



计算机组成原理

第 16讲

左德承

哈尔滨工业大学计算学部
容错与移动计算研究中心

硬布线与微程序对比

- **硬布线：** 同步逻辑、繁，快，贵，难改
 - 一条指令多个时钟周期、一个时钟周期一个状态、一个状态对应一组并发信号
 - 适合RISC计算机，如MIPS，ARM
- **微程序：** 存储逻辑、简、慢、廉，易改
 - 控制流存储为微指令微程序，一个状态对应一条微指令
 - 一条指令对应多条微指令、状态字等同与存储器地址
 - 适合CISC等功能较复杂的系列机 X86、IBM S/360、DEC VAX
 - 可写控存方便修复出厂故障 Intel Core 2 、Intel Xeon

第 N/A 章 数据通路与时序控制

书上没有，请结合PPT
及参考教材进行学习

NA.1 数据通路

NA.2 设计处理器的步骤

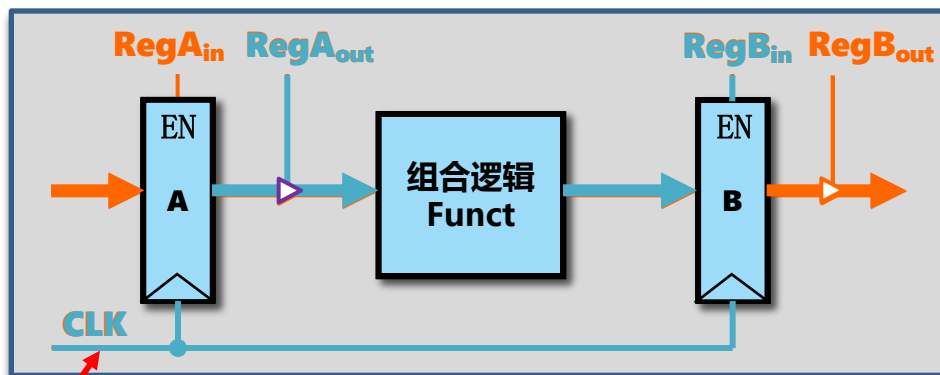
NA.3 单周期MIPS控制器设计

NA.4 多周期MIPS控制器设计

数据通路 (DataPath)

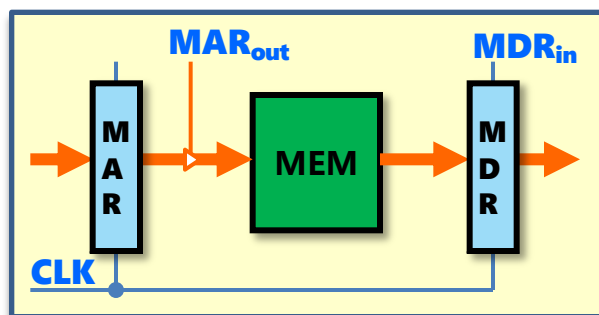
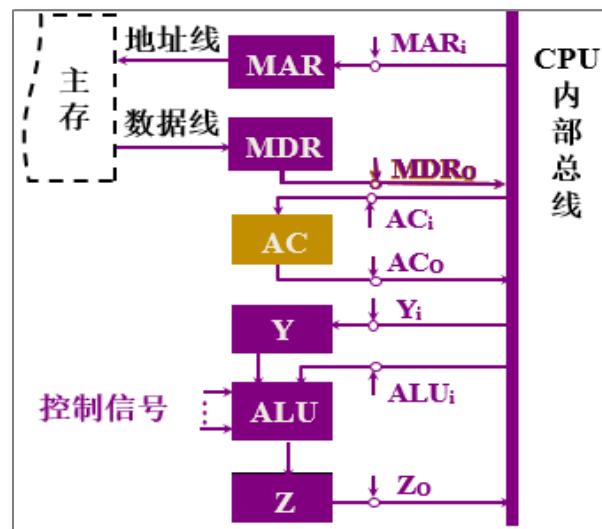
- 数据通路 → 各执行部件间传送信息的路径 (数据流)
 - ◆通路的建立由控制信号控制，受时钟驱动 (控制流)
 - ◆不同指令、同一指令的不同执行阶段：数据通路不同
 - ◆分类：共享通路（总线）、专用通路（非总线）
 - 指令执行流程、执行效率
 - 微操作控制信号的时序安排

数据通路的抽象模型（寄存器传输）

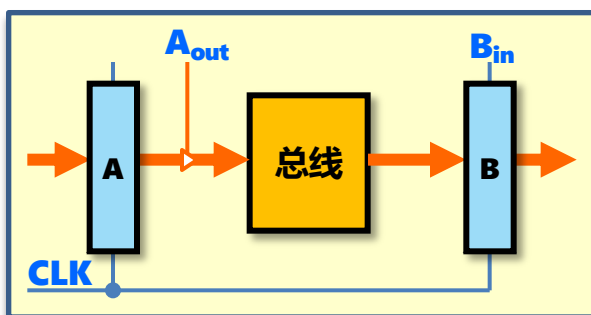


Funct (A) → B

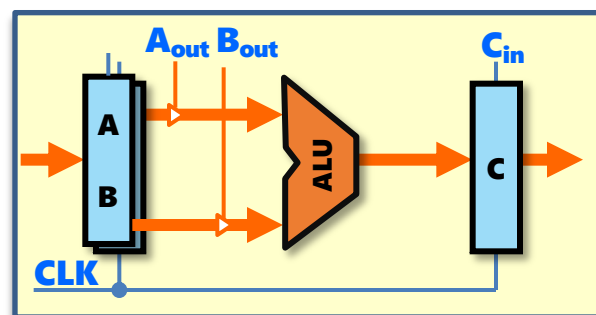
时钟频率?



访存通路

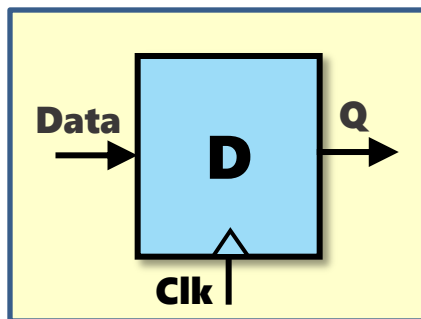


总线传输

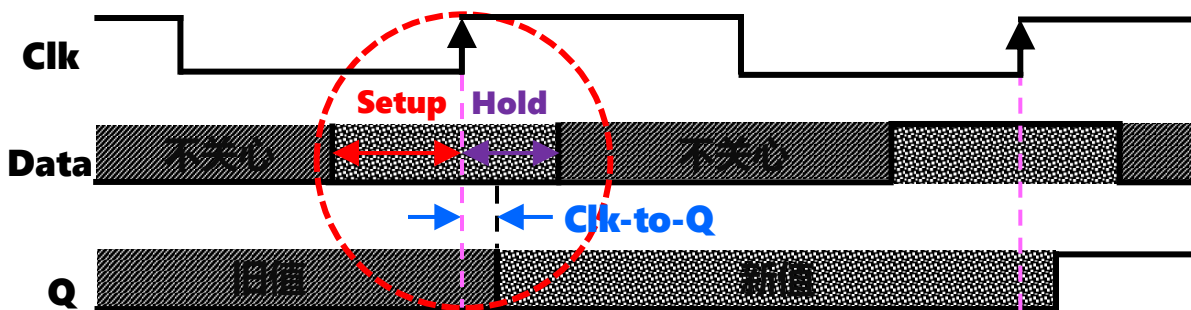


运算通路

时钟频率约束：D触发器定时模型

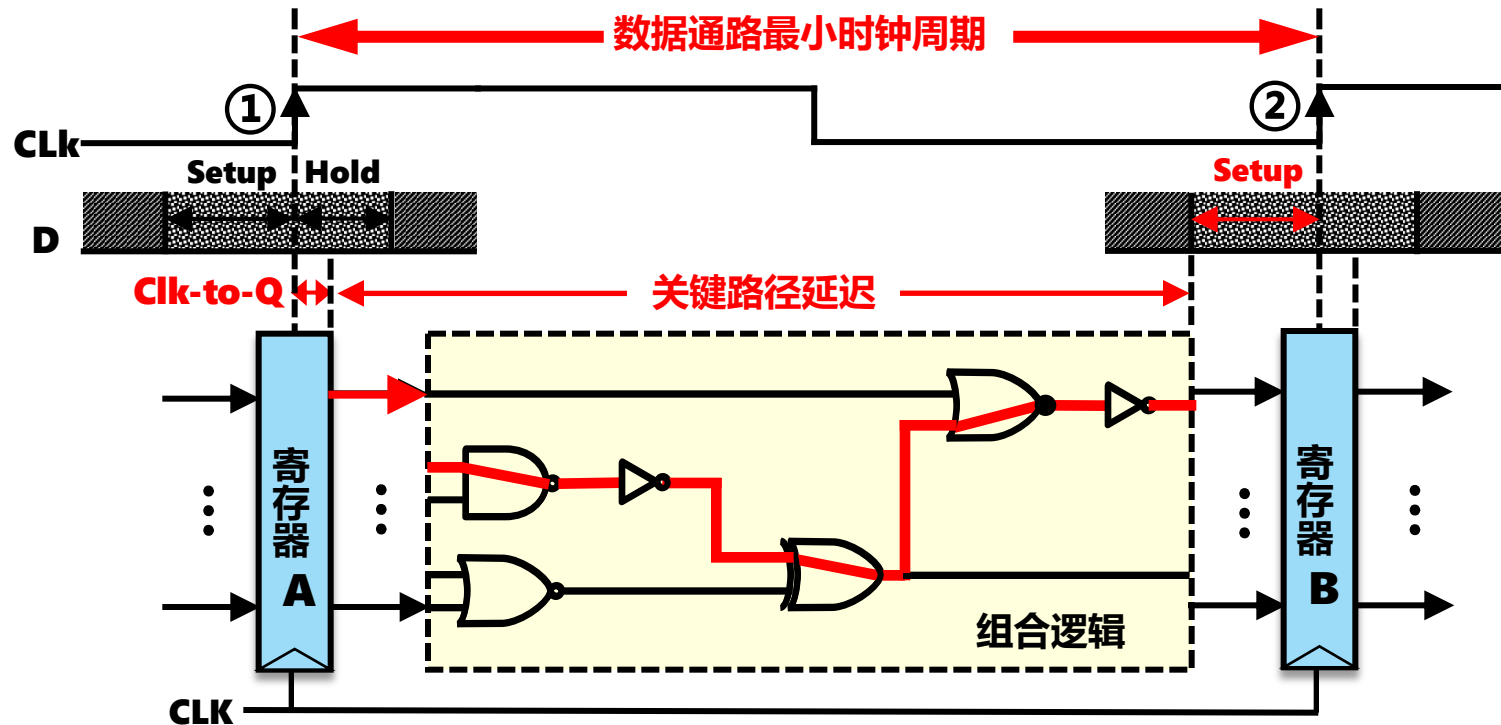


- 时钟触发前输入须稳定一段 **建立时间 (Setup Time)**
- 时钟触发后输入须稳定一段 **保持时间 (Hold Time)**
- 时钟触发到输出稳定的时间 **触发器延迟 (Clk_to_Q)**



数据通路与时钟周期

➔ 确定时钟周期的 **最小值**：



■ 时钟周期 > Clk_to_Q + 关键路径时延 + Setup Time

数据通路分类

- 共享通路（总线型）

- ◆ 主要功能部件连接在**公共的总线**上

- ➔ 各部件通过总线传输数据，结构简单，实现容易

- ➔ 并发性较差，需分时使用总线，效率低

- 专用通路（非总线型）

- ◆ 主要功能部件间有**独立的线路**连接

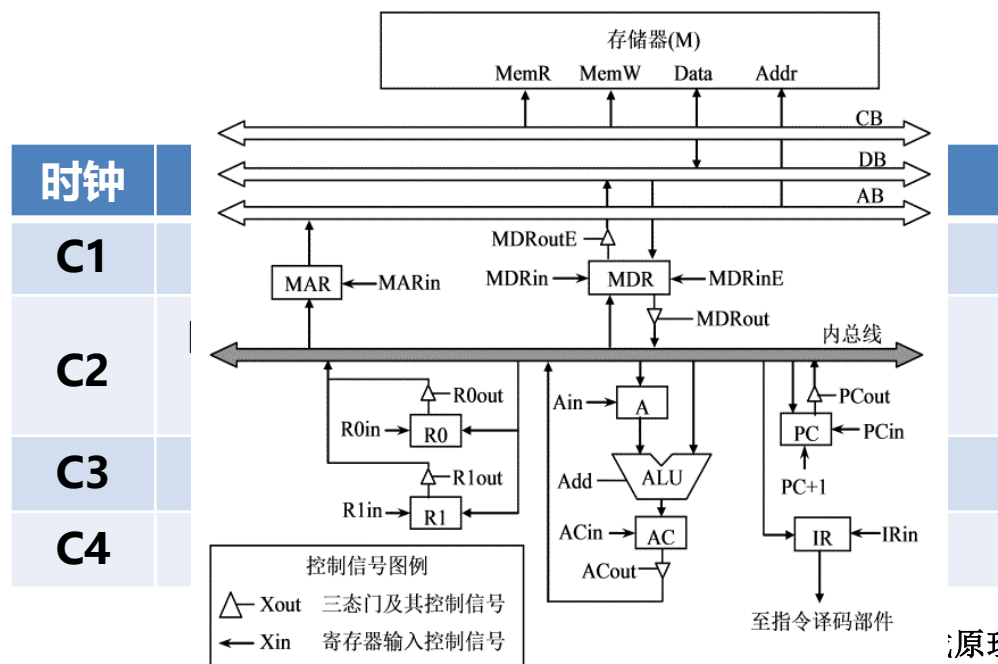
- ➔ 并发度高，性能佳，设计复杂，成本高

- ➔ 可以看作多总线结构

数据通路举例：2009考研真题

- 某机字长16位，指令16位定长；
- 指令 **ADD (R1), R0** 的功能为 $(R0) + ((R1)) \rightarrow (R1)$ ，即将R0中数据与R1内容所指向的主存单元的数据相加，并将结果送入R1内容所指向的主存单元中；
- 数据通路图中控制信号为1表示有效，假设MAR输出一直处于使能状态

