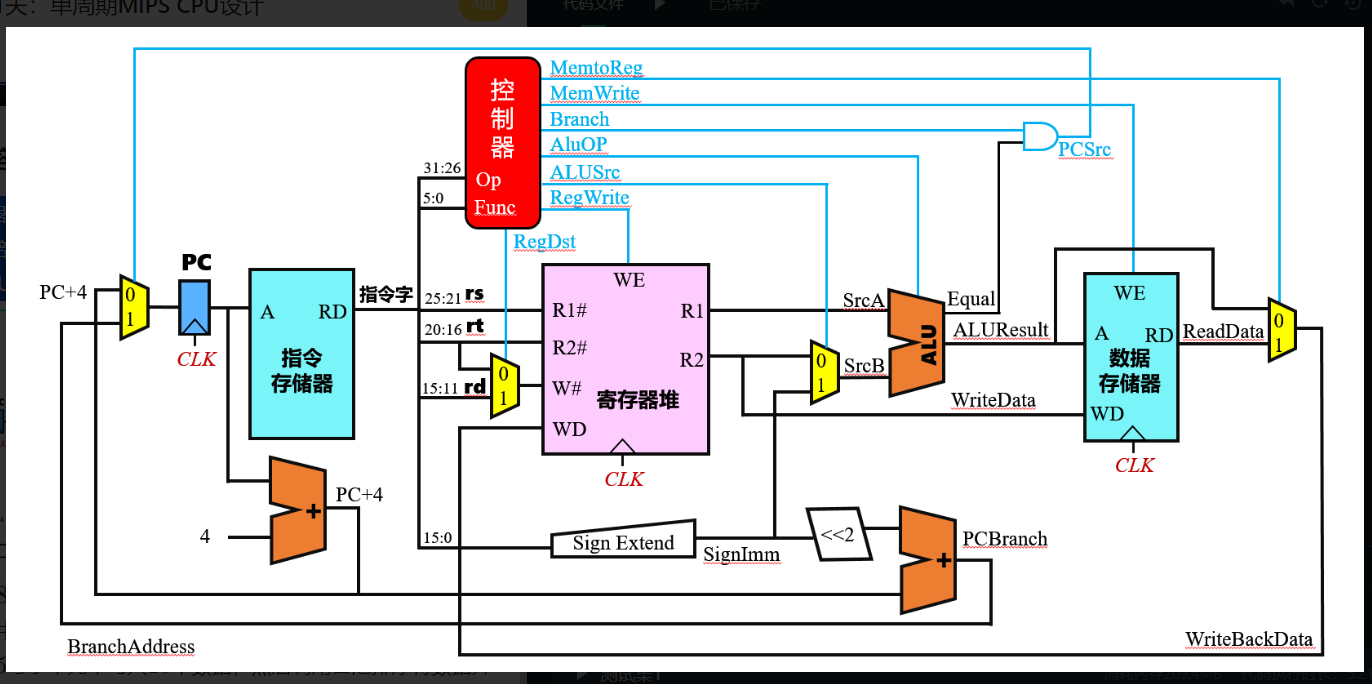
1. **实验目的**

学生掌握控制器设计的基本原理，能利用硬布线控制器的设计原理，在 Logisim 平台中设计实现 MIPS 单周期 CPU。

1. **实验内容**

利用运算器实验，存储系统实验中构建的运算器、寄存器文件、存储系统等部件以及 Logisim 中其它功能部件，构建一个32位 MIPS CPU 单周期处理器。数据通路如下图所示：

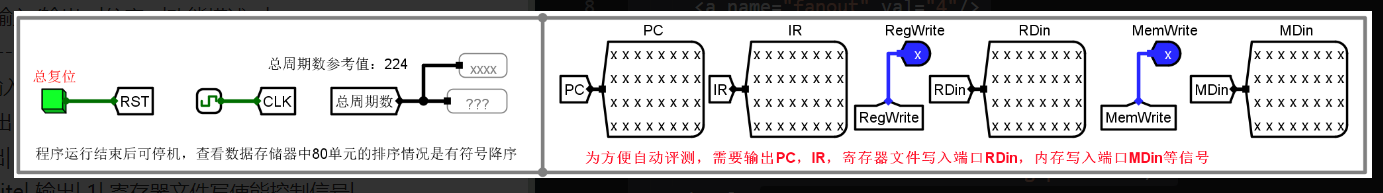


要求支持8条 MIPS 核心指令，最终设计实现的 MIPS 处理器能运行实验包中的冒泡排序测试程序 sort.asm，该程序自动在数据存储器0~15号字单元中写入16个数据，然后利用冒泡排序将数据升序排序，要求统计指令条数与 MARS 中的指令统计数目进行对比。



