1　实验概述

1.1　实验名称

CPU 设计---单周期和多周期 MIPS CPU(8 条指令)。

1.2　实验所需软件及设备

(1) Logisim2.7.1 软件 1 套
(2) 微型计算机 1 台；

1.3　实验目的

理解 MIPS 单周期和多周期处理器的基本原理，能分别利用硬布线控制器和微程序控制器的设计原理在 Logisim 平台中设计实现 MIPS 单周期和多周期处理器。

1.4　实验基础

硬布线和微程序控制器基本原理、MIPS 指令执行流程、寄存器相关知识、存储器访问相关基础知识等。

1.5　实验内容及要求

完成单周期硬布线控制器和多周期微程序控制器电路的设计。

了解单周期硬布线控制器和多周期微程序控制器电路的基本概念及使用原理，支持表 1 中的 8 条核心指令，实现 MIPS 处理器能运行实验包中 sort 排序程序，利用冒泡排序对数据进行升序排序。

表 1 MIPS 指令及功能描述

#	MIPS 指令	RTL 功能描述
1	add $rd,$rs,$rt	R[$rd]←R[$rs]+R[$rt],溢出时产生异常,且不修改 R[$rd]
2	slt $rd,$rs,$rt	R[$rd]←R[$rs]<R[$rt],小于置 1,有符号比较
3	addi$rt,$rs,imm	R[$rt]←R[$rs]+$SignExt_{16b}$(imm),溢出产生异常
4	lw$rt,imm($rs)	R[$rt]←$Mem_{4B}$(R[$rs]+$SignExt_{16b}$(imm))
5	sw$rt,imm($rs)	Mem_{4B}(R[$rs]+$SignExt_{16b}$(imm))←R[$rt]
6	beq $rs,$rt,imm	if(R[$rs]=R[$rt]) PC ← PC+$SignExt_{18b}$({imm,00})
7	bne $rs,$rt,imm	if(R[$rs]!=R[$rt]) PC ← PC+$SignExt_{18b}$({imm,00})
8	syscall	系统调用,这里用于停机

2　CPU 设计实验

2.1　单周期 MIPS CPU 设计

（1）设计思路

（1.1）总体设计思想

MIPS CPU 特点：单周期 MIPS CPU 指令是定长的，全部的指令都在一个时钟周期内完成，性能取决于执行最慢的指令。单周期 MIPS CPU 不能设 AR DR 和 IR 寄存器。单周期 MIPS CPU 采用程序和数据分开存放的哈佛结构。单周期 MIPS CPU 的运算器和 PC 累加器分离。

设计原理和思路：首先实现单周期硬布线控制器，然后再实现取指令，ADD 指令，LW 指令，SW 指令，BEQ 指令的数据通路。其中控制器详见控制器设计过程，而因为没有 IR 寄存器，所以取指令的实现依靠 PC+4 直接放入 PC，ADD 指令依靠 MIPS 寄存器文件和 ALU 进行写入寄存器，写出寄存器和运算，将 Rs 和 Rt 的值加起来放入 Rd，LW 指令注意低 16 位为立即数，需要扩展为 32 位，然后与 Rs 相加形成主存要寻找的地址，读出来后写入 Rt。SW 指令是注意 Rs 和立即数扩展后相加得到的地址要作为主存写入数据的地址，而要写入的数据直接由 Rt 提供。BEQ 指令则要判断 Rs 和 Rt 的值是否相等，如果相等则把立即数扩展为 32 位，并左移两位，然后加上 PC 的值，再写回到 PC。这里注意 Rs 是 21-25 位，Rt 是 16-20 位，Rd 是第 11-15 位，或者立即数是第 0-15 位。

我们的控制器要产生 MemToReg，MemWrite，Beq，Bne，ALU_OP，AluSrc，RegWrite 信号，那么在单周期硬布线 CPU 设计过程中，取指令不需要什么控制信号，只需将 PC 的值分两路，一路连接指令存储器读取指令，一路连接加法器，让其加 4 再写回 PC。我们的 ADD 指令则需要 Rs，Rt，Rd 连接到寄存器文件对应部分，寄存器文件 WE 接口肯定要连接 RegWrite 信号，这里因为要写入 ALU 的数据有立即数扩展和 R2，以及要写入寄存器文件的 Din 有数据存储器的数据和 ALU 计算结果，所以这两处要加一个多路选择器，选择信号分别是 AluSrc 和 MemToReg，还有 ALU 的加法需要连接 ALU_OP 信号，以及数据存

储器读写需要连接 MemWrite 信号，而由于我们 BEQ 指令和 BNE 指令存在，我们需要一个与门将 Beq 信号和 ALU 相等信号连接，需要一个与门将 Bne 信号和 ALU 相等信号取反连接，再连一个或门，连接到 PC 前一个多路选择器上，根据情况选择是 PC+4 还是 PC+(32[IMM]<<2)写入 PC，这就是具体控制信号的连接，具体数据通路详见下方。

