



安徽大學  
Anhui University

# 系統硬件綜合訓練 多周期硬布線CPU

至誠至堅 博學篤行



安徽大学  
Anhui University

# 设计内容

## • 学习慕课 “计算机硬件系统设计”

- (1) 6.1 单周期MIPS架构
- (2) 6.2 多周期MIPS架构 (学习)
- (3) 6.3 单周期MIPS CPU设计 (8条指令)
- (4) 6.4 多周期MIPS CPU微程序控制器设计
- (5) 6.5 多周期MIPS CPU硬布线控制器设计 (学习)
- (6) 6.6单周期MIPS CPU设计 (24条指令)
- (7) 6.7MIPS中断处理机制

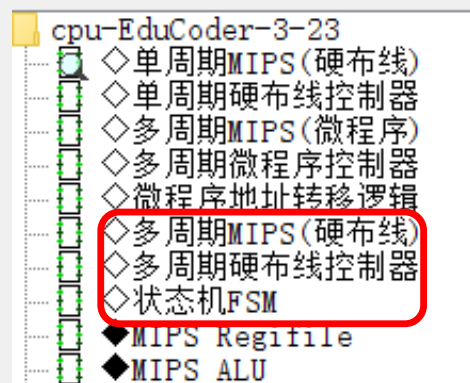
## • 学习慕课 “计算机组成与体系结构”

- (1) 5.5MIPS指令概述
- (2) 5.6MIPS指令详解

## 电路设计:

- (1)多周期MIPS(8指令微程序)
- (2)多周期MIPS(8指令硬布线)
- (3)单周期MIPS(24指令硬布线)
- (4)单周期MIPS(支持中断)

cpu-EduCoder-3-23.circ.circ



至誠至堅 博學篤行



## 多周期硬布线MIPS CPU设计任务：

- 掌握多周期MIPS CPU(8条指令和)硬布线控制器设计的基本原理；
- 设计多周期硬布线8条指令的MIPS CPU(包含在.circ文件中的多周期MIPS(硬布线)电路、多周期硬布线控制器电路、状态机FSM电路)；
- 实验中多周期微操作信号由控制器存储器产生并不是真正意义上的硬布线，需要将该电路进行修改，将控制器用完全的硬布线组合逻辑电路实现（核心是用四位现态值输入，16位控制信号输出的组合逻辑电路去取代控制存储器；注意4入16出电路可以在电路内部分解成例如2个4入8出的子电路，可以用辅助报表文件产生）
- 将sort.hex和自己写一段程序在上述CPU上测试运行。
- 选做：增加新的指令并进行测试。（可选扩展指令集参见后表）

opcode	指令助记符	RTL 功能描述	备注
0 (func:32)	add \$rd, \$rs, \$rt	$R[\$rd] \leftarrow R[\$rs] + R[\$rt]$	RegDest, Aluop=0101, Regwrite
0 (func:12)	syscall	系统调用	PC不变 Halt,Regwrite,RegDest
0 (func:42)	slt \$rd, \$rs, \$rt	$R[\$rd] \leftarrow R[\$rs] < R[\$rt]$	小于置1, 有符号比较, RegDest, Aluop=1011,Regwrite
04	beq \$rs, \$rt, imm	If ( $R[\$rs] = R[\$rt]$ ), $PC \leftarrow PC + 4 + \text{SignExt}_{18b}(\{\text{imm左移2位}\})$	Aluop=1011,Beq, (imm是 <b>字地址</b> , 左移2位转换成字节地址)
05	bne \$rs, \$rt, imm	If ( $R[\$rs] \neq R[\$rt]$ ), $PC \leftarrow PC + 4 + \text{SignExt}_{18b}(\{\text{imm左移2位}\})$	Aluop=1011,Bne, (imm是 <b>字地址</b> , 左移2位转换成字节地址)
08	addi \$rt, \$rs, imm	$R[\$rt] \leftarrow R[\$rs] + \text{SignExt}_{16b}(\text{imm})$	AluSrc, Aluop=0101, Regwrite
35	lw \$rt, imm(\$rs)	$R[\$rt] \leftarrow \text{Mem}(R[\$rs] + \text{SignExt}_{16b}(\text{imm})\text{之和右移2位})$	AluSrc, Aluop=0101, MemToReg, Regwrite, (rs和imm都是 <b>字节地址</b> , 加法和要右移2位转换成字地址才能访问数据M)
43	sw \$rt, imm(\$rs)	$\text{Mem}(R[\$rs] + \text{SignExt}_{16b}(\text{imm})\text{之和右移2位}) \leftarrow R[\$rt]$	AluSrc,Aluop=0101, MemWrite, (rs和imm都是 <b>字节地址</b> , 加法和要右移2位转换成字地址才能访问数据M)

