***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： | 校交1701班 |
| 学 号： | U20713666 |
| 姓 名： | 蒋为 |
| 电 话： | 18277380223 |
| 邮 件： | [1142529848@qq.com](mailto:1142529848@qq.com) |
| 完成日期： | 2019/12/13 |



目 录

[1 CPU设计实验 2](#_Toc27489830)

[1.1 设计要求 2](#_Toc27489831)

[1.2 方案设计 2](#_Toc27489832)

[1.3 实验步骤 4](#_Toc27489833)

[1.4 故障与测试 10](#_Toc27489834)

[1.5 测试与分析 10](#_Toc27489835)

[2 总结 13](#_Toc27489836)

[2.1 实验总结 13](#_Toc27489837)

[2.2 实验心得 13](#_Toc27489838)

[参考文献 14](#_Toc27489839)

# CPU设计实验

## 设计要求

构建一个32位MIPS CPU处理器，包括单周期硬布线CPU、多周期微程序CPU以及多周期硬布线CPU，该处理器应支持核心指令集中列出的所有指令，见表1.1。具体指令功能参见附件中的MIPS标准文档。最终设计完成的CPU应能运行标准测试程序。

表 1.1 核心指令集

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **指令** | **格式** | **备注** |
| 1 | Add | add $rd, $rs, $rt | 指令功能及指令格式  参考MIPS32指令集 |
| 2 | Add Immediate | addi $rt, $rs, immediate |
| 3 | Load Word | lw $rt, offset($rs) |
| 4 | Store Word | sw $rt, offset($rs) |
| 5 | Branch on Equal | beq $rs, $rt, label |
| 6 | Branch on Not Equal | bne $rs, $rt, label |
| 7 | Set Less Than | slt $rd, $rs, $rt |
| 8 | syscall（display or exit） | syscall | 系统调用，用于停机 |

## 方案设计

### 单周期CPU功能部件

1. 指令计数器PC，用于存放当前指令执行的位置。
2. 指令计数器IR，用于存放CPU当前要执行的指令。
3. 数据存储器，因为是单周期，所有指令都必须在一个时钟周期内完成，只采用一个存储器同时完成对指令和数据的操作是困难的，所以还要再加一个数据存储器。
4. 位扩展器Extern，用于将I型指令中的16位立即数扩展为32位。
5. 单周期硬布线控制器，产生控制信号，依据指令控制各个部件。
6. 寄存器堆RegFile，提供32个MIPS通用寄存器。
7. 算术逻辑单元ALU，用于实现需要的运算功能。

### 多周期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. 存储器Mem，包括数据存储器以及指令存储器。
3. 指令寄存器IR，存放当前的指令。
4. 数据存储器DR，存放当前要操作的数据。
5. 寄存器堆RegFile，提供32个MIPS通用寄存器。
6. 位扩展器，用于将I型指令中的16位立即数扩展为32为。
7. 多周期微程序/硬布线控制器，产生控制信号，控制指令执行的数据通路。
8. 算数逻辑单元ALU，用于实现需要的运算功能。
9. 增加了三个寄存器A，B，C，暂存RegFile和ALU的数据输出。

### 地址转移逻辑器

### 执行完一条一条指令后，下一步该执行哪条指令，根据实验要求的8条核心指令指令分析，可以知道有时会顺序执行，继续执行下一条语句，有时也会有条件分支，跳转到当前地址加上偏移地址。所以有两种可能的跳转，一次只能选择一个值，需要一个控制信号，由操作控制器生成。

