SOPC WORLD 2004

Using the Chip Editor in the Quartus II Design Environment

ALTERA.

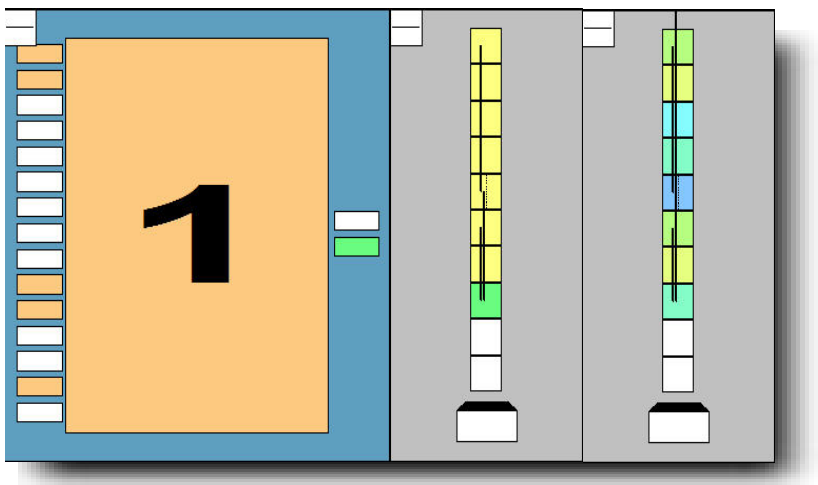
Agenda

- Introduction to the Chip Editor
- The Resource Property Editor
  - LE Editor
  - I/O Editor
- The Chip Editor Tools
- Chip Editor Applications
  1. Design Analysis
  2. Design Flaw
  3. Timing Verification
  4. Last Minute ECO
  5. Using the PLL Editor

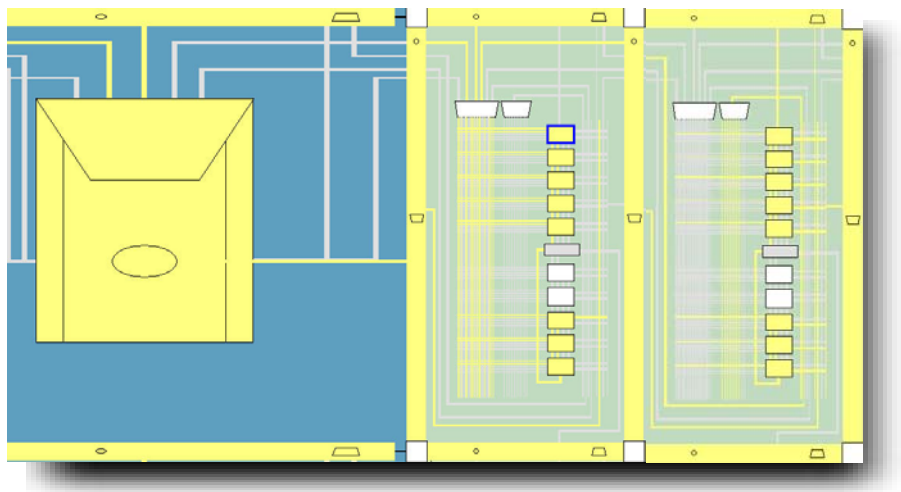
Getting Started With Chip Editor

- What is the Chip Editor?
  - A Graphical Interface for Viewing Detailed Information About the Target Design & the Target Architecture
  - Designs Can Be Modified Without Performing a Recompilation!

More Detail Than the Quartus II Floorplan Editor !



Quartus II v2.2 Floorplan Editor



Quartus II v3.0 Chip Editor

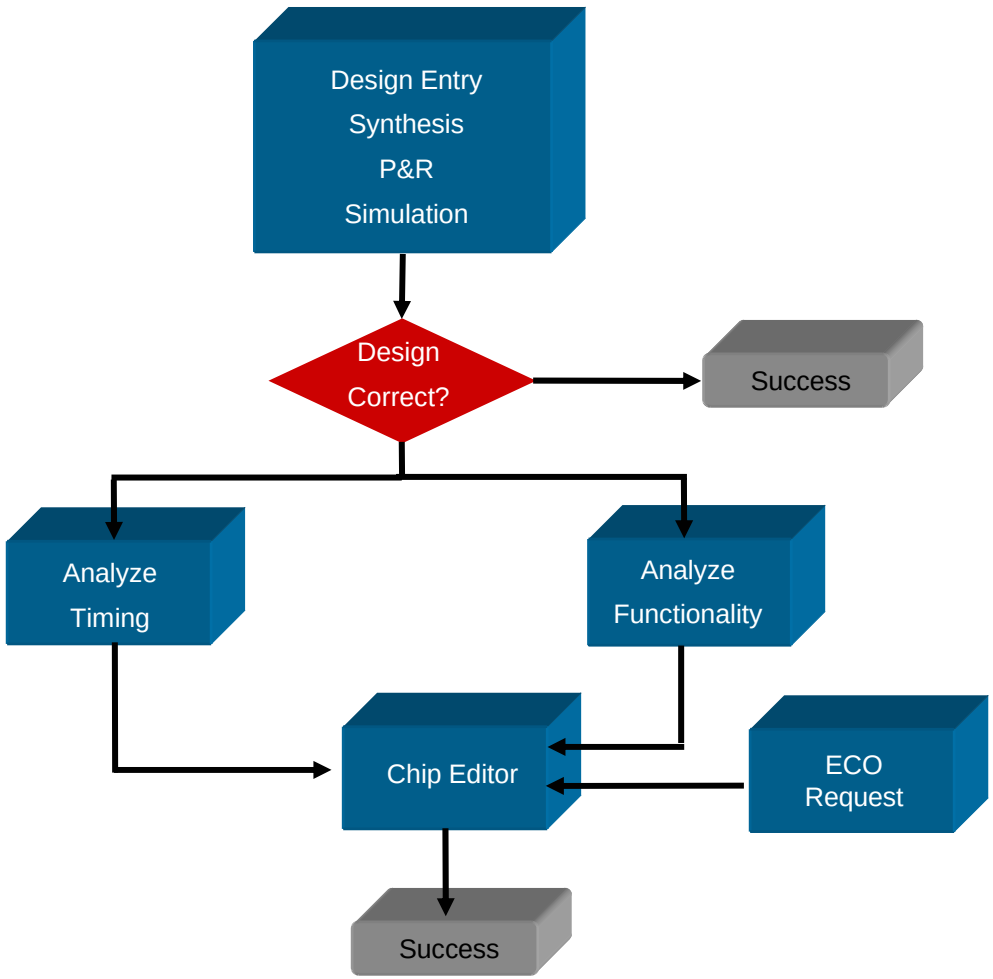
SOPC WORLD 2004

ALTERA.

Getting Started with Chip Editor

- Why Would My Customer Use the Chip Editor?
  - Quick Turn-around-Time
    - Corrects Design Flaws
    - Last Minute ECO
    - Tweak Timing Assignments
- What Type of Customer Should use the Chip Editor?
  - A Customer Who Has to Perform a ***Minor*** Change to Their Design
  - Only Advanced Customers
    - Detailed Knowledge of the Target Architecture
    - Detailed Knowledge of the Design Being Implemented

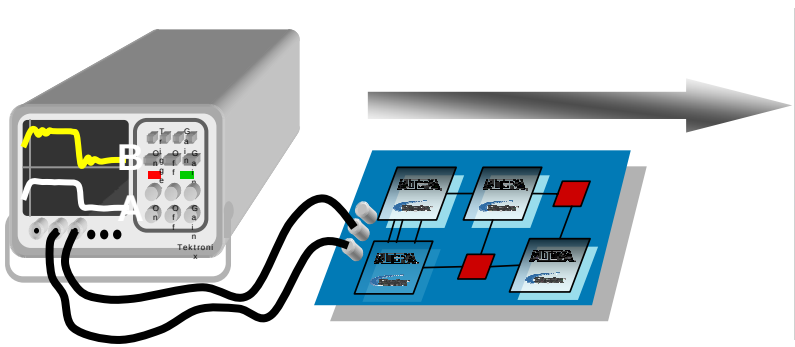
General Flow with Chip Editor

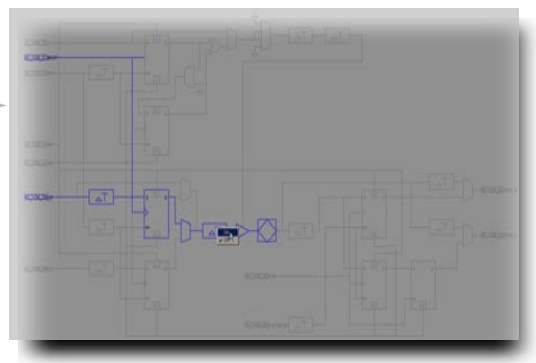


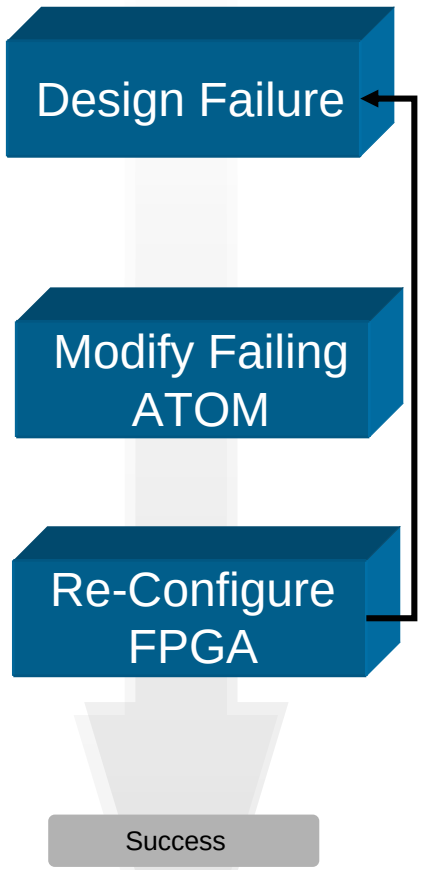
Customer Flow with Chip Editor

Case 1: Incorrect Functionality

- Use the Chip Editor to Modify Device Resources to Correct Functionality







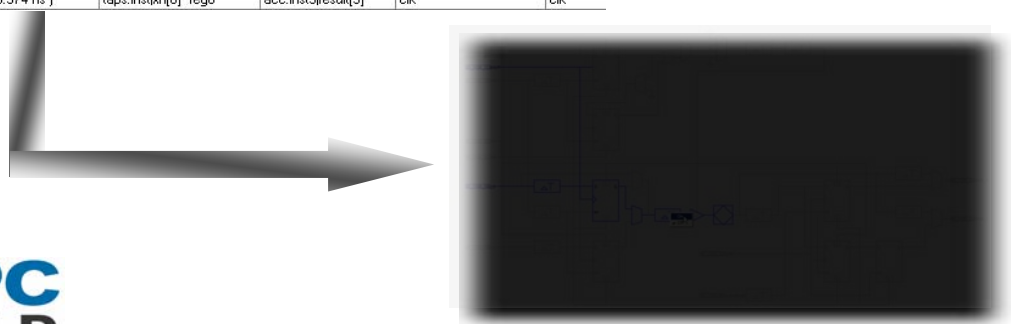
Customer Flow With Chip Editor

Case 2: Timing Analysis

- Trace Critical Path with the Chip Editor to Determine where Improvements can be Made

Clock Setup: 'clk'

	Slack	Actual fmax (period)	Source Name	Destination Name	Source Clock Name	Destinat
1	-0.279 ns	106.73 MHz (period = 9.369 ns)	taps:inst|xn[0]~reg0	acc:inst3|result[11]	clk	clk
2	-0.182 ns	107.85 MHz (period = 9.272 ns)	state_m:inst1|filter~10	acc:inst3|result[11]	clk	clk
3	-0.177 ns	107.91 MHz (period = 9.267 ns)	taps:inst|xn[0]~reg0	acc:inst3|result[10]	clk	clk
4	-0.177 ns	107.91 MHz (period = 9.267 ns)	taps:inst|xn[0]~reg0	acc:inst3|result[9]	clk	clk
5	-0.177 ns	107.91 MHz (period = 9.267 ns)	taps:inst|xn[0]~reg0	acc:inst3|result[8]	clk	clk
6	-0.177 ns	107.91 MHz (period = 9.267 ns)	taps:inst|xn[0]~reg0	acc:inst3|result[7]	clk	clk
7	-0.177 ns	107.91 MHz (period = 9.267 ns)	taps:inst|xn[0]~reg0	acc:inst3|result[6]	clk	clk
8	-0.080 ns	109.05 MHz (period = 9.170 ns)	state_m:inst1|filter~10	acc:inst3|result[10]	clk	clk
9	-0.080 ns	109.05 MHz (period = 9.170 ns)	state_m:inst1|filter~10	acc:inst3|result[9]	clk	clk
10	-0.080 ns	109.05 MHz (period = 9.170 ns)	state_m:inst1|filter~10	acc:inst3|result[8]	clk	clk
11	-0.080 ns	109.05 MHz (period = 9.170 ns)	state_m:inst1|filter~10	acc:inst3|result[7]	clk	clk
12	-0.080 ns	109.05 MHz (period = 9.170 ns)	state_m:inst1|filter~10	acc:inst3|result[6]	clk	clk
13	0.101 ns	111.25 MHz (period = 8.989 ns)	state_m:inst1|filter~12	acc:inst3|result[11]	clk	clk
14	0.116 ns	111.43 MHz (period = 8.974 ns)	taps:inst|xn[0]~reg0	acc:inst3|result[5]	clk	clk



