

浙江大学

本科实验报告

课程名称：数字逻辑设计

姓 名：颜晗。

学 院：计算机科学与技术学院

系：

专 业：计算机科学与技术

学 号：3200105515

指导教师：蔡铭

2021 年 12 月 4 日

浙江大学实验报告

课程名称： 数字逻辑设计 实验类型： 综合

实验项目名称： 全加器的设计实现

学生姓名： 颜晗 专业： 计算机科学与技术 学号： 3200105515

同组学生姓名： 指导老师： 蔡铭

实验地点： 东四 509 实验日期： 2021 年 11 月 15 日

一、实验目的和要求

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. 了解加法器在 CPU 中的地位
4. 掌握 FPGA 开发平台进行简单的 I/O 数据交互

二、实验内容和原理

实验内容：

任务一：设计 4 位串行进位加法器；

任务二：实现 4 位加法器应用。（两个晶体管作为输入，另两个显示输出）

实验原理：

1. 加法原理（一位全加器与四位全加器）

一位：三个输入：两个操作数与一个低位进位数

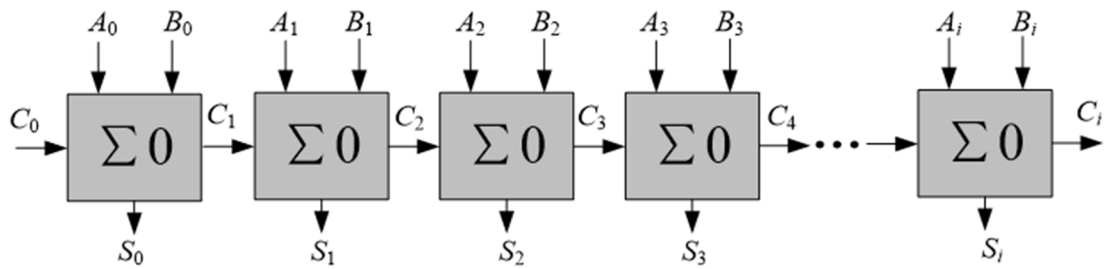
两个输出：全加和以及进位（向高位输出）

三个输入共八种状态，可以绘制真值表，并使用卡诺图进行优化：

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

$$S_i = A_i \oplus B_i \oplus C_i$$
$$C_{i+1} = A_i B_i + B_i C_i + C_i A_i$$

四位及多位：多位全加器可由一位全加器将进位串接构成；高位的运算依赖于低位的进位，因此这种全加器在位数多时会有很大的延迟，耗时较长。



三、实验过程和数据记录

任务一：设计 4 位串行进位加法器

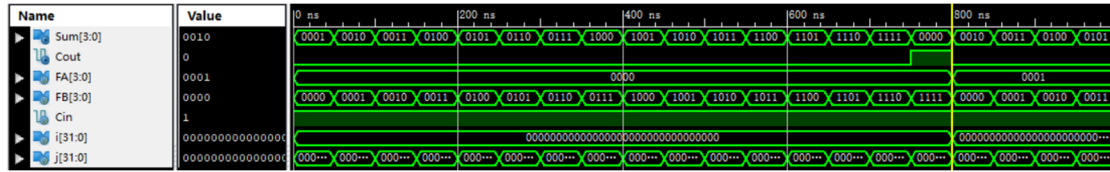
1. 新建工程。
2. 新建一位全加器的源文件（Verilog module 或 Schematic）
3. 根据化简的函数设计原理图或代码。

```
module Adder1b_ver(  
    input wire a,b,ci,  
    output wire s,co  
);  
    and mo(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
```

4. 直接在原工程新建四位全加器的源文件或新建一个工程以及源文件进行设计。（注意要调用前面设计的一位全加器）

```
module Adder4b_ver(  
    FA, FB, Cin, Sum, Cout  
);  
    parameter SIZE = 4;  
    input [SIZE-1:0] FA, FB;  
    output [SIZE-1:0] Sum;  
    input Cin;  
    output Cout;  
    wire [1:SIZE-1] Temp;  
  
    Adder1b_ver  
    adder_1(FA[0], FB[0], Cin, Sum[0], Temp[1]),  
    adder_2(FA[1], FB[1], Temp[1], Sum[1], Temp[2]),  
    adder_3(FA[2], FB[2], Temp[2], Sum[2], Temp[3]),  
    adder_4(FA[3], FB[3], Temp[3], Sum[3], Cout);  
  
endmodule
```

