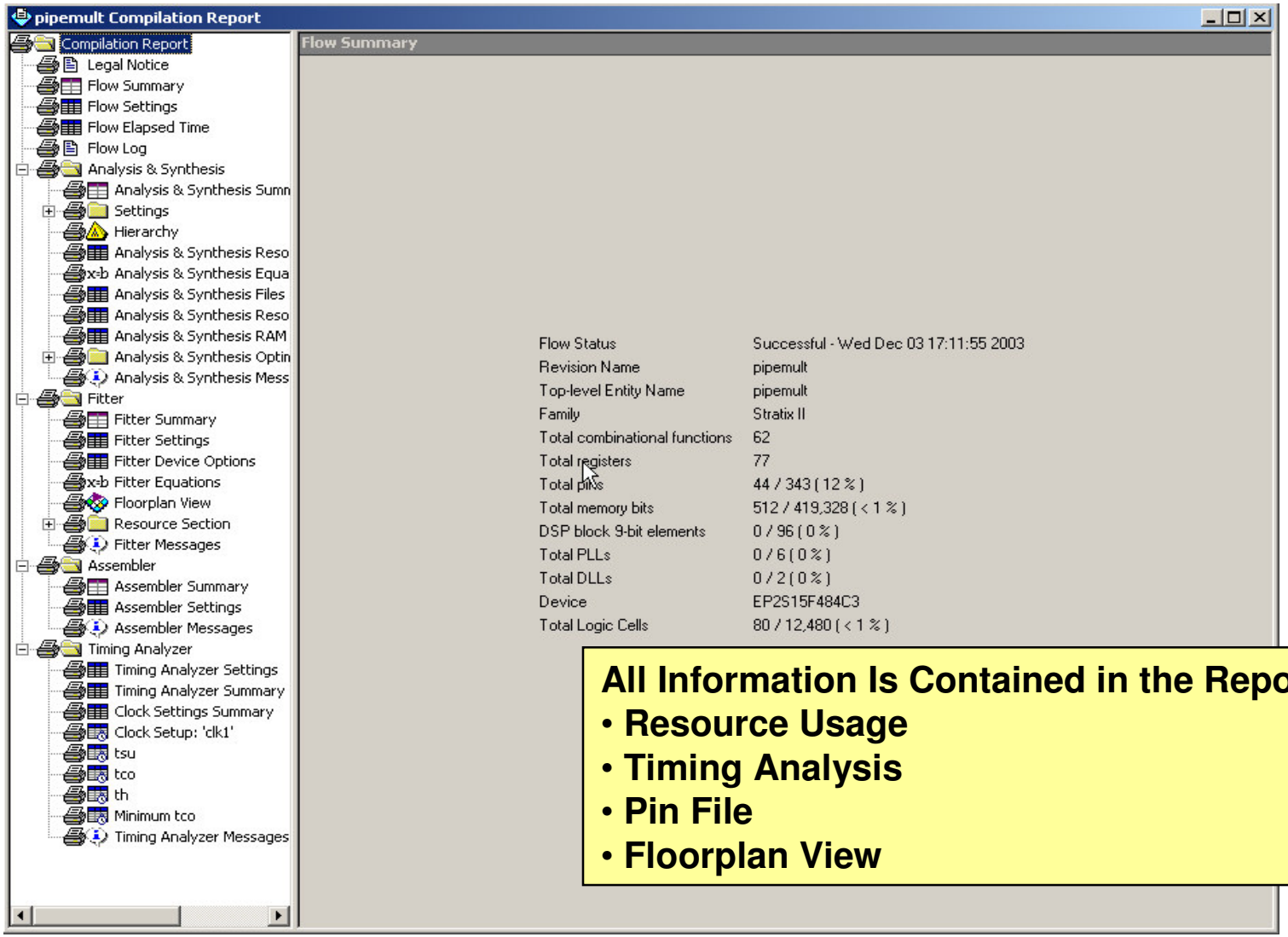
Hit the Start Button



- **Status Bars Scroll to Indicate Progress**
- **Message Window Displays Informational, Warning, & Error Messages**

20 YEARS of ALTERA INNOVATION

Compilation Report

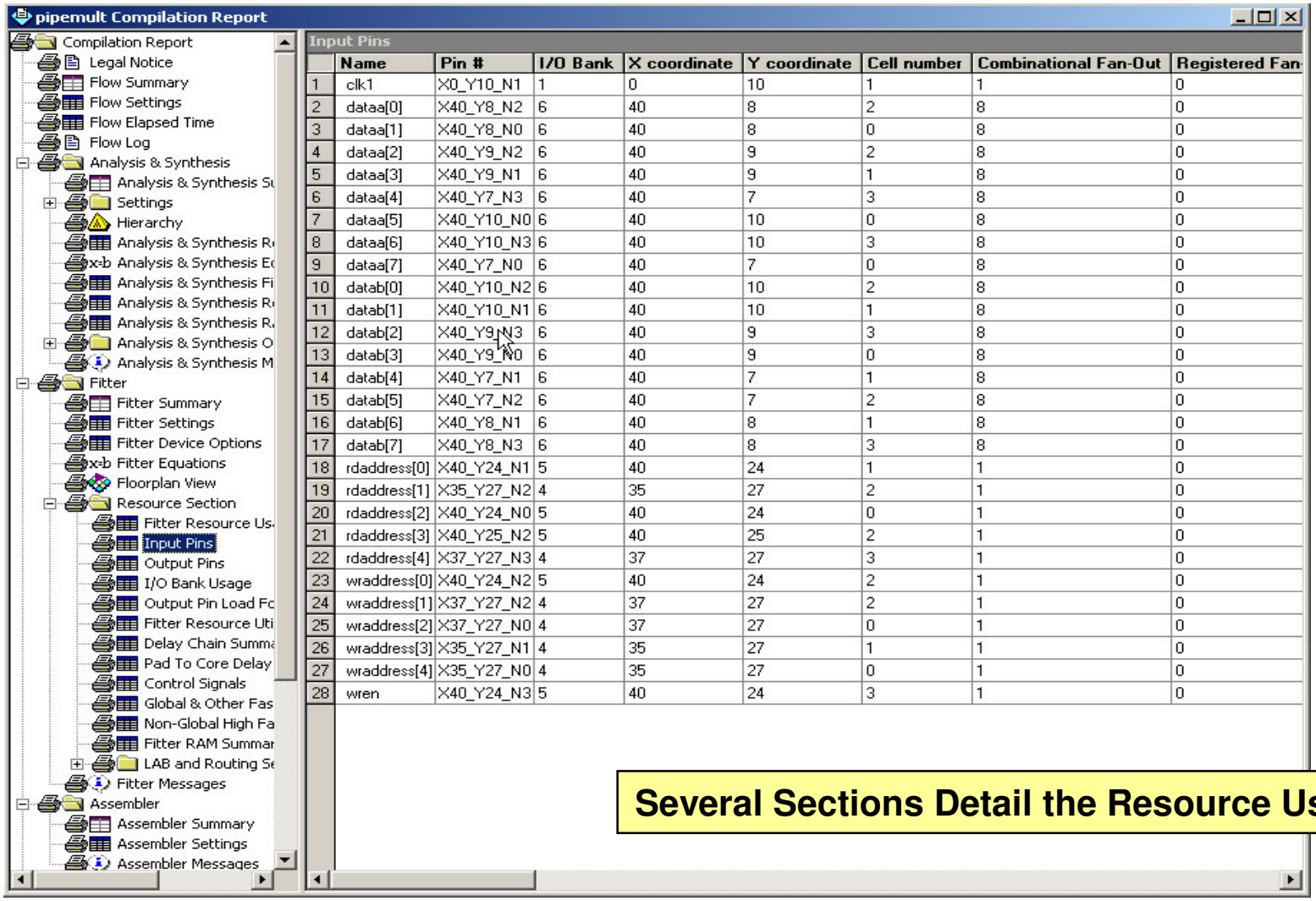


All Information Is Contained in the Report

- **Resource Usage**
- **Timing Analysis**
- **Pin File**
- **Floorplan View**

20 YEARS *of* ALTERA INNOVATION

Resource Usage



Input Pins

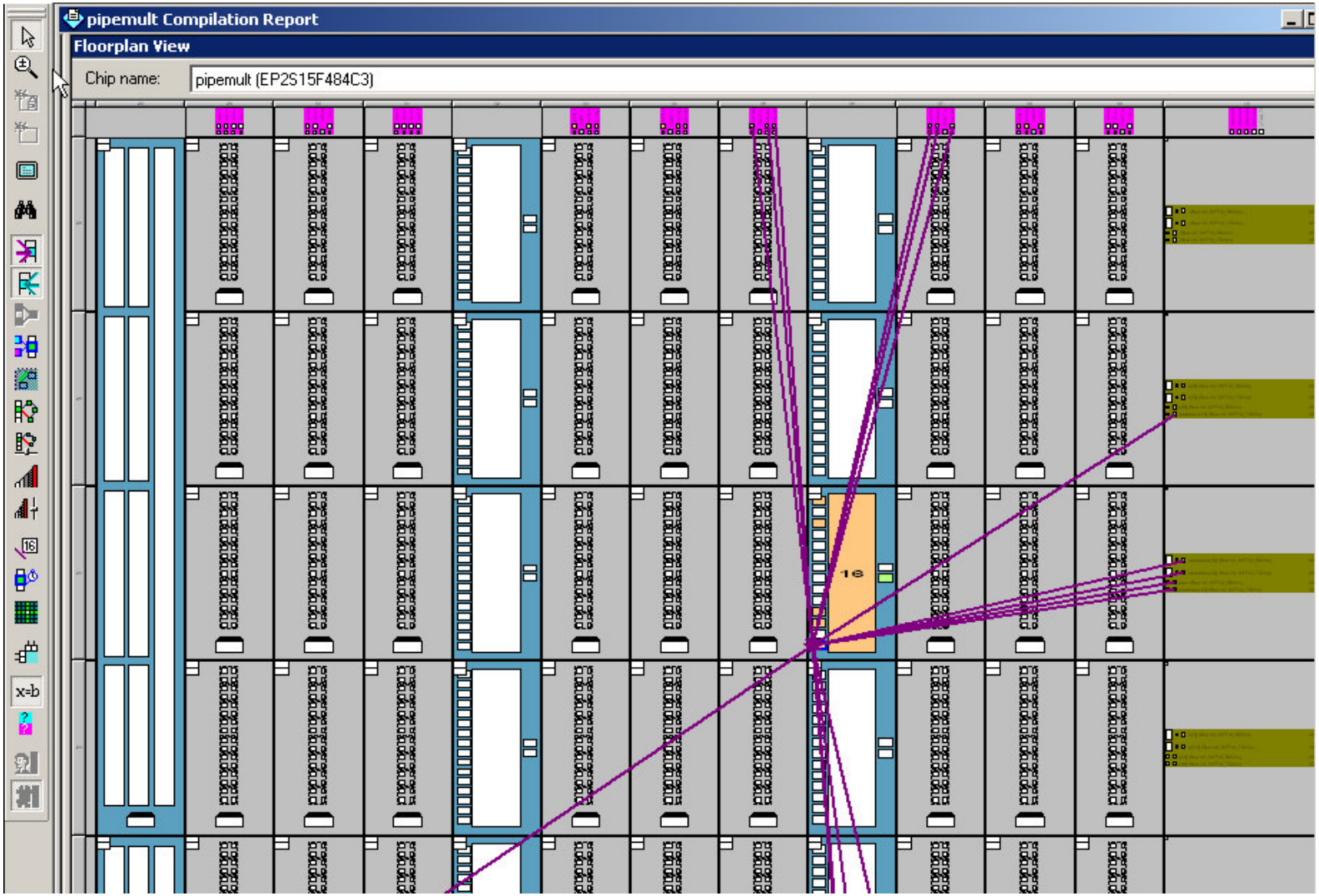
	Name	Pin #	I/O Bank	X coordinate	Y coordinate	Cell number	Combinational Fan-Out	Registered Fan
1	clk1	X0_Y10_N1	1	0	10	1	1	0
2	dataa[0]	X40_Y8_N2	6	40	8	2	8	0
3	dataa[1]	X40_Y8_N0	6	40	8	0	8	0
4	dataa[2]	X40_Y9_N2	6	40	9	2	8	0
5	dataa[3]	X40_Y9_N1	6	40	9	1	8	0
6	dataa[4]	X40_Y7_N3	6	40	7	3	8	0
7	dataa[5]	X40_Y10_N0	6	40	10	0	8	0
8	dataa[6]	X40_Y10_N3	6	40	10	3	8	0
9	dataa[7]	X40_Y7_N0	6	40	7	0	8	0
10	datab[0]	X40_Y10_N2	6	40	10	2	8	0
11	datab[1]	X40_Y10_N1	6	40	10	1	8	0
12	datab[2]	X40_Y9_N3	6	40	9	3	8	0
13	datab[3]	X40_Y9_N0	6	40	9	0	8	0
14	datab[4]	X40_Y7_N1	6	40	7	1	8	0
15	datab[5]	X40_Y7_N2	6	40	7	2	8	0
16	datab[6]	X40_Y8_N1	6	40	8	1	8	0
17	datab[7]	X40_Y8_N3	6	40	8	3	8	0
18	rdaddress[0]	X40_Y24_N1	5	40	24	1	1	0
19	rdaddress[1]	X35_Y27_N2	4	35	27	2	1	0
20	rdaddress[2]	X40_Y24_N0	5	40	24	0	1	0
21	rdaddress[3]	X40_Y25_N2	5	40	25	2	1	0
22	rdaddress[4]	X37_Y27_N3	4	37	27	3	1	0
23	wraddress[0]	X40_Y24_N2	5	40	24	2	1	0
24	wraddress[1]	X37_Y27_N2	4	37	27	2	1	0
25	wraddress[2]	X37_Y27_N0	4	37	27	0	1	0
26	wraddress[3]	X35_Y27_N1	4	35	27	1	1	0
27	wraddress[4]	X35_Y27_N0	4	35	27	0	1	0
28	wren	X40_Y24_N3	5	40	24	3	1	0

Several Sections Detail the Resource Usage

Floorplan

- Available Views
  - Last Compilation
  - Chip Editor
  - Timing Closure
- Uses
  - Make Assignments
  - View Placement
  - View Connections

Floorplan View



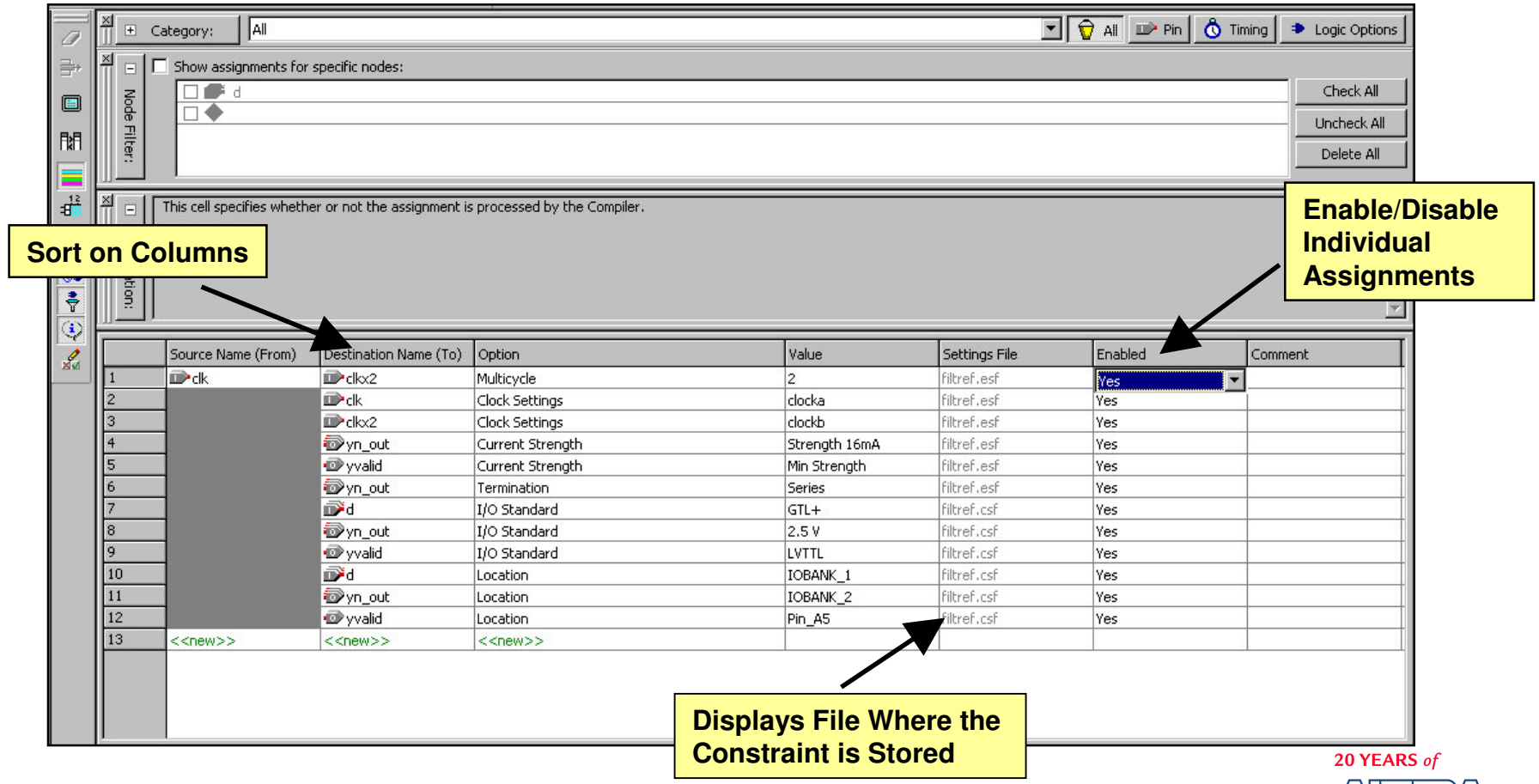
Assignments

- Assignment Editor
- Example Assignments
- I/O Assignments & Analysis

- Perform Analysis & Elaboration before Obtaining Hierarchy & Node Information

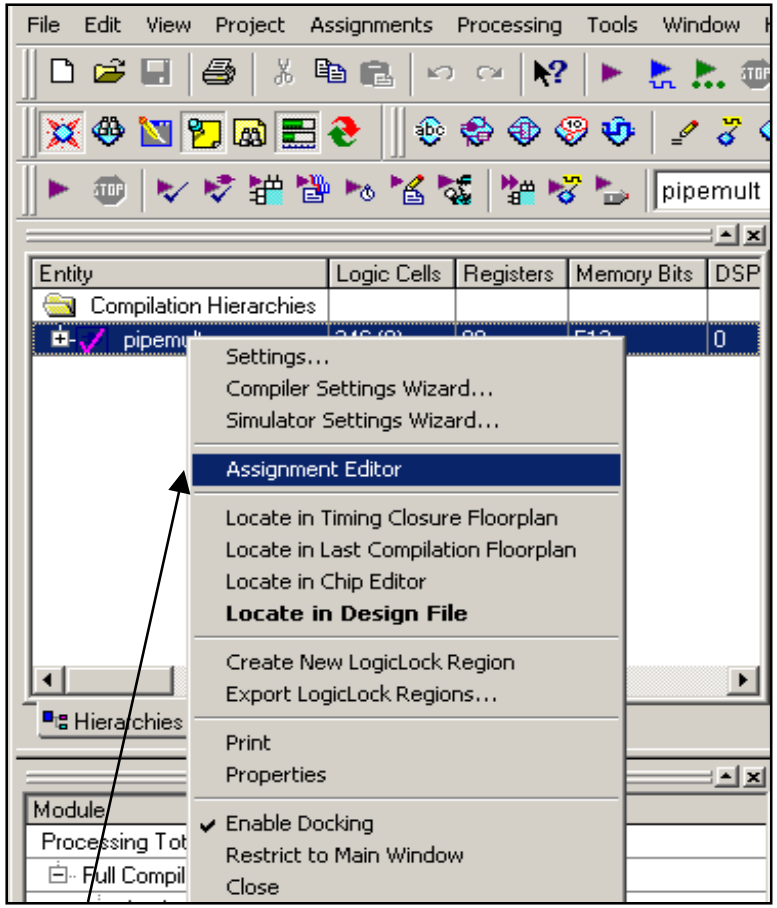
Assignment Editor (AE)

- Provides Spreadsheet Assignment Entry & Display
  - Can Copy & Paste from Clipboard

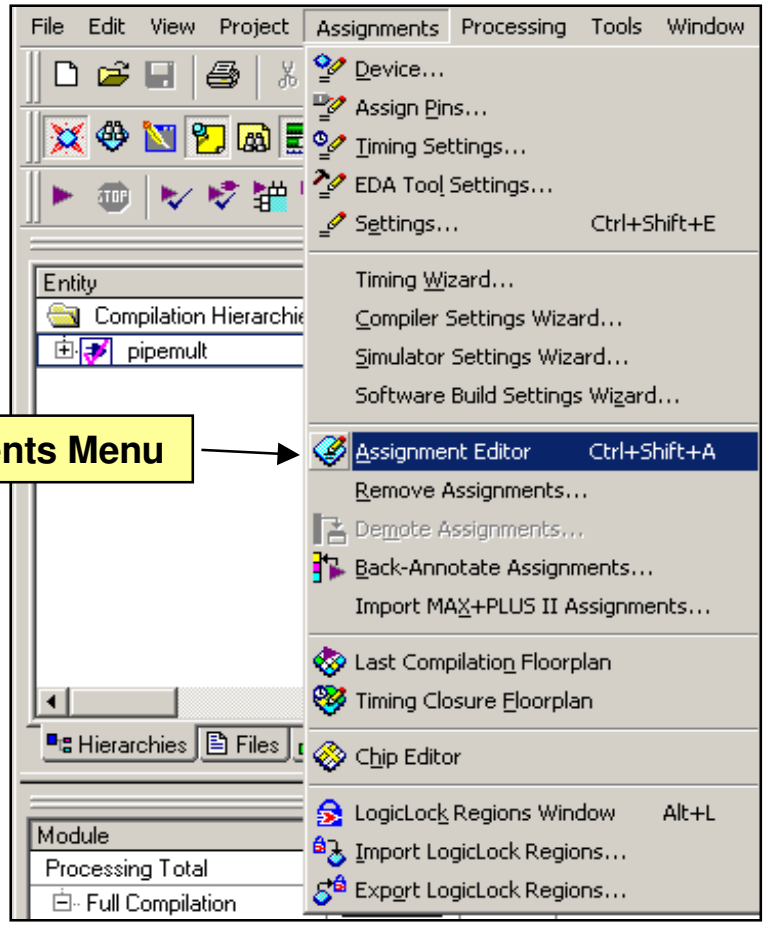


	Source Name (From)	Destination Name (To)	Option	Value	Settings File	Enabled	Comment
1	clk	clkx2	Multicycle	2	filtref.esf	Yes	
2		clk	Clock Settings	clocka	filtref.esf	Yes	
3		clkx2	Clock Settings	clockb	filtref.esf	Yes	
4		yn_out	Current Strength	Strength 16mA	filtref.esf	Yes	
5		yvalid	Current Strength	Min Strength	filtref.esf	Yes	
6		yn_out	Termination	Series	filtref.esf	Yes	
7		d	I/O Standard	GTL+	filtref.csf	Yes	
8		yn_out	I/O Standard	2.5 V	filtref.csf	Yes	
9		yvalid	I/O Standard	LVTTL	filtref.csf	Yes	
10		d	Location	IOBANK_1	filtref.csf	Yes	
11		yn_out	Location	IOBANK_2	filtref.csf	Yes	
12		yvalid	Location	Pin_A5	filtref.csf	Yes	
13	<<new>>	<<new>>	<<new>>				

Opening Assignment Editor



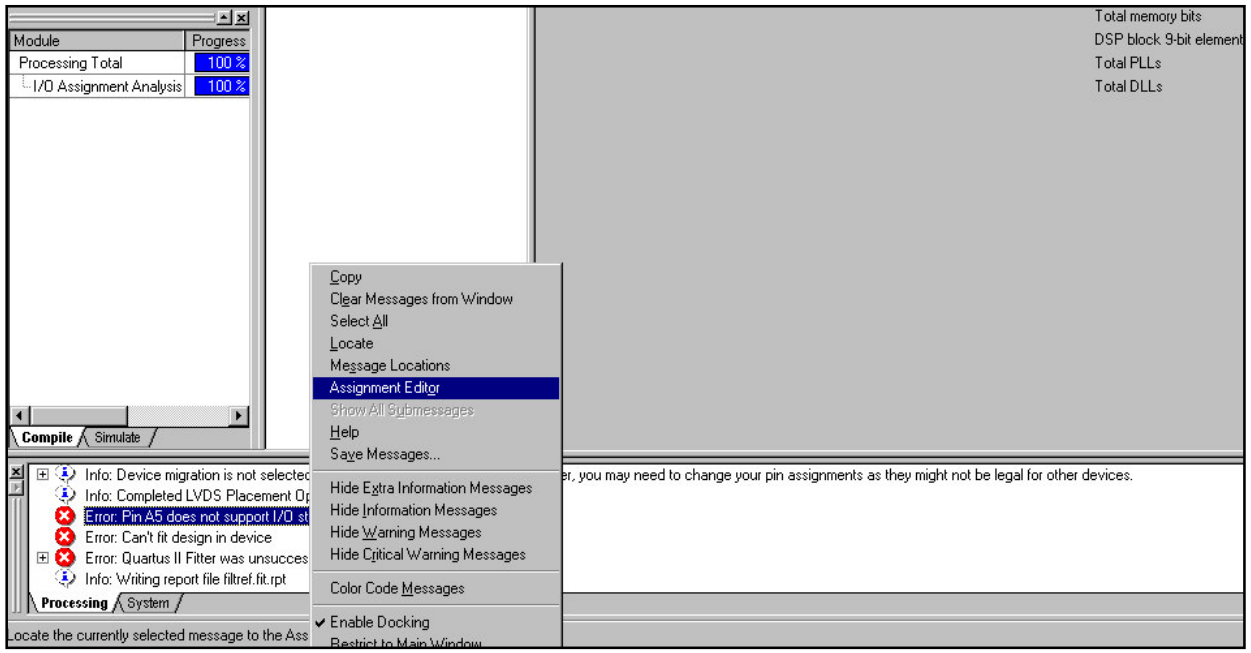
Assignments Menu



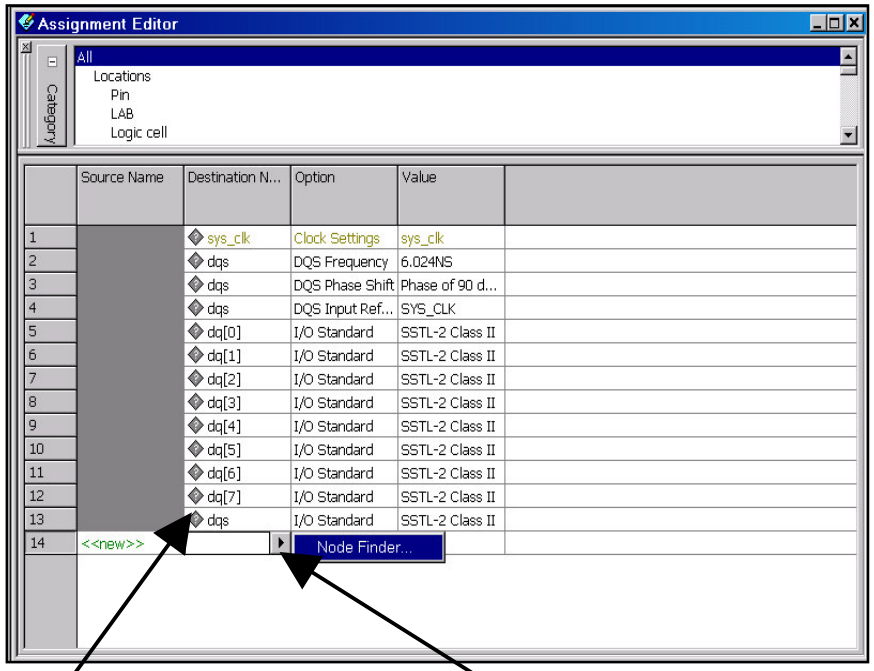
Invoke the Assignment Editor by Highlighting an Entity in the Hierarchy View & Right- Clicking

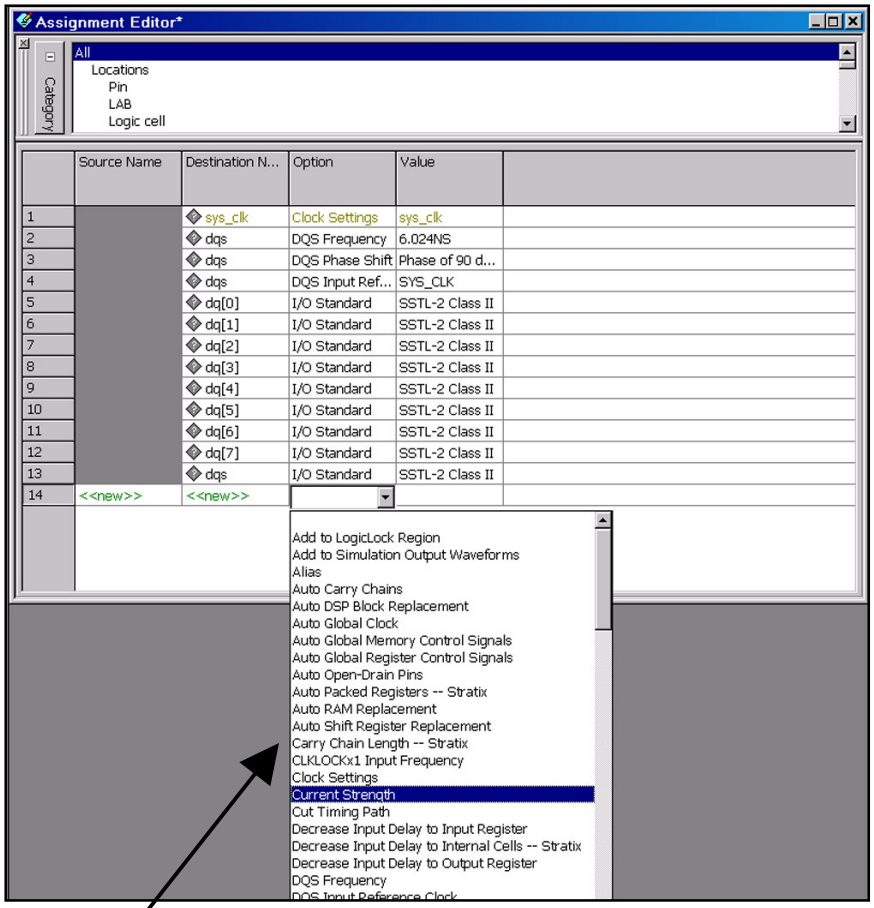
Opening Assignment Editor (cont.)

- Locate to Assignment Editor from Timing Report, Messages, Etc



Using Assignment Editor





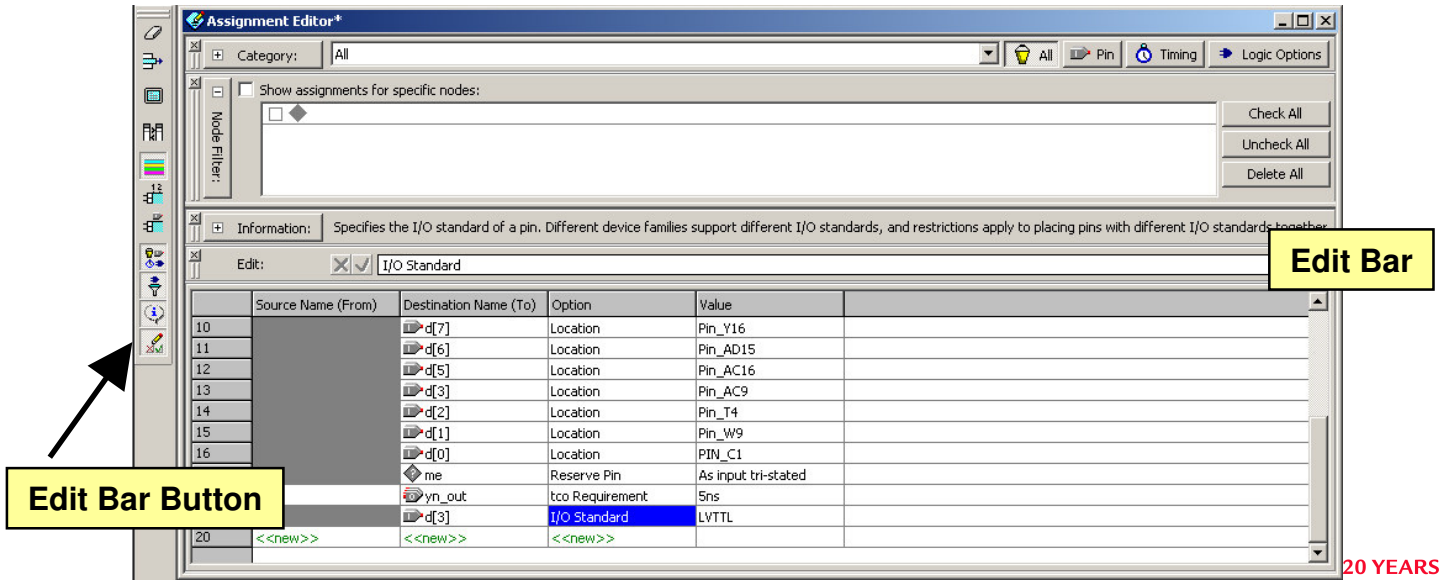
Double-Click Cells to Edit or Add New Assignment

Launches Node Finder

Selecting an Option Determines which Fields Are Available for Editing

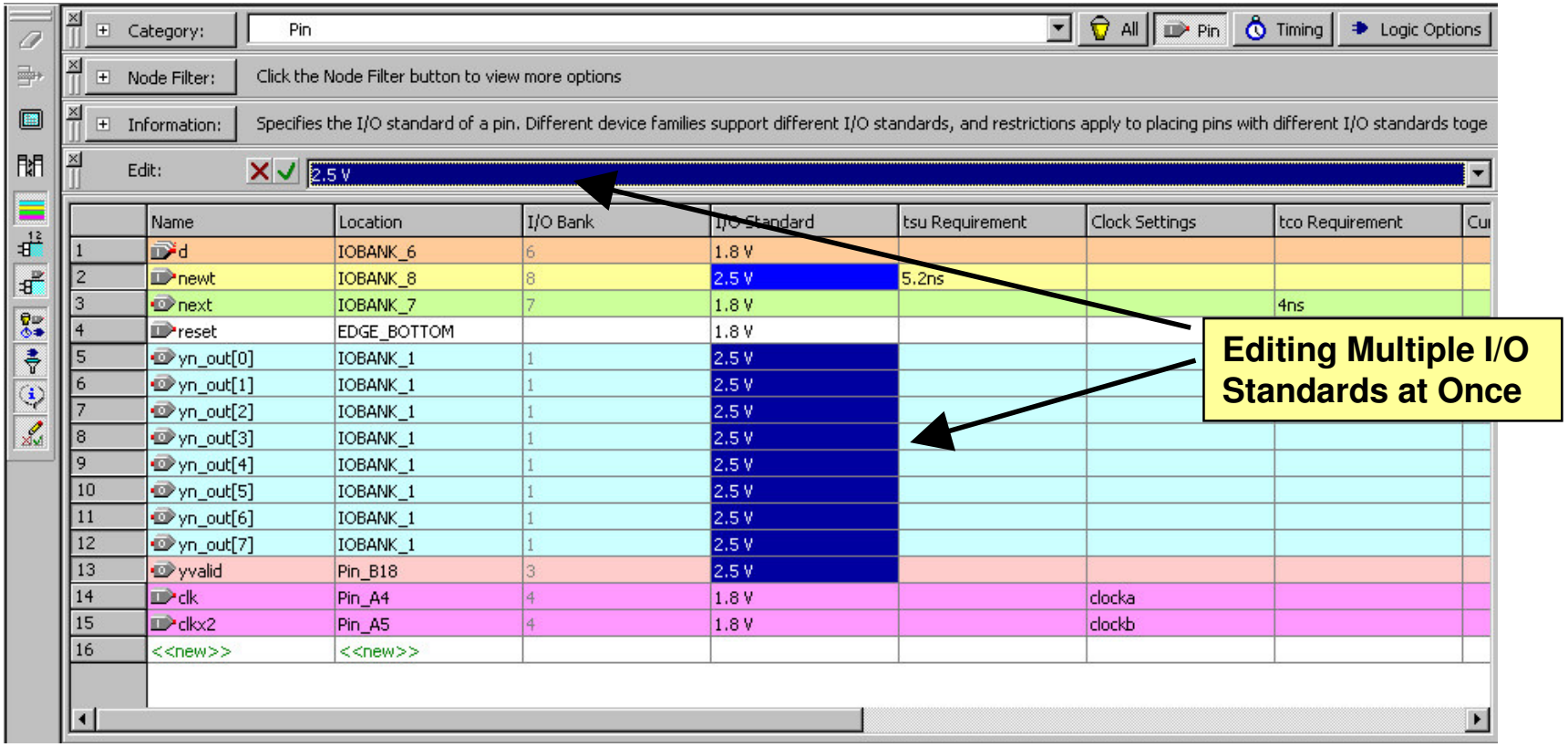
Entering Assignments

- Other Ways to Edit Cells
  - Single-Click & Type Name Directly
    - Useful for Known Node Names
  - Edit Menu
  - Edit Bar 



Editing Multiple Assignments

- Edit Multiple Constraints Simultaneously
  - Select Multiple Pins & Use Edit Bar (Click Check Mark to Accept)



	Name	Location	I/O Bank	I/O Standard	tsu Requirement	Clock Settings	tco Requirement	Cu
1	d	IOBANK_6	6	1.8 V				
2	newt	IOBANK_8	8	2.5 V	5.2ns			
3	next	IOBANK_7	7	1.8 V			4ns	
4	reset	EDGE_BOTTOM		1.8 V				
5	yn_out[0]	IOBANK_1	1	2.5 V				
6	yn_out[1]	IOBANK_1	1	2.5 V				
7	yn_out[2]	IOBANK_1	1	2.5 V				
8	yn_out[3]	IOBANK_1	1	2.5 V				
9	yn_out[4]	IOBANK_1	1	2.5 V				
10	yn_out[5]	IOBANK_1	1	2.5 V				
11	yn_out[6]	IOBANK_1	1	2.5 V				
12	yn_out[7]	IOBANK_1	1	2.5 V				
13	yvalid	Pin_B18	3	2.5 V				
14	clk	Pin_A4	4	1.8 V		clocka		
15	clkx2	Pin_A5	4	1.8 V		clockb		
16	<<new>>	<<new>>						

Node Finder

Search by Name Using Wildcards (? or *)

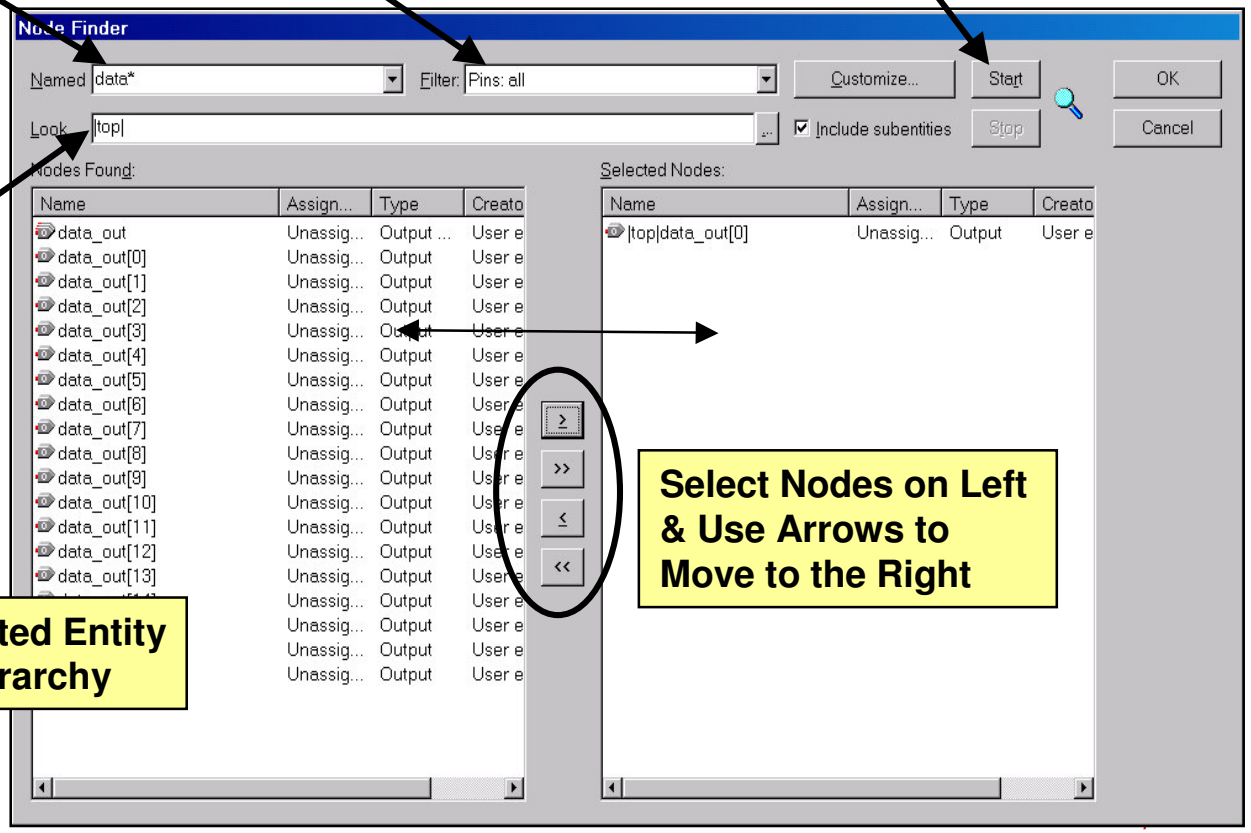
Use Filter to Select the Nodes to Be Displayed

Start Displays Nodes Meeting Search Criteria

Locate Nodes in a Certain Level of Hierarchy

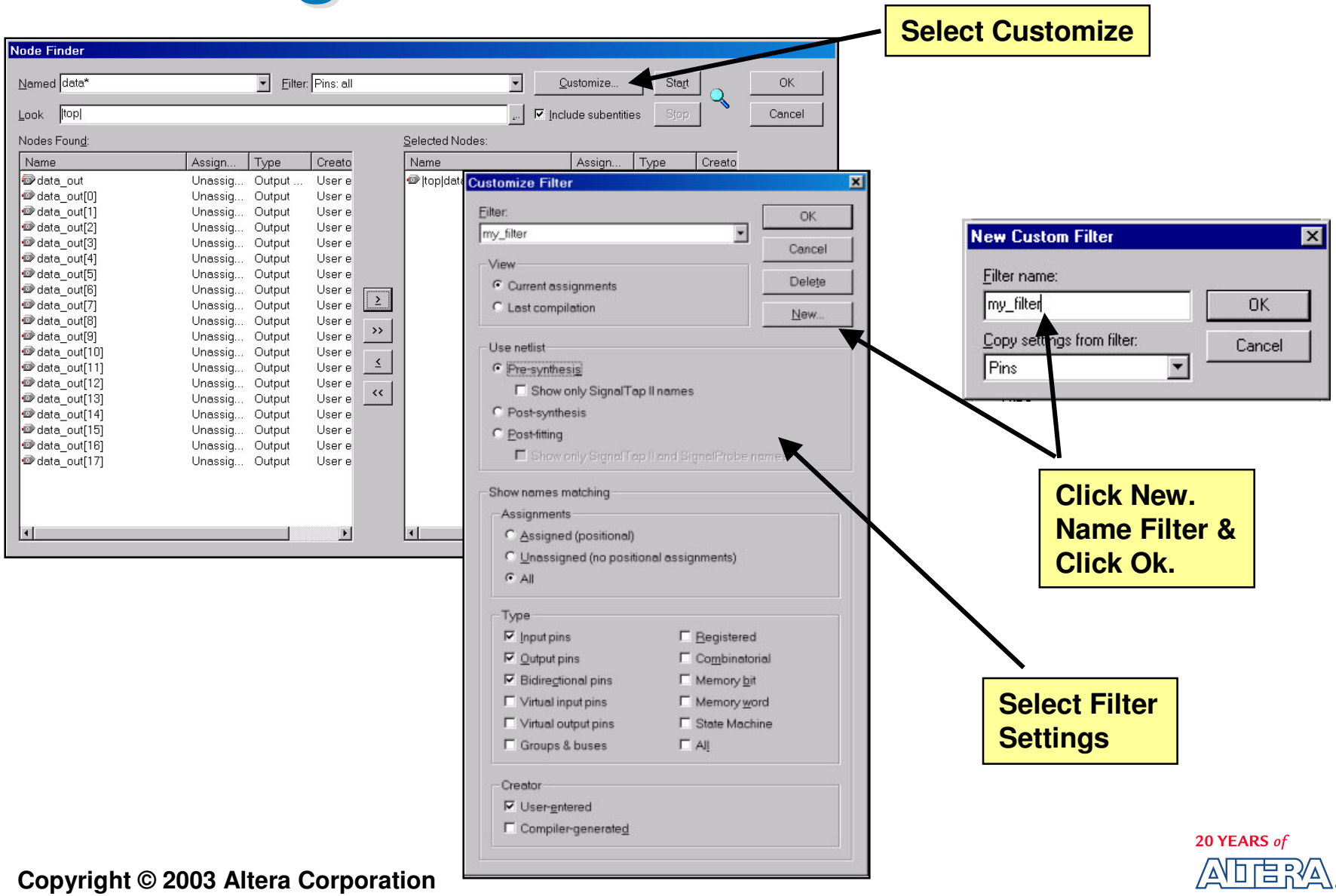
List of Nodes in Selected Entity & Lower Levels of Hierarchy

Select Nodes on Left & Use Arrows to Move to the Right



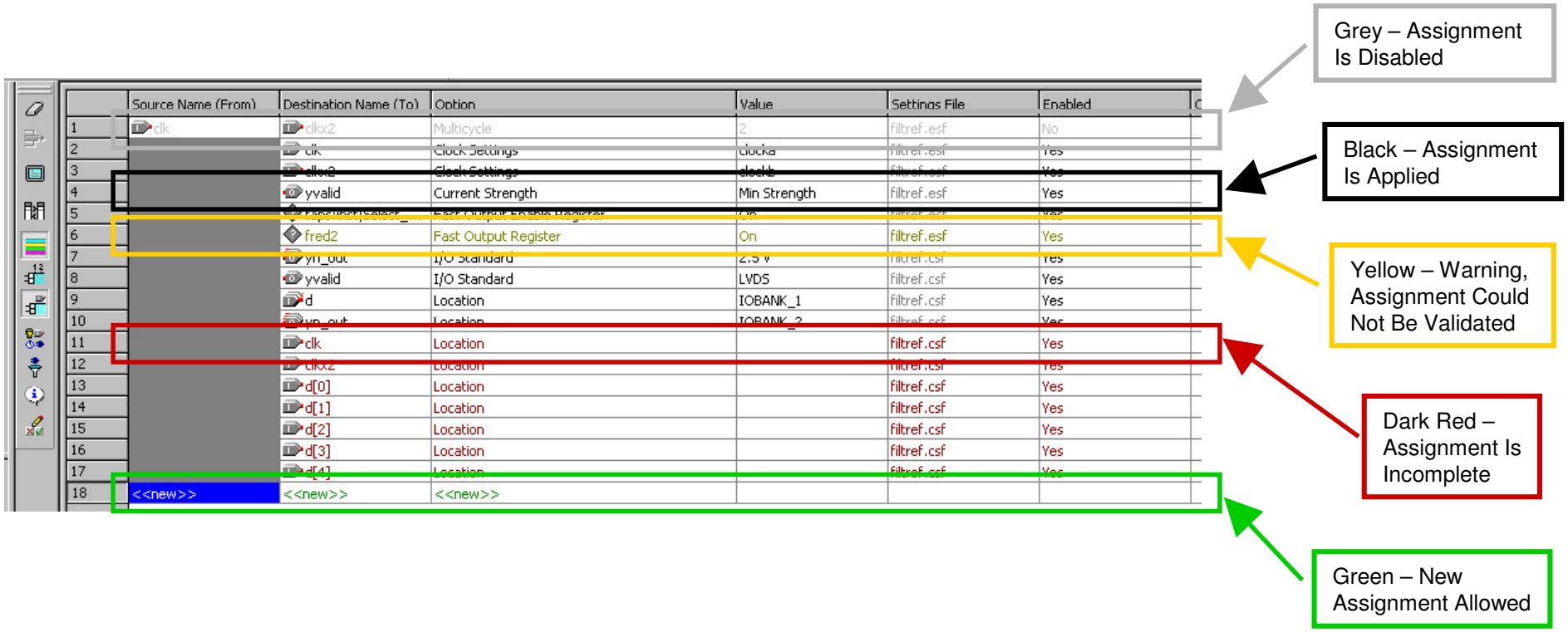
ALTERA INNOVATION

Creating a Custom Filter



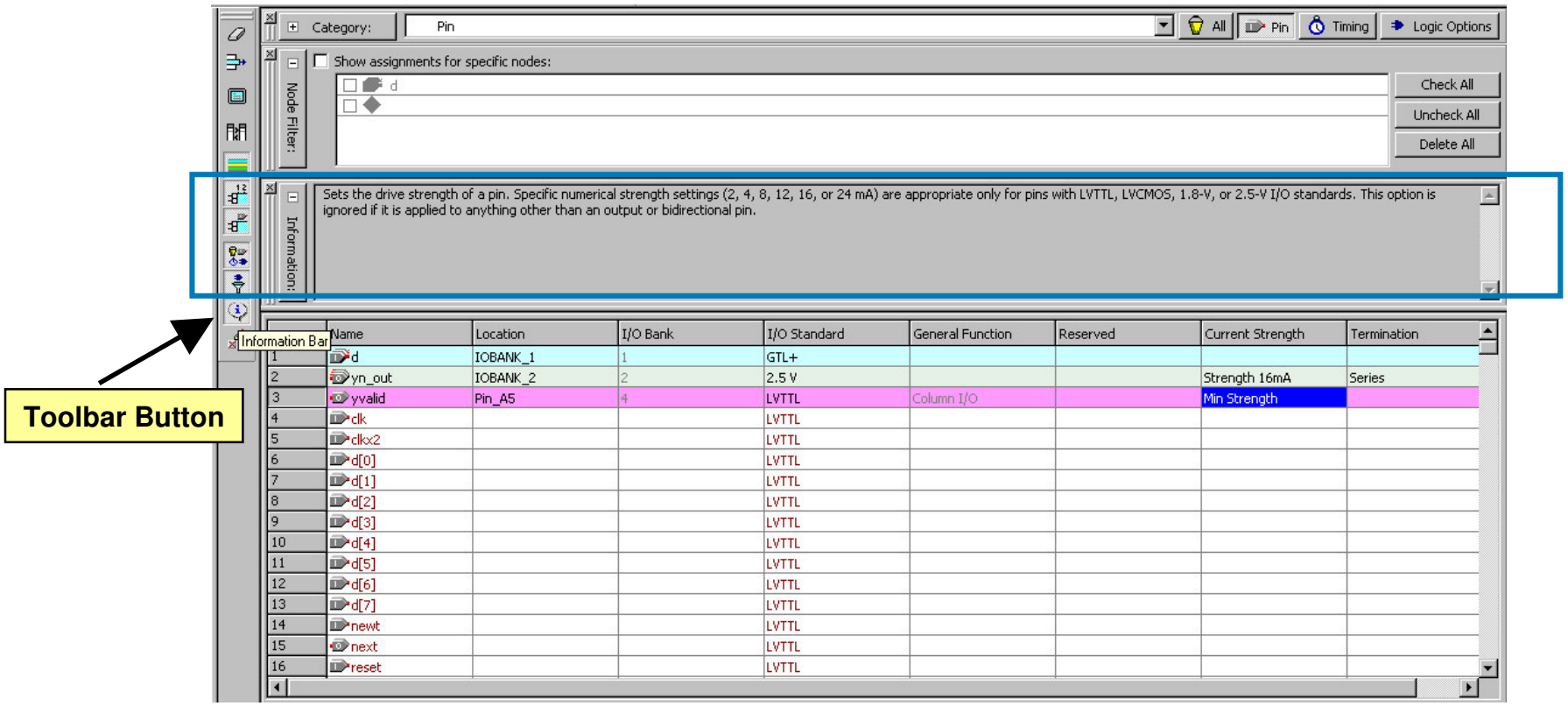
AE Dynamic Checking

- Validity of Constraint Checked during Entry
- Color-Coded to Display Status

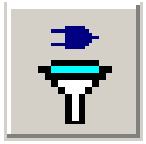


AE Information Bar

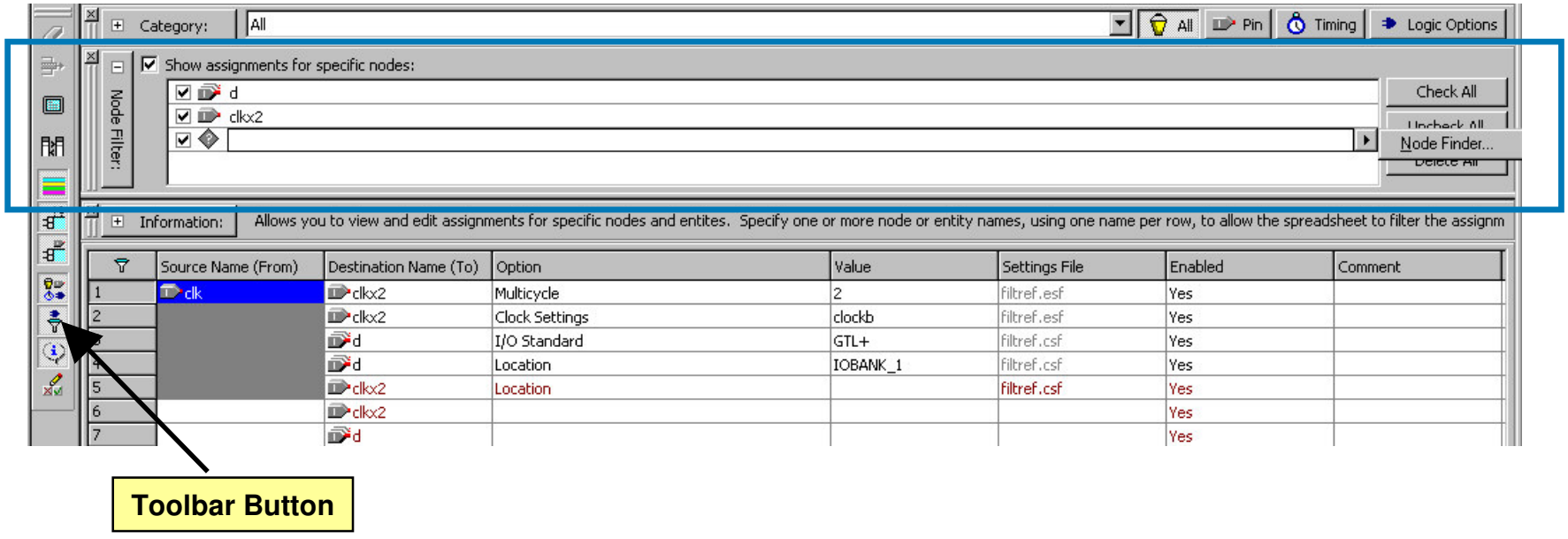
- Displays Information on Selected Cell
- Can Be Hidden or Minimized



