第 3 章 流水线技术

内容提要

1. 流水线的基本概念
2. 流水线的性能指标
3. 非线性流水线的调度
4. 流水线的相关与冲突
5. 流水线的实现(以 MIPS 为例)

3.1 流水线的基本概念

3.1.1 什么是流水线

工业上的流水线大家一定都很熟悉了,例如汽车装配生产流水线等。在这样的流水线中,整个装配过程被分为多道工序,每道工序由一个人或多人完成,各道工序所花的时间也差不多。整条流水线流动起来后,每隔一定的时间间隔(差不多就是一道工序的时间)就有一辆汽车下线。如果我们跟踪装配一辆汽车的全过程,会发现其总的装配时间并没有缩短,但由于多辆车的装配在时间上错开后,重叠进行,因此最终能达到总体装配速度(吞吐率)的提高。

在计算机中也可以采用类似的方法,把一个重复的过程分解为若干个子过程(相当于上面的工序),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他的子过程并行进行。这就是**流水线技术**(pipelining)。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)。

把流水线技术应用于指令的解释执行过程,就形成了指令流水线。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。图 3.1 是一条浮点加法流水线,它把执行过程分解为求阶差、对阶、尾数相加、规格化 4 个子过程,每一个子过程在各自独立的部件上完成。如果各段的时间相等,都是 Δt,那么,虽然完成一次浮点加法所需要的总时间(从"入"到"出")还是 $4\Delta t$,但若在输入端连续送入加法任务,则从加法器的输出端来看,却是每隔一个 $\triangle t$ 就能出一个浮点加法结果。因此,该流水线能把浮点加法运算的速度提高 3 倍。

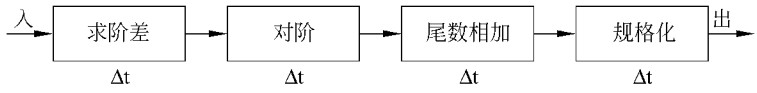


图 3.1　浮点加法流水线

一般采用时空图来描述流水线的工作过程。图 3.2 是上述 4 段流水线的时空图。图中横坐标表示时间，纵坐标表示空间，即流水线中的流水段。格子中的数字 1 代表第 1 个运算，2 代表第 2 个运算，…。第 1 个运算在时刻 0 进入流水线；第 2 个运算在时刻 1 进入流水线，同时第 1 个运算离开“求阶差”段而进入“对阶”段；第 3 个运算在时刻 2 进入流水线，同时第 1 个运算离开“对阶”段而进入“尾数相加”段，第 2 个运算离开“求阶差”段而进入“对阶”段；第 4 个运算在时刻 3 进入流水线，同时第 1 个运算离开“尾数相加”段而进入“规格化”段，第 2 个运算离开“对阶”段而进入“尾数相加”段，第 3 个运算离开“求阶差”段而进入“对阶”段；以此类推。

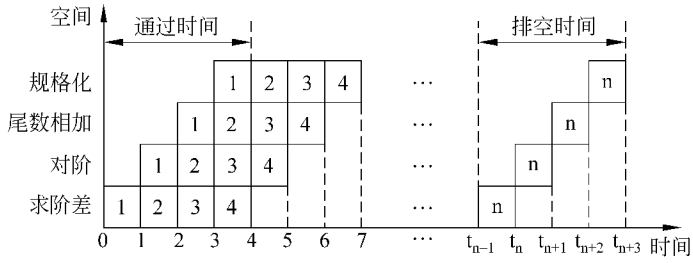


图 3.2　浮点加法流水线的时空图

从上面的分析可以看出，流水技术有以下特点：

(1) 流水线把一个处理过程分解为若干个子过程，每个子过程由一个专门的功能部件来实现。因此，流水线实际上是把一个大的处理功能部件分解为多个独立的功能部件，并依靠它们的并行工作来提高处理速度(吞吐率)。

(2) 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和断流，因为时间最长的段将成为**流水线的瓶颈**(Bottleneck of a Pipeline)，此时流水线中的其他功能部件就不能充分发挥作用。因此瓶颈问题是流水线设计中必须解决的。

(3) 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**。其作用是在相邻的两段之间传送数据，以提供后面流水段要用到的信息。其另一个作用是隔离各段的处理工作，避免相邻流水段电路的相互干扰。

