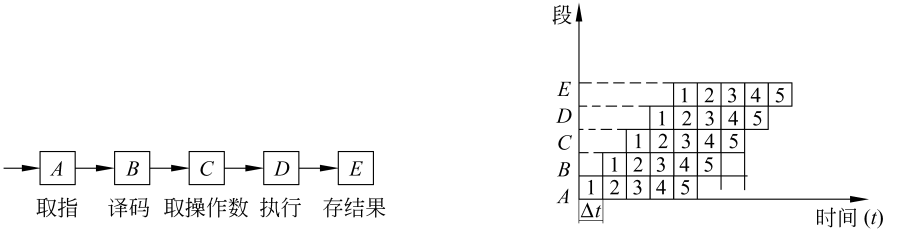
# 第 3 章　流水线处理机

本章介绍流水线的概念、结构，以及线性流水线的性能分析。

## 3.1　流水线的概念

**1. 什么是流水线**

把一条指令的执行过程分成几个子过程，例如，分成取指、译码、取操作数、执行和存放结果这 5 个子过程，这样，就可以用 5 个子部件来执行指令，分别处理 5 个子过程。设 5 个子部件分别为 $A$、$B$、$C$、$D$ 和 $E$，如图 3.1(a)所示。这样，下一条指令就没有必要等到上一条指令的 5 个子过程全部处理完毕后再进行处理，只要上一条指令的第一个子过程(取指)在 $A$ 部件上处理完毕，进入 $B$ 部件进行第二个子过程(译码)处理的同时，第二条指令即可进入 $A$ 部件处理其第一个子过程。诸如此类，如图 3.1(b)所示，每隔 $\Delta t$ 时间便可进入一条新指令。把各子部件看作是流水线上的各道工序，那么，每条指令就像要加工的产品，所以，人们把这种工作方式叫做流水线。




(a) 流水线结构　　(b) 流水线的时空图

图 3.1　流水线的结构和时空图

流水线结构中的各子部件也被称做功能段，简称段。图 3.1(b)反映了流水线执行指令过程中段与时间的关系，故叫时空图。

**2. 流水线的分类**

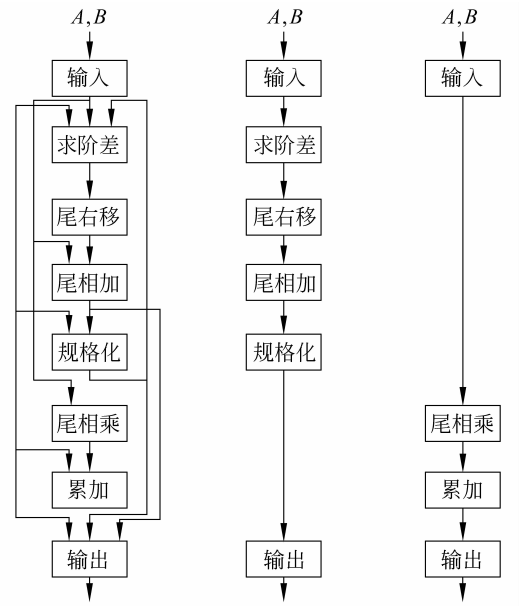
(1) 按功能多少分为两种。

① 单功能流水线(unifunction pipeline)　它是只能实现一种功能的流水线，例如，只能完成浮点加的浮点加流水线和只能完成浮点乘的浮点乘流水线。

使用单功能流水线的计算机，其多种功能是靠多条单功能流水线分别完成的，如 Cray-1 机具有 12 条单功能流水线；我国银河-1 机具有 18 条单功能流水线；Pentium 处理机有一条 5 段整型数运算流水线和一条 8 段的浮点数运算流水线。

② 多功能流水线(multifunction pipeline)　它是通过各种不同功能模块的不同连接组合而实现多种功能的流水线，如美国 Texas 公司生产的 TI ASC 机有 4 条流水线，每条都由

