***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： | U202215357 |
| 姓 名： | 王文涛 |
| 电 话： |  |
| 邮 件： |  |
| 完成日期： | 2024-06-22 |



# CPU设计实验

## 设计要求

利用logisim平台中现有运算部件设计支持中断的微程序控制器的单总线CPU，使得MIPS程序能在单总线结构上运行，最终能运行简单的排序程序sort-5-int.hex。运算器功能以及输入输出引脚见下表，在主电路中详细测试自己封装的运算器。

## 方案设计

### MIPS指令译码器设计

指令译码器将指令字翻译成一根根的指令译码信号，将32位MIPS 指令字译码生成LW、SW、BEQ、SLT、ADDI、OtherInstr信号。

对于LW、SW、BEQ、ADDI等I型指令，可以通过操作码OP区分。



图 1‑1 I型指令机器码

SLT为R型指令，OP皆为0，可以根据扩展操作码funct区分。



图 1‑2 R型指令机器码

使用比较器即可根据指令的OP和funct生成对应的指令译码信号，若不是LW、SW、BEQ、SLT、ADDI信号则为OtherInstr信号。

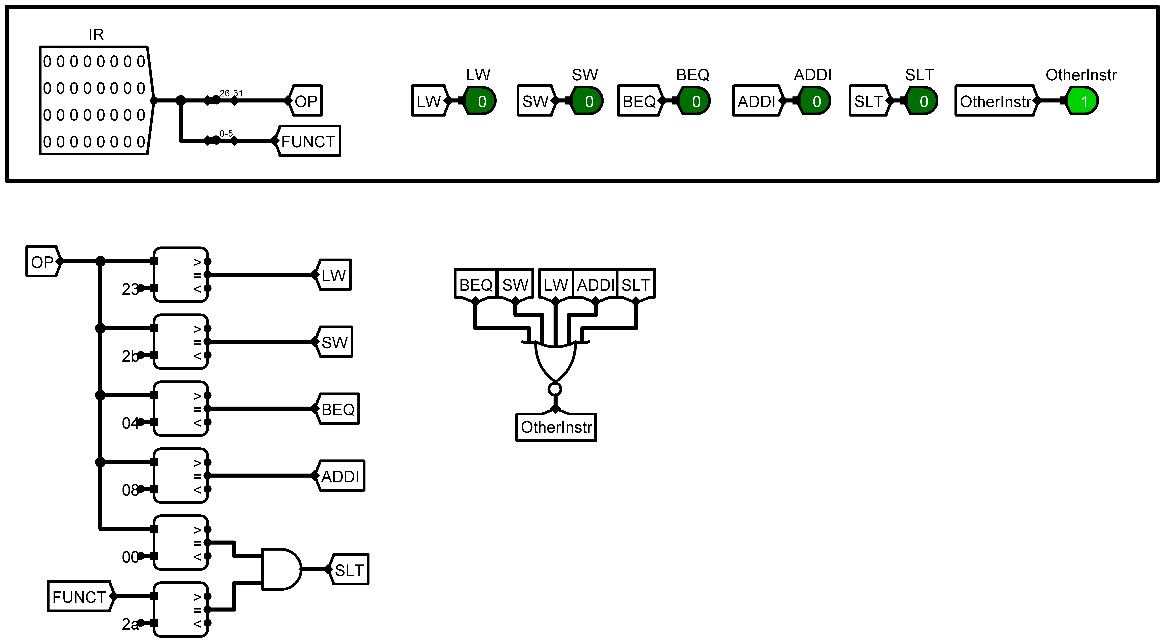


图 1‑3指令译码器电路图

### 支持中断的微程序入口查找逻辑

微程序按照取指令，LW、SW、BEQ、SLT、ADDI、ERET的顺序放置在控制存储器中。由于取指令需要4个节拍，LW指令地址从4开始。LW、SW、BEQ需要5个节拍，SLT、ADDI需要3个节拍，所以入口地址分别为9、14、19、22、25。填写6号EXCEL表格中的微程序入口地址表格如下图所示。



图 1‑4微程序入口地址表

将生成的逻辑表达式复制到logisim中。得到电路图如下。

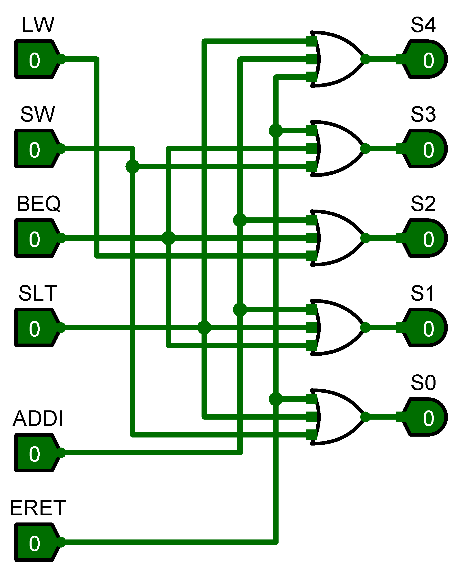


图 1‑5微程序入口查找逻辑电路图

### 支持中断的微程序条件判别测试逻辑

表 1‑1引脚功能

|  |  |  |  |
| --- | --- | --- | --- |
| 引脚 | 输入/输出 | 位宽 | 功能描述 |
| P0 | 输入 | 1 | 条件测试位，根据指令功能进行微程序分支 |
| P1 | 输入 | 1 | 条件测试位,根据PSW.equal标记进行分支 |
| P2 | 输入 | 1 | 条件测试位，表示是最后一条微指令，需要判断中断请求IntR |
| equal | 输入 | 1 | 相等标记 |
| intR | 输入 | 1 | 中断请求信号 |
| S3~S0 | 输出 | 3 | 后续地址多路选择控制信号 |

