

## 实验 5 常用组合电路模块的设计和应用

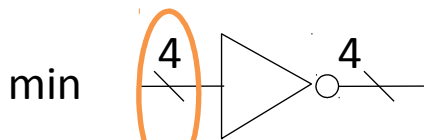
# 一、实验预习

从头到尾认真阅读教材，特别是“实验目的”部分，阅读两遍。

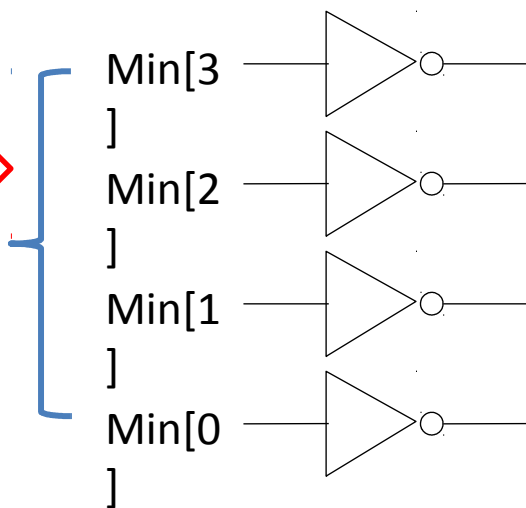
注意

•本实验主要培养规范，关键字：层次结构、模块调用、参数定义和参数传递、文件管理。

•教程中符号说明：



等效

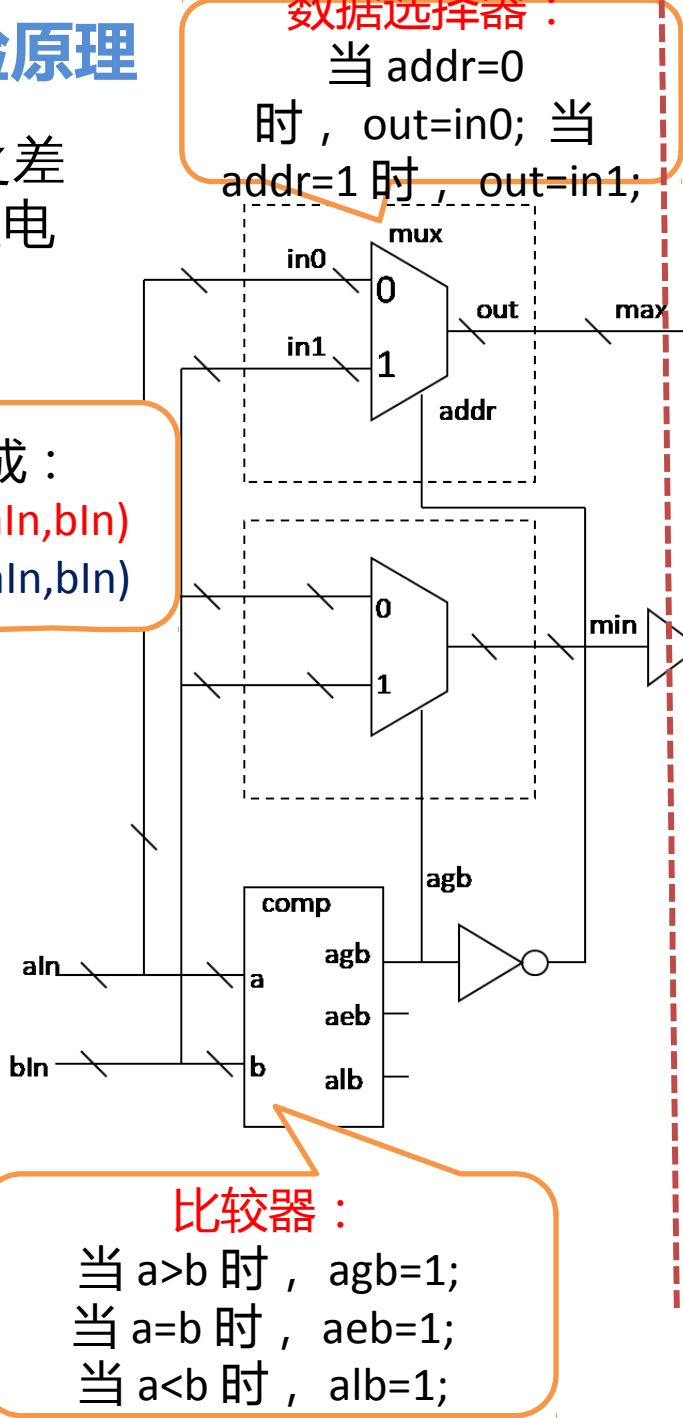


向量（数组、总线）：其中4为向量宽度 - 表示4根信号线。为了简化框图，教材中的省略向量宽度，由上下文可知

