

《EDA技术》试验

实验七

复杂系统设计与实现 1

专业： 计算机科学与技术

姓名： 高星杰

学号： 2021307220712

班级： 计科 2102

报告上交时间：2023 年 5 月 16 日

一、实验目的

1、复杂系统设计与实现

二、实验任务及要求

任务一、32 位超前进位加法器设计与实现： 可参考《计算机原理与设计实训指导》第 3.2.2 节-，要求弄懂细节，能讲清楚设计思路，写实验报告时请讲清楚设计思路，仅有结果则不得分。

附加题：

附加题 1：供电控制电路设计与实现，3 个工厂由甲和乙两个变电站供电。若一个工厂用电，则由甲站供电；若两个工厂用电，则由乙站供电；若三个工厂用电，则由甲、乙两站同时供电。请设计一个供电控制电路，要求用三个开关分别表示工厂用电请求，用两个 led 灯显示供电结果。

附加题 2：输血血型控制系统设计与实现，输血者与受血者的血型必须符合下述原则：O 型血可以输给任意血型的人，但 O 型血只能接受 O 型血；AB 型血只能输给 AB 型，但 AB 型能接受所有血型；A 型血能输给 A 型和 AB 型，但只能接受 A 型或 O 型血；B 型血能输给 B 型和 AB 型，但只能接受 B 型或 O 型血。提示：可用 2 位二进制对 4 种血型编码，用 4 个开关分别输入供需双方血型，led 指示灯指示输血情况是否正确。

三、实验原理与步骤

任务一、32 位超前进位加法器设计与实现：

设计原理：超前进位加法器就是不等上一位计算完下一位已经开始计算了，就是同时计算类似于并行计算，而不是串行。

32位超前进位加法器设计

原理：

$$s_i = a_i \oplus b_i \oplus c_i$$

$$c_{i+1} = a_i b_i + b_i c_i + a_i c_i$$

$$c_i = a_i b_i + b_i c_{i-1} + a_i c_{i-1}$$

$$= a_i b_i + c_{i-1} (a_i + b_i)$$

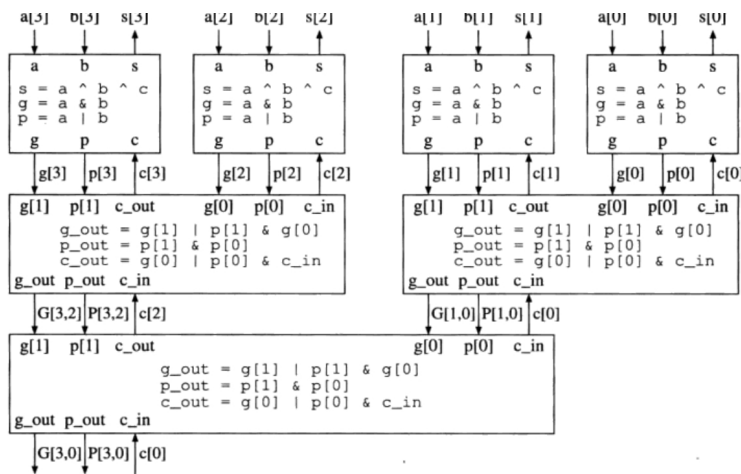
$$= a_i b_i + c_{i-1} (a_i + b_i)$$

设 $g[i] = a_i b_i$ $p[i] = a_i + b_i$

$$c_i = g[i] + c_{i-1} p[i]$$

然后可以根据这个公式推的每一位对应的公式，然后实现电路，可以发现每一位结果的函数互相不依赖，但是如果这样实现电路会太复杂，

所以设计出这样的电路



总体思路是：这个是一个四位的超前进位加法器，可以看出其是由两个 2 位的超前进位加法器和一个 GP 生成器组成。

和一个 GP 生成器组成。

所以我们可以再将其封装，两个 4 位加法器和一个 GP 生成器可以组成一个 8 位的超前进位加法器，然后不断封装再组合，最后由两个 16 位的超前进位加法器和一个 GP 生成器组合而成一 32 位的超前进位加法器。

GP 生成器的输入应该是 i 位的 g 、 p 和 c 和 $i+1$ 位的 g 、 p ，输出 $i+1$ 位的 g 、 p 的 c 和 i 位的 c ，具体的算式由刚才推得的算式得出。

思路确定后再进行代码编写。

Verilog 代码:

GP 生成器

```
module g_p (g,p,c_in,g_out,p_out,c_out);
input [1:0] g,p;
input c_in;
output g_out,p_out,c_out;
assign g_out=g[1]|p[1]&g[0];
assign p_out=p[1]&p[0];
assign c_out=g[0]|p[0]&c_in;
endmodule
```

