



## 本章重点难点

- 微处理器（CPU）；
- CPU的内部和外部结构；
- 微处理器级总线的概念；
- CPU的功能结构；
- 8086寄存器结构；
- 存储器组织与分段、I/O端口地址空间；



## 2.1 微处理器的外部结构

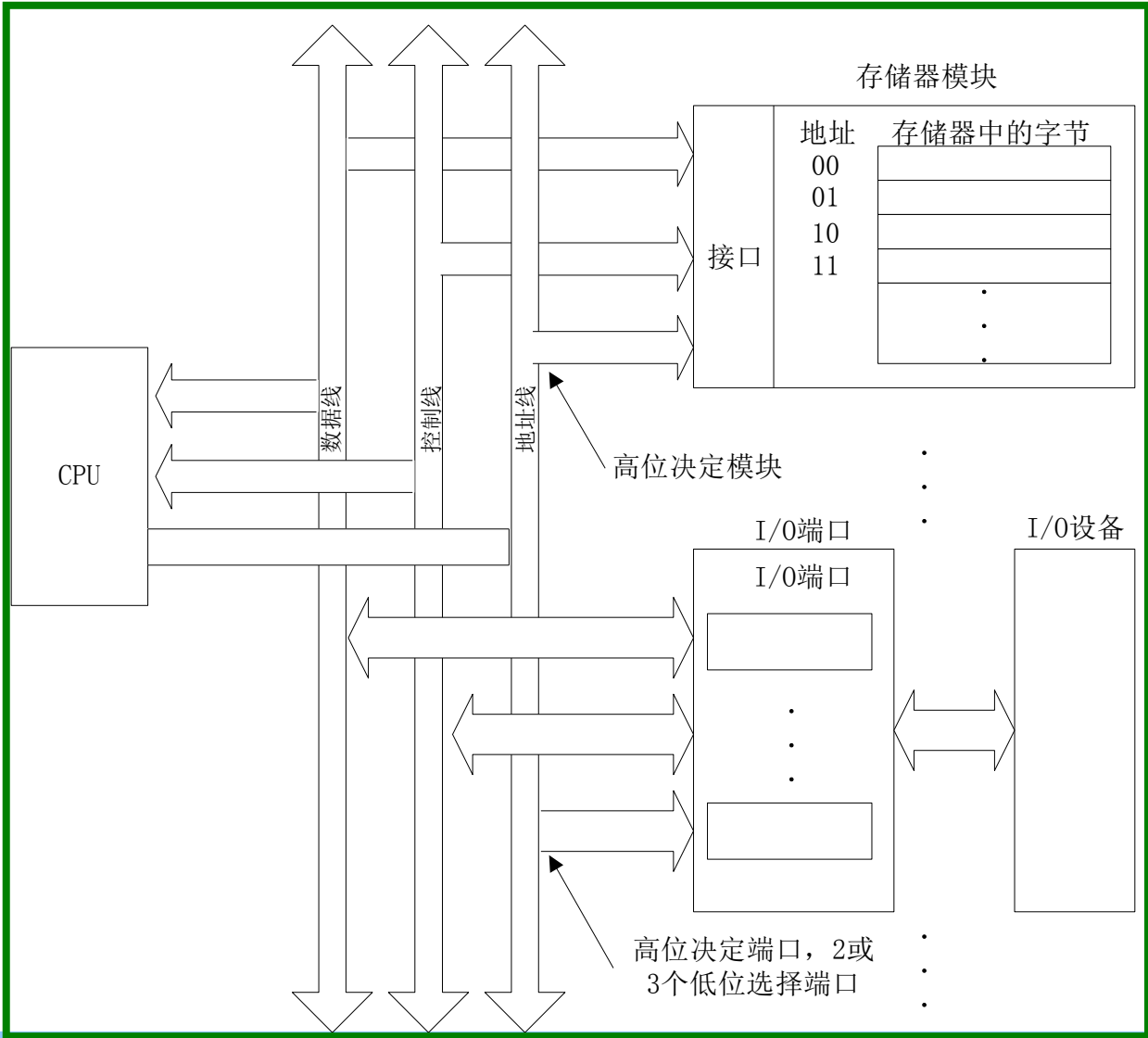
8086 CPU片有40个管脚，微处理器通过这些引脚与外部的逻辑部件连接，完成信息的交换。CPU的这些引脚信号称为**微处理器级的总线**，它应该能够完成下列功能：

