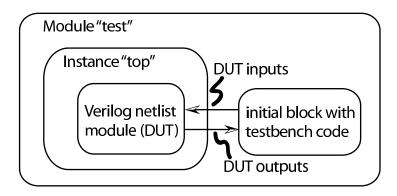
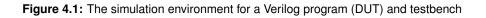
Chapter 4

Verilog Simulation





20 CHAPTER 4: Verilog Simulation

```
`timescale lns / 100ps
module test;

wire Cout;
reg Cin;
wire [1:0] Sum;
reg [1:0] A;
reg [1:0] B;

twoBitAdd top(Cout, Sum, A, B, Cin);
 `include "testfixture.verilog"
endmodule
```

Figure 4.2: Verilog code for a DUT/testbench simulation environment

X Set	up Enviror	iment				
ОК	Cancel		ĸ			Help
Run I	Directory	add2bit.rur	1			
Desig	jn To Simu	late				
Libr	ary	labl		Cell	add2bitj	
Viev	v	schematič				
Libi	rary Mana(jer Browser	Hierarchy I	Editor Browser		

(Copyright © 2005, 2010, Cadence Design Systems, Inc. All rights reserved worldwide. Reprinted with permission.)

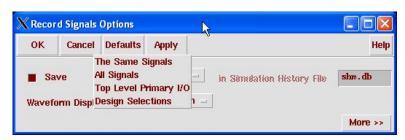
Figure 4.3: Dialog box for initializing a simulation run directory

റ	-1
_	
_	

		lab1-add2bit-schematic add2bit.run1	I Inactive 1
File	Setup	Stimulus Simulation Info Debug	Help
••	*		
×	*		
»;	Ċ		
		▶	
	-		
r.	À.		
	\bigotimes		
*	۹		
*			
<u>j</u>	G.		
*	Ch	Verilog_XL>	

Figure 4.4: The initial Verilog-XL simulation control window

22 CHAPTER 4: Verilog Simulation



(Copyright ©2005, 2010, Cadence Design Systems, Inc. All rights reserved worldwide. Reprinted with permission.)

Figure 4.5: The Record Signals dialog box



(Copyright © 2005, 2010, Cadence Design Systems, Inc. All rights reserved worldwide. Reprinted with permission.)

Figure 4.6: Dialog to create a new testfixture template

OK	Canc	el Apply			н
Aode Copy		elect 🔷	Edit 4	♦ Copy ◇ Delete	
Dire	ctory	/home/e	1b/10	_CAD/cadence1.5/add2bit.run1	
File	Name	testfix	ture.	veriloq	
		testfix testfix		No. 1	
		testfix testfix verilog	ture.	veritime	View File
Copy	To:				
Run	Direct	tory		/home/elb/IC_CAD/cadence1.	5/add2bit.run1
File	Name	testfixture.new			
Design In		tance Pa	th	test.top	
~	ent Te	est. Fixtur	e		

Figure 4.7: The Verilog-XL Stimulus Options form

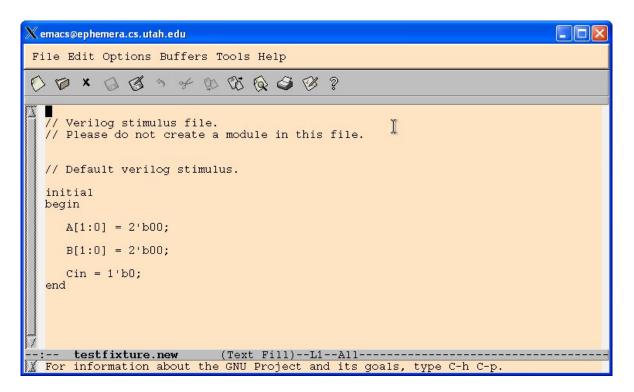


Figure 4.8: Testfixture template for the two-bit adder

```
25
```

```
○ ♥ × ◎ ♂ > ★ ◎ ♡ Q ♀ ♡ ?
 // Verilog stimulus file.
 // Please do not create a module in this file.
 // Default verilog stimulus.
 initial
 begin
    A[1:0] = 2'b00; // 2'b00 means "two bits in binary with value 00"
    B[1:0] = 2'b00;
    Cin = 1'b0;
 $display("Starting simulation..."); // $display is a "printf" in Verilog
                            _____
 //-----
 #20 // Wait for 20 simulation time units
     // Then print out the curent values...
 $display("A=%b B=%b Cin=%b, Cout-Sum=%b%b", A, B, Cin, Cout, S);
     // Check to see if the outputs are correct!
 if (S != 2'b00) $display("ERROR: Sum should be 00, is %b", S);
 if (Cout != 0) $display("ERROR: Cout should be 0, it %b", Cout);
 //-----
 // Change the input, wait for some simulation time, check again
 A = 2'b01;
 #20
 $display("A=%b B=%b Cin=%b, Cout-Sum=%b%b", A, B, Cin, Cout, S);
 // Check outputs with concatenated values...
 if ({Cout,S} != 3'b001)
    $display("ERROR: Cout-Sum should be 001, is %b", {Cout,S});
 //------
 B = 2'b11;
 #20
 $display("A=%b B=%b Cin=%b, Cout-Sum=%b%b", A, B, Cin, Cout, S);
 if ({Cout,S} != 3'b100)
    $display("ERROR: Cout-Sum should be 100, is %b", {Cout,S});
 $display("Simulation finished... ");
 end
```

Figure 4.9: An example testfixture for the two-bit adder

```
// Default Verilog stimulus.
integer i,j,k;
initial
begin
   A[1:0] = 2'b00;
  B[1:0] = 2'b00;
Cin = 1'b0;
$display("Starting simulation...");
for(i=0;i<=3;i=i+1)</pre>
begin
 for(j=0;j<=3;j=j+1)</pre>
  begin
   for(k=0;k<=1;k=k+1)
    begin
     #20
      $display("A=%b B=%b Cin=%b, Cout-Sum=%b%b", A, B, Cin, Cout, S);
      if \{\{Cout, S\} \mid = A + B + Cin\}
          $display("ERROR: Cout-Sum should equal %b, is %b",
                     (A + B + Cin), {Cin,S});
      Cin=~Cin; // invert Cin
     end
    B[1:0] = B[1:0] + 2'b01; // add the bits
   end
 A = A+1; // shorthand notation for adding
 end
$display("Simulation finished... ");
end
```

