Introduction

This application note assumes familiarity with the Quartus II incremental compilation flow and floorplanning with the LogicLock feature. This application note contains rules and guidelines for creating a floorplan with the Design Separation Flow.

A basic tenet of a secure and reliable system is the physical and functional independence of the critical subsystems in the design. Systems with redundancy require physical independence to ensure fault isolation—that a failure or corruption of any single subsystem will not adversely affect any other part of the system. Furthermore, should errors occur, physical independence simplifies analysis by enabling developers to evaluate each subsystem separately.

Traditionally, systems that require redundancy implement critical IP structures using multiple devices. The Quartus® II Design Separation flow, used in the Cyclone® III LS device family, provides the ability to design physically independent structures on a single device. This allows system designers to achieve a higher level of integration on a single FPGA, and alleviates increasingly strict Size Weight and Power (SWaP) requirements. Figure 1 illustrates this concept.

Figure 1. Achieving Higher Level Integration on a Single Cyclone III LS Device

The new Quartus II Design Separation flow introduces a secured region to the compilation flow. When you use this flow with the Cyclone III LS device family, a secured region provides physical independence through controlled routing and a boundary of unused resources. Multiple secured regions can be created on a single Cyclone III LS device without unwanted mutual interference. By restricting routing resources and providing a physical guard band of unused LABs, faults or unintended signals originating in one secured region are prevented from adversely affecting any other design block on the device.

The Quartus II Design Separation flow fits into the incremental compilation and LogicLock™ features to floorplan a design by adding the secured region constraint. The following three chapters in the Quartus II Handbook are companion references to this application note.
Design Flow Overview

The Quartus II Design Separation flow isolates design partitions into physically independent regions. Each physically isolated region is known as a secured region in the Design Separation flow. The Design Separation flow uses the incremental compilation flow for partitioning logic and LogicLock region location assignments for floorplanning secured regions. For each design entity that requires physical independence, create a logical design partition for that design entity and assign the design partition to a LogicLock location constraint.

During the floorplanning stage, an additional constraint (a security attribute) is applied to a LogicLock region to confine all routing to the region. When this attribute is applied, a boundary consisting of unused logic array blocks (LABs), known as a fence, is created around the LogicLock region. Signals may only exit or enter a secured region through a Security Routing Interface. A Security Routing Interface is an attribute applied to a LogicLock region.

Figure 2 shows a diagram of the top-down incremental compilation flow with modified floorplan constraints that support the Design Separation flow. Red boxes in the flow diagram highlight steps that are specific to a design with the Design Separation flow enabled. A brief description is given for each of the steps in the flow below and serves as a quick-start guide for the Design Separation flow. Subsequent sections discuss the flow in greater detail.
The following subsections discuss the flow shown in Figure 2 in greater detail.

1. **Set up the design hierarchy for secure partitioning**—Each secured region can only contain a single design partition. Your design hierarchy must support partitioning along logical hierarchical boundaries. If necessary, create wrapper files to create logical boundaries in the design hierarchy to support the design entities that you must separate from the remainder of the design.

2. **Run analysis and elaboration, or any compilation flow that includes this step**—Elaboration is the part of the synthesis process that identifies your design’s hierarchy.
3. **Create design partitions for secured regions**—Each secured region must contain only a single partition.

4. **Create a design floorplan with security attributes**—LogicLock location assignments, create a floorplan for, at minimum, all the entities to be secured in your design. Use the security assignments in the LogicLock Regions window in the Quartus II software to specify the security level of each LogicLock region. Fencing regions are automatically created in your floorplan to isolate the secured LogicLock regions.

5. **Assign design partitions to secured regions**—Assign design partitions to secured LogicLock regions to separate them from each other and from all other hierarchy blocks.

6. **Add I/O pins as members of the secured region that sources or sinks the I/O pin**—If a secured region interfaces with one or more I/O pins, you may want to make the I/O pins members of the secured region. If a secured region has I/O pins as members, that region must overlap the I/O pads. You can add I/O pins as members of a secured LogicLock region using the LogicLock Region Properties dialog box.

7. **Create Security Routing Interface regions**—Each Security Routing Interface is a LogicLock region with the Security Routing Interface attribute applied to it. No logic can be placed into a Security Routing Interface. Only routing resources can be used within a Security Routing Interface. Each Security Routing Interface must abut one or two secured regions. After you’ve created an interface region for each secured region, assign signals that enter or exit the secured region to a routing interface.

   For signals routing between secured regions with different security attributes or between a secured region and an unsecured region, you must lower the security attribute for the signal exiting the stricter security region.

   You can assign signals to a Security Routing Interface and modify the security level of a signal using the Security tab in the LogicLock Regions Properties dialog box. The Security tab is automatically populated with all input and output signals from a secured region after the Quartus II software runs analysis and synthesis, and Partition Merge.

8. **Make I/O assignments**—I/O banks cannot be shared between secured regions. If a secured region contains I/O pins as members, the entire I/O bank is usable only by the secured region that sinks or sources the I/O pin.

9. **Make design changes, set netlist type for each design partition, and compile**—After you perform a full compilation on your initial project, modify your floorplan and assignments as necessary to achieve a floorplan that meets your design requirements. You can set each partition to a netlist type post-fit, with a preservation level of placement or a netlist type with a lower level of preservation.
Creating Design Partitions for the Design Separation Flow

The Full incremental compilation option must be turned on if you use the Design Separation flow. Design partitions in the incremental compilation flow are synthesized and placed independently with minimal dependencies on the rest of the design. The functional independence of each design partition in the incremental compilation flow makes this a natural fit for the Design Separation flow.

Each secured region floorplan assignment uses a single design partition in the incremental compilation flow to identify the functional elements that belong to a secured region. Design partition assignments are made along entity boundaries in the hierarchy of your RTL design. Because only a single design partition may be used in a secured region, you must plan your design entities such that logic that requires physical isolation from the rest of the design is packed into a single design entity. Create wrapper files where necessary to reorganize your hierarchy, so that all your secured regions are contained within a single entity or module in your RTL.

When you use the Design Separation flow, most of the rules and guidelines for creating design partitions are the same as the incremental compilation flow without the Design Separation flow enabled. You can use the following features in the Quartus II software to help you create partition assignments:

- **Incremental Compilation Advisor**—Ensures your design follows Altera® recommendations for creating partitions and floorplan assignments.
- **Design Partition Planner**—A visual tool that allows you to explore your design hierarchy as well as view the connectivity between different entities within your design.
- **Chip Planner**—A graphical tool that presents a physical map of all the resources available on the device. After you run the Fitter, you can use the **Partition Display** task to get a visual indication of the size and location of the partition assignments on the device floorplan.

The design guidelines for creating design partitions when using incremental compilation also apply when the Design Separation flow is enabled. The following are the important considerations:

- Register the inputs and outputs of a design partition to avoid cross-boundary logic optimizations and to maintain timing performance along the signal path.
- Minimize the number of I/O paths that cross between partition boundaries to keep logic paths within a single partition for optimization. This makes partitions more independent for both logic and placement optimization.
- Avoid logic that requires cross-boundary logic optimizations.

For a detailed discussion about design guidelines to create partitions, refer to the *Best Practices for Incremental Compilation Partitions and Floorplan Assignments* chapter in volume 1 of the *Quartus II Handbook*.

When you design with the Design Separation flow, there are some restrictions and special considerations that differ from the normal incremental compilation flow. These are discussed in “Merging of PLL Resources” and “Child Partitions”.
Merging of PLL Resources

In the regular Quartus II incremental compilation flow, the Fitter is able to use the same PLL resource on the device when multiple partitions instantiate a PLL with the same parameters. This resource merge happens even if it requires optimization across design partitions. When the Design Separation flow is enabled, PLL merging across design partitions is disabled. This helps maintain the physical separation between design partitions. Merging is disabled regardless of the security levels of the region, even if LogicLock regions in a Cyclone III LS design contain no security attributes. For partitions that require shared PLL resources, the PLL must be instantiated outside of the partitions.

Child Partitions

In the Design Separation flow, each secured region contain a single partition. Child partitions are design partitions created from a sub-entity of an existing design partition. Child partitions are not allowed in a secured region, because this creates more than one partition within a secured region.

Creating a Design Floorplan with Secured Regions

The Quartus II software uses LogicLock location assignments to map logic in your design hierarchy to physical resources on the device. The Chip Planner provides a visual floorplan of the entire device and allows you to move and resize your LogicLock location constraints on the floorplan of the device. The Design Separation flow adds an additional constraint (a security attribute) to each LogicLock region for further constrain routing to achieve physical isolation between LogicLock regions, which turns the LogicLock region into a secured region. Signals that require connectivity between two secured regions or between a secured region and unsecured logic are assigned to a special LogicLock region known as a Security Routing Interface region. This LogicLock region is a controlled region and limits the routing of the contained signals to only the destination this region abuts.

To create fault isolation between secured regions, the Design Separation flow selectively shuts off routing around the periphery of a secured region. Because signal connectivity at the boundary of the secured region is unused, any faults that occur within the secured region are prevented from adversely affecting neighboring regions. This is because no physical connection exists to propagate the fault outside of the region.

Cyclone III LS devices use a MultiTrack interconnect architecture consisting of row and column interconnects that span fixed distances to achieve signal connectivity between LABs. In the horizontal direction, row interconnects use wire resources that span 1 LAB, 4 LABs, and 24 LABs. These row-routing resources are Direct Link interconnects, R4 interconnects and R24 interconnects, respectively. In the vertical direction, routing resources span distances of 1 LAB, 4 LABs, and 16 LABs. These column routing resources are Register Chain Interconnects, C4 interconnects and C16 interconnects, respectively. When the Design Separation flow is enabled for a LogicLock region, routing wires (C4, C16, R4, and R16) that span outside the border of a boundary are turned off. Each secured region uses an unused boundary (or a fence) of LBUs to guardband against the faults from wire resources spanning a length of 1 LAB (direct link and register chain routing resources) from affecting a neighboring region.
The rules and guidelines for floorplanning generally coincide with those in a compilation flow when the Design Separation flow is not enabled. However, there are some special considerations for the relative placement of secured regions in your design floorplan. Because each secured region is a keep-out region for routing resources from other LogicLock regions, ensure that a routing path exists between secured regions that have valid communication interfaces. Furthermore, the routing path (encapsulated in a Security Routing Interface) must be simple enough for your timing requirements to be met. The routing channel shouldn’t follow a circuitous path.

