

# 浙江大学

## 本科实验报告

课程名称：数字逻辑设计

姓名：沈一芑

学院：计算机学院

系：计算机系

专业：计算机科学与技术

学号：3220101827

指导教师：马德

2023 年 11 月 13 日

# 浙江大学实验报告

课程名称: 数字逻辑设计 实验类型: ISE 电路设计

实验项目名称: 多路选择器设计与应用

学生姓名: 沈一芃 专业: 计算机科学与技术 学号: 322010827

同组学生姓名: 无 指导老师: 马德

实验地点: 东 4 509 实验日期: 2023 年 11 月 13 日

## 一、实验目的和要求:

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

## 二、实验内容和原理

### 1. 实验内容:

- 原理图方式设计 4 位加减法器
- 实现 4 位 ALU 及应用设计

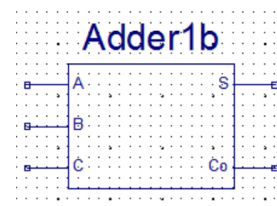
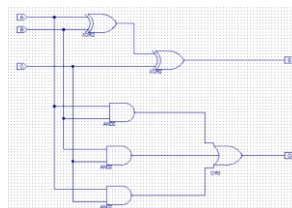
### 2. 实验原理:

- 1 位全加器

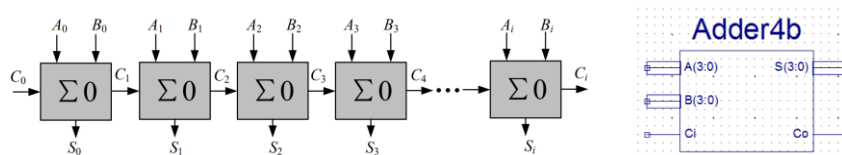
三个输入位: 数据位  $A_i$  和  $B_i$ , 低位进位输入  $C_i$

二个输出位: 全加和  $S_i$ , 进位输出  $C_{i+1}$

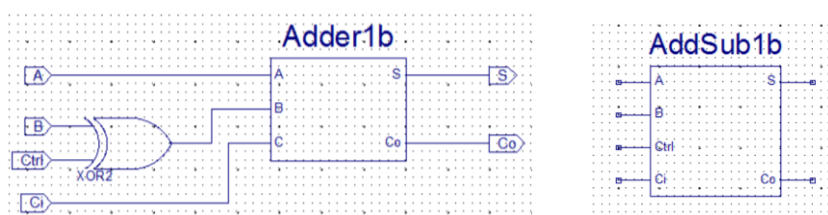
$A_i$	$B_i$	$C_i$	$S_i$	$C_{i+1}$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



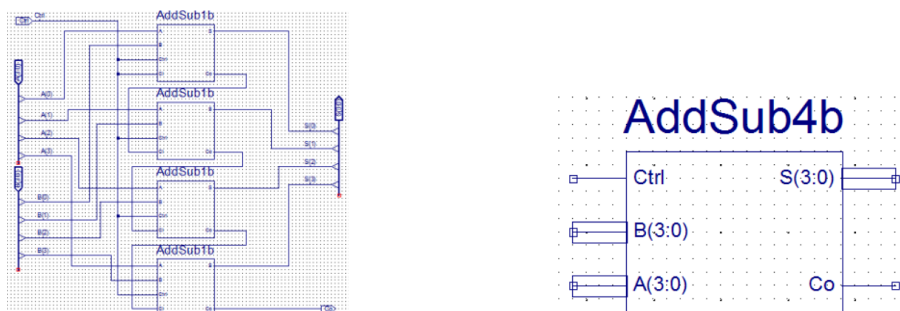
- 多位串行进位加法器  
由一位全加器将进位串接构成  
低位进位  $C_0$  为 0,  $C_i$  为高位进位输出



