

组成原理实验课程第一次实验报告

实验名称	加法器改进			班级	张金老师班
学生姓名	姚知言	学号	2211290	指导老师	董前琨
实验地点	实验楼 A308		实验时间	2024 年 3 月 22 日	

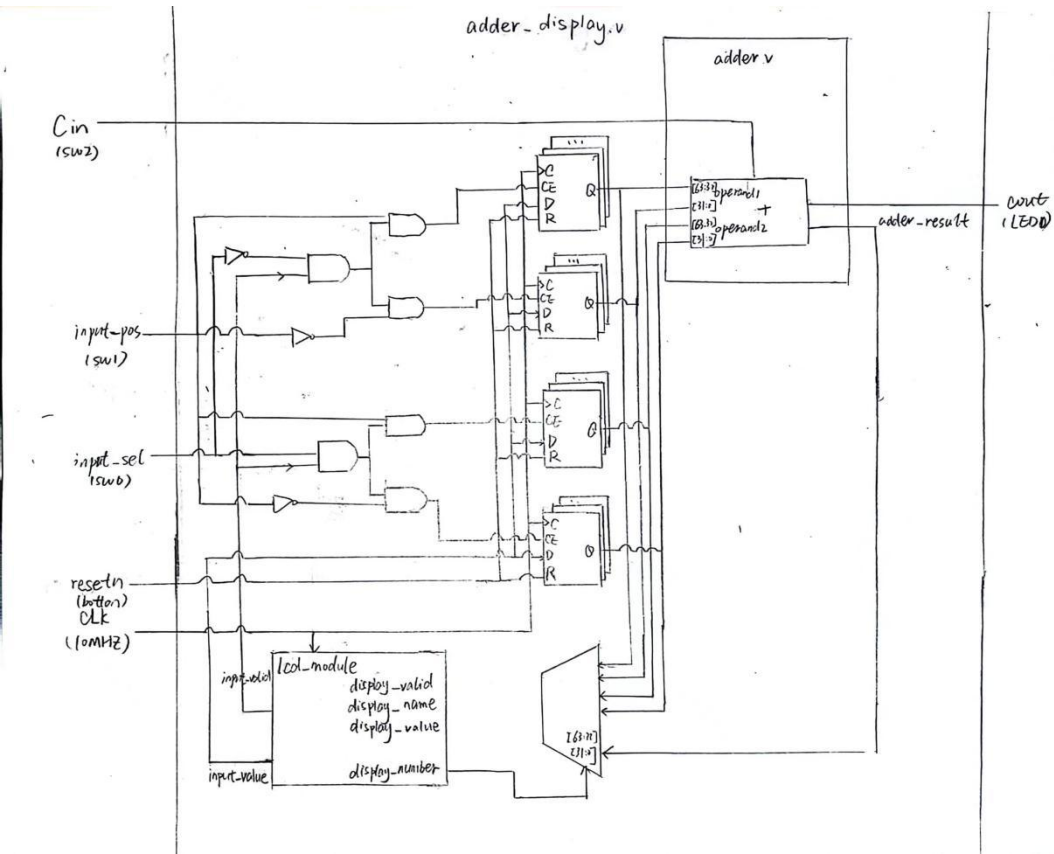
1、实验目的

- 熟悉 LS-CPU-EXB-002 实验箱和软件平台。
- 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。
- 理解并掌握加法器的原理和设计。
- 熟悉并运用 verilog 语言进行电路设计。
- 为后续设计 cpu 的实验打下基础。

2、实验内容说明

- 了解软硬件平台；
- 掌握定点加法的工作原理；
- 确定定点加法的输入输出端口设计；
- 画好顶层模块框图；
- 编写 verilog 代码；
- 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；
- 完成调用定点加法模块的外围模块的设计，并编写代码；
- 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；
- 撰写实验报告。

