



河南大學
Henan University

汇编语言与接口技术

——第 1 章 微型计算机概述

主讲教师：舒高峰

电子邮箱：gaofeng.shu@henu.edu.cn

联系电话：13161693313

目录

- 01 微型计算机概念
- 02 计算机系统组成
- 03 汇编语言及其特点
- 04 计算机数据表示

01 | 基本概念-计算机分类

计算机按性能指标可分为巨型机 (supercomputer)、大型机 (mainframe)、中型机 (midrange)、小型机 (minicomputer) 和**微型机 (microcomputer)**。



巨型机-银河Ⅲ

高速度和大容量



大型机-IBM z15

速度快、应用于军事技术科研领域



中型机-FACOM M170

介于大型计算机和小型计算机之间

01 | 基本概念-计算机分类

计算机按性能指标可分为巨型机 (supercomputer)、大型机 (mainframe)、中型机 (midrange)、小型机 (minicomputer) 和**微型机 (microcomputer)**。



小型机-台式机

结构简单、造价低、性能价格比突出



微型机-笔记本、平板电脑

体积小、重量轻、价格低

从**系统结构**和**基本工作原理**上说，**微型计算机**和其他几类计算机并没有本质上的区别

01 | 基本概念-微型机

微型机

- 由大规模集成电路组成的、体积较小的电子计算机
- 以**微处理器 (Microprocessor)** 为基础，配以**内存储器及输入输出 (I/O) 接口电路和系统总线**所组成的计算机。

微型机特点

- 体积小、重量轻
- 价格低廉
- 可靠性高、结构灵活
- 应用面广



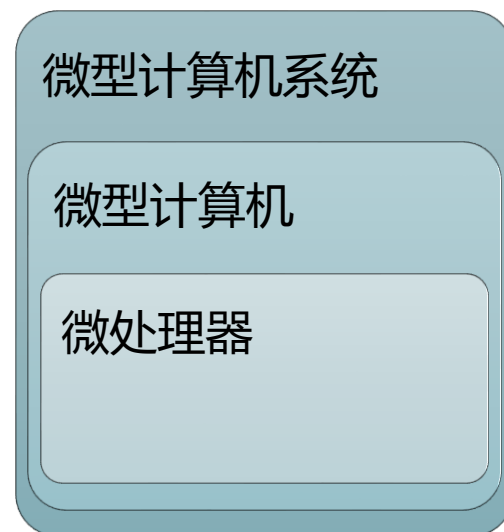
01 | 基本概念-微处理器

微处理器

- 由一片或几片大规模集成电路组成的，具有**运算器**和**控制器**功能的中央处理单元 (Central Processing Unit, CPU)
- 按字长可分为 4 位、8 位、16 位、32 位和 64 位微处理器

三者关系

- 微处理器=CPU
- **微机**=**微处理器**+存储器+I/O 接口+系统总线
- **微机系统**=**微机**+外围设备+电源+系统软件



01 | 基本概念-微处理器的发展

微处理器的产生与发展

	时间	代表产品名称	CPU 字长	集成度 (晶体管/片)	功能
第一代	71~73	Intel 4004	4	1200	家用电器、计算器和简单的控制设备
第二代	73~78	Intel 8080	8	4900	教学、实验系统和工业控制、智能仪器
第三代	78~84	Intel 8086	16	29000	数据处理和管理系统
第四代	85~92	Intel 80386	32	27.5万~100万	个人计算机
第五代	93~04	Intel 80586 Pentium (奔腾)	32/64	310万~4200万	个人计算机、视频音像处理
第六代	05~	Intel 酷睿	32/64	2.91亿	个人计算机、消费电子产品、个性化定制笔记本电脑

