高等教育自学考试辅导丛书

计算机系统结构辅导与练习

计算机及应用专业（独立本科段）

（2001 年版）

李学干　　主编

经 济 科 学 出 版 社

责任编辑：王东岗
责任校对：孙　昉
版式设计：代小卫
技术编辑：邱　天

计算机系统结构辅导与练习
计算机及应用专业（独立本科段）
李学干　主编
经济科学出版社出版、发行　新华书店经销
社址：北京海淀区阜成路甲28号　邮编：100036
总编室电话：88191217　发行部电话：88191540
网址：www.esp.com.cn
电子邮件：esp@esp.com.cn
天宇星印刷厂印刷
河北三河新路装订厂装订
787×1092　16开　14印张　360000字
2001年4月第一版　2001年4月第一次印刷
印数：00001—10100册
ISBN 7-5058-2480-5／F·1872　定价21.00元

跨越于从巨型到微型计算机的整个系列之中了。这种非常引人注目的局面是近一二十年里形成的。

总之，不管是对研究计算机结构设计的技术人员，还是对硬件设计、软件设计和应用系统设计的技术人员来讲，较为系统全面地学习、了解和掌握计算机系统结构的基本知识、内容和发展等是非常必要的。

通过对本课程的学习，应使自学者能进一步树立和加深计算机系统的整体概念，特别是着眼于正确掌握有关计算机系统结构的基本概念、基本原理，了解目前采用的比较成熟的基本结构，掌握结构设计的基本思想和方法，从而提高分析问题和解决问题的能力。同时，也让自学者能了解到近十几年里，在并行处理和系统结构技术上的一些重要进展及今后可能的发展趋势。

二、课程内容的重点

《计算机系统结构》共分 8 章。

第 1 章　计算机系统结构的基本概念

本章的重点是：计算机系统结构、计算机组成、计算机实现三者的定义及所包含的内容；有关透明性问题的判断；软件和硬件的功能分配原则；软件可移植性的途径、方法、适用场合、存在问题和对策；有关并行性的概念；系统结构中开发并行性的途径和类型等。

第 2 章　数据表示与指令系统

本章的重点是：自定义数据表示；浮点数尾数的基值选择；数的下溢处理方法；寻址方式中的再定位技术；信息在存储器中按整数边界存储的概念；操作码和指令字格式的优化；CISC 指令系统的改进途径综述；RISC 概念及所采用的基本技术等。

第 3 章　总线、中断与输入输出系统

本章的重点是：总线的控制方式；数据宽度；中断的分类和分级；中断处理（完）次序的安排和实现；通道流量的分析和设计等。

第 4 章　存储体系

本章的重点是：段页式和页式虚拟存储器的原理；页式虚拟存储器的地址映象；LRU、FIFO、OPT 替换算法进行页面替换的过程模拟，命中率的计算；LRU 替换算法对页地址流的堆栈处理模拟及性能分析；Cache 存储器的直接映象和组相连映象规则；用 LRU 替换算法进行 Cache 块替换的硬件实现及替换过程模拟；Cache 存储器的透明性及性能分析等。

第 5 章　重叠、流水和向量处理机

本章的重点是：重叠方式中的“一次重叠”原理及各种相关的处理；流水处理机的主要性能及其分析，局部性相关的处理；流水处理机的全局性相关的处理，中断的处理；单功能非线性流水线的调度；在向量处理机中，向量指令之间的串行、并行和链接执行的分析，及所需时间的计算等。

第 6 章　阵列处理机

本章的重点是：阵列处理机与流水线处理机结构特点的对比；SIMD 计算机的基本单级互联网络及互联函数；多级互联网络中的立方体和混洗交换网络；并行存储器实现无冲突访问的存储单元分布规律等。

跨越于从巨型到微型计算机的整个系列之中了。这种非常引人注目的局面是近一二十年里形成的。

总之，不管是对研究计算机结构设计的技术人员，还是对硬件设计、软件设计和应用系统设计的技术人员来讲，较为系统全面地学习、了解和掌握计算机系统结构的基本知识、内容和发展等是非常必要的。

通过对本课程的学习，应使自学者能进一步树立和加深计算机系统的整体概念，特别是着眼于正确掌握有关计算机系统结构的基本概念、基本原理，了解目前采用的比较成熟的基本结构，掌握结构设计的基本思想和方法，从而提高分析问题和解决问题的能力。同时，也让自学者能了解到近十几年里，在并行处理和系统结构技术上的一些重要进展及今后可能的发展趋势。

二、课程内容的重点

《计算机系统结构》共分 8 章。

第 1 章　计算机系统结构的基本概念

本章的重点是：计算机系统结构、计算机组成、计算机实现三者的定义及所包含的内容；有关透明性问题的判断；软件和硬件的功能分配原则；软件可移植性的途径、方法、适用场合、存在问题和对策；有关并行性的概念；系统结构中开发并行性的途径和类型等。

