Purpose

ModelSim is a third-party software program used to simulate digital designs. ModelSim - Altera Starter Edition is packaged with Quartus II and can be linked to it to provide commercial-grade simulation. This tutorial will show you how to use ModelSim to simulate a VHDL design. It is assumed that you know how to enter and compile a design in Quartus II.

Sample VHDL Design

Create a project in Quartus II and enter the following VHDL code in the Quartus Text Editor. Compile the design after it has been entered.

```
-- truth_table_1.vhd
-- Output y goes HIGH when any of the following values of d[3..0] are present:
--          0011, 0110, 1001, 1101

library ieee;
use ieee.std_logic_1164.all;

entity truth_table_1 is
    port(
        d :  in std_logic_vector(3 downto 0);
        y : out std_logic
        );
end truth_table_1;

architecture tt of truth_table_1 is
begin
    with d select
        y    <=   '1' when "0011",
                  '1' when "0100",
                  '1' when "1001",
                  '1' when "1101",
                  '0' when others;
end tt;
```

Quartus II Setup

The following steps will enable the Quartus II design to compile more efficiently and to use the ModelSim simulator tool.

Set up file path for ModelSim:

Tools > Options > EDA Tool Options > ModelSim-Altera (See Figure 2 **and** Figure 3**).**

Browse to Quartus file path

e.g., for Altera version 12.1, service pack 1 (12.1sp1):

C:\altera\12.1sp1\modelsim_ase\win32aloem

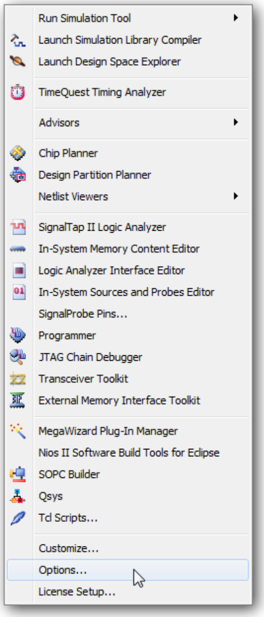
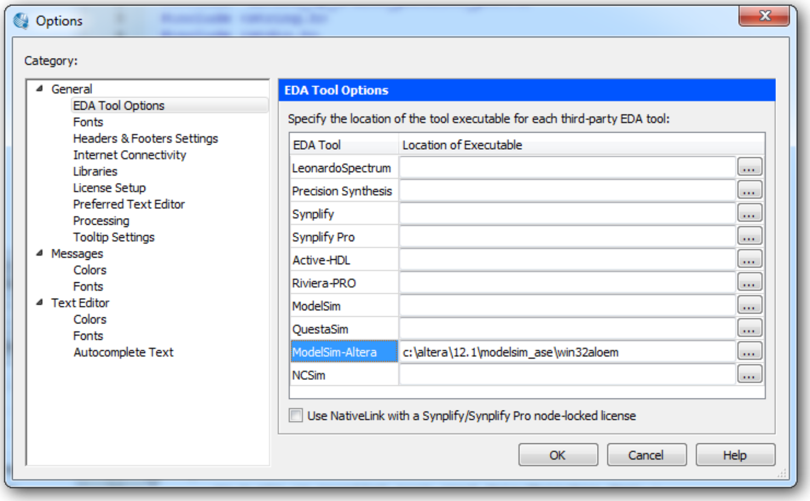

Run Simulation Tool
Launch Simulation Library Compiler
Launch Design Space Explorer
TimeQuest Timing Analyzer
Advisors
Chip Planner
Design Partition Planner
Netlist Viewers
SignalTap II Logic Analyzer
In-System Memory Content Editor
Logic Analyzer Interface Editor
In-System Sources and Probes Editor
SignalProbe Pins...
Programmer
JTAG Chain Debugger
Transceiver Toolkit
External Memory Interface Toolkit
MegaWizard Plug-In Manager
Nios II Software Build Tools for Eclipse
SOPC Builder
Qsys
Tcl Scripts...
Customize...
Options...
License Setup...

Figure 1 Quartus II Tools Menu



EDA Tool Options

Specify the location of the tool executable for each third-party EDA tool:

EDA Tool	Location of Executable
LeonardoSpectrum	
Precision Synthesis	
Synplify	
Synplify Pro	
Active-HDL	
Riviera-PRO	
ModelSim	
QuestaSim	
ModelSim-Altera	c:\altera\12.1\modelsim_ase\win32aloem
NCSim	

□ Use NativeLink with a Synplify/Synplify Pro node-locked license

Figure 2 Set Path to ModelSim

Select the **Assignments** menu, then **Settings**, shown in Figure 1.

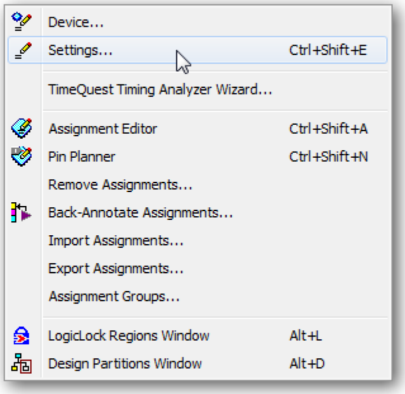
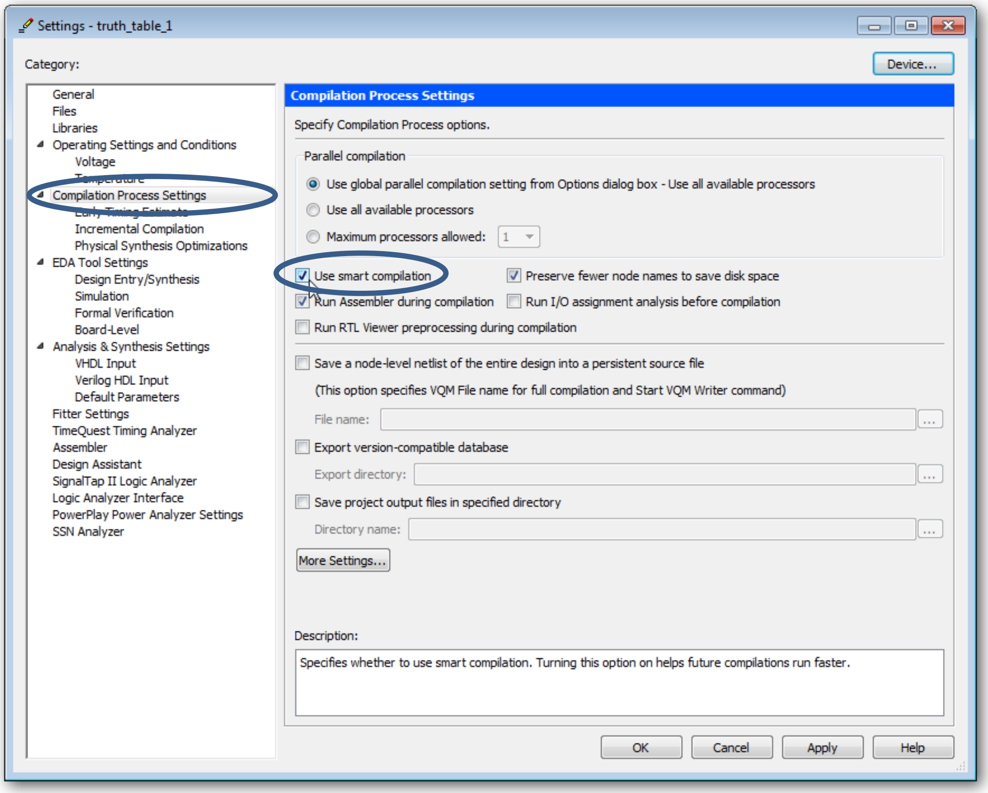

Device...
Settings... Ctrl+Shift+E
TimeQuest Timing Analyzer Wizard...
Assignment Editor Ctrl+Shift+A
Pin Planner Ctrl+Shift+N
Remove Assignments...
Back-Annotate Assignments...
Import Assignments...
Export Assignments...
Assignment Groups...
LogicLock Regions Window Alt+L
Design Partitions Window Alt+D

Figure 3 Assignments Menu

Three things will be set up in the **Settings** dialog box: **Smart Compilation**, **EDA Tool Settings**, and **Simulation** (a subheading of **EDA Tool Settings**).

