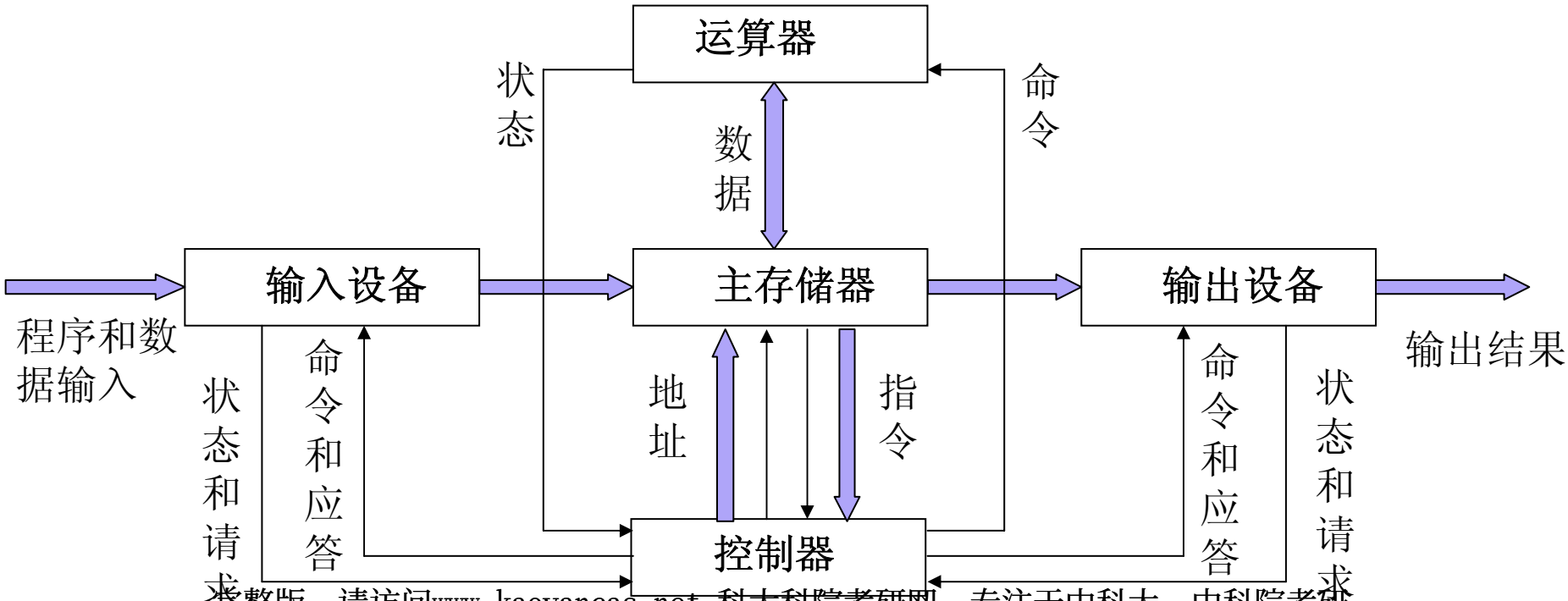
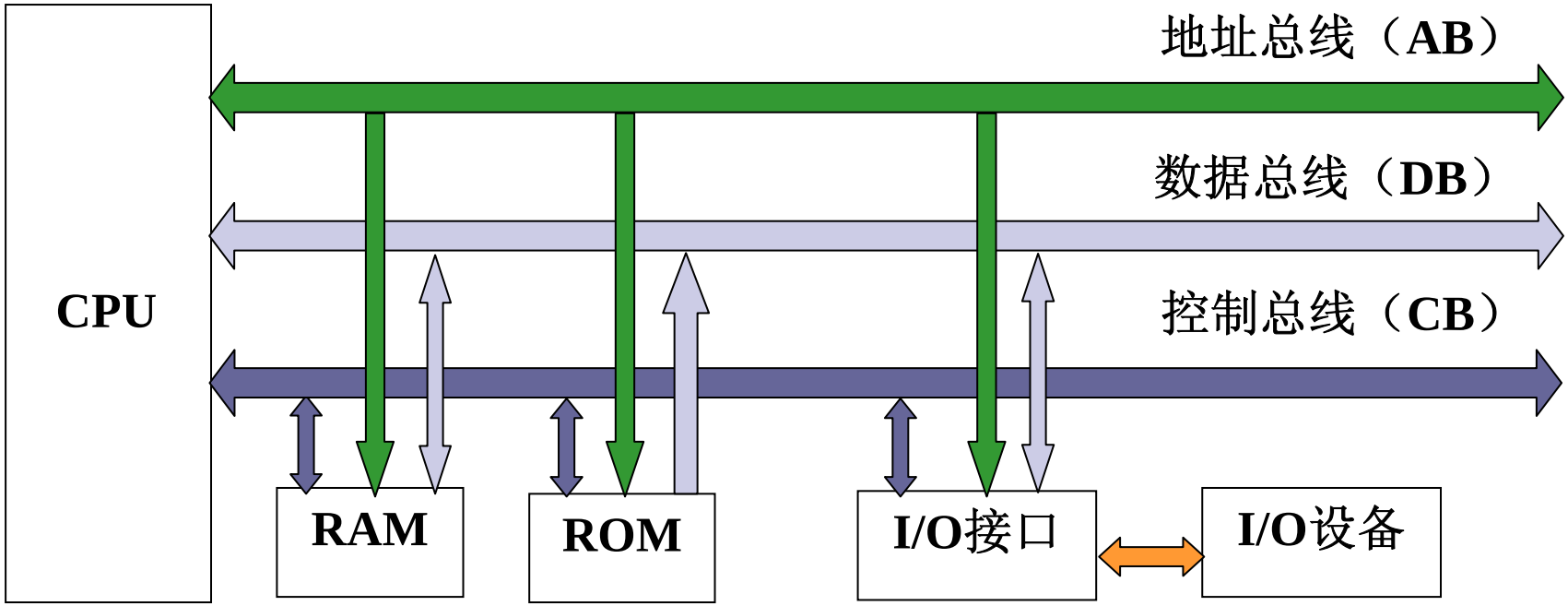
1-2、微型计算机系统