字长变长

集成度高

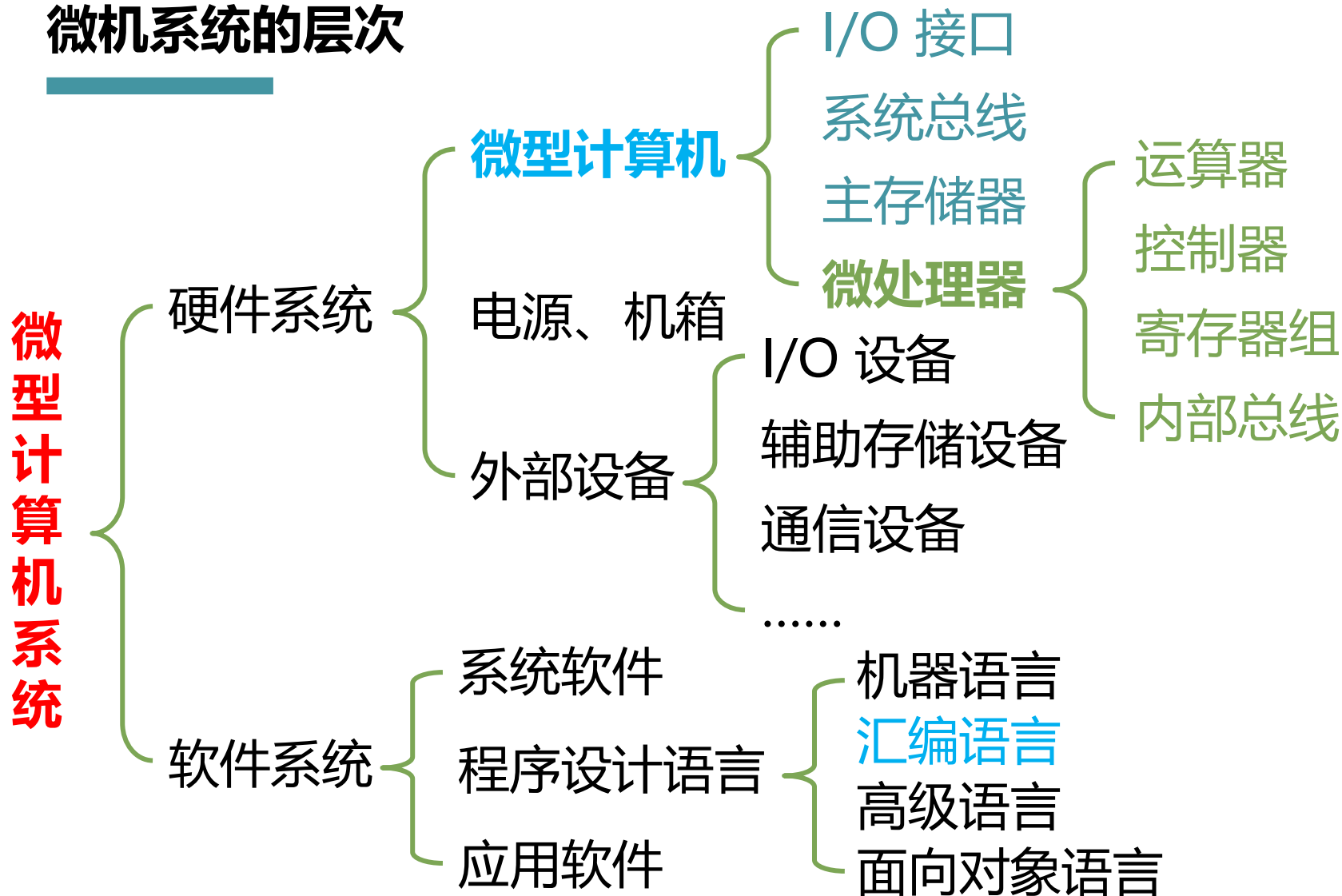
功能多样

目录

- 01 微型计算机概念
- 02 计算机系统组成
- 03 汇编语言及其特点
- 04 计算机数据表示

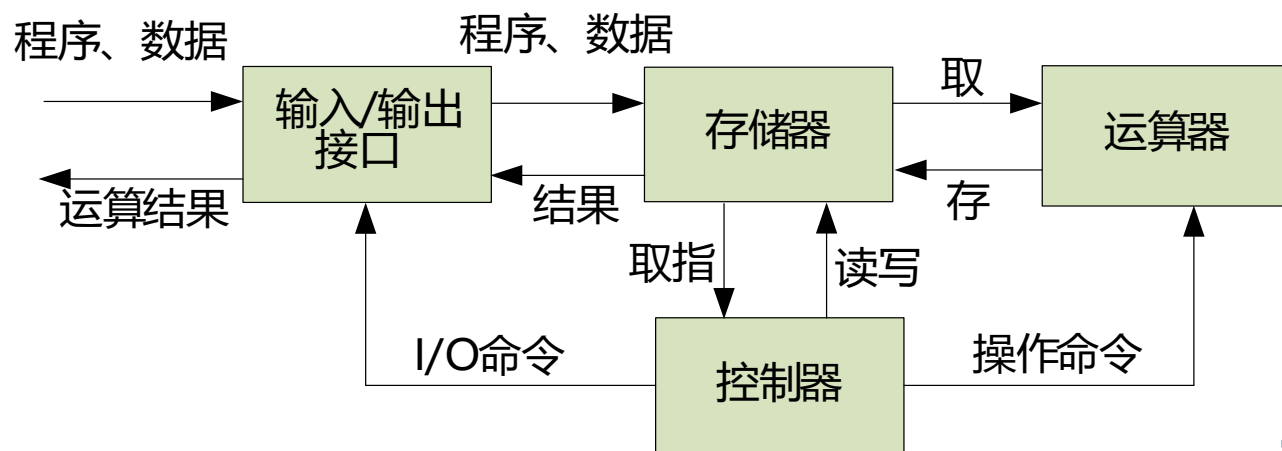
02 | 系统组成-微机系统层次

微机系统的层次



02 | 系统组成-计算机的一般结构

一般计算机结构



冯·诺依曼 (1903~1957)
John von Neumann

部件功能

- **控制器**：发布操作命令、控制信号等
- **运算器**：进行算术和逻辑运算
- **存储器**：存储程序、数据、中间结果和运算结果
- **I/O 接口**：与外部存储器传输程序、数据、运算结果等

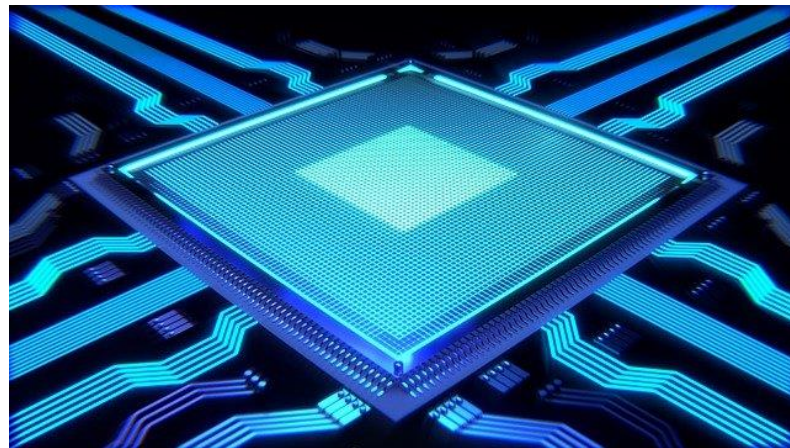
02 | 系统组成-CPU

内部结构

- 运算器+控制器+寄存器组+译码器+总线接口部件

主要功能

- 执行机器指令
- 进行算术和逻辑运算
- 暂存少量数据
- 访问存储器、外设数据
-



02 | 系统组成-CPU功能1-执行机器指令

机器指令 (Machine Instructions)

- CPU 能够直接识别并执行的**二进制编码**
 - CPU 一条接一条地**依次执行**存放在存储器中的机器指令

1010000100000000000100000

指令集

- CPU 能够执行的全部机器指令
 - **CPU 决定机器指令**，不同种类 CPU，其指令集往往不相同

指令分类