**电路框架**

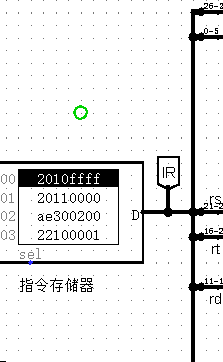
cpu.circ



**电路引脚**  
| 信号 |输入/输出 |位宽 |功能描述 |  
| ------------ | ------------ | ------------ | ------------ |  
| CLK| 输入| 1|时钟信号|  
| PC| 输出| 32|程序寄存器的值|  
| IR| 输出| 32| 当前指令字|  
| RegWrite| 输出| 1| 寄存器文件写使能控制信号|  
| RDin| 输出| 32| 寄存器文件写入端口的数据|  
| MemWrite| 输出| 1| 存储器写使能控制信号|  
| MDin| 输出| 32| 存储器写入端口的数据|

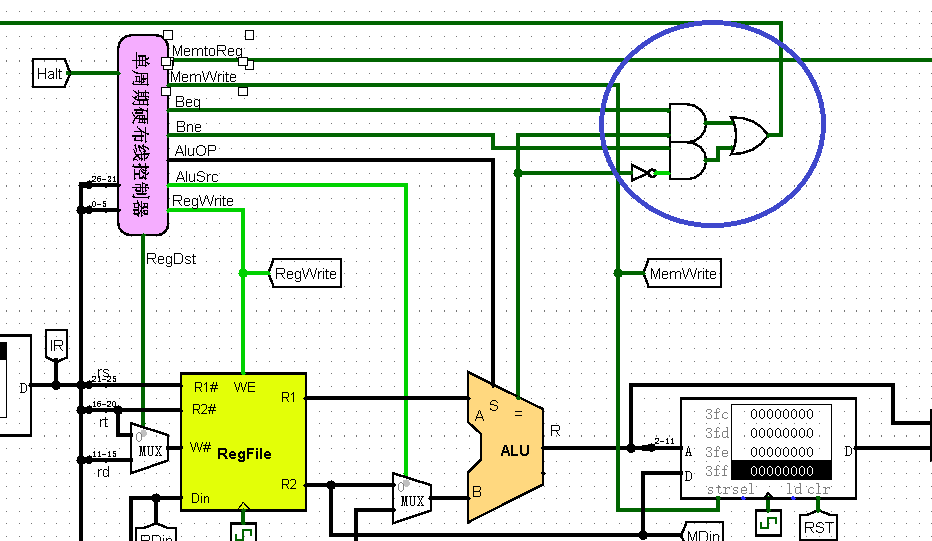
1. **设计电路**
2. **分离不同类型指令**

使用分线器，将不同类型的指令的不同段分离，如下：

****

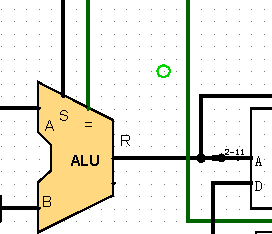
1. **设计PCsrc信号**

结合硬布线控制器，可知Beq和Bne对应数据通路中的Branch信号，Beq和Bne只要有一个有效时PCsrc为真，因此可设计出如下的逻辑电路：



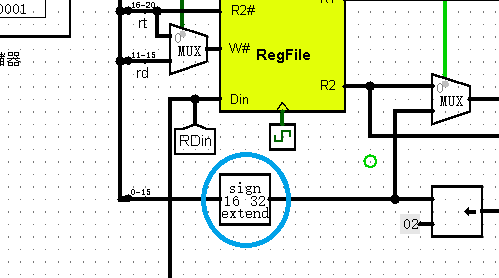
1. **设计输出储存器的地址码：**

32位的输入信号经过分线器的选择，均选取2-11位，即低2位不用，高20不用。MIPS指令的字长位定长的32位即4字节，所以指令计数器形式上的PC=PC+1实际上是PC=PC+4，而，所以低两位不会变化故不取。而储存器的容量为1024，所以只取2-11位。