Timing Analysis Failure

Adjust Timing Settings

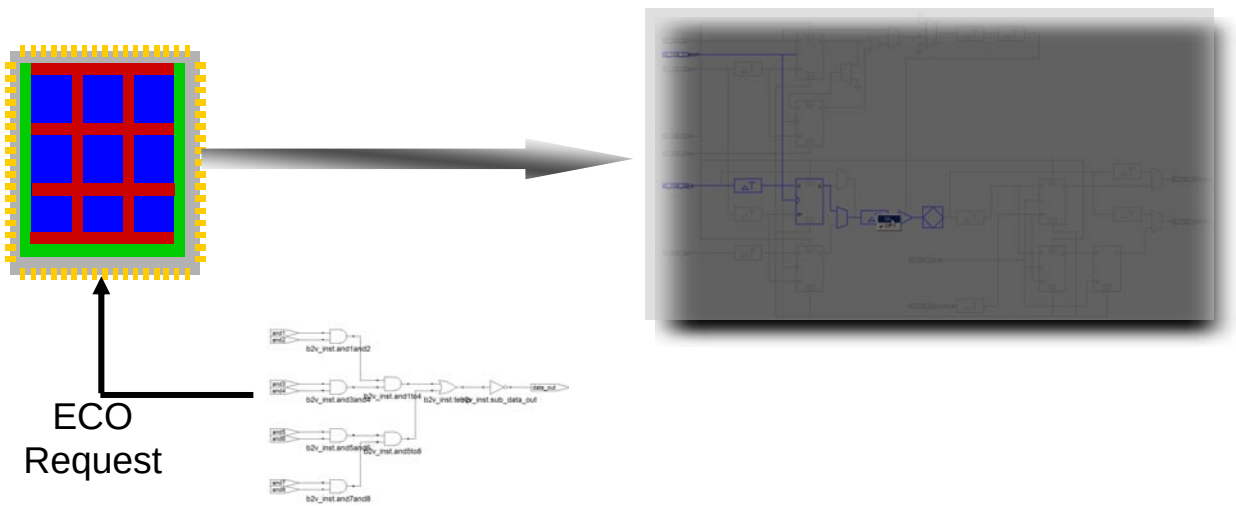
Re-Configure FPGA

Success

Customer Flow With Chip Editor

Case 3: ECO Requests

- Use the Chip Editor to Modify Properties of Design to Implement ECO



ECO Request

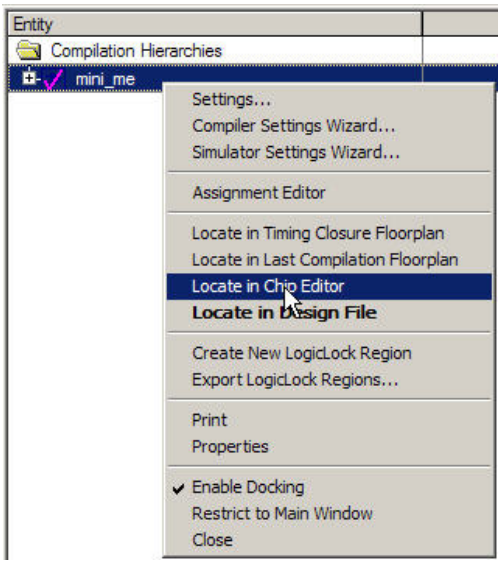
Adjust Properties

Re-Configure FPGA

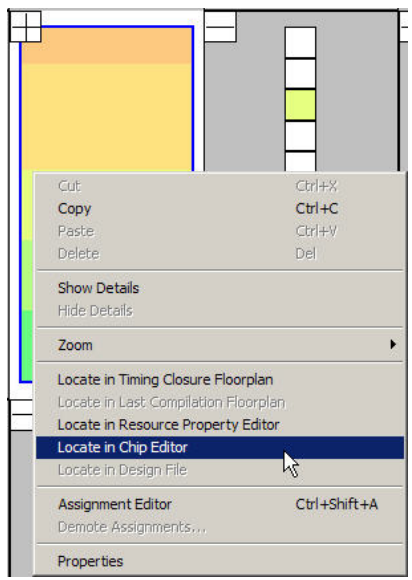
Success

Launching Chip Editor

1. Compilation Hierarchy

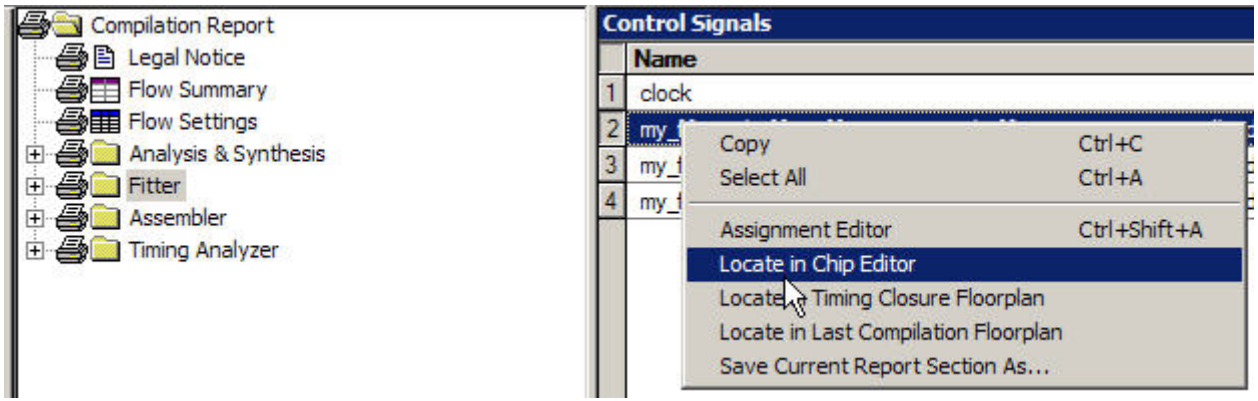


2. Floorplan Editor

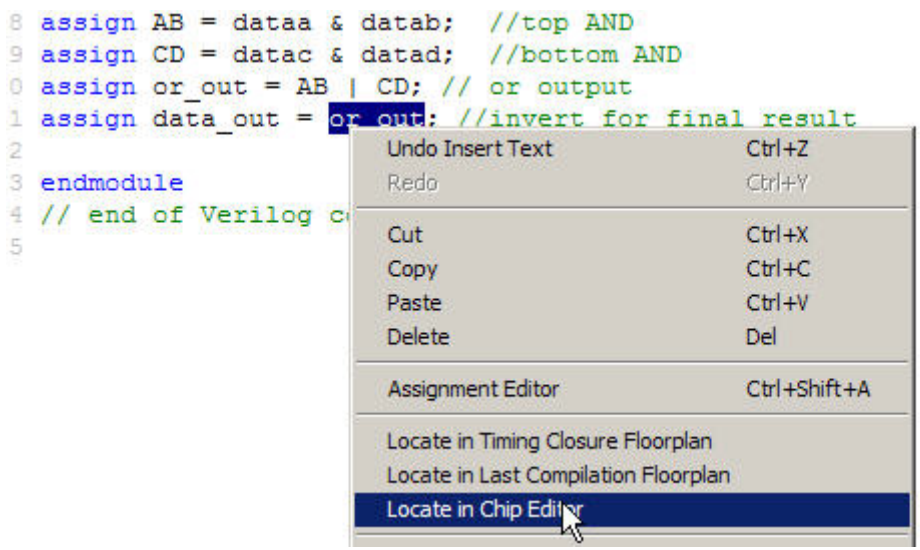


Launching Chip Editor

3. Compilation Report



4. Design Source Code

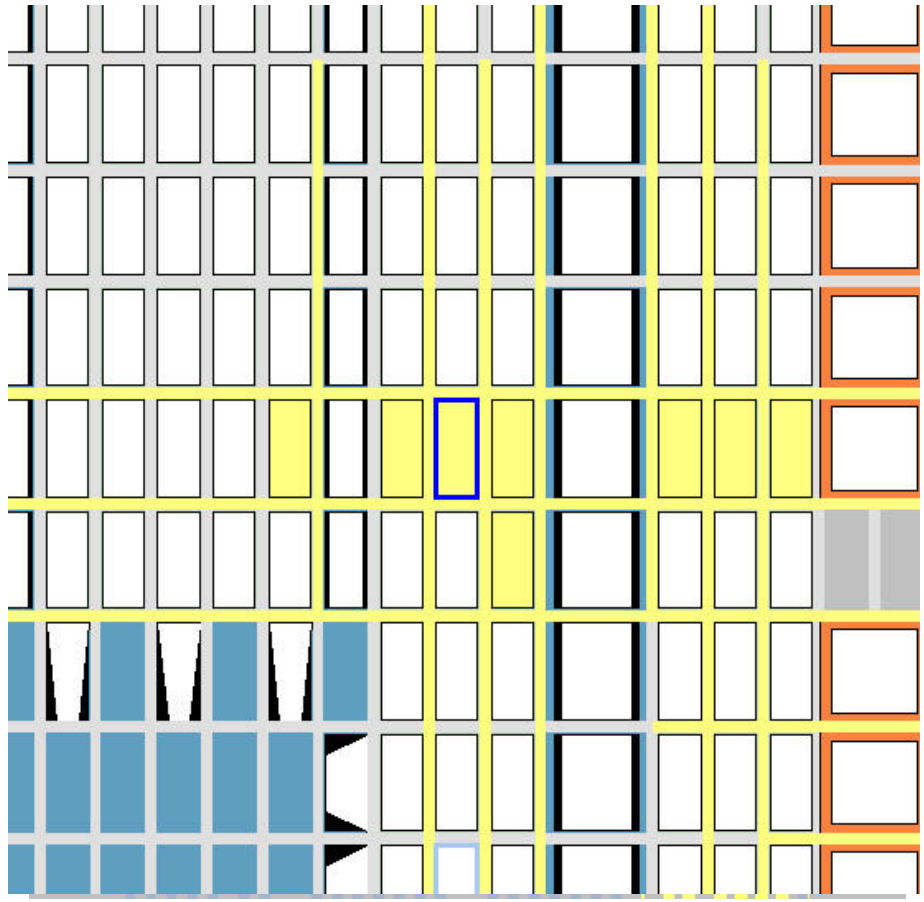


Chip Editor Floorplan

- Can be used to View Details of the Device Resources in an Altera FPGA
  - Device Routing Channels
  - Routing Paths Between Device Resources
    - Internal Routing Channels Within LABs
- Hierarchical Abstraction Level Used to View Device Resources
  - Higher Zoom-Level → Less Detail
- Launch the Resource Property Editor
  - Option in the Right-Mouse Click Menu

Chip Editor Views

- High Level View is Similar to the Timing Closure Floorplan View:
  - Used Device Resources are Shown in Yellow
  - Tooltips Available for all Device Resources Except Routing Channels
- Second Level Reveals Routing Channels in More Detail
  - Tooltips Available for All Routing Channels

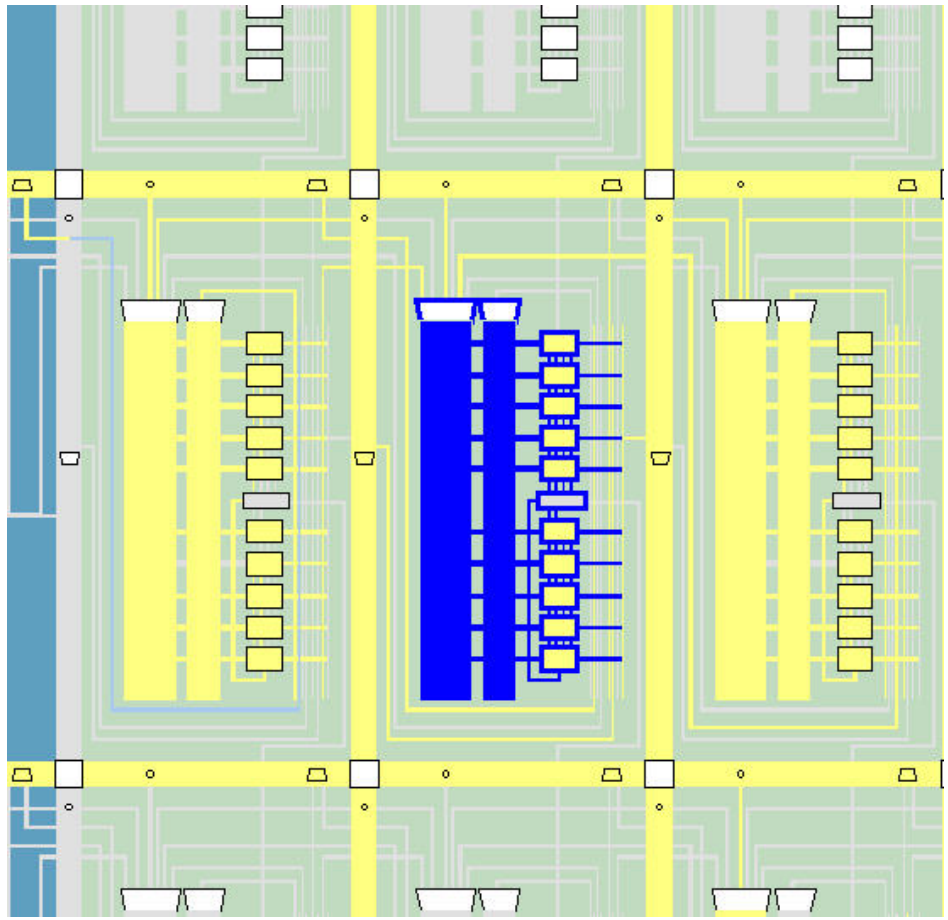


SOPC WORLD 2004

ALTERA.

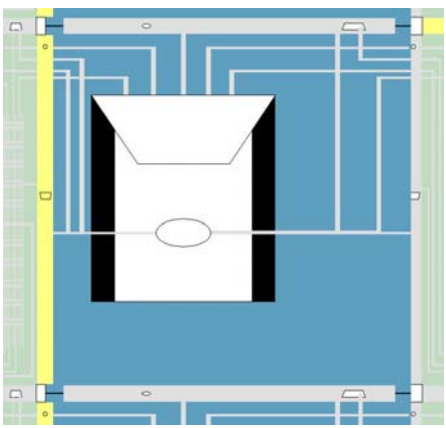
Chip Editor Views

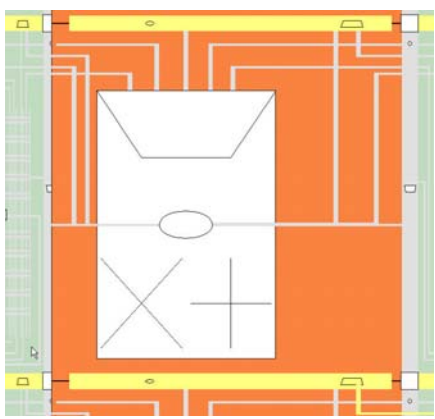
- Subsequent Zoom Levels Provide More Details
  - Routing Channels *in* and *out* of Device Resources Are Shown

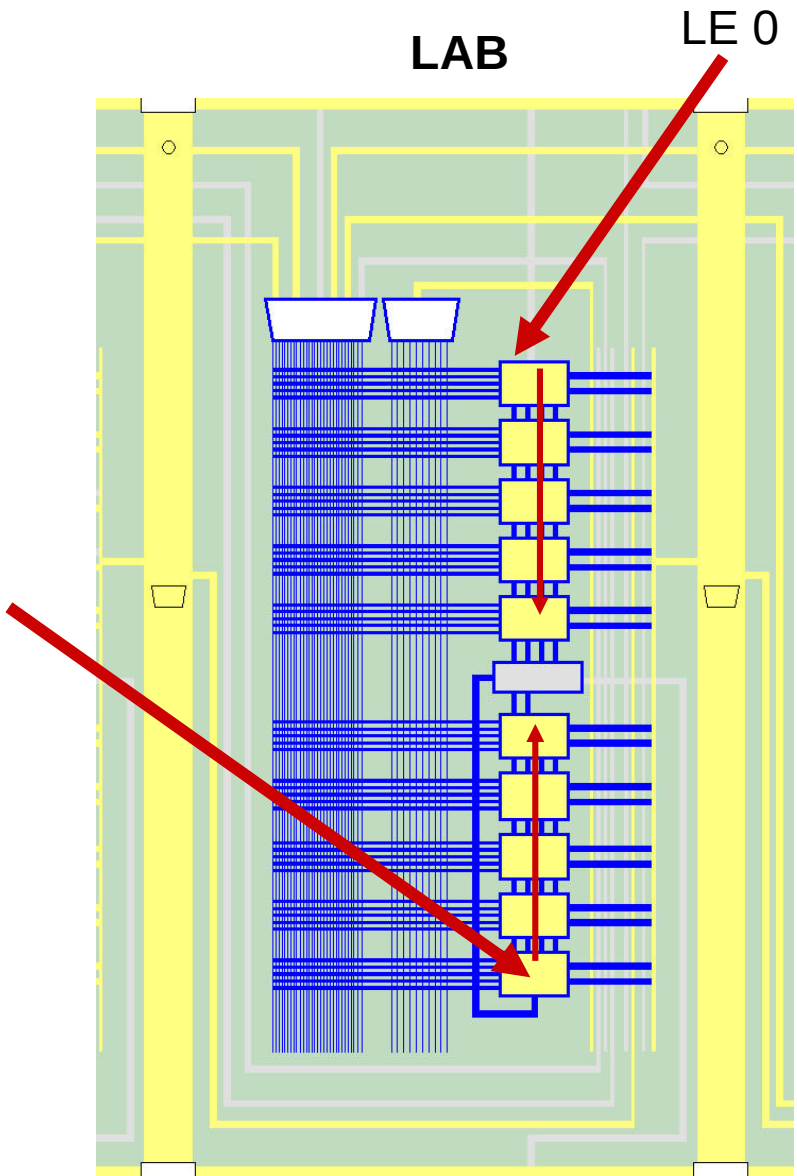


Chip Editor Views

- Bottom-Level View Reveals Internal Routing Channels *in* and *out* of LABs
  - LEs Are Shown as They Are in Silicon

