

Introduction

As FPGA designs grow larger and larger in density, the need to analyze the design for performance, routing congestion, logic placement, and the ability to execute Engineering Change Orders (ECOs) becomes critical.

Beginning with the Quartus II software version 6.1, you can use Chip Planner (Floorplan and Chip Editor) as a powerful tool to perform design analysis, create a design floorplan, and implement ECOs in your design.

For ECOs and design analysis, Chip Planner works directly on the post place-and-route database of your design so you can implement design changes in minutes without performing a full compilation. The changes you make are restricted to particular device resources, so the timing performance of the remaining portions of your design are not affected. Design rule checks are performed on all changes to prevent illegal modifications to your design.

For viewing and creating assignments for a design floorplan in supported devices, Chip Planner can be used to perform all the functions you would perform with the Timing Closure Floorplan in the Quartus II software.

Table 15–1 lists the device families supported by Chip Planner (Floorplan and Chip Editor), Timing Closure Floorplan, or both.



For device families that support both the Timing Closure Floorplan and the Chip Planner, Altera recommends using Chip Planner for design analysis.

<i>Table 15–1. Chip Planner Device Support (Part 1 of 2)</i>		
Device Family	Timing Closure Floorplan	Chip Planner (Floorplan and Chip Editor)
Stratix® III	—	✓
Stratix II	✓	✓
Stratix II GX	✓	✓
Stratix GX	✓	✓
Stratix	✓	✓
HardCopy® II	—	✓
Cyclone® III	✓	✓

Table 15–1. Chip Planner Device Support (Part 2 of 2)

Device Family	Timing Closure Floorplan	Chip Planner (Floorplan and Chip Editor)
Cyclone II	✓	✓
Cyclone	✓	✓
MAX® II	✓	✓
MAX 7000	✓	—
ACEX®	✓	—
FLEX 10K® FLEX® 10KA FLEX 10KE FLEX 6000	✓	—
APEX™ II APEX 20KC APEX 20KE	✓	—

This chapter describes how to use Chip Planner for design analysis and implementing ECOs in your design. This chapter covers the following topics:

- [Chip Planner Overview](#)
- [Design Analysis Using Chip Planner](#)
- [Resource Property Editor](#)
- [Change Manager](#)
- [Performing ECOs with Chip Planner](#)
- [Common ECO Applications](#)
- [Post ECO Steps](#)

Chip Planner Overview

Chip Planner provides a visual display of chip resources. It can show logic placement, LogicLock and custom regions, relative resource usage, detailed routing information, fan-ins and fan-outs, paths between registers, and delay estimates for paths. With Chip Planner, you can view critical path information, physical timing estimates, and routing congestion.

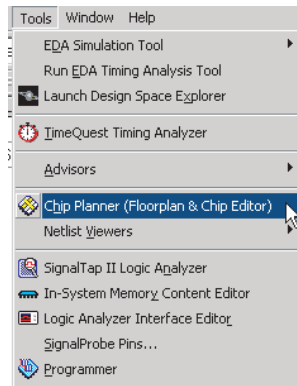
Chip Planner can perform assignment changes, such as creating and deleting resource assignments, as well as post-compilation changes, such as creating, moving, and deleting logic cells and I/O atoms. By using Chip Planner in conjunction with the Resource Property Editor, you can change connections between resources and make post-compilation changes to the properties of logic cells, I/O elements, and PLLs. With

Chip Planner, you can now view and create assignments for a design floorplan, perform power and design analyses, and implement ECOs all in a single tool.


Invoking Chip Planner

To invoke Chip Planner, on the Tools menu, click **Chip Planner (Floorplan & Chip Editor)**. See [Figure 15-1](#).

Figure 15-1. Invoking the Chip Planner in the Quartus II User Interface



Other methods to invoke Chip Planner include:

- Click the Chip Planner icon  on the Quartus II software toolbar
- Use the right-click menu from the Compilation Report
- Use the right-click menu from the Project Navigator window
- Use the right-click menu from the RTL source code
- Use the right-click menu from the Timing Closure Floorplan
- Use the right-click menu from the Node Finder
- Use the right-click menu from the Simulation Report
- Use the right-click menu from the RTL Viewer

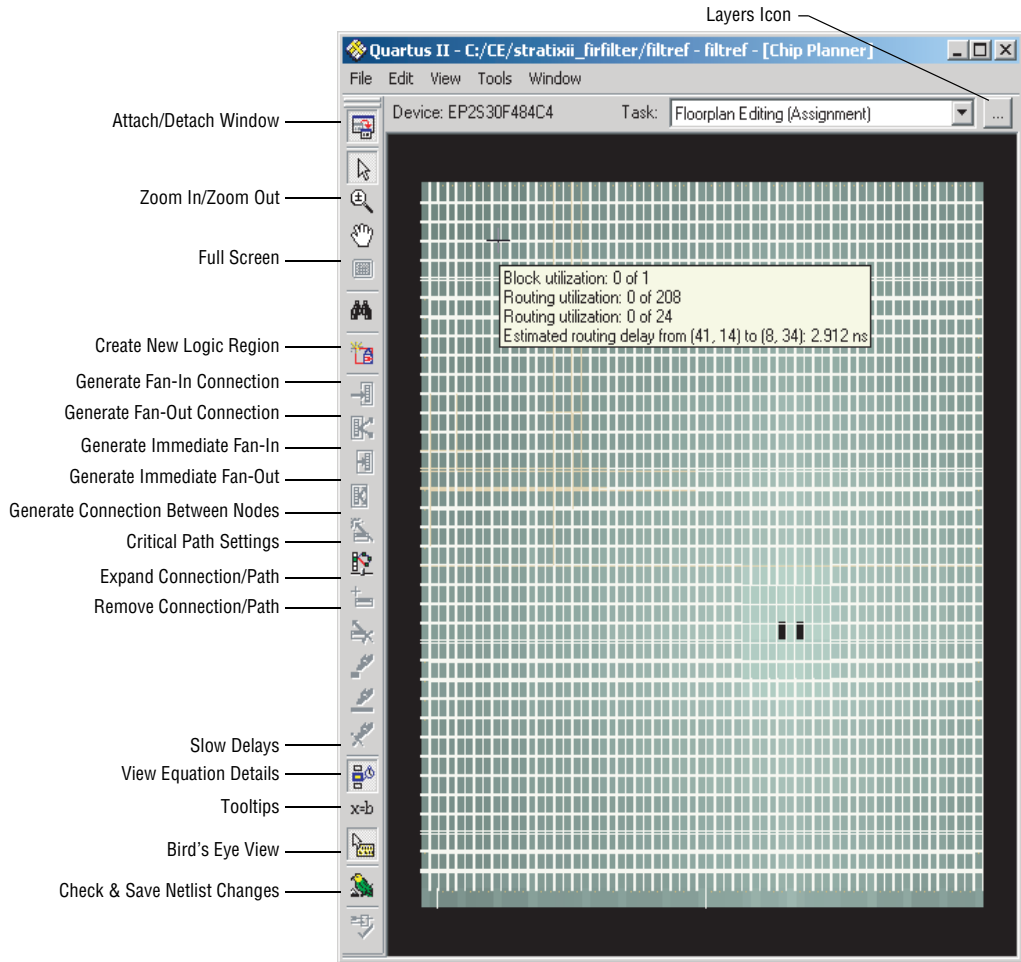


If the device in your project is not supported by Chip Planner and you attempt to invoke Chip Planner, the following message appears: “Can’t display Chip Planner: the current device family is unsupported.” In this case, use the Timing Closure Floorplan for these devices.

Chip Planner Toolbar

Chip Planner gives you powerful capabilities for design analyses with a very user friendly Graphical User Interface. Many functions within the Chip Planner can be selected or performed either from the menu items or clicking the icons on the toolbar. [Figure 15-2](#) shows an example of the Chip Planner Toolbar and provides descriptions for commonly used icons available from the Chip Planner toolbar.

Figure 15–2. Chip Planner Toolbar



You can also customize the icons present on the Chip Planner's toolbar by either clicking **Customize** Chip Planner on the Tools menu (if the Chip Planner window is attached) or by clicking **Customize** on the Tools menu (if the Chip Planner window is detached).

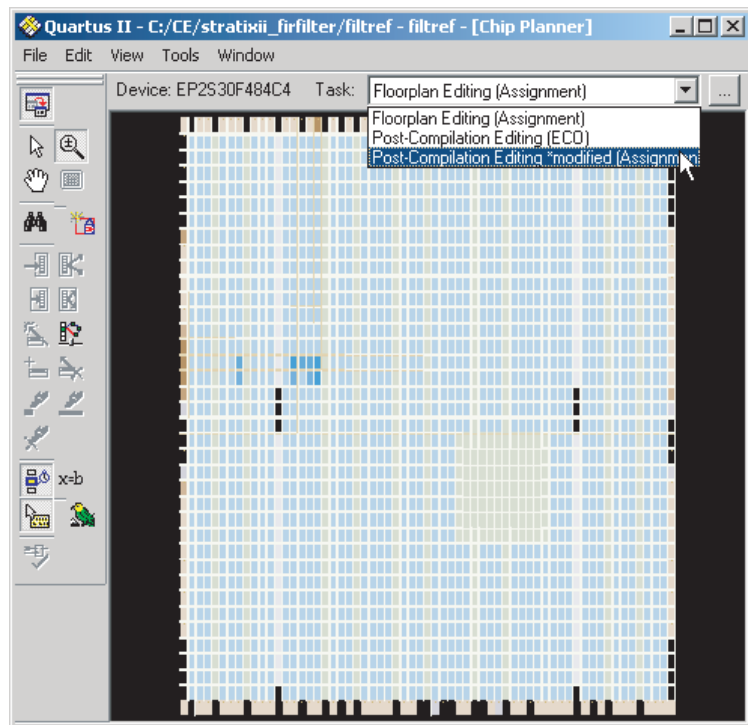
Chip Planner Tasks and Layers

Chip Planner has predefined tasks that enable you to quickly implement ECO changes or manipulate assignments for the floorplan of the device. The following are the predefined tasks in Chip Planner:

- Post-Compilation Editing (ECO)
- Floorplan Editing (Assignment)
- Power Analysis (Assignment) — available for only Stratix III devices

Figure 15–3 shows the tasks available in the Floorplanning tool.

Figure 15–3. Tasks in Chip Planner

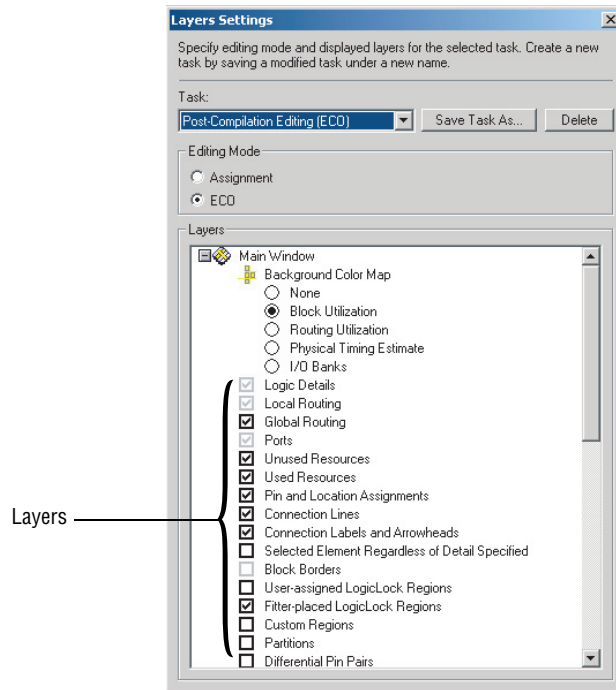


In Chip Planner, layers allow you to specify the graphic elements that are displayed for a given task. You can turn off the display of specific graphic elements, which increases the window refresh speed and reduces visual clutter when viewing complex designs. The Background Color Map indicates the relative level of resource usage for different areas of the

device. For example, Routing Utilization indicates the relative routing utilization and Physical Timing Estimate indicates the relative physical timing.

Each predefined task in Chip Planner has a Background Color Map, a set of displayed layers, and an editing mode associated with it. Click the **Layers** icon (shown in Figure 15-4) to see which layers are selected and which background color maps are used for a particular task.

Figure 15-4. Layers in Chip Planner



Chip Planner operates in either **Assignment** or **ECO** mode. You can use either of these modes to perform design analyses. You can use the Floorplan Editor (Assignment) task with Assignments mode to manipulate LogicLock™ regions and location assignments in your design. The Post Compilation Changes (ECO) task with ECO editing mode is used to implement ECO changes in your design. The Power Analysis (Assignment) task is used to view high and low power resources in a Stratix III device.

You can also create and save your own custom tasks. When creating a custom task, you can turn on or off any layer in your task. Layers can be turned on or off by checking the appropriate box located next to each layer. You can also select different Background Color Maps to be used for your custom task.

Chip Planner Features

In addition to predefined tasks, the Chip Planner integrated toolset includes Resource Property Editor and Change Manager. Collectively they provide the following features:

- **Chip Planner Tasks**—Lets you view post-compilation placement, connections, and routing paths; create LogicLock regions and location assignments, create new logic cells and/or I/O atoms, and move existing logic cells and I/O atoms while having full view of your design.
- **Resource Property Editor**—Lets you make changes to the properties and parameters of device resources, and modify connectivity between certain types of device resources.
- **Change Manager**—Maintains an ongoing record of your post-compilation changes, and ensures that conflicting changes do not occur. Additionally, the Change Manager lets you write out a tool command language (.tcl) file that allows you to reproduce the changes you made.

Design Analysis Using Chip Planner

Chip Planner assists you in visually analyzing your design at any stage of your design cycle. The following sections discuss design analysis procedures and views of Chip Planner, which can be used with any predefined task of Chip Planner (unless explicitly stated that a certain procedure requires a particular task or editing mode).

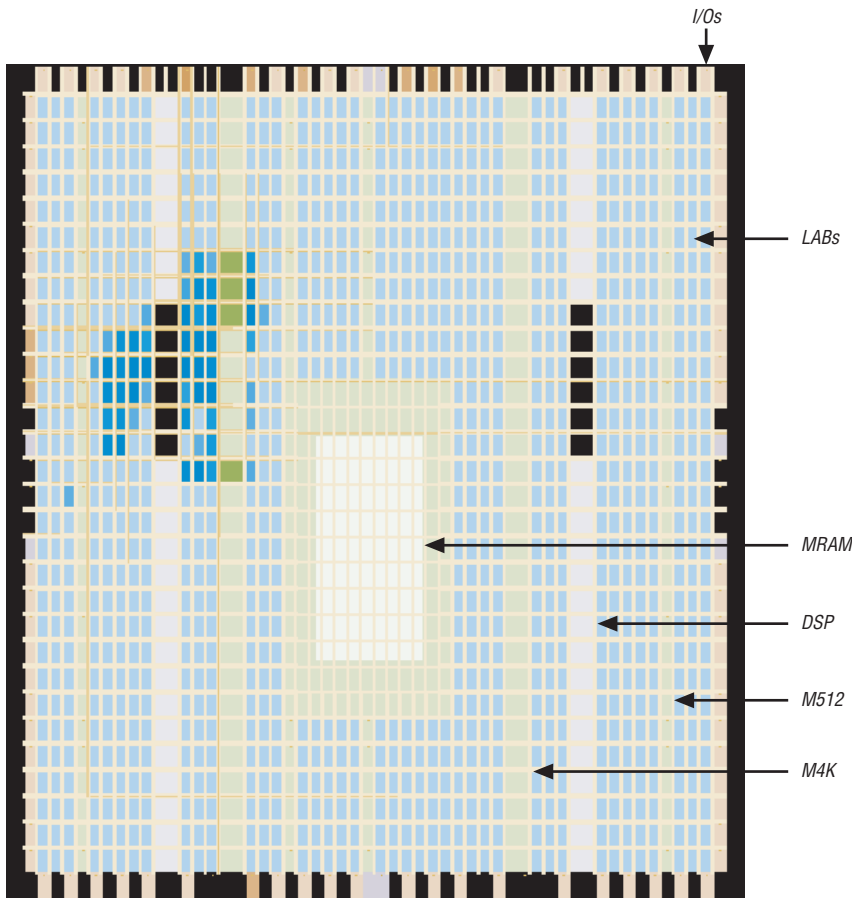
Chip Planner Floorplan Views

The Chip Planner uses a hierarchical zoom viewer that shows various abstraction levels of the targeted Altera device. As you increase the zoom level, the level of abstraction decreases, thus revealing more detail about your design.

