

## Introduction

The Altera® Quartus® II software, version 2.2, easily interfaces with EDA tools such as the Verplex Conformal LEC software and Synplicity Synplify software. In fact, the Quartus II software has built-in support for verifying the logical equivalence between the synthesized netlist from Synplicity Synplify and the post-fit Verilog Quartus Mapped (.vqm) files using Verplex Conformal LEC software.

This application note discusses:

- Formal Verification
- Setting up the Quartus II software to generate the VQM file and Conformal LEC script
- Comparing designs using Conformal LEC software
- Known issues and limitations

## Related Links

- Altera web site:
  - *Using the Conformal LEC Software with the Quartus II Software,*  
[www.altera.com/support/software/nativelink/verification/lec/eda\\_view\\_lec\\_using.htm](http://www.altera.com/support/software/nativelink/verification/lec/eda_view_lec_using.htm)
  - *Using the Quartus II Software with other EDA tools,*  
[www.altera.com/support/software/nativelink/quartus2/eda\\_view\\_using\\_eda.htm](http://www.altera.com/support/software/nativelink/quartus2/eda_view_using_eda.htm)

## Formal Verification

Formal verification uses exhaustive mathematical techniques to verify design functionality. There are two types of formal verification: equivalence checking and model checking. This application note discusses equivalence checking.

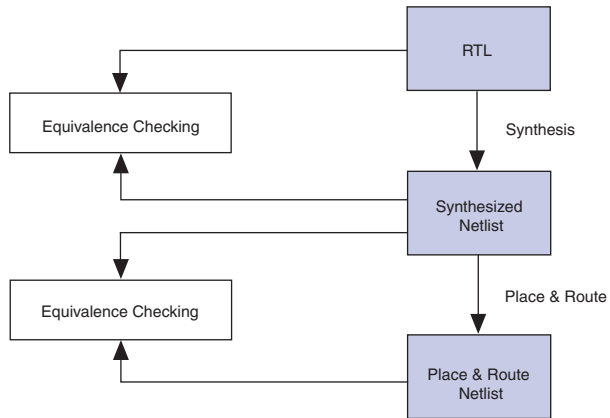
## Equivalence Checking

Equivalence checking is used to compare the functional equivalence between the original design and the modified/revised design by means of mathematical techniques, rather than by performing extensive simulation using test vectors.

## Where to Apply Formal Verification?

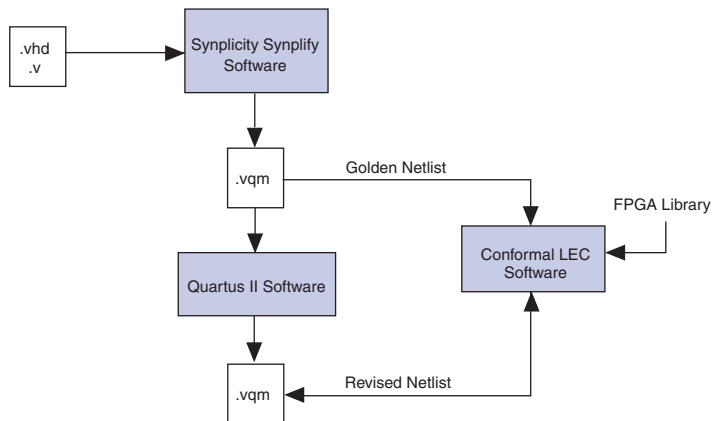
Formal verification can be applied at every stage of the Altera FPGA design flow. See [Figure 1](#).

**Figure 1. Altera FPGA Design Flow**



[Figure 2](#) shows the formal verification flow supported by Altera using Synplicity Synplify and Conformal LEC software.

