计 算 机 体 系 结 构

第八讲

计算机科学与技术学院

舒燕君

Recap

- 流水线的基本概念
  - ✓ 流水技术的定义、 特点
- 流水线的分类
  - ✓ 实现的功能（单功能、多功能）
  - ✓ 连接方式（静态、动态）
  - ✓ 子过程的粒度（部件级、处理机级和处理机间）
  - ✓ 数据表示（标量、向量）
  - ✓ 反馈回路（线性、非线性）
- **MIPS**的一种简单实现
- **MIPS**基本流水线
  - ✓ 将数据通路划分流水段，加入段间寄存器文件
  - ✓ 每功能段完成指令子过程的相应操作
  - ✓ 多路选择器的控制

3.1.1 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。

3.1.1 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”。

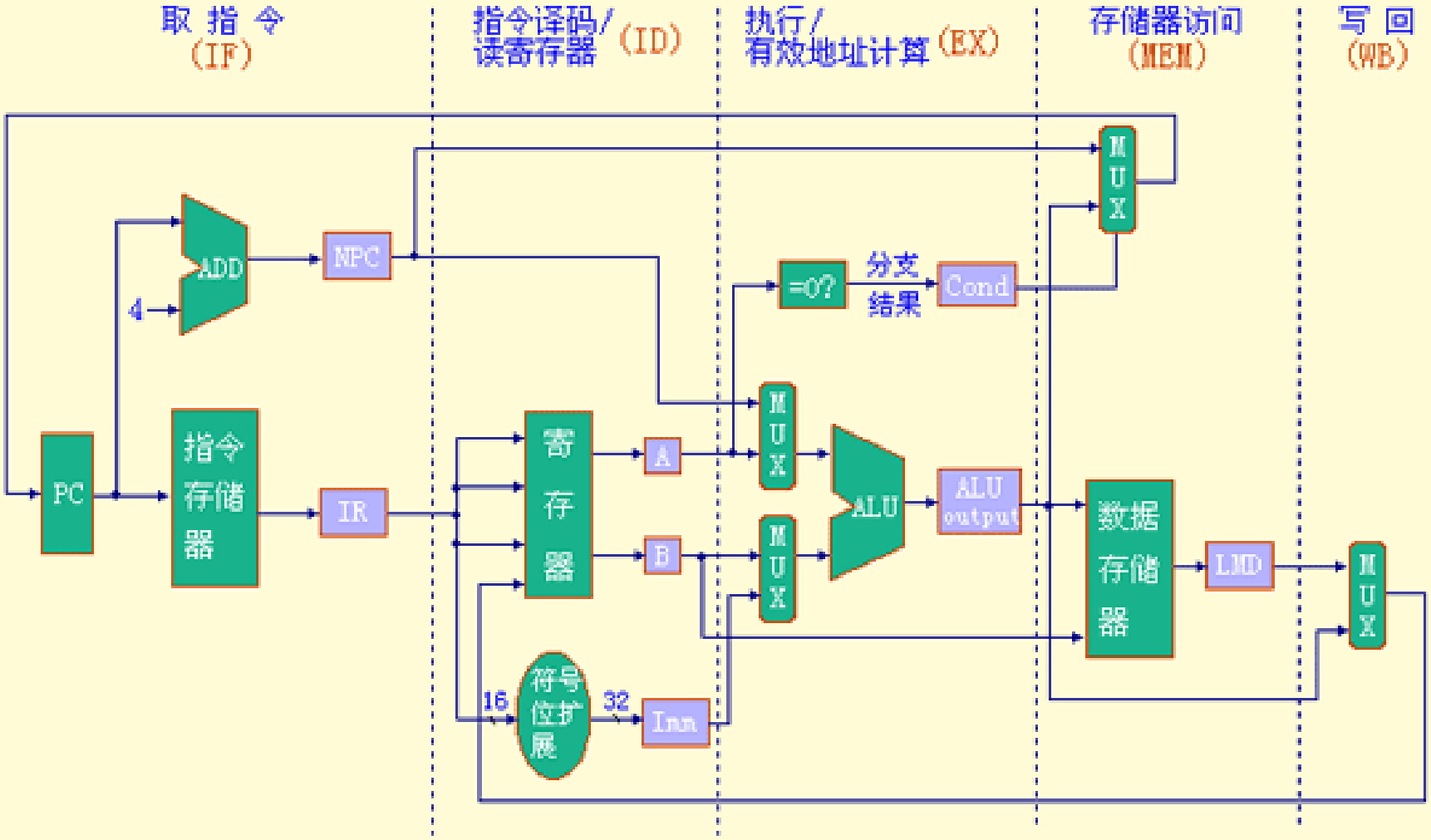
② 每个子过程由专用的功能段实现。

③ 各功能段的时间应基本相等，通常为1个时钟周期（1拍）。

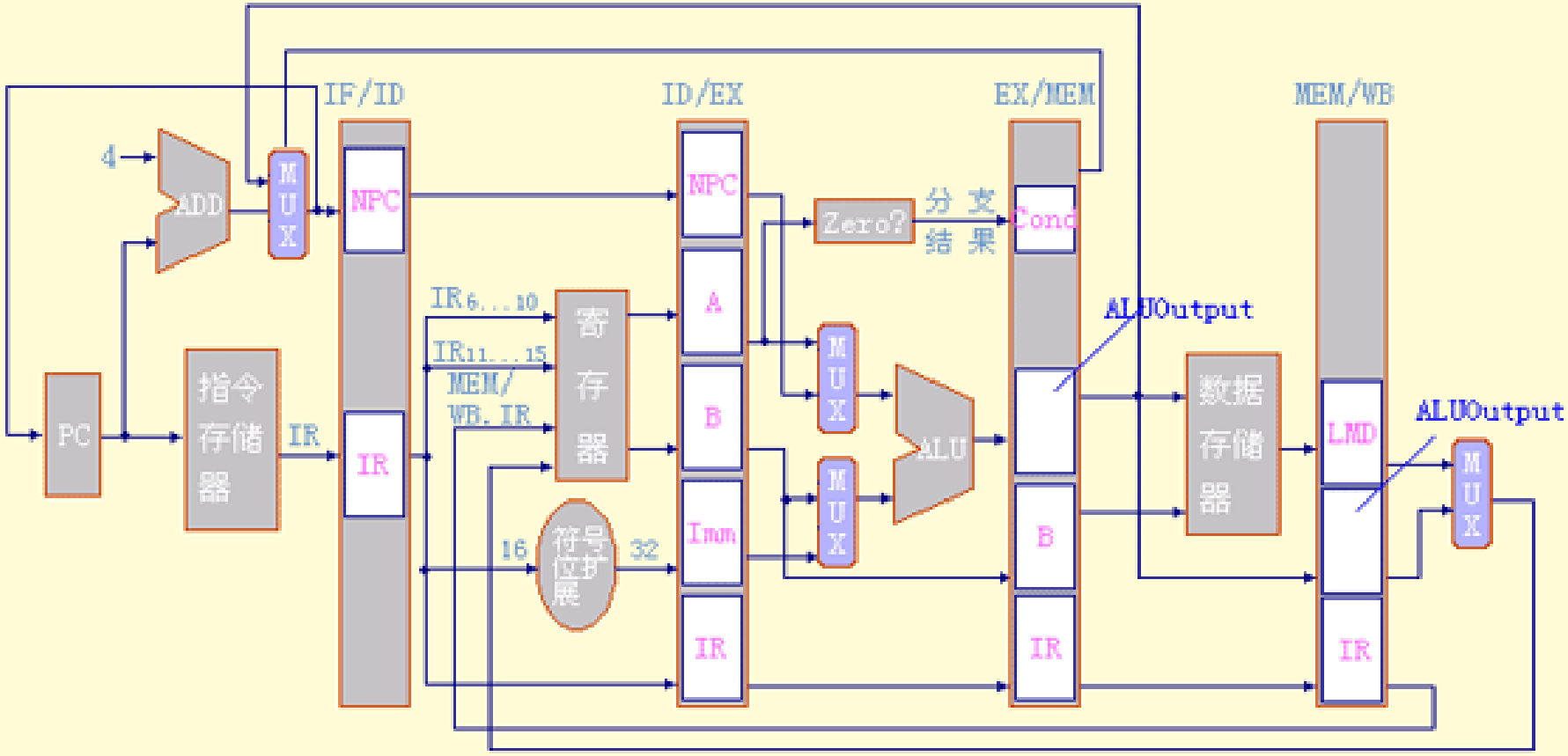
④ 流水线需要经过一定的通过时间才能稳定。

⑤ 流水技术适合于大量重复的时序过程。

实现DLX指令的一种简单数据通路



DLX 流水线中
对多路寄存器MUX 的控制



3.2 MIPS基本流水线

3.2.1 基本MIPS流水线

3.2.2 流水线性能分析

3.2.2 流水线性能分析

三项性能指标：吞吐率、加速比和效率

1.吞吐率

是衡量流水线速度的重要指标

- 吞吐率是指单位时间内流水线所完成的任务数或输出结果的数量。
- 最大吞吐率 TP_{max} 是指流水线在达到稳定状态后所得到的吞吐率。
- 设流水线由m段组成，完成n个任务的吞吐率称为实际吞吐率，记作 TP。

