

# 华中科技大学

## 计算机组成原理实验报告

实验名称: **CPU 设计实验**

班 级: **人工智能 2204 班**

学 号: **U202214123**

姓 名: **陈博**

报告日期: **2024 年 6 月 2 日**

# 目录

1. 实验内容及目的.....	3
1.1 多周期 MIPS CPU（微程序控制器）设计实验.....	3
1.2 多周期 MIPS CPU（硬布线控制器）实验.....	3
2. 实验电路、设计思路及结果.....	3
2.1 多周期 MIPS（微程序控制器）设计实验.....	3
2.2 多周期 MIPS（硬布线控制器）设计实验.....	8
3. 实验中遇到的问题.....	10
3.1 多周期 MIPS（微程序控制器）设计实验.....	10
3.2 多周期 MIPS（硬布线、微程序控制器）设计实验.....	11
4. 实验总结.....	11

# 1. 实验内容及目的

## 1.1 多周期 MIPS CPU（微程序控制器）设计实验

理解多周期 MIPS CPU 的基本原理，能在 Logisim 平台中设计实现多周期 MIPS CPU，使用微程序控制器，实现 8 条核心指令，能运行冒泡排序测试程序。

## 1.2 多周期 MIPS CPU（硬布线控制器）实验

理解多周期 MIPS CPU 的基本原理，能在 Logisim 平台中设计实现多周期 MIPS CPU，使用硬布线控制器，实现 8 条核心指令，能运行冒泡排序测试程序。

# 2. 实验电路、设计思路及结果

## 2.1 多周期 MIPS（微程序控制器）设计实验

多周期 MIPS CPU 设计的顶层设计图如图 1 所示，指令由程序计数器 PC 提供指令的地址，在内存中访问存储单元取出指令到指令计数器，通过指令译码，识别指令的操作码和操作数，进而从通用寄存器组或者主存中取出操作数，然后通过不同的操作方式，进行移位或者加减，最后存入主存或者通用寄存器组。而本次实验，要实现的是产生各个部件控制信号的控制

### 多周期MIPS CPU实验 — 顶层视图

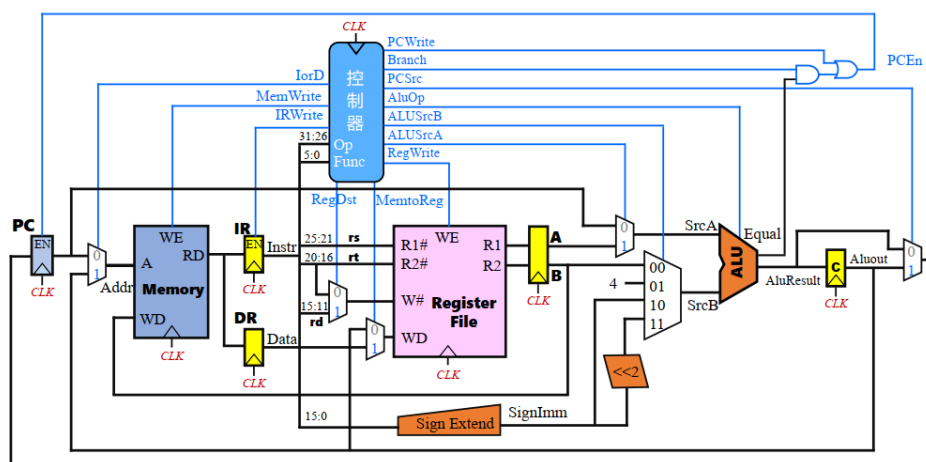


图 1 多周期 MIPS CPU 设计的顶层设计图

微程序控制器是利用软件思想设计硬件的技术，将存储技术和程序设计相结合，回避复杂时序逻辑设计。具体是通过将控制信号序列像程序一样存储起来，然后依序执行微程序即可生成控制信号序列。为了设计微程序，需要列出程序转移图，如图 2 所示。微程序控制器的顶层设计图如图 3 所示。

