VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Name: UJJVAL RAGHAVENDRA L

Reg no: 22BEC0464

Subject: FPGA-Based System Design Lab

Subject Code: BECE406E

Assignment-2

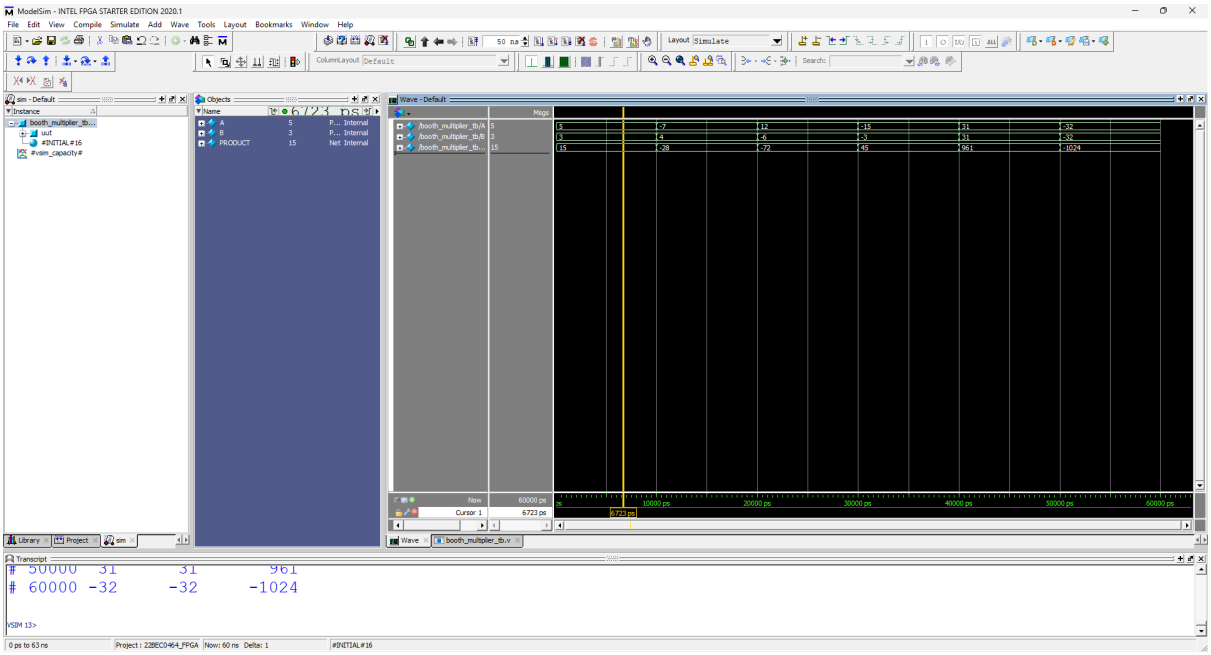
Booth Multiplier

1. Behavioral Modeling

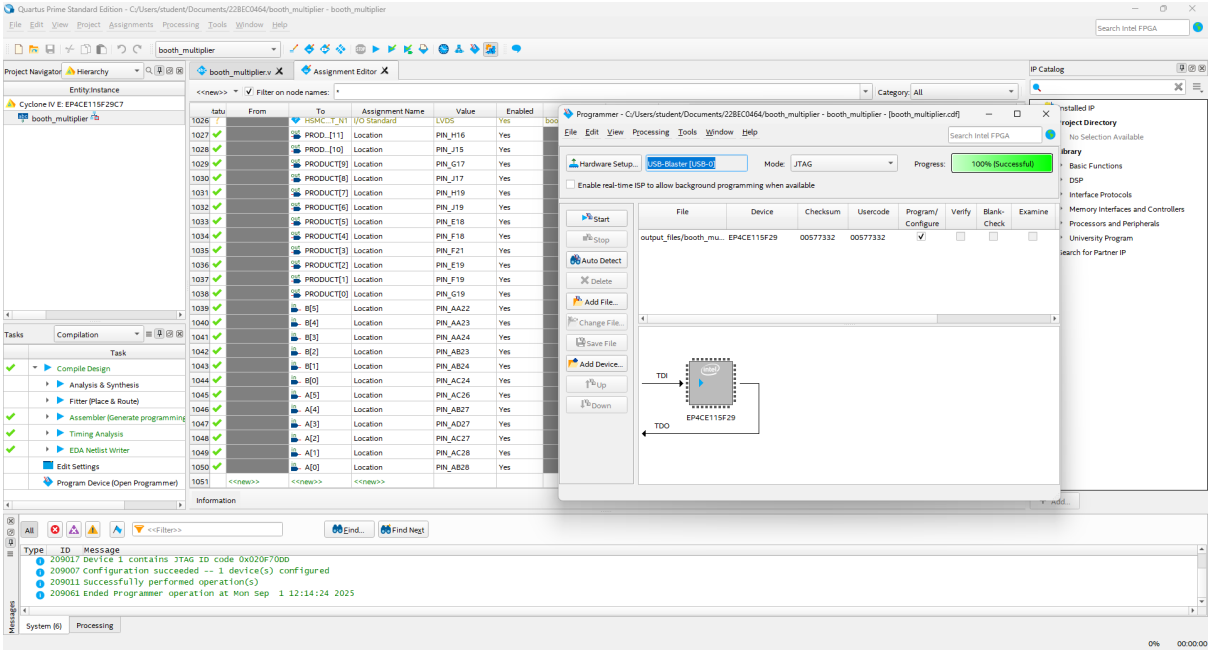
```
`timescale 1ns/1ps

module booth_multiplier(PRODUCT,A,B);
  output reg signed [11:0] PRODUCT;   // 12-bit signed result (since 6-bit × 6-bit)
  input signed [5:0] A,B;             // Two 6-bit signed inputs
  integer i;
  reg Qres;                           // Stores previous Q-1 bit for Booth's
algorithm

  always@(A or B) begin
    PRODUCT = 12'd0;                  // Initialize product to zero
    Qres = 1'b0;                      // Initialize Q-1 = 0

    for(i=0; i<6; i=i+1) begin
      case ({B[i], Qres})             // Check pair (Q[i], Q-1)
        2'd2: PRODUCT[11:6] = PRODUCT[11:6] - A;  // 10 → Subtract A
        2'd1: PRODUCT[11:6] = PRODUCT[11:6] + A;  // 01 → Add A
      endcase

      PRODUCT = PRODUCT >> 1;         // Arithmetic right shift
      PRODUCT[11] = PRODUCT[10];      // Sign-extension
      Qres = B[i];                    // Update Q-1
    end
  end
endmodule
```

2. Behavioral Modeling TestBench

