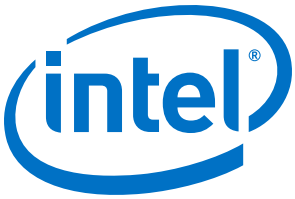


Third-party Logic Equivalence Checking Tools User Guide

Intel® Quartus® Prime Pro Edition

Updated for Intel® Quartus® Prime Design Suite: **18.0**

Subscribe
Send Feedback

UG-20189 | 2018.06.28
Latest document on the web: PDF | HTML

Contents

1. OneSpin 360 EC-FPGA* Software Support

You can optionally use the third-party OneSpin 360 EC-FPGA* sequential equivalence checking tool to verify the logic equivalence between specific netlists following compilation. The 360 EC-FPGA software can help you to confirm that aggressive Compiler optimizations do not introduce unexpected results.

For example, in the current version of the Intel® Quartus® Prime Pro Edition software, you can use the OneSpin 360 EC-FPGA software to confirm the logic equivalence between the routed and retimed netlists after circuit retiming in Intel Stratix® 10 designs, or after making changes to initial conditions. The 360 EC-FPGA tool can confirm that the routed and retimed netlists remain functionally equivalent after those changes.

Note: OneSpin 360 EC-FPGA logic equivalence checking is supported only for Intel Stratix 10 designs in the Intel Quartus Prime Pro Edition software.

1.1. Verifying Post-Route Retiming with OneSpin 360 EC-FPGA software

Enabling OneSpin logic equivalence for Intel Stratix 10 devices involves obtaining the 360 EC-FPGA software and license, adding commands to the Intel Quartus Prime Settings File (`.qsf`), compiling the design, and running the 360 EC-FPGA tool.

Follow these steps to verify functional equivalence between the routed and retimed netlists with OneSpin 360 EC-FPGA software:

1. Obtain the 360 EC-FPGA* software and license for Intel Quartus Prime Retiming Verification from OneSpin Solutions.
2. To enable formal verification in the flow, open your Intel Stratix 10 project in the Intel Quartus Prime Pro Edition software, and add the following assignments to the *<project_name>*`.qsf` file. These assignments direct the Compiler to generate the `onespin.tcl` script during compilation:

```
set_global_assignment -name ENABLE_FORMAL_VERIFICATION ON
set_global_assignment -name EDA_GENERATE_FUNCTIONAL_NETLIST ON -section_id
eda_simulation
set_global_assignment -name EDA_SIMULATION_TOOL "ModelSim (Verilog)"
set_global_assignment -name EDA_TIME_SCALE "1 ps" -section_id
eda_simulation
set_global_assignment -name EDA_OUTPUT_DATA_FORMAT "VERILOG HDL"
-section_id eda_simulation
```

   Note: Although your RTL can be Verilog HDL, SystemVerilog, or VHDL, the Intel Quartus Prime EDA Netlist Writer requires that you specify the assignments only in Verilog HDL, as the example shows.

3. To run full compilation (including Synthesis, Plan, Place, Route, and Retime stages), click **Start Compilation** on the Compilation Dashboard. The Compiler preserves a snapshot of the results at each stage.

4. To generate a Verilog Output File (`.vo`) containing a snapshot of the routed netlist, type the following command in the Console:

   ```
   quartus_eda --snapshot=routed <project_name>
   ```

   The `.vo` generates to `<project_directory>/simulation/modelsim/routed/<project_name>.vo`.

5. To generate a `.vo` containing a snapshot of the retimed netlist, type the following command in the Console:

   ```
   quartus_eda --snapshot=retimed <project_name>
   ```

   The `.vo` generates to `<project_directory>/simulation/modelsim/retimed/<project_name>.vo`.

