***2022***



**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： | CS2003 |
| 学 号： | U202015360 |
| 姓 名： | 胡沁心 |
| 电 话： | 13656789548 |
| 邮 件： | [13656789548@163.com](mailto:13656789548@163.com) |
| 完成日期： | 2022-07-08 |



# CPU设计实验

## 设计要求

本实训项目帮助学生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MIPS 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eret的支持，需要中断服务程序配合。

## 方案设计

### 微程序实现

1. 指令译码器设计

表1-1 片引脚与功能描述

|  |  |  |  |
| --- | --- | --- | --- |
| 引脚 | 输入/输出 | 位宽 | 功能描述 |
| IR | 输入 | 32 | MIPS指令字 IR |
| 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 |
| SLT | 输出 | 1 | 当前指令为slt指令时输出为1 |
| OtherInstr | 输出 | 1 | 当前指令为其它指令时输出为1 |

Mips指令长度为固定的32位。

表1-2 Mips的指令格式

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 6 | 5 | 5 | 5 | 5 | 6 |
| op | rs | rt | rd | shamt | funct |

每个指令对应不同的指令编码。

表1-3 指令编码

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| sw | lw | beq | addi | slt |
| op=011011 | op=100011 | op=000100 | op=001000 | op=000000  funct=011010 |

如果不属于以上5个指令，则进行中断处理。

电路图：

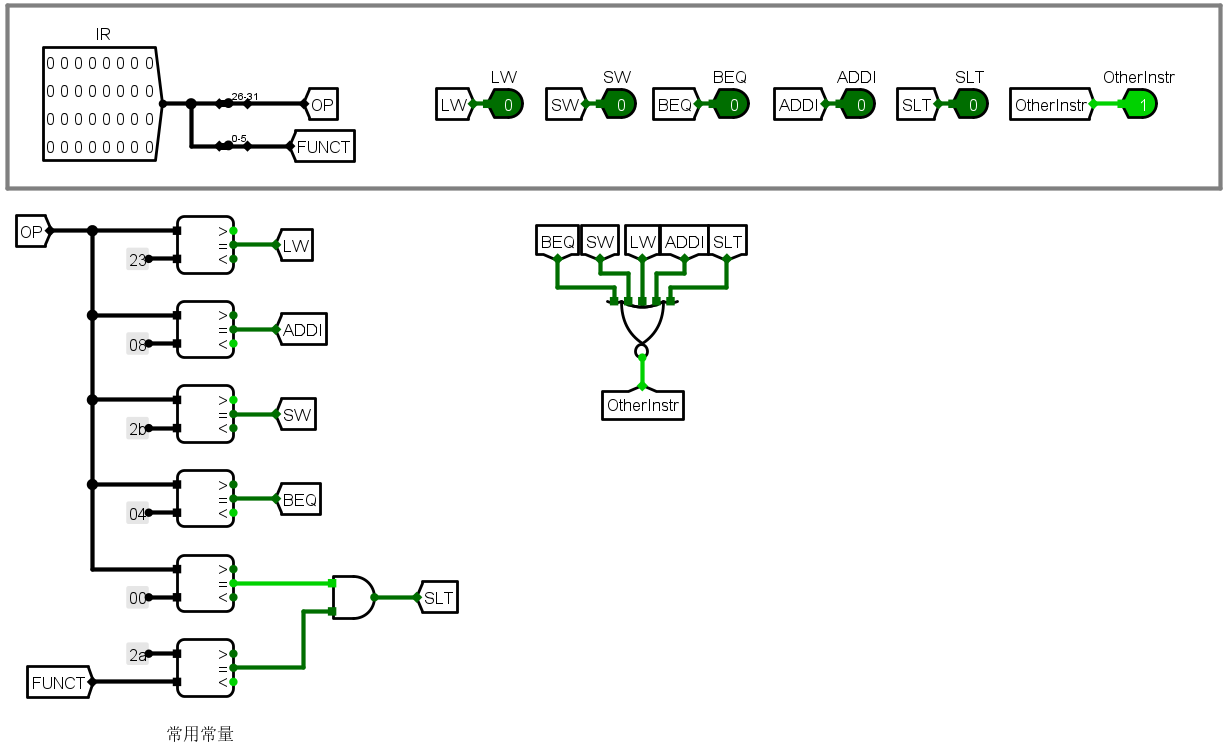


图1-1 指令译码器电路图

1. 微程序入口查找逻辑设计

表1-4 片引脚与功能描述

|  |  |  |  |
| --- | --- | --- | --- |
| 引脚 | 输入/输出 | 位宽 | 功能描述 |
| LW | 输入 | 1 | LW 指令译码信号 |
| SW | 输入 | 1 | SW指令译码信号 |
| BEQ | 输入 | 1 | BEQ指令译码信号 |
| ADDI | 输入 | 1 | ADDI指令译码信号 |
| SLT | 输入 | 1 | SLT指令译码信号 |
| OtherInstr | 输入 | 1 | ERET指令译码信号 |
| S4~S0 | 输出 | 1 | 微程序地址入口地址 |

