***2019***



**计算机组成原理 ·实验报告·**

j0242087[1]

|  |  |
| --- | --- |
| 专 业： | 计算机科学与技术 |
| 班 级： | CS1703 |
| 学 号： | U201714670 |
| 姓 名： | 范唯 |
| 电 话： | 13451158896 |
| 邮 件： | U201714670@hust.edu.cn |
| 完成日期： | 2019-12-12 |



目 录

**[1 CPU设计实验 2](#_Toc927169066)**

[1.1 设计要求 2](#_Toc789149630)

[1.2 方案设计 3](#_Toc378827538)

[1.3 实验步骤 9](#_Toc1812901458)

[1.4 故障与调试 13](#_Toc936820970)

[1.5 测试与分析 14](#_Toc1947426633)

**[2 总结与心得 17](#_Toc601156904)**

[2.1 实验总结 17](#_Toc1881010040)

[2.2 实验心得 17](#_Toc1028974793)

**[参考文献 18](#_Toc293536660)**

# CPU设计实验

## 设计要求

本实验需要利用实验包已经给出的组建和自己设计的组建构建一个32位MIPS CPU处理器.其中包含了单周期MIPS(硬布线)CPU、多周期MIPS(微程序)CPU、多周期MIPS(硬布线)CPU.

这三个CPU需要支持加、减、乘、除，逻辑与、或、非、异或运算、逻辑左移、逻辑右移，算术右移运算，支持常用程序状态标志（有符号溢出OF、无符号溢出CF，结果相等Equal).

核心指令集见下表.

**表1.1 核心指令集**

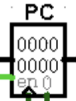
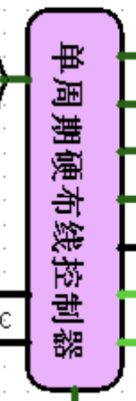
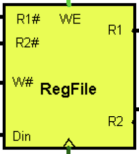
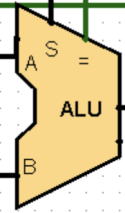


最终设计完成的 CPU 应能运行教师提供的标准 测试程序，程序存储在 Logisim ROM 模块中(指令存储器、数据存储器分开)。

## 方案设计

### 单周期MIPS(硬布线)CPU设计

\* 单周期MIPS(硬布线)CPU功能部件:

1. 指令计数器PC，用于存放当前指令执行位置的地址；
2. 指令存储器IM，存储CPU要执行的指令；
3. 数据存储器DM，
4. 立即数扩展器，用于将I型指令中的16位立即数扩展为32位；
5. 控制器，产生控制信号，控制指令执行的数据通路；
6. 寄存器，提供32个MIPS通用寄存器；
7. 算术逻辑单元ALU，产生运算结果。

\* 构建数据通路

绘制主要功能部件输入来源表，该表主要用于描述控制类信号，仅保留数据类信号.并且合并输入源。计算每一个数据项的输入种类，若有多个输入来源，则需要控制信号配合多路选择器进行选择。

\* 列出相关内容

如所有功能部件、多路选择器控制信号、运算操作选择的产生条件,利用译码电路生成各指令译码信号，然后以行为单位将各个产生信号的条件相加（逻辑或）即可得到控制信号的逻辑表达式。

>RegDst为1表示RegiFile写回地址由R型指令rd字段给出

>RegWrite为1打开RegiFile写使能

>AluSrcB为1表示ALU的第二个操作数将由立即数扩展器S-EXT给出

>bne为1表示指令为bne

>beq为1表示指令为beq，控制PC数据来；

>MemWrite为1打开数据存储器DM写使能

>MemToReg为1表示从数据存储器DM中选数据送入RegiFile

>Halt为停机信号，为1时系统停机

>加法信号

>比较信号

\* 构建单周期硬布线控制器电路：

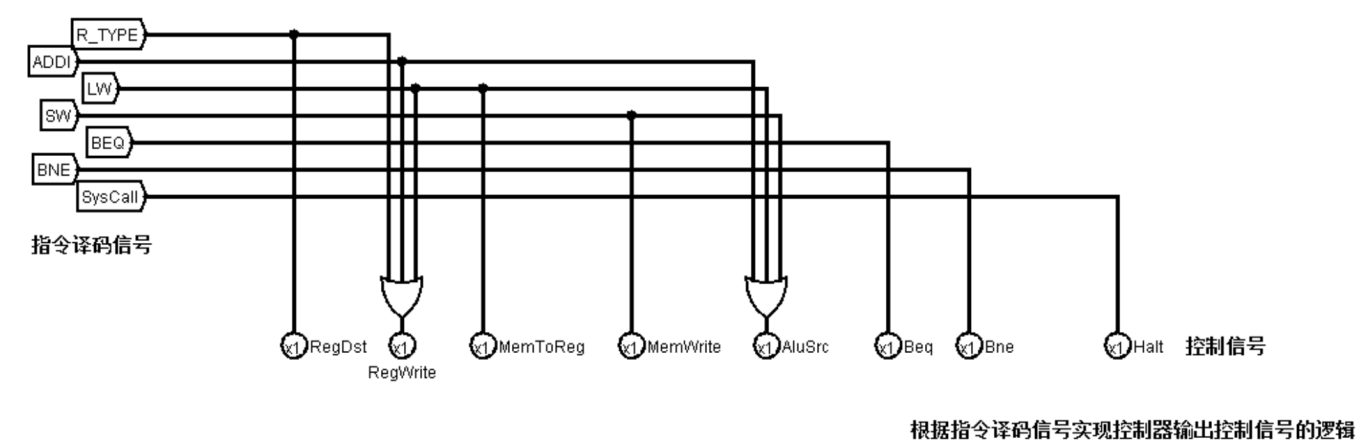
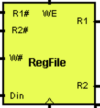
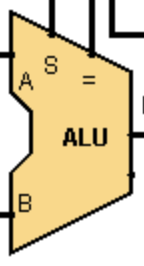


