



计算机组成原理

第 17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

续接 第 8 章 CPU 的结构和功能

8.1 CPU 的结构

8.2 指令周期

8.3 指令流水

举例：MIPS 五段流水线

8.4 中断系统

8.3 指令流水

一、如何提高机器速度

1. 提高访存速度

高速芯片

Cache

多体并行

2. 提高 I/O 和主机之间的传送速度

中断

DMA

通道

I/O 处理机

多总线

3. 提高运算器速度

高速芯片

改进算法

快速进位链

• 提高整机处理能力

高速器件

改进系统结构，开发系统的并行性

二、系统的并行性

8.3

1. 并行的概念

并行 { **并发** 两个或两个以上事件在 **同一时间段** 发生
同时 两个或两个以上事件在 **同一时刻** 发生

时间上互相重叠

2. 并行性的等级

过程级（程序、进程）	粗粒度	软件实现
指令级（指令之间） （指令内部）	细粒度	硬件实现

三、指令流水原理

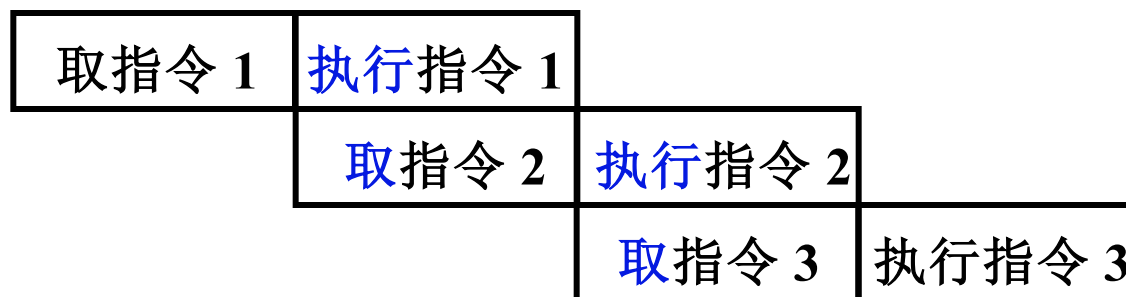
8.3

1. 指令的串行执行



取指令 取指令部件 完成 总有一个部件 空闲
执行指令 执行指令部件 完成

2. 指令的二级流水



若 取指 和 执行 阶段时间上 完全重叠
指令周期 减半 速度提高 1 倍

3. 影响指令流水效率加倍的因素

8.3

(1) 执行时间 > 取指时间



(2) 条件转移指令 对指令流水的影响

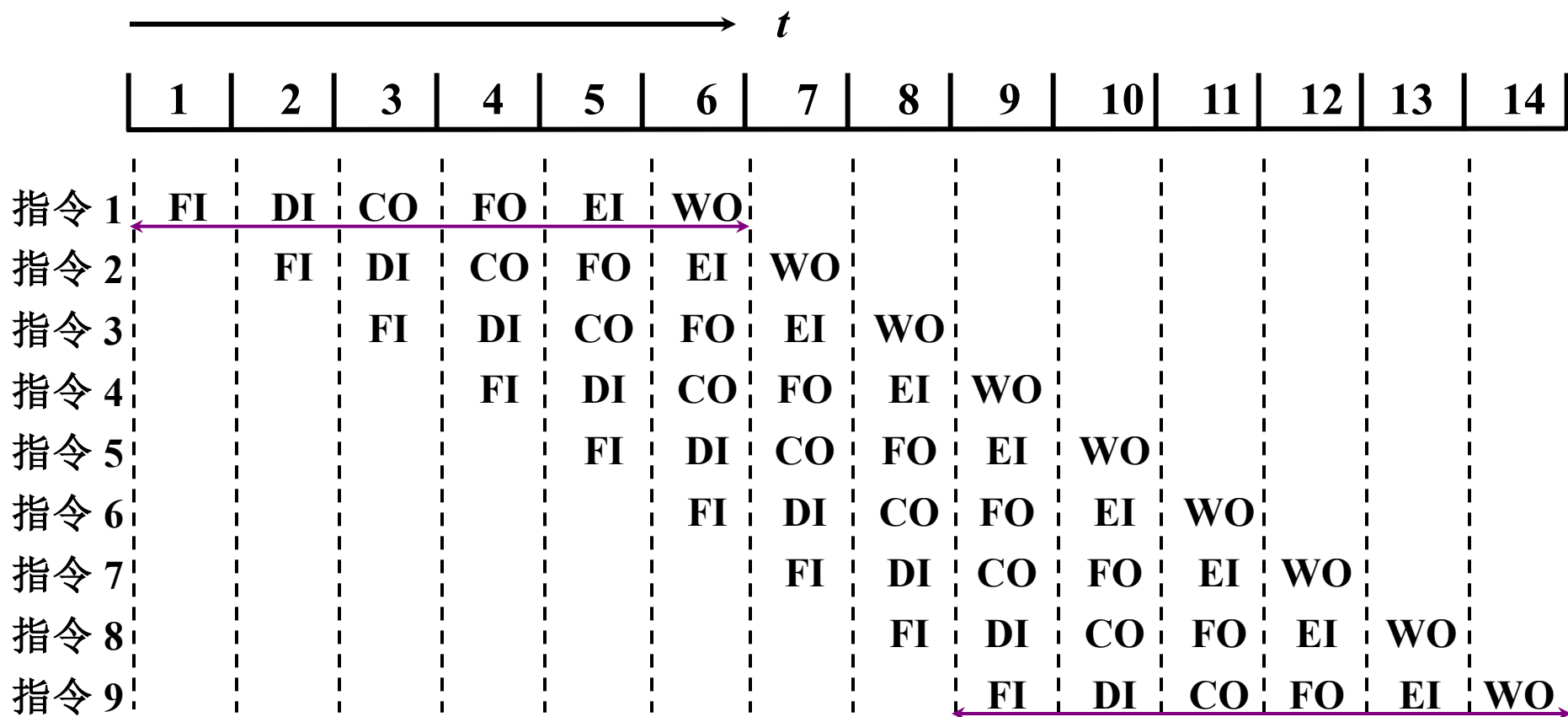
必须等 上条 指令执行结束，才能确定 下条 指令的地址，
造成时间损失

猜测法

解决办法 ？

4. 指令的六级流水

8.3



完成 一条指令

6 个时间单位

串行执行

$6 \times 9 = 54$ 个时间单位

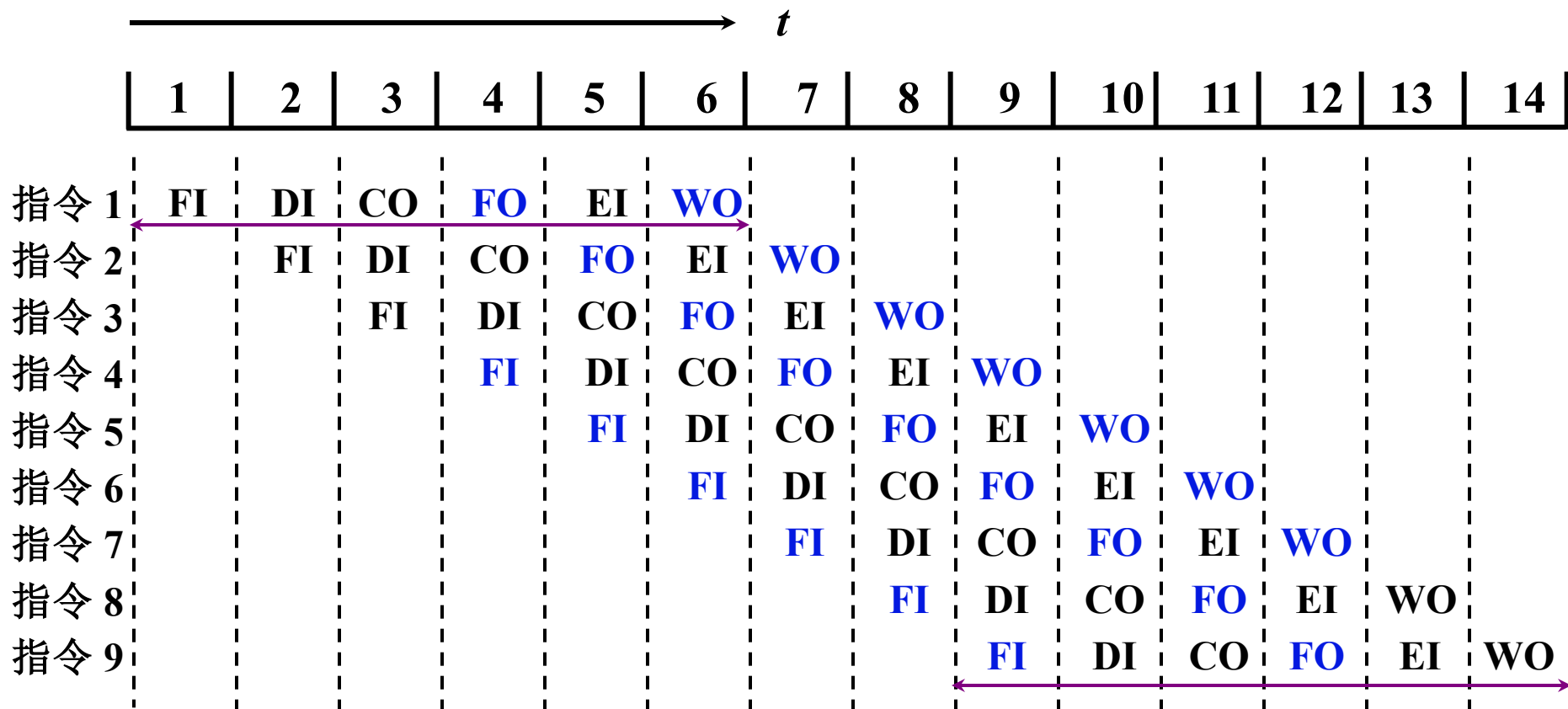
六级流水

14 个时间单位

三、影响指令流水线性能的因素

8.3

1. 结构相关 不同指令争用同一功能部件产生资源冲突



解决办法

- 停顿
- 指令 1 与指令 4 冲突
- 指令 2 与指令 5 冲突
- 指令 1、指令 3、指令 6 冲突
- 指令存储器和数据存储器分开
- 指令预取技术 (适用于访存周期短的情况)

2. 数据相关

8.3

不同指令因重叠操作，可能改变操作数的 读/写 访问顺序

- 写后读相关 (RAW)

SUB R_1 , R_2 , R_3 ; $(R_2) - (R_3) \rightarrow R_1$

ADD R_4 , R_5 , R_1 ; $(R_5) + (R_1) \rightarrow R_4$

- 读后写相关 (WAR)

