详解计算机体系结构

 计算机体系结构是指那些对程序员可见的系统属性，还包括设计思想与体系结构。

    计算机体系结构（ComputerArchitecture）是程序员所看到的计算机的属性，即概念性结构与功能特性。按照计算机系统的多级层次结构，不同级程序员所看到的计算机具有不同的属性。一般来说，低级机器的属性对于高层机器程序员基本是透明的，通常所说的计算机体系结构主要指机器语言级机器的系统结构。经典的关于“计算机体系结构（computerarchitecture）”的定义是1964年C.M.Amdahl在介绍IBM360系统时提出的，其具体描述为“计算机体系结构是程序员所看到的计算机的属性，即概念性结构与功能特性”

　　**计算机体系结构-基本概念**

****

计算机体系结构

　　计算机体系结构就是指适当地组织在一起的一系列系统元素的集合，这些系统元素互相配合、相互协作，通过对信息的处理而完成预先定义的目标。通常包含的系统元素有：计算机软件、计算机硬件、人员、[数据库](http://www.kokojia.com/list/6.html%22%20%5Co%20%22%E6%95%B0%E6%8D%AE%E5%BA%93%22%20%5Ct%20%22_blank)、文档和过程。其中，软件是程序、数据结构和相关文档的集合，用于实现所需要的逻辑方法、过程或控制；硬件是提供计算能力的电子设备和提供外部世界功能的电子机械设备(例如传感器、马达、水泵等)；人员是硬件和软件的用户和操作者；数据库是通过软件访问的大型的、有组织的信息集合；文档是描述系统使用方法的手册、表格、图形及其他描述性信息；过程是一系列步骤，它们定义了每个系统元素的特定使用方法或系统驻留的过程性语境。

　　**计算机体系结构-8种属性**

****

计算机体系结模型

　　1•机内数据表示：硬件能直接辨识和操作的数据类型和格式

　　2•寻址方式：最小可寻址单位、寻址方式的种类、地址运算

　　3•寄存器组织：操作寄存器、变址寄存器、控制寄存器及专用寄存器的定义、数量和使用规则

　　4•指令系统：机器指令的操作类型、格式、指令间排序和控制机构

　　5•存储系统：最小编址单位、编址方式、主存容量、最大可编址空间

　　6•中断机构：中断类型、中断级别，以及中断响应方式等

　　7•输入输出结构：输入输出的连接方式、处理机/存储器与输入输出设备间的数据交换方式、数据交换过程的控制

　　8•信息保护：信息保护方式、硬件信息保护机制。

　　**计算机体系结构-发展历程**

****

虚存-磁盘存储器地址转换

**计算机系统已经经历了四个不同的发展阶段**

**第一阶段**

　　60年代中期以前，是计算机系统发展的早期时代。在这个时期通用硬件已经相当普遍，软件却是为每个具体应用而专门编写的，大多数人认为软件开发是无需预先计划的事情。这时的软件实际上就是规模较小的程序，程序的编写者和使用者往往是同一个(或同一组)人。由于规模小，程序编写起来相当容易，也没有什么系统化的方法，对软件开发工作更没有进行任何管理。这种个体化的软件环境，使得[软件设计](http://www.kokojia.com/list/320.html)往往只是在人们头脑中隐含进行的一个模糊过程，除了程序清单之外，根本没有其他文档资料保存下来。

**第二阶段**

　　从60年代中期到70年代中期，是计算机系统发展的第二代。在这10年中计算机技术有了很大进步。多道程序、多用户系统引入了人机交互的新概念，开创了计算机应用的新境界，使硬件和软件的配合上了一个新的层次。实时系统能够从多个信息源收集、分析和转换数据，从而使得进程控制能以毫秒而不是分钟来进行。在线存储技术的进步导致了第一代数据库管理系统的出现。计算机系统发展的第二代的一个重要特征是出现了“软件作坊”，广泛使用产品软件。但是，“软件作坊”基本上仍然沿用早期形成的个体化软件开发方法。随着计算机应用的日益普及，软件数量急剧膨胀。在程序运行时发现的错误必须设法改正；用户有了新的需求时必须相应地修改程序；硬件或[操作系统](http://www.kokojia.com/s1071/)更新时，通常需要修改程序以适应新的环境。上述种种软件维护工作，以令人吃惊的比例耗费资源。更严重的是，许多程序的个体化特性使得它们最终成为不可维护的。“软件危机”就这样开始出现了。1968年北大西洋公约组织的计算机科学家在联邦德国召开国际会议，讨论软件危机课题，在这次会议上正式提出并使用了“软件工程”这个名词，一门新兴的工程学科就此诞生了。



