

# 浙江大学

## 本科实验报告

lab5: 常用组合电路模块的设计和应用

课程名称: 数字系统设计实验 (周一 6, 7 节)

---

姓 名: 李昕

---

学 院: 信息与工程学院

---

学 号: 3230103034

---

指导老师: 屈民军、唐奕

---

2025 年 4 月 13 日

# 浙江大学实验报告

专业： 信息工程  
姓名： 李昕  
学号： 3230103034  
日期： 2025 年 4 月 13 日  
地点： 东 4-216

课程名称： 数字系统设计实验（周一 6，7 节） 指导老师： 屈民军、唐奕 成绩： \_\_\_\_\_  
实验名称： lab5: 常用组合电路模块的设计和应用 实验类型： 设计性实验 同组学生姓名： \_\_\_\_\_

## 一、 实验目的

- (1) 掌握用 Verilog HDL 描述数据选择器、加法器、比较器等基本电路模块。
- (2) 理解“自顶而下”的数字设计方法，掌握系统层次结构设计。
- (3) 掌握模块调用、参数定义与传递的方法。
- (4) 熟悉 ModelSim 功能仿真流程和 Vivado 综合、引脚约束、实现流程。
- (5) 认识到文件管理在项目设计中的重要性。

## 二、 实验任务和要求

- (1) 设计一个**两数之差绝对值**电路，输入为 4 位无符号二进制数  $a_{In}$  和  $b_{In}$ ，输出二者差值的绝对值  $|a_{In}-b_{In}|$ 。要求：
  - 采用多层次结构设计。
  - 利用数据选择器、加法器、比较器等模块实现。
- (2) 设计一个**模式比较器**电路：输入为两个 8 位无符号二进制数  $a$  和  $b$  以及模式控制信号  $m$ 。要求：
  - 当  $m = 0$  时，输出较大值；当  $m = 1$  时，输出较小值。
  - 采用模块调用及参数传递的方法完成设计。

### 三、 实验原理

#### 1. 电路的总体设计

##### 1.1 绝对值电路

两个无符号数之差的绝对值电路可以通过以下公式计算, 由于减法可以通过补码加法实现, 也可以化成下面的式子。

$$\text{out} = \max(\text{aln}, \text{bln}) - \min(\text{aln}, \text{bln}) \quad (1)$$

$$\text{out} = \max(\text{aln}, \text{bln}) + (-\min(\text{aln}, \text{bln}) + 1) \quad (2)$$

该电路主要有三部分组成:

- 数值比较器 (**comp**): 比较输入两数的大小。若  $a > b$ ,  $\text{agb} = 1$ ; 否则,  $\text{agb} = 0$ 。
- 数据选择器: 若  $\text{in0}$  为 1, 则输出  $\text{out}$  为 0;  $\text{in1}$  为 1,  $\text{out}$  为 1。
- 全加器

