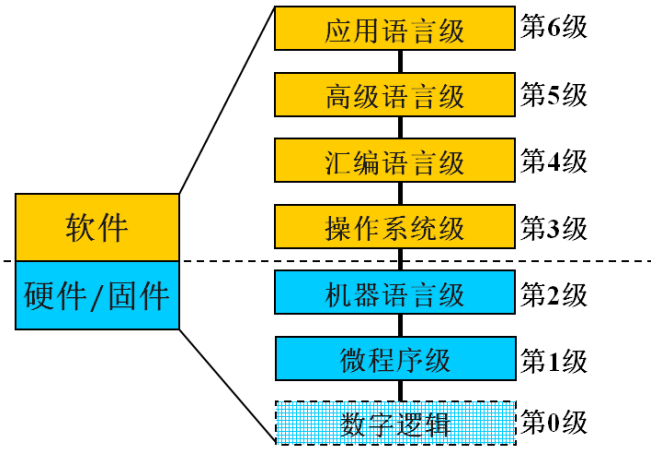
《计算机体系结构》复习提纲

第一章 基本概念

1.1 多级层次结构和机器级的实现技术

一，多级层次结构概念



二，机器级的实现技术

- 翻译和解释或这两者的结合是各机器级实现的主要方法。
- 掌握翻译或解释概念和方法，注意两者的区别。
- 问题：某级机器用硬件实现，还是用软件实现？
  - 在逻辑功能上，软件和硬件是等效的。原理上，软件实现的功能完全可以由硬件实现，硬件实现的功能也完全可以由软件模拟完成。
  - 但软件和硬件的性能价格比是不等效的。

1.2 系统结构、组成与实现及相互关系

系统结构定义：

计算机体系结构是对各机器级界面的划分、定义及上下级功能分配。

- 按照计算机系统的多级层次结构，不同机器级的界面有很大不同，

这意味着每个机器级都有其系统结构。

- **计算机体系结构概念的实质是计算机系统中软硬件界面的确定，其界面之上的是软件的功能，界面之下的是硬件和固件的功能。**
- **“指令集结构”就是软硬件之间的界面之一。**
- 传统机器级界面(软硬件界面)：机器语言程序员所看到的计算机系统的属性。包括：
  - ✓ 数据表示
  - ✓ 寻址方式
  - ✓ 寄存器组织
  - ✓ 指令集
  - ✓ 存储系统
  - ✓ 中断系统
  - ✓ 机器工作状态的定义和切换
  - ✓ 输入输出系统
  - ✓ 信息保护

透明性定义：

本来存在的事物或属性，从某个角度看却好象不存在。

传统机器级的属性对高级语言程序员来说是透明的。

哪些透明？作业1-6，1-7

计算机组成的定义：

对系统结构的逻辑实现。包括：机器级内部的数据流、控制流的组成及逻辑设计、部件功能、部件间的联系等。

所解决的问题：

在所希望达到的性能价格下，如何最佳、最合理地把各种设备和部件组织在一起，以实现所确定的系统结构。

计算机组成的设计内容：

- ✓ 数据通路宽度
- ✓ 专用部件设置
- ✓ 操作对部件的共享程度
- ✓ 功能部件的并行度
- ✓ 控制机构的组成方式
- ✓ 缓冲和排队技术
- ✓ 预估、预判技术
- ✓ 可靠性技术 等

计算机实现的定义：

对计算机组成的物理实现，包括：

- ✓ 处理机、主存等部件的物理结构
- ✓ 器件的集成度和速度
- ✓ 器件、模块、插件、底板的划分和连接
- ✓ 专用器件设计
- ✓ 信号传输；
- ✓ 电源、冷却、微组装技术、
- ✓ 整机装配技术等

其中，器件技术在实现技术中起主导作用

系统结构、组成和实现的关系：

- ■ 三个不同的概念，即互相联系，又互相影响。
- ■ 不同系统结构使用不同组成技术。
- ■ 相同系统结构可以采用不同的组成。
- ■ 一种组成可以有不同的实现。
- ■ 组成也会影响系统结构。

例如：

- ■ 指令系统的确定属系统结构；
- ■ 取指、取操作数、运算送结果等具体操作及其排序方式属组成；
- ■ 具体电路、器件的设计及装配技术等属实现。

例如：

- ■ 确定指令系统中是否要设置乘法指令属计算机系统结构；
- ■ 乘法指令是采用专门的高速乘法器实现，还是靠用加法器和移位器经时序信号控制其相加和移位来实现属计算机组成。

