***2021***



**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： | CS1908 |
| 学 号： | U201915160 |
| 姓 名： | 陈千鹤 |
| 电 话： | 15319872135 |
| 邮 件： | [1278095698@qq.com](mailto:1278095698@qq.com) |
| 完成日期： | 2021-11-09 |



目 录

[1 CPU设计实验 2](#_Toc88303759)

[1.1 设计要求 2](#_Toc88303760)

[1.2 方案设计 2](#_Toc88303761)

[1.3 实验步骤 10](#_Toc88303762)

[1.4 故障与调试 10](#_Toc88303763)

[1.5 测试与分析 11](#_Toc88303764)

[2 总结与心得 14](#_Toc88303765)

[2.1 实验总结 14](#_Toc88303766)

[2.2 实验心得 14](#_Toc88303767)

[参考文献 15](#_Toc88303768)

# CPU设计实验

## 设计要求

利用logisim平台中现有运算部件构建一个RISC-V单总线CPU，支持5条典型RISC-V指令在单总线CPU上运行，最终CPU能运行内存冒泡排序。

利用logisim平台中现有运算部件实现RISC-V现代时序中断机制，为采用现代时序单总线结构的RISC-V 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，在现代时序微程序控制器的基础上增加硬件数据通路，增加中断返回指令meret的支持，完成中断服务程序的配合。

利用logisim平台中现有运算部件实现RISC-V单总线CPU (变长指令周期3级时序)，利用该时序构造硬布线控制器，支持5条典型RISC-V指令在单总线CPU上运行，最终CPU能运行内存冒泡排序。

设计核心指令见表1.1。

表1.1 核心指令集

|  |  |  |  |
| --- | --- | --- | --- |
| # | 指令 | 汇编代码 | 功能描述 |
| 1 | lw | Lw rd,imm(rs1) | R[rd] ← M[R[rs1] + SignExt(imm)] |
| 2 | sw | sw rs2,imm(rs1) | M[R[rs1] + SignExt(imm)] ← R[rs2] |
| 3 | beq | beq rs1,rs2,imm | if(R[rs1] == R[rs2]) PC ← PC + SignExt(imm) << 1 |
| 4 | slt | slt rd,rs1,rs2 | If (rs1 < rs2) R[rd] ← 1 else R[rd] ← 0 |
| 5 | addi | addi rd,rs1,imm | R[rd] ← R[rs1] + SignExt(imm) |