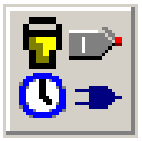
AE Node Filter Bar



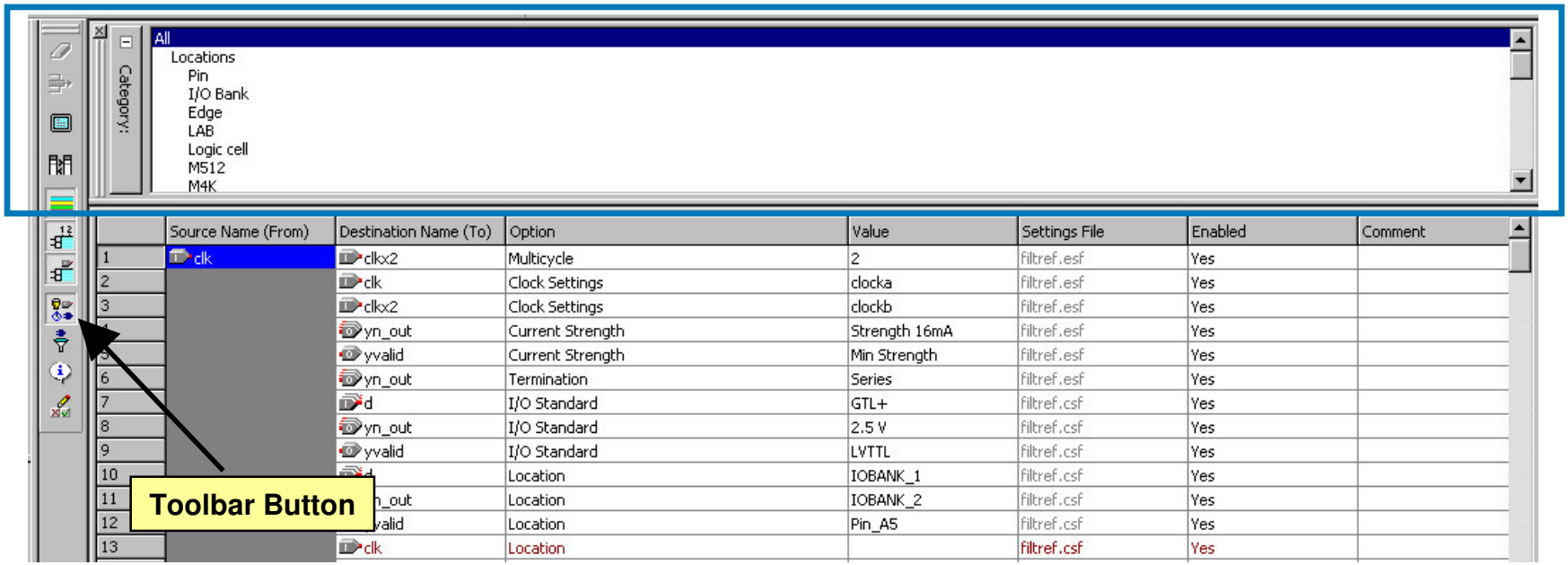
- Enables/Disables Selective Filtering of Constraints Displayed Based on Node Name
- Can Be Hidden or Minimized



AE Category Bar

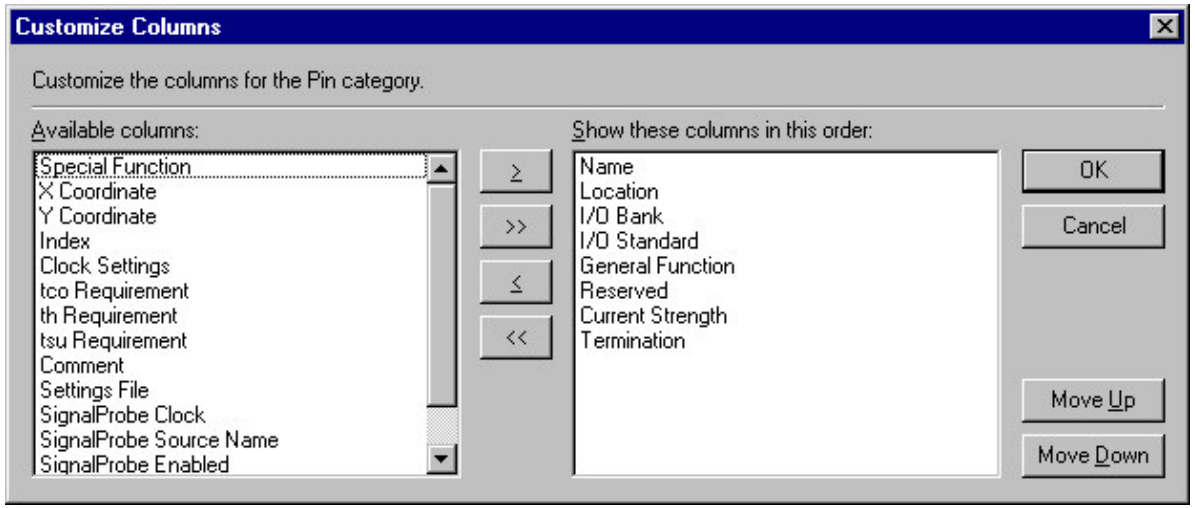


- Selects Category of Assignments to View
  - Ex. Pin Assignments, Timing Assignments
- Can Be Hidden or Minimized



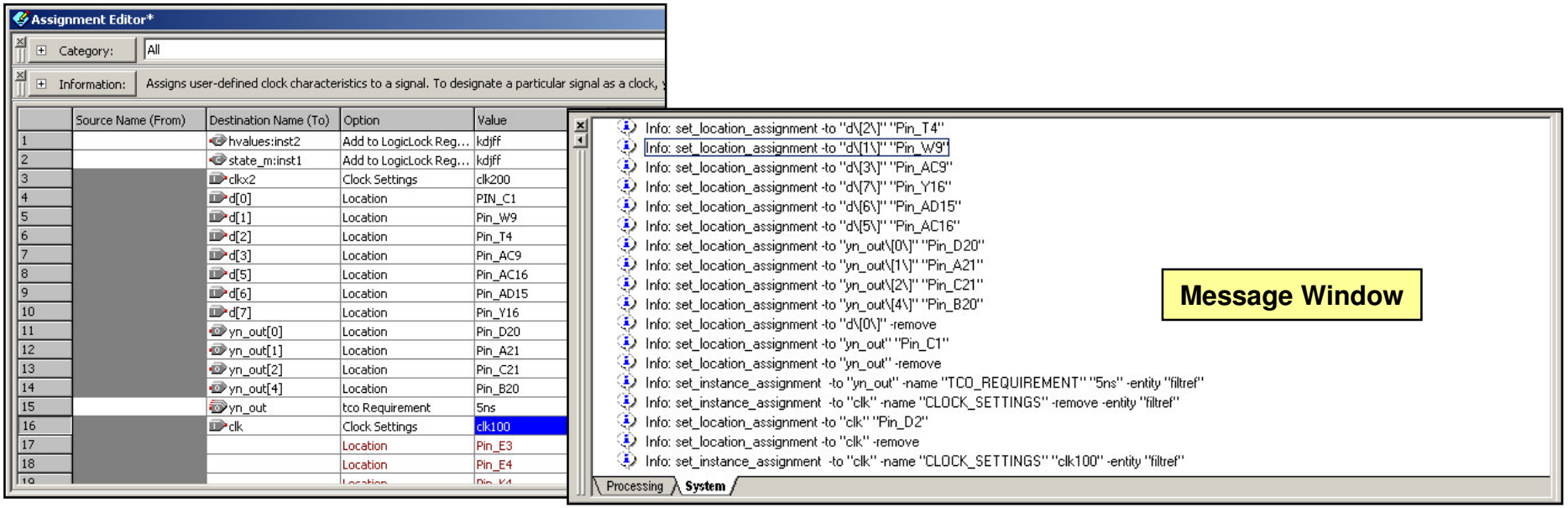
AE Customizable Columns

- Each Category Has a Set of Customizable Columns
  - Ex. Include Common Timing Assignments in Pin Category
- Comment Column for Each Assignment



AE Tcl Commands

- Equivalent Tcl Commands Are Displayed as Assignments Are Entered
  - Manually Copy to Create Tcl Scripts
  - Export Command (File Menu) Writes All Assignments to a Tcl File



Example Assignments

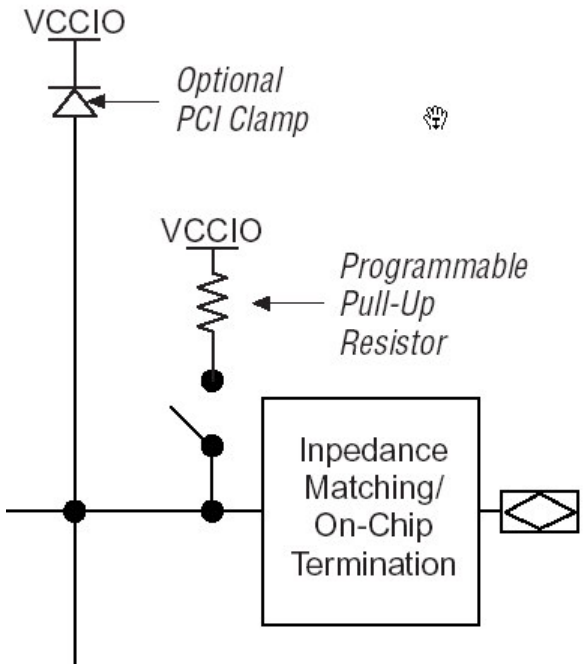
- Optimization Technique
- PCI I/O
- Output Pin Load

OPTIMIZATION TECHNIQUE

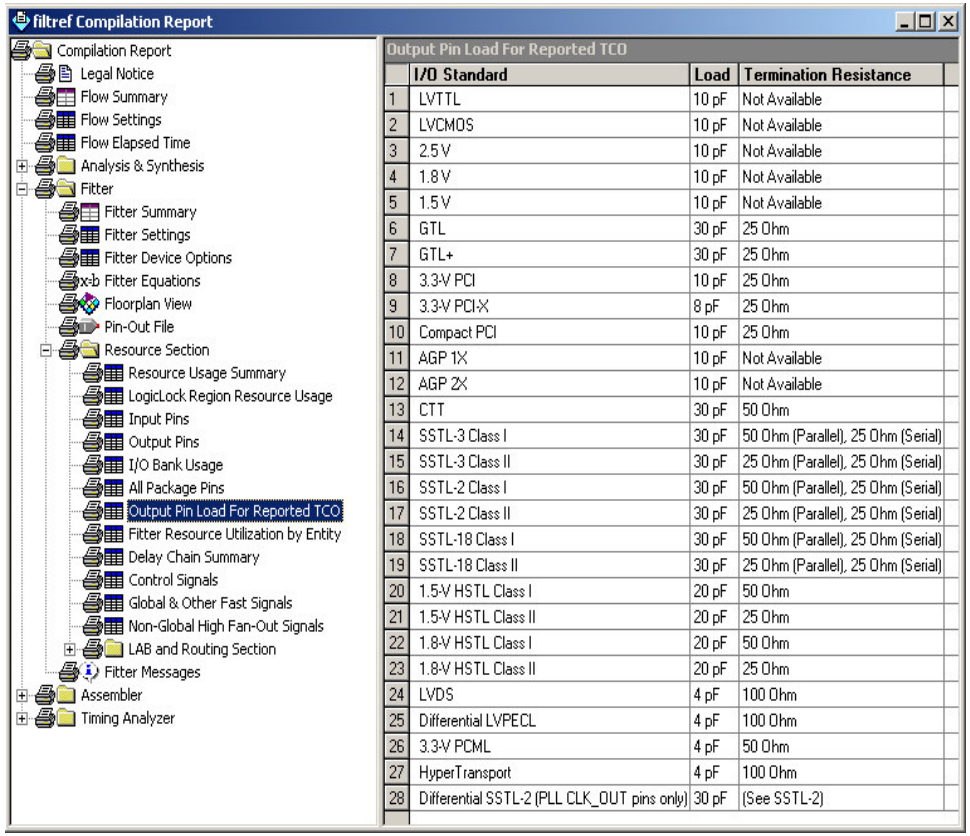
- Selects Synthesis Optimization Goal
  - Speed (Default)
  - Area
- Applies Only to Hierarchical Entities
- Effects Synthesis & Logic Mapping
- Only Applies to Quartus II Integrated Synthesis

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



Output Pin Load



Output Pin Load For Reported TCO

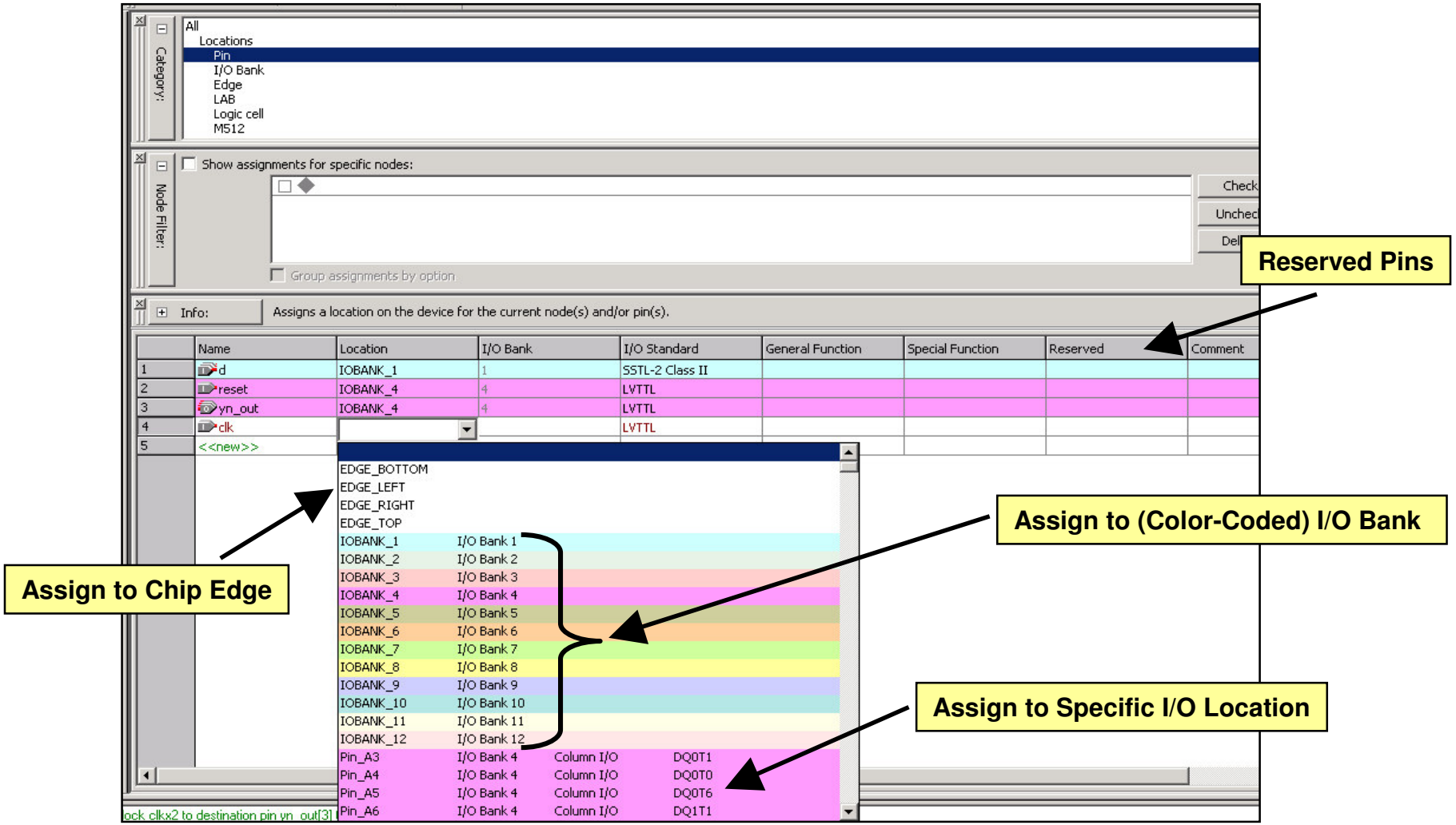
	I/O Standard	Load	Termination Resistance
1	LVTTL	10 pF	Not Available
2	LVCMOS	10 pF	Not Available
3	2.5 V	10 pF	Not Available
4	1.8 V	10 pF	Not Available
5	1.5 V	10 pF	Not Available
6	GTL	30 pF	25 Ohm
7	GTL+	30 pF	25 Ohm
8	3.3-V PCI	10 pF	25 Ohm
9	3.3-V PCI-X	8 pF	25 Ohm
10	Compact PCI	10 pF	25 Ohm
11	AGP 1X	10 pF	Not Available
12	AGP 2X	10 pF	Not Available
13	CTT	30 pF	50 Ohm
14	SSTL-3 Class I	30 pF	50 Ohm (Parallel), 25 Ohm (Serial)
15	SSTL-3 Class II	30 pF	25 Ohm (Parallel), 25 Ohm (Serial)
16	SSTL-2 Class I	30 pF	50 Ohm (Parallel), 25 Ohm (Serial)
17	SSTL-2 Class II	30 pF	25 Ohm (Parallel), 25 Ohm (Serial)
18	SSTL-18 Class I	30 pF	50 Ohm (Parallel), 25 Ohm (Serial)
19	SSTL-18 Class II	30 pF	25 Ohm (Parallel), 25 Ohm (Serial)
20	1.5-V HSTL Class I	20 pF	50 Ohm
21	1.5-V HSTL Class II	20 pF	25 Ohm
22	1.8-V HSTL Class I	20 pF	50 Ohm
23	1.8-V HSTL Class II	20 pF	25 Ohm
24	LVDS	4 pF	100 Ohm
25	Differential LVPECL	4 pF	100 Ohm
26	3.3-V PCML	4 pF	50 Ohm
27	HyperTransport	4 pF	100 Ohm
28	Differential SSTL-2 (PLL CLK_OUT pins only)	30 pF	(See SSTL-2)

- 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
- Must Be Applied to Output or Bidirectional Pins

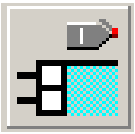
I/O (Pin) Assignments

- Make I/O Assignments Quickly
  - Assignment Editor
  - Settings Dialog Box (Device)
  - Scripting
  - Floorplan
- Specific Pin Location Unnecessary
  - IO Bank & Chip Edge Location Assignments
  - Reserved Pins with or without Locations

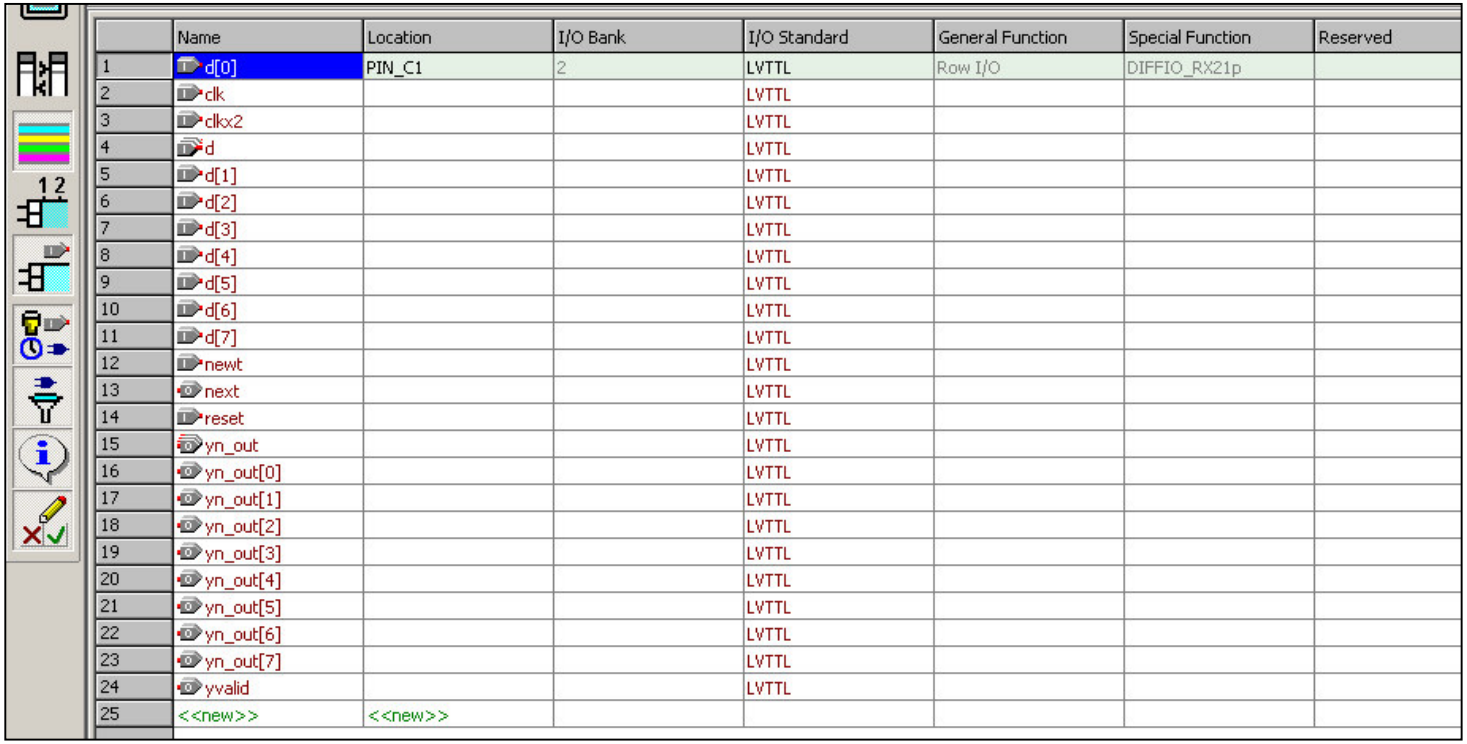
Assignment Editor I/O Assignments



AE Show All Known Pin Names

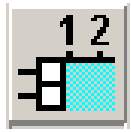


- Populate Spreadsheet with List of All Pin Names in Design
  - Assigned & Not Assigned

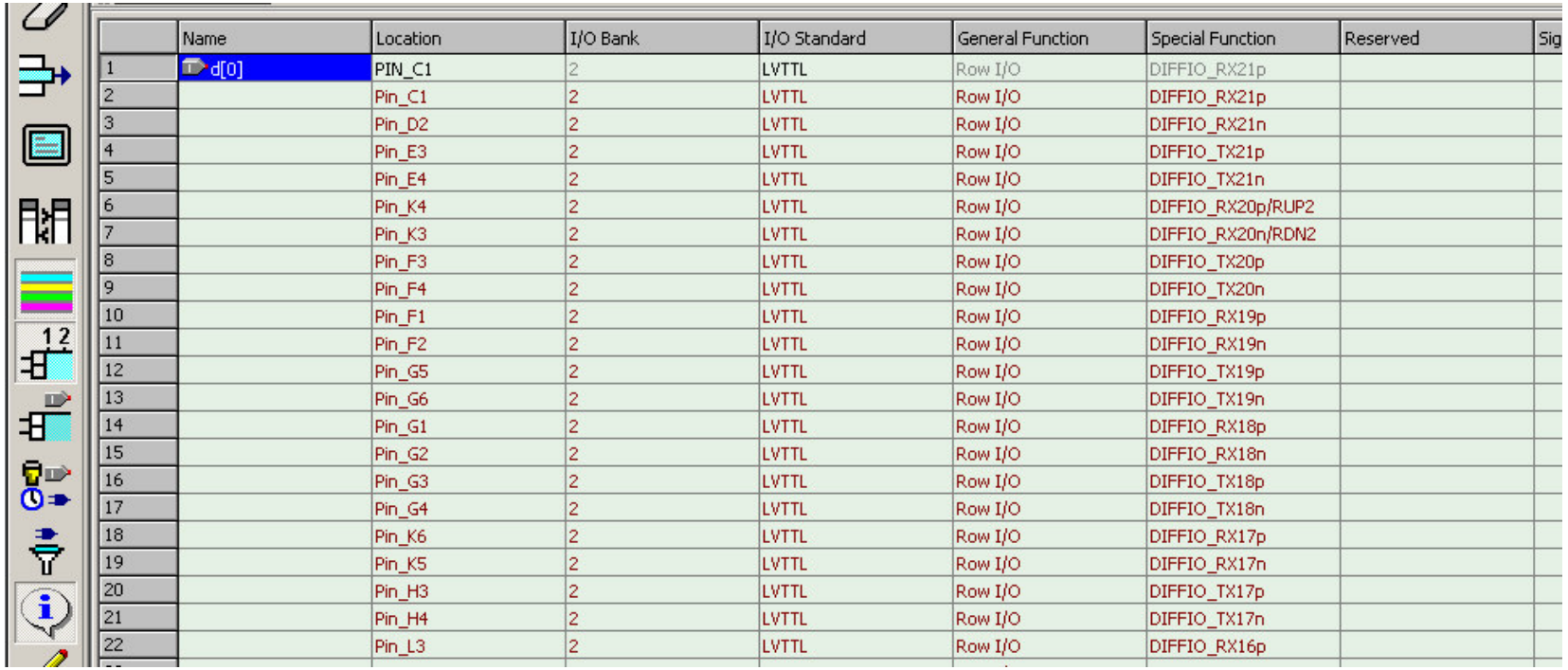


	Name	Location	I/O Bank	I/O Standard	General Function	Special Function	Reserved
1	d[0]	PIN_C1	2	LVTTL	Row I/O	DIFFIO_RX21p	
2	clk			LVTTL			
3	clkx2			LVTTL			
4	d			LVTTL			
5	d[1]			LVTTL			
6	d[2]			LVTTL			
7	d[3]			LVTTL			
8	d[4]			LVTTL			
9	d[5]			LVTTL			
10	d[6]			LVTTL			
11	d[7]			LVTTL			
12	newt			LVTTL			
13	next			LVTTL			
14	reset			LVTTL			
15	yn_out			LVTTL			
16	yn_out[0]			LVTTL			
17	yn_out[1]			LVTTL			
18	yn_out[2]			LVTTL			
19	yn_out[3]			LVTTL			
20	yn_out[4]			LVTTL			
21	yn_out[5]			LVTTL			
22	yn_out[6]			LVTTL			
23	yn_out[7]			LVTTL			
24	yvalid			LVTTL			
25	<<new>>	<<new>>					

AE Assignable Pin Numbers

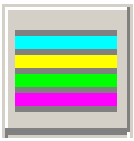


- Show All Assignable Pin Numbers
  - Populate Spreadsheet with All Pin Numbers Available for Assignment



	Name	Location	I/O Bank	I/O Standard	General Function	Special Function	Reserved	Sig
1	d[0]	PIN_C1	2	LVTTL	Row I/O	DIFFIO_RX21p		
2		Pin_C1	2	LVTTL	Row I/O	DIFFIO_RX21p		
3		Pin_D2	2	LVTTL	Row I/O	DIFFIO_RX21n		
4		Pin_E3	2	LVTTL	Row I/O	DIFFIO_TX21p		
5		Pin_E4	2	LVTTL	Row I/O	DIFFIO_TX21n		
6		Pin_K4	2	LVTTL	Row I/O	DIFFIO_RX20p/RUP2		
7		Pin_K3	2	LVTTL	Row I/O	DIFFIO_RX20n/RDN2		
8		Pin_F3	2	LVTTL	Row I/O	DIFFIO_TX20p		
9		Pin_F4	2	LVTTL	Row I/O	DIFFIO_TX20n		
10		Pin_F1	2	LVTTL	Row I/O	DIFFIO_RX19p		
11		Pin_F2	2	LVTTL	Row I/O	DIFFIO_RX19n		
12		Pin_G5	2	LVTTL	Row I/O	DIFFIO_TX19p		
13		Pin_G6	2	LVTTL	Row I/O	DIFFIO_TX19n		
14		Pin_G1	2	LVTTL	Row I/O	DIFFIO_RX18p		
15		Pin_G2	2	LVTTL	Row I/O	DIFFIO_RX18n		
16		Pin_G3	2	LVTTL	Row I/O	DIFFIO_TX18p		
17		Pin_G4	2	LVTTL	Row I/O	DIFFIO_TX18n		
18		Pin_K6	2	LVTTL	Row I/O	DIFFIO_RX17p		
19		Pin_K5	2	LVTTL	Row I/O	DIFFIO_RX17n		
20		Pin_H3	2	LVTTL	Row I/O	DIFFIO_TX17p		
21		Pin_H4	2	LVTTL	Row I/O	DIFFIO_TX17n		
22		Pin_L3	2	LVTTL	Row I/O	DIFFIO_RX16p		

AE Show I/O Banks in Color



- Enable/Disable I/O Color Coding of Spreadsheet Based upon Floorplan

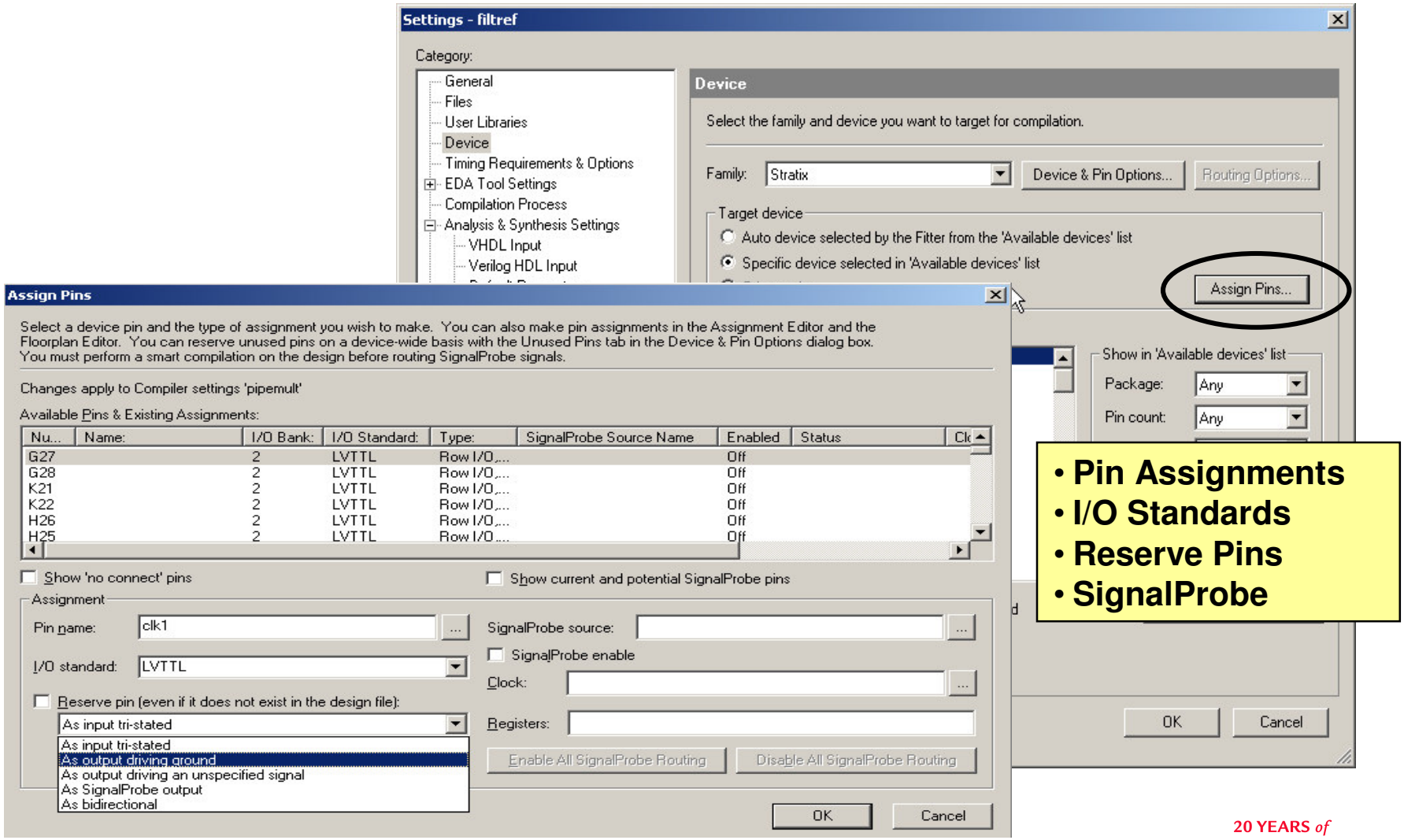
Assignment Editor*

Category: Pin | All | Pin | Timing | Logic Options

Information: This cell specifies the pin name to which you want to make an assignment.

	Name	Location	I/O Bank	I/O Standard	General Function	Special Function	Reserved
1	d[0]	PIN_C1	2	LVTTL	Row I/O	DIFFIO_RX21p	
2	d[1]	Pin_W9	8	LVTTL	Column I/O	DQ7B1	
3	d[2]	Pin_T4	1	LVTTL	Row I/O	DIFFIO_RX5n	
4	d[3]	Pin_AC9	8	LVTTL	Column I/O	DQ6B7	
5	d[5]	Pin_AC16	7	LVTTL	Column I/O	PGM1	
6	d[6]	Pin_AD15	7	LVTTL	Column I/O	RUnLU	
7	d[7]	Pin_Y16	7	LVTTL	Column I/O	nRS	
8	yn_out[0]	Pin_D20	4	LVTTL	Column I/O	DQ1T2	
9	yn_out[1]	Pin_A21	4	LVTTL	Column I/O	DQS1T	
10	yn_out[2]	Pin_C21	4	LVTTL	Column I/O	DQ1T3	
11	yn_out[4]	Pin_B20	4	LVTTL	Column I/O	DQ1T4	
12		Pin_C1	2	LVTTL	Row I/O	DIFFIO_RX21p	
13		Pin_D2	2	LVTTL	Row I/O	DIFFIO_RX21n	
14		Pin_E3	2	LVTTL	Row I/O	DIFFIO_TX21p	
15		Pin_E4	2	LVTTL	Row I/O	DIFFIO_TX21n	
16		Pin_K4	2	LVTTL	Row I/O	DIFFIO_RX20p/RUP2	
17		Pin_K3	2	LVTTL	Row I/O	DIFFIO_RX20n/RDN2	
18		Pin_F3	2	LVTTL	Row I/O	DIFFIO_TX20p	
19		Pin_F4	2	LVTTL	Row I/O	DIFFIO_TX20n	
20		Pin_F1	2	LVTTL	Row I/O	DIFFIO_RX19p	
21		Pin_F2	2	LVTTL	Row I/O	DIFFIO_RX19n	
22		Pin_G5	2	LVTTL	Row I/O	DIFFIO_TX19p	
23		Pin_G6	2	LVTTL	Row I/O	DIFFIO_TX19n	
24		Pin_G1	2	LVTTL	Row I/O	DIFFIO_RX18p	

Setting Dialog Box - Device

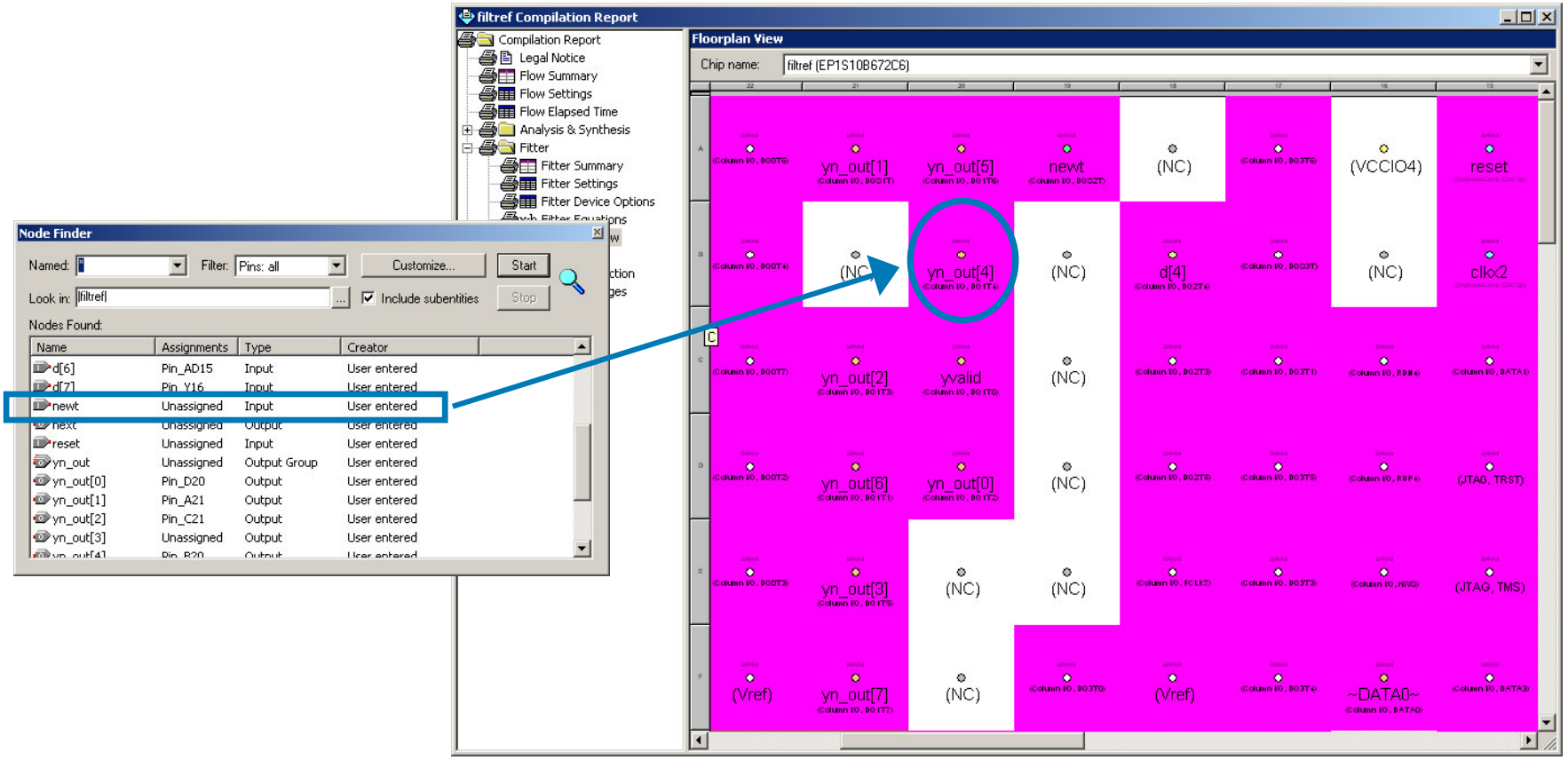


- **Pin Assignments**
- **I/O Standards**
- **Reserve Pins**
- **SignalProbe**

20 YEARS of ALTERA INNOVATION

Drag & Drop I/O Assignments

- Drag & Drop from Node Finder to Floorplan



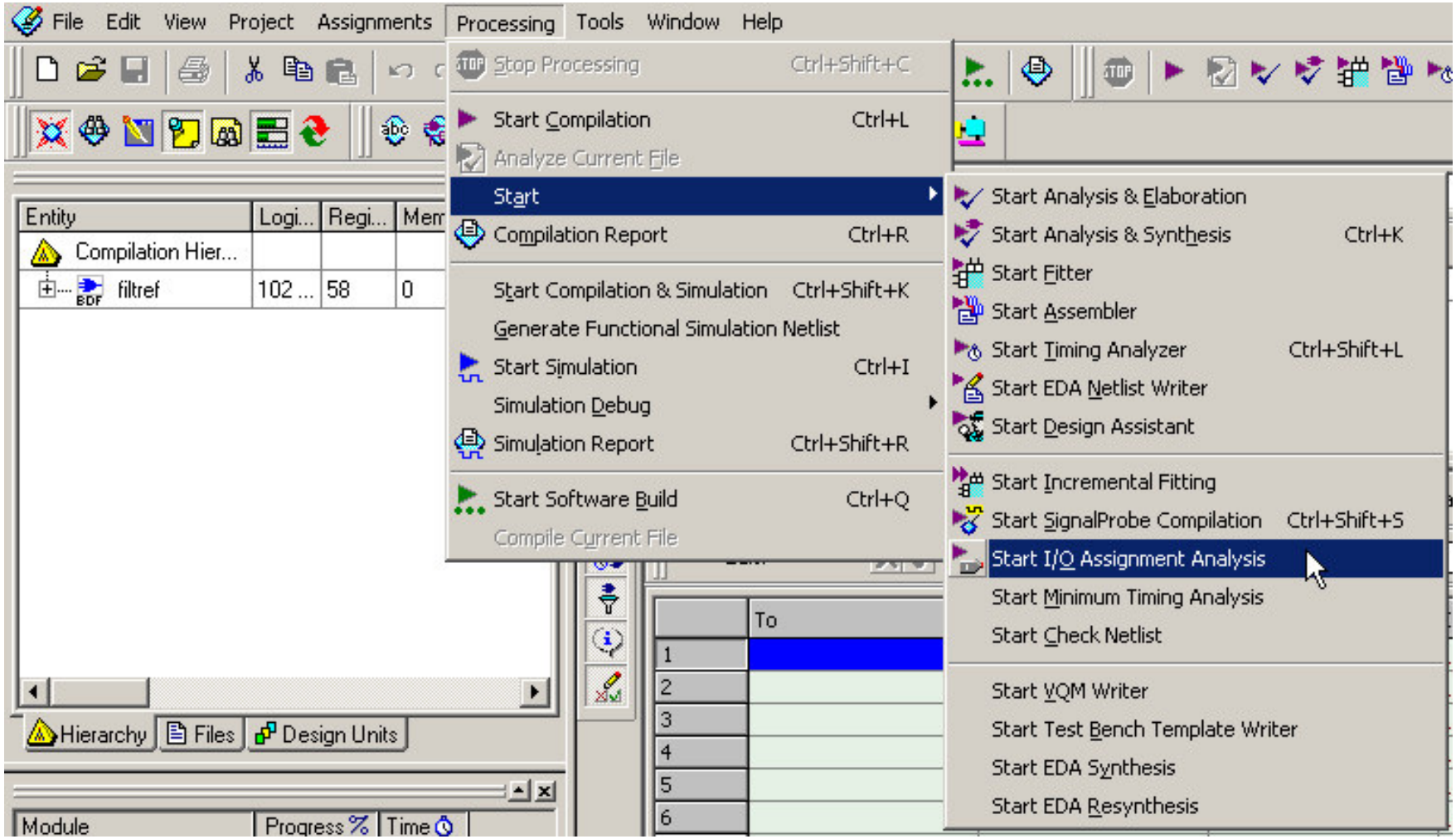
I/O Assignment Problems

- I/O Selected & Verified Too Far into Design Cycle
- Variety of Complex I/O Standards on Single Device
  - Not All Supported on Every I/O Block
- I/O Placement Limitations
  - Current Strength
  - Single-Ended vs. Differential
  - Single-Ended vs. V_{ref}

I/O Assignment Solution

- I/O Assignment Analysis Command
  - Quickly Checks Legality of I/O Assignments
    - Checks All Assignments
    - Does Not Stop at First Failure
  - Allows User to Identify & Correct Pin-Related Issues without Full Compilation
  - Has Two Usages
    - Performing Legality Checks Based on User Reserved Pin Assignments with Partial or No Design Files
    - Performing Legality Checks Based on User I/O Assignments with a Complete Design

I/O Assignment Analysis



I/O Analysis 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

Legality Checks

Rule (subset of the rules)	Incomplete Design	Complete Design
Differential Pin Spacing		●
# of Output/Bidirs Toggling with a VREF		●
Blocks Directly Feeding or Fed by I/O pins		●
Current Strength	●	●
DC Current Draw	●	●
Pin Location	●	●
I/O Bank Capacity	●	●
Voltage Conflict (Vref, VCCIO)	●	●
On-Chip Termination	●	●

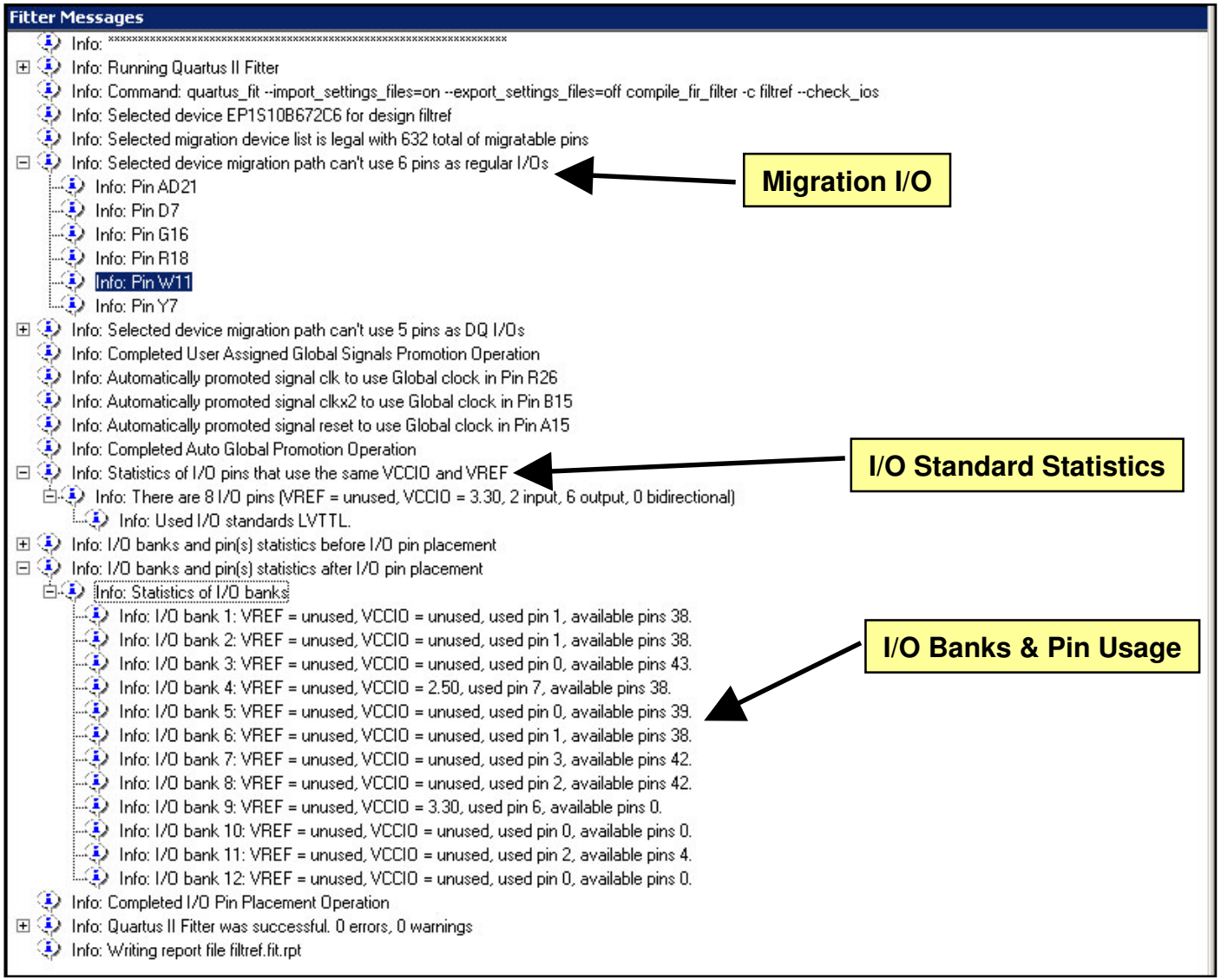
I/O Assignment Analysis Notes

- Uses Toggle Rate Assignment
  - Set Toggle_rate = 0 for Very Low Frequency Output Pins
  - Allows Single Ended Pins to be Placed Closer to Differential Pins
- Invokes Smart Recompile for I/O Features
  - Delay Chains
  - Drive Strengths
  - Bus Hold

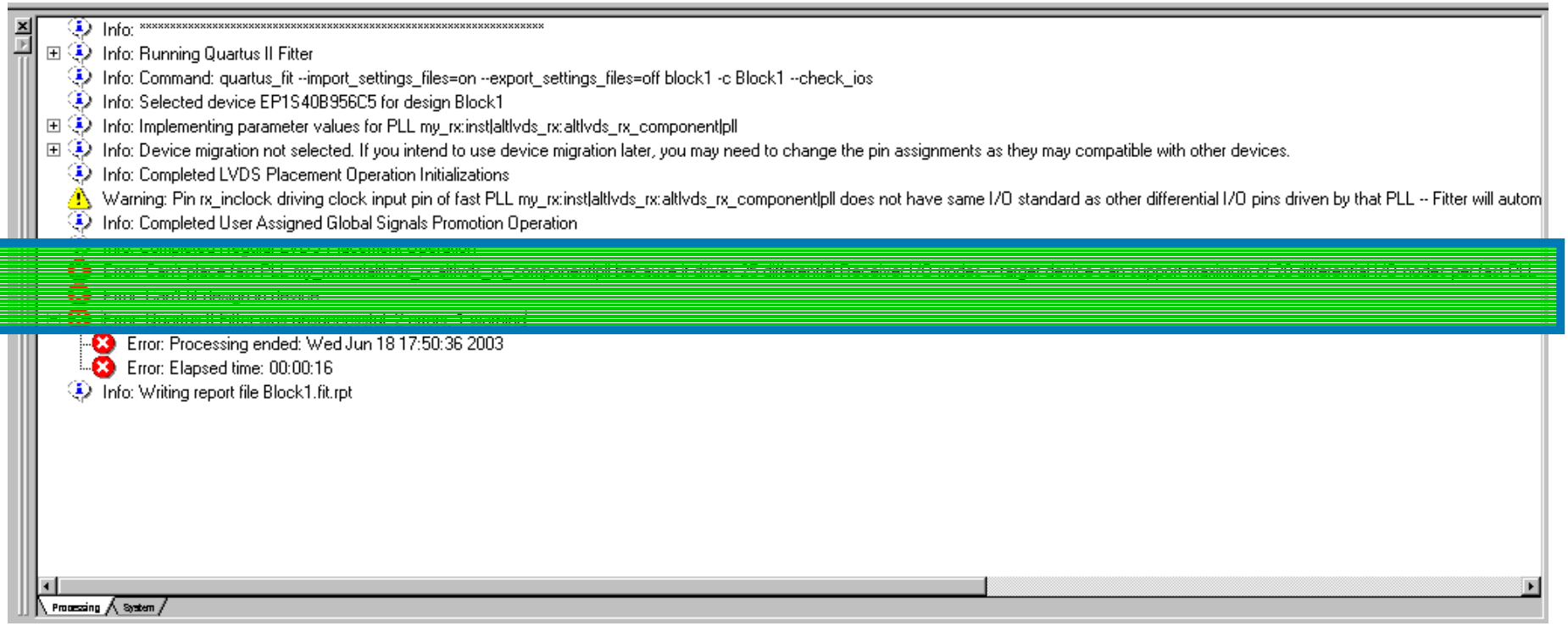
I/O Assignment Analysis Output

- Detailed Messages on
  - Incorrect I/O Assignments & Problem Description
  - Compiler Assumptions that User did Not Specify
  - Pins Changing Functionality with Pin-Migration
  - Used I/O Standards & Voltages per I/O Bank
- Compilation Report (Fitter Section)
  - I/O Pin Tables
    - Show User & Fitter Assigned I/O
  - Shows Partial Placement Results in Floorplan to Help Debug Fitting Errors
- Messages Locate to Assignment Editor & Floorplan to Explain Placement Failure

I/O Assignment Analysis Messages



Example Error Message

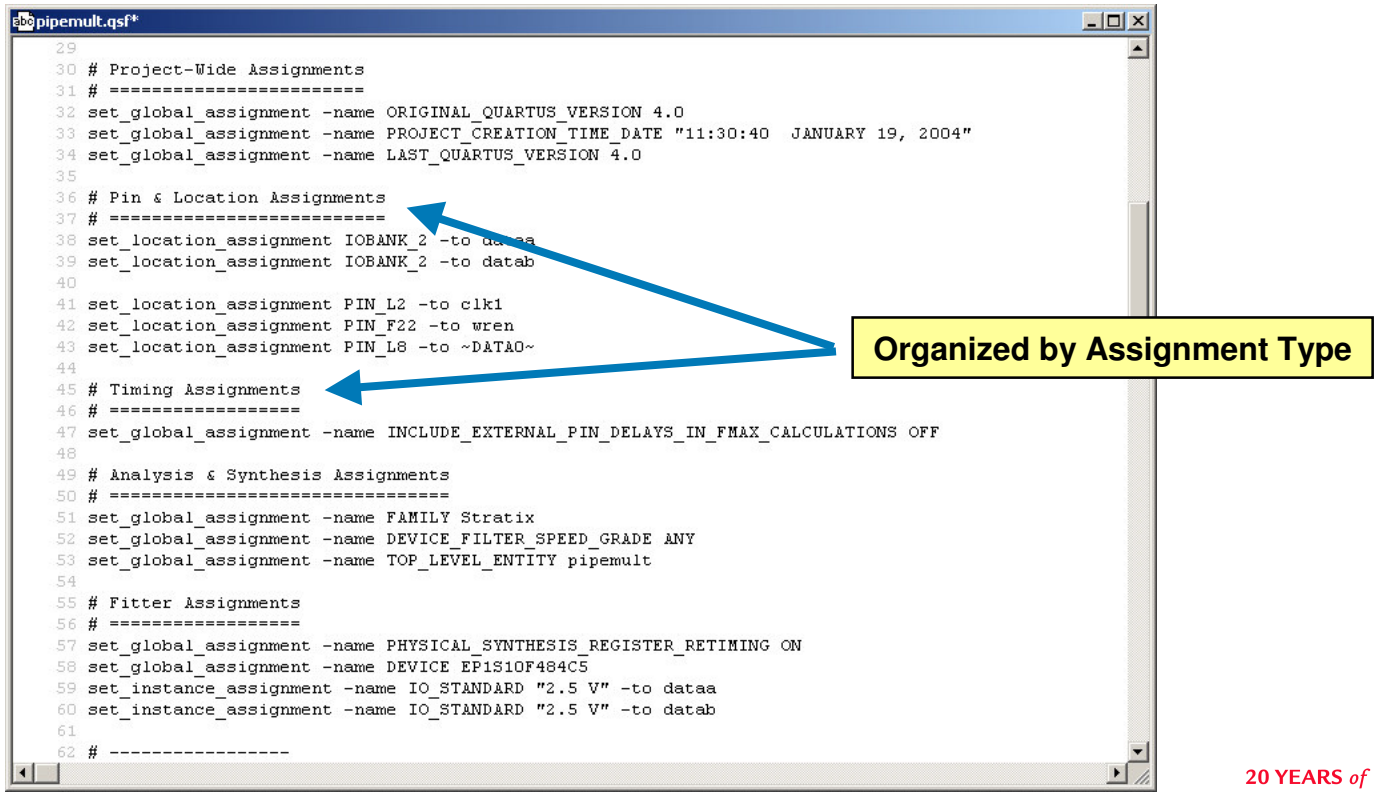


Using I/O Assignment Analysis

1. Use Complete Design Files or Reserve Pins if Incomplete
2. Make Pin-Related Assignments
3. Start I/O Assignment Analysis
4. Review Report File for Errors
5. Modify & Correct Illegal Assignments, if Any
6. Re-Run Analysis until Errors Are Resolved
7. Back-Annotate Pins to Lock Locations