A fitter generated floorplan is not possible while a security attribute is applied to a LogicLock region; that is, the size attribute cannot be Auto, and the state attribute cannot be Floating for a secure LogicLock region.

You can leverage a fitter-generated floorplan, created without security attributes, as a starting point to create a final floorplan for the Design Separation flow.

To use a fitter-generated floorplan as an initial floorplan, apply Reserved attributes to LogicLock regions that must be physically isolated from the rest of the design. A Fitter-generated floorplan with Reserved attributes generates non-overlapping LogicLock regions. You can modify the initial floorplan by adjusting the relative placement for each of the secured regions, taking into account the connectivity requirements of each region.

Subsequent sections further detail the rules and guidelines for floorplanning that are specific to the Design Separation flow.

For a detailed discussion about using the settings and options of the Chip Planner feature, refer to the Analyzing and Optimizing the Design Floorplan chapter in volume II of the Quartus II Handbook.

Security Attributes

If you have a Design Separation flow, the Security Attributes column is added in the LogicLock Regions window and a Security Tab is added to the LogicLock Regions Properties dialog box. The security attribute applies a constraint to a LogicLock region, making a region either a secured region or a routing interface from where signals enter or exit a secured region. The LogicLock Regions window and the LogicLock Regions Properties dialog box are shown in Figure 3 and Figure 4, respectively, with security features added for the Design Separation flow.

Figure 3. Security Attribute Column Enabled When Design Separation Flow Enabled
Creating a Design Floorplan with Secured Regions

Figure 4. Security Tab Enabled When Design Separation Flow Enabled

Table 1 lists a summary for the Security attributes available for a licensed Design Separation flow.

Table 1. Security Attributes for LogicLock Regions (Part 1 of 2)

<table>
<thead>
<tr>
<th>Security Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unsecured</td>
<td>Disables constraint for physical isolation.</td>
</tr>
<tr>
<td>C1</td>
<td>Creates a secured region. Physically isolates the LogicLock region by restricting routing resources from leaving the region. Creates a border of unused LABS with a width of one, around the LogicLock region. Applying this attribute to a LogicLock region sets the global assignment LL_REGION_SECURITY_LEVEL C1 to the LogicLock region.</td>
</tr>
</tbody>
</table>
Secured Regions

When you apply a secured region attribute (C1 or C2) to an existing LogicLock region, the LogicLock region must have a fixed size with a locked origin. Each secured region must have a minimum size of 8 LABs in both the horizontal and vertical dimensions. A region smaller than 8 × 8 LABs may be non-routable when using the Design Separation flow.

Child regions are not allowed when creating a secured region. In the non-secured compilation flow, child regions are useful primarily for the following two reasons:

- To ensure logic in a child partition is physically contained inside the LogicLock region of the parent partition
- To make non-rectangular LogicLock regions by using the child region as a keep-out region for logic contained within the parent region (using the limited attribute

The first case is not necessary, because a secured region only contains a single partition. In the second case, a simpler flow is available for the Design Separation flow to create non-rectangular LogicLock regions.

For more information about making non-rectangular LogicLock regions with the Design Separation flow, refer to “Creating Non-Rectangular Regions” on page 14.

A secured region must contain all physical device resources it requires to complete compilation. I/O pads that are members of a secured region must be contained within the boundaries of the secured region that sources or sinks it. That is, a secured region must overlap the I/O pads that are members of the region. If the logic in the secured region instantiates a PLL or a clock buffer, those physical device resources must also be overlapped by the region.

Security Routing Interface

A LogicLock region with the security attribute of Security Routing Interface creates a routing channel for signals to and from a secured region. No logic may be placed in a Security Routing Interface. Each Security Routing Interface can connect two secured regions, or a secured region one or more unsecured regions. If you are connecting two secured regions, a fence is automatically placed around the interface region. Each signal entering or leaving a secured region is assigned to a Security Routing Interface through the Security tab in the LogicLock Regions Properties dialog box.
For information about assigning signals to a Security Routing Interface, refer to “Security Assignments to Signals” on page 20.

For information about the number of signals that can be contained in a Security Routing Interface, refer to “Routing Restrictions” on page 28.

**Design Separation Flow Location Assignments in the Chip Planner**

The Chip Planner allows you to visually modify the size and location of LogicLock regions. This section describes the attributes of LogicLock regions within the context of the Design Separation flow.

When the Design Separation flow is enabled, the Chip Planner indicates the fencing region around each secured region by shading the fencing region brown. Security Routing Interfaces are shaded green. Illegal placements that violate secured region boundaries are highlighted in red at the location where the violation occurs. Figure 5 shows the LogicLock regions with security attributes in the Chip Planner.
Fencing Regions

The Quartus II software automatically adds a fencing region, which is a border of unused LABs, when you apply a C1 or C2 attribute to a LogicLock region. No logic may be placed into a fence region. The Fitter does not use any routing wires that go outside the fence boundary that are within a secured region. Because Direct Drive and Carry Chains can be used at the edge of a secured region, the fencing region prevents signals being driven on length 1 wires (in the horizontal and vertical directions) from leaving the secured region.

Notes to Figure 5:
(1) Floorplan Editing Task.
(2) Unused fence around a secured region.
(3) Security violation, created from a LogicLock region being placed within a fence boundary of a secured region.
(4) Security Routing Interface Region connecting two secured regions.
(5) Security Routing Interface Region connecting secured region and unsecure logic.
The fence size around a secured region is generally 1 LAB in the horizontal and 1 LAB in the vertical direction. There are two regions that have a special consideration for the fencing region:

- **Vertical I/O regions**
- **Areas around the configuration engine**

I/O banks along the top and bottom of the chip use only vertical routing wires to and from the I/O Elements (IOEs). The heavy use of C4 wires from IOEs creates a four LAB fence between the vertical I/O banks and a secured region. Secured regions requiring a connection to I/O in the top or bottom banks of the device optimally use resources if you add the I/O signals as members of the secured region and overlap the corresponding I/O pads the floorplan. Figure 6 compares the fence sizes in two secured regions: one is five LABs away from the bottom of the device and the other four LABs away from the bottom of the device.

**Figure 6.** Vertical Fence Size Near I/O Banks

The configuration engine is a hard IP that handles the configuration of the device. In addition to configuration services, it routes the control signals for the CRC detection circuit and the internal oscillator into the core logic on the device. If you have the Design Separation flow enabled, a one LAB fence is automatically added around the configuration engine whenever a secured region occupies the same LAB column as the configuration engine. The configuration engine is a region notched out of the left side of the device and halfway down the device.
All control signals to and from the configuration engine route from the right edge of the configuration engine. If you use any instantiated WYSIWYG that use any control signals to and from the configuration engine, the signals must either interface to unsecured logic or they must interface with a secured region through a Security Routing Interface.

If your design requires signals to and from the configuration engine, a secured region that directly abuts the configuration engine signal interface (along the right side of the configuration engine) causes a fit error.

**Figure 7** shows the configuration engine with a fencing region in the floorplan for a Cyclone III LS device.

**Figure 7.** Control Block in Chip Planner

Fencing regions between two secured regions allowed to overlap. That is, two adjacent secured regions can be separated by a one LAB fence. The Chip Planner provides a warning in the form of a security violation if any LogicLock gets placed within the boundary of a secured region. Security violations are highlighted in red and the tooltip of a secured region indicates the locations of all security violations. You may receive an error if you try to compile a design with a security violation. **Figure 8** shows two regions with overlapping fences and a security violation from an unsecured region.
Creating Non-Rectangular Regions

Non-rectangular regions may be created by creating multiple rectangular regions and merging them. Create non-rectangular regions in the Chip Planner by performing the following steps:

1. Use the Create LogicLock Region button to construct the desired region shape using multiple rectangular LogicLock regions.
2. While holding down the ctrl-key, select all of the rectangular regions that you’ve created.
3. On the right-click menu, point to LogicLock Region and click Merge.
4. Enter the name of the newly created LogicLock region in the Merge Regions dialog box.

Use non-rectangular regions sparingly. For the Quartus II software version 9.0, support for non-rectangular regions is preliminary, because they can cause poor route times and have an adverse affect on performance when used with the Design Separation flow.

If a secured non-rectangular region contains a sub-region that is less than 8 × 8 LABs, it increases the chances of a non-routable situation occurring. Subregions that deterministically require the use of certain routing resources may not fit successfully if violation of the secured region is required. As a general guideline, each sub-region should be 8 × 8 LABs or larger to ensure that routing resources of length four are

---

Notes to Figure 8:
(1) One Lab Fence can be shared between regions.
(2) Tooltips list all locations of a security violation.
readily available. In Figure 9, each of the sub regions of region 2 (labeled A, B, C, and D) are less than 8 × 8 LABS in dimension. These regions can potentially run into a no-fit situation. Depending on the placement and connectivity of LABs, certain routes may be difficult to achieve. For example, the Fitter would not be able to route a connection from LAB 1 to LAB 2 in region A directly. While another path maybe possible, a series of hops that do no leave the LogicLock region may not be available and may not satisfy the timing requirements of the route.

**Figure 9. Non-Rectangular LogicLock Regions** *(Note 1)*

---

**Guidelines for the Relative Placement of Secure LogicLock Regions**

Because each secured region is a keep-out region for placement and routing of any logic that is not a member of the secured region, there may be a few guidelines to consider as you lay out your floorplan. Placement that does not account for the connectivity requirements between LogicLock regions may cause poor performance or a non-routable design. The guidelines for floorplanning when using the Design Separation flow include:

- Creating a complete floorplan, including location assignments for unsecure logic.
- There should be a non-circuitous route between secured regions requiring a routing region. As a general guideline, routing regions between secured regions should be rectangular.

---

**Note to Figure 9:**

(1) Each subregion should be at least 8 × 8 LABS in dimension or a non-routable situation may occur.
Security routing interface between secured regions cannot intersect with other routing regions; secured regions and their routing edges must be able to fit on a single plane. A secured region must overlap any physical resources (such as I/Os, PLLs, CLKCTRL) that are instantiated by the design partition contained in the secured region.

If possible, abut the secured region to the edge of the device.

Create A Complete Floorplan

Allocate a region for each partition in your design. If you have a large secured region that divides the device into multiple disjointed regions, and you have unsecured logic that is not floorplanned, the design may be un-routable.

