

全国计算机自学考试全程过关必备丛书
◆计算机及其应用专业◆

COMPUTER

计算机系统结构

习题与真题解析

(本科)

孙方 等编著



中国水利水电出版社
www.waterpub.com.cn

全国计算机自学考试全程过关必备丛书

计算机系统结构习题与真题解析

孙方 等编著

中国水利水电出版社

内 容 提 要

本书是按照“全国高等教育自学考试指导委员会”制定的“计算机及应用专业（独立本科段）”进行编写的。全书共分三个部分：第一部分介绍了考核知识点与要求，并且给出了配套教材的课后习题部分的相应的分析解答；第二部分对 1994 年以来的自考试卷和作者平时积累的经典题目给出了相应的分析和解答；第三部分是最近几年的自考试卷解析。

本书可以作为计算机及其应用专业自学考试的配套辅导教材，同时也可以作为计算机类专业的教师和学生的辅导和自学参考书。

图书在版编目（CIP）数据

计算机系统结构习题与真题解析/孙方等编著. —北京：中国水利水电出版社，2003

（全国计算机自学考试全程过关必备丛书）

ISBN 7-5084-1808-5

I. 计… II. 孙… III. 计算机体系结构—高等教育—自学考试—解题
IV. TP303-44

中国版本图书馆 CIP 数据核字（2003）第 102490 号

书 名	计算机系统结构习题与真题解析
作 者	孙方 等编著
出版、发行	中国水利水电出版社（北京市三里河路 6 号 100044） 网址：www.waterpub.com.cn E-mail: mchannel@public3.bta.net.cn（万水） sale@waterpub.com.cn 电话：（010）63202266（总机）、68331835（营销中心）、82562819（万水）
经 售	全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京市天竺颖华印刷厂
规 格	787×1092 毫米 16 开本 13 印张 289 千字
版 次	2004 年 1 月第一版 2004 年 1 月北京第一次印刷
印 数	0001—5000 册
定 价	18.00 元

凡购买我社图书，如有缺页、倒页、脱页的，本社营销中心负责调换

版权所有·侵权必究

前 言

计算机系统结构是计算机系统中必不可少的系统软件。《计算机系统结构》这门专业课在整个计算机专业的课程设置中也具有很重要的地位。对于这类计算机专业的基础学科，学生往往觉得能看懂书，却做不好题目。

本书就是针对全国高等教育自学考试指导委员会组编的指定教材《计算机系统结构》（李学干主编，全国高等学校自学考试指定教材，经济科学出版社，2000）出版的一本辅导性教材。学生通过学习这本书，能够对《计算机系统结构》教材的各个部分的重点、难点和主要内容的理解及应用，得到更进一步的提高，最终达到融会贯通的程度。

本书通过对自学考试中的经典题型和近年来几次全国考试的试题的分析，使学生能够更好地把握考试规律，对自学考试的特点也有更深的了解，同时也能启发学生在自学中思考问题，引导学生分析理解问题，从而让学生在自学能力和应试能力方面都有很大的提高。

书中的许多题目是本人在多年教学中积累的，也有不少是历年考试中的经典题目，其中不少是学生在考试中容易出错且容易忽视的题目。本书共分三个部分：第一部分介绍了考核知识点与要求，并且给出了配套教材的课后习题部分的相应的分析解答；第二部分对1994年以来的自考试卷和作者平时积累的经典题目给出了相应的分析和解答；第三部分是最近几年的自考试卷解析。

本书与教材的各个模块内容相互衔接，但又具有相对的独立性；它不脱离教材，但又不是课本的重复，基本做到了不仅讲知识，更强调方法；不仅注意解答，更注意对题目的分析；不仅注意思路的点拨，更注意解题方法的培养和能力的提高。本书仅是一本自学辅导书，学习本课程时应以教材为主，复习和做习题时可参考本书。无论是书上的复习题，还是本书中的试题和模拟试题，读者都应先试着自己解答，然后再翻阅参考解答，这样，可以加深对计算机系统结构的理解和掌握。

自学考试是一项相当复杂的任务，要求学生要具备自强不息、坚持不懈的精神和意志，要有过程、有步骤、有条理、有侧重地进行备考，才能达到所期望的结果。最后我们衷心的希望您在自学考试中一帆风顺，也希望这本书能助您一臂之力。