Quartus Settings File (QSF)

- Stores All Settings & Assignments
- Uses Tcl Syntax

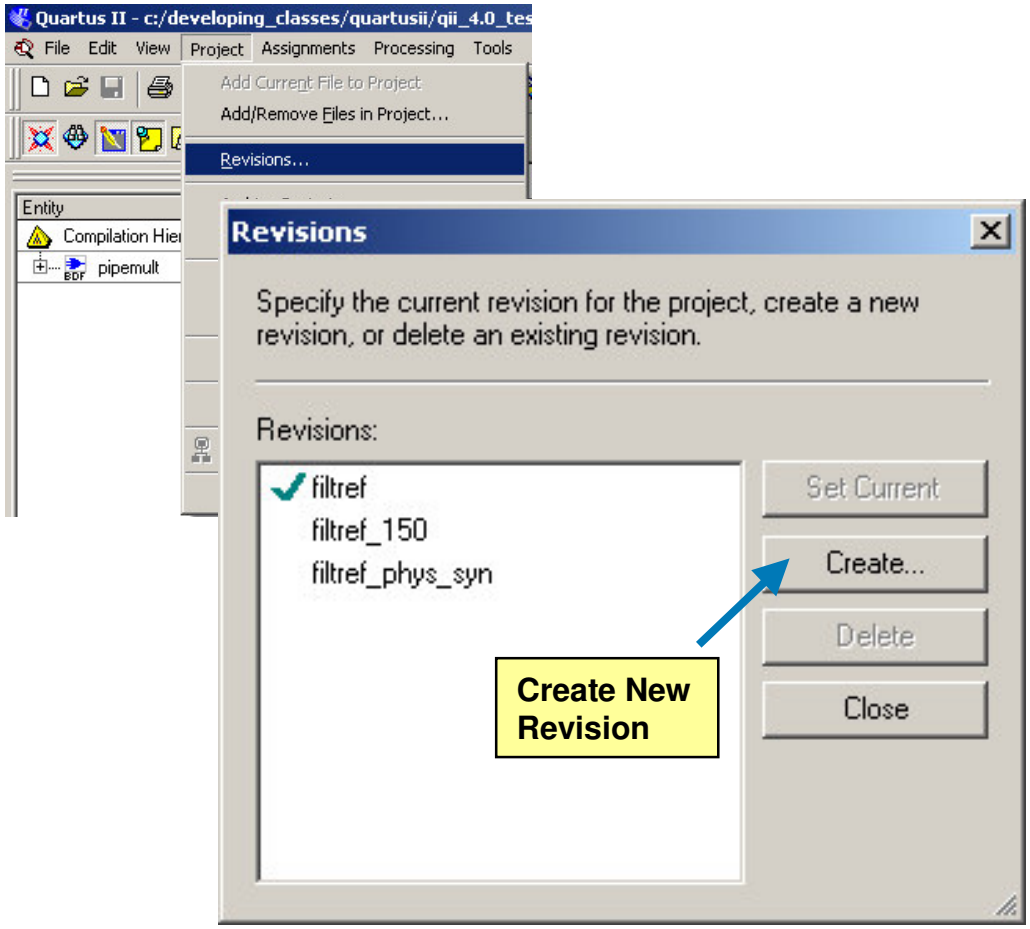


Design File Management

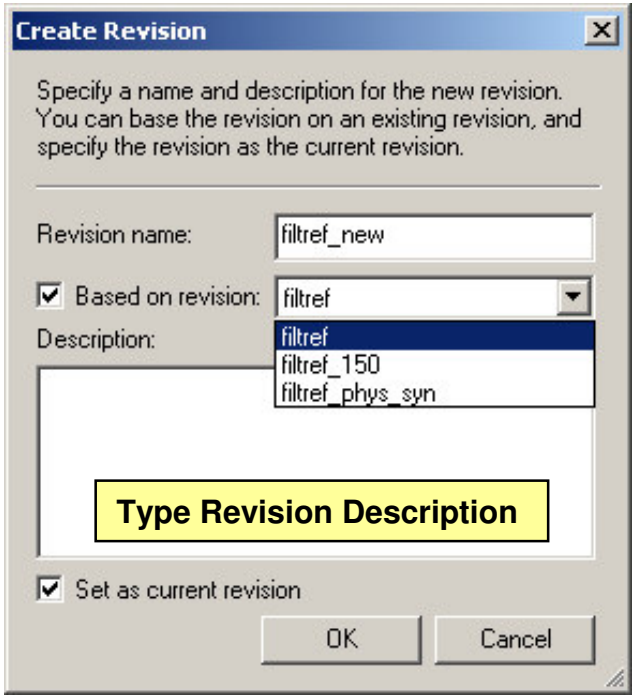
- Project Archive & Restore
  - Stores All Project Files
    - Design Files
    - Settings File
    - Output Files
- Revisions
  - Stores Only QSF
  - Allows Designer to Try Different Options

Creating a Revision

- Project ⇒ Revisions

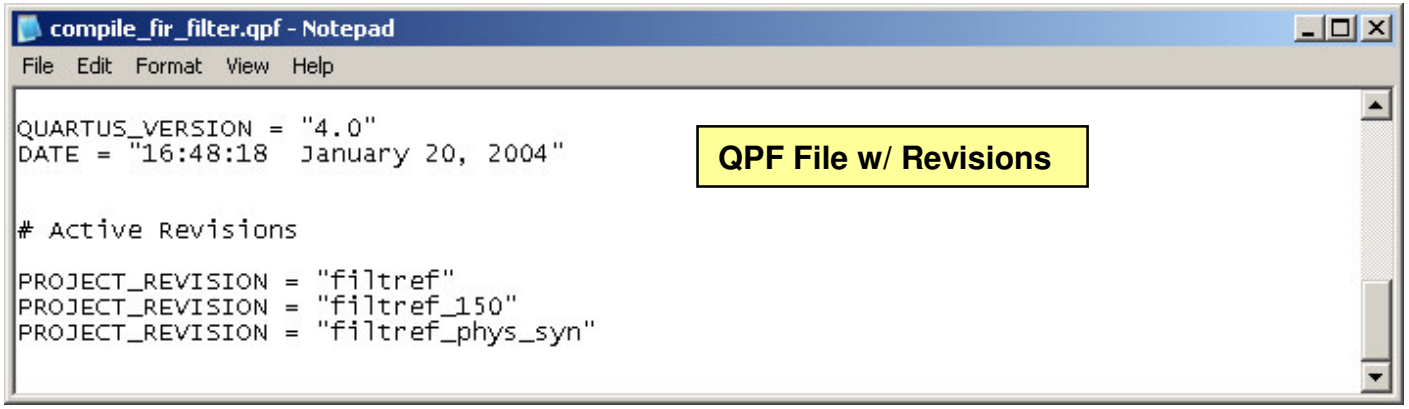


Base Revision on Any Previous Revision



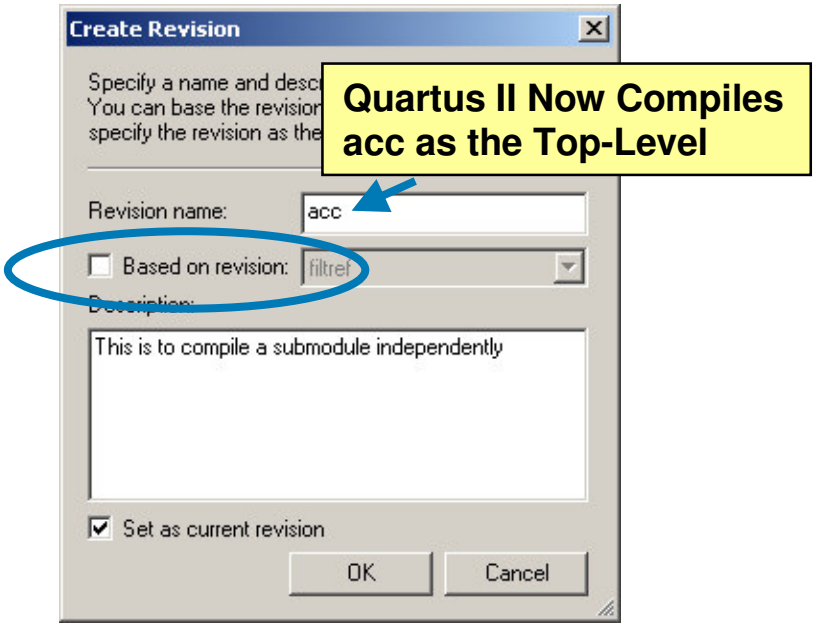
Project Revision Support

- Active Revision Names Stored in QPF
- QSF Created for Each Revision
  - <Revision_name>.QSF
- Text File Created for Each Revision
  - <Revision_Name>_description.TXT



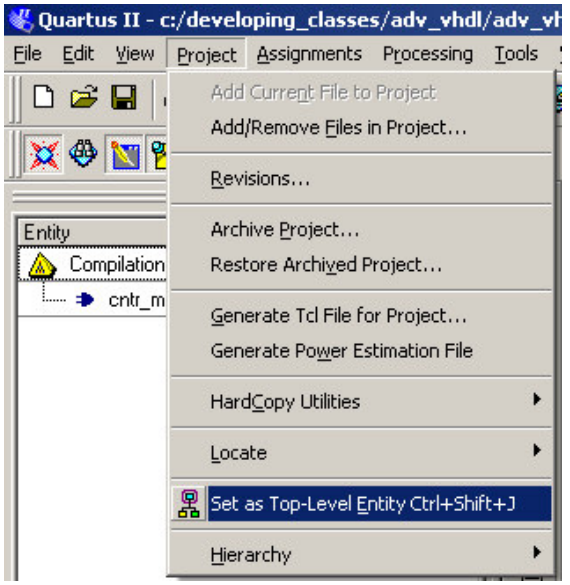
Compiling a Submodule

1. Create a New Revision
2. Disable “Based on revision”



Creates New .QSF File as a Revision

1. Open Submodule File
2. Select “Set as Top-Level Entity” from Project Menu



Reuses Current .QSF File

Exercise 2

Please go to Exercise 2 in the Exercise Manual

Exercise Summary

- Created Project
- Compiled Design
- Gathered Information from the Compilation Report
- Used Assignment Editor to Change Logic Usage
- Used I/O Analysis to Check I/O Placement

Compilation Summary

- Compiling a Design
  - Compiler Settings
  - Synthesis & Fitting Options
  - Compilation Report
- Assignments
  - Assignment Editor
  - I/O Assignment Analysis
- Managing Revisions

20 YEARS *of*

ALTERA®

INNOVATION

Designing with Quartus II

Timing Analysis

QUARTUS® II

Features

- Built-In Static Timing Analysis
- Single & Multi-Clock Timing Analysis
- Standard & Minimum Timing Analysis
- Timing Export for Analysis in Another EDA Tool

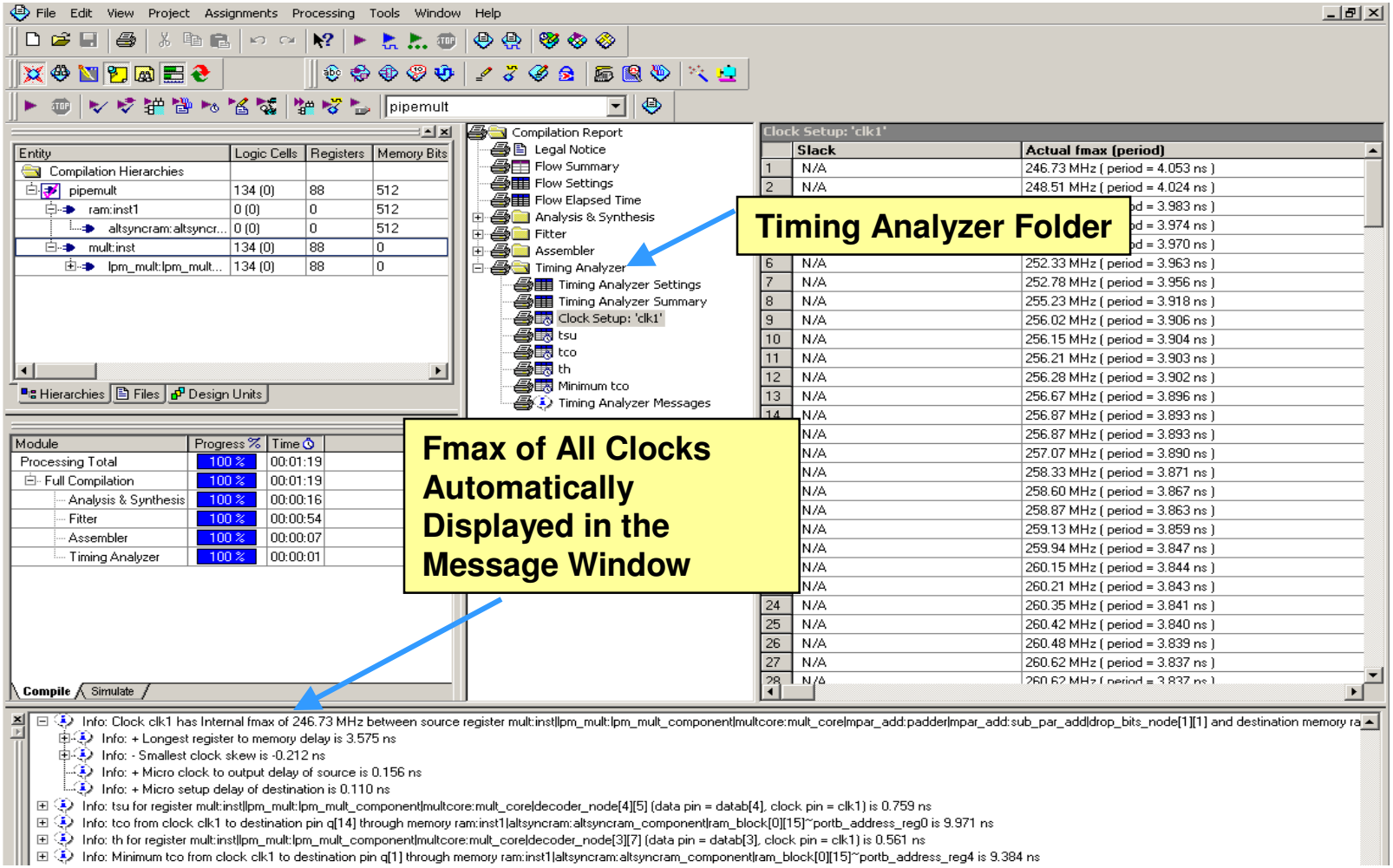
Timing Analysis Agenda

- Single Clock Analysis
- Timing Assignments
  - Global & Individual
- Multi-Clock Analysis
- Multi-Cycle Analysis
- Minimum Timing

Running Timing Analysis

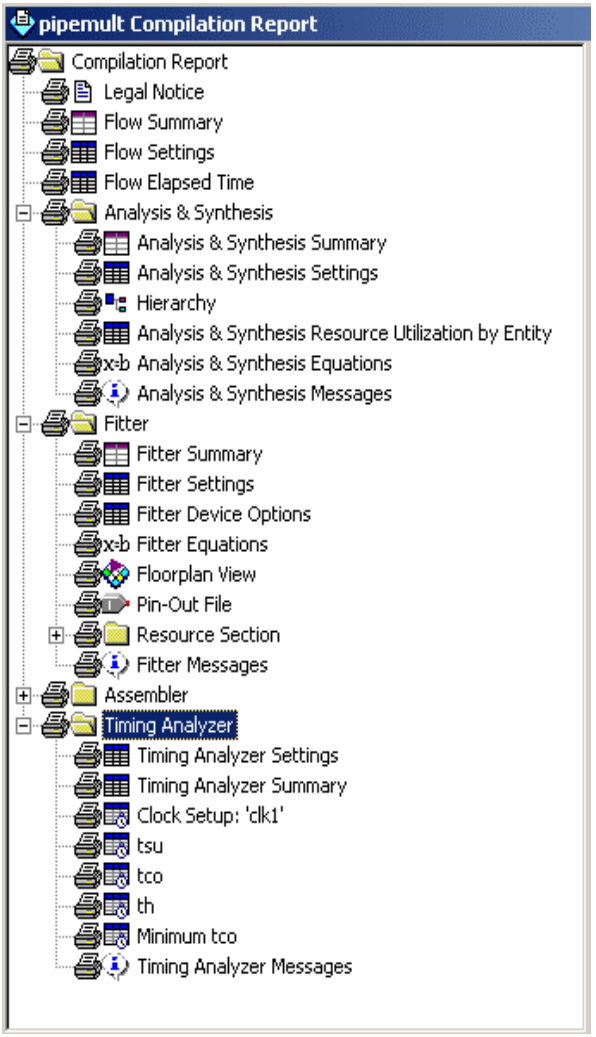
- Automatically
  - Use Full Compilation
- Manually
  - Processing Menu ⇒ Start ⇒ Start Timing Analysis
  - Tcl Scripts
  - Uses
    - Changing Speed Grade
    - Annotating Netlist with Delay Information

Timing Analysis



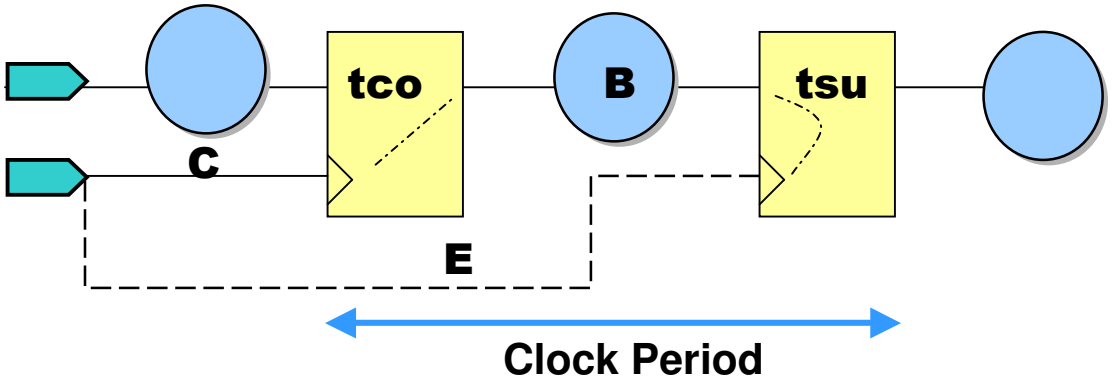
Reporting Timing Results

- Timing Analyzer Section of Compilation Report
  - Summary
    Timing Analyses
  - Clock Setup (fmax)
  - Clock Hold
  - tsu (Input Setup Times)
  - th (Input Hold Times)
  - tco (Clock to Out Delays)
  - tpd (Pin to Pin Delays)
  - Minimum tpd & tco



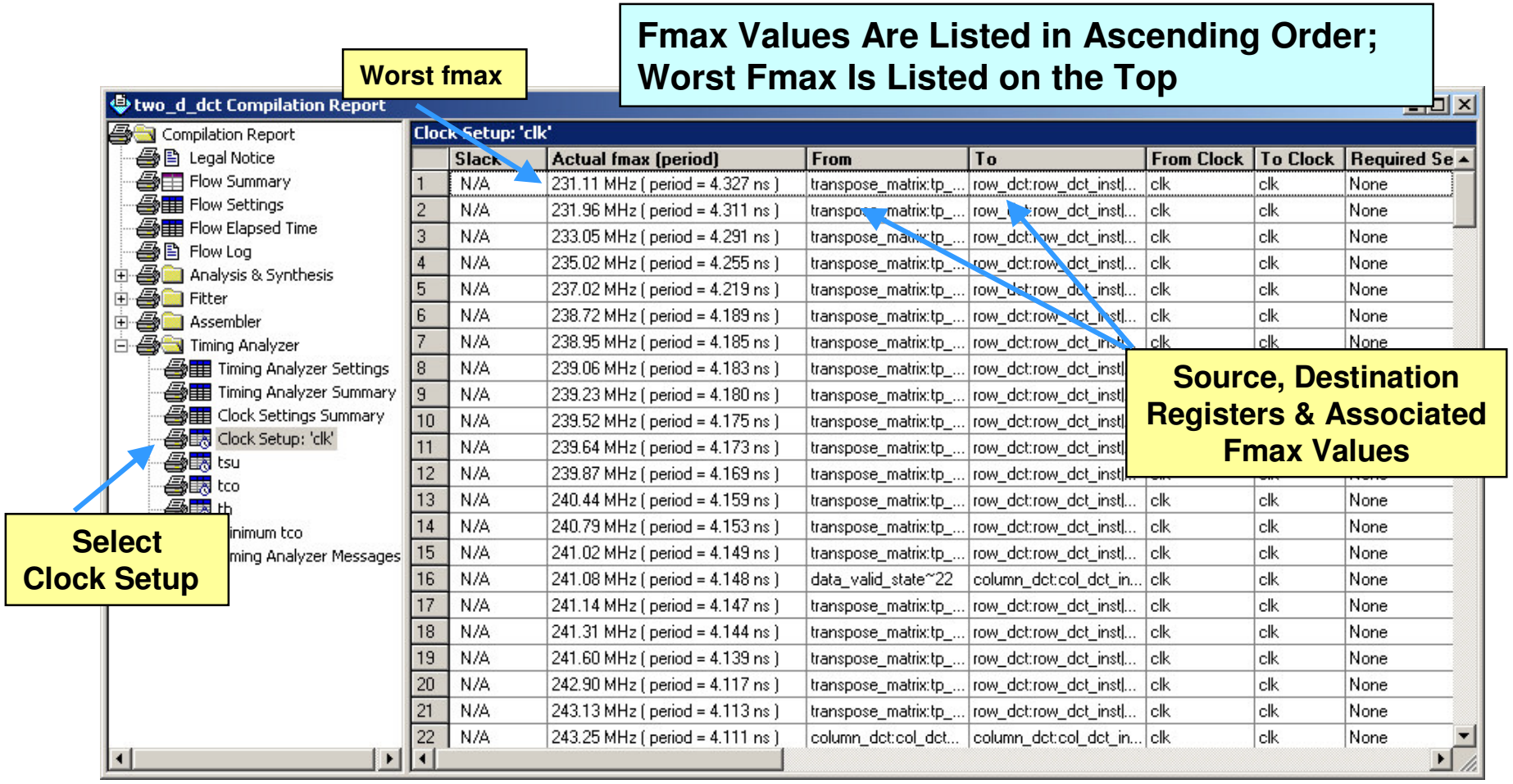
Clock Setup (fmax)

- Worst-Case Clock Frequency
  - Without Violating Internal Setup & Hold Times



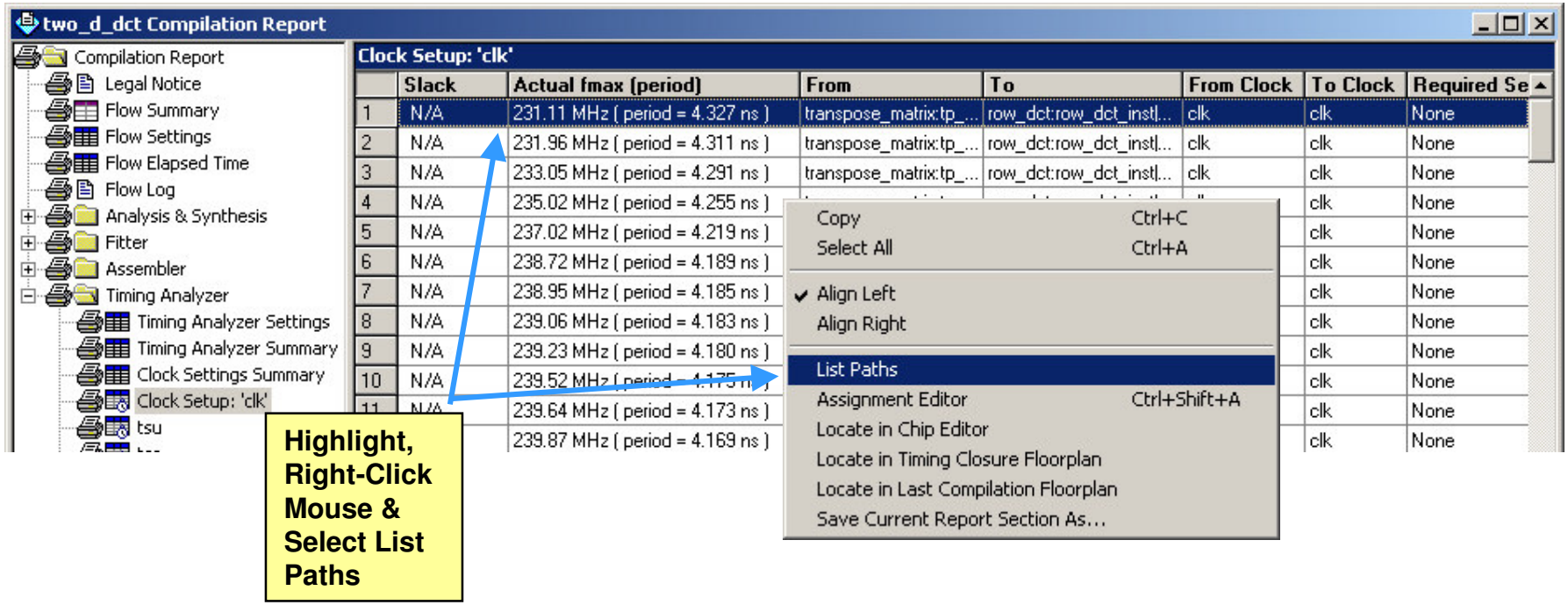
Clock Period	= Clock-to-Out + Data Delay + Setup Time - Clock Skew
	= tco + B + tsu - (E - C)
fmax	= 1/Clock Period

Clock Setup (fmax) Tables



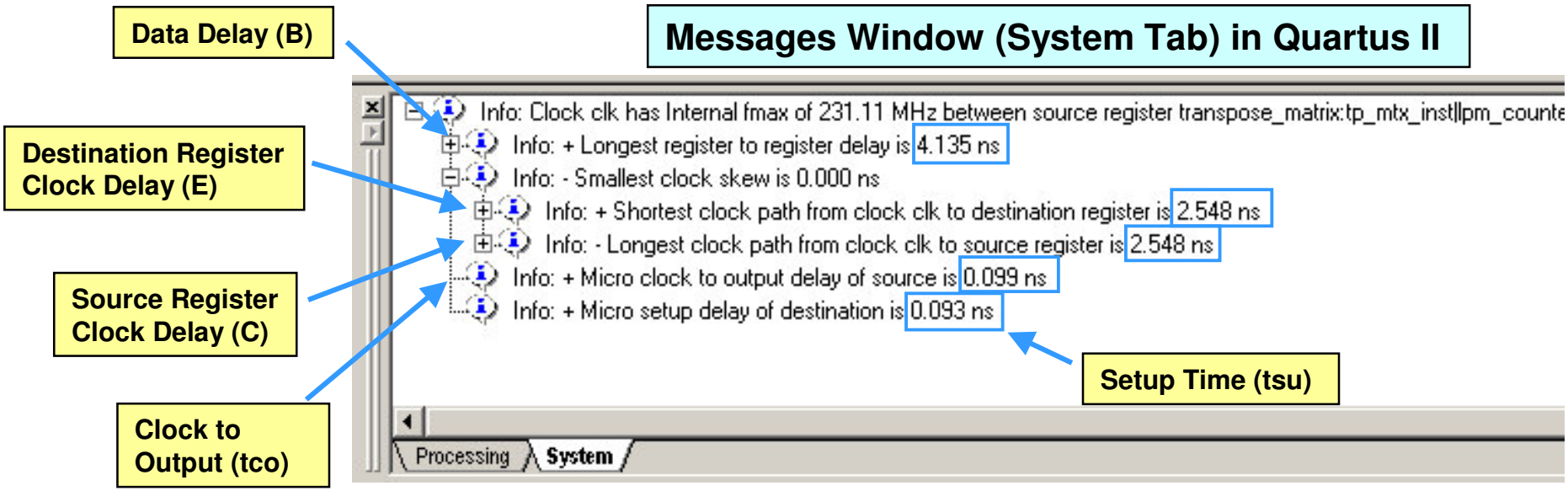
fmax Analysis

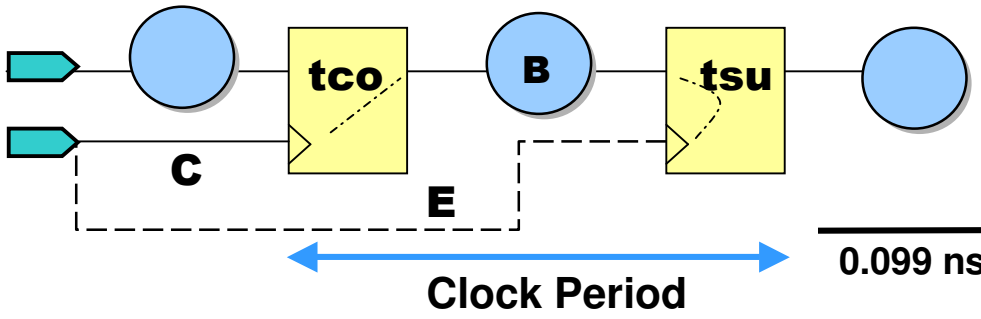
- To Analyze the Path More Closely



Similar Steps for All Timing Path Analysis in Quartus II

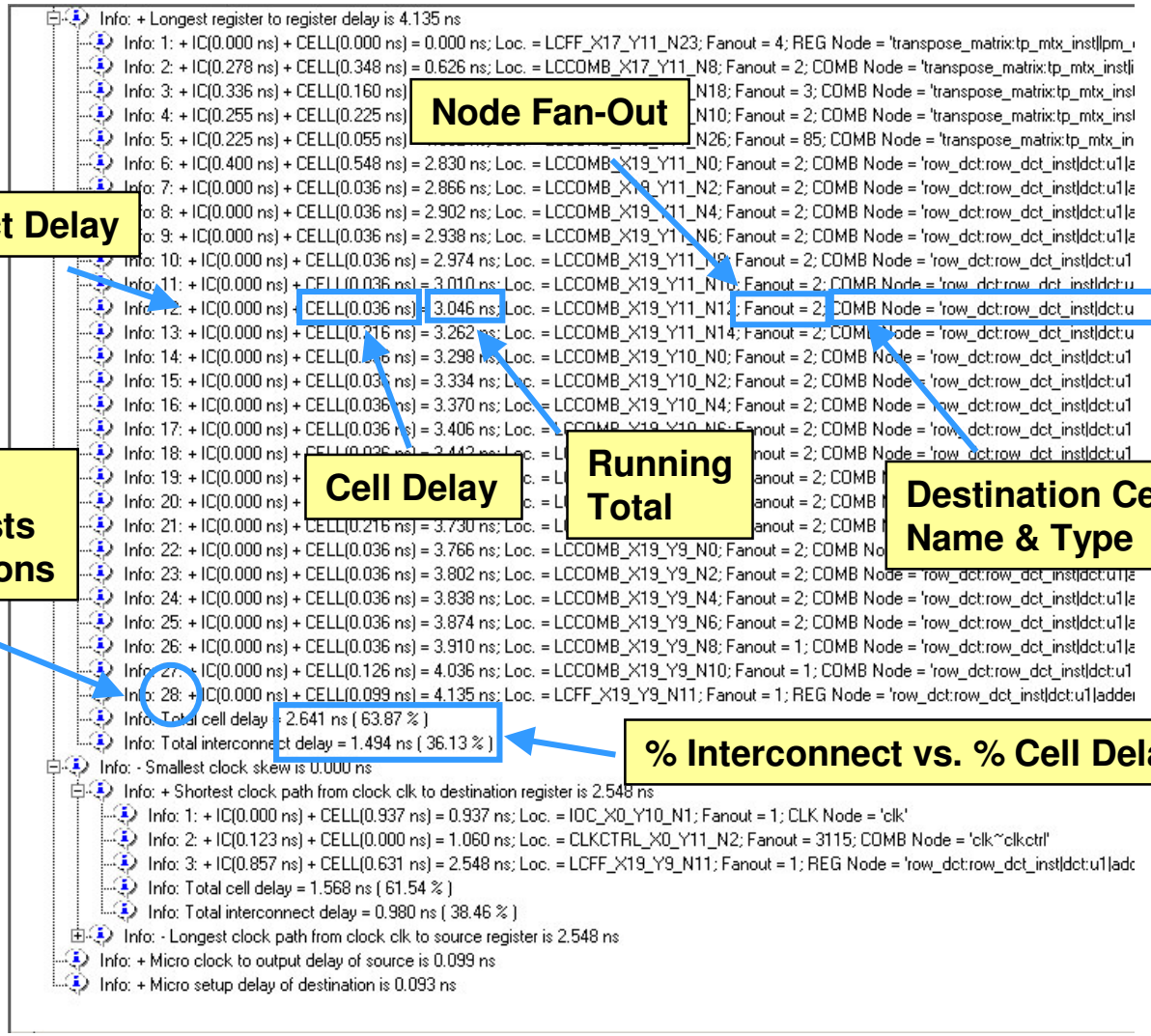
fmax Analysis Details





$$\frac{1}{0.099\text{ ns} + 4.135\text{ ns} + 0.093\text{ ns} - 0.000\text{ ns}} = 231.11\text{ MHz}$$

fmax Analysis Details (cont.)

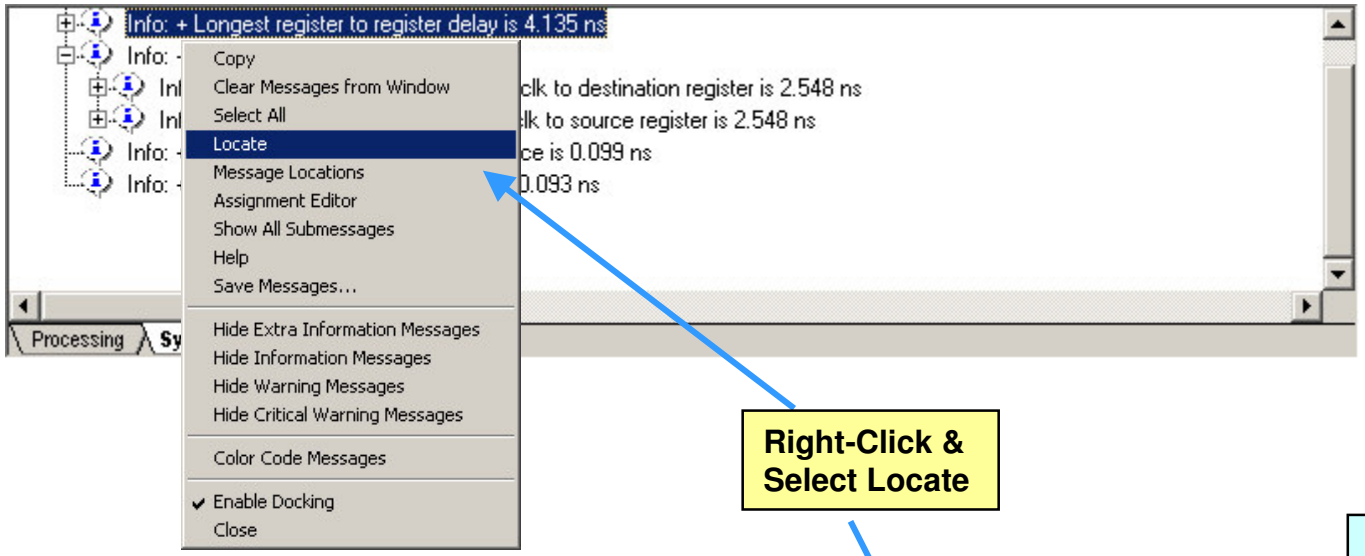


Notes:

1) This Timing Convention Is Similar for All Timing Path Analysis
2) Carry Chain Delay Is Included in Cell Delay Number

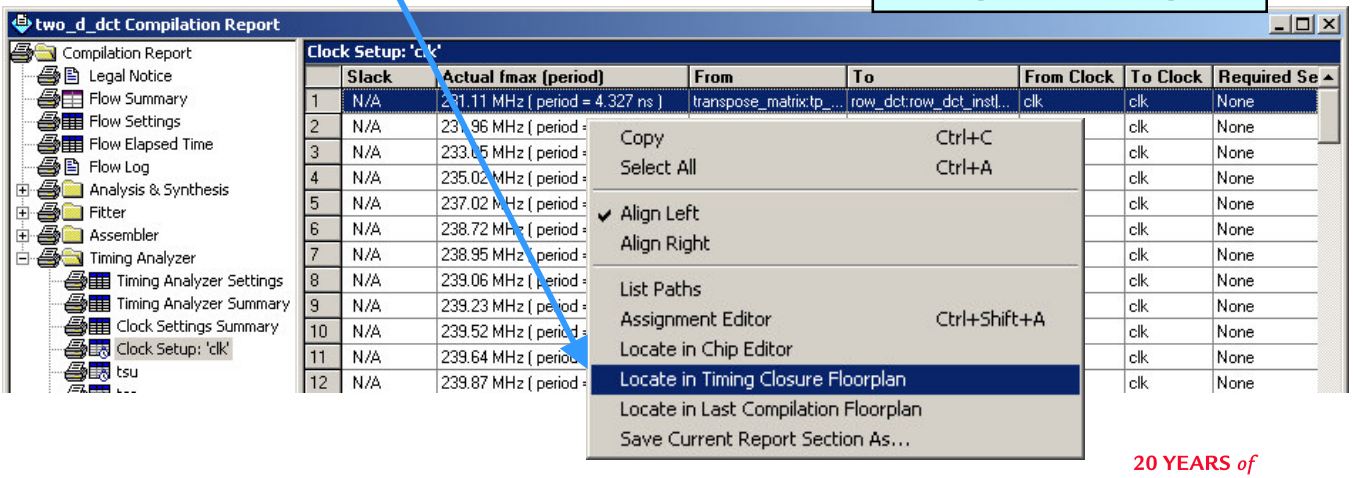
Locate Delay Path in Floorplan

Message Window



Right-Click & Select Locate

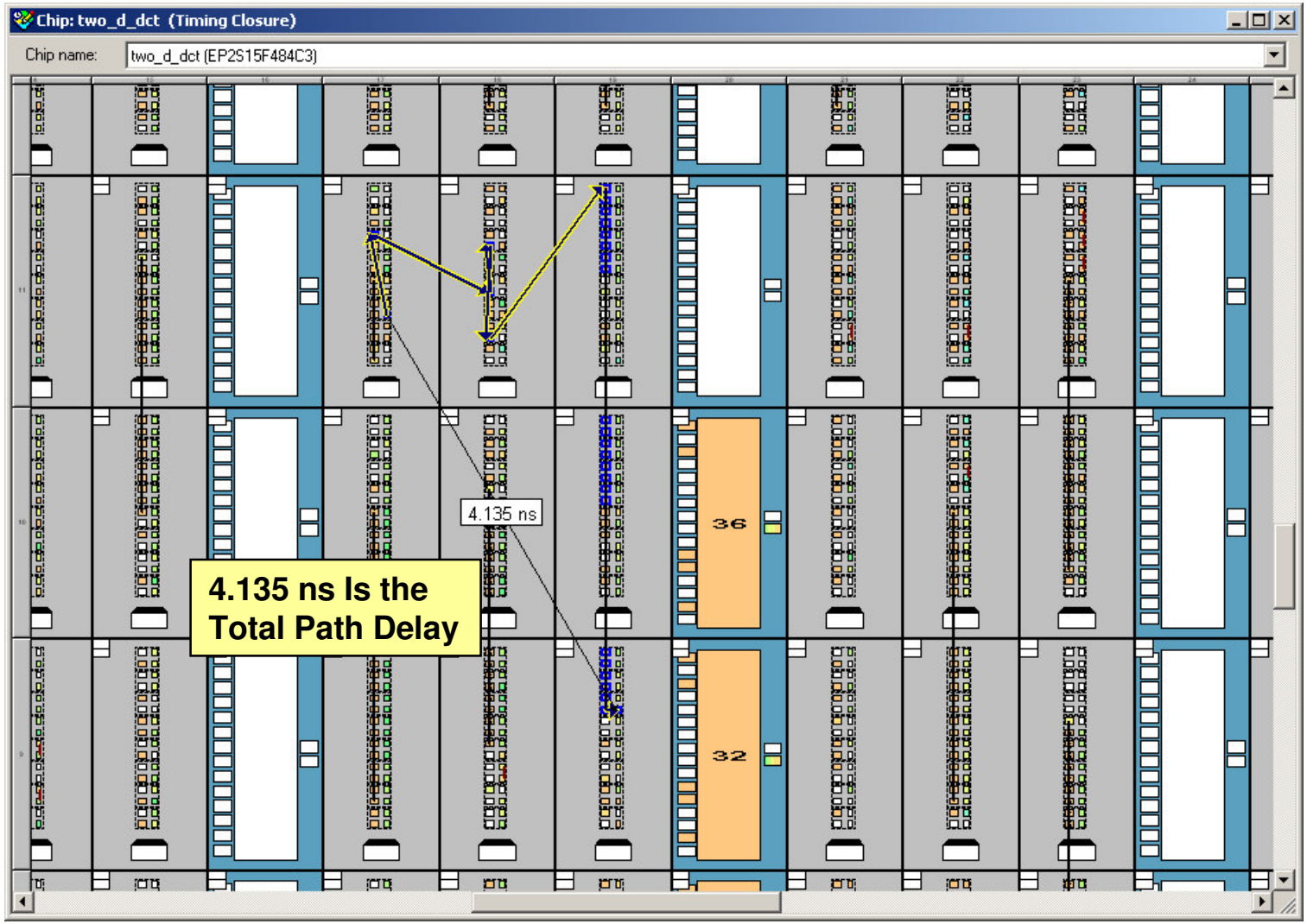
Compilation Report



Notes:

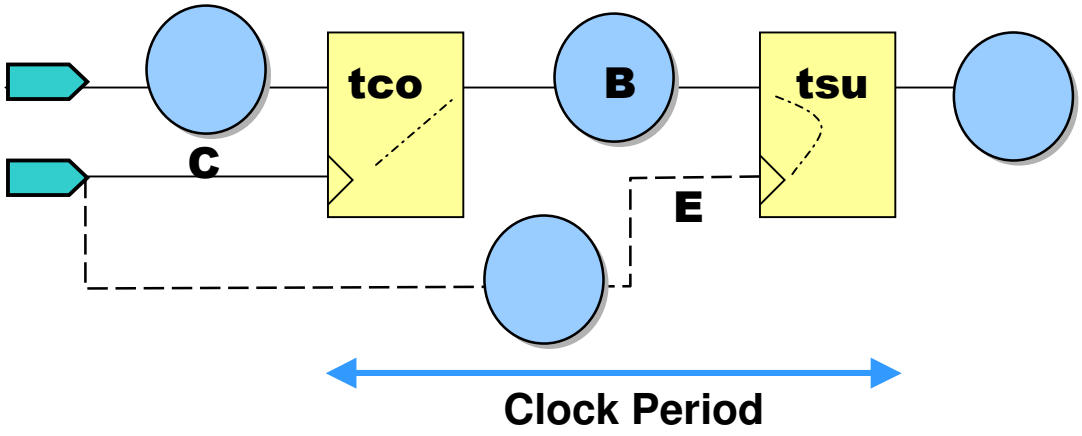
1) **Use Similar Procedure for All Timing Path Analysis**

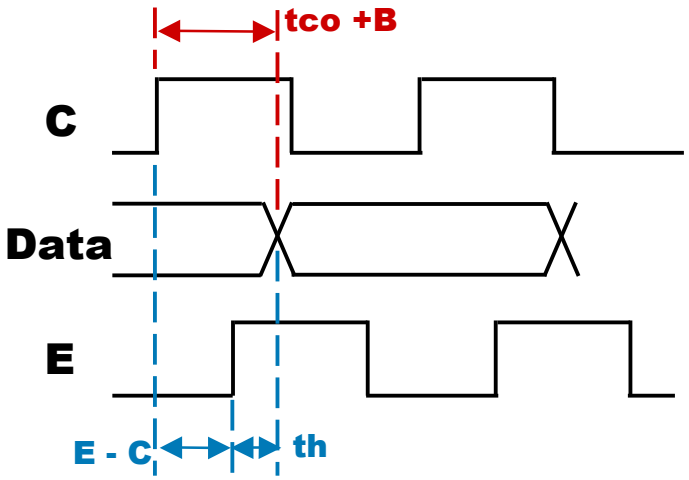
Locate Delay Path in Floorplan



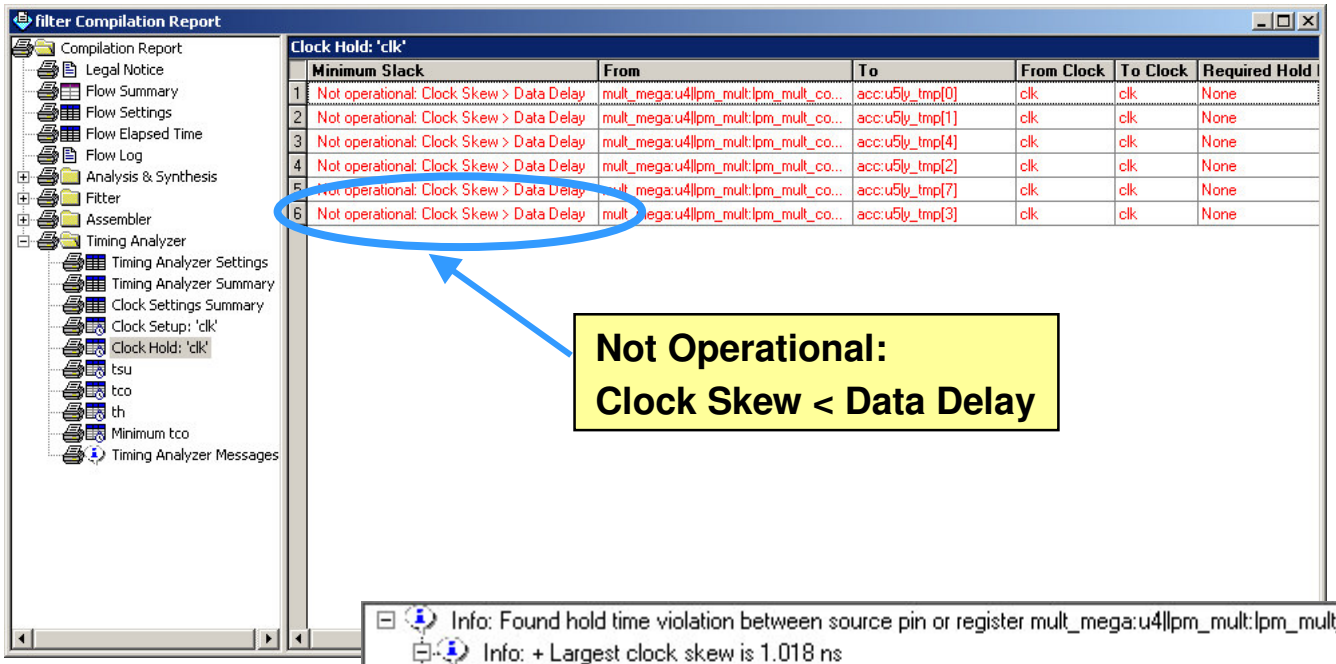
Clock Hold Analysis

- Checks Internal Register-Register Timing
  - Report Occurs Only When Hold Violations Occur
- Results Usually When Data Delay (B) is Greater than Clock Skew (E-C)
  - Non-Global Clock Routing
  - Gated Clocks



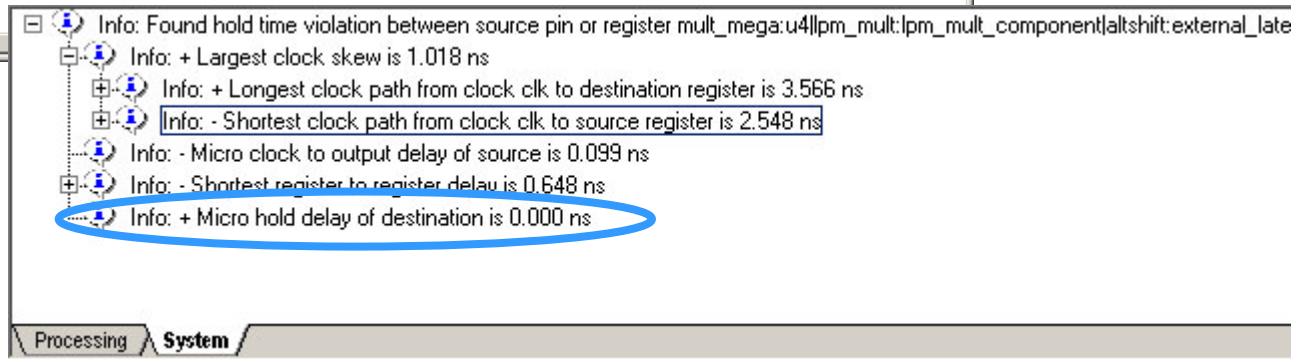


Hold Time Violations Table



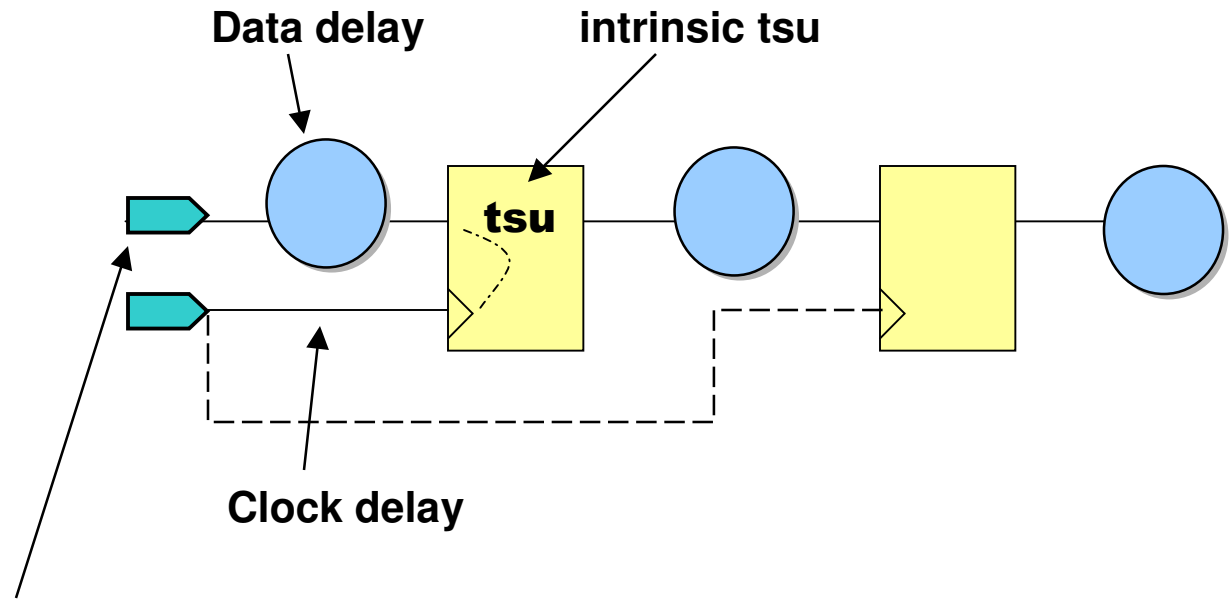
Not Operational:
Clock Skew < Data Delay

Discover Internal Hold Time Issues before Simulation



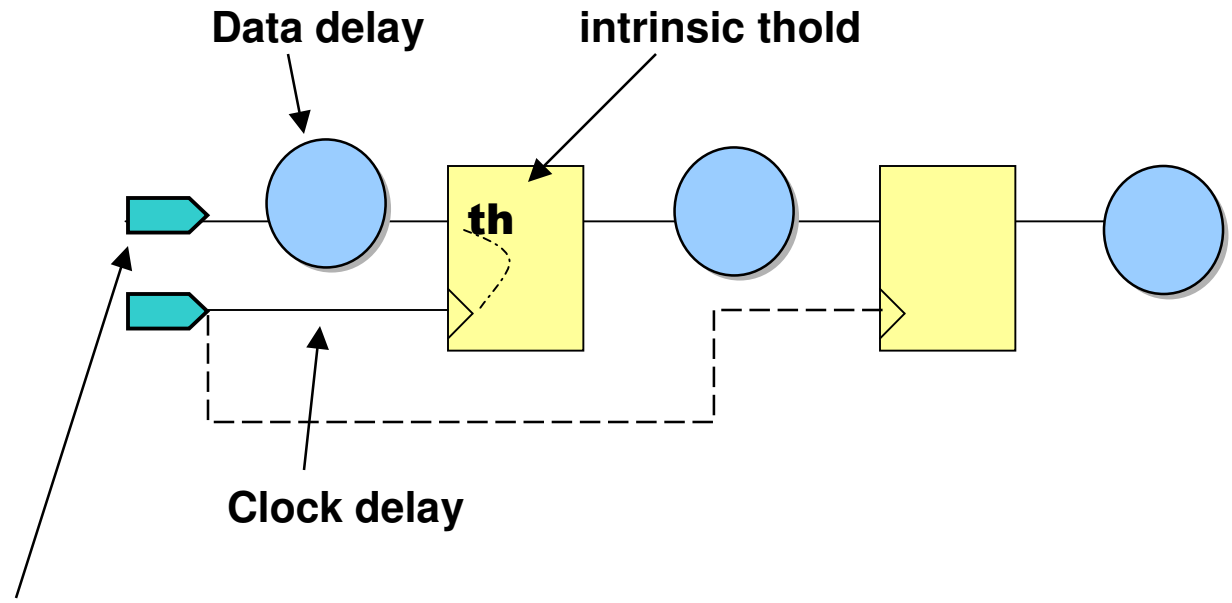
List Paths Window

I/O Setup Time Analysis (tsu)



