University of Colorado Boulder
ECEE Department

ECEN 2350 - Digital Logic - Fall 2023
Location: Engineering Center, ECCR 1B40, MWF 1:25PM - 2:15PM
Instructor: Dr. Mona ElHelbawy
Lab #6: FSM Controller
Date of Experiment: November 16th, 2023
Names: Connor Sorrell

Description

In this lab, we will implement several things in order to multiply two 4-bit unsigned binary numbers (our FSM Controller). First, we will put together an accumulator and shift register in order to implement the 4 bit multiplication. Next we will implement a finite state machine. We will then connect the finite state machine to the shift register and accumulator, as well as the universal seven segment code.

Multiply FSM

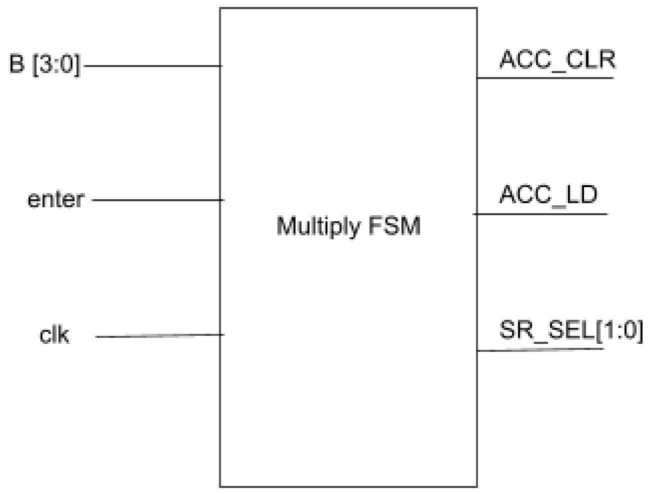
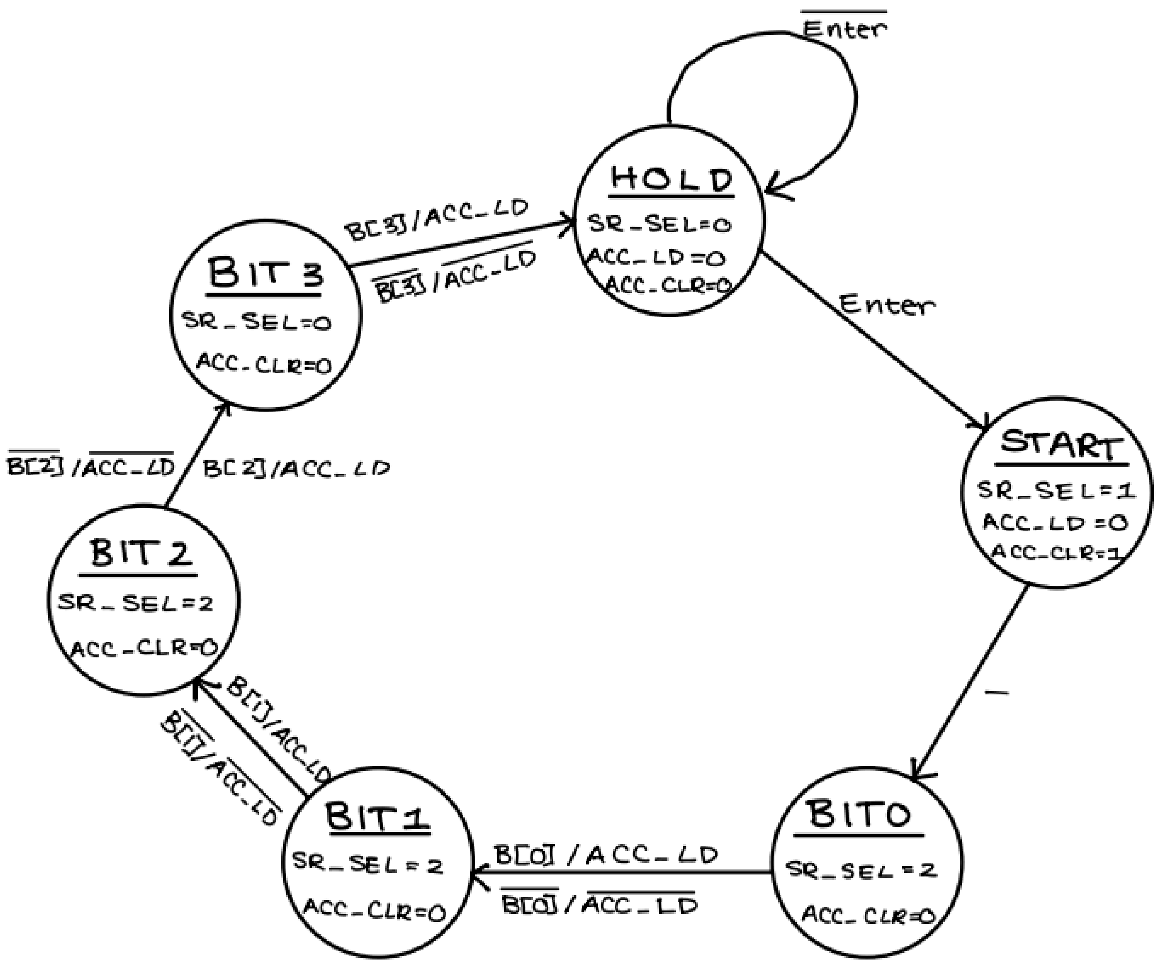