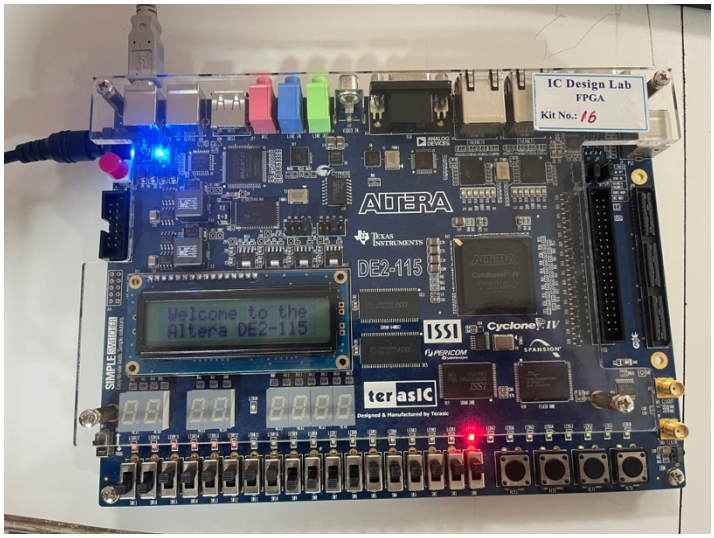
```
`timescale 1ns / 1ps

module booth_multiplier_tb;

  // Testbench signals
  reg signed [5:0] A, B;
  wire signed [11:0] PRODUCT;

  // Instantiate the Booth_Multiplier module
  booth_multiplier uut (
    .PRODUCT(PRODUCT),
    .A(A),
    .B(B)
  );

  initial begin
    $dumpfile("booth_multiplier_tb.vcd");
    $dumpvars(0, booth_multiplier_tb);
    // ... your stimulus and $finish ...

    // Display header
    $display("Time\tA\tB\tProduct");

    // Apply test vectors
    A = 6'd5;   B = 6'd3;    #10;  //  5 * 3 = 15
    $display("%0t\t%d\t%d\t%d", $time, A, B, PRODUCT);

    A = -6'd7; B = 6'd4;    #10;  // -7 * 4 = -28
    $display("%0t\t%d\t%d\t%d", $time, A, B, PRODUCT);

    A = 6'd12; B = -6'd6;  #10;  // 12 * -6 = -72
    $display("%0t\t%d\t%d\t%d", $time, A, B, PRODUCT);

    A = -6'd15; B = -6'd3; #10;  // -15 * -3 = 45
    $display("%0t\t%d\t%d\t%d", $time, A, B, PRODUCT);

    A = 6'd31; B = 6'd31;  #10;  // Max positive 6-bit numbers
    $display("%0t\t%d\t%d\t%d", $time, A, B, PRODUCT);

    A = -6'd32; B = -6'd32; #10; // Max negative 6-bit numbers
    $display("%0t\t%d\t%d\t%d", $time, A, B, PRODUCT);

    // End simulation
    #10;
    $finish;
  end

endmodule
```