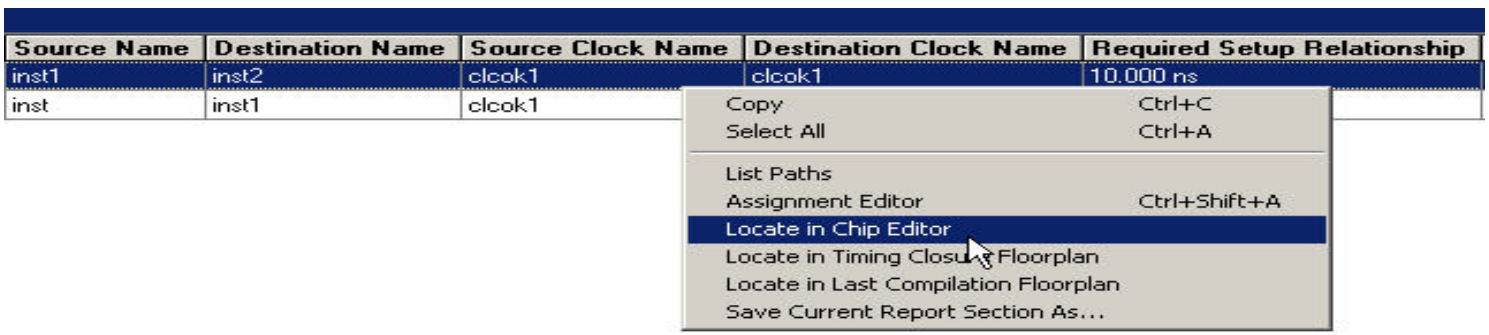




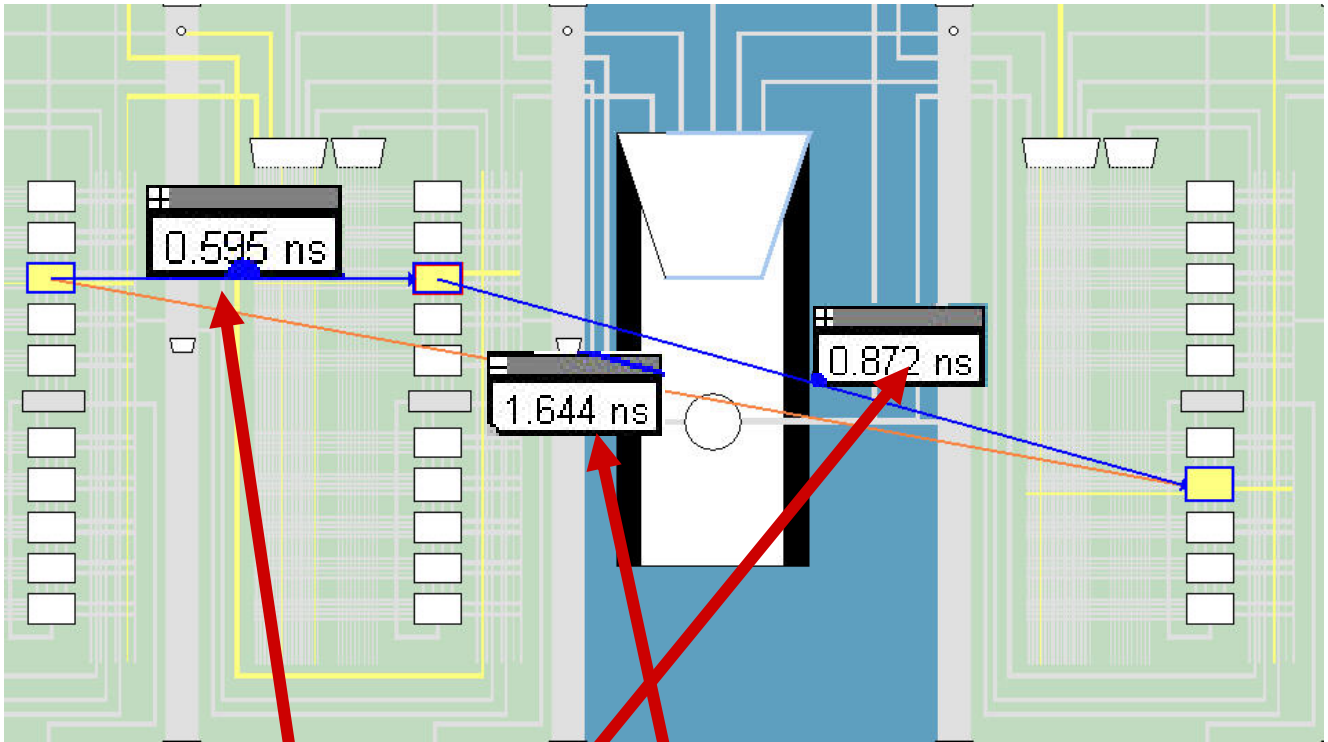


Example: From TAN Report

- From Timing Analysis Report
  - Traces Exact Path Between **Source** and **Destination** Registers
  - The Interconnect Delay & Total Delay Are Shown When the **Show Delay**  Option is Used



Example: From TAN Report (cont'd)



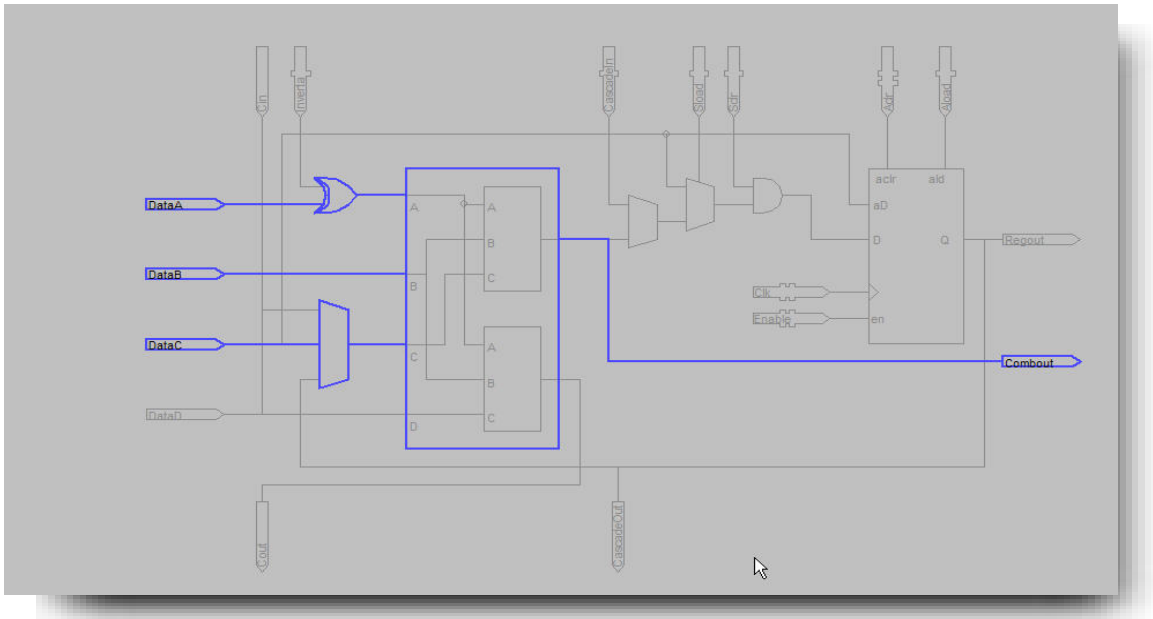
Info: - Longest register to register delay is 1.644 ns
Info: 1: + IC(0.000 ns) + CELL(0.000 ns) = 0.000 ns; Loc. = LC_X2_Y1_N2; REG Node = 'inst1'
Info: 2: + IC(0.595 ns) + CELL(0.087 ns) = 0.682 ns; Loc. = LC_X3_Y1_N2; COMB Node = 'inst11'
Info: 3: + IC(0.872 ns) + CELL(0.090 ns) = 1.644 ns; Loc. = LC_X5_Y1_N8; REG Node = 'inst2'
Info: Total cell delay = 0.177 ns
Info: Total interconnect delay = 1.467 ns

Agenda

- Introduction to the Chip Editor
- The Resource Property Editor
  - LE Editor
  - I/O Editor
  - PLL Editor
- The Chip Editor Tools
- Chip Editor Applications
  1. Design Analysis
  2. Design Flaw
  3. Timing Verification
  4. Last Minute ECO
  5. Using the PLL Editor

Logic Element Editor

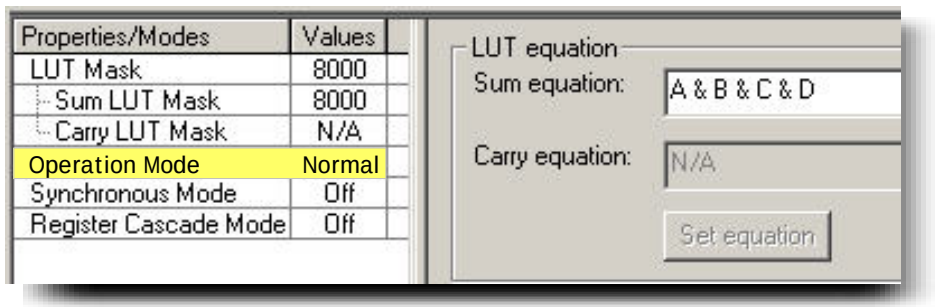
- Allows Changes to be Made to the Properties of an LE



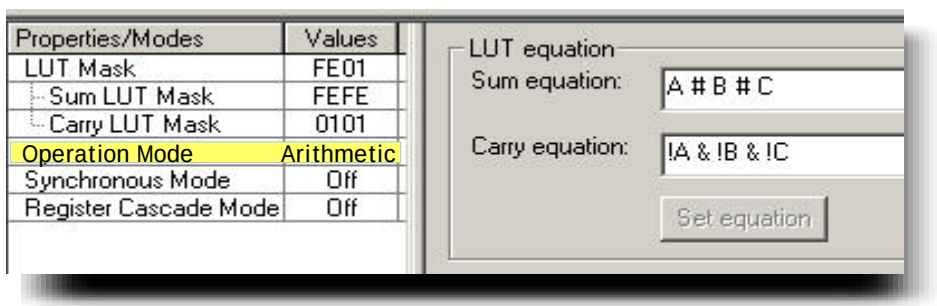
1. Locate LE in Chip Editor Floorplan
2. Select **Locate in Resource Property Editor**

LE Modes of Operations

- Operation Mode = {Arithmetic, Normal}
  - In **Normal Mode** the LUT is a Function of DATAA, DATAB, DATAC, & DATAD



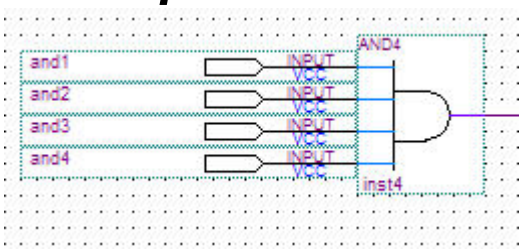
  - In **Arithmetic Mode** the LUT is a Function of DATAA, DATAB & CARRY IN
    - DATAD is not Connected



