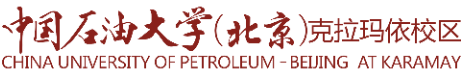
实 验 报 告





课 程 ：计算机组成原理

实验名称：单总线现代时序CPU设计实验

学 院 ：石油学院

专 业 ：数据科学与大数据技术

学 号 ：

姓 名 ：

2024年秋季学期

**一、实验目的**

掌握现代时序硬布线控制器、微程序控制器设计的基本原理，能在Logisim平台中基于单总线结构实现支持5条MIPS指令的现代时序处理器。

**二、实验内容**

本实验将依次设计：

① 指令译码器

② 地址转移逻辑

③ 微程序控制存储器

④ FSM状态机

等电路，以逐步实现：

⑤ 微程序控制器

⑥ 硬布线控制器

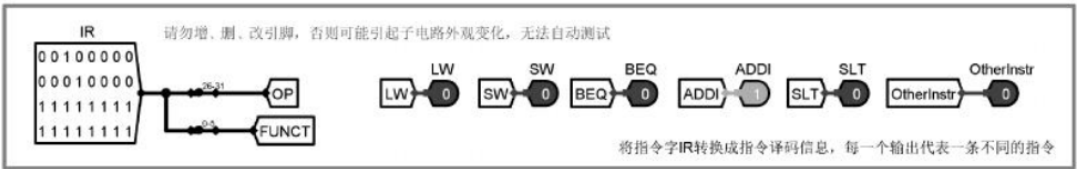
**三、实验步骤**

1.设计MIPS指令译码器

• MIPS指令译码器负责将定长的32位MIPS指令字翻译为指令译码信号，以此识别指令功能，具体电路封装与引脚功能描述如表5所示。



• 指令译码器电路框架如下图所示。



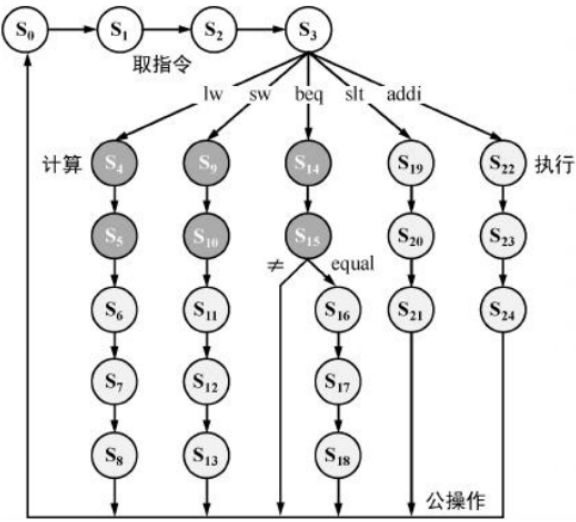
• 五种指令（LW、SW、BEQ、ADDI、SLT）及其操作编码的方式如右图所示。根据对应指令的OP字段和Funct字段的值，利用Logisim中的比较器组件和基本逻辑门电路输出各指令译码信号。

2.设计微程序控制器

• 对于图5.1所示的单总线数据通路，采用现代时序分析5条MIPS指令的指令周期及数据通路，可以得到图5.18所示的指令执行状态转换图。

• 图5.18中一个状态对应一个时钟周期，微操作控制信号的值仅与现态有关。控制存储器中的微指令可以与状态转换图中的状态一一对应，状态的编号值可以转换成微指令地址；而某一状态需要给出的微操作控制信号可以映射到对应微指令操作控制字段中的控制信号位；状态之间的切换关系可以对应微指令之间的执行顺序，用于设置判别测试字段和下址字段

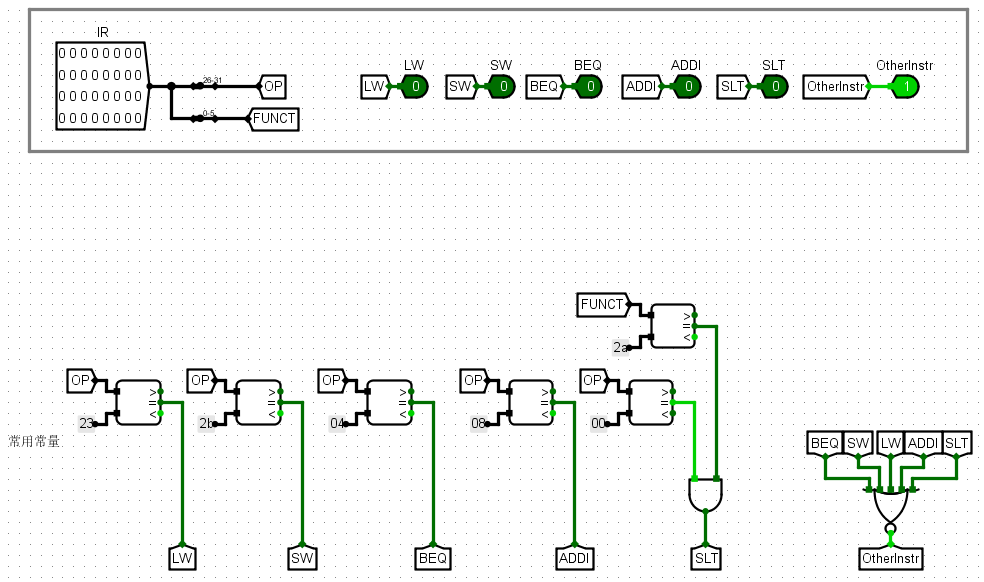
结合图5.18，参考教材第6章的原理设计微指令格式，构建5条MIPS指令的微程序。本实验只需要用slt指令替换原有的add指令