STA M, R_2 ; $(R_2) \rightarrow M$ 存储单元

ADD R_2 , R_4 , R_5 ; $(R_4) + (R_5) \rightarrow R_2$

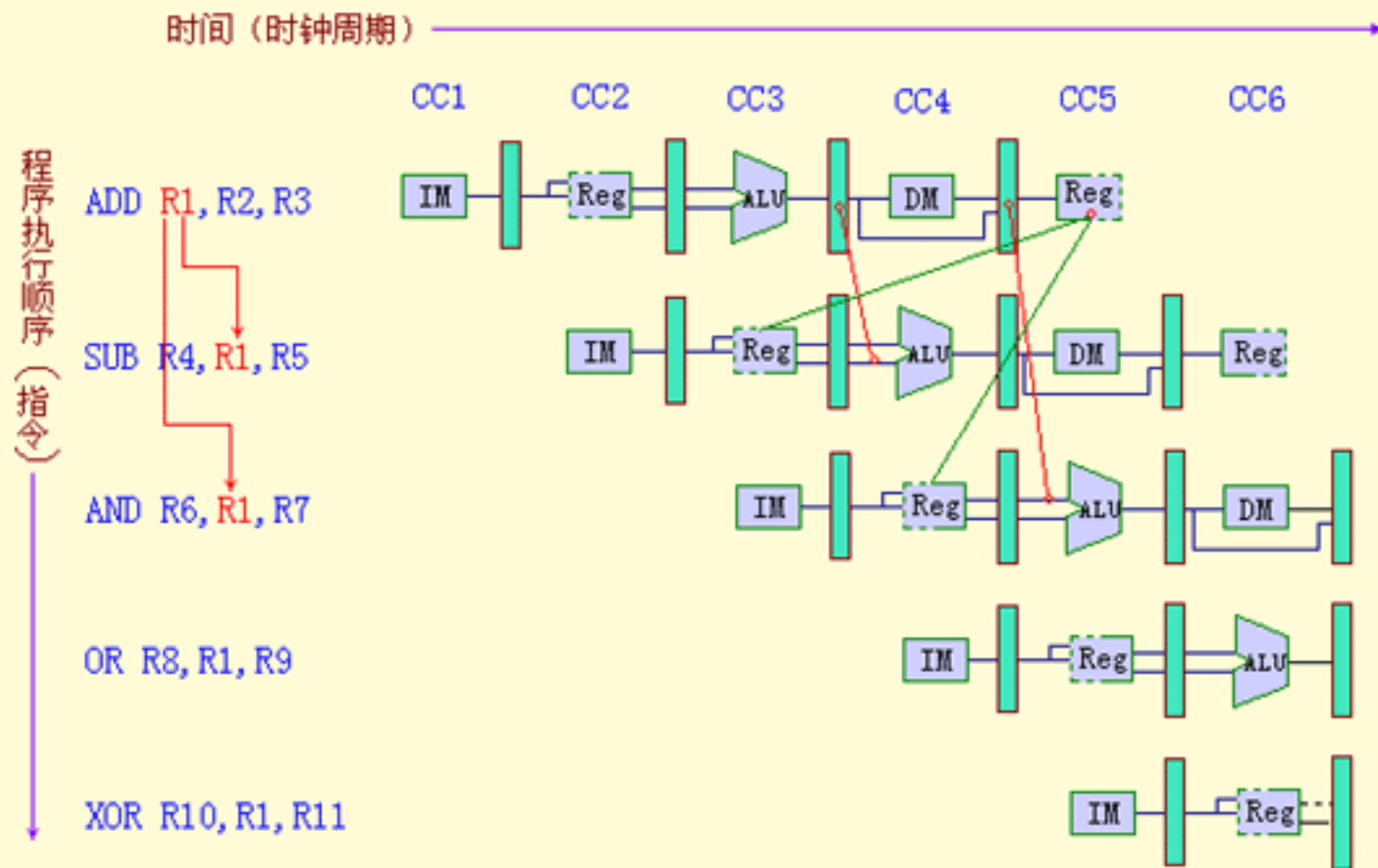
- 写后写相关 (WAW)

MUL R_3 , R_2 , R_1 ; $(R_2) \times (R_1) \rightarrow R_3$

SUB R_3 , R_4 , R_5 ; $(R_4) - (R_5) \rightarrow R_3$

2024/6/5 解决办法 • 后推法 • 采用 旁路技术 (定向技术)


采用定向技术消除数据相关



3. 控制相关

8.3

由转移指令引起



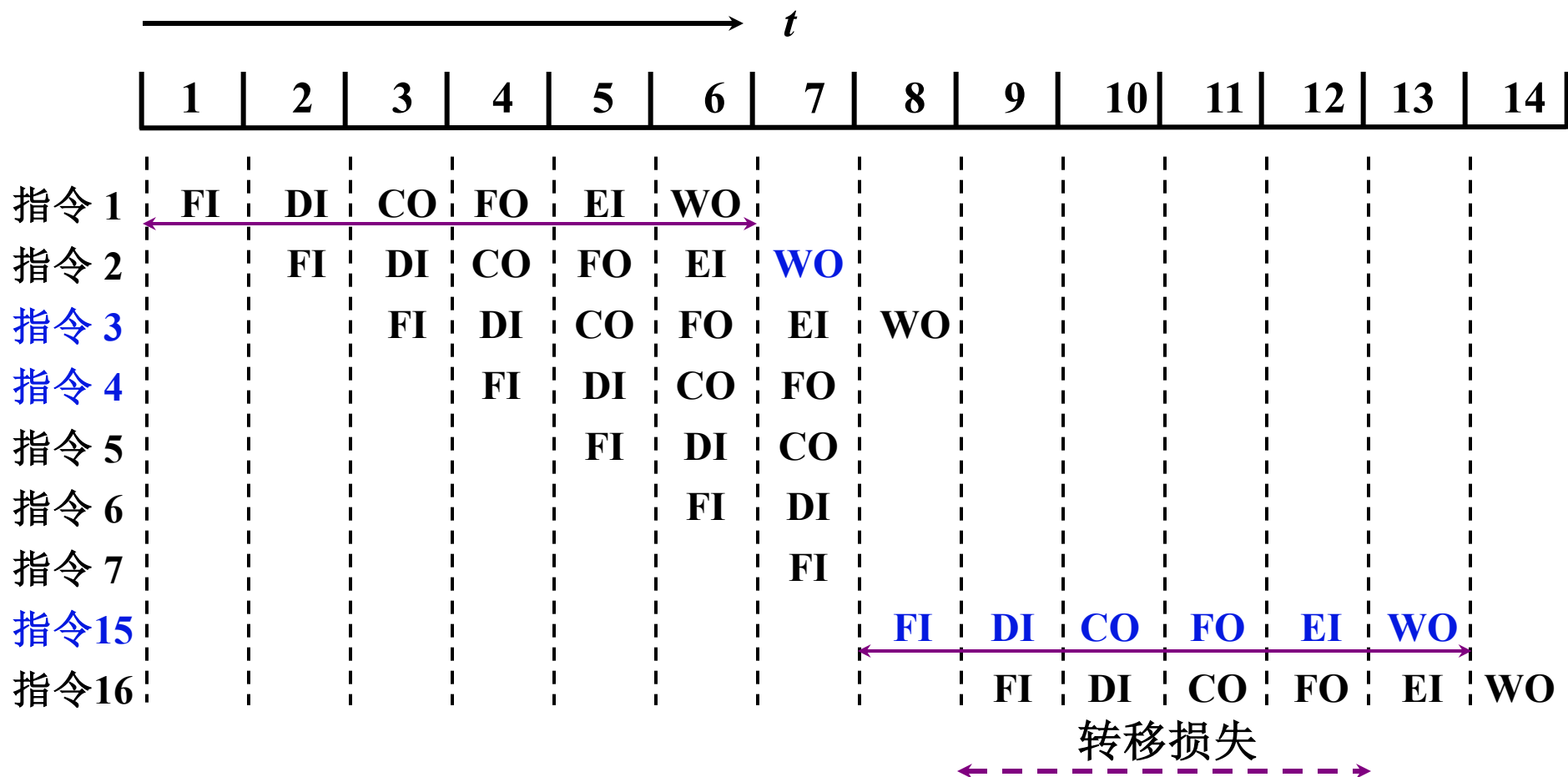
LDA # 0
LDX # 0
M ADD X, D
INX
CPX # N
BNE M
DIV # N
STA ANS

BNE 指令必须等
CPX 指令的结果
才能判断出
是转移
还是顺序执行

3. 控制相关

8.3

设 指令3 是转移指令



四、流水线性能

8.3

1. 吞吐率

单位时间内 流水线所完成指令 或 输出结果 的数量

设 m 段的流水线各段时间为 Δt

- 最大吞吐率

$$T_{pmax} = \frac{1}{\Delta t}$$

- 实际吞吐率

连续处理 n 条指令的吞吐率为

$$T_p = \frac{n}{m \cdot \Delta t + (n-1) \cdot \Delta t}$$

2. 加速比 S_p

m 段的流水线的速度与等功能的非流水线的速度之比

设流水线各段时间为 Δt

完成 n 条指令在 m 段流水线上共需

$$T = m \cdot \Delta t + (n-1) \cdot \Delta t$$

完成 n 条指令在等效的非流水线上共需

$$T' = nm \cdot \Delta t$$

则

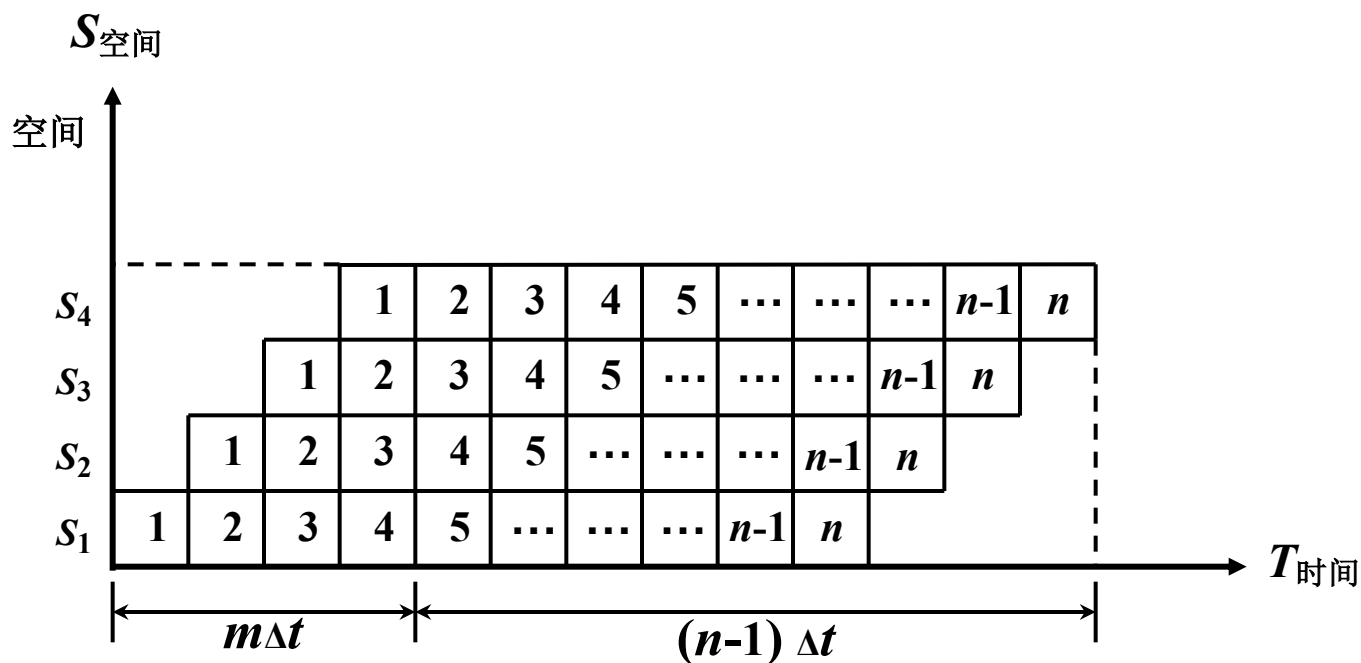
$$S_p = \frac{nm \cdot \Delta t}{m \cdot \Delta t + (n-1) \cdot \Delta t} = \frac{nm}{m + n - 1}$$

