

SHANGHAI JIAO TONG UNIVERSITY



计算机系统结构实验报告

实验名称: 简单的类MIPS单周期处理器部件实现-控制器，ALU

姓 名： 洪瑄锐

学 号： 517030910227

班 级： F1703302

手 机： 15248246044

邮 箱： [1204378645@qq.com](mailto:1204378645@qq.com)

目录

[1. 实验目的 2](#_Toc8782566)

[2. 实验原理与代码实现 2](#_Toc8782567)

[2.1 主控制器Ctr的实现 2](#_Toc8782568)

[2.1.1 主要思想 2](#_Toc8782569)

[2.1.2 Ctr模块代码 4](#_Toc8782570)

[2.1.3 仿真测试 5](#_Toc8782571)

[2.2运算单元控制器ALUCtr的实现 7](#_Toc8782572)

[2.2.1主要思想 7](#_Toc8782573)

[2.2.2 ALUCtr模块代码 8](#_Toc8782574)

[2.2.3 仿真测试 8](#_Toc8782575)

[2.3 ALU的实现 9](#_Toc8782576)

[2.3.1 主要思想 9](#_Toc8782577)

[2.3.3 仿真测试 11](#_Toc8782578)

[3. 感想 12](#_Toc8782579)

[4. 参考文献 12](#_Toc8782580)

# 实验目的

1. 理解CPU控制器，ALU的原理；
2. 主控制器Ctr的实现；
3. 运算单元控制器ALUCtr的实现；
4. ALU的实现；
5. 使用功能仿真。

# 2. 实验原理与代码实现

## 2.1 主控制器Ctr的实现

### 2.1.1 主要思想

主控制单元（Ctr）的输入为指令的opcode字段，操作码经Ctr的译码，给ALUCtr、Data Memory、Registers、Muxs等部件输出正确的控制信号。

**各控制信号所代表的含义：**

1. **RegDst：**

R型指令和I型指令目标寄存器在指令编码中的位置不同，前者为Inst[15-11]，后者为Inst[20-16]，RegDst信号用于根据指令类型选择目标寄存器。

1. **Branch：**

用于判断是否为跳转指令，在beq指令中如果两个寄存器中的值相等则跳转到与当前指令的下一条指令距离为指定立即数的指令，Branch信号与zero信号相与，如果结果为1则进行pc=pc+4+(immediate<<2)运算，否则pc=pc+4。

1. **MemRead：**

判断当前指令是否需要读内存。

1. **MemtoReg：**

判断当前指令是否需要将内存中的数据读入寄存器，一般用于lw指令。

1. **ALUOp：**

根据指令类型判断算术逻辑运算单元ALU需要执行什么运算，如add、sub、or等等。

1. **Memwrite：**

判断当前指令是否需要写内存，一般用于sw指令。

1. **ALUSrc：**

算术逻辑运算单元ALU的操作数可能来自于寄存器，也可能是立即数，根据指令类型确定操作数来源。

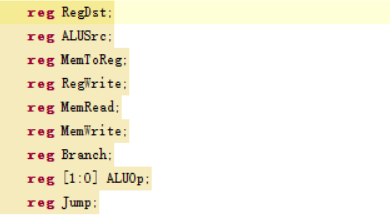
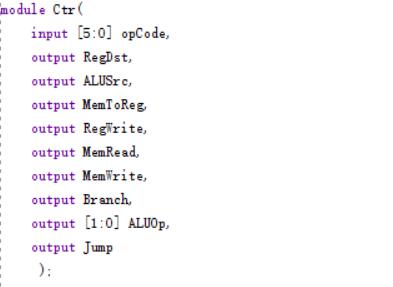
1. **RegWrite：**

