[과제 3] 4bit x 4bit = 8bit Multiplier(Unsigned Magnitude) 설계

```
//과제 3
//이름: 이창민
//학번:2019043890
//융합전자공학부
```

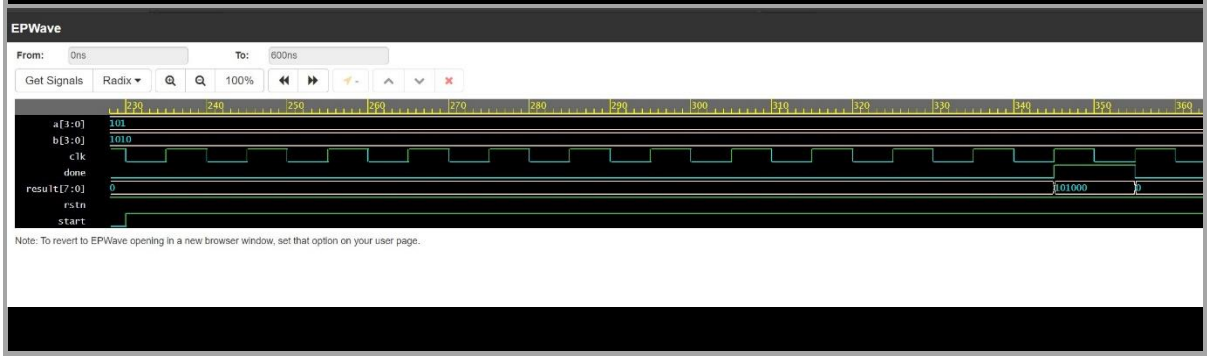
```
// Code your design here
module mul8_uns (
  input [3:0] a,
  input [3:0] b,
  input clk, rstn, start,
  output reg [7:0] result,
  output reg done
);

  localparam IDLE = 3'b000,
             START = 3'b001,
             LSB = 3'b010,
             ADD = 3'b011,
             SHIFT = 3'b100,
             DONE = 3'b101;

  reg [7:0] r_multiplicand, r_product;
  reg [3:0] r_multiplier;
  reg [2:0] r_state,next_state;
  reg [1:0] r_count;

always @(posedge clk, negedge rstn) begin
    if (!rstn) r_state <= IDLE;
    else r_state<=next_state;
end
always @(*) begin
  case(r_state)
    IDLE:
      begin
        if (start) begin
            next_state = START;
        end
        else begin
            next_state = IDLE;
        end
      end
    START:
      begin
        next_state=LSB;
      end
```

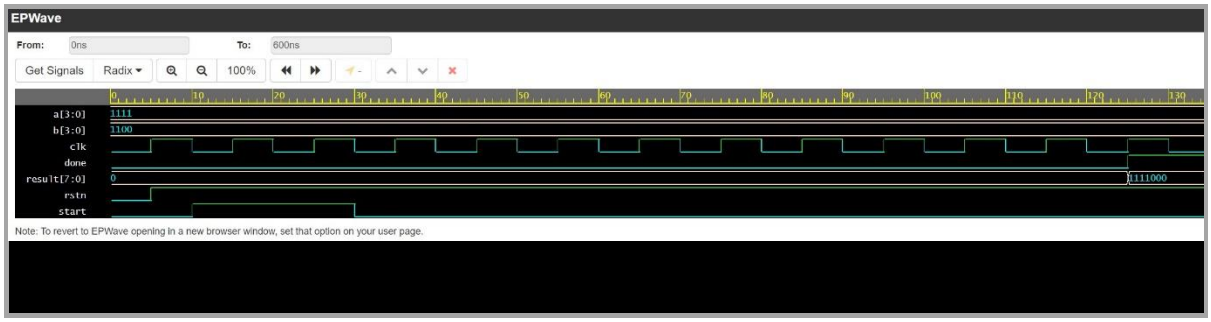
```
    LSB:
      begin
        if(r_multiplier[0]) next_state = ADD;
        else next_state = SHIFT;
      end
    ADD:
      begin
        next_state = SHIFT;
      end
    SHIFT:
      begin
        if(r_count != 0)next_state=LSB;
        else next_state = DONE;
      end
    DONE:
      begin
        next_state= IDLE;
      end
    default:
      begin
        next_state= IDLE;
      end
    endcase
end
always @(posedge clk, negedge rstn) begin
    if (!rstn) begin
      r_multiplicand <= 0;
      r_multiplier <= 0;
      r_product <= 0;
      r_count <= 4;
      result <= 0;
      done <= 0;
    end
    else begin
      case (next_state)
        IDLE:
          begin
            r_multiplicand <= 0;
            r_multiplier <= 0;
            r_product <= 0;
            r_count <= 4;
            result <= 0;
            done <= 0;
          end
        START:
        begin
            r_multiplicand <= {4'b0000, a};
```

```
            r_multiplier = b;
            r_product <= 0;
            r_count <= 4;
            result <= 0;
            done <= 0;
        end
        LSB:begin
            r_multiplicand <= r_multiplicand; //순차회로에서 자기자신을
기억해라
            r_multiplier <= r_multiplier;
            r_count<=r_count -1;

            result <= 0;
            done <= 0;
        end

        ADD:
          begin
            r_product = r_multiplicand + r_product;
          end
        SHIFT:
          begin
            r_multiplicand = r_multiplicand << 1;
            r_multiplier = r_multiplier >> 1;

          end
        DONE:
          begin
            result = r_product;
            done = 1;
          end
        endcase
      end
      end

endmodule
```

```
// Code your testbench here
module tb;
    reg clk, rstn, start;
    reg [3:0] a,b;

    wire [7:0]result;
    wire done;

    initial begin
        clk=0;
```

```
        forever begin
            #5 clk=!clk;
        end
    end

    mul8_uns mul8_uns(a,b,clk, rstn, start, result, done);


    initial begin
        a=15; b=12;
    #200 a=5; b=10;

    #400 $finish;
    end

    initial begin
        rstn = 0;
        #5 rstn=1;
    end

    initial begin
        start = 0;
        #10 start = 1;
        #20 start = 0;
        #200 start = 1;
    end

    initial begin
        $dumpfile("wave.vcd");
        $dumpvars(0,tb);
    end
endmodule
```





노트북에 VIVADO 가 설치가 안돼서 https://edaplayground.com/ 를 이용하였습니다.

testbench 에서 두가지 인풋을 여유있는 시간 텀을 두어 곱해보았고 정확한 값이 나오는 것을 확인할 수 있었습니다.