6. To launch OneSpin 360 EC-FPGA, type the following command in the Console within the project directory:

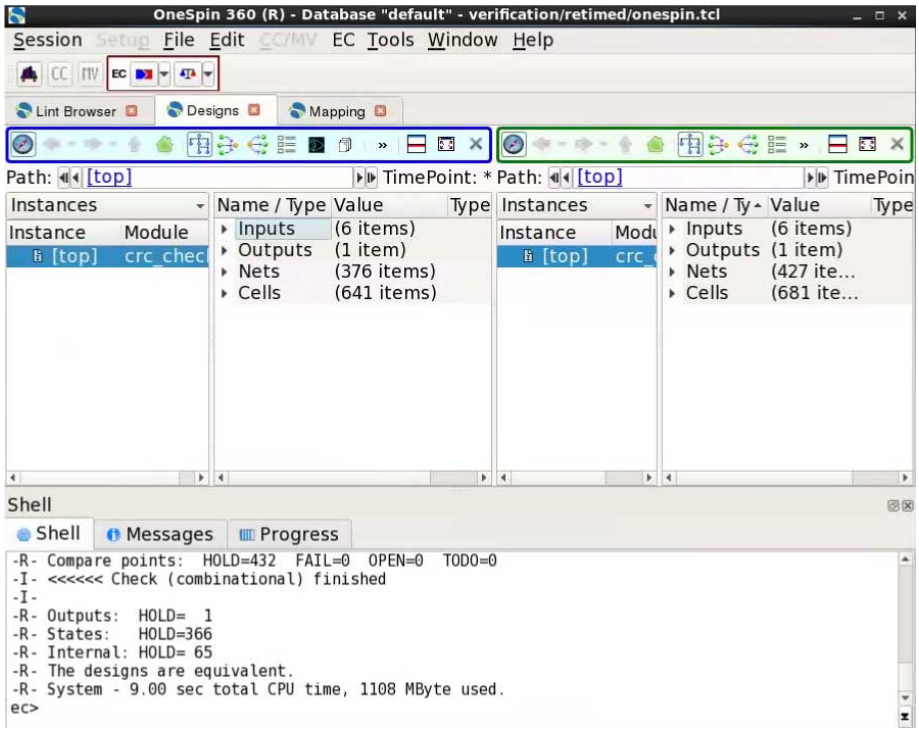
   ```
   onespin -i verification/retimed/onespin.tcl
   ```

   The `-i` option opens the 360 EC-FPGA software GUI.

7. Follow the instructions in OneSpin's 360 EC-FPGA software documentation to run 360 EC-FPGA and confirm logic equivalence of the routed and retimed netlists. 360 EC-FPGA compares the routed and retimed netlists and reports the functional equivalence.

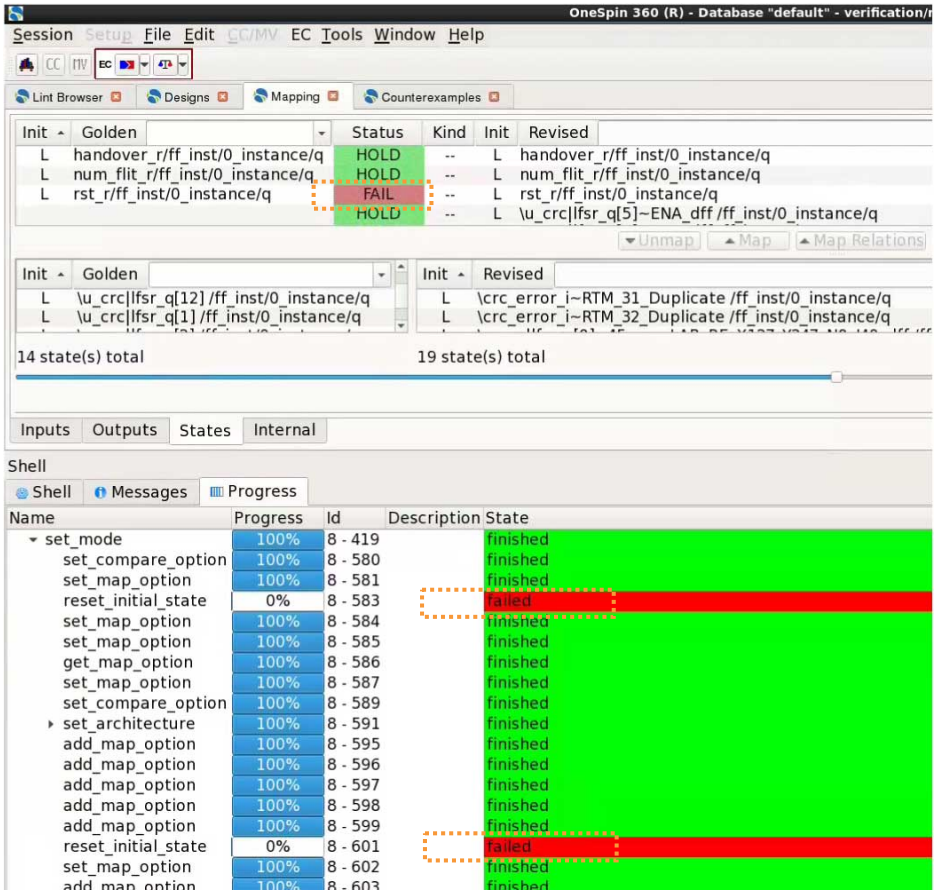
If 360 EC-FPGA reports that the netlists are functionally equivalent, then retiming optimization did not change the functionality of the design.

Figure 1. 360 EC-FPGA* Shows Functionally Equivalent Design



If 360 EC-FPGA reports failed points in the netlist, review the Intel Quartus Prime compilation messages for any warning that relates to this failed point before further debugging the designs in 360 EC-FPGA.

