

计算机组织与体系结构

第十二讲

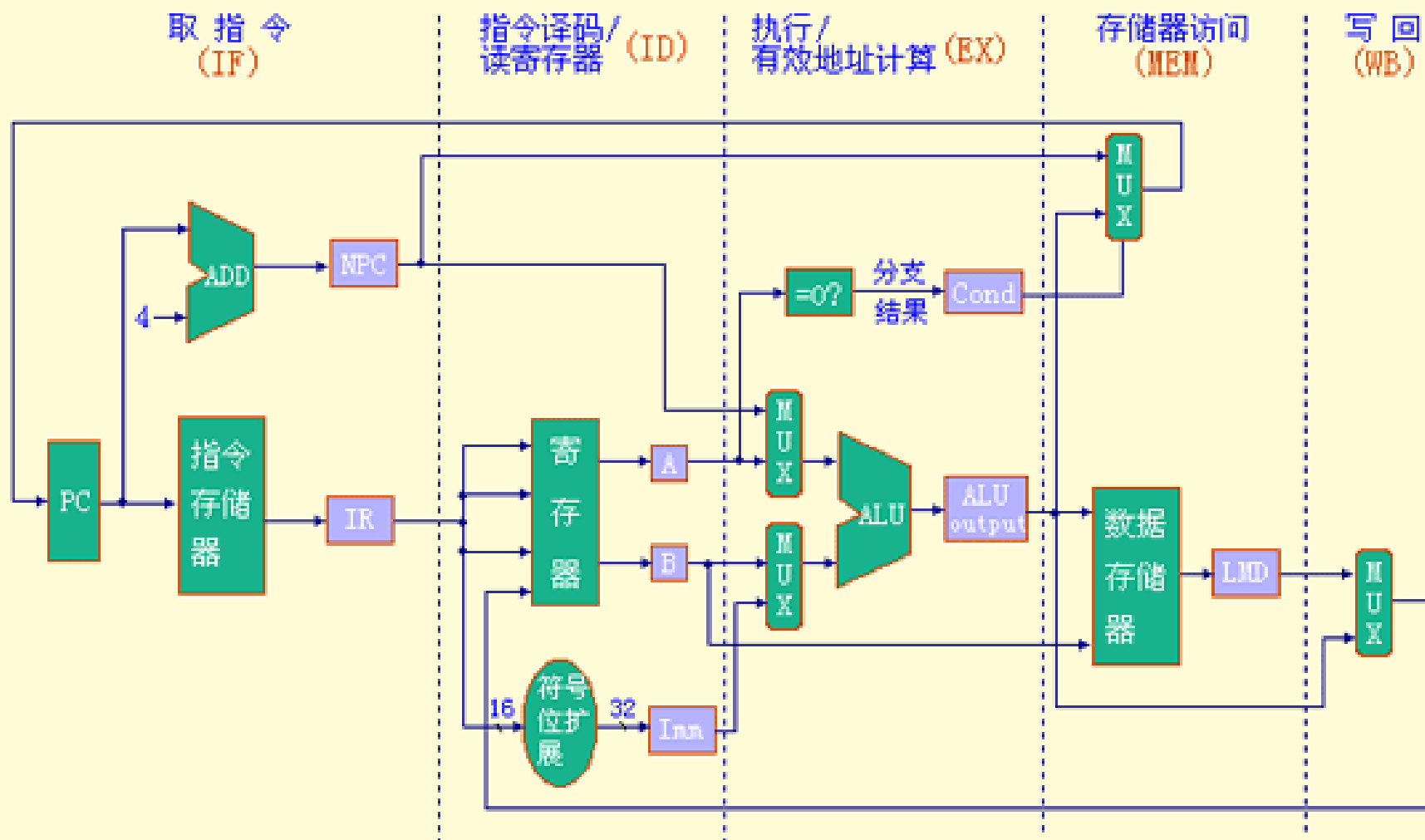
计算机科学与技术学院

舒燕君

Recap

- 多级时序系统（**x86**）
 - ✓ 控制方式（同步控制、异步控制、联合控制、人工控制）
 - ✓ 实例：**8085**（以输出指令为例）
- **MIPS CPU**的简单实现
 - ✓ **CPU** 的结构（**32**个寄存器、**1**个**PC**、**1**个**IR**、指令存储器和数据存储器分离）
 - ✓ 给定指令系统（**8**条指令）
 - ✓ 指令的格式描述
 - ✓ 数据通路（取指、译码、执行、访存、写回）
 - ✓ 性能分析及改进

实现DLX指令的一种简单数据通路



第 1 章 计算机系统概论

第 2 章 计算机系统量化分析基础

第 3 章 总线

第 4 章 指令系统

第 5 章 CPU设计与实现

第 6 章 基本流水线技术

第 7 章 指令级并行

第 8 章 存储系统的结构与优化

第 9 章 IO系统

第6章 流水线技术

6.1 流水线概述

6.2 MIPS的基本流水线

6.3 流水线中的冲突

6.4 实例分析：MIPS R4000

6.5 向量处理机

6.1 流水线概述

6.1.1 流水线基本概念

6.1.2 流水线分类

6.1.1 流水线的基本概念

1. 产品生产流水线

(1) 一个问题

假设某产品的生产需要4道工序，该产品生产车间以前只有1个工人，1套生产该产品的机器。该工人工作8小时，可以生产120件（即每4分钟生产1件）。

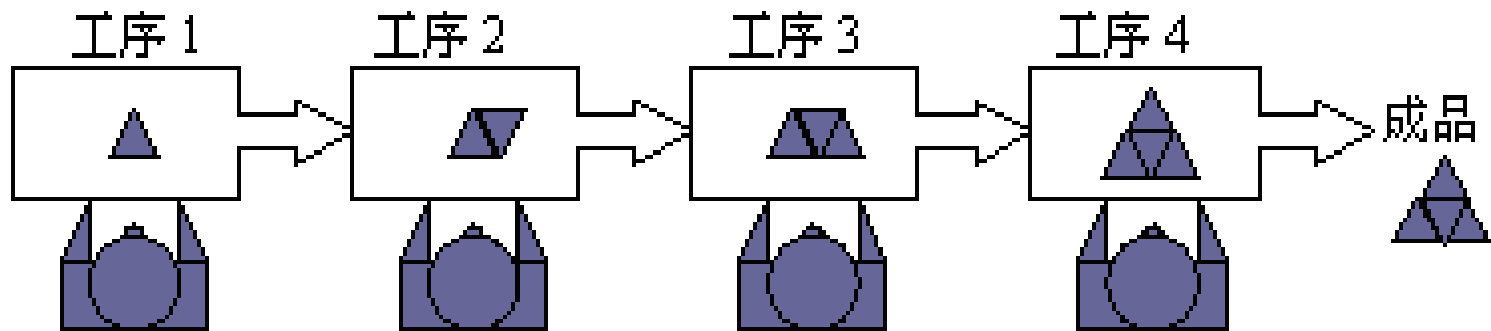
要将该产品日产量提高到480件，如何能实现目标？

6.1.1 流水线基本概念

(2) 两种解决方案

方案一：增加3名工人、3套设备。

方案二：产品生产采用流水线方式，分为4道工序；增加3名工人，每人负责一道工序。



6.1.1 流水线基本概念

(3) 两种方案的工作过程对比

两种方案中，单件产品的生产时间均不变。

但在稳定情况下，

方案一：每4分钟，4件产品同时进入流水线，4件成品同时离开流水线，需要增加3套设备。

方案二：每分钟，1件产品进入流水线，1件成品离开流水线，不需要增加任何设备。

6.1.1 流水线基本概念

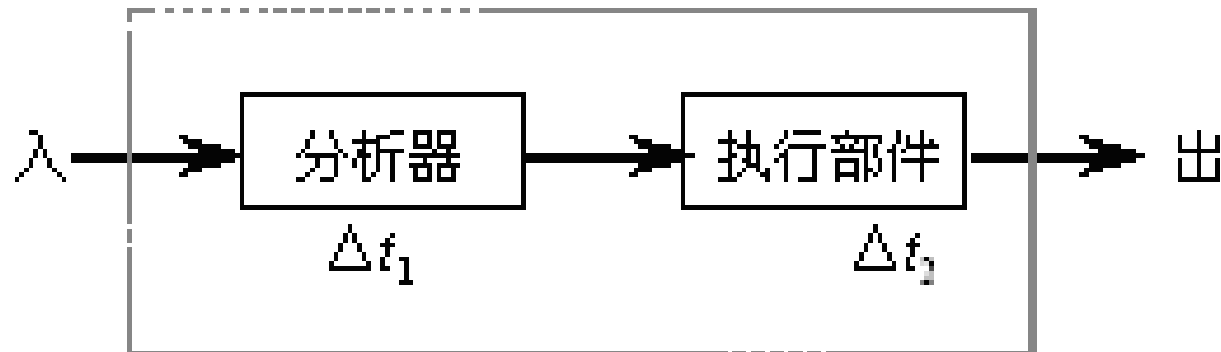
(4) 方案二的主要特点

每件产品还是要经过4道工序处理，单件产品的加工时间并没有改变，但它将各个工人的操作时间重叠在一起，使得每件产品的产出时间从表面上看是从原来的4分钟缩减到1分钟，提高了产品的产出率。

