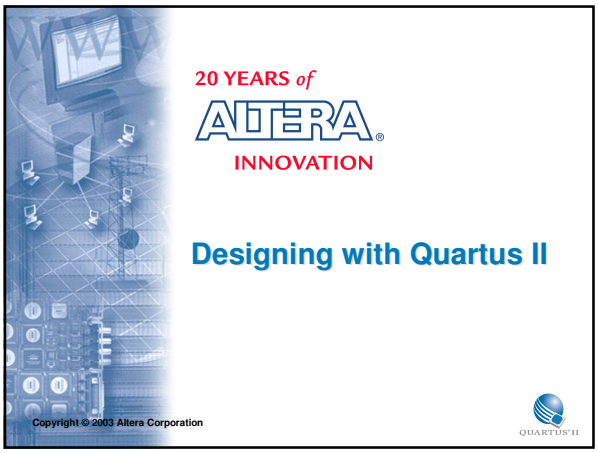
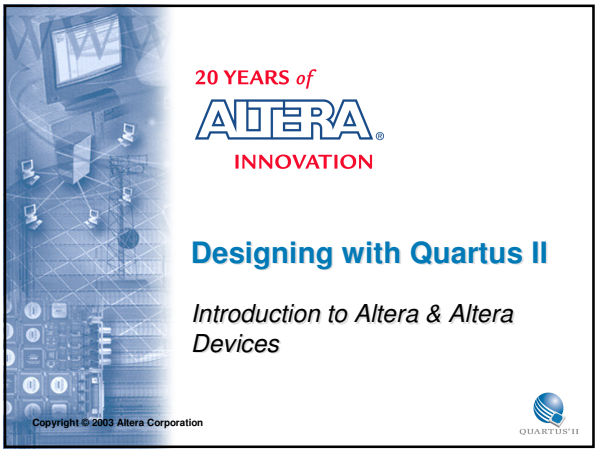


Class Agenda

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



The Programmable Solutions Company®

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

4

Introduction to Altera Devices

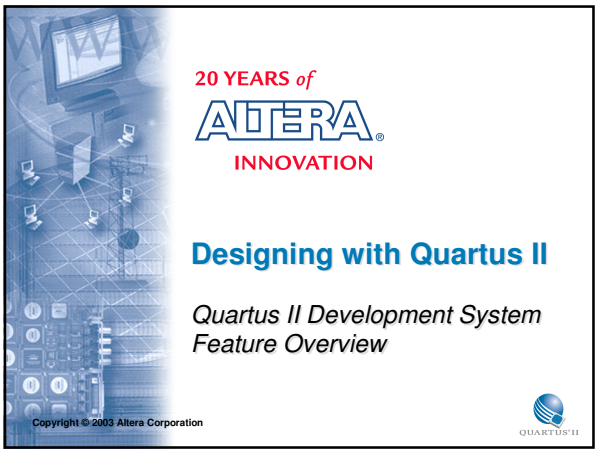
- Programmable Logic Families
  - High & Medium Density FPGAs
    - Stratix™ II, Stratix, APEX™ II, APEX 20K, & FLEX® 10K
  - Low-Cost FPGAs
    - Cyclone™ & ACEX® 1K
  - FPGAs with Clock Data Recovery
    - Stratix GX & Mercury™
  - CPLDs
    - MAX® 7000 & MAX 3000
  - Embedded Processor Solutions
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

5

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

6



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

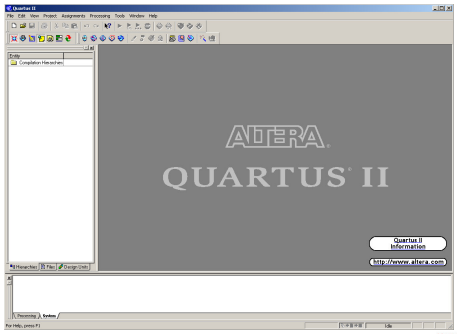
9

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

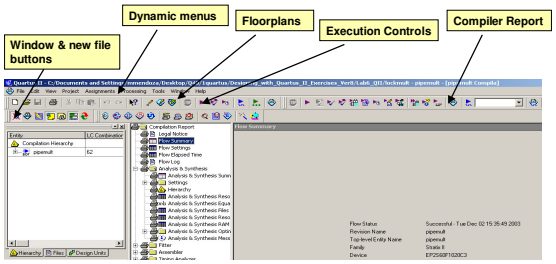
10

Quartus II Operating Environment



11

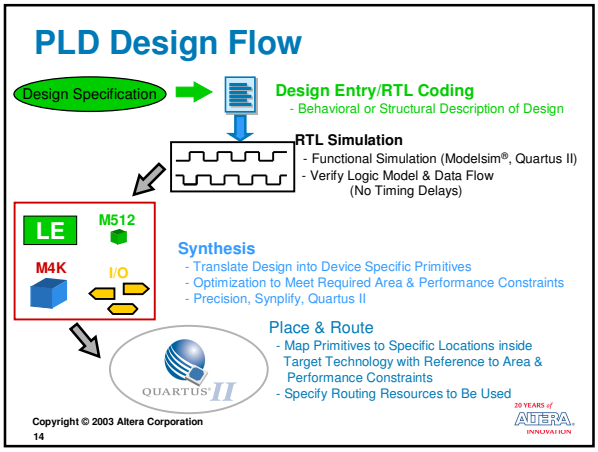
Main Toolbar & Modes

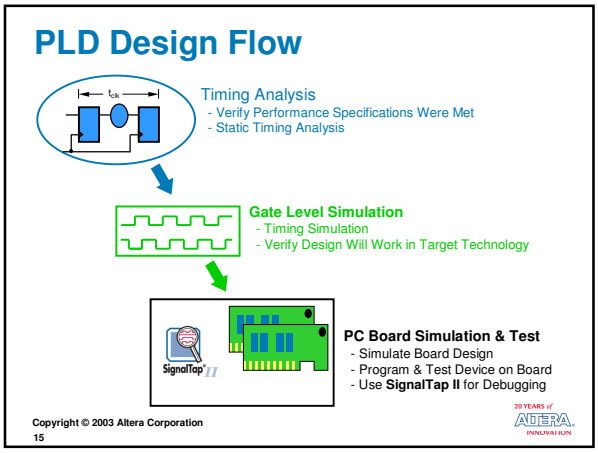


To Reset Views: Tools ⇒Toolbars>Reset All; Restart Quartus II

12


20 YEARS of
ALTERA
INNOVATION
Designing with Quartus II
Design Methodology
Copyright © 2003 Altera Corporation
QUARTUS II


PLD Design Flow
Design Specification
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)
LE
M512
M4K
I/O
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
QUARTUS II
Copyright © 2003 Altera Corporation
14


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
SignalTap II
PC Board Simulation & Test
- Simulate Board Design
- Program & Test Device on Board
- Use SignalTap II for Debugging
Copyright © 2003 Altera Corporation
15

20 YEARS of
ALTERA
INNOVATION

Designing with Quartus II

Design Entry

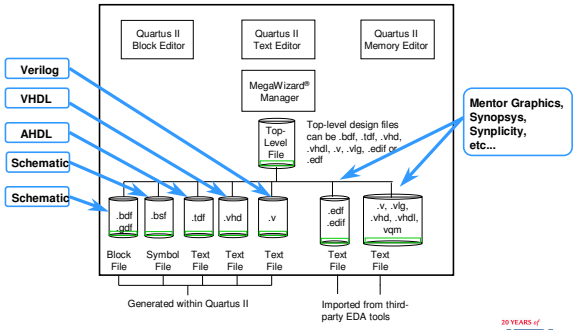
QUARTUS II

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

17

Design Entry Files



18

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)

19

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

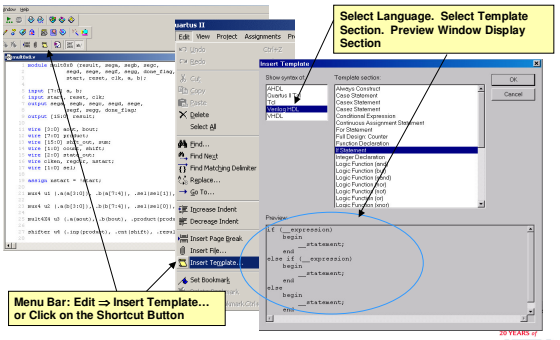
20

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

21

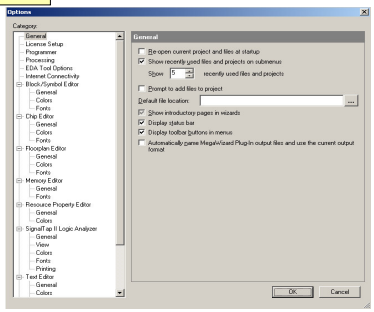
HDL Templates



22

Quartus II Options

Tools ⇒ Options...



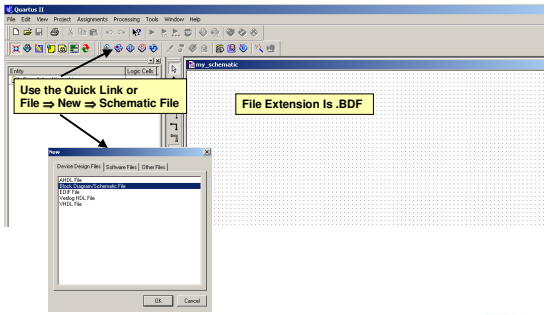
23

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

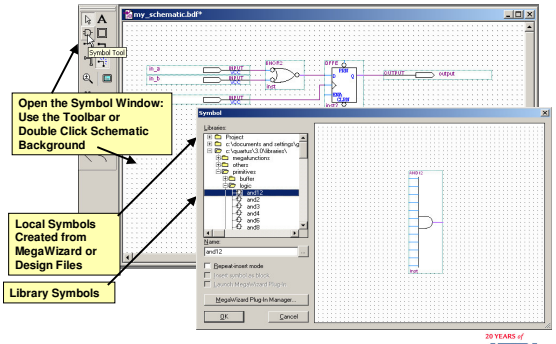
24

Create Schematic



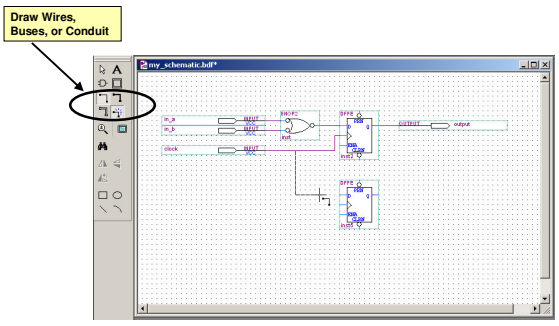
25

Insert Symbols



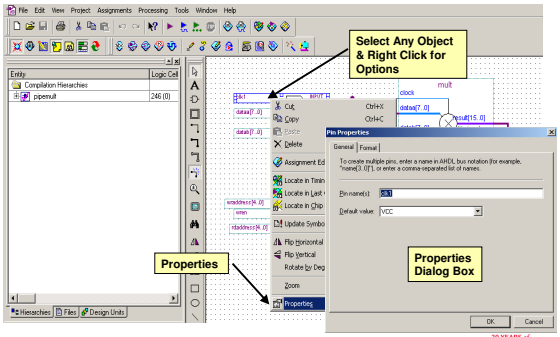
26

Connect Wires & Buses



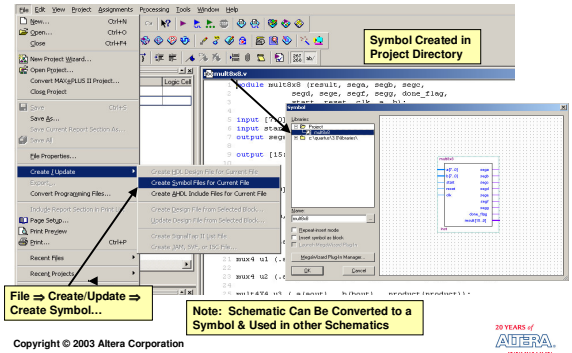
27

Change Names & Properties



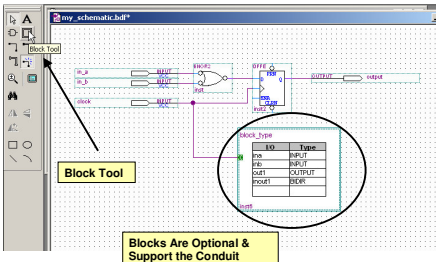
28

Create Symbols



29

Optional Block Design Entity



30

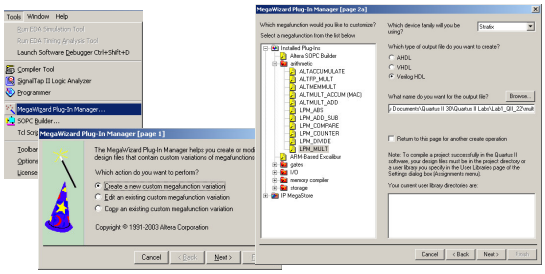
Megafunctions

- 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

31

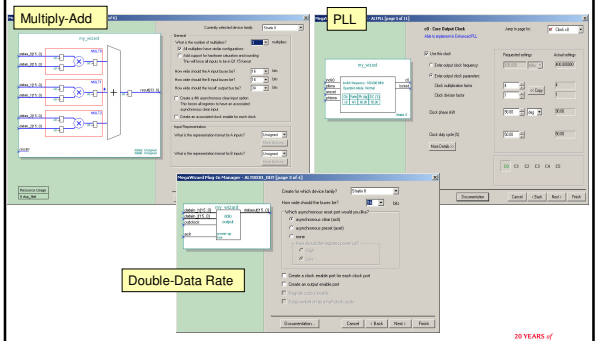
MegaWizard Plug-In Manager

- Eases Implementation of Megafunctions & IP



32

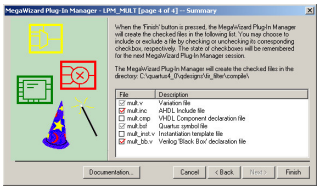
MegaWizard Examples



33

MegaWizard Output File Selection

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



34

Memory Editor

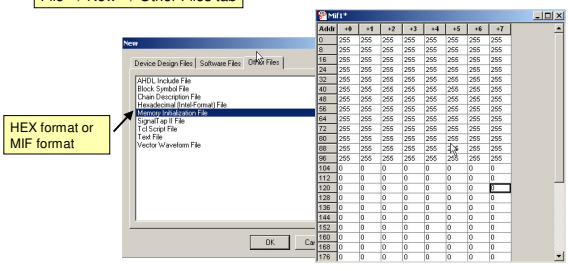
- Create or Edit Memory Initialization Files in Intel Hex (.HEX) or Altera-Specific (.MIF) Format
- Design Entry
  - Use to Initialize Your Memory Block (Ex. RAM, ROM) during Power-Up
- Simulation
  - Use to Initialize Memory Blocks before Simulation

