**实验报告**

专业： 混合班

姓名： 许展风

学号： 3210100658

日期： 3月22日

地点： 东4-223

课程名称： 数字系统设计实验 指导老师： 屈民军 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： 设计型

1. 实验目的
2. 掌握用Verilog HDL描述数据选择器、加法器和比较器等电路模块。
3. 了解“自顶而下”的数字设计方法，掌握系统层次结构的设计。
4. 掌握模块调用的方法，掌握参数定义和参数传递的方法。
5. 掌握ModelSim功能仿真的工作流程，进一步了解Vivado的工作流程。
6. 认识到文件管理的重要性。
7. 实验任务与要求
8. 设计两数之差的绝对值电路：电路输入两个4位无符号二进制数，输出两数之差的绝对值。要求采用多层次结构设计电路，先设计好基本模块，再设计顶层模块去调用基本模块来实现功能。
9. 设计模式比较器电路：电路输入两个8位无符号二进制数a、b和一个模式控制信号m，当m=0时，输出两数中的最大值，m=1时，输出两数中的最小值。同样要求采用多层次结构设计，并且通过参数传递的方法，使得不用编写功能重复的基本模块。
10. 实验原理与设计
11. 电路的总体设计
    1. 两数之差的绝对值电路：

对于输入的两个4位无符号二进制数a、b，需要先比较大小，将大数减去小数，而减法可以通过补码加法实现，因此整体的公式是：

out = Max(aIn, bIn) – ( ~ Min(aIn, bIn) + 1)

因此该电路由三个基本模块组成：数值比较器（comp）比较两数大小，输出agb=1，表示a大于b，agb=0，表示a小于b；数据选择器（mux）根据信号agb来选出最大值或最小值；最后由全加器模块实现四位补码加法。注意，最后的补码加法要对最小值取反并加1，所以第一位全加器的进位参数ci为1。

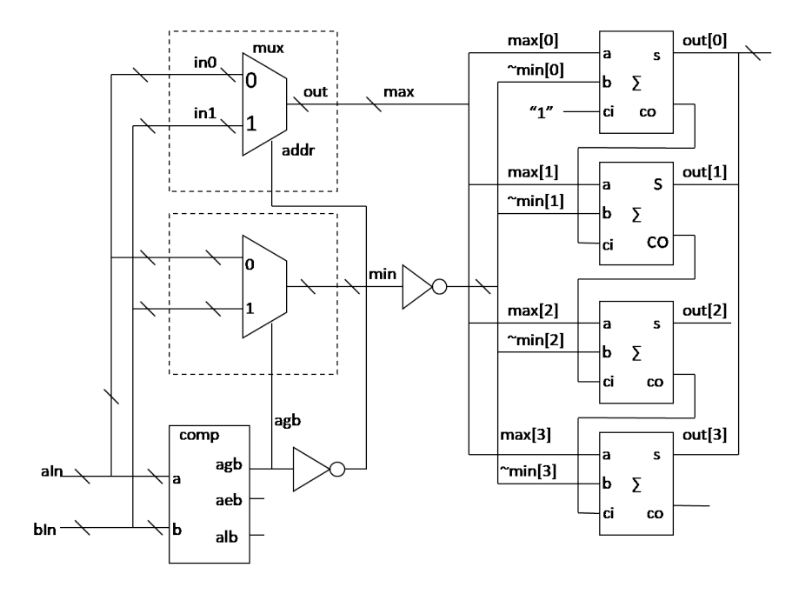


图 1两数之差绝对值电路设计

* 1. 模式比较器电路的顶层设计：

模式比较器电路相比于上文电路，多了一个输入信号m来决定输出最大值还是最小值，相当于要再设计一个组合电路模块，输入agb与m，输出addr。由真值表分析，可以看出addr的产生只是一个agb与m的同或运算，逻辑比较简单，可以直接在顶层模块中实现，所以最后我们只需要两个基本模块：数值比较器和数值选择器。

