

# 浙江大学

## 本科实验报告

课程名称： 数字系统设计实验（周三 9，10 节）

姓 名：

学 院： 信息与工程学院

专 业： 电子科学与技术

学 号：

指导教师： 屈民军、唐奕

2020 年 5 月 11 日

# 浙江大学实验报告

专业：电子科学与技术

姓名：\_\_\_\_\_

学号：\_\_\_\_\_

课程名称：数字系统设计实验 指导老师：屈民军、唐奕

实验名称：常用组合电路模块的设计和应用

一、实验目的

二、实验任务与要求

三、实验原理

四、主要仪器设备

五、实验步骤与过程

六、实验调试、实验数据记录

七、实验结果和分析处理

八、讨论、心得

## 一、实验目的

- 掌握用 Verilog HDL 描述数据选择器、加法器和比较器等电路模块。
- 了解“自顶而下”的数字设计方法，掌握系统层次结构的设计。
- 掌握模块调用的方法，掌握参数定义和参数传递的方法。
- 掌握 ModelSim 的功能仿真的工作流程，进一步了解 Vivado 的工作流程。
- 认识到文件管理的重要性。

## 二、实验任务与要求

(1) 设计两数之差的绝对值电路：电路输入  $a_{in}$ 、 $b_{in}$  为 4 位无符号二进制数，电路输出  $out$  为两数之差的绝对值，即  $out = |a_{in} - b_{in}|$ 。要求用多层次结构设计电路，即调用数据选择器、加法器和比较器等基本模块来设计电路。

(2) 设计模式比较器电路：电路的输入为两个 8 位无符号二进制数  $a$ 、 $b$  和一个模式控制信号  $m$ ；电路的输出为 8 位无符号二进制数  $y$ 。当  $m=0$  时， $y = \text{MAX}(a, b)$ ；而当  $m=1$  时，则  $y = \text{MIN}(a, b)$ 。要求用多层次结构设计电路，即调用数据选择器、加法器和比较器等基本模块来设计电路。

## 三、实验原理

### 1. 电路的总体设计

#### (1) 两数之差的绝对值电路设计

两个无符号数之差的绝对值电路计算如下，减法可由加法补码完成，因此 (1) 式可以转化成 (2) 式计算。

$$out = \text{Max}(a_{in}, b_{in}) - \text{Min}(a_{in}, b_{in}) \quad (1)$$

$$out = \text{Max}(a_{in}, b_{in}) + (\sim \text{Min}(a_{in}, b_{in}) + 1) \quad (2)$$

实验名称：常用组合电路模块的设计和应用 姓名：\_

如图所示，其中数值比较器实现比较输入两数的大小；比较结果 agb 控制连两个数据选择器，数据选择器分别选出输入两数的最大值 Max 和最小值 Min；最后由全加器组成的思维加法器实现 Max 与 (-Min) 的补码加法。

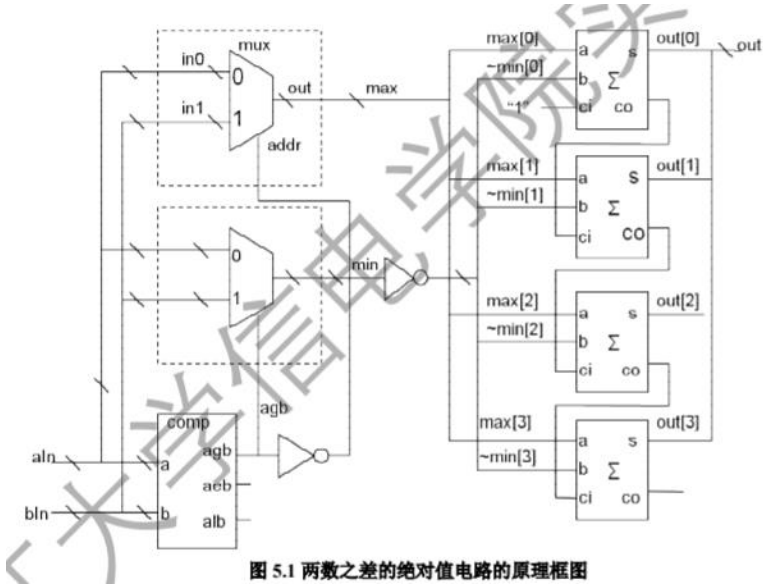


图 5.1 两数之差的绝对值电路的原理框图

(2) 模式比较器电路的顶层设计

如图所示，通过调用比较器模块实现 a、b 大小的比较，然后由组合电路产生数据选择的地址信号 addr，最后由二选一数据选择器得到输出信号。

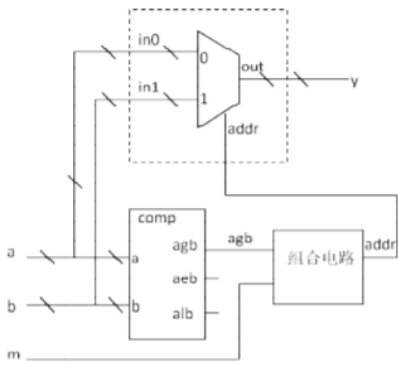


图 5.2 模式比较器的的原理框图

表 5.1 产生 addr 组合电路的功能表

m	agb	addr	备注
0	0	1	当 m=0, 且 a≤b 时, y 取大值 b
0	1	0	当 m=0, 且 a>b 时, y 取大值 a
1	0	0	当 m=1, 且 a≤b 时, y 取小值 a
1	1	1	当 m=1, 且 a>b 时, y 取小值 b

实验名称：\_\_\_\_常用组合电路模块的设计和应用\_\_\_\_ 姓名：\_\_

## 2. 模块的 Verilog HDL 描述

### (1) 两数之差的绝对值电路设计

该任务由顶层设计模块 (abs\_dif) 和功能模块数据选择器 (mux\_2to1)、数据比较器 (comp)、一位全加器 (full\_adder) 实现。在顶层设计模块中, 先调用数据比较器 (comp) 模块实现输入两数 aIn 和 bIn 大小的比较, 若 aIn 大于 bIn, 则得到数据比较器 (comp) 模块输出 agb=1, 否则 agb=0; 然后 agb 作为二选一数据选择器的地址传入两个数据选择器 (mux\_2to1) 模块, 得到最大值 max 和最小值 min; 最后通过调用四个一位全加器模块实现四位二进制数 max 与 min 的相减, 即最大值 max 与最小值 min 补码的相加。

代码如下:

```
module abs_dif(aIn,bIn,out);
    input [3:0]  aIn, bIn;
    output[3:0]  out;

    wire agb;
    comp #(N(4))comp_inst( .a(aIn), .b(bIn), .agb(agb), .aeb(), .alb());

    wire[3:0] max,min;
    mux_2to1 #(N(4))  mux1(.out(max), .in0(aIn), .in1(bIn), .addr(~agb));
    mux_2to1 #(N(4))  mux2(.out(min), .in0(aIn), .in1(bIn), .addr(agb));

    wire[2:0] c;
    full_adder  adder0(.a(max[0]),.b(~min[0]), .s(out[0]), .ci(1'b1),.co(c[0]));
    full_adder  adder1(.a(max[1]),.b(~min[1]), .s(out[1]), .ci(c[0]),.co(c[1]));
    full_adder  adder2(.a(max[2]),.b(~min[2]), .s(out[2]), .ci(c[1]),.co(c[2]));
    full_adder  adder3(.a(max[3]),.b(~min[3]), .s(out[3]), .ci(c[2]),.co() );
endmodule
```

### (2) 模式比较器的电路设计

该任务由顶层设计模块 (ModeComparator) 和功能模块数据选择器 (mux\_2to1)、数据比较器 (comp)、一位全加器 (full\_adder) 实现。在顶层设计模块中, 先调用数据比较器 (comp) 模块实现输入两数 a 和 b 大小的比较, 若 a 大于 b, 则得到数据比较器 (comp) 模块输出 agb=1, 否则 agb=0; 然后 agb 与模式输入 m 的同或值 ( $\sim (agb \wedge m)$ ) 作为二选一数据选择器的地址传入数据选择器 (mux\_2to1) 模块, 得到输出结果 out。

实验名称：\_\_\_\_常用组合电路模块的设计和应用\_\_\_\_ 姓名：\_\_\_\_

代码如下：

```
module ModeComparator(a, b, m, y);
    input [7:0] a, b;
    input m;
    output [7:0] y;
    wire agb;
    comp #(N(8)) comp_inst( .a(a), .b(b), .agb(agb), .aeb(), .alb());
    mux_2to1 #(N(8)) mux1(.out(y), .in0(a), .in1(b), .addr(~(agb^m)));
endmodule // ModeComparator
```

