

For Quartus Prime 15.1

# 1 Introduction

An effective way of determining the correctness of a logic circuit is to simulate its behavior. This tutorial provides an introduction to such simulation using Altera's Quartus Prime CAD system.

Introduction to Simulation

of Verilog Designs

The simulation method used in this tutorial is based on drawing waveforms, similar to timing diagrams, that are inputs for a simulator tool. The outputs of the simulator are also in the form of waveforms. This tutorial is intended for students who are taking a course in logic circuit design. We show how to use the Simulation Waveform Editor tool provided in the Quartus Prime software to perform a simulation of a circuit specified in Verilog HDL. Only a very basic understanding of Verilog is needed for this purpose.

### **Contents**:

- Design Project
- Creating Waveforms for Simulation
- Simulation
- Making Changes and Resimulating
- Concluding Remarks

The Simulation Waveform Editor tool is available for use with Altera's Quartus II software version 13.0 or later. It allows the user to apply inputs to the designed circuit, usually referred to as *test vectors*, in the form of waveforms and to observe the outputs generated in response.

In this tutorial, the reader will learn about:

- Test vectors needed to test the designed circuit
- · Using the Simulation Waveform Editor tool to draw test vector waveforms
- Functional simulation, which is used to verify the functional correctness of a synthesized circuit
- Timing simulation, which is used to verify the timing of signals in a synthesized circuit

This tutorial is aimed at the reader who wishes to simulate circuits defined by using the Verilog hardware description language. An equivalent tutorial is available for the user who prefers the VHDL language.

## 2 Design Project

To illustrate the simulation process, we will use a very simple logic circuit that implements the *majority* function of three inputs,  $x_1$ ,  $x_2$  and  $x_3$ . The circuit is defined by the expression

$$f(x_1, x_2, x_3) = x_1 x_2 + x_1 x_3 + x_2 x_3$$

In Verilog, this circuit can be specified as follows:

 $\begin{array}{ll} \mbox{module majority3 (x1, x2, x3, f);} \\ \mbox{input } x1, x2, x3; \\ \mbox{output } f; \\ \mbox{assign } f = (x1 \ \& \ x2) \mid (x1 \ \& \ x3) \mid (x2 \ \& \ x3); \\ \mbox{endmodule } \end{array}$ 

Implement this circuit as follows:

- Create a project *majority3*.
- Include a file *majority3.v*, which corresponds to the verilog snippet above.
- Select the correct device that is associated with the DE-series board. A list of device names for the DE-series boards can be found in Table 1.
- Compile the design.

Board	Device Name
DE0-CV	Cyclone V 5CEBA4F23C7
DE0-Nano	Cyclone IVE EP4CE22F17C6
DE0-Nano-SoC	Cyclone V SoC 5CSEMA4U23C6
DE1-SoC	Cyclone V SoC 5CSEMA5F31C6
DE2-115	Cyclone IVE EP4CE115F29C7

Table 1. DE-series FPGA device names

# 3 Creating Waveforms for Simulation

To create test vectors for your design, select File > New... > Verification/Debugging Files > University Program VWF in the Quartus Prime window where the design project is open. This opens the Simulation Waveform Editor tool, shown in Figure 1, which allows you to specify the desired input waveforms.

🕤 Simul	lation Wavefo	rm Editor - D:/	simulation	_intro/majority3 - majo	ority3 - [Wavefor	rm2.vwf]			
<u>File</u>	dit <u>V</u> iew <u>S</u>	imulation <u>H</u> elp	, 🗕					Search altera.co	om 🌖
	🔍 🐹 🕹 J	1. <u>z</u> XI XI		X X2 X =   🗞 🖏	🍋 🔜 晄				
Master T	Time Bar: 0 ps			Pointer: 887.12 ns	Interval:	887.12 ns	Start:	End:	
	Name	Value at	0 ps	160.0 ns	320.0 ns	480.0 ns	640.0 ns 8	00.0 ns	960.0 ns 🔺
	- Turno	0 ps	0 ps						
		+	•						
								0%	00:00:00

Figure 1. The Waveform Editor window.

For our simple circuit, we can do a complete simulation by applying all eight possible valuations of the input signals  $x_1$ ,  $x_2$  and  $x_3$ . The output *f* should then display the logic values defined by the truth table for the majority function.

We will run the simulation for 800 ns; so, select  $\mathsf{Edit} > \mathsf{Set} \mathsf{End} \mathsf{Time...}$  in the Waveform Editor and in the pop-up window that will appear specify the time of 800 ns, and click OK. This will adjust the time scale in the window of Figure 1.

Before drawing the input waveforms, it is necessary to locate the desired signals in the implemented circuit. In

FPGA jargon, the term "node" is used to refer to a signal in a circuit. This could be an input signal (input node), output signal (output node), or an internal signal. For our task, we need to find the input and output nodes. This is done by using a utility program called the Node Finder.

In the Waveform Editor window, select Edit > Insert > Insert Node or Bus.... In the pop-up window that appears, which is shown in Figure 2, click on Node Finder.

🕥 Insert Noo	de or Bus	×
Name:		ОК
Туре:	INPUT	Cancel
Value type:	9-Level 🔻	
Radix:	Binary 🔻	Node Finder
Bus width:	1	
Start index:	0	
🗌 Display g	ray code count as binary cou	int

Figure 2. The Insert Node or Bus dialog.

The Node Finder window is presented in Figure 3. A filter is used to identify the nodes of interest. In our circuit, we are only interested in the nodes that appear on the *pins* (i.e. external connections) of the FPGA chip. Hence, the filter setting should be Pins: all. Click on List, which will display the nodes as indicated in the figure. In a large circuit there could be many nodes displayed. We need to select the nodes that we wish to observe in the simulation. This is done by highlighting the desired nodes and clicking on the > button. Select the nodes labeled x1, x2, x3, and f, which will lead to the image in Figure 4. Click OK in this window and also upon return to the window in Figure 2. This returns to the Waveform Editor window, with the selected signals included as presented in Figure 5.