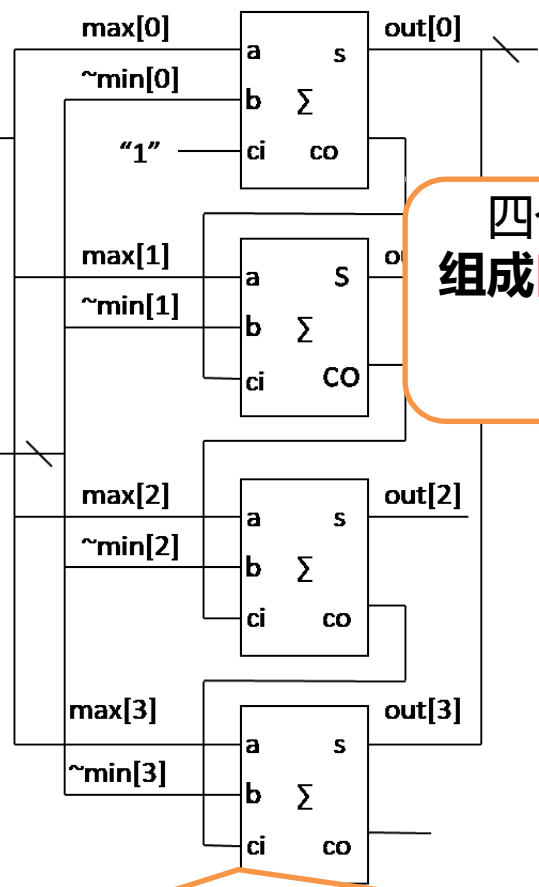
# 二、实验原理

## 1. 两数之差的绝对值电路

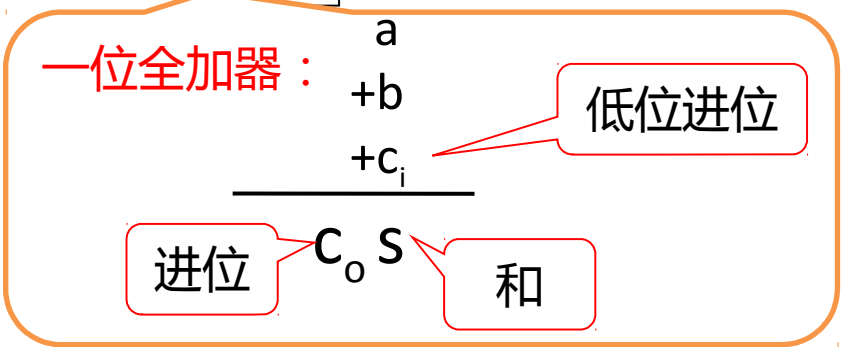
左边完成：  
 $max = max(aIn, bIn)$   
•  $min = min(aIn, bIn)$



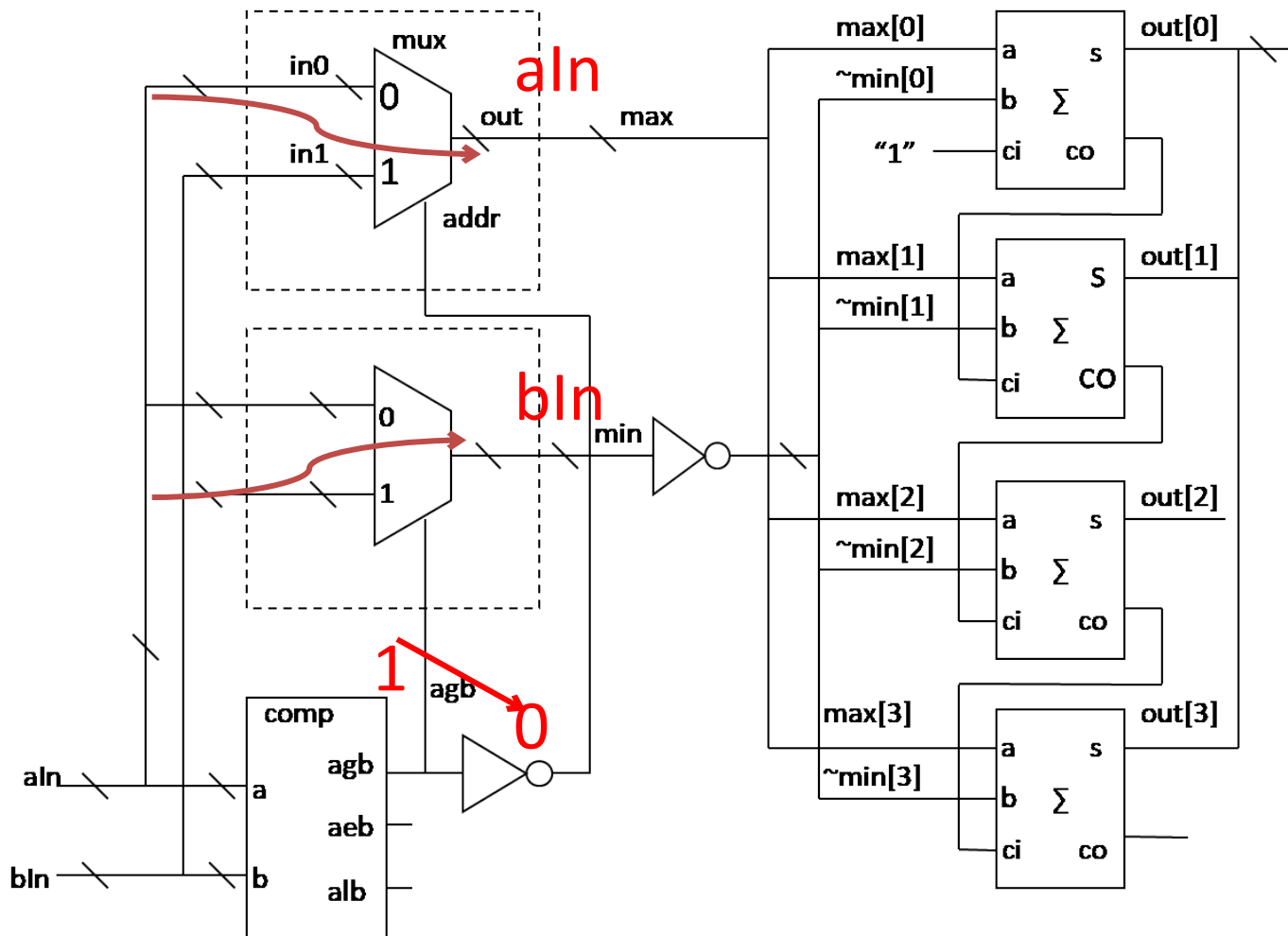
$$max-min = max + (-min)$$
$$= max + (\sim min + 1)$$



