**第一章 计算机系统结构的基本概念** **21071003 高立扬**

1. 计算机系统结构含义，透明性

（1）含义：计算机系统结构：指**机器语言程序的设计者**或是**编译程序设计者**所看到的**计算机系统的概念性结构与功能特性**。

（2）**透明性**：一种本来存在，有差异的事物和属性，从某种角度上看又好像不存在的现象，被称为是"透明性"

2. 计算机系统结构、组成、实现之间关系

（1）区别

①结构：机器语言级的**程序员**所了解的**计算机的属性**，即**外特性**

②组成：计算机系统结构**逻辑实现**

③实现：计算机系统结构**物理实现**

（2）联系：是三个完全不同的概念，相互间有着十分密切的**依赖关系和相互影响**。

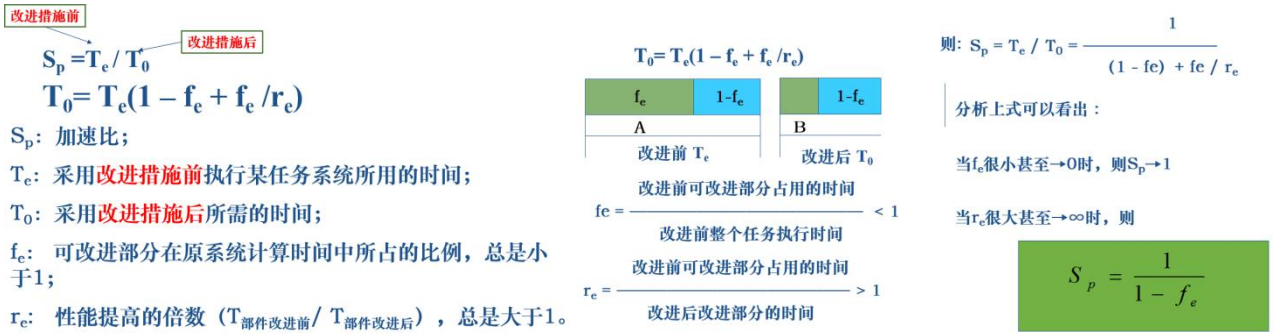
3. 分类（FLYNN）

弗林分类法是按照**数据流**和**指令流**的**多倍性**对计算机系统分类

4. 计算机系统设计的原则（三条），阿姆达尔（Amdahl）定律

（1）加速那些使用频率高的部件——提高整个计算机性能

（2）**Amdahl 定律**：系统某一部件由于采用某种改进的执行方式后，整个系统的性能提高了，衡量指标**加速比**



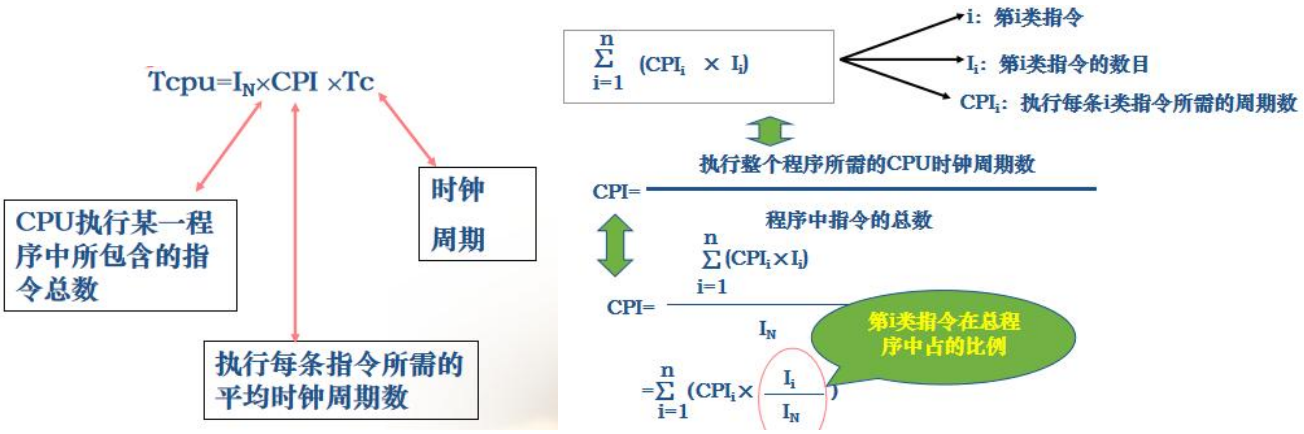

（3）程序访问局部性原理：程序往往重复使用它刚刚使用过的数据和指令。

①时间局部性：近期被访问的代码，很可能不久又将再次被访问。（循环语句）

②空间局部性： 是指地址上相邻近的代码可能会被连续的访问。（顺序语句，字符串，数组）

**5. ★★★计算机性能指标参数 （CPI、MIPS、加速比）★★★**

（1）CPU 性能（Tcpu）：CPU 执行用户程序所用的时间

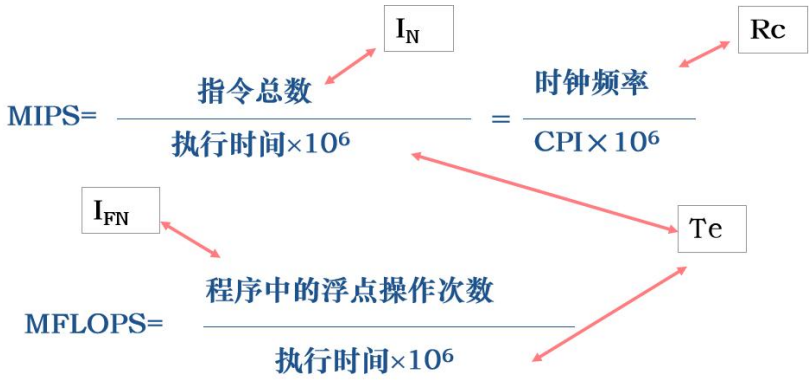



（2）CPI（cycles per instruction）：每条指令执行所需的时钟周期数

（3）MIPS（Million Instructions per Second）和 MFLOPS（Million Floating-point Operations per Second）

IPS:含义是每秒执行的指令条数。 MIPS:以百万来计量，每秒执行多少百万条指令。

MFLOPS:指每秒百万次浮点操作次数。




6. 计算机系统的层次结构


7. 性能评价结果数据的处理方法

（1）内容：如处理速度、存储容量，I/O 设备的能力，使用的方便性，系统的可靠性，可维护性，容错能力，给予系统软件的支持功能、体积等等

（2）性能指标：主频、机器字长、主存容量、运算速度、兼容性（向上向下向前向后）