多周期MIPS状态图

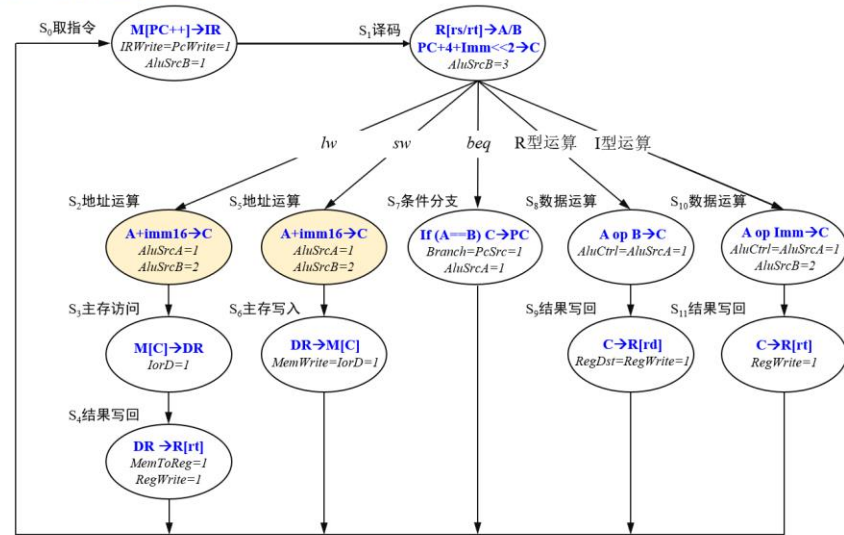


图 2 多周期 MIPS 状态图

# 多周期MIPS CPU实验 — 构建微程序控制器

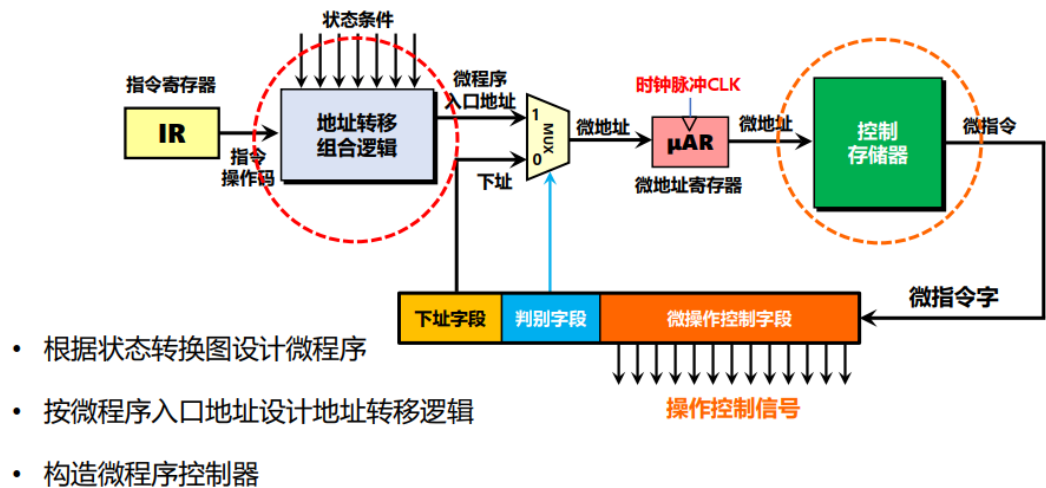


图 3 微程序控制器的顶层设计图

首先是控制存储器的设计，控制存储器存放的是微指令，微指令是由控制信号序列、测试信号 P 和下址字段组成，这些都可以由状态转移图得出，通过将

微指令功能	状态	微指令地址	IoRD	PcSrc	AluSrcA	AluSrcB	MemToReg	RegDst	IrWrite	PcWrite	RegWrite	MemWrite	MemRead	BEQ	BNE	AluControl	P	下址字段	微指令	十六进制		
取指令	0	0000	0	0	0	01	0	0	1	1	0	0	0	1	0	0	0	0	0001	0000100110010000000001	13201	
译码	1	0001	0	0	0	11	0	0	0	0	0	0	0	0	0	0	0	1	0000	000110000000000000000000	30010	
LW1	2	0010	00	0	1	10	0	0	0	0	0	0	0	0	0	0	0	0	0011	000110000000000000000011	60003	
LW2	3	0011	1	0	0	00	0	0	0	0	0	0	1	0	0	0	0	0	0100	10000000001000000100	100204	
LW3	4	0100	0	0	0	00	1	0	0	0	1	0	0	0	0	0	0	0	0000	000001000110000000000000	8800	
SW1	5	0101	0	0	1	10	0	0	0	0	0	0	0	0	0	0	0	0	0110	00110000000000000000110	60006	
SW2	6	0110	1	0	0	00	0	0	0	0	0	1	0	0	0	0	0	0	0000	1000000000100000000000	100400	
R型运算1	7	0111	0	0	1	00	0	0	0	0	0	0	0	0	0	0	0	10	0	1000	001000000000000010010000	40048
