Cory Snooks
Aaron Pederson
EE 465 Lab 1
9/9/2015

Verilog Coding and Test Bench for Simple Functions

Introduction

The purpose of this lab was to simulate a circuit using Verilog in Multisim. The desired outputs are shown in Figure 1 and the schematic view is shown in Figure 3. To get the proper delays, three output registers were used. The output was saved into oRESULTA or oRESULTB for one clock cycle before being added or assigned to oRESULT. The timescale was set to 10 ns to simulate a 50 MHz clock.

oRESULT= iA0*iB0+iA1*iB1+ iA0* iA1* iB0*iB1 when iSEL=0

oRESULT= iA0*iB0+iA1*iB1 when iSEL=1

Figure 1: The required outputs are shown above for different values of iSEL.

- iCLK — input clock signal, 50MHz
- iRST_N — input reset signal, active Low
- iA0, iB0, iA1, iB1 — 8-bit binary unsigned integer input signals
- iSEL — input selection signal
- oRESULT — 17-bit binary unsigned integer output signal

Figure 2: The variable parameters used in this lab are shown above.

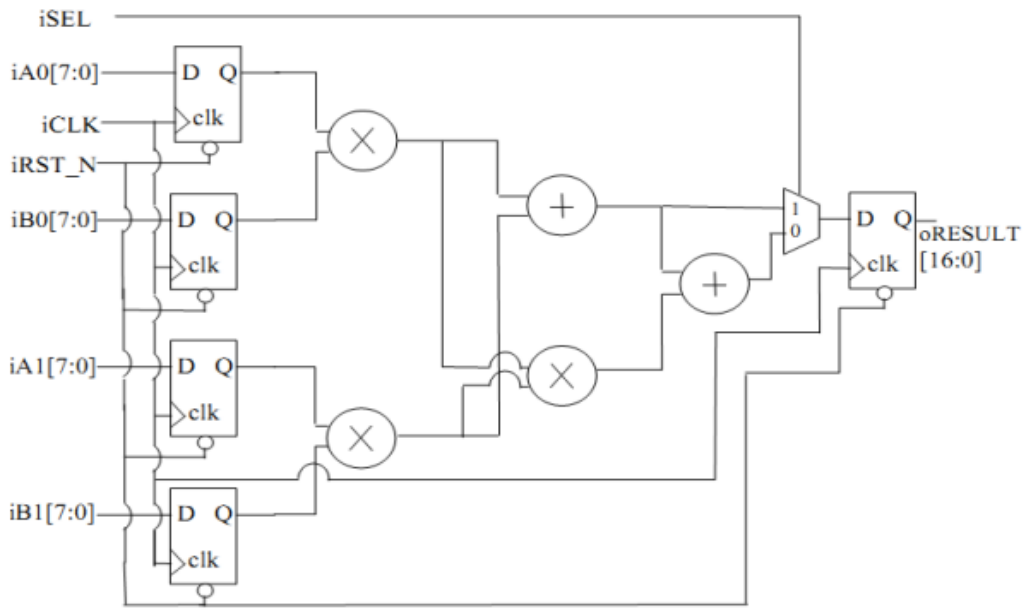

Figure 3: The schematic view of the required output is shown above.

Module Code

```
module test(iCLK, iRST_N, iSEL,  iA0, iA1, iB0, iB1, oRESULT);
input iCLK, iRST_N, iSEL, iA0, iA1, iB0, iB1;
output oRESULT;

wire iCLK, iRST_N;
wire [7:0] iA0, iA1, iB0, iB1;
reg [16:0] oRESULTA, oRESULTB, oRESULT;

always @ (posedge iCLK)begin
 if(iRST_N)begin
  oRESULTA <= iA0 * iB0 + iA1 * iB1;
  oRESULTB <= iA0 * iB0 + iA1 * iB1 + iA0 * iA1 * iB0 * iB1;
  if(iSEL)begin
     oRESULT <= oRESULTA ;
  end
  else begin
     oRESULT <= oRESULTB;
  end
 end
 else begin
  oRESULT <= 0;
 end
end

endmodule
```

