

# 计算机组成实验三

## 简单的类MIPS单周期处理器实验—控制器，ALU

### 5140219355

### 朱耀明

[计算机组成实验三 1](#_Toc451018558)

[简单的类MIPS单周期处理器实验—控制器，ALU 1](#_Toc451018559)

[5140219355 1](#_Toc451018560)

[朱耀明 1](#_Toc451018561)

[1． 总述 2](#_Toc451018562)

[1.1 实验名称 2](#_Toc451018563)

[1.2 实验目的 2](#_Toc451018564)

[1.3 实验范围 2](#_Toc451018565)

[2． 控制器、ALU实现 2](#_Toc451018566)

[2.1工程文件配置 2](#_Toc451018567)

[2.2 MIPS指令分析 3](#_Toc451018568)

[2.3主控制单元模块Ctr 3](#_Toc451018569)

[2.4 ALU单元模块AluCtr 6](#_Toc451018570)

[2.5 ALU运算器模块 7](#_Toc451018571)

[3．仿真 9](#_Toc451018572)

[3.1Ctr模块仿真 9](#_Toc451018573)

[3.2 AluCtr 模块仿真 10](#_Toc451018574)

[3.3 ALU模块仿真 11](#_Toc451018575)

[4.小结 13](#_Toc451018576)

[4.1心得 13](#_Toc451018577)

[4.2细节分析 13](#_Toc451018578)

# 1． 总述

## 1.1 实验名称

简单的类MIPS单周期处理器实验—控制器，ALU

## 1.2 实验目的

理解CPU控制器， ALU的原理

## 1.3 实验范围

本次实验将覆盖以下范围

ISE的使用

Spartan 3E实验板的使用

使用Verilog HDL进行逻辑设计

CPU控制器的实现

ALU的实现

# 2． 控制器、ALU实现

## 2.1工程文件配置

在ISE新建工程的界面，设置Project Name与Project Location后，Top-Level Source Type选择HDL. 在 Device Properties中各属性填写如下(如Fig.1)：

Product Category: ALL

Family: Spartan3E

Device: XC3S500E

Package: FG320

Speed: -4

Synthesis Tool: XST(VHDL/Verilog)

Simulator: ISim(VHDL/Verilog)，也可用Modelsim仿真

Preferred Language: Verilog

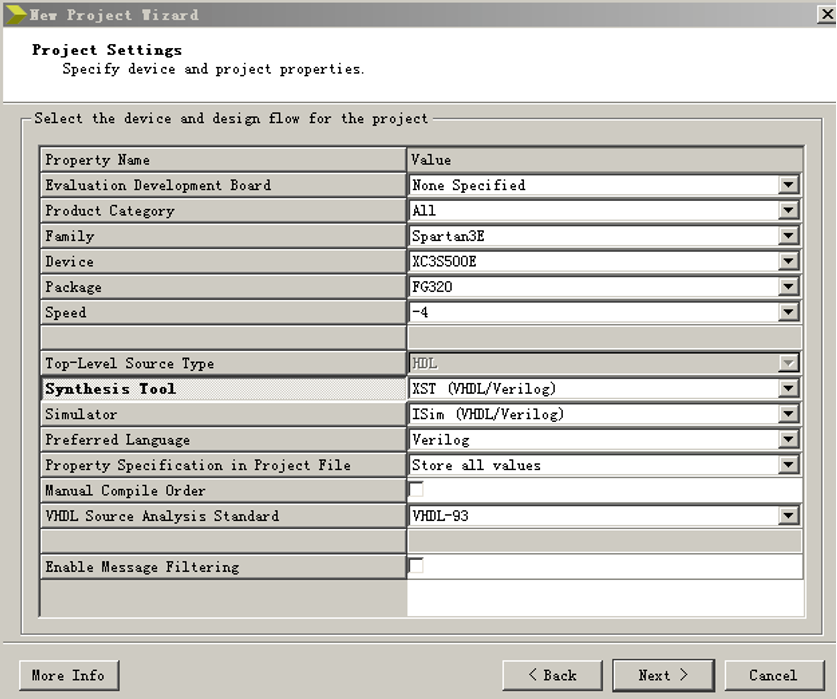


Figure 项目设置

## 2.2 MIPS指令分析

MIPS指令是典型的RISC型指令，所有指令长度为32位，分为R-type，I-type以及J-type三种类型(如Fig.2)。



Figure MIPS三种指令的格式

其中opcode域传入主控制单元模块用于产生控制信号，funct域传入ALU控制模块产生运算器控制信号，immediate是立即数，进行符号扩展为32位后作为运算数，address与PC中的地址计算出一个新指令地址使指令跳转。