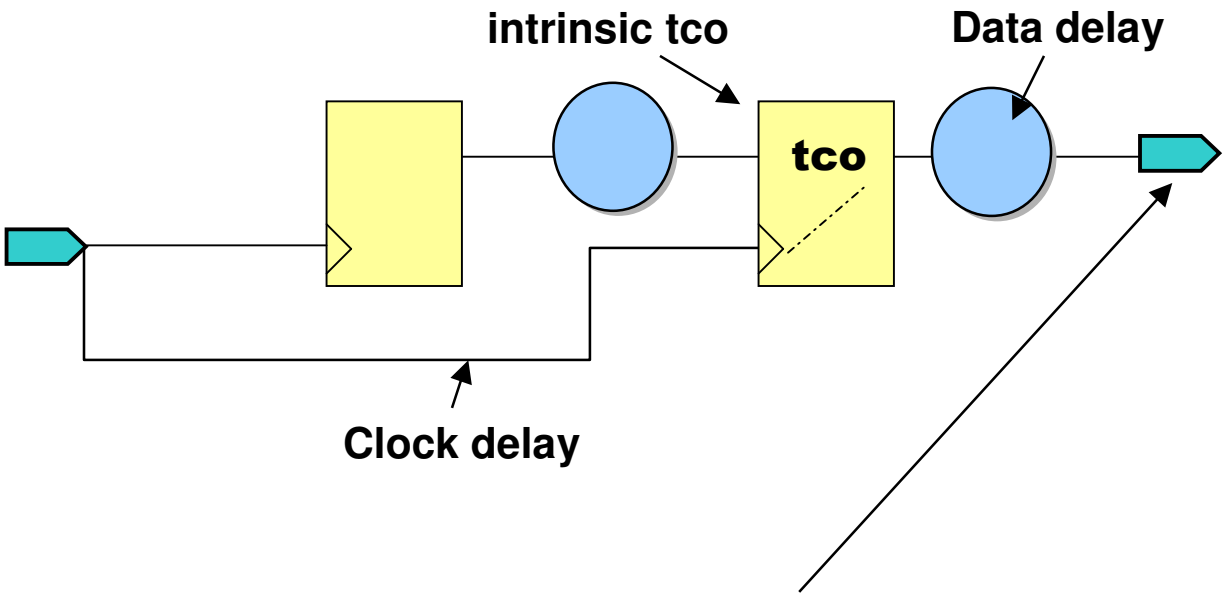
tsu = data delay - clock delay + intrinsic tsu

I/O Hold Time Analysis (th)



th = clock delay - data delay + intrinsic thold

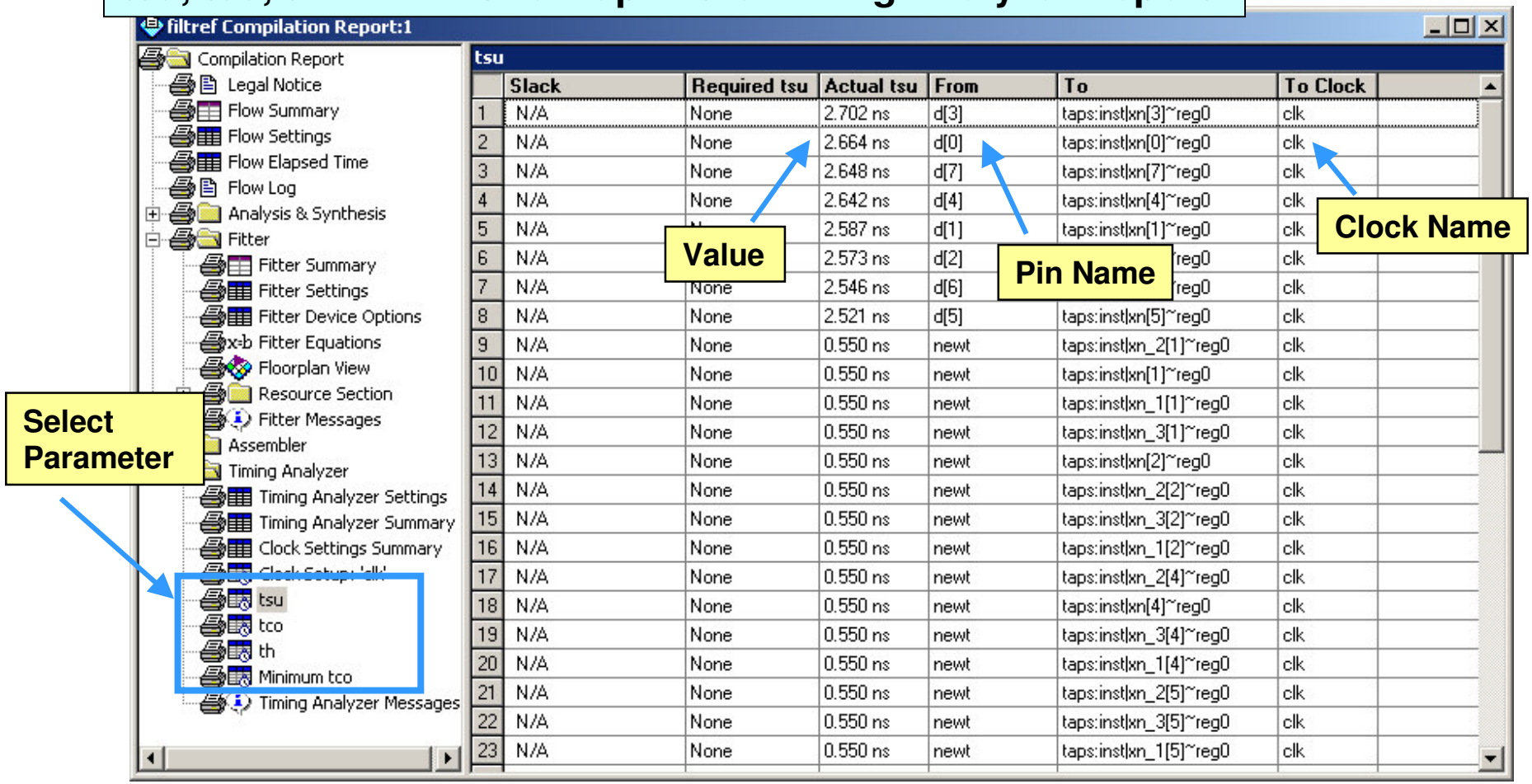
I/O Clock-to-Output Analysis (tco)



clock delay + intrinsic tco + data delay = tco

I/O Timing Analyzer

tsu, tco, th Will All Show up in the Timing Analyzer Report



tsu

	Slack	Required tsu	Actual tsu	From	To	To Clock
1	N/A	None	2.702 ns	d[3]	taps:inst|xn[3]~reg0	clk
2	N/A	None	2.664 ns	d[0]	taps:inst|xn[0]~reg0	clk
3	N/A	None	2.648 ns	d[7]	taps:inst|xn[7]~reg0	clk
4	N/A	None	2.642 ns	d[4]	taps:inst|xn[4]~reg0	clk
5	N/A		2.587 ns	d[1]	taps:inst|xn[1]~reg0	clk
6	N/A		2.573 ns	d[2]	reg0	clk
7	N/A	None	2.546 ns	d[6]	reg0	clk
8	N/A	None	2.521 ns	d[5]	taps:inst|xn[5]~reg0	clk
9	N/A	None	0.550 ns	newt	taps:inst|xn_2[1]~reg0	clk
10	N/A	None	0.550 ns	newt	taps:inst|xn[1]~reg0	clk
11	N/A	None	0.550 ns	newt	taps:inst|xn_1[1]~reg0	clk
12	N/A	None	0.550 ns	newt	taps:inst|xn_3[1]~reg0	clk
13	N/A	None	0.550 ns	newt	taps:inst|xn[2]~reg0	clk
14	N/A	None	0.550 ns	newt	taps:inst|xn_2[2]~reg0	clk
15	N/A	None	0.550 ns	newt	taps:inst|xn_3[2]~reg0	clk
16	N/A	None	0.550 ns	newt	taps:inst|xn_1[2]~reg0	clk
17	N/A	None	0.550 ns	newt	taps:inst|xn_2[4]~reg0	clk
18	N/A	None	0.550 ns	newt	taps:inst|xn[4]~reg0	clk
19	N/A	None	0.550 ns	newt	taps:inst|xn_3[4]~reg0	clk
20	N/A	None	0.550 ns	newt	taps:inst|xn_1[4]~reg0	clk
21	N/A	None	0.550 ns	newt	taps:inst|xn_2[5]~reg0	clk
22	N/A	None	0.550 ns	newt	taps:inst|xn_3[5]~reg0	clk
23	N/A	None	0.550 ns	newt	taps:inst|xn_1[5]~reg0	clk

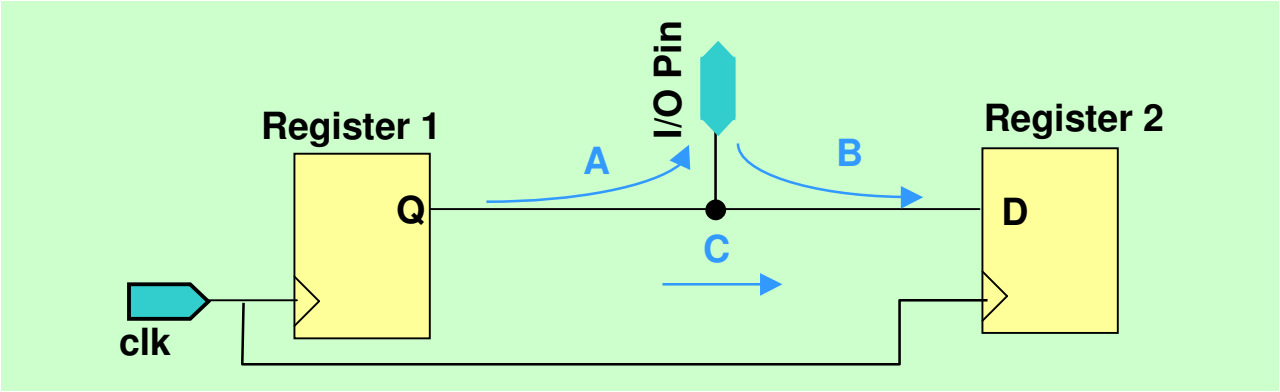
***Note:** Timing Analysis of tpd is similar*

Timing Analysis Options

- Used to Limit which Paths Are Displayed
- Global Cut Timing Options (On by Default)
  - Cut Paths between Unrelated Clock Domains
  - Cut Off Feedback from I/O Pins (Next Slide)
  - Cut Off Clear & Preset Signal Paths
  - Cut Off Read during Write Signal Paths
- Timing Analyzer Options
  - Display Paths that Do Not Meet Timing Only

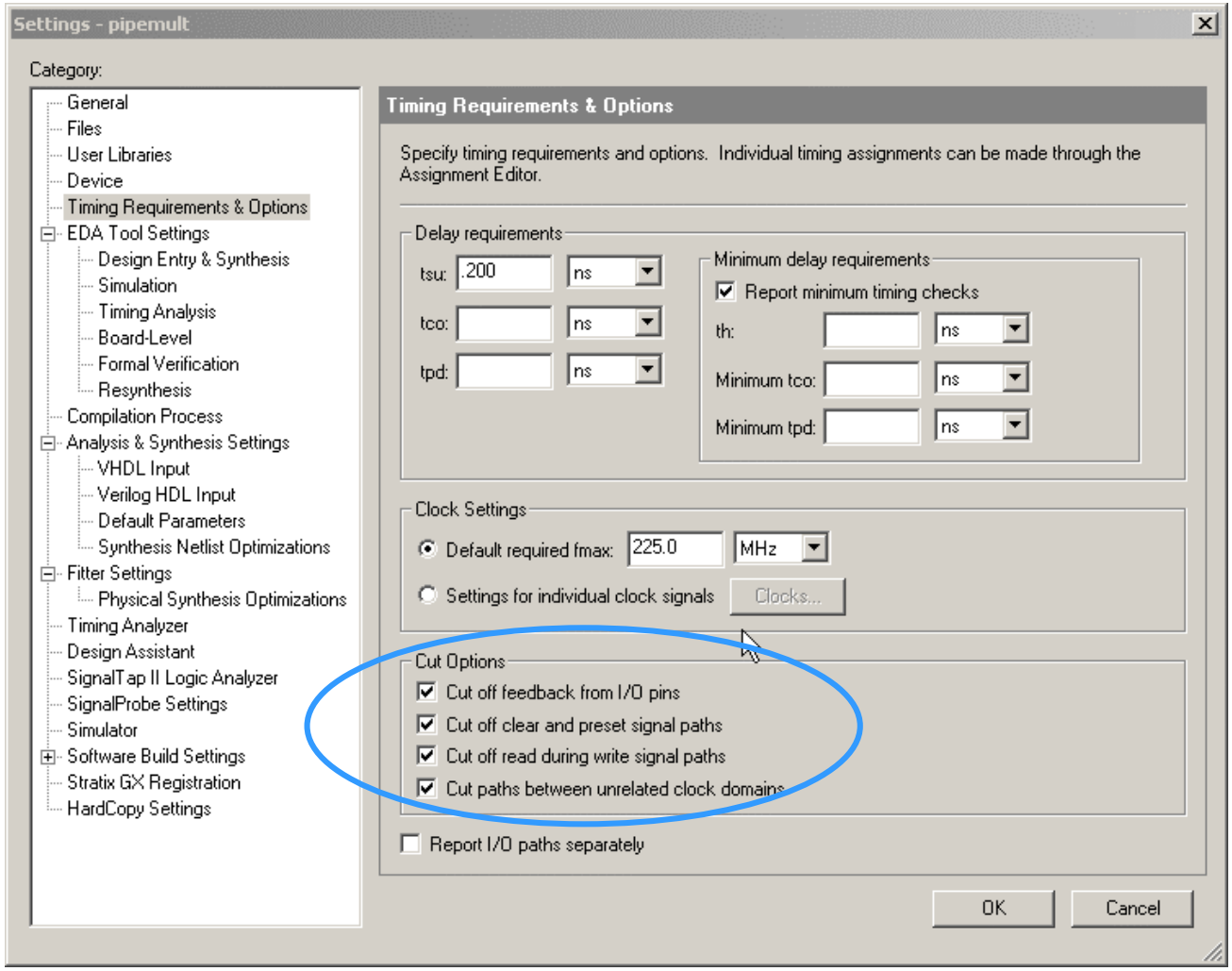
Cut Off Feedback from I/O Pin

- Breaks Bidirectional I/O Pin from Analysis
- When On, Paths A & B Are Valid; C Is Not
- When Off, Paths A, B, & C Are Valid



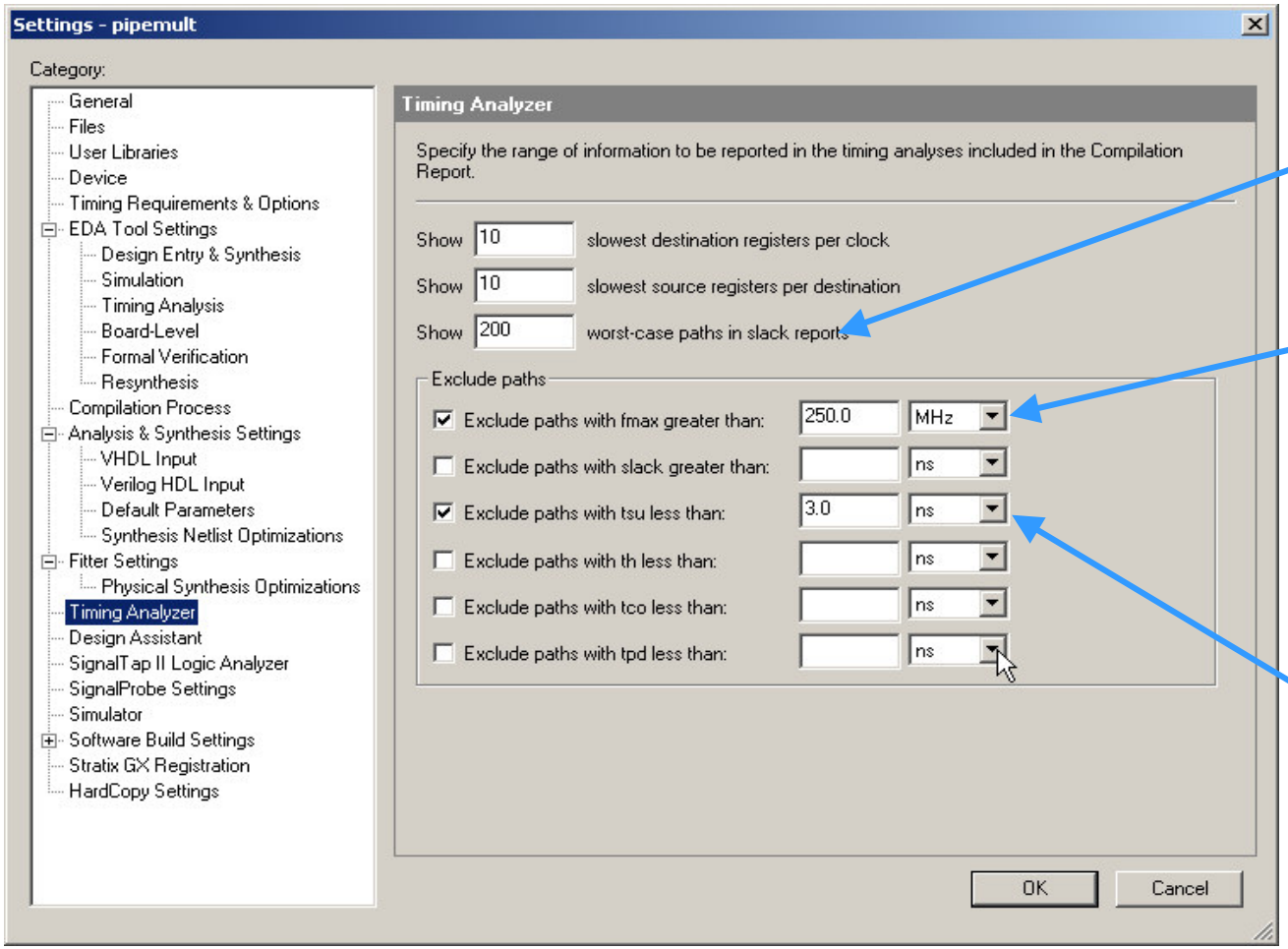
Cut Options

Assignments ⇒ Settings ⇒ Timing Requirements & Options



Timing Analyzer Options

Assignments ⇒ Settings ⇒ Timing Analyzer



List 200 Paths

List Paths with Fmax Less than 250 MHz

List Paths with Tsu Greater than 3 ns

Single-Clock Timing Analysis

- Performed Automatically during Each Compile
- Detects Clocks Automatically If No Assignments Are Made
  - Single or Multiple Asynchronous Clock Domains
  - Still Important to Specify Clock Frequencies
- Displays Results in the Timing Analyzer Report

Exercise 3

Please go to Exercise 3 in the Exercise Manual

Exercise Summary

- Single Clock Timing Analysis
- List Timing Paths & View Details
  - Locate in Floorplan

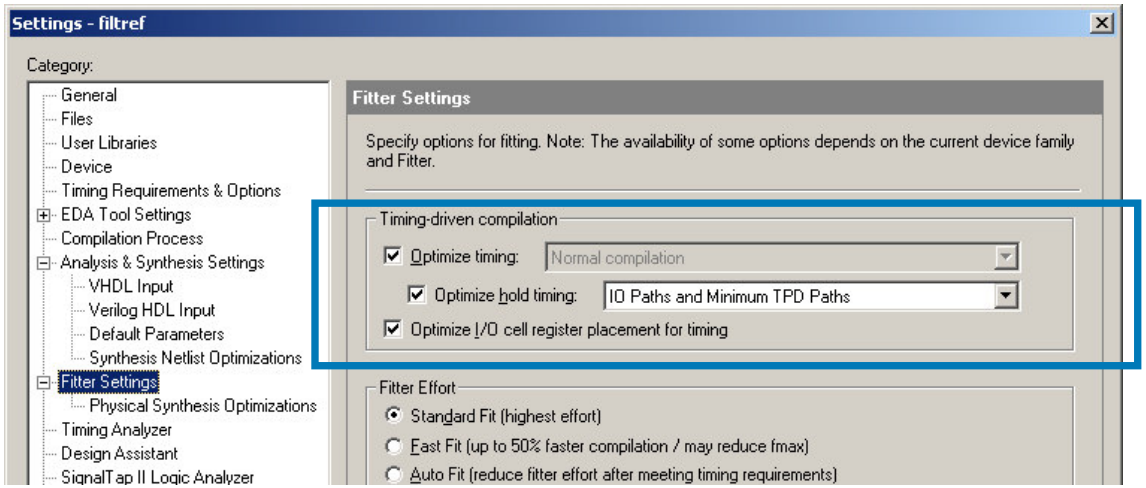
Timing Assignments

- **VERY IMPORTANT!!**
- Have a Major Impact on Design Compilation
  - Specify **ALL** Timing Requirements for Your Design
  - Fitter Works Hardest on the Worst Timing
  - Timing Will Be Reported in Red If Not Met
- Types
  - Internal & I/O Timing
  - Maximum & Minimum
- Can Be Assigned Globally or Individually
  - Individual Assignments Better

Timing Driven Compilation (TDC)

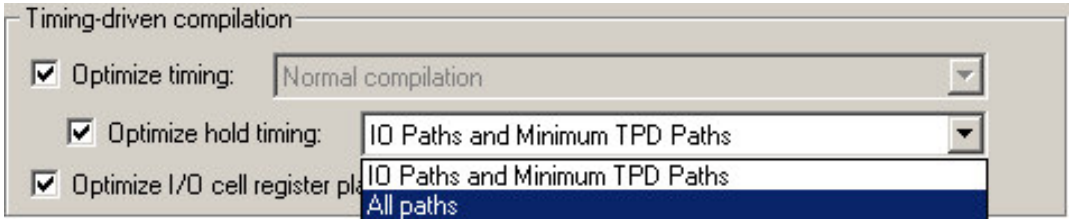
- Directs Fitter to Place & Route Logic to Meet Timing Assignments
  - Optimize Timing
    - Placing Nodes in Critical Paths Closer Together
  - Optimize I/O Cell Register Placement
    - Moving Registers into I/O Cells

Assignments ⇒ Settings ⇒ Fitting Settings



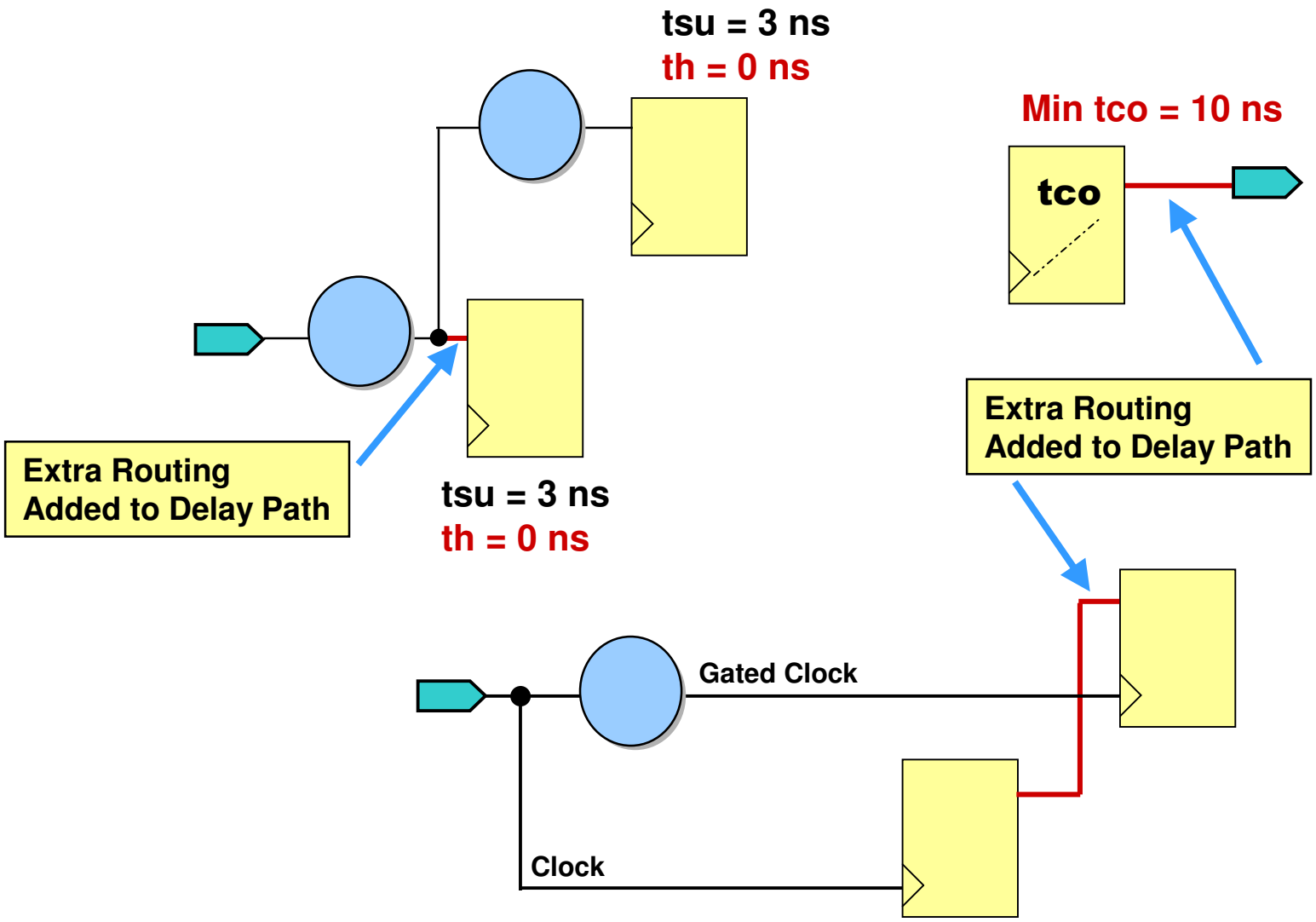
Optimize Hold Timing

- Modifies Place & Route to Meet Hold or Minimum Timing Requirements
  - May Add Additional Routing in Path
  - Supported in Stratix II, Stratix, Stratix GX, Cyclone Devices



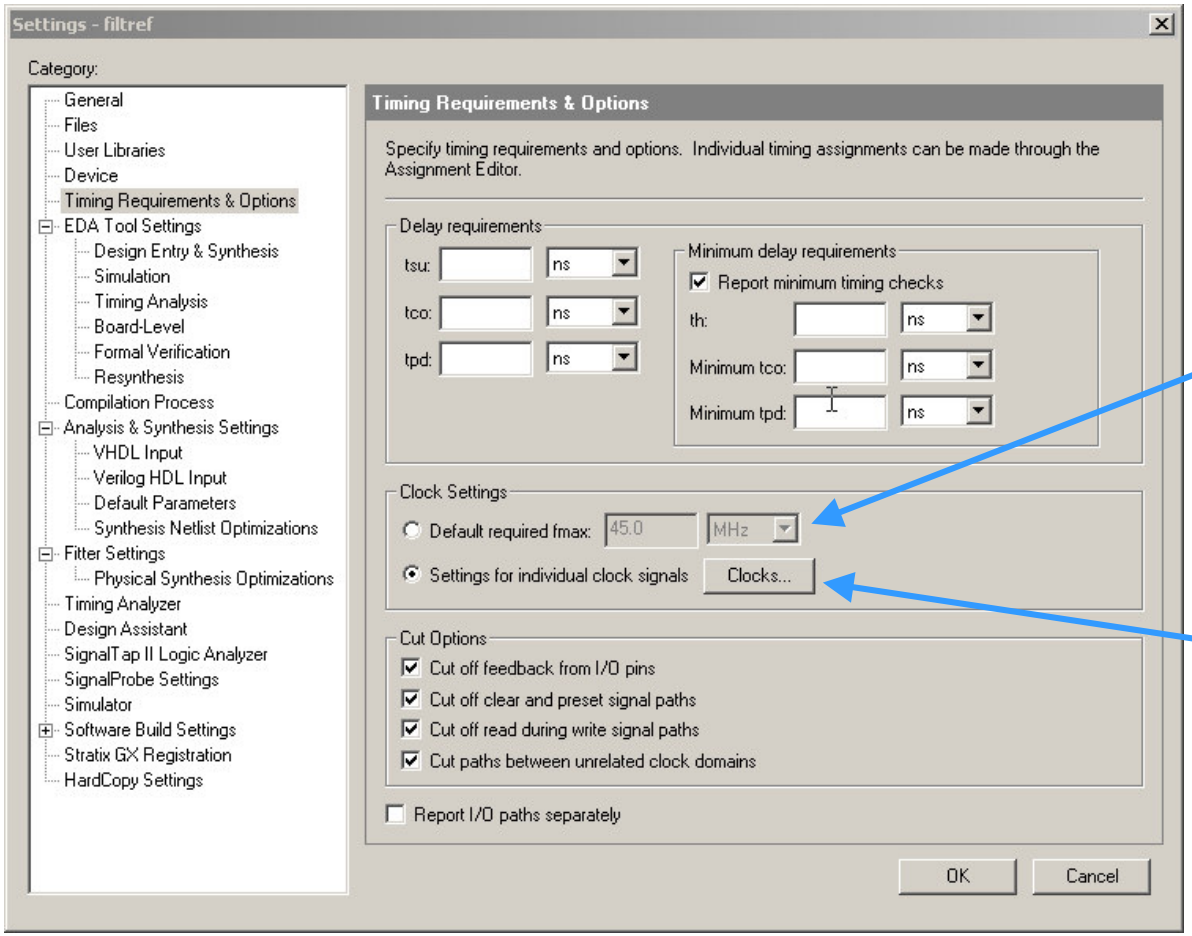
- Settings
  - Any I/O Paths
  - All Paths (I/O & Internal)

Optimize Hold Time Examples



Global Clock Assignments

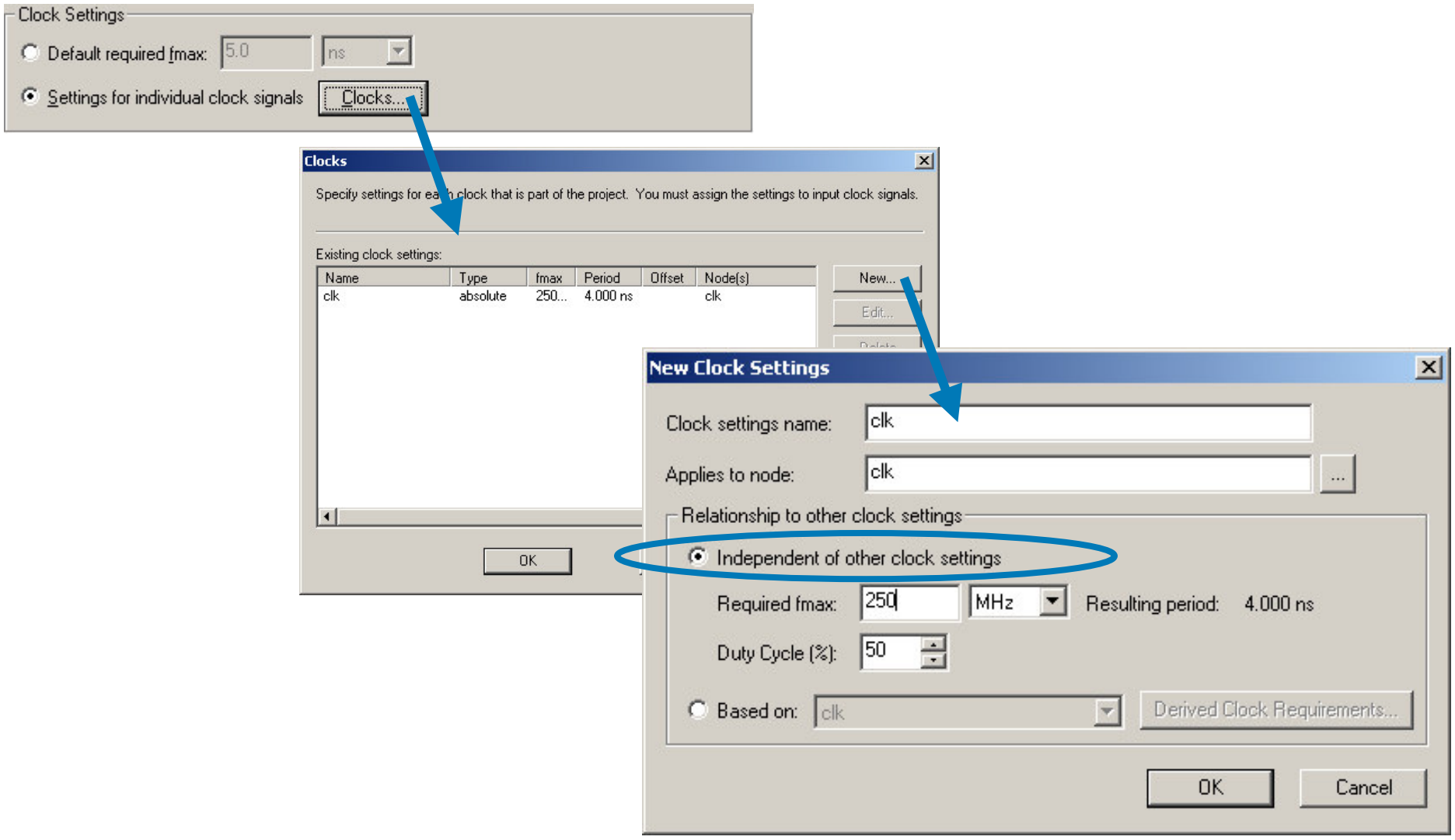
Assignments ⇒ Settings ⇒ Timing Requirements & Options



Global Clock Assignment for a Single Clock Design

For Designs with Multiple Asynchronous Clocks, Enter Required Fmax for Each Individual Clock

Asynchronous Global Clocks



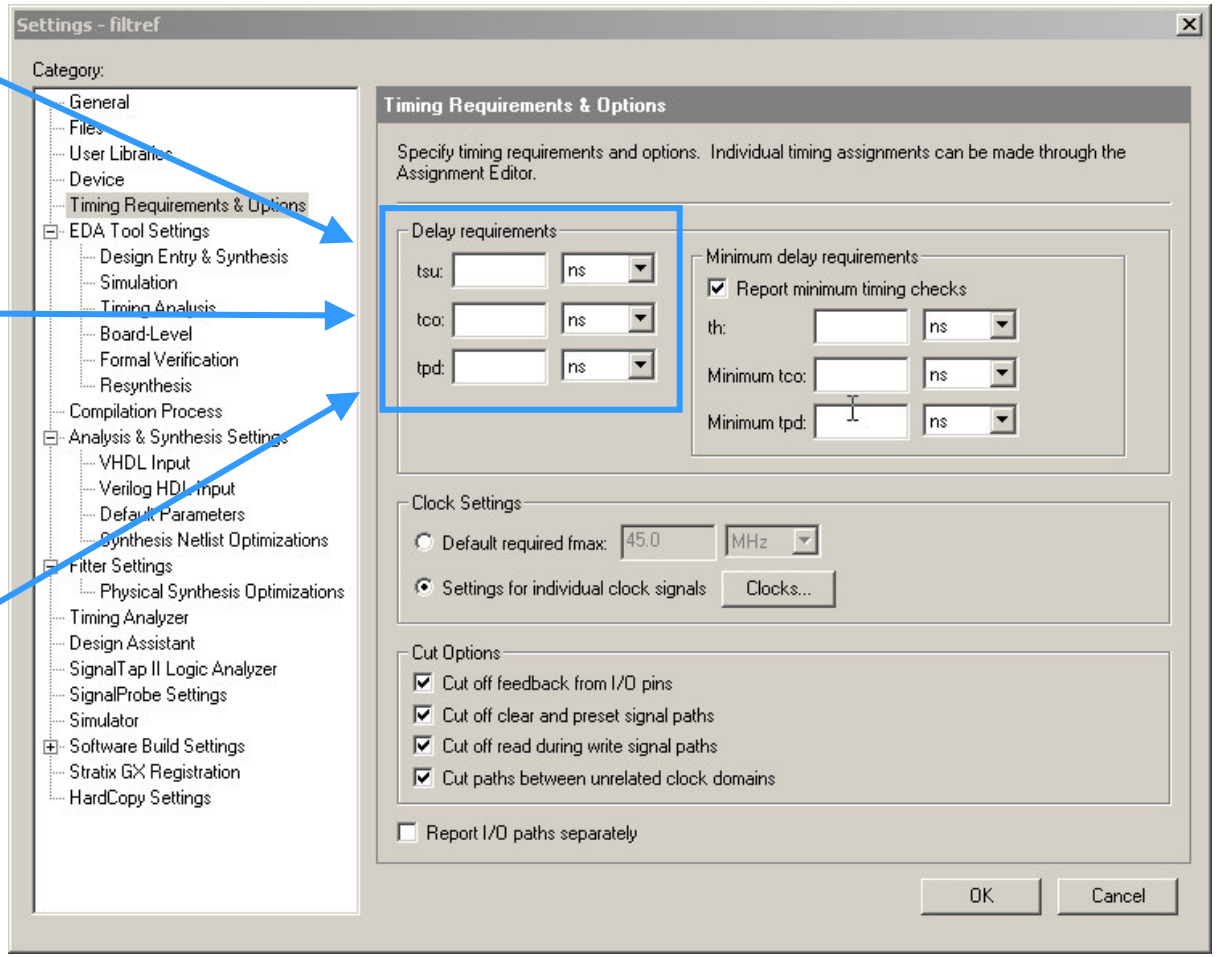
Global Maximum I/O Delay

Global tsu
All Inputs or Bidirectional Pins Driving Registers Must Have a tsu Less Than This Value

Global tco
All Registers Driving Outputs or Bidirectional Pins Must Have a tco Less Than this Value

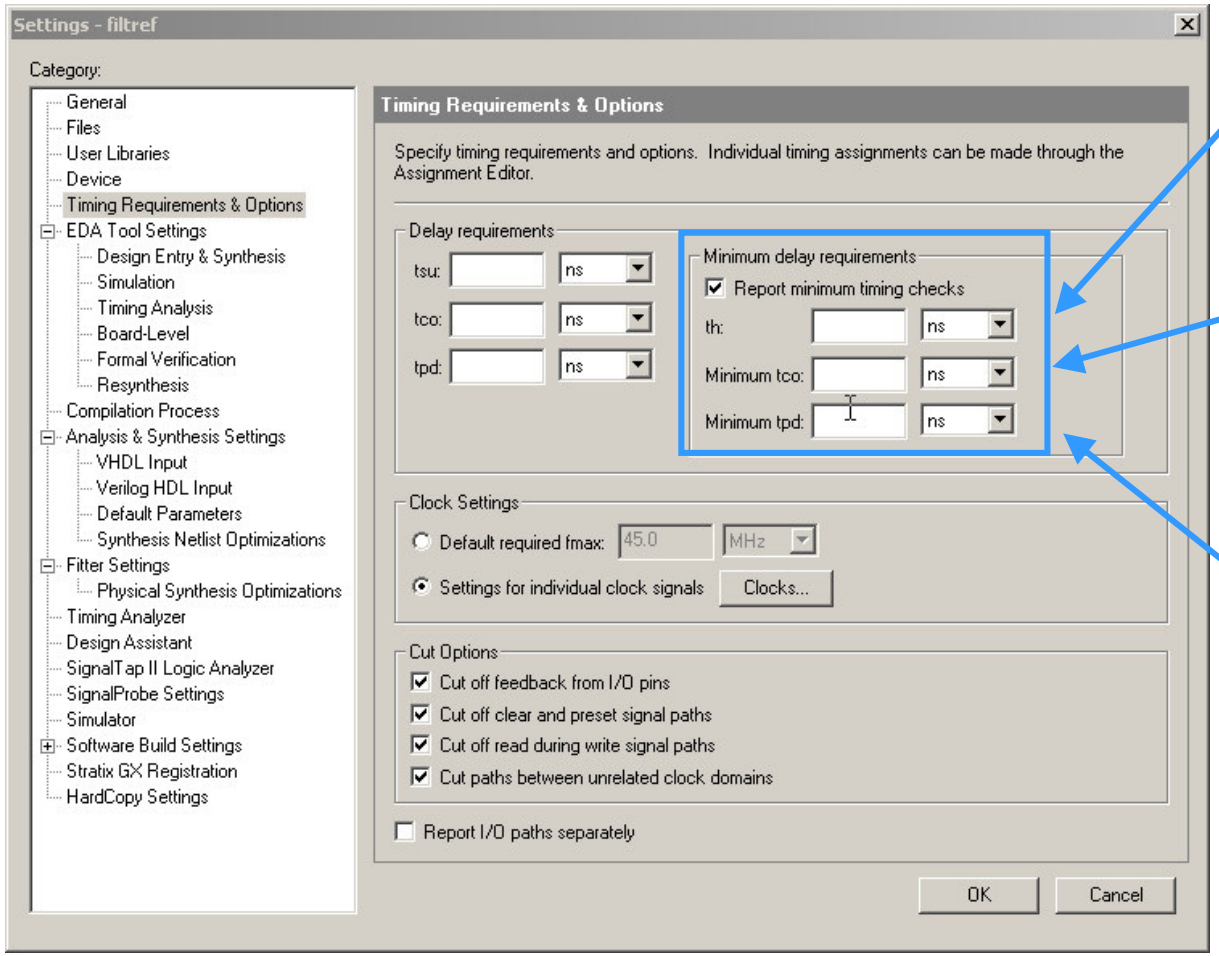
Global tpd
All Direct Pin-to-Pin Path Delays Must Be Less Than This Value

Assignments ⇒ Settings ⇒ Timing Requirements & Options



Global Minimum I/O Delay

Assignments ⇒ Settings ⇒ Timing Requirements & Options



Global th
All Inputs or Bidirectional Pins Driving Registers Must Have a th Less Than This Value

Global Minimum tco
All Registers Driving Outputs or Bidirectional Pins Must Have a tco Greater Than this Value

Global Minimum tpd
All Direct Pin-to-Pin Path Delays Must Be Greater Than This Value

Slack Calculations

- Timing Margin Comparing Actual Timing to Timing Requirements
- Appear Only When Timing Assignments Are Made

Slack = Required Time – Actual Time

Positive Slack

– Timing Requirement Met **(BLACK)**

Negative Slack

– Timing Requirement Not Met **(RED)**

Global Timing Assignments Examples

Clock Setup: 'clk1'

	Slack	Actual fmax (period)	From	To	From Clock	To Clock	Required Setup Relationsh
1	1.165 ns	304.97 MHz (period = 3.279 ns)	mult:inst|lpm_mult:...	ram:inst2|altsy...	clk1	clk1	4.444 ns
2	1.205 ns	308.74 MHz (period = 3.239 ns)	mult:inst|lpm_mult:...	ram:inst2|altsy...	clk1	clk1	4.444 ns
3	1.215 ns	309.69 MHz (period = 3.229 ns)	mult:inst|lpm_mult:...	ram:inst2|altsy...	clk1	clk1	
4	1.247 ns	312.79 MHz (period = 3.197 ns)	mult:inst|lpm_mult:...	ram:inst2|altsy...	clk1	clk1	
5	1.253 ns	313.38 MHz (period = 3.191 ns)	mult:inst|lpm_mult:...	ram:inst2|altsy...	clk1	clk1	
6	1.266 ns	314.66 MHz (period = 3.178 ns)	mult:inst|lpm_mult:...	ram:inst2|altsy...	clk1	clk1	
7	1.293 ns	317.36 MHz (period = 3.151 ns)	mult:inst|lpm_mult:...	ram:inst2|altsy...	clk1	clk1	4.444 ns
8	1.300 ns	318.07 MHz (period = 3.144 ns)	mult:inst|lpm_mult:...	ram:inst2|altsy...	clk1	clk1	4.444 ns

fmax Timing Assignment Values Are BLACK, Because Actual fmax Exceeds the Required fmax

tsu

	Slack	Required tsu	Actual tsu	From	To	To Clock
73	-0.003 ns	0.200 ns	0.203 ns	dataa[1]	mult:inst|lpm_mult:lpm_mult_component|mult_9up:au...	clk1
74	-0.002 ns	0.200 ns	0.202 ns	dataa[5]	mult:inst|lpm_mult:lpm_mult_component|mult_9up:au...	clk1
75	-0.001 ns	0.200 ns	0.201 ns	dataa[1]	mult:inst|lpm_mult:lpm_mult_component|mult_9up:au...	clk1
76	0.001 ns	0.200 ns	0.199 ns	dataa[4]	mult:inst|lpm_mult:lpm_mult_component|mult_9up:au...	clk1
77	0.002 ns	0.200 ns	0.198 ns	dataa[2]	mult:inst|lpm_mult:lpm_mult_component|	
78	0.004 ns	0.200 ns	0.196 ns	datab[3]	mult:inst|lpm_mult:lpm_mult_component|	
79	0.007 ns	0.200 ns	0.193 ns	datab[6]	mult:inst|lpm_mult:lpm_mult_component|	
80	0.009 ns	0.200 ns	0.191 ns	dataa[1]	mult:inst|lpm_mult:lpm_mult_component|	
81	0.009 ns	0.200 ns	0.191 ns	dataa[2]	mult:inst|lpm_mult:lpm_mult_component|	
82	0.009 ns	0.200 ns	0.191 ns	datab[5]	mult:inst|lpm_mult:lpm_mult_component|mult_9up:au...	clk1

t_{SU} timing assignment Values Are RED Because Actual t_{SU} Falls below Required t_{SU}

Individual Timing Assignments

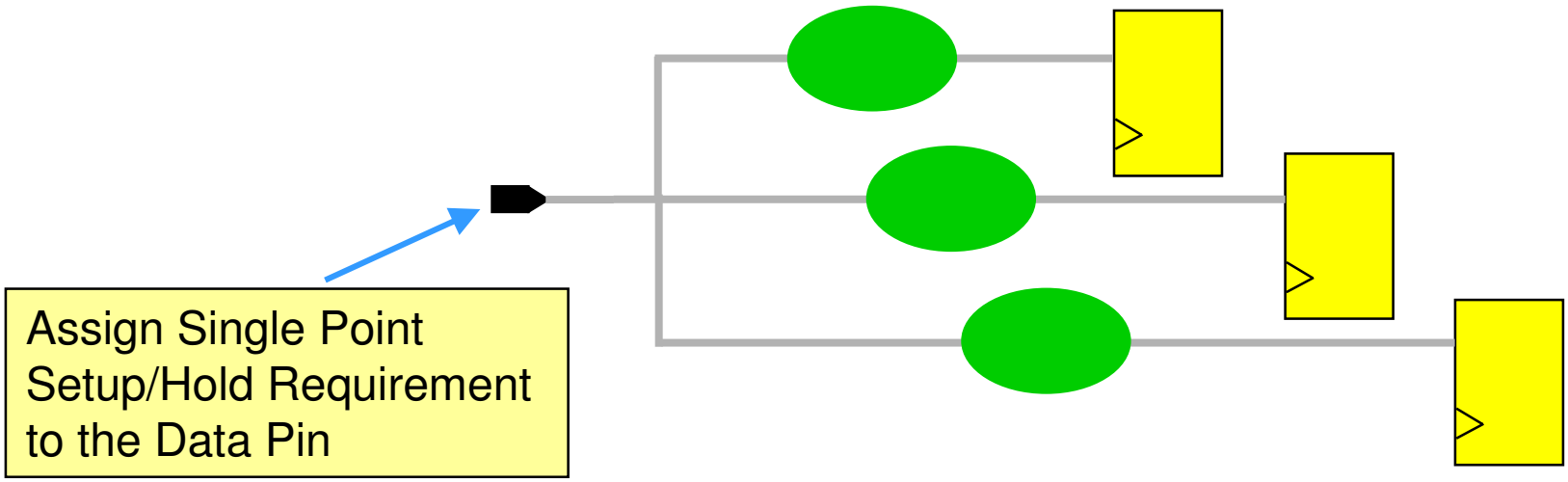
- Available Settings
  - tsu (Max)
  - th (Max)
  - tco (Max & Min)
  - tpd (Max & Min)
    - Pin-to-Pin
    - Point-to-Point (I/O→Reg; Reg→Reg; Reg→I/O)
- Available Assignment Types
  - Single-Point
  - Point-to-Point
  - Wildcard (* or ?)

Individual Assignment Targets

- Registers (all)
- Clock Pins (tsu, tco, th)
- Input Pins (tsu, th, tpd)
- Output Pins (tco, tpd)
- Bi-Directional Pins (all)

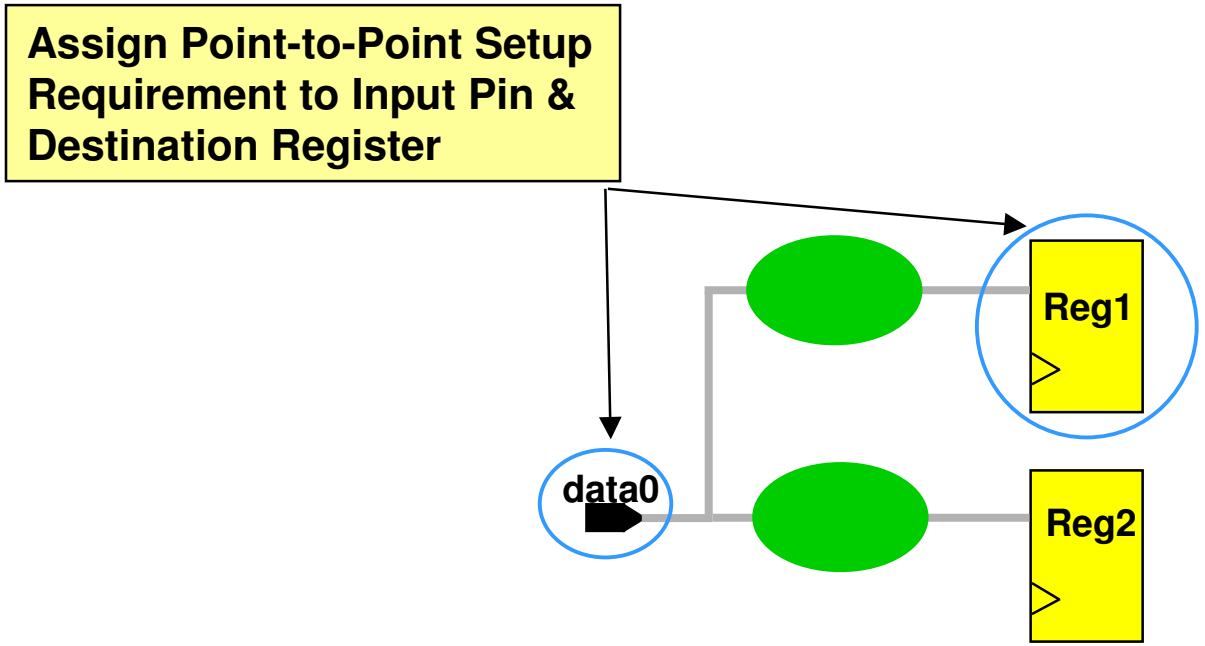
Single-Point tsu/th

- Setup/Hold Timing Required on Every Register Fed by Pin



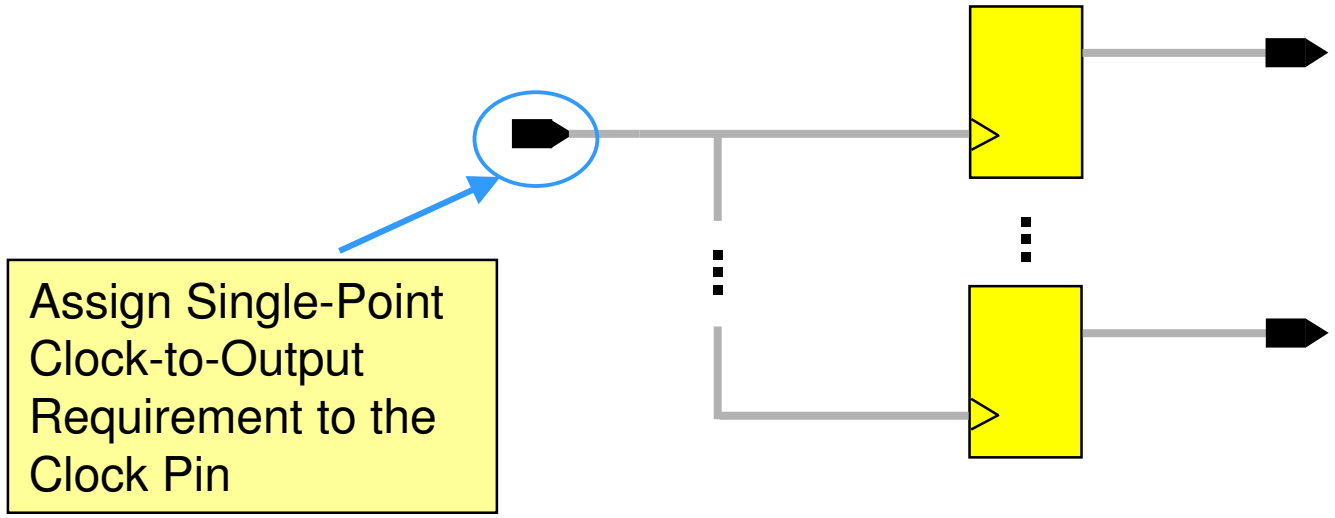
Point-to-Point tsu/th

- Setup/Hold Timing Required Only on the Specified Path



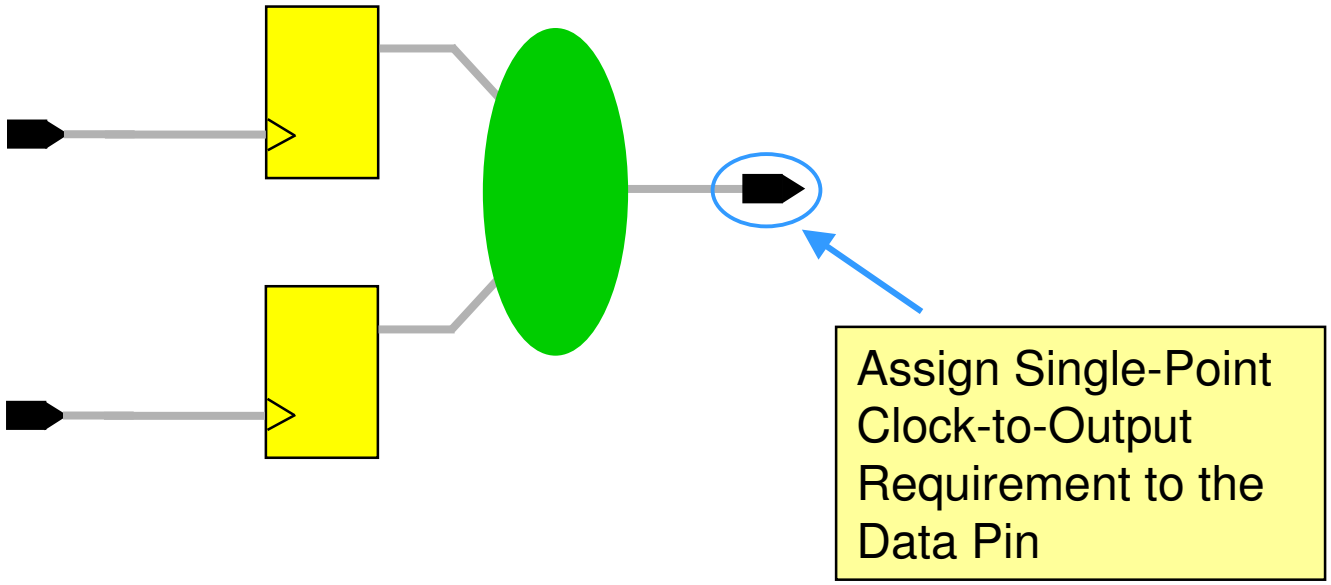
Single-Point tco

- Clock-to-Out Timing Required on All tco Paths Starting from the Specified Clock



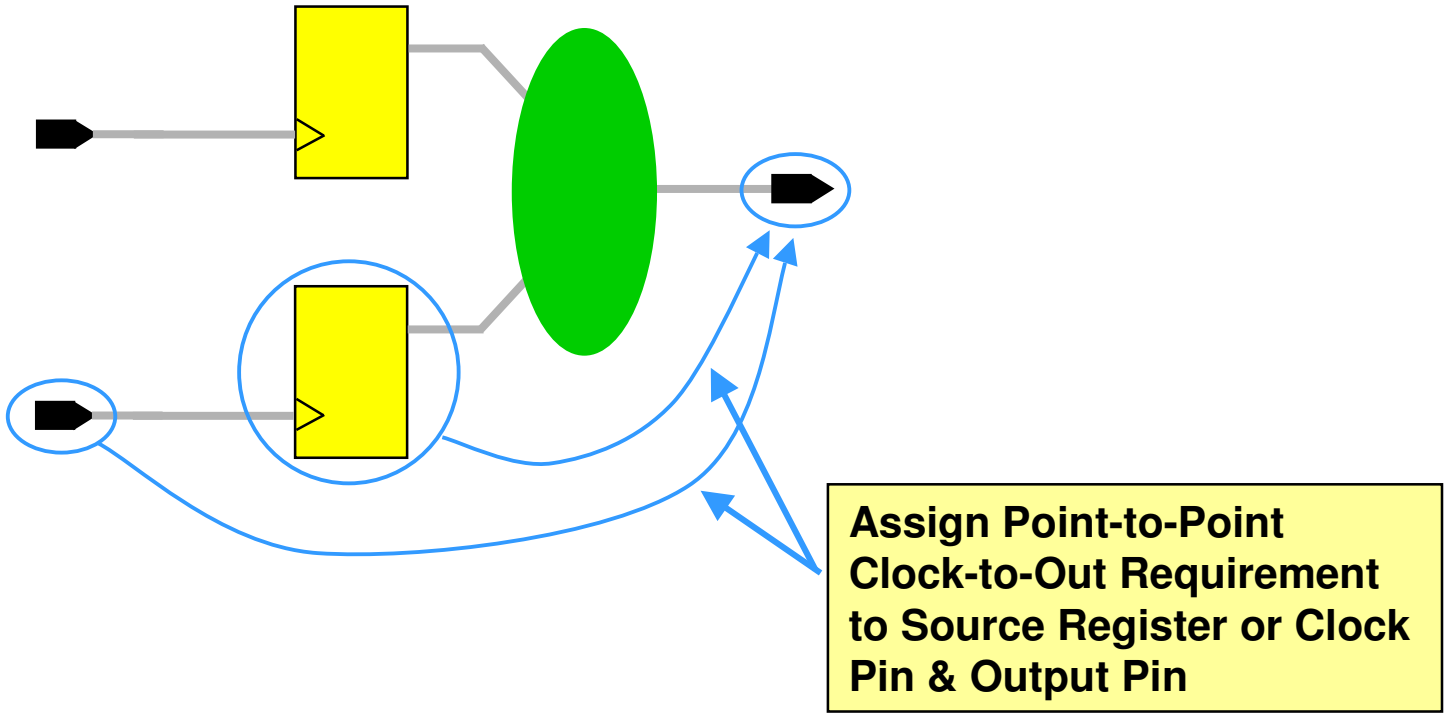
Single-Point tco (cont.)