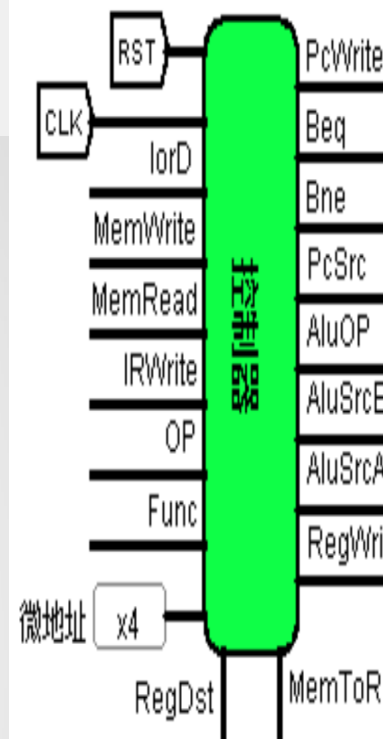
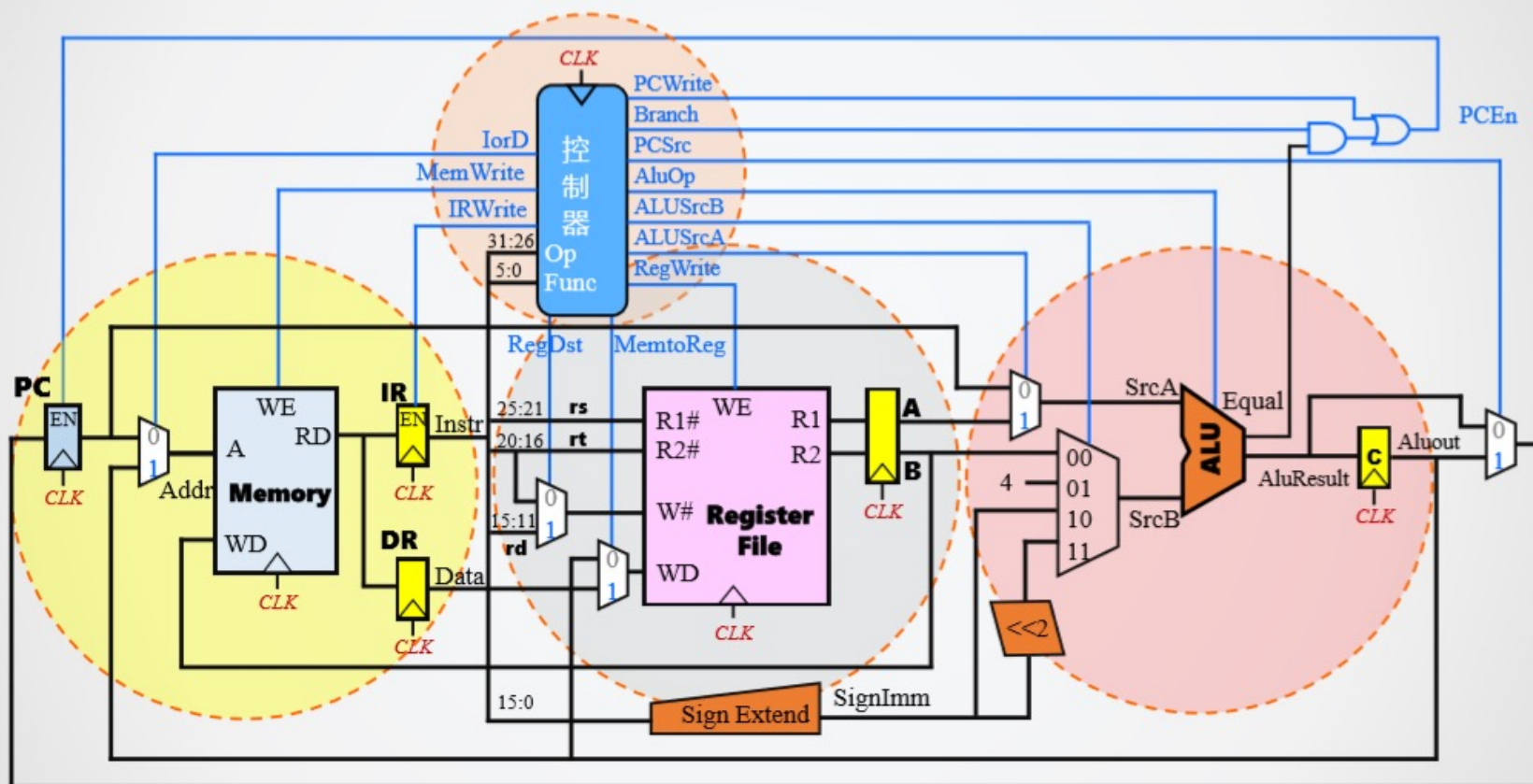
**备注：** 跳转指令偏移地址imm求解公式： 目标字节地址=当前地址+4+imm\*4