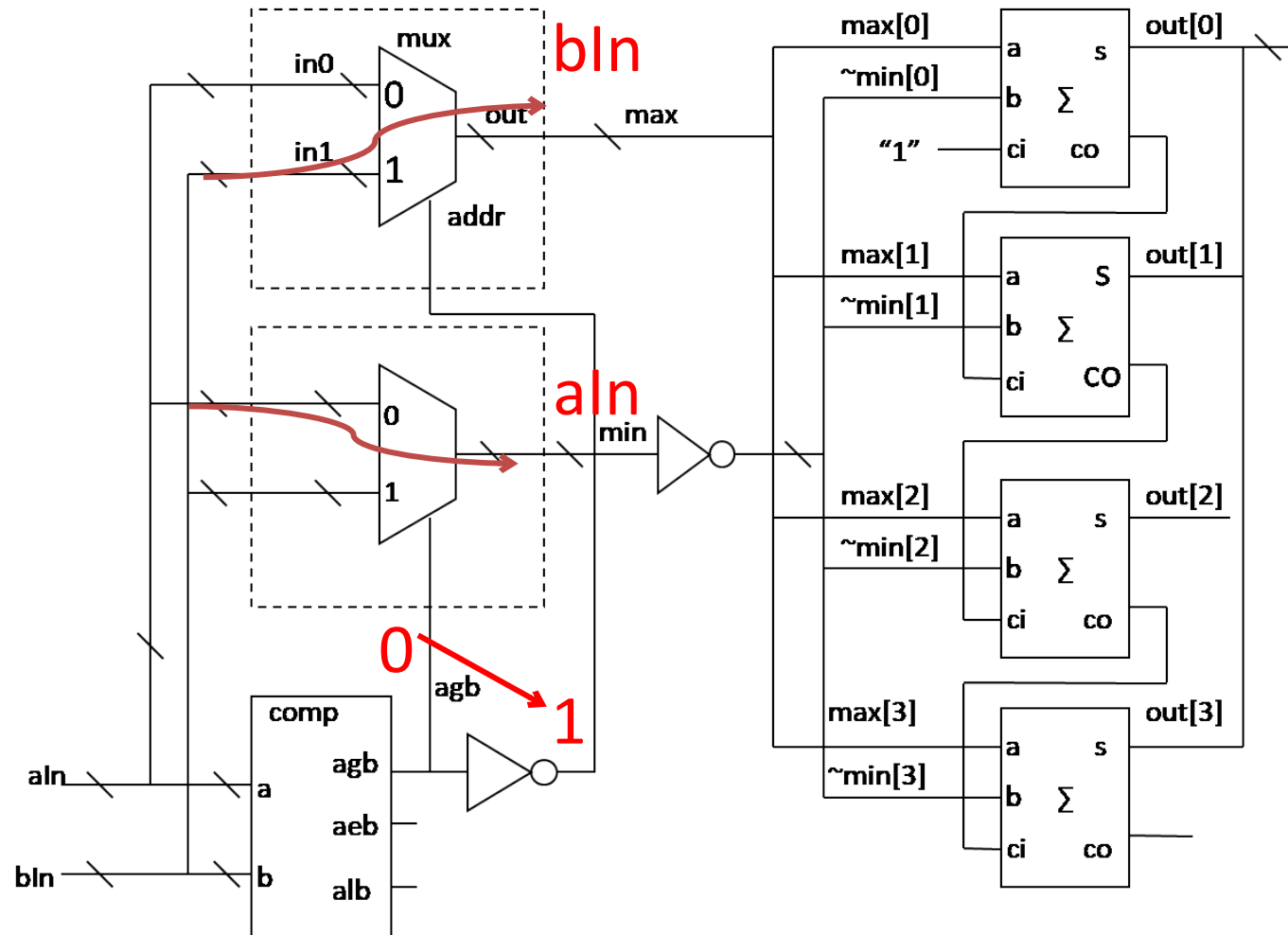
四个一位全加器  
组成四位加法器完成：  
 $max-min$



## ( 1 ) aIn>bIn 情况：

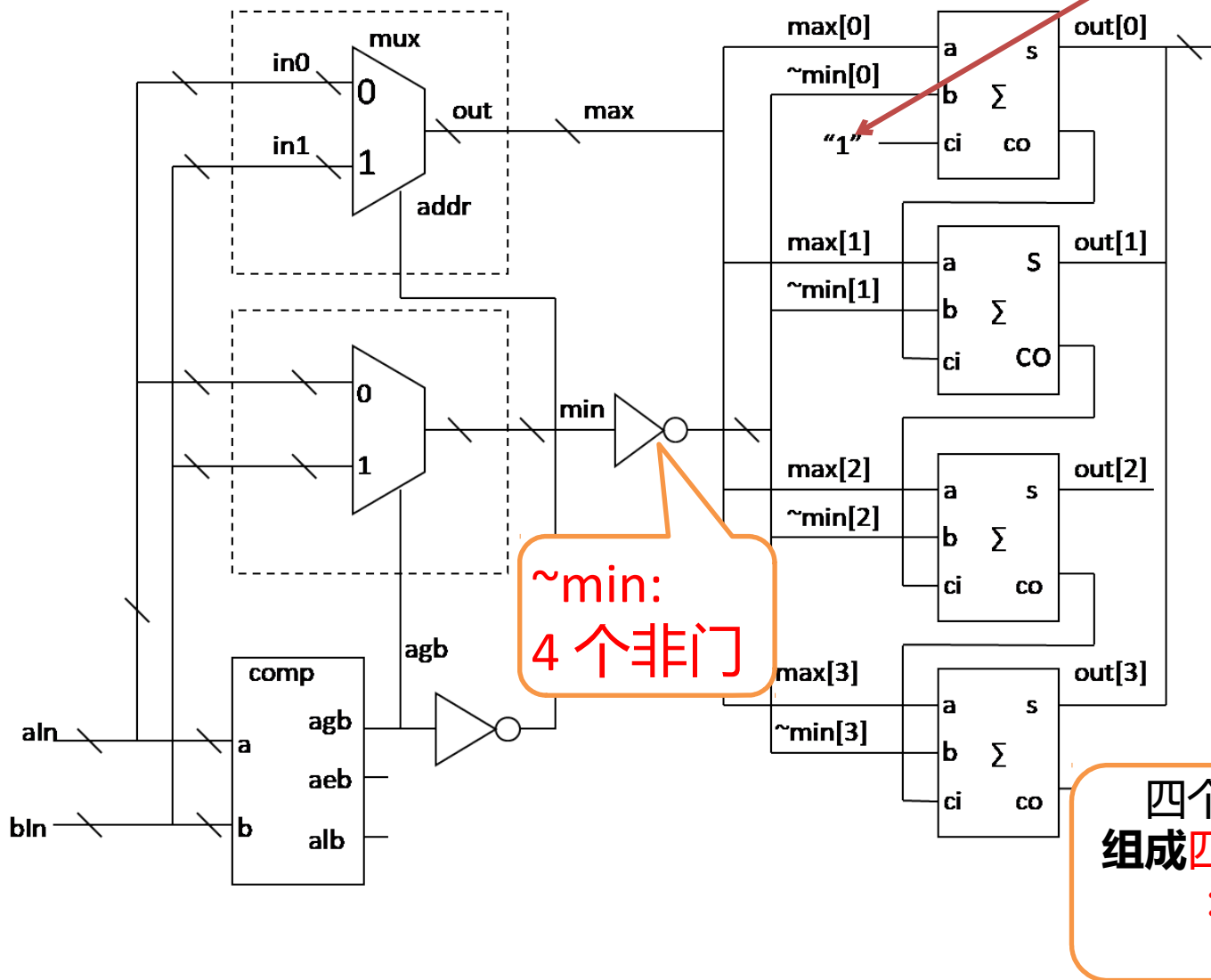


( 2 )  $aIn \leq bIn$  情况：

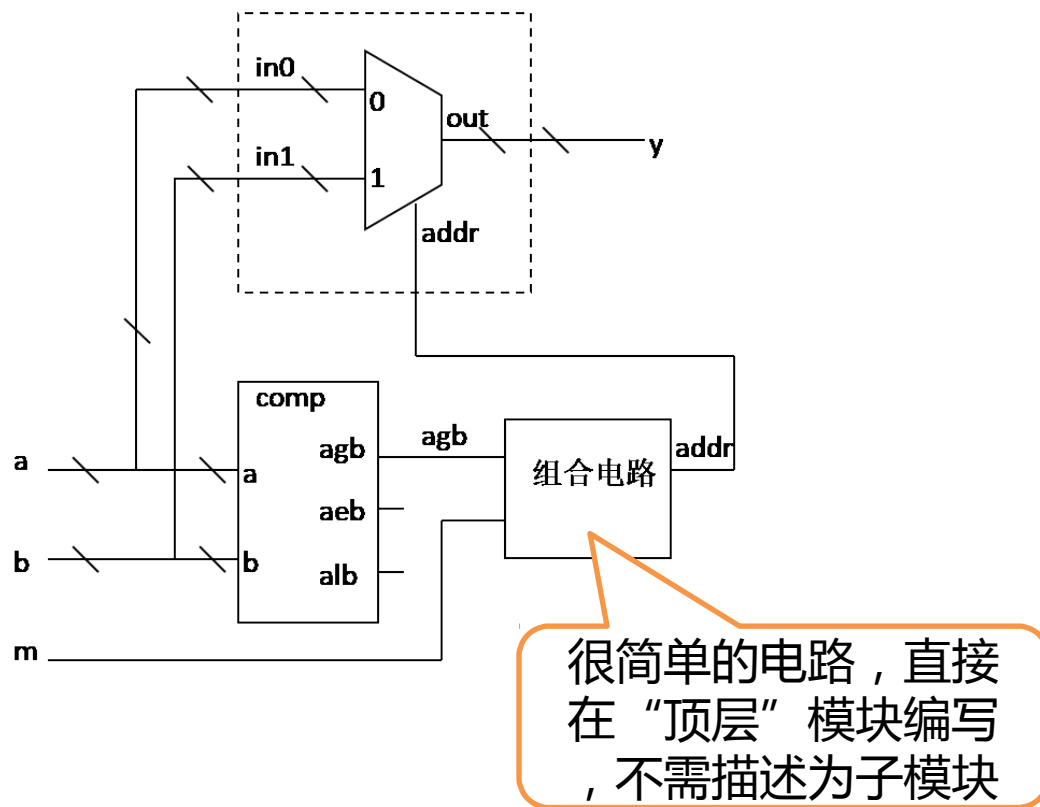


### ( 3 ) max-min

$$\text{max-min} = \text{max} + (\sim\text{min} + 1)$$



## 2 . 模式比较器电路



注意

- 任务 1、任务 2 都有：比较器、数据选择器
- 区别？位数不同

## 三、实验目的

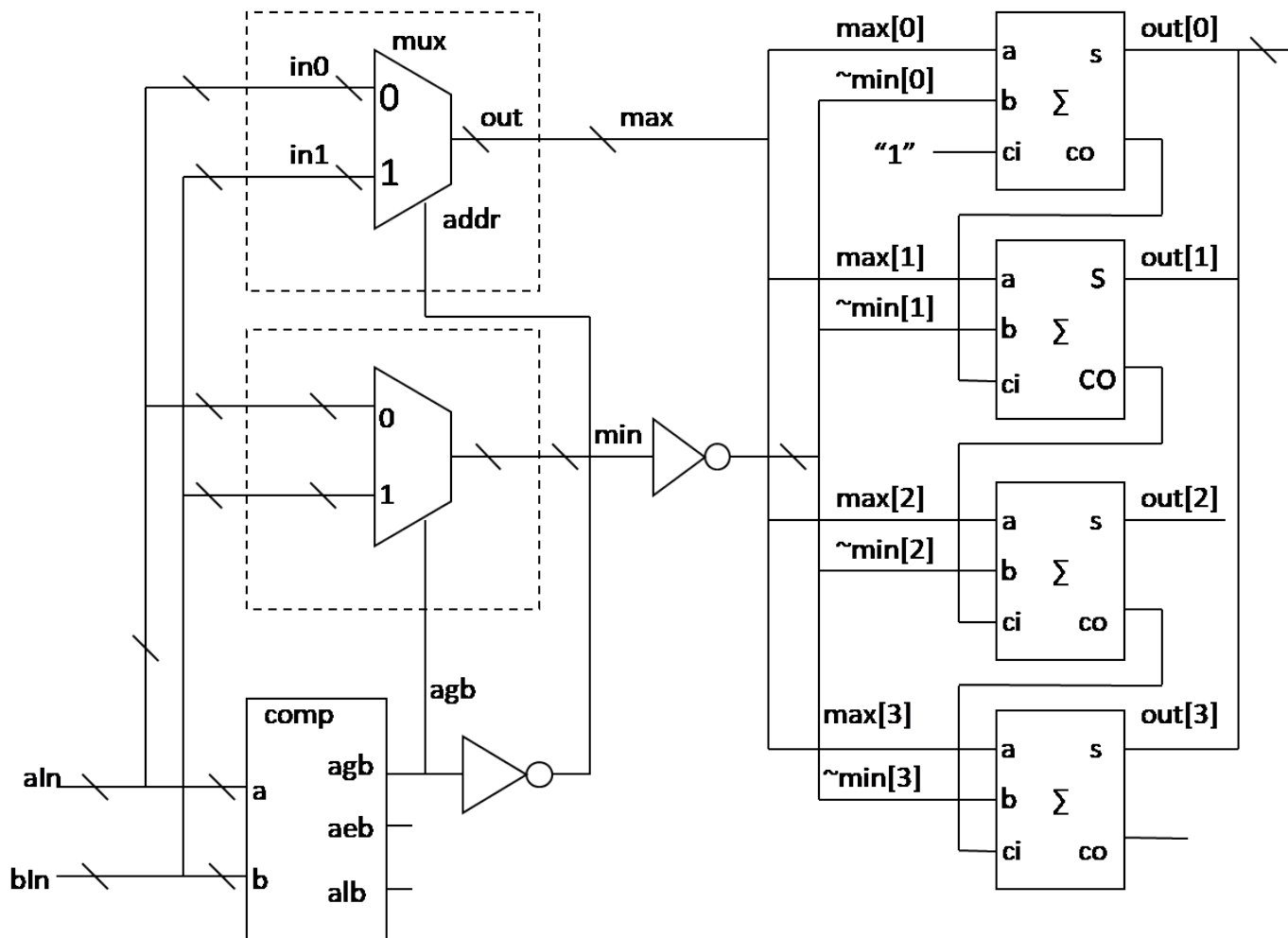