1 位加法器:

```
module add(a,b,c,g,p,s);
input a,b,c;
output g,p,s;
assign s=a^b^c;
assign g=a&b;
assign p=a|b;
endmodule
```

2 位加法器:

```

module cla2(a,b,cin,gout,pout,s);
input[1:0] a,b;
input cin;
output gout,pout;
output[1:0]s;
wire[1:0]g,p;
wire cout;
add add0(a[0],b[0],cin,g[0],p[0],s[0]);
add add1(a[1],b[1],cout,g[1],p[1],s[1]);
g_p g_p0(g,p,cin,gout,pout,cout);
endmodule

```

4 位加法器:

```

module cla4(a,b,cin,gout,pout,s);
input[3:0] a,b;
input cin;
output gout,pout;
output[3:0]s;
wire[1:0]g,p;
wire cout;
cla2 cla0(a[1:0],b[1:0],cin,g[0],p[0],s[1:0]);
cla2 cla1(a[3:2],b[3:2],cout,g[1],p[1],s[3:2]);
g_p g_p0(g,p,cin,gout,pout,cout);
endmodule

```

8 位加法器:

```

module cla8(a,b,cin,gout,pout,s);
input[7:0] a,b;
input cin;
output gout,pout;
output[7:0]s;
wire[1:0]g,p;
wire cout;
cla4 cla0(a[3:0],b[3:0],cin,g[0],p[0],s[3:0]);
cla4 cla1(a[7:4],b[7:4],cout,g[1],p[1],s[7:4]);
g_p g_p0(g,p,cin,gout,pout,cout);
endmodule

```

16 位加法器:

```

module cla16(a,b,cin,gout,pout,s);
input[15:0] a,b;
input cin;
output gout,pout;
output[15:0]s;
wire[1:0]g,p;
wire cout;
cla8 cla0(a[7:0],b[7:0],cin,g[0],p[0],s[7:0]);
cla8 cla1(a[15:8],b[15:8],cout,g[1],p[1],s[15:8]);
g_p g_p0(g,p,cin,gout,pout,cout);
endmodule

```

32 位加法器:

```

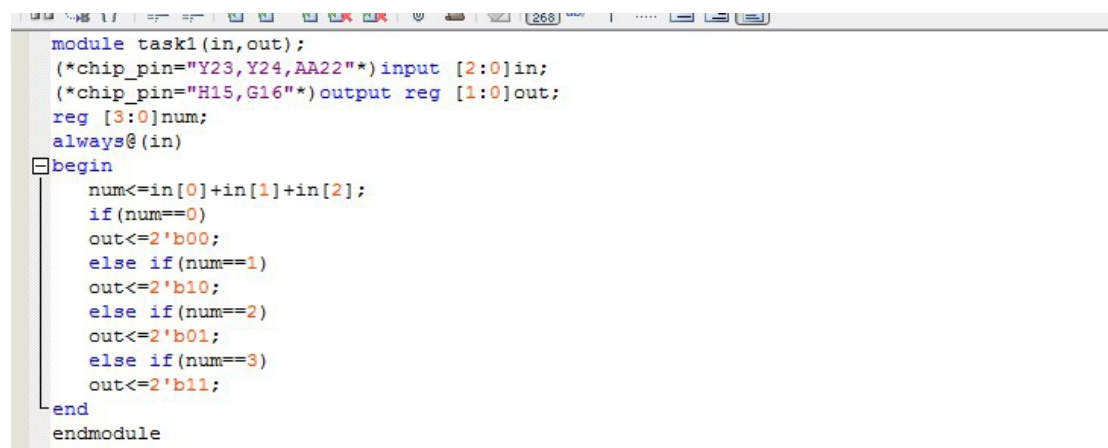
module cla32(a,b,cin,gout,pout,s);
input[31:0] a,b;
input cin;
output gout,pout;
output[31:0]s;
wire[1:0]g,p;
wire cout;
cla16 cla0(a[15:0],b[15:0],cin,g[0],p[0],s[15:0]);
cla16 cla1(a[31:16],b[31:16],cout,g[1],p[1],s[31:16]);
g_p g_p0(g,p,cin,gout,pout,cout);
endmodule

```

附加题 1: 供电控制电路设计与实现

实验原理: 运用行为描述计算总共有多少个工厂用电然后根据数量来分类, 分配不同的变电站。

Verilog 代码:



```

module task1(in,out);
(*chip_pin="Y23,Y24,AA22"*)input [2:0]in;
(*chip_pin="H15,G16"*)output reg [1:0]out;
reg [3:0]num;
always@(in)
begin
    num<=in[0]+in[1]+in[2];
    if(num==0)
    out<=2'b00;
    else if(num==1)
    out<=2'b10;
    else if(num==2)
    out<=2'b01;
    else if(num==3)
    out<=2'b11;
end
endmodule