例如：

- ■ 主存容量与编址方式（按位、按字节还是字编址等）的确定属计算机系统结构；
- ■ 而为达到性能价格要求，主存速度应选多快，采用何种逻辑结构(例如是否采用多体交叉)等则属计算机组成。

1.3 Amdahl定律

$$S_n = \frac{T_{old}}{T_{new}} = \frac{1}{(1 - Fd - Fe) + \frac{Fd}{Sd} + \frac{Fe}{Se}}$$

1.4 计算机系统的设计思路

从多级层次结构出发，计算机系统的设计有三种思路：

- 由上往下
- 由下往上
- 由中间开始。中间为软硬件界面。

软件的可移植性定义。

实现软件可移植性的几种技术，掌握这些技术的主要特点。

- 统一高级语言
- 采用系列机思想
- 模拟与仿真。宿主机，虚拟机，目标机，仿真机。

软件兼容的定义。注意与可移植性的区别。

软件兼容的分类：

- 向上(下)兼容
- 向前(后)兼容

系列机的定义，系列机软件的兼容性要求：

- 系列机软件必须保证向后兼容，力争做到向上兼容!

兼容机的定义

软件兼容的缺点

仿真与模拟的定义与区别

1.5 并行性及计算机系统分类

并行性定义：

可以同时进行运算或操作的特性，包含两重含义：

同时性(simultaneity)：同一时刻发生

并发性(concurrency)：同一时间间隔内发生

并行性等级：不同角度不同分级

开发并行性的三种途径：

- 时间重叠：
  - ✓ 引入时间概念，让多个处理过程轮流使用同一套硬件设备的各个部分，以加快硬件周转而赢得速度。
- 资源重复：
  - ✓ 引入空间因素，重复设置多套硬件。
  - ✓ 提高速度，同时提高可靠性。
- 资源共享
  - ✓ 利用软件让多个用户轮流使用同一套资源。

要求：掌握各自的特点，能举例说明。

多机系统包括：

- 多处理机系统
  - ✓ 由多台处理机组成的单一计算机系统
- 多计算机系统
  - ✓ 由多台独立的计算机组成的系统

要求：了解它们的定义，之间的差别，实现的并行性，能举出例子。

多机系统的耦合度

耦合度的定义。

耦合度分为：

- 最低耦合
- 松散耦合
- 紧密耦合

要求：通信方式，举例说明耦合度

弗林(Flynn)分类法：

- 指令流
- 数据流
- 多倍性

它将计算机系统分为四类：

- SISD（单指令流，单数据流）

- SIMD（单指令流，多数据流）
- MISD（多指令流，单数据流）
- MIMD（多指令流，多数据流）

要求：能举例说明 哪种机器属于哪个分类。

作业：1-6，1-8，1-9

第二章 数据表示与指令系统

2.1 自定义数据表示

数据表示：可以被硬件直接识别和指令系统直接调用的数据类型。

注意：数据表示与数据结构的关系

自定义数据表示的定义：由数据本身来表明数据类型，使计算机内的数据具有自定义能力。分为：

- 带标志符的数据表示
- 数据描述符

注意它们之间的的区别。

编址方式和程序定位方法：

- 寻址技术：寻找操作数及其他信息的地址的技术
- 编址方式：指对各种存储设备进行编码的方法
- 部件的编址方式主要有三种：
  - 统一编址
  - 分类编址
  - 隐含编址

大多数计算机采用分类编址方式。通常将主存、通用寄存器和堆栈分类编址。

- 定位方式
  - 何时确定程序在主存中的物理地址？
  - 以何种方法确定程序在主存中的物理地址？
- 程序定位方法
  - 相关概念：
    - ✓ 逻辑地址
    - ✓ 主存物理地址
    - ✓ 程序定位
  - 程序定位方法：
    - ✓ 静态再定位

✓ 动态再定位

2.2 指令格式优化

优化的目的。

指令的组成。

操作码的优化方法：

- 定长编码
- 全哈夫曼编码
- 扩展操作码编码

要求：能根据题目要求设计操作码；会构造哈夫曼树，会分配编码会计算操作码优化。

信息源熵

- 信息源包含的平均信息量
- 对操作码而言就是操作码的最短平均码长（理想情况）。
- 计算公式：

$$H = -\sum p_i \log_2 p_i$$

其中：H—信息源熵

p_i—第i个操作码出现的概率

信息冗余量计算公式：

$$R = 1 - \frac{H}{\text{实际平均码长}}$$

实际平均码长计算公式：

$$L = \sum p_i l_i$$

其中：p_i—第i个操作码出现的概率

l_i—第i个操作码的长度

2.3 指令系统的改进

指令系统的设计、发展和改进有两种不同的途径和方向：

- 增强指令功能。基本思路
- 简化指令系统。基本思路

CISC定义

- CISC结构追求的目标：
  强化指令功能，减少程序的指令条数，以达到提高性能的目的。
- 从以下几个方面考虑优化：
  - ✓ 面向目标程序的优化实现
  - ✓ 面向高级语言的优化实现
  - ✓ 面向操作系统的优化实现

