***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： | ZY1901 |
| 学 号： | U201914869 |
| 姓 名： | 李宇 |
| 电 话： | 17720338099 |
| 邮 件： | [cavalryly@gmail.com](mailto:cavalryly@gmail.com) |
| 完成日期： | 2021-12-14 |



目 录

[1 CPU设计实验 1](#_Toc90883224)

[1.1 设计要求 1](#_Toc90883225)

[1.2 具体方案 2](#_Toc90883226)

[1.3 实验步骤 11](#_Toc90883227)

[1.4 故障与调试 12](#_Toc90883228)

[1.5 测试与分析 13](#_Toc90883229)

[2 总结与心得 15](#_Toc90883230)

[2.1 实验总结 15](#_Toc90883231)

[2.2 实验心得 15](#_Toc90883232)

[参考文献 17](#_Toc90883233)

# CPU设计实验

## 设计要求

### 单总线RISC-V CPU设计（变长指令周期3级时序）

要求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RISC-V指令在单总线CPU上运行，最终CPU能运行内存冒泡排序。

需要完成的部件有：

* + 指令译码器设计
  + 时序发生器FSM设计
  + 时序发生器输出函数设计
  + 硬布线控制器组合逻辑单元
  + 硬布线控制器设计
  + 单总线CPU设计

### RISC-V现代时序中断机制实现

要求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，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meret的支持，需要中断服务程序配合。

需要完成的部件有：

* + 指令译码器设计
  + 支持中断的微程序入口查找逻辑
  + 支持中断的微程序条件判别测试逻辑
  + 支持中断的微程序控制器设计
  + 支持中断的微程序单总线CPU设计
  + 支持中断的现代时序硬布线控制器状态机设计
  + 支持中断的现代时序硬布线控制器设计

## 具体方案

### 单总线RISC-V CPU设计（变长指令周期3级时序）

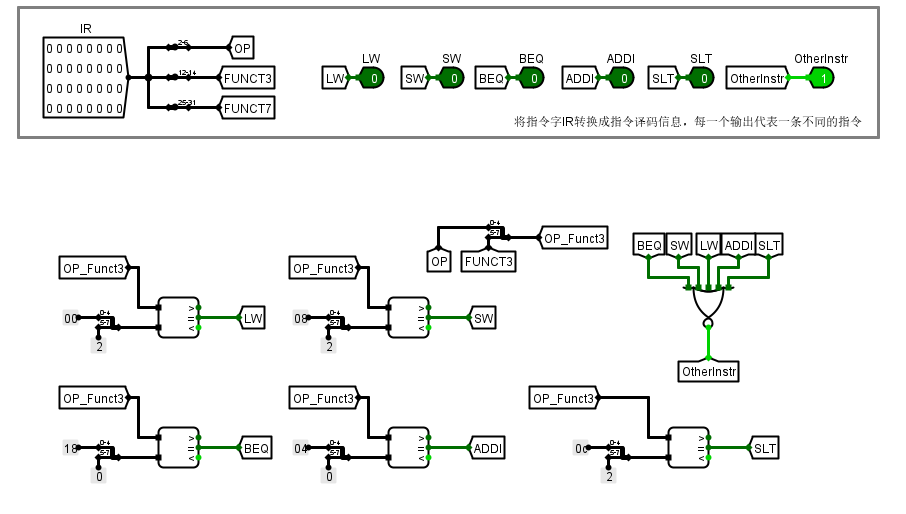
* + 指令译码器设计

设计如**图 1‑1**所示，将输入的32位RISC-V指令字译码成不同的指令译码信号，每条指令由操作码和操作码扩展唯一确定。因此可以利用比较器比较输入指令的指定位和指令操作码以及其扩展来确定是否输出该指令。

指令扩展码如**表 1‑1指令扩展码对应表**所示

|  |  |  |
| --- | --- | --- |
| 指令 | funct3 | opcode |
| LW | 2 | 0 |
| SW | 2 | 0x08 |
| BEQ | 0 | 0x18 |
| SLT | 2 | 0xc |
| ADDI | 0 | 0x4 |

**表 1‑1指令扩展码对应表**



**图 1‑1 指令译码器设计**

* + 时序发生器FSM设计

时序发生器主要包括状态寄存器，状态机组合逻辑，输出函数组合逻辑三部分。其中状态机负责现态与次态的转换，输出函数根据当前状态生成状态周期电位和节拍周期电位。

实验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，通过分析每条指令的周期填写相应excel表即可完成所需要求。