电路的原理框图如图??所示。

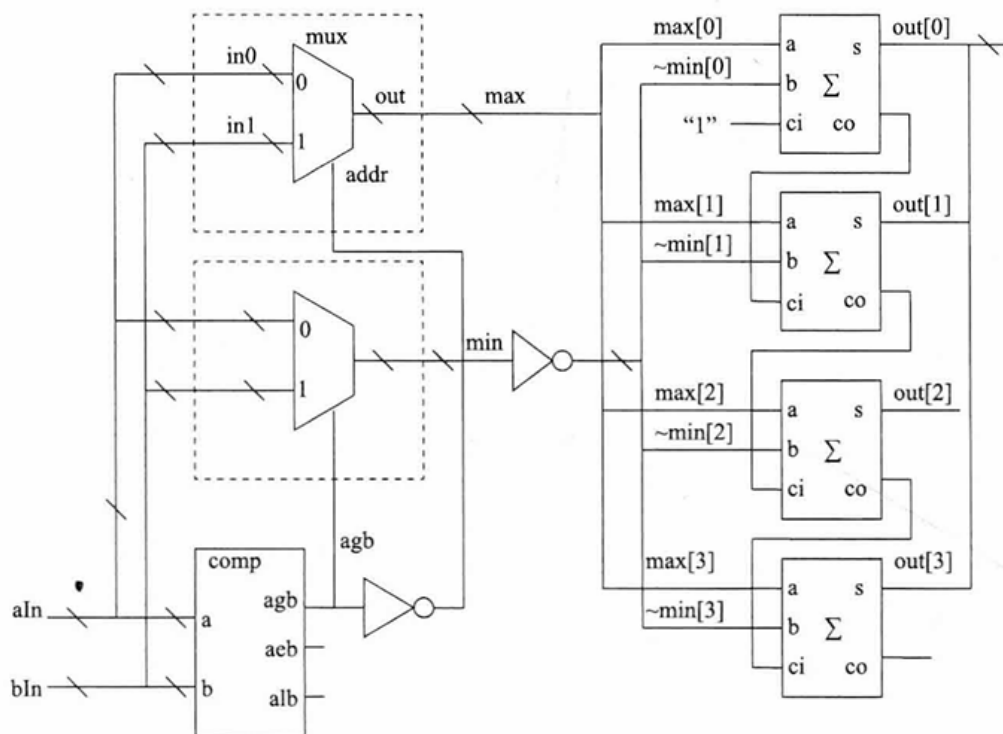


图 1: 两数之差的绝对值电路的原理框图

1.2 模式比较器

- 当模式信号  $m=0$  时，输出较大值，即  $y = MAX(a, b)$ 。
- 当  $m=1$  时，输出较小值，即  $y = MIN(a, b)$ 。

其设计框图如图??所示。

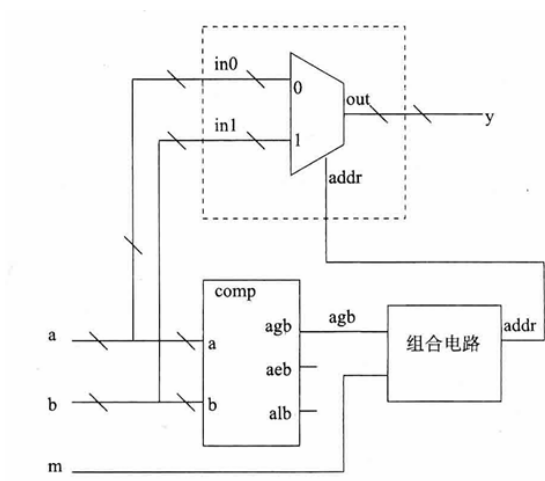


图 2: 模式比较器电路的原理框图

其中的组合电路产生数据选择的地址信号（addr 模块），功能如下表??所示。

m	agb	addr	备注
0	0	1	当 $m = 0$ ，且 $a \leq b$ 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 $a \leq b$ 时， $y$ 取小值 $a$
1	1	1	当 $m = 1$ ，且 $a > b$ 时， $y$ 取小值 $b$

表 1: 产生 addr 组合电路的功能表

2. 模块的 Verilog HDL 描述

2.1 顶层模块的 Verilog HDL 描述

两数之差的绝对值电路：

绝对值电路设计中，接收两个 4 位输入 aIn 和 bIn，并输出它们差的绝对值 out。通过一个 4 位比较器 comp，判断输入的大小关系，生成信号 agb 表示 aIn 是否大于 bIn。然后使用两个 4 位多路选择器 mux\_2to1，根据比较结果选择较大值和较小值分别赋给 max 和 min。最后，通过一组 4 位全加器 full\_adder，计算 max 减去 min 的结果，得到两数差的绝对值并输出。

代码如下:

//顶层设计的Verilog代码

//功能: 实现两数差的绝对值

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

    //数据比较器实例
    comp #(4) comp_inst(.a(aIn),.b(bIn),.agb(agb),.aeb(),.alb());

    //数据选择器实例
    wire[3:0] max,min;

    mux_2to1 #(4) mux1(.out(max), .in0(aIn), .in1(bIn), .addr(~agb));

    mux_2to1 #(4) mux2(.out(min), .in0(aIn), .in1(bIn), .addr(agb));

    //全加器实例, 注意信号组可拆开使用以及端口接常数的方法

    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
```

模式比较器的电路:

模式比较器的顶层设计模块 ModeComparator, 用于根据模式信号 m 和输入数据 a、b 的比较结果, 选择输出 y。模块首先通过一个 8 位比较器 comp 比较输入 a 和 b 的大小, 生成比较结果信号 agb。然后通过一个地址生成器 addr, 结合模式信号 m 和比较结果 agb, 生成控制信号 addr。最后, 利用一个 8 位多路选择器 mux\_2to1, 根据控制信号 addr 在输入 a 和 b 中选择一个作为输出 y, 实现基于模式的条件选择功能。

代码如下:

```
module ModeComparator(  
    input m,  
    input [7:0] a,  
    input [7:0] b,  
    output [7:0] y  
);  
    wire [7:0] agb;  
    wire addr;  
    comp #(n(8)) comp_inst (.a(a),.b(b),.agb(agb),.aeb(),.alb());  
    addr addr1(.agb(agb),.m(m),.addr(addr));  
    mux_2to1 #(n(8)) mux1(.in0(a),.in1(b),.addr(addr),.out(y));  
endmodule
```

## 2.2 各功能模块的设计

### 数据选择器模块 (2 选 1):

```
//数据选择器  
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 == 0)?in0:in1;  
endmodule
```

### 数据比较器模块:

```
module comp(a,b,agb,aeb,alb);  
    parameter n=1;  
    input[n-1:0] a,b;  
    output agb;  
    output aeb;  
    output alb;  
    assign agb = (a[n-1:0]>b[n-1:0])?1:0;  
    assign aeb = (a[n-1:0]==b[n-1:0])?1:0;  
    assign alb = (a[n-1:0]<b[n-1:0])?1:0;  
endmodule
```

### 一位全加器模块:

一位全加器有三个输入: 两个加数  $a, b$  和低位的进位  $c_{i-1}$ 。全加器有两个输出端: 本位  $s$  及进位  $c_o$ 。其功能主要由以下逻辑表达式实现:

$$S_i = A_i \oplus B_i \oplus C_{i-1}$$

$$C_i = A_i B_i + B_i C_{i-1} + A_i C_{i-1}$$

本位由两个加数与上一位的进位相异或所得到; 进位则是如果本位三个加数至少有两个为 1 时, 进位的值就为 1。