### （3）各功能模块的设计

#### ①数据选择器的设计

代码如下：

```
module mux_2to1(out, in0, in1, addr);
    parameter N=1;
    output[N-1:0] out;
    input[N-1:0] in0, in1;
    input addr;
    assign out=addr?in1:in0;
endmodule
```

#### ②数据比较器的设计

代码如下：

```
module comp(a, b, agb, aeb, alb);
    parameter N=1;
    input[N-1:0] a, b;
    output reg agb, aeb, alb;
    always @*
        begin
            if(a>b) {agb, aeb, alb}=3'b100;
            else if(a==b) {agb, aeb, alb}=3'b010;
            else {agb, aeb, alb}=3'b001;
        end
endmodule
```

实验名称：\_\_\_\_常用组合电路模块的设计和应用\_\_\_\_ 姓名：\_\_\_\_

### ③一位全加器的设计

代码如下：

```
module full_adder(a, b, s, ci, co);  
    input a, b;  
    output reg s;  
    input ci;  
    output reg co;  
    always @*  
        begin  
            {co, s}=a+b+ci;  
        end  
endmodule
```

## 四、主要仪器设备

- (1) 装有 Vivado、ModelSim SE 软件的计算机。
- (2) Nexys Video 开发板或 Basys3 开发板。

## 五、实验步骤和过程

### 1. 任务一 两数之差绝对值电路的设计

- (1) 编写一位全加器的 Verilog HDL 代码，并用 ModelSim 软件进行功能仿真。
- (2) 编写 n 位二选一数据选择器的 Verilog HDL 代码及其测试代码，并用 ModelSim 软件进行功能仿真。
- (3) 编写 n 位比较器的 Verilog HDL 代码，并用 ModelSim 软件进行功能仿真。
- (4) 对两数之差的绝对值电路进行功能仿真。
- (5) 建立 Vivado 工程，对工程进行综合、引脚约束、实现。

### 2. 任务二 模式比较器

编写模式比较器的 Verilog HDL 代码，并用 ModelSim 软件进行功能仿真。

实验名称：常用组合电路模块的设计和应用 姓名：\_\_\_\_\_

## 六、实验结果记录及分析

### 1. 任务一 两数之差绝对值电路的设计

#### (1) 一位全加器的 ModelSim 仿真结果

/full_adder_tb_v/a	1'b1								
/full_adder_tb_v/b	1'b1								
/full_adder_tb_v/ci	1'b1								
/full_adder_tb_v/s	1'b1								
/full_adder_tb_v/co	1'b1								

分析：由图可见当  $a=1, b=1, ci=1$  时， $\{co, s\}=2'b11$ ，当  $a, b, ci$  为其它值时，一位全加器结果仍然正确，因此一位全加器模块功能实现。

#### (2) n 位二选一数据选择器的 ModelSim 仿真结果

/mux_2to1_tb_v/in0	4'b1000	4b0000	4b0101	4b0110		4b1110	4b1000		
/mux_2to1_tb_v/in1	4'b1011	4b0000	4b1010	4b1101		4b1001	4b1011		
/mux_2to1_tb_v/addr	1'b0								
/mux_2to1_tb_v/out	4'b1000	4b0000	4b0101	4b0110	4b1101	4b1001	4b1011	4b1000	

分析：图中为 4 位二选一数据选择器的仿真结果。由图可见当  $in0=4'b0110, in1=4'b1101, addr=0$  时， $out=4'b0110$ ，正确输出了地址为 0 的值。当  $in0, in1, addr$  为其它值时，结果仍然正确，因此 n 位二选一数据选择器功能实现。

#### (3) n 位比较器的 ModelSim 仿真结果

/comp_tb/a	4'b0010	4b0000	4b0011	4b1011	4b1010	4b1111	4b0101	4b0000	4b0011	4b0010
/comp_tb/b	4'b1111	4b0000	4b1101		4b1001	4b0001	4b0101	4b0110	4b1110	4b1111
/comp_tb/agb	1'b0									
/comp_tb/aeb	1'b0									
/comp_tb/alb	1'b1									

分析：图中为 4 位比较器的仿真结果。由图可见当  $a=4'b1010, b=4'b1001$  时， $agb=1, aeb=0, alb=0$ ，正确判断出了  $a$  和  $b$  的大小，即  $a>b$ 。当  $a, b$  为其它值时，结果仍然正确，因此 n 位比较器功能实现。