8 个功能模块组成，使用时可根据需要组合连接，实现各种功能，如图 3.2 所示。




(a) 8 功能流水线结构　(b) 做浮点加　(c) 做定点乘

图 3.2　TI ASC 机多功能流水线结构及功能

(2) 按是否为线性，可分为两种。

① 线性流水线(linear pipeline)　它是指每个功能模块只有一次数据通过的流水线。

② 非线性流水线(nonlinear pipeline)　它是指某些模块有反馈回路或前馈回路的流水线，如图 3.3 所示。

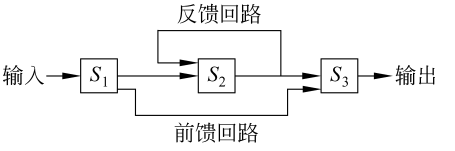



图 3.3　非线性流水线结构

非线性流水线与线性流水线的区别如下：

- 非线性流水线有反馈回路或前馈回路，或两者兼有。
- 非线性流水线执行一条指令时，各功能模块不是都执行一次。
- 表示非线性流水线除连接图外，还需有预约表一起来说明；线性流水线其预约表是确定的，而非线性流水线可对应多张预约表。
- 非线性流水线的输出端不一定是最后模块。

(3) 在多功能流水线中，按在同一时间内是否能实现多种功能来分又可分为两种。

① 静态流水线(static pipeline)　它是指在同一时间内，多功能流水线的各功能模块只能按一种固定方式连接，实现一种固定功能的流水线。其特点是，一个时刻只能有一种组合形式，如 TI ASC 机在做浮点加，应按图 3.2(b)组合；若浮点加没有执行完，运算器不能改为其他组合。优点是控制较简单，缺点是运算效率不高。

② 动态流水线(dynamic pipeline)　它是指同一时间内，多功能流水线的各功能模块可根据多种运算需要，连接成多种方式，同时能执行多种功能的流水线。其特点是，各功能模

块可以做到在同一时间内，某些功能模块正在实现某种运算（如浮点加）时，而另一些功能模块却在实现另一种运算（如定点乘）。优点是效率和功能模块利用率比静态流水线要高，缺点是相关控制复杂。

目前，采用静态流水线的处理机居多。

**3. 流水线级别**

（1）按功能部件分为 3 个级别。

① 处理机级流水线　它也叫指令流水线（instruction pipeline）。这种级别的流水线是把一条指令的执行过程，分为多个子过程，每个子过程在一个独立的功能部件中完成。如在所谓先行控制器中，一条指令的执行过程可分为 5 个子过程，如图 3.4 所示。