1. Verilog HDL 描述（设计）简单组合电路模块
2. 电路层次结构的设计
  - 模块调用的方法
  - 参数定义和参数传递的方法
3. 文件管理
4. 实验流程

## 四、设计要求

1. 任务 1、任务 2 都有：比较器、数据选择器
  - 区别？位数不同
  - 要求：两个任务共用一个比较器、数据选择器模块文件夹
  - 方法：参数定义和参数传递
2. 任务 1
  - 已提供的“顶层设计”：读懂，初步了解层次结构
  - 编写组合电路模块代码：掌握组合电路 HDL 描述
3. 任务 2
  - 基本上只需编写“顶层设计”代码：掌握层次结构和参数传递



## 五、顶层设计



## 调用模块方法：

模块名 <#(. 参数 1 (值 1) , . 参数 1 (值 2) ... )> 实例名 (

- . 端口 1 (与端口 1 相连的信号),
- . 端口 2 (与端口 2 相连的信号), ...

...);

```
module abs_diff(aIn,bIn,out);
```

```
input [3:0] aIn,bIn;
```

```
output[3:0] out;
```

```
// 比较器实例，在此实例描述中
```

```
// 注意空脚的连接方法
```

```
wire agb;
```

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

```
// 二选一数据选择器实例，注意 addr 信号的连接及参数的传递
```

```
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));
```

```
// 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;
```