Figure 4.10: Another testfixture for the two-bit adder

```
// Default Verilog stimulus.
reg [1:0] ainarray [0:4]; // define memory arrays
reg [1:0] binarray [0:4]; // to hold input and result
reg [2:0] resultsarray [0:4];
integer i;
initial
begin
/* A simple Verilog testfixture for testing a 2-bit adder */
   $readmemb("ain.txt", ainarray); // read values into
$readmemb("bin.txt", binarray); // arrays from files
   $readmemb("results.txt", resultsarray);
   A[1:0] = 2'b00; // initialize inputs
   B[1:0] = 2'b00;
   Cin = 1'b0;
   $display("Starting...");
   #10
   $display("A = %b, B = %b, Cin = %b, Sum = %b, Cout = %b",
             A, B, Cin, Sum, Cout);
   for(i=0; i<=4; i=i+1) // loop through all values in arrays</pre>
   begin
      A = ainarray[i]; // set the inputs
      B = binarray[i]; // from the memory arrays
      #10
       $display("A = %b, B = %b, Cin = %b, Sum = %b, Cout = %b",
                 A, B, Cin, Sum, Cout);
      // check against results array
      if ({Cout,Sum} != resultsarray[i])
         $display("Error: Sum should be %b, is %b instead",
                     resultsarray[i],Sum);
   end
   $display("...Done");
   $finish;
end
```

Figure 4.11:	A testfixture	using	values	from	external	files

01	01	010
10	10	100
11	11	110
00	11	011
01	11	100
(a) ain.txt	(b) bin.txt	(c) results.txt

Figure 4.12: Data files used in Figure 4.11



(Copyright ©2005, 2010, Cadence Design Systems, Inc. All rights reserved worldwide. Reprinted with permission.)

Figure 4.13: Dialog box for re-netlisting a previously netlisted design

X icfb - Log: /home/	elb/CDS.log			
File Tools Options				Help 1
(ir CELL NAME invXl nor2 nand2 xor2 End Net1	tlist Configuration Info cremental data only) VIEW NAME behavioral behavioral behavioral behavioral ist Configuration Info ell - add2bit, view - sch 28 14:36:47 2006	NOTE *Stopping View* *Stopping View* *Stopping View* *Stopping View*	•	
Start Plotting			4	
mouse L:	M :		R:	

(Copyright © 2005, 2010, Cadence Design Systems, Inc. All rights reserved worldwide. Reprinted with permission.)

Figure 4.14: The netlisting log for the two-bit adder

y

		lab1-add2bit-schematic add2bit.run1 test 0.00 ns Stopped 1
File	Setup	Stimulus Simulation Info Debug Help
**		For technical assistance please contact the Cadence Response Center at
×	►	1-877-CDS-4911 or send email to support@cadence.com
►I	Ů	For more information on Cadence's Verilog-XL product line send email to talkv@cadence.com
1	÷.	Compiling source file "/home/elb/IC_CAD/cadence1.5/add2bit.run1/testfixtur Compiling included source file "testfixture.new" Continuing compilation of source file "/home/elb/IC_CAD/cadence1.5/add2bit
	-11	Compiling source file "/uusoc/facility/cad_common/local/Cadence/lib/UofU_I Compiling source file "/uusoc/facility/cad_common/local/Cadence/lib/UofU_I Compiling source file "/uusoc/facility/cad_common/local/Cadence/lib/UofU_I
1	Å	Compiling source file "/uusoc/facility/cad_common/local/Cadence/lib/UofU_I Compiling source file "ihnl/cds0/netlist"
Z	\bigotimes	Compiling source file "ihml/cdsl/netlist" Compiling source file "ihml/cds2/netlist" Compiling source file "ihml/cds3/netlist"
	Đ	Highest level modules: test
* .	۲	Type ? for help C1 >
P	Ŧ	
	JAN 2	Cl >

Figure 4.15: The Verilog-XL window after netlisting

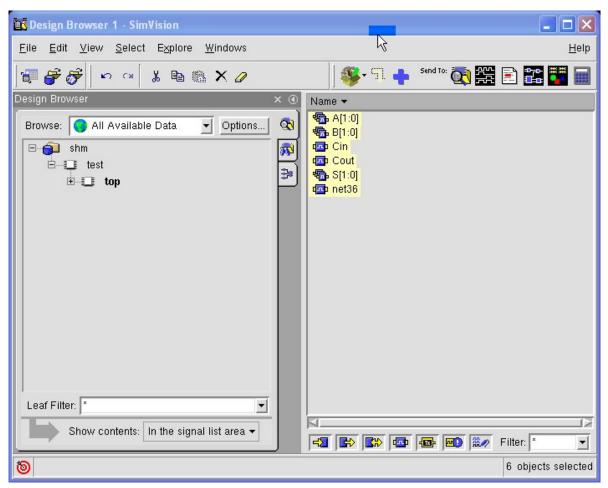
		lab1-add2bit-schematic add2bit.run1 test	Inactive	2.
File	Setup	Stimulus Simulation Info Debug	Н	elp
••	*	Compiling source file "si.inp" Compiling source file "/uusoc/facility/cad_common/local/Cadence/li	b/VofU_D	
×	۲	Compiling source file "/uusoc/facility/cad_common/local/Cadence/li Compiling source file "/uusoc/facility/cad_common/local/Cadence/li Compiling source file "/uusoc/facility/cad_common/local/Cadence/li	b/UofU_D	
»)	Ö	Compiling source file "ihml/cds0/netlist" Compiling source file "ihml/cds1/netlist"	1,0010_1	
	×.	Compiling source file "ihml/cds2/netlist" Compiling source file "ihml/cds3/netlist" Highest level modules:		
		test Type ? for help		
1	i.	c1 > .		
<u>j</u>	\bigotimes	Starting simulation A=00 B=00 Cin=0, Cout-Sum=000 A=01 B=00 Cin=0, Cout-Sum=001		
	۲	A=01 B=11 Cin=0, Cout-Sum=100 Simulation finished 0 simulation events (use +profile or +listcounts option to count)	+ 66 acc	
۲.		CPU time: 0.0 secs to compile + 0.0 secs to link + 0.0 secs in sin End of Tool: XVERILOG-XL 05.60.001-p Mar 28, 2006 14:45:52		
<u>j</u>	Ś	Verilog-XL	2	12
1	n 2			_

Figure 4.16: Result of running with the testbench from Figure 4.9

🛱 Waveform 1 - SimVision					
	For <u>m</u> at <u>W</u> indows				Help
	•			Send To: 🐌	
Search Names: Signal -		imes: Value 🕶	A. A.		
1.44			ni kun nu ka	Time: @ 7 0 - 1 0	00,000,000,i 🗹 🔍 🗮 👫
x_1 TimeA \checkmark = 0 × • Baseline = 0				I [0.1,0	**************************************
Cursor-Baseline = 0	Baseline = 0				
	Cursor - 0	200,000,000,000ps	400,000,000,000ps	600,000,000,000ps	800,000,000,000ps
Name +					
		100,000,000,000 1300,000,0	000,000 500,000,00	0,000	1,000,000,000,000ps
0					0 objects selected

(Copyright ©2005, 2010, Cadence Design Systems, Inc. All rights reserved worldwide. Reprinted with permission.)