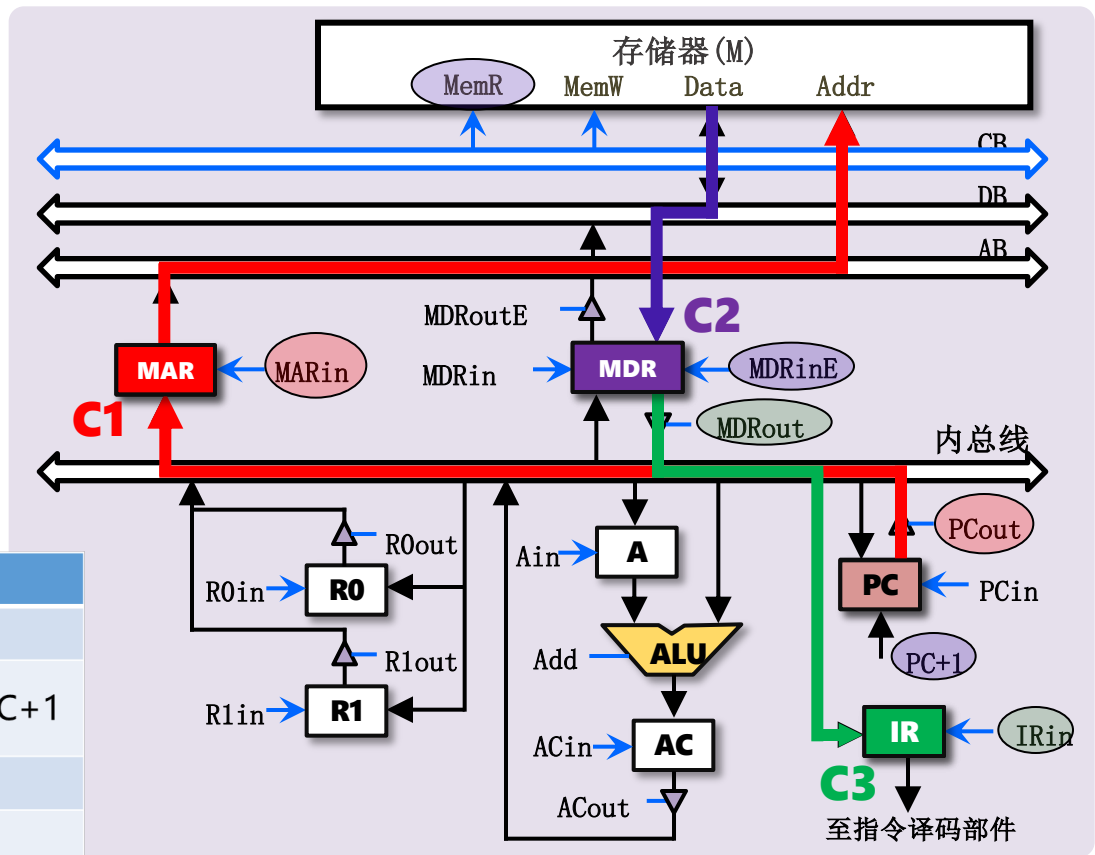
- 右表为取指令和译码阶段每个节拍(时钟周期)的功能和控制信号，请按相同方式给出执行阶段各节拍的功能和有效控制信号。



数据通路举例 → 取指周期

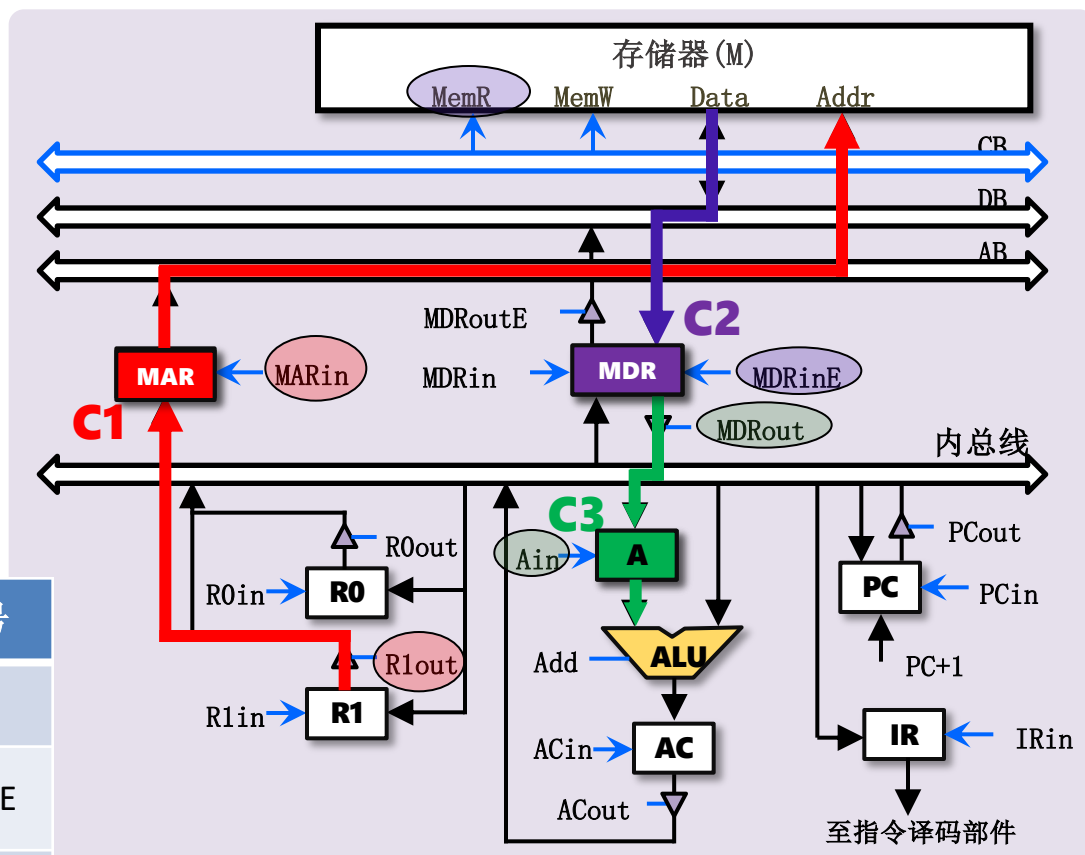
$M[PC++] \rightarrow IR$

时钟	功能	有效控制信号
C1	$MAR \leftarrow (PC)$	Pcout, MARin
C2	$MDR \leftarrow M(MAR)$ $PC \leftarrow (PC) + 1$	MemR, MDRinE, PC+1
C3	$IR \leftarrow (MDR)$	MDRout, IRin
C4	指令译码	无



数据通路举例 → 执行周期

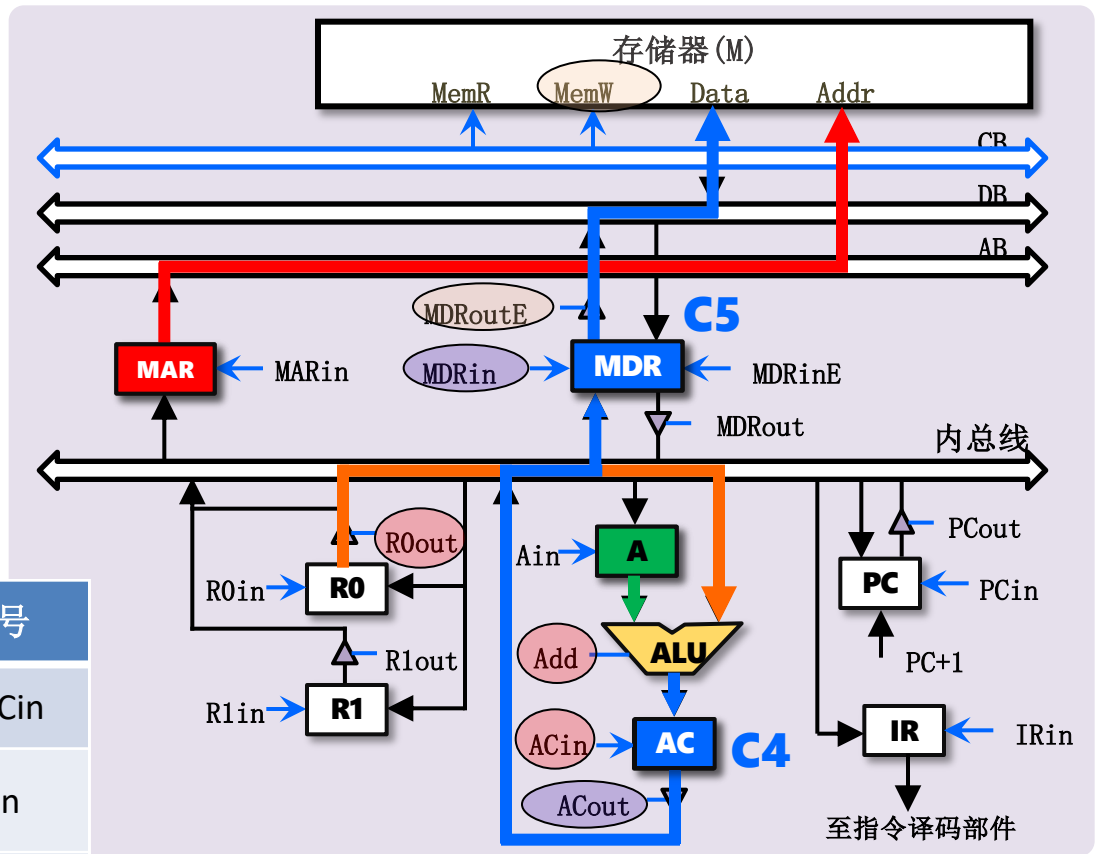
$((R1)) \rightarrow A$



时钟	功能	有效控制信号
C1	$MAR \leftarrow (R1)$	R1out, MARin
C2	$MDR \leftarrow M(MAR)$	MemR, MDRinE
C3	$A \leftarrow (MDR)$	MDRout, Ain

数据通路举例 → 执行周期

$(R0) + ((R1)) \rightarrow (R1)$



时钟	功能	有效控制信号
C4	$AC \leftarrow ALU$	R0out, ADD, ACin
C5	$MDR \leftarrow AC$	ACout, MDRin
C6	$M(MAR) \leftarrow (MDR)$	MDRoutE, MemW

数据通路小结

- 数据通路是数据传送的路径，通路建立依赖控制信号序列
 - 具体包括：数据流、控制流
- 不同指令功能，不同寻址方式，不同主机结构
 - 数据通路不同
 - 执行周期不同，定长/变长指令周期
- 取指执行的过程就是不断建立数据通路的过程