## 方案设计

### RISC-V指令译码器设计

利用比较器等功能模块将32位RISC-V 指令字译码生成LW、SW、BEQ、SLT、ADDI、OtherInstr等指令译码信号。

指令译码器是控制器核心功能部件，负责将指令字翻译成一根根的指令译码信号，每一根指令译码信号代表一条具体的指令，如图1.1中的I1...Im。

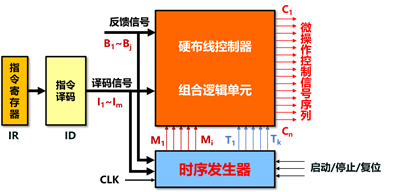


图1.1 指令译码器

具体电路设计见图1.2。

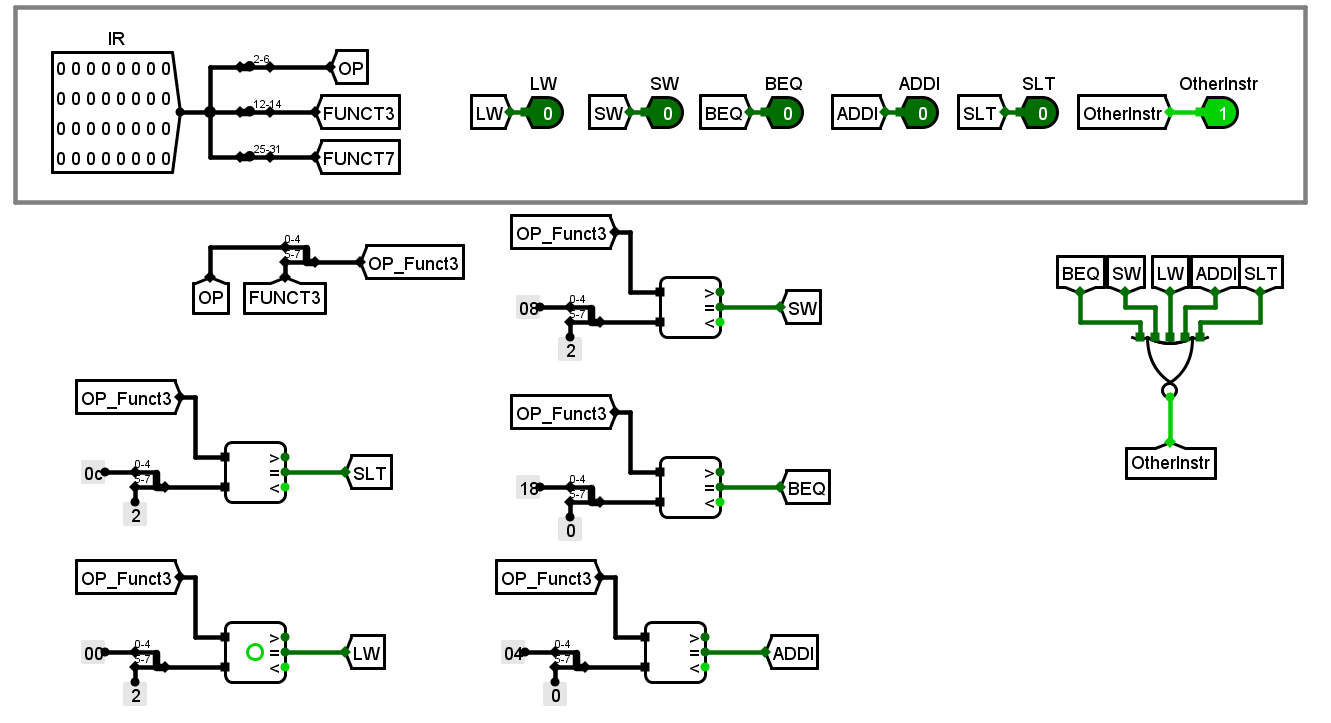


图1.2 RISC-V指令译码器设计

### 单总线CPU微程序入口查找逻辑

微程序控制器中微程序分支的基本原理如图1.3，根据指令译码信号生成5位的微程序入口地址，利用多路选择器选择入口地址。电路设计如图1.4。

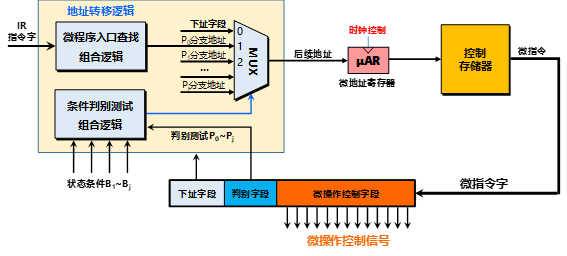


图1.3 微程序控入口查找

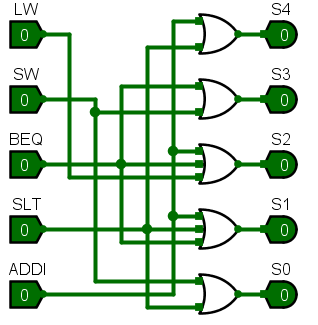


图1.4 微程序地址转移逻辑电路

### 单总线CPU微程序条件判别测试逻辑

根据微指令字中的判别测试字段和条件反馈信息生成后续地址的多路选择信号，实现对应组合逻辑。填写EXCEL表格中的组合逻辑真值表，在logisim中利用分析组合逻辑电路功能自动生成电路。电路设计如图1.5。

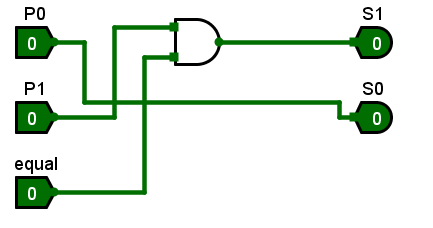


图1.5 条件判别测试逻辑电路

### 单总线CPU微程序控制器设计

微程序控制器中微程序分支的基本原理如图1.3，地址转移逻辑生成入口地址，条件判别逻辑控制下一条命令的选择。将微程序入口查找逻辑，判别测试逻辑，控制存储器等部件进行适当连接，实现微程序控制器的主要数据通路，使微程序加载到控制存储器中。填写EXCEL文件中的微程序自动生成表自动产生微程序设计。电路设计如图1.6。