- 数据传送指令、算术逻辑运算指令、转移指令、处理器控制指令、其他指令等

02 | 系统组成-CPU功能2-暂存少量数据

指令作用

- 对数据进行各种运算或者处理

寄存器组 (Registers)

- 存放运算数据和运算结果，还可以给出存储单元地址
- 利用寄存器存放运算数据和运算结果，效率最高
- CPU 内可以用于存放运算数据和运算结果的寄存器数量很有限

02 | 系统组成-CPU功能3-访问存储器

指令作用

- 对数据进行各种运算或者处理

存储器 (Memory)

- CPU 能够直接访问的**计算机系统的物理内存**
- **目标程序**在存储器中，待处理的**数据**也在存储器中

系统总线 (System Bus)

- 信息从一个或多个源部件传送到一个或多个目的部件的一组**传输线**
- 系统总线包含有三种不同功能的总线，即**数据总线**、**地址总线**和**控制总线**

02 | 系统组成-CPU功能4-总结

指令作用

- 对数据进行各种运算或者处理

代码示例

```
int x = 1;
int y = 2;
void cf12( void )
{
    y = x*x + 3;
    return;
}
```

C

```
1. MOV    AX, VARX
2. IMUL   AX, AX
3. ADD    AX, 3
4. MOV    VARX, AX
5. RET
```

存储单元

存储单元

汇编

执行机器指令 (1, 2, 3, 4, 5)
进行算术和逻辑运算 (2, 3)
暂存少量数据 (1, 2, 3)
访问存储器 (1, 4)

