***2024***



**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： | CS2207 |
| 学 号： | U202215576 |
| 姓 名： | 王嘉成 |
| 电 话： | 19307087726 |
| 邮 件： | [1502969644@qq.com](mailto:13456@qq.com) |
| 完成日期： | 2024-06-14 |



# CPU设计实验

## 设计要求

利用logisim平台中现有部件构建一个支持中断的CPU，可支持LW，SW，BEQ，ADDI，SLT，ERET等MIPS指令集，支持中断机制，该CPU能按照一定周期自动读取存放于RAM中的指令并实现功能，以及运行包含上述指令的简单程序，具体指令功能见下表。

|  |  |  |
| --- | --- | --- |
| **#** | **指令** | **指令功能 (RTL描述)** |
| **1** | **lw rt,imm(rs)** | **R[rt] ← M[R[rs] + SignExt(imm)]** |
| **2** | **sw rt,imm(rs)** | **M[R[rs] + SignExt(imm)]← R[rt]** |
| **3** | **beq rs,rt,imm** | **if(R[rs]==R[rt]) PC←PC+4+SignExt(imm)<< 2** |
| **4** | **addi rt,rs,imm** | **R[rt] ← R[rs] + SignExt(imm)** |
| **5** | **add rd,rs,rt** | **R[rd] ← R[rs] + R[rt]** |

表 1 MIPS指令集及相应功能

## 方案设计

### MIPS指令译码器设计

查找MIPS指令码表可知，各个指令的OP字段和FUNCT字段如下表所示，只需要将输入的IR操作码分离出OP字段和FUNCT字段，并与相应常量进行比较即可确认输入的指令,再将各个信号通过或非门汇总给OtherInstr。

|  |  |  |
| --- | --- | --- |
| 指令 | OP | FUNCT |
| ADDI | 0x08 | 无 |
| LW | 0x23 | 无 |
| SW | 0x2b | 无 |
| BEQ | 0x04 | 无 |
| SLT | 0x00 | 0x2a |