6.1.1 流水线基本概念

2. 计算机中的流水线

◆ 指令流水线



◆ 功能部件流水线

6.1.1 流水线基本概念

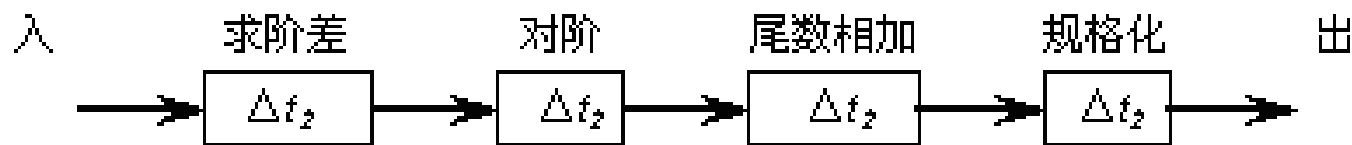
3.流水技术的定义

将一重复的时序过程分解为若干子过程，每个子过程都可有效地在其专用功能段上与其它子过程同时执行，这种技术称为流水技术。

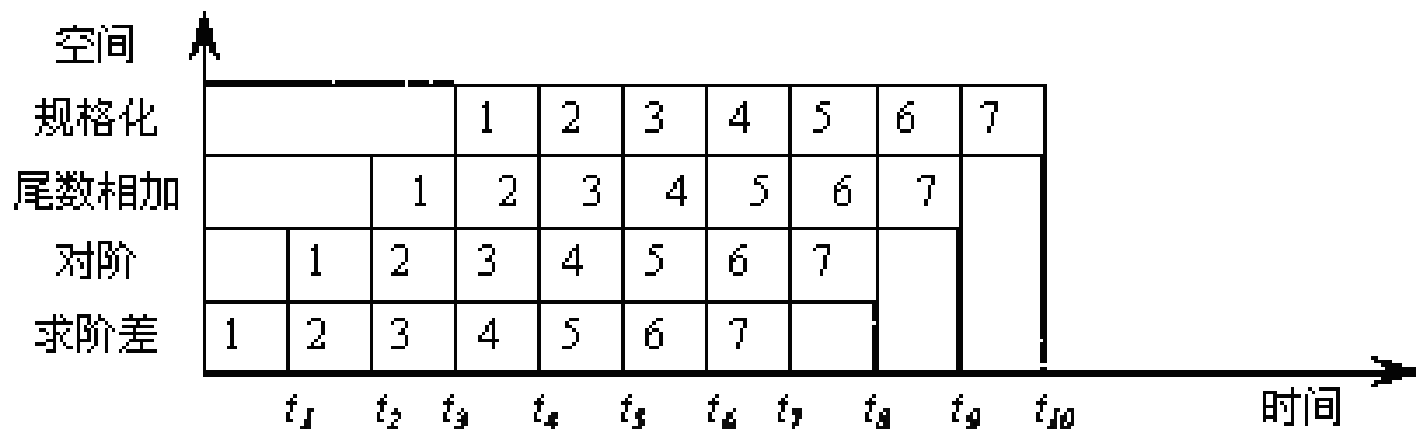
4.时空图

从时间和空间两个方面描述流水线的工作过程，横坐标表示时间，纵坐标表示各流水段。

流水线技术原理



(a) 浮点加法流水线



(b) 描述流水线工作的时空图

6.1.1 流水线基本概念

5.流水线的特点

- ① 流水过程由多个相关的子过程组成，这些子过程称为流水线的“级”或“段”。段的数目称为流水线的“深度”。
- ② 每个子过程由专用的功能段实现。
- ③ 各功能段的时间应基本相等，通常为1个时钟周期（1拍）。
- ④ 流水线需要经过一定的通过时间才能稳定。
- ⑤ 流水技术适合于大量重复的时序过程。

6.1 流水线概述

6.1.1 流水线基本概念

6.1.2 流水线分类

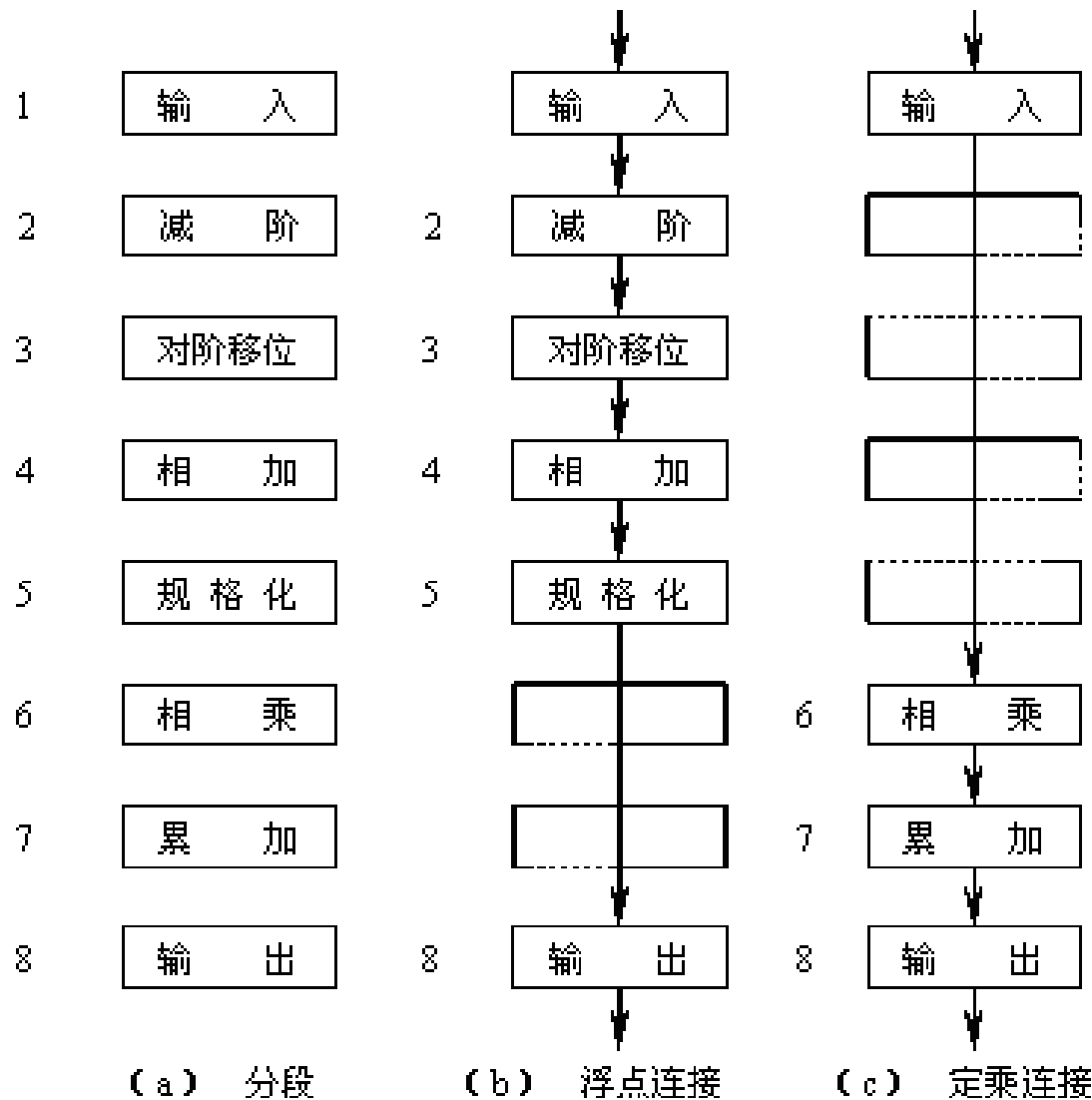
6.1.2 流水线的分类

1. 单功能流水线和多功能流水线

- 按流水线所完成的功能分类
- 单功能流水线，是指只能完成一种固定功能的流水线。
例如：功能单元流水线
- 多功能流水线，是指各段可以进行不同的连接，从而完成不同的功能。