3 Node Finder		
Named: *		Filter: Pins: all   OK  OK  Cancel
Nodes Found:		Selected Nodes:
Name out f in x1 in x2 in x3	Type Output Input Input Input	Name Type

Figure 3. The Node Finder dialog.

Named: *			Filter: Pi	ns: all		_	ОК
Look in: *					List		Cancel
Nodes Found:			Sele	cted Nodes:			
Name	Ту			Name		Туре	
out f	Output		些	f	Output		
in x1	Input		2 📮	x1	Input		
i≞_ x2	Input		≥ i≞_	x2	Input		
in x3	Input		< in-	x3	Input		
		6	~~				

Figure 4. The selected signals.

Observe that in Figure 5 all input signals are at logic level 0. The output, f is shown as undefined. Next, we have to draw the input waveforms. Then, we will simulate the circuit, which will produce the output waveform.

To make it easier to draw the input waveforms, the Waveform Editor displays dashed grid lines. The spacing of the grid lines can be adjusted by selecting Edit > Grid Size..., and in the pop-up box in Figure 6 specifying the desired size. The spacing of grid lines in Figure 5 is 20 ns. Another convenience in drawing is to have transitions of a waveform snap on grid lines. This feature is activated by clicking on the Snap to Grid icon  $\overrightarrow{m}$ , or by selecting the command Edit > Snap to Grid.

е	<u>E</u> dit <u>V</u> iew	<u>Simulation</u> <u>H</u> e	elp 🤜				Search alte	ra.com
	0 🔊 🔎	1, <u>z</u> )( <u>r</u> )	H WY XC X	X2 XB 🗞 🗞 🌴	a 📾 🎼			
aster	r Time Bar: 0	ps		Pointer: 39.66 ns	Interval: 39.66 ns	Start:	End:	
	Name	Value at 0 ps	0 ps 0 ps	160.0 ns	320.0 ns	480.0 ns	640.0 ns	800.0 ns
ut	f	вх		******			******	*****
-	<b>x</b> 1	В 0						
-	x2	В 0						
-	x3	B 0						
(								• • • •

Figure 5. Signals in the Waveform Editor window.

Grid Size	×
Base grid on time period	
Period: 10.0	ns 🔻
ОК	Cancel
	Base grid on time period Period: 10.0

Figure 6. Specifying the grid spacing.

Input waveforms can be drawn in different ways. The most straightforward way is to indicate a specific time range and specify the value of a signal. To illustrate this approach, click the mouse on the x1 waveform near the 400-ns point and then drag the mouse to the 800-ns point. The selected time interval will be highlighted in blue, as depicted in Figure 7. Change the value of the waveform to 1 by clicking on the Forcing High (1) icon A, as illustrated in Figure 8.

ile	<u>E</u> dit <u>V</u> iew	Simulation He	ilp 🤜				Search alt	era.com
	역 🐹 🕹	h, Z, Xī, ⟩	₩ ₩ \c \&	X2 XB 🔩 👯 🦓	a 副 账			
laste	r Time Bar: 0	ps	• • P	ointer: 618.7 ns	Interval: 618.7 ns	Start: 400.0 ns	End:	800.0 ns
	Name	Value at 0 ps	0 ps 0 ps	160.0 ns	320.0 ns	480.0 ns	640.0 ns	800.0 ns
out	f	ВX		****		****	******	*****
in_	<b>x1</b>	B 0						
in_	x2	B 0						
in_	x3	В 0						
								• • • • •

Figure 7. Selection of a time interval.

ile	<u>E</u> dit <u>V</u> iew	Simulation <u>H</u> e	elp 🤜				Search a	ltera.com
	역 🐹 🕹	<u>ት ጃ  (</u> ጀ )	₩ 🔀 🗕	X2 XB 🗞 🗞 🊈				
laster	r Time Bar: 0	ps		pinter: 760.15 ns	Interval: 760.15 ns	Start: 400.0 i	ns End:	800.0 ns
	Name	Value at 0 ps	0 ps 0 ps	160.0 ns	320.0 ns	480.0 ns	640.0 ns	800.0 ns
out	f	вх			*****			******
in	<b>x</b> 1	В 0						
in	x2	В 0						
in_	<b>x</b> 3	В 0						
4								· · · · ·

Figure 8. Drawing the waveform for *x1* 

In creating the waveform for xI, we used the icon 1 to implement the logic value 1. Another possibility is to invert the value of the signal in a selected time interval by using the Invert icon 3. We will use this approach to create the waveform for x2, which should change from 0 to 1 at 200 ns, then back to 0 at 400 ns, and again to 1 at 600 ns. Select the interval from 200 to 400 ns and click on the icon. Then do the same for the interval from 600 to 800 ns, as illustrated in Figure 9.

ile	<u>E</u> dit <u>V</u> iew	Simulation He	elp 🤜				Search alte	era.com
	ම් 💥 🕤	· <mark>አ </mark> 🏒 🗎 🕽	₩ ₩ Хс Х	X2 XB 🗞 🗞 🍫	· [詞 ]];			
laster	Time Bar: 0	ps	• • P	ointer: 712.56 ns	Interval: 712.56 ns	Start: 600.0 n	s End: 8	300.0 ns
	Name	Value at	0 ps	160.0 ns	320.0 ns	480.0 ns	640.0 ns	800.0 ns
		0 ps	0 ps					
out	f	вх		******	*****	******	*******	*****
n	<b>x</b> 1	B 0						
n	x2	B 0						
n	x3	B 0						
4				i i i i i i i				<u> </u>

Figure 9. Drawing the waveform for *x*2.