本实验需要模拟的指令有：

R型：add, sub, and, or, slt

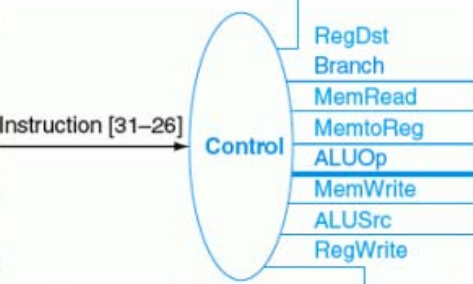
I型：lw, sw, beq

J型：jump

## 2.3主控制单元模块Ctr

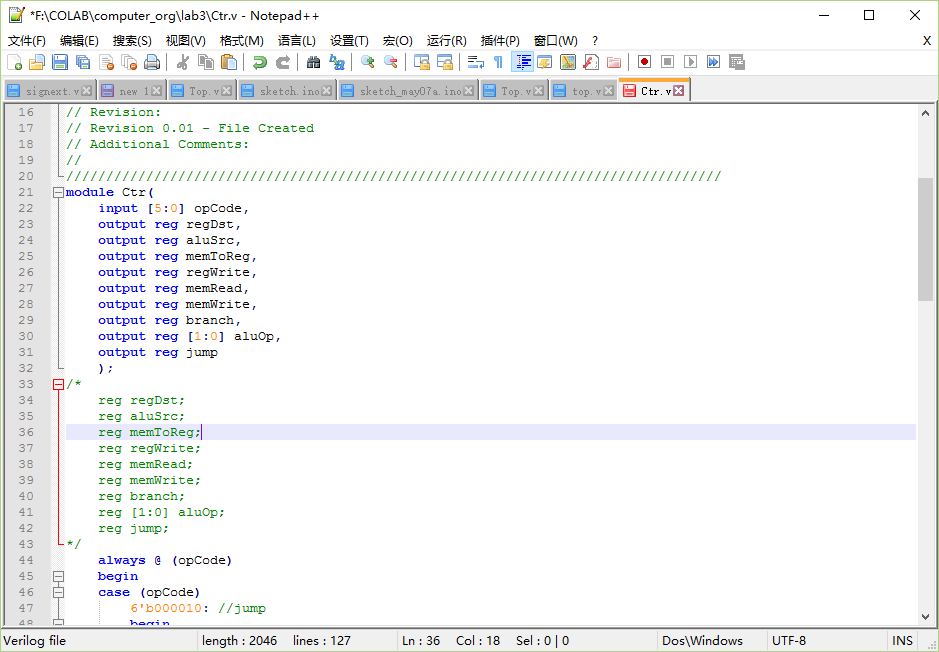
2.3.1 模块描述

主控制单元输入为指令的opCode字段，即操作码。操作码经过主控单元的译码，给ALUCtr，Data Memory，Register，Muxs等部件输出正确的控制信号(如Fig.3)。



Figure

模块输入输出信号定义见Fig.4：



Figure

2.3.2译码功能实现

信号的赋值遵循Fig.5 展示的规则(注：Jump 指令编码是 000010， Jump 信号输出 1，其余输出 0)



Figure

按照书上的指导，使用case语句来实现机器码到Ctr控制信号的译码操作。实现见Fig.6

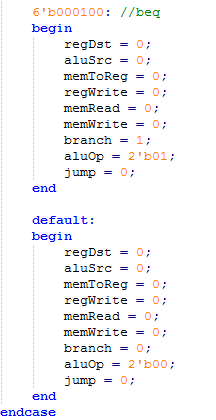
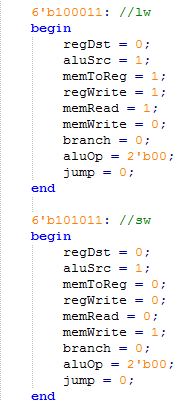
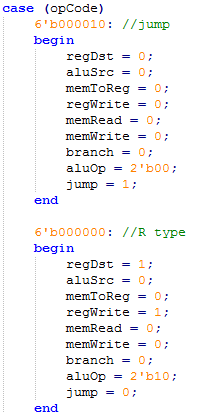


Figure 6 Ctr模块译码实现

## 2.4 ALU单元模块AluCtr

2.4.1模块描述

AluCtr控制模块通过之前Ctr模块得到的aluop[1:0]判断指令类型。对于R型指令，AluCtr通过对funct域进行译码(如图Fig.7，Fig.8)，得到对应的控制信号。