（1.2）各种指令数据通路

取指令数据通路

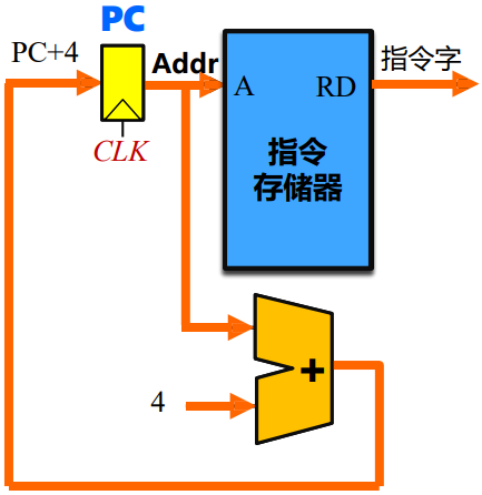

图 1.1.2-1 取指令

ADD 指令数据通路

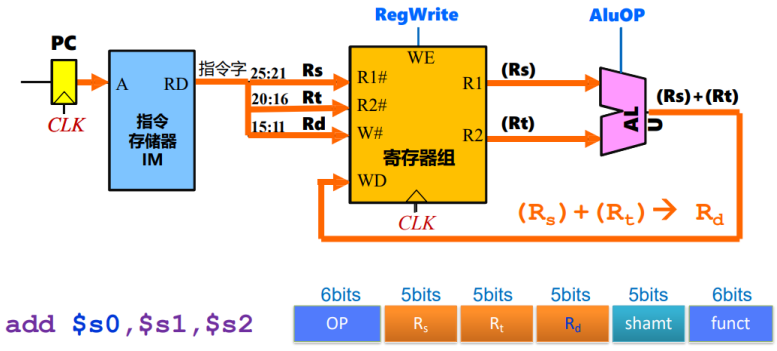

图 1.1.2-2 ADD 指令

LW 指令数据通路

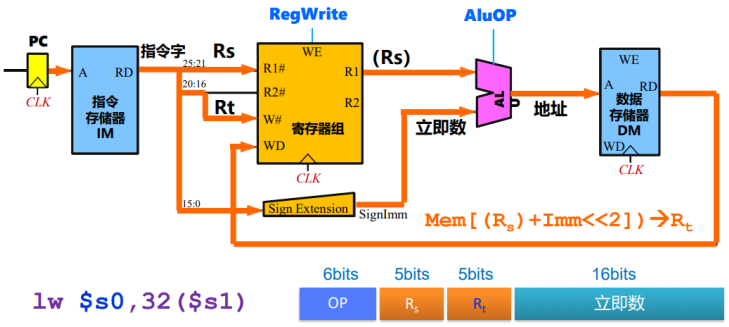

图 1.1.2-3 LW 指令

SW 指令数据通路

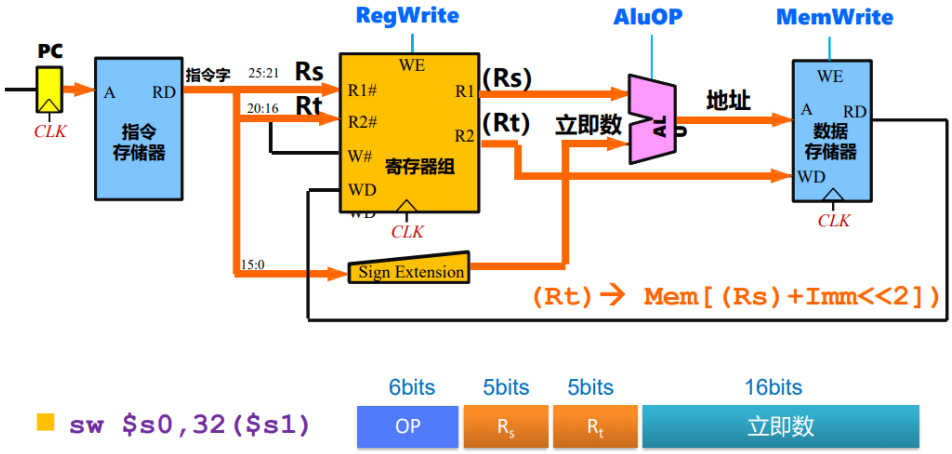

图 1.1.2-4 SW 指令

BEQ 指令数据通路

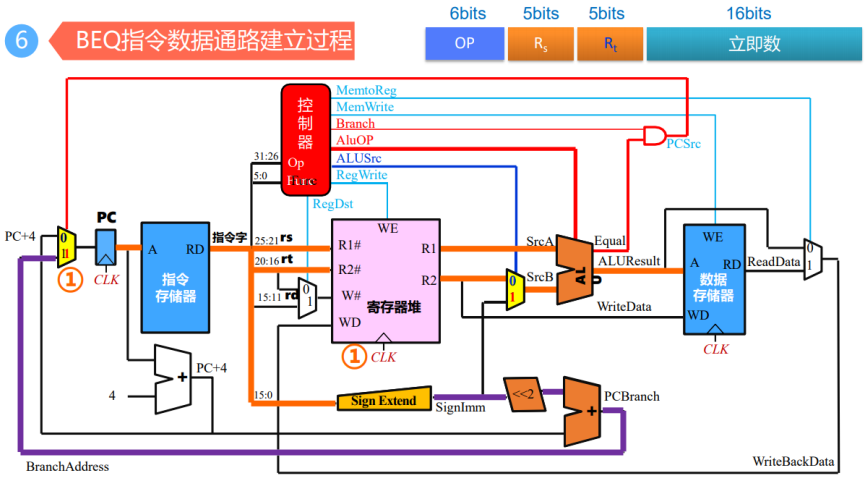

图 1.1.2-5 BEQ 指令

综合数据通路

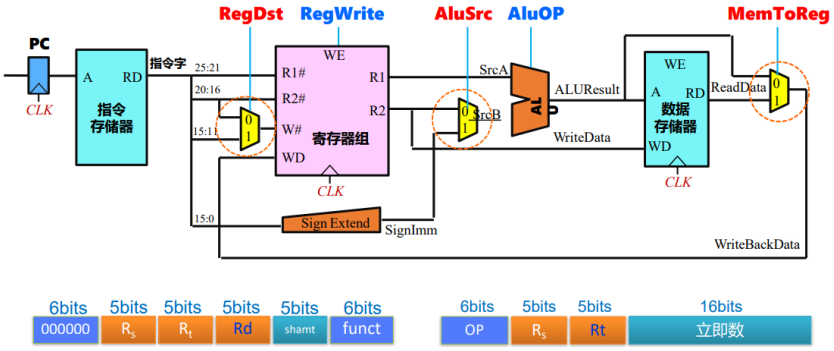

图 1.1.2-6 综合数据通路

（1.3）单周期 MIPS 控制器设计过程

首先看指令译码逻辑，OP 是第 26-31 位，FUNC 是 0-5 位，根据 OP 和 FUNC 字段的值，查找 MIPS 指令集，能确定出 LW，SW，BEQ 等信号，注意 R_TYPE 是 R 型运算指令，SYSCALL 是特殊 R 型指令，不属于这个类别。所以用 ADD 和 SLT 或后与上 SYSCALL 的非才是 R_TYPE。

再考虑 ALU 控制逻辑部分，只有有 SLT 信号时需要比较是否相等，其余都是加法，所以通过 SLT 进行选择，如果 SLT=0，则选择 5，也就是进行加法，如果 SLT=1，则选择 b，进行比较。

最后根据指令译码信号实现控制器输出控制信号的逻辑。产生条件由下图给出：

#	控制信号	信号说明	产生条件
1	MemToReg	写入寄存器的数据来自存储器	lw指令
2	MemWrite	写内存控制信号	sw指令　未单独设置MemRead信号
3	Beq	Beq指令译码信号	Beq指令
4	Bne	Bne指令译码信号	Bne指令
5	AluOP	运算器操作控制符	加法，比较两种运算
6	AluSrcB	运算器第二输入选择	Lw指令，sw指令，addi
7	RegWrite	寄存器写使能控制信号	寄存器写回信号
8	RegDst	写入寄存器选择控制信号	R型指令
9	Halt	停机信号，取反后控制PC使能端	syscall指令