一、微型计算机

- 计算机基本结构：冯.诺依曼结构，早期以CPU为核心，现演化为以存储器为核心。

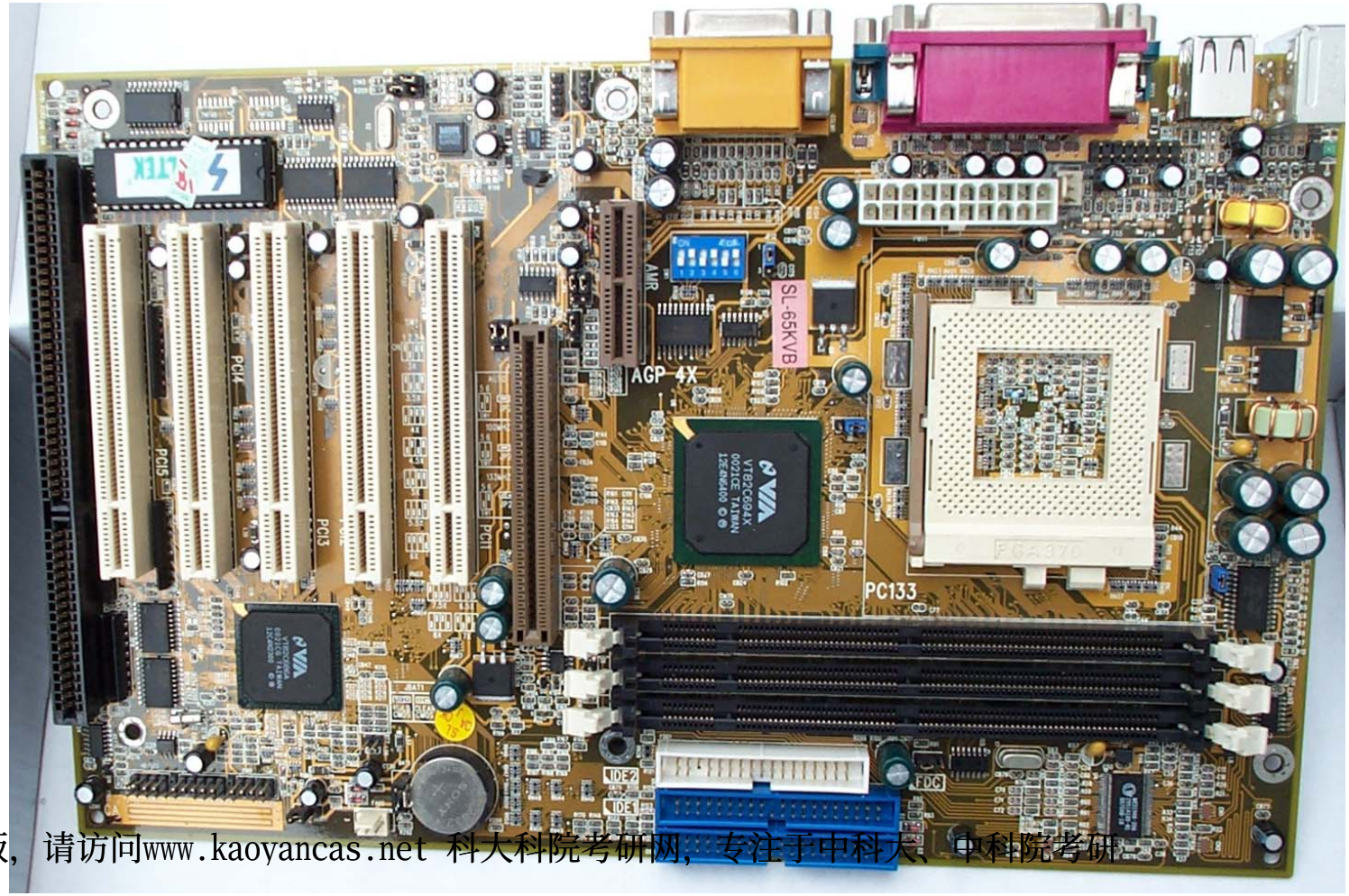


微型计算机的硬件结构



- 地址由CPU或DMA控制器产生，一般是单向的。
- 数据总线是双向的。