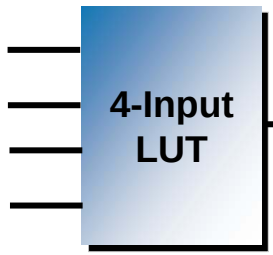
Legal Changes to an LE

1. LUT Equation
   - Can Only Use Inputs That Are Currently Utilized by the LE

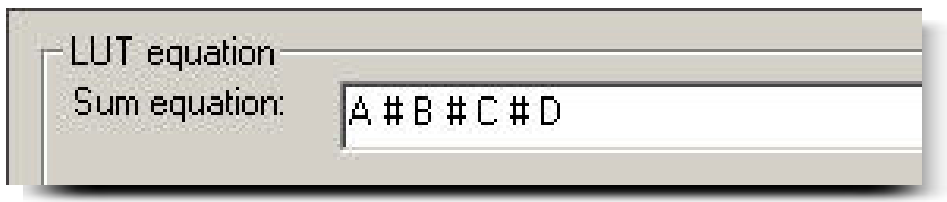
Example

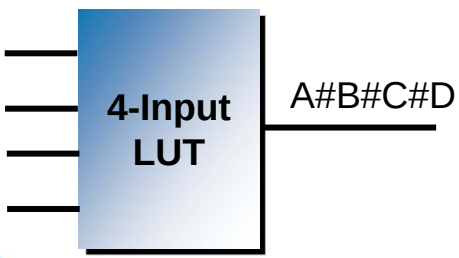


Quartus P&R



A&B&C&D



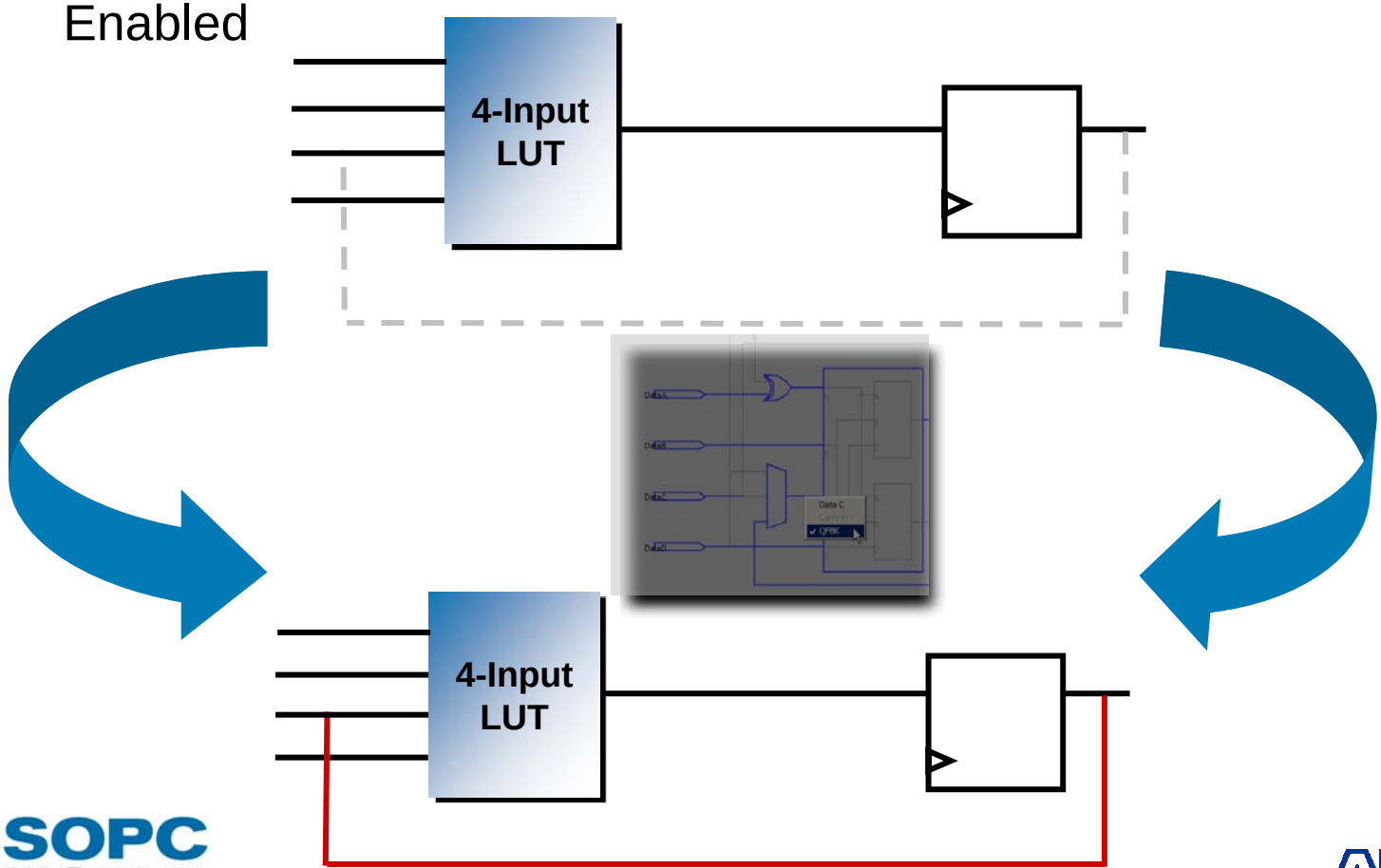


A#B#C#D

Actual Implementation

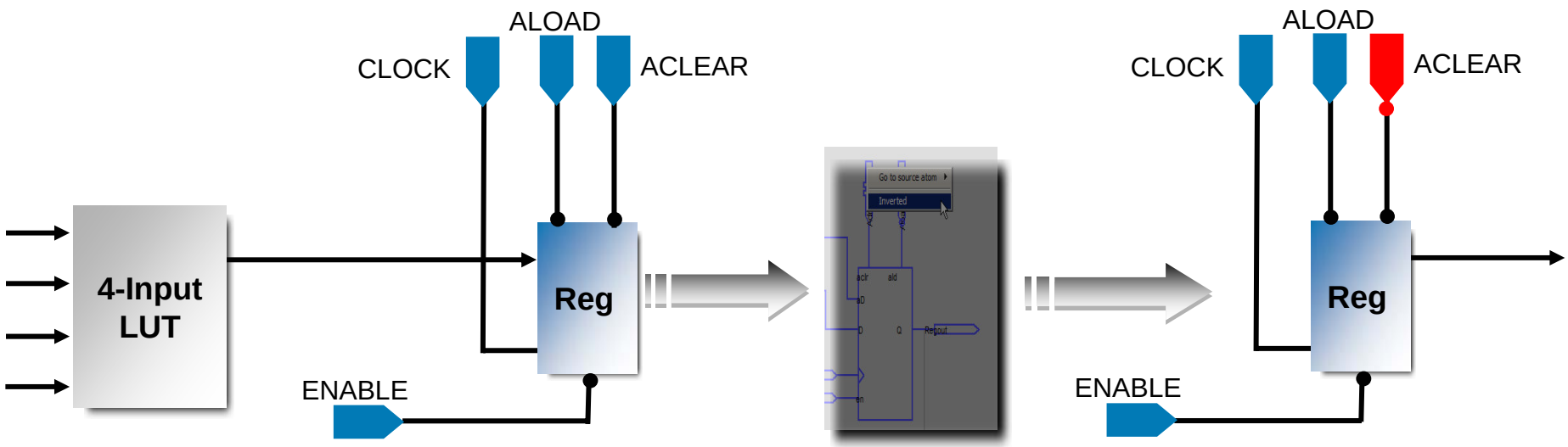
Legal Changes to an LE

2. Feedback Path in the LE
   - DATAC Can No Longer Be Utilized if the Feedback Path is Enabled



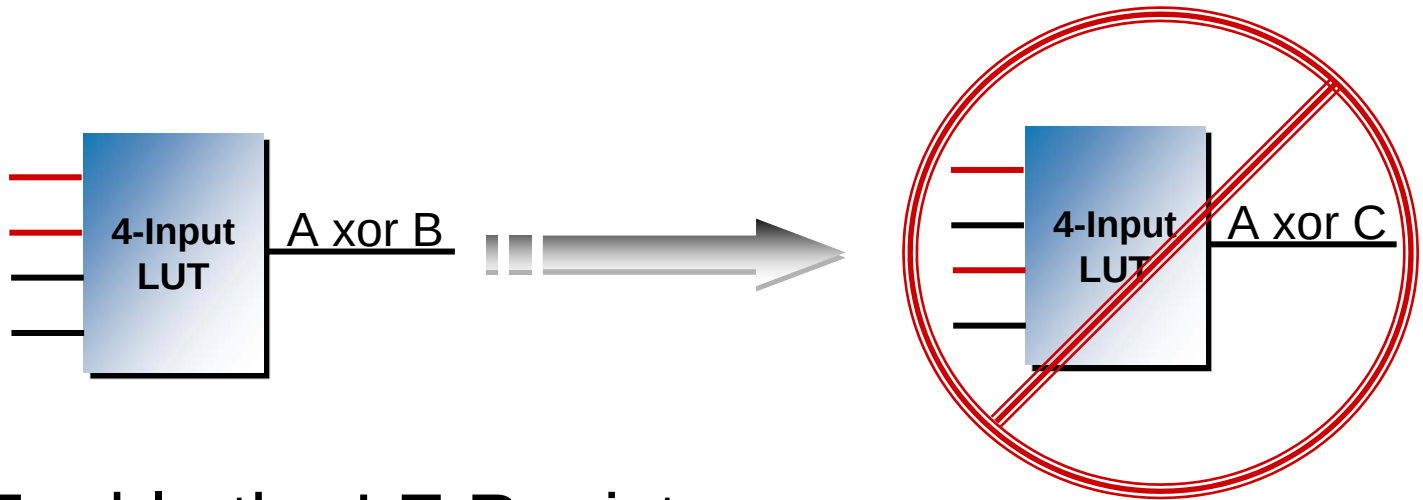
Legal Changes to an LE

3. Signal Inversion
   - Only ALOAD, ACLEAR, CLOCK, ENABLE are Invertible



Illegal Changes to an LE

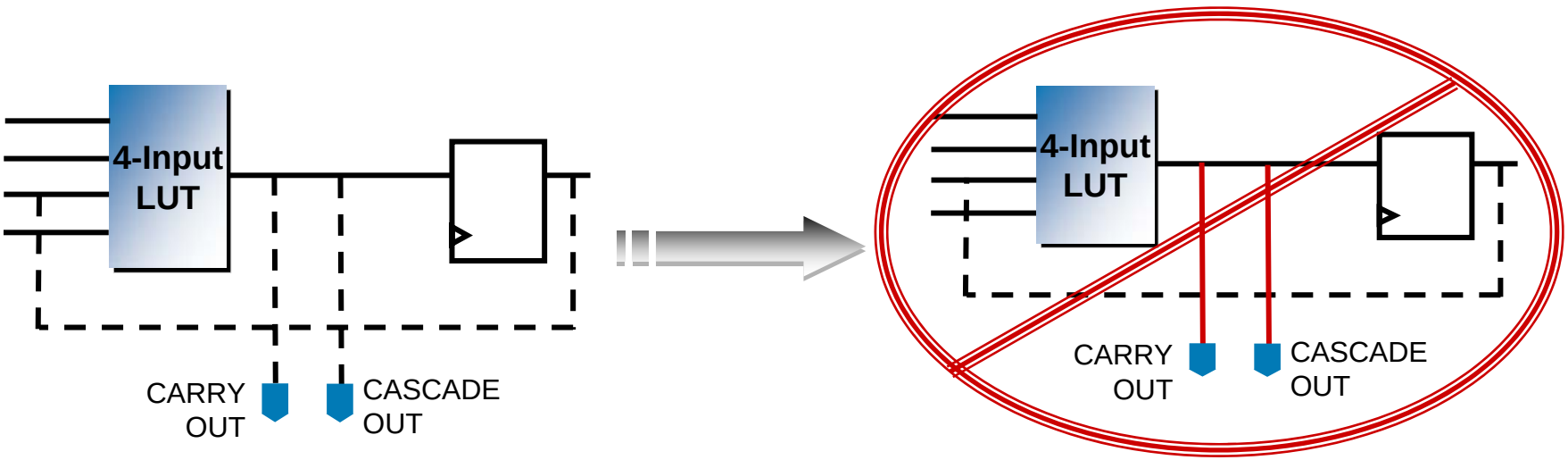
1. LUT Rotation
   - LUT Inputs Cannot be Swapped
   - Unused Inputs Cannot be Added to the LUT Equation



2. Enable the LE Register
   - Unused Register Can Not Be Enabled

Illegal Changes to an LE

3. Manual Routing
   - Unused Routes Out of the LE Can Not Be Enabled
     - COMBOUT, CARRY OUT, CASCADE OUT

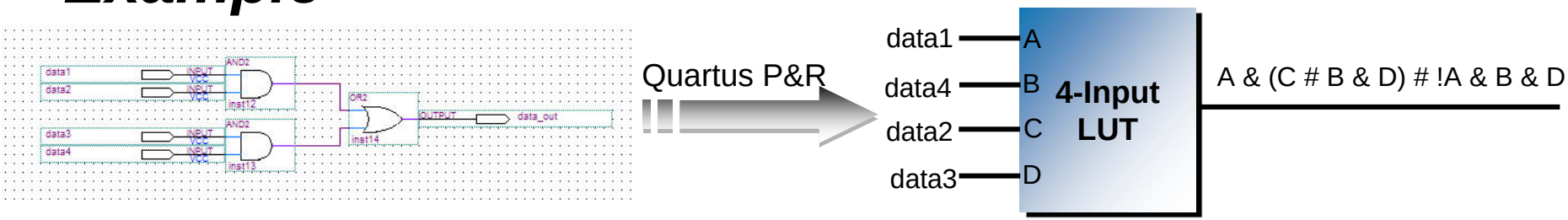


4. Disable the Feedback Path in an LE
   - If the Feedback Path is Enabled with the Chip Editor, Re-fit the Design to Wipe out Change

Caveats of the LUT Equation

- The Logic That Was Implemented May Have Been Optimized and Absorbed into an LE

Example

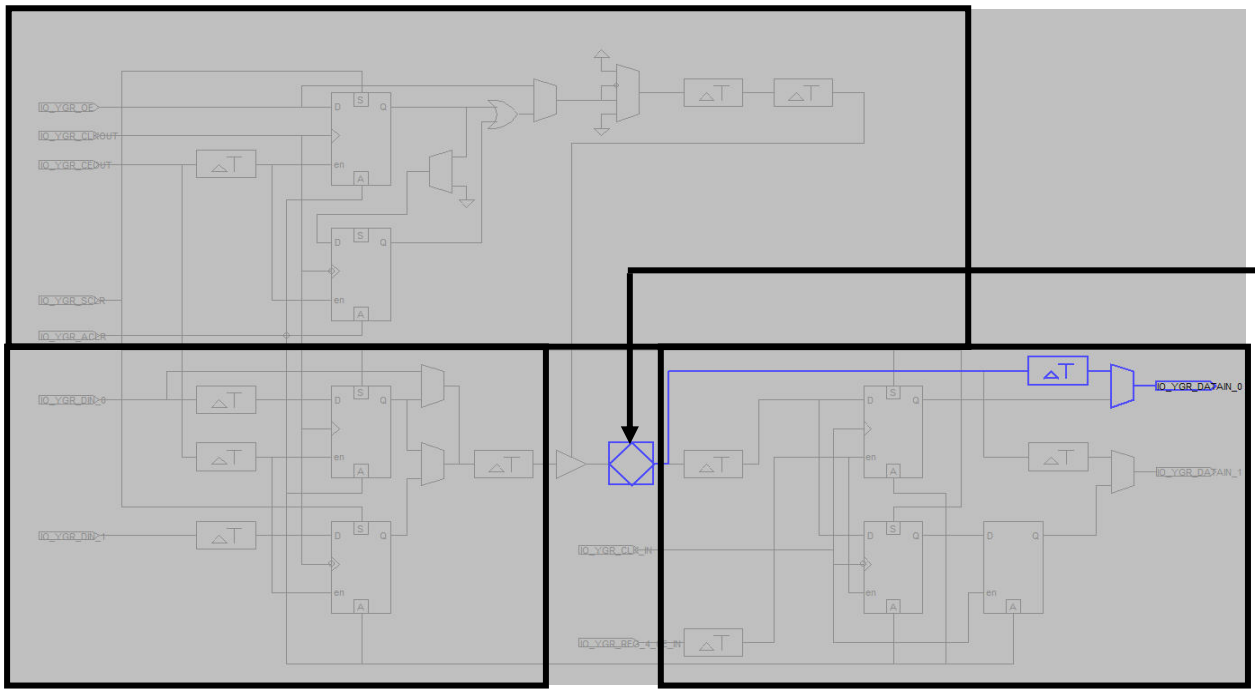


Only use the LUT Equation if ….

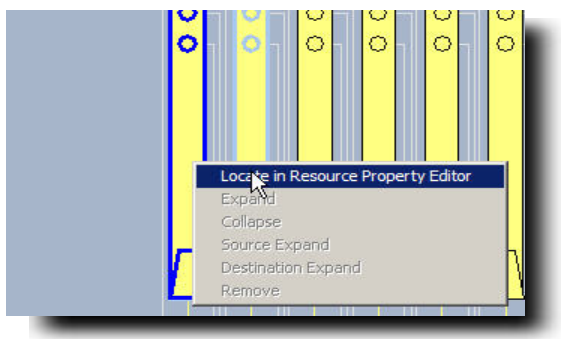
1. You are Very Familiar with the Design
2. You Understand the Optimization that Quartus II Performed

I/O Editor

- Allows I/O Properties to Be Modified

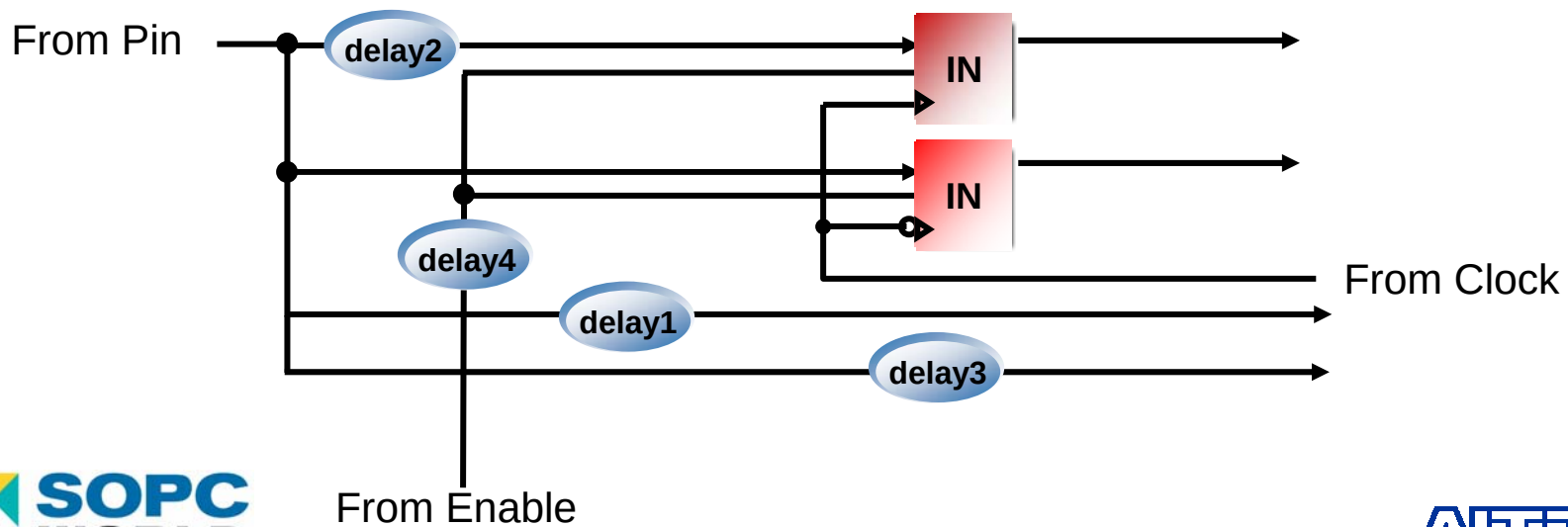


1. Locate I/O in Chip Editor
2. Select **Locate in Resource Property Editor**



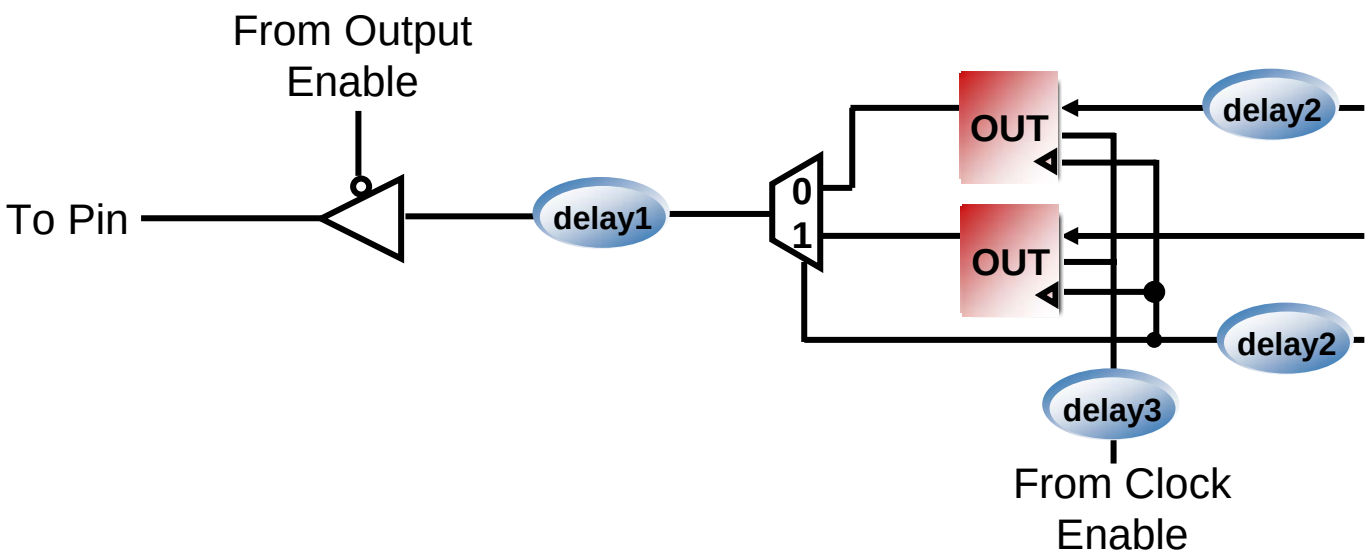
Legal Changes – I/O Timing

- Input Delay Options
  1. Input Pin to Logic Array 0
  2. Input Pin to Input Register Delay
  3. Input Pin to Logic Array 1
  4. Input Clock Enable Delay



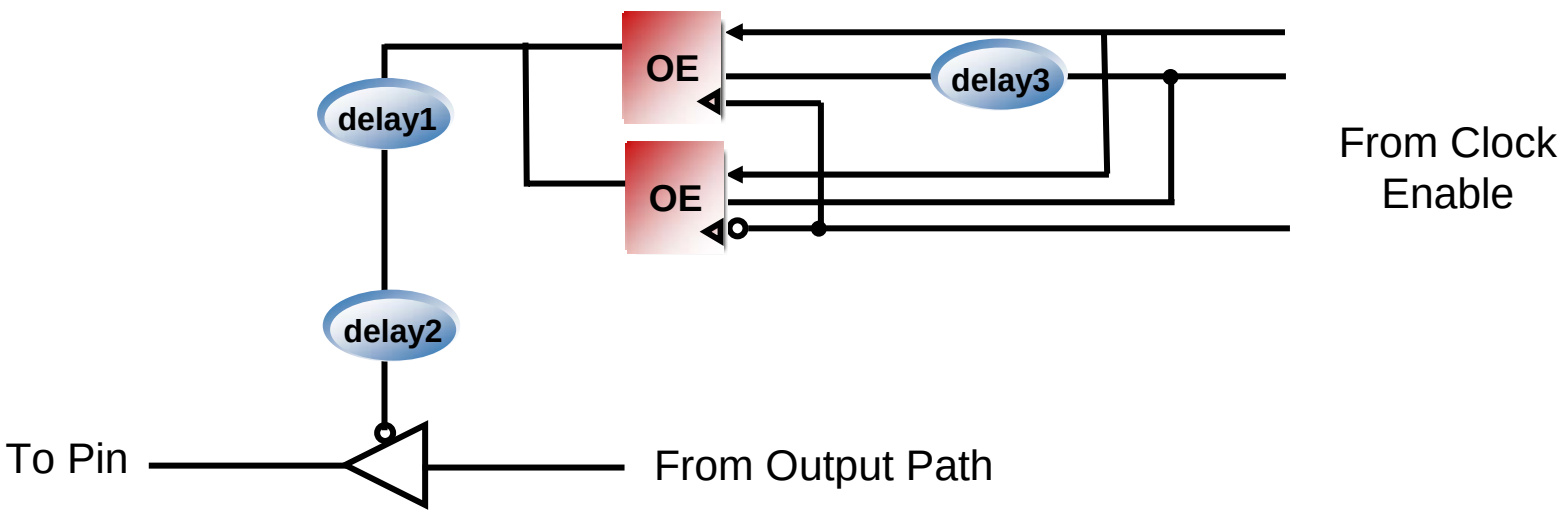
Legal Changes – I/O Timing (cont'd)