Figure 10 shows a hypothetical scenario where this can be problematic. If there is an unsecure partition and it does not contain any location assignment, the placement algorithms may use any unallocated space on the device to make logic assignments. In the case of the floorplan in Figure 10, the source and sink registers do not have a valid path through the device, because all routing channels are occupied by Secured Region 1 and Secured Region 2.

Figure 10. Non-Routable Placement Scenario

If a complete floorplan is not possible for all partitions in your design, you can use empty LogicLock regions with the Reserved attribute to keep the Fitter from placing any logic in a region that can potentially cause a no-fit. For the example provided in Figure 10, an empty region can be placed in the upper left hand corner of the device to prevent any logic that has not been floorplanned from being placed there. This is shown in Figure 11.
Routability Between Regions

The Quartus II software cannot create auto-generated location constraints of any region with a security attribute applied to it. If you use a Fitter-generated placement as a starting point for a floorplan for the Design Separation flow, an optimal floorplan in a design without separation may not work in the same design with security attributes applied to a region. In a floorplan without secured regions, only the placement of logic is restricted. Any routing resources on the device are available for the Fitter to use, and a route may go through a region. Secured regions reserve all routing resources within the LogicLock boundary to the design partition contained in the region.

Having a circuitous route between two regions degrades performance and may cause a non-routable design. Modify any regions that have signal connectivity and must route around a secured region to achieve a connection. **Figure 12** shows a floorplan that does not contain disjointed parts. However, the source region must route around a secured region to connect to the sink region.

**Figure 11.** Empty Reserved Region Preventing Fitter From Placing Logic
Figure 12. Relative Placement of Regions Containing a Circuitous Path \textit{(Note 1)}

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{ figura12.png}
\caption{Initial placement from Fitter may cause routability issues in the same design with secured regions.}
\end{figure}

**Planarity**

A fence is automatically created around a Security Routing Interface connecting two secured regions. No other routing resources may pass through a Security Routing Interface connecting two secured regions. This implies that if you model all secured regions as nodes in a routing graph and all Security Routing Interfaces as the edges, all nodes and their edges must fit on a planar graph (that is, none of the edges can intersect). If you have five or more secured regions on the device, and each secured region contains signals that fan-out to multiple secured regions, a planar floorplan may not be possible. Figure 13 shows the routing graph with five nodes. A complete graph having each pair of distinct vertices connected by an edge is not possible without having any of the edges cross. If the topology of your floorplan contains such a non-routable arrangement, your design hierarchy must be rearranged to collapse related design partitions into a single design partition.

Figure 13. Non-Planar Routing Graph: Connection BD Not Possible

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{ figura13.png}
\caption{Complete graph having each pair of distinct vertices connected by an edge is not possible without having any of the edges cross.}  
\end{figure}
If your secured regions and Security Routing Interfaces can be modeled as a planar graph, but have a high degree of connectivity between the components, you may have to rearrange the shape, size, or location of the secured regions to generate a routable floorplan. For instance, the hypothetical floorplan shown in Figure 14 does not have a valid routing path BD (between region B and region D). The modified floorplan of Figure 15 shows how all of the required connections can be achieved on a planar surface.

**Figure 14.** Non-Routable Connection BD

You can use the Design Partition Planner for a visual indication of the connectivity between design partitions. This tool helps you determine if the secured regions in the design can be arranged on a planar floorplan. The Design Partition Planner is shown in Figure 16.
Physical Resources

All physical resources that are required by the secured region must be contained inside the boundary of the secured region. This includes the following:

- I/O pins connected to the secured region
- Primitives that have been instantiated within the secured region, such as PLLs, clock configuration engines

Security Assignments to Signals

Each signal that enters or exits in a secured region must be explicitly assigned to a Security Routing Interface. In addition to being assigned to a Security Routing Interface, each signal contains a Security Level attribute. The security level for each signal is automatically assigned a default value and matches the secured region that sources the signal. The possible values of the security level of a signal are Auto, Unsecured, C1, and C2. An assignment of Auto sets the default security level for the signal.

A signal with a security attribute may connect to a region with the same security level or a security level that is higher. For example, a signal with a security level of Unsecured can drive logic in a region set to Unsecured, C1, or C2 and a signal with a security level of C1 can drive logic in a region set to C1 or C2. A signal originating from a secured region may not drive logic in a region with a lower security level. If you have a signal from a higher security level that must drive logic in a lower security level, have the Fitter honor the connection by explicitly lowering the security level of the signal.

Each Security Routing Interface connects two regions at most. If a signal fans out to multiple regions, assign the signal to multiple Security Routing Interface regions; one interface region per destination.
Assignments of signals to Security Routing Interfaces and the security level of signals can be modified using the Security tab in the LogicLock Region Properties dialog box, shown in Figure 17.

To assign a signal to a Security Routing Interface, perform the following steps:

1. On the Security Tab of the LogicLock Regions Properties dialog box, double-click on the signal name in the Signals list. This list is pre-populated with the names of signals entering and exiting the secured region after you have performed analysis and synthesis and a Partition Merge. Double-clicking on a signal name invokes the Security Assignments dialog box shown in Figure 18.

Notes to Figure 17:

1. Signals list is populated automatically after analysis and elaboration.
2. Columns in the signal list describe the Security Level, the Security Routing Interface the signal is assigned to, and whether the signal is an output or input to the region.
2. Alternatively, you can select multiple names in the Signal list by holding down the ctrl key and clicking the Edit button.

3. If necessary, lower the security level of the signal by specifying the Security level option.

3. Select the Security Routing Interface for signal(s) assignment. Signals that fan-out or fan-in to multiple regions can be assigned to multiple Security Routing Interfaces.

**Signal Names**

The list of signals entering and leaving a secured region are signal names from the post-map netlist. Signal outputs from a secured region are taken from the output port name, as specified in the top-level RTL entity contained in the secured region. Signal inputs to a secured region are taken from the name of the register or LUT that drives the port. In the secure compilation flow, all output port names are preserved through the compilation process. The post-map region output signals listed in the signal list coincide with the signal name in the post-fit netlist. However, combinational signal names from unsecure or unpartitioned logic that feed a secured region may change through the compilation process. Many of the RTL signals are optimized during the process of synthesis and place-and-route. Frequently, RTL signal names may not appear in the post-fit netlist after optimization. For example, the compilation process can add tildes (“~”) to nets that are fanning out from a node, making it difficult to decipher which signal nets they actually represent. Registered signals tend to maintain the same signal names in both the post-map and post-fit netlists. When possible, use registered signals as inputs into a secured region, and register the output signals from a secure partition.
Global Signals

Global signals are low-skew routing lines that drive throughout the device. Global signals do not require an interface region to drive into a secured region. In the Cyclone III LS device family, there are 20 global routing resources for use with high fan-out signals, such as clocks or control signals. Each global signal is accessed by a clock-configuration engine, located on the periphery boundary of the device. Each clock-configuration engine can be driven directly by external clock pins, PLL outputs, or a signal generated from internal logic.

For more information about the clock networks in Cyclone III LS devices, refer to the Clock Networks and PLLs in Cyclone III LS Devices chapter in volume 1 of the Cyclone III LS Device Handbook.

In a compilation flow without the Design Separation flow enabled, signals with a high fan-out (such as clock pins and control signals) are automatically promoted to use global clock resources. When the Design Separation flow is enabled, automatic global promotion is not turned on. Signals with high fan-out requiring global routing resources must be automatically promoted in order to drive a clock-configuration engine.

Signals cannot be promoted onto a global routing resource through a global signal assignment from within a secured region. The Fitter only allows a clock promotion assignment to a signal if the signal is in an unsecured region. If you have a signal inside of a secured region that must use a global routing resource, you must first route the signal outside of the secured region before applying a global promotion assignment. The signal must be assigned to a Security Routing Interface and the security level of the signal must be lowered.

For a global promotion assignment to be honored, there must be an available clock configuration engine that is not overlapped by a secured region, and an available routing path to the clock-configuration engine. The clock-configuration engines are located on each side of the device, along the horizontal and vertical axis that run through the center of the device. There are 5 clock configuration engines on each side of the device. Figure 19 and Figure 20 show the location of the clock-configuration engine and the PLLs for a 3CLS70 device in the Chip Planner floorplan.
Figure 19. PLL and Clock-Control Block Location on a EPC3SL70 Device
Figure 20. Location of PLLs and Clock Control Buffers - Cyclone III LS Device Family

Notes to Figure 20:

1. There are five clock configuration engines on each side.
2. Only one of the corner CDCLK pins in each corner can feed the clock configuration engines at a time. You can use the other CDCLK pins as general purpose I/O pins.
3. Remote clocks cannot be used to feed the PLLs.
4. Dedicated clock paths can feed into this PLL. However, these are not fully-compensated paths.
PLLs and clock-configuration engines can be manually instantiated in the design partition of a secured region using the ALTPLL and ALTCLKCTRL megafunctions, respectively. Instantiation of the ALTCLKCTRL megafunction within a secure partition forces the global promotion of the signal driving the clock-control buffer. To generate a valid placement when you instantiate PLLs or clock-configuration engines, the secured region containing the physical resource must overlap a free PLL, a free clock-configuration engine, or both.

There are certain restrictions when you instantiate a PLL within a secured region. Secured regions with a PLL that are fed by an external clock pin must contain the PLL and a valid clock pin that can drive the PLL. Each PLL has a set dedicated clock-configuration engine that it can access, located to the right (clockwise) of the PLL in the device floorplan. The PLL outputs must drive the clock-configuration engines through an unsecured region. For the PLL to access the clock configuration engine, the security level of the PLL output must be assigned a Security Routing Interface and its security level must be lowered to unsecure.

The clock-configuration engine associated with the PLL must not be covered by any secure LogicLock region. There are two sets of dedicated clock pins that can drive a PLL input. The pads for the clock input pins are co-located with the clock control buffers. If you use the clock input pin that is co-located with the clock-control buffer associated with the PLL, the clock pin cannot be added as a member of the secured region. Instead, you must assign the clock pin to a Security Routing Interface that is connected with the secured region.