(4) 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。

(5) 流水线需要有**通过时间**和**排空时间**。它们分别是指第一个任务和最后一个任务从进入流水线到流出结果的那个时间段，如图 3.2 所示。在这两个时间段中，流水线都不是满负荷。经过“通过时间”后，流水线进入满载工作状态，整条流水线的效率才能得到充分发挥。

3.1.2 流水线的分类

流水线可以从不同的角度和观点来分类。下面是几种常见的分类。

1. 部件级、处理机级及处理机间流水线

按照流水技术用于计算机系统的等级不同，可以把流水线分为3种：部件级流水线、处理机级流水线和系统级流水线。

部件级流水线是把处理机中的部件进行分段，再把这些分段相互连接而成。它使得运算操作能够按流水方式进行。图3.1中的浮点加法流水线就是一个典型的例子。这种流水线也称为运算操作流水线(Arithmetic Pipeline)。

处理机级流水线又称**指令流水线**(Instruction Pipeline)。它是把指令的执行过程按照流水方式进行处理，即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。3.4.1节中论述的5段指令流水线就是一个例子，它能同时重叠执行5条指令。

系统级流水线是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。前一台处理机的输出结果存入存储器中，作为后一台处理机的输入。这种流水线又称宏流水线(Macro Pipeline)。

2. 单功能流水线与多功能流水线

这是按照流水线所完成的功能来分类。

(1) 单功能流水线

单功能流水线(Unifunction Pipeline)是指流水线的各段之间的连接固定不变、只能完成一种固定功能的流水线。如前面介绍的浮点加法流水线就是单功能流水线。若要完成多种功能，可采用多条单功能流水线。例如Cray-1巨型机有12条单功能流水线。

(2) 多功能流水线

多功能流水线(Multifunction Pipeline)是指各段可以进行不同的连接，以实现不同的功能的流水线。美国TI公司ASC处理机中采用的运算流水线就是多功能流水线，它有8个功能段，按不同的连接可以实现浮点加减法运算和定点乘法运算，如图3.3所示。

3. 静态流水线与动态流水线

多功能流水线可以进一步分为静态流水线和动态流水线两种。

(1) 静态流水线

静态流水线(Static Pipelines)是指在同一时间内，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的流水线。当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。例如，上述ASC的8段只能或者按浮点加减运算连接方式工作，或者按定点乘运算连接方式工作。在图3.4中，当要在n个浮点加法后面进行定点乘法时，必须等最后一个浮点加法做完、流水线排空后，才能改变连接，开始新的运算。

(2) 动态流水线

动态流水线(Dynamic Pipelines)是指在同一时间内，多功能流水线中的各段可以按照不同的方式连接，同时执行多种功能的流水线。它允许在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。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。动态流水线的优点是：更加灵活，能提高各段的使用率，能提高处理速度。但其控制复

