

Chapter 5:流水线和向量处理机

计算：加速比、吞吐率、效率、时空图、调度

重点：

- 一次重叠和相关处理

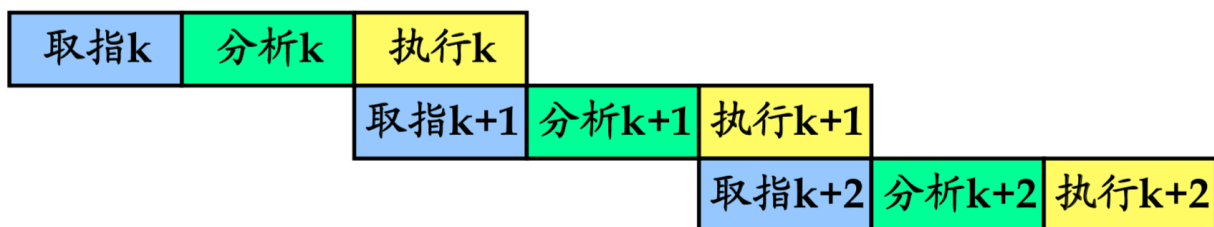
- 流水方式的基本概念、流水线处理及性能

- 非线性流水线的调度、性能

- 向量的处理方式

5.1重叠方式

一次重叠



二次重叠：



重叠执行需要解决的问题：

1.功能部件冲突 ->12结构相关或资源相关

设置独立的部件，流水线每个步骤都有独立的部件

2.访存冲突

主存同一时间只能访问一个存储单元

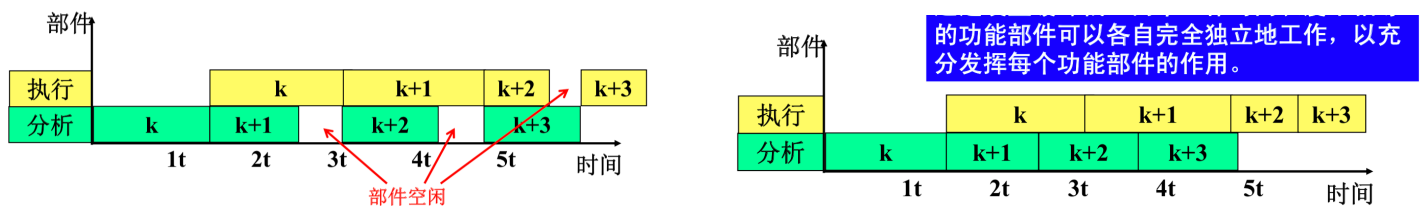
计算机体系结构第五章

解决方法:

1. 分开存放操作数和指令
2. Cache: 分成两个Cache: 指令Cache和数据Cache->哈佛结构
3. 多体交叉存储器, 采用地位交叉存取方式
4. 增加指令缓冲寄存器FIFO, 从指令缓冲存储区取指令, 避免访问冲突, pc1预取指令

3.同步

执行和分析的时间不完全相同, 可能造成功能部件浪费



解决方法:

1. 先行控制: 缓冲技术+预处理技术

缓冲深度: 寄存器个数: 太少: 不起作用, 太多: 浪费, 逻辑控制复杂

	类型	种类	解决方法
相关	数据相关 (局部相关)	指令相关、操作数相关 (主存数据相关、寄存器数据相关)、变址相关	尽可能避免发生数据相关: 指令推后分析、设定专用通路
	控制相关 (全局相关)	无条件转移相关、一般条件转移相关、符合条件转移相关、中断、程序调用	增设无条件转移指令分析器、采用转移预测技术、采用短循环程序处理技术

