计算机体系结构[1]

一、 计算机体系结构的基本概念

计算机体系结构是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和功能特性。Amdahl 所定义的体现结构是指程序员面对的是硬件的系统。所关心的是如何合理的进行软硬件功能的分配。

计算机系统结构是指机器语言级的程序员所了解的计算机的属性，即外特性。可以包含数据表示，寄存器定义、数量、使用方式，指令系统，中断系统，存存储系统，IO 系统等。

计算机组成是计算机结构的逻辑实现。可以包含数据通路宽度，专用部件设置，缓冲技术，优化处理等。

计算机的实现是指其计算机组成的物理实现。包括处理机，主存部件的物理结构，器件的集成度，速度的选择，模块、硬件、插件底板的划分和连接。

从使用语言的角度，可以把计算机系统按功能从高到低分为 7 级：0 应用语言机器级、1 高级程序语言机器级、2 汇编语言机器级、3 操作系统机器级、4 传统机器语言机器级、5 微程序机器级和 6 电子线路级。3～6 级为虚拟机，其语言功能均由软件实现。

硬件功能分配的基本原则：（1）功能要求。首先是应用领域对应的功能要求，其次是对软件兼容性的要求；（2）性能要求。如运算速度，存储容量，可靠性，可维护性和人机交互能力等；（3）成本要求。

体系结构设计的方法有三种：由上而下—从考虑如何满足应用要求开始设计；由下而上—基于硬件技术所具有的条件；由中间开始的

方法。

体系设计的步骤：需求分析、需求说明、概念性设计、具体设计、优化和评价。

计算机体系结构的分类：（1）弗林 FLYNN 分类法：按指令流和数据流将计算机分为 4 类：①单指令流、单数据流－Single Instruction Stream Single Data Stream ，SISD。计算机，即传统的单处理机，通常用的计算机多为此类，如脉动阵列计算机 systolic array ；②单指令流、多数据流－Multiple ，SIMD。典型代表是并行处理机。其并行性在于指令一级。如 ILLIAC、PEPE、STARAN、MPP等；③MISD计算机；④MIMD计算机。多处理机系统，实现全面并行的理想结构。可以通过共享存储器和消息传递来耦合系统，每个处理器分别执行系统分配的程序，同时执行多个指令流对多个数据流不同的处理，如 IBM3081/3084，Cray-2 等。//弗林分类法基本上是对除流水线处理机外的诺衣曼型控制流计算机进行分类，而不包括对像数据流计算机这种非诺衣曼型机器进行分类；（2）冯氏分类法。依据是并行度－即计算机在单位时间内能够处理的最大二进制位数。据此分为 4 类：①字串位串 Word Serial and Bit Serial 。WSBS计算机。只有一个串行的处理部件，每字长 1 位；②字并位串 Parallel 。WPBS计算机。只有一个处理部件。该部件处理字长 n 位；③字串位并 WSBP 计算机。有多个处理部件。每个处理部件字长 1 位；④字并位并 WPBP 计算机。有多个处理部件，各部件字长也并行，如 ILLICA2计算机具有 64 个字长 64 位的处理单元。

冯.诺衣曼型计算机体系结构及其发展（1）是存储程序计算机的别称。在体系结构有着如下特点：①机器以运算器位中心，使用单一处理部件来完成计算、存储及通信工作；②采用存储程序的原理，使用线性组织的定长存储单元来存储程序，存储时对指令和数据不加区别；③存储空间的单元是直接寻址的，每个单元位数固定；④使用二进制机器语言，其指令完成基本操作码的简单操作；⑤对计算机进行集中的顺序控制。（2）两个最主要的特征：一是计算机内部信息流动是由指令驱动的，而指令执行顺序由指令计数器决定；二是计算机的应用仍主要面向数值计算和数据处理。（3）发展：①数据流计算机DFM 只要所需的操作数齐备就可以执行，这时只取决于执行部件的并行处理能力；②智能计算机。主要处理一些非数值化信息。

体系结构并行技术的发展（1）并行性是指在同一时刻或同一时间间隔内完成两种或两种以上性质相同或不同的工作的特性。具有同时性和并发性二重性。（2）等级划分：（由低到高）：①按执行程序的等级划分：指令内部、指令之间、任务或进程之间、作业或程序之间；②处理数据等级划分：字串位串、字串位并、字并位串、字并位并；③按信息加工的等级划分：存储器操作并行、处理器操作并行、指令任务作业并行；（2）并行性的技术途径：①时间重叠。多个处理过程在时间上错开，如流水线处理机；②资源重复。重复设置硬件资源来提高计算机的性能。如阵列处理机；③资源共享。用软件方法让多个用户按一定时间顺序轮流使用同一套件资源，以提高计算机设备利用率。如多道程序分时系统。