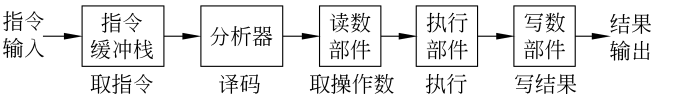



图 3.4　先行控制器流水线

② 功能部件级流水线　它也叫运算流水线（arithmetic pipeline），如加法流水线和乘法流水线等。

③ 多处理器流水线　它也叫宏流水线（macro pipeline）。这种流水线是由多个处理器通过存储器串行连接起来的流水线，如图 3.5 所示。

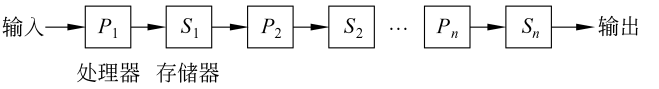



图 3.5　多处理器流水线示意图

在该流水线中，各个处理器对同一数据流的不同部分进行处理。前一个处理器的处理结果存入其后的存储器中，作为后一个处理器的输入数据。

（2）按并行度，分为以下几种。

① 超标量流水线（superscalar pipeline）　这是指一个时钟周期内可同时发射多条指令的流水线。可以说，超标量流水线采用的是空间上的并行性。

② 超流水线（super pipeline）　这是指一个时钟周期内可同时执行多条指令的流水线。超流水线采用的是时间上的并行性。

③ 超标量超流水线（superpipeline superscalar pipeline）　这是时间上与空间上均并行的流水线。

上述 3 种流水线与一般的单发射流水线的时空图如图 3.6 所示。

其中，图 3.6(a)表示单发射流水线执行 3 条指令（$I_1$，$I_2$ 和 $I_3$）的情况，每条指令的 4 个子过程（取指 IF、译码 ID、执行 EX 和写回 WR）都是在一个时钟周期内完成。单发射流水线的设计目标是每个时钟周期平均执行一条指令，即它的指令级并行度的期望值为 1。图 3.6(b)表示在一个时钟周期内能同时发射 3 条指令的超标量流水线。为实现这样的功能，该流水线的取指、译码、执行和写回这 4 个部件都应各有 3 套。图 3.6(c)表示一个时钟周期内可同时执行 3 条指令的超流水线。图 3.6(d)表示超标量超流水线。

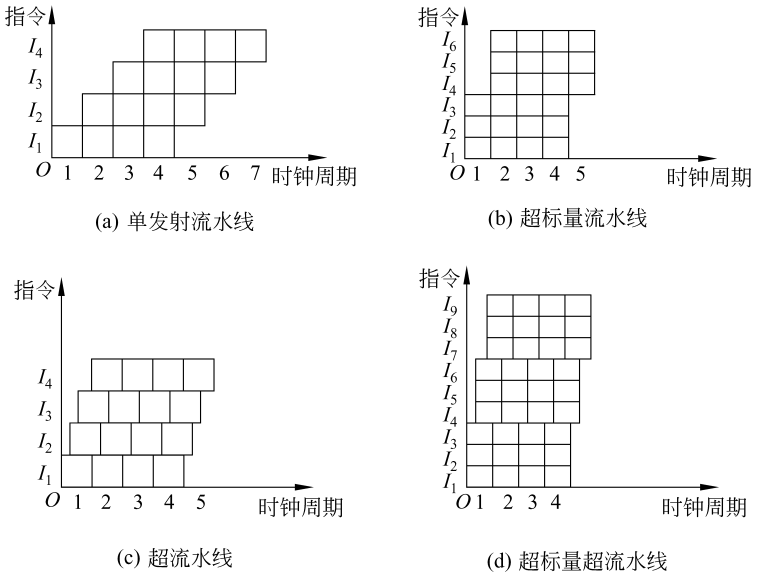



图 3.6　按并行度所分的 4 种流水线的时空图

## 3.2　流水线结构

这里介绍运算功能流水线和执行指令的两种流水线的结构。

**1. 运算流水线结构**

以浮点加和乘法流水线为例，介绍运算流水线的结构。

(1) 浮点加流水线　其结构框图如图 3.7 所示。

对该图做如下 5 点说明。

- 浮点加法分 3 步完成，即对阶、求尾和和规格化这 3 步。
- 流水线由 4 个功能段实现上述 3 步。4 个功能段分别是求阶差、尾右移、求尾和和规格化。
- 图中的 $R_i$ 为锁存器，作为本功能段的输出锁存和下一功能段的输入来源。
- 流水线中有几处要进行逻辑或运算。这几处是 $R_3$、右移位器、$R_6$ 和 $R_9$。
- 右移位器的工作原理是根据 $R_3$ 的内容移动尾数，即把尾数右移($R_3$)位，移出的部分存放在 $R_8$ 中。

(2) 乘法流水线　乘法运算归根结底是做加法运算，因此先介绍两种乘法流水线中所使用的加法器。

① 进位传送加法器(carry-propagation adder，CPA)　其功能是把两个 $n$ 位二进制数相加，产生算术和，同时产生一位向高位的进位，故又叫先行进位加法器。其符号图如图 3.8(a)所示。

图中，$A$、$B$ 是参加运算的两个 $n$ 位二进制数。例如，$n=4$，$A=1011$，$B=0111$，那么，该加法器的运算功能如下：

$$
\begin{array}{rl}
 & A=1011 \\
+ & B=0111 \\
\hline
 & S=10010=A+B \\
 & \blacktriangle \\
 & C_{out}
\end{array}
$$

