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

   存最常用的指令和数据

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

   部分并行　　并行

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　（不重叠，只是一个大时间段内都做了）
- 同时　两个或两个以上事件在 同一时刻 发生　（流水线属于后者）
}

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）（ILP）	细粒度	硬件实现

三、指令流水原理

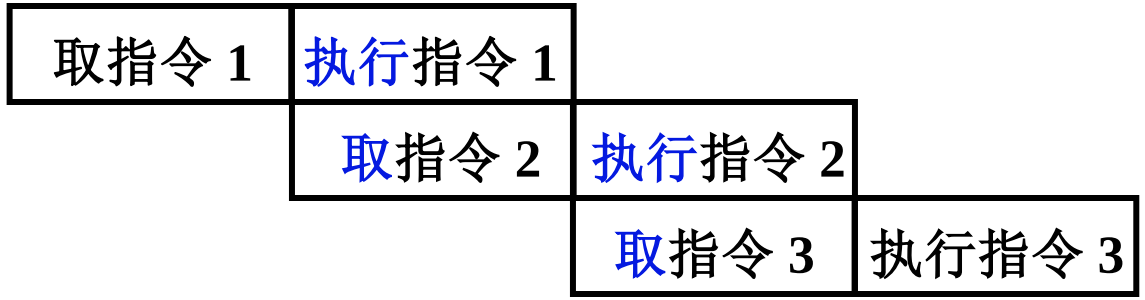
8.3

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

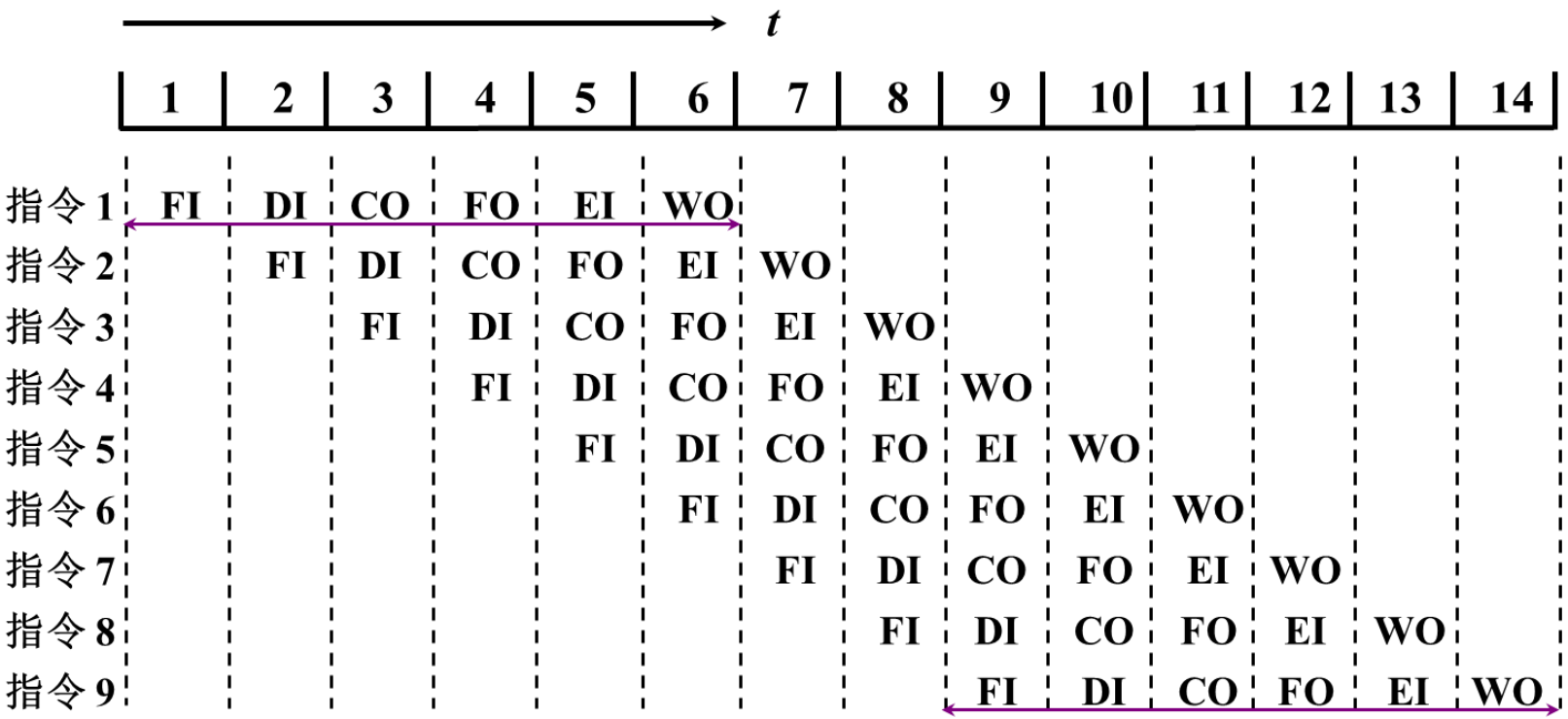
必须等 上条 指令执行结束，才能确定 下条 指令的地址，

造成时间损失　　猜测法

解决办法 ？　　“分支预测”

