计算机组成原理第 1 次实验报告

实验名称：加法器的改进——实现 32 位减法器

学号：2313546　　姓名：蒋枘言　　班次：1078

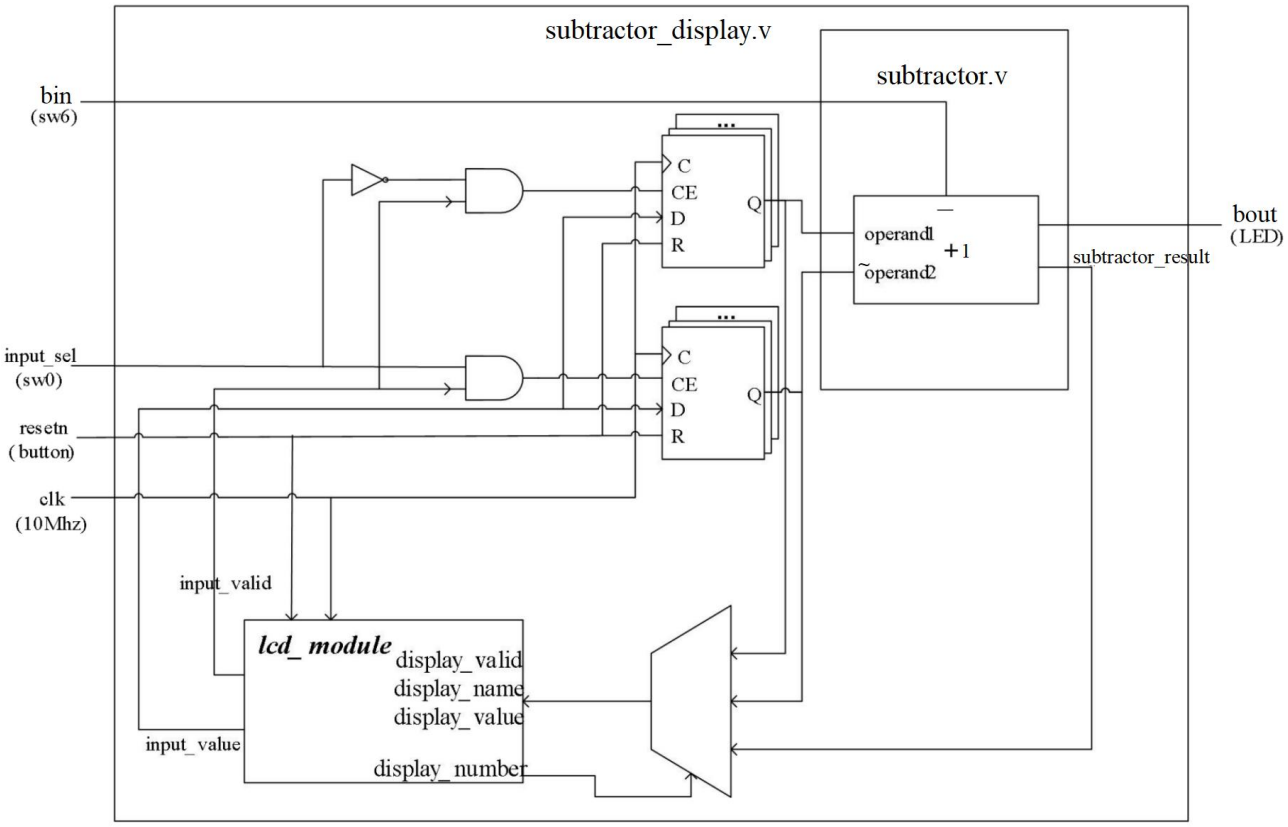
一、实验目的

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。

二、实验内容说明

将定点加法器的实验进行改进，实现一个减法器。注意：减法器拿到的 32 位输入都应该理解成数据的补码形式，最终得到差的补码，进行输出。

三、实验原理图



四、实验步骤

创建 design sources，创建源文件 subtractor.v，输入以下 Verilog 代码，实现一个 32 位全减器。

```
module subtractor(
    input   [31:0] operand1, //  被减数
    input   [31:0] operand2, //  减数
    input bin, //  借位输入（来自低位）
    output [31:0] result, //  差
    output bout //  借位输出（向高位）
    );
    assign {bout,result} = operand1 + ~operand2 + 1 - bin;
endmodule
```

