**组成原理实验课程第1次实报告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 实验名称 | 数据运算：定点加法 | | | 班级 | 李涛 |
| 学生姓名 | 张丛 | 学号 | 2113662 | 指导老师 | 董前琨 |
| 实验地点 | 实验楼A区308 | | 实验时间 | 2023.03.21 | |

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**

1. 做好预习：

1） 了解软硬件平台；

2） 掌握定点加法的工作原理；

3） 确定定点加法的输入输出端口设计；

4） 在课前画好设计框图或实验原理图；

5） 如果对 FPGA 板了解的话，可确定设计中与 FPGA 板上交互的接口，画出

包含外围模块的整体设计框图。

2. 实验实施：

1） 确认定点加法的设计框图的正确性；

2） 编写 verilog 代码；

3） 对该模块进行仿真，得出正确的波形，截图作为实验报告结果一项的材

料；

4） 完成调用定点加法模块的外围模块的设计，并编写代码；

5） 对代码进行综合布局布线下载到实验箱里 FPGA 板上，进行上板验证。

3. 实验检查：

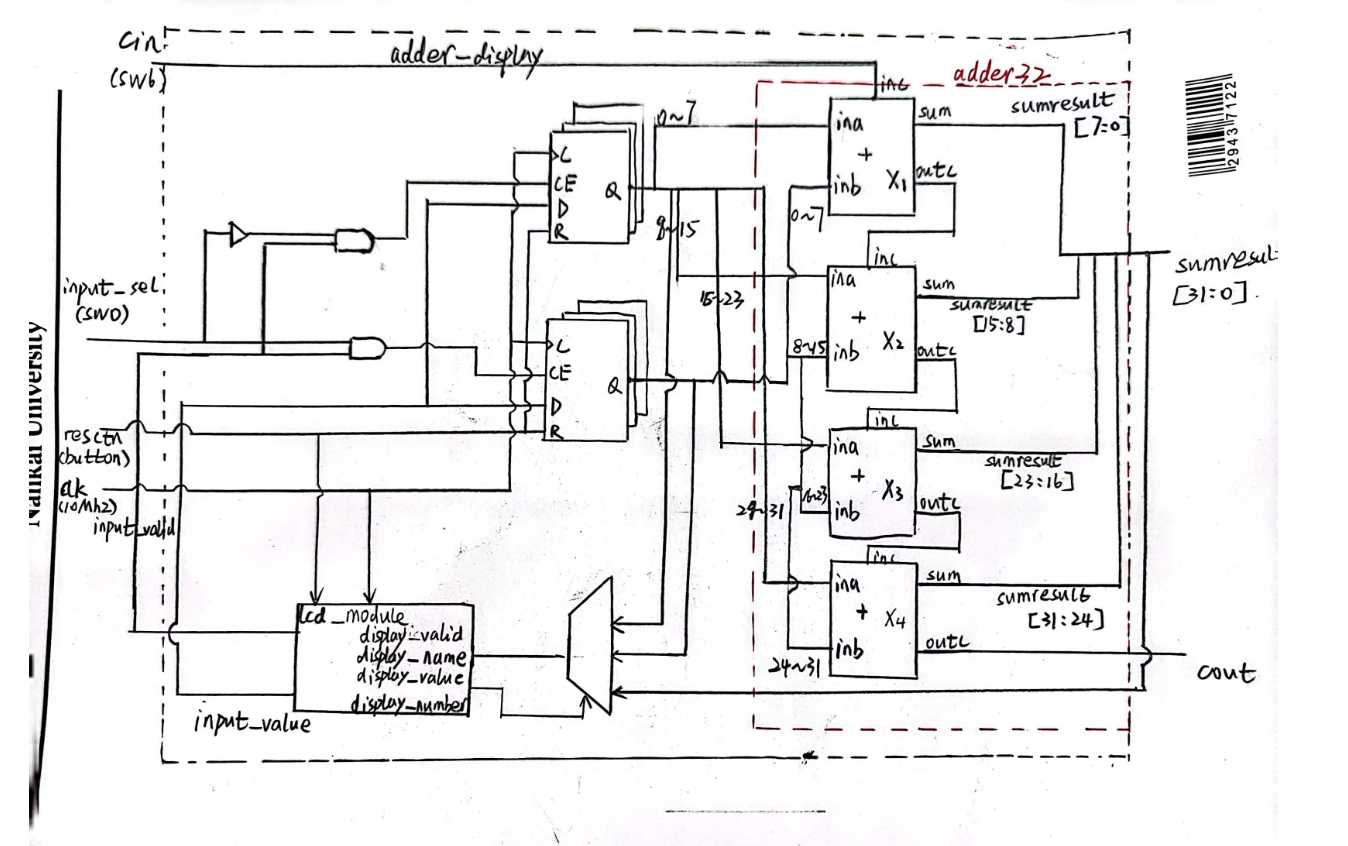
1） 完成上板验证后，让指导老师或助教进行检查，进行现场演示，可对演

示结果进行拍照作为实验报告结果一项的材料。

4. 实验报告的撰写：

1） 实验结束后，需按照规定的格式完成实验报告的撰写。

1. **实验原理图**



1. **实验步骤**
2. 编写8位全加器：

module adder8(

input [7:0] ina,

input [7:0] inb,

input inc,

output [7:0] sum,

output outc

);

assign {outc, sum} = ina +inb+inc;

