

浙江大学

本科实验报告

课程名称: 计算机逻辑设计基础

姓 名: 仇国智

学 院: 竺可桢学院

专 业: 计算机科学与技术

学 号: 3220102181

指导教师: 董亚波

2023 年 11 月 10 日

一、实验目的和要求

1 实验目的

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

2 实验任务

- 2.1 任务 1：原理图方式设计 4 位加减法器
- 2.2 任务 2：实现 4 位 ALU 及应用设计

二、实验内容和原理

1 实验内容

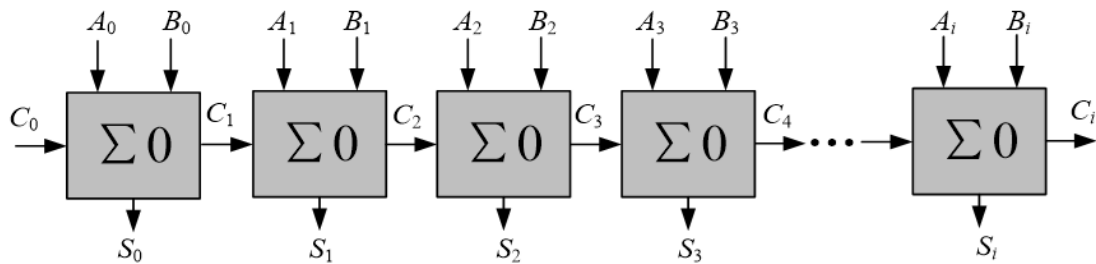
- 1.1 原理图方式设计 4 位加减法器
- 1.2 实现 4 位 ALU 及应用设计

2 实验原理

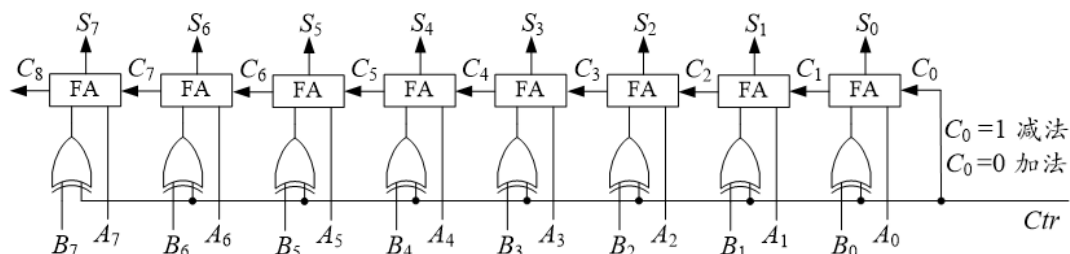
- 2.1 1 位全加器。三个输入位：数据位 A_i 和 B_i ，低位进位输入 C_i ；二个输出位：全加和 S_i ，进位输出 C_{i+1} 。根据真值表得 $S_i = A_i \oplus B_i \oplus C_i$ $C_{i+1} = A_i B_i + A_i C_i + B_i C_i$

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

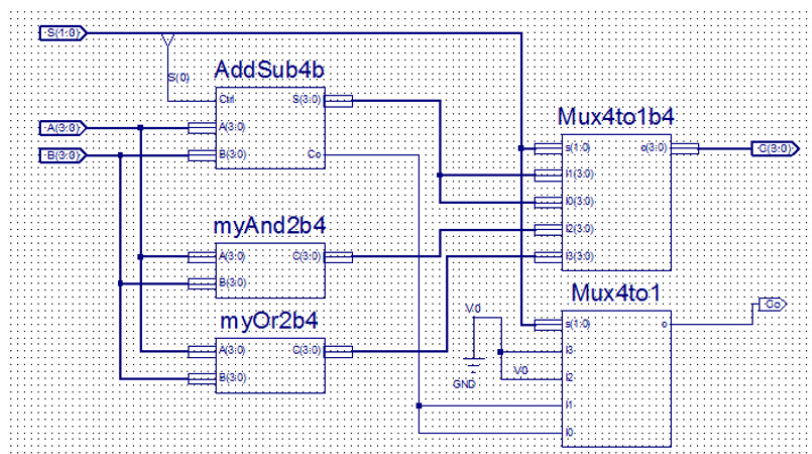
- 2.2 多位串行进位加法器：由一位全加器将进位串接构成，低位进位 C_0 为 0， C_i 为高位进位输出。