- 控制信号有些是单向，有些是双向，传送CPU发出的控制信息、或者传送部件产生的状态信息。

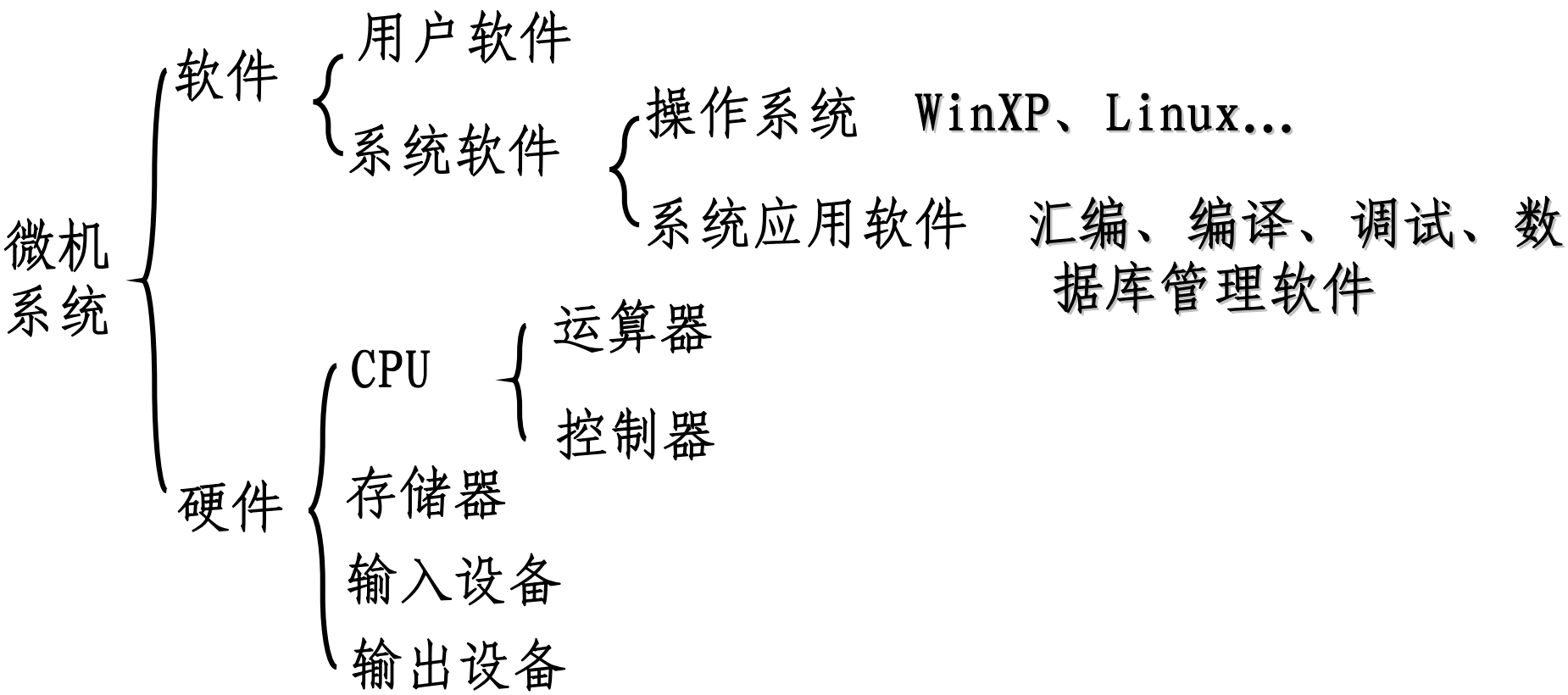
- 主机：包含了除输入/输出设备以外的所有部件，是一个能独立工作的系统，所以有时也将主机称为微型计算机。
- 主板：



