***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： | CS2201 |
| 学 号： | U202215365 |
| 姓 名： | 叶俊江 |
| 电 话： | 15767792195 |
| 邮 件： | [1993783308@qq.com](mailto:13456@qq.com) |
| 完成日期： | 2024-06-20 |



# CPU设计实验

## 设计要求

利用logisim平台完成MIPS现代时序中断机制实现，任务由如下几部分组成：

·MIPS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

在Logisim中打开MipsOnBusCpu-1.circ文件完成电路连接后，再将文件使用文本编辑打开并将代码复制到头歌平台中评测，七个关卡都应评测通过。

## 方案设计

### MIPS指令译码器设计

子电路引脚功能如下图表1-1所示。

表1-1 指令译码器电路引脚与功能描述

|  |  |  |  |
| --- | --- | --- | --- |
| 引脚 | 输入/输出 | 位宽 | 功能描述 |
| IR | 输入 | 32 | MIPS指令字 |
| LW | 输出 | 1 | 当前指令为lw指令时输出为1 |
| SW | 输出 | 1 | 当前指令为sw指令时输出为1 |
| BEQ | 输出 | 1 | 当前指令为beq指令时输出为1 |
| ADDI | 输出 | 1 | 当前指令为addi指令时输出为1 |

子电路要求根据输入的指令字IR进行译码（IR已分为OP和FUNC两部分），判断是否为LW/SW/BEQ/ADDI/SLT指令或者是其他的指令，查阅MIPS32指令手册可知各个指令对应的OP与FUNC的值如下表1-2所示。

表1-2 各指令对应OP与FUNC的值

|  |  |  |
| --- | --- | --- |
| 指令 | OP | FUNC |
| SLT | 000000 | 101010 |
| BEQ | 000100 |  |
| ADDI | 001000 |  |
| LW | 100011 |  |
| SW | 101011 |  |

那么我们只需要将OP和FUNC分别与对应的常数值比较即可，若不是LW/SW/BEQ/ADDI/SLT这五种指令时Otherlnstr输出为1。电路图如下图1-1所示。

