

This chapter describes how to perform different types of simulations with the Quartus II simulator.

Introduction

With today's FPGAs becoming faster and more complex, designers face challenges in validating their designs. Simulation verifies the correctness of the design, reducing board testing and debugging time.

The Altera® Quartus® II simulator is included in the Quartus II software to assist designers with design verification. The Quartus II simulator has a comprehensive set of features that are covered in the following sections:

- "Simulation Flow" on page 1–2
- "Waveform Editor" on page 1–5
- "Simulator Settings" on page 1–13
- "Simulation Report" on page 1–16
- "Debugging with the Quartus II Simulator" on page 1–19
- "Scripting Support" on page 1–21

The Quartus II simulator supports the following device families:

- ACEX® 1K
- APEX™ 20KC, APEX 20KE, APEX II
- Arria® GX
- Cyclone® III, Cyclone II, Cyclone
- FLEX® 10K, FLEX 10KA, FLEX 10KE, FLEX 6000
- HardCopy® II, HardCopy
- MAX® II, MAX 3000A, MAX 7000AE, MAX 7000B, MAX 7000S
- Stratix® III, Stratix II, Stratix, Stratix GX, Stratix II GX



The Quartus II simulator does not support newer devices introduced after Stratix III and Quartus II software version 8.1 and onwards. Use the ModelSim-Altera Edition to run simulations on designs targeting device introductions after Stratix III. For more information about the ModelSim-Altera Edition simulator, refer to the *Mentor Graphics ModelSim Support* chapter in volume 3 of the *Quartus II Handbook*.

In the Quartus II software version 10.0 and onwards, the Quartus II simulator and Waveform Editor is removed. Therefore, you can run your simulation in the EDA simulators. The following EDA simulators are supported in the Quartus II software:

- ModelSim
 - ModelSim SE
 - ModelSim Altera Edition (AE)
 - ModelSim Altera Starter Edition (ASE)
- VCS/VCS-MX
- NCSim
- Active-HDL
- Riviera-PRO



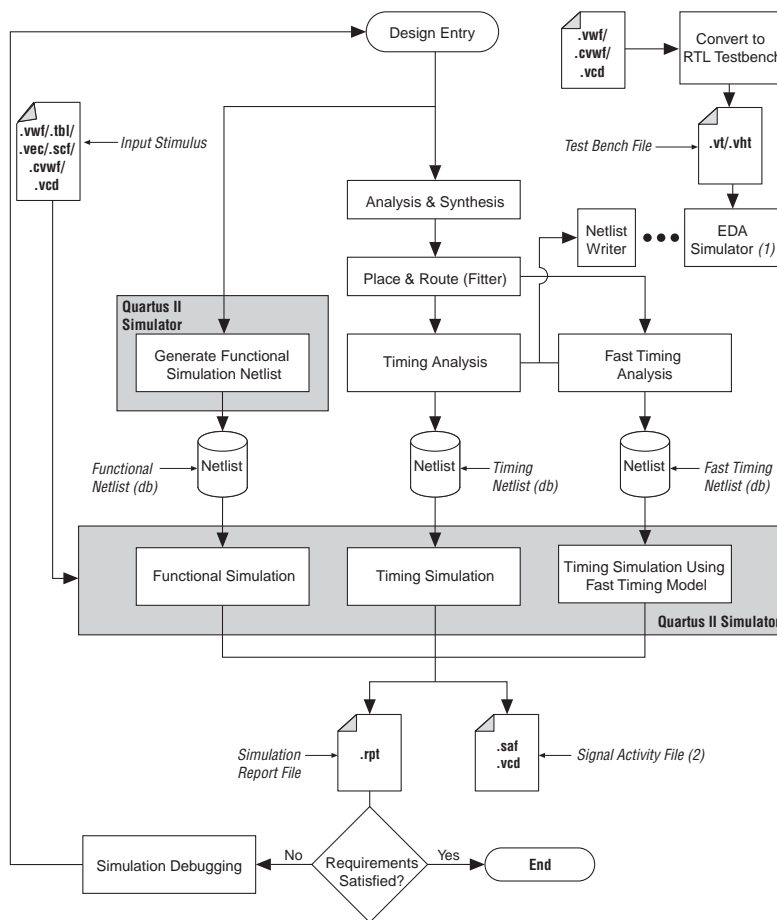
For more information, refer to the *Simulating Designs with EDA Tools* chapter in volume 3 of the *Quartus II Handbook*.

Simulation Flow

You can perform functional and timing simulations with the Quartus II simulator. Both types of simulation verify the correctness and behavior of your design. Functional simulations are run at the beginning of the Quartus II design flow and timing simulations are run at the end.

[Figure 1–1](#) shows the Quartus II simulator flow.

Figure 1-1. Simulation Flow



Notes to Figure 1-1:

- (1) For more information about EDA simulators, refer to the *Simulation* section in volume 3 of the *Quartus II Handbook*.
- (2) You can use Signal Activity Files (.saf) or Value Change Dump Files (.vcd) in the PowerPlay power analyzer to check power resources.

Your design simulation can happen at the functional level, in which the logical behavior of your design is verified and no timing information is used in simulation (refer to Figure 1-1). Timing simulation can happen after your design is compiled (synthesized and placed and routed) and after you use the timing data of your design's resources. In timing simulation, the logical behavior of your design is verified with the device's worst-case timing models. Timing simulation using the fast timing model is also a type of timing simulation in which best-case timing data is used.

To perform functional simulations with the Quartus II simulator, you must first generate a functional simulation netlist. A functional netlist file is a flattened netlist extracted from the design files that does not contain timing information.

For timing simulations, you must first perform place-and-route and static timing analysis to generate a timing simulation netlist. A timing simulation netlist includes timing delays of each device atom block and the routing delays.

If you want to use third-party EDA simulation tools, you can generate a netlist using EDA Netlist Writer. You can use this netlist with your testbench files in third-party simulation tools.



For more information about third-party simulators, refer to the respective EDA Simulation chapter in the *Simulation* section in volume 3 of the *Quartus II Handbook*.

