Tutorials Home

VERILOG BASIC TUTORIAL

Verilog Introduction

Installing Verilog and Hello World

Simple comparator Example

Code Verification

Simulating with verilog

Verilog Language and Syntax

Verilog Syntax Contd..

Verilog Syntax - Vector Data

Verilog $monitor

Verilog Gate Level Modeling

Verilog UDP

Verilog Bitwise Operator

Viewing Waveforms

Full Adder Example

Multiplexer Example

Always Block for Combinational ckt

if statement for Combinational ckt

Case statement for Combinational ckt

Hex to 7 Segment Display

casez and casex

full case and parallel case

Verilog for loop

Verilog localparam and parameter

SEQUENTIAL CKT TUTORIAL

Sequential Circuits

Serial Shift Register

Binary Counters

Ring Counter

MISC VERILOG TOPICS

Setting Path Variable

Verilog Tutorial Videos

Verilog Interview questions #1

Verilog Interview questions #2

Verilog Interview questions #3

Verilog Books

Synchronous and Asynchronous Reset

Verilog TUTORIAL | Verification

If our code is correct it will produce the result according to the following table

Table: A one bit comparator

Input x	Input y	Output z
0	0	1
0	1	0
1	0	0
1	1	1

Every verilog implementation goes through extensive verification. How do we verify that the circuit behaves as expected ? We basically provide stimulus to the circuit at its input port and check its output. We change the input and check the output again. We continue doing it till we exhaust all possible inputs. If output under all conditions is as expected, the circuit stands verified.

Here is how a stimulus verification code looks like

```
// referencedesigner.com verilog tutorial
// testbench for comparator module
`timescale 1ns / 1ps
module stimulus;
        // Inputs
        reg x;
        reg y;
        // Outputs
        wire z;
        // Instantiate the Unit Under Test (UUT)
        comparator uut (
                .x(x),
                .y(y),
                .z(z)
        );

        initial begin
                // Initialize Inputs
        x = 0;
        y = 0;

        #20 x = 1;
        #20 y = 1;
        #20 y = 0;
        #20 x = 1;
        #40;
    end

                initial begin
                 $monitor("x=%d,y=%d,z=%d \n",x,y,z);
                 end

endmodule

```

We will try to make you understand the code. The code

```
`timescale 1ns / 1ps
```

defines the timescale. It tells that the simulation will run in steps of 1ns and has a precision value of 1ps. We will come back to it later.

We need a way for the stimulus to simulate the input and the output pins of the comparator module. We do this by defining two

Left and Right shift << and >>

Negative Numbers

Blocking Vs Non Blocking

wand and wor

delay in verilog

$dumpfile and $dumpvars

Useful Resources

Verilog Examples

VERILOG QUIZS

Verilog Quiz # 1

Verilog Quiz # 2

Verilog Quiz # 3

Verilog Quiz # 4

Verilog Quiz # 5

Verilog Quiz # 6

Verilog Quiz # 7

Verilog Quiz # 8

Verilog Quiz # 9

OTHER TUTORIALS

Verilog Simulation with Xilinx ISE

VHDL Tutorial

register variables x and y and a wire variable z as in

```
// Inputs
reg x;
reg y;
// Outputs
wire z;
```

We will discuss more about registers and wires later. The next statement creates an instance of the comparator module.

```
comparator uut (
.x(x),
.y(y),
.z(z)
);
```

The following section of the code defines the values of the input at different interval of time.

```
initial begin
// Initialize Inputs
x = 0;
y = 0;

#20 x = 1;
#20 y = 1;
#20 y = 1;
#20 x = 0;
#40
end
```

Initially the value of x and y are both 0. At t = 50 ns , we assign the value of x as 1. So at t = 50 ns we have x =1 and y = 0. At t = 60 we assign the value of as 1. So at t = 60 we have x =1 and y =1. So and and so forth. At each step, the value of the out z changes in accordance with the design and this is what we need to verify.

The following code monitor the values of x, y and z at EACH change.

```
initial begin
$monitor("x=%d,y=%d,z=%d \n",x,y,z);
 end
```

It will basically display the values of x, y and z as follows.

```
x=0,y=0,z=1
x=1,y=0,z=0
x=1,y=1,z=1
x=0,y=1,z=0
```

So how you wish to compile and run the comparator and the test bench ? You can do it right here onlline or using the icarus tool (if we have installed it). To see the code in action follow the link here (Opens in a new tab), see the code and the results. You can also make changes in the code and run the program again.

In the next page we will see how to test this code using the simulation with Icarus tool.

< Previous Next >

Copyright © 2009 Reference Designer

Tutorials | Products | Services | Contact Us