35

Create Memory Initialization File

File ⇒ New ⇒ Other Files tab

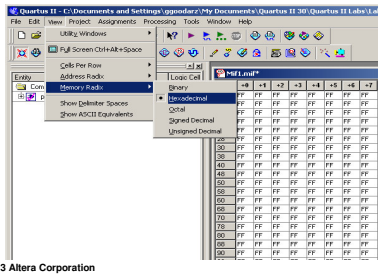
HEX format or MIF format



36

Change Options

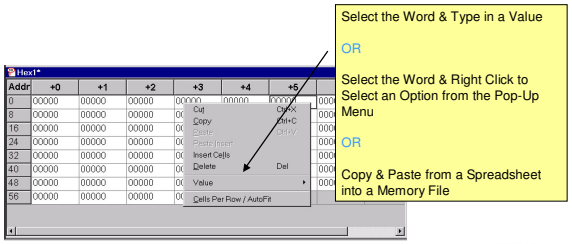
- View Options of Memory Editor
  - View ⇒ Select from Available Options



37

Edit Contents

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



38

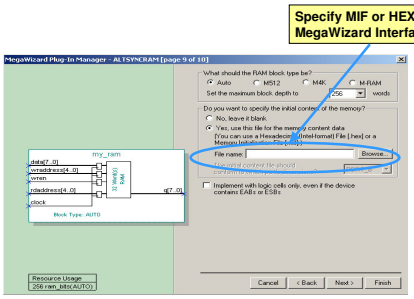
Memory Size Wizard

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

39

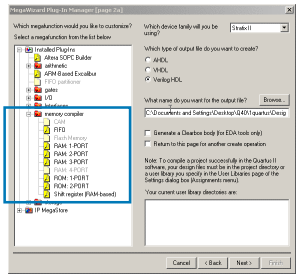
Using Memory File in Design



40

Memory Compiler

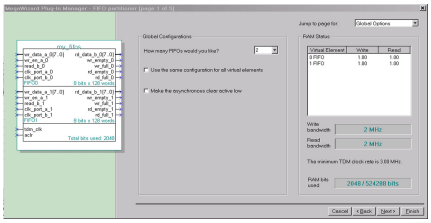
- Memory Functions are Organized by Type
- MegaWizard is Family-Aware
  - Select Memory Type & Parameters
  - Appropriate Megafunction Selected Automatically



41

FIFO Partitioner

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



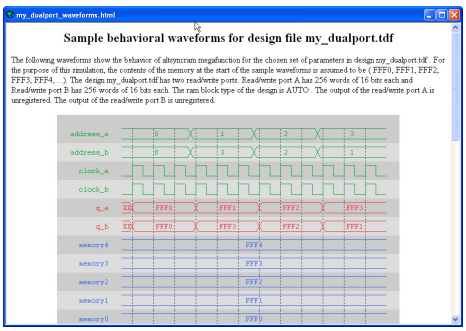
42

Memory Behavioral Waveforms

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

43

Example Memory Waveform



44

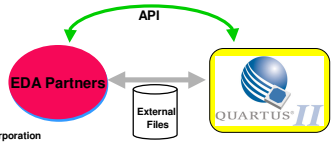
EDA Interfaces Introduction

- 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

45

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



46

WYSIWYG ATOM Primitives

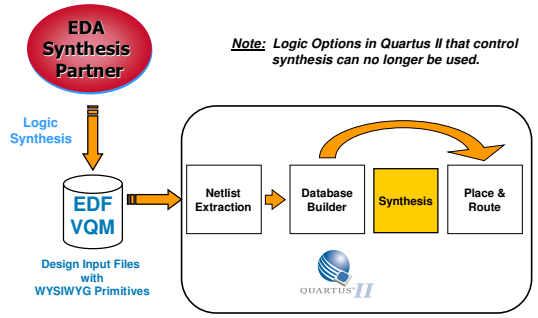
- Set of Design Primitives that Support WYSIWYG Compilation
- Provide Direct Control of How a Design Is Technology-Mapped to a Specific Target Device
- Allow Synthesis Vendors to Provide an Optimal Realization of a Design for Each Architecture

```
00104   stratix_lcell par_err_high_0_and2_62_2 (
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_2.operation_mode="normal";
00112 defparam par_err_high_0_and2_62_2.lut_mask="9669";
00113 defparam par_err_high_0_and2_62_2.synch_mode="off";
00114 defparam par_err_high_0_and2_62_2.sum_lutc_input="datac";
```

Synplify .vqm

47

WYSIWYG Compilation Flow



48

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

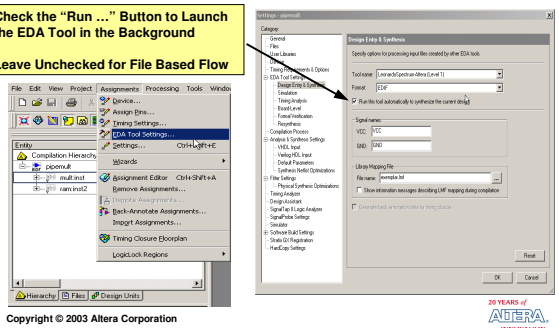
49

Quartus II Driven Flow

Assignments ⇒ EDA Tool Settings

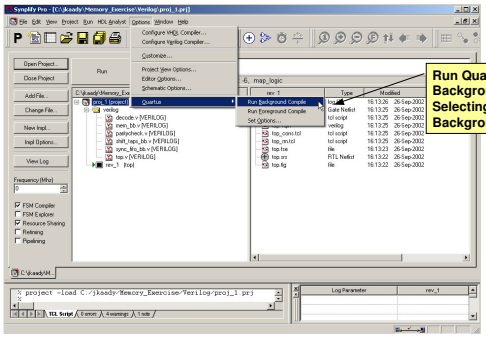
Check the "Run ..." Button to Launch the EDA Tool in the Background

Leave Unchecked for File Based Flow



50

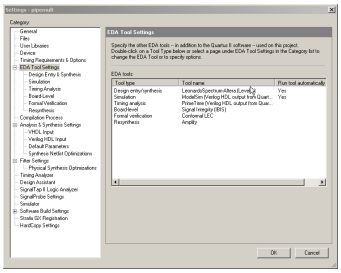
EDA Driven Flow



51

File Based Flow

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



52

Third Party Tool Support

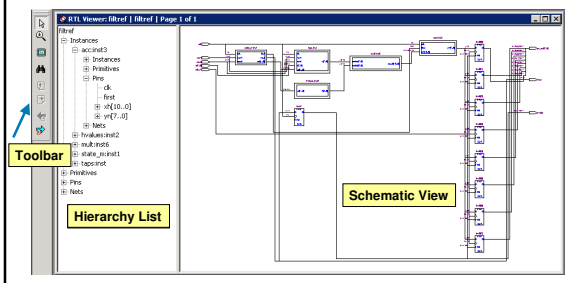
Synthesis Tools
- LeonardoSpectrum™
- Precision
- DesignCompiler
- FPGA Compiler II™
- FPGA Express™
- Synplify
- Synplify Pro
- Amplify

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

53

RTL Viewer

- Graphically Represents Results of Synthesis



54

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

55

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

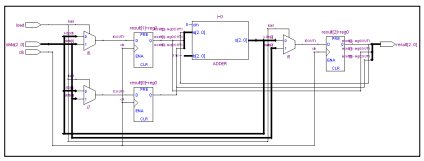
56

RTL Viewer Features

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

57

Schematic View



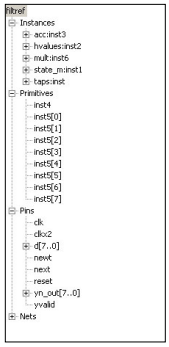
- Represents Design Using Logic Blocks & Nets
  - Registers
  - Muxes
  - Gates
  - Adders

Hierarchy List

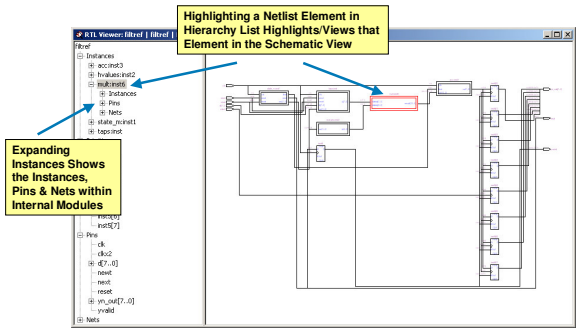
- Traverses between Design Hierarchy
- Views Logic Schematic for Each Hierarchical Level
- Breaks down Design into Netlist Elements

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



61

Hierarchy Navigation

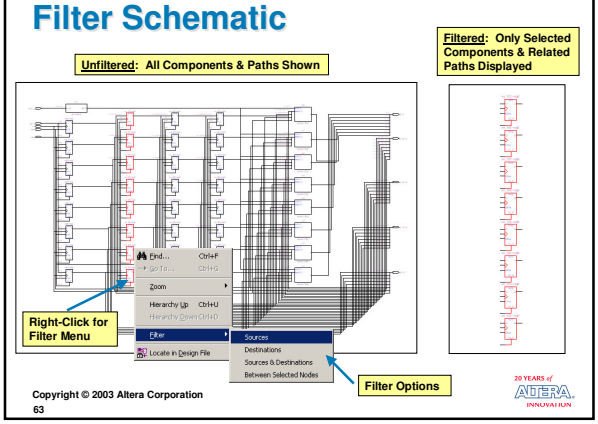
- Hierarchy List
  1. Click on Netlist Element (Ex. Instance)
  2. Schematic View Automatically Adjusts Hierarchical Level to Show Element
- Schematic View
  - Mouse Pointer Indicates Action

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

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

62

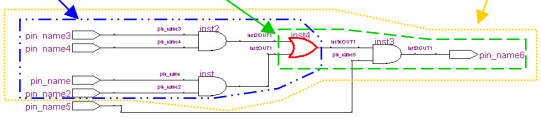
Filter Schematic



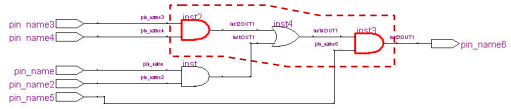
63

Filter Options

- Sources, Destinations, Sources & Destinations



- Between Selected Nodes



64

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)

65

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)

66

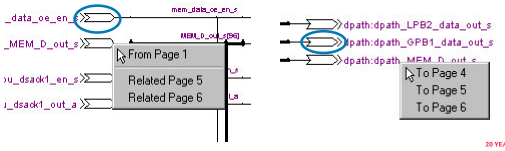
Managing Pages

- 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

70

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

71

Summary

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

72

20 YEARS of ALTERA INNOVATION

Designing with Quartus II

Quartus II Projects

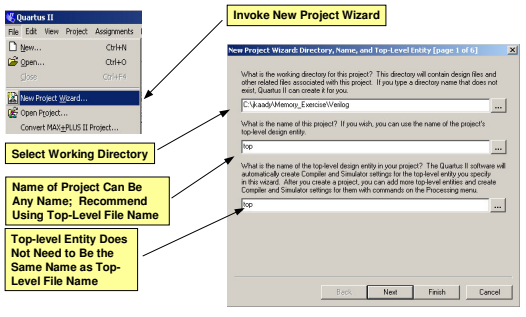
QUARTUS II

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

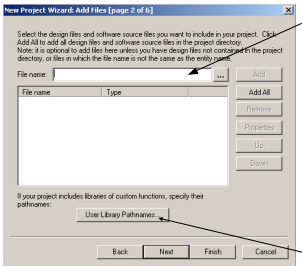
74

New Project Wizard



75

Add Files



Add Design Files

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

Notes:

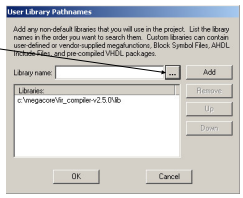
- ***Files in project directory do not need to be added***
- ***Add top level file if filename & entity name are not the same***

Add User Library Pathnames & Files

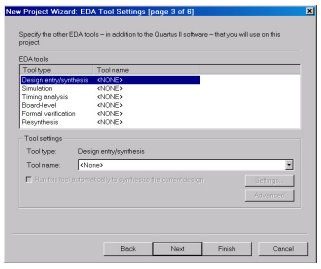
User Libraries

Add User Library Paths

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



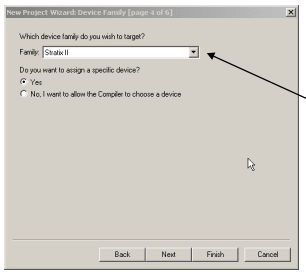
EDA Tool Settings



Choose EDA Tools

Add or Change Settings Later

Device Selection

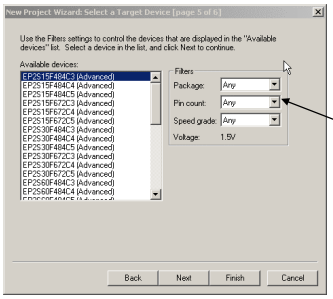


Choose the Device Family

Auto Device or Choose a Specific Device

79

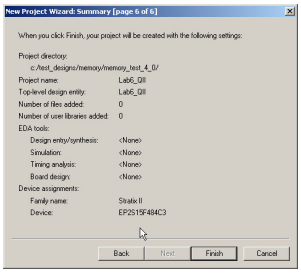
Device Selection



Use Filters for Auto Selection or Picking a Specific Device

80

Done!