Smart Compilation - **Category: Compilation Process Settings:** Check **Use Smart Compilation**



Compilation Process Settings

Specify Compilation Process options.

Parallel compilation

● Use global parallel compilation setting from Options dialog box - Use all available processors

○ Use all available processors

○ Maximum processors allowed: 1

☑ Use smart compilation ☑ Preserve fewer node names to save disk space

☑ Run Assembler during compilation ☐ Run I/O assignment analysis before compilation

☐ Run RTL Viewer preprocessing during compilation

☐ Save a node-level netlist of the entire design into a persistent source file

(This option specifies VQM File name for full compilation and Start VQM Writer command)

File name:

☐ Export version-compatible database

Export directory:

☐ Save project output files in specified directory

Directory name:

More Settings...

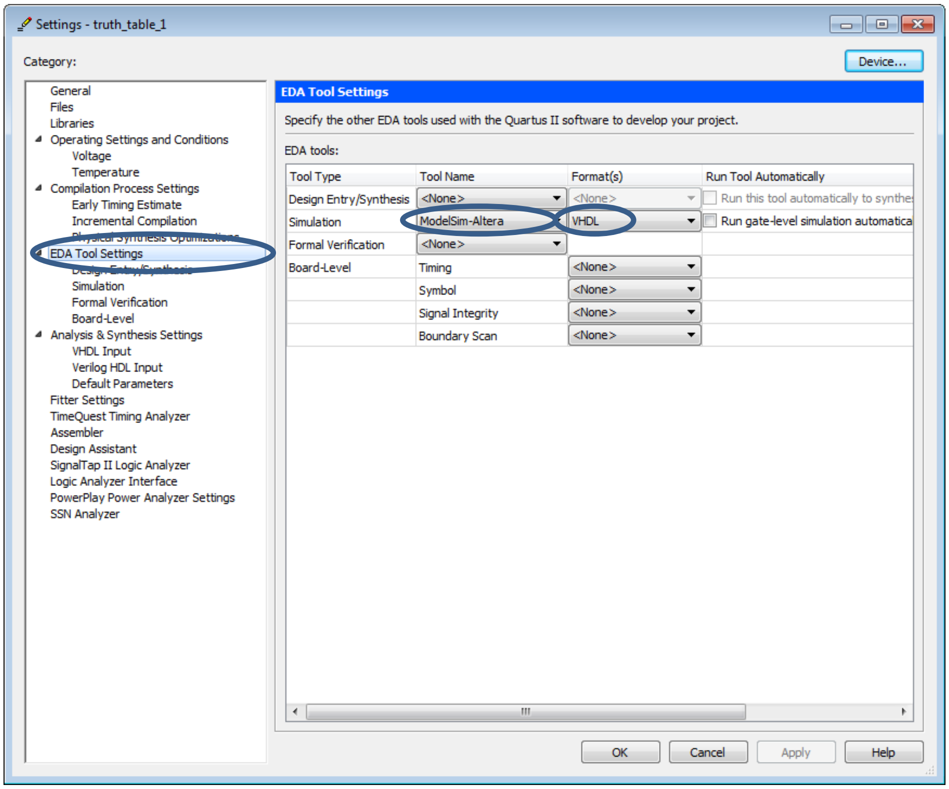
Description:

Specifies whether to use smart compilation. Turning this option on helps future compilations run faster.

Figure 4 Smart Compilation (Speeds up Future Complications of the Design)

EDA Tool Settings - Select the following:

Category: EDA Tool Settings: Tool Name - ModelSim - Altera; **Format(s) -** VHDL



Category:

- General
- Files
- Libraries
- Operating Settings and Conditions
  - Voltage
  - Temperature
- Compilation Process Settings
  - Early Timing Estimate
  - Incremental Compilation
  - Physical Synthesis Optimizations
- EDA Tool Settings
  - Design Entry/Synthesis
  - Simulation
  - Formal Verification
  - Board-Level
- Analysis & Synthesis Settings
  - VHDL Input
  - Verilog HDL Input
  - Default Parameters
- Fitter Settings
- TimeQuest Timing Analyzer
- Assembler
- Design Assistant
- SignalTap II Logic Analyzer
- Logic Analyzer Interface
- PowerPlay Power Analyzer Settings
- SSN Analyzer

EDA Tool Settings

Specify the other EDA tools used with the Quartus II software to develop your project.

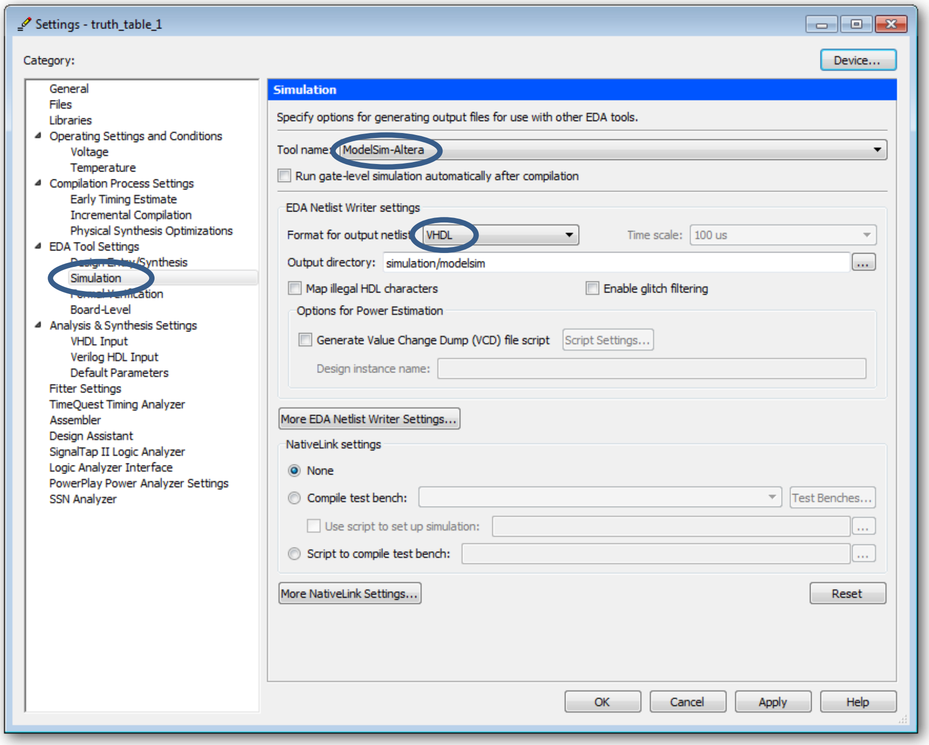
EDA tools:

Tool Type	Tool Name	Format(s)	Run Tool Automatically
Design Entry/Synthesis	<None>	<None>	□ Run this tool automatically to synthes
Simulation	ModelSim-Altera	VHDL	□ Run gate-level simulation automatica
Formal Verification	<None>		
Board-Level	Timing	<None>	
	Symbol	<None>	
	Signal Integrity	<None>	
	Boundary Scan	<None>	

Figure 5 EDA (Electronic Design Automation) Tools Settings

Simulation:

Category: EDA Tool Settings, Simulation: Tool name: ModelSim-Altera; **Format for output netlist:** VHDL



Settings - truth_table_1

Category:

- General
- Files
- Libraries
- Operating Settings and Conditions
  - Voltage
  - Temperature
- Compilation Process Settings
  - Early Timing Estimate
  - Incremental Compilation
  - Physical Synthesis Optimizations
- EDA Tool Settings
  - Design Entry/Synthesis
  - Simulation
  - Formal Verification
  - Board-Level
- Analysis & Synthesis Settings
  - VHDL Input
  - Verilog HDL Input
  - Default Parameters
- Fitter Settings
- TimeQuest Timing Analyzer
- Assembler
- Design Assistant
- SignalTap II Logic Analyzer
- Logic Analyzer Interface
- PowerPlay Power Analyzer Settings
- SSN Analyzer

Simulation

Specify options for generating output files for use with other EDA tools.