Endmodule

模块中包含三个输入：两个8位加数以及一个进位；

两个输出：一个8位的和以及一个进位；

{}为位拼接运算符，将进位输出与和拼接在一起使用，表示一个整体信号。

1. 利用8位全加器实现32位全加器：

module adder32(a,b,cin,sumresult,cout);

input [31:0] a,b;

input cin;

output [31:0] sumresult;

output cout;

wire c1;

wire c2;

wire c3;

adder8 x1(

.ina(a[7:0]),

.inb(b[7:0]),

.inc(cin),

.sum(sumresult[7:0]),

.outc(c1)

);

adder8 x2(

.ina(a[15:8]),

.inb(b[15:8]),

.inc(c1),

.sum(sumresult[15:8]),

.outc(c2)

);

adder8 x3(

.ina(a[23:16]),

.inb(b[23:16]),

.inc(c2),

.sum(sumresult[23:16]),

.outc(c3)

);

adder8 x4(

.ina(a[31:24]),

.inb(b[31:24]),

.inc(c3),

.sum(sumresult[31:24]),

.outc(cout)

);

Endmodule

此模块利用四个8位全加器实现32位全加器，其中低8位的八位全加器将输出进位（outc）传递为高8位的输入进位(inc)。

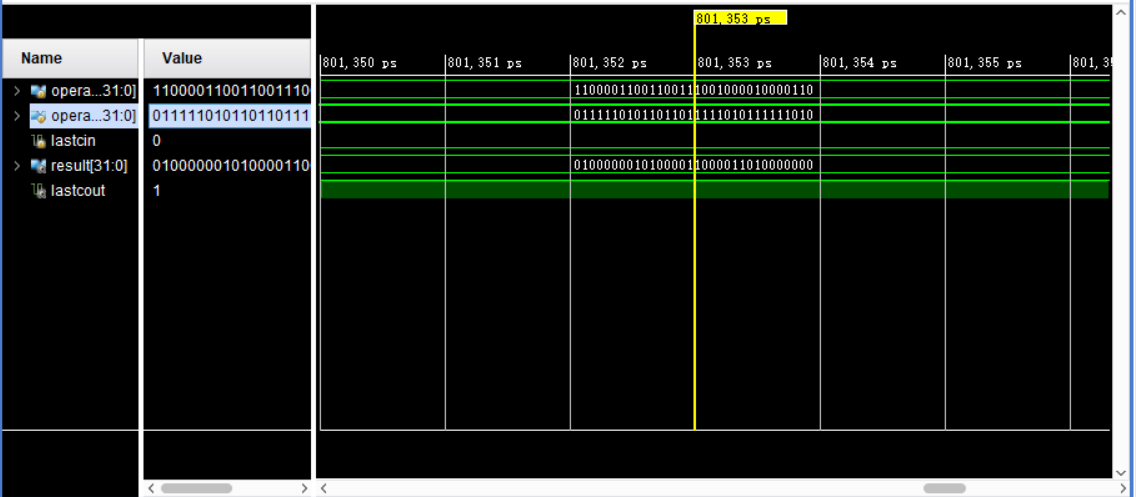
代码中,包含三个输入：32位加数a、b以及进位cin；

两个输出：32位和sumresult以及进位cout；

三个wire变量c1、c2、c3用于连接四个八位全加器，四个全加器x1、x2、x3、x4分别计算[7:0][15:8][23:16][24:17][31:24]对应位数的加法结果，并将进位往高8位传递（最后的进位传递到cout中），将此8为的加法结果存储在sumresult对应位数中。

