Lab 11

A Taste of Data path + Control Design Example: Factorial Circuit (Open Ended Lab)

Spring 2025

Submitted by: **Mohsin Sajjad**

Registration No: **22pwsce2149**

Class Section: **A**

"On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work."

Mohsin Sajjad

Student Signature: ______________

Submitted to:

Engr. Faheem Jan

Month Day, Year (22 05, 2025)

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

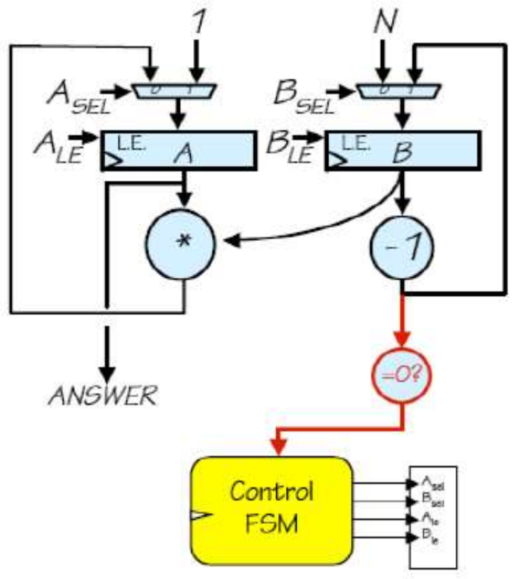
A Taste of Data path + Control Design Example: Factorial Circuit (Open Ended Lab)

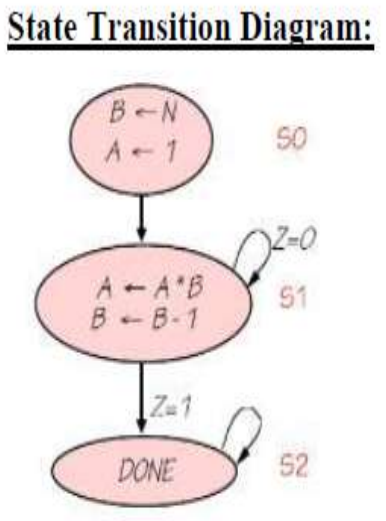
Objective:

To implement a circuit that calculates factorial of a number.

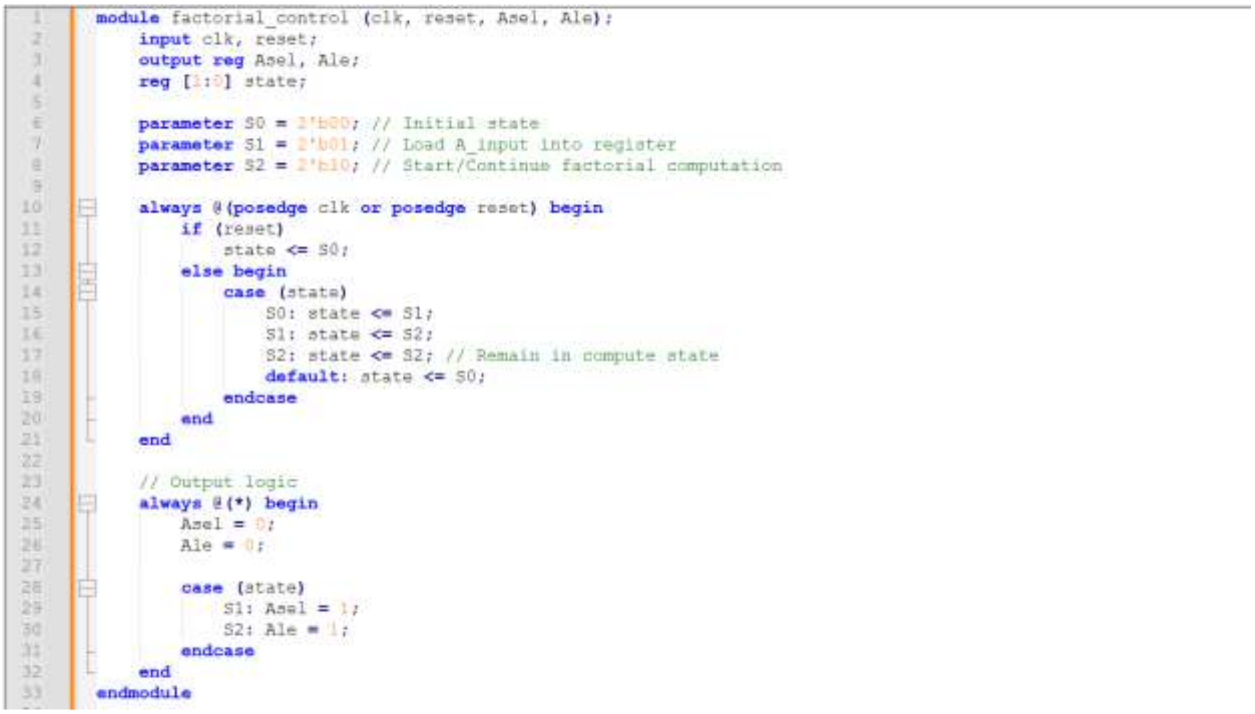
Block Diagram:

The datapath for calculating the factorial of an “N” bit number is given below. The datapath is controlled by a Finite State Machine (FSM). FSM generates signals Asel, Ale, Bsel and Ble at the correct times to operate the datapath. Input to FSM is a signal “Z” when Z=0 means the operation of the machine is complete and ANSWER can be read. The STG is also given in the following diagram.

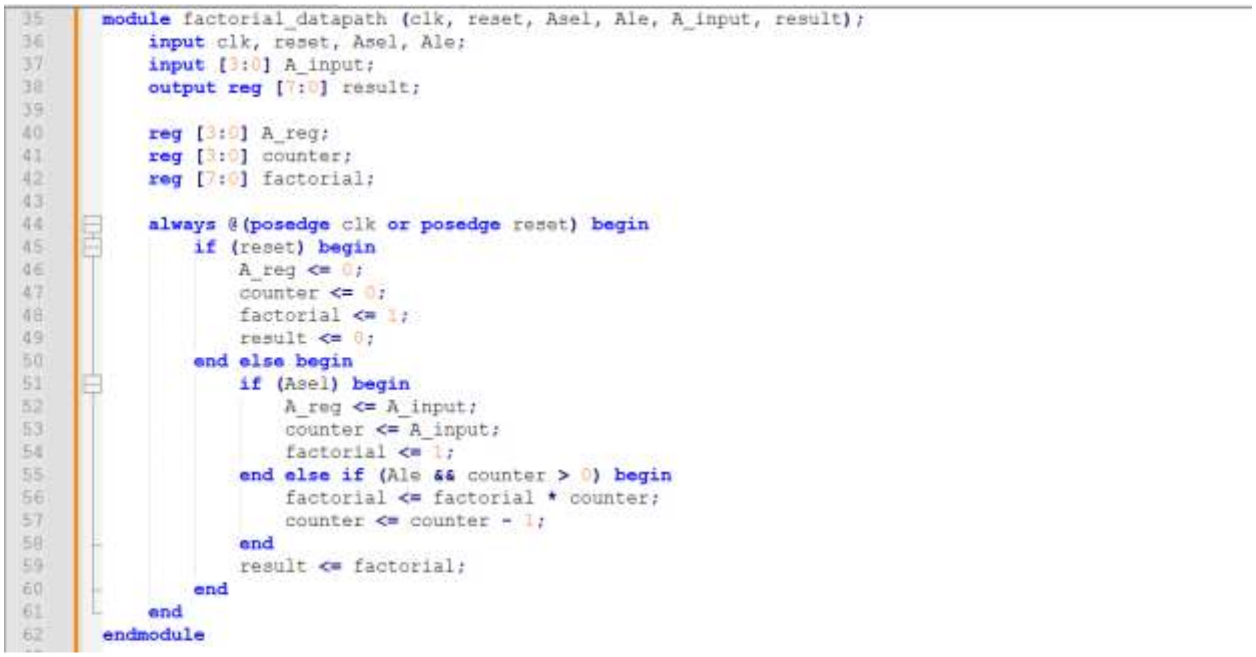




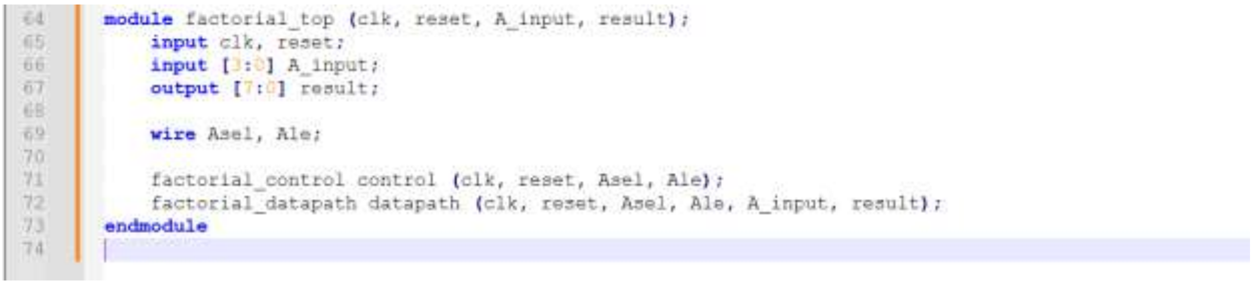
CODE:



```
module factorial_control (clk, reset, Asel, Ale);
    input clk, reset;
    output reg Asel, Ale;
    reg [1:0] state;

    parameter S0 = 2'b00; // Initial state
    parameter S1 = 2'b01; // Load A_input into register
    parameter S2 = 2'b10; // Start/Continue factorial computation

    always @(posedge clk or posedge reset) begin
        if (reset)
            state <= S0;
        else begin
            case (state)
                S0: state <= S1;
                S1: state <= S2;
                S2: state <= S2; // Remain in compute state
                default: state <= S0;
            endcase
        end
    end

    // Output logic
    always @(*) begin
        Asel = 0;
        Ale = 0;

        case (state)
            S1: Asel = 1;
            S2: Ale = 1;
        endcase
    end
endmodule
```