- 与存储器之间交换信息(指令及数据)；
- 与I/O设备之间交换信息；
- 能输入和输出必要的信号。



# 第2章 8086CPU结构与功能

## 微处理器的外部结构





## 2.1 微处理器的外部结构

按功能分，这些总线可以分为三种：

- ( 1 ) 传送信息 ( 指令或数据 ) 的数据总线  
( Data Bus )
- ( 2 ) 指示欲传信息的来源或目的地址的地址总线  
( Address Bus )
- ( 3 ) 管理总线上活动的控制总线  
( Control Bus )



# 第2章 8086CPU结构与功能

## 2.1 微处理器的外部结构

### 地址总线：

CPU通过地址总线输出地址码来选择某一存储单元或某一称为I/O端口的寄存器，是单向的。

地址码的位数决定了地址空间的大小。

n位地址总线可有  $2^n$  个地址 (  $0 \sim 2^n - 1$  ) 。

- 16位地址总线：65536 ( 64KB )
- 20位地址总线：1MB
- 32位地址总线：4GB

$$2^{20} = (2^{10})^2 = (1024)^2 = 1MB$$



## 2.1 微处理器的外部结构

- 8086/8088地址总线：20位
- 存储器地址总线20位，地址空间：1MB
- I/O地址总线16位（低16位）地址空间：64KB



# 第2章 8086CPU结构与功能

## 2.1 微处理器的外部结构

### 数据总线：

- 用于CPU和存储器或I/O接口之间传送数据，是双向的。
- 微处理器数据总线的条数决定CPU和存储器或I/O设备一次能交换数据的位数，是区分微处理器是多少位的依据。
- 8086CPU的数据总线是16条，就说8086CPU是16位微处理器。



## 2.1 微处理器的外部结构

### 控制总线：

管理总线上的活动，用来传送自CPU发出的控制信息或外设送到CPU的状态信息，大部分是单向的，有一些是双向的。





## 2.2 微处理器的内部结构

- 微处理器是组成计算机系统的核心部件，它具有运算和控制的功能。具体地讲，CPU 应具有下述基本功能:



## 2.2 微处理器的内部结构

1. 进行算术和逻辑运算；
2. 具有接收存储器和I/O接口来的数据和发送数据给存储器和I/O接口的能力；
3. 可以暂存少量数据；
4. 能对指令进行寄存、译码并执行指令所规定的操作；
5. 能提供整个系统所需的定时和控制信号；
6. 可响应I/O设备发出的中断请求。



## 2.2 微处理器的内部结构

### 1.CPU内部结构及各部分功能简介

典型的CPU内部结构如图所示。



# 第2章 8086CPU结构与功能

## 典型的CPU内部结构

CPU

### 控制器

程序计数器 (PC)

指令寄存器 (IR)

指令译码器 (ID)

控制逻辑部件

堆栈指示器 (SP)

状态寄存器 (PSW)