Tool name: ModelSim-Altera

□ Run gate-level simulation automatically after compilation

EDA Netlist Writer settings

Format for output netlist: VHDL Time scale: 100 us

Output directory: simulation/modelsim

□ Map illegal HDL characters □ Enable glitch filtering

Options for Power Estimation

□ Generate Value Change Dump (VCD) file script

Design instance name:

NativeLink settings

● None

○ Compile test bench:

□ Use script to set up simulation:

○ Script to compile test bench:

Figure 6 EDA SImulation Settings

Starting ModelSim in Quartus II

Either click the button for **Gate Level Simulation**, circled in Figure 5, or from the Quartus II **Tools** menu, select **Run Simulation Tool, Gate Level Simulation…**

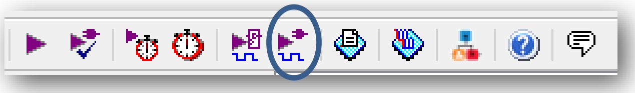
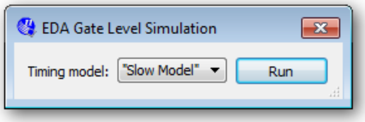


Figure 7 Quartus II Processing Toolbar

The dialog in Figure 6 will open. Select **"Slow Model"** and click **Run**.



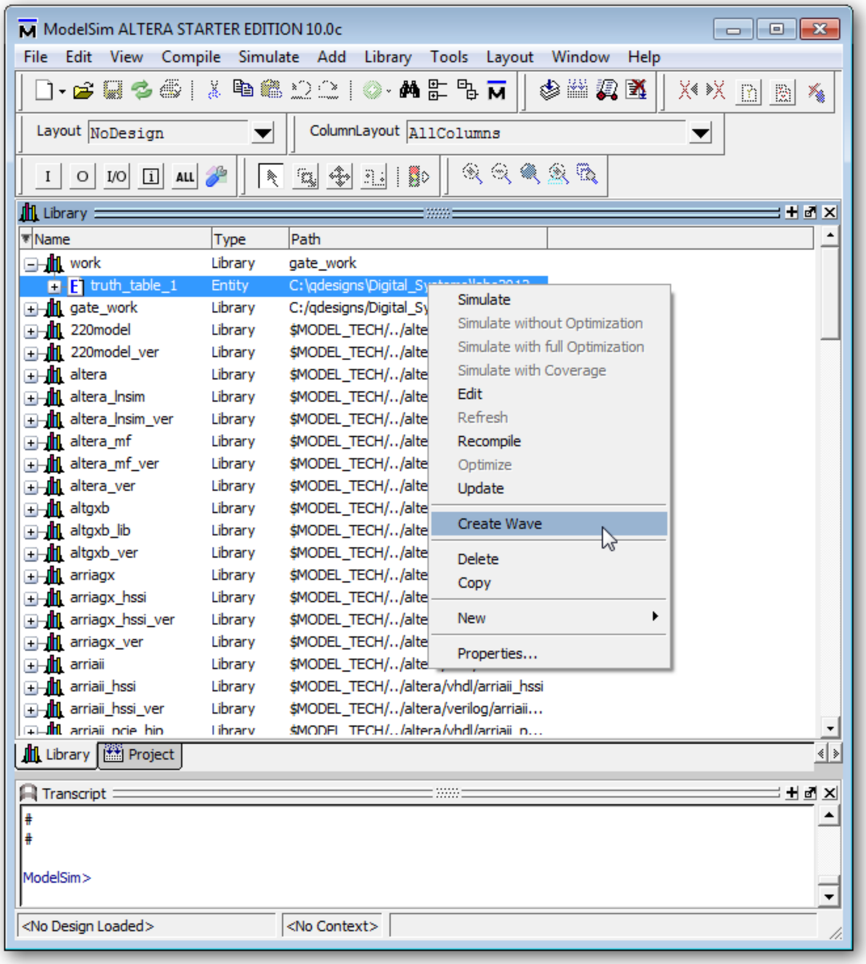
EDA Gate Level Simulation

Timing model: "Slow Model"

Figure 8 EDA Gate Level Simulation

Creating an input waveform in ModelSim

When ModelSim opens, you will see a **Wave** window and a **Library** window, which is shown in Figure 7. Ignore the **Wave** window for now. Expand the **work** library by clicking on the + sign in front of the word **work**. Select the entity labeled **truth_table_1**, right click it, and choose **Create Wave** from the popup menu. The **Wave** window will now include the signals for **d** and **y**, as shown in Figure 8.



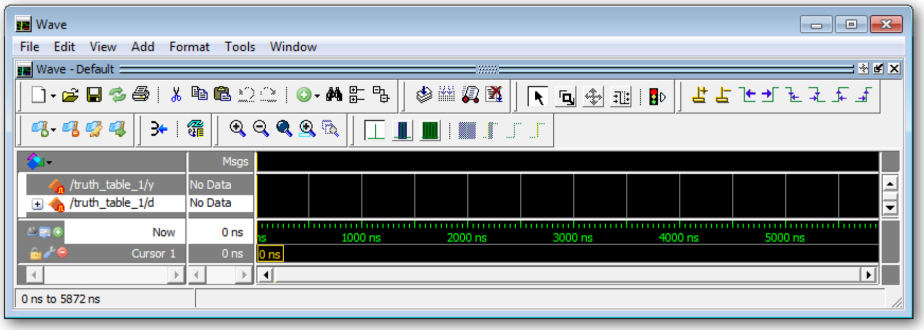
Name	Type	Path
work	Library	gate_work
truth_table_1	Entity	C:\qdesigns\Digital_Sy
gate_work	Library	C:/qdesigns/Digital_Sy
220model	Library	$MODEL_TECH/../alte
220model_ver	Library	$MODEL_TECH/../alte
altera	Library	$MODEL_TECH/../alte
altera_lnsim	Library	$MODEL_TECH/../alte
altera_lnsim_ver	Library	$MODEL_TECH/../alte
altera_mf	Library	$MODEL_TECH/../alte
altera_mf_ver	Library	$MODEL_TECH/../alte
altera_ver	Library	$MODEL_TECH/../alte
altgxb	Library	$MODEL_TECH/../alte
altgxb_lib	Library	$MODEL_TECH/../alte
altgxb_ver	Library	$MODEL_TECH/../alte
arriagx	Library	$MODEL_TECH/../alte
arriagx_hssi	Library	$MODEL_TECH/../alte
arriagx_hssi_ver	Library	$MODEL_TECH/../alte
arriagx_ver	Library	$MODEL_TECH/../alte
arriaii	Library	$MODEL_TECH/../alte
arriaii_hssi	Library	$MODEL_TECH/../altera/vhdl/arriaii_hssi
arriaii_hssi_ver	Library	$MODEL_TECH/../altera/verilog/arriaii...
arriaii_pcie_hip	Library	$MODEL_TECH/../altera/vhdl/arriaii_p...

```
#
#

ModelSim>
```

<No Design Loaded> <No Context>

Figure 9 ModelSim Library Window

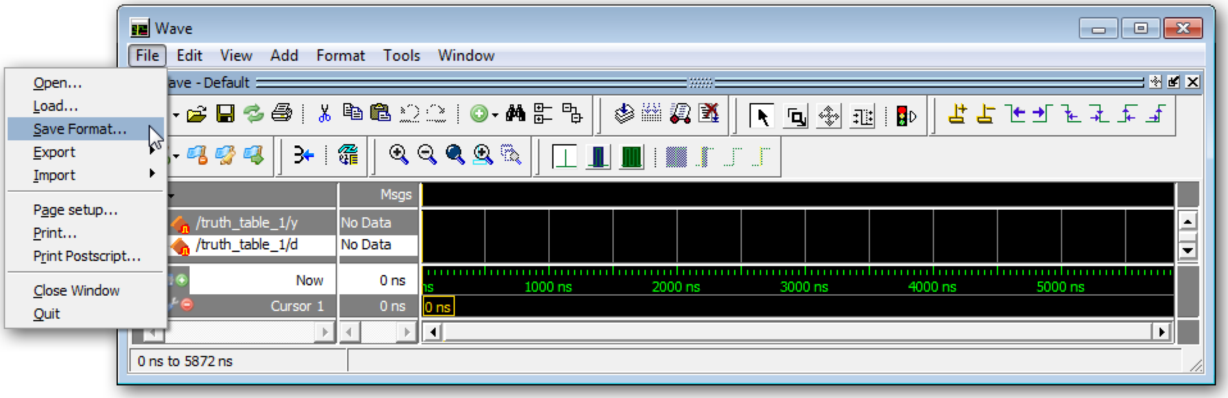


	Msgs
/truth_table_1/y	No Data
/truth_table_1/d	No Data
Now	0 ns
Cursor 1	0 ns

0 ns to 5872 ns

Figure 10 Waveform View with Signals Added (no waveform data yet)

Save the **Wave** window by selecting **Save Format** from the file menu, as shown in Figure 9.