例如：TI ASC多功能流水线

TI ASC的多功能流水线



6.1.2 流水线的分类

2. 静态流水线和动态流水线

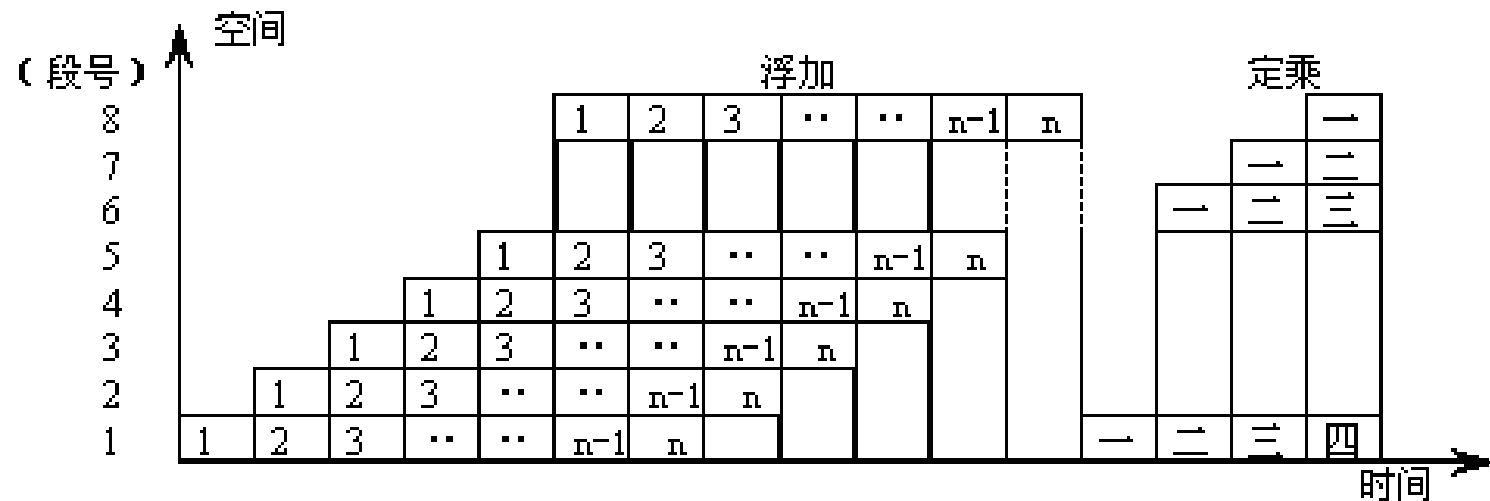
- 按同一时间内流水段的连接方式划分
- **静态流水线**，是指在同一时间内，流水线的各段只能按同一种功能的连接方式工作。

例如：TI ASC的流水线

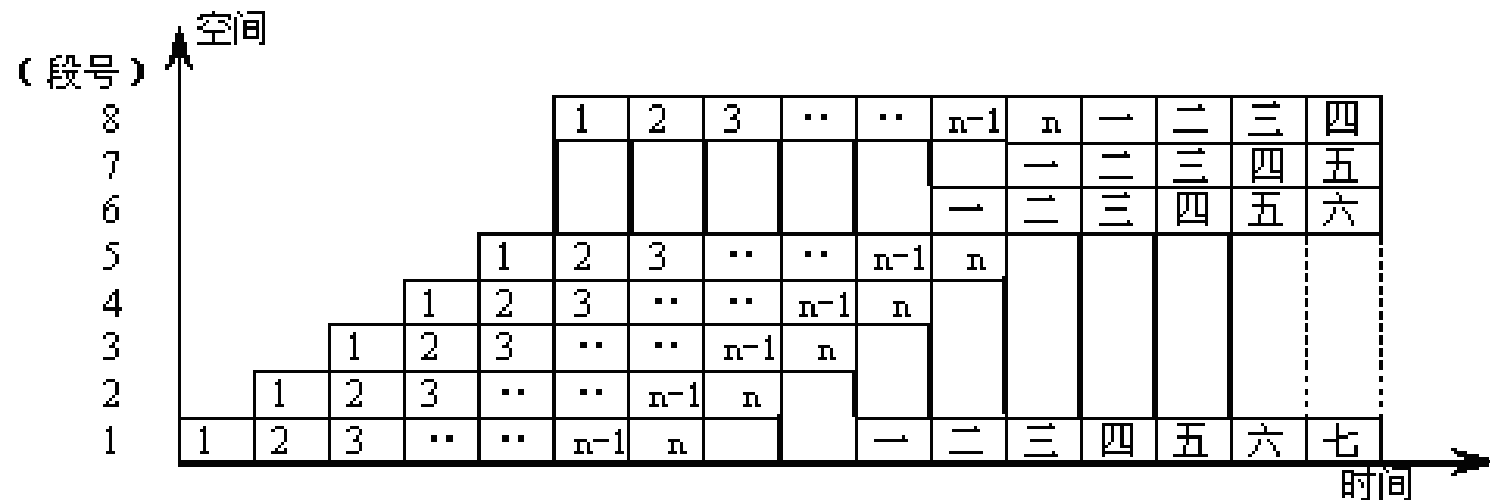
适合于处理一串相同的运算操作

- **动态流水线**，是指在同一时间内，当某些段正在实现某种运算时，另一些段却在实现另一种运算，会使流水线的控制变得很复杂

动、静态流水线时空图



(a) 静态流水线



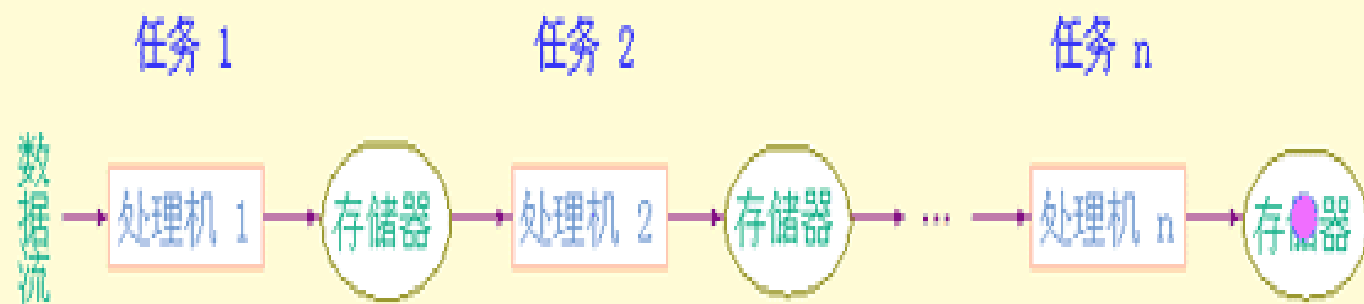
(b) 动态流水线

6.1.2 流水线的分类

3. 部件级、处理机级及处理机间流水线

- 按流水的级别划分
- 部件级流水线，又叫**运算操作流水线**，是把处理机的**算术逻辑部件分段**，使得各种数据类型的操作能够进行流水。
- 处理机级流水线，又叫**指令流水线**，是把**解释指令的过程按照流水方式处理**。
- 处理机间流水线，又叫**宏流水线**，是**由两个以上的处理机串行地对同一数据流进行处理，每个处理机完成一项任务**。

宏流水线



6.1.2 流水线的分类

4. 标量流水处理机和向量流水处理机

- 按照数据表示来进行分类
- **标量流水处理机**，是指处理机不具有向量数据表示，仅对标量数据进行流水处理。
例如：IBM 360/91，Amdahl 470V/6等
- **向量流水处理机**，是指处理机具有向量数据表示，并通过向量指令对向量的各元素进行处理。
例如：TI ASC、STAR-100、CYBER-205、CRAY-1、YH-1等

6.1.2 流水线的分类

5. 线性流水线和非线性流水线

- 按照是否有反馈回路来进行分类
- **线性流水线**是指流水线的**各段串行连接**，没有反馈回路。
- **非线性流水线**是指流水线中**除有串行连接的通路外，还有反馈回路**。

存在**流水线调度问题**。

确定什么时候向流水线引进新的输入，从而使新输入的数据和先前操作的反馈数据在流水线中不产生冲突，此即所谓**流水线调度问题**。

6.2 MIPS基本流水线

6.2.1 基本MIPS流水线

6.2.2 流水线性能分析

6.2.1 基本的MIPS流水线

1.一种简单的MIPS流水线

将5.4节中的[数据通路](#)流水化，

- ◆ 数据通路中的每一个周期就成为流水线的一段
- ◆ 每个时钟周期启动一条指令
——得到了一条简单的MIPS流水线。

简单MIPS流水线的流水过程：

- ◆ [时-空图](#)
- ◆ [按时间错开的数据通路](#)

6.2.1 基本的MIPS流水线

2.实现流水技术应解决的一些问题

(1) 应保证流水线各段不会在同一时钟周期内使用相同的通路资源。

- ◆ 例如，不能要求一个ALU既做有效地址计算，又做减法操作
- ◆ IF与Mem两个阶段都要访问存储器，怎样避免访存冲突？
- ◆ ID和WB两个阶段都要访问寄存器，是否存在冲突？怎样避免？

