

第一章 计算机系统结构概论

考点一：计算机系统的层次结构

1、计算机系统层次结构由高到低依次为：应用语言机器级、高级语言机器级、汇编语言机器级、操作系统机器级、传统机器语言机器级、和微程序机器级。

考点二：计算机系统结构、组成和计算机实现

1、计算机系统结构也称计算机系统的体系结构，指的是传统机器语言机器级的系统结构。它是软件和硬件/固件的交界面，是机器语言、汇编语言程序设计者或编译程序设计者看到的机器物理系统的抽象。

2、计算机组成指的是计算机系统结构的逻辑实现，包括机器级内的数据流和控制流的组成以及逻辑设计等。

3、计算机实现指的是：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、成板的划分等。

4、计算机结构、组成和实现

计算机系统结构	确定指令系统中是否要设乘法指令 主存容量与编址方式（按位、按字节还是按字访问等）的确定
计算机组成	乘法指令的实现 为达到性能价格要求，主存速度应为多少，逻辑结构是否采用多体交叉
计算机实现	乘法器、加法移位器的物理实现 主存器件的选定、逻辑设计、微组装技术的使用装技术的确定和选择

考点三：计算机系统的软、硬件取舍及定量设计原理

1、软、硬件功能分配的基本原则：

原则一：高性能价格比，主要从实现费用、速度和其它性能要求来综合考虑。

原则二：要考虑到准备采用和可能采用的组成技术，使它尽可能不要过多或不合理地限制各种组成、实现技术的采用。

原则三：不能仅从“硬”的角度去考虑如何便于应用组成技术的成果和发挥器件技术的进展，还应从“软”的角度把为编译和操作系统的实现以及为高级语言程序的设计提供更多、更好的硬件支持放首位。

2、程序访问的局部性定律：包括时间上和空间上的两个局部性。

3、软件的可移植性：指软件不修改或只经少量修改就可由一台机器移到另一台机器上运行，同一软件可应用于不同的环境。

4、实现软件移植的技术主要有：

- 统一高级语言；
- 采用系列机；
- 模拟和仿真；

考点四：系统结构中的并行性开发及计算机系统的分类

1、并行开发的途径：时间重叠、资源重复和资源共享。

2、计算机系统的分类

(1) 弗林分类方法：

单指令单数据流 (SISD) ——单处理器计算机；

单指令多数据流 (SIMD) ——阵列处理机；

多指令单数据流 (MISD) ——实际上很少见；

多指令多数据流 (MIMD) ——多处理机；

弗林分类方法反映大多数计算机的并行工作方式和结构特点，作用范围较广，但只能对控制流机器分类。

(2) 1978 年，美国的库克提出用指令流和执行流及多倍性来描述计算机系统总控制器的结构特点。

单指令流单执行流 (SISE) ——单处理机系统。

单指令流多执行流 (SIME) ——带多操作部件的处理机。

多指令流单执行流 (MISE) ——带指令级多道程序的单处理机。

多指令流多执行流 (MIME) ——典型的多处理机系统。

第二章 数据表示寻址方式指令系统

考点一：数据表示

- 1、数据表示：指能由机器硬件直接识别和引用的数据类型。由硬件实现的数据类型。
- 2、数据结构：面向计算机系统软件、面向应用领域所需处理的数据类型。由软件实现的数据类型。
- 3、自定义数据表示包括**标志符数据表示**和**数据描述符**两类。

(1) 标志符数据表示

类型标志	数据值
------	-----

- 为了缩短高级语言和机器语言的语义差距，计算机中的每个数据都带有如图类型标志。
- 用于指明数据类型（如二进制整数、浮点数、字符串还是地址等），将**数据类型和数据值本身**直接联系在一起。
- 对高级程序员透明。

(2) 数据描述符

为进一步减少标志符所占的存储空间，发展出了数据描述符。

与标志符的区别：

- 标识符是和每一个数据相连的，合存在一个存储单元中，描述单个数据的类型特征。
- 数据描述符是和数据分开存放的，用来描述所要访问的数据是整块的还是单块的，访问该数据块或数据元素需要的地址以及其他信息等。