We will use a third approach to draw the waveform for x3. This signal should alternate between logic values 0 and 1 at each 100-ns interval. Such a regular pattern is indicative of a *clock* signal that is used in many logic circuits. Even though there is no clock signal in our example circuit, it is convenient to specify x3 in this manner. Click on the x3 input, which selects the entire 800-ns interval. Then, click on the **Overwrite Clock** icon  $\boxed{3}$ , as indicated in Figure 10. This leads to the pop-up window in Figure 11. Specify the clock period of 200 ns and the duty cycle of

50%, and click OK. The result is depicted in Figure 12.

🕤 Sin	ulation Wave	form Editor - D;	/simulation_intro	o/majority3 - majority	3 - [Waveform.vwf]*	_		- 0 X
File	<u>E</u> dit <u>V</u> iew	Simulation Hel	p 🤜				Search alte	ra.com 🜖
	역 💥 🕹	. <b></b> ∖⊑ ∖	H 🚻 XC 📧	X2 XB 🔩 👯 🊈	<b>武</b>			
Maste	r Time Bar: 0	ps		Overwrite Clock	Interval: 78.0 ns	Start: 0 ps	End: 8	00.0 ns
	Name	Value at 0 ps	0 ps 0 ps	160.0 ns	320.0 ns	480.0 ns	640.0 ns	800.0 ns 🌋
out	f	вх		******	*****	*****	******	
in	<b>x</b> 1	B 0						
in	x2	B 0						
in	x3	B 0						
•		Þ	•					
Overw	rite Clock						0	% 00:00:00

Figure 10. Drawing the waveform for *x3*.

Clock			x
-Base w	aveform	on time period	·
Period:	200.0		ns 🔻
Offset:	0.0		ns 🔻
Duty cy	cle (%):	50	<u>*</u>
		ок	Cancel

Figure 11. Defining the clock characteristics

ile	<u>E</u> dit <u>V</u> iew	Simulation He	elp 🤜				Search alte	ra.com
	بي 🔍	┟╫╧╎╧	₩ ₩ XC X0	X2 XB 💐 🗞 🆧	<u>ا ا ا ا ا ا ا ا ا ا ا ا ا ا ا ا ا ا ا</u>			
laster	r Time Bar: 0	) ps	• •	Pointer: 187.72 ns	Interval: 187.72 ns	Start: 0 ps	End: 8	00.0 ns
	Name	Value at 0 ps	0 ps 0 ps	160.0 ns	320.0 ns	480.0 ns	640.0 ns	800.0 ns 4
out	f	ВX			****		****	*****
in	x1	B 0						
in	x2	В 0						
in	<b>x</b> 3	В 0						
4					III.			

Figure 12. The completed input waveforms.

Save the waveform file using a suitable name; we chose the name *majority3.vwf*. Note that the suffix *vwf* stands for *vector waveform file*. VWF files that are added to the Quartus Prime project can be accessed at any time in the Project Navigator Widget's Files tab.

## 4 Simulation

The Simulation Waveform Editor performs the simulation by using the simulation tool known as *ModelSim*. ModelSim-Altera Edition is strongly recommended for use with the Simulation Waveform Editor, as it contains the Altera device libraries necessary for simulations. To use a standard version of ModelSim, the path to its executables must be specified in the Quartus Prime software under Tools > Options... > EDA Tool Options. If both ModelSim and ModelSim-Altera are available, the simulator will preferentially use ModelSim-Altera.

#### 4.1 Functional Simulation

Now that we have created the input vector waveform, we can simulate the circuit. In the Simulation Waveform Editor, select Simulation > Run Functional Simulation, or click on the icon 3. A pop-up window will show the progress of the simulation, then automatically close when it is complete. A second Simulation Waveform Editor window then opens the output waveform, as depicted in Figure 13. The output waveform is read-only, so any changes in simulation have to be done by modifying the *majority3.vwf* file and resimulating the circuit. Observe that the output *f* is equal to 1 whenever two or three inputs have the value 1, which verifies the correctness of our design.

	Edit ⊻iew et Xan en	Simulation He	•	2 XB   R R R A	副既			Search altera.co	om
laster	r Time Bar: 0	ps	• Po	inter:	Interval:	Start:		End:	
	Name	Value at 0 ps	0 ps 0 ps	160.0 ns	320.0 ns	480.0 ns	64	0.0 ns	800.0 ns 🖌
out	f	B 0							
in	x1	В 0							
in	x2	В 0							
in_	<b>x</b> 3	В 0							
4		1	• •			m			

Figure 13. Result of the functional simulation.

### 4.2 Timing Simulation

To observe the actual propagation delays in our circuit, we have to perform a timing simulation. (Note that for FPGA devices with preliminary timing models that the timing simulation results may be the same as functional simulation results.) In the Simulation Waveform Editor, select Simulation > Run Timing Simulation, or click on the icon  $\underbrace{\underbrace{\baselim}}_{i}$ . A pop-up window will show the progress of the simulation, then automatically close when it is complete. A second Simulation Waveform Editor window then opens the output waveform. The output waveform is read-only, so any changes in simulation have to be done by modifying the *majority3.vwf* file and resimulating the circuit.

The timing simulation shows that there are delays when signals change from one value to another. Figure 14 shows the waveform, zoomed in at 300 ns to show the propagation delay between x3 and f. The waveform indicates that the maximum delay is approximately 6 ns.