**第二章 指令系统**

1. 软件兼容性要求

继承软件资产，保证软件向后兼容和向上兼容。

2. 数据类型和数据表示

（1）数据表示：指在计算机中能由硬件直接辩认,指令系统可以直接调 用的数据类型。

①基本数据表示：定点数据表示、浮点数据表示 ②高级数据表示：自定义数据表示、向量数据表示

（2）数据类型：指一组数据值的集合，还定义了可作用于这个集合上的操作集。

①基本数据类型 ②结构数据类型 ③抽象数据类型和访问指针。

3. 自定义数据表示定义、分类及优,缺点

（1）目的：为了缩短机器语言同高级语言对数据属性的说明之间的语义差距

（2）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

（3）分类

①带标志符的数据表示：描述简单数据，标志符是和每个数据值相连，存在同一存贮单元内

优点：简化了指令系统、容易检出程序编制中的错误、简化了编译程序、支持数据库系统、简化了程序设计、便于软件测试，支持应用软件开发

缺点：数据字长增加、降低了指令的微观执行速度、与其他计算机的兼容性差，硬件复杂

②数据描述符：用来描述复杂和多维数据，如向量、数组、记录等，描述符专用来描述所要访问数据的特性，它和数据字分开存储，机器经描述符形成访问每个元素的地址及其他信息，增加一级以上寻址，（描述符

或数据字）而数据字本身又是带标志符数据表示。

特点：

- 标志符要与每个数据相连，两者合存在一个存储器单元中；而描述符则和数据分开放；
- 要访问数据集中的元素时，必须先访问描述符，这就至少要增加一级寻址；
- 描述符可看成是程序一部分，而不是数据一部分，因为它是专门来描述要访问的数据的特性。

## 4. 哈夫曼概念及在计算机中应用，扩展操作码编码方法

**例1：现设一台模型机，共有7种不同的指令，使用频度如表所示。若用定长操作码表示，则需要3位。**

| 指令 | 使用频度 |
|---|---|
| I1 | 0.40 |
| I2 | 0.30 |
| I3 | 0.15 |
| I4 | 0.05 |
| I5 | 0.04 |
| I6 | 0.03 |
| I7 | 0.03 |

**操作码的信息源熵：信息源所包含的平均最短信息量.**

$H=-\sum P_i\log_2 P_i$，其中$P_i$为第i个信息源的频度

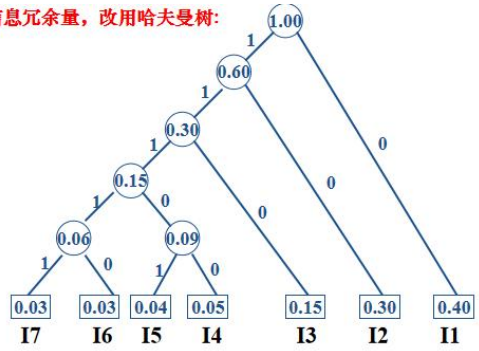
**操作码表示的平均长度**

$L=\sum l_i*P_i$　　$L_i$：第i个操作码的长度

$H=-\sum P_i\log_2 P_i=0.40*1.32+0.30*1.74+0.15*2.74+0.05*4.32+0.04*4.64+0.03*5.06+0.03*5.06=2.17$