Testbench

```
`timescale 10ns/1ns
module test_tb ();

 reg iCLK_t, iRST_N_t, iSEL_t;
 reg [7:0] iA0_t, iA1_t, iB0_t, iB1_t;
 wire [16:0] oRESULT_t;

 test X(iCLK_t, iRST_N_t, iSEL_t,  iA0_t, iA1_t, iB0_t, iB1_t, oRESULT_t);

 initial $display ("Test control");
          initial $display ("%10s %10s %10s %10s %10s %10s %10s %10s",
    "iCLK_t", "iRST_N_t", "iSEL_t", "iA0_t", "iA1_t",
    "iB0_t", "iB1_t", "oRESULT_t");
  initial $monitor ("%10b %10b %10b %10d %10d %10d %10d %10d",
    iCLK_t, iRST_N_t, iSEL_t,  iA0_t, iA1_t, iB0_t, iB1_t, oRESULT_t);
```

```
initial begin

   iCLK_t = 0;
   iRST_N_t = 0;
   iSEL_t = 0;
   iA0_t = 0;
   iA1_t = 0;
   iB0_t = 0;
   iB1_t = 0;

   #1
   iRST_N_t = 1;

  #3
   iSEL_t = 1;
   iA0_t = 2;
   iA1_t = 3;
   iB0_t = 5;
   iB1_t = 1;

   #10
   iSEL_t = 0;
   iA0_t = 2;
   iA1_t = 3;
   iB0_t = 5;
   iB1_t = 1;

   #10
   iSEL_t = 1;
   iA0_t = 2;
   iA1_t = 3;
   iB0_t = 5;
   iB1_t = 1;

   #10
   iSEL_t = 0;
   iA0_t = 53;
   iA1_t = 26;
   iB0_t = 120;
   iB1_t = 5;

   #10
   iSEL_t = 0;
   iA0_t = 26;
   iA1_t = 1;
   iB0_t = 15;
   iB1_t = 19;
```

```
  #4
  $stop;
end

 always #1 iCLK_t = ~iCLK_t;
 endmodule
```

Figure 4 shows that the output was delayed two clock cycles after a change was recorded for the inputs. The first result is iA0 * iB0 + iA1 * iB1 = 2*5 +3*1 = 13 because iSEL is high. The second result is after iSEL goes low and is iA0 * iB0 + iA1 * iB1 + iA0 * iA1 * iB0 * iB1 = 13 + 2*3*5*1 = 43.