4. 指令的六级流水

8.3



完成 一条指令

串行执行

六级流水

6 个时间单位

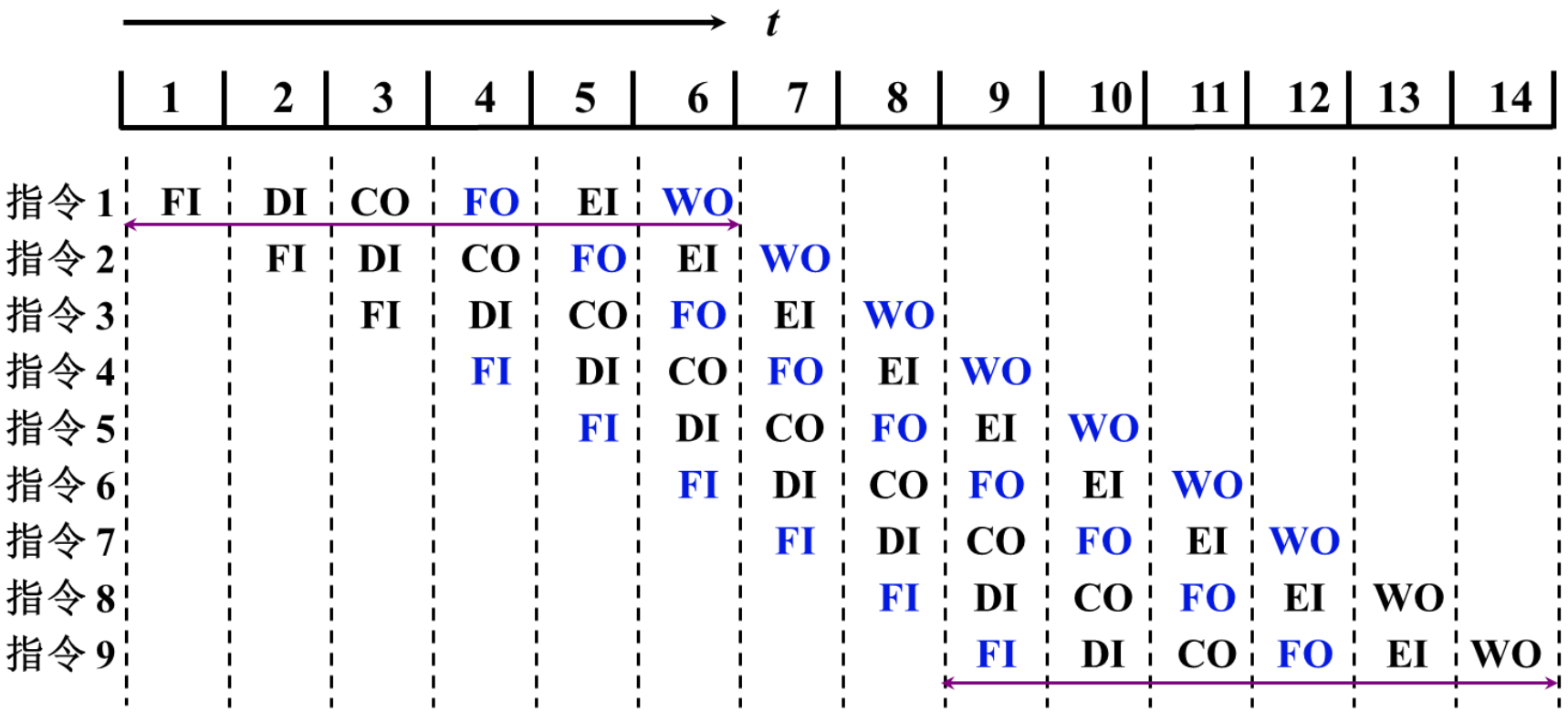
6 × 9 = 54 个时间单位

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1、指令3、指令 6 冲突

- 指令存储器和数据存储器分开

指令 2 与指令 5 冲突　　…

- 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）→ R_1

  ADD R_4，R_5，R_1 ；（R_5）+（R_1）→ R_4