- Clock-to-Out Timing Required on All Paths from Any Clock to Specified Output Pin



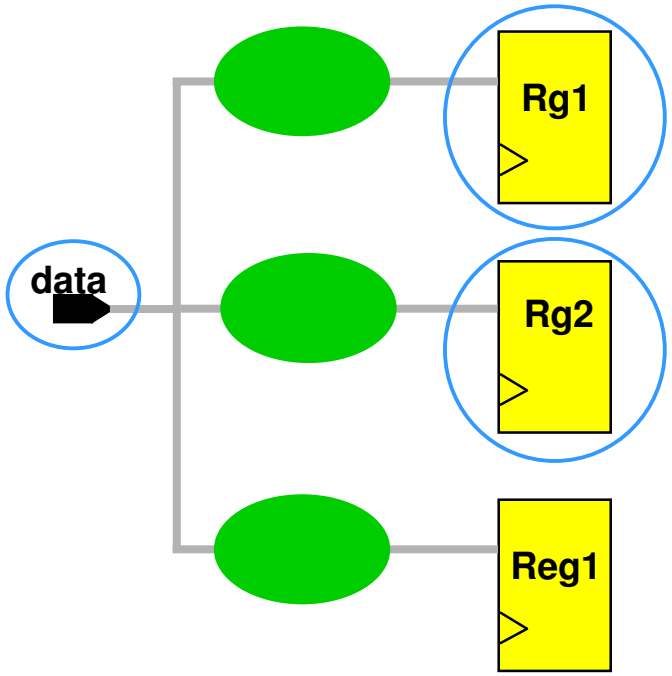
Point-to-Point tco

- Clock-to-Out Timing Required Only on the Specified Path



Wildcard Assignment

- Indicates All Targets with a Character or String
  - '*' - Zero or More Characters
  - '?' – Single Character

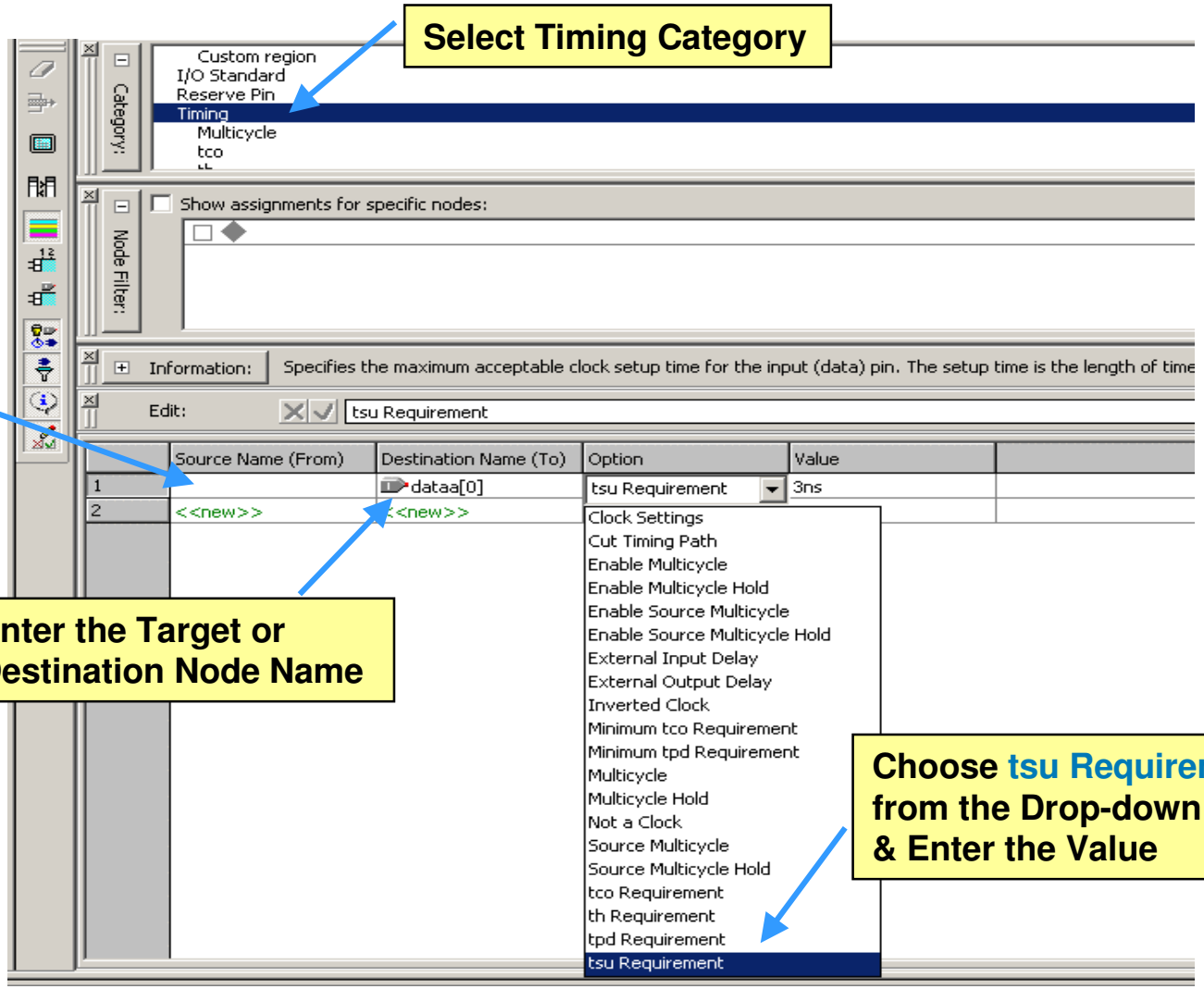


Point-to-Point Setup Using Wildcard:

Assign Setup Requirement from "data" to "Rg*" or "Rg?"

Ex. Assigning Setup Requirement

Assignment Editor Is Used for All Individual Timing Requirements



Use Source Name (From) to Create a Point-to-Point Requirement

Enter the Target or Destination Node Name

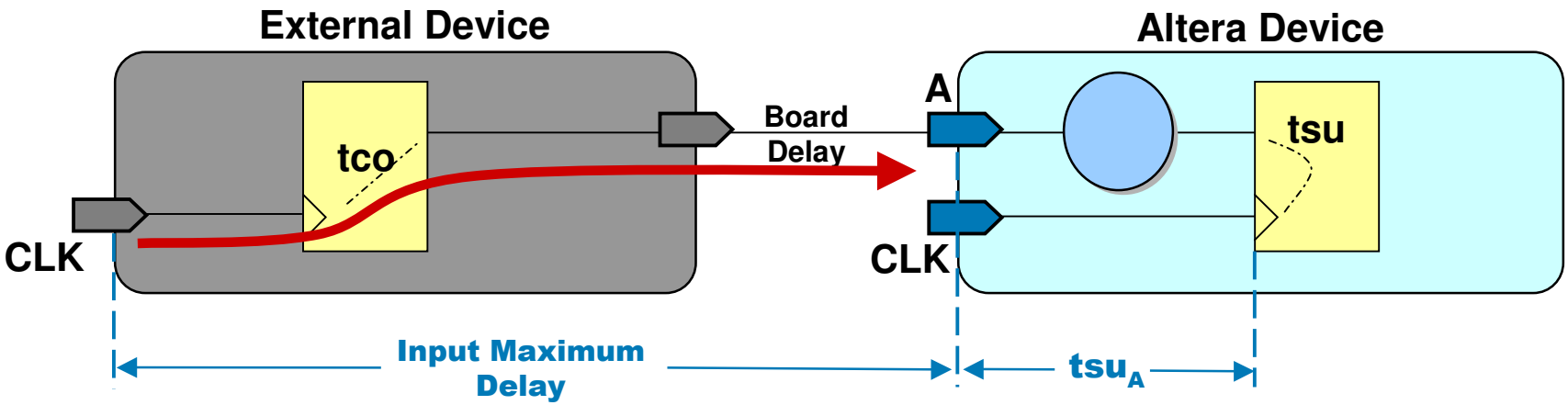
Choose tsu Requirement from the Drop-down List & Enter the Value

External Delay Assignments

- Specify System-Level Timing Constraints
- Constrain I/O Timing
  - Same as tsu, th, tco & Minimum tco
- Include I/O Timing as Part of Clock Timing Analysis Report
  - Clock Setup(fmax)
  - Clock Hold
- Settings
  - Input Minimum/Maximum Delay
  - Output Minimum/Maximum Delay

Input Maximum Delay

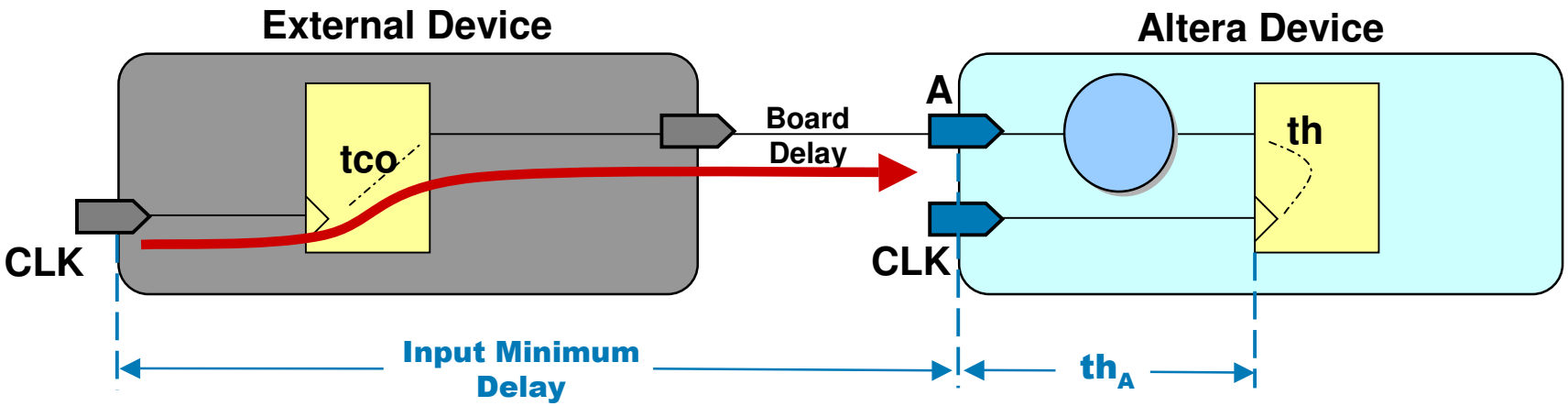
- Maximum Delay from External Device to Altera I/O
  - Represents External Device tco + Board Delay
- Constrains Input Pin tsu



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

Input Minimum Delay

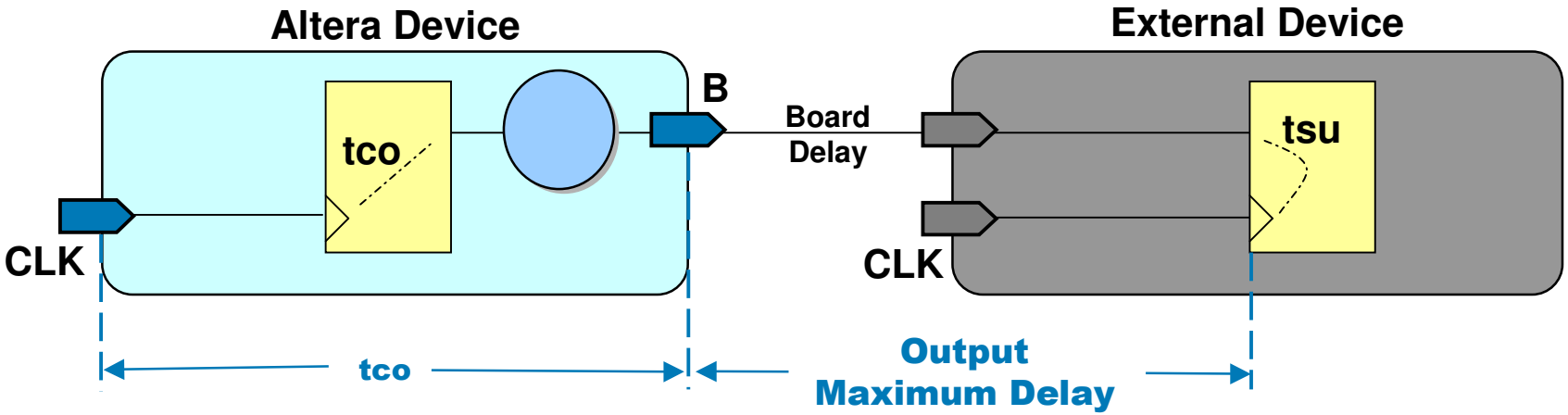
- Minimum Delay from External Device to Altera I/O
  - Represents External Device tco + Board Delay
- Constrains Input Pin th



$th_A \leq$ *Input Minimum Delay*

Output Maximum Delay

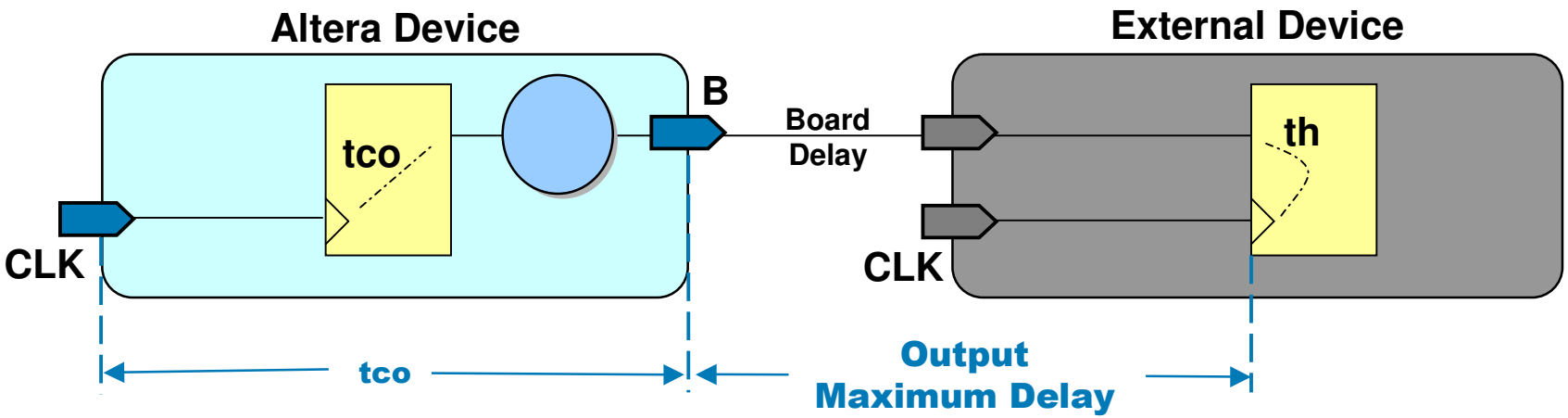
- Maximum Delay from Altera I/O to External Device
  - Represents External Device tsu + Board Delay
- Constrains Output Pin tco



$tco_B \leq t_{CLK}$ - *Output Maximum Delay*

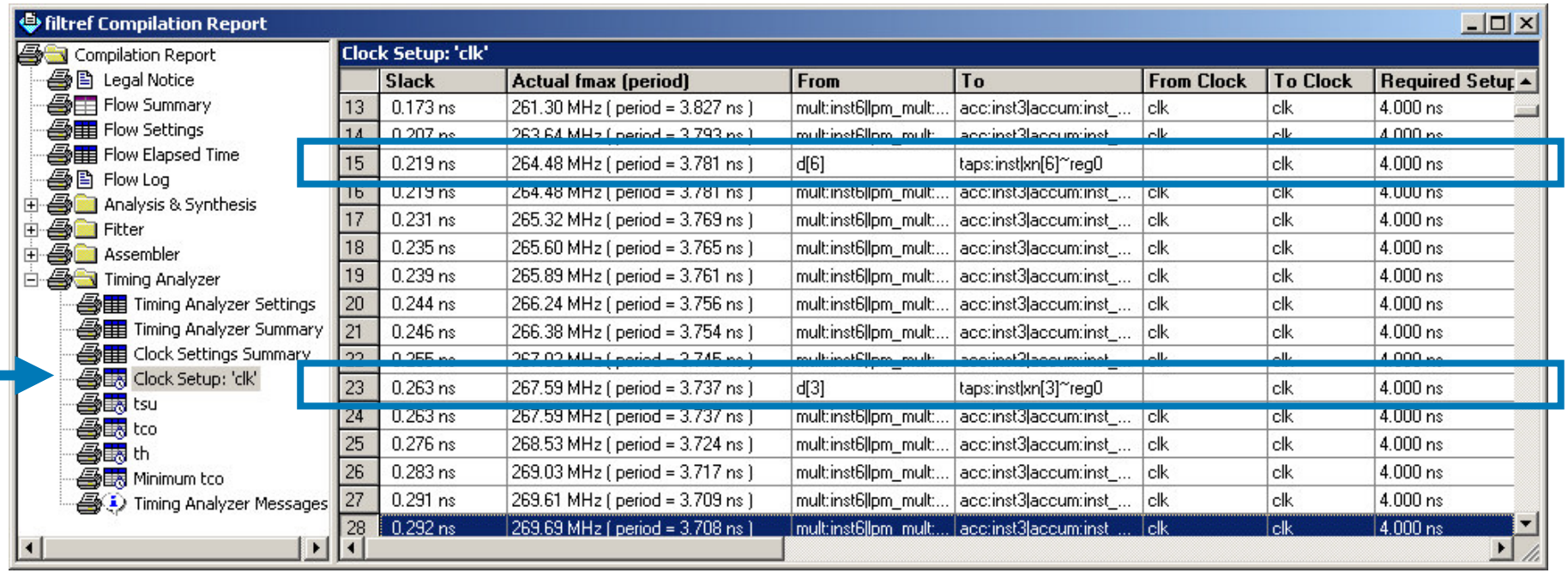
Output Minimum Delay

- Minimum Delay from Altera I/O to External Device
  - Represents External Device th - Board Delay
- Constrains Output Pin Minimum tco



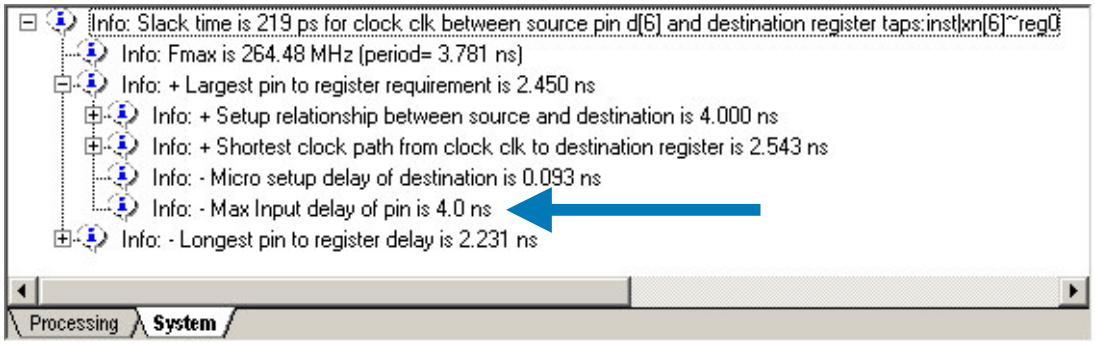
$tco_B \geq$ *Output Minimum Delay*

Example Input Maximum Delay



	Slack	Actual fmax (period)	From	To	From Clock	To Clock	Required Setup
13	0.173 ns	261.30 MHz (period = 3.827 ns)	mult:inst6|lpm_mult:...	acc:inst3|accum:inst_...	clk	clk	4.000 ns
15	0.219 ns	264.48 MHz (period = 3.781 ns)	d[6]	taps:inst|xn[6]~reg0		clk	4.000 ns
17	0.231 ns	265.32 MHz (period = 3.769 ns)	mult:inst6|lpm_mult:...	acc:inst3|accum:inst_...	clk	clk	4.000 ns
18	0.235 ns	265.60 MHz (period = 3.765 ns)	mult:inst6|lpm_mult:...	acc:inst3|accum:inst_...	clk	clk	4.000 ns
19	0.239 ns	265.89 MHz (period = 3.761 ns)	mult:inst6|lpm_mult:...	acc:inst3|accum:inst_...	clk	clk	4.000 ns
20	0.244 ns	266.24 MHz (period = 3.756 ns)	mult:inst6|lpm_mult:...	acc:inst3|accum:inst_...	clk	clk	4.000 ns
21	0.246 ns	266.38 MHz (period = 3.754 ns)	mult:inst6|lpm_mult:...	acc:inst3|accum:inst_...	clk	clk	4.000 ns
23	0.263 ns	267.59 MHz (period = 3.737 ns)	d[3]	taps:inst|xn[3]~reg0		clk	4.000 ns
25	0.276 ns	268.53 MHz (period = 3.724 ns)	mult:inst6|lpm_mult:...	acc:inst3|accum:inst_...	clk	clk	4.000 ns
26	0.283 ns	269.03 MHz (period = 3.717 ns)	mult:inst6|lpm_mult:...	acc:inst3|accum:inst_...	clk	clk	4.000 ns
27	0.291 ns	269.61 MHz (period = 3.709 ns)	mult:inst6|lpm_mult:...	acc:inst3|accum:inst_...	clk	clk	4.000 ns
28	0.292 ns	269.69 MHz (period = 3.708 ns)	mult:inst6|lpm_mult:...	acc:inst3|accum:inst_...	clk	clk	4.000 ns

Input Maximum Delay (d) = 4 ns

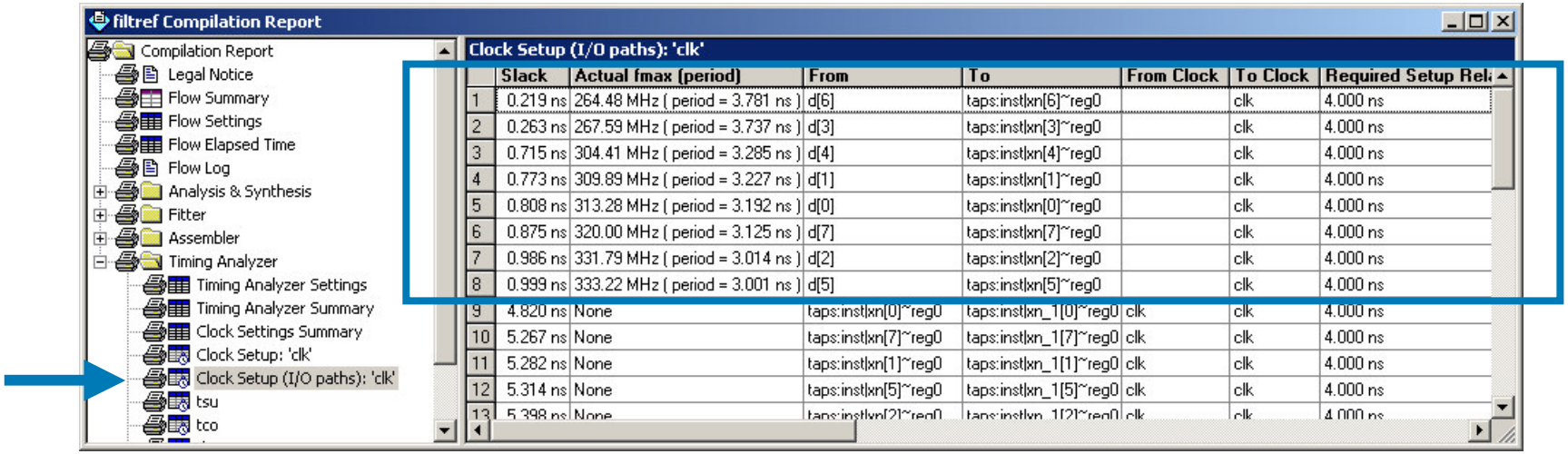


Notice:

1) **Input Pin d(6) & d(3) Timing Information Is Included with Clock Setup (fmax) Analysis**
2) **Input Delay Has Been Added to List Path Calculation**

Clock Setup/Hold (I/O Paths)

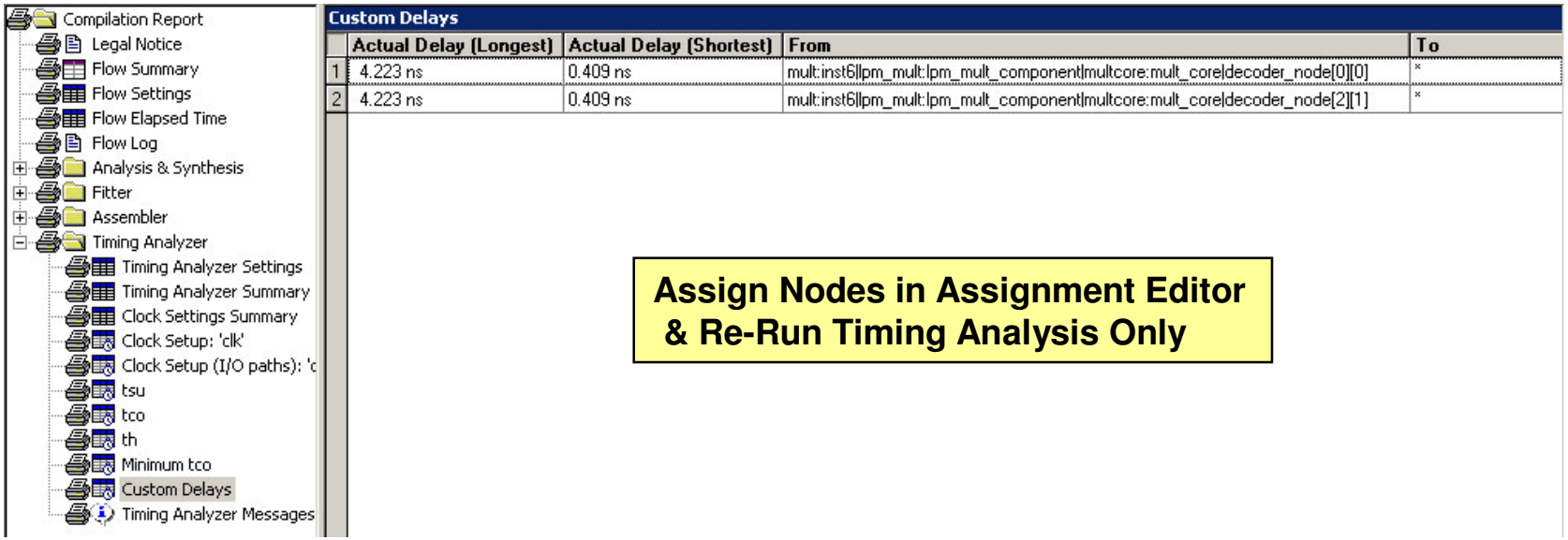
- External (System-Level) I/O Timing Reported Separately in Timing Analyzer
  - Option Found in **Assignments ⇒ Settings ⇒ Timing Requirements & Options**



Other Individual Timing Assignments

- Cut Timing Path
  - Removes Paths from TDC & Timing Analysis
  - Ex. Test Logic
- Not a Clock
  - Disables Clock Identification
  - Ex. Gate Inputs to Gated Clocks
- Inverted Clock
  - Marks Inverted Clocks Generated with Complex Logic

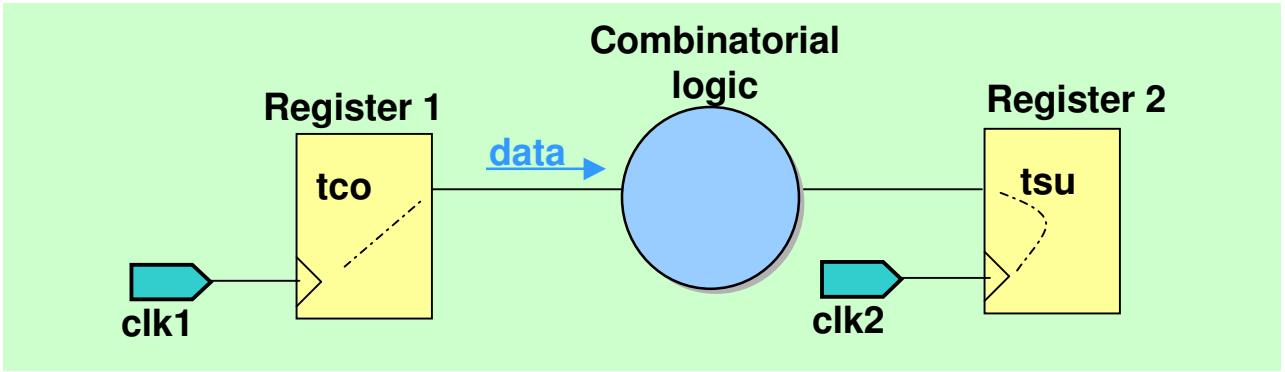
Report Delay Assignment

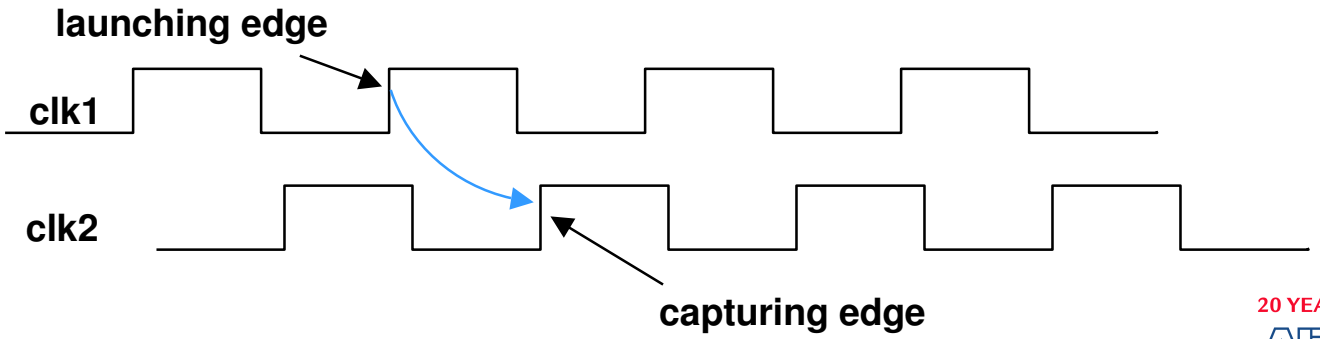


- Reports Delay between Selected Nodes
  - Pin-Register
  - Register-Register
  - Register-Pin
- Reports Results in Custom Delay Section of Timing Analyzer

Multi-Clock Frequency Analysis

- Analyzes Timing on Register-to-Register Paths Controlled by Different Synchronous Clocks
  - Individual Clocks Treated as Same Frequency & Phase Unless Specified

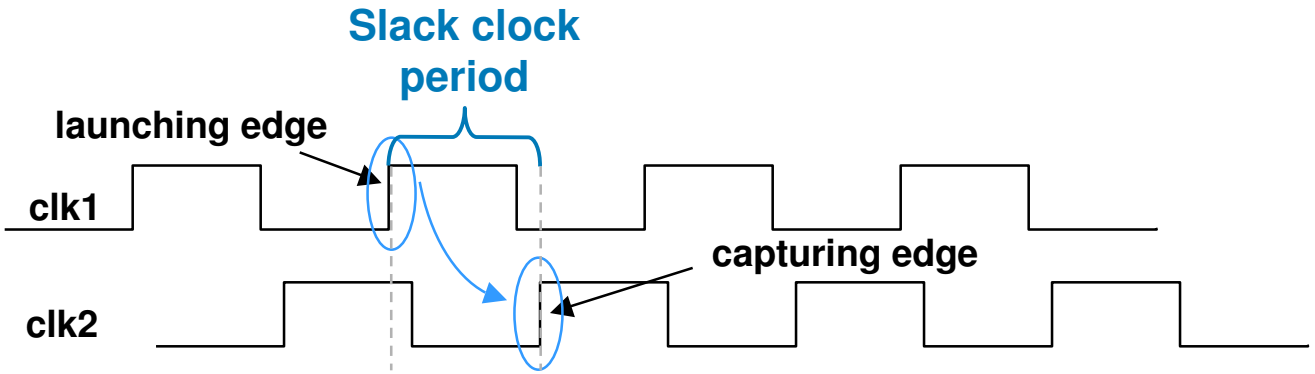


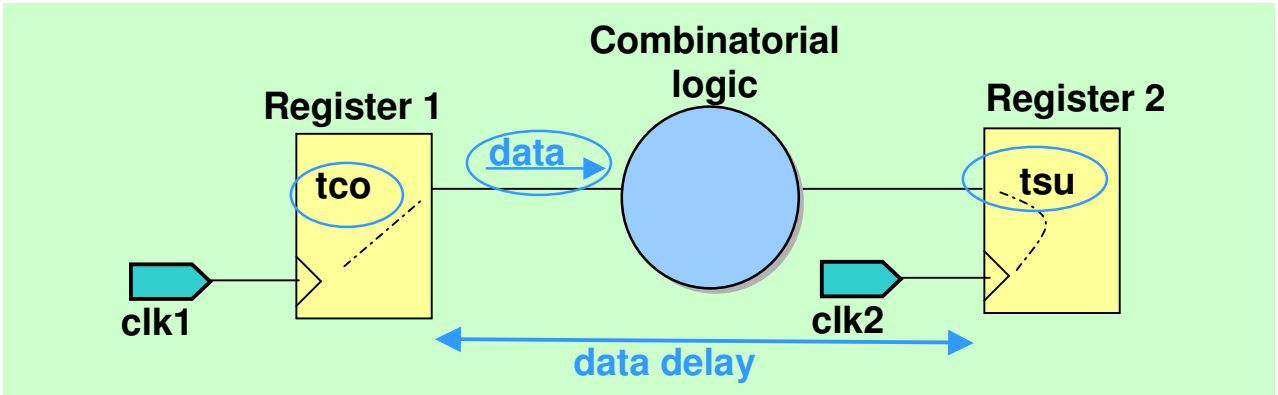


Slack Equations for Multi-Clock

Slack = Required Time - Actual Time

Slack = Slack Clock Period - (tco + Data Delay + tsu)





Steps for Multi-Clock Analysis

1) Create Clock Settings & Assign to Clock Node
   - Define a Base Clock
   - Define Clock(s) Relative to the Base Clock
2) Recompile or Restart Timing Analysis
3) Examine Results in Timing Analyzer Report

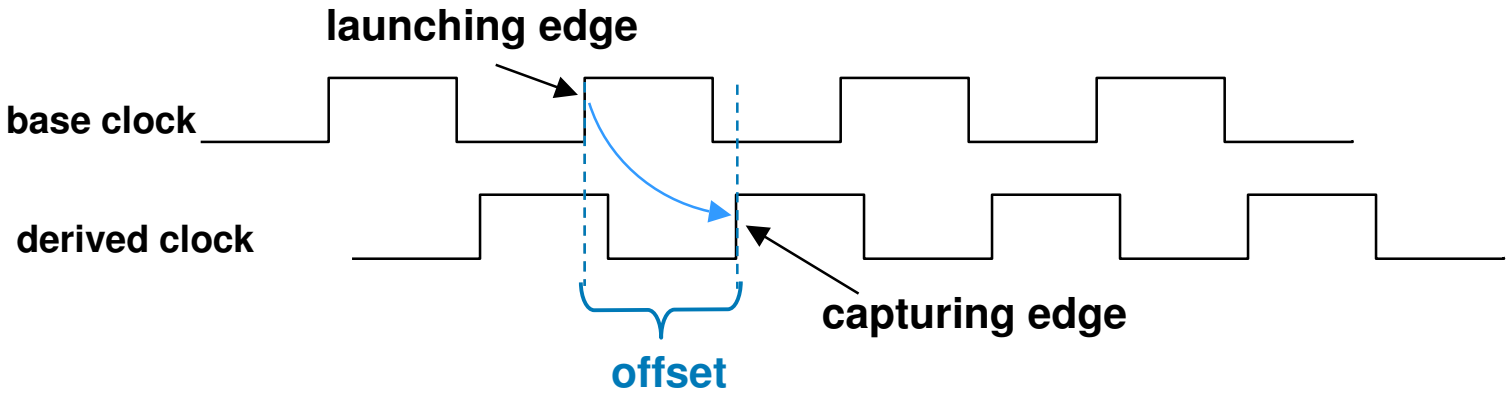
Note: May Also Use Timing Wizard

1) Create Clock Settings

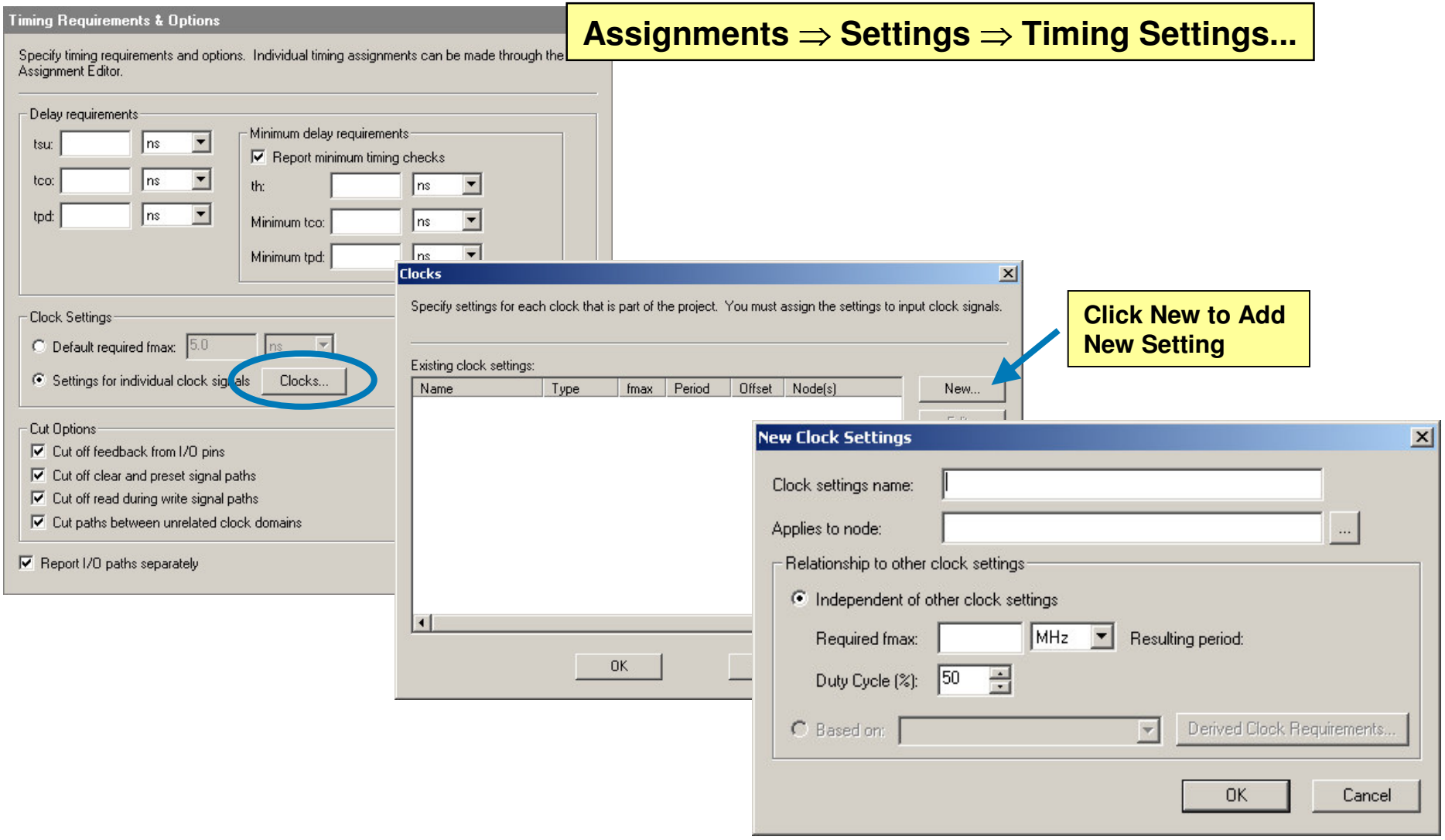
- Set Clock Relationships
  - Assign Base Clock
  - Assign Derived Clock(s) Referenced to Base

derived clock = base clock x (ratio) + offset

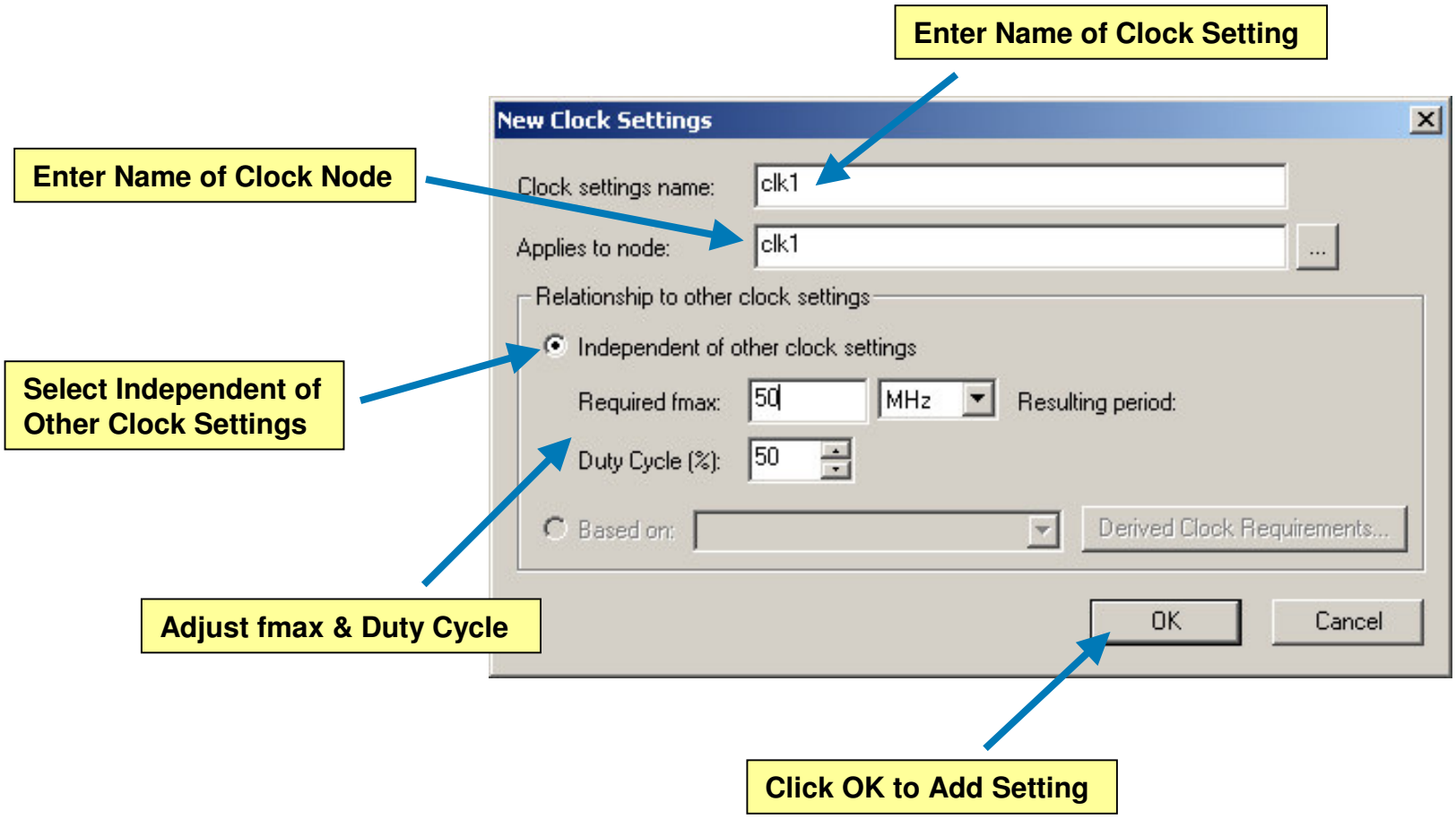
Note: Multiple Base & Derived Clocks Allowed



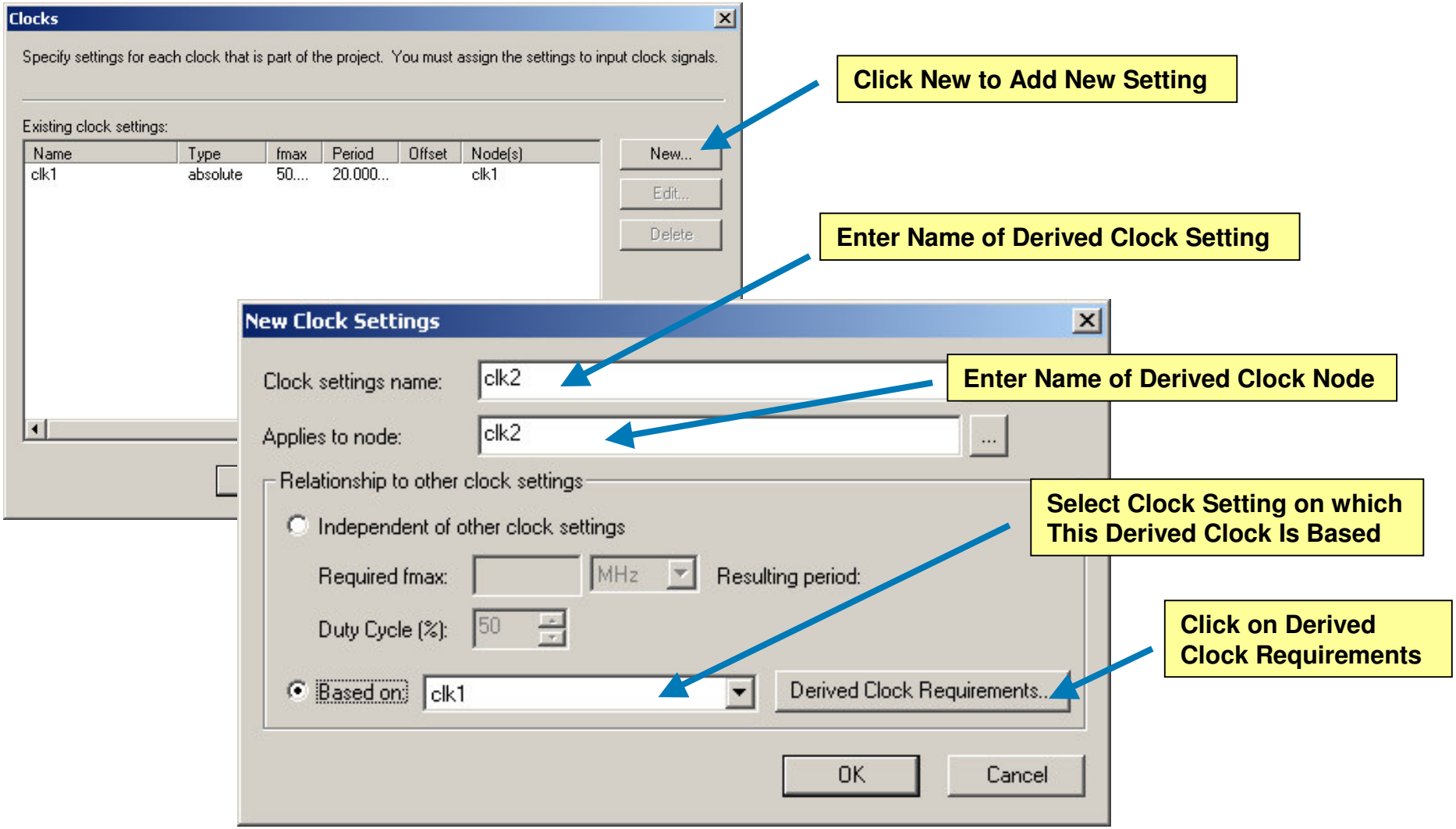
Base Clocks



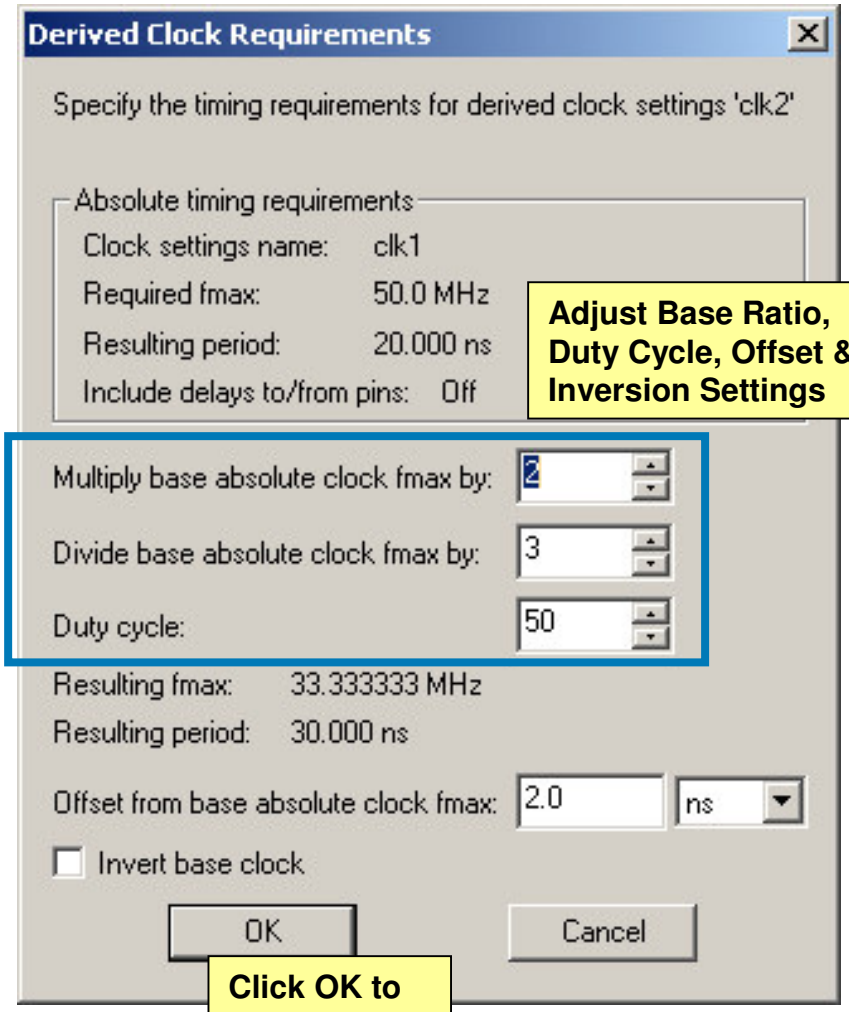
Base Clocks (cont.)