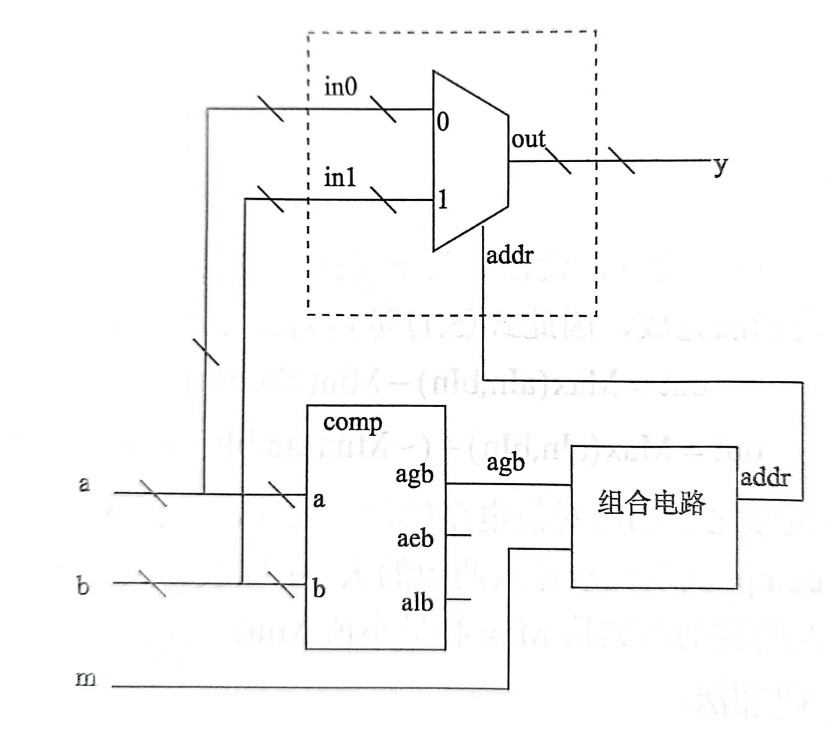


图 2模式比较器电路设计

1. 模块的Verilog HDL描述
   1. 顶层模块编写
      1. 两数之差绝对值电路的顶层模块设计是由教材直接提供的，源码这里就不再展示了。通过学习例程，可以发现几个要点。在这个顶层模块中，它调用了方案设计中的三个基本模块，通过模块实例化的方法，同时进行了参数传递，是基本模块符合要求。因为这里处理的输入信号是4位二进制的，所以比较器和数据选择器传入的参数都是n=4。同时可以学习到模块实例化中模块接口信号描述的规划性。在全加器实例中，则注意到把信号组拆开，一位一位处理。
      2. 仿照例程，依据实验方案，编写模式比较器的顶层模块

module ModeComparator (a,b,m,out);

input [7:0] a, b;

input m;

output [7:0] out;

wire agb;

comp #(.n(8))comp\_1(.a(a), .b(b), .agb(agb), .aeb(), .alb());

mux\_2to1 #(.n(8))mux\_1(.out(out), .in0(a), .in1(b), .addr((m && agb || ~m && ~agb)));

endmodule

定义模块的接口，两个8位的输入信号、一个1位输入信号，一个8位输出信号。接着实例化数据比较器模块，传入参数n=8，比较输入信号a和b，并输出agb信号。方案设计指出addr信号即是m与agb的同或运算，因此在数据选择器中，也要注意addr信号的连接。

* 1. 各功能模块的设计
     1. 数据选择器设计。基本模块的设计，一要掌握模块的逻辑功能，二要实现它的通用标准化，在本设计中体现在使用参数n表示数据位数，实现对任意位数信号的处理。接口描述教材已经给出，下面只展示逻辑功能实现部分代码。

reg [n-1:0] out;

always @(in0 or in1 or addr) begin

if(addr) out = in1;

else out = in0;

end

在这里运用了always语句来描述，在敏感信号变化后，即执行下面的if语句，根据addr信号来决定输出信号out的值。

* + 1. 数据比较器设计。其功能与数据选择器是类似的，也可以用always语句来描述。

reg agb, aeb, alb;

always @(a or b) begin

if (a > b) begin

agb = 1;

aeb = 0;

alb = 0;

end

else if (a == b) begin

agb = 0;

aeb = 1;

alb = 0;

end

else begin

agb = 0;

aeb = 0;

alb = 1;

end

end

* + 1. 全加器的设计，只需要实现一位全加器，教材已经给出了真值表以及表达式，本位和s是三个输入的异或，而进位是三个输入两两与运算后再或运算。运算逻辑可以简单的用assign语句来表达。

assign s = a ^ b ^ ci;

assign co = (a && b) || (a && ci) || (b && ci);