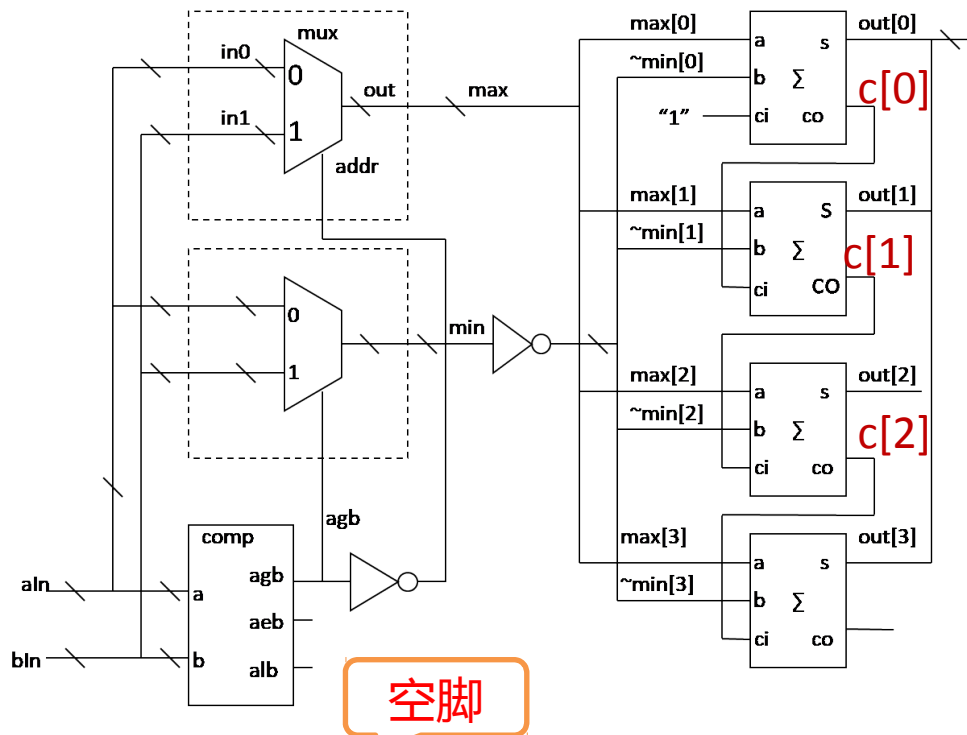
```
full_adder adder0(.a(max[0]),.b(~min[0]), .s(out[0]), .ci(1'b1),.co(c[0]));
```

```
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
```



空脚

常数

## 六、组合电路描述方法

组合电路，敏感信号表达式：always @(\*)  
过程描述中：被赋值的变量 ----reg 类型

组合电路描述组合电路方法：

- 真值表（卡诺图、最小项之和、最大项之积）：  
case 语句过程块描述
- 功能表：if-else 语句过程块或 case 语句过程块
- 表达式：assign 赋值
- 电路图：结构描述或 assign 赋值