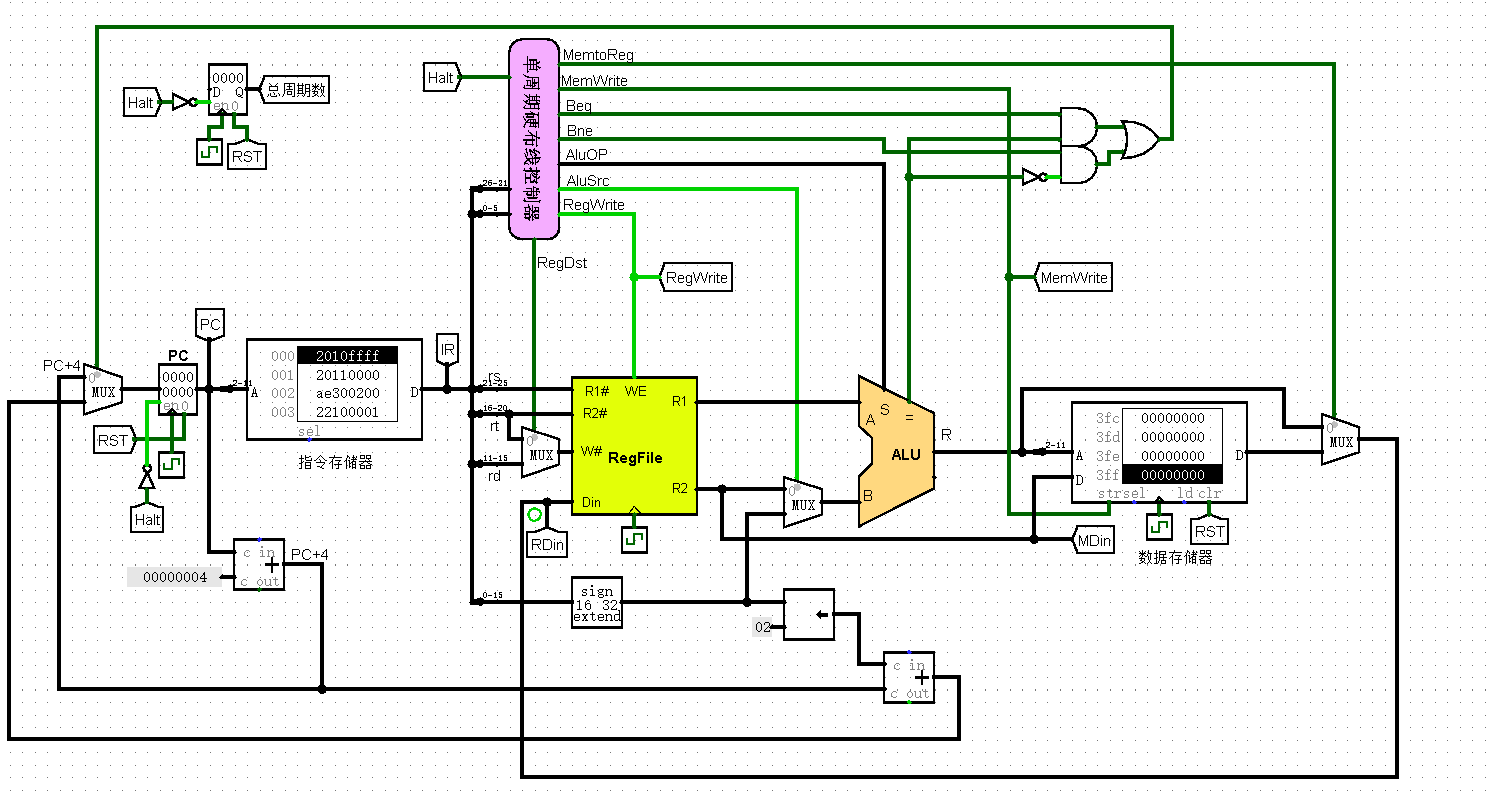
****

**4、设计ALU输入端**

根据数据通路以及所学知识可得，需要将I型指令的imm立即数带符号扩展到32位才能做运算。如下：



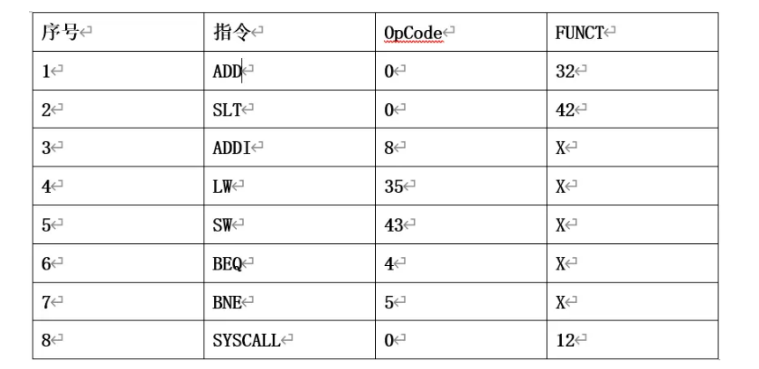
**则总的数据通路：**

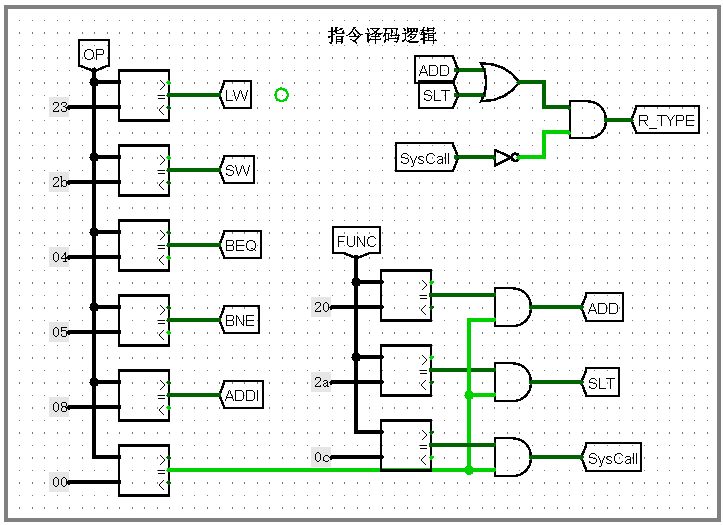