图 1.1.3-1 控制信号产生条件

（2）电路图

（2.1）硬布线控制器电路图

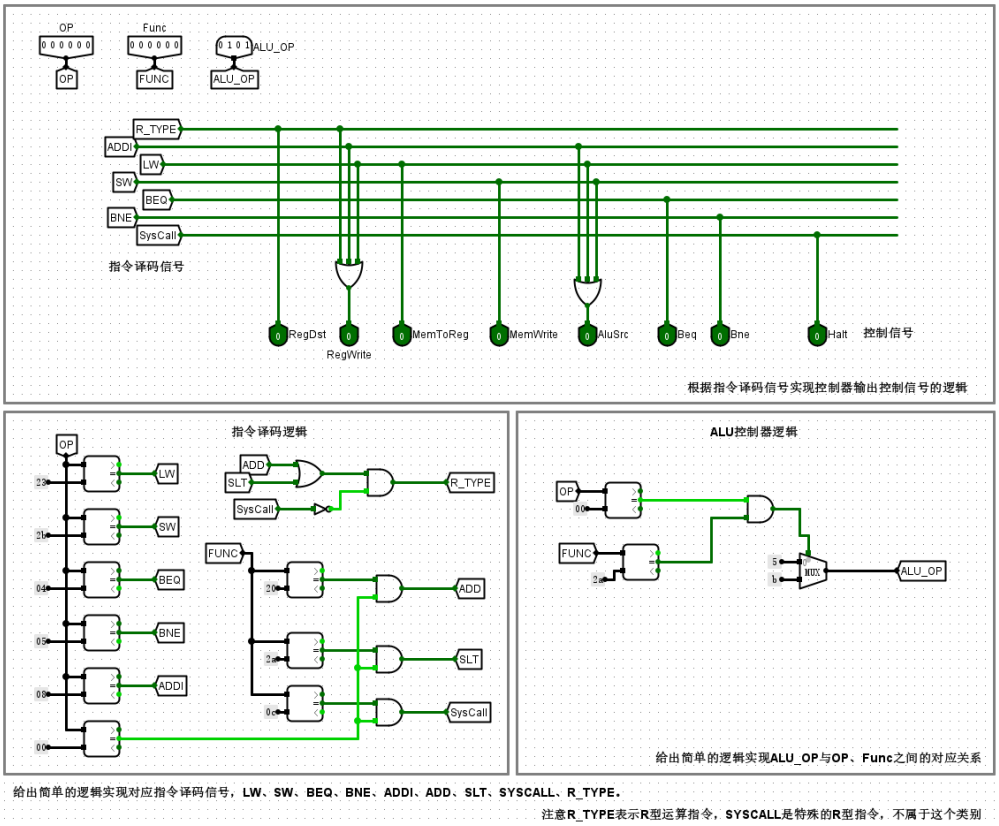

图 1.2.1-1 硬布线控制器电路图

（2.2）CPU 设计图

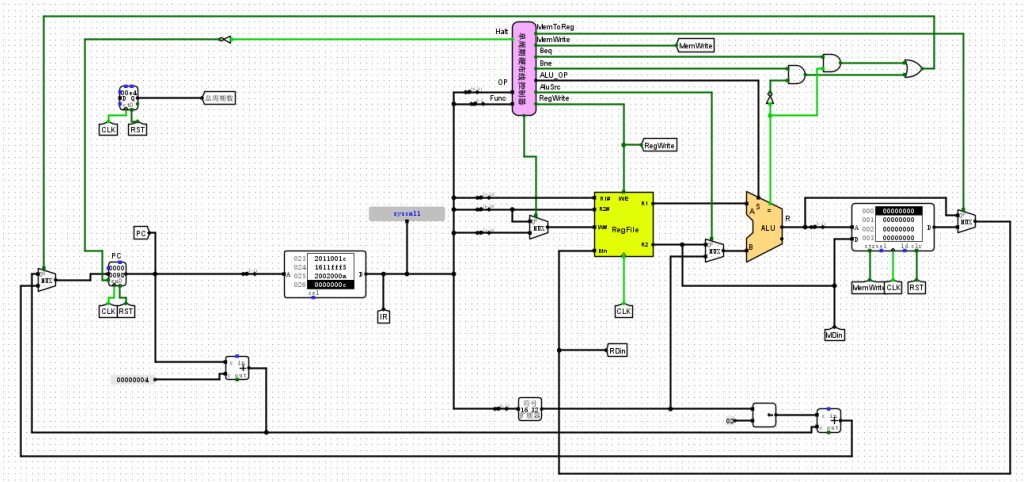

图 1.2.2-1 CPU 设计图

（3）测试图

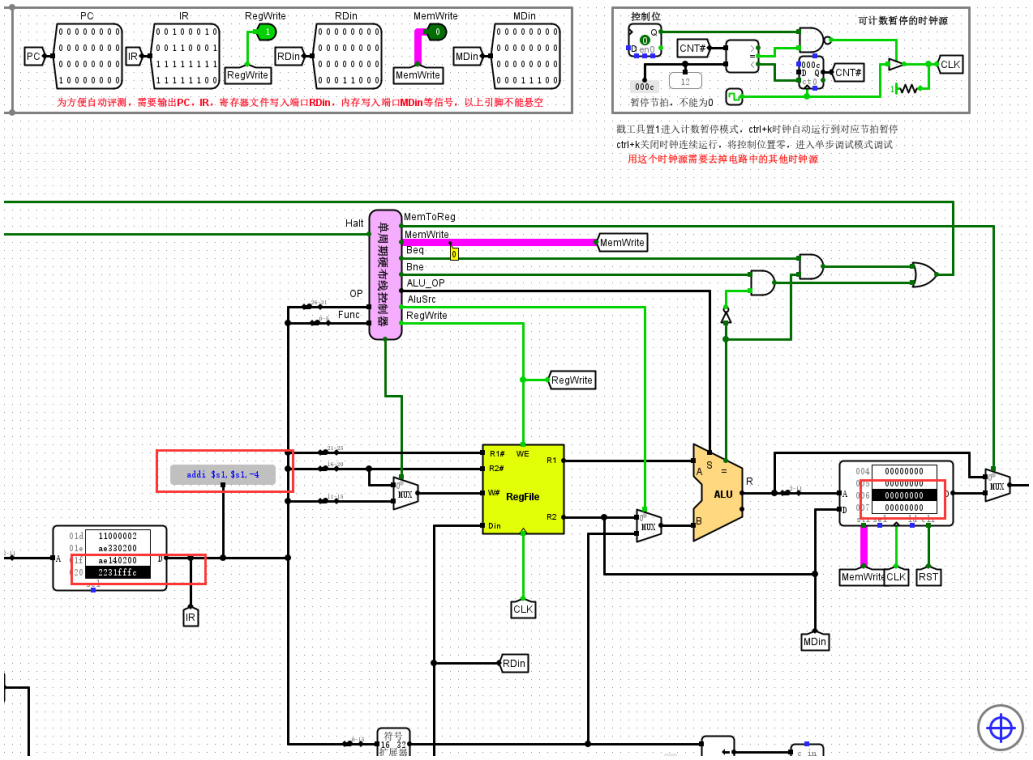

图 1.3-1 单周期 MIPS CPU 测试图 1

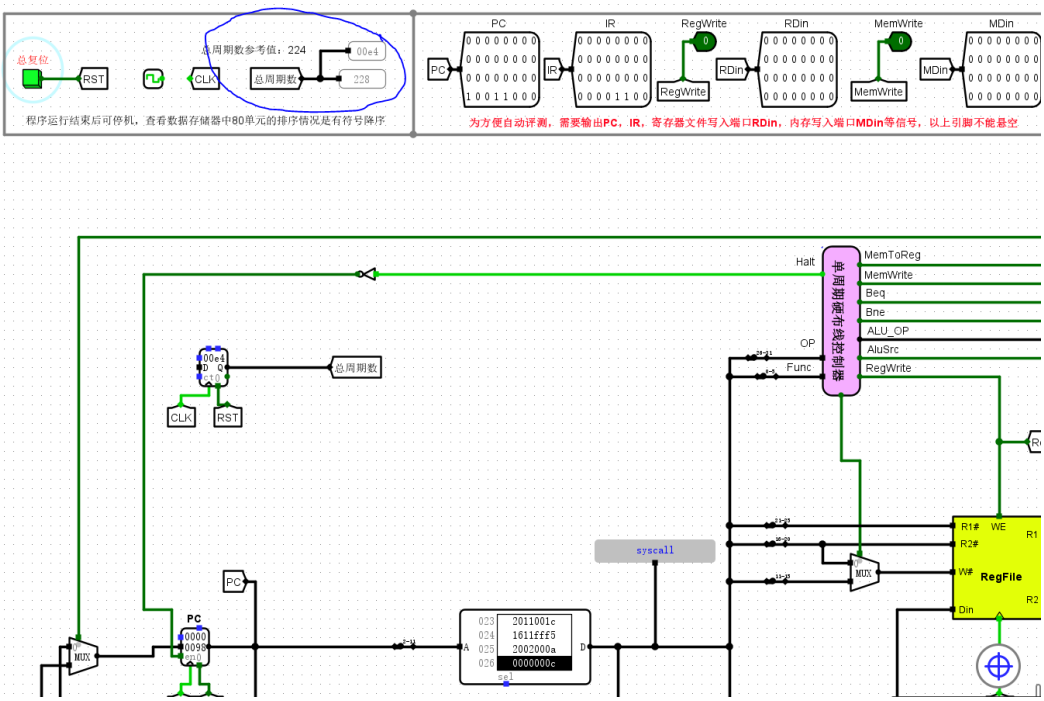

图 1.3-2 单周期 MIPS CPU 测试图 2

本地测试如上图，正常，且已通过头歌平台测试。

（4）故障与测试分析

(4.1)