```
module full_adder(
    input a, b, ci,
    output s, co
);
    assign s = (a^b)^ci;
    assign co = a&b|b&ci|a&ci;
endmodule
```

## 四、 主要仪器设备

- Nexys Video 开发板 1 块。
- 装有 Vivado、ModelSim SE 软件的计算机。

## 五、 实验结果与分析

### 1. 任务一: 两数之差的绝对值电路的设计

#### 1.1 一位全加器电路的 ModelSim 仿真结果

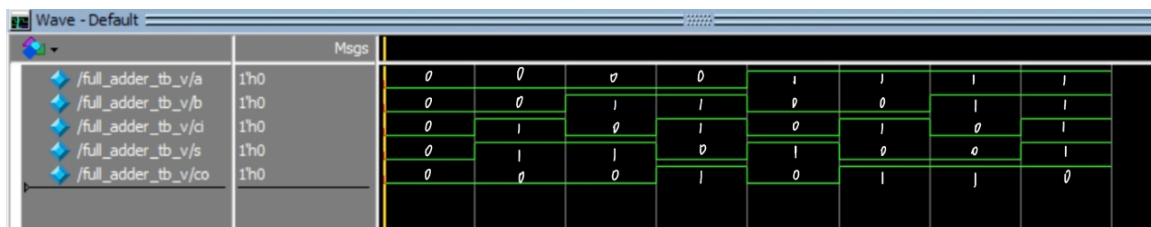


图 3: 一位全加器电路的 ModelSim 仿真结果

仿真结果表明: 当  $a, b, ci$  变化时, 输出的  $s$  和  $co$  正确反映了全加器的逻辑功能。

## 1.2 n 位二选一数据选择器的 ModelSim 仿真结果

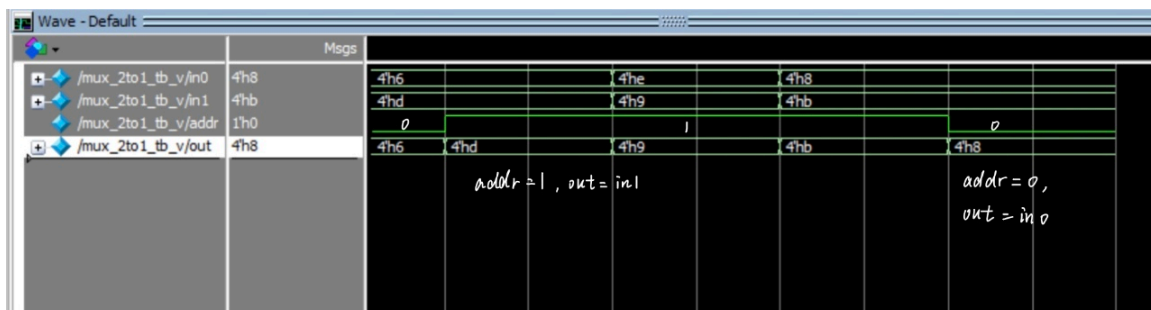


图 4: n 位二选一数据选择器的 ModelSim 仿真结果

当  $addr=0$  时,  $out$  即为  $in_0$  的值; 当  $addr=1$  时,  $out$  即为  $in_1$  的值; 因此  $n$  位二选一数据选择器功能实现。

## 1.3 n 位比较器的 ModelSim 仿真结果

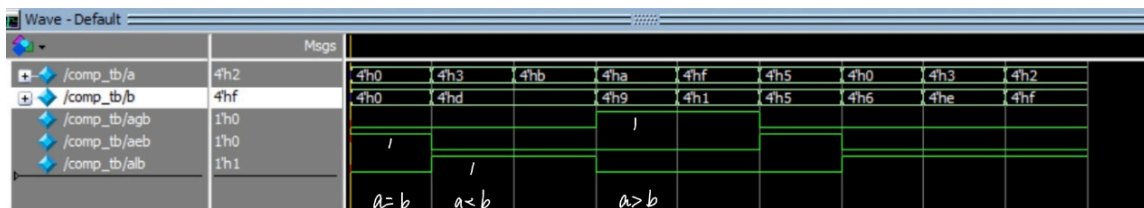


图 5: n 位比较器的 ModelSim 仿真结果

分析:

- 当  $a=4'h0$ ,  $b=4'h0$  时, 即  $a=b$ ,  $aeb=1$ ;
- 当  $a=4'h3$ ,  $b=4'hd$  时, 即  $a<b$ ,  $alb=1$ ;
- 当  $a=4'ha$ ,  $b=4'h9$  时, 即  $a>b$ ,  $agb=1$ ;

