

1 计算机系统的多级层次结构，， 1: M0 微程序（微指令）机器、2: M1 传统（机器指令）机器、3: M2 操作系统（作业控制）机器、4: M3 汇编语言机器、5: M4 高级语言机器、6: M5 应用语言机器；1-2 为实际机器，3-6 为虚拟机器

2, 各个机器级的实现的依靠，， 翻译和解释

3, 翻译, translation, 是先用转换程序将高级机器级上实现的程序整个地变换成低级机器级上等效的程序, 然后再在低级机器上实现的技术。

4, 解释, interpretation, 在低级机器级上用它的一串语句或指令来仿真高级机器级上的一条语句或指令的功能, 通过高级机器语言程序中的每条语句或指令逐条解释来实现的技术。

5, M0-M5 各级的实现方式，， M0 用硬件实现, M1 用微程序（固件实现）， M2 到 M5 大多采用软件实现。

6, 虚拟机器的定义，， 以软件为主实现的机器

7, 实际机器的定义，， 以硬件或固件实现的实际机器

8 透明的定义，， 客观存在的事物或属性从某个角度看不到, 称之为透明

9 计算机系统结构的定义，， 是系统结构的一部分, 指的是传统机器级的系统结构, 其界面之上包括操作系统、汇编语言、高级语言和应用语言级中所有的软件功能, 界面之下包括所有硬件和固件的功能。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, 或者编译程序设计者看到的机器物理系统的抽象。是研究软、硬件之间的功能分配以及对传统机器级界面的确定, 提供机器语言、汇编语言设计者或编译程序生成系统为使其设计或生成的程序能在机器上正确运行应看到或遵循的计算机属性。

10 计算机系统结构的属性

（需了解一部分），， 1、数据表示；2、寻址方式；3、寄存器组织；4、指令系统；5、存储系统组织；6、中断机构；7、系统机器级的管态和用户态的定义与切换；8、I/O 结构；9、信息保护方式和保护机构等等。

11 计算机系统结构不包含的内容，， 不包含“机器级内部”的数据流和控制流的组成, 逻辑设计和器件设计等。

12 计算机组成的定义，， 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。主要围绕提高速度、提高操作的并行度、重叠度、功能的分散、设置专用功能部件等方面来设计。

13 计算机组成设计包含的内容（需了解一部分），， 1、数据通路宽度；2、专用部件的设置；3、各种操作系统对部件的共享程度；4、功能部件的并行度；5、控制机构的组成方式；6、缓冲和排队技术；7、预估、预判技术；8、可靠性技术等等

14 计算机实现的定义，， 指的是计算机组成的物理实现。着眼于器件技术和微组装技术。

15 计算机系统结构、组成和实现的区别举例，， 是否设置乘法指令属于系统结构, 是用高速乘法器还是加法器和移位器实现数据组成, 乘法器加法器的物理实现属于实现。

16 结构、组成、实现三者的相互影响，， 三者互不相同, 但相互影响。结构的变化可能会引起组成的变化, 组成的变化也可能会引起结构的变化等等。需综合考虑价格、速度、性能、实现等因素。

17 计算机系统结构设计的作用，， 主要是进行软、硬件功能分配。

18 计算机系统结构设计软硬件取舍的原则，， 硬件高-》速度快、成本高、降低灵活性和适应性。软件高-》速度慢、成本低、灵活性高。

原则 1、在现有的硬件（主要是逻辑器件和存储器件）条件下, 系统要有高的性能价格比。经常用的功能用硬件, 产量大的计算机系统, 增大硬件功能实现。

原则 2、准备采用和可能采用的组成技术要尽可能不要过多或不合理的限制各种组成、实现技术的采用。

原则 3、不能只从硬件角度考虑如何便于应用组成的实现, 还要从软件的角度把如何编译和操作系统的实现以及为高级语言程序设计提供更好更多的硬件支持放在首位。

19 计算机系统的设计思路（多层结构），， 1、由上而下：满足应用开始, 适合专用机设计, 不适合通用机设计,

2、由下而上：不管应用要求, 从到手硬件开始设计；软、硬件脱节是以上两种设计的主要缺点。

3、中间开始：从层次结构中软硬件交界面设计, 目前主要是传统机器和操作系统机器级之间。

20 软件的可移植性的定义，， 指的是软件不修改或经过少量修改就可由一台机器搬到另外一台机器上运行, 同一软件可应用于不同的环境。



- 21 软件移植的基本技术, , 1、统一高级语言, 2、采用系列机(系列机上可实现软件兼容, 中档机性能价格比较高), 3、模拟与写真。
- 22 模拟的定义, , 用机器语言(第二层级)程序解释实现软件移植的方法称为模拟(需要通过机器语言和微程序两重解释)。
- 23 模拟的宿主机和模拟机的定义, , 进行模拟的机器称为宿主机, 被模拟的机器称为虚拟机。
- 24 宿主机模拟/仿真目标机的范围, , 1、机器语言; 2、存储体系; 3、I/O 系统; 4、控制台的操作; 5、形成虚拟机的操作系统
- 25 模拟适用的情况, , 适合于移植运行时间短, 使用次数少, 时间没有约束限制的软件。
- 26 仿真的定义, , 用微程序(第一层级)直接解释另一种机器指令系统的方法叫仿真(只需要通过微程序一重解释)。
- 27 仿真的宿主机和仿真机的定义, , 进行仿真的机器称为宿主机, 被仿真的机器称为目标机。
- 28 模拟和仿真的区别, , 区别在于解释的语言, 仿真用微程序解释, 解释程序存在于控制存储器中; 模拟用机器语言解释, 解释程序存在主存中。仿真更快, 不灵活; 模拟更适合两种机器结构差异大的情况, 灵活但速度不快。
- 29 计算机的性能的衡量标准, , 1、硬件: 主频、CPU 速度、字长、数据类型、主存容量、寻址范围、存储体系、I/O 处理能力、I/O 设备、指令系统等; 2、软件: 高级语言状况、操作系统功能、用户程序等; 3、可靠性; 4、可用性等多种指标的综合。
- 30 计算机应用的分类, , 1、数据处理; 2、信息处理; 3、知识处理; 4、智能处理。
- 31 计算机器件的发展, , 电子管-晶体管-小规模集成电路-大规模集成电路-超大规模集成电路
- 32 非用户片的定义, 功能片, 功能由器件厂生产时定死了的器件。速度较慢, 价格便宜。
- 33 现场片, , 用户可根据需要改变器件内部功能。速度一般, 价格一般。
- 34 用户片, , 按用户要求生产的高集成度 VLSI 器件。速度较快, 价格贵。
- 35 提高计算机系统性能的有效途径, , 开发并行性, 挖掘潜在的并行性, 提高并行处理和操作的程度。
- 36 并行性的定义, , 解题中具有同时进行运算或操作的特性, 只要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, 在时间上重叠, 都体现了并行性。
- 37 执行角度看的并行性由低到高的 4 个等级, , 1、指令内部: 指令内各个微操作之间的并行; 2、指令之间; 3、任务或进程之间; 4、作业或程序之间。
- 38 数据处理的并行性由低到高的 4 个等级, , 1、位串字串: 同时处理一个字的一位, 没有并行性; 2、位并字串: 同时处理一个字的全部位; 3、位片串字并: 同时对许多字的同一位(俗称片)进行处理; 4、全并行: 同时对许多字的全部或部分位进行处理。
- 39 信息加工的并行性由低到高的 4 个等级, , 1、存储器操作并行: 相联处理机; 2、处理器操作步骤并行: 流水线处理机; 3、处理器操作并行: 阵列处理机; 4、指令、任务、作业并行: 多处理机。
- 40 并行性开发的途径, , 1、时间重叠(重叠流水); 2、资源重复(多个机器处理同一任务); 3、资源共享(多个用户轮流使用同一套资源)。
- 41 3T 目标的定义, , 1TFlops 计算能力, 1Tbyte 主存容量, 1Tbytes/s 的 I/O 带宽。
- 42 并行处理计算机按结构分类, , 1、流水线计算机(主要通过时间重叠, 多个部件在时间上交错重复预算和处理); 2、阵列处理机(通过资源重复, 即同类型机器实现空间商的并行); 3、多处理机(资源共享; 共享主存紧耦合、不共享主存松耦合); 4、数据流计算机(数据驱动)。
- 43 耦合的分类, , 1、最低耦合: 计算机之间无物理连接, 通过磁盘、磁带等对主机输入输出; 2、松散耦合: 多台计算机通过 a 通道或 b 通信线路实现互连; 3、紧密耦合: 共享主存。
- 44 计算机系统的分类(指令流数据流分类法)(弗林分类法), , 1、单指令流单数据流: SISD(流水方式的单处理机); 2、SIMD(流水方式); 3、MISD(阵列); 4、MIMD(多处理机)。
- 45 弗林分类法的计算机分类举例, 25 页图,
- 46 计算机系统的分类(指令流和执行流分类)(库克分类法), , 1、单指令流单执行流: SISE(单处理机); 2、SIME(带多操作部件的处理机); 3、MISE(带指令级多道程序的处理机); 4、MIME(多处理机)。



- 47 计算机系统的分类（数据处理的并行度）（冯泽云分类法），， 1、字串位串：WSBS，每次只处理一个字的一位（位串行机）；2、字串位并：WSBP，每次处理一个字的  $n$  位（位并行机）；3、字并位串：WPBS，每次处理  $m$  个字的 1 位（阵列处理机）；4、字并位并：WPBP，每次处理  $m$  个字的  $n$  位（多处理机）。
- 48 数据表示的定义，， 数据表示是指能由机器硬件直接识别和引用的数据类型；数据表示是数据结构的组成元素。
- 49 软件要处理的数据结构的种类，， 串、队、栈、向量、队列、阵列、链表、树、图等。
- 50 数据表示的确定实质，， 软、硬件的取舍。
- 51 标识符数据表示的定义，， 机器中每个数据都带类型标志位，将数据类型与数据本身直接联系在一起的数据表示方式。
- 52 标识符数据表示的优点，， 1、简化了指令系统和程序设计；2、简化了编译程序；3、便于实现一致性校验；4、能由硬件自动变换数据类型；5、为软件调试和应用软件开发提供了支持。
- 53 标识符数据表示的缺点，， 1、增加所占主存空间；2、降低指令执行速度。
- 54 数据描述符的定义，， 对于属性相同的元素，采用分开存放的描述符来表示访问的数据的地址及其他信息的符号。
- 55 数据描述符表示数据或描述符的方式，， 前三位为 000 表示该字是数据，前三位为 101 表示该字为描述符。
- 56 标识符和数据描述符的区别，， 标识符与每个数据相连，合存于同一存储单元，描述单个数据的类型特征；描述符与数据分开存放，用于描述所要访问的数据是单个还是整块的，访问该数据块或数据元素所要的地址及其他信息等。
- 57 向量数组数据表示的特点，， 1、快速形成元素地址；2、便于实现各元素成块预取；3、节省存储空间，减少处理时间。
- 58 堆栈机的特点，， 1、由高速寄存器组成的硬件堆栈，访问速度是寄存器的，容量是主存的；2、可直接对堆栈中的数据进行各种运算和处理；3、有力的支持高级语言程序的编译；4、有力支持子程序的嵌套和递归调用；5、使用较少地址码，或者相对寻址，存储效率高。
- 59 数据表示的原则，， 1、看系统效率是否提高，即是否减少了实现时间和存储时间；2、看其通用性和利用率是否高。
- 60 浮点数尾数基数的取值特性，， 尾数基数越大，浮点数范围越大，增加可表示数的个数，减少移位次数，提高运算速度。但会降低数据的表示精度，使数值分布变稀。
- 61 浮点数尾数的下溢处理方法分类，， 1、截断法：实现简单，不增加硬件，误差大；2、舍入法：实现简单，不增加硬件，最大误差小，处理速度慢。3、恒置 1 法：实现简单，不增加硬件，最大误差大；4、查表舍入法：误差最小，速度最快，但需增加硬件。
- 62 寻址方式的定义，， 指令按什么方式寻找（或访问）所需要的操作数或信息。
- 63 计算机寻址方式（对象）的分类，， 1、面向主存；2、面向寄存器（速度快）；3、面向堆栈（减轻高级语言负担，地址节省好，支持嵌套、递归等）。
- 64，指令逻辑地址形成真地址的方式分类，， 1、立即；2、直接；3、间接；4、相对；5、变址。
- 65，静态再定位定义，， 在目的程序装入主存时，由程序装入软件方法把目的程序的逻辑地址变换为物理地址，程序执行时物理地址不再改变。
- 66 动态再定位定义（基址寻址），， 增加相应标志来指明地址码是否需要增加基址，在执行每条指令时才形成访存物理地址。
- 67 信息在存储器中按整数边界存储的定义，， 为了确保任何时候需要的信息都只有一个存储周期访问到，信息在主存中存放的地址必须是该信息宽度的整数倍。
- 68 指令系统设计的原则，， 1、优化机器的性能价格比；2、有利于指令系统的发展和改进；3、满足系统的基本功能。
- 69 指令的构成，， 操作码和地址码
- 70 指令的优化的定义，， 如何用最短的位数来表示指令的操作信息和地址信息。