参与本书编写工作的除封面署名外还有郑晓飞、熊玲菲、张俊林等人。由于时间紧迫，加上编者水平有限，书中难免有不足之处，恳请读者批评指正。

编 者

2003年10月

目 录

前言

第一部分 配套教材习题解答	1
第1章 计算机系统结构	2
第2章 数据表示与指令系统	6
第3章 总线、中断与输入输出系统	17
第4章 存储体系	23
第5章 重叠流水和向量处理机	34
第6章 阵列处理机	46
第7章 多处理机	53
第8章 其他计算机系统结构	62
第二部分 历年试卷仿真题及经典题目解析	67
第1章 计算机系统结构	68
第2章 数据表示与指令系统	79
第3章 总线、中断与输入输出系统	93
第4章 存储体系	111
第5章 重叠流水和向量处理机	127
第6章 阵列处理机	142
第7章 多处理机	150
第8章 其他计算机系统结构	158
第三部分 最新自学考试试卷解析	165
2001年4月高等教育自学考试全国统一命题考试计算机系统结构	166
2001年4月高等教育自学考试全国统一命题考试计算机系统结构参考解答	170
2002年4月高等教育自学考试全国统一命题考试计算机系统结构	176
2002年4月高等教育自学考试全国统一命题考试计算机系统结构参考解答	180
计算机系统结构全真模拟试卷(一)	186
计算机系统结构全真模拟试卷(一) 参考解答	189
计算机系统结构全真模拟试卷(二)	195
计算机系统结构全真模拟试卷(二) 参考解答	198

第一部分 配套教材习题解答

指定教材是广大考生参加自学考试的最重要的学习资料，教材上的习题都紧扣自学考试大纲和各章节的内容。因此，对考生来讲，钻研这些习题的解题方法就成了学习中的重中之重。

本部分按指定教材的章节顺序并结合作者的教学经验和考生经常遇到的问题对每章后的习题进行详细的分析与解答，让考生真正做到知其然和知其所以然。

第 1 章 计算机系统结构

1.1 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、3 和 4 级上一段等效程序各需要运行多长时间？

答：在第 2 级上等效程序需运行： $(N/M)*Ks$ 。在第 3 级上等效程序需运行： $(N/M)*(N/M)*Ks$ 。在第 4 级上等效程序需运行： $(N/M)*(N/M)*(N/M)*Ks$ 。

1.2 硬件和软件在什么意义上是等效的？在什么意义上是不等效的？试举例说明。

答：软件和硬件在逻辑功能上是等效的。原理上，软件的功能可用硬件或固件完成，硬件的功能也可用软件模拟完成，但是实现的性能价格比和难易程度不同。

例如，我们可以为应用语言级、高级语言级、操作系统级提供更多更好的硬件支持，改变硬、器件迅速发展而软件日益复杂、开销过大的状况；或直接用硬件或固件实现，发展高级语言机器或操作系统计算机结构。编译程序、操作系统等许多用机器语言软件子程序实现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；在硬件实现上就简单容易，硬件成本低，灵活性和适应性好。

1.3 试以实例说明计算机系统结构、计算机组成与计算机实现之间的相互关系和影响。

答：计算机系统结构、计算机组成与计算机实现互不相同，但是又互相影响。

计算机系统结构是指多级层次结构中传统机器级的结构，它是软件和硬件的主要界面，是让编制的机器语言程序、汇编语言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。

计算机实现指的是计算机组成的物理实现。

(1) 计算机的系统结构相同，但可采用不同的组成。如 IBM370 系列有 115、125、135、158、168 等由低档到高档的多种型号机器。从汇编语言、机器语言程序设计者看到的概念性结构相同，均是由中央处理机/主存、通道、设备控制器、外设 4 级构成。其中，中央处理机都有相同的机器指令和汇编指令系统，只是指令的分析、执行在低档机上采用顺序进行，在高档机上采用重叠、流水或其他并行处理方式。

(2) 相同的组成可有多不同的实现。如主存器件可用双极型的也可用 MOS 型的；可用 VLSI 单片，也可用多片小规模集成电路组搭。

(3) 计算机的系统结构不同，会使采用的组成技术不同，反之组成也会影响结构。如为实现： $A := B + CD := E * F$ ，可采用面向寄存器的系统结构，也可采用面向主存的三地