CISC主要缺点。

RISC定义、特点、关键技术

- 设计RISC机器应当遵循的一般原则
- RISC采用的基本技术、关键技术：
  - ✓ 简单的寻址方式
  - ✓ 延时转移技术
  - ✓ 重叠寄存器窗口技术
  - ✓ 指令取消技术
  - ✓ 优化编译系统设计的技术 等

逻辑实现用级联和微程序结合的原则。

RISC结构存在的不足：

- 指令减少，加重了汇编程序员的负担，增加了机器语言程序的长度，占用了较大的存贮空间，加大了指令的信息流量
- RISC上的编译程序要比CISC上的难写

作业：2-4，2-9

第三章　输入输出系统

3.1 I/O系统的控制方式

在计算机系统中，把处理机与主存储器之外的部分统称为I/O系统。

I/O系统功能：

对指定外设进行I/O操，同时完成许多其它的管理和控制任务，如：编址、准备信息通路、信息传送、格式变换、中断请求等。上述功能由三个部分协同完成：

- 输入输出机器指令
- 输入输出设备及其控制器硬件
- 操作系统

输入输出系统的性能包括：

- 输入输出速度；
- 用户从输入到输出的等待时间；
- CPU和主存的利用率；
- I/O系统的兼容能力，可扩展能力，综合处理能力，性能价格比等。

I/O系统的控制方式：三种

- 程序控制方式
- DMA方式
- I/O处理机方式

要求：掌握每种方式的特点和优缺点

I/O处理机方式有两种形式：

- 通道方式
- 外围处理机方式

3.2 通道处理机

通道处理机特点：

- 它拥有通道指令和通道程序

- 可与CPU并行工作
- 由通道统一管理外设

通道处理机的工作有三个阶段：

- 通道开始选择设备期
- 通道数据传送期
- 通道数据传送结束期

完成一次I/O操作需要两次进管。

通道类型：按信息传送的方式，分为：

- 字节多路通道
- 数组多路通道
- 选择多路通道

它们的速度和容量是不同的。注意掌握每种通道的特点。

通道流量分析

- 通道极限流量
- 实际最大流量

掌握三种通道的通道极限流量的计算方法。

字节多路通道的极限流量：

$$f_{max \cdot byte} = 1/(T_S + T_D)$$

字节多路通道的实际最大流量（求和）：

$$f_{byte \cdot j} = \sum_{i=1}^{p_j} f_{i \cdot j}$$

数组多路通道的极限流量：

$$f_{max \cdot block} = K/(T_S + KT_D)$$

数组多路通道的实际最大流量（求最大值）：

$$f_{block \cdot j} = \max_{i=1}^{p_j} f_{i \cdot j}$$

选择多路通道的极限流量

$$f_{max \cdot select} = N/(T_S + NT_D)$$

选择多路通道的实际最大流量（求最大值）

$$f_{select \cdot j} = \max_{i=1}^{p_j} f_{i \cdot j}$$

其中：

fi，j— 第 j 号通道上第 i 个设备的数据传输率

Pj — 第 j 号通道上的设备数

流量设计的基本原则

极限流量≥实际最大流量

$$f_{max} \geq f$$

要求：

- 会进行流量的计算设计
- 正确安排优先级
- 正确画出通道工作的示意图
- 正确解决出现的问题

作业：3-3，3-6

第四章　存贮体系

4.1 存贮体系及其性能

对存贮器的基本性能要求：

- 大容量
- 高速度
- 低价格

但容量、速度和价格是相互矛盾的。**解决矛盾的措施：**

- ✓ 改进工艺和技术，降低成本、提高速度
- ✓ 使用存贮器系统，例如至少有主存和辅存两种存贮器
- ✓ 构成并行主存系统
- ✓ 采用存贮体系

存贮体系（存贮层次）

- 定义。**多级存贮层次。**
- **主存—辅存存贮层次**
  - ✓ 主要解决容量问题
  - ✓ 对应用程序员透明
- **Cache—主存存贮层次**
  - ✓ 主要解决速度问题
  - ✓ 对应用程序员和系统程序员都透明