（1）最大吞吐率

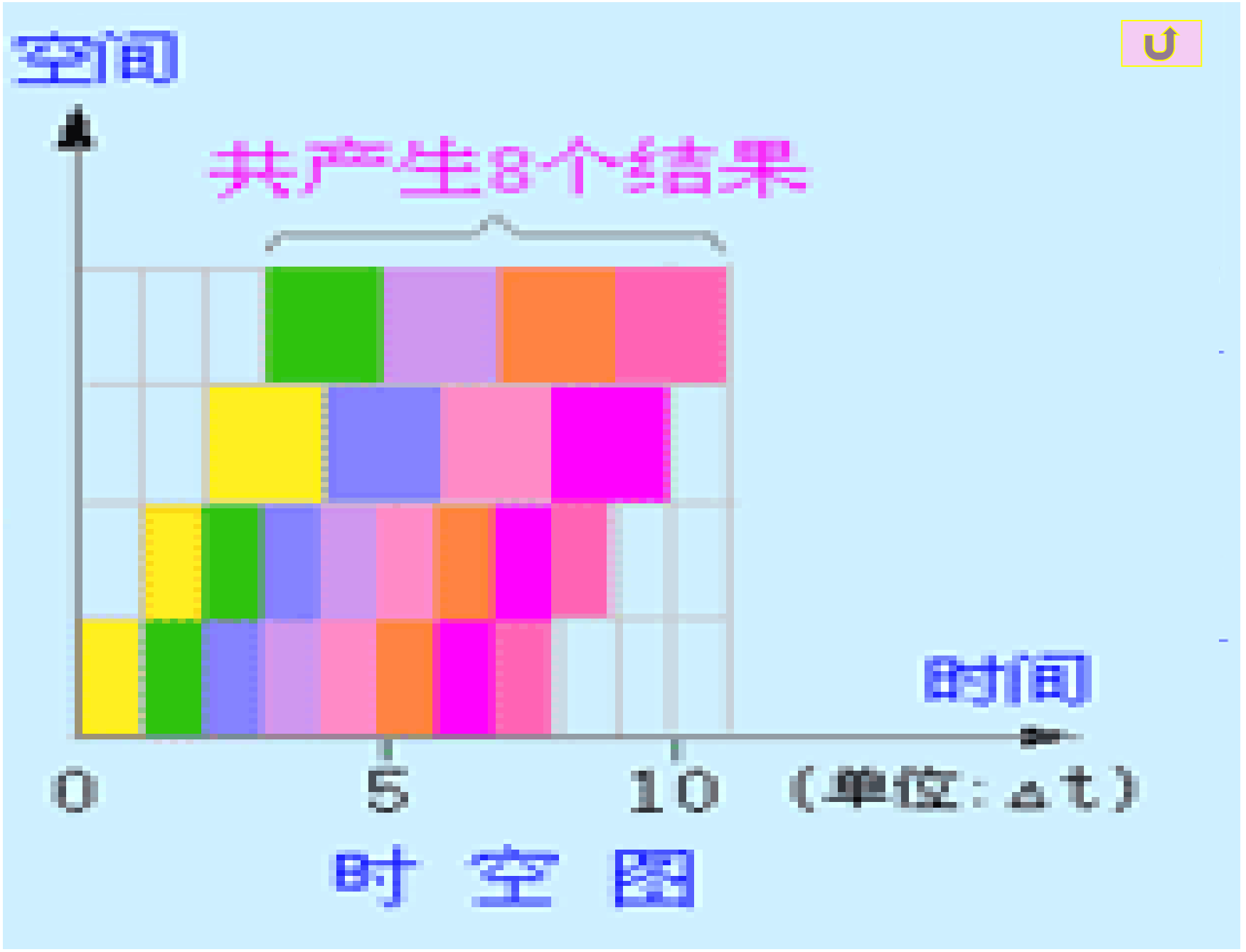
- 假设流水线各段的时间相等，均为Δt_0，则：

$$TP_{max} = 1/\Delta t_0$$

- 假设流水线各段时间不等，第i段时间为Δt_i，则：

$$TP_{max} = 1/max\{\Delta t_i\}$$

  - 最大吞吐率取决于流水线中最慢一段所需的时间，该段成为流水线的瓶颈
  - 消除瓶颈的方法
    - 细分瓶颈段
    - 重复设置瓶颈段



时 空 图

（2）实际吞吐率

- 若各段时间相等（假设均为Δt_0），则完成时间

$$T_{流水} = m\Delta t_0 + (n-1)\Delta t_0$$ （说明）

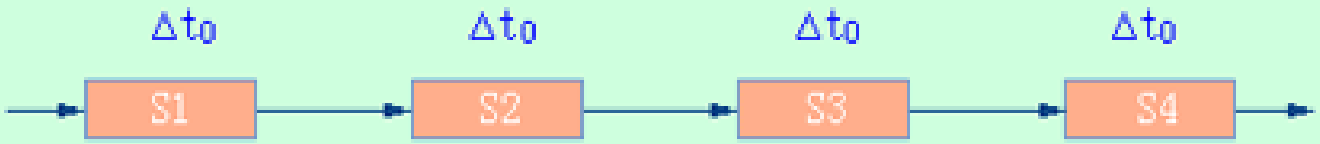
实际吞吐率 $$TP = \frac{n}{T_{流水}} = \frac{n}{m \cdot \Delta t_0 + (n-1) \cdot \Delta t_0}$$

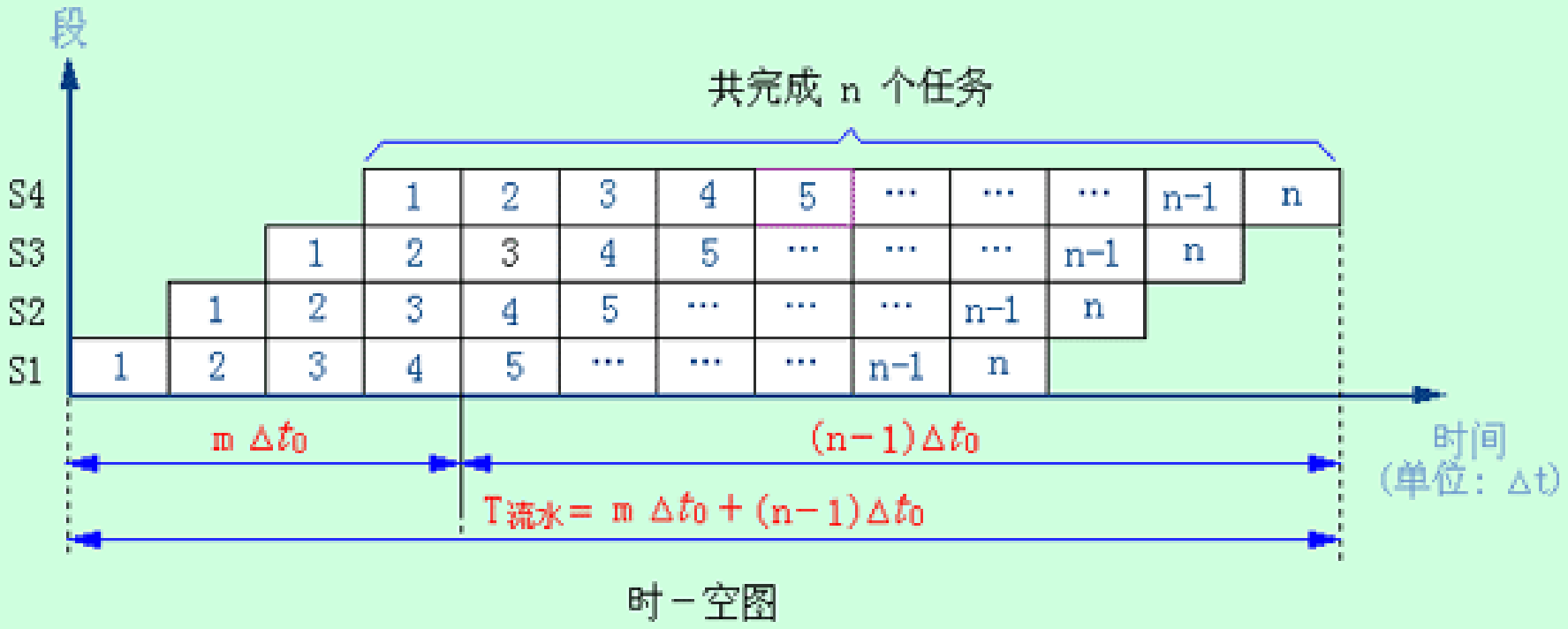
$$= \frac{1}{(1 + \frac{m-1}{n})\Delta t_0} = \frac{TP_{max}}{1 + \frac{m-1}{n}}$$