址寻址方式的系统结构。要提高运行速度,可让相加与相乘并行,为此这两种结构在组成上都要求设置独立的加法器和乘法器。但对面向寄存器的系统结构还要求寄存器能同时被访问,而对面向主存的三地址寻址方式的系统结构并无此要求,倒是要求能同时形成多个访存操作数地址和能同时访存。又如微程序控制是组成影响结构的典型,通过改变控制存储器中的微程序,就可改变系统的机器指令,改变结构,如果没有组成技术的进步,结构的进展是不可能的。

综上所述,系统结构的设计必须结合应用考虑,为软件和算法的实现提供更多更好的支持,同时要考虑可能采用和准备采用的组成技术。应避免过多地或不合理地限制各种组成、实现技术的采用和发展,尽量做到既能方便地在低档机上用简单便宜的组成实现,又能在高档机上用复杂较贵的组成实现,这样,结构才有生命力;组成设计上面决定于结构,下面受限于实现技术。然而,它可与实现折中权衡。例如,为达到速度要求,可采用简单的组成但却是复杂的实现技术,也可用复杂的组成但却是一般速度的实现技术,前者要求高性能的器件,后者可能造成组成设计的复杂化和更多地采用专用芯片。

组成和实现的权衡取决于性能价格比等因素;计算机系统结构、组成和实现所包含的具体内容随不同时期和不同的计算机系统会有差异。软件的硬化和硬件的软化都反映了这一事实。

VLSI 的发展更使计算机系统结构、组成和实现融为一体,难以分开。

1.4 什么是透明性?对计算机系统结构而言,下列哪些是透明的,哪些是不透明的?

【分析】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;存储器最小编址单位;Cache 存储器。

答:一种本来存在的事物或属性,但从某种角度来看似乎不存在的概念称为透明性。

在一个计算机系统中,低层机器的属性往往对高层机器的程序员是透明的。不同层次级别的程序员所面对的计算机系统的属性是不同的,若某个计算机系统属性不被某个层次的程序员所看到,则称对该层次的程序员是透明的。

存储器的模 m 交叉存取:透明(组成)。

浮点数据表示:不透明(系统结构)。

I/O 系统是采用通道方式还是外围处理机方式:不透明。

数据总线宽度:透明(组成)。

字符行运算指令:不透明(系统结构)。

阵列运算部件:透明(组成)。

通道是采用结合型还是独立型:透明(组成)。

PDP-11 系列的单总线结构:透明(组成)(教材中认为 PDP-11 系列中的单总线结构并不是系统结构所以对于计算机系统结构是透明的)。

访问方式保护:不透明(系统结构)。

程序性中断:不透明(系统结构)。

串行、重叠还是流水控制方式:透明(组成)。

堆栈指令：不透明。

存储最小编址单位：不透明（系统结构）。

Cache 存储器：透明（组成）。

1.5 从机器（汇编）语言程序员看，下列哪些是透明的？

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。

答：从机器（汇编）语言程序员看，也就是从计算机系统结构看的内容，透明的有：指令缓冲器，主存地址寄存器（是计算机中的缓冲技术，是全硬件实现的）；时标发生器，乘法器，先行进位链，移位器（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）。

1.6 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。

答：对系统程序员和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部分无论对系统程序员还是应用程序员都应当是透明的。

对系统程序员透明的有：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。

对应用程序员是透明的有：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；指令缓冲寄存器。

“执行”指令对系统程序员和应用程序员都是不透明的。

1.7 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顺序。

(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 类不同字长扩展操作码。

(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 号通用寄存器改作堆栈指示器。

答：可以考虑的有：(1)、(3)、(4)、(6)、(7)。

不可以考虑的有：(2)、(5)、(8)。

原因分析：系列机设计主要是为了解决软件环境要求相对稳定和硬件、器件技术迅速发展的矛盾。因此要保证在相当长的时期内系列结构基本不变，使系列内各机器从程序设计者的角度去看，系列机都具有相同的机器属性，以实现软件兼容（必须保证向后兼容，力争向上兼容）。这样发展系列机一种新型号机器要保证系列机的指令系统、数据表示及概念性结构的稳定，（2）（5）（8）不予考虑，需要指出的是（1）虽然属于计算机系统结构的内容，但只要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，可使计算机的性能和效率变得更好就可以。