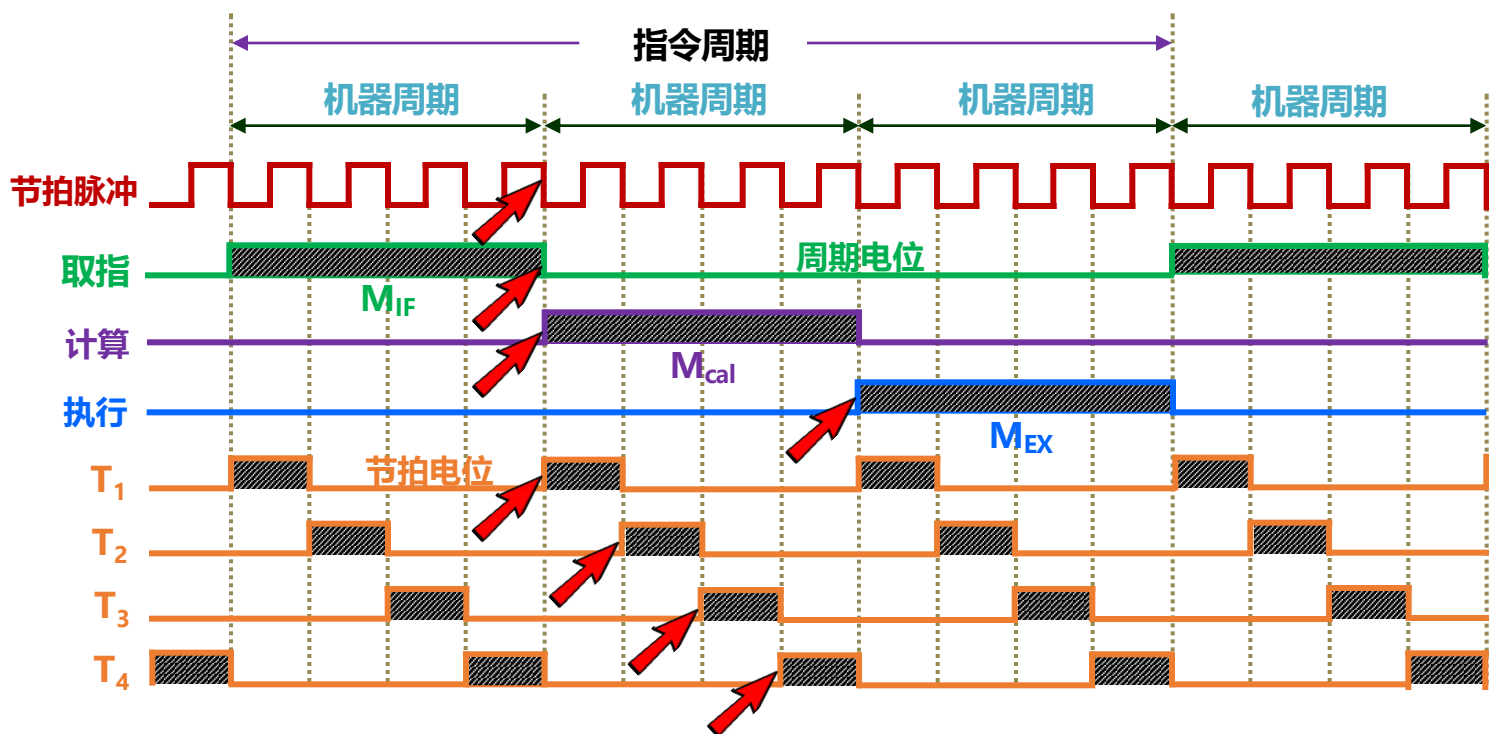
那么：给定指令集架构，如何设计数据通路？

◆ 决策：共享通路 or 专用通路？

◆ 权衡：实现难易，硬件成本，并发性，效率差异

定长指令周期的三级时序发生器

NA.1



构建时序发生器? 输入: 节拍脉冲 输出: M_{IF} , M_{cal} , M_{EX} , $T_1 \sim T_4$

设计处理器的步骤

1. **选定指令系统**：将指令的功能用硬件描述语言表示出来
2. **设计功能部件和数据通路**：根据指令功能设计功能部件，并考虑如何互连
3. **设计控制点**：设计所有数据通路所需的控制信号
4. **设计控制信号序列**：生成机器指令与控制信号的关系表
5. **构建状态转换表**：确定机器周期及节拍状态转换关系
6. **实现控制器**
 - **微程序控制器**：设计微指令，构建微程序的操作控制字段；设计地址转移逻辑，即微程序后继地址字段
 - **组合逻辑控制器**：写出每个控制信号所对应的逻辑表达式，依据化简后的结果设计组合逻辑电路

应用实例：MIPS CPU控制器设计

两种设计思想：

① 定长指令周期：单周期实现

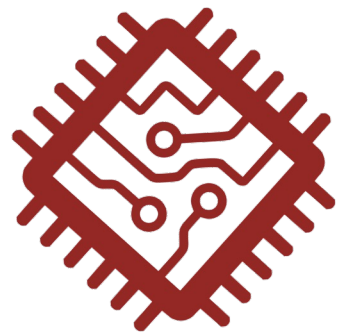
◆所有指令均在一个时钟周期内完成， $CPI=1$

◆性能取决于最慢的指令，时钟周期过长

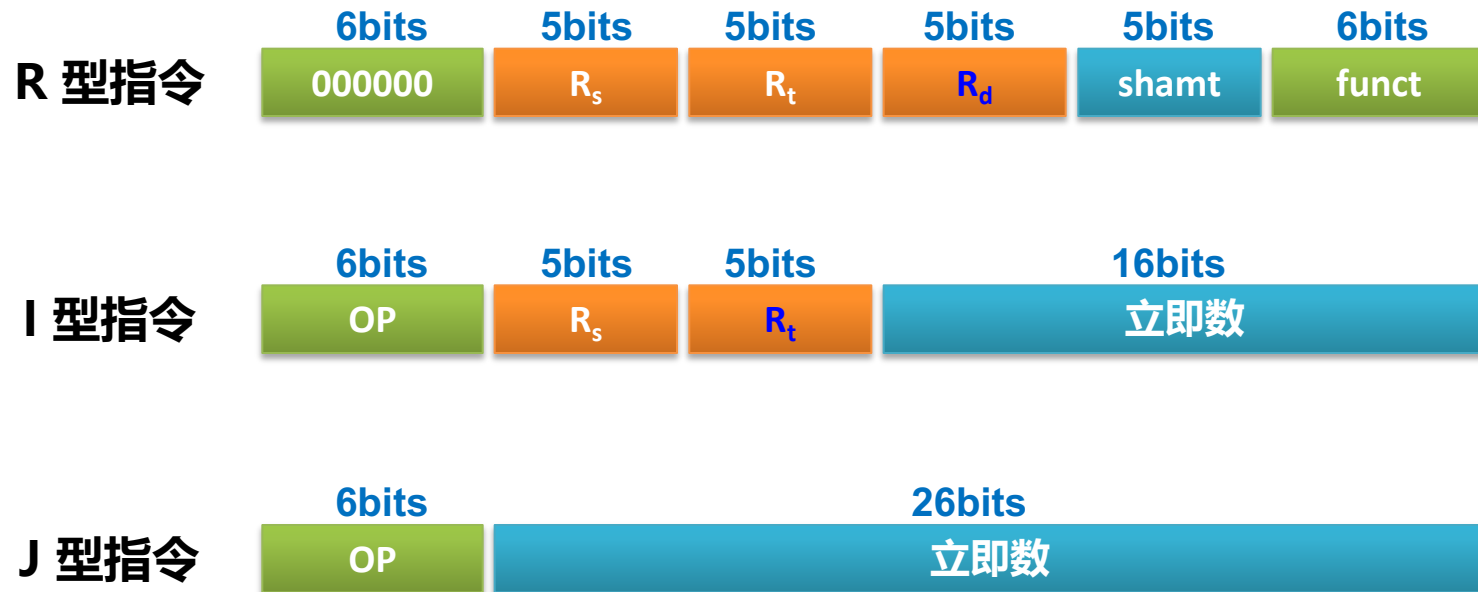
② 变长指令周期：多周期实现

■缩短时钟周期，复用器件或数据通路

■可支持流水操作，提升性能



MIPS指令格式



R型指令格式



■ add $\$s1, \$s2, \$s3$



■ sub $\$s0, \$s1, \$s2$



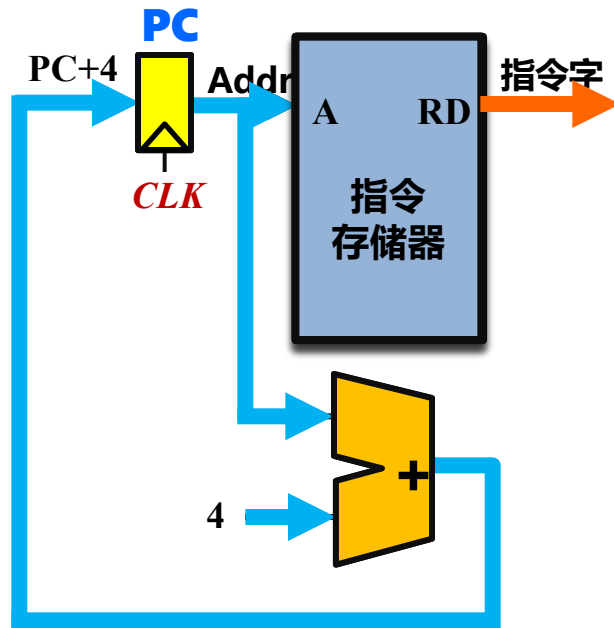
■ sll $\$s0, \$s1, 2$



I型指令格式



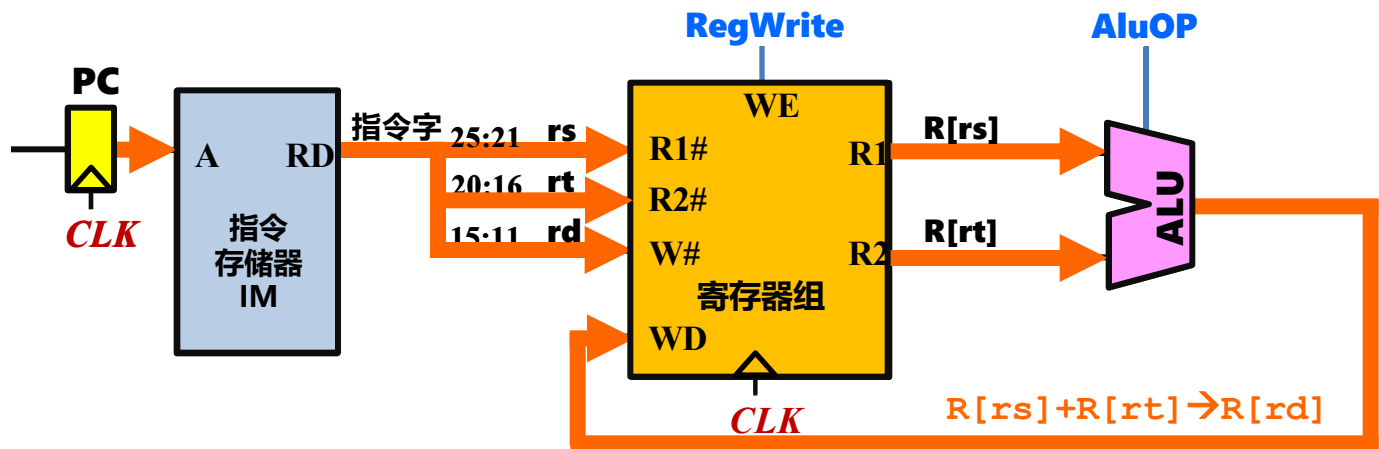
取指令数据通路



$$M[PC++] \rightarrow IR$$

- 单周期不能设置AR,DR,IR寄存器
- 程序和数据分开存放——哈佛结构
 - ◆ 指令存储器 数据存储器
 - ◆ 指令cache 数据cache
- 运算器和PC累加器分离