71 哈夫曼压缩概念的基本思想，， 当各种事件发生的概率不均等时，采用优化技术对发生概率最高的事件用最短的位数（时间）来表示（处理），而对概率较低的，允许用较长的位数（时间）来表示（处理），使平均位数（时间）缩短。

72 哈夫曼编码的特点，， 不是唯一的，树形状不同，但只要采用全哈夫曼编码，操作码的平均时长是唯一的。

73 指令系统设计的方向，， 1、复杂指令系统计算机（CISC）：增强原有指令功能以及设置更为复杂的新指令取代原有软件子程序的功能，实现软件功能化。

2、精简指令系统计算机（RISC）：减少指令种数和简化指令功能用来降低硬件设计的复杂度，提高指令的执行速度。

74 面向目标程序优化的思路原则，， 1、优化指令及指令串使用频度；2、增设复合指令取代原有宏指令或子程序，提高速度，降低存储空间。

75 面向目标程序优化的思路 1：静态使用频度，， 对程序统计出指令及其指令串使用频度称为静态使用频度，按静态使用频度改进指令系统是减少目标程序的占用空间。

76 面向目标程序优化的思路 2：动态使用频度，， 对程序执行过程中统计出指令及其指令串使用频度称为动态使用频度，按动态使用频度改进指令系统是减少目标程序的执行时间。

77 面向高级语言优化的思路原则，， 尽可能的缩短高级语言和机器语言的语义差距，支持高级语言编译，缩短编译长度和时间。

78 面向高级语言优化的思路，， 1、优化高级语言使用频度；2、面向编译，优化代码生成；3、改进指令系统，使之与各种语言间的语义差异缩短。4、使机器分别面向各种高级语言的多种指令系统，多种系统结构；5、发展高级语言机器。

79 死锁的定义，， 一组进程中每个进程都只占有为完成该进程所必须的部分资源，并未获得全部资源，从而无法进行下去。

80 RISC 设计原则，， 1、只选择使用频度很高的指令；2、减少指令系统寻址方式，一般不超过两种；3、使所有指令都在一个周期内完成；4、扩大通用寄存器数量；5、大多数指令采用硬联控制实现；6、精简指令和优化编译程序，更好的支持高级语言。

81 CISC 和 RISC 的比较，， RISC 优点：1、指令简单，2、机器执行速度快，3、成本低且可靠，4、可直接支持高级语言实现；缺点：1、要完成复杂功能指令多，2：对浮点运算和虚拟存储器支持不足，3：编译程序比 CISC 难写。

82 输入输出系统的组成，， 1、输入输出设备；2、设备控制器；3、与输入输出操作有关的软硬件。

83 输入输出系统经历的阶段，， 1、程序控制 I/O；2、直接存储器访问（DMA）；3、I/O 处理机方式（分为通道及外围处理机两种方式）。

84 按信息传送方向的总线分类，， 1、单向传输；2、双向传输（半双向、全双向）

85 按用法的总线分类，， 1、专用（只连接一个物理设备）；2、非专用（同一时刻只有一对部件可使用总线进行通讯）。

86 总线按优先次序确定的方式，， 1、串行链接；2、定时查询；3、独立请求。

87 信息在总线上的传送方法，， 1、同步（部件间的信息传送由定宽、定距的系统时标同步）；2、异步（分为单向控制和请求/回答双向控制两种）。I/O 总线中使用最广泛的是异步双向互锁通讯方式。

88 数据宽度的分类，， 1、单字（单字节）：适合输入机，打字机等低速设备；2、定长块：适合磁盘等高速设备；3、可变长块：适合优先级高的中高速磁带、磁盘设备；4、单字加可变长块：适合速度低但优先级较高的设备。

