组成原理实验课程第__一__次实报告

实验名称	4 个 32 进制数的加法器			班级	李涛老师
学生姓名	田晋宇	学号	2212039	指导老师	董前琨
实验地点	实验楼 B306		实验时间	2024.03.21	

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

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：

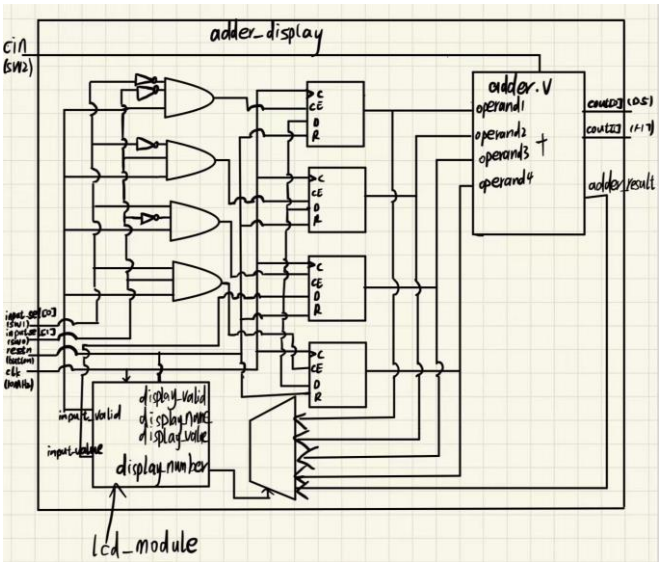
1、除了修改 adder 模块，display、testbench 和约束文件都有修改，注意加法器进位，四个数加法会出现 2 位的进位。

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。

4、实验报告模板参考百度云盘文件，注意提交截至时间为 3 月 28 日下午 18：00。

三．实验原理图



整个工程由顶层模块 adder_display.v 实现，其中包含了两个子模块，一个是实现加法器的 adder.v 模块，一个是完成 lcd 显示的 lcd_display.v 模块。

四．实验步骤

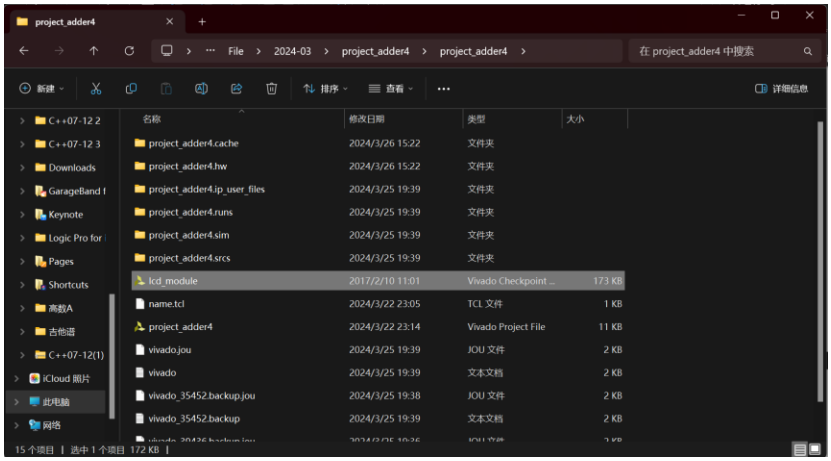
首先是对 adder.v 模块的修改，原代码实现的是两个 32 位二进制数的相加，因此我们对代码进行如下的修改，基本的思想是实现多个两个 32 位二进制数的相加的串并联：

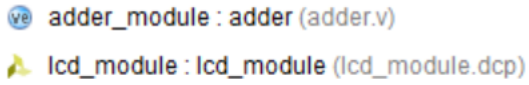
```
module adder(
    input [31:0] operand1,
    input [31:0] operand2,
    input [31:0] operand3,
    input [31:0] operand4,
    input cin,
    output [1:0]cout,
    output [31:0]result
    );
    wire [31:0] result1, result2;
    wire cout1, cout2, cout3;

    assign {cout1,result1} = operand1+operand2+cin;
    assign {cout2,result2} = operand3+operand4;
    assign {cout3,result} = result1+result2;
    assign cout=cout1+cout2+cout3;
endmodule
```