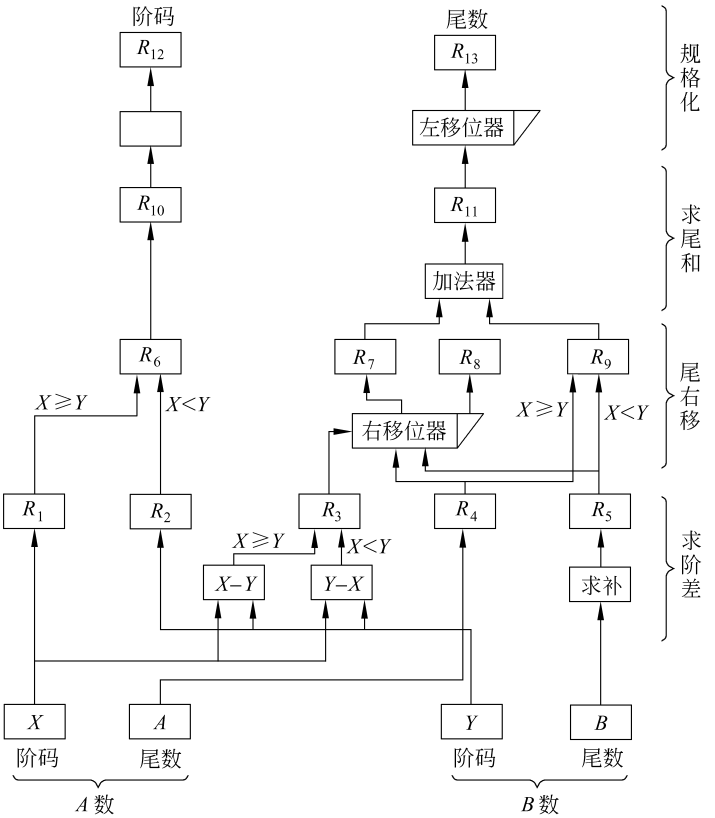
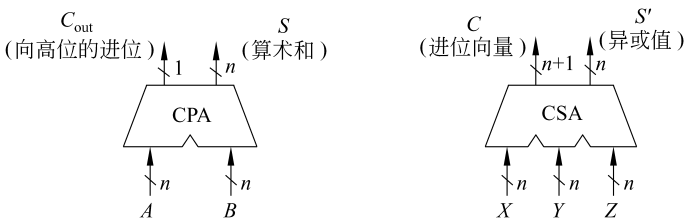



图 3.7　浮点加流水线的逻辑结构图




(a) 进位传送加法器　　(b) 进位保留加法器

图 3.8　两种加法器符号图

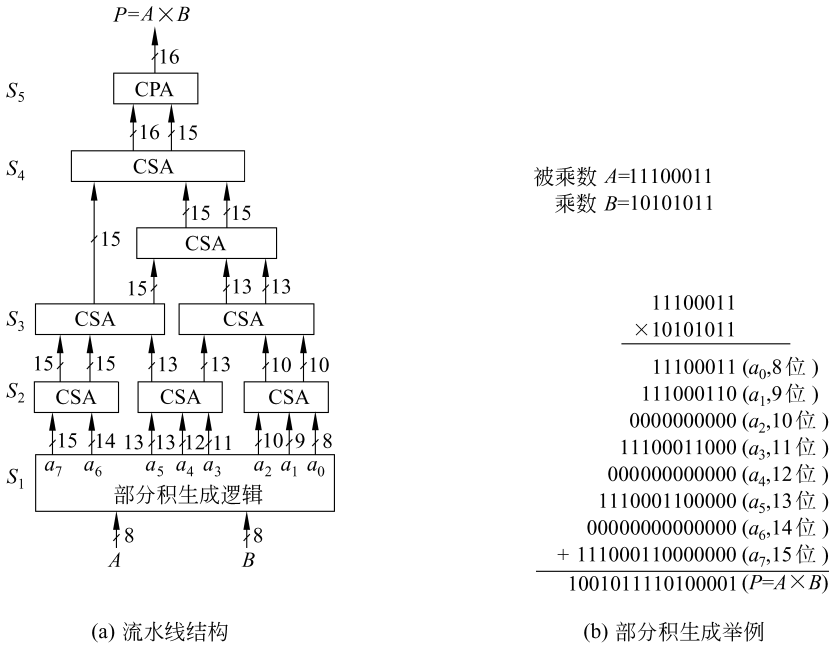
② 进位保留加法器(carry-save adder, CSA)　其功能是对 3 个 $n$ 位数进行异或运算；同时求出这 3 个数的进位向量，暂时保留；其算术和需下一级加法器求出，故又叫伪加法器。其符号图如图 3.8(b)所示。