89 中断的相关说明，， 引起中断的各种事件称中断源，中断源向中断系统发出的中断申请叫中断请求。中断响应是允许中断 CPU 现程序的运行，转去对请求进行预处理，包括保存好断点及现场，调出处理该中断的中断服务程序，准备运行。这部分工作通过交换新旧程序状态字（PSW）来实现。

90 中断系统的功能，， 1、中断请求的保存和清除；2、优先级的确定；3、中断断点及现场的保存；4、对中断请求的分析处理返回等。

91 根据信息传送方式的通道分类，， 1、字节多路：字符类低速设备；2、选择：优先级高的高速设备；3、数组多路：磁盘等高速设备。



- 92 通道流量计算公式, , 通道流量=Σ字节数/请求时间
- 93 对存储器的基本要求, , 1、容量大; 2、价格便宜; 3、速度快。
- 94 并行主存系统的定义, , 能并行读出多个 CPU 字的单体多字、多体单字、多体多字的交叉访问主存系统
- 95 存储体系的定义, , 通过 I/O 处理机和多道程序让程序的调入调出由软硬件来完成, 使速度接近于主存, 容量是辅存, 价格接近辅存, 称这种系统为存储体系。
- 96 解决主存和 CPU 速度差异的办法, , 1、在 CPU 中设置通用寄存器; 2、采用存储器的多体交叉并行存取来提高主存的等效速度; 3、采用 cache 存储器。
- 97 虚拟存储器的存储管理方式分类, , 1、段式(分段存储, 需要段地址表, 查表速度低段间会浪费); 2、页式(将主存空间和程序空间机械地等分成固定大小的页, 一般在 512B~几 K, 地址包含页号和页内位移, 需要设置页表记录地址对应关系); 3、段页式。
- 98 基址的定义, , 段、页等存放在主存中的起始地址。
- 99 段式管理的要求, , 系统要为每道程序分别设置段印象表, 还要操作系统为整个主存系统建立实主存管理表(包括占用区域表和可用区域表)。
- 100 段式和页式存储的特点, , 段式中每个段独立, 有利于程序员灵活实现段的链接, 修改段的容量, 每个段只包含一种类型的对象, 易于对特定类型的保护实现, 易于实现多个用户、进程对公用段的管理。页式对程序员透明, 所需映像表硬件少, 地址变换速度快, 浪费比段式少。
- 101 段页式存储管理的定义, , 实存机械等分成固定大小的页, 程序按模块分段, 每个段又分成与主存页面大小相同的页。
- 102 段页式与段式的区别, , 段页式段的起点不是任意的, 必须位于主存页面的起点。
- 103 段页式存储的主要问题, , 地址变换过程至少需要查表两次。
- 104 实页冲突, , 主存中的每个页面位置可对应多个虚页, 有可能发生两个以上的虚页想进入主存同一个页面位置的页面争用。
- 105 全相联印象的定义, , 每道程序的任何虚页都可以映像装入任何实页位置。全相联映像的实页冲突概率最低。
- 106 替换算法的分类, , 1、随机算法(RAND); 2、先进先出法(FIFO); 3、近期最少使用法(LRU); 4、优化替换算法(OPT), 理想化算法。3-4 为堆栈形替换算法。
- 107 命中率的一些说明, , 命中率是评价存储体系性能的重要指标, 程序地址流、替换算法、分配给程序的实页数都会影响命中率。
- 108 页面失效频率(PFF)的定义, , 设置一个值, 根据失效率来比较这个值, 当失效率低于此值则减少主页数, 反之则增加。
- 109 页面失效的定义, , 如果当前页在主存中, 跨页存放的那一页不在主存中, 就会在取指令、取操作数、或间接寻址等访问过程中发生页面失效。
- 110 颠簸的定义, , 指令或操作数跨页存放的那些页轮流从主存中被替换出去的现象。
- 111 快表和慢表的定义, , 快速硬件存放当前正在使用的虚实地址印象关系称为快表; 原先存放虚实地址映像关系的表称为慢表。
- 112 访问 cache 的时间, , 访问 cache 的时间一般是访问主存时间的 1/4~1/10, 一般为 50ns
- 113 cache 物理安放位置, , 为了发挥 cache 的高速性, 较小 cpu 与 cache 之间的传输延迟, 应该让 cache 在物理位置上尽量靠近处理机或就放在处理机中。
- 114 cache 映象与变换的方法分类, , 1、全相联映象及变换(主存中任意一块都可映象转入到 cache 中任意一块位置); 块冲突概率最低, 空间利用率高, 但代价大, 查表速度慢。2、直接映象及变换(主存空间按 cache 大小等分成区, 每个区内的各块只能按位置一一对应到 cache 的相应位置上); 优缺点和 1 相反。3、组相联映象及变换(将 cache 和主存空间都分成组, 各组之间是直接映象, 组内各块之间是全相联映象)。
- 115 cache 替换算法的分类, , 1、堆栈法; 2、比较对法(让各块成对组合, 用触发器的状态表示该比较对内两块访问的远近次序, 再经门电路找到 LRU 块)
- 116, cache 的透明性说明, , cache 存储器的地址变换和块替换算法是全硬实现的, 因此 cache 对应用程序员和系统程序员都是透明的, 对处理机和主存之间的信息交往也是透明的。



117 解决主存和 cache 内容不一致的方法, , 1、写回法/抵触修改法 (CPU 只写回 cache, 仅当需要替换时, 才将改写过的 cache 写回内存): 一般单处理机使用; 2、写直达法/存直达法 (CPU 写入 cache 同时, 也利用处理机和主存之间的直接通路, 写入主存): 多处理机使用。