（3）（4）（6）（7）可以改进系统软件的性能，并不会影响高级语言应用软件的兼容。因为数据通路宽度，专用部件的设置和可靠性技术等这些因素对系统结构是透明的。

1.8 并行处理计算机除分布处理、MPP 和机群系统外，有哪四种基本结构？列举它们各自要解决的主要问题。

答：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流水线计算机，阵列处理机，多处理机和数据流计算机四种不同的结构。

流水线计算机主要通过时间重叠，让多个部件在时间上交错重叠地并行执行运算和处理，以实现时间上的并行。它主要应解决拥塞控制、冲突防止、流水线调度等问题。

阵列处理机主要通过资源重复实现空间上的并行。它主要应解决处理单元灵活、规律的互连模式和互联网络设计，数据在存储器中的分布算法等问题。

多处理机主要通过资源共享，让一组计算机在统一的操作系统全盘控制下，实现软件和硬件各级上的相互作用，达到时间和空间上的异步并行。它主要应解决：处理机间互连等硬件结构，进程间的同步和通信，多处理机调度等问题。

数据流计算机没有共享变量的概念，指令执行的顺序只受指令中数据的相关性制约。数据是以表示某一操作数或参数已准备就绪的数据令牌直接在指令之间传递。它主要应解决：研究合适的硬件组织和结构以及高效执行的数据流语言等问题。

1.9 计算机系统的 3T 性能目标是什么？

答：计算机系统的 3T 性能目标是指要求计算机系统能有的 1TFLOPS 的计算能力、1TBYTE 的主存容量、1TBYTE/S 的 I/O 带宽。

第2章 数据表示与指令系统

2.1 数据结构和机器的数据表示之间是什么关系？确定和引入数据表示的基本原则是什么？

答：数据表示是能由硬件直接识别和引入的数据类型。数据结构反映各种数据元素或信息单元之间的结构关系。数据结构要通过软件映像变换成机器所具有的各种数据表示实现，所以数据表示是数据结构的组成元素。不同的数据表示可为数据结构的实现提供不同的支持，表现在实现效率和方便性不同。数据表示和数据结构是软、硬件的界面。

除基本数据表示不可少外，高级数据表示的引入遵循以下原则：

- (1) 看系统的效率是否提高，是否减少了实现时间和存储空间。
- (2) 看引入这种数据表示后，其通用性和利用率是否提高。

2.2 标志符数据表示与描述符数据表示有何区别？描述符数据表示与向量数据表示对向量数据结构所提供的支持有什么不同？

答：标志符数据表示指将数据类型与数据本身直接联系在一起，并合存在一个存储单元，用于描述单个数据的类型等属性，让机器中的每个数据都带类型标志位。

数据描述符指数据的描述与数据分开存放，描述所访问的数据是整块的还是单个的，及访问该数据块或数据元素的地址信息。它具备标志符数据表示的优点，并减少了标志符数据表示所占的空间，为向量和数组结构的实现提供支持。

数据描述符方法优于标志符数据表示，数据的描述与数据分开，描述所访问的数据是整块的还是单个的，及访问该数据块或数据元素的地址信息，减少了标志符数据表示所占的空间。用描述符方法实现阵列数据的索引比用变址方法实现要方便，且便于检查出程序中的阵列越界错误。但它不能解决向量和数组的高速运算问题。而在有向量、数组数据表示的向量处理机，硬件上设置有丰富的向量或阵列运算指令，配有流水或阵列方式处理的高速运算器，不仅能快速形成向量、数组元素地址，更重要的是便于实现把向量各元素成块预取到中央处理机，用一条向量、数组指令流水或同时对整个向量、数组高速处理，如让硬件越界判断与元素运算并行。

2.3 堆栈型机器与通用寄存器型机器的主要区别是什么？堆栈型机器系统结构为程序调用的哪些操作提供了支持？

答：堆栈型机器指的是有堆栈数据表示的机器。它与一般通用寄存器型机器不同。通用寄存器型机器对堆栈数据结构实现的支持是较差的，表现在：

- (1) 堆栈操作的指令少，功能单一。
- (2) 堆栈在存储器内，访问堆栈速度低。
- (3) 堆栈通常只用于保存子程序调用时的返回地址，少量用堆栈实现程序间的参数传递。

堆栈型机器为堆栈数据结构的实现提供有力的支持,表现在:

(1) 有高速寄存器组成的硬件堆栈,并与主存中堆栈区在逻辑上组成整体,使堆栈的访问速度是寄存器的,容量是主存的。

(2) 丰富的堆栈指令可对堆栈中的数据进行各种运算和处理。

(3) 有力地支持高级语言的编译。

(4) 支持子程序的嵌套和递归调用(包括保存子程序的返回地址,保存条件码,保存关键寄存器内容,保存必要的全局型、局部型参数,为子程序开辟存放局部变量和中间结果的工作区)。

2.4 设某机阶值 6 位、尾数 48 位,阶符和数符不在其内,当尾数分别以 2、8、16 为基时,在非负阶、正尾数、规格化数的情况下,求出其最小阶、最大阶、阶的个数、最小尾数值、最大尾数值、可表示的最小值和最大值及可表示的规格化数的总个数。

答:

所求各值 r_m		$2(m'=48)$	$8(m'=16)$	$16(m'=12)$
最小阶	0	0	0	0
最大阶	2^p-1	63	63	63
阶的个数	2^p	64	64	64
最小尾数值	r_m^{-1}	1/2	1/8	1/16
最大尾数值	$1-r_m^{-m'}$	$1-2^{-48}$	$1-2^{-16}$	$1-2^{-12}$
可表示的最小值	r_m^{-1}	1/2	1/8	1/16
可表示的最大值	$r_m^{2^p-1}(1-r_m^{-m'})$	$2^{63}*(1-2^{-48})$	$2^{63}*(1-2^{-16})$	$2^{63}*(1-2^{-12})$
可表示的总个数	$2^p r_m^{m'}(r_m-1)/r_m$	2^{53}	$7*2^{51}$	$15*2^{50}$

2.5 (1) 浮点数系统使用的阶基 $r_p=2$, 阶值位数 $p=2$, 尾数基值 $r_m=10$, 以 r_m 为基的尾数位数 $m'=1$, 按照使用的位数来说, 等价于 $m=4$, 试计算在非负阶、正尾数、规格化情况下的最小尾数值、最大尾数值、最大阶值、可表示的最小值和最大值及可表示数的个数。

(2) 对于 $r_p=2$, $p=2$, $r_m=4$, $m'=2$, 重复以上计算。

答:

所求的各项值	$r_m=10, m'=1, m=4$
最小尾数值	1/10
最大尾数值	$1-10^{-1}$
最大阶值	3
可表示的最小值	1/10
可表示的最大值	$10^3(1-10^{-1})$
可表示的总个数	$2^2 10(10-1)/10$
最小尾数值	1/4

续表

所求的各项值	$r_m=4, m'=2, m=4$
最大尾数值	$1-4^{-2}$
最大阶值	3
可表示的最小值	$1/4$
可表示的最大值	$4^3(1-4^{-2})$
可表示的总个数	$4 \times 16 \times 3/4$

2.6 由 4 位数（其中最低位为下溢处理之附加位）经 ROM 查表舍入法，下溢处理成 3 位结果，设计使下溢处理平均误差接近于零的 ROM 表，列出 ROM 编码地址表与内容的对应关系。

答：用查表舍入法对浮点数尾数进行下溢处理，是用 ROM 或 PAL 存放下溢处理表。通常，下溢处理表的内容安排成当尾数最低 $K-1$ 位全“1”时以截断法设置处理结果，即输出 $K-1$ 位的全“1”，其余情况按舍入法设置下溢处理结果。这样，截断法的负误差可补偿舍入法产生的正误差，使平均误差为零。

ROM 编码地址表与内容的对应关系见下表：

地址	内容
0000	000
0001	001
0010	001
0011	010
0100	010
0101	011
0110	011
0111	100
1000	100
1001	101
1010	101
1011	110
1100	110
1101	111
1110	111
1111	111

2.7 变址寻址和基址寻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个大地址空间中任意 64 个地址之一的寻址机构。

答：变址寻址是对诸如向量、数组等数据块运算的支持，以利于实现程序的循环。基址寻址是对从逻辑地址空间到物理地址空间变换的支持，有利于实现程序的动态再定位。基址寻址和变址寻址同时都会用到。基址寻址将装入程序形成物理地址改成由地址加法器硬件形成，加快了地址变换的速度。

将大地址空间 64 个地址分块，用基址寄存器指出程序所在的块号，用指令 6 位地址码表示该块内 $2^6=64$ 个地址之一，这样基址和变址相结合可以访问大地址空间中任意 64 个地址之一。还有一种方法是规定基地址用程序计数器。程序计数器存放的是当前所执行指令的下一条指令所在主存中的地址（或偏移地址），可以通过使用无条件转移指令来修改 PC 的内容，实现在一个大的地址空间中的访问。