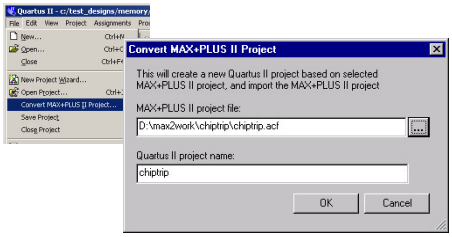


Review Results & Click on Finish

81

MAX+PLUS II to Quartus II

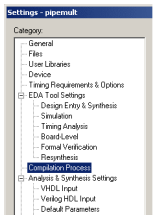
- Convert MAX+PLUS II Projects into Quartus II Projects
- Assignments Automatically Translated

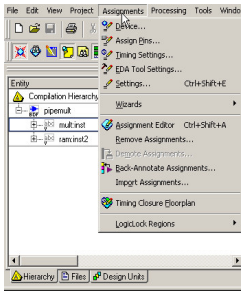


82

Assignments Menu

- Accesses All Assignments & Settings
- Opens Settings Dialog Box

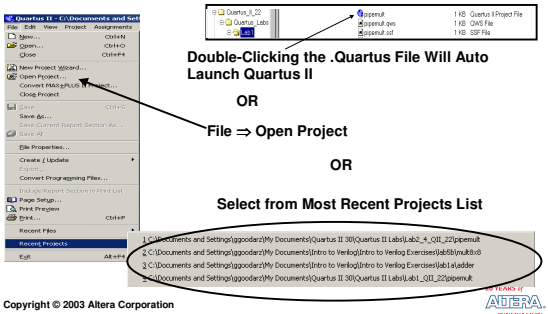




83

Editing Project Settings

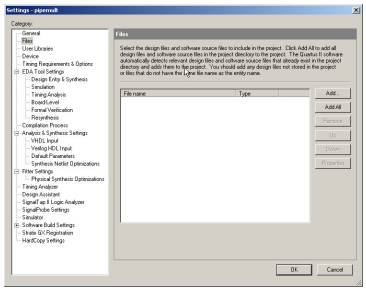
- Open Existing Project First to Edit Settings



84

Editing Project Settings (cont.)

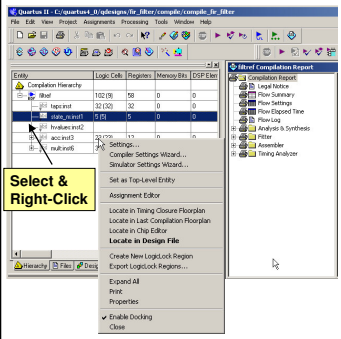
- Assignments ⇒ Settings



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

85

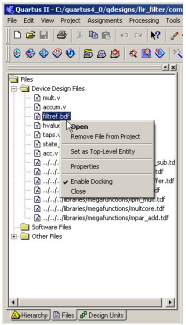
Project Navigator – Hierarchy Tab



- 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

86

Project Navigator – Files Tab

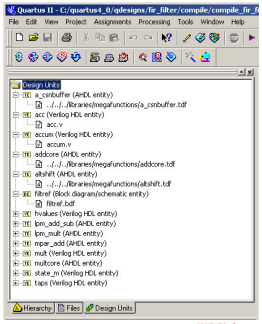


- Shows All Files in Project
  - All Source Files Appear under Design Files
- Uses
  - Open Files
  - Remove Files from Project
  - Set New Top-Level Entity

87

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



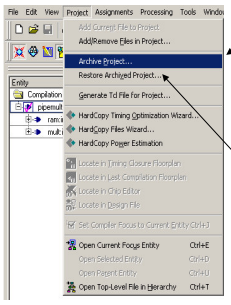
88

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

89

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

90

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

91

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

93

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

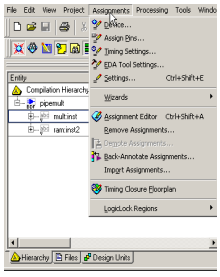
94

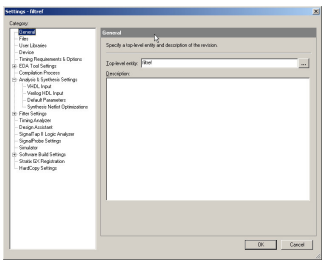
Settings

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

95

Editing Settings



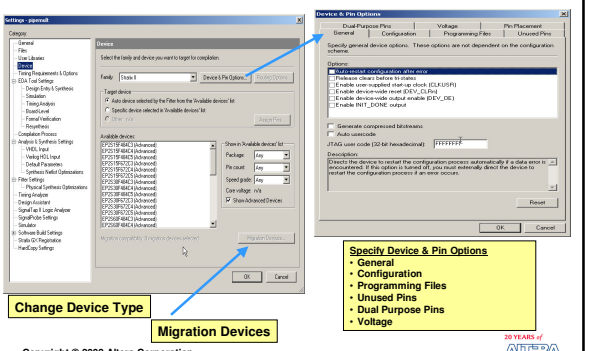


Assignments ⇒ Settings

Several Options Control Compilation & Device Options

96

Device Selection



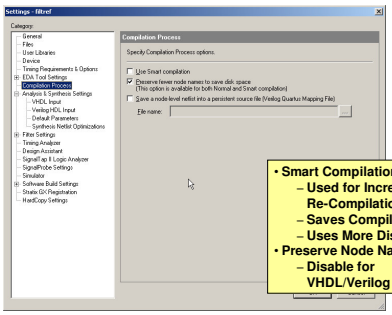
Change Device Type

Migration Devices

Specify Device & Pin Options
- General
- Configuration
- Programming Files
- Unused Pins
- Dual Purpose Pins
- Voltage

97

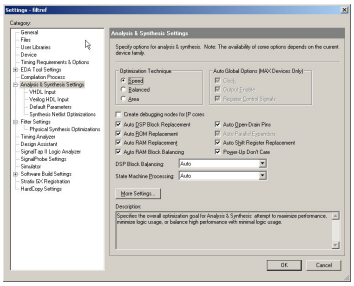
Compilation Process



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

98

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

99

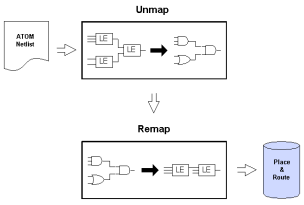
Synthesis Netlist Optimizations

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

100

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

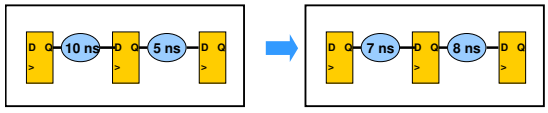


4-5 % Improvement

101

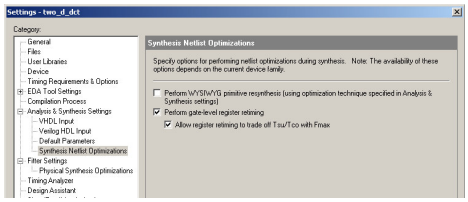
Gate-Level Register Retiming

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



102

Enabling Synthesis Optimizations



Created/Modified Nodes Noted in Compilation Report

103

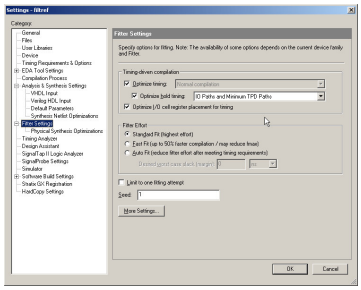
20 YEARS of ALTERA INNOVATION

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**



104

20 YEARS of ALTERA INNOVATION

Physical Synthesis

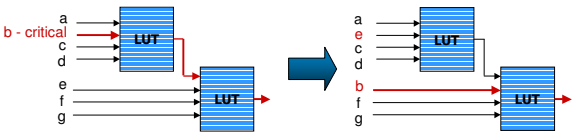
- Re-Synthesis Based on Fitter Output
  - Makes Incremental Changes that Improve Results for a Given Placement in Altera Device
  - Compensates for Routing Delays from Fitter
    - Routing Delays Large Part of Typical Critical Path Delay
- Types
  - Combinational Logic
  - Registers
    - Register Duplication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

105

20 YEARS of ALTERA INNOVATION

Combinational Logic

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

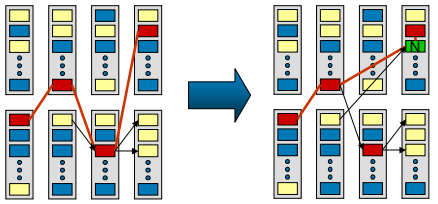


- Allows LUT Duplication to Enable Further Optimizations on the Critical Path

106

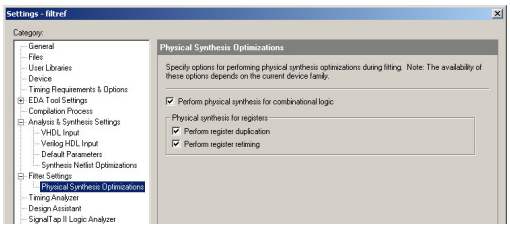
Register Duplication

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



107

Enabling Physical Synthesis



Created/Modified Nodes Noted in Compilation Report

108

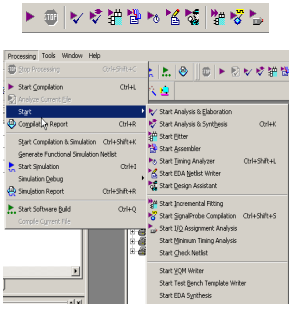
More Netlist Optimizations Info

- 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

109

Processing Options

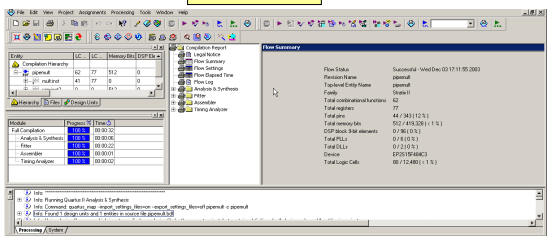
- Start Compilation
  - Perform Full Compilation
- Start Analysis & Elaboration
  - Check Syntax & Build Database Only
- Start Analysis & Synthesis
  - Synthesize Code & Estimate Timing
- Start Fitter
- Start Assembler
- Start Timing Analysis
- Start I/O Assignment Analysis
- Start Design Assistant



110

Status & Message Windows

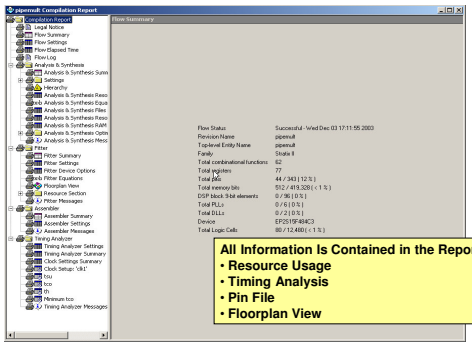
Hit the Start Button



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

111

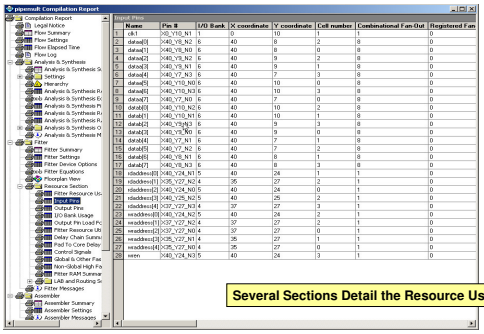
Compilation Report



All Information Is Contained in the Report
- Resource Usage
- Timing Analysis
- Pin File
- Floorplan View

112

Resource Usage



Several Sections Detail the Resource Usage

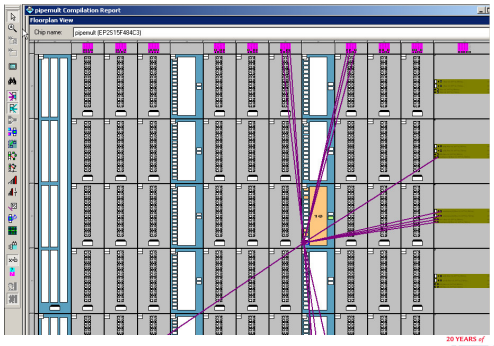
113

Floorplan

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

114

Floorplan View



115

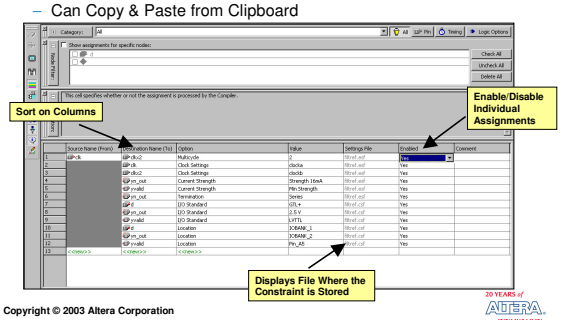
Assignments

- Assignment Editor
- Example Assignments
- I/O Assignments & Analysis
- Perform Analysis & Elaboration before Obtaining Hierarchy & Node Information

116

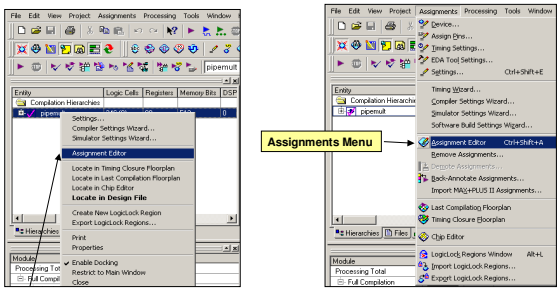
Assignment Editor (AE)

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



117

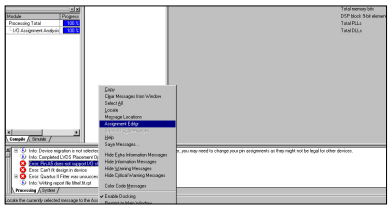
Opening Assignment Editor



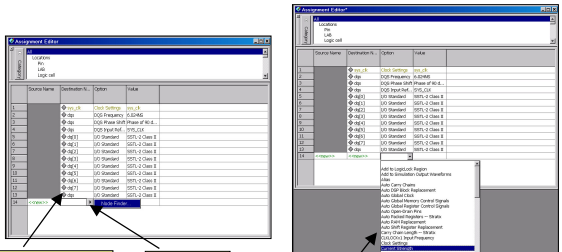
Invoke the Assignment Editor by Highlighting an Entity in the Hierarchy View & Right- Clicking

Opening Assignment Editor (cont.)