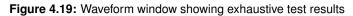
Figure 4.17: Waveform window without any signals



(Copyright © 2005, 2010, Cadence Design Systems, Inc. All rights reserved worldwide. Reprinted with permission.)

Figure 4.18: Design browser with signals selected

Waveform 1 - SimVision	
<u>File Edit View Explore Format Windows</u>	<u>H</u> elp
🧬 🧬 🗠 🖂 👔 🛍 🗙 🖓 🦏 🛄 🎋	🖀 📑 🗐
Search Names: Signal 🗸 🚺 🖌 🛝 Search Times: Value 🗸 🕺	
TimeA → = 0 → ps → ft&- ft ft <td></td>	
Baseline = 0 Baseline = 0 Cursor-Baseline = 0 TimeA = 0 Name ✓ Cursor ✓ 0	Ops 800,►
3≥ g→3g, β[1:0] ¹ h 0 0 <u>(1 /2 /3 (0 /1 /2)))))))))))))))))))))))))))))))))</u>	
φ-%- s(1.0) '▶× 0(11)2)2(2)3(3)0(12)2(2)3(3)0(0(1)2)2(3)3(0)0(1)2)(3)3(0)0(1)2)(3)3(0)0(1)(1)(1)(1)(1)(1)(1)(1)(1)(1)(1)(1)(1)(
0	objects selected



Waveform 1 - SimVision											
<u>F</u> ile <u>E</u> dit <u>V</u> iew Ex <u>p</u> lore F	or <u>m</u> at <u>W</u> indows										<u>H</u> el
🚰 😚 🗠 🖂 🐰 🐚 I	8 × 🗟 🖷	. 🤹 🍂					%	Send Send	^{ro:} 👧 🚆	i 🗈 🔀	
Search Names: Signal 🕶	<u> </u>	M Search	Times: Value 🔻								
x ₁ TimeA 💌 = 0	ps 🗾 🗗 🚺	1 4 4					► Tim	e: 🖁 🔁 🗐	7,200ps : 4	408,4(🚽 📿	+-
Baseline = 0 Cursor-Baseline = 0											
Constant and the second second	Cursor 🕶	200,000ps	240,00	Ops	280,000)ps	320,000p)S	360,000)ps	40►
₩ 	'h 0	1	<u> </u>	<u> </u>	111	<u></u>	2			<u>.</u>	
🔧 🖽 📲 🖓 🗃	'h 0)[1)(2)(3		<u>(</u> 0)(1)(2
	0										
	х								1		
Ė 🎫 bus(Cout, S)	'h x	2 2	(3%3	XXX4	4	5)))(2	3	3)())(4	4
		⊧ <mark>⊲</mark> ∙	100,000			00,000	500,000	600,000) 700),000 844,80	00ps >
)										0 object	s selec

Figure 4.20: Waveform window showing outputs as a bus

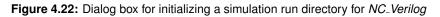
-Printer		
Command: -	lpr –l	1
	7	
Print range		
	Variables visible in waveform window	
	Selected variables	
	All variables	
	[◆ Times visible in waveform window	
	↓ Specified times from 197,200ps to 408	,400ps
	All times	
	sform 1 - SimVision	
Title: Wave Designer Name: Company Name:	eform 1 - SimVision	
Title: Wave Designer Name:	əform 1 - SimVision	
Title: Wave Designer Name: Company Name:	sform 1 - SimVision	
Title: Wave Designer Name: Company Name: Note:	eform 1 - Sim∀ision Paper Size: Letter (8.5 × 11)	
Title: Wave Designer Name: Company Name: Note:		
Title: Wave Designer Name: Company Name: Note:	Paper Size: Letter (8.5 × 11)	
Title: Wave Designer Name: Company Name: Note: Company Name: Print paper	Paper Size: Letter (8.5 × 11) Orientation: Landscape Colors: Black on white	

(Copyright © 2005, 2010, Cadence Design Systems, Inc. All rights reserved worldwide. Reprinted with permission.)

Figure 4.21: Printing dialog box for SimVision

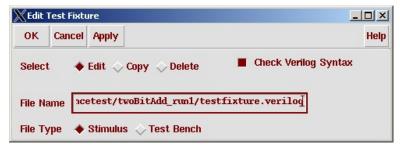
nds Setup Re	sults	
Run Directory		
/home/elb/IC	_CAD/cadencetest/	/twoBitAdd_runl
Top Level Desig	In	
Library	Cell	View
tutorial	twoBitAdd	schematic
	Browse	Hierarchy Edito
	IS	
Simulate Option	1.7	

(Copyright ©2005, 2010, Cadence Design Systems, Inc. All rights reserved worldwide. Reprinted with permission.)



Reco	rd Signal	s Setup		
ок	Cancel	Apply		Help
Trace		All Signals	-	
In the	Scope	test		
and				
	 ↓ 1 ♦ All 			
Subsc)own the Hie	erarchy	

Figure 4.23: The Record Signals dialog box



(Copyright © 2005, 2010, Cadence Design Systems, Inc. All rights reserved worldwide. Reprinted with permission.)

Figure 4.24: Dialog to create a new testfixture template

🚾 Console - SimVision	
<u>File Edit View Simulation Windows</u>	<u>H</u> elp
] 🚰 💞 🕺 🛍 🛍 🗙	
	5
<pre>csim> csim> source /uusoc/facility/cad_tools/Cadence/IUS56ISR200603091112/tools/i csim> database -open shmWave -shm -default -into shm.db reated default SHM database shmWave csim> probe -create -shm test -all -depth 1 reated probe 1 csim></pre>	.nca/fil
SimVision simulator	

Figure 4.25: SimVision Console window

39

Design Browser 1 - Sim¥ision		
<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>S</u> elect Explore Sim <u>u</u> lation <u>W</u> indo	WS	<u>H</u> elp
ti 🚰 ở ∽ ↔ š 🖻 🛍 X 🧷	🐝 - 🖓 🔶 Send To: 🔃	😤 🗈 🖀 🕎 🖩
TimeA ▼ = 0 ▼ ps ▼ 15t -	Search Times: Value 🗸	
🖬 - 📖 🔣 🔂 🔂 🛛 🐠 🗐 🞯 0 + 0		
Design Browser 🛛 🗙 🤆	Name 🕶	Value (as recorded) 🕶
Browse: All Available Data Options	B[1.0]	'h x 'h x x 'h x x Filter: *
Time Value		1 object selected

(Copyright © 2005, 2010, Cadence Design Systems, Inc. All rights reserved worldwide. Reprinted with permission.)