(2) PC计算问题

为了能够在每个时钟周期启动一条新的指令，流水线必须在IF段获得下一条指令的地址，并将其保存在PC中。

但是，分支指令会改变PC的值，而且只有在Mem段结束时，这个新值才会被写入PC，出现矛盾。

解决方法：

对于顺序执行，可以修改数据通路，在IF段完成PC计算。但分支指令如何处理？

(3) 合理划分流水段，每段内的操作都必须在一个时钟周期内完成。

(4) 流水线寄存器设计

- ◆ 为防止寄存器中的值在为流水线中某条指令所用时被流水线中其它的指令所重写，可在流水线各段之间设置流水线寄存器文件，也称锁存器。
- ◆ 流水线寄存器文件的命名
- ◆ 段A与B之间的流水线寄存器文件称为A/B
- ◆ 流水线寄存器的作用
- ◆ 流水线寄存器文件的构成

3.MIPS流水线的操作

在任一时刻，流水中的指令仅在流水线中的某一段内执行操作。

因此，只要知道每一流水段在各指令下进行何种操作，就知道了整个流水线的操作。

下表给出了MIPS流水线中每一段的操作

4.MIPS流水线中多路选择器的控制

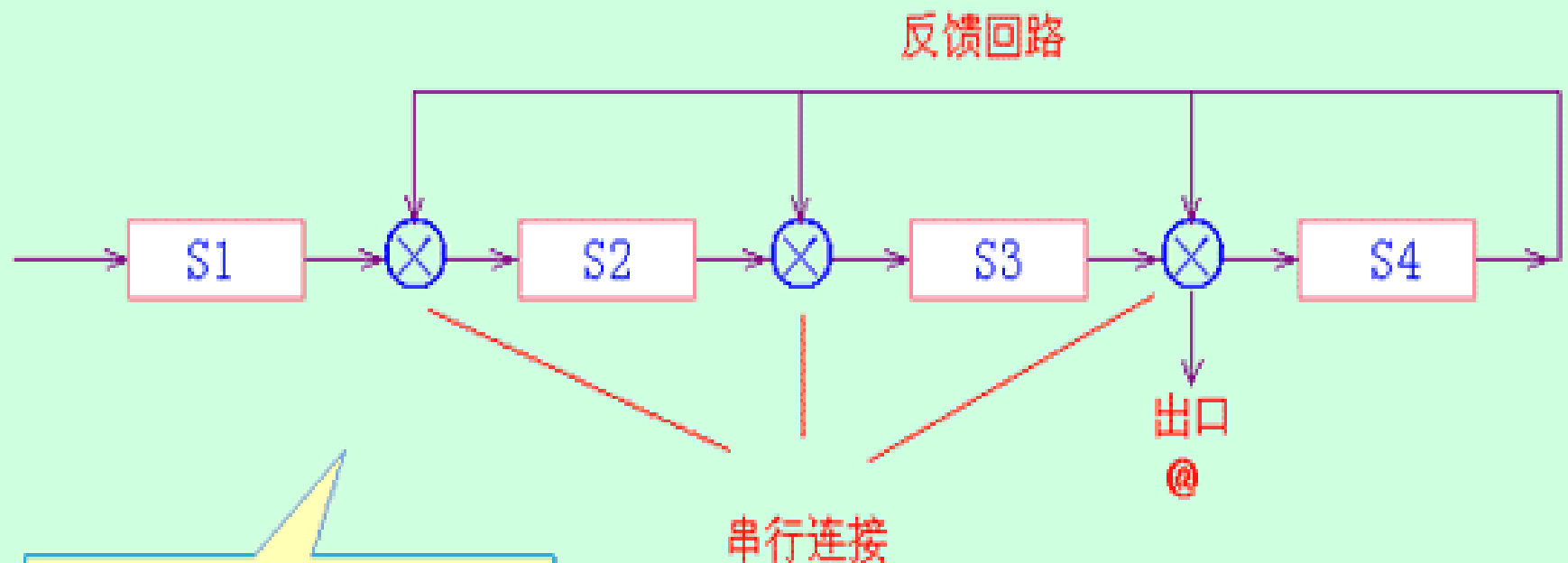
主要是确定[如何控制那四个多路选择器](#)：

- ◆ ALU输入端的两个MUX由ID/EX.IR所指出的指令类型控制
- ◆ IF段的MUX由EX/MEM.Cond域的值控制
- ◆ WB段的MUX由当前指令类型(Load/ALU)控制

非线性流水线



(举 例)



