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

**Bài tập tuần 6**

**Bài 1:**

*module* cau2(SW, KEY, HEX0);

    input [0:0]SW; // T

    input [1:0]KEY; // clk = key0, rst = key1

    output [6:0]HEX0;

    wire [3:0]q, t;

    FF\_T(SW[0], KEY[1], KEY[0], q[0]);

    and(t[1], SW[0], q[0]);

    FF\_T(t[1], KEY[1], KEY[0], q[1]);

    and(t[2], t[1], q[1]);

    FF\_T(t[2], KEY[1], KEY[0], q[2]);

    and(t[3], t[2], q[2]);

    FF\_T(t[3], KEY[1], KEY[0], q[3]);

    Decoder\_HEX(q, HEX0);

*endmodule*

*module* FF\_T(t, rst, clk, q);

    input t, rst, clk;

    output reg q;

    always@(posedge clk or negedge rst)

        if(!rst)

            q <= 1'b0;

        else

            if(t)

                q <= ~q;

*endmodule*

*module* Decoder\_HEX(c,HEX);

    input[3:0]c;

    output [6:0]HEX;

    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;

        4'b1010: HEX = 7'b0001000; //A

        4'b1011: HEX = 7'b0000011; //B

        4'b1100: HEX = 7'b1000110; //C

        4'b1101: HEX = 7'b0100001; //D

        4'b1110: HEX = 7'b0000110; //E

        4'b1111: HEX = 7'b0001110; //F

        default: HEX = 7'b1111111;

    endcase

*endmodule*

**Bài 2:**

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

    input clock\_50;

    input [0:0]KEY,SW;

    output [6:0]HEX0;

    wire [3:0]a;

    counter(SW[0],clock\_50,KEY[0],a);

    seg70(a,HEX0);

*endmodule*

*module* counter(enable,clock,reset,count);

    input reset,clock,enable;

    output reg[3:0]count;

    reg [25:0]delay;

    always @(posedge clock)

    begin

    if (delay==49999999)

        delay<=0;

        else

        delay<=delay+1;

    end

    always@(posedge clock , negedge reset)

    if(reset==0)

    count<=0;

    else

    if (enable)

        if (delay==0)

            if(count==9)

                count<=0;

    else

        count<=count+1;

*endmodule*

*module* seg70 (bcd, HEX);

    input [3:0] bcd;

    output reg [6:0]HEX;

    always @(bcd)

    case (bcd) //abcdefg

            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;

            4'b1010: HEX = 7'b0001000; //A

            4'b1011: HEX = 7'b0000011; //B

            4'b1100: HEX = 7'b1000110; //C

            4'b1101: HEX = 7'b0100001; //D

            4'b1110: HEX = 7'b0000110; //E

            4'b1111: HEX = 7'b0001110; //F

            default: HEX = 7'b1111111;

    endcase

*endmodule*

**Bài 3:**

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

    input [1:0]SW;

    input [0:0]KEY;

    input CLOCK\_50;

    output [6:0]HEX0,HEX1,HEX2,HEX3;

    wire [2:0]dem;

    //assign dem[0] = 0;

    count\_and\_delay (KEY[0], SW[0], CLOCK\_50, dem);

    decoder\_FPGA (dem, HEX0);

    decoder\_FPGA (dem - 1, HEX1);

    decoder\_FPGA (dem - 2, HEX2);

    decoder\_FPGA (dem - 3, HEX3);

*endmodule*

*module* count\_and\_delay (reset, enable, clock, dem);

    input reset,enable,clock;

    output reg[2:0]dem;

    reg [25:0]delay;

    always @(posedge clock)

    begin

    if (delay==49999999)

        delay <= 0;

    else

        delay <= delay +1;

    end

    always@(posedge clock)

    if(!reset)

        dem = 0;

    else

        if(enable)

            begin

                if(delay == 0)

                    dem = dem + 1;

            end

*endmodule*

*module* decoder\_FPGA(c,hex);

    input [2:0]c;

    output reg[6:0]hex;

    always@(c)

    case (c)

    3'b 000: hex = 7'b 1111111; //led off

    3'b 001: hex = 7'b 0001110; //F

    3'b 010: hex = 7'b 0001100; //P

    3'b 011: hex = 7'b 0000010; //G

    3'b 100: hex = 7'b 0001000; //A

    default : hex= 7'b1111111;//led off;

    endcase

*endmodule*

**Bài 4:**

*module* cau4(SW,KEY,CLOCK\_50,HEX0,HEX1,HEX2,HEX3);

    input [1:0]SW;

    input [0:0]KEY;

    input CLOCK\_50;

    output [6:0]HEX0,HEX1,HEX2,HEX3;

    wire [2:0]dem;

    count\_and\_delay (KEY[0],SW[0],SW[1],CLOCK\_50,dem);

    decoder\_FPGA (dem,HEX0);

    decoder\_FPGA (dem-1,HEX1);

    decoder\_FPGA (dem-2,HEX2);

    decoder\_FPGA (dem-3,HEX3);

*endmodule*

*module* count\_and\_delay (reset,enable,select,clock,dem);

    input reset,enable,clock,select;

    output reg[2:0]dem;

    reg [25:0]delay;

    always @(posedge clock)

    begin

    if (delay==49999999)

        delay <= 0;

    else

        delay <= delay +1;

    end

    always@(posedge clock)

    if(!reset)

        dem = 0;

    else

        if(enable)

            begin

                if(delay == 0)

                    dem = dem + (select?1:-1);

            end

*endmodule*

*module* decoder\_FPGA(c,hex);

    input [2:0]c;

    output reg[6:0]hex;

    always@(c)

    case (c)

