第一章重点

- 对计算机系统结构哪些是透明的
- Flynn分类法（掌握到会画图水平）
- 计算机系统设计的主要方法（三种）
- 解决软件兼容的方法（三种）
- 并行性开发的途径（三种）

第一章内容要点

- 系统结构的基本概念
- 计算机系统结构的分类
- 软件对系统结构的影响
- 并行性开发的途径

第一章主要内容（1）

- 系统结构的基本概念
  - 系统结构：程序设计者能看到的计算机的属性，即软硬件的交界面。
  - 翻译：先用转换程序将高一级机器极上的程序整个地变换成低一级机器级上可运行的等效程序，然后再在低一级机器上去实现的技术。
  - 解释：在低一级机器级上用它的一串语句或指令来仿真高一级机器上的一条语句或指令的功能，通过对高一级机器语言程序中的每条语句或指令逐条解释来实现。
  - 透明性：本来存在的事物或属性，从某种角度看似乎不存在。

第一章主要内容（2）

- 计算机系统结构的分类
  - Flynn分类法

    单指令流单数据流SISD;

    单指令流多数据流SIMD;

    多指令流单数据流MISD;

    多指令流多数据流MIMD;

四个图示

第一章主要内容（3）

- 软件对系统结构的影响
  - 系列机：一个厂家内生产的具有相同的系统结构，但具有不同的组成和实现的一系列不同型号的机器。
  - 兼容机：不同厂商生产的具有相同系统结构的计算机。
  - 兼容性分类：向上（下）兼容、向前（后）兼容
  - 解决软件兼容的方法：系列机、模拟和仿真、统一的高级语言。

第一章主要内容（4）

- 并行性开发的途径
  - 时间重叠
  - 资源重复
  - 资源共享

第二章重点

- 数据结构和数据表示的区别与联系
- 浮点数尾数下溢的处理方法
- 自定义数据表示的类型
  - 带标志符的数据表示（指令数据不等长解决办法）
  - 数据描述符
- 程序定位的方法
- 前缀性编码
- 程序和数据存放在同一存储器中的优点

第二章难点

- 各种编码的平均码长
- 构造Huffman树
- 扩展编码法
- 计算机指令系统优化的两个方向

第二章主要内容（1）

- 数据结构和数据表示的区别与联系

硬件	软件
数据表示	数据结构

第二章主要内容（2）

- 尾数下溢处理主要方法
  - 截断法
  - 下舍上入法
  - 恒置"1"法
  - 查表舍入法
  - R*舍入法

要求：各种方法的优缺点比较！

第二章主要内容（3）

- 自定义数据表示
  - 数据描述符和标志符的区别
    1. 标志符只作用于一个数据，而描述符作用于一组数据
    2. 标志符与数值存放在同一个地址单元中，描述符单独占用一个地址单元
  - 带标志符的数据表示中指令和数据不等长的解决办法
    1. 仍混存
    2. 增长指令字，扩充其功能
    3. 分开存放

第二章主要内容（4）

- 程序定位的方法
  - 直接定位方式

    直接使用实际的主存物理地址来编写和编译程序
  - 静态再定位方式

    利用一个专门设计的装入程序，在把程序装入主存的过程中，程序中的逻辑地址转换成物理地址
  - 动态再定位

    程序装入主存时，指令和地址不作任何修改，只把其主存的起始地址存入与该程序对应的基址寄存器中

第二章主要内容（5）

- 前缀性编码
  - 前缀性：任何一个代码都不能成为另一个代码的头部。
  - Huffman编码属于前缀性编码

第二章主要内容（6）

- 程序和数据存放在同一个存储器中的优缺点

  指令和数据混存在同一存储器中，在执行过程中，指令可以当操作数一样被修改。这样一来，由于共用同一套外围电路而节省了硬件；也由于指令和数据不加区分同等对待而简化了存储管理；有由于指令可以修改带来程序可以修改的灵活性等许多好处。但也因此造成了程序不易编制和调试，软件不易进行故障诊断和排错，程序可靠性正确性难以得到保证等许多不良后果。程序可以修改也不利于实现程序的可再入性和程序的递归调用，不利于指令和数据的并行存取及组成上采用重叠流水来提高速度。由于指令和数据字长要求不尽相同，也就不利于优化存储器的字长和充分利用存储空间。

