汇编语言程序设计
第一章-第二章

张华平 副教授 博士

Email: kevinzhang@bit.edu.cn

Website: http://www.nlpir.org/

@ICTCLAS张华平博士

大数据搜索挖掘实验室（wSMS@BIT）

参考书及答疑

- 教　　材：Intel80x86/Pentium汇编语言程序设计(第3版) 北京理工大学出版社 张雪兰、谭毓安、李元章
- 办公地点：中心教学楼1013

课时安排

时间安排：9月20日-12月20日(36)

重点章节：第2章(4h)、第3章(8h)、第4章(4h)、第5章(4h) 第6章(6h)、第9章(4h)

部分讲解：第7、8章(2h) 复习(2h) 导论2h

实 验 课：待定（6:30-9:10）

课程简介

- 发挥计算机硬件特性
- 满足苛刻的实时处理要求
- 空间、速度的要求
- 有助于对计算机底层的了解

课程相关知识

- 计算机组成原理/体系结构/接口技术
- 计算机硬件知识/微处理器
- 嵌入式/微控制器
- 指令集
- 操作系统

主要教学内容

- 微型计算机硬件系统简介
- 寻址方式与指令系统
- 汇编语言程序组织/基本结构/程序设计
- 子程序设计/宏指令设计
- I/O程序设计
- 系统功能及中断调用/模块化程序设计
- 保护模式及其编程

软硬件平台

平　台：Intel 80X86/Pentium

DOS/虚拟8086模式(V86)

