EE 271
iVerilog Tutorial

University of Washington - Department of Electrical Engineering

James K. Peckol

Working with iVerilog and gtkwave

You can download and install Icarus Verilog on your personal machines. If you do, make certain that the path to the installation does not contain any spaces. The same is true for any file names.

Working with iVerilog and gtkwave as design tools is rather straight forward. The following steps will get you up and running in short order.

The Files

Let’s work with the simple system modeled by the multifile Verilog programs, *andorTop0.v* and *andOr0.v*, that follow. In your working directory, you have the two files shown.

```
//  andorTop0.v

`include "andOr0.v"

module testBench;

  //  connect the two modules
  wire [1:0]        X, Y;
  wire [1:0]       XandY,  XorY;

//  declare an instance of the AND module
  AndOr    myAndOr (XandY[1:0], XorY[1:0], X[1:0], Y[1:0]);

//  declare an instance of the tester module
  Tester    aTester (X[1:0], Y[1:0], XandY[1:0], XorY[1:0]);

// file for gtkwave

  initial
        begin
          $dumpfile("andor0.vcd");
          $dumpvars(1,myAndOr);
        end

endmodule
```

```
//  andorTop0.v cont.

module Tester (xOut, yOut, XandYin, XorYin);

  input  [1:0]      XandYin, XorYin;
  output [1:0]    xOut, yOut;
  reg [1:0]         xOut, yOut;

  parameter stimDelay = 20;

  initial  // Response
  begin
       $display("\t\t xOut  yOut \t XandYin  XorYin \t    Time ");
       $monitor("\t\t %b\t %b  \t %b \t  %b", xOut, yOut, XandYin,
                                                        XorYin, $time );
  end

  initial  // Stimulus
  begin
                    xOut = 'b00; yOut = 'b10;
    #stimDelay xOut = 'b10;
    #stimDelay yOut = 'b01;
    #stimDelay xOut = 'b11;

    #(2*stimDelay);                   // needed to see END of simulation
    $finish;                          // finish simulation
  end

endmodule
```

```
// andOr0.v

// Compute the logical AND and OR of inputs A and B.
module AndOr(AandB, AorB, A, B);
  output [1:0]    AandB, AorB;
  input [1:0]     A, B;

  and       myAnd [1:0] (AandB[1:0], A[1:0], B[1:0]);
  or        myOr [1:0] (AorB[1:0], A[1:0], B[1:0]);
endmodule
```

The Process

1. Create a Verilog source files for the circuit or system that you are modeling and testing using Notepad++ or your favorite editor. Above we have the files *andorTop0.v* and *andOr0.v.*

   As above, to provide the data for the gtwave tool, you must structure your test bench as…

```
`include "myDesign0.v"

module testBench;

        //  connect the two modules
        wire  inputs;
        wire  outputs;

        // declare an instance of the MyDesign module
        MyDesign        myDesign(outputs, inputs);

        // declare an instance of the Tester module
        Tester          myTester (outputs, inputs);

        // file specifications for gtkwave
        initial
        begin
                // dump file is for dumping all the variables in a simulation
                $dumpfile("gfxFile.vcd");

                //  dumps all the variables in module myDesign and below
                //  but not modules instantiated in myDesign into the dump
                file.
                $dumpvars(1,myDesign);
        end

endmodule
```

1. On your PC, go to the *Start* button and select *run*. Enter *cmd* and navigate to where your source file is.
2. Create an executable by typing