介绍模块功能：

●输入：

operand1（32 位）：被减数。

operand2（32 位）：减数。

bin（1 位）：来自低位的借位输入。

●输出：

result（32 位）：减法的结果（差）。

bout（1 位）：向高位的借位输出。

{bout, result}将 33 位运算结果拆分为 bout（最高位，即借位标志）和 result（低 32 位，即减法得到的的差）。

●核心逻辑：assign {bout,result} = operand1 + ~operand2 + 1 - bin;（实现补码减法）

operand1-operand2 在硬件中通过**补码**加法实现，即等价于 operand1+(~operand2+1)。这里的~operand2 是对减数按位取反得到反码，+1 即生成补码。

●借位处理：若 bin 为 1，表示需要额外减去 1（来自低位的借位）。若 operand1 +~operand2+1-bin 的结果为负（需借位），bout 为 0；若结果非负（无需借位），bout 为 1。

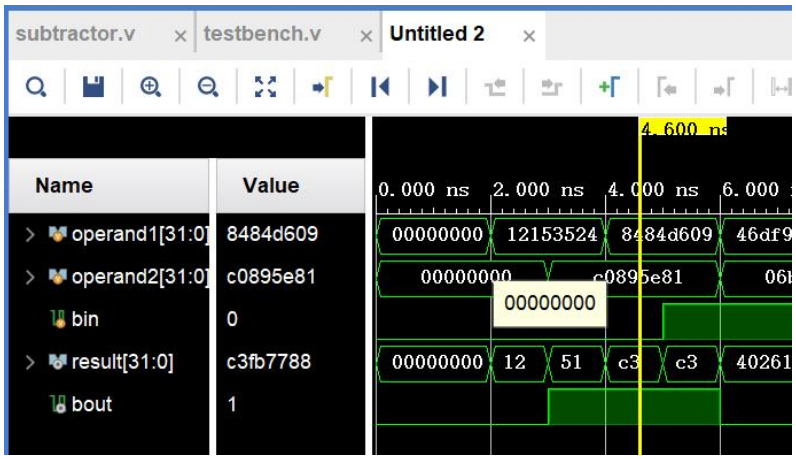
五、实验结果分析

使用两种方式进行验证。

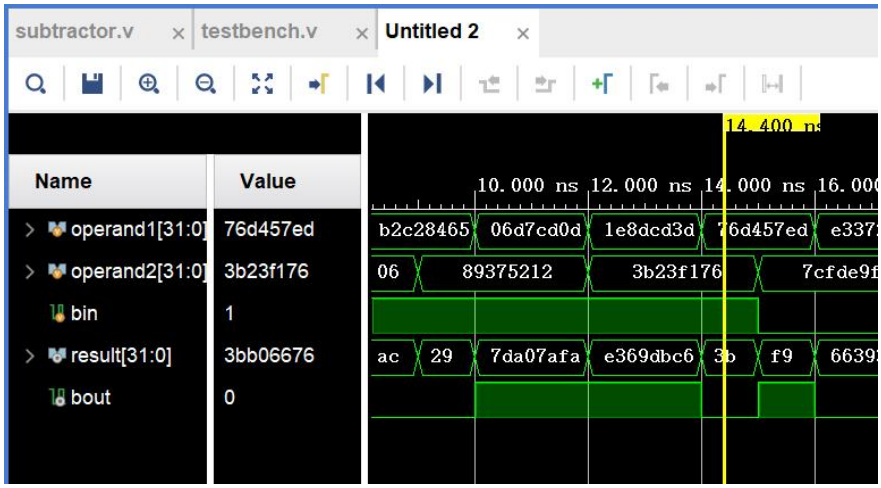
（1）通过**仿真实验**进行验证

创建 simulation sources，创建文件 testbench.v，用于仿真实验（此时 subtractor.v 应为 Design Sources 的顶层模块，testbench.v 应为 Simulation Sources 的顶层模块）。

写好 testbench.v 里的代码后，进行综合和实现（先点击 Run Synthesis，再点击 Run Implementation）。接着进行仿真（点击 Run Simulation），会自动出现一个波形图。随机选取几个时刻进行验证，如下图所示：