3、实验原理图



输入端：如图所示，在 `resetn`，`input_valid` 满足输入条件的情况下，通过控制 `input_sel` 和 `input_pos` 分别控制向哪一个加数输入，输入到高位还是低位。当 `input_sel` 为 0 的时候，向加数 1 输入，否则向加数 2 输入。当 `input_pos` 为 0 的时候，向低 32 位输入，否则向高 32 位输入。

输出端：1-6 号屏幕分别输出加数 1 的高 32 位，低 32 位，加数 2 的高 32 位，低 32 位，和的高 32 位，低 32 位。

4、实验步骤

(1) `adder.v`

```
module adder(
    input  [63:0] operand1,
    input  [63:0] operand2,
    input          cin,
    output [63:0] result,
    output          cout
);
    assign {cout,result} = operand1 + operand2 + cin;
endmodule
```

//主要对位宽进行了调整。

(2) `testbench.v`

```
.....
    reg [63:0] operand1;
    reg [63:0] operand2;
    reg cin;
    wire [63:0] result;
    wire cout;
//对位宽进行了调整
.....
    always #10 operand1[63:32] = $random;
    always #10 operand2[63:32] = $random;
    always #10 operand1[31:0] = $random;
    always #10 operand2[31:0] = $random;
```

//由于 random 只能生成 32 位的结果，需要对前 32 位和后 32 位分别生成随机数

.....

(3) `adder_display.v`

```
.....
    input clk,
    input resetn,
    input input_sel, //0:输入为加数 1(add_operand1);1:输入为加数 2(add_operand2)
    input input_pos, //0:输入为低位([31:0]);1:输入为高位([63:32])
    input sw_cin,
    output led_cout,
    //增加了判断输入高低位的拨码开关
.....
    reg [63:0] operand1;
```

```

reg [63:0] operand2;
reg cin;
wire [63:0] result;
wire cout;
//调整了位宽
.....
always @(posedge clk)
begin
    if (!resetn)
    begin
        adder_operand1 <= 64'd0;//调整了位宽
    end
    else if (input_valid && !input_sel)
    begin
        if(!input_pos)//选择传值到高位还是低位
        begin
            adder_operand1[31:0] <= input_value;
        end
        else
        begin
            adder_operand1[63:32] <= input_value;
        end
    end
end
always @(posedge clk)
begin
    if (!resetn)
    begin
        adder_operand2 <= 64'd0;//调整了位宽
    end
    else if (input_valid && input_sel)
    begin
        if(!input_pos)//选择传值到高位还是低位
        begin
            adder_operand2[31:0] <= input_value;
        end
        else
        begin
            adder_operand2[63:32]<= input_value;
        end
    end
end
always @(posedge clk)
begin

```

```

case(display_number)
  6'd1 :
  begin
    display_valid <= 1'b1;
    display_name  <= "ADD1H";
    display_value <= adder_operand1[63:32];
  end
  6'd2 :
  begin
    display_valid <= 1'b1;
    display_name  <= "ADD1L";
    display_value <= adder_operand1[31:0];
  end
  6'd3 :
  begin
    display_valid <= 1'b1;
    display_name  <= "ADD2H";
    display_value <= adder_operand2[63:32];
  end
  6'd4 :
  begin
    display_valid <= 1'b1;
    display_name  <= "ADD2L";
    display_value <= adder_operand2[31:0];
  end
  6'd5 :
  begin
    display_valid <= 1'b1;
    display_name  <= "RESUH";
    display_value <= adder_result[63:32];
  end
  6'd6 :
  begin
    display_valid <= 1'b1;
    display_name  <= "RESUL";
    display_value <= adder_result[31:0];
  end //选通 32 位高低位输出和高低位操作数
  default :
  begin
    display_valid <= 1'b0;
    display_name <= 40'd0;
    display_value <= 64'd0;//这个当时改了，但是现在感觉反而应该保持
  end
end