Figure 1.1: Black box diagram of the Multiply FSM showing input and output ports.



$\overline{\text{Enter}}$
HOLD
SR_SEL=0
ACC_LD=0
ACC_CLR=0
Enter
START
SR_SEL=1
ACC_LD=0
ACC_CLR=1
–
BIT0
SR_SEL=2
ACC_CLR=0
B[0] / ACC_LD
$\overline{\text{B[0]}}$ / $\overline{\text{ACC\_LD}}$
BIT1
SR_SEL=2
ACC_CLR=0
B[1]/ACC_LD
$\overline{\text{B[1]}}$/$\overline{\text{ACC\_LD}}$
BIT2
SR_SEL=2
ACC_CLR=0
$\overline{\text{B[2]}}$ /$\overline{\text{ACC\_LD}}$
B[2]/ACC_LD
BIT3
SR_SEL=0
ACC_CLR=0
B[3]/ACC_LD
$\overline{\text{B[3]}}$ /$\overline{\text{ACC\_LD}}$

Figure 1.2: FSM State Diagram

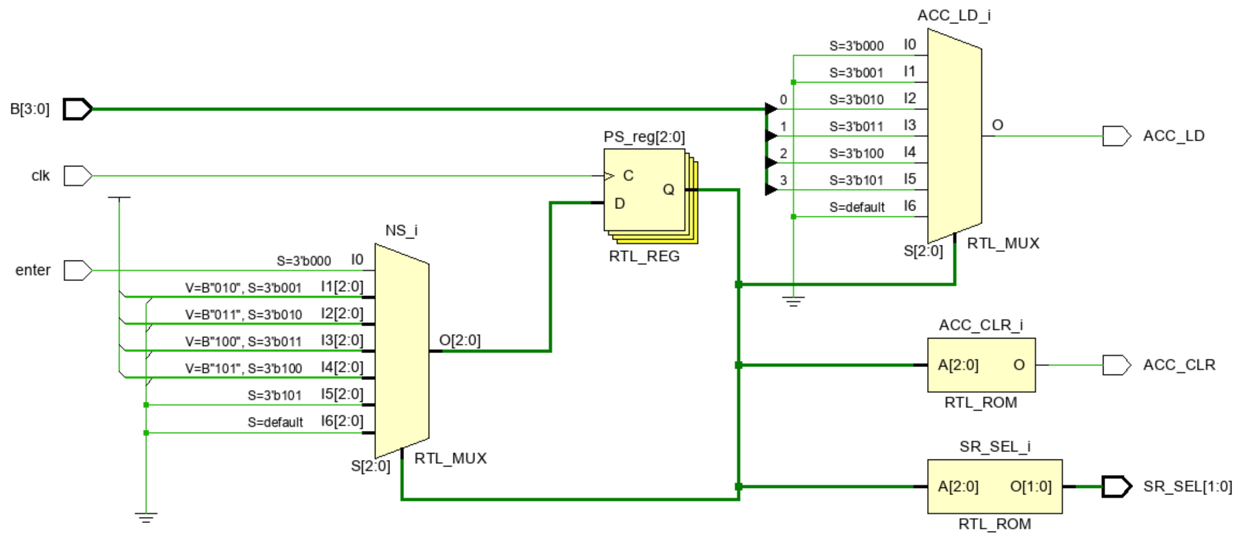


Figure 1.3: Structural Model for the Multiply FSM.

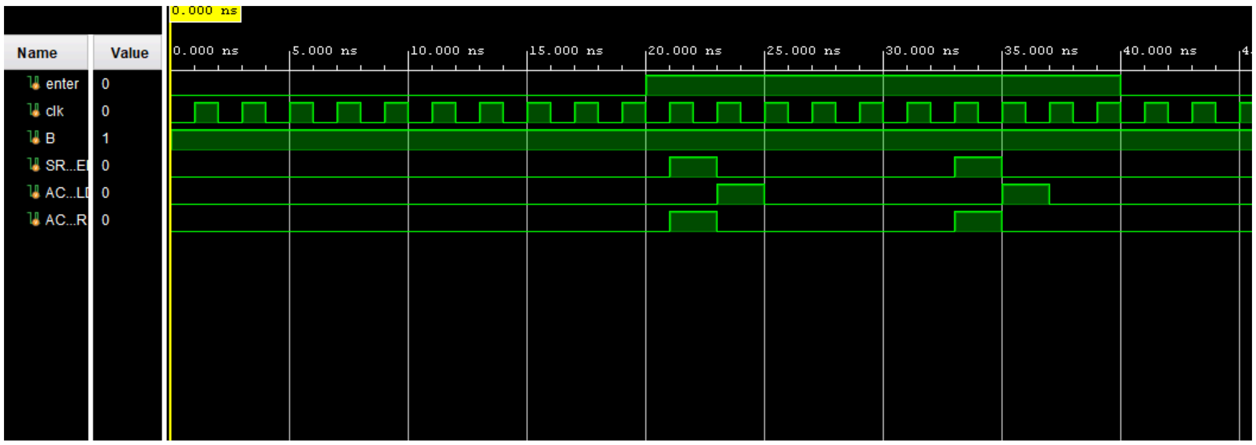


Figure 1.4: Simulation timing diagram for the Multiply FSM. Shows the expected intermediate and final outputs.

Multiply FSM Design Code:

```
module FSM(
  input enter,
  input clk,
  input [3:0] B,
  output logic [1:0] SR_SEL,
  output logic ACC_LD, ACC_CLR
  );

  logic [2:0] NS;
  logic [2:0] PS = 0;