3. 效率

8.3

流水线中各功能段的 **利用率**

由于流水线有 **建立时间** 和 **排空时间**
因此各功能段的 **设备不可能** 一直 处于 **工作** 状态



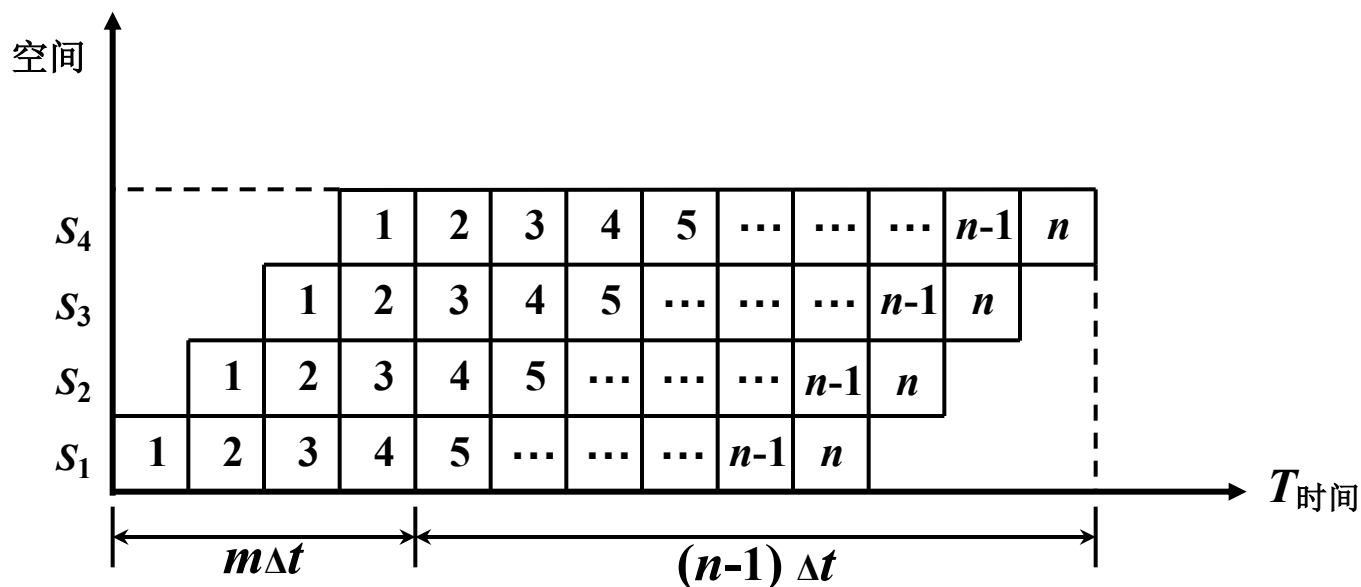
3. 效率

8.3

流水线中各功能段的 **利用率**

效率 = $\frac{\text{流水线各段处于工作时间的时空区}}{\text{流水线中各段总的时空区}}$

$$= \frac{mn\Delta t}{m(m+n-1)\Delta t}$$



五、流水线的多发技术

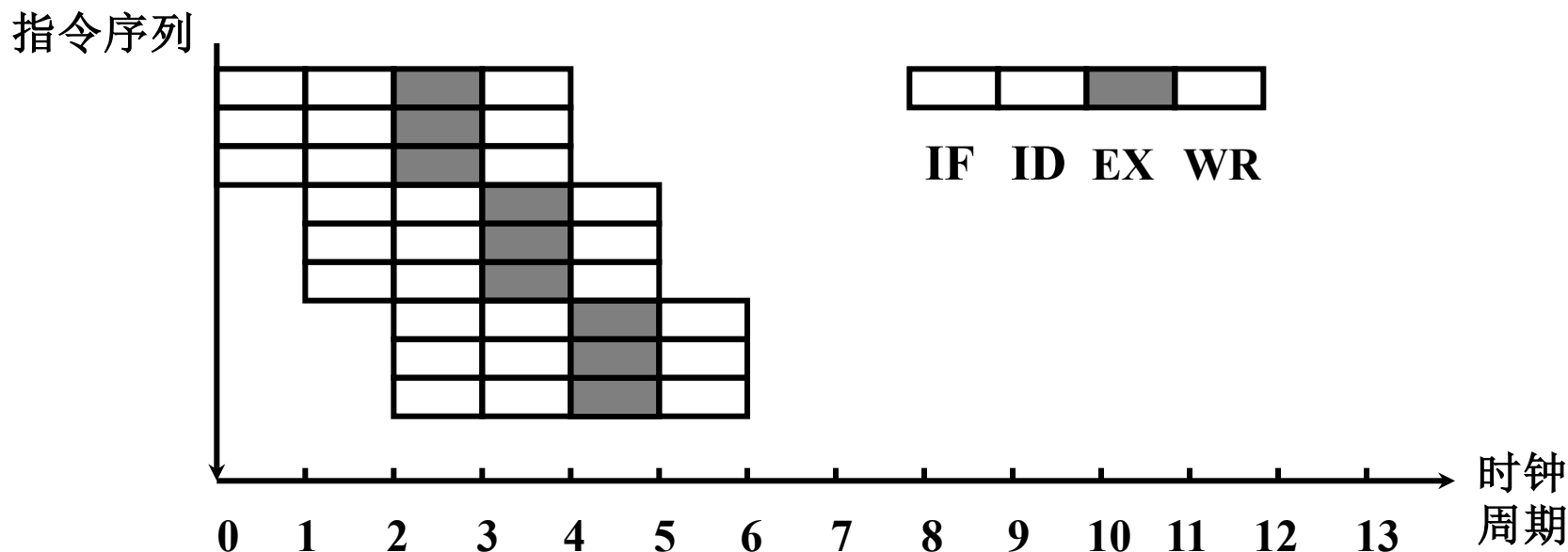
8.3

1. 超标量技术

➤ 每个时钟周期内可 **并发多条独立指令**
配置多个功能部件

➤ **不能调整** 指令的 **执行顺序**

通过编译优化技术，把可并行执行的指令搭配起来



2. 超流水线技术

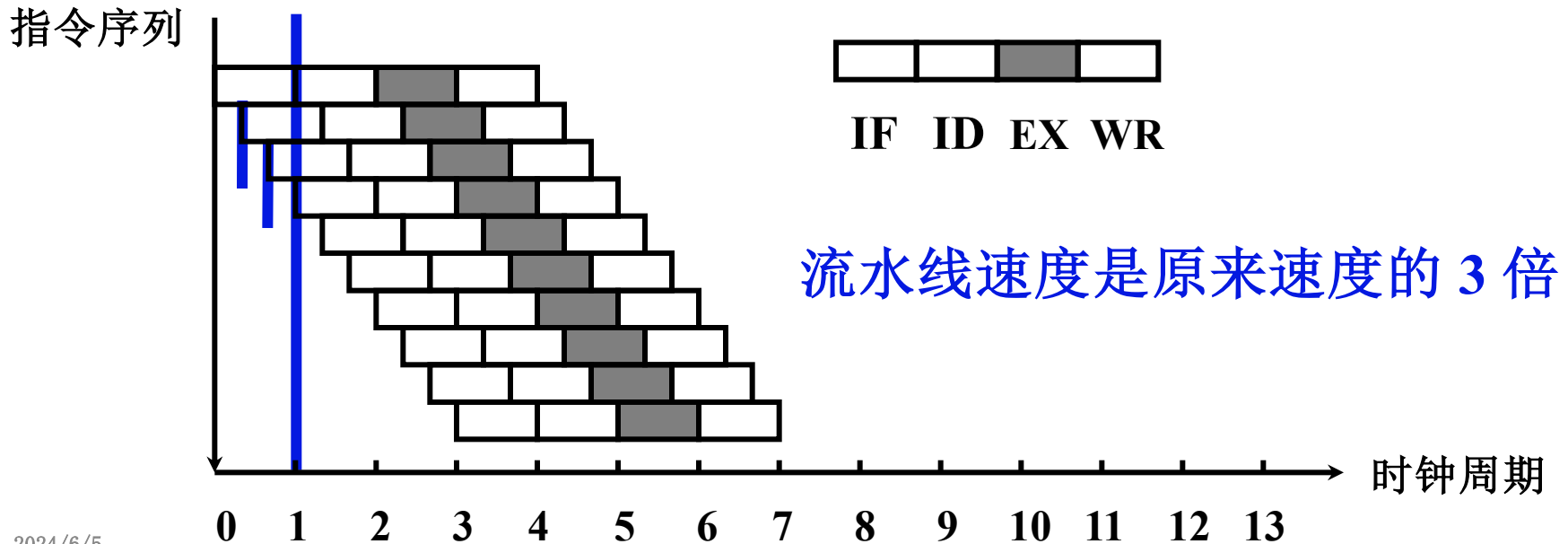
8.3

- 在一个时钟周期内再分段（3段）

在一个时钟周期内 一个功能部件使用多次（3次）

- 不能调整 指令的 执行顺序

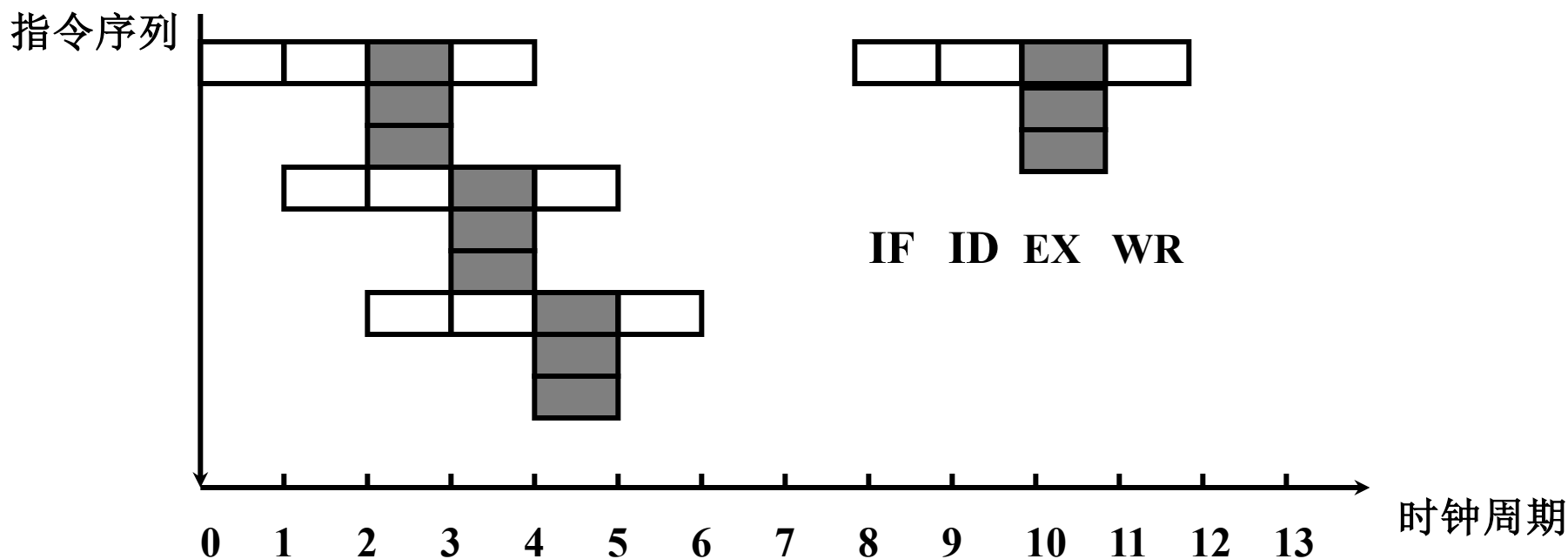
靠编译程序解决优化问题



3. 超长指令字技术

8.3

- 由编译程序 **挖掘** 出指令间 **潜在** 的 **并行性**，
将 **多条** 能 **并行操作** 的指令组合成 **一条**
具有 **多个操作码字段** 的 **超长指令字**（可达几百位）
- 采用 **多个处理部件**

