[과제3] 4bit x 4bit = 8bit Multiplier(Unsigned Magnitude) 설계

|  |
| --- |
| //과제3  //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 VIVADO가 설치가 안돼서 <https://edaplayground.com/> 를 이용하였습니다.

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