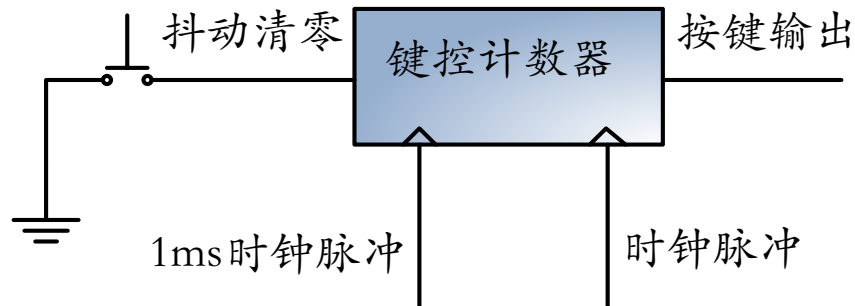
2.3 用负数补码加法实现，减数当作负数求补码；共用加法器，用“异或”门控制求反，低位进位 C_0 为 1。 $C_0=0$, $A+B$, $C_1=1$, $A-B$ 。



2.4 4 位 ALU 功能定义：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 C_o 和结果 $C(3:0)$ 。
myAnd2b4、myOr2b4 分别是 4 位 2 输入与门和 4 位 2 输入或门。