第 2 章　数据表示与指令系统

本章的重点是：自定义数据表示；浮点数尾数的基值选择；数的下溢处理方法；寻址方式中的再定位技术；信息在存储器中按整数边界存储的概念；操作码和指令字格式的优化；CISC 指令系统的改进途径综述；RISC 概念及所采用的基本技术等。

第 3 章　总线、中断与输入输出系统

本章的重点是：总线的控制方式；数据宽度；中断的分类和分级；中断处理（完）次序的安排和实现；通道流量的分析和设计等。

第 4 章　存储体系

本章的重点是：段页式和页式虚拟存储器的原理；页式虚拟存储器的地址映象；LRU、FIFO、OPT 替换算法进行页面替换的过程模拟，命中率的计算；LRU 替换算法对页地址流的堆栈处理模拟及性能分析；Cache 存储器的直接映象和组相连映象规则；用 LRU 替换算法进行 Cache 块替换的硬件实现及替换过程模拟；Cache 存储器的透明性及性能分析等。

第 5 章　重叠、流水和向量处理机

本章的重点是：重叠方式中的“一次重叠”原理及各种相关的处理；流水处理机的主要性能及其分析，局部性相关的处理；流水处理机的全局性相关的处理，中断的处理；单功能非线性流水线的调度；在向量处理机中，向量指令之间的串行、并行和链接执行的分析，及所需时间的计算等。

第 6 章　阵列处理机

本章的重点是：阵列处理机与流水线处理机结构特点的对比；SIMD 计算机的基本单级互联网络及互联函数；多级互联网络中的立方体和混洗交换网络；并行存储器实现无冲突访问的存储单元分布规律等。

第 7 章　多处理机

本章的重点是：多处理机的机间互连；程序中的并行算法、并行性分析；并行语言中有关并行任务的派生和汇合的表示及时空图描述等。

第 8 章　其他计算机结构

本章内容包括有脉动阵列机、MPP、机群系统、数据流机、归约机、智能机等的一般介绍，重点在数据流机的数据流机器语言的数据流图表示上。

三、课程的基本要求

1. 建立好有关计算机系统结构的基本概念，弄清结构、组成和实现三者的定义、内含和相互关系，透明性概念，软件的可移植性手段，系统结构和并行性的发展。

2. 掌握数据表示、寻址方式、指令系统、总线、中断、I/O 系统、存储体系等的软、硬件功能分配，设计思路，优化改进途径及性能分析。了解 RISC 的结构特点，掌握其所采用的基本技术。

3. 掌握用重叠、流水等组成方式提高系统速度的原理，时空图画法，性能指标计算，各种相关的处理，流水线的调度，向量的流水处理，向量指令间的并行与链接等。理解向量、超标量、超长指令字、超流水线等处理机的结构原理和工作方式。

4. 掌握阵列处理机的结构原理、互联网络设计，了解某些典型的并行算法。

5. 掌握多处理机的结构、特点、机间互连、程序的并行性分析、并行任务的派生与汇合。了解任务粒度的分析和性能模型建立的方法与某些结论。

6. 一般了解现代其他有代表性系统的结构。

对基本概念和原理要着眼于理解，能归纳出基本要点。对基本结构要理解其构成思想，掌握分析和思考问题的方法。对各章的重点和难点要领会其实质，要融会贯通，会综合运用。对书中的各种典型习题能正确解答。

四、学习中应注意的问题和学习方法

本课程的内容广泛，涉及面较宽，要求自学者有数字逻辑、程序设计语言、概率论、数理统计、计算机组成原理、数据结构、操作系统等方面的基础知识。课程的内容概念性和理论性很强，且有一定的深度，学习起来难度是比较大的。尤其是对只接触过微处理机，而对中、大型机缺乏感性知识的自学者来讲，深刻领会和吃透教材中各章节的知识点，应该说是要下一定功夫的。

自学本课程时，应注意下述几点：

1. 在学习本课程的教材之前，自学者应仔细阅读教材后所附的自学考试大纲，对大纲中所列的基本要求，本课程与相关课程的联系等要有一定的了解。

2. 在学习教材每一章前，自学者应了解清楚自学考试大纲上，对该章所定的学习目的和总的要求、考核要求和考核知识点，弄清其中的重点和难点在什么地方，以及对这些知识点的能力层次要求。

3. 学习时，应依据教材和本辅导教材，对其中的重点和难点部分多花些时间消化领会，对所学的基本原理、方法等理论性、概念性较强的部分，应着眼于理解其精神实质，进行必要的归纳和小结，理出有关要点和思路。