5. 四位全加器的仿真



任务二：实现 4 位加法器应用

1. 新建工程
2. 将相关模块添加进工程（pbdebounce 模块，Adder4b 模块，clkdiv 模块，Mux4to1b4 模块、MC14495 模块，CreateNumber 模块）
3. 新建顶层模块源文件。

```
module addmachine(
    input clk_100mhz,
    input RSTN,
    input BN,
    input [1:0] btn,
    input [7:0] SW,
    output K_ROW,
    output [3:0] AN,
    output [7:0] SEGMENT
);

wire [31:0] clkdiv;
wire V0;
wire V5;
wire [7:0] num;
wire [3:0] out;
wire LE;
wire Point;
wire [3:0] sum;
wire co;

clkdiv f1(.clk(clk_100mhz),.rst(RSTN),.clkdiv(clkdiv));
createNumber f2(.btn(btn[1:0]),
    .clk(clkdiv[17]),
    .num(num[7:0]));

Adder4b_ver add1(.FA(num[3:0]),
    .FB(num[7:4]),
    .Cin(V0),
    .Sum(sum[3:0]),
    .Cout(co));

Mux4to1b4 f3(.IO(num[3:0]),
    .I1(num[7:4]),
    .I2(sum[3:0]),
    .I3({V0,V0,V0,co}),
    .s(clkdiv[18:17]),
    .o(out[3:0]));

Mux4to1 f4(.IO(SW[0]),
    .I1(SW[1]),
    .I2(SW[2]),
    .I3(SW[3]),
    .s(clkdiv[18:17]),
    .o(Point));

Mux4to1 f5(.IO(SW[4]),
    .I1(SW[5]),
    .I2(SW[6]),
    .I3(SW[7]),
    .s(clkdiv[18:17]),
    .o(LE));

Mux4to1b4 f6(.IO({V5, V5, V5, V0}),
    .I1({V5, V5, V0, V5}),
    .I2({V5, V0, V5, V5}),
    .I3({V0, V5, V5, V5}),
    .s(clkdiv[18:17]),
    .o(AN[3:0]));

MyMC14495 f7(.D(out[3:0]),
    .LE(LE),
    .Point(Point),
    .LED(SEGMENT[6:0]),
    .p(SEGMENT[7]));

BUF f8(.I(BN), .O(K_ROW));
VCC f9(.P(V5));GND f10(.G(V0));

endmodule
```

