

浙江大学

本科实验报告

课程名称： 数字逻辑设计

姓 名：

学 院： 计算机科学与技术学院

专 业： 计算机科学与技术

学 号：

指导教师： 马德

2021 年 12 月 27 日

浙江大学实验报告

课程名称：数字逻辑设计 实验项目名称：全加器、加减法器及 ALU 基本原理

学生姓名： 专业：计算机科学与技术 学号：

指导老师：马德 实验日期：2021 年 12 月 20 日

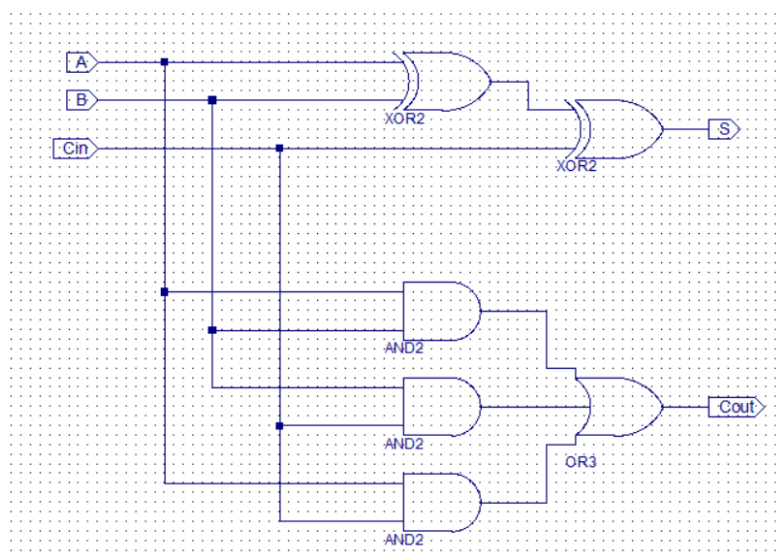
一、实验目的

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