**Note:** timing simulations are only supported by Cyclone IV and Stratix IV FPGAs. If your Quartus project is not setup for a Cyclone IV or Stratix IV device, the result of running a timing simulation will be identical to the functional simulation.

ile	Edit View	Simulation Help	9	Se	arch altera.com
	e 🔍 🔍	, 노 프 )도 )	Ξ 骝 Xē Xā Xā Xē 🗟 🗟 🗟 🖼		
laster	r Time Bar: 0	ps	Pointer: 60.81 ns Interval: 60.81 ns	Start:	End:
	Name	Value at 0 ps	300.0 ns	310.0 ns	
out	f	в 0			
in	x1	В 0			
in	x2	B 0			
in	<b>x</b> 3	B 0			
4					

Figure 14. Result of the timing simulation, zoomed in at 300 ns.

# 5 Making Changes and Resimulating

Changes in the input waveforms can be made using the approaches explained above. The circuit can then be resimulated using the altered waveforms. For example, change the waveform for  $x_1$  to have the logic value 1 in the interval from 100 to 240 ns, as indicated in Figure 15. Now, simulate the circuit again. The result is given in Figure 16. If errors in the circuit are discovered, then these errors can be fixed by changing the Verilog code and recompiling the design using the Quartus Prime software.

ile	<u>E</u> dit <u>V</u> iew	Simulation <u>H</u> e	lp 🤜				Search alte	era.com
	🔍 👗 🕘	- ት 🚣 🔀	₩ <u>X</u> C X <u>&amp;</u>	X2 XB 🗞 🗞 🌤	<b>≣</b> ₩			
laste	r Time Bar: 0	ps	• • P	ointer: 218.13 ns	Interval: 218.13 ns	Start: 100.0	ns End:	240.0 ns
	Name	Value at 0 ps	0 ps 0 ps	160.0 ns	320.0 ns	480.0 ns	640.0 ns	800.0 ns
ut	f	ВX			*********	******		*****
n	<b>x</b> 1	В 0						
n	x2	B 0						
n	<b>x</b> 3	B 0						
۰.								· · · · · ·

Figure 15. Modified input waveforms.

	<u>E</u> dit <u>V</u> iew	Simulation He					Search altera	com
	<u>e</u> 💥 🕑	h <b>z</b> χī ⟩	H WY XC XS	X2 XB   № № № %	副影			
lastei	r Time Bar: 0	ps	Pr	pinter: 712.56 ns	Interval: 712.56 ns	Start:	End:	
	Name	Value at 0 ps	0 ps 0 ps	160.0 ns	320.0 ns	480.0 ns	640.0 ns	800.0 ns 🗳
out	f	B 0						
in	<b>x</b> 1	В 0						
in	x2	В 0						
in	<b>x</b> 3	В 0						
					111			

Figure 16. Result of the new simulation.

# 6 Concluding Remarks

The purpose of this tutorial is to provide a quick introduction to the Simulation Waveform Editor, explaining only the rudimentary aspects of functional and timing simulations. Details about additional features of the Simulation Waveform Editor can be found in the appendix of this document.

To learn about more about simulating circuits using ModelSim, please refer to the tutorials *Introduction to Model-Sim's Graphical Waveform Editor*, and *Using ModelSim to Simulate Logic Circuits*, which are available on Altera's University Program website.

# A Simulation Waveform Editor

In section 3 we introduced the Waveform Editor tool, which is used to view and edit waveforms that are used in simulation. Additional features of the Waveform Editor are described in this appendix.

### A.1 Waveform Editor Toolbar

The Waveform Editor window is illustrated in Figure 1. The tool includes several commands which can be accessed by using the mouse, including File, Edit, View, Simulation, and Help. Below these commands, as shown in the figure, there is a toolbar that contains a number of icons which are useful when manipulating waveforms. This toolbar should be visible by default, but if it is not visible, then right-click near the top of the window (below the title bar) and select Waveform Editor in the menu that appears.

The toolbar icons are described below.

Selection Tool 💽

This tool is used to select waveform intervals and apply changes. To make a selection, click on any part of a waveform and drag the blue box across the desired interval. It's possible to select multiple waveforms at the same time, as shown in Figure 1, or select entire waveform(s) by clicking on its name(s).

ile I	Edit View	Simulation He	lp 🤜				Search alte	ra.com
	ല് 👗 ല	ϧϪϪϪ	₩ WK XC XX	X2 XB 💐 💐 🏘	, <b>≣</b> ;			
laster	Time Bar: 0	) ps	• • •	Pointer: 778.66 ns	Interval: 778.66 ns	Start: 380.0	ns End: 7	20.0 ns
	Name	Value at	0 ps	160.0 ns	320.0 ns	480.0 ns	640.0 ns	800.0 ns
		0 ps	0 ps					
out	f	ВX		*******	*********	*********	~~~~~~	$\times$
in_	<b>x</b> 1	B 0						
in	x2	B 0						
in	x3	в 0						
4					<u> </u>			

Figure 1. Using the Selection Tool to select a portion of multiple waveforms.

Double clicking the selection tool anywhere on a waveform will select the largest interval with the same value from where the cursor points. Double clicking on a selected interval brings up the window to set arbitrary values for that interval.

### Zoom Tool 🔍

This tool is used to zoom in or zoom out in the waveform display, as indicated in Figure 2. Left-clicking zooms into the display and right-clicking zooms out.

	🗨 🔉 🖞	5 <b>ㅗ ႍ</b> ×도 >	स ₩४ \c X	5 X? XB 📌 💐	2 🚈 📾 晄				
laster	Time Bar:	0 ps	• •	Pointer: 247.21 ns	s Interva	al: 247.21 ns	Start:	End:	
	Name	Value at 0 ps	0 ps 0 ps	160.0 ns	320	.0 ns	480.0 ns	640.0 ns	800.0 ns 4
out	f	ВX				*****	~~~~~	****	*****
in	<b>x</b> 1	В 0			+				
in	x2	В 0							
in	<b>x</b> 3	В 0							
•		)			1 1 1 1 1	· · · · · · ·			• • • •

Figure 2. Using the Zoom Tool.