Windows/保护模式

MASM5.1　MASM6.11　MASM32

上机过程（实模式）

上机过程：masm→link→.exe / .com

编辑：temp.asm

汇编：masm temp.asm→temp.obj

连接：link temp.obj→temp.exe

调试方式：Debug

第一章 基础知识

- 二进制、八进制、十进制、十六进制转换
- 二进制算术/逻辑运算
- 编码

  数字编码

  字符编码 → ASCII码

  汉字编码 → GB码

  压缩BCD码（1个字节表示2个十进制数字）

  /非压缩BCD码（1个字节表示1个十进制数字）

第二章 硬件系统简介

2.1 微型计算机系统简介

2.2 汇编语言概述

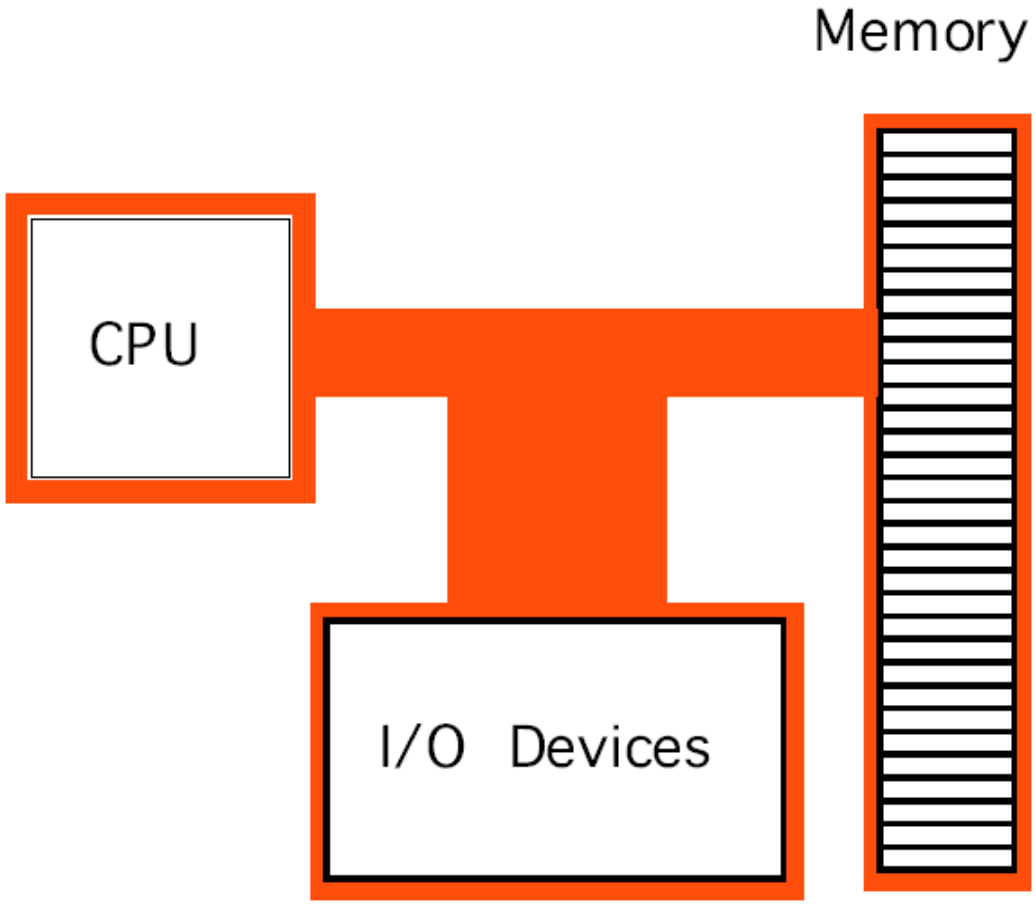
2.3 Intel公司微处理器简介

2.4 程序可见寄存器组

2.5 存 储 器

2.6 外部设备

2.1 计算机系统硬件组成[1]—总体结构



Von Neumann system

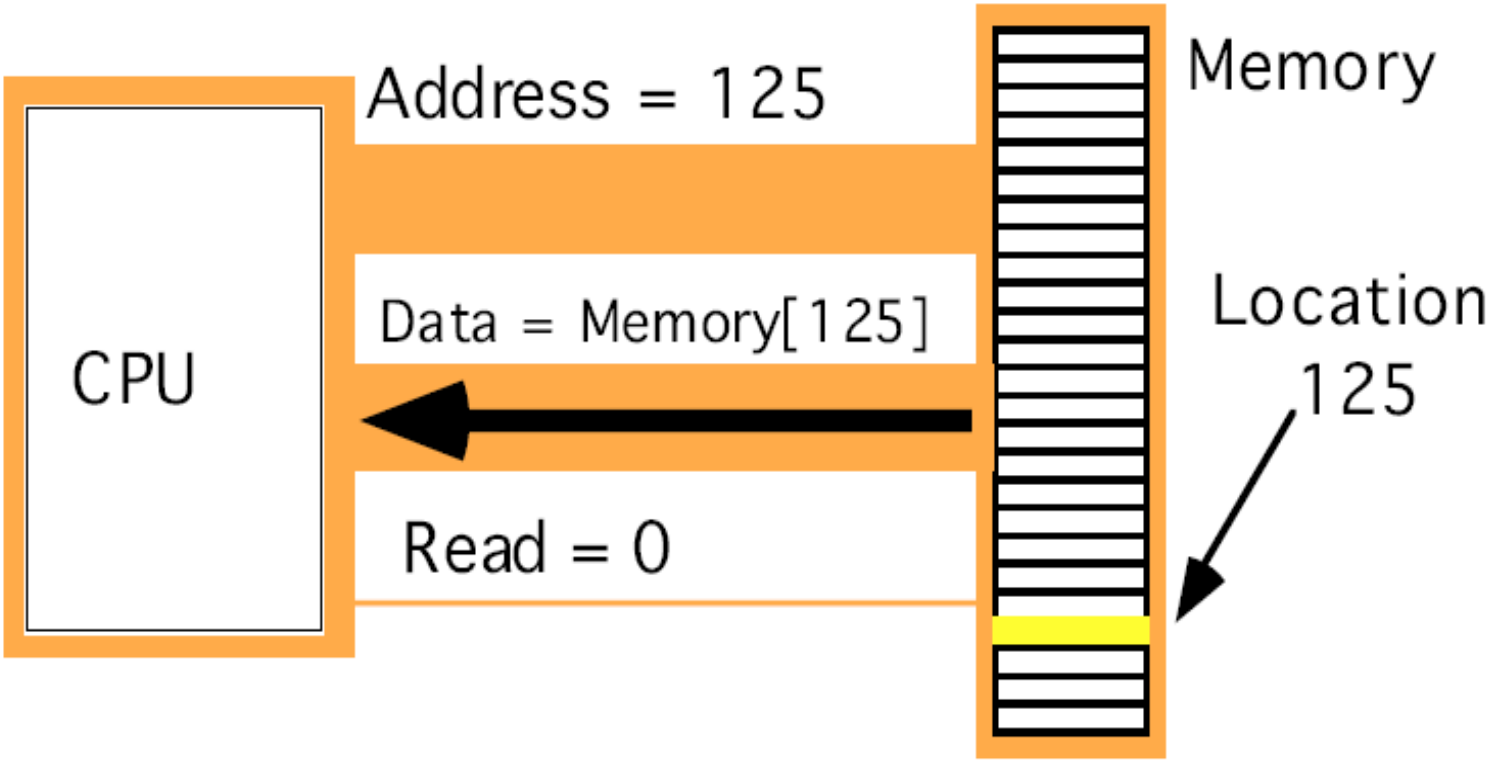
计算机系统硬件组成[2]—组成部分

- CPU
- I/O Device
- Memory
- System Bus
  - Data Bus
  - Address Bus
  - Control Bus

计算机系统硬件组成[3]—总线

CPU	DB	AB
8086	16	20
80386DX	32	32
Pentium III	64	36
Itanium	64	44

计算机系统硬件组成[4]—读操作



Read Operation

计算机系统软件组成

- 系统软件

  操作系统、翻译程序、辅助程序。

- 应用软件

  为解决各类实际问题而设计的程序。

2.2 程序设计语言—分类

机器语言：直接用二进制代码的机器指令表示的语言。

汇编语言：用指令助记符、符号地址、标号等符号书写程序的语言。

高级语言：高级语言是一种类似于人类语言的语言。

实现：123+456 → SUM

C语言实现：

```
main( )
{
  int       a,b,sum;
  a=123;b=456;
  sum=a+b;
}
```

机器语言实现 123+456 → SUM[1]

用DEBUG环境下的E命令把机器指令及数据输入内存：

①键入程序代码：

```
- e cs:100 a1 0f 01 03 06 11 01
          a3 13 01 b8 00 4c cd 21
```

②为数据分配空间：

```
- e ds:10f 7b 00 c8 01 00 00
```

③运行并得到结果：[G=100　D DS:10F L 2]

```
43 02
```

4302是加法和579的十六进制表示形式，并以逆序存储为0243。

机器语言实现 123+456 → SUM[2]