表 2 指令码与OP及FUNCT字段对应关系

最终得到电路图如下图所示。

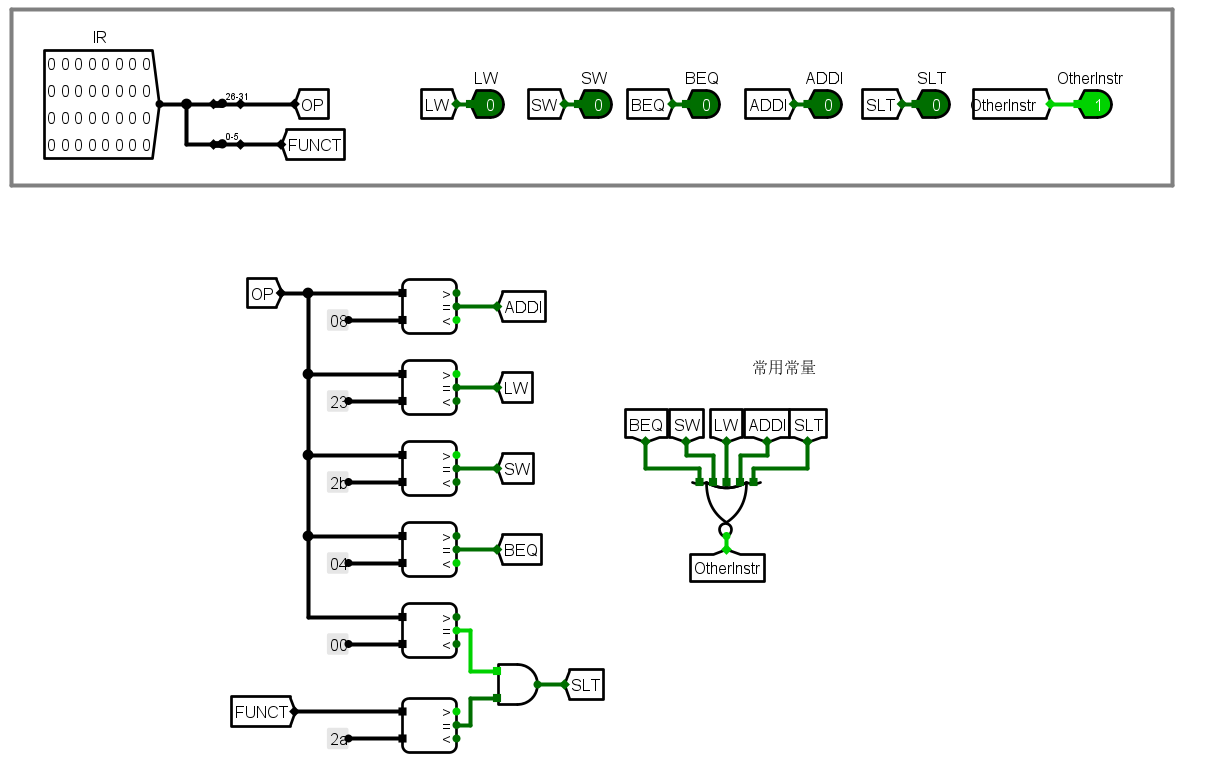


图 1 指令译码器电路图

### 支持中断的微程序入口查找逻辑

根据实验提示下载6号EXCEL表格，并根据电路测试结果填写好其中的微程序入口地址表格，然后将自动生成的逻辑表达式输入Logisim中自动生成电路。



图 2 微程序地址入口表



图 3 微程序入口查找自动生成表达式

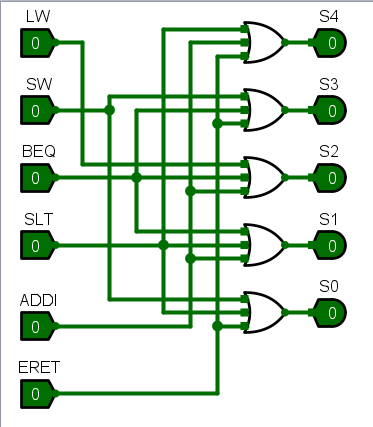


图 4 入口查找逻辑电路图

### 支持中断的微程序条件判别测试逻辑

根据实验提示下载4号EXCEL表格，并根据电路测试结果填写好其中的组合逻辑真值表，然后将自动生成的逻辑表达式输入Logisim中自动生成电路。



图 5 组合逻辑真值表

最终表达式为

S2=~P0&P1&P2&~equal&~IntR+~P0&~P1&P2&~equal&~IntR+~P0&~P1&P2&equal&~IntR+~P0&P1&~P2&~equal&~IntR+~P0&P1&~P2&~equal&IntR

S1=~P0&P1&~P2&equal&~IntR+~P0&P1&P2&equal&~IntR+~P0&P1&P2&equal&IntR+~P0&P1&P2&~equal&IntR+~P0&~P1&P2&~equal&IntR+~P0&~P1&P2&equal&IntR

S0=P0&~P1&~P2&~equal&~IntR+P0&~P1&~P2&~equal&IntR+P0&~P1&~P2&equal&~IntR+P0&~P1&~P2&equal&IntR+~P0&P1&P2&~equal&IntR+~P0&~P1&P2&~equal&IntR+~P0&~P1&P2&equal&IntR

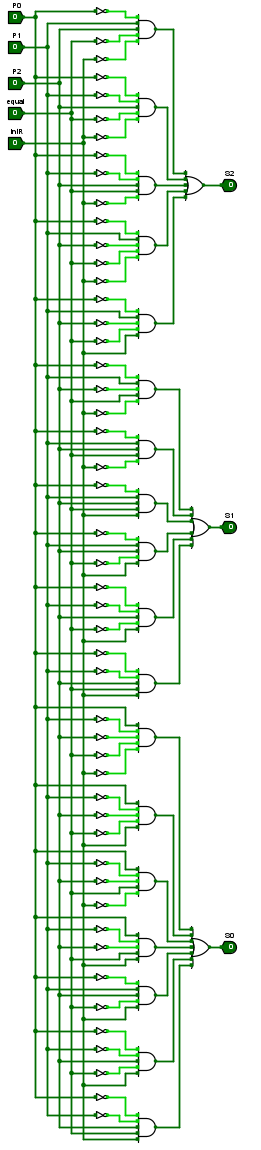


图 6 条件判断测试逻辑电路图

### 支持中断的微程序控制器设计

根据实验提示下载6号EXCEL表格，并根据微程序每个节拍所谓产生的信号填写好其中的微程序自动生成表，然后将自动生成的十六进制微指令输入Logisim中的只读存储器中。

在此基础上，为了实现中断，引入了P2表示表明微程序即将结束。因此需要在=将每条指令的最后一条微指令的P2置为1。另外，BEQ指令的第二条微指令为了实现分支跳转，也需要将P2置为1。ERET中断指令1，2按照教材讲解填写即可。

