

计算机系统结构

第3章 流水线技术

目录

- 3.1 流水线的基本概念
- 3.2 流水线的性能指标
- 3.3 非线性流水线的调度 不考
- 3.4 流水线的相关与冲突
- 3.5 流水线的实现

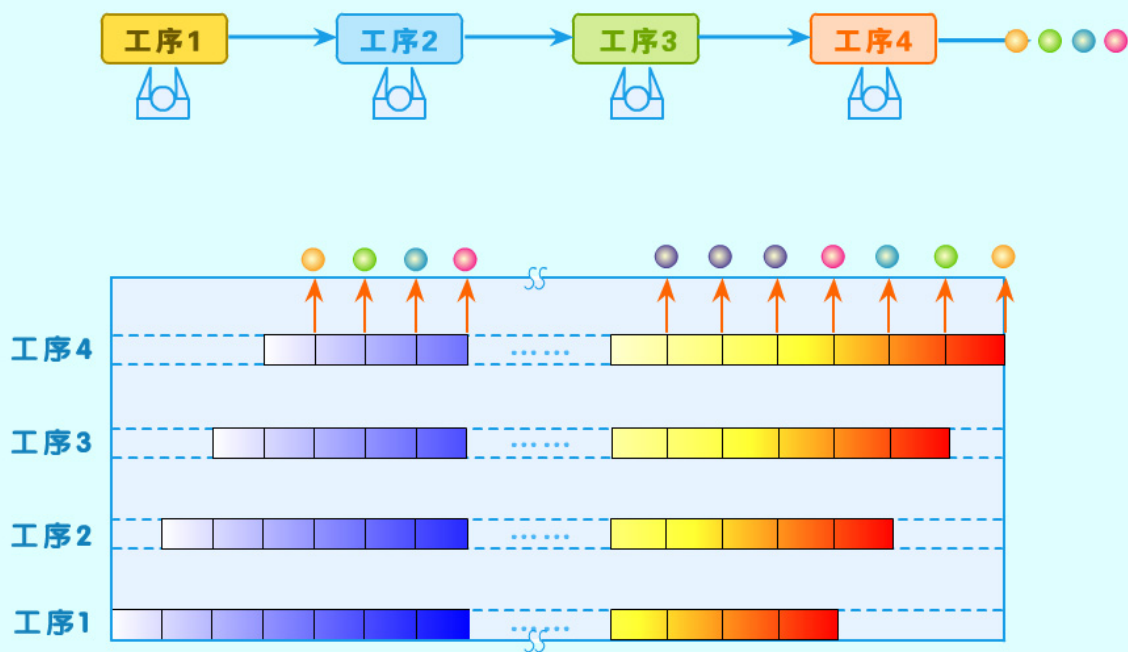
3.1 流水线的基本概念

3.1.1 什么是流水线

1. 工业生产流水线

1769年英国陶瓷工厂把一个人从头到尾完成的制陶工艺分成几十道专门工序。
亨利·福特在福特公司发明流水线装配工艺，用于汽车生产。

流水线生产过程的抽象描述



3.1 流水线的基本概念

2. 流水线技术（Pipelining）

- 把一个重复的过程分解为若干个子过程，每个子过程由专门的功能部件来实现。
- 把多个处理过程在时间上错开，依次通过各功能段，这样，每个子过程就可以与其它子过程并行进行。

3. 流水线中的每个子过程及其功能部件称为流水线的级或段（Stage），段与段相互连接形成流水线。流水线的段数称为流水线的深度（Pipeline Depth）。

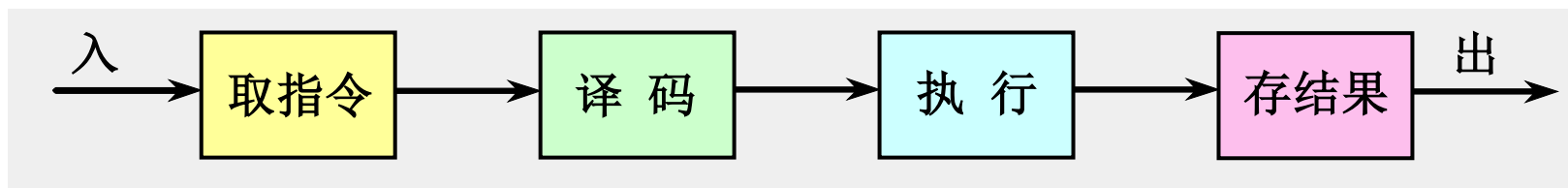
3.1 流水线的基本概念

4. 指令流水线

- 把指令的解释过程分解为译码和执行两个子过程，并让这两个子过程分别用独立的译码部件和执行部件来实现。

理想情况：速度提高一倍

- 4段指令流水线



3.1 流水线的基本概念

5. 浮点加法流水线

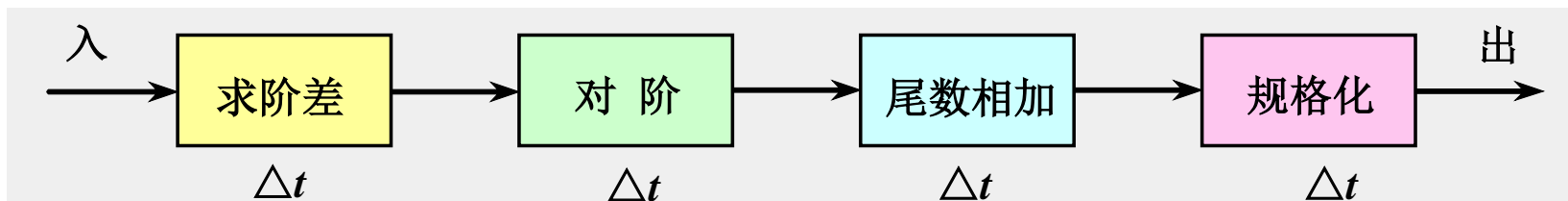
- 把流水线技术应用于运算的执行过程，就形成了运算操作流水线，也称为部件级流水线。
- 把浮点加法的全过程分解为求阶差、对阶、尾数相加、规格化四个子过程。

浮点数求和例： $x = .1010 * 2^{10}$, $y = .1100 * 2^{01}$

$x + y = .1010 * 2^{10} + .0110 * 2^{10} = (.1010 + .0110) * 2^{10}$

尾数求和： $(.1010 + .0110)$

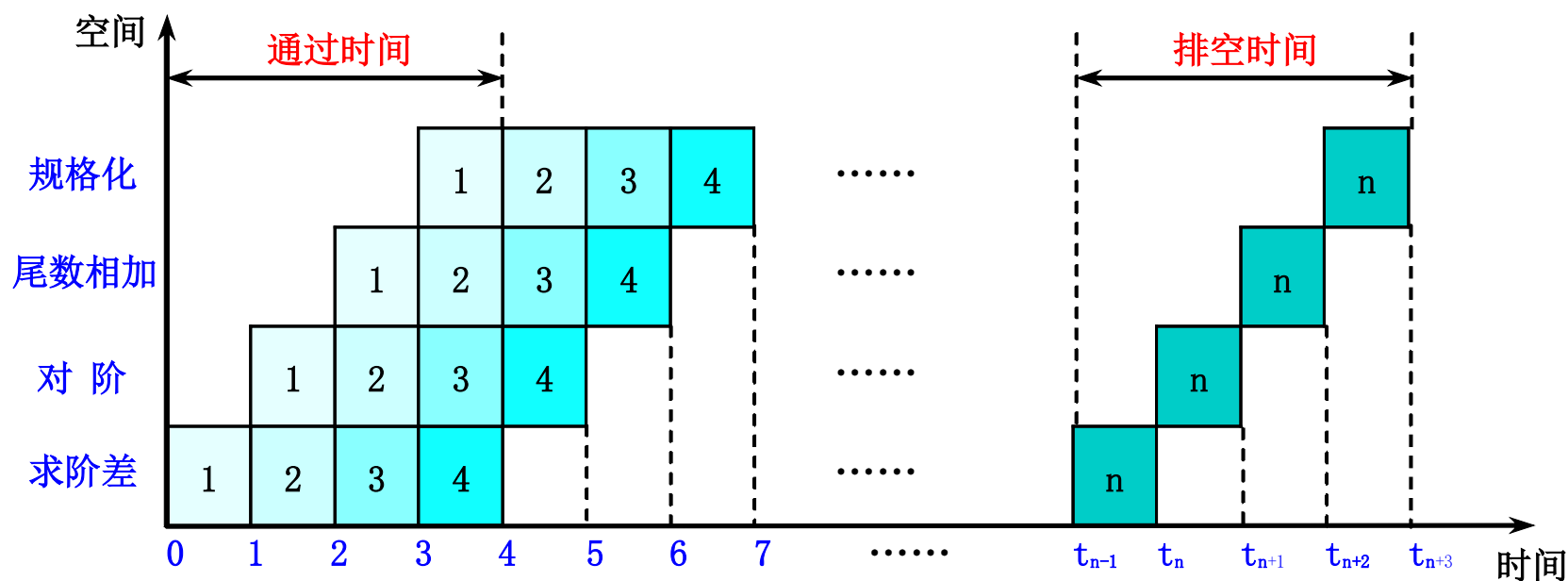
理想情况：速度提高3倍



3.1 流水线的基本概念

6. 时一空图

- 时一空图从时间和空间两个方面描述了流水线的工作过程。
时一空图中，横坐标代表时间，纵坐标代表流水线的各个段。
- 通过时间：第一个任务从进入流水线到流出结果的时间段
- 排空时间：最后一个



浮点加法流水线的时空图

3.1 流水线的基本概念

7. 流水技术的特点

- 流水线把一个处理过程分解为若干个子过程（段），每个子过程由一个专门的功能部件来实现。
- 流水线中各段的时间应尽可能相等，否则将引起流水线堵塞、断流。

时间最长的段将成为流水线的瓶颈。

- 流水线每一个段的后面都要有一个缓冲寄存器（锁存器），称为流水寄存器。
 - ▣ 作用：在相邻的两段之间传送数据，以保证提供后面要用到的信息，并把各段的处理工作相互隔离。
- 流水技术适合于大量重复的时序过程，只有在输入端不断地提供任务，才能充分发挥流水线的效率。