- Output Delay Options
  1. Output Pin Delay
  2. Logic Array to Output Register Delay
  3. Output Clock Enable Delay



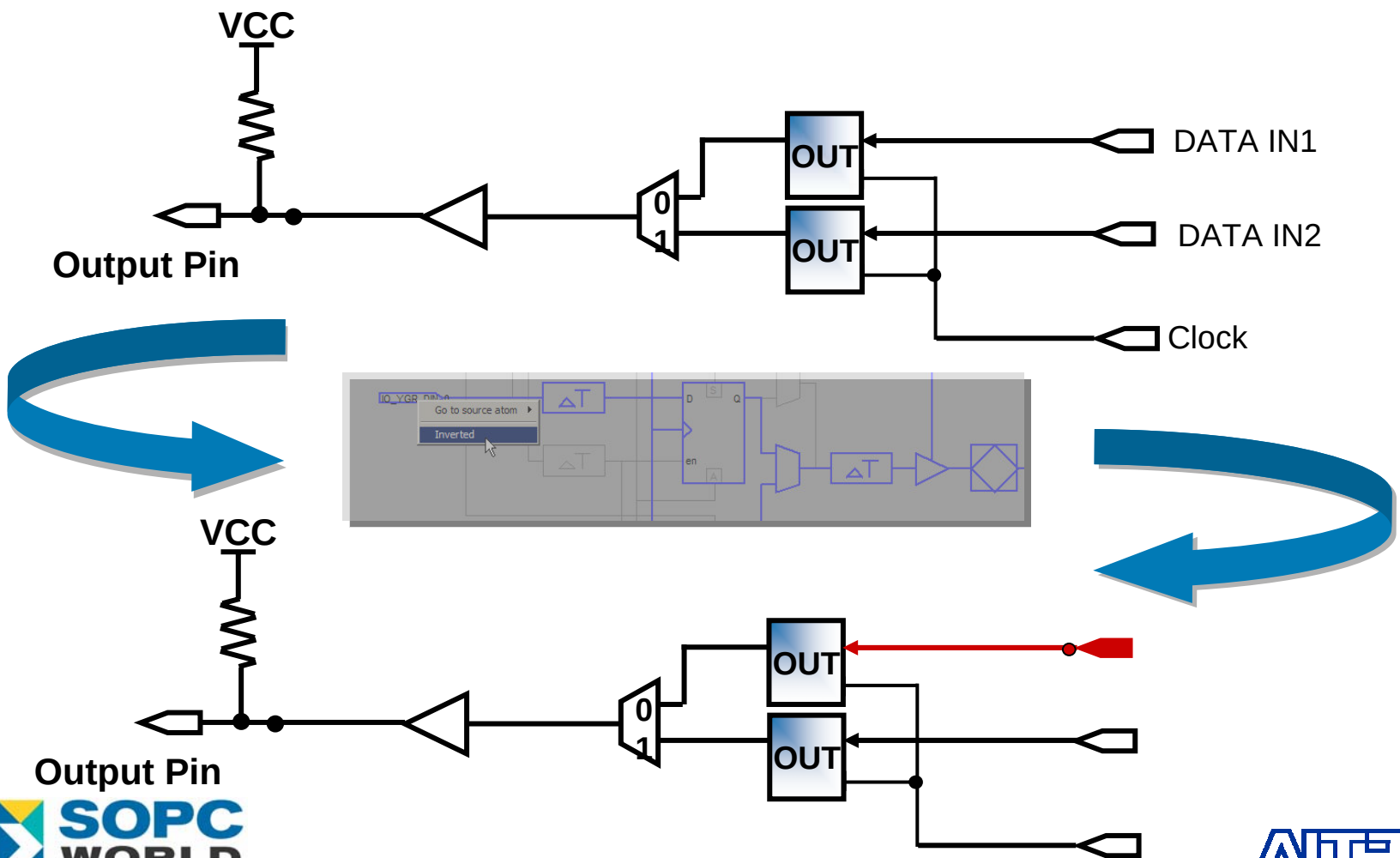
Legal Changes – I/O Timing (cont'd)

- Output Enable Delay Options
  1. Output Enable to Register TCO Delay
  2. Output TZX Delay
  3. Output Enable Clock Enable Delay



Other Legal Changes – I/O Editor

- Signal Inversion



Other Legal Changes – I/O Editor

- Reset/Clear Options for Sync/Async Signals

From/To	None	Reset	Preset
None		No	No
Reset	Yes		Yes
Preset	Yes	Yes	

- Power Up Options
  - High or Low for All Registers in the I/O

Illegal Changes with the I/O Editor

- Adjusting the Current Strength That is Not Supported By the I/O Standard

I/O Standard	Legal Setting	Notes
3.3V LVTTL	24, 16, 12, 8, 4	
3.3V LVCMOS	24, 12, 8, 4, 2	
2.5V	16, 12, 8, 2	LVTTL/LVCMOS
1.8V	12, 8, 2	LVTTL/LVCMOS
1.5V LVCMOS	8, 4, 2	

- Enabling the Register
  - If the Register is Not Used in the LE it Can Not be Enabled

Agenda

- Introduction to the Chip Editor
- The Resource Property Editor
  - LE Editor
  - I/O Editor
- The Chip Editor Tools
- Chip Editor Applications
  1. Design Analysis
  2. Design Flaw
  3. Timing Verification
  4. Last Minute ECO
  5. Using the PLL Editor

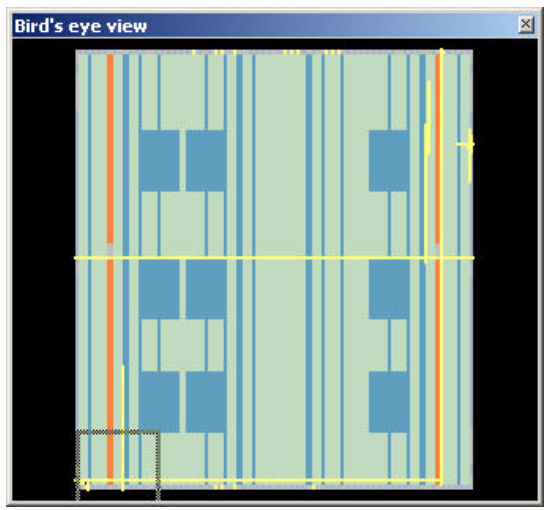
Chip Editor Toolbar Options

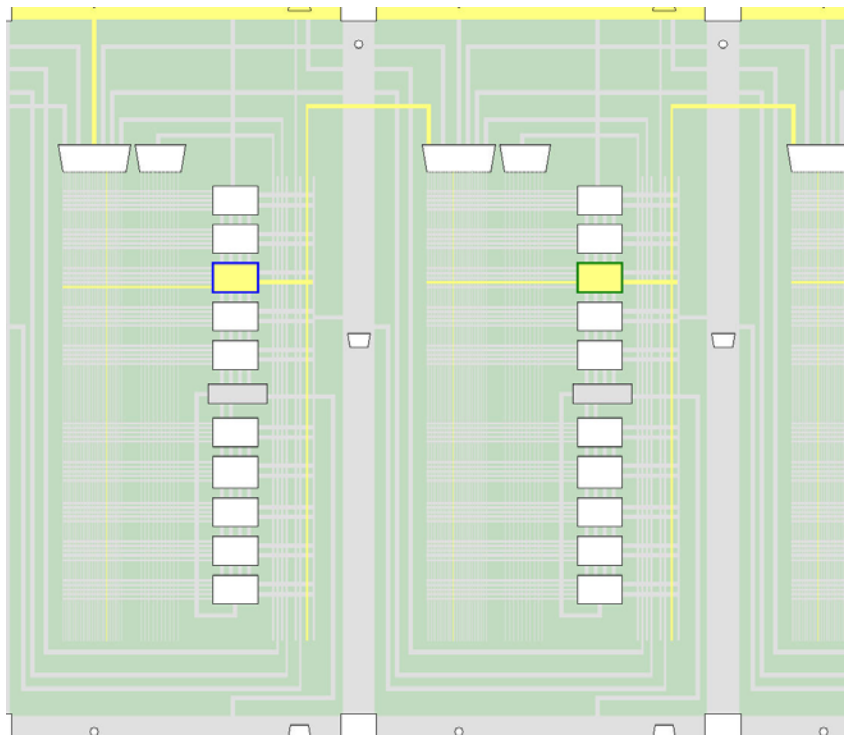
- Many New Options Exist That Allows Easily Navigation within the Chip Editor
  - Bird's Eye View
  - Route Fan-In and Fan-out
  - Detailed Tooltips

- Accessible Through the Chip Editor's Toolbar

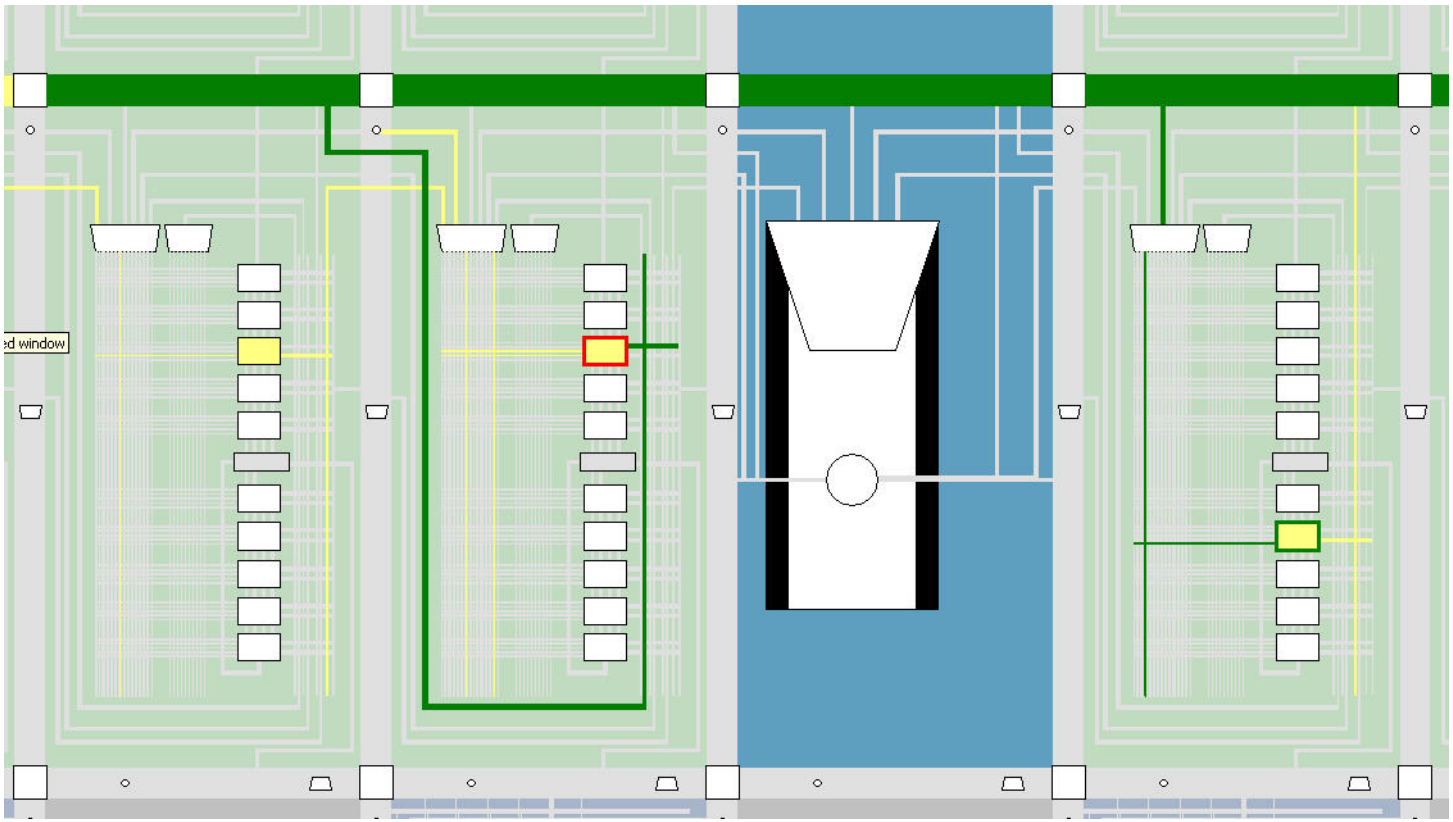
Bird’s Eye View

- Bird’s Eye View
  - Provides an Overall View of the Entire Device
  - Used to Navigate Through the Chip Editor Floorplan