### 多周期控制器

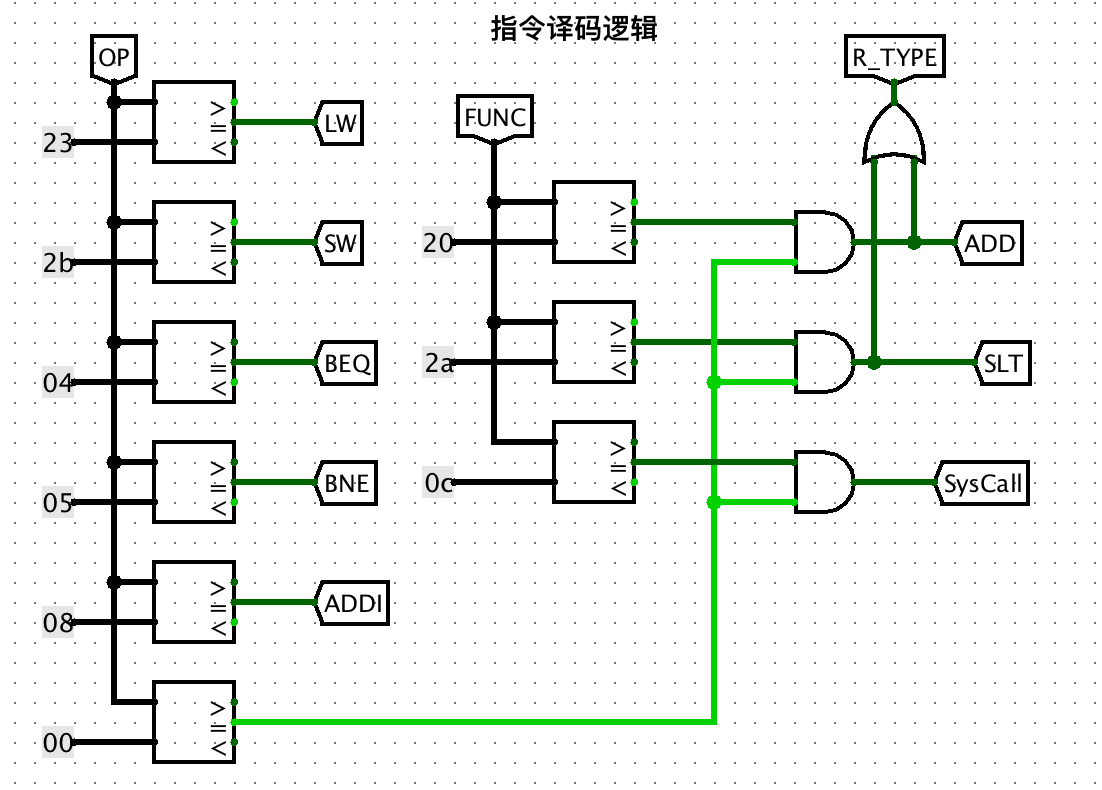
操作控制器根据指令的操作吗和func产生控制信号。

1. 列出所有指令下，各个数据项的值。
2. 将每一项数据合并，如果有一项数据有多种情况，说明该数据需要用控制信号选择，否则可以不用选择，直接直连。
3. 控制信号综合起来，列出各指令下的值，从而就可以得出控制信号的逻辑表达式，进而使用logisim的自动生成电路功能生成相应电路。
4. 多周期控制器可分为微程序和硬布线控制器，两种各有其实现方法。

## 实验步骤

### 指令解析

将输入的32位操作吗码用分线器接出来，最高6位位op，21-25位位RS寄存器编号，16-20位为RT寄存器编号，11-15位为RD寄存器编号，0-5位为功能码func。取低16位作为I型指令的立即数操作数。单周期和多周期指令译码方式相同，如图1-1所示：



**图1-1**

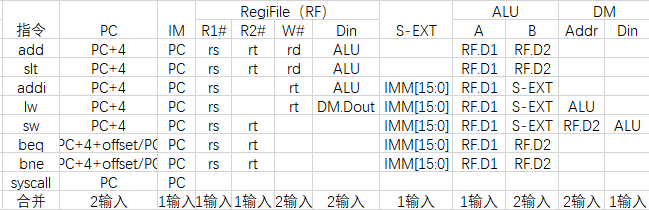
### 实现地址转移逻辑

### PC+4接到MUX的0输入端，16位立即数扩展结果左移两位，加上PC+4，接到MUX的1输入端。实验中的分支转移指令位BEQ和BNE，两条指令均为I型指令，所以可以利用ALU的equal信号，选择信号即为beq\*equal+bne\*equal形成分支信号。