故障现象 1：输出只有 0

原因分析：没有导入相应的指令数据

解决方案：导入 sort.hex 进入指令存储器

(4.2)

故障现象 2：参与平台测试时，输出结果只有若干行，就不往下继续测试了。

原因分析：在进行扩展时，没有选择符号扩展，选择的是 0 扩展，出现了此问题。

解决方案：将扩展器由 0 扩展修改为符号扩展

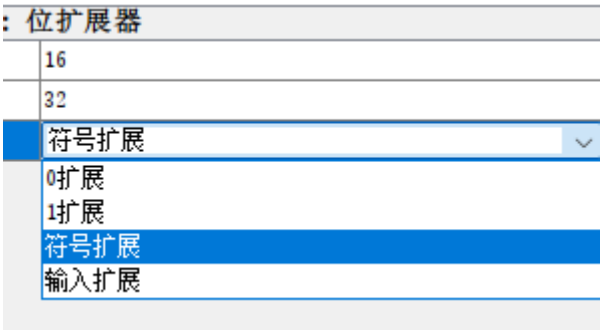

图 1.4.2-1 修改扩展器

2.2 多周期 MIPS 微程序 CPU 设计

（1） 设计思路

（1.1）总体设计思想

多周期 MIPS CPU 特点：将指令存储器与数据存储器合并，不再进行区分，分时使用部分功能部件，增加了 IR，DR 等寄存器。主要功能单元输出端增加寄存器锁存数据，传输通路延迟变小，时钟周期变短。

设计原理：首先实现多周期微程序控制器，控制器具体实现详见下一部分。实现完控制器后，再实现取指令，ADD 指令，LW 指令，BEQ 指令的数据通路等。取指令要注意 PC 的累加和不同于单周期的部分，分为取指令，

译码及取操作数两个时钟周期。ADD 指令是将两个寄存器中的数据相加再写入寄存器中，分为运算和写入两个时钟周期。SW 指令分为计算地址和访存两个时钟周期。LW 指令是将对应地址的数据从数据存储器中读出来，分为算地址，访内存，写回三个时钟周期。BEQ 指令只有送目标地址一个时钟周期，用于判断取指令是否用分支地址。总体思路类似于单周期 MIPS CPU，都是设计控制器给出操作信号，然后连接正确的数据通路。下面给出每个指令的具体时钟周期。

参考实验教材 P113，我们得到具体指令的时钟周期。取指令分为两个时钟周期，分别是 T1:IR<-(M[PC]),PC<-(PC)+4，T2:A<-(R[IR[25:21]]，B<-(R[IR[20:16]])，C<-(PC)+(sign-extend(IR[15:0]<<2)。ADD 指令有 T3：将 AB 两个寄存器相加写到 C 中，T4：将 C 写到 Rd 中。LW 指令有 T3：A 中数据加上立即数的 32 位扩展送到 C 中，T4：主存中 C 地址的数据送到 DR，T5：DR 中数据送到目的寄存器 Rd。SW 指令有 T3：将寄存器 A 中的数据与立即数 32 位扩展相加送入寄存器 C，将寄存器 B 中数据写到主存以 C 为地址的区域。BEQ 有 T3：if(A == B) PC <- (C)。

根据上述指令的具体时钟周期，再参考部分资料，可以实现具体数据通路，并连接电路图。连接电路图时，仍然要注意控制器各个控制信号输出连接，比如取指令阶段的 T1，我们需要 IrWrite 信号连接 IR，需要 ALU_OP 连接 ALU 的选择端，ALUSrcB 连接 ALU 的 SrcB 部分的多路选择器的选择部分，PCen 信号也要给到 PC 寄存器。其他指令也要注意各个控制信号的连接，最后根据各个指令的时钟周期的数据通路和控制信号进行连接。数据通路如下。