Figure 7 R型指令的funct域

模块输入输出信号定义见Fig.9

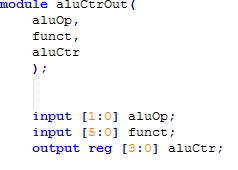
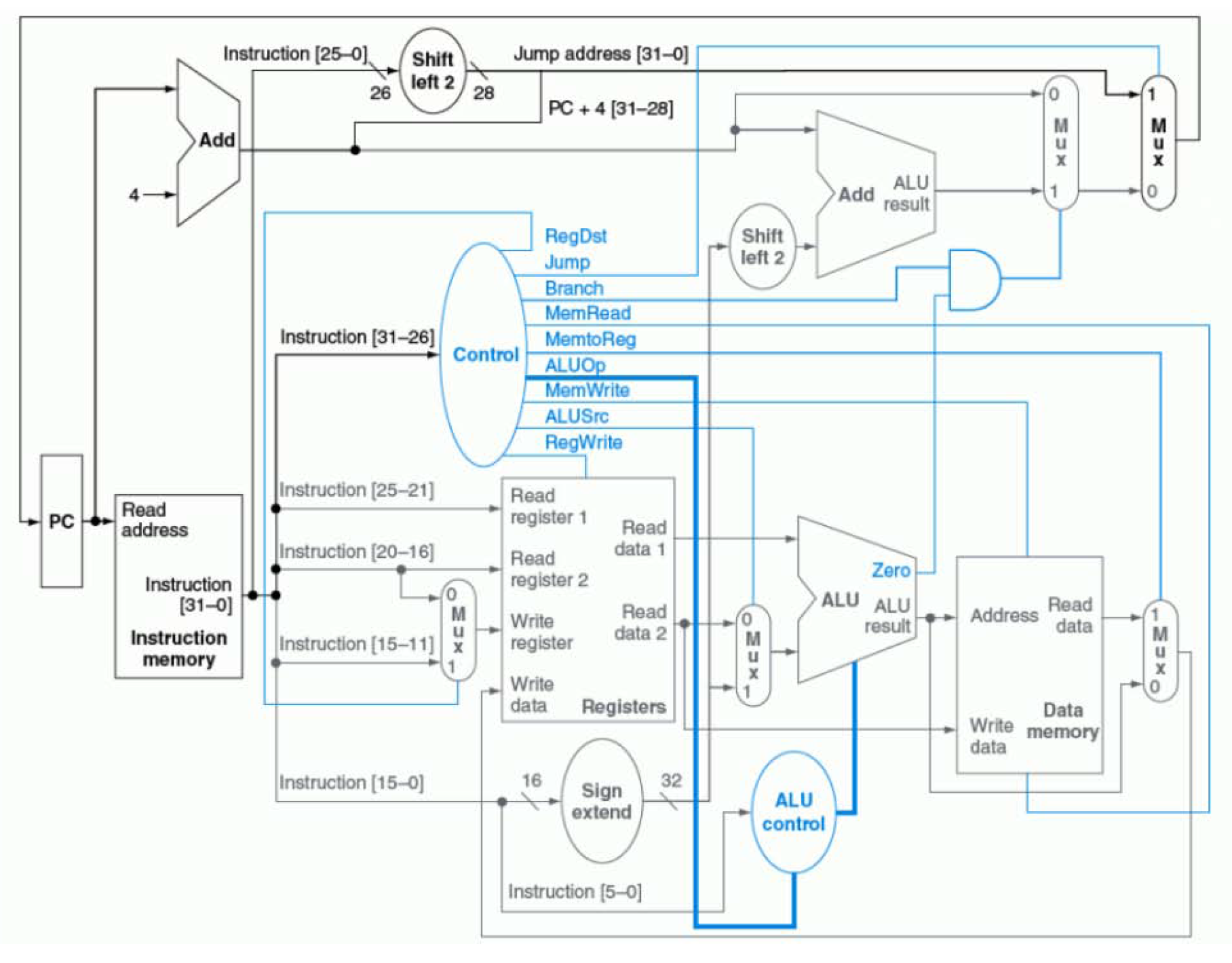