这里需要考虑：

- 给代码和数据分配内存空间。
- 熟悉机器指令及其格式。
- 把十进制数转换成十六进制数。
- 熟悉数据在内存中的存放顺序。例如，123的十六进制数表示为007b，若在内存中占用一个字，其存放顺序为7b 00。同样c8 01为456的十六进制表示形式。

汇编语言实现 123+456 → SUM

```
code            segment
                org     100h
                assume  cs:code, ds:code
main  proc  near

                mov     ax, a
                add     ax, b
                mov     sum, ax
                mov     ax, 4c00h
            Int     21h

A       dw      123
B       dw      456
sum  dw         ?

main            endp
code            ends
                end     main
```

反汇编代码

反汇编出来的代码部分：

```
12F8:0100 A10F01      MOV    AX,[010F]
12F8:0103 03061101    ADD    AX,[0111]
12F8:0107 A31301      MOV    [0113],AX
12F8:010A B8004C      MOV    AX,4C00
12F8:010D CD21         INT    21
```

用a、b、sum名字定义的数据在机器中显示为：

```
12F8:010f   7B 00 C8 01 43 02
```

汇编语言定义

汇编语言是一种符号化了的机器语言，即用指令助记符、符号地址、标号等符号书写程序的语言。

三种语言的比较

	机器语言	汇编语言	高级语言
基本形式	二进制	助记符	语句
编译程序	不需要	需要	需要
执行效率	高	高	低
占用空间	少	少	多
CPU依赖性	依赖	依赖	不依赖
编程难度	复杂	中等	容易

汇编语言特点

特点：占用空间少、执行速度快、直接控制硬件能力强，但不容易掌握、开发周期较长且可移植性差。

应用：空间、时间要求苛刻；中断处理程序、外设驱动程序、系统软件的核心程序、游戏的关键部分。

基础：应熟悉机器的内部结构及与编程相关的硬件知识，例如CPU的寄存器组、存储器结构、外设、中断系统等。

2.3 Intel 80X86系列CPU

CPU	数据总线	地址总线	寻址能力	工作模式
8086	16	20	1M	实模式
8088	8	20	1M	实模式
80286	16	24	16M	实模式、保护模式
80386	32	32	4G	实模式、保护模式
80486	32	32	4G	实模式、保护模式
Pentium	64	36	64G	实模式、保护模式
Pentium IV	64	36	64G	实模式、保护模式