Assignment of I/O Pins

There are three rules, in addition to the regular pin assignment rules, that you must consider when the Design Separation Flow is enabled:

- I/O pins that are connected to a secured region must be assigned as a member of that secured region or assigned to a Security Routing Interface region that abuts the secured region.
- Secured regions with I/O pins as members cannot share the I/O banks with any other region.
- I/O pins associated with different secured regions or with different security levels may not use adjacent pins.

I/O pins may be added as members of a secured region, typically when directly connected to the secured region (in the LogicLock Regions Properties dialog box, under the General tab, click Add node). If an I/O pin is a member of a secured region, the I/O pad must be physically contained within the region. The secured region must overlap the I/O resource.

If you do not add the I/O pin as a member of the secured region, you must assign the I/O signal to a Security Routing Interface that abuts the secured region. This Security Routing Interface must connect the secured region to the root region or another unsecured region. Explicitly lower the security level of any output signals from the secured region that are connected to I/O pins.

I/O signals that are routed out to an unsecured logic are no longer guaranteed to be physically isolated from other signals in the design.
Each I/O pin is adjacent to eight other pins: four along the horizontal and vertical axes, and four in the two diagonal axes (see Figure 21).

**Figure 21.** Pin Adjacency

Pins from different I/O banks may not share an adjacent I/O pin if one of the I/O banks contains pins that are members of a secured region. Pins that are adjacent to a signal in a secured region (a secured signal), which belong to a different I/O bank than the secured signal, should be assigned to ground in the Quartus II software. For example, in Figure 21, pin E4 is assigned a signal from a secured region, and I/O banks 1 and 8 belong to different LogicLock regions. Pins D4, D5, and E5 are assigned to GND to ensure that no signal adjacencies exist between the I/O banks.

As a general rule, all unused I/O pins should be assigned to GND in the Quartus II software and assigned to a ground plane on the PCB. By default, the Quartus II software assigns unused pins to ground. You can configure this option in the **Unused Pins** tab of the **Device and Pin Options** page of the **Settings** dialog box.

For information about the pinouts and pin adjacencies for the Cyclone III LS device family, refer to the **Cyclone III LS Device Pin-Out** tables. For additional information and guidance about I/O assignments, refer to the **Cyclone III LS Device Family Pin Connection Guidelines** for the Cyclone III LS device family and the **I/O Management** chapter in volume 2 of the **Quartus II Handbook**.
Routing Restrictions

Column and Row interconnect routing resources on the Cyclone III LS device family are staggered, with a group of routing elements starting at each LAB location. Each routing element is driven by the LAB location where the wire starts and can reach any LAB destination along the length of the routing element. Figure 22 shows a set of staggered R4 interconnects.

**Figure 22.** Staggered R4 Interconnects

The Fitter disables routing wires near the edge of a secured region, where routing is confined within the region. Figure 23 shows the used routing elements on a design with secured regions using the Chip Planner.
Most long wire connections are not used in secured regions. However, all wire bundles terminate along the edges of the device. Hence, there are a number of stubbed wires driving toward each edge of the device. If a secured region abuts the device boundary, you can often get an increase in routability, because all of the routing interconnects that start inside the region and drive toward the edge of the device can be used.

Notes to Figure 23:
(1) No routing resources reach outside of LL region boundaries, except for global routing signals and signals through interface regions.
(2) Chip Planner view, configured using Layer settings and using the background color map I/O banks, with only Global Routing and Used Resources options selected.
I/O pads along the top and bottom of the device can only use column interconnects to drive into the device fabric. The shortest routing element from the I/O to core logic is a C4 routing wire. I/O pads on the left and right sides of the device can use both C4 and R4 routing elements to reach their LAB destinations. Because column I/Os are restricted to using C4 interconnects going into the device, a 4 LAB fence is created around secured regions when the boundary of the secured region is within 4 LABs of the top and bottom I/O pads.

Secured regions should be sized at a minimum of $8 \times 8$ LABs. If a region is smaller than $8 \times 8$ LABs, a connection between two LABs that violates the secured region boundary may occur. For example, in Figure 24 any elements along the middle axis of the $7 \times 7$ LAB array cannot use any C4 or R4 routing elements, because a C4 routing element would reach outside the secured region.

Figure 24. LogicLock Region Requires a Minimum of 8x8 LABs

Number of Signals in Routing Interfaces

For the Cyclone III LS device family, every LAB has 68 routing elements (R4) driving horizontally in each direction and 48 routing elements (C4) driving vertically in each direction. The number of connections that can be directly driven by an LAB is 17 in the horizontal direction and 12 in the vertical direction. To guarantee routability, Altera recommends that you have a routing interface height of at least 1 LAB for every 17 signals routing either left or right, and a routing interface width of 1 LAB for every 12 signals routing either up or down. Figure 25, Table 2, and Table 3 illustrate this concept. Figure 25 shows three secured regions with two security routing regions, one routing signals horizontally and the other routing signals vertically. Table 2 and Table 3 list the maximum and recommended number of signals crossing each security region.
As a general guideline, keep the Security Routing Interface channel length between the two connecting secured regions as short as possible and the width of the channel as wide as possible. Having a short and wide channel increases the number of routing resources available between two secured regions.

**Table 2.** Maximum Number of Signals Assigned in an Interface Region

<table>
<thead>
<tr>
<th>To</th>
<th>From</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A</td>
</tr>
<tr>
<td>A</td>
<td>—</td>
</tr>
<tr>
<td>B</td>
<td>$68 \times H_{AB}$</td>
</tr>
<tr>
<td>C</td>
<td>—</td>
</tr>
</tbody>
</table>

**Table 3.** Recommended Number of Signals to Ensure Routability

<table>
<thead>
<tr>
<th>To</th>
<th>From</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A</td>
</tr>
<tr>
<td>A</td>
<td>—</td>
</tr>
<tr>
<td>B</td>
<td>$17 \times H_{AB}$</td>
</tr>
<tr>
<td>C</td>
<td>—</td>
</tr>
</tbody>
</table>

Notes to Figure 25:
1. $H_{AB}$ is the smaller height in the region and the height of the routing interface.
2. Minimum $W_{AB}$ is 1. Changing $W_{AB}$ does not affect Table 3. $W_{BC}$ is the smaller width in the region and the width of the routing interface.
3. Minimum $H_{BC}$ is 1. Changing $H_{BC}$ does not affect Table 3.
You can use the **Routing Congestion** task in the Chip Planner to get a visual indication of the routing utilization between secured regions. Routing resources are filtered by type. Utilization of each routing resource type is highlighted on a color gradient over the range that you specify. This tool can be useful to help you adjust region sizes and Security Routing Interface channel widths to help you achieve an optimal floorplan. A design with the **Routing Congestion** task in the Chip Planner and R24 routing utilization is shown in Figure 26.

**Figure 26.** Routing Congestion
Application Example: Modify a Fitter Generated Floorplan for the Design Separation Flow

In this example, the design contains five partitions that must be packed into secured regions. Figure 27 shows a block diagram of the design, the entities of the design, and the connectivity between the five secure partitions.

Figure 27. Connectivity between Five Secure Partitions

The following procedure outlines a recommended design flow for creating a floorplan for this design.

1. Create a LogicLock region for each partition that must be packed into a secured region. Set each with the attributes of **Auto**, **Floating**, **On**, and **Unsecured** for the **Size**, **State**, **Reserved**, and **Security Attributes** columns, respectively. Running an initial placement with these settings generates non-overlapping LogicLock regions that can be used as an initial floorplan.
2. On the Processing menu, point to **Start** and click **Start Early Timing Estimate** to run an initial place-and-route operation. This gives you an idea of the size of each region, as well as a general idea of how the LogicLock regions must be placed relative to other LogicLock regions to achieve timing closure. The floorplan generated by the early timing estimate is shown in **Figure 28**.

**Figure 28.** Initial Fitter Generated Floorplan

3. Set the size and origin of each LogicLock region to the previous fitting results. In the LogicLock Regions window, select the LogicLock regions, right-click and select **Set Size and Origin to Previous Fitter Results**.

4. Use the Design Partition Planner to view the connectivity between the different regions. You can experiment with different placement positions of the blocks relative to one another by dragging and dropping each of the design partitions. The wire bundles between design partitions help you to determine a placement that has non-overlapping routing channels.
You must also consider the connectivity to the I/O banks when arranging your floorplan. You can toggle the display of the connections between the partitions and the I/O banks within the Design Partition Planner to help you properly allocate I/O resources, as well as avoid conflicts between I/O connections and inter-partition signals. To display routing between partitions and the I/O banks, turn on Display connections to I/O banks in the Bundle Configuration dialog box.

**Figure 29.** Design Partition Planner
5. Set each LogicLock region to the desired security attribute. In the Chip Planner, adjust the size and placement of each LogicLock region using the relative placement you created with the Design Partition Planner. Note the following when modifying the floorplan:

a. Create a complete floorplan. If there is unsecured logic that is non-contiguous due to the placement of a secure region, use an empty reserved LogicLock region to prevent a non-routable placement.

b. Each secured region must be a minimum of 8 x 8 LABs

c. Each region that has I/O pins added as members of the LogicLock region should overlap the I/O bank that it is connected to. You can use the I/O bank background color map to visualize the boundaries between the I/O banks (Figure 30).

d. All global resources (such as clock pins, and PLLs) that are required by unsecured logic must not be covered by a secured region.

Figure 30. I/O Banks Layers Setting for Viewing Connectivity of LogicLock Regions to I/O Banks

6. Create Security Routing Interfaces between each of the secured regions. Assign all signals entering or exiting a region to a Security Routing Interface.
7. The final floorplan result for this example is shown in Figure 31.

**Figure 31.** Final Floorplan

---

**Report Panels**

After the Fitter successfully places and routes a design with secured regions, the Quartus II software generates a Security Report. Use the Security Report to review the secured regions, their associated routing interfaces, all inputs and outputs from each secured region, and the I/O bank usage for each secured region. The security reports are located in the Fitter folder of the Compilation Reports (Figure 32).

**Figure 32.** Design Separation Flow Security Report
Secured LogicLock Region Summary

This report provides a summary of the all secured regions in your design. Table 4 provides a description of each of the columns in the Secured LogicLock Region Summary Compilation Report.