时空图

流水线的时-空图

（各段时间相等）



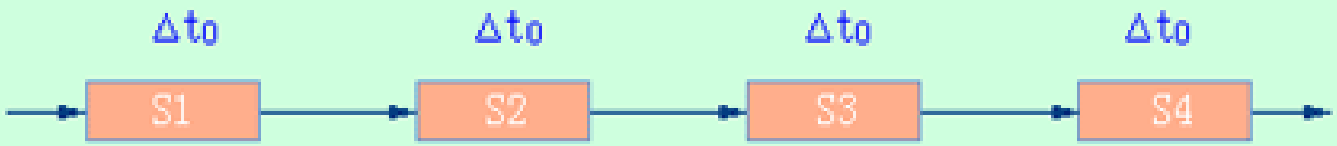


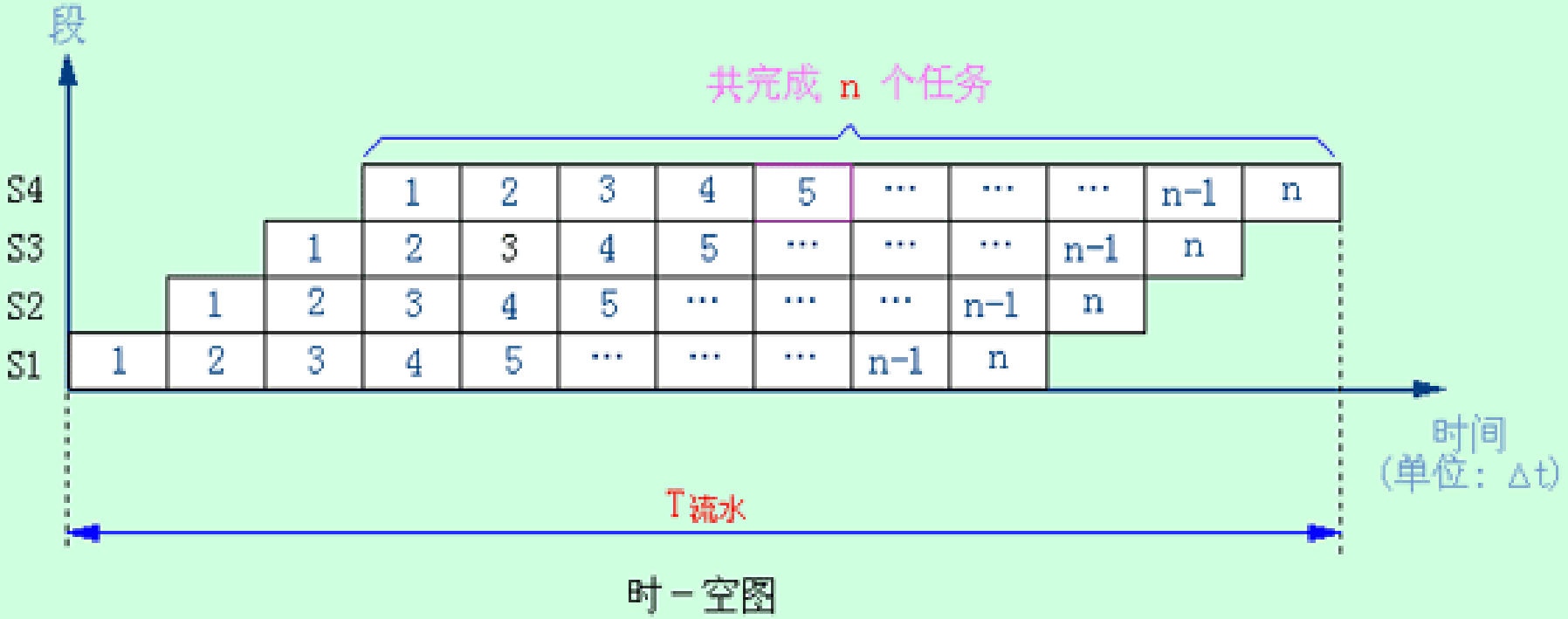
时－空图

$$T_{流水} = m\,\Delta t_0 + (n-1)\Delta t_0$$

流水线的时-空图

（各段时间相等）

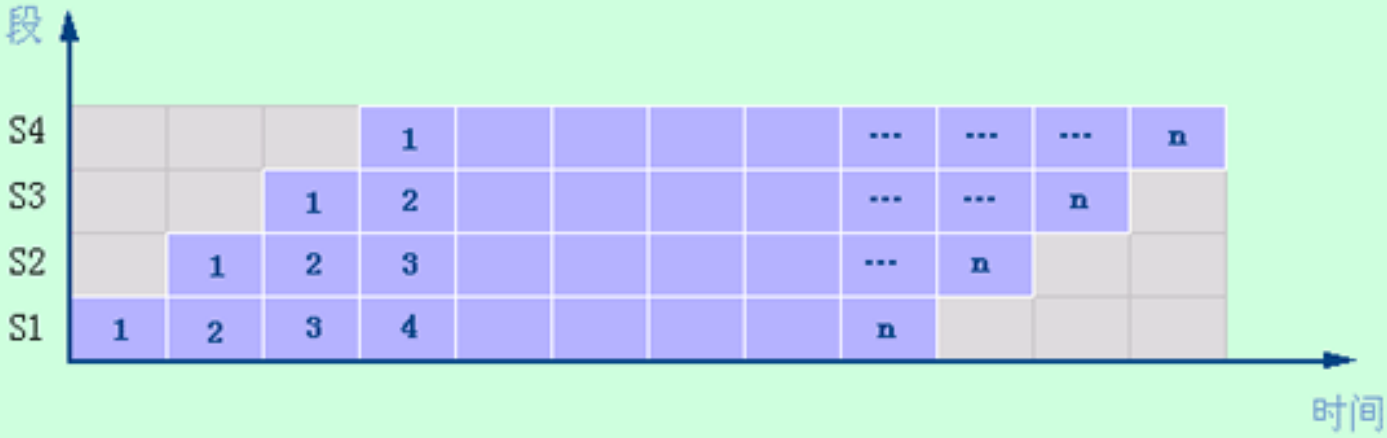




时－空图

吞吐率 $TP = \frac{n}{T_{流水}}$

流水段的效率



所以 $e_2 = \dfrac{n\Delta t_0}{T_{流水}}$

- 若各段时间不等（假设第i段为Δt_i），则完成时间

  时空图

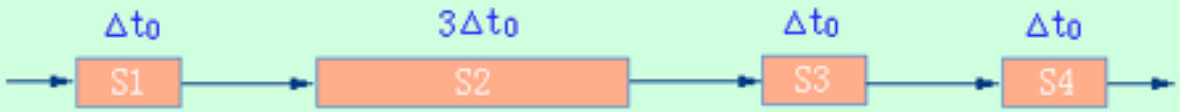
$$T = \sum_{i=1}^{m} \Delta t_i + (n-1)\Delta t_j$$

