

5.6 提高CPU性能的技术

5.6.1 流水CPU技术

□ 并行处理技术

- **同时性** 指两个以上事件在同一时刻发生；
- **并发性** 指两个以上事件在同一时间间隔内发生。

□ 计算机并行处理技术形式：

- **时间并行**：指时间重叠。
- **空间并行**：指资源重复。
- **时间并行+空间并行**：指时间重叠和资源重复的综合应用。例如，奔腾CPU采用了超标量流水技术。

□ 流水线技术设计方法：

将一个大的功能部件分成几个独立的功能部件，并行工作以提高执行速度的技术。

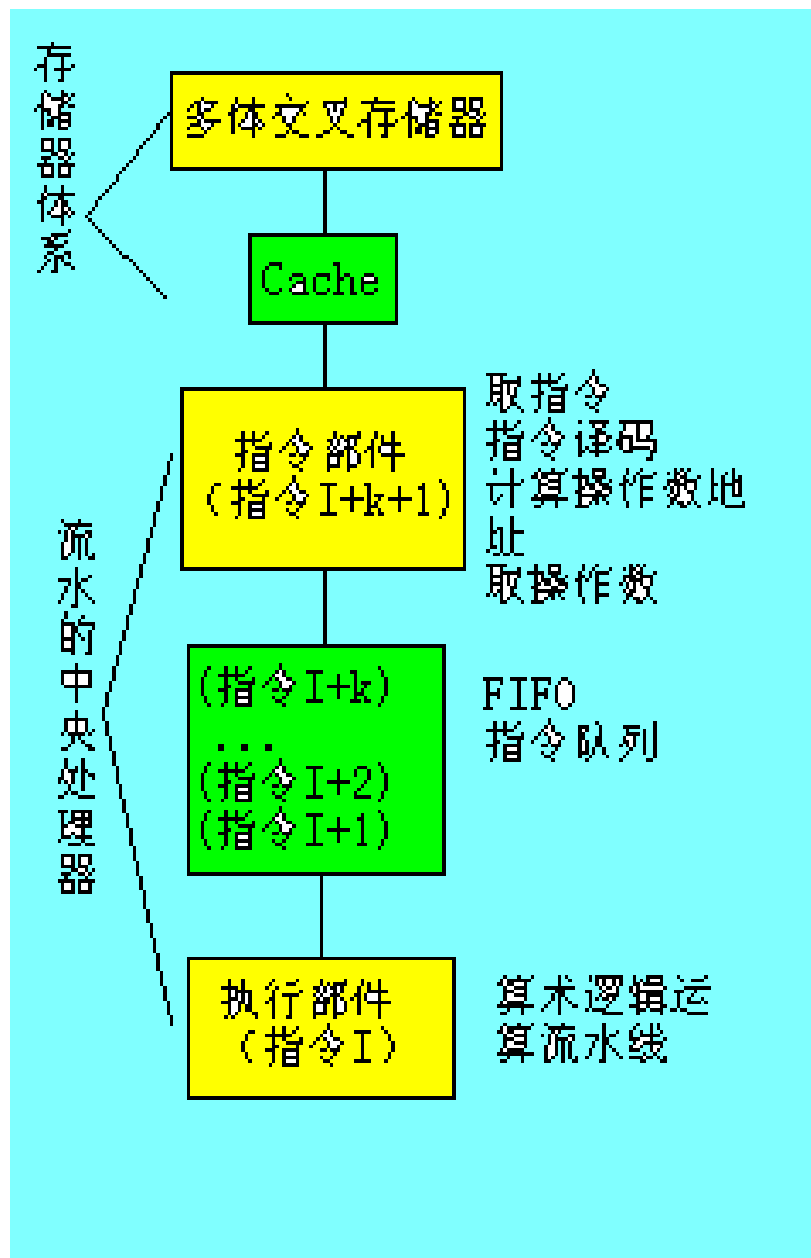
□ 流水线中功能部件必须满足的条件：

- 流水线中的任务必须是连续的，流水线应是完全充满的。
- 分解的任务是有联系的。
- 段与段之间传送字任务时，必须通过高速缓冲寄存器。
- 流水线中各段的执行时间应尽可能相等。