若P0=0，P1=0，P2=0，表示按顺序取地址，地址加一，多路选择控制信号为0。

若P0=1，P1=0，P2=0，表示取址周期结束，下一步取微指令的入口地址，多路选择控制信号为1。

若P0=0，P1=1，P2=1，表示beq指令计算周期结束，需要判断equal标记和intR信号。若equal=1,无论intR为何值，都需要先跳转到beq分支。多路选择控制信号为2。若equal=0，intR=0，则跳转到取址微指令入口重新取址，多路选择控制信号为4。若equal=0，intR=1，则跳转到中断处理程序入口，多路选择控制信号为3。

若P0=0，P1=0，P2=1，表示指令执行周期结束，需要判断intR信号。若intR=0，则重新进入取址周期，多路选择控制信号为4。若intR=1，则跳转到中断处理程序入口，多路选择控制信号为3。

根据上述分析填写4号EXCEL表格中的组合逻辑真值表如下图。

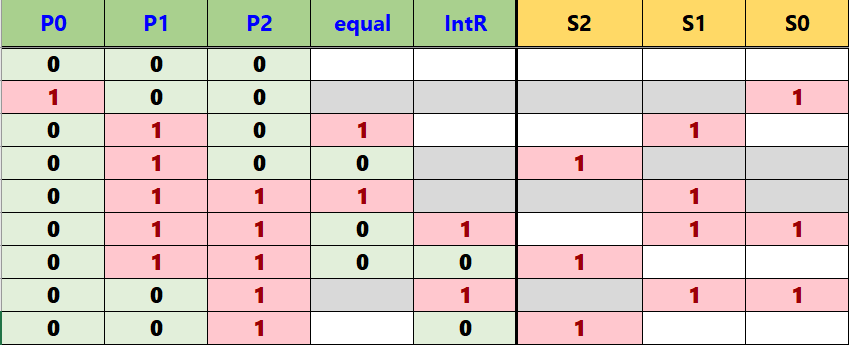


图 1‑6判别测试逻辑

将得到的逻辑表达式导入电路，得到如下电路图。

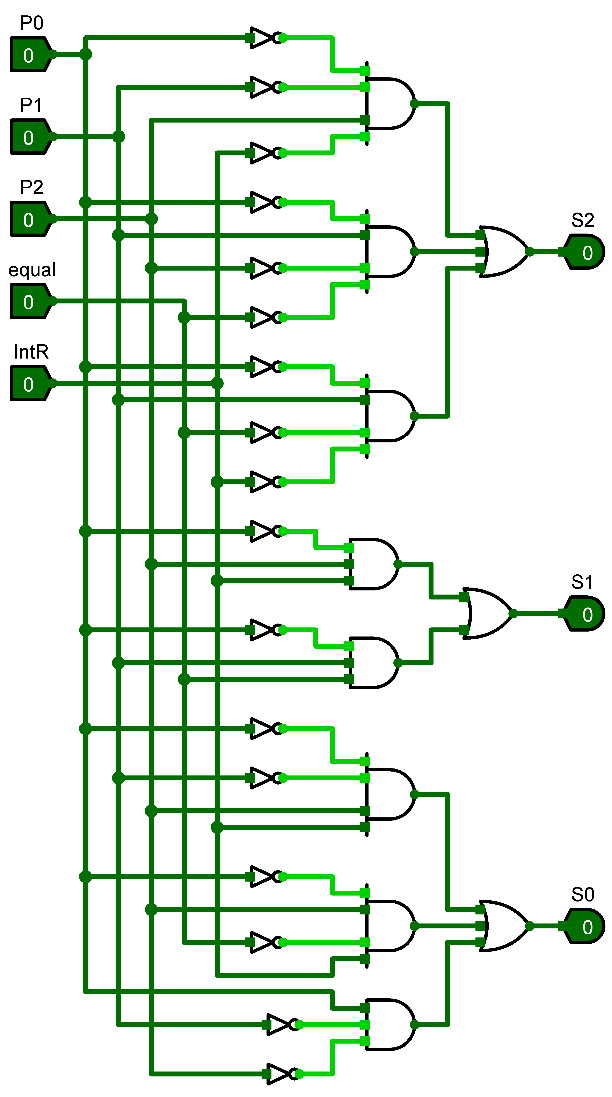


图 1‑7微程序条件判别测试逻辑电路图

### 支持中断的微程序控制器设计

首先填写6号EXCEL文件中的微程序自动生成表。