虽然流水线仅由四段构成，
但有些段可能要重复通过。

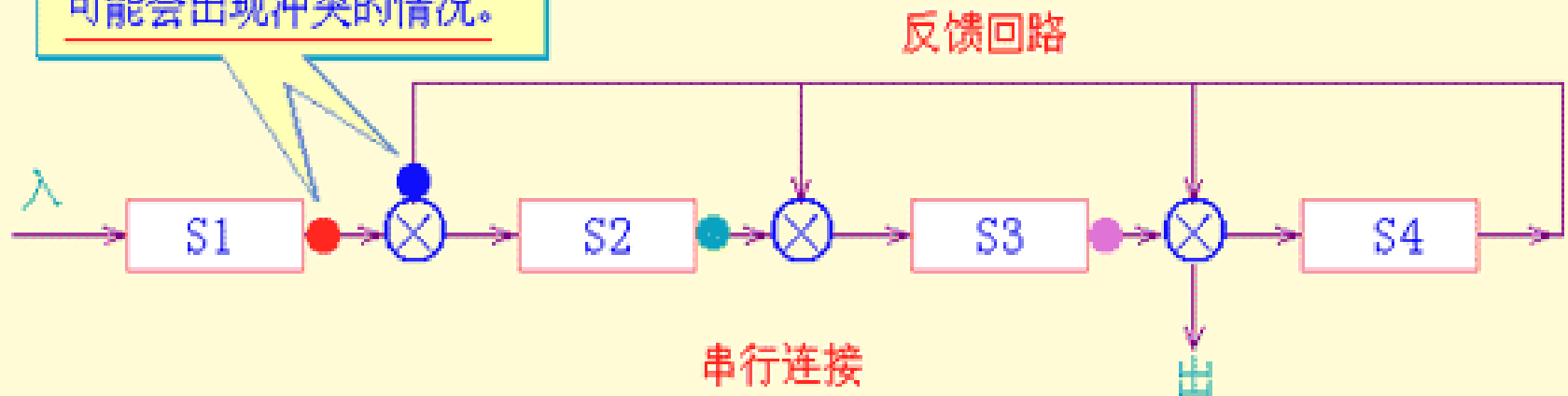
例如任务 @：

→ S1 → S2 → S3 → S4 → S2 → S3 → S4 → S3 →

流水线的调度问题



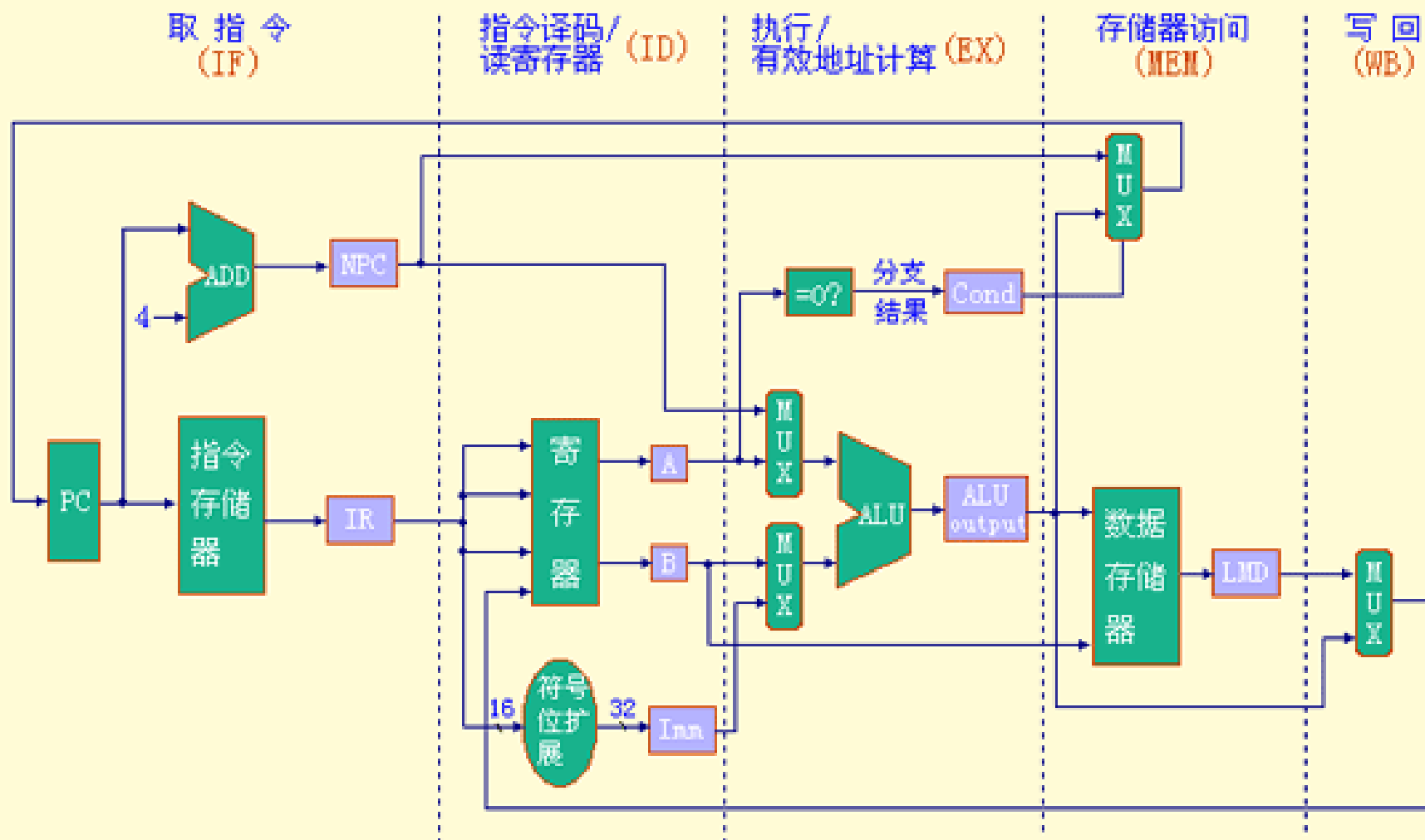
由于有反馈，流水线中可能会出现冲突的情况。



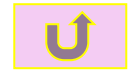
所以，

在非线性流水线中，一个重要的问题是确定什么时候向流水线引进新的输入，从而使新输入的数据和先前操作的反馈数据在流水线中不产生冲突。这就是所谓的流水线调度问题。

实现DLX指令的一种简单数据通路



一种简单的MIPS流水线



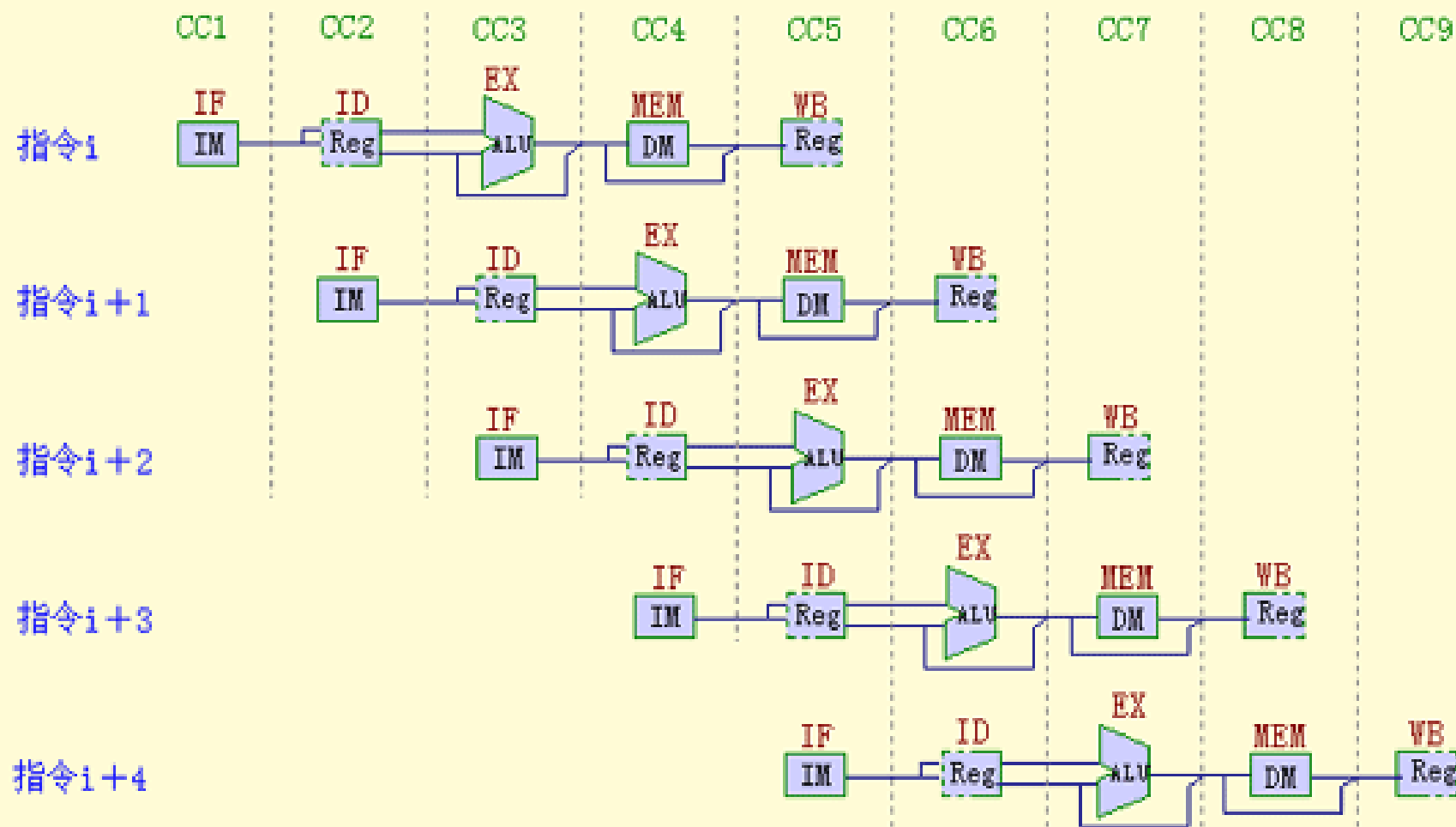
指令编号	时钟周期								
	1	2	3	4	5	6	7	8	9
指令i	IF	ID	EX	ME M	WB				
指令i+1		IF	ID	EX	ME M	WB			
指令i+2			IF	ID	EX	ME M	WB		
指令i+3				IF	ID	EX	ME M	WB	
指令i+4					IF	ID	EX	ME M	W B