Forcing Unknown(X)

This tool allows the selected part of a waveform to be set to the value Unknown (x). An example is given in Figure 3, using the majority3 function circuit that was described in section 2. The value of the signal x3 has been set to unknown for the first half of the simulation. Running the simulation with these input values results in the output waveform f that is shown in the figure. Note that the value of f is unknown between 200 to 400 ns.

ile	<u>E</u> dit <u>V</u> iew	Simulation He	elp 🤜				Search altera	a.com
	2 💥 🗅	5 슈 <b>프</b> )(토 )	$(\underline{H} \ \underline{WV} \ \underline{\langle \underline{C} \ } \ \underline{\langle \underline{N} \ } \ )$	(?) XB   ₹ ₹ ₹ 7	a 🚮 🎀			
Maste	r Time Bar: 0	ps	• • Po	inter:	Interval:	Start:	End:	
	Name	Value at 0 ps	0 ps 0 ps	160.0 ns	320.0 ns	480.0 ns	640.0 ns	800.0 ns
out	f	В 0			****			
in	x1	В 0						
in	x2	В 0						
in	<b>x</b> 3	вх		*******	******			
			• •		1 1 1 1 1 1 1	 T		4

Figure 3. Setting the value of an input to Unknown (X).

## Forcing Low (0) 🕑 and Forcing High(1) 🖪

These tools are used to force the selected part of a waveform to the value low (0) or high (1), as shown in Figures 4 and 5, respectively.

ile	<u>E</u> dit <u>V</u> iew	Simulation He	elp 🤜				Search at	tera.com
	م 🔉 🖸	አት 🛋 🗵 )	(H ₩X ) (C ) (M	X2 XB 🗟 🗞 🏘	· []			
	r Time Bar: 0	~		Pointer: 84.61 ns	Interval: 84.61 ns	Start: 0 ps	End:	400.0 ns
	Name	Value at 0 ps	0 ps 0 ps	160.0 ns	320.0 ns	480.0 ns	640.0 ns	800.0 ns 1
out	f	ВX		****				
in	<b>x</b> 1	В 0						1
in	x2	В 0						
in	<b>x</b> 3	в 0						
4								· · · · · ·

Figure 4. Forcing  $x_1$  to be low from 0 to 400 ns.

	<u>E</u> dit <u>V</u> iew	Simulation <u>H</u> e					Search alte	era.com
	ى 🏡 왿	· J Z XI >	₩ ₩ \€ \&	X2 XB 🔩 👯 🌴	• 🔳 🎼			
	r Time Bar: 0			Pointer: 14.54 ns	Interval: 14.54 ns	Start: 400.0	ns End:	300.0 ns
	Name	Value at 0 ps	0 ps 0 ps	160.0 ns	320.0 ns	480.0 ns	640.0 ns	800.0 ns 1
out	f	ВX					******	*****
in	<b>x</b> 1	в 0						
in	x2	B 0						
in	<b>x</b> 3	В 0						

Figure 5. Forcing  $x_1$  to be high from 400 to 800 ns.

## High Impedance (Z)

This tool forces the selected waveform to the value High Impedance (Z), as shown in Figure 6. The high impedance value represents a signal that has not been set to any specific value—that is, an input pin that is not connected. Forcing output waveforms to have high impedance does not affect the output simulation waveforms.

	Edit <u>V</u> iew	Simulation He		X2 XB 🗟 📽 🖓			Search altera	com
_	r Time Bar: 0			pinter:	Interval:	Start:	End:	
	Name	Value at 0 ps	0 ps 0 ps	160.0 ns	320.0 ns	480.0 ns	640.0 ns	800.0 ns 🖉
out	f	B 0					×	
in	<b>x</b> 1	В 0					+-	
in	x2	В 0						
in	x3	В 0						
4					III			

Figure 6. Setting a signal to high impedance.

Weak Low (L) 🗵 and Weak High (H) 🗵

These tools are used to set a signal to the values Weak Low (L) or Weak High (H), which represents a circuit in which a bidirectional signal is pulled down or up by using a resistor. Examples are shown in Figures 7 and 8.

ile	<u>E</u> dit <u>V</u> iew	Simulation He	lp 🤜				Search alter	a.com
	)역 🐹 관	<u> 노 조 (</u> )	H 🚻 XC XX X	2 🗡 🗟 🦓 🦓	. <b></b>			
laste	r Time Bar: 0	ps V	Veak Low (L) Poin	ter: 232.67 ns	Interval: 232.67 ns	Start: 200.0 n	s End: 40	00.0 ns
	Name	Value at 0 ps	0 ps 0 ps	160.0 ns	320.0 ns	480.0 ns	640.0 ns	800.0 ns
out	f	ВX		******	*****		*******	*****
in	<b>x</b> 1	В 0			L )			
in	x2	В 0						
in_	<b>x</b> 3	В 0						
٠.		,						· · · · ·

Figure 7. Changing the *x1* signal to be weak low from 200 to 400 ns.

ile	<u>E</u> dit <u>V</u> iew	Simulation <u>H</u> e	lp 🤜				Search alt	era.com
	역 👗 🕑	<u>ት  ጂ  (፲</u>	₩ XC X	X2 XB 🔩 🔩 🌴	· 🔜 晄			
	r Time Bar: 0		~~~	) hter: 15.86 ns	Interval: 15.86 ns	Start: 400.0 i	ns End:	600.0 ns
	Name	Value at 0 ps	0 ps 0 ps	160.0 ns	320.0 ns	480.0 ns	640.0 ns	800.0 ns 4
out	f	вх			*********	*****		
in	<b>x</b> 1	в 0				н		
in_	x2	В 0						
in	x3	В 0						
4								

Figure 8. Changing the *x1* signal to be weak high from 400 to 600 ns.