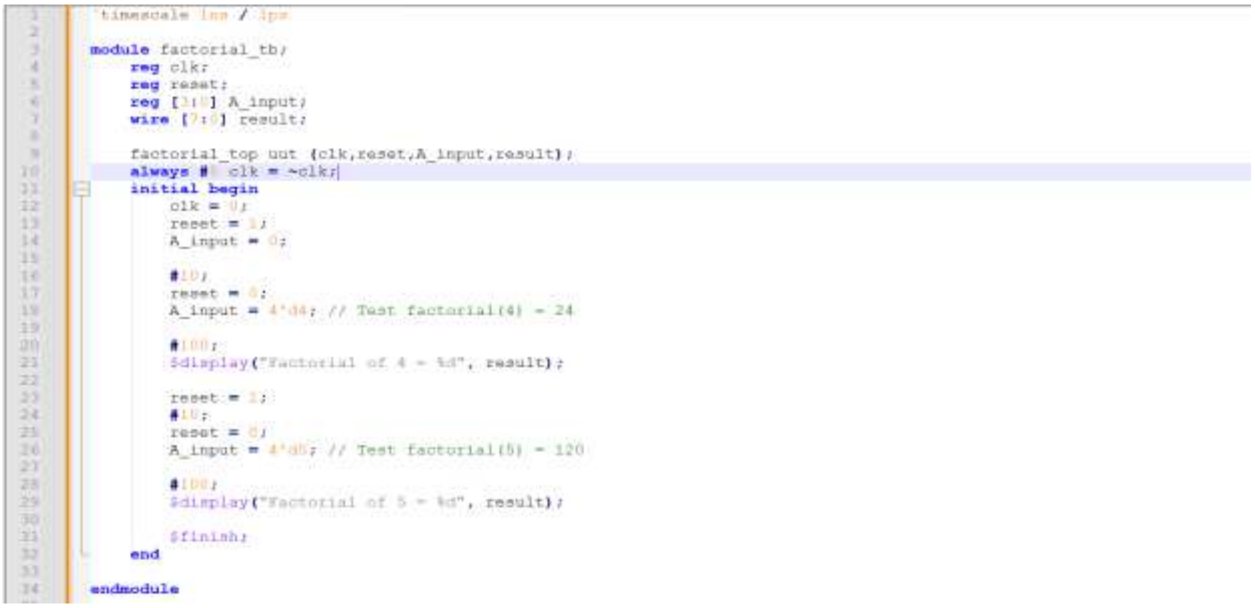


```
module factorial_datapath (clk, reset, Asel, Ale, A_input, result);
    input clk, reset, Asel, Ale;
    input [3:0] A_input;
    output reg [7:0] result;

    reg [3:0] A_reg;
    reg [3:0] counter;
    reg [7:0] factorial;

    always @(posedge clk or posedge reset) begin
        if (reset) begin
            A_reg <= 0;
            counter <= 0;
            factorial <= 1;
            result <= 0;
        end else begin
            if (Asel) begin
                A_reg <= A_input;
                counter <= A_input;
                factorial <= 1;
            end else if (Ale && counter > 0) begin
                factorial <= factorial * counter;
                counter <= counter - 1;
            end
            result <= factorial;
        end
    end
endmodule
```



