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

**Bài tập tuần 1**

**Bài 1:**

module cau1(SW, LEDR);

input [9:0]SW;

output [9:0]LEDR;

assign LEDR = SW;

endmodule

**Bài 2:**

module cau2(SW,LEDR);

input [9:0]SW;

output [3:0]LEDR;

mux\_2\_1(SW[3:0],SW[7:4], SW[9], LEDR[3:0]);

endmodule

module mux\_2\_1(x, y, s, m);

input [3:0]x,y;

input s;

output [3:0]m;

assign m[0] = (x[0] & ~s | y[0] & s);

assign m[1] = (x[1] & ~s | y[1] & s);

assign m[2] = (x[2] & ~s | y[2] & s);

assign m[3] = (x[3] & ~s | y[3] & s);

endmodule

**Bài 3:**

module cau3(SW,LEDR);

input [7:0]SW;

output [0:0]LEDR;

mux\_2\_1(SW[4:0],SW[7:5], LEDR[0]);

endmodule

module mux\_2\_1(x, s, m\_out);

input [4:0]x;//

input [2:0]s;

output m\_out;

wire[3:1]m;

assign m[1] = ( x[2] & ~s[0] | x[3] & s[0] );

assign m[2] = ( x[0] & ~s[0] | x[1] & s[0] );

assign m[3] = ( m[2] & ~s[1] | m[1] & s[1] );

assign m\_out = ( m[3] & ~s[2] | x[4] & s[2] );

endmodule

**Bài 4:**

module cau4(SW, HEX0);

input [2:0]SW;

output [0:6]HEX0;

dec3bit dec(SW[2:0],HEX0);

endmodule

module dec3bit(c,leds);

input [2:0]c;

output reg [0:6]leds;

always @(c)

case (c) //abcdefg //0123456

0: leds = 7'b1001000;//H

1: leds = 7'b0110000;//E

2: leds = 7'b1110001;//L

3: leds = 7'b0000001;//O

default: leds = 7'bx;

endcase

endmodule

**Bài 5:**

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

input [9:7] SW;

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

DECODER\_HEX0 (SW[9:7],HEX0);

DECODER\_HEX1 (SW[9:7],HEX1);

DECODER\_HEX2 (SW[9:7],HEX2);

DECODER\_HEX3 (SW[9:7],HEX3);

endmodule

module DECODER\_HEX0(c,led0);

input [9:7]c;

output reg [0:6]led0;

always @(c)

begin

case (c)

3'b 000 : led0 = 7'b0001000; //A

default : led0 = 7'b1111111;

endcase

end

endmodule

module DECODER\_HEX1(c,led1);

input [9:7]c;

output reg [0:6]led1;

always @(c)

begin

case(c)

3'b 000 : led1 = 7'b0100001;//G

default : led1 = 7'b1111111;

endcase

end

endmodule

module DECODER\_HEX2(c,led2);

input [9:7]c;

output reg [0:6]led2;

always @(c)

begin

case(c)

3'b 000 : led2 = 7'b0011000; //P

default : led2 = 7'b1111111;

endcase

end

endmodule

module DECODER\_HEX3(c,led3);

input [9:7]c;

output reg [0:6]led3;

always @(c)

begin

case(c)

3'b 000 : led3 = 7'b0111000; //F

default : led3 = 7'b1111111;

endcase

end

endmodule

**Bài 6:**

module cau6(SW, HEX0, HEX1, HEX2, HEX3);

input [9:7] SW;

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

DECODER\_HEX0 (SW[9:7],HEX0);

DECODER\_HEX1 (SW[9:7],HEX1);

DECODER\_HEX2 (SW[9:7],HEX2);

DECODER\_HEX3 (SW[9:7],HEX3);

endmodule

module DECODER\_HEX0 (c,led0);

input [9:7]c;

output reg [0:6]led0;

always @(c)

begin

case (c)

3'b 000 : led0 = 7'b0001000; //A

3'b 011 : led0 = 7'b0100001;//G

3'b 010 : led0 = 7'b0011000; //P

3'b 001 : led0 = 7'b0111000; //F

default : led0 = 7'b1111111;

endcase

end

endmodule

module DECODER\_HEX1 (c,led1);

input [9:7]c;

output reg [0:6]led1;

always @(c)

begin

case (c)

3'b 001 : led1 = 7'b0001000; //A

3'b 000 : led1 = 7'b0100001;//G

3'b 011 : led1 = 7'b0011000; //P

3'b 010 : led1 = 7'b0111000; //F

default : led1 = 7'b1111111;

endcase

end

endmodule

module DECODER\_HEX2 (c,led2);

input [9:7]c;

output reg [0:6]led2;

always @(c)

begin

case (c)

3'b 010 : led2 = 7'b0001000;//A

3'b 001 : led2 = 7'b0100001;//G

3'b 000 : led2 = 7'b0011000;//P

3'b 011 : led2 = 7'b0111000;//F

default : led2 = 7'b1111111;

endcase

end

endmodule

module DECODER\_HEX3 (c,led3);

input [9:7]c;

output reg [0:6]led3;

always @(c)

begin

case (c)

3'b 011 : led3 = 7'b0001000;//A

3'b 010 : led3 = 7'b0100001;//G

3'b 001 : led3 = 7'b0011000;//P

3'b 000 : led3 = 7'b0111000;//F

default : led3 = 7'b1111111;

endcase

end

endmodule

**Bài 7:**

module cau7(SW, HEX0);

input [3:0] SW;

output [0:6]HEX0;

DECODER(SW[3:0],HEX0);

endmodule

module DECODER(c,led0);

input [3:0]c;

output reg [0:6]led0;

always @(c)

begin

case (c)

4'b 0000 : led0 = 7'b0000000;

4'b 0001 : led0 = 7'b1001111;

4'b 0010 : led0 = 7'b0010010;

4'b 0011 : led0 = 7'b0000110;

4'b 0100 : led0 = 7'b1001100;

4'b 0101 : led0 = 7'b0100100;

4'b 0110 : led0 = 7'b0100000;

4'b 0111 : led0 = 7'b0001111;

4'b 1000 : led0 = 7'b0000000;

4'b 1001 : led0 = 7'b0000100;

default : led0 = 7'b1111111;

endcase

end

endmodule