**组成原理实验课程第 1 次实验报告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 实验名称 | 数据运算：定点加法 | | | 班级 | 李涛 |
| 学生姓名 | 胡博浩 | 学号 | 2212998 | 指导老师 | 董前琨 |
| 实验地点 | 津南实验楼A308 | | 实验时间 | 2024.3.21 | |

1. **实验目的**

（1）熟悉LS-CPU-EXB-002 实验箱和软件平台。

（2）掌握利用该实验箱各项功能开发组成原理和体系结构实验的方法。

（3）理解并掌握加法器的原理和设计。

（4）熟悉并运用verilog语言进行电路设计。

（5）为后续设计cpu的实验打下基础。

1. **实验内容说明**

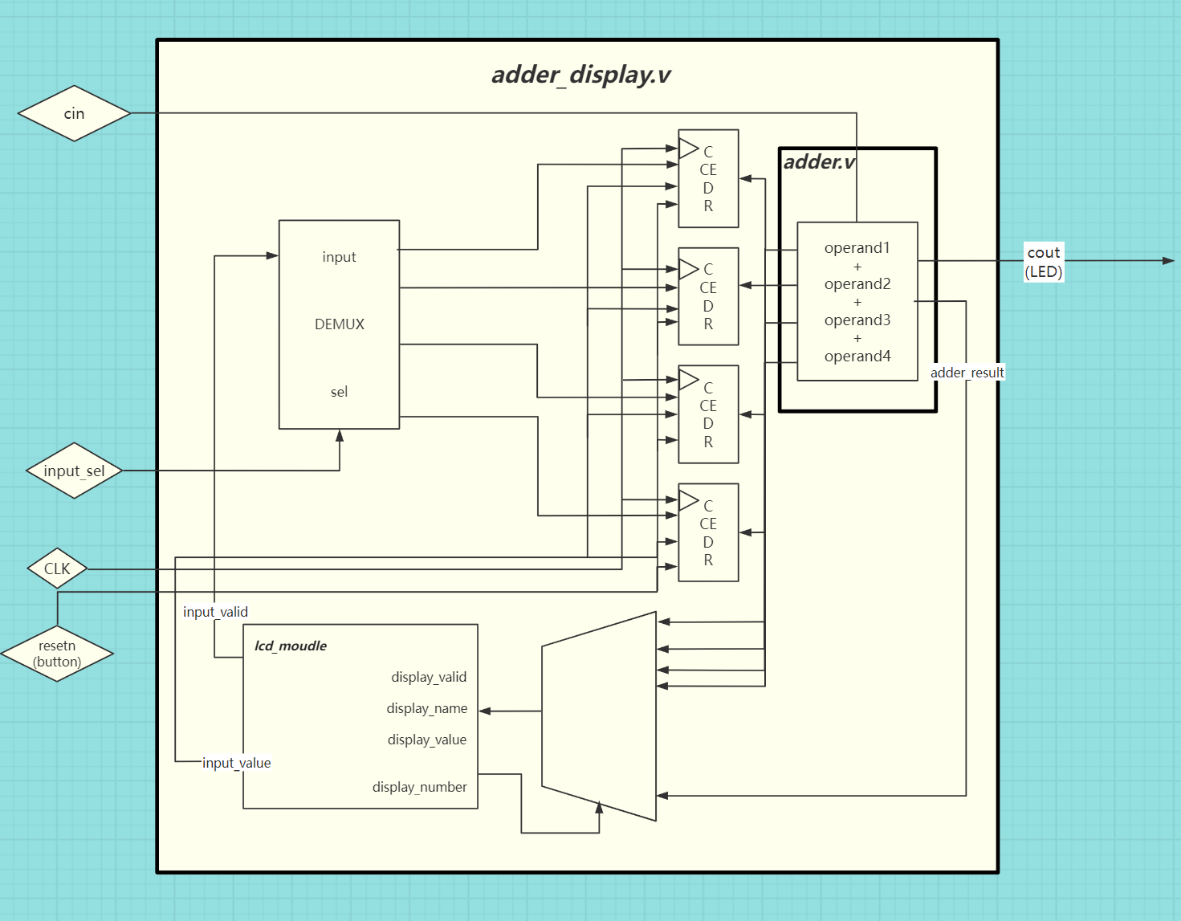
请结合实验指导手册中的实验一（加法器实验）完成功能改进，实现一个能做4个32位数的加法的加法器，注意以下几点：