****

**5、设计单周期硬布线控制器**

**5.1 指令译码逻辑**

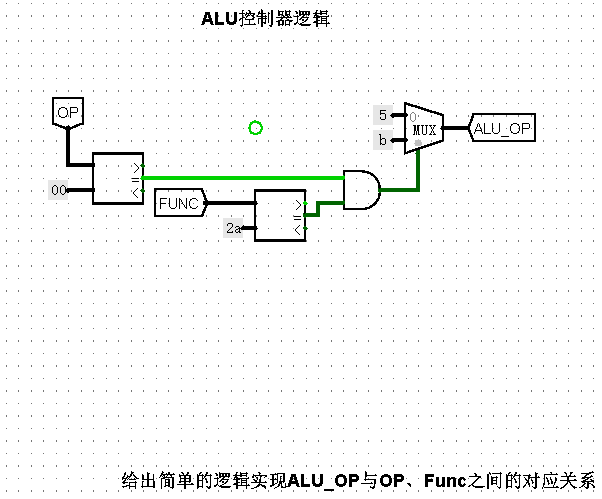
根据以上指令的OpCode字段和FUNCT字段结合logism自带的比较器，可以设计出指令译码逻辑部分，其中SysCall输入特殊的R型指令，所以R\_TYPE指令形成的信号中SysCall要取反再与其他R型指令取或的结果做与。

