

实验 5 常用组合电路模块的设计和应用

一、实验目的

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

二、实验任务

(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(Mode-Dependent Comparator)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， $y=MAX(a,b)$ ；而当 $m=1$ 时，则 $y=MIN(a,b)$ 。要求用多层次结构设计电路，即调用数据选择器和比较器等基本模块来设计电路。

三、实验原理与顶层设计

I.任务一 两数之差的绝对值电路的设计

(1) 电路的总体设计

两个无符号数之差的绝对值电路可由式(1)计算。由于电路难以直接进行减法运算，但是可由补码加法完成减法运算，因此采用式(2)计算两数之差的绝对值。

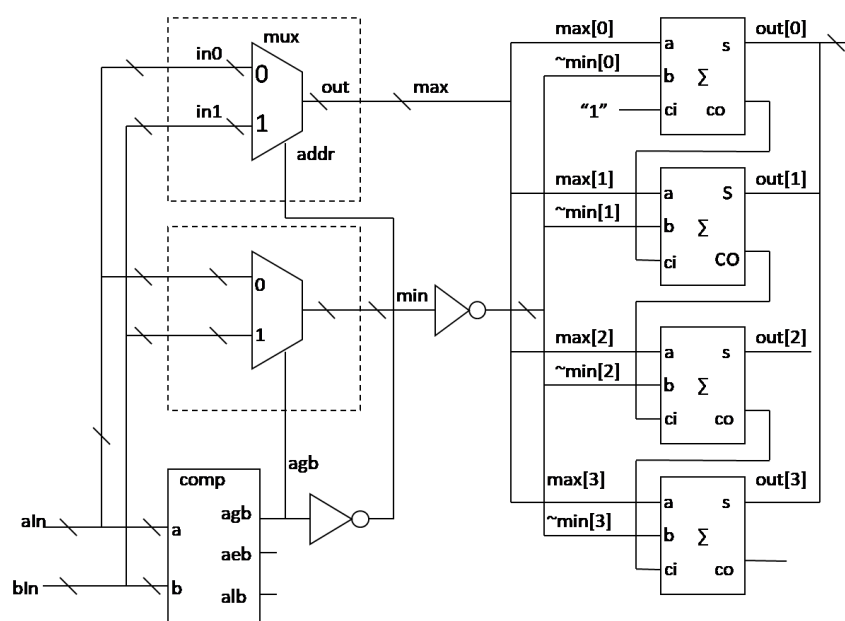
$$Out = Max(a_{in}, b_{in}) - Min(a_{in}, b_{in}) \quad (1)$$

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

根据计算式可以得到该任务数字电路设计的总体思路：

1. 首先使用数值比较器比较两输入数大小，比较结果 agb 用于控制两个数据选择器；
2. 再使用数据选择器分别选出输入两数的最大值 Max 和最小值 Min ；
3. 最后由全加器组成的四位加法器实现 Max 与 $(-Min)$ 的补码加法运算。

根据计算式可画出两数之差的绝对值电路的原理框图如下：



(2) 顶层模块的 Verilog HDL 描述

根据上述分析，编写该电路顶层设计的 Verilog HDL 代码：

```
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
```

根据顶层设计的 Verilog HDL 代码，可以根据自顶向下的数字设计方法，确定各个子模块的输入端口与参数定义，从而编写各个子模块的 Verilog HDL 代码。

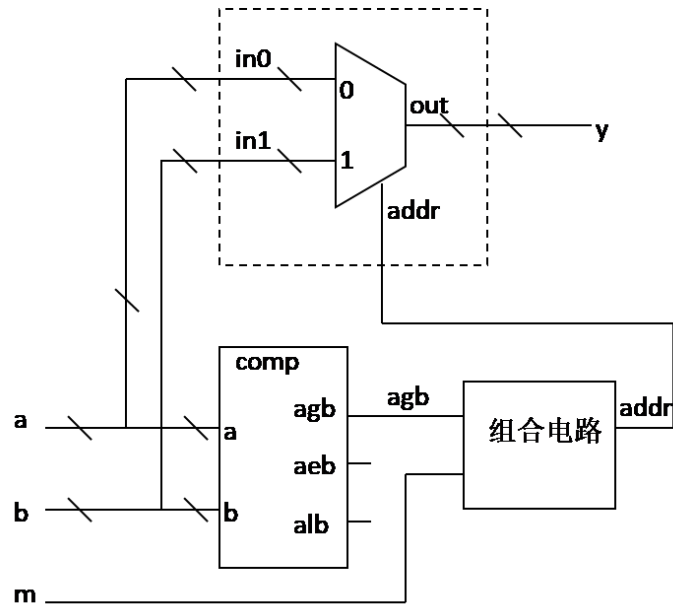
II.任务二 模式比较器电路的设计

(1) 电路的总体设计

模式比较器电路的顶层设计方法与任务一类似，设计调用数据比较器与数据选择器等子模块来实现模式比较功能，该数字电路的总体设计思路如下：

- 1. 首先使用数值比较器比较两输入数大小，比较结果 agb 用于控制后继组合电路；
- 2. 再使用组合电路子模块对 agb 与模式控制信号 m 进行处理，输出 addr 作为控制数据选择器数据选择的地址信号；
- 3. 数据选择器根据地址信号进行输出。

根据上述分析画出模式比较器的原理框图如下：



依据该电路需要实现的功能，可以得到产生 addr 组合电路的功能表如下：

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 取大值 a
1	1	1	当 m=1, 且 $a > b$ 时, y 取大值 b

从功能表可以看出，addr 产生电路为一个同或门。

(2) 顶层模块的 Verilog HDL 描述

根据上述分析，编写该电路顶层设计的 Verilog HDL 代码：

```
module ModeComparator(a,b,m,y);
    input m;
    input [7:0] a,b;
    output [7:0] y;
    wire agb;
    comp #(.n(8))compare1(.a(a),.b(b),.agb(agb),.aeb(),.alb());
    wire addr;
    xnor_gate xnor1(.a(agb),.b(m),.out(addr));
    mux_2to1 #(.n(8))mux(.out(y),.in0(a),.in1(b),.addr(addr));
endmodule
```

同时为了方便功能实现，我们额外编写一个同或门的子模块：

```
module xnor_gate(a,b,out);
    input a,b;
    output reg out;
    always @ (*)begin
        case({a,b})
            2'b00:out=1;
            2'b01:out=0;
            2'b10:out=0;
            2'b11:out=1;
        endcase
    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、实验步骤与仿真分析

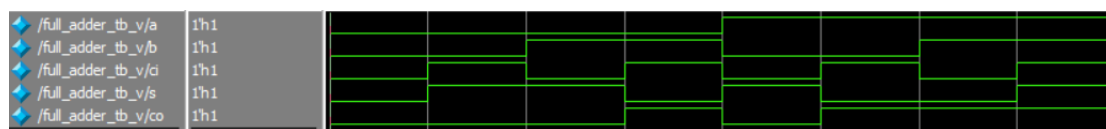
I.任务一 两数之差的绝对值电路的设计

- (1) 编写一位全加器的 Verilog HDL 代码，并用 ModelSim 软件进行功能仿真；

全加器子模块的 Verilog HDL 代码：

```
module full_adder(a,b,s,ci,co);
    input a,b,ci;
    output s,co;
    assign {co,s}=a+b+ci;
endmodule
```

ModelSim 软件仿真波形：



仿真结果分析：

a	b	ci	s	co
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0

1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

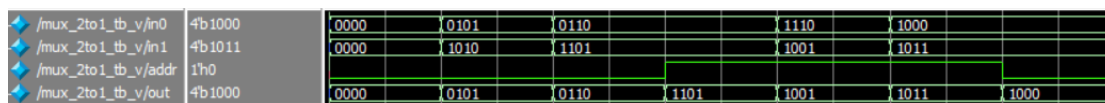
仿真波形图从上到下一次代表输入信号 a、输入信号 b、输入进位 ci、输出信号 s 和输出进位 co，通过波形图获得真值表，由真值表可知 Verilog HDL 代码编写的全加器模块成功实现功能。

(2) 编写 n 位二选一数据选择器的 Verilog HDL 代码，并用 ModelSim 进行功能仿真；

n 位二选一数据选择器子模块的 Verilog HDL 代码：

```
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
```

ModelSim 软件仿真波形：



仿真结果分析：

In0	In1	addr	out
0000	0000	0	0000
0101	1010	0	0101
0110	1101	0	0110
0110	1101	1	1101
1110	1001	1	1001
1000	1011	1	1011
1000	1011	0	1000

仿真波形图从上到下依次为输入信号 In0、输入信号 In1、输入地址信号 addr 和输出信号 out，由列出的仿真结果可知，当 addr=0 时，输出信号 out=In0；当 addr=1 时，输出信号 out=In1。而输入数据的位数可由参数 parameter 进行修改。所以 Verilog HDL 代码编写的二选一数据选择器模块成功实现功能。

(3) 编写 n 位比较器的 Verilog HDL 代码，并用 ModelSim 软件进行功能仿真；

n 位比较器的 Verilog HDL 代码：

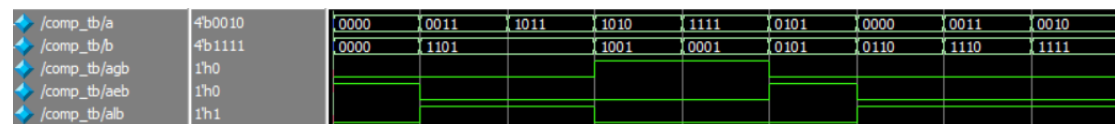
```
module comp(a,b,agb,aeb,alb);
    parameter n=1;
```

```

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

```

ModelSim 软件仿真波形:



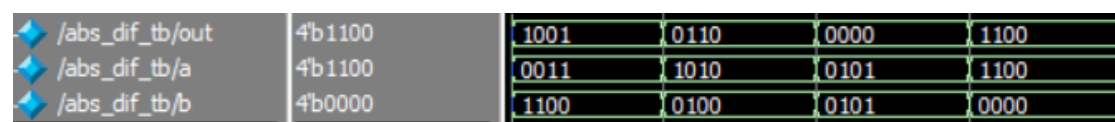
仿真结果分析:

a	b	agb	aeb	alb
0000	0000	0	1	0
0011	1101	0	0	1
1011	1101	0	0	1
1010	1001	1	0	0
1111	0001	1	0	0
0101	0101	0	1	0
0000	0110	0	0	1
0011	1110	0	0	1
0010	1111	0	0	1

仿真波形图从上到下依次为输入信号 a、输入信号 b、输出信号 aeb、输出信号 aeb 和输出信号 alb，由列出的仿真结果可知，当 a 大于 b 时 agb=1，当 a=b 时 aeb=1，当 a<b 时，alb=1。而输入数据的位数可由参数 parameter 进行修改。所以 Verilog HDL 代码编写的 n 位比较器模块成功实现功能。

(4) 对两数之差的绝对值电路进行功能仿真;

通过两数之差的绝对值电路顶层设计调用数据比较器、数据选择器和全加器等模块，实现对两个 4 位无符号二进制数的差的绝对值的求解，顶层设计及子模块的 Verilog HDL 代码均在前文给出，不再赘述。ModelSim 仿真波形图如下:



仿真结果分析:

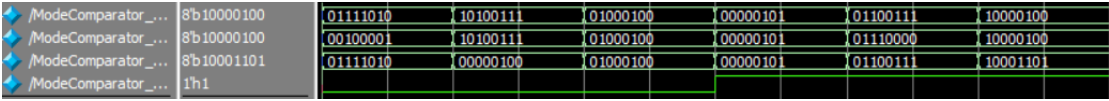
a	b	out
0011	1100	1001
1010	0100	0110
0101	0101	0000
1100	0000	1100

仿真波形图从上到下依次位输出信号 out，输入信号 a 和输入信号 b，根据表中列出的仿真结果可知，本次实验编写的两数之差的绝对值电路顶层设计及其子模块 Verilog HDL 代码成功实现了计算两个 4 位无符号二进制数的差的绝对值的功能。

II.任务二 模式比较器电路的设计

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

通过模式比较器顶层设计调用数据比较器、数据选择器和同或门等模块，实现对两个 8 位无符号二进制数的模式控制比较，顶层设计及新增的 xnor_gate 子模块的 Verilog HDL 代码均在前文给出，不再赘述。ModelSim 仿真波形图如下：



仿真结果分析：

a	b	m	out
00100001	01111010	0	01111010
10100111	00000100	0	10100111
01000100	01000100	0	01000100
00000101	00000101	1	00000101
01110000	01100111	1	01100111
10000100	10001101	1	10000100

仿真波形图从上到下依次为输出信号 out、输入信号 a、输入信号 b 和模式控制信号 m，根据仿真结果可知，当 m=0 时，输出为输入信号中较大的数；当 m=1 时，输出为输入信号中较小的数。因此本次实验编写的模式比较器电路顶层设计及其子模块 Verilog HDL 代码成功实现了模式比较的功能。

六、开发实验板设计验证

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。

使用 Nexys Video 开发板对两数之差的绝对值电路设计进行验证，假定左边四个逻辑开关的输入代表 a，右边四个逻辑开关的输入代表 b，验证结果如下：

①当输入两个相同大小的 4 位无符号二进制数时，输出均为 0：

输入 a=1010, b=1010, 输出 0000



输入 a=0101, b=0101, 输出 0000



②当输入信号 a 大于输入信号 b 时，输出结果为 a-b：

输入 a=1100, b=0011, 输出 1001



输入 a=1010, b=0101, 输出 0101



③当输入信号 a 小于输入信号 b 时，输出结果为 b-a：

输入 a=1000, b=1001, 输出 0001



输入 a=0100, b=1001, 输出 0101



分析以上实验数据可知，任务一设计的数字电路成功实现了对两数之差的绝对值的计算。

七、实验中遇到的问题及解决方法

(1) 由于初次使用 Verilog HDL 语言进行数字电路设计，代码编写过程中难免出错，例如分号的使用、“endmodule”等语句的疏漏都曾发生过。通过 ModelSim 对代码编译的报错提示对代码进行修改与完善。

(2) 由于不熟悉 Verilog HDL 语言中的位运算操作符，在实现模式比较器中的同或门组合模块的时候，没有使用简单的 assign 语句直接实现。通过另外编写一个 xnor_gate 子模块，并通过顶层设计对同或门子模块进行调用实现功能。

(3) 使用过程中出现了 ModelSim 无法正常打开的故障。通过重新复制破解文件、运行批处理生成许可证、设置环境变量解决软件问题。

八、思考题

原理框图不便绘制，故仅语言描述。

(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？

利用比较器判断两个补码数的大小（补码比较时需先判断符号位，再比较数值位），然后输出 max-min，其中减法的实现方式与无符号数相同。由于补码表示下 max 和 min 的真值大小关系与无符号比较不同，需设计有符号比较器。得到 max 和 min 后，使用加法器计算 $\max + (\sim \min + 1)$ ，结果即为绝对值的补码形式。

(2) 能否用加法器实现比较器？

计算 $a-b$ ，通过结果的符号位判断 a 与 b 的大小关系。对于无符号数，使用加法器计算 $a + (\sim b + 1)$ ，若最高位进位 co 为 1，则 $a \geq b$ ；若为 0，则 $a < b$ 。对于有符号数，计算 $a-b$ 后，结果的最高位（符号位）为 0 表示 $a \geq b$ ，为 1 表示 $a < b$ 。因此用一个加法器配合简单的逻辑门即可得到比较结果。

(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？

在实例化时使用 $\#()$ 传递参数，显式指定参数名和值，可避免顺序错误，提高可读性。

例如：模块名 $\#(\text{参数名 1}(\text{值 1}), \text{参数名 2}(\text{值 2}), \dots)$ 实例名 (端口连接)；

当模块有多个参数时，实例格式如下：

模块 `module example #(parameter WIDTH=8, parameter DEPTH=16) ( ... );`

实例化 `example #(.WIDTH(4), .DEPTH(32)) inst1 ( ... );`