每个指令对应不同的微程序入口地址：



图1-2 微程序入口地址表

得到逻辑表达式：

S4= SLT+ADDI+ERET

S3= SW+BEQ+ERET

S2= LW+BEQ+ADDI

S1= BEQ+SLT+ADDI

S0= SW+SLT+ERET

将逻辑表达式填入analyze circuit中的expression中，自动生成电路

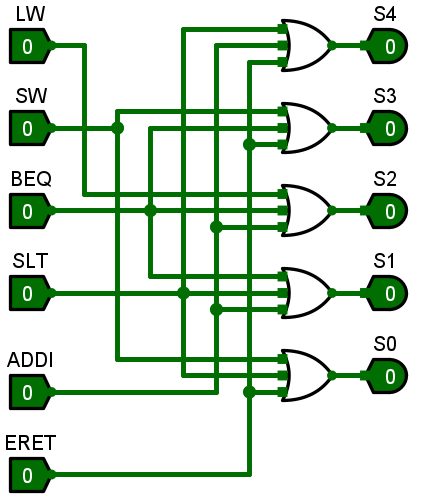


图1-3 微程序入口查找逻辑电路

1. 条件判别测试逻辑设计

表1-5 片引脚与功能描述

|  |  |  |  |
| --- | --- | --- | --- |
| 引脚 | 输入/输出 | 位宽 | 功能描述 |
| P0 | 输入 | 1 | 判别测试位，为1表示要根据指令功能进行微程序分支 |
| P1 | 输入 | 1 | 判别测试位，为1表示要根据equal标志进行微程序分支 |
| P2 | 输入 | 1 | 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 |
| equal | 输入 | 1 | 条件状态位，表示运算相等 |
| IntR | 输入 | 1 | 中断请求信号 |
| S2~S0 | 输出 | 1 | 后续地址多路选择控制信号 |

不同的输入条件对应不同的选择序号：

1. P0=1，跳转到微程序入口。

2. P1=1：若如果EQUAL=1，跳转到beq分支，否则：若IntR=1，跳转到中断，否则跳转到取指。

3. P2=1：若IntR=1，跳转到中断，否则跳转到取指。

4. 其余情况都跳转到顺序地址。



图1-4 判别测试位对应的转移地址

表1-6 不同跳转地址对应的序号

|  |  |
| --- | --- |
| 跳转地址 | S2~S0 |
| 循序地址 | 000 |
| 微程序入口 | 001 |
| beq分支 | 010 |
| 中断入口 | 011 |
| 取指入口 | 100 |

根据表1-6和图1-4可以得到以下对应关系：

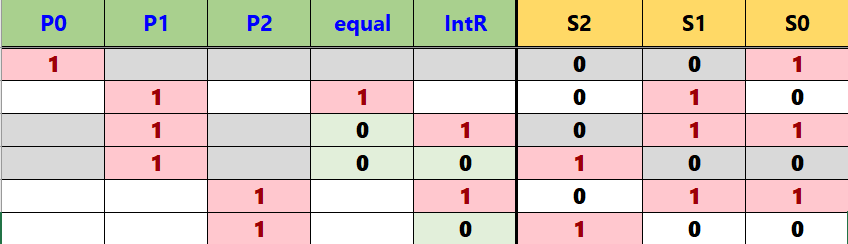


图1-5 判别测试位与目标地址序号的对应真值表

得到逻辑表达式：

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

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

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

将逻辑表达式填到expression中，自动生成电路

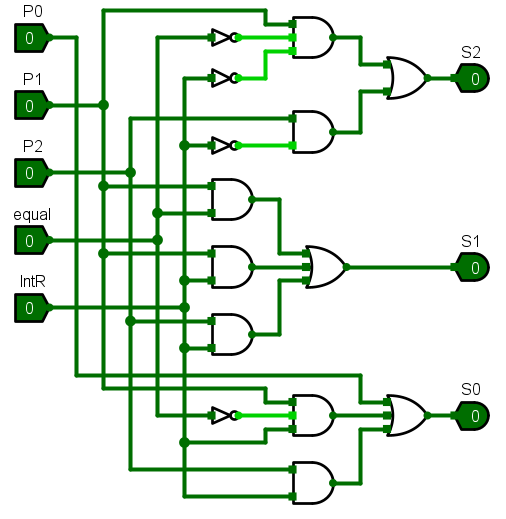


图1-6 条件判别测试逻辑电路

1. 微程序控制器设计

表1-7 片引脚与功能描述

|  |  |  |  |
| --- | --- | --- | --- |
| 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 |
| clk | 输入 | 1 | 时钟信号 |
| mAddr | 输出 | 5 | 当前微指令地址 |
| ControlBus | 输出 | 22 | 微指令字的控制字段 |
| IntSignals | 输出 | 5 | 微指令字的中断控制信号 |