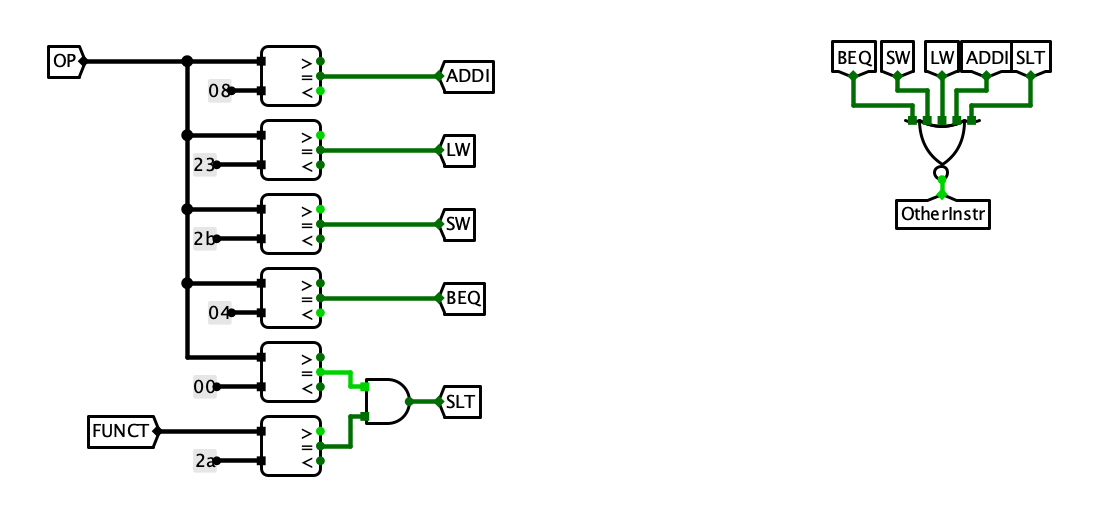


图 1-1 指令译码器电路图

### 支持中断的微程序入口查找逻辑

子电路引脚功能如下表1-3所示。

表1-3 支持中断的微程序入口查找逻辑电路引脚与功能描述

|  |  |  |  |
| --- | --- | --- | --- |
| 引脚 | 输入/输出 | 位宽 | 功能描述 |
| LW | 输入 | 1 | 为1时表示为lw指令 |
| SW | 输入 | 1 | 为1时表示为sw指令 |
| BEQ | 输入 | 1 | 为1时表示为beq指令 |
| ADDI | 输入 | 1 | 为1时表示为addi指令 |
| SLT | 输入 | 1 | 为1时表示为slt指令 |
| ERET | 输入 | 1 | 为1时表示为lw指令 |
| S4～S0 | 输出 | 1 | 5位微程序入口地址 |

根据教材中微指令格式和6条指令的微程序设计原理，由于我们加入了中断功能，那么微指令操作的控制字段就需要添加相应的内容，设计完成后可得输入指令为LW/SW/BEQ/ADDI/SLT/ERET的时候，微程序入口地址为4/9/14/19/22/25，相应的Excel表格填写如下图1-2所示。



图 1-2 微程序入口地址表

电路图如下图1-3所示。



图 1-3 微程序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

子电路引脚功能如下表1-4所示。

表1-4 支持中断的微程序条件判别测试逻辑电路引脚与功能描述

|  |  |  |  |
| --- | --- | --- | --- |
| 引脚 | 输入/输出 | 位宽 | 功能描述 |
| P0 | 输入 | 1 | 条件判断位，用于指令译码 |
| P1 | 输入 | 1 | 条件判断位，用于beq分支 |
| P2 | 输入 | 1 | 条件判断位，用于中断检测 |
| equal | 输入 | 1 | 运算操作数相等标志位 |
| IntR | 输入 | 1 | 中断请求信号 |
| MuxSel | 输出 | 2 | MUX选择控制信号 |

根据教材中支持中断微程序控制器原理图，再结合判别位测试位的定义，设计条件判别测试组合逻辑。当P0～P2中出现1时，表示需要进行分支判断，随后再根据equal和IntR的值，判断是否进入入口地址/beq分支/中断响应入口/取值微程序入口。相应的Excel文件填写如下图1-4所示。



图 1-4 判别测试逻辑真值表

由于电路图是使用Logisim的自动生成功能生成的，较为复杂，故在此处不再列出电路图。

### 支持中断的微程序控制器设计

子电路引脚功能如下表1-5所示。

表1-5 支持中断的微程序控制器设计电路引脚与功能描述

|  |  |  |  |
| --- | --- | --- | --- |
| 引脚 | 输入/输出 | 位宽 | 功能描述 |
| IR | 输入 | 32 | MIPS指令字 |
| equal | 输入 | 1 | 条件反馈信号，表示运算相等 |
| IntR | 输入 | 1 | 中断请求信号，高电平有效 |
| mAddr | 输出 | 1 | 当前微指令地址 |
| ControlBus | 输出 | 1 | 微指令字的控制字段 |

将微程序入口查找逻辑、判别测试逻辑以及控制存储器等部件进行连接，实现微程序控制器的主要数据通路，设计微程序并且加载到控制存储器中。

我们将微程序入口查找逻辑的输出与MUX的“入口地址”输入进行连接，将判别测试逻辑的输出与MUX的选择控制端进行连接，将mAddr通过加法器加1后得到下一微指令地址，连接到MUX的“顺序地址”输入，电路图如下图1-5所示。