Figure 8 AluCtr Figure 9

2.4.2 译码功能实现

对于要实现的指令，AluOp[1:0],funct域与AluCtr输出的对应关系如下

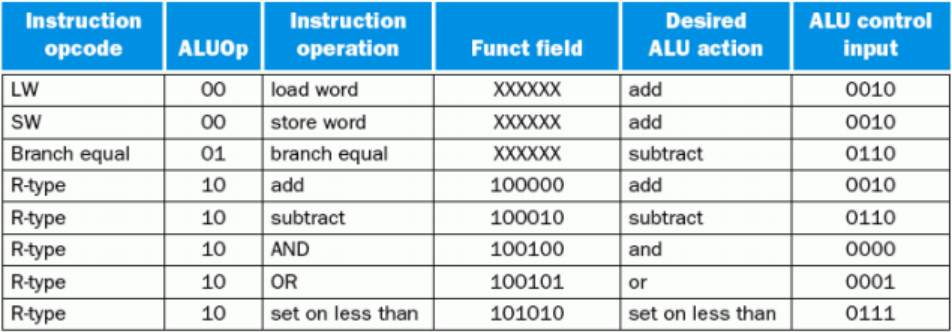


Figure 10 aluCtrOut 和 alu 操作的对应关系

与Ctr模块类似，可以使用casex语句实现AluCtr模块译码功能。

具体实现见Fig. 11

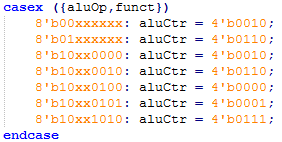


Figure 11

由于aluCtrOut 和 alu 操作的对应关系一图可知，该模块的输出与输入不是一一映射的。因此，在实现中引入了不定项x.对应地，在模块代码中使用了casex语句取代case语句。case语句是一种全等比较，而casex语句则会忽略比较位中出现的x或者z，只要其他位相匹配则判定相等。

## 2.5 ALU运算器模块

2.5.1模块描述

根据AluCtr模块输出的控制信号，运算器(ALU)会对传入的两个操作数进行运算，由aluRes输出运算结果。若执行减法操作的运算结果为零，则置zero输出信号为1.(如图Fig.12)