The Quartus II simulator supports Functional, Timing, and Timing using Fast Timing Model simulations. The following sections describe how to perform these simulations.

Functional Simulation

To run a functional simulation, you must perform the following steps:

1. On the Processing menu, click **Generate Functional Simulation Netlist**. This flattens the functional simulation netlist extracted from the design files. The netlist does not contain timing information.
2. On the Assignments menu, click **Settings**. The **Settings** dialog box appears.
3. In the **Category** list, select **Simulator Settings**. The **Simulator Settings** page appears.
4. In the **Simulation mode** list, select **Functional**.
5. In the **Simulation input** box, specify the vector source. You must specify the vector file to run the simulation.
6. Click **OK**.
7. On the Processing menu, click **Start Simulation**.

Timing Simulation

To run a timing simulation, you must perform the following steps:

1. On the Processing menu, click **Start Compilation** or click the **Compilation** button on the toolbar. This flattens the design and generates an internal netlist with timing delay information annotated.
2. On the Assignments menu, click **Settings**. The **Settings** dialog box appears.
3. In the **Category** list, select **Simulator Settings**. The **Simulator Settings** page appears.
4. In the **Simulation mode** list, select **Timing**.
5. In the **Simulation input** list, specify the vector source. You must specify the vector file to run the simulation.
6. Click **OK**.
7. On the Processing menu, click **Start Simulation**.

Timing Simulation Using Fast Timing Model Simulation

To run a timing simulation using a fast timing model, you must perform the following steps:

1. On the Processing menu, point to **Start** and click **Start Analysis and Synthesis**.
2. On the Processing menu, point to **Start** and click **Start Fitter**.



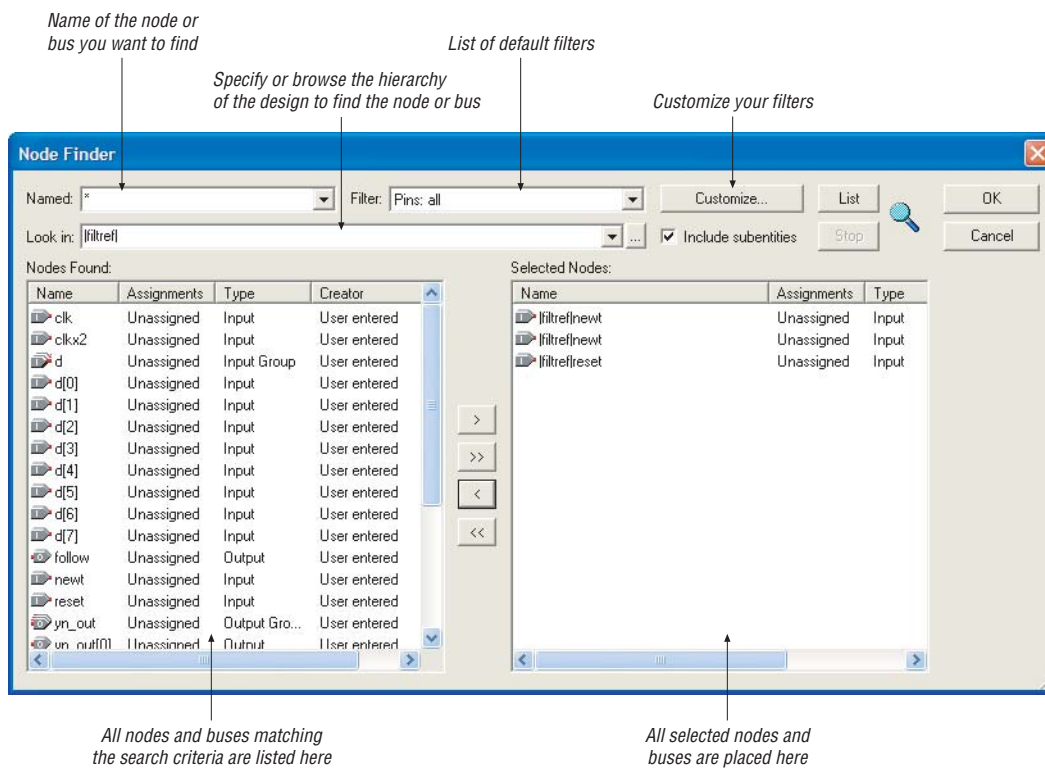
You must perform fast timing analysis before you can perform a timing simulation using the fast timing models.

3. On the Processing menu, point to **Start** and click **Start Classic Timing Analyzer (Fast Timing Model)**.
4. On the Assignments menu, click **Settings**. The **Settings** dialog box appears.
5. In the **Category** list, select **Simulator Settings**. The **Simulator Settings** page appears.
6. In the **Simulation mode** list, select **Timing using Fast Timing Model**.
7. In the **Simulation input** box, specify the vector source. You must specify the vector file to run the simulation.
8. Click **OK**.
9. On the Processing menu, click **Start Simulation**.

Waveform Editor

The most common input stimulus for the Quartus II simulator is Vector Waveform File (**.vwf**). You can use the Quartus II Waveform Editor to generate a **.vwf**.

Figure 1-3. Node Finder Dialog Box



You can use the Node Finder to find your nodes for simulation among all the nodes and buses in your design. Use the Node Finder to filter and add nodes to your waveform. The Node Finder is equipped with multiple default filter options. By using the correct filter in the Node Finder, you can find the internal node's name and add it to your .vwf for simulation.


 Your node might not appear in the simulation waveform and might be ignored during simulation. This happens because the node has been renamed or synthesized away by the Quartus II software. To prevent this from happening, Altera recommends using the register and pin nodes to simulate your design.

Table 1-1 describes 12 of the Node Finder default filters.

Table 1-1. Filter Options (Part 1 of 2)

Filter	Description
Pins: input	Finds all input pin names in your design file or files.
Pins: output	Finds all output pin names in your design file or files.
Pins: bidirectional	Finds all bidirectional pin names in your design file or files.
Pins: virtual	Finds all virtual pin names.
Pins: all	Finds all pin names in your design file or files.
Registers: pre-synthesis	Finds all user-entered register names contained in the design after design elaboration, but before physical synthesis does any synthesis optimizations.