下列不是 CPU 的主要功能的是

- ☒ A 存储目标程序
- ☐ B 暂存少量数据
- ☐ C 执行机器指令
- ☐ D 访问内存单元

提交

目录

- 01 微型计算机概念
- 02 计算机系统组成
- 03 汇编语言及其特点
- 04 计算机数据表示

03 | 汇编语言-计算机软件系统

软件系统

- 微型计算机为了方便用户使用和充分发挥微机硬件效能所必备的**各种程序的总称**
- 软件系统包括系统软件、**程序设计语言**、应用软件

程序设计语言

- 用来编写程序的语言，是**人和计算机之间交换信息所用的一种工具**，又称编程环境。
- 程序设计语言通常可分为三类：机器语言、**汇编语言**、高级语言

03 | 汇编语言 - 三类编程语言

机器语言 (Machine Language)

- 用二进制代码表示的计算机能直接识别和执行的一种
机器指令的集合

汇编语言 (Assembly Language)

- 用助记符或其他标号代替机器指令的程序设计语言
 - 需要汇编程序解译执行

高级语言 (High-level Programming Language)

- 一种对计算机细节具有很强抽象性的高度封装的编程语言
 - 需要编译程序翻译执行

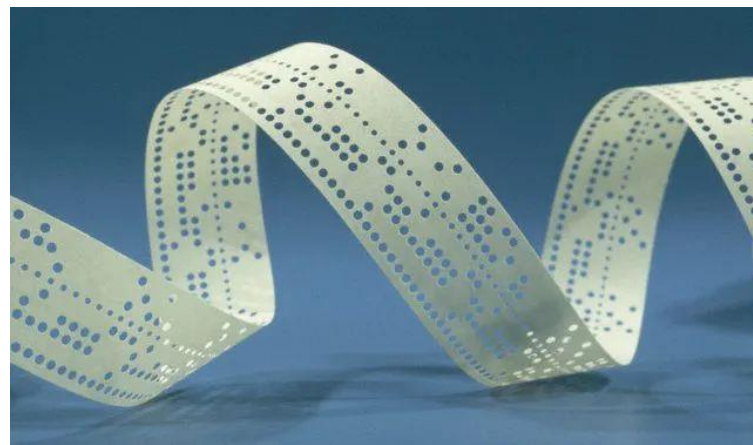
03 | 汇编语言-机器语言

机器指令

- CPU能直接识别并执行
- 由 0、1 二进制代码形式表示，一条指令为计算机的一个基本操作
- 每条指令包括**操作码**和**操作数**两部分

机器语言的特点

- 直接作用于硬件的指令，**执行效率高**
- 指令不直观，编写、调试非常麻烦，**程序可读性不强**



03 | 汇编语言-机器语言举例

示例

- 应用 8086 CPU完成运算: $S = 768 + 12288 - 1280$

机器码

```
1011000000000000000000000011
00000101000000000000110000
00101101000000000000000101
```

出错的机器码

```
1011000000000000000000000011
00000101000000000000110000
00010110100000000000000101
```



03 | 汇编语言-汇编语言相关概念

汇编指令

- 用助记符描述的机器指令
- 汇编指令与机器指令一一对应

汇编语言指令系统

- 包括汇编指令、伪指令、宏指令三大类指令

汇编语言程序 (源程序)

- 使用汇编语言指令系统中的指令编写的应用程序

汇编程序 (翻译程序)

- 将汇编语言程序翻译成机器语言程序的系统软件
 - 常用的汇编程序为 MASM (宏汇编程序)