则信息冗余量=1-H/操作码的实际平均长度=1-2.17/3=0.28(即28%)

**为减少此信息冗余量，改用哈夫曼树:**




$\sum P_i l_i=0.40*1+0.30*2+0.15*3+0.05*5+0.04*5+0.03*5+0.03*5=2.20$(位)

这种编码的信息冗余为1-2.17/2.20≈1.36%

**操作码的扩展**

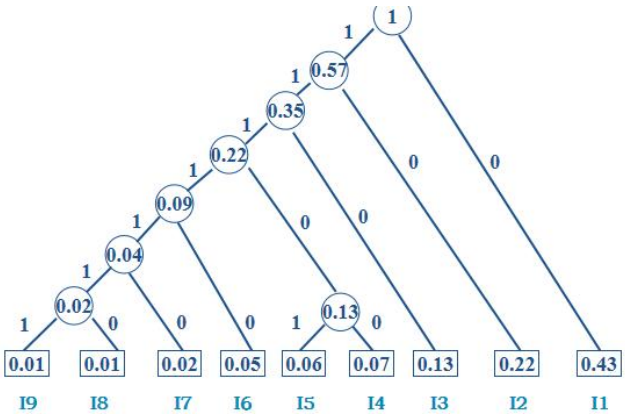
| 指　令 | 频　度 (Pi) | 操作码OP 使用哈夫曼编码 | OP长度 li | 用哈夫曼概念的扩展操作码 | OP长度 li |
|---|---|---|---|---|---|
| I1 | 0.40 | 0 | 1 | 0 0 | 2 |
| I2 | 0.30 | 1 0 | 2 | 0 1 | 2 |
| I3 | 0.15 | 1 1 0 | 3 | 1 0 | 2 |
| I4 | 0.05 | 1 1 1 0 0 | 5 | 1 1 0 0 | 4 |
| I5 | 0.04 | 1 1 1 0 1 | 5 | 1 1 0 1 | 4 |
| I6 | 0.03 | 1 1 1 1 0 | 5 | 1 1 1 0 | 4 |
| I7 | 0.03 | 1 1 1 1 1 | 5 | 1 1 1 1 | 4 |

**这种表示方法的平均长度是2.30位，信息冗余量约为5.65%。**

**例2：某模型机有9条指令，使用频率如下表。**

1. 构造Huffman树；
2. 写出Huffman的一种编码，并计算其平均码长；
3. 限定扩展编码只能有3种长度，求等长扩展下平均码长最短的扩展操作码编码方案及其平均码长。

| 指令 | 频度 | 指令 | 频度 |
|---|---|---|---|
| I1 | 0.43 | I6 | 0.05 |
| I2 | 0.22 | I7 | 0.02 |
| I3 | 0.13 | I8 | 0.01 |
| I4 | 0.07 | I9 | 0.01 |
| I5 | 0.06 | | |



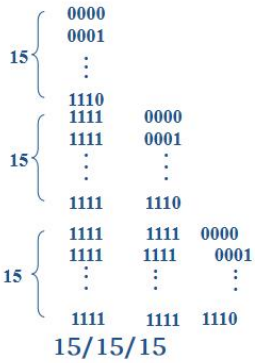

| 指　令 | 频　度 (Pi) | 操作码OP 使用哈夫曼编码 | OP长度 li | 用哈夫曼概念的扩展操作码 | OP长度 li |
|---|---|---|---|---|---|
| I1 | 0.43 | 0 | 1 | 0 0 | 2 |
| I2 | 0.22 | 1 0 | 2 | 0 1 | 2 |
| I3 | 0.13 | 1 1 0 | 3 | 1 0 | 2 |
| I4 | 0.07 | 1 1 1 0 0 | 5 | 1 1 0 0 | 4 |
| I5 | 0.06 | 1 1 1 0 1 | 5 | 1 1 0 1 | 4 |
| I6 | 0.05 | 1 1 1 1 0 | 5 | 1 1 1 0 | 4 |
| I7 | 0.02 | 1 1 1 1 1 0 | 6 | 1 1 1 1 0 0 | 6 |
| I8 | 0.01 | 1 1 1 1 1 1 0 | 7 | 1 1 1 1 0 1 | 6 |
| I9 | 0.01 | 1 1 1 1 1 1 1 | 7 | 1 1 1 1 1 0 | 6 |

**使用哈夫曼编码平均码长**

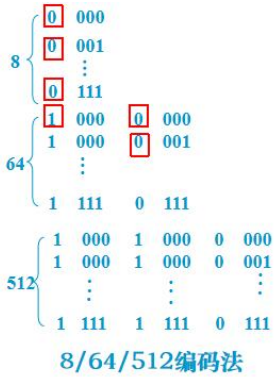
$L=\sum p_i l_i=0.43*1+0.22*2+0.13*3+0.07*5+0.06*5+0.05*5+0.02*6+0.01*7+0.01*7=2.42$(位)