Table 1-1. Filter Options (Part 2 of 2)

Filter	Description
Registers: post-fitting	Finds all user-entered register names in your design file or files that survived physical synthesis and fitting.
Design Entry (all names)	Finds all user-entered names in your design file or files.
Post-Compilation	Finds all user-entered and compiler-generated names that do not have location assignments and survived fitting.
SignalTap II: pre-synthesis	Finds all internal device nodes in the pre-synthesis netlist that can be analyzed by the SignalTap® II Logic Analyzer.
SignalTap II: post-fitting	Finds all internal device nodes in the post-fitting netlist that can be analyzed by the SignalTap II Logic Analyzer.
SignalProbe	Finds all SignalProbe device nodes in the post-fitting netlist.

To customize your own filters in the Node Finder, you must perform the following steps:

- a. Click **Customize**. The **Customize Filter** dialog box appears.
 - b. To configure settings, click **New**. The **New Custom Filter** dialog box appears.
 - c. In the **Filter name** dialog box, type the name of the custom filter.
 - d. In the **Copy settings from filter** list, select the filter setting.
 - e. Click **OK**.
 - f. You can now customize your filters in the **Customize Filter** dialog box.
6. In the **Look in** dialog box, you can view and edit the current search hierarchy path. You can type the search hierarchy path or you can browse for the hierarchy path by clicking the browse button.

You can move up the search hierarchy by selecting hierarchical names in the **Select Hierarchy Level** dialog box. This ensures that in a large design with many signals, you can specify which hierarchy you would like to get the node from to reduce the amount of signals displayed.

7. After you have configured the filter and specified the correct hierarchy in the **Node Finder** dialog box, click **List** to display all relevant nodes or buses.

Select any number of nodes or buses from the **Nodes Found** list and click the > button to include them in the waveform, or you can click the >> button to include all nodes and buses displayed in the **Nodes Found** list.

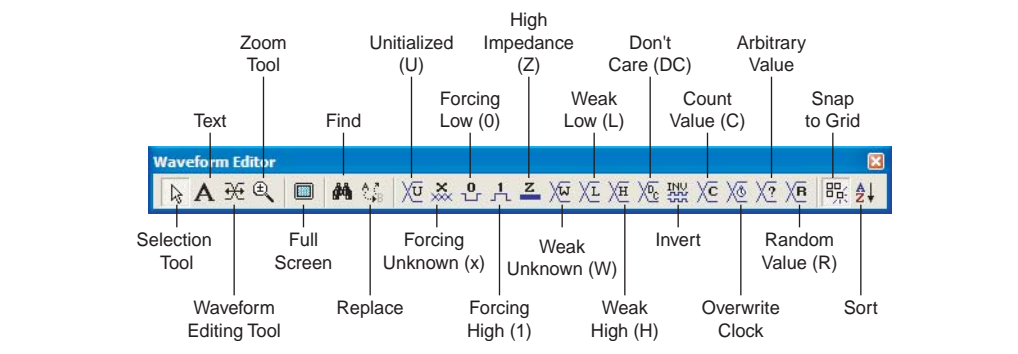
8. Click **OK**.




You can also add nodes to the Waveform Editor by dragging nodes from the Project Navigator, Netlist Viewers, or Block Diagram, and dropping them into the Waveform Editor.

9. Create a waveform for a signal. The Quartus II Waveform Editor toolbar includes some of the most common waveform settings, making waveform vector drawings easier and user friendly. [Figure 1-4](#) shows the options available on the Waveform Editor toolbar.

Figure 1-4. Waveform Editor Toolbar



10. After you edit your waveform, save the waveform. On the File menu, click **Save As**. The **Save As** dialog box appears. Type your file name, specify the file type, and click **Save**.

 Instead of using the Node Finder to insert your nodes for your **.vwf** file, you can also drag-and-drop any nodes from the Netlist Viewer to your Simulation Vector Waveform File. For more information about Netlist Viewers, refer to the *Analyzing Designs with the Quartus II Netlist Viewers* chapter in volume 1 of the *Quartus II Handbook*.

Count Value

Count Value applies a count value to a bus to increment the value of the bus by a specified time interval. Instead of manually editing the values for each node, the Count Value feature on the Waveform Editor toolbar automatically creates the counting values for buses. This feature enables you to specify a starting value for a bus, what time interval to increment, and when to stop counting. You can also configure transition occurrences while setting the count type and increment number. When you click on the **Count Value** button in the Waveform Editor toolbar, the **Count Value** dialog box appears. You can also open the **Count Value** dialog box by right-clicking the selected node, pointing to **Value**, and clicking **Count Value**.

Clock

You can use the Clock feature in the Waveform Editor toolbar to automatically generate the clock wave, rather than drawing each clock triggering pulse. To generate a clock signal with the **Clock** dialog box, click the **Overwrite Clock** button on the Waveform Editor toolbar. You can also determine the start and end time of a clock signal, whether to manually configure the period (the offset and the duty cycle), or whether to generate the clock based on a specified clock.

Arbitrary Value

Arbitrary Value allows you to overwrite a node value over the selected waveform, waveform interval, or across one or more nodes or groups. To overwrite a node value, perform the following steps:

1. Select a node or a bus and click the **Arbitrary Value** button on the Waveform Editor toolbar (refer to [Figure 1-4](#)). The **Arbitrary Value** dialog box appears.

2. Under **Time range**, specify the start and end time you want to overwrite for the node value.
3. In the **Radix** list, select the radix type.
4. Specify the new value you want overwritten in the **Numeric or named value** box.
5. Click **OK**.

Random Value

Random Value allows you to generate random node values over the selected waveform, waveform interval, or across one or more nodes or groups.

You can generate random node values by every grid interval, every half grid interval, at random intervals, or at fixed intervals.