4.转移。控制相关

5.指令、数据相关（冯诺伊曼结构）。数据相关

指令相关（属于数据相关）的原因：冯氏结构允许修改指令和数据（指令和数据不作区分）

处理：1. 不允许修改指令，程序设计灵活性差

2. 允许修改指令，指令相关转换为数据相关，统一按照数据相关进行处理

数据相关处理：

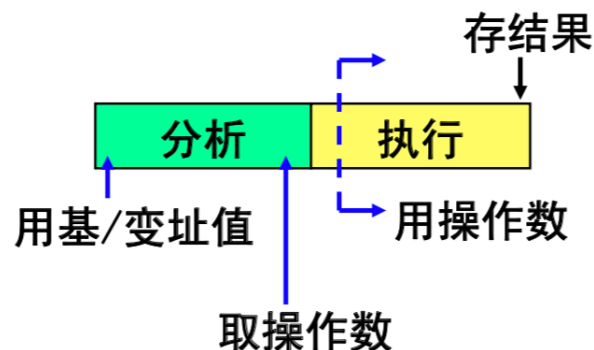
1.主存空间数据相关，先写后读关联

解决方法：无需增加硬件，直接将下一条指令的读向后推迟一个周期

2.寄存器组数据相关

解决方法：

要根据寄存器中存储的数据不同来处理，不同类型数据使用阶段不同：



通用寄存器数据相关分为 1操作数相关：

解决：1推后读 2设置相关专用通路（直接将运算结果送到运算器）

2.基址/变址值

解决：1. 推后读 一次二次都向后推，推迟时间看 Δ 2. 设置相关通路, 送入寄存器的同时直接送到地址形成机构

控制相关：

尽可能不使用条件转移指令

使用不成功的转移指令（对流水线影响小）

也可以采用RISC的延迟转移技术

5.2流水方式

基于时间重叠的并行处理技术，不能加快一条指令的执行，能够加快多条指令的执行。

每个子过程时间应该尽量相等：

流水线工作阶段可以分为：建立、满载（所有部件都开始工作）、排空

≥ 8 个流水段称为超流水线

复杂指令执行阶段也采用流水线

分类：单功能/多功能流水线：单功能：只能实现一种功能的流水线：比如只能完成除法

静态/动态流水线：静态：一个时间段只能运行一种功能 动态：控制复杂

线性/非线性：线性：没有反馈回路 非线性：有反馈回路

标量/向量：处理的数据类型

顺序/乱序（同步/异步）：流出顺序与流入顺序是否相同

最大吞吐率与实际吞吐率：实际吞吐率小于最大吞吐率，因为流水线实际工作还有建立和排空时间。

加速比：采用流水线与不采用流水线时间的比值。 >1

段数越多，其他开销越大：比如设计和控制更加复杂。

效率：设备实际使用时间占整个运行时间的比

时空图阴影部分面积与最大的矩形面积之比

流水线调度：资源冲突

线性流水线：每隔 Δt 输入一个任务即可

非线性：一次任务在执行过程中多次经过某一段

解决：二维预约表法

流水线启动距离：等待时间、两个人物之间的时间间隔

流水线的冲突：几个任务争用同一个流水段

- ① 构建延迟禁止向量 **F**
- ② 构建初始冲突向量 **C**
- ③ 计算流水线的新的冲突向量
- ④ 构造用冲突向量表示的流水线状态图
- ⑤ 确定调度方案

找出平均间隔最小的就是最佳调度

流水线相关处理：

局部性相关：典型：先写后读，解决方法：推后读/设置专用通路

全局性相关：典型：中断/转移，解决方法：猜测/加快形成条件码/延迟转移

流水线中断处理

流水线中断特点：

第 i 条指令中断的时候， $i+1$ ， $i+2$...已经部分进入流水线

对于异步流水线，有的指令可能已经流到 i 之前

解决方法：

不精确断点法：不允许后面的指令再进入，已经进入的执行完毕，然后再转入中断处理程序

精确断点法：只保护第 i 条指令的现场， i 之后指令现场都能恢复

向量数据表示与流水处理方式结合：向量流水处理机

向量处理方式：

水平：不符合流水线方式

垂直：多体交叉存储，设置向量寄存器组

分组纵横：向量过长，分组寄存器存不下整个向量（组内垂直，组间水平）

关键问题：存储器系统能够满足运算部件带宽要求

两种方法：

存储器-存储器结构

寄存器-寄存器结构

V_i 冲突：寄存器出现先写后读的情况

*提高向量处理机性能的方法：

计算机体系结构第五章

向量平衡点：为了使向量硬件设备和标量硬件设备利用率相等，程序中向量代码所占的百分比

采用链接技术，加快一串向量指令的执行

采用循环开采技术，加快循环处理

采用多处理机系统，提高性能

流水线分类：

部件级/处理机级/处理机间级

单功能/多功能：是否只能实现一种功能如加减法

静态/动态：一个时间段内只能按一种功能进行流水/

线性/非线性：线性没有反馈回路/非线性有反馈回路

标量/向量：是否有向量数组表示

同步/异步

乱序/顺序

指令级并行（ILP）：

每个时钟周期内解释多条指令

超标量处理机：每个周期发射 m 条指令，每条流水线执行不相关的指令，流水机的度为 m

超流水处理机：每个时钟周期分为 n 份，

超标量超流水处理机：结合上面两个

超长指令字处理机：一个指令可以实现多个功能，一条长指令实现多个操作的并行执行

■ 性能比较

- 最高：超标量处理机；
- 其次：超标量超流水线处理机；
- 最低：超流水线处理机。

主要原因：

- (1) 超流水线处理机的启动延迟比超标量处理机大。
- (2) 条件转移造成的损失，超流水线处理机要比超标量处理机大。
- (3) 超标量处理机指令执行部件的冲突要比超流水线处理机小。

Chapter 6 并行处理机与互连网络

也称为阵列处理机

属于SIMD计算机

从CU看，指令串行执行

从PE看，数据并行处理

两种基本构型：

分布式存贮器

集中共享存贮器

互连网络特点：

网络规模：节点个数

节点的度

距离：两个节点之间的最少边数

网络直径：任意两个节点直接距离的最大值

节点间线长：两个节点直接连线的长度

计算机体系结构第五章

等分宽度：分成两个相等部分切断最小边数

对称性：任何节点看网络的拓扑结构都是相同的，容易实现

交换开关：

不允许出现多个输入对应一个输出->产生冲突

只允许一对一映射称为**置换连接**， $n \times n$ 交叉开关

三种控制方式：

- 1) 级控制：每一级开关使用一个控制信号
- 2) 单元级控制：一个开关一个控制信号
- 3) 部分级控制：部分采用一级控制一个信号，部分单独控制

全排列网络：了解就行