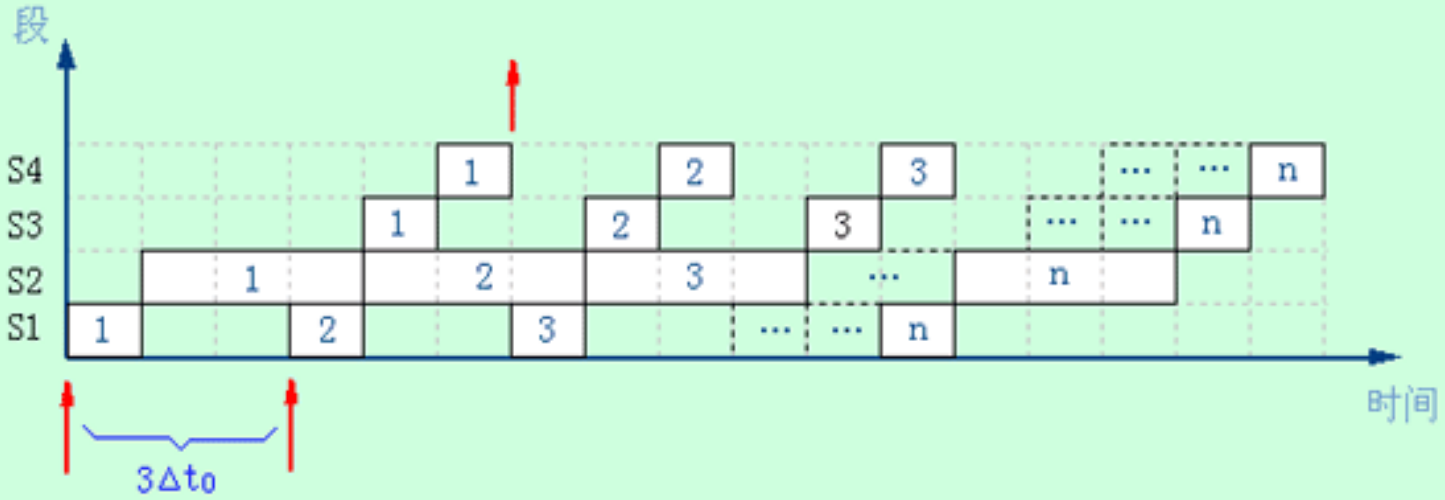
这里，$\Delta t_j = max\{\Delta t_i\}$

实际吞吐率 $$TP = \frac{n}{\sum_{i=1}^{m} \Delta t_i + (n-1)\Delta t_j}$$

流水线的时-空图

（各段时间不等）





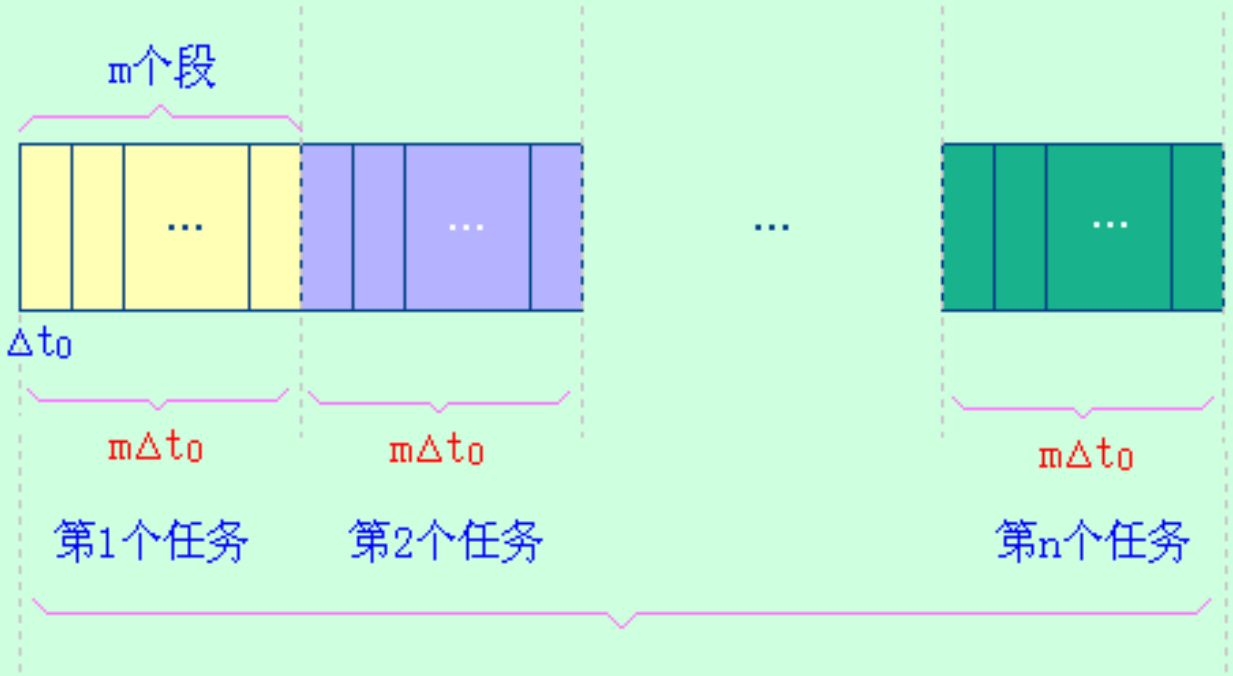
2. 加速比

- 加速比是指流水线速度与等功能的非流水线速度之比。
- 根据定义可知，加速比$S = T_{非流水} / T_{流水}$
- 若流水线为m段，每段时间均为Δt_0，则

$$T_{非流水} = nm\Delta t_0,\ T_{流水} = m\Delta t_0 + (n-1)\Delta t_0$$

$$S = \frac{mn}{m+n-1} = \frac{m}{1+\frac{m-1}{n}}$$

非流水方式所需的时间



总的时间为 $T_{非流水线} = n \times m\Delta t_0$

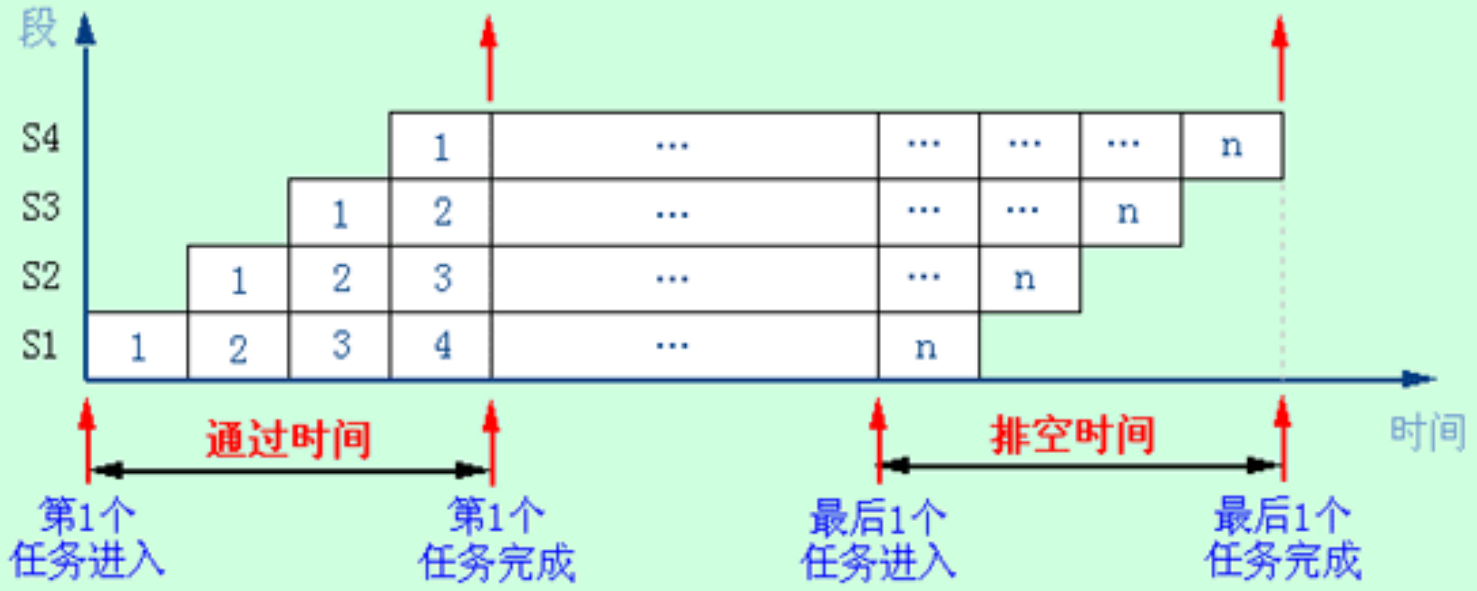
3.效率

- 效率指流水线的设备利用率。
- 由于流水线有通过时间和排空时间，所以流水线的各段并非一直满负荷工作，E<1
- 若各段时间相等，则各段效率也相等，即e1 = e2 = e3 =… = nΔt_0/$T_{流水}$ (解释)
- 整个流水线效率

$$E = \frac{n\Delta t_0}{T_{流水}} = \frac{n}{m+n-1} = \frac{1}{1+\frac{m-1}{n}}$$

当n>>m时，E≈1

通过时间和排空时间


段
S4
S3
S2
S1
1
2
3
4
…
n
通过时间
排空时间
时间
第1个
任务进入
第1个
任务完成
最后1个
任务进入
最后1个
任务完成

3.效率

- 从时-空图上看，效率就是n个任务所占的时空区与m个段总的时空区之比

- 根据这个定义，可以计算流水线各段时间不等时的流水线效率

$$E=\frac{n\text{ 个任务占用的时空区}}{m\text{ 个段总的时空区}}$$

4. 吞吐率、加速比和效率的关系

- $E = n\Delta t_0 / T_{流水} = mn\Delta t_0 / (T_{流水} m) = S/m$

  效率是实际加速比S与最大加速比m之比。

- $E = n\Delta t_0 / T_{流水} = (n/T_{流水}) \cdot \Delta t_0 = TP\Delta t_0$

  当Δt_0不变时，流水线的效率与吞吐率呈正比。为提高效率而采取的措施，也有助于提高吞吐率。

5. 流水线性能分析实例

例1　在静态流水线上计算$\sum_{i=1}^{m} A_i B_i$，问吞吐率、加速比、效率各是多少?