Generating a Testbench

You can export your **.vwf** file as a VHDL Test Bench File (**.vht**) or Verilog Test Bench File (**.vt**). This is useful when you want to use a vector waveform in different EDA tools. You must run an analysis and elaboration before you can export a waveform vector. To export a waveform vector, have your vector waveform open and perform the following steps:

1. On the File menu, click **Export**. The **Export** dialog box appears.
2. In the **Save as type** list, select **VHDL Test Bench File (*.vht)** or **Verilog Test Bench File (*.vt)**.
3. You can optionally turn on the **Add self-checking code to file** option. This option adds additional logic to check the results of the output and compares it to the original **.vwf** file.



You must open your project in the Quartus II software before you can export a **.vwf** file.



For more information about using the generated testbench in other EDA tools, refer to the respective EDA simulator chapter in the *Simulation* section in volume 3 of the *Quartus II Handbook*.

Grid Size

When you select portions of your waveform, the selection area snaps to time intervals specified in the **Grid Size** dialog box. You can customize the grid size in the Waveform Editor. You can change the grid size based on the clock settings or by setting the time period. To customize the grid size, on the Edit menu, click **Grid Size**.

Time Bars

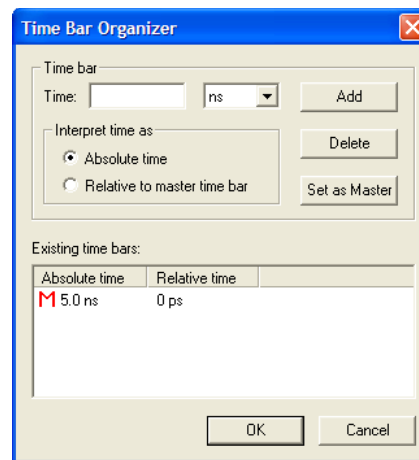
Add time bars in the Waveform Editor to compare edges between different signals. You can also use time bars to jump forward and backward to the next edge transition in the selected signal, and read the logic level of signals by sliding the Time Bar in your waveform. The logic level is displayed in the **Value at** column of the Waveform Editor.

The **Time Bar Organizer** dialog box enables you to create, delete, and edit a time bar, and to create a master time bar. Only one master time bar is allowed per waveform file. To use the Time Bar Organizer, on the Edit menu, point to **Time Bar** and click **Time Bar Organizer**.



Under **Existing time bars**, in the **Absolute time** column, the red **M** indicates the master time bar (Figure 1-5).

Figure 1-5. Time Bar Organizer Dialog Box



Stretch or Compress a Waveform Interval

You can stretch or compress a waveform interval in the Waveform Editor, which enables you to analyze the effects on a waveform. For example, you can check the behavior of your design at high speeds for a short interval by using the compress option to compress the waveform. You can also use this feature to delay the transition of a signal by stretching the waveform.

You have to specify the original start and end time, and the new time for the waveform you want to stretch or compress. If you want to stretch or compress all the nodes or buses, deselect all nodes and buses and set the stretch or compress feature.

To stretch or compress a waveform interval, on the Edit menu, point to **Value** and click **Stretch or Compress Waveform Interval**. The **Stretch or Compress Waveform Interval** dialog box appears.

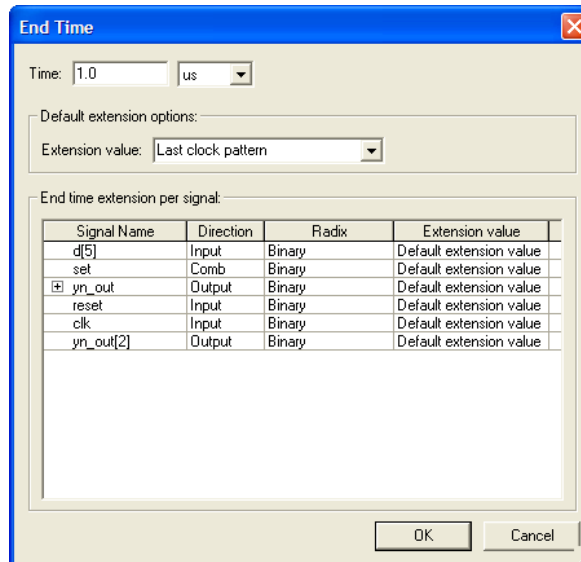
The “To time value” end time specified in the **Stretch or Compress Waveform Interval** dialog box cannot be larger than the “End Time” specified in the **Simulator Settings** page of the **Settings** dialog box. Otherwise, the Quartus II software displays a message indicating the invalid time value.

End Time

The End Time setting enables you to change the end time of the **.vwf** file. The end time represents the maximum length of time in the **.vwf** file. You can specify the end time and your preferred time unit, and have different extension values for different nodes or buses. With the waveform open, specify the end time by performing the following steps:

1. On the Edit menu, click **End Time**. The **End Time** dialog box appears (Figure 1-6).

Figure 1-6. End Time Dialog Box



2. In the **Time** box, specify the end time and select the time unit in the **Time** list.
3. Under **Default extension options**, in the **Extension value** list, select the value.
4. Under **End time extension per signal**, you can select specific extension values for each signal by clicking in the **Extension value** column.



The options in the **End time** dialog box are different settings than those under **Simulation period** in the **Settings** dialog box. Simulation period is the period that the Quartus II software simulates the stimuli. End time is the maximum length of time in the .vwf file. For information on the simulation period, refer to [Table 1-2 on page 1-13](#).

Arrange Group or Bus in LSB or MSB Order

You can arrange a group or bus in LSB or MSB order. If you arrange in LSB order, the LSB is on top and the MSB is at the bottom. If you arrange in MSB order, the MSB is on top and the LSB is at bottom.

To arrange a group or bus in LSB or MSB order, perform the following steps:

1. Select the bus that you want to change the LSB or MSB order. You can also select multiple buses in the waveform editor.
2. On the Edit menu, point to **Group and Bus Bit Order** and click either **MSB on top**, **LSB on Bottom** to change the bus or group in MSB order, or click **LSB on top**, **MSB on Bottom** to change the bus or group in LSB order.