Figure 4.26: SimVision Design Browser window

₩a	weform 1 - Sim¥ision															_	
<u>F</u> ile	<u>E</u> dit <u>V</u> iew Explore For <u>m</u> a	at Sim <u>u</u> lation	<u>W</u> indows														<u>H</u> elp
8	🔗 🗠 🕹 🛍 🚳	× 🕷 🖷									% -		Send To:	Q) 🗄	1		
Sear	rch Names: Signal 🕶	<u> </u>	🛝 🛛 Search Time	es: Value 🔻													
	TimeA 💌 = 527,200 💌 ps	🖸 📠 - 🕅 🗹	<u> </u>		6)	644,0)00ps +	• 0		Time	9 7	0:6	14,000p:			
×⊙	Baseline = 0 Cursor-Baseline = 527,200ps		Baseline = 0											T	moA -	527,200p	
	Name 🔻	Cursor 🕶	0	100,000ps		200,00	l0ps	3	00,000ps	s	400,0	00ps		500,00		600,	0
E.		1 'h 0	0,1,1,2,	: <u>)3 %3 %0</u>	1)(2	2 3	X 3)(0	X 0)(1	1)/(2)(3	X 3)(0	0 (1		2 3	жо х о	1 1)(2 (2)	3
	····· ➡ Cout ····· ➡ Cin 	1 0 'h 1													2		
	⊡	'h 3	(0	<u>^*</u>)(1		<u>/-</u>	<u>"Л" </u>)(2	<u></u>		<u></u>	3			<u>/č</u>	٥,
J															- 142 144. - 144.		
0															0 0	bjects se	lected

(Copyright ©2005, 2010, Cadence Design Systems, Inc. All rights reserved worldwide. Reprinted with permission.)

Figure 4.27: Waveform window showing the output of exhaustive simulation

ок	Cano	:el	Defaults	Help			
Library I	lame		tutorial	-			
Cell Nam	ie	n	and2j				
View Na	me	behavioral					
Tool			Verilog-Edito	r 💷			
Library p	oath file	;					

Figure 4.28: Dialog box for creating a behavioral view

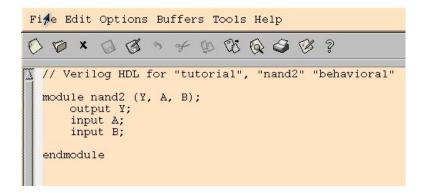


Figure 4.29: Behavioral view template based on the nand2 symbol

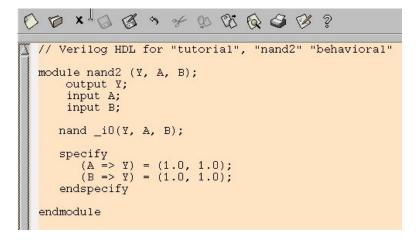


Figure 4.30: Complete behavioral description of a nand2 cell

Fools Design	Help	6
/ Verilog HDL for "UofU_Digital_v1_1", "nand2" "beh	avioral"	
odule nand2 (Y, A, B);		
output Y;		
input A;		
input B;		
nand _iO(Y, A, B);]		
specify		
$(A \Rightarrow Y) = (1.0, 1.0);$		
$(B \Rightarrow Y) = (1.0, 1.0);$		
endspecify		
ndmodule		

(Copyright © 2005, 2010, Cadence Design Systems, Inc. All rights reserved worldwide. Reprinted with permission.)

Figure 4.31: Read only window for simulation of behavioral view

```
// Verilog HDL for "Ax", "see4" "behavioral"
// Four in a row detector - written by Allen Tanner
module see4 (clk, clr, insig, saw4);
   input clk, clr, insig;
   output saw4;
   parameter s0 = 3'b000; // initial state, saw at least 1 zero parameter s1 = 3'b001; // saw 1 one parameter s2 = 3'b010; // saw 2 ones
   parameter s3 = 3'b011; // saw 3 ones
   parameter s4 = 3'b100; // saw at least, 4 ones
   reg [2:0] state, next_state;
   always @(posedge clk or posedge clr) // state register
     begin
         if (clr) state <= s0;
        else state <= next_state;</pre>
     end
   always @(insig or state) // next state logic
     begin
        case (state)
                 s0: if (insig) next_state = s1;
                      else next_state = s0;
                 s1: if (insig) next_state = s2;
                     else next_state = s0;
                 s2: if (insig) next_state = s3;
                      else next_state = s0;
                 s3: if (insig) next_state = s4;
                      else next_state = s0;
                 s4: if (insig) next_state = s4;
                     else next_state = s0;
            default: next_state = s0;
        endcase
     end
// output logic
assign saw4 = state == s4;
endmodule //see4
```

Figure 4.32: A simple state machine described in Verilog: see4.v

44 CHAPTER 4: Verilog Simulation

// Top-level test file for the see4 Verilog code
module test;
// Remember that DUT outputs are wires, and inputs are reg
wire saw4;
reg clk, clr, insig;
// Include the testfixture code to drive the DUT inputs and
// check the DUT outputs
'include "testfixture.v"
// Instantiate a copy of the see4 function (named top)
see4 top(clk, clr, insig, saw4);
endmodule //test