```

附加题 2：输血血型控制系统设计与实现

实验原理：首先对各个血型进行编码：O 为 00、AB 为 01、A 为 10、B 为 11，其次同样是行为描述，根据输血者血型分类，然后判断接受输血的人的血型是否符合要求。

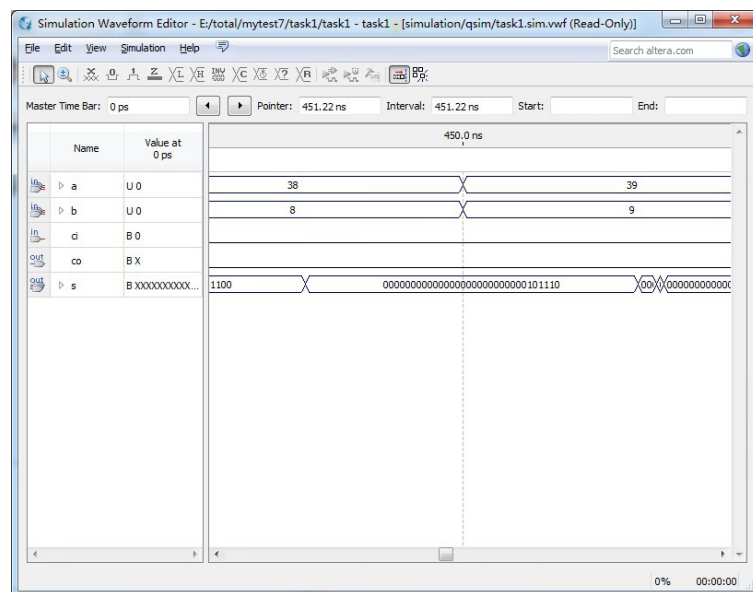
Verilog 程序：

```
module task2(in1,in2,out);
(*chip_pin="Y23,Y24"*)input [1:0]in1;
(*chip_pin="AA22,AA23"*)input [1:0]in2;
(*chip_pin="H15"*)output reg out;
parameter O=2'b00,AB=2'b01,A=2'b10,B=2'b11;
always@(in1,in2)
begin
case(in1)
O:out<=1;
AB:if(in2==AB)out<=1;else out<=0;
A:if(in2==A||in2==AB)out<=1;else out<=0;
B:if(in2==B||in2==AB)out<=1;else out<=0;
default out<=0;
endcase
end
endmodule
```