6. 掌握 ALU 的设计方法

二、操作方法与实验步骤

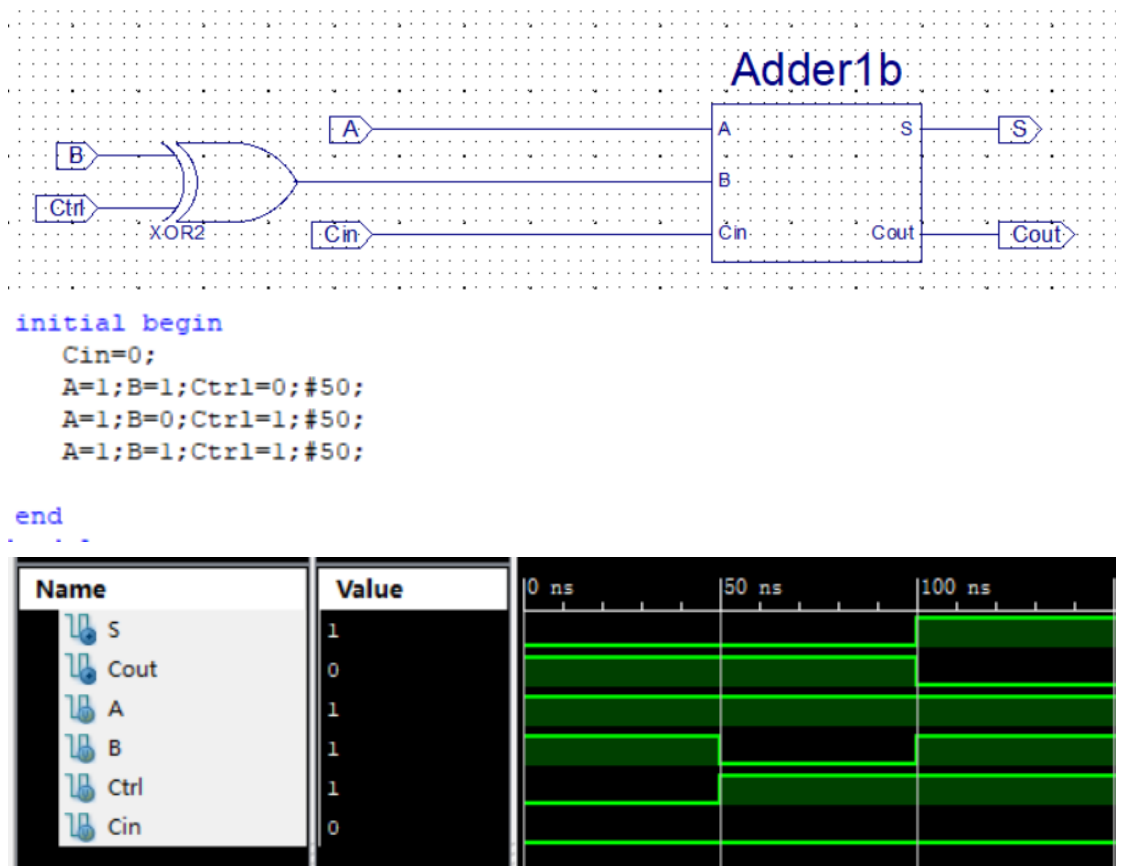
1. 4 位加减法器设计

- ① 建立 MyAdder 工程。采用如下电路图，绘制一位加法器 Adder1b 器件，并建立器件符号。

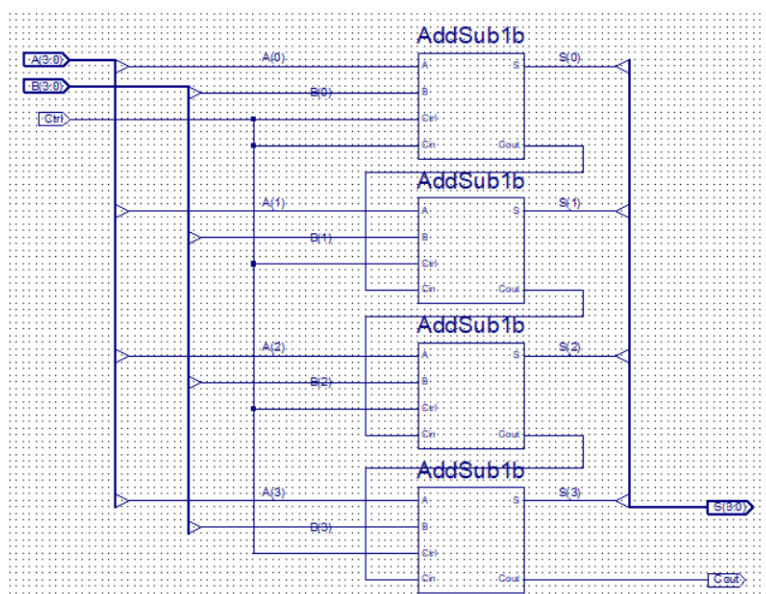


- ② 调用 Adder1b 模块，构建一位加减法器 AddSub1b 器件。通过仿真激

励进行功能验证。



- ③ 检查无误后，生成 AddSub1b 的器件符号，以其为基础，搭建四位加减法器 AddSub4b。

