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

**Bài tập tuần 7**

**Bài 1:**

*module* cau1(SW, HEX0, HEX1, CLOCK\_50, KEY);

    input [1:0]SW;

    input [0:0]KEY;

    input CLOCK\_50;

    output [6:0]HEX0, HEX1;

    wire [3:0]a, b;

    counter(CLOCK\_50, KEY[0], a, b, SW[1], SW[0]);

    display\_0\_9(a, HEX0);

    display\_0\_9(b, HEX1);

*endmodule*

*module* counter(clock, reset, BCD0, BCD1, enable, even\_odd);

    input reset,clock, enable, even\_odd;

    output reg[3:0]BCD0, BCD1;

    reg [25:0]count;

    always @(posedge clock)

        begin

            if(count==49999999)

                count<=0;

            else

            count=count+1;

        end

    always@(posedge clock)

        if(!reset)

            begin

                BCD0<=0;

                BCD1<=0;

            end

        else

            if(enable)

            begin

                if(even\_odd == 0) // EVEN

                    begin

                        if (count == 0)

                        begin

                            if(BCD0 % 2 == 0)

                                if(BCD0 == 8)

                                    begin

                                        BCD0 <= 0;

                                        if(BCD1 == 9)

                                            BCD1 <= 0;

                                        else

                                            BCD1 <= BCD1 + 1;

                                    end

                                else

                                    BCD0 <= BCD0 + 2;

                            else

                                BCD0 <= BCD0 + 1;

                        end

                    end

                else // ODD

                    begin

                        if (count == 0)

                        begin

                            if(BCD0 % 2 == 1)

                                if(BCD0 == 9)

                                    begin

                                        BCD0 <= 1;

                                        if(BCD1 == 9)

                                            BCD1 <= 0;

                                        else

                                            BCD1 <= BCD1 + 1;

                                    end

                                else

                                    BCD0 <= BCD0 + 2;

                            else

                                BCD0 <= BCD0 + 1;

                        end

                    end

            end

*endmodule*

*module* display\_0\_9(c,hex);

    input [3:0]c;

    output reg [6:0]hex;

    always@(c)

        case(c)

            4'b0000 :hex =7'b1000000;

            4'b0001 :hex =7'b1111001;

            4'b0010 :hex =7'b0100100;

            4'b0011 :hex =7'b0110000;

            4'b0100 :hex =7'b0011001;

            4'b0101 :hex =7'b0010010;

            4'b0110 :hex =7'b0000010;

            4'b0111 :hex =7'b1111000;

            4'b1000 :hex =7'b0000000;

            4'b1001 :hex =7'b0010000;

            default :hex =7'b1111111;

        endcase

*endmodule*

**Bài 2:**

*module* cau2(SW, HEX0, HEX1, CLOCK\_50, KEY);

    input [1:0]SW;

    input [0:0]KEY;

    input CLOCK\_50;

    output [6:0]HEX0, HEX1;

    wire [3:0]a, b;

    counter(CLOCK\_50, KEY[0], a, b, SW[1], SW[0]);

    display\_0\_9(a, HEX0);

    display\_0\_9(b, HEX1);

*endmodule*

*module* counter(clock, reset, BCD0, BCD1, enable, even\_odd);

    input reset,clock, enable, even\_odd;

    output reg[3:0]BCD0, BCD1;

    reg [25:0]count;

    always @(posedge clock)

        begin

            if(count==49999999)

                count<=0;

            else

            count=count+1;

        end

    always@(posedge clock)

        if(!reset)

            begin

                BCD0 <= 0;

                BCD1 <= 0;

            end

        else

            if(enable)

            begin

                if(even\_odd == 0) // EVEN

                    begin

                        if (count == 0)

                        begin

                            if(BCD0 % 2 == 0)

                                if(BCD0 == 0)

                                    begin

                                        BCD0 <= 8;

                                        if(BCD1 == 0)

                                            BCD1 <= 9;

                                        else

                                            BCD1 <= BCD1 - 1;

                                    end

                                else

                                    BCD0 <= BCD0 - 2;

                            else

                                BCD0 <= BCD0 - 1;

                        end

                    end

                else // ODD

                    begin

                        if (count == 0)

                        begin

                            if(BCD0 % 2 == 1)

                                if(BCD0 == 1)