5.6.2 流水CPU的结构

□ 流水计算机的系统组成

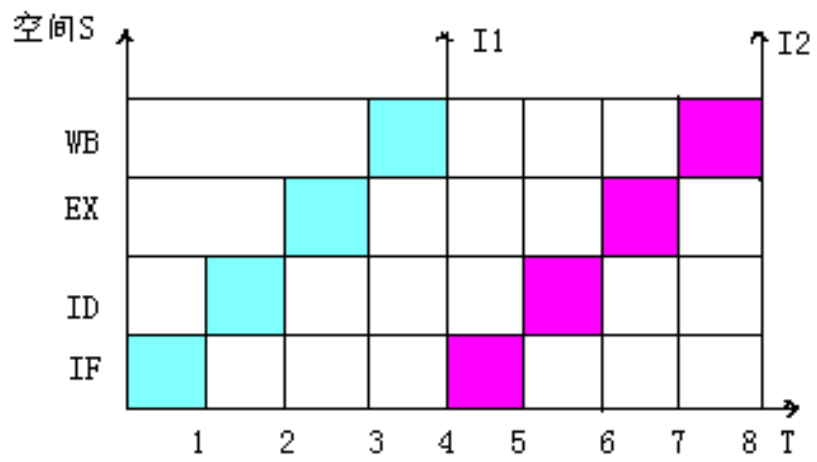
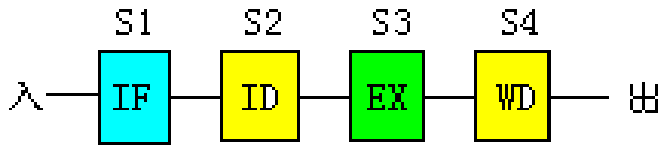
- 指令部件
- 指令队列
- 执行部件



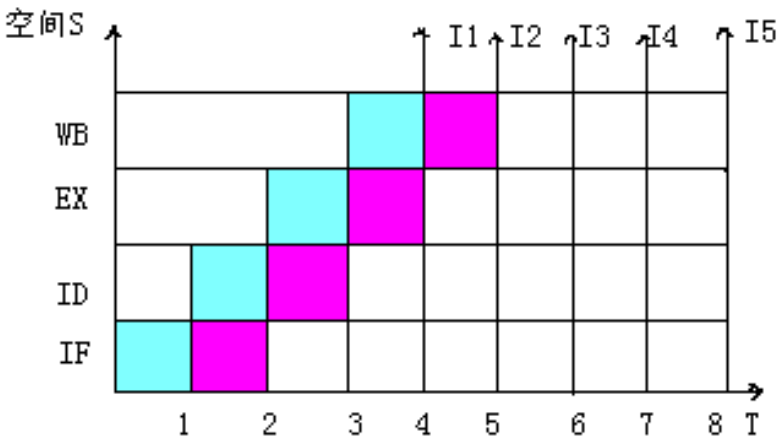
流水CPU的时空图

设指令周期包含四个子过程:

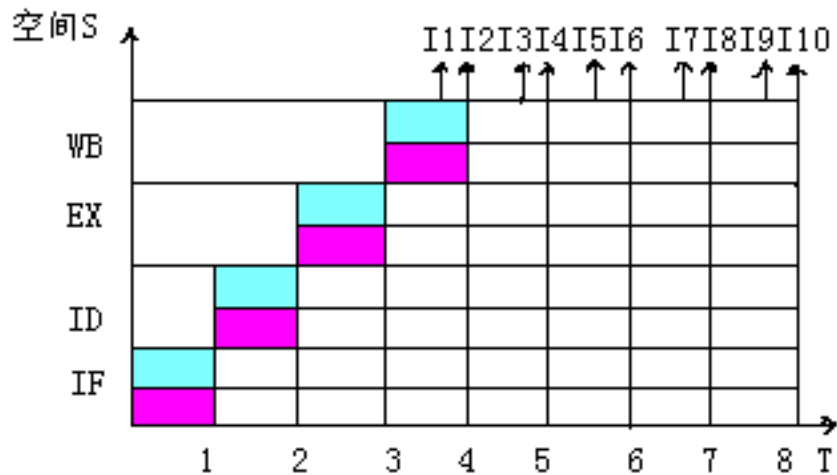
- 非流水线时空图
- 标量流水线时空图
- 超标量流水线时空图



(b) 非流水线时空图



(c) 标量流水线时空图



(d) 超标量流水线时空图

5.6.3 流水线性能

1. 吞吐率

单位时间内 流水线所完成指令 或 输出结果 的 数量

设 m 段的流水线各段时间为 Δt

- 最大吞吐率（满载后）

$$T_{pmax} = \frac{1}{\Delta t}$$

- 实际吞吐率 $TP = \frac{\text{完成的任务数 } n}{\text{完成 } n \text{ 个任务的时间}}$

连续处理 n 条指令的吞吐率
为

$$T_p = \frac{n}{m \cdot \Delta t + (n-1) \cdot \Delta t}$$

2. 加速比 S_p

m 段的流水线的速度与等功能的非流水线的速度之比

设流水线各段时间为 Δt

完成 n 条指令在 m 段流水线上共需

$$T = m \cdot \Delta t + (n-1) \Delta t$$

完成 n 条指令在等效的非流水线上共需

$$T' = nm \cdot \Delta t$$

则

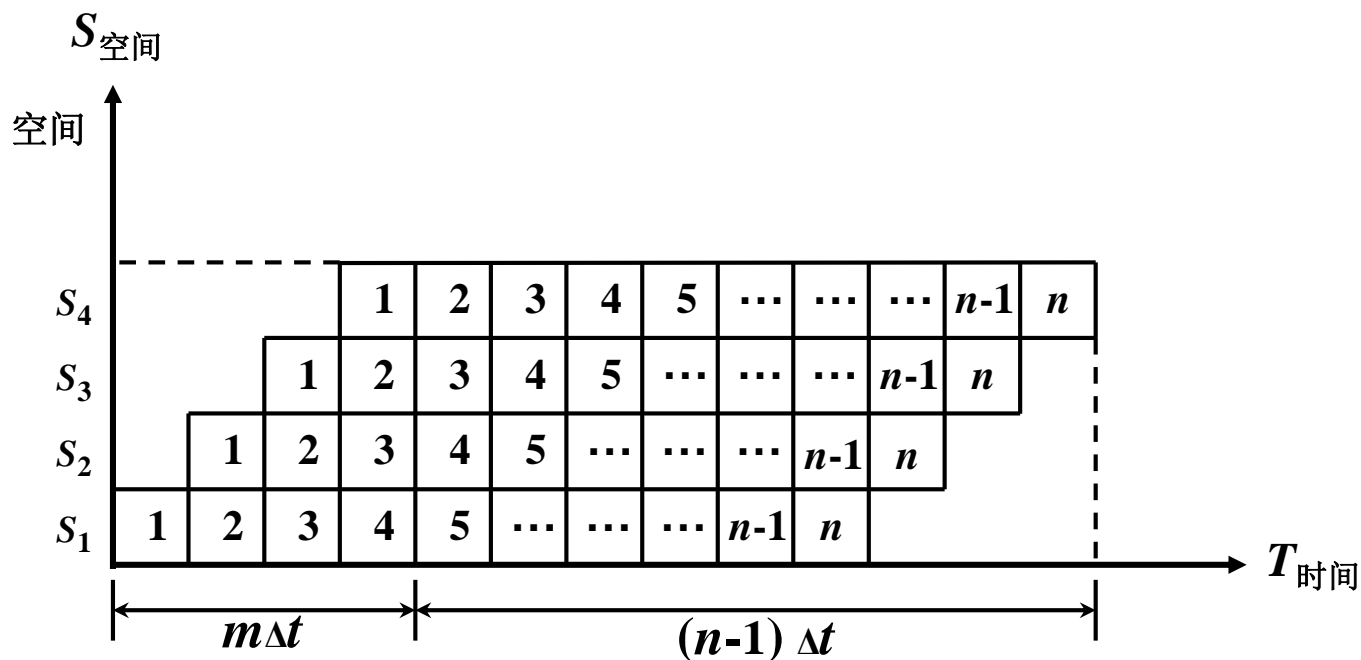
$$S_p = \frac{nm \cdot \Delta t}{m \Delta t + (n-1) \Delta t} = \frac{nm}{m + n - 1}$$