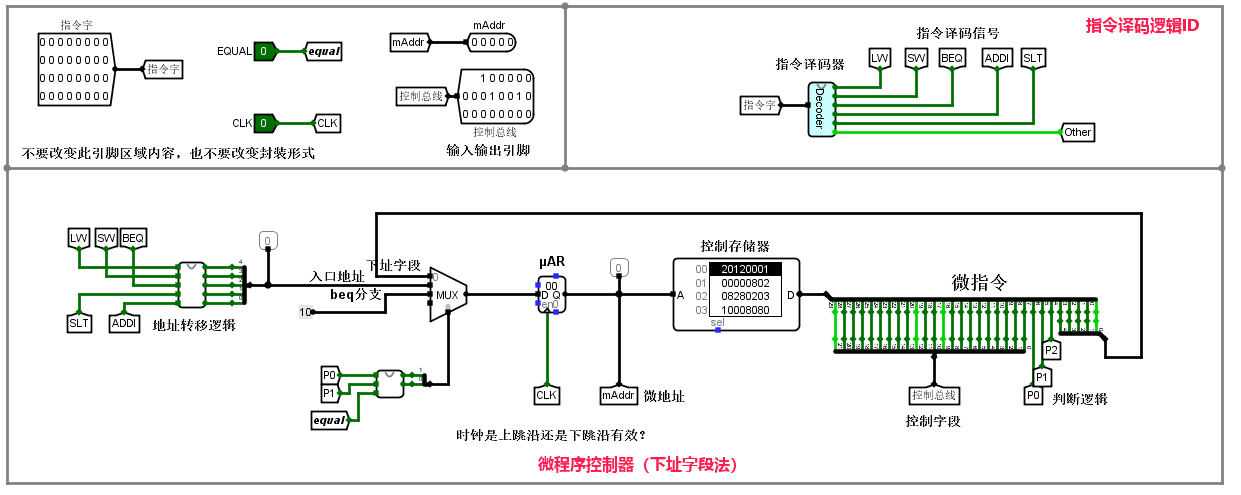


图1.6 微程序控制器电路

### 采用微程序的单总线CPU设计

在完成前面的任务后，在RAM中加载sort-5-riscv.hex程序，进行联调。程序应该运行至0x7c1节拍停下，指令计数为251。电路设计如图1.7。

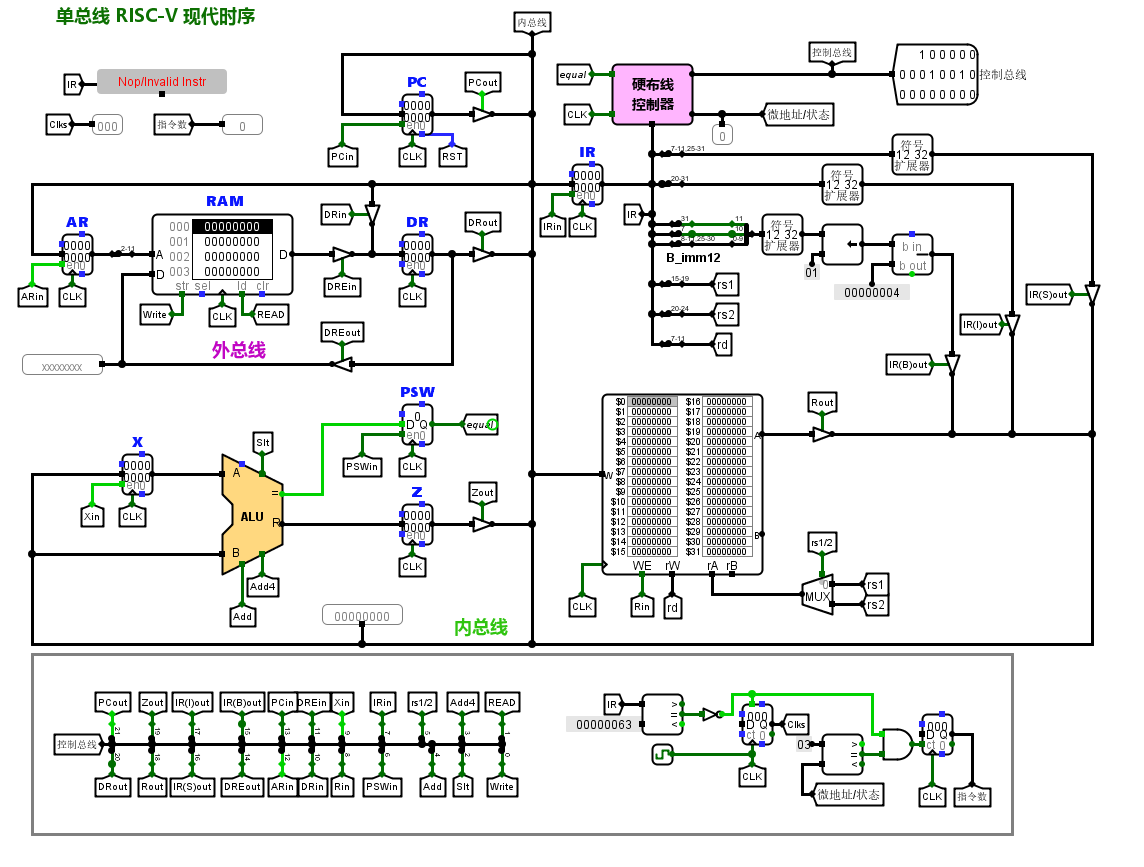


图1.7 单总线CPU(微程序)电路

### 现代时序硬布线控制器状态机设计

状态机如图1.8所示，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，按状态图填写excel表，自动生成次态逻辑表达式后，在logisim中自动生成该电路。状态转换表如图1.9。