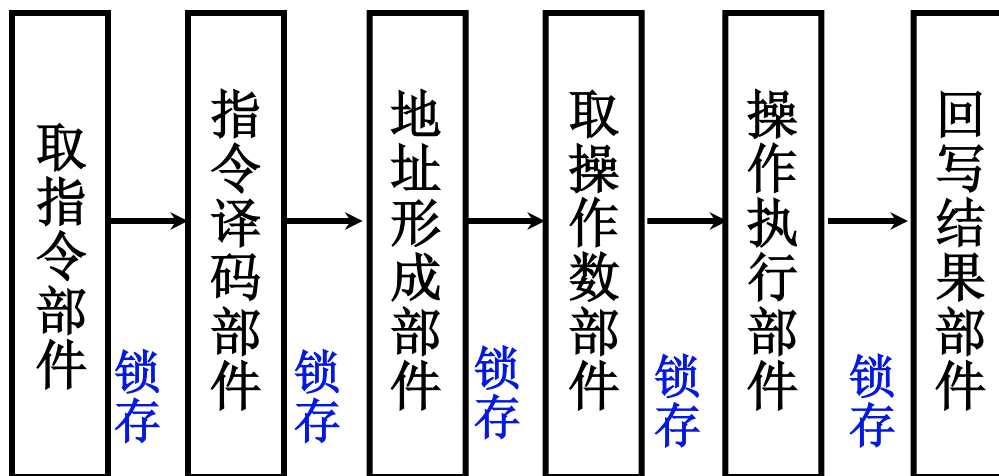


六、流水线结构

8.3

1. 指令流水线结构

完成一条指令分 6 段，每段需一个时钟周期



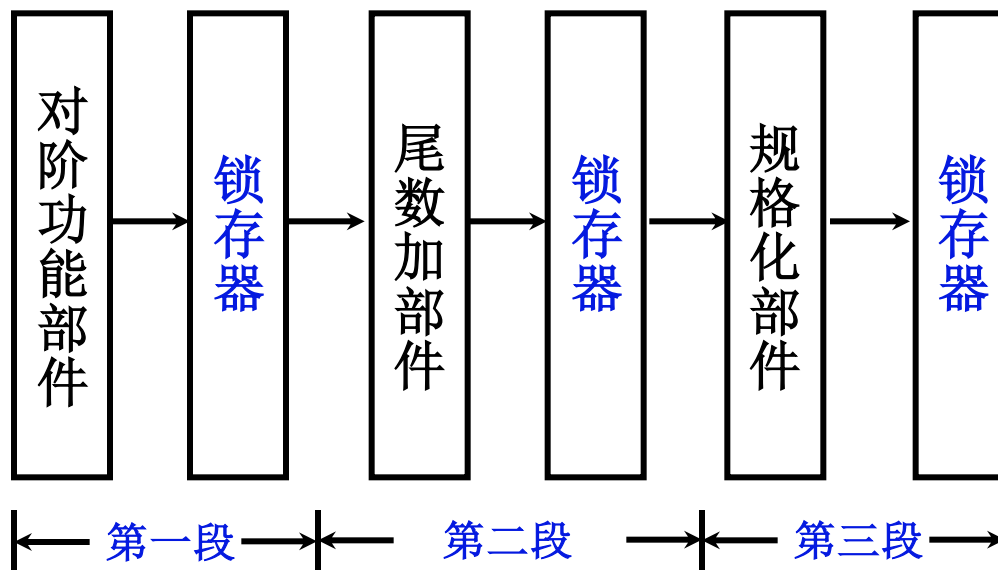
若 流水线不出现断流 1 个时钟周期出 1 结果

不采用流水技术 6 个时钟周期出 1 结果

理想情况下，6 级流水 的速度是不采用流水技术的 6 倍

2. 运算流水线

完成 浮点加减 运算 可分
对阶、尾数求和、规格化 三段



分段原则 每段 操作时间 尽量 一致

七、MIPS五段流水线

8.3

MIPS CPU实现方案

- 单周期方案

- 性能受限于最慢的指令
- 结构简单，实现容易

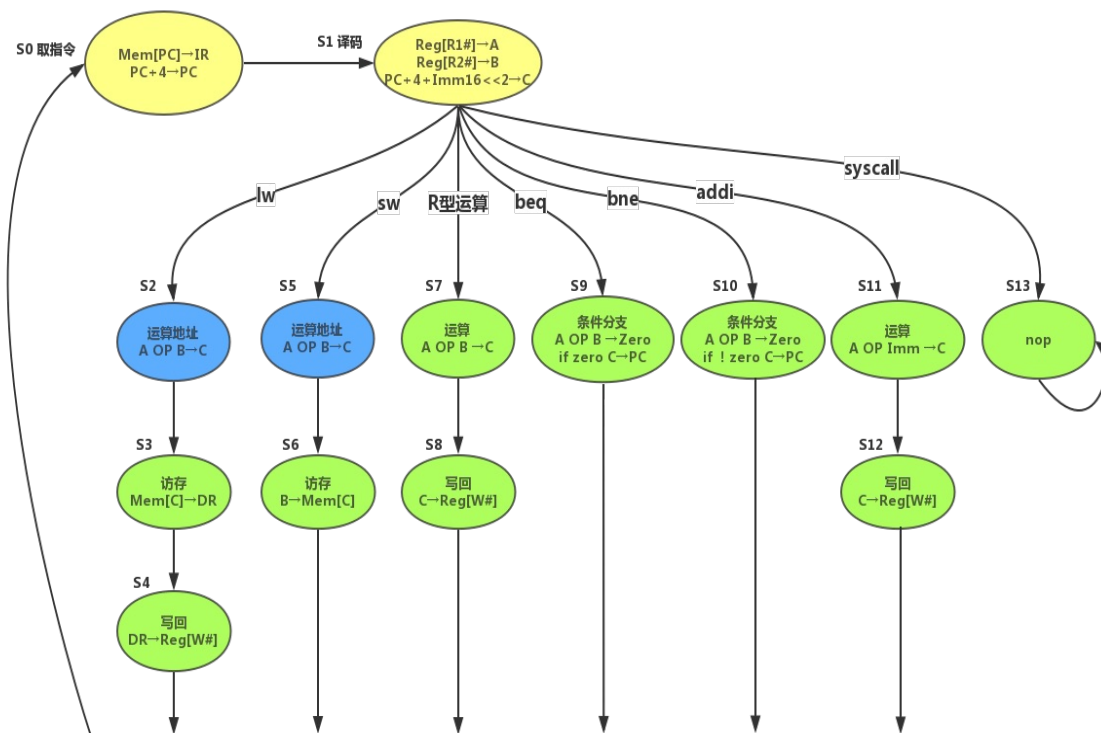
- 多周期方案

- 传统多周期**

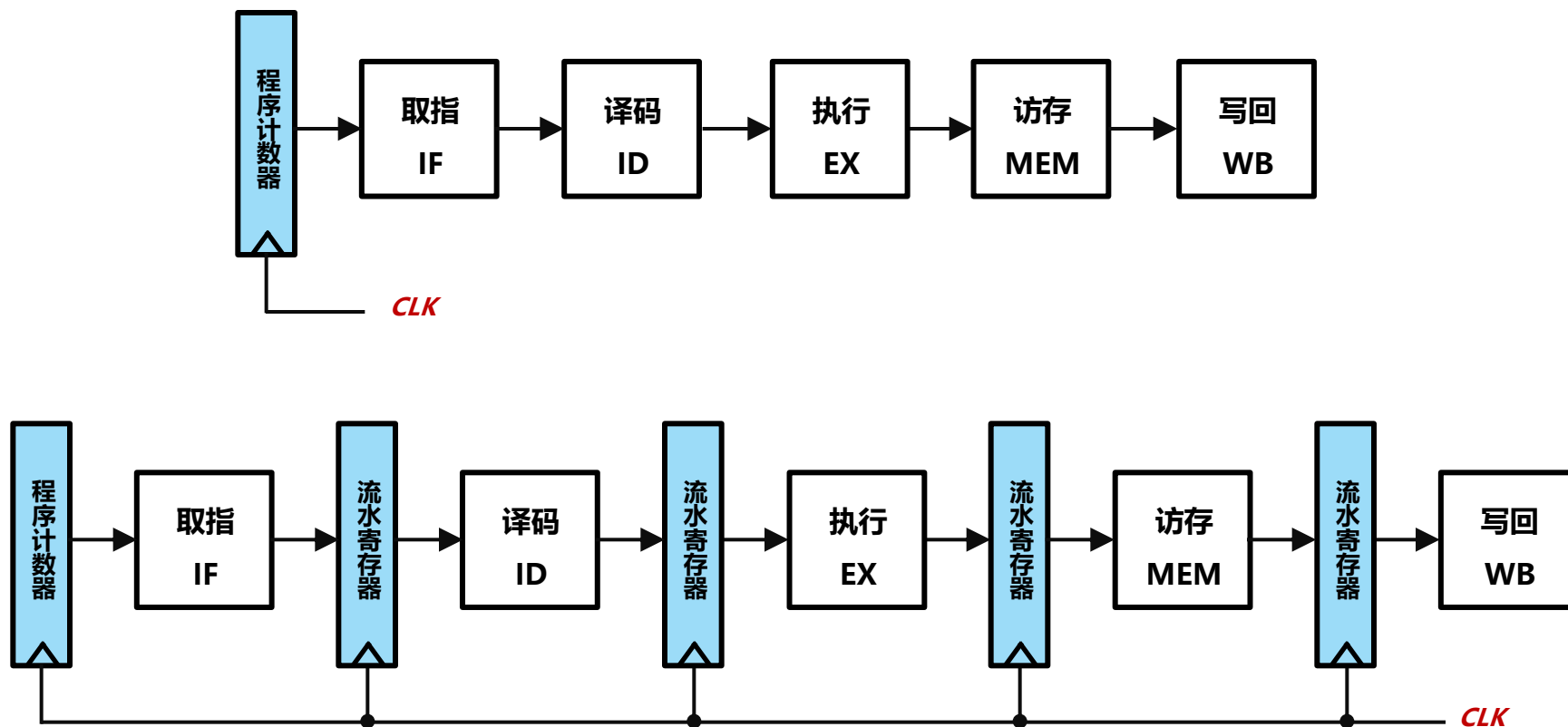