****

****

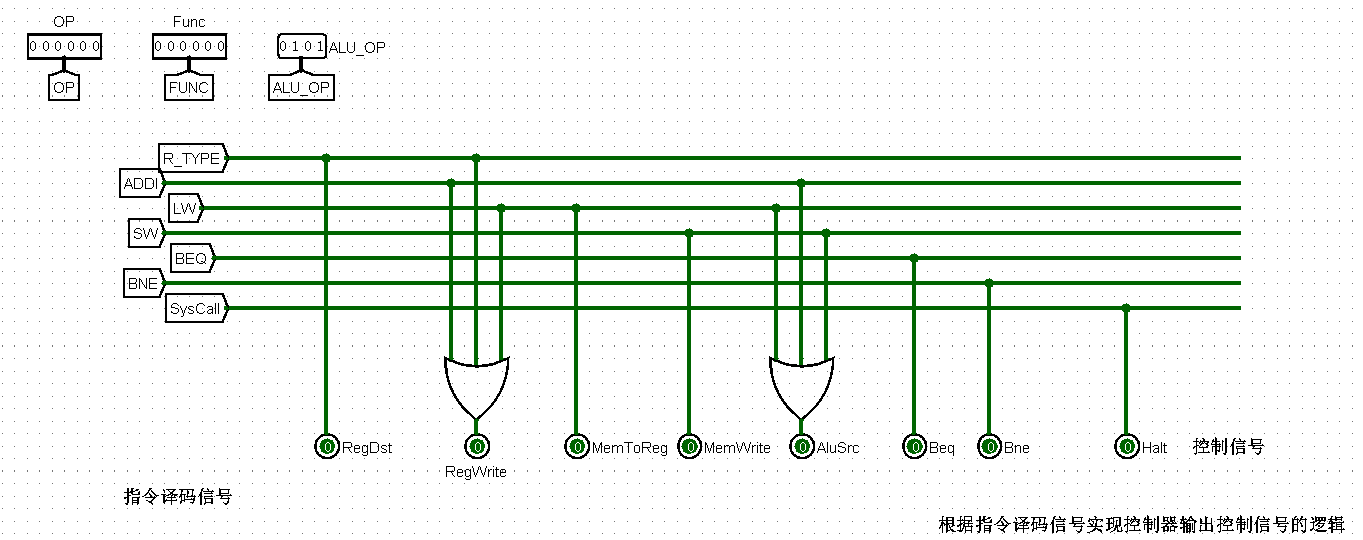
**5.2** **ALU控制逻辑**

由指令控制逻辑，可得只有SLT需要用到ALU的比较运算，其他均为加法运算，所以可以直接复制一个SLT的隧道加上数据选择器，直接设计ALU的控制逻辑，不需要用到给定的OP和FUNC字段的隧道。