                                    begin

                                        BCD0 <= 9;

                                        if(BCD1 == 0)

                                            BCD1 <= 9;

                                        else

                                            BCD1 <= BCD1 - 1;

                                    end

                                else

                                    BCD0 <= BCD0 - 2;

                            else

                                BCD0 <= BCD0 - 1;

                        end

                    end

            end

*endmodule*

*module* display\_0\_9(c,hex);

    input [3:0]c;

    output reg [6:0]hex;

    always@(c)

        case(c)

            4'b0000 :hex =7'b1000000;

            4'b0001 :hex =7'b1111001;

            4'b0010 :hex =7'b0100100;

            4'b0011 :hex =7'b0110000;

            4'b0100 :hex =7'b0011001;

            4'b0101 :hex =7'b0010010;

            4'b0110 :hex =7'b0000010;

            4'b0111 :hex =7'b1111000;

            4'b1000 :hex =7'b0000000;

            4'b1001 :hex =7'b0010000;

            default :hex =7'b1111111;

        endcase

*endmodule*

**Bài 3:**

*module* cau3(LEDR,KEY,CLOCK\_50,HEX0,HEX1,HEX2,HEX3);

    input [0:0]CLOCK\_50,KEY;

    output [7:0]LEDR;

    output [6:0]HEX0,HEX1,HEX2,HEX3;

    wire [3:0]bcd0, bcd1, bcd2, bcd3, led0, led1;

    Counter(CLOCK\_50,KEY[0],bcd0,bcd1,bcd2,bcd3,led0);

    decoder a (bcd0,HEX0);

    decoder b (bcd1,HEX1);

    decoder c (bcd2,HEX2);

    decoder d (bcd3,HEX3);

    assign LEDR[7:0] = led0;

*endmodule*

*module* Counter(clock,reset,bcd0,bcd1,bcd2,bcd3,led0);

    input clock,reset;

    output reg [3:0]bcd0,bcd1,bcd2,bcd3;

    output reg [7:0]led0;

    reg [25:0]count;

    always @(posedge clock)

            begin

                if (count==99999)

                    count <=0;

                else

                    count <= count + 1;

            end

    always@(posedge clock)

            begin

                    if (!reset)

                    begin

                        bcd0<=0;

                        bcd1<=0;

                        bcd2<=0;

                        bcd3<=0;

                        led0<=0;

                    end

                    else

                        if (count == 0)

                        begin

                            if (bcd0 == 9)

                            begin

                                bcd0 <= 0;

                                if (bcd1 == 5)

                                    begin

                                    bcd1 <= 0;

                                        if (bcd2 == 9)

                                        begin

                                        bcd2 <= 0;

                                            if (bcd3 == 5)

                                                begin

                                                bcd3 <= 0;

                                                if (led0 == 23)

                                                    begin

                                                        led0 <= 0;

                                                        bcd0 <= 0;

                                                        bcd1 <= 0;

                                                        bcd2 <= 0;

                                                        bcd3 <= 0;

                                                    end

                                                else

                                                    begin

                                                        if (led0 == 23)

                                                            led0 <= 0;

                                                        else led0 <= led0 + 1;

                                                    end

                                                end

                                            else bcd3 <= bcd3 + 1;

                                            end

                                            else bcd2 <= bcd2 + 1;

                                                end

                                            else bcd1 <= bcd1 + 1;

                                    end

                                else bcd0 <= bcd0 + 1;

                            end

            end

*endmodule*

*module* decoder(c,hex);

    input [3:0]c;

    output reg [6:0]hex;

    always@(c)

    case (c)

        4'b 0000: hex = 7'b 1000000;

        4'b 0001: hex = 7'b 1111001;

        4'b 0010: hex = 7'b 0100100;

        4'b 0011: hex = 7'b 0110000;

        4'b 0100: hex = 7'b 0011001;

        4'b 0101: hex = 7'b 0010010;

        4'b 0110: hex = 7'b 0000010;

        4'b 0111: hex = 7'b 1111000;

        4'b 1000: hex = 7'b 0000000;