图中，$X$、$Y$、$Z$ 为 3 个 $n$ 位二进制数，假如，$n=6$，$X=001011$，$Y=010101$，$Z=111101$，那

么，该加法器的运算功能如下：

$$
\begin{array}{rr}
 & X=001011 \\
 & Y=010101 \\
\oplus & Z=111101 \\
\hline
 & S'=100011 \\
+ & C=111010 \\
\hline
 & S=1011101=S'+C=X\oplus Y\oplus Z
\end{array}
$$

③ 乘法流水线　使用 CSA 和 CPA 就可以组成乘法流水线，如图 3.9 所示。




(a) 流水线结构　　(b) 部分积生成举例

图 3.9　乘法流水线的逻辑结构图

对流图做如下 6 点说明。

- 该流水线可实现两个 8 位二进制数相乘运算。
- 功能段 $S_1$ 产生 8 个部分积，分别是 8 位的 $a_0$，9 位的 $a_1$、10 位的 $a_2$、11 位的 $a_3$、12 位的 $a_4$、13 位的 $a_5$、14 位的 $a_6$ 和 15 位的 $a_7$。
- 功能段 $S_2$ 用 3 个 CSA 把 8 个部分积合并为 6 个数。
- 功能段 $S_3$ 用 2 个 CSA 把 6 个数合并为 4 个数。
- 功能段 $S_4$ 用 1 个 CSA 把 3 个数合并为 2 个数。
- 功能段 $S_5$ 用 1 个 CPA 产生一个 16 位的积。