3.1 流水线的基本概念

3.1.2 流水线的分类

从不同的角度和观点，把流水线分成多种不同的种类。

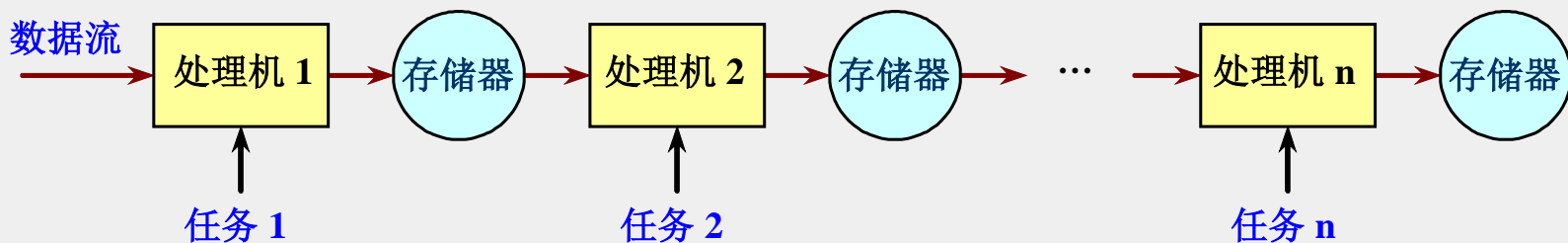
1. 部件级、处理机级及系统级流水线

（按照流水技术用于计算机系统的等级不同）

- **部件级流水线**（运算操作流水线）：把处理机中的部件分段，再把这些分段相互连接起来，使得各种类型的运算操作能够按流水方式进行。
- **处理机级流水线**（指令流水线）：把指令的执行过程按照流水方式处理。把一条指令的执行过程分解为若干个子过程，每个子过程在独立的功能部件中执行。

3.1 流水线的基本概念

- **系统级流水线**（宏流水线）：把多台处理机串行连接起来，对同一数据流进行处理，每个处理机完成整个任务中的一部分。



处理机间流水线

将多台处理机串行连接起来，对同一数据流进行处理，每个处理机完成整个任务中的一部分。



(PB级，甚至是EB级)



计算框架Map Reduce
一条简单的流水线



分布式架构

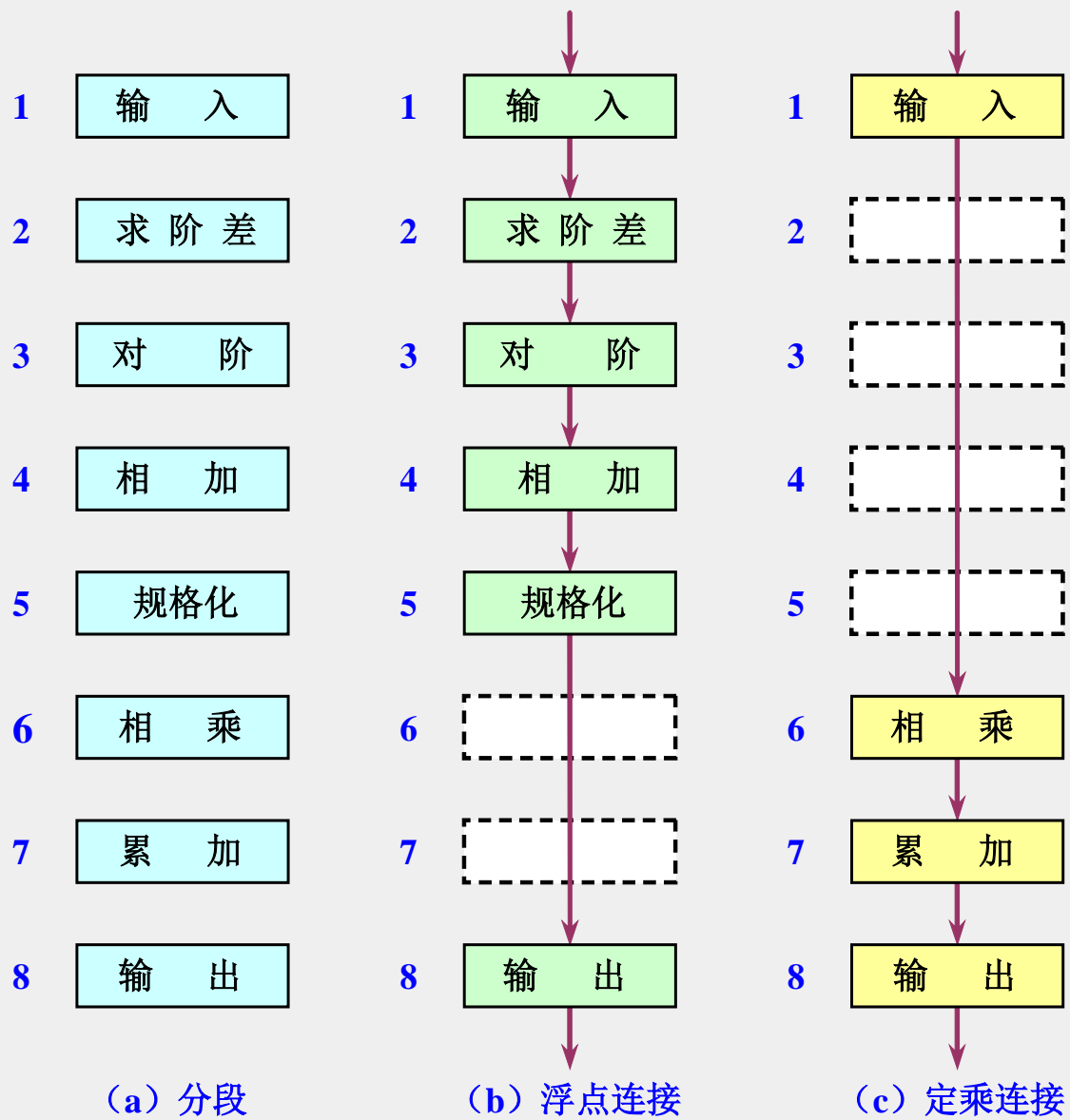
3.1 流水线的基本概念

2. 单功能流水线与多功能流水线

（按照流水线所完成的功能来分类）

- **单功能流水线：**只能完成一种固定功能的流水线。
- **多功能流水线：**流水线的各段可以进行不同的连接，以实现不同的功能。

例：TI公司ASC处理机的多功能流水线
实现浮点加减法+定点乘法运算



TI公司ASC处理机的多功能流水线

3.1 流水线的基本概念

3. 静态流水线与动态流水线

（多功能流水线按照同一时间内各段之间的连接方式分类）

- **静态流水线（Static Pipeline）**：在同一时间内，多功能流水线中的各段只能按同一种功能的连接方式工作。流水线要切换到另一种功能时，必须等前面的任务都流出流水线之后，才能改变连接。
 - 对于静态流水线来说，只有当输入的是一串相同的运算任务时，流水的效率才能得到充分的发挥。

3.1 流水线的基本概念

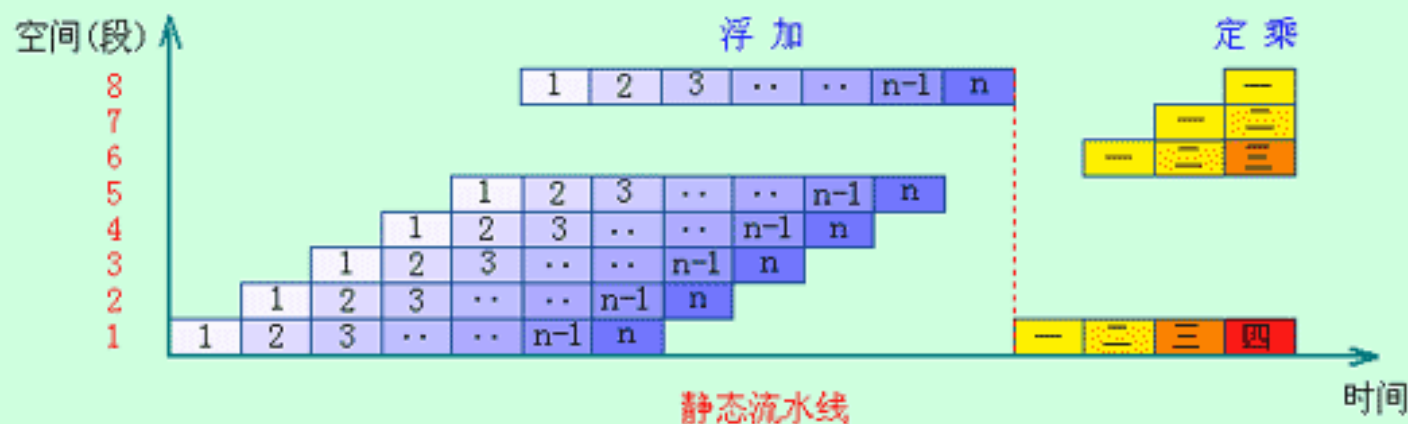
- **动态流水线 (Dynamic Pipeline)**：在同一时间内，多功能流水线中的各段可以按照不同的方式连接，同时执行多种功能。当然，多功能流水线中的任何一个功能段只能参加到一种连接中。
 - **优点**

灵活，能够提高流水线各段的使用率，从而提高处理速度。
 - **缺点**

控制复杂。
- 静、动态流水线时空图的对比

静、动态流水线的时空图

假设该流水线要先做几个浮点加法，然后再做一批定点乘法。



3.1 流水线的基本概念

4. 线性流水线与非线性流水线

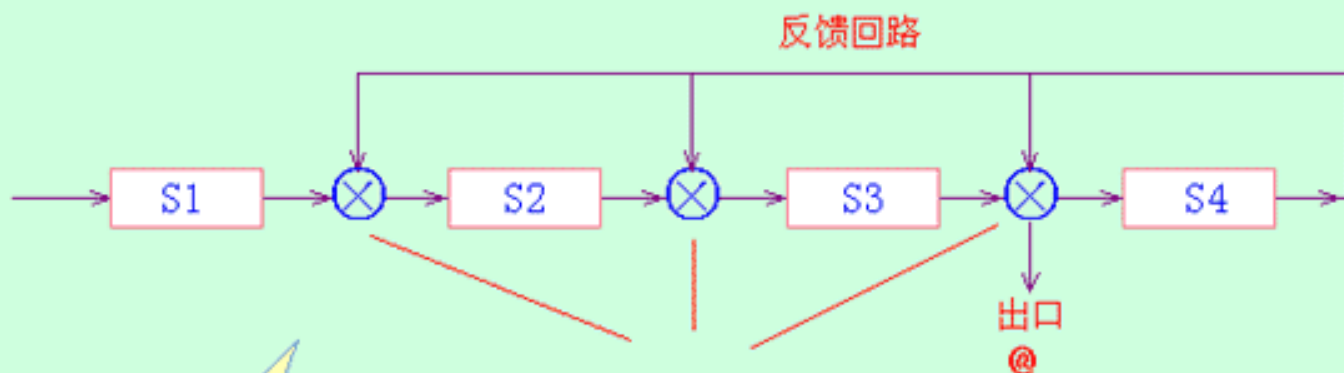
（按照流水线中是否有反馈回路来进行分类）

- **线性流水线（Linear Pipeline）**：流水线的各段串行连接，没有反馈回路。数据通过流水线中的各段时，每一个段最多只流过一次。
- **非线性流水线（Nonlinear Pipeline）**：流水线中除了有串行的连接外，还有反馈回路。
- 非线性流水线的**调度问题**
 - 确定什么时候向流水线引进新的任务，才能使该任务不会与先前进入流水线的任务发生冲突——争用流水段。