将mips微程序的控制信号填入excel表中，生成16进制编码。

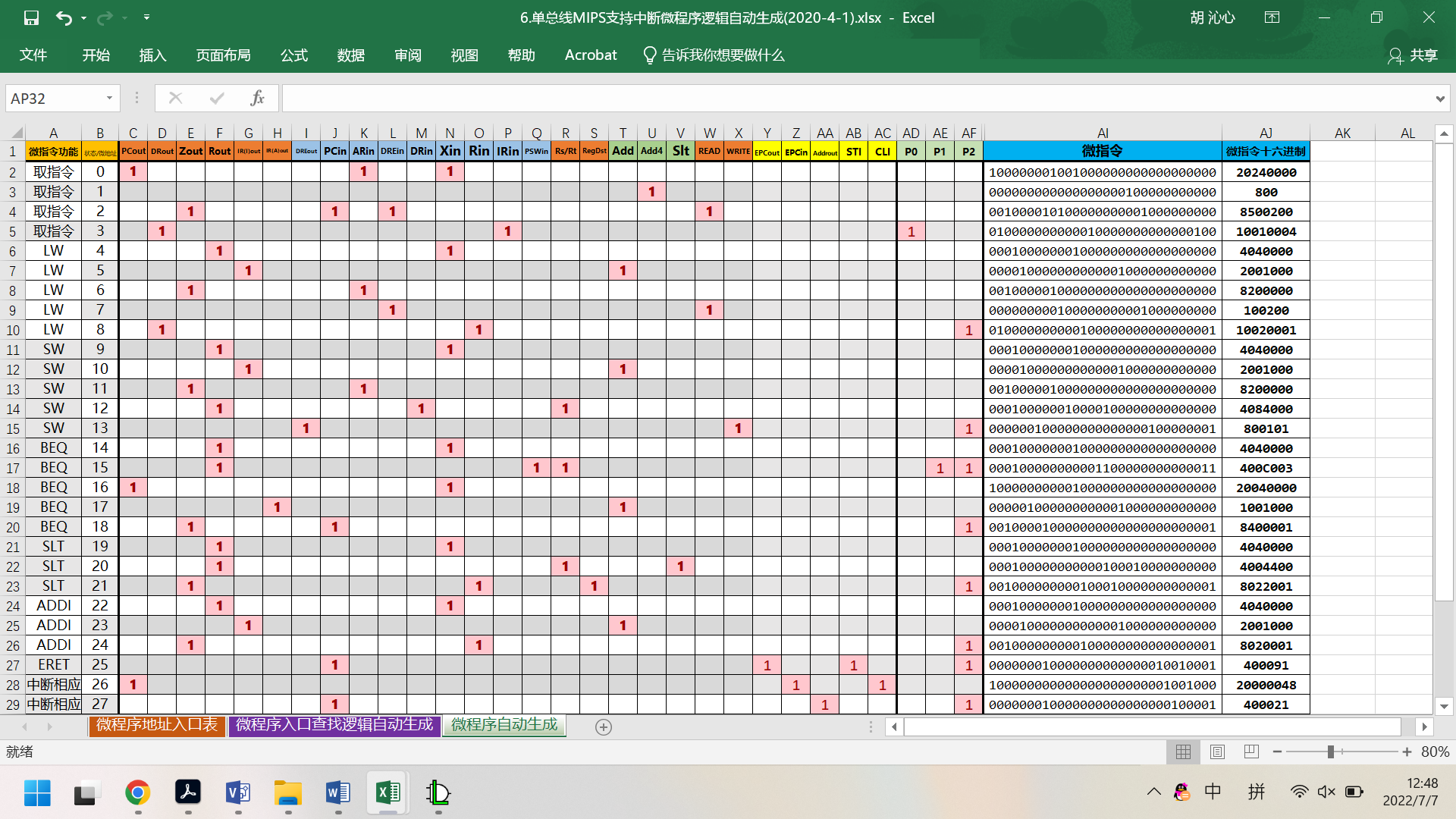


图1-7 微指令编码表

1)指令译码逻辑：

指令字经过指令译码器电路得到的6个指令信号，作为取指周期结束后进入分支的选择。

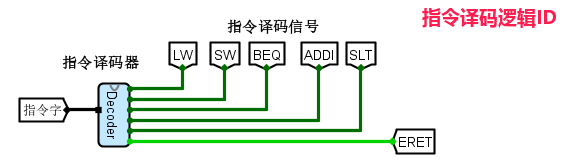


图1-8 指令字译码电路

2)地址序号生成逻辑：

多路选择器输出下个地址的序号。入口信号说明如下：

1.顺序地址为现地址+1。

2.入口地址为6个指令信号经微程序入口查找逻辑电路处理后的5位输出数据。

3.从图1-8可以得到beq分支的入口为0x10，中断入口为0x1a，取指入口为0x0。

4.多路选择器的选择位：判别位信号经过条件判别测试逻辑电路处理后的3位数据。

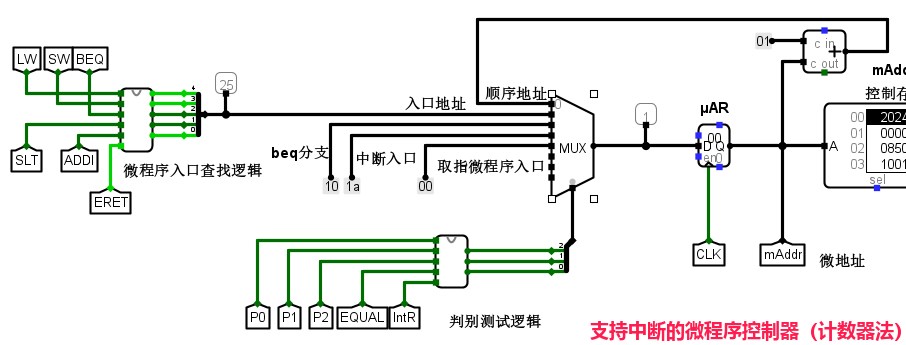


图1-9 微程序地址转移电路

3)微程序指令编码逻辑：

图1-8含有所有指令的十六进制表示。把十六进制编码按顺序填到ROM中。ROM前连接一个寄存器，按时钟输出执行的地址的编号，存放下一地址的编号。控制存储器输出30位微指令。