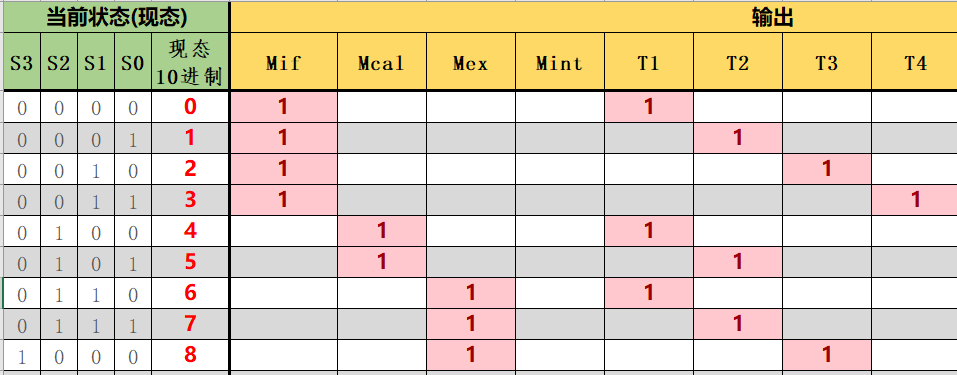
填写的excel如**图 1‑2**所示，在取指周期的最后一个节拍，根据输入信号的不同进行次态的转移。



**图 1‑2 时序FSM状态机组合逻辑设计**

* + 时序发生器输出函数设计

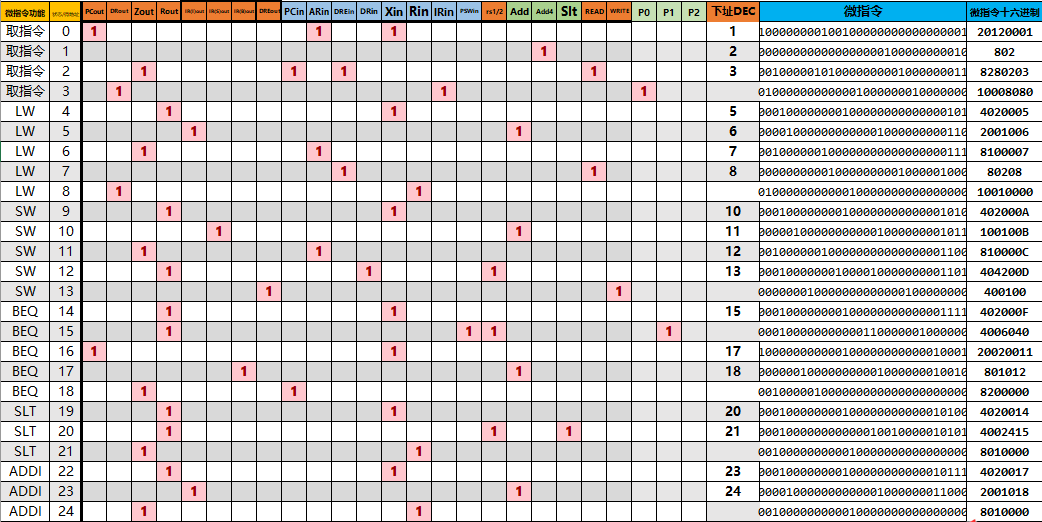
上述部件已经完成了FSM状态机逻辑，要完成完整的时序发生器，还需要完成时序发生器的输出函数设计，根据书上指导以及状态寄存器的现态输入，填写excel如**图 1‑3**所示，得到状态周期点位和节拍电位信号。Mif为取指周期占4个节拍，Mcal为计算周期占2个节拍，Mex为执行周期占3个节拍，不同指令机器周期数不同，其中beq指令较为特殊，可能没有执行周期。