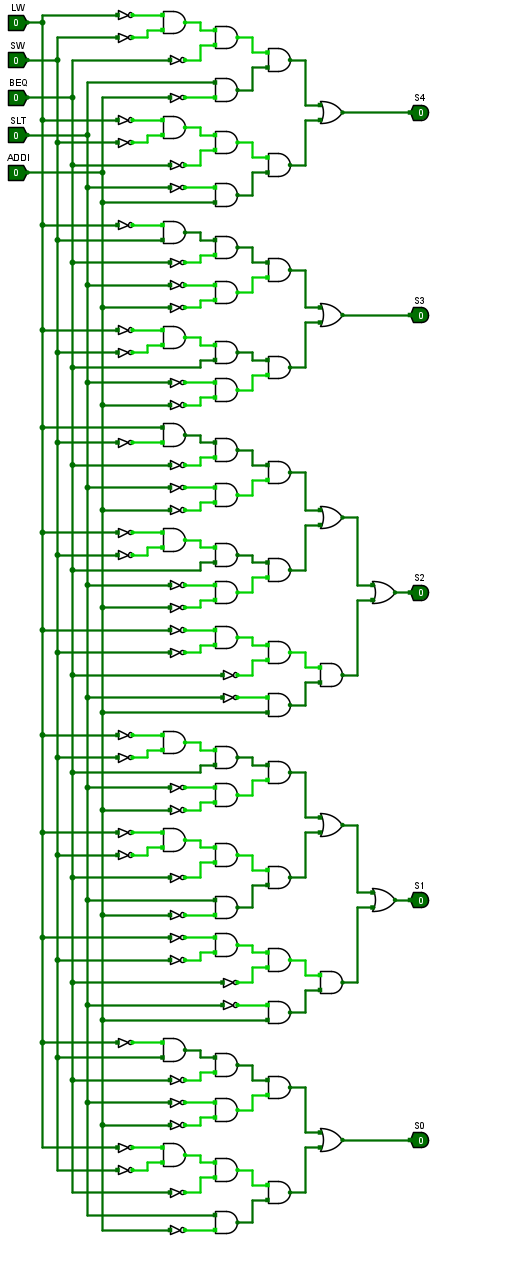
**四、实验结果**

1.MIPS指令译码器

连接译码器



2、单总线结构 MIPS CPU 结构的微程序入口地址产生逻辑图

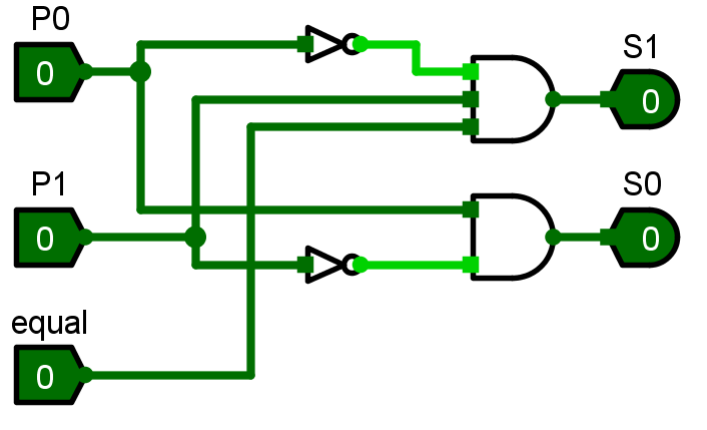


3、完成单总线结构 MIPS CPU 结构的微程序入口地址表格



4.设计微程序控制器

微程序条件判断测试产生逻辑

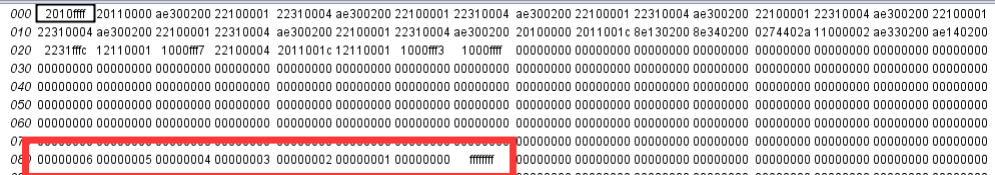






5.测试

1. 将数据文件sort-5.hex加载到主存储器中，以便进行排序算法的验证。
2. 将“3.单总线MIPS微程序地址转移逻辑自动生成”表格生成的微指令导入到控制存储器中。
3. 执行操作： 按下组合键ctrl+k以自动执行程序。由于冒泡排序完成的标志是进入无限循环状态，在这一点上时钟周期数和指令数将停止增加，此时应按下ctrl+k来暂停程序。
4. 查看结果： 检查外部存储器，可以发现数据已经按照降序排列。



**五、问题和感想**

在设计微程序控制器时，如何准确地根据指令周期及数据通路分析出指令执行状态转换图。首先，需要深入理解每条MIPS指令的执行周期，包括取指、译码、执行、访存和写回等阶段。

分析每条指令在数据通路中的流动，包括寄存器、ALU、内存等组件之间的数据交换。此外，将理论应用于实践，特别是在Logisim平台上实现电路设计，需要对工具有深入的了解和操作熟练度。

通过这次实验，我不仅加深了对现代时序CPU设计原理的理解，还学习到了如何在实际环境中应用这些理论知识。我发现理论与实践之间存在差距，实际操作中会遇到许多预料之外的问题，这要求我不断地调试和优化设计。