复杂计算机体系结构图

**第三阶段**

　　计算机系统发展的第三代从20世纪70年代中期开始，并且跨越了整整10年。在这10年中计算机技术又有了很大进步。分布式系统极大地增加亍计算机系统的复杂性，局域网、广域网、宽带数字通信以及对“即时”数据访问需求的增加，都对软件开发者提出了更高的要求。但是，在这个时期软件仍然主要在工业界和学术界应用，个人应用还很少。这个时期的主要特点是出现了微处理器，而且微处理器获得了广泛应用。以微处理器为核心的“智能”产品随处可见，当然，最重要的智能产品是个人计算机。在不到10年的时间里，个人计算机已经成为大众化的商品。

**第四阶段**

　　在计算机系统发展的第四代已经不再看重单台计算机和程序，人们感受到的是硬件和软件的综合效果。由复杂操作系统控制的强大的桌面机及局域网和广域网，与先进的应用软件相配合，已经成为当前的主流。计算机体系结构已迅速地从集中的主机环境转变成分布的客户机／[服务器](http://www.kokojia.com/s1029/%22%20%5Co%20%22%E6%9C%8D%E5%8A%A1%E5%99%A8%22%20%5Ct%20%22_blank)(或浏览器／服务器)环境。世界范围的信息网为人们进行广泛交流和资源的充分共享提供了条件。软件产业在世界经济中已经占有举足轻重的地位。随着时代的前进，新的技术也不断地涌现出来。面向对象技术已经在许多领域迅速地取代了传统的软件开发方法。

**总结**

　　软件开发的“第四代技术”改变了软件界开发计算机程序的方式。专家系统和人工智能软件终于从实验室中走出来进入了实际应用，解决了大量实际问题。应用模糊逻辑的人工神经网络软件，展现了模式识别与拟人信息处理的美好前景。虚拟现实技术与[多媒体](http://www.kokojia.com/list/333.html)系统，使得与用户的通信可以采用和以前完全不同的方法。遗传算法使我们有可能开发出驻留在大型并行生物计算机上的软件。

　　**计算机体系结构-基本原理**

****

计算机体系结构

　　计算机体系结构解决的是计算机系统在总体上、功能上需要解决的问题，它和计算机组成、计算机实现是不同的概念。一种体系结构可能有多种组成，一种组成也可能有多种物理实现。

　　计算机系统结构的逻辑实现，包括机器内部数据流和控制流的组成以及逻辑设计等。其目标是合理地把各种部件、设备组成计算机，以实现特定的系统结构，同时满足所希望达到的性能价格比。一般而言，计算机组成研究的范围包括：确定数据通路的宽度、确定各种操作对功能部件的共享程度、确定专用的功能部件、确定功能部件的并行度、设计缓冲和排队策略、设计控制机构和确定采用何种可靠技术等。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、底板的划分与连接，专用器件的设计，信号传输技术，电源、冷却及装配等技术以及相关的制造工艺和技术。