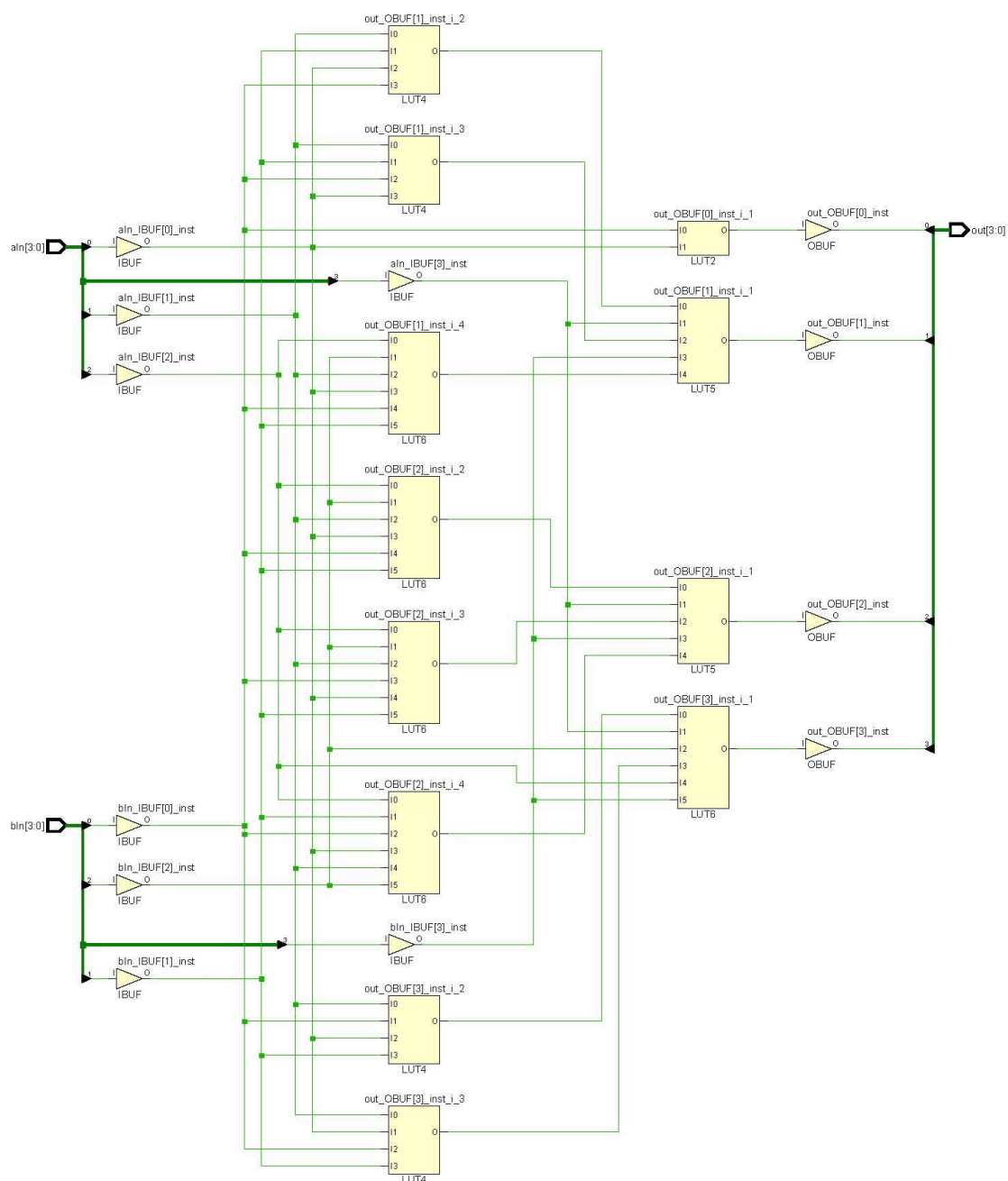
#### (4) 两数之差的绝对值的 ModelSim 仿真结果

/abs_dif_tb/out	4'b1100	4b1001		4b0110		4b0000		4b1100	
/abs_dif_tb/a	4'b1100	4b0011		4b1010		4b0101		4b1100	
/abs_dif_tb/b	4'b0000	4b1100		4b0100		4b0101		4b0000	

