***2020***



**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： | CS1804 |
| 学 号： | U201814604 |
| 姓 名： | 黄俊淇 |
| 电 话： | 13977133562 |
| 邮 件： | [2474268727@qq.com](mailto:2474268727@qq.com) |
| 完成日期： | 2020.12.8 |



目 录

[1 CPU设计实验 2](#_Toc58171809)

[1.1 设计要求 2](#_Toc58171810)

[1.2 方案设计 2](#_Toc58171811)

[1.3 实验步骤 8](#_Toc58171812)

[1.4 故障与调试 15](#_Toc58171813)

[1.5 测试与分析 15](#_Toc58171814)

[2 总结与心得 16](#_Toc58171815)

[2.1 实验总结 16](#_Toc58171816)

[2.2 实验心得 16](#_Toc58171817)

[参考文献 17](#_Toc58171818)

# CPU设计实验

## 设计要求

（1）理解定长指令周期三级时序系统的设计，能利用该时序构造硬布线控制器，支持5条典型MIPS指令在单总线CPU上运行，CPU能运行内存冒泡排序。最终调试运行标准测试程序sort-5.hex。

（2）理解变长指令周期三级时序系统的设计，能利用该时序构造硬布线控制器，支持5条典型MIPS指令在单总线CPU上运行，CPU能运行内存冒泡排序。最终调试运行标准测试程序sort-5.hex。

5条典型指令如图1.1所示。

图形用户界面, 文本, 应用程序, 聊天或短信

描述已自动生成

图1.1 5条典型指令

（3）理解现代时序控制器中断机制的实现原理，能为采用现代时序单总线结构的MIPS CPU增加中断处理机制，可实现多个外部按键中断事件的随机处理，本实验需要完成现代时序微程序控制器的基础上完成，需要增加硬件数据通路，增加中断返回指令eret的支持，需要中断服务程序配合。最终调试运行标准测试程序sort-5.hex。

## 方案设计

### 单总线CPU设计（定长指令周期）

（1）指令译码逻辑

指令译码逻辑将指令的操作码字段转换为对应的运算信号，如addi信号，lw信号，sw信号等。

查询实验包内的《MIPS32指令手册》得到LW、SW、BEQ、ADDI指令的OP字段如表1. 1所示。

表1. 1 部分指令OP字段值

|  |  |
| --- | --- |
| 指令 | OP字段 |
| LW | 0x23 |
| SW | 0x2b |
| BEQ | 0x04 |
| ADDI | 0x08 |
| SLT | 0x2a |

只需使用比较器将OP与各指令的OP字段值进行比较便可得到以上指令对应的控制信号。

而SLT为R型指令，其Func字段为0，所以SLT指令译码时还需加上Func字段的比较。

（2）时序发生器FSM

由于是定长指令周期，故FSM就是s0->s1->s2->…->s10。

（3）时序发生器输出函数

可以由图1.5填入相应的真值表生成相应电路。

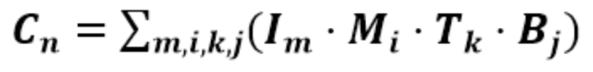
表格

描述已自动生成

图1.2输出函数真值表

（4）硬布线控制器组合逻辑单元

可以通过下式列出所有微操作信号的产生条件，并填入相应的excel表格生成逻辑表达式。



（5）硬布线控制器

通过状态机即可实现。

下降沿选择原因呢：为了让其他计算在时钟调变之前完成，如果选择上升沿会导致计数重复等问题。

（6）单总线CPU设计

在单总线CPU中，将sort-5中的指令复制到RAM 中，进行仿真。最终实现降序排列。内存数据如图1.9所示。

文本

描述已自动生成

图1.3内存数据

### 单总线CPU设计（变长指令周期）

1. 指令译码器与定长指令周期相同。
2. 单总线CPU微程序入口查找逻辑

得到如表1.2所示入口地址

表1. 2 指令对应入口地址

|  |  |
| --- | --- |
| 指令 | 入口地址（十进制） |
| LW | 4 |
| SW | 9 |
| BEQ | 14 |
| ADDI | 22 |
| SLT | 19 |