Figure 2. 360 EC-FPGA* Shows Failed Points in Netlist



If the 360 EC-FPGA* reports open states without failure, this indicates that the design complexity may exceed the capability of the 360 EC-FPGA software. Contact OneSpin support for assistance with designs that exceed 360 EC-FPGA software capabilities.

1.2. OneSpin 360 EC-FPGA Software Support Revision History

This document has the following revision history.

Document Version	Intel Quartus Prime Version	Changes
2018.06.28	18.0.0	• Initial release of document.

Related Information

Documentation Archive

For previous versions of the *Intel Quartus Prime Handbook*, search the documentation archives.

A. Intel Quartus Prime Pro Edition User Guides

Refer to the following user guides for comprehensive information on all phases of the Intel Quartus Prime Pro Edition FPGA design flow.

Related Information

- Getting Started User Guide
  Introduces the basic features, files, and design flow of the Intel Quartus Prime Pro Edition software, including managing Intel Quartus Prime Pro Edition projects and IP, initial design planning considerations, and project migration from previous software versions.
- Platform Designer User Guide
  Describes creating and optimizing systems using Platform Designer, a system integration tool that simplifies integrating customized IP cores in your project. Platform Designer automatically generates interconnect logic to connect intellectual property (IP) functions and subsystems.
- Design Recommendations User Guide
  Describes best design practices for designing FPGAs with the Intel Quartus Prime Pro Edition software. HDL coding styles and synchronous design practices can significantly impact design performance. Following recommended HDL coding styles ensures that Intel Quartus Prime Pro Edition synthesis optimally implements your design in hardware.
- Compiler User Guide
  Describes set up, running, and optimization for all stages of the Intel Quartus Prime Pro Edition Compiler. The Compiler synthesizes, places, and routes your design before generating a device programming file.
- Design Optimization User Guide
  Describes Intel Quartus Prime Pro Edition settings, tools, and techniques that you can use to achieve the highest design performance in Intel FPGAs. Techniques include optimizing the design netlist, addressing critical chains that limit retiming and timing closure, and optimization of device resource usage.
- Programmer User Guide
  Describes operation of the Intel Quartus Prime Pro Edition Programmer, which allows you to configure Intel FPGA devices, and program CPLD and configuration devices, via connection with an Intel FPGA download cable.
- Block-Based Design User Guide
  Describes block-based design flows, also known as modular or hierarchical design flows. These advanced flows enable preservation of design blocks (or logic that comprises a hierarchical design instance) within a project, and reuse of design blocks in other projects.

- Partial Reconfiguration User Guide
  Describes Partial Reconfiguration, an advanced design flow that allows you to reconfigure a portion of the FPGA dynamically, while the remaining FPGA design continues to function. Define multiple personas for a particular design region, without impacting operation in other areas.
- Third-party Simulation User Guide
  Describes RTL- and gate-level design simulation support for third-party simulation tools by Aldec*, Cadence*, Mentor Graphics*, and Synopsys* that allow you to verify design behavior before device programming. Includes simulator support, simulation flows, and simulating Intel FPGA IP.
- Third-party Synthesis User Guide
  Describes support for optional synthesis of your design in third-party synthesis tools by Mentor Graphics*, and Synopsys*. Includes design flow steps, generated file descriptions, and synthesis guidelines.
- Debug Tools User Guide
  Describes a portfolio of Intel Quartus Prime Pro Edition in-system design debugging tools for real-time verification of your design. These tools provide visibility by routing (or "tapping") signals in your design to debugging logic. These tools include System Console, Signal Tap logic analyzer, Transceiver Toolkit, In-System Memory Content Editor, and In-System Sources and Probes Editor.
- Timing Analyzer User Guide
  Explains basic static timing analysis principals and use of the Intel Quartus Prime Pro Edition Timing Analyzer, a powerful ASIC-style timing analysis tool that validates the timing performance of all logic in your design using an industry-standard constraint, analysis, and reporting methodology.
- Power Analysis and Optimization User Guide
  Describes the Intel Quartus Prime Pro Edition Power Analysis tools that allow accurate estimation of device power consumption. Estimate the power consumption of a device to develop power budgets and design power supplies, voltage regulators, heat sink, and cooling systems.
- Design Constraints User Guide
  Describes timing and logic constraints that influence how the Compiler implements your design, such as pin assignments, device options, logic options, and timing constraints. Use the Interface Planner to prototype interface implementations, plan clocks, and quickly define a legal device floorplan. Use the Pin Planner to visualize, modify, and validate all I/O assignments in a graphical representation of the target device.
- PCB Design Tools User Guide
  Describes support for optional third-party PCB design tools by Mentor Graphics* and Cadence*. Also includes information about signal integrity analysis and simulations with HSPICE and IBIS Models.
- Scripting User Guide
  Describes use of Tcl and command line scripts to control the Intel Quartus Prime Pro Edition software and to perform a wide range of functions, such as managing projects, specifying constraints, running compilation or timing analysis, or generating reports.