程序的局部性及定义

程序的局部性包括：

- 时间局部性
- 空间局部性

程序的局部性是预判的基础。预判的准确性是存贮层次设计好坏的主要标志，很大程度上取决于所使用的算法和地址映像与变换方式。

存贮层次的性能：对于两级存贮层次

- 每位价格c
- 命中率H：在第一级存贮器访问到所需数据的概率

- T1：第一级存贮器的访问时间
- T2：第二级存贮器的访问时间

则两级存贮层次的等效访问时间：

- 同时启动：TA = H×T1 + (1 - H)×T2
- 不同时启动：TA = T1 + (1 - H)×T2

4.2 虚拟存贮器

- 虚拟存贮器是一个存贮体系。
- 它对应用程序员透明，对系统程序员基本不透明。
- 它允许应用程序员使用比实际主存空间大得多的程序（虚存）空间访问主存。
- 每次访存时都要进行虚地址到实地址转换。

虚拟存贮器的管理

- 段式管理
- 页式管理
- 段页式管理

虚拟存贮器采用地址映像方法、通过地址映像表机构实现程序的定位。

4.3 页式虚拟存贮器

一般原理

- 地址映像算法：全相联
- 地址变换：页表，相联目录表

注意它们的区别。

注意装入位。

注意如何查页表。

会计算虚地址的虚页号。

- 地址映像
- 地址变换
- 页面失效

- 页面争用或实页冲突

注意**外页表**和**内页表**的区别。

注意**快表**和**慢表**的区别及关系。

多级页表。P99

替换算法

- 先进先出算法
- LRU算法
- 优化替换算法

注意堆栈型算法及其特点。

重要结论：

- LRU、OPT为堆栈型算法，FIFO不是。
- 堆栈型算法的命中率随分配实页数的增加而单调上升，至少不会下降。
- FIFO为非堆栈型算法，实页数的增加有时可能会降低命中率。
- 可以用堆栈实现堆栈型替换算法。

提高虚拟存贮器的等效访问速度的措施：

- 提高主存命中率
  - ✓ 程序地址流、页面调度策略、替换算法、页面大小、以及分配给程序的实页数（主存容量）等都会影响主存命中率。
- 加快内部地址变换
  - ✓ 增加快表

快表：用快速硬件构成小容量的“相联目录表”，存放当前正在使用的虚实地址映象关系。

- ✓ 查表时，同时查快表和慢表，并将快表中不存在的内容从慢表中调入。替换算法一般也采用LRU

4.4 高速缓冲存贮器Cache

Cache特点

- 地址映象与变换，掌握：

- ✓ 全相联映象与变换
- ✓ 直接映象与变换
- ✓ 组相联映象与变换

- 替换算法：LRU
- Cache全部采用硬件实现

全相联映象与变换：

- 特点：主存中的任意一块均可以装入到Cache内的任意一块位置上。
- 实现：采用目录表硬件方式实现。

直接映象与变换：

- 特点：主存中的每一块只能装入到Cache内唯一一个指定的块位置上。
- 实现：设置一个按地址访问的表存贮器（区号表），存放Cache中每一块目前被主存中哪个区的对应块所占用。