2.5 抖动原因：按键按下或放开时，存在机械震动。抖动时间一般在 10~20ms。按键去抖动方法：延时一段时间后再监测一次，以避开机械抖动。



去抖动原理

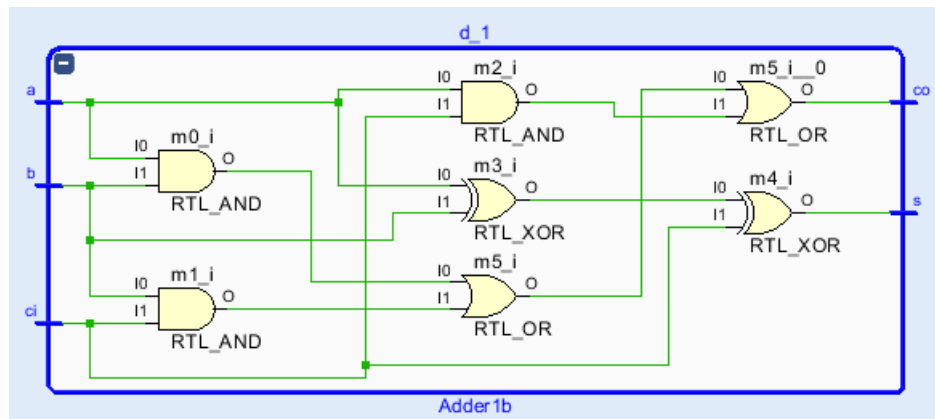
三、实验过程和数据记录

1 数据选择器设计

1.1 原理图方式设计 1 位加法器和 1 位加减法器并对 1 位加法器进行仿真

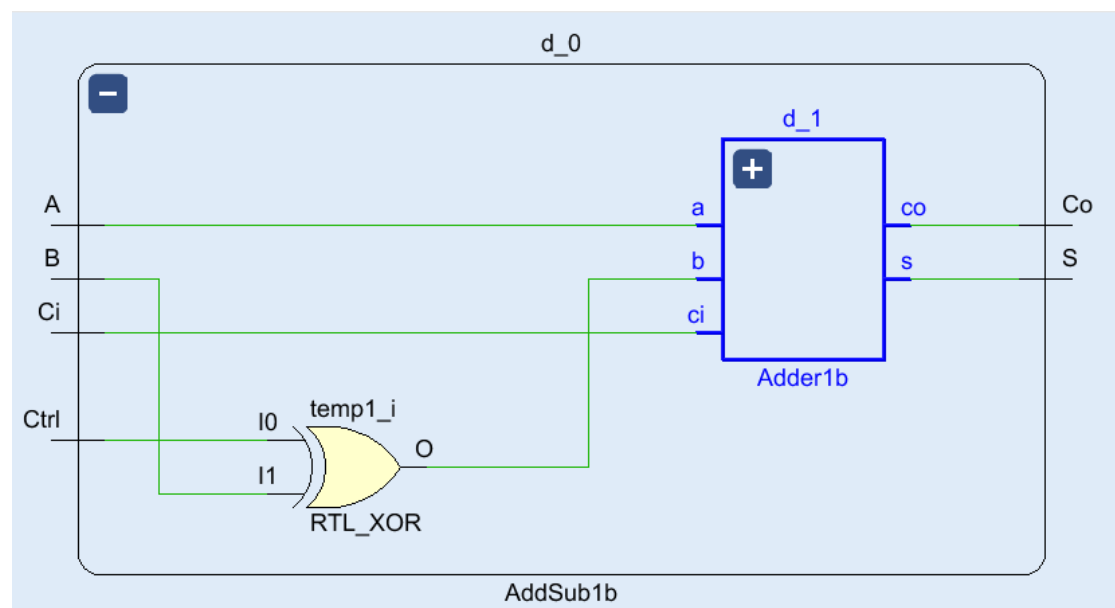
1 位加法器：

```
`timescale 1ns / 1ps
module Adder1b (
    input wire a,
    input wire b,
    input wire ci,
    output wire s,
    output wire co
);
    and m0 (c1, a, b);