（1）除了修改adder模块，display、testbench和约束文件都有修改，注意加法器进位，四个数加法会出现2位的进位。

（2）实验报告中需要补充原理图，并对原理图进行解释说明。原理图参照图2.40进行修改，建议使用visio画图（别的画图软件也可，不会画图的可以手绘然后照片放报告里面）。

（3）实验报告中需要有仿真结果（波形截图），并针对图中的数据解释说明（某一时刻哪些数相加，进位情况如何，结果如何，是否验证的模块的正确性），还需要有实验箱上箱验证的照片，同样，针对照片中的数据也需要解释说明。

1. **实验原理图**

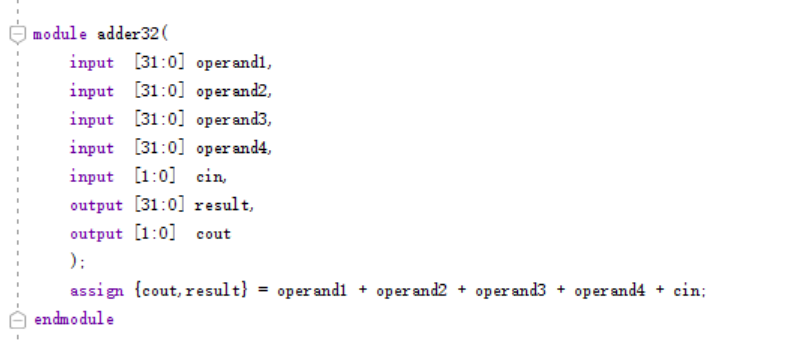


原理图展示了一个模块化的设计，主模块为adder\_display，它调用了两个子模块：adder用于加法运算，lcd\_module用于显示。在adder\_display模块的输入端，cin是加法器的进位输入，而input\_sel是一个选择器信号，用于选择输入到加法器的操作数。这些输入信号通过拨码开关进行输入，允许用户手动设置操作数。加法器的输出端连接到了一个名为cout的信号线上，代表加法运算的进位输出。此外，cout还连接到了一个LED灯上，用于显示加法运算的进位结果。当进位发生时，LED灯会亮起，提供直观的视觉反馈。此外，图中还包含了一个DEMUX模块，它是一个数据选择器，用于根据sel信号选择并传递不同的操作数到加法器。通过调整sel信号的值，可以选择operand1、operand2、operand3或operand4中的任意一个作为加法器的输入。

综上所述，通过调用子模块、设置输入信号和连接显示器，该设计能够实现加法运算结果的显示功能。同时，通过拨码开关和选择器信号，用户可以灵活地选择和配置输入操作数以及观察运算结果。

1. **实验步骤**

（1）加法器adder32.v



1. **代码修改：添加两个加数(operand3,operand4)，将进位输入(cin)和进位输出(cout)修改为2位，求和计算也相应改变。**
2. **功能：计算四个32位输入与一个进位输入的和，并输出求和结果以及进位输出。**

（2）加法器展示模块adder\_display.v

module adder\_display(

//时钟与复位信号

input clk,

input resetn, //后缀"n"代表低电平有效

//拨码开关，用于选择输入数和产生cin

input [1:0]input\_sel,

//0:输入为加数1(add\_operand1);1:输入为加数2(add\_operand2);2:输入为加数3(add\_operand3);1:输入为加数4(add\_operand4);

input [1:0]sw\_cin,

//led灯，用于显示cout

output [1:0] led\_cout,

//触摸屏相关接口，不需要更改

output lcd\_rst,

output lcd\_cs,

output lcd\_rs,

output lcd\_wr,

output lcd\_rd,

inout[15:0] lcd\_data\_io,

output lcd\_bl\_ctr,

inout ct\_int,

inout ct\_sda,

output ct\_scl,

output ct\_rstn

);

//-----{调用加法模块}begin

reg [31:0] adder\_operand1;

reg [31:0] adder\_operand2;

reg [31:0] adder\_operand3;

reg [31:0] adder\_operand4;

wire [1:0] adder\_cin;

wire [31:0] adder\_result ;

wire [1:0] adder\_cout;

adder32 adder\_module(

.operand1(adder\_operand1),

.operand2(adder\_operand2),

.operand3(adder\_operand3),

.operand4(adder\_operand4),

.cin (adder\_cin ),

.result (adder\_result ),

.cout (adder\_cout )

);

assign adder\_cin = sw\_cin;

assign led\_cout = adder\_cout;