各部分的功能

- CPU
  - 1）取指和指令译码
  - 2）算术逻辑运算
  - 3）传送数据
  - 4）程序流向控制
- 存储器
  - 分为RAM和ROM，存储数据和程序
- IO接口
  - 将外部设备与CPU或存储器互联
- 总线
  - CPU、存储器和IO接口之间传输地址、数据和状态信息的公共通道

微机系统的基本组成



微处理器

- CPU—中央处理单元，由算术逻辑单元ALU、累加器、寄存器（Reg）组、指令指针寄存器IP、段寄存器、时序和控制部件以及内部总线组成。其中：
  - ALU和累加器负责进行各种算术和逻辑运算
  - 寄存器组存放正在处理的数据和运算结果
  - IP就是程序计数器
  - 段寄存器用于内存的分段管理
  - 时序控制部件负责指挥和控制运算器等各个部件工作
  - 内部总线在各个部件之间传送数据

二、存储器（1）

- 主存储器（简称为主存或内存）
  - 存放当前正在使用或经常使用的数据，可以和**CPU**直接交换信息，是计算机的工作存储器。
  - 程序和数据事先都必须存放在主存中，工作时，计算机所执行的指令和数据都从主存中取出，处理结果一般也都存放在主存中。
  - 主存的存取速度快而容量相对较小。 。
  - 内存现一般为半导体存储器，有：
    - RAM：SRAM、DRAM和NVRAM
    - ROM：PROM、EPROM、EEPROM、Flash
  - **Cache**

二、存储器（2）

- 辅助存储器
  - 存储容量大，但存取速度较慢。
  - 辅助存储器中存放着计算机系统中几乎所有的信息。计算机运行时，辅助存储器的信息需要先调入主存后，才能被**CPU**使用。
  - 目前微机系统中使用的辅助存储器，主要是磁表面存储器（磁盘、磁带）和光盘存储器。
  - 接口有：**IDE/ATA**、**SCSI**（包括**U160**和**U320**等）、**IBM SSA**、**FC**（**Fiber Channel**）和**FC-AL**等。
  - 教材图**1-5**是微机系统的存储体系示意图。

存储器的组织（1）

- 为了正确地对存储器进行读写访问，通常以字节（或以字或双字）为单位将存储器划分为一个个存储单元，并依次对每一个存储单元赋予一个序号（从零开始的无符号整数），该序号称为存储单元的地址。
- 存储单元中存放的数据或指令称为存储单元的内容。地址是识别存储器中不同存储单元的唯一标识。