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

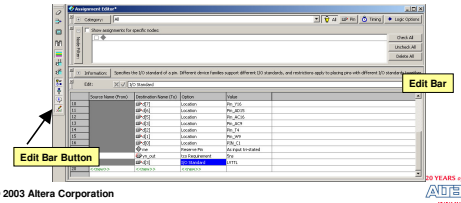


Using Assignment Editor



Entering Assignments

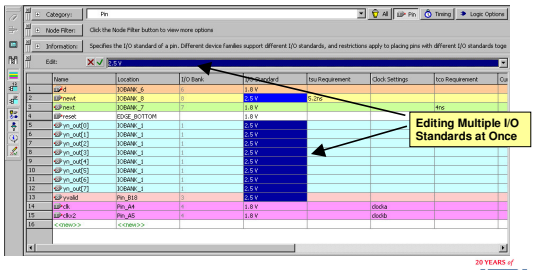
- Other Ways to Edit Cells
  - Single-Click & Type Name Directly
    - Useful for Known Node Names
  - Edit Menu
  - Edit Bar



121

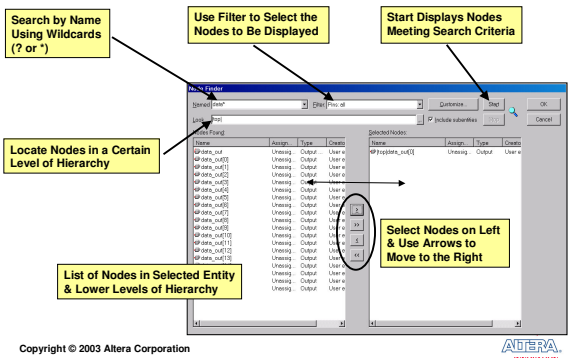
Editing Multiple Assignments

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



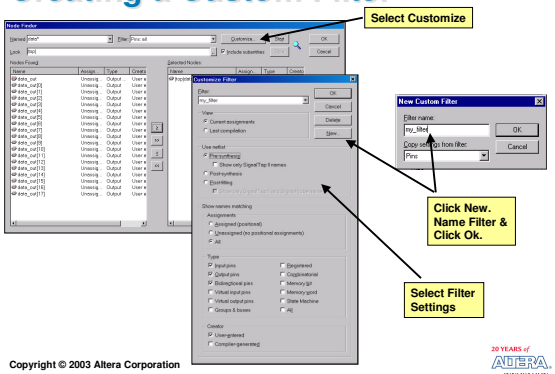
122

Node Finder



123

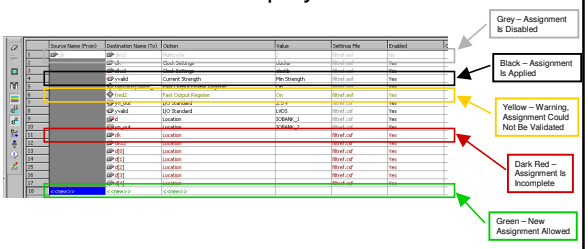
Creating a Custom Filter



124

AE Dynamic Checking

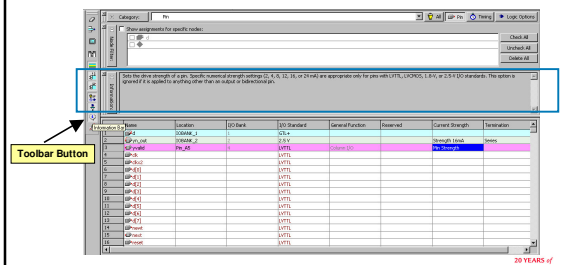
- Validity of Constraint Checked during Entry
- Color-Coded to Display Status



125

AE Information Bar

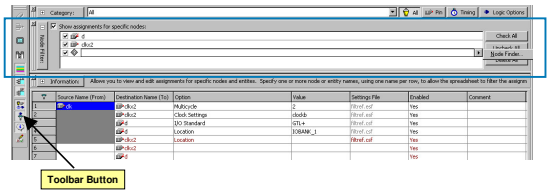
- Displays Information on Selected Cell
- Can Be Hidden or Minimized



126

AE Node Filter Bar

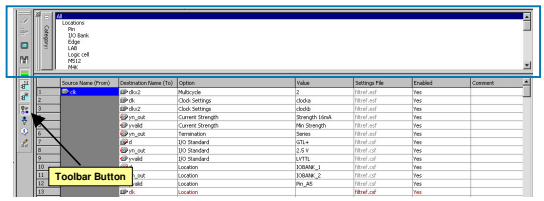
- Enables/Disables Selective Filtering of Constraints Displayed Based on Node Name
- Can Be Hidden or Minimized



127

AE Category Bar

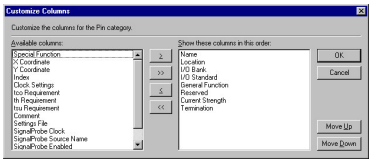
- Selects Category of Assignments to View
  - Ex. Pin Assignments, Timing Assignments
- Can Be Hidden or Minimized



128

AE Customizable Columns

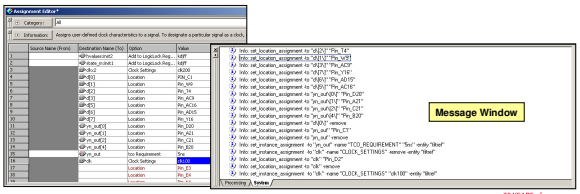
- Each Category Has a Set of Customizable Columns
  - Ex. Include Common Timing Assignments in Pin Category
- Comment Column for Each Assignment



129

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



130

Example Assignments

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

131

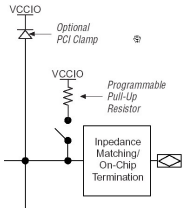
OPTIMIZATION TECHNIQUE

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

132

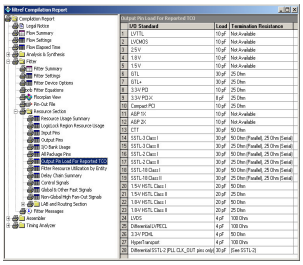
PCI I/O

- 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



133

Output Pin Load



- 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

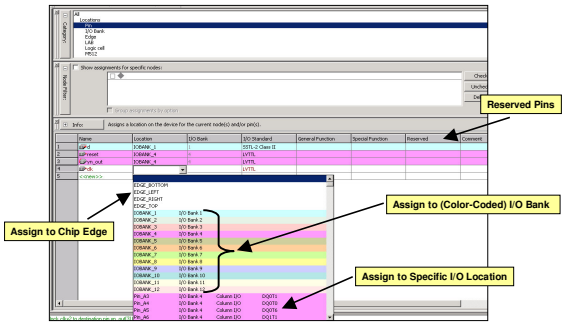
134

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

135

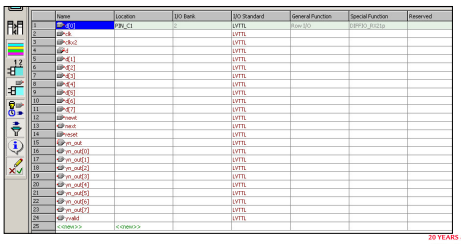
Assignment Editor I/O Assignments



136

AE Show All Known Pin Names

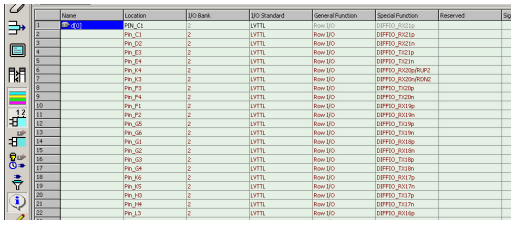
- Populate Spreadsheet with List of All Pin Names in Design
  - Assigned & Not Assigned



137

AE Assignable Pin Numbers

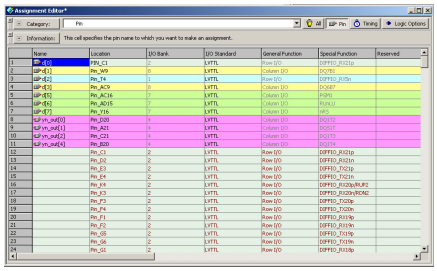
- Show All Assignable Pin Numbers
  - Populate Spreadsheet with All Pin Numbers Available for Assignment



138

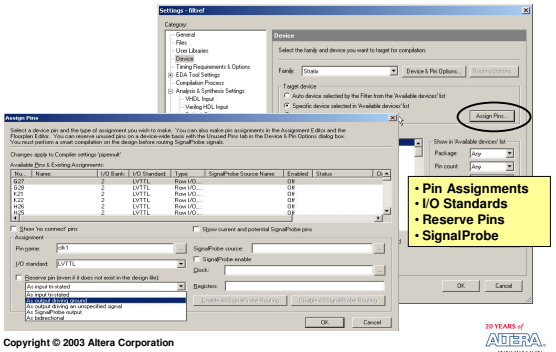
AE Show I/O Banks in Color

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



139

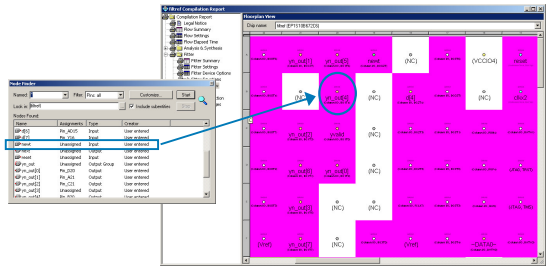
Setting Dialog Box - Device



140

Drag & Drop I/O Assignments

- Drag & Drop from Node Finder to Floorplan



141

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}

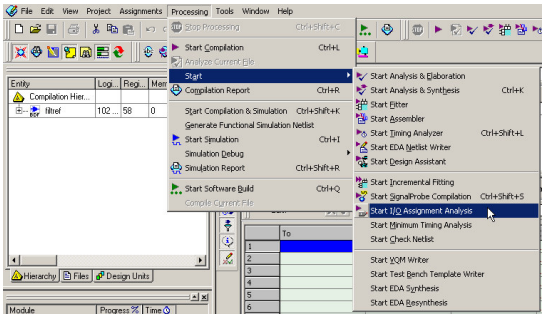
142

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

143

I/O Assignment Analysis



144

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

145

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	●	●

146

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

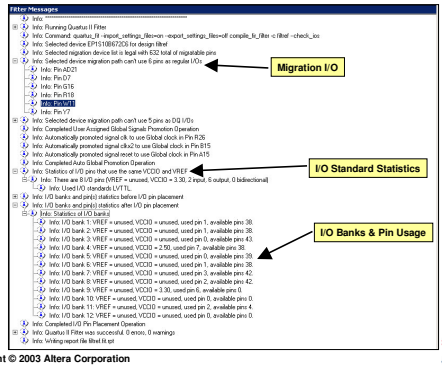
147

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

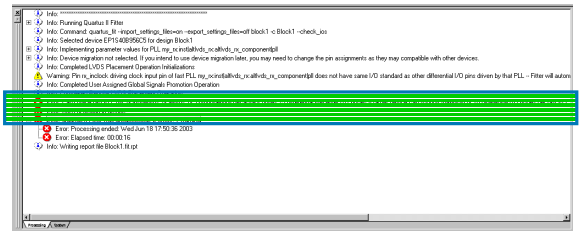
148

I/O Assignment Analysis Messages



149

Example Error Message



150

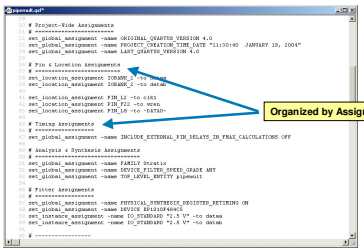
Using I/O Assignment Analysis

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

151

Quartus Settings File (QSF)