**Figure 2. Formal Verification Flow Using Synplify & Conformal LEC Software**



## Setting Up the Quartus II Software to Generate the VQM File & Conformal LEC Script

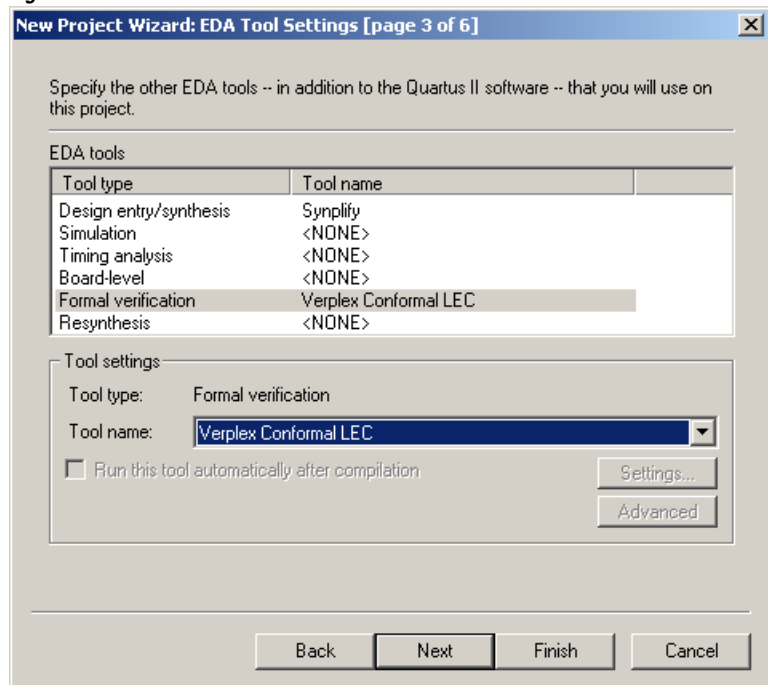
The following steps describe how to set up the Quartus II software environment to generate the place and route, post-fit VQM netlist file and Conformal LEC script compatible for formal verification.

1. If you have not done so, create a new project or open an existing project.
2. Choose **EDA Tools Settings** (Assignments menu)
3. Select **Design entry/synthesis** in the EDA tools list, and scroll to **Synplify** in the Tool name pull-down list (Tool settings window). Next, select **Formal verification** in the EDA tools list, and scroll to **Verplex Conformal LEC** in the Tool name pull-down list (Tool settings window). See [Figure 3](#).



The Quartus II software allows up to six EDA tools to be selected in the EDA tools list.

**Figure 3. EDA Tools Selection**



4. Click **OK**.
5. Select **Start** (Processing menu), and choose **Start >Analysis & Elaboration** from the pop-up window.

If your project includes any of the following, the synthesized VQM netlist file from Synplify contains black boxes:

- Altera library of parameterized modules (LPM's) components
- Encrypted intellectual property (IP) cores
- Entities that are defined in the design format other than Verilog HDL, VHDL, or EDIF