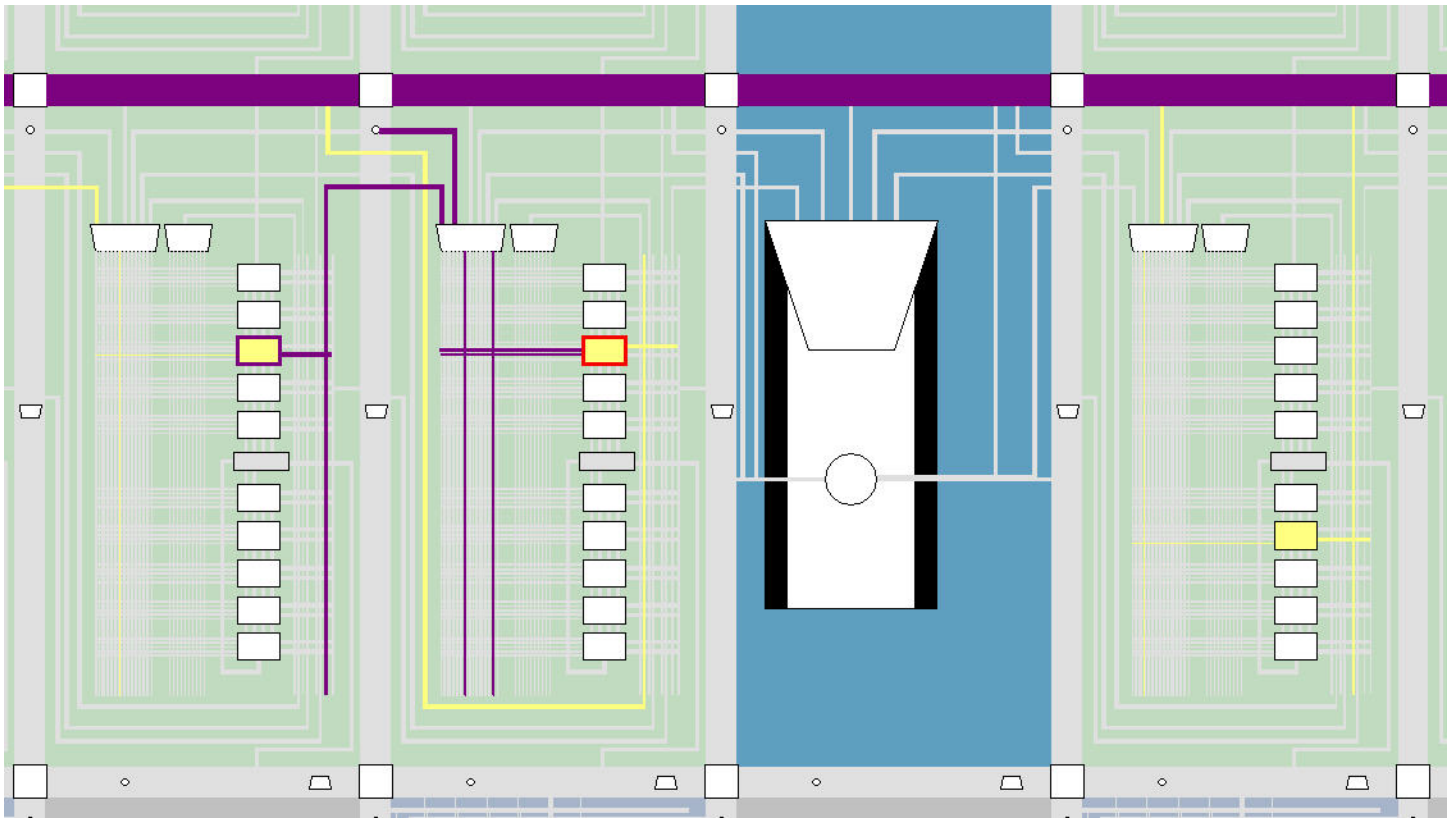


Route Fan-Out



- Show Routing Fan-Out 

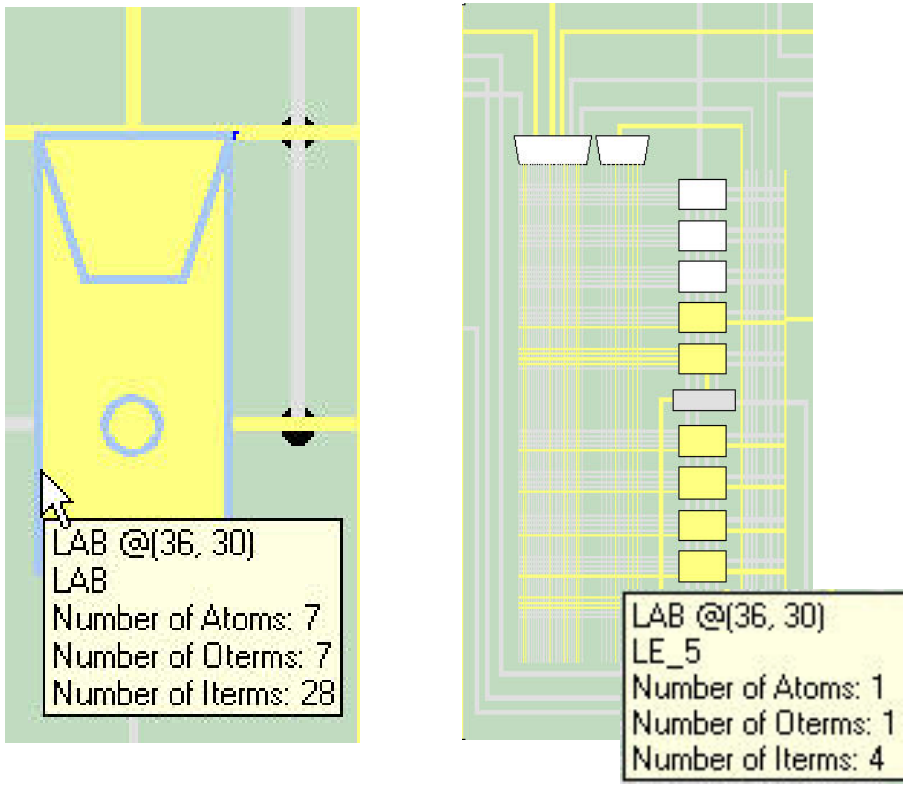
Route Fan-In



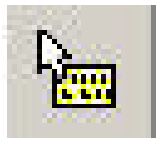
- Show Routing Fan-In

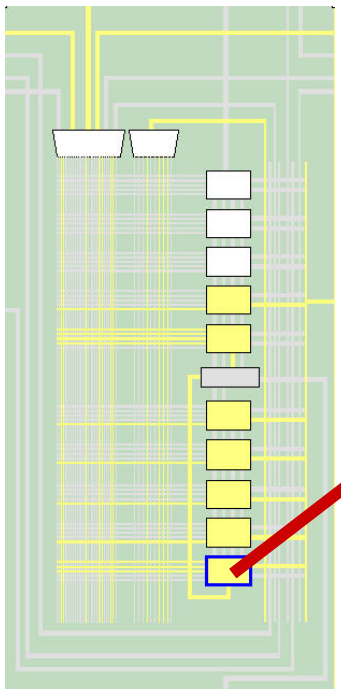
Tooltips in the Chip Editor

- Tooltips in the Chip Editor Provide General Information Regarding the Selected ATOM
  - Device Resource Name, e.g. LAB, M4K, etc
  - Location
  - Number of Atoms
  - Number of Oterms
  - Number of Iterms



Detailed Tooltips

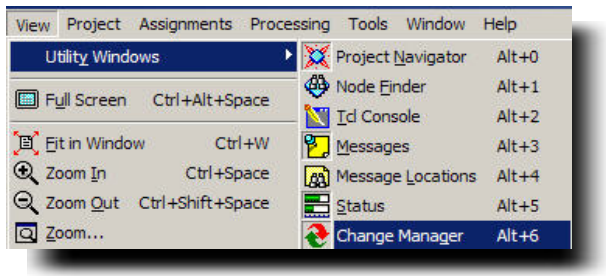
- Detailed Tooltips are Enabled with the  Icon
  - Provides the Names of all ATOMS, Oterms, and Iterms Assigned to the Device Resource



```
LAB @(36, 30)
LE_5
Atoms Assigned:
  |Block1|my_mux:inst7|lpm_mux:lpm_mux_component|mux_s9c:auto_generated|w_result15w~85
Oterms Assigned:
  w_result15w~85
Iterms Assigned:
  w_result15w~85_DATAA[1]
  w_result15w~85_DATAB[2]
  w_result15w~85_DATAC[3]
  w_result15w~85_DATAD[4]
```

Change Manager

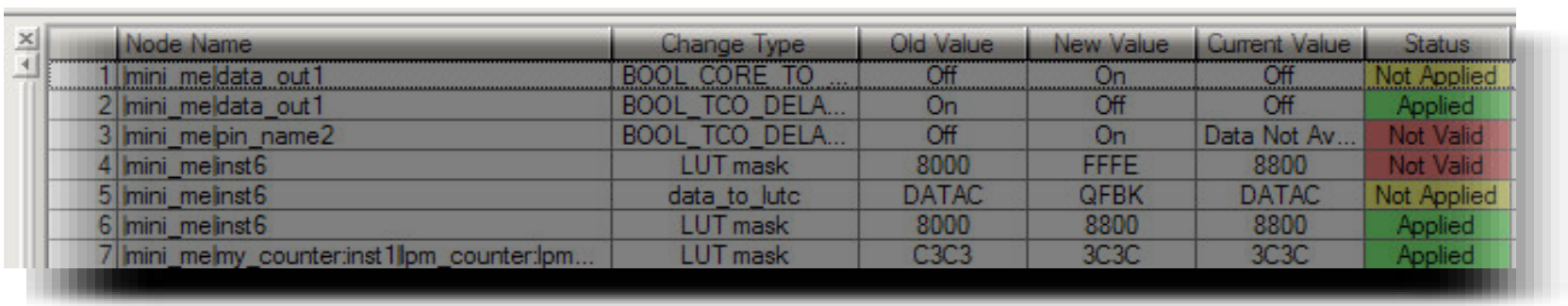
- A Utility that Reports all Changes Made with the Chip Editor
  - Change Manager (View menu)



- A Number of Operations Can Be Performed Within the Change Manager
  - Restore Old Value
  - Export to Tcl
  - Locate ATOM in the Resource Property Editor
  - Check & Save Netlist

The Change Manager (cont'd)

Column	Description
Node Name	Name of the Node that was Modified with Chip Editor
Change Type	Old Value of the Modified Node
Current Value	New Value of the Modified Node
Status	Current State of the Change Made to the Specified Node

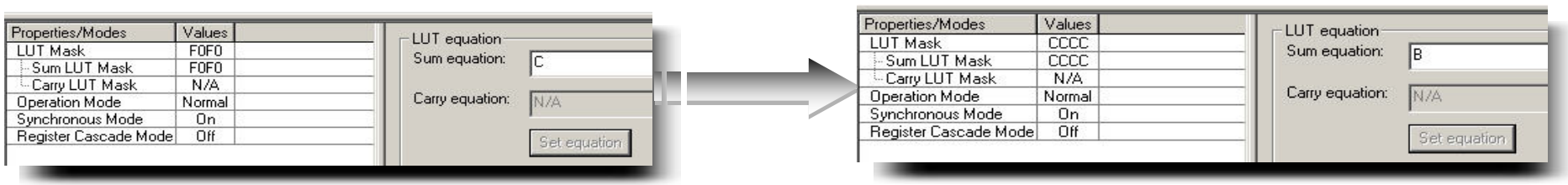


	Node Name	Change Type	Old Value	New Value	Current Value	Status
1	|mini_me|data_out1	BOOL_CORE_TO_...	Off	On	Off	Not Applied
2	|mini_me|data_out1	BOOL_TCO_DELA...	On	Off	Off	Applied
3	|mini_me|pin_name2	BOOL_TCO_DELA...	Off	On	Data Not Av...	Not Valid
4	|mini_me|inst6	LUT mask	8000	FFFE	8800	Not Valid
5	|mini_me|inst6	data_to_lutc	DATAC	QFBK	DATAC	Not Applied
6	|mini_me|inst6	LUT mask	8000	8800	8800	Applied
7	|mini_me|my_counter:inst1|lpm_counter:lpm...	LUT mask	C3C3	3C3C	3C3C	Applied

ATOM Level Design Rule Checker

- Ensures Changes Made to an ATOM Do Not Violate Any of the Device and/or Software Constraints

Example : Modify the LUT Equation to Include Unused Inputs

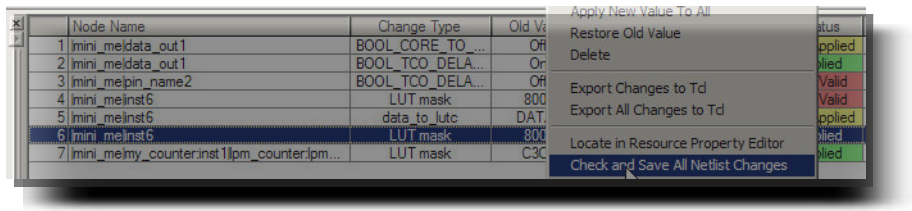


Result : Error: LCELL ATOM is Dependent on Unconnected Input Ports

- ATOM Level Checker Runs Automatically When the ATOM Level Changes Are Saved

Design Rule Checker - Circuit Level

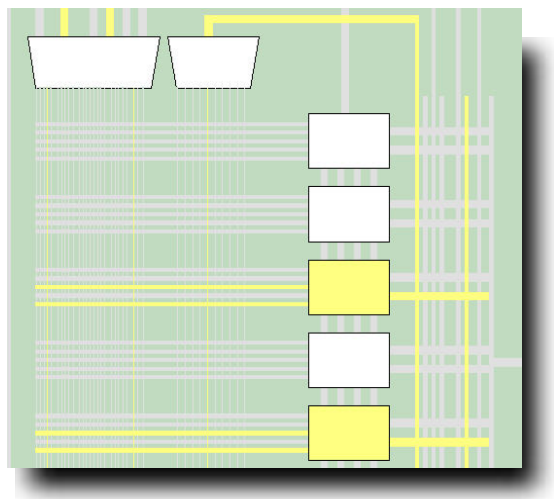
- Ensures that All Independent Changes to the ATOM Do Not Result in an Illegal Circuit



- Cases Exist When the ATOM Level DRC Passes, but the Circuit Level DRC Fails

Example: *Two LEs in a LAB*

Invert the SLOAD for One but Not the Other



Result: *Error: LAB Has 2 sload Signals, but Only 1 Signal is Sllowed*

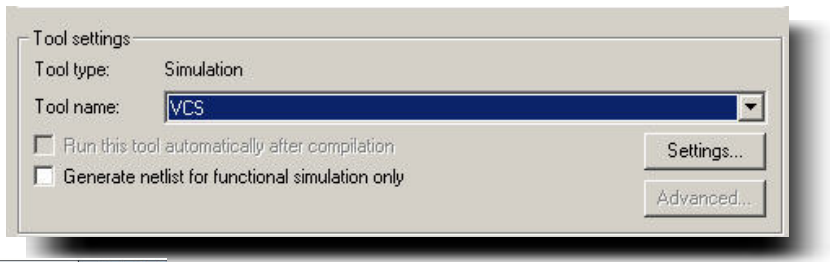
Cause: *SLOAD Signal in a LAB is a Single Wire*

SOPC WORLD 2004

ALTERA.

Post - Chip Editor Options

- Simulation
  - Generate VO/VHO After Changes are Made with Chip Editor

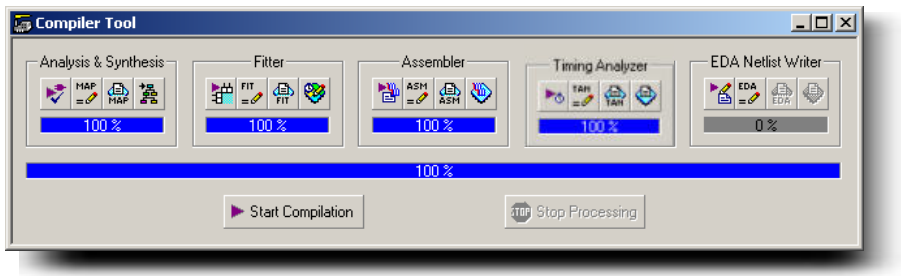


- Allows Gate-Level Simulation to Be Performed with New Changes in a Third-Party Simulation Tool

Post - Chip Editor Options (cont'd)

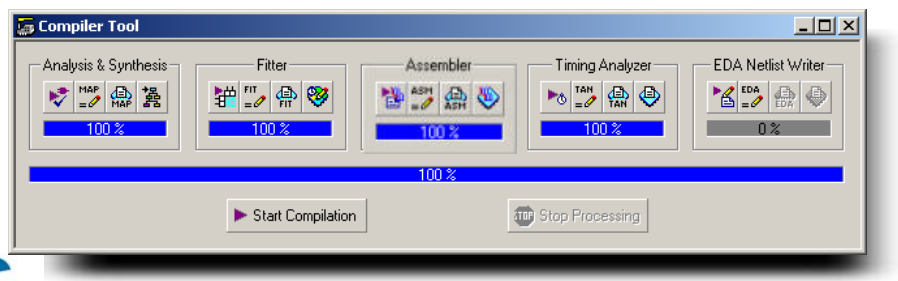
3. Quartus Timing Analysis

- Verify Timing After Changes Are Made with the Chip Editor



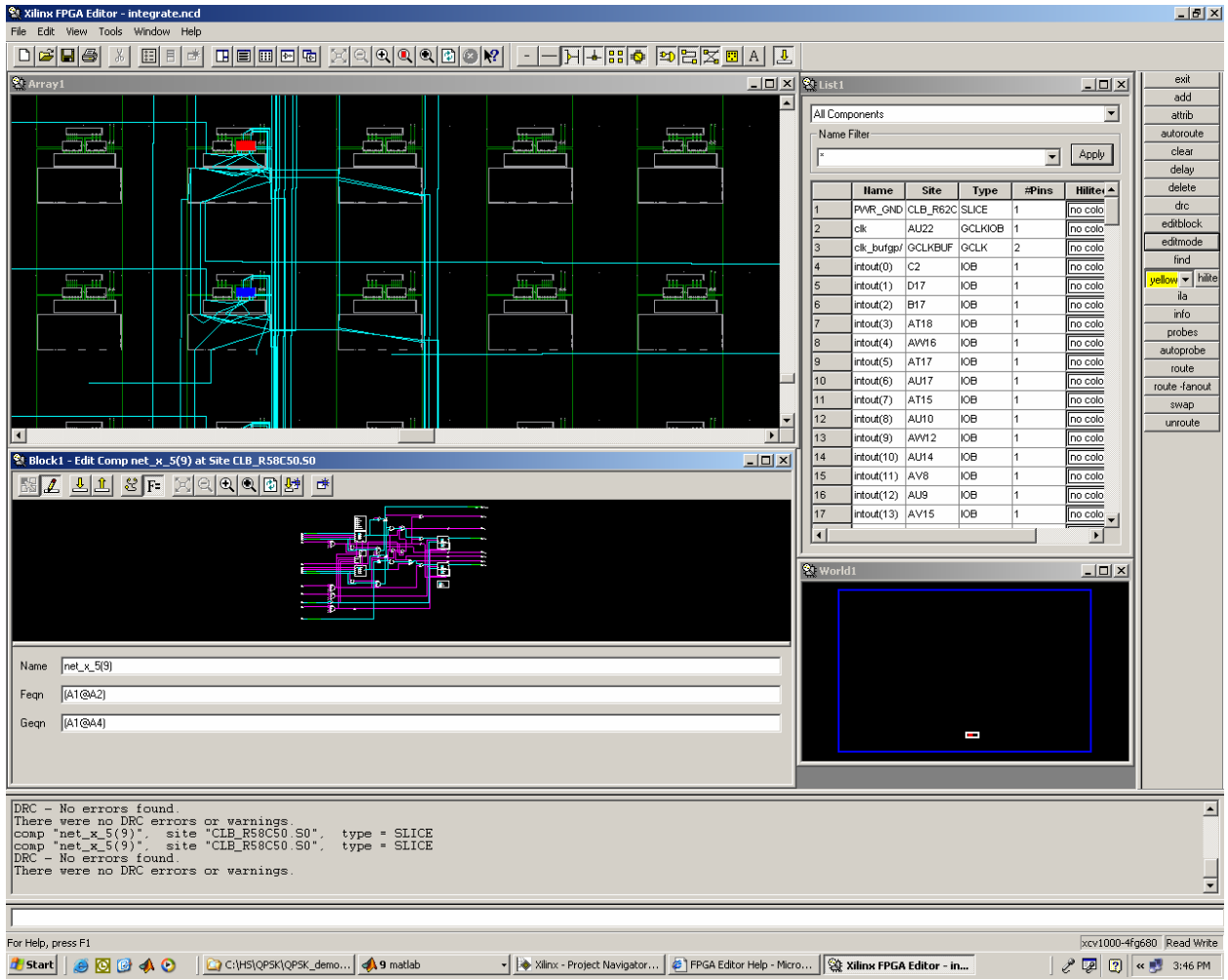
4. Quartus Assembler

- Verify Circuit Behavior on the PCB



Xilinx FPGA Editor

- Advantages:
  - Capabilities far Exceed Those in Chip Editor
  - Able to Modify / Create Anything with the Editor
- Disadvantages:
  - No Way to Track Changes
  - Very Little Interaction Between TAN and Editor