- Stores All Settings & Assignments
- Uses Tcl Syntax



152

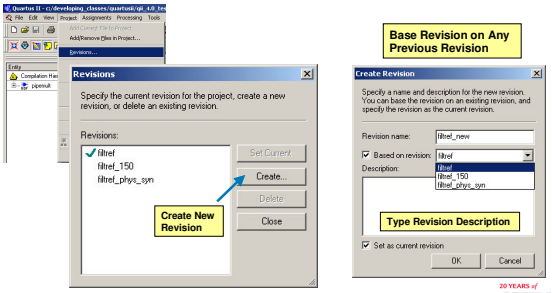
Design File Management

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

153

Creating a Revision

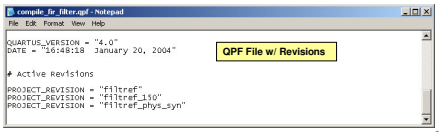
- Project ⇒ Revisions



154

Project Revision Support

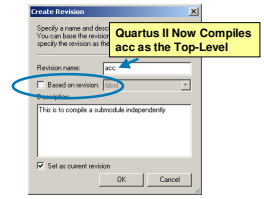
- Active Revision Names Stored in QPF
- QSF Created for Each Revision
  - <Revision_name>.QSF
- Text File Created for Each Revision
  - <Revision_Name>_description.TXT



155

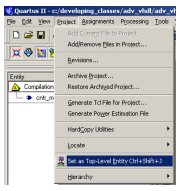
Compiling a Submodule

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

156

Exercise 2

Please go to Exercise 2 in the Exercise Manual

157

Exercise Summary

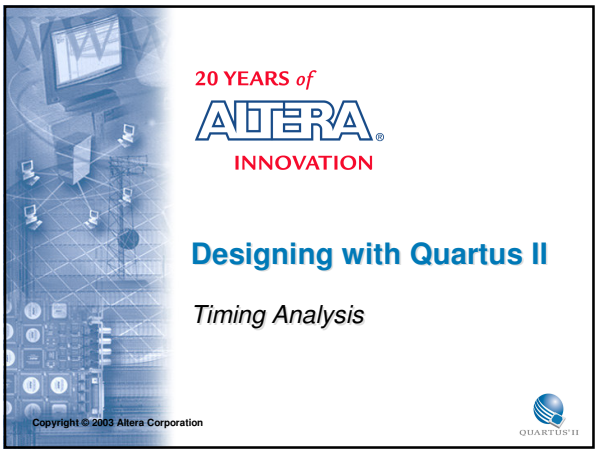
- Created Project
- Compiled Design
- Gathered Information from the Compilation Report
- Used Assignment Editor to Change Logic Usage
- Used I/O Analysis to Check I/O Placement

158

Compilation Summary

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

159



Features

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

161

Timing Analysis Agenda

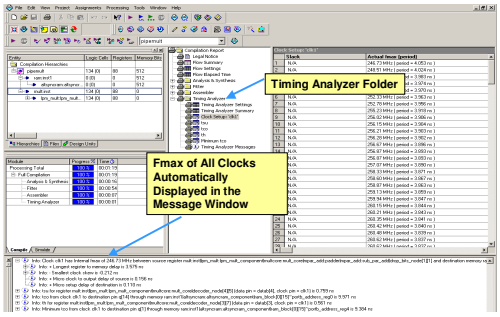
- Single Clock Analysis
- Timing Assignments
  - Global & Individual
- Multi-Clock Analysis
- Multi-Cycle Analysis
- Minimum Timing

162

Running Timing Analysis

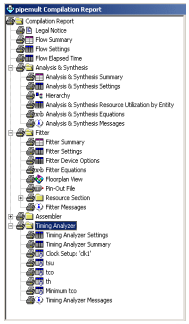
- Automatically
  - Use Full Compilation
- Manually
  - Processing Menu ⇒ Start ⇒ Start Timing Analysis
  - Tcl Scripts
  - Uses
    - Changing Speed Grade
    - Annotating Netlist with Delay Information

Timing Analysis



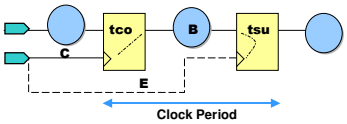
Reporting Timing Results

- 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

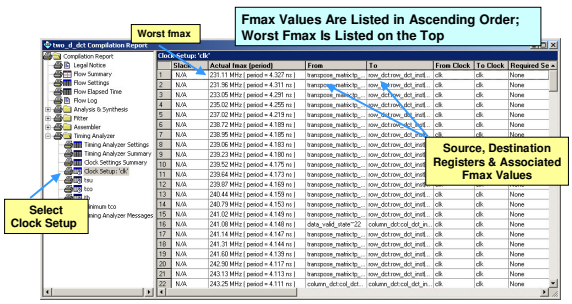


Clock Period = Clock-to-Out + Data Delay + Setup Time - Clock Skew
= tco + B + tsu - (E - C)

fmax = 1/Clock Period

166

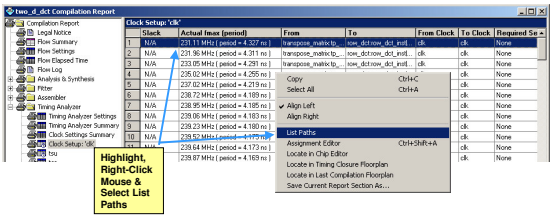
Clock Setup (fmax) Tables



167

fmax Analysis

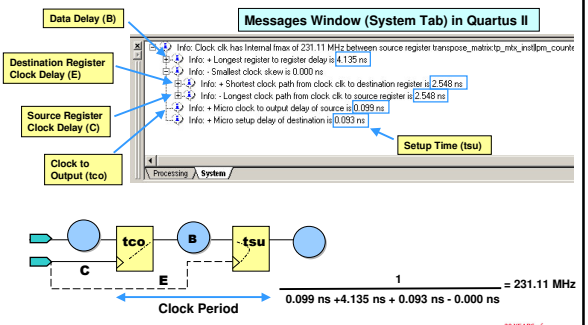
- To Analyze the Path More Closely



Similar Steps for All Timing Path Analysis in Quartus II

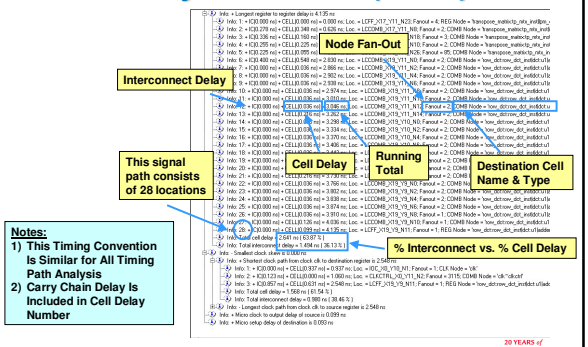
168

fmax Analysis Details



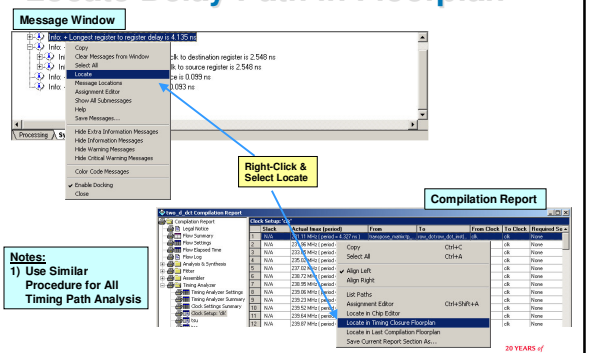
169

fmax Analysis Details (cont.)



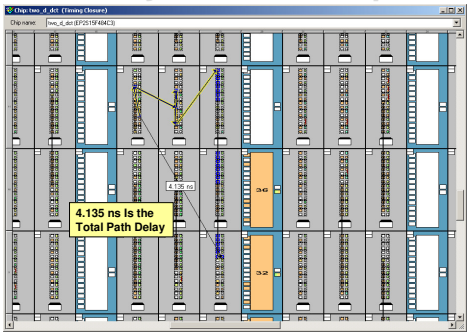
170

Locate Delay Path in Floorplan



171

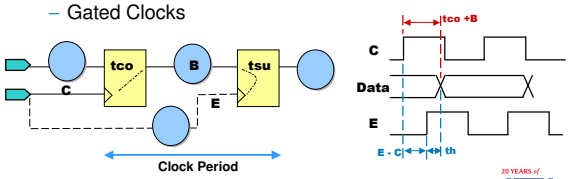
Locate Delay Path in Floorplan



172

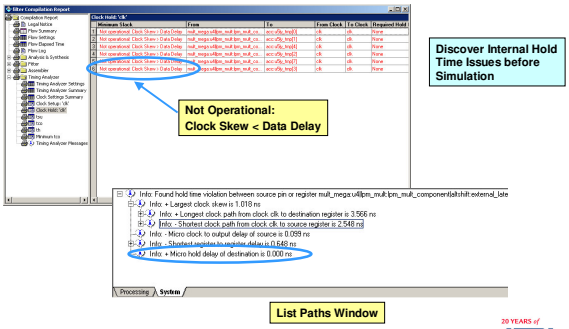
Clock Hold Analysis

- 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



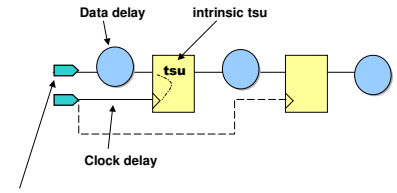
173

Hold Time Violations Table



174

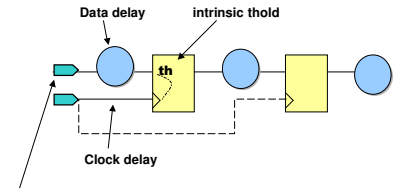
I/O Setup Time Analysis (tsu)



tsu = data delay - clock delay + intrinsic tsu

175

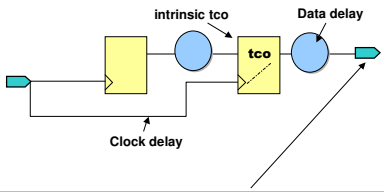
I/O Hold Time Analysis (th)



th = clock delay - data delay + intrinsic thold

176

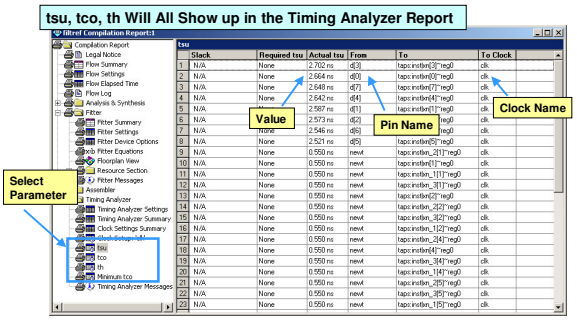
I/O Clock-to-Output Analysis (tco)



clock delay + intrinsic tco + data delay = tco

177

I/O Timing Analyzer



Note: *Timing Analysis of tpd is similar*

178

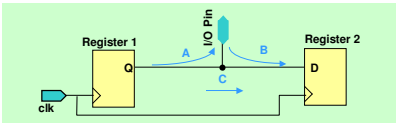
Timing Analysis Options

- 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

179

Cut Off Feedback from I/O Pin

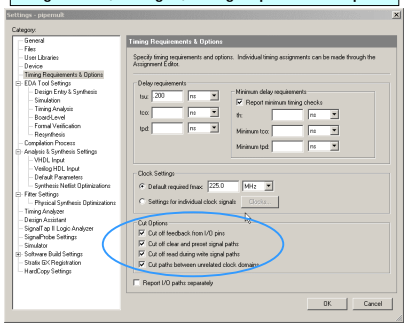
- Breaks Bidirectional I/O Pin from Analysis
- When On, Paths A & B Are Valid; C Is Not
- When Off, Paths A, B, & C Are Valid



180

Cut Options

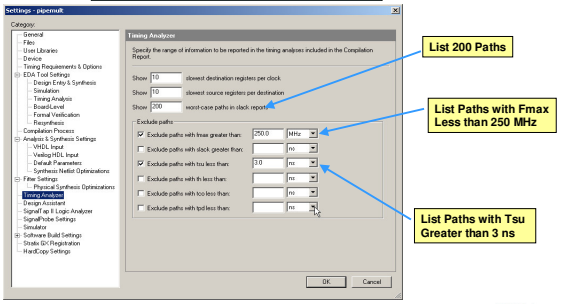
Assignments ⇒ Settings ⇒ Timing Requirements & Options



181

Timing Analyzer Options

Assignments ⇒ Settings ⇒ Timing Analyzer



182

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

183

Exercise 3

Please go to Exercise 3 in the Exercise Manual

184

Exercise Summary

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

185

Timing Assignments

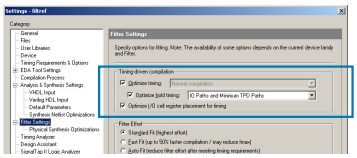
- **VERY IMPORTANT!!**
- Have a Major Impact on Design Compilation
  - Specify **ALL** Timing Requirements for Your Design
  - Fitter Works Hardest on the Worst Timing
  - Timing Will Be Reported in Red If Not Met
- Types
  - Internal & I/O Timing
  - Maximum & Minimum
- Can Be Assigned Globally or Individually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

186

Timing Driven Compilation (TDC)

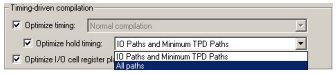
- Directs Fitter to Place & Route Logic to Meet Timing Assignments
  - Optimize Timing
    - Placing Nodes in Critical Paths Closer Together
  - Optimize I/O Cell Register Placement
    - Moving Registers into I/O Cells