编写完 adder.v 模块之后我们便可以编写仿真文件对模块进行测试，操作数及进位信号 cin 均为随机生成，：

```
module testbench;
reg [31:0] operand1;
reg [31:0] operand2;
reg [31:0] operand3;
reg [31:0] operand4;
reg [1:0]cin;

wire [1:0]cout;
wire [31:0]result;

adder add1(.operand1(operand1),.operand2(operand2),.operand3(operand3),.operand4(operand4),.cin(cin),.cout(cout),.result(result));

initial begin
// Initialize Inputs
operand1 = 0;
operand2 = 0;
operand3 = 0;
operand4 = 0;
cin = 0;
// Wait 100 ns for global reset to finish
#100;
// Add stimulus here
end

always #10 operand1 = $random;
always #10 operand2 = $random;
always #10 operand3 = $random;
always #10 operand4 = $random;
always #10 cin = {$random} % 2;

endmodule
```

然后我们将原本预设好的 lcd_module 模块导入：





接着我们对顶层模块 adder_displar.v 进行改写：

```
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: 为加数2 (add_operand2)
    input 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
);
```

现在有四个操作数，因此在输入 input_sel 时需要用两位二进制数进行表示，00 表示输入第一个数，01 表示输入第二个数，10 表示输入第三个数，11 表示输入第四个数。对于四个 32 位二进制数的加法，有可能出现两位的进位，因此将 led_cout 也改为两位二进制数。

```
// ——— {调用加法模块}
reg [31:0] adder_operand1;
reg [31:0] adder_operand2;
reg [31:0] adder_operand3;
reg [31:0] adder_operand4;
wire adder_cin;
wire [31:0] adder_result;
wire [1:0]adder_cout;
adder 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;
```

在调用加法器模块时，要求输入四个 32 位二进制数，将所有操作数都添加入内。

```
// 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'd2) 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==2'd3) begin
        adder_operand4 <= input_value;
    end
end
```

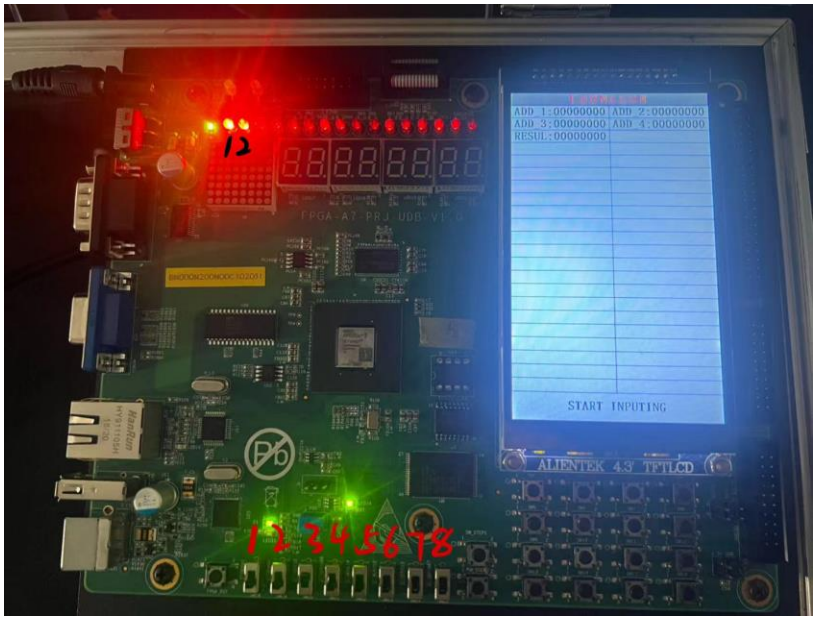
在接下来的条件判断中，增加以上代码，如果拨码开关拨向 10 和 11，将分

别进行 operand3 和 operand4 的输入。

```
6'd3: begin
    display_valid <= 1'b1;
    display_name <= "ADD_3";
    display_value <= adder_operand2;
end
6'd4: begin
    display_valid <= 1'b1;
    display_name <= "ADD_4";
    display_value <= adder_operand2;
end
6'd5: begin
    display_valid <= 1'b1;
    display_name <= "RESUL";
    display_value <= adder_result;
end
```

Lcd 屏幕显示的部分增加以上代码，使其对 operand3 和 operand4 进行显示。

接下来完成对约束文件的修改，我们用到实验箱的接口为拨码开关 1，2，3，1 号和 2 号拨码开关用于表示输入的操作数，3 号拨码开关用于表示输入的进位信号。Led 灯用到 1 号和 2 号来表示输出的进位信号，通过查询引脚对应关系表可知，我们需要用到的对应编号分别为 H7，D5，AC21，AC24，AC22。