典型习题解答

- 第一章习题由同学记录或者自己完成
- 第二章11.设计IBM370那样有基地址寄存器的机器的另一种办法是，每条指令不用现在的基地址寄存器地址（4位）加位移量（12位）共16位作为地址码，而是让每条指令都有一个24位的直接地址。针对下面两种情况评价一下这个方法的优、缺点：

  （1）数据集中于有限几块，但这些块分布在整个存储空间

  （2）数据均匀分布在整个地址空间中

第11题答案

答：对于第一种情况，由于数据集中在几块中，那么采用基址寄存器加位移量的方式比较好。在这种情况下，由于数据被分在几块，在寻址时，只要修改基址寄存器的值，而不需要修改指令便可寻址，若让指令拥有一个24位的直接地址，则每次块间转移时，地址全要修改，比上一种方法麻烦。对于第二种情况，由于数据均匀分布在整个地址空间中，用第二种方法比较好。因为分布较平均，所以两种寻址修改的频率差不多，由于后一种方法有较大的寻址空间2^24，所以采用第二种方法。我认为，IBM370设计者认为前者可能性要大一些。数据都是随程序一块块装入的，相对而言便由每个程序组成一个个块，用基址寄存器就方便一些。

第12题

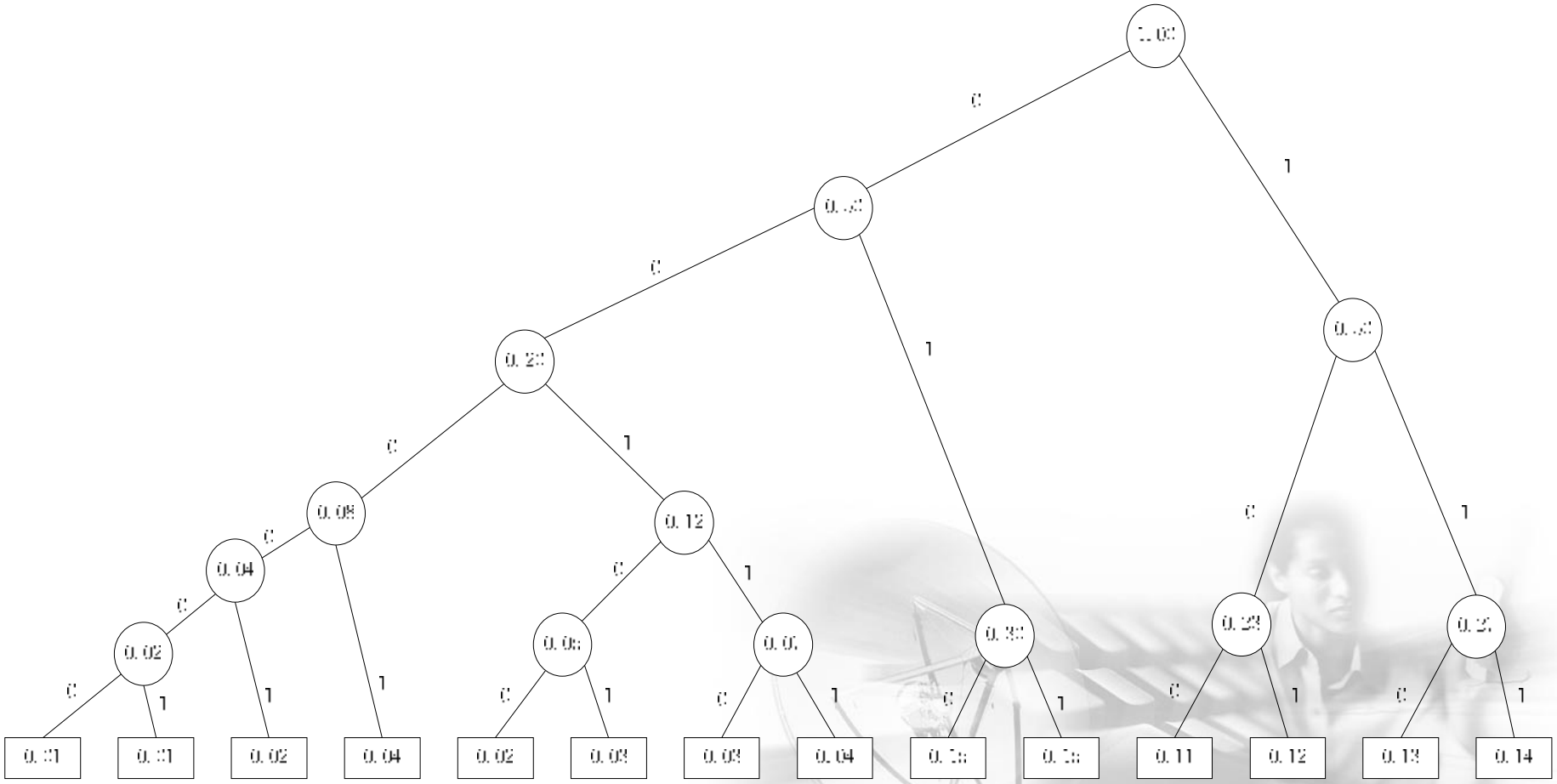
- 经统计，某机14条指令使用频度分别为：0.01、0.15、0.12、0.03、0.02、0.04、0.02、0.04、0.01、0.13、0.15、0.14、0.11、0.03。分别求出等长的二进制编码、Huffman编码、只能有两种码长的扩展操作码编码等3种方式的操作码平均码长。