R型运算2	8	1000	0	0	0	00	0	1	0	0	1	0	0	0	0	0	0	0	0000	0000001001000000000000	4800	
BEQ	9	1001	0	1	1	00	0	0	0	0	0	0	0	0	1	0	01	0	0000	0110000000001001000000	C0120	
BNE	10	1010	0	1	1	00	0	0	0	0	0	0	0	0	0	1	01	0	0000	011000000000101000000	C00A0	
ADDI1	11	1011	0	0	1	10	0	0	0	0	0	0	0	0	0	0	0	0	1100	00110000000000000001100	6000C	
ADDI2	12	1100	0	0	0	00	0	0	0	0	1	0	0	0	0	0	0	0	0000	0000000001000000000000	800	
SYSCALL	13	1101	0	0	0	00	0	0	0	0	0	0	0	0	0	0	0	0	1101	00000000000000000000101	D	

接下来是地址转移逻辑的设计，通过状态转化图，我么可以知道各个微程序的入口地址编号，输入到微程序地址转移逻辑自动生成 Excel 中，就产生了微程序入口地址，如图 5 所示，最后将最后的生成逻辑复制到软件中即可，如图 6 所示。

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
机器指令译码信号										微程序入口地址					
R_Type	ADDI	LW	SW	BEQ	BNE	SYSCALL	xxx1	xxx2	xxx3	入口地址 10进制	S4	S3	S2	S1	S0
1										7	0	0	1	1	1
	1									11	0	1	0	1	1
		1								2	0	0	0	1	0
			1							5	0	0	1	0	1
				1						9	0	1	0	0	1
					1					10	0	1	0	1	0
						1				13	0	1	1	0	1

[illegible]