### 工作寄存器

地址寄存器



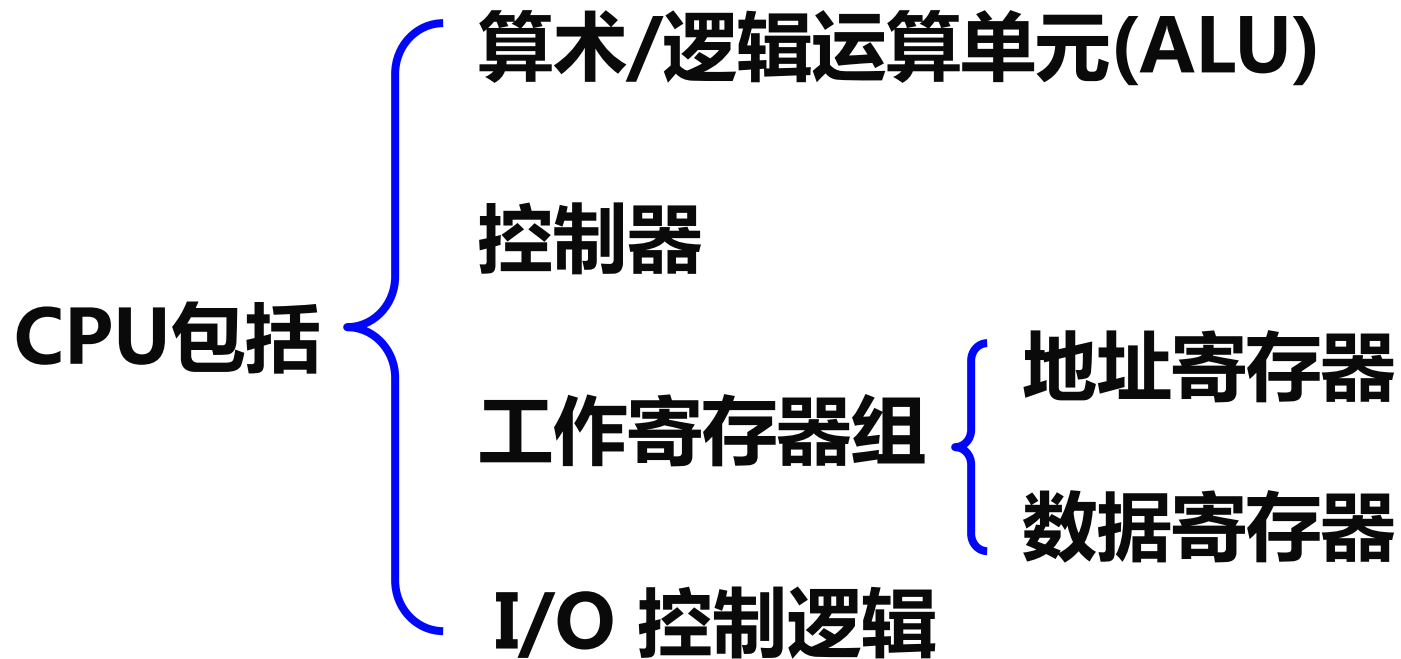
数据寄存器



I/O控制逻辑



## 2.2 微处理器的内部结构





## 2.2 微处理器的内部结构

### ① 算术逻辑运算单元 (Arithmetic/Logic Unit , ALU )

它是运算器的核心，几乎所有的算术运算，逻辑运算和移位操作都是由 ALU 完成的。



## 2.2 微处理器的内部结构

### ② 工作寄存器

暂存用于寻址和计算过程的信息。工作寄存器分为两组：**数据寄存器组和地址寄存器**。但有的寄存器兼有双重用途。

- 数据寄存器用来暂存操作数和中间运算结果；
- 地址寄存器用于操作数的寻址。



# 第2章 8086CPU结构与功能

## 2.2 微处理器的内部结构

### ③ 控制器

是CPU的“**指挥中心**”，完成指令的读入，寄存，译码和执行。由6部分组成：

程序计数器(PC : **Program Counter**)

指令寄存器(IR : **Instruction Register**)

指令译码器(ID : **Instruction Decoder**)

控制逻辑部件

处理机状态字PSW ( **Processor State Word** )

堆栈指示器(SP) **Stack Pointer**





## 2.2 微处理器的内部结构

### 程序计数器（PC）

- **用于保存下一条要执行的指令的地址**，即由它提供一个存储器地址，按此地址从对应存储器单元取出的内容，就是要执行的指令。
- 一般指令是顺序存放在存储器内的，所以程序计数器也叫**指令地址计数器**。由此可见，在程序执行过程中要实现程序的转移，就要改变程序计数器 PC 的内容。



## 2.2 微处理器的内部结构

### 指令寄存器（IR）

保存从存储器中读入的当前要执行的指令。

### 指令译码器（ID）

对指令寄存器中保存的指令进行译码分析。

### 控制逻辑部件

根据ID对指令的译码分析，发出相应的一系列的节拍脉冲和电位（控制信号），去完成指令的所有操作。



## 2.2 微处理器的内部结构

### 处理器状态字 (PSW)

暂存处理器当前的状态。PSW中的各位用来指示诸如算术运算结果的正/负，是否为零，是否有进位或借位，是否溢出等标志。条件转移指令将根据 PSW 中的某一位的状态决定程序是否转移。

### 堆栈指示器 (SP)

是在对按后进先出原则组织的称为堆栈的专用存储区进行操作时提供地址。堆栈用于子程序调用时保存返回地址和工作寄存器的内容。



## 2.2 微处理器的内部结构

### ④ I/O控制逻辑

包括 CPU 中与输入/输出操作有关的逻辑。  
其作用是处理输入/输出操作。

## 2.3 微处理器的功能结构

8086/8088是Intel公司生产的第三代微处理器芯片。其特点如下：

- ➡ 具有20条地址线，直接寻址能力达1MB。
- ➡ {
  - 8086有16条数据线，为16位微处理器。
  - 8088有8条数据线，为准16位微处理器。
- ➡ 片内总线和ALU均为16位，可进行8位和16位操作。