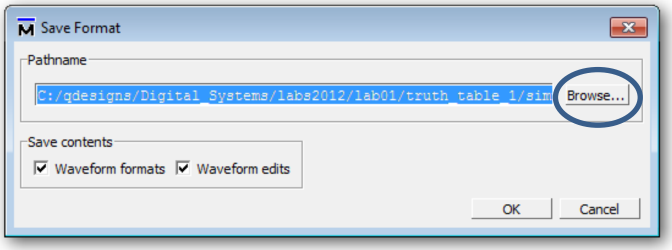


	Msgs
/truth_table_1/y	No Data
/truth_table_1/d	No Data
Now	0 ns
Cursor 1	0 ns

0 ns to 5872 ns

Figure 11 Saving the Wave File

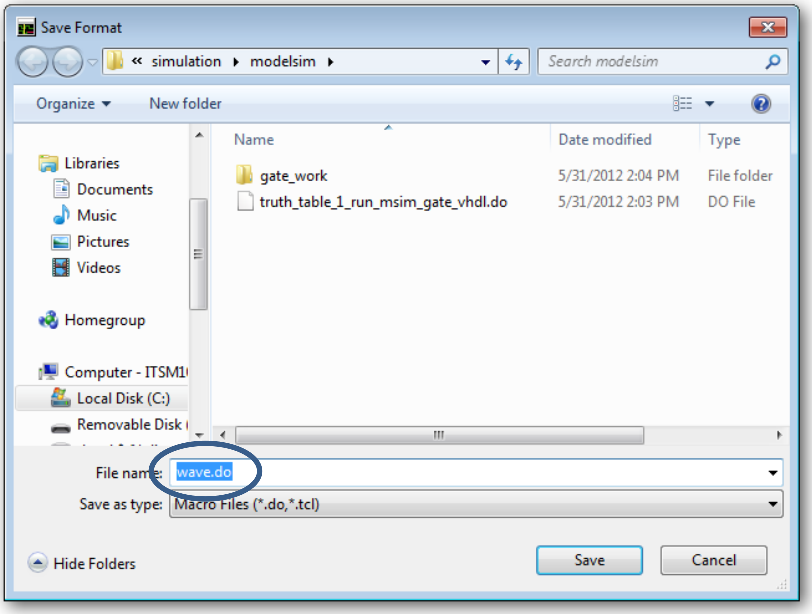
In the dialog box shown in Figure 10, click **Browse**. The box in Figure 11 will open and suggest that you save the file with the name **wave.do**. It is not necessary to select this name, but for now, do so by clicking **Save**. Overwrite the existing file if it asks you to do so. You will return to Figure 10; close that box by clicking **OK**.



Pathname: C:/qdesigns/Digital_Systems/labs2012/lab01/truth_table_1/sim

Save contents: ☑ Waveform formats ☑ Waveform edits

Figure 12 Save Format Dialog Box



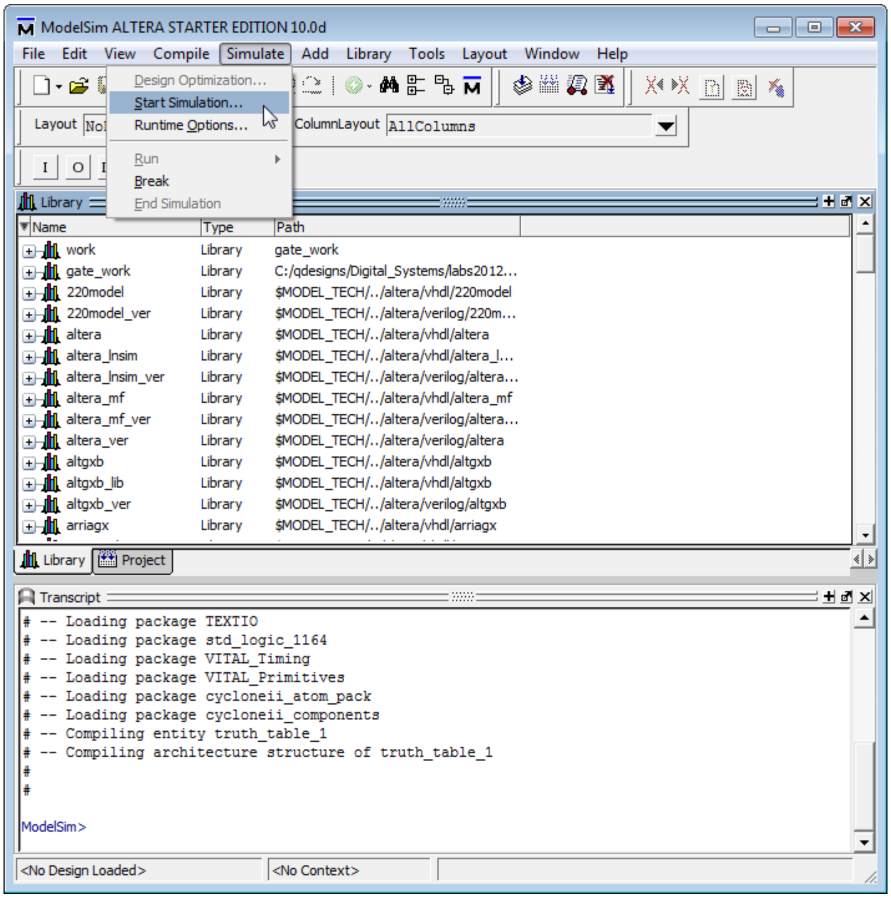
Name	Date modified	Type
gate_work	5/31/2012 2:04 PM	File folder
truth_table_1_run_msim_gate_vhdl.do	5/31/2012 2:03 PM	DO File

File name: wave.do

Save as type: Macro Files (*.do,*.tcl)

Figure 13 Saving the Wave Format

From the ModelSim window, select the **Simulate** menu, then **Start Simulation**, as shown in Figure 12. The dialog box in Figure 13 will open. Select the **truth_table_1** entity from the **work** library and click **OK**.

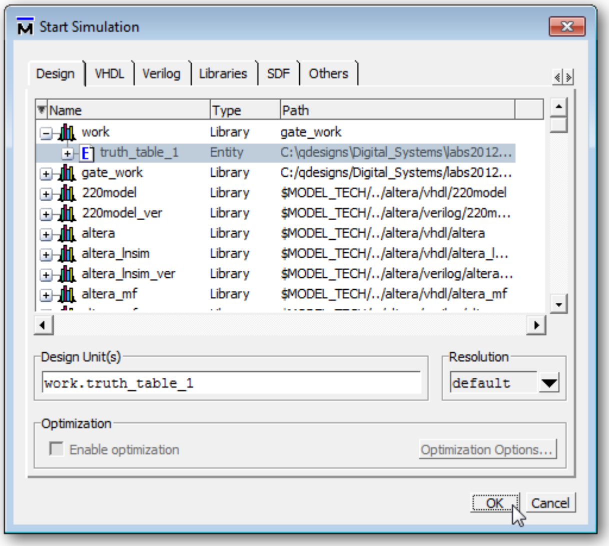


Name	Type	Path
work	Library	gate_work
gate_work	Library	C:/qdesigns/Digital_Systems/labs2012...
220model	Library	$MODEL_TECH/../altera/vhdl/220model
220model_ver	Library	$MODEL_TECH/../altera/verilog/220m...
altera	Library	$MODEL_TECH/../altera/vhdl/altera
altera_lnsim	Library	$MODEL_TECH/../altera/vhdl/altera_l...
altera_lnsim_ver	Library	$MODEL_TECH/../altera/verilog/altera...
altera_mf	Library	$MODEL_TECH/../altera/vhdl/altera_mf
altera_mf_ver	Library	$MODEL_TECH/../altera/verilog/altera...
altera_ver	Library	$MODEL_TECH/../altera/verilog/altera
altgxb	Library	$MODEL_TECH/../altera/vhdl/altgxb
altgxb_lib	Library	$MODEL_TECH/../altera/vhdl/altgxb
altgxb_ver	Library	$MODEL_TECH/../altera/verilog/altgxb
arriagx	Library	$MODEL_TECH/../altera/vhdl/arriagx

```
# -- Loading package TEXTIO
# -- Loading package std_logic_1164
# -- Loading package VITAL_Timing
# -- Loading package VITAL_Primitives
# -- Loading package cycloneii_atom_pack
# -- Loading package cycloneii_components
# -- Compiling entity truth_table_1
# -- Compiling architecture structure of truth_table_1
#
#

ModelSim>
```