    3'b 000: hex = 7'b 1111111; //led off

    3'b 001: hex = 7'b 0001110; //F

    3'b 010: hex = 7'b 0001100; //P

    3'b 011: hex = 7'b 0000010; //G

    3'b 100: hex = 7'b 0001000; //A

    default : hex= 7'b1111111;//led off;

    endcase

*endmodule*

**Bài 5:**

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

    input [0:0]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[0]);

    display\_0\_9(a, HEX0);

    display\_0\_9(b, HEX1);

*endmodule*

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

    input reset,clock, enable;

    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(count==0)

                    begin

                        if(BCD0==9)

                            begin

                                if(BCD1==9)

                                    begin

                                        BCD1<=0;

                                        BCD0<=0;

                                    end

                                else

                                    begin

                                        BCD1<=BCD1+1;

                                        BCD0<=0;

                                    end

                            end

                        else

                            BCD0<=BCD0+1;

                    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;

            //4'b 1010: hex = 7'b 0100000;

            //4'b 1011: hex = 7'b 0000011;

            //4'b 1100: hex = 7'b 1000110;

            //4'b 1101: hex = 7'b 0100001;

            //4'b 1110: hex = 7'b 0000110;

            //4'b 1111: hex = 7'b 0001110;

            default :hex =7'b1111111;

        endcase

*endmodule*

**Bài 6:**

*module* cau6(CLOCK\_50,KEY,SW,HEX0, HEX1);

    input CLOCK\_50;

    input [0:0]KEY;

    input [0:0]SW;

    output [6:0]HEX0, HEX1;

    wire [3:0]a,b;

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

        display\_0\_9 (a, HEX0);

        display\_0\_9 (b, HEX1);

*endmodule*

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

    input reset,CLOCK\_50,enable;

    output reg[3:0]BCD0, BCD1;

    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==0)

        begin

            if(delay==0)

                    begin

                        if(BCD0==0)

                            begin

                                BCD0<=9;

                                    if(BCD1==0)

                                        BCD1<=9;

                                    else

                                        BCD1<=BCD1-1;

                            end

                        else

                            BCD0<=BCD0-1;

                    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;

            //4'b 1010: hex = 7'b 0100000;

            //4'b 1011: hex = 7'b 0000011;

            //4'b 1100: hex = 7'b 1000110;

            //4'b 1101: hex = 7'b 0100001;

            //4'b 1110: hex = 7'b 0000110;

            //4'b 1111: hex = 7'b 0001110;

            default :hex =7'b1111111;

        endcase

*endmodule*

**Bài 7:**

*module* cau7(CLOCK\_50,KEY,SW,HEX0, HEX1);

    input CLOCK\_50;

    input [0:0]KEY;

    input [1:0]SW;

    output [6:0]HEX0, HEX1;

    wire [3:0]a,b;

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

        display\_0\_9 (a, HEX0);

        display\_0\_9 (b, HEX1);

*endmodule*

*module* counter(CLOCK\_50,reset,enable,select,BCD0, BCD1);

    input reset,CLOCK\_50,enable,select;

    output reg[3:0]BCD0, BCD1;

    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<=9;

                BCD1<=9;

            end

        else

        if (enable)

        begin

            if(select==0)

            begin

            if(delay==0)

                    begin

                        if(BCD0==0)

                            begin

                                BCD0<=9;

                                    if(BCD1==0)

                                        BCD1<=9;

                                    else

                                        BCD1<=BCD1-1;

                            end

                        else

                            BCD0<=BCD0-1;

                    end

                    end

                    else

                        begin

            if(delay==0)

                    begin

                        if(BCD0==9)

                            begin

                                BCD0<=0;

                                    if(BCD1==9)

                                        BCD1<=0;

                                    else

                                        BCD1<=BCD1+1;

                                end

                        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 8:**

*module* cau8(SW, HEX0, HEX1, HEX2, CLOCK\_50, KEY);

    input [0:0]SW;

    input [0:0]KEY;

    input CLOCK\_50;

    output [6:0]HEX0, HEX1, HEX2;

    wire [3:0]a, b,c;

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

    display\_0\_9(a, HEX0);

    display\_0\_9(b, HEX1);

    display\_0\_9(c, HEX2);

*endmodule*

*module* counter(clock,reset,BCD0, BCD1, BCD2, enable);

    input reset,clock, enable;

    output reg[3:0]BCD0, BCD1,BCD2;

    reg [25:0]count;

    always @(posedge clock)

            begin

                if(count==1000000)

                    count<=0;

                else

                    count=count+1;

            end

    always@(posedge clock)

        if(!reset)

            begin

                BCD0<=0;

                BCD1<=0;

                BCD2<=0;

            end

        else

            if(enable)//mach dem len

            begin

                if(count==0)

                    begin

                        if(BCD0==9)

                            begin

                                if(BCD1==9)

                                    begin

                                        if (BCD2==9)

                                            begin

                                                BCD2<=0;

                                                BCD1<=0;

                                                BCD0<=0;

                                            end

                                        else

                                            begin

                                                BCD2<=BCD2+1;

                                                BCD1<=0;

                                                BCD0<=0;

                                            end

                                    end

                                else

                                    begin

                                        BCD1<=BCD1+1;

                                        BCD0<=0;

                                    end

                            end

                        else

                            BCD0<=BCD0+1;

                    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*