　　　**计算机体系结构-分类**

****

指令与数据模型

**（1）Flynn分类法**

　　1966年，Michael.J.Flynn提出根据指令流、数据流的多倍性（multiplicity）特征对计算机系统进行分类，定义如下。

　　•指令流：机器执行的指令序列

　　•数据流：由指令流调用的数据序列，包括输入数据和中间结果

　　•多倍性：在系统性能瓶颈部件上同时处于同一执行阶段的指令或数据的最大可能个数。

　　Flynn根据不同的指令流-数据流组织方式把计算机系统分为4类。

　　1•单指令流单数据流（SingleInstructionStreamSingleDataStream，SISD）

　　SISD其实就是传统的顺序执行的单处理器计算机，其指令部件每次只对一条指令进行译码，并只对一个操作部件分配数据。

　　2•单指令流多数据流（SingleInstructionStreamMultipleDataStream，SIMD）

　　SIMD以并行处理机为代表，结构如图，并行处理机包括多个重复的处理单元PU1～PUn，由单一指令部件控制，按照同一指令流的要求为它们分配各自所需的不同的数据。

　　3•多指令流单数据流（MultipleInstructionStreamSingleDataStream，MISD）

　　MISD的结构，它具有n个处理单元，按n条不同指令的要求对同一数据流及其中间结果进行不同的处理。一个处理单元的输出又作为另一个处理单元的输入。

　　4•多指令流多数据流（MultipleInstructionStreamMultipleDataStream，MIMD）

　　MIMD的结构，它是指能实现作业、任务、指令等各级全面并行的多机系统，多处理机就属于MIMD。



　冯式分类法

　**（2）冯式分类法**

　　1972年冯泽云提出用最大并行度来对计算机体系结构进行分类。所谓最大并行度Pm是指计算机系统在单位时间内能够处理的最大的二进制位数。设每一个时钟周期△ti内能处理的二进制位数为Pi，则T个时钟周期内平均并行度为Pa=(∑Pi)／T(其中i为1，2，…，T)。平均并行度取决于系统的运行程度，与应用程序无关，所以，系统在周期T内的平均利用率为μ=Pa/Pm=(∑Pi)/(T\*Pm)。用最大并行度对计算机体系结构进行的分类。用平面直角坐标系中的一点表示一个计算机系统，横坐标表示字宽(N位)，即在一个字中同时处理的二进制位数；纵坐标表示位片宽度(M位)，即在一个位片中能同时处理的字数，则最大并行度Pm=N\*M。

**由此得出四种不同的计算机结构：**

　　①字串行、位串行(简称WSBS)。其中N＝1，M＝1。

　　②字并行、位串行(简称WPBS)。其中N＝1，M>1。

　　③字串行、位并行(简称WSBP)。其中N>1，M＝1。

　　④字并行、位并行(简称WPBP)。其中N>1，M>1。

　　**计算机体系结构-技术革新**

****

可信终端体系结构

　　计算机体系结构以图灵机理论为基础，属于冯•诺依曼体系结构。本质上，图灵机理论和冯•诺依曼体系结构是一维串行的，而多核处理器则属于分布式离散的并行结构，需要解决二者的不匹配问题。

　　首先，串行的图灵机模型和物理上分布实现的多核处理器的匹配问题。图灵机模型意味着串行的编程模型。串行程序很难利用物理上分布实现的多个处理器核获得性能加速.与此同时,并行编程模型并没有获得很好的推广，仅仅局限在科学计算等有限的领域.研究者应该寻求合适的机制来实现串行的图灵机模型和物理上分布实现的多核处理器的匹配问题或缩小二者之间的差距，解决“并行程序编程困难，串行程序加速小”的问题。

　　在支持多线程并行应用方面，未来多核处理器应该从如下两个方向加以考虑。第一是引入新的能够更好的能够表示并行性的编程模型。由于新的编程模型支持编程者明确表示程序的并行性，因此可以极大的提升性能。比如Cell处理器提供不同的编程模型用于支持不同的应用。其难点在于如何有效推广该编程模型以及如何解决兼容性的问题。第二类方向是提供更好的硬件支持以减少并行编程的复杂性。并行程序往往需要利用锁机制实现对临界资源的同步、互斥操作，编程者必须慎重确定加锁的位置，因为保守的加锁策略限制了程序的性能，而精确的加锁策略大大增加了编程的复杂度。一些研究在此方面做了有效的探索。比如，SpeculativeLockElision机制允许在没有冲突的情况下忽略程序执行的锁操作，因而在降低编程复杂度的同时兼顾了并行程序执行的性能。这样的机制使得编程者集中精力考虑程序的正确性问题，而无须过多地考虑程序的执行性能。更激进的，TransactionalCoherenceandConsistency(TCC)机制以多个访存操作（Transaction）为单位考虑数据一致性问题，进一步简化了并行编程的复杂度。

　　主流的商业多核处理器主要针对并行应用，如何利用多核加速串行程序仍然是一个值得关注的问题。其关键技术在于利用软件或硬件自动地从串新程序中派生出能够在多核处理器上并行执行的代码或线程。多核加速串行程序主要有三种方法，包括并行编译器、推测多线程以及基于线程的预取机制等。在传统并行编译中，编译器需要花费很大的精力来保证拟划分线程之间不存在数据依赖关系。编译时存在大量模糊依赖，尤其是在允许使用指针（如C程序）的情况下，编译器不得不采用保守策略来保证程序执行的正确性。这大大限制了串行程序可以挖掘的并发程度，也决定了并行编译器只能在狭窄范围使用。为解决这些问题，人们提出推测多线程以及基于线程的预取机制等。然而，从这种概念提出到现在为止，这个方向的研究大部分局限于学术界，仅有个别商业化处理器应用了这种技术，并且仅仅局限于特殊的应用领域。我们认为动态优化技术和推测多线程（包括基于线程的预取机制）的结合是未来的可能发展趋势。

