



دانشگاه صنعتی شریف

دانشکده‌ی مهندسی برق

آزمایشگاه مدارهای منطقی و سیستم های دیجیتال

گزارش آزمایش جلسه ۴

سیدبردیا برائی نژاد (۹۲۱۰۱۶۶۹)

مهدی میر (۹۲۱۰۲۸۴۶)

استاد: دکتر تابنده

۱. برای قسمت اول آزمایش کد زیر را اول میزنیم:

```
module com(num,a,b,c,d,e,f,g);
    //num is the input number
    //a,...,g are names of 7seg LED's
    input [3:0] num;
    output reg a,b,c,d,e,f,g;
    always @(*)
    begin
        case(num)
            0: {a,b,c,d,e,f,g} = 7'b1111110; //g is off
            1: {a,b,c,d,e,f,g} = 7'b0110000; //b,c are on
            2: {a,b,c,d,e,f,g} = 7'b1101101; //c,f are off
            3: {a,b,c,d,e,f,g} = 7'b1111001; //f,e are off
            4: {a,b,c,d,e,f,g} = 7'b0110011; //a,d,e are off
            5: {a,b,c,d,e,f,g} = 7'b1011011; //b,e are off
            6: {a,b,c,d,e,f,g} = 7'b1011111; //b is off
            7: {a,b,c,d,e,f,g} = 7'b1110000; //a,b,c are on
            8: {a,b,c,d,e,f,g} = 7'b1111111; //nothing is off
            9: {a,b,c,d,e,f,g} = 7'b1111011; //just e is off
        endcase
    end
endmodule
```

پس از کامپایل کد بالا روی CPLD پروگرام میکنیم و با دادن ورودی های مختلف خروجی را مشاهده میکنیم.

۲. ابتدا کد زیر را در QUARTUS میزنیم:

```
module com(a,b,Cin,Cout,s);
    //a,b are numbers
    //Cin is carry in
    //Cout is carry out
    //s is sum
    input [3:0] a,b;
    input Cin;
    output reg [3:0] s;
    output reg Cout;
    reg [4:0] x;
    always @(*)
    begin
        x = a+b+Cin;
        if (x<10)
            {Cout,s}=x;
        else
            {Cout,s}=x+6;
    end
endmodule
```

پس از کامپایل کد بالا روی CPLD پروگرام میکنیم و با دادن ورودی های مختلف خروجی را مشاهده میکنیم.

۳. کد زیر را در برنامه استفاده کردیم:

```
module com(a,b,c,aBTb,bBTa,aETb);
    //a , b are 4bit numbers and c is cascade
    //aBTb means a>b,bBTa means b>a,aETb means a=b
    input [3:0] a,b;
    input [2:0] c;
    output aBTb, bBTa, aETb;
    wire [3:0] w1;
    wire [2:0] w2;
    wire w3;
    assign w1[0] = a[0] ^ b[0],
    w1[1] = a[1] ^ b[1],
    w1[2] = a[2] ^ b[2],
    w1[3] = a[3] ^ b[3],
    w3 = w1[0] & w1[1] & w1[2] & w1[3],
    w2[0] = w3 & c[0],
    w2[1] = w3 & c[1],
    w2[2] = w3 & c[2];
    assign aETb = w2[0],
    aBTb = (a[3] & ~b[3]) | (w1[3] & a[2] & ~b[2]) | (w1[3] & w1[2] & a[1] & ~b[1]) | (w1[3] & w1[2] & w1[1] & a[0] & ~b[0]) | w2[1],
    bBTa = w2[2] | ~(aETb|aBTb);
endmodule
```

سپس کد بالا را روی CPLD پروگرام کردیم و نتیجه را مشاهده کردیم و توسط TA تایید شد.

۴. کد زیر را در برنامه نوشتیم:

```
module com(a,b,c,aBTb,bBTa,aETb);
    //a , b are 4bit numbers and c is cascade
    //aBTb means a>b,bBTa means b>a,aETb means a=b
    input [3:0] a,b;
    input [2:0] c;
    output reg aBTb, bBTa, aETb;
    always @(*)
    begin
        if (a>b)
            {aETb, aBTb, bBTa} = 3'b010;
        else if (a<b)
            {aETb, aBTb, bBTa} = 3'b001;
        else if (a==b)
            begin
                case(c)
                    1: {aETb, aBTb, bBTa} = 4;
                    2: {aETb, aBTb, bBTa} = 2;
                    4: {aETb, aBTb, bBTa} = 1;
                endcase
            end
        end
    end
endmodule
```

سپس پس از کامپایل کد بالا آن را پروگرام کردیم و نتیجه را مشاهده کردیم. توسط TA تایید شد.

پرسش ۱) کدهای بخش ۳،۳ و ۳،۴ از نظر پیاده سازی، چه تفاوتی با هم دارند؟

کد بخش ۳،۳ در سطح جریان داده است اما کد ۳،۴ رفتار است یعنی کد ۳،۳ به زبان ماشین نزدیک تر است و بیشتر به ساختاری که توسط گیت ها پیاده می گردد نزدیک تر است.