3.1 流水线的基本概念

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。

串行连接

例如任务 @：

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

3.1 流水线的基本概念

5. 顺序流水线与乱序流水线

（根据任务流入和流出的顺序是否相同来进行分类）

- **顺序流水线（In-order Pipeline）**：流水线输出端任务流出的顺序与输入端任务流入的顺序完全相同。每一个任务在流水线的各段中是一个跟着一个顺序流动的。
- **乱序流水线（Out-of-order Pipeline）**：流水线输出端任务流出的顺序与输入端任务流入的顺序可以不同，允许后进入流水线的任务先完成（从输出端流出）。

也称为无序流水线、错序流水线、异步流水线

3.1 流水线的基本概念

6. 标量处理机与向量流水处理机

- 把指令执行部件中采用了流水线的处理机称为**流水线处理机**。
- **标量处理机**：处理机不具有向量数据表示和向量指令，仅对标量数据进行流水处理。
- **向量流水处理机**：具有向量数据表示和向量指令的处理机。

向量数据表示和流水技术的结合。

3.2 流水线的性能指标

3.2.1 吞吐率

吞吐率：在单位时间内流水线所完成的任务数量或输出结果的数量。

$$TP = \frac{n}{T_k}$$

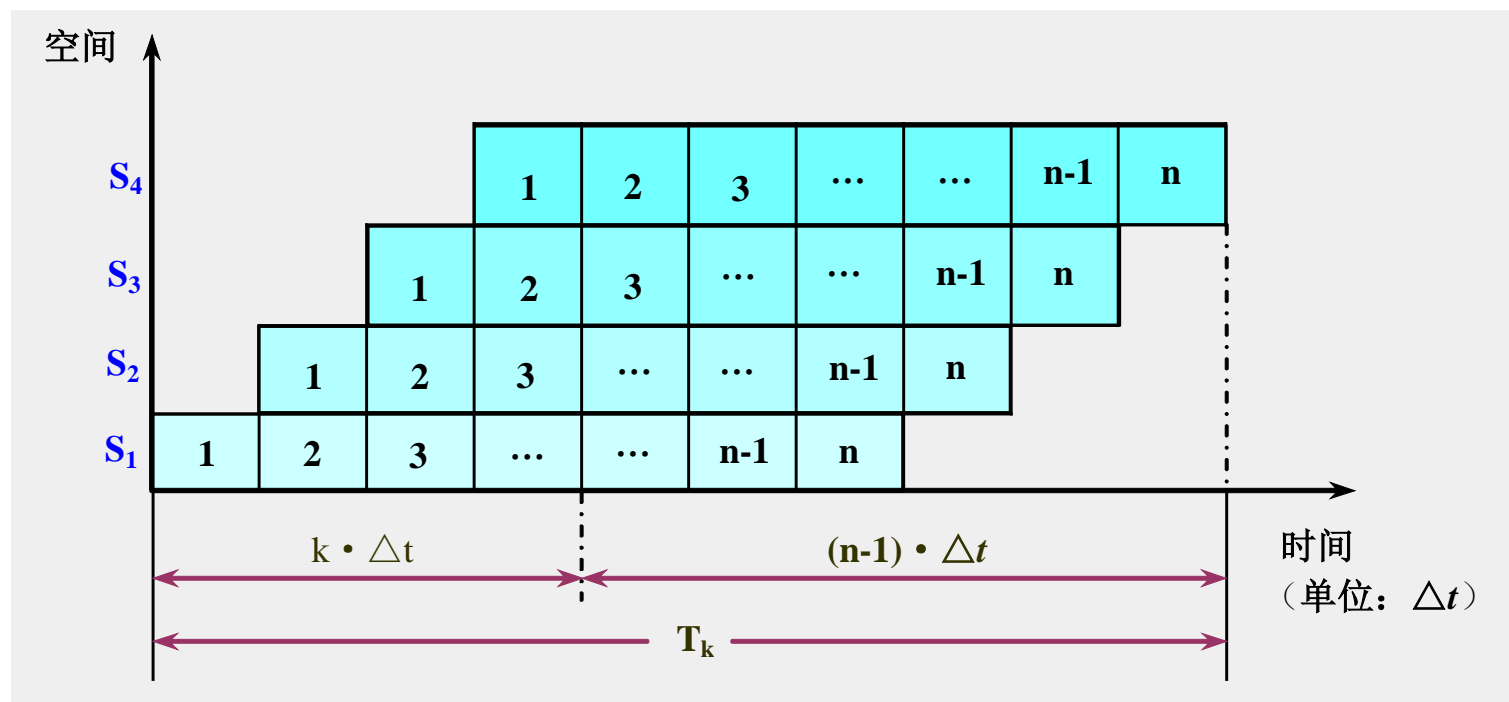
n ：任务数

T_k ：处理完成 n 个任务所用的时间

3.2 流水线的性能指标

1. 各段时间均相等的流水线

➤ 各段时间均相等的流水线时空图



3.2 流水线的性能指标

- 流水线完成 n 个连续任务所需要的总时间为：
(假设一条 k 段线性流水线)

$$T_k = k \Delta t + (n-1) \Delta t = (k+n-1) \Delta t$$

- 流水线的实际吞吐率

$$TP = \frac{n}{(k+n-1)\Delta t}$$

- 最大吞吐率

$$TP_{\max} = \lim_{n \rightarrow \infty} \frac{n}{(k+n-1)\Delta t} = \frac{1}{\Delta t}$$

3.2 流水线的性能指标

➤ 最大吞吐率与实际吞吐率的关系

$$TP = \frac{n}{k + n - 1} TP_{\max}$$

- ❑ 流水线的实际吞吐率小于最大吞吐率，它除了与每个段的时间有关外，还与流水线的段数 k 以及输入到流水线中的任务数 n 等有关。
- ❑ 只有当 $n \gg k$ 时，才有 $TP \approx TP_{\max}$ 。

3.2 流水线的性能指标

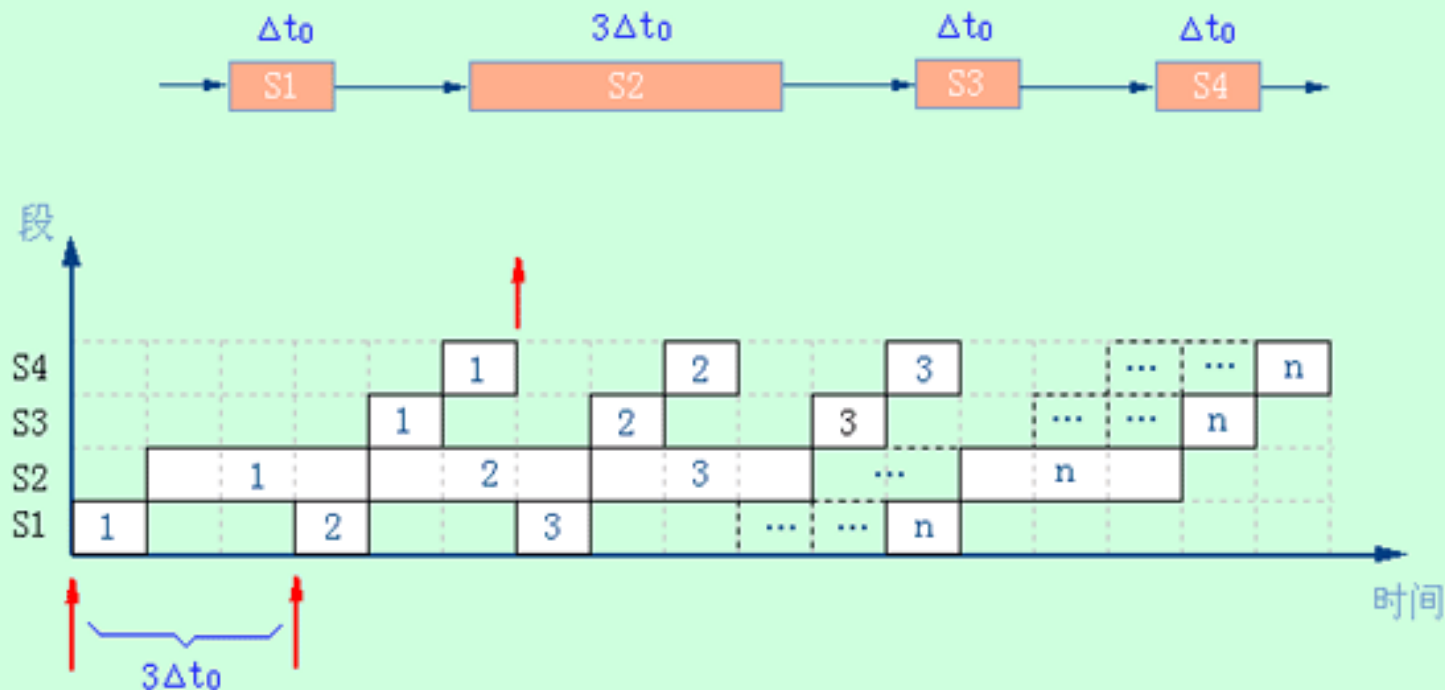
2. 各段时间不完全相等的流水线

例1 一条4段的流水线

流水线中这种时间最长的段称为流水线的**瓶颈段**。

流水线的时-空图
(各段时间不等)

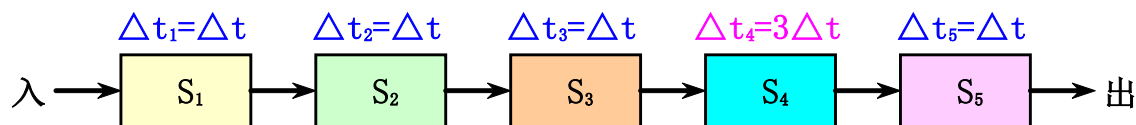
S1, S3, S4各段的时间: Δt
S2的时间: $3\Delta t$ (瓶颈段)