第12题答案

- 答：因为共有14条指令，若用等长的二进制编码至少需要4位才能表示出14种指令（2^3<14<2^4).所以等长的二进制编码的平均码长为4！

采用Huffman编码：先构造Huffman树！

第12题答案


1.00
0
1
0.40
0.60
0
1
0.20
0
1
0.08
0.12
0
1
0.04
0
1
0.02
0
1
0.01
0.01
0.02
0.04
0.02
0.03
0.03
0.04
0.23
0.37
0
1
0.13
0.24
0
1
0.11
0.12
0.13
0.14

第12题答案

Huffman编码的平均码长:

H=（0.11+0.12+0.13+0.14＋0.15＋0.15）×3+0.04×4+（0.02＋0.02+0.03+0.03+0.04）×5＋（0.01+0.01）×6＝3.38

扩展编码：

采用2－6法扩展结果如下：

H=（0.15＋0.15＋0.14）×2＋[1－(0.15+0.15+0.14)]×6＝4.24

采用3－6法扩展结果如下：

H＝（0.15+0.15+0.14+0.13+0.12+0.11+0.04）×3＋（0.04＋0.03+0.03+0.02+0.02+0.01+0.01）×6＝3.48

根据编码规则，采用**3－6**法扩展操作码。

不需继续判断，因为等长的二进制编码才4位！！

第14题

若机器要求有如下形式的指令：

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？如果其中单地址指令254条呢？说明理由！

第14题答案

答：如果单地址指令255条，不能以扩展码为其编码，如果单地址指令254条，可以用扩展码为其编码。因为：

0 0 0 0	0 0 1 1	0 1 0 1	地址	地址	地址

1 . . . 1	0 0 1 1	0 0 1 1	0 0 1 1	0 0 1 1	0 0 1 1	0 0 1 1	0 0 1 1	0 1 0 1	地址

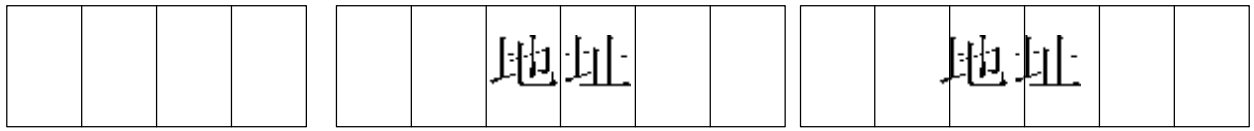
第15题

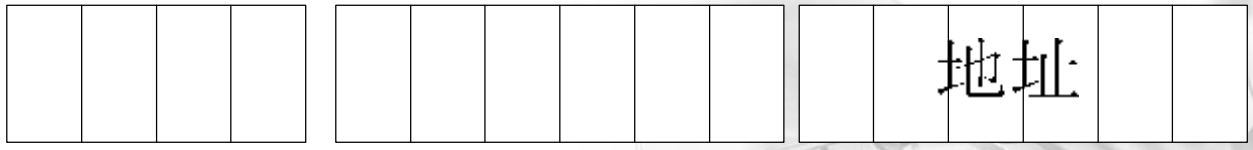