分析：当  $a=4'b0011, b=4'b1100$  时， $a<b, |a-b|=1100+1101=1001$ （弃掉溢出位）= $out$ ；当  $a=4'b1010, b=4'b0100$  时， $a>b, |a-b|=1010+1100=0110$ （弃掉溢出位）= $out$ ；其余依然正确。因此两数之差的绝对值模块功能实现。

实验名称：常用组合电路模块的设计和应用 姓名：\_\_

(5) Vivado 工程综合后得到的原理图





实验名称: 常用组合电路模块的设计和应用 姓名: \_\_\_\_\_

## 2. 任务二 模式比较器

+	/ModeComparator_tb/out	8'b10000100	8'b01111010	8'b10100111	8'b01000100	8'b000000101	8'b01100111	8'b10000100
+	/ModeComparator_tb/a	8'b10000100	8'b00100001	8'b10100111	8'b01000100	8'b000000101	8'b01110000	8'b10000100
+	/ModeComparator_tb/b	8'b10001101	8'b01111010	8'b000000100	8'b01000100	8'b000000101	8'b01100111	8'b10001101
	/ModeComparator_tb/m	1'b1						

分析：图为 8 位二进制数的模式比较器仿真结果。当  $a=8'b00100001$ ,  $b=8'b01111010$ ,  $m=0$  时,  $out=8'b01111010$ , 模式比较器正确输出了较大值；当  $a=8'b01100111$ ,  $b=8'b01110000$ ,  $m=1$  时,  $out=8'b01100111$ , 模式比较器正确输出了较小值；其余依然正确。因此模式比较器模块功能实现。

## 七、思考题

(1) 求两数之差的绝对值电路, 若输入为有符号数 (补码), 该怎样设计? 请画出原理框图并做简要说明。

答:

若输入为补码，求两数之差的绝对值，绝对值始终为正且不会发生溢出，故可以将输入值设置为补码形式，输出值为非符号数形式，下面分以下情况分析：

①a=0111, b=0011, 则 out=0111-0011=0100; 与输入非符号数绝对值结果相同

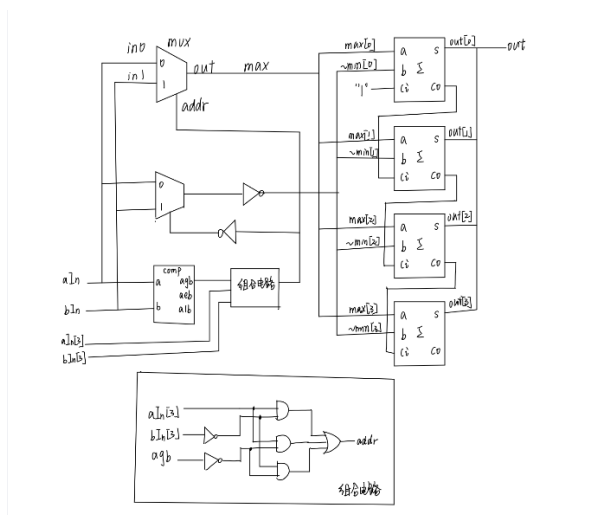
②  $a=1011$ ,  $b=0011$ , 则  $a$  表示十进制 -5,  $b$  表示十进制 3,  $out=8=1000=0011+0101=b+(\sim a)+1$

③a=1101, b=1001, 则  $\text{out} = |-3 - (-7)| = 4 = 1101 - 1001$ ; 与输入非符号数绝对值结果相同

得出结论, 只需要在之前设计的绝对值电路的二选一数据选择器的地址端做一定修改即可。

重新设计后 `addr=aIn[3]&&(~(bIn[3]))`    `||`    `aIn[3]&&(~agb)`    `||`    `(~bIn[3])&&(~agb);`