题目：

1. 高级语言经编译程序的翻译形成汇编语言程序；
2. 传统机器语言机器级，是用微指令程序来解释机器指令；微指令由硬件直接执行；
3. Amdahl加速比定律：加速比：Sp＝1/（1-Fe+Fe/Se ），其中 Fe 为被改进部分的执行时间所占的百分比的大小；Se 是其性能提高的倍数。//局部性原理：程序趋向于重用它当前已经在使用的指令和数据。包括时间局部性和空间局部性。时间局部性是指当前访问的项目在最近的将来还会被访问；空间局部性是指某个项目及其附近地址的其他项目会同时被引用。
4. 实现软件移植的基本技术有：统一的高级语言、采用阵列机、模拟和仿真；
5. 仿真是指用微程序直接解释另一台计算机的机器指令系统；模拟指用机器语言解释实现软件移植的方法；
6. 多机系统的耦合度可分为：最低耦合、松散耦合和紧密耦合三种类型；

二、 指令系统

指令系统又称指令集 Instruction Set，它对计算机系统有全剧性影响，即指令的功能将直接反映系统功能。指令集发展有两个趋势：CISC 和 RISC;

指令集体系结构的分类（1）分类依据，可以有 5 种：操作数在 CPU 中的存储方式；显示操作数的数量；操作数的位置；指令的操作；操作数的类型和大小。（2）按暂存机制分类：依据在 CPU 内部存储操作数的区别，可以把指令集体系结构分为 3 类：堆栈 stack、累加器 accumulator、寄存器即 a set of registers。①堆栈机。主要操作是压入和弹出，其他操作还有加、减、比较等；优点是：表示数值的模型简单、指令长度短。②累加器类机器是有一个隐含操作数的机器。例如 PDP-8、Motorola6809；优点是机器的内部状态很少，指令也比较短。③寄存器为基础的指令系统优点是：速度更快、数值表示上有很强的适应性。例如 IBM360、DEC VAX。（3）通用寄存器 general-purpose resister machine，简称 GPR机。其关键性优点起因于编译程序能有效的使用寄存器，无论是计算表达式的值，还是从更为全局的角度使用寄存器来保存变量的值。可以分为 3 类：①寄存器-寄存器 resister-resister 。只能对存储器有存取指令，所有操作在两个寄存器之间进行，操作结果送入第三个寄存器中；优点是：速度快、指令具有良好的正交编码模型；如 RISC和 Cray 计算机；②寄存器-存储器类 register-memory 。在指令中，由寄存器内容加上存储器内容寻址构成寻址技术。如 VAX、IBM360、Motorola68000、PDP11等。优点是：数据不需要寄存器装入就能存取、指令大小适中；③存储器-存储器 memory-memory，如 VAX2和 IBM370，优点是紧凑、不需要消耗临时寄存器。

指令格式（1）指令编码方法，通常有 3 种：①正交法 orthogonal

method。对流水线计算机特别适应，采用微程序控制时微程序数量可以较少；②整体法 integrated 。可以把使用频率高的操作数通操作数地址码组合起来，加以缩短优化，而使用频率低的操作码可以较长些,从而节省存储容量,但需要较大的微程序存储器;③混合法 mixed。把以上两个方法优点结合。(2)指令格式。最普通的是:操作码 opcode-操作数 operand/ 地址。操作码字段表明操作类型；操作数/地址字段指明具体的操作数，也可以指明操作数地址，通常是和寻址方式相配合形成的。(3）寻址技术。即指令按什么方式寻找所需的操作数或信息，它影响主存规模速度和存取方式。寻址方式对于应用程序是透明的。①编址方式：统一编址—把各个不部件统一编成从 0 开始的一维线性地址空间；局部编址—指导这些部件适当分类，各自从 0 开始单独编址，形成多个一维的线性地址空间；隐含编址—地址隐含于操作码中；②程序定位方式。程序定位是把指令和数据中的逻辑地址转变成主存物理地址的过程,有三种方法:直接定位方式、静态定位方式、动态定位方式；③寻址方式。大多计算机都将主存、通用寄存器、堆栈分类编址，因此就有分别面向寄存器、堆栈和主存的寻址方式。

指令的优化（1）指令格式的优化。就是从整个指令系统的利用率角度出发，尽量设法减少指令中冗余信息量，以便用最少的位数提供足够的操作信息和地址信息。包括操作码的优化和操作数的优化；（2）哈夫曼编码。左 1 右 0。（3）理论码长—信息源熵。任意随机事件的出现概率为 Pi，则它的信息量 Ii ＝-log2 （Pi），则平均信息量为 H=-和（Pi*log2 （Pi）），由此式的结果 H即为理论码长。信息