流水线可以看成是按时间错开的数据通路序列

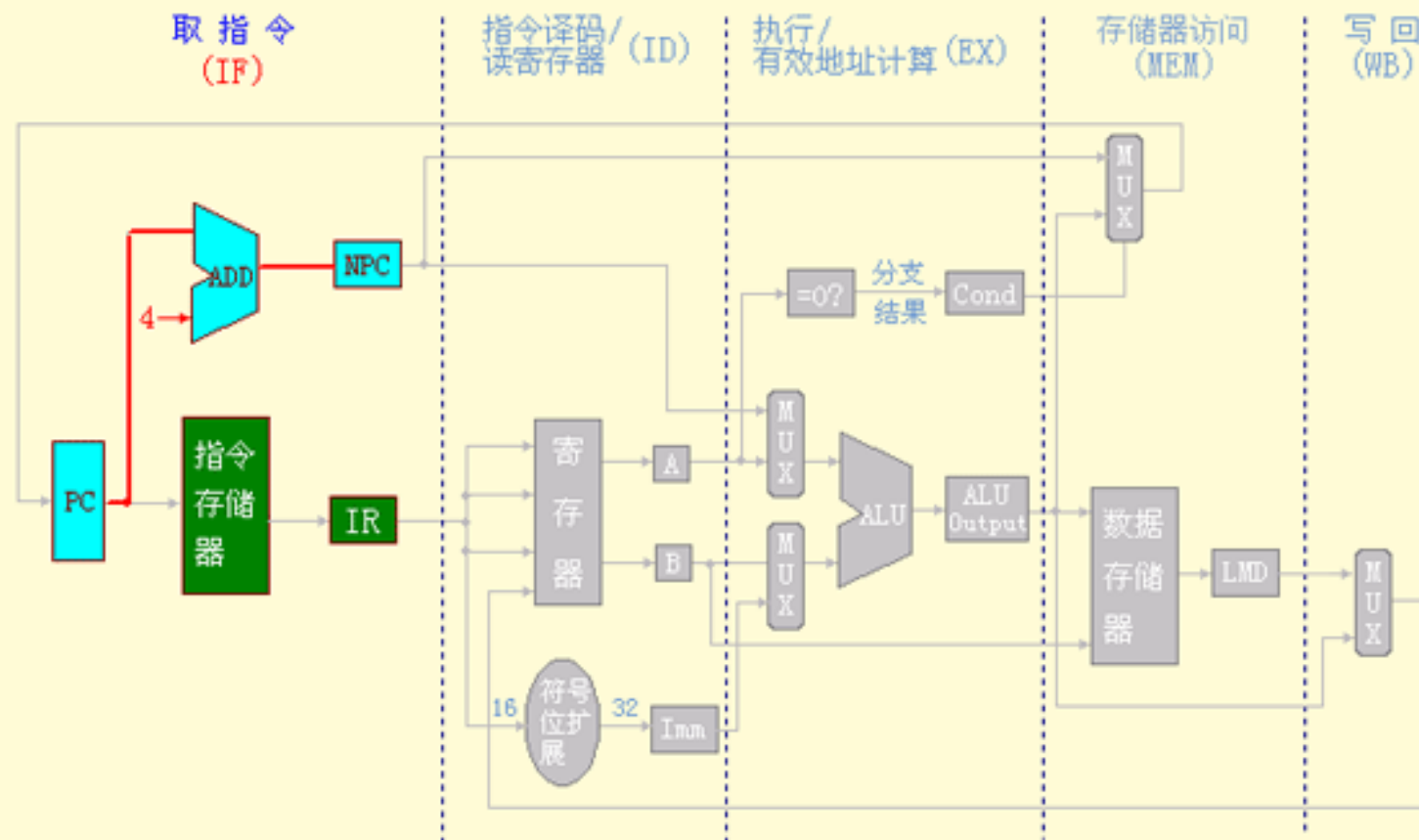


时间（时钟周期）

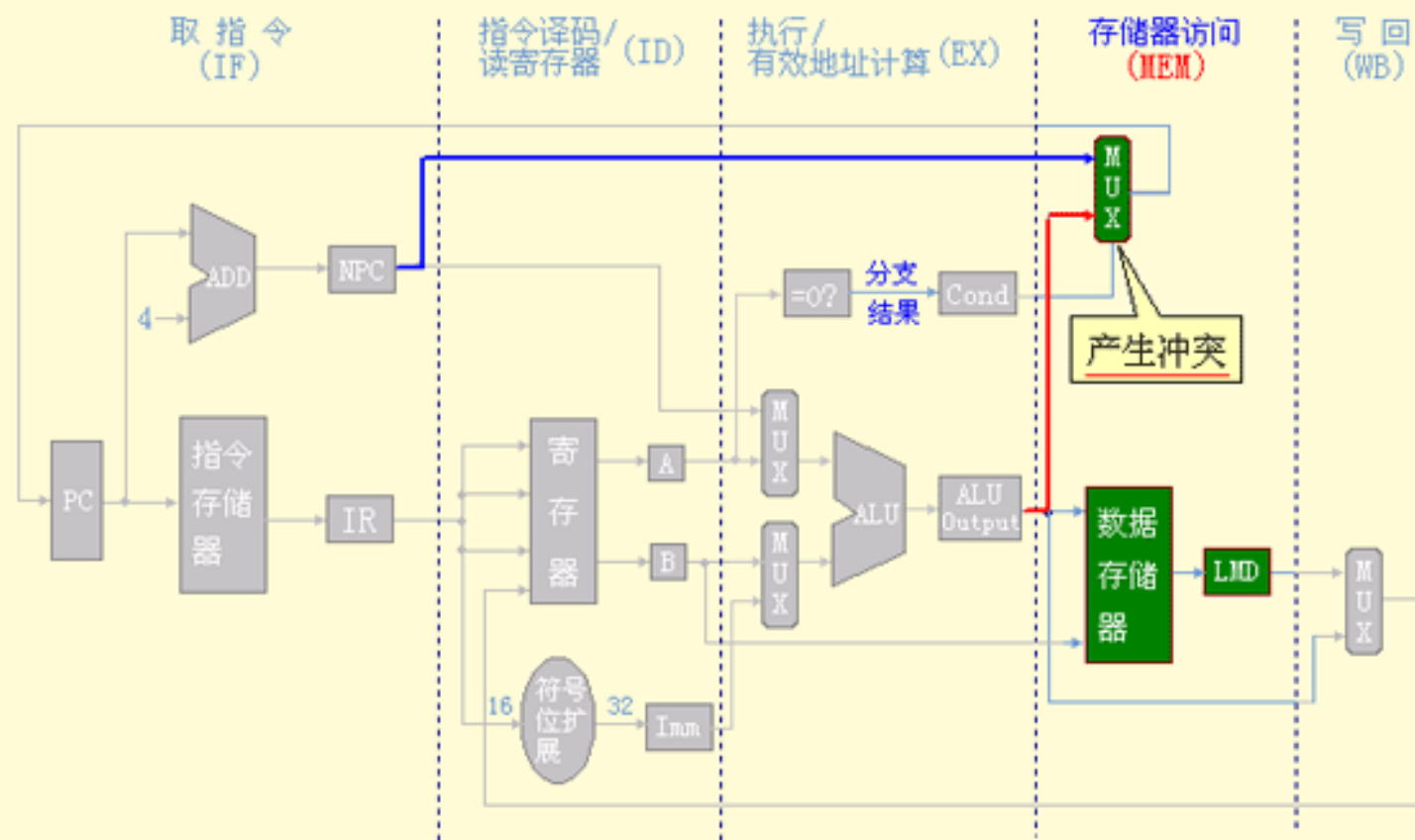
程序执行顺序（指令）

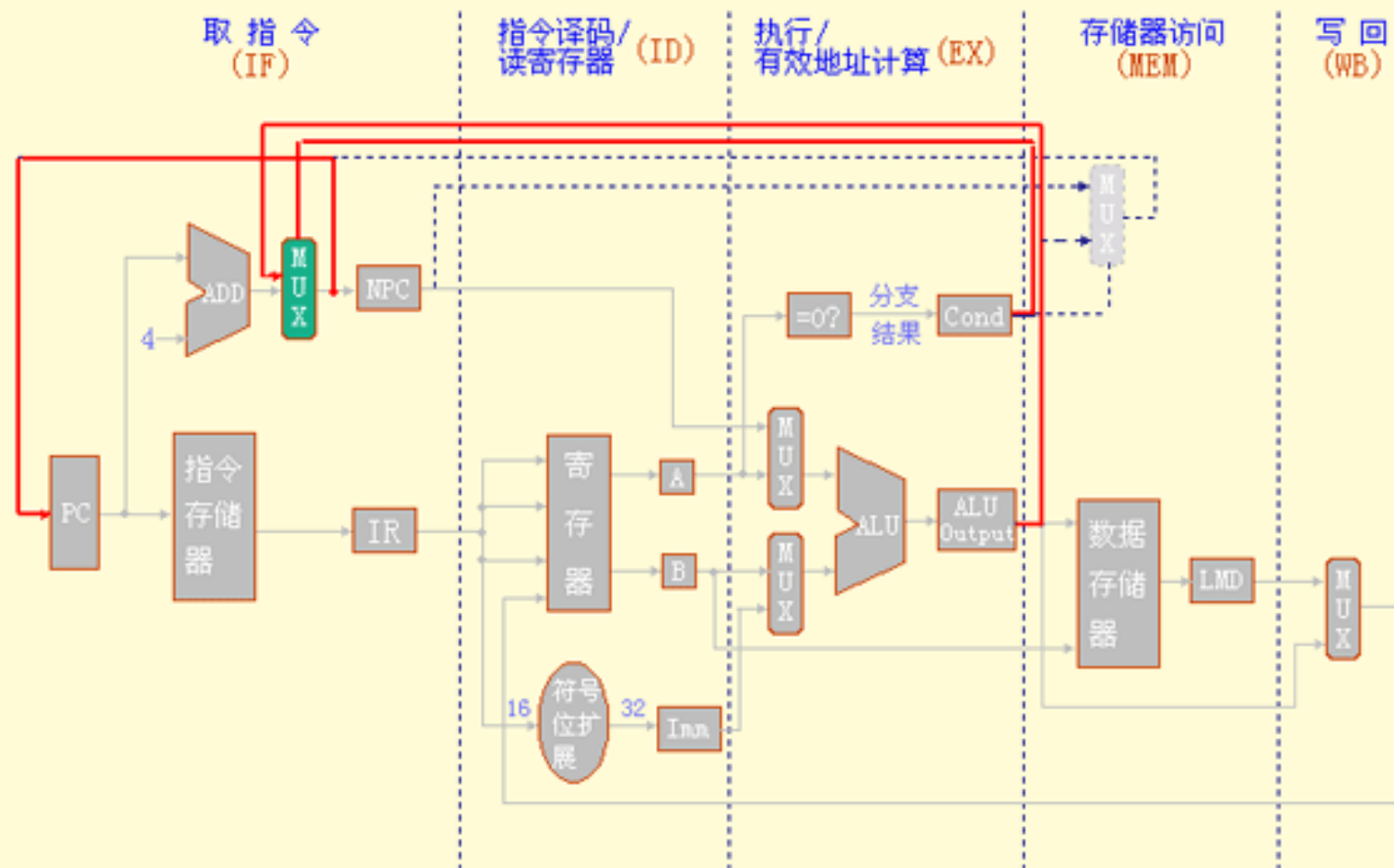


PC值顺序加4