### 单周期硬布线控制器

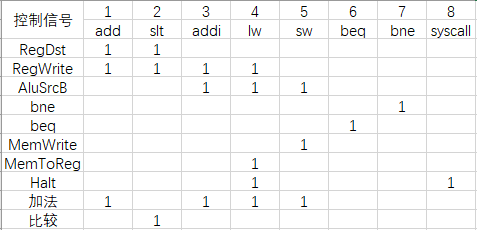
### 构建数据通路，主要功能部件的输入值，具体如表1-1所示

表 1.1 单周期功能部件输入来源表

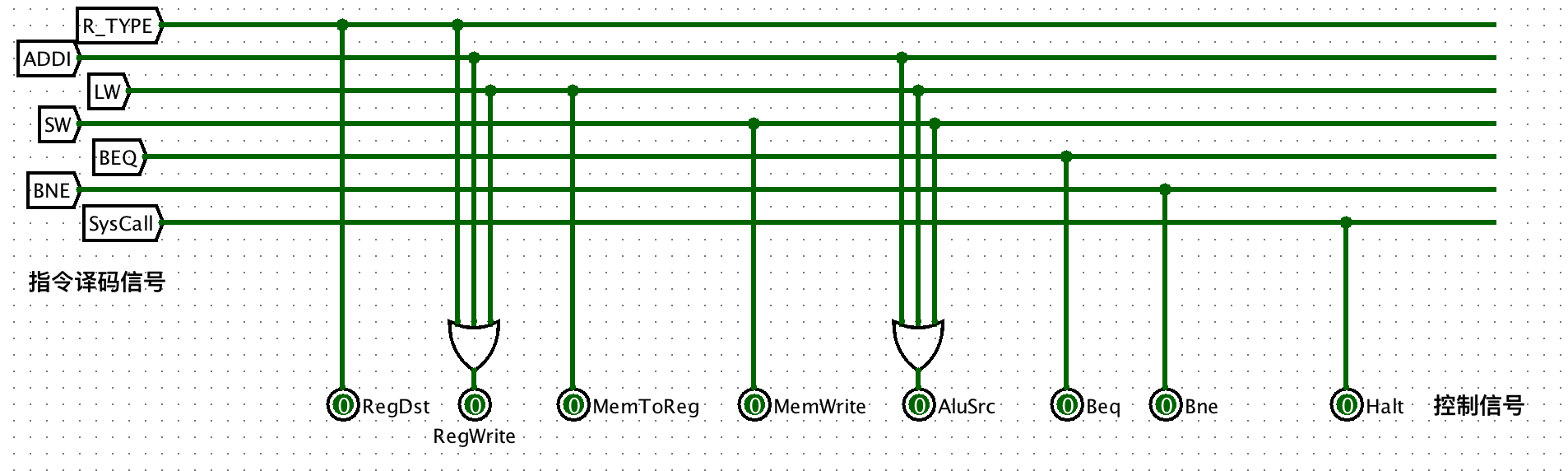


（2）列出所有功能部件，多路选择器控制信号，运算操作数选择产生的条件，如表1-2所示：

表 1.2 单周期硬布线控制器信号表



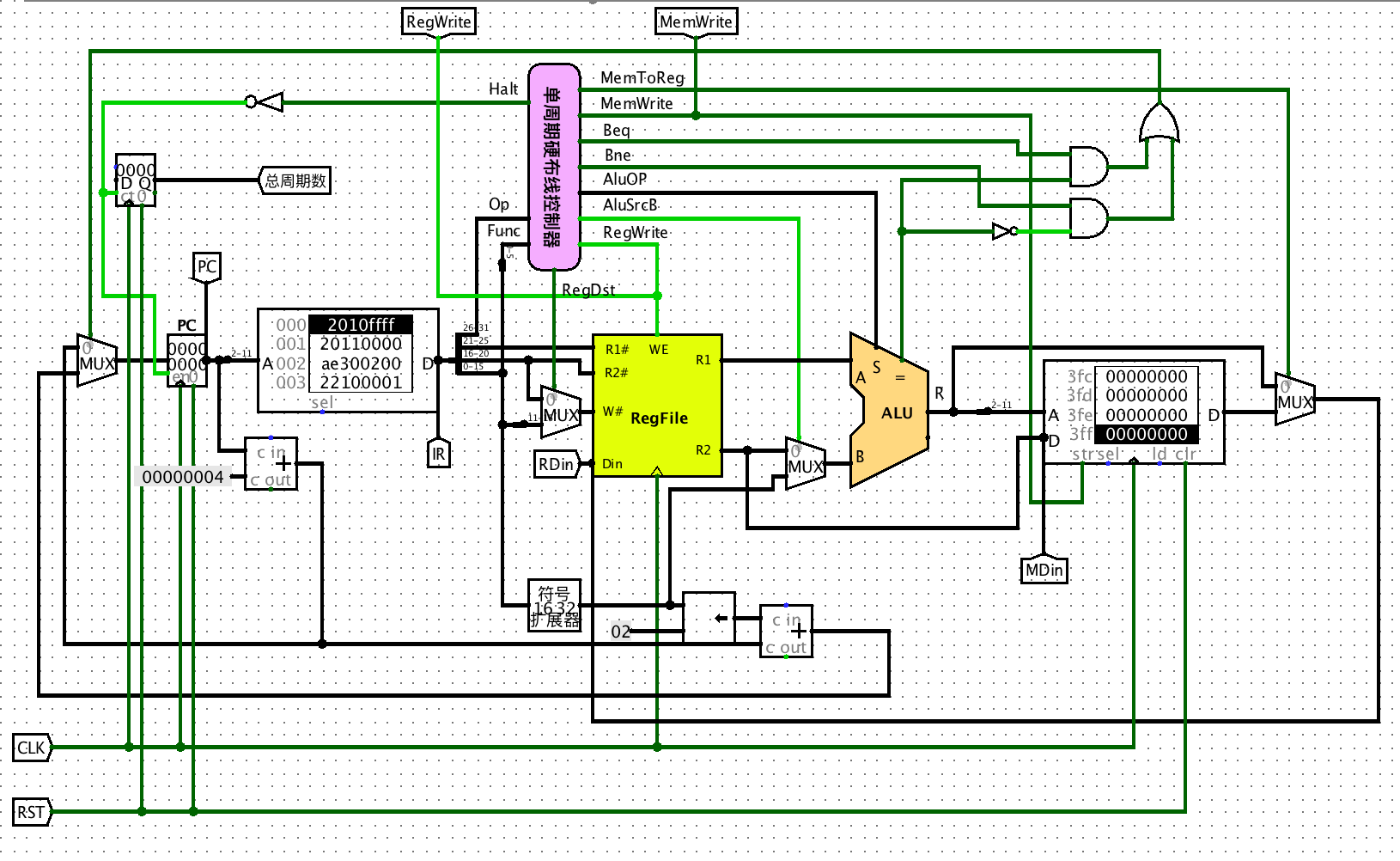
（3）最终形成的单周期硬布线控制电路如图1-2所示：



**图1-2**

### 单周期CPU结构图

按照mooc上的逻辑图链接可得图1-3：



**图1-3**

### 多周期的实现

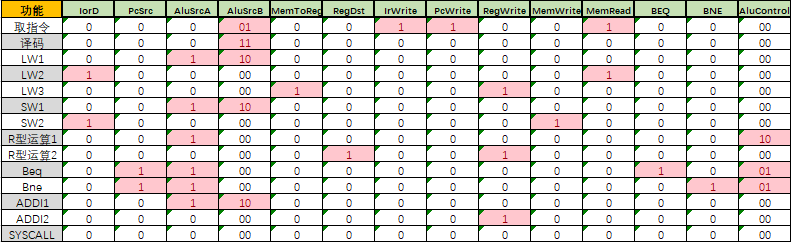