（1.2）各种指令数据通路

取指令数据通路

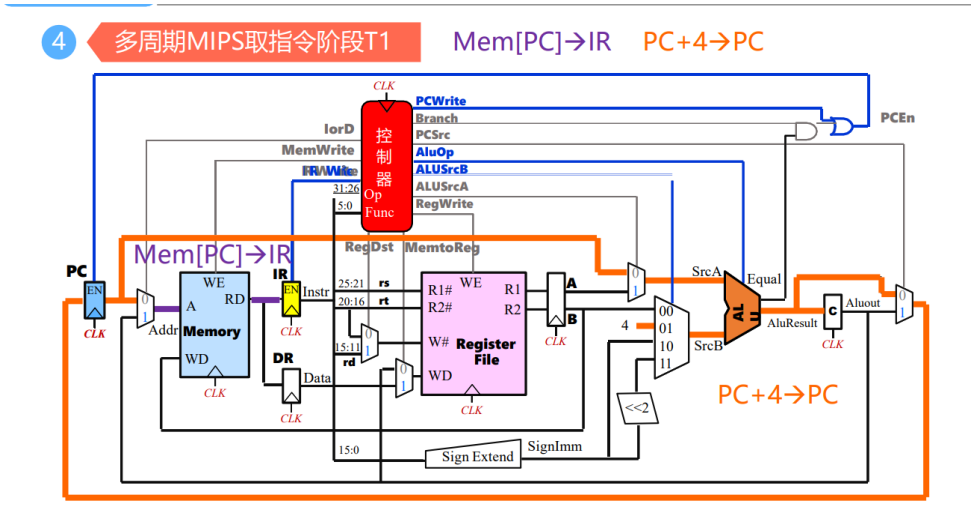

图 2.1.2-1 取指令 1

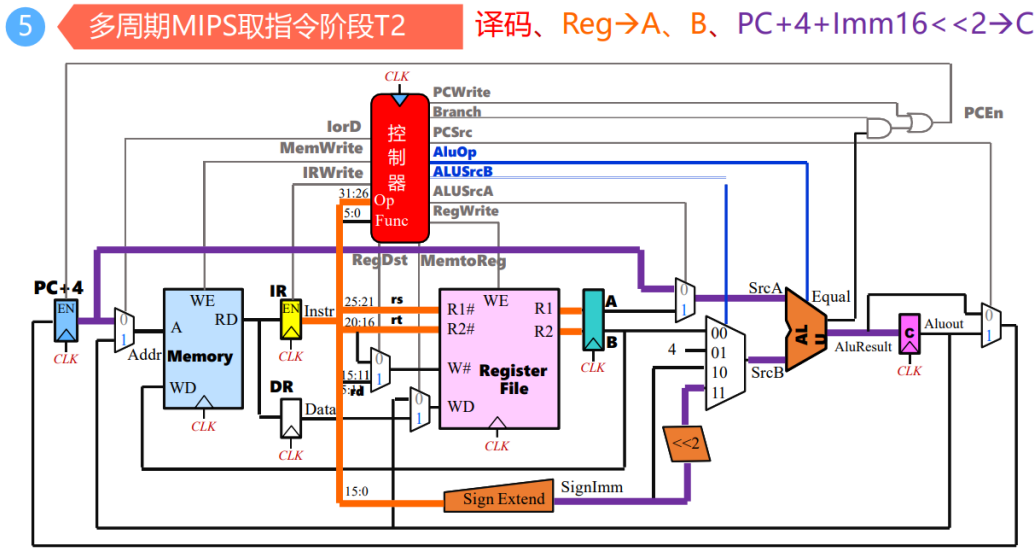

图 2.1.2-2 取指令 2

R 型指令数据通路

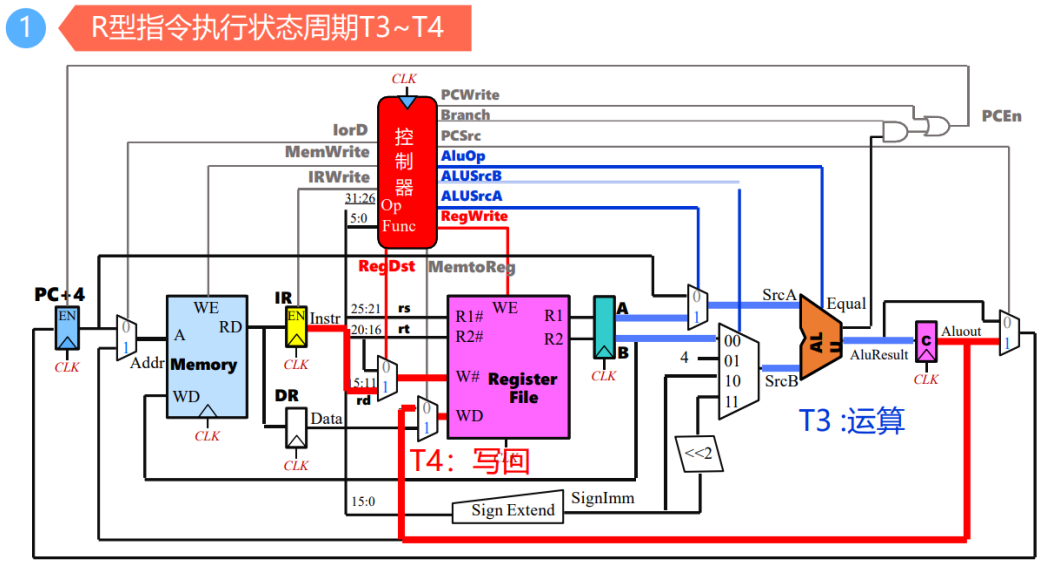

图 2.1.2-3 R 型指令

LW 指令数据通路

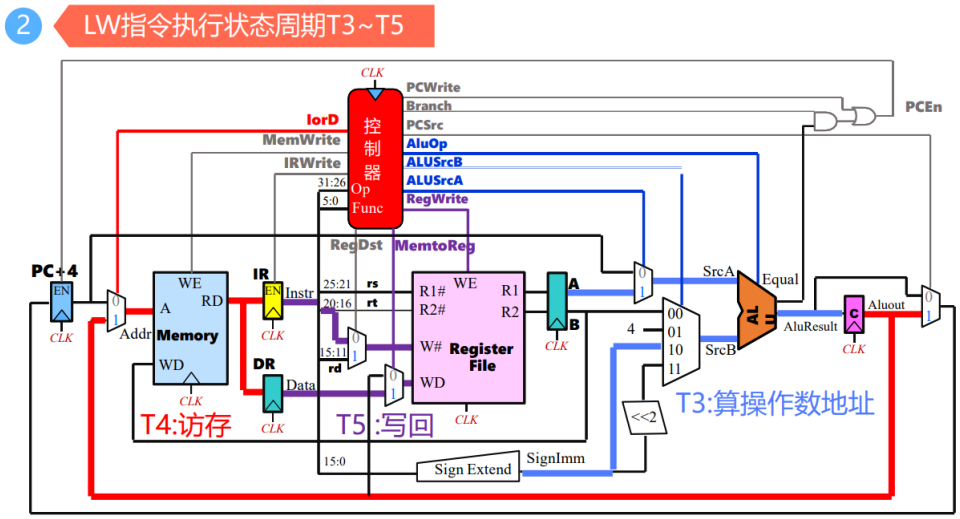

图 2.1.2-4　LW 指令

BEQ 指令数据通路

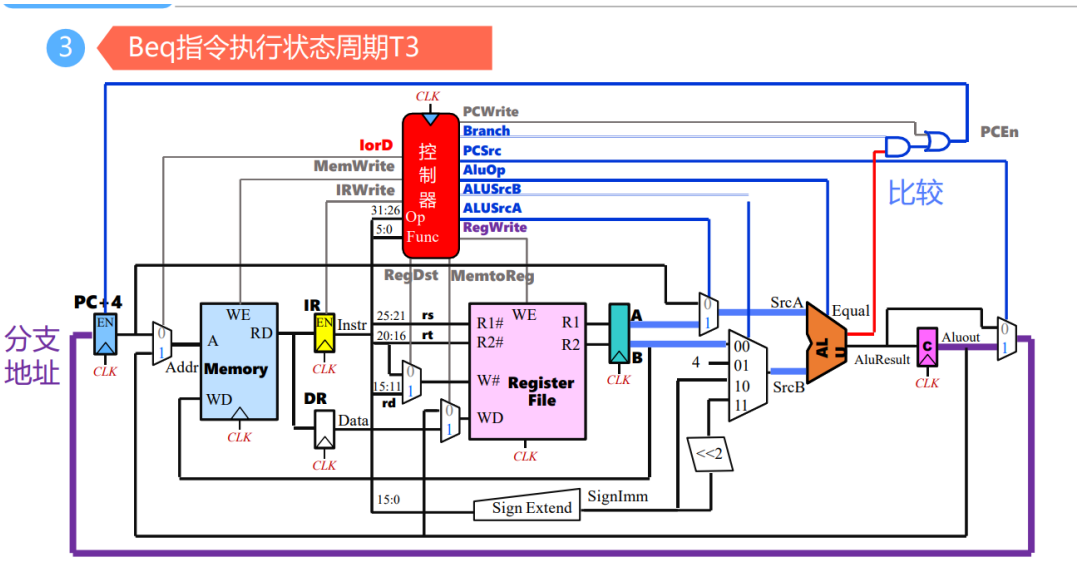

图 2.1.2-5　BEQ 指令

（1.3）多周期 MIPS 微程序控制器设计过程

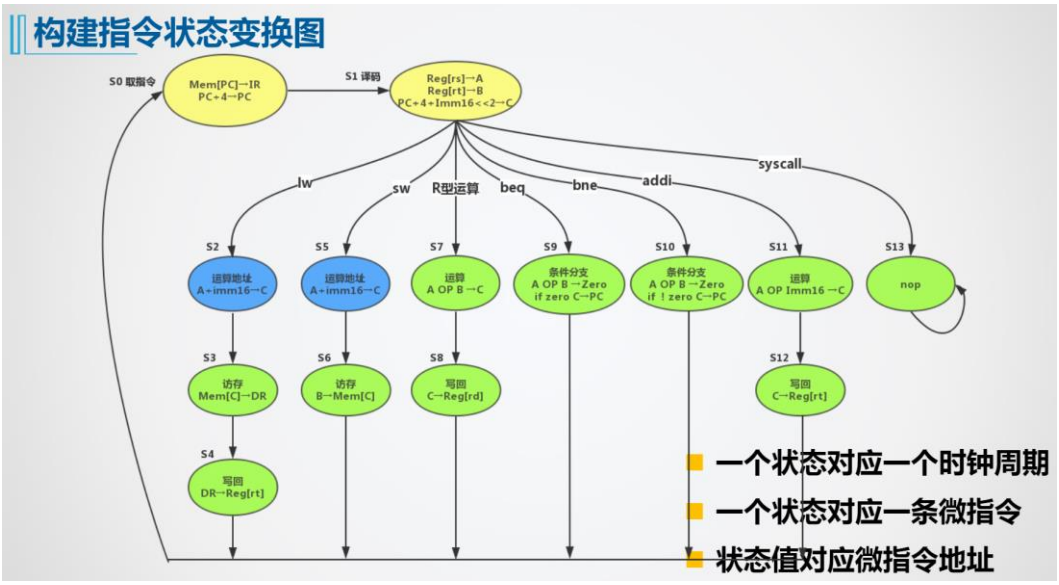

图 2.1.3-1 状态转换图

我们先看状态转换图，一个状态对应一个时钟周期，一个状态对应一条微指令，状态值对应微指令地址。第一个状态 S0，取指令，以 PC 为地址，访存，

取得数据放入 IR，然后 PC+4，它只有一个次态，S1，译码，执行对应的三个时钟周期，那么接下来，它有 S2，S5，S7，S9，S10，S11，S13 等状态，分别根据 lw，sw，R 型运算，beq，bne，addi 和 syscall 几个信号进入对应的状态，完成对应的几个时钟周期，在每个时钟周期内给出对应信号，完成响应操作，进入对应的次态，直到完成该条指令，再次回到 S0 状态，进行取指，执行下一条指令。如此循环。

再看指令译码逻辑，与单周期类似，都是根据 OP，FUNC 字段，根据 MIPS 指令集，然后用比较器，能够得到 LW，SW，BEQ 等控制信号。