存储器的组织（2）

- 不同的微机系统中的存储体系结构也不同。
  - 在80x86系统中，16位微机配置两个存储体，分别连接数据总线D15～D8和D7～D0；
  - 32位微机配置4个存储体，分别连接数据总线D31～D24、D23～D16、D15～D8和D7～D0，一次可以传送32位数据；
  - Pentium以上微机配置8个存储体，分别连接64位数据总线。
- 图1-6是8086~80486物理存储体系的示意图。

三、I/O接口

- 负责实现CPU或存储器与外设之间的信息交换。
- 外设的差异性：　　多种多样的外设
- 接口的功能

  速度匹配　　格式转换　　电平匹配

  时序控制　　中断管理等
- 主要接口芯片：
  - 不可编程：**74xx373、74xx244、74xx245等等；**
  - 可编程：**8255A、8251/8250/16650、8253/8254、8259A、8237A等等。**

四、总线（1）

- 总线定义：各部件之间传送信息的公共通道。各个功能部件是通过总线互连构成系统。
- 总线标准：国际标准、组织标准、企业标准、临时或约定标准
- 总线标准的四个特性

1）物理特性　　2）功能特性

3）电气特性　　4）时间特性

四、总线（2）

- 总线分类

1. 元件内部总线：微处理器内部的总线
2. 元件级总线（片总线）：元件彼此互连，构成模块。如**Motel**总线
3. 系统总线：模块之间互连构成系统。如**ISA**、**EISA**、**PCI**、**Multibus**、**VME**、**CrossBar**等。
4. 外总线：系统之间互连。如**IEEE-488**、**USB**、**EIA-RS232/422/423**

总线结构

- 单总线：CPU访问内存和访问IO使用同一条总线。对总线的控制较为简单，但是总线分时工作造成传输性能受限。（见教材图1-7）
- 双总线：
  - 面向CPU的双总线。CPU通过两条总线分别连接内存和IO。缺点是IO与内存交换数据要通过CPU转发。（见教材图1-8）
  - 面向主存的双总线。所有设备挂在总线上，主存与CPU之间增加一条高速存储总线。（见教材图1-9）

五、微机性能指标（Benchmarks）

- 常见指标：主频、字长、内存容量、运算速度等
- 主频快，不一定性能高，性能还取决于字长、总线带宽、**Cache**、结构等诸多因素。例如**3.4GHz**的**Pentium 4**不如**2.8GHz Pentium Xeon**
- 衡量计算机系统指标要根据用途考虑
  - **1990**年代前后使用的**MIPS**，一台满配的**DEC VAX 780**运算能力被确认为一个标准**MIPS**。
  - **HPC**应用：**SPEC int**，**SPEC flp**等基准测试

——**SPEC**（**Standard Performane Evaluation Corporation**，标准性能评估公司）

性能评测指标举例

- **TPC(Transaction Processing Council，事务处理委员会）指标（参见www.tpc.org)**
  - **tpm-C** **OLTP（联机事务处理）应用**
  - **tpm-H** **Decision Support for Ad Hoc Queries**
  - **tpm-APP** **应用服务器和Web服务能力**
  - **tpm-D** **OLAP（联机分析）数据仓库应用、BI**
  - **… …**
- 应该根据应用领域选择测试指标，例如：统计分析和数据库应用经常选择SAAP公司的基准测试程序，但是对于视频处理和多媒体应用却没有太大的意义。

1-3 计算机数据格式

- 计算机中的数据分为两类：

（1）数：用来直接表示量的多少，有大小之分，能够进行加减乘除等运算。

（2）码：通常指代码或编码，在计算机中用来描述某种信息。

数的表示

- 任何一种数制表示的数都可以写成按位权展开的多项式之和。

$$N = d_{n-1}b^{n-1} + d_{n-2}b^{n-2} + d_{n-3}b^{n-3} + \cdots\cdots d_{-m}b^{-m}$$

式中：**n** — 整数的总位数。

m — 小数的总位数。

$d_{下标}$ — 表示该位的数码。

b — 表示进位制的基数。

$b^{上标}$ — 表示该位的位权。

数制和表示法

计数制	基数	数　码	进位关系
二进制	2	0、1	逢二进一
八进制	8	0、1、2、3、4、5、6、7	逢八进一
十进制	10	0、1、2、3、4、5、6、7、8、9	逢十进一
十六进制	16	0、1、2、3、4、5、6、7、8、9	
		A、B、C、D、E、F	逢十六进一