Assignments ⇒ Settings ⇒ Fitting Settings



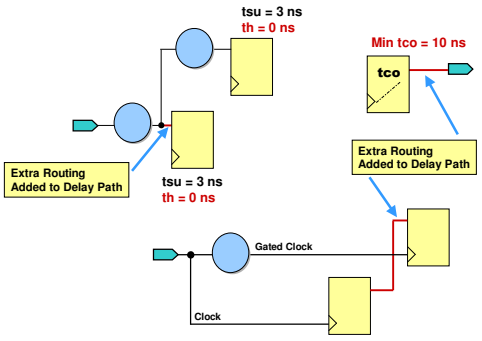
Optimize Hold Timing

- 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



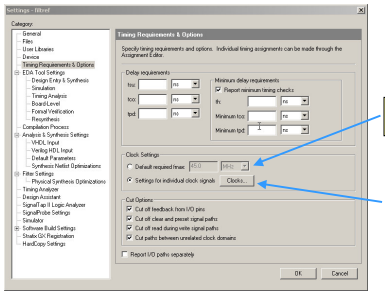
- Settings
  - Any I/O Paths
  - All Paths (I/O & Internal)

Optimize Hold Time Examples



Global Clock Assignments

Assignments ⇒ Settings ⇒ Timing Requirements & Options

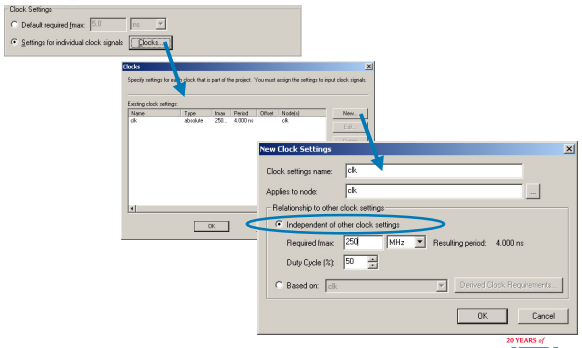


Global Clock Assignment for a Single Clock Design

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

190

Asynchronous Global Clocks



191

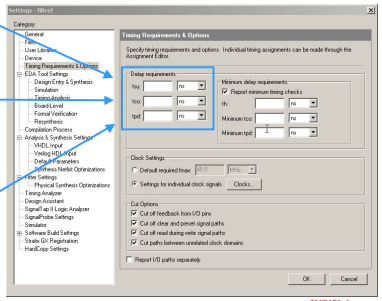
Global Maximum I/O Delay

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

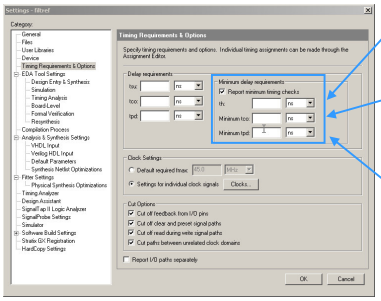
Assignments ⇒ Settings ⇒ Timing Requirements & Options



192

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

193

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)**

194

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)	multinstlpm_mult...	ram:inst2\altsy...	clk1	clk1	4.444 ns
2	1.205 ns	308.74 MHz (period = 3.239 ns)	multinstlpm_mult...	ram:inst2\altsy...	clk1	clk1	4.444 ns
3	1.215 ns	309.69 MHz (period = 3.229 ns)	multinstlpm_mult...	ram:inst2\altsy...	clk1	clk1	
4	1.247 ns	312.79 MHz (period = 3.197 ns)	multinstlpm_mult...	ram:inst2\altsy...	clk1	clk1	
5	1.253 ns	313.38 MHz (period = 3.191 ns)	multinstlpm_mult...	ram:inst2\altsy...	clk1	clk1	
6	1.266 ns	314.65 MHz (period = 3.178 ns)	multinstlpm_mult...	ram:inst2\altsy...	clk1	clk1	
7	1.293 ns	317.36 MHz (period = 3.151 ns)	multinstlpm_mult...	ram:inst2\altsy...	clk1	clk1	
8	1.300 ns	318.07 MHz (period = 3.144 ns)	multinstlpm_mult...	ram:inst2\altsy...	clk1	clk1	4.444 ns

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

	Slack	Required tsu	Actual tsu	From	To	To Clock
73	-0.003 ns	0.200 ns	0.203 ns	datab[1]	multinstlpm_multpm_mult_component\mult_3up:au...	clk1
74	-0.002 ns	0.200 ns	0.202 ns	datab[3]	multinstlpm_multpm_mult_component\mult_3up:au...	clk1
75	-0.001 ns	0.200 ns	0.201 ns	datab[1]	multinstlpm_multpm_mult_component\mult_3up:au...	clk1
76	0.001 ns	0.200 ns	0.199 ns	datab[4]	multinstlpm_multpm_mult_component\mult_3up:au...	clk1
77	0.002 ns	0.200 ns	0.198 ns	datab[2]	multinstlpm_multpm_mult_component\mult_3up:au...	clk1
78	0.004 ns	0.200 ns	0.196 ns	datab[3]	multinstlpm_multpm_mult_component\mult_3up:au...	clk1
79	0.007 ns	0.200 ns	0.193 ns	datab[3]	multinstlpm_multpm_mult_component\mult_3up:au...	clk1
80	0.009 ns	0.200 ns	0.191 ns	datab[1]	multinstlpm_multpm_mult_component\mult_3up:au...	clk1
81	0.009 ns	0.200 ns	0.191 ns	datab[2]	multinstlpm_multpm_mult_component\mult_3up:au...	clk1
82	0.009 ns	0.200 ns	0.191 ns	datab[5]	multinstlpm_multpm_mult_component\mult_3up:au...	clk1

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

195

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 ?)

196

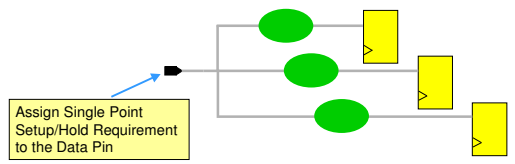
Individual Assignment Targets

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

197

Single-Point tsu/th

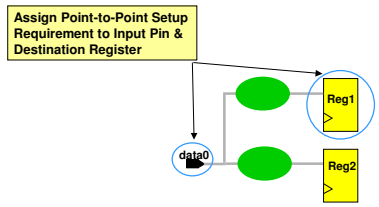
- Setup/Hold Timing Required on Every Register Fed by Pin



198

Point-to-Point tsu/th

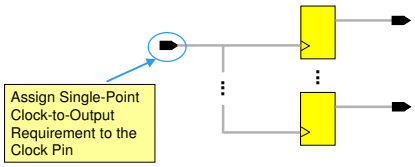
- Setup/Hold Timing Required Only on the Specified Path



199

Single-Point tco

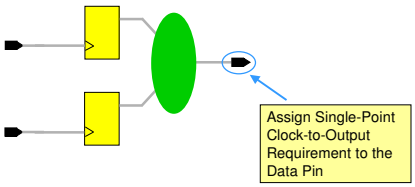
- Clock-to-Out Timing Required on All tco Paths Starting from the Specified Clock



200

Single-Point tco (cont.)

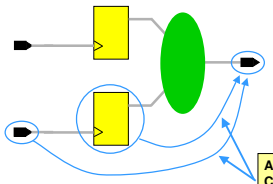
- Clock-to-Out Timing Required on All Paths from Any Clock to Specified Output Pin



201

Point-to-Point tco

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

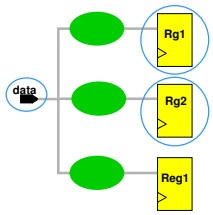


Assign Point-to-Point Clock-to-Out Requirement to Source Register or Clock Pin & Output Pin

202

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?"

203

Ex. Assigning Setup Requirement

Assignment Editor Is Used for All Individual Timing Requirements

Select Timing Category

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

204

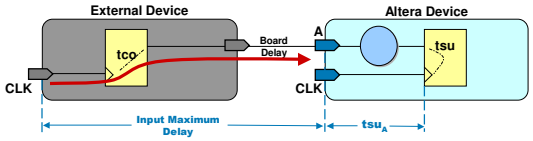
External Delay Assignments

- 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

205

Input Maximum Delay

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

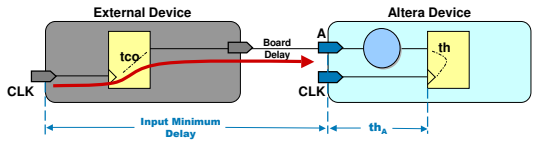


$tsu_A \leq t_{CLK}$ – *Input Maximum Delay*

206

Input Minimum Delay

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

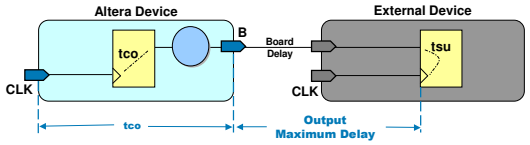


$th_A \leq$ *Input Minimum Delay*

207

Output Maximum Delay

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

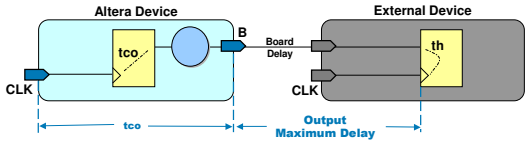


$tco_B \leq t_{CLK}$ - *Output Maximum Delay*

208

Output Minimum Delay

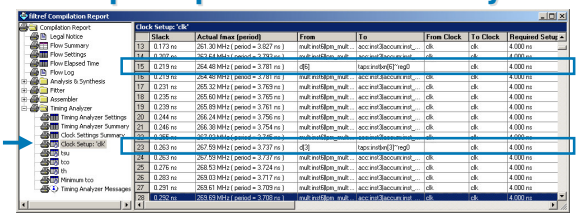
- Minimum Delay from Altera I/O to External Device
  - Represents External Device th - Board Delay
- Constrains Output Pin Minimum tco



$tco_B \geq$ *Output Minimum Delay*

209

Example Input Maximum Delay



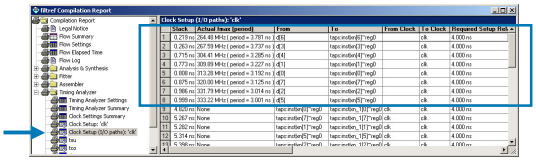
Input Maximum Delay (d) = 4 ns

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**

210

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**



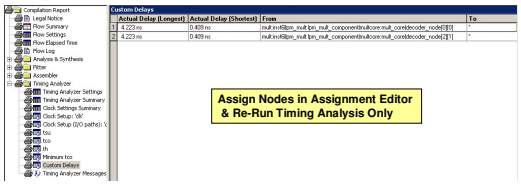
211

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

212

Report Delay Assignment

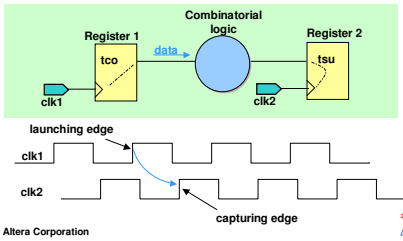


- Reports Delay between Selected Nodes
  - Pin-Register
  - Register-Register
  - Register-Pin
- Reports Results in Custom Delay Section of Timing Analyzer

213

Multi-Clock Frequency Analysis

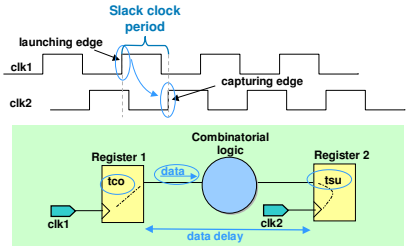
- Analyzes Timing on Register-to-Register Paths Controlled by Different Synchronous Clocks
  - Individual Clocks Treated as Same Frequency & Phase Unless Specified



214

Slack Equations for Multi-Clock

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



215

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

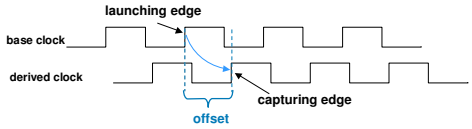
216

1) Create Clock Settings

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

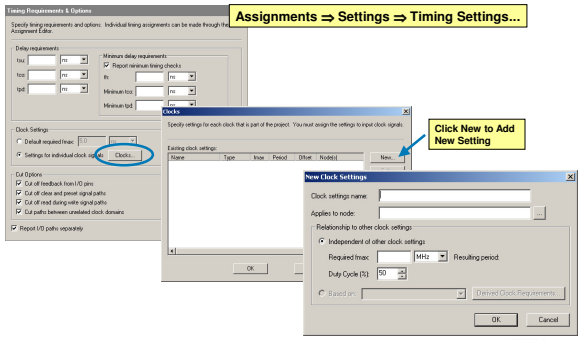
derived clock = base clock x (ratio) + offset

Note: Multiple Base & Derived Clocks Allowed



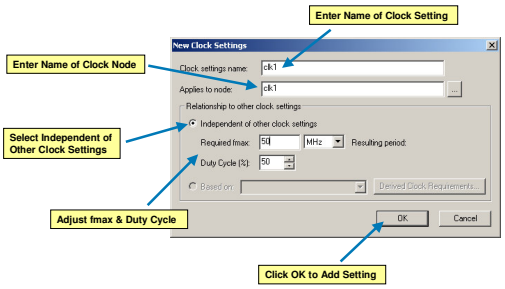
217

Base Clocks



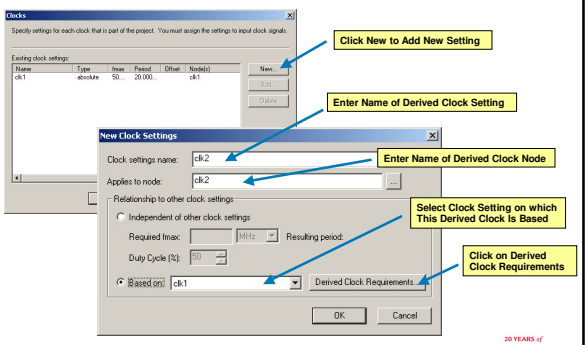
218

Base Clocks (cont.)