**图 1‑3 时序FSM输出函数设计**

* + 硬布线控制器组合逻辑单元

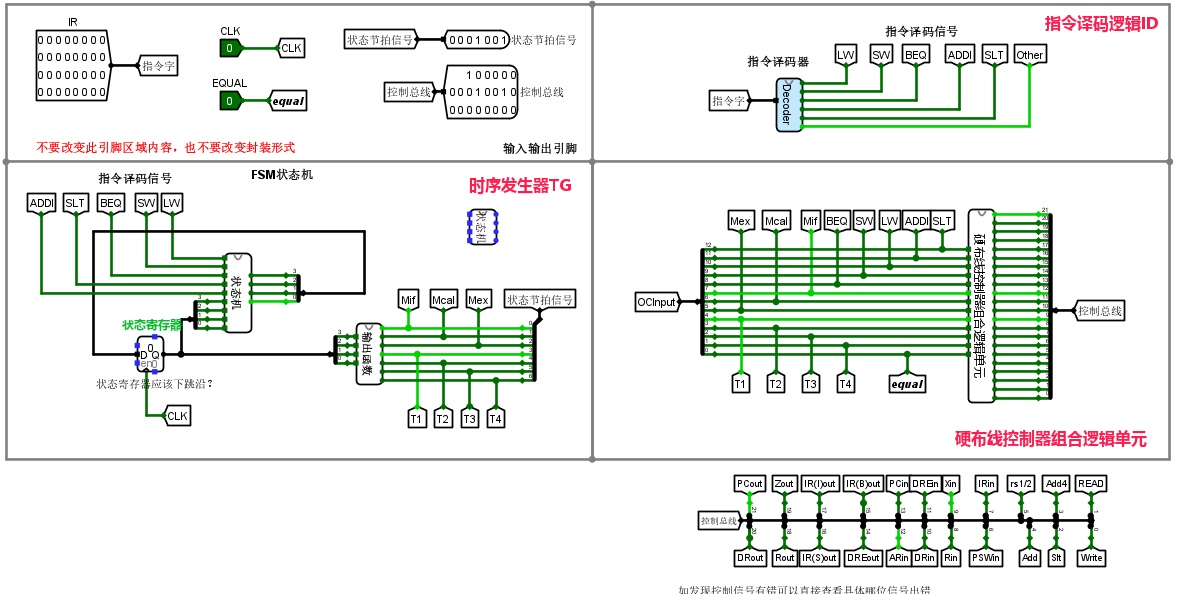
微操作控制信号都是反馈信号，指令译码信号，状态周期电位，节拍电位的组合逻辑函数，根据数据通路列出所有微操作信号的产生条件，并根据其填写excel，如**图 1‑4 硬布线控制器组合逻辑单元**所示。



**图 1‑4 硬布线控制器组合逻辑单元**

* + 硬布线控制器设计

硬布线控制器子电路中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，如**图 1‑5**所示



**图 1‑5 硬布线控制器设计**

至此完成硬布线控制器设计，总结步骤如下：

* 1. 分析指令执行的数据通路，列出每条指令在所有寻址方式下的执行操作流程和每 一步所需要的控制信号。
  2. 对指令的操作流程进行细化，将每条指令的每个微操作分配到具体机器周期的具 体时间节拍信号上，即对操作控制信号进行同步控制。
  3. 根据控制信号同步控制方式构造合适的时序发生器。
  4. 对每一个控制信号进行逻辑综合，得到每个控制信号的逻辑表达式。在对控制信号 进行逻辑综合的过程中，要考虑每一个控制信号在不同指令、不同机器周期和不同节拍的 有效情况，不能遗漏，否则对应的指令将由于缺少控制信号而不能正确执行。
  5. 最后采用逻辑门或 PLA 或 ROM 实现逻辑表达式的功能。
  + 单总线CPU设计

正确实现上述部件的功能后，在设计好的CPU进行最终联调，测试排序程序能否正常运行，仅需在ram中加载编写好的数据映像，然后ctrl+k运行即可。

### 单总线RISC-V CPU设计（现代时序带中断操作）

* + 指令译码器设计

由于都采用相同的指令集，译码器的设计并没有变化，同1.2.1中设计相同。

* + 支持中断的微程序入口查找逻辑

根据指令译码信号生成5位的微程序入口地址，相较之前微程序入口查找逻辑，需要增加eret对应微程序入口查找逻辑，填写excel如**图 1‑6 支持中断的微程序入口查找逻辑**所示，即增加eret指令的微程序入口地址即可，注意机器指令译码信号的表项应与logisim中分析组合电路自动生成的输入项保持一致，不能有多余的字母或表示。

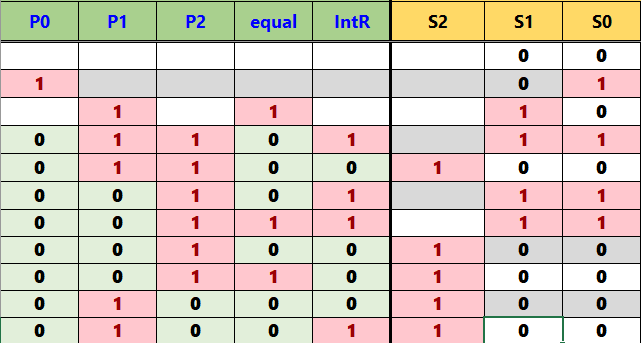


**图 1‑6 支持中断的微程序入口查找逻辑**

* + 支持中断的微程序条件判别测试逻辑

要求根据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，并根据其生成组合逻辑电路

实验中我们需给出各个判别条件下应该选择的下址，即地址转移控制信号，如S=0是下一条地址，默认为当前地址+1，S=1代表取下一条指令入口地址，S=2是beq指令的equal条件成立的情况下的跳转地址，S=3则是中断处理程序的地址，S=4表示回到取指令阶段，其地址。注意S=2和S=3情况下的地址即1.2.2微程序入口转移程序的对应值。根据书上指导填写excel如**图 1‑7**所示。