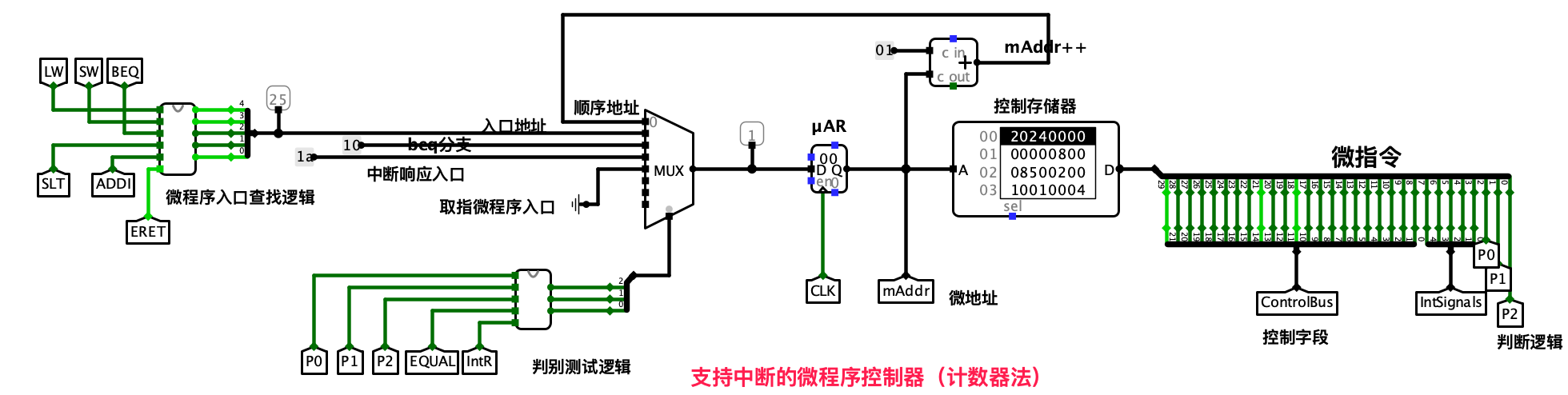


图 1-5 微程序控制器电路图

其中控制存储器中的数据（微指令）由实验包中Excel文件自动生成，Excel的填写情况如图1-6所示。

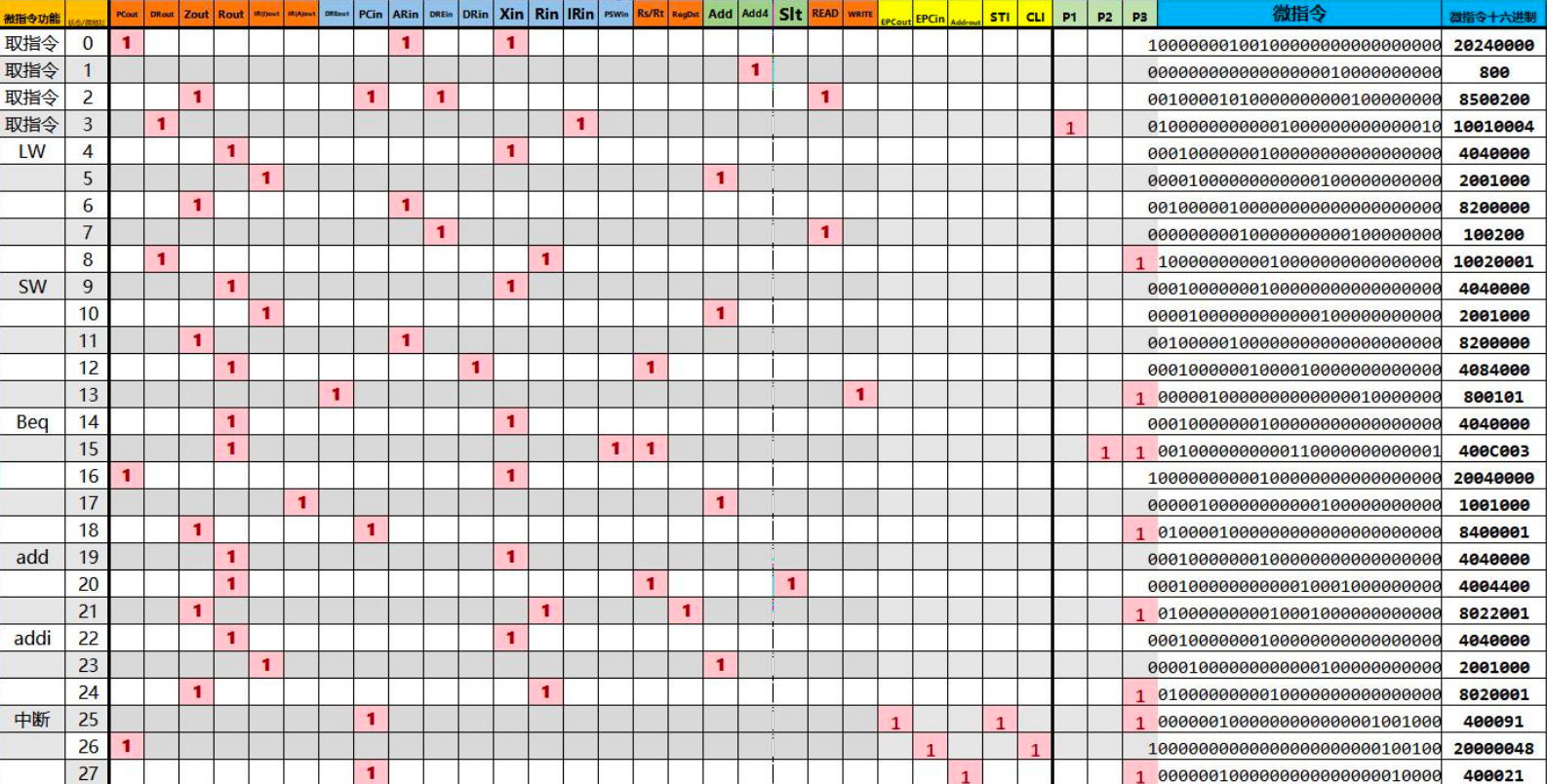


图 1-6 微指令表

值得注意的是，每一个微程序的最后一条指令都需要设置P2=1，并且只有中断微程序才需要设置中断控制信号。

### 支持中断的微程序单总线CPU设计

子电路引脚功能如下表1-6所示。

表1-6 支持中断的微程序单总线CPU设计电路引脚与功能描述

|  |  |  |  |
| --- | --- | --- | --- |
| 引脚 | 输入/输出 | 位宽 | 功能描述 |
| ControlBus | 输出 | 22 | 控制总线，用于EduCoder系统自动测试 |
| 中断控制信号 | 输出 | 5 | 中断控制信号 |

本关主要是要在单总线数据通路中添加与中断相关的硬件模块，主要是包括异常程序设计计数器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，并在本馆进行最终的联调，测试CPU能否正常响应两个按键所对应的中断服务程序。（注意不同按键的中断服务程序入口地址可以利用MARS汇编器汇编源程序查看label地址得到）电路图如下图1-7所示。