Table 4. Secured Logic Region Compilation Report

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Secured Logic Lock Region</td>
<td>Lists the all secured logic lock regions in the design</td>
</tr>
<tr>
<td>Security Attribute</td>
<td>Security attribute (unsecured, C1, C2, or Security Routing Interface) of the region</td>
</tr>
<tr>
<td>Partition Assigned</td>
<td>Lists the design partition assigned to the secured region</td>
</tr>
<tr>
<td>Number of Input Signals</td>
<td>Number of inputs and fan-outs into a region. The input counts the number of unique drivers that feed a secured region. Fan-out counts the total number unique destinations being fed by the input signals into the secure region. Refer to Figure 33 for an illustration of input signals and fan-outs to a region.</td>
</tr>
<tr>
<td>Number of Output Signals</td>
<td>Number of outputs and fan-outs from a region. The output counts the number of unique drivers sourcing a signal from the secured region. Fan-out counts the total number of unique destinations fed by the output signal.</td>
</tr>
</tbody>
</table>

Figure 33. Input Signals and Fan-Outs to a Region

<table>
<thead>
<tr>
<th>Secured Region A</th>
<th>Secured Region B</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Diagram" /></td>
<td></td>
</tr>
</tbody>
</table>

Secured Region A - Number of Output Signals (Total Fanout) : 1
Secured Region B - Number of Input Signals (Total Fanout) : 1

Security Routing Interfaces

This report summarizes the Security Routing Interfaces. Table 5 provides a description for each of the columns in the Security Routing Interfaces Compilation Report.

Table 5. Security Routing Interface Compilation Report (Part 1 of 2)

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Name</td>
<td>Lists the all the Security Routing Interfaces in the design</td>
</tr>
<tr>
<td>Abutting Region A</td>
<td>First region that the Security Routing Interface abuts (touches the blue border of the secured region)</td>
</tr>
<tr>
<td>Abutting Region B</td>
<td>Second region that the Security Routing Interface abuts (touches the blue border of the secured region)</td>
</tr>
</tbody>
</table>
Secured LogicLock Region Inputs and Outputs

This set of reports provide a detailed list of every signal that enters or exits a secured region. There is one report per secured region.

Security I/O Bank Usage

This report displays the secured LogicLock region associated with each I/O bank, counts the number of pins within each region, and counts the number of pins in use. Table 6 provides a description for each of the columns in the Secured LogicLock Region Inputs and Outputs report.

Table 6. Secured LogicLock Region Input and Output Compilation Report

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>I/O Bank</td>
<td>Lists all available I/O banks on the device.</td>
</tr>
<tr>
<td>Associated Region</td>
<td>An I/O bank becomes associated with a secured LogicLock region if any portion of the I/O bank is covered by the region. If no secured region covers an I/O bank, “Unsecured Logic” is shown, and all pins of the I/O bank are available for unsecured use.</td>
</tr>
<tr>
<td>Pin Locations Used / Pin Locations Covered by Region</td>
<td>Displays the ratio of pins with a signal assignment in the I/O bank to the number of possible I/O pin assignments.</td>
</tr>
</tbody>
</table>

Quartus Settings File Syntax

This section contains the syntax description for each of the Quartus Settings File (.qsf) assignments in the Design Separation flow.

LL_SECURITY_ROUTING_INTERFACE

This command changes a LogicLock region assignment to a Security Routing Interface.

Type: Boolean; (ON/OFF - Defaults to OFF)

Syntax:

```
set_global_assignment -name LL_SECURITY_ROUTING_INTERFACE <value> \ 
-section_id <section_identifier>
```
LL_REGION_SECURITY_LEVEL

This command identifies the security level of a LogicLock region.

Type: Enumeration—defaults to UNSECURED

- C1
- C2
- UNSECURED

Syntax:
set_global_assignment -name LL_REGION_SECURITY_LEVEL <value> \ 
-section_id <section_identifier>

LL_MEMBER_OF_SECURITY_ROUTING_INTERFACE

This command assigns a signal I/O from a secured region to a Security Routing Interface. Both value and section ID denotes the name of the routing interface region.

<to> specifies the name of the signal.

Type: String

Syntax:
set_instance_assignment -name \ 
LL_MEMBER_OF_SECURITY_ROUTING_INTERFACE <value> -to <to> \ 
-section_id <section_id>

LL_SIGNAL_SECURITY_LEVEL

This command sets the security level of a signal. The default value is the security level of the region that generates the signal. This assignment may only be used to lower a security level.

Type: Enumeration

- UNSECURED
- C1
- C2

Syntax:
set_instance_assignment -name LL_SIGNAL_SECURITY_LEVEL <value> \ 
-to <to> -section_id <section_id>
Table 7 lists the Quartus II compilation messages related to the Design Separation flow.

<table>
<thead>
<tr>
<th>Message and Description</th>
<th>Cause</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Message</strong>: “Can’t specify LogicLock region <code>&lt;region_name&gt;</code> as secured, because it is the root region”</td>
<td>You tried to specify the root region to be secured. This operation is not supported.</td>
<td>Consider creating a new LogicLock region and specify that as secured through the Security Level property.</td>
</tr>
<tr>
<td><strong>Message</strong>: “LogicLock region <code>&lt;region_name&gt;</code> cannot contain child regions and be secured at the same time”</td>
<td>A LogicLock region cannot be secured and contain child regions at the same time. If you receive this message while performing an operation through the GUI, the operation is ignored. If you currently have assignments that violate this rule, the LogicLock region’s Security Level property is ignored.</td>
<td>Make sure that no region is secured and contains child regions at the same time.</td>
</tr>
<tr>
<td><strong>Message</strong>: “LogicLock region <code>&lt;region_name&gt;</code> cannot be a child of another user-created region and be secured at the same time”</td>
<td>A LogicLock region cannot be secured and be a child region of another user-created region at the same time. If you receive this message while performing an operation through the GUI, the operation is ignored. If you currently have assignments that violate this rule, the LogicLock region’s Security Level property is ignored.</td>
<td>Make sure that no region is secured and is a child region of another user-created region at the same time.</td>
</tr>
<tr>
<td><strong>Message</strong>: “Can’t specify LogicLock region <code>&lt;region_name&gt;</code> as secured, because the current device is automatically selected”</td>
<td>When the current device is automatically selected, no LogicLock region can be secured. A secured region must have a fixed size and a locked origin, both of which are impossible when the device is set to Auto. If you receive this message while performing an operation through the GUI, the operation is ignored. If you currently have assignments that violate this rule, the LogicLock region’s Security Level property is ignored.</td>
<td>If you want to specify one or more LogicLock regions as secured, use a specific device instead of an automatically selected device.</td>
</tr>
<tr>
<td><strong>Message</strong>: “LogicLock region <code>&lt;region_name&gt;</code> cannot be Auto, Floating, Non-Reserved or Soft, because it is secured”</td>
<td>A LogicLock region cannot be secured and be Auto, Floating, Non-Reserved or Soft. If you receive this message while performing an operation through the GUI, the operation is ignored. If you currently have assignments that violate this rule, the LogicLock region is updated to Reserved and Non-Soft, and has a fixed size and a locked origin.</td>
<td>Make sure that no region is secured while it is also Auto, Floating, Non-Reserved or Soft.</td>
</tr>
<tr>
<td><strong>Message</strong>: “LogicLock region <code>&lt;region_name&gt;</code> cannot be Auto, Floating, Non-Reserved or Soft”</td>
<td>A LogicLock region cannot be secured and be Auto, Floating, Non-Reserved or Soft. If you receive this message while performing an operation through the GUI, the operation is ignored.</td>
<td>Make sure that no region is secured while it is also Auto, Floating, Non-Reserved or Soft.</td>
</tr>
<tr>
<td><strong>Message</strong>: “Can’t specify the security level of LogicLock region <code>&lt;region_name&gt;</code> to be <code>&lt;security_attribute&gt;</code>, either because you do not have the required license or the currently selected device family does not support specifying security level”</td>
<td>You attempted to set the security level of the specified LogicLock region to a value that is not supported by either the current Quartus II software license, or the currently selected device family, or both. If you get this message while performing an operation through the GUI, the operation has been ignored. If you have security assignments in the .qsf, the assignments are ignored.</td>
<td>Make sure that you have the required license and that the currently selected device family supports the feature, and then retry the operation.</td>
</tr>
</tbody>
</table>
Table 7. Design Separation Flow Messages (Part 2 of 10)