由仿真结果,  $n$  位比较器正确判断出了  $a, b$  的大小, 其功能实现。



## 1.4 两数之差绝对值的 ModelSim 仿真结果

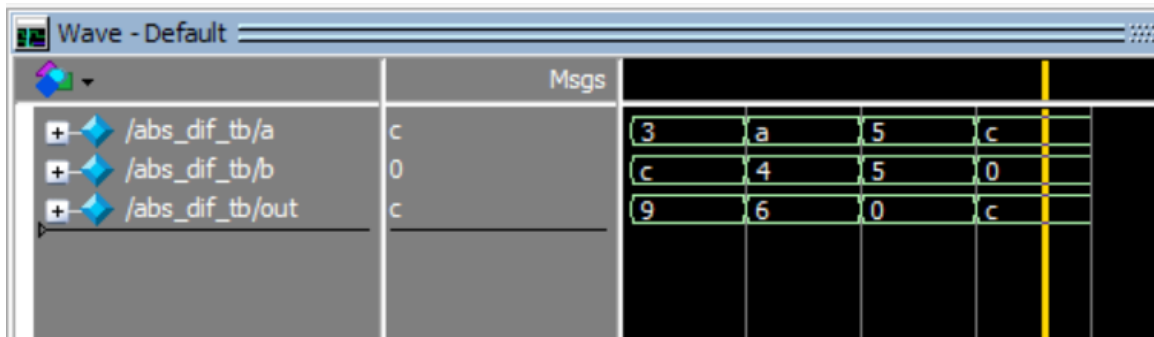


图 6: 两数之差绝对值的 ModelSim 仿真结果

- 当  $a=3, b=c$  时,  $|a-b|=|3-12|=9$ ;
- 当  $a=a, b=4$  时,  $|a-b|=|10-4|=6$ ;
- 当  $a=5, b=5$  时,  $|a-b|=|5-5|=0$ ;
- 当  $a=c, b=0$  时,  $|a-b|=|c-0|=c$ ;

该电路正确计算出了  $a, b$  的绝对值之差, 其功能实现。

## 1.5 Vivado 工程综合后得到的原理图

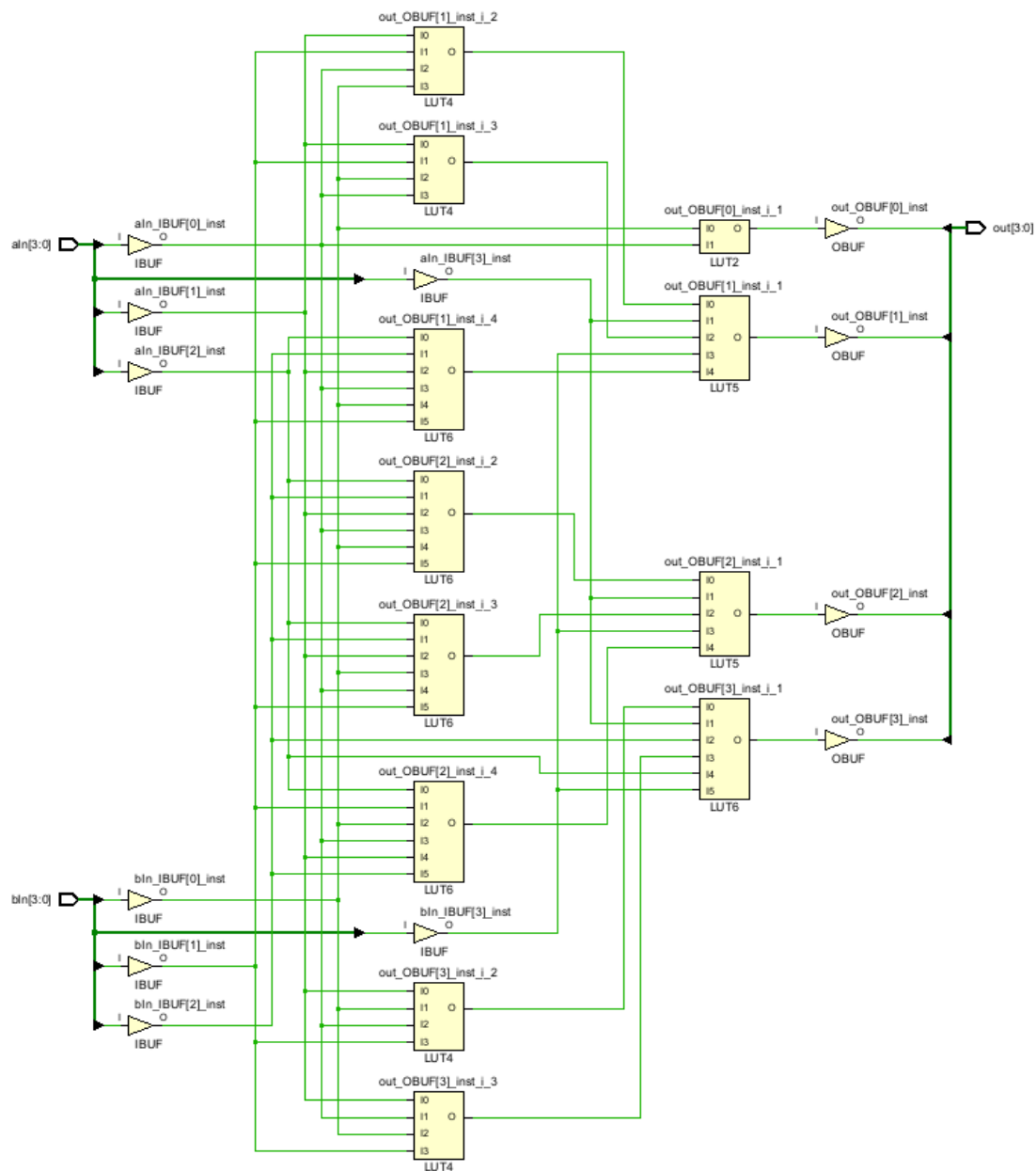


图 7: n 位比较器的 ModelSim 仿真结果

I/O 名称	I/O	引脚编号 (Nexys Video)	引脚编号 (Basys3)	接口类型	说明
aln[0]	Input	E22	V17	LVC MOS33	逻辑开关
aln[1]	Input	F21	V16	LVC MOS33	逻辑开关
aln[2]	Input	G21	W16	LVC MOS33	逻辑开关
aln[3]	Input	G22	W17	LVC MOS33	逻辑开关
bln[0]	Input	H17	W15	LVC MOS33	逻辑开关
bln[1]	Input	J16	V15	LVC MOS33	逻辑开关
bln[2]	Input	K13	W14	LVC MOS33	逻辑开关
bln[3]	Input	M17	W13	LVC MOS33	逻辑开关
out[0]	Output	T14	U16	LVC MOS33	LED 指示灯
out[1]	Output	T15	E19	LVC MOS33	LED 指示灯
out[2]	Output	T16	U19	LVC MOS33	LED 指示灯