Figure 4.33: Top-level Verilog code for simulating see4 named seetest.v

```
// Four ones in a row detector testbench (testfixture.v)
// Main tests are in an initial block
initial
begin
  clk = 1'b0; // initialize the clock low
  clr = 1'bl; // start with clr asserted
  insig = 1'b0; // insig starts low
   #500 clr = 1'b0; // deassert clear and start running
   // use the send_test task to test the state machine
   send_test(32'b0011_1000_1010_1111_0000_0111_1110_0000);
   send_test(32'b0000_0001_0010_0011_0100_0101_0110_0111);
   send_test (32'b1000_1001_1010_1011_1100_1101_1110_1111);
   send_test(32'b1011_1111_1101_1111_1111_1100_1011_1111);
   // Print something so we know we're done
   $display("\nSaw4 simulation is finished...");
   $display("If there were no 'ERROR' statements, then everything worked!\n");
   $finish;
end
// Generate the clock signal
always #50 clk = ~clk;
// this task will take the 32 bit input pattern and apply
// those bits one at a time to the state machine input.
// Bits are changed on negedge so that they'll be set up for
// the next active (posedge) of the clock.
task send_test;
  input [31:0]pat; // input bits in a 32-bit array
  integer i;
                   // integer for looping in the for statement
 begin
    for(i=0;i<32; i=i+1) // loop through each of the bits in the pat array
      begin
        // apply next input bit before next rising clk edge
        @(negedge clk)insig = pat[i];
          // remember to check your answers!
          // Look at last four bits to see if saw4 should be asserted
          if ((i > 4)
             && ({pat[i-4],pat[i-3],pat[i-2],pat[i-1]} == 4'b1111)
             \&\& (saw4 != 1))
             $display("ERROR - didn't recognize 1111 at pat %d,", i);
          else if ((i > 4))
                  && ({pat[i-4],pat[i-3],pat[i-2],pat[i-1]} != 4'b1111)
                  \&\& (saw4 == 1))
                  $display("ERROR - signalled saw4 on %b inputs at step %d",
                           {pat[i-3],pat[i-2],pat[i-1],pat[i]}, i);
      end // begin-for
         // begin-task
   end
          // send_test
endtask
```



46 CHAPTER 4: Verilog Simulation

--->sim-xl -f test.txt Tool: VERILOG-XL 05.10.004-s Jul 29, 2006 20:50:01 Copyright (c) 1995-2003 Cadence Design Systems, Inc. All Rights Reserved. Unpublished -- rights reserved under the copyright laws of the United States. Copyright (c) 1995-2003 UNIX Systems Laboratories, Inc. Reproduced with Permission. THIS SOFTWARE AND ON-LINE DOCUMENTATION CONTAIN CONFIDENTIAL INFORMATION AND TRADE SECRETS OF CADENCE DESIGN SYSTEMS, INC. USE, DISCLOSURE, OR REPRODUCTION IS PROHIBITED WITHOUT THE PRIOR EXPRESS WRITTEN PERMISSION OF CADENCE DESIGN SYSTEMS, INC. RESTRICTED RIGHTS LEGEND Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 or subparagraphs (c)(1) and (2) of Commercial Computer Software --Restricted Rights at 48 CFR 52.227-19, as applicable. Cadence Design Systems, Inc. 555 River Oaks Parkway San Jose, California 95134 For technical assistance please contact the Cadence Response Center at 1-877-CDS-4911 or send email to support@cadence.com For more information on Cadence's Verilog-XL product line send email to talkv@cadence.com Compiling source file ``see4.v'' Compiling source file ``seetest.v'' Warning! Code following 'include command is ignored [Verilog-CAICI] ``seetest.v'', 6: Compiling included source file ``testfixture.v'' Continuing compilation of source file ``seetest.v'' Highest level modules: test Saw4 simulation is finished ... If there were no 'ERROR' statements, then everything worked! L17 ``testfixture.v'': \$finish at simulation time 13200 1 warning 0 simulation events (use +profile or +listcounts option to count) CPU time: 0.0 secs to compile + 0.0 secs to link + 0.0 secs in simulation End of Tool: VERILOG-XL 05.10.004-s Jul 29, 2006 20:50:02 --->

Figure 4.35: Output of stand-alone Verilog-XL simulation of seetest.v

```
<previous text not included...>
Compiling included source file ``testfixture.v''
Continuing compilation of source file ''seetest.v''
Highest level modules:
test
ERROR - didn't recognize 1111 at pat
ERROR - didn't recognize 1111 at pat
                                               10,
                                               11,
ERROR - didn't recognize 1111 at pat
                                              5,
                                              6,
ERROR - didn't recognize 1111 at pat
ERROR - didn't recognize 1111 at pat
                                             15,
ERROR - didn't recognize 1111 at pat
                                              16,
ERROR - didn't recognize 1111 at pat
                                              17,
18,
ERROR - didn't recognize 1111 at pat
ERROR - didn't recognize 1111 at pat
                                              20,
ERROR - didn't recognize 1111 at pat
                                              21,
ERROR - didn't recognize 1111 at pat
                                              27,
                                              28,
29,
ERROR - didn't recognize 1111 at pat
ERROR - didn't recognize 1111 at pat
ERROR - didn't recognize 1111 at pat
                                              30,
Saw4 simulation is finished...
If there were no 'ERROR' statements, then everything worked!
L17 ``testfixture.v'': $finish at simulation time 13200
1 warning
0 simulation events (use +profile or +listcounts option to count)
CPU time: 0.0 secs to compile + 0.0 secs to link + 0.0 secs in
simulation
End of Tool:
                VERILOG-XL
                              05.10.004-s Jul 29, 2006 21:21:49
```

Figure 4.36: Result of executing a faulty version of see4.v

🔛 Console - Sim¥ision	
<u>F</u> ile <u>E</u> dit <u>V</u> iew Sim <u>u</u> lation <u>W</u> indows	<u>H</u> elp
) 🚰 💞 % 🐚 🛍 🗙	
Console output for Verilog-XL (design test.) This simulation finished on Sun Jul 30 13:37:17 MDT 2006. Press Return or click on the X to remove this console tab.	×
1 warning 0 simulation events (use +profile or +listcounts option to count) CPU time: 0.0 secs to compile + 0.0 secs to link + 0.0 secs in simulation End of Tool: VERILOG-XL 05.60.001-p Jul 30, 2006 13:37:17 SimVision (Done)	<u></u>

Figure 4.37: Control console for stand-alone Verilog-XL simulation using SimVision

<mark>∰Design Browser 1 - Sim¥ision</mark> Eile <u>E</u> dit <u>V</u> iew <u>S</u> elect Explore Sim <u>u</u> lation <u>W</u> indows	3	× Help
🗃 💣 💞 🗠 🖙 🗼 🖻 🛍 🗙 🥒	🛛 🥵 - 🖕 🛉 Send To: 💽	🚟 🖹 🖀 🐺 🗐
13,200,000,(▼ ns ▼ ,ित्ते ▼ 13,200,000,(▼ ns ▼ , 1 →	Search Times: Value 🔻	
Design Browser 🛛 🗙 🛈	Name 👻	Value (as recorded) -
Browse: All Available Data Options		0 1 0 'h 0 'b 000 'b 001 'b 010 'b 011 'b 100 0 'h 0
Show contents: In the signal list area -		Filter: × V
Send selected object(s) to target design browser (click an		4 objects selected