First (Highest) Level View

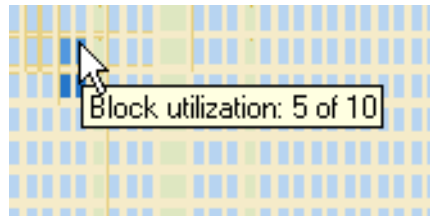
The first (highest) zoom level provides a high-level view of the entire device floorplan. This view provides a level of detail similar to the Field View in the Quartus II Timing Closure floorplan. You can locate and view the placement of any node in your design. [Figure 15–5](#) shows the Chip Planner's Floorplan first level view of a Stratix device.

Figure 15–5. Chip Planner’s First (Highest) Level Floorplan View

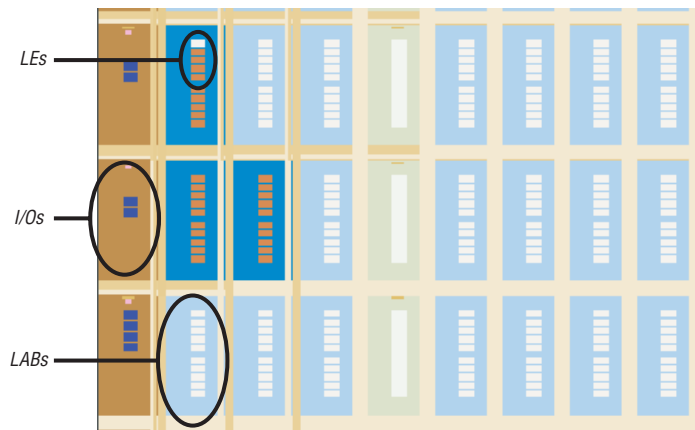


Each resource is shown in a different color, which makes it easier to distinguish between resources. The Chip Planner floorplan uses a gradient color scheme in which the color becomes darker as the utilization of a resource increases. For example, as more LEs are used in the LAB, the color of the LAB becomes darker.

When you place the mouse pointer over a resource at this level, a tooltip appears that describes, at a high level, the utilization of the resource (Figure 15–6).

Figure 15–6. Tooltip Message: First Level View**Second Level View**

As you zoom in, you see an increase in the level of detail. [Figure 15–7](#) shows the second-level view of the Chip Planner Floorplan for a Stratix device.

Figure 15–7. Chip Planner's Second Level Floorplan View

At this level you can see the contents of LABs and I/O banks. You also see the routing channels that are used to connect resources.

When you place the mouse pointer over an LE or ALM at this level, a tooltip is displayed ([Figure 15–8](#)) that shows the name of the LE/ALM, the location of the LE/ALM, and the number of resources that are used with that LAB. When you place the mouse pointer over an interconnect, the tooltip shows the routing channels that are used by that interconnect.

Figure 15–8. Tooltip Message: Second Level View

Third Level View

At the third and lowest level, you can see each routing resource that is used within a LAB in the FPGA. [Figure 15–9](#) shows the level of detail at the third level view for a Stratix device.

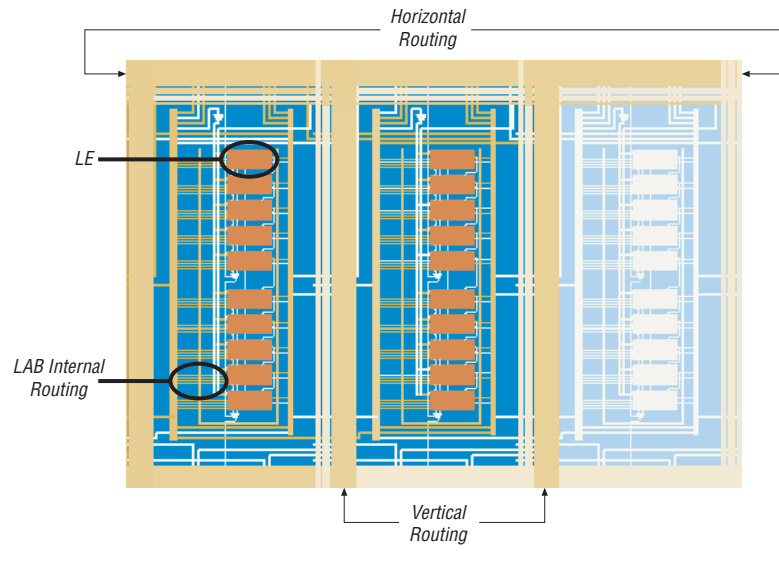
From the third level, you can move LEs, ALMs, and I/Os from one physical location to another. You can move a resource by selecting, dragging, and dropping it into the desired location. At this level you can also create new LEs and I/Os. To create a resource, right-click at the location you want to create the resource and click **Create Atom**. To delete a resource, right-click on the resources you want to delete and click **Delete Atom**.



You can only delete a resource after all of its fan-out connections are removed. Moving nodes in the Floorplan Editing (Assignment) task creates an assignment. However, if you move logic nodes in the Post-Compilation Editing (ECO) task then that change is considered an ECO change.



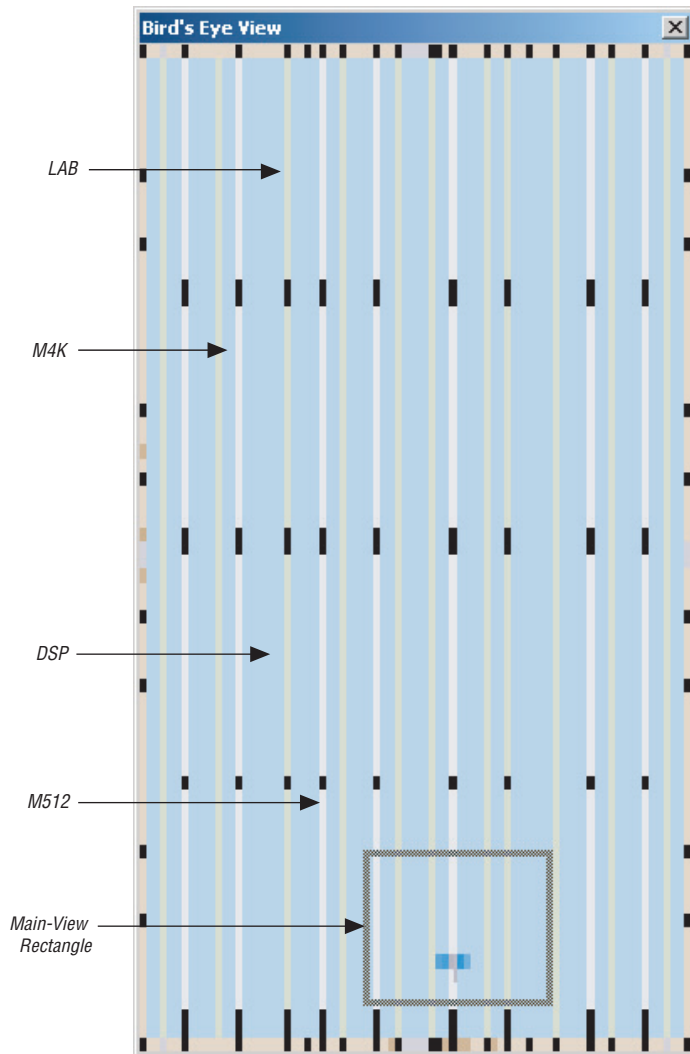
For more information on Floorplan Assignments, refer to [“Viewing Assignments in Chip Planner” on page 15–27](#). For more information on performing ECOs, refer to [“Performing ECOs with Chip Planner” on page 15–53](#).

Figure 15–9. Chip Planner's Third Level Floorplan View

Bird's Eye View

The Bird's Eye View (Figure 15–10) displays a high-level picture of resource usage for the entire chip and provides a fast and efficient way to navigate between areas of interest in the Chip Planner.

Figure 15–10. Bird's Eye View



The Bird's Eye View is displayed as a separate window that is linked to the Chip Planner floorplan. When you select an area of interest in the Bird's Eye View, the Chip Planner floorplan automatically refreshes to show that region of the device. As you change the size of the main-view rectangle in the Bird's Eye View window, the main Chip Planner

floorplan window also zooms in (or zooms out). You can make the main-view rectangle smaller in the Bird's Eye View to see more detail on the Chip Planner floorplan window.

The Bird's Eye View is particularly useful when the parts of your design that you are interested in are at opposite ends of the chip and you want to quickly navigate between resource elements without losing your frame of reference.

Viewing Architecture-Specific Design Information

With Chip Planner, you can view the following architecture-specific information related to your design:

- Device routing resources used by your design—Visually examine how blocks are connected, as well as the signal routing that connects the blocks.
- LE configuration—View how a logic element (LE) is configured within your design. For example, you can view which LE inputs are used, if the LE utilizes the register, the look-up table (LUT), or both, as well as the signal flow through the LE.
- ALM configuration—View how an adaptive logic module (ALM) is configured within your design. For example, you can view which ALM inputs are used, if the ALM utilizes the registers, the upper LUT, the lower LUT, or all of them. You can also view the signal flow through the ALM.
- I/O configuration—View how the device I/O resources are used. For example, you can view which components of the I/O resources are used, if the delay chain settings are enabled, which I/O standards are set, and the signal flow through the I/O.
- PLL configuration—View how a phase-locked loop (PLL) is configured within your design. For example, you can view which control signals of the PLL are used along with the settings for your PLL.
- Timing—View the delay between the inputs and outputs of FPGA elements. For example, you can analyze the timing of DATAB input to the COMBOUT output.

Modifying Device Properties with Chip Planner

You can modify the following properties of an Altera device with Chip Planner:

- LEs and ALMs
- I/O cells
- PLLs
- Connections between elements

■ Placement of elements

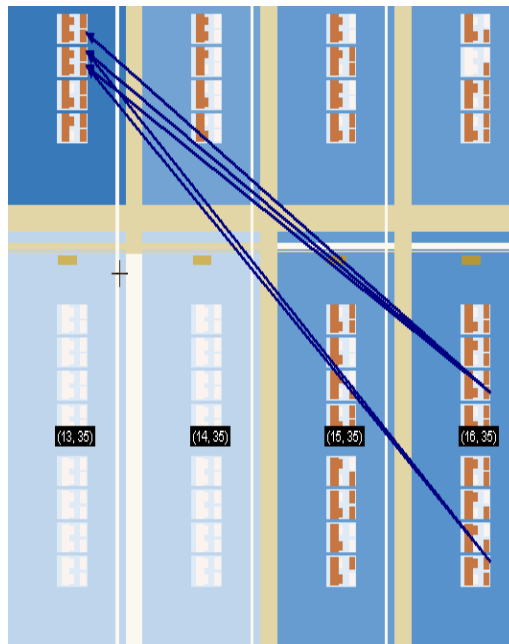


For more information on LEs, ALMs and other resources of the FPGA device, refer to the Device handbook of the FPGA you are using, or refer to “Resource Property Editor” on page 15–32.

Viewing Critical Paths

Critical paths are timing paths in your design that have a negative slack. These timing paths can span from device I/Os to internal registers, registers to registers or registers to devices I/Os. The View Critical Paths feature displays routing paths in the Chip Planner, as shown in Figure 15–11. The criticality of a path is determined by its slack and is shown in the timing analysis report. To view critical paths in the Chip Planner, on the View menu, click **Critical Path Settings**. In the Critical Path Settings dialog box, click **Show Path** (see Figure 15–12 on page 15–17).

Figure 15–11. Chip Planner with Critical Path Enabled



When viewing critical paths, you can specify the clock in the design you want to view. You determine the paths to be displayed by specifying the slack threshold in the slack field.



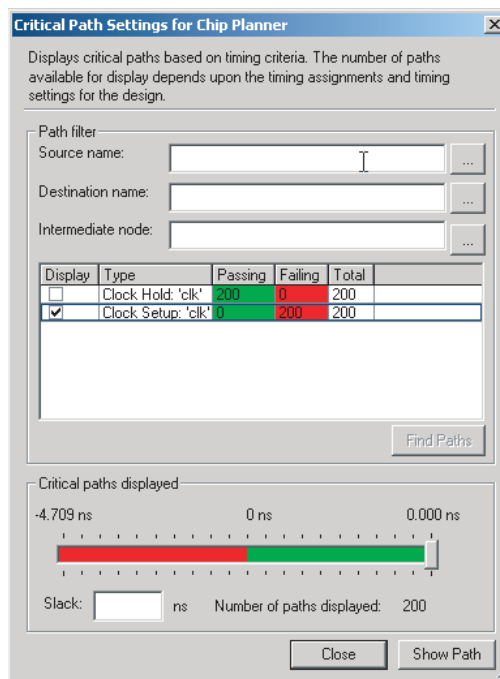
Timing settings must be made and a timing analysis performed for paths to be displayed in the floorplan.



For more information about performing static timing analysis with the classic timing analyzer, refer to the *Classic Timing Analyzer* chapter in volume 3 of the *Quartus II Handbook*.

For more information on performing static timing analysis with the TimeQuest timing analyzer, refer to the *TimeQuest Timing Analyzer* chapter in volume 3 of the *Quartus II Handbook*.