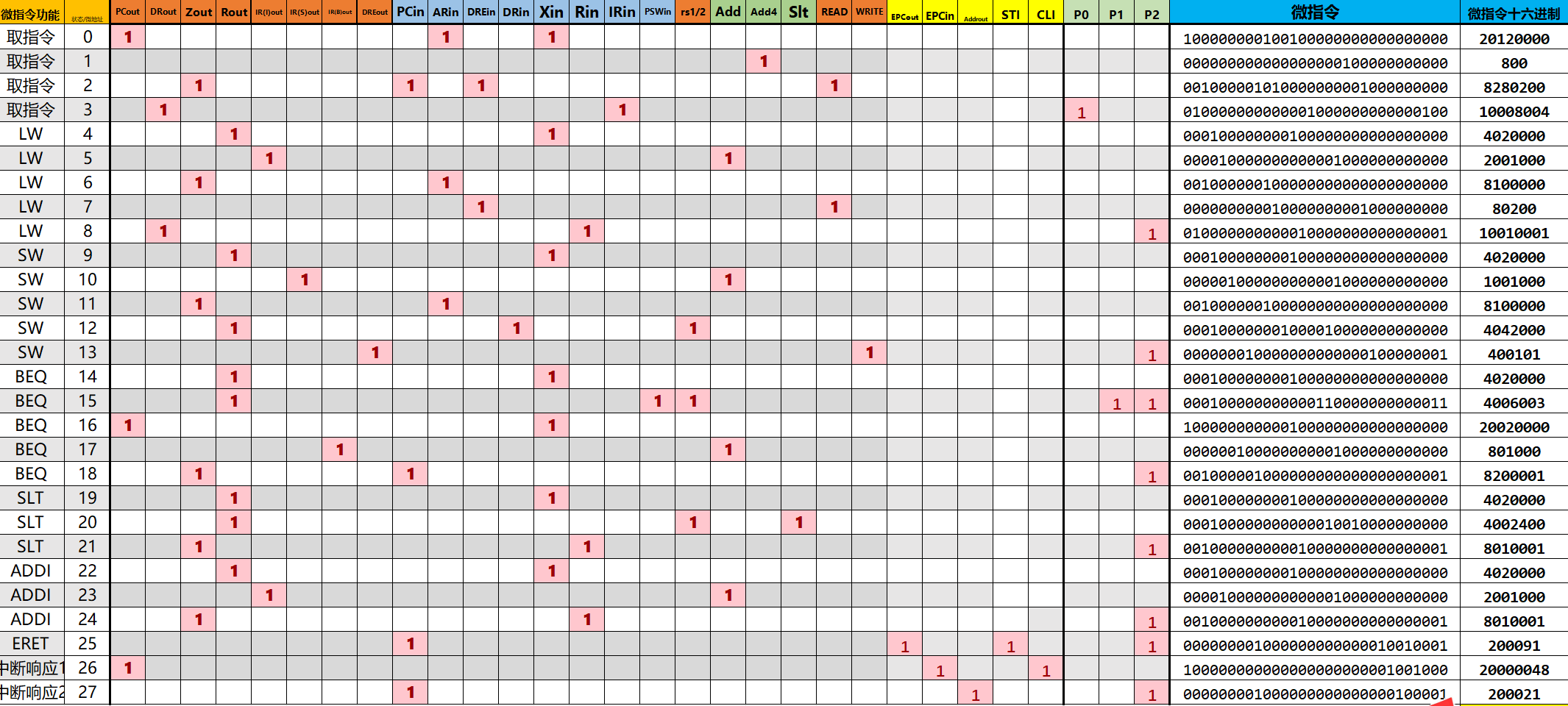


图 7 微程序自动生成表

然后在Logisim中对电路进行连线，将条件判别测试逻辑作为多路选择的条件接入选择端，mAddr作为顺序地址，入口查找逻辑作为入口地址，常量0x10作为beq分支，0x1a作为中断响应入口，以及要将控制存储器里的内容改为之前得到的十六进制微指令，最后需要将时钟寄存器改为下降沿有效，才能对上周期。