SOPC WORLD 2004

ALTERA

Agenda

- Introduction to the Chip Editor
- The Resource Property Editor
  - LE Editor
  - I/O Editor
  - PLL Editor
- The Chip Editor Tools
- Chip Editor Applications
  1. Design Analysis
  2. Design Flaw
  3. Timing Verification
  4. Last Minute ECO
  5. Using the PLL Editor

Application: Design Analysis

Problem:

Clock Setup: 'clk'

	Slack	Actual fmax (period)	Source Name	Destination Name	Source Clock Name	Destinat
1	-0.279 ns	106.73 MHz (period = 9.369 ns)	taps:inst|xn[0]~reg0	acc:inst3|result[11]	clk	clk
2	-0.182 ns	107.85 MHz (period = 9.272 ns)	state_m:inst1|filter~10	acc:inst3|result[11]	clk	clk
3	-0.177 ns	107.91 MHz (period = 9.267 ns)	taps:inst|xn[0]~reg0	acc:inst3|result[10]	clk	clk
4	-0.177 ns	107.91 MHz (period = 9.267 ns)	taps:inst|xn[0]~reg0	acc:inst3|result[9]	clk	clk
5	-0.177 ns	107.91 MHz (period = 9.267 ns)	taps:inst|xn[0]~reg0	acc:inst3|result[8]	clk	clk
6	-0.177 ns	107.91 MHz (period = 9.267 ns)	taps:inst|xn[0]~reg0	acc:inst3|result[7]	clk	clk
7	-0.177 ns	107.91 MHz (period = 9.267 ns)	taps:inst|xn[0]~reg0	acc:inst3|result[6]	clk	clk
8	-0.080 ns	109.05 MHz (period = 9.170 ns)	state_m:inst1|filter~10	acc:inst3|result[10]	clk	clk
9	-0.080 ns	109.05 MHz (period = 9.170 ns)	state_m:inst1|filter~10	acc:inst3|result[9]	clk	clk
10	-0.080 ns	109.05 MHz (period = 9.170 ns)	state_m:inst1|filter~10	acc:inst3|result[8]	clk	clk
11	-0.080 ns	109.05 MHz (period = 9.170 ns)	state_m:inst1|filter~10	acc:inst3|result[7]	clk	clk
12	-0.080 ns	109.05 MHz (period = 9.170 ns)	state_m:inst1|filter~10	acc:inst3|result[6]	clk	clk
13	0.101 ns	111.25 MHz (period = 8.989 ns)	state_m:inst1|filter~12	acc:inst3|result[11]	clk	clk
14	0.116 ns	111.43 MHz (period = 8.974 ns)	taps:inst|xn[0]~reg0	acc:inst3|result[5]	clk	clk

How do I Examine the Route in my Critical Path?

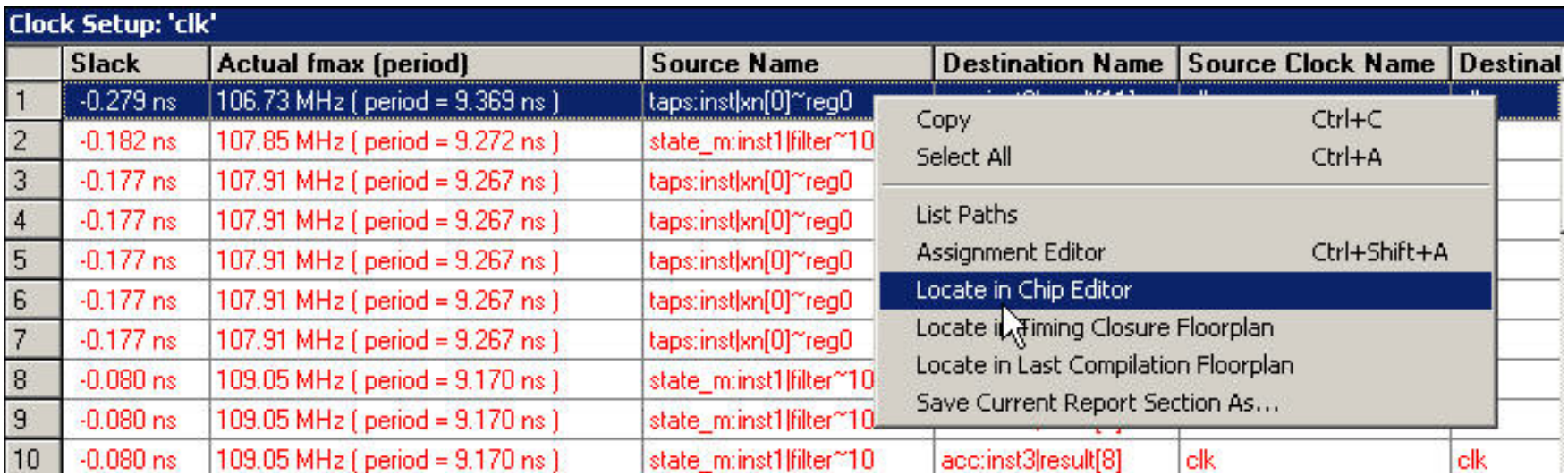
What Route Resources did Quartus II use to Connect Between the Source and Destination Nodes?

SOPC WORLD 2004

ALTERA.

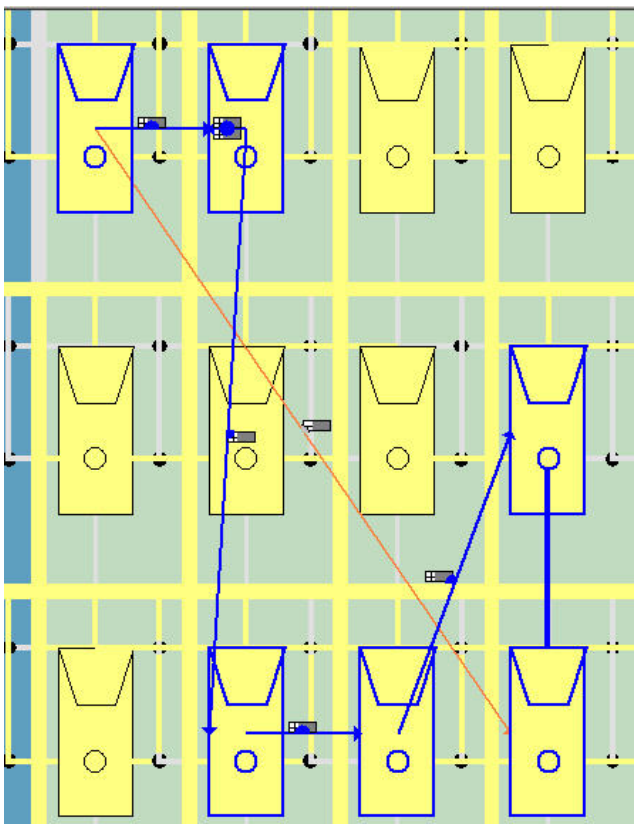
Design Analysis - Solution

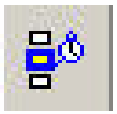
- Right-Mouse Click on Path
  - Select **Locate in Chip Editor**

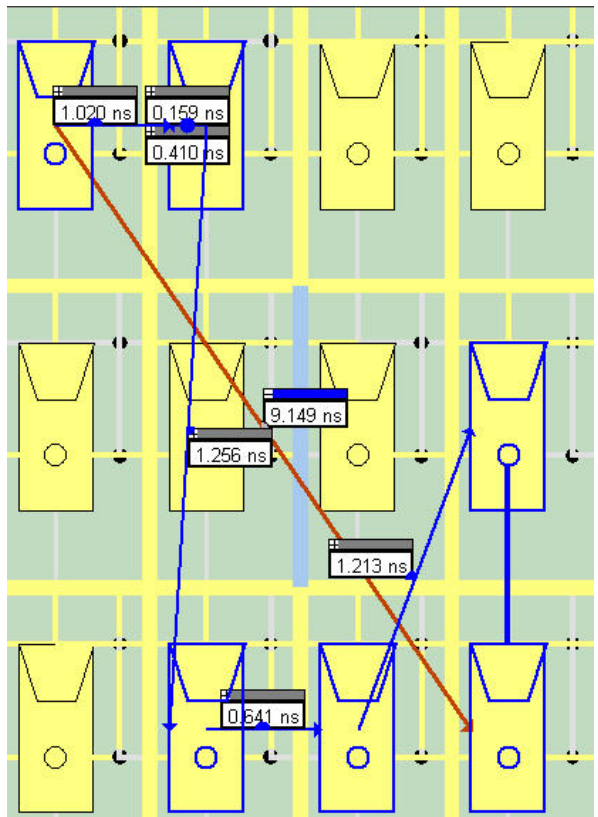


Design Analysis - Solution

- Locate in Chip Editor



- Enable Timing Delay 

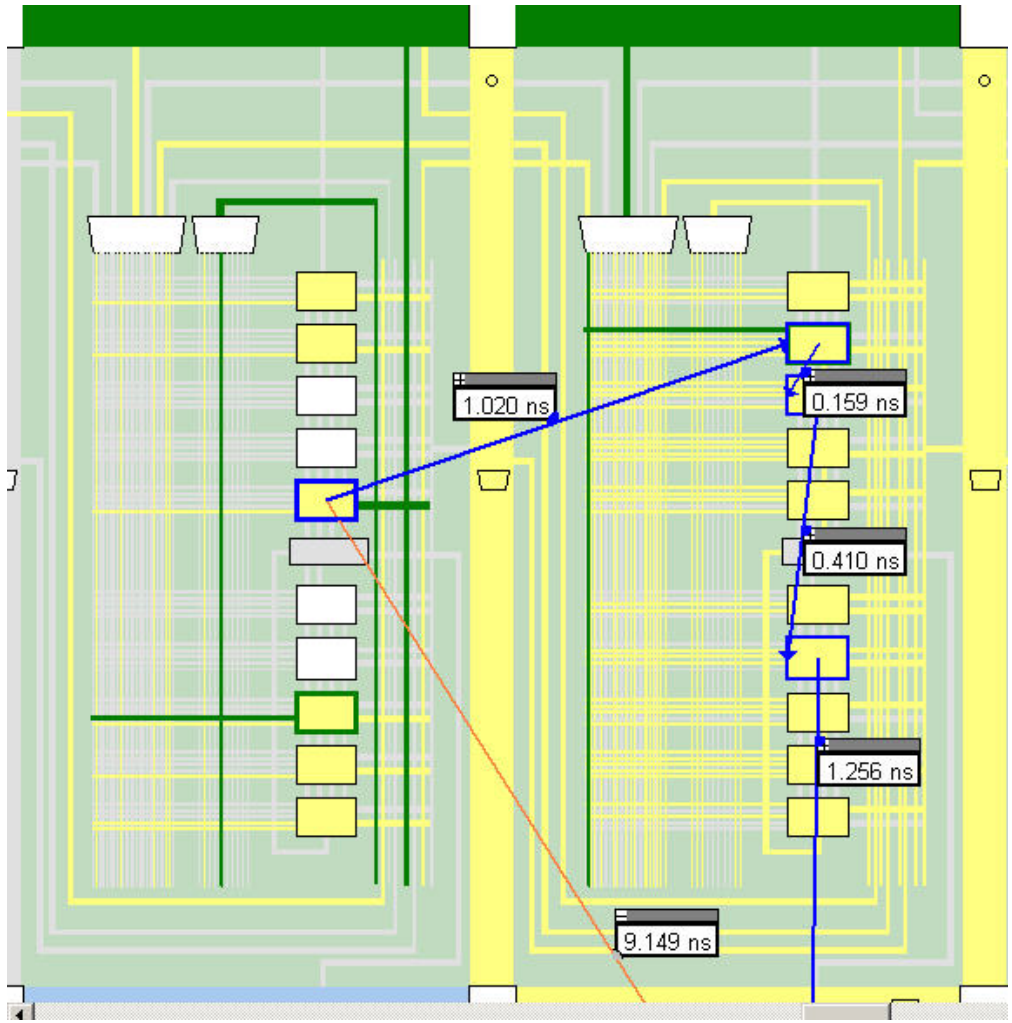


SOPC WORLD 2004

ALTERA.

Design Analysis - Solution

- Select LE_4 and Enable the **Route Fan-Out**
- The Exact Route is Displayed with Timing Values



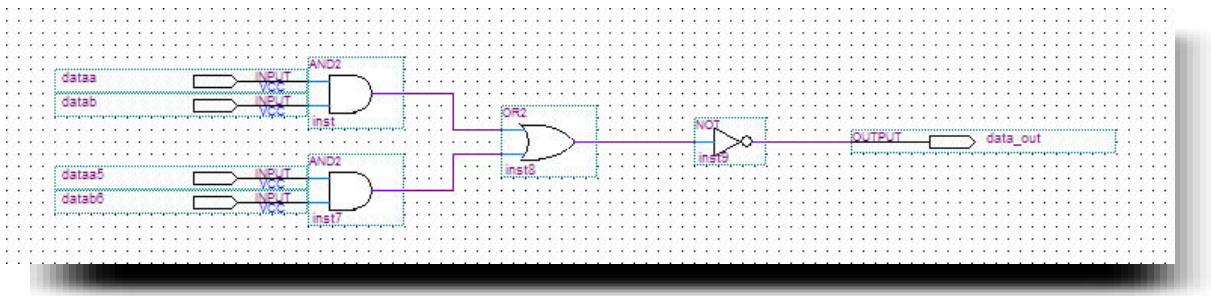
SOPC WORLD 2004

ALTERA.

Application: Design Flaw

Problem:

Design Specification



Verilog Implementation

```
// Verilog code for AND-OR-INVERT gate
module test (my_input1, my_input2, my_input3, my_input4, data_out);
input my_input1, my_input2, my_input3, my_input4;
output data_out;
wire data_out;
wire AB, CD, or_out;

assign AB = my_input1 & my_input2;  //top AND
assign CD = my_input3 & my_input4;  //bottom AND
assign or_out = AB | CD; // or output
assign data_out = or_out; //invert for final result

endmodule
// end of Verilog code
```

Designer did not Invert the Output of the OR Gate !

SOPC WORLD 2004

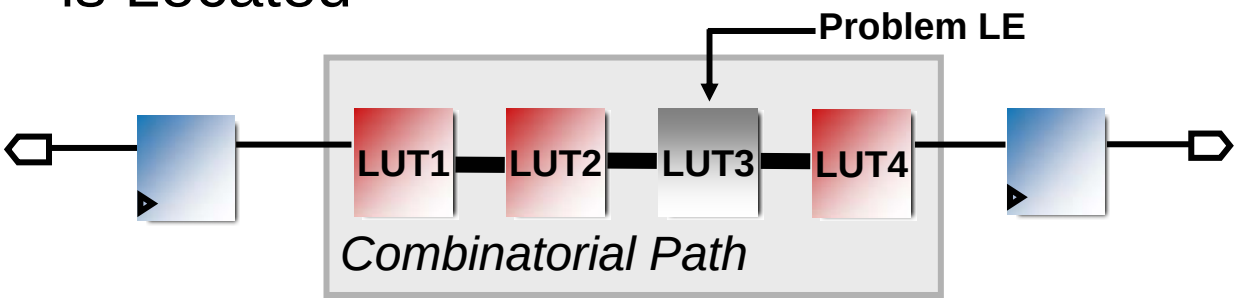
ALTERA.