解:　(1) 确定适合流水处理的计算过程

(2) 画时-空图

(3) 性能计算

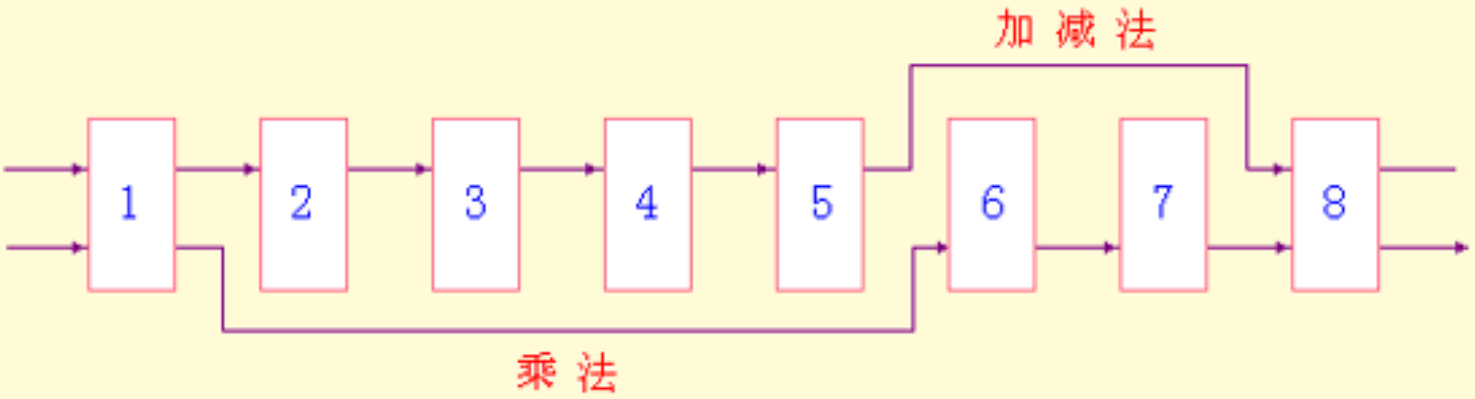
吞吐率 $TP = 7/20 \cdot \Delta t_0$

加速比 $S = 34 \cdot \Delta t_0 / 20 \cdot \Delta t_0 = 1.7$

效率 $E=(4\times4+3\times6)/(8\times20)=0.21$

静 态 流 水 线

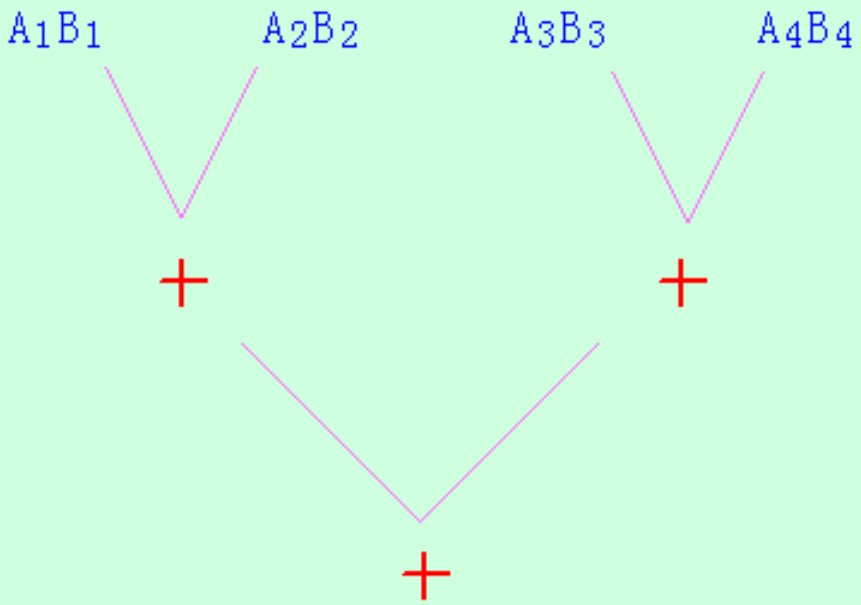
（举 例）



$\sum_{i=1}^{4} A_i B_i$ 的计算过程

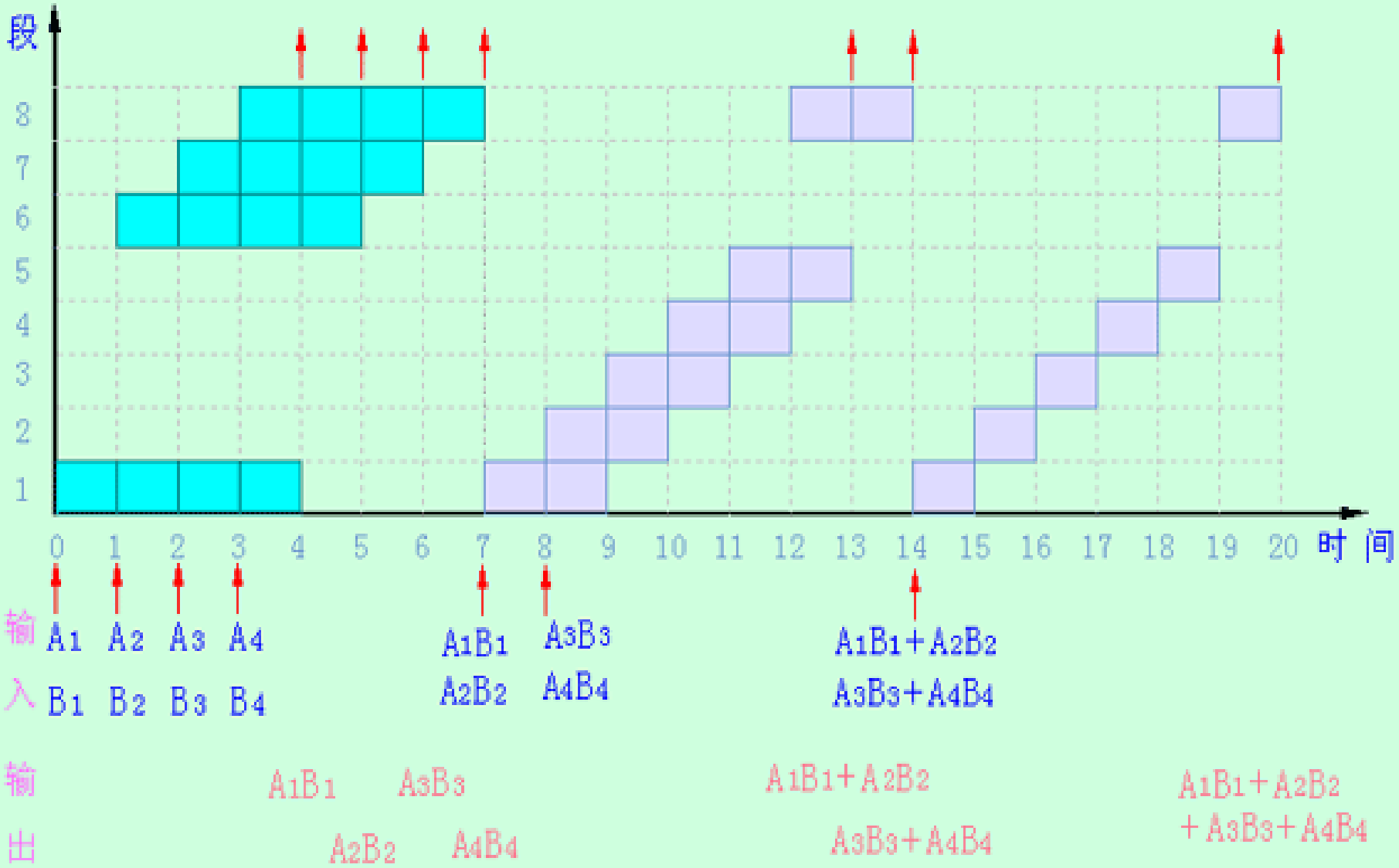
（例 3.1）

$$\sum_{i=1}^{4} A_i B_i = A_1B_1 + A_2B_2 + A_3B_3 + A_4B_4$$



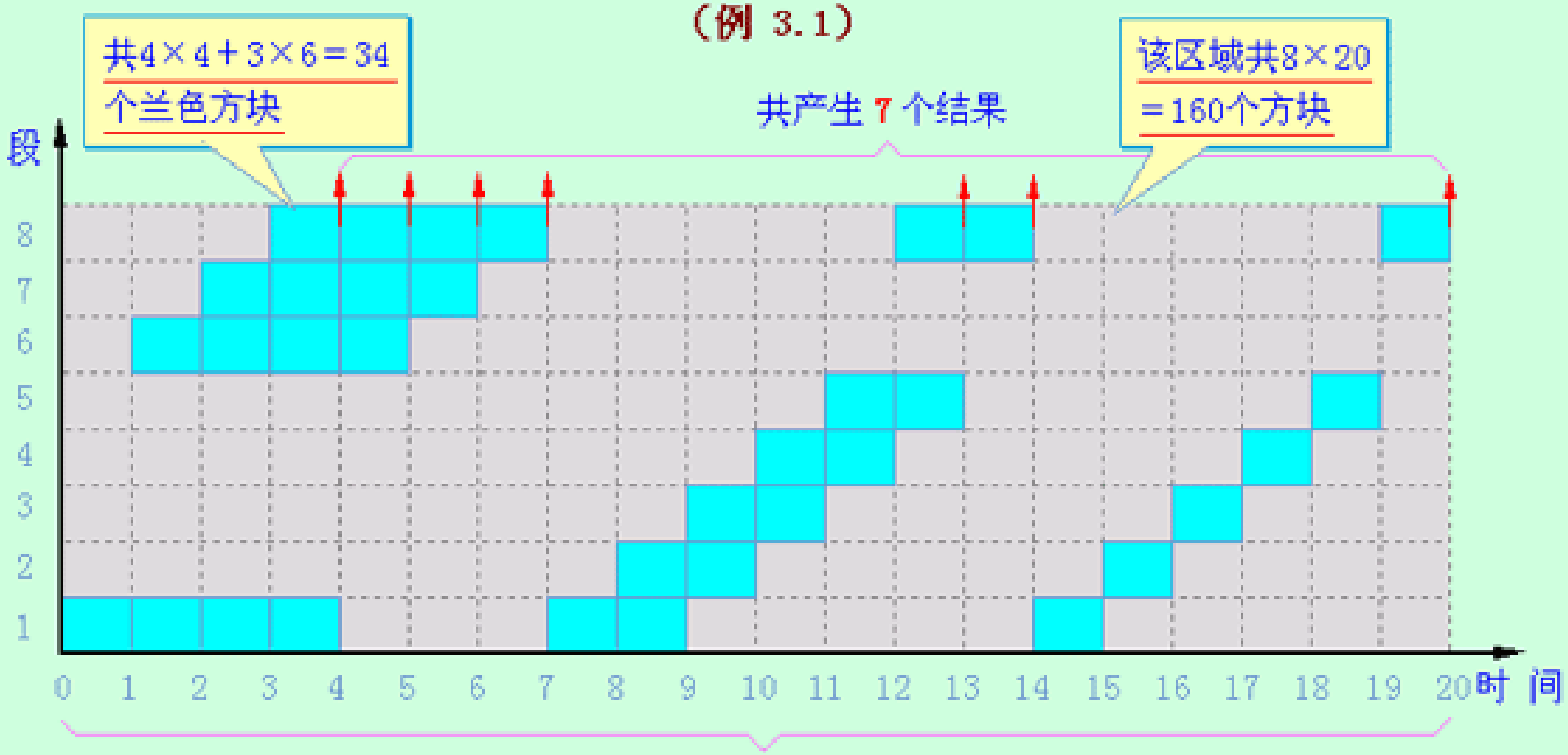
时 空 图

（例 3.1）


段
8
7
6
5
4
3
2
1
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
时 间
输入
A1 A2 A3 A4
B1 B2 B3 B4
A1B1
A2B2
A3B3
A4B4
A1B1+A2B2
A3B3+A4B4
输出
A1B1 A3B3
A2B2 A4B4
A1B1+A2B2
A3B3+A4B4
A1B1+A2B2
+A3B3+A4B4