6. If your project does not contain any black boxes in the VQM netlist file, skip to [Step 7](#).

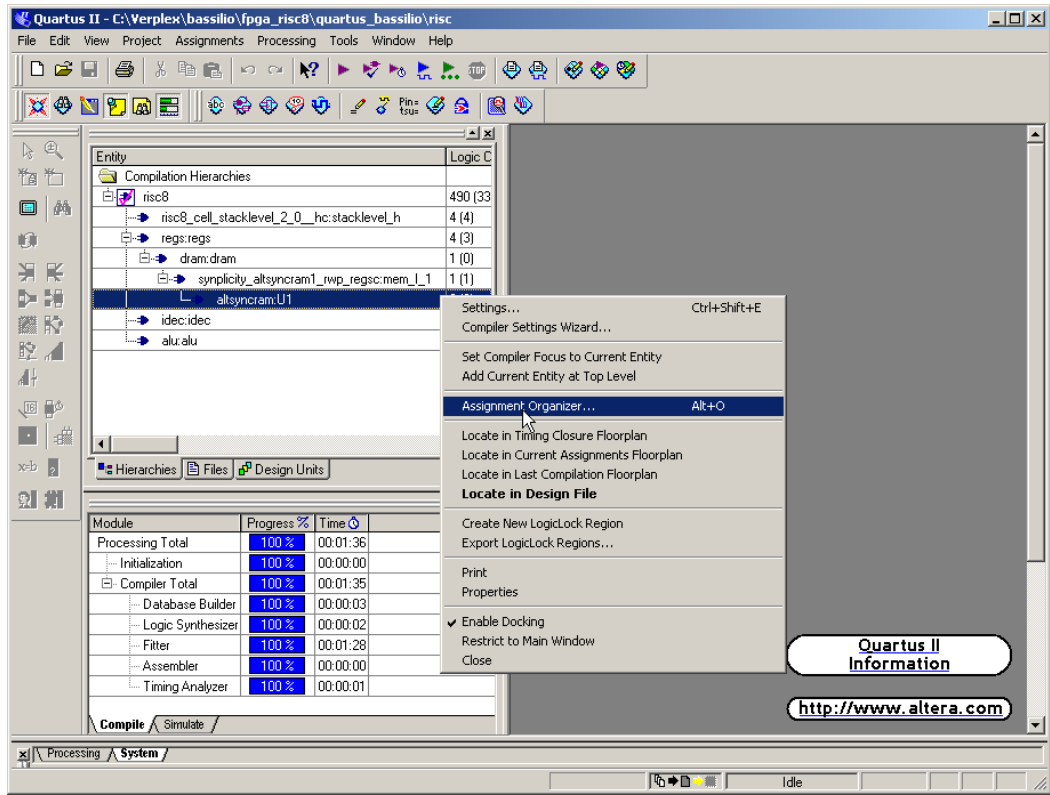
For Quartus 2.2 users:

The hierarchy boundaries of the VQM netlist black boxes need to be preserved. To do so, set the **Preserve Hierarchical Boundary** option to Firm by performing the following steps:

- a. Identify the black boxes from the **Hierarchies** window by expanding the top level design folder (click on the + sign next to the module name) as shown in the [Figure 4](#).
- b. Once the black box (*altsyncram*, in the [Figure 4](#) example) is identified, right-click on the black box and select **Assignment Organizer** in the pop-up window.

At this point, the option **Preserve Hierarchical Boundary** is set on the *altsyncram* entity and not on the Synplicity\_*altsyncram* module that instantiates.

Figure 4. Identifying Synthesized VQM Netlist File Black Boxes &amp; Preserving the Hierarchical Boundary