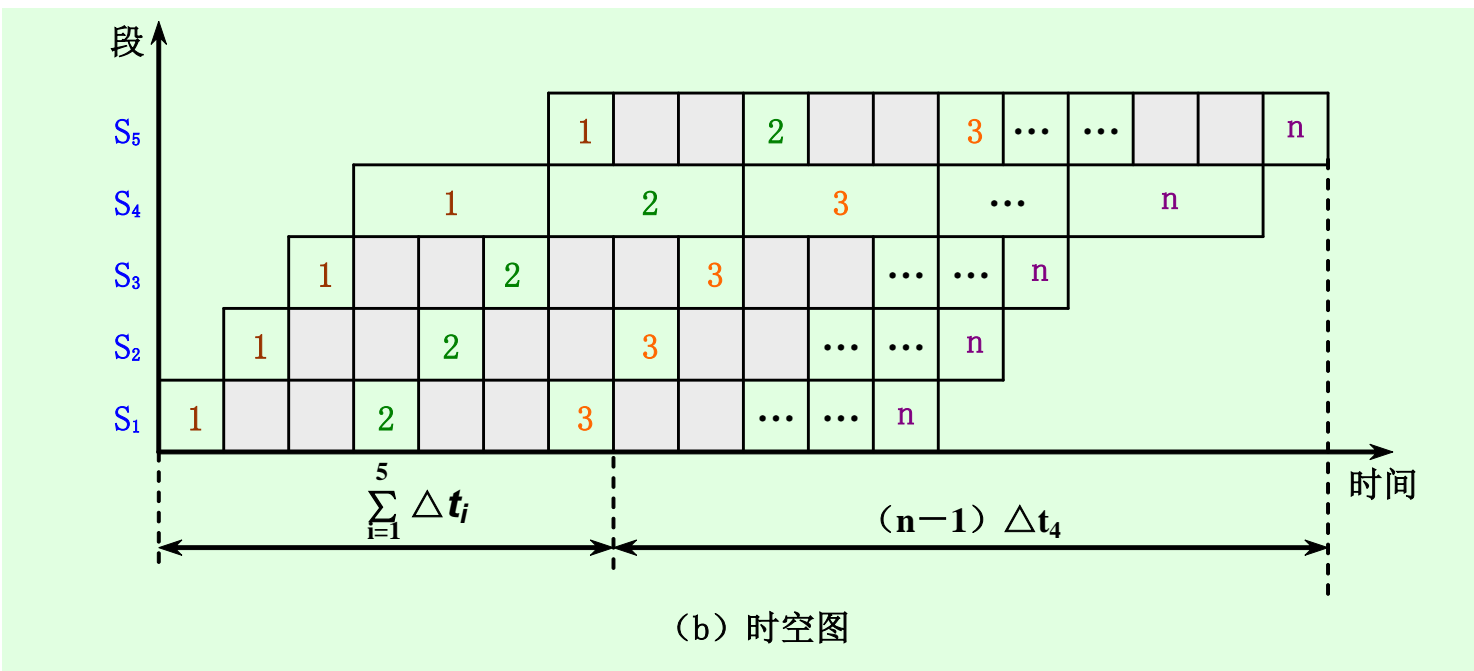
3.2 流水线的性能指标

例2：一条5段的流水线

- S_1, S_2, S_3, S_5 各段的时间: Δt
- S_4 的时间: $3 \Delta t$ (瓶颈段)



(a) 流水线



3.2 流水线的性能指标

- 各段时间不等的流水线的实际吞吐率为：

(Δt_i 为第 i 段的时间，共有 k 个段)

$$TP = \frac{n}{\sum_{i=1}^k \Delta t_i + (n-1) \max(\Delta t_1, \Delta t_2, \dots, \Delta t_k)}$$

- 流水线的最大吞吐率为：

$$TP_{\max} = \frac{1}{\max(\Delta t_1, \Delta t_2, \dots, \Delta t_k)}$$

对前面例2中的5段流水线

最大吞吐率为：

$$TP_{\max} = \frac{1}{3\Delta t}$$

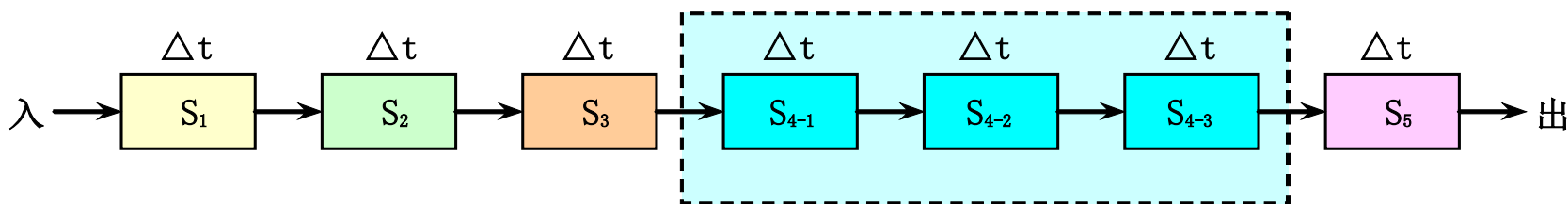
3.2 流水线的性能指标

3. 解决流水线瓶颈问题的常用方法

➤ 细分瓶颈段

例如：对前面的5段流水线

把瓶颈段 S_4 细分为3个子流水线段： S_{4-1} ， S_{4-2} ， S_{4-3}



改进后的流水线的吞吐率：

$$TP_{\max} = \frac{1}{\Delta t}$$

3.2 流水线的性能指标

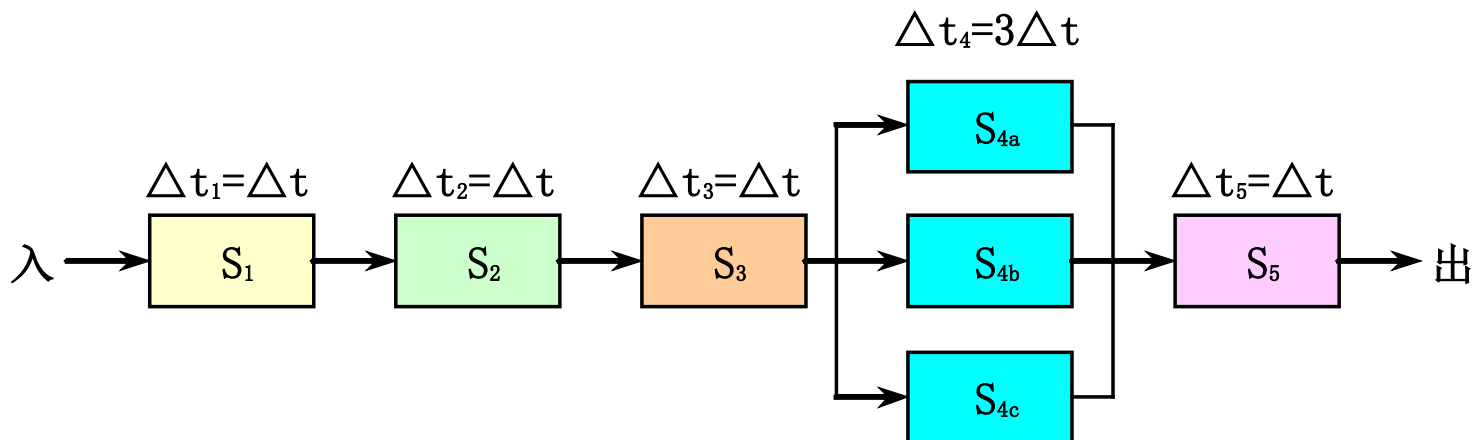
➤ 重复设置瓶颈段

- 举例：

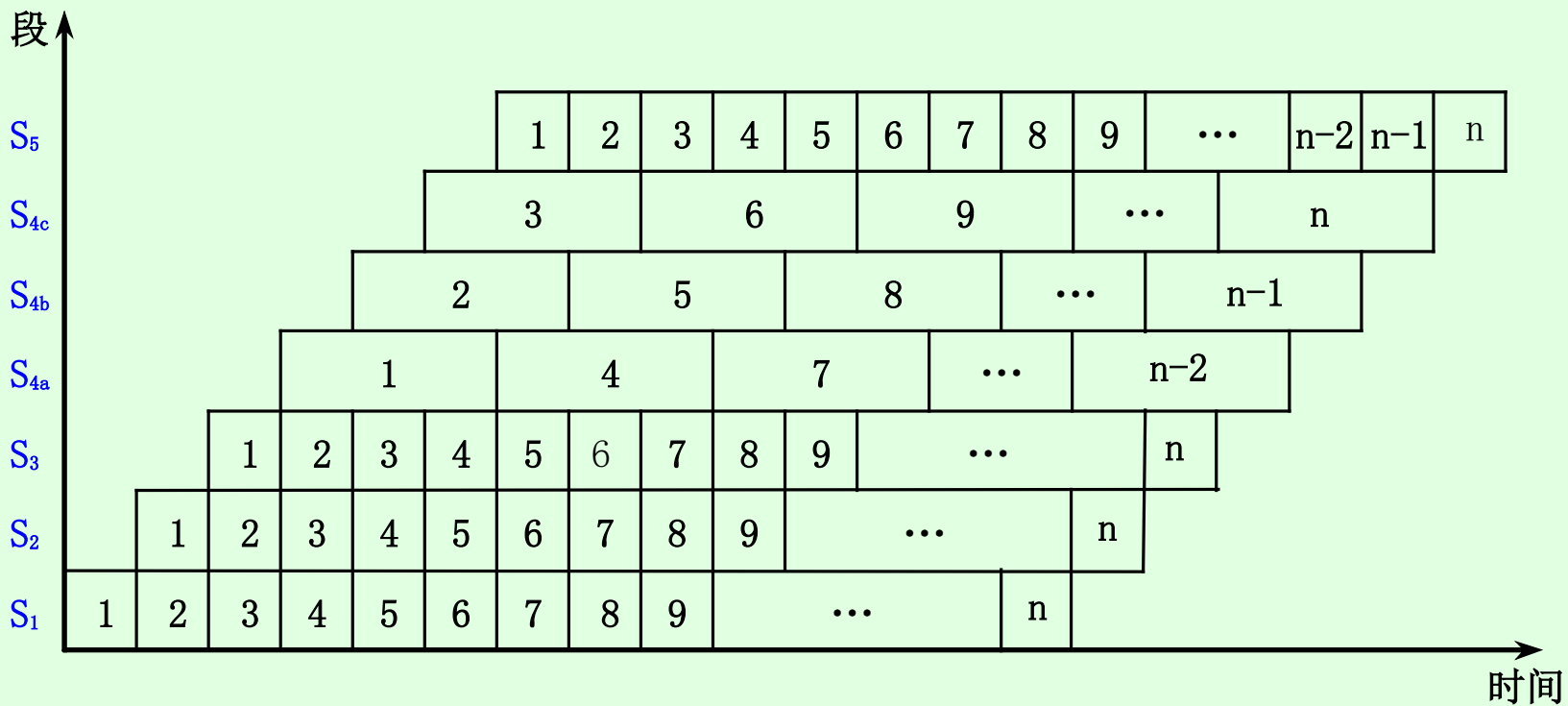
- **缺点：** 控制逻辑比较复杂，所需的硬件增加了。

例如： 对前面的5段流水线

重复设置瓶颈段 S_4 ： S_{4a} ， S_{4b} ， S_{4c}



3.2 流水线的性能指标



重复设置瓶颈段后的时空图

3.2 流水线的性能指标

3.2.2 流水线的加速比

加速比：完成同样一批任务，不使用流水线所用的时间与使用流水线所用的时间之比。

假设：不使用流水线（即顺序执行）所用的时间为 T_s ，使用流水线后所用的时间为 T_k ，则该流水线的加速比为：

$$S = \frac{T_s}{T_k}$$

3.2 流水线的性能指标

1. 流水线各段时间相等（都是 Δt ）

- 一条 k 段流水线完成 n 个连续任务

所需要的时间为： $T_k = (k + n - 1)\Delta t$

- 顺序执行 n 个任务

所需要的时间： $T_s = nk\Delta t$

- 流水线的实际加速比为：

$$S = \frac{nk}{k + n - 1}$$

- 最大加速比

$$S_{\max} = \lim_{n \rightarrow \infty} \frac{nk}{k + n - 1} = k$$

当 $n \gg k$ 时， $S \approx k$

思考：流水线的段数愈多愈好？

3.2 流水线的性能指标

2. 流水线的各段时间不完全相等时

- 一条 k 段流水线完成 n 个连续任务的实际加速比为：

$$S = \frac{n \sum_{i=1}^k \Delta t_i}{\sum_{i=1}^k \Delta t_i + (n-1) \max(\Delta t_1, \Delta t_2, \dots, \Delta t_k)}$$