性 能 计 算

（例 3.1）



处理时间 = $20\Delta t$

吞吐率 $TP = \frac{7}{20\Delta t}$

非流水处理所需时间 $= 4\times4\Delta t + 3\times6\Delta t = 34\Delta t$

加速比 $S = \frac{34\Delta t}{20\Delta t} = 1.7$

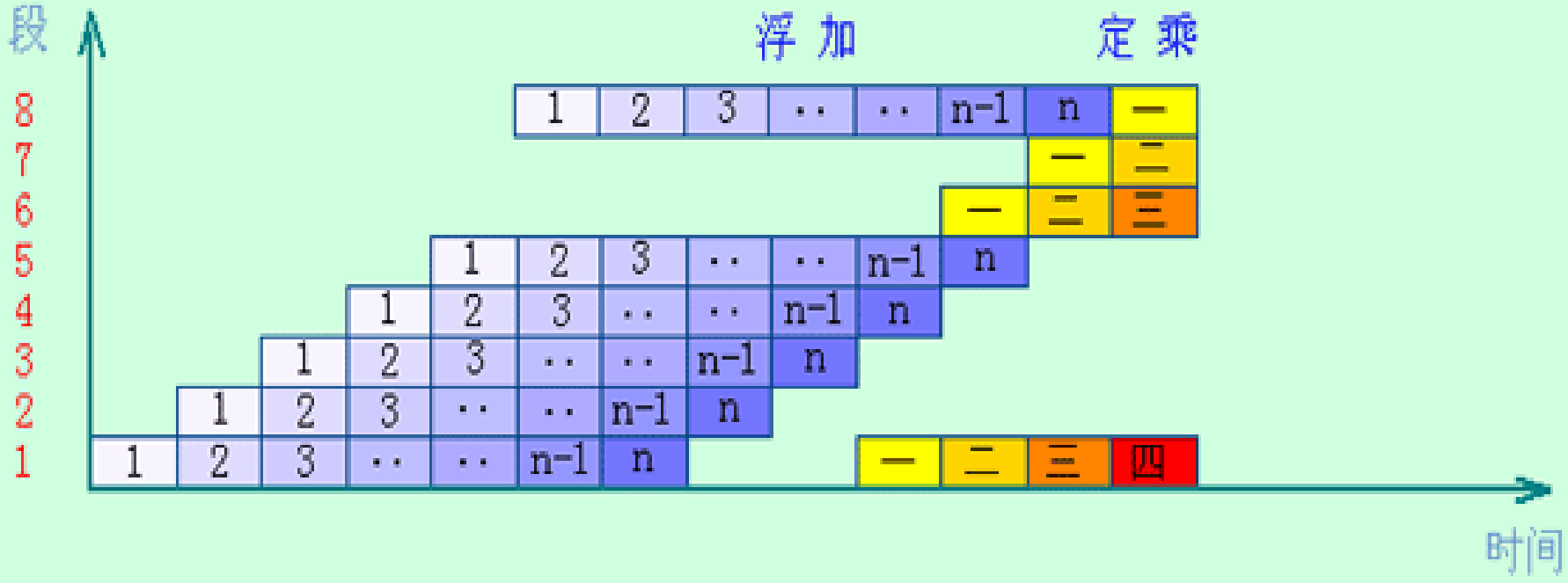
效率 $E = \frac{34}{160} = 0.21$

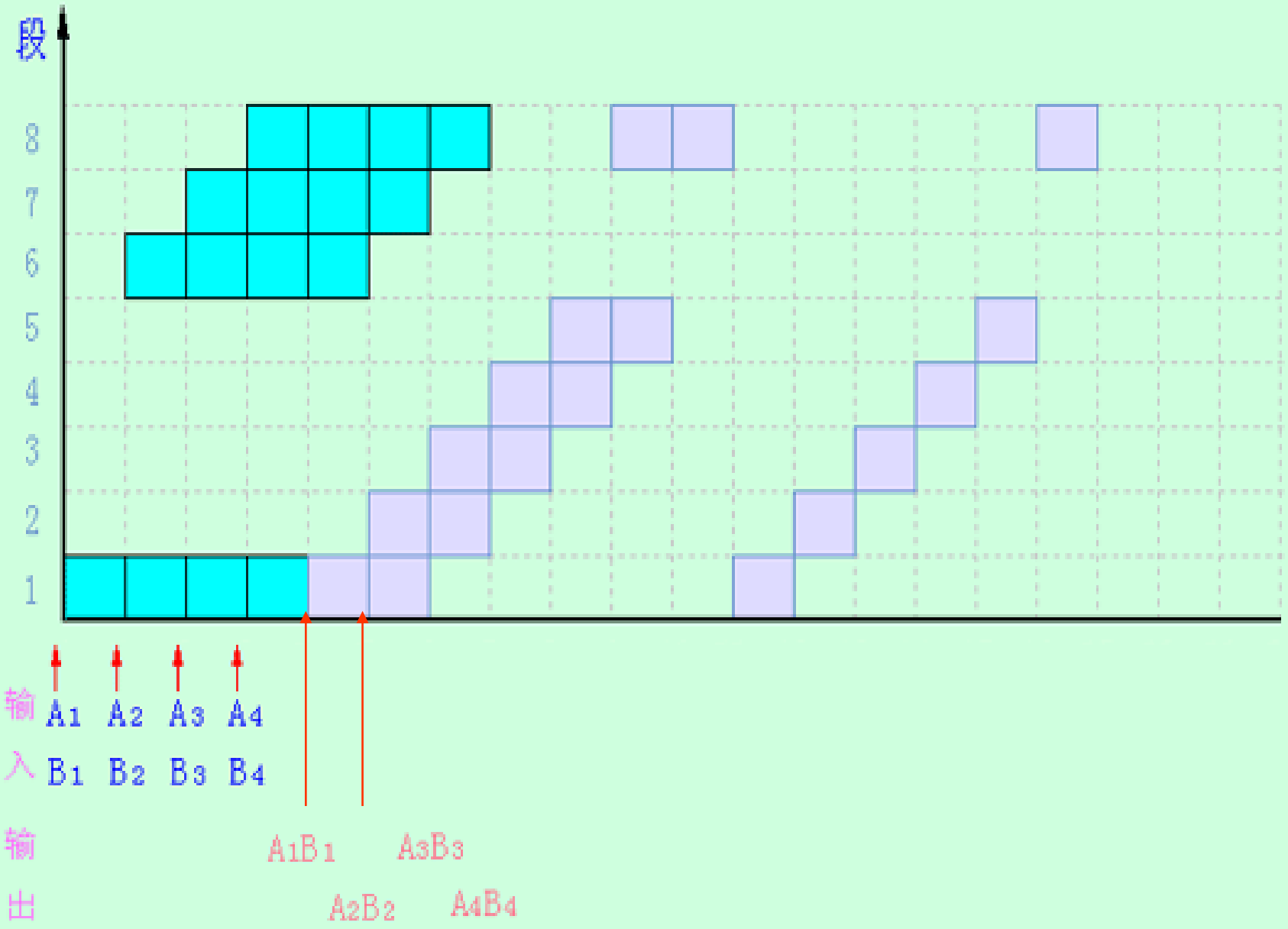
上述方案性能不高!

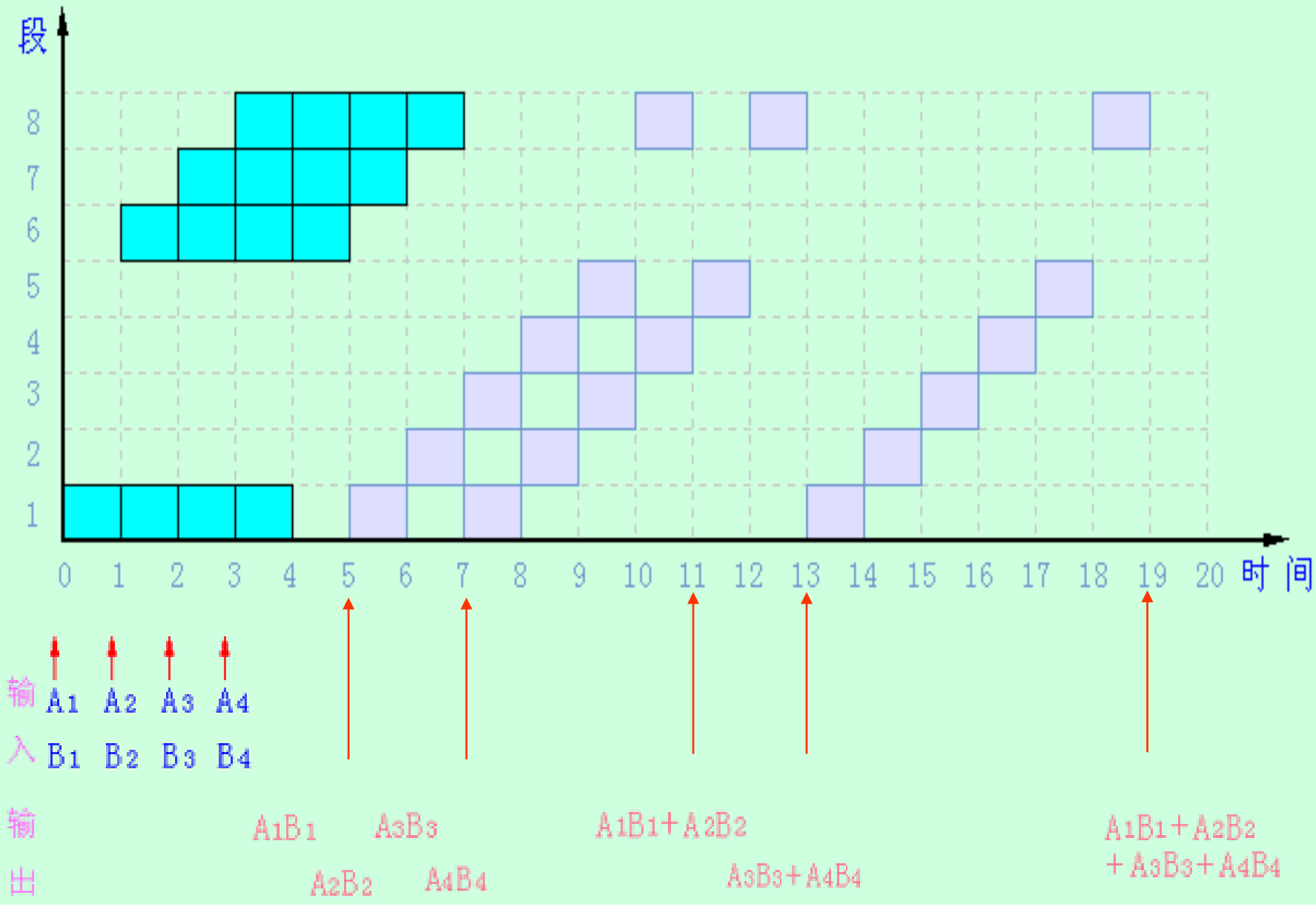
- 静态多功能流水线在对某种功能进行处理时，总有某些段处于空闲状态
- 功能切换增加了前一种功能的排空时间和后一种功能的通过时间
- 需要把输出回传到输入（冲突）

能否通过动态流水线改进其性能?