4. 学习中若遇到一时理解不深或搞不清的问题，可将其暂时搁置，等学习完其他各章后，再来重读、领会和加深。

5. 对于典型的应用或计算、设计等问题，应领会其分析的方法和解题的思路和步骤。

6. 学完每一个章节后，应认真完成教材中所列的习题，不要急于去看本辅导教材所提供的解答。等到做完后，再和本书提供的解题范例或步骤进行比较，分析其间有什么大的差别和问题，通过融会贯通，就能进一步加深对所学知识的理解和掌握，提高灵活运用知识和分析、解题的能力。不能采用背题和死记参考答案的办法，否则，题目稍加变化，就不会做了。

第二部分

各章要点、题例分析及测试题解答

在本部分中，按照全国高等教育自学考试委员会审定的计算机及应用专业（独立本科段）《计算机系统结构自学考试大纲》规定的考核要求，对《计算机系统结构》教材中各章的基本知识点，进行精简提炼和归纳说明，给出要点和结论。使考生能明白课程中各章内容的重点和难点在什么地方。

通过对代表性的习题和题例进行分析解答，以及配备的大量自测题和参考答案，来加深对内容的知识点、重点、难点的领会和掌握，了解考核的深度、广度和考核的形式。

因为同一知识点，完全可以以不同的形式来命题，所以，最终的考试题型和题目不见得会和这里所提供的自测题相同，但是只要掌握了这些内容和要点，领会了解题的方法和技巧，考生就能比较容易地通过对本课程的考核。

第 1 章　计算机系统结构的基本概念

一、概述

本章着眼于建立和掌握计算机系统结构设计应具备的基本知识和概念，为进一步深入学习后续各章打下基础。

本章的基本要求是：

1. 领会一台完整的通用计算机系统，可以被看成是由多个不同机器级构成的多级层次结构，每一级都可以看成是一台机器，都有其自己的机器语言和实现方法。这样的多级层次结构一般可分为哪几级，各机器级所处的相对位置及所用的主要实现方法。

2. 掌握计算机系统结构、计算机组成和计算机实现三者的定义，各自研究的方面和内容。领会计算机系统结构是软件和硬件的主要分界面的概念。理解计算机系统结构、计算机组成和计算机实现三者间存在着相互的影响。领会透明性概念，能从不同角度对某个具体问题正确选择是否应设计成透明的结论。

3. 领会一个功能分别用软件和硬件实现的优点和缺点。掌握在功能分配中的软、硬件比例取舍的基本原则。领会计算机系统分别采取“由上往下”、“由下往上”设计的方法、各自存在什么问题。对通用机为什么应采取“从中间开始”向两边设计的方法，如何进行设计，这样的设计有什么优点。

4. 理解系统结构设计为什么要解决好软件的可移植性。掌握采用统一高级语言、系列机、模拟和仿真三种途径来实现软件移植，各自的方法、适用场合、存在问题及应采取的对策。领会系列机软件所谓向前、向后、向下、向上兼容的定义，以及系列机对软件兼容的基本要求。能正确判断在系列机中发展新型号机器时，哪些做法是可取的。

5. 了解应用和器件的发展对系统结构设计的影响。理解非用户片、现场片和用户片的定义，以及器件发展是如何改变了逻辑设计的传统做法。

6. 领会并行性的定义，并行性的二重含义和开发并行性的三种途径。掌握各种并行性等级的划分和并行性级别高低的顺序。了解计算机系统沿三种不同的并行性发展途径开发出的多机系统类型和特点。了解多机系统的耦合度概念。了解计算机按指令流、数据流及其多倍性进行分类的方法，以及典型机器结构的例子。

本章的重点是：计算机系统结构、计算机组成、计算机实现三者的定义及所包含的内容；有关的透明性问题判断；软件和硬件的功能分配原则；软件可移植性的途径、方法、适用场合、存在问题和对策；有关并行性的概念；系统结构中开发并行性的途径和类型等。

本章的难点是：透明性的判断与分析。

二、本章要点及题例的分析解答

（一）计算机系统的多级层次结构

1．多级层次结构的划分

现代通用的计算机系统从功能和概念上，可以看成是由多个机器级组成的一个层次结构。在层次结构中，按由高到低的次序分别是应用语言机器级、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。对每一个机器级的用户来说，都可以将此机器级看成是一台独立的机器，都可以有自己的机器语言。

2．各机器级的实现技术

各个机器级的实现技术不外乎是翻译或是解释，或者是翻译和解释的结合。

翻译，是先用转换程序将高一级机器级上的程序，整个地变换成低一级机器级上等效的程序，然后再在低一级机器级上实现的技术。解释，是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，并通过对高级机器级程序中的每条语句或指令逐条解释来实现的技术。

采用翻译技术实现的典型例子有：用翻译程序将应用语言机器级上的应用程序包，翻译成高级语言程序，用编译程序实现将高级语言源程序转换成机器语言目标程序，用汇编程序实现将汇编语言源程序转换成机器语言目标程序。解释实现的典型例子有用解释方式在传统机器上执行高级语言程序，用微指令程序解释实现机器指令，用微程序或机器指令程序解释实现操作系统的操作命令等。

