Lab 10

Top module

```
module top(
    input clk,
    input [31:0] x,
    input [31:0] t,
    input [31:0] n,
    input in_ready,
    input in_valid,
    output [31:0] result,
    output out_ready,
    output out_valid
    );

    wire [31:0] ans;
    reg [31:0] result=0;
    wire [31:0] c;
    wire [31:0] data1;
    wire [31:0] data2;
    wire [31:0] data3;
    wire [31:0] data4;
    wire [31:0] data5;
    wire [31:0] data6;

   assign c=32'b01000000_00010011_01100100_01011010;
```

```
floating_point_1 ln(
    .aclk(clk),                                 // input wire aclk
    .s_axis_a_tvalid(in_valid),             // input wire s_axis_a_tvalid
    .s_axis_a_tready(out_ready),              // output wire
s_axis_a_tready
    .s_axis_a_tdata(n),               // input wire [31 : 0] s_axis_a_tdata
    .m_axis_result_tvalid(out_valid),  // output wire
m_axis_result_tvalid
    .m_axis_result_tready(in_ready),  // input wire
m_axis_result_tready
    .m_axis_result_tdata(data1)    // output wire [31 : 0]
m_axis_result_tdata
    );

//   floating_point_multiply mul1(
//  .aclk(clk),                                 // input wire aclk
//  .s_axis_a_tvalid(in_valid),             // input wire s_axis_a_tvalid
//  .s_axis_a_tready(in_ready),              // output wire
s_axis_a_tready
//  .s_axis_a_tdata(data1),               // input wire [31 : 0]
s_axis_a_tdata
//  .s_axis_b_tvalid(in_valid),             // input wire s_axis_b_tvalid
//  .s_axis_b_tready(in_ready),              // output wire
s_axis_b_tready
//  .s_axis_b_tdata(c),               // input wire [31 : 0] s_axis_b_tdata
//  .m_axis_result_tvalid(out_valid),  // output wire
m_axis_result_tvalid
//  .m_axis_result_tready(out_ready),  // input wire
m_axis_result_tready
```

```
//  .m_axis_result_tdata(data2)    // output wire [31 : 0]
m_axis_result_tdata
// );


 floating_point_divide div1(
   .aclk(clk),                                  // input wire aclk
   .s_axis_a_tvalid(in_valid),            // input wire s_axis_a_tvalid
   .s_axis_a_tready(in_ready),              // output wire
s_axis_a_tready
   .s_axis_a_tdata(data1),                // input wire [31 : 0]
s_axis_a_tdata
   .s_axis_b_tvalid(in_valid),            // input wire s_axis_b_tvalid
   .s_axis_b_tready(in_ready),              // output wire
s_axis_b_tready
   .s_axis_b_tdata(t),                // input wire [31 : 0] s_axis_b_tdata
   .m_axis_result_tvalid(out_valid),  // output wire
m_axis_result_tvalid
   .m_axis_result_tready(out_ready),  // input wire
m_axis_result_tready
   .m_axis_result_tdata(data3)    // output wire [31 : 0]
m_axis_result_tdata
 );



 floating_point_multiply mul2(
   .aclk(clk),                                  // input wire aclk
   .s_axis_a_tvalid(in_valid),            // input wire s_axis_a_tvalid
```

```
   .s_axis_a_tready(in_ready),           // output wire
s_axis_a_tready
   .s_axis_a_tdata(data3),             // input wire [31 : 0]
s_axis_a_tdata
   .s_axis_b_tvalid(in_valid),           // input wire s_axis_b_tvalid
   .s_axis_b_tready(in_ready),           // output wire
s_axis_b_tready

.s_axis_b_tdata(32'b01000000_00000000_00000000_00000000)
,              // input wire [31 : 0] s_axis_b_tdata
   .m_axis_result_tvalid(out_valid),  // output wire
m_axis_result_tvalid
   .m_axis_result_tready(out_ready),  // input wire
m_axis_result_tready
   .m_axis_result_tdata(data4)    // output wire [31 : 0]
m_axis_result_tdata
  );



    floating_point_0 sqrt(
      .aclk(clk),                          // input wire aclk
      .s_axis_a_tvalid(in_valid),           // input wire s_axis_a_tvalid
      .s_axis_a_tready(out_ready),           // output wire
s_axis_a_tready
      .s_axis_a_tdata(data4),             // input wire [31 : 0]
s_axis_a_tdata
      .m_axis_result_tvalid(out_valid),  // output wire
m_axis_result_tvalid
```