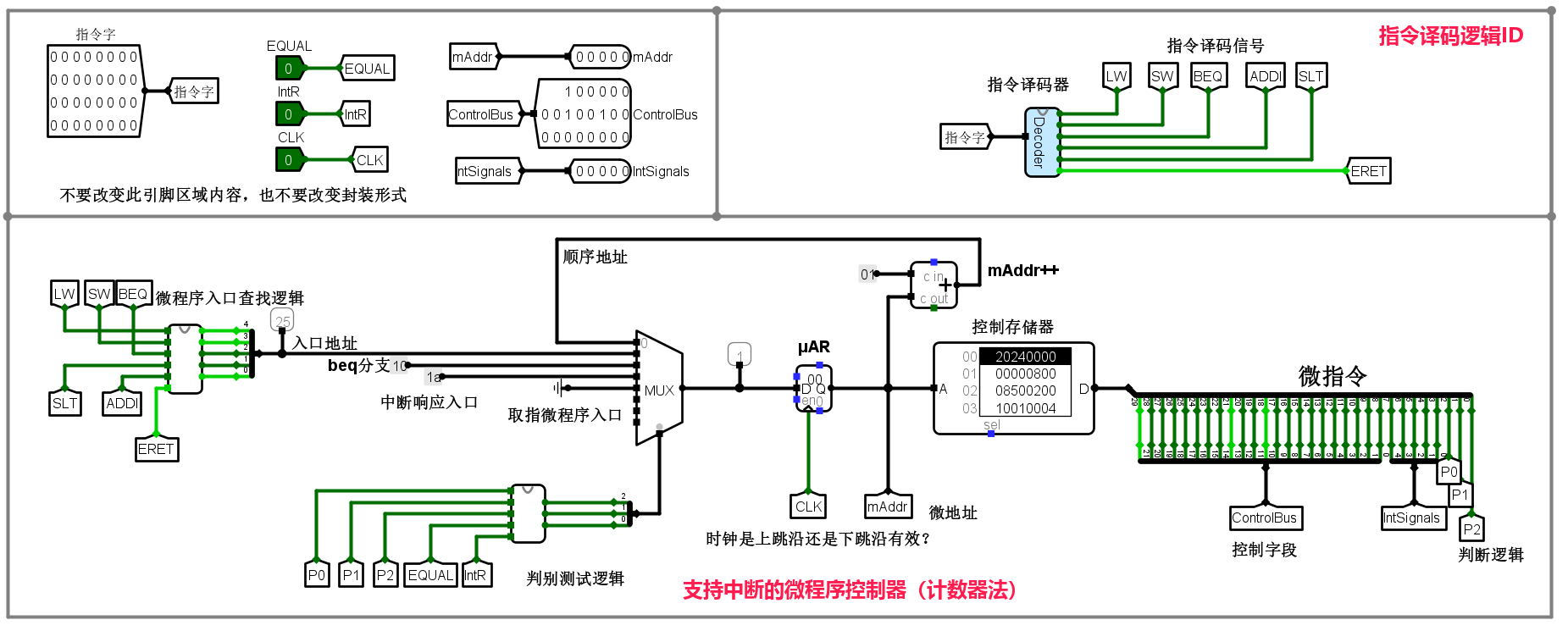


图 8 微程序控制器电路图

### 支持中断的现代时序硬布线控制器状态机设计

根据实验提示下载5号EXCEL表格，并根据电路测试结果填写好其中的状态转换表，然后将自动生成的逻辑表达式输入Logisim中自动生成电路。

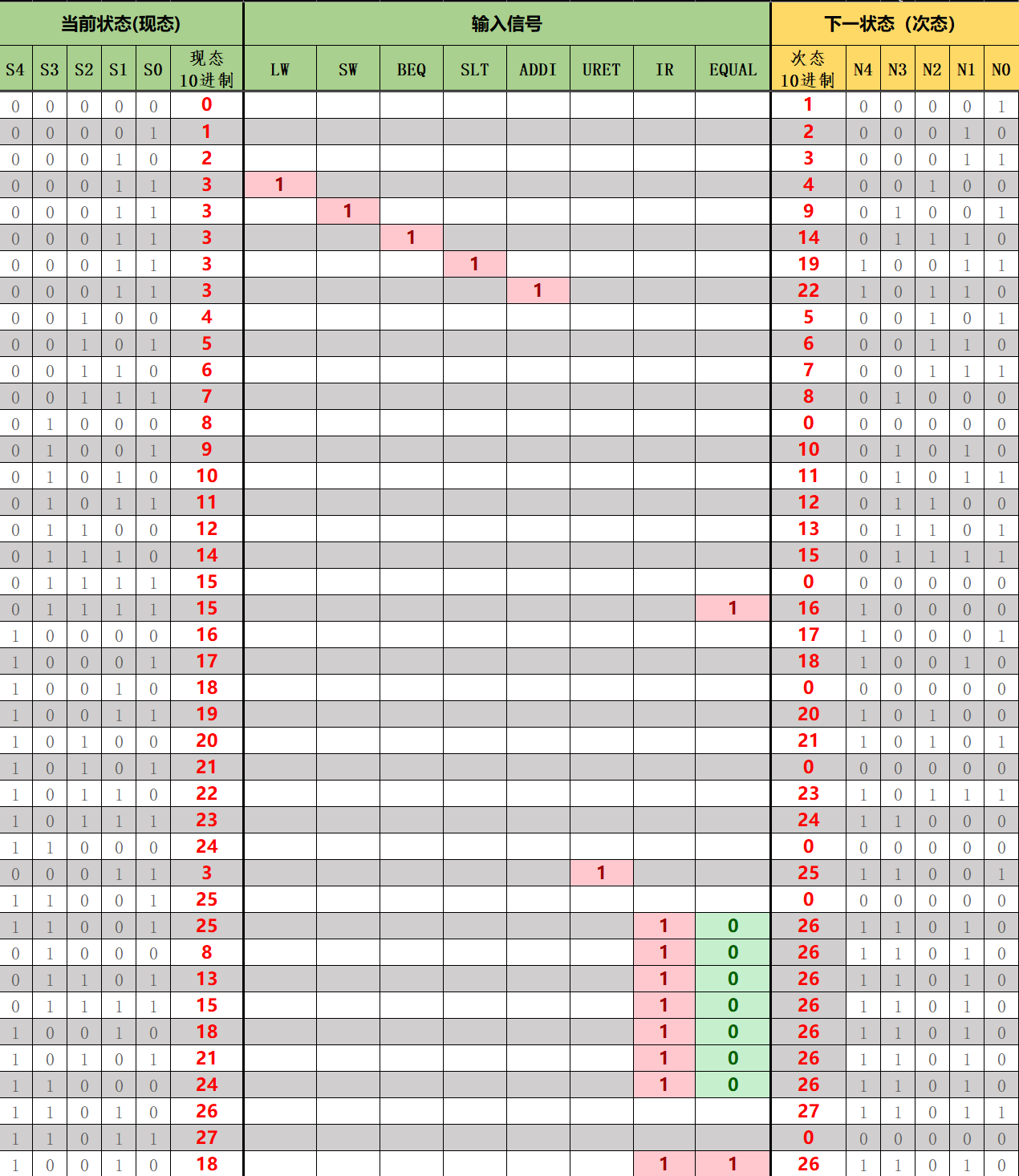


图 9 状态转换表