软件和硬件在逻辑功能上是等效的，但是，在性能、价格、实现的难易程度上却是各不相同的，是不等效的。

就目前状况来讲，应用语言机器级、高级语言机器级、汇编语言机器级、操作系统机器级是以软件为主来实现的，但有时也根据性能价格的要求不同，增加了不少硬件的支持。我们称以软件为主实现的机器为虚拟机器。微程序机器级和用组合逻辑控制的传统机器语言机器级是用硬件实现的，而采用微程序控制的传统机器语言机器级是用固件实现的。我们称以硬件或固件为主实现的机器为实际机器。固件（firmware）是一种具有软件功能的硬件，如将软件固化在只读存储器这种大规模集成电路的硬、器件上，就是一种固件。

3．典型题例的分析与解答

〔**题** 1.1〕有一台经解释实现的计算机，可以按功能划分成 4 级。每一级为了执行一条指令，需要下一级的 N 条指令来解释。如果执行第 1 级的一条指令要 Kns 时间，那么执行第 2、第 3 和第 4 级的一条指令各需要用多少时间？

〔**分析**〕因为计算机按功能分成级时，最底层是第 1 级，向上依次为第 2、第 3 和第 4 级。解释方式的执行，是在低级机器级上用它的一串语句或指令，来仿真高一级机器级上的

一条语句或指令的功能。而且是通过对高级机器级程序中的每条语句或指令逐条加以解释来实现的。

〔解答〕执行第 2、第 3 和第 4 级的一条指令各需 KN ns、KN^2 ns、KN^3 ns 的时间。

〔题 1.2〕操作系统机器级的某些指令就采用传统机器级的某些指令，这些指令可以直接用微程序解释实现，而不必由操作系统自己来实现。你认为这样做有哪些好处？请答出主要的两点即可。

〔解答〕这样做，可以加快操作系统中操作命令解释的速度，同时也节省了存放解释操作命令这部分解释程序所占的存储空间，简化了操作系统机器级的设计，也有利于减少传统机器级的指令条数。

〔题 1.3〕有一个计算机系统可按功能分成 4 级，每级的指令互不相同，每一级的指令都比其下一级的指令在效能上强 M 倍，即第 i 级的一条指令能完成第 i－1 级的 M 条指令的计算量。现若需第 i 级的 N 条指令解释第 i＋1 级的一条指令，而有一段第 1 级的程序需要运行 Ks，问在第 2、第 3 和第 4 级上一段等效的程序各需要运行多长时间？

〔分析〕因为从功能意义上讲，第 i 级的一条指令能完成第 i－1 级的 M 条指令的计算量，但第 i 级的一条指令的执行，都是靠第 i－1 级的 N 条指令的执行来解释完成。已知，第 1 级的一段程序运行时间为 Ks。第 2 级的一段程序从功能等效上讲，所需的指令条数应当是第 1 级上指令数的 1/M。而由第 1 级解释时又需要执行 N 条指令，所以，第 2 级一段等效程序就需要 $K\frac{N}{M}s$ 的时间。第 3、第 4 级则可依次类推。

〔解答〕第 2、第 3 和第 4 级上的一段等效程序分别需要 $K\frac{N}{M}s$、$K\frac{N^2}{M^2}s$ 和 $K\frac{N^3}{M^3}s$ 的时间。

〔题 1.4〕硬件和软件在什么意义上是等效的？在什么意义上又是不等效的？试举例说明。

〔解答〕硬件和软件在逻辑功能上是等效的。在原理上，用软件实现的功能完全可以用硬件或固件（微程序解释）来完成。用硬件实现的功能也可以通过用软件进行模拟来完成。只是反映在速度、价格、实现的难易程度上这两者是不同的。

例如，编译程序、操作系统等许多用机器语言软件子程序实现的功能，完全可以用组合电路硬件或微程序固件来解释实现。它们的差别只是软件实现的速度慢，软件的编制复杂，编程工作量大，程序所占的存储空间量较多，这些都是不利的；但是，所花硬件少，硬件实现上也就因此而简单容易，并且硬件的成本低，解题的灵活性和适应性较好，这些都是有利的。又如，乘除法运算，可以经机器专门设计的乘法指令用硬件电路或乘除部件来实现，也可以通过执行一个使用相加、移位、比较、循环等机器指令组成的机器语言子程序来实现。向量、数组运算，是在向量处理机中，直接使用向量、数组类指令和流水，或阵列等向量运算部件硬的方式来实现，但在标量处理机上也可以通过执行用标量指令组成的循环程序软的方式来完成。