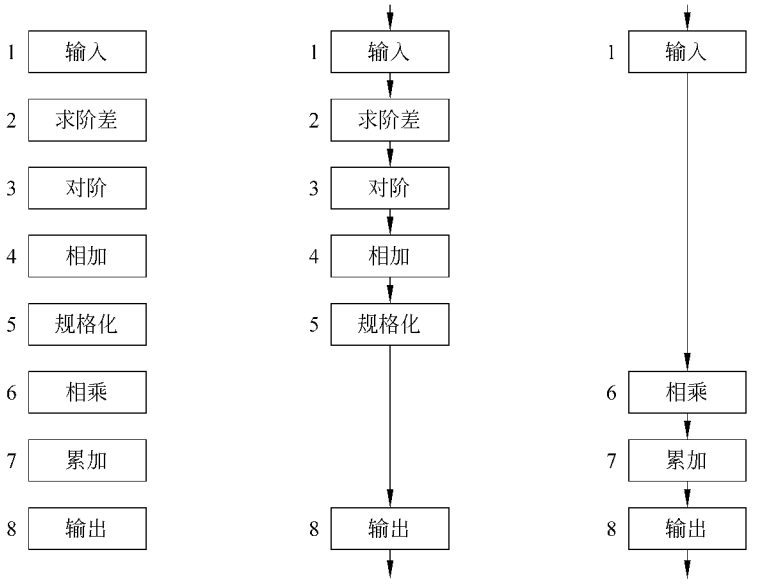


图 3.3　ASC 处理机的多功能流水线

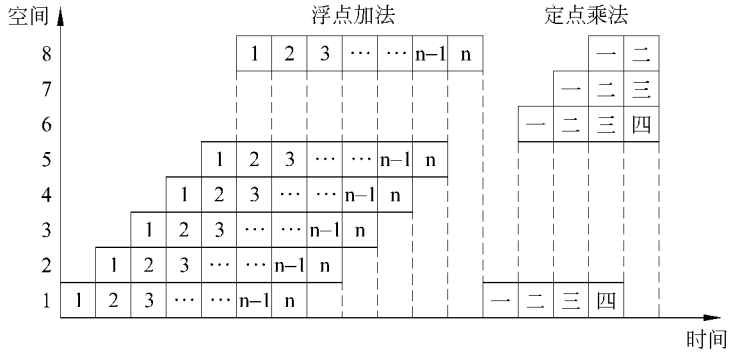


图 3.4　静态流水线的时空图

杂度增加了。

对于图 3.3 的情况，动态流水线的工作过程如图 3.5 所示。这里，定点乘法提前开始了（相对于静态流水线而言）。可以提前多少取决于任务的流动情况，要保证不能在公用段发生冲突。

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。如果交替输入不同的运算任务，则流水线的效率会降低到和顺序处理方式的一样。而动态流水线则不同，它允许多种运算在同一条流水线中同时进行。因此，在一般情况下，动态流水线的效率比静态流水线的高。但是，动态流水线的控制要复杂得多。所以目前大多数的流水线是静态流水线。

4. 线性流水线与非线性流水线

按照流水线中是否存在反馈回路，可以把流水线分为以下两类。

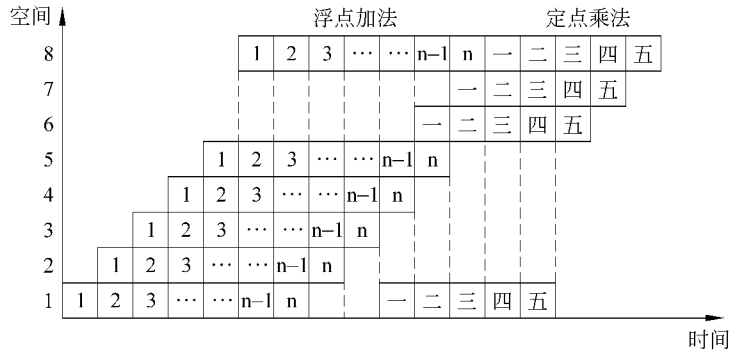


图 3.5　动态流水线的时空图

(1) 线性流水线

线性流水线(Linear Pipeline)是指各段串行连接、没有反馈回路的流水线。数据通过流水线中的各段时,每一个段最多只流过一次。

(2) 非线性流水线

非线性流水线(Nonlinear Pipeline)是指各段除了有串行的连接外,还有反馈回路的流水线。图 3.6 是一个非线性流水线的示意图。它由 4 段组成,经反馈回路和多路开关使某些段要多次通过。S_3 的输出可以反馈到 S_2,而 S_4 的输出可以反馈到 S_1。

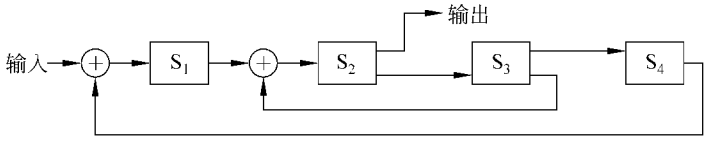


图 3.6　非线性流水线举例

非线性流水线常用于递归或组成多功能流水线。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任务,才能使该任务不会与先前进入流水线的任务发生争用流水段的冲突。这就是所谓的非线性流水线的调度问题,3.3 节将详细讨论这个问题。

5. 顺序流水线与乱序流水线

根据流水线中任务流入和流出的顺序是否相同,可以把流水线分为以下两种:

(1) 顺序流水线

在顺序流水线(in-order pipeline)中,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。

(2) 乱序流水线

在乱序流水线(out-of-order pipeline)中,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。这种流水线又称为无序流水线、错序流水线、异步流水线。

通常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。如果处理机具有向量数据表示和向量指令,则称之为向量流水处理机,简称向量机;否则就称之为标量流水处理机。