图 1.2.1 单周期MIPS(硬布线)CPU控制器电路

### 多周期MIPS(微程序)CPU设计

\*多周期MIPS(微程序)CPU功能部件

1. 存储器Mem，包括数据存储器以及指令存储器；
2. 指令计数器PC.
3. 指令寄存器IR，存放当前指令；
4. 数据存储器DR，存放要操作的数据；
5. 寄存器堆RegiFile，提供32个MIPS通用寄存器；
6. 立即数扩展器S-EXT
7. 控制器，产生控制信号
8. 算术逻辑单元ALU.

\* 控制多周期信号

在多周期MIPS(微程序)CPU中不同指令对应不同的时钟周期数，因此不能像单周期控制器一般一次性给出一条指令对应的所有控制信号，而每一条指令的执行又可以拆解为取指、译码、执行.取指和译码两个阶段所有指令对应的数据通路相同，因此现在的任务退化为分析每条指令执行阶段需要的时钟周期并给出每个时钟周期下所有的控制信号，指令的执行控制应遵循:有序进行、不会破坏系统中保存的结果，仅仅对指令本身功能要求对系统状态做出改变、保证不同周期下指令的每一步执行所用到的数据都来自正确来源，保证每一步的运算结果保存到正确输出。

\* 控制信号说明如下：

>PcSrc：PC输入源，0表示顺序寻址，1表示跳跃寻址；

>AluControl：ALU控制信号，即加法或比较运算；

>AluSrcA：ALU第一输入选择，PC还是寄存器输出；

>AluSrcB：ALU第二输入选择，R型指令输入为寄存器输出

>RegWrite：寄存器RegiFile写使能；

>RegDst：RegiFile的第二个寄存器编号由R型指令rd给出；

>MemToReg：lw指令，写入寄存器的数据来自存储器。

>PCWrite：PC写使能控制，取指令周期，分支指令执行；

>IorD：指令还是数据，0表示指令，1表示数据；

>Irwrite：指令寄存器写使能；

>MemWrite：写内存控制信号；

>MemRead：读内存控制信号；

>Beq：beq指令译码信号；

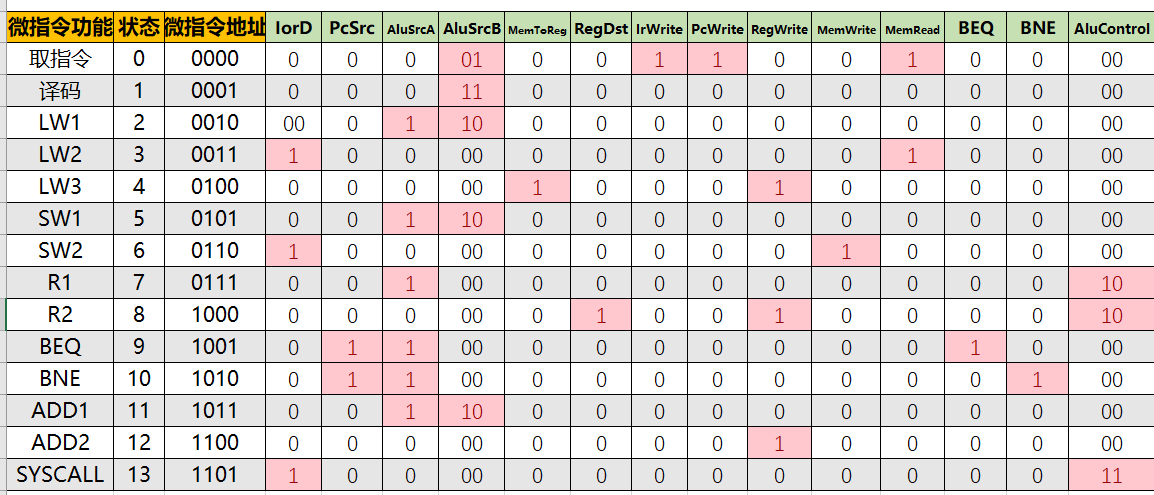
>Bne：bne指令译码信号；

表1.2.1控制信号表

\* 微程序地址转移设计

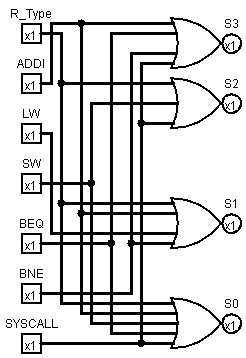
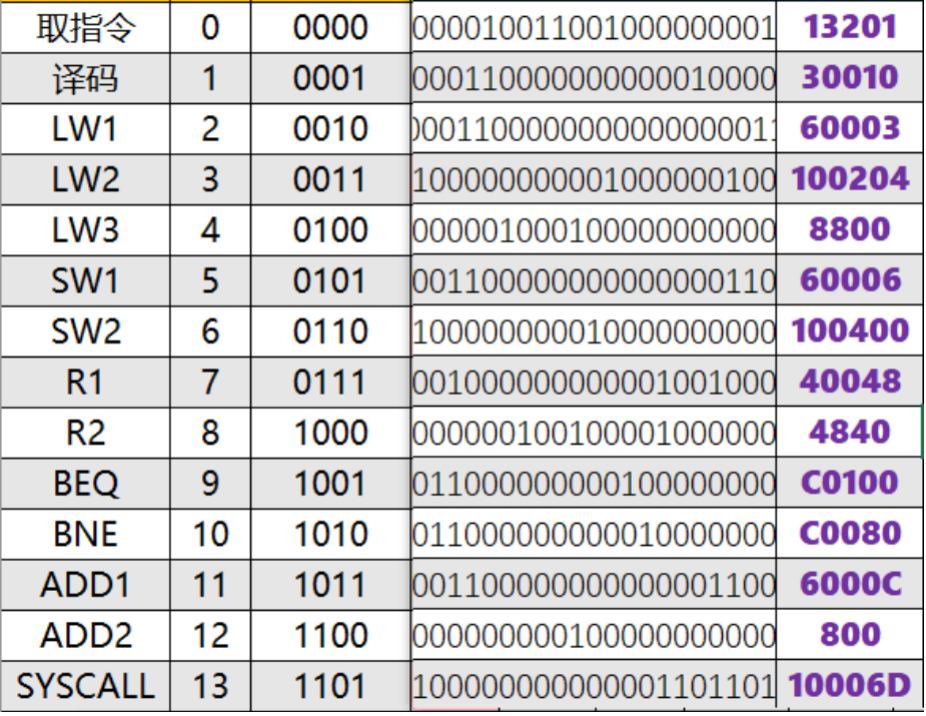