5

产生了地址转移逻辑后，将转移逻辑输入到自动生成电路中，如图 7 所示。  
产生了图 8 所示的地址转移逻辑电路图。



图 7 地址转移逻辑生成

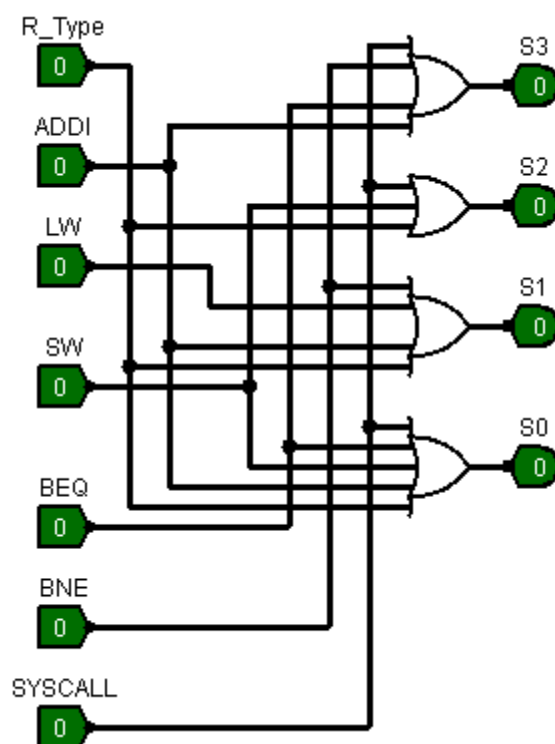


图 8 地址转移逻辑电路图

最后，将他们组合到一起，产生多周期微程序控制器，电路图如图 9 所示，  
然后，根据多周期 MIPS 的顶层设计图，对整体框架进行连线，最后的实验电

路图如图 10 所示，进行冒泡排序，实验结果如图 11 所示。

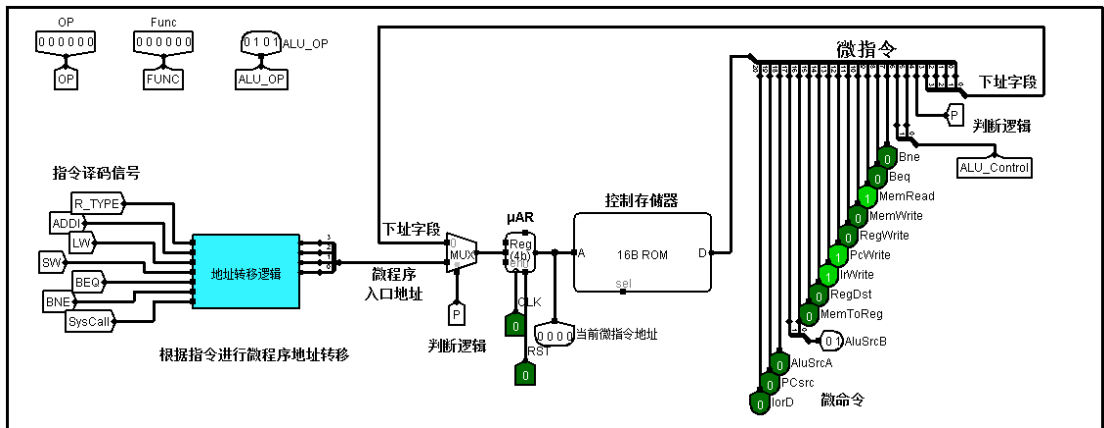


图 10 多周期 MIPS 控制存储器的设计电路图

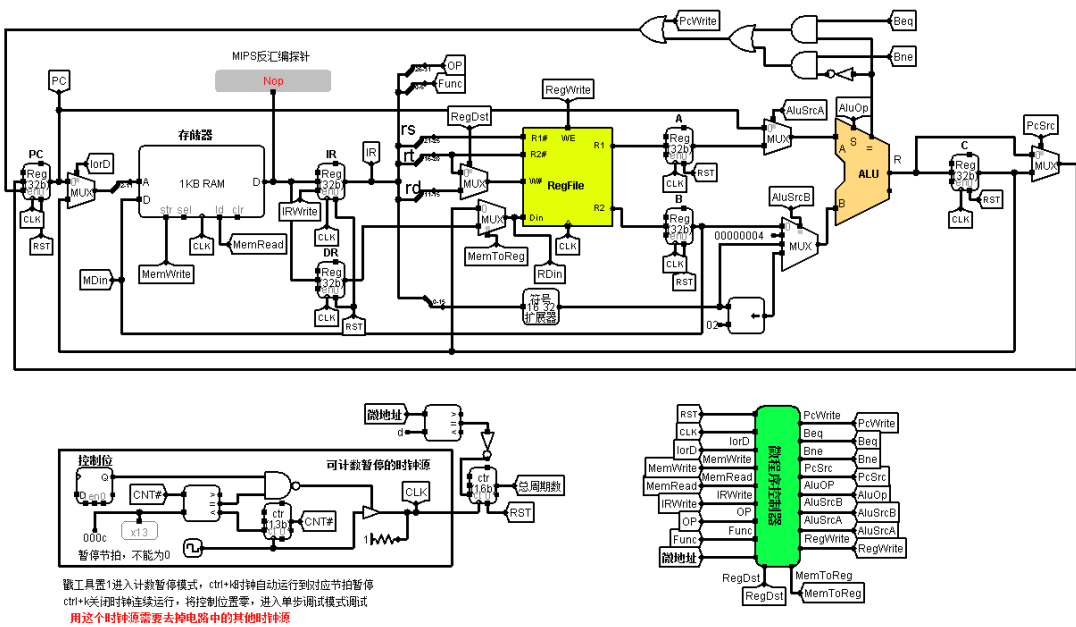
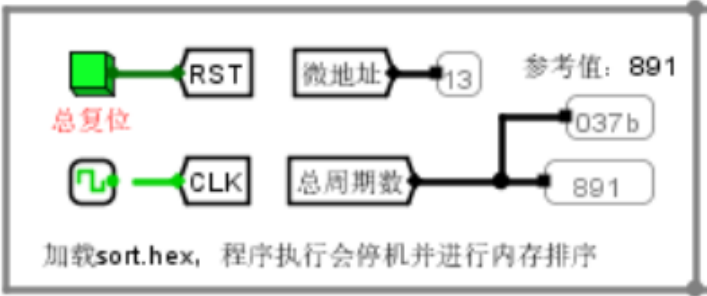


