

1. 必做内容：向量翻转

```
module top_module(  
    input  [7:0] in,  
    output [7:0] out  
);  
// Your codes should start from here.  
// .....  
    reg [7:0] temp;  
  
    always @ (*) begin  
        temp[7] = in[0];  
        temp[6] = in[1];  
        temp[5] = in[2];  
        temp[4] = in[3];  
        temp[3] = in[4];  
        temp[2] = in[5];  
        temp[1] = in[6];  
        temp[0] = in[7];  
    end  
  
    assign out = temp;  
// End of your codes.  
endmodule
```

2. 必做内容：最大值问题

(1)使用assign语句重新完成模块功能

```
    input  [7:0]      num1, num2,  
    output [7:0]      max  
);  
// Your codes should start from here.  
    assign max=((num1>num2)?num1:num2);  
// End of your codes.  
endmodule
```

(2)获取三个数的最大值

```
    input      [7:0]      num1, num2, num3,  
    output reg [7:0]      max  
);  
// Your codes should start from here.  
always @(*) begin  
    if (num1>num2)  
        max=num1;
```

```

        else
            max=num2;

    end

    always @(*) begin
        if(num3>max)
            max=num3;
        else
            max=max;
    end
    // End of your codes.
endmodule

```

(3)例化MAX2

```

    input  [7:0]      num1, num2, num3,
    output [7:0]      max
    reg [7:0] temp
);

module MAX2 (
    input      [7:0]      num1, num2,
    output reg [7:0]      max
);
always @(*) begin
    if (num1 > num2)
        max = num1;
    else
        max = num2;
end
endmodule

// Your codes should start from here.
MAX2 max2_1(.num1(num1), .num2(num2), .max(temp));
MAX2 max2_2(.num1(num3), .num2(temp), .max(max));
// End of your codes.
endmodule

```

3.必做内容：1的个数

```

    input      [2:0]      in,
    output reg [1:0]      out
);
// Your codes should start from here.
always @(*) begin
    out = (in[0])?(out+1'b1):out;
    out = (in[1])?(out+1'b1):out;
    out = (in[2])?(out+1'b1):out;
end

```

```
end
// End of your codes.
endmodule
```

4.选择性必做内容：题目2：Verilog运算符 (a = 8'b0011_0011, b = 8'b1111_0000)

```
input  [7:0]      a, b,
output [7:0]      c, d, e, f, g, h, i, j, k, l
);
assign c = a & b; //按位与 c=8'b0011_0000
assign d = a || b; //逻辑或 d=8'b1111_0011
assign e = a ^ b; //按位异或 e=8'b1100_0011
assign f = ~a; //按位非 f=8'b1100_1100
assign g = {a[2:0], b[3:0], {1'b1}}; //拼接运算符 g=8'0110_0001
assign h = b >>> 3; //算数右移 h=8'b1111_1110
assign i = &b; //归约与 i=8'b0000_0000
assign j = (a > b) ? a : b; //判断最大值 j=8'b1111_0000
assign k = a - b; //减 k=-8'b1011_1101
assign l = !a; //逻辑非 l=8'b0000_0000
endmodule
```