Simulator Settings

You can enhance your output, reduce debugging time, and provide better coverage before running a simulation. This section covers the different simulation modes supported by the Quartus II simulator. Additionally, the Quartus II simulator offers common setup features like glitch filtering, setup and hold violation detection, and simulation coverage.

To set up simulation settings, perform the following steps:

1. On the Assignments menu, click **Settings**. The **Settings** dialog box appears.
2. In the **Category** list, select **Simulator Settings**. The **Simulator Settings** page appears.

Table 1-2 shows the options in the **Simulator Settings** page.

Table 1-2. Quartus II Simulator Settings (Part 1 of 2)

Settings and Options	Description
Simulation mode (1)	<p>Functional</p> <p>This simulation mode uses a pre-synthesis compiler database to simulate the logical performance of a project without the timing information. This mode enables you to check the functionality of the design. All nodes and buses are preserved in this simulation because functional simulation is performed before synthesis, partitioning, or fitting. A .vwf is required to perform this simulation mode.</p> <p>Timing</p> <p>This simulation mode uses the compiled netlist that includes timing information. With this simulation mode, you can check setup, hold violation, glitches, and simulation coverage. You can remove nodes or buses using the Quartus II Compiler when logic is optimized. This simulation mode uses the worst case timing model.</p> <p>Timing using Fast Timing Model</p> <p>This simulation mode is similar to timing simulation but this mode uses the best-case timing model.</p>
Simulation input	<p>You must include the vector file in the Simulation input box. You can type the name of the file or use the browse button to open the Select File dialog box. In the Files of type list, you can select Vector Waveform File (*.vwf), Compressed Vector Waveform File (*.cvwf), Value Change Dump File (*.vcd), Vector Table Output File (*.tbl), Vector Text File (*.vec), Simulation Channel File (.scf), or All Files (*.*).</p> <p>.tbl files contain input vectors and output logic levels in a tabular-format list. You can generate this file using a .vwf. However, if you would like to maintain, view, or update the vectors, .vwf files offer better visibility. .vwf or .tbl formats are interchangeable. You can generate .tbl files from .vwf files and vice versa. You can create a .vwf with the Waveform Editor. For more information about the Waveform Editor, refer to “Waveform Editor” on page 1-5.</p> <p>The Quartus II software also supports MAX+PLUS® II simulation vector files, such as .vec and .scf files.</p> <p>A .cvwf is the simplified version, non-readable format of the .vwf format. This file type is in binary format and is generally smaller in file size. You can use .cvwf files in the Waveform Editor and simulation.</p> <p>A .vcd is an ASCII file that contains header information, variable definitions, and the value changes for specified variables, or all variables, in a given design. The value changes for a variable are given in scalar or vector format based on the nature of the variable.</p>

Table 1-2. Quartus II Simulator Settings (Part 2 of 2)

Settings and Options	Description
Simulation period	The simulation period determines the length of time that the simulator runs the stimuli with the maximum period being equal to the end time of a .vwf . If the simulation period is configured shorter than the end time, all signals beyond the simulation period are displayed as Unknown (X). Therefore, you can also shorten the simulation period or end the simulation earlier by selecting End Simulation at and specifying the time and selecting the time unit. If the simulation period is configured longer than the end time, the simulation will stop at the end time. For information on the end time, refer to “ End Time ” on page 1-11.
Glitch filtering options	Specifies whether to enable glitch filtering for simulations. You can select one of the following options: Auto —The simulator performs glitch filtering when .saf generation is enabled in the Simulation Output Files page of the Settings dialog box. Always —The simulator always performs glitch filtering, even if .saf generation is not enabled. Never —The simulator never performs glitch filtering, even if .saf generation is enabled.
More Settings	If you click More Settings , the More Simulator Settings dialog box appears. The following options are available under Existing option settings . Cell Delay Model Type Specifies the type of delay model to be used for cell delays: transport or inertial. The default is transport. Interconnect Delay Model Type Specifies the type of delay model to be used for interconnect delays: transport or inertial. The default is transport. Preserve fewer signal transition to reduce memory requirements This option is effective on lower performance workstations because turning on this option flushes signal transitions from memory to disk for memory optimization.

Note to Table 1-2:

- (1) The Quartus II simulator may flag an error message if zero-time oscillation occurs in your design. Zero-time oscillation occurs when a particular output signal does not achieve a stable output value at a particular fixed time, which may be due to your design containing combinational logic path loops.

Simulation Verification Options

Table 1-3 shows the options in the **Simulation Verification** page.

Table 1-3. Quartus II Simulation Verification

Settings and Options	Description
Check outputs	<p>Check outputs checks expected outputs against actual outputs in the simulation report. After turning on Check outputs, click the Waveform Comparison Settings button. The Waveform Comparison Settings dialog box appears.</p> <p>In the Waveform Comparison Settings dialog box, you can specify the waveform comparison time frame and the comparison options. You can also set the tolerance level for all the signals by specifying the tolerance limit in the Default comparison timing tolerance box. The Maximum comparison mismatches box is the amount of mismatches the Quartus II simulator is allowed to accept before it stops comparing.</p> <p>You can also set the type of transition the comparison should trigger in the Waveform Comparison Settings dialog box. You can assign trigger comparisons based on Input signal transition edges, All signal transition edges, or Selected Signal transition edges.</p> <p>To customize the waveform comparison matching rules, you can also click the Comparison Rules button. The Comparison Rules dialog box appears, allowing you to customize the comparison matching rules.</p>
Setup and hold time violation detection	<p>This option detects setup and hold time violation. Setup time is the period required by a synchronous signal to stabilize before the arrival of a clock edge. Hold time is the time required by a synchronous signal to maintain after the same clock edge. If the Setup and hold time violation detection option is turned on, a warning in the Messages window appears if any setup or hold time violation is detected during the simulation. This option is only for Timing and Timing using Fast Timing Model simulation modes.</p>
Glitch detection	<p>Conditions occur when two or more signals toggle simultaneously and can cause glitches or unwanted short pulses. The Glitch detection option enables you to detect glitches and specify the time interval that defines a glitch. If two logic level transitions occur in a period shorter than the specified time period, the resulting glitch is detected and reported in the Processing tab of the Messages window.</p> <p>If you turn on the Glitch detection option, you can specify the acceptable glitch width. A Messages window appears when a pulse is smaller than the specified glitch width that is detected. The Glitch detection option is only available for Timing and Timing using Fast Timing Model simulation modes.</p>
Simulation coverage reporting	<p>This option reports the ratio of outputs (coverage) actually simulated to the number of outputs in the netlist and is expressed as a percentage. When you turn on the Simulation coverage reporting option, the Report Settings button is available. If you click Report Settings, the Report Settings dialog box appears. The three types of coverage reports you can select from are Display complete 1/0 value coverage report, Display missing 1-value coverage report, and Display missing 0-value coverage report.</p>
Disable setup and hold time violation detection for input registers of bi-directional pins	<p>This option enables you to disable setup and hold time violation detection in input registers of all bidirectional pins in the simulated design during Timing or Timing using Fast Timing Model simulation.</p>