3.2 流水线的性能指标

衡量流水线性能的主要指标有吞吐率、加速比和效率。

3.2.1 流水线的吞吐率

流水线的**吞吐率** TP(throughput)是指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。

$$TP = \frac{n}{T_K} \tag{3.1}$$

其中 n 为任务数，T_k 是处理完 n 个任务所用的时间。该式是计算流水线吞吐率最基本的公式。

1. 各段时间均相等的流水线

图 3.7 是各段时间均相等(都是 Δt)的线性流水线的时空图。这里假设段数为 k，连续输入 n 个任务。第一个任务输入后，经过 kΔt 的时间从输出端流出(完成)。此后的 n－1 个 Δt 中，每个 Δt 时间完成一个任务。在这种情况下，流水线完成 n 个连续任务所需要的总时间为：

$$T_k = k\Delta t + (n-1)\Delta t = (k+n-1)\Delta t \tag{3.2}$$

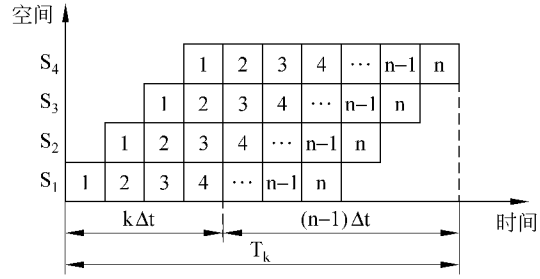


图 3.7 各段时间相等的流水线时空图

将式(3.2)代入式(3.1)，得流水线的实际吞吐率为：

$$TP = \frac{n}{(k+n-1)\Delta t} \tag{3.3}$$

这种情况下的最大吞吐率为：

$$TP_{max} = \lim_{n\to\infty} \frac{n}{(k+n-1)\Delta t} = \frac{1}{\Delta t} \tag{3.4}$$

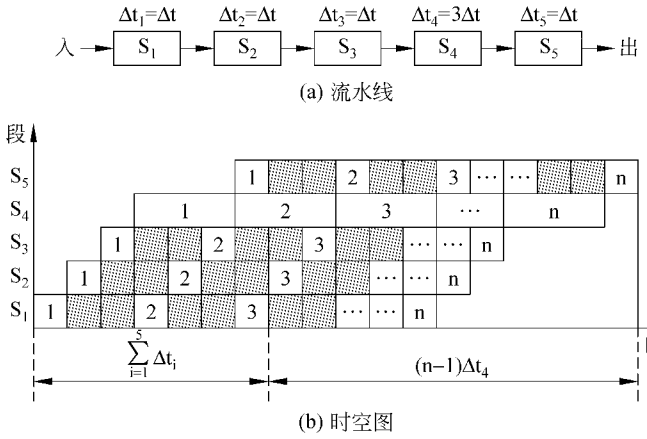
最大吞吐率与实际吞吐率的关系是：

$$TP = \frac{n}{k+n-1} TP_{max} \tag{3.5}$$

从式(3.5)可以看出，流水线的实际吞吐率总是小于最大吞吐率，它除了与每个段的时间有关外，还与流水线的段数 k 和输入到流水线中的任务数 n 有关。只有当 $n \gg k$ 时，才有 $TP \approx TP_{max}$。

2. 各段时间不完全相等的流水线

在图 3.8(a)所示的流水线中，各段时间不完全相等。其中 S_1，S_2，S_3，S_5 各段的时间都是 Δt，S_4 的时间是 $3\Delta t$，是其他各段时间的 3 倍。S_4 是该流水线的瓶颈段。除了第一个任务外，其余(n－1)个任务必须按瓶颈段的时间间隔 $\max(\Delta t_1, \Delta t_2, \cdots, \Delta t_k)$ 连续流入流水线。图 3.8(b)是该流水线的时空图，图中的灰色方格表示相应流水段在这一段时间内是空闲的。



(a) 流水线

(b) 时空图

图 3.8　各段时间不等的流水线及其时空图

一般地，各段时间不等的流水线的实际吞吐率为：

$$TP = \frac{n}{\sum_{i=1}^{k} \Delta t_i + (n-1)\max(\Delta t_1, \Delta t_2, \cdots, \Delta t_k)} \tag{3.6}$$

其中 Δt_i 为第 i 段的时间，共有 k 个段。分母中的第一部分是流水线完成第一个任务所用的时间，第二部分是完成其余 n－1 个任务所用的时间。

流水线的最大吞吐率为：

$$TP_{max} = \frac{1}{\max(\Delta t_1, \Delta t_2, \cdots, \Delta t_k)} \tag{3.7}$$