图 10 多周期 MIPS（微程序控制器）设计电路图



080 00000006 00000005 00000004 00000003 00000002 00000001 00000000 ffffffff

图 11 多周期 MIPS（微程序控制器）设计实验结果

## 2.2 多周期 MIPS（硬布线控制器）设计实验

硬布线控制器的设计原理是对每一个状态进行编号，每一个状态对应单独的控制信号序列，因此执行一条指令需要多少时钟周期就安排多少时钟周期，不会存在任何时钟节拍的浪费。控制信号由现态产生，而次态由状态机产生。因此，本次实验中我们需要设计的就是控制器的状态转移逻辑。

根据状态转化图，将每个状态的现态和次态分别输入到硬布线控制器状态转换逻辑自动生成的 Excel 中，结果如图 12 和图 13 所示。

当前状态(现态)					输入信号							下一状态 (次态)				
S3	S2	S1	S0	现态 10进制	R_Type	LW	SW	BEQ	BNE	SYSCALL	ADDI	次态 10进制	N3	N2	N1	N0
0	0	0	0	0								1	0	0	0	1
0	0	0	1	1	1							7	0	1	1	1
0	0	0	1	1		1						2	0	0	1	0
0	0	0	1	1			1					5	0	1	0	1
0	0	0	1	1				1				9	1	0	0	1
0	0	0	1	1					1			10	1	0	1	0
0	0	0	1	1						1		13	1	1	0	1
0	0	0	1	1							1	11	1	0	1	1
0	0	1	0	2								3	0	0	1	1
0	0	1	1	3								4	0	1	0	0
0	1	0	0	4								0	0	0	0	0
0	1	0	1	5								6	0	1	1	0
0	1	1	0	6								0	0	0	0	0
0	1	1	1	7								8	1	0	0	0
1	0	0	0	8								0	0	0	0	0
1	0	0	1	9								0	0	0	0	0
1	0	1	0	10								0	0	0	0	0
1	0	1	1	11								12	1	1	0	0
1	1	0	0	12								0	0	0	0	0
1	1	0	1	13								13	1	1	0	1

