The goal of the assignment is to develop an understanding for the different Verilog testbench concepts.

***Homework 6: PARTS 1&2***

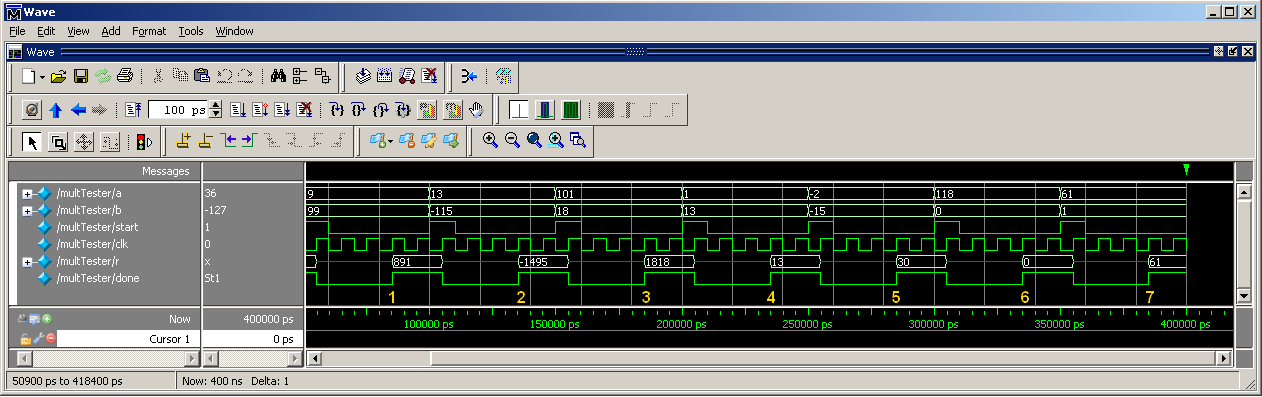
The Verilog code for this section can be found in the Pt1&2 folder of the ZIP file submitted:

*mult.v*

*multTester.v*

In running the multTester module in ModelSim, the following waveform was generated:

**Simulation Waveform Output of multTester #1**



In the *mult.v* module, an RTL delay of 3 full clocks was inserted to mimic the delay in bit multiplication. In the *multTester.v* module, a total of seven (7) multiplication operations were performed, all representing different sorts of scenarios. Here is a details analysis of the results seen above:

*Scenario 1 – 5 = random input*

*Scenario 6 = multiplication of two negative numbers*

*Scenario 7 = multiplication by 0*

*Scenario 8 = multiplication by 1*

**Scenario 1 CORRECT**

* a = 9, b = 99 🡪 r = a \* b = 891 after 3 clk periods

**Scenario 2 CORRECT**

* a = 13, b = -115 🡪 r = a \* b = -1495 after 3 clk periods

**Scenario 3 CORRECT**

* a = 101, b = 18 🡪 r = a \* b = 1818 after 3 clk periods

**Scenario 4 CORRECT**

* a = 1 , b = 13 🡪 r = a \* b = 13 after 3 clk periods

**Scenario 5 CORRECT**

* a = -2, b = -15 🡪 r = a \* b = 30 after 3 clk periods

**Scenario 6 CORRECT**

* a = 118, b = 0 🡪 r = a \* b = 0 after 3 clk periods

**Scenario 7 CORRECT**

* a = 61 , b = 1 🡪 r = a \* b = 61 after 3 clk periods

This concludes the analysis for Homework 6, Parts 1 & 2.

***Homework 6: PARTS 3&4***

The Verilog code for this section can be found in the Pt3&4 folder of the ZIP file submitted:

*multFault.v*

*multFaultTester.v*

In order to create an Interactive Testbench, I performed multiplication for a total of 20 times in the testbench, and I waited for the *done* variable to turn to 1 before performing the next multiplication:

*initial begin*

*repeat (20) begin*

*wait (done == 1'b0);*

*wait (done == 1'b1);*

*a <= $random;*

*b <= $random;*

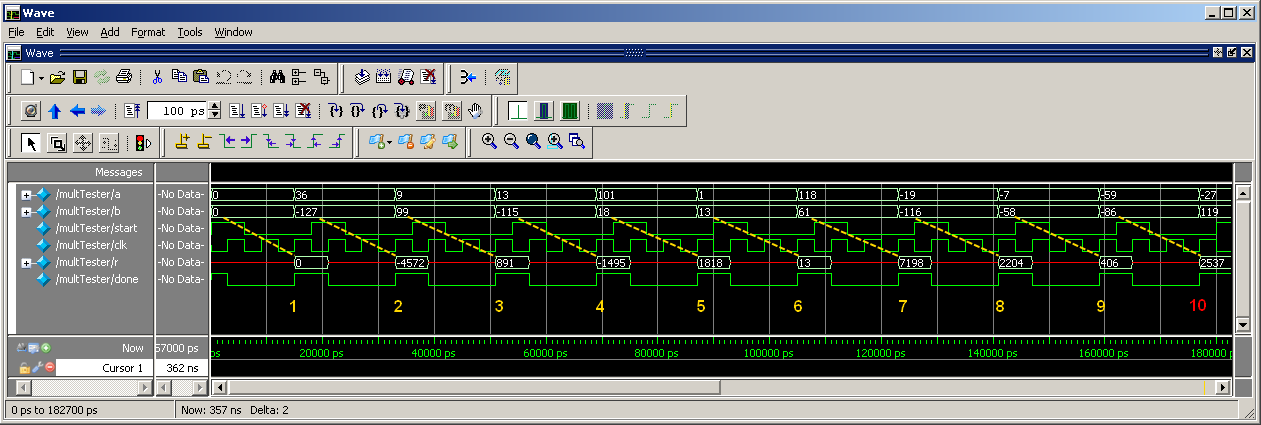
*end*

*#25; $finish;*

*end*

In order to implement a forced fault into this Verilog module, I created an internal counter called *internalError*. After each multiplication, this counter increments by one. As soon as this counter reaches a value indicating multiplication has happened 9 times, it force inserts a fault in the 10th multiplication in a row by dividing the result by 2. See the output below for confirmation:

**Simulation Waveform Output of multFaultTester, Pt 1**



**Scenario 1 CORRECT**

* a = 0 , b = 0 🡪 r = a \* b = 0 after 2 clk periods

**Scenario 2 CORRECT**

* a = 36 , b = -127 🡪 r = a \* b = -4572 after 2 clk periods

**Scenario 3 CORRECT**

* a = 9 , b = 99 🡪 r = a \* b = 891 after 2 clk periods

**Scenario 4 CORRECT**

* a = 13 , b = -115 🡪 r = a \* b = -1495 after 2 clk periods

**Scenario 5 CORRECT**

* a = 101, b = 18 🡪 r = a \* b = 1818 after 2 clk periods

**Scenario 6 CORRECT**

* a = 1, b = 13 🡪 r = a \* b = 13 after 2 clk periods

**Scenario 7 CORRECT**

* a = 118, b = 61 🡪 r = a \* b = 7198 after 2 clk periods

**Scenario 8 CORRECT**

* a = -19, b = -116 🡪 r = a \* b = 2204 after 2 clk periods

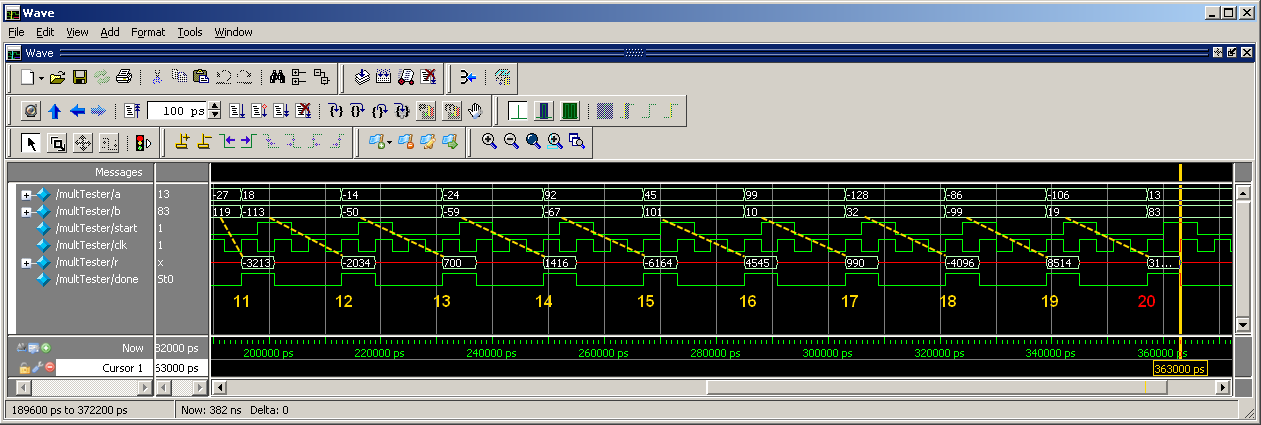
**Scenario 9 CORRECT**

* a = -7, b = -58 🡪 r = a \* b = 406 after 2 clk periods

**Scenario 10 FAULT**

* a = -59, b = -86 🡪 r = a \* b = 2537 🡺 Should be 5074!

**Simulation Waveform Output of multFaultTester, Pt 2**



**Scenario 11 CORRECT**

* a = -27 , b = 119 🡪 r = a \* b = -3213 after 2 clk periods

**Scenario 12 CORRECT**

* a = 18 , b = -113 🡪 r = a \* b = -2034 after 2 clk periods

**Scenario 13 CORRECT**

* a = -14 , b = -50 🡪 r = a \* b = 700 after 2 clk periods

**Scenario 14 CORRECT**

* a = -24 , b = -59 🡪 r = a \* b = 1416 after 2 clk periods

**Scenario 15 CORRECT**

* a = 92, b = -67 🡪 r = a \* b = -6164 after 2 clk periods

**Scenario 16 CORRECT**

* a = 45, b = 101 🡪 r = a \* b = 4545 after 2 clk periods

**Scenario 17 CORRECT**

* a = 99, b = 32 🡪 r = a \* b = -4096 after 2 clk periods

**Scenario 18 CORRECT**

* a = -128, b = -116 🡪 r = a \* b = 2204 after 2 clk periods

**Scenario 19 CORRECT**

* a = -86, b = -99 🡪 r = a \* b = 8514 after 2 clk periods

**Scenario 20 FAULT**

* a = -106, b = -19 🡪 r = a \* b = 31761 🡺 Should be 2014!

The above output shows that for exactly 10% of the time, the circuit produces an incorrect output.

This concludes the analysis for Homework 6, Parts 3 & 4.