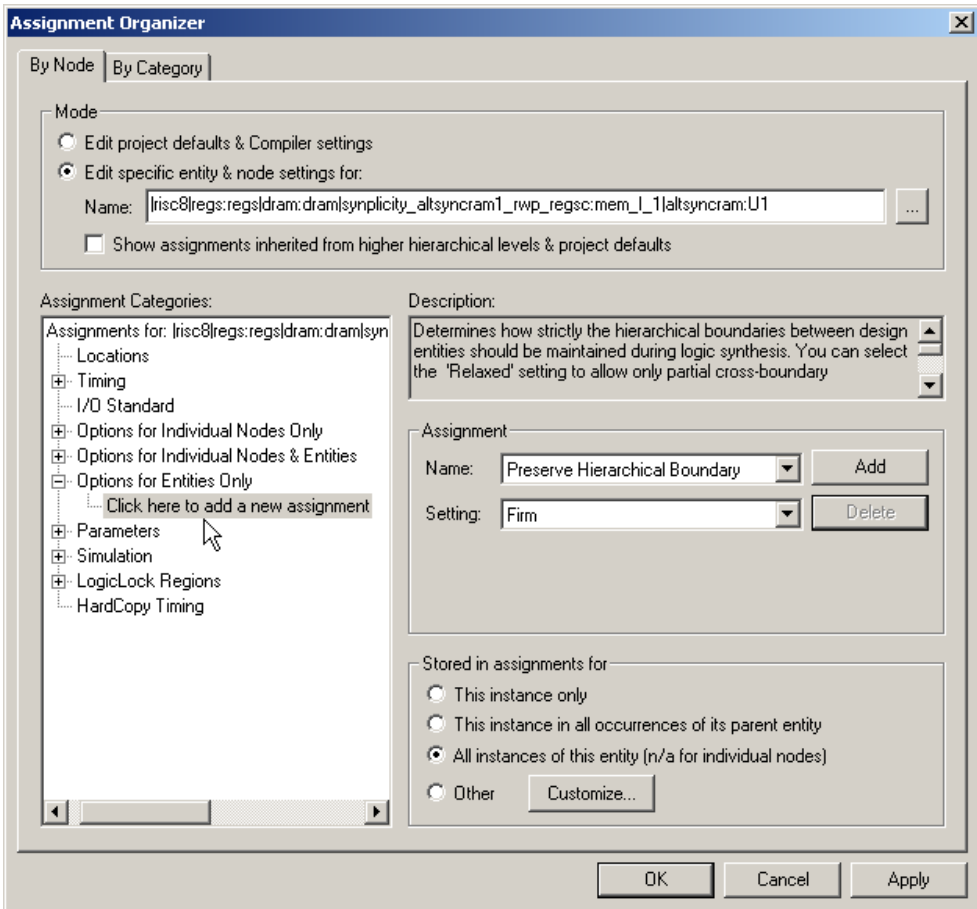
- c. Expand the **Options for Entities Only** folder in the Assignment Categories window and choose **Click here to add a new assignment**. See Figure 5.
- d. Scroll to **Preserve Hierarchical Boundary** in the Assignment Name window and **Firm** in the Assignment Setting window.
- e. Click **Add**.
- f. Check the **All instances of this entity (n/a for individual nodes)** box in the Stored in assignments for window.
- g. Click **OK** to close the window.
- h. Repeat steps *a* through *g* for all black boxes in the design.

For Quartus II 3.0 software users:

Setting the property Preserve Hierarchy Boundary to Firm within Quartus II 3.0 software is available only through Tcl commands. After identifying all the black boxes in the design either create and source a Tcl file within Quartus II software or enter the following commands for each of the black boxes. For example if the black boxes identified are altsyncram and altpll then the set of Tcl commands to be used are:

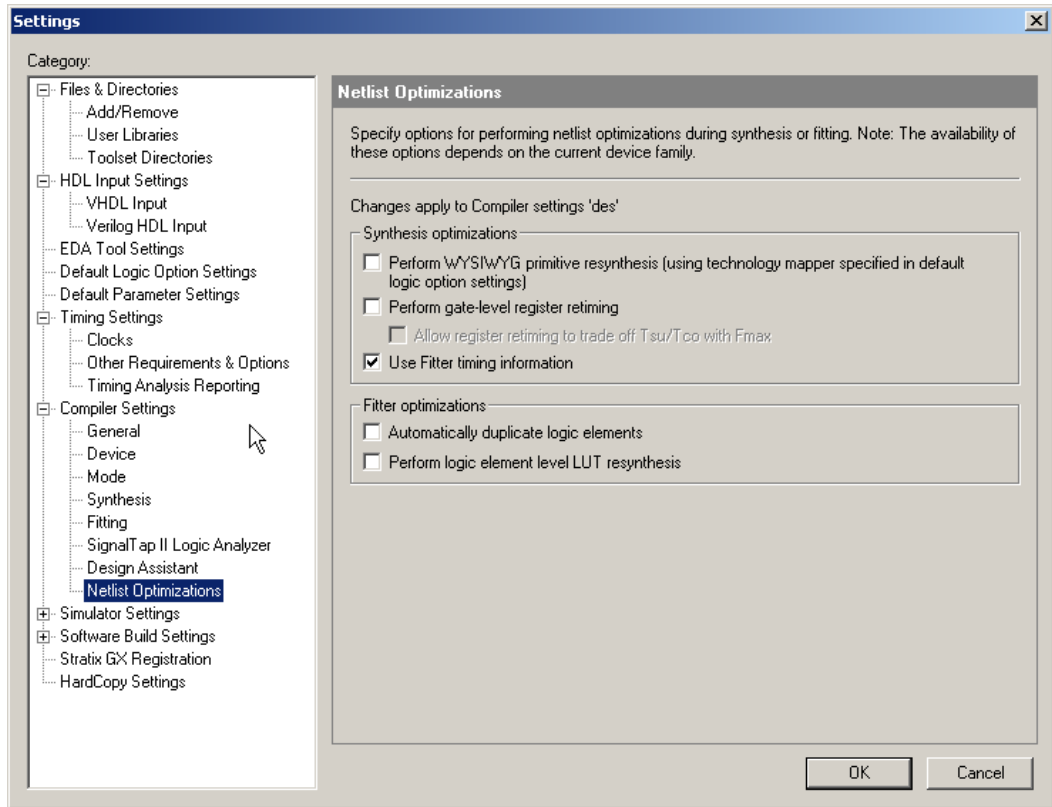
```
package require ::quartus::project
set_global_assignment -entity altsyncram -name preserve_hierarchical_boundary firm
set_global_assignment -entity altpll -name preserve_hierarchical_boundary firm
```

Figure 5. Assignment Organizer Window



7. Choose **Settings** (Assignments menu).
8. In the **Settings** dialog box, expand the Compiler Settings folder (Category window) and select **Netlist Optimizations**.
9. In the Synthesis optimizations window, ensure that both the **Perform WYSIWYG primitive resynthesis (using technology mapper specified in default logic option settings)** and **Perform gate-level register retiming** options are unchecked. Ensure also that both the **Automatically duplicate logic elements** and **Perform logic element level LUT resynthesis** options in the Fitter optimizations window are not checked. See [Figure 6](#).

Figure 6. Setting Parameters for Netlist Optimizations



Retiming on a design usually results in moving and merging registers along the critical path. Because equivalence checkers compare the cones of logic terminating at registers, it is necessary that the registers not be moved, or duplicated, during Quartus II optimization. If the options in this section are not selected, the LEC script could be presented with a different set of compare points, and the resulting netlist will be difficult to compare against the reference netlist file.

10. Perform full compilation of the design either by selecting **Start Compilation** (Processing menu) or by clicking the start compilation arrow icon located in the tool bar.

The Quartus II software compiler generates:

- VQM file: *<design\_name>.vqm*
- Script file: *<design\_name>.vlc* to be used with Conformal LEC software
- A black box file: *<design\_name>\_bbox.v* that contains all the user defined black box entities in the design at:  
*/<project directory>/fv/lec* directory

The *<design\_name>\_bbox.v* file contains the module description of only those entities that are not defined in the formal verification library. For example if there is a reference to black box `altdpram` in the design, the *<design\_name>\_bbox.v* files does not contain the module description for `altdpram` as it is defined in the `mfs_bbox.v` file of the formal verification library.

## Comparing Designs Using Conformal LEC Software

This section discusses using Conformal LEC software to compare designs.

### Black Boxes in the Conformal LEC Flow

A module has to be treated as a black box if the corresponding Conformal LEC formal verification model is not available. As discussed in the [“Setting Up the Quartus II Software to Generate the VQM File & Conformal LEC Script” on page 3](#), the synthesized VQM netlist from Synplify contains black boxes if your project includes any of the following:

- LPM components
- Encrypted IP cores
- Entities that are defined in design format other than Verilog HDL, VHDL, or EDIF

Although every LPM component is treated as a black box by Synplify, if a corresponding Conformal LEC verification model exists, the LPM component is replaced by logic cells in the Quartus II software-generated VQM netlist file. For example, if the design has references to `LPM_MULT` and `LPM_ROM`, only `LPM_ROM` is treated as a black box because the corresponding Conformal LEC verification model is not available. The `LPM_MULT` is replaced by logic cells in the Quartus II software-generated VQM netlist file.