3.2 流水线的性能指标

3.2.3 流水线的效率

流水线的效率：流水线中的设备实际使用时间与整个运行时间的比值，即流水线设备的利用率。

由于流水线有通过时间和排空时间，所以在连续完成 n 个任务的时间内，各段并不是满负荷地工作。

1. 各段时间相等

➤ 各段的效率 e_i 相同

$$e_1 = e_2 = \cdots = e_k = \frac{n\Delta t}{T_k} = \frac{n}{k + n - 1}$$

3.2 流水线的性能指标

- 整条流水线的效率为：

$$E = \frac{e_1 + e_2 + \cdots + e_k}{k} = \frac{ke_1}{k} = \frac{kn\Delta t}{kT_k}$$

- 可以写成：

$$E = \frac{n}{k + n - 1}$$

- 最高效率为：

$$E_{\max} = \lim_{n \rightarrow \infty} \frac{n}{k + n - 1} = 1$$

当 $n \gg k$ 时， $E \approx 1$ 。

- 当流水线各段时间相等时，流水线的效率与吞吐率成正比。

$$E = TP \cdot \Delta t$$

3.2 流水线的性能指标

- 流水线的效率是流水线的实际加速比 S 与它的最大加速比 k 的比值。

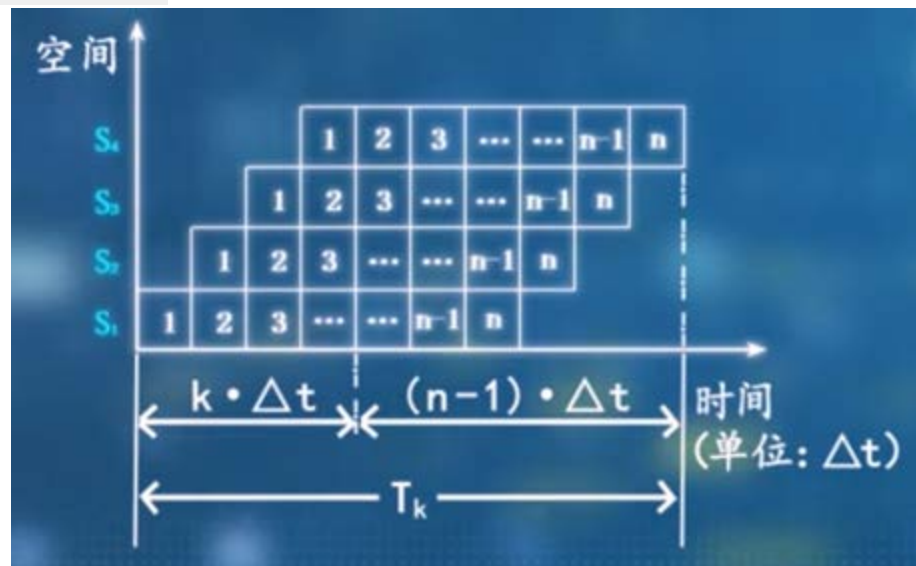
$$E = \frac{S}{k}$$

当 $E=1$ 时， $S=k$ ，实际加速比达到最大。

3.2 流水线的性能指标

3. 从时空图上看，效率就是 n 个任务占用的时空面积和 k 个段总的时空面积之比。

$$E = \frac{n \text{ 个任务实际占用的时空区}}{k \text{ 个段总的时空区}}$$



当各段时间不相等时:

$$E = \frac{n \cdot \sum_{i=1}^k \Delta t_i}{k \left[\sum_{i=1}^k \Delta t_i + (n-1) \cdot \max(\Delta t_1, \Delta t_2, \dots, \Delta t_k) \right]}$$

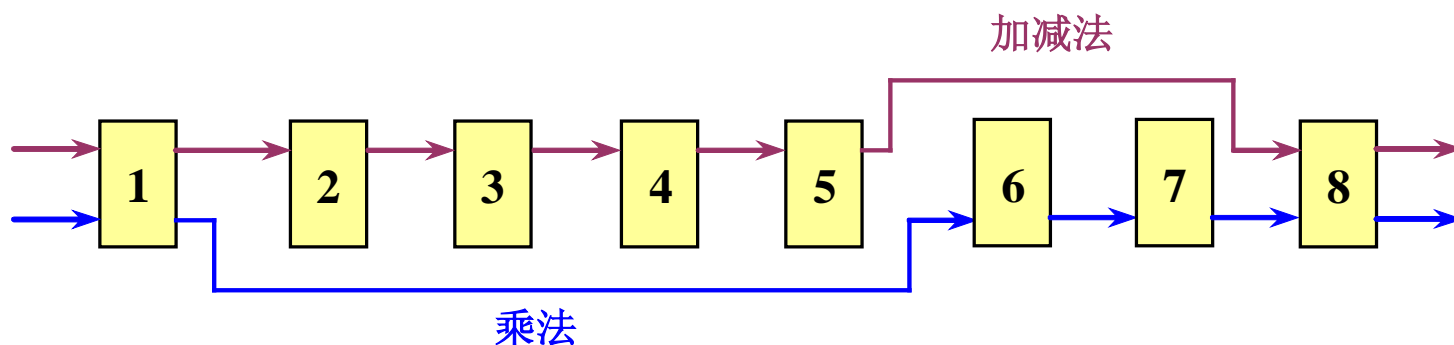
3.2 流水线的性能指标

3.2.4 流水线的性能分析举例

例3.1 设在下图所示的静态流水线上计算：

$$\prod_{i=1}^4 (A_i + B_i)$$

流水线的输出可以直接返回输入端或暂存于相应的流水寄存器中，试计算其吞吐率、加速比和效率。



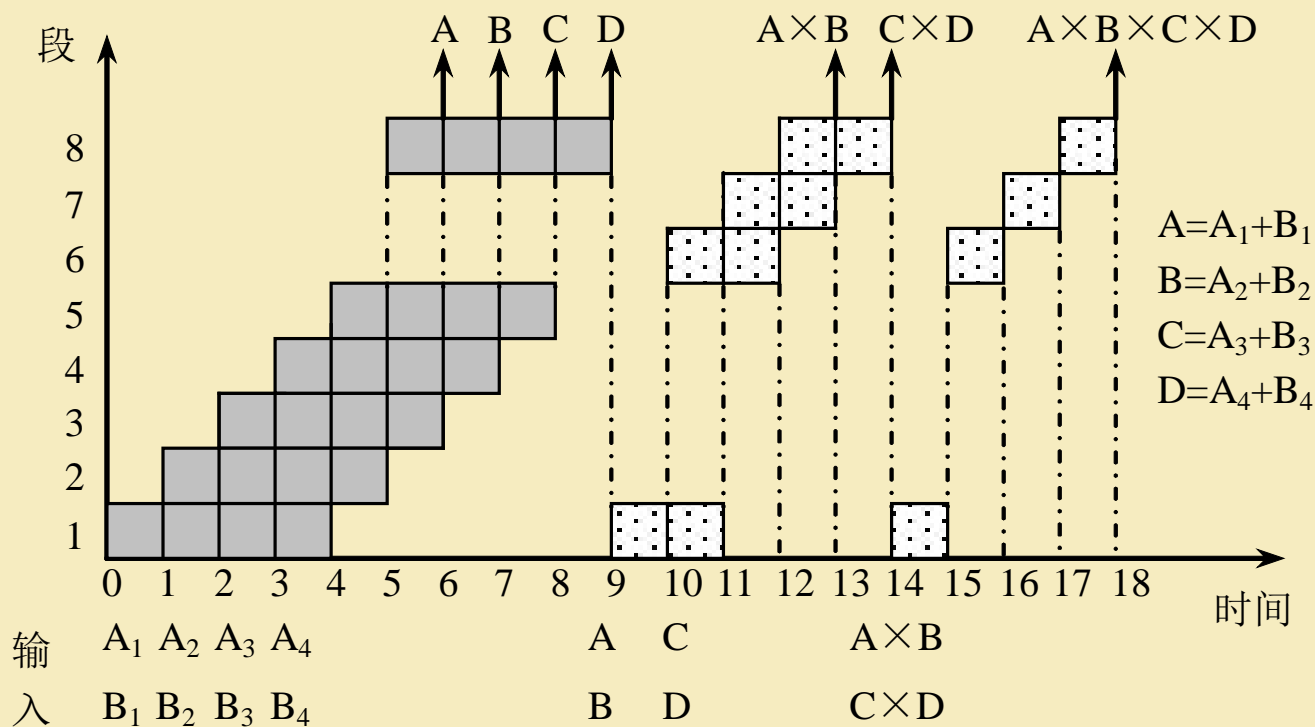
(每段的时间都为 Δt)

