

组成原理实验课程第一次实报告

实验名称	加法器实验			班级	张金老师
学生姓名	杨冰雪	学号	2110508	指导老师	董前琨
实验地点	实验楼 A 区 306		实验时间	2023、3、13	

1、实验目的

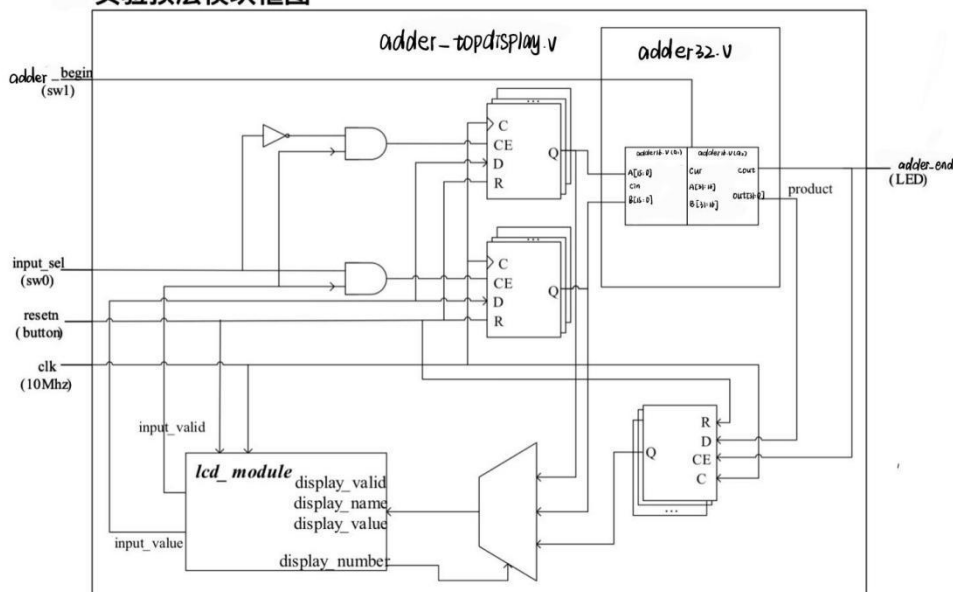
- 1.熟悉 LS-CPU-EXB-002 实验箱和软件平台。
- 2.掌握利用该实验箱各项功能开发组成原理和体系结构实验的方法。
- 3.理解并掌握 32 位加法器和 16 位加法器的原理和设计。
- 4.熟悉并掌握 Verilog 语言进行电路设计。
- 5.为后续设计 CPU 的实验打下基础。

2、实验内容说明

- 1.设计两个模块，16 位加法器和 32 位加法器，其中 32 位加法器通过调用 16 位加法器实现。
- 2.针对 32 位加法器进行仿真验证。
- 3.针对 32 位加法器进行上实验箱验证

3、实验原理图

实验顶层模块框图



adder32.v 模块里调用了两个 adder16.v 模块，第一个 adder16.v 实例化为 a1，其输入为两个[31:0]数的前 16 位，和一个进位，其计算结果为一个进位和一个 16 位的数，这个 16 位的数就是输出的[31:0]数的前 16 位结果，其进位作为下一个 adder16.v 实例化对象 a2 的进位，输入为两个[31:0]数的后 16 位，其计算结果同样为一个进位和一个 16 位的数，进位就是整个结果的进位，而 16 位数就是输出数的后 16 位，从而得到结果。

4、实验步骤

1. 设计一个 16 位的加法器。把实验给的 adder.v 的文件中输入和输出的数由 32 位改为 16 位，即可实现两个 16 位的数和进位相加得到结果。

```

module adder16(
    input[15:0]A,
    input[15:0]B,
    input cin,
    output[15:0]out,
    output cout
);
    assign {cout,out}=A + B + cin;
endmodule

```

2. 设计一个 32 位加法器。输入变为两个 32 位的数，通过调用两个 16 位加法器，得到 32 位输出数和进位。

```

module adder32(
    input[31:0]A,
    input[31:0]B,
    input cin,
    output[31:0] out,
    output cout
);
    wire cur;
    adder16 a1(A[15:0], B[15:0], cin, out[15:0], cur);
    adder16 a2(A[31:16], B[31:16], cur, out[31:16], cout);
endmodule

```

3. 对 32 位加法器进行仿真，先对[31:0]A、[31:0]B 和 cin 初始化为 0，再用 always 语句通过随机数给他们随机赋值，可以直接使用实验给的 testbench 文件。仿真可以验证我们设计的 32 加法器是否正确。

4. 创建顶层文件 adder_topdisplay.v, 里面包含了 32 位加法器模块和 lcd_module 模块，adder_topdisplay 和 lcd_module 都不需要修改。

5. 添加约束文件 adder.xdc，不需要修改。（注：以上所说的修改不包括变量名变化和模块名改变的修改）

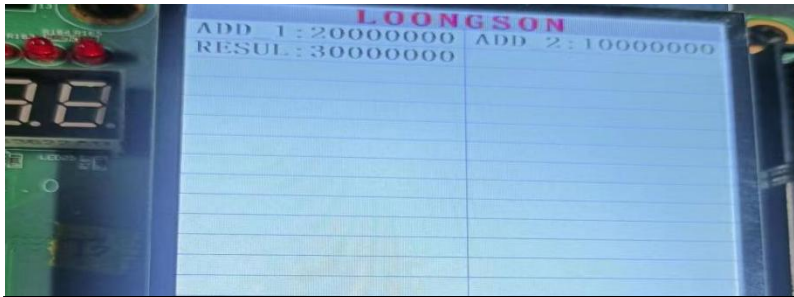
5、实验结果分析

Name	Value
> A[31:0]	00001010011011101001
> B[31:0]	10001001000110011011
> cin	0
> out[31:0]	10010011100010000100
cur	1
cout	0

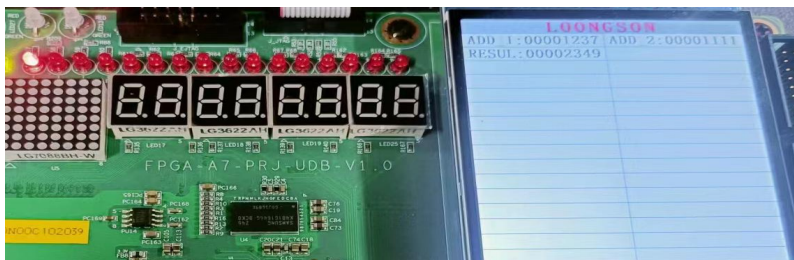
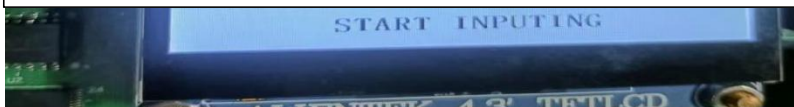
输入为 00001010011011101001001100010 和 10001001000110011011010000010，cin 为 0。

输出为 10010011100010000100011100100，cout 为 0。

结果正确。



输入为 20000000 和 10000000，cin 为 0
 结果为 30000000
 结果正确



输入为 00001237 和 00001111，cin 为 1。
 输出为 00002349。
 结果正确。



6、 总结感想

学会能够熟练使用 Verilog 语言写模块，能够独立创建项目并进行仿真验证和在试验箱上进行验证。
 并且学会使用顶层模块调用子模块，以前只是会直接写 32 位加法器来计算两个 32 位数，现在学会调用两个 16 位加法器来形成 32 位加法器，同样的还可以用两个 32 位加法器来形成 64 位加法器。