#1.

(a)

wire는 단일게이트 or continuous assignment의 출력을 연결하기 위한 용도로 사용된다. 다음의 코드에서는 always가 사용되는 procedual assignment 이므로 reg를 wire로 바꾸면 에러가 발생한다.



(b)

@*의 의미는 always 문 내의 모든 변수를 포함한다는 뜻이므로 always @*는 모든 입력값이 변할때 표현식이 실행된다는 뜻이다.



(c)

sel은 2bit이므로 표현가능한 수는 2'b00, 2'b01, 2'b10, 2'b11 4가지이고 2'b00인 경우 out에 a를 할당. 2'b01인 경우 out에 b를 할당 2'b10인 경우 out에 c를 할당 2'b11 인 경우 out에 d를 할당 한다는 의미이다.

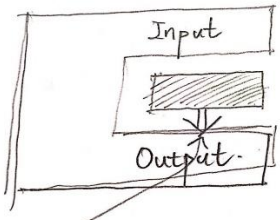


(d)

case구문의 각 실행문마다 break가 없으므로 해당 실행문부터 실행되며 endcase로 case문을 마무리하게 된다.

#2

(a)




출력으로 연결해야 하는 부분은 wire 즉, net 타입으로 해야만 오류가 안난다.



(b).

mux_4to1 모듈에 u1 이라는 이름으로 instantiation (모듈 가져오기)가 실행된 것이다.

a는 i0, b는 i1, c는 i2
d는 i3, sel은 s, out은 o를 가리키게 된다.



(c).

#10은 10만큼의 딜레이가 있다는 뜻이고
{i3, i2, i1, i0}은 i3가 MSB,
i0가 LSB라는 뜻이다.

s가 0일때는 {0,0,0,1} = 1
s가 1일때는 {0,0,1,0} = 2
s가 2일때는 {0,1,0,0} = 4
s가 3일때는 {1,0,0,0} = 8 로
설정해준다.

(d)

always @(s) 라는 것은

s의 입력값이 변할때 22, 23행이 실행되고

s값에 따라 이전에 할당된 값들이 출력되게

하는 조합회로이루어진다.



#5를 쓴 경우 출력 값

```
[2021-04-28 07:10:13 EDT] iverilog '-Wall' design.sv testbench.sv  && unbuffer vvp a.out
time =                  25, sel = 00, input = 0001,output=1
time =                  35, sel = 01, input = 0001,output=0
time =                  45, sel = 10, input = 0001,output=0
time =                  55, sel = 11, input = 0001,output=0
time =                  75, sel = 00, input = 0010,output=0
time =                  85, sel = 01, input = 0010,output=1
time =                  95, sel = 10, input = 0010,output=0
time =                 105, sel = 11, input = 0010,output=0
time =                 125, sel = 00, input = 0100,output=0
time =                 135, sel = 01, input = 0100,output=0
time =                 145, sel = 10, input = 0100,output=1
time =                 155, sel = 11, input = 0100,output=0
time =                 175, sel = 00, input = 1000,output=0
time =                 185, sel = 01, input = 1000,output=0
time =                 195, sel = 10, input = 1000,output=0
time =                 205, sel = 11, input = 1000,output=1
Done
```

#5를 뺀 경우의 출력 값

```
[2021-04-28 07:18:22 EDT] iverilog '-Wall' design.sv testbench.sv  && unbuffer vvp a.out
time =                  20, sel = 00, input = 0001,output=x
time =                  30, sel = 01, input = 0001,output=1
time =                  40, sel = 10, input = 0001,output=0
time =                  50, sel = 11, input = 0001,output=0
time =                  70, sel = 00, input = 0010,output=0
time =                  80, sel = 01, input = 0010,output=0
time =                  90, sel = 10, input = 0010,output=1
time =                 100, sel = 11, input = 0010,output=0
time =                 120, sel = 00, input = 0100,output=0
time =                 130, sel = 01, input = 0100,output=0
time =                 140, sel = 10, input = 0100,output=0
time =                 150, sel = 11, input = 0100,output=1
time =                 170, sel = 00, input = 1000,output=1
time =                 180, sel = 01, input = 1000,output=0
time =                 190, sel = 10, input = 1000,output=0
time =                 200, sel = 11, input = 1000,output=0
Done
```

(e) 딜레이가 5만큼 없어지고

0001일 때 sel이 기존에 00에서 01으로 되고

0010일 때 sel이 기존의 01에서 10으로 되고

0100일 때 sel이 기존의 10에서 11으로 되고

1000일 때 sel이 기존의 11에서 00으로 된다.

3번

```
// Code your design here
module decoder3_to_8(in, out, en);
  input[2:0] in;
  input en;
  output [7:0] out;
    reg [7:0] out;

  always @(in or en)
    begin
      if(en)
        begin
          out=8'd0;
          case(in)
            3'b000: out[0] = 1'b1;
            3'b001: out[1] = 1'b1;
            3'b010: out[2] = 1'b1;
            3'b011: out[3] = 1'b1;
            3'b100: out[4] = 1'b1;
            3'b101: out[5] = 1'b1;
            3'b110: out[6] = 1'b1;
            3'b111: out[7] = 1'b1;
            default: out=8'd0;
          endcase
        end
      else
        out = 8'd0;
    end
endmodule
```

4번

```
// Code your testbench here
// or browse Examples
module decoder_testbench;
  wire[7:0] out;
  reg en;
  reg[2:0] in;
  integer i;

  decoder3_to_8 dut(in, out, en);

  initial begin
    $monitor("en=%b, in=%d, out=%b", en, in, out);
    for(i = 0 ; i <16 ; i=i+1)
      begin
        {en,in} = i;
        #1;
      end
  end
endmodule
```

결과값

```
[2021-04-28 07:32:48 EDT] iverilog '-Wall' design.sv testbench.sv  && unbuffer vvp a.out
en=0, in=0, out=00000000
en=0, in=1, out=00000000
en=0, in=2, out=00000000
en=0, in=3, out=00000000
en=0, in=4, out=00000000
en=0, in=5, out=00000000
en=0, in=6, out=00000000
en=0, in=7, out=00000000
en=1, in=0, out=00000001
en=1, in=1, out=00000010
en=1, in=2, out=00000100
en=1, in=3, out=00001000
en=1, in=4, out=00010000
en=1, in=5, out=00100000
en=1, in=6, out=01000000
en=1, in=7, out=10000000
Done
```