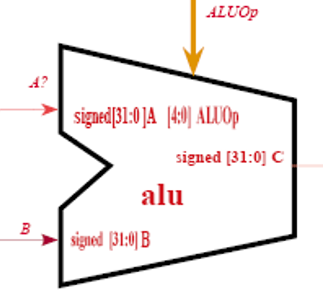
# 第五讲 ALU模块设计

原理概念：ALU是cpu的核心部件,它完成一系列算术运算和逻辑运算电路，。

原理框图:

：

实验目的：

1. 根据ALU原理框图，使用verilog语言设计并实现一个32位的ALU模块，运算指令码长度[4:0]；
2. 一个32位数输入到A口，或由sw10-sw7输入4位数到A口；
3. 一个32位数输入到B口，或由SW6-SW3输入4位数到B口
4. 运算器暂只支持”加、减 “2种操作，输入控制信号ALUOp（[4:0]）；
5. 结果输出到C[31:0]；
6. 由sw[12]控制数码管显示A，B，C，Zero四个数。

## 一、ALU模块接口定义

**Module alu**(

**input** signed [31:0] A, B, //alu input num

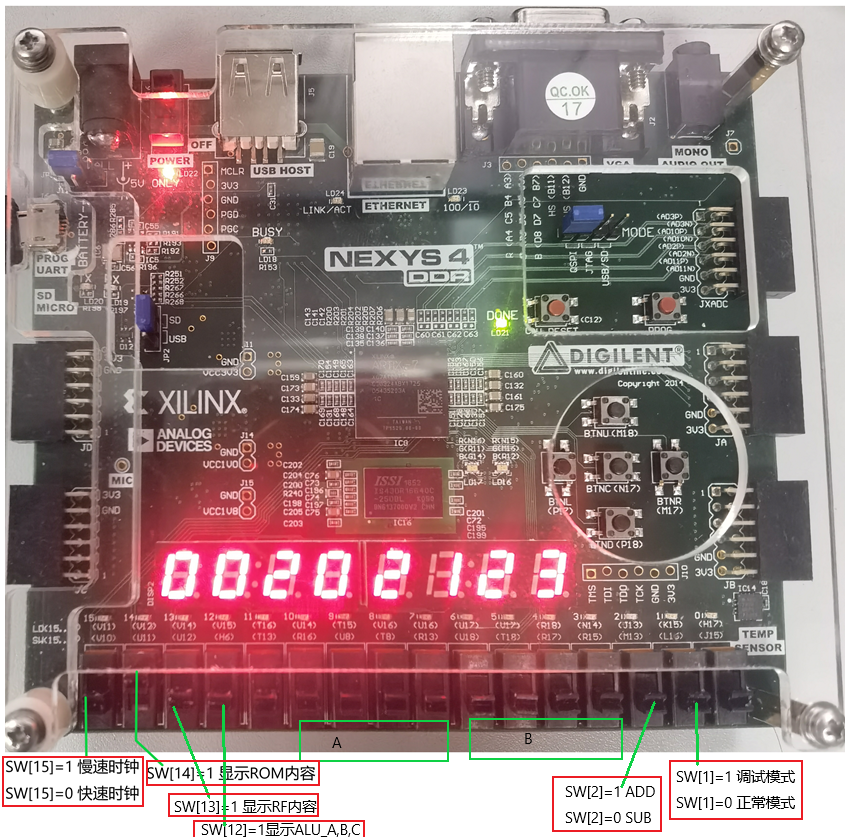
**input** [4:0] ALUOp, //alu how to do   
**output** signed [31:0] C, // alu result

**output reg** Zero

);

**Endmodule**

## 二、NEXYS4主板开关定义



说明：SW[12]=1 ：循环显示ALU中的A，B，C，Zero的内容。

SW[2]= 1 ：C=A+B

SW[2]= 0 ：C=A-B

**实验任务一**

1) sw[10]--sw[7] -----> **A口**

2) sw[6]--sw[3] -----> **B口**

3) sw[2] =1 **c口**=A+B ;

sw[2] =0 **c口**=A-B ;

