

**Câu 1 : Thiết kế mạch đếm lên/xuống theo mã BCD (0-9) hiển thị ra led 7 đoạn HEX0 với tần số khoảng 1Hz được chia từ clock 50MHz, chân Up/Down được gán cho SW0**

```
module cau1(SW,HEX0,CLOCK_50);
input CLOCK_50;
input SW;
output [0:6] HEX0;
reg [3:0] dem = 4'b0000;
reg [25:0] tanso;
always @ (posedge CLOCK_50)
    tanso <= tanso + 1'b1;
always @ (posedge tanso[25])
    begin
        if (SW==1 && dem==4'd9) dem=4'd0;
            else if (SW==1 && dem != 4'd9) dem = dem + 1'b1;
            else if (SW==0 && dem ==4'd0) dem = 4'd9;
            else dem = dem - 1'b1;
    end
assign HEX0 = (dem==4'd0) ? 7'b0000001:
    (dem==4'd1) ? 7'b1001111:
    (dem==4'd2) ? 7'b0010010:
    (dem==4'd3) ? 7'b0000110:
    (dem==4'd4) ? 7'b1001100:
    (dem==4'd5) ? 7'b0100100:
    (dem==4'd6) ? 7'b0100000:
    (dem==4'd7) ? 7'b0001111:
    (dem==4'd8) ? 7'b0000000:
    (dem==4'd9) ? 7'b0000100: 7'b1111111;
Endmodule
```

**Câu 2 : Thiết kế mạch cộng 2 số 4 bit (sử dụng các switch làm ngõ vào) hiển thị ra led 7 đoạn HEX1, HEX0 dưới dạng số thập phân**

```
module cau2(input [7:0] SW,output reg [0:6] HEX0,HEX1);  
reg [4:0] donvi,chuc;  
always @(*)  
begin  
donvi=(SW[7:4] + SW[3:0]) % 4'd10;  
chuc=(SW[7:4] + SW[3:0]) / 4'd10;  
case(donvi)  
0: HEX0=7'b0000001;  
1: HEX0=7'b1001111;  
2: HEX0=7'b0010010;  
3: HEX0=7'b0000110;  
4: HEX0=7'b1001100;  
5: HEX0=7'b0100100;  
6: HEX0=7'b0100000;  
7: HEX0=7'b0001111;  
8: HEX0=7'b0000000;  
9: HEX0=7'b0000100;  
default: HEX0=7'b1111111;  
endcase  
case(chuc)  
0: HEX1=7'b0000001;  
1: HEX1=7'b1001111;  
2: HEX1=7'b0010010;  
3: HEX1=7'b0000110;  
default: HEX1=7'b1111111;  
endcase end endmodule
```

**Câu 3 : Thiết kế mạch dịch led sử dụng 8 led xanh, led sẽ chạy từ trái qua phải và đổi lại từ phải qua trái với xung clock khoảng 1Hz được chia từ clock 50Mhz.**

```
module cau3(input CLOCK_50, output reg [7:0] LEDG);
```

```
reg flag;
```

```
reg [25:0] tanso;
```

```
always @(posedge CLOCK_50)
```

```
    tanso <= tanso + 1'b1;
```

```
always @(posedge tanso[25])
```

```
    if (LEDG[7:0] == 8'b00000000)
```

```
        LEDG[7:0] = 8'b10000000;
```

```
    else
```

```
        if (flag==0)
```

```
            LEDG[7:0] <= {LEDG[0],LEDG[7:1]};
```

```
        else
```

```
            LEDG[7:0] <= {LEDG[6:0],LEDG[7]};
```

```
always @(posedge tanso[25])
```

```
    if (LEDG[7:0]==8'b01000000)
```

```
        flag = 0;
```

```
    else if (LEDG[7:0]==8'b000000010)
```

```
        flag = 1;
```

```
endmodule
```

**Câu 4 : Thiết kế mạch dịch led sử dụng 8 led đỏ với xung clock khoảng 1Hz được chia từ clock 50Mhz, mỗi lần bấm và thả ngay KEY thì led sẽ đứng yên, bấm và thả trong 3s led sẽ dịch trái, bấm và thả trong 5s led sẽ dịch phải**

```
module cau4(CLOCK_50,KEY,LEDR);
```

```
input CLOCK_50;
```

```
input [0:0]KEY;
```

```

output [7:0] LEDR;
reg [7:0] LEDR;
reg [25:0] tanso;
reg [3:0] flag;
reg [3:0] t;
always @(posedge CLOCK_50)
    tanso <= tanso + 1'b1;
always @(posedge tanso[25])
    if (KEY[0:0] == 1)
        flag = 0;
    else
        flag <= flag + 1'b1;
always @(posedge KEY[0:0])
    t = flag;
always @(posedge tanso[25])
    begin
        if (LEDR==0)
            LEDR <= LEDR + 1'b1;
        else
            if (t<=1)
                LEDR <= LEDR;
            else if (t>=1 & t<=4)
                LEDR = {LEDR[6:0],LEDR[7]};
            else if (t>=4)
                LEDR = {LEDR[0],LEDR[7:1]};
        end
    endmodule