安徽大学  
Anhui University

多周期8指令CPU通路图（同时适合微程序和硬布线控制器）

## 多周期MIPS CPU数据通路

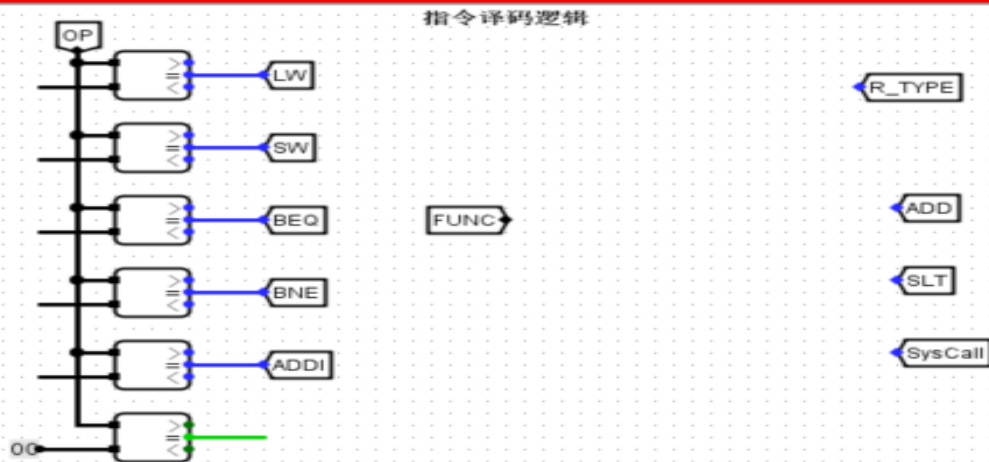
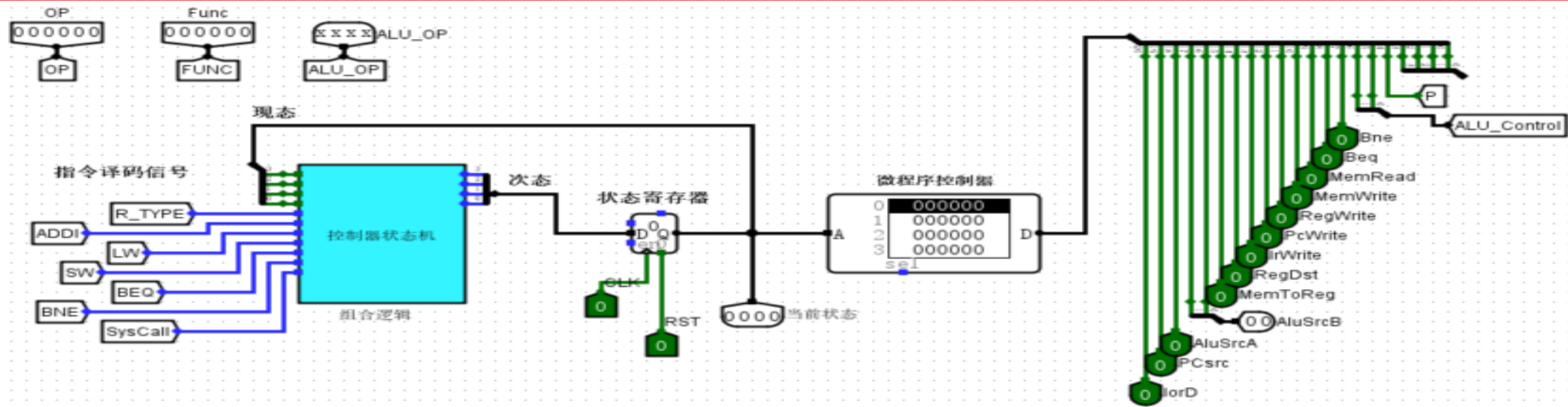






