**Họ và tên: Hà Minh Khuê MSSV: 1513079**

**Bài tập tuần 3**

**Câu 1:**

module cau1(SW, LEDR);

input [2:0]SW;

output [1:0]LEDR;

RS\_FF (SW[0], SW[1], SW[2], LEDR[0], LEDR[1]);

endmodule

module RS\_FF(R, S, clk, Qa, Qb);

input R, S, clk;

output wire Qa, Qb;

wire R\_g, S\_g;

and(R\_g, R, clk);

and(S\_g, S, clk);

nor(Qa, R\_g, Qb);

nor(Qb, S\_g, Qa);

endmodule

**Câu 2:**

module cau2(SW,LEDR);

input [1:0]SW;

output [1:0]LEDR;

D\_LATCH (LEDR[0], LEDR[1], SW[0],SW[1]);

// led0-Qa..led1-Qb..sw0-clk..sw1..D

endmodule

module D\_LATCH(Qa, Qb, clk, D);

input D, clk;

output wire Qa, Qb;

wire R\_g, S\_g;

nand (S\_g, D, clk);

nand (R\_g, ~D, clk);

nand (Qa, S\_g, Qb);

nand (Qb, R\_g, Qa);

endmodule

**Câu 3:**

module cau3(SW, LEDR);

input [1:0]SW;

output [0:0]LEDR;

wire Qm;

// sw1-clk..sw0-D..led0-Q

FF\_D(Qm, SW[0], ~SW[1]);

FF\_D(LEDR[0], Qm, SW[1]);

endmodule

module FF\_D(Q, D, Clk);

input Clk, D;

output reg Q;

always@(posedge Clk)

begin

if(D == 1'b1)

Q <= 1'b1;

else

Q <= 1'b0;

end

endmodule

**Câu 4:**

module cau4(SW, LEDR);

input [1:0]SW;

output [2:0]LEDR;

FF\_D\_Ex(SW[1], SW[0], LEDR[0], LEDR[1], LEDR[2]);

endmodule

module FF\_D\_Ex(d, clk, q1, q2, q3);

input d, clk;

output reg q1, q2, q3;

always @(clk)

if(clk)

q1 <= d;

always @(posedge clk)

q2 <= d;

always @(negedge clk)

q3 <= d;

endmodule

**Câu 5:**

module cau5(SW,KEY,HEX0,HEX1,HEX2,HEX3);

input [7:0]SW;

input [1:0]KEY;

output [6:0]HEX0,HEX1,HEX2,HEX3;

wire [7:0]Q;

Decoder\_HEX (SW[3:0],HEX3);

Decoder\_HEX (SW[7:4],HEX2);

D\_LATCH (SW[3:0],KEY[0],KEY[1],Q[3:0]);

Decoder\_HEX(Q[3:0],HEX1);

D\_LATCH (SW[7:4],KEY[0],KEY[1],Q[7:4]);

Decoder\_HEX(Q[7:4],HEX0);

endmodule

module D\_LATCH(D, clk, Reset, Q);

input D, clk, Reset;

output reg Q;

always@(posedge clk, negedge Reset)

if(~Reset)

Q <= 0;

else

Q <= D;

endmodule

module Decoder\_HEX(c, hex);

input [3:0]c;

output reg [6:0]hex;

always @(c)

case (c) //abcdefg

0: hex = 7'b0000001;

1: hex = 7'b1001111;

2: hex = 7'b0010010;

3: hex = 7'b0000110;

4: hex = 7'b1001100;

5: hex = 7'b0100100;

6: hex = 7'b0100000;

7: hex= 7'b0001111;

8: hex= 7'b0000000;

9: hex= 7'b0000100;

default: hex = 7'b1111111;

endcase

endmodule