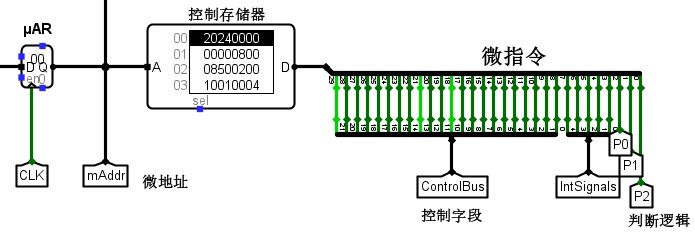


图1-10 微程序指令编码电路

### 硬布线实现

1. 指令译码器设计

与微程序部分的指令译码器相同。

1. 硬布线状态机设计

表1-8 片引脚与功能描述

|  |  |  |  |
| --- | --- | --- | --- |
| 引脚 | 输入/输出 | 位宽 | 功能描述 |
| S4~S0 | 输入 | 1 | 现态 |
| LW | 输入 | 1 | LW 指令译码信号 |
| SW | 输入 | 1 | SW指令译码信号 |
| BEQ | 输入 | 1 | BEQ指令译码信号 |
| ADDI | 输入 | 1 | ADDI指令译码信号 |
| STL | 输入 | 1 | SLT指令译码信号 |
| ERET | 输入 | 1 | ERET指令译码信号 |
| IR | 输入 | 1 | 中断请求信号 |
| EQUAL | 输入 | 1 | 条件反馈信号 |
| N4~N0 | 输出 | 1 | 次态 |