Figure 4.38: Hierarchy browser for the see4 example

₩aveform 1 - Sim¥ision					<u>_ ×</u>
<u>File E</u> dit <u>V</u> iew Ex <u>p</u> lore F	or <u>m</u> at Sim <u>u</u> lation <u>V</u>	<u>V</u> indows			<u>H</u> elp
ở ở 🗠 🗠 🖁 🚱 (e × 🖷 🖷 🐮			- 🛉 50	^{ind To:} 🐼 🚟 🖹 🖀 🐺 🗐
Search Names: Signal 🕶	<u> </u>	Search Times: Value 🕶			
x ₂ TimeA ▼ = 13,200 ▼	- 🛨 🗖 - 🚮 🗉 🖉	.		Time: Set [0 : 6945s 🔄 🔜 🛨 🛶
× (c) Baseline = 0 Cursor-Baseline = 13,200	Is B	aseline = 0			
Name -	Cursor 👻 0	1000s	2000s 3000s	4000s 5	000s 6000s
 Image: state [2:0] Image: state [2:0] Image: state [2:0] 	'h 0 0))())(4_)(0			∠(<u> </u>
	'h 0 0	<u>)(()()</u> 4)(∘	<u> </u>		
insig	0				
	0				
		<u>k</u>		17000 8000 9000	10,000 11,000 13,200s
1					4 objects selected

Figure 4.39: Waveform viewer after running the see4 example

```
---> sim-nc -f test.files
ncverilog: 05.10-s014: (c) Copyright 1995-2004 Cadence Design Systems,
Inc
file: see4.v
      module worklib.see4:v
              errors: 0, warnings: 0
file: seetest.v
       module worklib.test:v
               errors: 0, warnings: 0
              Caching library 'worklib' ..... Done
       Elaborating the design hierarchy:
       Building instance overlay tables: ..... Done
       Generating native compiled code:
               worklib.see4:v <0x3d4ece8f>
                     streams: 5, words: 1771
               worklib.test:v <0x37381383>
                     streams: 6, words: 4703
       Loading native compiled code: ..... Done
       Building instance specific data structures.
       Design hierarchy summary:
                        Instances Unique
              Registers: 2 2
Registers: 7
                                          7
               Scalar wires:
                                 4
3
                                          _
                                        3
              ocks: Initial blocks:
Cont -
                                 1
                                          1
               Cont. assignments: 1
                                         1
              Pseudo assignments: 3
                                         4
       Writing initial simulation snapshot: worklib.test:v
Loading snapshot worklib.test:v ..... Done
ncsim> source
/uusoc/facility/cad_tools/Cadence/LDV/tools/inca/files/ncsimrc
ncsim> run
Saw4 simulation is finished ...
If there were no 'ERROR' statements, then everything worked!
Simulation complete via $finish(1) at time 13200 NS + 0
./testfixture.v:17 $finish;
ncsim> exit
--->
```

Figure 4.40: Output of stand-alone NC_Verilog simulation of seetest.v

🧱 Console - SimVision	
<u>File</u> <u>E</u> dit <u>V</u> iew Sim <u>u</u> lation <u>W</u> indows	<u>H</u> elp
] 🚰 💞 % 🛍 🛍 🗙	
🖿 - 🕅 🔣 📅 🚱 🕜 13,200ns + 0	🐼 🔊 🐏
Saw4 simulation is finished If there were no 'ERROR' statements, then everything worked!	X
Simulation complete via \$finish(1) at time 13200 NS + 0 ./testfixture.v:17	7
SimVision simulator	

Figure 4.41: Control console for *NC_Verilog* through *SimVision*

---> sim-vcs -f test.files Chronologic VCS (TM) Version X-2005.06-SP2 -- Sat Jul 29 21:49:35 2006 Copyright (c) 1991-2005 by Synopsys Inc. ALL RIGHTS RESERVED This program is proprietary and confidential information of Synopsys Inc. and may be used and disclosed only as authorized in a license agreement controlling such use and disclosure. Parsing design file 'see4.v' Parsing design file 'seetest.v' Parsing included file 'testfixture.v'. Back to file 'seetest.v'. Top Level Modules: test No TimeScale specified Starting vcs inline pass ... 1 module and 0 UDP read. recompiling module test make: Warning: File 'filelist' has modification time 41 s in the future if [-x ../simv]; then chmod -x ../simv; fi g++ -o ../simv -melf_i386 -m32 5NrI_d.o 5NrIB_d.o gzYz_1_d.o SIM_1.o /uusoc/facility/cad_tools/Synopsys/vcs/suse9/lib/libvirsim.a /uusoc/facility/cad_tools/Synopsys/vcs/suse9/lib/libvcsnew.so /uusoc/facility/cad_tools/Synopsys/vcs/suse9/lib/ctype-stubs_32.a -ldl -lc -lm -ldl /usr/lib64/gcc/x86_64-suse-linux/4.0.2/../../../x86_64-suse-linux/bin/ld: warning: libstdc++.so.5, needed by /uusoc/facility/cad_tools/Synopsys/vcs/suse9/lib/libvcsnew.so, may conflict with libstdc++.so.6 ../simv up to date make: warning: Clock skew detected. Your build may be incomplete. CPU time: .104 seconds to compile + .384 seconds to link --->

Figure 4.42: Output of running sim-vcs on files.txt

54 CHAPTER 4: Verilog Simulation

```
---> sim-simv simv

Chronologic VCS simulator copyright 1991-2005

Contains Synopsys proprietary information.

Compiler version X-2005.06-SP2; Runtime version X-2005.06-SP2; Jul 29

21:49 2006

Saw4 simulation is finished...

If there were no 'ERROR' statements, then everything worked!

$finish at simulation time 13200

V C S S i m u l a t i o n R e p o r t

Time: 13200

CPU Time: 0.090 seconds; Data structure size: 0.0Mb

Sat Jul 29 21:49:54 2006

--->
```

Figure 4.43: Output of stand-alone VCS simulation of seetest.v using the compiled simv simulator