out[3]	Output	T16	V19	LVC MOS33	LED 指示灯

表 2: 引脚约束内容

## 1.6 引脚约束

## 2. 模式比较器设计

## 2.1 ModelSim 仿真波形图

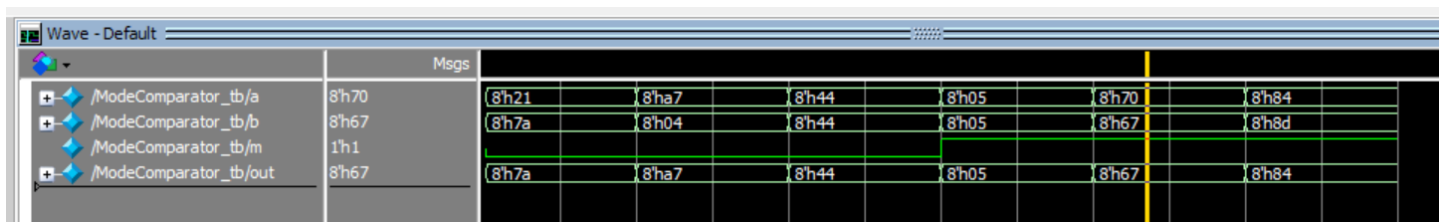


图 8: 模式比较器的 ModelSim 仿真结果

当  $m=1$  时, 输出  $a$  和  $b$  中的较小值; 当  $m=0$  时, 输出  $a$  和  $b$  中的较大值; 说明模式比较器设计逻辑正确。

## 六、 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。

考虑补码的概念: 若  $b < 0$ , 则  $-b = b + 1$ :

分类讨论:

(1) 若  $a > 0, b > 0$ , 则原结果保持不变;

(2) 若  $a < 0, b < 0$ , 则原先的绝对值电路由于未考虑电路的符号, 会导致错误;

(3) 若  $a < 0, b < 0$ , 以 -3 和 -5 为例, 两者的补码恰好是 1101, 1011, 绝对值恰好是 2, 不影响绝对值相减的结果;

由此, 我们只需要处理  $a < 0, b > 0$  (一正一负) 的特殊情况即可。即当  $aIn[3]$  是 1,  $bIn[3]$  是 0 时,  $addr$  无条件为 1;  $aIn[3]$  是 0,  $bIn[3]$  是 1 时,  $addr$  无条件为 0。

构建真值表如下:

表 3: 真值表

$aIn[3]$	$bIn[3]$	$agb$	$addr$	$\sim addr$
0	1	x	0	1
1	0	0	1	0
1	1	0	1	0
1	1	1	0	1
0	0	1	0	1
0	0	0	1	0

$$\therefore addr = aIn[3] \& (bIn[3] + agb) + (bIn[3] \& (agb))$$