**2. 执行指令流水线**

流水线处理机不仅运算部件是以流水线方式工作的，而且它的指令部件也是以流水线方式工作的。例如，一个七段执行指令流水线其组织如图 3.10 所示。

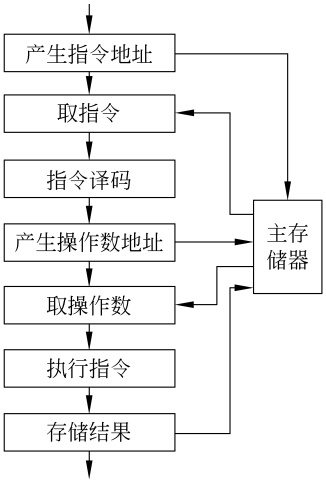



图 3.10　执行指令流水线示意图

## 3.3 线性流水线的性能分析

表示流水线性能的主要参数有吞吐率、加速比和效率，这里介绍这 3 个参数。

**1. 吞吐率(throughput rata，TP)与最佳段数**

(1) 定义　流水线的吞吐率是指其单位时间所处理的指令条数。

(2) 公式　利用流水线时空图很容易推出吞吐率的计算公式。$k$ 功能段流水线执行 $n$ 条指令的时空图如图 3.11 所示。

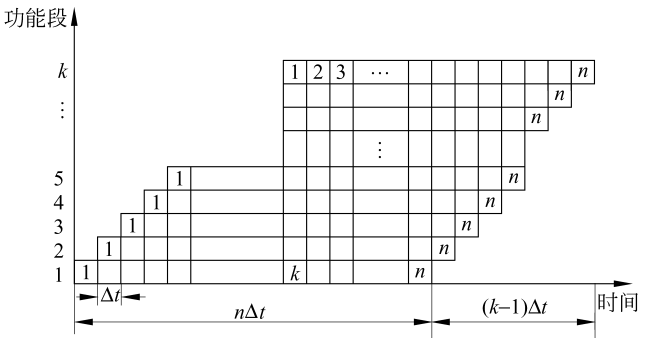



图 3.11　$k$ 段流水线的时空图

假定每个功能段的执行时间均为 $\Delta t$，从时空图不难看出，执行 $n$ 条指令的时间 $T_n$ 为

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

于是，吞吐率 $\mathrm{TP} = \dfrac{\text{指令条数}}{\text{执行时间}}$

$$= \frac{n}{(n+k-1)\Delta t} \tag{3.1}$$

(3) 吞吐率的最大值　当 $n >> k$ 时，可求出吞吐率的最大值 $\mathrm{TP}_{\max}$，即

$$\mathrm{TP}_{\max} = \lim_{n\to\infty} \frac{n}{(n+k-1)\Delta t}$$

$$= \lim_{n\to\infty} \frac{1}{\left(1 + \dfrac{k-1}{n}\right)\Delta t} = \frac{1}{\Delta t} \tag{3.2}$$

对于 $\mathrm{TP}_{\max}$，应该有如下认识。

① $\mathrm{TP}_{\max}$　实际上是实现不了的。主要原因如下：

- 流水线存在着指令上线时间和排空时间；
- 由于多种原因，流水线执行的任务很难做到连续进行。

一般认为 TP 值能达到 $\mathrm{TP}_{\max}$ 的 1/3 就算不错了。

② 从式(3.2)中可以看出，$\Delta t$ 的值越小，$\mathrm{TP}_{\max}$ 的值就越大，那么是不是 $\Delta t$ 越小越好呢？答案是否定的，这一点，可从 $\Delta t$ 与 $k$ 的关系式

$$\Delta t = \frac{\text{指令周期}}{k} \tag{3.3}$$

分析入手。

从该式可以看出，$\Delta t$ 的值变小，$k$ 值就要变大，即增加功能段的数目。而在流水线上，每个功能段的后面都必须设有一个锁存器；功能段的增加，就意味着锁存器数目的增加，这样，锁存器的总延迟时间就会增加，致使流水线执行速度下降。从另一方面看，锁存器个数的增加，会造成流水线成本的增加。因此，设计流水线时，分段不宜过多，要根据最佳性能价格比来选择最佳段数。

③ TP 的实际值　一般情况下，流水线中各功能段的执行时间是不会完全相等的。在这种情况下，流水线的吞吐率应按下式计算：

$$\mathrm{TP}=\frac{n}{\sum_{i=1}^{k}\Delta t_i+(n-1)\max\limits_{i=1\sim k}(\Delta t_i)} \tag{3.4}$$

式中，$\Delta t_1$、$\Delta t_2$、…、$\Delta t_k$ 分别为第一功能段至第 $k$ 功能段的执行时间，$\sum_{i=1}^{k}\Delta t_i$ 是流水线完成第 1 条指令所用时间，$(n-1)\max\limits_{i=1\sim k}(t_i)$是完成其余 $n-1$ 条指令所用时间。

这时，流水线的最大吞吐率为

$$\mathrm{TP}_{\max}=\frac{1}{\max\limits_{i=1\sim k}(\Delta t_i)} \tag{3.5}$$

(4) 最佳性能价格比和段数　设指令周期为 $t_0$，每个锁存器的延迟时间为 $d$，则流水线的最大吞吐率为

$$\mathrm{TP}'_{\max}=\frac{1}{t_0/k+d} \tag{3.6}$$

如果所有功能段的总成本为 $a$，$b$ 为每个锁存器的成本，那么流水线的总成本为

$$c=a+bk \tag{3.7}$$

这样，根据 A. G. Larson 的流水线性能价格比的定义，便有

$$\mathrm{PCR}=\frac{\mathrm{TP}'_{\max}}{c}=\frac{1}{t_0/k+d}\cdot\frac{1}{a+bk} \tag{3.8}$$