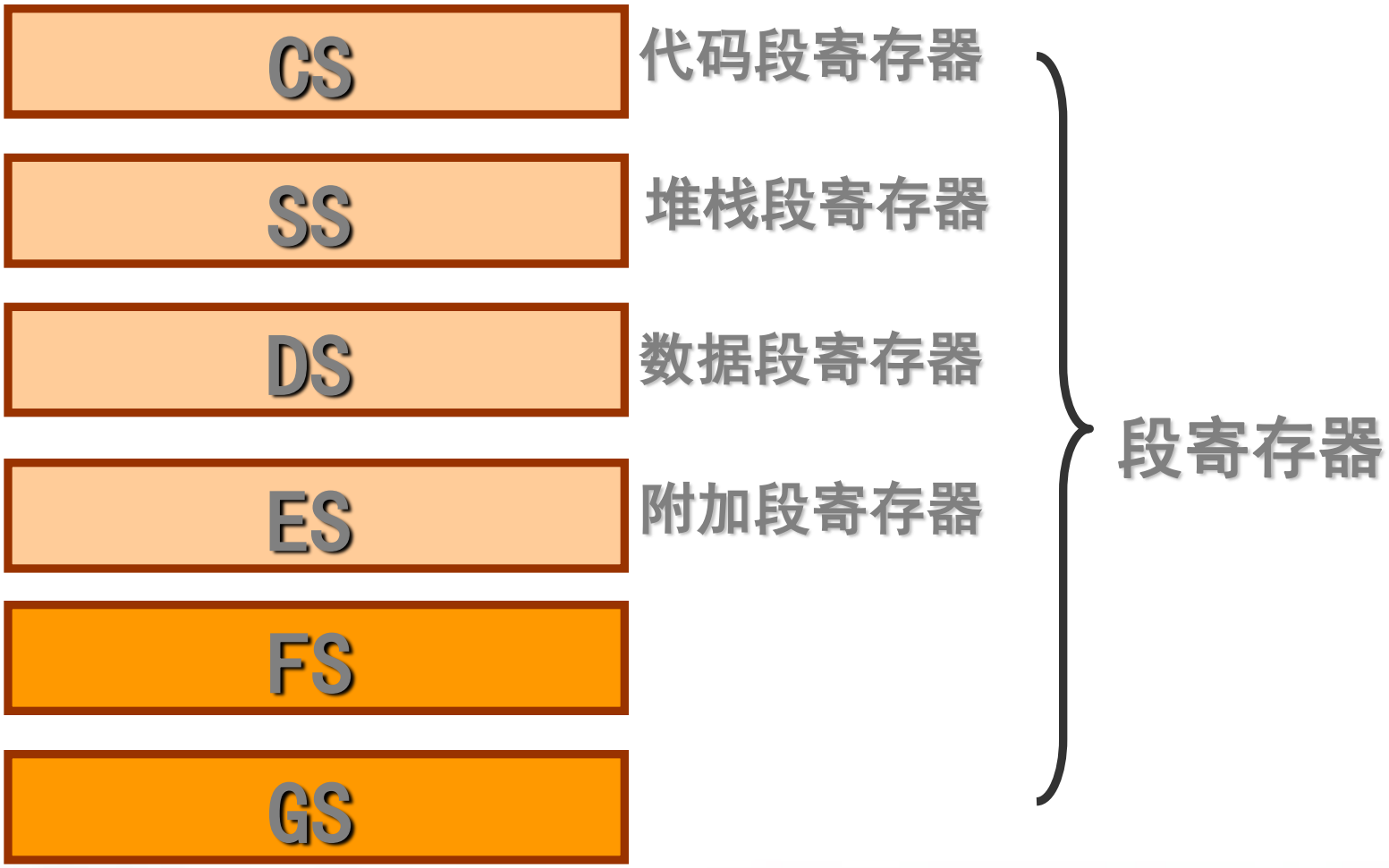
2.4 程序可见寄存器组

程序可见寄存器组包括多个8位、16位和32位寄存器。如下页图所示。阴影部分只对80386（含80386）以上CPU有效。

1. 通用寄存器
2. 段寄存器
3. 控制寄存器

8086~Pentium 程序可见寄存器组

8086~Pentium 段寄存器



1.通用寄存器---数据寄存器

数据寄存器

8位：AL AH BL BH CL CH DL DH

16位：AX BX CX DX

32位：EAX EBX ECX EDX

[386]

通用寄存器---指针寄存器[1]

堆栈指针寄存器SP、ESP（386以上）：

功能：存放当前堆栈段栈顶偏移量，总是与SS堆栈段寄存器配合存取堆栈中的数据。

说明：实模式使用SP；

保护模式使用ESP。

通用寄存器---指针寄存器[2]

基址指针寄存器BP、EBP（386以上）：

功能：存放地址的偏移量部分或数据。

若存放偏移量时，缺省情况与SS配合。

说明：实模式使用BP；

保护模式使用EBP。

通用寄存器---变址寄存器

变址寄存器SI、DI、ESI、EDI：

功能：存放地址的偏移量部分或数据。若存放偏移量时，缺省情况与DS配合。

说明：实模式使用SI、DI；

保护模式使用ESI、EDI。

通用寄存器说明

注意：

除SP、ESP堆栈指针不能随意修改、需要慎用外，其它通用寄存器都可以直接在指令中使用，用以存放操作数，这是它们的通用之处，其它专用用途在具体指令中介绍。

2.段寄存器

简介：IBM PC机的存储器采用分段管理方法组织，因此一个物理地址用段基址和偏移量表示。一个程序可以由多个段组成。

段寄存器功能：段寄存器存放段基址。在实模式下存放段基地址，在保护模式下存放段选择子。

段寄存器[1]

代码段寄存器CS：指定当前代码段，代码段中存放当前正在运行的程序段。

堆栈段寄存器SS：指定当前堆栈段。

说明：堆栈段是在内存开辟的一块特殊区域，其中的数据访问原则是后进先出（LIFO），允许插入和删除的一端叫做栈顶。IBM PC机中SP（或ESP）指向栈顶，SS指向堆栈段基地址。

段寄存器[2]

数据段寄存器DS：指定当前运行程序所使用的数据段。

附加数据段寄存器ES：指定当前运行程序所使用的附加数据段。

段寄存器FS和GS：指定当前运行程序的另外两个存放数据的存储段（只对80386以上机器有效）。

段寄存器说明

说明：

虽然DS、ES、FS、GS（甚至于CS、SS）所指定的段中都可以存放数据，但DS是主要的数据段寄存器，在默认情况下使用DS所指向段的数据。若要引用其它段中的数据，通常需要显式说明。

3.控制寄存器

控制寄存器包括指令指针寄存器和标志寄存器。

注意：在程序中不能直接引用控制寄存器名。

控制寄存器---指令指针寄存器

指令指针寄存器IP、EIP

功能：总是与CS段寄存器配合指出下一条要执行指令的地址，其中存放偏移量部分。实模式使用IP，保护模式使用EIP。

控制寄存器---标志寄存器

标志寄存器（FLAGS）

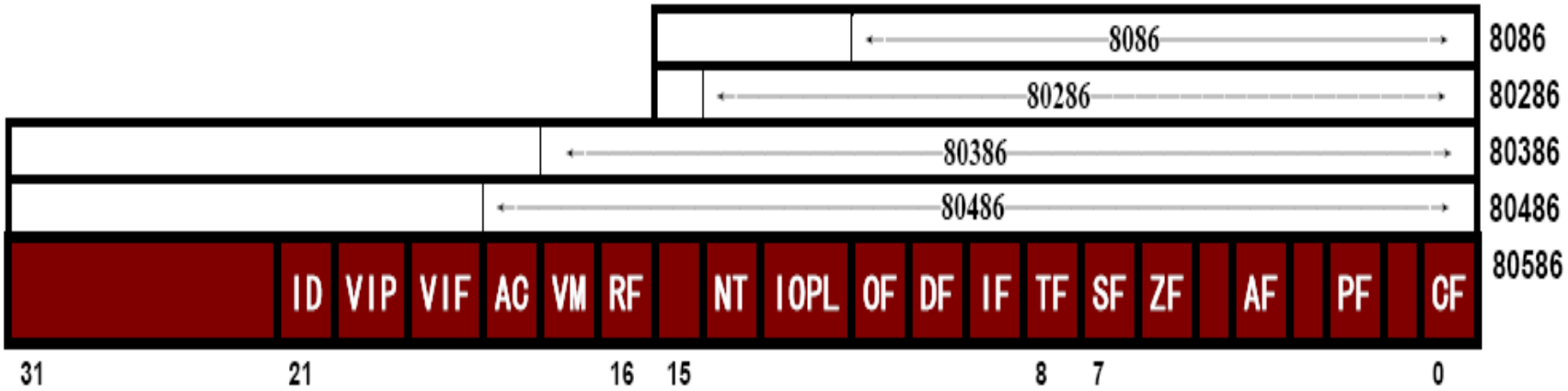
标志寄存器也被称为状态寄存器，由运算结果特征标志和控制标志组成。

重点：各档CPU均有的标志。即8086拥有的9个标志。

标志寄存器说明

8086/286为16位

80386/486/586以上为32位



标志寄存器---运算结果特征标志

运算结果特征标志：

用于记录程序中运行结果的特征。8086~Pentium的标志寄存器均含有CF、PF、AF、ZF、SF、OF这6位标志。

FLAGS---CF

CF：进位标志，记录运算结果的最高位向前产生的进位或借位。可用于检测无符号数运算时是否发生溢出。

CF＝1　有进位或借位

CF＝0　无进位或借位

FLAGS---PF

PF：奇偶标志，记录运算结果最低8位中含1的个数。可用于检测数据传送过程中是否发生错误。

PF＝1　个数为偶数

PF＝0　个数为奇数

FLAGS---AF

AF：辅助进位标志，记录运算结果最低4位向前产生的进位或借位。只有在执行十进制运算指令时才关心此位。

AF＝1有进位或借位

AF＝0无进位或借位

FLAGS---ZF / SF

ZF：零标志，记录运算结果是否为0。

ZF＝1　运算结果为零

ZF＝0　结果非零

SF：符号标志，记录运算结果的符号。

SF＝1　运算结果为负

SF＝0　结果非负

FLAGS---OF

OF：溢出标志，记录运算结果是否超出了机器所能表示的范围。可用于检测带符号数运算时是否发生溢出。

OF＝1　运算结果超出范围
OF＝0　结果未超出

FLAGS---控制标志

控制标志：

控制标志控制处理器的操作，要通过专门的指令才能使控制标志发生变化。

控制标志包括：IF、DF、TF等

FLAGS---IF / DF / TF

IF：中断允许标志。IF的控制只对外部可屏蔽中断请求（INTR）起作用。

IF＝1/0　允许/禁止 CPU响应INTR

DF：方向标志。专服务于字符串操作指令，指示串操作时操作数地址的增减方向。

DF＝1/0　串操作时操作数地址为自动减/增量

TF：陷阱标志。用于程序调试。

TF＝1/0　CPU处于单步/连续方式

FLAGS---IOPL / NT[80286]

IOPL（I/O Privilege Level）：特权标志，占两位。当在保护模式工作时，IOPL指定要求执行I/O指令的特权级。

NT：嵌套任务标志。保护模式在执行中断返回指令IRET时要测试NT值。

FLAGS---RF / NT[80386]

RF： 重启动标志。该标志控制是否接受调试故障。

RF＝0时接受；RF＝1时忽略。

VM： 虚拟方式标志。当CPU处于保护模式时，若VM＝1则切换到虚拟模式，以允许执行多个DOS程序；否则CPU工作在一般的保护模式。

FLAGS---AC / NT[80486]

AC： 地址对齐检查标志。若AC＝1时进行地址对齐检查。若AC＝0时不进行地址对齐检查。

所谓地址不对齐是指以下情形：一个字从奇地址开始，或一个双字不是从4的倍数的地址开始。

FLAGS---AC / NT[Pentium]

ID：标识标志。若ID＝1，则表示Pentium支持CPUID指令，CPUID指令给系统提供Pentium微处理器有关版本号及制造商等信息。

VIP：虚拟中断挂起标志。与VIF配合，用于多任务环境给操作系统提供虚拟中断挂起信息。

VIF：虚拟中断标志，是虚拟方式下中断标志位的映像。

2.5 存储器---基本概念[1]

一、数据

1、二进制位：存储信息的基本单位，1Gb。

2、字节：存取信息的基本单位，1GB。编号b_7～b_0 。

3、字：一个字16位，占用两个存储单元。其位编号为b_{15}～b_0。

4、双字：一个双字32位，占用四个存储单元。其位编号为b_{31}～b_0。

5、四字：一个四字64位，占用八个存储单元。其位编号为b_{63}～b_0。

二、存储器地址

为了正确地区分不同的内存单元，给每个单元分配一个存储器地址，地址从0开始编号，顺序递增1。在机器中地址用无符号二进制数表示，可简写为十六进制数形式。

三、单元的内容

一个存储单元中存放的信息称为该单元的内容。

1. 访问字、双字、四字：

访问时只需给出最低单元的地址号即可，然后依次存取后续字节。

2. 逆序存放：

按照Intel公司的习惯，其低地址中存放低位字节数据，高地址中存放高位字节数据，这就是所谓“逆序存放”含义。

存储器---基本概念[3]

例. 内存现有以下数据（后缀H表示是十六进制数）：

地址： 0 1 2 3 4 5

内容：12H 34H 45H 67H 89H 0AH.....

则对于不同的数据类型，该单元的数据是：

$(0)_{字节}$=

$(3)_{字}$ =

$(1)_{双字}$=

存储器---存储器分段管理

IBM PC机的存储器采用分段管理的方法。存储器采用分段管理后，一个内存单元地址要用段基地址和偏移量两个逻辑地址来描述，表示为　　，其段基址和偏移量的限定、物理地址的形成视CPU工作模式决定。

存储器---实模式存储器寻址[1]

3种工作模式

实模式：微处理器只可以寻址最低的1M字节。

保护模式：寻址4GB

虚拟86模式：寻址类似于8086

存储器---实模式存储器寻址[2]

对段基址的限定：

只要工作在实模式，段基址必须定位在地址为16的整数倍上，这种段起始边界通常称做节或小段。

对段长的限定：

在实模式下段长不能超过64K。

存储器---实模式存储器寻址[3]

存储器采用分段管理后，其物理地址的计算方法为：

10H×段基址＋偏移量

简便的计算方法：因为段基址和偏移量一般用十六进制数表示，直接在段基址的最低位补以0H，再加上偏移量。

实模式存储器寻址示例

例．某内存单元的地址用十六进制数表示为1234:5678，则其物理地址为

12340H＋5678H＝179B8H。

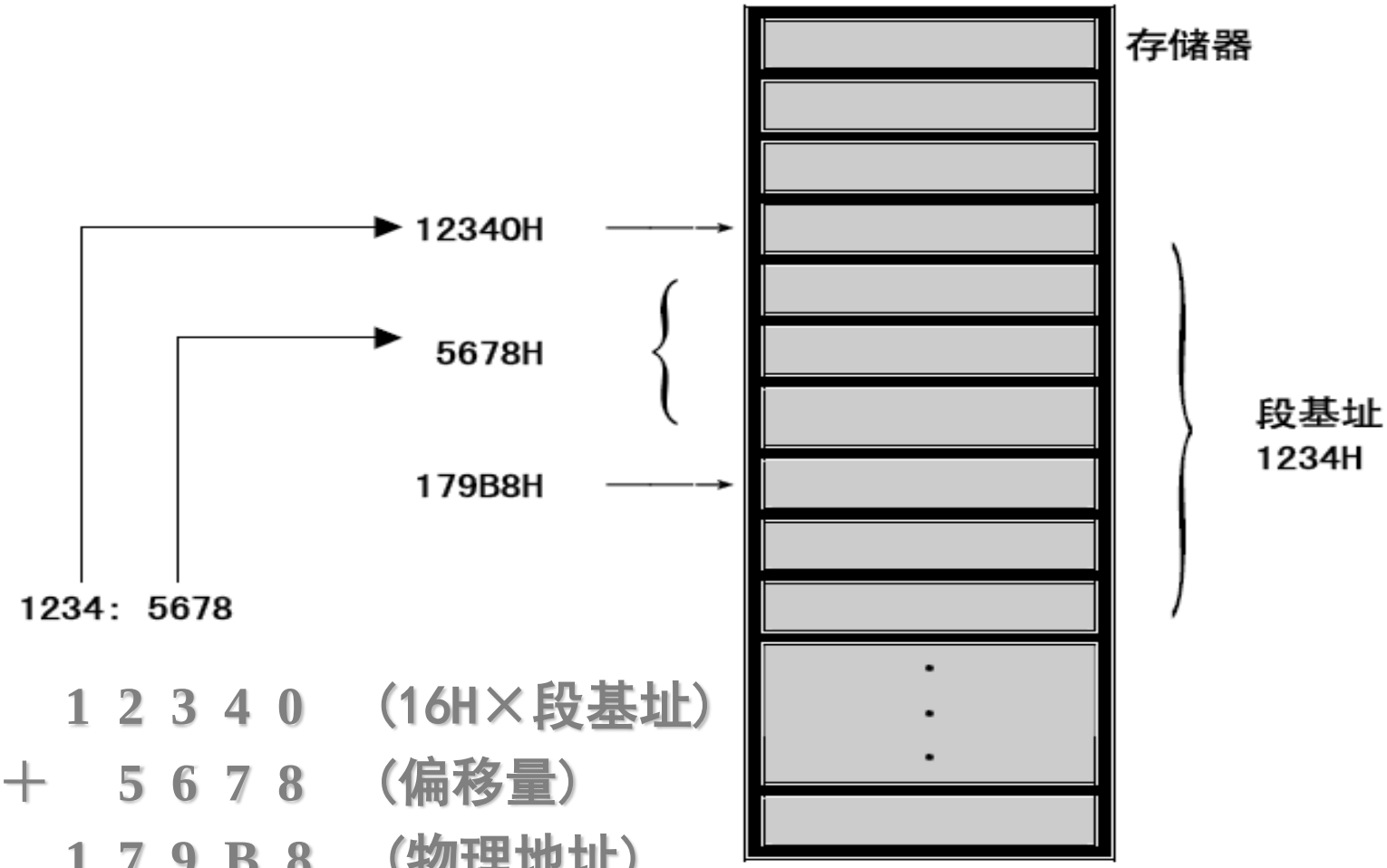

存储器
12340H
5678H
179B8H
段基址
1234H
1234：5678
1 2 3 4 0 （16H×段基址）
＋ 5 6 7 8 （偏移量）
1 7 9 B 8 （物理地址）

图2-5 物理地址的形成

实模式寻址约定规则

程序执行时，其当前段的段基址放在相应的段寄存器中，偏移量视访问内存的操作类型决定，其规律如下页表所示。

操作类型	约定段寄存器	允许指定的段寄存器	偏移量
1. 指令	CS	无	IP
2. 堆栈操作	SS	无	SP
3. 普通变量	DS	ES、SS、CS	EA
4. 字符串指令的源串地址	DS	ES、SS、CS	SI
5. 字符串指令的目标串地	ES	无	DI
6. BP用作基址寄存器	SS	DS、ES、CS	EA

保护模式寻址

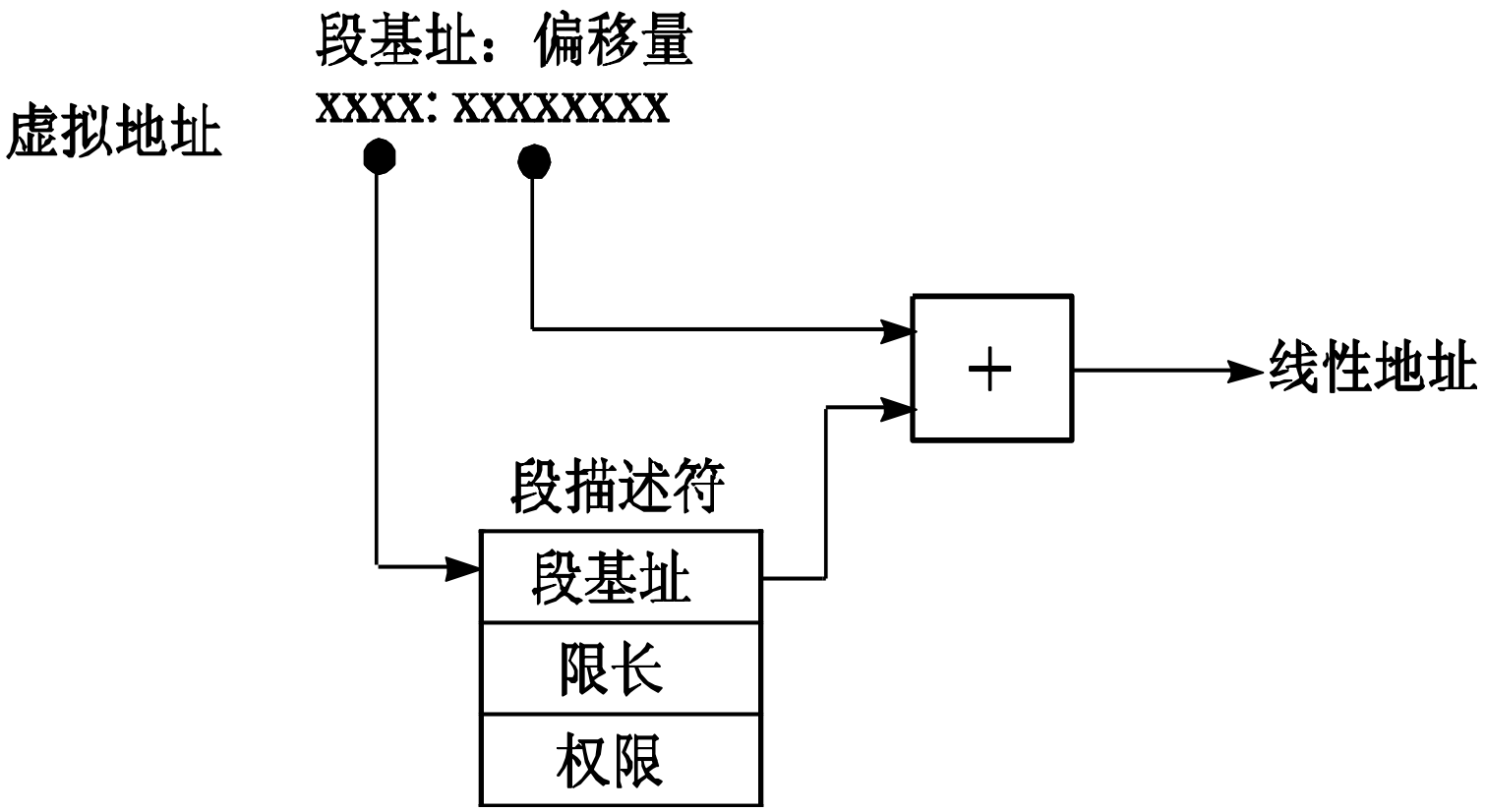
在保护模式下，其内存管理既可以使用分段机制访问多达4 GB（386/486）或64 GB（Pentium）的内存空间，也可以使用分页机制访问多达16 TB的虚拟存储器。总之，保护模式打破了实模式只允许访问装在内存第一个1 MB之内的程序和数据的限制。

段寄存器中存放一个段选择符（不是实模式下段基址的高16位值），通过段选择符从段描述符中得到32位的段基址，再与偏移量相加，得到32位的线性地址。

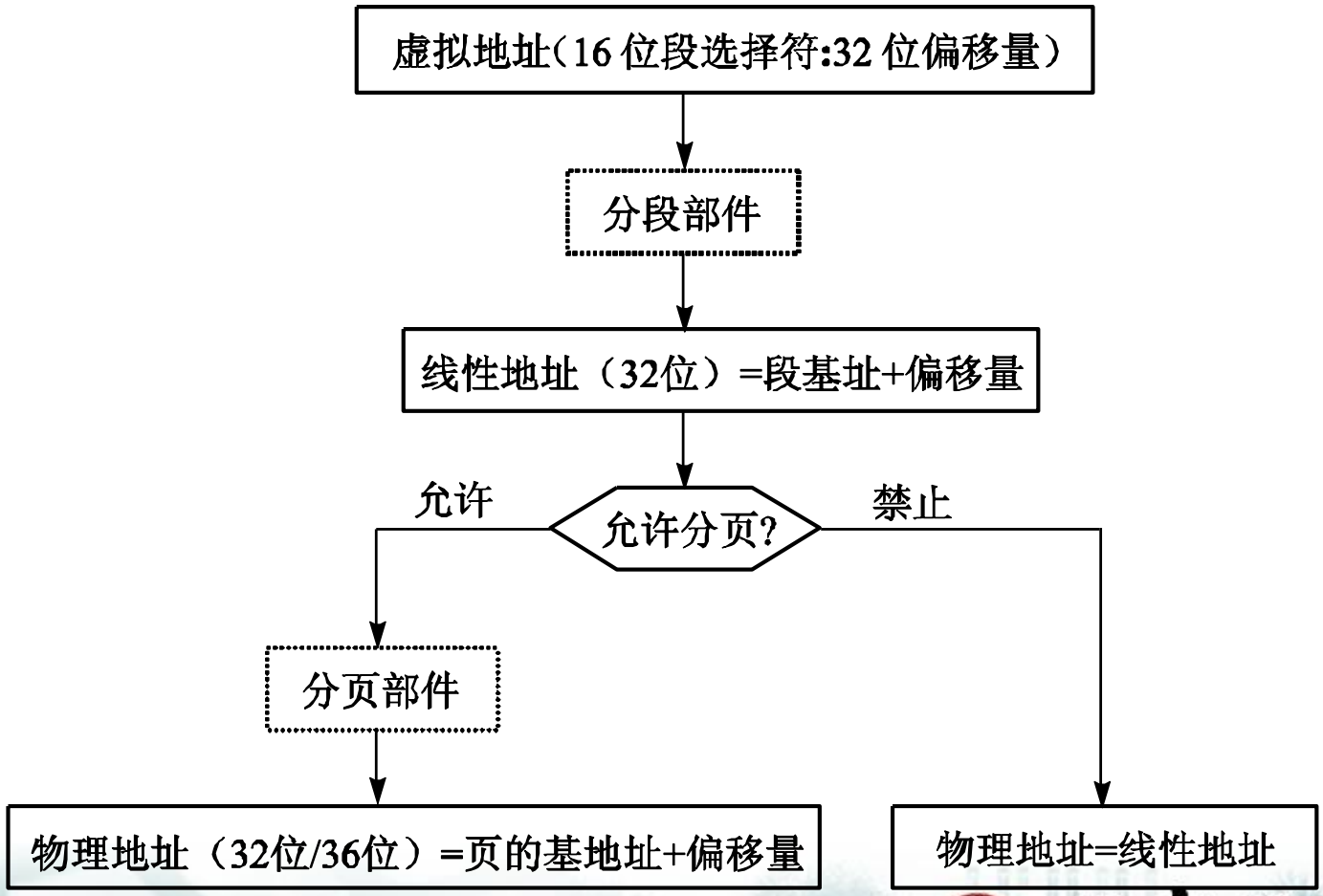
若允许分页功能，则CPU的分页部件将线性地址转换为物理地址，否则线性地址直接作为物理地址使用。

物理地址的形成过程比实模式复杂，但是自动形成，编程者不必关心。

虚拟地址到线性地址的转换



虚拟地址到物理地址的转换



保护模式存储器管理

- 段描述符
- 段选择符
- 分页机制

2.6 外部设备

外部设备包括输入设备、输出设备、外存储器。

I/O地址空间：外设与主机的信息交换是通过外设接口进行的，每个接口中都有一组寄存器，用来存放要交换的数据、状态和命令信息。为了能区分这些寄存器并且便于主机访问，系统给每个接口中的寄存器赋予一个端口地址（或称端口号），由这些端口地址组成了I/O地址空间。

IBM PC系列机所提供的I/O地址总线宽度总是16位的，所以允许最大的I/O寻址空间为64K。在IBM PC系列机中，由于I/O地址空间是独立编址的，因此系统需要提供独立的访问外设指令。

作业

- 习题2

2.3　　2.4　2.5　2.6　2.16

2.3　12F8:0100　1A2F:0103　1A3F:0003　1A3F:A1FF

2.5　12FA:0000　03　06　11　A3　13　01

12FA:0002的字节型、字型、双字

感谢关注聆听！

张华平
Email: kevinzhang@bit.edu.cn
微博：@ICTCLAS张华平博士
实验室官网：
http://www.nlpir.org

大数据千人会