浮点数运算，可以直接通过设置浮点运算指令用硬件来实现，也可以将两个定点数分别表示浮点数的阶码和尾数，通过程序方法把浮点数阶码和尾数的运算映象变换成两个定点数的运算，用子程序软的方式来实现。十进制数的运算，可以通过专门设置十进制运算类指令和专门的十进制运算部件硬的方式完成，或者通过设置 BCD 数的表示和若干 BCD 数运算的校正指令来软硬结合地实现，也可以先经 10 转 2 的数制转换子程序，将十进制数转成二进

制数，再用二进制运算类指令运算，所得结果又调用 2 转 10 的数制转换子程序转换成十进制数结果以全软的方式实现。

（二）计算机系统结构、组成与实现

1. 计算机系统结构

计算机系统结构是指多级层次结构中传统机器级的结构，它是软件和硬件/固件的主要交界面，是让编制的机器语言程序、汇编语言源程序，以及将高级语言源程序编译生成的机器语言目标程序，能在机器上正确运行所应看到的计算机属性。计算机系统结构是与汇编语言程序或机器语言程序所能实现的功能，以及要用到的数据类型、寻址方式等密切相关。

2. 计算机组成与实现

计算机组成，主要指的是机器级内部数据流和控制流的组成及逻辑设计。它与指令和编程等没有直接关系，主要是看硬件系统在逻辑上如何组织。计算机组成主要与计算机操作的并行度、重叠度，部件的共享度等有关，直接影响系统的速度和价格。

计算机实现，指的是计算机组成的物理实现，它着眼于用什么样的器件技术和微组装技术。它也直接影响到系统的速度和价格。

3. 结构、组成与实现三者的内涵

教材中列举了计算机系统结构、计算机组成和计算机实现各自应研究的一些内容和方面，它们都是对各个设计不透明的方面。

例如，机器中应设哪些机器指令和汇编指令、主存的容量和编址方式、寄存器的数量和使用规定等，均由计算机系统结构设计来确定。而指令中微操作顺序的编排，主存是否采用多体并行交叉组织是计算机组成设计考虑的。是否采用超大规模集成电路，如何将部件在物理上组装到一起，则是由计算机实现设计来考虑的。

4. 计算机的透明性概念

在计算机中，客观存在的事物或属性从某个角度看不到，称对他是透明的。计算机中的“透明”与社会生活中的透明，涵义正好相反。社会生活中所称的“透明”，是要公开，让大家看得到的意思，而计算机中的“透明”，则是指看不到的意思。

对目前多数的通用计算机来说，采用什么系列机，机器级和汇编级的指令系统，指令的条数、种类、功能、格式和编码，主存的容量、编址空间和所用的编址方式，硬件直接识别的数据类型、格式和种类，I/O 系统采用通道处理机还是外围处理机，I/O 设备的编址，I/O 接口的使用规定等，对计算机系统结构都设计成不透明的。而系列机内部搞哪几种型号的计算机，指令的解释采用顺序、重叠还是流水，乘法指令是用加法器和移位器经一连串时钟脉冲控制实现其操作，还是用专门的高速乘法器来实现，主存采用单体还是多体交叉并行组织，数据总线线数的多少即数据通路宽度的大小，通道采用结合型还是独立型，系统采用单总线还是多总线，控制器微操作信号是用微程序控制器产生，还是用组合逻辑电路控制器产生等，所有这些对计算机组成设计来说都应是不透明的。

5．结构、组成与实现三者的相互影响

相同结构的计算机可以因速度不同而采用不同的组成，相同的组成也可有多种不同的实现。这都取决于计算机系统的性能、价格及器件技术的状况。

结构不同会影响到可用的组成技术有不同，而不同的组成又会反过来影响到系统结构的设计。因此，系统结构的设计必须结合应用来考虑，要为软件和算法的实现提供更多更好的硬件支持，同时要考虑可能采用和准备采用哪些组成技术，不能过多或不合理地限制各种组成、实现技术的采用与发展。

组成与实现可以折衷权衡，它主要取决于器件的来源、厂家的技术特长和性能价格比能否优化。应当在当时的器件技术条件下，使价格不增或只增很少的情况下，去尽可能地提高系统的性能。

6．典型题例的分析与解答

〔**题** 1.5〕什么是透明性概念？对计算机系统结构，下列哪些是透明的？哪些是不透明的？

存储器的模 m 交叉存取；浮点数据表示；I/O 系统是采用通道方式还是外围处理机方式；数据总线宽度；字符行运算指令；阵列运算部件；通道是采用结合型还是独立型 PDP－11 系列的单总线结构；访问方式保护；程序性中断；串行、重叠还是流水控制方式；堆栈指令；存储器的最小编址单位；Cache 存储器。