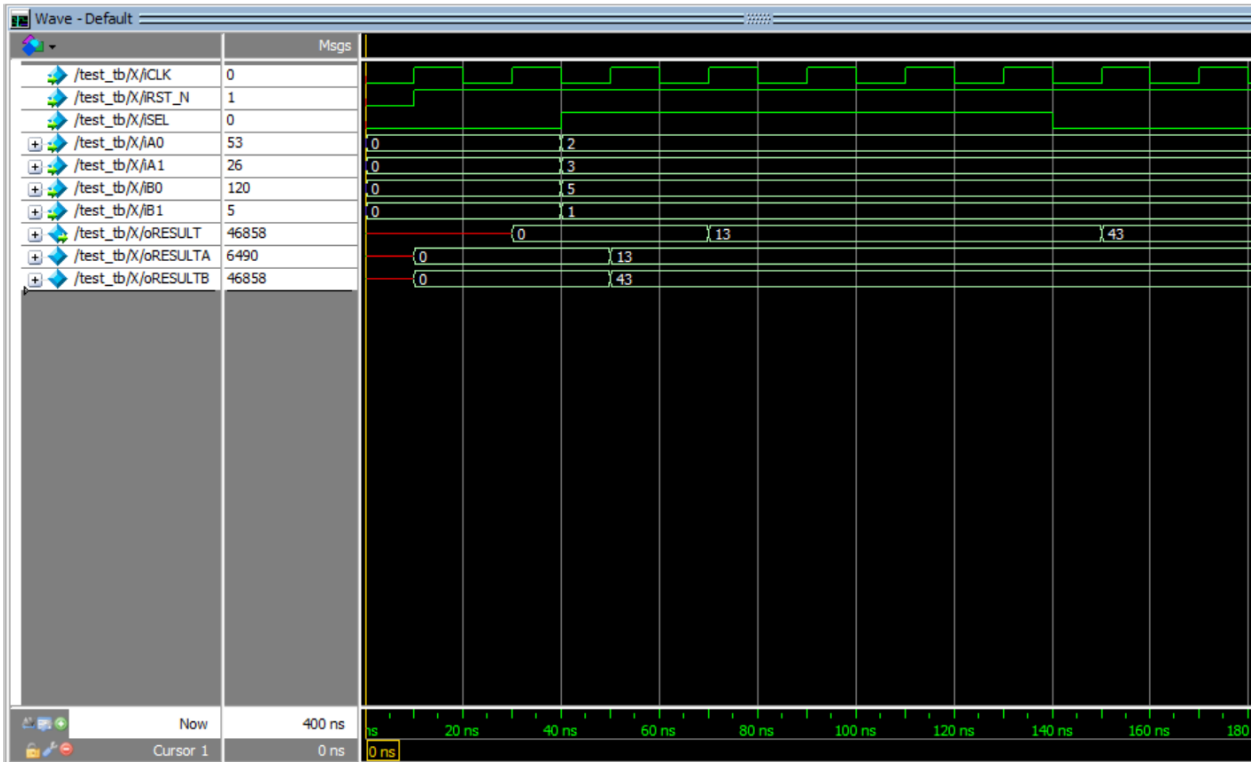

Figure 4: The wave view from the test bench output is shown above.

Output

```
# Test control
#     iCLK_t   iRST_N_t     iSEL_t      iA0_t      iA1_t      iB0_t      iB1_t  oRESULT_t
#          0          0          0          0          0          0          0          x
#          1          1          0          0          0          0          0          x
#          0          1          0          0          0          0          0          x
#          1          1          0          0          0          0          0          0
#          0          1          1          2          3          5          1          0
#          1          1          1          2          3          5          1          0
#          0          1          1          2          3          5          1          0
#          1          1          1          2          3          5          1         13
#          0          1          1          2          3          5          1         13
#          1          1          1          2          3          5          1         13
#          0          1          1          2          3          5          1         13
#          1          1          1          2          3          5          1         13
#          0          1          1          2          3          5          1         13
#          1          1          1          2          3          5          1         13
#          0          1          0          2          3          5          1         13
#          1          1          0          2          3          5          1         43
#          0          1          0          2          3          5          1         43
#          1          1          0          2          3          5          1         43
#          0          1          0          2          3          5          1         43
#          1          1          0          2          3          5          1         43
#          0          1          0          2          3          5          1         43
#          1          1          0          2          3          5          1         43
#          0          1          0          2          3          5          1         43
#          1          1          0          2          3          5          1         43
#          0          1          1          2          3          5          1         43
#          1          1          1          2          3          5          1         13
#          0          1          1          2          3          5          1         13
#          1          1          1          2          3          5          1         13
#          0          1          1          2          3          5          1         13
#          1          1          1          2          3          5          1         13
#          0          1          1          2          3          5          1         13
#          1          1          1          2          3          5          1         13
#          0          1          1          2          3          5          1         13
#          1          1          1          2          3          5          1         13
#          0          1          0         53         26        120          5         13
#          1          1          0         53         26        120          5         43
#          0          1          0         53         26        120          5         43
#          1          1          0         53         26        120          5      46858
#          0          1          0         53         26        120          5      46858
#          1          1          0         53         26        120          5      46858
```

Figure 5: The output values are shown above.

Conclusion

This lab demonstrated how easy it is to go from concept to testing using Verilog. The only tricky part was getting the output to change with the correct delays to match the actual circuit output.