- 提升性能，复用器件
- 异步控制，变长指令周期

- 指令流水线**

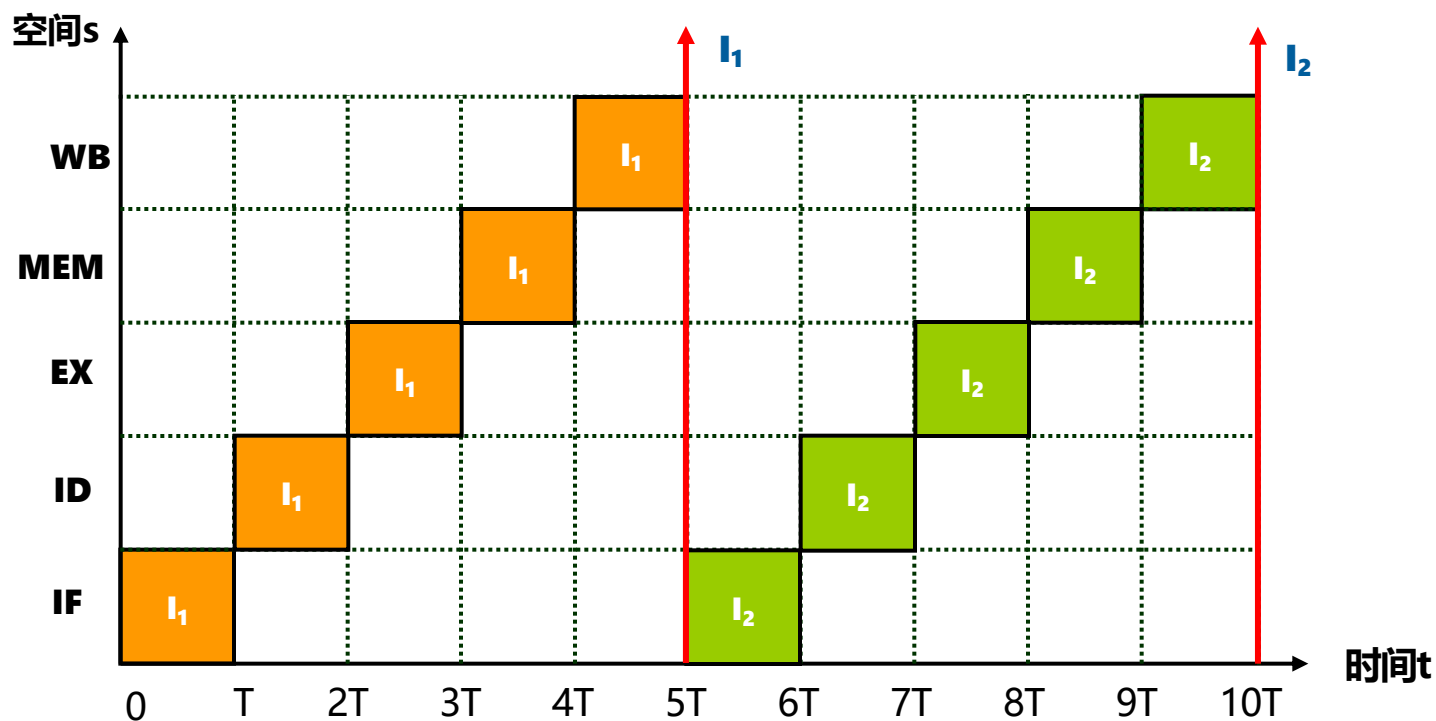
- 多指令并行，提升性能
- 部件并发



1. MIPS单周期、多周期与流水线



单周期时空图

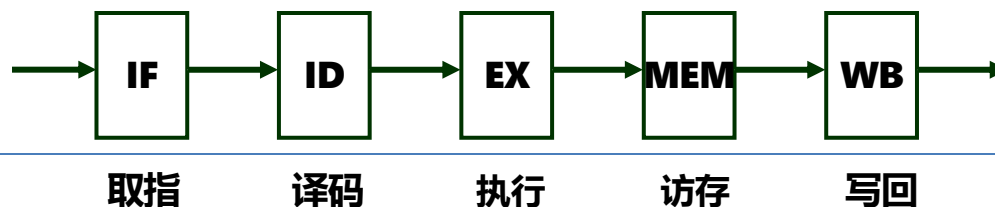
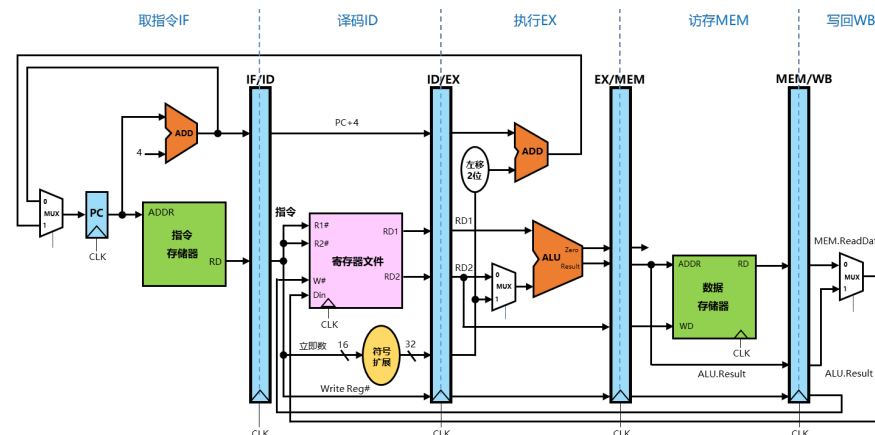


完成n条指令需要 $5nT$

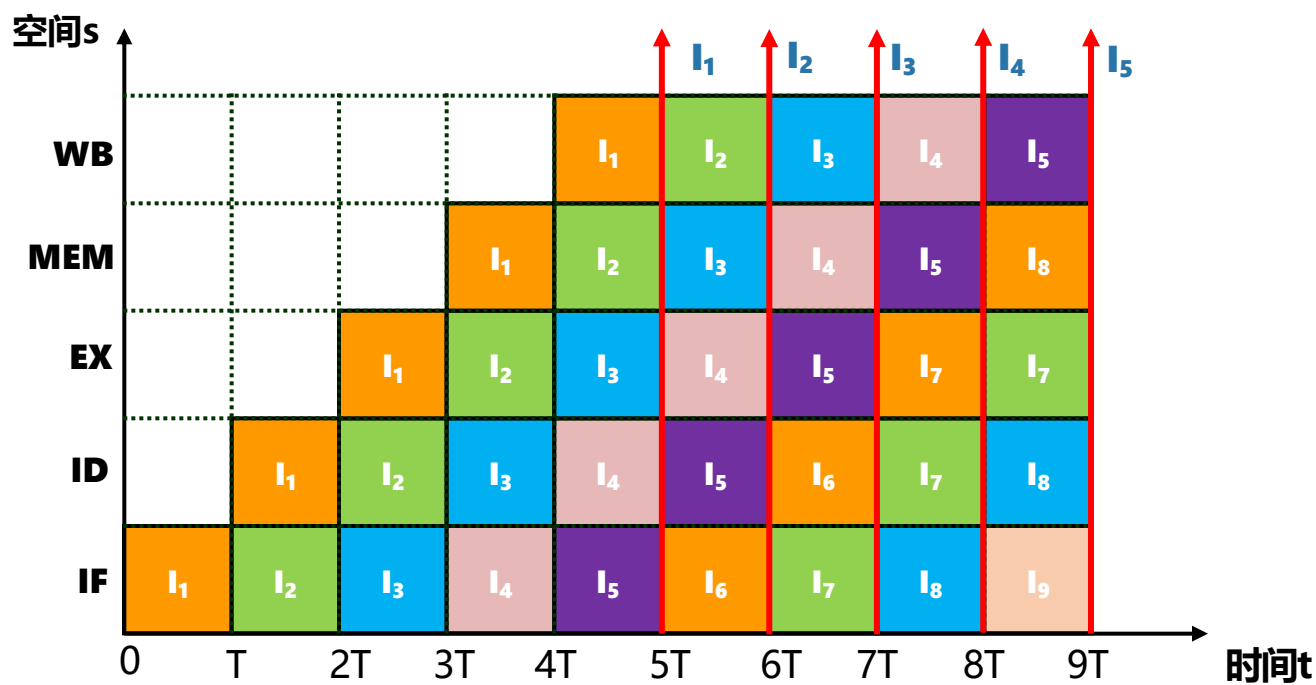
流水线指令运行动态

■ 数据通路细分为5段，各段完全并发

- 取指令 IF (Instruction Fetch)
- 指令译码 ID (Instruction Decode)
- 执行运算 EX (Execution)
- 访存阶段 MEM
- 结果写回 WB (Write Back)