Figure 15–12. Critical Path Settings for the Chip Planner

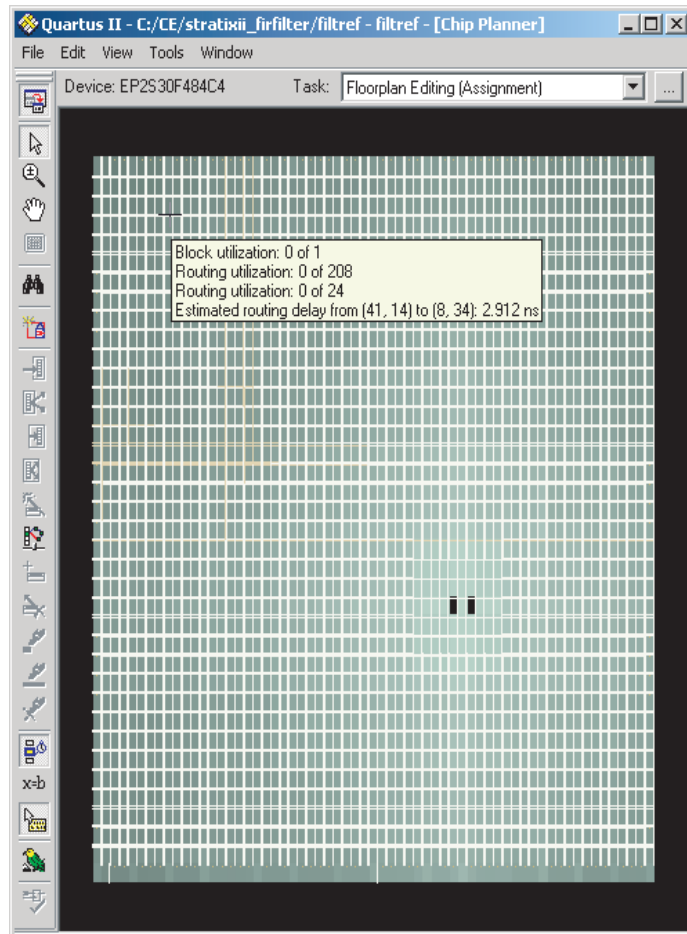


Viewing Physical Timing Estimates

In Chip Planner, you can select a resource and see the approximate delay to any other resource on the device. After you select a resource, the delay is represented by the color of potential destination resources. The lighter the color of the resource, the longer the delay.

To see the physical timing map of the device, in the Chip Planner, click the "Layers" icon located next to the **Task** drop down menu. Under the Background Color Map, select the **Physical Timing Estimate** radio button. Select a source and move your cursor to a destination resource. The Chip Planner displays the approximate routing delay between your selected source and destination register (Figure 15-13).

Figure 15-13. Viewing Physical Timing Estimates in the Chip Planner





You can use the physical timing estimate information when manually attempting to improve the Fitter results by moving logic in a device or when creating LogicLock regions to group logic together. This feature allows you to estimate the physical routing delay between different nodes so that you can place critical nodes and modules closer together, and move non-critical or unrelated nodes and modules further apart. As well as reducing delay between critical nodes, you can make placement assignments to reduce the routing congestion between critical and noncritical entities and modules. This allows the Quartus II Fitter to meet the design timing requirements. Note that moving logic and creating manual placements is an advanced technique to meet timing requirements and must be done after careful analysis of the design. Moving nodes in the Floorplan Editing (Assignment) task creates an assignment. However, if you move logic nodes in the Post-Compilation Editing (ECO) task then that change is considered an ECO change.

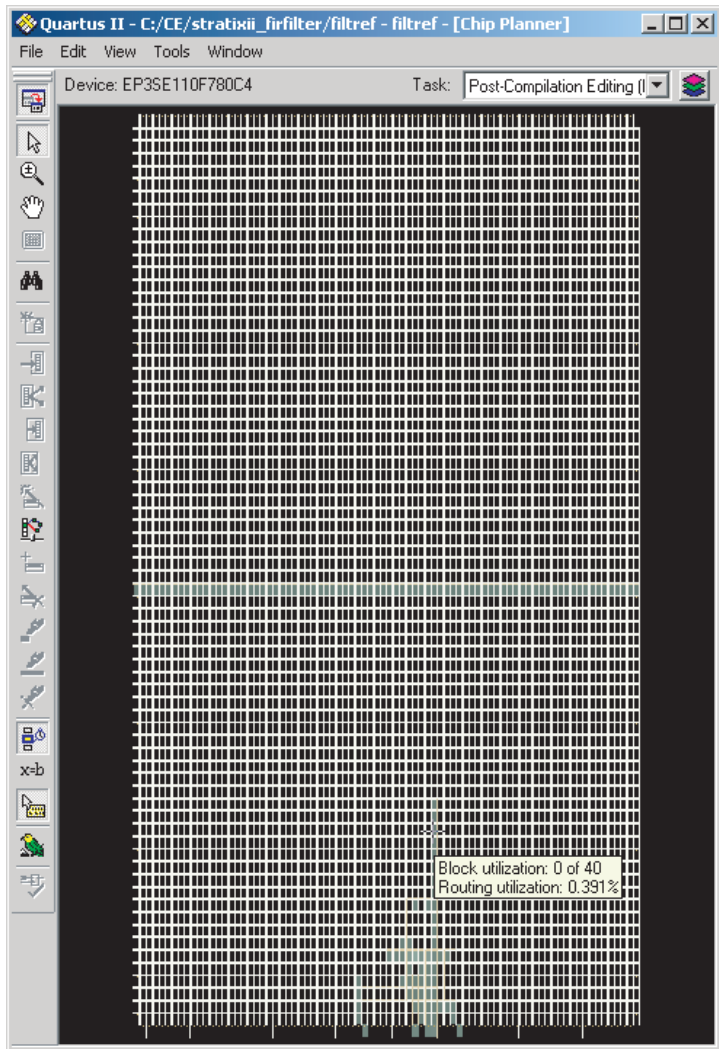


For more information on Floorplan Assignments, refer to [“Viewing Assignments in Chip Planner” on page 15–27](#). For more information on performing ECOs, refer to [“Performing ECOs with Chip Planner” on page 15–53](#).

Viewing Routing Congestion

The Routing Congestion view allows you to determine the percentage of routing resources used after a compilation. This feature identifies where there is a lack of routing resources. The congestion is visually represented by the color and shading of logic resources. The darker shading represents a greater routing resource utilization. To view the routing congestion in the Chip Planner, click the “Layers” icon located next to the **Task** menu. Under **Background Color Map**, select the **Routing Utilization** map ([Figure 15–14](#)).

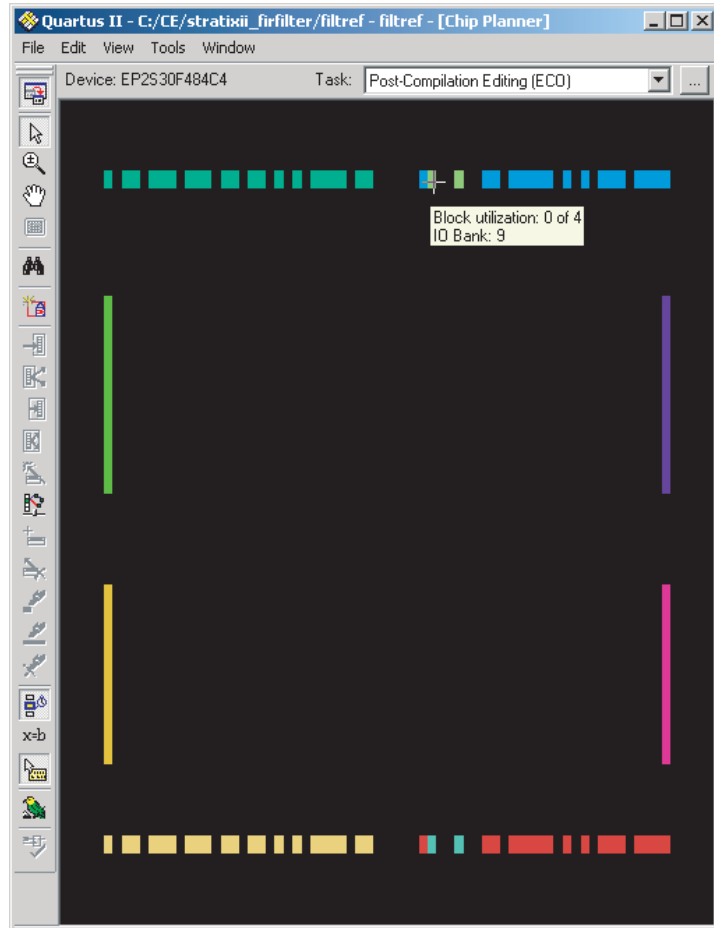
Figure 15–14. Viewing Routing Congestion Map in the Chip Planner



Viewing I/O Banks

Chip Planner can show all the I/O banks of the device displayed in different colors. To see the I/O bank map of the device, click the "Layers" icon located next to the **Task** menu. Under **Background Color Map**, select the **I/O Banks** map. See [Figure 15–15](#).

Figure 15–15. Viewing I/O Banks in the Chip Planner

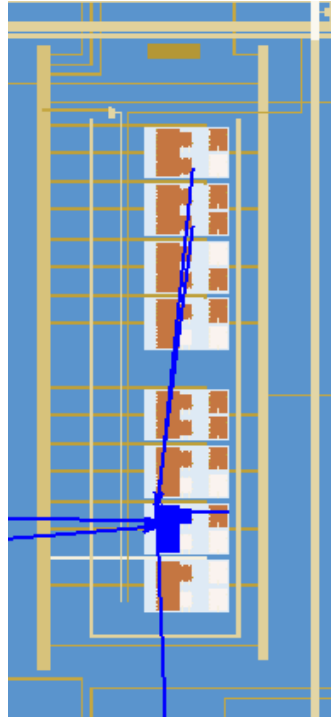


All predefined tasks in Chip Planner show the Block Utilization Map as the default Background Color Map by default.

Generating Fan-In & Fan-Out Connections

This feature enables you to view the atoms that fan-in to or fan-out from the selected atom. To remove the connections displayed, use the **Clear Connections** icon in the Chip Planner toolbar. [Figure 15–16](#) shows the fan-in connections for the selected resource.

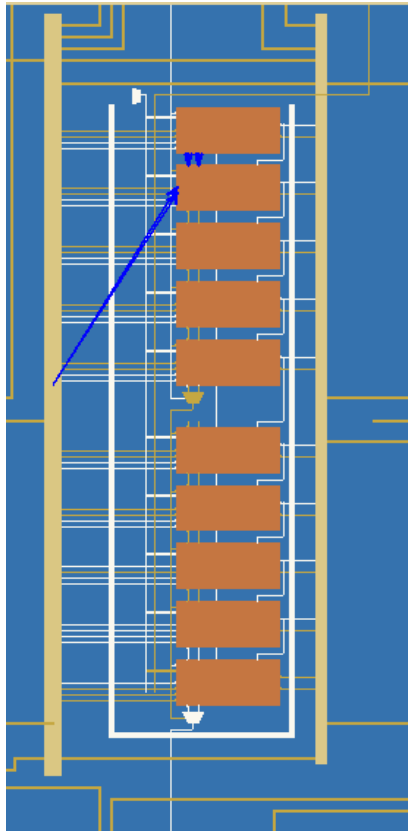
Figure 15–16. Generated Fan-In



Generating Immediate Fan-In & Fan-Out Connections

This feature enables you to view the immediate resource that is the fan-in or fan-out connection for the selected atom. For example, selecting a logic resource and choosing to view the immediate fan-in enables you to see the routing resource that drives the logic resource. You can generate immediate fan-in and fan-outs for all logic resources and routing resources. To remove the connections that are displayed, use the **Clear Connections** icon in the toolbar. [Figure 15–17](#) shows the immediate fan-out connections for the selected resource.

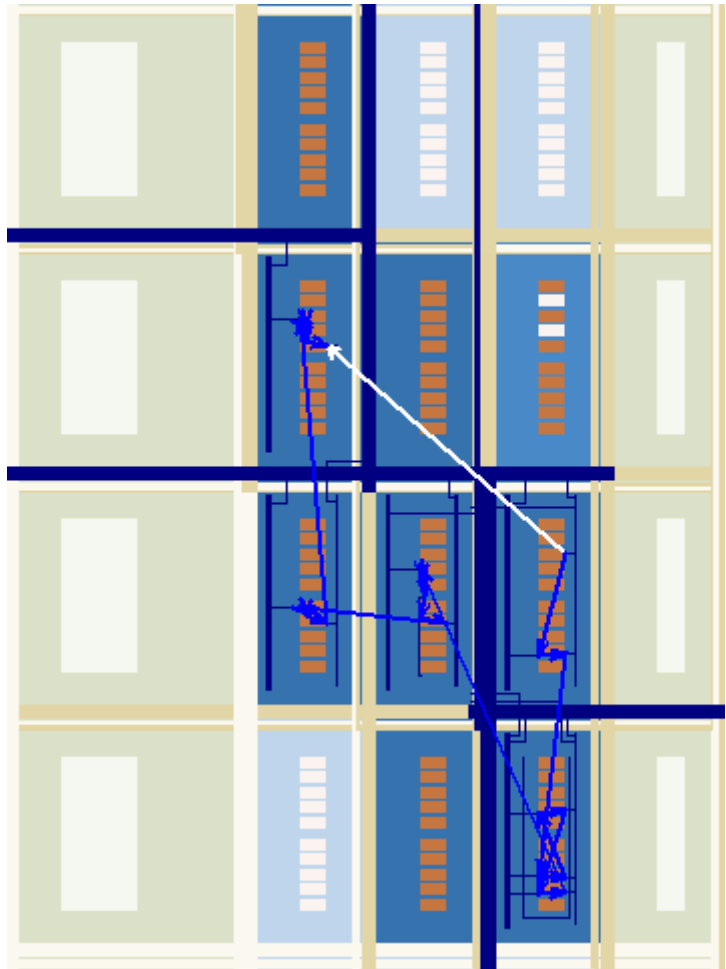
Figure 15–17. Immediate Fan-Out Connection



Highlight Routing

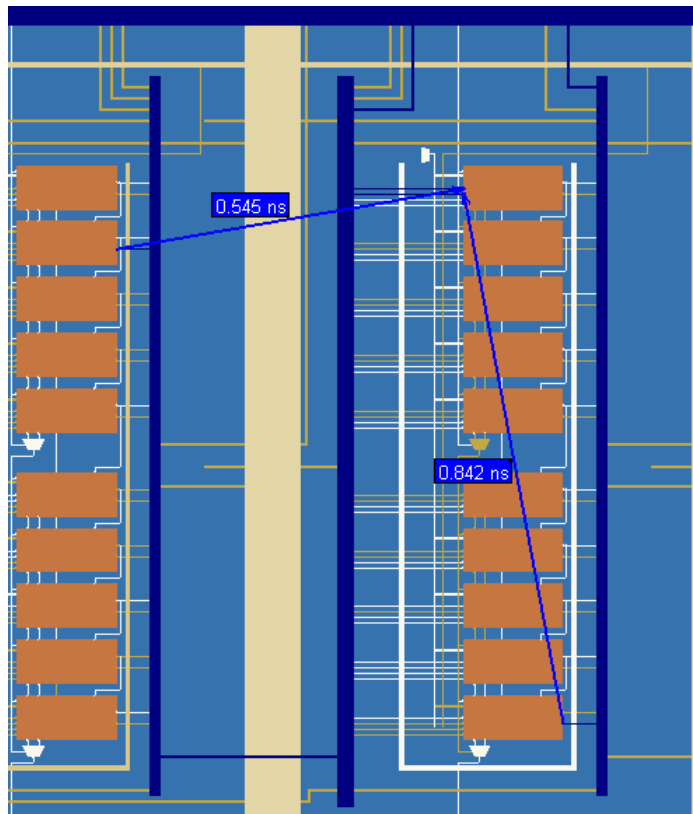
This feature enables you to highlight the routing resources used for a selected path or connection. [Figure 15–18](#) shows the routing resources used between two logic elements.