图 12 硬布线控制器状态转换逻辑自动生成 Excel

S3	S2	S1	S0	R_Type	LW	SW	BEQ	BNE	SYSCALL	ADDI	最小项表达式	N3	N2	N1	N0
$\sim S3 \& \sim S2 \& \sim S1 \& \sim S0$											$\sim S3 \& \sim S2 \& \sim S1 \& \sim S0$				$\sim S3 \& \sim S2 \& \sim S1 \& \sim S0$
$\sim S3 \& \sim S2 \& \sim S1 \& S0$				$R\_Type$							$\sim S3 \& \sim S2 \& \sim S1 \& S0 \& R\_Type$				$\sim S3 \& \sim S2 \& \sim S1 \& S0 \& R\_Type$
$\sim S3 \& \sim S2 \& \sim S1 \& S0$					$LW$						$\sim S3 \& \sim S2 \& \sim S1 \& S0 \& LW$				$\sim S3 \& \sim S2 \& \sim S1 \& S0 \& LW$
$\sim S3 \& \sim S2 \& \sim S1 \& S0$						$SW$					$\sim S3 \& \sim S2 \& \sim S1 \& S0 \& SW$				$\sim S3 \& \sim S2 \& \sim S1 \& S0 \& SW$
$\sim S3 \& \sim S2 \& \sim S1 \& S0$							$BEQ$				$\sim S3 \& \sim S2 \& \sim S1 \& S0 \& BEQ$				$\sim S3 \& \sim S2 \& \sim S1 \& S0 \& BEQ$
$\sim S3 \& \sim S2 \& \sim S1 \& S0$								$BNE$			$\sim S3 \& \sim S2 \& \sim S1 \& S0 \& BNE$				$\sim S3 \& \sim S2 \& \sim S1 \& S0 \& BNE$
$\sim S3 \& \sim S2 \& \sim S1 \& S0$									$SYSCALL$		$\sim S3 \& \sim S2 \& \sim S1 \& S0 \& SYSCALL$				$\sim S3 \& \sim S2 \& \sim S1 \& S0 \& SYSCALL$
$\sim S3 \& \sim S2 \& \sim S1 \& S0$										$ADDI$	$\sim S3 \& \sim S2 \& \sim S1 \& S0 \& ADDI$				$\sim S3 \& \sim S2 \& \sim S1 \& S0 \& ADDI$
$\sim S3 \& \sim S2 \& S1 \& \sim S0$											$\sim S3 \& \sim S2 \& S1 \& \sim S0$				$\sim S3 \& \sim S2 \& S1 \& \sim S0$
$\sim S3 \& \sim S2 \& S1 \& S0$											$\sim S3 \& \sim S2 \& S1 \& S0$				$\sim S3 \& \sim S2 \& S1 \& S0$
$\sim S3 \& S2 \& \sim S1 \& \sim S0$											$\sim S3 \& S2 \& \sim S1 \& \sim S0$				$\sim S3 \& S2 \& \sim S1 \& \sim S0$
$\sim S3 \& S2 \& \sim S1 \& S0$											$\sim S3 \& S2 \& \sim S1 \& S0$				$\sim S3 \& S2 \& \sim S1 \& S0$
$\sim S3 \& S2 \& S1 \& \sim S0$											$\sim S3 \& S2 \& S1 \& \sim S0$				$\sim S3 \& S2 \& S1 \& \sim S0$
$\sim S3 \& S2 \& S1 \& S0$											$\sim S3 \& S2 \& S1 \& S0$				$\sim S3 \& S2 \& S1 \& S0$
$S3 \& \sim S2 \& \sim S1 \& \sim S0$											$S3 \& \sim S2 \& \sim S1 \& \sim S0$				$S3 \& \sim S2 \& \sim S1 \& \sim S0$
$S3 \& \sim S2 \& \sim S1 \& S0$											$S3 \& \sim S2 \& \sim S1 \& S0$				$S3 \& \sim S2 \& \sim S1 \& S0$
$S3 \& \sim S2 \& S1 \& \sim S0$											$S3 \& \sim S2 \& S1 \& \sim S0$				$S3 \& \sim S2 \& S1 \& \sim S0$
$S3 \& \sim S2 \& S1 \& S0$											$S3 \& \sim S2 \& S1 \& S0$				$S3 \& \sim S2 \& S1 \& S0$
$S3 \& S2 \& \sim S1 \& \sim S0$											$S3 \& S2 \& \sim S1 \& \sim S0$				$S3 \& S2 \& \sim S1 \& \sim S0$
$S3 \& S2 \& \sim S1 \& S0$											$S3 \& S2 \& \sim S1 \& S0$				$S3 \& S2 \& \sim S1 \& S0$
$S3 \& S2 \& S1 \& \sim S0$											$S3 \& S2 \& S1 \& \sim S0$				$S3 \& S2 \& S1 \& \sim S0$
$S3 \& S2 \& S1 \& S0$											$S3 \& S2 \& S1 \& S0$				$S3 \& S2 \& S1 \& S0$
$S3 \& S2 \& \sim S1 \& \sim S0$											$S3 \& S2 \& \sim S1 \& \sim S0$				$S3 \& S2 \& \sim S1 \& \sim S0$

图 13 硬布线控制器状态转换逻辑自动生成结果



最后，根据这些转化逻辑，输入到 Logisim 组合逻辑自动生成的功能中，我们最后得到了硬布线控制器状态转移逻辑的电路图，如图 14 所示，整个多周期 MIPS 实验电路图如图 15 所示，实验结果如图 16 所示。

