组成原理课程第一次实验报告

实验名称：加法器扩展

学号：2310764 姓名：王亦辉 班次：计科一班

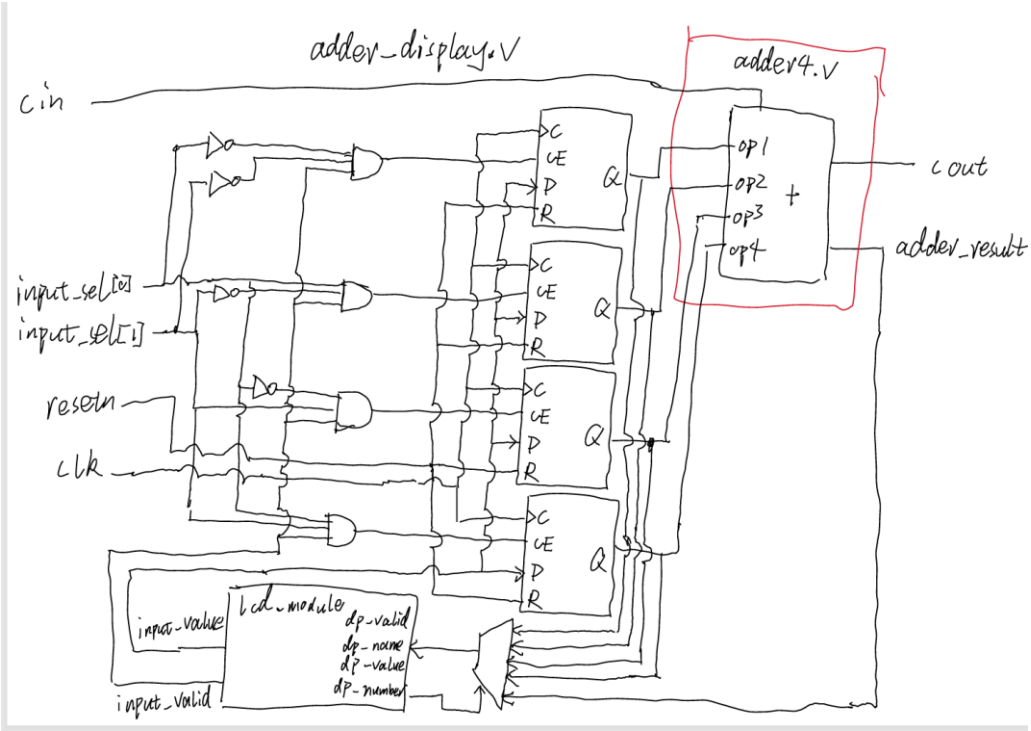
一、实验目的

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

实现四个数相加的加法器，进行相应模块的调整，然后将文件烧录到 FPGA 板上观察结果。

三、实验原理图



主要需要实现对四个数进行加法的模块，adder4.v。然后输入部分等需要修改，以输入四个数，以及在显示部分显示四个输入。

四、实验步骤

a) adder4.v 模块。

```
adder adder_module(
    .operand1(operand1),
    .operand2(operand2),
    .cin     (adder_cin1     ),
    .result  (adder_result1  ),
    .cout    (adder_cout1    )
);

adder adder_module2(
    .operand1(operand3),
    .operand2(operand4),
    .cin     (adder_cin2     ),
    .result  (adder_result2  ),
    .cout    (adder_cout2    )
);

adder adder_module3(
    .operand1(adder_result1),
    .operand2(adder_result2),
    .cin     (adder_cin3     ),
    .result  (adder_result3  ),
    .cout    (adder_cout3    )
);
```

```
assign adder_cin1 = cin;
assign adder_cin2 = cin[1];
assign adder_cin3 = & cin;
```

```
assign cout = adder_cout1 + adder_cout2 + adder_cout3;
```

i. 调用三次 adder 模块，完成加法，得到 adder_result3 和三个 adder_cout

ii. 由于进位输入变成两位，需要拆解 cin 分配给三个 adder_cin。

iii. 最后输出进位是三个 adder 的输出进位之和，都是第 33 的进位。

b) adder_display.v 模块

```
//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:输入为加数2(add_operand2)
input[1:0] sw_cin, // 00:0  ;  01:1 ; 10:2 ; 11:3

//led灯，用于显示cout
output[1:0] led_cout,
```

```
//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[1] && !input_sel[0])
    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[1] && input_sel[0])
    begin
        adder_operand4 <= input_value;
    end
end
```

```
end
6'd3 :
begin
    display_valid <= 1'b1;
    display_name  <= "ADD_3";
    display_value <= adder_operand3;
end
6'd4 :
begin
    display_valid <= 1'b1;
    display_name  <= "ADD_4";
    display_value <= adder_operand4;
end
```