- 读后写相关（WAR）

  STA M，R_2 ；（R_2）→ M 存储单元

  ADD R_2，R_4，R_5 ；（R_4）+（R_5）→ R_2

- 写后写相关（WAW）

  MUL R_3，R_2，R_1 ；（R_2）×（R_1）→ R_3

  SUB R_3，R_4，R_5 ；（R_4）－（R_5）→ R_3

等待结果写入后才执行下一条指令

解决办法 • 后推法 • 采用 旁路技术 不需要等到写入，算完就用

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

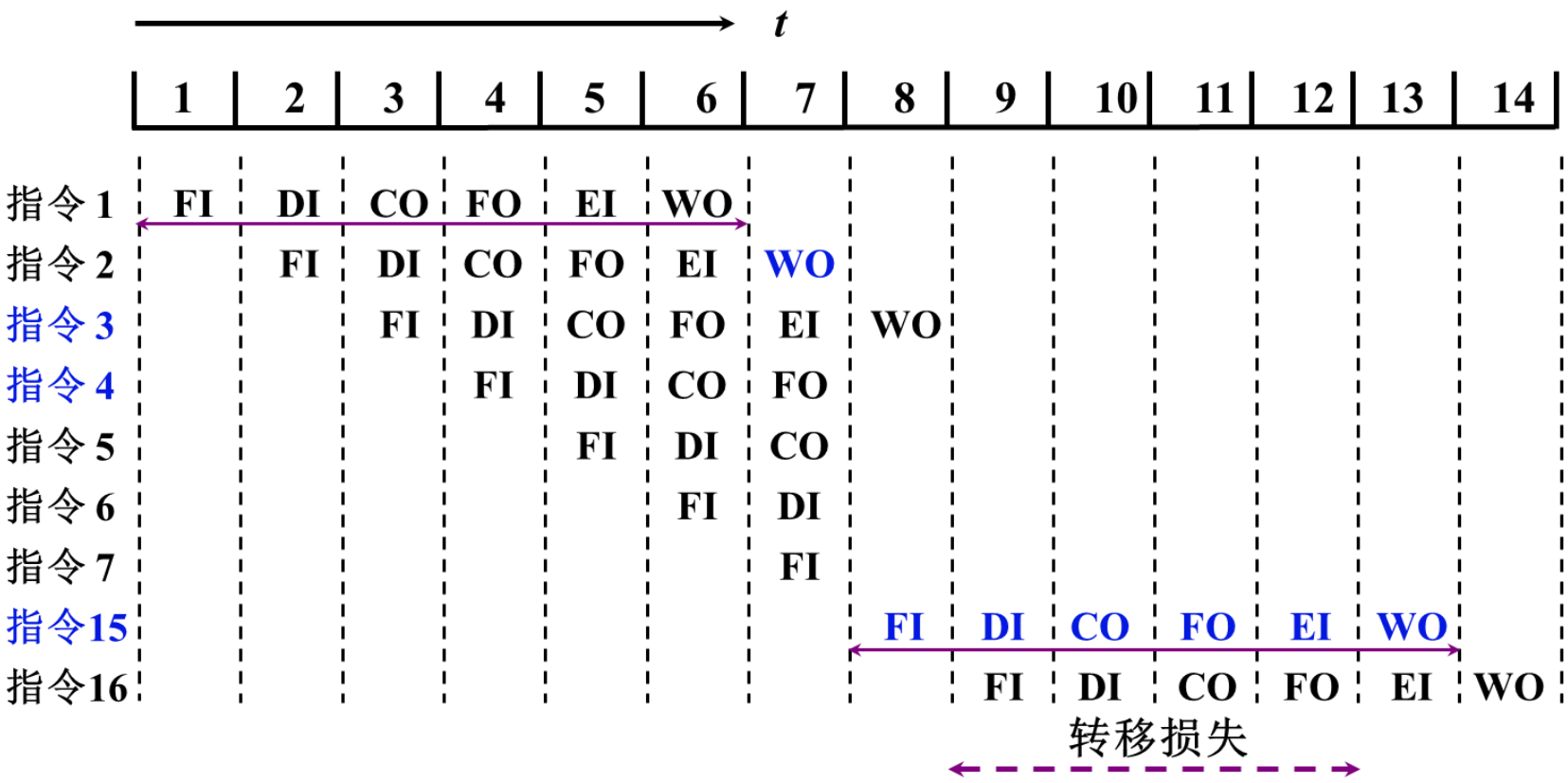
是转移

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

一共被分成m段

- 最大吞吐率　　满负荷运转，没有发生冲突etc

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

- 实际吞吐率

连续处理 n 条指令的吞吐率为

总共输出n个结果

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