03 | 汇编语言-汇编和汇编程序

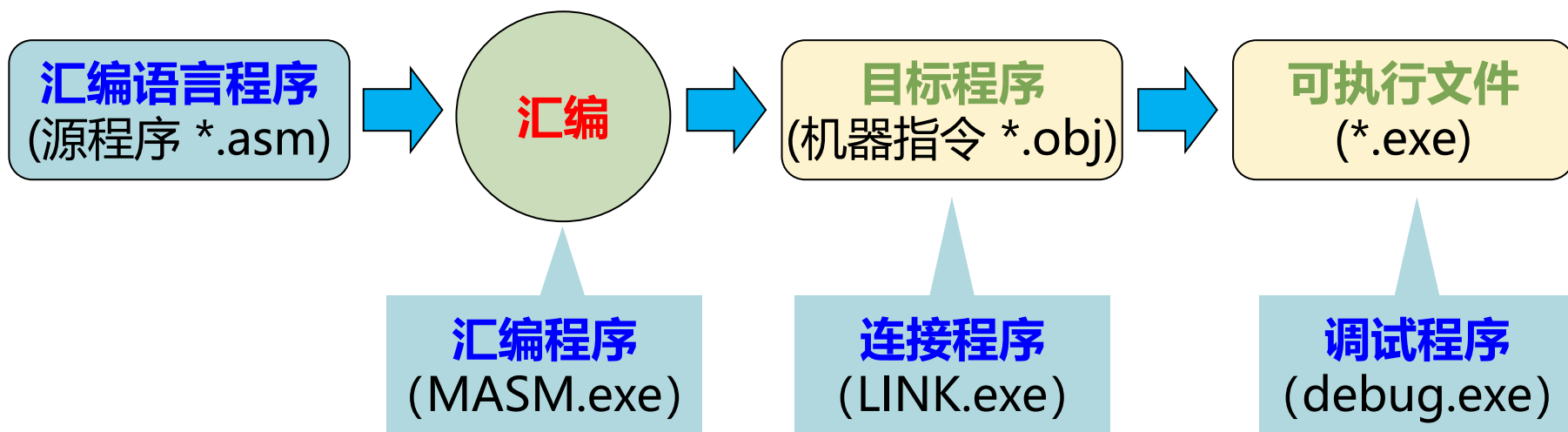
汇编

- 把汇编源程序翻译成目标程序的过程称为汇编

汇编程序

- 把完成汇编工作的工具或程序叫做汇编程序（汇编器）

汇编、源程序、汇编程序的关系



03 | 汇编语言-汇编指令举例

机器指令

- 1000100111011000 (89D8)

实际操作

- 寄存器 BX 的内容送到 AX 中

汇编指令

- `MOV AX, BX`
- 这种写法与人类语言接近，便于阅读和记忆

03 | 汇编语言-汇编语言优缺点

优点

- 执行效率高，程序的可读性较机器语言强

缺点

- 与机器相关，可移植性较差
- 汇编指令功能不强，程序编写复杂，且 Debug 调试麻烦

应用领域

- 执行时间/存储容量有较高要求
- 软件要直接和有效控制硬件
- 需要提高大型软件效率
- 没有合适的高级语言

03 | 汇编语言 - 高级语言

高级语言

- 用与**自然语言相近**的符号描述指令的语言
- 基本脱离了硬件系统，更容易学习和掌握。例如：
C++、Delphi、Java、C#

“翻译”

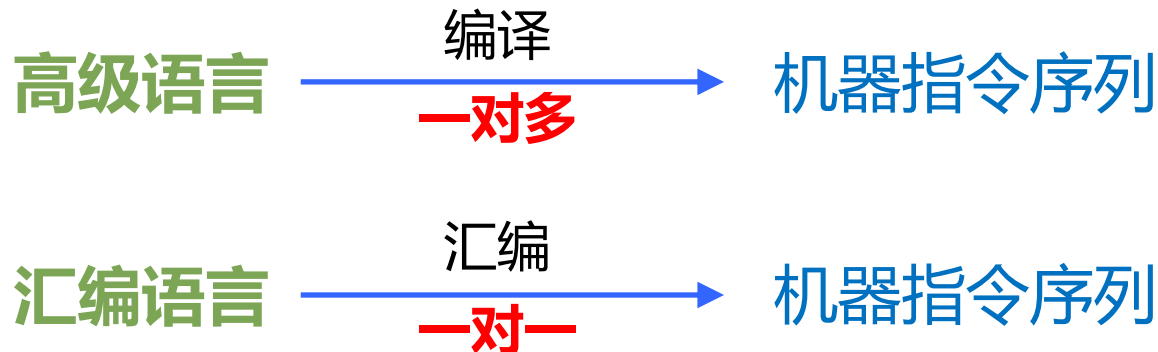
- **高级语言编写的程序不能被计算机识别**，必须“翻译”成机器语言才可执行，常用两种转换方法：
 - 解释程序：一边执行一边“翻译”；
 - 编译程序：执行之前一次性“翻译”。

03 | 汇编语言 - 编程语言的发展

编程语言的发展

- 机器语言 → 汇编语言 → 高级语言

与机器指令的关系



03 | 汇编语言 - 编程语言的发展

编程语言的发展

```
count = i+3;
```

```
mov ax, [2000h]  
add ax, 3  
mov [2002h], ax
```

```
a10020  
050300  
a30220
```

高级语言



汇编语言



机器语言

机器 (处理器)

03 | 汇编语言-要明确的概念

机器内部的数据形式

- 补码、ASCII码

机器内部的数据保存位置

- CPU内部的寄存器、主存中的存储单元

汇编语言中的数据类型

- 字节型、字型、双字型、字符串类型

一个数值的多种含义

- 操作码编码、二进制补码、ASCII码、.....