再看 ALU 控制器逻辑，根据 ALU_Control 的值决定运算器运算选择控制信号 ALU_OP 的值，ALU_Control==00 时做加法，01 时做减法，10 时由 Func 决定，如果有 ADD 信号，就选 5，如果有 SLT，就选 b，二者进行异或，再进一个多路选择器，根据 ALU_Control 选择加，减，比较等功能。

最后微程序控制器连接参考原理框图：

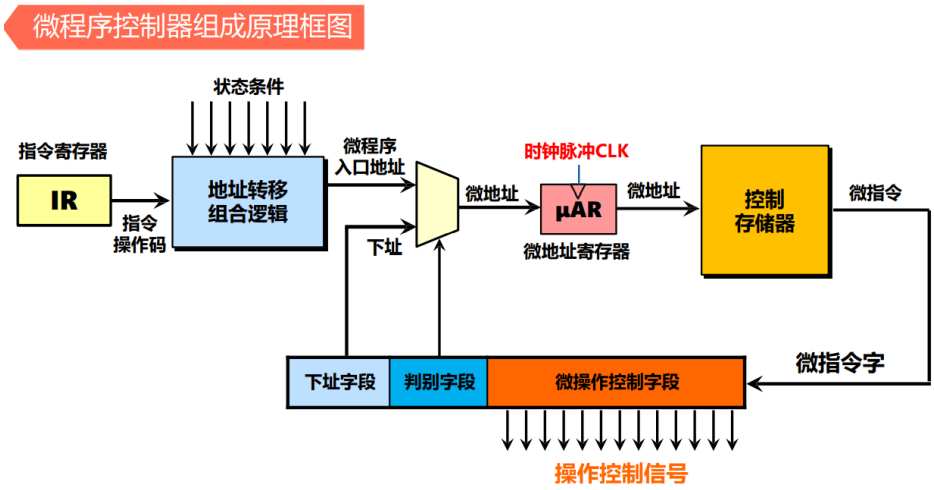

图 2.1.3-2 微程序控制器组成原理框图

这里地址转移组合逻辑，我们用 Excel 表自动生成。

机器指令译码信号							微程序入口地址				
R_Type	ADDI	LW	SW	BEQ	BNE	SYSCALL	入口地址10进制	S3	S2	S1	S0
1							7	0	1	1	1
	1						11	1	0	1	1
		1					2	0	0	1	0
			1				5	0	1	0	1
				1			9	1	0	0	1
					1		10	1	0	1	0
						1	13	1	1	0	1

图 2.1.3-3 微程序入口地址

这里我们的入口地址，7，11 等是根据图 2.1.3-1 状态转换图来填的。最后生成如图逻辑函数。

R_Type	ADDI	LW	SW	BEQ	BNE	SYSCALL	最小项表达式	S3	S2	S1	S0
R_Type&							R_Type		R_Type+	R_Type+	R_Type+
	ADDI&						ADDI	ADDI+		ADDI+	ADDI+
		LW&					LW			LW+	
			SW&				SW		SW+		SW+
				BEQ&			BEQ	BEQ+			BEQ+
					BNE&		BNE	BNE+		BNE+	
						SYSCALL&	SYSCALL	SYSCALL+	SYSCALL+		SYSCALL+