组相联映象与变换：

- 特点：
  - ✓ 将Cache分成G组，每个组有S块。Cache地址：

组号 g	组内块号 s	块内地址

  - ✓ 将主存按每组G个主存块分成数个组，主存块大小与Cache块相同。

标记 tag	组号 g （Index）	块内地址

  - ✓ 主存地址组号与Cahce组号之间采用直接映象，该主存块与Cache组内各块之间采用全相联映象。

可以看出，组相联映象介于全相联映象和直接映象之间：

- ✓ 若S = Cache的块数，就变成了全相联映象

- ✓ 若S = 1，就变成了直接映象

Cache的透明性：Cache更新方法

实现Cache一致性的**两种“写”策略：**

- 写直达法
- 写回法

两种 “写”调块方法：

- 不按写分配法
- 按写分配法

Cache的预取算法：

- 按需取：在出现Cache不命中时，把一个块取到Cache中来
- 恒预取：无论Cache是否命中，都把下一块取到Cache中
- 不命中预取

Cache的性能

作业：4-4，4-7

第五章　重叠、流水、向量处理机

5.1 一次重叠及相关处理

一次重叠定义及特点

在第K条指令完成之前就开始执行第K+1条指令。任何时间，指令分析部件和指令执行部件都只有相邻的两条指令在重叠解释

重叠对计算机组成的要求

- 访存冲突
- 功能部件冲突
- 同步
- 转移
- 相关

一次重叠及相关处理

相关

- 定义：邻近指令之间出现了某种关联，为了避免出错而使得它们不能被同时解释执行的现象。
- 原因：先写后读。
- 类型：
  - ✓ 指令相关
  - ✓ 主存数相关
  - ✓ 寄存器组相关
- 解决方法：根据相关类型，采取不同的方法
  - ✓ 不允许修改指令
  - ✓ 允许修改指令，但改变指令的执行方式，将指令相关转换为数相关。**主要有两种解决方法：**
    - ◆ 推后读
    - ◆ 设置相关专用通路

5.2 流水方式

流水方式的特点

- 将指令的执行过程分解得更细，分为更多的子过程，并让每个子过程分别由独立的子部件完成
- 按时间重叠方法，可以同时处理多条指令
- 机器的最大吞吐率成倍提高

流水线的分类：

- 单功能流水线
- 多功能流水线
- 静态流水线
- 动态流水线
- 标量流水处理机
- 向量流水处理机
- 线性流水线
- 非线性流水线

流水线的性能：

- 吞吐率（会计算）
- 加速比（会计算）
- 效率（会计算）

吞吐率

- 最大吞吐率：满负荷时达到
- 实际吞吐率

最大吞吐率

- 它受限于流水线中最慢子过程所需要的时间
- 称流水线中经过时间最长的子过程为瓶颈子过程
- 消除瓶颈子过程的方法有：
  - ✓ 细分瓶颈过程
  - ✓ 重复设置瓶颈功能段

实际吞吐率TP

- 指流水线单位时间内实际处理的指令条数或输出的结果数
- 若流水线在T流水时间内处理了n条指令，则TP = n/T流水

要求：掌握线性流水线的性能计算方法

$$TP_{max}=1/\max\{\triangle t_1,\triangle t_2,\triangle t_3,\triangle t_4,\ldots\}$$

$$TP = n/T_{流水}$$

$$\eta=\frac{\eta_1+\eta_2+\ldots+\eta_m}{m}=\frac{m\cdot\eta_0}{m}=\frac{m\cdot n\cdot\Delta t_0}{m\cdot T}$$

$$加速比 = T_{顺序执行}/T_{流水}$$

从时—空图上来看，效率实际上就是n个任务所占用的时空区面积与m个段所占用的总的时空区的面积之比

$$TP=\frac{任务数}{从开始流入到n个任务全部流出的时间}$$

$$\eta=\frac{n个任务的加权时—空区}{m个段的总的加权时—空区}$$

要求：熟练掌握时空图的画法，合理安排任务输入，进行正确的计算，包括吞吐率，效率，加速比。

流水线相关与中断

流水线相关分为两类：

- 局部性相关
- 全局性相关

局部性相关解决方法有两种：

- 推后读
- 设置相关专用通路

5.3 流水线相关与中断

全局性相关解决方法：

- 关键是能正确判断判断转移分支。方法：
- 猜测法
- 加快和提前形成条件码
- 采用延迟转移

- 加快短循环程序的处理

中断处理方法：

- 不精确断点法
- 精确断点法

正确理解“不精确”和“精确”

流水线的调度

1，线性流水线的调度

线性流水线无反馈，每个任务只通过每个段一次，不会发生冲突，因此只需每隔 Δt 输入一个任务即可。

2，非线性流水线的调度

非线性流水线段间有反馈回路，如果每拍向流水线输入任务，将会发生资源冲突

3，二维预约表调度方法（掌握）

用于非线性流水线的调度。

它利用类似时—空图的方法，得到一个任务使用流水线各段的时间关系表（预约表）。

根据预约表可以得出一个任务使用各段所需**间隔的拍数**。间隔这些拍数就会产生争用。

将流水线中所有各段对一个任务流过时会产生争用的节拍间隔数汇集在一起，构成延迟**禁止表F。**

根据延迟禁止表F 可以得到任务刚进入流水线时的**初始冲突向量C(C_n C_{n-1}…C_i…C_1)，**

其中：

n=N-1

C_i = 1 表示间隔i拍会产生冲突

C_i = 0 表示间隔i拍不会产生冲突

随着流水线中的任务每拍向前推进一个功能段，原先禁止后续任务进入流水线的间隔拍数应相应地减1，这意味着可以将冲突向量放在一个移位器中，**每拍右移1位，让左面移出的空位补“0”。**