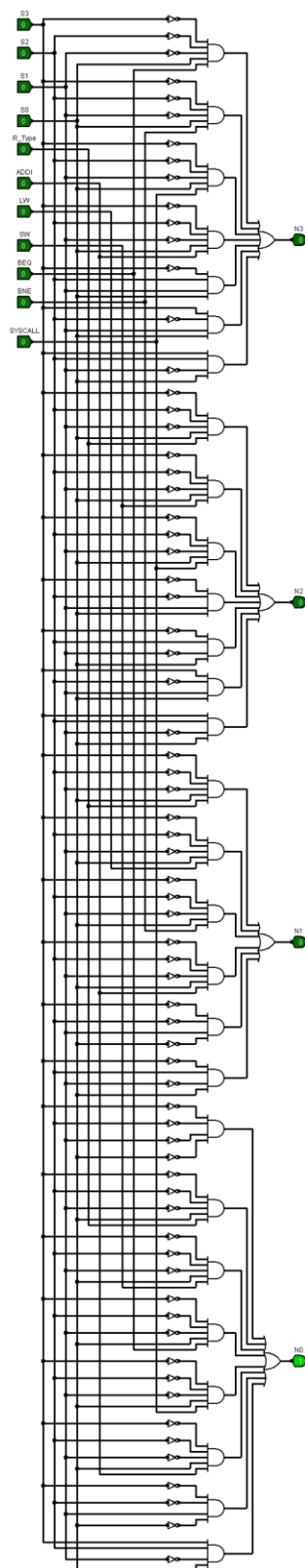


图 14 硬布线控制器状态转移逻辑电路图

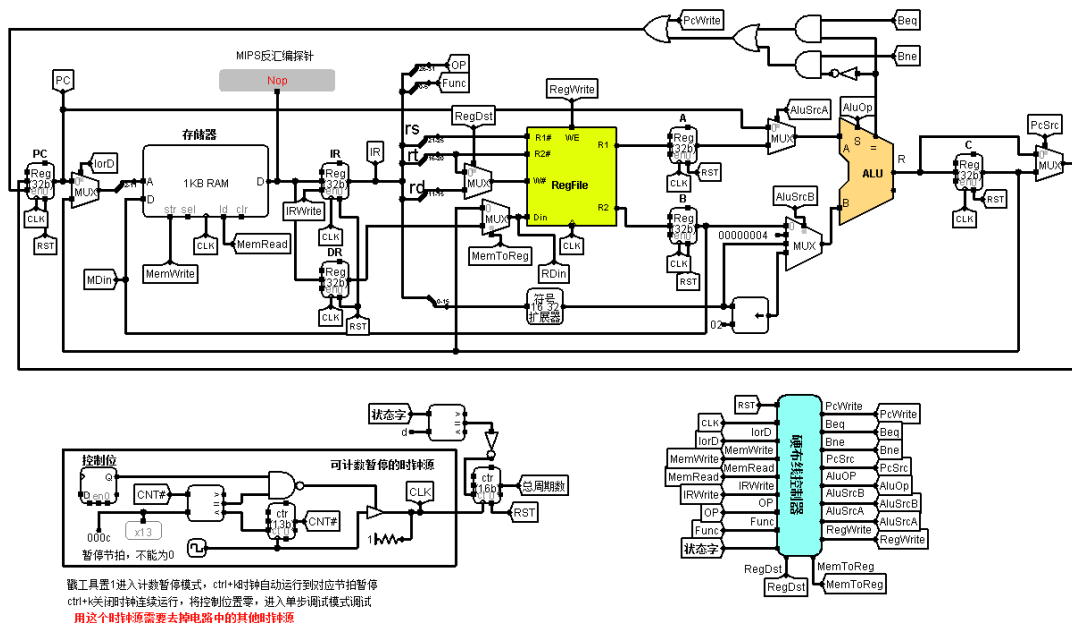
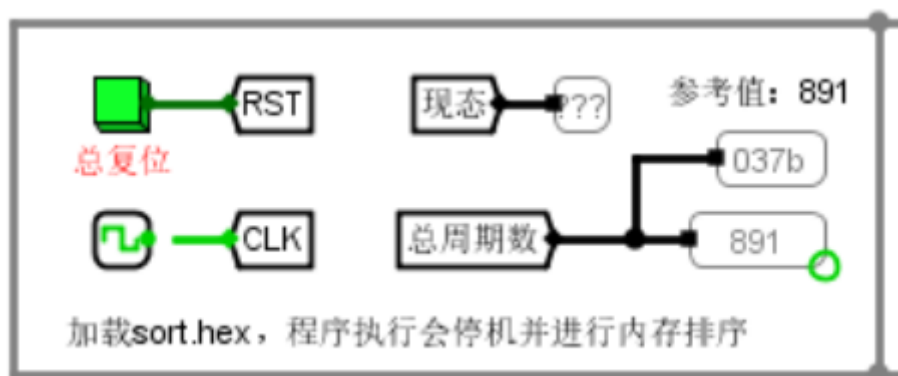