#### Invert 🐺

This tool inverts the value of a selected waveform, as shown in Figure 9. Low signals become high, weak low signals become weak high, and vice versa for both cases. The Invert tool has no effect on a signal that is set to high impedance or unknown.

🕥 Sim	ulation Wave	eform Editor - D;	/simulation_intro	o/majority3 - majority	/3 - [majority3.vwf]*	_		- • ×
<u>F</u> ile	<u>E</u> dit <u>V</u> iew	Simulation <u>H</u> el	p 🤜				Search alt	era.com 🜖
	🔍 👗 😃	ᆔᆂᄣ		🧟 🔀 🐝 🦓 🖉	1111日間			
Master	Time Bar: 0	ps	Invert Po	inter: 31.73 ns	Interval: 31.73 ns	Start: 100.0 n	s End:	260.0 ns
	Name	Value at 0 ps	0 ps 0 ps	160.0 ns	320.0 ns	480.0 ns	640.0 ns	800.0 ns 🛸
out	f	BX						
in_	x1	В 0						
<u>in</u>	x2	B 0						
<u>in</u>	<b>x</b> 3	B 0						
•		Þ	•					
Overw	rite Invert						10	0% 00:01:52

Figure 9. Inverting the *x1* signal from 100 to 260 ns.

## Count Value 🗵

This tool allows a waveform to be partitioned into sections, in which the value is incremented by a specified amount. The Count Value tool can only be applied to a single waveform or a grouped waveform (see section B.1). The options that are available when using the Count Tool are illustrated in Figure 10.

Radix:	Binary
Start value:	000
Increment by:	1
Count type	
Binary	
<ul> <li>Binary</li> <li>Gray co</li> </ul>	de
· ·	

Figure 10. Options available for for the Count Value tool.

As an example, Figure 11 shows the 3-bit input signal called *count* set to increment by one every 100 ns.

ile	<u>E</u> dit <u>V</u> iew	Simulation <u>H</u> e	lp	-													Sea	rch al	tera.c	om	
	역 🐹 🕹	ት 🚄 🌾	H H	<u>x</u> Xe	<u>xa</u> x <u>a</u>	Xe II	ે રે	n n	<b></b> 8	Ŗ											
laste	r Time Bar: 0	ps			Pointe	er: 563	.17 ns		Inte	erval:	563.1	7 ns		Start:	0 ps			End:	800.0	) ns	
	Name	Value at 0 ps		ps ps		160.0	ns		3	320. <mark>0</mark>	ns		48	).0 ns		e	640.0 r	ıs		800.0	) ns
in	⊳ count	B 000	Ī	000	ТX	001	ΞX	01		C	011	X	100	ĹΧ	101	TX.	110	)	ХŤ	111	D

Figure 11. An example of using the Count Value tool.

### Overwrite Clock 💆

This tool is used to generate a periodic waveform, which is often used as a clock signal. The options available when using the Overwrite Clock tool are shown in Figure 12.

Clock			×
-Base w	aveform o	on time period	
Period:	200.0		ns 🔻
Offset:	0.0		ns 🔻
Duty cy	cle (%):	50	×
		ок	Cancel

Figure 12. Options available for the Overwrite Clock tool.

In the example of Figure 13, the  $x_3$  signal has been generated with a period of 200 ns, an offset of 0 ns, and a duty cycle of 50%.

ile	<u>E</u> dit <u>V</u> iew	Simulation <u>H</u> e	ilp 🤜				Search al	tera.com
	ى 🏡 🝳	. <mark>ዲ (</mark> ፲ )	₩ 💥 🗡	X2 XB 🗟 👯 🏘	- 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1			
laster	r Time Bar: 0	ps	• •	Pointer: 1.32 ns	Interval: 1.32 ns	Start: 0 ps	End:	800.0 ns
	Name	Value at 0 ps	0 ps 0 ps	160.0 ns	320.0 ns	480.0 ns	640.0 ns	800.0 ns 🗳
out	f	BX		******	*****	******	******	******
n	<b>x</b> 1	B 0					-	
n	x2	В 0						
n	x3	в 0						
٠.								

Figure 13. An example of using the Overwrite Clock tool.

#### Arbitrary Value

This tool allows a signal to be set to an arbitrary value, which is particularly useful for specifying the value of a multibit waveform. The options available when using the Arbitrary Value tool are shown in Figure 14.

0	Arbitrary Value	×
	Arbitrary Value	
	Radix:	Binary -
	Numeric or named value:	111 -
		OK Cancel
L		

Figure 14. Options available for for the Arbitrary Value tool.

As an example, in Figure 15 the *count* signal is set to three different arbitrary binary values as specified by the

user.

	(iew <u>S</u> imulation <u>H</u> e & ① <u>土 ヱ</u> X正 )		X2 XB 📌 👯 🚈			Search	altera.com
laster Time B		$\square$	Pointer: 216.81 ns	Interval: 216.81 ns	Start: 180.0	ns End:	300.0 ns
Nan	ne Value at 0 ps	0 ps 0 ps	160.0 ns	320.0 ns	480.0 ns	640.0 ns	800.0 ns 🗳
b cour	nt B 000		00 X 111		000		

Figure 15. The Arbitrary Value tool is used to set values for the *count* signal.

## Random Values 🗵

This tool assigns random values to the selected waveform(s), with several options as shown in Figure 16.

0	Random Values
	Generage random values
	Every grid interval
	<ul> <li>Every half grid interval</li> </ul>
	At random intervals
	At fixed intervals
	Interval period:
	OK Cancel

Figure 16. Various options available for the Random Value tool.