具体表达式和电路图过于复杂，就不在这里完全展示了。

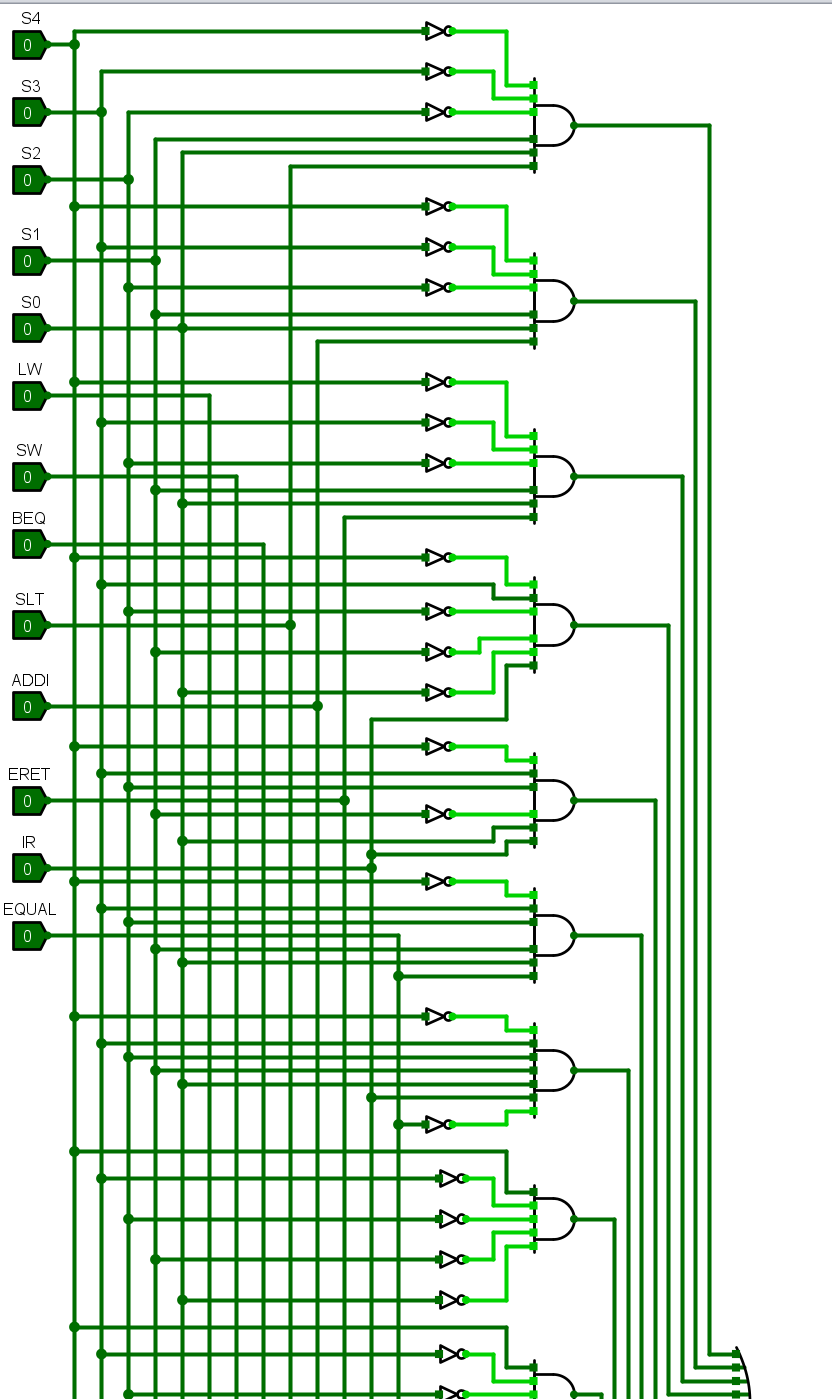


图 10 硬布线状态机部分电路图

### 支持中断的现代时序硬布线控制器设计

根据状态机内容进行连线，状态寄存器的输出，即次态，作为下一个状态机的现态时序输入，LW、SW、BEQ等信号摆好位置作为状态机的输入，然后控制存储器里的内容改为十六进制微指令，方便之后微指令直接调出使用。