对此式求导，得到一个大于 0 的极值，该值就是最佳性能价格比。这时的 $k$ 值就是最佳段数 $k_0$，其值为

$$k_0=\sqrt{\frac{t_0 a}{db}} \tag{3.9}$$

目前，一般处理机的流水线功能段数为 2～10，极少有超过 15 段的流水线。有时把 8 段及其以上的流水线称为超流水线，而把具有这样流水线的处理机称为超流水线处理机。Pentium 4 就是这样的处理器，它具有 20 段的流水线。

(5) 提高吞吐率的方法　主要方法如下：

① 采用快速部件　如快速多位移位线路。

② 改进流水线结构，解决瓶颈(bottle neck)　流水线中执行时间最长的功能段就是影响流水线吞吐率提高的瓶颈，解决方法有以下两种。

- 分解瓶颈功能段　就是把瓶颈功能段分解为几个功能段，如图 3.12(b)所示。图中的方块表示流水线的功能段，其中的数字代表着功能段的序号，其下方为执行时间。原流水线的第 2 段为瓶颈，如图 3.12(a)所示。图 3.12(b)表示把瓶颈分解为 3 个

串行的功能段，去掉了瓶颈。

- 重复瓶颈功能段　如图 3.12(c)所示。其解决瓶颈的方法是，在 1 号功能段的后面设有数据分配器，负责把连续的 3 个任务分别分配给 2-1、2-2 和 2-3 这 3 个功能段，并行执行；而在 3 号功能段的前面设有数据收集器，其功能是依次从 2-1、2-2 和 2-3 这 3 个功能段收集它们的处理结果，并分时传送给 3 号功能段。

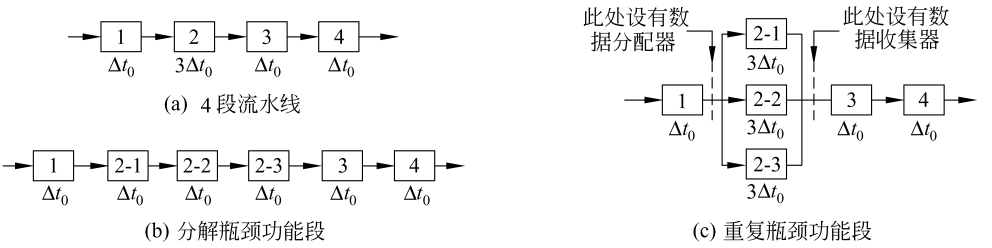



图 3.12　解决流水线瓶颈的方法

**2. 加速比(speedup)**

(1) 定义　顺序执行某任务所用时间 $T_0$ 与流水线执行所用时间 $T_k$ 之比，称做流水线的加速比，记作

$$S = \frac{T_0}{T_k} \tag{3.10}$$

(2) 公式　在各功能段执行时间均相等时，加速比的公式为

$$S = \frac{nk\Delta t}{(k+n-1)\Delta t} \tag{3.11}$$

式中，$k$ 为流水线的功能段数，$n$ 为指令条数，$\Delta t$ 为每功能段的执行时间。于是，$k\Delta t$ 就是指令周期，$nk\Delta t$ 为顺序执行 $n$ 条指令所需的时间；而 $(k+n-1)\Delta t$ 为 $k$ 段流水线执行 $n$ 条指令所需时间。

这时，加速比 $S$ 的最大值为

$$\begin{aligned} S_{\max} &= \lim_{n\to\infty} \frac{kn}{k+n-1} \\ &= \lim_{n\to\infty} \frac{k}{\frac{k}{n}+1-\frac{1}{n}} = k \end{aligned} \tag{3.12}$$