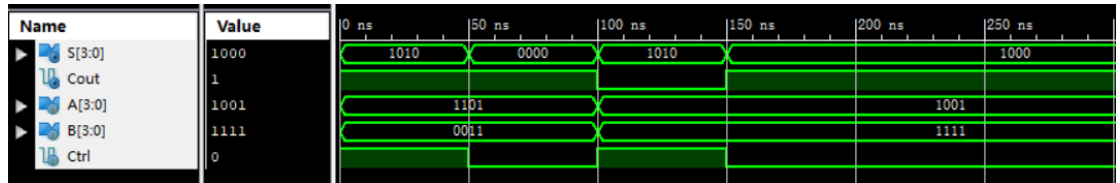


- ④ 通过仿真激励代码进行功能验证。调试无误后，生成逻辑器件。

```

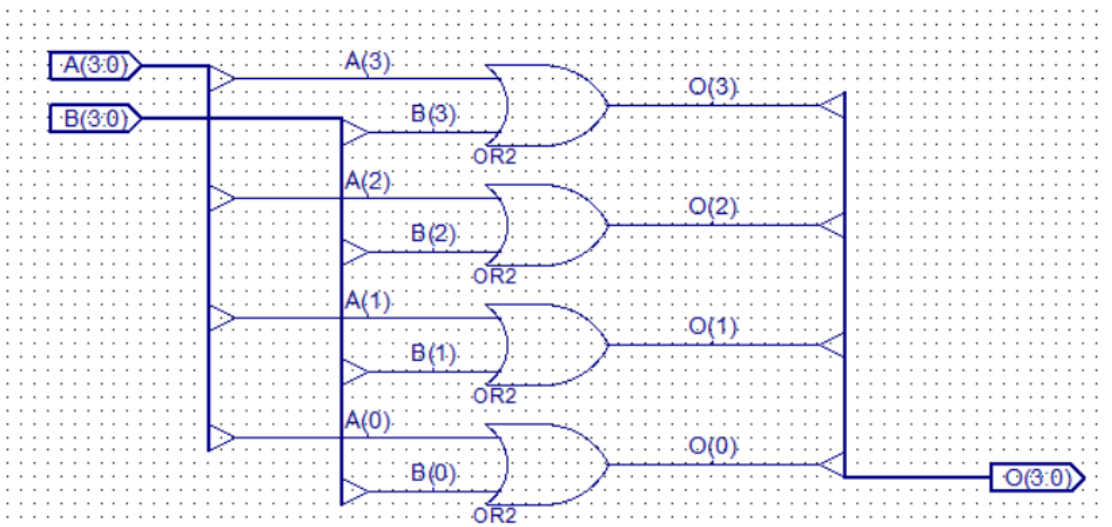
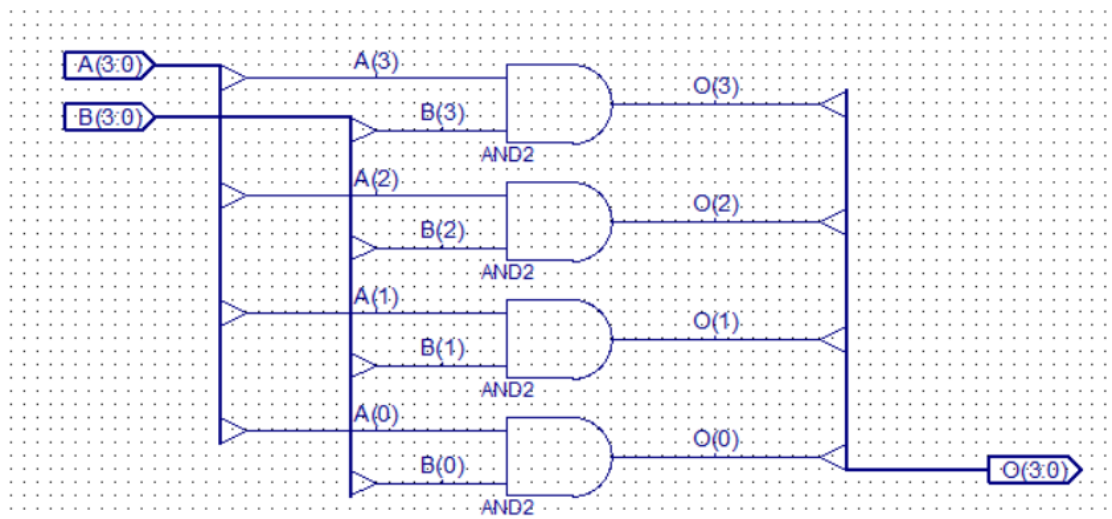
initial begin
    A=4'b1101;B=4'b0011;Ctrl=1;#50;
    A=4'b1101;B=4'b0011;Ctrl=0;#50;
    A=4'b1001;B=4'b1111;Ctrl=1;#50;
    A=4'b1001;B=4'b1111;Ctrl=0;#50;
end