根据下图填写取指令周期控制信号，其中T4取址周期结束，下一节拍取微指令的入口地址，P1=1。



图 1‑8取指令周期控制信号

根据下图填写取Lw、Sw、Beq、Slt、Addi控制信号，在每个指令的最后一个节拍P3=1，表示指令结束。

其中在Beq的第二节拍计算周期结束时需要根据equal的值判断是否跳转到Beq执行周期分支，所以P2=1，P3=1。

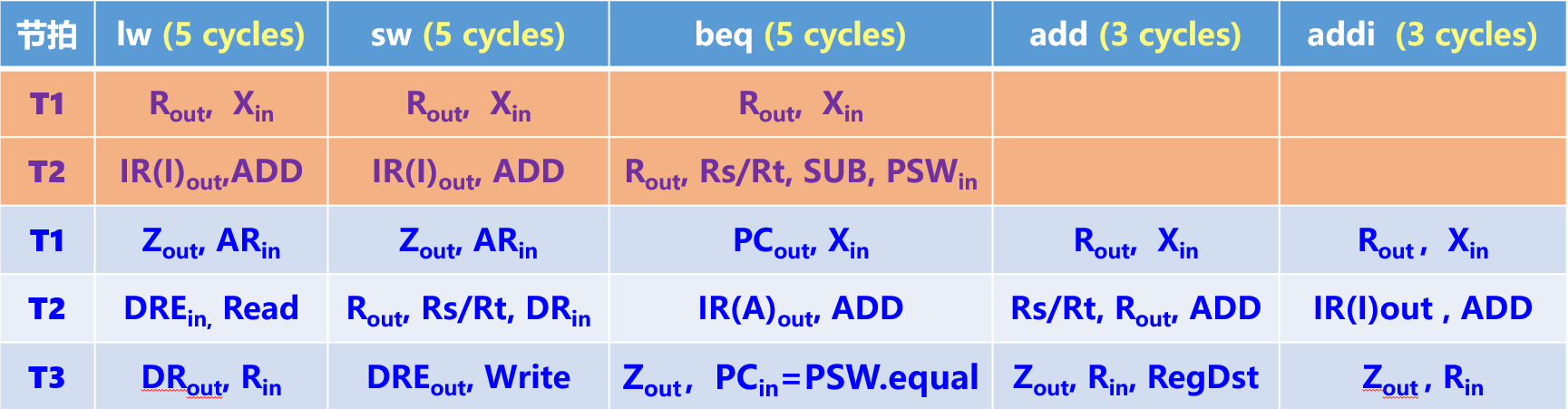


图 1‑9指令控制信号

Eret指令只有一个节拍。根据下图填写，其中STI、为中断控制信号。P3=1。



图 1‑10 eret指令控制信号

接下来两个微地址存放中断响应微指令。根据下图填写。P字段均为0。

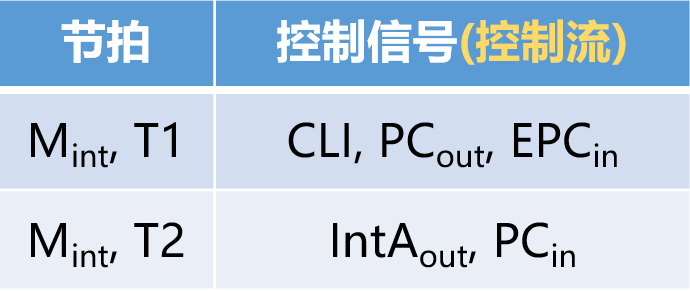


图 1‑11中断响应控制信号

填写6号EXCEL文件中的微程序自动生成表（未截取生成的微指令）如下。

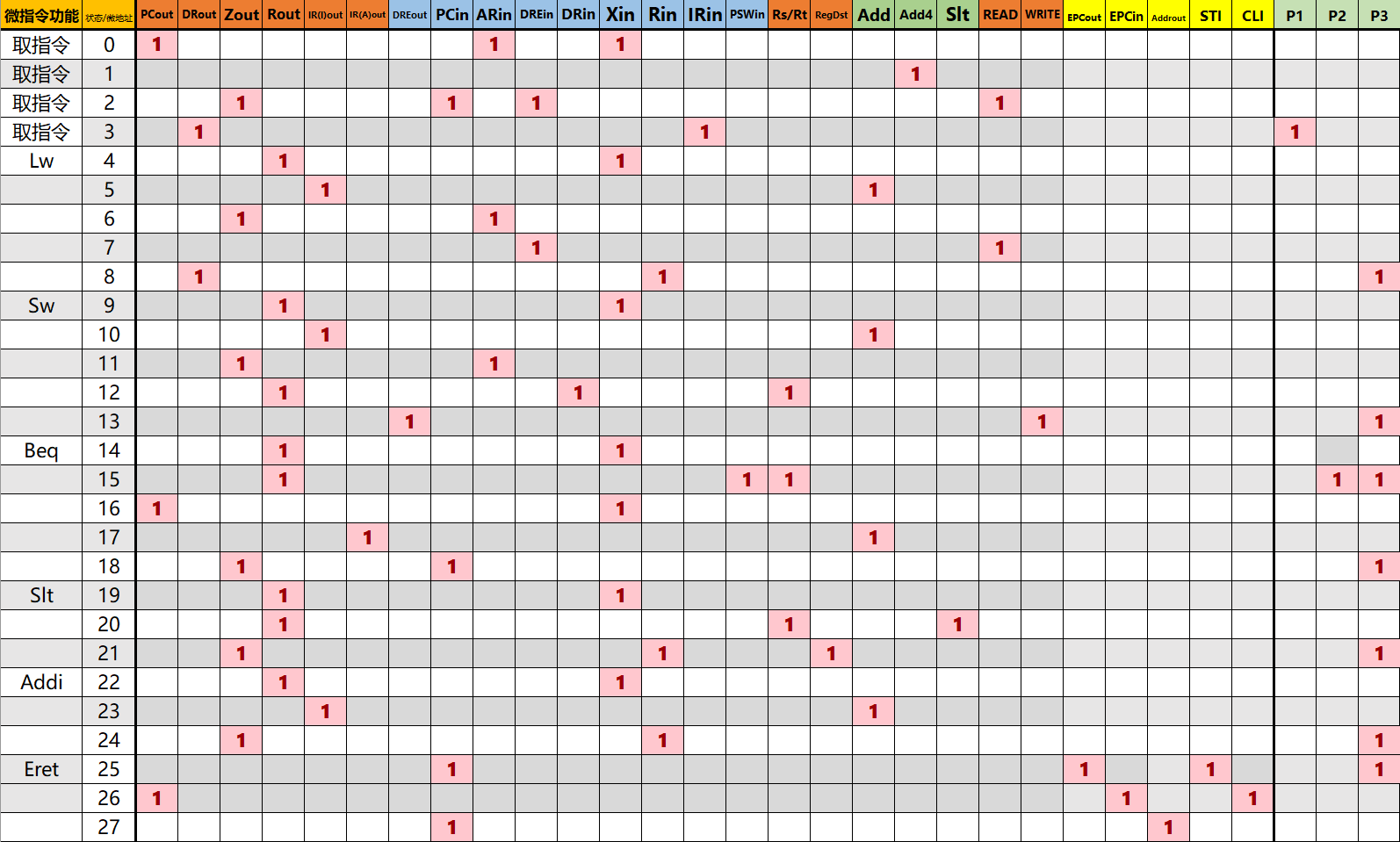


图 1‑12微程序自动生成表

将微指令十六进制编码直接复制粘贴到电路图的ROM中。

接着完善电路图。将加法器的输出（mAddr++）连到顺序地址处，表示指令地址加一。将微程序入口查找逻辑输出连接到入口地址。根据微程序可知beq分支地址为0x10，中断响应入口为0x1a，取址微程序入口为0。