3. 效率

流水线中各功能段的 **利用率**

由于流水线有 **建立时间** 和 **排空时间**

因此各功能段的 **设备不可能** 一直 处于 **工作** 状态

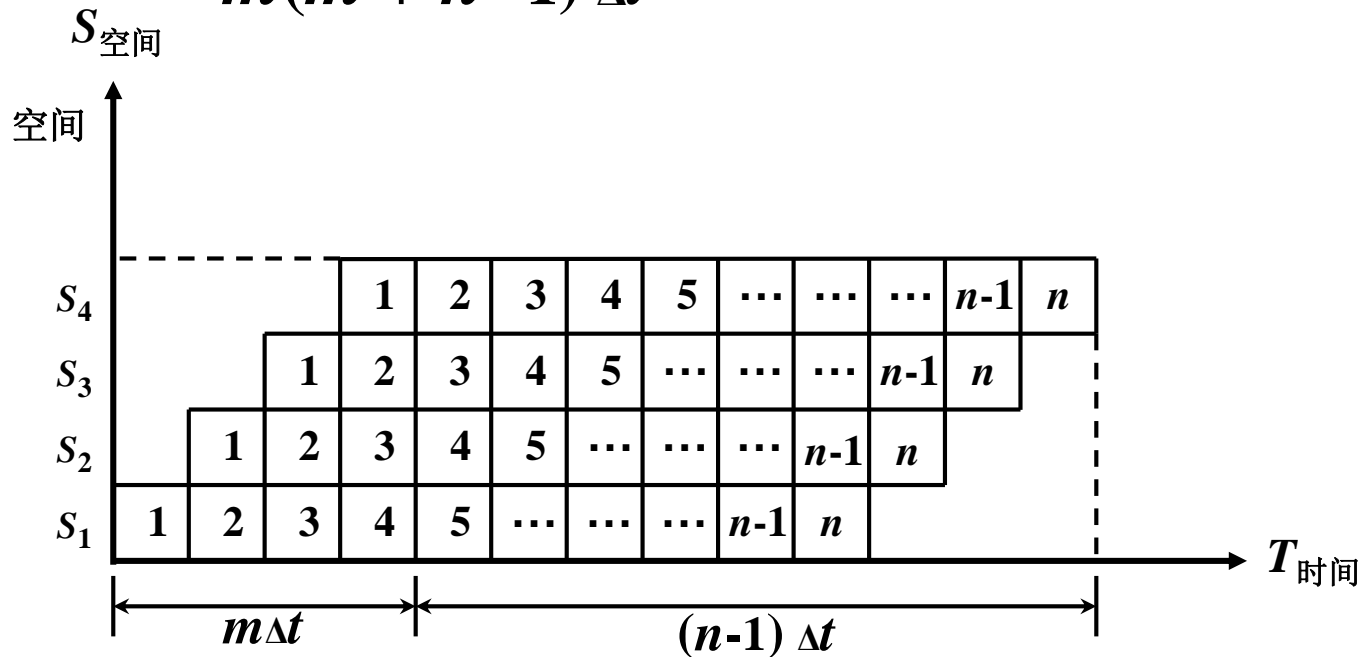


3. 效率

流水线中各功能段的 **利用率**

效率 = $\frac{\text{流水线各段处于工作时间的时空区}}{\text{流水线中各段总的时空区}}$

$$= \frac{mn\Delta t}{m(m+n-1)\Delta t}$$



□ 流水线中的主要问题

1. 资源相关

指多条指令进入流水线后在**同一机器时钟周期内**争用**同一个功能部件**所发生的冲突。

2. 数据相关

若在一个程序中，如果必须等**前一条指令执行完毕后，才能执行后一条指令**，那么这两条指令就是数据相关的

3. 控制相关

当**执行转移指令**时，使流水线发生**断流**的问题。

1. 资源相关:

设指令流水线分为：取指、译码、计算有效地址或执行、访存取数、写寄存器五段。

解决办法： 1. 停顿流水线

2. 重复设置存储器，数据和指令分开存放

3. 采用指令预取技术

指令 \ 时钟	1	2	3	4	5	6	7	8
I1 (Load)	IF	ID	EX	MEM	WB			
I2		IF	ID	EX	MEM	WB		
I3			IF	ID	EX	MEM	WB	
I4				IF	ID	EX	MEM	WB
I5					IF	ID	EX	MEM

大部分计算机数据和指令放在一个存储单元中，且只有一个端口

2. 数据相关

- 例执行： ADD R1, R2, R3
 SUB R4, R1, R5
 AND R6, R1, R7
- 解决办法： 1. 停顿流水线
 2. 采用定向技术（旁路技术或相关通路技术）：
 3. 增设运算结果缓冲寄存器

指令 \ 时钟	1	2	3	4	5	6	7	8
ADD	IF	ID	EX	MEM	WB			
SUB		IF	ID	EX	MEM	WB		
AND			IF	ID	EX	MEM	WB	

三种数据相关

- **写后读相关（RAW）**：指令j试图在指令i写入寄存器之前就读出该寄存器的内容，这样，指令j就会错误地读出该寄存器中的旧内容。

MUL R1,R2 ;

ADD R3,R1 ;

- **读后写相关（WAR）**：指令j试图在指令i读出寄存器之前就写入该寄存器，这样，指令i就错误地读出该寄存器中的新内容。

MUL R1,R2 ;

MOV R2,0 ;

- **写后写相关（WAW）**：指令j试图在指令i写入寄存器之前就写入该寄存器，这样，两次写操作的先后次序被颠倒，就会错误的使指令i写入的值成为该寄存器的内容。

MUL R1,R2 ;

MOV R1,0 ;

例4：流水线中有三类数据相关冲突：写后读（RAW）相关；读后写（WAR）相关；写后写（WAW）相关。判断下列指令存在哪种类型的数据相关。

(1) I1: ADD R1, R2, R3 ; $(R2) + (R3) \rightarrow R1$
 I2: SUB R4, R1, R5 ; $(R1) - (R5) \rightarrow R4$

RAW。

(2) I3: STO M(x), R3 ; $(R3) \rightarrow M(x)$
 I4: ADD R3, R4, R5 ; $(R4) + (R5) \rightarrow R3$

WAR

(3) I5: MUL R3, R1, R2 ; $(R1) \times (R2) \rightarrow R3$
 I6: ADD R3, R4, R5 ; $(R4) + (R5) \rightarrow R3$

WAW

3. 控制相关:

解决办法: 1. 延迟转移法

2. 转移预测法

3. 加快和提高形成条件码

4. 加快短循环程序的处理

5. 采用优化延迟转移技术

5.6.4 其他提高CPU性能方法

- 主要方法：
 - 改进芯片：与微电子技术发展密切相关。
 - 改进系统结构。
- RISC技术：为提高指令运行速度，简化指令的技术。
RISC特点：
 - 使用等长指令；
 - 寻址方式少，没有存储器间接寻址方式；
 - 只有取数和存数指令访问存储器，没有SS型指令；
 - 指令功能简单，控制器多以硬布线为主；
 - 大部分指令在一个处理周期内完成，支持指令流水线技术。
 - CPU中通用寄存器较多，且优化使用。
 - 采用优化的编译程序，可以有效支持高级语言程序。

- **MMX技术**：把各种不同的电子媒质集成起来，统一进行存储、处理和传输的扩展结构技术。新增了专用的数据类型、寄存器和指令。
 - 采用SIMD（单指令多数据处理）型指令。
 - 具有“饱和”运算功能。
 - 具有“积和”运算的能力。
 - 具有比较指令。
 - 具有转换指令。
- **动态执行技术**：通过预测程序流来调整指令的执行，并且分析程序的数据流来选择指令执行的最佳顺序。