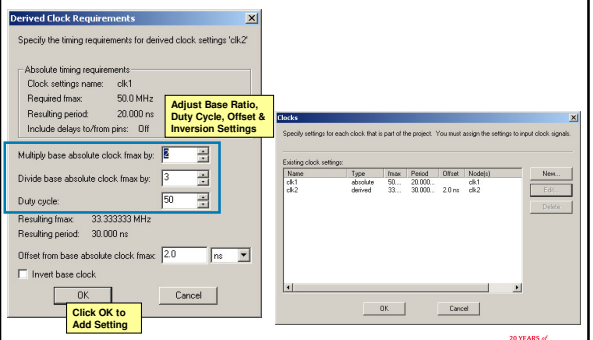
219

Derived Clocks



220

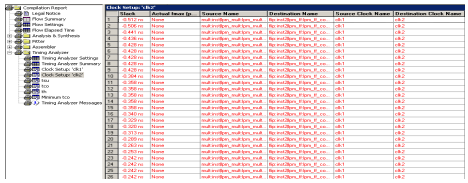
Derived Clocks (cont.)



221

3) Examine Timing Analyzer Report

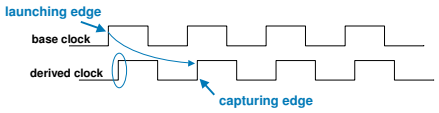
- Positive Slack - Timing Was Met (BLACK)
- Negative Slack - By How Much Timing Was Not Met (RED)
- Must Alter Design or Use Timing Assignments (Multi-Cycle) to Resolve Negative Slack



222

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



223

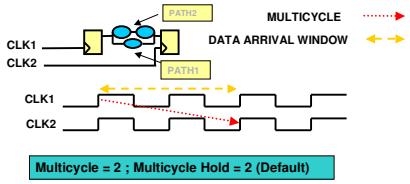
Assigning Multi-Cycle Paths

- 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

224

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

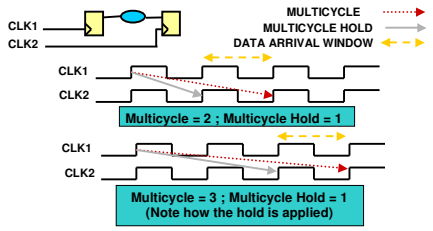


Multicycle = 2 ; Multicycle Hold = 2 (Default)

225

Multi-Cycle Hold Assignment

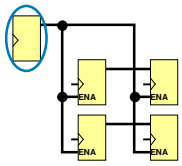
- **Minimum** Point-to-Point Timing
  - Data Must Arrive after Hold Time
  - Used in Conjunction with a Multi-cycle Assignment



226

Clock Enable Multi-Cycle Assignments

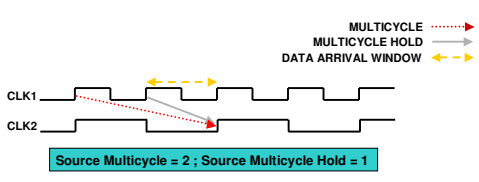
- Assigns Multi-Cycle to Source of Clock Enable
  - I/O Pin
  - Register
- Settings
  - Clock Enable Multicycle (Maximum)
  - Clock Enable Multicycle Hold (Minimum)



227

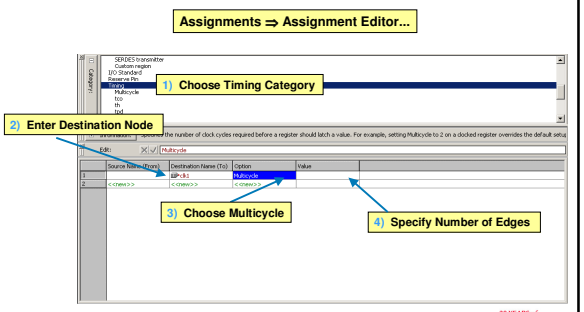
Source Mult-Cycle Assignments

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



228

Ex. Assigning a Multi-Cycle



229

Minimum Timing Analysis

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

230

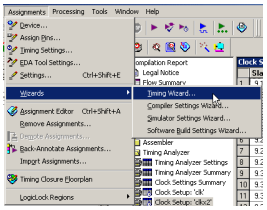
Running Minimum Timing Analysis

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

231

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



232

Exercise 4

Please go to Exercise 4 in the Exercise Manual

233

Exercise Summary

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

234

Timing Analysis Summary

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

235

More Timing Analysis Info

- Quartus II Handbook: Using Quartus II Timing Analysis

236

20 YEARS of ALTERA INNOVATION

Designing with Quartus II

Simulation

Quartus II Simulation

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

238

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

239

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

240

Simulator Features (cont.)

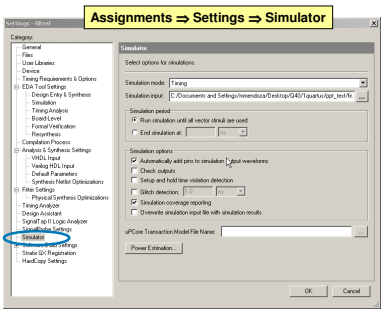
- Performs PowerGauge™ Analysis
- Converts VWF into HDL Testbench
- Generates HDL Testbench Template
- Supports Breakpoints
- Performs Automatically
  - Adding Output Pins to Output Waveform File
  - Checking Outputs at End of Simulation

241

Simulator Settings

- Mode
- Input File
- Period
- Options

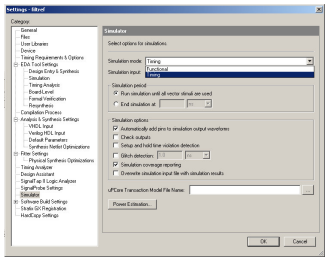
Assignments ⇒ Settings ⇒ Simulator



242

Simulator Mode

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

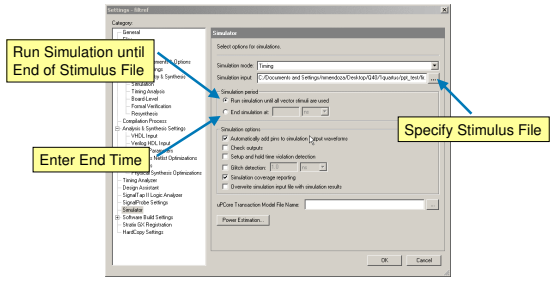


* Must Generate Functional Simulation Netlist (Processing Menu) before Performing Functional Simulation

243

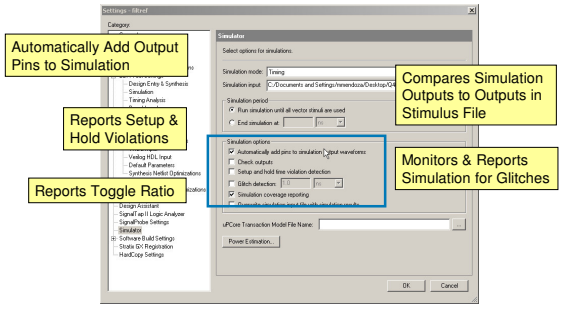
Simulator Input & Period

- Specifies Stimulus & Length of Simulation Period



244

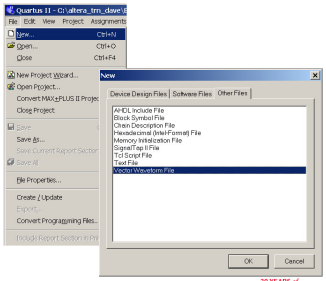
Simulator Options



245

Create New Vector Waveform File

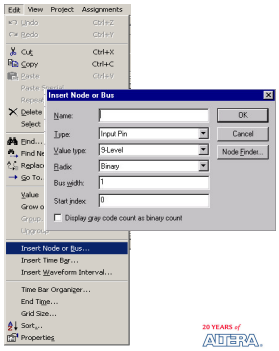
- Select **File ⇒ New ⇒ Vector Waveform File** (Other Files Tab)



246

Insert Nodes

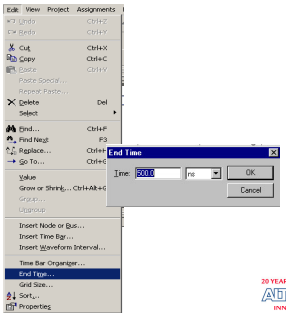
- Select **Insert Node or Bus** (Edit Menu)
  - VWF Must Be Open
  - Use Node Finder



247

Specify End Time

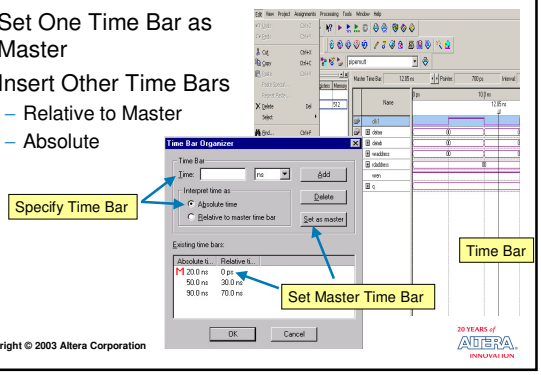
- Maximum Length of Simulation Time
  - Edit Menu



248

Insert Time Bars

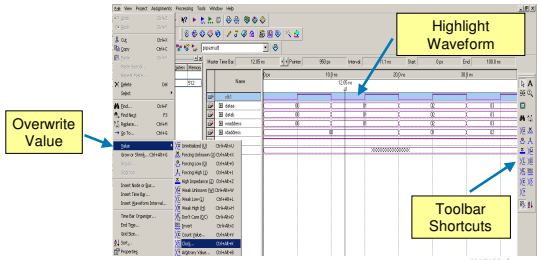
- Set One Time Bar as Master
- Insert Other Time Bars
  - Relative to Master
  - Absolute



249

Draw Stimulus Waveform

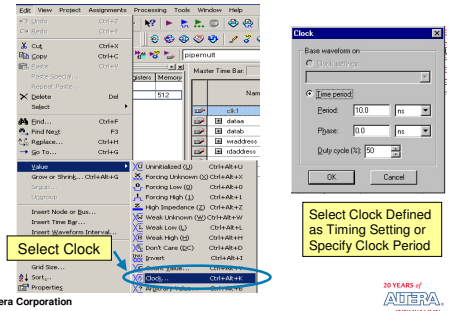
- Highlight Portion of Waveform to Change
- Overwrite Value with Desired Value



250

Create Clock

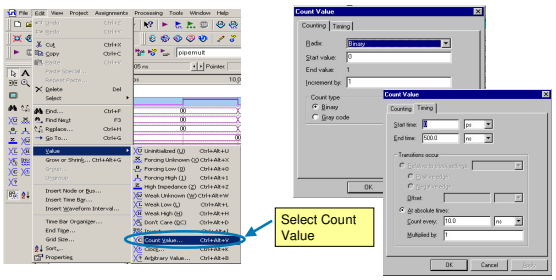
- Highlight Waveform & Enter Period



251

Creating Counting Pattern

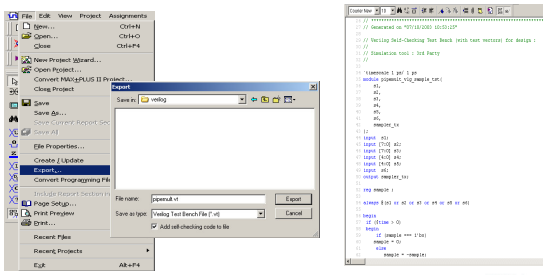
- Highlight Waveform & Enter Pattern



252

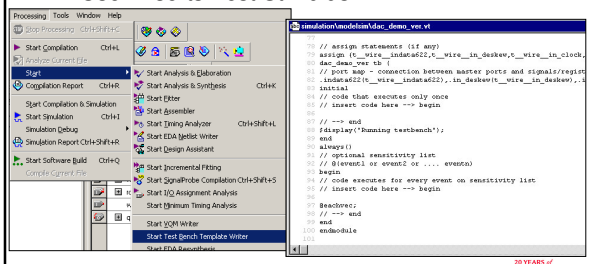
Waveform to Testbench Generator

- Converts VWF into HDL Testbench



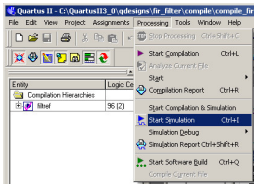
Testbench Template Generator

- Generates HDL Testbench Template
  - User Inserts Test Stimulus



Starting Simulation

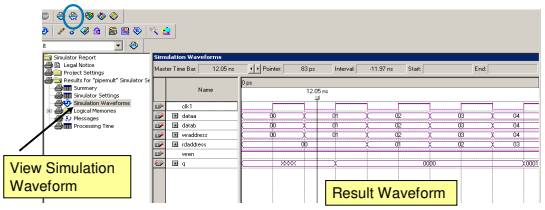
- Processing Menu ⇒ Start Simulation 



- Scripting

Simulator Report

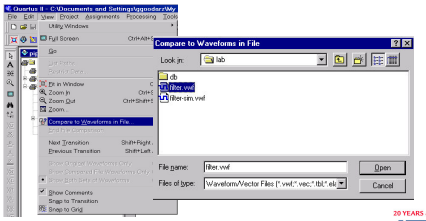
- Displays Simulation Result Waveform



256

Comparing Waveforms

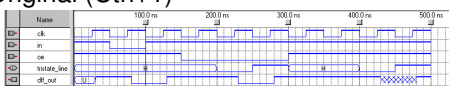
- Select Compare to Waveforms (View Menu)
  - Simulation Waveform Must Be Open
- Select VWF Comparison File