计数制的书写规则

（1）在数字后面加写相应的英文字母作为标识。

如：二进制数的100可写成100B，十六进制数100可写成100H

（2）在括号外面加数字下标。

如：$(1011)_2$ 表示二进制数的1011，$(2DF2)_{16}$ 表示十六进制数的2DF2

数制转换（1）（参见教材P15～P16）

（1）十进制整数转换为二进制整数（除2倒取余）

- 采用基数2连续去除该十进制整数，直至商等于“0”为止，然后逆序排列余数。

（2）十进制小数转化为二进制小数（乘2顺取整）

- 连续用基数2去乘以该十进制小数，直至乘积的小数部分等于“0”，然后顺序排列每次乘积的整数部分。

（3）十进制整数转换为八进制或十六进制整数（除8或者除16倒取余）

- 采用基数8或基数16连续去除该十进制整数，直至商等于“0”为止，然后逆序排列所得到的余数。

数制转换（2）

（4）十进制小数转换为八进制或十六进制小数（乘8或者乘16顺取整）

- 连续用基数8或基数16去乘以该十进制小数，直至乘积的小数部分等于“0”，然后顺序排列每次乘积的整数部分。

（5）二、八、十六进制数转换为十进制数

- 用其各位所对应的系数，按“位权展开求和”的方法就可以得到。其基数分别为2、8、16。

数制转换（3）

（6）二进制数转换为八进制数

- 从小数点开始分别向左或向右，将每3位二进制数分成1组，不足3位数的补0，然后将每组用1位八进制数表示。

（7）八进制数转换为二进制数

- 将每位八进制数用3位二进制数表示。

（8）二进制数转换为十六进制数

- 从小数点开始分别向左或向右，将每4位二进制数分成1组，不足4位的补0，然后将每组用一位十六进制数表示。

（9）十六进制数转换为二进制数

- 将每位十六进制数用4位二进制数表示。

【例1】将十进制整数105转换为二进制整数，采用“除2倒取余”的方法，过程如下：

2 | 105	
2 | 52	余数为1
2 | 26	余数为0
2 | 13	余数为0
2 | 6	余数为1
2 | 3	余数为0
2 | 1	余数为1
0	余数为1

所以，105＝1101001B

【例2】将十进制小数0.8125转换为二进制小数，采用“乘2顺取整”的方法，过程如下：

0.8125×2＝1.625　　取整数位1

0.625×2＝1.25　　取整数位1

0.25×2＝0.5　　取整数位0

0.5×2＝1.0　　取整数位1

所以，0.8125＝（0.1101）B

如果出现乘积的小数部分一直不为“0”，则可以根据精度的要求截取一定的位数即可。

【例3】将十进制整数2347转换为十六进制整数，采用“除16倒取余”的方法，过程如下：

16 | 2347

16 | 146　　余数为11（十六进制数为B）

16 | 9　　余数为2

0　　余数为9

所以，2347＝92BH

二、计算机中数值数据的表示

- 基本概念
  - 在计算机内部表示二进制数的方法称为数值编码，把一个数及其符号在机器中的表示加以数值化，称为机器数。
  - 机器数所代表的数称为数的真值。
- 表示一个机器数，应考虑以下三个因素：
  - 1. 机器数的范围
  - 2. 机器数的符号
  - 3. 机器数中小数点的位置

机器数的范围

- 字长为8位，无符号整数的最大值是：

  11111111B＝255D

  此时机器数的范围是**0~255**。

- 字长为**16**位，无符号整数的最大值是：
  - **1111111111111111B＝FFFFH＝65535D**
  - 此时机器数的范围是**0~65535**。

机器数的符号

- 在算术运算中，数据是有正有负的，将这类数据称为带符号数。
- 为了在计算机中正确地表示带符号数，通常规定每个字长的最高位为符号位，并用**0**表示正数，用**1**表示负数。

机器数中小数点的位置

- 小数点位置的两种约定：

1. 小数点的位置固定不变，这时的机器数称为“定点数”。由于定点位置不同，一般又分为两种情况。
   - 对于整数，小数点在最低位的右边，称为定点整数。
   - 对于纯小数，小数点在符号位之后，称为定点小数。
2. 小数点的位置可以浮动，这时的机器数称为“浮点数”。