118 cache 写不命中的处理方法, , 1、按写分配法: cache 写不命中时, 除写入主存外, 还要把单元所在的块从主存内容调入 cache, 写回法多采用按写分配法。2、不按写分配法: cache 写不命中时, 只写入主存, 写直达法多使用不按写分配法。

119, cache 的取算法相关说明, , cache 的取算法一般采用按需取进法, 即在 cache 块失效时, 才将要访问的字所在的块取进。何时取块分为恒预取 (使不命中率下降 75%~80%) 和不命中时预取 (使不命中率下降 30%~40%) 两种方法, 但恒预取传输量要求大。块的大小不宜超过 256 字节。

120 cache 的命中率, , 可达到 0.996

121 解释一条机器指令的微操作分类, , 1、取指令; 2、分析指令; 3、执行指令。

122 一次重叠的定义, , 指令分析部件和指令执行部件在任何时候都有两条相邻的指令在重叠解释的方式称之为一次重叠。

123 重叠在程序中的注意事项, , 应尽量减少使用条件转移指令

124 相关的定义, , 因机器语言程序中邻近指令之间出现关联, 需要同时解释而出错的现象。如数相关、指令相关等。

125 指令相关的处理方法, , 设置执行指令

126 数相关的处理方法, , 1、推后读; 2、设置相关通道。

127 流水的定义, , 流水是重叠的引申, 是把重叠的指令由两个过程细化分为多个子过程。

128 流水的分类, , 1、按扩展方向: 向下扩展分法和向上扩展分法; 2、按功能: 单功能和多功能流水线; 3、静态流水动态流水; 4、按数据表示: 标量流水和向量流水; 5、各功能段之间是否有反馈回路: 线性和非线性。

129 流水线处理机的吞吐率, ,  $TP=1/\text{最大单独过程时间}$

130 消除瓶颈的方法, , 1、将瓶颈进一步细分; 2、重复设置多套瓶颈段并联

131 流水最大吞吐率, ,  $TP_{\max}=1/\max \Delta T$

132 流水实际吞吐率, ,  $TP=n/(m\Delta t_0+(n-1)\Delta t_0)=TP_{\max}/(1+(m-1)/n)$  就是结果/时间; m: 流水线的段数, n: 完成任务数

133 流水加速比, ,  $Sp=nm\Delta t_0/(m\Delta t_0+(n-1)\Delta t_0)=m/(1+(m-1)/n)$  就是顺序时间/实际时间

134 流水线各段效率, ,  $n/(m+(n-1))$  阴影面积/全部时空面积

135 流水工作计算举例, 139 页,

136 全局性相关的定义, , 转移指令和其后指令间存在关联, 使之不能同时解释, 造成对流水机器的吞吐率下降的影响比指令相关、主存操作数相关和通用寄存器组相关及基址值或变址值相关严重得多, 称为全局相关, 后者称为局部相关。

137 任务在流水线中流动顺序的安排和控制方式分类, , 1、顺序流动方式或同步流动方式; 2、异步流动方式

138 读写相关的说明, , 写写相关和先读后写相关只有在异步流动时才能发生

139 全局性相关的处理方法, , 1、猜测法; 2、加快和提前形成条件码; 3、采取延迟转移; 4、加快短循环程序的处理

140  $V_i$  冲突定义, , 并行工作的各向量指令的源向量或结果向量使用了相同的  $V_i$  (向量寄存器)

141 功能部件冲突定义, , 同一功能部件被要求并行工作的多条向量指令使用。

142 链接技术的定义, , 在前一条指令的第一个结果分量到达向量寄存器组并可以用作本条向量指令的源操作数时, 立即启动本条指令工作形成链。是提高机器整体运算速度的重要措施。

143 阵列处理机的分类, , 1、分布存储器; 2、集中式存储器;

144 阵列处理机的特点, , 1、单指令流多数据流; 2、采用资源重复方式; 3、利用并行性中的同时性, 而不是并发性; 4、设备利用率没有流水高, 硬件价格低才能体现性价比; 5、适用于向量处理。