**用哈夫曼概念的扩展操作码平均码长**

$L=\sum p_i l_i=0.43*2+0.22*2+0.13*2+0.07*4+0.06*4+0.05*4+0.02*6+0.01*6+0.01*6=2.52$(位)




15/15/15




8/64/512编码法

5. 指令系统编码方法

（1）**正交法**

• 指令中的每个分段（包括操作码、操作数地址等） 相互独立，操作数地址的编码同操作码无关，反之亦然 。

• 优点：对流水机特别适用，微程序控制数量减少。

（2）**整体法**

• 指令中各个分段在译码时相互有关，操作码同操作数地址的分界线并不清楚。

• 优点：可以把使用频度高的操作码同操作数地址码组合起来，加以缩短优化，而使用频度低的可以较长些，这样可以节省存贮容量。

• 缺点：在用微程序控制时，微程序数量较多，需要有较大的微程序存贮器。

（3）**混合法**

• 这种方法把上两种方法的优点结合起来。

6. 程序定位方式

（1）程序定位：指令和数据中的逻辑地址（相对地址）转变成主存物理地址（绝对地址）的过程。

（2）分类：

①**直接定位方式** ：直接使用实际主存物理地址来编写或编译程序，目前大多不用这种方式。

②**静态定位方式**：专门用装入程序来完成，一旦装入主存就不能再变动了，这种方式实现简单，不需要增加硬件设备，但不够灵活，主存利用率不高，多个用户不能共享主存。

