Some sample submission for “Booth Multiplier”

Verilog file and Testbench

Sample format for submission:

Student Name= _____ Roll No.= _____

Verilog file:

Test bench

Student Name= Kartikay Goel Roll No.= 180101033

Verilog file:

```
module BoothMultiplier(X, Y, Z);
    integer m;
    input signed [3:0] X;
    input signed [3:0] Y;
    output signed [7:0] Z;
    reg signed [7:0] Z;
    reg [1:0] temp;

    reg res1;
    reg [3:0] res2;

    always @ (X, Y)
    begin
    Z = 8'd0;
    res1 = 1'd0;
    for (m = 0; m < 4; m = m + 1)
    begin
    temp = {X[i], res1};
    res2 = - Y;
 case (temp)
    2'd2 : Z [7 : 4] = Z [7 : 4] + res2;
```

```
      2'd1 : Z [7 : 4] = Z [7 : 4] + Y;
      default : begin end
      endcase
      Z = Z >> 1;
      Z[7] = Z[6];
      res1 = X[i];
         end
      if (Y == 4'd8)
         begin
            Z = - Z;
         end
      end
endmodule
```

Test bench

```
module BoothMultiplierTestBench;


   // Inputs
   reg [3:0] X;
   reg [3:0] Y;

   // Outputs
   wire [7:0] Z;

   // Instantiate the Unit Under Test (UUT)
   BoothMultiplier uut (
      .X(X),
      .Y(Y),
```

```
        .Z(Z)
    );

    initial begin
        // Initialize Inputs
        X = 0;
        Y = 0;

        // Wait 100 ns for global reset to finish
        #100;
        X=-5;
        Y=7;
        // Add stimulus here
     end
endmodule
```

Student Name= Kartikeya Saxena Roll No.= 180101034

Verilog file:

```
`timescale 1ns / 1ns
module booth_multiplier(m, r, out);
input [3:0] m;
input [3:0] r;
output reg [7:0] out;


reg signed [8:0] a, s, p;
integer i;


always@(m or r)
```

```
begin
a = {m, 5'b00000};
s = {4'b0 - m, 5'b00000};
p = {4'b0, r, 1'b0};
for(i = 0; i <= 3; i = i + 1)
begin
if(p[0] == 0 & p[1] == 1)
p = p + s;
if(p[0] == 1 & p[1] == 0)
p = p + a;
p = p >>> 1;
end
out = p[8:1];
end
endmodule
```

Test bench

```
`timescale 1ns / 1ns
module booth_multiplier_test_bench();
dreg signed [3:0] m;
reg signed [3:0] r;
wire signed [7:0] out;

booth_multiplier x(.m(m), .r(r), .out(out));

initial
begin
$dumpfile("output.vcd");
```

```
$dumpvars(0, x);
end

initial
Begin(
$monitor("time = %3d, m = %d, r = %d, p = %d\n", $time, m, r, out);
end

initial
begin
#10 m = 3;
r = 4;
#10 m = 2;
r = -1;
#10 m = 3;
r = -5;
#10 m = -5;
r = -7;
#10 $finish;
end
endmodule
```

Sample format for submission:

Student Name= Kushal Sangwan Roll No.= 180101096

Verilog file:

```
module boothalgo(
   input signed [3:0] m,
   input signed[3:0] q,
```

```
    output reg signed [7:0] ans
    );

reg q0;
reg[1:0] decide;
reg signed [3:0] accu,qnew;
reg signed [8:0] container;
integer i;
reg[8:0] powerof2;
always@(m or q)
begin
    q0=0;
    accu=0;
    powerof2=1;
    qnew=q;
    for( i=0;i<4;i=i+1)
    begin
        decide={qnew[0],q0};
        if(decide==2)
            accu=accu-m;
        else if(decide==1)
            accu=accu+m;
        else
            q0=q0;

        container={accu,qnew,q0};
      container=container>>>1;
        q0=container[0];
        qnew=container[4:1];
```

```
        accu=container[8:5];
    end
    ans={accu,qnew};
    if(ans[7]!=0)
    begin
        for( i=0;i<9;i=i+1)
        begin
            powerof2=powerof2*2;
        end
            ans=powerof2-ans;
    end
end




endmodule
```

Test bench

```
module boothtest;

    // Inputs
    reg [3:0] m;
    reg [3:0] q;

    // Outputs
    wire [7:0] ans;
```

```
// Instantiate the Unit Under Test (UUT)
boothalgo uut (
    .m(m),
    .q(q),
    .ans(ans)
);

initial begin
    // Initialize Inputs
    m = -2;
    q = 7;

    // Wait 100 ns for global reset to finish
    #100;

    m=3;
    q=8;

    #100;

    m=-5;
    q=-4;

    #100;

    // Add stimulus here

end
```

```
endmodule
```