DVE - TopLevel.1 - [Data.1]	
	w Help
0 x1s 🕶 🔂 🖓 🖓	
= ∓	N : : : : : : : : : : : : : : : : : : :
I 🖻 📴 📲 🕶 🖼 🕶 🗷 I 🕼 🔍 🔍	
	1 module test;
Sim=inter.vpd	3は ▼ 2
Hierarchy 🗸 Type Variable	Valu 4 reg clk, clr, insig;
Lest (test) Mod Copy	5 6 B `include "testfinture.v"
Mar send_test Task - D-it Est Show Sc	7
top (see4) Mod Show So	· sees up(cir, cir, inary, sees);
ternen Barrier Barrie	Schematic 10
🕂 😒 s 🗾 Show M	pry 12
Add To y	
ti⊷ © s Add To L	incont (nave.) Curry
	mave.
Add To V	ches
E <u>x</u> pand <i>i</i>	
Collapse Select Al	
	Annotation pto => /home/elb/txt/book/fsmtest/vcs_b/seetest.v 1 F Reuse
test.top	
Set Bus.	seetest.v
dv=> symopsys : connect -toolargs -uc Chronologic VCS simulator copyright Set Expr	sionstoolin /tmp/vcs_20081031192305_15881_elbstdin -toolout /tmp/vcs
Contains Synopsys proprietary inform Delete A	reakpoints
Compiler version Y-2006.06-SP1; Runt VCD+ Writer Y-2006.06-SP1 Copyright Set Brea	= 31 13+23 2008
/kome/elb/bst/book/femtest/vos_b/ist /kome/elb/bst/book/femtest/vos_b/ist	
set udve_fid VPD0 Dump	
Add <u>F</u> ord	•
•	
dve>	
	Stopped: test 0 × 1s

(Reprinted by permission of Synopsys, Inc. Copyright © 2005, 2010 Synopsys, Inc. All Rights Reserved)

Figure 4.44: Console window for controlling a VCS simulation through DVE

📴 DYE - TopLevel.1 - [Data.1]	
🔄 File Edit View Simulator Signal Scope Trace Window Help	_ 8 ×
13200 x1s 🔻 🛛 🚅 🕮 🗍 🔏 🖺 🕄 🗡	
↓ ● ④ ④ ④ ④ ● ●	모 🗟 물 🕹 🗘 수 수 불 🖉 끓 중 중
🛛 🖻 📴 🍟 🕶 🗷 🖛 🐼 🕶 🖉 🗍 b; 0; 0; 0) 🔘	
×	1 module test;
Sim=inter.vpd	2
Hierarchy 🗸 Type Variable Valu	3 ovire savi; 4 reg clk, clr, insig;
test (test) Mod = w4	5 6 🗄 `imclude "testficture.v"
Task	7
top (see4) Mod ∎ insig €	9
	10 11 endmodule
	12
	/home/elb/txt/book/fsmtest/vcs_b/seetest.v 1 V Reuse
test I I I	
	seetest.v
/home/elb/but/book/famitest/vos b/inter.vpd fid is VPD0	
set ::dwe_fid VPD0 Sawi simulation is finished	
If there were no 'EEROR' statements, then everything worked:	
\$finish at similation time 13200	
VCS Simulation Report Time: 13200	
CPU Time: 0.000 seconds; Data structure size: 0. Fri Oct 31 13:26:22 2008	0995
Similator Terminated	
Log (History	
dve>	
	💭 💿 terminated: test 13200 x 1s 🔛 🖅 🛱 😰 📝

(Reprinted by permission of Synopsys, Inc. Copyright © 2005, 2010 Synopsys, Inc. All Rights Reserved)

Figure 4.45: The result of running the see4 simulation in DVE

💑 D¥E - TopLevel.2 - [\		
🖺 <u>Fi</u> le <u>E</u> dit <u>V</u> iew Si <u>m</u> ul		
13200 ×1s	• 🛛 🚰 🖉 🖉	X 🖻 🛱 🗙 📅 🙀 🔄 🖉 🐣 🗛
	📙 🗳 री) (ज़ी (र्ग (👗 🖡 Any Edge 🔄 🛛 🛱 🌰 🏝 🗇 🖒
	3 2 0 1. 2 -	- 🖸 - 🖬 - ℤ 🛛 ຊ 🔍 🔍 🔘 🥘 🥹 😡 😡
× _ =	- 	
Name Val		
⊡- Group1		
🕂 🛛 state[2:0]	3'h0	
	StO	
⊕- 🛛 next_sta	3'h0	O)()()()(4)(0)()()(4)()(0)()(0)()()
clr	StO	
D- clk	St1->St0	
New Group		
	0	5000
		₩
🕞 Wave.1		
test.top.saw4 St0		💭 💿 terminated: test N/A 🛛 🔝 😰 🔽 🥢

(Reprinted by permission of Synopsys, Inc. Copyright © 2005, 2010 Synopsys, Inc. All Rights Reserved)

Figure 4.46: Waveform window for DVE after running the see4 simulation

Xicfb - Log: /home/el	b/CDS.log		>
File Tools Options			Help 1
(in CELL NAME invXl nor2 nand2 twoBitAdd FullAdder nand2_a invXl_a End Net:	etlist Configuration Info acremental data only) VIEW NAME behavioral behavioral behavioral behavioral schematic schematic schematic list Configuration Info cell - twoBitAdd, view - s 30 16:03:00 2006	NOTE *Stopping View* *Stopping View* *Stopping View* *Stopping View*	
mouse L:	M	R:	
->			

(Copyright © 2005, 2010, Cadence Design Systems, Inc. All rights reserved worldwide. Reprinted with permission.)

Figure 4.47: A netlisting log for the two-bit adder that stops at behavioral views of the standard cell gates

💥 ¥erilog	g Netlisting	Options		
ОК	Cancel	Defaults	Apply	Help
Netlist	ing Mode	\diamondsuit Er	itire Desi	n 🔶 Incremental 😞 Off
Netlist	These Vie	ws beha	vioral [f	unctional schematic cmos_sch
Netlist	For LAI/LI	MSI Models		More >>

(Copyright ©2005, 2010, Cadence Design Systems, Inc. All rights reserved worldwide. Reprinted with permission.)