冗余量＝1-理论码长/操作码的平均长度。（4）等长扩展码。哈夫曼编码方法形成的指令码很不规则，长度不一。事实上计算机采用等长扩展码，介于等长二进制编码和全哈夫曼编码之间的一种编码方式，仍利用哈夫曼思想，对概论高的指令用短码，概率低的用长码，但在整体上只采用了有限的几种码长。如下表：

指令	频度	哈夫曼码	等长码	等长扩展码
I1	0.40	0	000	00
I2	0.30	10	001	01
I3	0.15	110	010	10
I4	0.05	11100	011	1100
I5	0.04	11101	100	1101
I6	0.03	11110	101	1110
I7	0.03	11111	110	1111
平均码长		2.20	3	2.30

理论码长为 2.17。

指令系统的复杂化（1）CISC 和 RISC 的目标是相同的，都是为了提高性能，减少语义差距，改善性能价格比。目前多用 CISC 类型，如 IBM360/370 和 4300 系列等；（2）指令系统复杂化的实现措施：①面向目标代码的优化。按静态使用频度（程序中出现的百分比）改进可以减少存储空间；按动态使用频率（执行过程中出现的百分比）改进可以减少目标程序运行的执行时间；②面向高级语言的优化，就是尽可能缩小高级语言和机器语言之间的语义差异，以利于支持高级

语言的编译系统，左端编译程序的长度和编译所需时间；③面向操作系统的优化。就是进一步缩小操作系统和体系结构之间的语义差异，减少辅助时间，节省操作系统软件占用空间；（3）CISC的主要弊端：指令集过于庞杂；微程序技术是其重要支柱，这降低了处理速度；难以优化编译使之生成真正的目标代码；强调完善的中断机制，导致动作的烦杂；给芯片设计带来很多困难，出错几率增大，不利于大批量生产。

RISC 技术—精简指令集计算机 Reduced Instruction Set Computer。（1）基本特征：精简指令数量——般在 100 条左右；简化指令格式—在 1～2 种之内并让全部指令具有相同长度；采用单周期指令—几乎所有指令在一个机器周期内完成；采用寄存器操作—尽量减少访存操作；硬件控制逻辑—大多指令采用硬件控制实现，少数用微程序实现；优化编译程序。（2）RISC体系结构：①数据类型。有 2 种表示方法，一是用操作码编码表示，一是通过操作数内部标志位表示，即自定义数据表示；字长 64 位，包括整型数据和浮点数据，支持附加数据类型；②寻址方式，有：立即寻址方式、寄存器直接寻址方式、寄存器间接寻址、相对寻址方式、变址值方式和位移量方式。③寄存器模型和寄存器管理。有三种模型：窗口模型 Windows Cache 模型、矩阵 Matrix 模型；④存储器管理。（3）RISC的主要技术：①重叠寄存器窗口技术。有利于合理利用有限的芯片面积，特别是支持最费时的过程调用和返回操作；伯克利设计的重叠窗口有 8 个，每个共有 32 个 32 位的寄存器，其中 10 个全局性寄存器，10 个局部性寄

存器，6 个高位寄存器，6 个地位寄存器，其典型调用时间是 2 微秒；②优化编译技术。可以合理分配寄存器，提高寄存器的使用效率，减少访存次数等。③超流水线及超标量技术。超流水线 superpipeline 技术是一种并行处理技术，通过细化流水，增加级数和提高主频，使得在每个机器周期内能完成一个甚至两个浮点操作，实质是以时间换取空间；超标量 superscalar 技术也是并行处理技术，通过内装多条流水线来同时执行多个处理，实质是以空间换取时间；④硬线逻辑与微程序相结合。

MIPS和 MFLOPS（1）MIPS每秒执行的百万次指令数。（2）MFLOPS每秒浮点运算的百万次数。

三、 存储系统

存储器的层次结构（1）存储器以存取速度为主要标准依次排列：最快的是与 CPU 同在一块芯片上的寄存器 resister，其次是高速缓冲存储器 cache－memory，然后是主存储器 main－memory，接着是辅助存储器 auxiliary－storage，最低层是海量存储器 mass－storage。（2）有两个重要的存储层次，一是主－辅层次，以扩大存储容量位宗旨，多用软件管理来实现。另一个 Cache－主存层次，以提高存取速度为宗旨，均用硬件方法实现；（3）实现存储体系的依据正是局部性原理，包括时间局部性和空间局部性。存储体系的性能参数有：①平均字节价格 C＝（C1S1+C2S2）／（S1+S2）；为了使得存储系统的字节价格接近辅存的字节价格，要求主存容量远小于辅存容量；②命中率 H＝N1/