```

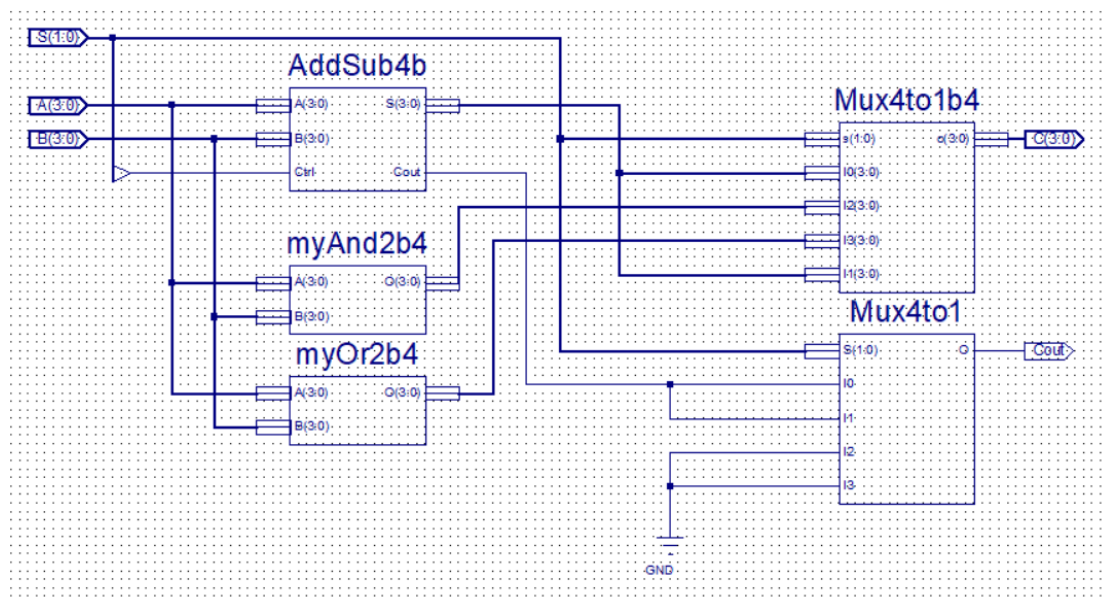


2. ALU 电路设计

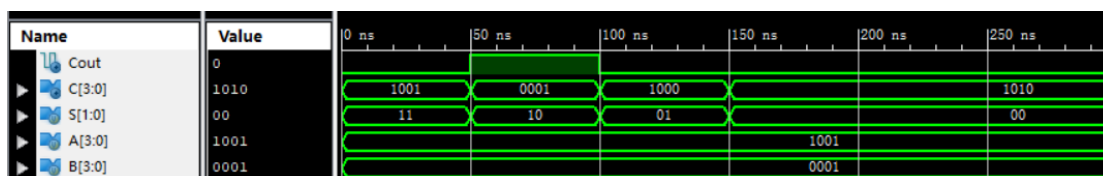
- ① 建立 myALU 工程，将 AddSub4b、AddSub1b、Adder1b 器件的相关文件复制到工程根目录。
- ② 通过原理图方式，绘制 myAnd2b4 及 myOr2b4 器件，验证后生成逻辑器件。