原码

- 正数的符号位为0，负数的符号位为1，其它位按照一般的方法来表示数的绝对值。用这样的表示方法得到的就是数的原码。
- 【例】当机器字长为8位二进制数时：
  - X＝＋1011011　　[X]原码＝01011011
  - Y＝－1011011　　[Y]原码＝11011011
  - [＋1]原码＝00000001　　[－1]原码＝10000001
  - [＋127]原码＝01111111　　[－127]原码＝11111111
- 原码表示的整数范围是：
  - $-(2^{n-1}-1)\sim+(2^{n-1}-1)$，其中n为机器字长。
  - 8位二进制原码表示的整数范围是－127~＋127。

反码

- 对于带符号数，正数的反码与其原码相同，负数的反码为其原码除符号位以外的各位按位取反。
- 【例】当机器字长为8位二进制数时：
  - Y＝－1011011　[Y]原码＝11011011
    [Y]反码＝10100100
  - [＋1]反码＝00000001　　[－1]反码＝11111110
  - [＋127]反码＝01111111　　[－127]反码＝10000000
- 负数的反码与原码有很大区别，反码通常用作求补码过程中的中间形式。
- 反码表示的整数范围与原码相同。

补码

- 正数的补码与其原码相同，负数的补码为其反码在最低位加1。

【例】（1）X＝＋1011011　（2） Y＝－1011011

（1）根据定义有：　[X]原码＝01011011　}　X为正数，补码与原码相同

[X]补码＝01011011

（2）根据定义有：　[Y]原码＝11011011

[Y]反码＝10100100

[Y]补码＝10100101

- 补码表示的整数范围是：

-2^{n-1}～＋$(2^{n-1}-1)$，其中n为机器字长。

如：　8位二进制补码表示的整数范围是－128～＋127。

补码与真值之间的转换

- 正数补码的真值等于补码本身；负数补码转换为其真值时，先按位求反，再末位加1，即可得到该负数补码对应的真值。

【例】[X]补码＝01011001B，[Y]补码＝11011001B，分别求其真值X。

（1）[X]补码代表的数是正数，其真值：

$$X = +1011001B$$
$$= +(1\times2^6+1\times2^4+1\times2^3+1\times2^0)$$
$$= +(64+16+8+1) = +(89)D$$

（2）[Y]补码代表的数是负数，则真值：

$$Y = -([1011001]\text{求反}+1)B$$
$$= -(0100110+1)B$$
$$= -(0100111)B$$
$$= -(1\times2^5+1\times2^2+1\times2^1+1\times2^0)$$
$$= -(32+4+2+1) = -(39)D$$

二—十进制编码——BCD码

- **Binary-Coded Decimal**，又称为“二—十进制编码”，解决用二进制数表示十进数的问题。常用的是**8421**码。
- **BCD**码的两种形式：

1．压缩**BCD**码

一个字节表示**2**位十进制数。例如：十进制数**89D**用压缩**BCD**码表示为：**10001001B** 。

2．非压缩**BCD**码

一个字节表示**1**位十进制数，低**4**位来表示**0**～**9**，高**4**位为**0**。例如：十进制数**89D**用非压缩**BCD**码表示为：

00001000 00001001B

ASCII码－美国信息交换标准代码

- **ASCII －American Standard Code for Information Interchange**
- 用于给西文字符编码，包括英文字母的大小写、数字、专用字符、控制字符等。
- 由**7**位二进制数组合而成，可以表示**128**种字符。
- 在**ASCII**码中，按其作用可分为：
  - **34**个控制字符；
  - **10**个阿拉伯数字
  - **52**个英文大小写字母；
  - **32**个专用符号

扩展ASCII码－ANSI码

- ANSI（American National Standard Institute）
- 使用8位二进制数来表示一个字符，可以对256个字符、符号等进行编码。
- ANSI开始的128个字符的编码和ASCII定义的一样，只是在最高位上加个0。
- 除了表示ASCII码中的128个字符外，ANSI编码还有128个符号可以表示，如版权符、英镑符、外国语言字符等。
- 还有Unicode、EBCDIC 和各种汉字编码等，这些与本课程关系不大，感兴趣的同学可以自学。