    and m1 (c2, b, ci);
    and m2 (c3, a, ci);
    xor m3 (s1, a, b);
    xor m4 (s, s1, ci);
    or m5 (co, c1, c2, c3);
endmodule
```



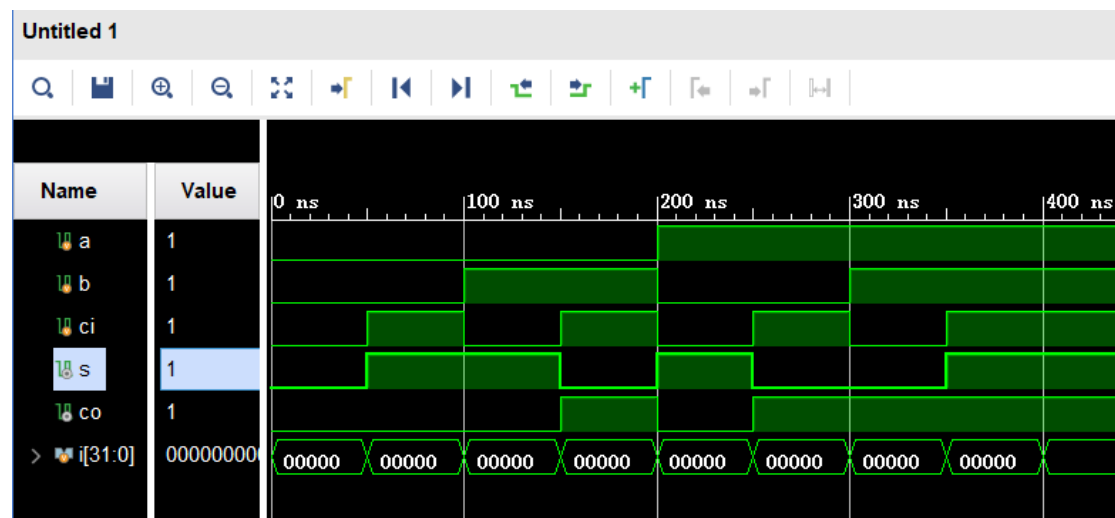
1 位加减法器:

```
module AddSub1b (
    A,
    B,
    Ctrl,
    Ci,
    S,
    Co
);
    input wire A, B, Ctrl, Ci;
    output wire S, Co;
    wire temp1=Ctrl^B;
    Adder1b d_1(A,temp1,Ci,S,Co);
endmodule
```