若让第三个任务流入流水线后，即不与第一个任务发生功能段冲突，也不与第二个任务发生功能段冲突，**新的冲突向量就应当是第一个任务当前的冲突向量与第二个任务初始的冲突向量按位“或”。**

按照这样的思路，选择各种可能的拍数输入信任务，又可以产生新的冲突向量，一直到不再产生不同的冲突向量为止，这样就考虑了所有的可能性。

由此可以画出用冲突向量表示的流水线**状态转移图。**

只要从状态图中的初始状态出发，找到一个间隔拍数呈周期性重复的方案，并按此方案进行调度，就不会发生功能段冲突。

显然，可以找到多个调度方案。**问题：哪一种调度方案最佳呢？**

所谓最佳调度方案是指：

- 流水线吞吐率最高
- 控制简单

根据上述状态图，可以找出所有的调度方案，并计算出每种方案的平均间隔拍数。平均间隔拍数最小的就是吞吐率最高的。

为了简化控制，也可以采用等间隔调度，但常常会使吞吐率和效率下降。

5.4 向量的流水处理

向量处理机定义

向量的处理方式主要有三种：

- 水平（横向）处理方式
- 垂直（纵向）处理方式
- 分组（纵横）处理方式

问题：哪种适合流水处理？

指令级高度并行的超级处理机

代表性的例子是：

- 超标量（Superscalar）处理机
- 超长指令字（VLIW）处理机

- 超流水线（Superpipelining）处理机

常规的标量流水线单处理机是在每个△t时间内解释完一条指令。称这种流水机的度（ILP）为1。

超标量处理机采用多指令流水线，在每个△t时间内同时解释完m条指令。称这种流水机的度（ILP）为m。

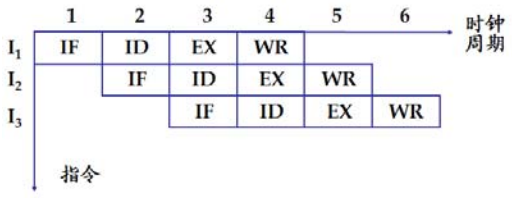
机器类型	k段流水线基准标量处理机	m度超标量	n度超流水线	(m,n)度超标量超流水
机器流水线周期	1个时钟周期	1	1/n	1/n
同时发射指令条数	1条	m	1	m
指令发射等待时间	1个时钟周期	1	1/n	1/n
指令级并行度ILP	1	m	n	m×n

超标量、超流水、超标量超流水处理机的主要性能

单发射：

- 每个周期只取一条指令、只译码一条指令，只执行一条指令，只写回一个运算结果。
- 取指部件和译码部件各设置一套。
- 可以只设置一个多功能操作部件，也可以设置多个独立的操作部件。
- 操作部件中可以采用流水线结构，也可以不采用流水线结构。

单发射处理机的指令流水线时空图



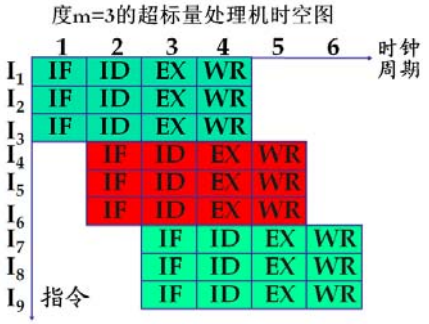
多发射：

- 每个周期同时取多条指令、同时译码多条指令，同时执行多条指令，同时写回多个运算结果。
- 需要多个取指令部件，多个指令译码部件和多个写结果部件。
- 设置多个指令执行部件，复杂的指令执行部件一般采用流水线结构。

超标量处理机

- 一个时钟周期内能够同时发射多条指令的处理机称为超标量处理机。
- 必须有两条或两条以上能够同时工作的指令流水线。

指令执行时序：

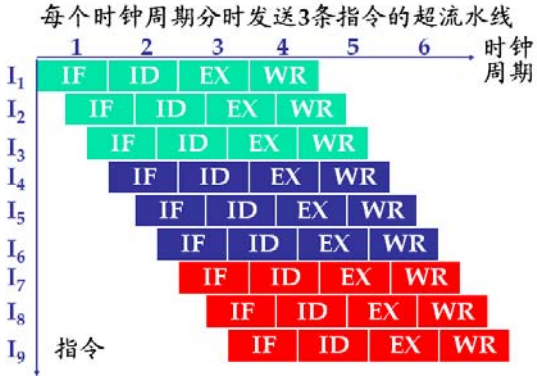


超流水线处理机

两种定义：

- 一个周期内能够分时发射多条指令的处理机称为超流水线处理机。
- 指令流水线有8个或更多功能段的流水线处理机称为超流水线处理机。

指令执行时序：每隔1/n个时钟周期发射一条指令，流水线周期为1/n个时钟周期



超长指令字处理机(VLIW)