图 15 多周期 MIPS（硬布线控制器）设计电路图



080 00000006 00000005 00000004 00000003 00000002 00000001 00000000 ffffffff

图 16 多周期 MIPS（硬布线控制器）实验结果

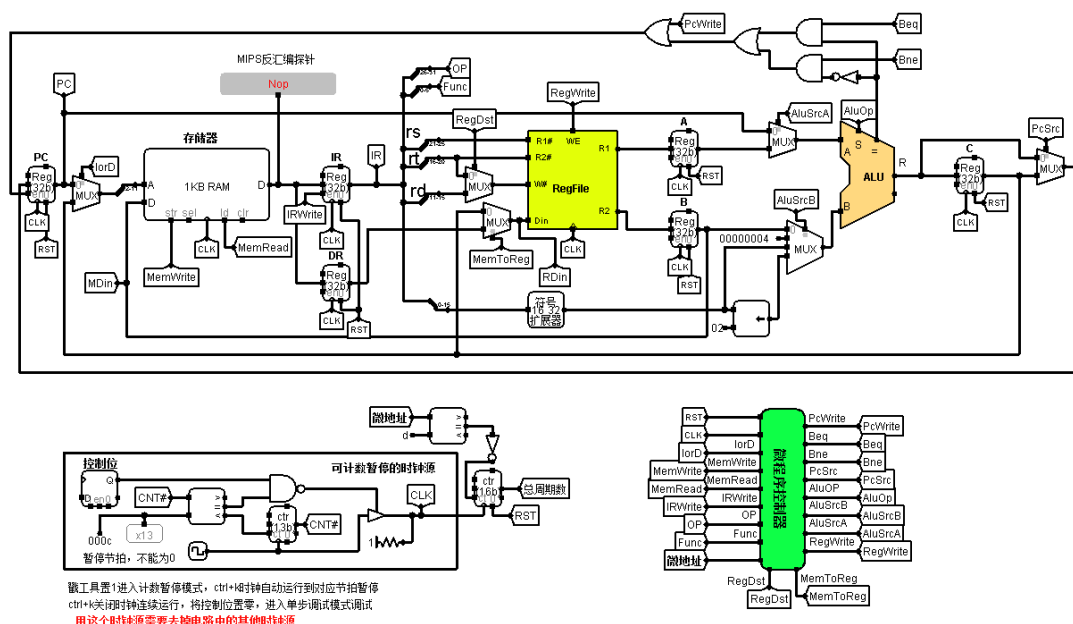
### 3. 实验中遇到的问题

#### 3.1 多周期 MIPS（微程序控制器）设计实验

在自动生成组合逻辑电路的时候，使用这个功能不太熟悉，导致生成的电路与原电路重合，将原电路覆盖掉了，后来通过重新拷贝一份作业，重新生成电路，更改生成电路名字才解决这个问题。

## 3.2 多周期 MIPS（硬布线、微程序控制器）设计实验

在连接顶层电路的时候，由于电路比较复杂，第一次连玩漏了好多线，比如 RST，然后也漏了一些隧道，而且有些逻辑门看错了，还是第一次实验的小毛病，然后，通过调试，看看数据通路哪里该通过的信号没有通过，将这些线重新检查一下，然后调整就很顺利的完成了。



## 4. 实验总结

本次实验是最后一次计算机组成原理实验，这此实验的工作量主要来源于顶层的连线，由于组合逻辑自动生成的 Excel 已经给好，这一部分相对比较简单，但还是需要细心的连好线，比对好每一个控制信号。理论课之讲过单周期和单总线的 CPU，通过这次实验，让我对多周期 CPU 的原理及设计有了更加深刻的认识。通过这 4 次实验，让我对计算机的组成原理有了更加直观的感受，进一步了解计算机这一伟大发明。