```

**Câu 5 : Thiết kế mạch dịch led sử dụng 8 led xanh với clock khoảng 1Hz được chia từ clock 50Mhz sao cho 2 led chạy từ ngoài vào giữa.**

```
module cau5(CLOCK_50,LEDG);
input CLOCK_50;
output [7:0] LEDG;
reg [7:0] LEDG = 8'b10000001;
reg [25:0] tanso;
always @(posedge CLOCK_50)
    tanso <= tanso + 1'b1;
always @(posedge tanso[25])
    begin
        LEDG[7:4] <= {LEDG[4],LEDG[7:5]};
        LEDG[3:0] <= {LEDG[2:0],LEDG[3]};
    end
endmodule
```

**Câu 6: Thiết kế mạch hiển thị chữ “HELLO” lên 5 led 7 đoạn với 2 ngõ vào được gán với switch, sao cho nếu switch = 2'b00 thì chữ “HELLO” nhấp nháy với tần số khoảng 1 Hz, switch = 2'b01 thì khoảng 2 Hz, switch = 2'b10 thì khoảng 4 Hz và switch = 2'b11 thì khoảng 8 Hz. Clock được lấy từ xung clock 50MHz.**

```
module cau6(SW,CLOCK_50,HEX0,HEX1,HEX2,HEX3,HEX4);
input CLOCK_50;
input [1:0] SW;
output [0:6] HEX0,HEX1,HEX2,HEX3,HEX4;
reg [25:0] tanso;
wire clock;
```

```

always @(posedge CLOCK_50)
    tanso <= tanso + 1'b1;
assign clock = (SW==2'b00) ? tanso[25] :
    (SW==2'b01) ? tanso[24] :
    (SW==2'b10) ? tanso[23] : tanso[22];
assign HEX0 = (clock) ? 7'b00000001 : 7'b11111111;
assign HEX1 = (clock) ? 7'b11100001 : 7'b11111111;
assign HEX2 = (clock) ? 7'b11100001 : 7'b11111111;
assign HEX3 = (clock) ? 7'b01100000 : 7'b11111111;
assign HEX4 = (clock) ? 7'b10010000 : 7'b11111111;
endmodule

```

**Câu 7 : Thiết kế mạch đổi từ số nhị phân 9 bit (lấy từ switch) thành số thập phân hiển thị lên HEX2, HEX1 và HEX0.**

```

module cau7( input [8:0] SW, output reg [0:6] HEX0, HEX1,HEX2);
reg [3:0] tram,chuc,donvi;
always @(*)
    begin
        donvi = SW[8:0] % 4'd10;
        chuc = (SW[8:0] /4'd10) % 4'd10;
        tram = (SW[8:0] / 4'd10) / 4'd10;
        case(donvi)
            0: HEX0 = 7'b00000001;
            1: HEX0 = 7'b10011111;
            2: HEX0 = 7'b00100010;
            3: HEX0 = 7'b00000110;
            4: HEX0 = 7'b1001100;

```

```
5: HEX0 = 7'b0100100;  
6: HEX0 = 7'b0100000;  
7: HEX0 = 7'b0001111;  
8: HEX0 = 7'b0000000;  
9: HEX0 = 7'b0000100;  
default: HEX0 = 7'b1111111;
```

endcase

case(chuc)

```
0: HEX1 = 7'b0000001;  
1: HEX1 = 7'b1001111;  
2: HEX1 = 7'b0010010;  
3: HEX1 = 7'b0000110;  
4: HEX1 = 7'b1001100;  
5: HEX1 = 7'b0100100;  
6: HEX1 = 7'b0100000;  
7: HEX1 = 7'b0001111;  
8: HEX1 = 7'b0000000;  
9: HEX1 = 7'b0000100;  
default: HEX1 = 7'b1111111;
```

endcase

case(tram)

```
0: HEX2 = 7'b0000001;  
1: HEX2 = 7'b1001111;  
2: HEX2 = 7'b0010010;  
3: HEX2 = 7'b0000110;  
4: HEX2 = 7'b1001100;  
5: HEX2 = 7'b0100100;  
6: HEX2 = 7'b0100000;
```

```

        7: HEX2 = 7'b00011111;
        8: HEX2 = 7'b00000000;
        9: HEX2 = 7'b00000100;
        default: HEX2 = 7'b11111111;
    endcase

end

endmodule