1 位加法器仿真：

```
module adder1b_test3 ();
    reg a, b, ci;
    wire s, co;
    integer i;
    Adder1b test3 (
        a,
        b,
        ci,
        s,
        co
    );
    initial begin
        for (i = 0; i < 8; i = i + 1) begin
            {a,b,ci}=i;
            #50;
        end
    end
endmodule
```



1.2 原理图方式设计 4 位加法器和 4 位加减法器并对 4 位加减法器进行仿真

4 位加法器：

```
module Adder4b (
    A,
    B,
    Ci,
```

```

        S,
        Co
    );
    input wire [3:0] A, B;
    input wire Ci;
    output wire [3:0] S;
    output wire Co;
    wire C1, C2, C3;
    Adder1b
        d_0 (
            A[0],
            B[0],
            Ci,
            S[0],
            C1
        ),
        d_1 (
            A[1],
            B[1],
            C1,
            S[1],
            C2
        ),
        d_2 (
            A[2],
            B[2],
            C2,
            S[2],
            C3
        ),
        d_3 (
            A[3],
            B[3],
            C3,
            S[3],
            Co);
endmodule

```

4 位加減法器:

```

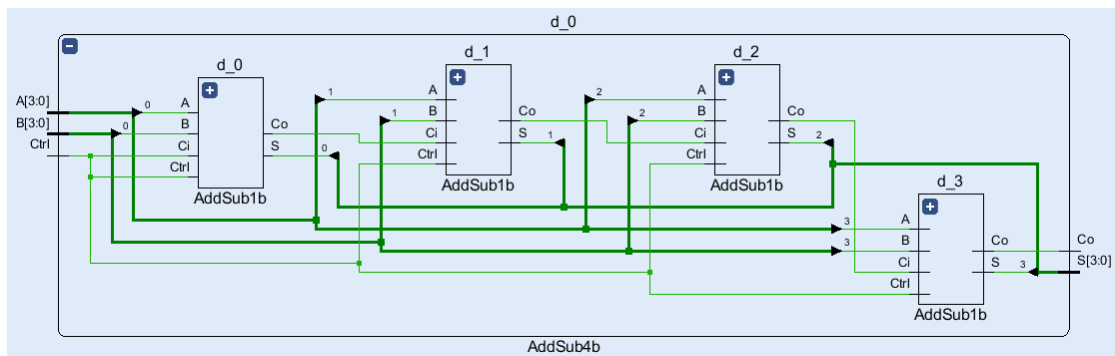
module AddSub4b(

    A,B,Ctrl,S,Co

);

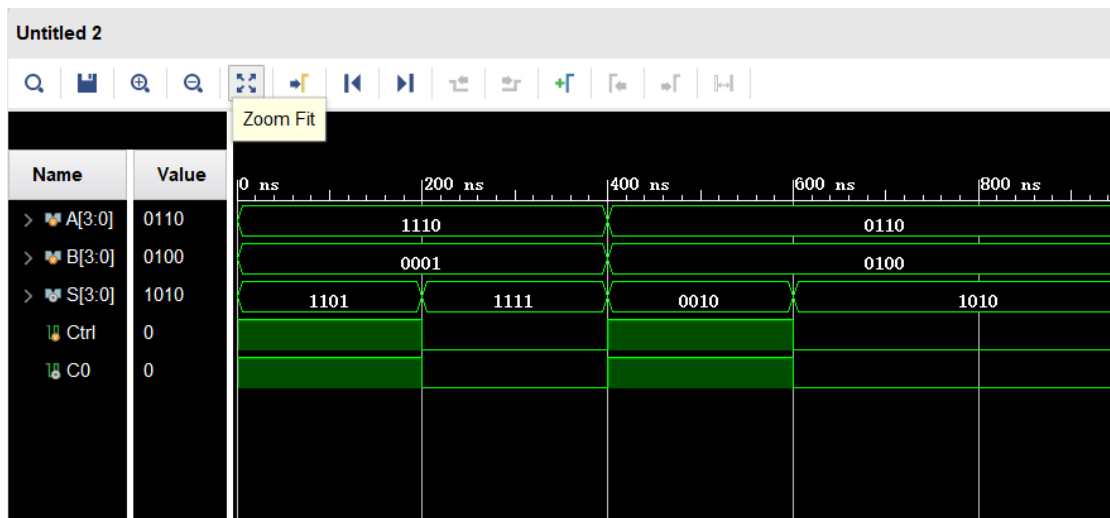
```

```
input wire[3:0]A,B;
input wire Ctrl;
output wire[3:0]S;
output wire Co;
wire C1,C2,C3;
  AddSub1b
  d_0 (
    A[0],
    B[0],
    Ctrl,
    Ctrl,
    S[0],
    C1
  ),
  d_1 (
    A[1],
    B[1],
    Ctrl,
    C1,
    S[1],
    C2
  ),
  d_2 (
    A[2],
    B[2],
    Ctrl,
    C2,
    S[2],
    C3
  ),
  d_3 (
    A[3],
    B[3],
    Ctrl,
    C3,
    S[3],
    Co
  );
endmodule
```