## 控制信号功能说明 (8条核心指令集)

#	控制信号	信号说明	产生条件
1	PCWrite	PC写使能控制	取指令周期, 分支指令执行
2	IorD	指令还是数据	0表示指令, 1表示数据
3	IRwrite	指令寄存器写使能	高电平有效
4	MemWrite	写内存控制信号	sw指令
5	MemRead	读内存控制信号	lw指令 取指令
6	Beq	Beq指令译码信号	Beq指令
7	Bne	Bne指令译码信号	Bne指令
8	PcSrc	PC输入来源	顺序寻址还是跳跃寻址
9	AluOP	运算器操作控制符 4位	ALU_Control控制, 00加, 01减, 10由Funct定
10	AluSrcA	运算器第一输入选择	
11	AluSrcB	运算器第二输入选择	Lw指令, sw指令, addi
12	RegWrite	寄存器写使能控制信号	寄存器写回信号
13	RegDst	写入寄存器选择控制信号	R型指令
14	MemToReg	写入寄存器的数据来自存储器	lw指令



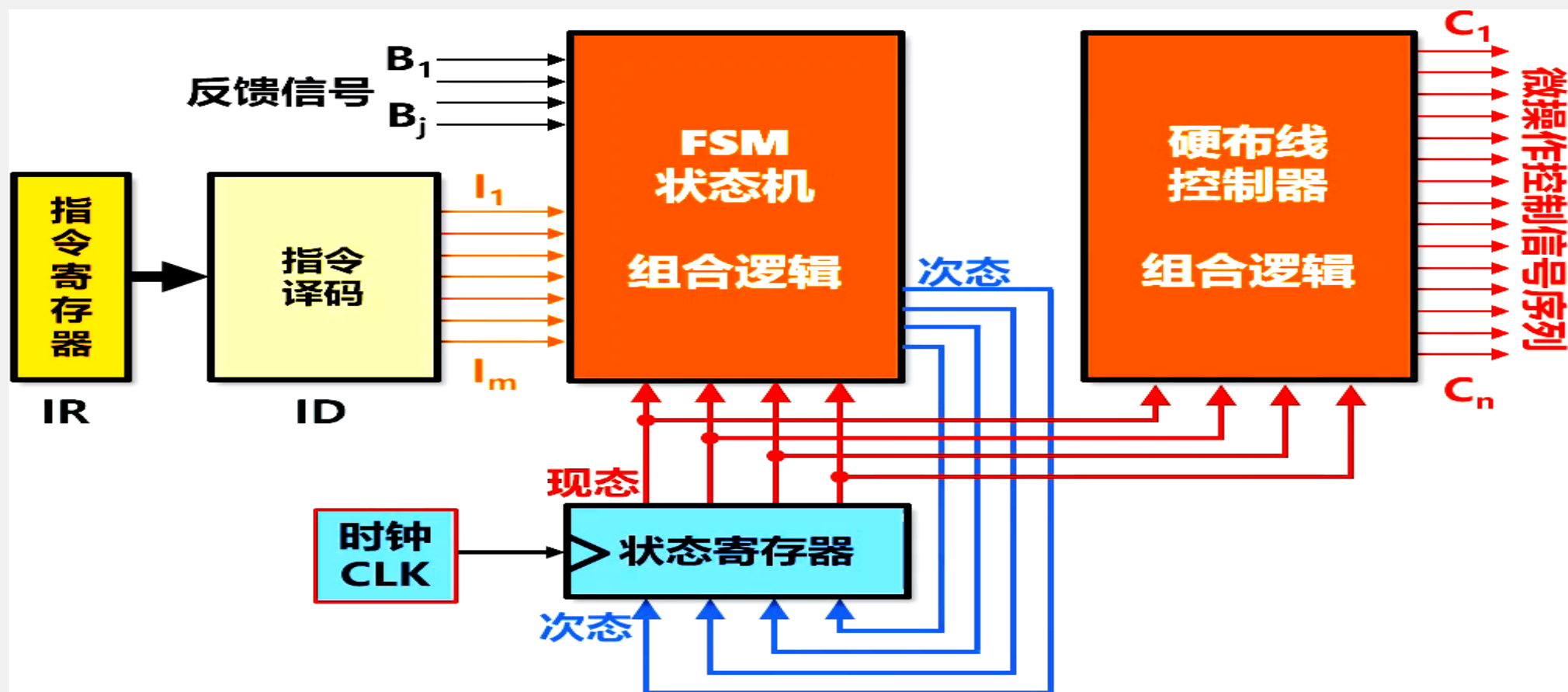
### ALU控制器逻辑

根据ALU\_Control的值决定运算器运算选择控制信号ALU\_OP的值





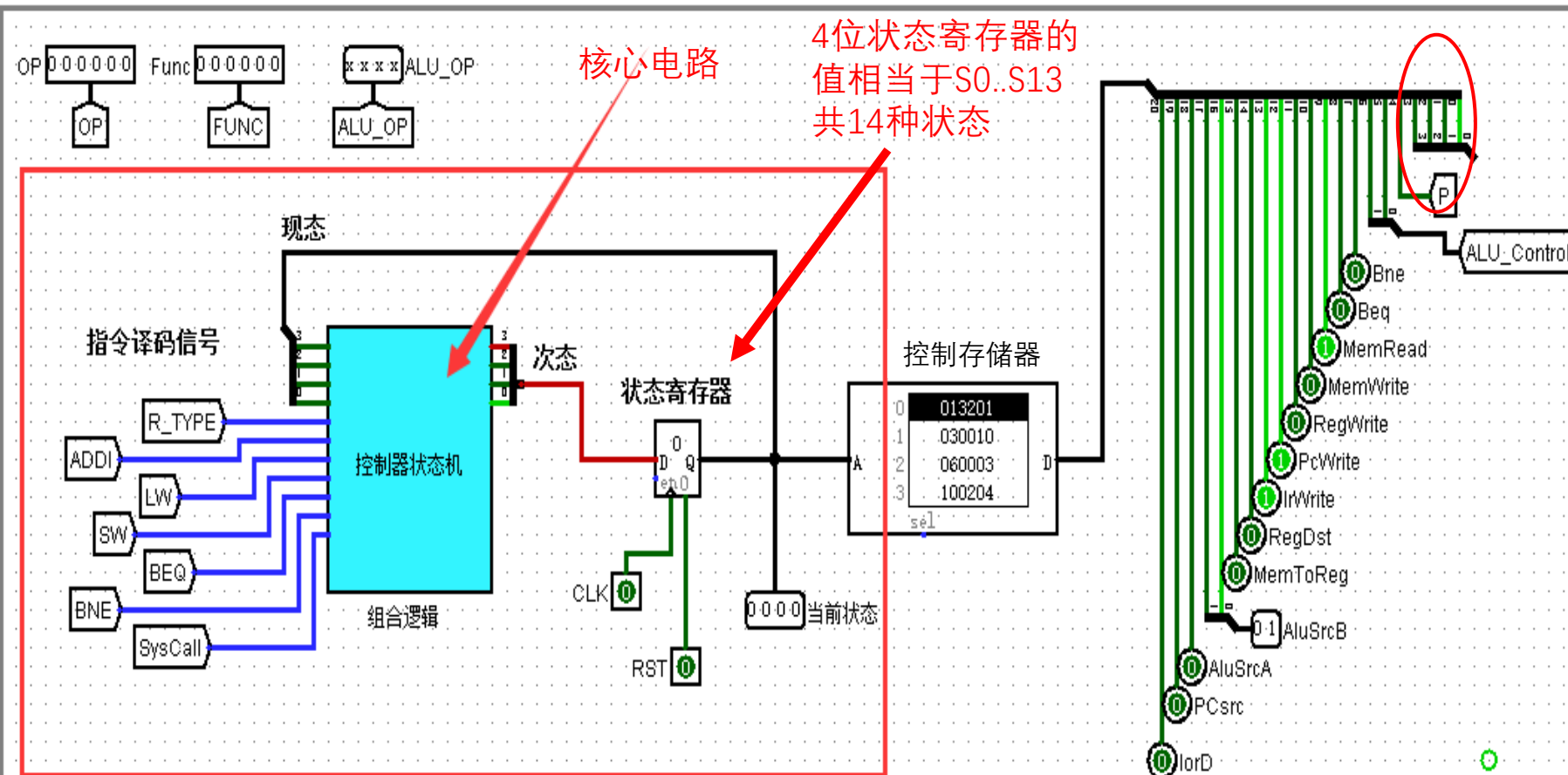
# 硬布线控制器







## 硬布线控制器（可以考虑控存去掉5位）

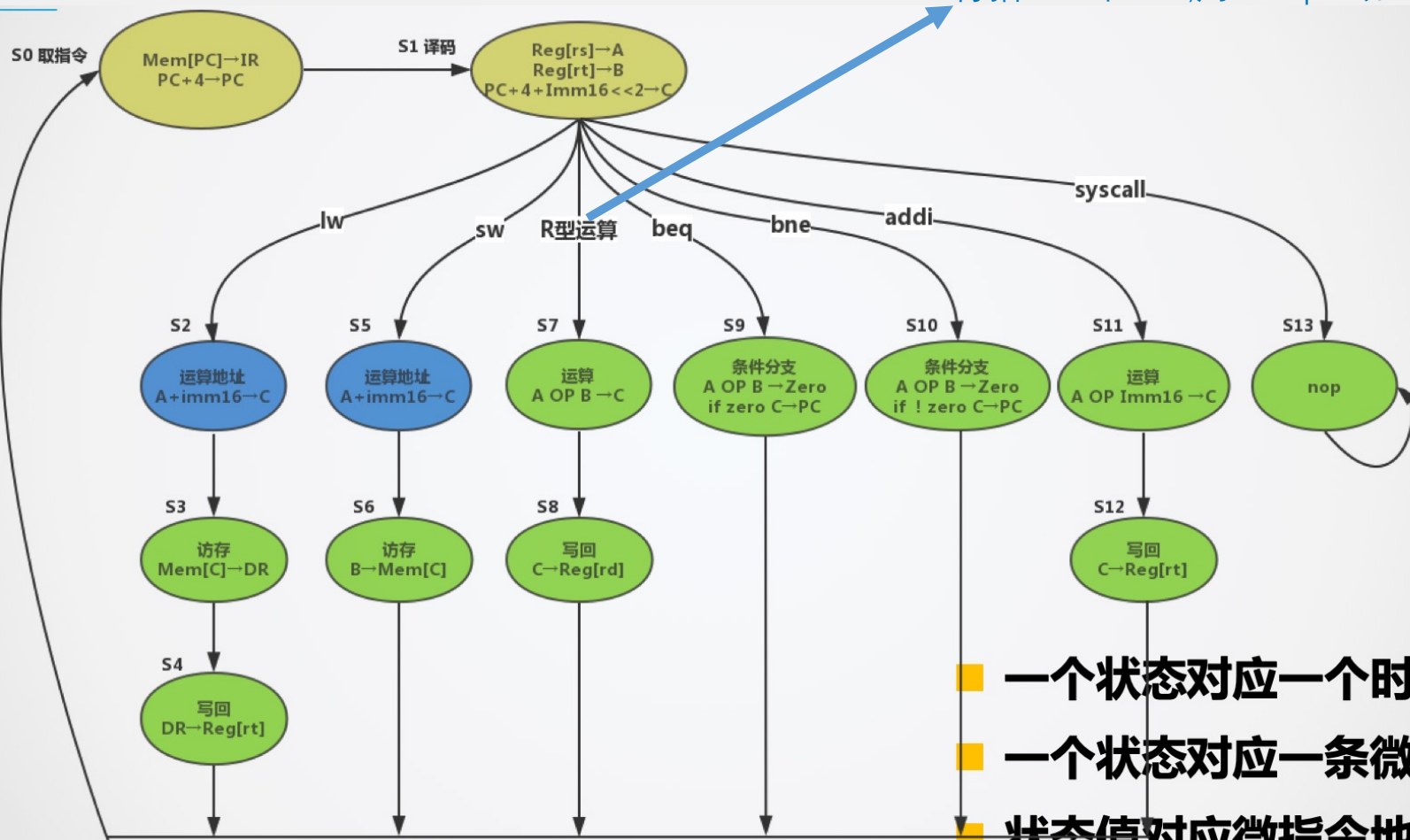


微指令	状态	地址
取指令	S0	0
译码	S1	1
LW1	S2	2
LW2	S3	3
LW3	S4	4
SW1	S5	5
SW2	S6	6
R型运算	S7	7
R型运算	S8	8
Beq	S9	9
Bne	S10	10
ADDI1	S11	11
ADDI2	S12	12
SYSCALL	S13	13
微程序地址		



# 指令状态转换图

特指ADD和SLT,用Aluop区分



微指令	状态	地址
取指令	S0	0
译码	S1	1
LW1	S2	2
LW2	S3	3
LW3	S4	4
SW1	S5	5
SW2	S6	6
R型运算	S7	7
R型运算	S8	8
Beq	S9	9
Bne	S10	10
ADDI1	S11	11
ADDI2	S12	12
SYSCALL	S13	13
微程序地址		