③**动态定位方式**：利用类似变址寻址方法，由硬件支持完成。只把主存的起始地址装入该程序对应的基址寄存器中，指令的地址不需全部修改。

• 优点：在程序执行时由硬件形成主存物理地址，主存利用率高，多个用户可以共享同一个程序段，支持虚拟存储器实现。

• 缺点：需要硬件支持，实现的算法比较复杂。

7. 两种指令系统风格，特点（RISC、CISC）

**(一)复杂指令集计算机（CISC)**

**特点：**

**✓指令的控制执行是采用微程序控制技术，有专用的寄存器。**

**✓控制器十分复杂，占用了大量CPU芯片面积，有些复杂指令用的很少，难以用优化编译生成高效目标代码。**

**✓处理器的执行效率不高。**

**✓指令系统与软件之间语义差别越来越大，软件设计任务十分繁重，整个设计风格不是十分经济有效的。**

**缺点：**

**指令系统庞大 → 硬件复杂 → 执行速度低**

**→ 编译程序复杂、长 → 部分指令使用效率低**

**(二) 精简指令系统计算机（RISC)**

**基本思想：通过减少指令总数和简化指令的功能来降低硬件设计的复杂程度，提高指令执行速度，使指令简单，有效可行。**

**PENTIUM处理器（二者结合）：**

**RISC特征：某些指令以硬连线来实现，并能在一个时钟周期执行完；**

**CISC特征：用微代码实现，需要2-3个时钟周期的执行时间，有多种寻址方式，多种指令长度，为数不多的通用寄存器。**

8. 简述 RISC 的设计思想起源和主要技术

（1）RISC 设计思想的起源

①20%-80%定律 ②系统设计中硬件和软件之间折衷 ③VLSI 工艺技术发展

（2）RISC 主要技术

①流水线结构和指令调度：RISC 主要特点之一是充分提高流水线效率。

②寄存器窗口（寄存器窗口技术，8 个）：显著减少过程调用和返回执行时间、执行的指令条数和访问存储器的次数。

③优化编译技术

• RISC 指令系统条数少，简单对称，这减轻了编译程序的负担.；

• RISC 寻址方式简单，只有 LOAD 和 STORE 指令访问存储器。其它操作均在通用寄存器中进行，这简化了寻址方式和访存操作 ；

• 大多数指令在一个周期内完成，为优化编译器进行调整指令流序列，减少相关，提高并行度带来了方便；

## 第三章 存储系统

1. 存储系统三个特性（局部性、一致性、包含性）

（1）局部性：时间&空间

（2）包含性：在容量大的存储器中，一定能找到上层存储信息的副本

（3）一致性：副本修改，以保持同一信息的一致性

2. 存储层次结构概念和性能参数（T，S，C）

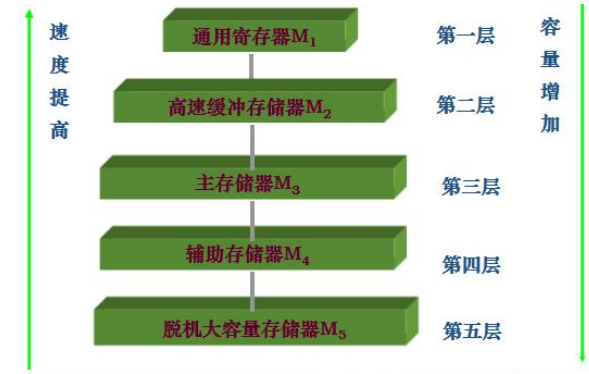
**存储层次结构概念和性能参数（T，S，C）**

**存储系统：两个或两个以上的速度、容量、价格不同的存储器采用硬件，软件或软、硬件结合的办法联接成一个系统。**

**目的：存放计算机系统中所需要处理的程序与数据 。**

**计算机存储系统三个基本参数：**

- **存储容量S　以字节数表示，单位为B、KB、MB、GB、TB等。**
- **存储器速度T　存储器访问周期，与命中率有关。**
- **存储器价格C　表示单位容量的平均价值单位为＄C/bit或＄C/KB。**




每级存储器的性能参数可以表示为Ti，Si，Ci。存储系统的性能可表示为：$T_i<T_{i+1}$；$S_i<S_{i+1}$；$C_i>C_{i+1}$。

3. 提高存储器频带宽度方法

（1）频带宽度：单位时间内所能访问的数据量

（2）解决频带平衡的方法

①多个存储器并行工作，并用并行访问和交叉访问等方法；

②设置各种缓冲存储器；　③采用 Cache 存储系统。

（3）并行存储器

①多体并行访问存储器

- 优点：简单、容易。
- 缺点：访问的冲突大。
- 主要冲突：
  - 取指令冲突（条件转移时）
  - 读操作数冲突（需要的多个操作数不一定都存放在同一个存储字中）
  - 写数据冲突（必须凑齐 n 个数才一起写入存储器）
  - 读写冲突（要读出的一个字和要写入的一个字处在同一个存储字内时，无法在一个存储周期内完成）。

②多体并行交叉访问存储器

- 高位交叉访问
  - 缺点：由于程序局部性的原理，近期所用到的指令和数据往往都集中在一个体内，就会出现并行访问冲突，只有一个存储模块在不停地忙碌，其他空闲。只有当指令跨越两个存储模块时，才并行工作。
  - 优点：扩大存储容量非常方便。如果在多任务或多用户的应用状态下，可以将不同的任务分别存放在不同的体内，减少了访问冲突，发挥了并行访问的优点。

- 低位交叉访问

4. CACHE 引入目的、特点（和虚拟存储器比）、需解决的问题

高速缓冲存储器：存在于主存与 CPU 之间的一级存储器，由静态存储芯片(SRAM)组成，容量比较小但速度比主存高很多，接近于 CPU 的速度。

Cache 的功能**：存放那些近期需要运行的指令与数据。

目　　的**：提高 CPU 对存储器的访问速度。

5. CACHE-主存地址映象变换概念？几种主要方式（全相联、直接、组相联）特点计算

（1）三种

①全相联　　优点：命中率较高，Cache 的存储空间利用率高；　缺点：线路复杂，成本高，速度低。

②直接　　优点：简单　　缺点：命中率低。

③组相联　　优点：速度快，命中率高；

（2）思考

①**块冲突概率最低**的 cache 地址映像方式是：**全相连映像**

②**块冲突概率最高**的 cache 地址映像方式是：**直接相连映像**

6. 常用替换算法特点及相关计算

（1）随机法：(Random, RAND)

（2）先进先出法(First-In First-Out, FIFO)

（3）近期最少使用法(Least Recently Used, LRU)

（4）最久没有使用法（Least Frequently Used, LFU）

7. CACHE 写操作的更新策略，CACHE 的性能（命中率，加速比，平均访问时间）

**（1）全写法，亦称写直达法(WT 法——Write through)**

①定义：在对 Cache 进行写操作的同时，也对主存该内容进行写入。

②优点：Cache 及主存与内容同时更新。所以**一致性保持的比较好，可靠性比较高，操作过程比较简单**。如果 Cache 发生错误，可以从主存得到纠正。

③缺点：全写法**每次写操作都要访问主存**，所以，**写操作的速度得不到改善**，仍然是访主存的速度。

④改善：常采用一个高速的小容量的缓冲存储器，将要写入的数据和地址先写到这个缓冲存储器，使得 Cache 可以尽快的执行下次访问，然后再将缓冲存储器的内容写入主存。

**（2）写回法(WB 法——Write back)**

①在 CPU 执行写操作时，只写入 Cache，不写入主存；需要替换时，把修改过的块写回主存。

②优点：Cache 的**速度比较高**，Cache 的**命中率比较高**，所以 Cache 与主存之间的**通信量大大降低**。

③缺点：有一段时间 Cache 内容与主存内容不一致，所以**可靠性比全写法差**，而且**控制操作比较复杂**。

④改善：写回法在替换时要将一块内容写回主存；此时 CPU 不能继续访问 Cache 及主存，可能处于等待状态。为改善这一缺陷，可以设置一个高速的数据缓冲存储器，先将写入的数据与地址存入此缓冲区，以使 CPU 可以继续工作。而缓冲存储器可以与 CPU 的处理工作并行，将数据写入主存。