对 4 位加减法器进行仿真

```
module AddSub4b_test1 ();
    reg [3:0] A, B;
    wire [3:0] S;
    reg Ctrl;
    wire C0;
    AddSub4b test1 (
        A,
        B,
        Ctrl,
        S,
        C0
    );
    initial begin
        A[3:0] = 4'b1110;
        B[3:0] = 4'b0001;
        Ctrl = 1;
        #200;
        Ctrl = 0;
        #200;
        A[3:0] = 4'b0110;
        B[3:0] = 4'b0100;
        Ctrl = 1;
        #200;
        Ctrl = 0;
        #200;
    end
endmodule
```

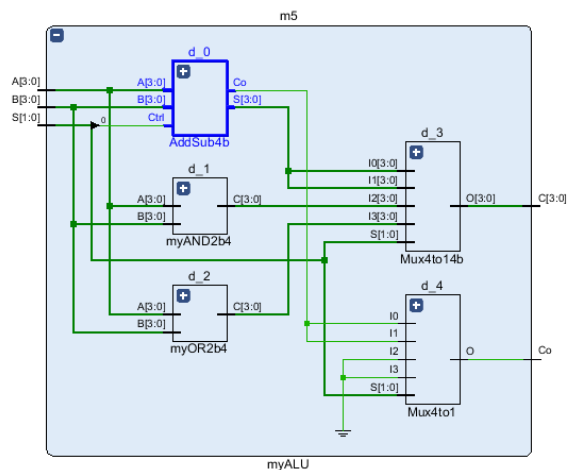


1.3 用设计图设计 myALU，并进行仿真。

```

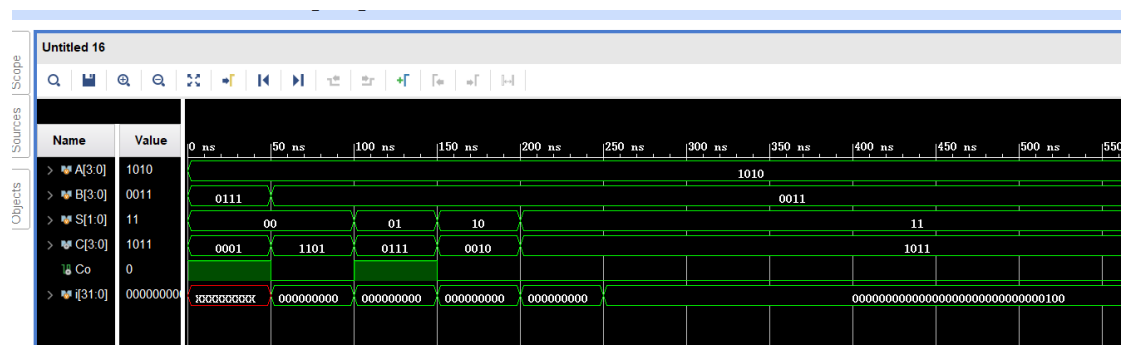
module myALU(A,B,S,C,Co);
input wire[3:0]A,B;
input wire[1:0]S;
output wire[3:0]C;
output wire Co;
wire[3:0] add_sub,myand,myor;
wire c_add_sub;
AddSub4b d_0(A,B,S[0],add_sub,c_add_sub);
myAND2b4 d_1(A,B,myand);
myOR2b4 d_2(A,B,myor);
Mux4to14b d_3(S,add_sub,add_sub,myand,myor,C);
Mux4to1 d_4(S,c_add_sub,c_add_sub,0,0,Co);
endmodule

```



对 ALU 进行仿真

```
module ALU_test2 ();
    reg [3:0] A, B;
    reg [1:0] S;
    wire [3:0] C;
    wire Co;
    ALU test2 (
        A,
        B,
        S,
        C,
        Co
    );
    integer i;
    initial begin
        A[3:0] = 4'b1010;
        B[3:0] = 4'b0111;
        S[1:0] = 2'b00;
        #50;
        B[3:0] = 4'b0011;
        for (i = 0; i <= 3; i = i + 1) begin
            S[1:0] = i;
            #50;
        end
    end
endmodule
```



1.4 设计防抖动模块

```
`timescale 1ns / 1ps
module pbdebounce(
    input wire clk_1ms,
    input wire button,
    output reg pbreg
```

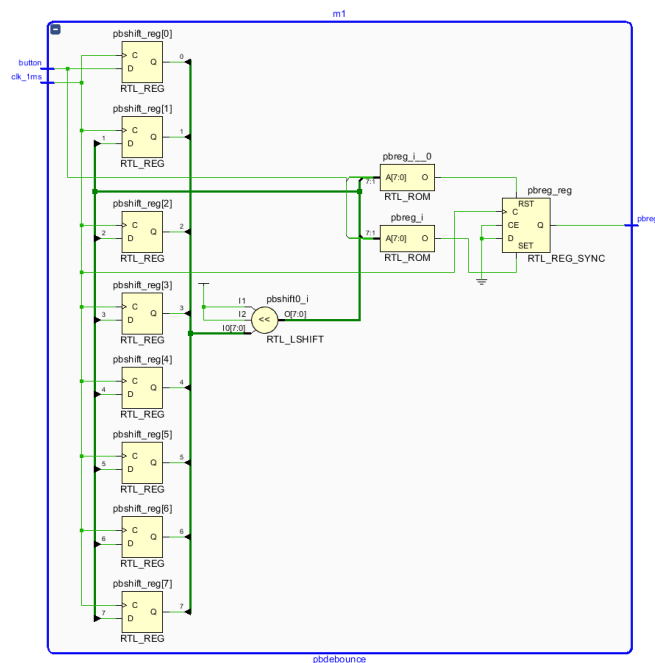
```

);

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

```



1.5 设计按键数据输入模块

```

module CreateNumber (
    input wire [ 3:0] btn,
    input wire [ 3:0] sw,
    output reg [15:0] num
);
    wire [3:0] A1, B1, C1, D1;
    initial num <= 16'b1010_1011_1100_1101;
    AddSub4b
    a1 (