分支指令改变PC值





DLX流水线的数据通路



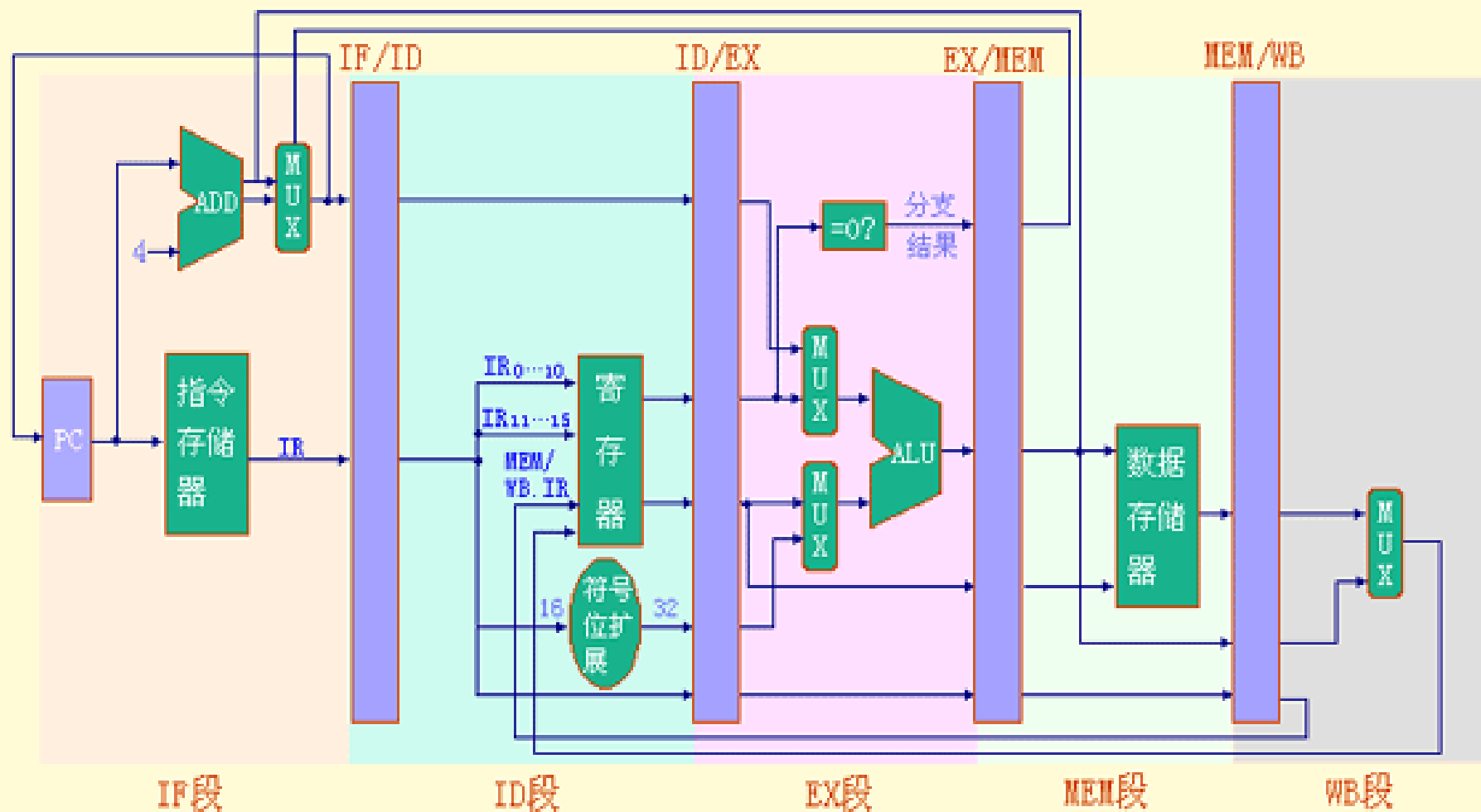
取指令
(IF)

指令译码/
读寄存器 (ID)

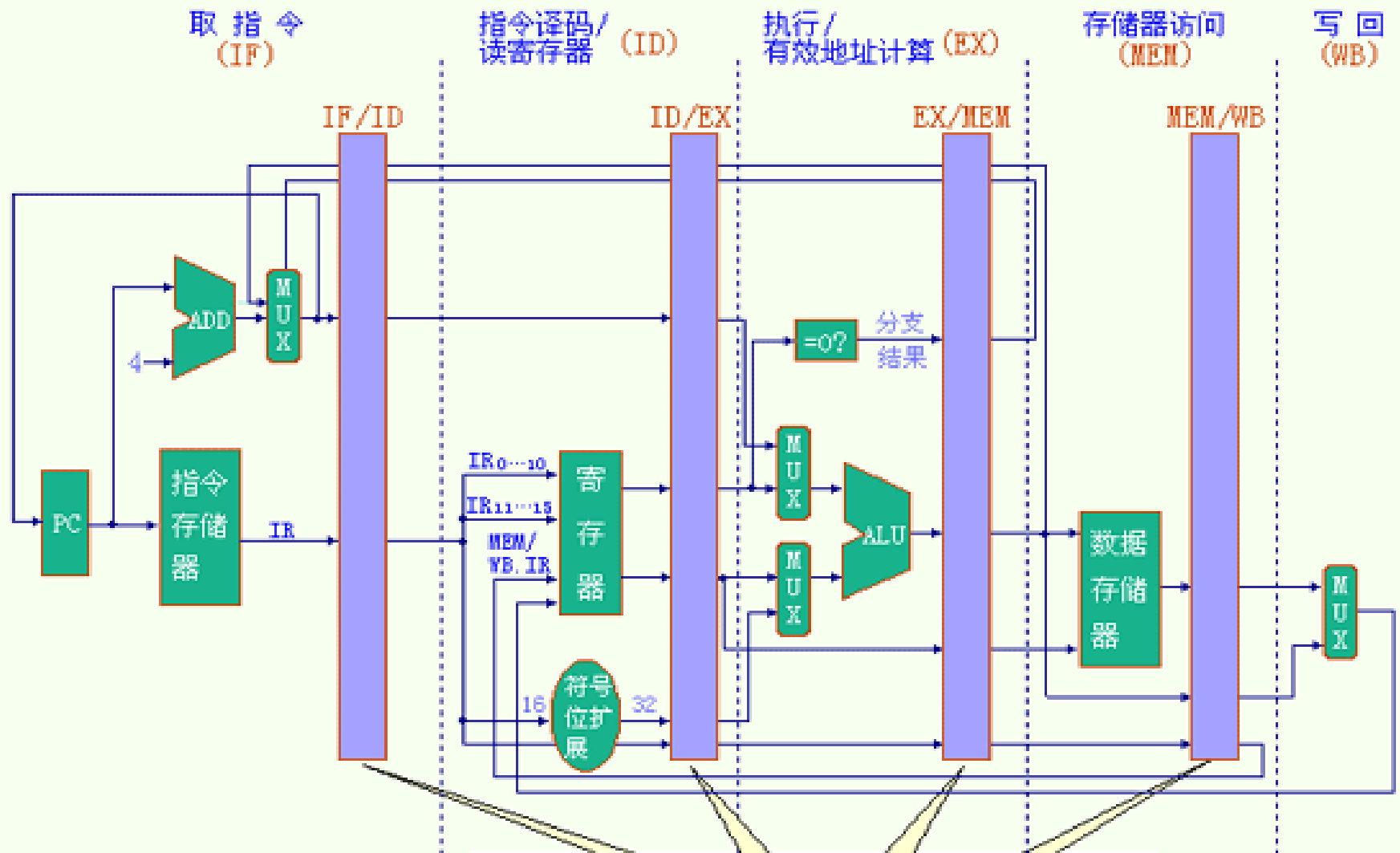
执行/
有效地址计算 (EX)

存储器访问
(MEM)

写回
(WB)