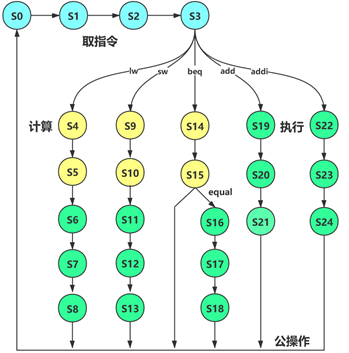
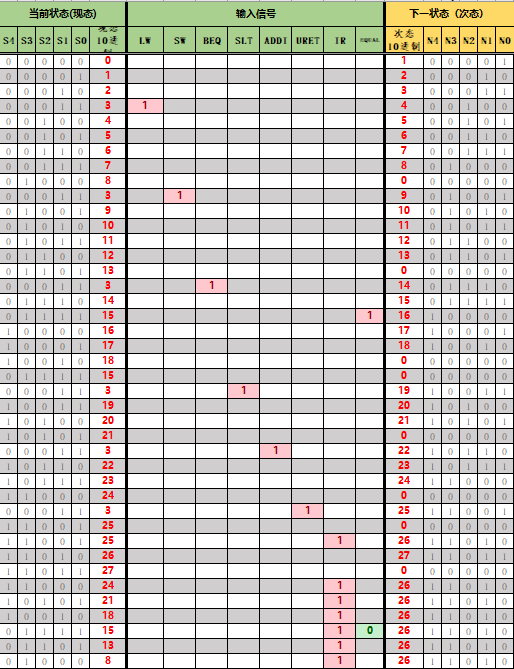
 

图1.8 现代时序硬布线状态机 图1.9 状态转换表

### 现代时序硬布线控制器设计

现代时序系统中硬布线控制器的设计原理图如图1.10，将现态信号输入硬布线控制器，生成微指令控制信号序列。电路设计图如图1.11

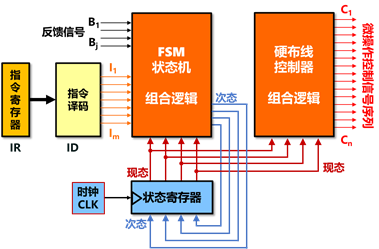
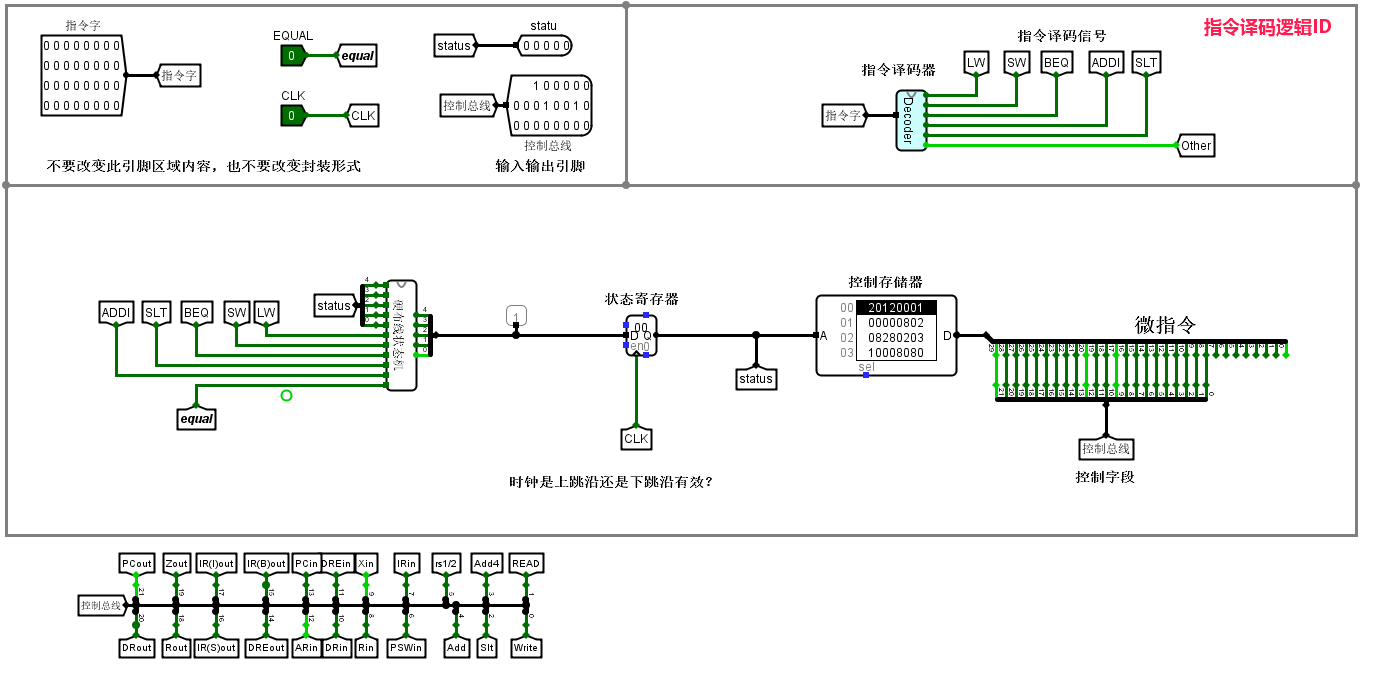
 

图1.10 现代时序系统中硬布线控制器的原理图 图1.11 硬布线状态机电路

### RISC-V现代时序中断机制实现

在实现了RISC-V单总线CPU的基础上，增加了中断机制，因此，也只需要在RISC-V单总线CPU的基础上增加中断指令位，修改指令译码器、微程序入口查找逻辑、微程序条件判断逻辑、为程序控制器、单总线CPU、硬布线控制器状态机和硬布线控制器即可。电路整体设计如图1.12。