```

```

        num[3:0],
        4'b0001,
        sw[0],
        A1
    ),
    a2 (
        num[7:4],
        4'b0001,
        sw[1],
        B1
    ),
    a3 (
        num[11:8],
        4'b0001,
        sw[2],
        C1
    ),
    a4 (
        num[15:12],
        4'b0001,
        sw[3],
        D1
    );
always @(posedge btn[0]) num[3:0] <= A1;
always @(posedge btn[1]) num[7:4] <= B1;
always @(posedge btn[2]) num[11:8] <= C1;
always @(posedge btn[3]) num[15:12] <= D1;
endmodule

```

1.6 设计顶层 top 模块。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。

```

`timescale 1ns / 1ps
module top (
    input wire clk,
    input wire [1:0] BTN,
    input wire [1:0] SW1,

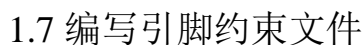
```

```

    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 (
    {0, 0, btn_out[0], btn_out[1]},
    {0, 0, SW1[0], SW1[1]},
    num[15:0]
);
myALU m5 (
    num[3:0],
    num[7:4],
    SW2[1:0],
    C[3:0],
    Co
);
DispNum m6 (
    .clk(clk),

```

```
endmodule
```



```
set_property IOSTANDARD LVCMOS15 [get_ports {SW2[0]}]
```

```

set_property PACKAGE_PIN AF10 [get_ports {SW2[1]}]
set_property PACKAGE_PIN AF13 [get_ports {SW2[0]}]

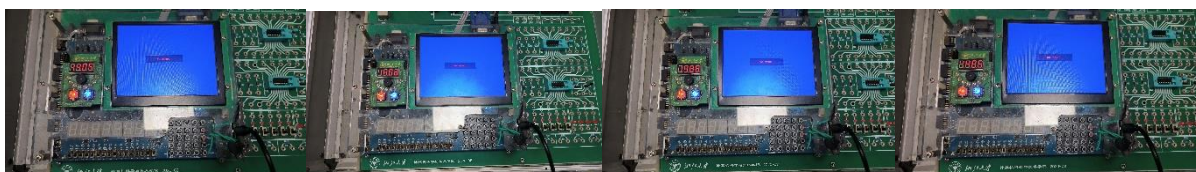
set_property PACKAGE_PIN W16 [get_ports BTN4]
set_property PACKAGE_PIN AC18 [get_ports clk]
set_property IOSTANDARD LVCMOS18 [get_ports BTN4]
set_property IOSTANDARD LVCMOS18 [get_ports clk]

set_property PACKAGE_PIN AD21 [get_ports {AN[0]}]
set_property PACKAGE_PIN AC21 [get_ports {AN[1]}]
set_property PACKAGE_PIN AB21 [get_ports {AN[2]}]
set_property PACKAGE_PIN AC22 [get_ports {AN[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {AN[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {AN[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {AN[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {AN[0]}]

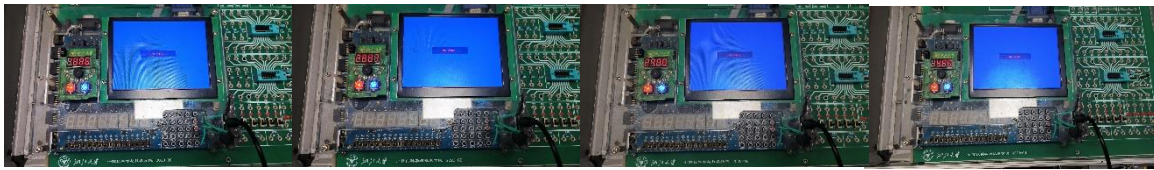
```