原理框图如下：



实验名称：常用组合电路模块的设计和应用 姓名：\_\_\_\_\_

代码如下：

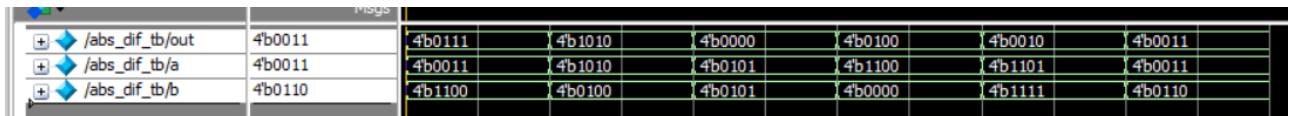
```
module abs_dif(aIn,bIn,out);
input [3:0]  aIn, bIn;
output[3:0]  out;

wire agb;
comp #(N(4))comp_inst( .a(aIn), .b(bIn), .agb(agb), .aeb(), .alb());

wire[3:0] max,min;
wire addr;
assign addr=aIn[3]&&(~(bIn[3])) || aIn[3]&&(~agb) || (~bIn[3])&&(~agb);//修改了地址端输入
mux_2to1 #(N(4))  mux1(.out(max), .in0(aIn), .in1(bIn), .addr( addr ));
mux_2to1 #(N(4))  mux2(.out(min), .in0(aIn), .in1(bIn), .addr( ~addr ));

wire[2:0] c;
full_adder  adder0(.a(max[0]),b(~min[0]),.s(out[0]),.ci(1'b1),.co(c[0]));
full_adder  adder1(.a(max[1]),b(~min[1]),.s(out[1]),.ci(c[0]),.co(c[1]));
full_adder  adder2(.a(max[2]),b(~min[2]),.s(out[2]),.ci(c[1]),.co(c[2]));
full_adder  adder3(.a(max[3]),b(~min[3]),.s(out[3]),.ci(c[2]),.co() );
endmodule
```

仿真结果：



Signal	Value
/abs_dif_tb/out	4'b0011
/abs_dif_tb/a	4'b0011
/abs_dif_tb/b	4'b0110

分析：后两组测试数据为新增测试数据。当  $a=4'b1101$ ,  $b=4'b1111$  时,  $out=|-3-(-1)|=2=4'b0010$ ; 当  $a=4'b0011$ ,  $b=4'b0110$  时,  $out=|3-6|=3=4'b0011$ ; 当  $a=4'b0011$ ,  $b=4'b1100$  时,  $out=|3-(-4)|=7=4'b0111$ ; 三种情况均正确。

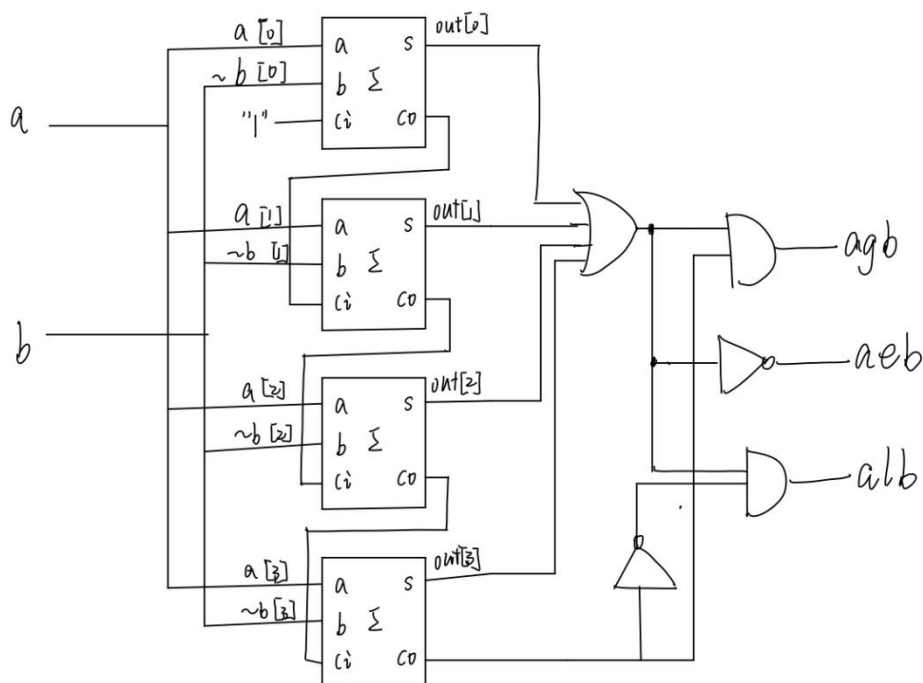
(2) 能否用加法器实现比较器, 若能, 请画出原理框图。

答：

能, 通过调用加法器模块计算两数之差, 计算  $a-b$ , 即  $a+(\sim b)+1$ , 此方法产生的溢出及输出值来比较两数大小。若  $a>b$ , 则溢出位为 1, 其余各位不全为 0; 若  $a=b$ , 则溢出位为 0, 其余各位为 0; 若  $a<b$ , 则不产生溢出。