〔**分析**〕所谓透明就是看不到，不属于其管的部分。对计算机系统结构是否透明，首先要弄清教材第 1 章的 1.2.1 节中有关计算机系统结构的定义和所包含的属性内容。简单来说，凡是编写机器语言和汇编语言程序都要用到的数据表示、指令系统、寻址方式、寄存器组织、机器级 I/O 结构、存储容量及其编址方式、中断机构、系统管态和目态间的切换、信息保护方式和机构等都对计算机系统结构是不透明的。而全部由硬件实现的，或是在机器语言、汇编语言编程中不会出现和不需要了解的部分，以及只影响机器的速度和价格的逻辑实现（计算机组成）和物理实现（计算机实现）的那些部分，对计算机系统结构来说都是透明的。

〔**解答**〕客观存在的事物或属性，从某个角度去看，却看不到，称这些事物和属性对他是透明的。透明了就可以简化这部分的设计，然而因为透明而无法控制和干预，这会带来一些不利的影响。因此，透明性的取舍要正确选择。

对计算机系统结构，透明的应当是：存储器的模 m 交叉存取、数据总线宽度、阵列运算部件、通道是采用结合型还是独立型、PDP－11 系列的单总线结构、串行、重叠还是流水控制方式，Cache 存储器。

对计算机系统结构不透明的是：浮点数据表示、I/O 系统采用通道方式还是外围处理机方式、字符行运算指令、访问方式保护、程序性中断、堆栈指令、存储器最小编址单位。

〔**题** 1.6〕从机器（汇编）语言程序员看，以下哪些是透明的？

指令地址寄存器；指令缓冲器；时标发生器；条件码寄存器；乘法器；主存地址寄存器；磁盘外设；先行进位链；移位器；通用寄存器；中断字寄存器。

〔**分析**〕从机器（汇编）语言程序员看，实际上也就是从计算机系统结构看的内容。

指令地址寄存器就是程序计数器，汇编语言或机器语言程序都要用到它，其位数多少会

影响到可执行程序的空间大小。指令缓冲器、主存地址寄存器都属于计算机组成中的缓冲器技术，是全硬件实现的，系统程序不参与对它们的管理。时标发生器、乘法器、先行进位链、移位器等都属于计算机组成中的专用部件配置，它只影响机器的速度和价格，与软件编程无关。条件码寄存器是存放指令执行后，生成反映结果状态或特征的标志码，它要供转移等指令使用，是编程要用到的。磁盘外设的种类、编址方式、容量等都是磁盘管理服务程序要用到的。通用寄存器的数量、位数、编址、使用规定在汇编语言程序和机器语言程序中都会直接用到。中断字寄存器是用来记录每一个中断类中，各个中断源发生中断请求的状况，它将为中断服务程序在处理中断时要用到的。

〔**解答**〕从机器（汇编）语言程序看透明的有：指令缓冲器、时标发生器、乘法器、主存地址寄存器、先行进位链和移位器。

〔**题** 1.7〕下列哪些对系统程序员是透明的？哪些对应用程序员是透明的？

系列机各档不同的数据通路宽度；虚拟存储器；Cache 存储器；程序状态字；“启动 I/O”指令；“执行”指令；指令缓冲寄存器。

〔**分析**〕系统程序员是编写诸如操作系统、编译程序等各种系统软件的人员。应用程序员是指利用计算机及所配的系统软件支持，来编写解决具体应用问题的程序员。他们都可以使用汇编语言或机器语言来编写程序，当然也可以用高级语言来编写程序。所以，对系统程序员或应用程序员不透明的，应包括计算机系统结构所包含的方面。而属全硬件实现的计算机组成所包含的方面，如系列机各档不同的数据通路宽度、Cache 存储器、指令缓冲寄存器等，无论是对系统程序员，还是应用程序员都应当是透明的。对目前高性能计算机系统来讲，大多数都是多用户环境，应用程序（也称算态、目态或用户态程序）中是不允许使用管态（也称系统态、监督态）中所用的特权指令。

例如，大型多用户系统中，程序状态字是用于反映计算机系统在当前程序的各种关键状态（它并不是 IBM PC 计算机那种狭义的所谓程序状态字），它是操作系统用于管理计算机系统资源及其使用状况的，用户不能直接对程序状态字内容进行读、写和访问的，只能由系统来管理。“启动 I/O”指令是大型机中的一种管态指令，属于特权指令，只能在操作系统程序中使用（参见教材中第 3 章的 3.4.1 节所介绍）。用户程序是不能用它来直接启动 I/O 通道和设备的。虚拟存储器（参见教材中第 4 章 4.1.3 节）是一个主存—辅存两级存储层次。它对应用程序是完全透明的，使应用程序不必做任何修改就可以在系统上运行。但是，在操作系统中必须配置有相应的管理软件，能对其虚实外部地址的映象和变换、程序的换道、程序由辅存调入主存、主存页面的替换、存储保护等进行管理，所以对系统程序员来说是不透明的。“执行”指令（参见教材中第 5 章 5.1.2 节）是 IBM370 等系列机上用于解决程序在执行过程中不准修改指令，又允许将指令放在操作数区中做修改，以满足指令在执行过程中允许修改的要求。这种指令无论是用户程序，还是系统程序，都希望可以被使用的，所以，“执行”指令应设计成对应用程序和系统程序都是不透明的。