多周期处理器中，指令的执行需要占用多个时钟周期，不同的指令需要的时钟周期数不一定相同，相同指令在不同时钟节拍下产生的控制信号也不同。具体情况如下：

1. 不区分指令存储器和数据存储器，全部保存在同一个存储器中。
2. ALU和RegFile可以在一条指令中执行多次。
3. 增加了A，B，C三个寄存器辅助存储数据。
4. PC计数器由于不同指令的时钟周期数不同，因此PC不再仅由时钟周期控制，而是增加了专门的写操作控制信号。

### 多周期控制信号

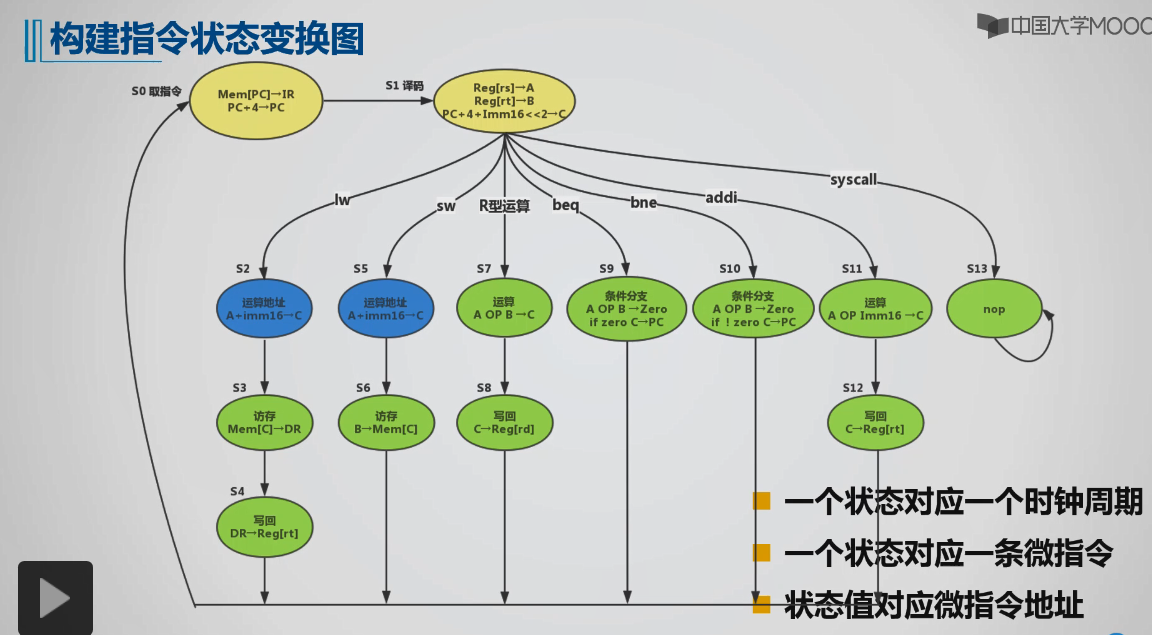
多周期CPU控制下，不同指令对应的时钟周期数不一定相同，每条指令一般分为“取指->译码->执行”三个阶段，其中取值和译码两个阶段对应的数据通路是相同的，因此任务为分析每条指令执行阶段需要的时钟周期并给出每个时钟周期下所有的控制信号，指令的执行控制应有序进行，不破坏系统中保存的控制信号，保证每一步执行所用到的数据来源都全部正确。所有指令不同阶段的控制信号表如表1-3所示：