Figure 15–18. Highlight Routing



Show Delays

You can view the timing delays for the highlighted connections when generating connections between elements. For example, you can view the delay between two logic resources or between a logic resource and a routing resource. [Figure 15–19](#) shows the delays between a number of logic elements.

Figure 15–19. Show Delays

Exploring Paths in the Chip Planner

You can use the Chip Planner to explore paths between logic elements. The following example uses the Chip Planner to traverse paths from the Timing Analysis report.

Locate Path from the Timing Analysis Report to the Chip Planner

To locate a path from the Timing Analysis Report to the Chip Planner, perform the following steps:

1. Select the path you want to locate.

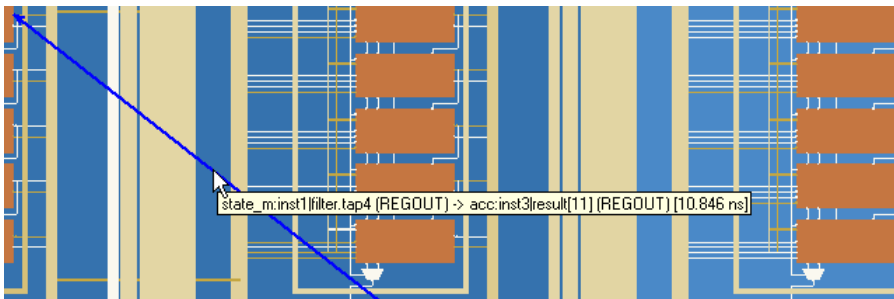
- Right-click on the path in the Timing Analysis Report, right-click **Locate**, then click **Locate in Chip Planner (Floorplan & Chip Editor)** (Figure 15–20).

Figure 15–20. Locate a Timing Path from the Chip Planner Timing Analysis Report

Clock Setup: 'clk'						
	Slack	Actual fmax (period)	From	To	From Clock	To Clock
37	0.085 ns	208.46 MHz (period = 4.915 ns)	state_m_inst1 filter.tap2	acc.inst3 result[4]	clk	clk
38			aps:instkx_1[0]	acc.inst3 result[10]	clk	clk
39			aps:instkx_1[1]	acc.inst3 result[11]	clk	clk
40			aps:instkx_1[0]	acc.inst3 result[9]	clk	clk
41			aps:instkx_2[0]	acc.inst3 result[7]	clk	clk
42			aps:instkx[1]	acc.inst3 result[11]	clk	clk
43			aps:instkx_1[1]	acc.inst3 result[10]	clk	clk
44			state_m_inst1 filter.tap2	acc.inst3 result[3]	clk	clk
45						
46						
47						
48						
49	0.195 ns	208.12 MHz (period = 4.805 ns)				
50	0.199 ns	208.29 MHz (period = 4.801 ns)				
51	0.222 ns	209.29 MHz (period = 4.778 ns)				
52	0.231 ns	209.69 MHz (period = 4.769 ns)				

Figure 15–21 shows the path that is displayed in the Chip Planner.

Figure 15–21. Resulting Path

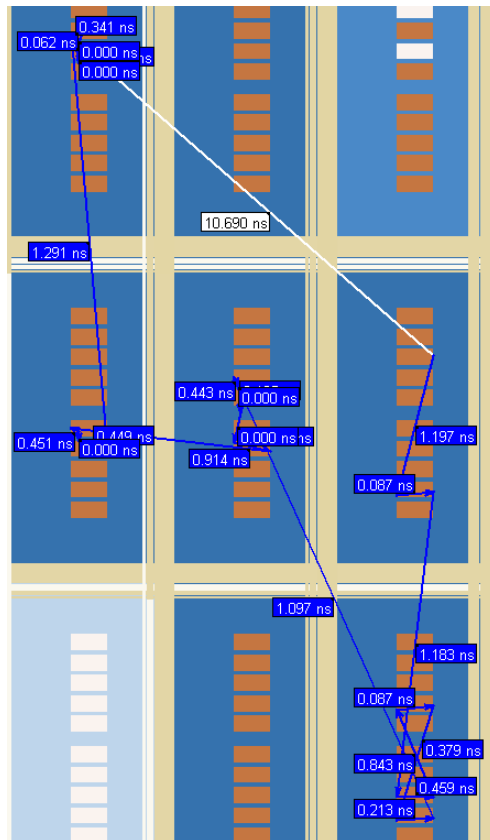


Analyzing Connections for a Path

To determine the connections between items in the Chip Planner, use the **Expand Connections/Paths** icon on the toolbar. To add the timing delays between each connection, use the **Show Delays** icon on the toolbar.

Figure 15–22 shows the connections for the selected path that are displayed in the Chip Planner.

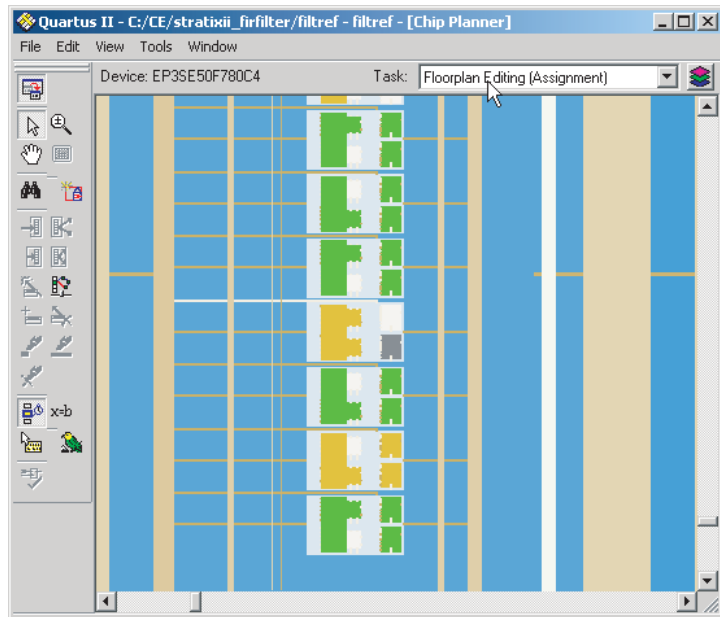
Figure 15–22. Path Analysis



Viewing Assignments in Chip Planner

Location Assignments can be viewed by selecting the appropriate layer set from the tool. To view Location Assignments in Chip Planner, select the task **Floorplan Editing (Assignment)** or any custom task with “Assignment” editing mode. See [Figure 15–23](#).

Chip Planner shows location assignments graphically, by displaying assigned resources in a particular color (gray, by default). You can create or move an assignment by dragging the selected resource to a new location.

Figure 15–23. Viewing Assignments in Chip Planner

You can make node and pin location assignments, and assignments to LogicLock regions and custom regions, using the drag and drop method in Chip Planner. The assignments that you create are applied by the Fitter during the next place and route operation.



To learn more about working with location assignments, refer to the Quartus II Help.

Connection Between LogicLock Regions

You can view and edit LogicLock regions using Chip Planner. Select the task **Floorplan Editing (Assignment)** or select any custom task with "Assignment" as the editing mode to manipulate LogicLock regions.



To learn more about LogicLock methodology, refer to the LogicLock *Design Methodology* chapter in volume 2 of the Quartus II Handbook.

The Chip Planner shows the connections between LogicLock regions. Rather than showing multiple connection lines from one LogicLock region to another, you can select the option of viewing connections

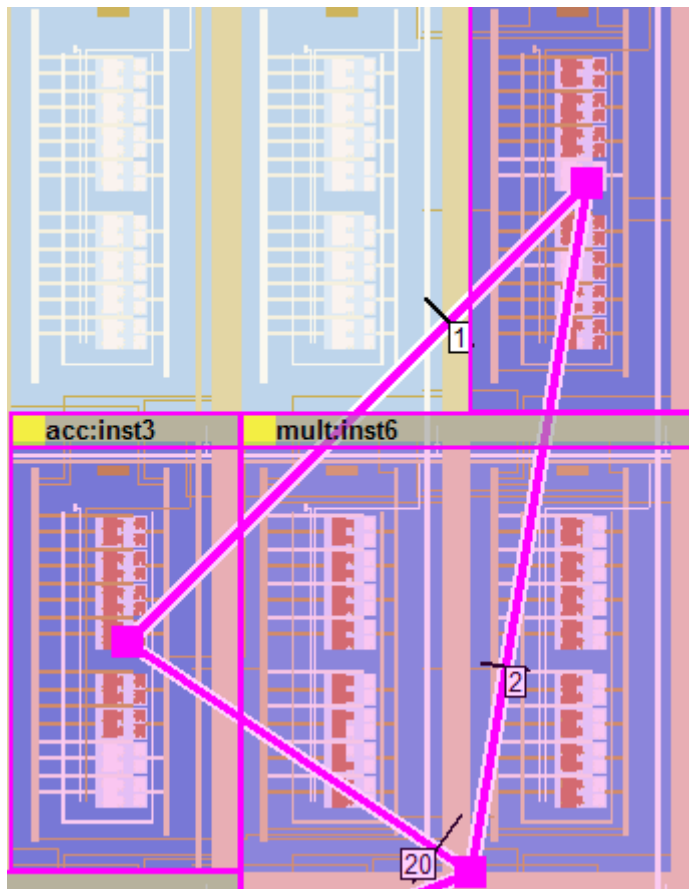
between LogicLock regions as a single bundled connection (Figure 15–24). To use this option, open the Chip Planner floorplan, on the View menu, click **Generate inter-region bundles**.

In the **Generate inter-region bundles** dialog box, specify the **Source node to region fanout less than** and the **Bundle width greater than** values.



For more information on the parameters in the **Generate inter-region bundles** dialog box, refer to the Quartus II Help.

Figure 15–24. Generate Inter-Region Bundles Box



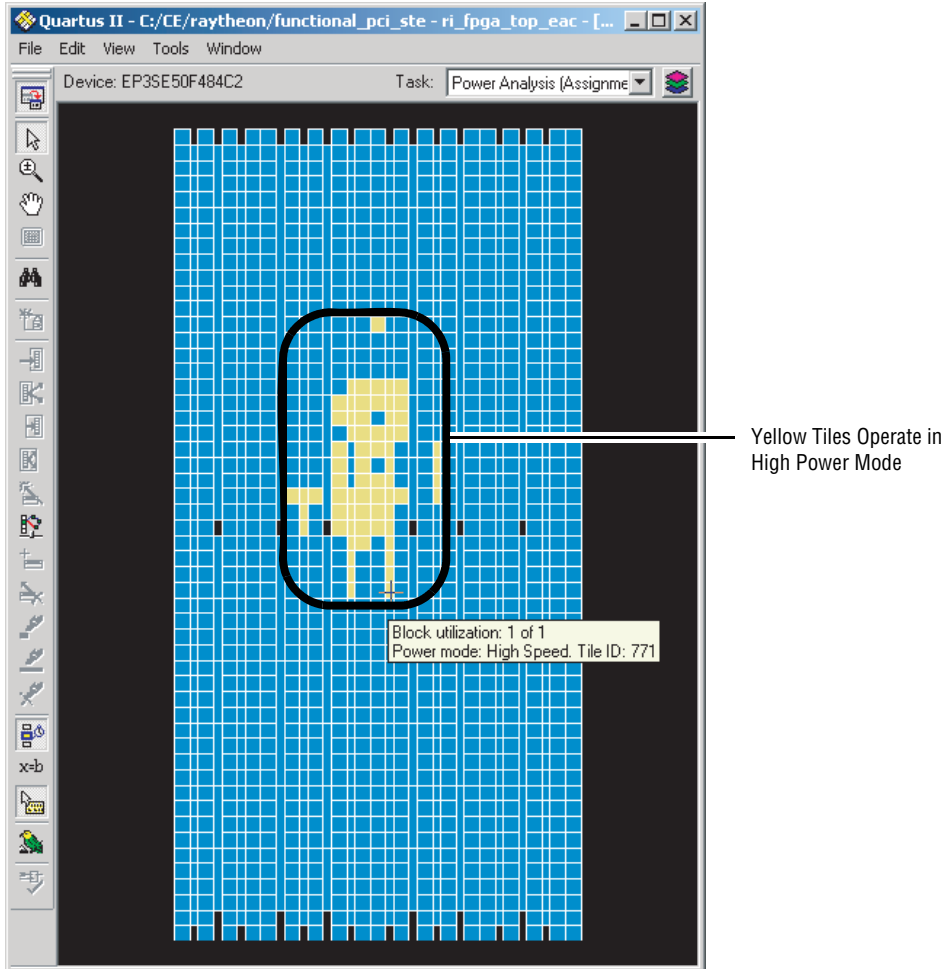
Viewing High and Low Power Tiles in Stratix III Devices

Chip Planner has a predefined task, **Power Analysis (Assignment)**, which shows the power map of a Stratix III device. Stratix III devices have Adaptive Logic Modules (ALMs) which can operate in either high power or low power mode. The high power and low power mode is set during the fitting process in the Quartus II software. These ALMs are grouped together to form a larger block called a tile.



To learn more about power analyses and optimizations in Stratix III devices, refer to application note *AN 437: Power Optimization in Stratix III FPGAs*.

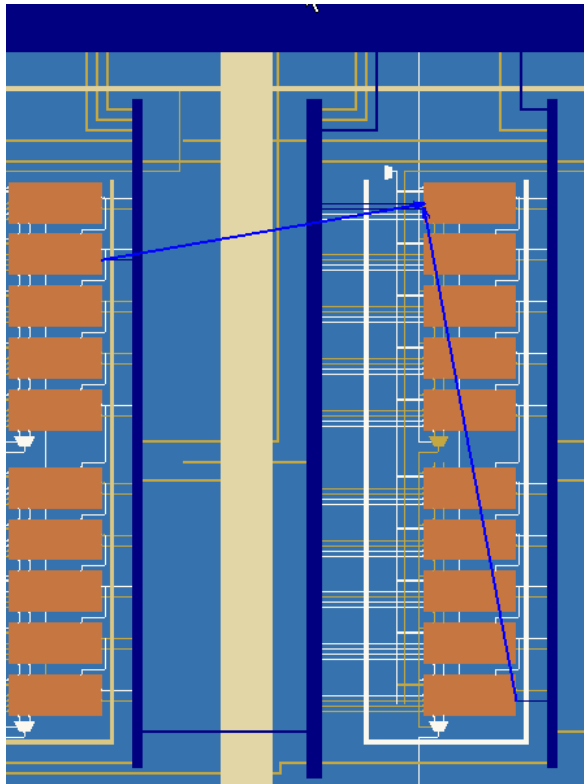
When the **Power Analysis (Assignment)** task is selected in Chip Planner for Stratix III devices, low and high power tiles are displayed in different colors; yellow colored tiles operate in a high power mode, while blue colored tiles operate in a low power mode (see [Figure 15–25](#)). The editing mode in the Power Analysis task is “Assignment.” In this mode, you can perform all floor plan-related functions for this task, however, you cannot edit any tiles to change the its power mode.

Figure 15–25. Viewing High and Low Power Tiles in a Stratix III Device

Routing Channels for a Path

To determine the routing channels between connections, click the Highlight Routing icon on the toolbar. [Figure 15–26](#) shows the routing channels used for the selected path in the Chip Planner.