Simulation Output Files Options

Table 1-4 shows the options in the **Simulation Output Files** page.

Table 1-4. Quartus II Simulation Output Files

Setting and Options	Description
Simulation output waveform	<p>Specify the simulation output waveform options.</p> <p>Automatically add pins to simulation output waveforms</p> <p>This option automatically adds all outputs that are available in the design to the waveform reports. If your design has large amounts of outputs, turning on this option ensures all outputs are monitored during simulation.</p> <p>Overwrite simulation input file with simulation results</p> <p>This option overwrites the vector source file with simulation results. This option is ignored when the Check outputs setting is turned on. This option adds the result to the vector file and generally, it can give you more visibility during the debugging process. (1)</p> <p>Group bus channel in simulation results</p> <p>This option automatically groups bus channels in the output waveform that are shown in the simulation reports. By turning off this option, all output waveforms have a node to represent each bus signal.</p>
Signal activity output for power analysis	<p>When you perform your simulation with the Quartus II simulator, you can generate a .saf, which is used by the PowerPlay Power Analyzer to assist you with power analysis. (2), (3)</p>
VCD output for power analysis	<p>When you perform simulation with the Quartus II simulator, you can generate a .vcd, which is used by the PowerPlay Power Analyzer to assist you with power analysis. (2), (3)</p>

Notes to Table 1-4:

- (1) A backup copy of the source vector file is saved under the **db** folder with the name `<project>.sim_ori.<vector file format type>`.
- (2) Instead of using the **.saf** or Generate **.vcd** (***.vcd**), you can also save your output waveform as a **.vcd** to perform power analysis.
- (3) For more information about the PowerPlay Power Analyzer, refer to the *PowerPlay Power Analysis* chapter in volume 3 of the *Quartus II Handbook*.

Simulation Report

Comprehensive reports are shown after the completion of each simulation. These reports are important to ensure designs meet timing and logical correctness. These simulation reports also play an important role in debugging.

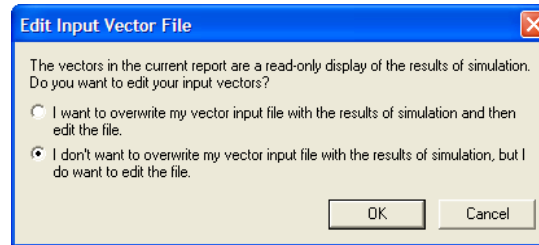
Simulation Waveform

Simulation Waveforms are part of the Simulation report. In this report, the stimuli and the results of the simulation are displayed.

You can export the simulation waveform as a VHDL Test Bench File or a Verilog Test Bench File for use in other EDA tools. You can also save a simulation as a **.vwf** file or Vector Table Output File for use with the Quartus II software.

When you try to edit the Simulation Waveform, the **Edit Input Vector File** dialog box appears, asking whether you would like to edit the vector input file with the results of the simulation or if you would like to overwrite the vector input file with other vector inputs (refer to Figure 1-7).

Figure 1-7. Edit Input Vector File



You can overwrite your simulation input file with the simulation results so that your input vector file is updated with the resulting waveform after a simulation. For more information, refer to the **Overwrite simulation input file with simulation results** option in [Table 1-2 on page 1-13](#).

If you do not want to overwrite the simulation input file in every simulation run, perform the following to overwrite simulation input files with simulation results after a simulation:

On the Processing Menu, point to **Simulation Debug** and click **Overwrite Vector Inputs with Simulation Outputs**.

Simulating Bidirectional Pin

A bidirectional pin is represented in the waveform by two channels. One channel represents the input to the bidirectional pin, and the other channel represents the output from the bidirectional pin. You can enter the input channel into the waveform by using the **Node Finder** dialog box. The output channel is automatically created by the Quartus II simulator and named `<bidir pin name> ~result`.

Logical Memories Report

The Quartus II software writes out the contents of each memory module after simulation. Therefore, if you use memory cells in your design, you can analyze the contents of the logic memory structures in the device in the Logical Memories Report. The Logical Memories Report displays individual reports for each memory block and contains the data stored in the memory cell used at the end of simulation.

After being simulated, a memory module's contents are stored in the Logical Memories section of the simulation report file.

To view this section, perform the following steps:

1. On the Processing menu, click **Simulation Report**. The Simulation Report window appears.
2. In the report window, click the "+" next to **Logical Memories**.

Simulation Coverage Reports

The **Coverage Summary** report contains the following summary information for the simulation:

- Total toggling coverage as a percentage
- Total nodes checked in the design
- Total output ports checked
- Total output ports with complete 1/0-value coverage
- Total output ports with no 1/0-value coverage
- Total output ports with no 1-value coverage
- Total output ports with no 0-value coverage

The **Complete 1/0-Value Coverage** report lists the following information:

- Node name
- Output port name
- Output port type for output ports that toggle between 1 and 0 during the simulation

The **Missing 0-Value Coverage** report and **Missing 1-Value Coverage** report list the following information:

- Node name
- Output port name
- Output port type for output ports that do not toggle to the designated value

For more information about Simulation Coverage reports, refer to the **Simulation coverage reporting** option in [Table 1-2 on page 1-13](#).