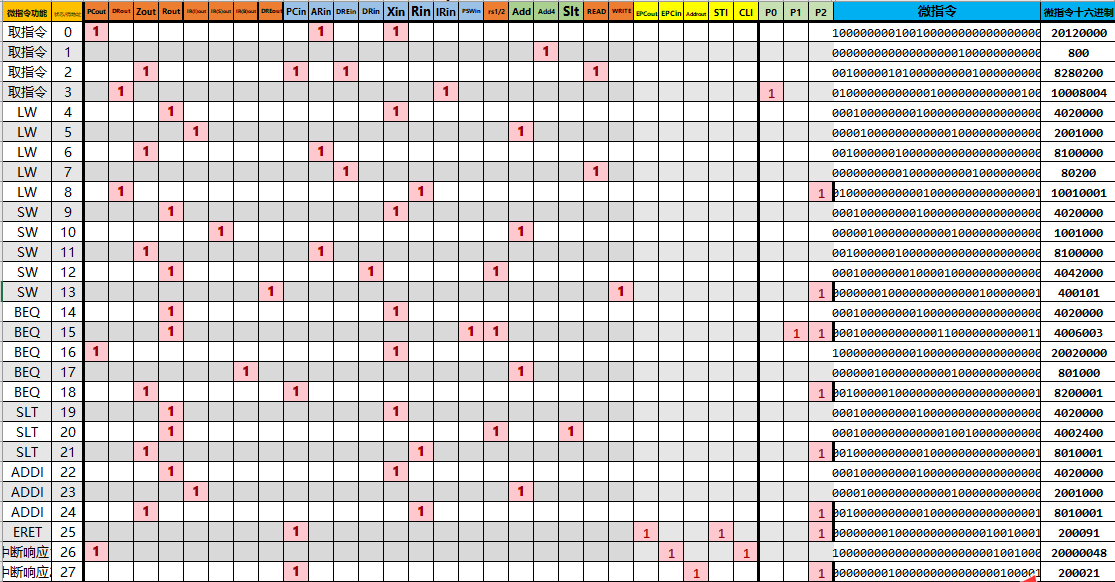


**图 1‑7 支持中断的微程序条件判别测试逻辑**

* + 支持中断的微程序控制器设计

完善实验中给的的微程序框架，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，设计微程序并加载到控制存储器中。

微程序地址对应硬布线的时序，可对照硬布线控制信号输出表填写控制信号，生成相应指令字，注意除了增加的指令，之前填写好的指令字也有所变化，不能直接拷贝过去。填写excel表如**图 1‑8**所示。



**图 1‑8支持中断的微程序控制器设计**

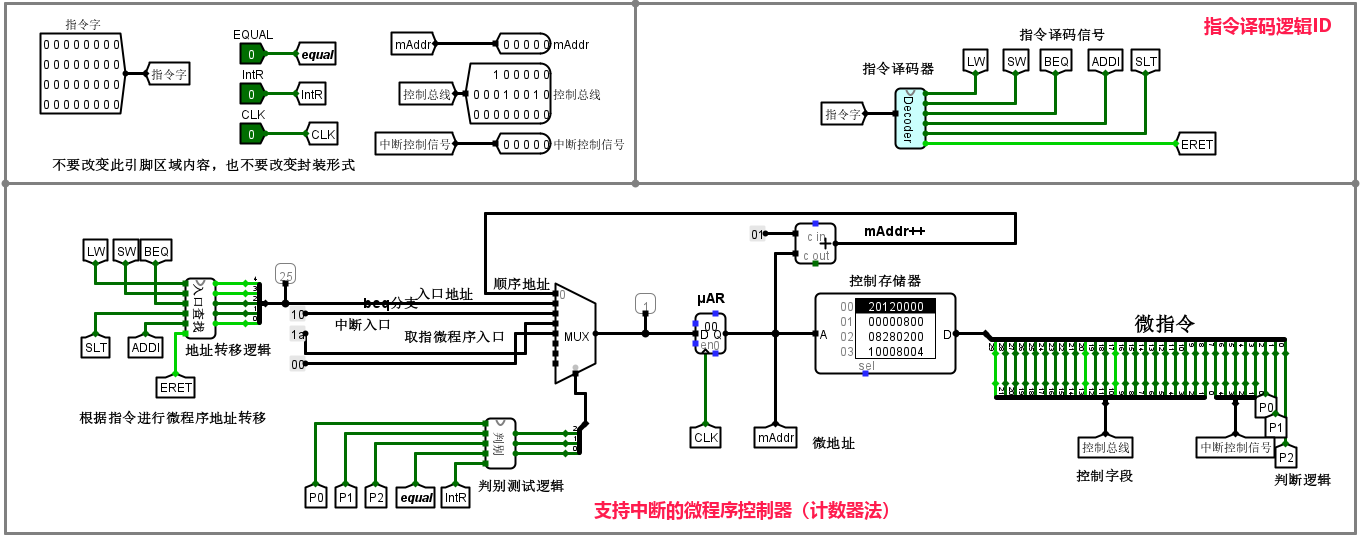
P0为判别测试位，为1表示要根据指令功能进行微程序分支，P1为判别测试位，为1表示要根据equal标志进行微程序分支，P2为判别测试位，为1表示是微程序的最后一条微指令，可能需要进行中断响应。