4、数据表示的原则

原则 1：看系统的效率是否提高，是否减少了实现时间和所需的存储空间；

原则 2：看引入数据表示后，其通用性和利用率是否提高；

考点二：寻址方式

1、寻址方式指的是指令按什么方式寻找（访问）到所需的操作数或信息的。

2、寻址方式的三种面向

- 面向主存：主要访问内存，少量访问寄存器
- 面向通用寄存器：多数在寄存器，少量在内存
- 面向堆栈：主要在堆栈，可减轻编译负担

3、寻址方式在指令中的指明（两种指明方式）：

(1) 占用操作码中的某些位来指明

(2) 不占用操作码，而是在地址码部分专门设置寻址方式位字段指明

相对第二种方式更灵活，操作码短，但需要专门的寻址方式位字段。

4、逻辑地址：是程序员用的地址。

物理地址：是程序在主存中的实际地址。

5、程序在主存中的定位技术

- 静态定位
- 动态再定位
- 虚实地址映像表

6、物理主存中信息的存储分布

- (1) 任意存储：降低了存储访问速度。
- (2) 按整数边界存储：存放的地址必须是该信息宽度的整数倍。

会存在一定的存储空间的浪费，但访问每个信息时，均只需一个主存周期。即也体现了速度和价格的权衡。

考点三：指令系统设计的基本原则

1、指令系统的设计包括：操作类型、寻址方式和具体操作内容和指令格式设计。

2、指令系统设计步骤：

- 根据应用，初拟出指令的分类和具体的指令。
- 试编出用该指令系统设计的各种高级语言的编译程序。
- 对各种算法编写大量的测试程序并进行模拟测试，看指令系统的操作码和寻址方式效能是否较高。
- 将程序中高频出现的指令串复合改成一条强功能新指令，即改用硬件方式来实现。而将频度很低的指令改成用基本的指令组成的指令串来实现，即用软件来实现。

3、指令类型一般分为两类：

- 非特权型指令：主要供应用程序员使用，也可以供系统程序员使用。包括算术逻辑运算、数据传送、浮点运算、字符串等。
- 特权型指令：只供系统程序员使用，用户无权使用。

4、指令系统设计的基本原则

- (1) 规整性；
- (2) 对称性；
- (3) 独立性和全能性；
- (4) 正交性；
- (5) 可组合性；
- (6) 可扩充性；

考点四：指令系统的设计和优化

1、指令是由**操作码**和**地址码**两部分组成的。

2、**操作码的优化**（哈夫曼压缩概念）表示主要是为了缩短指令字长，减少程序总位数及增加指令字能表示的操作信息和地址信息。

3、指令格式的优化：为了不降低指令的存取速度，就要维持指令字按整数边界存储。

4、指令字格式优化的措施概括起来，包括以下几点：

- (1) 采用扩展操作码，并根据指令的频度 P_i 的分布情况，选择合适的编码方式，以缩短操作码的平均码长。
 - (2) 采用诸如基址、变址、相对、寄存器、寄存器间接、段式存放、隐式指明等多种寻址方式，以缩短地址码的长度，并在有限的地址长度内提供更多的地址信息。
 - (3) 采用 0123 等多种地址制，以增强指令的功能，这样从宏观上就越能缩短程序的长度，并加快程序的执行速度
 - (4) 在同种地址制内，在采用多种地址形式，如寄存器，寄存器-寄存器，寄存器-主存、主存-主存等让每种地址字段可以有多种长度，且让长操作码与短地址码进行组配。
 - (5) 在维持指定字在存储器中按整数边界存储的前提下，使用多种不同的指令字长度。
- 将以上措施综合使用，可以使信息冗余量减少，操作数的寻址灵活，操作码的备用码点数增多，有利于以后对指令系统进行扩充。

考点五：指令系统的发展和改进

一种是**复杂指令系统计算机 CISC**，通过增强原有指令的功能，以及设置更为复杂的新指令已取代原先由软件子程序完成的功能，实现软件功能的硬化。

按 CISC 方向发展和改进指令系统可以从**面向目标程序**，**面向高级语言**，**面向操作系统**的优化实现三个方面来叙述。

一种是**精简指令系统计算机 RISC**，通过减少指令总数和简化指令功能来降低硬件设计的复杂度，提高指令的执行速度。

获取完整资料 请下载 APP 或关注公众号



扫码下载 app



扫码关注公众号