图1.2.1微程序地址转移模块

\* 多周期MIPS(微程序)CPU控制器

表1.2.2 微指令合集



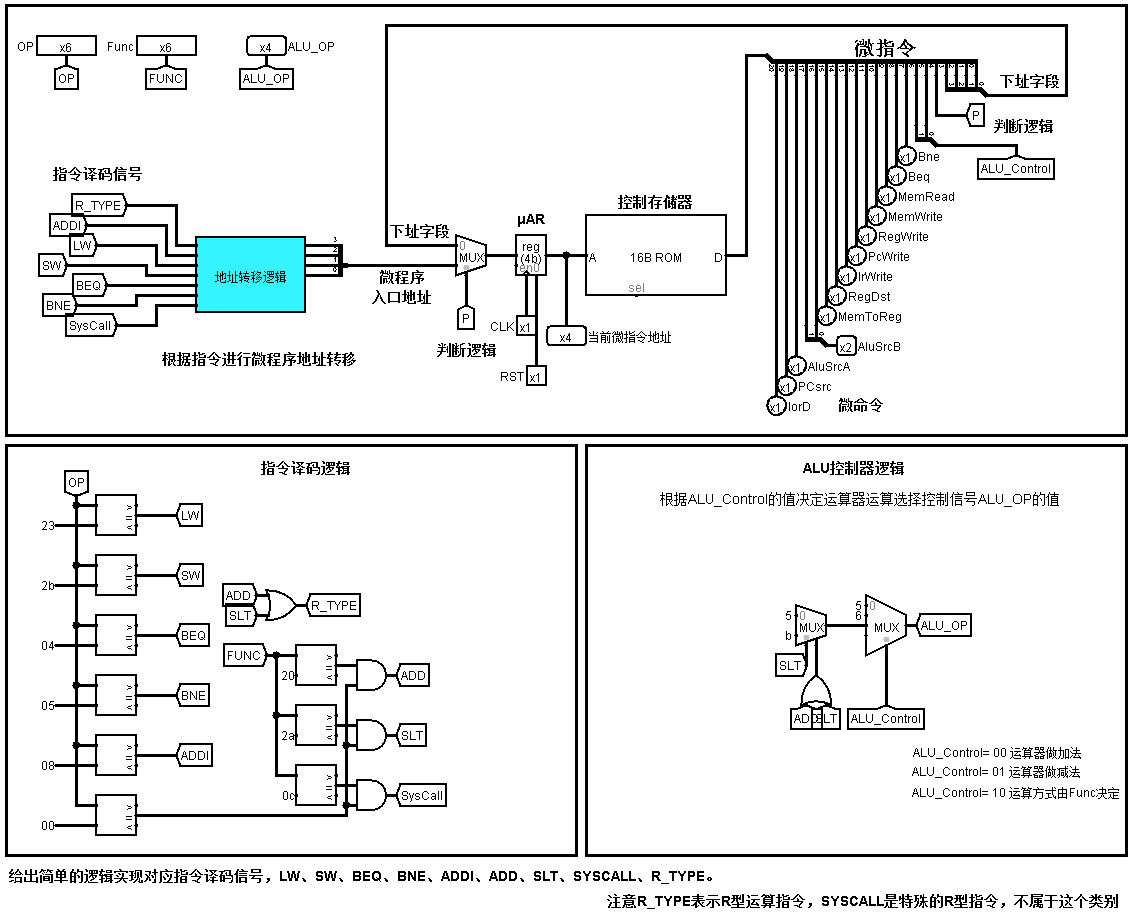


图1.2.2 微程序控制器

### 多周期MIPS(硬布线)CPU设计

\*多周期MIPS(硬布线)CPU功能部件

除了控制器不同,其他一致.不作重复赘述

\*控制多周期信号

与微程序CPU一致.不作赘述

\*多周期MIPS(硬布线)控制器

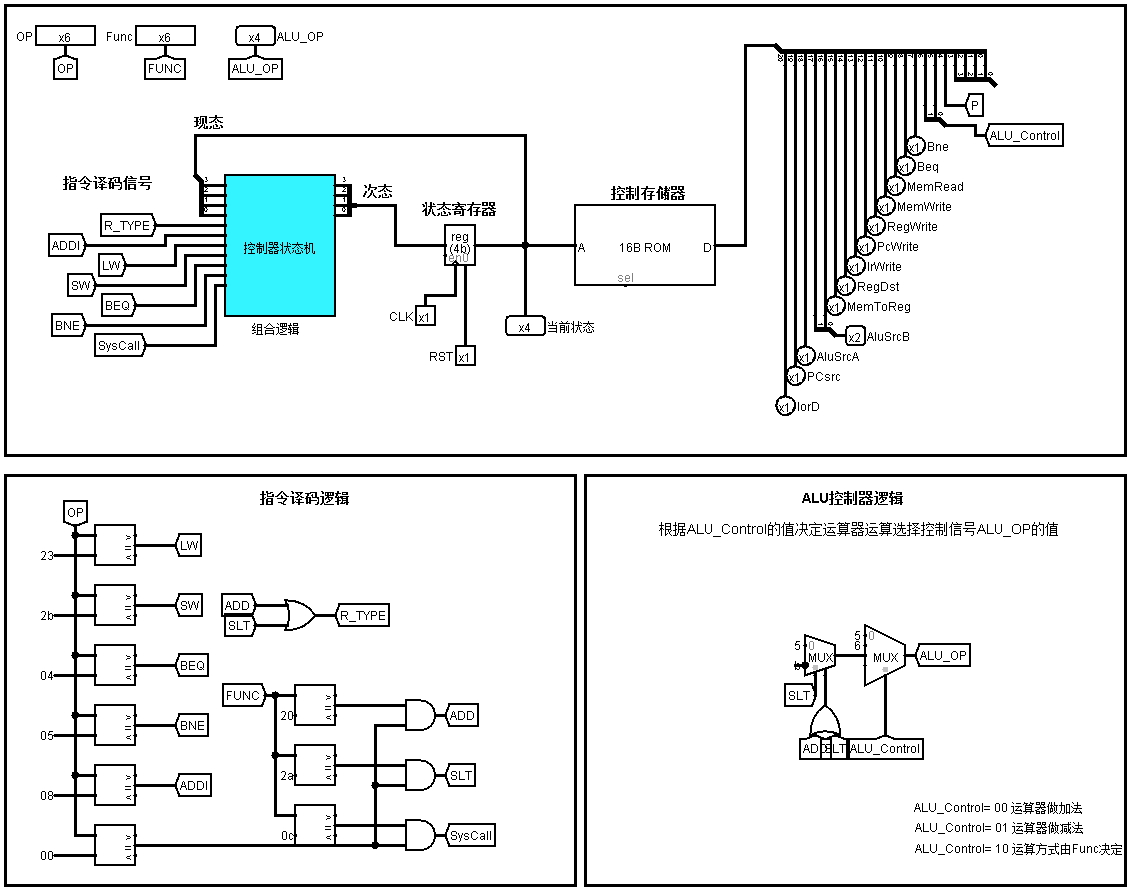
其状态编码和微程序控制器一致,但是需要增加有限状态机FSM实现现态到次态的越迁.

**表1.2.3 状态转换表**



\* 多周期MIPS(硬布线)CPU控制器

采用21位二进制编码，弃用下址字段.