<table>
<thead>
<tr>
<th>Message and Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Message</strong>: “LogicLock region <code>&lt;region_name&gt;</code> cannot have security routing member, because it is not a Security Routing Interface”</td>
</tr>
<tr>
<td><strong>Cause</strong>: You attempted to add one or more security routing members to a LogicLock region that is not specified as a Security Routing Interface.</td>
</tr>
<tr>
<td><strong>Action</strong>: Set the specified region to be a Security Routing Interface, and then retry the operation.</td>
</tr>
<tr>
<td><strong>Message</strong>: “Can’t turn off the Security Routing Interface property of LogicLock region <code>&lt;region_name&gt;</code> because, it contains security routing members”</td>
</tr>
<tr>
<td><strong>Cause</strong>: You attempted to turn off the Security Routing Interface property of a LogicLock region that still contains routing members. You must clear the security routing members before performing this operation.</td>
</tr>
<tr>
<td><strong>Action</strong>: Consider removing the existing assignments to the specified region and retry the operation.</td>
</tr>
<tr>
<td><strong>Message</strong>: “LogicLock region <code>&lt;region_name&gt;</code> cannot be a Security Routing Interface, because it contains placement-controlling members”</td>
</tr>
<tr>
<td><strong>Cause</strong>: You attempted to set a LogicLock region that contains placement-controlling members to be a Security Routing Interface. A LogicLock region cannot contain both placement-controlling members and routing-controlling members at the same time.</td>
</tr>
<tr>
<td><strong>Action</strong>: Consider removing the all member assignments to the specified region and then retry the operation.</td>
</tr>
<tr>
<td><strong>Message</strong>: “LogicLock region <code>&lt;region_name&gt;</code> cannot be a Security Routing Interface, because it contains security level assignments to its output signals”</td>
</tr>
<tr>
<td><strong>Cause</strong>: You attempted to set a LogicLock region that contains signals with security level assignments to be a Security Routing Interface. A LogicLock region cannot contain both LL_SIGNAL_SECURITY_LEVEL assignments and routing-controlling members (that is, LL_MEMBER_OF_SECURITY_ROUTING_INTERFACE) at the same time.</td>
</tr>
<tr>
<td><strong>Action</strong>: Consider removing the existing LL_SIGNAL_SECURITY_LEVEL assignments from the specified region, and then retry the operation.</td>
</tr>
<tr>
<td><strong>Message</strong>: “LogicLock region <code>&lt;region_name&gt;</code> cannot be a Security Routing Interface, because it is the root region”</td>
</tr>
<tr>
<td><strong>Cause</strong>: You attempted to set the root LogicLock region to be a Security Routing Interface. This operation is not supported.</td>
</tr>
<tr>
<td><strong>Action</strong>: Make sure that the root region is not set as a Security Routing Interface.</td>
</tr>
<tr>
<td><strong>Message</strong>: “LogicLock region <code>&lt;region_name&gt;</code> cannot contain child regions and be a Security Routing Interface at the same time”</td>
</tr>
<tr>
<td><strong>Cause</strong>: A LogicLock region cannot be a Security Routing Interface and contain child regions at the same time. If you receive this message while performing an operation through the GUI, the operation is ignored. If you currently have assignments that violate this rule, the LogicLock region’s Security Routing Interface assignments are ignored.</td>
</tr>
<tr>
<td><strong>Action</strong>: Make sure that no region is a Security Routing Interface and contain child regions at the same time.</td>
</tr>
<tr>
<td><strong>Message</strong>: “LogicLock region <code>&lt;region_name&gt;</code> cannot be a child of another user-created region and be a Security Routing Interface at the same time”</td>
</tr>
<tr>
<td><strong>Cause</strong>: A LogicLock region cannot be a Security Routing Interface and be a child region of another user-created region at the same time. If you receive this message while performing an operation through the GUI, the operation is ignored. If you currently have assignments that violate this rule, the LogicLock region’s Security Routing Interface assignments are ignored.</td>
</tr>
<tr>
<td><strong>Action</strong>: Make sure that no region is a Security Routing Interface and is a child region of another user-created region at the same time.</td>
</tr>
<tr>
<td>Message and Description</td>
</tr>
<tr>
<td>-------------------------</td>
</tr>
</tbody>
</table>
| **Message**: “Can’t specify LogicLock region `<region_name>` as a Security Routing Interface, because the current device is automatically selected”  
**Cause**: When the current device is automatically selected, no LogicLock region can be a Security Routing Interface. A Security Routing Interface must have a fixed size and a locked origin, both of which are impossible when the device is set to `Auto`. If you receive this message while performing an operation through the GUI, the operation is ignored. If you currently have assignments that violate this rule, the LogicLock region’s Security Routing Interface assignments are ignored.  
**Action**: If you want to specify one or more LogicLock regions as Security Routing Interface, use a specific device instead of the automatically selected device.  
|  
| **Message**: “LogicLock region `<region_name>` cannot be **Auto**, **Floating**, **Non-reserved**, and/or **Soft** when it is a Security Routing Interface”  
**Cause**: A LogicLock region cannot be a Security Routing Interface and be **Auto**, **Floating**, **Non-reserved**, and/or **Soft**. If you get this message while performing an operation through the GUI, the operation is ignored. If you currently have assignments that violate this rule, the LogicLock region has been updated to be **Reserved** and **Non-Soft**, and have a fixed size and a locked origin.  
**Action**: Make sure that no region is a Security Routing Interface while it is also **Auto**, **Floating**, **Non-reserved**, and/or **Soft**.  
|  
| **Message**: “LogicLock region `<region_name>` cannot have an LL_SECURITY_LEVEL assignment when it is a Security Routing Interface. A Security Routing Interface automatically has a security level equal to the maximum security level of all signals assigned to it.”  
**Cause**: You attempted to specify an LL_SECURITY_LEVEL assignment to a LogicLock region that is specified as a Security Routing Interface. A Security Routing Interface does not accept a user-specified security level, because it automatically has a security level equal to the maximum security level of all signals assigned to it.  
**Action**: Make sure that no region that is specified as a Security Routing Interface has an associated LL_SECURITY_LEVEL assignment.  
|  
| **Message**: “LogicLock region `<region_name>` cannot have LL_SIGNAL_SECURITY_LEVEL assignments when it is a Security Routing Interface. A Security Routing Interface is reserved for routing signals, and cannot contain logic that generates signals.”  
**Cause**: A LogicLock region specified as a Security Routing Interface cannot have LL_SIGNAL_SECURITY_LEVEL assignments associated with it. A Security Routing Interface is reserved for routing signals into and/or out of one or more secured LogicLock regions, and therefore cannot contain logic that generates signals. Any LL_SIGNAL_SECURITY_LEVEL assignments associated with the specified region are ignored.  
**Action**: Make sure that no region that is specified as a Security Routing Interface has an associated LL_OUTPUT_SIGNAL_SECURITY_LEVEL assignment.  
|  
| **Message**: “LogicLock region `<region_name>` cannot have placement-controlling members (that is, LL_MEMBER_OF) when it is a Security Routing Interface. A Security Routing Interface is reserved for routing signals, and cannot contain user logic.”  
**Cause**: A LogicLock region specified as a Security Routing Interface cannot have LL_MEMBER_OF assignments associated with it. A Security Routing Interface is reserved for routing signals into and/or out of one or more secured LogicLock regions, and therefore cannot contain user logic. Any LL_MEMBER_OF assignments associated with the specified region are ignored.  
**Action**: Make sure that no region that is specified as a Security Routing Interface has an associated LL_MEMBER_OF assignment.  

---

Table 7. Design Separation Flow Messages (Part 3 of 10)
### Table 7. Design Separation Flow Messages (Part 4 of 10)

<table>
<thead>
<tr>
<th>Message and Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Message:</strong> “Can’t specify LogicLock region <code>&lt;region_name&gt;</code> as a Security Routing Interface, either because you do not have the required license or the currently selected device family does not support the Design Separation flow”</td>
</tr>
<tr>
<td><strong>Cause:</strong> You attempted to specify a LogicLock region to be a Security Routing Interface, which is not supported by either the current Quartus II software license, or the currently selected device family, or both. If you get this message while performing an operation through the GUI, the operation has been ignored. If you have LL_SECURITY_ROUTING_INTERFACE assignments in the <code>.qsf</code>, the assignments have been ignored.</td>
</tr>
<tr>
<td><strong>Action:</strong> Make sure that you have the required license and the currently selected device family supports the feature, and then retry the operation.</td>
</tr>
<tr>
<td><strong>Message:</strong> “Can’t add signal <code>&lt;signal_name&gt;</code> to LogicLock region <code>&lt;region_name1&gt;</code> as a security routing member because it is already assigned to LogicLock region <code>&lt;region_name2&gt;</code>.</td>
</tr>
<tr>
<td><strong>Cause:</strong> You tried to add a new signal to a LogicLock region as a security routing member, but such member already exists in another region. A signal cannot be assigned to more than one region.</td>
</tr>
<tr>
<td><strong>Action:</strong> Ensure that the same signal is not assigned to more than one LogicLock region.</td>
</tr>
<tr>
<td><strong>Message:</strong> “LogicLock region <code>&lt;region_name&gt;</code> does not contain a security routing member with signal name <code>&lt;signal_name&gt;</code>.”</td>
</tr>
<tr>
<td><strong>Cause:</strong> You tried to perform an operation for the specified security routing member that does not exist.</td>
</tr>
<tr>
<td><strong>Action:</strong> Ensure that the selected security routing member exists before performing the operation again.</td>
</tr>
<tr>
<td><strong>Message:</strong> “LogicLock region <code>&lt;region_name&gt;</code> cannot have LL_SIGNAL_SECURITY_LEVEL assignments, because it is not specified as a secured region. Only secured regions can change the security level of their signals.”</td>
</tr>
<tr>
<td><strong>Cause:</strong> A LogicLock region not specified as secured cannot have LL_SIGNAL_SECURITY_LEVEL assignments associated with it. The LL_SIGNAL_SECURITY_LEVEL assignment can only be used to lower the security level of a signal coming out of a secured region. If you receive this message while performing an operation in the GUI, the operation is ignored. If there are violating assignments in the <code>.qsf</code>, any LL_OUTPUT_SIGNAL_SECURITY_LEVEL assignments specified for an unsecured region are ignored.</td>
</tr>
<tr>
<td><strong>Action:</strong> Either remove the LL_SIGNAL_SECURITY_LEVEL assignments to avoid this message, or set the specified region as secured to have the LL_SIGNAL_SECURITY_LEVEL assignments take effect.</td>
</tr>
<tr>
<td><strong>Message:</strong> “Can’t set LogicLock region <code>&lt;region_name&gt;</code> to unsecured because it contains security level assignments for its signals (that is, LL_SIGNAL_SECURITY_LEVEL). Remove the assignments from the Security tab of the LogicLock Region property sheet, and then try the operation again.”</td>
</tr>
<tr>
<td><strong>Cause:</strong> A LogicLock region cannot be set as unsecured while it still has LL_SIGNAL_SECURITY_LEVEL assignments associated with it.</td>
</tr>
<tr>
<td><strong>Action:</strong> Remove the LL_SIGNAL_SECURITY_LEVEL assignments associated with the region, and then try the operation again.</td>
</tr>
<tr>
<td><strong>Message:</strong> “Can’t specify <code>&lt;signal_name&gt;</code> as a signal of LogicLock region <code>&lt;region_name&gt;</code> because LogicLock region <code>&lt;region_name&gt;</code> has already specified it as a signal”</td>
</tr>
<tr>
<td><strong>Cause:</strong> You tried to specify a signal as a signal of a LogicLock region, using the LL_SIGNAL_SECURITY_LEVEL assignment, but another region also specifies the same signal as its output.</td>
</tr>
<tr>
<td><strong>Action:</strong> Ensure that the same signal is not specified by two or more LL_SIGNAL_SECURITY_LEVEL assignments.</td>
</tr>
<tr>
<td><strong>Message:</strong> “Signal name <code>&lt;signal_name&gt;</code> is invalid. Ignoring output signal assignment for LogicLock region <code>&lt;region_name&gt;</code>.”</td>
</tr>
<tr>
<td><strong>Cause:</strong> You tried to create a LL_SIGNAL_SECURITY_LEVEL assignment using an invalid signal name. The signal name must be a valid hierarchy path. Wildcard characters are not supported. The LL_SIGNAL_SECURITY_LEVEL assignment is ignored.</td>
</tr>
<tr>
<td><strong>Action:</strong> Specify a valid signal name to have the assignment recognized.</td>
</tr>
</tbody>
</table>
Table 7. Design Separation Flow Messages (Part 5 of 10)