各分支入口地址对应关系说明如**表 1‑2**所示

|  |  |
| --- | --- |
| 类型 | 值 |
| 顺序地址 | 当前指令地址+1 |
| 入口地址 | 地址转移逻辑部分给出 |
| 中断入口地址 | 26 |
| beq分支地址 | 16 |
| 取值微程序入口地址 | 0 |

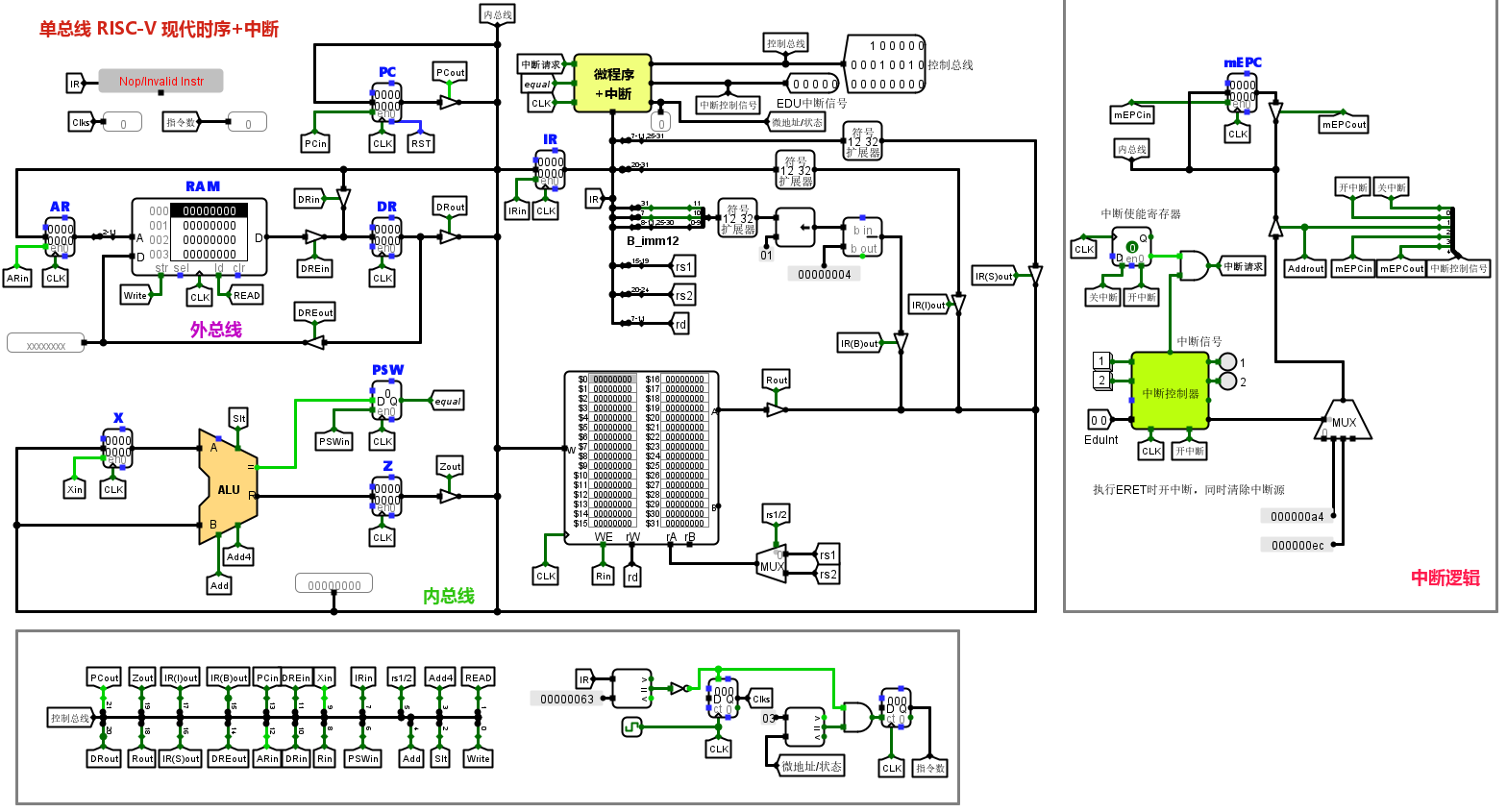
**表 1‑2 分支入口地址对应表**

在ram中加载好数据映像后，根据上述支持中断的微条件判别测试逻辑连接电路，如**图 1‑9**所示。



**图 1‑9 支持中断的微程序控制器设计**

* + 支持中断的微程序单总线CPU设计



**图 1‑10支持中断的微程序单总线CPU设计**

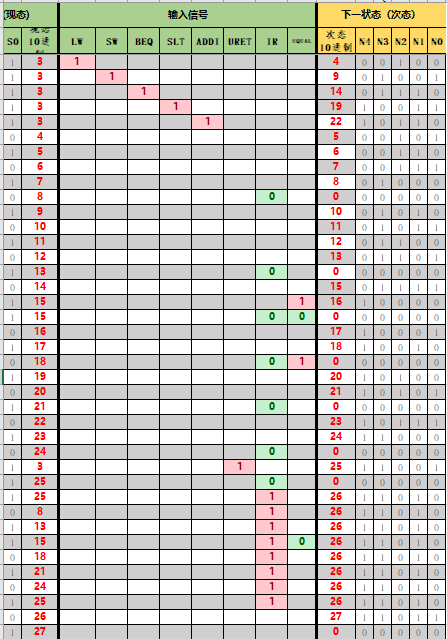
如**图 1‑10**所示，在同1.2.1的联调实验中一样加载数据映像后，还需完成中断所需的硬件逻辑，主要包括异常程序地址计数器mEPC，中断使能寄存器IE，中断控制器等模块。不同按键的中断服务程序入口地址可以利用RARS汇编器汇编源程序查看lable地址得到。需要注意的是：因为响应中断的前提是开中断，因此中断请求信号是中断信号与寄存器取非端进行与操作的结果