数据类型

- 字节**Byte**，字**Word**，双字**Double Word**
- 最基本存储单元是字节
- 在**x86**系统中，低字节放置低地址内存单元，高字节存放在高地址单元。
- 汇编语言伪指令**DB**、**DW**和**DD**等分别用于定义字节、字和双字等（例题**1-10**）

（在**16**位或**16**位以上**x86**微机系统中，为方便读写，一般将一个字的低**8**位（**L**）存放在偶地址（低地址单元），高**8**位（**H**）存放在奇地址（高地址单元）。但是在图**1-10**和例**1-10**中，***DA2****两个字的存储位置？*）

浮点数

- 在科学计算中，为了能表示特大或特小的数，采用“浮点数”或称“科学表示法”表示实数。
- 浮点数三要素：尾数M（mantissa）、指数E（exponent，又称阶码）和基数B（base）。
- 任意一个数N可以表示成下列形式的浮点数：

$$N = M \times B^{E}$$

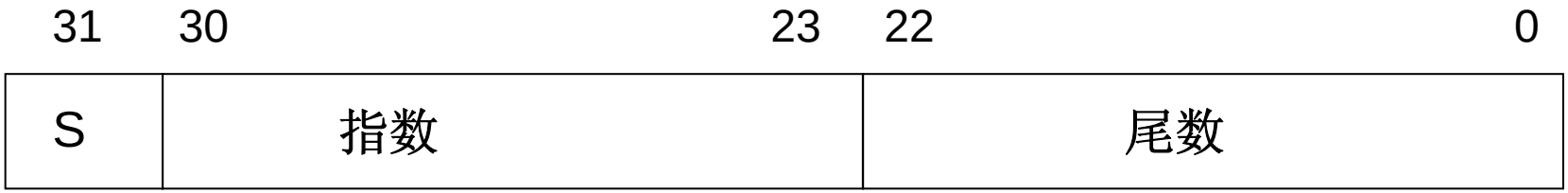
- 基数B常有2、8、16等数值。
- 在计算机中，M采用定点小数，E采用有符号整数
- M和E决定了浮点数的精度，E指明小数点在B进制数据中的位置，E和B决定了浮点数的表示范围，超过最大或最小范围都会造成溢出。

浮点数在计算机中的表示方法

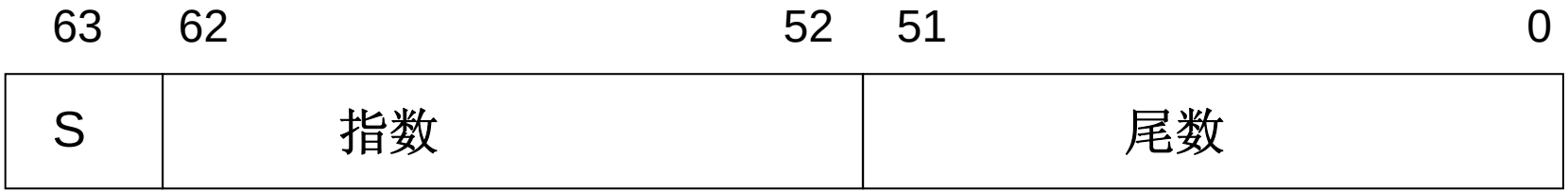
- 为了便于软件移植和数据交换，浮点数在计算机中的表示方法必须统一，为此IEEE在1985年制定了IEEE-754标准。该标准要点有：
  - 基数采用2
  - 有符号数阶码E用移码表示，单精度的偏移量为127，双精度偏移量为1023（比一般移码的偏移量少1）。
  - 尾数M用规格化原码表示
  - 规格化方法：尾数必须是大于1并且小于2。规格化后的尾数（小数点前）的最高位总是1，该位1不存储。

单精度和双精度浮点数

- 单精度浮点数： 4字节
  - 1个符号位S，8位阶码，23/24位尾数（定点小数）



- 双精度浮点数： 8字节
  - 1个符号位S，11位阶码，52/53位尾数（定点小数）



习题：

- P20：
  - 5
  - 7
  - 8
  - 13
  - 14
  - 15
  - 16