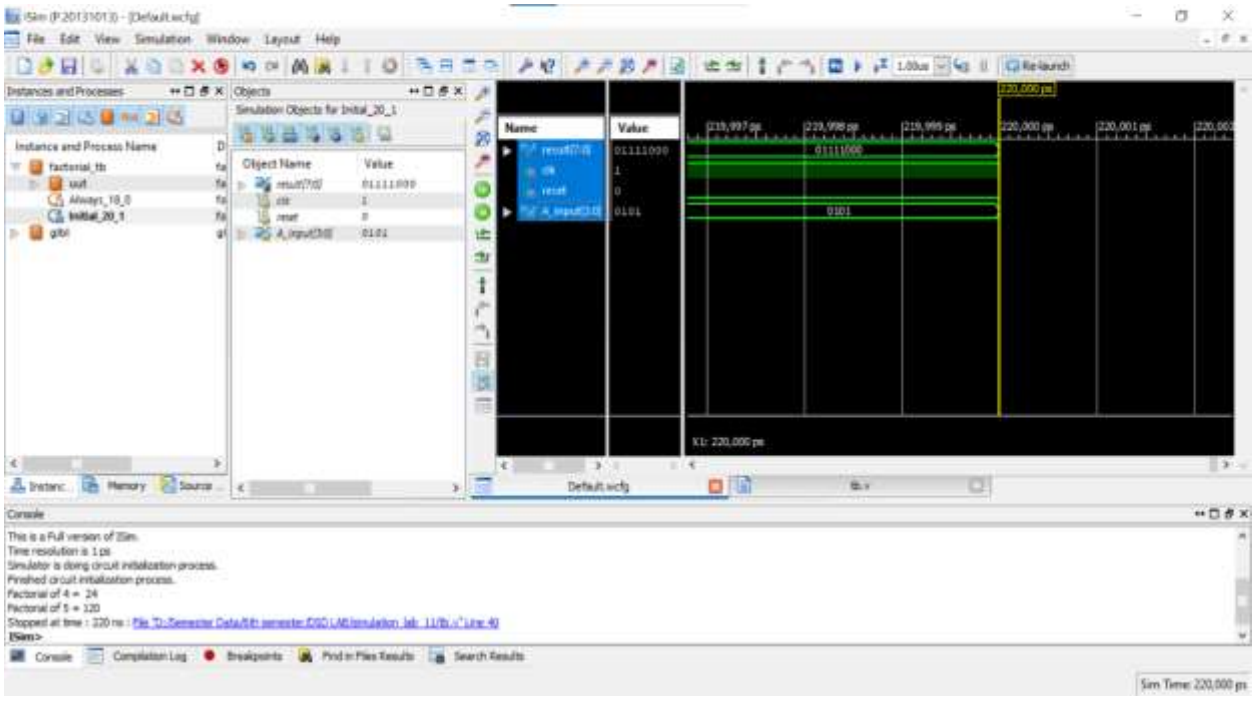
```verilog
module factorial_top (clk, reset, A_input, result);
    input clk, reset;
    input [3:0] A_input;
    output [7:0] result;

    wire Asel, Ale;

    factorial_control control (clk, reset, Asel, Ale);
    factorial_datapath datapath (clk, reset, Asel, Ale, A_input, result);
endmodule
```

Test-Bench:



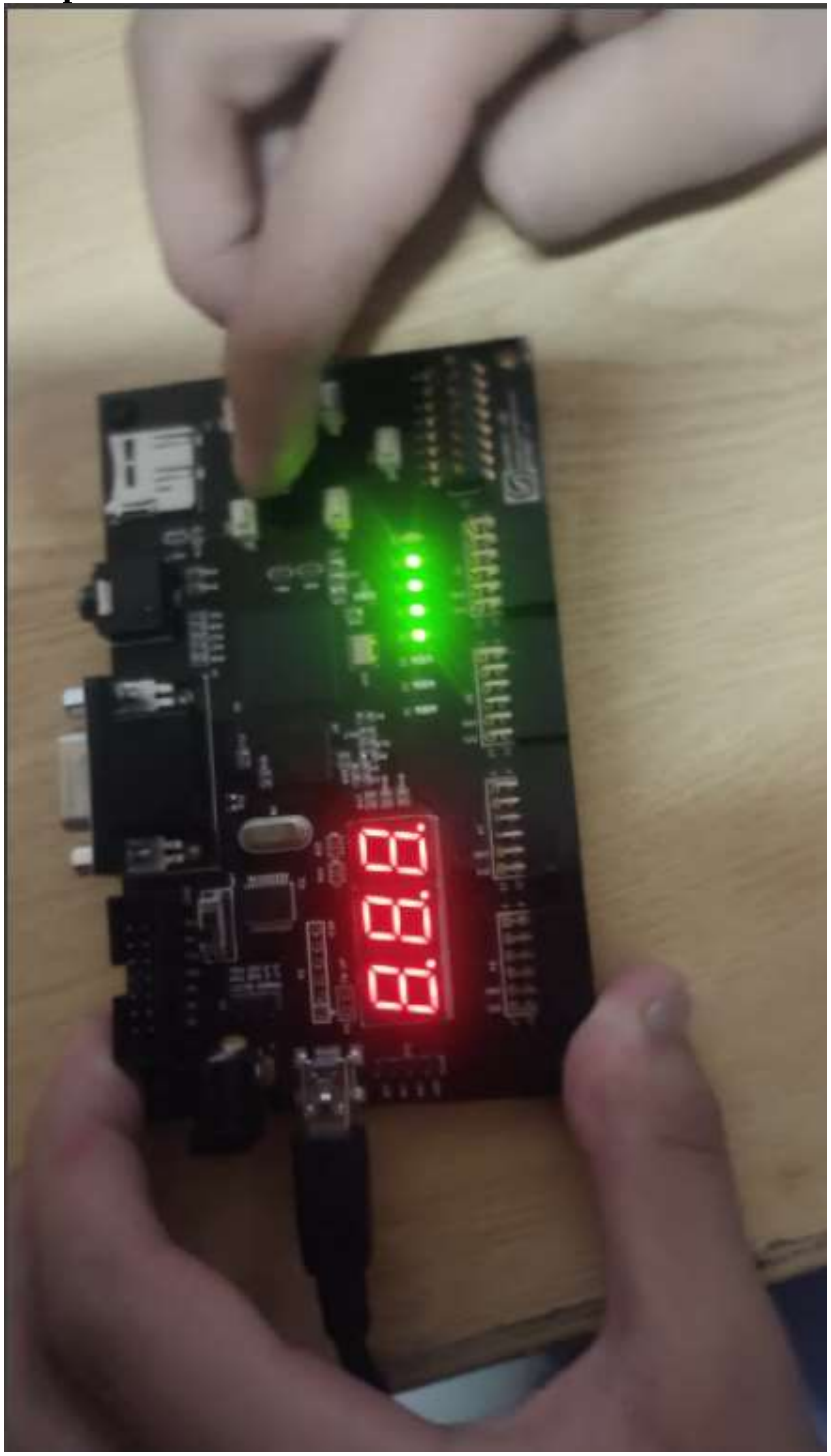
```verilog
`timescale 1ns / 1ps

module factorial_tb;
    reg clk;
    reg reset;
    reg [3:0] A_input;
    wire [7:0] result;

    factorial_top uut (clk,reset,A_input,result);
    always #5 clk = ~clk;
    initial begin
        clk = 0;
        reset = 1;
        A_input = 0;

        #10;
        reset = 0;
        A_input = 4'd4; // Test factorial(4) = 24

        #100;
        $display("Factorial of 4 = %d", result);

        reset = 1;
        #10;
        reset = 0;
        A_input = 4'd5; // Test factorial(5) = 120

        #100;
        $display("Factorial of 5 = %d", result);

        $finish;
    end

endmodule
```

OUTPUT:



```
This is a Full version of ISim.
Time resolution is 1 ps
Simulator is doing circuit initialization process.
Finished circuit initialization process.
Factorial of 4 =  24
Factorial of 5 = 120
Stopped at time : 220 ns : File "D:/Semester Data/6th semester/DSD LAB/simulation_lab_11/tb.v" Line 40
ISim>
```

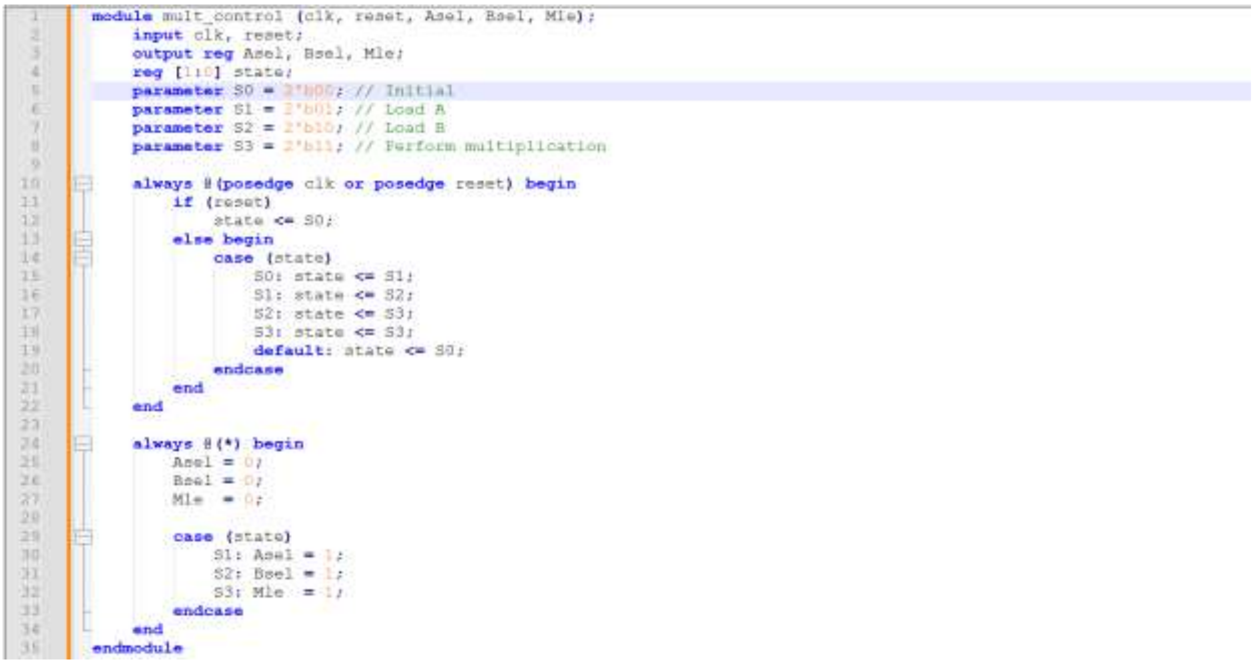
Output on FPGA:



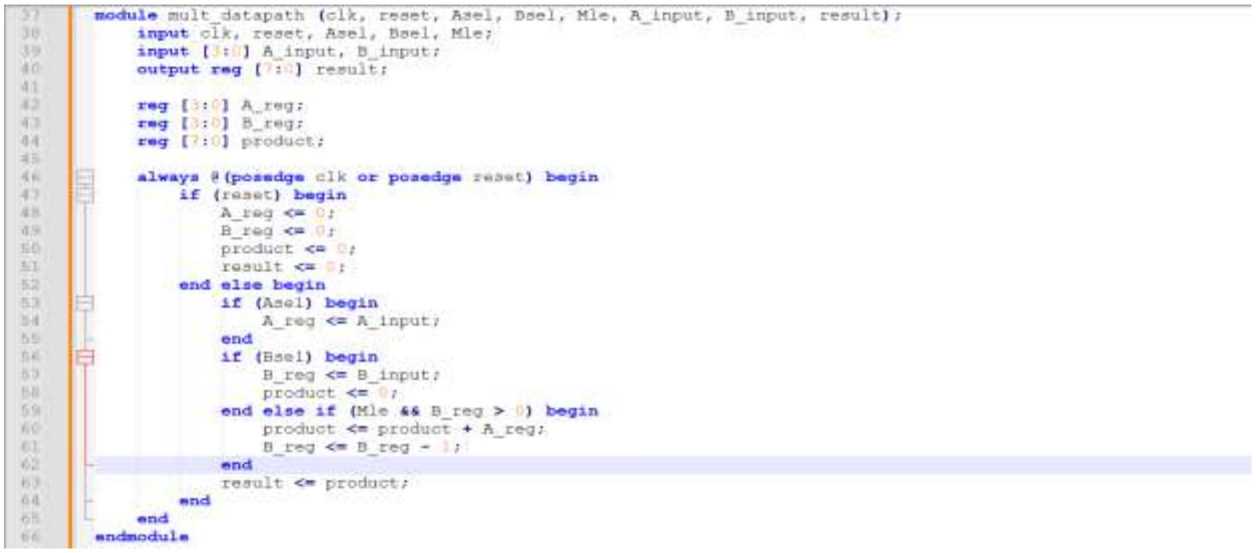
Task 2: Implement Multiplication for repeated addition using data path and control path (FSM)

Note: you can do simulation as well

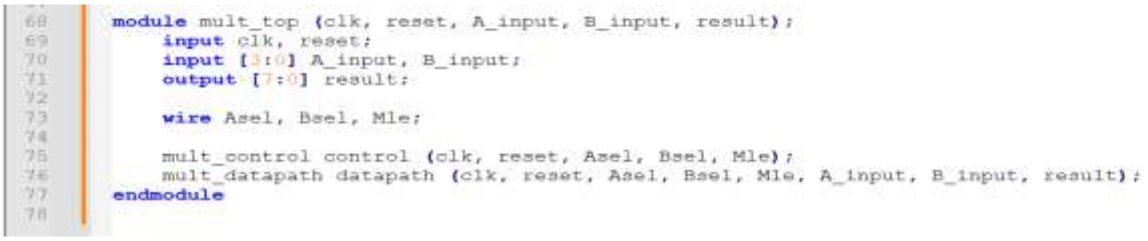
CODE:



```verilog
module mult_control (clk, reset, Asel, Bsel, Mle);
    input clk, reset;
    output reg Asel, Bsel, Mle;
    reg [1:0] state;
    parameter S0 = 2'b00; // Initial
    parameter S1 = 2'b01; // Load A
    parameter S2 = 2'b10; // Load B
    parameter S3 = 2'b11; // Perform multiplication

    always @(posedge clk or posedge reset) begin
        if (reset)
            state <= S0;
        else begin
            case (state)
                S0: state <= S1;
                S1: state <= S2;
                S2: state <= S3;
                S3: state <= S3;
                default: state <= S0;
            endcase
        end
    end

    always @(*) begin
        Asel = 0;
        Bsel = 0;
        Mle  = 0;

        case (state)
            S1: Asel = 1;
            S2: Bsel = 1;
            S3: Mle  = 1;
        endcase
    end
endmodule
```