//-----{调用加法模块}end

//---------------------{调用触摸屏模块}begin--------------------//

//-----{实例化触摸屏}begin

//此小节不需要更改

reg display\_valid;

reg [39:0] display\_name;

reg [31:0] display\_value;

wire [5 :0] display\_number;

wire input\_valid;

wire [31:0] input\_value;

lcd\_module lcd\_module(

.clk (clk ), //10Mhz

.resetn (resetn ),

//调用触摸屏的接口

.display\_valid (display\_valid ),

.display\_name (display\_name ),

.display\_value (display\_value ),

.display\_number (display\_number),

.input\_valid (input\_valid ),

.input\_value (input\_value ),

//lcd触摸屏相关接口，不需要更改

.lcd\_rst (lcd\_rst ),

.lcd\_cs (lcd\_cs ),

.lcd\_rs (lcd\_rs ),

.lcd\_wr (lcd\_wr ),

.lcd\_rd (lcd\_rd ),

.lcd\_data\_io (lcd\_data\_io ),

.lcd\_bl\_ctr (lcd\_bl\_ctr ),

.ct\_int (ct\_int ),

.ct\_sda (ct\_sda ),

.ct\_scl (ct\_scl ),

.ct\_rstn (ct\_rstn )

);

//-----{实例化触摸屏}end

//-----{从触摸屏获取输入}begin

//根据实际需要输入的数修改此小节，

//建议对每一个数的输入，编写单独一个always块

//当input\_sel为0时，表示输入数为加数1，即operand1

always @(posedge clk)

begin

if (!resetn)

begin

adder\_operand1 <= 32'd0;

end

else if (input\_valid && input\_sel==0)

begin

adder\_operand1 <= input\_value;

end

end

//当input\_sel为1时，表示输入数为加数2，即operand2

always @(posedge clk)

begin

if (!resetn)

begin

adder\_operand2 <= 32'd0;

end

else if (input\_valid && input\_sel==1)

begin

adder\_operand2 <= input\_value;

end

end

//当input\_sel为2时，表示输入数为加数3，即operand3

always @(posedge clk)

begin

if (!resetn)

begin

adder\_operand3 <= 32'd0;

end

else if (input\_valid && input\_sel==2)

begin

adder\_operand3 <= input\_value;

end

end

//当input\_sel为3时，表示输入数为加数4，即operand4

always @(posedge clk)

begin

if (!resetn)

begin

adder\_operand4 <= 32'd0;

end

else if (input\_valid && input\_sel==3)

begin

adder\_operand4 <= input\_value;

end

end

//-----{从触摸屏获取输入}end

//-----{输出到触摸屏显示}begin

//根据需要显示的数修改此小节，

//触摸屏上共有44块显示区域，可显示44组32位数据

//44块显示区域从1开始编号，编号为1~44，

always @(posedge clk)

begin

case(display\_number)

6'd1 :

begin

display\_valid <= 1'b1;

display\_name <= "ADD\_1";

display\_value <= adder\_operand1;

end

6'd2 :

begin

display\_valid <= 1'b1;

display\_name <= "ADD\_2";

display\_value <= adder\_operand2;

end

6'd3 :

begin

display\_valid <= 1'b1;

display\_name <= "ADD\_3";

display\_value <= adder\_operand3;

end

6'd4 :

begin

display\_valid <= 1'b1;

display\_name <= "ADD\_4";

display\_value <= adder\_operand4;

end

6'd5 :

begin

display\_valid <= 1'b1;

display\_name <= "RESUL";

display\_value <= adder\_result;

end

default :

begin

display\_valid <= 1'b0;

display\_name <= 40'd0;

display\_value <= 32'd0;

end

endcase

end

//-----{输出到触摸屏显示}end

//----------------------{调用触摸屏模块}end---------------------//

endmodule

1. **代码修改：修改部分用红色标注。大致分为四部分**

**①选择信号、sw\_cin、led\_out：均修改为两位**

**②调用加法模块的部分：匹配加法器的修改**

**③从触摸屏获取输入：添加加数3和4的部分**

**④输出到触摸屏显示：1,2,3,4为加数、5为结果result**

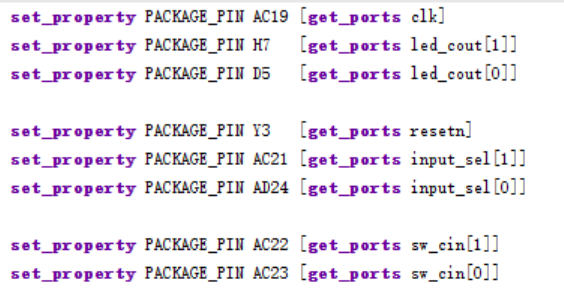
1. **功能：根据选择信号，输入不同的加数，并调用加法模块得到结果，同时实时显示各个数。**