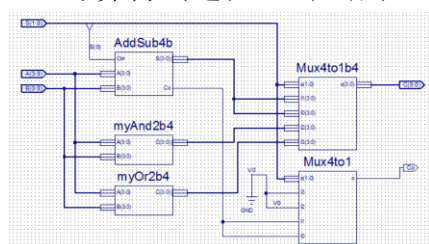
- 1 位加减法器  
用负数补码加法实现，减数当作负数求补码  
用“异或”门控制求反，低位进位  $C_0$  为 1  
其余用正常加法器实现



- 多位串行进位全减器  
用负数补码加法实现，减数当作负数求补码  
用“异或”门控制求反，低位进位  $C_0$  为 1  
其余共用加法器实现



- ALU 设计  
两个 4 位操作数  $A(3:0)$ ,  $B(3:0)$   
 $S(1:0)$  是 ALU 的功能选择引脚，分别选择选择加、减、与、或操作  
■  $S(1:0) = 00$ :  $C = A + B$       ■  $S(1:0) = 01$ :  $C = A - B$   
■  $S(1:0) = 10$ :  $C = A \& B$       ■  $S(1:0) = 11$ :  $C = A | B$   
ALU 计算得到进位  $Co$  和结果  $C(3:0)$



### 三、 主要仪器设备

#### 实验设备：

- 装有 Xilinx ISE 14.7 的计算机 1 台
- SWORD 开发板 1 套

### 四、 操作方法与实验步骤

#### 1. 数据选择器设计

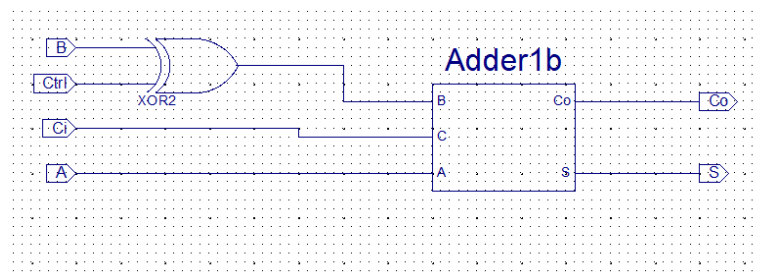
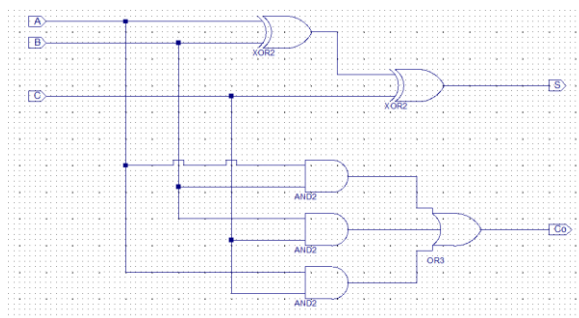
- 工程名称用 MyAdder。

Top Level Source Type 用 Schametic

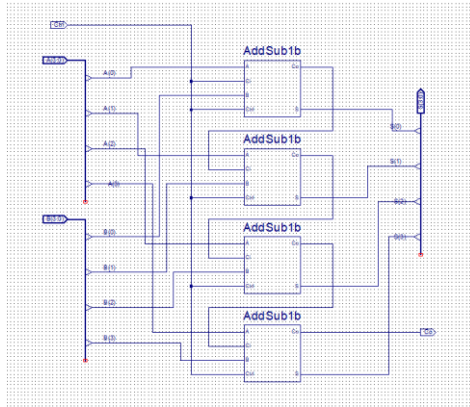
- 新建源文件

类型是 Schematic，文件名称用 AddSub1b。

原理图方式进行设计



- 创建 symbol
- 新建源文件
- 类型是 Schematic，文件名称用 AddSub4b
- 右键设为 “Set as Top Module”
- 原理图方式进行设计，调用前面设计的 AddSub1b



## 2. 计分板设计

- 新建工程

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”