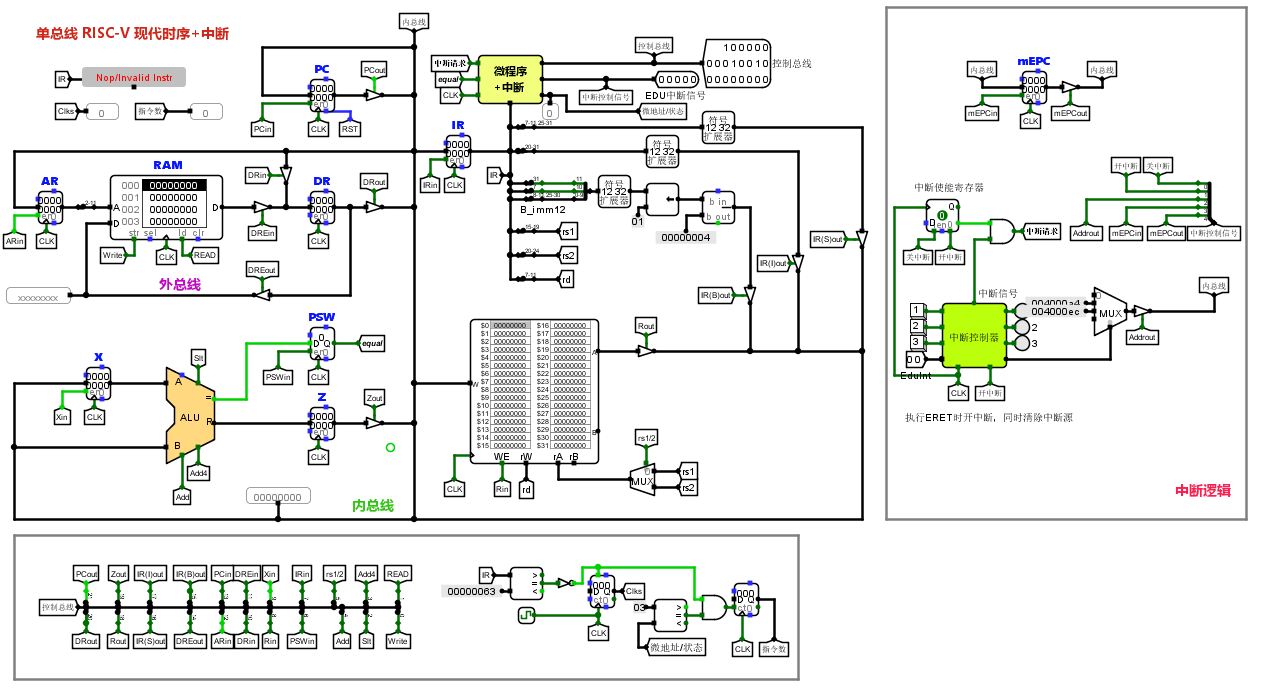


图1.12 单总线 RISC-V 现代时序+中断

### 变长指令周期---时序发生器FSM设计

单总线结构中采用变长指令周期，不同指令机器周期数不同，每个机器周期节拍数也是可变化的，具体状态图如图1.13。按状态图填写相应的excel表，自动生成次态逻辑表达式后，在logisim中自动生成该电路。

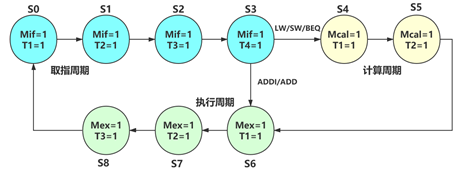


图1.13 变长指令周期时序发生器状态图

### 变长指令周期---时序发生器输出函数设计

时序发生器包括状态机和输出函数两部分，输出函数为组合逻辑，输入为状态寄存器的现态输出，输出为状态周期电位和节拍电位信号，根据时序输出要求，填写excel表，自动生成各输出信号的逻辑表达式，在logisim中自动生成电路。表格填写如图1.14。

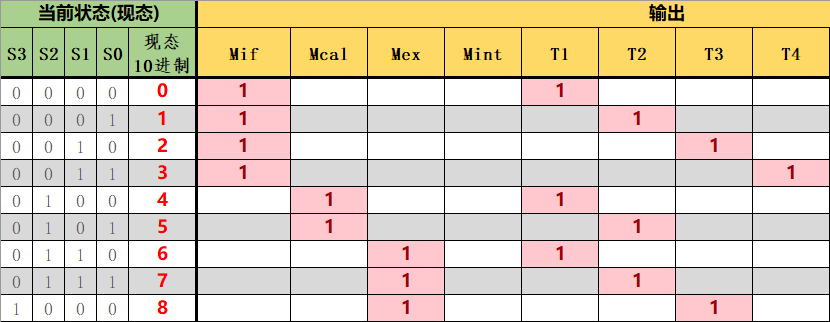


图1.14 时序输出状态表

### 硬布线控制器组合逻辑单元

控制器框架如图1.15，所有微操作控制信号都是反馈信号，指令译码信号，状态周期电位，节拍电位的组合逻辑函数。列出所有微操作信号的产生条件，填写excel表格，自动生成逻辑表达式，然后在Logisim中自动生成电路。输入输出状态表如图1.16。