指令流水线时空图

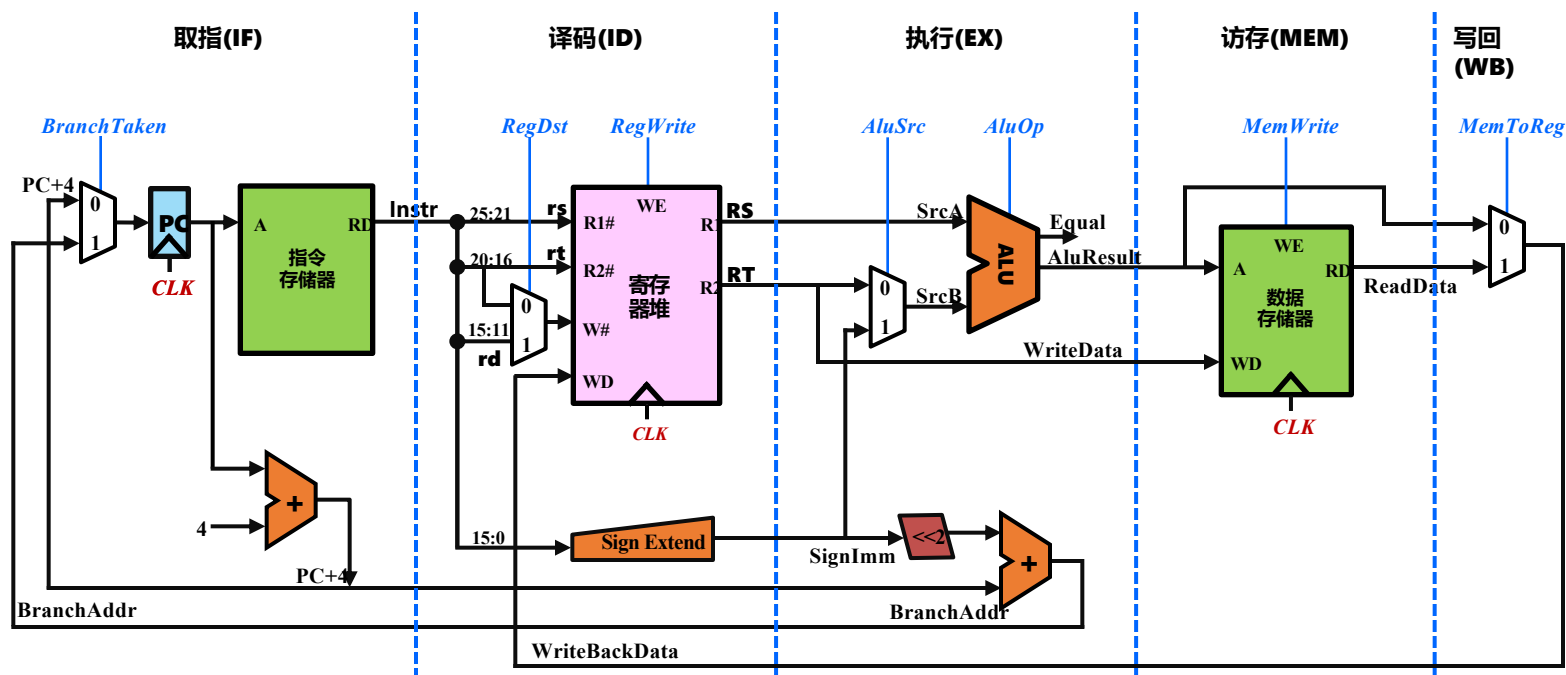


完成n条指令的时间=完成第一条指令时间5T+(n-1)*T=(n+4)T

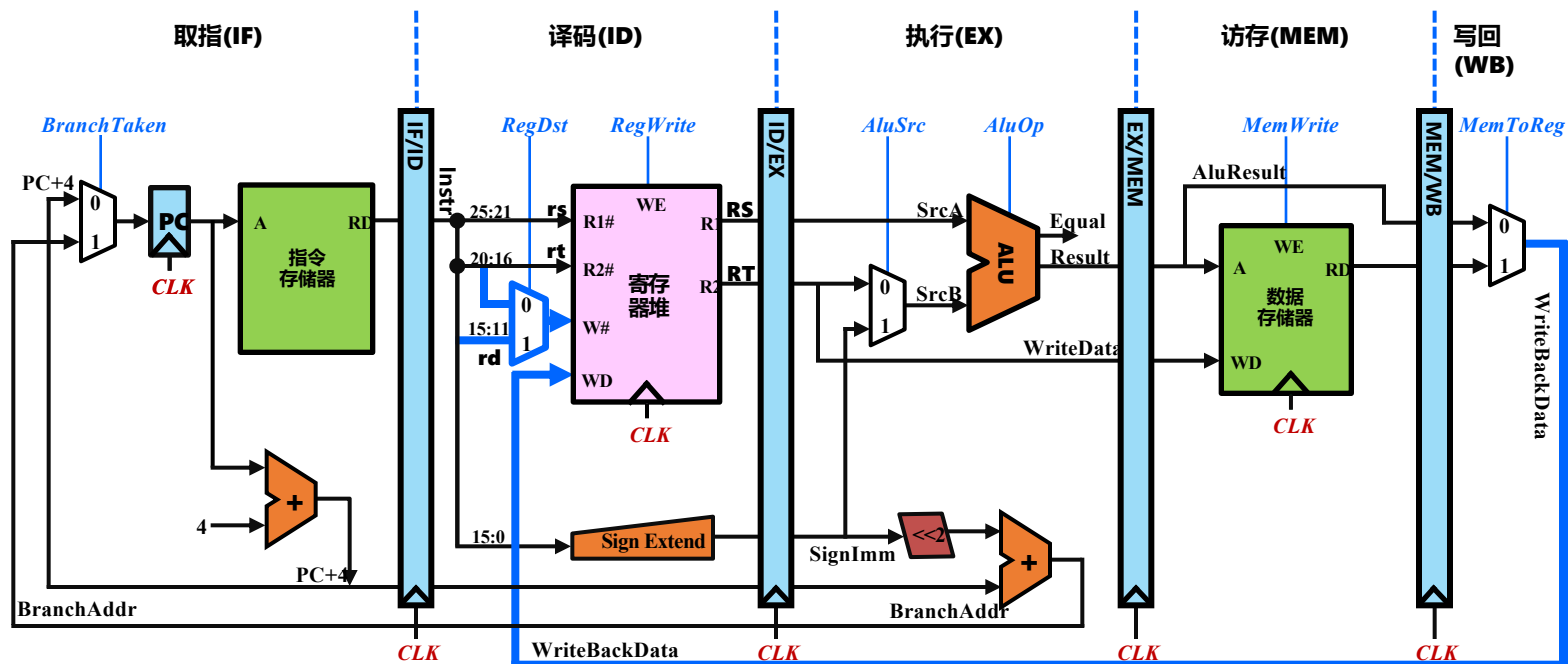
2.流水线数据通路

8.3

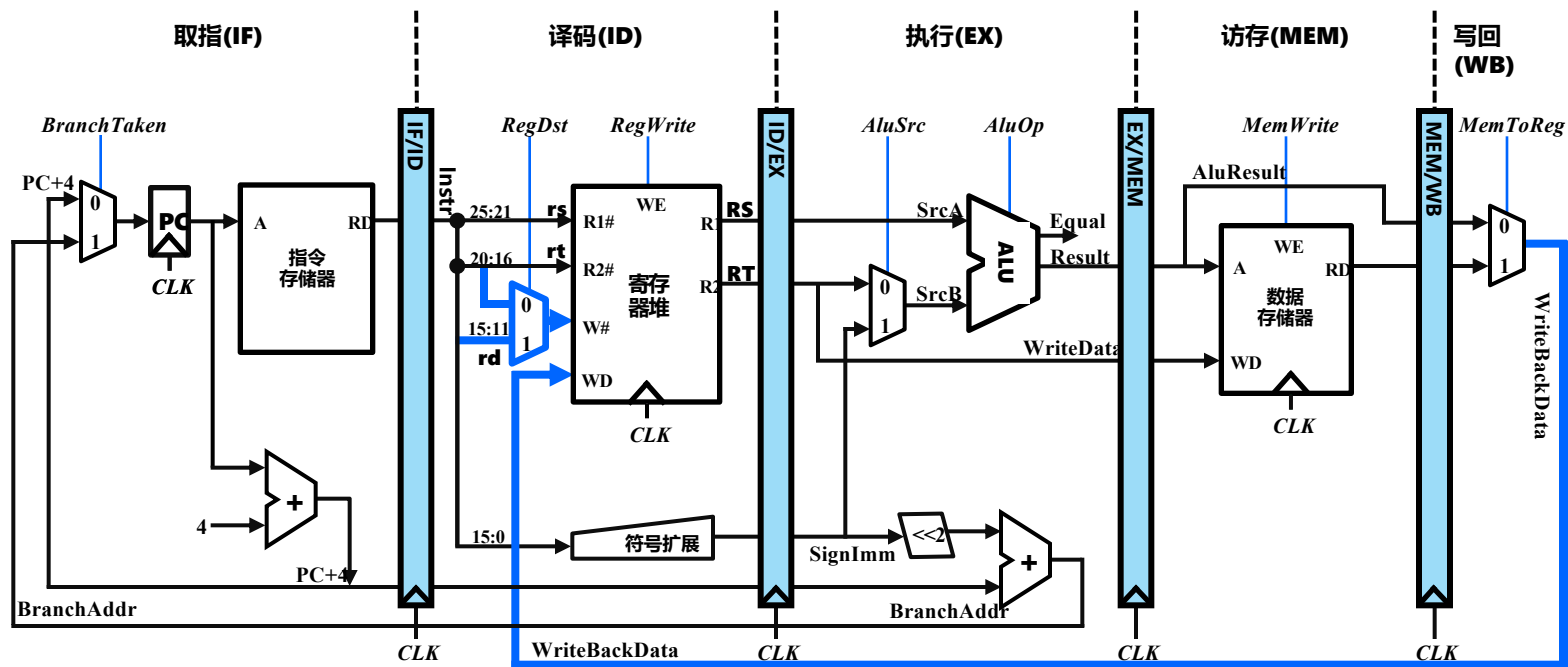
单周期MIPS处理器数据通路



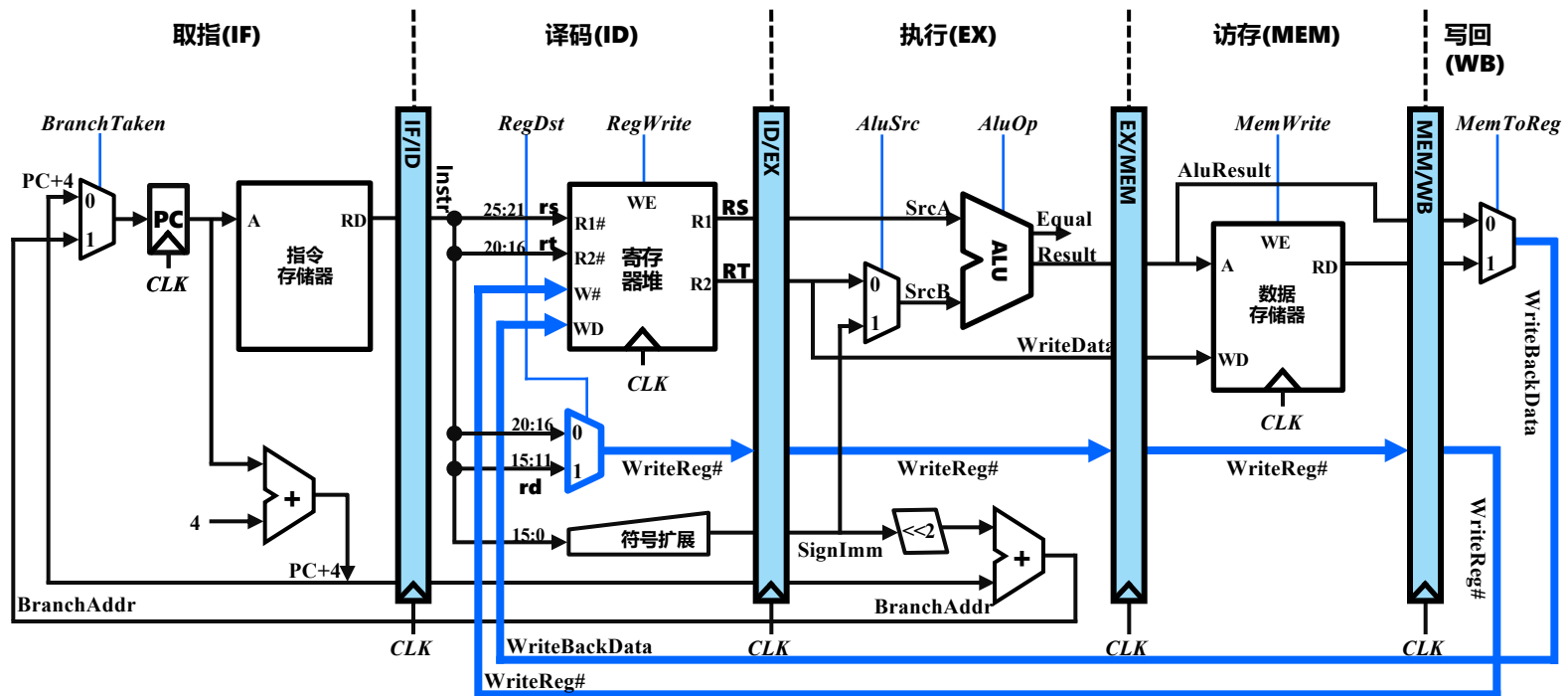
单周期MIPS数据通路流水线改造^{8.3}



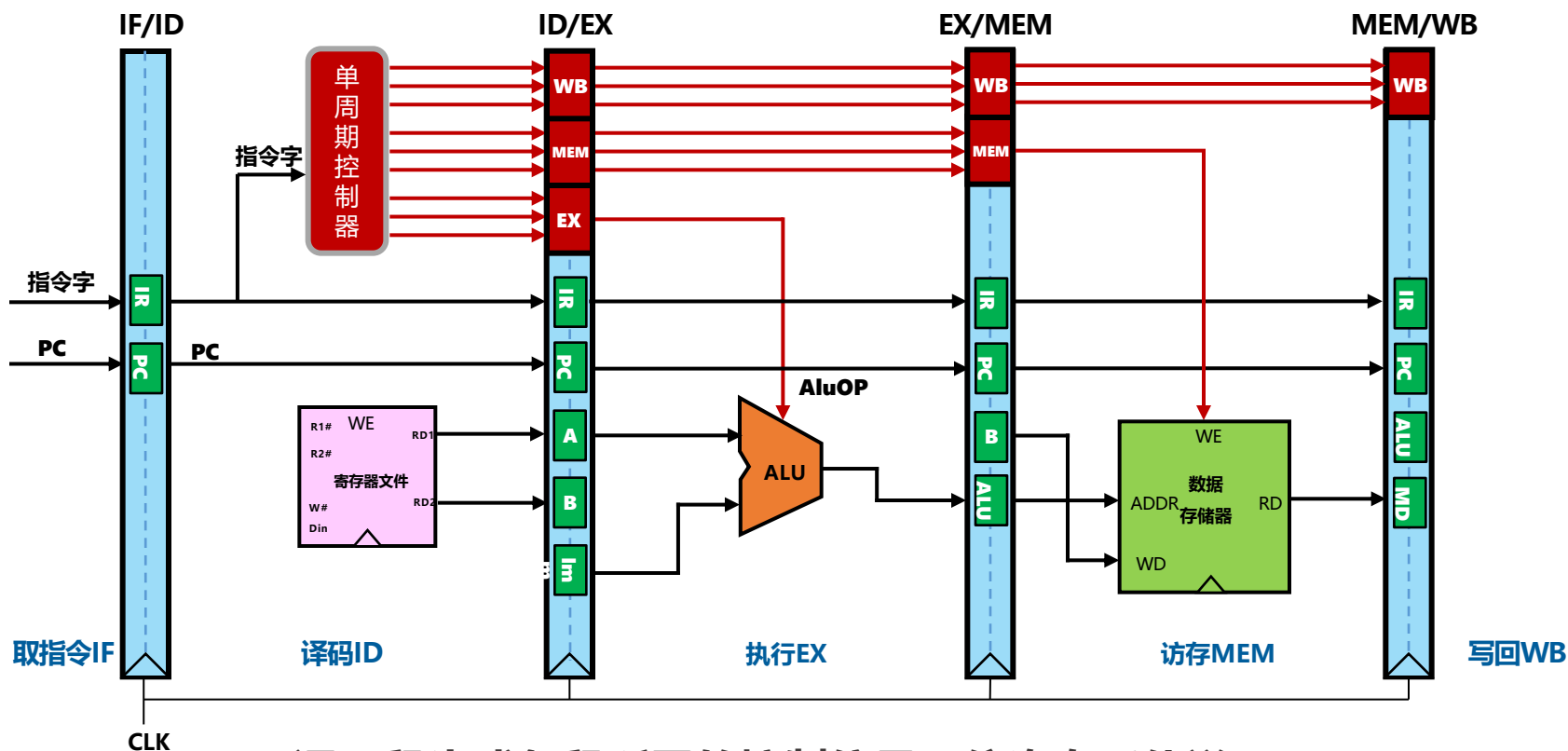
单周期MIPS数据通路流水线改造^{8.3}



流水线中写回数据通路改造^{8.3}



5段指令流水线数据与信号传递

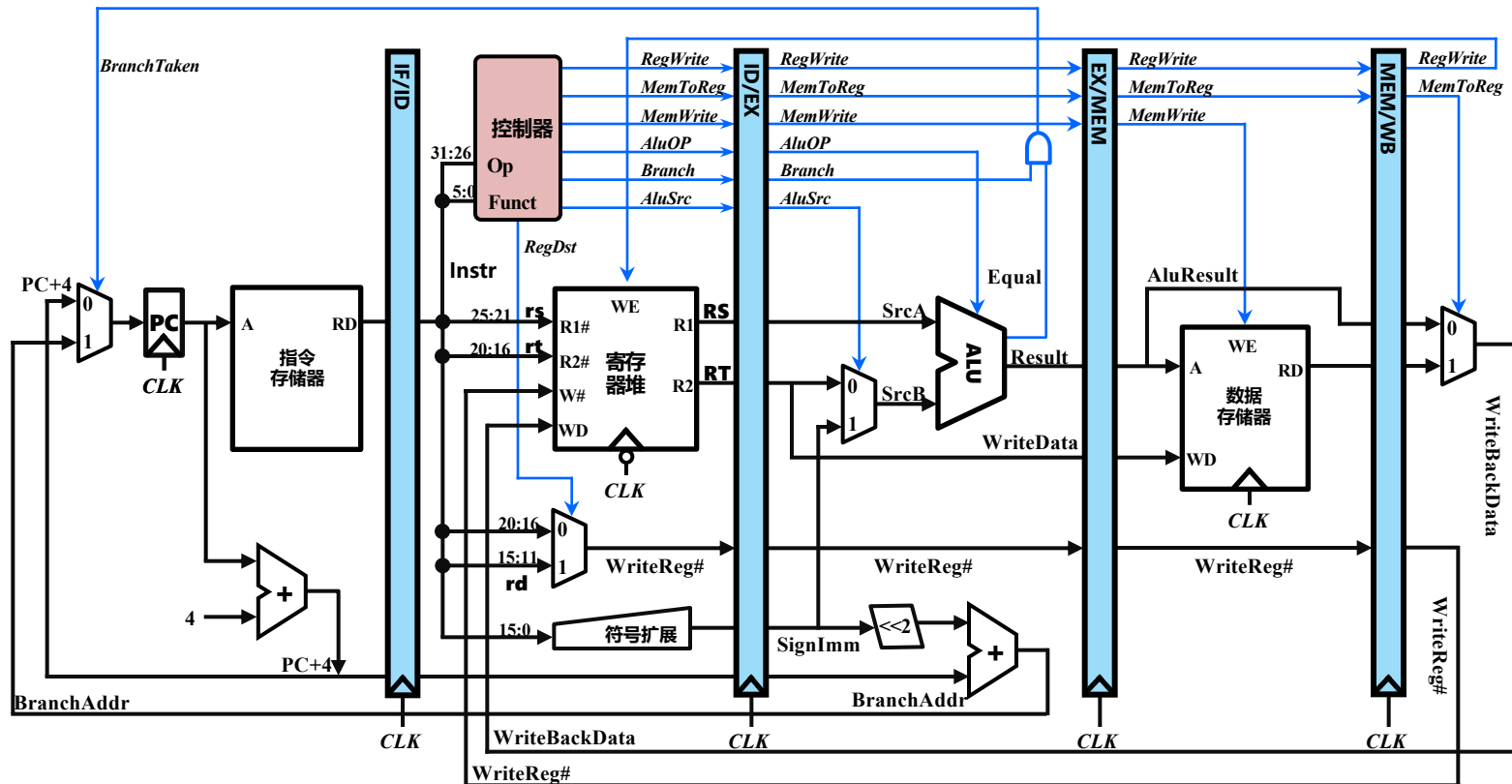


译码段生成各段所需的控制信号，依次向后传递

