***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： | CS1906 |
| 学 号： | U201915116 |
| 姓 名： | 廖翔 |
| 电 话： | 15027278369 |
| 邮 件： | liao1120x@gmail.com |
| 完成日期： | 2021-12-14 |



目 录

[1 CPU设计实验 2](#_Toc90801379)

[1.1 设计要求 2](#_Toc90801380)

[1.2 方案设计 4](#_Toc90801381)

[1.3 实验步骤 13](#_Toc90801382)

[1.4 故障与调试 13](#_Toc90801383)

[1.5 测试与分析 14](#_Toc90801384)

[2 总结与心得 15](#_Toc90801385)

[2.1 实验总结 15](#_Toc90801386)

[2.2 实验心得 15](#_Toc90801387)

[参考文献 16](#_Toc90801388)

# 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)

利用变长指令周期三级时序系统构造硬布线控制器，包括如下几项：

* RISC-V 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
* 变长指令周期--硬布线控制器设计
* 变长指令周期—单总线CPU设计

图示, 示意图

描述已自动生成

图1-1 单总线RISC-V三级时序CPU结构图

### RISC-V现代时序中断机制实现

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，包括以下几项：

* RISC-V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

图示, 示意图

描述已自动生成

图1-2 单总线RISC-V现代时序+中断CPU结构图

## 方案设计

### 单总线RISC-V CPU设计(变长指令周期3级时序)

* RISC-V 指令译码器设计

输入:32位指令IR

输出:指令译码信息。如BEQ代表BEQ指令。

实现方式:通过查阅具体指令，从取出的FUNCT3和FUNCT7中选择合适的进行比较，若匹配就说明是该指令。这里以sw指令为例，查阅RISC-V官方文档，可知sw指令如下：

表格

描述已自动生成

图1-3 RISC-V sw指令

因此，将OP部分与FUNCT\_3拼接为OP\_Funct3，然后前五位与图中01000，即8H，后三位与010，即2H比较，若匹配，则说明是SW指令。

电路图:

图示, 示意图

描述已自动生成

图1-4 译码器电路图

* 时序发生器FSM设计

输入:当前状态

输出:次态

实现方式:根据课本上的三级时序状态转换图，填写EXCEL表格，再将最后结果复制进logisim中自动生成电路。具体状态图和EXCEL表格如下:

图示

描述已自动生成

图1-5 变长指令周期三级时序状态机

表格

描述已自动生成

图1-6 单总线RISC-V三级时序产生器组合逻辑自动生成表

* 时序发生器输出函数设计

输入:当前状态

输出:当前状态对应的机器周期和节拍电位

实现方式:同样的，根据课本上的时序状态转换表填写对应的EXCEL表，根据结果即可自动生成逻辑电路。具体状态表和EXCEL表格如下:

表格

描述已自动生成

图1-6 时序发生器状态转换表

表格, 日历

中度可信度描述已自动生成

图1-7 单总线RISC-V三级时序产生器输出函数自动生成表

* 硬布线控制器组合逻辑单元

输入:机器周期和节拍电位

输出:控制信号

实现方式:按照课本，根据每条指令所处的机器周期和对应的节拍电位，填写EXCEL表格中的控制信号输出，然后将结果在logisim中自动生成电路。EXCEL表格如下:

游戏的屏幕截图

中度可信度描述已自动生成

图1-8 单总线RISC-V控制信号表

* 变长指令周期--硬布线控制器设计

输入:指令字,时钟信号,EQUAL信号

输出:控制总线输出

实现逻辑:当前状态和指令译码输出作为状态机的输入，输出的次态作为下一次的输入，根据状态即可控制各个节拍信号的输出。

电路图:

图示

中度可信度描述已自动生成

图1-9 单总线RISC-V硬布线控制器电路图

* 变长指令周期—单总线CPU设计

这部分不需要我们做什么，只需要把sort.hex中内容导入RAM中，然后运行检查最后的结果即可，同时也能够更加直观地看到整个RISC-V CPU的组成和运行逻辑，图已在前面附上，这里不再贴。

### RISC-V现代时序中断机制实现

* RISC-V指令译码器设计

这部分与4.2.1中的译码器设计相同。

* 支持中断的微程序入口查找逻辑

输入:机器指令译码信号

输出:微程序入口地址

实现方式:同样的，根据课本上各机器指令对应的入口填写EXCEL表格，将最后的结果到logisim生成电路即可。具体转换图和EXCEL表如下:

图示

描述已自动生成

图1-10 指令执行状态转换图

需要注意到这个转换图中没有加上中断入口。

表格

描述已自动生成

图1-11 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

输入:微程序流程控制信号,相等关系信号EQUAL

输出:微程序地址转移控制信号

实现方式:首先要明确各个流程控制信号的含义。

译码测试位P0:其值为1代表微指令后续地址应该根据指令译码情况选择指令对应

的微程序入口地址，即微程序入口查找组合逻辑输出的地址。

判别测试位P1:其值为1代表要根据equal标志进行微程序分支。

结束微指令判别测试位P2:其值为1表示当前微指令是微程序的最后一条微指令，

这一条微指令执行完毕表示当前机器指令执行结束。

如果P2为1切控制器收到中断请求，没有别的分支需要执行，就可以将判别测试

位P2对应的微程序分支地址，也就是中断响应微程序的入口地址作为后续地址。

同时要理解微程序地址转移信号S(S2S1S0)，S=0代表是下一条地址，S=1代表是

取入口地址，S=2代表取BEQ指令且当前运算是相等的跳转地址，S=3代表是中断处理程序的第一个微程序地址，S=4代表回到取指令的第一条微程序。

根据以上几点，再加上地址转移逻辑中的译码器设计，可以填写出EXCEL表格如

下:

表格

描述已自动生成

图1-12 单总线结构现代时序(带中断)微程序条件判别逻辑自动生成表

* 支持中断的微程序控制器设计

输入:指令IR，EQUAL信号

输出:当前微程序地址和控制总线

实现方式:这个部分最重要也是最困难的地方就是解决如何根据微程序地址生成控制总线。首先，要根据微程序填写对应的控制信号输出表，这个部分在RISC-V版教材中就包含，可直接填写在EXCEL表中，下一步就是根据状态转换关系设计控制信号P1P2P3，P1P2P3的含义在之前已经介绍，那么设计的原则如下:

1. 如果当前为取指令的最后一条微程序，则需要标记P0表示要根据入口查找逻辑得到入口地址
2. 如果当前指令需要判断EQUAL分支，则需要标记P1表示需要跳转到处理EQUAL分支的微程序
3. 如果为当前指令的最后一条微程序，则需要标记P2表示结束。

根据以上三条原则，可以填写出EXCEL表格如下:

图表, 白板

中度可信度描述已自动生成

图1-13 单总线结构现代时序(带中断)微程序控制器微指令

得到控制存储器的内容后，下一步将各个入口地址连接即可得到最后的电路图如下:

图示, 示意图

描述已自动生成

图1-14 单总线结构现代时序(带中断)微程序控制器电路图

* 支持中断的微程序单总线CPU设计

这部分大体结构与变长指令周期单总线CPU相同，这里主要介绍中断逻辑的连

接。

查找中断地址:根据中断控制器的中断类型输出进行分支判断，使用多路选择器。

对于分支1则接上1号中断的中断入口地址的常量，对于分支2则接上2号中断的中断入口地址的常量。

IE寄存器:开中断的时候异步置为0，关中断的时候异步置为1，输出为与寄存器

保存内容相反的部分。

mEPC寄存器:与其他锁存器是同一种构造，具体而言，在写入信号为0时，通过使能端控制寄存器忽略时钟信号，不寄存输入端内容，输出接一个三态门，只有在EPC输出信号有效的情况下才进行输出，不能进行输出的时候无法输出到内总线。

观察这个不带中断的程序，可以发现在0x63处停止，第42行是最后一条正常程序，因此有一个中断入口指令为第41条。同样的， 可以看到中断的第一条指令为00810113，那么接着往下寻找该指令，可以看到第61行也有该指令，于是可以得到另一条中断入口指令为第59条。因为每一条指令由4个字节组成，所以第一个中断指令的地址为0XA4,第二个中断指令的地址为0XEC。

根据上述内容，可以连出中断逻辑如下:

图示

描述已自动生成

图1-15现代时序中断逻辑结构

* 支持中断的现代时序硬布线控制器状态机设计

输入:当前状态

输出:次态

实现方式:该部分需要在原来的现代时序硬布线控制器状态机上加上对中断的处理，需要注意到EXCEL表中IR一列需要专门拖出来。其实没什么需要太多注意的地方，就是在每一个有中断的地方跳转至中断执行入口，具体的EXCEL表如下:

应用程序, 表格, Excel

描述已自动生成

图1-16 支持中断的现代时序硬布线控制器状态机逻辑自动生成表

最后将结果导入logisim中自动生成电路即可。

* 支持中断的现代时序硬布线控制器设计

输入:指令字，EQUAL，时钟信号

输出:控制总线信号

实现方式:这个部分比较简单，直接将之前已经得到的微指令导入即可，主要思想与前面相同，这里不再赘述。电路图如下:

图示, 示意图

描述已自动生成

图1-17 支持中断的现代时序硬布线控制器结构图

## 实验步骤

1. 按照方案设计中的内容在logisim中连接电路图
2. 提交circ文件到Educoder平台进行评测，修改

## 故障与调试

### 中断实验中无法触发中断

**故障现象：**中断始终无法触发

**原因分析：**IE寄存器连线有误

**解决方案：**IE寄存器应该连接触发器当前状态的非值

### 支持中断的现代时序硬布线状态机

**故障现象：**将表格结果导入logisim仍然有误

**原因分析：**EXCEL表格中还有一栏，只是被隐藏较深

**解决方案：**EXCEL表格需要把IR一栏拖出，之前忽略了

## 测试与分析

### Educoder平台测评

测试样例全部通过。

### CPU运行结果

图示, 示意图

描述已自动生成

图1-18 CPU运行结果

# 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. 通关logisim的所有测试样例，通过动手实践的方式从数据表示，运算器，存储系统，CPU几个维度来深刻认识到计算机的组成原理。
2. 实现完各个部件后，最后组装成CPU，并能运行一个简单的冒泡排序算法。

## 实验心得

* 在数电实验的基础上再来进行组员实验，不需要再像数电实验是关心内部组

件的实现逻辑，更重要的是能将各个组件组装起来，满足我们所需要的功能。

* 在理论上了解一些部件的原理后，通过实验的方式能够更加加深我们的印象，

比如先行加法器，虽然书上的讲解已经非常详细，但是在logisim上将异或门，时钟信号等连接起来，将器件接口接出，让我们的理解更加深刻。

* 今年我们使用的是RISC-V版本的新CPU实验，虽然与MIPS区别不太大，但

也能看到大萝卜老师在制作EXCEL表，电路的辛苦，建议就是可以在circ文件中把一些东西讲的更清晰点，然后及时地更新Educoder吧，其余的我觉得都做的很好。

# 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: 廖翔 嵌入图片包含 文本  描述已自动生成签名图片** |
| 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:** |