（3）仿真tb.v



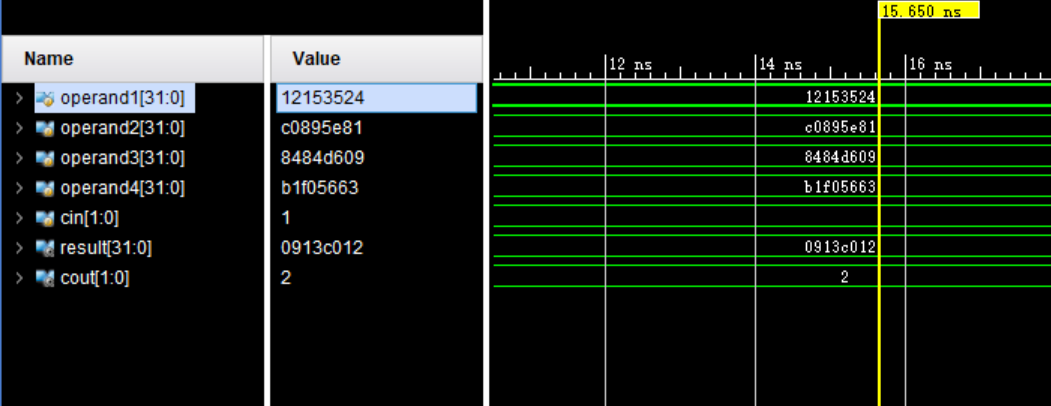
1. **代码修改：添加了两个寄存器（operand3,operand4），将进位输入和输出修改为两位，添加了两个随机数，并修改cin的生成范围。**
2. **功能：可以实现四个32为数的加法器实验的仿真模拟**

（4）约束文件mycons.xdc

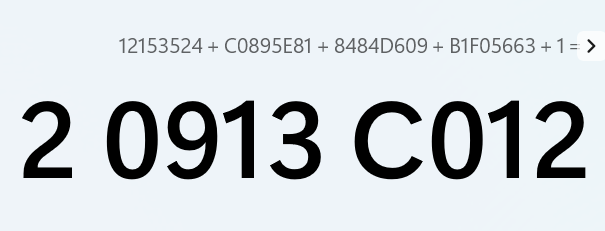


1. **代码修改：led输出绑定两个，选择信号匹配两个拨码开关，进位输入用两个拨码开关实现。**
2. **功能：实现了实验箱上led灯、拨码开关、触摸屏的连接，程序因而可以在实验箱上运行。**
3. **实验结果分析**

（1）仿真实验

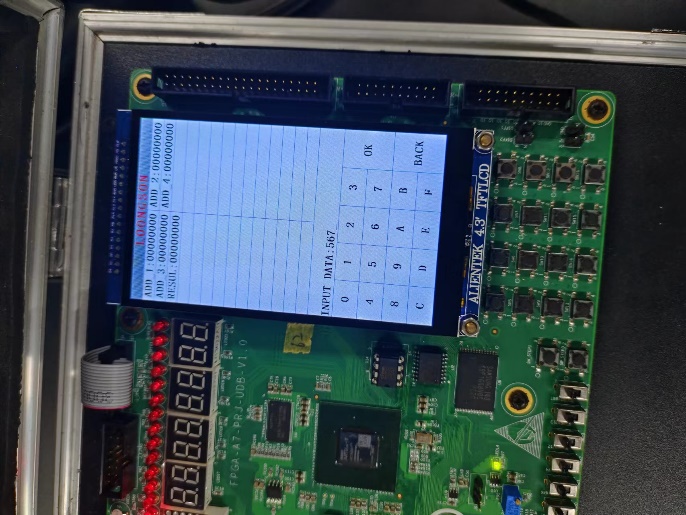
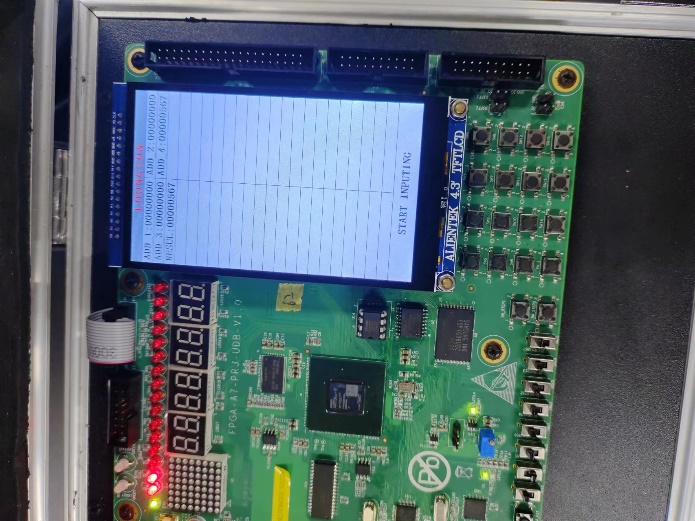