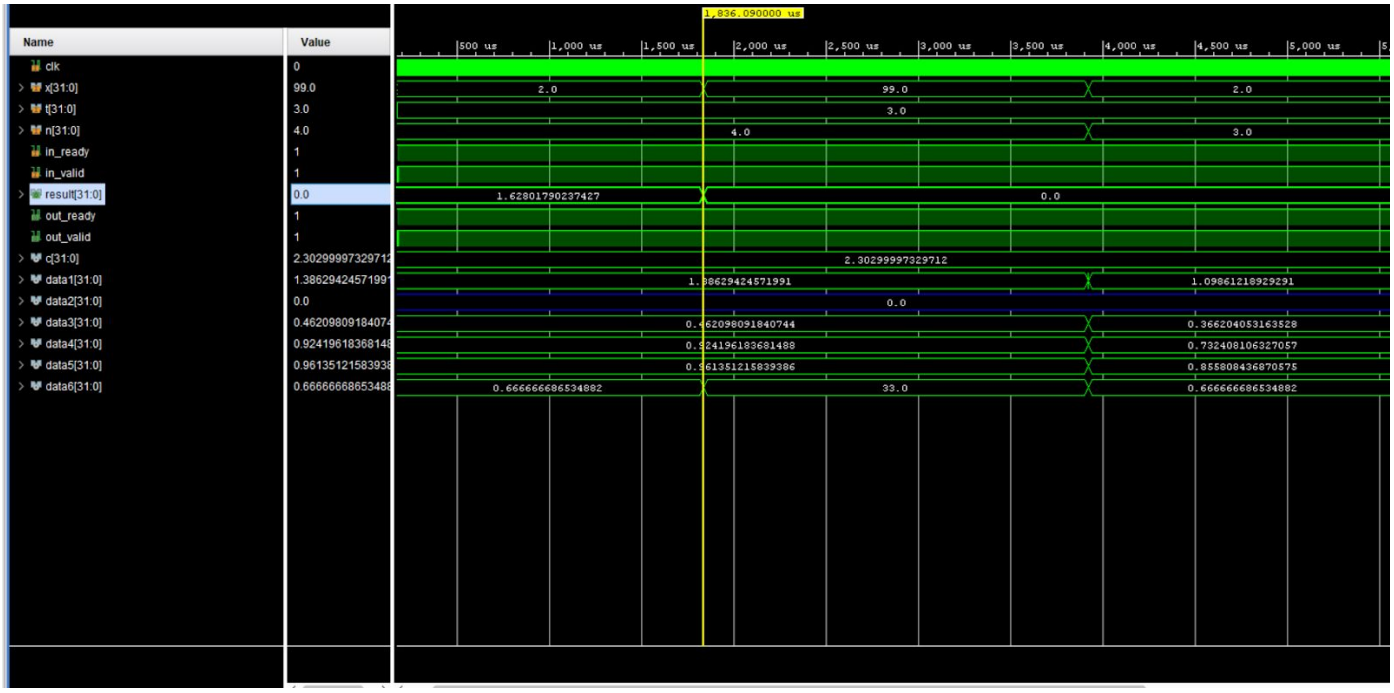
```
      .m_axis_result_tready(in_ready),  // input wire
m_axis_result_tready
      .m_axis_result_tdata(data5)    // output wire [31 : 0]
m_axis_result_tdata
    );


    floating_point_divide div2(
      .aclk(clk),                                // input wire aclk
      .s_axis_a_tvalid(in_valid),            // input wire s_axis_a_tvalid
      .s_axis_a_tready(in_ready),              // output wire
s_axis_a_tready
      .s_axis_a_tdata(x),              // input wire [31 : 0]
s_axis_a_tdata
      .s_axis_b_tvalid(in_valid),            // input wire s_axis_b_tvalid
      .s_axis_b_tready(in_ready),              // output wire
s_axis_b_tready
      .s_axis_b_tdata(t),              // input wire [31 : 0]
s_axis_b_tdata
      .m_axis_result_tvalid(out_valid),  // output wire
m_axis_result_tvalid
      .m_axis_result_tready(out_ready),  // input wire
m_axis_result_tready
      .m_axis_result_tdata(data6)    // output wire [31 : 0]
m_axis_result_tdata
     );
```

```
    floating_point_add addition(
      .aclk(clk),                                // input wire aclk
      .s_axis_a_tvalid(in_valid),            // input wire s_axis_a_tvalid
      .s_axis_a_tready(in_ready),            // output wire
s_axis_a_tready
      .s_axis_a_tdata(data6),              // input wire [31 : 0]
s_axis_a_tdata
      .s_axis_b_tvalid(in_valid),            // input wire s_axis_b_tvalid
      .s_axis_b_tready(in_ready),            // output wire
s_axis_b_tready
      .s_axis_b_tdata(data5),              // input wire [31 : 0]
s_axis_b_tdata
      .m_axis_result_tvalid(out_valid),  // output wire
m_axis_result_tvalid
      .m_axis_result_tready(out_ready),  // input wire
m_axis_result_tready
      .m_axis_result_tdata(ans)    // output wire [31 : 0]
m_axis_result_tdata
    );

    always @ (*)
    begin
      if(x>t || n<=t)
         result<=32'b0;
      else
         result<=ans;
    end

Endmodule
```

Testbench

```
module tb(
    );
    reg clk=0;

    always #5 clk=~clk;

    reg [31:0] x=0;
    reg [31:0] t=0;
    reg [31:0] n=0;
    reg in_ready=0;
    reg in_valid=0;
    wire [31:0] result;
    wire out_ready;
    wire out_valid;

    top fun1(clk,x,t,n,in_ready,in_valid,result,out_ready,out_valid);

    initial
    begin
        in_valid=1;
        in_ready=1;
        #10 x=32'b01000000_00000000_00000000_00000000;
        #10 t=32'b01000000_01000000_00000000_00000000;
        #10 n=32'b01000000_10000000_00000000_00000000;
    end

Endmodule
```

Output


1,836.090000 us
Name
Value
clk
0
x[31:0]
99.0
t[31:0]
3.0
n[31:0]
4.0
in_ready
1
in_valid
1
result[31:0]
0.0
out_ready
1
out_valid
1
c[31:0]
2.30299997329712
data1[31:0]
1.38629424571991
data2[31:0]
0.0
data3[31:0]
0.46209809184074
data4[31:0]
0.92419618368148
data5[31:0]
0.96135121583938
data6[31:0]
0.66666668653488
500 us
1,000 us
1,500 us
2,000 us
2,500 us
3,000 us
3,500 us
4,000 us
4,500 us
5,000 us
2.0
99.0
2.0
3.0
4.0
3.0
1.62801790237427
0.0
2.30299997329712
1.38629424571991
1.09861218929291
0.0
0.46209809184074
0.366204053163528
0.92419618368148
0.732408106327057
0.96135121583938
0.855808436870575
0.66666668653488
33.0
0.66666668653488