4. 建立引脚约束文件

```

NET "clk_100mhz"      LOC=AC18 | IOSTANDARD = LVCMOS18;
NET "RSTN"            LOC = AF10 | IOSTANDARD = LVCMOS15;

NET "SW[0]"           LOC = AA10 | IOSTANDARD = LVCMOS15;#POINT
NET "SW[1]"           LOC = AB10 | IOSTANDARD = LVCMOS15;
NET "SW[2]"           LOC = AA13 | IOSTANDARD = LVCMOS15;
NET "SW[3]"           LOC = AA12 | IOSTANDARD = LVCMOS15;

NET "SW[4]"           LOC = Y13  | IOSTANDARD = LVCMOS15;#LES
NET "SW[5]"           LOC = Y12  | IOSTANDARD = LVCMOS15;
NET "SW[6]"           LOC = AD11  | IOSTANDARD = LVCMOS15;
NET "SW[7]"           LOC = AD10  | IOSTANDARD = LVCMOS15;

NET "SEGMENT[0]"      LOC = AB22  | IOSTANDARD = LVCMOS33 ;#a
NET "SEGMENT[1]"      LOC = AD24  | IOSTANDARD = LVCMOS33 ;#b
NET "SEGMENT[2]"      LOC = AD23  | IOSTANDARD = LVCMOS33 ;
NET "SEGMENT[3]"      LOC = Y21   | IOSTANDARD = LVCMOS33 ;
NET "SEGMENT[4]"      LOC = W20   | IOSTANDARD = LVCMOS33 ;
NET "SEGMENT[5]"      LOC = AC24  | IOSTANDARD = LVCMOS33 ;
NET "SEGMENT[6]"      LOC = AC23  | IOSTANDARD = LVCMOS33 ;#g
NET "SEGMENT[7]"      LOC = AA22  | IOSTANDARD = LVCMOS33 ;#poi
NET "AN[3]"           LOC = AC22  | IOSTANDARD = LVCMOS33 ;
NET "AN[2]"           LOC = AB21  | IOSTANDARD = LVCMOS33 ;
NET "AN[1]"           LOC = AC21  | IOSTANDARD = LVCMOS33 ;
NET "AN[0]"           LOC = AD21  | IOSTANDARD = LVCMOS33 ;

NET "btn[1]"          LOC = V14   | IOSTANDARD = LVCMOS18 ;
NET "btn[0]"          LOC = W14   | IOSTANDARD = LVCMOS18 ;

NET "K_ROW"           LOC = V17   | IOSTANDARD = LVCMOS18 ;
NET "BN"              LOC = AE10   | IOSTANDARD = LVCMOS15 ;

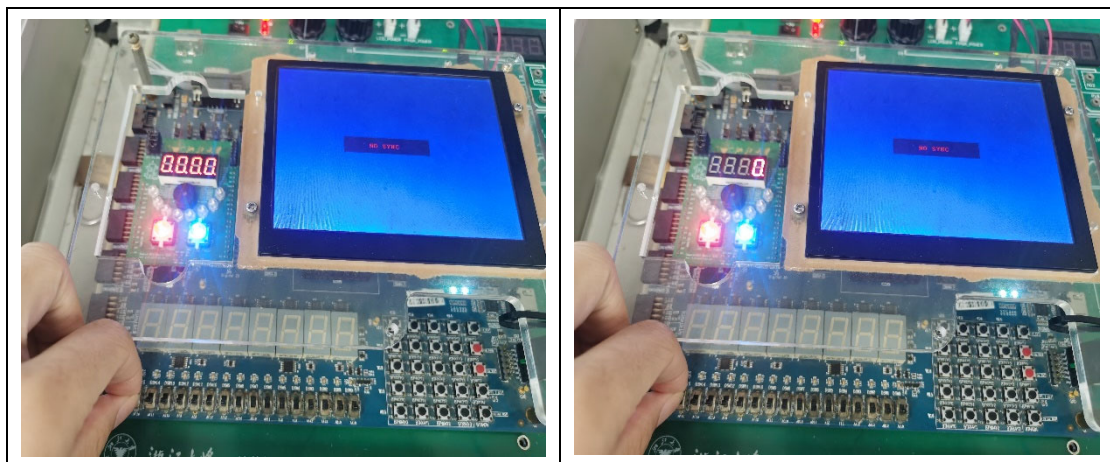
```