（N1+N2；③存取时间 T＝HT1+（1-H）T2；④存储器利用率 u＝Sa/S，Sa 是程序“活跃”部分所占用的存储空间，S 是可利用的存储空间总容量。

并行存储器（1）并行存储技术也是存储器中的体系结构问题，它既能扩大存储容量，又能提高访问速度。把存储器分成多个模块，在一次访问的时间内，就能并行的读出更多信息量，具有这样组织形式的存储器称为并行存储器 Parallel memory 。又称为存储器的多体交叉访问 multiple module interleaved memory ；（2）访问控制方法：①同时访问。可以一次提供多个数据或多条指令，适合对多数据流或多指令流进行并行处理。应注意频带宽度的问题，保证处理单元接收/处理数据的速率要和并行存储器同时读写数据的速率相匹配；采用交叉开关总线；②轮流访问。在对并行多体存储器访问时，各模块按一定的顺序轮流启动各自的访问周期。降低了对带宽匹配的要求。采用分时共享总线。

虚拟存储器（1）虚拟存储器的管理方式。决定于主存与虚存间不同的地址影响方式，分别是段式管理、页式管理和段页式管理 3 种方式。①段式管理。地址映象—将虚存空间分段，主存的空间按这种段来分配和管理。段是按程序的逻辑功能来划分的。当程序从辅存调入主存时，是按段分配主存空间，需要建立一个包括段长度和主存起始地址的段表，存放在主存中；地址转换—在段式管理中，主存地址格式包括段号和段内地址，虚存地址格式包括用户号、段号和段内地址。②页式管理。将主存空间和虚存空间按固定大小划分成块，每块

称为一页。页的大小和划分与程序逻辑功能无关。③段页式管理。将虚拟存储空间按段式管理，主存空间则按页式管理。存在虚拟空间的程序按逻辑关系分段，每一段又可分成固定大小的页。主存则只分成若干大小相同的页。许多大型机都采取该管理方式；（2）页面替换算法。在虚拟存储器中，由于虚拟空间比主存空间大得多，会出现当主存中所有页已经全部被占用，而 CPU需要的指令却在主存中找不到，从而产生页面失效 Page fault 。这是需要从辅存中调入新页，并把主存中已经不用的旧页替换出去。常用的替换算法有：①随机算法 RAND 算法简单，易于实现；②先进先出 FIFO；③近期最少使用算法 LRU；④优化替换算法 OPT—预先知道将要使用哪些页面，替换时把下次调用该页时的时间间隔最大的页面调出去。这是一种理想算法。

高速缓冲存储器 Cache（1）为弥补主存速度不足，在处理机和主存之间设置一个高速小容量的 Cache，构成“Cache- 主存”层次，其在本质上是一个两级的“页”式系统；（2）“Cache- 主存”和“主存–辅存”的比较：前者目的是提高存储系统速度，后者是扩大容量；两者工作原理相同，都需要地址变换，但失效时，后者采取页面替换，前者采用块 block 替换；前者通过硬件实现地址变换和块替换，后者则是由操作系统来管理的；主辅层次的两种存储介质有很大区别，不易匹配，而前者则便于匹配；Cache 对应用程序员和系统程序员都是透明的，而主辅层次则对系统程序员不完全透明。（3）在有 Cache 的系统中，访问主存请求的优先级安排次序是 Cache- 通道–写数–读数–

取指令。(4) 地址映象和变换。地址映象是指每个主存按什么规则装入 Cache 中。有全相联映象、直接映象、组相联映象。(5) Cache 的块替换算法。有 RAND FIFO、LRU等；LRU替换算法的硬件实现有：①堆栈法。从栈底到栈顶的几何位置反映了各块近期最久未被访问的次序。②比较对法。让各块成对组合，用触发器状态表示每个比较对内的访问次序，从而找出被替换的块。综上所述，设计替换算法实现应考虑到：如何对每次访问进行记录和符合根据所记录信息来判定哪个块是近期内最久未被访问的。(6) Cache 的块表示。在级相联或直接映象 Cache 中，地址的数据结构由 3 个部分组成：标志 tag －给出块帧地址；索引 index －组相联中通过它选择组号；块内位移 block offset －给出在一个块内所找数据的地址。(7) Cache 的写策略，即更新主存内容的算法。①写直达法 write through ：只要 CPU有写操作，在写入 Cache 同时，也通过“Cache－主存”通路直接写入主存；②写回法 write back。在 CPU执行写操作时，信息只写入 Cache，仅当某块被替换时，才把曾被写入过的 Cache 块先送回主存，然后再调入新块。//写直达法的可靠性高，但增加了访问主存的流量，写回法则相反，减少了不必要的访存，但可靠性受影响，常需要在 Cache 中增加更多的冗余信息位来提高其内容可靠性。

题目：

1. 衡量一个存储层次体系性能主要从平均字节价格、命中率、存取时间等三个方面考虑；