# 数字电路实验 Lab 1 实验报告

奚项正 PB23000020

## 必做内容

### 题目 1：向量翻转（2 分）

```
module top_module(
    input  [7:0]    in,
    output [7:0]    out
);
// Your codes should start from here.

assign out [7:0] = {in [0], in [1], in [2], in [3], in [4], in [5], in [6], in [7]}; //逐位翻转并拼接之

// End of your codes.
endmodule
```

### 题目 2：最大值问题（3 分）

(1) 请使用 assign 语句重新完成该模块的功能

```
module MAX2 (
    input  [7:0]         num1, num2,
    output [7:0]         max
);
// Your codes should start from here.

assign max = (num1 > num2) ? num1 : num2; //使用三元运算符实现

// End of your codes.
endmodule
```

(2) 获得三个数的最大值，使用 always 和 if-else 语句完成该功能

```
module MAX3 (
    input       [7:0]         num1, num2, num3,
    output reg  [7:0]         max
);
// Your codes should start from here.

always @(*) begin
```

```
    if (num1 > num2) //先比较并将num1与num2中更大者暂存于max
        max = num1;
    else
        max = num2;

    if (max < num3) //再比较max与num3，如果num更大则更新max的值为num3
        max = num3;
end

// End of your codes.
endmodule
```

(3) 获得三个数的最大值，通过例化 MAX2 模块实现该功能

```
module MAX3 (
    input   [7:0]         num1, num2, num3,
    output  [7:0]         max
);
// Your codes should start from here.

wire [7:0] max1;

MAX2 max2_1 ( //先比较并将num1与num2中更大者暂存于max1
    .num1(num1),
    .num2(num2),
    .max(max1)
);

MAX2 max2_2 ( //再比较并将max1与num3中更大者存于max
    .num1(max1),
    .num2(num3),
    .max(max)
);

// End of your codes.
endmodule
```

## 题目 3：一的个数（3 分）

```
module Count4Ones(
    input       [2:0]         in,
    output reg  [1:0]         out
);
// Your codes should start from here.

reg [2:0] temp; //用于临时存储in的值

always @(*) begin
```

```
    out = 2'd0; //将out赋初值为0
    temp = in;
    if (temp[0]) out = out + 2'd1; //逐位判断temp的各位是否为1，若是则out增加1
    if (temp[1]) out = out + 2'd1;
    if (temp[2]) out = out + 2'd1;
end

// End of your codes.
endmodule
```

## 选择性必做内容

### 题目 2：Verilog 运算符（2 分）

当 a = 8'b0011_0011, b = 8'b1111_0000 时各输出信号的值如下：

```
c = 8'b0011_0000
d = 8'b1111_0011
e = 8'b1100_0011
f = 8'b1100_1100
g = 8'b0110_0001
h = 8'b0001_1110
i = 1'b0
j = 8'b1111_0000
k = 8'b0100_0011
l = 1'b0
```