将判别测试逻辑的输出使用分线器（注意分线器从上到下端口为2，1，0）连接到MUX的选择端。

最终电路图如下。

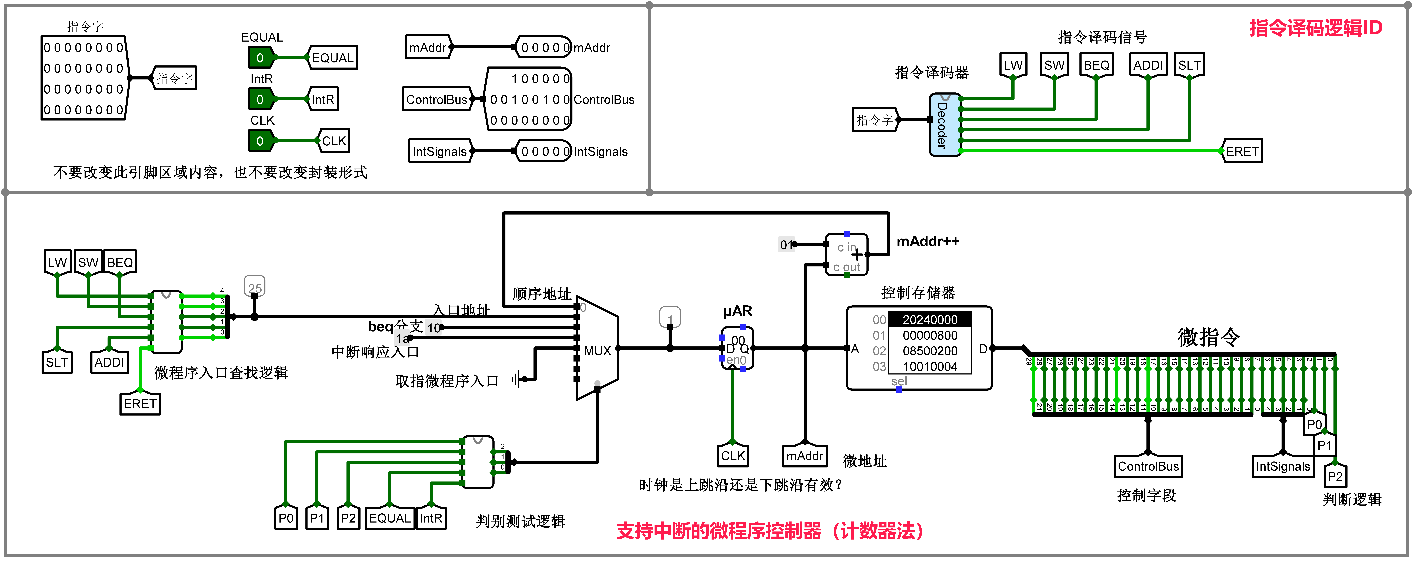


图 1‑13微程序控制器电路图

### 支持中断的微程序单总线CPU设计

不同按键的中断服务程序入口地址需要利用MARS汇编器汇编源程序查看lable地址。将sort-5-int.asm汇编文件导入MARS，查看lable得到中断程序（intProgram）的地址分别为0xa4和0xec。

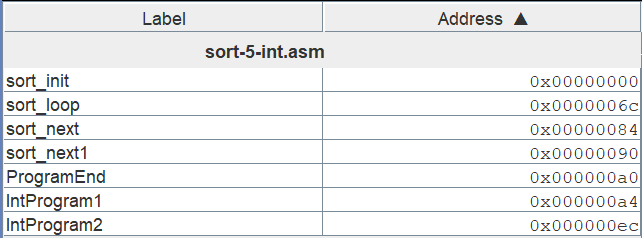


图 1‑14中断程序地址

电路图只需根据中断异常处理数据通路，补充中断逻辑部分即可。

EPC寄存器输入端口直接与内部总线相连，输出端口通过三态门与内部总线相连，由信号控制。使能端由控制。

中断使能寄存器只由开中断和关中断信号直接控制，状态的非值与中断控制器的输出（intR）通过与门连接，输出中断请求。关中断后，中断请求始终为1，起到了屏蔽中断请求的效果。