　　冯•诺依曼体系结构的一维地址空间和多核处理器的多维访存层次的匹配问题。本质上，冯•诺依曼体系结构采用了一维地址空间。由于不均匀的数据访问延迟和同一数据在多个处理器核上的不同拷贝导致了数据一致性问题。该领域的研究分为两大类：一类研究主要是引入新的访存层次。新的访存层次可能采用一维分布式实现方式。典型的例子是增加分布式统一编址的寄存器网络。全局统一编址的特性避免了数据一致性地考虑。同时，相比于传统的大容量cache访问，寄存器又能提供更快的访问速度。TRIPS和RAW都有实现了类似得寄存器网络。令另外，新的访存层次也可以是私有的形式。比如每个处理器和都有自己私有的访存空间。其好处是更好的划分了数据存储空间，已洗局部私有数据没有必要考虑数据一致性问题。比如Cell处理器为每个SPE核设置了私有的数据缓冲区。另一类研究主要涉及研制新的cache一致性协议。其重要趋势是放松正确性和性能的关系。比如推测Cache协议在数据一致性未得到确认之前就推测执行相关指令，从而减少了长迟访存操作对流水线的影响。此外，TokenCoherence和TCC也采用了类似的思想。

　　计算机系统发展的第三代从20世纪70年代中期开始，并且跨越了整整10年。在这10年中计算机技术又有了很大进步。分布式系统极大地增加亍计算机系统的复杂性，局域网、广域网、宽带数字通信以及对"即时"数据访问需求的增加，都对软件开发者提出了更高的要求。但是，在这个时期软件仍然主要在工业界和学术界应用，个人应用还很少。这个时期的主要特点是出现了微处理器，而且微处理器获得了广泛应用。以微处理器为核心的"智能"产品随处可见，当然，最重要的智能产品是个人计算机。在不到10年的时间里，个人计算机已经成为大众化的商品。



　　网络体系结构中涉及到了：协议、实体、接口

　　计算机网络中实现通信就必须依靠网络通过协议。在20世纪70年代，各大计算机生产商的产品都拥有自己的网络通信协议。但是不同的厂家生产的计算机系统就难以连接，为了实现不同厂商生产的计算机系统之间以及不同网络之间的数据通信，国际标准话组织ISO（开放系统互连参考模型）即OSI/RM也称为ISO/OSI，该系统称为开放系统。

  **外部体系结构**

　　程序的多样性和单一的体系结构的匹配问题。未来的应用展现出多样性的特点。一方面，处理器的评估不仅仅局限于性能，也包括可靠性，安全性等其他指标。另一方面，即便考虑仅仅追求性能的提高，不同的应用程序也蕴含了不同层次的并行性。应用的多样性驱使未来的处理器具有可配置、灵活的体系结构。TRIPS在这方面作了富有成效的探索，比如其处理器核和片上存储系统均有可配置的能力，从而使得TRIPS能够同时挖掘指令级并行性、数据级并行性及指令级并行性。

　　多核和Cell等新型处理结构的出现不仅是处理器架构历史上具有里程碑式的事件，对传统以来的计算模式和计算机体系架构也是一种颠覆

　　2005年，一系列具有深远影响的计算机体系结构被曝光，有可能为未来十年的计算机体系结构奠定根本性的基础，至少为处理器乃至整个计算机体系结构做出了象征性指引。随着计算密度的提高，处理器和计算机性能的衡量标准和方式在发生变化，从应用的角度讲，讲究移动和偏向性能两者已经找到了最令人满意的结合点，并且有可能引爆手持设备的急剧膨胀。尽管现在手持设备也相对普及，在计算能力、可扩展性以及能耗上，完全起步到一台手持设备应该具备的作用；另一方面，讲究性能的服务器端和桌面端，开始考虑减少电力消耗赶上节约型社会的大潮流。

　　Cell本身适应这种变化，同样也是它自己创造了这种变化。因而从它开始就强调了不一样的设计风格，除了能够很好地进行多倍扩展外，处理器内部的SPU(SynergisticProcessorUnit协同处理单元)具有很好的扩展性，因而可以同时面对通用和专用的处理，实现处理资源的灵活重构。也就意味着，通过适当的软件控制，Cell能应付多种类型的处理任务，同时还能够精简设计的复杂。