其中IE寄存器在关中断时异步置1，开中断时异步置0，输出为寄存器保存内容相反部分。

mEPC寄存器通过mEPCin进行输入信号的控制，通过三态门进行输出信号的控制，只有在mEPCout信号有效时才进行输出。

* + 支持中断的现代时序硬布线控制器状态机设计

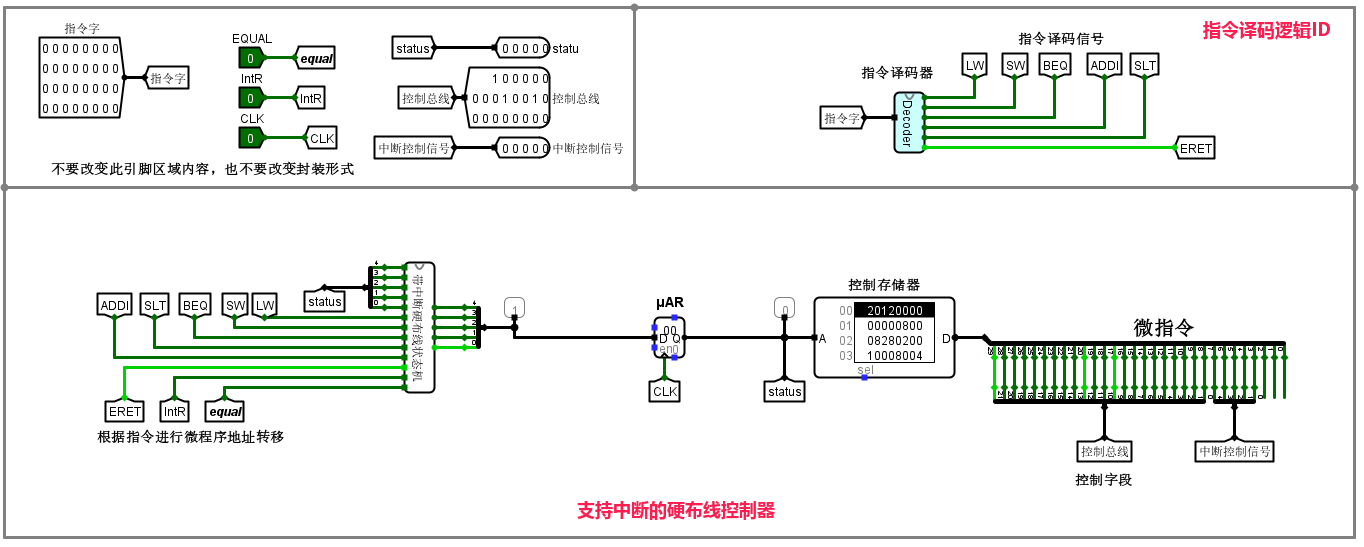
根据状态转移图填写excel表如**图 1‑11**所示。



**图 1‑11支持中断的现代时序硬布线控制器状态机设计**

* + 支持中断的现代时序硬布线控制器设计

在实现指令译码、现代时序状态机模块后，最终实现硬布线控制器的集成，完成硬布线控制器框架连接，加载数据映像如**图 1‑12**所示，注意硬布线控制器组合逻辑不需要实现直接采用微程序控制器的控制存储器代替即可，完成测试后用硬布线控制器替换cpu中的微程序控制器进行程序测试，因为硬布线的状态是与微程序的地址一一对应的。



**图 1‑12 支持中断的现代时序硬布线控制器设计**

## 实验步骤

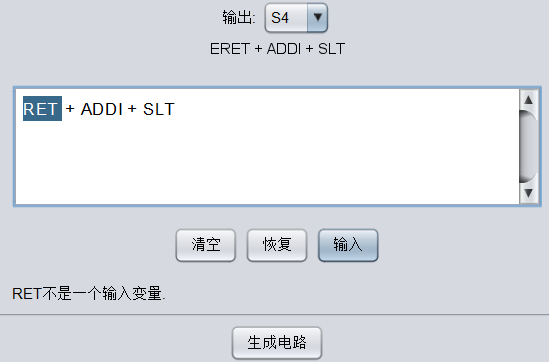
每个实验的流程大致如下：

1. 分析表格/电路
2. 填写表格中相应表项
3. 自动生成逻辑表达式
4. 在logisim中利用分析组合逻辑电路功能自动生成电路，或分析后自己手动连接电路
5. 进行联调测试。