中断控制器的中断号作为选择，将终端号对应的中断处理程序入口地址输出到内部总线。

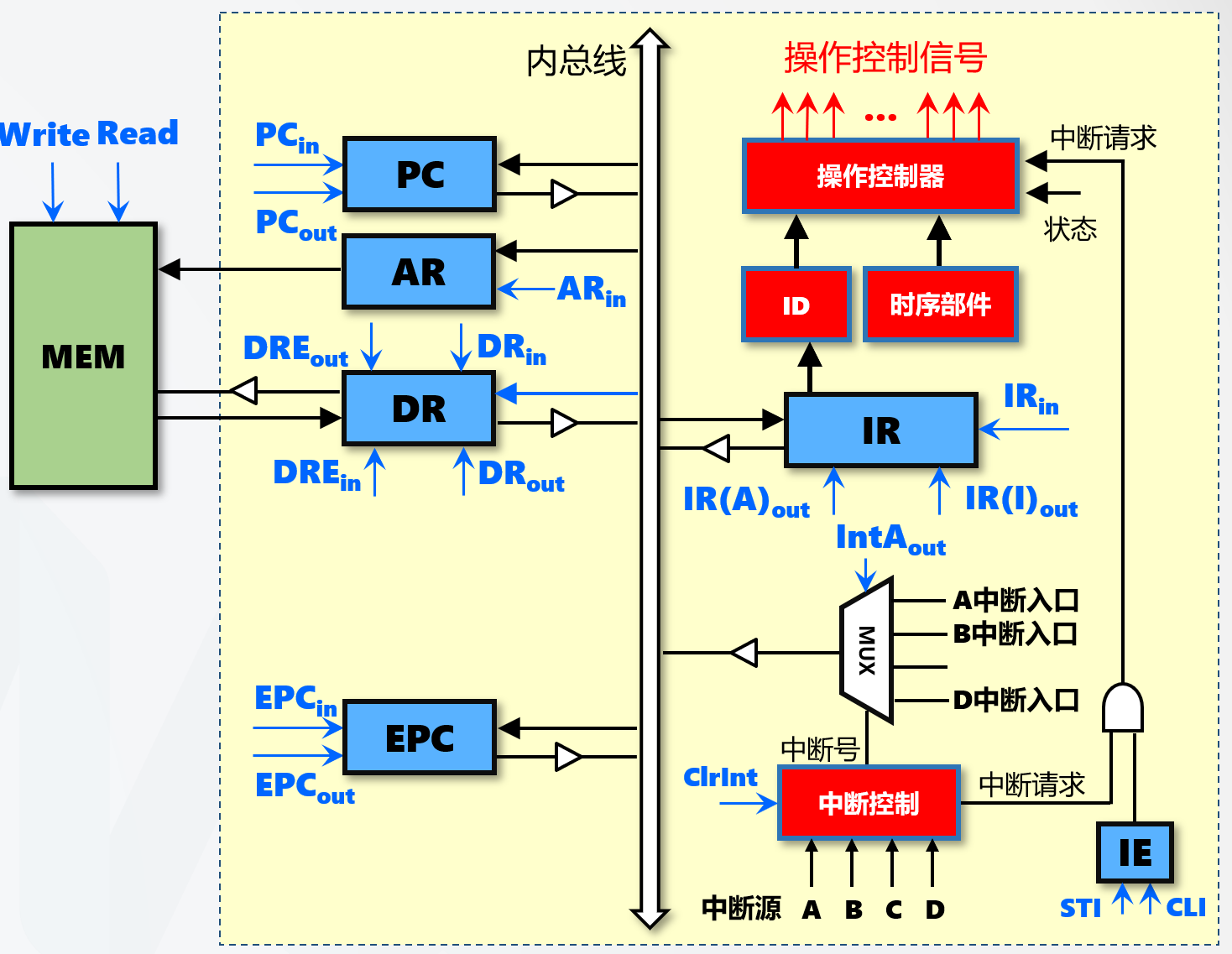


图 1‑15中断异常处理数据通路

最终电路图如下图所示。

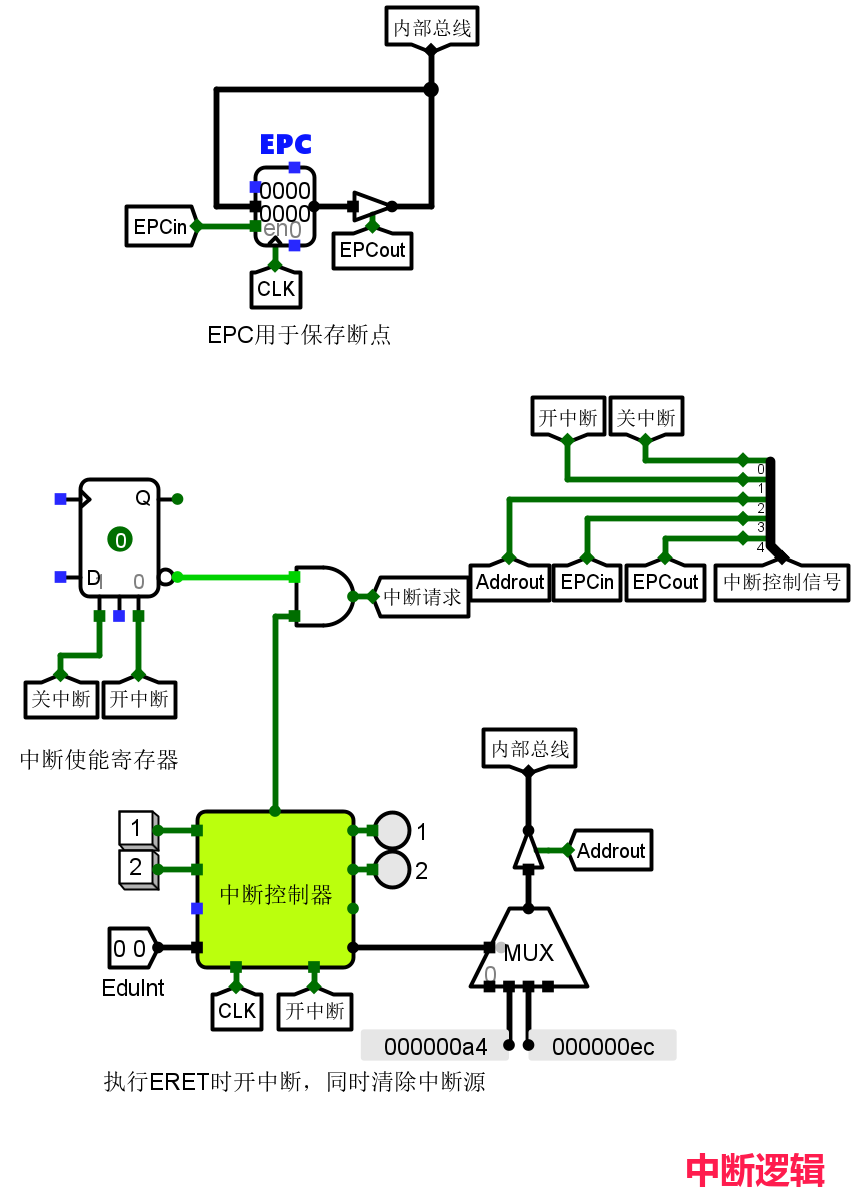


图 1‑16单总线CPU电路图

### 支持中断的现代时序硬布线控制器状态机设计

根据下图所示的现代时序状态机可知，一共存在27个状态。

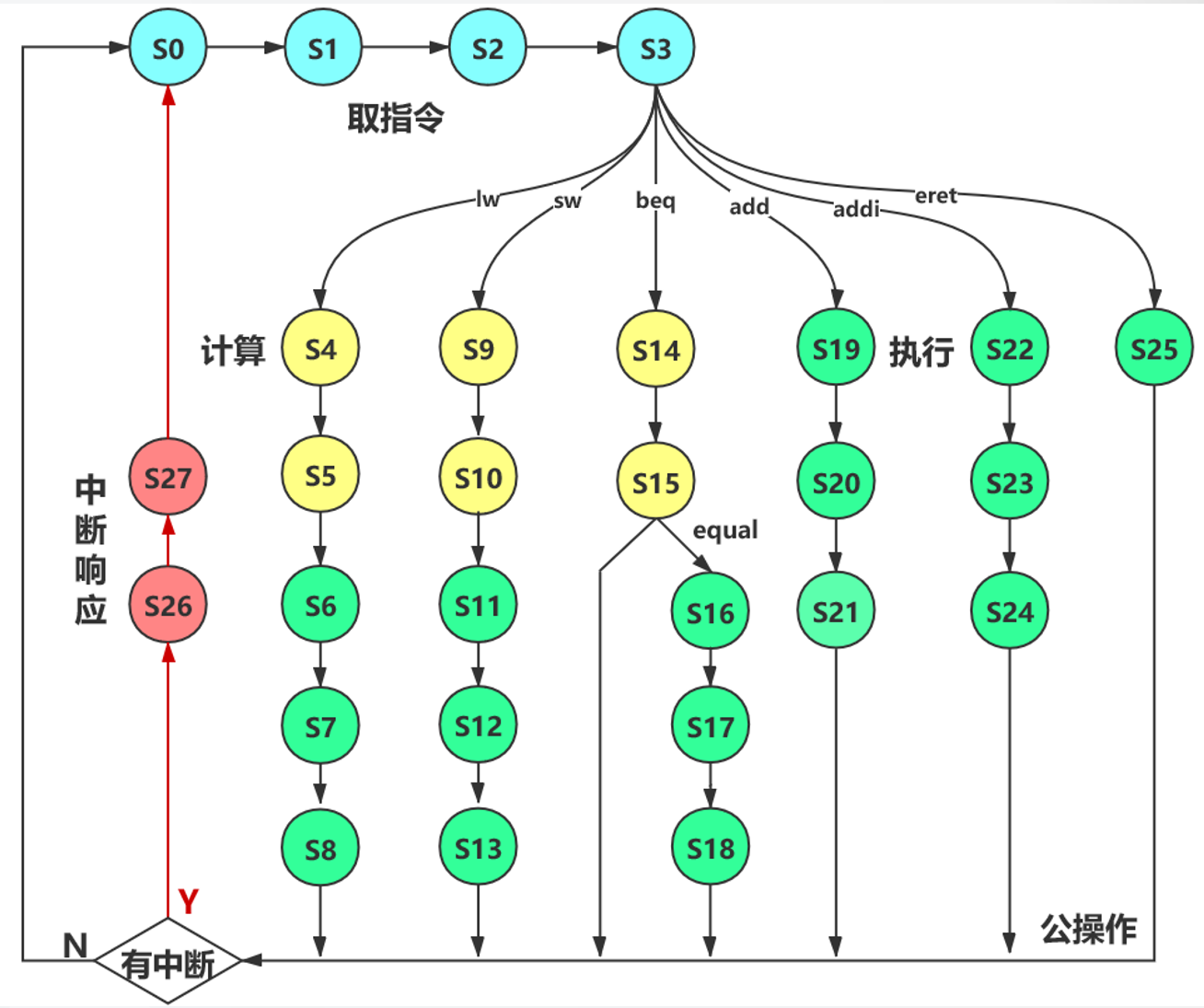


图 ‑现代时序状态机

状态在取指令周期，计算周期，执行周期和中断响应周期中分别自然递增，无需信号。

在取指令周期的最后一个状态S3时，根据指令信号lw、sw、beq、add、addi、eret的值决定进入S4,S9,S14,S19,S22,S25中的一个。

在执行周期的最后一个状态S8,S13,S15,S18,S21,S24,S25时。若中断信号IR的值为1则进入中断响应S26，否则直接进入取指令周期S0。

特别的，进入beq分支的S15状态时，需要同时根据equal信号和IR信号决定下一状态。若equal=1，则进入执行周期中的S16。若equal=0，IR=1，则进入中断响应周期S26。若equal=0，IR=0，则直接进入取指令周期S0。