对于图 3.8 的例子，最大吞吐率为：

$$TP_{max} = \frac{1}{3\Delta t} \tag{3.8}$$

从式(3.6)和式(3.7)可以看出，当流水线各段的时间不完全相等时，流水线的最大吞吐率和实际吞吐率由时间最长的那个段决定，这个段就成了整条流水线的瓶颈。这时，瓶颈段一直处于忙碌状态，而其余各段则在许多时间内都是空闲的，硬件使用效率低。

可以用下面的两种方法来消除瓶颈段。

(1) 细分瓶颈段

这是把流水线中的瓶颈段切分为几个独立的功能段，从而使流水线各段的处理时间都相等。在图 3.9 中，把瓶颈段 S_4 细分为 3 个子流水段：$S_{4\text{-}1}$，$S_{4\text{-}2}$，$S_{4\text{-}3}$。这样，所产生的流水线的各段时间均为 Δt，每隔 Δt 流出一个结果。

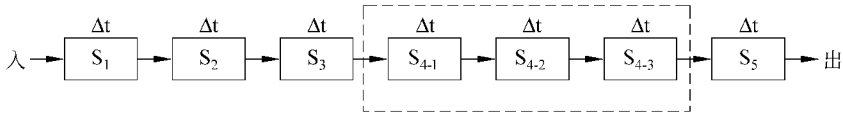
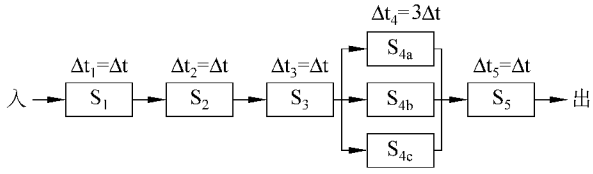


图 3.9　细分瓶颈段

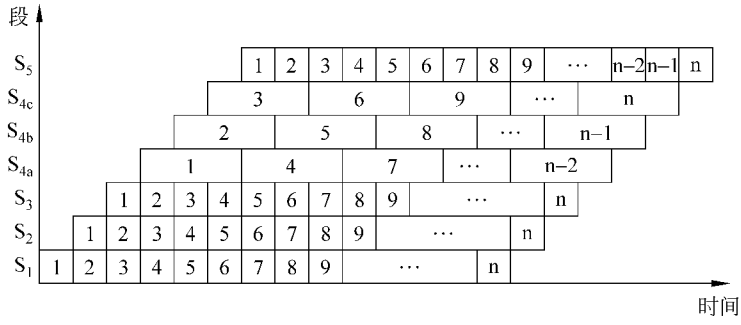
(2) 重复设置瓶颈段

如果无法把瓶颈段再细分，则可以采用重复设置瓶颈段的方法来解决问题。重复设置的段并行工作，在时间上依次错开处理任务。这种方法的缺点是控制逻辑比较复杂，所需要的硬件也增加了。

图 3.10 给出了把 S_4 重复设置后的流水线及时空图。这里，从 S_3 到并列的 S_{4a}，S_{4b}，S_{4c} 之间需要设置一个数据分配器，它把从 S_3 输出的第一个任务分配给 S_{4a}，第二个任务分配给 S_{4b}，第三个任务分配给 S_{4c}，之后以此重复。而在 S_{4a}，S_{4b}，S_{4c} 到 S_5 之间需要设置一个数据收集器，依次分时将数据收集到 S_5 中。改进后的流水线能做到每隔 Δt 流出一个结果。



(a) 重复设置瓶颈段



(b) 对应的流水线时空图

图 3.10　重复设置瓶颈段

对于图 3.8 的例子，这两种方法都能使改进后的流水线的吞吐率达到：

$$TP_{max} = \frac{1}{\Delta t} \tag{3.9}$$

3.2.2　流水线的加速比

流水线的加速比(speedup)是指使用顺序处理方式处理一批任务所用的时间与流水线使用流水处理方式处理同一批任务所用的时间之比。设顺序执行所用的时间为 T_s，按流水线方式处理所用的时间为 T_k，则流水线的加速比为：

$$S = \frac{T_s}{T_k} \tag{3.10}$$

假设流水线各段时间都是 Δt，则一条 k 段流水线完成 n 个连续任务所需要的时间为：$T_k=(k+n-1)\Delta t$。而这 n 个任务若是顺序执行，则所需要的时间为：$T_s=nk\Delta t$。代入式(3.10)，得流水线的实际加速比为：

$$S=\frac{nk}{k+n-1} \tag{3.11}$$

这种情况下的最大加速比为：

$$S_{max}=\lim_{n\to\infty}\frac{nk}{k+n-1}=k \tag{3.12}$$

即当 $n\gg k$ 时，流水线的加速比等于流水线的段数。从这个意义上看，流水线的段数愈多愈好，但这会给流水线的设计带来许多问题，对此后面将作进一步讨论。

当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：

$$S=\frac{n\sum_{i=1}^{k}\Delta t_i}{\sum_{i=1}^{k}\Delta t_i+(n-1)\max(\Delta t_1,\Delta t_2,\cdots,\Delta t_k)} \tag{3.13}$$