图 2.1.3-4 地址逻辑自动生成

综上，我们就能得到一个微程序控制器，

（2） 电路图

（2.1） 微程序控制器电路图

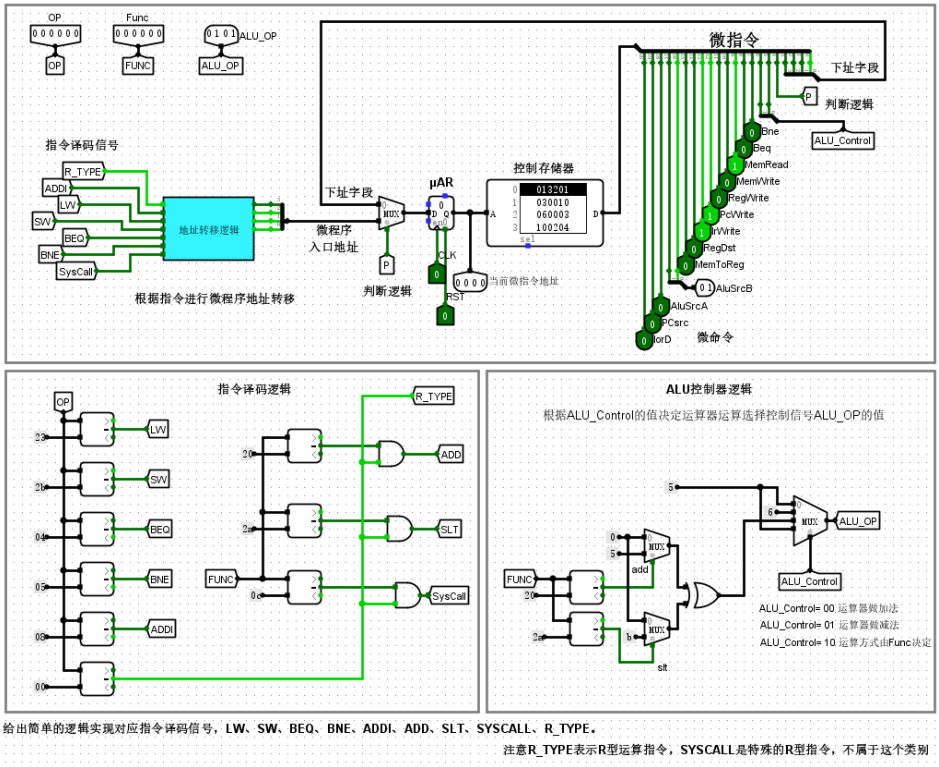

图 2.2.1-1 微程序控制器

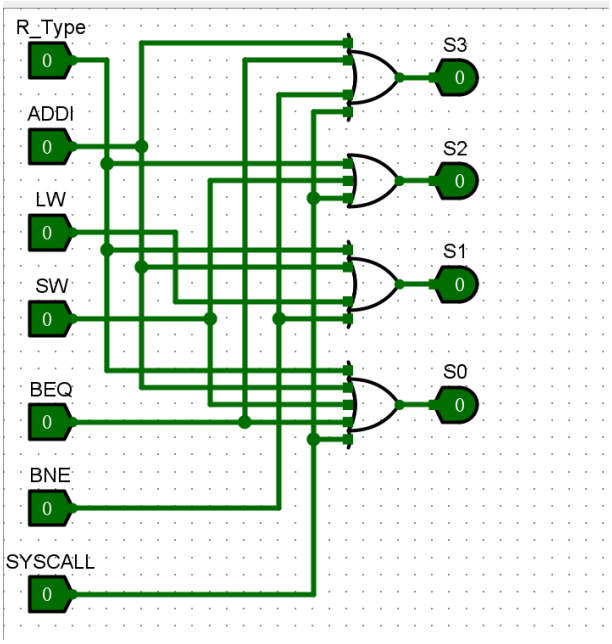

图 2.2.1-2 微程序地址转移逻辑

（2.2） CPU 设计图

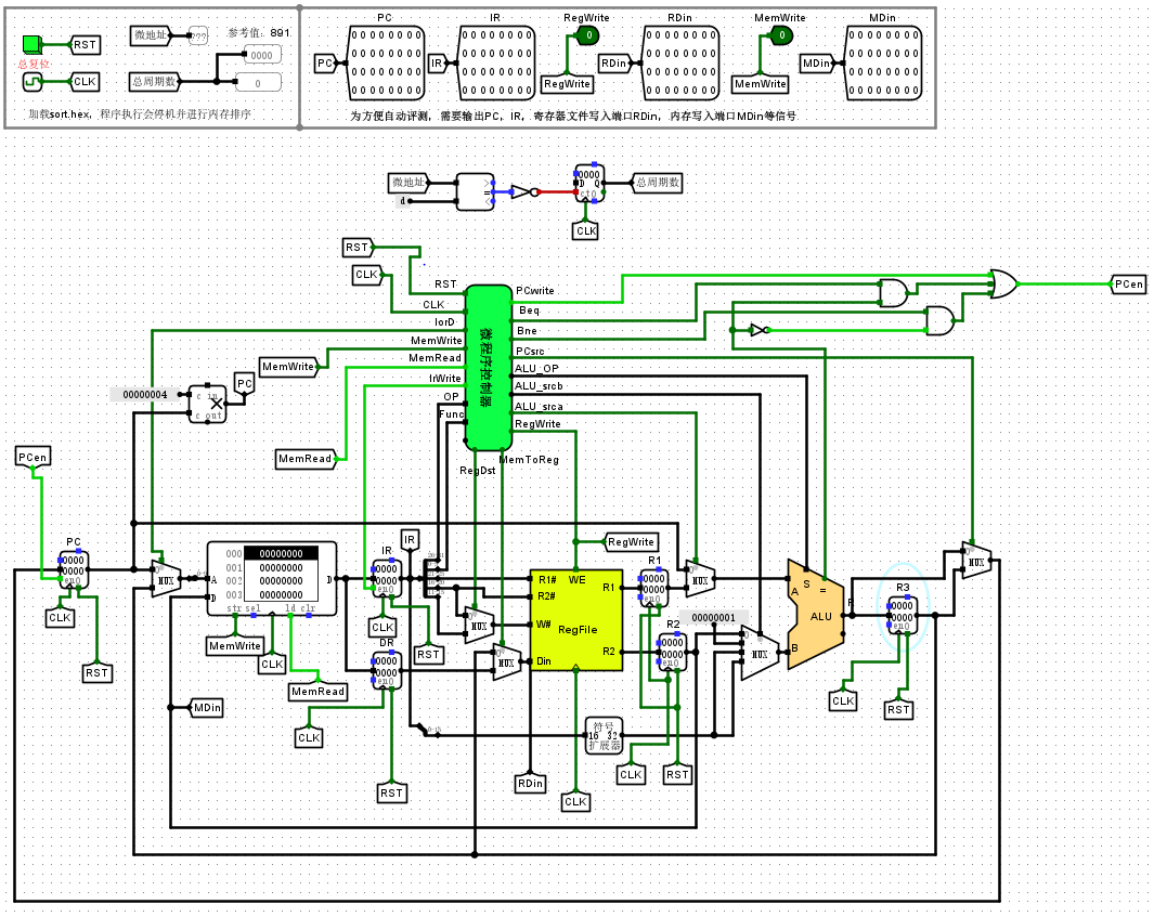

图 2.2.1-3 MIPS CPU 多周期

（3） 测试图

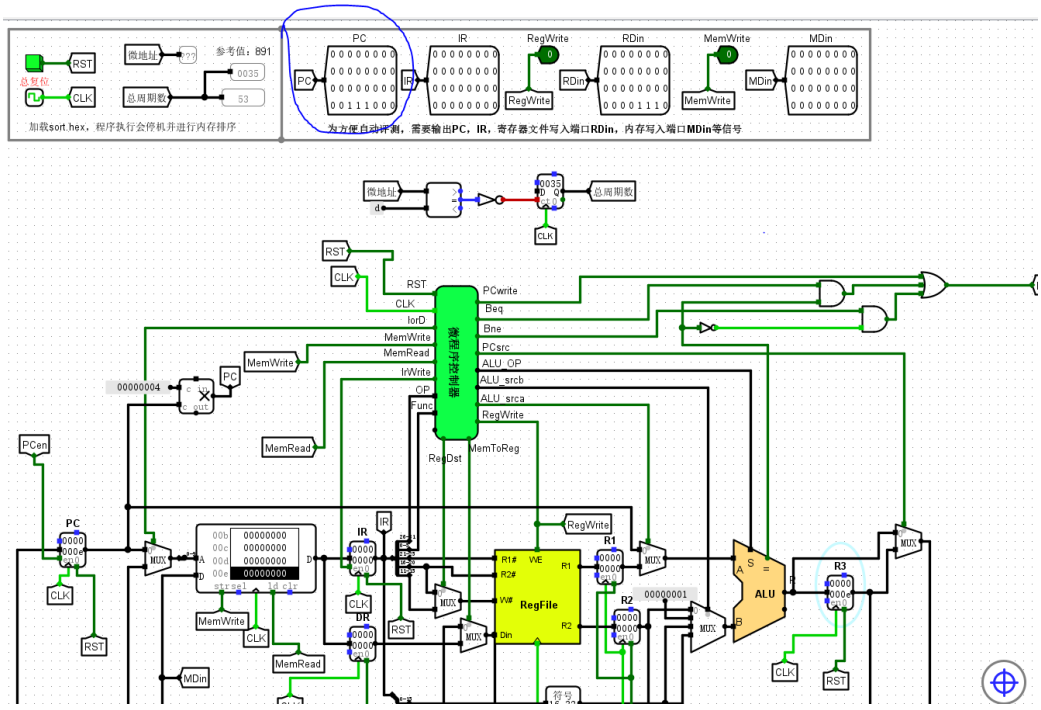

图 2.3.1-1 测试图 1