以上，实现了32位全加器。

1. 添加仿真文件，进行仿真实验
2. 添加展示外围模块，添加约束文件，连接实验箱进行上板验证
3. **实验结果分析**
4. 仿真实验

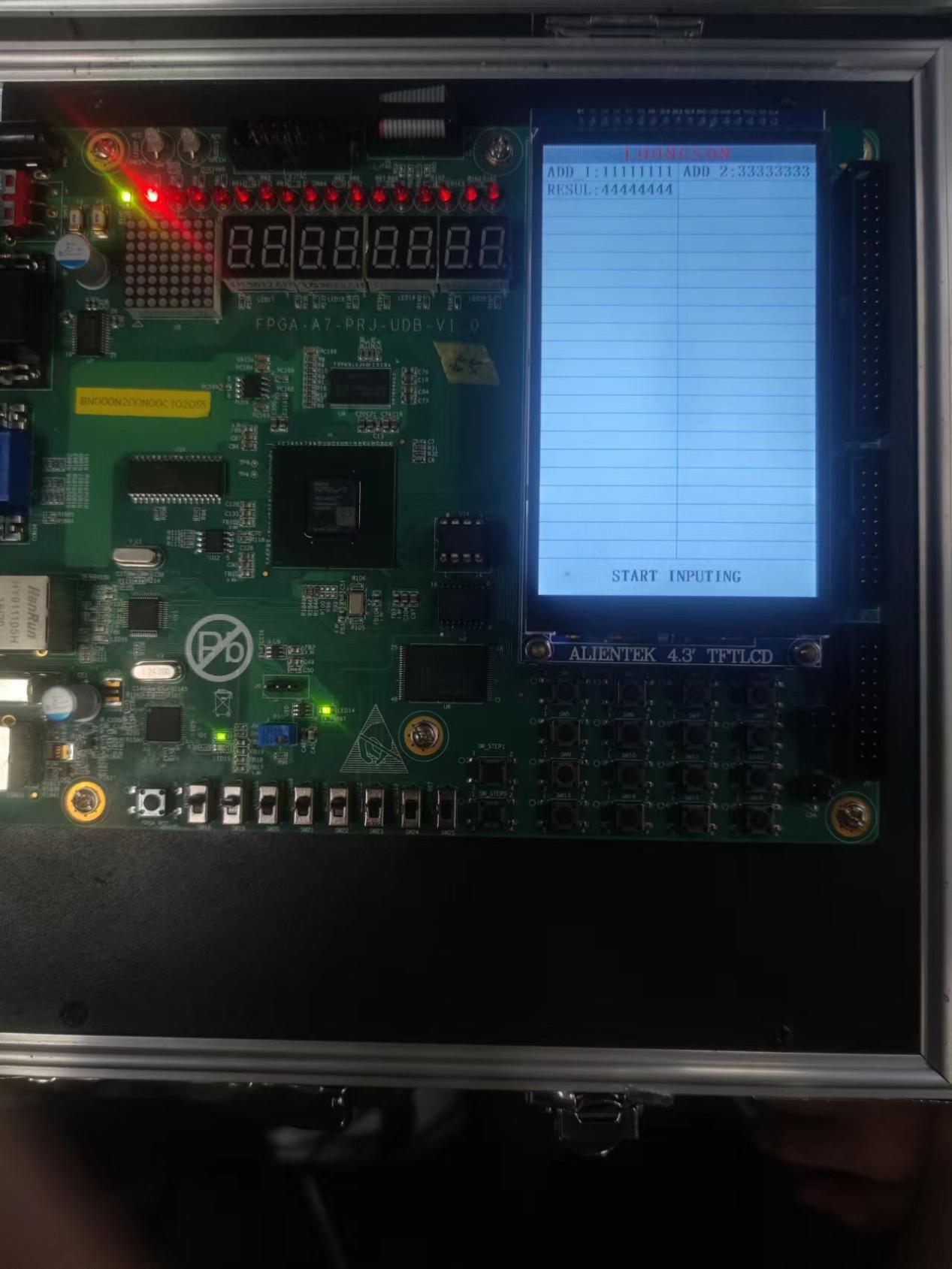


输入：加数operand1、operand2以及进位lastcin。

输出：和result，进位lastcout。

观察波形图以及二进制值，可知实现了32位全加器。

1. 上板验证



输入：ADD\_1(11111111),ADD\_2(33333333)

结果：RESUL(44444444)

可知结果是正确的，实验是成功的。

1. **总结感想**

从0开始的verilog学习生活，刚开始接触肯定是很难的，要学习语法学习结构等等。而且课上学习了如何编写8为全加器，但用8位实现32位还是很难的。

除了verilog语言的问题，学习vivado软件也是很重要的，如何编写文件、导入文件，如何添加展示外围模块、如何添加约束文件，如何仿真、如何连接实验箱、如何在实验箱操作。总之步骤很多很杂，但一步步学习也挺有乐趣的，最终能成功实验也让人高兴。

相信以后的实验会比第一次难，但我也相信我会对verilog语言越来越熟练、对实验软件的操作越来越熟练。

另外，估计还得复习一下数字逻辑。