表 1.3 指令控制信号表

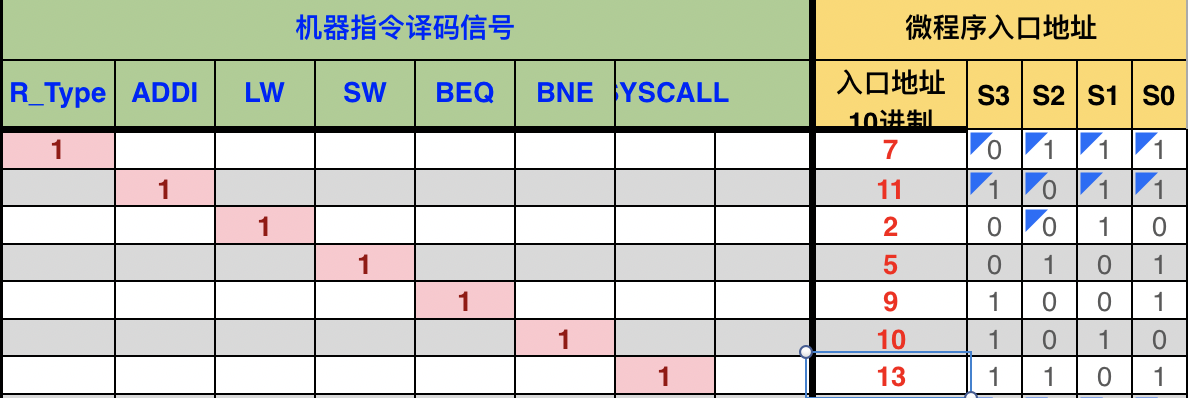


### 多周期微程序控制器

状态编码图由mooc提供的图1-4所示的指令转换图得出，根据实验提供的excel表算出状态编码如图1-5所示，相应微指令也可得出如图1-6所示：



**图1-5**

****

**图1-6**

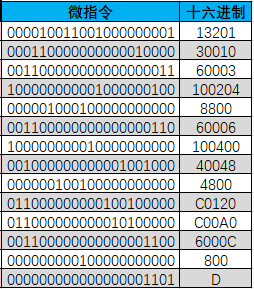
生成的地址转移逻辑表达式为

S3 = ADDI + BEQ + BNE + SYSCALL；

S2 = R\_Type + SW + SYSCALL；

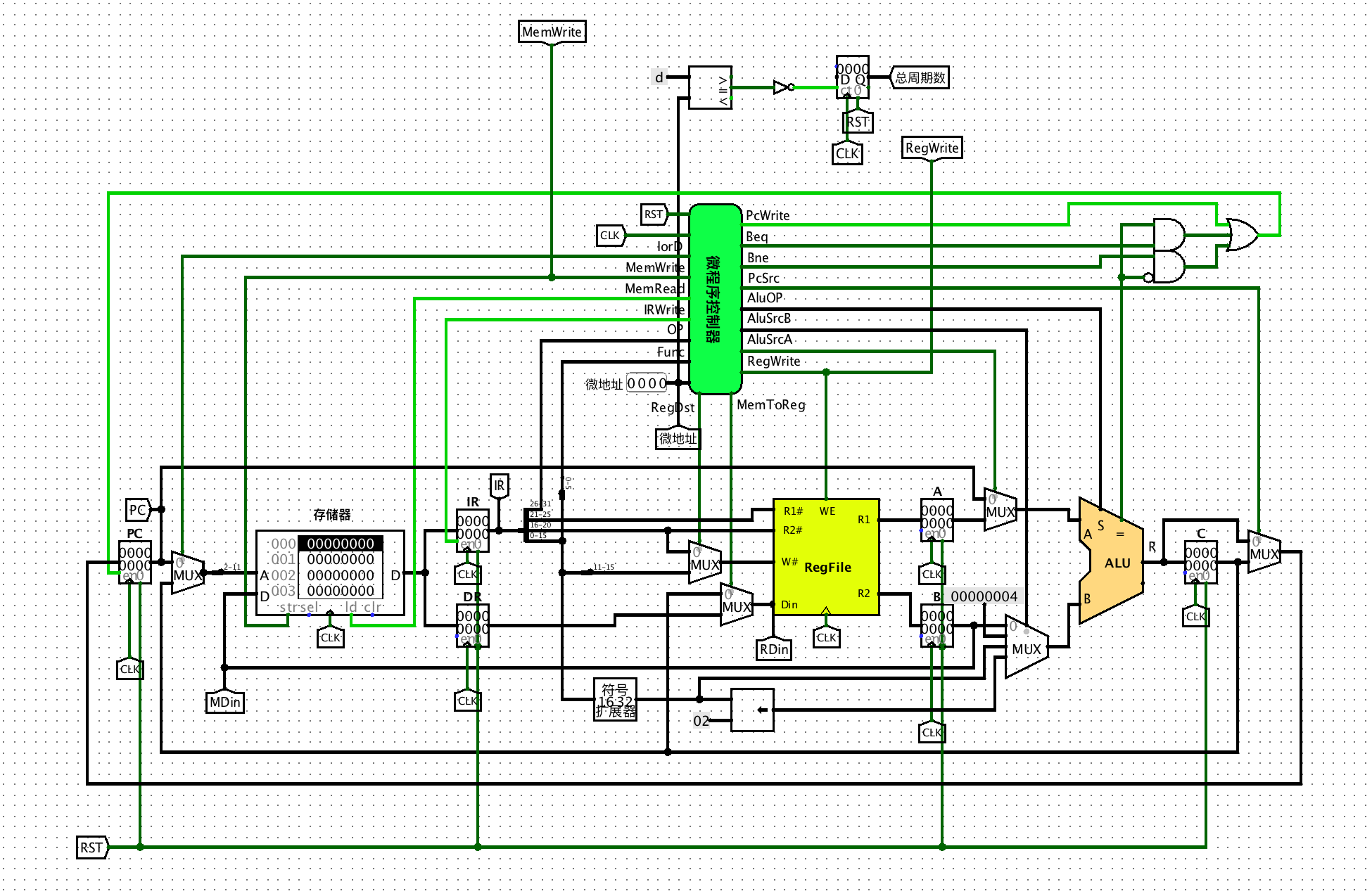
S1 = R\_Type + ADDI + LW + BNE；

S0 = R\_Type + ADDI + SW + BEQ + SYSCALL。



**图1-7 微指令编码**

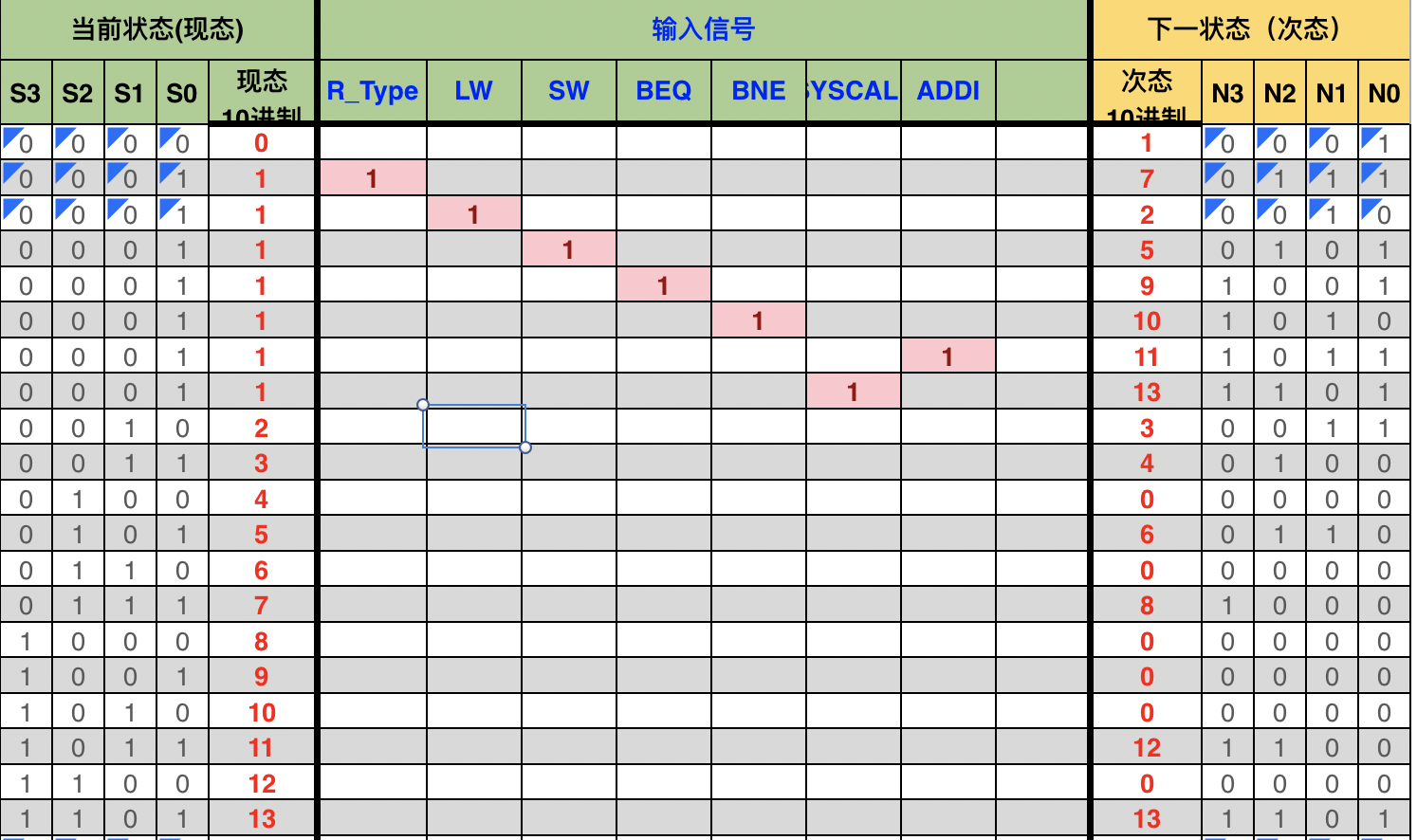