3.2 流水线的性能指标

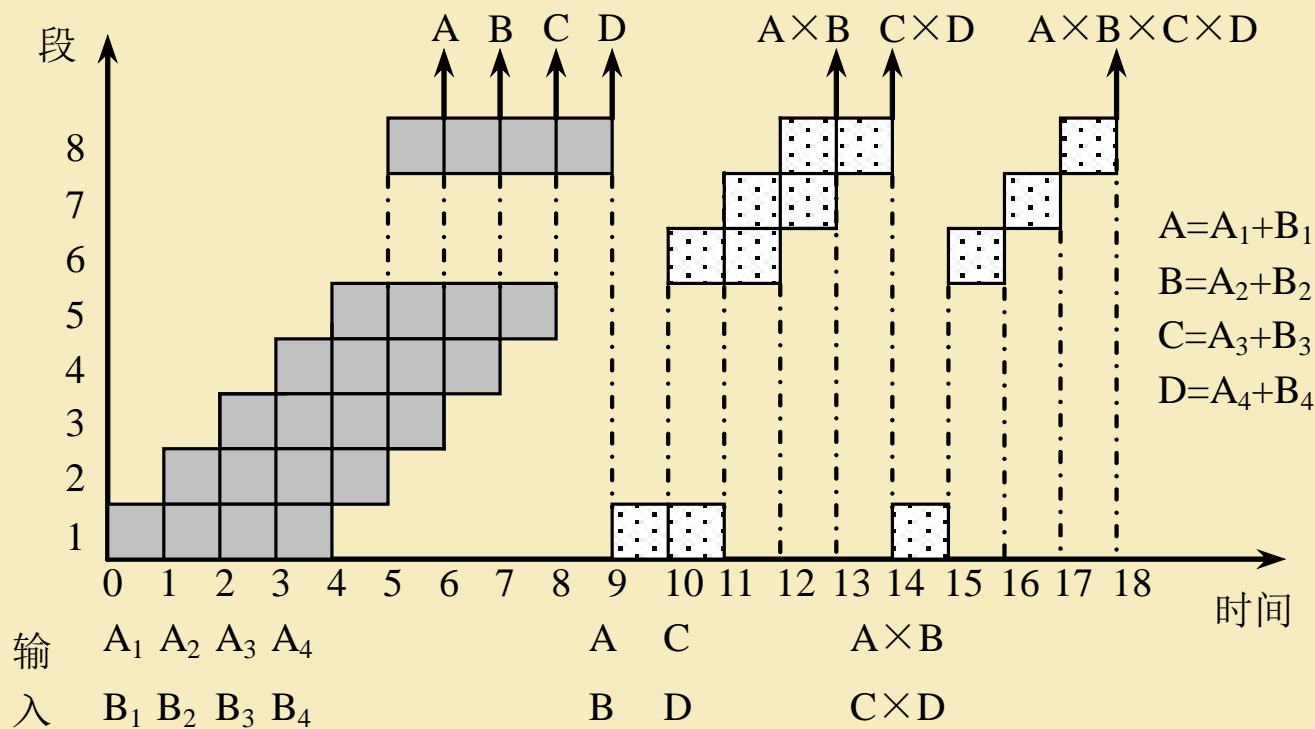
解：（1）选择适合于流水线工作的算法

- 先计算 A_1+B_1 、 A_2+B_2 、 A_3+B_3 和 A_4+B_4 ；
- 再计算 $(A_1+B_1) \times (A_2+B_2)$ 和 $(A_3+B_3) \times (A_4+B_4)$ ；
- 然后求总的乘积结果。

（2）画出时空图



3.2 流水线的性能指标



吞吐率：18个时间间隔，7个结果： $TP = \frac{n}{T_k} = \frac{7}{18\Delta t}$

加速比：无流水线一次求和 $6\Delta t$ ，一次求积 $4\Delta t$ ，7个结果共需 $36\Delta t$ ： $S = \frac{36\Delta t}{18\Delta t} = 2$

效率：阴影面积比8个段总时间是空区面积： $E = \frac{4 \times 6 + 3 \times 4}{8 \times 18} = 0.25$

3.2 流水线的性能指标

- ▣ 流水线的效率 25%

可以看出，在求解此问题时，该流水线的效率不高。

➤ 主要原因

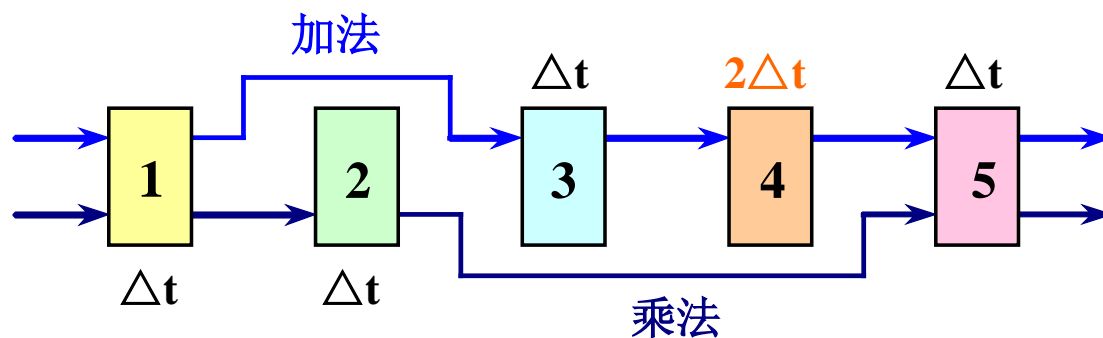
- ▣ 多功能流水线在做某一种运算时，总有一些段是空闲的；
- ▣ 静态流水线在进行功能切换时，要等前一种运算全部流出流水线后才能进行后面的运算；
- ▣ 运算之间存在关联，后面有些运算要用到前面运算的结果；
- ▣ 流水线的工作过程有建立与排空部分。

3.2 流水线的性能指标

例3.2 有一条动态多功能流水线由5段组成，加法用1、3、4、5段，乘法用1、2、5段，第4段的时间为 $2\Delta t$ ，其余各段时间均为 Δt ，而且流水线的输出可以直接返回输入端或暂存于相应的流水寄存器中。若在该流水线上计算：

$$\sum_{i=1}^4 (A_i \times B_i)$$

试计算其吞吐率、加速比和效率。



3.2 流水线的性能指标

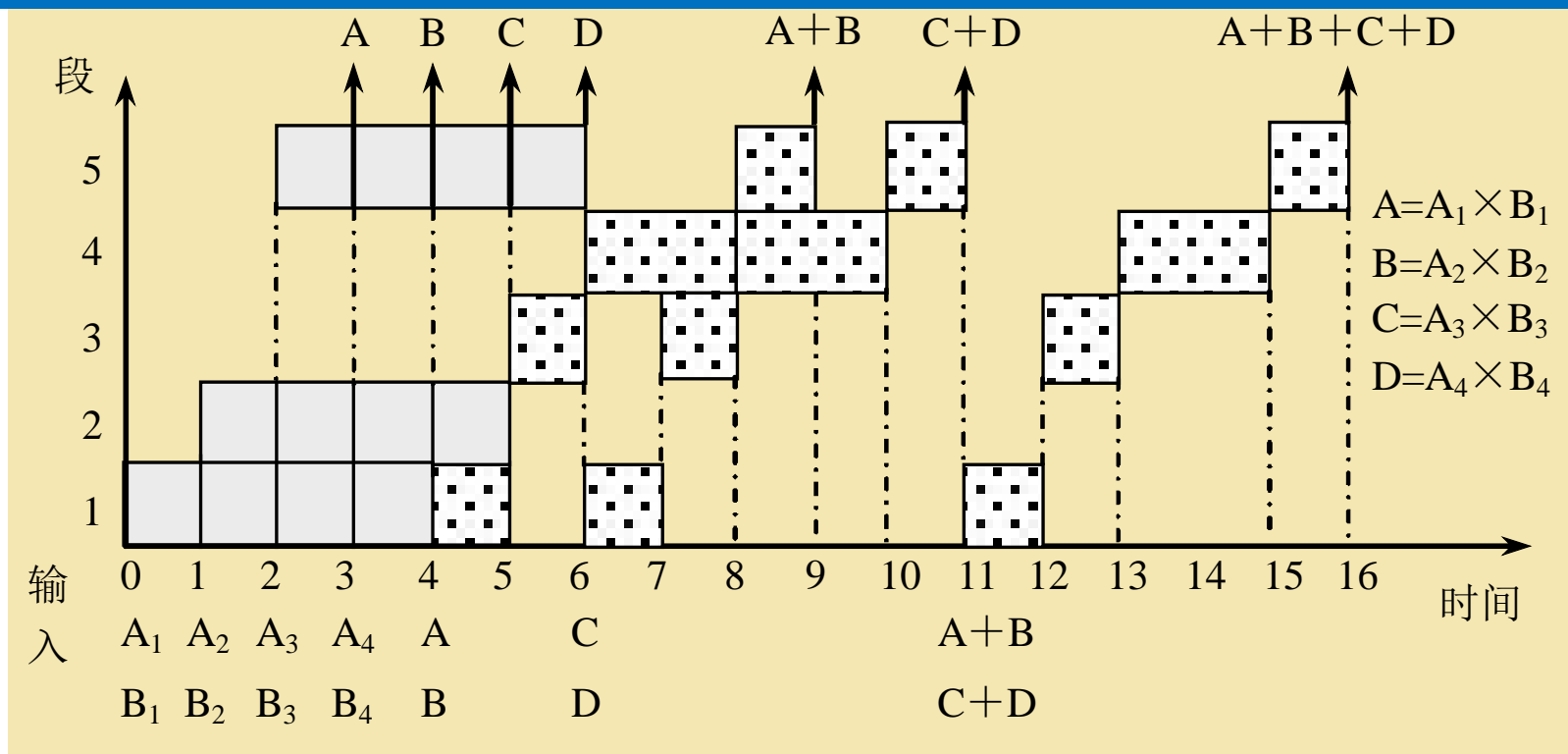
解：(1) 选择适合于流水线工作的算法

- 应先计算 $A_1 \times B_1$ 、 $A_2 \times B_2$ 、 $A_3 \times B_3$ 和 $A_4 \times B_4$ ；
- 再计算 $(A_1 \times B_1) + (A_2 \times B_2)$
 $(A_3 \times B_3) + (A_4 \times B_4)$ ；
- 然后求总的累加结果。

(2) 画出时空图

(3) 计算性能

3.2 流水线的性能指标



吞吐率：16个时间间隔，7个结果：
$$TP = \frac{7}{16\Delta t}$$

加速比：无流水线一次求积 $3\Delta t$ ，一次求和 $5\Delta t$ ，7个结果共需 $27\Delta t$ ：
$$S = \frac{27\Delta t}{16\Delta t} \approx 1.69$$

效率：阴影面积比8个段总时间是空区面积：
$$E = \frac{4 \times 3 + 3 \times 5}{5 \times 16} \approx 0.338$$

3.2 流水线的性能指标

3.2.5 流水线设计中的若干问题

1. 瓶颈问题

- 理想情况下，流水线在工作时，其中的任务是同步地每一个时钟周期往前流动一段。
- 当流水线各段不均匀时，机器的时钟周期取决于瓶颈段的延迟时间。
- 在设计流水线时，要尽可能使各段时间相等。

3.2 流水线的性能指标

2. 流水线的额外开销

- 流水寄存器需要建立时间和传输延迟
 - **建立时间**：在触发写操作的时钟信号到达之前，寄存器输入必须保持稳定的时间。
 - **传输延迟**：时钟信号到达后到寄存器输出可用的时间。
- 时钟偏移开销
 - 流水线中，时钟到达各流水寄存器的最大差值时间。（时钟到达各流水寄存器的时间不是完全相同）
- ◆ 流水线并不能减少（一般是增加）单条指令的执行时间，但却能提高吞吐率。
- ◆ 增加流水线的深度（段数）可以提高流水线的性能。
- ◆ 流水线的深度受限于流水线的额外开销。
- ◆ 当时钟周期小到与额外开销相近时，流水线已没意义。
===》流水线的段数愈多愈好？