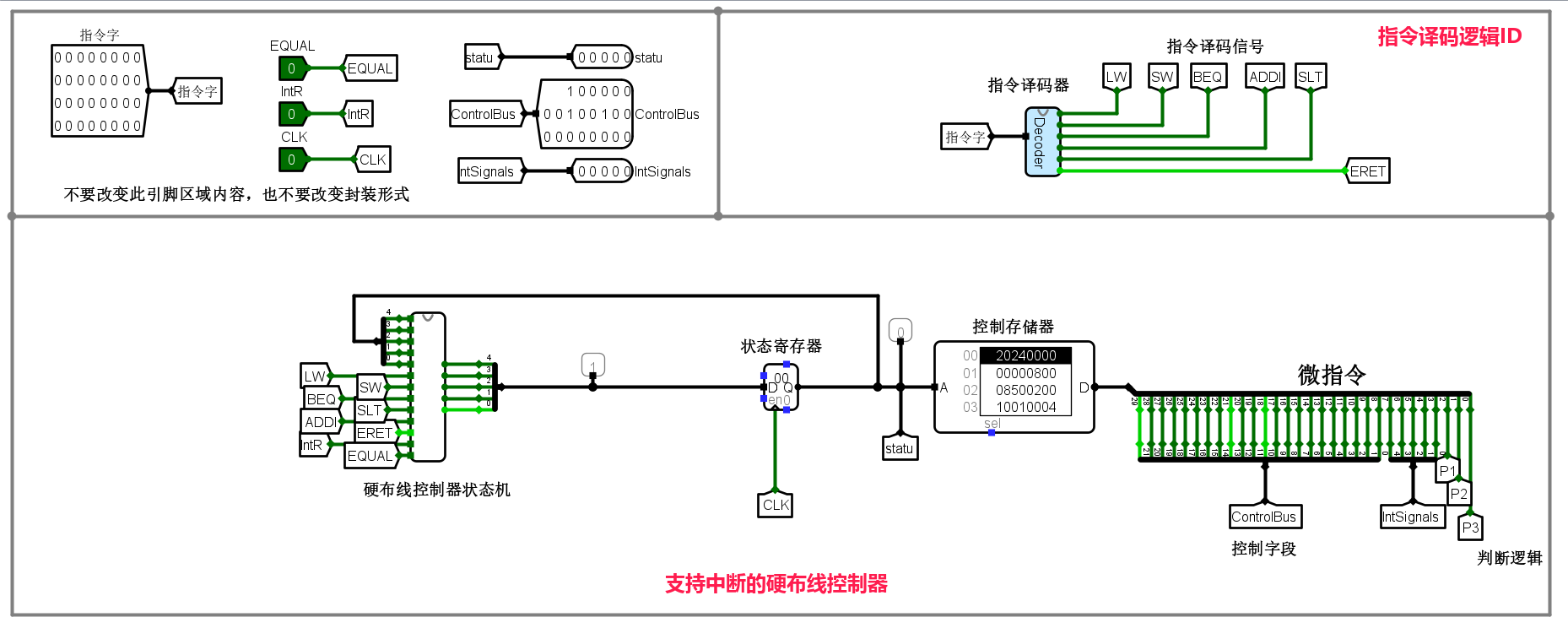


图 11 硬布线控制器电路图

### 支持中断的微程序单总线CPU设计

这一节我们只需要处理中断逻辑部分内容，中断1、2分别对应多路选择器的1、2端口，先对IntNo输入进行选择处理，中断控制器根据中断的类型，使程序运行不同的中断程序，所以中断控制器的IntNo端与一多路选择器的选择端相连，多路选择器的输入端为两中断程序入口地址。