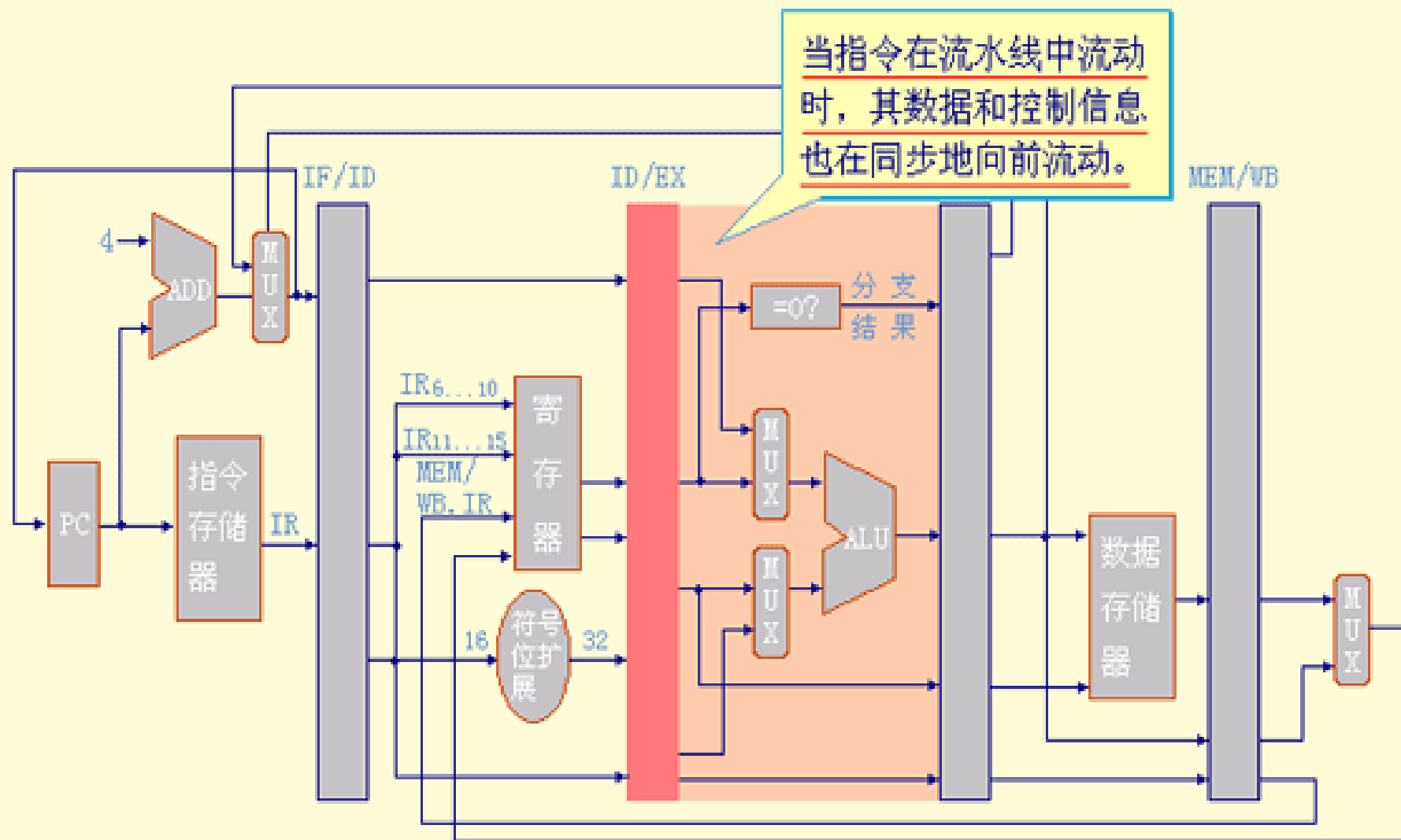


DLX流水线寄存器的命名

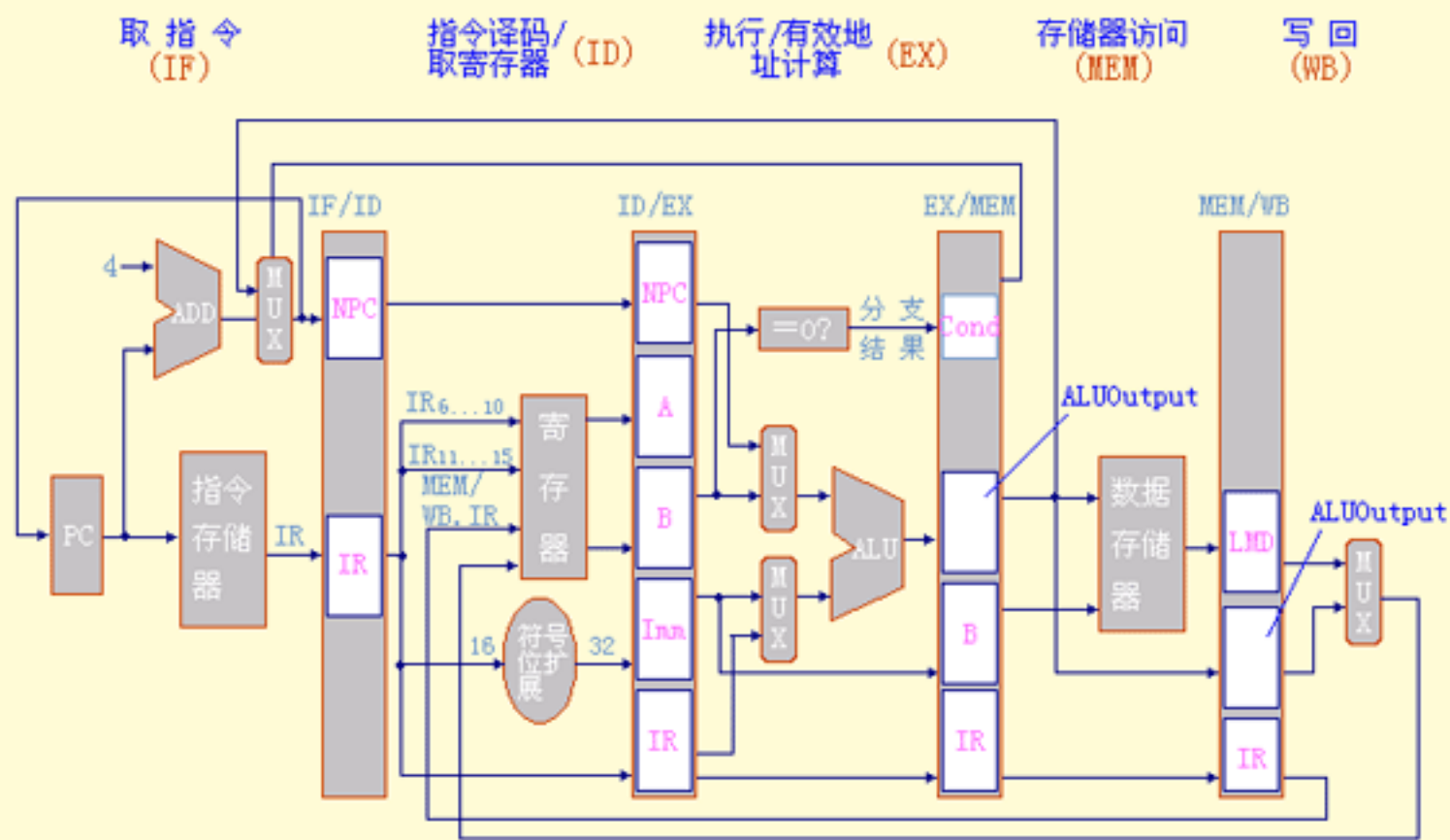


流水线寄存器组中所包含的寄存器的命名类似于域的命名。即采用：寄存器组 · 寄存器名

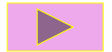
流水线寄存器的作用



流水线寄存器的构成



MIPS流水线的每个流水段的操作



流水段	任何指令类型		
IF	IF/ID. IR \leftarrow Mem[PC] IF/ID. NPC, PC \leftarrow (if EX/MEM. cond{EX/MEM. ALUOutput} else {PC+4});		
ID	ID/EX. A \leftarrow Regs[IF/ID. IR _{6...10}]; ID/EX. B \leftarrow Regs[IF/ID. IR _{11...15}]; ID/EX. NPC \leftarrow IF/ID. NPC; ID/EX. IR \leftarrow IF/ID. IR; ID/EX. Imm \leftarrow (IR ₁₆) ¹⁶ ##IR _{16...31} ;		
	ALU 指令	Load/Store 指令	分支指令
EX	EX/MEM. IR \leftarrow ID/EX. IR; EX/MEM. ALUOutput \leftarrow ID/EX. A op ID/EX. B 或 EX/MEM. ALUOutput \leftarrow ID/EX. A op ID/EX. Imm; EX/MEM. cond \leftarrow 0;	EX/MEM. IR \leftarrow ID/EX. IR; EX/MEM. B \leftarrow ID/EX. B EX/MEM. ALUOutput \leftarrow ID/EX. A + ID/EX. Imm; EX/MEM. cond \leftarrow 0;	EX/MEM. ALUOutput \leftarrow ID/EX. NPC + ID/EX. Imm; EX/MEM. cond \leftarrow (ID/EX. A op 0);

MIPS流水线的每个流水段的操作（续）



流水段	任何指令类型		
	ALU 指令	Load/Store 指令	分支指令
MEM	$\text{MEM/WB. IR} \leftarrow \text{EX/MEM. IR};$ $\text{MEM/WB. ALUOutput} \leftarrow \text{EX/MEM. ALUOutput};$	$\text{MEM/WB. IR} \leftarrow \text{EX/MEM. IR};$ $\text{MEM/WB. LMD} \leftarrow \text{Mem}[\text{EX/MEM. ALUOutput}];$ 或 $\text{Mem}[\text{EX/MEM. ALUOutput}] \leftarrow \text{EX/MEM. B};$	
WB	$\text{Regs}[\text{MEM/WB. IR}_{16 \dots 20}] \leftarrow \text{MEM/WB. ALUOutput};$ 或 $\text{Regs}[\text{MEM/WB. IR}_{11 \dots 15}] \leftarrow \text{MEM/WB. ALUOutput};$	$\text{Regs}[\text{MEM/WB. IR}_{11 \dots 15}] \leftarrow \text{MEM/WB. LMD};$	

DLX 流水线中 对多路寄存器MUX 的控制