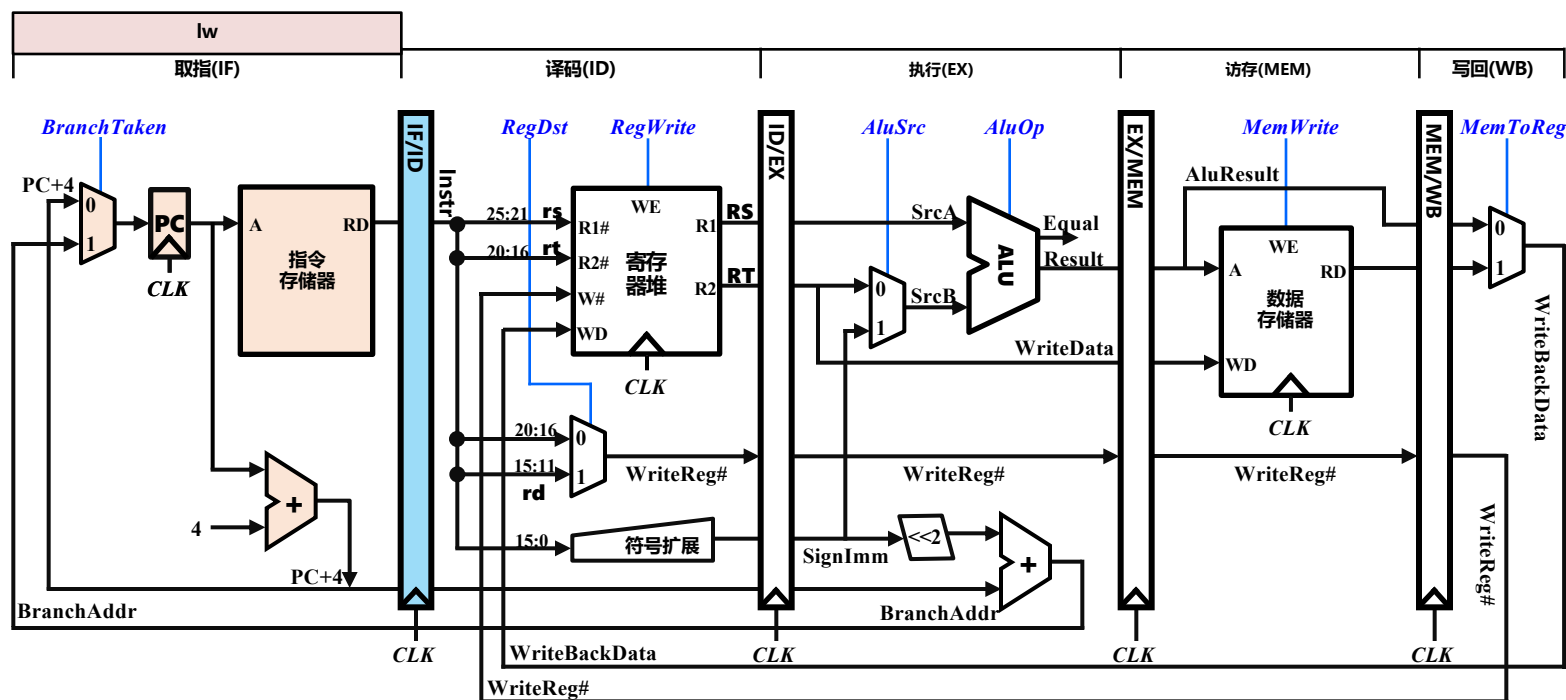
控制信号分类

控制信号	位置	来源	功能说明
BranchTaken	IF	EX	分支跳转信号，为 1 表示跳转，由 EX 段的 Branch 信号与 equal 标志进行逻辑与生成
RegDst	ID	ID	写入目的寄存器选择，为 1 时目的寄存器为 rd 寄存器，为 0 时为 rt 寄存器
RegWrite	ID	WB	控制寄存器堆写操作，为 1 时数据需要写回寄存器堆中的指定寄存器
AluSrc	EX	EX	ALU 的第二输入选择控制，为 0 时输入寄存器 rt，为 1 时输入扩展后的立即数
AluOp	EX	EX	控制 ALU 进行不同运算，具体取值和位宽与 ALU 的设计有关
MemWrite	MEM	MEM	控制数据存储器写操作，为 0 时进行读操作，为 1 时进行写操作
MemToReg	WB	WB	为 1 时将数据存储器读出数据写回寄存器堆，否则将 ALU 运算结果写回