VQM netlist files written by the Quartus II software contain the black box hierarchy when the assignment name **Preserve Hierarchical Boundary** is set to **Firm** (see [Figure 5](#)) on a black box and **VERPLEX CONFORMAL LEC** is selected as the **Formal verification tool** (see [Figure 3](#)). If the **Preserve Hierarchical Boundary** is not set to **Firm** on the black box, the Quartus II software replaces the black box with logic cells and the VQM netlist file no longer contains the black box hierarchy or preserves the port interface, which results in a mismatch with Conformal LEC software.

### Using Conformal LEC Software

Conformal LEC software can be used to verify the VQM netlists generated by Synplify and the Quartus II software either from the command line or from the graphical user interface (GUI) using the `<design_name>.vlc` script file generated by the Quartus II software.

#### *Using Conformal LEC Software from the Command Line*

Type the following at the command prompt:

```
%lec -dofile /<path to project  
directory>/fv/clec/<design_name>.vlc -nogui
```

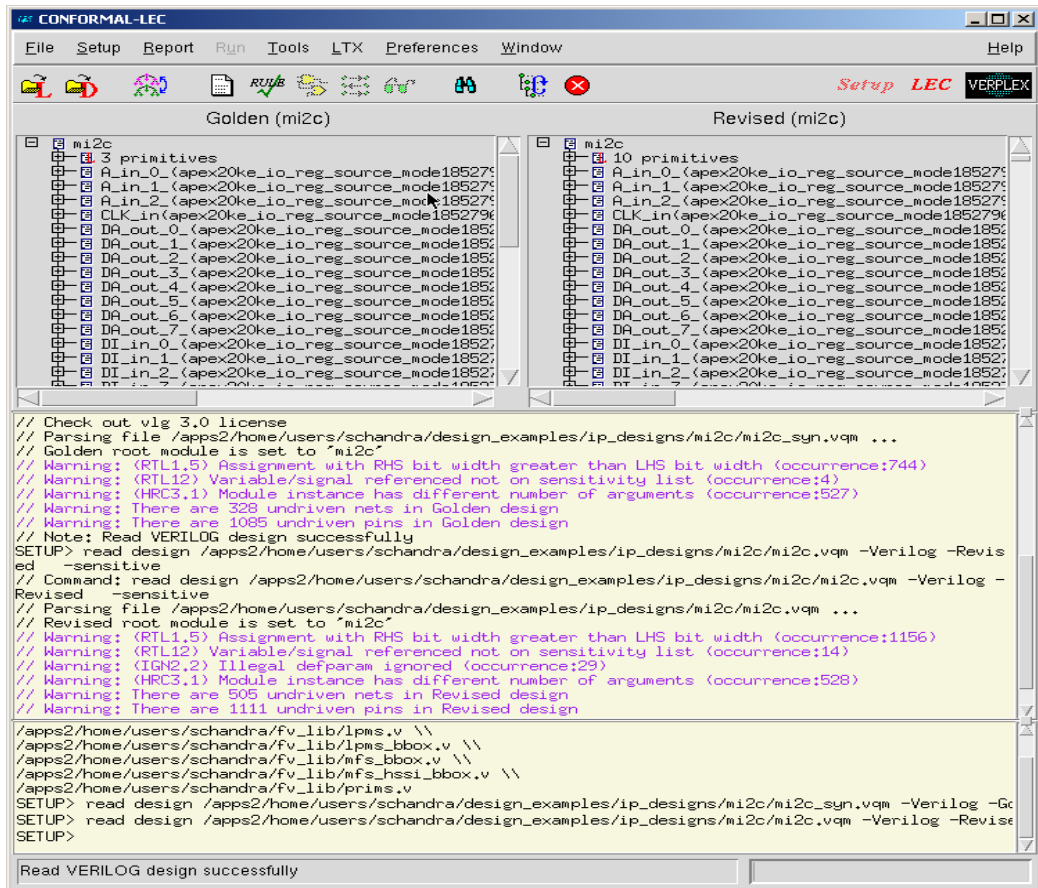
#### *Using Conformal LEC Software from the GUI*