根据指令编码器的输出信号，将现态、输入信号和次态填入excel表中，得到以下状态转移逻辑。

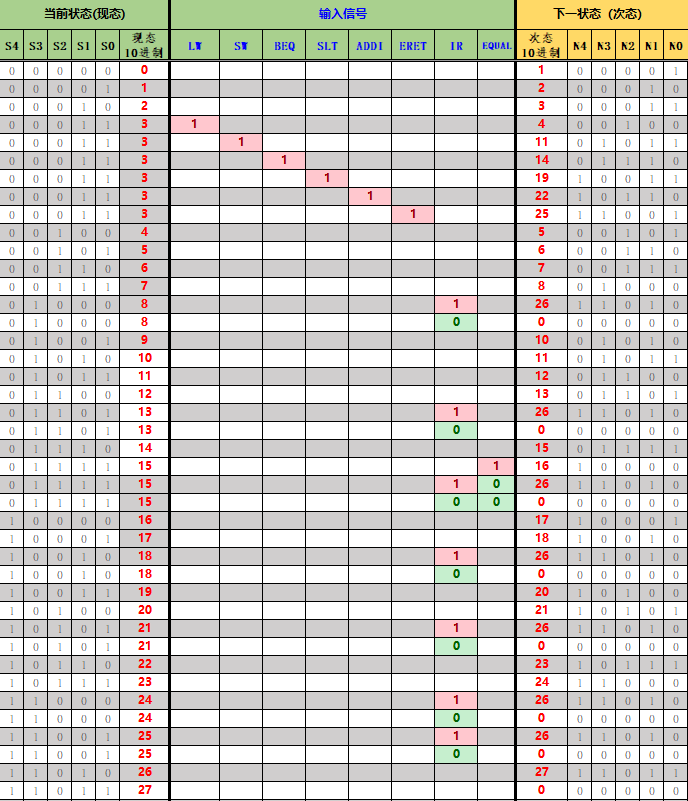


图1-11 状态转移表

excel自动生成逻辑表达式，将表达式填入expression中，自动生成电路。

1. 硬布线控制器设计

表1-9 片引脚与功能描述

|  |  |  |  |
| --- | --- | --- | --- |
| 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 |
| clk | 输入 | 1 | 时钟信号 |
| statu | 输出 | 5 | 现态 |
| ControlBus | 输出 | 22 | 微指令字的控制字段 |
| IntSignals | 输出 | 5 | 微指令字的中断控制信号 |

1)指令译码逻辑：

与微程序部分相同。

2)状态转移逻辑：

5位现态信号和7位判别位经硬布线控制状态机电路，输出5位次态信号。

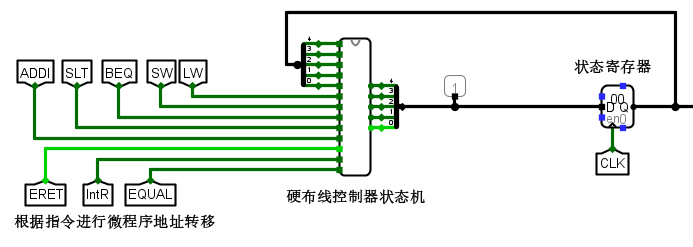


图1-12 硬布线状态转移电路

3)硬布线指令编码逻辑：

与微程序指令编码相同，statu输出5位现态信号。

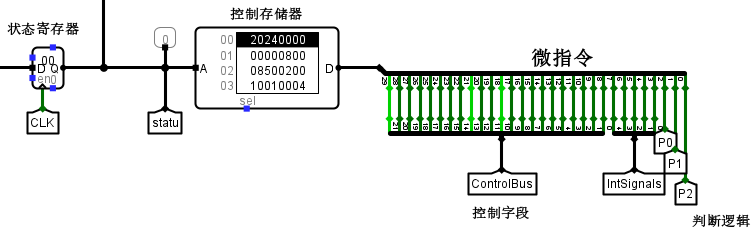


图1-13 硬布线指令编码电路

### 单总线CPU+中断实现

表1-10 片引脚与功能描述

|  |  |  |  |
| --- | --- | --- | --- |
| 引脚 | 输入/输出 | 位宽 | 功能描述 |
| Instr | 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 |
| mAddr | 输出 | 5 | 当前微指令地址 |
| ControlBus | 输出 | 5 | 微指令字的控制字段 |

中断实现：

1.首先将控制器输出的中断控制信号分成5个微操作

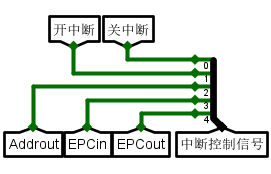


图1-14 中断控制信号

2.设计断点保存电路，实现断点的保存和输出。当EPCin为高电平时断电存入寄存器，EPCout为高电平时输出断点到内部总线

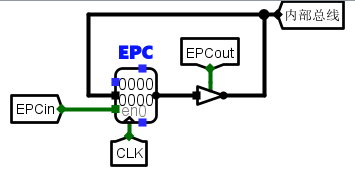


图1-15 断点保存电路

3. 设计中断控制电路。中断使能寄存器IE在可以接收中断时值为 0，否则为 1。IE输出信号和中断信号都为高电平时输出CPU的中断请求信号。中断控制器输出的中断号接到多路选择器的选择位，两个数据位是两个中断程序的地址，当Addrout为高电平时将中断程序地址输入到内部总线，执行中断程序。