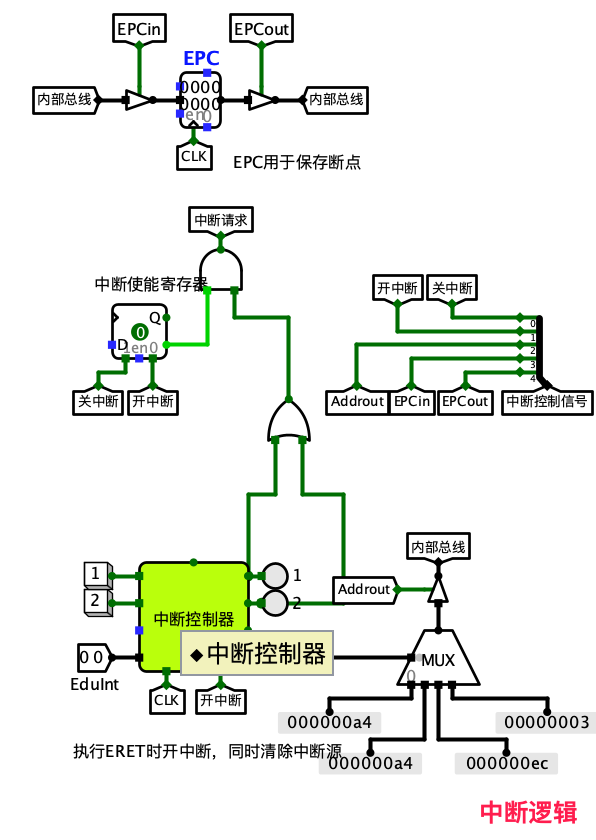


图 1-7 中断逻辑电路图

### 支持中断的现代时序硬布线控制器状态机设计

子电路引脚功能如下表1-7所示。

表1-6 支持中断的微程序单总线CPU设计电路引脚与功能描述

|  |  |  |  |
| --- | --- | --- | --- |
| 引脚 | 输入/输出 | 位宽 | 功能描述 |
| LW | 输入 | 1 | 为1表示为lw指令 |
| SW | 输入 | 1 | 为1表示为sw指令 |
| BEQ | 输入 | 1 | 为1表示为beq指令 |
| SLT | 输入 | 1 | 为1表示为slt指令 |
| ADDI | 输入 | 1 | 为1表示为addi指令 |
| ERET | 输出 | 1 | 为1表示为eret指令 |
| IntR | 输入 | 1 | 中断请求信号 |
| equal | 输入 | 1 | 运算操作数相等标志位 |
| 现态S0~S4 | 输入 | 5 | FSM状态机现态 |
| 次态N0~N4 | 输出 | 5 | FSM状态机次态 |