3.2 流水线的性能指标

3. 冲突问题

后面计算要用到前面的计算结果；指令流水线中指令之间存在关联，则要相互等待引起流水线停顿。

-----流水线设计中要解决的**重要问题之一**。

- 在非线性流水线中，存在反馈回路，当一个任务在流水线中流过时，可能要多次经过某些段。
- 流水线调度要**解决的问题**：

应按什么样的时间间隔向流水线输入新任务，才能既不发生功能段使用冲突，又能使流水线有较高的吞吐率和效率？

3.3 非线性流水线的调度

3.3.1 单功能非线性流水线的最优调度

- 向一条非线性流水线的输入端连续输入两个任务之间的时间间隔称为非线性流水线的**启动距离** (Initiation Interval) 。
- 会引起非线性流水线功能段使用冲突的启动距离则称为**禁用启动距离**。
- 启动距离和禁用启动距离一般都用时钟周期数来表示，是一个正整数。
- 若两个任务进入非线性流水线的的时间间隔不是禁用启动距离，那么这两个任务之间不会产生冲突。
- **预约表(Reservation Table)**
 - 横向（向右）：时间（一般用时钟周期表示）
 - 纵向（向下）：流水线的段

3.3 非线性流水线的调度

例：一个5功能段非线性流水线预约表

时间 功能段	1	2	3	4	5	6	7	8	9
S1	√								√
S2		√	√					√	
S3				√					
S4					√	√			
S5							√	√	

- ❑ 如果在第n个时钟周期使用第k段，则在第k行和第n列的交叉处的格子里有一个√。
- ❑ 如果在第k行和第n列的交叉处的格子里有一个√，则表示在第n个时钟周期要使用第k段。

3.3 非线性流水线的调度

1. 根据预约表写出禁止表F(Forbidden List)

➤ **禁止表F**：一个由禁用启动距离构成的集合。

➤ 具体方法

对于预约表的每一行的任何一对 $\sqrt{}$ ，用它们所在的列号相减（大的减小的），列出各种可能的差值，然后删除相同的，剩下的就是禁止表的元素。

➤ 在上例中

- 第一行的差值只有一个：8；
 - 第二行的差值有3个：1, 5, 6；
 - 第3行只有一个 $\sqrt{}$ ，没有差值；
 - 第4和第5行的差值都只有一个：1；
- 其禁止表是： $F = \{ 1, 5, 6, 8 \}$

禁止表仅考虑了相连两个任务之间的冲突，不相连的任务也会发生冲突。=>

3.3 非线性流水线的调度

2. 根据禁止表F写出初始冲突向量 C_0

(进行从一个集合到一个二进制位串的转变)

➤ **冲突向量(Collision Vector) C:**

一个N位的二进制位串。

➤ 设 $C_0 = (c_N c_{N-1} \dots c_i \dots c_2 c_1)$ ，则：

$$C_i = \begin{cases} 1 & i \in F \\ 0 & i \notin F \end{cases}$$

□ $c_i=0$ ：允许间隔*i*个时钟周期后送入后续任务

□ $c_i=1$ ：不允许间隔*i*个时钟周期后送入后续任务

➤ 对于上面的例子

$$F = \{ 1, 5, 6, 8 \}$$

$$C_0 = (10110001)$$

3.3 非线性流水线的调度

3. 根据初始冲突向量 C_0 画出状态转换图

- 当第一个任务流入流水线后，初始冲突向量 C_0 决定了下一个任务需间隔多少个时钟周期才可以流入。
- 在第二个任务流入后，新的冲突向量是怎样的呢？
 - 假设第二个任务是在与第一个任务间隔 j 个时钟周期流入，这时，由于第一个任务已经在流水线中前进了 j 个时钟周期，其相应的禁止表中各元素的值都应该减去 j ，并丢弃小于等于0的值。
 - 对冲突向量来说，就是逻辑右移 j 位（左边补0）。
 - 新冲突向量，就是对它们的冲突向量进行“或”运算。

$$SHR^j(C_0) \vee C_0$$

其中： SHR^j 表示逻辑右移 j 位

3.3 非线性流水线的调度

- 推广到更一般的情况

假设: C_k : 当前的冲突向量

j : 允许的时间间隔

则新的冲突向量为:

$$SHR^{(j)}(C_k) \vee C_0$$

- 对于把新的冲突向量作为当前冲突向量, 反复使用所有允许的时间间隔都按上述步骤求出其新的冲突向量, 并且上述步骤, 直到不再产生新的冲突向量为止。

3.3 非线性流水线的调度

- 从初始冲突向量 C_0 出发，反复应用上述步骤，可以求得所有的冲突向量以及产生这些向量所对应的时间间隔。由此可以画出用冲突向量表示的流水线状态转移图。
- 流水线状态转移图三要素：
 - 流水线状态集合（所有可能的冲突向量）
 - 有向弧：表示状态转移的方向
 - 弧上的数字：表示引入后续任务（从而产生新的冲突向量）所用的时间间隔（时钟周期数）
- 从流水线状态转移图可以找到所有可行的调度方案。

3.3 非线性流水线的调度

对于上面的例子

(1) $C_0 = (10110001)$

引入后续任务可用的时间间隔为：2、3、4、7个时钟周期

如果采用2，则新的冲突向量为：

$$(00101100) \vee (10110001) = (10111101)$$

如果采用3，则新的冲突向量为：

$$(00010110) \vee (10110001) = (10110111)$$

如果采用4，则新的冲突向量为：

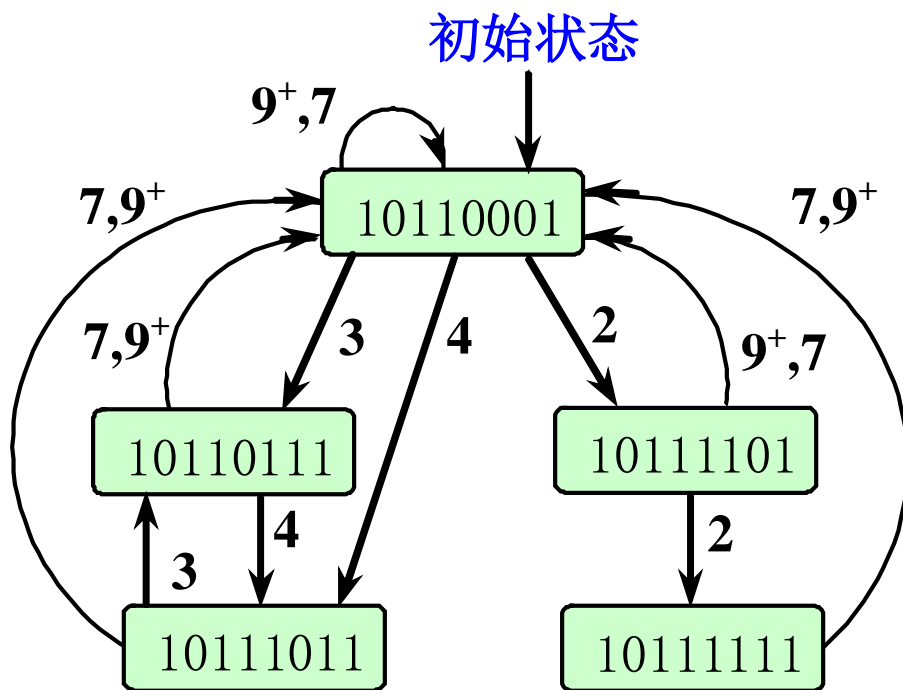
$$(00001011) \vee (10110001) = (10111011)$$

如果采用7，则新的冲突向量为：

$$(00000001) \vee (10110001) = (10110001)$$

3.3 非线性流水线的调度

- (2) 对于新向量 (10111101) ，其可用的时间间隔为2个和7个时钟周期。用类似上面的方法，可以求出其后续的冲突向量分别为 (10111101) 和 (10110001) 。
- (3) 对于其他新向量，也照此处理。
- (4) 在此基础上，画出状态转移示意图。



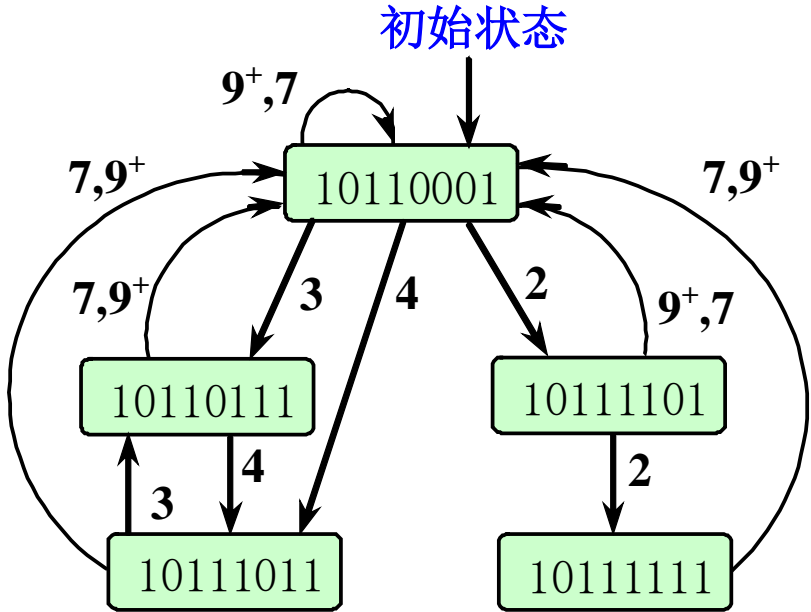
3.3 非线性流水线的调度

4. 根据状态转换图写出最优调度方案

- 根据流水线状态图，由初始状态出发，任何一个闭合回路即为一种调度方案。
- 列出所有可能的调度方案，计算出每种方案的平均时间间隔，从中找出其最小者即为最优调度方案。

3.3 非线性流水线的调度

各种调度策略及平均时间间隔（延迟拍数）



调度策略	平均时间间隔
(2,7)	4.5
(2,2,7)	3.67
(3,7)	5
(3,4)	3.5
(3,4,3,7)	4.25
(3,4,7)	4.67
(4,3,7)	4.67
(4,7)	5.5
(7)	7

3.3 非线性流水线的调度

- 上例中，各种调度方案及其平均间隔时间。
 - ❑ 最佳方案：(3, 4)
 - ❑ 平均间隔时间：3.5个时钟周期（吞吐率最高）
 - ❑ 方案(4, 3)的平均间隔时间也是3.5，但它不是最佳方案，为什么？
- 方案(3, 4)是一种不等时间间隔的调度方案，与等间隔的调度方案相比，在控制上要复杂得多。为了简化控制，也可以采用等间隔时间的调度方案，但吞吐率和效率往往会下降不少。
 - ❑ 在上述例子中，等时间间隔的方案只有一个：(7)，其吞吐率下降了一半。