For example, in Figure 17, the signal  $x_1$  has been given random values.

ile	<u>E</u> dit <u>V</u> iew	Simulation H	elp 🤜				Search all	tera.com
	ى 🏡 🍳	· 뉴 프 )도	xe 💥 xe xa	X2 XB 🔩 👯 🌴	a 🚮 🎼			
laste	r Time Bar: 0	ps		Pointer: 672.9 ns	Interval: 672.9 ns	Start: 140.0	ns End:	320.0 ns
	Name	Value at 0 ps	0 ps	160.0 ns	320.0 ns	480.0 ns	640.0 ns	800.0 ns 4
out	f	ВX			****			
in_	<b>x</b> 1	В 0						
in	x2	в 0						
in	x3	в 0						
4					· · · · · · · · · · · · · · · · · · ·			· · · · · ·

Figure 17. An example of the Random Value tool being used.

#### Snap to Grid 🛋

This option allows selections made with the Selection Tool to snap to the light grey grid lines running vertically down the waveform display. This option can be toggled on and off by pressing the Snap to Grid button. It is set to on by default. Figure 18 shows an example of the Selection Tool being used with the Snap to Grid option turned off.

ile	<u>E</u> dit <u>V</u> iew	<u>Simulation</u> <u>H</u> e	lp 🤜				Search a	ltera.com
	🔍 🐹 😃	- ት 🚄 🗵 🕽	H W XC X	X2 XB 💐 💐 🆓	ei ₽;			
lastei	Time Bar: 0	ps		Pointer: 538.05 ns	Interval: 538.05 ns	Start: 163.93	ns End:	330.5 ns
	Name	Value at 0 ps	0 ps 0 ps	160.0 ns	320.0 ns	480.0 ns	640.0 ns	800.0 ns
out	f	ВX			*****		******	******
n	x1	В 0						
n	x2	В 0						
n	<b>x</b> 3	В 0						
۰.								4

Figure 18. An example of the Snap to Grid option turned off.

### Snap to Transition 🔢

This option allows the Selection Tool to automatically extend a selection to the first transition encountered on both sides of the selection of one or more waveforms. For example, with the Snap to Transition option turned on, the Selection Tool rectangle shown in Figure 19 would be expanded to create the selections illustrated in Figure 20. This option can be toggled on and off by pressing the Snap to Transition button, and is set to off by default.

ile	<u>E</u> dit <u>V</u> iew	Simulation He	elp 🤜				Search alter	ra.com
	🔍 🐹 🕑	ト <u>ニ</u> (正)	(H INK )(C )(Q	X2 XB 💐 👯 🎇 🊈	a 📰 🔛			
Master	r Time Bar: 0	ps		Pointer: 278.94 ns	Interval: 278.94 ns	Start:	End:	
	Name	Value at 0 ps	0 ps 0 ps	160.0 ns	320.0 ns	480.0 ns	640.0 ns	800.0 ns
out	f	вх		****	****			*****
in	x1	В 0						
in	x2	В 0						
in	x3	в 0						
						1.011111		

Figure 19. Making a selection with the Snap to Transition option enabled.

	<u>E</u> dit <u>V</u> iew	Simulation <u>H</u> e	lp 🤜			3	Search alter	a.com
	역 🐹 🕑	<u>ት 👗 🌾</u>	H W XC X	X2 XB 🗞 🗞 🦄	ia   🛲 📴			
Master	r Time Bar: 0	ps	F F	Pointer: 11.9 ns	Interval: 11.9 ns	Start: 200.0	ns End: 5	00.0 ns
	Name	Value at 0 ps	0 ps 0 ps	160.0 ns	320.0 ns	480.0 ns	640.0 ns	800.0 ns
out	f	вх		*******	******	******	*******	*****
in	x1	В 0						
	x2	В 0						
in_								
in_	<b>x</b> 3	B 0		┛┊┊┊┆┣┷				1 1 1 1

Figure 20. The expanded selection resulting from Figure 19.

# **B** Using Multibit Signals

This section describes features of the Simulation Waveform Editor that are useful for dealing with multibit signals.

### **B.1 Grouping and Ungrouping Signals**

Individual signals can be grouped together to create a multibit waveform. This is done by first selecting the desired waveforms by clicking on their names in the left side of the Waveform Editor with the key Ctrl pressed as indicated in Figure 21. Then, as shown in the figure, the grouping of signals is done by right-clicking on the selection and choosing Grouping > Group....

🕤 Sim	ulation Wave	eform Edit	tor - D	):/simulation_intro/majority3 - (	majority	3 - [majority3.vwf]*				
<u>F</u> ile	<u>E</u> dit <u>V</u> iew	<u>S</u> imulatio	n <u>H</u>	elp 🤜				Se	arch altera.c	om 🤇
	🔍 👗 😃	ት <mark>ጀ</mark>	XE	XH 🚻 XC XX X2 XB 🛃	n Za	ai 📴				
Master	r Time Bar: 0	ps		Pointer: 1.32 ns		Interval: 1.32 ns	Start: 0 ps		End: 800.0	) ns
	Name	Valu 0		0 ps 160.0 ns 0 ps		320.0 ns	480.0 ns	640.0	ns	800.0 ns 🌋
in_	x1	В 0	×	Delete	Del	The second se				
<u>in</u>	x2	В 0		Insert Node or Bus						
<u>in</u>	<b>x</b> 3	В 0		Grouping	•	Group				
				Reverse Group or Bus Bit Order		Ungroup				
				Radix	•					- + -
Create	a group from t	he selecte	- PL	Properties					100%	00:01:52

Figure 21. An example of grouping signals.

In the options dialogue that opens, illustrated in Figure 22, a name must be assigned to the group, as well as a radix. In the example shown, the name *count* has been chosen with a binary radix.