The following are individual reports and their definitions:

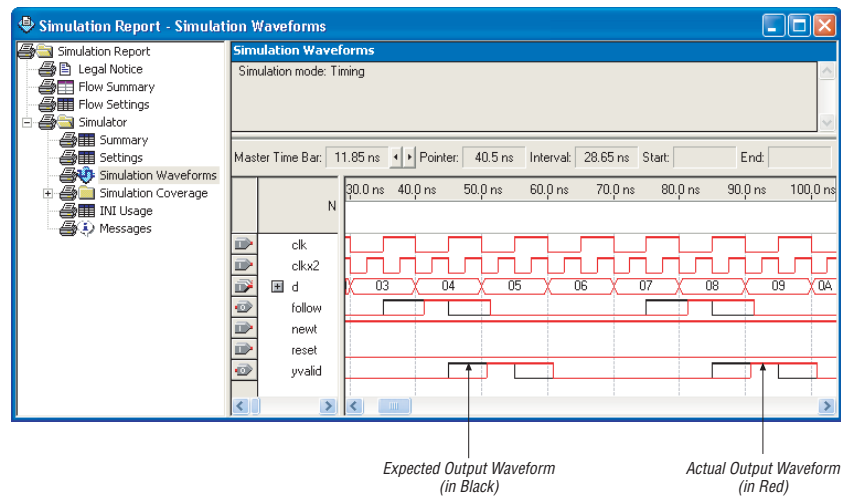
- **Complete 1/0 value coverage report**—Displays all the nodes or buses that toggle between 1 and 0 during simulation.
- **Missing 1-value coverage** and **Missing 0-value coverage reports**—Displays all the nodes that do not toggle to the designated value.

Comparing Two Waveforms

You can compare your simulation results against previous simulations using the compare option. To compare two waveforms in the Simulation Report, turn on the **Check outputs** option. For more information about the **Check outputs** option, refer to [Table 1-2 on page 1-13](#). With the **Check outputs** option turned on, the two comparable waveforms are visible in black and red. The black waveforms represent the original output or the expected output, and the red waveforms represent the compared output or the actual output.

Figure 1-8 shows an example of expected output waveform versus actual output waveform.

Figure 1-8. Example of Simulation Waveform from the Simulation Report When Check Output is Turned On



Debugging with the Quartus II Simulator

The Quartus II software includes tools to help with simulation debugging. This section covers some debugging tools and their use.

Breakpoints

Inserting breakpoints into the simulation process enables the simulator to break at the desired time or on the desired node or bus condition. You can monitor the activity of nodes or buses during specified times and pinpoint the cause of mismatched signal levels between expected and actual. To use breakpoints, perform the following steps:

1. On the Processing menu, point to **Simulation Debug** and click **Breakpoints**. The **Breakpoints** dialog box appears.
2. In the Equation text box, click **condition**. You can configure the logical conditions of individual nodes or buses, or you can set the time.
3. After you configure the equation conditions, select the action for the Quartus II simulator. In the **Action** pull-down list, select **Stop**, **Warning Message**, **Error Message**, or **Information Message**. This selection defines the action when the condition is met.
4. You can also enter the text that appears when the simulator encounters the breakpoint. If you do not make an entry in this box, the Quartus II software displays a default message.

Updating Memory Content

If your design includes memories, when the simulator stops at a breakpoint, you can view and edit the contents of the memories. To view your memories during a breakpoint in the simulation, on the Processing menu, point to **Simulation Debug** and click **Embedded Memory**.

Last Simulation Vector Outputs

The **Last Simulation Vector Outputs** command opens the Output Simulation Waveforms report generated by the last simulation. To use this command, on the Processing menu, point to **Simulation Debug** and click **Last Simulation Vector Outputs**.

You can open the current input vectors that you defined in the **Simulator Settings** dialog box with the **Current Vector Inputs** command. To use this command, on the Processing menu, point to **Simulation Debug** and click **Current Vector Inputs**. Lastly, you can overwrite the vector source file with the simulation outputs that open the resulting file.

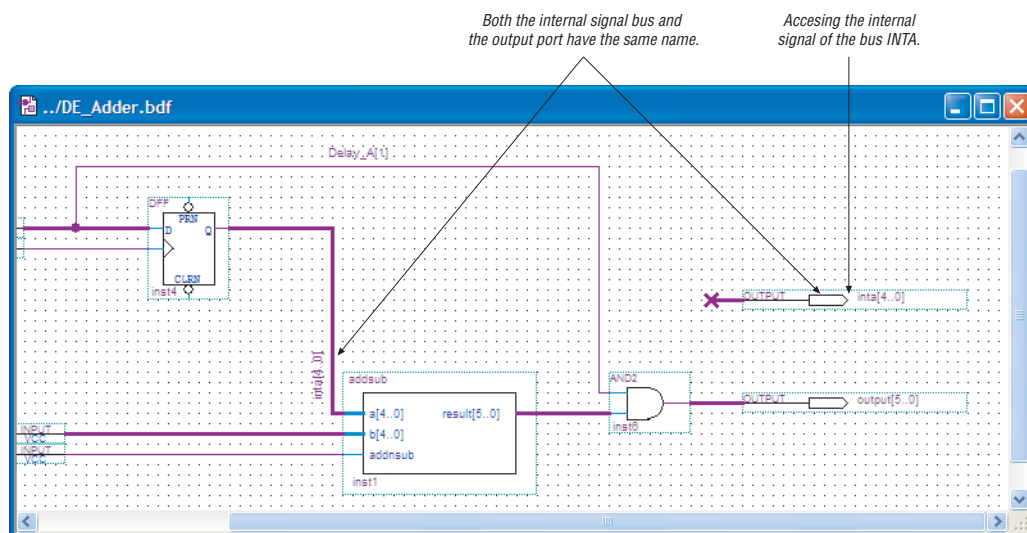
Conventional Debugging Process

During the design phase, tapping out internal signals is a common practice to debug simulation errors. Therefore, the Quartus II software enables you to tap out the signal for simulation debug and also enables you to pull out the internal signal to the physical I/O. The Quartus II software also offers SignalTap II and SignalProbe to further assist you with debugging.