该时刻的被减数 operand1 为 8484d609（以下验证的数如未特殊说明均为**十六进制**），减数 operand2 为 c0895e81，无借位 bin，相减 operand1-operand2=-3c048878，转换为无符号整形应为-3c048878+100000000=c3fb7788，与 result 一致，并且此时 bout 应该为 1，故结果正确。

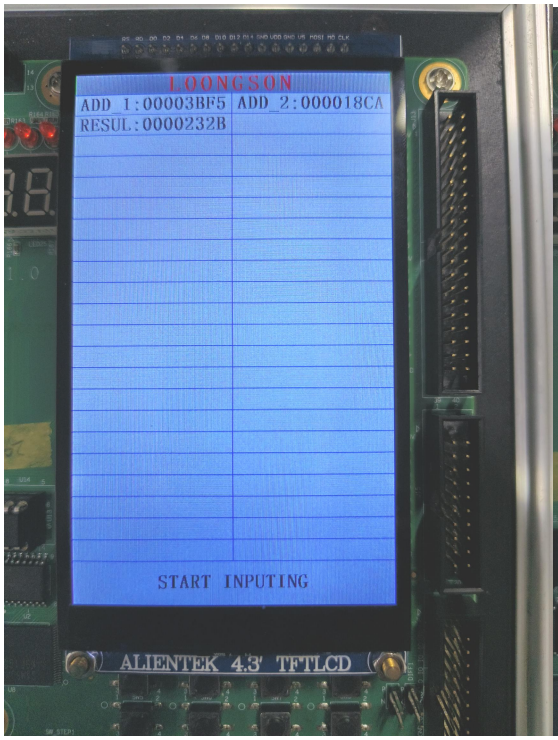


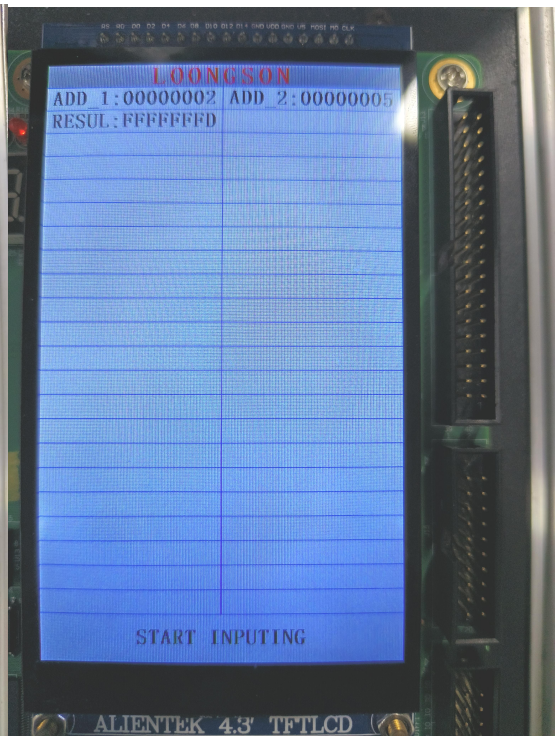
该时刻的被减数 operand1 为 76d457ed，减数 operand2 为 3b23f176，有借位 bin，相减 operand1-operand2-bin=3bb06676，与 result 一致，并且此时 bout 应该为 0，故结果正确。

（2）通过**实验箱**进行验证

创建 design sources，创建文件 subtractor_display.v；创建 constraints，创建文件 mycons.xdc；再将 lcd_module.dcp 添加进 Design Sources（此时 subtractor_display.v 应为 Design Sources 的顶层模块）；接着点击 Generate Bitstream，成功后通过 Open Target->Auto Connect 连接上实验箱。

在实验箱的显示屏上输入数据（最左边的拨片控制输入的是第几个操作数），进行验证，如下图所示。





（注意：此处两个操作数的名称忘记改了，由于直接复制的老师给的代码，所以仍然叫 ADD_1 和 ADD_2，但是不影响结果验证）（我的错）

00003BF5-000018CA	00000002-00000005
=0000232B	=-00000003，转化为无符号数为 FFFFFFFD

结果均正确！

六、总结感想

这是计算机组成原理的第一次实验，也是第一次接触到 Verilog 语言，刚开始还不太会，也不太会用 Vivado 软件，后来在老师的指导下逐渐入门（课下又把这几次实验课的视频回放看了一遍），目前已经会用 Vivado 写简单的 Verilog 代码设计简单的逻辑电路，感觉很有收获。