  parameter [2:0] HOLD = 3'b000, START = 3'b001, BIT0=3'b010, BIT1 = 3'b011, BIT2 = 3'b100, BIT3 = 3'b101;

  always_ff @ (posedge clk)
  begin
    PS <= NS;
  end

  always_comb
  begin

    SR_SEL = 0; ACC_LD = 0; ACC_CLR = 0;
    case(PS)
    HOLD:begin

    SR_SEL = 0;
    ACC_LD = 0;
    ACC_CLR = 0;
    if (enter)
      NS = START;
    else
      NS = HOLD;
  end

  START:
  begin

    SR_SEL = 1;
    ACC_LD = 0;
    ACC_CLR = 1;
    NS = BIT0;
  end
```

```
    BIT0:
    begin
      SR_SEL = 2;
      ACC_CLR = 0;

      if (B[0])
        ACC_LD = 1;
      else
        ACC_LD = 0;
      NS = BIT1;
    end

    BIT1: begin
      SR_SEL = 2;
      ACC_CLR = 0;
      if(B[1])
        ACC_LD = 1;
      else
        ACC_LD = 0;
      NS=BIT2;
    end

    BIT2:begin
    SR_SEL = 2;
    ACC_CLR = 0;
    if (B[2])
      ACC_LD = 1;
    NS = BIT3;
    end

    BIT3:begin
      SR_SEL = 2;
      ACC_CLR = 0;
      if (B[3])
        ACC_LD = 1;
      else
        ACC_LD = 0;
      NS = HOLD;
      end
    default:
      NS = HOLD;
    endcase
    end
  endmodule
```