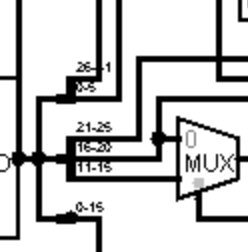


**图1.2.3多周期MIPS(硬布线)CPU控制器**

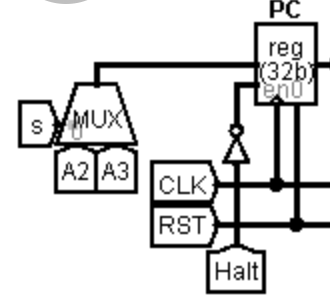
## 实验步骤

1. 单周期MIPS(硬布线)CPU组装

\* 指令解析

图1.3.1单周期指令解析模块

\* 逻辑地址转移

图1.3.2 逻辑地址转移模块

\* 单周期MIPS(硬布线)CPU组装

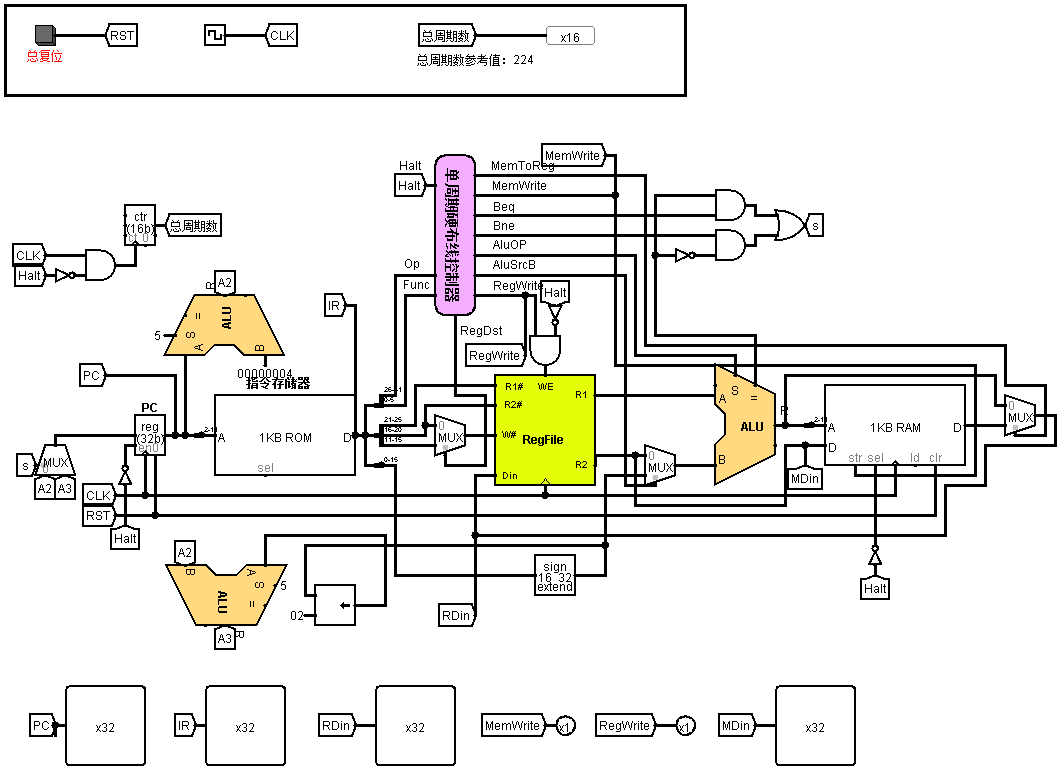
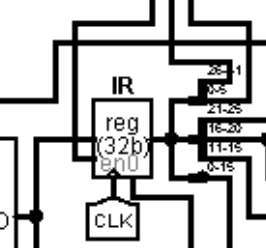