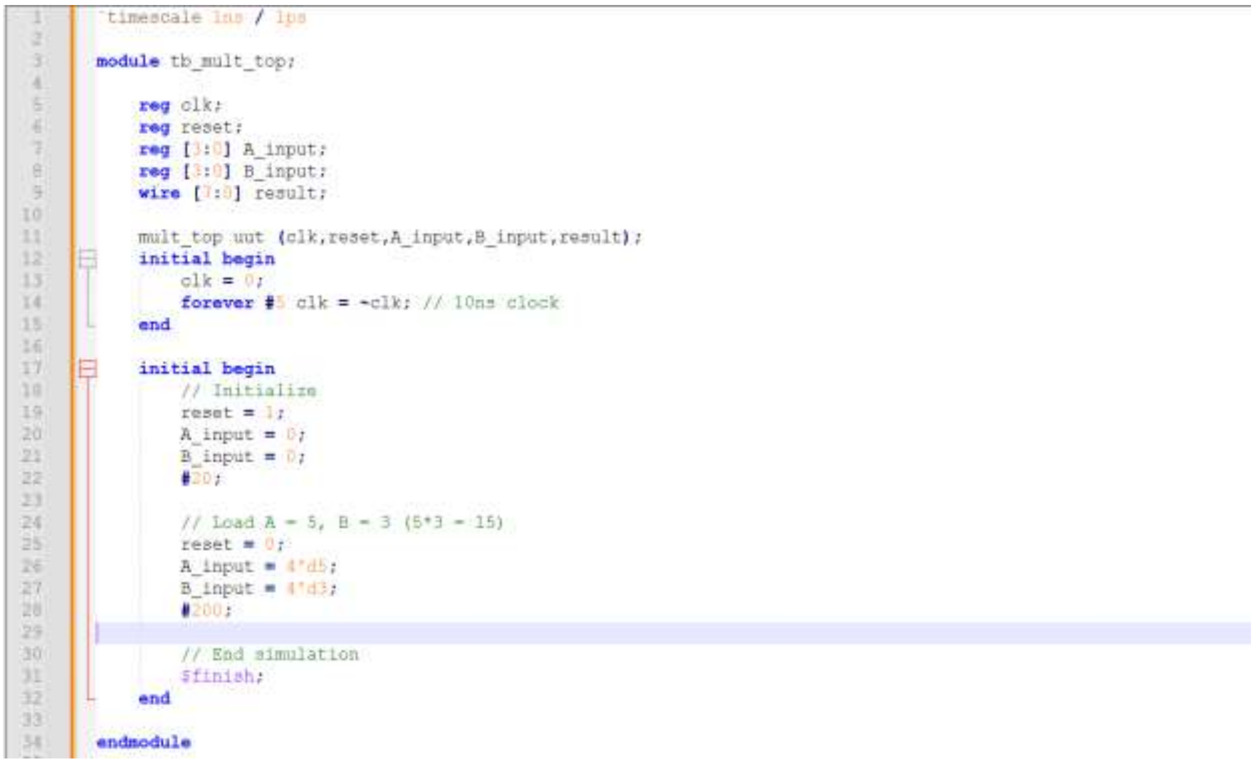


```verilog
module mult_datapath (clk, reset, Asel, Bsel, Mle, A_input, B_input, result);
    input clk, reset, Asel, Bsel, Mle;
    input [3:0] A_input, B_input;
    output reg [7:0] result;

    reg [3:0] A_reg;
    reg [3:0] B_reg;
    reg [7:0] product;

    always @(posedge clk or posedge reset) begin
        if (reset) begin
            A_reg <= 0;
            B_reg <= 0;
            product <= 0;
            result <= 0;
        end else begin
            if (Asel) begin
                A_reg <= A_input;
            end
            if (Bsel) begin
                B_reg <= B_input;
                product <= 0;
            end else if (Mle && B_reg > 0) begin
                product <= product + A_reg;
                B_reg <= B_reg - 1;
            end
            result <= product;
        end
    end
endmodule
```