Figure 15–26. Highlight Routing



Resource Property Editor

You can view and edit the following resources with the Resource Property Editor:

- **Logic Element** (Stratix, Stratix GX, Cyclone II, Cyclone, MAX II) on page 15–33
- **Adaptive Logic Module** (Stratix II, Stratix II GX, Stratix III) on page 15–38
- **FPGA I/O Elements** (I/O resources) on page 15–42
- **PLL Properties** on page 15–60

You can view (but not edit) the following resources with the Resource Property Editor

- RAM
- DSP blocks

Logic Element

An Altera Logic Element (LE) contains a four-input LUT, which is a function generator that can implement any function of four variables. In addition, each LE contains a register that is fed by the output of the LUT or by an independent function generated in another LE.

You can use the Resource Property Editor to view and edit any LE in the FPGA. Open the Resource Property Editor for an LE by choosing **Locate in Resource Property Editor** (right-click menu) on an LE from one of the following views:

- Timing Closure Floorplan
- RTL Viewer
- Technology Map Viewer
- Node Finder
- Chip Planner



For more information about LE architecture for a particular device family, refer to the device family handbook or data sheet.

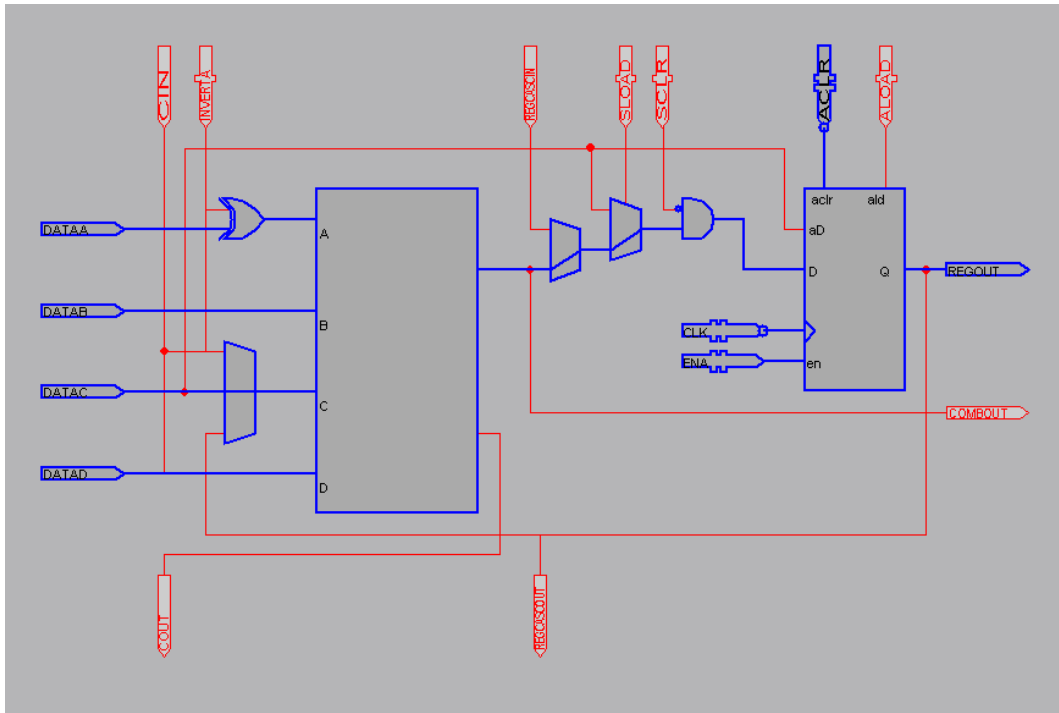
You can use the Resource Property Editor to change the following LE properties:

- Create new LE atoms
- Move existing LE atoms
- Data input to the LUT
- LUT mask or LUT equation

Logic Element Schematic View

Figure 15–27 shows how the LE appears in the Resource Property Editor. Figure 15–27 also shows an LE that uses the DATAC and DATAD inputs and the COMBOUT output.

Figure 15–27. Stratix LE Architecture Notes (1), (2)



Notes to Figure 15–27:

- (1) By default, the Quartus II software displays the used resources in blue and the unused in gray. For Figure 15–27, the used resources are in blue and the unused resources are in red.
- (2) For more information on the Stratix device’s LE Architecture, refer to the *Stratix Device Handbook*.

LE Properties

Figure 15–28 shows the properties that can be viewed for a selected LE in the Resource Property Editor. On the View menu, click **View Properties** to view these properties.

Figure 15–28. LE Properties

Properties/Modes	Values	Sum Equation	D & (B # C) # ID & A & IB	Node:	filterfaps_inst1xn_1[7]
LUT Mask	FC22	Carry Equation	N/A	COMBOUT(0)	REGOUT(0)
Sum LUT Mask	FC22			ACLK(0)	N/A
Carry LUT Mask	N/A			CLK(0)	N/A
Operation Mode	normal			DATAA(0)	459 ps
Synchronous Mode	on			DATAB(0)	332 ps
Register Cascade Mode	off			DATAD(0)	87 ps
Latch Type	none				

Mode of Operation

An LE can operate in either normal or arithmetic mode.



For more information about the modes of operation, refer to volume 1 of the *Stratix Device Handbook*, volume 1 of the *Cyclone Device Handbook*, or the *MAX II Device Handbook*.

When an LE is configured in normal mode, the LUT in the LE can implement a function of four inputs.

When the LE is configured in arithmetic mode, the LUT in the LE is divided into two 3-input LUTs. The first LUT generates the signal that drives the output of the LUT, while the second LUT generates the carry-out signal. The carry-out signal can only drive a carry-in signal of another LE.

Sum & Carry Equation

You can change the logic function implemented by the LUT by changing the SUM & CARRY equation. When the LE is configured in normal mode, you can only change the SUM equation. When the LE is configured in arithmetic mode, you can change both the SUM and the CARRY equation.

The LUT mask is the hexadecimal representation of the LUT equation. When you change the LUT equation, the Quartus II software automatically changes the LUT mask.

When you change the LUT mask, the Quartus II software automatically computes the LUT equation.

Synchronous Mode

When an LE is in synchronous mode, the synchronous load (`sload`) and synchronous clear (`sclr`) signals are used. You can change the LE to synchronous mode by connecting (or disconnecting) the `sload` and `sclr` signals.

You can invert either the `sload` or `sclr` signal feeding into the LE. If the design uses the `sload` signal in an LE, the signal and its inversion state must be the same for all other LEs in the same LAB. For example, if two LEs in a LAB have the `sload` signal connected, both LEs must have the `sload` signal set to the same value. This is also true for the `sclr` signal.

Register Cascade Mode

When register cascade mode is enabled, the cascade-in port feeds the input to the register. The register cascade mode is used most often when the design implements a series of shift registers. You can change the register cascade mode by connecting (or disconnecting) the cascade-in. However, if you create this port, you must ensure that the source port LE is directly above the destination LE.

Cell Delay Table

The cell delay table describes the propagation delay from all inputs to all outputs for the selected LE.

LE Connections

On the View menu, click **View Properties** to view the connections that feed in and out of an LE. Figure 15–29 shows the LE connections in the **View Properties** window.

Figure 15–29. View Properties

Input Port name	Signal name	Latch info	Output Port name	Signal name	Latch info
ACLR	filterfreset	N/A	COMBOUT	filterfaps:inst1Selector0~14	N/A
ALOAD	<Disconnected>	N/A	COUT	<Disconnected>	N/A
CIN	<Disconnected>	N/A	REGCASCOUT	<Disconnected>	N/A
CLK	filterfclk	N/A	REGOUT	filterfaps:inst1xn_1[7]	N/A
DATAA	filterfaps:inst1xn[7]	N/A			
DATAB	filterfstate_minst1sel[1]	N/A			
DATAD	filterfstate_minst1sel[0]	N/A			
ENA	filterfreset	N/A			

Delete an LE

To delete an LE, perform the following steps:

1. Locate the LE in the Resource Property Editor. Choose **Locate** from the right-click menu, click **Locate in Resource Property Editor** on an LE.
2. Remove Fan-out Connections. After you locate the LE in the Resource Property Editor, delete the fan-out connections. On the right-click menu, choose **Remove**, and click **Fanouts** on all outputs. Figure 15–30 shows the dialog box that appears.

Create a New LE

To create an LE with Chip Planner, perform the following steps:

1. On an empty location in Chip Planner, right-click **Create Atom**. The Create Logic Cell dialog box is displayed.
2. Specify the name of the atom in the **Create Logic Cell Atom** dialog box and click **OK**.
3. After you have created the atom, connect it to other atoms by selecting **Edit Connection** and clicking **Other** on an input. In the dialog box that appears, enter the name of the signal that you want to connect to. If you do not know the name of the signal, use the Node Finder to find the signal.
4. To modify the functionality of an LE by setting the sum equation, double-click in the **Sum Equation** field of the logic cell properties and modify the equation. [Figure 15–32](#) shows the Sum Equation and Carry Equation fields.

Figure 15–32. LUT Equation

Sum Equation	A \$ B \$ C
Carry Equation	A & !B & !C # !A & !(C # !B)

Adaptive Logic Module

The basic building block of logic in the Stratix II, Stratix II GX, and Stratix III architecture is the adaptive logic module (ALM). The ALM provides advanced features with efficient logic utilization. Each ALM contains a variety of LUT-based resources that can be divided between two adaptive LUTs (ALUTs). With up to eight inputs to the two ALUTs, each ALM can implement various combinations of two functions. This adaptability allows the ALM to be completely backward-compatible with four-input LUT architectures. One ALM can implement any function with up to six inputs and certain seven-input functions. In addition to the adaptive LUT-based resources, each ALM contains two programmable registers, two dedicated full adders, a carry chain, a shared arithmetic chain, and a register chain. The ALM can efficiently implement various arithmetic functions and shift registers using these dedicated resources.

You can implement the following types of functions in a single ALM:

- Two independent 4-input functions
- An independent 5-input function and an independent 3-input function
- A 5-input function and a 4-input function, if they share one input
- Two 5-input functions, if they share two inputs
- An independent 6-input functions
- Two 6-input functions, if they share four inputs and share function
- Certain 7-input functions

You can change the following ALM properties:

- Create new ALM atoms
- Move existing ALM atoms
- LUT mask or LUT equation

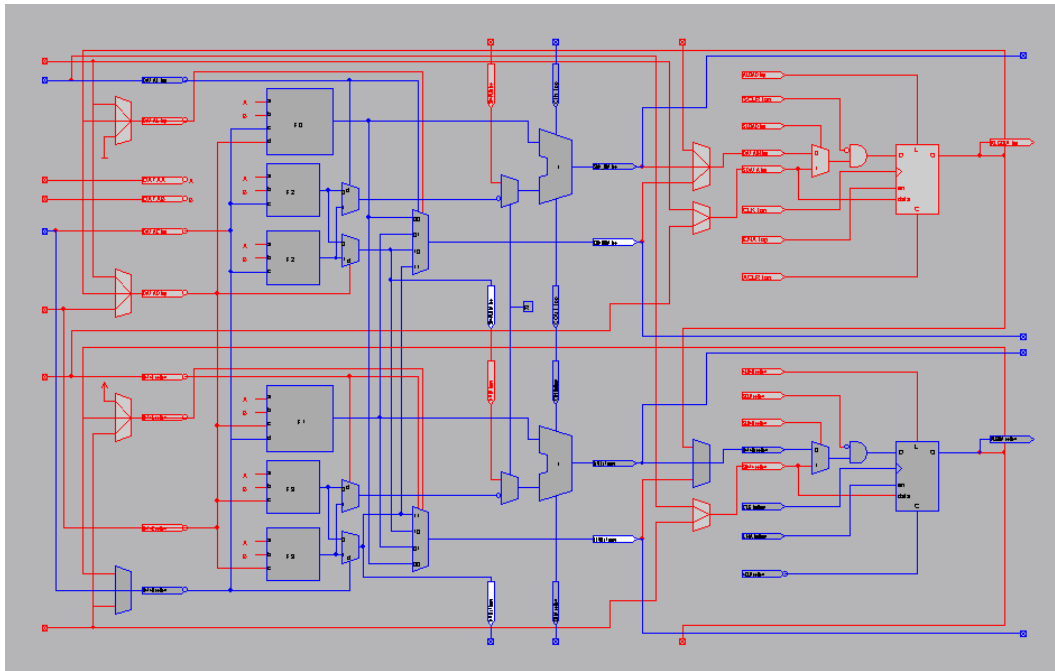
ALM Schematic

You can view and edit any ALM with the Resource Property Editor (Figure 15-33). To view a specific ALM in the Resource Property Editor, right-click the ALM in the Timing Closure Floorplan, RTL Viewer, Node Finder, or Chip Planner. From the drop-down menu, click **Locate in Resource Property Editor**.



For a detailed description of the ALM, refer to the device handbooks for Stratix II, Stratix II GX, or Stratix III device families.

Figure 15–33. ALM Schematic *Note (1)*



Note to Figure 15–33:

- (1) By default, the Quartus II software displays the used resources in blue and the unused in gray. For Figure 15–33, the used resources are in blue and the unused resources are in red.

ALM Properties

The properties displayed for the ALM include an equations table that shows the name and location of each of the two combinational nodes and two register nodes in the ALM, the individual LUT equations for each of the combinational nodes and the combout, sumout, carryout, and shareout equations for each combinational node.

Figure 15–34 illustrates the properties used by the selected ALM.

Figure 15–34. ALM Properties

Properties/Modes	Values		
F0 LUT Mask	FFFF		
F1 LUT Mask	FFFF		
F2 LUT Mask	0000		
F3 LUT Mask	0000		

[-] Top Combinational Node	filterfaps.instbxn_1[4]~feeder
Location String	LCCOMB_X31_Y12_N12
Latch Type	none
F0 LUT Equation	vcc
F1 LUT Equation	N/A
F2 LUT Equation	gnd
F3 LUT Equation	N/A
Combout Equation	F
Sumout Equation	N/A
Carryout Equation	N/A
Shareout Equation	N/A
[-] Top Register Node	filterfaps.instbxn_1[4]
Location String	LCFF_X31_Y12_N13
[-] Bottom Combinational Node	filterfaps.instbxn[4]~feeder
Location String	LCCOMB_X31_Y12_N14
Latch Type	none
F0 LUT Equation	N/A
F1 LUT Equation	vcc
F2 LUT Equation	N/A
F3 LUT Equation	gnd
Combout Equation	F
Sumout Equation	N/A
Carryout Equation	N/A
Shareout Equation	N/A
[-] Bottom Register Node	filterfaps.instbxn[4]
Location String	LCFF_X31_Y12_N15

In Figure 15–34, the ALM uses the bottom combinational node. The equation for that node is described in the `combout` equation. To determine which inputs in the ALM are used, examine the equation. In this example, `DATAE` and `DATAF` control the select signals that drive the bottom multiplexer and the `DATAA`, `DATAB` and `DATAC` signals drive the LUTs. The equations for the atom use variables that correspond to the input of the ALM. For example, the `DATAA` input corresponds to the “A” variable in the equation.

ALM Timing Table

Figure 15–35 describes the propagation delay from all inputs to all outputs for the selected ALM.

Figure 15–35. ALM Timing

Node: snapl xmit_packet:xmit_packet dest_id[7]~620	
	COMBOUT(0)
DATAA(0)	382 ps
DATAD(0)	275 ps
DATAE(0)	155 ps
DATAF(0)	53 ps

ALM Connections

On the View menu, click **View Properties** to view the connections that feed in and out of an LE (Figure 15–36).