可见，当 $n>>k$，且流水线各功能段执行时间 $\Delta t$ 相等时，流水线的最大加速比为流水线的功能段数。

(3) 实际值　实际上，$n>>k$ 和各功能段执行时间 $\Delta t$ 相等这两个前提，在一般情况下，难以达到。这时，加速比的值为

$$S = \frac{n \cdot \sum_{i=1}^{k} \Delta t_i}{\sum_{i=1}^{k} \Delta t_i + (n-1) \cdot \max_{i=1\sim k}(\Delta t_i)} \tag{3.13}$$

式中，$\Delta t_i$ 为第 $i$ 功能段的执行时间，$\max_{i=1\sim k}(\Delta t_i)$ 是取各功能段执行时间的最大值。

**3. 效率(efficiency)**

(1) 定义　用来表示流水线的设备利用率，用时空值定义。即 $n$ 条指令所占用时空值

与 $k$ 个功能段总的时空值之比为流水线的效率，用 $E$ 来表示。

(2) 公式　实际上，$n$ 条指令所占用的时空值就是顺序执行 $n$ 条指令所使用的总时间 $T_0$，而一条 $k$ 段流水线完成 $n$ 条指令的总时空为 $k \cdot T_k$（$T_k$ 为流水线完成 $n$ 条指令所用的总时间），因此，一条 $k$ 段流水线的效率为

$$E = \frac{n \text{ 条指令占用的时空}}{k \text{ 段流水线完成 } n \text{ 条指令的总时空}} = \frac{T_0}{kT_k} \tag{3.14}$$

(3) 最大值　在流水线各段执行时间 $\Delta t$ 均相等，且 $n$ 条指令连续执行的情况下：

$$E = \frac{nk\Delta t}{k(k+n-1)\Delta t} = \frac{n}{k+n-1} \tag{3.15}$$

$E$ 的最大值为

$$E_{\max} = \lim_{n\to\infty} \frac{1}{\frac{k}{n}+1-\frac{1}{n}} = 1 \tag{3.16}$$

由此可见，当 $n>>k$，且各段 $\Delta t$ 相等时，流水线的效率达到了最大值 1。这时，流水线的各段均处于忙状态。从时空图上看，每一段都是有效的。

(4) 实际值　一般来说，流水线各段的执行时间均相等是很难做到的。当各功能段的执行时间不相等时，$E$ 的实际值为

$$E = \frac{n \cdot \sum_{i=1}^{k} \Delta t_i}{k\left(\sum_{i=1}^{k} \Delta t_i + (n-1) \cdot \max_{i=1\sim k}(\Delta t_i)\right)} \tag{3.17}$$

(5) $E$ 与 TP 和 $S$ 的关系　由以上所推出的公式可以看出，它们有如下关系：

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

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

**4. 流水线性能分析举例**

**【例 3.1】** 在一条单功能、线性 4 段浮点加流水线上，若实现 4 个浮点数相加，在输入任务不要求顺序累加的情况下，试计算 TP、$S$ 和 $E$。

**解**　假定该流水线 4 段的执行时间均相同，设为 $\Delta t$，则顺序累加 4 个浮点数（$A$、$B$、$C$ 和 $D$）的时空图如图 3.13 所示。

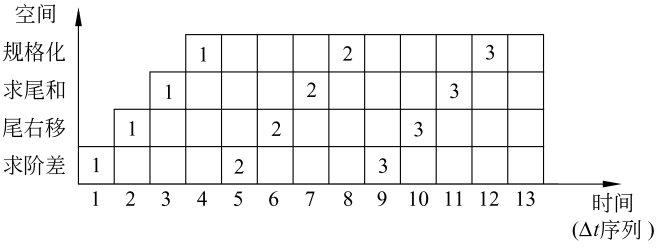



图 3.13　4 段浮点加流水线实现 4 个浮点数相加的时空图