目录

- 01 微型计算机概念
- 02 计算机系统组成
- 03 汇编语言及其特点
- 04 计算机数据表示**

04 | 数据表示-数据分类

进制

- 十进制 (Decimal)、二进制 (Binary)、十六进制 (Hexadecimal)

数据格式

- 真值、机器数

有无符号

- 无符号数、有符号数

小数点位置

- 定点数、浮点数

字符表示

- ASCII 码、BCD 码 (8421 码)

04 数据表示-数据类型

位 (二进制位, binary digit (bit))

- 1 位二进制数

字节 (Byte)

- 8 位二进制位构成 1 个字节

字 (Word)、双字 (Doubleword)、四字 (Quadword)

- 一个字为 16 位 (即 2 个字节)
- 双字、四字长度依次位字的 2、4 倍

字符串

- 由若干个字节组成一个字符串

1 四字 = 2 双字 = 4 字 = 8 字节 = 64 位

04 | 数据表示-机器数

广义机器数

- 固定长度的**二进制编码**
- 可以表示**数值、字符、指令**.....

狭义机器数

- 计算机内部表示的**二进制数值**
 - 数值正负号也以 0、1 表示
 - 数的大小受机器字长的限制。一般以字节（8位）为单位，一般为 8 位、16 位、32 位等

机器数编码

- 原码、反码、**补码**等

04 数据表示-机器数编码

原码 (Sign-Magnitude Form)

- 符号位：0 — 正，1 — 负
- 数值位：与数据真值相同

反码 (Ones' Complement)

- 正数：与原码相同
- 负数：符号位 — 1；数值位 — **真值数值各位取反**

补码 (Two's Complement)

- 正数：与原码相同；
- 负数：符号位 — 1；数值位 — **真值数值各位取反，末位加 1**

真值	+5	-5
原码	0 0101	1 0101
反码	0 0101	1 1010
补码	0 0101	1 1011

补码的补码就是原码

$(-11)_{10} = (\textcolor{red}{1}1011)_2$ 的补码是多少?

- ☐ A $\textcolor{red}{1}1011$
- ☐ B $\textcolor{red}{1}0100$
- ☐ C $\textcolor{red}{0}0100$
- ☒ D $\textcolor{red}{1}0101$

提交

04 数据表示-补码的常用表示方法

十六进制表示

- 4 位二进制转换为 1 位十六进制数据
 - 如：0101B → 05H、1011B → 0BH
- 习惯上说 16 位数据，指的是二进制的位数
 - 如：20H 为 8 位数据，1234H 为 16 位数据

-1

- 表示：数据所有位为1
 - 8 位——0FFH、16 位——0FFFFH

	0000	0000
-	0000	0001
	1111	1111

负的最小值

- 表示：最高位为 1，其他位为 0
 - 8 位——80H、16 位——8000H

-127	1111 1111	原码
	1000 0001	补码
-128	1000 0000	补码

04 | 数据表示-有符号数位数扩展

扩展原因

- 两个不同位数的编码运算，需要符号位与数值位分别对齐
- 用位数少的数据为位数多的寄存器或存储单元赋值

扩展原则

- 保持数据性质和数值大小不变

扩展方法

- 在符号位与数值位之间添加 0 或者 1

04 数据表示-有符号数位数扩展方法

原码扩展 (0 扩展)

- 无论符号位为 0 或 1，原码扩展都做补 0 操作

补码扩展

- 若符号位为 0，则补 0；若符号位为 1，则补 1

示例

- 原码 0011 0111 → 0000 0000 0011 0111
- 补码 0011 0111 → 0000 0000 0011 0111
- 原码 1011 1100 → 1000 0000 0011 1100
- 补码 1011 1100 → 1111 1111 1011 1100

04 数据表示-机器数的运算 (补码加减法)

基本公式 (以定点整数为例)

- $[x \pm y]_{\text{补}} = [x]_{\text{补}} + [\pm y]_{\text{补}} \pmod{2^{n+1}}$

定点数补码加减法运算

- 符号位和数值位可同等处理
- 只要结果不溢出，将结果按 2^{n+1} 取模，即为本次运算结果。

示例

$$(-3) + (+2) = -1$$

$$(-3) - (+2) = (-3) + (-2) = -5$$

$$\begin{array}{r} 1101 \\ +) 0010 \\ \hline 1111 \end{array}$$

$$\begin{array}{r} 1101 \\ +) 1110 \\ \hline 11011 \end{array}$$

04 数据表示-字符数据1-ASCII码

ASCII 码

- 美国信息交换标准代码 (American Standard Code for Information Interchange, ASCII) 是一种国际上通用的, 7 位的字符二进制编码 (第 8 位为0)
- 键盘上可输入的字符都具有一个对应的ASCII码, 包括可见字符和控制字符。

常用的 ASCII 码 (牢记)