Multiply FSM Sim Code:

```
module FSM_sim();
  logic enter, clk, B, SR_SEL, ACC_LD, ACC_CLR;

  FSM FSM_sim (.*);

  always begin

    clk = 0;
    #1;
    clk = 1;
    #1;

  end

  initial
    begin


    enter = 0;
    B = 4'b0111;
    #20;
    enter = 1;
    #20;
    enter = 0;
    #20;


  end


endmodule
```

Lab 6 Top Module

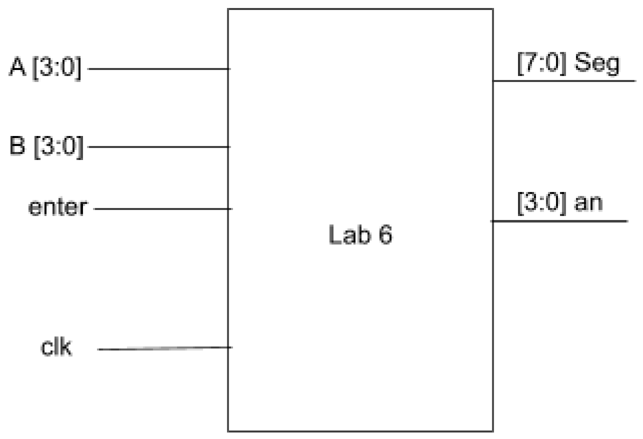


Figure 1.5: Black Box diagram for the Lab 6 Top Module showing the inputs and outputs.