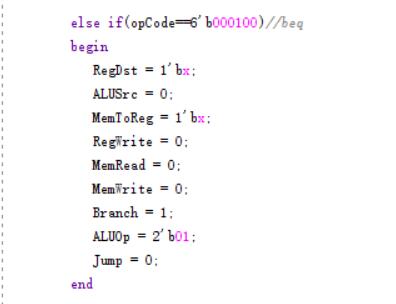
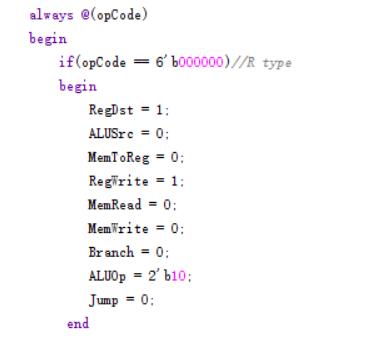
判断当前指令是否需要写寄存器。

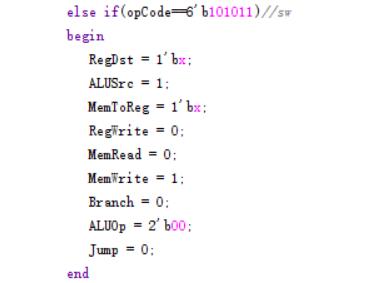
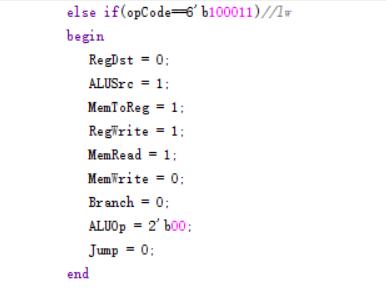
因此，根据实验指导书所给真值表编写代码即可。

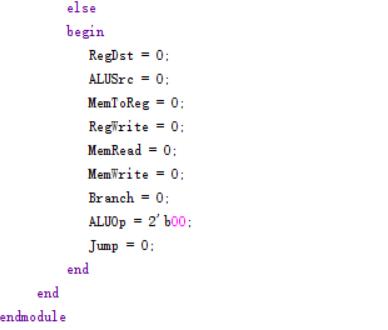
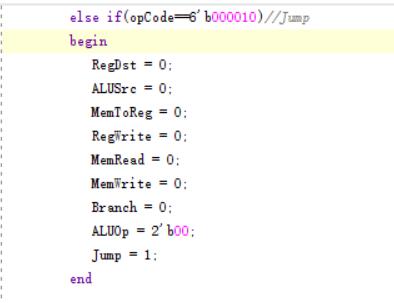