第一个指令的时间　　以后每经过Dt，都有一个指令完成

2. 加速比 S_p

8.3

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

$$T = m\,\Delta t + (n\text{-}1)\,\Delta t$$

完成 n 条指令在等效的非流水线上共需

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

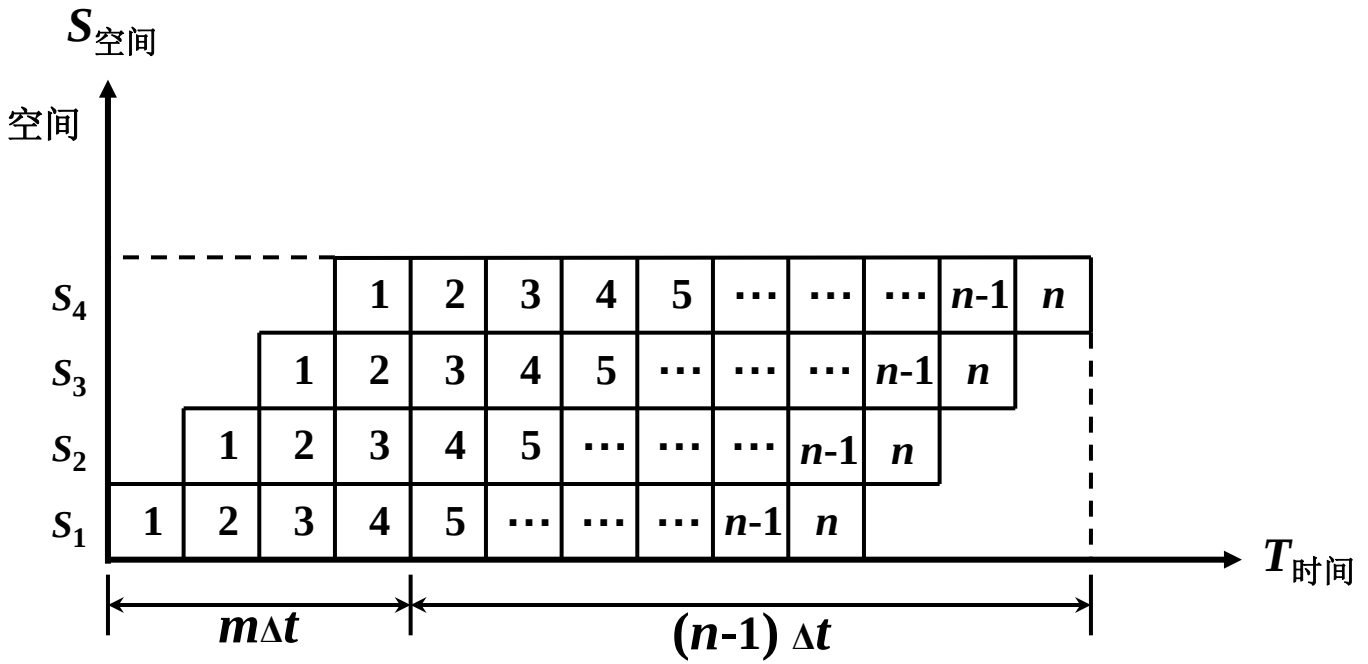
则

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