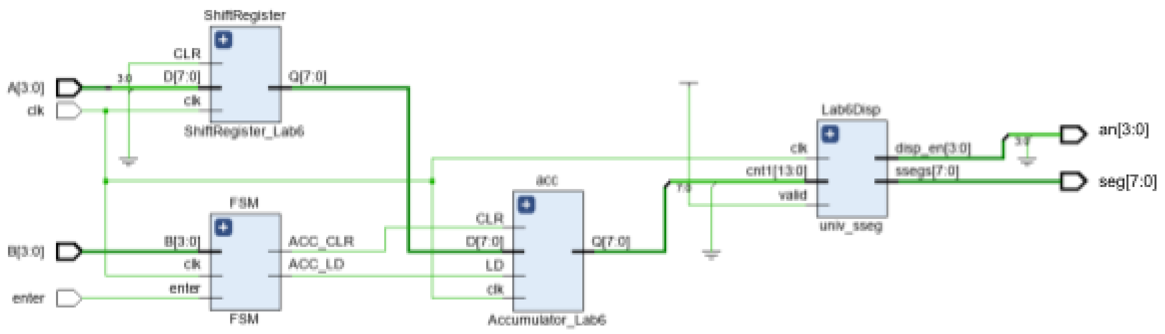


Figure 1.6: Elaborated Design of the Lab 6 Top Level Module

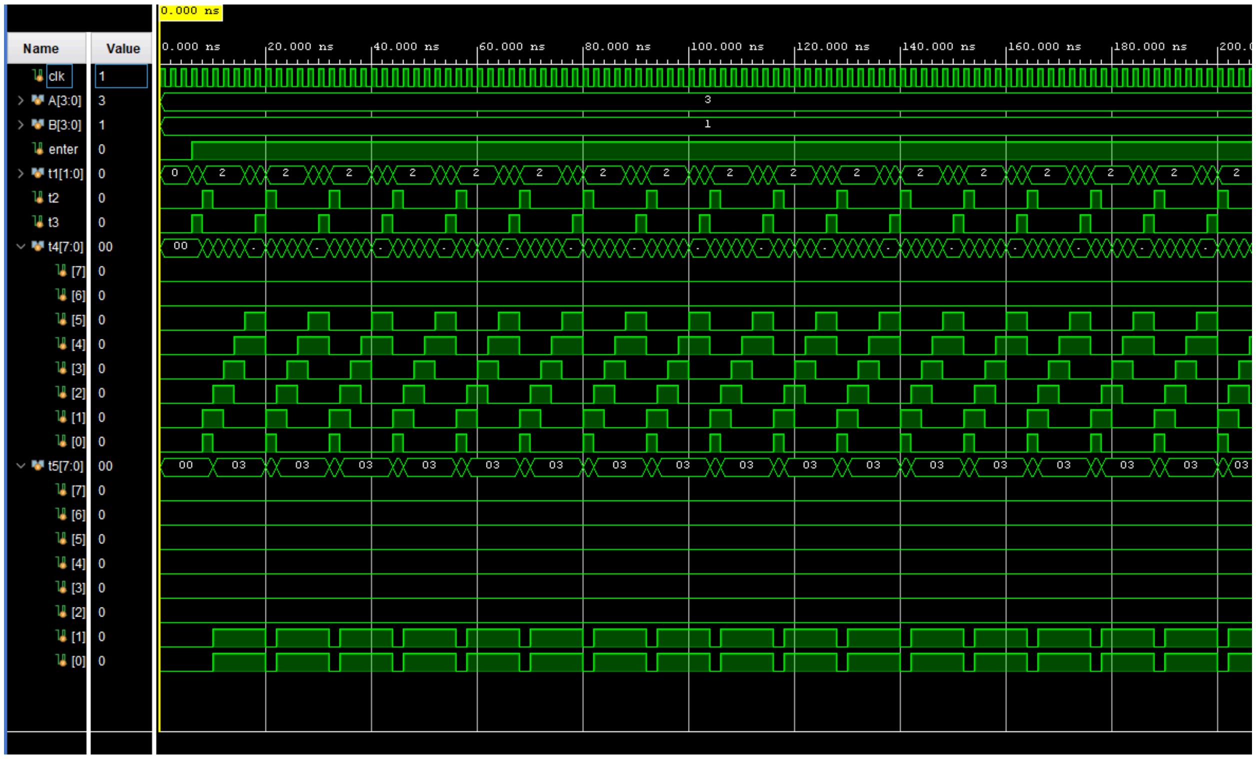


Figure 1.7: Simulation Timing Diagram showing A, B, PS, NS, t1, t2, t3, t4, and t5 intermediate outputs. When enter becomes 1 (4th clock cycle), we start to see values of t2 and t3. They then run through the accumulator and shift register and the intermediate outputs (t5) result as expected.

<u>Lab 6 Top Level Design Code:</u>

```
module Lab6(
    input clk,
    input [3:0] A,
    input [3:0] B,
    input enter,
    output [7:0] seg,
    output [3:0] an
    );
    logic [1:0] t1;
    logic t2,t3;
    logic [7:0] t4, t5;
    FSM FSM(.clk(clk), .enter(enter), .B(B), .SR_SEL(t1), .ACC_LD(t2), .ACC_CLR(t3) );
    ShiftRegister_Lab6 ShiftRegister (.clk(clk), .CLR(0), .SEL(t2), .D({4'b0,A}), .Q(t4));
    Accumulator_Lab6 acc (.clk(clk), .LD(t2), .CLR(t3), .D(t4), .Q(t5));
    univ_sseg Lab6Disp (.clk(clk), . cnt1({6'b0,t5}), .valid(1), .ssegs(seg), .disp_en(an));

endmodule
```

Lab 6 Top Level Simulation Code:

```
module Lab6_sim();

    logic clk;
    logic [3:0] A;
    logic [3:0] B;
    logic enter;
    logic [7:0] seg;
    logic [3:0] an;

    Lab6 sim(.*);

    initial begin
    clk = 0;
    end

    always
    begin
    clk = ~clk;
    #1;
    end

    initial
    begin
    enter = 0;
    A = 3;
    B = 1;
    #6
    enter = 1;
    #12;
    end
    endmodule
```