Design Flaw - Solution

1. Locate the “Problem LE”

Method 1

- Use “Design Knowledge” to Determine Where the LE is Located

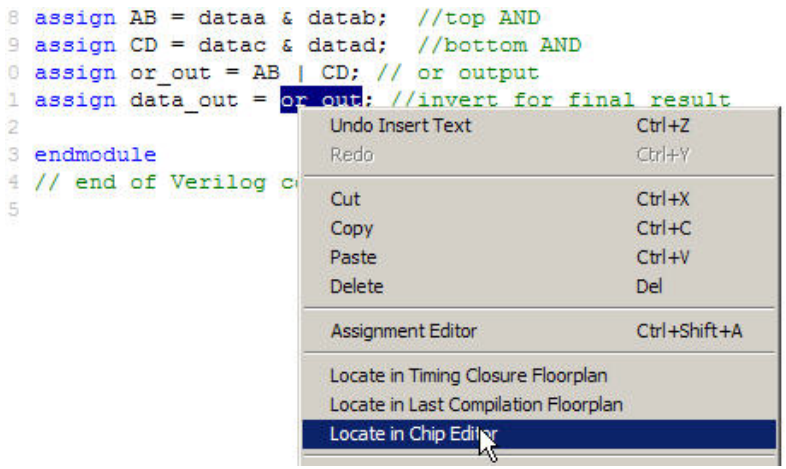


- Begin at Input Pin and Use the **“Go To Destination ATOM”** Option Until the “Problem LE” is Found

Design Flaw – Solution (cont'd)

Method 2

- Highlight the LE in the Source Code and Select **Locate in Chip Editor**



- This Method May Not Work if the Node Name Goes Through a Significant Change During the Synthesis Process

Design Flaw – Solution (cont'd)

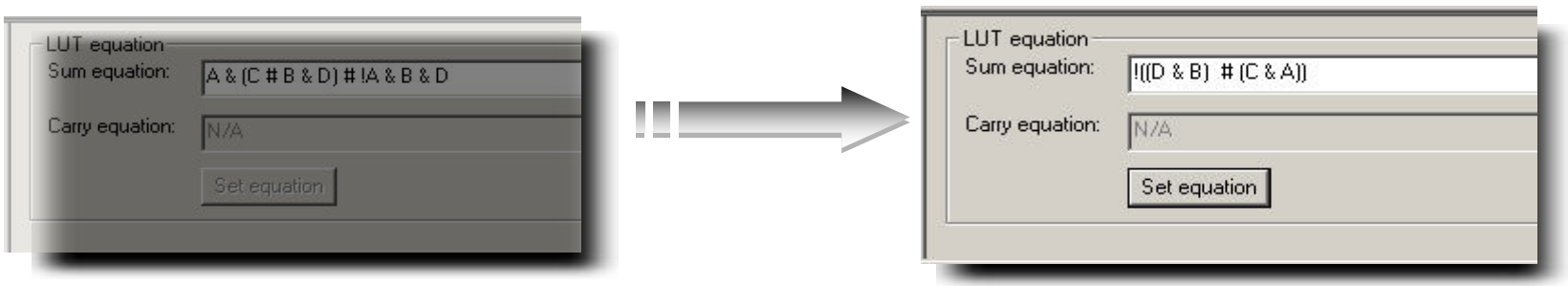
2. Determine the Signals that Drive the Four Inputs to the LE
   - Use the **"Go To Source ATOM"** Option to Determine Connectivity
   - Provides Mapping of RTL Signals to Signals That Are Used in the LUT Equation

Example:

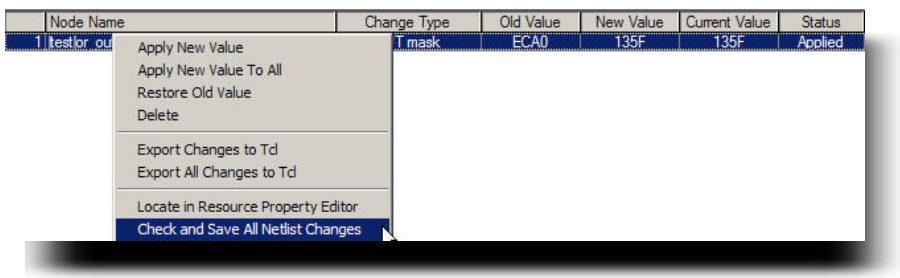
My Signal	LE Input	LUT Equation
my_input1	datad	D
my_input2	datab	B
my_input3	datac	C
my_input4	dataa	A

Design Flaw – Solution (cont'd)

3. Modify the LUT Equation in the LE Editor to Reflect the Changes



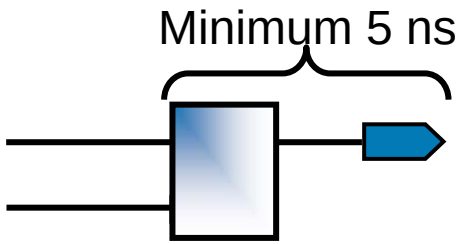
4. Run the Circuit Level DRC



5. Run the Assembler to Generate a POF file

Application: Timing Verification

Problem:



Design Specification Calls for a Minimum TCO

Quartus II Timing Analysis Reports the Following TCO Results:

Minimum tco

	Minimum Slack	Required Min tco	Actual Min tco	Source Name	Destination Name	Source Clock Name
1	-0.077 ns	5.000 ns	4.923 ns	inst3	data_out1	clk

Possible Solution:

The Delay Chain Settings can be Enabled with the I/O Property Editor to Increase the Output Delay to the Pin

Application: Timing Verification

- Determine if Adjusting the Delay Chain Settings Can Help Solve the Problem

Info: Minimum slack time is -77 ps for clock clk between source register inst3 and destination pin data_out1

+ Shortest register to pin delay is 2.445 ns

1: + IC(0.000 ns) + CELL(0.000 ns) = 0.000 ns; Loc. = IOC_X44_Y31_N2; REG Node = 'inst3'

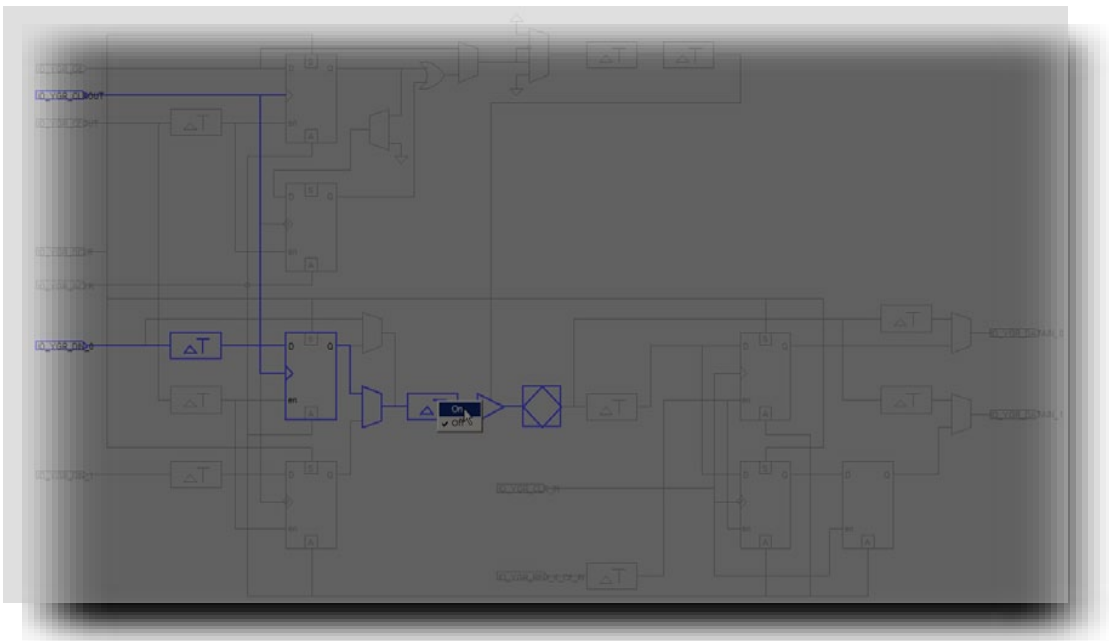
2: + IC(0.000 ns) + CELL(2.445 ns) = 2.445 ns; Loc. = Pin_D7; PIN Node = 'data_out1'

Total cell delay = 2.445 ns

- Solve this Problem in Two Ways
  1. Move the Register Outside the I/O Element
  2. Modify the Delay Chain Settings with the I/O Property Editor

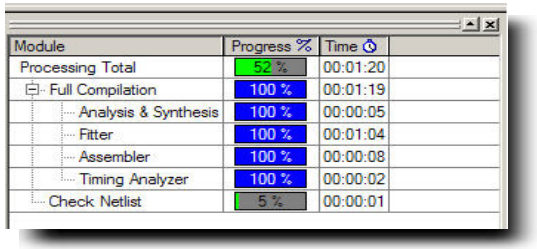
Timing Verification – Solution (cont'd)

- Locate I/O Element in the Chip Editor
  - Launch the Resource Property Editor
- Modify the Existing Delay Chain Setting



Timing Verification – Solution (cont'd)

3. Run a Circuit Level DRC



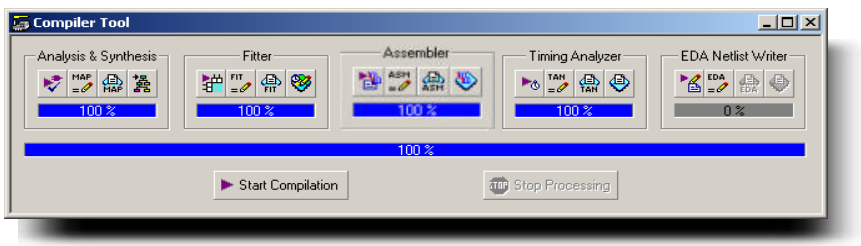
Module	Progress	Time
Processing Total	52 %	00:01:20
Full Compilation	100 %	00:01:19
Analysis & Synthesis	100 %	00:00:05
Fitter	100 %	00:01:04
Assembler	100 %	00:00:08
Timing Analyzer	100 %	00:00:02
Check Netlist	5 %	00:00:01

4. Run Quartus II Timing Analysis to Verify Correct Behavior

Minimum tco

	Minimum Slack	Required Min tco	Actual Min tco	Source Name	Destination Name	Source Clock Name
1	0.300 ns	5.000 ns	5.300 ns	inst3	data_out1	clk

5. Run the Assembler to Generate a POF file



Application – Last Minute ECO's

Problem: ***Design Changes Towards the Back End of the Design Cycle***

Original Design

```
module test( d, clk, clear, load, up_down, qd);
// Port Declaration
input   [7:0] d;
input   clk;
input   clear;
input   load;
input   up_down;
output  [7:0] qd;

reg     [7:0] cnt;
assign qd = cnt;
always @ (posedge clk)
begin
    if (!clear)
        cnt = 8'h00;
    else if (load)
        cnt = d;
    else if (up_down)
        cnt = cnt + 1;
    else
        cnt = cnt - 1;
end
endmodule
```

Clear Signal is Active Low

Possible Solutions to Resolving ECO:

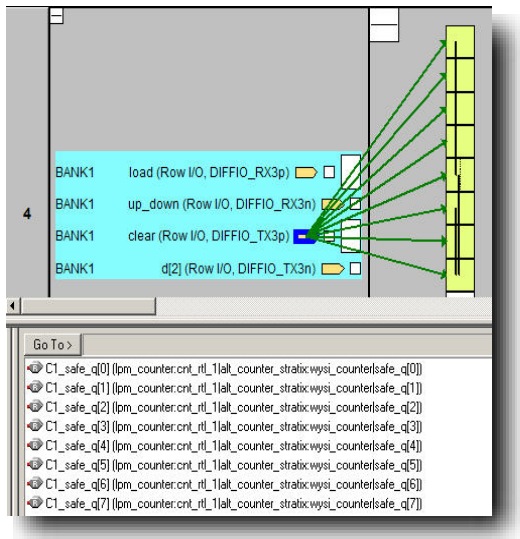
1. Make Changes to Source Code to Resolve ECO
2. Use Chip Editor to Perform the ECO

SOPC WORLD 2004

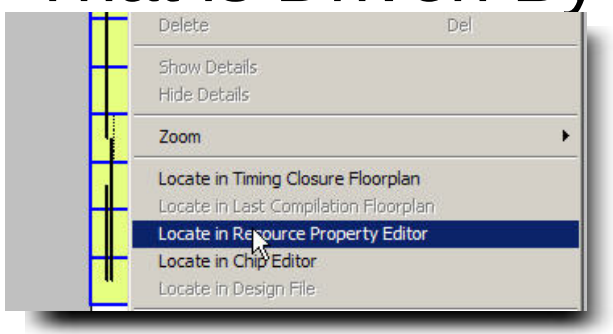
ALTERA.

ECO's - Solution

- Locate the **clear** Signal in the **Floorplan Editor**

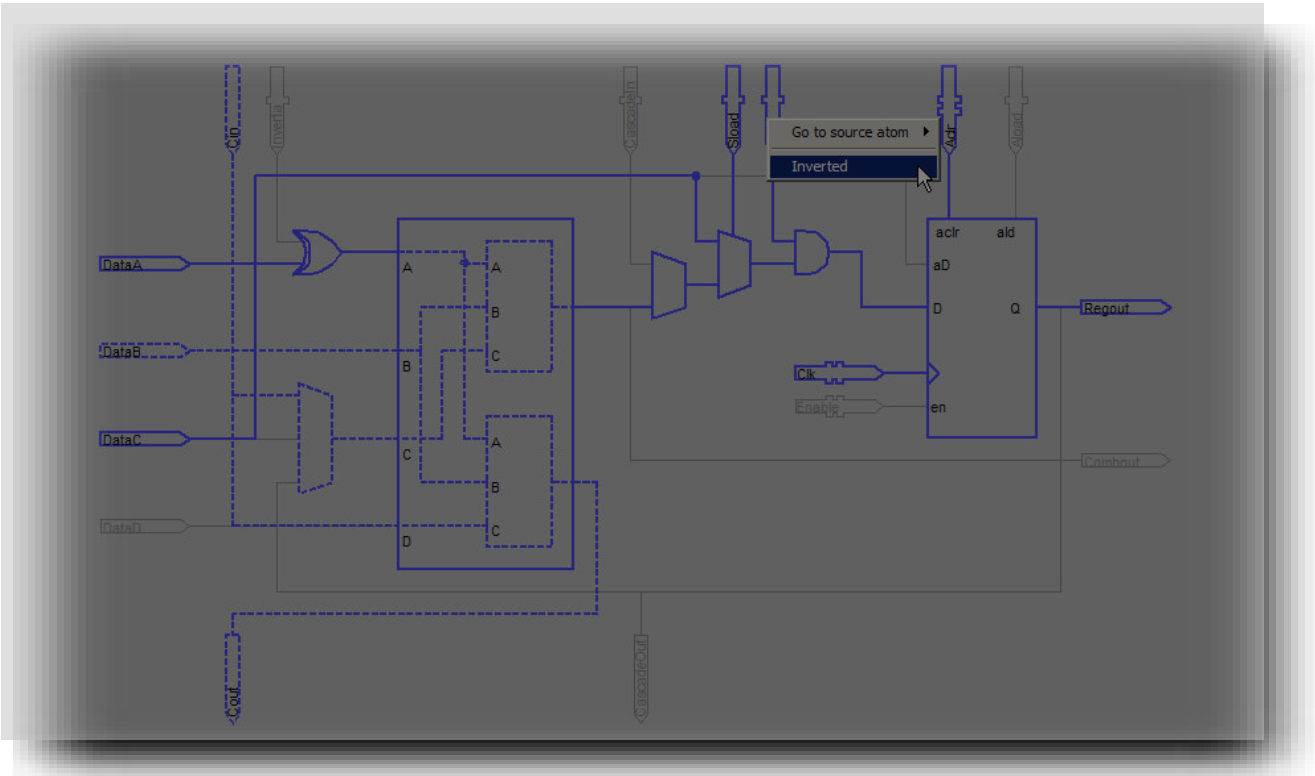


- Open the **Resource Property Editor** for Each of the Signals That is Driven By the **clear** Signal



ECO's – Solution (cont'd)

3. Invert the Value of the **synchronous clear** That Feeds the Register



POP Quiz

- ***Which resource can't modify in Chip Editor?***

1). Logic Element

2). I/O Element

3). M4K Memory Block

4). PLL

Summary

- Use the Chip Editor to Perform ***Minor*** Changes to your Design
  - May be Difficult to Use for New Customers
- Use this Feature to Modify Properties of ATOMS
  - A Logic Element
  - A PLL
  - An I/O
- Use This Feature to Analyze Your Critical Path
- Use This Feature Learn About the Target Architecture