Derived Clocks

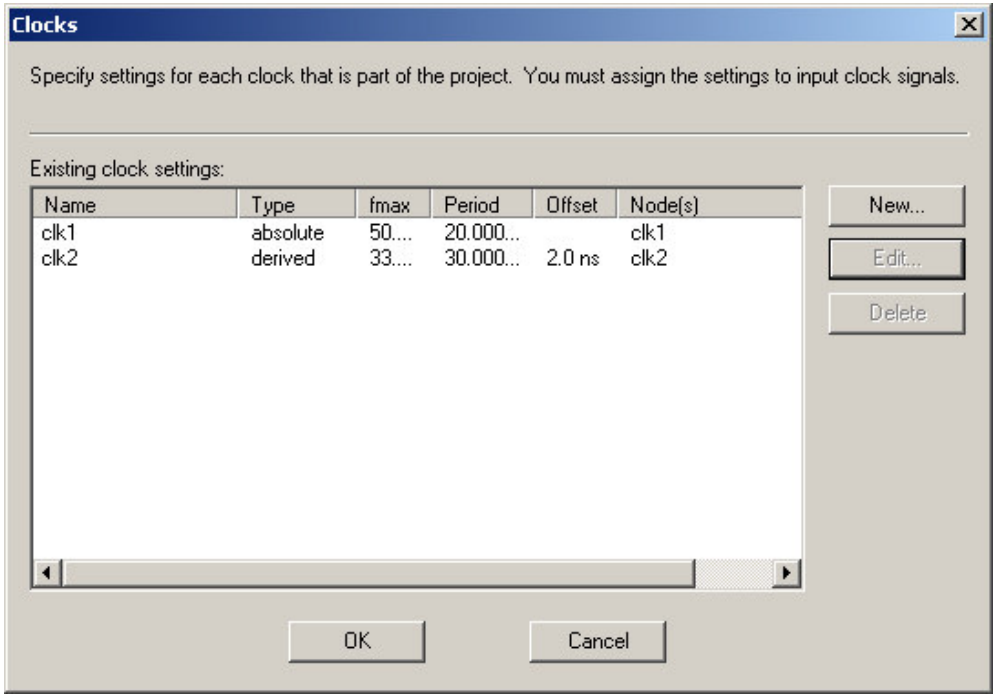


Derived Clocks (cont.)



Adjust Base Ratio, Duty Cycle, Offset & Inversion Settings

Click OK to Add Setting

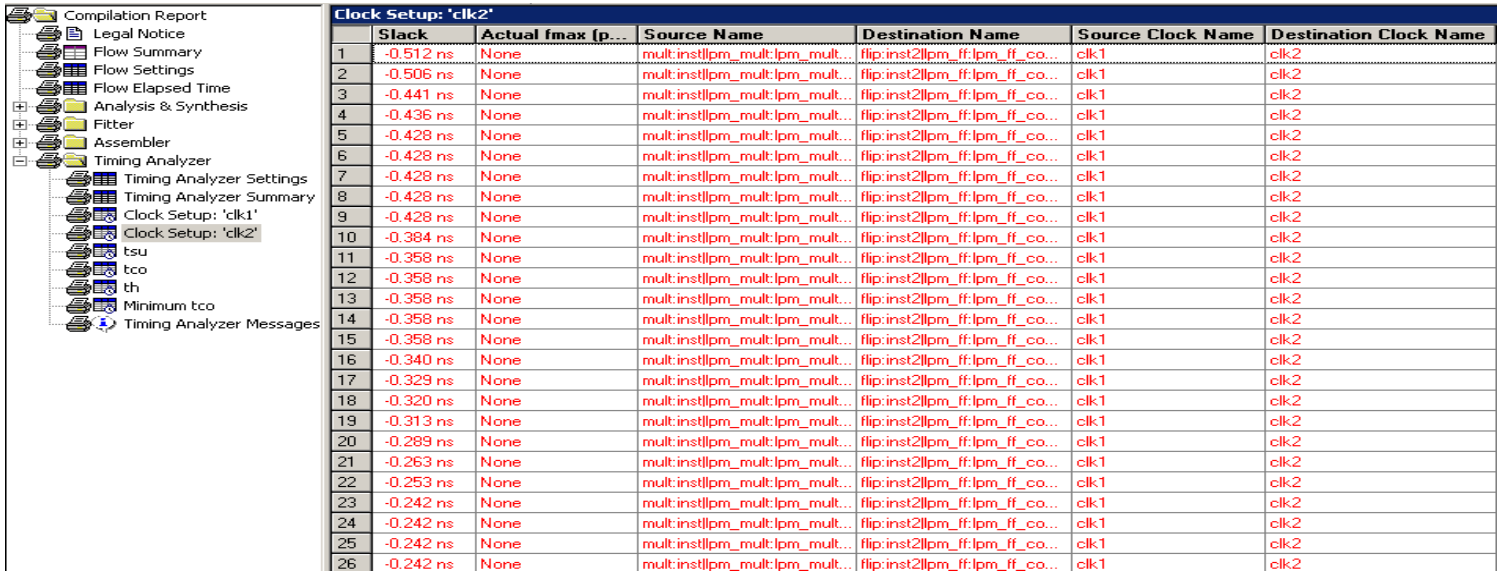


20 YEARS of ALTERA INNOVATION

3) Examine Timing Analyzer Report

- Positive Slack - Timing Was Met (BLACK)
- Negative Slack - By How Much Timing Was Not Met (RED)

- Must Alter Design or Use Timing Assignments (Multi-Cycle) to Resolve Negative Slack

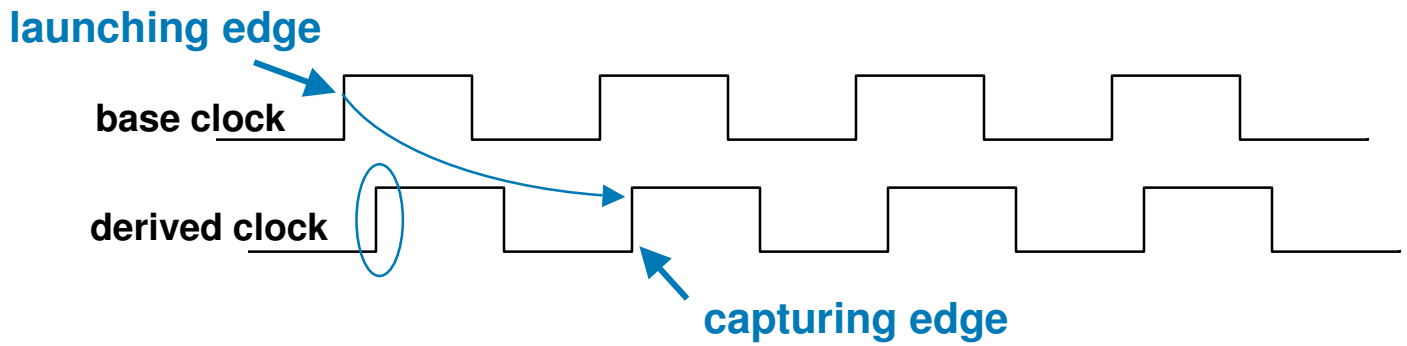


Clock Setup: 'clk2'

	Slack	Actual fmax (p...	Source Name	Destination Name	Source Clock Name	Destination Clock Name
1	-0.512 ns	None	mult:inst|lpm_mult:lpm_mult...	flip:inst2|lpm_ff:lpm_ff_co...	clk1	clk2
2	-0.506 ns	None	mult:inst|lpm_mult:lpm_mult...	flip:inst2|lpm_ff:lpm_ff_co...	clk1	clk2
3	-0.441 ns	None	mult:inst|lpm_mult:lpm_mult...	flip:inst2|lpm_ff:lpm_ff_co...	clk1	clk2
4	-0.436 ns	None	mult:inst|lpm_mult:lpm_mult...	flip:inst2|lpm_ff:lpm_ff_co...	clk1	clk2
5	-0.428 ns	None	mult:inst|lpm_mult:lpm_mult...	flip:inst2|lpm_ff:lpm_ff_co...	clk1	clk2
6	-0.428 ns	None	mult:inst|lpm_mult:lpm_mult...	flip:inst2|lpm_ff:lpm_ff_co...	clk1	clk2
7	-0.428 ns	None	mult:inst|lpm_mult:lpm_mult...	flip:inst2|lpm_ff:lpm_ff_co...	clk1	clk2
8	-0.428 ns	None	mult:inst|lpm_mult:lpm_mult...	flip:inst2|lpm_ff:lpm_ff_co...	clk1	clk2
9	-0.428 ns	None	mult:inst|lpm_mult:lpm_mult...	flip:inst2|lpm_ff:lpm_ff_co...	clk1	clk2
10	-0.384 ns	None	mult:inst|lpm_mult:lpm_mult...	flip:inst2|lpm_ff:lpm_ff_co...	clk1	clk2
11	-0.358 ns	None	mult:inst|lpm_mult:lpm_mult...	flip:inst2|lpm_ff:lpm_ff_co...	clk1	clk2
12	-0.358 ns	None	mult:inst|lpm_mult:lpm_mult...	flip:inst2|lpm_ff:lpm_ff_co...	clk1	clk2
13	-0.358 ns	None	mult:inst|lpm_mult:lpm_mult...	flip:inst2|lpm_ff:lpm_ff_co...	clk1	clk2
14	-0.358 ns	None	mult:inst|lpm_mult:lpm_mult...	flip:inst2|lpm_ff:lpm_ff_co...	clk1	clk2
15	-0.358 ns	None	mult:inst|lpm_mult:lpm_mult...	flip:inst2|lpm_ff:lpm_ff_co...	clk1	clk2
16	-0.340 ns	None	mult:inst|lpm_mult:lpm_mult...	flip:inst2|lpm_ff:lpm_ff_co...	clk1	clk2
17	-0.329 ns	None	mult:inst|lpm_mult:lpm_mult...	flip:inst2|lpm_ff:lpm_ff_co...	clk1	clk2
18	-0.320 ns	None	mult:inst|lpm_mult:lpm_mult...	flip:inst2|lpm_ff:lpm_ff_co...	clk1	clk2
19	-0.313 ns	None	mult:inst|lpm_mult:lpm_mult...	flip:inst2|lpm_ff:lpm_ff_co...	clk1	clk2
20	-0.289 ns	None	mult:inst|lpm_mult:lpm_mult...	flip:inst2|lpm_ff:lpm_ff_co...	clk1	clk2
21	-0.263 ns	None	mult:inst|lpm_mult:lpm_mult...	flip:inst2|lpm_ff:lpm_ff_co...	clk1	clk2
22	-0.253 ns	None	mult:inst|lpm_mult:lpm_mult...	flip:inst2|lpm_ff:lpm_ff_co...	clk1	clk2
23	-0.242 ns	None	mult:inst|lpm_mult:lpm_mult...	flip:inst2|lpm_ff:lpm_ff_co...	clk1	clk2
24	-0.242 ns	None	mult:inst|lpm_mult:lpm_mult...	flip:inst2|lpm_ff:lpm_ff_co...	clk1	clk2
25	-0.242 ns	None	mult:inst|lpm_mult:lpm_mult...	flip:inst2|lpm_ff:lpm_ff_co...	clk1	clk2
26	-0.242 ns	None	mult:inst|lpm_mult:lpm_mult...	flip:inst2|lpm_ff:lpm_ff_co...	clk1	clk2

Multi-Cycle Paths

- Intentionally Require More Than One Clock Cycle to Become Stable
  - Must Be Considered in Design Implementation
  - Must Tell Timing Analyzer to Account for Multiple Clock Edges in Clock Setup Calculation

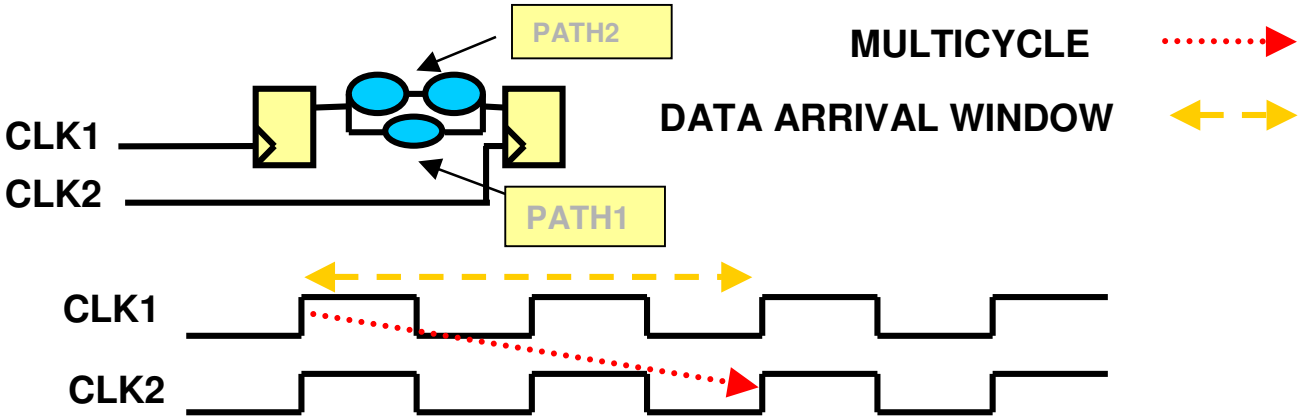


Assigning Multi-Cycle Paths

- Destination/Source Register
  - Applies to All Paths Leading from/to Register
- Register-to-Register
  - Applies to All Paths between One Source & One Destination Register
- Two Clock Domains
  - Applies to All Signals Traveling between Clock Domains
- Register Clock Enables
  - Applies to All Registers Controlled by Clock Enable

Multi-Cycle Assignment

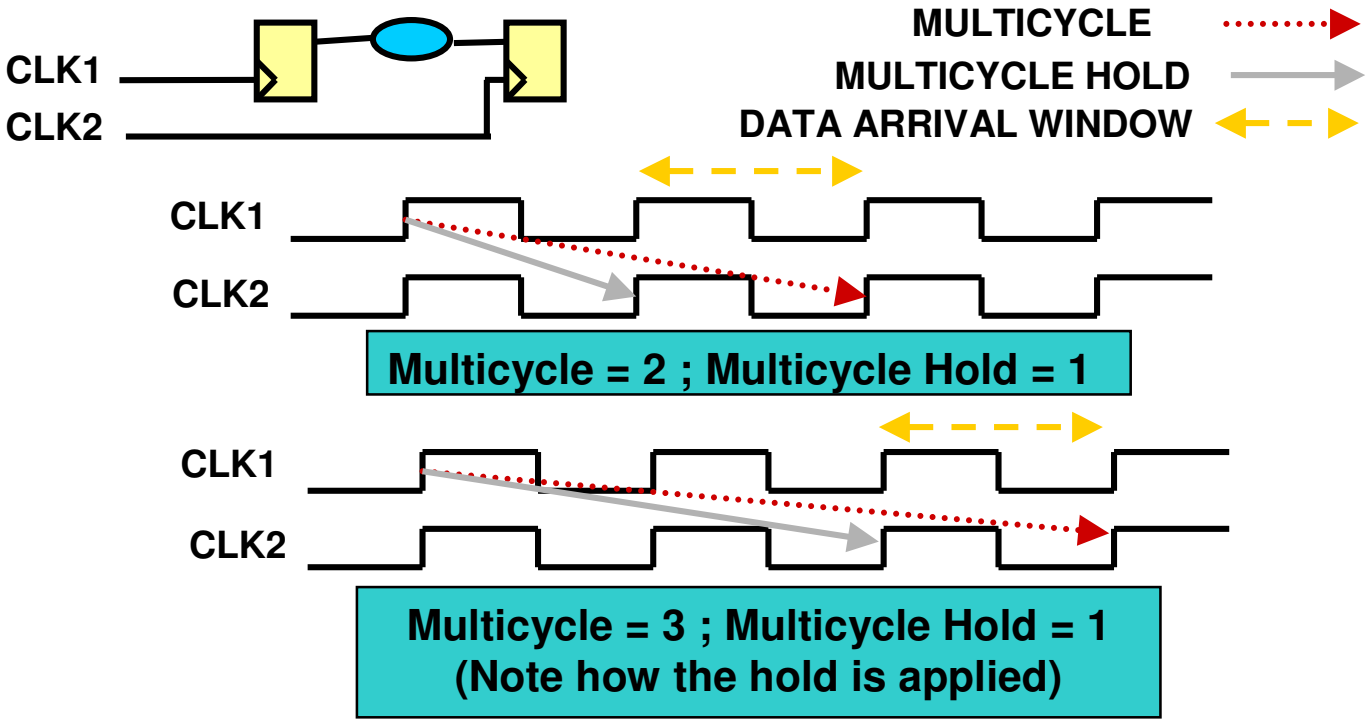
- **Maximum** Point-to-Point Tming
  - Data Cannot Arrive after Number of Cycles
  - Ex: One Path Is < **1 Cycle**, Other Path Is > **1 Cycle**
    - Circuit Requires Enables for Proper Operation



Multicycle = 2 ; Multicycle Hold = 2 (Default)

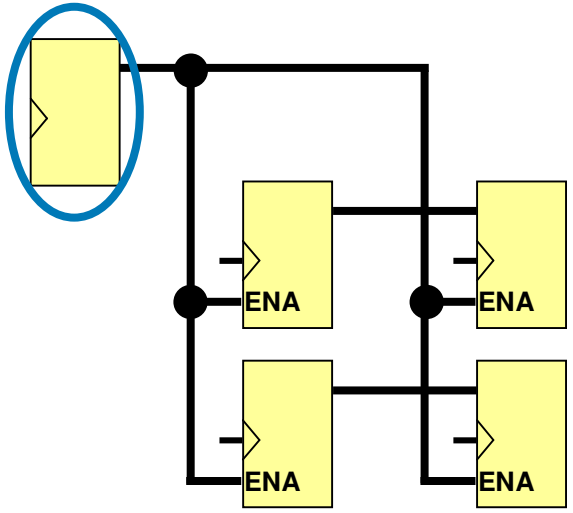
Multi-Cycle Hold Assignment

- **Minimum** Point-to-Point Timing
  - Data Must Arrive after Hold Time
  - Used in Conjunction with a Multi-cycle Assignment



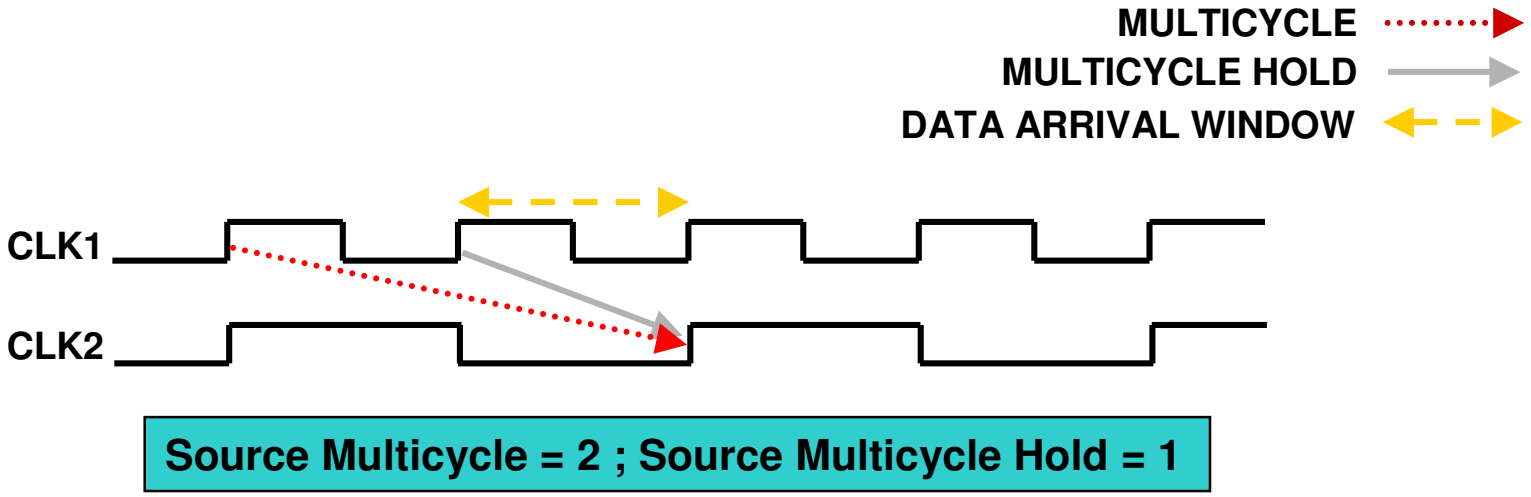
Clock Enable Multi-Cycle Assignments

- Assigns Multi-Cycle to Source of Clock Enable
  - I/O Pin
  - Register
- Settings
  - Clock Enable Multicycle (Maximum)
  - Clock Enable Multicycle Hold (Minimum)



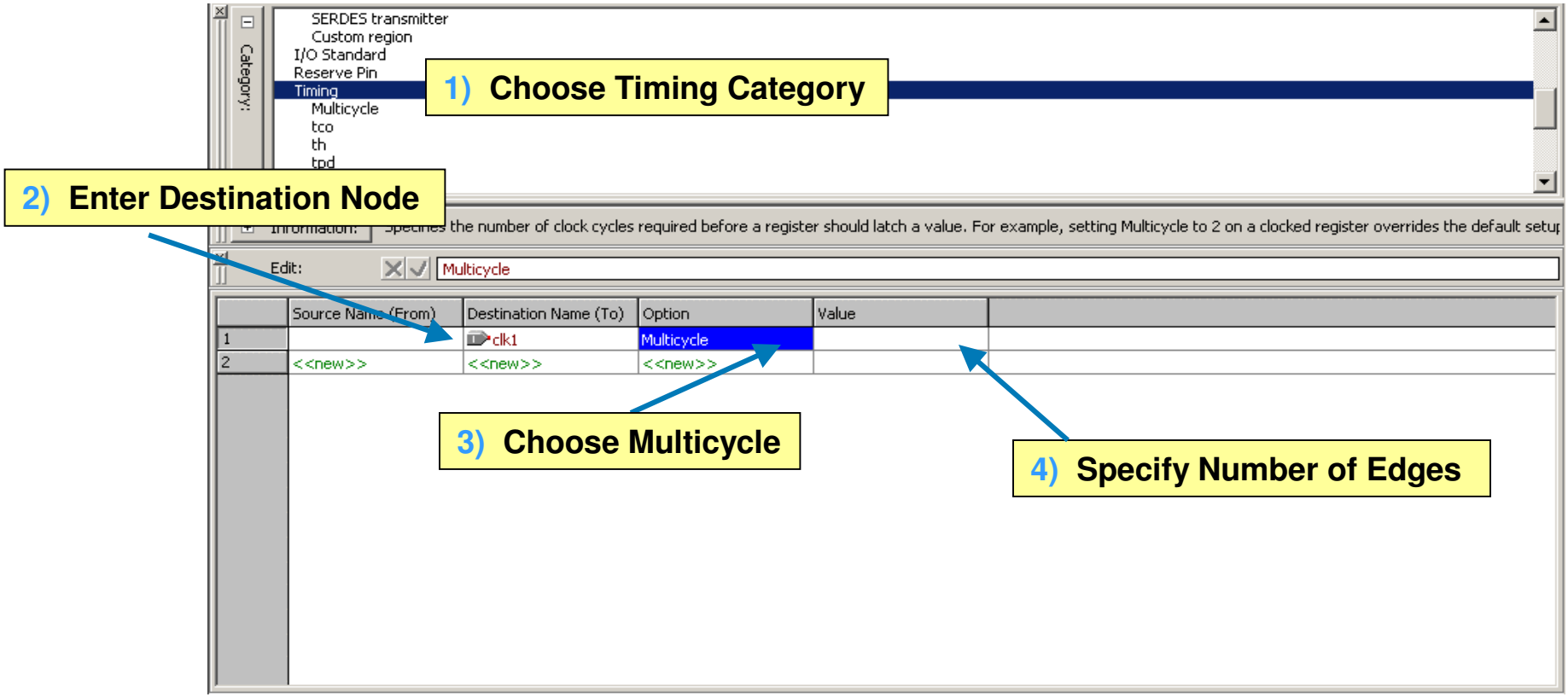
Source Mult-Cycle Assignments

- Used When Source Clock is Higher Frequency
- Settings
  - Source Multicycle & Multicycle Hold
  - Source Clock Enable Multicycle & Muticycle Hold



Ex. Assigning a Multi-Cycle

Assignments ⇒ Assignment Editor...



Minimum Timing Analysis

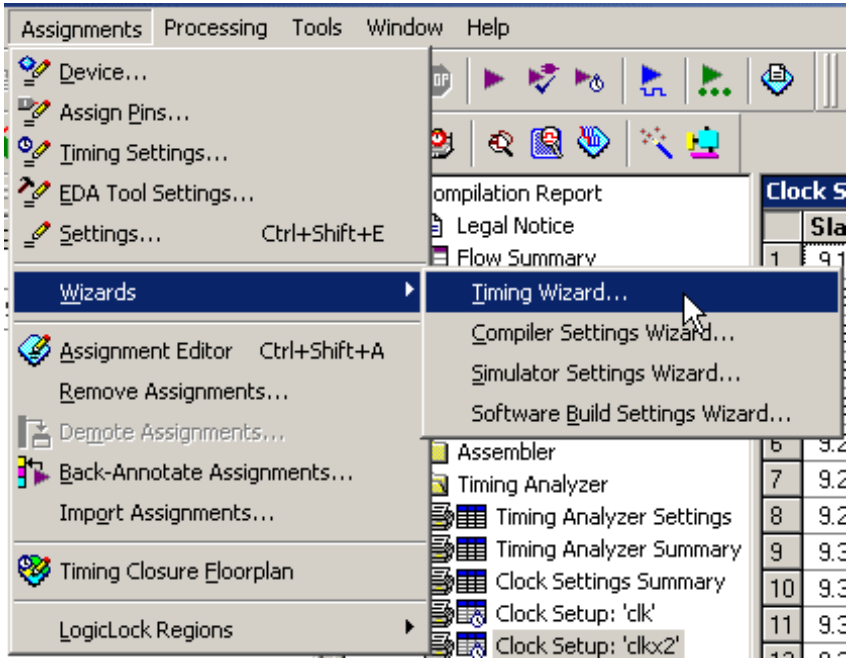
- Reports Best-Case I/O Results
  - Minimum tco
  - th
  - Minimum tpd
- Uses Fastest Timing Model
  - Fastest Process
  - Highest Voltage
  - Lowest Temperature

Running Minimum Timing Analysis

- Processing ⇒ Start ⇒ Start Minimum Timing Analysis
- Must Re-Run Standard Timing Analysis Afterwards
  - Netlist Annotated with Minimum Values
  - Previous Standard Analysis Overwritten

Timing Wizard

- Easy Way to Enter Timing Assignments
- Consolidates Timing Settings into One Menu
  - Individual Clock Settings OR Overall Circuit Frequency
  - Default System Timing
  - Default External Input/Output Delays
  - Enable/Disable Timing Analysis during Compilation
  - TDC



Exercise 4

Please go to Exercise 4 in the Exercise Manual

Exercise Summary

- Multi-Clock Timing Analysis
  - Slack Analysis
- Used Timing Wizard to Define Clocks
- Making Multi-Cycle Timing Assignments
  - Used the Node Finder

Timing Analysis Summary

- Single-Clock Timing Analysis
- Timing Assignments
- Multi-Clock Timing Analysis
- Multi-Cycle Timing Assignments
- Minimum Timing Analysis

More Timing Analysis Info

- Quartus II Handbook: Using Quartus II Timing Analysis

20 YEARS *of*

ALTERA®

INNOVATION

Designing with Quartus II

Simulation

QUARTUS® II

Quartus II Simulation

- Simulator Method & Features Overview
- Simulator Settings
- VWF File Creation
- Simulation Output
- 3rd Party Simulation

Supported Simulation Methods

- Quartus II
  - VWF (Vector Waveform File)
    - Primary Graphical Waveform File
  - VEC (Vector File)
    - Text-Based Input File
  - SCF (Simulator Channel File)
    - MAX+PLUS II Graphical Waveform File
  - TBL (Table File)
    - Text-Based Output File from Quartus II or MAX+PLUS II
  - Tcl/TK Scripting
- 3rd Party Simulators
  - Verilog/VHDL Testbench

Simulator Features

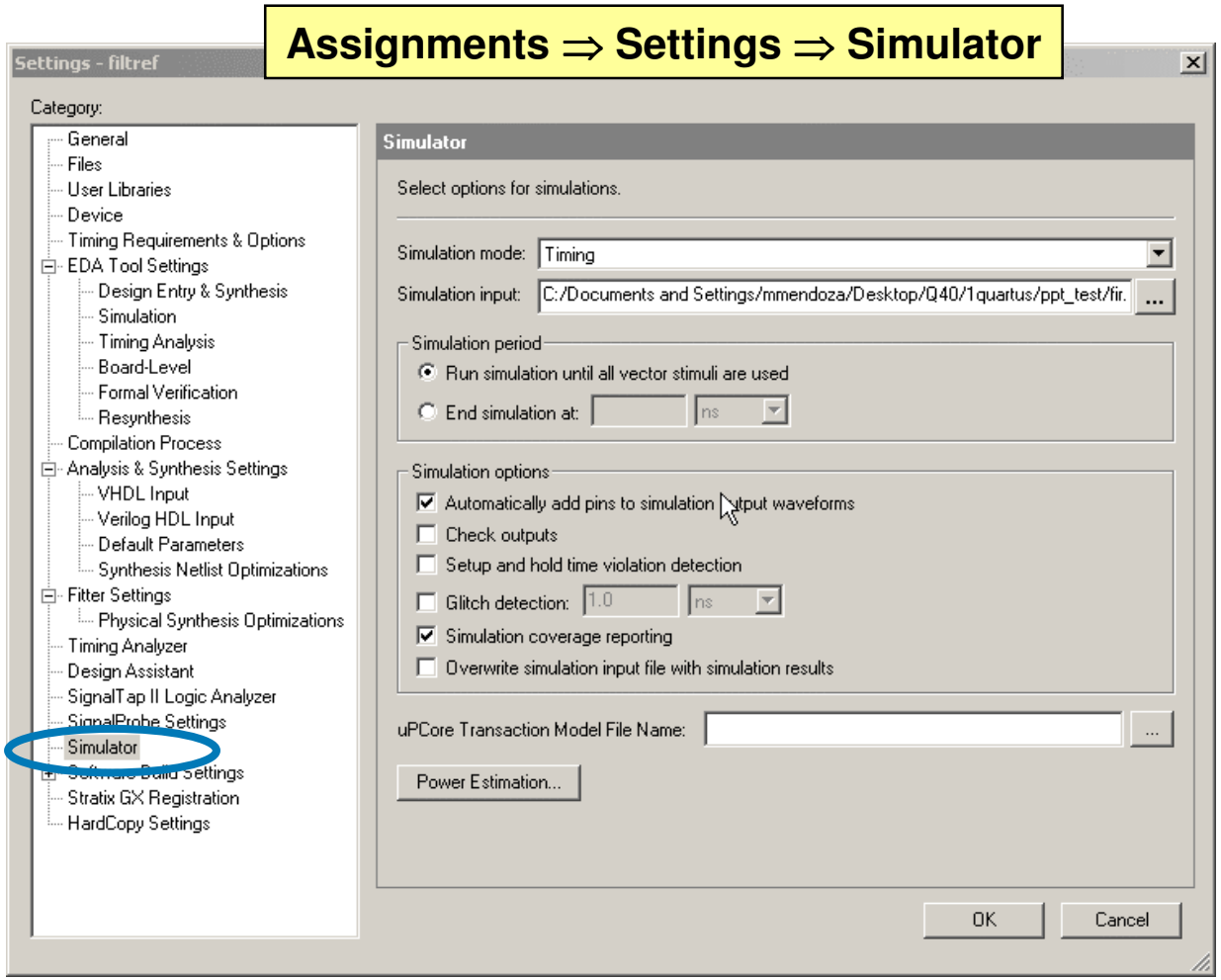
- Supports 9 Signal Values
  - 1 Forcing ‘1’
  - 0 Forcing ‘0’
  - XForcing Unknown
  - U Uninitialized
  - ZHigh Impedance
  - H Weak ‘1’
  - L Weak ‘0’
  - W Weak Unknown
  - DC Don’t Care

Simulator Features (cont.)

- Performs PowerGauge™ Analysis
- 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

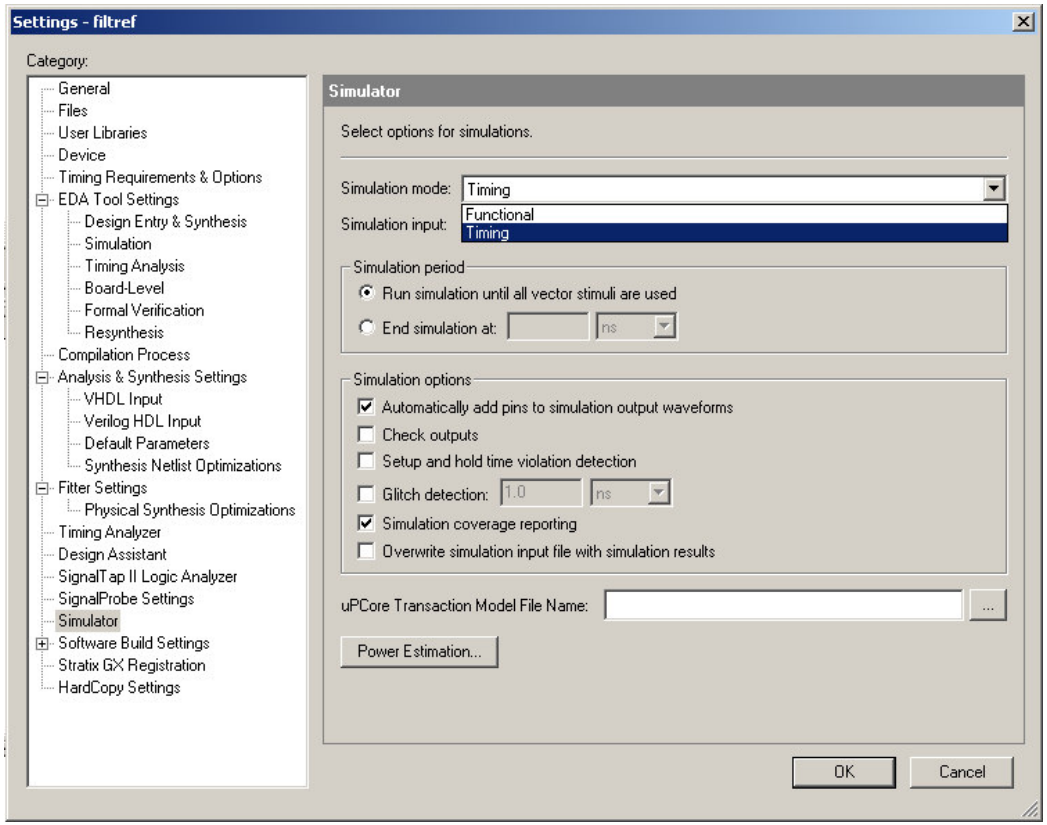
Simulator Settings

- Mode
- Input File
- Period
- Options



Simulator Mode

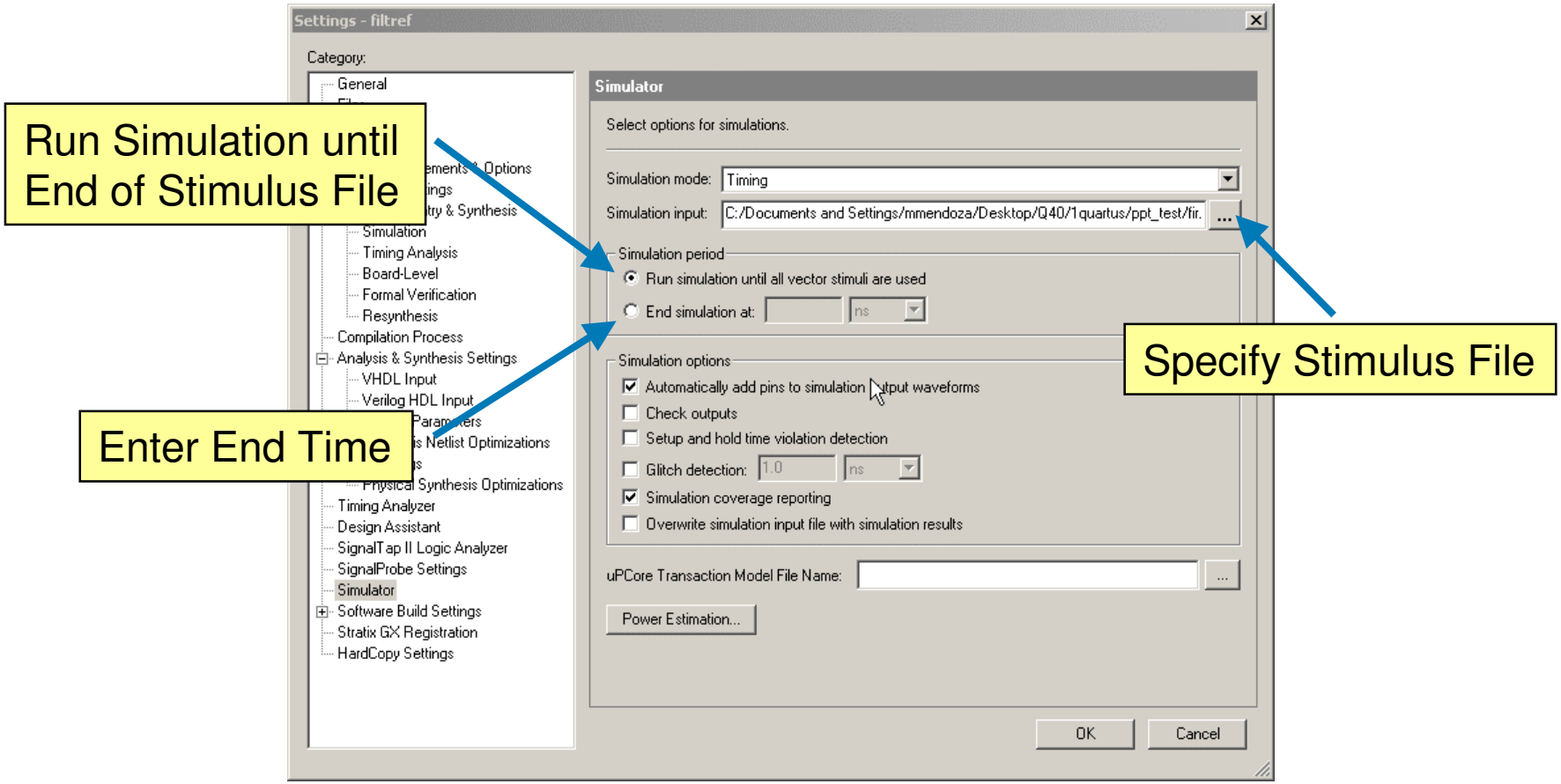
- Functional*
  - Type: RTL
  - Uses Pre-Synthesis Netlist
- Timing
  - Type: Gate-Level or Post-Place & Route
  - Uses Fully Compiled Netlist



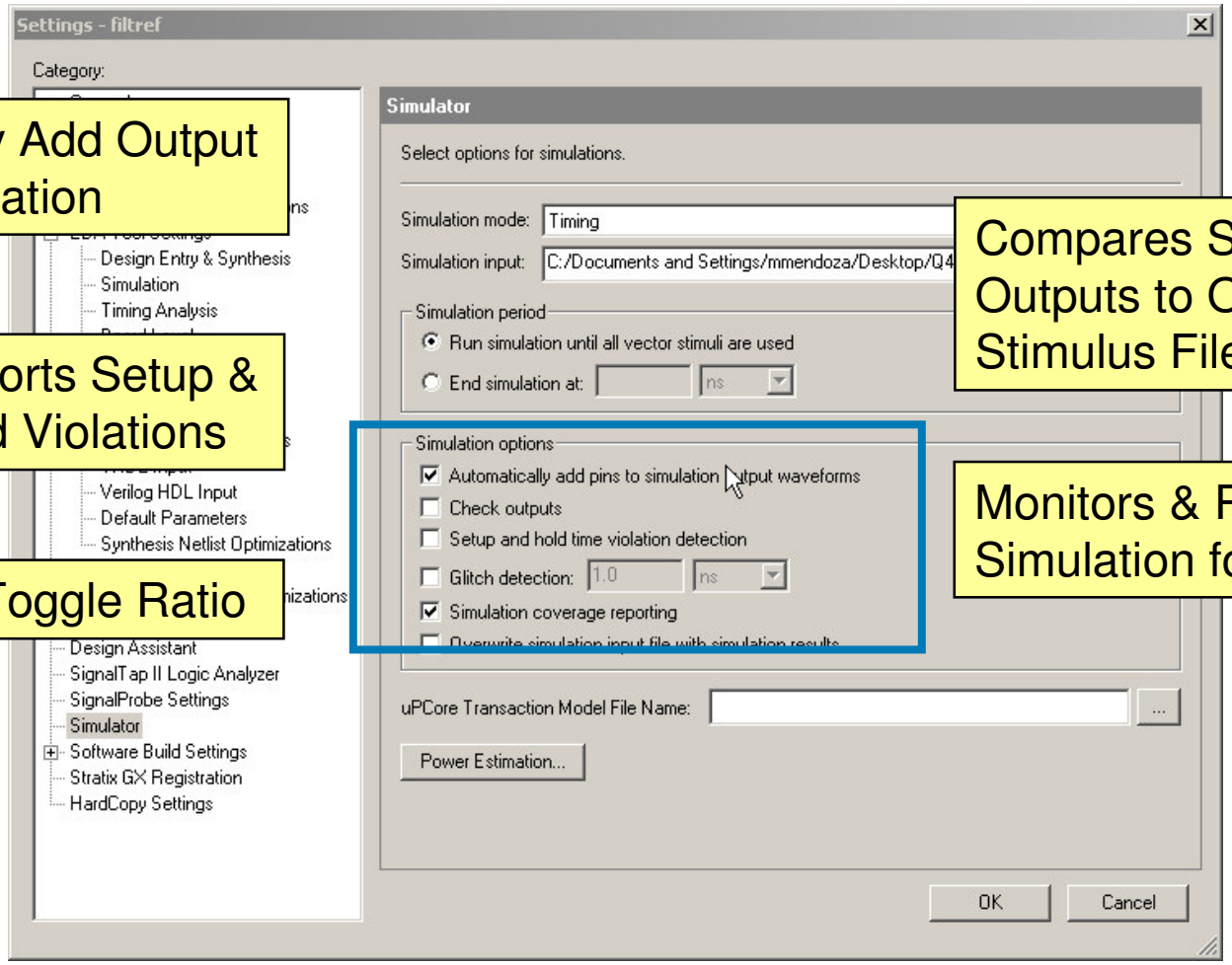
*** Must Generate Functional Simulation Netlist (Processing Menu) before Performing Functional Simulation**

Simulator Input & Period

- Specifies Stimulus & Length of Simulation Period



Simulator Options



Automatically Add Output Pins to Simulation

Reports Setup & Hold Violations

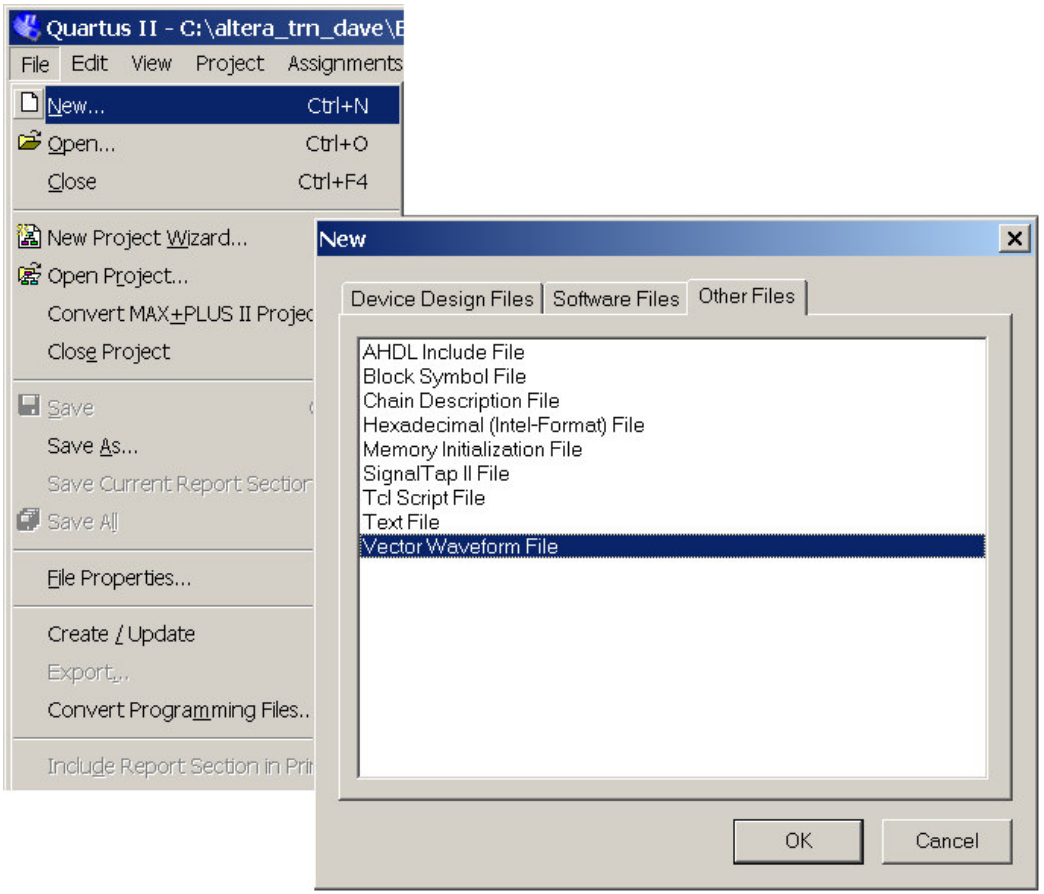
Reports Toggle Ratio

Compares Simulation Outputs to Outputs in Stimulus File

Monitors & Reports Simulation for Glitches

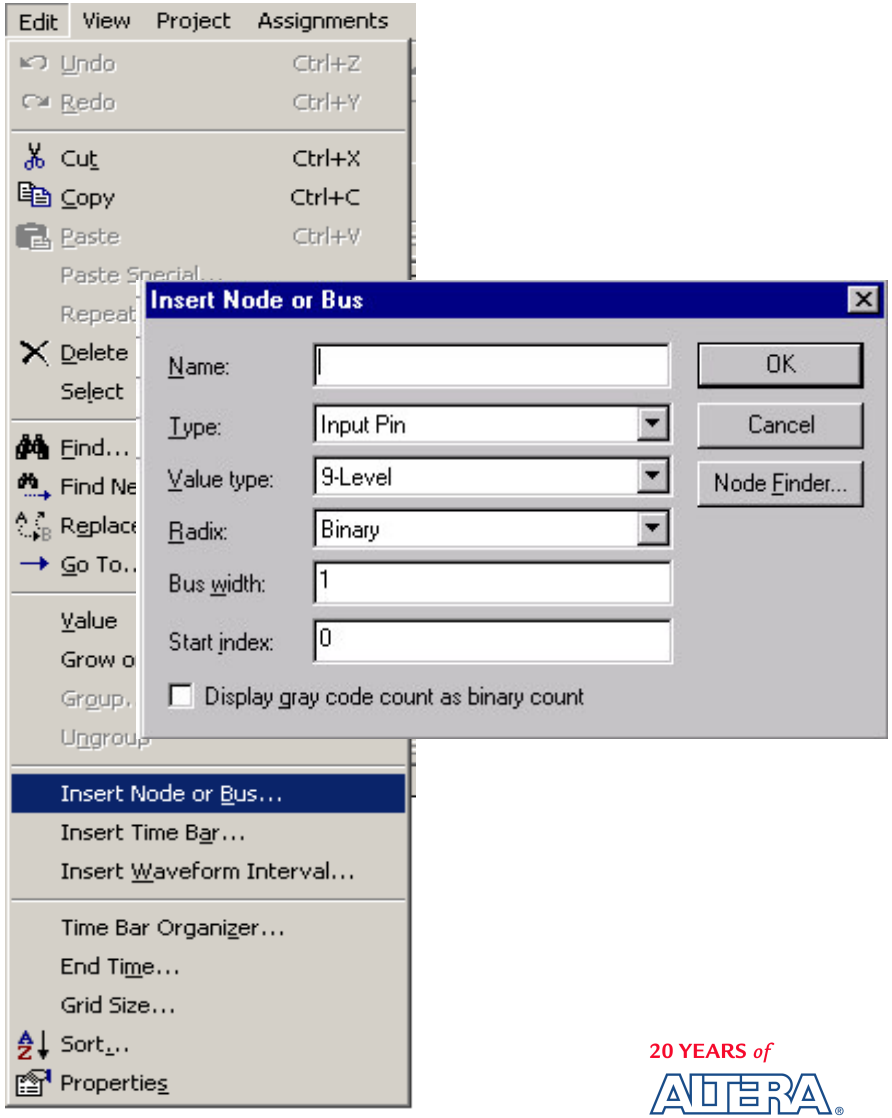
Create New Vector Waveform File

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



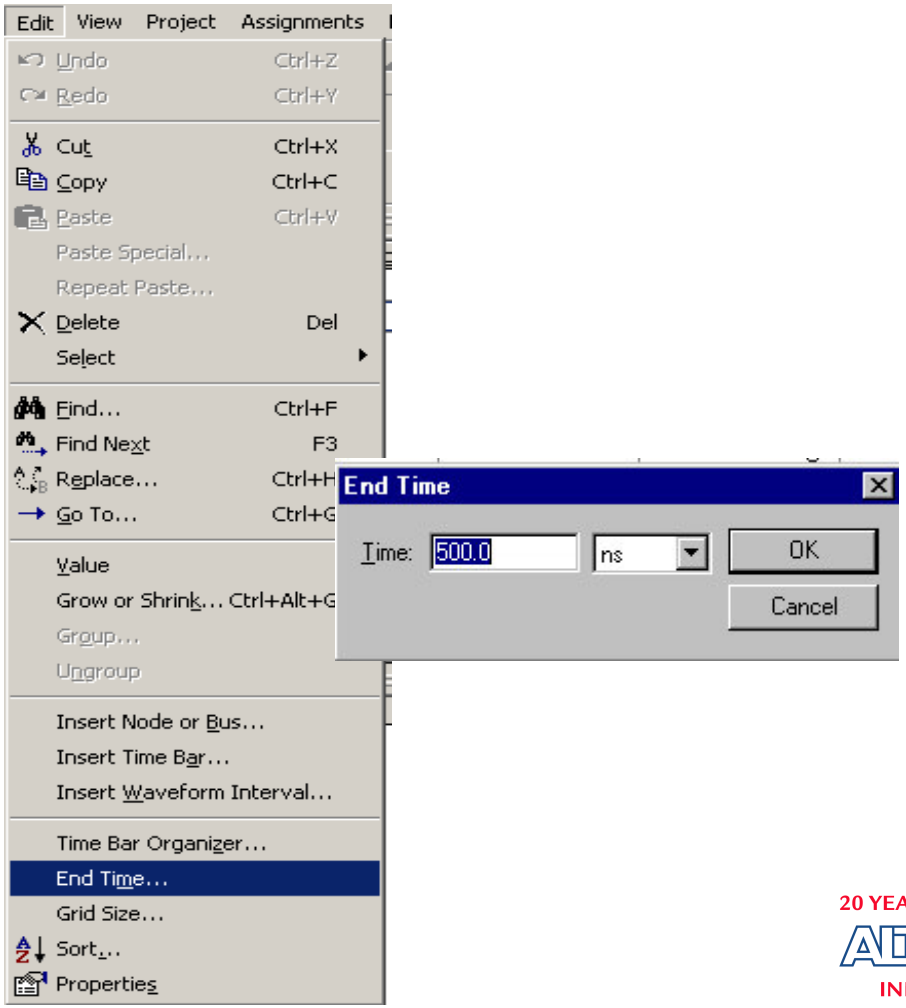
Insert Nodes

- Select **Insert Node or Bus** (Edit Menu)
  - VWF Must Be Open
  - Use Node Finder