## 故障与调试

### 表达式生成问题

**故障现象：**在微程序入口查找逻辑+中断中，利用logisim生成组合逻辑电路时出错，无法生成，出错现象如**图 1‑13 logisim故障部分截图**所示。



**图 1‑13 logisim故障部分截图**

**原因分析：**excel表中无eret项，而是ret项，生成的表达式含有不是输入变量的ret，logisim无法根据其生成组合逻辑电路。

**解决方案：**将excel表中ret改成eret，正确生成表达式即可。类似问题还有填写3号表时，按书上指导和实验要求更正excel表格即可。

### 微程序法填写表格时出错

**故障现象：**状态没有正常转移，educoder上评测不通过。

**原因分析：**填写3号excel表有误。

**解决方案：**分析评测集样例，发现是beq的判别控制字段有误，进行更正后正确实现状态转移。

### 现代时序中微程序条件判别逻辑出错

**故障现象：**判别逻辑出现不存在的状态，如6和7。

**原因分析：**填写4号excel表有误。

**解决方案：**错误的化简了输入，更正excel表后正确。

### 带中断的现代时序联调冒泡排序时候出错

**故障现象：**正确连接电路后。点击按键相应位置没有变化。

**原因分析：**加载了错误的数据映像。

**解决方案：**更正错误的数据映像，完成逻辑。

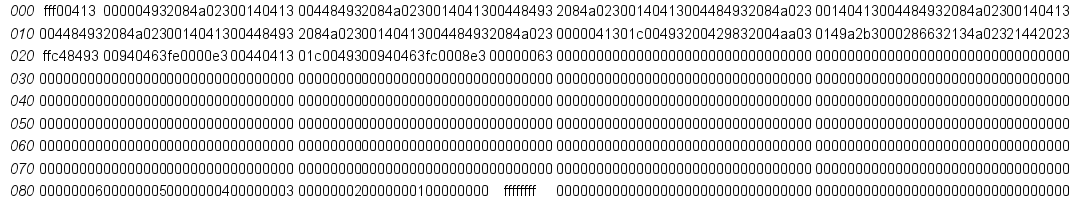
## 测试与分析

### Educoder测试

已通过educoder上全部测试样例。

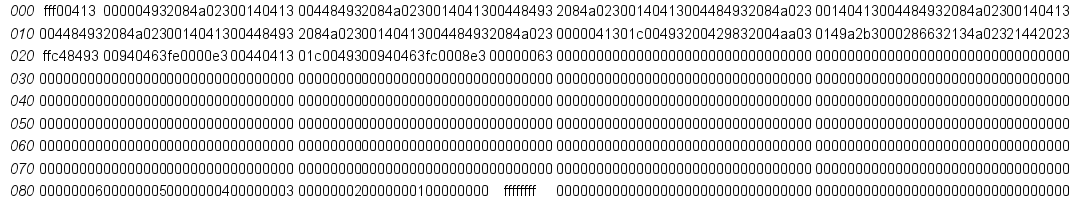
### Logisim测试

三级时序冒泡程序测试如**图 1‑14**所示，最终在执行0x81d个clks，251条指令后停下，最后一条指令为beq指令。



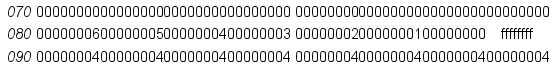
**图 1‑14三级时序冒泡程序测试**

带中断的的现代时序冒泡程序测试如**图 1‑15**所示，最终在执行0x1992个clks，252条指令后停下，最后一条指令为beq指令。



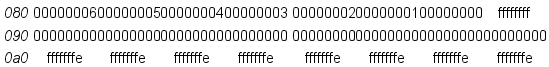
**图 1‑15带中断的现代时序冒泡程序测试**

测试带中断的现代时序CPU终端功能，按下四次按键1后，测试结果如**图 1‑16**所示



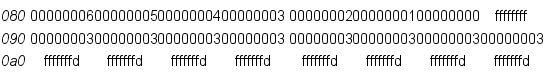
**图 1‑16 按下四下按键1后RAM中部分截图**

按下两次按键2后，测试结果如**图 1‑17**所示



**图 1‑17 按下两下按键2后RAM中部分截图**

按下三次按键1再按下三次按键2，测试结果如**图 1‑18 按下三下按键1再按下三下按键2后RAM中部分截图**所示。