〔**解答**〕系列机各档不同的数据通路宽度、Cache 存储器、指令缓冲寄存器属计算机组成，对系统程序员和应用程序员都是透明的。虚拟存储器、程序状态字、“启动 I/O”指令，对系统程序员是不透明的，而对应用程序员却是透明的。“执行”指令则对系统程序员和应用程序员都是不透明的。

（三）软、硬件取舍与计算机系统的设计思路

1. 软硬件取舍

计算机系统结构设计主要是确定软件和硬件的功能分配。在计算机系统上，一个功能用硬件实现可以提高其执行的速度，减少程序所需要的存储空间，降低软件部分所需的成本，但同时会提高硬件部分的成本，降低硬件的时间利用率和系统的灵活性和适应性。因此，在确定计算机系统软、硬件的功能分配比例时，应考虑在现有的硬件和器件条件下，如何使系统有高的性能价格比。从降低实现费用来分析，如果一个功能是经常用的基本单元功能，且是属固定不变的功能，才适合于采用硬件实现；而对产量大的计算机系统，增大硬件功能实现的比例才是有利的。我们不能去盲目追求扩大硬件功能实现的比例。

确定软、硬件功能分配时，还应考虑不能过多地限制各种组成和实现技术的采用。也还要考虑如何为编译和操作系统的实现、高级语言的编程等提供更多更好的硬件支持，以便缩短高级语言与机器语言、操作系统与计算机系统结构、程序设计环境与计算机系统结构之间的语义差距。否则，语义差距过大，软件实现的比例就会过大，将会使系统效率过分下降，这不适应当今计算机软件和硬件、器件发展的状况。

2. 计算机系统的设计思路

从计算机多级层次结构的上或下开始设计，有“由上往下”、“由下往上”和“由中间开始”向两边设计等三种不同的设计思路。

“由上往下”设计是先考虑如何满足应用要求，设计好面对使用者那级机器应具有哪些基本功能和特性，再逐级地向下设计各机器级，让每一级都优化于上一级来设计。这是一种专用计算机的设计思路，不适合于一般的通用计算机的设计。因为应用改变，会使软、硬件功能分配很不合理，急剧降低了系统的效率。厂家在设计时，往往也做不到下一级完全优化上一级来设计。

“由下往上”设计是不管应用要求，只根据已有器件、硬件状况，先设计好微程序机器级和传统机器级，再为不同应用配上多种不同的操作系统和编译系统软件，依次设计上面的各个机器级。这是一种通用机的设计思路。但由于软、硬件的脱节，软件得不到为优化软件设计所提供的硬件支持而显得十分繁杂。研制出的硬件机器的性能指标有可能是虚假的。所以，这种设计方法不好，已被淘汰。

好的计算机系统设计应采用从层次的中间开始向两边进行设计。这样，可以避免“由上往下”和“由下往上”地设计所带来的软件、硬件设计脱节的现象。既考虑能拿到的硬件、器件，又考虑应用中可能要用到的算法和数据结构，同时，还要考虑如何为操作系统、编译系统的实现提供更好的硬件支持，先确定好软件和硬件功能分配的界面，然后，再分头并行设计硬件和软件。这样做，不仅有利于缩短系统的研制时间，也有利于硬件和软件设计人员之间的交流协调，使软、硬件之间的功能分配更为合理，系统性能价格比更高。

（四）结构设计要解决好软件的可移植性

1. 实现软件移植的好处和技术

软件的可移植性指的是软件不用修改或只经少量的修改，就可以由一台机器搬到另一台机器上去运行，使得同一套软件可以应用于不同的硬件环境。这样，过去的计算机系统上所用的大量成熟可靠的软件，特别是应用软件，就可以在新的机器上长期使用，而不必重新编写，既大大减少了软件编制的工作量，又能迅速用上新的硬件和器件技术，更新系统，让新系统立即发挥效能，软件设计者也能有精力去开发全新的软件。

实现软件移植的基本技术有统一高级语言、采用系列机、模拟和仿真等。

2. 统一高级语言

统一高级语言是设计一种对各种应用领域都能比较高效的通用高级语言。这样，在结构相同以至完全不同的机器之间，通过配以不同的语言翻译程序就可以实现高级语言应用软件的移植。然而，不同的用途要求高级语言的语法和语义结构差别较大；人们对统一的高级语言应当有什么样基本的结构看法不一；厂家为了便于在机器上高效地翻译，在高级语言中引入了方言；用户为节省程序空间和提高其运行速度，经常在高级语言源程序中嵌入了汇编语言或其他语言的程序；用户的习惯势力，不愿使用新的语言，所有这些因素，使得在近期内难以统一出一种通用的高级语言。对此技术存在的问题采取的对策是，从长远的目标，还是要争取统一出一种通用的高级语言，但近期只能做相对的统一。