Figure 15–36. ALM Connections

Input Port name	Signal name	Latch info	Output Port name	Signal name	Latch info
ACL bottom	fitrefreset~clkctrl	N/A	COMBOUT bottom	fitrefitaps:instxn[4]~feeder	N/A
ACL top	fitrefreset~clkctrl	N/A	COMBOUT top	fitrefitaps:instxn_1[4]~feeder	N/A
ALOAD bottom	<Disconnected>	N/A	COUT bottom	<Disconnected>	N/A
ALOAD top	<Disconnected>	N/A	COUT top	<Disconnected>	N/A
CIN bottom	<Disconnected>	N/A	REGOUT bottom	fitrefitaps:instxn[4]	N/A
CIN top	<Disconnected>	N/A	REGOUT top	fitrefitaps:instxn_1[4]	N/A
CLK bottom	fitrefclk~clkctrl	N/A	SHAREOUT bottom	<Disconnected>	N/A
CLK top	fitrefclk~clkctrl	N/A	SHAREOUT top	<Disconnected>	N/A
DATAA	<Disconnected>	N/A	SUM_OUT bottom	<Disconnected>	N/A
DATAB	<Disconnected>	N/A	SUM_OUT top	<Disconnected>	N/A

FPGA I/O Elements

Altera FPGAs, with high-performance I/O elements packed with up to six registers, are equipped with support for a number of I/O element standards allowing you to run your design at peak speeds.



For a detailed description of the Stratix III, Stratix, Cyclone II, Cyclone, or MAX II device I/O elements, refer to the applicable device handbook.

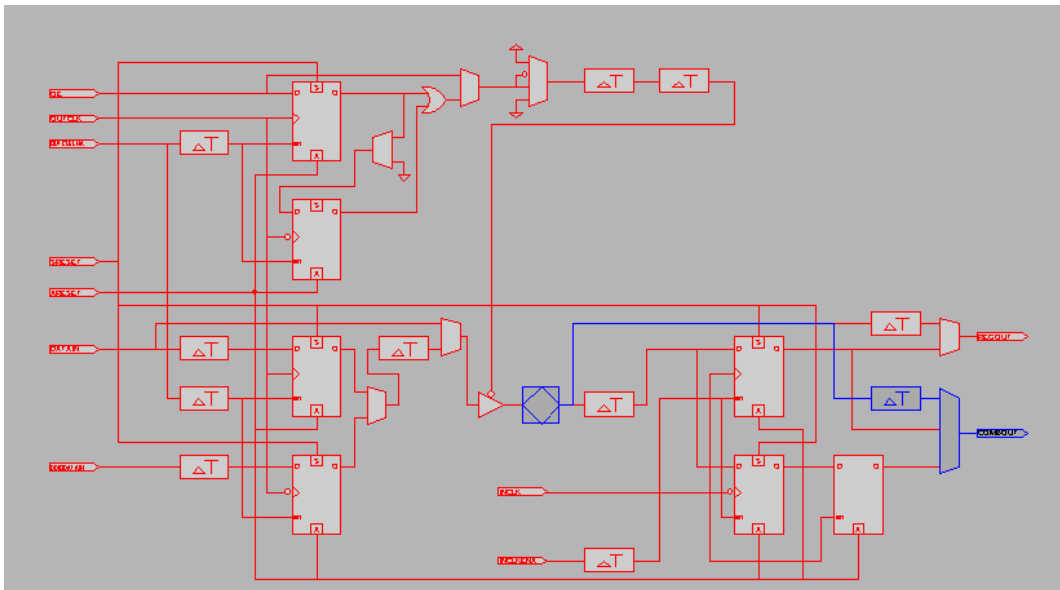
You can change the following I/O properties:

- Create new I/O atom
- Move existing I/O atom
- Delay chain
- Bus hold
- Weak pull up
- Slow slew rate
- I/O standard
- Current strength
- Extend OE disable
- PCI I/O
- Register reset mode
- Register synchronous reset mode
- Register power up
- Register mode

Stratix II, Stratix, & Stratix GX I/O Elements

The I/O elements in Stratix series device families contains a bidirectional I/O buffer, six registers, and a latch for a complete bidirectional single data rate or DDR transfer. [Figure 15–37](#) shows the Stratix and Stratix GX I/O element structure. The I/O element structure contains two input registers (plus a latch), two output registers, and two output enable registers.

Figure 15–37. Stratix & Stratix GX Device I/O Element or Structure *Notes (1), (2)*

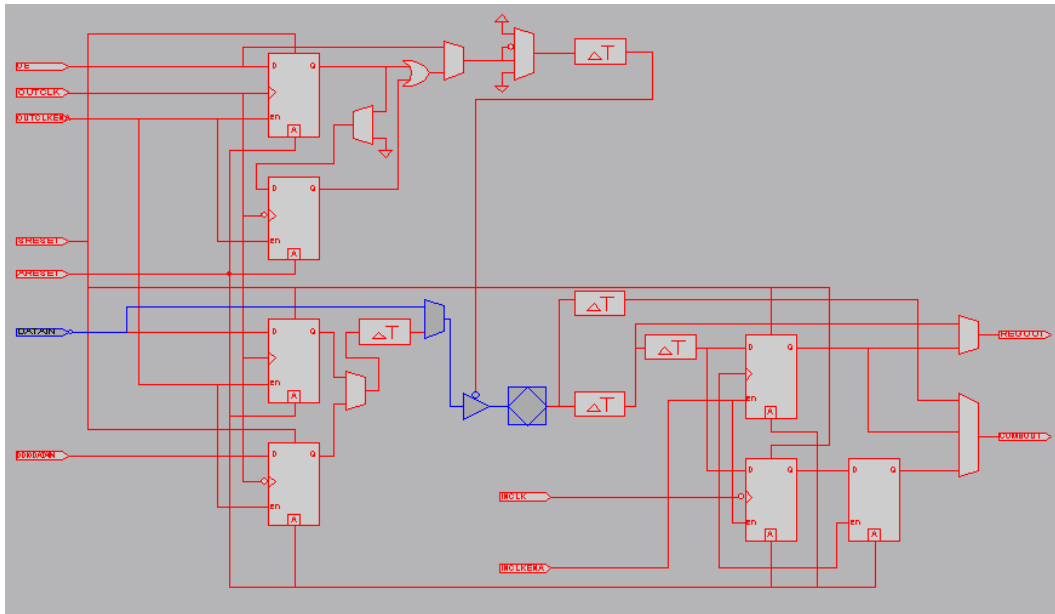


Notes to [Figure 15–37](#):

- (1) By default, the Quartus II software displays the used resources in blue and the unused in gray. In [Figure 15–37](#), the used resources are in blue and the unused resources are in red.
- (2) For more information about I/O elements in Stratix and Stratix GX devices, refer to the *Stratix Device Handbook* and the *Stratix GX Device Handbook*.

Figure 15–38 shows the Stratix II I/O element structure.

Figure 15–38. Stratix II Device I/O Element or Structure Notes (1), (2)



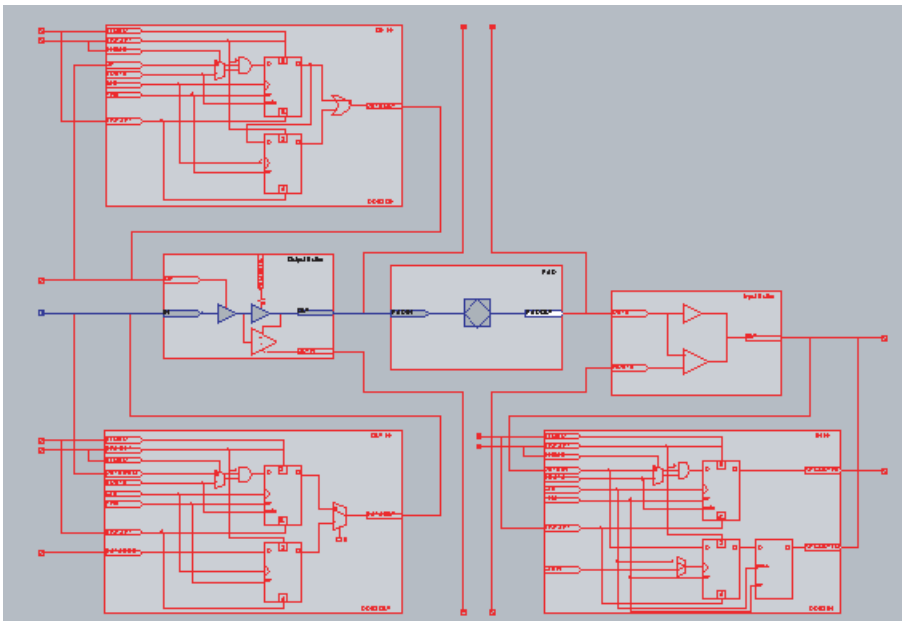
Notes to Figure 15–38:

- (1) By default, the Quartus II software displays the used resources in blue and the unused in gray. In Figure 15–38, the used resources are in blue and the unused resources are in red.
- (2) For more information about I/O elements in Stratix II devices, refer to the *Stratix II Device Handbook*.

Stratix III I/O Elements

The I/O element (IOE) in Stratix III devices contains a bi-directional I/O buffer and I/O registers to support a complete embedded bi-directional single data rate or DDR transfer (shown in Figure 15–39). The IOEs are located in I/O blocks around the periphery of the Stratix III device. The I/O registers are composed of the input path for handling data from the pin, the output path for handling data from the core to the pin, and the output enable (OE) Path for handling the OE signal for the output buffer. The output and OE paths are divided into output/OE registers, alignment registers, and HDR blocks. You can bypass each block of the output and output-enable path.

Figure 15–39. Stratix III Device I/O Element or Structure *Notes (1), (2)*



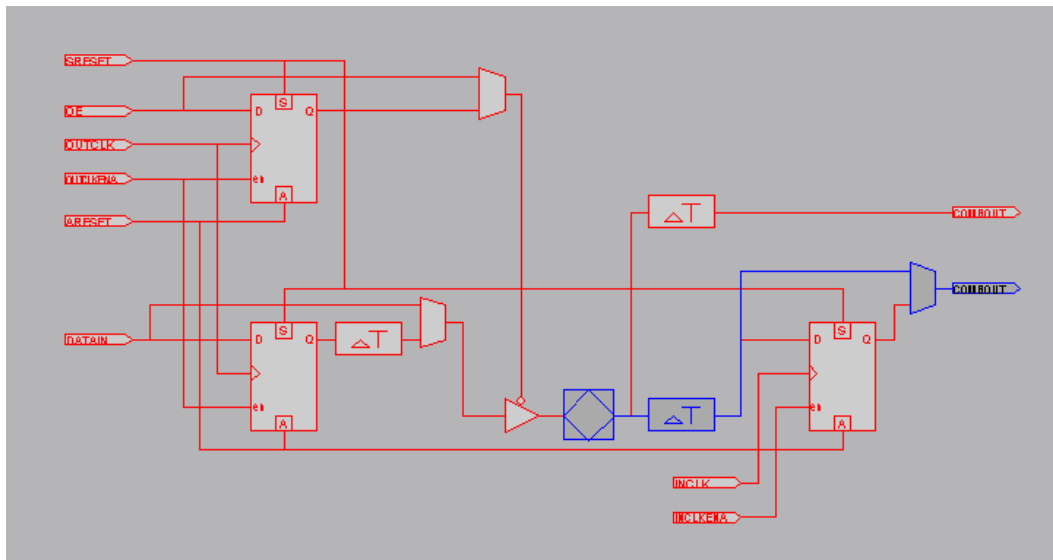
Notes to Figure 15–39:

- (1) By default, the Quartus II software displays the used resources in blue and the unused in gray. In Figure 15–39, the used resources are in blue and the unused resources are in red.
- (2) For more information about I/O elements in Stratix III devices, refer to the *Stratix III Device Handbook*.

Cyclone II & Cyclone I/O Elements

The I/O elements in Cyclone II and Cyclone devices contain a bidirectional I/O buffer and three registers for complete bidirectional single data-rate transfer. Figure 15–40 shows the Cyclone II and Cyclone I/O element structure. The I/O element contains one input register, one output register, and one output enable register.

Figure 15–40. Cyclone II & Cyclone Device I/O Elements or Structure Notes (1), (2)



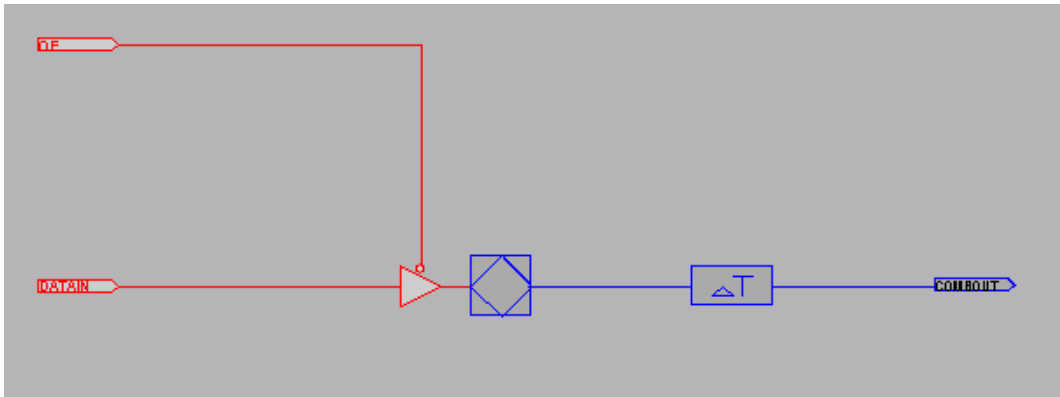
Notes to Figure 15–40:

- (1) By default, the Quartus II software displays the used resources in blue and the unused in gray. In Figure 15–40, the used resources are in blue and the unused resources are in red.
- (2) For more information about I/O elements in Cyclone II and Cyclone devices, refer to the *Cyclone II Device Handbook* and *Cyclone Device Handbook*, respectively.

MAX II I/O Elements

MAX II device I/O elements contain a bidirectional I/O buffer. Figure 15–41 shows the MAX II I/O element structure. Registers from adjacent LABs can drive to or be driven from the I/O element's bidirectional I/O buffers.

Figure 15–41. MAX II Device I/O Elements or Structure *Notes (1), (2)*



Notes for Figure 15–41:

- (1) By default, the Quartus II software displays the used resources in blue and the unused in gray. In Figure 15–41, the used resources are in blue and the unused resources are in red.
- (2) For more information about I/O elements in MAX II devices, refer to the *MAX II Device Handbook*.

I/O Elements Features in the Resource Property Editor

You use the Resource Property Editor to view, change connectivity, and edit the properties of the I/O elements. You use the Chip Planner to change placement, delete, and create new I/O elements. In addition, all of these operations can be performed on Stratix III, Stratix II GX, Stratix II, Stratix, Stratix GX, Cyclone II, Cyclone, and MAX II devices.