2.8 经统计，某机 14 条指令的使用频度分别为：0.01、0.15、0.12、0.03、0.02、0.04、0.02、0.04、0.01、0.13、0.15、0.14、0.11、0.03。分别求出用等长码、Huffman 编码、只有两种码长的扩展操作码等 3 种编码方式的操作码平均码长。

答：（1）机器共有 14 条指令，若操作码采用等长码表示需要 4 位操作码，平均码长为 4 位（若用 3 位最多只能表示 8 条指令，用 4 位最多的可表示 16 条指令）。

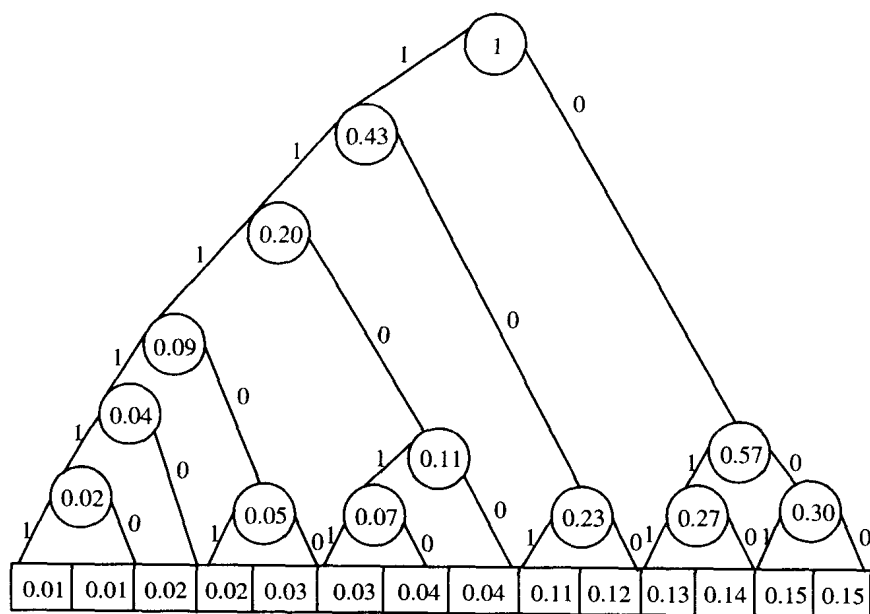
（2）Huffman（哈夫曼）编码：

指令	使用频度
I_1	0.01
I_2	0.15
I_3	0.12
I_4	0.03
I_5	0.02
I_6	0.04
I_7	0.02
I_8	0.04
I_9	0.01
I_{10}	0.13
I_{11}	0.15
I_{12}	0.14
I_{13}	0.11
I_{14}	0.03

①将所有 14 条指令的使用频度由小到大排序：

0.01 0.01 0.02 0.02 0.03 0.03 0.04 0.04 0.11 0.12 0.13 0.14 0.15 0.15

②选择其中最小的两个频度，合并成一个频度，直到全部频度连结完毕形成根结点。



列出各指令的编码和码长:

指令频度 (P_i)	Huffman 编码	码长 (l_i)
0.01	111111	6
0.01	111110	6
0.02	11110	5
0.02	11101	5
0.03	11100	5
0.03	11011	5
0.04	11010	5
0.04	1100	4
0.11	101	3
0.12	100	3
0.13	011	3
0.14	010	3
0.15	001	3
0.15	000	3

$$\text{Huffman 编码平均码长} = \sum_{i=1}^{14} p_i \cdot l_i = 3.38$$

(3) 只有两种码长的扩展操作码的编码:

指令频度 (P_i)	编码	码长 (l_i)
0.01	111111	6
0.01	111110	6

续表

指令频度 (P_i)	编码	码长 (l_i)
0.02	111101	6
0.02	111011	6
0.03	111001	6
0.03	110111	6
0.04	110101	6
0.04	110	3
0.11	101	3
0.12	100	3
0.13	011	3
0.14	010	3
0.15	001	3
0.15	000	3

平均码长

$$\begin{aligned}
 &= 6 \times (0.01 + 0.01 + 0.02 + 0.02 + 0.03 + 0.03 + 0.04) + 3 \times (0.11 + 0.12 + 0.13 + 0.14 + 0.15 + 0.15 + 0.04) \\
 &= 0.96 + 2.52 \\
 &= 3.48
 \end{aligned}$$