连接后的logisim图如图1-8所示：



**图1-8**

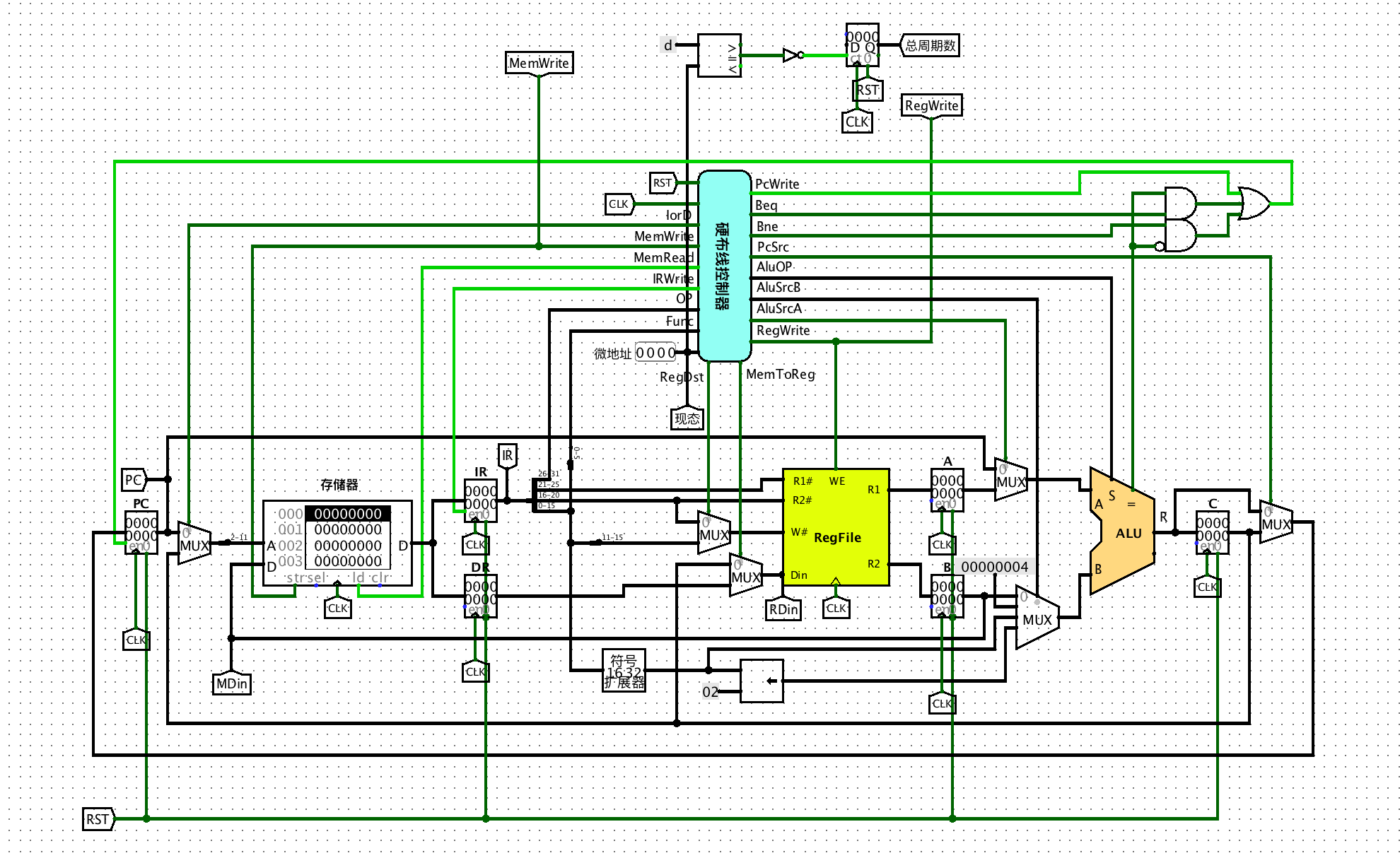
### 多周期硬布线控制器

多周期硬布线的状态编码和微程序相同，状态转移如图1-5的FSM图所示，配合实验提供的excel文件，按图填写，最后自动计算可得如图1-9所示的状态转换图：



**图1-9**

多周期硬布线PU总体结构图如图1-10所示：



**图1-10**

## 故障与测试

（1）在实现单周期硬布线存储器存储时，总是不能在80号存储单元降序排列数据，经过分析发现，是自己在PC取数阶段不同指令取出的位数错了才导致这个问题。

（2）CPU运行到结束后没有自动停机而是继续运行下去，后来发现是自己的Halt信号未连接到PC计数器和相应计数器的En使能端导致的。

## 测试与分析

### 单周期硬布线测试

测试结果如图1-11与1-12所示所示：



图1-11

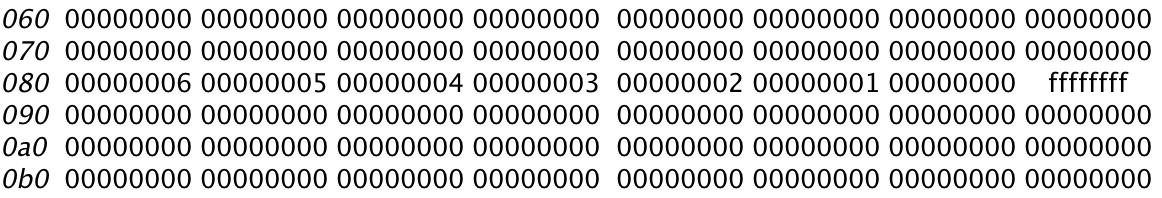