要实现的状态图如图1-8所示。

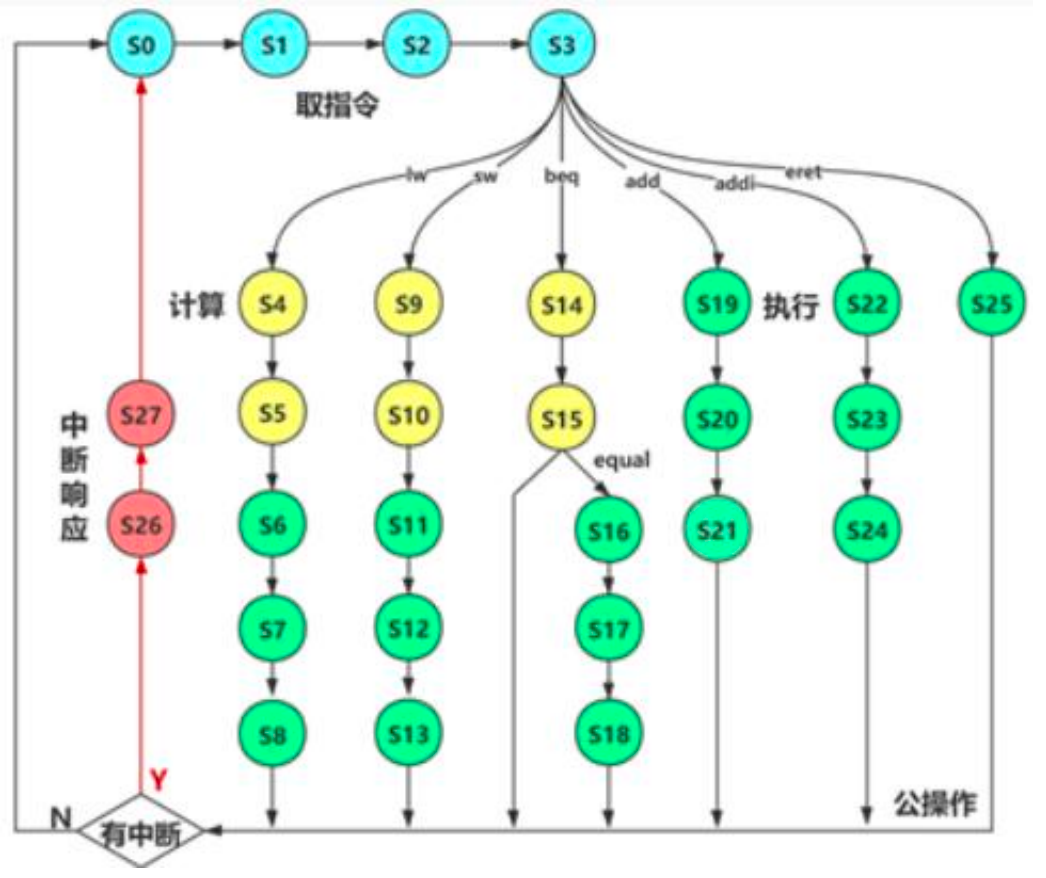


图 1-8 支持中断的现代时序状态机

根据上图可直接填写相应的Excel文件，自动生成组合逻辑，填写文件如图1-9所示。

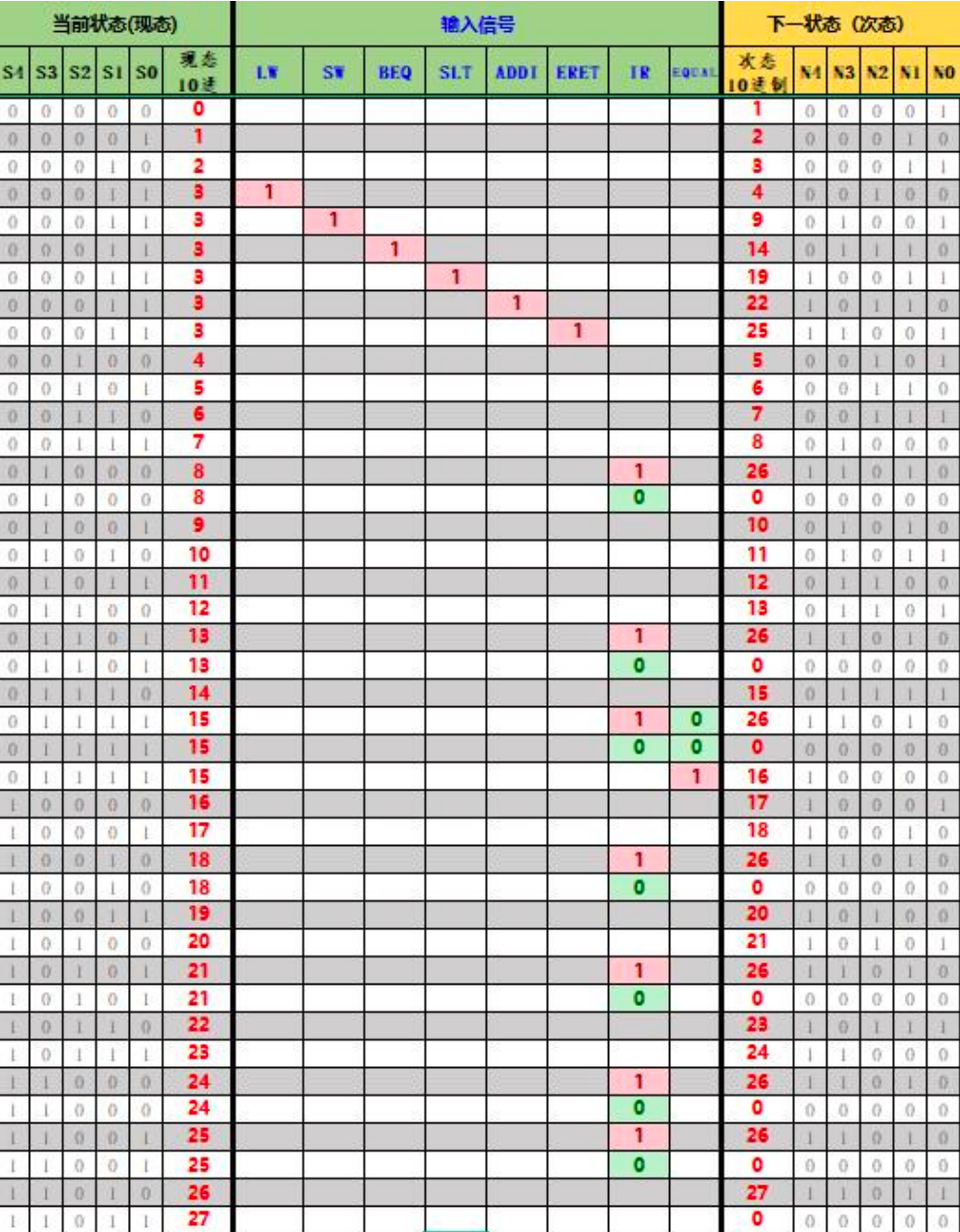


图 1-9 硬布线控制器状态机状态转换表

由于电路图是使用Logisim的自动生成功能生成的，较为复杂，故在此处不再列出电路图。

### 支持中断的现代时序硬布线控制器设计

子电路引脚功能如下表1-8所示。

表1-8 支持中断的现代时序硬布线控制器设计电路引脚与功能描述

|  |  |  |  |
| --- | --- | --- | --- |
| 引脚 | 输入/输出 | 位宽 | 功能描述 |
| IR | 输入 | 32 | MIPS指令字 |
| CLK | 输入 | 1 | 时钟输入 |