R型指令数据通路



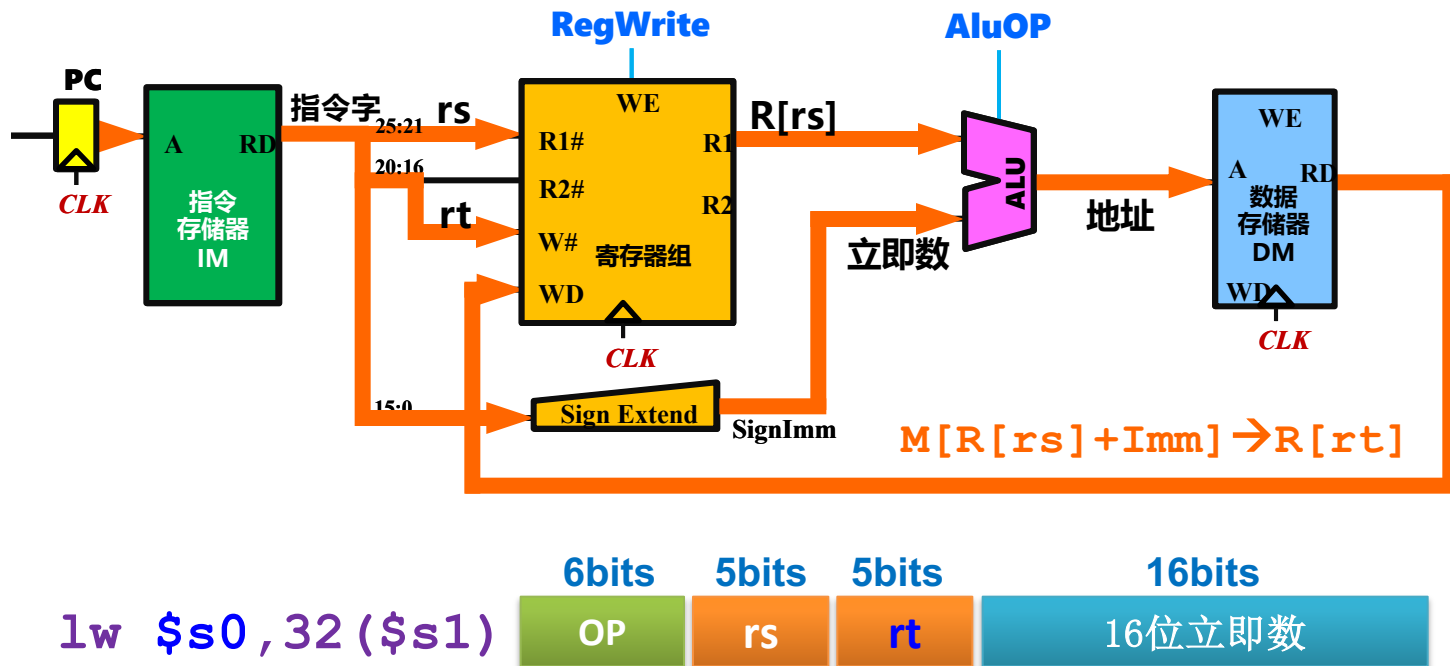
add \$s0, \$s1, \$s2



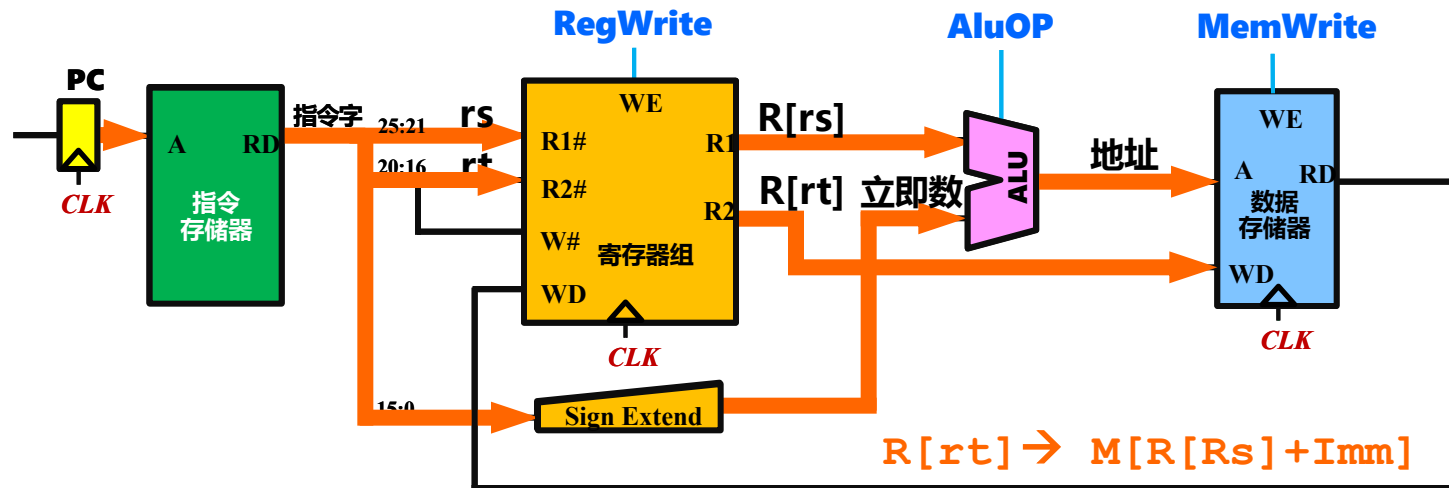
lw指令数据通路

LW $rt, offset(base)$

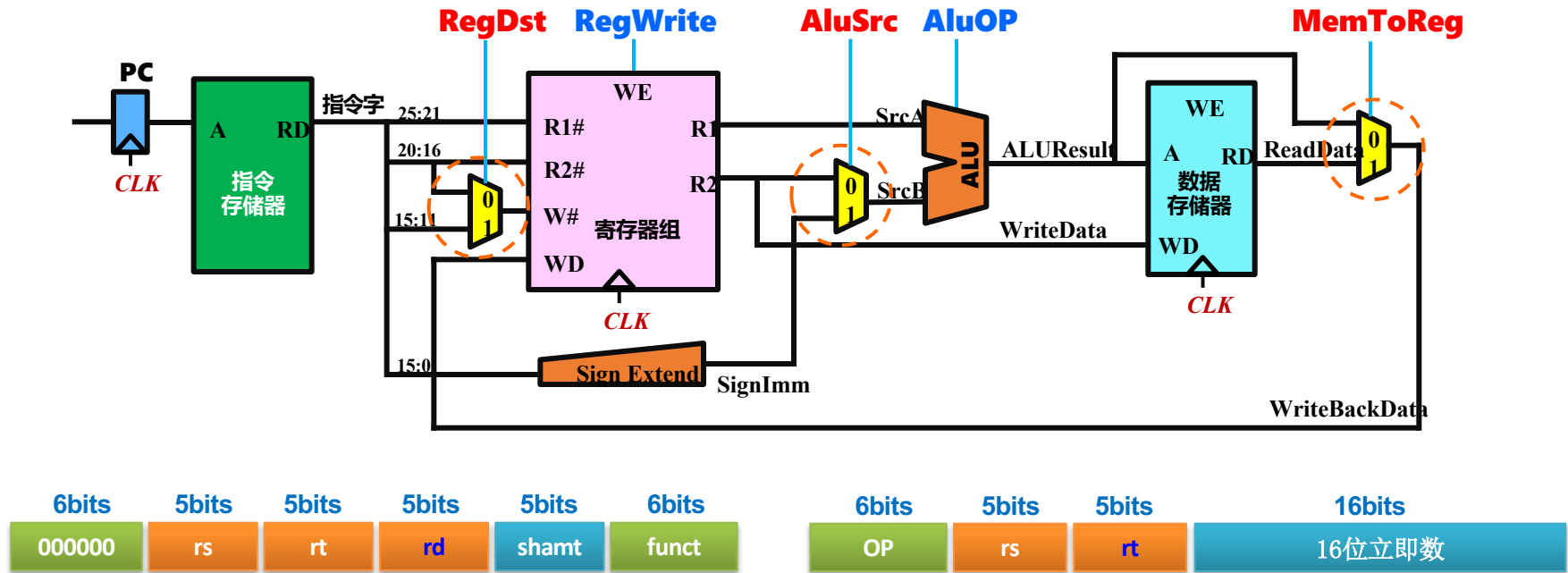
功能: $rt \leftarrow memory[base+offset]$



sw指令数据通路

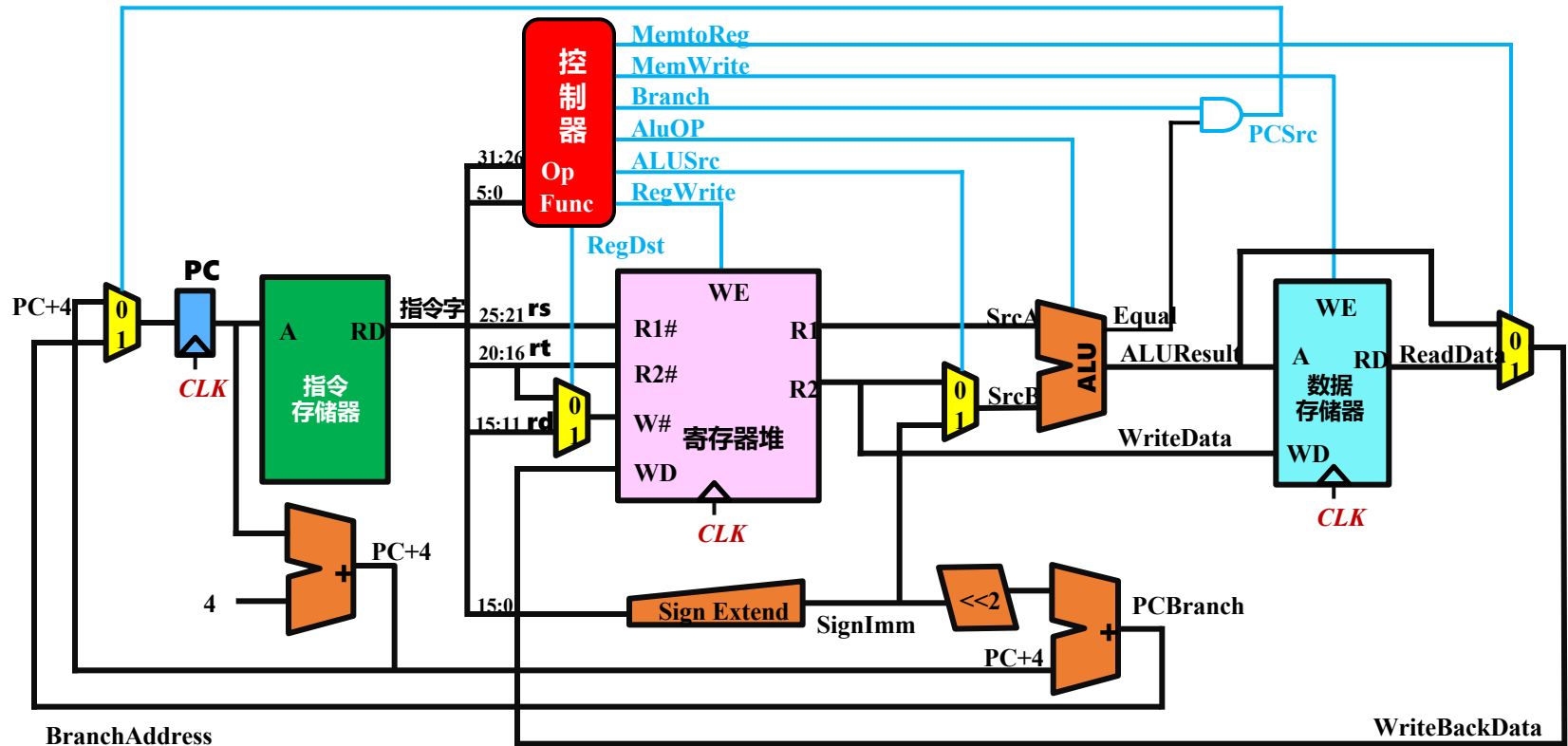


数据通路综合

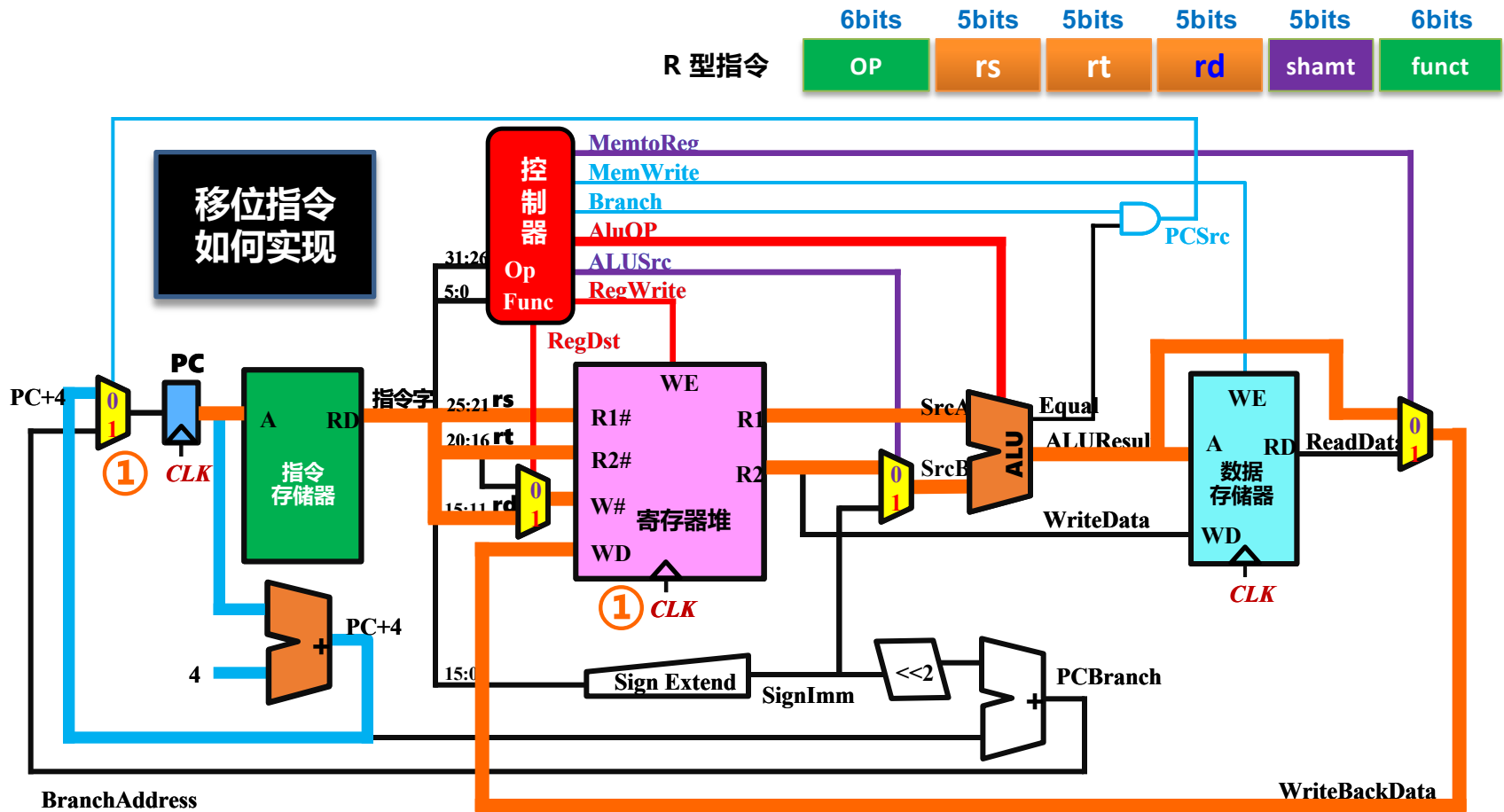


凡是遇到存在多个输入来源的情况，增加MUX，并引入**控点**

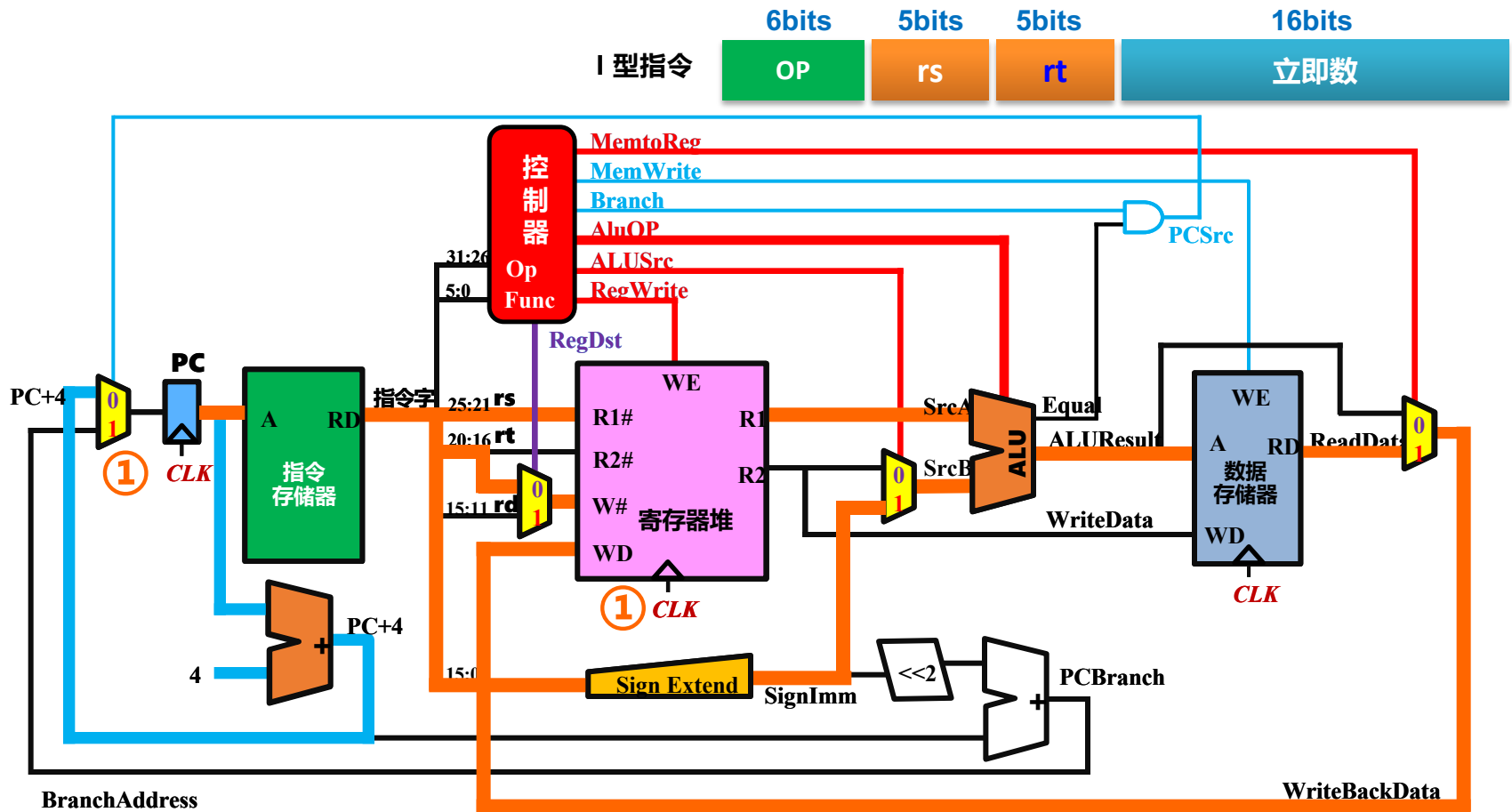