用三态门根据中断控制信号进行内总线的读取数据和输出数据。

mEPC用于保存断点，当收到mEPCin信号时，从内总线读取异常指令地址，收到mEPCout时将保存的地址输出到内总线。因此mEPCin连接寄存器的使能端，寄存器输出端通过eEPCout控制的三态门与内总线相连，同时寄存器的输入端与内总线相连。

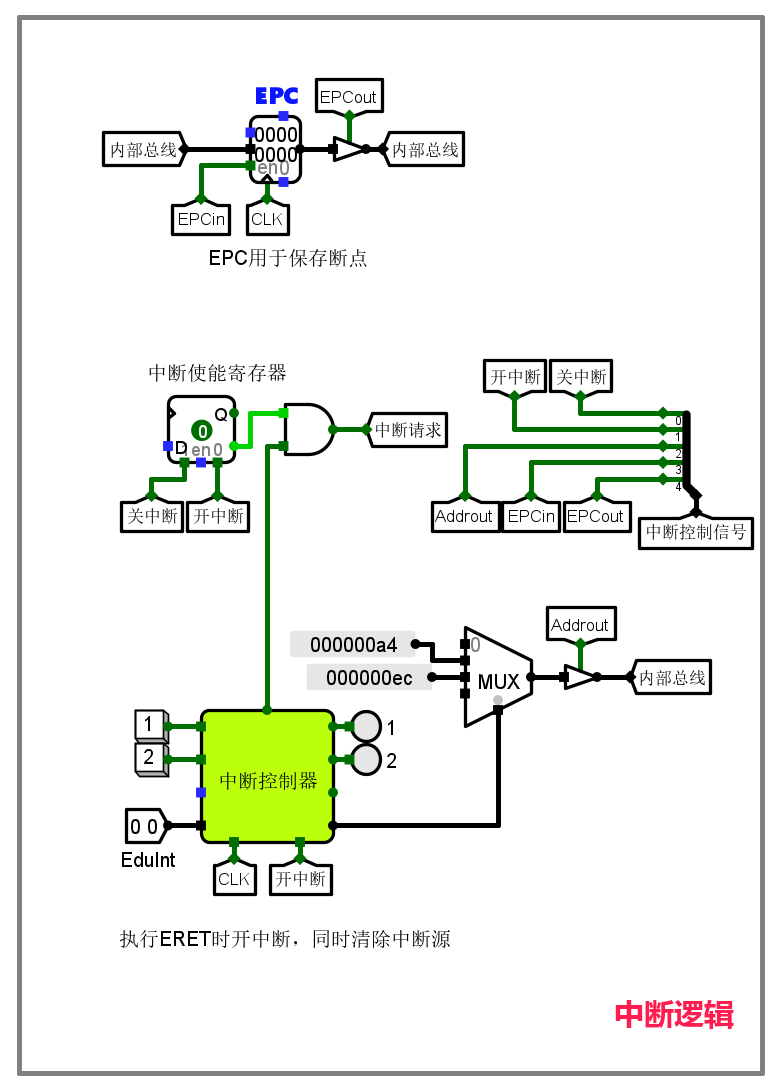


图 12 单总线CPU中断逻辑部分电路图

## 实验步骤

按照1.2的设计方案一步步完成头歌上的除了第5关的其他六关，最后再返回来完成第5关。

## 故障与调试

### 寄存器上升沿有效还是下降沿有效

**故障现象：在1.2.4中，一开始未改变寄存器的时钟沿，保持上升沿有效，导致测试一直未通过。**

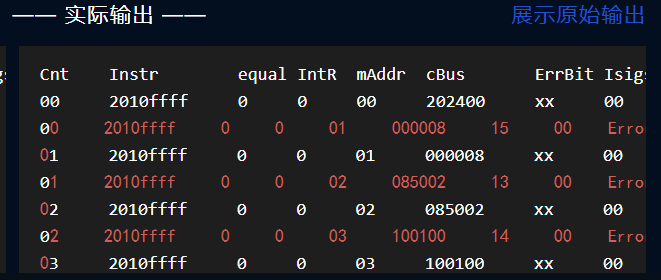


图 13 测试错误

**原因分析：如图13所示，保持上升沿有效会导致计数出错以至于后续输出都产生一定问题，微指令还没开始执行就被迫结束，必须得等下一个时钟周期到来才能接着执行。**

**解决方案：将寄存器的时钟沿改为下降沿有效，即可保持在调用完微指令后再结束周期。**

### 状态机设计中微指令跳转错误

**故障现象：在1.2.5中，状态机在现态为15时无法正确实现跳转，如当现态为15，equal=IR=1时，次态为26而不是16。**

**原因分析：这是由于在填写5号excel表时，IR或EQUAL部分忘记填写，导致几个会同时出现的变量在一起造成逻辑错误，而在此表中，不填写视为与其无关，对于几个不会同时出现的变量，这样做不会造成问题。**

**解决方案：现态为15时，将IR与EQUAL的取值分为不同情况依次填写。**

## 测试与分析

将Sort-5-int-riscv.hex加载到RAM中并运行，观察到程序进入死循环时节拍数为0x7c8，指令条数为251，内存中从0x80开始，按有符号数从大到小正确排列，表明电路正确且满足要求。