- 数字字符
- 大小写字母字符
- 常用控制字符

十进制	十六进制	对应字符	相关字符
10	0A	换行	
13	0D	回车	
32	20	空格	
48	30	0	1~9
65	41	A	B~Z
97	61	a	b~z

04 | 数据表示-字符数据2-BCD码

BCD 码

- BCD码 (Binary-Coded Decimal) 使用**二进制编码表示十进制数据**的常用编码方法
- 用 4 位二进制数表示一位 10 进制数值 (压缩 BCD 码)
- 从高到低各位的权值分别位 8, 4, 2, 1, 故 BCD 码又称为 **8421 码**
 - 其中 10 种有效编码, 6 种无效编码 (1010~1111)

主要用途

- 表示字符含义的数字, 如电话号码、邮政编码等
- 表示**十进制的数值**, 如十进制数值 20 表示为 20H

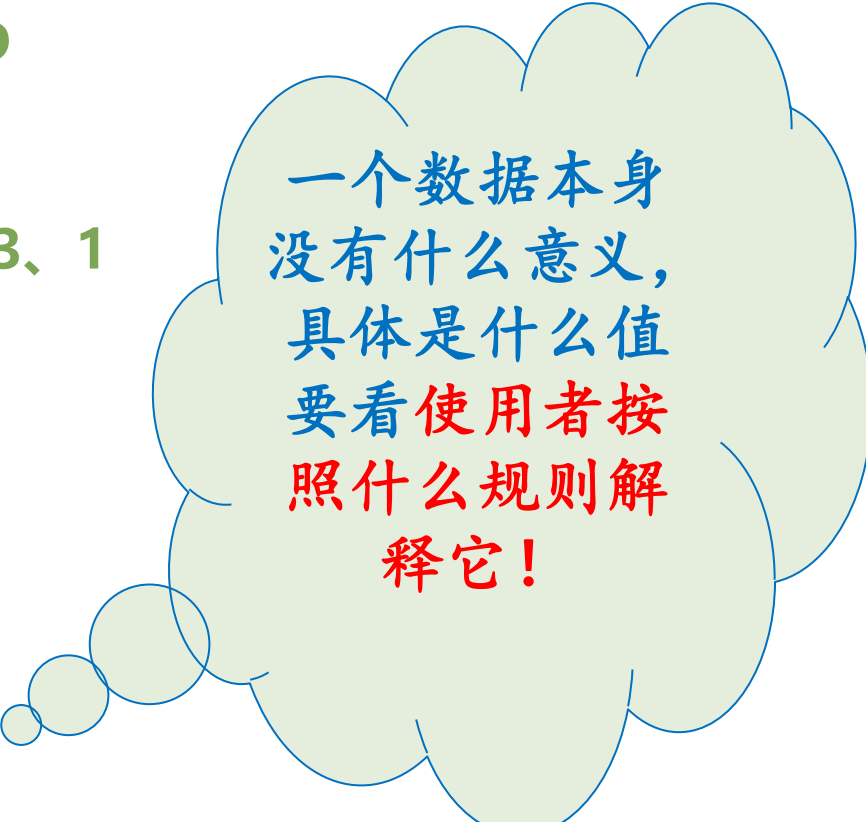
04 数据表示 - 一个数据的不同含义

31H

- 数值: $3 \times 16 + 1 = 49D$
- ASCII 码: 字符 '1'
- 压缩 BCD 码: 两个十进制数据 3、1
- 指令操作码: XOR