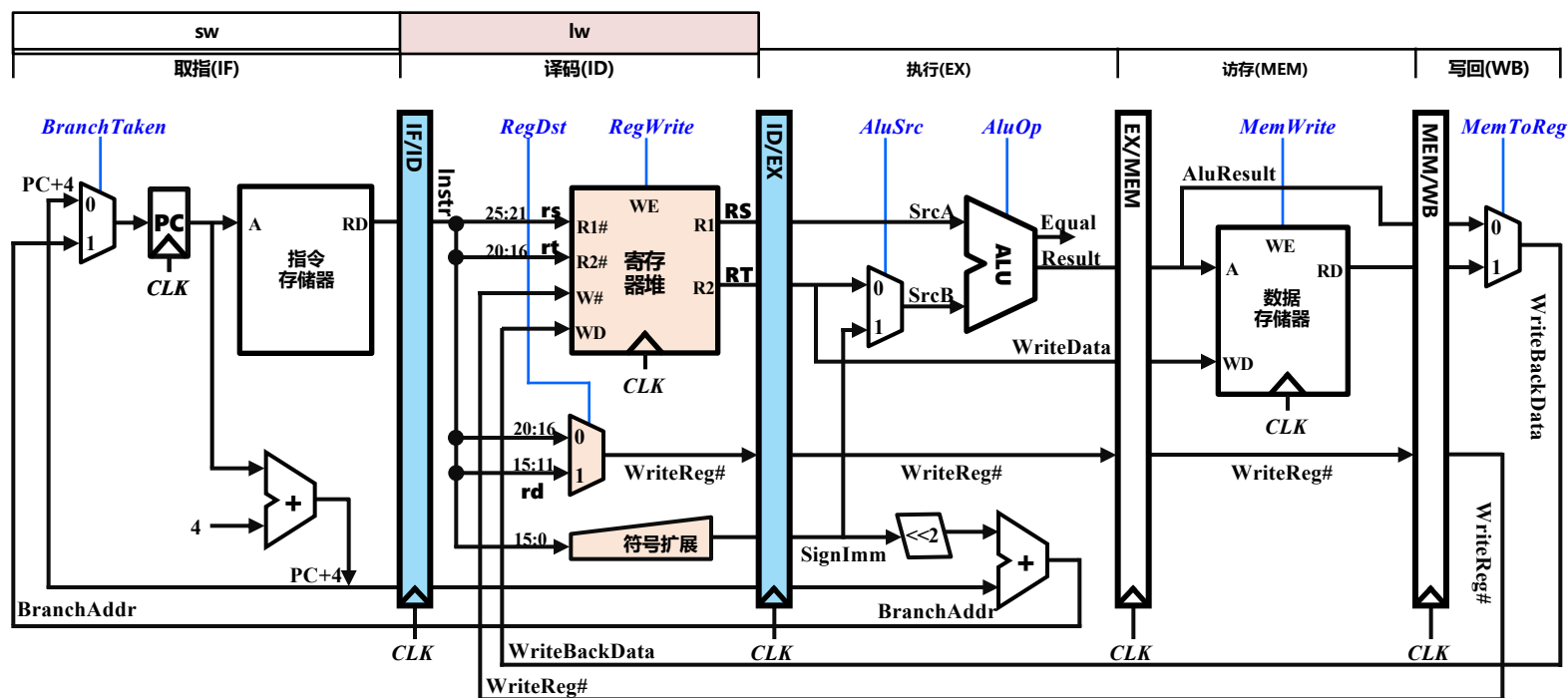
5段流水线控制信号与传递 8.3



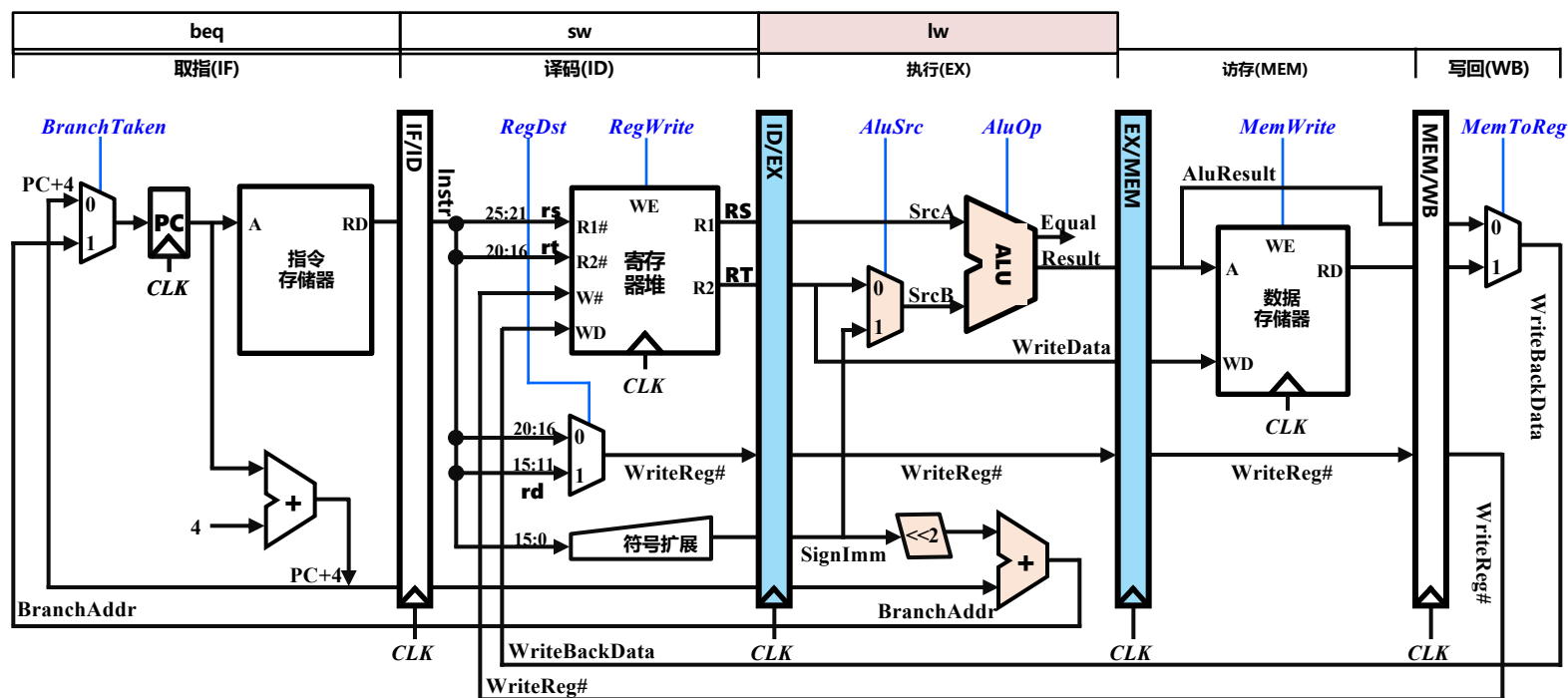
3. 指令在流水线中的执行过程



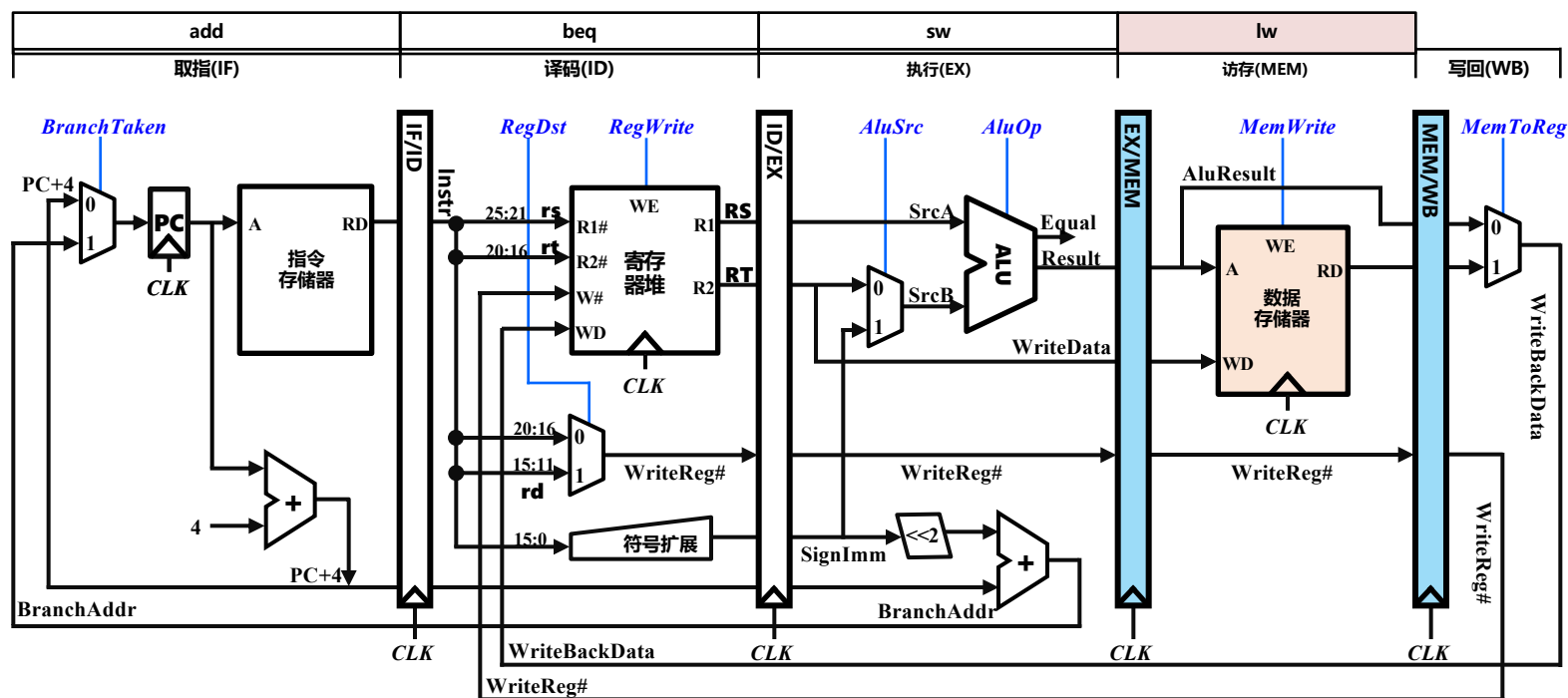
指令在流水线中的执行过程



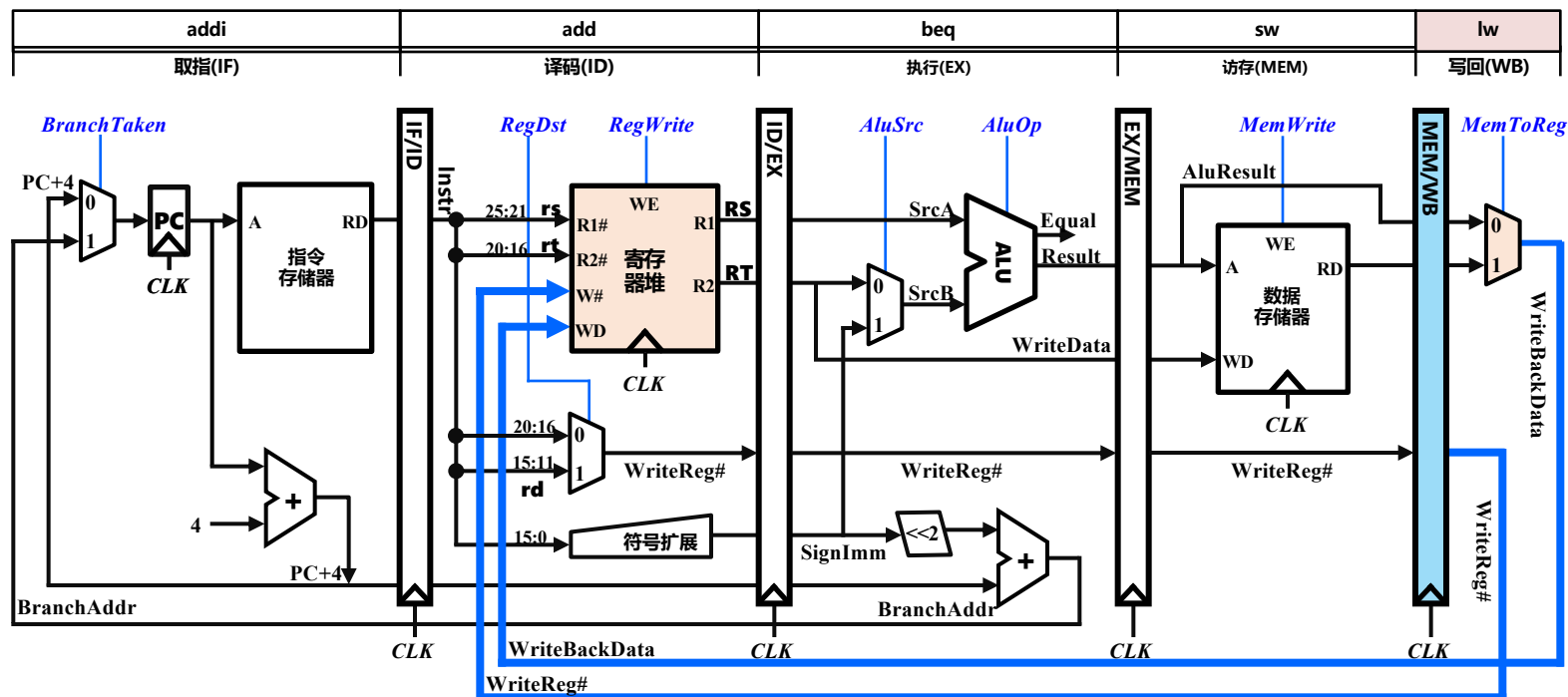
指令在流水线中的执行过程



指令在流水线中的执行过程



指令在流水线中的执行过程



MIPS流水线中的冲突与处理？