| equal | 输入 | 1 | 条件状态信号，相等标志 |
| 控制总线 | 输出 | 22 | 所有控制信号汇总而成的控制总线 |
| 当前状态值 | 输出 | 5 | 状态字的值 |

在实现指令译码、现代时序状态机模块后，最终完成硬布线控制器的封装。如下图1-10中完成了硬布线控制器框架连接，此时需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。

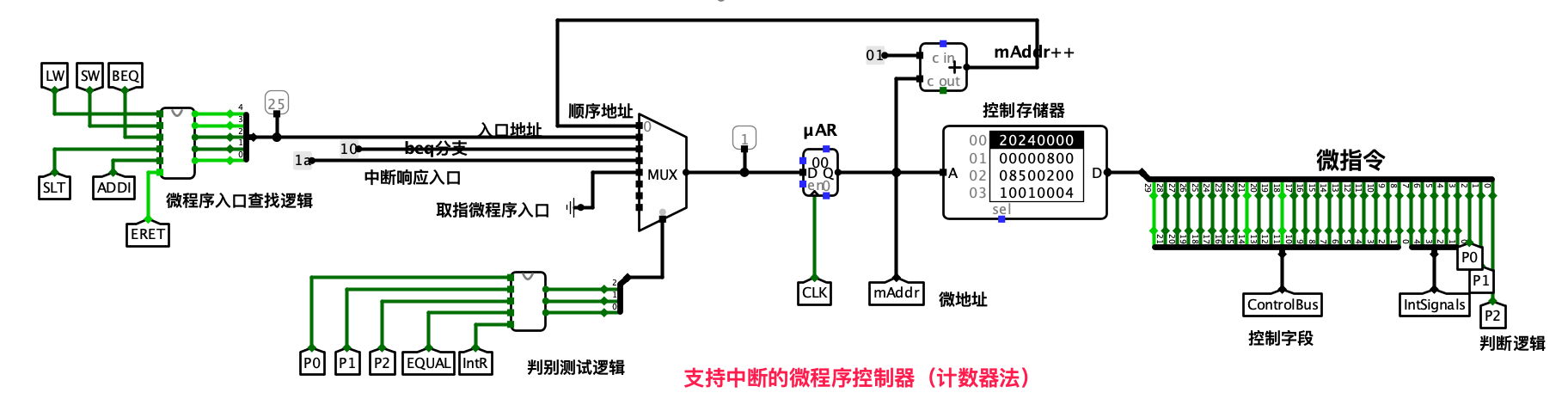


图 1-10 硬布线控制器电路图

## 实验步骤

1. 仔细阅读头歌平台各关卡中的实验提示，并了解各部分子电路的功能。
2. 根据提示及上课所学知识完成子电路的设计，对于部分关卡需要填写实验包里面提供的Excel文件来自动生成表达式，再使用Logisim的自动生成功能生成组合逻辑电路。
3. 提交代码文件到头歌平台进行评测。