1. 单总线CPU微程序条件判别测试逻辑

得到如图1.4所示的条件判别逻辑测试真值表。

图片包含 图表

描述已自动生成

图1.4条件判别真值表

1. 单总线CPU微程序控制器设计

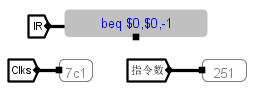
如图1.5所示，可以得到微指令信号，从而得到微程序

图表

描述已自动生成

图1.5微程序

1. 采用微程序的单总线CPU设计

在RAM中加载sort-5.hex程序，ctrl+k自动运行，程序应该运行至0x7c1节拍停下，指令计数为251，注意最后一条指令是一条beq分支指令，会跳回当前指令继续执行，是死循环。 

内存数据如下：

表格

描述已自动生成

1. 现代时序硬布线控制器状态机设计

利用数字逻辑电路相关知识设计现代时序硬布线核心部件状态机模块,实际状态机如图1.6所示：

图示

描述已自动生成

图1.6现代时序状态机

1. 现代时序硬布线控制器实现

通过硬布线状态机与状态寄存器即可实现。

### 现代时序中断机制

1. 指令译码逻辑

同现代时序。

1. 支持中断的微程序入口查找逻辑

入口地址如图1.7所示

图示

描述已自动生成

图1.7支持中断的入口地址

1. 支持中断的微程序条件判别测试逻辑

支持中断的条件判别如图1.8所示

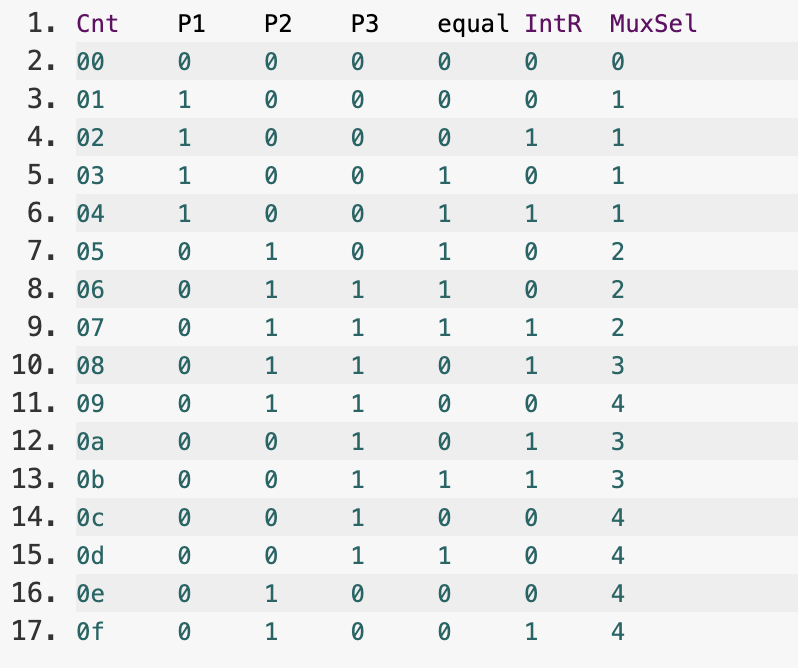


图1.8支持中断的条件判别

1. 支持中断的微程序控制器设计

支持中断的微指令表如图1.9所示

图表, 白板

描述已自动生成

图1.9支持中断的微指令表

1. 支持中断的微程序单总线CPU设计

在单总线数据通路中增加与中断相关的硬件模块，主要包括异常程序地址计数器EPC，中断使能寄存器IE，中断控制器等模块，需要在主电路中将这些模块进行有效连接。

1. 支持中断的现代时序硬布线控制器状态机设计

由1.2.3（2）中的图可以得到状态机真值表。

1. 支持中断的现代时序硬布线控制器设计

将1.2.3（4）中得到的微程序填入控制存储器，并连接状态寄存器。

## 实验步骤

### 单总线CPU构建（定长指令周期）

（1）指令译码器

整体译码电路如图1.10所示。



图1.10指令译码器电路

（2）时序发生器FSM

通过logisim的分析组合逻辑电路中的真值表功能生成电路。真值表如图1.11所示。

图片包含 表格

描述已自动生成

图1.11FSM真值表

FSM状态机电路如图1.12所示。

图示, 示意图

描述已自动生成

图1.12FSM电路

（3）时序发生器输出函数