循环输出 A、B、C、Zero的值，结尾加FFFFFFFF，测试是否满足运算要求。

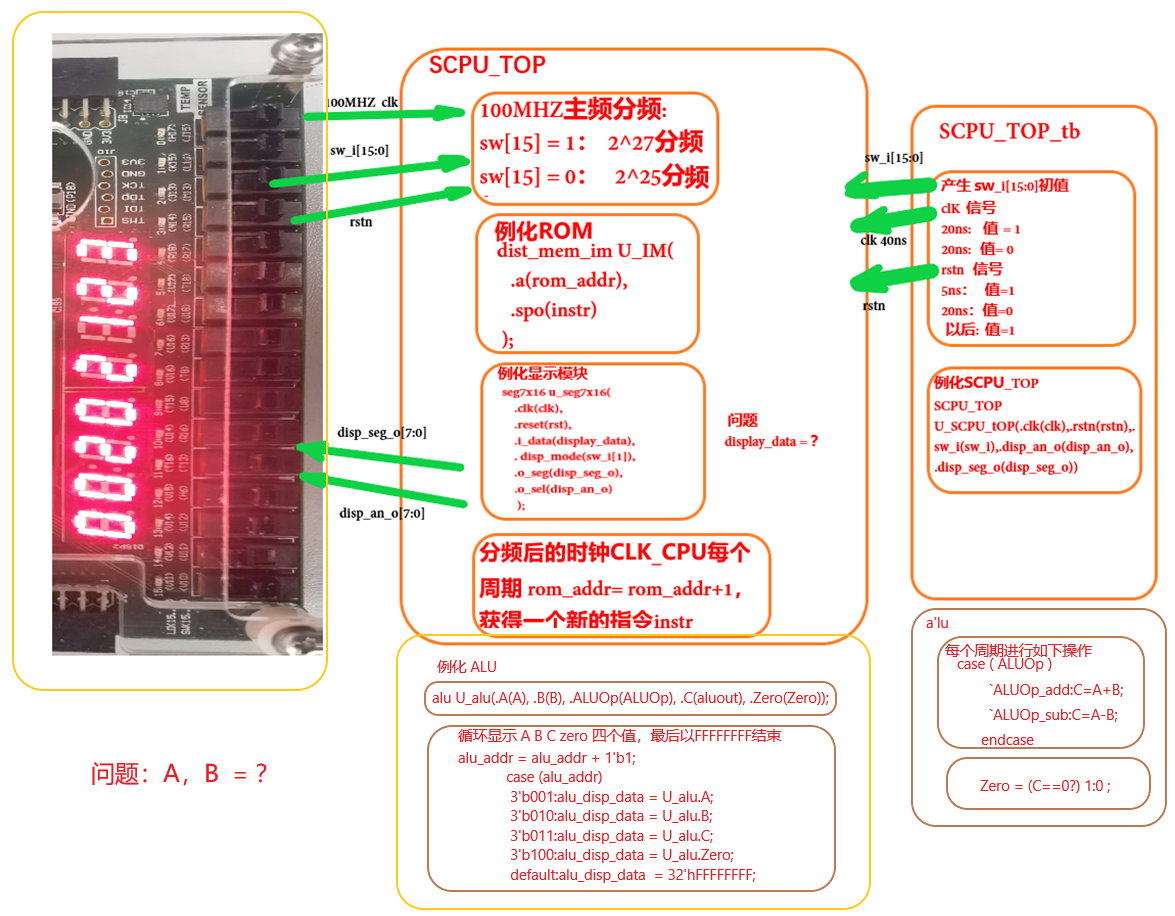
注明：[4:0]ALUOp 可以如下类似定义

// ALU control code

`define ALUOp\_add 5'b00001

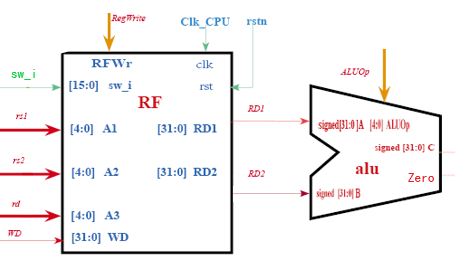
`define ALUOp\_sub 5'b00010

## 三、主模块、仿真模块和ALU模块功能定义



**实验任务二：**

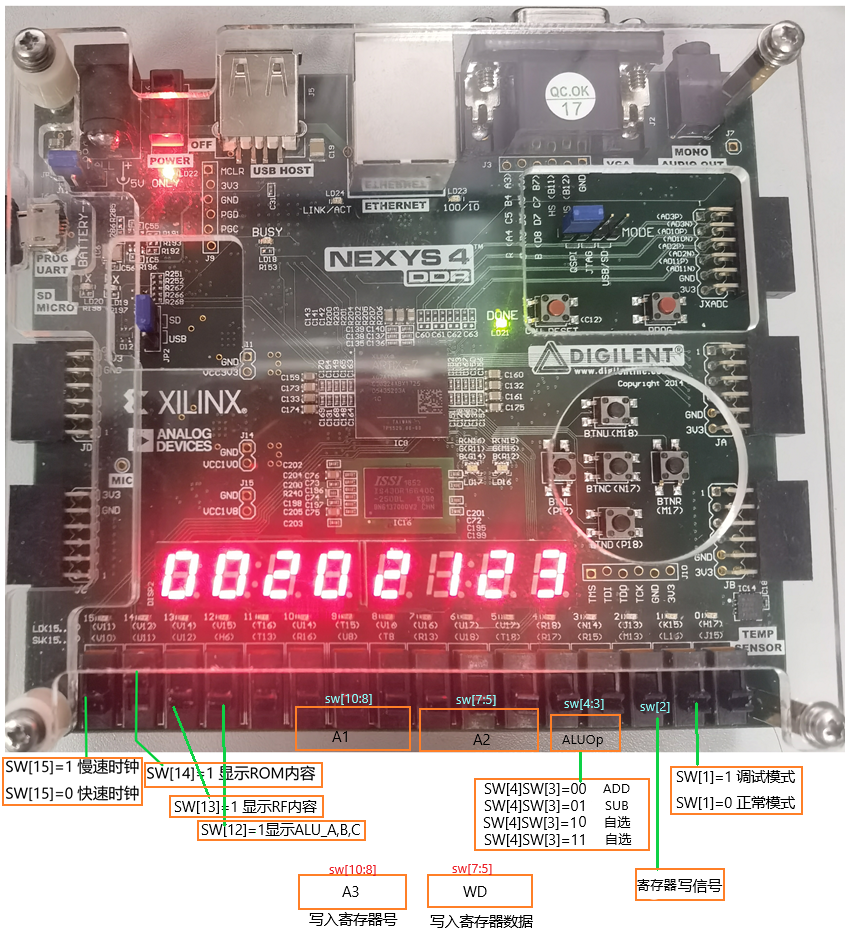
1）实现RF部件和ALU部件的关联，使得指定寄存器的数值送入ALU部件参与运算，具体参见下图：



2）通过**SW\_i**输入双端口要读出的寄存器编号**A1，A2**，并将读出的数值**RD1，RD2**送入ALU的**A，B**输入端，根据设定的**ALUOp**，计算结果，并显示。

3）通过**SW\_i**输入要写入寄存器编号和数值A3，WD，再通过步骤2），将计算结果显示出来。

4）NEXYS4主板开关定义如下：

****

NEXYS4主板开关定义说明：

a) sw[12]=1，循环显示A，B，C，Zero四个数值。

b) sw[10]--sw[8] -----> A1，A3 , 用来取代rs1，rd (缩减为3位)

c) sw[7]--sw[5] -----> A2，WD ，用来取代rs2，wd。(缩减为3位)

d) sw[4]--sw[3] -----> ALUOp。

e) sw[2]=0 , 写寄存器使能信号无效，不能修改寄存器，只能读寄存器 ，

sw[10:8] 代表A1寄存器号，

sw[7:5] 代表A2寄存器号。

f) sw[2]=1 , 写寄存器使能信号有效，可以修改寄存器 ，

sw[10:8] 代表A3写入寄存器号，

sw[7:5] 代表写入寄存器数值。

sw[1]=0, 非调试模式，可以修改对应寄存器数值

sw[1] =1, 调试模式，不能修改对应寄存器数值

注意：1）通过sw开关输入的A，B，WD运算数，它们是有符号数，位宽不到32位，有的输入的是3位或4位，那么它们的符号位如何处理？

1. RF寄存器组件的输入端口A1，A2，A3 的开关输入，需要扩展吗？