Figure 14 Start Simulation



Name	Type	Path
work	Library	gate_work
truth_table_1	Entity	C:\qdesigns\Digital_Systems\labs2012...
gate_work	Library	C:/qdesigns/Digital_Systems/labs2012...
220model	Library	$MODEL_TECH/../altera/vhdl/220model
220model_ver	Library	$MODEL_TECH/../altera/verilog/220m...
altera	Library	$MODEL_TECH/../altera/vhdl/altera
altera_lnsim	Library	$MODEL_TECH/../altera/vhdl/altera_l...
altera_lnsim_ver	Library	$MODEL_TECH/../altera/verilog/altera...
altera_mf	Library	$MODEL_TECH/../altera/vhdl/altera_mf

Design Unit(s): work.truth_table_1

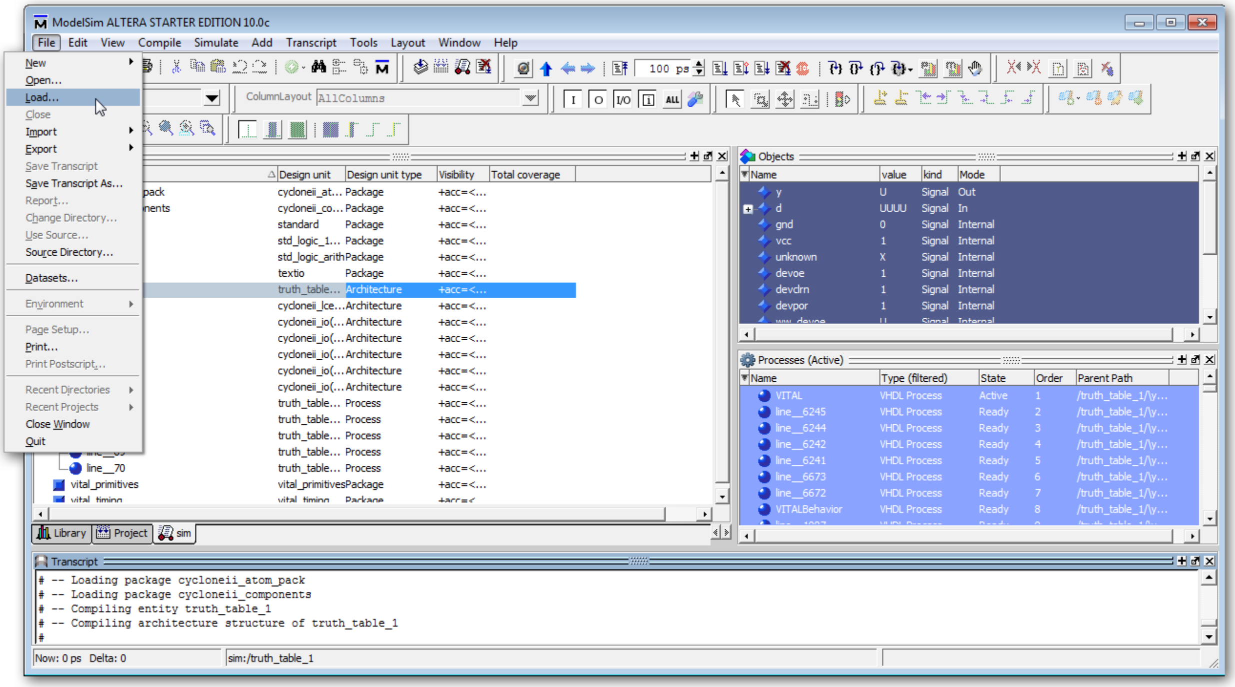
Resolution: default

Optimization

□ Enable optimization

Figure 15 Selecting the Entity to Simulate

After a few seconds, the window in Figure 14 will be available. From the **File** menu, select **Load** to load the previously-saved wave format file. As shown in Figure 15, select the previously-saved wave format, in this case **wave.do**. Click **Open**.



	Design unit	Design unit type	Visibility	Total coverage
pack	cycloneii_at...	Package	+acc=<...	
nents	cycloneii_co...	Package	+acc=<...	
	standard	Package	+acc=<...	
	std_logic_1...	Package	+acc=<...	
	std_logic_arith	Package	+acc=<...	
	textio	Package	+acc=<...	
	truth_table...	Architecture	+acc=<...	
	cycloneii_lce...	Architecture	+acc=<...	
	cycloneii_io(...	Architecture	+acc=<...	
	cycloneii_io(...	Architecture	+acc=<...	
	cycloneii_io(...	Architecture	+acc=<...	
	cycloneii_io(...	Architecture	+acc=<...	
	cycloneii_io(...	Architecture	+acc=<...	
	truth_table...	Process	+acc=<...	
	truth_table...	Process	+acc=<...	
	truth_table...	Process	+acc=<...	
	truth_table...	Process	+acc=<...	
line__70	truth_table...	Process	+acc=<...	
vital_primitives	vital_primitives	Package	+acc=<...	
vital_timing	vital_timing	Package	+acc=<	

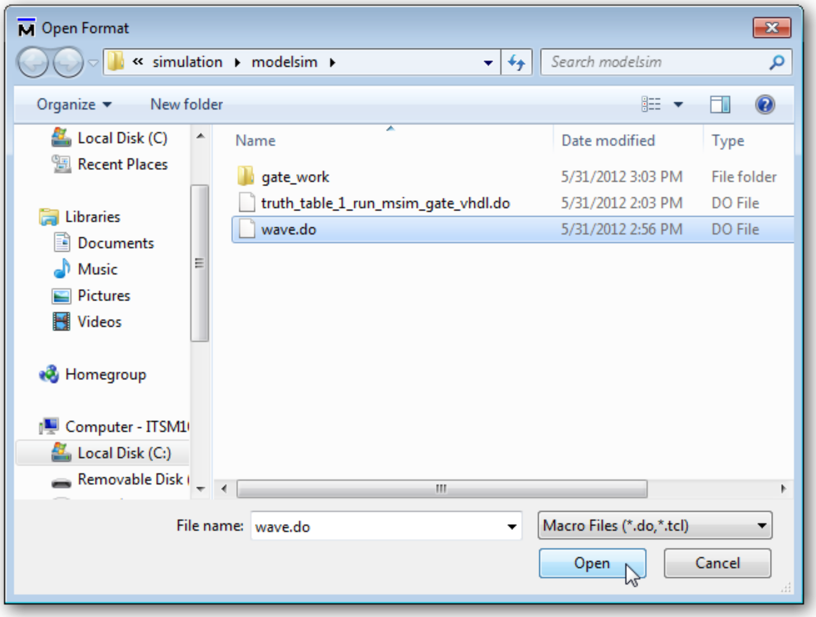
Name	value	kind	Mode
y	U	Signal	Out
d	UUUU	Signal	In
gnd	0	Signal	Internal
vcc	1	Signal	Internal
unknown	X	Signal	Internal
devoe	1	Signal	Internal
devclrn	1	Signal	Internal
devpor	1	Signal	Internal
ww_devoe	U	Signal	Internal

Name	Type (filtered)	State	Order	Parent Path
VITAL	VHDL Process	Active	1	/truth_table_1/\y...
line__6245	VHDL Process	Ready	2	/truth_table_1/\y...
line__6244	VHDL Process	Ready	3	/truth_table_1/\y...
line__6242	VHDL Process	Ready	4	/truth_table_1/\y...
line__6241	VHDL Process	Ready	5	/truth_table_1/\y...
line__6673	VHDL Process	Ready	6	/truth_table_1/\y...
line__6672	VHDL Process	Ready	7	/truth_table_1/\y...
VITALBehavior	VHDL Process	Ready	8	/truth_table_1/\y...

```
# -- Loading package cycloneii_atom_pack
# -- Loading package cycloneii_components
# -- Compiling entity truth_table_1
# -- Compiling architecture structure of truth_table_1
#
```