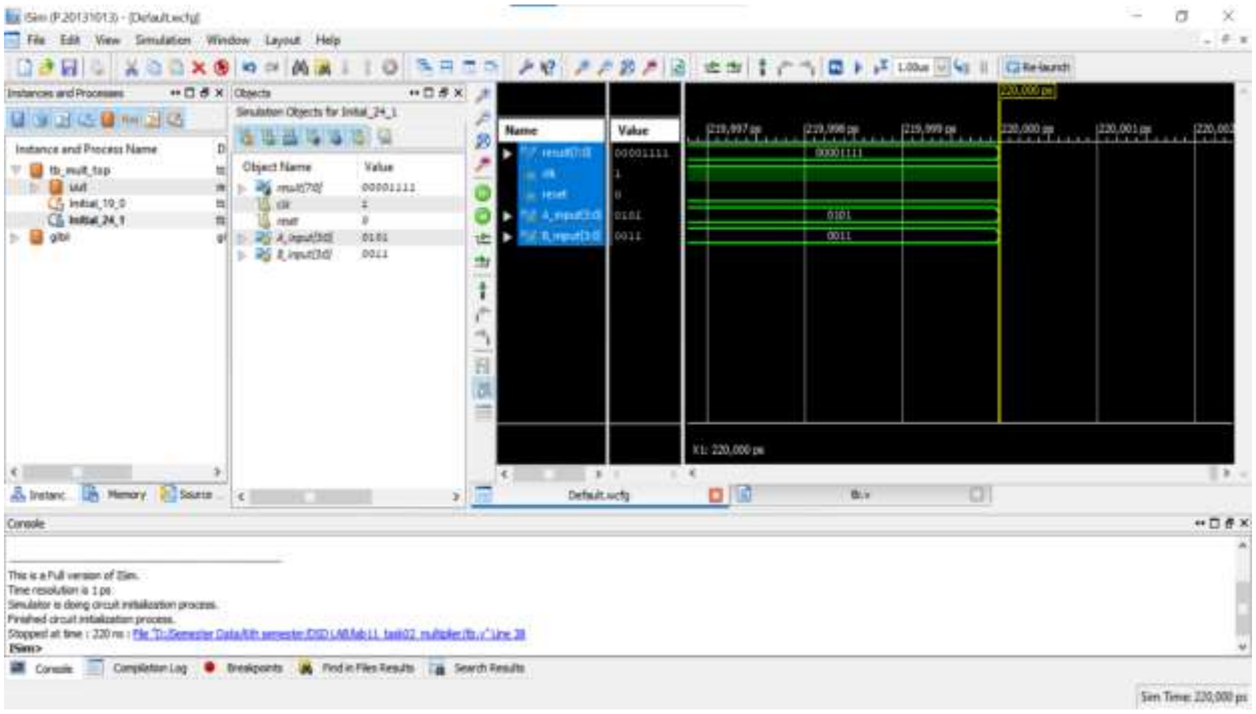
```verilog
module mult_top (clk, reset, A_input, B_input, result);
    input clk, reset;
    input [3:0] A_input, B_input;
    output [7:0] result;

    wire Asel, Bsel, Mle;

    mult_control control (clk, reset, Asel, Bsel, Mle);
    mult_datapath datapath (clk, reset, Asel, Bsel, Mle, A_input, B_input, result);
endmodule
```

Test-Bench:



```
`timescale 1ns / 1ps

module tb_mult_top;

    reg clk;
    reg reset;
    reg [3:0] A_input;
    reg [3:0] B_input;
    wire [7:0] result;

    mult_top uut (clk,reset,A_input,B_input,result);
    initial begin
        clk = 0;
        forever #5 clk = ~clk; // 10ns clock
    end

    initial begin
        // Initialize
        reset = 1;
        A_input = 0;
        B_input = 0;
        #20;

        // Load A = 5, B = 3 (5*3 = 15)
        reset = 0;
        A_input = 4'd5;
        B_input = 4'd3;
        #200;

        // End simulation
        $finish;
    end

endmodule
```

OUTPUT:



ISim (P.20131013) - [Default.wcfg]
File Edit View Simulation Window Layout Help
Instances and Processes
Objects
Simulation Objects for Initial_24_1
Re-launch
Default.wcfg
Console
Console Compilation Log Breakpoints Find in Files Results Search Results
Sim Time: 220,000 ps

Instance and Process Name
tb_mult_top
uut
Initial_10_0
Initial_24_1
glbl

Object Name	Value
result[7:0]	00001111
clk	1
reset	0
A_input[3:0]	0101
B_input[3:0]	0011

Name	Value
result[7:0]	00001111
clk	1
reset	0
A_input[3:0]	0101
B_input[3:0]	0011

```
220,000 ps
219,997 ps   219,998 ps   219,999 ps   220,000 ps   220,001 ps   220,002
00001111
0101
0011
X1: 220,000 ps
```

```
This is a Full version of ISim.
Time resolution is 1 ps
Simulator is doing circuit initialization process.
Finished circuit initialization process.
Stopped at time : 220 ns : File "D:/Semester Data/[illegible] semester/DSD LAB/lab11_task02_multiplier/tb_.v" Line [illegible]
ISim>
```