某机的指令字长16位，设有单地址指令和双地址指令两类，若每个地址字段均为6位，且双地址指令有X条，问单地址指令最多可以有多少条？

第15题答案

答： (2^4-X）*2^6

因为





第18题

设计一个在某方面功能比IBM370强，又希望现有IBM370程序能兼容的系列机。一些人认为可让全部RX指令（即R_1,X_2,B_2,D_2型）都改成采用间接寻址；另一些人则认为只增加两条IBM370没有的新指令，能间接从存储器取代码到寄存器的指令和把寄存器的代码间接地存到存储器的指令。你认为哪种方案比较好？为什么？

第18题答案

答：增加好，因为修改之后不健全！

第19题

为什么不少计算机采用由8个或更多个寄存器构成的通用寄存器组？它们为什么会使编译程序更复杂？你能设想什么技术的发展会使得不必采用通用寄存器组？

第19题答案

答：

① 采用通用寄存器组，可减少指令访存时间，加快指令执行速度，在设置一些如基址、变址寄存器后，更加快了速度，而且在重入性等方面也有了提高。

② 在一些如mov reg,m 的指令中，要判断原来的内容是否有用，若有用还要进行保护，这要编译程序来做，因此增加了它的难度。在一些语句上，也需要在编译上要进行优化，如赋值语句等，要考虑中间结果、变量等的存、放都增加了编译的难度。

③ 我认为要替代通用寄存器技术，就必须提高存储技术与堆栈技术。提高存储器速度便可减少如寄存器间址，mov reg m 的技术。采用堆栈技术，在程序调用及一些运算上将更容易、更快，且减小了编译的难度。

学生练习题

第20题

第24题（自己总结）

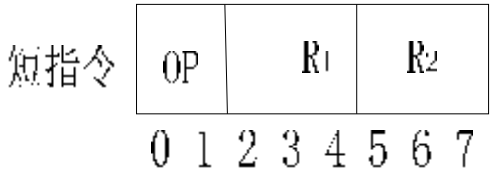
第20题答案

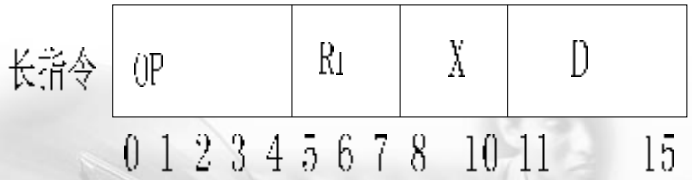
① 全Huffman编码：H＝2.61

② 优化指令操作码形式：H＝2*(0.3+0.24+0.2)+5*(0.07+0.07+0.06+0.03+0.02+0.01)=2.78(位）

③ 可使用8个可编址的通用寄存器。

④ 指令字格式





⑤ 因在访存指令中，偏移量为5位，则最大相对位移量为2^5-1个字节