Now: 0 ps Delta: 0   sim:/truth_table_1

Figure 16 Loading the Previously-Saved Wave Format



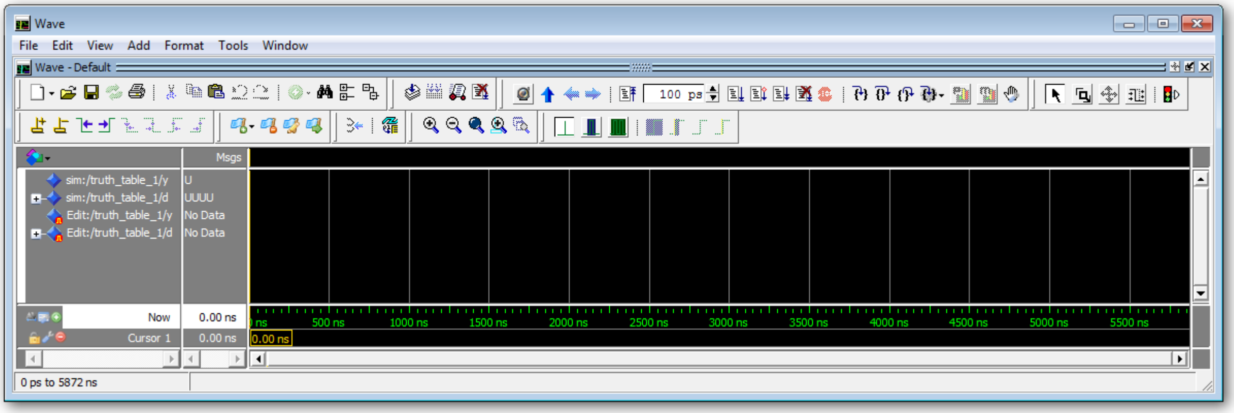
Name	Date modified	Type
gate_work	5/31/2012 3:03 PM	File folder
truth_table_1_run_msim_gate_vhdl.do	5/31/2012 2:03 PM	DO File
wave.do	5/31/2012 2:56 PM	DO File

File name: wave.do

Macro Files (*.do,*.tcl)

Figure 17 Selecting the Previously-Saved Wave Format

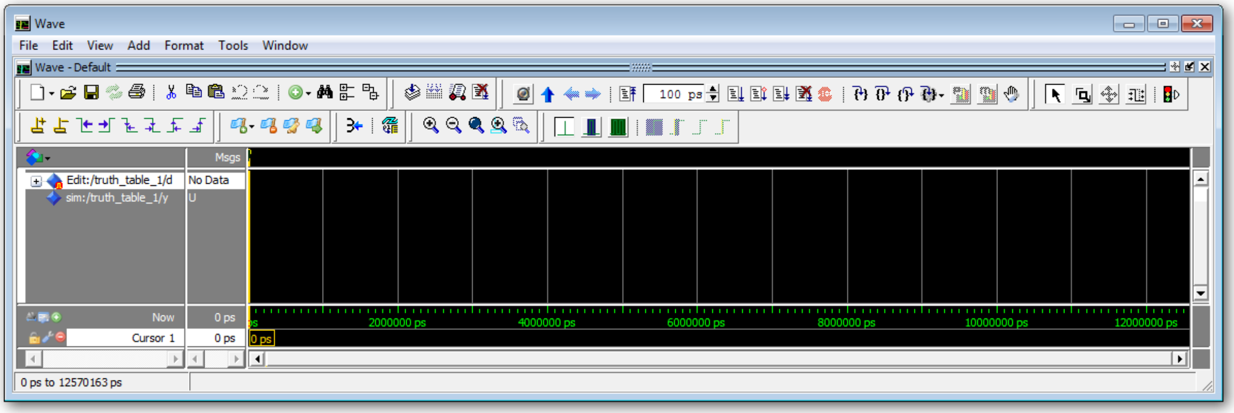
Figure 16 shows the **Wave** window with two sets of waveforms. The top two are simulation waveforms (indicated by **sim:**) that are generated by the simulator. The bottom two are editable waveforms (indicated by **edit:**), which can be changed by the waveform editor. We want the **d** input waveforms to be editable and the **y** output waveform to be generated when we run the simulator. Select and delete the two unwanted waveforms: **sim:/truth_table_1/d** and **edit:/truth_table_1/y**. This will leave the **Wave** window looking like Figure 17.



	Msgs
sim:/truth_table_1/y	U
sim:/truth_table_1/d	UUUU
Edit:/truth_table_1/y	No Data
Edit:/truth_table_1/d	No Data
Now	0.00 ns
Cursor 1	0.00 ns

0 ps to 5872 ns

Figure 18 Wave File with Editable Waveforms Added



	Msgs
Edit:/truth_table_1/d	No Data
sim:/truth_table_1/y	U
Now	0 ps
Cursor 1	0 ps

0 ps to 12570163 ps

Figure 19 Deleting Unused Waveforms

To test the VHDL design, we will place a 4-bit count on the **d** input. We will count from 0 to 1600 ns, with the waveform changing every 100 ns. Start by highlighting and right-clicking on the **d** waveform, as shown in Figure 18. Select **Edit** from the popup menu, then **Create/Modify Waveform** from the popup menu arising from the first one. A default pattern generator window appears as shown in Figure 19. Set the pattern type to **Counter** with an **End Time** of 1600 and a **Time Unit** of ns, as shown in Figure 20.

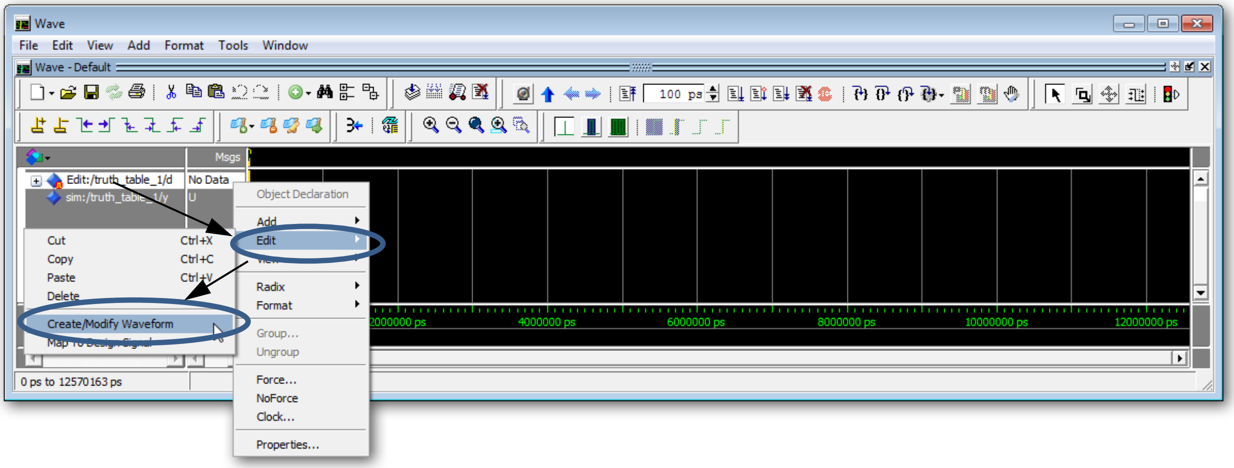
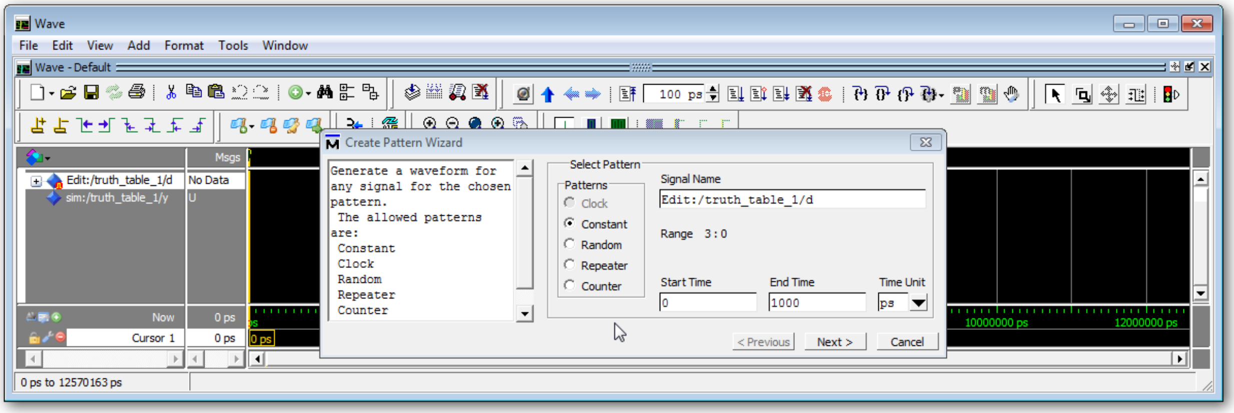