<table>
<thead>
<tr>
<th>Message and Description</th>
</tr>
</thead>
</table>
| **Message:** “LogicLock region `<region_name>` does not contain a signal with name `<signal_name>`”  
**Cause:** You tried to perform an operation for the specified signal that does not exist.  
**Action:** Ensure that the selected signal exists before performing the operation again.  |
| **Message:** “Can’t specify security level `<security_level>` for signal `<signal_name>`, because it is higher than `<security_level>` which is the security level of its source region `<source_region_name>`”  
**Cause:** You tried to set the security level of a secured region to a value that is lower than the security level of at least one of its signals. A region must be as secure as its signals.  
**Action:** Review the security requirements of the design. Consider lowering the security level of the signals, and then retry the operation.  |
| **Message:** “Can’t specify LogicLock region `<region_name>` as secured because it has a non-contiguous shape”  
**Cause:** You attempted to specify a LogicLock region that has a non-contiguous shape to be secured. A secured LogicLock region must be contiguous. The requested operation is ignored.  
**Action:** Change the shape of the LogicLock region to be contiguous using the Chip Planner, and then retry the operation.  |
| **Message:** “Pin `<pin_name>` with security level `<security_level>` of secured region `<secure_region_name>` cannot be placed in the I/O bank `<I0_bank_num>`, because the bank already has `<number_of_pins>` pin(s) with security level `<security_level>` of secured region `<region_name>`”  
**Cause:** You assigned the pin to the specified pin location. However, the Fitter cannot place the pin at the location because the I/O bank has the specified number of pins from an incompatible secured region.  
**Action:** Delete or change the location assignment.  |
| **Message:** “Pin `<pin_name>` has security level `<security_level>`”  
**Cause:** The specified pin has the specified security level.  
**Action:** No action is required.  |
| **Message:** “I/O bank `<I0_bank_num>` has security level `<security_level>` which is contributed by `<number_of_pins>` pin(s) in the bank”  
**Cause:** The specified I/O bank has specified number of pin(s) with the specified security level.  
**Action:** No action is required.  |
| **Message:** “Following placement warning has caused a placement error, because the Fitter setting, **Force Fitter to Avoid Periphery Placement Warnings**, is turned on”  
**Cause:** The **Force Fitter to Avoid Periphery Placement Warnings** setting is turned on, causing a Fitter error due to placement warnings.  
**Action:** Change the **Force Fitter to Avoid Periphery Placement Warnings** setting to **On**.  |
| **Message:** “The following placement warning has caused a placement error because the Fitter setting, **Allow Placement Warnings**, is off”  
**Cause:** The **Allow Placement Warnings** setting is off, causing a Fitter error due to placement warnings.  
**Action:** Change the **Allow Placement Warnings** setting to **On**.  |
| **Message:** “Pin `<pin_name>` with security level `<security_level>` cannot be placed in pin location `<pin_location>` of the I/O bank `<I0_bank_num>`, because the pin location is too close to `<number>` I/O bank(s) with incompatible secured region(s)”  
**Cause:** You assigned the pin at the specified pin location. However, the Fitter cannot place the pin at the location because the pin location is adjacent to pin location(s) of other I/O bank(s) with incompatible security region(s).  
**Action:** Delete or Change the location assignment or the security level assignment, or place output driving ground pin(s) at the adjacent pin location(s) of incompatible I/O bank(s).  |
Table 7. Design Separation Flow Messages (Part 6 of 10)

<table>
<thead>
<tr>
<th>Message and Description</th>
</tr>
</thead>
</table>
| **Message:** “Package trace of pin location `<pin_location1>` is adjacent to the package trace of pin location `<pin_location2>`”  
**Cause:** Refer to the parent message for more information.  
**Action:** Refer to the parent message for more information. |
| **Message:** “Pin Location `<pin_location1>` is adjacent to the pin location `<pin_location2>` on the package”  
**Cause:** Refer to the parent message for more information.  
**Action:** Refer to the parent message for more information. |
| **Message:** “Pin `<pin_name>` with security level `<security_level>` of secured region `<region_name>` at the pin location `<pin_location>`”  
**Cause:** Refer to the parent message for more information.  
**Action:** Refer to the parent message for more information. |
| **Message:** “Pin `<pin_name>` with security level `<security_level>` of secured region `<region_name>` cannot be placed at the pin location `<pin_location>`, because the pin location is too close to `<number>` pin(s) of incompatible secured region(s)”  
**Cause:** You assigned the pin at the specified pin location. However, the Fitter cannot place the pin at the location because the pin location is too close to pin(s) of incompatible secured region(s).  
**Action:** Delete or change the location assignment or the security assignment. |
| **Message:** “Engineering Change Order (ECO) Fitter does not support designs with secured regions”  
**Cause:** You directed the Fitter to check and save all netlist changes after making ECO changes to the design; however, the design contains secured regions that prevent the ECO Fitter from proceeding.  
**Action:** No action is required |
| **Message:** “Design Separation flow failed due to the following `<number>` errors.”  
**Cause:** This message is shown because one or more errors occurred during a Design Separation flow compilation. The errors are listed as sub-messages.  
**Action:** Refer to the sub-messages to determine an action. |
| **Message:** “Assignment of security level `<security_level>` to signal `<signal_name>`”  
**Cause:** This information message lists the signal names and their assignments.  
**Action:** Refer to the error message to determine an action. END_ACTION |
| **Message:** “Signal `<signal_name>` has an LL_SIGNAL_SECURITY_LEVEL assignment to region `<region_name>` but is an output of a different region”  
**Cause:** You attempted to compile a design with incorrect assignments. The specified signal is not an output of the LogicLock region to which its LL_SIGNAL_SECURITY_LEVEL refers.  
**Action:** Remove the incorrect LL_SIGNAL_SECURITY_LEVEL assignment, and then try to compile the design again. |
| **Message:** “Found multiple conflicting LL_SIGNAL_SECURITY_LEVEL assignments to names that refer to signal `<signal_name>`.”  
**Cause:** You attempted to compile a design with incorrect assignments. There are multiple conflicting assignments to the same signal, and the alias has different names. The names used are listed.  
**Action:** Remove one or more of the incorrect LL_SIGNAL_SECURITY_LEVEL assignments, and then try to compile the design again. |
# Table 7. Design Separation Flow Messages (Part 7 of 10)

<table>
<thead>
<tr>
<th>Message and Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Message:</strong> “Signal <code>&lt;signal_name&gt;</code> is an output of secured region <code>&lt;source_region_name&gt;</code> and is missing a Security Routing Interface assignment to route from region <code>&lt;source_region_name&gt;</code> to node <code>&lt;node_name&gt;</code> in secured region <code>&lt;destination_region_name&gt;</code>.”</td>
</tr>
<tr>
<td><strong>Cause:</strong> You attempted to compile a design with incomplete assignments. The specified signal is fed by and/or feeds a secured region and must be assigned as a member of one or more Security Routing Interfaces. The missing Security Routing Interface must abut the two regions given.</td>
</tr>
<tr>
<td><strong>Action:</strong> Add the correct LL_MEMBER_OF_SECURITY_ROUTING_INTERFACE assignment, and then try to compile the design again.</td>
</tr>
<tr>
<td><strong>Message:</strong> “Signal <code>&lt;signal_name&gt;</code> is an output of secured region <code>&lt;secured_region&gt;</code> and is missing a Security Routing Interface assignment to route from region <code>&lt;region_name&gt;</code> to unsecured logic node <code>&lt;node_name&gt;</code>.”</td>
</tr>
<tr>
<td><strong>Cause:</strong> You attempted to compile a design with incomplete assignments. The specified signal is fed by and/or feeds a secured region and must be assigned as a member of one or more Security Routing Interfaces. The missing Security Routing Interface must abut the two regions given.</td>
</tr>
<tr>
<td><strong>Action:</strong> Add the correct LL_MEMBER_OF_SECURITY_ROUTING_INTERFACE assignment, and then try to compile the design again.</td>
</tr>
<tr>
<td><strong>Message:</strong> “Signal <code>&lt;signal_name&gt;</code> is an output of unsecured logic, and is missing a Security Routing Interface assignment to route from unsecured logic to node <code>&lt;node_name&gt;</code> in secured region <code>&lt;region_name&gt;</code>.”</td>
</tr>
<tr>
<td><strong>Cause:</strong> You attempted to compile a design with incomplete assignments. The specified signal is fed by and/or feeds a secured region and must be assigned as a member of one or more Security Routing Interfaces. The missing Security Routing Interface must abut the two regions given.</td>
</tr>
<tr>
<td><strong>Action:</strong> Add the correct LL_MEMBER_OF_SECURITY_ROUTING_INTERFACE assignment, and then try to compile the design again.</td>
</tr>
<tr>
<td><strong>Message:</strong> “Security routing interface <code>&lt;routing_interface&gt;</code> does not abut one or two secured regions”</td>
</tr>
<tr>
<td><strong>Cause:</strong> You attempted to compile a design with an incorrect floorplan. LL_SECURITY_ROUTING_INTERFACE regions are to be used for entering and exiting secured regions and therefore must abut one or two secured regions.</td>
</tr>
<tr>
<td><strong>Action:</strong> Change the location or the size of the LL_SECURITY_ROUTING_INTERFACE region, and then try to compile the design again.</td>
</tr>
<tr>
<td><strong>Message:</strong> “Security routing interface <code>&lt;routing_interface&gt;</code> abuts two secured regions as well as one or more unsecured regions”</td>
</tr>
<tr>
<td><strong>Cause:</strong> You attempted to compile a design with an incorrect floorplan. LL_SECURITY_ROUTING_INTERFACE regions that abut two secured regions may not simultaneously abut unsecured regions.</td>
</tr>
<tr>
<td><strong>Action:</strong> Change the location or size of the LL_SECURITY_ROUTING_INTERFACE region, and then try to compile the design again.</td>
</tr>
<tr>
<td><strong>Message:</strong> “Fencing for region <code>&lt;region_name&gt;</code> is violated by <code>&lt;number&gt;</code> other regions”</td>
</tr>
<tr>
<td><strong>Cause:</strong> You attempted to compile a design with an incorrect floorplan. The region specified has a fencing area around it in which other regions may not be placed.</td>
</tr>
<tr>
<td><strong>Action:</strong> Change the location or size of the LL_SECURITY_ROUTING_INTERFACE region or the surrounding regions, and then try to compile the design again.</td>
</tr>
<tr>
<td><strong>Message:</strong> “Region <code>&lt;region1_name&gt;</code> and region <code>&lt;region2_name&gt;</code> overlap, but this overlap is not allowed”</td>
</tr>
<tr>
<td><strong>Cause:</strong> You attempted to compile a design with an incorrect floorplan. The specified regions overlap, but this overlap is not allowed due to one or more security settings.</td>
</tr>
<tr>
<td><strong>Action:</strong> Change the location or size of the regions to remove the overlap, and then try to compile the design again.</td>
</tr>
</tbody>
</table>
Table 7. Design Separation Flow Messages (Part 8 of 10)