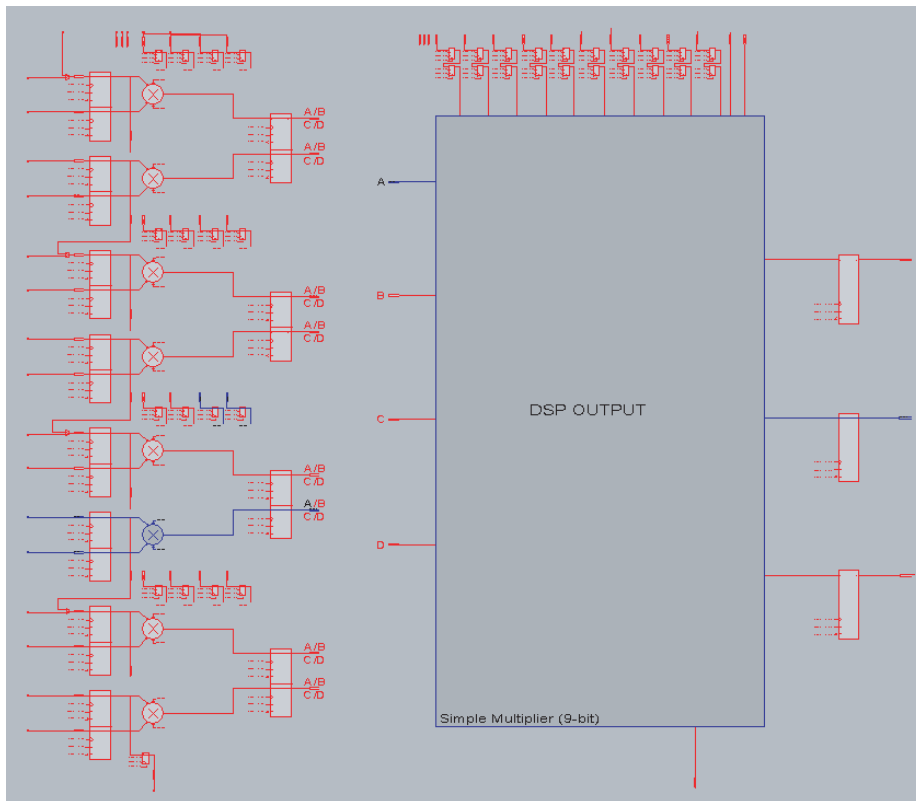
FPGA RAM Blocks

In the Quartus II software beginning with version 6.1, you can view the architecture of different RAM blocks in the device. Figure 15–42 shows a M9k RAM view in a Stratix III device. You can only view the connections and properties of RAM blocks in resource property editor.

FPGA DSP Blocks

Dedicated hardware DSP circuit blocks on Altera devices provide performance benefits for the critical DSP functions in your design. Beginning with version 6.1 of the Quartus II software, you can view the architecture of DSP blocks in resource property editor for Stratix and Cyclone series of devices. [Figure 15–43](#) shows a view of a DSP architecture in a Stratix III device. You can only view the properties and connections of DSP blocks in resource property editor.

Figure 15–43. DSP Block View in a Stratix III Device *Notes (1)*



Notes for [Figure 15–43](#):

- (1) By default, the Quartus II software displays the used resources in blue and the unused in gray. In [Figure 15–43](#), the used resources are in blue and the unused resources are in red.

Change Manager

The Change Manager maintains a record of every change that you perform with the Resource Property Editor. Each row in the Change Manager represents one change that you made with the Resource Property Editor. The changes are numbered sequentially, such that the larger the number, the more recent the change.


More complex changes are marked in the Change Manager with a . You can expand a complex entry in the Change Manager to reveal all the changes that occurred. An example of a complex change is the creation or deletion of an atom.

Table 15–2 summarizes the information shown by the Change Manager.

Column Name	Description
Index	Identifies, by a sequential number, change records corresponding to changes made in the Chip Planner or Resource Property Editor. In the case of complex change records, the index column identifies not only the main change, but also any component changes.
Node Name	Uniquely identifies the resource to which a change has been made.
Change Type	Identifies the type of change that has been made to the resource.
Old Value	Lists the value of the resource immediately prior to the change being made.
Target Value	Lists the desired target value (new value) that you have established using the Resource Property Editor, Chip Planner, or SignalProbe.
Current Value	Lists the value of the resource in the netlist that is currently active in memory (as opposed to the value in the netlist saved on disk, which may be different if you have made changes and not yet used the Check and Save All Netlist Changes command). The Current Value field can contain one of four possible values: <ul style="list-style-type: none"> ● current value = old value. Indicates that no change has been applied. ● current value = target value. Indicates that a change has been applied. ● current value = neither old nor target value. Indicates that an invalid change was attempted. ● current value = <code>Data Not Available</code> Indicates that you may be using an incomplete data set. This can occur in the following instances: <ul style="list-style-type: none"> — No netlist exists because the design in use has not been compiled on the current computer. — The node you are trying to change does not exist in the netlist because the source code has been changed. — The node you are trying to change does not exist in the netlist because the node name was originally assigned by the Compiler, and you have subsequently performed a Fitter operation using different optimization parameters, resulting in changed node names.
Disk Value	Lists the current value of the resource on disk.
Comment	Lets you add a comment to a change record in the Change Manager . To add a comment to a change record, double-click in the Comment field of the record you want to annotate, and type the desired comment.

After you complete all of your design modifications, check the integrity of the netlist by right-clicking in the Change Manager and clicking **Check & Save All Netlist Changes**. If the applied changes successfully pass the netlist check, they are written to disk. If the changes do not pass the netlist check, all changes made since the previous successful netlist check are reversed. Figure 15–44 shows the Change Manager.

Colored indicators in the **Current Value** and **Disk Value** columns indicate the present status of the data in those columns. Green in the **Current Value** column indicates that the change has occurred and the value there is the same as the value in memory. Blue in the **Disk Value** column indicates that the change has successfully passed. Choose **Check & Save All Netlist Changes** (right-click menu), and the value in that column is the same as the value on disk.

Figure 15–44. Change Manager Results

Index	Node Name	Change Type	Old Value	Target Value	Current Value	Disk Value
1	test1	SignalProbe	Disconnected	filterinst4	Disconnected	Disconnected
2	mult_inst6lpm_mult_lpm_mult_componentlmu...	Modify Source	filteritaps.in...	Disconnected	filteritaps instxn [4]	filteritaps instxn [4]
3	mult_inst6lpm_mult_lpm_mult_componentlmu...	Modify Source	filteritaps.in...	Disconnected	filteritaps instxn [4]	filteritaps instxn [4]
4	taps instxn_1 [4] Feeder:DATAF:0	Modify Source	filteritaps.in...	Disconnected	filteritaps instxn [4]	filteritaps instxn [4]
5	mult_inst6lpm_mult_lpm_mult_componentlmu...	Modify Source	filteritaps.in...	Disconnected	Disconnected	Disconnected
6	mult_inst6lpm_mult_lpm_mult_componentlmu...	Modify Source	filteritaps.in...	Disconnected	Disconnected	Disconnected
7	taps instxn_1 [0] Feeder:DATAF:0	Modify Source	filteritaps.in...	Disconnected	Disconnected	Disconnected

Change Manager: Netlist Check Required - 3 Pending Changes



For more information about SignalProbe pins, refer to the *Quick Design Debugging Using SignalProbe* chapter in volume 3 of the *Quartus II Handbook*.



Each line in the Change Manager represents a change record. Simple changes appear as a single line. More complex changes, which require that several actions be performed to achieve the change, appear as a single line marked by a **+**. Click **+** to show all the component actions performed as part of the change.

Complex Changes in the Change Manager

Certain types of changes that you make in the **Resource Property Editor** or the **Chip Planner** (including creating or deleting atoms and changing connectivity) may appear to be self-contained, but these changes are actually composed of multiple actions. These types of changes are recorded with a **+** sign in the **Index** column. Figure 15–45 shows the Change Manager.

Figure 15–45. Change Manager

Index	Node Name	Change Type	Old Value	Target Value	Current Value	Disk Value
1	test1	SignalProbe	Disconnected	filterinst4	Disconnected	Disconnected








In the example shown in [Figure 15–45](#), a new atom is created. The change record in the **Change Manager** is a single-line representation of the actual change actions that occurred. You expand the change record to show the component actions that make up the change by clicking  ([Figure 15–46](#)).

Figure 15–46. Change Manager Results

Index	Node Name	Change Type	Old Value	Target Value	Current Value	Disk Value
 1	test	New Lcell Comb	None	Exists	Exists	None
2	Iniosii_mandelbrot_toplmandelbrot_sys:instr...	Location Index	LCCOMB_X...	LCCOMB_X...	LCCOMB_X...	LCCOMB_...
3	Iniosii_mandelbrot_toplmandelbrot_sys:instr...	Location Index	LCCOMB_X...	LCCOMB_X...	LCCOMB_X...	LCCOMB_...
4	Iniosii_mandelbrot_toplmandelbrot_sys:instr...	Location Index	LCCOMB_X...	LCCOMB_X...	LCCOMB_X...	LCCOMB_...
 5	Iniosii_mandelbrot_toplmandelbrot_sys:instr...	Modify Source	Iniosii_mand...	Disconnected	Disconnected	Disconnec...

After clicking , you can see that creation of an atom consists of three actions:

-  The creation of a new logic cell
-  The creation of an output port on the newly created logic cell
-  The assignment of a location index to the newly created logic cell

You cannot select individual components of a complex change record; if you select any part of a complex change record, the entire complex change record is selected.

 For examples of managing changes with the Change Manager, refer to “Example of Managing Changes With the Change Manager” in the Quartus II Help.

Managing SignalProbe Signals

The SignalProbe assignments that you create from the **SignalProbe Pins** dialog box under the Tools menu are recorded in the Change Manager. After you have created a SignalProbe assignment, you can use the Change Manager to quickly disable SignalProbe assignments by selecting **Revert to Last Saved Netlist** from the right-click menu in the Change Manager.

Figure 15–47. Change Manager Results after a SignalProbe Pin is Created

Index	Node Name	Change Type	Old Value	Target Value	Current Value	Disk Value
1	test1	SignalProbe	Disconnected	fitref1inst4	fitref1inst4	Disconnected
1.1	test1	New Output Pin	None	Exists	Exists	None
1.2	test1:DATAIN:0	Modify Source	Disconnected	fitref1inst4	fitref1inst4	Disconnected



For more information about SignalProbe pins, refer to the chapter *Quick Design Debugging Using SignalProbe* in volume 3 of the *Quartus II Handbook*.

Exporting Changes

You can export all your changes to a tool command language (Tcl) script, a Comma Separated Value (.csv) file, or a Text (.txt) file. The Tcl file enables you to write a script that reapplies changes that were erased by compilation. You can also write a script that applies to other Quartus II software projects that you create. The Comma-Separated Value or Text files provide a list of changes in a tabular format. To export changes, perform the following steps:

1. On the right-click menu, choose **Export Changes**.
2. Specify the Tcl file name.
3. Click **OK**.

The resulting Tcl script can also implement similar changes to another Quartus II design.

Performing ECOs with Chip Planner

Chip Planner lets you quickly perform ECOs on your post compilation netlist. To perform ECOs, you must select the task **Post Compilation Editing (ECO)** or any custom task with ECO as the editing mode. Common examples of ECOs are changing the properties of a PLL inside your device, changing the drive strength of an I/O, correcting a functional flaw in the design or routing an internal node or register to an output pin (SignalProbe ECO).



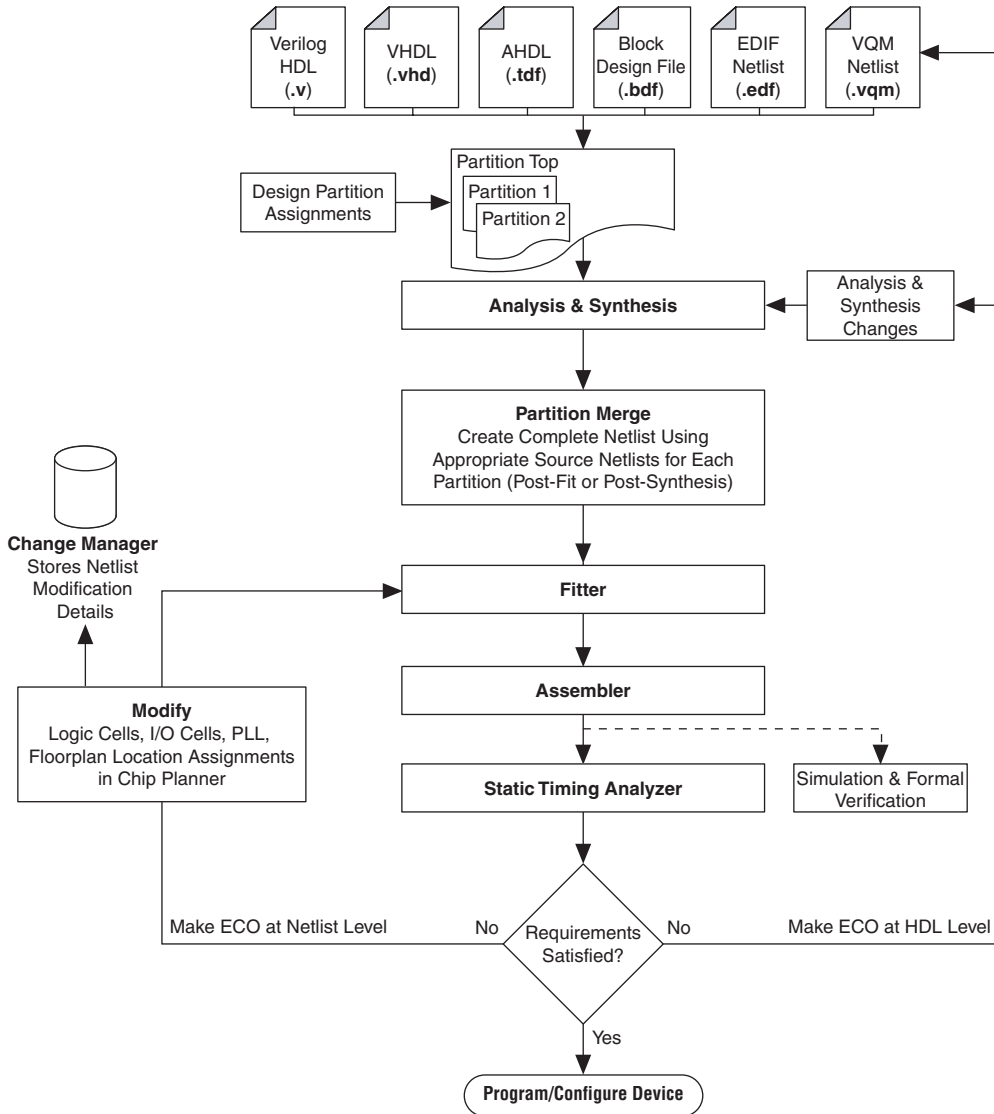
Refer to [“Common ECO Applications” on page 15–58](#) to learn how to use the Chip Planner to perform ECOs. SignalProbe changes are also implemented as ECOs. To learn more about SignalProbe ECO flow, refer to chapter *Quick Design Debugging with SignalProbe* in volume 3 of the *Quartus II Handbook*.