0FFH

- 数值
 - 无符号数: 255
 - 有符号数: -127 (原码)
-1 (补码)
- 指令操作码: JMP



一个数据本身
没有什么意义，
具体是什么值
要看使用者按
照什么规则解
释它！

04 | 数据表示-数据的存储

存放位置

- 以二进制形式表示的数据和代码存放在存储器 (内存) 之中

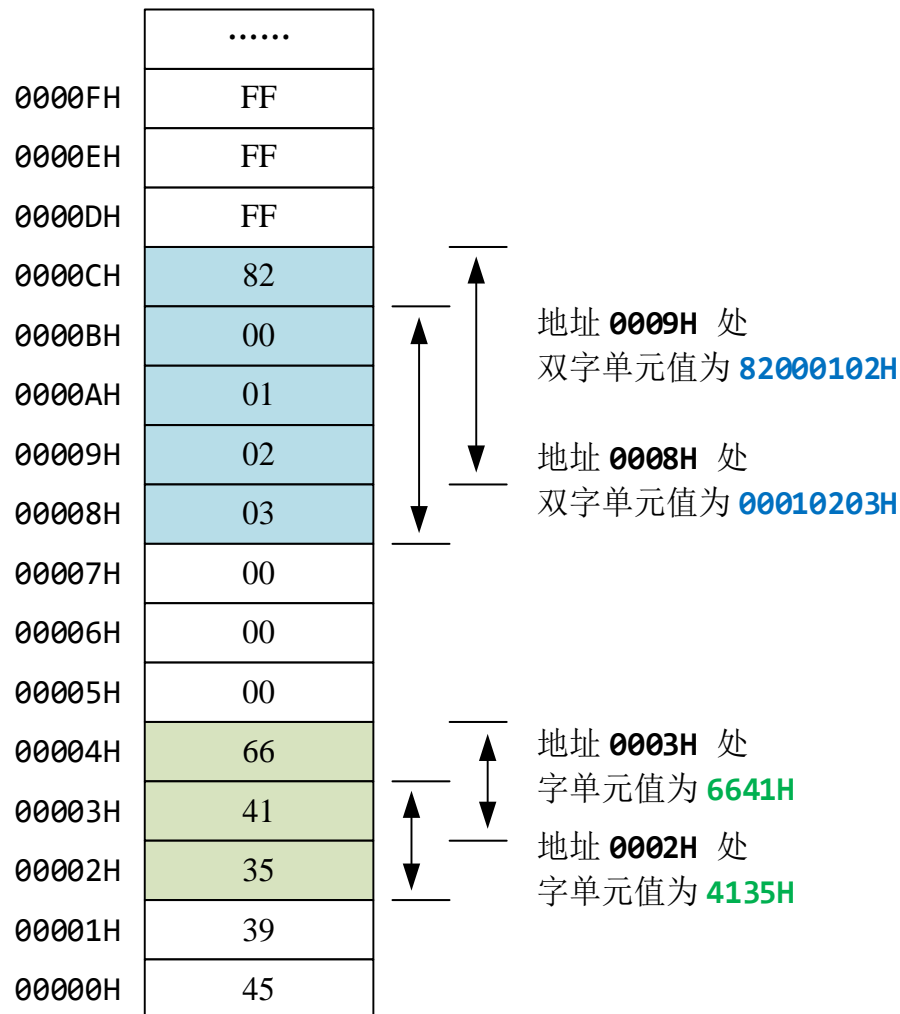
字节存储单元

- 内存看作为一个很大的由一系列**基本存储单元**线性地组成，每一个基本存储单元有一个唯一的地址
- 通常，**基本存储单元由 8 个连续的位构成，可用于存储一个字节的**数据，故基本存储单元也被称为**字节存储单元**

04 数据表示-数据的存储

类比

- 内存数据可以看作一维数组，**内存地址**可以看作为数组元素的**下标**
- **两个**连续的**字节**存储单元构成**一个字**存储单元
- **字存储单元的地址**是较低的字节存储单元的地址。
“高高低低”规则。
- **四个**连续的**字节**存储单元构成**一个双字**存储单元。



本章小结

- 了解微处理器、微型机、微型机系统的概念和关系
- 熟悉 CPU 的主要功能
- 理解汇编语言的特点和应用场合
- 理解汇编语言程序中的数据类型
- 掌握计算机内部的数据表示
 - 特别是补码的表示、特殊值的表示
 - 不同编码的数据位数扩展方法
- 熟悉补码的加减法运算规则，并能应用
- 熟悉常用的字符数据的编码形式
 - ASCII 码、BCD 码



河南大學
Henan University

0110000101100110011000010
11000110110010101101101
1000100110100101101010011
0101001101110001000000110
1000011010100110101001100
1010110111001111001011001
0100100000011001010110100
1011010000110001001100101
0010000000100000011010010
1111001011001010110111001
1011010111001101100011011
110010111011101100010011001101101001000
0001110011011110010110010100100000011000110
1100101011100100110001001100100001000000110
0101011010000111001101100001011011010111100
0001000000111100101110111011100010110010101
1011010111100001100100011101010111000101101
0000111001001100001011011100110101000100000
0110010001101001011010000110001001101000011
0001001011001010110100101101000011000100110
0101001000000010000001101001011110010110010
1011011100110110101110011011000110111100101
1101110110001001100110011011010010000001110
0110111100110010101101001011010000110001001
1001010010000000100000011010010111100101100
1010110111001101101011100110110001101111001
0111011101100010011001100110110100100000011
1001101111001100101011010010110100001100010
0110010100100000001000000110100101111001011
0010101101110011011010111001101100011011110
0101110111011000100110011001101101001000000
1110011011110011001010110100101101000011000
1001100101001000000010000001101001011110010

Q&A

主讲教师：舒高峰

电子邮箱：gaofeng.shu@henu.edu.cn

联系电话：13161693313