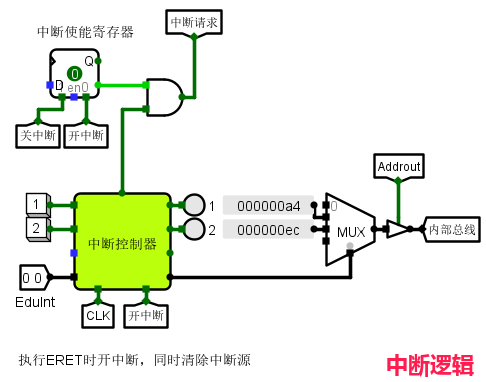


图1-16 中断控制电路

单总线CPU总体结构图如下 (微程序)。微程序封装可以改成硬布线封装。

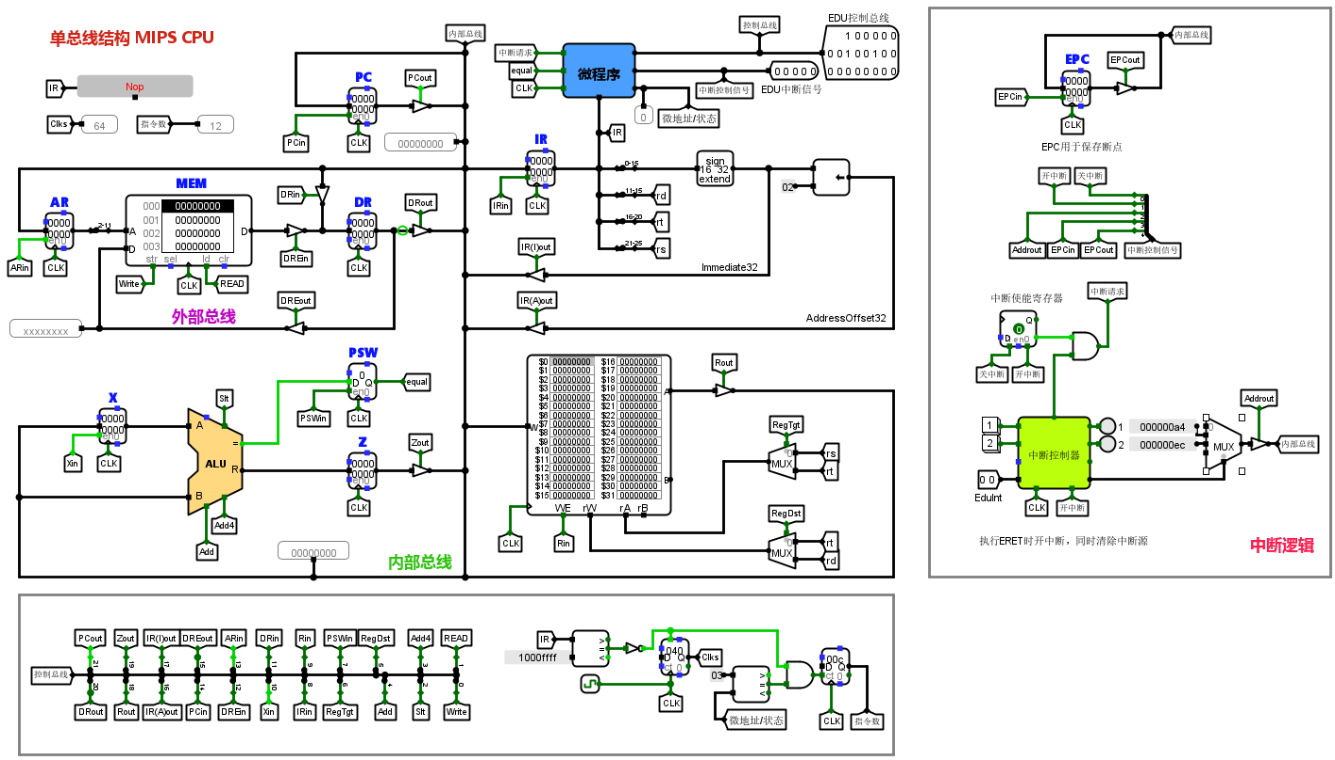


图1-17 总体结构图

## 实验步骤

1. 设计指令译码器电路。
2. 设计微程序入口查找逻辑电路和微程序入口查找逻辑电路。利用excel自动生成逻辑表达式，再用logisim的analyze circuit功能自动生成电路。
3. 设计微程序控制器。利用excel生成微指令编码，写入控制存储器。用指令译码器封装、微程序入口查找逻辑封装、微程序入口查找逻辑封装和部分电路组成微程序控制器。
4. 设计硬布线状态机。利用excel生成逻辑表达式，再用logisim的analyze circuit功能自动生成电路。
5. 设计硬布线控制器。利用excel生成微指令编码，写入控制存储器。用指令译码器封装、硬布线状态机封装和部分电路组成硬布线控制器。
6. 设计单总线CPU+中断电路的中断逻辑硬件实现部分。
7. 测试电路。