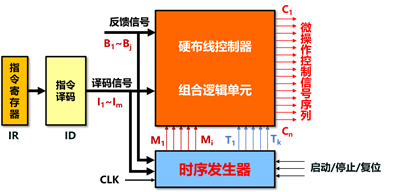


图1.15 控制器架构



图1.16 输入输出状态表

### 变长指令周期---硬布线控制器设计

将时序状态机与输出函数、状态寄存器正确连接，实现时序发生器逻辑。lw、sw、beq、slt、addi和现态输入状态机，输出次态，并且列用输出函数获得状态节拍信号。电路图如图1.17。

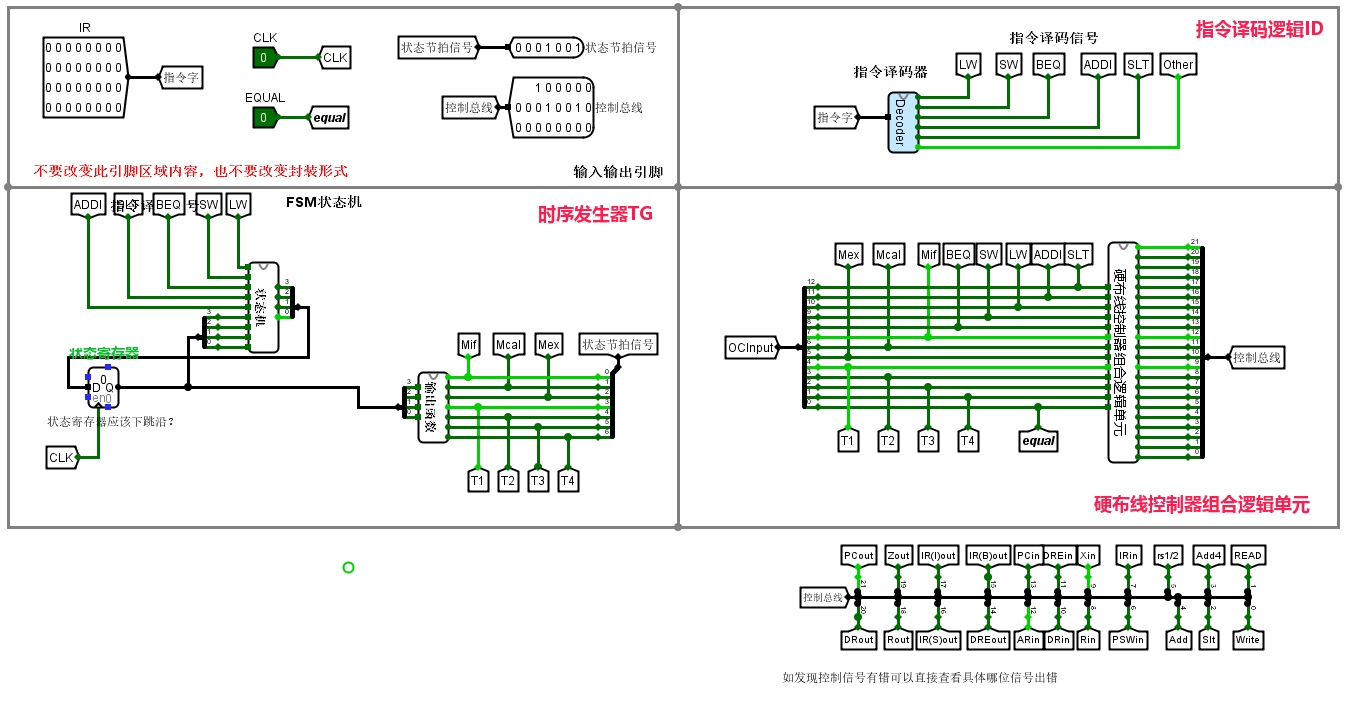


图1.17 硬布线控制器

### 变长指令周期---单总线CPU设计

在完成前面的任务后，在RAM中加载sort-5-riscv.hex程序，ctrl+k自动运行，程序应该运行至0x81d节拍停下，指令计数为251。电路设计如图1.18。