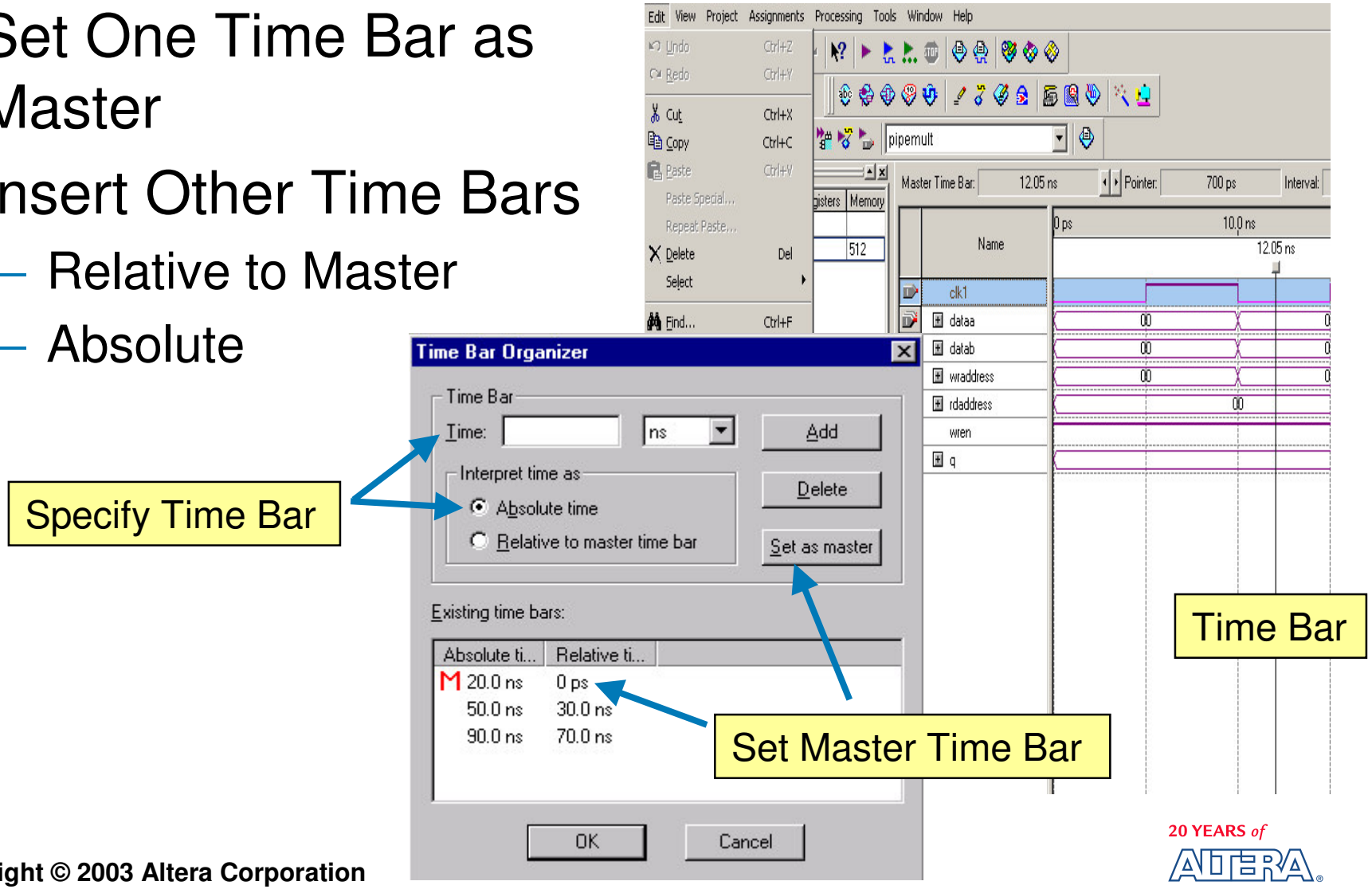
Specify End Time

- Maximum Length of Simulation Time
  - Edit Menu



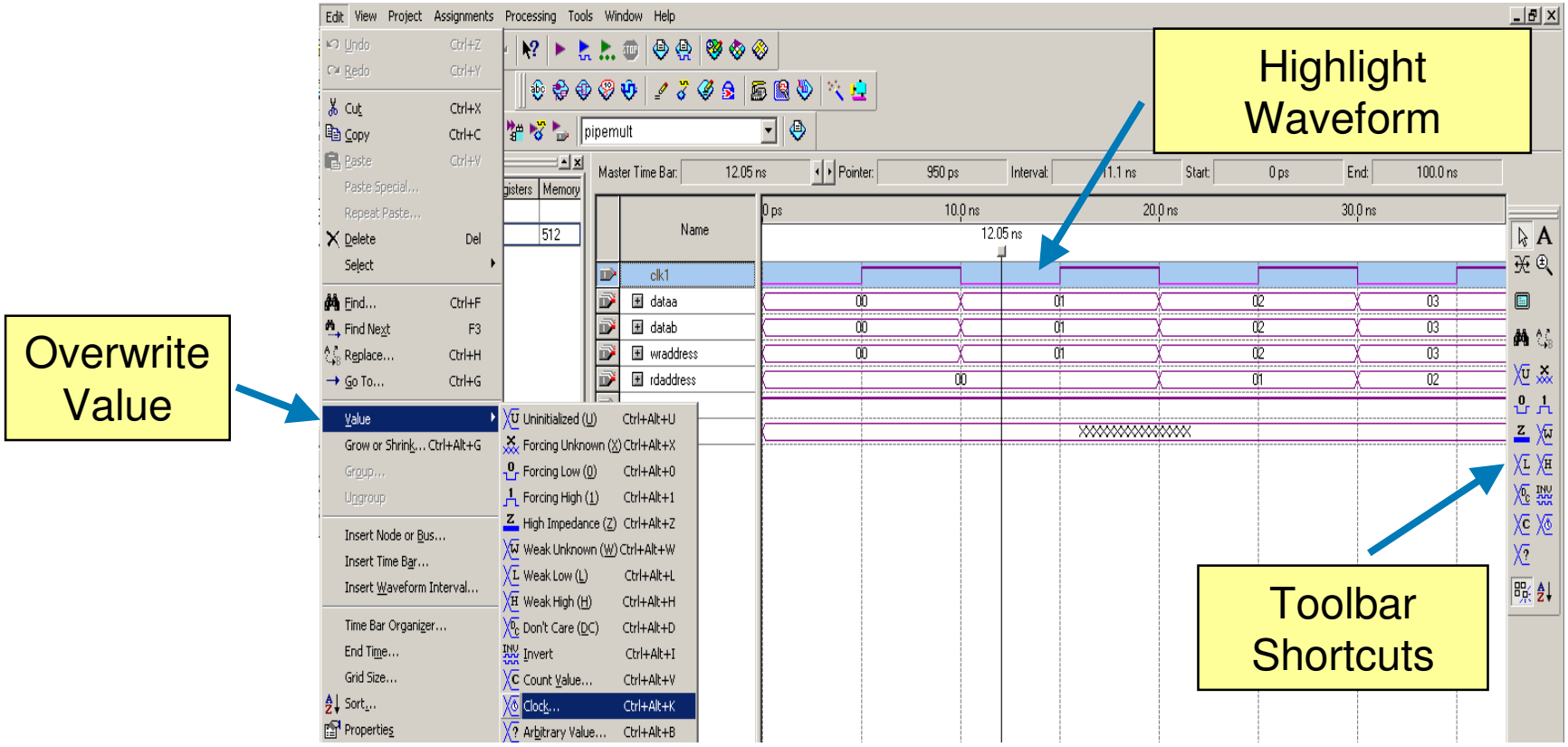
Insert Time Bars

- Set One Time Bar as Master
- Insert Other Time Bars
  - Relative to Master
  - Absolute



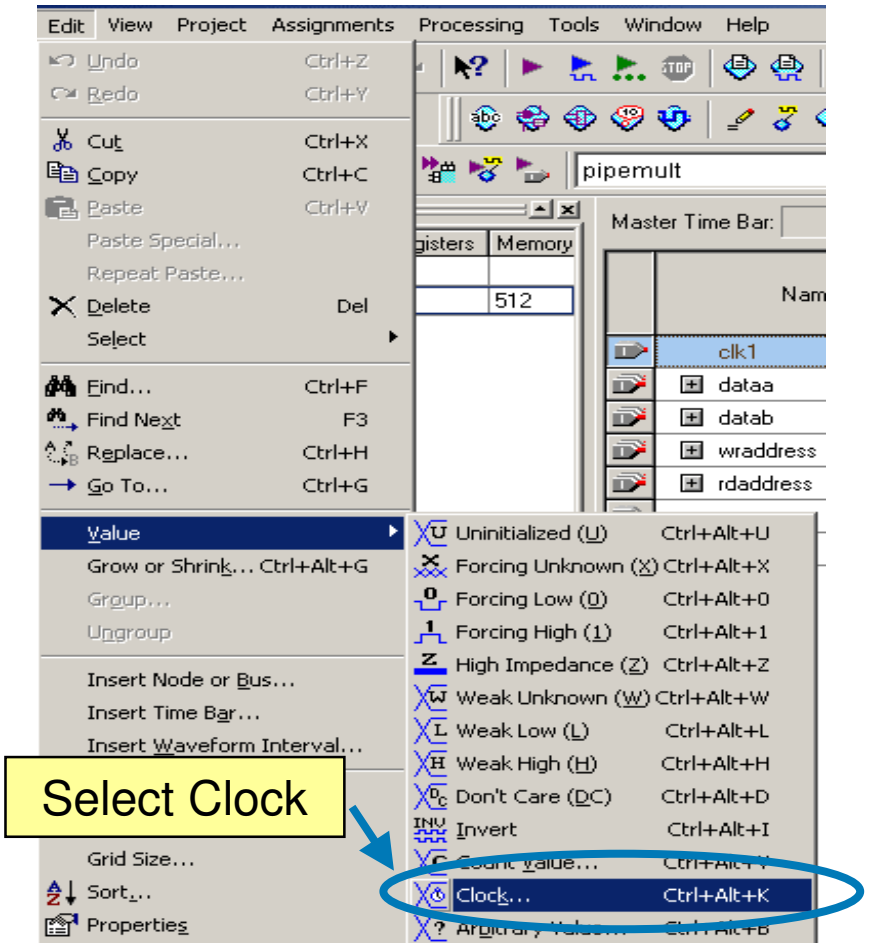
Draw Stimulus Waveform

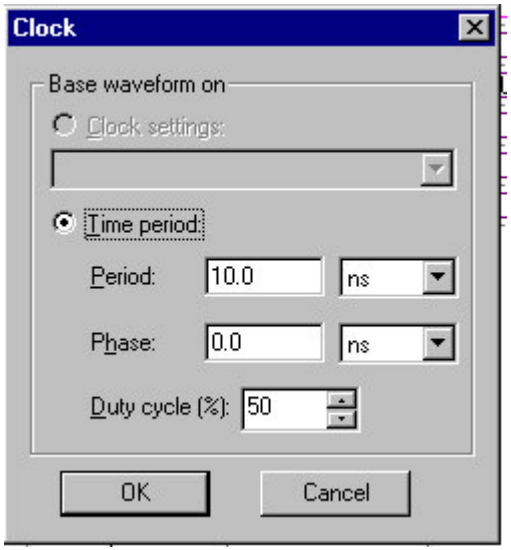
- Highlight Portion of Waveform to Change
- Overwrite Value with Desired Value



Create Clock

- Highlight Waveform & Enter Period

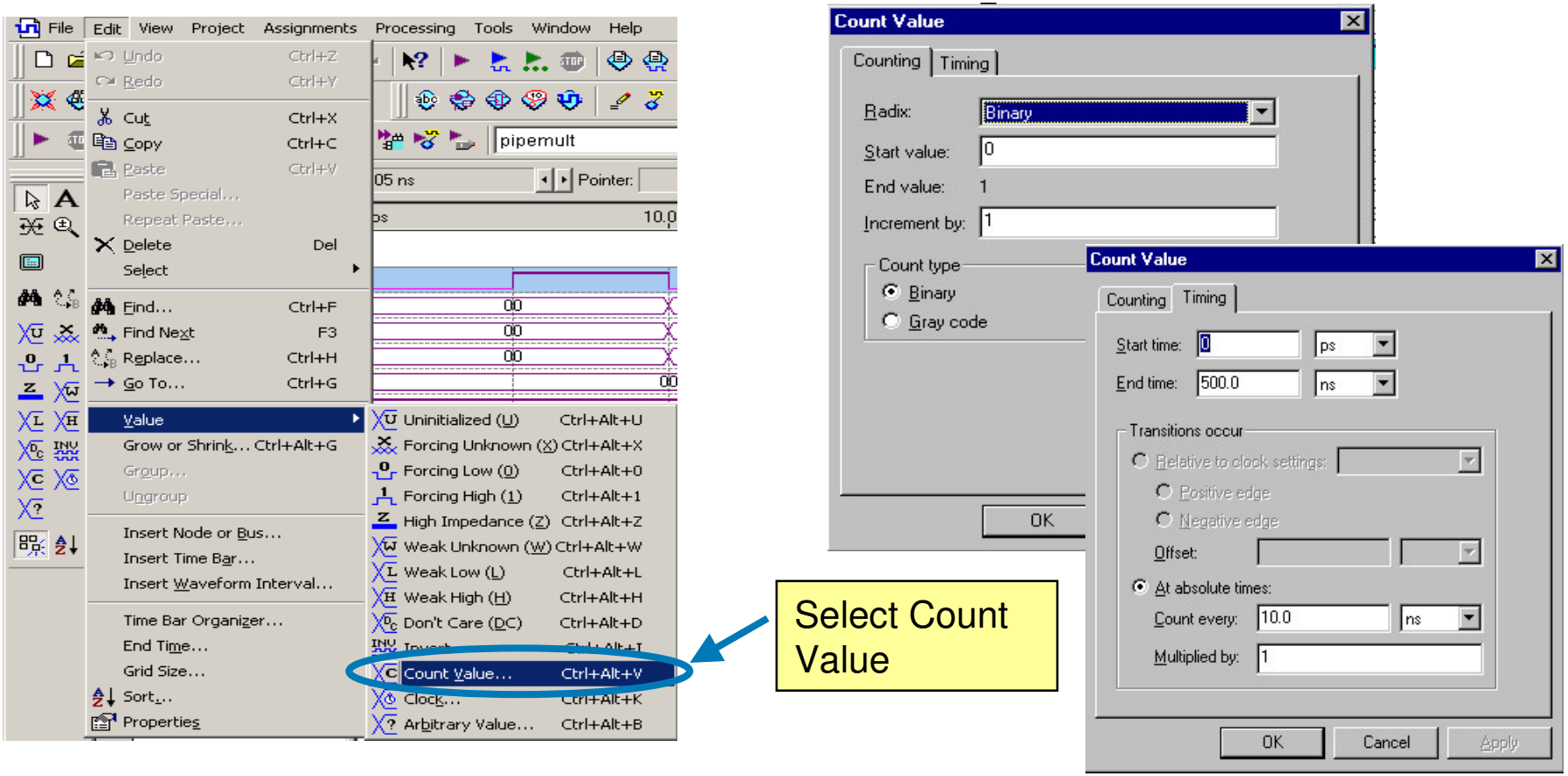




Select Clock Defined as Timing Setting or Specify Clock Period

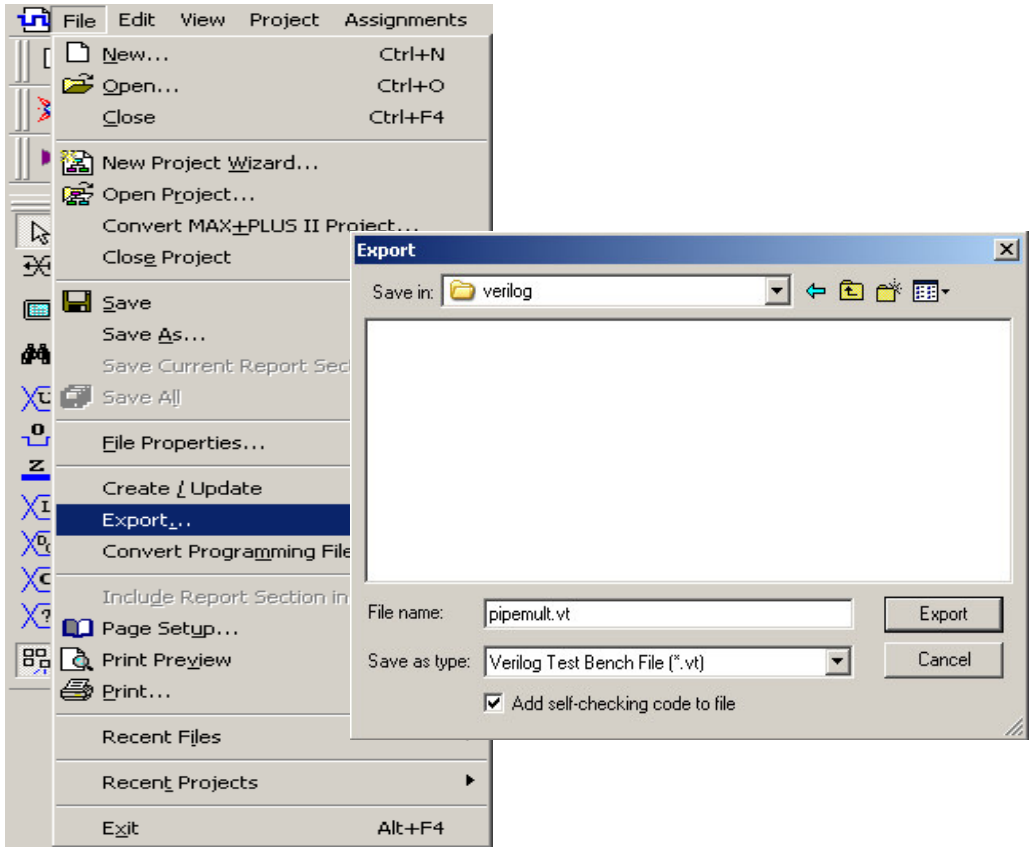
Creating Counting Pattern

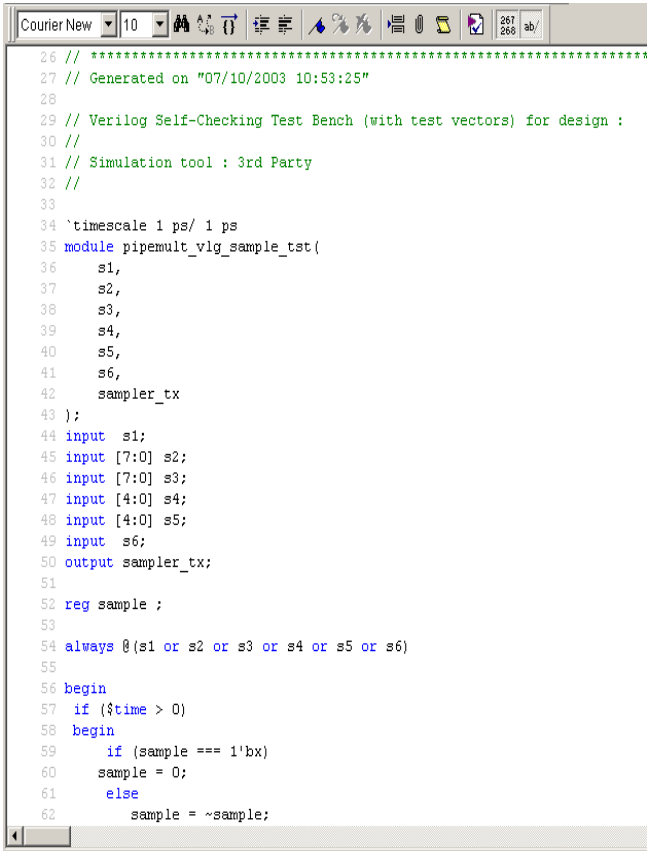
- Highlight Waveform & Enter Pattern



Waveform to Testbench Generator

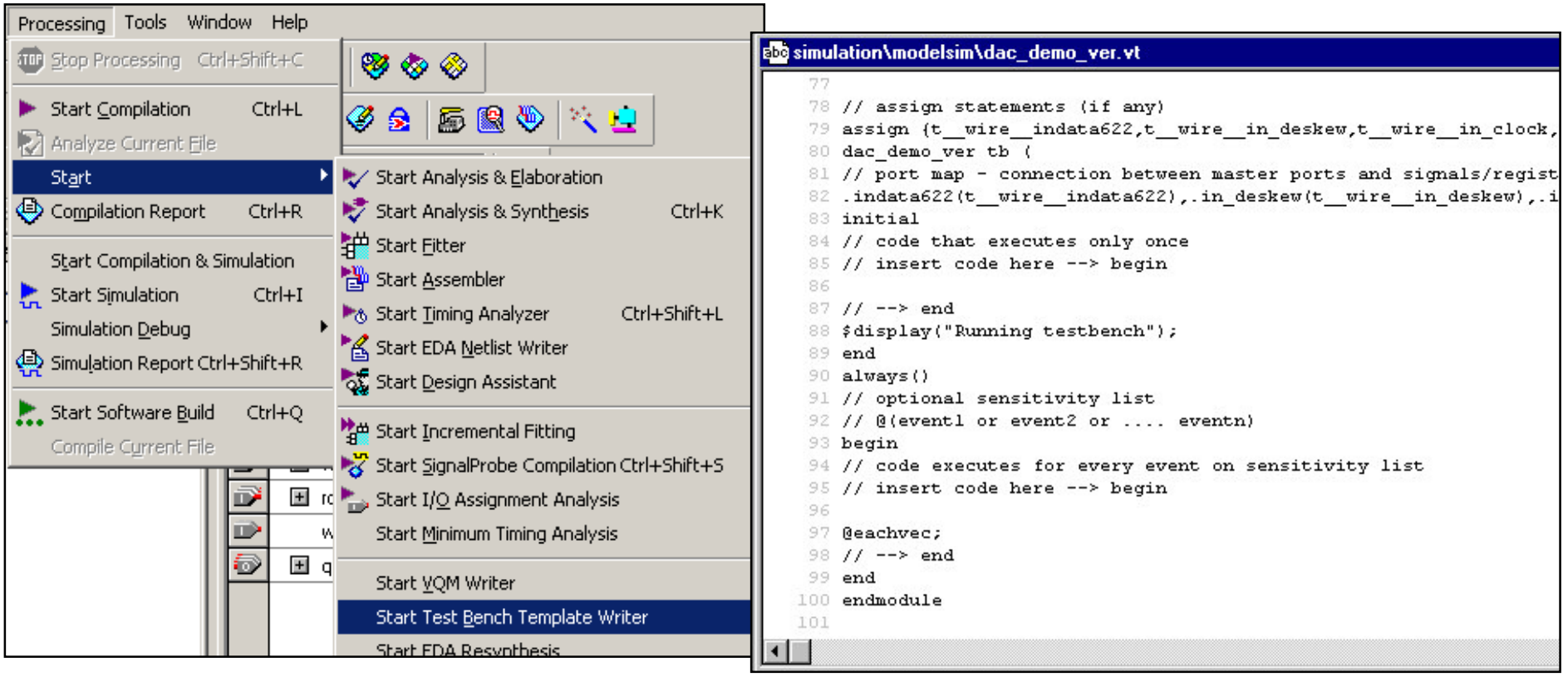
- Converts VWF into HDL Testbench





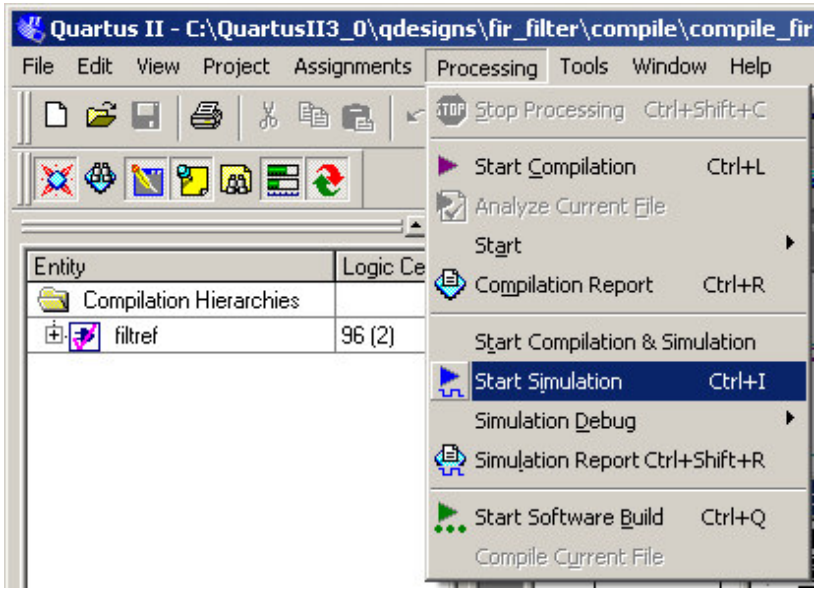
Testbench Template Generator

- Generates HDL Testbench Template
  - User Inserts Test Stimulus



Starting Simulation

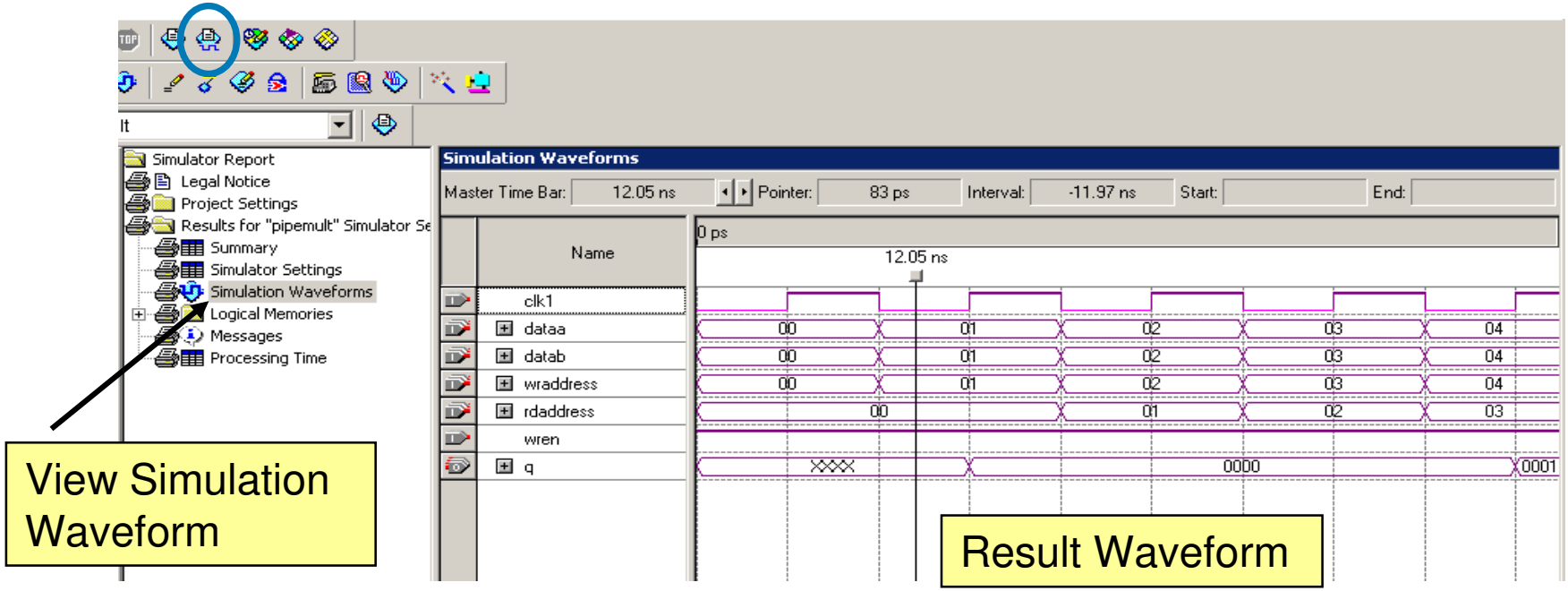
- Processing Menu ⇒ Start Simulation



- Scripting

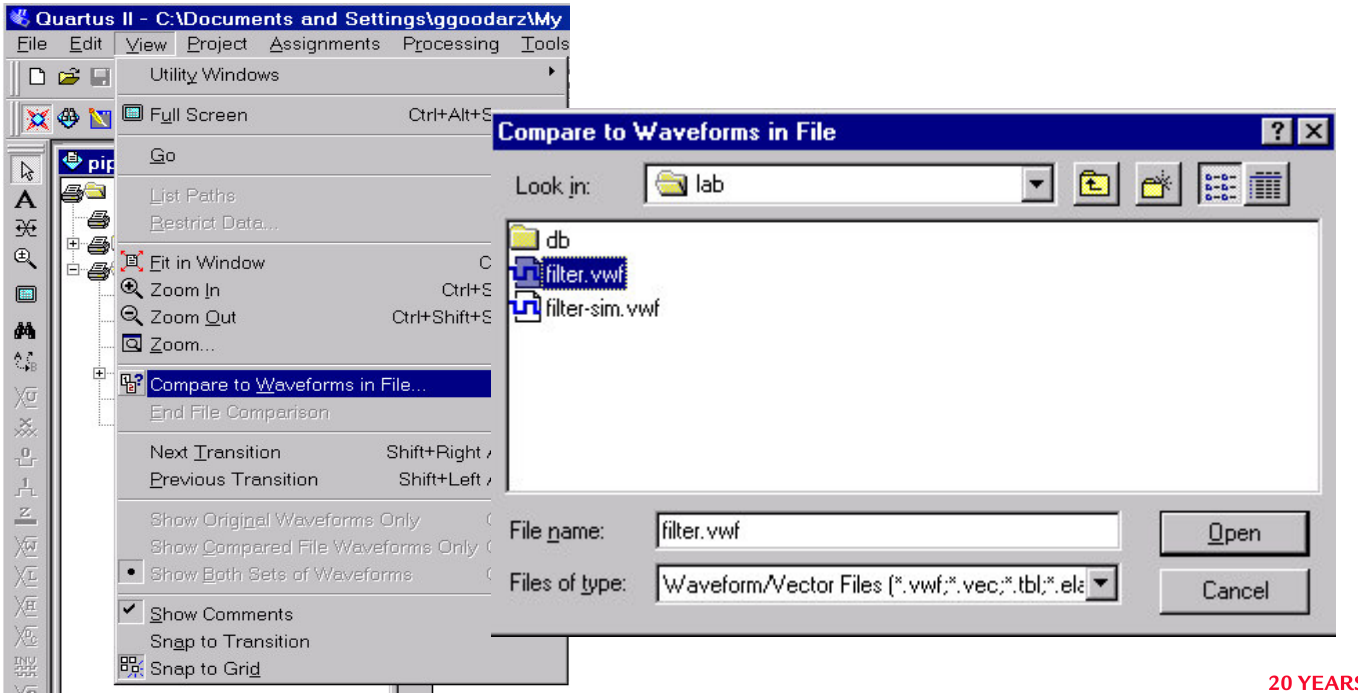
Simulator Report

- Displays Simulation Result Waveform



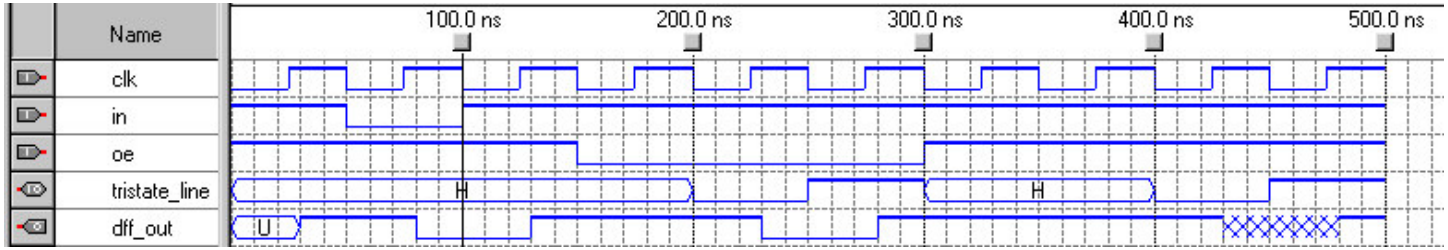
Comparing Waveforms

- Select Compare to Waveforms (View Menu)
  - Simulation Waveform Must Be Open
- Select VWF Comparison File

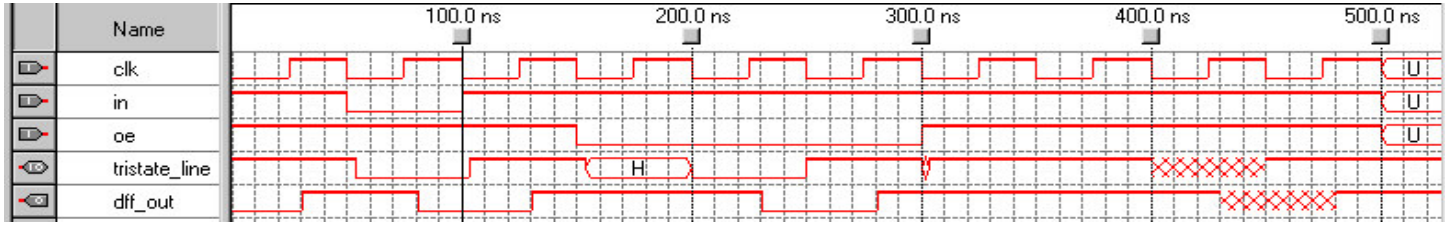


Compared Waveforms (Simulator Report)

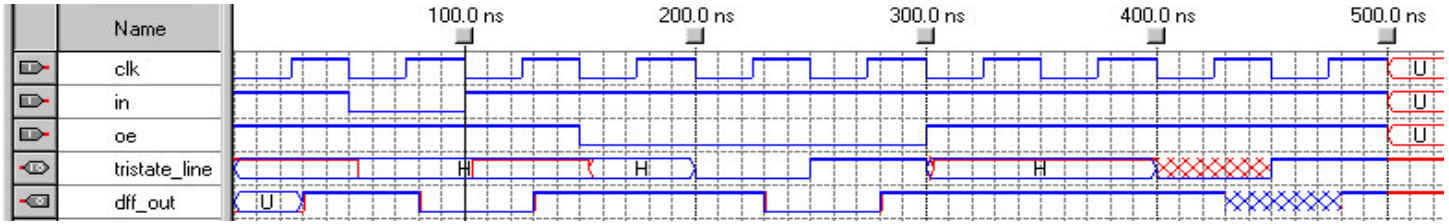
- Original (Ctrl+1)



- Actual (Ctrl+2)

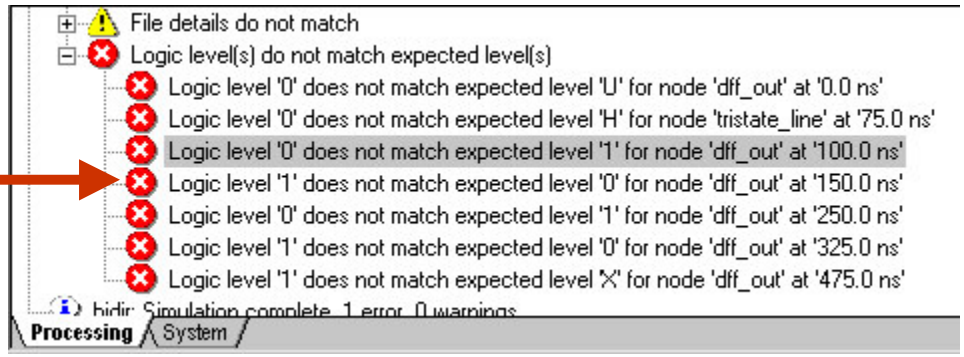


- Compared (Ctrl+3)

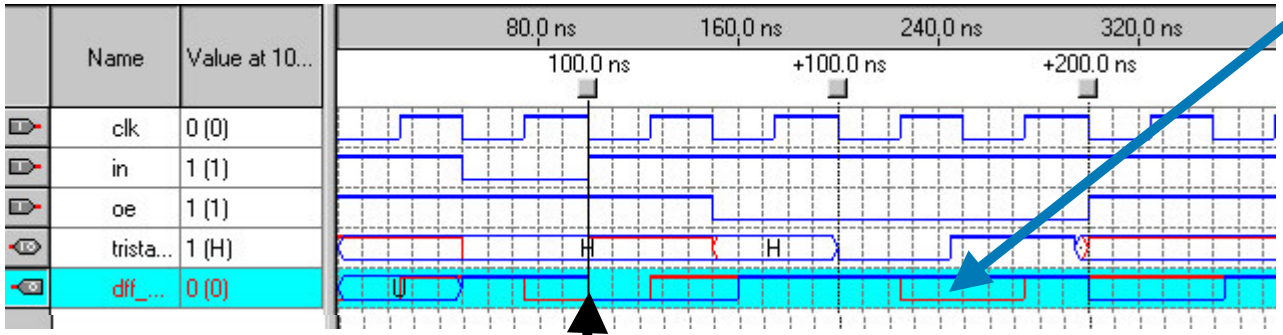


Using Message Window to Debug

1
Double clicking on message



2
Highlights signal with unexpected value



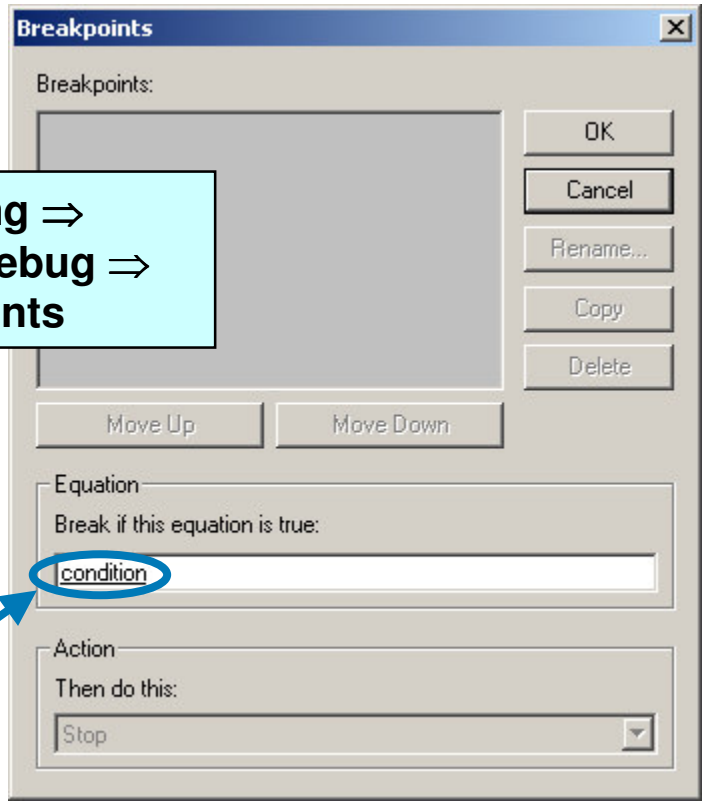
3
Creates time bar at occurrence of unexpected value

Breakpoints

- Interrupts Simulation at Specified Points
- Consists of 2 Parts
  - Equation (Condition)
  - Action
    - Stop
    - Give Error
    - Give Warning
    - Give Info

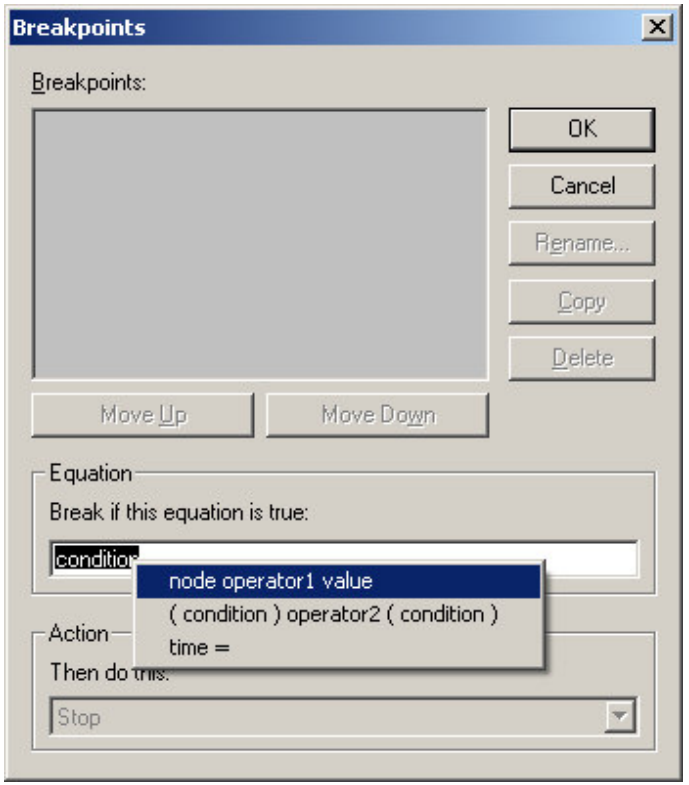
Processing ⇒ Simulation Debug ⇒ Breakpoints

Click on condition to Build Equation

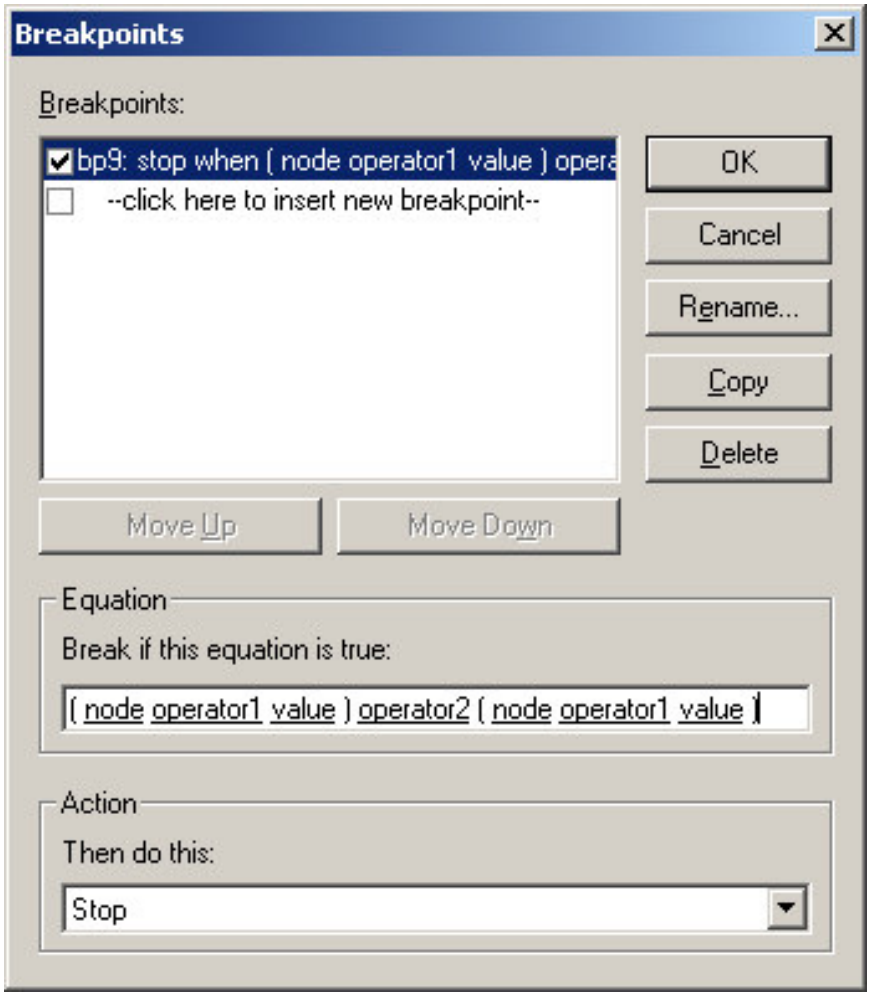


Breakpoint Conditions

- <Node> <Operator1> <Value>
  - Single Condition
  - Ex. ena = 1
- Time = <Value>
  - Single Condition
  - time = 500ns
- <Condition> <Operator2> <Condition>
  - Complex Tests
  - ena = 1 && time > 500ns

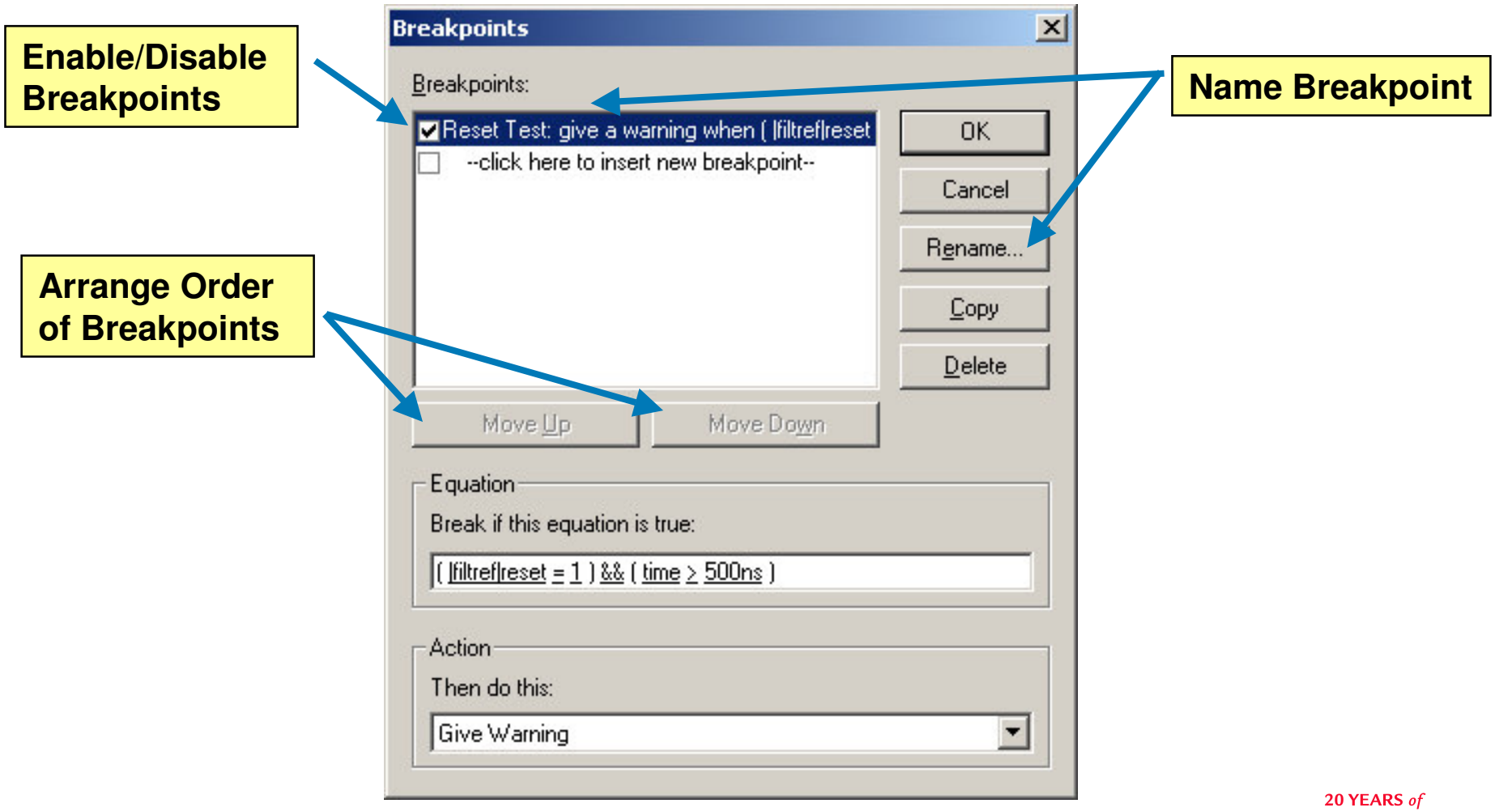


Breakpoint Equations (cont.)



- Node
  - Opens Node Finder
- Operator1
  - <, >, =
- Operator2
  - && (AND)
  - || (OR)

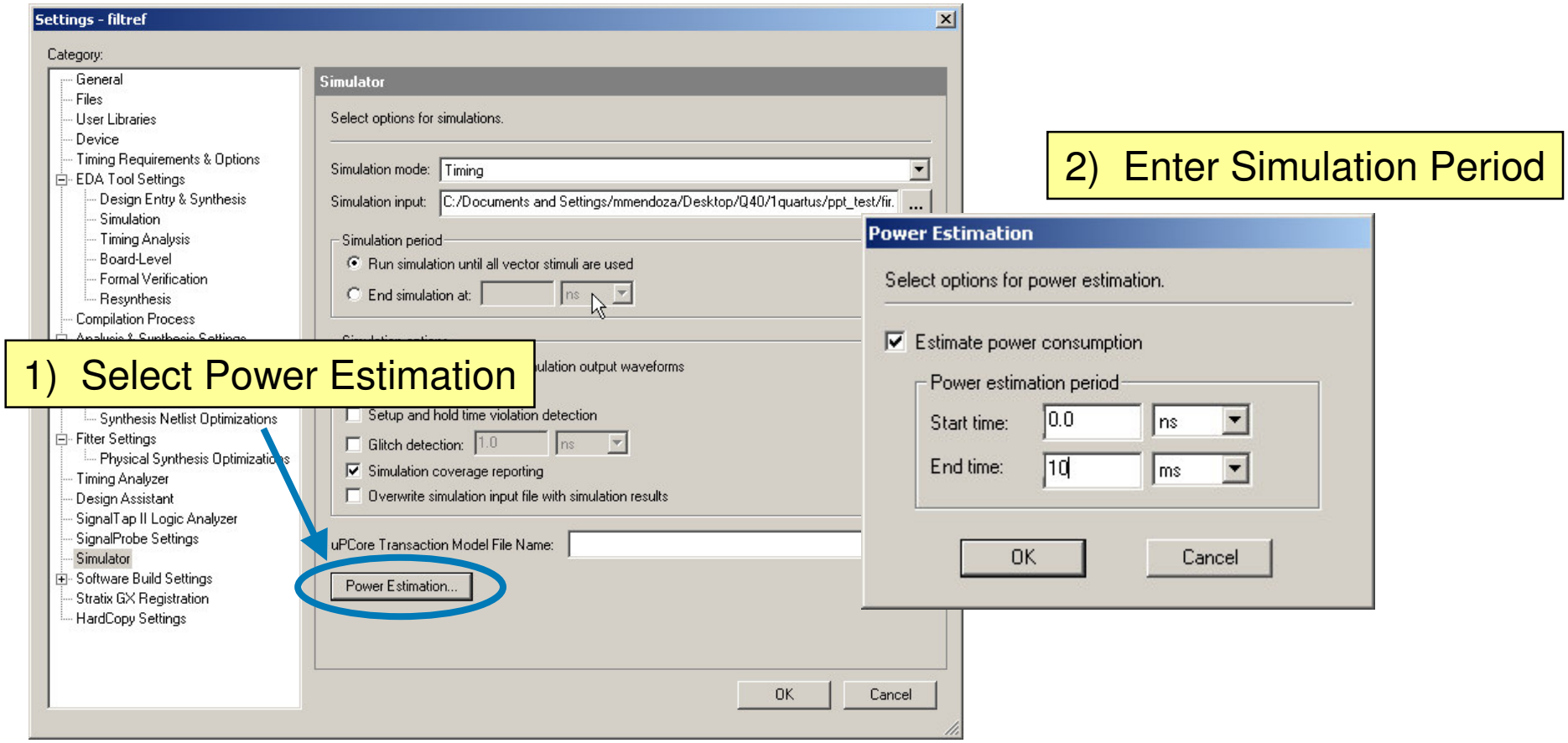
Example Breakpoint



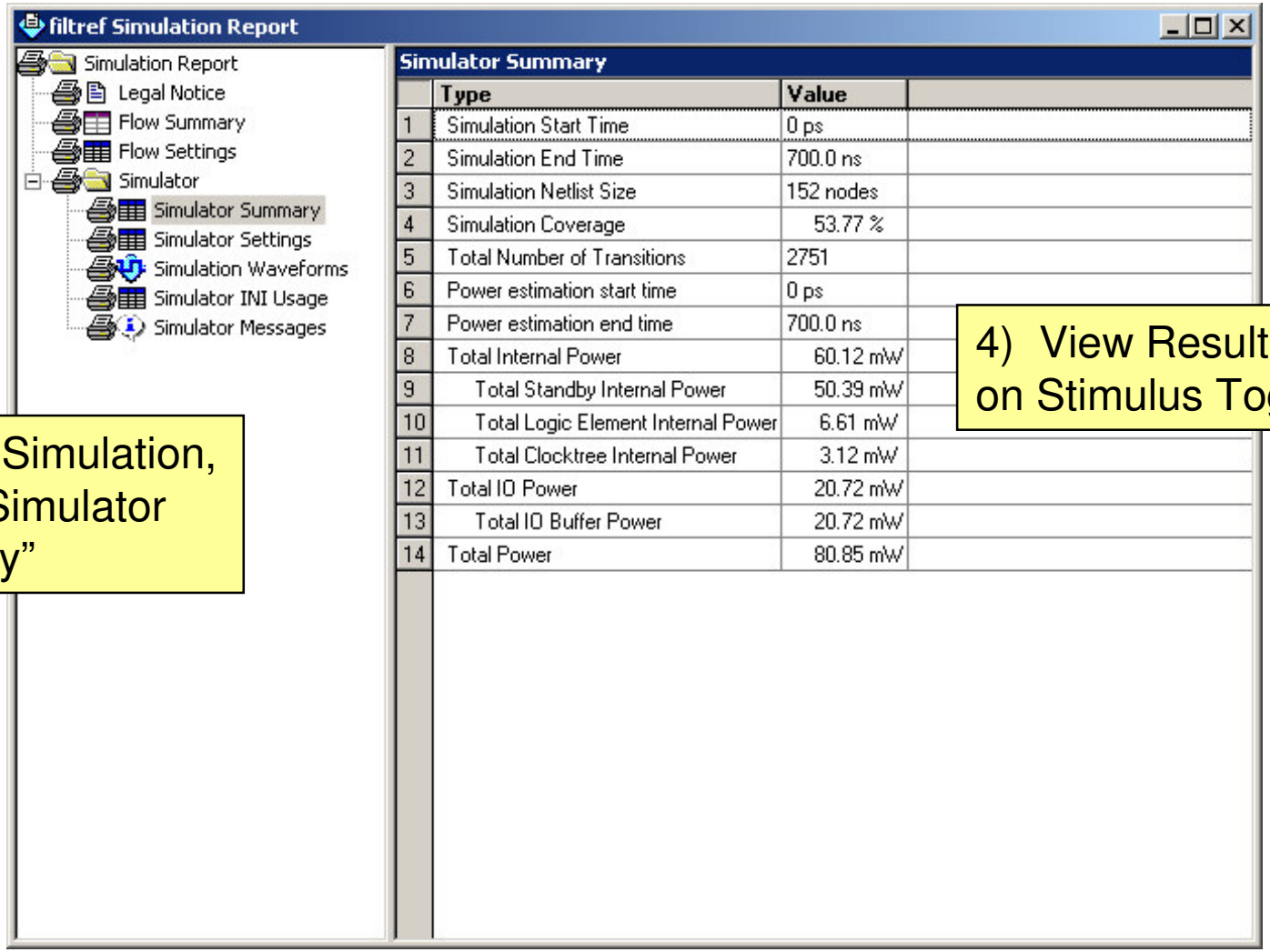
PowerGauge Analysis Software

- Estimates Power Consumption Based on Toggle Rates
  - Derives Toggles Rate from Simulation Stimulus
  - Uses Quartus II Simulator
- Supports Multiple I/O Standards
- Supports Stratix, Stratix GX, Cyclone MAX 7000AE, MAX 7000B, MAX 3000 Families
- 3rd Party Simulation Tools Output PWF File that Can Be Read by Quartus II Simulator

Power Analysis



Power Analysis (cont.)