### 2.1.2 Ctr模块代码









采用if else语句实现真值表。

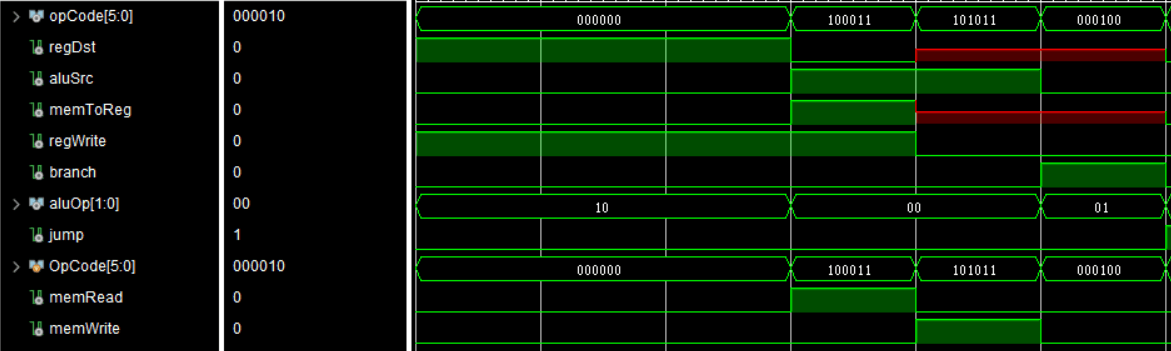
### 2.1.3 仿真测试

通过给opcode赋值观察各信号是否为对应的值来判断试验是否成功。

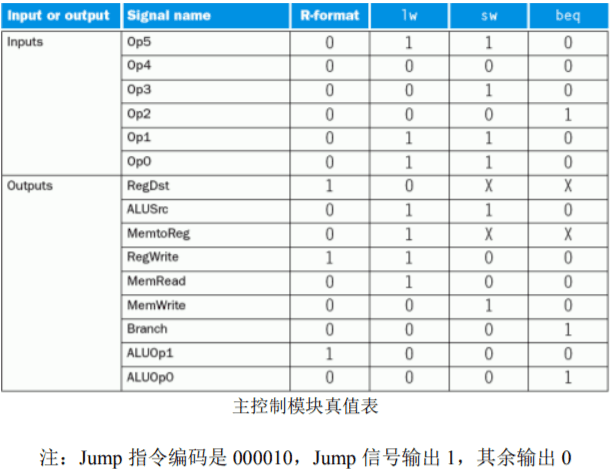
**部分仿真代码如下所示：**



**仿真结果：**



**真值表：**



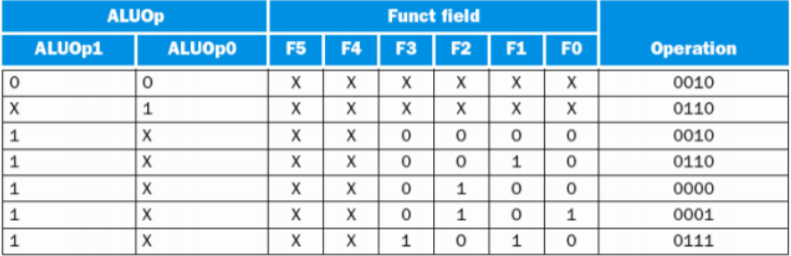
经过与真值表对比，仿真结果正确。

## 2.2运算单元控制器ALUCtr的实现

### 2.2.1主要思想

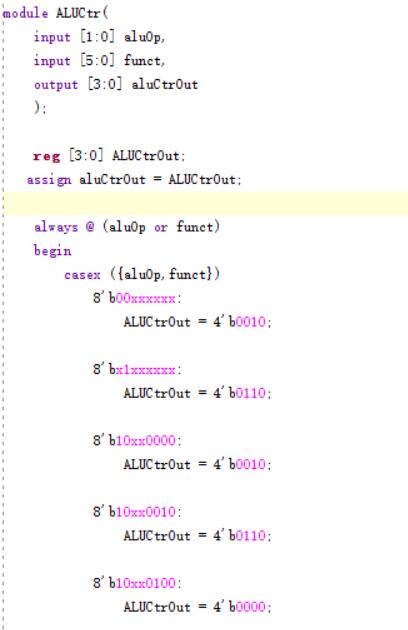
ALU的控制单元模块（ALUCtr）是根据主控制器的ALUOp来判断指令类型。根据指令的后6位即功能码区分R型指令，如add、sub、or等，综合操作码opcode和功能码funct控制ALU做正确的操作。因此，要想实现ALUCtr只要综合aluop和funct输出aluCtrOut即可。