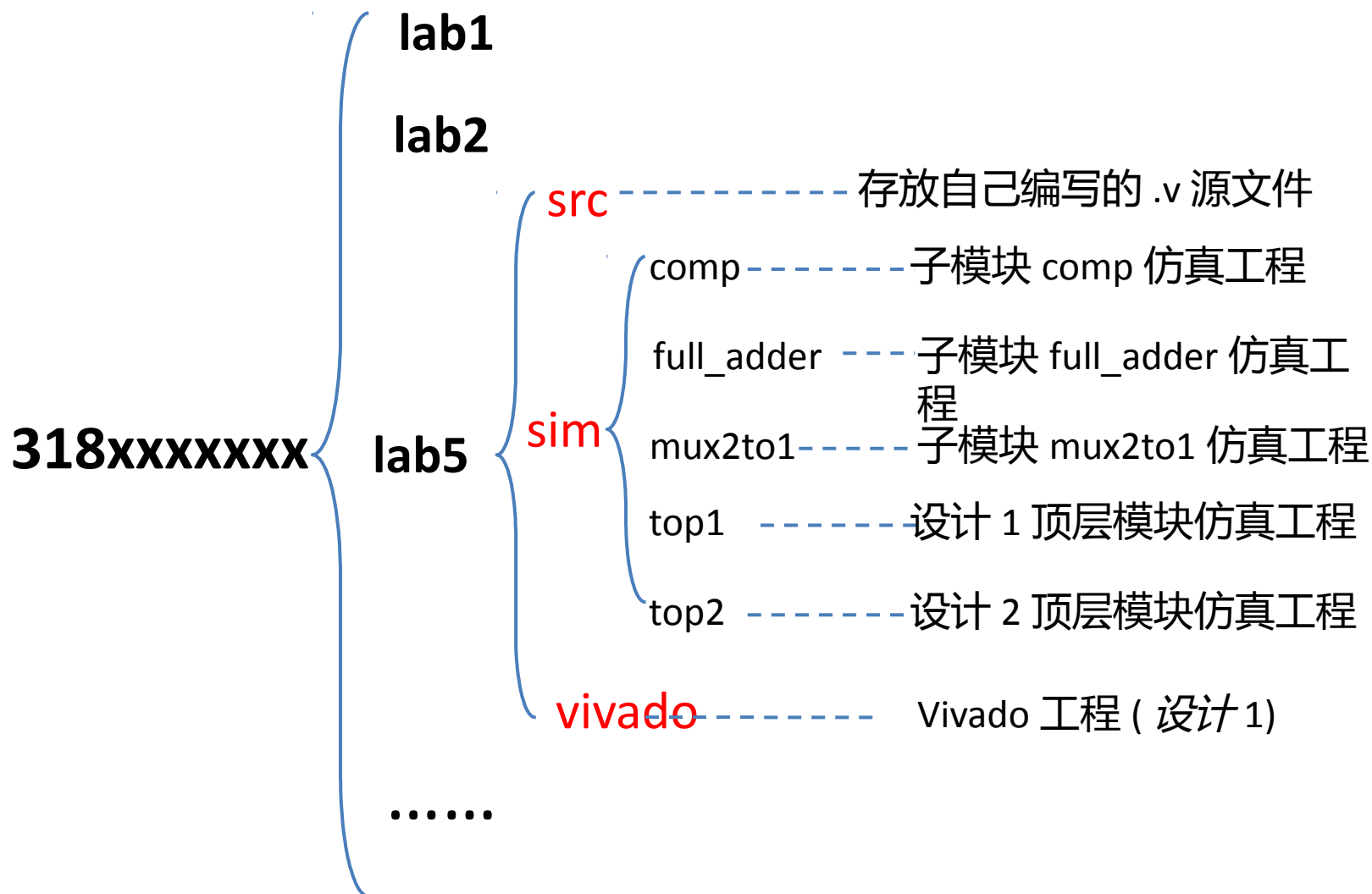
被赋值的变量 ----wire 类型

- 一位全加器真值表、表达式：教材（阎石） 4.3.4 章节 p192
- 数据选择器概念：教材（阎石） 4.3.3 章节 p188
- 比较器概念：教材（阎石） 4.3.5 章节 p197，本例允许使用比较运算符

注意

- 要求自学 Verilog HDL 语言
- 边学边做，边做边学
- 切记：不会实验？ --No. 做做就会了。

# 七、文件管理



# 八、实验步骤

## 1. 一位全加器的设计

- 编写一位全加器的 Verilog HDL 代码（编辑器可用写字板，Notepad++ 等编辑软件，建议用 Notepad++），保存在 318xxxxxxx\lab5\src 文件夹中，文件后缀为 .v
- 建一个 Modelsim 工程，工程保存在 318xxxxxxx\lab5\sim\full\_adder 文件夹中，往工程中添加老师提供的测试文件 full\_adder\_tb.v 和你刚才编写的全加器的 Verilog HDL 代码。
- 参照实验一介绍的方法，调试一位全加器。

## 2. 数据选择器的设计

与一位全加器设计方法相同，注意文件存放位置。

## 3. 比较器的设计

与一位全加器设计方法相同，注意文件存放位置。

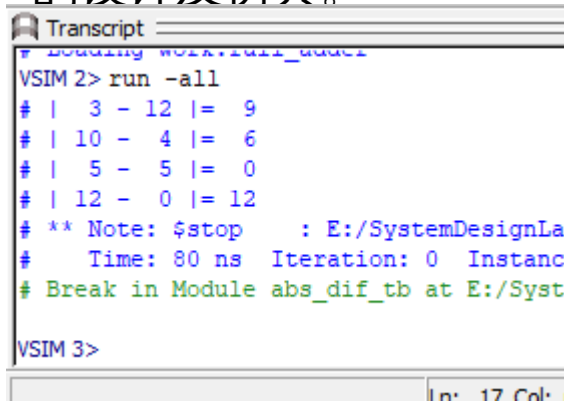


注意

上面 3 个步骤次序可调换

## 4. 任务一顶层设计

- 编写任务一的**顶层**代码（教材中已提供**顶层**代码，输入编辑即可），保存在 318xxxxxxx\lab5\src 文件夹中，文件后缀为 .v