IO Ports													
Name	Direction	Neg Diff Pair	Package Pin	Fixed	Bank	I/O Std	Vcco	Vref	Drive Strength	Slew Type	Pull Type	Off-Chip Termination	#I_TERM
All ports (20)													
AN[3]	OUT		AC22	✓	12	LVCMOS33*	3.300	12	✓	SLOW	NONE	FP_VTT_50	✓
AN[2]	OUT		AB21	✓	12	LVCMOS33*	3.300	12	✓	SLOW	NONE	FP_VTT_50	✓
AN[1]	OUT		AC21	✓	12	LVCMOS33*	3.300	12	✓	SLOW	NONE	FP_VTT_50	✓
AN[0]	OUT		AD21	✓	12	LVCMOS33*	3.300	12	✓	SLOW	NONE	FP_VTT_50	✓
BTN[1]	IN		V14	✓	32	LVCMOS18	1.800				NONE	NONE	✓
BTN[0]	IN		W14	✓	32	LVCMOS18	1.800				NONE	NONE	✓
SEGMENT[8]	OUT			✓	12	LVCMOS33*	3.300	12	✓	SLOW	NONE	FP_VTT_50	✓
SEGMENT[7]	OUT		AA22	✓	12	LVCMOS33*	3.300	12	✓	SLOW	NONE	FP_VTT_50	✓
SEGMENT[6]	OUT		AC23	✓	12	LVCMOS33*	3.300	12	✓	SLOW	NONE	FP_VTT_50	✓
SEGMENT[5]	OUT		AC24	✓	12	LVCMOS33*	3.300	12	✓	SLOW	NONE	FP_VTT_50	✓
SEGMENT[4]	OUT		W20	✓	12	LVCMOS33*	3.300	12	✓	SLOW	NONE	FP_VTT_50	✓
SEGMENT[3]	OUT		Y21	✓	12	LVCMOS33*	3.300	12	✓	SLOW	NONE	FP_VTT_50	✓
SEGMENT[2]	OUT		AD23	✓	12	LVCMOS33*	3.300	12	✓	SLOW	NONE	FP_VTT_50	✓
SEGMENT[1]	OUT		AD24	✓	12	LVCMOS33*	3.300	12	✓	SLOW	NONE	FP_VTT_50	✓
SEGMENT[0]	OUT		AB22	✓	12	LVCMOS33*	3.300	12	✓	SLOW	NONE	FP_VTT_50	✓
SW1[1]	IN			✓	33	LVCMOS15*	1.500				NONE	NONE	✓
SW1[0]	IN		AB10	✓	33	LVCMOS15*	1.500				NONE	NONE	✓
SW2[1]	IN			✓	33	LVCMOS15*	1.500				NONE	NONE	✓
SW2[0]	IN		AF10	✓	33	LVCMOS15*	1.500				NONE	NONE	✓
SW2[0]	IN		AF13	✓	33	LVCMOS15*	1.500				NONE	NONE	✓
Scalar ports (2)													
BTN4	OUT		W16	✓	32	LVCMOS18	1.800	12	✓	SLOW	NONE	FP_VTT_50	✓
clk	IN		AC18	✓	32	LVCMOS18	1.800				NONE	NONE	✓

1.8 烧录至测试板，进行实际测试
验证 ALU 运算 (+,-,or,and) 结果



验证按按钮控制 a，b 递增递减：



四、实验结果分析

波形图符合运算要求：

1 位加法器符合真值表

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

4 位加减法器：

$b1110-b0001=b1101$ $co=1$ $b1110+b0001=b1111$ $co=0$

$b0110-b0100=b0010$ $co=1$ $b0110+b0100=b1010$ $co=0$

4 位 ALU：

$b1010+b0111=b0001$ $co=1$

$b1010+b0011=b1101$ $co=0$ $b1010-b0011=b0111$ $co=1$

$b1010\&b0011=b0010$ $co=0$ $b1010|b0011=b1011$ $co=0$

下载验证按按钮 A、B 依次递增或递减，ALU 运算也均符合预期

五、讨论与心得

通过延时的方式可以避免按键振动带来得多次判断。

此外在做减法运算时，由于采用的取反加一再相加的方式，导致未算数溢出时 Co 反而为 1。