根据上述分析。填写5号excel表如下。

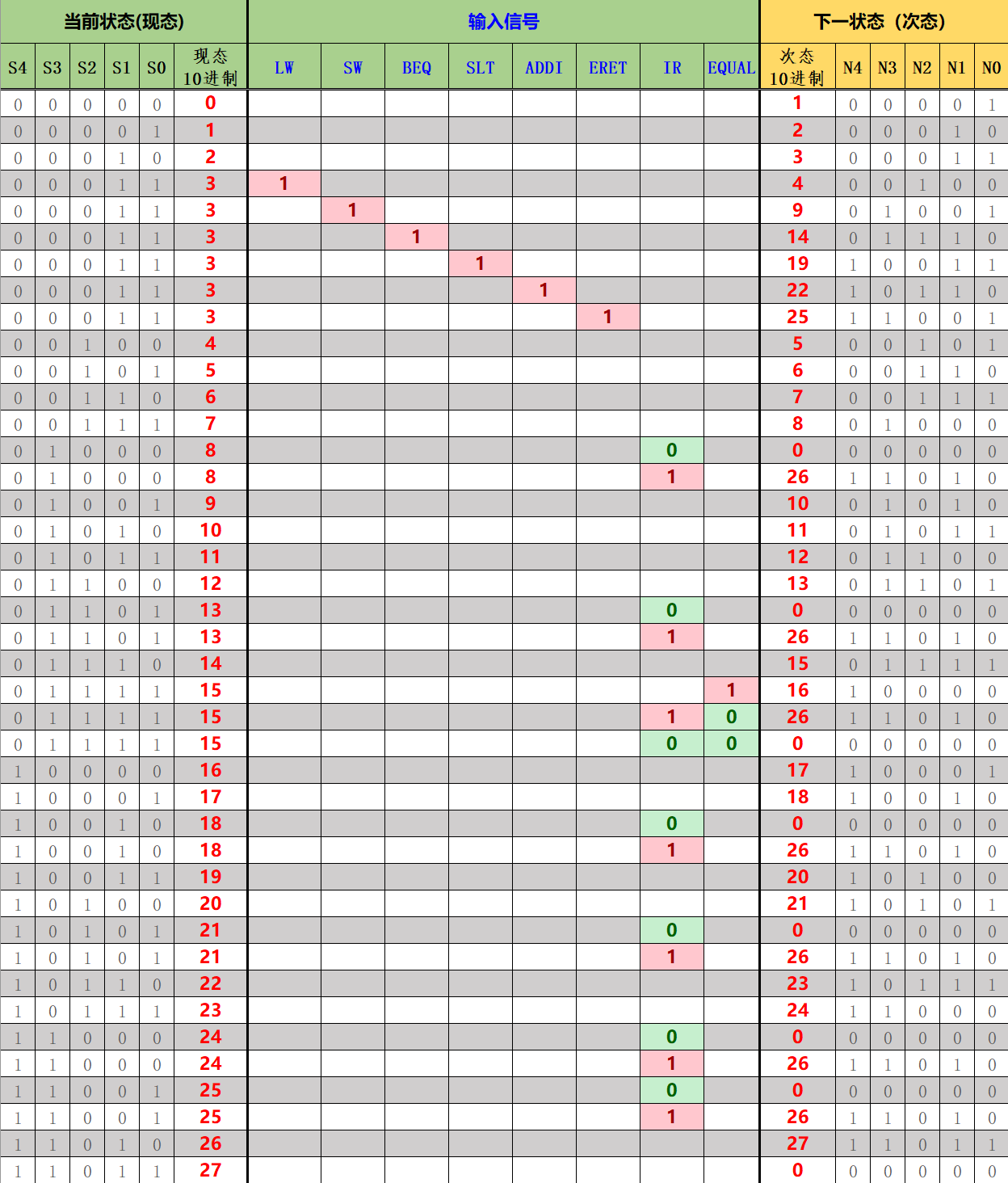


图 1‑18 单总线MIPS硬布线控制器状态机逻辑自动生成表

将表格自动生成的次态逻辑表达式导入电路图。生成最终生成电路图。由于电路图过大，不在这里贴出。

### 支持中断的现代时序硬布线控制器设计

本实验完成硬布线控制器框架连接即可。

由上一个实验可知，硬布线控制器状态机的前四个引脚为现态S4~S0，所以将状态寄存器的输出通过分线器连接到前四个引脚。

硬布线控制器状态机后面的引脚依次为LW、SW、BEQ、SLT、ADDI、ERET、IR、EQUAL，将对应引脚连接即可。

硬布线控制器状态机的输出端通过分线器连接到状态寄存器的输入端。

最终的电路图如下。

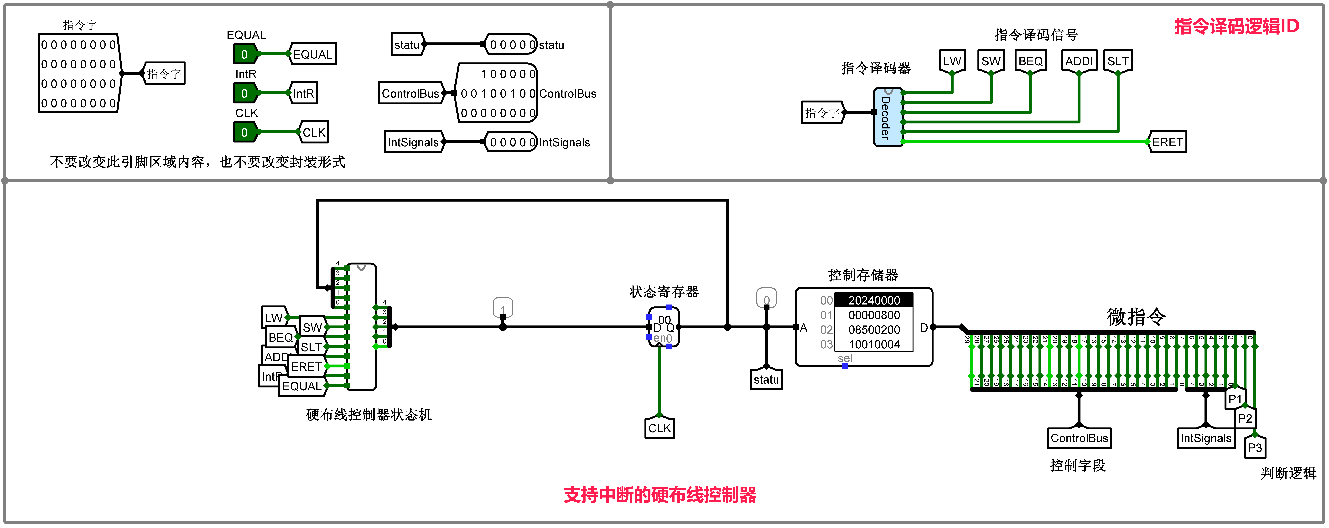


图 1‑19硬布线控制器电路图

## 实验步骤

1. 下载资料包到本地，使用logisim打开MipsOnBusCpu-1.circ文件。
2. 根据头歌要求，在计算机组成原理PPT上查找对应知识。
3. 在一些关卡使用提供的excel表格，填写完成后将表达式导入对应电路图。
4. 在logisim上实现最初的版本。
5. 在头歌上进行测评，如果没通过，则研究测试集，同时在logisim中进行测试，修改电路。
6. 如果遇到了无法解决的问题则向同学和老师请教。