S Group	×
Group name:	count
Radix:	Binary
🔲 Display gray 🛛	code count as binary count
(	OK Cancel

Figure 22. Select a name and radix for the group of signals.

The resulting group of signals is shown in Figure 23. The multibit waveform can be expanded in the waveform editor to display its individual signals.

ile	<u>E</u> dit <u>V</u> iew	Simulation He	ilp 🤜				Search altera.	com
	) <mark>Q</mark>   🔉 🕑	Λ <mark>Ζ</mark> χΈ χ	(H WY ) C X X	2 🖉 🗟 🗞 🦓				
laste	er Time Bar: 0	ps	Poi	nter: 0 ps	Interval: 0 ps	Start: 0 ps	End: 0 ps	3
	Name	Value at 0 ps	0 ps 0 ps	160.0 ns	320.0 ns	480.0 ns	640.0 ns	800.0 ns '
-	count	B 000	000	001 X 0	10 X 011 X	100 X 101	X 110 X	111
in	x1	В 0						
in	x2	В 0						
in	<b>x</b> 3	В 0						
۰.		•						

Figure 23. An example of expanding a multibit signal.

A multibit signal can be ungrouped by right-clicking on the group of signals and selecting Grouping > Ungroup....

It is also possible to create hierarchical groupings of signals as illustrated in Figure 24. In this example, the two bit signal called *level2* is combined with the signal called  $x_3$  to create the three bit signal called *level1*. It is only possible to group and ungroup top-level signals.

				ntro/majority3 - majorit	/3 - [majority3.vwf]*			- 0 X
File	Edit View	Simulation He	lp 🤜				Search alte	ra.com 🕠
	) <b>e</b>   💥 😃	, <u>z</u> ∖ī ∖	Ξ₩X XC X	8 X2 XB 🕏 👯 🖓	<b>■</b> 194			
Maste	r Time Bar: 0 p	)S	• •	Pointer: 783.95 ns	Interval: 783.95 ns	Start: 0 ps	End: 0	ps
	Name	Value at 0 ps	0 ps 0 ps	160.0 ns	320.0 ns	480.0 ns	640.0 ns	800.0 ns 🔦
-	⊿ level1	B 00, 0		00, 0	01, 0	Z0, 0 X Z0, 1	X 11, 0 X	11, 1
-	⊳ level2	B 00		00	01 X	Z0	X 11	
in_	<b>x</b> 3	В 0						
4					<u> </u>			
		,					04	% 00:00:00

Figure 24. An example of hierarchical groups.

It is also possible to group input and output signals, as shown in Figure 25.

ile	Edit View	Simulation He	elp 🤜				Se	arch alter	a.com
	9 🔍 🖄 🖞	, 九 프 )(트 )	KH INV KC KE	X2 XB 💐 👯 🍓 🊈	<u>■</u> 账				
laste	r Time Bar: 0	ps	• •	Pointer: 642.49 ns	Interval: 642.49 ns	Start:	0 ps	End: 0	ps
	Name	Value at 0 ps	0 ps	160.0 ns	320.0 ns	480.0 ns	640.0	ns	800.0 ns '
5	⊿ inout	B 000X		000X	010X X	100X X	101X 11	ox X	111X
in_	<b>x</b> 1	В 0							
in_	x2	В 0							
in_	x3	В 0							
out	f	ВX		********	*****	******	*******	$\times\!\!\times\!\!\times$	*****
-									1000

Figure 25. An example of grouping input and output signals.

#### B.2 Reverse Group or Bus Bit Order

In Figure 23, the three bit signal count is displayed as the 3-tuple  $x_1x_2x_3$ . It is possible to reverse the order in which the bits are displayed as illustrated in Figure 26. This is done by right-clicking on the name of the multibit signal and selecting Reverse Group or Bus Bit Order, as seen in the figure.

ile	<u>E</u> dit <u>V</u> iew	Simulation H	elp	•					Search alter	a.com
	🔍 🐹 😃	<u>ት 🖬 X</u> ፤	<mark>∕</mark> ₩ Į	💥 XC XX X2 XB 😪 💐	🊈 🚮 B	账				
laste	r Time Bar: 0	ps		Pointer: 0 ps	Inte	erval: 0 ps	Start:	0 ps	End: 8	00.0 ns
	Name	Value at 0 ps		ps 160.0 ns	3	320.0 ns	480.0 ns	64	40.0 ns	800.0 ns
-	▲ count	B 000		000		10 X	100 🔨	101 X	110	111
n	x1	В 0	×	Delete	Del					
n	x2	В 0		Insert Node or Bus	H					
n	<b>x</b> 3	В 0		Grouping Reverse Group or Bus Bit Orde						
				Radix						

Figure 26. Reversing the bit order on a group of signals.

The effects of the bit reversal can be seen in Figure 27. The count waveform is now displayed as the 3-tuple  $x_3x_2x_1$ .

ile	<u>E</u> dit <u>V</u> iew	Simulation <u>H</u> e	lp 🤜				Search altera	.com
	🝳   👗 ଏ	- <mark>አ </mark> ( )	H ₩ XC	XT X2 XB 🗟 👯 🖓 🚈	<b>武</b> 武			
laste	r Time Bar: 0	ps		Pointer: 785.27 ns	Interval: 785.27 ns	Start: 0 ps	End: 80	0.0 ns
	Name	Value at 0 ps	0 ps 0 ps	160.0 ns	320.0 ns	480.0 ns	640.0 ns	800.0 ns
-	⊿ count	B 000		000 X	010 X	001 X 101	χ 011 χ	111
in	x3	В 0						
in	x2	В 0						
in	<b>x</b> 1	B 0						
۰.			• •					•

Figure 27. The result of reversing the bit order in Figure 26.

Copyright ©2015 Altera Corporation.