单周期MIPS数据通路



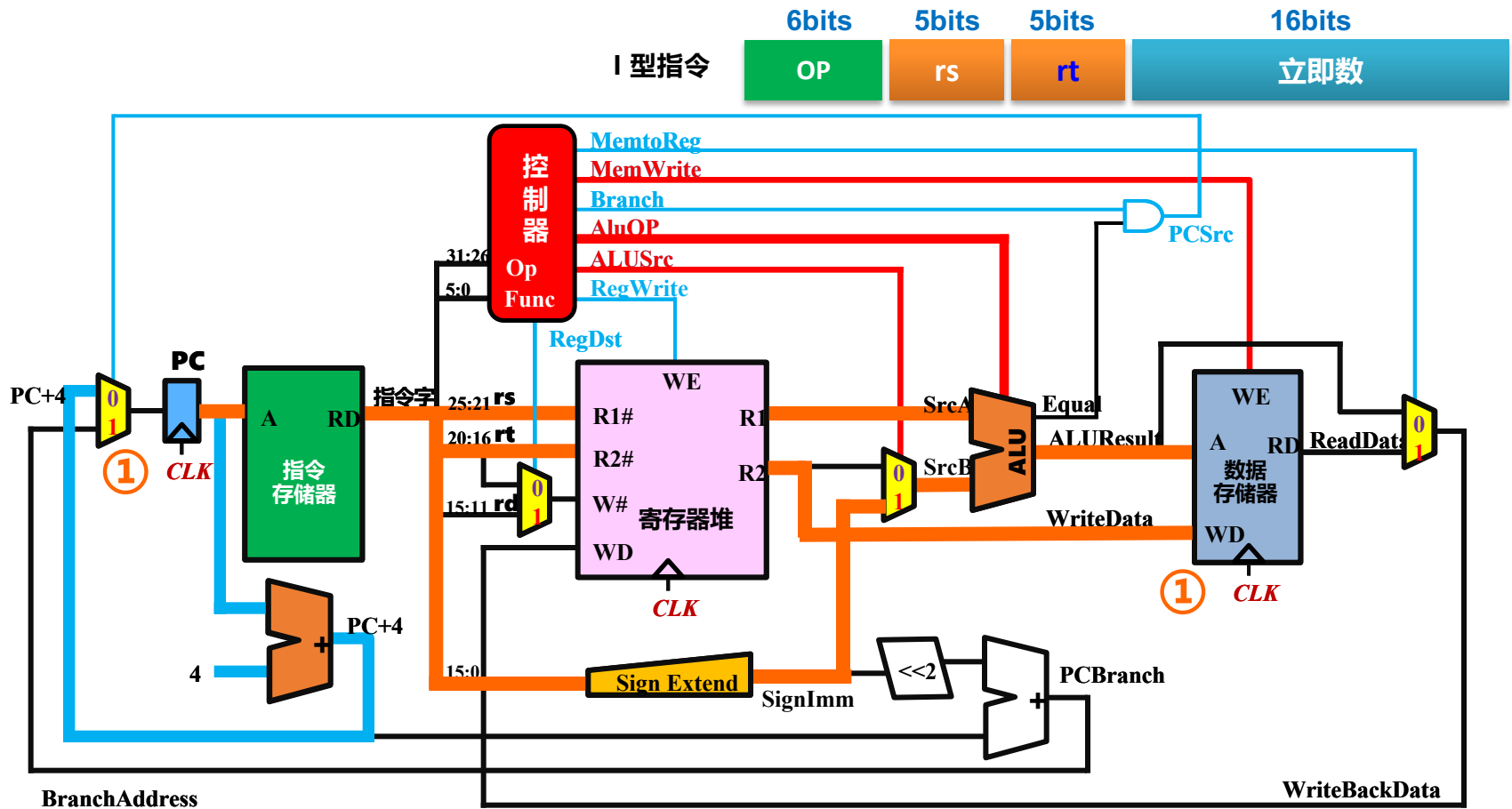
R型指令数据通路建立过程



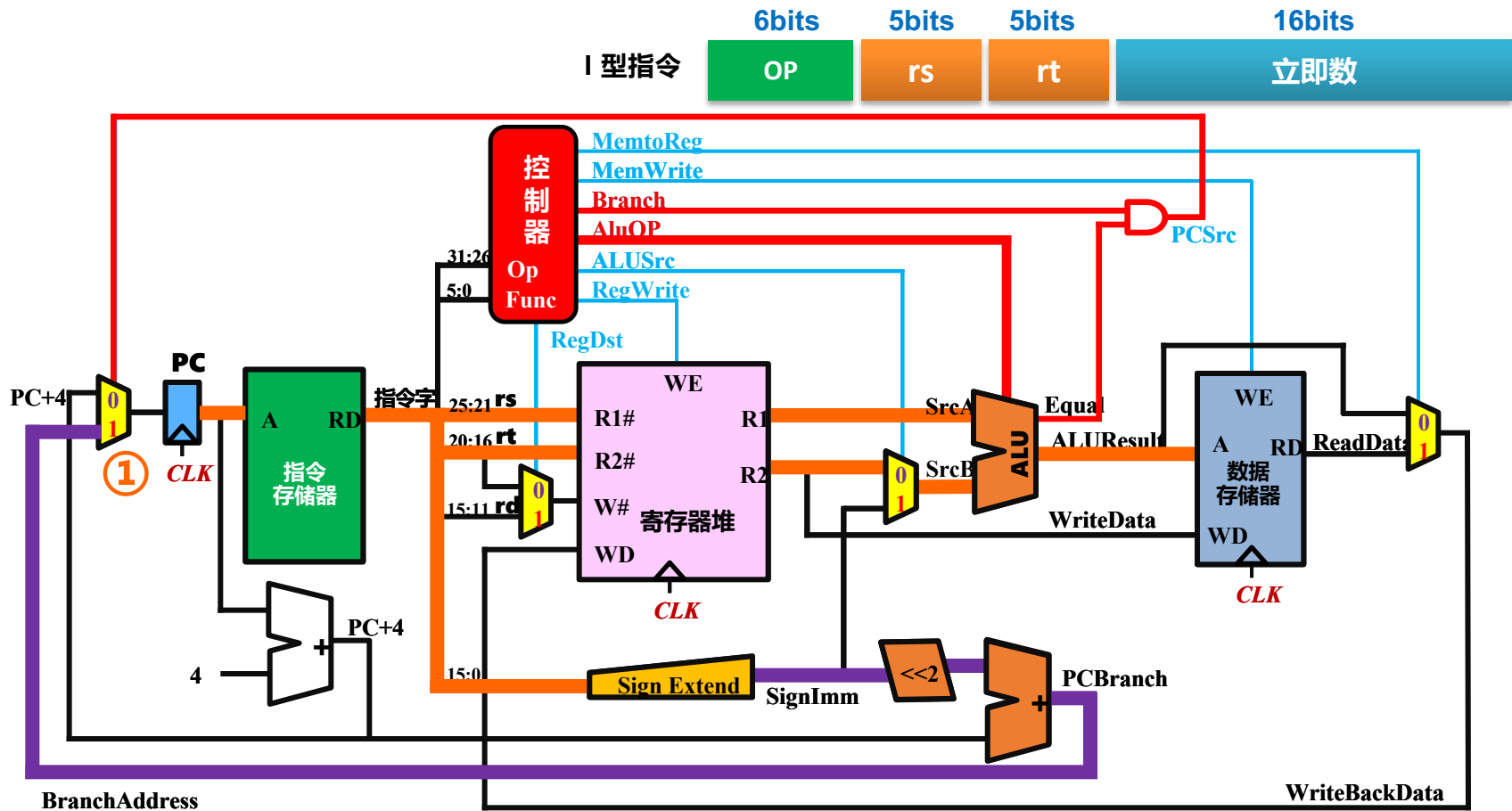
lw指令数据通路建立过程



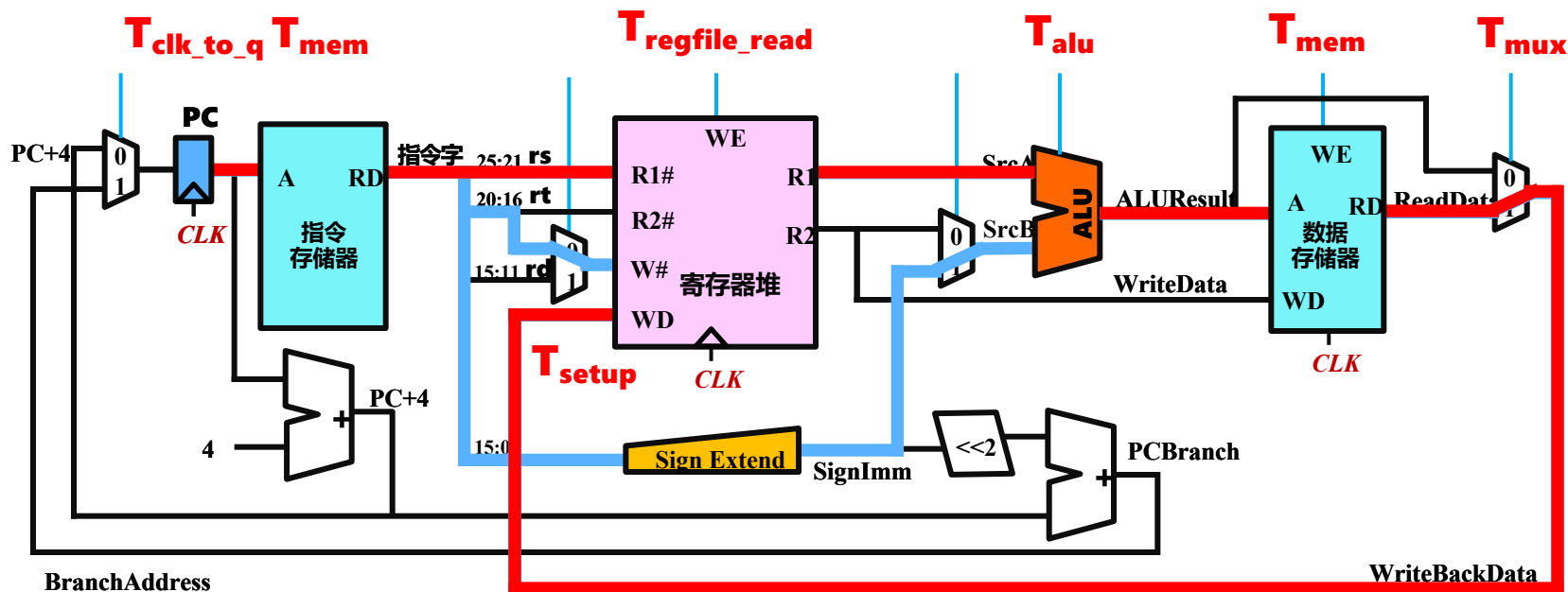
sw指令数据通路建立过程



beq指令数据通路建立过程



单周期MIPS关键路径 LW指令

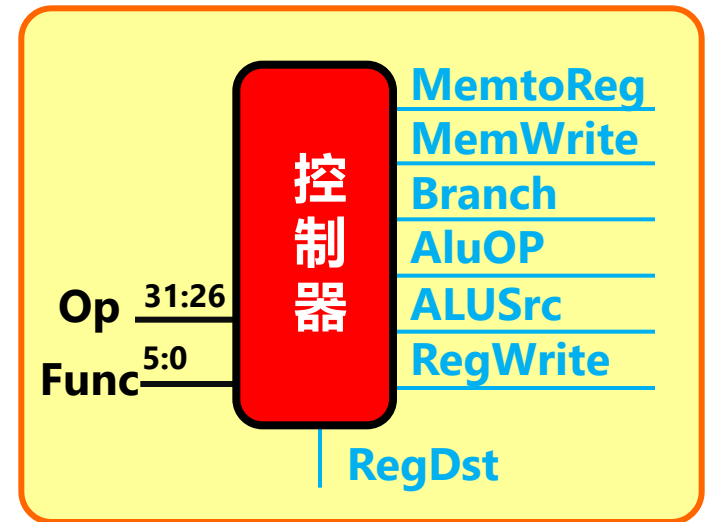


性能取决于最慢的指令 → 时钟周期过长，

∴ slow

单周期MIPS控制器设计

- 控制器类型：必须是组合逻辑电路
- 输入信号
 - 指令字opcode, funct字段（12位）
- 输出信号
 - 多路选择器选择信号
 - 内存访问控制信号
 - 寄存器写使能信号