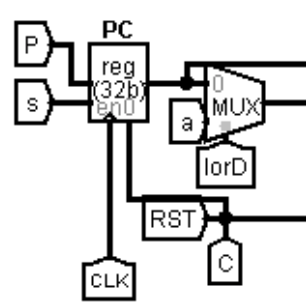
图1.3.3 单周期MIPS(硬布线)CPU组装图

1. 多周期MIPS(微程序)CPU组装

\* 指令解析

图1.3.4 指令解析模块

\* 逻辑地址转移

图1.3.5 逻辑地址转移

\* 多周期MIPS(微程序)CPU组装

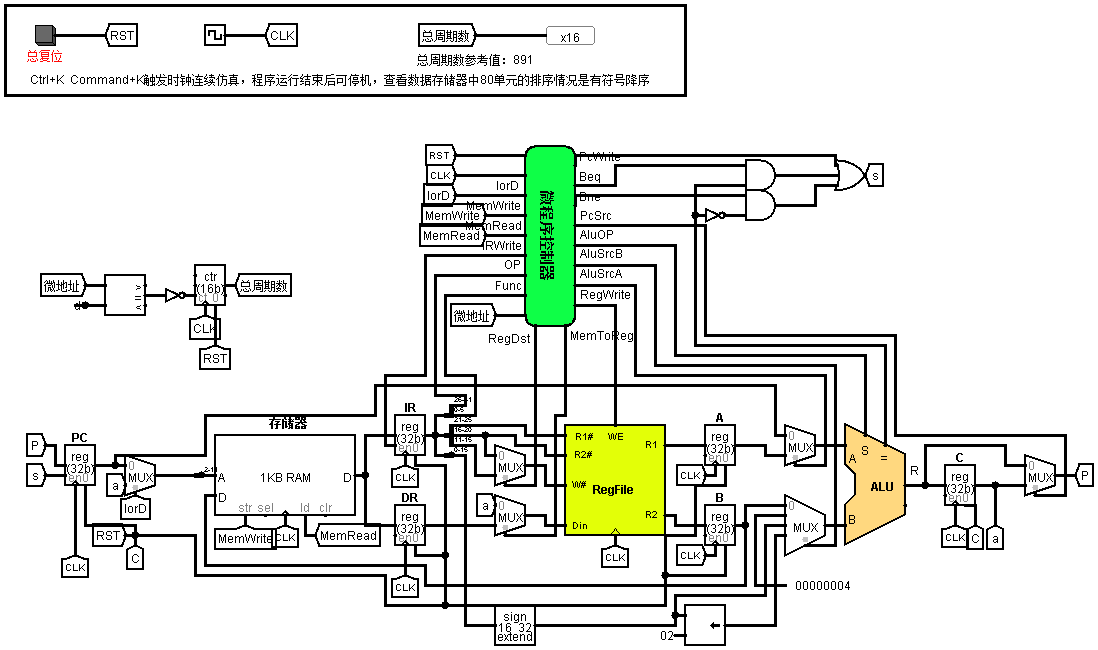


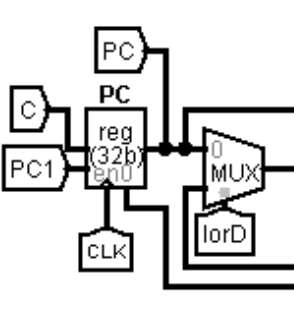
图1.3.6多周期MIPS(微程序)CPU组装

1. 多周期MIPS(硬布线)CPU组装

\* 指令解析

图1.3.4 指令解析模块

\* 逻辑地址转移

图1.3.5 逻辑地址转移

\* 多周期MIPS(微程序)CPU组装

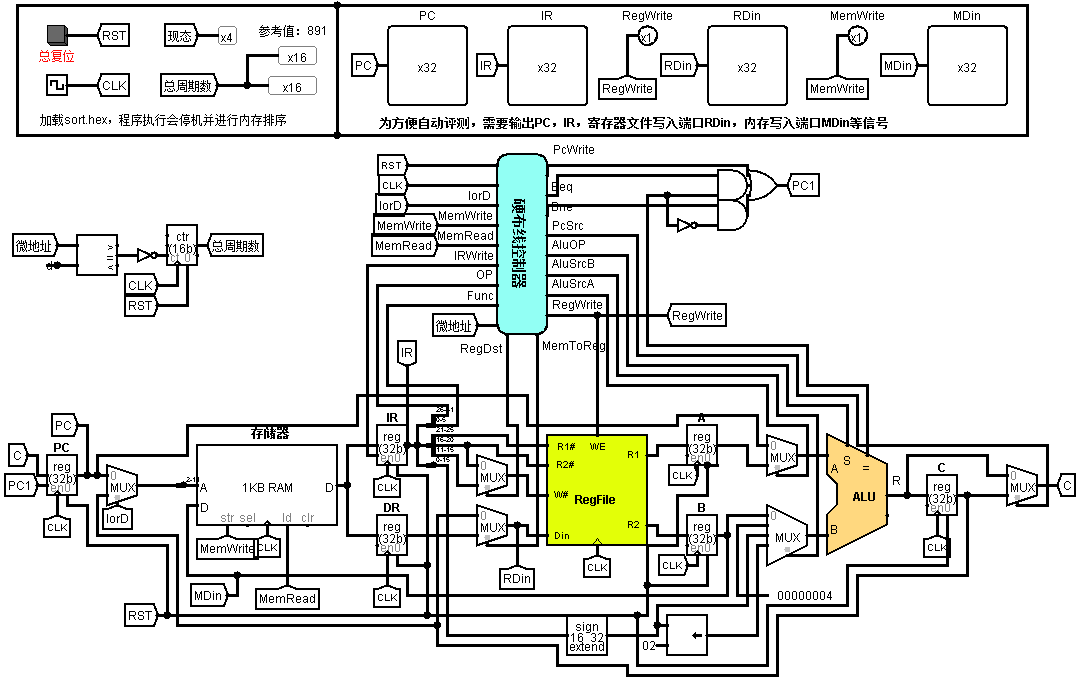


图1.3.6多周期MIPS(微程序)CPU组装

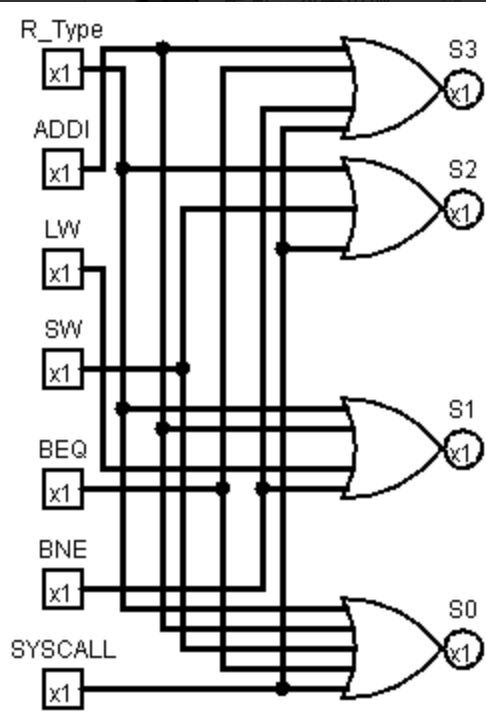
## 故障与调试

### 程序无法自行停止

**故障现象：程序执行完成后无法停止,一直执行后面的空指令.**

**原因分析：控制器中对SYSCALL语句解析错误.