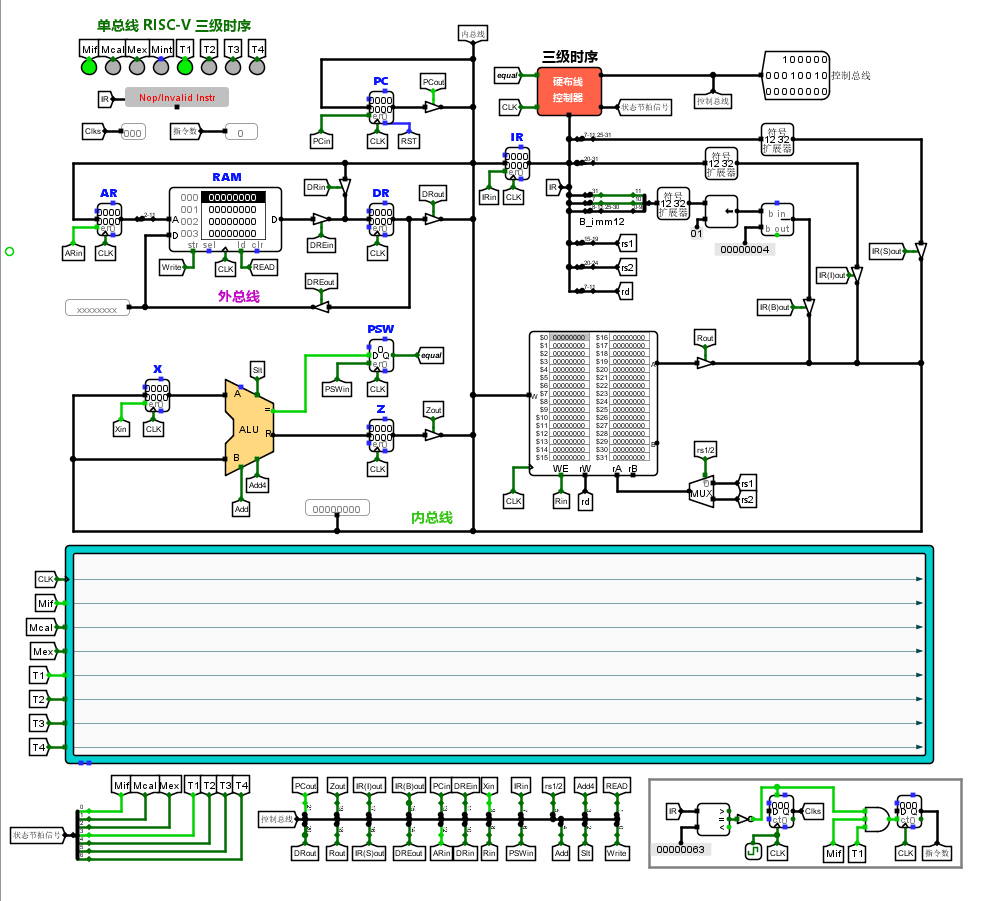


图1.18 单总线CPU(3级时序)电路

## 实验步骤

1. 实现RISC-V指令译码器
2. 实现微程序入口查找逻辑
3. 实现微程序条件判别逻辑测试逻辑
4. 实现微程序控制器
5. 实现单总线CPU
6. 实现硬布线控制器状态机
7. 实现硬布线控制器
8. 实现RISC-V中断机制
9. 实现时序发生器状态机
10. 实现时序发生器输出函数
11. 实现硬布线控制器
12. 实现变长周期单总线CPU

## 故障与调试

### 控制存储器微程序指令问题

**故障现象：**无法正常实现控制存储器中的命令，命令总和与预期值少8。

**原因分析：**RISC-V微程序指令与MIPS不同，每条指令结束时需要有P2标志位，因此如果按照MIPS去实现会导致微指令不正确。

**解决方案：**在每条指令结束的微指令中使P2置1.

### 微程序条件判别逻辑问题

**故障现象：**可以通过测试，但是无法完成联调。

**原因分析：** P0为1时，其他应为任意项，仅该关卡测试时，仅测试了P0为1，其它项全为0的情况，正常通过，但是联调无法正常运行

**解决方案：**重写逻辑判别状态表，P0为1时，其他均为任意项。

### 带中断的CPU设计中中断地址错误

**故障现象：**CPU运行时无法正常完成中断

**原因分析：** 中断地址错误。观察不带中断的程序，程序在00000063停止；再带中断的程序中可以找到改地址的下一个即为中断指令00810113，因此可以发现，中断指令位置在43和61行。

**解决方案：**根据中断指令位置计算中断指令地址为004000a4和004000ec。

## 测试与分析

微程序判别逻辑真值表如表1.2。

表1.2 微程序判别逻辑

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 输入 （填1或0，不填为无关项x） | | | | | 输出 (只填写为1的情况) | | |
| P0 | P1 | P2 | equal | IntR | S2 | S1 | S0 |
| 1 |  |  |  |  |  |  | 1 |
| 0 | 1 | 0 | 1 | 0 |  | 1 |  |
| 0 | 1 | 1 | 1 | 0 |  | 1 |  |
| 0 | 1 | 1 | 0 | 1 |  | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 | 1 |  |  |
| 0 | 0 | 1 | 0 | 1 |  | 1 | 1 |
| 0 | 0 | 1 | 1 | 1 |  | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 | 1 |  |  |
| 0 | 0 | 1 | 1 | 0 | 1 |  |  |
| 0 | 1 | 0 | 0 | 0 | 1 |  |  |
| 0 | 1 | 0 | 0 | 1 | 1 |  |  |
| 0 | 0 | 0 | 0 | 0 |  |  |  |