```

32 位，只是做实验的时候没发现也没有造成什么问题，就暂且保留在这里

```

        endcase
    end
endmodule

```

(4) mycons.xdc

```

set_property PACKAGE_PIN AC19 [get_ports clk]
set_property PACKAGE_PIN H7 [get_ports led_cout]
set_property PACKAGE_PIN Y3 [get_ports resetn]
set_property PACKAGE_PIN AC21 [get_ports input_sel]
set_property PACKAGE_PIN AD24 [get_ports input_pos]//为新增的 input pos 分配开关
set_property PACKAGE_PIN AC22 [get_ports sw_cin]//将 cin 的开关调整到第三个

```

```

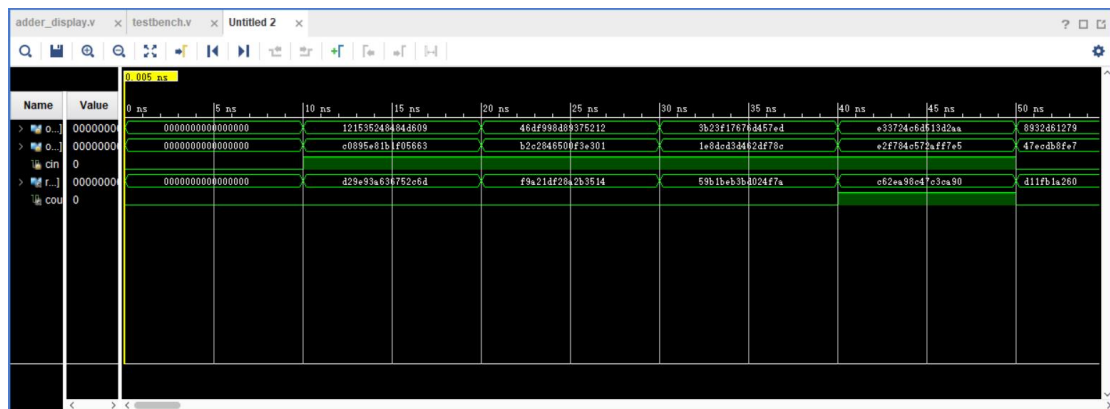
set_property IOSTANDARD LVCMOS33 [get_ports clk]
set_property IOSTANDARD LVCMOS33 [get_ports led_cout]
set_property IOSTANDARD LVCMOS33 [get_ports resetn]
set_property IOSTANDARD LVCMOS33 [get_ports input_sel]
set_property IOSTANDARD LVCMOS33 [get_ports input_pos]
set_property IOSTANDARD LVCMOS33 [get_ports sw_cin]

```

.....

5、实验结果分析

(1) 仿真结果



仿真结果总览。可以看到前两行为两个 64 位的操作数输入 operand1 和 operand2，第三行为 1 位二进制数末位进位输入 cin，第四行为一个 64 位操作数输出 result，第五行位一个 1 位二进制向上进位输出 cout。

可以看到，在前 10ns 中，五个操作数的结果都为 0，符合初始设置。自此之后，每过 10ns，前三行的输入进行一次随机数的生成，并同时计算出后两行的输出。总体上来看，结果基本符合预期结果。



切换为二进制形式查看，任取两例较为有代表性的模拟进行分析：首先，在 300-310ns 中 operand1 的结果为：

010010011100011001011101100100111010110010110111100101001011001

，operand2 的结果为：

1000001000111111001011000000010001101101110010110110100111011011
， cin 为 1。

从而，result 结果为：

1100110000000101100010011001100000011010100000110011010000110101

， cout 为 0。

符合 $\text{operand1} + \text{operand2} + \text{cin} = \{\text{cout}, \text{result}\}$ 。

为验证 cout 的正常运行，另取 310-320ns 中结果，operand1 的结果为：

0110110010110000101101111101100110111011010001011110001001110110

， operand2 的结果为：

1011011010100100001001100110110101100101001110110100100111001010

， cin 为 0。

从而，result 结果为：

0010001101010100110111100100011100100000100000010010110001000000

， cout 为 1。

同样符合 $\text{operand1} + \text{operand2} + \text{cin} = \{\text{cout}, \text{result}\}$ 。

