LAB 05

OBJECTIVE: Develop, implement, and simulate a 2x2 Multiplier & 4x4 Multiplier.

4x4 Multiplier:

```
22  module Multiplier4x4(
23      input [3:0] a,
24      input [3:0] b,
25      output [7:0] product
26      );
27  wire s0[3:0],s1[3:0],s2[3:0];
28  wire c0,c1,c2;
29        reg p[3:0][3:0];  // double array for partial products
30        integer i,j;
31
32
33        always@(a or b)
34        begin
35              for(i=0;i<=3;i=i+1)        // nested for loops.
36                 for(j=0;j<=3;j=j+1)
37                    p[i][j]<= a[j] & b[i]; // to generate partial products.
38
39        end
40
41
42        RCA4x4 RCA0({1'b0,p[0][3],p[0][2],p[0][1]},{p[1][3],p[1][2],p[1][1],p[0][0]},1'b0,{s0[3],s0[2],s0[1],s0[0]},c0);
43        RCA4x4 RCA1({1'b0,s0[3],s0[2],s0[1]},{p[2][3],p[2][2],p[2][1],p[2][0]},c0,{s1[3],s1[2],s1[1],s1[0]},c1);
44        RCA4x4 RCA2({1'b0,s1[3],s1[2],s1[1]},{p[3][3],p[3][2],p[3][1],p[3][0]},c1,{s2[3],s2[2],s2[1],s2[0]},c2);
45
46
47
48        assign product[0]=p[0][0];
49        assign product[1]=s0[0];
50        assign product[2]=s1[0];
51        assign product[3]=s2[0];
52        assign product[4]=s2[1];
53        assign product[5]=s2[2];
54        assign product[6]=s2[3];
55        assign product[7]=c2;
56  endmodule
```

Verilog Test Fixture Code:

```
25  module MulTF;
26
27      // Inputs
28      reg [3:0] a;
29      reg [3:0] b;
30      // Outputs
31      wire [7:0] product;
32      // Instantiate the Unit Under Test (UUT)
33      Multiplier4x4 uut (
34          .a(a),
35          .b(b),
36          .product(product)
37      );
38      initial begin
39          // Initialize Inputs
40          a = 0;
41          b = 0;
42          // Wait 100 ns for global reset to finis
43          #100;
44
45          // Add stimulus here
46          a=4'b0100;
47          b=4'b1000;
48      end
49  endmodule
```

Wave Output: