

新一代汇编语言程序设计

第一章 新一代汇编语言程序设计概述

- 微机系统简介
- 数据表示与类型
- 汇编语言基本概念



XU Aiping

计算机学院

第一章 新一代汇编语言程序设计概述

1.1 微机系统简介

一、微机系统组成结构

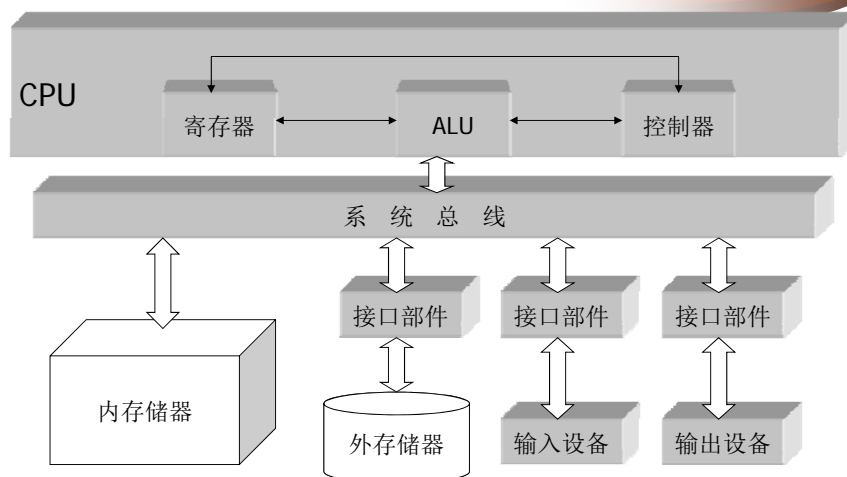
见下页图



XU Aiping

计算机学院

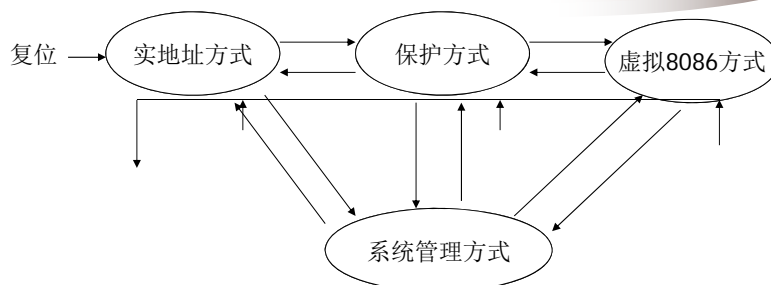
PC系列机的基本组成结构



Pentium处理器特性

- 奔腾系列**32位PC**增加了一系列新的功能，采用并融入了许多新的设计方法与技术。是**Pentium**系列仍与过去**32位Intel 80386**和**80486**系列**cPu**兼容，同时也兼容先前的**8086 / 8088**和**80286**，因而具有良好的软、硬件兼容性。
- **Pentium**支持的数据总线位数达到**64位**；支持的物理地址位数是**32位**；内部寄存器仍是**32位**。

二、80X86处理器（32位）的工作方式



1、实地址方式 (Real Mode)

- 处理器复位后被初始化为实地址模式。
- 内存进行分页处理，指令寻址的地址就是内存中实际的物理地址。
- 不支持优先级。
- 中断处理用中断向量表来定位中断服务程序，每个中断向量为4个字节。
- 实地址方式的**32位CPU**相当于可以进行**32位**处理的高速**8086**。
- 程序分段的大小最大为**64 KB**，段基址和偏移量都用**16位**表示，所以称这样的段为“**16位段**”。

2、保护方式 (Protected Model)

- 2. 保护方式
- 保护方式又称P方式(Pmtected Mode)，它是32位CPU的固有工作方式。由实地址方式可以进入32位保护方式，用于支持多任务的需要，程序员可以使用多达64 TB的虚拟(逻辑)存储空间。
- 它们的段基址和段内偏移量都是32位的，这样的段称为“32位段”。
- 这种工作方式之所以称为保护方式，是因为处理器在处理多任务时，每个程序各自分开，在自己的空间运行。实地址方式下，一个瞎跑的程序会使整个系统崩溃，而在保护方式下，一个程序最多也就是损坏自己。
- 为此，保护方式提供了一些保护设施，如：①任务地址空间的分离；②4个特权级的建立；③特权指令的使用；④段和页的访问权限(如：只读、只执行)；⑤段的上限检查。



XU Aiping

计算机学院

3、虚拟8086方式 (Virtual-8086 Model)

- 在保护方式下通过设置控制标志，使32位CPU可以转入V86方式，从而使得多个DOS程序可以同时运行，且相互独立，是一种在保护方式下运行的类似实方式的工作环境。
- V86方式下的程序都是运行在最低特权级下，而实地址方式的程序是运行在最高特权级下。



XU Aiping

计算机学院

4、系统管理方式 (System Manager Model)

- 当处理器接收到系统管理方式中断时，就会进入系统管理方式 **SM**
- **SM**不是为应用软件访问而设计的，而是由机器内的固件（装有程序的**ROM**）来控制的

小结：

- ◆ 对于**32位**的**CPU**，在实地址方式下执行**16位**段的程序；
- ◆ 保护方式下可以对**32位**和**16位**的程序单独或混合操作；
- ◆ 虚拟**8086**方式在保护方式下可以执行多个任务的**16位**程序；
- ◆ 系统管理方式为系统设计人员所使用。



XU Aiping

计算机学院

1.2 数据表示与数据类型

1、数的二进制和十六进制表示

- **4位**二进制数对应**1位16进制**数
- **16进制**数用带后缀字母**H**的数据表示，如 **1234H**
- 一个**16进制**的数开始数字不是**0~9**，则在开始数字 前加一个数字**0**表示，如 **0B1CFH**
- 二进制数用带后缀字母**B**表示，如 **10101100 B**



XU Aiping

计算机学院

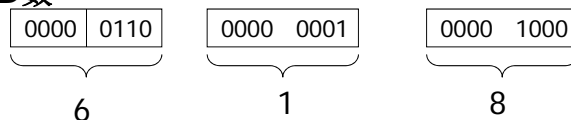
1.2 数据表示与数据类型

1.2.2、BCD码数据表示

➤ 压缩BCD数



➤ 非压缩BCD数



XU Aiping

计算机学院


1.2 数据表示与数据类型

1.2.3、非数值数据表示

字符编码：ASCII码（美国标准信息交换码）

表 1.1 ASCII 码表

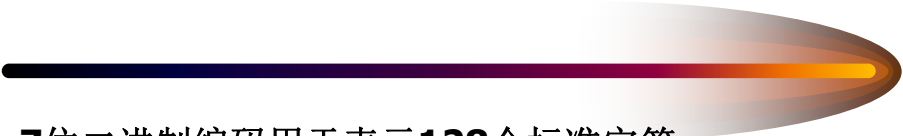
十六进制低位	十六进制高位							
	0	1	2	3	4	5	6	7
0			SP 空	0	@	P	,	P
1			!	1	A	Q	a	q
2			"	2	B	R	b	r
3			#	3	C	S	c	s
4			\$	4	D	T	d	t
5			%	5	E	U	e	u
6			&	6	F	V	f	v
7	BEL		^	7	G	W	g	w
8	BS		(8	H	X	h	x
9)	9	I	Y	i	y
A	LF		*	:	J	Z	j	z
B		ESC	+	;	K	[k	{
C			,	<	L	\	l	!
D	CR		-	=	M]	m	}
E			.	>	N	^	n	~
F			/	?	O	_	o	del

- 
- 在进行程序设计时应注意：
 - (1)空格SP的ASCII码为20H。换行LF的ASCII码为0AH或10。回车CR的ASCII码为0DH或13。
 - (2)字符0~9的ASCII码为30H-39H。即相应的数字加上十六进制数30H。由于数字符号的编码是连续的，只要记住数字符号的编码从30H开始，就可推出其他数字符号的编码。
 - (3)字母中大写字母A—Z的ASCII码为41H-5AH，小写字母a~z的ASCII码是61H~7AH。因大写、小写字母符号的编码分别是连续的，同样只要记住大写字母符号的编码从41H开始和小写字母的编码从61H开始，就可推出其他大、小写字母符号的编码。



XU Aiping

计算机学院



7位二进制编码用于表示128个标准字符

用8位二进制编码可表示扩展ASCII码，共256个

汉字编码： GB2312-80 国标编码16位编码：高8位表示汉字的区号，低8位表示汉字的位号。



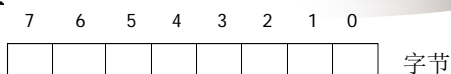
XU Aiping

计算机学院

1.2 数据表示与数据类型

1.2.4、基本数据类型

➤ 字节和字



- ◆ 一个字节可表示一个**ASCII**字符，也可分为**2**个**4**位的位组；
- ◆ 字的低**8**位称为低字节，高**8**位称为高字节；
- ◆ 双字由**2**个字组成，低**16**位称为低位字，高**16**位称为高位字。



XU Aiping

计算机学院

1.3 汇编语言基本概念

1.3.1 汇编语言相关特征

● 机器语言

是一种用二进制代码编写指令和程序的语言

例：**B0 04** 传送指令，将**4**送到累加器**AL**中

● 汇编语言

用助记符代替指令的操作码，用数字或符号代替指令中操作数或操作数存放位置的一种程序设计语言

例：**MOV AL, 04** ;传送指令，将**4**送到累加器**AL**中



XU Aiping

计算机学院

- 无论是基本汇编语言、宏汇编语言，还是新一代汇编语言，统称为汇编语言。用户用汇编语言编写的程序称为汇编语言源程序(**.asm**)，系统提供的能将汇编语言源程序翻译生成机器目标代码(**.obj**)的程序，称之为汇编程序(**Assembler**)。



- 机器语言和汇编语言是一种面向机器的语言，也称为低级语言。而高级语言是面向科学计算和实际问题的语言。汇编语言依赖于机器的硬件特性，而硬件特性的充分发挥也离不开汇编语言，它们的关系如下图所示。

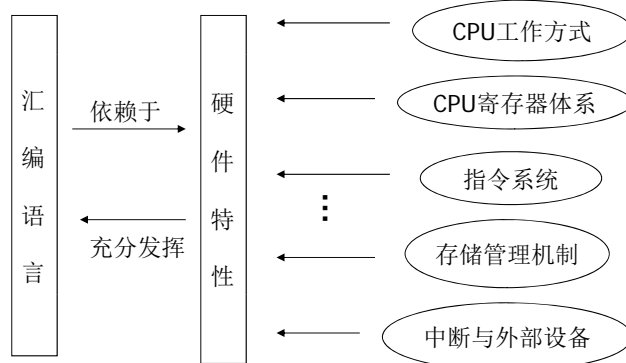


XU Aiping

计算机学院

1.3 汇编语言基本概念

- 汇编语言与机器特性密切相关



XU Aiping

计算机学院

1.3 汇编语言基本概念

1.3.2 汇编语句行基础知识

基本格式

[名字] 操作符 [操作数] [; 注释]

- 操作符可以是机器指令、伪指令(伪操作)或宏指令的助记符。如MOV、ADD等。机器指令、伪指令、宏指令的助记符。
- 操作数：操作数项有多个操作数时，用逗号，分开
- 语句行的名字是程序员自己建立的具有特定意义的字符序列。源程序中的名字可用大写的英文字母A-Z、小写字母a-x数字0-9以及一些特殊字符如?、@、等组成，一般以字母开头。名字的长度≤31个字符才有效。名字最好使用形象的或有意义的单词或单词的缩略形式表示，例如PORT1表示端口1的值。名字不能使用保留字。保留字是汇编语言中具有特定意义的名字，如指令助记符、伪指令名、寄存器名、参数关键字等。



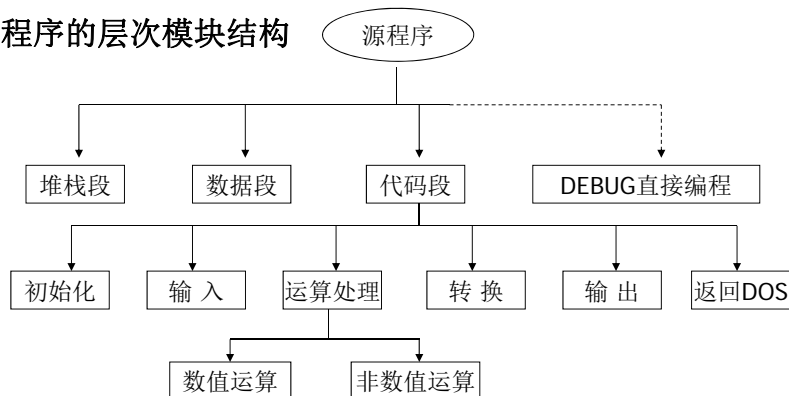
XU Aiping

计算机学院

1.3 汇编语言基本概念

1.3.3 汇编语言源程序示例

1、程序的层次模块结构



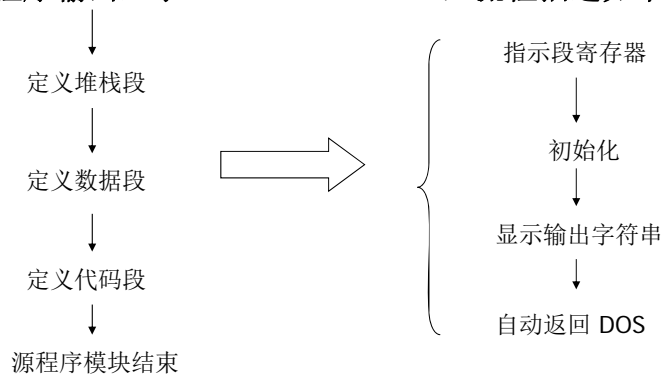
XU Aiping

计算机学院

1.3 汇编语言基本概念

2、源程序的分段结构及示例说明

例：程序输出显示：“Welcome !”，流程描述如下：



XU Aiping

计算机学院

源程序SAMPLE.ASM:

```
STACKS SEGMENT STACK
    DW 128 DUP(?)
STACKS ENDS
DATAS SEGMENT
    STRING DB 'Welcome!',13h,10h,'$'
DATAS ENDS
CODES SEGMENT
    ASSUME CS:CODES,DS:DATAS
START: MOV AX,DATAS
        MOV DS,AX
        LEA DX,STRING
        MOV AH,09H
        INT 21H
        MOV AX,4C00H
        INT 21H
CODES ENDS
        END START
```

堆栈段

数据段

代码段

源程序模块结束



XU Aiping

计算机学院

源程序SAMPLE.ASM:

```

STACKS SEGMENT STACK
    DW 128 DUP(?)
STACKS ENDS
DATAS SEGMENT

    DATAS ENDS
    CODES SEGMENT
        ASSUME CS:CODES,DS:DATAS
START:
        MOV AX,DATAS
        MOV DS,AX

        MOV AX,4C00H
        INT 21H
    CODES ENDS
    END START

```

堆栈段

数据段

代码段

源程序模块结束

武汉大学
Wuhan University

XU Aiping

计算机学院

本章结束

习题: 1.1, 1.3, 1.5., 1.9

武汉大学
Wuhan University

XU Aiping

计算机学院