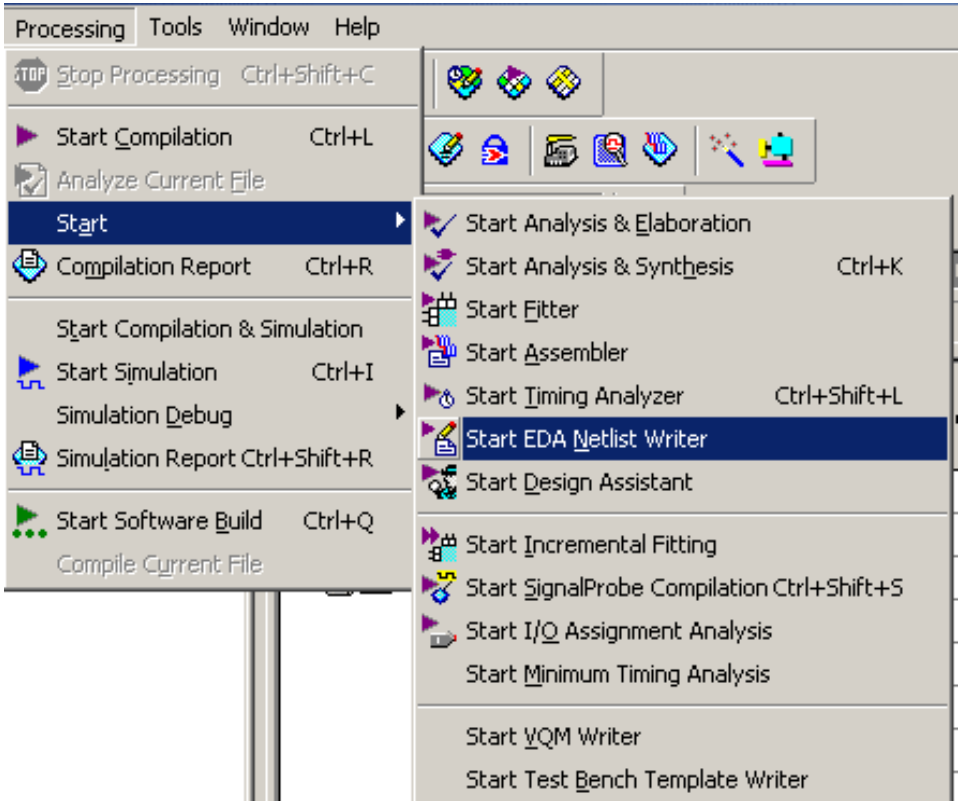
	Type	Value
1	Simulation Start Time	0 ps
2	Simulation End Time	700.0 ns
3	Simulation Netlist Size	152 nodes
4	Simulation Coverage	53.77 %
5	Total Number of Transitions	2751
6	Power estimation start time	0 ps
7	Power estimation end time	700.0 ns
8	Total Internal Power	60.12 mW
9	Total Standby Internal Power	50.39 mW
10	Total Logic Element Internal Power	6.61 mW
11	Total Clocktree Internal Power	3.12 mW
12	Total IO Power	20.72 mW
13	Total IO Buffer Power	20.72 mW
14	Total Power	80.85 mW

3) After Simulation, Select "Simulator Summary"

4) View Results Based on Stimulus Toggle Rate

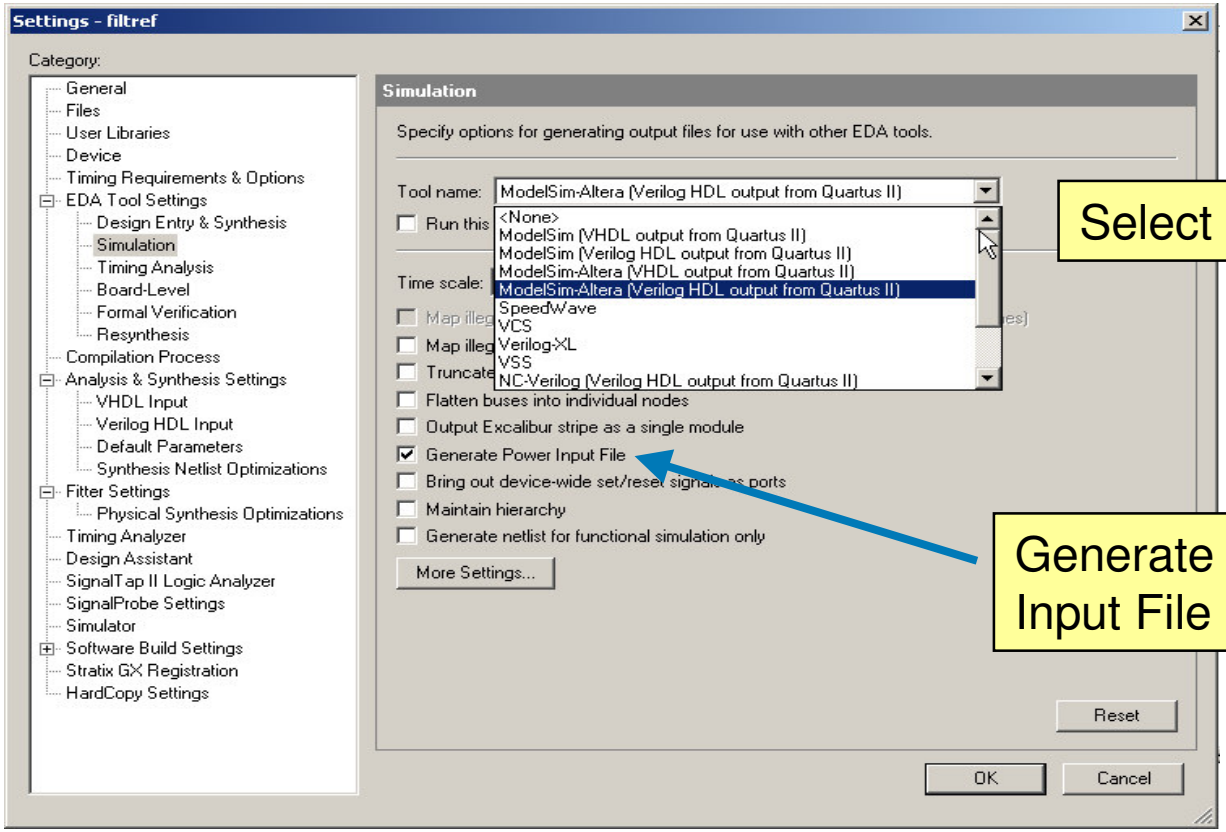
Using 3rd Party Simulators

- Mentor Graphics
  - ModelSim
- Cadence
  - VERILOG-XL
  - NC-Verilog
  - NC-VHDL
- Synopsys
  - VCS
  - VSS
  - Scirocco



Specify Simulator

- Select EDA Tools Settings
  - Assignments Menu



3rd Party Simulation Files

- Functional Simulation
  - Use 220models & altera_mf Megafunction Model Files
- VHDL Timing Simulation
  - Use Quartus II-Generated VHO & SDO Files
  - Use <device_name>_ATOMS.VHD & <device_name>_ATOMS_COMPONENTS.VHD Files
    - Located in eda\sim_lib Directory
- Verilog Timing Simulation
  - Use Quartus II-Generated VO & SDO Files
  - Use <device_name>_ATOMS.VO File
    - Located in eda\sim_lib Directory

Exercise 5

Please go to Exercise 5 in the Exercise Manual

Exercise Summary

- Learn about Quartus II Simulator
- Draw Signals in VWF Files
- Functional Simulation
  - View Simulation Results

Simulation Summary

- Functional & Timing Simulation
- Creating a Vector Waveform File
- Power Analysis

20 YEARS *of*

ALTERA®

INNOVATION

Designing with Quartus II

Programming/Configuration

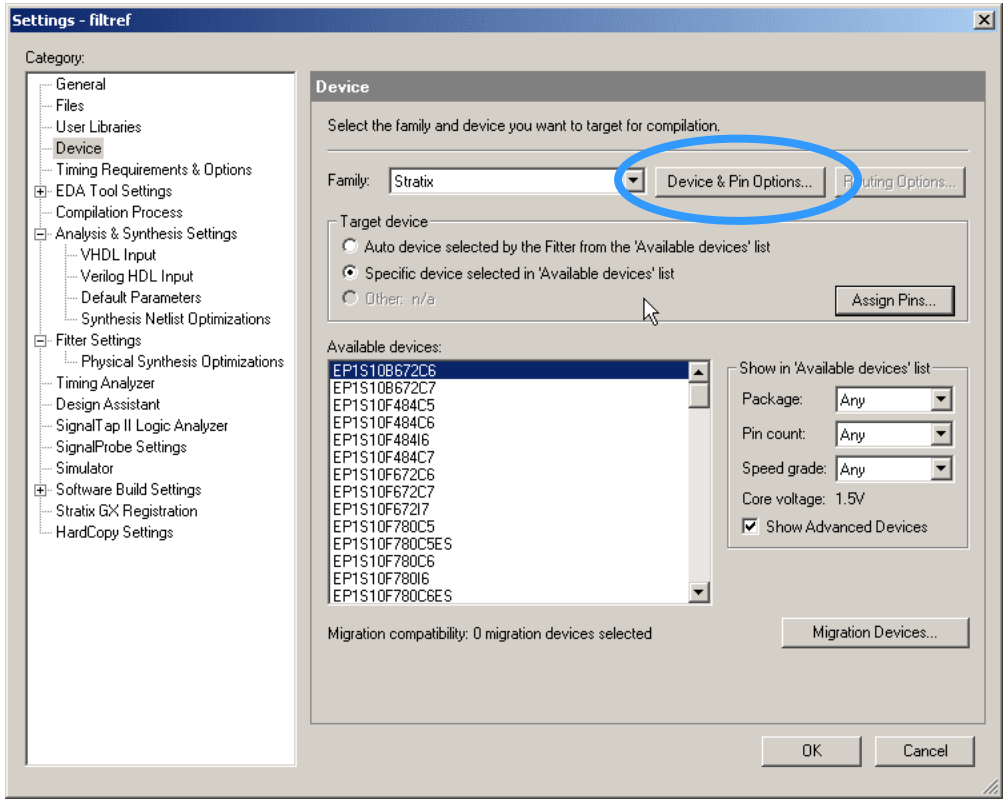
QUARTUS® II

Programming/Configuration

- Setting Device Options
- Assembler Module
- Programmer & Chain Description File
  - Programming Directly with Quartus II
- File Conversion
  - Creating Multi-Device Programming Files

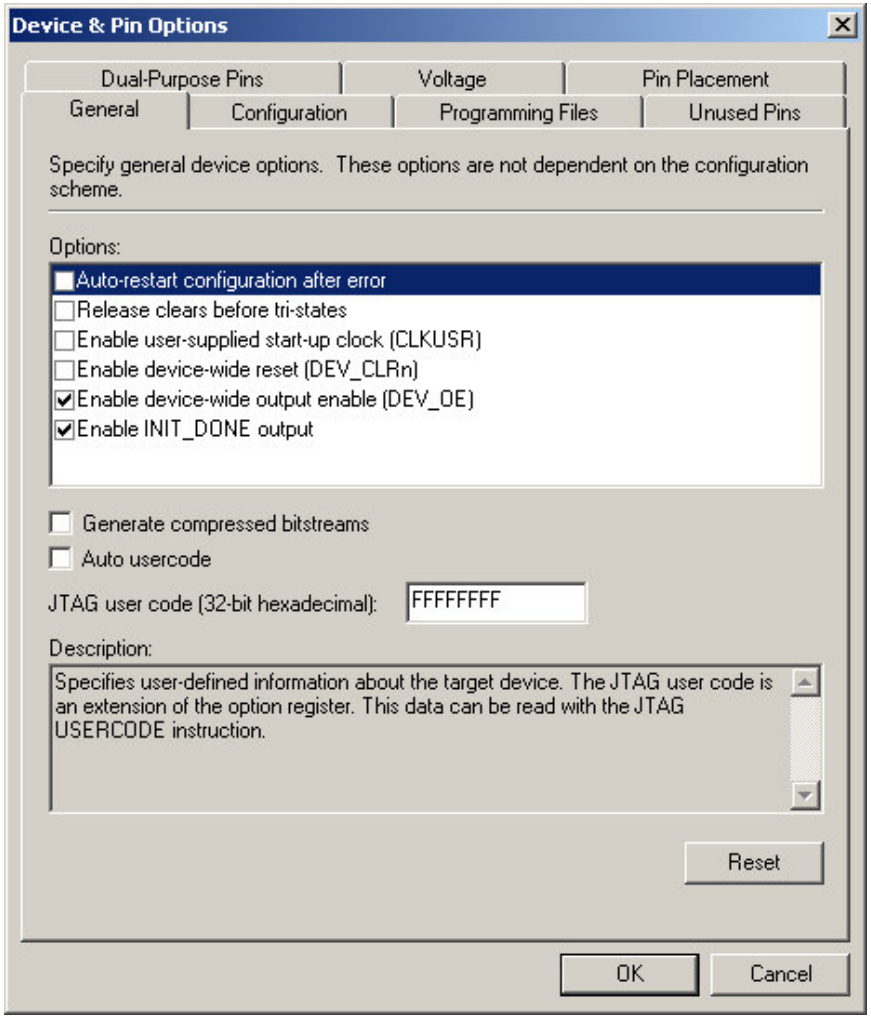
Setting Device Options

- Assignments ⇒ Device ⇒ Device & Pin Options Button



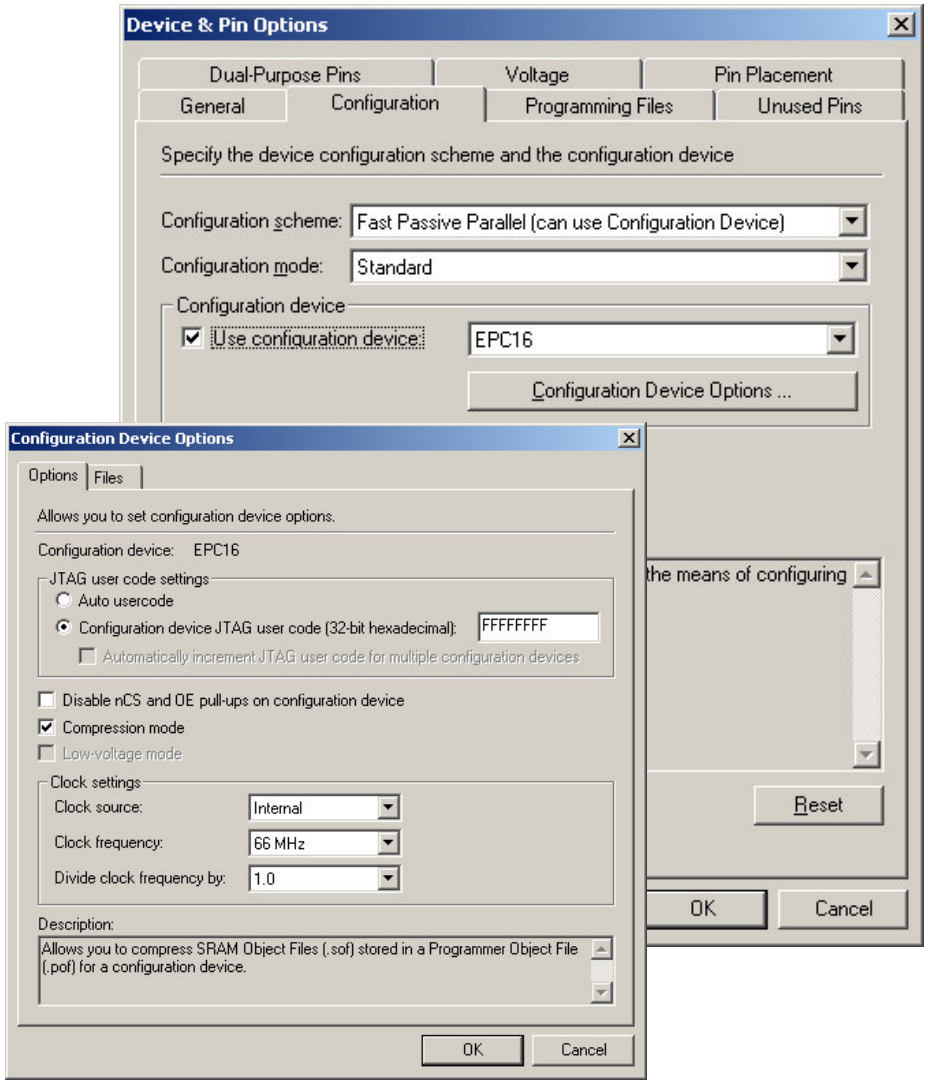
Device Options Control Configuration & Initialization of Device

General Tab



- Device Options Not Dependent on Configuration Scheme
  - Enable Device-Wide Clear
  - Enable Device-Wide Output Enable
  - Enable Initialization Done Output Pin

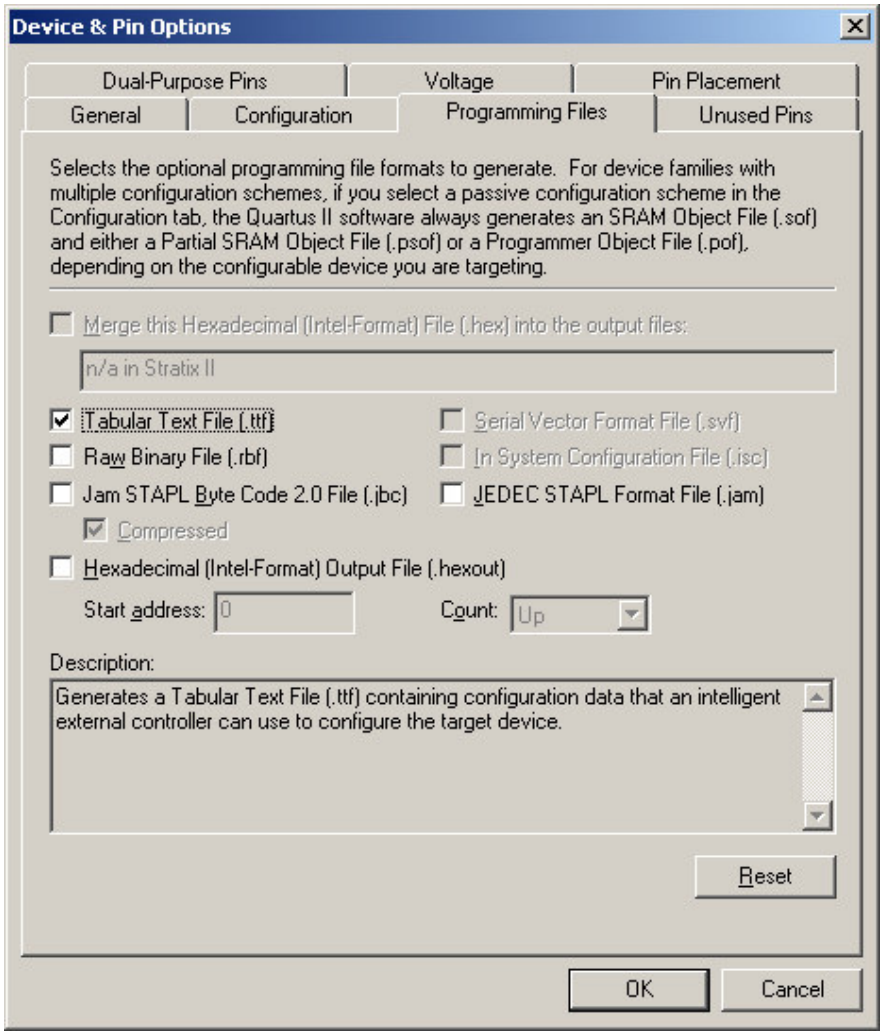
Configuration Tab



- Choose Device Configuration Mode & Available Options
  - Generates Correct Configuration & Programming Files Every Compilation
- Enables Special Features of Configuration Devices
  - Enable Programming File Compression
  - Set Configuration Clock Frequency

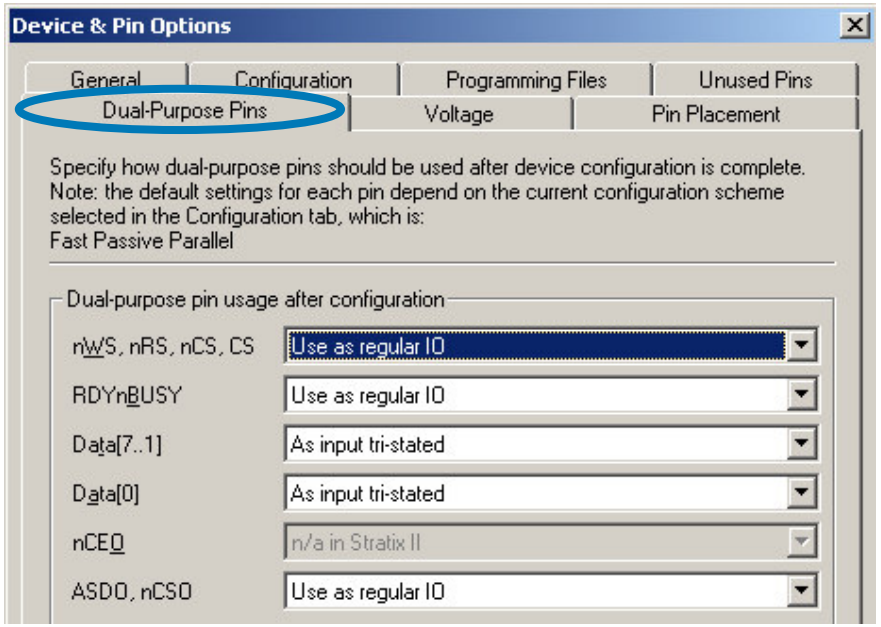
Programming Files Tab

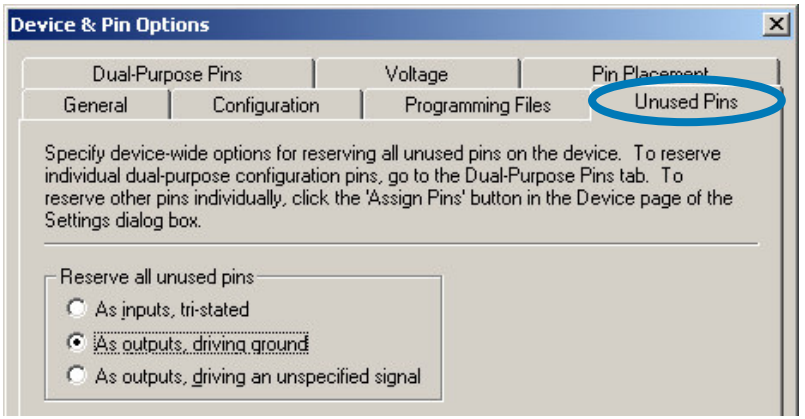
- Output Files Always Created
  - POF (Programming Object File)
  - SOF (SRAM Object File)
- Other Selectable Output Files
  - JAM (JEDEC STAPL)
  - JBC (JAM Byte-Code)
  - RBF (Raw Binary File)
  - HEXOUT (Intel Hex Format)



Unused & Dual-Purpose Pins Tabs

- Selects Usage of Dual-Purpose Pins after Configuration Is Complete
- Indicates State of All Unused I/O Pins after Configuration Is Complete

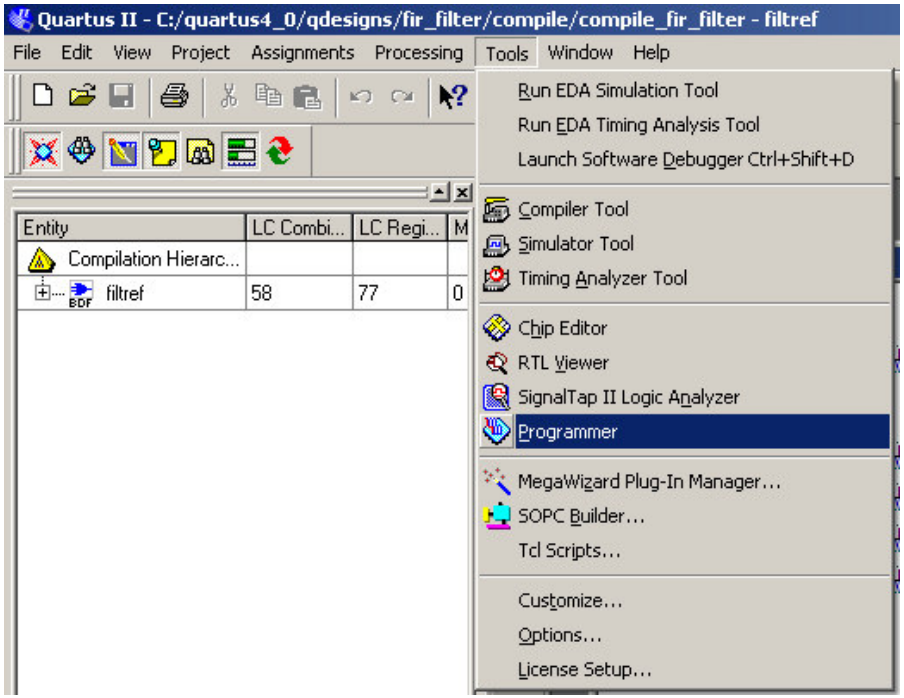




Quartus II Assembler Module

- 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

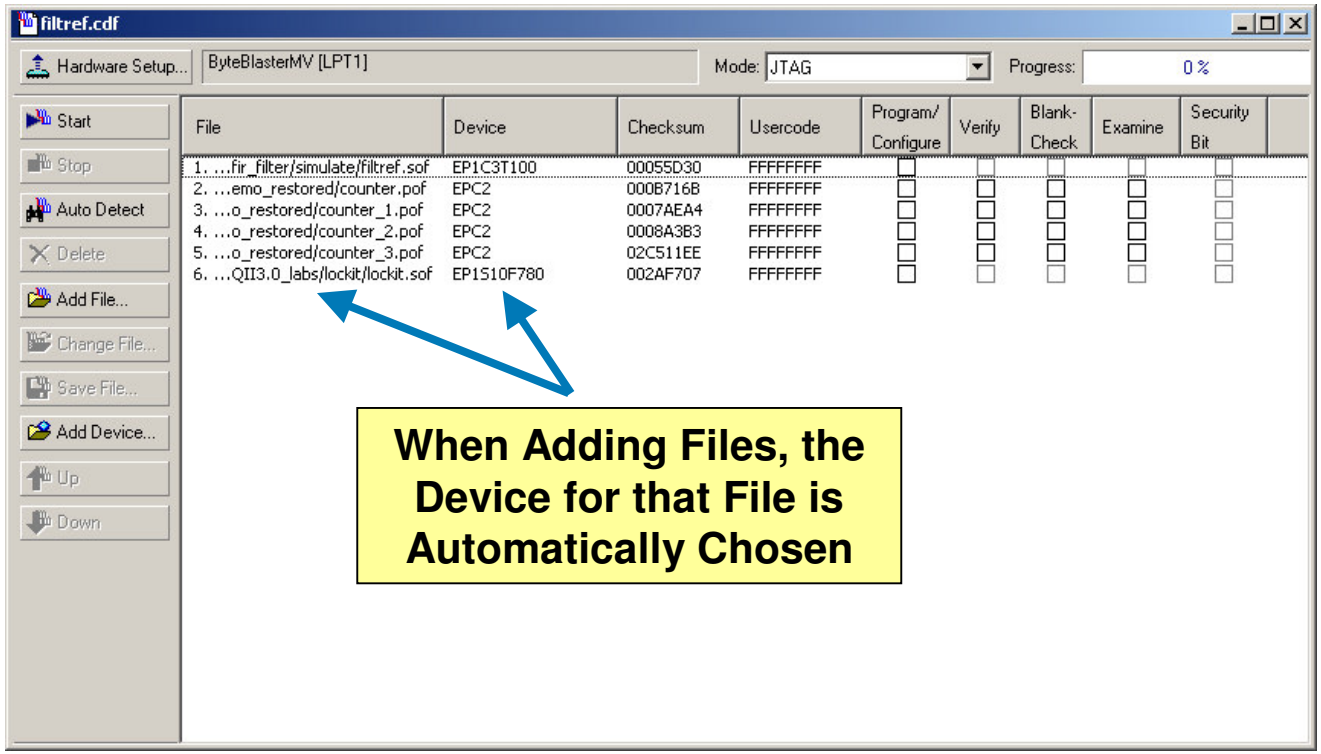
Open Programmer



- Enables Device Programming
  - ByteBlaster™ II or ByteBlasterMV™ Cables
  - USB-Blaster
  - MasterBlaster™ Cable
  - APU (Altera Programming Unit)
- Opens Chain Description File (.CDF)
  - Stores Device Programming Chain Information

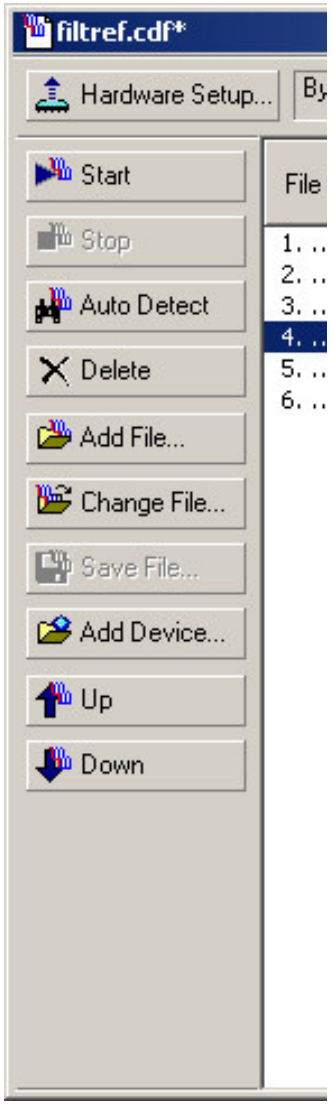
CDF File

- Lists Devices & Files for Programming or Configuration
- Programs/Configures in Top-to-Bottom Order



Programmer Toolbar

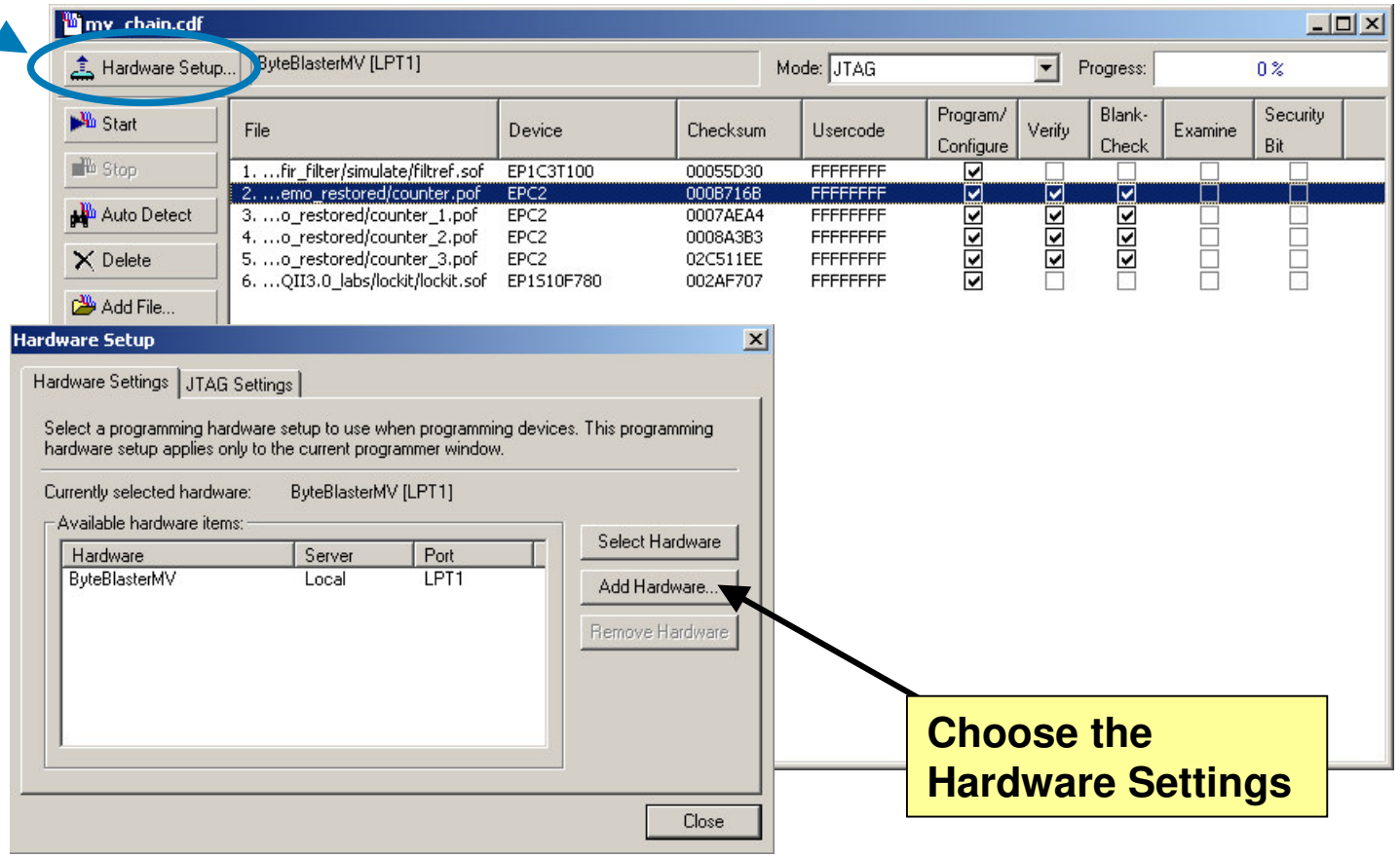
- Start Programming
- Auto Detect Devices in JTAG Chain
- Add/Remove/Change Devices in Chain
- Add/Remove/Changes Files in Chain
- Change Order of Files in Chain
- Setup Programming Hardware



Note: *All Options are available the Edit Menu except Start Programming & Auto Detect which are available in the Processing Menu*

Setting up Programming Hardware

Click on the Hardware Setup Button



Choose the Hardware Settings

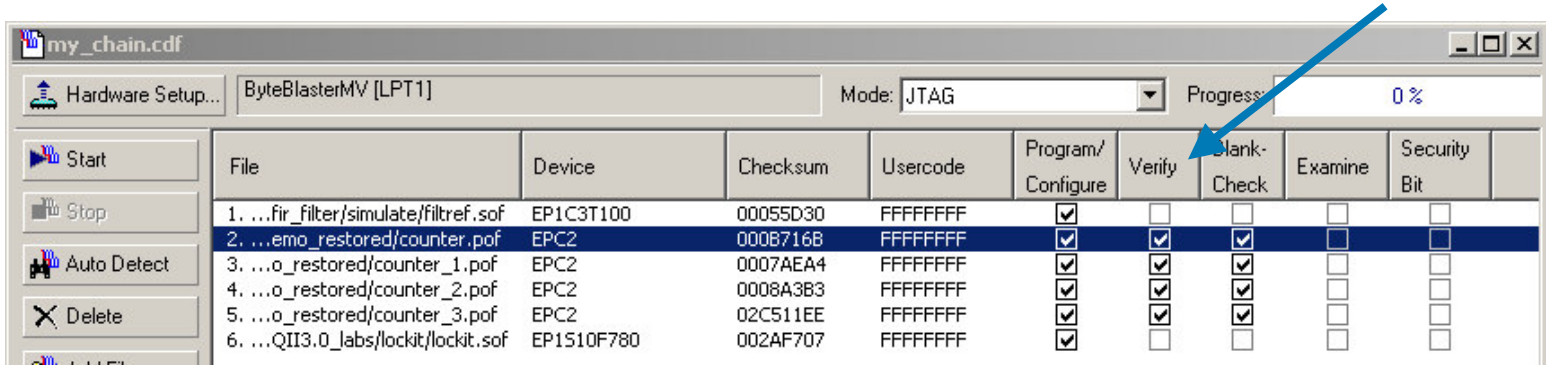
Chain Programming Modes

- Altera FPGAs, CPLDs & Non-Altera Devices
  - JTAG
- Altera FPGAs Only
  - Passive Serial
  - Active Serial
- CPLDs & Configuration Devices in APU
  - In-Socket

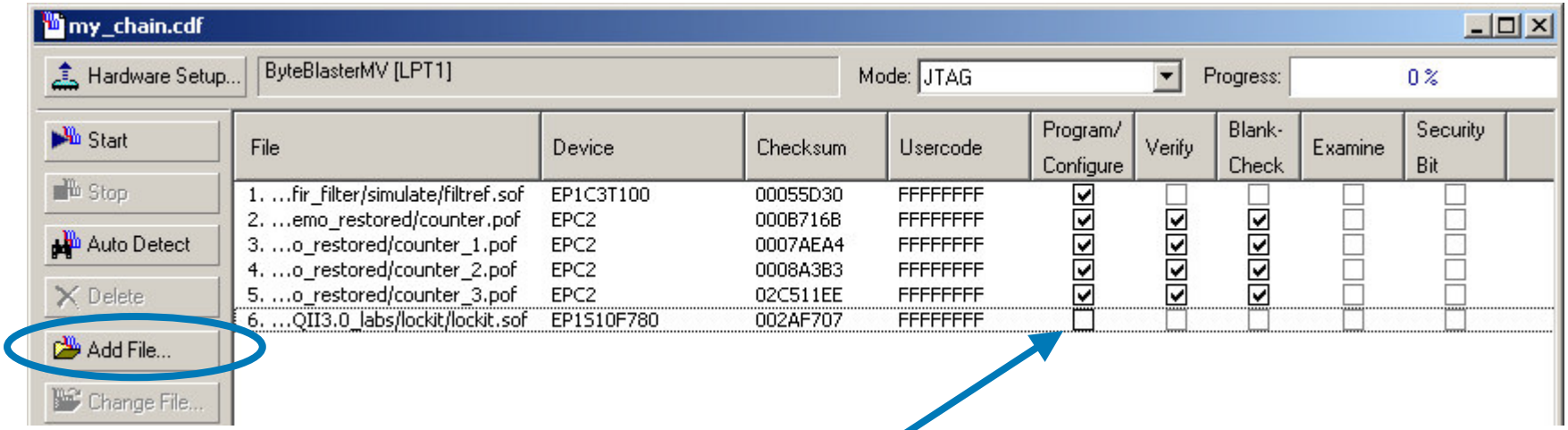
Programming Options

- Program/Configure
  - Applies to All Devices
- Verify, Blank-Check & Examine
  - Configuration Devices
  - MAX 7000 & MAX 3000
- Security Bit
  - MAX 7000 & MAX 3000

To Program, Verify, Blank-Check, or Examine the Device, Check the Appropriate Boxes

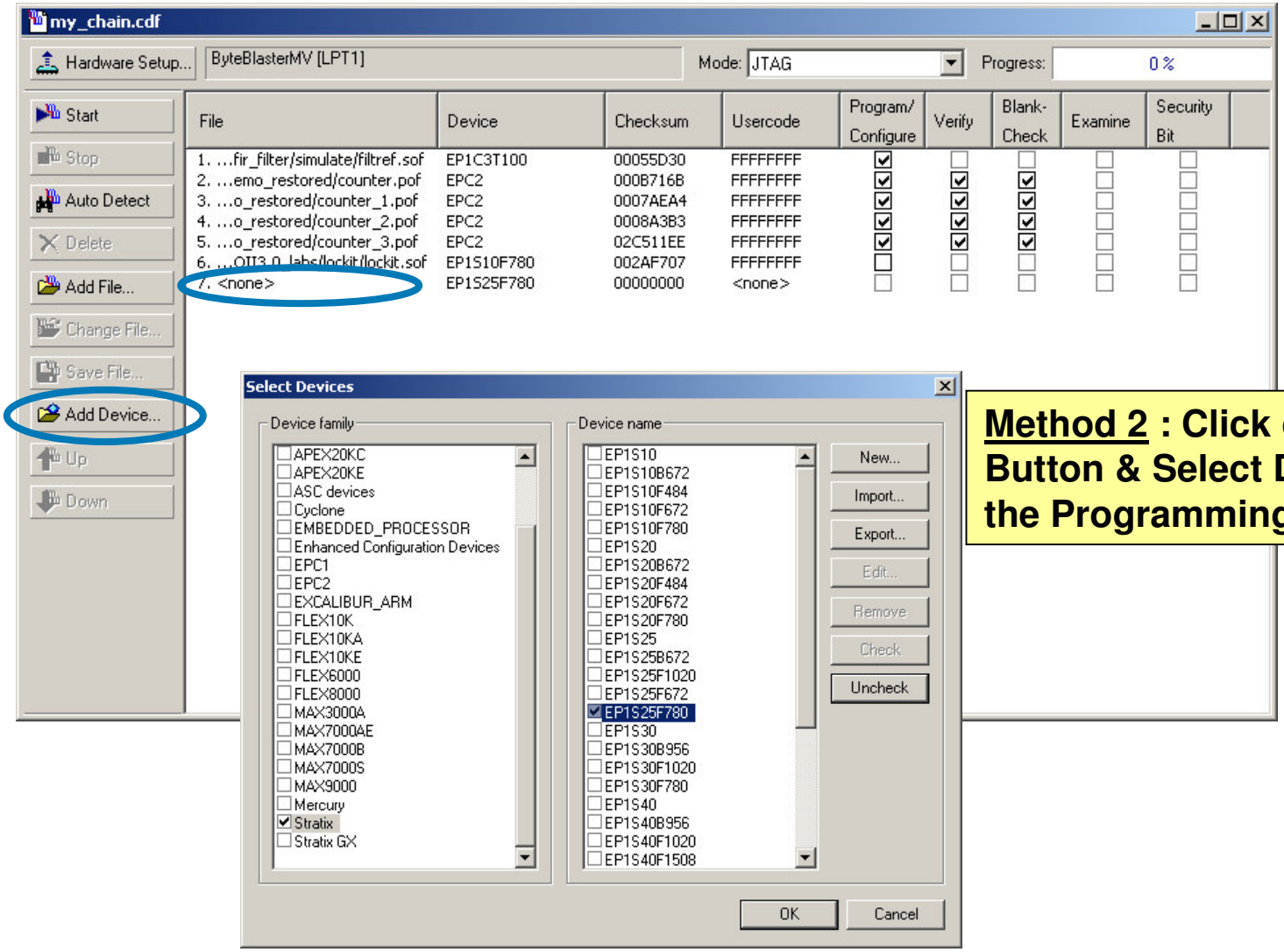


Bypassing Devices in JTAG Chain (1)



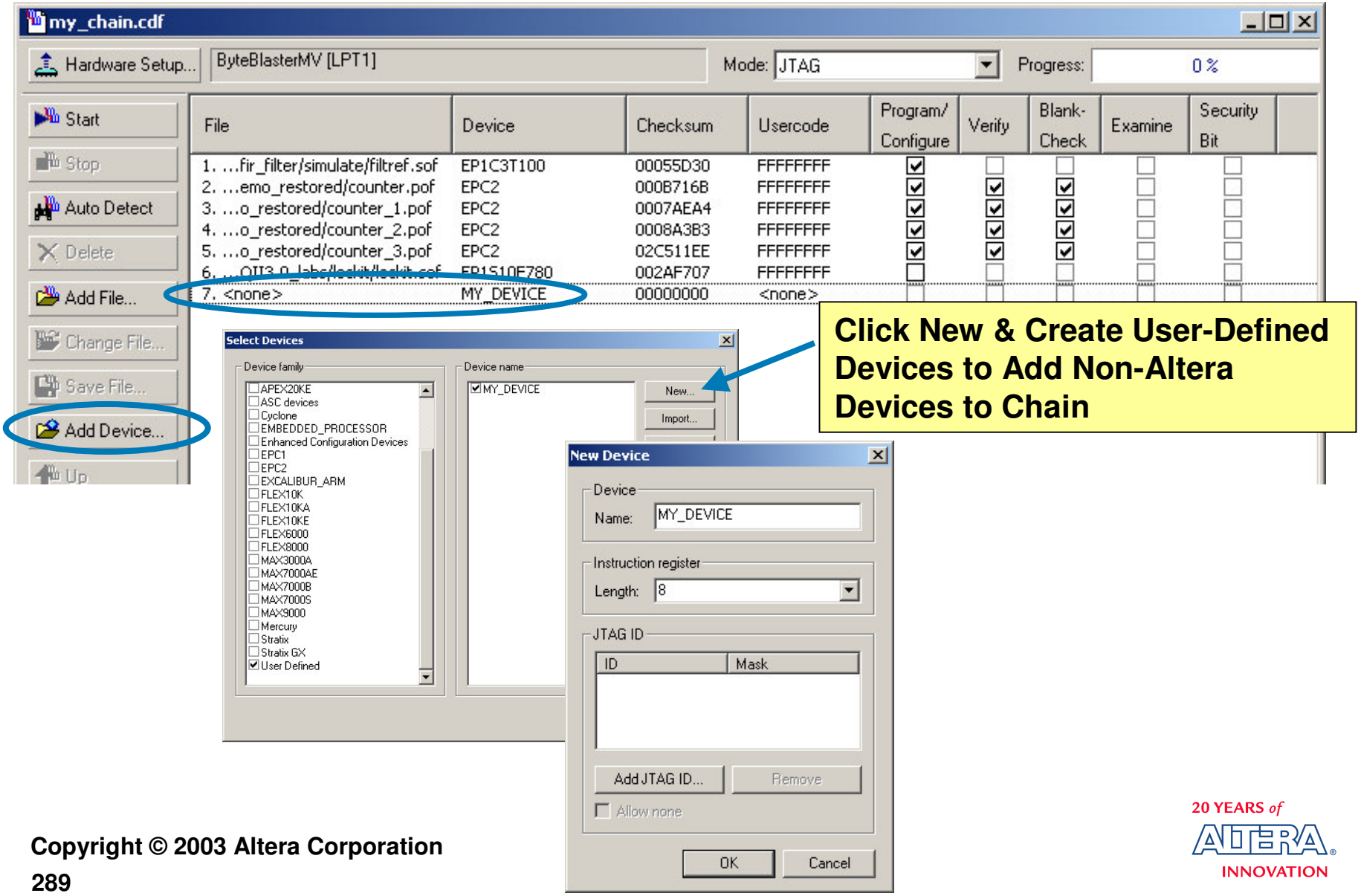
Method 1 : Add Programming File & Leave Program/Configure Box Unchecked

Bypassing Devices in JTAG Chain (2)



Method 2 : Click on Add Device Button & Select Device to Leave the Programming File Field Blank

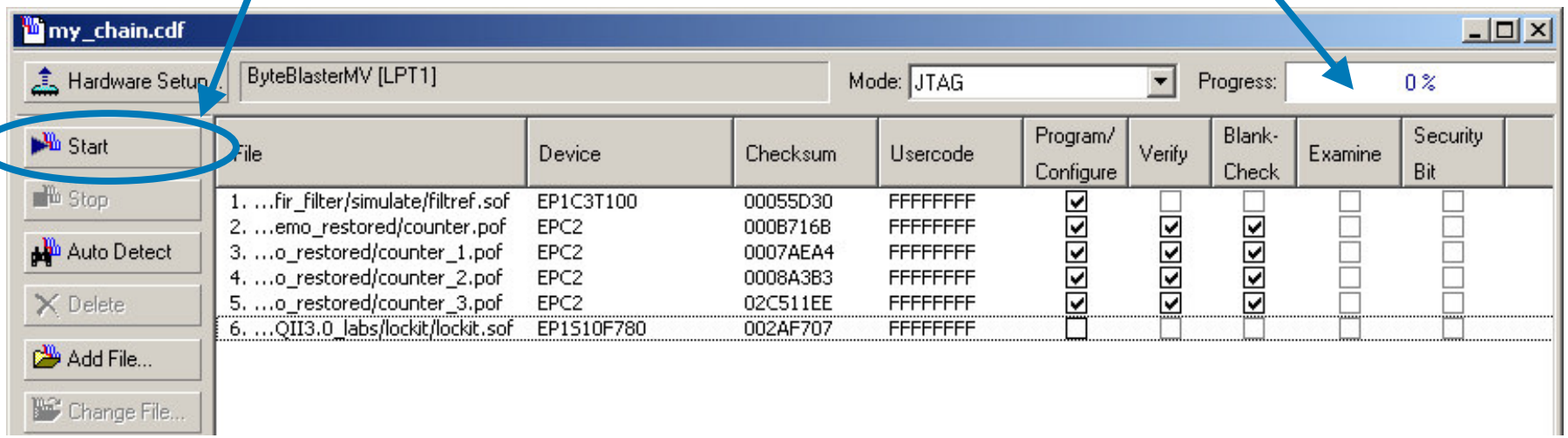
Adding Non-Altera Device to Chain



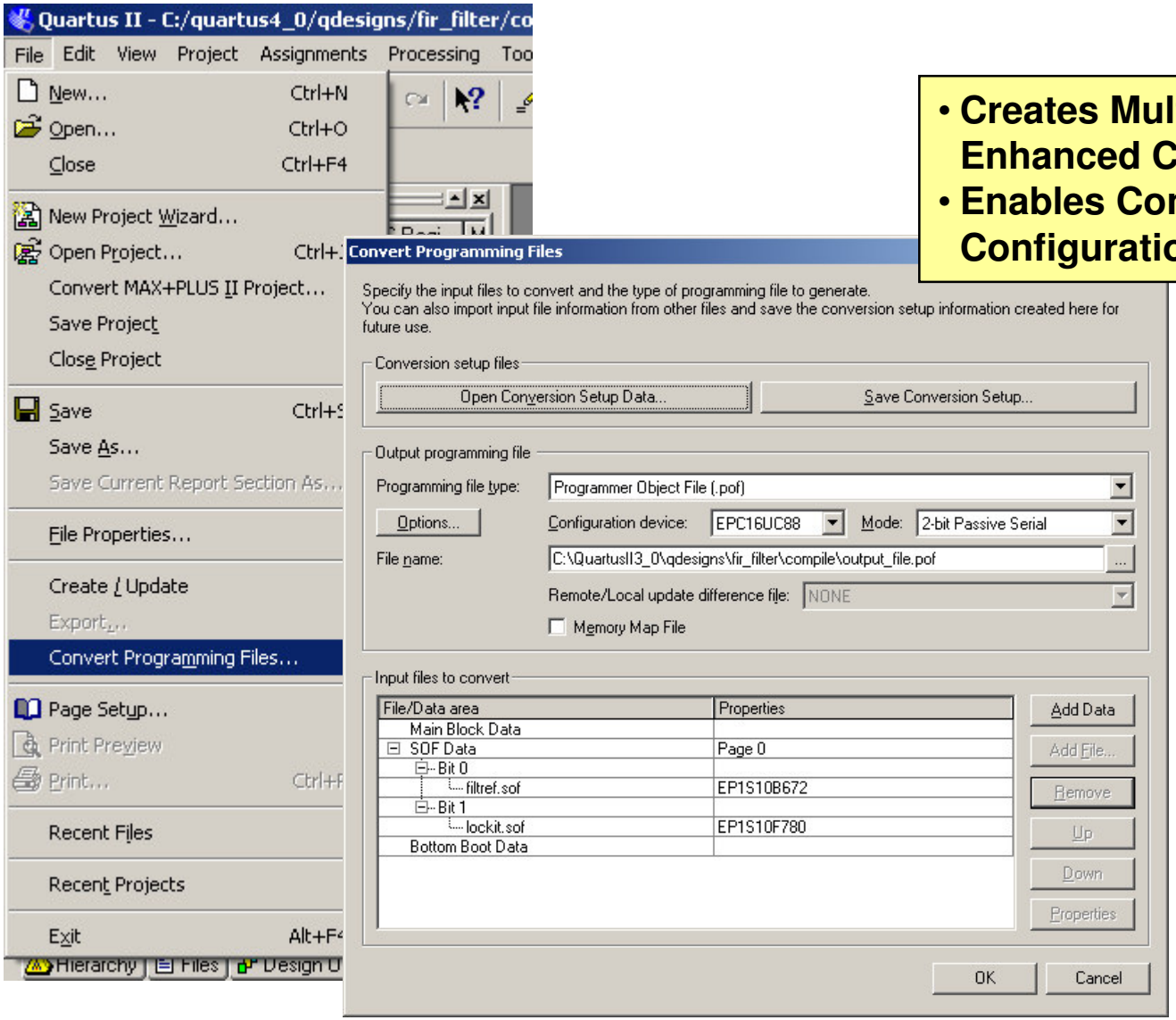
Starting the Programmer

Click Program Button Once CDF File & Hardware Setup Are Complete

Progress Field Shows the Percentage of Completion for the Programmer

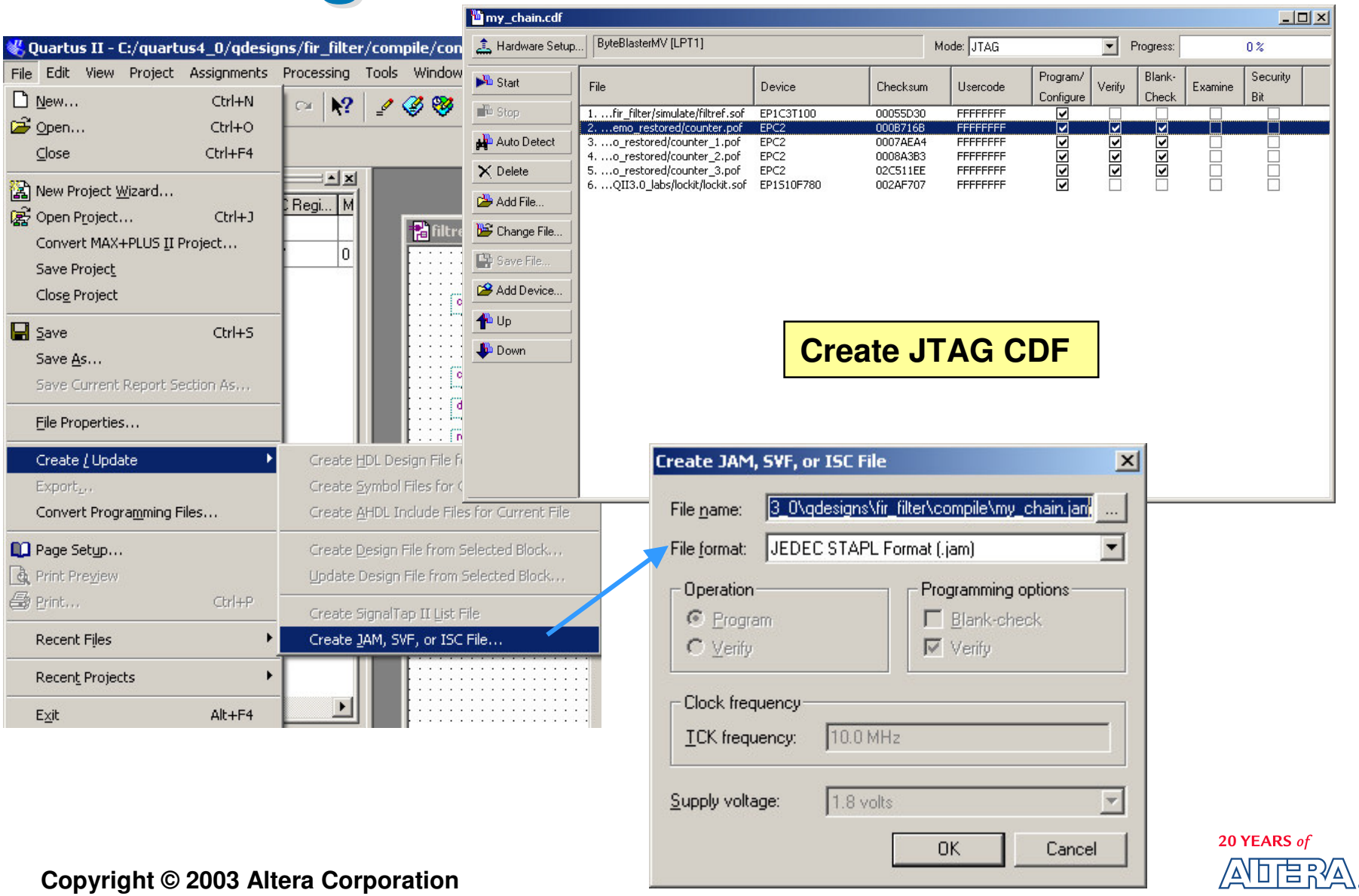


Converting SOF Programming Files



- **Creates Multi-Device .POF for Enhanced Configuration Devices**
- **Enables Compression & Other Configuration Device Options**

Creating Jam™ STAPL



Exercise 6 or 7 (Optional)

Please go to Exercise 6 or 7 in the Exercise Manual

Programming/Configuration Summary

- Setting Device Options
- Generating Programming Files
- Programming Device or Devices in Chain
- Converting Programming Files

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 day in length (8 working hours).

On-Line Training

With Altera's on-line 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 on-line course will take approximately 2-3 hours to complete.

www.altera.com/training

View Training Class Schedule & Register for a Class

Advanced Quartus II Course

- Accerating Design Cycles Using Quartus II
  - LogicLock
  - Chip Editor
  - FPGA Optimization
  - SignalTap II & SignalProbe
  - Command-Line & Tcl Scripts
  - HardCopy Software Support

Altera Technical Support

- Reference Quartus II On-Line Help
- 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