Lab 2

1. Partial sum multilpication of two number.

Design code:

```
module PartialProductMultiplication(
    input [3:0] A,
    input [3:0] B,
    output reg [7:0] Product
);
    reg [7:0] partial_products[3:0];
    integer i;

  always @(A, B) begin
        // Initialize the product and partial products to 0
        Product = 8'b0;
        for (i = 0; i < 4; i = i + 1)
            partial_products[i] = 8'b0;

        // Generate partial products
        for (i = 0; i < 4; i = i + 1) begin
            if (B[i])
                partial_products[i] = A << i;
        end

        // Sum the partial products
    for (i = 0; i < 4; i = i + 1)
            Product = Product + partial_products[i];

    end
endmodule
```

Testcode:

```
module testbench;
    reg [3:0] A;
    reg [3:0] B;
    wire [7:0] Product;
    // Instantiate the multiplication module
    PartialProductMultiplication dut (A, B, Product);

    initial begin

       $dumpfile("dump.vcd");
      $dumpvars(1);

        // Test case 1
        A = 4'b1100; // 3 in decimal
          B = 4'b1101; // 2 in decimal
          #10;


       // Test case 2
        A = 4'b1010; // 10 in decimal
        B = 4'b0101; // 5 in decimal
        #10;

        // Test case 3
        A = 4'b0111; // 7 in decimal
        B = 4'b0011; // 3 in decimal
        #10;
    end

      always @* begin
        $display("A: %b, B: %b, Product: %b", A, B, Product);
      end

endmodule
```

2. Unsigned number multiplication of two number.

Design Code:

```
module Unsigned_Multiplication(
    input [3:0] M,
    input [3:0] Q,
    output reg [7:0] P
);
```

```
  integer count,C=0;
  reg [7:0] A_Q; // Concatenation of accumulator and multiplier

  always @(*) begin
    A_Q = {4'b0000, Q}; // Initialize with Q in LSB and
Accumulator 0

    for (count = 0; count < 4; count = count + 1) begin
      if (A_Q[0] == 1'b1) begin
        {C,A_Q[7:4]} = A_Q[7:4] + M; // Add M to accumulator
      end
     {C,A_Q} = {C,A_Q} >> 1; // Logical right shift
    end

    P = A_Q; // Assign result to output
  end
endmodule
```

Test Code:

```
module testbench;

  // Inputs
  reg [3:0] M;
  reg [3:0] Q;

  // Output
  wire [7:0] P;

  // Instantiate the module to be tested
 Unsigned_Multiplication dut (M, Q, P);

  // Initialize inputs
  initial begin

   $dumpfile("dump.vcd");
   $dumpvars(1);


    M = 4'b1111; // Example value for M (10 in decimal)
```

```
        Q = 4'b1111; // Example value for Q (12 in decimal)

        // Wait some time for the output to stabilize
        #10;

        // Display the inputs and output
        $display("Input M: %d, Q: %d", M, Q);
        $display("Output P: %d", P);


    end

endmodule
```