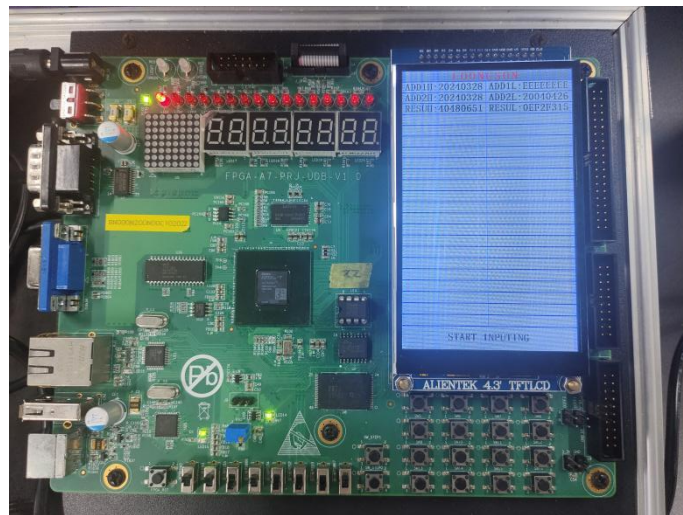
从而得出结论，即验证了正确性。

(2) 实验箱运行结果

输入输出同上，operand1 为 ADDIH（高 32 位），ADD1L（低 32 位），operand2 为 ADD2H（高 32 位），ADD2L（低 32 位），result 为 RESUH（高 32 位），RESUL（低 32 位），cin 为第三个开关，cout 为第一个 LED 灯。

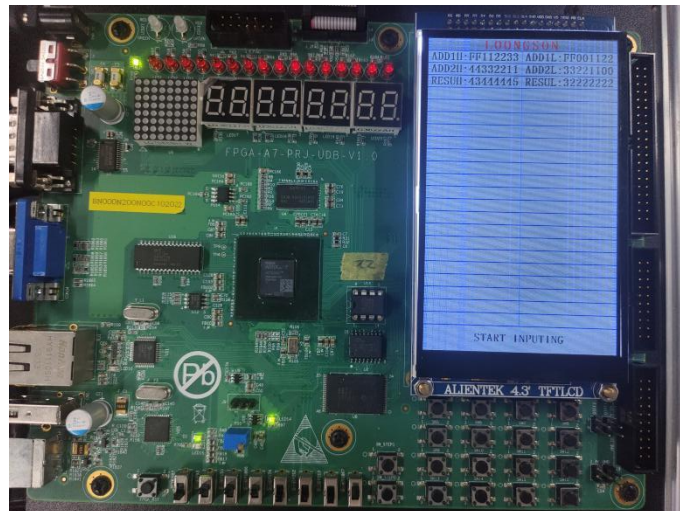
测试一：如下图所示，设置 operand1 的值为 20240328EEEEEEEE，operand2 的值为 2024032820040426，cin 为 1。

此时显示输出结果为 404806510EF2F315，cout 为 0，符合 $\text{operand1} + \text{operand2} + \text{cin} = \{\text{cout}, \text{result}\}$ ，验证了正确性。



测试二：如下图所示，设置 operand1 的值为 FF112233FF001122，operand2 的值为 4433221133221100，cin 为 0。

此时显示输出结果为 4344444532222222，cout 为 1，符合 $\text{operand1} + \text{operand2} + \text{cin} = \{\text{cout}, \text{result}\}$ ，验证了正确性。



6、总结感想

在本次实验中，我初次接触计算机组成原理实验，熟悉了 **vivado** 的操作流程和文件结构，通过用模拟数据 **simulation** 仿真和在试验箱中实际操作，验证设计正确性的方法。

同时，第一次接触 **verilog** 语言，学习了很多 **verilog** 语言的架构和语法，并最终将其应用在加法器的改进实验中，通过多文件的修改最终达到将 32 位加法器改装为 64 位加法器的效果。

接触了试验箱的结构和约束文件的使用，对硬件体系结构有了更深的认识。

这次实验也为后续实验的操作流程和期末的最终实验打下了基础。