Invoke Conformal LEC software using the command `LEC` from within the GUI.

Select **Do Dofile** (File menu) and select `/<path to project directory>/fv/clec/<design_name>.vlc` file.

Conformal LEC software performs functional comparisons between the VQM netlist files, and then outputs the result as shown in [Figure 7](#).

Figure 7. Conformal LEC Software GUI



The Conformal LEC software shows the original VQM netlist in the **Golden** window and the Quartus II generated VQM netlist in the **Revised** window (see Figure 7). The status section at the bottom of the **Conformal-LEC** window reports the results of the verification, including the number of compared DFFs and POs (Primary Outputs), as well as the number of DFFs and POs that are equivalent and non-equivalent respectively.

To investigate the results of the verification click the **Mapping Manager** icon in the toolbar, or choose **Mapping Manager** (Tools menu). The Conformal LEC software reports the mapped, unmapped, and compared points in the Mapped Points, Unmapped Points, and Compared Points windows respectively.



For more information on how to diagnose the non-equivalent points, refer to the Conformal LEC user manual.

## Known Issues & Limitations

This section discusses known issues and limitations with the formal verification flow using Synplify software, the Quartus II software, and Conformal LEC software:

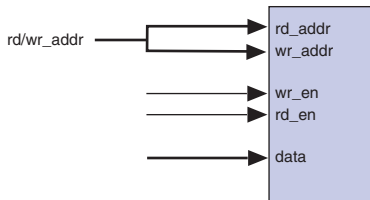
- Formal verification flow supports the APEX™ 20KE, APEX 20KC, APEX II, Cyclone™, Stratix™, and Stratix GX device families.
- Formal verification between RTL and Synplify-generated VQM files for Altera devices is not supported.
- Because the Quartus II software does not preserve the port names in the user-defined black box entities where the resulting entity is synthesized as a wire in the output netlist, black boxes containing logic that can be reduced to a wire in the revised netlist should be ignored in the Golden netlist.
  - For example a black box defined as:

```
module black_box (in,out);
input in;
output out;
endmodule
```
  - And the actual implementation represented by:

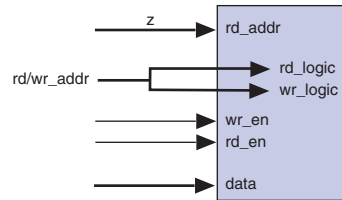
```
module black_box (in,out);
input in;
output out;
assign out = in;
endmodule
```
  - At this point, the Quartus II software replaces the black box with a wire because there is no logic inside the module.
- Unused logic optimized within the black box by the Quartus II software might result in an interface different from the synthesized VQM netlist.
- In designs with combinational feedback loops, the Conformal LEC software may incorrectly insert extra, unmapped cut points in the revised netlist.
- When multiple ports on a black box entity are driven by the same signal, the Quartus II software will push the connections into the entity, and use a single port to drive the internal logic. This action results in the remaining ports being unconnected on the black box entity in the revised netlist. See [Figure 8](#).