 - 运算器控制信号，指令译码信号

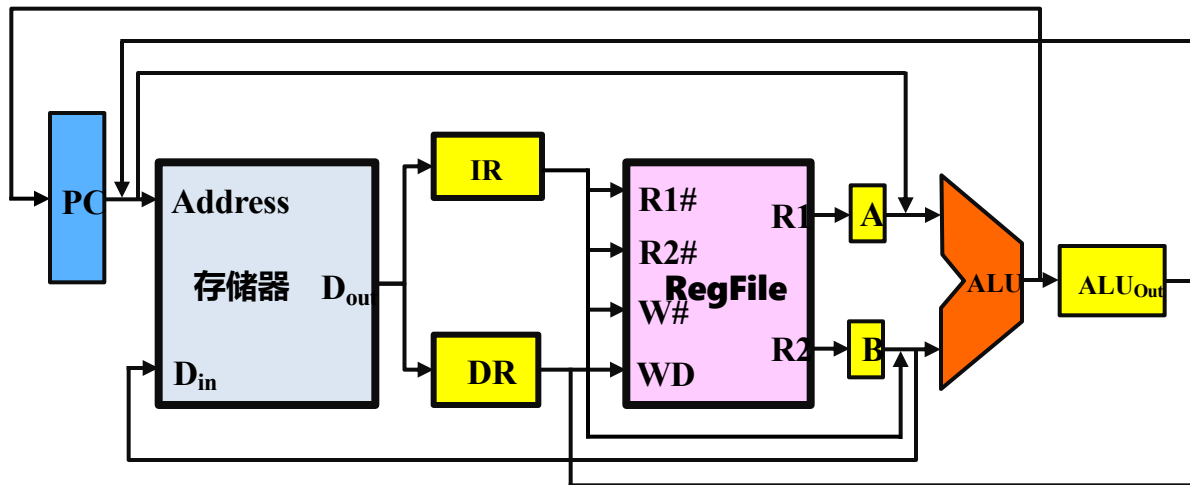


单周期数据通路小结

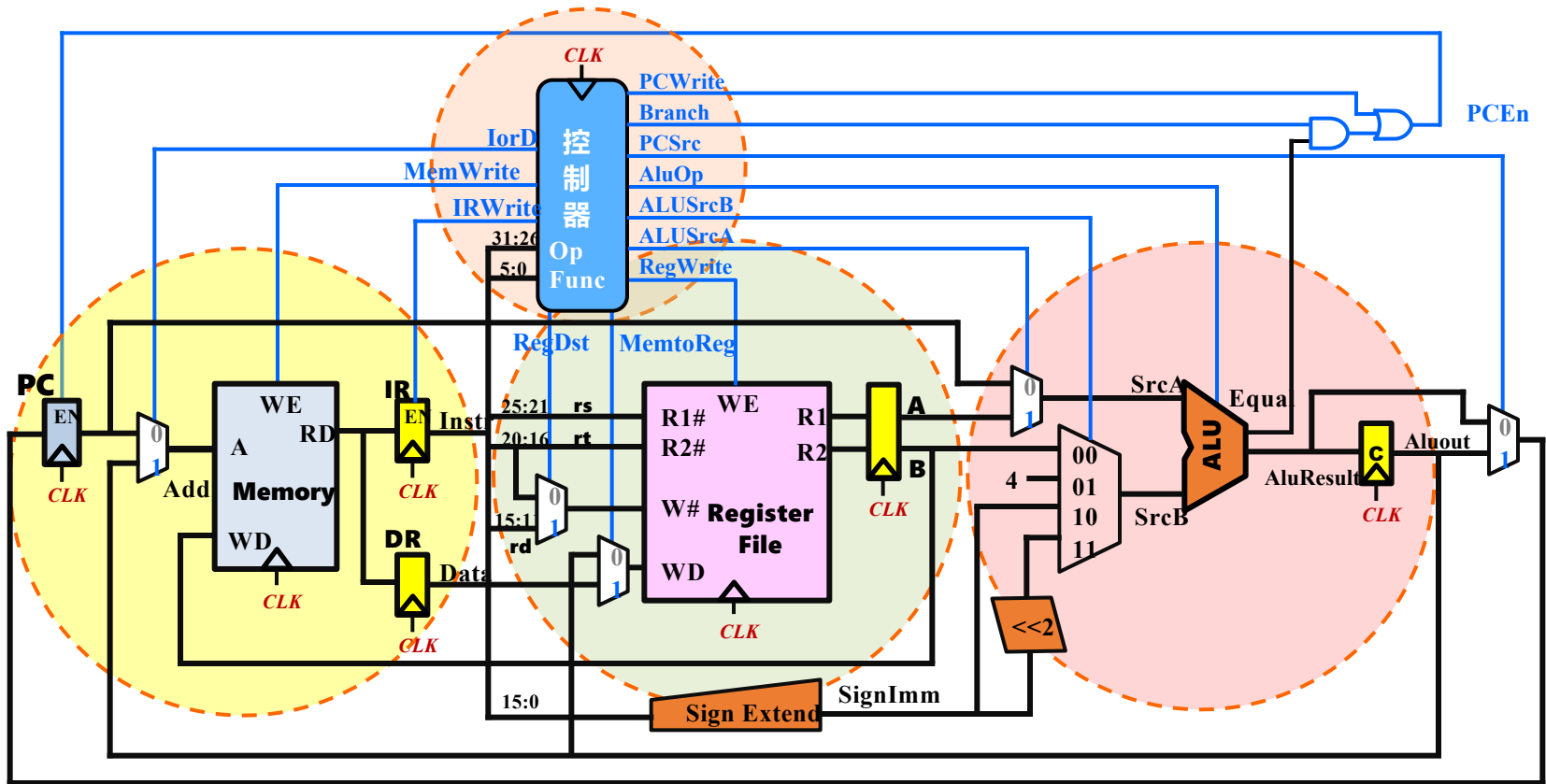
- 早期的小指令集计算机实现技术
- 单周期处理器的CPI为多少？
 - $CPI = 1$
 - 貌似CPI越小，则性能越好！
 - 实际一定如此吗？
- 单周期处理器的性能如何？
 - 计算机的性能除CPI外，还取决于时钟周期
 - 单周期处理器的时钟宽度为最复杂指令的执行时间
 - 而很多指令其实可以在更短的时间内完成
 - 因此：现代计算机采用多周期/流水方式

多周期MIPS数据通路特点

- 不再区分指令/数据存储器，分时使用部分功能部件
- 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