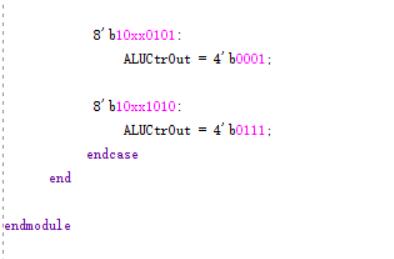
实验指导书所示真值表：



### 2.2.2 ALUCtr模块代码

根据实验指导书中真值表得：

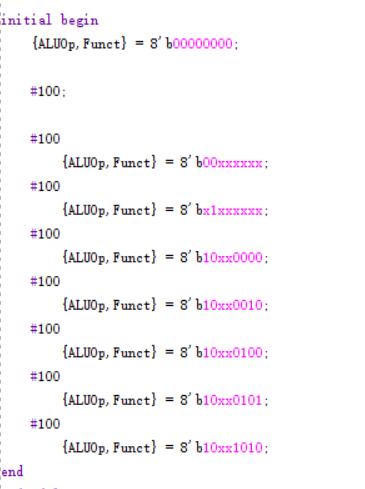




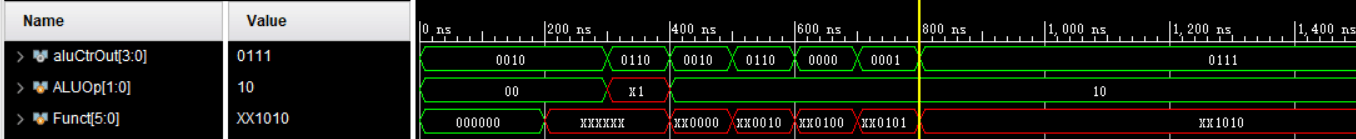
### 2.2.3 仿真测试

通过给ALUOp和Funct赋值检查aluCtrOut的值是否与真值表相符即可。

**部分代码：**



**仿真结果：**

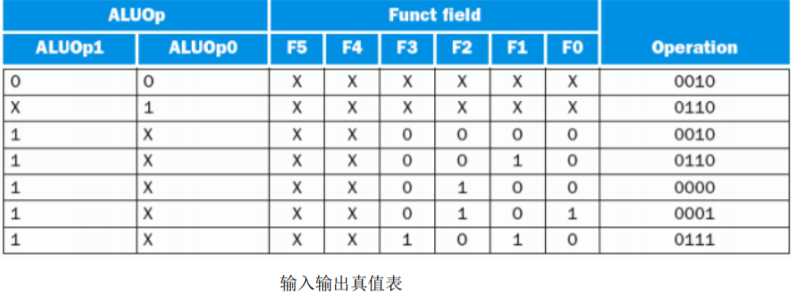


与真值表对应后可知仿真结果正确。

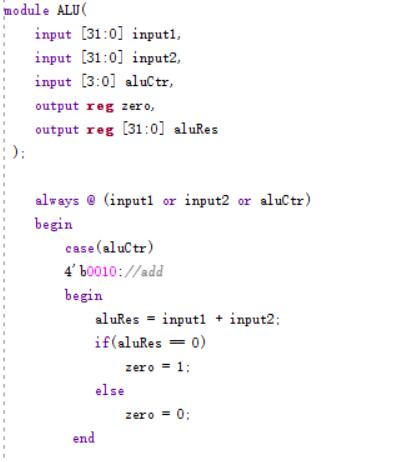
# 2.3 ALU的实现

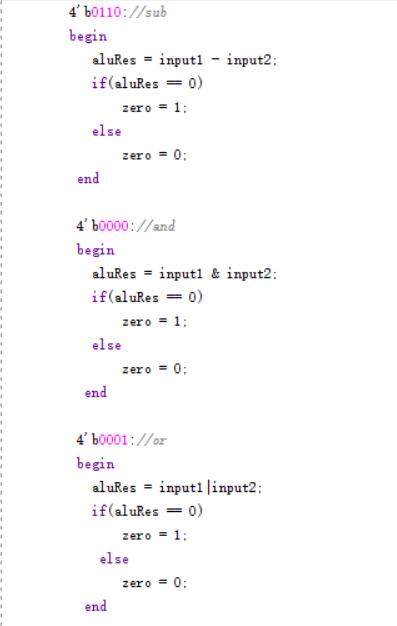
### 2.3.1 主要思想

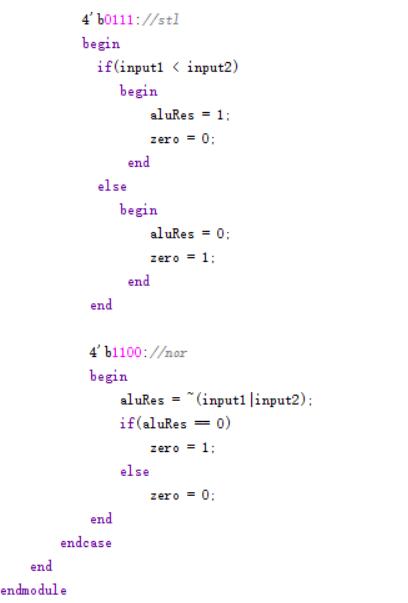
根据ALUCtr，ALU对两个输入执行对应的操作，ALURes为输出结果。若减法操作ALURes的结果为0时，则Zero的输出置为1。编写模块代码时，输入为ALUCtr，两个操作数，输出为ALURes，Zero，根据ALUCtr判断对两个操作数执行何种运算。