- 设计顶层模块

```

21 module TOP(
22     input wire clk,
23     input wire [1:0]BTN,
24     input wire [1:0]SW1,
25     input wire [1:0]SW2,
26     output wire [3:0]AN,
27     output wire [7:0]SEGMENT,
28     output wire BINX4
29 );
30
31     wire [7:0]num;
32     wire [1:0]btn_out;
33     wire [3:0]C;
34     wire Co;
35     wire [31:0] clk_div;
36     wire [15:0] disp_hexs;
37
38     assign disp_hexs[15:12] = num[3:0];
39     assign disp_hexs[11: 8] = num[7:4];
40     assign disp_hexs[ 7: 4] = {3'b000,Co};
41     assign disp_hexs[ 3: 0] = C[3:0];
42
43     pbdebounce m0(clk_div[17], BTN[0], btn_out[0]);
44     pbdebounce m1(clk_div[17], BTN[1], btn_out[1]);
45
46     clkdiv m2(.clk(clk),.rst(1'b0),.clkdiv(clk_div));
47
48     CreateNumber m3(.btn(btn_out),.sw(SW1),.num(num));
49
50     myALU_SCH m5(.A(num[3:0]),.B(num[7:4]),.S(SW2),.C(C),.o(Co));
51
52     DispNum m6(.clk(clk),.HEXS(disp_hexs),.LES(4'b0),.points(4'b0),.RST(1'b0),.AN(AN),.Segment(SEGMENT));
53
54     assign BINX4 = 1'b0;
55
56 endmodule

```

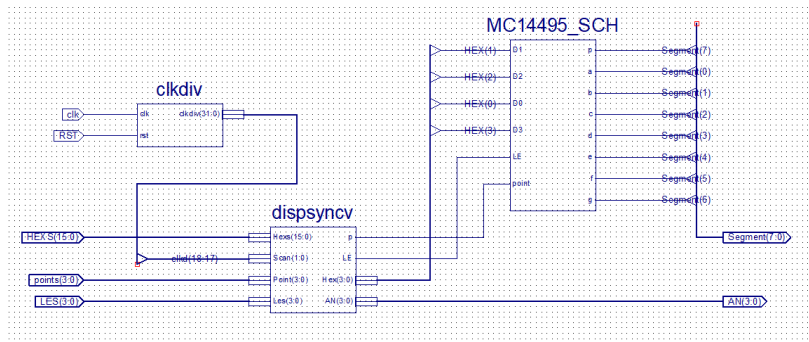
- 设计 CreateNumber 模块

```

21 module CreateNumber(
22     input wire [1:0] btn,
23     input wire [1:0] sw,
24     output reg [7:0] num
25 );
26     wire [3:0] A1, B1;
27
28     initial num <= 8'b1010_1011;
29
30     AddSub4b a1(.A(num[3:0]),.B(4'b0001),.Ctrl(sw[0]),.S(A1));
31     AddSub4b a2(.A(num[7:4]),.B(4'b0001),.Ctrl(sw[1]),.S(B1));
32
33     always@(posedge btn[0]) num[3:0]<= A1;
34     always@(posedge btn[1]) num[7:4]<= B1;
35
36 endmodule
37