3.2.3 流水线的效率

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

如果各段时间相等，如图 3.7 那样，则各段的效率 e_i 是相同的。

$$e_1=e_2=\cdots=e_k=\frac{n\Delta t}{T_k}=\frac{n}{k+n-1} \tag{3.14}$$

整条流水线的效率为：

$$E=\frac{e_1+e_2+\cdots+e_k}{k}=\frac{ke_1}{k}=\frac{kn\Delta t}{kT_k} \tag{3.15}$$

还可以写成：

$$E=\frac{n}{k+n-1} \tag{3.16}$$

最高效率为：

$$E_{max}=\lim_{n\to\infty}\frac{n}{k+n-1}=1 \tag{3.17}$$

显然，当 $n\gg k$ 时，流水线的效率接近最大值 1。这时流水线的各段均处于忙碌状态。

根据式(3.3)和式(3.16)，可得：

$$E=TP\cdot\Delta t \tag{3.18}$$

即当流水线各段时间相等时，流水线的效率与吞吐率成正比。

根据式(3.11)和式(3.16)，可得：

$$E=\frac{S}{k} \tag{3.19}$$

即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。只有当 E=1 时，

S=k，实际加速比达到最大。

式(3.15)中的分母 kT_k 实际上就是时空图中 k 个段和流水总时间 T_k 所围成的总面积，而分子 $kn\Delta t$ 则是时空图中 n 个任务实际占用的总面积。所以，从时空图上看，所谓效率，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。

虽然当流水线的各段时间不等时，各段的效率会不同，但同样也有上述的结论。因此，计算流水线效率的一般公式可以表示为：

$$E=\frac{\text{n 个任务实际占用的时空区的面积}}{\text{k 个段总的时空区的面积}} \tag{3.20}$$

画出流水线的时空图，然后根据上式来计算效率，是一种比较直观通用的方法。对于线性流水线、非线性流水线、多功能流水线、任务不连续的情况等都适用。

在各段时间不等的情况下，k 段流水线连续处理 n 个任务的流水线效率为：

$$E=\frac{n\sum_{i=1}^{k}\Delta t_i}{k\left[\sum_{i=1}^{k}\Delta t_i+(n-1)\cdot\max(\Delta t_1,\Delta t_2,\cdots,\Delta t_k)\right]} \tag{3.21}$$

3.2.4 流水线性能分析举例

例 3.1 要在图 3.3 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。

解 首先，应选择适合于流水线工作的算法。对于本题，应先计算 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。

其次，画出完成该计算的时空图，如图 3.11 所示，图中阴影部分表示相应的段在工作。

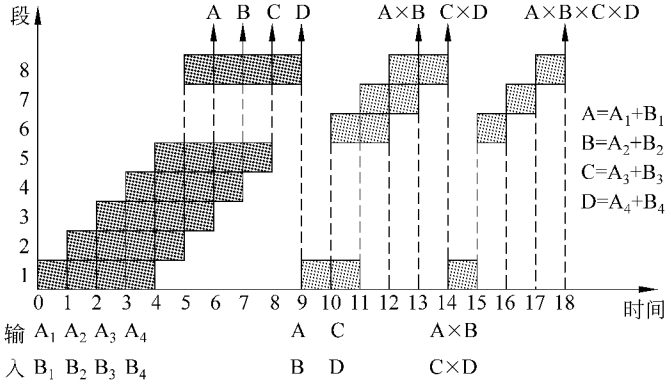


图 3.11 完成乘加运算的多功能静态流水线时空图

由图 3.11 可见，它在 18 个 Δt 时间中，给出了 7 个结果。所以吞吐率为：

$$TP=\frac{7}{18\Delta t}$$

如果不用流水线，由于一次求和需 6Δt，一次求积需 4Δt，则产生上述 7 个结果共需