Figure 20 Editing the "d" Input Waveform



```
Generate a waveform for
any signal for the chosen
pattern.
 The allowed patterns
are:
 Constant
 Clock
 Random
 Repeater
 Counter
```

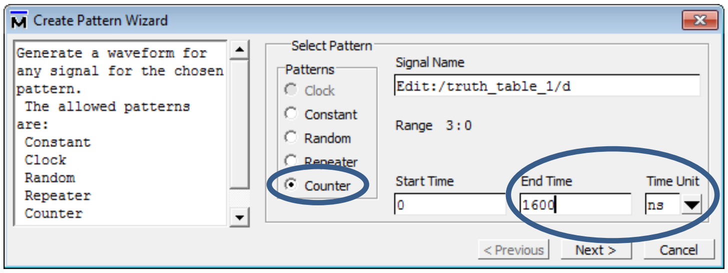
Patterns: ○ Clock ● Constant ○ Random ○ Repeater ○ Counter

Signal Name: Edit:/truth_table_1/d

Range 3:0

Start Time: 0 End Time: 1000 Time Unit: ps

Figure 21 Default State of the Pattern Editor Wizard



```
Generate a waveform for
any signal for the chosen
pattern.
 The allowed patterns
are:
 Constant
 Clock
 Random
 Repeater
 Counter
```

Patterns: ○ Clock ○ Constant ○ Random ○ Repeater ● Counter

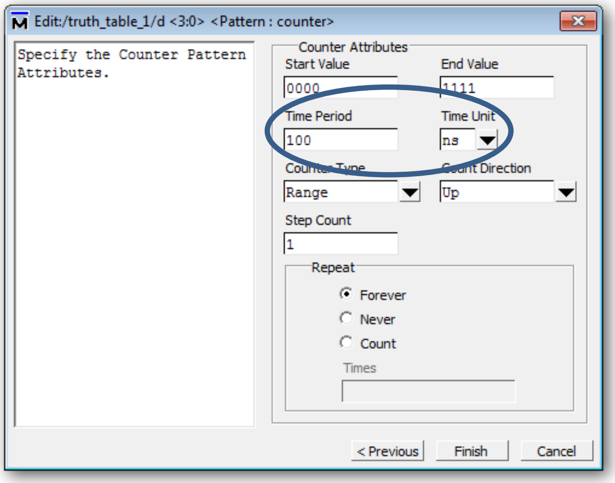
Signal Name: Edit:/truth_table_1/d

Range 3:0

Start Time: 0 End Time: 1600 Time Unit: ns

Figure 22 Setting the Pattern Wizard to Create a Counter with an End Time of 1600 ns

Set the **Time Period** of the counter to 100 and the **Time Unit** to ns, as shown in Figure 21. Click **Finish** to continue. Click **Zoom Full**, as shown in Figure 22, or select this option from the **View** menu in the **Wave** window, to see the entire input waveform from 0 to End Time. The counter waveform will appear on **d**, as shown in Figure 23.



Edit:/truth_table_1/d <3:0> <Pattern : counter>

Specify the Counter Pattern Attributes.

Counter Attributes

Start Value: 0000

End Value: 1111

Time Period: 100

Time Unit: ns

Counter Type: Range

Count Direction: Up

Step Count: 1

Repeat

● Forever

○ Never

○ Count

Times

Figure 23 Setting the Counter Time Period to 100 ns

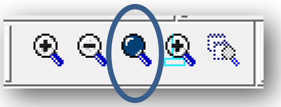
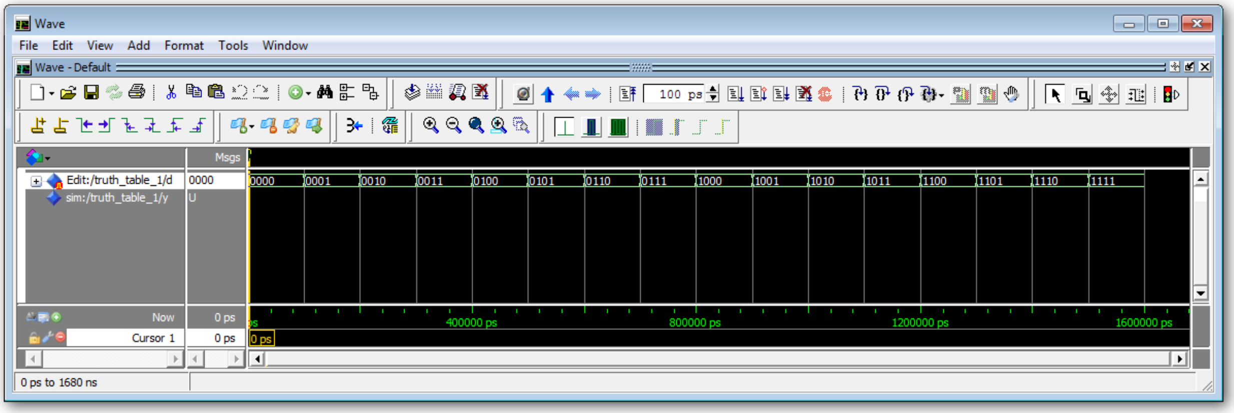


Figure 24 Zoom Full to See Counter Pattern from Start Time to End Time



	Msgs
Edit:/truth_table_1/d	0000
sim:/truth_table_1/y	U

0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111

Now 0 ps

Cursor 1 0 ps

0 ps to 1680 ns

Figure 25 Wave Window with 4-bit Counter Pattern on "d"

To generate the output waveform, click the **Run -All** button in the **Wave** window, as shown in Figure 24. The resultant output waveform is shown in Figure 25, with the **y** output HIGH when **d[3..0]** is one of the following codes: 0011, 0110, 1001, 1101.