2.9 若某机要求有：三地址指令 4 条，单地址指令 255 条，零地址指令 16 条。设指令字长为 12 位，每个地址码长为 3 位。问能否以扩展操作码为其编码？如果其中单地址码为 254 条呢？说明理由。

答：(1) 当单地址指令为 255 条时，不能用扩展码为其编码。

因为指令字长 12 位，每个地址码占 3 位，这样一来

三地址指令最多为： $2^{(12-3-3-3)} = 8$ 条，题目要求三地址指令需 4 条，可有 4 条编码用作扩展码。

单地址指令最多为： $4 \times 2^3 \times 2^3 = 256$ 条，题目要求单地址指令需 255 条，可有 1 条编码用作扩展码。

零地址指令最多为： $1 \times 2^3 = 8$ 条，题目要求单地址指令需 16 条，相互矛盾。也就是说当单地址指令 255 条时，不能用扩展码为其编码。

(2) 当单地址指令为 254 条时，能用扩展码为其编码。

因为指令字长 12 位，每个地址码占 3 位，这样一来

三地址指令最多为： $2^{(12-3-3-3)} = 8$ 条，题目要求三地址指令需 4 条，可有 4 条编码用作扩展码。

单地址指令最多为： $4 \times 2^3 \times 2^3 = 256$ 条，题目要求单地址指令需 254 条，可有 2 条编码用作扩展码。

零地址指令最多为： $2 \times 2^3 = 16$ 条，题目要求单地址指令需 16 条，也就是说当单地址指令 254 条时，能用扩展码为其编码。

2.10 某机指令字长 16 位。设有单地址指令和双地址指令两类。若每个地址字段为 6 位且双地址指令有 X 条, 问单地址指令最多有多少条?

答: 单地址指令最多有 $(16-x) \times 2^6$ 条。

2.11 何谓指令格式的优化? 简要列出包括操作码和地址码两部分的指令格式优化可采用的各种途径和思路。

答: 指令格式的优化指的是如何用最短的位数来表示指令的操作信息和地址信息, 使程序中指令的平均字长最短。

(1) 对操作码的优化, 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。可利用 Huffman 的压缩概念, 进行操作码的 Huffman 编码及扩展操作码编码。

Huffman 压缩概念的基本思想是, 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), 而对出现概率较低的允许用较长的位数 (时间) 来表示 (处理), 就会导致表示 (处理) 的平均位数 (时间) 的缩短。

①统计每种指令在程序中出现的概率 (使用频度)。

②利用 Huffman 算法, 构造哈夫曼树。将指令按出现的频度由小到大排序, 每次选择其中最小的二个频度合并成一个频度为它们二者之和的新结点, 再按该频度大小插到余下未参加结合的频度值中, 如此继续进行, 直至全部频度结合完毕形成根结点为止。之后, 对每个结点向下延伸, 分出二个分支, 分别用一位代码的“0”或“1”来表示。从根结点开始, 沿线到达各频度指令所经过的代码序列就构成该频度指令的 Huffman 编码。全 Huffman 编码是最优化的编码但不便于译码。

③结合用一般的二进制编码, 得到扩展操作码编码。

(2) 对地址码的优化。

①可通过采用多种寻址、地址制、地址形式、地址码长度和指令字长, 与可变长操作码的优化表示相结合, 可以构成冗余度尽可能少的指令字。

②采用 0、1、2、3 等多种地址制, 以增强指令功能且让常用的短操作与多个地址字段相结合。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。

③在同种地址制中再采用多种地址形式。

④在维持指令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。

2.12 某模型机 9 条指令使用频度为:

ADD (加)	30%	SUB (减)	24%	JOM (按负转移)	6%
STO (存)	7%	JMP (转移)	7%	SHR (右移)	2%
CIL (循环指令)	3%	CLA (清加)	20%	STP (停机)	1%

要求有两种指令字长, 都按双操作数指令格式编排, 采用扩展操作码并限制只能有两种操作码码长。设该机有若干通用寄存器, 主存为 16 位宽, 按字节编址, 采用按整数边界存储, 任何指令都在一个主存周期中取得, 短指令为寄存器-寄存器型, 长指令为寄存器-主存型, 主存地址应能变址寻址。