## 故障与调试

### 状态转移存在问题

**故障现象：**在“支持中断的微程序控制器设计”关卡中，后续近一半指令全部错误。

**原因分析：**在Excel文件中设计时，在SW微程序最后一条指令P3=1未设置，导致SW微程序执行后无法返回取指微程序。

**解决方案：**修改对应Excel文件中出错的跳转逻辑。

## 测试与分析

在“支持中断的微程序单总线CPU设计”关卡中可对设计的电路进行测试。在RAM中加载sort-5-int.hex数据镜像，使用快捷键⌘command+k即可运行，若无其他按键行为，程序会在1992节拍数停下，指令计数为252。测试结果如下图1-11所示。

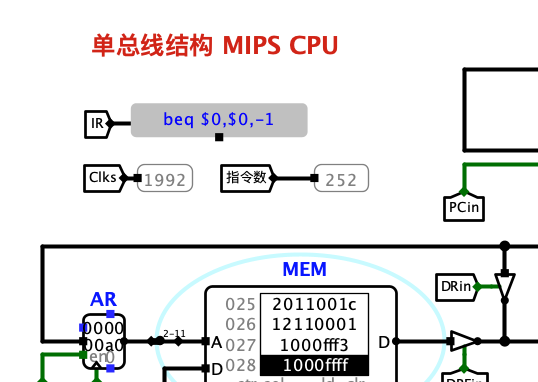


图 1-11 CPU测试结果

从测试结果可见符合我们的预期结果，故而电路设计正确。

内存数据如下图1-12所示。

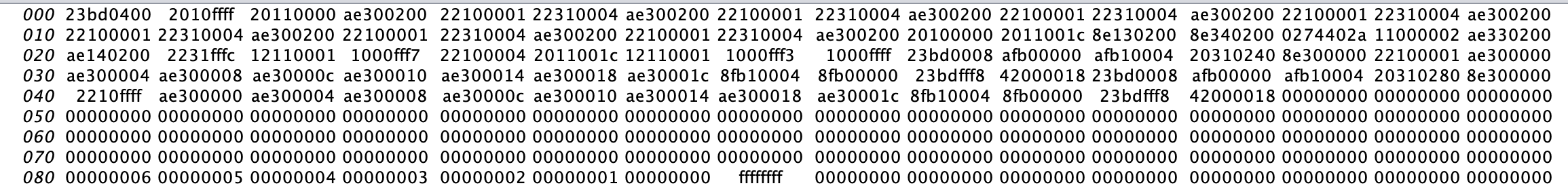


图 1-12 内存数据

## 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：本次实验实现了MIPS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。本次试验是在“单总线CPU设计（现代时序）”的基础上完成的，添加了中断机制。
2. 功能总结：本次实验实现的CPU可以按照微程序或硬布线的方式来处理数据，并且支持中断。

## 实验心得

1. 对logisim的使用进一步熟练，如利用表达式生成组合逻辑电路，一些常用的快捷键等功能，使设计电路的效率有所提升。
2. 通过本次实验，我对课本知识的理解更加深刻透彻，对CPU的结构和运行原理更加清晰，并且对硬布线和微程序两者的优势不足和不同之处认识更加充分。总的来说本次实验帮助我巩固了课本知识，激发了我对计算机硬件设计的兴趣，为日后的学习打下基础。

|  |
| --- |
| 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:** |