至此基本模块都设计完毕。

1. 主要仪器设备

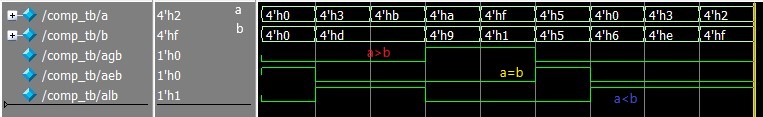
装有Vivado、ModelSim SE软件的计算机，Nexys Video开发板。

1. 实验步骤
2. 建立实验文件夹，将文件夹分类管理。
3. 编写好顶层模块、基本模块的代码，保存为.v文件在src文件夹中，并导入提供的测试文件。
4. 用ModelSim软件建立工程，分别对各个模块进行编译检错、测试仿真，输出波形验证，并记录波形。
5. 仿真完成后，用Vivado软件建立工程，进行综合，引脚约束，工程实现，生成比特流后，连接Nexys Video开发板，下载编程后实验验证。
6. 实验结果和分析处理
7. 仿真波形分析
   1. 数据选择器仿真



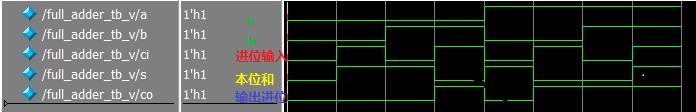
addr是控制信号，addr低电平时，out输出信号与in0输入信号相等；addr为高电平时，out输出信号与in1输入信号相等；从波形可以看出是符合要求的。且in1、in0、out都是满足要求的4位信号。

* 1. 数据比较器仿真



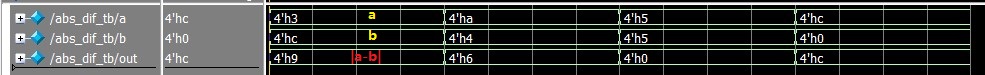
数据比较器比较输入信号a、b的大小，a>b时，只有agb为1；a=b时，只有aeb为1；a<b时只有alb为1，根据波形可以一一验证。

* 1. 全加器仿真



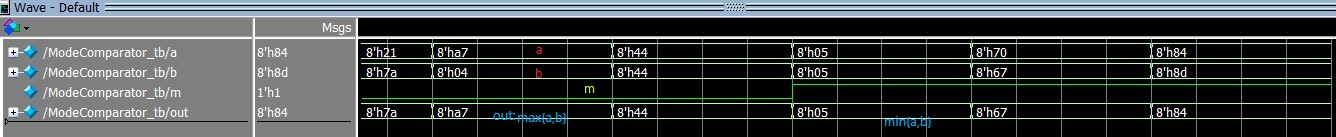
一位全加器实现一位加法，上面三行信号a、b、ci是输入，下面两行信号分别是本位和的结果和进位结果。例如最后一栏，a=1，b=1，ci=1，加法结果是11，本位和以及进位都是1，可以看到波形也都是高位，是正确的。

* 1. 两数之差绝对值仿真



电路输出a与b的绝对值之差，波形中的数值是16进制的，c-3=9，a-4=6，5-5=0，c-0=0，输出都是满足要求的。

* 1. 模式比较器仿真





放大图

模式比较器的控制信号是m，m为低电平，输出a、b中的最大值；m为高电平，输出a、b中的最小值。输入信号a、b，输出信号out，都是8位的。放大图中，m为低电平，a信号值a7大于b的04，因此输出a7。

至此，所有的基本模块、顶层模块都仿真验证完毕。

1. 讨论、心得

本实验的目的是训练数电实验的基础能力，从自顶向下的设计思路、层次化的设计方法，到电路模块设计、模块代码编写、Modelsim仿真、Vivado综合实现的实验流程，再到最具体基础的基本组合电路模块编写，简单体验了整个实验的流程。从各个方面都有所收获，特别是逐渐熟悉verilogHDL的编写和仿真时的纠错，文件管理的具体分类方法，去分析波形图等等。

1. 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。

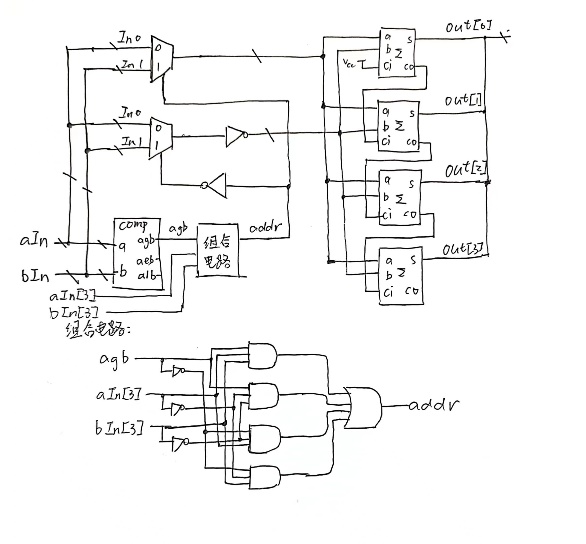
对于有符号数，有三种情况，a>0且b>0，a>0且b<0，a<0且b<0。处于第一种和第三种情况时，可以用无符号数的两数之差绝对值电路计算，因为高位符号位是一致的，不影响计算结果。例如，a=(1110)2=(-2)10，b=(1111)2=(-1)10，1110与1111之差的绝对值仍是1。

而对于第二种情况，a=(0010)2=(2)10，b=(1111)2=(-1)10，a与b的之差绝对值是|a+~b+1|，如果把补码形式输入原实验电路，则会计算|b+~a+1|，可以发现只要a与b调换就能得到正确值，分析原电路，可以发现只用addr取反就可以了。

把上述分析转化为逻辑符号，相当于:

addr = agb(aIn[3]bIn[3]’+aIn[3]’bIn[3])+agb’(aIn[3]bIn[3]+aIn[3]’bIn[3]’)

原理框图：



* 1. 能否用加法器实现比较器，若能，请画出原理框图。

用加法器计算两数之差，a+~b+1，分析不同情况下的输出结果：

a<b时，最高进位为0，且差不为0，例如a=0001，b=0010，out=01111;

a=b时，最高进位为1，差为0，例如a=b=0010，out=10000;

a>b时，最高进位为1，差不为0，例如a=0010，b=0001，out=10001;

转化为逻辑符号，

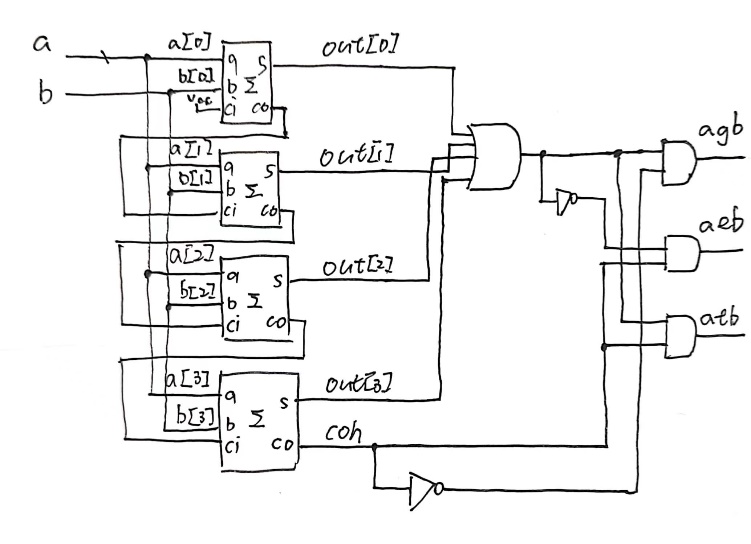
agb=coh’(out[0]+out[1]+out[2]+out[3])

aeb=coh(out[0]’out[1]’out[2]’out[3]’)

alb=coh(out[0]+out[1]+out[2]+out[3])

其中coh是最高位的进位输出，out是每位加法器的本为和输出。

原理框图：



* 1. 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？

参数传递有两种格式，

* 1. 在线显示重载参数方式：

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)

* 1. 隐式参数传递方式：

模块名 #(参数1值，参数2值，…) 例化模块名(端口列表)；

使用第二种参数传递方法，就要注意传递多个参数时的对应问题，其位置要与模块定义时对应。