- 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

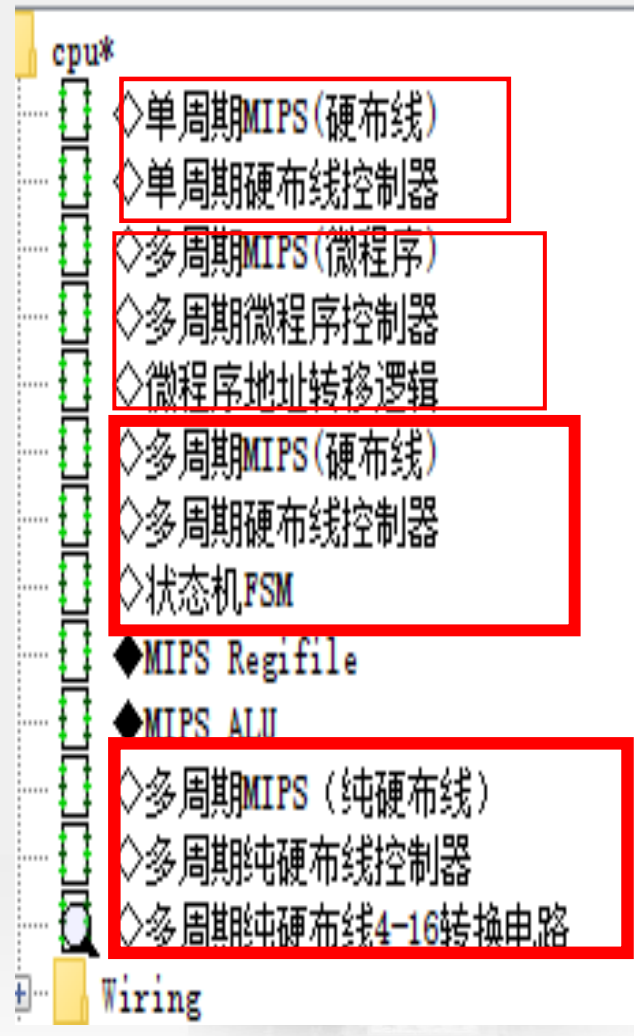
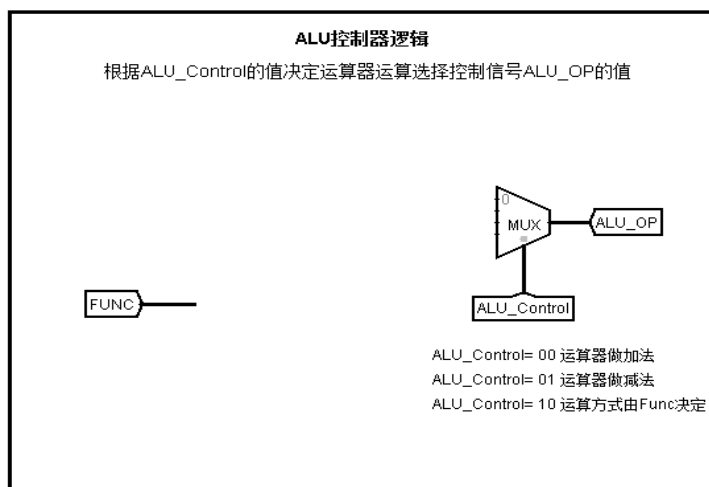
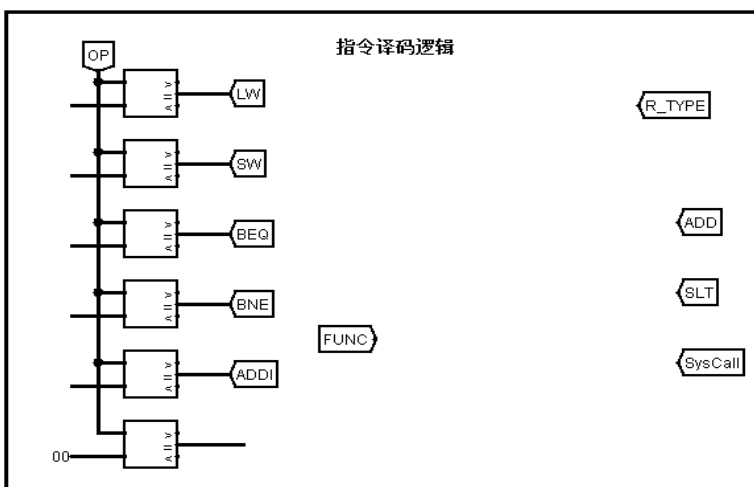
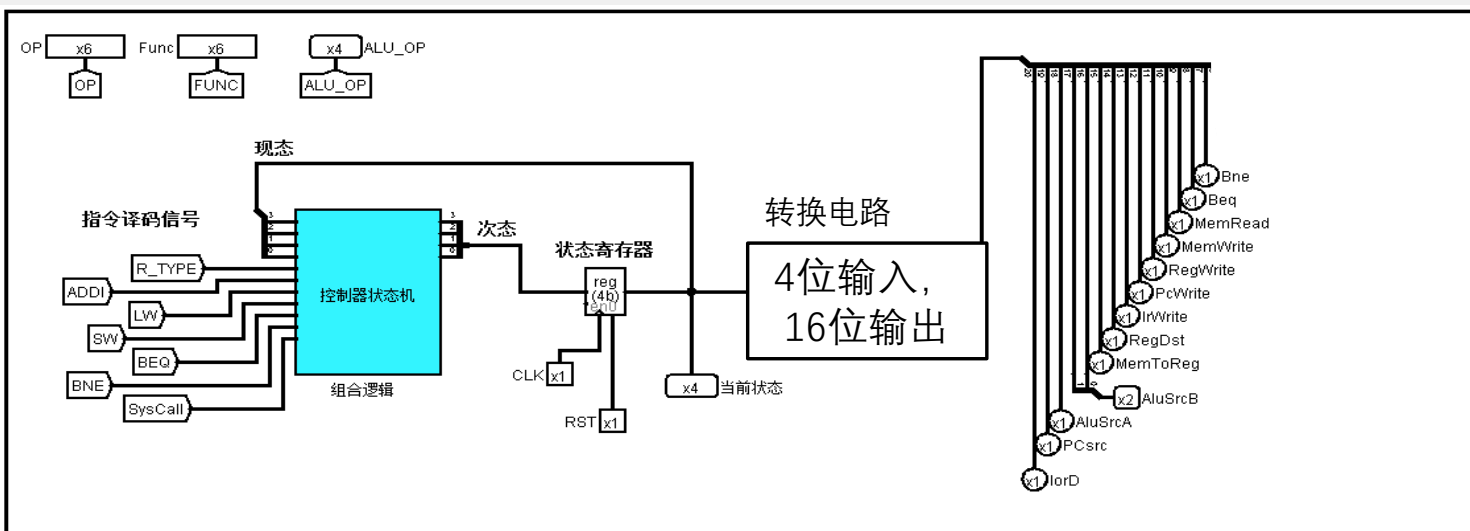