（3）写不命中

**问题：是否把包括所写字在内的一个块从主存读入cache?**


8. CACHE 命中率影响因素

①Cache 的容量对命中率的影响：**容量越大**则**命中率越高**。当容量由很小开始增加时命中率增加的比较明显当容量达到一定程度，容量增加命中率改善的并不大。

②Cache 块的大小对命中率的影响：当块的**容量加大命中率**明显的**增加**，但增加到**一定值后**反而出现块增加**命中率下降**的现象。这是因为块容量大到一定程度，进入块内的数据，已**不符合程序局部性规律**了；**块越大**在一定量的 **Cache 中包含的块数**就越小，则**命中率就降低**了。

③地址映像方式对命中率的影响：

（1）**直接相联法命中率**比较**低**。

（2）**全相联方式命中率**比较**高**，但**难以实现**。

（3）**组相联方式中**，主要是**分组的数目对命中率的影响比较明显**。由于主存与 Cache 的组之间是直接相联方式，当组数分的越多，则命中率就要下降，当组数比较少时这种影响不明显，当组数大到一定程度，则影响就很大。

3）Cache系统的加速比

等效的访问周期为T

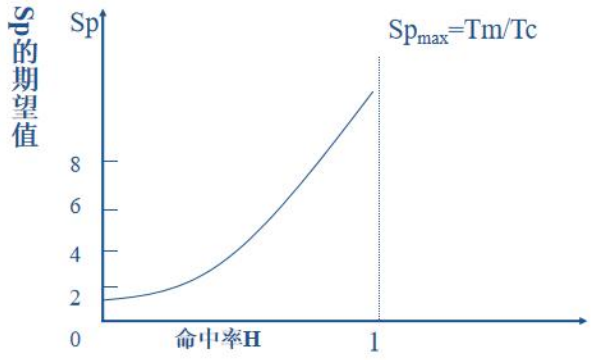
$$T = H_c T_c + (1 - H_c) T_m$$

Tc：Cache的访问周期；

Tm：主存储器的访问周期；

Hc：Cache的命中率

Cache加速比与命中率的关系




Cache系统的加速比$S_p$

$$S_p = \frac{T_m}{T} = \frac{T_m}{H_c T_c + (1 - H_c) T_m} = \frac{1}{H_c \frac{T_c}{T_m} + (1 - H_c)}$$

命中率越高，加速比越大。当Hc→1时，

$$S_p \to \frac{T_m}{T_c}$$

存储系统的访问效率：指高一级存储器的访问速度(容量小速度高的一级)与系统等效的访问速度之比。

$$e = \frac{T_c}{T} = \frac{1}{H_c + (1 - H_c) \frac{T_m}{T_c}}$$

9. 存储保护

（1）加界保护法：在 CPU 中设置了多个界限寄存器，由系统软件经特权指令指定，禁止越界。

（2）键保护方式：每次访问主存，首先进行键号比较，如果键号相等才允许访问。如同一把钥匙开一把锁。存放键与程序键键号的分配，由操作系统完成。

（3）环保护方式：将系统程序和各用户程序按其功能的性质和要求分为几个级别，分别授予不同的权限，如系统程序对安全的要求比较高，授权级别就比较高，用户程序的级别就可以低些，各级的关系如图所示。

## 第四章 流水线技术

提高处理速度和系统使用效率的三条途径：时间重叠、资源重复、资源共享

CPU 工作方式：顺序、重叠、流水

1. 流水概念，流水分级，分类

（1）概念：流水线技术是将一个**重复的时序过程**分成若干个**子过程**，每个子过程都可**有效的**在其**专用功能段**上和其它子过程**同时执行**的一种技术