<table>
<thead>
<tr>
<th>Message and Description</th>
<th>Cause</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Region <code>&lt;region_name&gt;</code> has its size set to <em>Auto</em>, but this functionality is not supported in the Design Separation flow”</td>
<td>You attempted to compile a design with incorrect region assignments. The specified region has its size set to <em>Auto</em>, but this functionality is not supported.</td>
<td>Set the size of the specified region to <em>Fixed</em>, and then try to compile the design again.</td>
</tr>
<tr>
<td>“Region <code>&lt;region_name&gt;</code> has its state set to <em>Floating</em>, but this functionality is not supported in the Design Separation flow”</td>
<td>You attempted to compile a design with incorrect region assignments. The specified region has its state set to <em>Floating</em>, but this functionality is not supported.</td>
<td>Fix the origin of the specified region, and then try to compile the design again.</td>
</tr>
<tr>
<td>“Secured region <code>&lt;region_name&gt;</code> is less than eight units tall or eight units wide”</td>
<td>You attempted to compile a design with incorrect region assignments. The specified secured region is smaller than the minimum size of eight units by eight units.</td>
<td>Increase the height or width of the specified region, and then try to compile the design again.</td>
</tr>
<tr>
<td>“Secured region <code>&lt;region_name&gt;</code> has nodes assigned to it that are neither pins nor design partitions”</td>
<td>You attempted to compile a design with incorrect region assignments. The specified region is secured and therefore may contain only a single design partition and pins.</td>
<td>Remove all members of the region that are neither pins nor design partitions, and then try to compile the design again.</td>
</tr>
<tr>
<td>“Secured region <code>&lt;region_name&gt;</code> has <code>&lt;number&gt;</code> design partitions assigned to it, but only one partition is supported”</td>
<td>You attempted to compile a design with incorrect region assignments. The specified region is secured and therefore may contain only a single design partition and pins.</td>
<td>Create a new secured region for every secured design partition, and then try to compile the design again.</td>
</tr>
<tr>
<td>“Secured region <code>&lt;region_name&gt;</code> has path member assignments, but this feature is not supported”</td>
<td>You attempted to compile a design with incorrect region assignments. The specified region is secured and therefore may not contain path member assignments.</td>
<td>Remove the incorrect path member assignments, and then try to compile the design again.</td>
</tr>
<tr>
<td>“LL_MEMBER_OF_SECURITY_ROUTING_INTERFACE assignment of signal <code>&lt;signal_name&gt;</code> to region `%2!s\” was ignored”</td>
<td>You attempted to compile a design with incorrect assignments. The particular assignment listed could not be used, possibly due to an incorrect signal name.</td>
<td>Remove the incorrect assignment, and then try to compile the design again.</td>
</tr>
<tr>
<td>“LL_SIGNAL_SECURITY_LEVEL assignment of signal <code>&lt;signal_name&gt;</code> to region <code>&lt;region_name&gt;</code> was ignored”</td>
<td>You attempted to compile a design with incorrect assignments. The particular assignment listed could not be used, possibly due to an incorrect signal name.</td>
<td>Remove the incorrect assignment, and then try to compile the design again.</td>
</tr>
<tr>
<td>“Found pin <code>&lt;pin_name&gt;</code> in secured region <code>&lt;region_name&gt;</code> but that pin is not connected to a port in “Top\””</td>
<td>You attempted to compile a design with secured region assignments. However, the partition in the secured region contains I/O atoms that are not connected to ports in the top-level design.</td>
<td>Modify the HDL to correct the problem before compiling again.</td>
</tr>
</tbody>
</table>
Table 7. Design Separation Flow Messages (Part 9 of 10)

<table>
<thead>
<tr>
<th>Message and Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Message</strong>: “Secured region <code>&lt;region_name&gt;</code> is invalid - assigning the top partition to a secured region is not allowed”</td>
</tr>
<tr>
<td><strong>Cause</strong>: You attempted to compile a design with incorrect assignments. The top partition is not allowed to be assigned to a secured region.</td>
</tr>
<tr>
<td><strong>Action</strong>: Remove the incorrect assignment, and then try to compile your design again.</td>
</tr>
<tr>
<td><strong>Message</strong>: “Secured region <code>&lt;region_name&gt;</code> is invalid - assigning a non-leaf partition <code>%2ls</code> to a secured region is not allowed”</td>
</tr>
<tr>
<td><strong>Cause</strong>: You attempted to compile a design with incorrect assignments. Only a partition with no children is allowed to be assigned to a secured region.</td>
</tr>
<tr>
<td><strong>Action</strong>: Remove the incorrect assignment, and then try to compile your design again.</td>
</tr>
<tr>
<td><strong>Message</strong>: “Found logic belonging to the same composite I/O group in multiple secured regions”</td>
</tr>
<tr>
<td><strong>Cause</strong>: You attempted to compile a design with incorrect assignments. The I/O group is not allowed to span more than one secured region.</td>
</tr>
<tr>
<td><strong>Action</strong>: Remove the incorrect assignment, and then try to compile your design again.</td>
</tr>
<tr>
<td><strong>Message</strong>: “Node <code>&lt;node_name&gt;</code> belongs to secured region <code>&lt;region_name&gt;</code>”</td>
</tr>
<tr>
<td><strong>Cause</strong>: You attempted to compile a design with incorrect assignments. The I/O group is not allowed to span more than one secured region.</td>
</tr>
<tr>
<td><strong>Action</strong>: Remove the incorrect assignment, and then try to compile your design again.</td>
</tr>
<tr>
<td><strong>Message</strong>: “Adding node <code>&lt;node_name&gt;</code> to secured region <code>&lt;region_name&gt;</code>”</td>
</tr>
<tr>
<td><strong>Cause</strong>: The specified node is being added to the specified secured region due to device constraints.</td>
</tr>
<tr>
<td><strong>Action</strong>: No action is required.</td>
</tr>
<tr>
<td><strong>Message</strong>: “Found wire connection between input port <code>&lt;input_port_name&gt;</code> and output port <code>&lt;output_port_name&gt;</code> in partition <code>&lt;partition_name&gt;</code>”</td>
</tr>
<tr>
<td><strong>Cause</strong>: The Fitter requires that logic exists along all paths through partitions in order to honor the secured region assignments.</td>
</tr>
<tr>
<td><strong>Action</strong>: No action is required. To avoid the warning, modify the design HDL to add logic between the specified ports, alter the interface of the specified partition, or modify the partition assignments.</td>
</tr>
<tr>
<td><strong>Message</strong>: “Found wire connections through partitions -- inserting buffer nodes to ensure that the secured region assignments are honored”</td>
</tr>
<tr>
<td><strong>Cause</strong>: The Fitter requires that logic exists along all paths through partitions in order to honor the secured region assignments.</td>
</tr>
<tr>
<td><strong>Action</strong>: No action is required. To avoid the warning, modify the design HDL to add logic between the specified ports, alter the interface of the specified partition, or modify the partition assignments.</td>
</tr>
<tr>
<td><strong>Message</strong>: “Signal <code>&lt;signal_name&gt;</code> has <code>&lt;number&gt;</code> destinations in secured region <code>&lt;region_name&gt;</code> with different PAD_TO_CORE_DELAY settings. Only one setting is allowed.”</td>
</tr>
<tr>
<td><strong>Cause</strong>: You attempted to compile a design with incorrect PAD_TO_CORE_DELAY settings. An inter-region signal may have at most one PAD_TO_CORE_DELAY setting per secured destination region.</td>
</tr>
<tr>
<td><strong>Action</strong>: Remove the incorrect PAD_TO_CORE_DELAY settings.</td>
</tr>
<tr>
<td><strong>Message</strong>: “Skipped processing of the Auto Global Clock logic options because the Design Separation flow is on”</td>
</tr>
<tr>
<td><strong>Cause</strong>: The Fitter skipped the processing of the various Auto Global options (for example, the Auto Global Clock logic option and the Auto Global Register Control Signals logic option) because there are secured regions in the design.</td>
</tr>
<tr>
<td><strong>Action</strong>: If you do not intend to use any secured regions, remove them from the design. Otherwise, use the global signal assignments to promote a specific signal to use global routing.</td>
</tr>
</tbody>
</table>
### Table 7. Design Separation Flow Messages (Part 10 of 10)

<table>
<thead>
<tr>
<th>Message and Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Message:</strong> “Design Separation flow is enabled. Support for this feature is preliminary in this release.”</td>
</tr>
<tr>
<td><strong>Cause:</strong> The Fitter has detected one or more regions with security settings. The routing operations performed will honor these security settings.</td>
</tr>
<tr>
<td><strong>Action:</strong> No action is required.</td>
</tr>
<tr>
<td><strong>Message:</strong> “Routing constraints are not available for designs that use the Design Separation flow. No routing will be preserved.”</td>
</tr>
<tr>
<td><strong>Cause:</strong> The Fitter has detected routing constraints. However, routing constraints may not be used in conjunction with the Design Separation flow in this release.</td>
</tr>
<tr>
<td><strong>Action:</strong> Remove the routing constraints.</td>
</tr>
</tbody>
</table>
Revision History

Table 8 shows the revision history for this application note.

Table 8. Template Revision History

<table>
<thead>
<tr>
<th>Date and Revision</th>
<th>Changes Made</th>
<th>Summary of Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>June 2009, v1.0</td>
<td>Initial Release.</td>
<td>—</td>
</tr>
</tbody>
</table>