**图 1‑18 按下三下按键1再按下三下按键2后RAM中部分截图**

# 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. 完成方案总结：

* 在单周期RISC-V CPU设计（变长指令周期3级时序）中，根据变长三级时序设计、构造硬布线控制器，实现支持五条典型RISC-V指令在单总线CPU上运行，完成后最终CPU能运行内存冒泡程序。
* 在单周期RISC-V CPU设计（现代时序）中，根据该时序设计，构造硬布线控制器，实现支持5条典型RISC-V指令在单总线CPU上运行，完成后最终CPU能运行内存冒泡排序。
* 在RISC-V现代时序中断机制实现实验中，为采用现代时序的单总线结构RISC-V CPU设计增加中断处理机制，增加了数据通路和指令支持，正确实现多个外部按键中断事件的随机处理，完成了现代时序CPU中断功能的增加。

1. 功能总结：

* 实现了32位RISC-V指令字定长指令的译码和指令的解析。
* 设计了时序发生器（FSM、组合逻辑）
* 设计了微程序控制器
* 设计了硬布线控制器
* 实现了微程序转移逻辑
* 完成了支持中断的现代时序单总线CPU
* 完成了三级时序单总线CPU

## 实验心得

心得与收获：

* 熟练使用上学期数电教学中学会的有关logisim的表达式，真值表等组合逻辑电路生成有关功能，并复习了和新掌握了有关元器件，更加深刻的了解了logisim这个软件，为将来可能进行的分析工作打下牢固基础。
* 在硬件知识上，熟悉并掌握了单总线CPU不同方案下的设计架构，并自己动手设计实现了部分功能，最终以冒泡排序程序的联调来验证CPU的可工作性，既提升了动手能力也收获了知识。
* 在软件知识上，熟悉并掌握了RISCV五条典型指令以及中断指令的编码，控制信号的产生，以及执行指令时数据通路的变化，实验不仅让我清晰的看到CPU是如何执行指令的，也让我学会如何去分析这些指令，从而更好地进行设计和调整。
* 光说不做假把式，光停留在课本中是远远不够的，通过实验中自己动手去设计实现CPU，不仅倒逼自己去更认真，更深刻的理解课本，同时也在不断踩坑和调试过程中对CPU这一计算机关键构成部件有了更深刻的理解明白道CPU内部各个运算器件如何协同工作来完成CPU的整个功能，不得不说这次实验和教材结合的相当好，而脱离教材所学的实验设计即使再好，也很难说是高明的。在此我还要感谢老师和同学们在学习中对我的帮助，特别是我的任课老师谭志虎老师在组原群里孜孜不倦的回答整个年级的问题，令人敬佩。

建议：

* 实验通过填写excel表格来生成表达式，通过logisim生成电路功能来完成复杂组合逻辑的设计而不用学生自己手连的设计是相当高明的，希望能够发扬光大。
* 实验在educoder平台上进行，但因学校先前是在MIP32架构下进行实验，因此educoder平台上很多内容都没有进行更新，例如实验介绍中的测试集和实验中实际的测试集也有不一致之处，实验时可能会给学生带来不必要的时间浪费，建议尽快更正。
* educoder平台上测试可以面向测试集调试，可以考虑适当隐藏测试集来进行学生能力的区分。

# 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:** |
| 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:** |