3. 效率

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

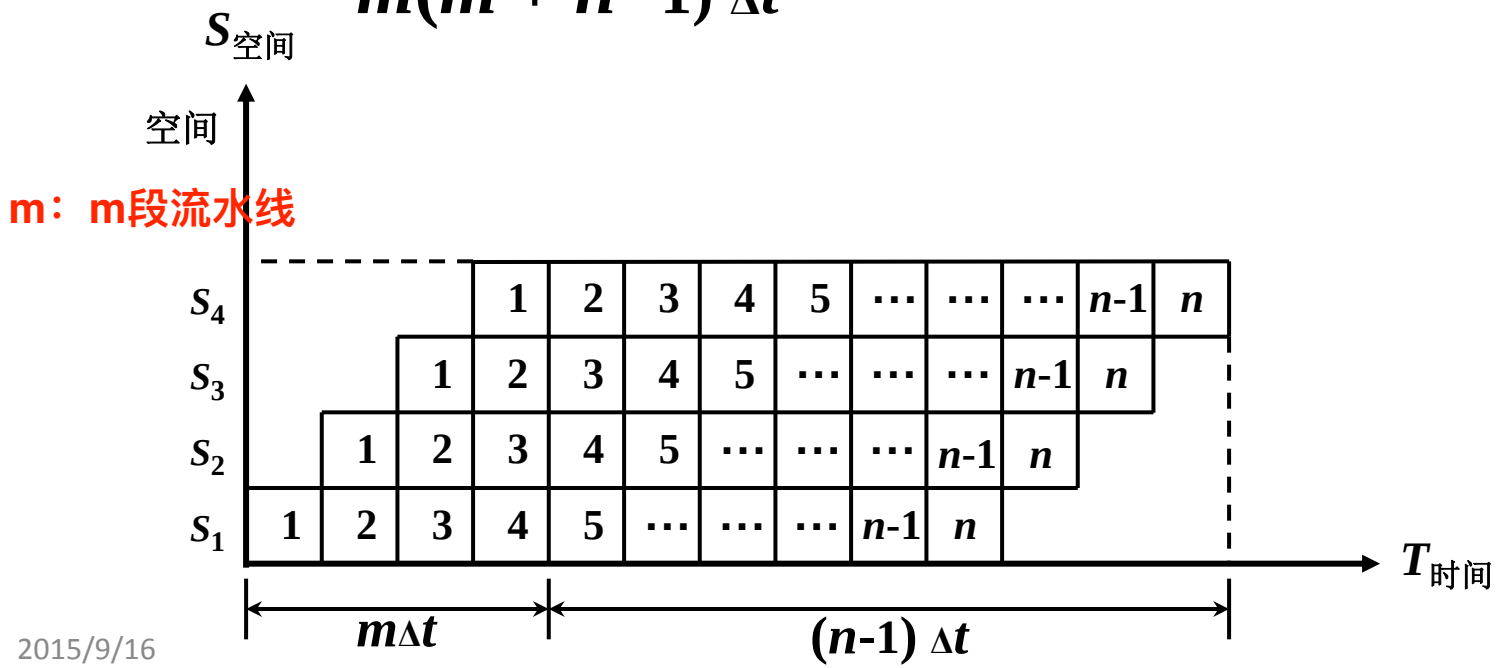
8.3

流水线中各功能段的 利用率

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

S执行/S总

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



五、流水线的多发技术

8.3