（2）分级：**操作部件级、指令级、处理机级或宏流水线级**

（3）分类

①按功能分类：单功能、多功能　　②按工作方式：静态、动态　　③连接方式：线性、非线性

## 2. 流水线性能指标及分析、计算（Tp、η、Sp 等）

（1）**吞吐率 Tp**：单位时间能流出的任务数　　**Tp = 任务数 / 总时间**

（2）**效率 η**：整个运行时间里，流水线的设备有多少时间是真正用于工作的　　**η = 格子数 /(总时间*状态数)**

（3）**加速比 Sp**：m 段流水线的速度与等效的非流水线的速度之比。　　**Sp = 格子数 / 总时间**

（4）解决瓶颈方法 ①瓶颈段细分　②重复设置瓶颈流水段

## 3. 流水线三种冲突（资源，数据，控制相关）的概念和处理方法

（1）概念：相近指令出现某种关联而使其不能在指定的时钟周期执行

（2）分类：资源相关（结构相关）、数据相关（RAW,WAR,WAW）、控制相关

（3）后果：错误的执行结果、流水线可能出现停顿，导致性能降低

（4）解决办法

①资源相关：停顿　　②数据相关：时间推后、旁路技术、专用通路技术、定向技术

③控制相关：加快和提前行程条件码、静态转移预测技术、猜测法、预取转移目标、加快短循环程序处理

## 4. 流水机器的中断处理

①不精确断点法　②精确断点法

## 5. 线性和非线性流水线的调度

（1）先进的流水调度方法 - 动态调度

①静态调度：借助软件对指令执行顺序进行调度，以减少由于流水线中存在相关冲突而引起流水线的停顿时间。目前比较流行。

②动态调度：通过硬件重新安排指令的执行顺序以减少流水的停顿。

③优点：

- 能处理某些在编译时无法知道的相关情况；
- 能简化编译程序设计；
- 使代码有可移植性。

④缺点：相应的硬件较为复杂

（2） 非线性流水线的冲突及调度*

①预约表　　②禁止表　　③冲突向量　④状态转移图

## 6. 一个周期能完成多条指令的计算机（多发射结构的 RISC）、三种超级计算机

（1）对一个周期能发射多条指令计算机有超标量、超流水、超长指令字计算机，此外还有数据流计算机也属于多发射结构。

（2）超标量计算机：每个时钟周期内能同时发射多条指令的计算机。每拍启动 3 条指令，则并行度=3

（3）超长指令字计算机：把多个能并行执行的操作组合在一起。每拍启动一条长指令，执行 3 个操作，相当于 3 条指令，并行度=3

（4）超级流水线计算机：每个时钟周期内能分时发射（启动）多条指令的计算机。并行度=3：每 1/3 拍启动一条指令

（5）超标量超流水计算机 ：结合了二者，每个时钟周期总共要发射指令 m*n 条

## 7. 向量的处理方法，向量处理机的结构分类

（1）处理方法

①水平处理法　　②垂直处理法　　③分组纵横处理法

（2）向量处理机的结构：由向量数据表示和流水线技术相结合构成的向量流水处理机。

8. 增强向量处理功能的方法，特别是链接技术

（1）处理办法：全并行、向量冲突、功能部件冲突、功能部件&向量冲突

（2）链接技术：利用向量指令间存在的先写后读的数据相关性，把向量寄存器 Vi 既作为源 R，又作为结果 R；将两条或多条向量指令链成一条链子，来加快向量指令序列的执行速度、提高向量操作的并行性和功能部件流水效能的技术。

**第五章 并行处理机和多处理机**

1. 并行性概念、粒度

（1）概念：在同一时刻或同一时间间隔内完成多个性质相同或不同的任务。

①同时性（simultaneity）:指两个或多个事件在同一时刻发生在多个资源中。

②并发性（concurrency）:指两个或多个事件在同一时间间隔内发生在多个资源中。

（2）并行性粒度：每次并行处理的规模大小

G=TW/TC

TW：所有处理器进行计算的时间总和；

TC：所有处理器进行通信的时间总和。（设系统共有 P 个处理器）

当 TC 较大时，通信量大，则 G 较小处理粒度较细。反之对于粗粒度的并行，通信量较小。

（3）小结

①**细粒度**并行性常在指令级或循环级上借助于并行化或向量化编译器来进行开发的。

②任务或作业步骤（过程级）**中粒度**并行性开发需要程序员和编译器的共同作用。

③开发程序作业级的**粗粒度**并行性主要取决于高效的操作系统和所用算法的效率。

④共享变量通信常用于支持中、细粒度计算。消息传递型多计算机用于中粒度和粗粒度的计算。通常情况下，**粒度越细，并行性潜力越大，通信和调度的开销也越大**。细粒度能提供较高的并行度，但与粗粒度计算相比，其通信开销也较大。大规模并行性通常是在细粒度级上开发。如：SIMD 或 MIMD 计算机上开发的数据并行性。

2. 单机并行发展的 3 条技术途径

①**时间重叠**：在并行性概念中引入时间因素。让多个处理过程在时间上相互错开，轮流重叠地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。