## 故障与调试

### 中断处理问题

**故障现象：**beq指令中断跳转问题

**原因分析：**beq的第二条微指令，如果equal=0则结束，当前指令是微程序的最后一条微指令，可能需要进行中断响应，但结束后没有判断是否要进行中断处理，真值表错误，如图1-18所示。

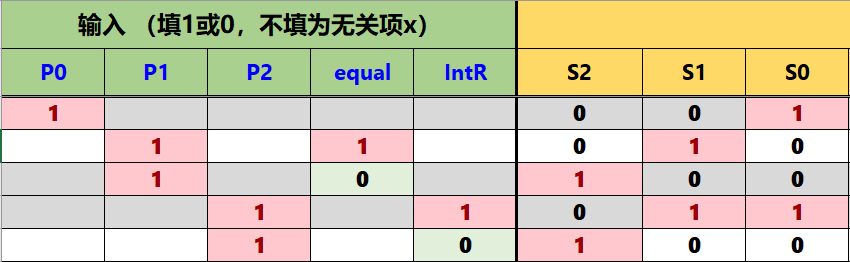


图1-18 错误真值表

**解决方案：** equal=0的情况分为两种，如果IntR=1，跳转到中断入口，否则转到微程序入口。真值表改为图1-19。

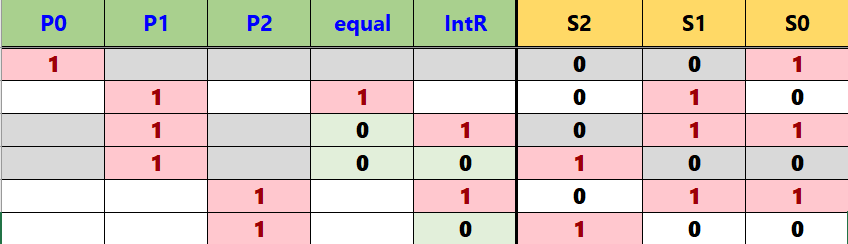


图1-19 正确真值表

### 控制字段输出错误

**故障现象：**指令控制字段出错

**原因分析：**beq的第二条微指令中，P2没有置1，如图1-20。当执行到该条指令时，如果equal=0，当前指令是微程序的最后一条微指令，P2为end判别位，应该置1。



图1-20 错误微指令

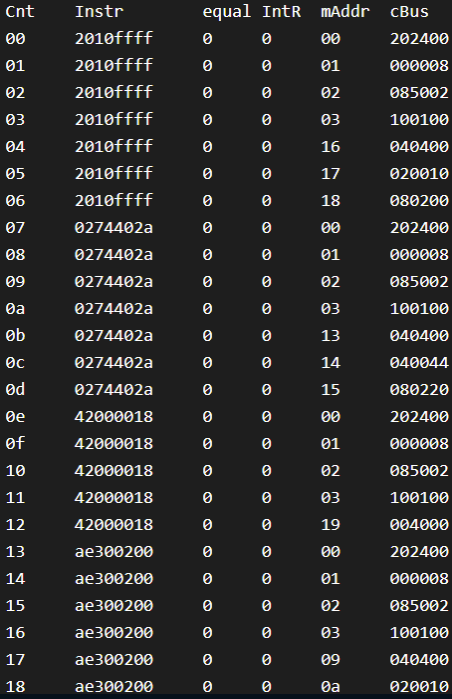
**解决方案：** 将第16条指令的P2位改为1

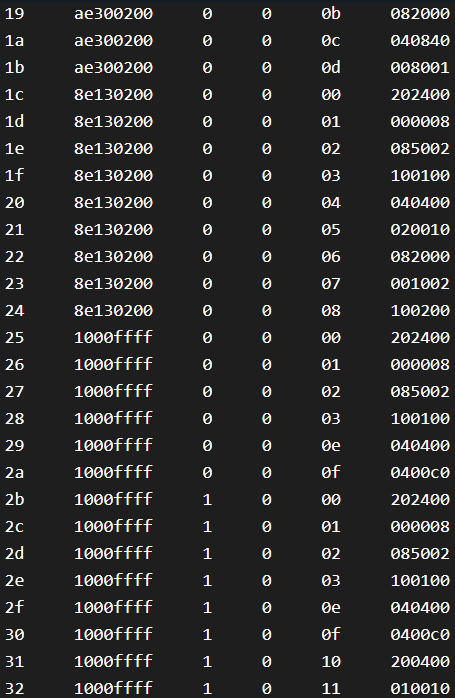
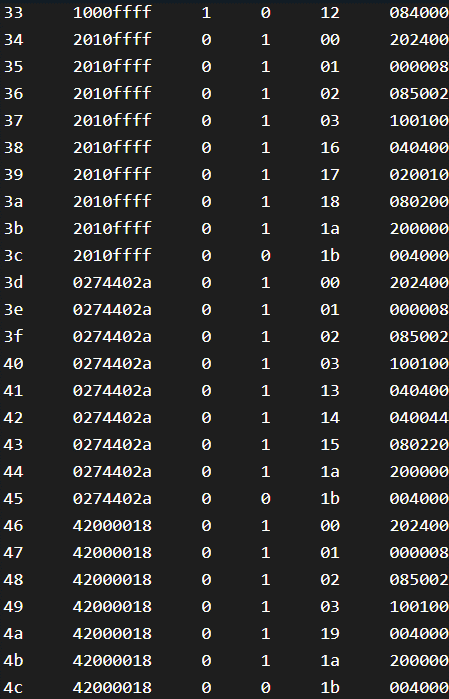