```

- 设计时钟计数分频器，并建立 DispNum 模块



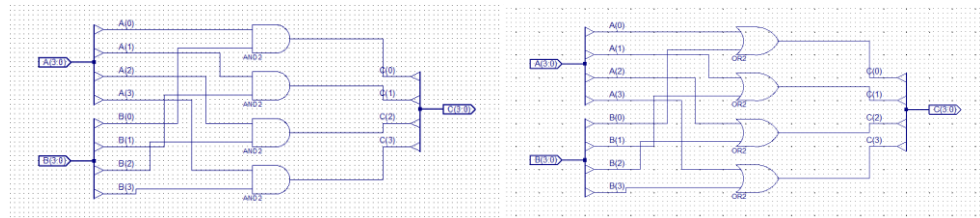
- 建立按键防抖模块

```

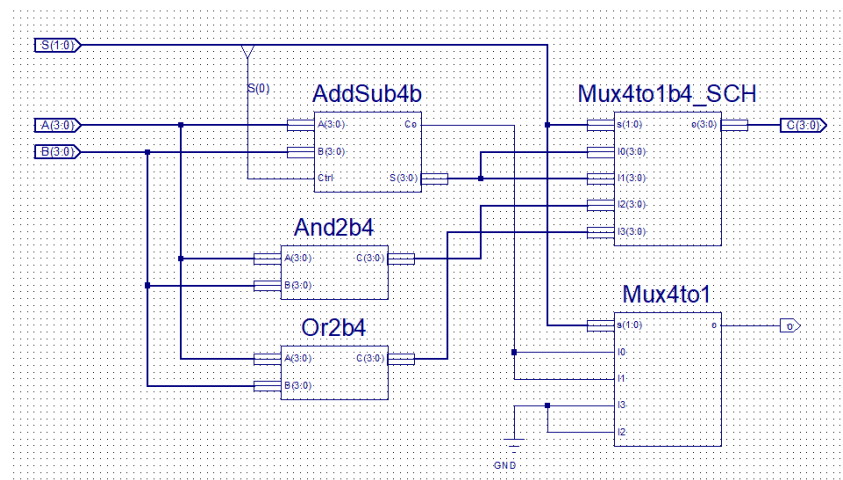
21 module pbdebounce(
22     input wire clk_lms,
23     input wire button,
24     output reg pbreg
25 );
26
27     reg [7:0] pbshift;
28
29     always@(posedge clk_lms) begin
30         pbshift=pbshift<<1;
31         pbshift[0]=button;
32         if(pbshift==8'b0)
33             pbreg=0;
34         if(pbshift==8'hFF)
35             pbreg=1;
36     end
37
38 endmodule

```

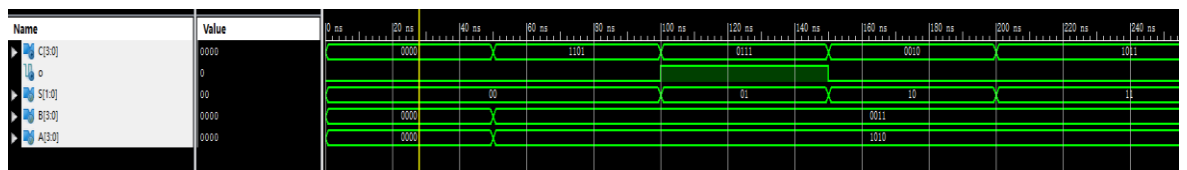
- 建立 andb4 模块与 orb4 模块



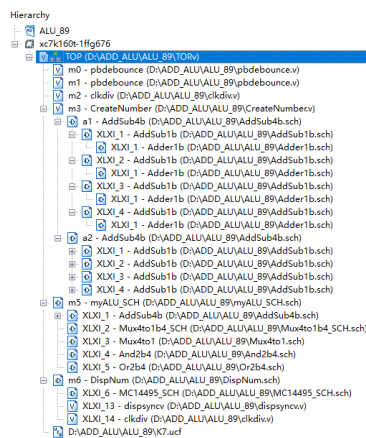
- ALU 模块如下:



- ALU 波形仿真如下:



- 总体文件结构如下:



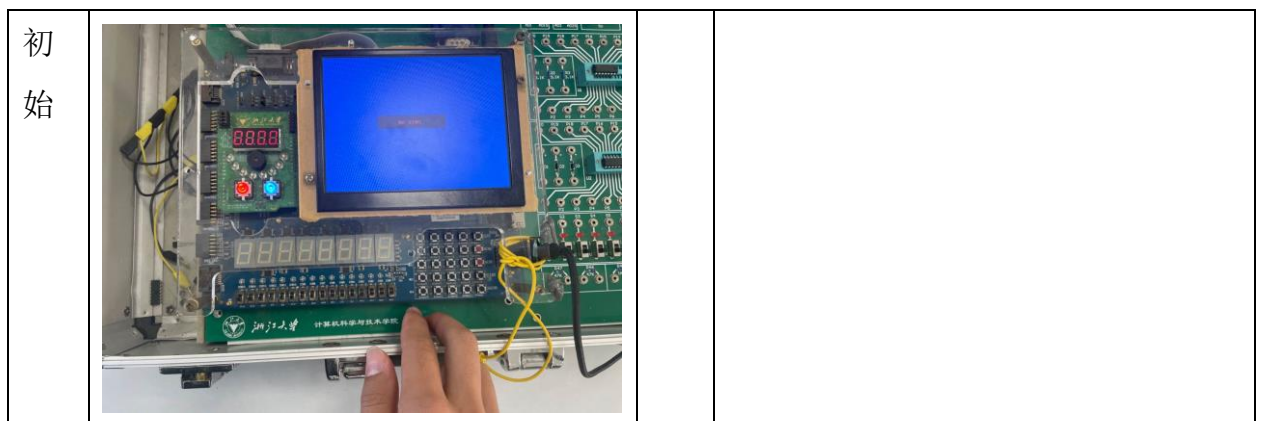
- 建立约束文件，生成执行文件

- 下载 bit 文件至 sword 板，根据真值表进行功能验证。

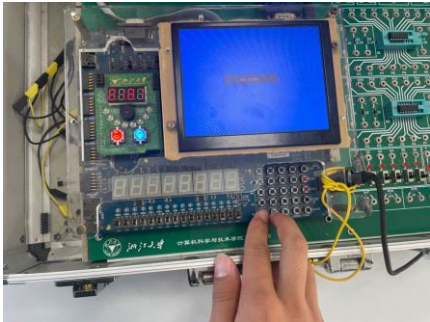


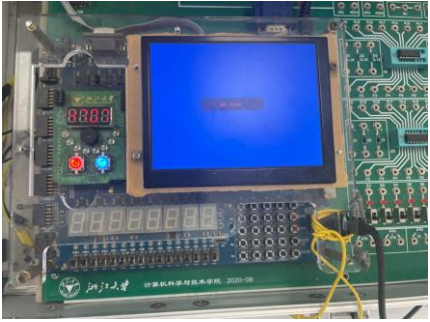


## 五、实验数据记录和处理

1. ALU 设计如下:

实验截图如下：





B+1		A+1	
与		或	
加		减	

## 六、 实验结果与分析

### 1. 数据选择器设计

0000	0000	0000	0000
按动 btn1	B+1	0101	正确
按动 btn0	A+1	1102	正确
00 加	0100+0111	1011	B 正确
01 减	0100-0111	1101	D 正确
10 与	0100&0111	0100	4 正确
11 或	0100 0111	0111	7 正确

功能正确。设计正确。



## 七、 讨论、心得

本次实验我理解了一位全加器的工作原理和逻辑功能，并就此搭建四位全加器。并在此基础上搭建了一位减法器 and 四位减法器。最终用上述模块搭建出自己的 ALU。

全加器是数字电路中的基本组件，它能够实现两个二进制数的加法运算，并考虑到进位的影响。全加器不仅可以用于加法运算，还可以通过一些简单的修改用于实现减法、乘法等其他运算。

通过对全加器的简单修改，我实现一个减法器，即：将减数取反并加 1（即求其二进制补码），然后将其与被减数输入到全加器中，从而实现减法运算。这种方法不仅简化了减法器的设计，还使得我们可以使用同样的硬件来实现加法和减法运算。

最后，我还学习了 ALU（算术逻辑单元）的基本原理及在 CPU 中的作用，以及 ALU 的设计方法。ALU 是计算机中的核心组件，它负责执行所有的算术运算和逻辑运算。在这个部分的实验中，我设计并实现了一个 4 位的 ALU。这个 ALU 可以执行加法、减法、与运算、或运算。

总的来说，这次实验让我深入理解了数字电路的基本原理和设计方法，对我今后的学习和研究将会有很大的帮助。我期待着在未来的实验中，能够设计和实现更为复杂的数字电路。

实验日期：2023.11.13