3.3 非线性流水线的调度

3.3.2 多功能非线性流水线的调度

以双功能（功能A和B）非线性流水线为例。

1. 状态转移图中结点状态的表示

- 由两个冲突向量构成的冲突矩阵，这两个冲突向量分别对应于下一个任务的功能是A类和B类的情况。

2. 初始结点有两个，分别对应于第一个任务是A类和B类的情况。

- 当第一个任务是A类时，冲突矩阵为 $M^{(0)}_A$ 。
- 当第一个任务是B类时，冲突矩阵为 $M^{(0)}_B$ 。

3.3 非线性流水线的调度

$$M_A^{(0)} = \begin{bmatrix} C_{AA} \\ C_{AB} \end{bmatrix} \quad M_B^{(0)} = \begin{bmatrix} C_{BA} \\ C_{BB} \end{bmatrix}$$

其中：

- C_{pq} ($p, q \in \{A, B\}$) 表示的是：在一个 p 类任务流入流水线后，对后续 q 类任务的冲突向量。
它们可以由预约表求得。
- C_{pq} 共有 $2^2=4$ 个。
- 对于 N 功能流水线，这种冲突向量有 N^2 个。

3.3 非线性流水线的调度

3. 后续状态的冲突矩阵

$$\text{SHR}^{(i)}(M_k) \vee M_r^{(0)}$$

其中：

- M_k ：当前状态
- r ：下一个流入任务的类型（A或B）
- i ：当前状态允许的流入 r 型任务的时间间隔
- $\text{SHR}^{(i)}(M_k)$ ：把当前状态中的各冲突向量逻辑右移 i 位

例如： $\text{SHR}^{(3)}(M_k) \vee M_A^{(0)}$ 表示的是：把当前状态 M_k 中的各冲突向量逻辑右移3位，再与初始矩阵 $M_A^{(0)}$ 进行“或”运算。

3.3 非线性流水线的调度

4. 双功能非线性流水线的最优调度例

例3.3 有一条3段双功能非线性流水线，实现的功能是A和B，其预约表分别如表1和表2所示。各段的通过时间都是一个时钟周期。请找出该流水线单独处理A类任务和单独处理B类任务以及混合处理两类任务的最优调度方案。

表1 A类对象预约表

段 \ 时间	1	2	3	4	5
S1	√			√	
S2		√			
S3			√		√

表2 B类对象预约表

段 \ 时间	1	2	3	4	5
S1		√			√
S2				√	
S3	√		√		

3.3 非线性流水线的调度

解：（1）把两个预约表重叠起来，得到如表3所示的预约表。

段 \ 时间	1	2	3	4	5
S1	A	B		A	B
S2		A		B	
S3	B		AB		A

（2）由预约表求初始冲突向量和初始冲突矩阵

$$M_A^{(0)} = \begin{bmatrix} C_{AA} \\ C_{AB} \end{bmatrix}$$

$$M_B^{(0)} = \begin{bmatrix} C_{BA} \\ C_{BB} \end{bmatrix}$$

3.3 非线性流水线的调度

表1 A类对象预约表

段 \ 时间	1	2	3	4	5
S1	√			√	
S2		√			
S3			√		√

表2 B类对象预约表

段 \ 时间	1	2	3	4	5
S1		√			√
S2				√	
S3	√		√		

- C_{AA} : 一个A类任务流入流水线后, 对下一个A类任务进入流水线的
的时间间隔的限制。

由预约表表1可知, 禁用时间间隔是2和3, 故禁止表为: {2, 3},
所以 $C_{AA} = (0110)$ 。

- C_{BB} : 一个B类任务流入流水线后, 对下一个B类任务进入流水线的
的时间间隔的限制。

由预约表表2可知, 禁用时间间隔是2和3, 所以 $C_{BB} = (0110)$ 。

3.3 非线性流水线的调度

段 \ 时间	1	2	3	4	5
S1	A	B		A	B
S2		A		B	
S3	B		AB		A

- C_{AB} : 一个A类任务流入流水线后, 对下一个B类任务进入流水线的
时间间隔的限制。(冲突: B没完, A到了) 由预约表3:
 - 为了避免在S1发生冲突, 禁用时间间隔是: $4-2=2$
 - 在S2, 不会发生冲突, 这是因为根据表3, A类任务先于B类
任务通过S2, 而现在的实际情况又是A类任务先于B类任务
流入流水线;
 - 为了避免在S3发生冲突, 禁用时间间隔是:
 $3-1=2, 5-1=4, 5-3=2$
 - 综合起来, 有: $C_{AB} = (1010)$

3.3 非线性流水线的调度

段 \ 时间	1	2	3	4	5
S1	A	B		A	B
S2		A		B	
S3	B		AB		A

- C_{BA} 表示一个B类任务流入流水线后，对下一个A类任务进入流水线的时间间隔的限制。（冲突：A没完，B到了）由预约表3：
 - 为了避免在S1发生冲突，禁用时间间隔有：
 $2-1=1, 5-1=4, 5-4=1$
 - 为了避免在S2发生冲突，禁用时间间隔是： $4-2=2$
 - 在S3，不会发生冲突，B类任务先于A类任务通过S3，或者同时通过S3（第3个时钟周期），而现在的实际情况又是B类任务先于A类任务流入流水线。
 - 综合起来，有： $C_{AB} = (1011)$

3.3 非线性流水线的调度

■ 初始矩阵为:

$$M_A^{(0)} = \begin{bmatrix} C_{AA} \\ C_{AB} \end{bmatrix} = \begin{bmatrix} 0110 \\ 1010 \end{bmatrix}$$

$$M_B^{(0)} = \begin{bmatrix} C_{BA} \\ C_{BB} \end{bmatrix} = \begin{bmatrix} 1011 \\ 0110 \end{bmatrix}$$

3.3 非线性流水线的调度

(3) 由初始冲突矩阵画出状态图

如果第一个流入的任务是A类，初始状态就是 $M_A^{(0)}$ ；如果第一个流入的任务是B类，则初始状态是 $M_B^{(0)}$ 。

求所有后续状态的冲突矩阵：

例如：在流入一个A类任务后，从 $C_{AA} = (0110)$ 可知，可以隔一个或4个时钟周期再流入一个A类任务。

假设是前者，则把初始状态 $M_A^{(0)}$ 中的各冲突向量同时右移一位（即进行 $SHR^{(1)}$ 操作），再与 $M_A^{(0)}$ 进行或运算，可以得到新的冲突矩阵。根据该矩阵的第一个冲突向量 (0111) 可知，只有隔4个时钟周期流入一个A类任务，才能不发生冲突。把冲突矩阵中的各冲突向量同时右移4位（即进行 $SHR^{(4)}$ 操作），再与 $M_A^{(0)}$ 进行按位或运算，可以得到新的冲突矩阵。

3.3 非线性流水线的调度

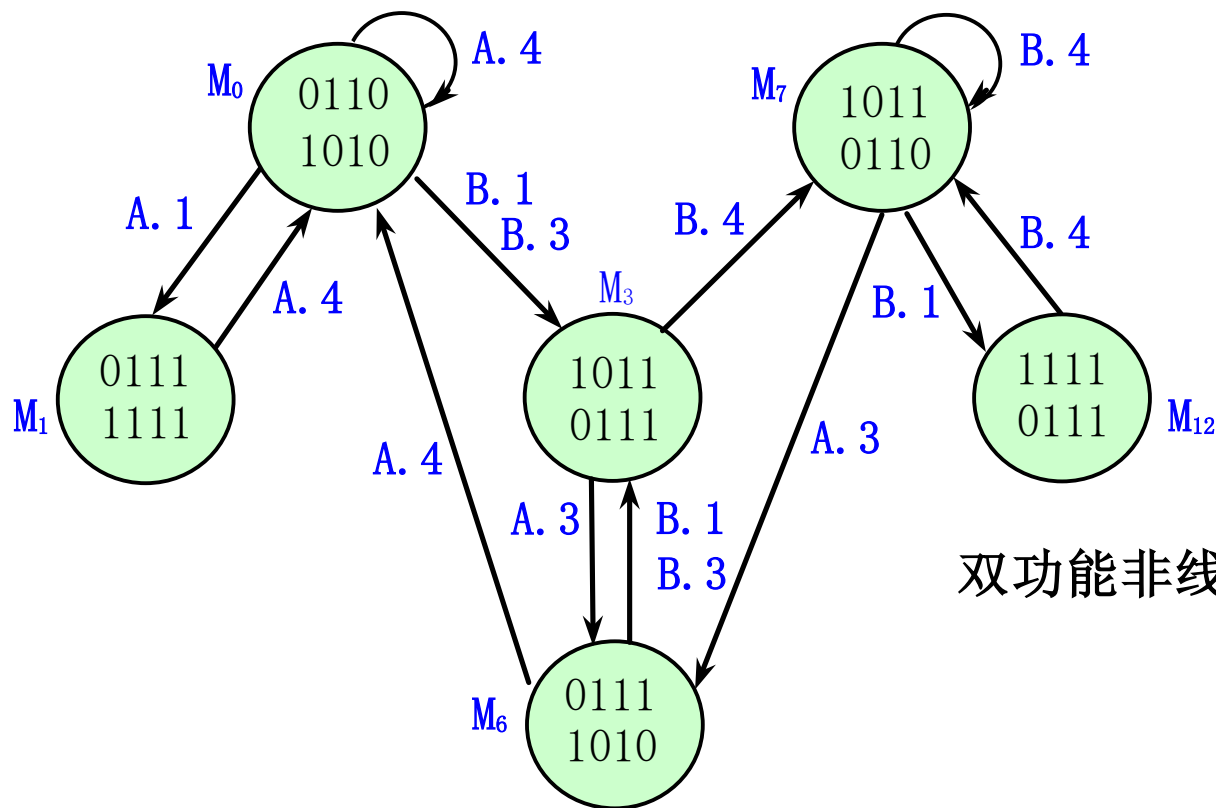
再如，在流入一个A类任务后，从 $C_{AB} = (1010)$ 可知，可以隔一个或3个时钟周期再流入一个B类任务。假设是前者，则把初始状态 $M_A^{(0)}$ 中的各冲突向量同时右移一位（即进行SHR⁽¹⁾操作），再与 $M_B^{(0)}$ 进行或运算，可以得到新的冲突矩阵。

据此可知，允许的流入为：或者是隔3个时钟周期流入一个A类任务，或者是隔4个时钟周期流入一个B类任务。按类似与上述类似的方法，又可以得到新的冲突矩阵。

求出所有可能的状态后，就可以画出状态图。

弧线上的标记“ $r.i$ ”表示：隔 i 个时钟周期流入 r 类的任务。

3.3 非线性流水线的调度



双功能非线性流水线的状态图

(4) 由状态图得出最优调度方案

- ◆ 只流入A类任务的最优调度方案是： (A.1, A.4)
- ◆ 只流入B类任务的最优调度方案是： (B.1, B.4)
- ◆ 混合流入A、B两类任务的最优调度方案是： (B.1, A.3, A.4)