- ③ 建立 ALU 器件的原理图文件 myALU，采用已有器件绘制 ALU 原理图，并进行检查。通过仿真波形进行验证。



```
initial begin
    S=2'b11;A=4'b1001;B=4'b0001;#50;
    S=2'b10;A=4'b1001;B=4'b0001;#50;
    S=2'b01;A=4'b1001;B=4'b0001;#50;
    S=2'b00;A=4'b1001;B=4'b0001;#50;
end
```



- ④ 根据示例代码，设计顶层模块 Top 如下。

```
module top(
    input wire clk,
    input wire [1:0] BTN,
    input wire [1:0] SW1,
    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(SW1[1:0]),.num(num[15:0]));
    myALU m5(.A(disp_hexs[15:12]),.B(disp_hexs[11:8]),.S(SW2[1:0]),.C(C),.Cout(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
```

⑤ 编写引脚约束，将程序下载到实验板上进行验证。

```
1 NET "clk" LOC = AC18 | IOSTANDARD = LVCMOS18;
2 NET "clk" TNM_NET = TM_CLK;
3 TIMESPEC TS_CLK_100M = PERIOD "TM_CLK" 10 ns HIGH 50%;
4
5 NET "SW1[0]" LOC = AD10 | IOSTANDARD = LVCMOS15;
6 NET "SW1[1]" LOC = AD11 | IOSTANDARD = LVCMOS15;
7 NET "SW2[0]" LOC = Y12 | IOSTANDARD = LVCMOS15;
8 NET "SW2[1]" LOC = Y13 | IOSTANDARD = LVCMOS15;
9
10 NET "BTN[0]" LOC = W14 | IOSTANDARD = LVCMOS18;
11 NET "BTN[0]" CLOCK_DEDICATED_ROUTE = FALSE;
12 NET "BTN[1]" LOC = V14 | IOSTANDARD = LVCMOS18;
13 NET "BTN[1]" CLOCK_DEDICATED_ROUTE = FALSE;
14 NET "BTN4" LOC = W16 | IOSTANDARD = LVCMOS18;
15
16 NET "SEGMENT[0]" LOC = AB22 | IOSTANDARD = LVCMOS33;#a
17 NET "SEGMENT[1]" LOC = AD24 | IOSTANDARD = LVCMOS33;#b
18 NET "SEGMENT[2]" LOC = AD23 | IOSTANDARD = LVCMOS33;#c
19 NET "SEGMENT[3]" LOC = Y21 | IOSTANDARD = LVCMOS33;#d
20 NET "SEGMENT[4]" LOC = W20 | IOSTANDARD = LVCMOS33;#e
21 NET "SEGMENT[5]" LOC = AC24 | IOSTANDARD = LVCMOS33;#f
22 NET "SEGMENT[6]" LOC = AC23 | IOSTANDARD = LVCMOS33;#g
23 NET "SEGMENT[7]" LOC = AA22 | IOSTANDARD = LVCMOS33;#point
24
25 NET "AN[0]" LOC = AD21 | IOSTANDARD = LVCMOS33;
26 NET "AN[1]" LOC = AC21 | IOSTANDARD = LVCMOS33;
27 NET "AN[2]" LOC = AB21 | IOSTANDARD = LVCMOS33;
28 NET "AN[3]" LOC = AC22 | IOSTANDARD = LVCMOS33;
```

三、实验数据记录和处理

1. 4 位加减法器设计

经测试，通过原理图方式绘制的 AddSub4b 器件能够完成 4 位加减法器的功能。经模拟过程的波形验证，器件设计无误，可详见以上步骤。

2. ALU 电路设计

经过简单的编写及调试，在实验板上验收时，所有增减操作、加减与或操作、进位显示、按键效果等都能正常工作。

实验结果经验证无误。

四、实验结果与分析

1. 实验结果正常，经过验收。
2. 注意到按键防抖动模块的作用。通过加入按键防抖动模块，能够有效避免一次按键多次响应导致数据不可读情况的发生，但也存在一定按键不响应的情况。