图1-21 正确微指令

## 测试与分析

Educoder上的单总线CPU+中断的测试数据，全部通过。



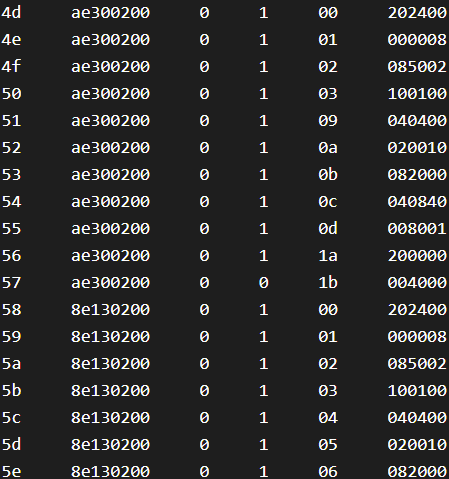
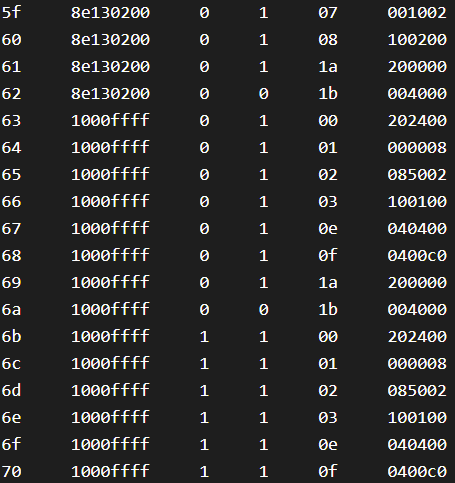
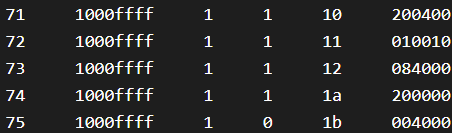
  

图1-22 测试用例

## 实验总结

本次实验主要完成了如下几点工作：

1. 设计了指令译码器电路
2. 设计了微程序入口查找逻辑电路
3. 设计了条件判别测试逻辑电路
4. 实现了支持中断的现代时序微程序控制器
5. 设计了硬布线状态机
6. 实现了支持中断的现代时序硬布线控制器
7. 实现了中断相关硬件模块的设计

## 实验心得

1. 熟悉了logisim的各个部件，尤其是寄存器、ROM、计数器和多路选择器的使用，学会了使用工程化方法自动生成组合逻辑电路。
2. 熟悉了微程序和硬布线的现代时序的控制器的设计方法。微程序现代时序设计是通过条件判别位来控制输出的地址序号。硬布线现代时序设计是设计一个有限状态机，状态机的次态由输入信号(指令类型)和现态共同决定。
3. 了解了CPU处理中断的方式以及单总线CPU上简单的支持单级中断的硬件如何设计。处理单级中断主要是几个步骤：关中断， 保存断点，判断中断类型，中断程序地址送到内部总线，中断处理完后断点地址送回内部总线，开中断。开关中断对应的硬件是中断使能寄存器IE，存取断点地址对应的硬件是异常程序地址寄存器 ，判断中断类型取中断服务程序地址对应的硬件是中断控制器。
4. 整个实验下来，动手实践的过程让我对理论课堂内的知识有了更好的掌握， 也让我对计算机的硬件组成有了一个更完整更深入的认识。同时也让我回顾了上学期的数字逻辑课程，让我了解到硬件方面的知识对理解计算机内部结构的重要性。

对课程的建议：汇编、计算机组成原理和计算机系统基础三门课的课程内容有很多重叠，如果能整合一下效率会更高。

|  |
| --- |
| 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: 胡沁心** |
| 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:** |