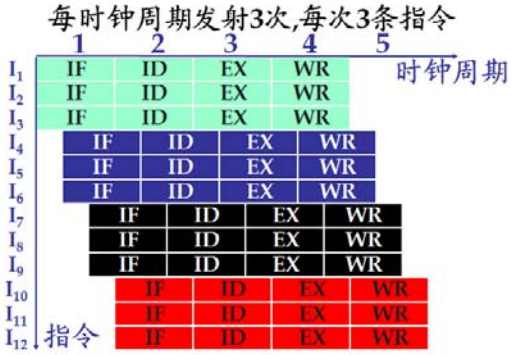
- VLIW将多个能并行执行的不相关或无关的操作先行压缩，组合在一起，形成一条有多个操作段的超长指令。

- 运行时，这条超长指令控制多个相互独立的功能部件并行操作。

超标量超流水线处理机

- 把超标量与超流水线技术结合在一起，就成为超标量超流水线处理机。
- 超标量超流水线处理机在一个时钟周期内分时发射指令n次，每次同时发射指令m条，每个时钟周期总共发射指令m · n条。

指令执行时序：



作业：5-1，5-2，5-6，5-12，5-17

第六章 并行处理机和相联处理机

6.1 并行处理机原理

并行（阵列）计算机是并行处理计算机中的一种重要结构，主要通过资源重复实现并行。

一，并行处理机的定义

并行处理机也称为阵列处理机，它是一种重复设置了多个同样的处理单元(PE)，按照一定方式把这些PE互相连接，在统一的控制部件(CU)作用下，PE各自对分配的数据并行地完成同一条指令所规定的操作，实现操作级并行的SIMD计算机。

二，并行处理机的基本构形（掌握）

根据存贮器组成方式的不同，并行处理有两种基本构形：

- 分布式存贮器的并行处理机
- 集中式共享存贮器的并行处理机

三，并行处理机的特点（了解）

与同样擅长向量处理的流水线处理机相比，它具有以下一些特点：

- 它利用的是资源重复，而不是时间重叠
- 它利用的是并行性中的同时性，而不是并发行
- 它使用简单而规整的互联网络ICN来确定PE之间的连接模式
- PE之间互连灵活，在某些专门问题上的工作效率要比流水线处理机高
- 它的专用性更强。如果习惯上把流水线处理机归属为通用计算机的话，并行处理机则被看成专用机
- 并行处理机的结构与所采用的并行算法紧密结合

6.2 互联网络

在SIMD计算机中，PE之间以及PE与存贮体都要通过互联网络（ICN）进行信息交换

ICN限定了并行处理机适用的解题算法的类型，也对整个系统的性

能产生明显影响，因此，ICN是设计重点。

1，SIMD互联网络的设计目标

- 结构不要复杂，以降低成本
- 互连要灵活，以满足算法和应用需要
- PE之间传送信息的步数尽可能少，以提高速度
- 可以用一系列规整单一的基本构件或其组合实现，模块性好，便于用VLSI实现，并满足系统的可扩充性