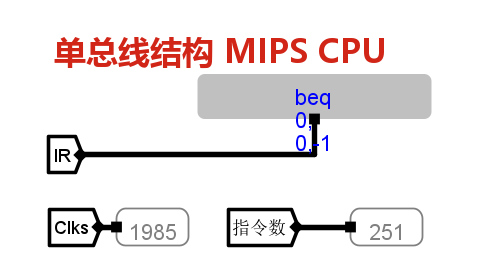


图 14 程序结束时的节拍数与指令条数

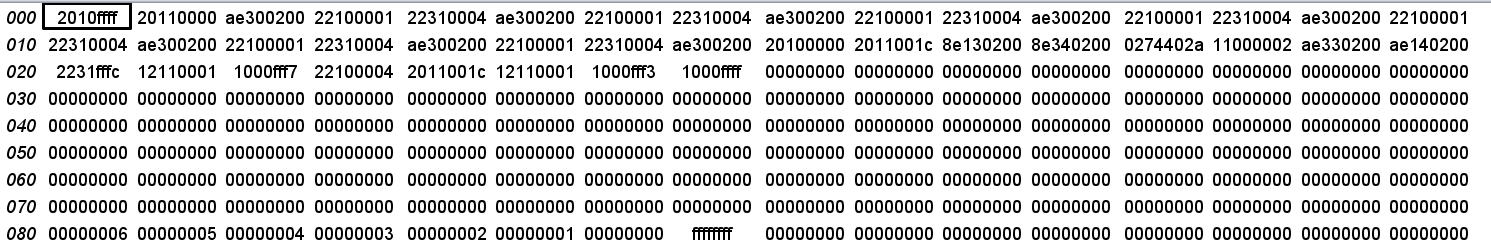


图 15 RAM中断的数的排序

头歌上的通关任务也是全部完成。

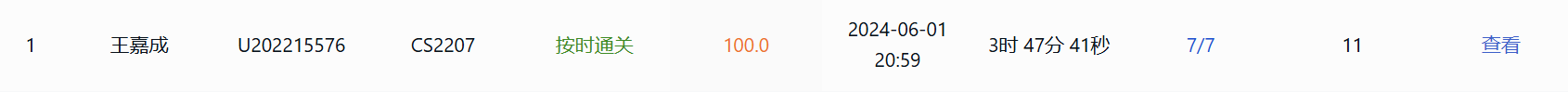


图 16 头歌通关记录

## 实验总结

本次实验主要完成了如下几点工作：

1. 完成CPU及其部分组件的方案总结：设计了包括MIPS指令译码器、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、支持中断的微程序单总线CPU、支持中断的现代时序硬布线控制器状态机、支持中断的现代时序硬布线控制器和支持中断的微程序单总线CPU方案。
2. CPU运行功能总结：完成了程序运行测试，实现了头歌上所有关卡的通关和最终加载程序的测试和运行。

## 实验心得

1. 掌握现代CPU的基本原理，通过聚焦现代时序和三级时序两种主要时序方式，以及硬布线和微程序控制方式的设计，还包括中断功能的实现，加深了我对教材第六章内容的掌握。
2. 熟悉使用Logisim工具，在实验中，充分利用了Logisim的基础功能和一些较高阶的功能，进行组合逻辑电路的分析。通过Excel表格的填写和逻辑表达式的生成，我感受到合理运用工具对学习过程的巨大便利性。
3. 深入了解CPU模块设计，通过实验，我掌握了CPU各个模块的功能及其逻辑实现，并进行了具体的电路设计，这种动手实践使我对CPU内部各运算器件的协同工作有了更加清晰的认识。

|  |
| --- |
| 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。  **IMG_20231225_212145_edit_437746824338412**特此声明！  **作者签字: 王嘉成** |
| 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:** |