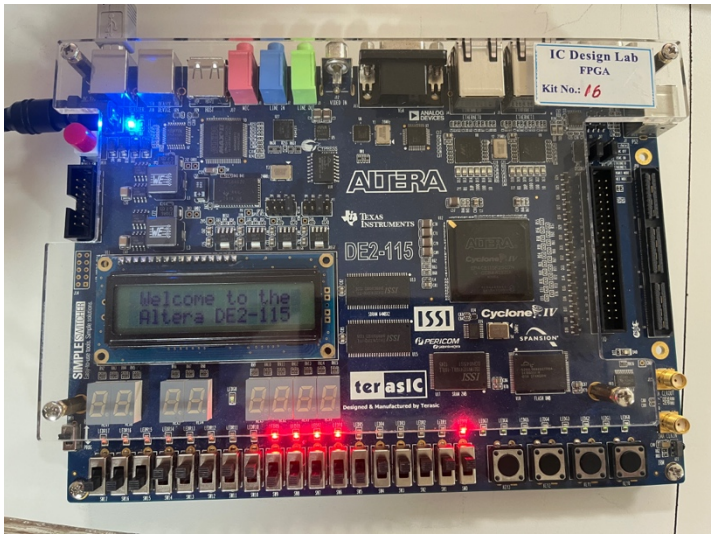
Output Verification Using ModelSim



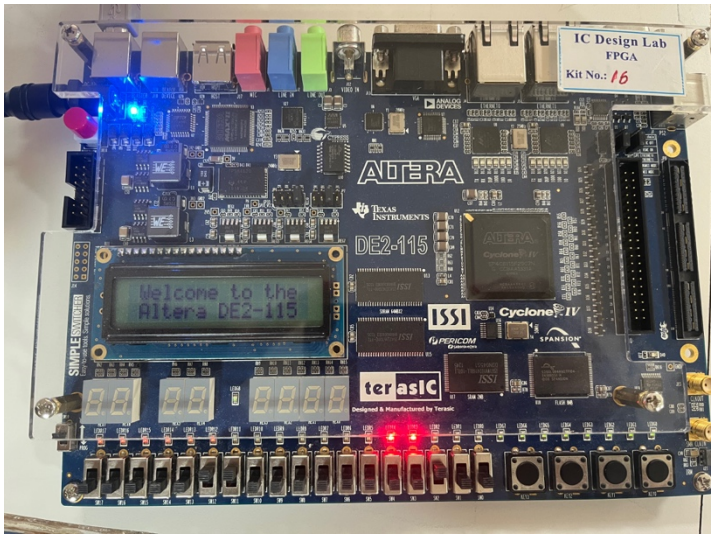
Pin Assignment on FPGA Using Quartus Prime



A=11_1111(2’s Complement=1); B=11_1111(2’s Complement=1); Product=0000_0000_0001(Signed Output)



A=01_1111(31); B=01_1111(31); Product=0011_1100_0001(961)



A=00_0100(4); B=00_0110(6); Product=0000_0001_1000(24)