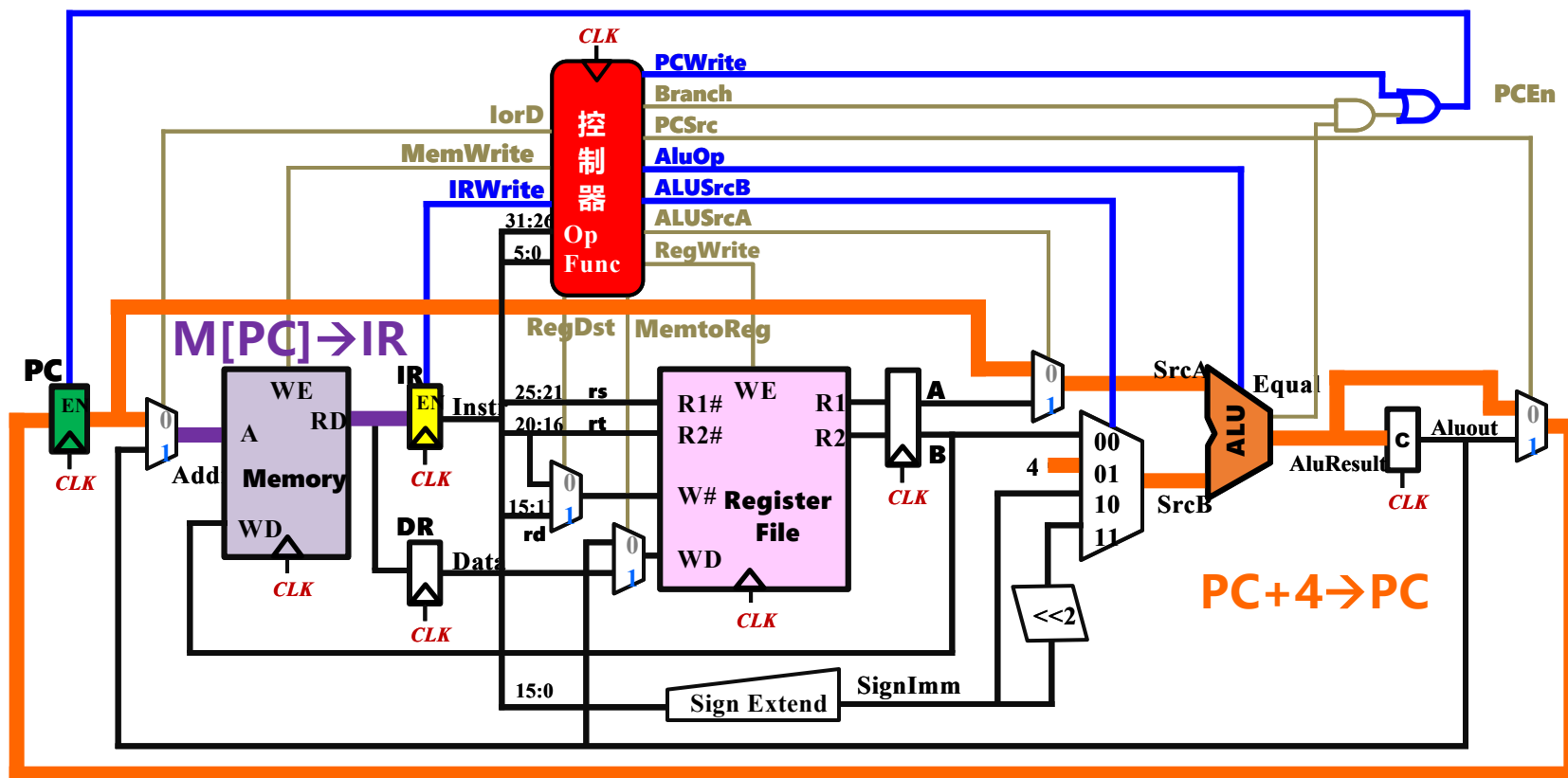
多周期MIPS CPU数据通路



- 插入锁存器，将指令执行流程拆分成多个节拍
- 以便充分发挥功能部件之间的并行性

多周期MIPS取指令阶段T1

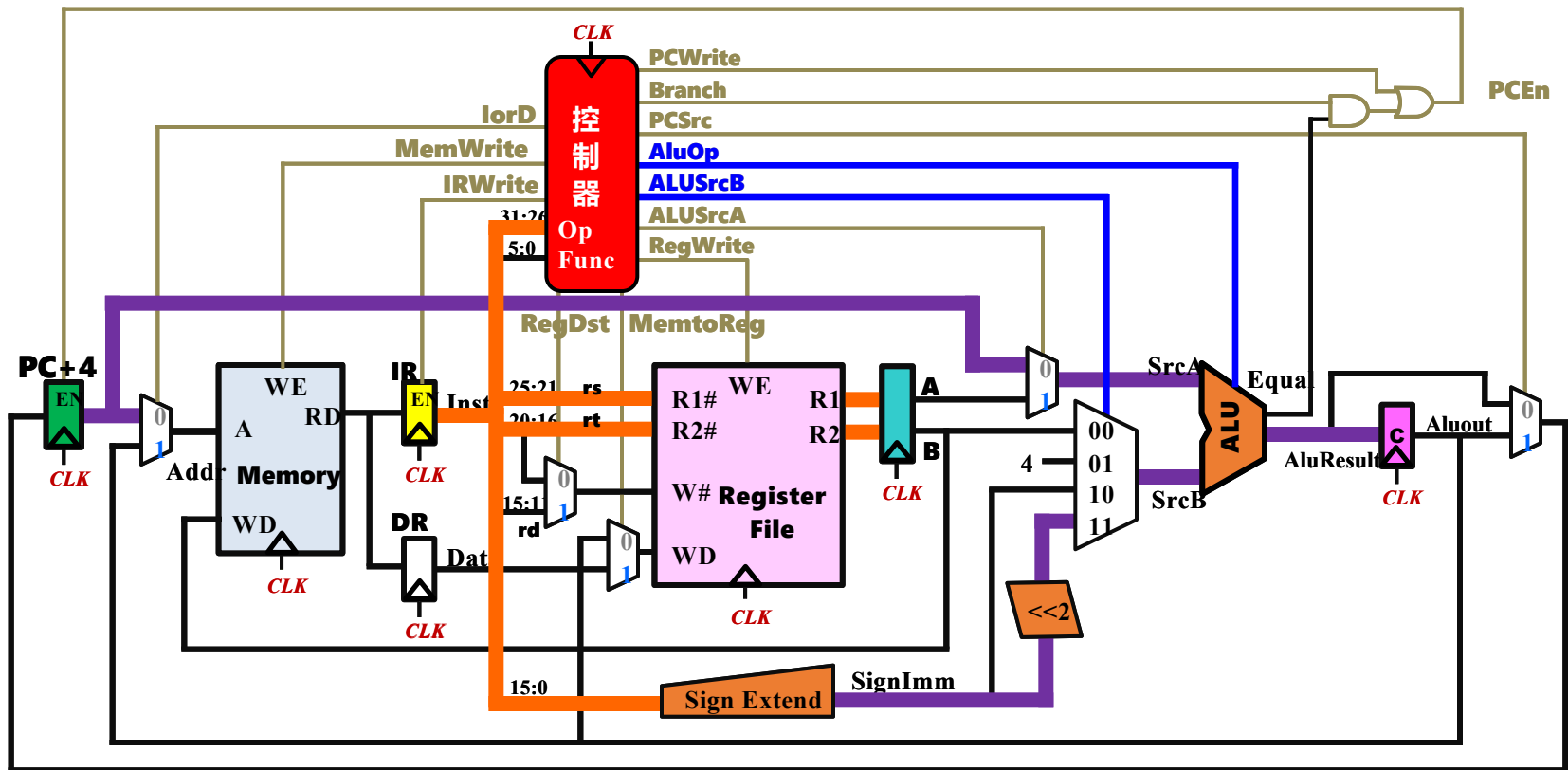
$M[PC] \rightarrow IR$ $PC+4 \rightarrow PC$