微指令编码如表1.3。

表1.3 微指令编码

| 微指令功能 | 状态/微地址 | 微指令 | 微指令十六进制 |
| --- | --- | --- | --- |
| 取指令 | 0 | 100000000100100000000000000000 | 20120000 |
| 取指令 | 1 | 000000000000000000100000000000 | 800 |
| 取指令 | 2 | 001000001010000000001000000000 | 8280200 |
| 取指令 | 3 | 010000000000001000000000000100 | 10008004 |
| LW | 4 | 000100000000100000000000000000 | 4020000 |
| LW | 5 | 000010000000000001000000000000 | 2001000 |
| LW | 6 | 001000000100000000000000000000 | 8100000 |
| LW | 7 | 000000000010000000001000000000 | 80200 |
| LW | 8 | 010000000000010000000000000001 | 10010001 |
| SW | 9 | 000100000000100000000000000000 | 4020000 |
| SW | 10 | 000001000000000001000000000000 | 1001000 |
| SW | 11 | 001000000100000000000000000000 | 8100000 |
| SW | 12 | 000100000001000010000000000000 | 4042000 |
| SW | 13 | 000000010000000000000100000001 | 400101 |
| BEQ | 14 | 000100000000100000000000000000 | 4020000 |
| BEQ | 15 | 000100000000000110000000000011 | 4006003 |
| BEQ | 16 | 100000000000100000000000000000 | 20020000 |
| BEQ | 17 | 000000100000000001000000000000 | 801000 |
| BEQ | 18 | 001000001000000000000000000001 | 8200001 |
| SLT | 19 | 000100000000100000000000000000 | 4020000 |
| SLT | 20 | 000100000000000010010000000000 | 4002400 |
| SLT | 21 | 001000000000010000000000000001 | 8010001 |
| ADDI | 22 | 000100000000100000000000000000 | 4020000 |
| ADDI | 23 | 000010000000000001000000000000 | 2001000 |
| ADDI | 24 | 001000000000010000000000000001 | 8010001 |
| ERET | 25 | 000000001000000000000010010001 | 200091 |
| 中断响应1 | 26 | 100000000000000000000001001000 | 20000048 |
| 中断响应2 | 27 | 000000001000000000000000100001 | 200021 |

单总线RISC-V三级时序产生器输出函数真值表如表1.4。

表1.4 单总线RISC-V三级时序产生器输出函数真值表

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 当前状态(现态) | | | | | 输出 | | | | | | | |
| S3 | S2 | S1 | S0 | 现态 10进制 | Mif | Mcal | Mex | Mint | T1 | T2 | T3 | T4 |
| 0 | 0 | 0 | 0 | 0 | 1 |  |  |  | 1 |  |  |  |
| 0 | 0 | 0 | 1 | 1 | 1 |  |  |  |  | 1 |  |  |
| 0 | 0 | 1 | 0 | 2 | 1 |  |  |  |  |  | 1 |  |
| 0 | 0 | 1 | 1 | 3 | 1 |  |  |  |  |  |  | 1 |
| 0 | 1 | 0 | 0 | 4 |  | 1 |  |  | 1 |  |  |  |
| 0 | 1 | 0 | 1 | 5 |  | 1 |  |  |  | 1 |  |  |
| 0 | 1 | 1 | 0 | 6 |  |  | 1 |  | 1 |  |  |  |
| 0 | 1 | 1 | 1 | 7 |  |  | 1 |  |  | 1 |  |  |
| 1 | 0 | 0 | 0 | 8 |  |  | 1 |  |  |  | 1 |  |

# 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. 本次实验设计了RISC-V单总线CPU，实现了指令译码器、入口查找逻辑、条件判别逻辑、硬布线控制器等组件，完成了RISC-V单总线CPU，运行了标准测试程序。
2. 本次实验设计了RISC-V单总线CPU的中断机制，实现了带中断机制的指令译码器、入口查找逻辑、条件判别逻辑、硬布线控制器等组件，完成了RISC-V的中断功能，运行了标准测试程序。
3. 本次实验设计了变长周期三级时序RISC-V单总线CPU，实现了译码器、时序发生器、硬布线控制器等组件，完成了变长周期RISC-V单总线CPU，运行了标准测试程序。

## 实验心得

1. 加深了对于CPU设计的理解，在学习MIPS的基础上，尝试自行学习实现了RISC-V单总线CPU，提升了思考和设计的能力。
2. 学习了RISC-V单总线CPU与MIPS的不同，了解了RISC-V设计的新特点与优势。
3. 建议以后可以自行选择完成方式，自选使用Logisim模拟还是使用Verilog。用Logisim完成仿真实验太费眼睛了，每次做完都会视疲劳很久。

# 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,吴非，肖亮.计算机组成原理.北京:人民邮电出版社，2021年.
4. 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.

|  |
| --- |
| 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: 陈千鹤 C:\Users\Dell\Desktop\ABB6373717E3F47946740879A7F7F538.png** |
| 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） | 课程目标1  工具应用  （10分） | 课程目标2  设计实现  （70分） | 课程目标3  验收与报告  （20分） | 最终评定  （100分） | | 得分 |  |  |  |  | |
| **指导教师签字:** |