   ```
   iverilog andorTop0.v
   ```

   which produces a file named *a.out*

   or specify the name of the output file by typing

   ```
   iverilog -o andorTop0 andorTop0.v
   ```

   which produces a file named *andorTop0*
3. Run the executable by typing either

   ```
   vvp a.out
   ```

   or

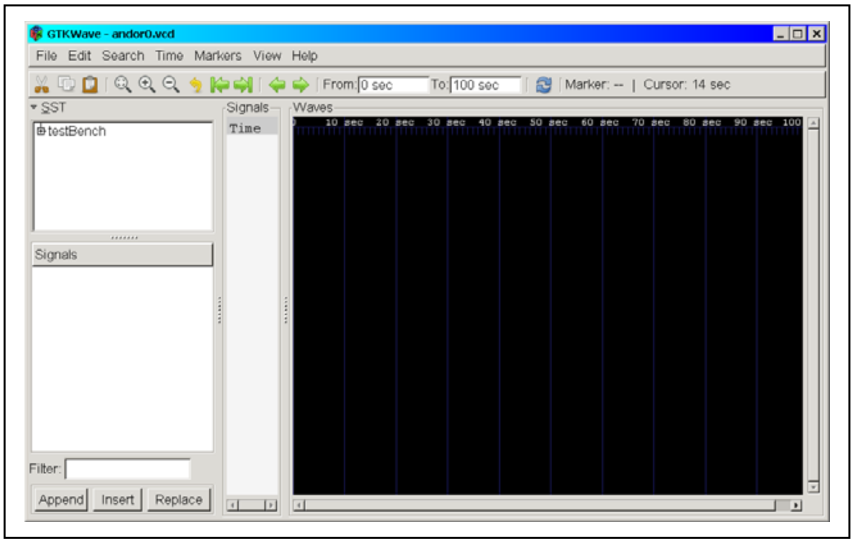
```
vvp andorTop0
```

as appropriate. In addition to executing your program and producing a state output, the vvp tool will also produce a file with a .vcd extension that we use as the input to gtkwave. If we named the dumpfile above *andor0.vcd*, this is the file that the vvp tool will produce.

4. To run the gtkwave tool to get the timing diagram, type

```
gtkwave andor0.vcd
```

to create the initial display window,



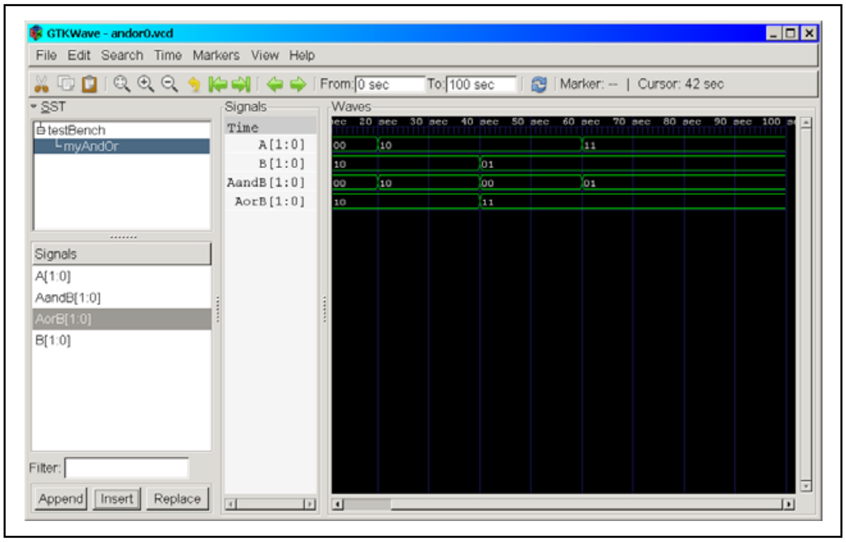
GTKWave - andor0.vcd

From: 0 sec To: 100 sec | Marker: -- | Cursor: 14 sec

File Edit Search Time Markers View Help
SST
testBench
Signals
Time
Waves
10 sec 20 sec 30 sec 40 sec 50 sec 60 sec 70 sec 80 sec 90 sec 100
Signals
Filter:
Append
Insert
Replace

Select and expand *testBench* then *myAndOr* to provide a list of signals that can be displayed. Choose then click on a displayed signal then select *Insert* to add that signal waveform to the timing diagram.

The following diagram displays all of the top-level signals for the module instance *myAndOr*.



GTKWave - andor0.vcd

From: 0 sec To: 100 sec | Marker: -- | Cursor: 42 sec