- 举例I
- 举例II　　这样行不行?　　正确方案


段
浮 加
定 乘
8
7
6
5
4
3
2
1
1 2 3 .. .. n-1 n
1 2 3 .. .. n-1 n
1 2 3 .. .. n-1 n
1 2 3 .. .. n-1 n
1 2 3 .. .. n-1 n
1 2 3 .. .. n-1 n
1 2 3 .. .. n-1 n
一 二 三 四
一
二
三
时间


段
8
7
6
5
4
3
2
1
输入
A1 A2 A3 A4
B1 B2 B3 B4
输出
A1B1 A3B3
A2B2 A4B4


段
8
7
6
5
4
3
2
1
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
时 间
输入
A1 A2 A3 A4
B1 B2 B3 B4
输出
A1B1
A3B3
A1B1+A2B2
A1B1+A2B2
+A3B3+A4B4
A2B2
A4B4
A3B3+A4B4

例2　在MIPS的非流水实现和基本流水线中，5个功能单元的执行时间：10/8/10/10/7ns。流水线额外开销为1ns，求相对于非流水指令实现而言，基本MIPS流水线的加速比是多少?

解：　$T_{非流水} = 10 + 8 + 10 + 10 + 7 = 45$

$T_{流水} = 10 + 1 = 11$

加速比$S = 45/11 \approx 4.1$

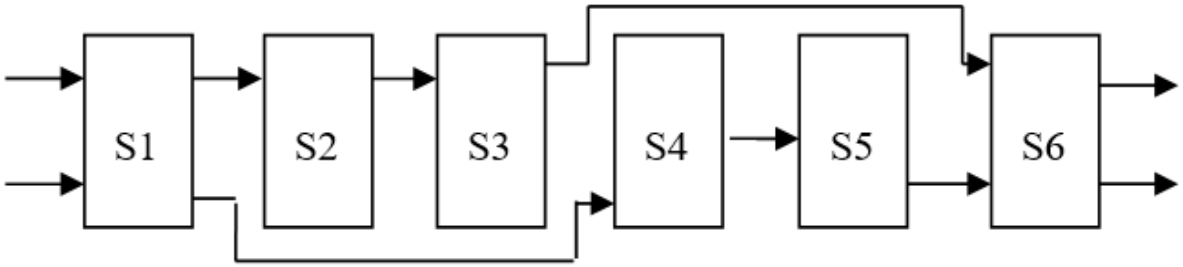
注：　流水线额外开销包括：流水寄存器的延迟（建立时间和传输延迟）以及时钟扭曲

6. 有关流水线性能的若干问题

- 流水线并不能减少（而且一般是增加）单条指令的执行时间，但能够提高吞吐率
- 增加流水线的深度可以提高流水线性能
- 流水线深度受限于流水线的延迟和额外开销
- 需要用高速锁存器作为流水线寄存器
  - Earle锁存器
- 指令之间存在的相关，产生了流水线的冲突，进而限制了流水线的性能

Quiz 2

1. 有假设有一条动态多功能流水线由6个功能段组成，如下图所示



其中，S1、S4、S5、S6组成乘法流水线，S1、S2、S3、S6组成加法流水线，各个功能段时间均为100 ns，假设该流水线的输出结果可以直接返回输入端，而且设置有足够的缓冲寄存器。若用该流水线以最快的方式计算以下算式：

$$\sum_{i=1}^{5} x_i y_i z_i$$

（1）请分析完成该计算的整个过程，画出完成该计算式的时空图

（2）请计算实际的吞吐率、加速比和效率。

注意：写上学号-姓名、自觉自主完成

第3章 流水线技术

3.1 流水线概述

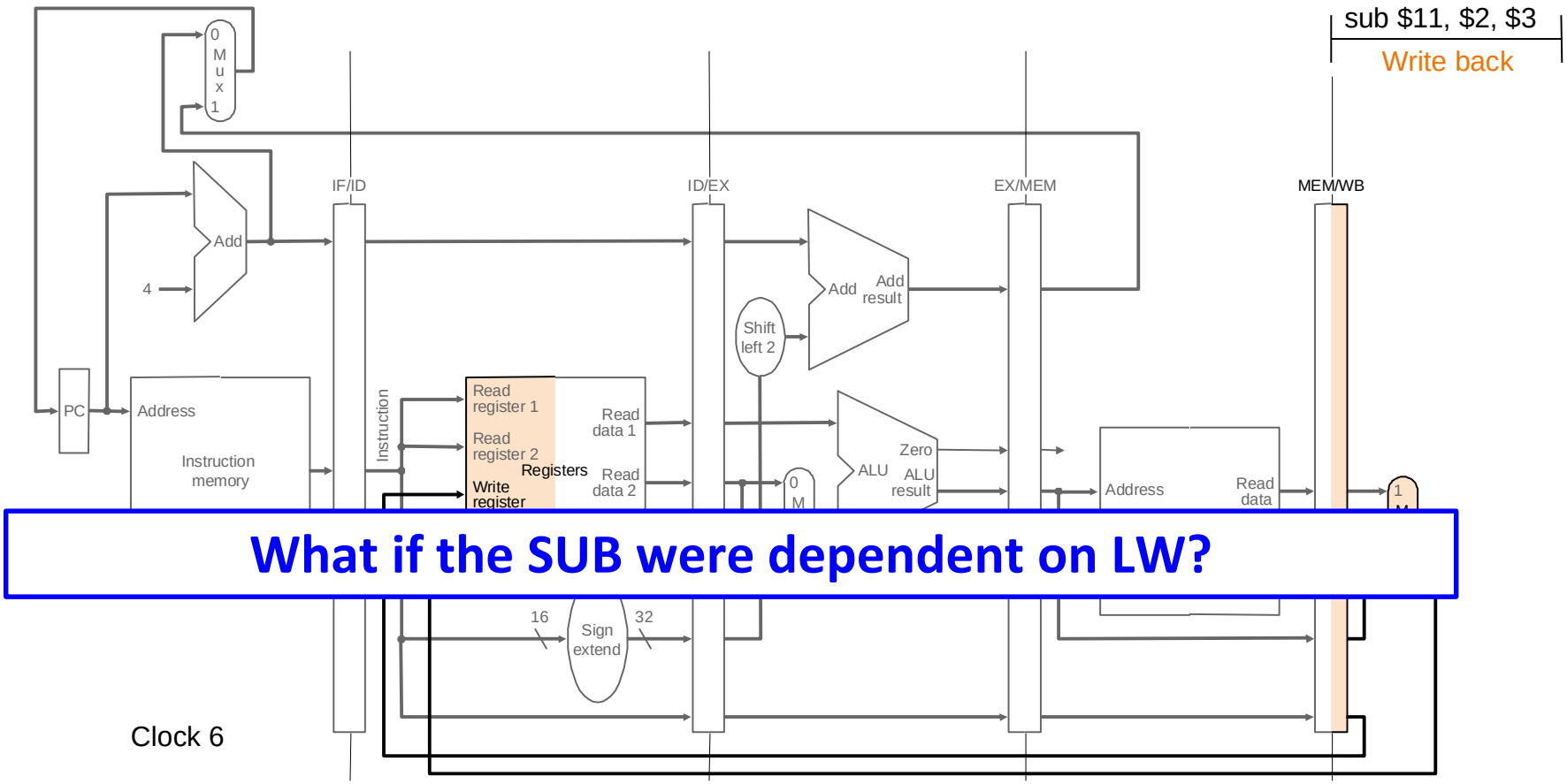
3.2 MIPS的基本流水线

3.3 流水线中的冲突

3.4 实例分析：MIPS R4000

3.5 向量处理机

Pipelined Operation Example



Based on original figure from [P&H CO&D, COPYRIGHT 2004 Elsevier. ALL RIGHTS RESERVED.]