Figure 4.48: The Setup Netlist dialog from Verilog-XL

59

Netlist	Jecop	1	1				
ок	Cancel	Defaults	Apply				Hel
Netlisti	ing Mode	¢١	Entire Design 🔶	Incrementa	l 🔷 Off	r	
Netlist	These Vie	ws bek	avioral functi	ional scher	matic c	mos_sch	٦
Netlist	For LAI/LI	MSI Mode	Is 🗌				
Genera	te Verilog	Test Fixt	ure Template 📕	ſ			
Netlist	Uppercas	e 🗌	Generate Pin	Мар		Preserve Buses	
Netlist	SwitchRC		Skip Null Port	t		Netlist Uselib	
Drop P	ort Range		Incremental (Config List		Symbol Implicit	
Assign	For Alias		Skip Timing I	nformation		Declare Global Locally	
Netlist	Explicitly		Support Esca	pe Names			
Stop N	etlisting a	t Views	behavioral fur	octional			
Global	Power Ne	ts [VDD <u>(</u>				
Global	Ground Ne	ets [GND !				
Global	Time Scale	e Overwri	te Schematic Tir	neScale			
Global	Sim Time	[1 <u>í</u>	_	Unit	ns 🖃	
	Sim Preci:	nion [100		Unit	ps 💷	

(Copyright ©2005, 2010, Cadence Design Systems, Inc. All rights reserved worldwide. Reprinted with permission.)

Figure 4.49: The Setup Netlist dialog from NC_Verilog

Xicfb - Log: /home/elb	/CDS.log		
File Tools Options			Help 1
CELL NAME	VIEW NAME	NOTE	<u> </u>
pmos nmos twoBitAdd FullAdder nand2_a nand2 nor2 invX1_a invX1_a xor2 End Net1 *** Netlisting of c End netlisting Jul	functional functional schematic schematic cmos_sch cmos_sch cmos_sch cmos_sch ist Configuration Info ell - twoBitAdd, view - s 30 16:03:37 2006	*Stopping View* *Stopping View* schematic successful.	
Ι			
mouse L: ->	M :	R:	

(Copyright ©2005, 2010, Cadence Design Systems, Inc. All rights reserved worldwide. Reprinted with permission.)

Figure 4.50: Netlisting result after removing behavioral from the verilogSimViewList

```
module NAND (out, in1, in2);
    output out;
    input in1, in2;
    assign #10 out = ~(in1 & in2);
endmodule
```

Figure 4.51: Verilog description of a NAND gate with explicit timing

```
module NAND (out, in1, in2);
    output out;
    reg out;
    input in1, in2;
    always @(in1 or in2)
        begin
        #10 out = ~(in1 & in2);
        end
endmodule
```

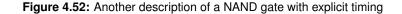


Figure 4.53: NAND description with a delay parameter

```
module nand2 (Y, A, B);
output Y;
input A;
input B;
nand _i0 (Y, A, B);
specify
   (A => Y) = (1.5, 1.0);
   (B => Y) = (1.7, 1.2);
endspecify
```

endmodule

Figure 4.54: NAND gate description with specify block

Rename Instances From Library NCSU_Analog_Parts To Library UofU_Analog_Parts	[tutorial	
To Library UofU_Analog_Parts		ices ——
UofU_Analog_Parts		Parts 🛛
	VofU_Analog_P ▼ Refresh Sess	

(Copyright © 2005, 2010, Cadence Design Systems, Inc. All rights reserved worldwide. Reprinted with permission.)

Figure 4.55: Rename Reference Library dialog box from Library Manager

File Tools Options		N	Help	1
(i CELL NAME pmos nanos nand2 End Net	etlist Configuration Info noremental data only) VIEW NAVE functional functional functional ist Configuration Info cell - nand2, view - scher 3 15:04:12 2006	NOTE *Stopping View* *Stopping View*		
<u>ः </u>			•	
mouse L: showClick ->	Info() M: sc	hHiMousePopUp()	R:simVerilogContinueSimulati	ion (

(Copyright ©2005, 2010, Cadence Design Systems, Inc. All rights reserved worldwide. Reprinted with permission.)

Figure 4.56: Netlist log for the nand2 cell

63

🚟 Waveform 1 - SimVision	
Eile Edit ⊻iew Explore Format Windows	<u>H</u> elp
🚰 🚰 🗠 🖙 👔 🛍 🗶 🌆 🦛 🏝 🎋	🔊 🎬 🖹 🖀 📰 🗐
Search Names: Signal - 💽 🛝 🛝 Search Times: Value - 🦓	
TimeA	s : 11,600ps 🚽 🔍 🕇 🗕 🕬
X O Baseline = 0 Cursor-Baseline = 0 Name ▼ Curso S 10,000ps 10,200ps 10,400ps 10,600ps 10,800ps Int a 0	11,200ps 11,400ps
Image: Book of the second se	
	100,000 125,400ps
log Send selected object(s) to target waveform window (click and hold for a menu of other options)	0 objects selected

(Copyright ©2005, 2010, Cadence Design Systems, Inc. All rights reserved worldwide. Reprinted with permission.)

Figure 4.57: Waveform from simulation of nand2 cell

🗙 Add Property		y 🕟		
ок	Cancel	Apply	Help	
Name	me netType			
Туре	string =			
Value	trirec			
Choice	s			

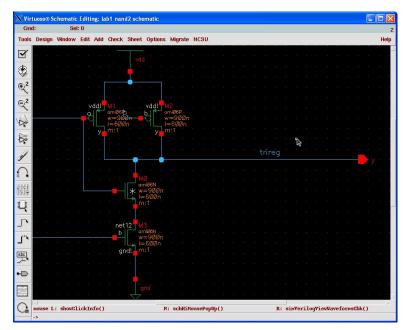
(Copyright © 2005, 2010, Cadence Design Systems, Inc. All rights reserved worldwide. Reprinted with permission.)

Figure 4.58: Adding a property to a net

🗙 Edit Object Properti	es 📐	
OK Cancel Apply	Defaults Previous Next	Help
арру то	urrent = 🛛 wire segment = 🔤 stem 🔳 user	
Property	Value	Display
Het Name	У	off 🖃
Width	♦ narrow ⇒ wide 0.0625	
Color	cadetBlue	
Line Style	solid =	
	Add Delete Modify	
User Property	Value	Display
netType	trired	value

(Copyright © 2005, 2010, Cadence Design Systems, Inc. All rights reserved worldwide. Reprinted with permission.)

Figure 4.59: A net with a netType property



(Copyright © 2005, 2010, Cadence Design Systems, Inc. All rights reserved worldwide. Reprinted with permission.)

Figure 4.60: Nand2 cell with r_nmos and trireg modifications



(Copyright © 2005, 2010, Cadence Design Systems, Inc. All rights reserved worldwide. Reprinted with permission.)

Figure 4.61: Waveform from simulation of nand2 cell with modifications