图1-12

由图可知，测试结果正确。

### 多周期微程序测试

测试结果如图1-13与1-14所示：

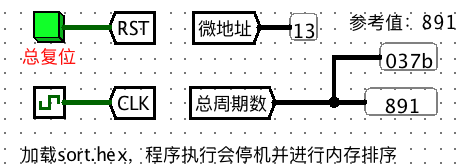


图1-13

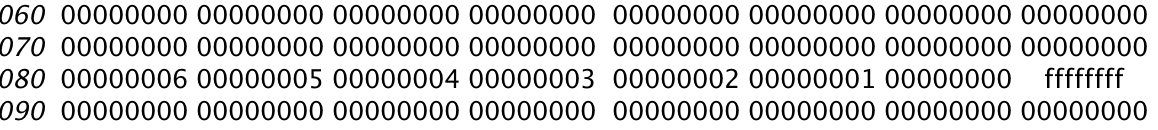


图1-14

### 多周期硬布线测试

测试结果如1-5与1-16所示：

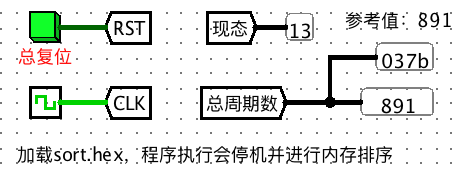


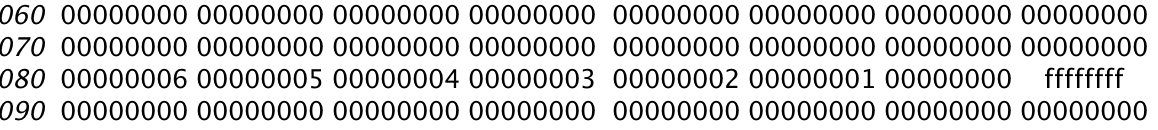
图1-15

图1-16

# 总结

## 实验总结

本次实验主要完成了如下几个工作：

1. 实现了MIPS32位定长指令的解析
2. 实现设计了程序如何执行指令
3. 实现了MIPS多周期CPU的控制存储器
4. 实现了MIPS多周期CPU微程序地址转移逻辑
5. 实现了MIPS多周期CPU硬布线控制器状态机
6. 实现了MIPS CPU数据通路的ligisim设计，能支持8条简单的指令运行。

## 实验心得

1. 学会了使用logisim的自动生成电路功能，这对编写有逻辑表达式的电路非常有帮助。
2. 学会了CPU的运行机制，大致上懂得了如何设计CPU并让其运行起来。
3. 熟悉使用了MIPS核心指令，并且学会了把PC指令分为三种不同类型指令的操作流程。
4. 对实验内容了解更深，课上听老师讲感觉迷迷糊糊，一到了做实验什么原理之类的都要弄清，这条线为什么连这里，为什么这里需要使用数据选择器都是有理由的。
5. 实验的MOOC对我的帮助很大，如果MOOC没有这次CPU实验的教程以及老师给出的资料，感觉自己可能完不成这次实验，计算机组成原理课程的MOOC真的是计算机学院第一MOOC，老师也是非常好的老师，很有耐心，要是每个课程都像计算机组成原理这门课就好了。

# 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:** |
| 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:** |