        4'b 1001: hex = 7'b 0010000;

    endcase

*endmodule*

**Bài 4:**

*module* cau4(CLOCK\_50, KEY, SW, HEX0, HEX1, LEDR);

    input CLOCK\_50;

    input [0:0]KEY;

    input [8:0]SW;

    output [6:0]HEX0, HEX1;

    output [9:0]LEDR;

    wire [3:0]a,b;

    counter (CLOCK\_50, KEY[0],SW[8], a, b, SW[3:0], SW[7:4], LEDR);

    display\_0\_9 (a, HEX0);

    display\_0\_9 (b, HEX1);

*endmodule*

*module* counter(CLOCK\_50, reset, enable, BCD0, BCD1, var0, var1, led);

    input reset,CLOCK\_50,enable;

    input [3:0]var0, var1;

    output reg[3:0]BCD0, BCD1;

    output reg[9:0]led;

    reg [25:0]delay;

    //tao delay 1s

    always @(posedge CLOCK\_50)

        begin

            if(delay==49999999)

                delay<=0;

            else

            delay=delay+1;

        end

    always@(posedge CLOCK\_50 or negedge reset)

        if(!reset)

            begin

                BCD0 <= 0;

                BCD1 <= 0;

            end

        else

        if (enable == 1)

        begin

            if(delay == 0)

                begin

                    if(BCD0 == 0)

                        begin

                            BCD0 <= 9;

                            if(BCD1 == 0)

                                begin

                                BCD1 <= 9;

                                end

                            else

                                BCD1 <= BCD1 - 1;

                        end

                    else

                        BCD0 <= BCD0 - 1;

                        if ((BCD0 == var0) && (BCD1 == var1))

                        begin

                            led = 10'b1111111111;

                        end

                end

        end

*endmodule*

*module* display\_0\_9(c,hex);

    input [3:0]c;

    output reg [6:0]hex;

    always@(c)

        case(c)

            4'b0000 :hex =7'b1000000;

            4'b0001 :hex =7'b1111001;

            4'b0010 :hex =7'b0100100;

            4'b0011 :hex =7'b0110000;

            4'b0100 :hex =7'b0011001;

            4'b0101 :hex =7'b0010010;

            4'b0110 :hex =7'b0000010;

            4'b0111 :hex =7'b1111000;

            4'b1000 :hex =7'b0000000;

            4'b1001 :hex =7'b0010000;

            default :hex =7'b1111111;

        endcase

*endmodule*

**Bài 5:**

*module* cau5(CLOCK\_50,KEY,HEX0,HEX1,LEDR,LEDG);

    input CLOCK\_50;

    input [0:0]KEY;

    output [0:6]HEX0,HEX1;

    output [0:0]LEDR,LEDG;

    wire [3:0]giay0,giay1;

    count (CLOCK\_50,KEY[0],giay0,giay1,LEDG[0],LEDR[0]);

    decoder\_BCD dv (giay0,HEX0);

    decoder\_BCD ch (giay1,HEX1);

*endmodule*

*module* count (CLK,CLR,giay0,giay1,green,red);

    input CLK,CLR;

    output reg [3:0]giay0,giay1;

    output reg green,red;

    reg [26:0]count;

    always @ (posedge CLK)

        if (count==10000000) count <= 0;

        else count <= count+1;

    always @ (posedge CLK)

    begin

        if (!CLR)

            begin giay0 <=0;

                 giay1 <=0;

                 green <=1;

                 red <=0;

            end

        else if (count==0)

                    if (giay0==0)

                        begin

                            giay0 <=9;

                            if (giay1==0)

                                begin

                                    giay1 <=2;

                                    green <=~green;

                                    red <=~red;

                                end

                            else

                                giay1 <=giay1-1;

                        end

                    else giay0 <=giay0-1;

    end

*endmodule*

*module* decoder\_BCD (i,led);

    input [3:0]i;

    output reg [6:0]led;

    always @(i)

    case(i)

    0:led=7'b0000001;

    1:led=7'b1001111;

    2:led=7'b0010010;

    3:led=7'b0000110;

    4:led=7'b1001100;

    5:led=7'b0100100;

    6:led=7'b0100000;

    7:led=7'b0001111;

    8:led=7'b0000000;

    9:led=7'b0000100;

    10:led=7'b0001000;

    11:led=7'b1100000;

    12:led=7'b0110001;

    13:led=7'b1000010;

    14:led=7'b0110000;

    15:led=7'b0111000;

    endcase

*endmodule*