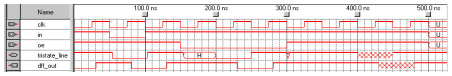
257

Compared Waveforms (Simulator Report)

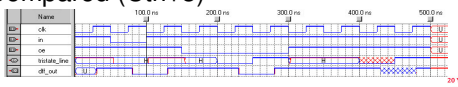
- Original (Ctrl+1)



- Actual (Ctrl+2)

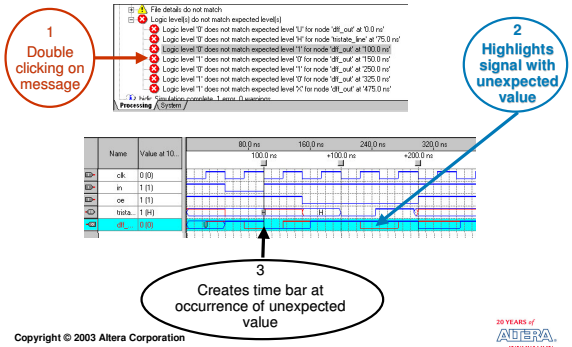


- Compared (Ctrl+3)



258

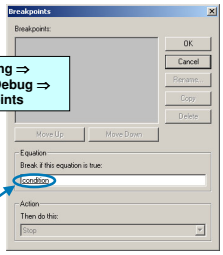
Using Message Window to Debug



259

Breakpoints

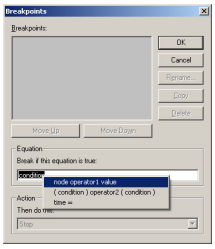
- Interrupts Simulation at Specified Points
- Consists of 2 Parts
  - Equation (Condition)
  - Action
    - Stop
    - Give Error
    - Give Warning
    - Give Info



260

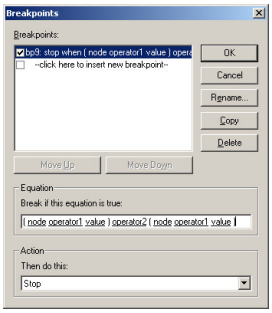
Breakpoint Conditions

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



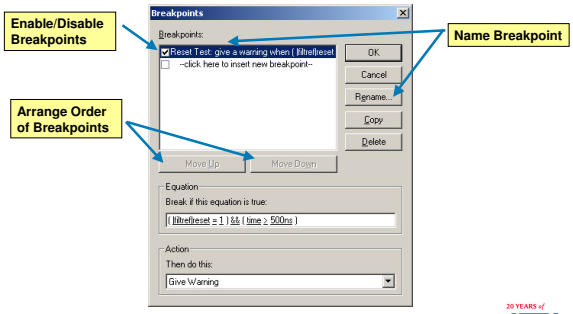
261

Breakpoint Equations (cont.)



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

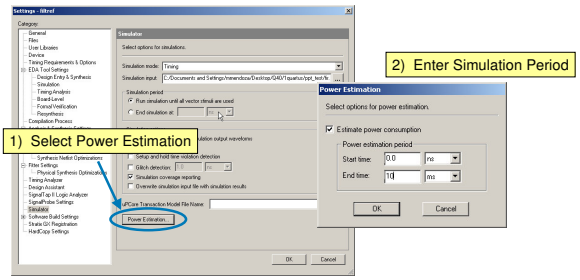
Example Breakpoint



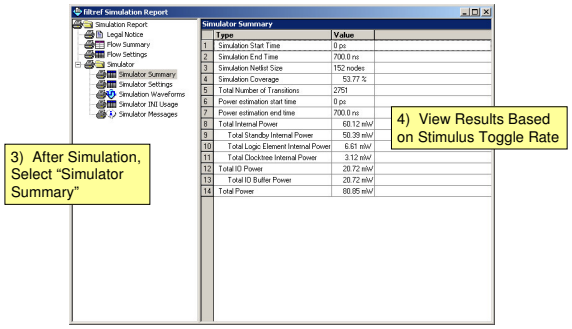
PowerGauge Analysis Software

- 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

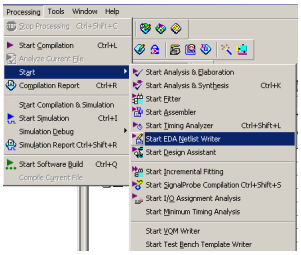


Power Analysis (cont.)



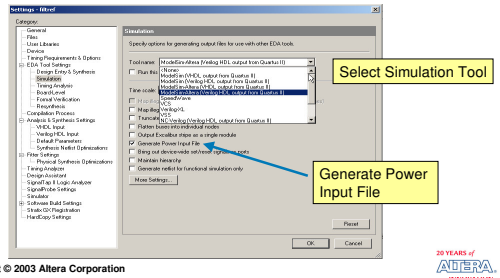
Using 3rd Party Simulators

- 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

271

Simulation Summary

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

272

20 YEARS of ALTERA INNOVATION

Designing with Quartus II

Programming/Configuration

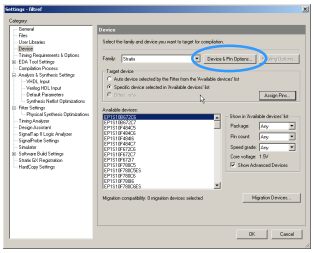
Programming/Configuration

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

274

Setting Device Options

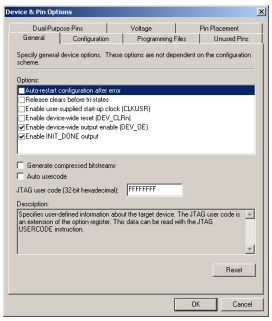
- Assignments ⇒ Device ⇒ Device & Pin Options Button



Device Options Control Configuration & Initialization of Device

275

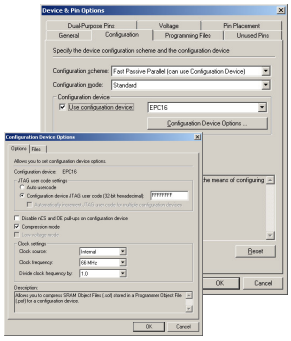
General Tab



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

276

Configuration Tab

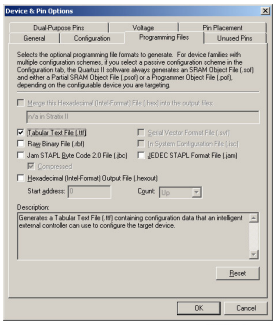


- Choose Device Configuration Mode & Available Options
  - Generates Correct Configuration & Programming Files Every Compilation
- Enables Special Features of Configuration Devices
  - Enable Programming File Compression
  - Set Configuration Clock Frequency

277

Programming Files Tab

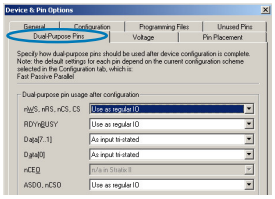
- Output Files Always Created
  - POF (Programming Object File)
  - SOF (SRAM Object File)
- Other Selectable Output Files
  - JAM (JEDEC STAPL)
  - JBC (JAM Byte-Code)
  - RBF (Raw Binary File)
  - HEXOUT (Intel Hex Format)

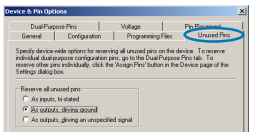


278

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



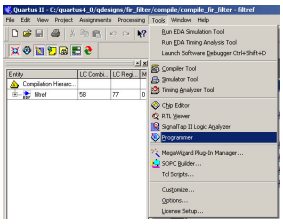


279

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

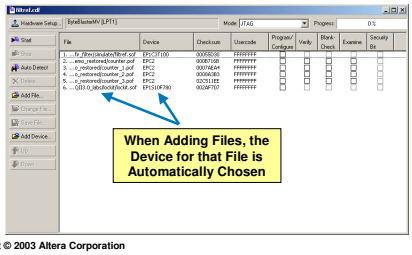
Open Programmer



- 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

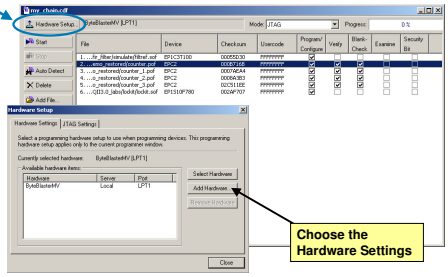
- Start Programming
- Auto Detect Devices in JTAG Chain
- Add/Remove/Change Devices in Chain
- Add/Remove/Changes Files in Chain
- Change Order of Files in Chain
- Setup Programming Hardware

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

283

Setting up Programming Hardware

Click on the Hardware Setup Button



Choose the Hardware Settings

284

Chain Programming Modes

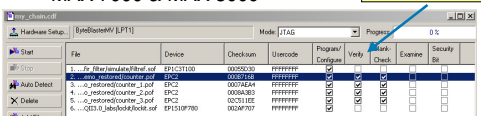
- Altera FPGAs, CPLDs & Non-Altera Devices
  - JTAG
- Altera FPGAs Only
  - Passive Serial
  - Active Serial
- CPLDs & Configuration Devices in APU
  - In-Socket

285

Programming Options

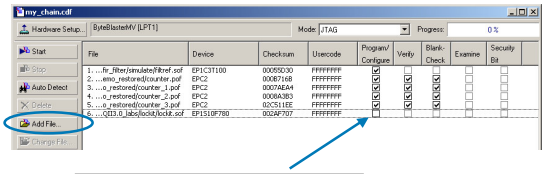
- Program/Configure
  - Applies to All Devices
- Verify, Blank-Check & Examine
  - Configuration Devices
  - MAX 7000 & MAX 3000
- Security Bit
  - MAX 7000 & MAX 3000

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



286

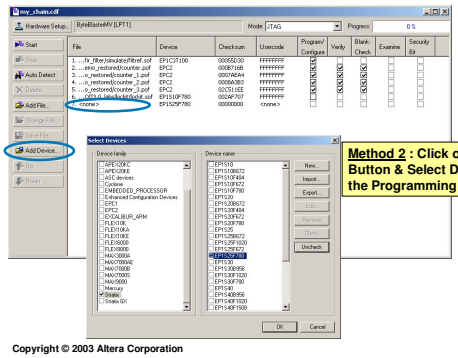
Bypassing Devices in JTAG Chain (1)



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

287

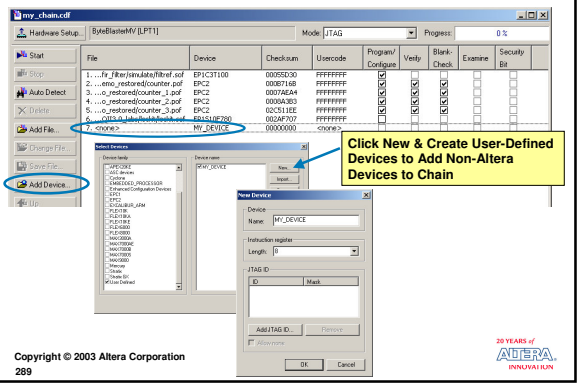
Bypassing Devices in JTAG Chain (2)



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

288

Adding Non-Altera Device to Chain

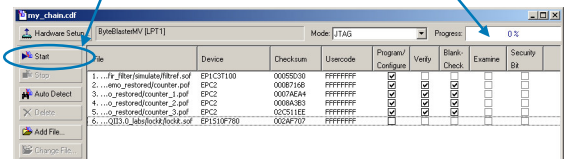


289

Starting the Programmer

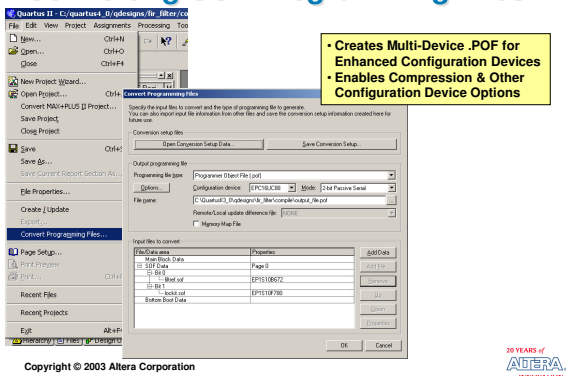
Click Program Button Once CDF File & Hardware Setup Are Complete

Progress Field Shows the Percentage of Completion for the Programmer



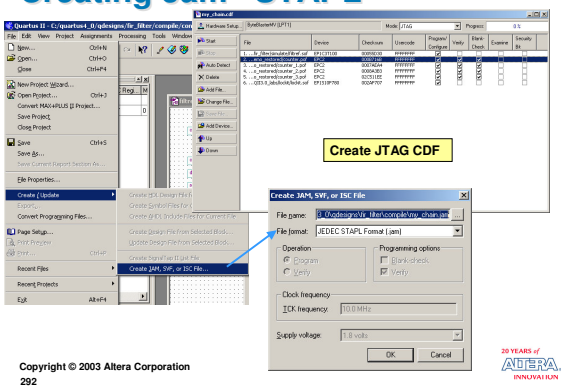
290

Converting SOF Programming Files



291

Creating Jam™ STAPL



292

Exercise 6 or 7 (Optional)

Please go to Exercise 6 or 7 in the Exercise Manual

293

Programming/Configuration Summary

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

294

Class Summary

- Design Entry Techniques
- Project Creation
- Compiler Settings & Assignment Editor
- Timing Analysis
- Simulation
- Programming/Configuration

295

Learn More through Technical Training

Instructor-Led Training	On-Line 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).	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

296

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

297

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

298