四、实验结果与分析

任务一、32 位超前进位加法器设计与实现；

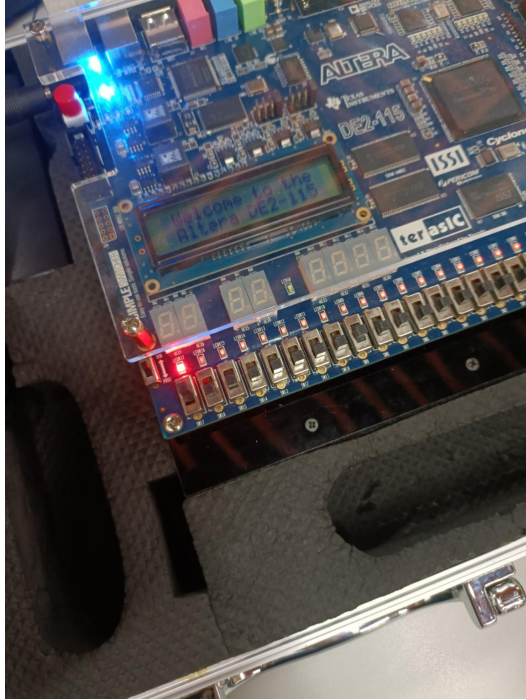
波形仿真：



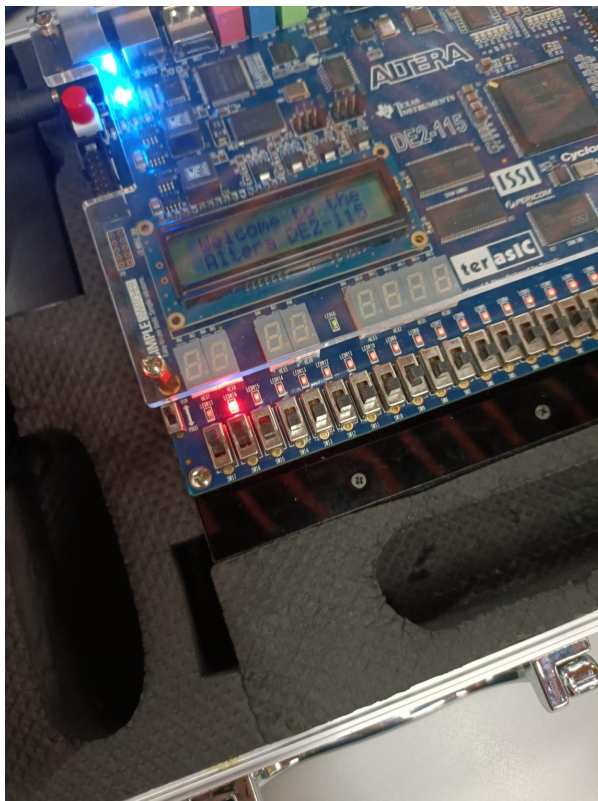
可以看到 $38+8=46$ 也就是 101110 说明实验结果正确。

附加题 1：供电控制电路设计与实现

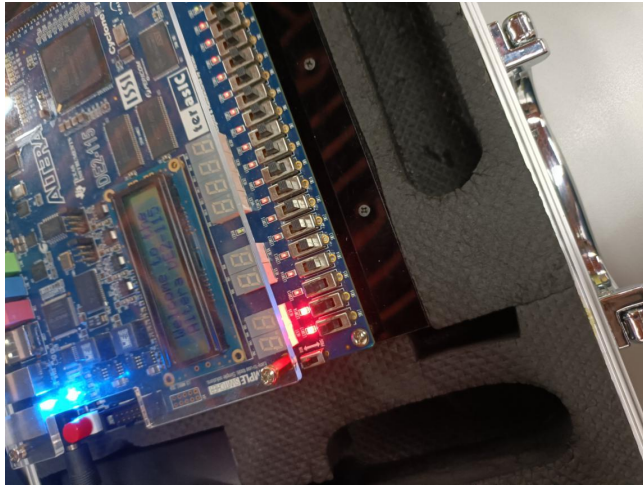
一个工厂亮 A:



两个工厂亮 B:



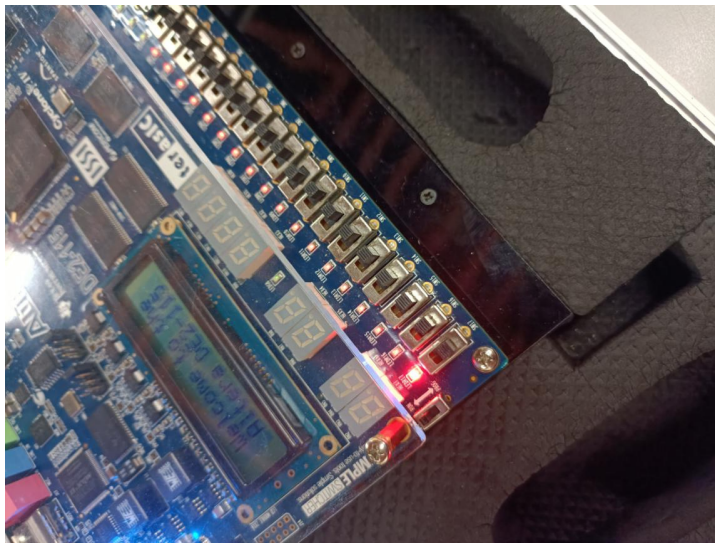
三个工厂 AB 都亮:



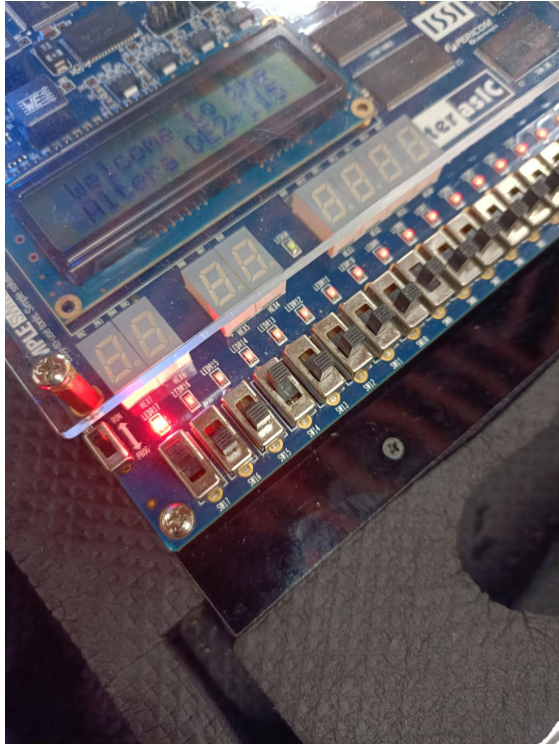
可以根据结果看出实验成功

附加题 2：输血血型控制系统设计与实现

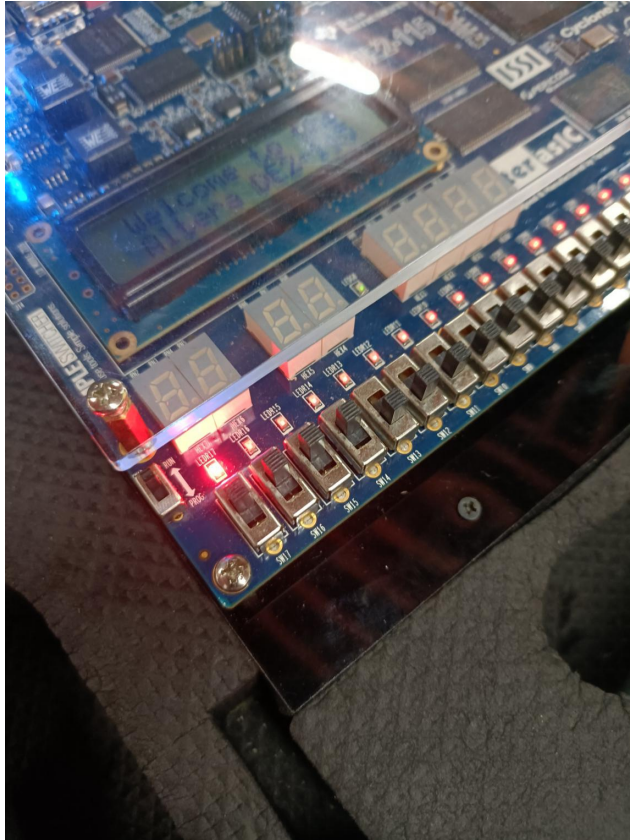
00: 00 (0 输入 0)



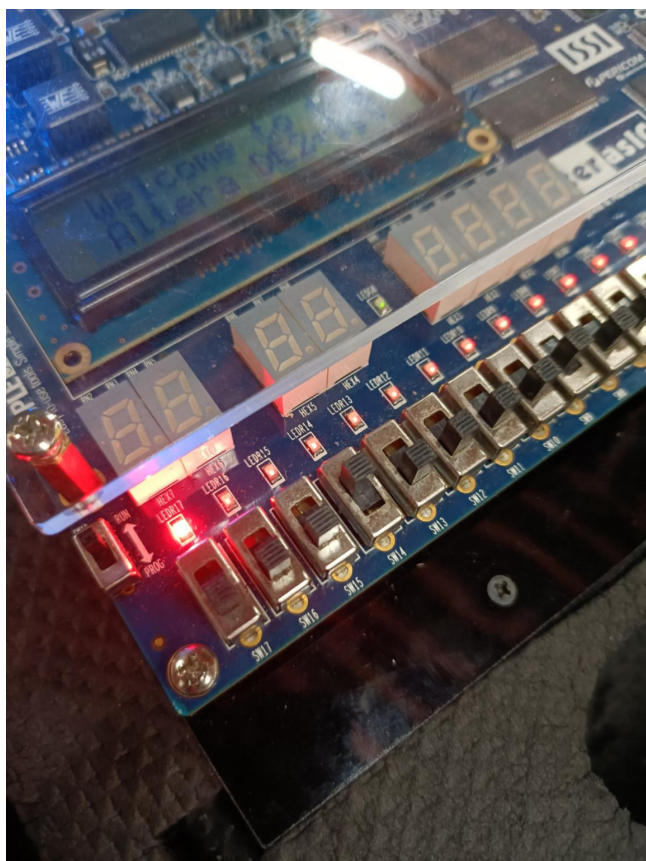
10:01 (A 输入 AB)



11: 11 (B 输入 B)



00: 01 (0 输入 AB)



根据结果可以看出实验成功

五、实验体会与讨论

本次实验内容不多，任务一主要是理解，理解之后再写代码就容易的多，之前只在课上听了超前进位加法器的理论还有些一知半解，现在在编写之后加深了理解，可以自己手动实现任意位的超前进位加法器了。附加题也比较容易，都是一次编成没有出错，对行为描述的编写方式更加熟练了。