2，确定PE之间通信的互联网络的因素包括：

- 操作方式
- 控制策略
- 交换方法
- 网络拓扑结构等

3，网络拓扑结构

网络拓扑结构指互连网络输入、输入端可以实现连接的模式。有两种模式：

- 静态

  两个PE之间的连接是固定的，不能重新配置成与其他PE连接。

- 动态

  在动态连接模式中，PE之间的连接可以通过设置开关单元的状态重新配置。

  动态网络拓扑结构有两类：

  ✓ 单级网络

  ✓ 多级网络

  动态单级网络的一级只有有限几种连接，因此必须经过多次循环通过，才能实现任意两个PE之间的通信，故也称其为**循环网络**。

  动态多级网络由多个动态单级网络串联组合而成，以实现任意两个PE之间的通信。在此基础之上，还可以将多级网络循环使用，以实现更复杂的互连。

4，互连函数

为了反映不同互连网络的连接特性，可以用一组互连函数来定义互连网络。

设互连网络有N（$N=2^n$）个入端和N个出端，则可以用0，1，2，… N-1标识每个端点。互连函数就表示了出端号和入端号的一一对应关系，即

出端号i = f(入端号j)　i,j=0, 1, 2, … N-1

互连函数可以直接用节点之间的连线图来表示，但很繁琐，也难以体现连接上的内在规律，因此常把入端x和出端f(x)用二进制数编码，从二者的二进制数编码上找出规律。

5，基本的单级互连网络

3种基本的单级互连网络，它们是：

- 立方体（掌握）
- PM2I （了解）
- 混洗交换（了解）

(1)，立方体单级互连网络

立方体单级互连网络源于立方体结构。它有3个互连函数：

- Cube0
- Cube1
- Cube2

推广到n维情况。N个节点的立方体单级互连网络共有n=log2N种互连函数，即

$$Cube_i(P_{n-1}...P_i...P_1P_0) = P_{n-1}...\overline{P_i}...P_1P_0$$

当 n>3 时，称之为**超级立方体网络**。

超级立方体网络的最大距离为n，即反复使用单级网络，最多经过n次就可以实现任意一对入端和出端的连接，而且任意两个节点之间至少有n条不同的路径，容错性很强。

(2)，PM2I单级互连网络

PM2I是Plus-Minus 2（加减2）单级网络的简称，能实现j号PE直接与j±2号PE连接，即

$$PM2_{+i}(j) = j + 2^i \bmod N$$
$$PM2_{-i}(j) = j + 2^i \bmod N$$
$$0 \le j \le N, 0 \le i \le n, n = \log 2N$$

因此，它共有2n-1个互连函数。

- PM2I比立方体单级网络灵活。
- PM2I单级网络的最大距离为[n/2]。
- 对于n=3，最多使用两次，既可以实现任意一对入端出端的连接。

(3)，混洗交换单级互连网络

混洗交换(shuffle-Exchange)单级互连网络包含两个互连函数：

- 全混(Perfect Shuffle)
- 交换(Exchange)

全混的连接规律是：

- 把全部按编码顺序排列的PE从中间分为两半，前一半和后一半在连接至输出端时正好一一隔开，如同洗扑克牌一样。
- 其互连函数为：

$Shuffle(P_{n-1}P_{n-2}\cdots P_1P_0)=(P_{n-2}P_{n-3}\cdots P_1P_0P_{n-1})$

其中 n=log2N

Shuffle函数有两个重要特征：

- 与Cube不同，shuffle函数是不可逆的
  - ✓ 若把入端当出端，出端当入端，则原网络变成另外一个网络
- 为实现全0和全1 PE与其他PE的任意连接，还必须增加Cube0交换函数，这样就得到了全混交换单级网络

全混交换单级网络的最大距离为2n-1。

第七章 多处理机

7.1 多处理机概念

1，多处理机定义（掌握）

多处理机具有两台以上的处理机。这些多处理机在逻辑上统一的操作系统控制下，通过共享主存或输入输出子系统或高速通信网络进行通信，实现作业、任务、指令、数据各个级别的并行

2，类型或分类（掌握）

由于应用的目的和结构的不同，多处理机系统有3种类型：

- 同构型多处理机系统
- 异构型多处理机系统
- 分布式多处理机系统

3，特点（掌握）

多处理机属于MIMD系统，实现的是作业、任务之间的并行。

4，主要技术

1)如何在硬件结构上解决好处理机、存贮器和I/O系统之间的互连，实现无冲突连接

2)如何最大限度地开发系统的并行性

3)如何选择分割任务和子任务的大小，即任务的粒度大小，使并行度提高

4)如何协调好各并行执行的任务与进程之间的同步

5)如何将各个任务分配到一个或多个处理机上，解决好处理机调度、任务调度、资源分配等等

6)如果处理机发生故障，如何对系统进行重组而不是瘫痪

多处理机的硬件结构

7.2 多处理机Cache一致性

一致性问题及产生的原因： P232

7.3 多处理机操作系统

一，多处理机操作系统的特点（了解）

- 程序执行的并行性
- 分布性
- 机器间通信与同步性
- 系统的容错性

二，多处理机操作系统的3种类型（了解）：

- **主从型：**管理程序运行在一台指定的处理机上（主处理机）。
- **各自独立型：**将控制功能分散给多台处理机，共同完成对整个系统的控制。
- **浮动型：**这是一种介于主从型和各自独立型的折衷方式。管理程序可以在处理机之间浮动。主控程序可以从一台处理机转移到另一台处理机。