Using Chip Planner in the ECO Design Flow

Bugs exist often in the RTL code or the design specifications change midway through the design cycle. The challenge lies in efficiently accommodating these types of design change issues. Traditionally, designers had to return to the source code, make the appropriate changes, and then go through the entire design flow again, including synthesis, place-and-route, and verification.

With Altera Chip Planner, you can shorten the design cycle time significantly. When changes are made to your design as ECOs, you do not have to perform a full compilation in the Quartus II software. Instead, you can make changes directly to the post place-and-route netlist, generate a new programming file, and test the revised design by performing a gate-level timing simulation and static timing analysis. This is performed without modifying the source code. You can continue to make changes to your design iteratively using Chip Planner until you correct the problem. [Figure 15-48](#) illustrates how Chip Planner is used in the ECO design flow.

Figure 15–48. Chip Planner ECO Design Flow



ECO Flow with Quartus II Incremental Compilation

Beginning with the Quartus II software version 6.1, the incremental compilation feature is turned on by default. The top-level design is automatically set to a design partition when the incremental compilation feature is on. A design partition during incremental compilation can have different netlist types (netlist types can be set to source HDL, post synthesis, or post-fit). The netlist type indicates whether that partition should be resynthesized or refit during the Quartus II incremental compilation. Incremental compilation saves you time and preserves the placement of unchanged partitions in your design if small changes must be made to some partitions late in the design cycle.



For more information about partitions, their netlist types and the Quartus II incremental compilation, refer to the *Quartus Incremental Compilation Feature for Hierarchical & Team-Based Design* chapter in volume 1 of the Quartus II *Handbook*.

The behavior of ECOs during an incremental compilation depends on the netlist type of your design partitions. To preserve ECO changes, the partitions must be set to a netlist type of post-fit with the Fitter preservation level set to Placement and Routing.

When the top-level partition netlist type is set to post-fit type, and you have ECOs in your design, during recompilation ECOs are preserved. If some ECOs involve a lower-level and top-level partition, and the lower-level partition is set to a netlist type other than post-fit (for example, post-synthesis), then during recompilation the Quartus II Fitter uses the post-fit netlist type for those partitions as well, and a warning message appears in the message window. This is because if ECO changes affect more than one partition or the connection between any partitions, the partitions become linked and all of the higher-level "parent" partitions up to their nearest common parent are all linked and are required to have the same netlist type.

The same rule applies when your top-level partition is set to post-synthesis and one of the lower-level partitions' netlist type is set to post-fit. If you have implemented an ECO that involves these partitions, during recompilation, the Quartus II Fitter uses the post-fit netlist for the top-level partition and the ECO change is retained.

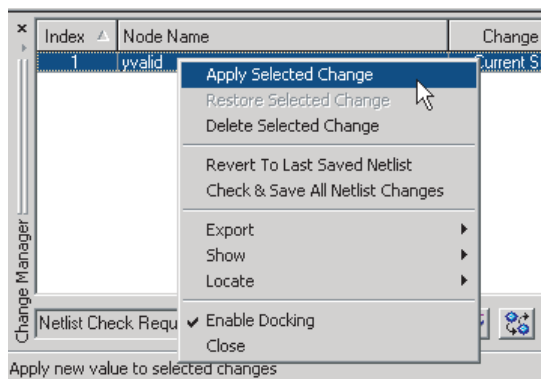
If the partitions are set to use the source code or a post-synthesis netlist, the software issues a warning and the post-fit ECO changes are not included in the new compilation. For example, if your top-level partition netlist type is set to post-synthesis, and either you have no other lower-level partitions or the lower-level partitions netlist type is also set to post-synthesis, then during recompilation your ECOs are not preserved and a warning message appears in the messages window,

indicating that ECO modifications are discarded; however, all of the ECO information is retained in the change manager. In this case, you can apply ECOs from the change manager and perform the Check & Save All Netlist Changes step as described in [“ECO Flow without the Quartus II Incremental Compilation”](#) on page 15–57.

ECO Flow without the Quartus II Incremental Compilation

If you do not use the Quartus II incremental compilation feature and have implemented ECOs, during recompilation of your design those ECOs are not preserved; however, all of the ECOs remain in the Change Manager. To apply a ECO, right-click the Change Manager and select **Apply Selected Change** (Figure 15–49). (If the Change Manager window is not visible at the bottom of your screen, from the View menu, point to **Utility Windows** and click **Change Manager**.)

Figure 15–49. Applying a Change Manager ECO



After applying the selected ECO, do one of the following:

- From the menu within the change manager, click **Check & Save All Netlist Changes**.
- or*
- From Processing menu, point to **Start** and click **Start Check & Save All Netlist Changes**.

Common ECO Applications

This section provides examples of the situations in which you might use an ECO to make a post-compilation change to your design. To help build your system quickly, you can use Chip Planner functions to to:

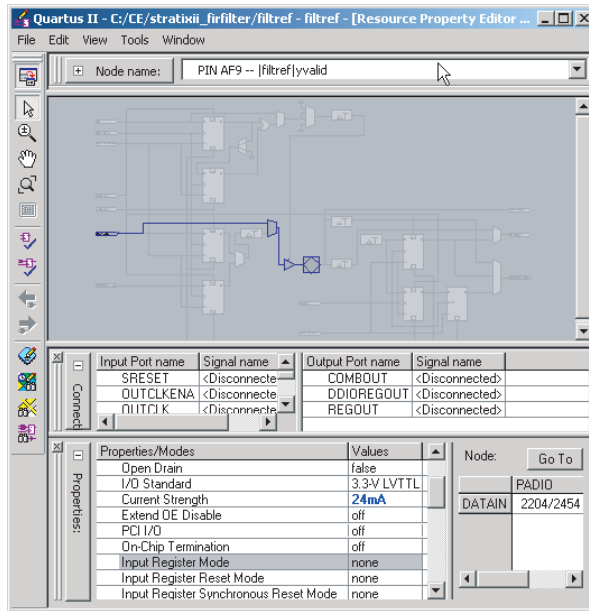
- [Adjust the Drive Strength of an I/O using Chip Planner](#)
- [Modifying the PLL Properties Using Chip Planner](#) (see [page 15–59](#))

Adjust the Drive Strength of an I/O using Chip Planner

To adjust the drive strength of an I/O, follow the steps in this section to run the fitter and assembler to incorporate the ECO changes into the netlist of the design.

1. In Chip Planner, select the **Post Compilation Editing (ECO)** task.
2. Locate the I/O in the **Resource Property Editor**, as shown in [Figure 15–50](#).

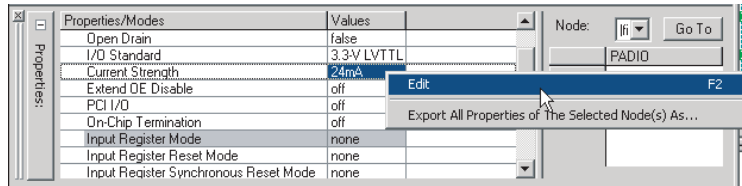
Figure 15–50. I/O in the Resource Property Editor



3. Click the **Drive Strength** box for the selected I/O, then click **Edit**.

- Change the value for the desired drive strength, as shown in [Figure 15–51](#).

Figure 15–51. Changing Drive Strength Value of an I/O



- Right-click the ECO change in the Change Manager tool and click **Check & Save All Netlist Changes** to apply the ECO change.



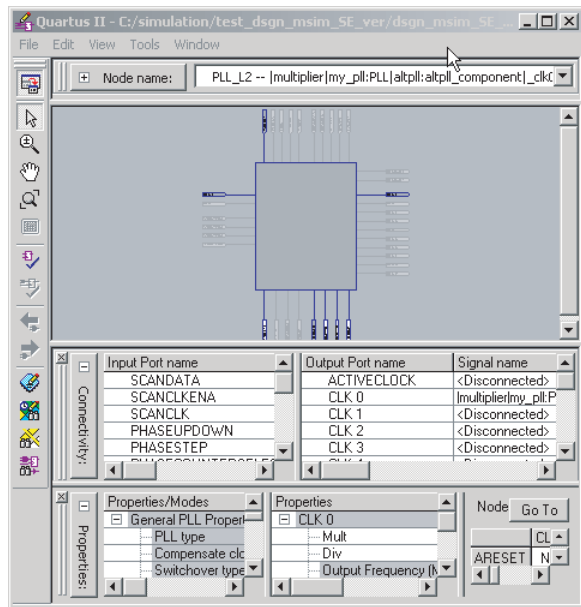
Changing the location of an input/output signal on the physical pins of the device can also be done using the ECOs flow in the Quartus II software. You can drag and move the signal from an existing pin location to a new location while in the Post Compilation Editing (ECO) task in Chip Planner. Afterwards, you can set **Check & Save All Netlist Changes** to compile the ECO.

Modifying the PLL Properties Using Chip Planner

PLLs are used to modify and generate clock signals to meet design requirements. Additionally, PLLs are used for distributing clock signals to different devices in a design, reducing clock skew between devices, improving I/O timing, and generating internal clock signals.

Within the Resource Property editor, use the equations window to set the new phase shift values of the PLL in your design. [Figure 15–52](#) shows a PLL view with its equations window in Resource Property Editor.

Figure 15–52. The PLL View in a Stratix Device



PLL Properties

The Resource Property Editor enables you to modify PLL options, such as phase shift, output clock frequency, and duty cycle. You can also change many of the PLL properties with the Resource Property Editor, including:

- Input frequency
- M VCO tap
- M initial
- M value
- N value
- M counter delay
- N counter delay
- M2 value
- N2 value
- SS counter
- Charge pump current
- Loop filter resistance
- Loop filter capacitance
- Counter delay
- Counter high

- Counter low
- Counter mode
- Initial
- VCO tap

Adjusting the Duty Cycle

Use the following equation to adjust the duty cycle of individual output clocks:

$$(1) \quad \begin{aligned} \text{High \%} &= \text{Counter High} / (\text{Counter High} + \text{Counter Low}) \\ \text{Low \%} &= \text{Counter Low} / (\text{Counter High} + \text{Counter Low}) \end{aligned}$$

Adjusting the Phase Shift

Use the following equations to adjust the phase shift of an output clock of a PLL:

$$(2) \quad \text{Phase Shift} = (\text{Period } V_{CO} \times 0.125 \times \text{Tap } V_{CO}) + (\text{Initial } V_{CO} \times \text{Period } V_{CO})$$



For a detailed description of the settings, refer to the Quartus II Help. For more information about Stratix device PLLs, refer to the *Stratix Architecture* chapter in volume 1 of the *Stratix Device Handbook*.

For normal node, calculate the phase shift with the following settings:

$$\begin{aligned} \text{Tap } V_{CO} &= \text{Counter Delay} - M \text{ Tap } V_{CO} \\ \text{Initial } V_{CO} &= \text{Counter Initial} - M \text{ Initial} \\ \text{Period } V_{CO} &= \text{In Clock Period} \times N / M \end{aligned}$$

For external feedback mode, calculate the phase shift with the following settings:

$$\begin{aligned} \text{Tap } V_{CO} &= \text{Counter Delay} - M \text{ Tap } V_{CO} \\ \text{Initial } V_{CO} &= \text{Counter Initial} - M \text{ Initial} \\ \text{Period } V_{CO} &= \text{In Clock Period} \times N / (M + \text{Counter High} + \text{Counter Low}) \end{aligned}$$

Adjusting the Output Clock Frequency

Use the following equation to adjust the PLL output clock in normal mode:

$$(3) \quad \text{Output Clock Frequency} = \text{Input Frequency} \cdot \frac{M_{\text{initial}}}{N_{\text{initial}} + \text{Counter High} + \text{Counter Low}}$$

Use the following equation to adjust the PLL output clock in external feedback mode:

$$(4) \quad \text{OUTCLK} = \text{INCLK} \cdot \frac{M_{\text{initial}} + \text{External Feedback Counter High} + \text{External Feedback Counter Low}}{N_{\text{initial}} + \text{Counter High} + \text{Counter Low}}$$

Adjusting the Spread Spectrum

Use the following equation to adjust the spread spectrum for your PLL:

$$(5) \quad \% \text{spread} = 1 - \frac{M_2 N_1}{M_1 N_2}$$

Post ECO Steps

This section highlights the operations you can perform after making an ECO change with Chip Planner.

Running the Quartus II Timing Analyzer

After you make an ECO change with the Chip Planner, you must perform static timing analysis of your design with the Quartus II timing analyzer to ensure that your changes have not adversely affected your design's timing performance.

For example, when you turn on one of the delay chain settings for a specific pin, you change the I/O timing. Therefore, to ensure that all timing requirements are still met, you should perform static timing analysis.

Whenever you change your design using Chip Planner, Altera also recommends that you perform a gate-level timing simulation on your design with either the Quartus II Simulator or any third-party EDA simulation tool.



For more information about performing a static timing analysis of your design, refer to the Quartus II *Classic Timing Analyzer* or Quartus II *TimeQuest Timing Analyzer* chapters in volume 3 of the Quartus II *Handbook*.

Generating a Programming File

After you have performed simulation and static timing analysis and are confident that the changes meet your design requirements, generate a new programming file with the Quartus II Assembler. Use the programming file to implement your modified design in an Altera device.

Conclusion

As the time-to-market pressure mounts, it is more and more important to produce a fully functional design in the shortest amount of time. To address this challenge, Altera developed Chip Planner in the Quartus II software suite. Chip Planner enables you to analyze and modify your design floorplan. Also, ECO changes made with Chip Planner do not require a full recompilation, eliminating the lengthy process of RTL modification, resynthesis, and another place-and-route cycle. In summary, Chip Planner shortens the verification cycle and brings timing closure to your design in a shorter period of time.

Document Revision History

Table 15–3 shows the revision history for this chapter.

Date and Version	Changes Made	Summary of Changes
March 2007 v7.0.0	Added Cyclone III device support to Table 15–1 .	—
November 2006 v6.1.0	<ul style="list-style-type: none"> ● Chapter 15 was formally Chapter 14. ● Chip Editor is now called Chip Planner (Floorplan & Chip Editor). ● Added additional information about the Chip Planner features. 	The Chip Editor & Floorplanning tools have been merged into the Chip Planner (Floorplan & Chip Editor). This chapter has been updated to reflect this feature change.
May 2006 v6.0.0	Chapter title changed. Updated for the Quartus II software version 6.0.0: <ul style="list-style-type: none"> ● Updated Chip Editor Chip View Update, ● Added information about the Resource Property Editor Update. ● Updated the Change Manager section. 	—
December 2005 v5.1.1	<ul style="list-style-type: none"> ● Minor typographic update. 	—
October 2005 v5.1.0	<ul style="list-style-type: none"> ● Updated for the Quartus II software version 5.1. ● Chapter 12 was formerly Chapter 11 in version 5.0. 	—
May 2005 v5.0.0	<ul style="list-style-type: none"> ● Updated information. ● Updated figures. ● Removed figures and tables. ● New functionality for Quartus II software 5.0. 	—
January 2005 v2.2.0	Change Manager section update	—
December 2004 v2.1.0	<ul style="list-style-type: none"> ● Chapter 11 was formerly Chapter 10. ● New figures added. ● Reorganized the chapter and updated information and figures. ● New functionality for Quartus II software 4.2. 	—
June 2004 v2.0.0	<ul style="list-style-type: none"> ● Updates to tables and figures. ● New functionality for Quartus II software 4.1. 	—