- 建一个 Modelsim 工程，工程保存在 318xxxxxxx\lab5\sim\top 文件夹中，往工程中添加老师提供的测试文件 abs\_dif\_tb.v 和你编写的**顶层、全加器、数据选择器、比较器等 共 5 个** Verilog HDL 代码。
- 参照实验一介绍的方法，调试任务一的设计，如出现如下图实验结果，则完成任务一的设计及仿真。



```
Transcript
# Loading work_full_adder
VSIM 2> run -all
# | 3 - 12 |= 9
# | 10 - 4 |= 6
# | 5 - 5 |= 0
# | 12 - 0 |= 12
# ** Note: $stop      : E:/SystemDesignLa
#      Time: 80 ns  Iteration: 0  Instanc
# Break in Module abs_dif_tb at E:/Syst
VSIM 3>
```

**5. 建立 Vivado 工程文件，对工程进行综合、引脚约束、实现，并下载到开发实验板中对设计进行验证。（此步骤等回校后进行）**

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

## 九、实验报告

1. 写出实验目的，设计原理、列出 Verilog HDL 代码并对设计作适当说明。
2. 记录 ModelSim 仿真波形，并对仿真波形作适当解释，分析是否符合预期功能。
3. 记录实验结果（开发板上测试），分析设计是否正确。
4. 记录实验中碰到的问题和解决方法。
5. 回答思考题

**注意：**

**实验过程注意截图，用于报告。**

**谢谢！**