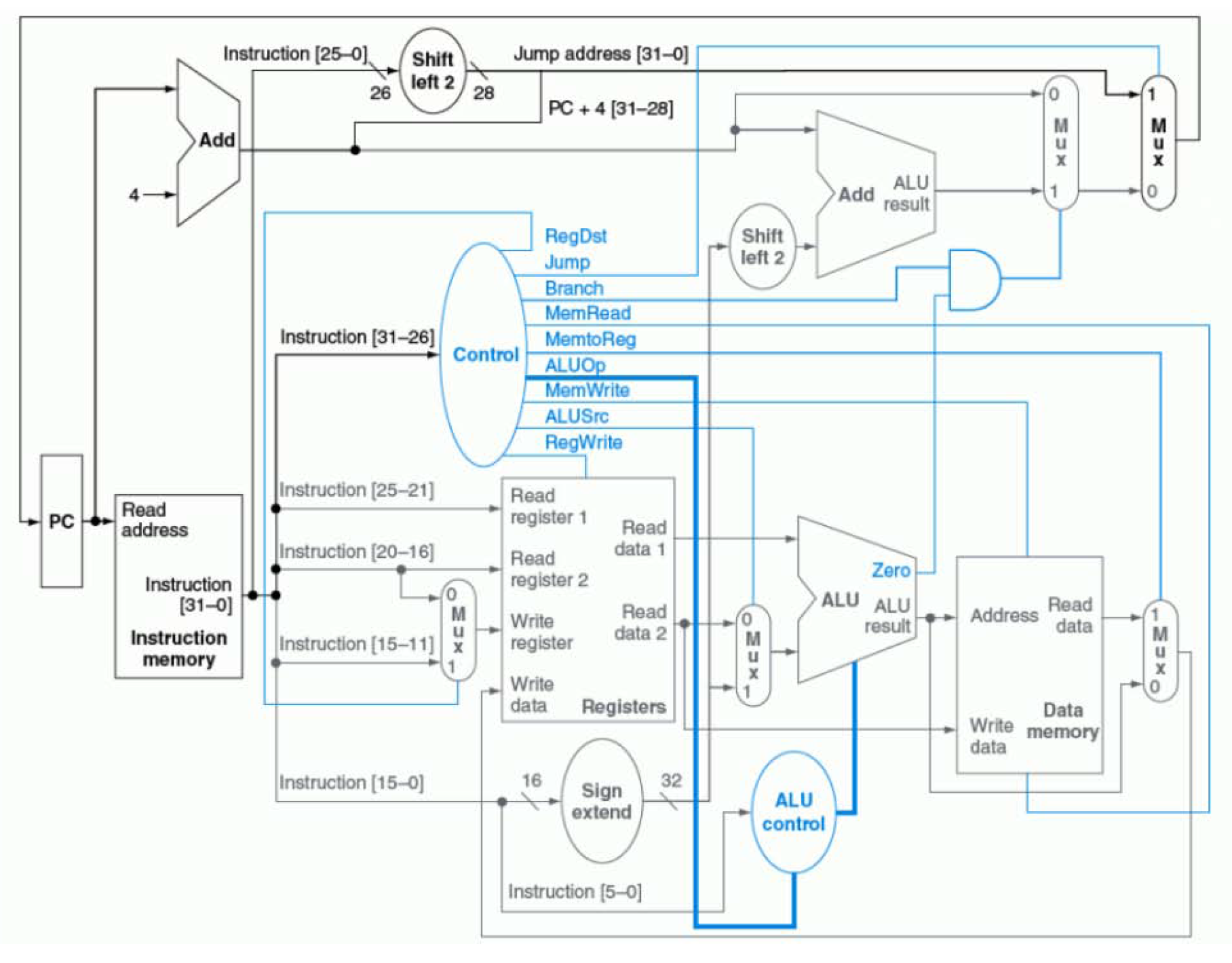


Figure 12

2.5.2模块实现

ALU模块的实现见Fig.13

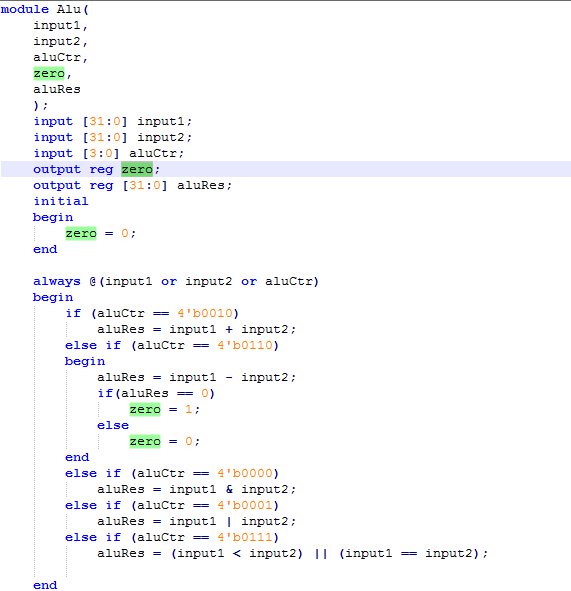


Figure 13

至此，控制器与ALU已经实现，之后需进行仿真对其是否能实现指令进行验证。

# 3．仿真

## 3.1Ctr模块仿真

右键选中 Hierarchy 窗口，点击 New Source，创建 Testbench 文件。如Fig.14

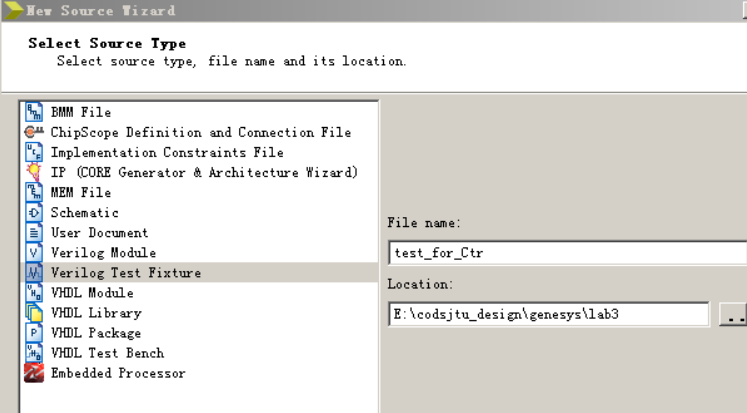


Figure 14

下一个窗口选择 Ctr 模块， 点击 Next。工具会自动为 Ctr 创建 Testbench 模板文件。

在文件中添加激励见Fig.15

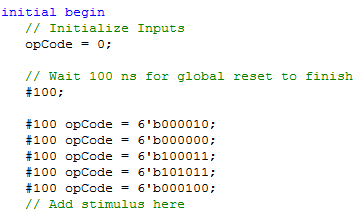


Figure 15

其中时间单位为ns

#100意味着等待100ns

之后，在View栏中选中Simulation，Hierarchy窗口中选择test\_for\_Ctr文件，在Processes窗口中选择Simulation Behavioral Model，双击开始仿真(见Fig.16)