**0x12153524+0xc0895e81+0x8484d609+b1f05663+1=0x0913c012(进位为2)**

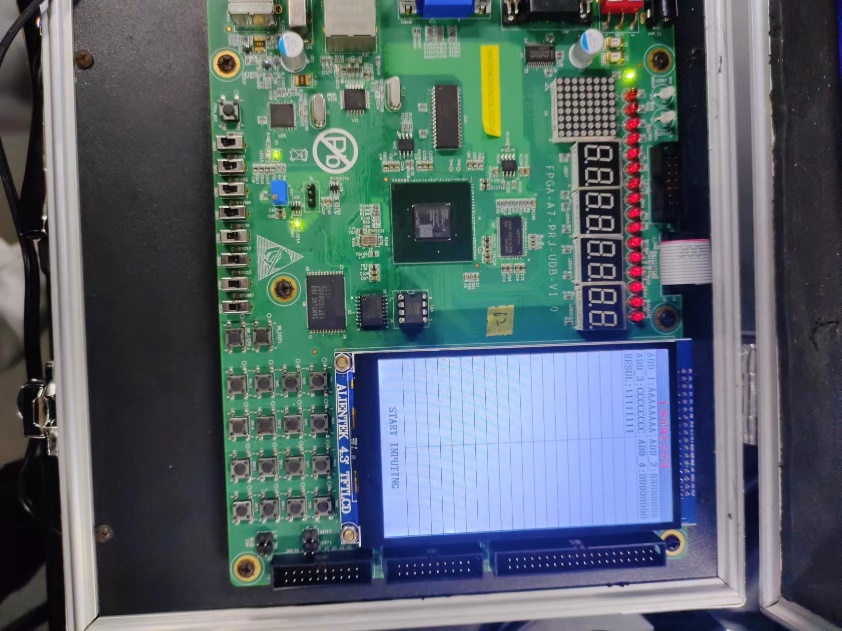
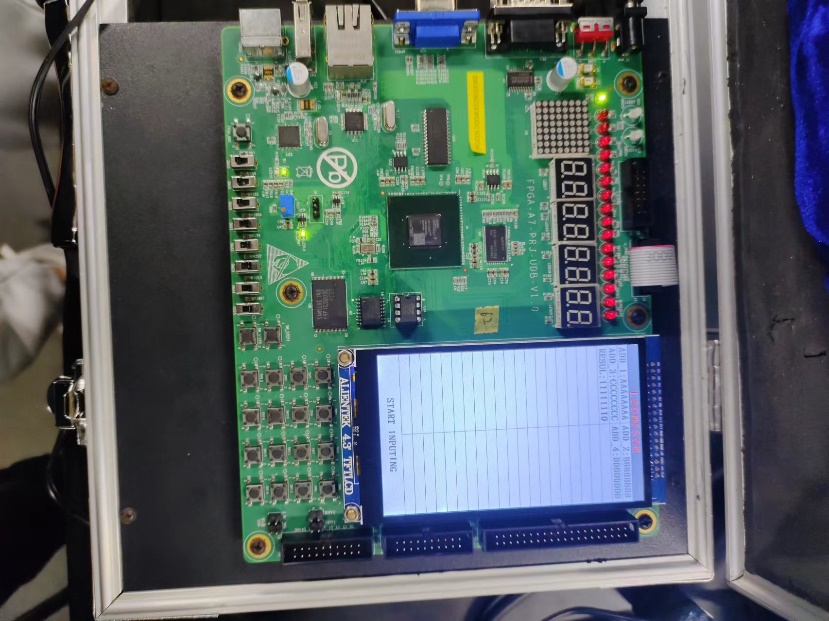
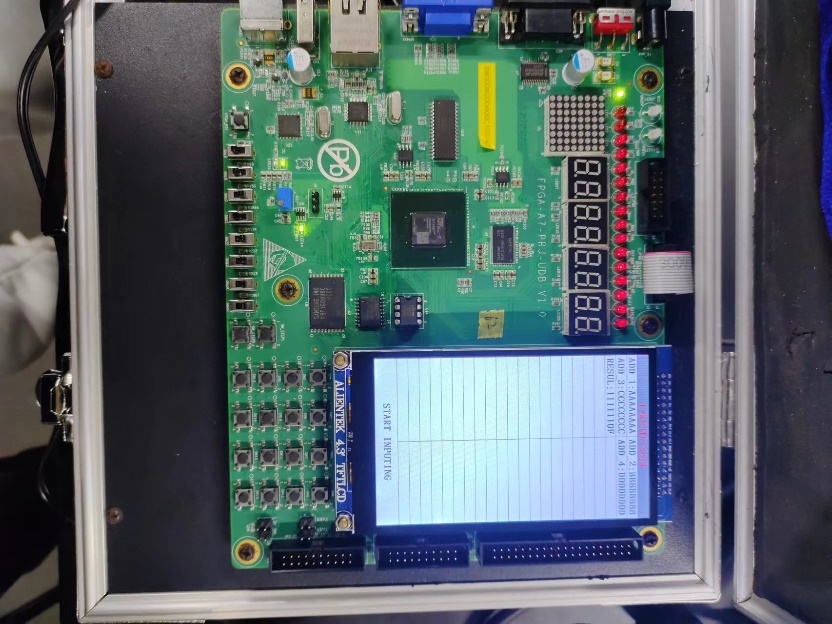
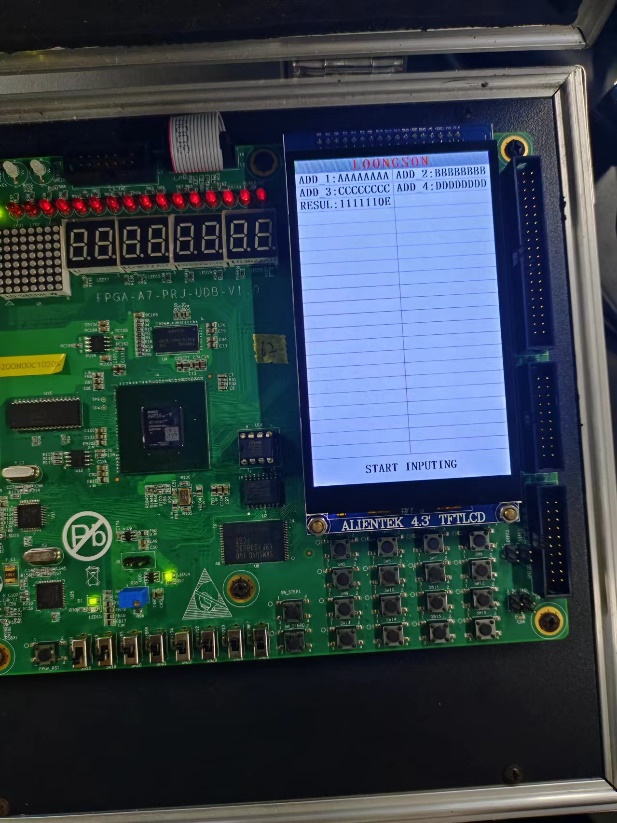


计算正确，说明仿真模拟实验成功。

（2）上箱实验

说明输入没有问题



**加数为AAAAAAAA,BBBBBBBB,CCCCCCCC,DDDDDDDD,进位输入分别为0,1,2,3**

显然结果无误，实验成功

1. **总结感想**

这次实验让我深入了解了在CPU内部运算中的一些重要操作，特别是对加法器进行修改和调整。通过这个过程，我进一步掌握了使用Verilog语言进行硬件描述和编程的基本技能。在调整显示屏显示位置的过程中，我遇到了一些问题，但通过仔细研究代码并修正错误，最终成功解决了问题。这次实验也让我更加熟悉了实验箱的布局结构和使用方法，提高了我的实验技能。

总的来说，这次实验对我来说是一次很好的学习和实践机会，让我在Verilog语言的应用和硬件实验方面都有了进一步的提升。我相信这些经验和技能将对我的未来学习和工作有很大的帮助。