Accessing Internal Signals for Simulation

You can conventionally debug by probing out the internal signals, which enables you to preserve the internal signals during synthesis. You can probe the internal signal by selecting the node or bus and specifying a name, and then adding an output port to the schematic with a similar name. **Figure 1-9** shows an example of accessing internal signals for simulation from a schematic diagram.

Figure 1-9. Example of Tapping Out Internal Signal



For timing simulations, the simulation netlist is based on the Compilation post-Synthesis and post-Fitting netlist. Therefore, some of the internal nodes or buses are optimized away during compilation of the netlist. If an internal node is optimized away, the Quartus II software displays a warning message similar to the following in the **Warning** tab of the Messages window:

Warning: Compiler packed, optimized or synthesized away node "DataU". Ignored vector source file node.

This internal node is ignored by the Quartus II simulator.


If you would like to tap out the D and Q ports of registers, turn on **Add D and Q ports of register node to Simulation Output Waveform** from the Assignment Editor. This feature is only available for functional simulations.

Scripting Support

You can run procedures and make settings described in this chapter in a Tcl script. You can also run some procedures at a command prompt. For detailed information about scripting command options, refer to the Quartus II Command-Line and Tcl API Help browser. To run the Help browser, type the following at the command prompt:

```
quartus_sh --qhelp ←
```

The *Quartus II Scripting Reference Manual* includes the same information in PDF form.

 For more information about Tcl scripting, refer to the *Tcl Scripting* chapter in volume 2 of the *Quartus II Handbook*. For information about all settings and constraints in the Quartus II software, refer to the *Quartus II Settings File Manual*. For more information about command-line scripting, refer to the *Command-Line Scripting* chapter in volume 2 of the *Quartus II Handbook*.

You can change the Functional, Timing, or Timing using Fast Timing Model simulation modes by typing the following at the command prompt:

```
simulation_mode <mode> ←
```

To initialize the simulation for the current design, use the following command. During initialization, the simulator builds the simulation netlist and sets the simulation time to zero.

The option `-ignore_vector_file` is set to **Off** by default when the source vector file exists for simulation. The Quartus II software ignores the source vector file during simulation if the option `-ignore_vector_file` is set to **On**. The `-end_time` option is used only when the `-ignore_vector_file` option is set to **On**.

```
initialize_simulation [-h | -help] [-long_help] [-check_outputs <On | Off>] \  
[-end_time <end_time>] [-glitch_filtering <On | Off>] [-ignore_vector_file <On | Off>] \  
[-memory_limiter <On | Off>] [-power_vcd_output <target_file>] \  
[-read_settings_files <On | Off>] [-saf_output <target_file>] \  
[-sim_mode <functional | timing | timing_using_fast_timing_model >] \  
[-vector_source <vector_source_file>] [-write_settings_files <On | Off>] \  
-simulation_results_format <.vwf | .cvwf | .vcd> -vector_source <vector source file>
```

To force the specified signal or group of signals to the specified value, type the following at the command prompt:

```
force_simulation_value [-h | -help] [-long_help] -node <hpath> <value> ←
```

To turn on the simulator to simulate the design for a specified time, type the following at a command prompt:

```
run_simulation [-h | -help] [-long_help] [-time <time>] ←
```



If you do not set a specific length of time for the simulation run, it runs a complete simulation.

To create a breakpoint with a specified equation and action, type the following at the command prompt:

```
create_simulation_breakpoint [-h | -help] [-long_help] \  
-action [Give Warning | Give Info | Give Error] \  
-breakpoint <breakpoint_name> -equation <equation> [-user_message <message_text>]↵
```

To delete a breakpoint with a specified name, type the following at the command prompt:

```
delete_simulation_breakpoint [-h | -help] [-long_help] -breakpoint <breakpoint_name> ↵
```

Conclusion

Simulation plays an important role in ensuring the quality of a product. The Quartus II software offers various tools to assist you with simulation and helps reduce debugging time with the introduction of features like Glitch Filtering and Breakpoints.

Referenced Documents

This chapter references the following documents:


- *Analyzing Designs with the Quartus II Netlist Viewers* chapter in volume 1 of the *Quartus II Handbook*
- *Command-Line Scripting* chapter in volume 2 of the *Quartus II Handbook*
- *Mentor Graphics ModelSim Support* chapter in volume 3 of the *Quartus II Handbook*
- *PowerPlay Power Analysis* chapter in volume 3 of the *Quartus II Handbook*
- *Quartus II Scripting Reference Manual*
- *Quartus II Settings File Manual*
- *Section I: Simulation* section in volume 3 of the *Quartus II Handbook*
- *Tcl Scripting* chapter in volume 2 of the *Quartus II Handbook*

Document Revision History

Table 1-5 shows the revision history for this chapter.

Table 1-5. Document Revision History

Date and Document Version	Changes Made	Summary of Changes
November 2009 v9.1	<ul style="list-style-type: none"> ■ Updated "Introduction" on page 1-1. 	Updated for the Quartus II 9.1 software release.
March 2009 v9.0	<ul style="list-style-type: none"> ■ No change to content. 	—
November 2008 v8.1	<ul style="list-style-type: none"> ■ Changed to 8½" × 11" page size. No change to content. 	—
May 2008 v8.0.0	<ul style="list-style-type: none"> ■ Updated "Introduction" on page 1-1. ■ Updated "Referenced Documents" on page 1-27. 	Updated for the Quartus II 8.0 software release

 For previous versions of the *Quartus II Handbook*, refer to the [Quartus II Handbook Archive](#).