输出函数电路如图1.13所示。

图示, 示意图

描述已自动生成

图1.13输出函数电路

（4）硬布线控制器组合逻辑单元

以Pcout为例，填入图1.14所示组合逻辑电路分析的表达式，生成所需电路。

图形用户界面, 文本, 应用程序

描述已自动生成

图1.14硬布线控制器组合逻辑单元生成

（5）硬布线控制器

电路如图1.15所示。

图示

描述已自动生成

图1.15硬布线控制器电路

（6）单总线CPU设计

加载镜像。如图1.16所示。

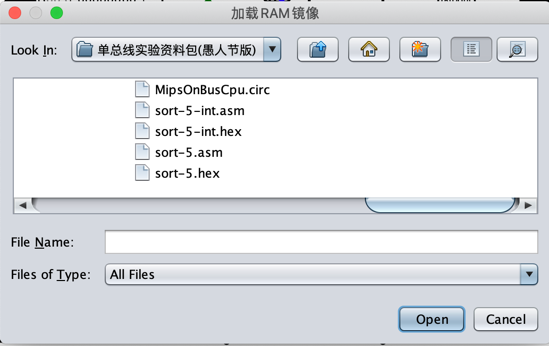


图1.16加载镜像

### 单总线CPU构建（变长指令周期）

1. 指令译码器与定长指令周期相同。
2. 单总线CPU微程序入口查找逻辑

将对应的入口地址填入Excel表格，如图1.17所示，从而得到逻辑表达式，如图1.18所示

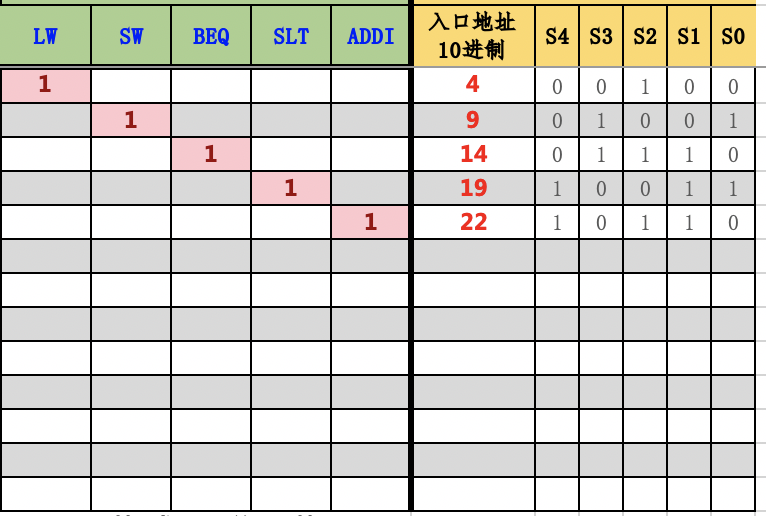


图1.17入口地址表



图1.18入口地址逻辑表达式

最后通过logisim自动生成电路，如图1.19所示

图示

描述已自动生成

图1.19入口地址电路

1. 单总线CPU微程序条件判别测试逻辑

通过真值表，在logisim中自动生成电路，如图1.20所示

图示

描述已自动生成

图1.20条件判别电路

1. 单总线CPU微程序控制器设计

将对应的微指令信号填入Excel中，可以得到微程序。

部分微程序如图1.21所示

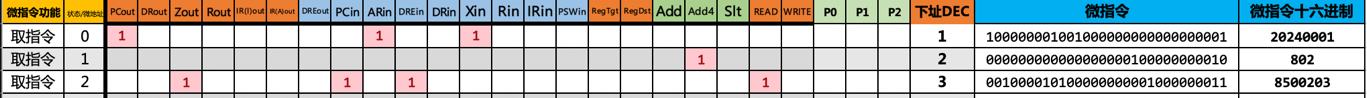


图1.21部分微程序

电路如图1.22所示

图示, 示意图

描述已自动生成

图1.22微程序控制器电路

1. 采用微程序的单总线CPU设计

与定长指令周期相同。

1. 现代时序硬布线控制器状态机设计

部分真值表如图1.23所示，根据真值表在logisim中自动生成电路

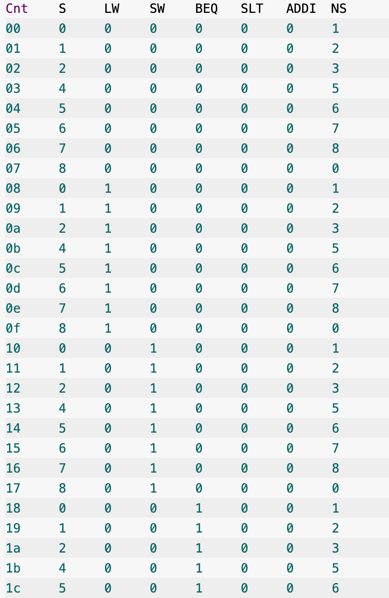


图1.23状态机部分真值表

部分电路如图1.24所示

图示, 示意图

描述已自动生成

图1.24状态机部分电路

1. 现代时序硬布线控制器实现

电路如图1.25所示

图示

描述已自动生成

图1.25 现代时序硬布线控制器电路

### 现代时序中断机制

1. 指令译码逻辑

同现代时序。

1. 支持中断的微程序入口查找逻辑

电路如图1.26所示

图示, 示意图

描述已自动生成

图1.26 中断的微程序入口查找

1. 支持中断的微程序条件判别测试逻辑