i. 修改一些变量的位宽。

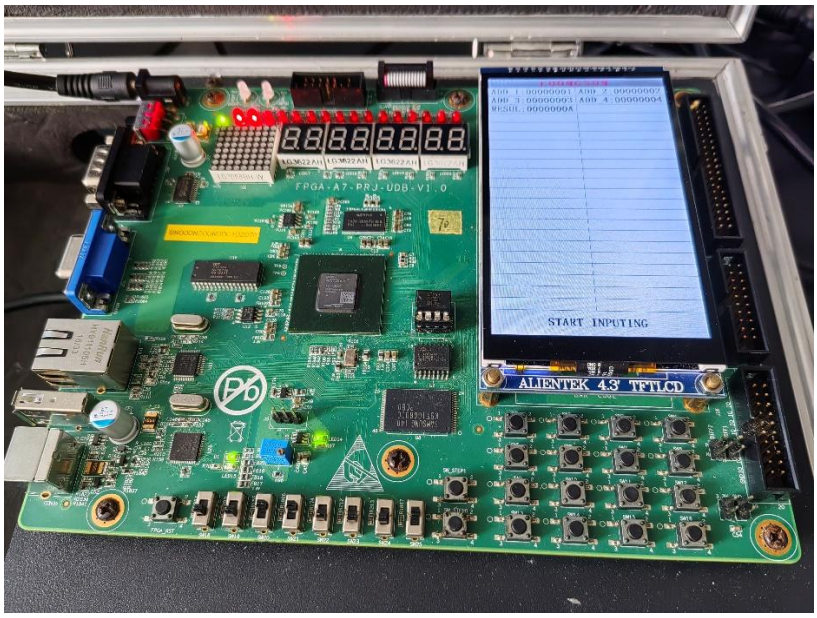
ii. 增加两个输入，并且根据 input_sel 修改选择逻辑。

iii. 显示模块中加入两个情况以显示新增的两个加数。

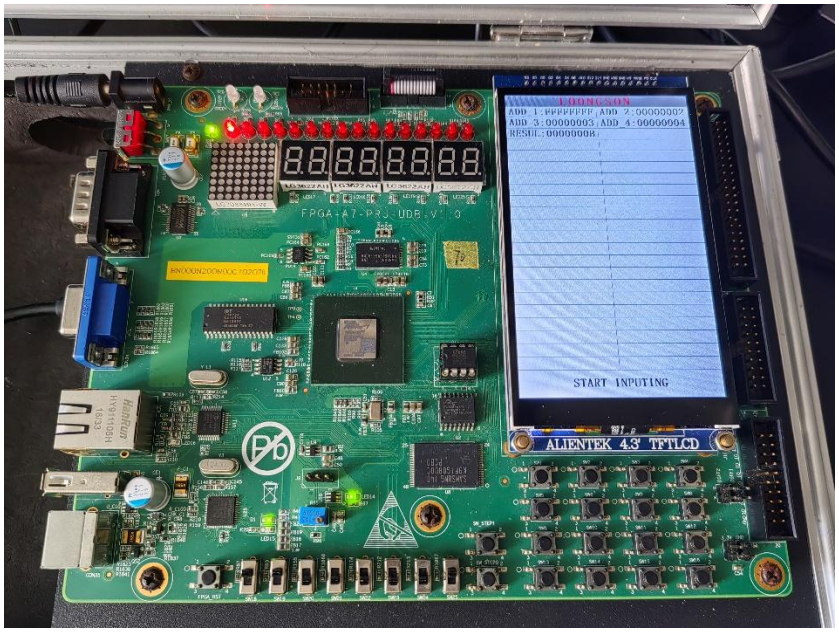
五、实验结果分析

拨码开关，从左到右，第一至第四个，分别是，input_sel[1], input_sel[0], cin[1], cin[0]。LED 灯，从左往右，第一至第四个，分别是，cout[1], cout[0]。

下面对溢出位和 cin 的四种情况进行验证。



输入：1+2+3+4+ 0(cin)
输出：溢出位为 00；resul 为 A
结果：1+2+3+4 = 10
正确。



输入：FFFFFFFF + 2+3+4+ 3(cin)
输出：溢出位为 01；RESUL 为 B。
结果：FFFFFFFF + 2 + 3 + 4 + 3 = FFFFFFFF + 1 + 10 = 01_00000000 + B
正确。

输入：FFFFFFFF + FFFFFFFF +3+4+ 2(cin)

输出：溢出位为 10；RESUL 为 7。

结果：FFFFFFFF + FFFFFFFF + 3 + 4 + 2 = FFFFFFFF + 1 + FFFFFFFF + 1 +7 = 10_00000000 + 7

正确。

输入：FFFFFFFF + FFFFFFFF + FFFFFFFF +4+ 1(cin)

输出：溢出位为 11；RESUL 为 2。

结果：FFFFFFFF + FFFFFFFF + FFFFFFFF + 4 + 1 = FFFFFFFF + 1 + FFFFFFFF + 1 + FFFFFFFF + 1 + 2 = 11_00000000 + 2。

正确。

六、总结感想

模块化让我们不用考虑触摸板显示部分是如何实现的，只需要专注修改 adder 以实现相应功能。