5. 所有文件检查无误后进行综合、映射、生成编程文件后上板验证。

四、实验结果分析

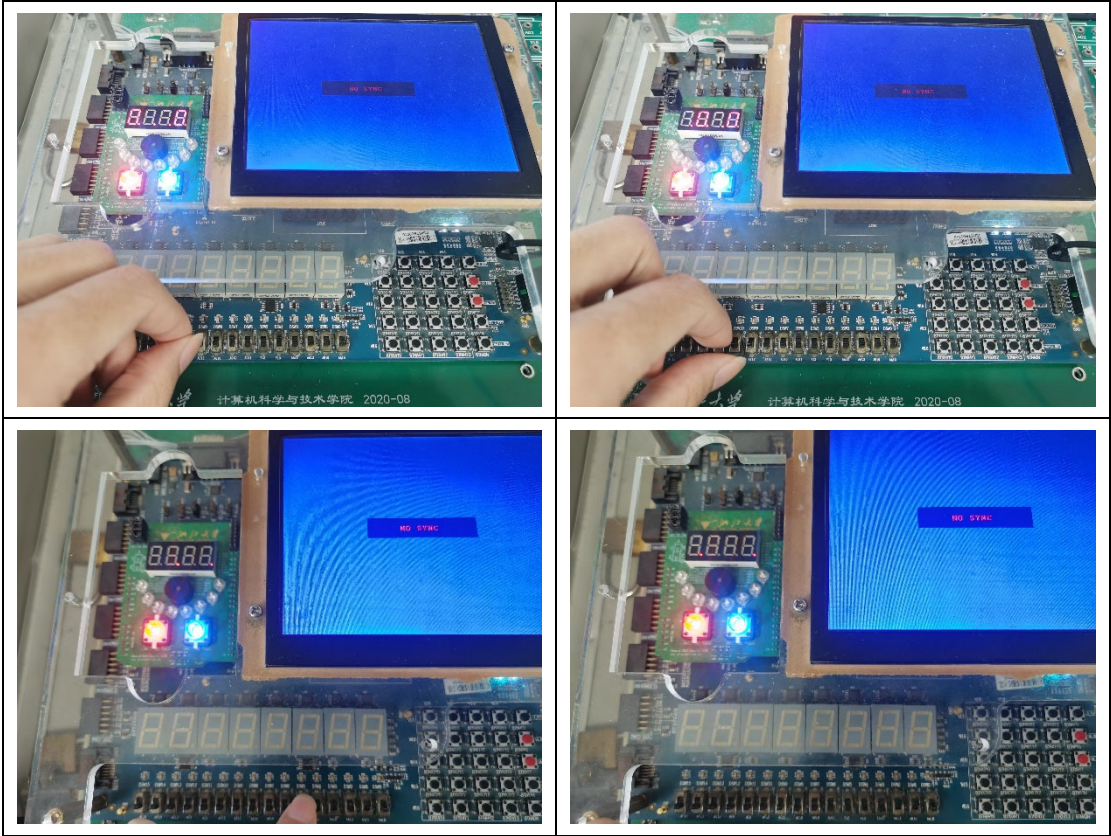
1. 复位键功能验证（整体使能）

复位键拨动为 1 时，无法进行操作。



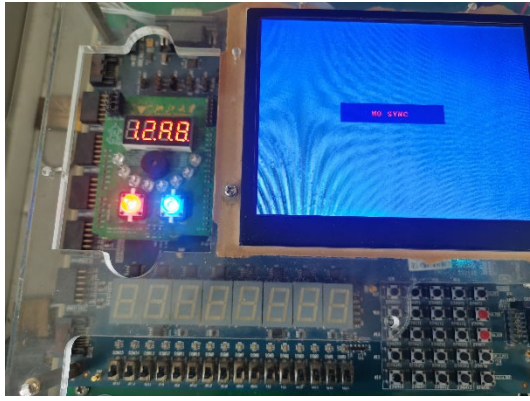
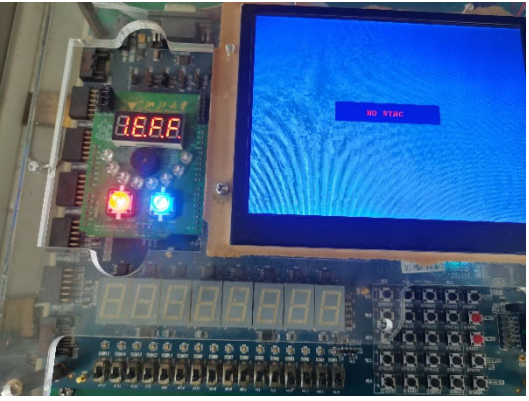
2. 单个数字与小数点的消隐控制

八个按键分别控制四个数字与四个小数点的消隐，即使不显示的数字在内部依旧会随操作变化。



3. 数字加法（后两段晶体管分别作为两个输入，前两个晶体管作为输出显示）
由于输入位数限制，加和最大为 $1e(16)$ 。

6=3+3	d=a+3

$12=a+8$	$1e=f+f$
	

五、实验总结与反思

本次实验基于实验七的模块又添加了全加器模块，最后相当于实现了一个极其简单的计算器；总体没有难度，在函数已经给定的情况下，编写代码实现模块已经不是问题，顶层模块相比实验七也只是增加了一个全加器，部分线路有所变化，只要仔细，很快就可以做完。加法是各电子器件中最为重要的运算之一，有大量复杂的运算实际上都可以转换为加法进行计算，理解硬件中加法的运算原理，对于我们深入了解知识有巨大的好处。