电路如图1.27所示

图示, 示意图

描述已自动生成

图1.27支持中断的条件判别电路

1. 支持中断的微程序控制器设计

电路如图1.28所示

图示, 示意图

描述已自动生成

图1.28支持中断的微程序控制器

1. 支持中断的微程序单总线CPU设计

电路如图1.29所示

图示

描述已自动生成

图1.29中断逻辑

1. 支持中断的现代时序硬布线控制器状态机设计

部分电路如图1.30所示

图示, 示意图

描述已自动生成

图1.30 支持中断的现代时序硬布线控制器状态机

1. 支持中断的现代时序硬布线控制器设计

电路如图1.31所示

图示

描述已自动生成

图1.31 支持中断的现代时序硬布线控制器

## 故障与调试

### 支持中断与不支持中断的微程序不相同

图1.32所示为不支持中断，1.33所示为支持中断，可以发现判断逻辑P0，P1，P2的位段并不相同，故使得微程序也不相同，写支持中断的一开始没有看清直接使用了不支持中断的微程序导致测试不通过。

图示

描述已自动生成

图1.32

图示

描述已自动生成

图1.33

## 测试与分析

通过educoder平台进行测试，完成所有预期测试。







# 总结与心得

## 实验总结

本次实验主要完成了如下几点工作：

1. 设计、实现了单周期MIPS CPU硬布线控制器指令译码逻辑；
2. 设计、实现了单周期MIPS CPU硬布线控制器控制逻辑；
3. 设计、实现了单周期MIPS CPU微程序控制器指令译码逻辑；
4. 设计、实现了单周期MIPS CPU微程序控制器逻辑；
5. 完成了单周期MIPS CPU三级时序/现代时序相应功能，并最终能实现冒泡排序。
6. 设计、实现了单周期MIPS CPU支持中断的微程序控制器指令译码逻辑；
7. 设计、实现了单周期MIPS CPU支持中断的微程序控制器逻辑；
8. 完成了单周期MIPS CPU支持中断的现代时序相应功能，并最终能实现冒泡排序。

## 实验心得

1. 加深了对MIPS相关指令的理解，初步掌握了从硬件角度实现指令的步骤。
2. 加深了对课程所学的理解，从硬件角度观察CPU是怎么执行上层给的一条条指令。
3. CPU实验被老师拆分成许多简单的子问题，相比来说，存储实验比CPU还要难，希望存储实验可以跟CPU一样变成不同的子问题，减小实现的难度。
4. 更加熟练logisim的使用，掌握了使用logisim中的分析组合逻辑电路功能自动生成电路，并熟悉多路选择器等硬件。

# 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:** |