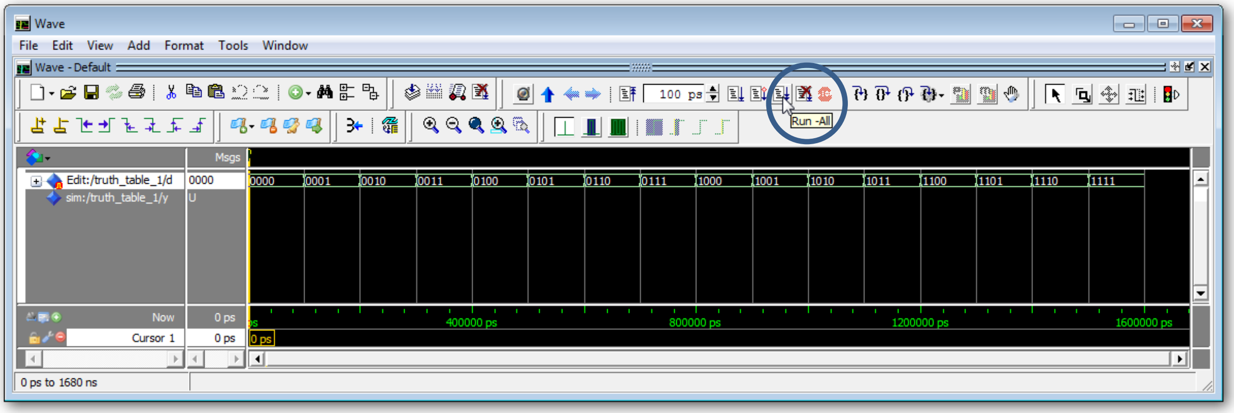


Figure 26 Run Simulation until End Time with Run All Button

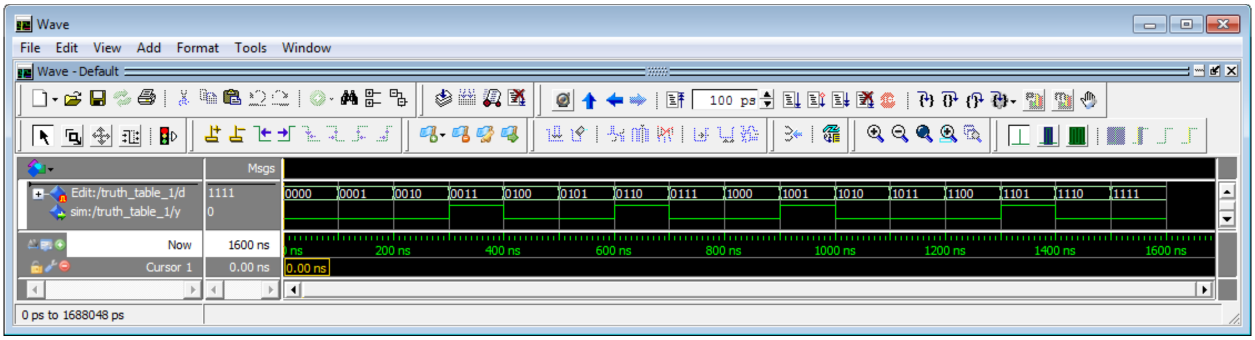


Figure 27 Completed Simulation (Output HIGH on 0011, 0110, 1001, and 1101)

To run the simulation again, set it back to the beginning with the **Restart** button, shown in Figure 26. Then use the **Run -All** button, as in Figure 24.

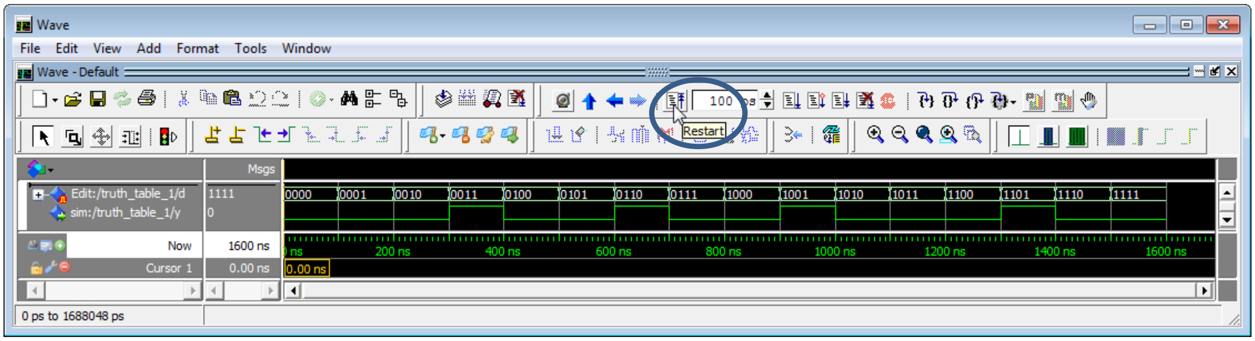
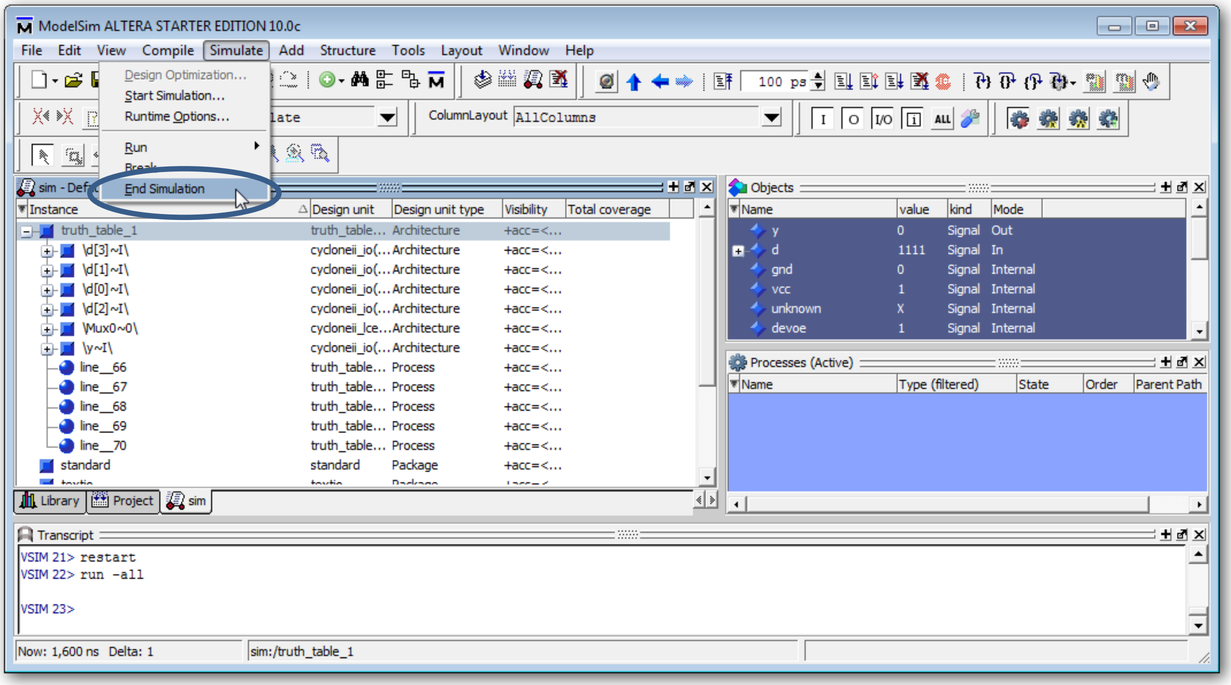


Figure 28 Restarting the Simulation

End the simulation by selecting **End Simulation** from the ModelSim **Simulate** menu, as shown in Figure 27. When this process is complete, close the ModelSim window. The dialog box in Figure 28 will appear, asking if you want to quit. Click **Yes**. The program will close and return to Quartus II. If the design needs to be changed, make the changes, compile the design in Quartus II, then run the gate level simulation and all following steps, beginning again at Figure 5.



Instance	Design unit	Design unit type	Visibility	Total coverage
truth_table_1	truth_table...	Architecture	+acc=<...	
\d[3]~I\	cycloneii_io(...	Architecture	+acc=<...	
\d[1]~I\	cycloneii_io(...	Architecture	+acc=<...	
\d[0]~I\	cycloneii_io(...	Architecture	+acc=<...	
\d[2]~I\	cycloneii_io(...	Architecture	+acc=<...	
\Mux0~0\	cycloneii_lce...	Architecture	+acc=<...	
\y~I\	cycloneii_io(...	Architecture	+acc=<...	
line__66	truth_table...	Process	+acc=<...	
line__67	truth_table...	Process	+acc=<...	
line__68	truth_table...	Process	+acc=<...	
line__69	truth_table...	Process	+acc=<...	
line__70	truth_table...	Process	+acc=<...	
standard	standard	Package	+acc=<...	

Name	value	kind	Mode
y	0	Signal	Out
d	1111	Signal	In
gnd	0	Signal	Internal
vcc	1	Signal	Internal
unknown	X	Signal	Internal
devoe	1	Signal	Internal

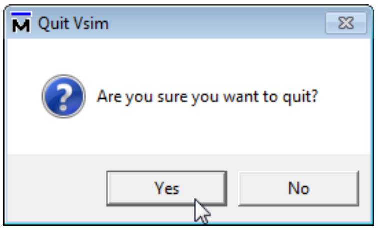
Name	Type (filtered)	State	Order	Parent Path

```
VSIM 21> restart
VSIM 22> run -all

VSIM 23>
```

Now: 1,600 ns Delta: 1 sim:/truth_table_1

Figure 29 Ending Simulation



Are you sure you want to quit?

Figure 30 Quit ModelSim Dialog to Return to Quartus II