多周期微程序CPU地址逻辑转移连线出错.s3端口未调用正常SYSCALL导致程序无法正常停止.**

**解决方案：将s3端口连接SYSCALL,恢复正常系统调用功能.**



**图1.4.1 微程序地址转移逻辑**

## 测试与分析

### 单周期MIPS(硬布线)CPU测试

\* 开始执行排序算法

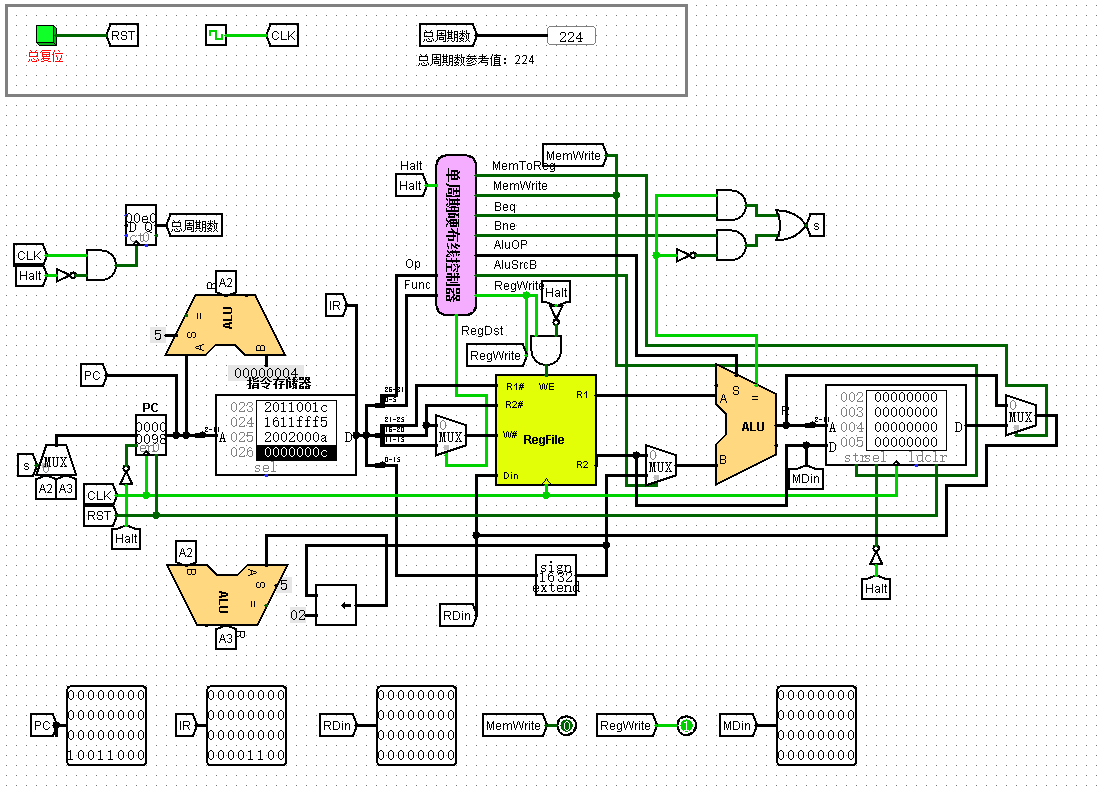


图1.5.1算法执行结果

\* 排序结果

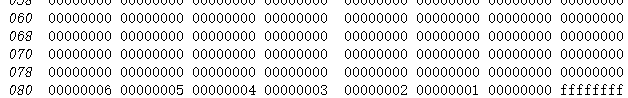


图1.5.2排序结果

### 多周期MIPS(微程序)CPU测试

\* 载入sort.hex开始执行

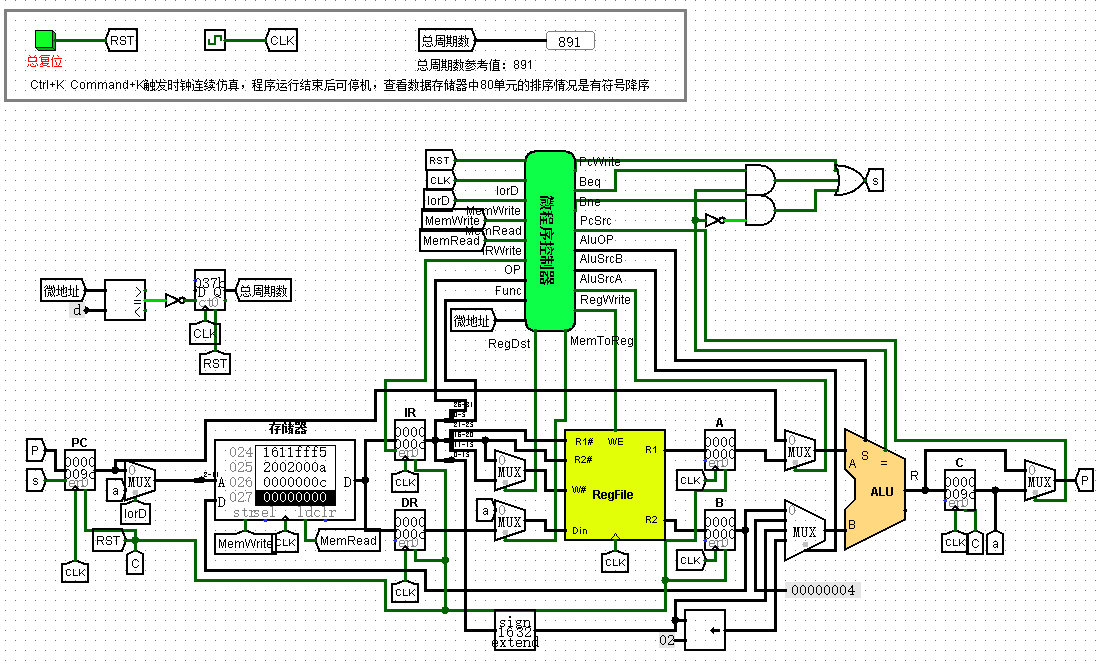


图1.5.3算法执行结果

\* 排序完成

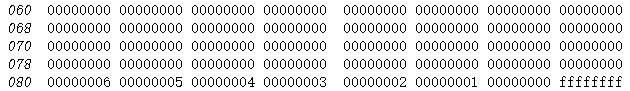


图1.5.4 排序结果

### 多周期MIPS(硬布线)CPU测试

\* 载入sort.hex开始执行

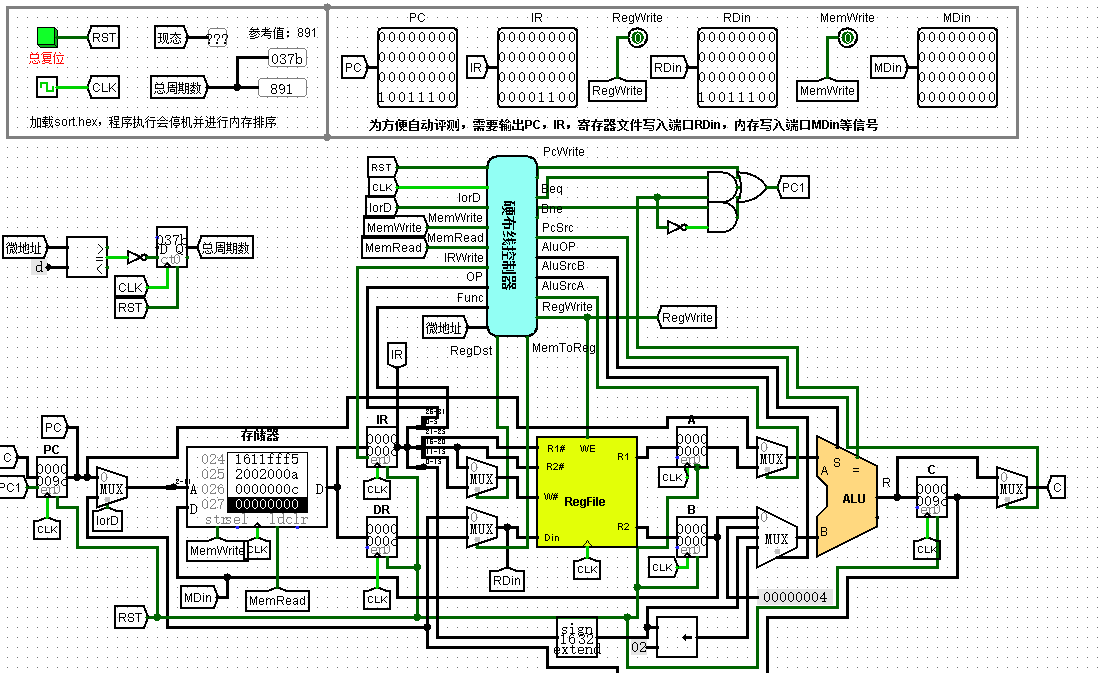


图1.5.5算法执行结果

\* 排序完成