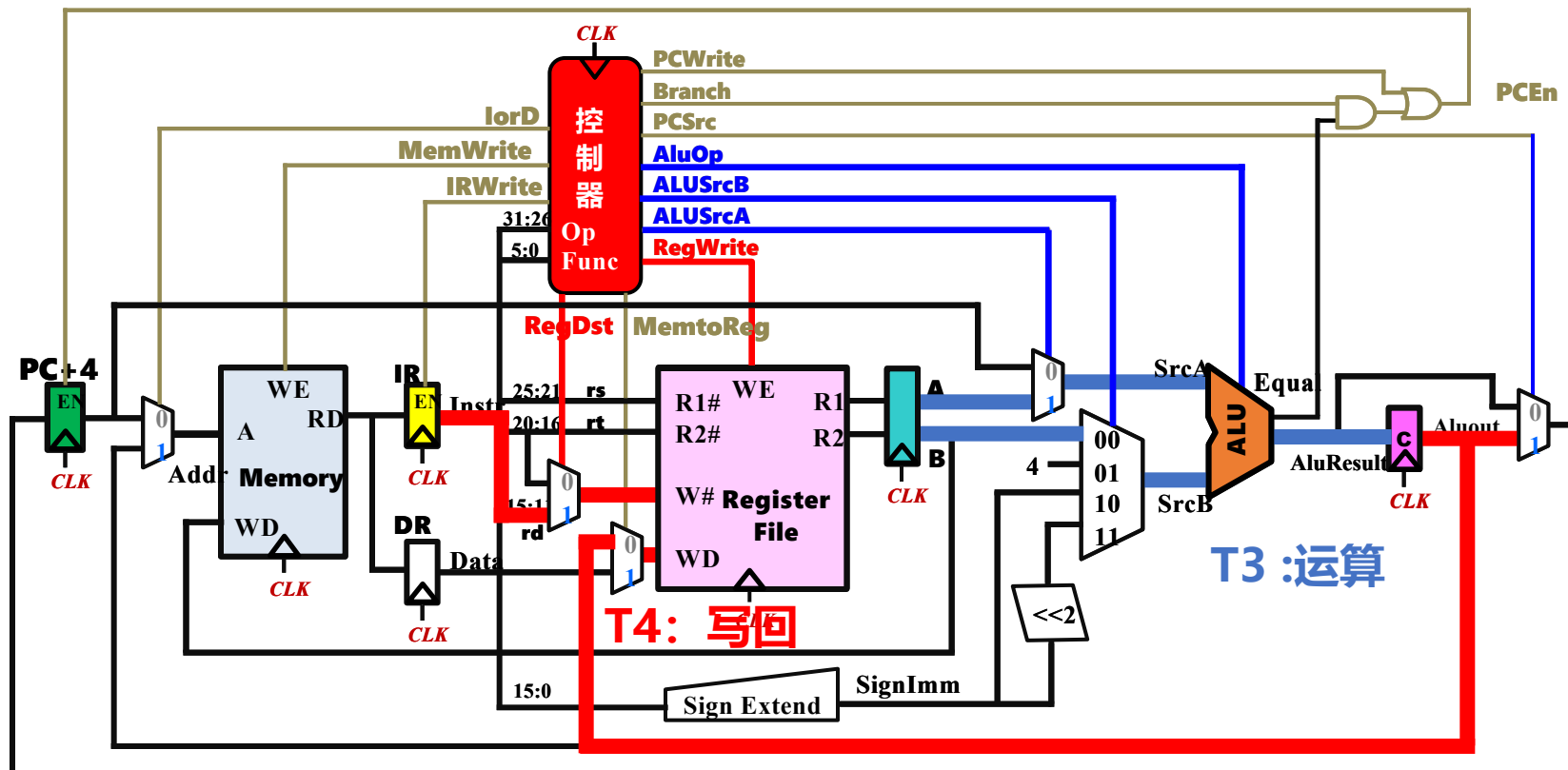
多周期MIPS取指令阶段T2

译码、Reg→A、B、

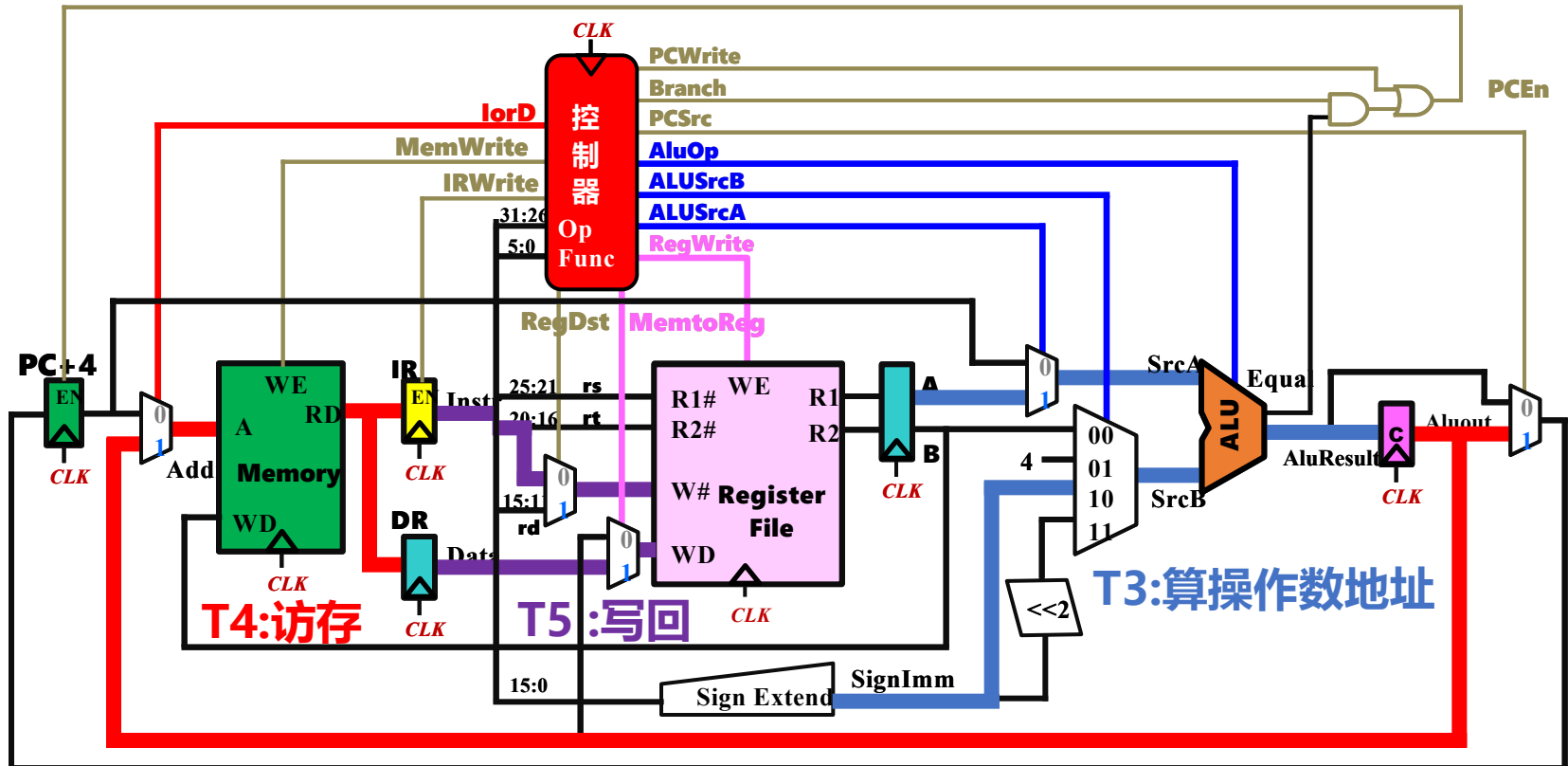
$PC+4+Imm16 \ll 2 \rightarrow C$



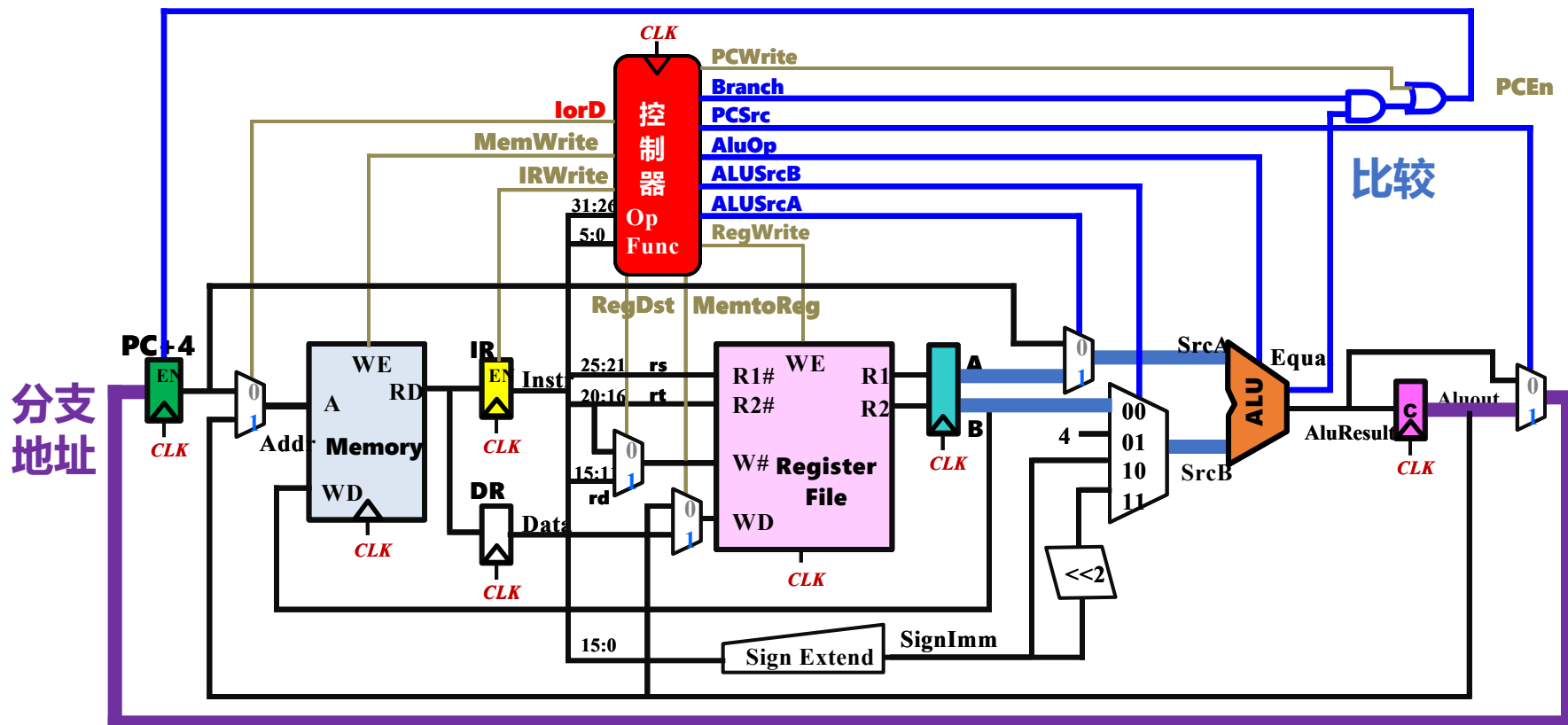
R型指令执行状态周期T3~T4



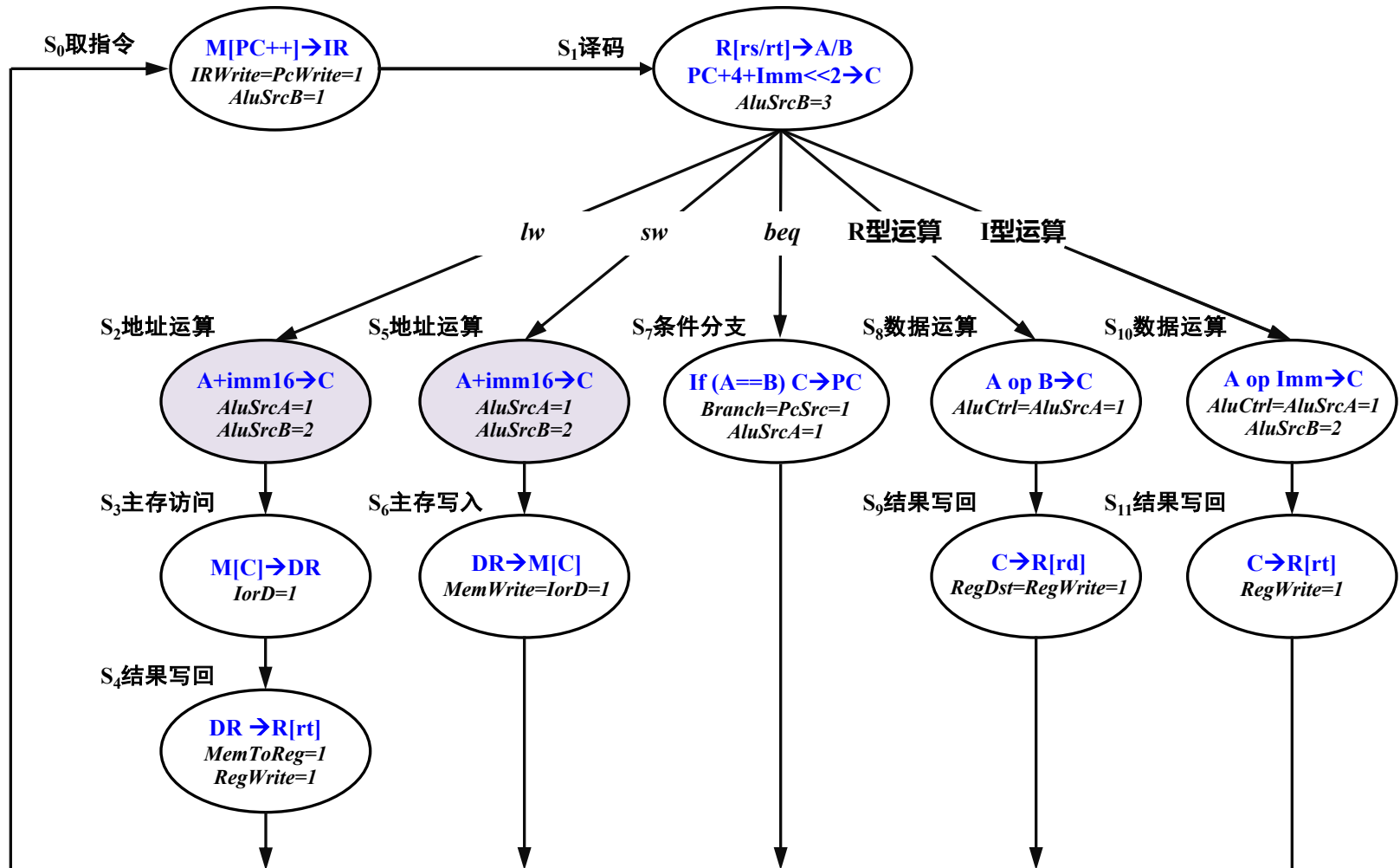
lw指令执行状态周期T3~T5



beq指令执行状态周期T3

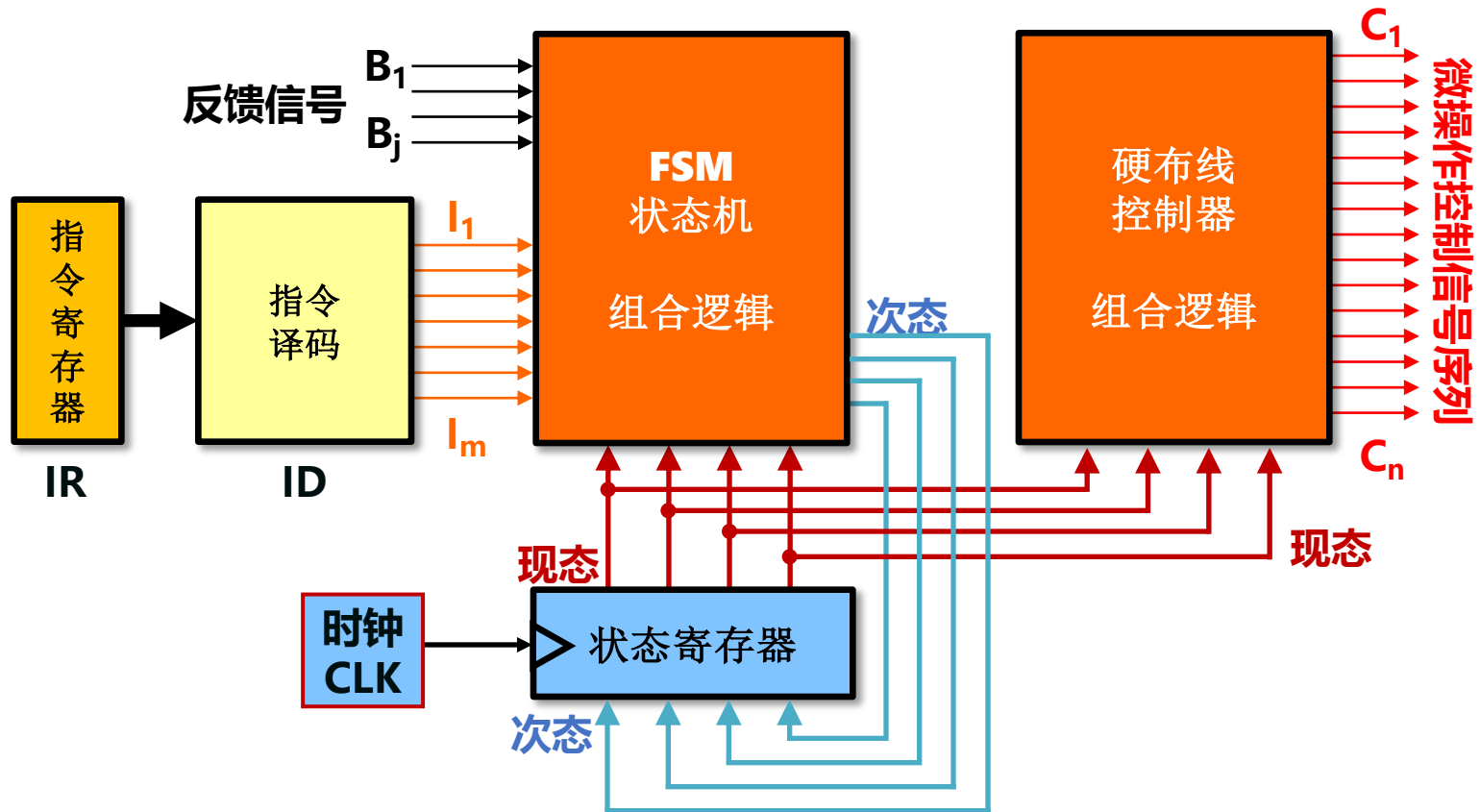


多周期MIPS状态转换图



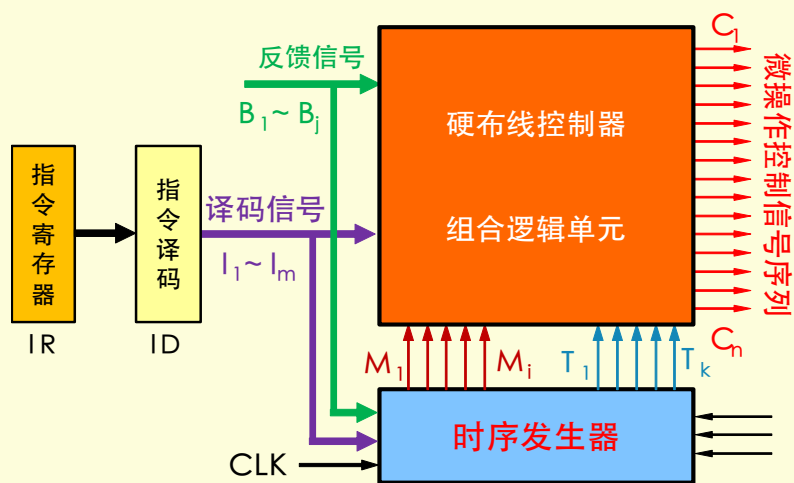
设计多周期控制器

设计思想：机器指令字 \rightarrow 控制器信号序列



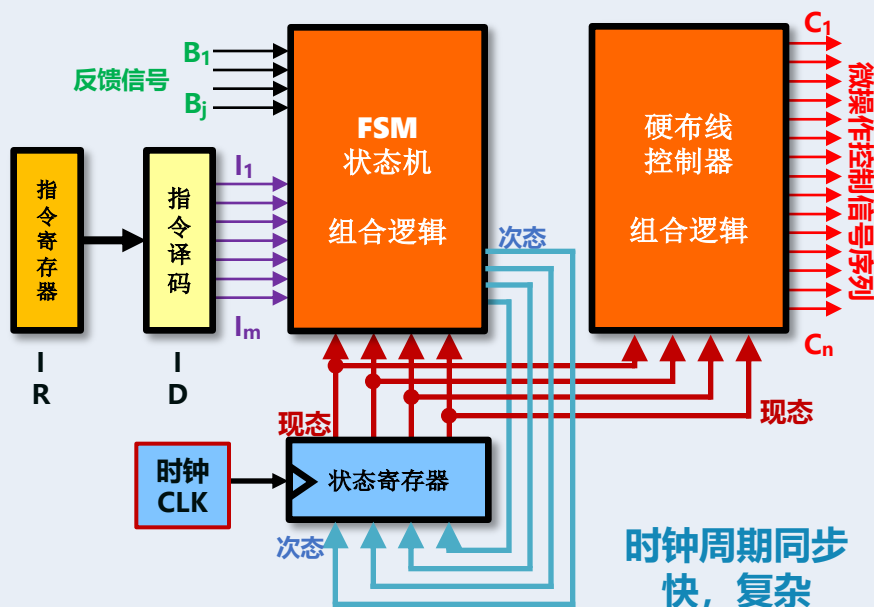
传统三级时序与现代时序对比

传统三级时序



机器周期同步，慢，简单

现代时序



时钟周期同步
快，复杂

5段流水线控制信号与传递