硬布线控制器状态转换逻辑自动生成(2019-4-22).xlsx  
用于产生控制器状态机FSM电路的辅助文件（11入4出电路）

[illegible]



# 纯硬布线控制器





## 多周期硬布线MIPS CPU设计任务：

- 掌握多周期MIPS CPU(8条指令和)硬布线控制器设计的基本原理；
- 设计多周期硬布线8条指令的MIPS CPU(包含在.circ文件中的多周期MIPS(硬布线)电路、多周期硬布线控制器电路、状态机FSM电路)；
- 实验中多周期微操作信号由控制器存储器产生并不是真正意义上的硬布线，需要将该电路进行修改，将控制器用完全的硬布线组合逻辑电路实现（核心是用四位现态值输入，16位控制信号输出的组合逻辑电路去取代控制存储器；注意4入16出电路可以在电路内部分解成例如2个4入8出的子电路，自己定义辅助报表文件产生，并随实验报告打包提交）
- 将sort.hex和自己写一段程序在上述CPU上测试运行。
- 选做：增加新的指令并进行测试。（可选扩展指令集参见后表）





#	指令类型	指令
1	R型指令	移位运算: SLL、SRA、SRL 算术运算: <b>ADD</b> 、ADDU、SUB 逻辑运算: AND、OR、NOR 比较运算: <b>SLT</b> 、SLTU 分支指令: JR 系统调用: <b>SYSCALL</b>
2	I型指令	分支指令: <b>BEQ</b> 、 <b>BNE</b> 立即数运算指令: <b>ADDI</b> 、ADDIU、SLTI、ANDI、ORI 访存指令: <b>LW</b> 、 <b>SW</b>
3	J型指令	• J、JAL