3.3 流水线中的冲突

3.3.1 流水线的结构冲突

3.3.2 流水线的数据冲突

3.3.3 流水线的控制冲突

3.3 流水线中的冲突

1. 什么是流水线冲突（Pipeline Hazard）？

流水线冲突是指相邻或相近的两条指令因存在某种关联，后一条指令不能在原先指定的时钟周期开始执行。

- 消除冲突的基本方法——暂停

暂停流水线中某条指令及其后面所有指令的执行，该指令之前的所有指令继续执行。

3.3 流水线中的冲突

2. 三种不同类型的冲突

- 结构冲突（Structural Hazard）：当指令在重叠执行过程中，硬件资源满足不了指令重叠执行的要求而发生的冲突。
- 数据冲突（Data Hazard）：因一条指令需要用到前面指令的结果，而无法与产生结果的指令重叠执行时发生的冲突。
- 控制冲突（Control Hazard）：当流水线遇到分支指令和其它会改变PC值的指令所引起的冲突。

3.3.1 流水线的结构冲突

1. 导致结构冲突的常见原因：

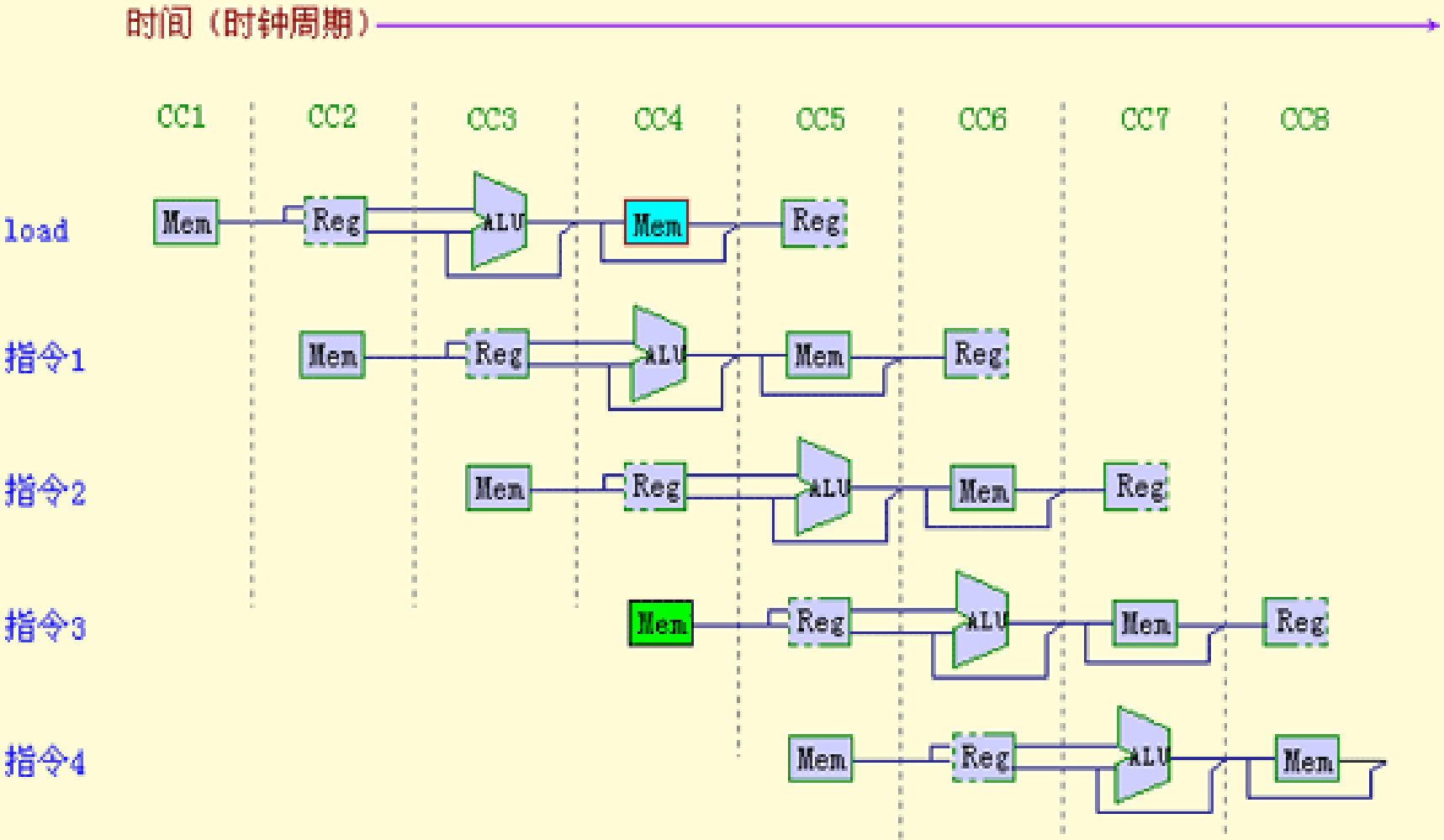
- 功能部件不是全流水
- 重复设置的资源数量不足

2. 实例：当数据和指令存在同一存储器中时，访存指令会引起存储器访问冲突。

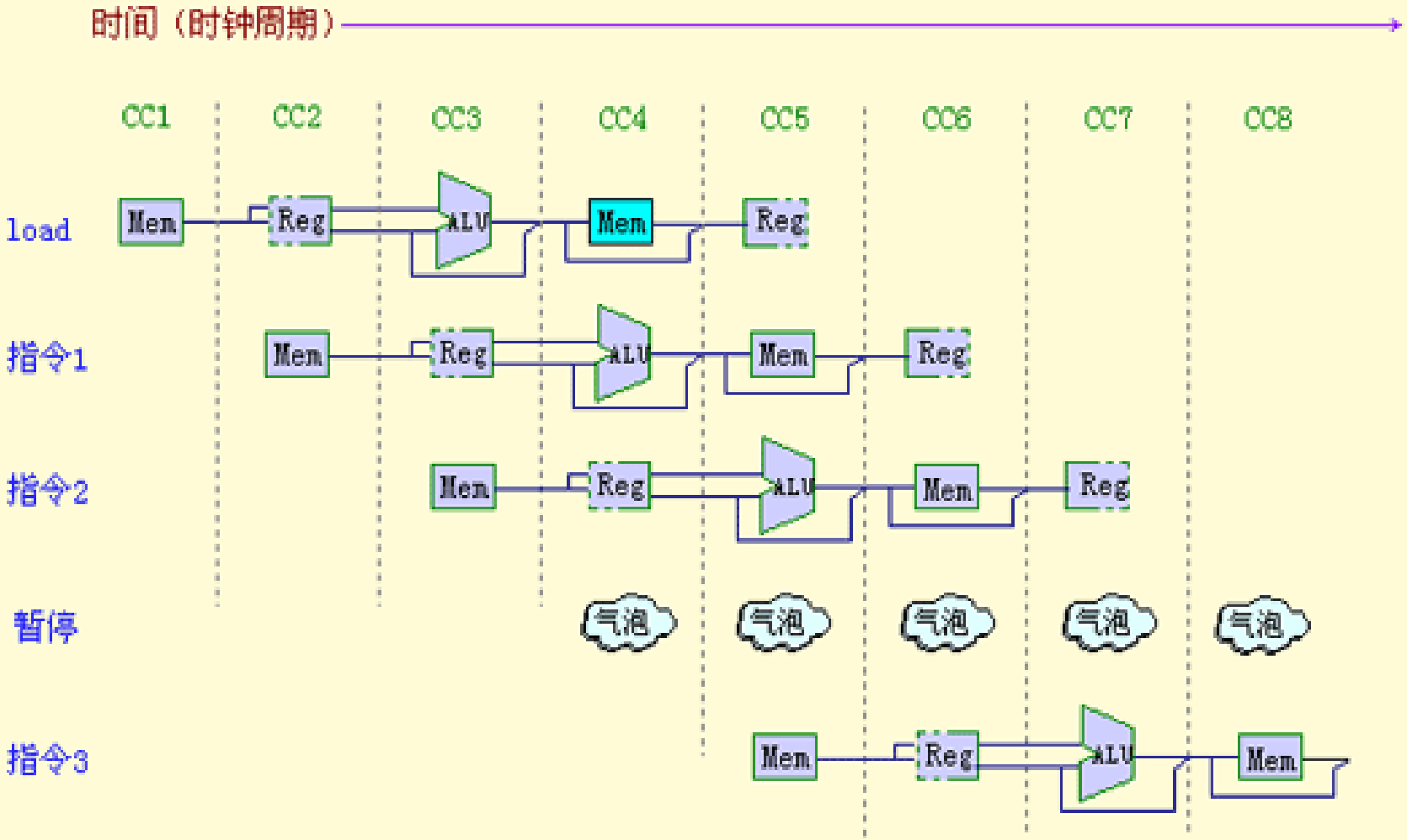
解决方法：

- I. 插入暂停周期 （时空图）
- II. 将指令存储器和数据存储器分离

结构相关举例——访存冲突


时间（时钟周期）
CC1
CC2
CC3
CC4
CC5
CC6
CC7
CC8
load
指令1
指令2
指令3
指令4
Mem
Reg
ALU

流水线气泡


时间（时钟周期）
CC1
CC2
CC3
CC4
CC5
CC6
CC7
CC8
load
指令1
指令2
暂停
指令3
Mem
Reg
ALU
气泡

引入暂停后的流水线时空图

	时钟周期									
指令编号	1	2	3	4	5	6	7	8	9	
指令i	IF	ID	EX	MEM	WB					
指令i+1		IF	ID	EX	MEM	WB				
指令i+2			IF	ID	EX	MEM	WB			
指令i+3				stall	IF	ID	EX	MEM	WB	
指令i+4						IF	ID	EX	MEM	WB

3.3.1 流水线的结构冲突

3. 避免结构冲突的方法：

- 所有功能单元完全流水化
- 设置足够多的硬件资源

  但是，硬件代价很大!

4. 有些设计方案允许结构冲突存在

- 降低成本
- 减少功能单元的延迟

例: 当前许多机器都没有将浮点功能单元完全流水，比如在MIPS实现中，浮点乘需要5个时钟周期，但对该指令不流水。请分析由此引起的结构冲突对mdljdp2基准程序在MIPS上运行的性能有何影响？为简单起见，假设浮点乘法服从均匀分布。

解: mdljdp2中浮点乘法出现的频率约为14%。

最坏情况：每个浮点乘都无法与其它操作重叠执行，都需要5个周期，此时CPI为1.56

最好情况：可以完全重叠执行，仅需要1个周期，此时没有性能损失