**Figure 8. Quartus II Software Consolidates Duplicate Ports into a Single Port to Drive Internal Logic**

*Black Box Ports Before Quartus II Software  
Pushes the Connections Into the Entity*



*Black Box Ports After Quartus II Software  
Pushes the Connections Into the Entity*



- Because of incorrect capitalization of port names in the revised netlist file, designs with the `lpm_dff` megafunction may produce mismatches on the entity and port names. Manually edit the Golden netlist to correct the port names:
  - For example, if the Synplify VQM netlist contains an `LPM_DFF` module:
 

```

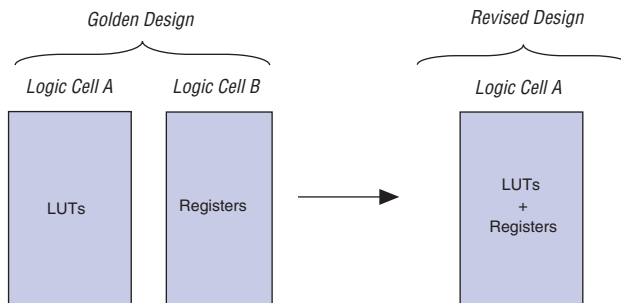
          - LPM_DFF U1 (
          - .Q(Q),
          - .DATA(DATA),
          - .CLOCK(CLOCK),
          - .ASET(ASET),
          - .ACLR(ACLR)
          - );
          - defparam U1.LPM_WIDTH = 1;
          - defparam U1.WIDTH = 1;
          - defparam U1.LPM_TYPE = "LPM_DFF";
          
```
  - Edit the file to replace the description:
 

```

          - lpm_dff U1 (
          - .Q(Q),
          - .DATA(DATA),
          - .CLOCK(CLOCK),
          - .ASET(ASET),
          - .ACLR(ACLR)
          - );
          - defparam U1.lpm_width = 1;
          - defparam U1.width = 1;
          - defparam U1.lpm_type = "LPM_DFF";
          
```
- For Stratix designs that contain RAM blocks, optimization of the RAM blocks in the Quartus II software results in the creation of new port names. This causes Conformal LEC software to report mismatches.

- The LEC software may list non-equivalent points on designs using the `lpm_mult` megafunction, where the width of the product is less than the sum of the widths of the inputs. This issue comes up because the Quartus II software truncates the least significant bits in the width of the product.
- Because the Quartus II software uses a default width of 30 and the LEC software uses a default value of one for the `lpm_width` parameter, the LEC software may report non-equivalent points on the width of the sum port with designs using the `lpm_mult` megafunction when the sum port is not connected. This results in a mismatch between the other 29 bits in the port width.
- Quartus II software groups logic cells (from the Golden design) that use only registers with logic cells that use only look-up-tables (LUTs), and then packs the logic cells into single logic cells with both the LUTs and the registers in the revised design—with an instance name taken from the LUT-only logic cell in the Golden design. The LEC software attempts to map the register-only logic cell name to an incorrect logic cell in the revised design, resulting in a non-equivalent point. Manually edit the mapping in the revised design to map to the correct logic cell. See Figure 9.

**Figure 9. Manually Edit Mapping**



101 Innovation Drive  
San Jose, CA 95134  
(408) 544-7000  
<http://www.altera.com>  
Applications Hotline:  
(800) 800-EPLD  
Literature Services:  
[lit\\_req@altera.com](mailto:lit_req@altera.com)

Copyright © 2003 Altera Corporation. All rights reserved. Altera, The Programmable Solutions Company, the stylized Altera logo, specific device designations, and all other words and logos that are identified as trademarks and/or service marks are, unless noted otherwise, the trademarks and service marks of Altera Corporation in the U.S. and other countries. All other product or service names are the property of their respective holders. Altera products are protected under numerous U.S. and foreign patents and pending warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera Corporation. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.



## Conclusion

Formal verification enables verification of the design during all stages from RTL to place and route. As designs become larger and larger, verification of designs using traditional methods is too time consuming. Thus, formal verification easily verifies that any modifications to the netlist in the physical domain have not altered from the Golden netlist. Advanced debugging capabilities within Conformal LEC software pinpoints the source of the differences between the Golden and Revised netlists, enabling the user to easily fix the differences.