**2.3.2 ALU模块代码**



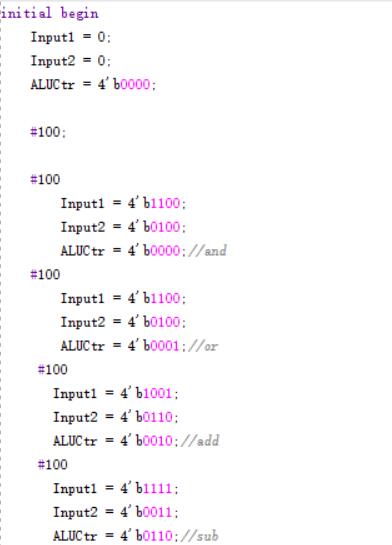


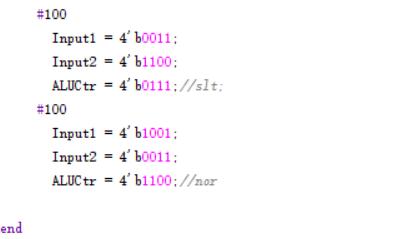


### 2.3.3 仿真测试

通过对操作数input1、input2和aluCtr赋值得到正确的运算结果aluRes和zero

**部分代码：**

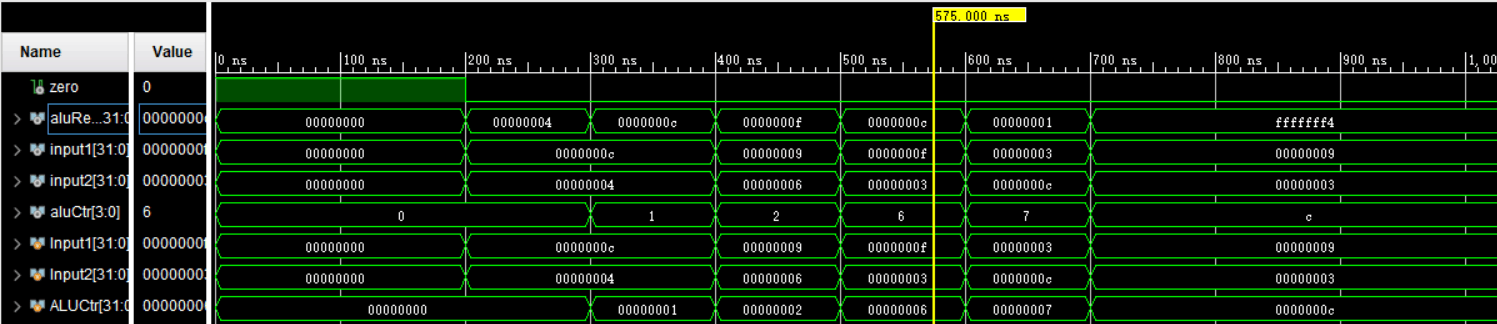




**仿真结果：**

1100b&0100b=0100b=**4h**；1100b｜0100b=1100b=**ch**；1001b+0110b=1111b=**fh**；

1111b-0011b=1100b=**ch**；slt 0011b 0110b=**1h**；nor 00000009b 00000003b=0101b=**fffffff4h**



分析可知仿真结果正确。

# 3. 感想

由于lab1和lab2实验指导书给出了完整的代码，所以lab3才是真正意义上的第一次实验。通过这次实验我有几点收获：

1. 我学习了verilog语言的基本语法，如module、always、ifelse、case等，还明白了模块与激励之间的关系，懂得了如何实例化，wire与reg的使用等。
2. 加深了对简单cpu中Ctr模块，ALUCtr模块，ALU模块的理解，了解了其内部实现。

感谢老师和同学们在本次实验中对我的教导和帮助。

# 4. 参考文献

《cs145实验指导书lab03》