②**资源重复**：并行性概念中引入空间因素。通过重复设置的硬件资源来提高系统可靠性或性能。

③**资源共享**：利用软件的方法让多个用户按一定时间顺序轮流地使用同一套资源，以提高其利用率，这样相应地提高整个系统的性能。

3. 阵列处理机分类

阵列处理机根据存贮器采用的组成方式不同分成两种基本构成。

①分布存贮 ②集中式共享存贮

4. 常用基本互联函数计算方法

（1）概念：一种由开关元件，按照一定的拓扑结构和控制方式构成的，用来实现计算机系统内部的多个功能部件或者是多个处理机之间的相互连接。

（2）互连函数

①输入输出对应表示法 ②循环表示法 ③函数表示法

（3）基本互连函数

①恒等置换 ②交换置换 ③方体置换 ④均匀洗牌置换 ⑤蝶式置换 ⑥移数置换 ⑦加减 2^i 置换

5. 三种单级互连网及两种多级互连网功能、工作方式

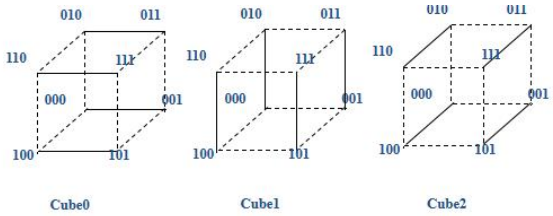
（1）单级

①立方体网

立方体单级网循环表示为：

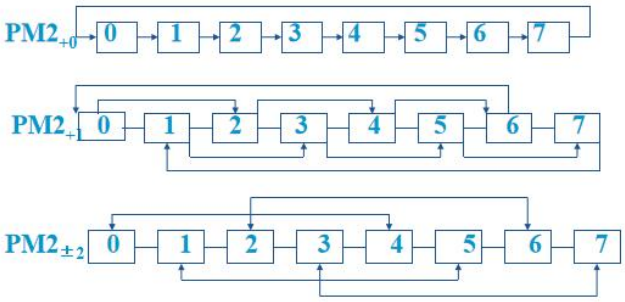
Cube0：（0　1）（2　3）（4　5）（6　7）　　Cube1：（0　2）（1　3）（4　6）（5　7）

Cube2：（0　4）（1　5）（2　6）（3　7）




立方体网共有n=$Log_2N$种互连函数，即为

$CUBE_K(X_{n-1}X_{n-2}...X_{k+1}X_kX_{k-1}...X_1X_0)=X_{n-1}X_{n-2}...X_{k+1}\overline{X_k}X_{k-1}...X_1X_0$　其中$X_k$为输入端标号的第K 位二进制代号,且0<=k<=n-1




PM2I互连网连接图

②PM2I 单级互连网

PM2I 单级网结点间的互连函数关系为加减 2i 置换,

对于 N=8，PM2I 单级网共有 2*3=6 个互连函数，循环表示为：

PM2+0：（0　1　2　3　4　5　6　7）

PM2-0：（7　6　5　4　3　2　1　0）

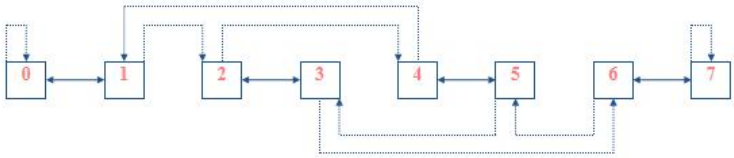
PM2+1：（0　2　4　6）（1　3　5　7）

PM2-1：（6　4　2　0）（7　5　3　1）

PM2±2：（0　4）（1　5）（2　6）（3　7）

③ 混洗交换单级互连网络

全混洗（Perfect shuffle）、交换（Exchange）



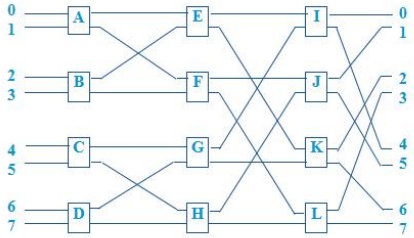

全混洗交换单级网

（2）多级互连网

➢多级立方体网

特点：第i级（0≤i≤n-1）控制信号为“1”时，处于交换状态，实现的是$cube_i$互连函数，当该信号为“0”时，相应单元处于直连状态代码不变，它们都采用两个功能（直接、交换）的交换单元。常用的多级立方体网有STARAN网，间接二进制n方体网等。

交换网络：级控
移数网络：部分级控



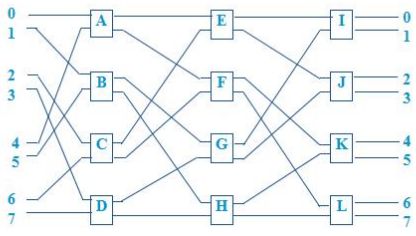

N=8间接二进制n方体互连网

三级STARAN交换网络实现的入出端连接

| | | 级控制信号（$f_2$ $f_1$ $f_0$） | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
| 入端号 | 0 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| | 1 | 1 | 0 | 3 | 2 | 5 | 4 | 7 | 6 |
| | 2 | 2 | 3 | 0 | 1 | 6 | 7 | 4 | 5 |
| | 3 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 |
| | 4 | 4 | 5 | 6 | 7 | 0 | 1 | 2 | 3 |
| | 5 | 5 | 4 | 7 | 6 | 1 | 0 | 3 | 2 |
| | 6 | 6 | 7 | 4 | 5 | 2 | 3 | 0 | 1 |
| | 7 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

➢多级混洗交换网络

又称omega网络，由n级相同的网络组成，每一级都包含一个全混拓扑和随后一列$2^{n-1}$个四功能交换单元，（直连、交换、上播、下播），采用单元控制方式。




N=8 多级混洗交换网