```

**Câu 8 : Thiết kế mạch hiển thị chữ “HI” dịch từ trái qua phải ở 3 led 7 đoạn (từ HEX0 đến HEX2), xung clock khoảng 1Hz được chia từ clock 50Mhz.**

```

module cau8(CLOCK_50,HEX0,HEX1,HEX2);
input CLOCK_50;
output [0:6] HEX0,HEX1,HEX2;
reg [0:6] HEX0,HEX1,HEX2;
reg [25:0] tanso;
reg [1:0] q;
always @(posedge CLOCK_50)
    tanso <= tanso + 1'b1;
always @(posedge tanso[25])
begin
    q <= q + 1'b1;
    if (q == 2)
        begin
            q <= 0;
        end
    case(q)
        0: begin

```



```

        HEX2 = 7'b1001000;
        HEX1 = 7'b1001111;
        HEX0 = 7'b1111111;
    end
1: begin
    HEX2 = 7'b1111111;
    HEX1 = 7'b1001000;
    HEX0 = 7'b1001111;
    end
2: begin
    HEX2 = 7'b1001111;
    HEX1 = 7'b1111111;
    HEX0 = 7'b1001000;
    end

endcase
end
endmodule

```

**Câu 9 : Thiết kế mạch led 8 bit sáng dần từ trái qua phải và tắt dần từ phải qua trái.**

```

module cau9(CLOCK_50,LEDG);
input CLOCK_50;
output [7:0] LEDG;
reg [7:0] LEDG;
reg [25:0] tanso;
reg flag;
always @(posedge CLOCK_50)

```

```

tanso <= tanso + 1'b1;
always @(posedge tanso[25])
    if (flag == 0)
        LEDG <= {1'b1, LEDG[7:1]};
    else
        LEDG <= {LEDG[6:0],1'b0};
always @(posedge tanso[25])
    if (LEDG == 8'b10000000 )
        flag <= 0;
    else
        if (LEDG == 8'b11111110)
            flag = 1;
endmodule

```

**Câu 10 : Thiết kế mạch bình phương 4 bit (sử dụng các switch làm ngõ vào) hiển thị ra led 7 đoạn HEX2, HEX1, HEX0 dưới dạng số thập phân.**

```

module cau10(SW,HEX0,HEX1,HEX2);
input [3:0] SW;
output [0:6] HEX0, HEX1, HEX2;
reg [0:6] HEX0, HEX1, HEX2;
reg [7:0] tram,chuc,donvi;
always @(*)
    begin
        tram = ((SW[3:0] * SW[3:0]) / 4'd10) / 4'd10;
        chuc = ((SW[3:0] * SW[3:0]) / 4'd10) % 4'd10;
        donvi = (SW[3:0] * SW[3:0]) % 4'd10;
        case(donvi)

```

```
0: HEX0 = 7'b00000001;  
1: HEX0 = 7'b10011111;  
2: HEX0 = 7'b0010010;  
3: HEX0 = 7'b00001110;  
4: HEX0 = 7'b10011100;  
5: HEX0 = 7'b0100100;  
6: HEX0 = 7'b0100000;  
7: HEX0 = 7'b00011111;  
8: HEX0 = 7'b00000000;  
9: HEX0 = 7'b0000100;  
default: HEX0 = 7'b11111111;
```

endcase

case(chuc)

```
0: HEX1 = 7'b00000001;  
1: HEX1 = 7'b10011111;  
2: HEX1 = 7'b0010010;  
3: HEX1 = 7'b00001110;  
4: HEX1 = 7'b10011100;  
5: HEX1 = 7'b0100100;  
6: HEX1 = 7'b0100000;  
7: HEX1 = 7'b00011111;  
8: HEX1 = 7'b00000000;  
9: HEX1 = 7'b0000100;  
default: HEX1 = 7'b11111111;
```

endcase

case(tram)

```
0: HEX2 = 7'b00000001;  
1: HEX2 = 7'b10011111;
```

```
2: HEX2 = 7'b0010010;  
3: HEX2 = 7'b0000110;  
4: HEX2 = 7'b1001100;  
5: HEX2 = 7'b0100100;  
6: HEX2 = 7'b0100000;  
7: HEX2 = 7'b0001111;  
8: HEX2 = 7'b0000000;  
9: HEX2 = 7'b0000100;  
    default: HEX2 = 7'b1111111;  
endcase  
end  
endmodule
```