145 SIMD 的交换方法, , 1、线路交换: 在源和目的地建立实际连接线路, 适用于大批量数据传输; 2、包交换: 将数据置于包内传送, 无实际连接通路, 适用于短数据信息; 3、线路/包交换。



- 146 SIMD 网络拓扑结构分类, , 1、静态拓扑: 线型、环形、星形等。2、动态拓扑: 单级 (需循环多次才能实现任意两个单元的通信, 也叫循环网络) 和多级 (多个单级网络组成) 两类。
- 147 动态网络的单级网络分类, , 1、立方体、2、PM2I、3、混洗交换单级网络。
- 148 单级网络的最大距离, , 1、立方体是  $N$ ; 2、PM2I 是  $N/2$ ; 3、混洗交换网络是  $2n-1$ 。
- 149  $N=8$  的多级立方体互连网络, 170 页,
- 150,  $N=8$  的多级混洗交换网络 (OMEGA), 173 页, 网络中的各级编号的次序和多级立方体网络正好相反。
- 151 种多级互连网络灵活性对比, , 由高到低: adm, omega, 间接二进制  $N$  方体, staran。
- 152 并行存储器的冲突访问定义, , 阵列处理机中, 如果处理机需要同时访问存储器, 就会出现频宽下降的情况。为避免存储器冲突, 一般要求存储器分体数为质数。
- 153 多处理机的定义及特性, , 多处理机具有两台以上的处理机, 在操作系统的控制下通过共享主存或输入/输出子系统或高速通讯网络进行通讯, 属于多指令流多数据流系统 (MIMD)。
- 154 多处理机的分类, , 1、紧耦合多处理机: 通过共享主存实现处理间的通讯, 其通讯速率受限于主存频宽。2、松耦合多处理机: 每台处理机都有一个容量较大的局部存储器, 用于存储经常使用的指令和数据, 以减少紧耦合系统中存在的访主存冲突。
- 155 多处理机的机间互连形式, , 1、总线形式; 2、环形互连形式; 3、交叉开关形式; 4、多端口存储器形式; 5、开关枢纽结构形式。
- 156 多处理机的任务粒度设置依据, , 程序用于有效计算的执行时间  $E$  与处理机间通讯等辅助开销时间  $C$  的比值。  $E/C$  较大时, 并行效率较高。
- 157 多处理机的操作系统分类, , 1、主从型操作系统; 2、各自独立型操作系统; 3、浮动型操作系统。
- 158 脉动阵列机相关说明, , 由一组处理单元 PE 构成, 运算时数据在阵列结构的各个处理单元间沿各自目的方向同步推进。适用于要求处理计算量很大的信号/图像处理。
- 159 脉动阵列机的特点, , 1、结构简单, 规整, 模块化强, 可扩充性好, 适合用大规模集成电路实现; 2、PE 间数据通信控制等都简单规整; 3、具有极高的并行性, 可通过流水获得很高的运算效率和吞吐率。4、实现特定任务和算法, 专用性强, 限制了应用范围。
- 160 数据流机的定义, , 只要一条或一组指令需要的操作数全部准备就绪, 就可激发相应的指令或指令组执行, 不需要程序计数器, 指令执行无序, 受数据流驱动而执行。分为静态和动态两类。
- 161 智能机的构成元素, , 1、知识库机; 2、推理机; 3、智能接口处理机。

