

# SOC Design

한양대학교 공과대학 컴퓨터소프트웨어학부

최가온(학번: 2019009261)

**[Homework 1]** Design a 4-bit, 4:1 mux with Enable signal (mux8b\_4to1\_en)

## 1. Code

[https://github.com/Gaon-Choi/ITE4003/tree/main/Week03/mux8b\\_4to1](https://github.com/Gaon-Choi/ITE4003/tree/main/Week03/mux8b_4to1)

```
module mux8b_4to1(a, b, c, d, sel, enable, out);
    input [3:0] a, b, c, d;
    input [1:0] sel;
    input enable;

    output [3:0] out;
    reg [3:0] out;

    always @(a, b, c, d, sel, enable) begin
        if (!enable) begin
            case (sel)
                2'b00 : out = a;
                2'b01 : out = b;
                2'b10 : out = c;
                2'b11 : out = d;
            endcase
        end
        else
            out = 4'b0000;
        end
    end
endmodule
```

4비트 4-to-1 멀티플렉서는 4비트 입력 4개를 받아 sel의 값에 따라 그 중 하나를 출력으로 내보내는 모듈이다. 아래의 진리표를 기반으로 위의 코드를 작성하였다.

enable	sel	out
0	00	a
	01	b
	10	c
	11	d
1	00	0000
	01	
	10	
	11	

a~d를 포함한 입력값과 sel, enable의 값이 바뀔 때마다 출력이 바뀌어야 하므로 이들을 sensitivity-list에 포함시켰다. enable 값이 0인 경우 sel의 값에 관계없이 b'0000을 내보내야 하므로 1순위로 enable값을 if-else문으로 검사하였다. enable 값이 1인 경우 sel의 값에 따라 a~d 중 하나를 선택하여 출력해야 하는데, sel값과 a~d는 각각 일대일 맵핑되므로 case문으로 구현하였다.

## 2. Testbench

```

module tb_mux8b_4to1();
    reg [3:0] a, b, c, d;
    reg [1:0] sel;
    reg enable;
    wire [3:0] out;

    mux8b_4to1 U0 (a, b, c, d, sel, enable, out);

    initial begin
        a = 4'b0001; b = 4'b0010; c = 4'b0100; d = 4'b1000;
        #20 a = 4'b0010; b = 4'b0011; c = 4'b0101; d = 4'b1110;
        #20 a = 4'b1101; b = 4'b1000; c = 4'b1110; d = 4'b1001;
        #20 a = 4'b1111; b = 4'b1010; c = 4'b0110; d = 4'b1111;
        #20 a = 4'b0011; b = 4'b1110; c = 4'b0101; d = 4'b1010;
        #20 a = 4'b1001; b = 4'b0011; c = 4'b1100; d = 4'b1110;
        #20 a = 4'b1011; b = 4'b0111; c = 4'b0101; d = 4'b1001;
    end

    initial begin
        sel = 2'b00;
        #20 sel = 2'b01;
        #30 sel = 2'b10;
        #40 sel = 2'b11;
    end

    initial begin
        enable = 1'b1;
        #10 enable = 1'b0;
    end
endmodule

```

Signal	Value
/tb_mux8b_4to1/a	1011
/tb_mux8b_4to1/b	0111
/tb_mux8b_4to1/c	0101
/tb_mux8b_4to1/d	1001
/tb_mux8b_4to1/sel	11
/tb_mux8b_4to1/en...	0
/tb_mux8b_4to1/out	1110

- 1) 매순간마다 a~d 중 하나를 출력하기 때문에 어느 것이 나왔는지 구별하기 위해 매번 a~d의 값을 서로 다르게 설정하였다. 단, 위의 코드에서 3개의 initial문은 병행적으로 수행된다.
- 2) enable값이 0일 때는 b'0000을 출력하는데 이를 확인하기 위해 초기 10μs를 사용하였고 제대로 동작함이 확인되었다. 메인 기능은 다수의 값 중 하나를 택하는 것이므로 그 이후에는 enable = 1로 설정한 것으로 추정된다.
- 3) a~d의 갱신주기는 20μs지만, sel의 갱신주기는 10, 20, 30μs로 다르게 설정하였는데 이러한 설정을 통해 30μs~50μs 구간에서 sel의 값이 일정하지만 a~d값이 이 사이에 달라져 출력값이 달라진 것을 확인할 수 있다. 또한, 반대의 경우로 40μs~60μs 구간에서는 a~d값이 일정하지만 sel의 값이 이 사이에 달라져 출력값이 달라짐을 확인할 수 있다.
- 4) sel의 값을 b'00, b'01, b'10, b'11로 4가지 경우의 수 모두 테스트해본 것은 각각의 값마다 a~d 중 하나를 제대로 선택하는지 테스트해보기 위함이다.