13	**LED灯**		
14	FPGA_LED1	H7	实验板放正，一排LED灯左起第一个。
15	FPGA_LED2	D5	依次类推
16	FPGA_LED3	A3	
17	FPGA_LED4	A5	
18	FPGA_LED5	A4	
19	FPGA_LED6	F7	
20	FPGA_LED7	G8	
21	FPGA_LED8	H8	
22	FPGA_LED9	J8	
23	FPGA_LED10	J23	
24	FPGA_LED11	J26	
25	FPGA_LED12	G9	
26	FPGA_LED13	J19	
27	FPGA_LED14	H23	
28	FPGA_LED15	J21	
29	FPGA_LED16	K23	实验板放正，一排LED灯左起第八个。
30			
31	**拨码开关**		实验板放正，拨下为1，拨上为0
32	FPGA_SW0	AC21	实验板放正，一排拨码开关左起第一个。
33	FPGA_SW1	AD24	依次类推
34	FPGA_SW2	AC22	
35	FPGA_SW3	AC23	
36	FPGA_SW4	AB6	
37	FPGA_SW5	W6	
38	FPGA_SW6	AA7	
39	FPGA_SW7	Y6	实验板放正，一排拨码灯左起第八个。

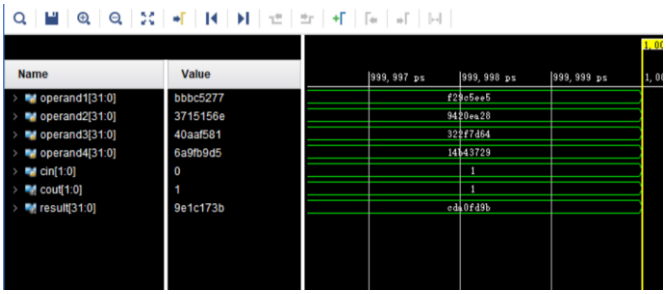
对约束文件进行修改：

adder.v | testbench.v | adder_display.v * | adder.xdc | Untitled 5

```
set_property PACKAGE_PIN AC19 [get_ports clk]
set_property PACKAGE_PIN H7   [get_ports led_cout[1]]
set_property PACKAGE_PIN D5   [get_ports led_cout[0]]
set_property PACKAGE_PIN Y3   [get_ports resetn]
set_property PACKAGE_PIN AC21 [get_ports input_sel[1]]
set_property PACKAGE_PIN AD24 [get_ports input_sel[0]]
set_property PACKAGE_PIN AD22 [get_ports sw_cin]

set_property IOSTANDARD LVCMOS33 [get_ports clk]
set_property IOSTANDARD LVCMOS33 [get_ports led_cout[1]]
set_property IOSTANDARD LVCMOS33 [get_ports led_cout[0]]
set_property IOSTANDARD LVCMOS33 [get_ports resetn]
set_property IOSTANDARD LVCMOS33 [get_ports input_sel[1]]
set_property IOSTANDARD LVCMOS33 [get_ports input_sel[0]]
set_property IOSTANDARD LVCMOS33 [get_ports sw_cin]
```

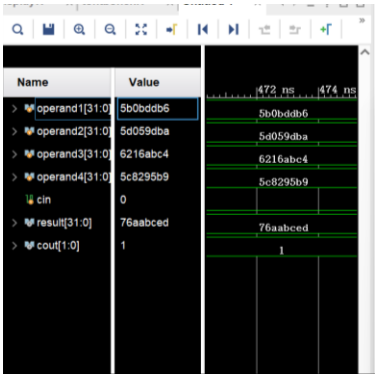
五．实验结果分析

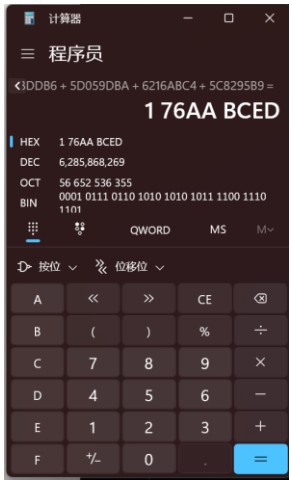
对 testbench.sim 文件进行仿真测试后得到的波形图如下：



通过计算验证波形图结果正确无误：

对如下在实验箱上进行验证：





在实验箱上通过对拨码开关的控制，依次输入 operand1,2,3,4,以及进位信号，计算结果与仿真结果相同，led 灯 2 号亮起，实验成功。

五．总结感想

本次实验是对上学期学习 verilog 知识的巩固，理解了加法器的原理，熟悉了 verilog 语言的语法和编程逻辑，对多模块工程的编写有了更好的掌握。熟悉 LS-CPU-EXB-002 实验箱和软件平台将程序烧录在试验箱的过程设计对代码的修改以及对硬件设备的操作，为之后的 cpu 设计实验打下良好基础。

注意： 1、班级用任课老师姓名表示，分别是李涛老师、张金老师。

2、实验报告提交的文件名为“学号_姓名_组成原理第一次实验.pdf”，注意要导出成 pdf 文件。