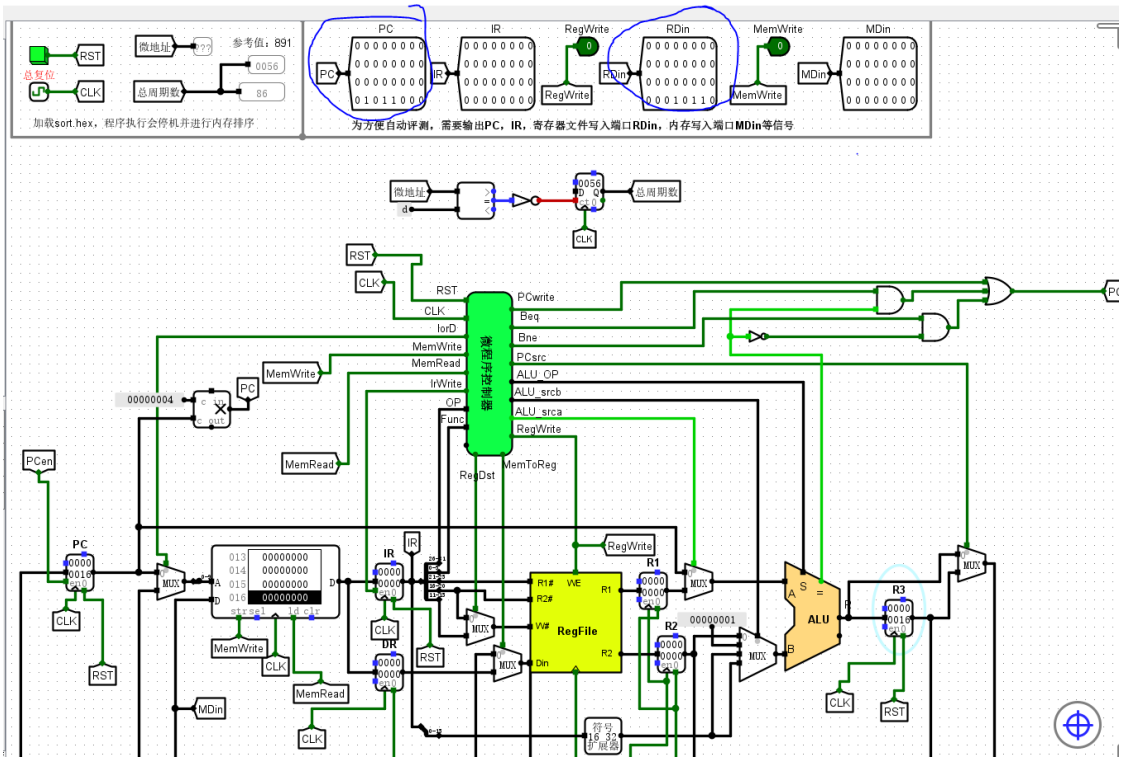

图 2.3.1-2 测试图 2

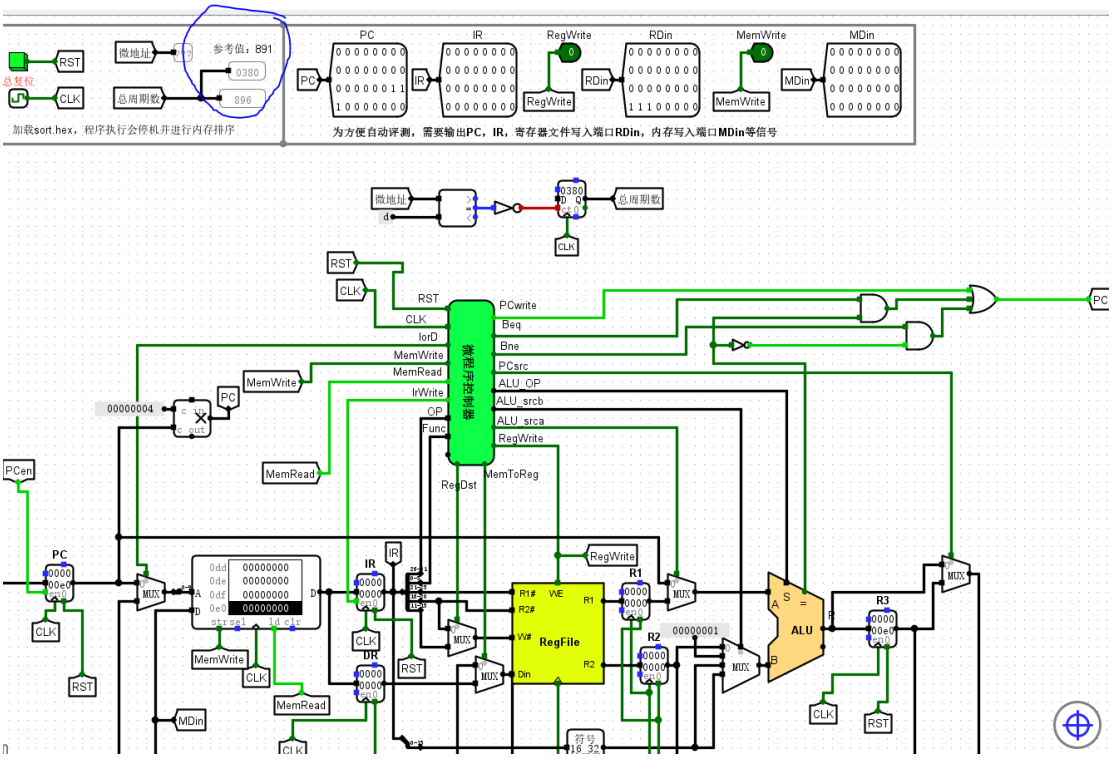

图 2.3.1-3 测试图 3

本地测试如上图，一切正常，也通过头歌平台测试，证明连接成功。

（4）故障与测试分析

(4.1)

故障现象 1：CPU 运行出错，输出全是 0

原因分析：没有正确地生成 PCen 信号。逻辑出错。

解决方案：PCen 信号应该由 Beq 和 ALU 相等信号与，再或上 ALU 相等的非和 Bne 的与，这样生成，重新根据逻辑连接电路即可。

(4.2)

故障现象 2：地址转移逻辑测试不通过

原因分析：错误看错状态，控制器地址转移逻辑生成表达式不正确

解决方案：输入正确的状态，重新通过 Excel 自动生成