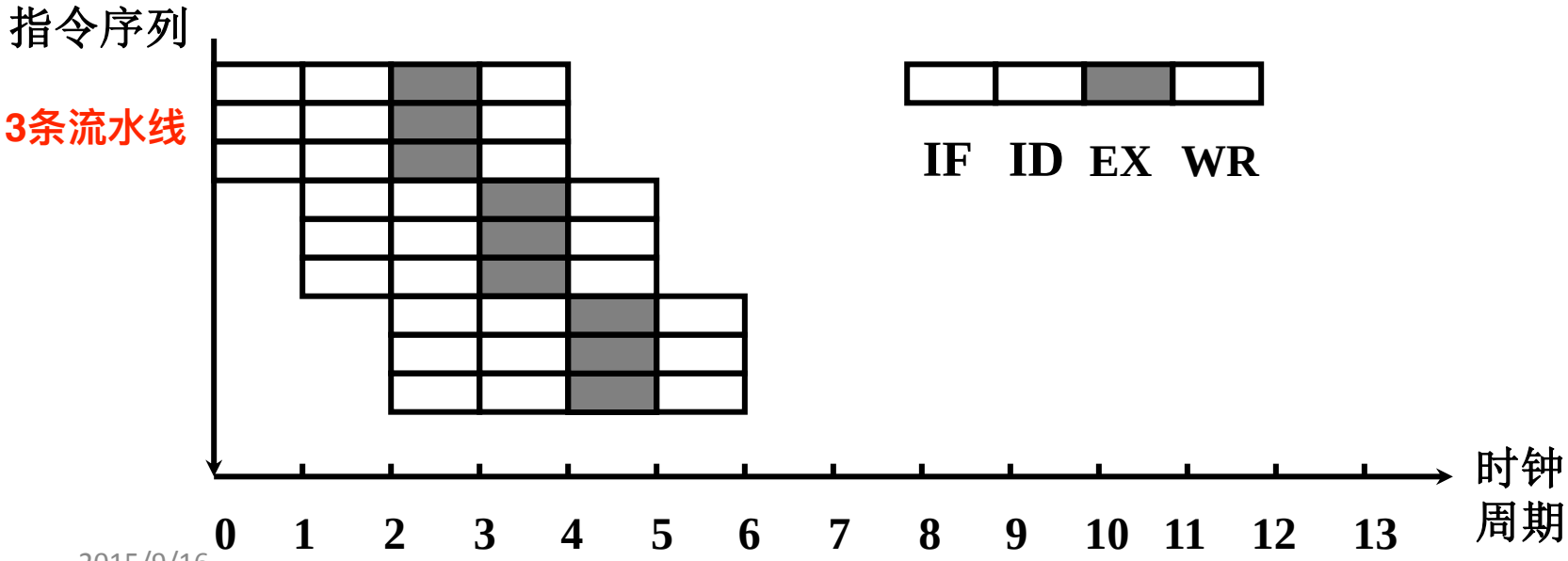
1. 超标量技术　使用多条流水线

- 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

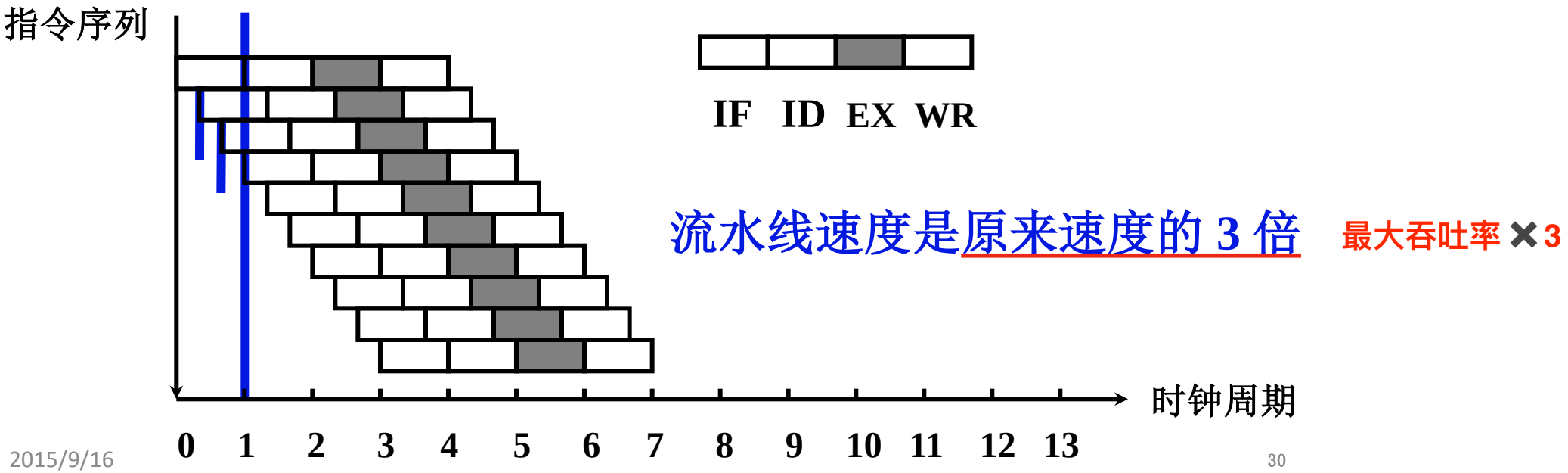
8.3

- 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



流水线速度是原来速度的 3 倍　最大吞吐率 ✕3

3. 超长指令字技术

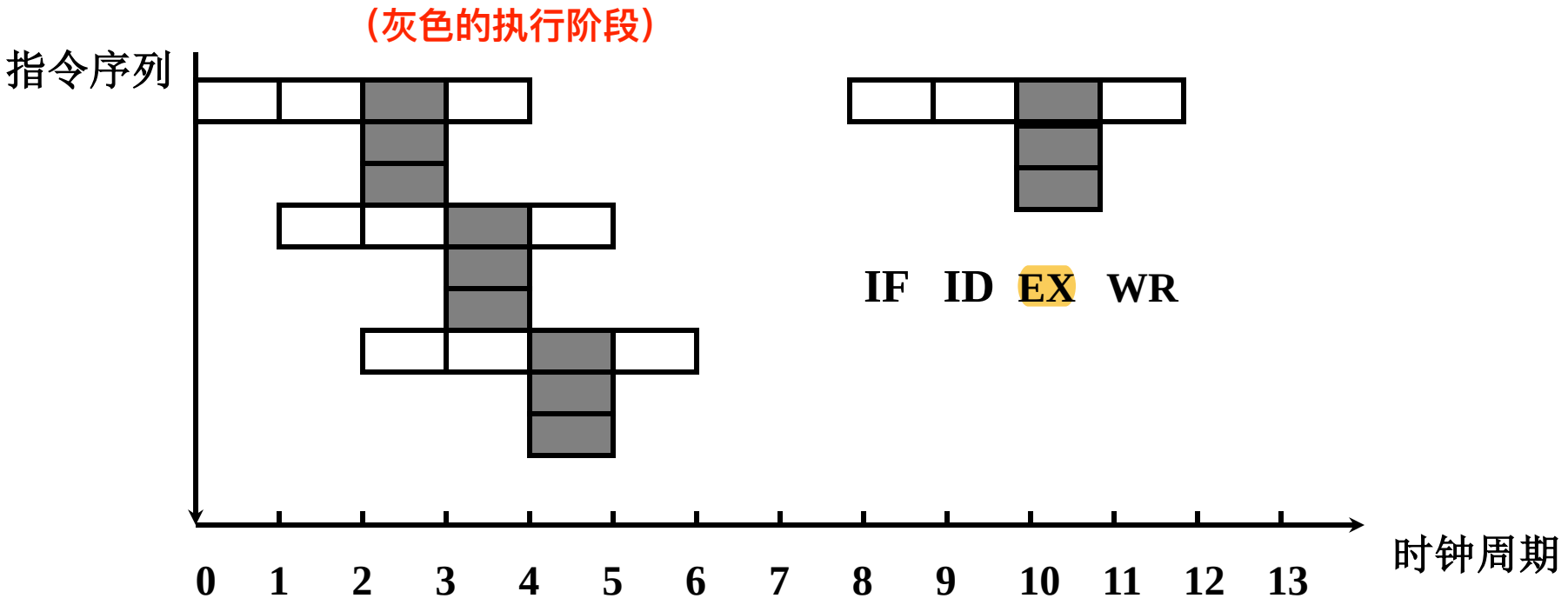
8.3