## 1、翻译和解释的区别和联系？

区别：翻译是整个程序转换，解释是低级机器的一串语句仿真高级机器的一条语句。

联系：都是高级机器程序在低级机器上执行的必须步骤。

## 2、为什么将计算机系统看成是多级机器构成的层次结构？

可以调整软、硬件比例;可以用真正的实处理机代替虚拟机器;可以在1台宿主机上仿真另一台。

## 3、计算机系统结构用软件实现和硬件实现各自的优缺点？

硬件优点：速度快，节省存储时间;缺点：成本高，利用率低，降低灵活性、适用性。

软件优点：成本低，提高灵活性、适用性;缺点：速度慢，增加存储时间、软件设计费。

## 4、就目前通用机来说，计算机系统结构的属性主要包括哪些？

数据表示、寻址方式、寄存器组织、指令系统、存储系统组织、中断系统、管态目态定义与转换、IO结构、保护方式和机构。

## 5、试述由上往下、由下往上设计思路 and 存在的问题？

由上往下：先考虑应用要求，再逐级往下考虑怎样实现。适用于专业机

由下往上：根据已有器件，逐级往上。六七十年代通用机设计思路。

以上方法存在的问题是软、硬件脱节。

## 6、采用统一高级语言方法、适用场合、存在问题和应采取的策略。

定义：是指为所有程序员使用的完全通用的高级语言。

适用场合：软件移植方便。

存在问题：目前语言的语法、语义结构不同;人们的看法不同;同一语言在不同机器上不通用;程序员的习惯

应采取的策略：可一定范围内统一汇编语言，结构相同机器间搞系列机。

## 7、由中间开始的设计思路及优点

既考虑应用也考虑现有器件，由软硬件分界面向两端设计。

优点：并行设计，缩短周期。

## 8、模拟和仿真的区别

模拟：机器语言解释，在主存中;仿真：微程序解释，在控制存储器中。

## 9、采用系列机方法、适用场合、好处、存在问题和应采取的策略

定义：根据软硬件界面的系列结构，设计软件 and 不同档次的系列机器。

适用场合：同一系列内软件兼容

好处：呼应“中间开始”设计思路;缓解软件要求稳定环境和硬件发展迅速的矛盾。

存在问题：软件兼容有时会阻碍系统结构的变革。

策略：坚持这一方法，但到一定时候要发展新系列，还可采用模拟仿真。

## 10、为什么要进行软件移植？

软件的相对成本越来越高，应重新分配软、硬件功能。但：

成熟软件不能放弃;已有软件修改困难;重新设计软件经济上不划算。

## 11、除了分布处理MPP和机群系统外，并行处理计算机按其基本结构特征可分为哪几种不同结构？例举它们要解决的问题。

流水线处理机：多个部件时间上并行执行。拥塞控制，冲突防止，流水线调度。

阵列处理机：空间上并行。处理单元灵活，规律的互连模式和互连网络设计，数据在存储器中的分布算法。

多处理机：时间和空间上的异步并行。多CPU间互连，进程间的同步和通讯，多CPU间调度。

数据流计算机：数据以数据令牌在指令间传递。硬件组织和结构，高效数据流语言。

## 12、采用模拟与仿真方法、适用场合、好处、存在问题和应采取的策略。

模拟

定义：用机器语言解释另一指令系统

适用场合：运行时间短，使用次数少，时间上无限制。

好处：可在不同系统间移植。

存在问题：结构差异大时，运行速度下降，实时性差。



策略：与仿真结合

仿真

定义：用微程序解释令—指令系统

适用场合：结构差别不大的系统

好处：运行速度快

存在问题：结构差别大时，很难仿真。

策略：与模拟结合，发展异种机连网。

### 13、多计算机系统和多处理机系统的区别

都属于多机系统，区别：

多处理机是多台处理机组成的单机系统，多计算机是多\*立的计算机。

多处理机中各处理机逻辑上受统一的OS控制，多计算机的OS逻辑上独立。

多处理机间以单一数据、向量、数组、文件交互作用，多计算机经通道或通信线路以数据流形式进行。

多处理机作业、任务、指令、数据各级并行，多计算机多个作业并行。

### 14、各种耦合度特征

最低：无物理连接，如脱机系统。

松散：通信线路互连，适于分布处理

紧密：总线或数据开关互联，实现数据、任务、作业级并行。

### 15、以实例说明计算机系统结构、组成、实现的相互关系与影响。

结构相同，可用不同的组成。如系列机中不同型号的机器结构相同，但高档机往往采用重叠流水等技术。

组成相同，实现可不同。如主存可用双极型，也可用MOS型等。 结构不同组成不同，组成的进步会促进结构的进步，如微程序控制。

结构的设计应结合应用和可能采用的组成。组成上面决定于结构，下面受限于实现。

组成与实现的权衡取决于性价比等;结构、组成、实现的内容不同时期会不同。

### 16、软件移植的途径，各受什么限制？

统一高级语言：只能相对统一

系列机：只能在结构相同或相近的机器间移植

模拟：机器语言差别大时，速度慢

仿真：灵活性和效率差，机器差异大时仿真困难。

### 17、并行处理数据的四个等级，给出简单解释，各举一例

位串字串：无并行性，如位串行计算机。

位并字串：一个字的所有位并行，如简单并行的单处理机。

位片串字并：多个字的同一位并行，如相连处理机。

全并行：同时处理多个字的多个位，如全并行阵列机。

### 18、设计乘法指令时，结构、组成、实现各考虑什么？

结构：是否设计乘法

组成：是否配置高速乘法器

实现：考虑器件集成度类型数量及微组装技术。

### 19、器件的发展对逻辑设计方法的影响

一是由逻辑化简转为采用组成技术规模生产，规模集成，并尽量采用通用器件

二是由全硬设计转为微汇编、微高级语言、CAD等软硬结合和自动设计

## 第二章

### 1、数据结构和机器的数据表示之间的关系？引入数据表示的基本原则？

数据结构要变换成数据表示来实现，不同的数据表示效率和方便性不同。它们是软硬件的交界面。

原则：1、是否提高效率减少时间，2、通用性和利用率。

### 2、简述三种面向的寻址方式的关系。

面向寄存器：速度快，增加硬件；



面向主存：速度稍慢，减少寄存器占用；

面向堆栈：速度慢，减轻编译负担。

三者各有特点，但可共同使用，不互相排斥。

3、堆栈型机器与通用寄存器型机器的主要区别？堆栈型机器对程序调用的哪些操作提供支持？

通用寄存器型对堆栈数据结构实现的支持较差：1、堆栈操作指令少；2、速度低；3、通常只用于保存返回地址。

堆栈型支持有力：1、硬件堆栈；2、堆栈指令丰富；3、支持高级语言编译；4、支持子程序嵌套和递归。可将以下信息全部压入栈：返回地址、条件码、关键寄存器内容、必要的全局或局部参数。为子程序开辟局部变量中间结果工作区。

4、比较寻址方式在指令中的两种指明方式。

操作码中的某位。缺点：不灵活。优点：指令总长短。

设寻址方式字段。缺点：指令总长长。优点：寻址灵活。

5、“机构型”和“策略型”的含义？

机构型功能指基本的通用的功能；策略型功能指不稳定的可能会改变的功能。

6、数据描述符和向量数据表示对向量数据结构所提供的支持有什么不同？

描述方法比变址方法简便，但不能解决向量数组高速运算问题；向量处理机能快速形成元素地址，能把元素成块预取到CPU，用一条指令同时对整个向量数组高速处理。

7、按CISC方向改进指令系统的思路具体办法？

思路可从面向目标程序、高级语言、操作系统三个方面考虑。

面向目标程序可采用的办法：1、对高频指令增强功能加快速度，低频指令合并和取消；2、复合指令代替子程序或宏。

8、huffman方法及优缺点？

构造方法：构造huffman树、每个节点用01表示、从根开始01编码

优点：平均码长最短，冗余少。缺点：种类多，不易译码。

9、huffman压缩概念的基本思想？

概率高的事件用短位数表示，低的用长位数表示。

10、以浮点数数据表示说明数的可表示精度、运算中的精度损失，尾数基值取小对哪个有利哪个不利？

数的可表示精度是数轴上数的离散程度，两个数间差越小，精度越高；运算中的精度损失指运算中尾数超长造成的损失。

尾数基值取小则可表示范围变小，个数减少，分布变密，精度提高，速度降低，对前者有利，对后者不利。

11、OS中哪些适合硬化或固化？哪些不适合？

高频使用的机构型功能适合，策略型功能不适合。

12、什么是程序的动态再定位？

程序在主存中的实际位置可以动态移动的定位技术。可以使用基址寄存器或映象表硬件。

13、缩短地址码长度的方法

基址、变址、相对寻址、分段、寄存器寻址、寄存器间接寻址等。

14、变址和基址各适用于何种场合？设计一种只用6位地址码就可以指向一大地址空间中任意64个地址之一的寻址机构。

变址适合标量计算机中，基址主要对逻辑与物理空间进行变换，支持动态再定位。

用6位地址码表示64个地址中的任意一个，可以用隐含寻址或PC自相对寻址形成物理地址。

15、设计RISC机器的一般原则及可采用的基本技术？

原则：精简指令条数、格式，让指令等长，一个周期内完成，增加通用寄存器，一般指令不可访存只能对寄存器操作，硬件实现为主，少量微程序解释，提高编译程序质量。

技术：按以上原则优化、实现指令，设置寄存器，指令采用重叠流水方式解释，采用延迟转移，提高便宜程序质量。

16、比较CISC和RISC，今后的发展是什么？



**CISC问题：**系统庞大导致成本高可靠性低，80%指令利用率低，性价比低。

**RISC优点：**简化指令系统，适合VLSI实现，解决了CISC的上述问题。

**缺点：**加重汇编语言负担，目标程序开销大，对浮点运算虚拟存储支持不强，对编译程序要求高。

**发展趋势：**二者互相结合，取长补短。

### 1、以**IBM370**为例说明为什么把中断分类以及分成几类。

大型多用途机器中断源多，每个中断源单独形成入口将导致硬件实现难，代价大，因此可归类，每类给中断服务程序入口由软件转入相应处理部分。**IBM370**中断分为6类：机器校验，访管，程序性，外部，I/O，重新启动。

