

*Dataflow & Behavioral model로
Verilog coding 하기*

2:1 mux

◇ assign 이용

```
module mux2to1 (w0, w1, s, f);  
    input w0, w1, s;  
    output f;  
  
    assign f = s ? w1 : w0;  
  
endmodule
```

2:1 mux

◇ always 0|용

```
module mux2to1 (w0, w1, s, f);  
    input w0, w1, s;  
    output reg f;  
  
    always @(w0 or w1 or s)  
        if(s==0) f = w0;  
        else    f = w1;  
  
endmodule
```

4:1 mux

◇ case 0|용

```
module mux4to1 (W, S, f);  
    input [3:0] W;  
    input [1:0] S;  
    output reg f;  
  
    always @(W or S)  
        case (S)  
            0: f = W[0]; //2'b00: 도 OK  
            1: f = W[1];  
            2: f = W[2];  
            3: f = W[3];  
        endcase  
  
endmodule
```

과제

- ◇ 8:1 mux를 아래의 세 가지 종류로 각각 설계

- 1) assign
- 2) always 와 if-else
- 3) always와 case

- ◇ 제출 파일 **5개**

- ◆ Mux8_1_assign.v
- ◆ Mux8_1_ifelse.v
- ◆ Mux8_1_case.v
- ◆ Mux8_1_top.v (testbench)
 - ◆ 3종류의 mux instance의 입력은 공동으로 사용 가능하나 출력은 독립적으로 할당
- ◆ 시뮬레이션 결과 웨이브폼 캡처를 doc나 hwp에

11/9(토) 자정까지

제출 포맷과 기한을 지키지 않았을 경우 0점