## 故障与调试

### 微程序控制器出错

**故障现象：**在头歌上提交代码时报错。

**原因分析：**寄存器为上升沿触发，导致经过一个时钟脉冲后，寄存器中的数据仍然没有改变，在下一个时钟脉冲时才改变，导致测试结果出错。

**解决方案：**将寄存器改为下降沿触发。

### 接口处数据传输问题

**故障现象：** 进入中断处理程序开中断后，中断请求无法输出。

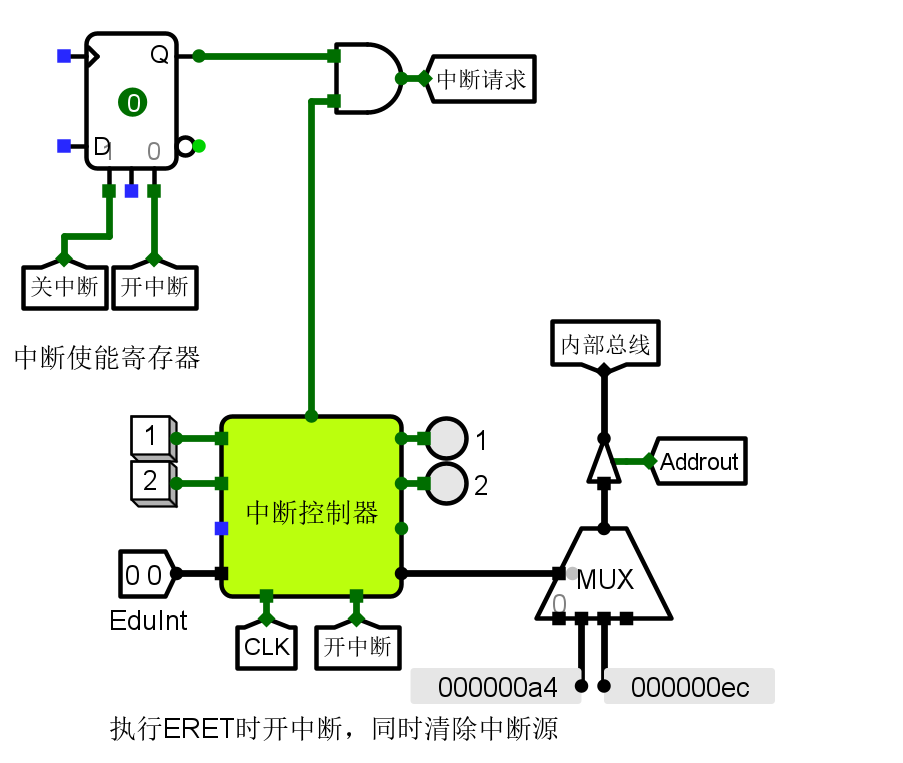


图1‑20 故障电路图

**原因分析：** 中断使能寄存器应输出低电平以实现在中断处理中屏蔽其他中断。

**解决方案：** 将触发器非值连接到与门。

## 测试与分析

测试用例为sort-5-int.hex程序，将其载入电路的RAM后，RAM储存内容如下图所示。

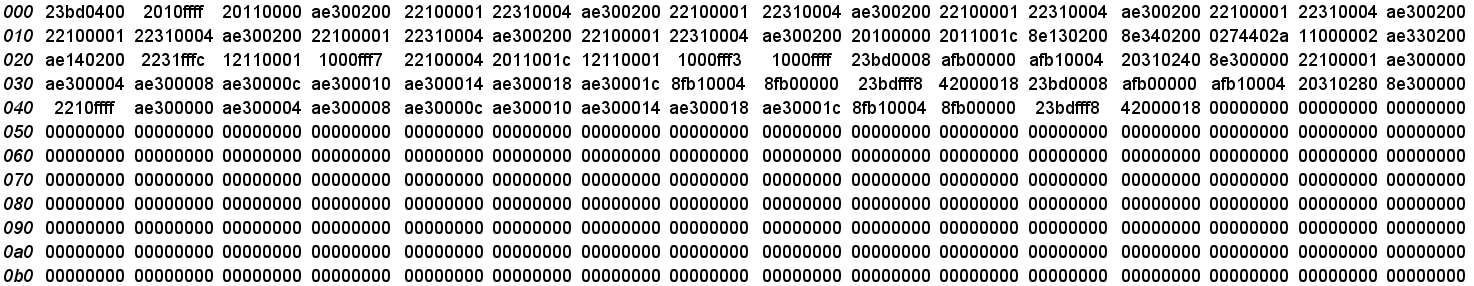


图 1‑21 CPU测试用例

## 实验总结

设计了基于MIPS支持中断的现代时序单总线CPU，实现了 MIPS 指令译码器，支持中断的微程序入口查找逻辑电路，支持中断的微程序条件判断电路，支持中断的微程序控制器和单总线 CPU，支持中断的现代时序硬布线状态机和控制器。补充了中断功能部件，完成了中断优先级处理和中断识别的逻辑。

## 实验心得

1. 在完成了计算机组成原理的所有课程设计之后，我对从书本中学习到的知识有了更加深刻的理解，其中最让我印象深刻也对我启发最大的就是CPU实验。在刚开始学习这门课时，老师说我们最后会自己设计CPU，当时我感觉这是一件完全超出自己能力的事情，但是在经过学习计算机组成原理并完成了一系列课程设计之后感觉设计CPU也没有那么难。实验与PPT是高度相关的，在做实验的过程中，我把PPT又认真看了一遍，之前有一些没有完全理解的地方也都搞懂了。由于这次实验可以看到测试集，降低了很大一部分难度，在写微程序自动生成表时，可以直接从测试集中得到各个字段的值。
2. 最后是一点建议。如果我没记错的话，课设的资料包是在我们的第一次实验课已经开始一段时间之后才提供给我们的，并且里面有好几个版本的资料。我一开始还因为下错了资料包浪费了很多时间。希望之后能在实验开始前将这些东西说明清楚。并且CPU中断实验有一关存在寄存器的bug需要更换logisim的版本，这个问题在头歌上有说明，但是我在做到这一关才看到，最后我用老版本的logisim将前面的实验全部重做一遍才解决（也许这并不是必须的），希望之后老师能提前将这些问题说明一下。

|  |
| --- |
| 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: 王文涛** |
| 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:** |