实验名称：常用组合电路模块的设计和应用 姓名：\_\_

原理框图如下（4 位数据比较器）：



代码如下：

```
module comp(a, b, agb, aeb, alb);
input [3:0] a, b;
output agb, aeb, alb;
wire [3:0] out;
wire [3:0] c;

full_adder adder0(.a(a[0]),.b(~b[0]),.s(out[0]),.ci(1'b1),.co(c[0]));
full_adder adder1(.a(a[1]),.b(~b[1]),.s(out[1]),.ci(c[0]),.co(c[1]));
full_adder adder2(.a(a[2]),.b(~b[2]),.s(out[2]),.ci(c[1]),.co(c[2]));
full_adder adder3(.a(a[3]),.b(~b[3]),.s(out[3]),.ci(c[2]),.co(c[3]));

assign agb = c[3] && (out[0] || out[1] || out[2] || out[3]) ;
assign aeb = (~out[0]) && (~out[1]) && (~out[2]) && (~out[3]);
assign alb = (~c[3]) && (out[0] || out[1] || out[2] || out[3]);
```

endmodule

实验名称： 常用组合电路模块的设计和应用 姓名：

仿真结果如下：

/comp_tb/a	4'b0000	4'b0000	4'b0011	4'b1011	4'b1010	4'b1111	4'b0101	4'b0000	4'b0011	4'b0010
/comp_tb/b	4'b0000	4'b0000	4'b1101		4'b1001	4'b0001	4'b0101	4'b0110	4'b1110	4'b1111
/comp_tb/agb	1'b0									
/comp_tb/aeb	1'b1									
/comp_tb/alb	1'b0									

分析：结果与之前的比较器模块相同。功能实现

(3) 调用模块时怎样进行参数传递？若模块有多个参数，模块实例的格式如何？

答：

参数传递有两种格式：

在线显式重载参数方式：

模块名 #(参数 1(参数 1 值), .参数 2(参数 2 值), ...) 例化模块名(端口列表);

用 # 隐式地参数传递方式：

模块名 #(参数 1 值, 参数 2 值, ...) 例化模块名(端口列表);

模块例化也有两种方式：信号名关联和位置关联

信号名关联：full\_adder adder0(.a(a[0]),.b(~b[0]),.s(out[0]),.ci(1'b1),.co(c[0]));

位置关联：full\_adder adder0(a[0],~b[0], out[0], 1'b1,c[0]);

其中信号名关联可以按任意顺序排列信号，位置关联则需与模块定义时一一对应。

## 八：实验心得

今年因疫情原因，实验课无法线下进行，在老师没有讲解的情况下，我们需要自学 Verilog HDL 语言以及 ModelSim、Vivado 软件，本来以为这会是一件特别困难的事。但后来真正接触 Verilog HDL 语言时，发现它其实没有想象中的难，我再一次体会到了学过 C 语言之后很多语言都能很快上手的感觉。后来去做仿真实验，发现其实这些软件跟我们之前学过的电路仿真软件学起来差不多，只要认真看老师给的课件，我们自己就能完成大部分实验。在学 Verilog HDL 语言过程中，我发现我们自己写的代码没有真正的开发板很难调试，即使过了编译也不知道功能是否正确实现，后来慢慢知道可以通过写测试代码来测试。于是在做思考题时，就在老师给的测试代码上进行了修改，用来测试自己写的思考题代码。

本次实验各功能模块的设计相对简单，书上其实都有这些模块的现成代码了，顶层设计模块因为有老师给的例子，也没有那么困难，仿真也基本顺利。比较困难的是前两道思考题，思考题需要自己去设计电

实验名称： 常用组合电路模块的设计和应用 姓名：     

路，实现功能，老师没有给思路，基本上需要靠自己。如果没有本次实验的铺垫，一下子可能想不出来设计思路，但是前期我已经完成了两个任务和各功能模块的仿真，于是就在此基础上进行了代码的修改。在第一道思考题时我一开始设计的地址端输入总是得不到正确结果，后来借助卡诺图分析得出了正确的地址输入。但我感觉两道思考题的代码写的并不好，虽然功能得到了实现，但是代码质量并不高，以后需要不断学习，不断提高自己代码的质量。