- 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

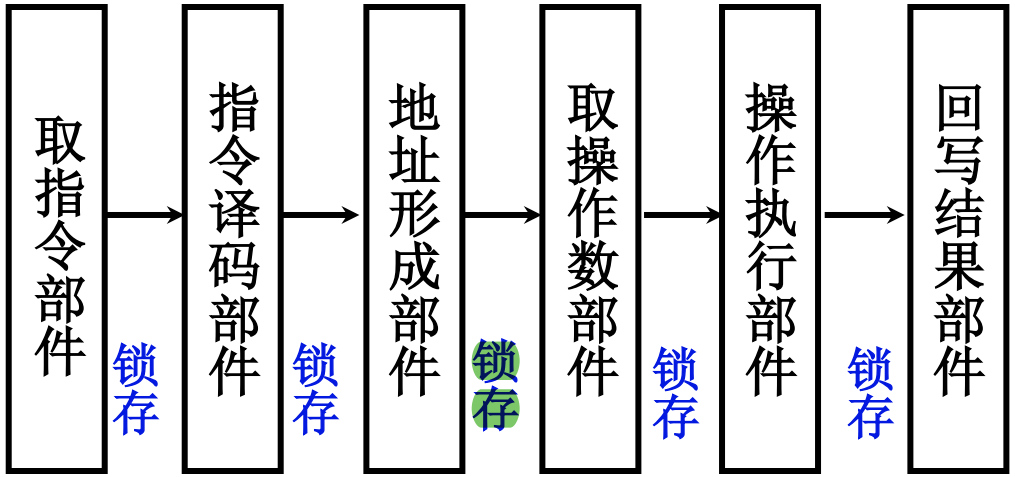


六、流水线结构

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

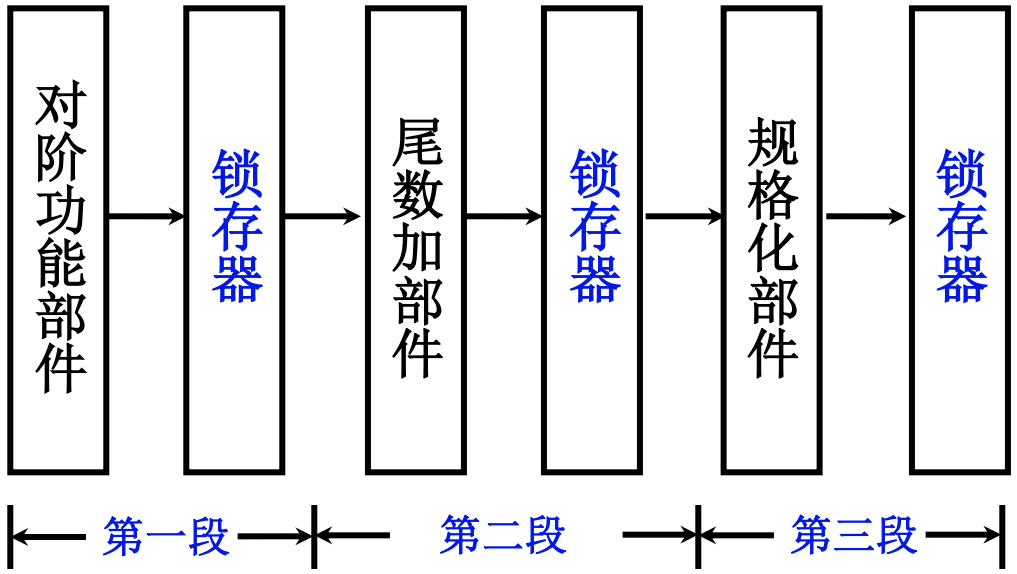
理想情况下，6 级流水 的速度是不采用流水技术的6 倍

2. 运算流水线

8.3

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