### 2、专用和非专用线各自的优缺点

**专用线：**只连接一对物理部件的总线。优点：不用争总线，控制简单，系统可靠。缺点：总线数多且长，成本高，利用率低，不易扩展。

**非专用线：**可被多种功能与部件共享，但同一时刻只能被一个部件使用。优点：集成度高，造价低，可扩展能力强，总线利用率高，易标准化。缺点：流量小，争用总线，部件效率低，可能成为瓶颈，可靠性差。

### 3、减少总线线数的方法

**线的组合：**减少按功能和传递方向所需的线数。

**编码：**对少数几根功能线进行编码取代多根单功能线。

**并/串-串/并：**在总线两端设置转换器，经分拆移位后在目的端形成完整的字。

### 4、比较几种数据宽度

**单字：**适合低速设备，不用指明传送信息单位，缺点是速度慢总线利用率低；

**定长块：**高速设备，充分利用总线宽度不用指明传送信息单位，但不灵活；

**可变长块：**优先级速度都高的设备，灵活需充分利用总线带宽，但需指定传送信号块大小。

**单字加定长块：**优先级高速度低的设备，短数据可用单字传送减少带宽浪费，信息块太小时成为单字方式总线利用率低。

**单字加可变长块：**普遍使用，灵活有效，但复杂开销大。

### 5、有通道情况下的**I/O**过程

a目态程序中可安排I/O广义指令

b运行到广义指令后，产生访管中断

cCPU响应中断后进入管态

d管理程序根据广义指令编写通道程序，进入通道选择设备期

e选择通道和子通道，取出指令，选择控制器和设备，发启动命令，结束通道选择设备期

f进入通道数据传送期，完成数据传送

g向CPU发I/O请求，第二次转管态，调出相应管理程序，之后CPU返回目态。

### 6、比较通道的三种类型

**字节多路：**单字节，适于大量低速设备，字节交叉，多次选择设备，分时共享，满负荷时对通道要求的实际流量应是所连各设备的流量之和。

**数组多路：**定长块，适于大量高速设备，成组交叉，多次选择设备，分时共享，满负荷时对通道要求的实际流量应是所连各设备的流量最大的那个。

**选择：**不定长块，高优先级高速设备，独占通道，一次选择设备，独占，满负荷时对通道要求的实际流量应是所连各设备的流量最大的那个。

### 7、为什么中断优先级从高到低一般为：机器校验、程序性和管理程序调用、外部、I/O、重新启动？

机器校验若不及时处理，系统将无法正常工作。

程序中断若低于外部和I/O中断可能导致混乱。只有处理完机器故障后，才能进入访管中断。

重新启动一般时间并不紧迫，所以放在最后。

### 8、集中式串行链接的过程，优缺点，硬件故障时通讯的可靠性。

a经公共总线向总线控制器申请

b总线不忙时，总线控制器响应请求，送出总线可用。

c总线可用信号在部件间串行通过，直至某个部件发生总线请求。



d该部件获得总线使用权

e数据传送，维持总线忙

f传送完成，去除总线忙

g总线请求再次建立时，重复新的分配过程。

优点：简单，线数少，可扩充性好，可靠性高。

缺点：对总线可用线及电路敏感，不灵活，速度慢。

9、集中式定时查询的过程，优缺点，硬件故障时通讯的可靠性。

a每个部件发总线请求

b总线不忙时，定时查询谁发的请求

c查询到后，查询停止，该部件获得总线使用权

d数据传送，维持总线忙

e传送完成，去除总线忙

f总线请求再次建立时，重复新的分配过程。

优点：灵活，可靠性高

缺点：线数多，扩展性差，控制复杂，总线分配受限计数信号。

10、集中式独立请求的优缺点，硬件故障时通讯的可靠性。

a每个部件有总线请求和总线准许

b总线未分配时，总线分配器根据某种算法仲裁哪个申请部件使用总线。

c数据传送

d传送完毕后除去总线已分配和总线准许。

优点：速度快，灵活，方便隔离失效部件

缺点：线数多，复杂。

11、在现代计算机系统中，中断系统的软硬件功能是怎样分配的，为什么这样分配？

中断响应要求快，一般用硬件实现。

中断的处理过程一般用软件，也可用硬件支持。

中断响应过程中现场的保存和恢复用硬件实现，以保证响应速度。

另一部分现场用软件实现，提高灵活性。

1、在分体交叉存储器中为什么实际频宽不随M增大而线性增大？

M大-数据总线长-负载重-增加门级数-增加延迟；

顺序取指效率可提高M倍，但出现转移效率就会下降。

2、段式存储管理的地址转换过程。

程序号、段号、段内偏移量

1)由程序号找到相应的段表基址寄存器，查到段表始址和段表长度。

2)检查是否越界，正常转3

3)由段表始址和段号找到装入位等相应表项

4)装入位为1转5，否则产生缺页中断

5)主存地址+段内偏移=物理地址