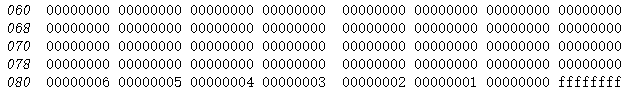


图1.5.6排序结果

# 总结与心得

## 实验总结

1. 完成了32位定长指令的译码和指令的解析
2. 完成了对各个寄存器，存储器的读写
3. 完成了控制信号的生成
4. 完成了使用多路选择器实现了对多个数据来源的选择和控制。
5. 完成了单周期MIPS(硬布线)CPU控制器；
6. 完成了多周期MIPS(微程序)CPU地址转移；
7. 完成了多周期MIPS(硬布线)CPU控制器状态机；
8. 完成了单周期和多周期的MIPS CPU的综合数据通路，实现了简单的冒泡程序的逻辑功能；

## 实验心得

1) 设计电路过程需要非常细心,不然会导致各种不能预料和难debug的错误.

2) 需要熟练使用logisim各种工具并运用到课堂学习的内容上,熟练使用各种快捷键和脚本,会大大减少工作量

3) 多用鼠标或者数位板画图,少用笔记本自带的触摸板.

4) 多看看已有的模版和已有的工具来解决问题,不需要凡事都需要自己造工具来解决,这样会更加高效率.

5) 画图前制定良好的计划章程,这样在实行过程中才不会难以发现隐蔽的错误.

# 参考文献

1. DAVID A.PATTERSON(美).计算机组成与设计硬件/软件接口(原书第5版).北京:机械工业出版社.
2. David Money Harris(美).数字设计和计算机体系结构（第二版）. 机械工业出版社
3. 谭志虎,秦磊华,胡迪青.计算机组成原理实践教程.北京:清华大学出版社，2018年.
4. 秦磊华，吴非，莫正坤.计算机组成原理. 北京:清华大学出版社，2011年.
5. 袁春风编著. 计算机组成与系统结构. 北京:清华大学出版社，2011年.
6. 张晨曦，王志英. 计算机系统结构. 高等教育出版社，2008年.

|  |
| --- |
| 一、原创性声明 |
| 本人郑重声明本报告内容，是由作者本人独立完成的。有关观点、方法、数据和文献等的引用已在文中指出。除文中已注明引用的内容外，本报告不包含任何其他个人或集体已经公开发表的作品成果，不存在剽窃、抄袭行为。  特此声明！  **作者签字: 签名** |
| 二、对课程实验的学术评语（教师填写） |
|  |
| 三、对课程实验的评分（教师填写） |
| |  |  |  |  | | --- | --- | --- | --- | | 评分项目  （分值） | 报告撰写  （30分） | 课设过程  （70分） | 最终评定  （100分） | | 得分 |  |  |  | |
| **指导教师签字:** |