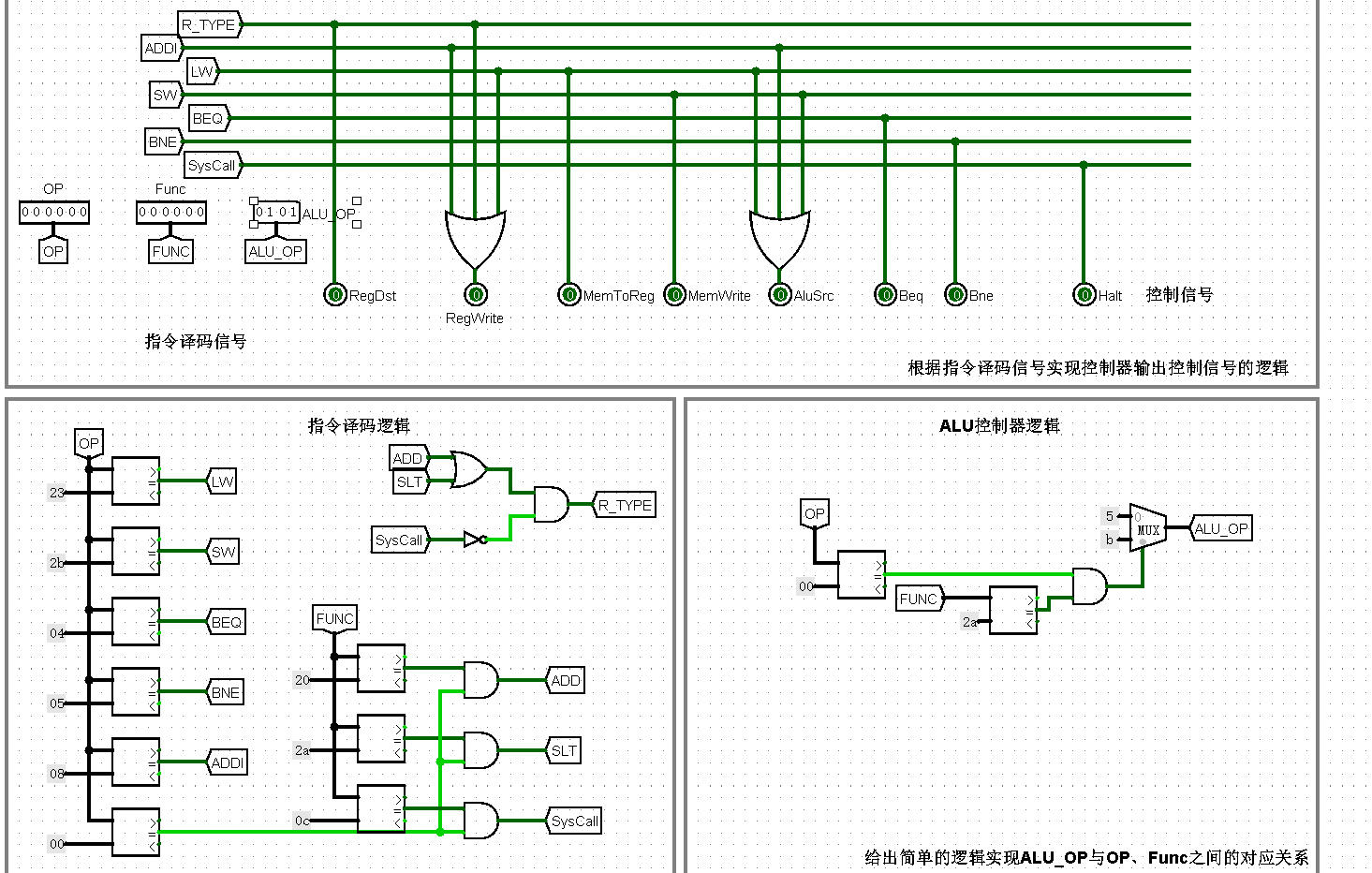


**5.3控制机器输出信号**

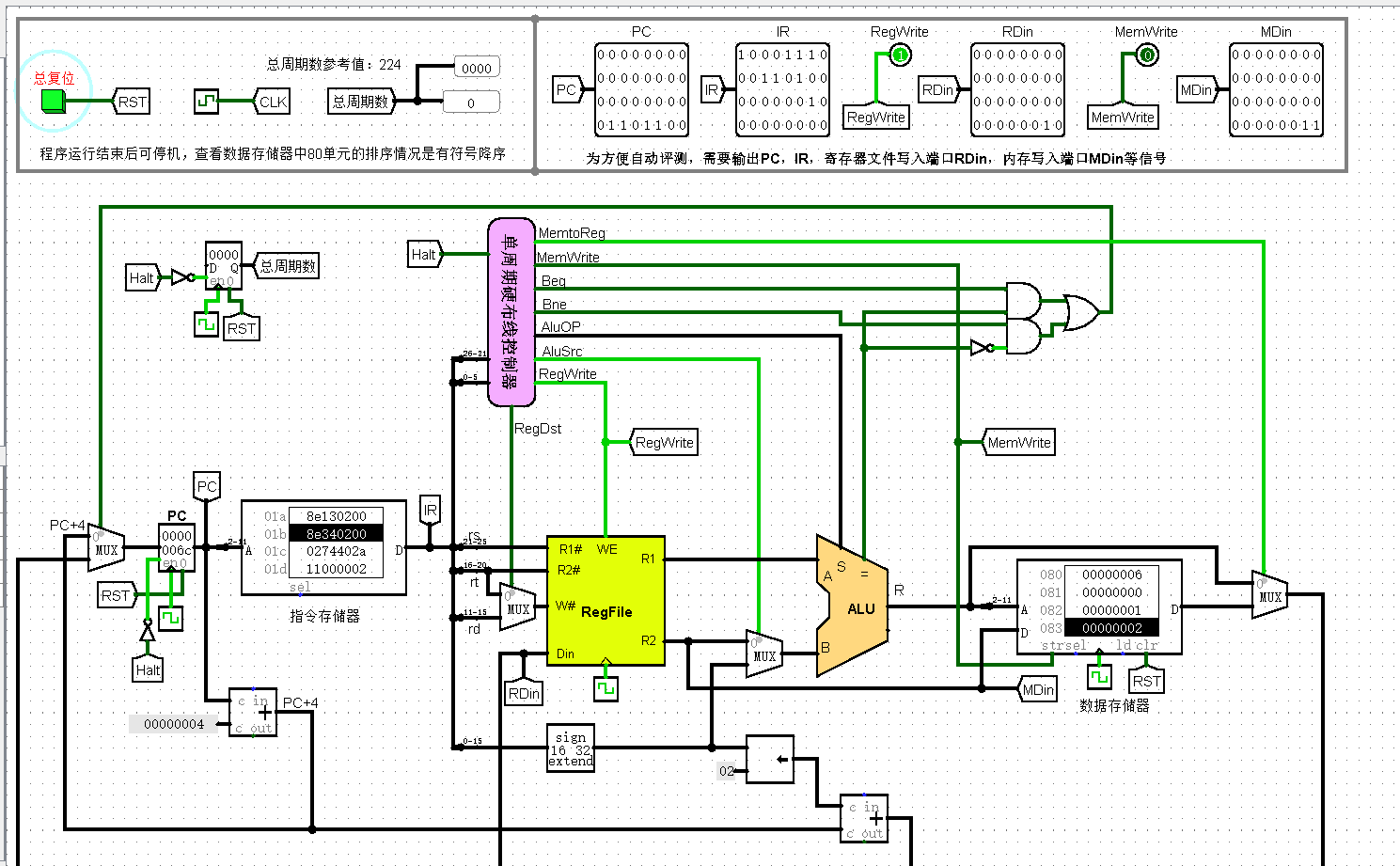
根据所给出控制信号的产生条件，结合数字逻辑知识，可以设计输出控制信号的逻辑

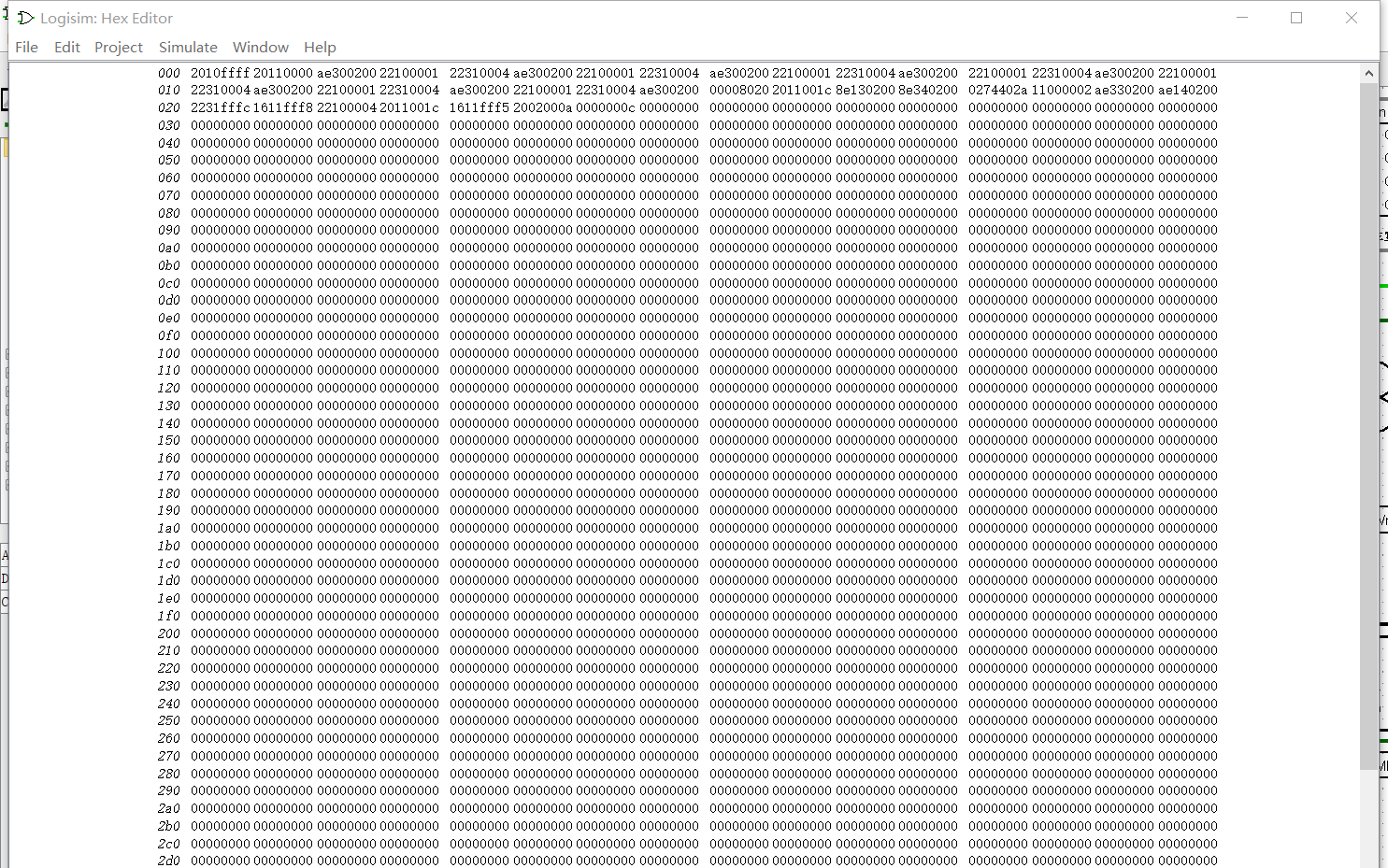
****

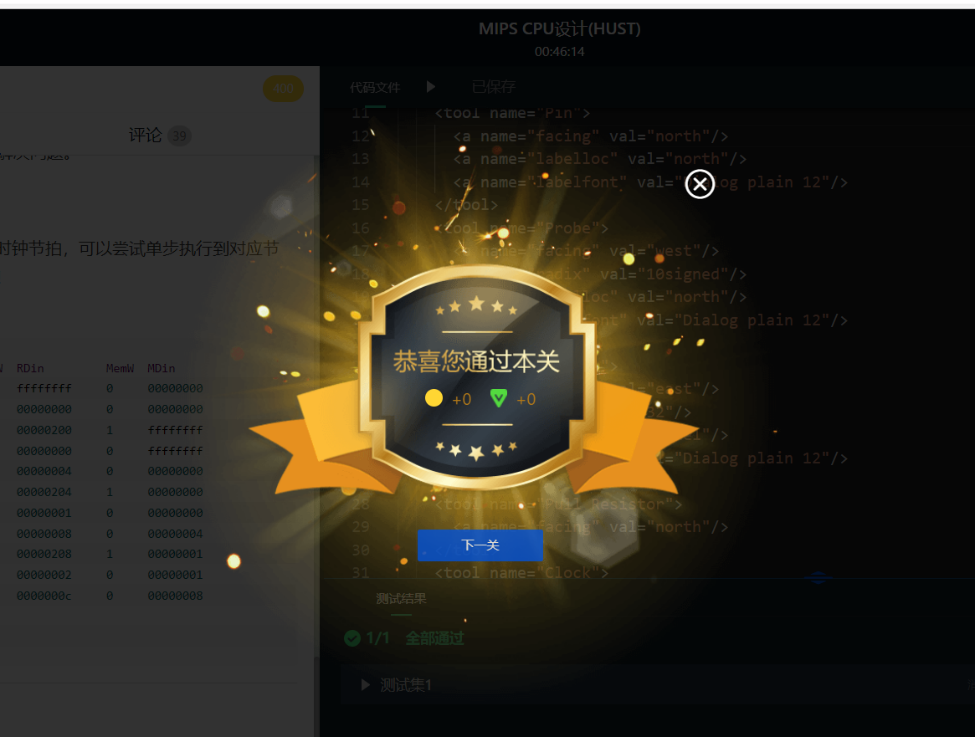
**单周期硬布线控制器总设计如图：**

****

1. **实验结果**

****

****



1. **实验总结/体会**

本次实验设计了单周期MIPS CPU，借助给定的数据通路，可以很快的连接好大致框架，其中PCsrc和输入储存器的地址段的选择是要解决的关键问题。PCsrc要根据数据通路中所给的信息，结合实验所给出的硬布线控制器基本框架，对已知数据通路进行适当的扩展，才能正确输出PCsrc信号。而输入储存器的地址则应结合RAM有关实验以及PC指令计数器在MIPS中是PC=PC+4，才能顺利的选出正确的十位地址码。在设计机器输出信号的时候应注意所给的提示，比如SysCall不在正常的R型指令中，所以设计R型指令信号的时候要对其进行取反处理。ALU控制器的设计时，只需要设计能选择两种运算方式的控制器即可，故不用所给的OP和FUNC隧道，直接选择SLT隧道加上数据选择器设计一个简洁的SLT控制器。