3. 采用系列机

采用系列机的技术可以使同一系列内各档机器在汇编语言上实现统一，又能使系列内，发展出多种新的机器。因此，它适合于在结构相同或相近的机器之间，实现汇编语言应用软件和部分系统软件的移植。系列机的设计就是由中间向两边设计的思路。采用系列机可以较好地解决软件设计环境要求相对稳定与硬件、器件和组成等技术飞速发展的矛盾。软件可以继续丰富和积累，又能不断更新器件、硬件和组成，使之短期内就能提供出性能更好、价格更便宜的新机器。

系列机应用软件的兼容，除了从速度和性能上有向上兼容和向下兼容之外，在产品进入市场的前和后上，又有向前兼容和向后兼容之分。系列机必须保证应用软件向后兼容，力争做到向上兼容。因为现在编制的应用程序，以后都能在新的机器上使用，这是系列机软件兼容的根本，至于其后研制的软件完全可以发展，不一定非要向前兼容。从用户观点来讲，当然希望在低档机器上的软件能照搬于高档机器上运行，因此，应力争做到向上兼容。

由于系列机结构变化有限，所以到一定时候会阻碍该系列的发展。对此问题采取的对策应当是，不能只局限于旧系列的发展，在适当时候应推出新的系列结构。

4. 模拟和仿真

模拟和仿真能在结构不同的机器之间实现机器语言程序的移植。用机器语言程序解释另一机器的机器指令实现软件移植的方法称为模拟。模拟方法在机器指令系统差异比较大的时

候，会使程序运行速度严重下降。用微程序直接解释另一机器的机器指令实现软件移植的方法称为仿真。仿真可以提高被移植软件的运行速度，但机器结构差异较大时，很难仿真。因此，对这些问题采取的对策是，在不同系列机器间的软件移植时，将模拟和仿真两种技术结合起来使用。让频繁使用并容易仿真的这部分机器指令采用仿真，以提高速度。而让很少使用，对速度要求不高的，难以仿真的这部分指令及 I/O 操作采用模拟实现。

模拟与仿真的区别是：模拟是用机器语言程序解释，其解释程序存在主存中；仿真是用微程序解释，其解释程序存在控制存储器中。

5．典型题例的分析与解答

〔**题** 1.8〕想在系列机中发展一种新型号机器，你认为下列哪些设想是可以考虑的，哪些则是不行的？为什么？

（1）新增加字符数据类型和若干条字符处理指令，以支持事务处理程序的编译。

（2）为增强中断处理功能，将中断分级由原来的 4 级增加到 5 级，并重新调整中断响应的优先次序。

（3）在 CPU 和主存之间增设 Cache 存储器，以克服因主存访问速率过低而造成的系统性能瓶颈。

（4）为解决计算误差较大，将机器中浮点数的下溢处理方法由原来的恒置“1”法，改为用 ROM 存放下溢处理结果的查表舍入法。

（5）为增加寻址灵活性和减少平均指令字长，将原等长操作码指令改为有 3 类不同码长的扩展操作码；将源操作数寻址方式由操作码指明改成，如 VAX—11 那种设寻址方式位字段指明。

（6）将 CPU 与主存间的数据通路宽度由 16 位扩展成 32 位，以加快主机内部信息的传送。

（7）为减少公用总线的使用冲突，将单总线改为双总线。

（8）把原 0 号通用寄存器改做堆栈指示器。

〔**分析**〕系列机发展新型号机器最主要的是必须保证应用软件的向后兼容。就是说，早先机器上运行的程序在后面的新机器上应能照样运行，只是后面出来的新机器因为增强了它的功能和速度，可以提高其性能。因此，对于那些不属于计算机系统结构，而属于计算机组成和实现的东西，不管是增加、删去，还是修改，都不会影响到汇编语言程序和机器语言程序在系列机上的兼容。但是，对于属于计算机系统结构的那些内容，为保证软件的向后兼容，则只能增加其新的功能或部件，而不能去删掉或更改已有的功能或部件，否则，就不能保证原有程序在新机器上的正确运行。

〔**解答**〕（1）可以。因为它虽然是属计算机系统结构的内容，但它是新增加的数据类型和指令，不会影响到已有指令所写的程序正确运行，只是现在用新增加的指令来写程，会使计算机的性能和效率变得更好。

（2）不可以。中断的分级和中断的响应次序等中断机构，都属于计算机系统结构的内容。中断分级由原来的 4 级增加到 5 级应当还是允许的，关键是重新调整了中断响应的优先次序，这就使原有程序的中断响应次序发生改变，会影响原有程序工作的正确性。

（3）可以。Cache 存储器是属计算机组成，它不会改变原有的系统程序和应用程序，不会影响到它们的正常运行。只是有了 Cache 存储器后，系统的性能有了明显的提高。