因此, 在原逻辑图的基础上, 只需要修改  $addr$ 。总设计框图如下:

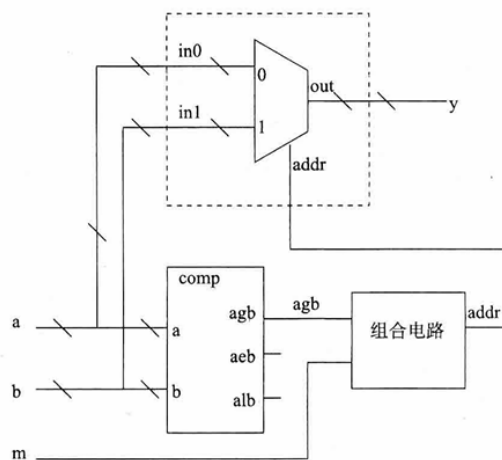


图 9: 总设计框图 (不变)

其中  $addr$  模块:

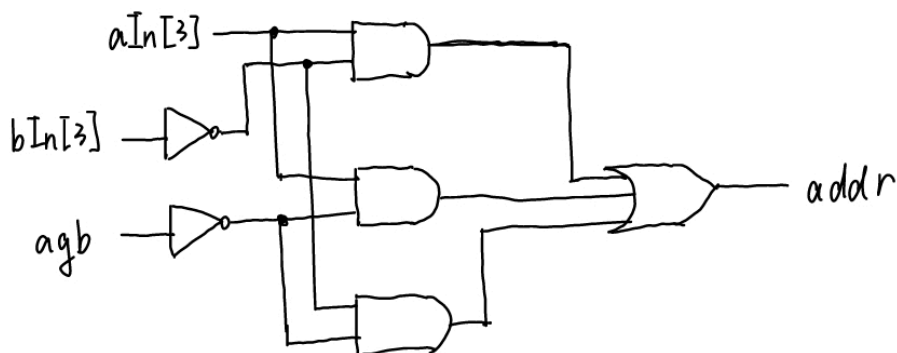


图 10: addr 模块

## 2. 用加法器实现比较器

利用  $(a-b)=a+(-b)+1$ , 使用全加器计算上述公式, 通过判断结果的正负来判断  $a, b$  大小, 如下图所示。

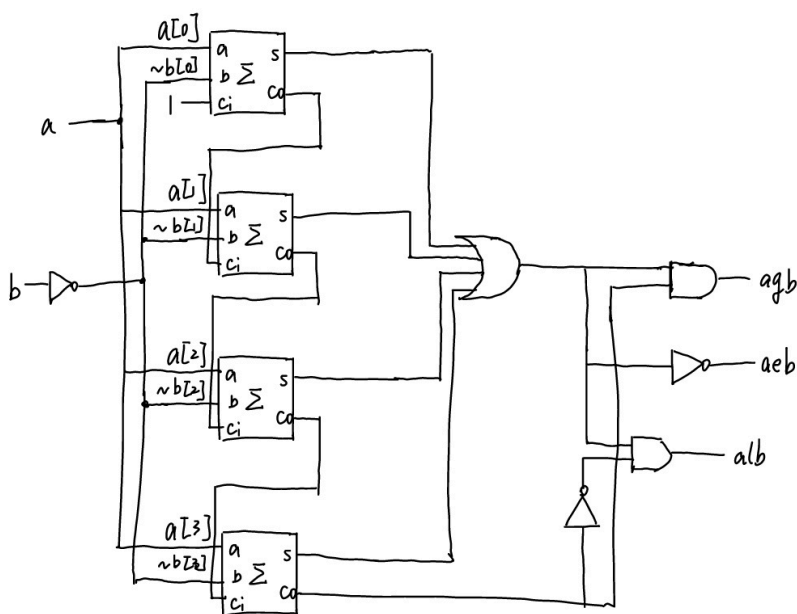


图 11: 全加比较器

### 3. 模块的参数传递

在 Verilog 中，模块的参数传递通过 `#(参数值)` 语法实现，允许在实例化模块时覆盖其内部定义的参数。

- 若按顺序传递: 参数值按模块定义中参数的声明顺序传递, 未指定的参数使用默认值, 格式为:

模块名 #(参数值1, 参数值2, ...) 实例名(端口连接);

- 若按名称传递, 显式指定参数名, 格式为. 参数名 (参数值), 顺序可自由调整。格式为:

模块名 #(.参数名1(值1), .参数名2(值2), ...) 实例名(端口连接);

## 七、 总结与心得

在本次实验中, 我第一次实践了 Verilog HDL 的实际应用, 掌握了 ModelSim 和 Vivado 软件的使用流程, 包括功能仿真、综合和实现等步骤。通过自己设计基本组合电路模块, 我加深了对这些理论知识的理解, 并体会到了模块化设计的优势。

此外, 我第一次接触到了“自顶而下”的数字设计方法。这种方法通过顶层模块的组织, 将复杂的系统分解为多个功能明确的子模块, 使设计过程更加清晰高效。此外, 我第一次使用 ModelSim 和波形调试工具进行调试, 尝试定位错误。

希望在未来的学习和设计中, 能够将这些经验应用到更复杂的电路设计中, 不断提升自己的专业能力, 为后续的研究和开发打下坚实的基础。