LAB 03

Object: To design, implement & simulate Flouting point multiplication and division through Data Flow Modeling.

LAB TASKS:

1. Floating-Point Multiplication:

Verilog Test Fixture:

```
26  module test;
27      // Inputs
28      reg [31:0] a;
29      reg [31:0] b;
30      reg [7:0] bias;
31      // Outputs
32      wire [31:0] out;
33      // Instantiate the Unit Under Test (UUT)
34      Floatingpoint uut (
35          .a(a),
36          .b(b),
37          .bias(bias),
38          .out(out)
39      );
40      initial begin
41          // Initialize Inputs
42          a = 0;
43          b = 0;
44          bias = 0;
45          // Wait 100 ns for global reset to finish
46          #100;
47          a= 32'b01000001110011000000000000000000;
48          b= 32'b11000010011111101000000000000000;
49          bias = 8'd127;
50          #100;
51          a= 32'b10000011110000000000000000000000;
52          b= 32'b01110000010000000000000000000000;
53          bias = 8'd127;
54          // Add stimulus here
55
56      end
57  endmodule
```

Code of mantissa:

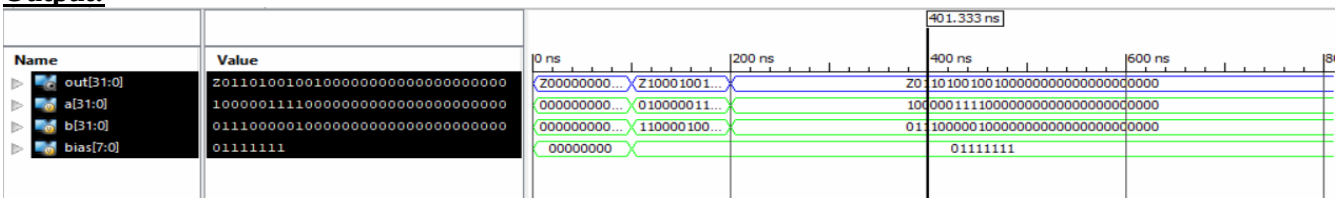
```
23  module mantisa(m0,e0,eout,mout);
24      input [47:0] m0;
25      input [7:0] e0;
26      output [7:0] eout;
27      output [22:0] mout;
28
29  reg [7:0]eout;
30  reg [22:0]mout;
31  always @ *
32  begin
33     if (m0[47]==1)
34        begin
35           eout <=e0+1;
36           mout <=m0[46:24];
37        end
38     else
39        begin
40           eout <=e0;
41           mout <=m0[45:23];
42        end
43     end
44  endmodule
```

Code of Floating point:

```
module Floatingpoint(
    input [31:0] a,
    input [31:0] b,
    input [7:0] bias,
    output [31:0] out
    );

wire [47:0]m0;
wire [22:0]mout;
wire [22:0]ma;
wire [22:0]mb;
wire [7:0]e1;
wire [7:0]e2;
wire [7:0]e0;
wire [7:0]eout;
wire sa,sb,so;
assign sa = a[31];
assign sa = b[31];
assign s0 = sa^sb;
assign e1[7:0] = a[30:23];
assign e2[7:0] = b[30:23];
assign e0[7:0] = e1+e2-bias;
assign ma[22:0] = a[22:0];
assign mb[22:0] = b[22:0];
assign m0[47:0] = {1'b1,ma}*{1'b1,mb};
mantisa inst(m0,e0,eout,mout);
assign out[31:0] ={so,eout,mout};
endmodule
```

Output:



2. Floating-Point Division:

Verilog Test Fixture:

```
module floatingpointTF;
    // Inputs
    reg [31:0] a;
    reg [31:0] b;
    reg [7:0] bias;
    // Outputs
    wire [31:0] out;
    // Instantiate the Unit Under Test (UUT)
    Floatingpoint uut (
        .a(a),
        .b(b),
        .bias(bias),
        .out(out)
    );
    initial begin
        // Initialize Inputs
        a = 0;
        b = 0;
        bias = 0;
        // Wait 100 ns for global reset to finish
        #100;
#100;
a= 32'b01000001110011000000000000000000;
b= 32'b11000010011111101000000000000000;
bias = 8'd127;
#100
a= 32'b10000011110000000000000000000000;
b= 32'b01110000010000000000000000000000;
bias = 8'd127;

    end
endmodule
```

Code of mantissa:

```
23  module mantisa(m0,e0,eout,mout);
24      input [47:0] m0;
25      input [7:0] e0;
26      output [7:0] eout;
27      output [22:0] mout;
28
29  reg [7:0]eout;
30  reg [22:0]mout;
31  always @ *
32  begin
33     if (m0[47]==1)
34        begin
35           eout <=e0+1;
36           mout <=m0[46:24];
37        end
38     else
39        begin
40           eout <=e0;
41           mout <=m0[45:23];
42        end
43     end
44  endmodule
```

Code of Floating point:

```
23  module Floatingpoint(
24      input [31:0] a,
25      input [31:0] b,
26      input [7:0] bias,
27      output [31:0] out
28      );
29
30  wire [47:0]m0;
31  wire [22:0]mout;
32  wire [22:0]ma;
33  wire [22:0]mb;
34  wire [7:0]e1;
35  wire [7:0]e2;
36  wire [7:0]e0;
37  wire [7:0]eout;
38  wire sa,sb,s0;
39  assign sa = a[31];
40  assign sb = b[31];
41  assign s0 = sa^sb;
42  assign e1[7:0] = a[30:23];
43  assign e2[7:0] = a[30:23];
44  assign e0[7:0] = e1-e2+bias;
45  assign ma[22:0] = a[22:0];
46  assign mb[22:0] = b[22:0];
47  assign m0[47:0] = {1'b1,ma}/{1'b1,mb};
48  mantisa inst(m0,e0,eout,mout);
49  assign out[31:0] = {s0,eout,mout};
50
51  endmodule
```

Output: