

浙江大学

本科实验报告

课程名称： 数字逻辑设计

姓 名： 蒋奕

学 院： 计算机学院

系： 计算机系

专 业： 计算机科学与技术

学 号： 3210103803

指导教师： 马德

2022 年 11 月 14 日

浙江大学实验报告

课程名称: 数字逻辑设计 实验类型: 绘图实验+模块调用

实验项目名称: 全加器的设计实现与加减法器和 ALU 基本原理与设计

学生姓名: 蒋奕 专业: 计算机科学与技术 学号: 3210103803

同组学生姓名: 任庭旭 指导老师: 马德

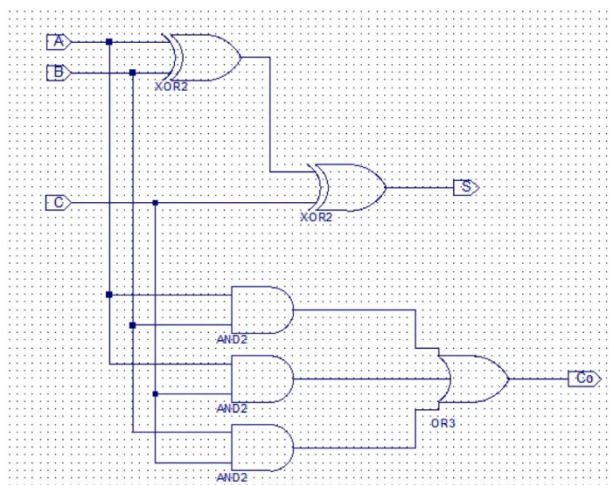
实验地点: 东 4-509 实验日期: 2022 年 11 月 14 日

一、实验目的:

- ① 掌握一位全加器的工作原理和逻辑功能
- ② 掌握串行进位加法器的工作原理和进位延迟
- ③ 掌握减法器的实现原理
- ④ 掌握加减法器的设计方法
- ⑤ 掌握 ALU 基本原理及在 CPU 中的作用与 ALU 的设计方法

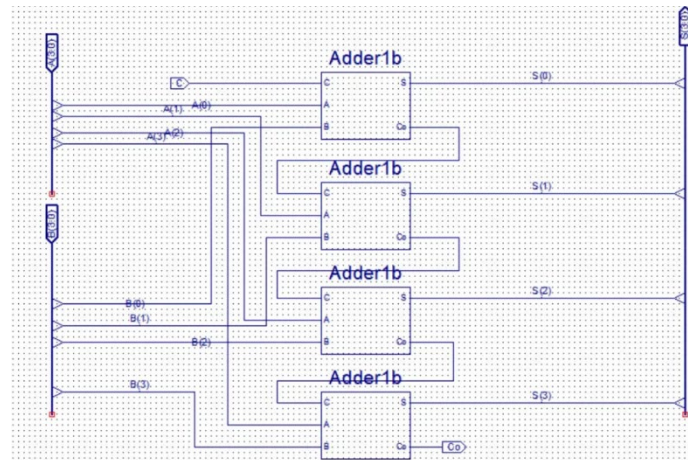
实验一: 原理图方式设计 4 位加减法器

新建工程, 命名为 MyAdder, top-level source type 为 HDL。新建类型为 schematic 的源文件, 命名为 Adder1b。用原理图方式设计, 具体如下所示



生成逻辑符号和.vf 文件: 点击 Process 窗口下 Design Utilities -> Create schematic symbol, 在工程文件夹里可以找到相应的.sym 文件。

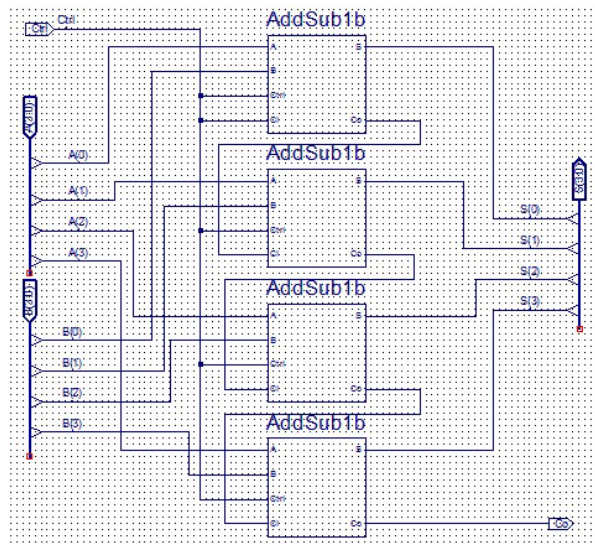
新建类型为 schematic 的源文件，命名为 Adder4b。调用 Adder1b 元件，用原理图方式设计，具体如下所示。



生成逻辑符号和.vf 文件：点击 Process 窗口下 Design Utilities -> Create schematic symbol，在工程文件夹里可以找到相应的.sym 文件。后进行波形仿真，激励输入至少 4 组。Check Syntax 无误后，点击 Process 窗口下 Design Utilities -> Create schematic symbol 生成逻辑符号和.vf 文件。

绘制加减法器原理图

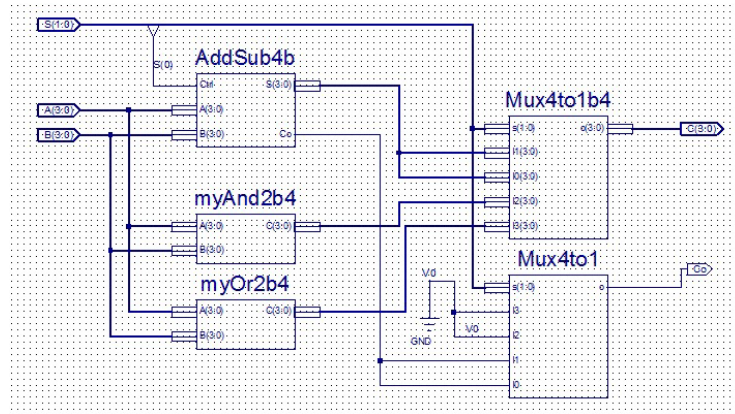
新建类型为 schematic 的源文件，命名为 MyAdder。用原理图方式实现加减法器的逻辑功能，具体如下



实验二：实现 4 位 ALU 及应用设计

工程名称用 MyALU。Top Level Source Type 用 HDL。新建源文件类型是 Verilog，文件名 Top。右键设为“Set as Top Module”。

新建源文件类型是 Schematic，文件名称用 myALU，用原理图方式设计 4 位 ALU，进行波形仿真，激励输入至少 4 组，覆盖 4 种操作
原理图如下所示：



两个 4 位操作数 A, B, 用按钮 BTNX4Y0 、BTNX4Y1 分别加或减 A、B, 具体是加还是减操作由 SW(1:0)控制, SW(15:14)是 4 位 ALU 的功能选择引脚, 分别选择选择加、减、与、或操作

SW(15:14) = 00: $C = A + B$

SW(15:14) = 01: $C = A - B$

SW(15:14) = 10: $C = A \& B$

SW(15:14) = 11: $C = A | B$

ALU 计算得到进位 Co 和结果 C, 4 位数码管, 分别表示 A、B、进位 Co 和结果 C。显示顺序要严格对应。

顶层模块 Top 设计

在 Top.V 中用行为描述进行设计

实例化 pbdebounce 模块对 2 个按键进行去抖

实例化 AddSub4b 模块实现 4 位加减法

实例化 clkdiv 模块, 提供 1ms 时钟

用 num[3:0]表示 A, 用 num[7:4]表示 B

实例化 CreateNumber 模块, 用 2 个按键对 num[7:4]、num[3:0]自增或自减

实例化 DispNum 模块, 显示 A、B、C0、C

根据设计修改 UCF

UCF 引脚定义

输入:

时钟: clk

按键控制输入: BTN[1]控制 A, BTN[0]控制 B, 关联到 BTNX4Y[0:1]

按键加/减方向控制: SW1[1]控制 A, SW1[0]控制 B, 关联到 DSW[1:0]

ALU 运算控制: SW2[1:0], 00-加, 01-减, 10-与, 11-或, 关联到 DSW[15:14]

输出:

数码管[0]: A - num[3:0]

数码管[1]: B - num[7:4]

数码管[2]: Co - Co

数码管[3]: C - C

BTNX4: 按键使能输出

设计顶层模块

新建源文件 top，在右键菜单里设为“Top Module”，Verilog 代码如下：

```
module Top(
    input wire clk,
    input wire [1:0]btn,
    input wire [1:0]SW,
    input wire [1:0]SW2,
    output wire [3:0]AN,
    output wire [7:0]SEGMENT,
    output wire BTNX4
);
wire [15:0] num;
wire [1:0] btn_out;
wire [3:0] C;
wire Co;
wire [31:0] clk_div;
wire [15:0] disp_hexs;
assign disp_hexs[15:12] = num[3:0]; //A
assign disp_hexs[11:8] = num[7:4]; //B
assign disp_hexs[7:4] = {3'b000, Co};
assign disp_hexs[3:0] = C[3:0];
pbdebounce m0(clk_div[17], BTN [0], btn_out[0]);
pbdebounce m1(clk_div[17], BTN [1], btn_out[1]);
clkdiv m2(.clk(clk),.rst(1'b0),.clkdiv(clk_div));
CreateNumber m3(.btn(btn_out[1:0]), .sw(SW[1:0]), .num(num));
MyALU
m5(.A(num[3:0]), .B(num[7:4]), .S(SW2[1:0]), .C(C[3:0]), .Co(Co));
DispNum m6(.clk(clk), .HEXS(disp_hexs), .LES(4'b0), .points(4'b0),
.RST(1'b0), .AN(AN), .Segment(SEGMENT));
assign BTNX4 = 1'b0; //Enable button inputs
endmodule
```

设计 CreateNumber 按键数据输入模块，Verilog 代码如下：

```
module CreateNumber(
    input wire [1:0] btn,
    input wire [1:0] sw,
    output reg [7:0] num
);
wire [3:0] A1,B1;
initial num <= 8'b 1010_1011;
myAddSub4b a1(.A(num[3:0]), .B(4'b0001), .Ctrl(sw[0]), .S(A1));
myAddSub4b b1(.A(num[7:4]), .B(4'b0001), .Ctrl(sw[1]), .S(B1));
always@(posedge btn[0]) num[ 3: 0] <= A1;
always@(posedge btn[1]) num[ 7: 4] <= B1;
endmodule
```

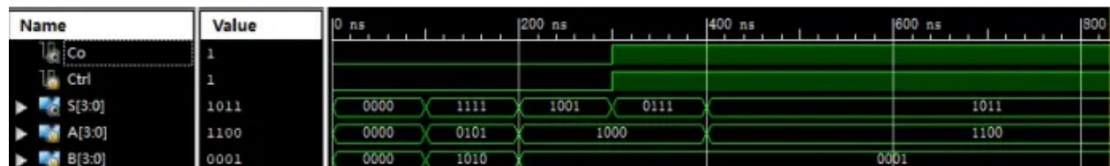
防抖动模块+ 分频器

```
module pbdebounce(  
    input wire clk_1ms,  
    input wire button,  
    output reg pbreg  
);  
reg [7:0] pbshift;  
always@(posedge clk_1ms) begin  
    pbshift<=pbshift<<1;  
    pbshift[0]<=button;  
    if (pbshift==8'b0)  
        pbreg<=0;  
    if (pbshift==8'hFF)  
        pbreg<=1;  
end  
endmodule
```

之后建立 k7.ucf 进行验证

二、实验数据记录和处理

实验一：原理图方式设计 4 位加减法器



实验仿真波形如上

对应的 Verilog 仿真代码如下：

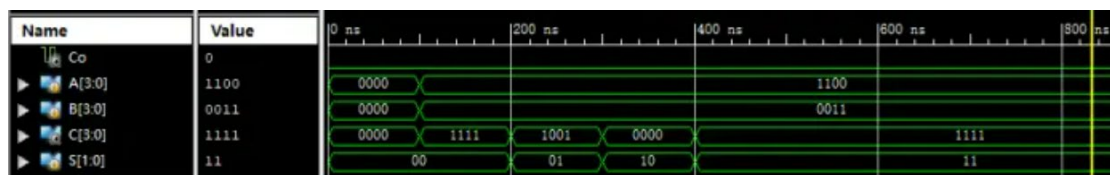
```
`timescale 1ns / 1ps  
module myAddSub4b_myAddSub4b_sch_tb();  
    reg Ctrl;  
    reg [3:0] A;  
    reg [3:0] B;  
    wire Co;  
    wire [3:0] S;  
    myAddSub4b UUT (  
        .Co(Co),  
        .S(S),  
        .Ctrl(Ctrl),  
        .A(A),
```

```

        .B(B)
    );
    initial begin
        Ctrl = 0;
        A = 0;B = 0;
        #100
        Ctrl = 0;
        A = 4'b0101;
        B = 4'b1010;
        #100
        Ctrl = 0;
        A = 4'b1000;
        B = 4'b0001;
        #100
        Ctrl = 1;
        A = 4'b1000;
        B = 4'b0001;
        #100
        Ctrl = 1;
        A = 4'b1100;
        B = 4'b0001;
    end
endmodule

```

实验二：实现 4 位 ALU 及应用设计



实验仿真波形如上

对应的 Verilog 仿真代码如下：

```

`timescale 1ns / 1ps
module myALU_myALU_sch_tb();
    reg [3:0] A;
    reg [3:0] B;
    reg [1:0] S;
    wire [3:0] C;
    wire Co;
    myALU UUT (
        .C(C),
        .Co(Co),
        .A(A),
        .B(B),

```

```

        .S(S)
    );
    initial begin
        A = 0;B = 0;S = 0;
        #100
        S = 2'b00;A = 4'b1100;B = 4'b0011;
        #100
        S = 2'b01;A = 4'b1100;B = 4'b0011;
        #100
        S = 2'b10;A = 4'b1100;B = 4'b0011;
        #100
        S = 2'b11;A = 4'b1100;B = 4'b0011;
    end
endmodule

```

K7. ucf 如下

```

NET "clk" LOC=AC18 | IOSTANDARD=LVC MOS18;
NET "SW[0]" LOC = AA10 | IOSTANDARD = LVC MOS15;#POINT
NET "SW[1]" LOC = AB10 | IOSTANDARD = LVC MOS15;
NET "SW2[0]" LOC = AA13 | IOSTANDARD = LVC MOS15;
NET "SW2[1]" LOC = AA12 | IOSTANDARD = LVC MOS15;
NET "SEGMENT[0]" LOC = AB22 | IOSTANDARD = LVC MOS33 ;#a
NET "SEGMENT[1]" LOC = AD24 | IOSTANDARD = LVC MOS33 ;#b
NET "SEGMENT[2]" LOC = AD23 | IOSTANDARD = LVC MOS33 ;
NET "SEGMENT[3]" LOC = Y21 | IOSTANDARD = LVC MOS33 ;
NET "SEGMENT[4]" LOC = W20 | IOSTANDARD = LVC MOS33 ;
NET "SEGMENT[5]" LOC = AC24 | IOSTANDARD = LVC MOS33 ;
NET "SEGMENT[6]" LOC = AC23 | IOSTANDARD = LVC MOS33 ;#g
NET "SEGMENT[7]" LOC = AA22 | IOSTANDARD = LVC MOS33 ;#point
NET "AN[3]" LOC = AC22 | IOSTANDARD = LVC MOS33 ;
NET "AN[2]" LOC = AB21 | IOSTANDARD = LVC MOS33 ;
NET "AN[1]" LOC = AC21 | IOSTANDARD = LVC MOS33 ;
NET "AN[0]" LOC = AD21 | IOSTANDARD = LVC MOS33 ;
NET "BTN[1]" LOC = V14 | IOSTANDARD = LVC MOS18 ;
NET "BTN[0]" LOC = W14 | IOSTANDARD = LVC MOS18 ;
NET "BTNX4" LOC = V17 | IOSTANDARD = LVC MOS18 ;

```

注：该实验已经在 2022.11.14 晚由李一涛助教验收通过

三、实验结果与分析

实验结果和预测结果一样，实验操作正确无误，成功地实现了对灯的控制。

四、讨论、心得（选填）

实验过程中我体会到要认真对待每一次操作，正如我多次因为不小心而操作不熟练使得实验进程受阻。Top 文件 verilog 代码补全：接口位数要仔细核对，尽量使用标准语法，例如 `.A(num[3:0])`，结构更清晰，更容易看出错误，而不是 `{num[3:0]}`，项数多了以后容易搞错。同时，需要注意，同一元件的接口声明要保持语法一致。