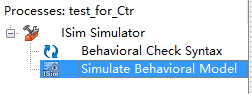


Figure 1

仿真波形图见Fig.17.由信号仿真波形间的联系，可见Ctl模块实现了我们需要的功能。

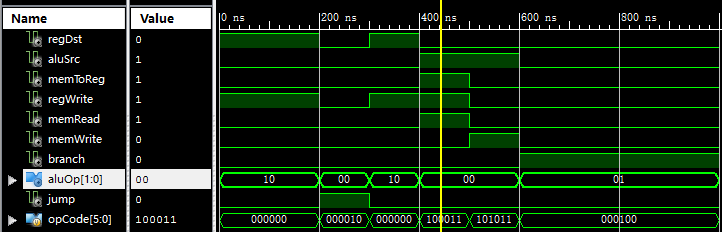


Figure 1

## 3.2 AluCtr 模块仿真

仿真文件的创建与基本设置与之前相似，此处不再赘述。

设置test\_for\_aluCtrOut文件后，添加激励如Fig.18

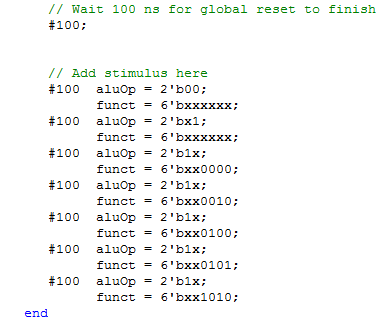


Figure 1

仿真波形如Fig.19所示

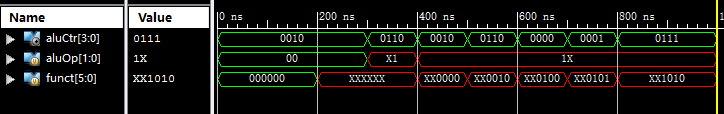
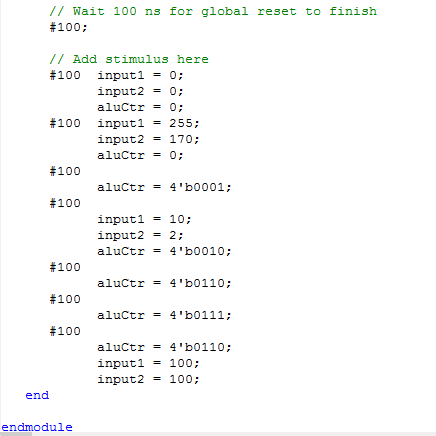


Figure 1

图中红色波形是因为在仿真代码中部分项使用了不定项x，由仿真波形，可验证AluCtr模块满足了设计要求。

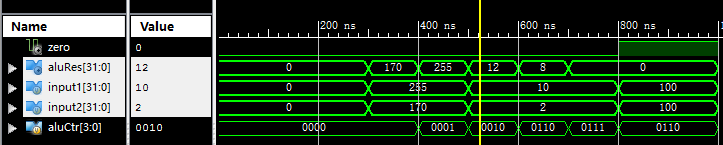
## 3.3 ALU模块仿真

与前面的仿真类似，设置test\_for\_alu文件。在ALU的仿真中，可以自行定义input的数值与运算。如图Fig.20



Figure

仿真波形见Fig.21（为了显示方便，input信号与aluRes信号以十进制无符号数形式显示）。



Figure

仿真波形显示ALU设计已实现。

至此，本实验的控制器、ALU设计和仿真都已完成。

# 4.小结

## 4.1心得

之前的lab1,lab2都是实验指导书给出了代码与设置步骤，进行按图索骥的ISE入门学习。lab3的实验指导书则留下了许多要我们自己去实现的地方。在《电子技术实验》这一门课程中，我们已经学习了一些verilog语言的基础语法，同时我借阅了图书馆的verilog相关书籍，从而完成了实验设计。

## 4.2细节分析

在设计AluCtr模块时，如果安装实验指导书上的输入输出真值表进行设计，可能出现同一条指令适用于多种casex情形的情况，而casex语句会自动匹配最先出现的适合情况，可能会导致一定错误。查询《计算机组成与设计》一书，我对实验指导书上的真值表进行了少量修改，将所有opCode为1x的情形设为了10情形，避免了冲突。

为了更好地区分代码中的各种部分(信号、常数、模块)，报告中的部分截图是在notepad++软件中打开.v代码文件进行截图的。