# 第2章 8086CPU结构与功能

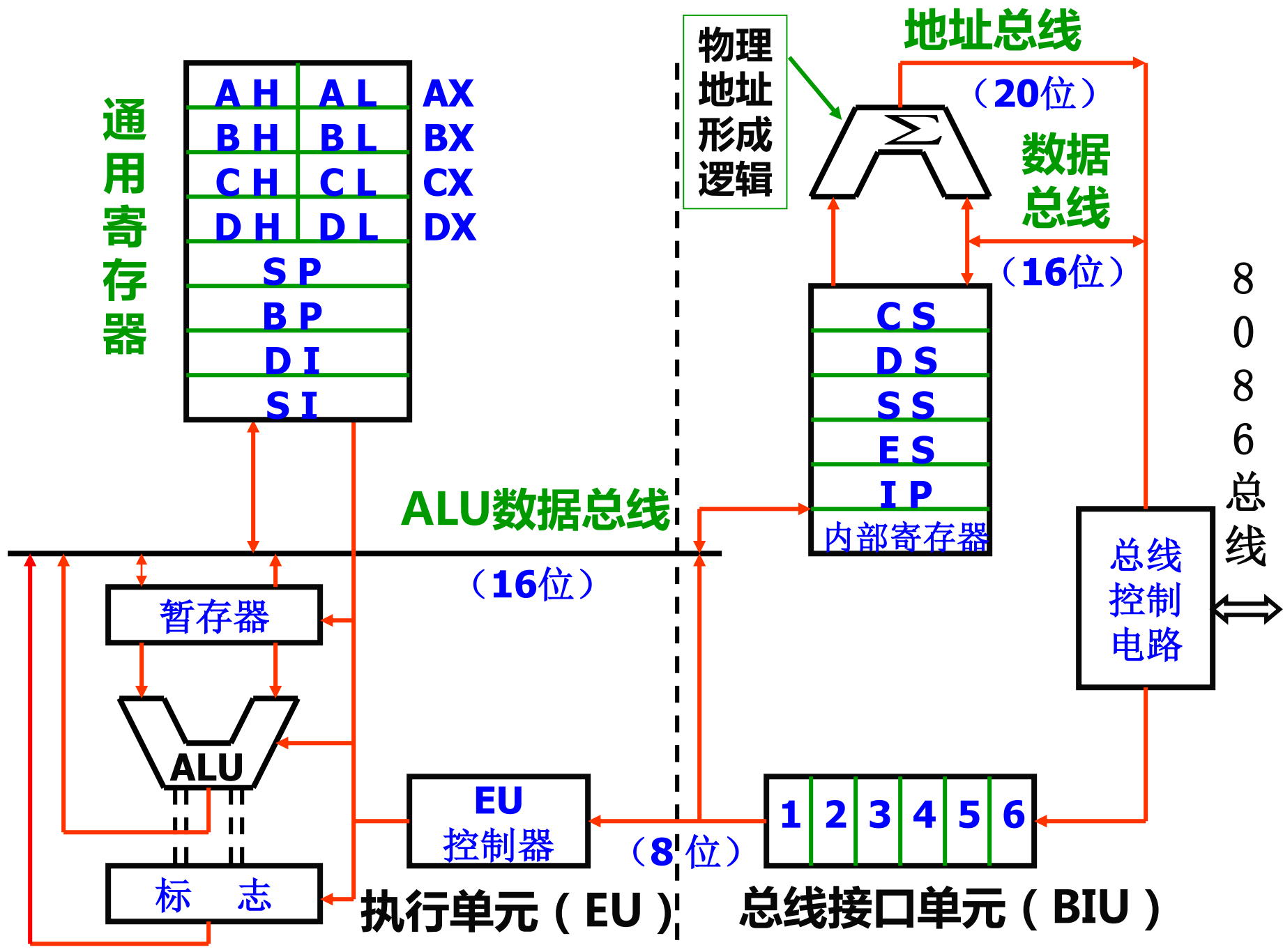
## 2.3 微处理器的功能结构

➡ 8086/8088均采用全新结构，片内均由两个独

立的逻辑单元组成：

- 总线接口单元(BIU)
- 执行单元(EU)

8086/8088CPU内部结构如图所示



## 2.3 微处理器的功能结构

### 一.总线接口单元 (BIU)

段寄存器 (CS, DS, ES, SS)

指令指针寄存器 (IP)

指令队列寄存器 :  $\left\{ \begin{array}{l} 8086\text{CPU} : 6\text{字节} \\ 8088\text{CPU} : 4\text{字节} \end{array} \right.$

地址形成逻辑

总线控制逻辑





## 2.3 微处理器的功能结构

- ◆ BIU主要负责从存储器指定区域取出指令，并将取出的指令送指令队列寄存器中排队；
- ◆ 当EU执行的指令需要和外部存储器或者I/O端口之间进行数据传送时，BIU就停止取指令，为EU服务，完成这次总线操作；
- ◆ 当指令队列满时，BIU也停止取指令的操作。



# 第2章 8086CPU结构与功能

## 2.3 微处理器的功能结构

### 二. 执行单元 ( EU )

- 通用寄存器 ( 8个 )
- 标志寄存器 ( FLAG ) (即PSW)
- 算术/逻辑运算单元 ( ALU )
- EU控制器

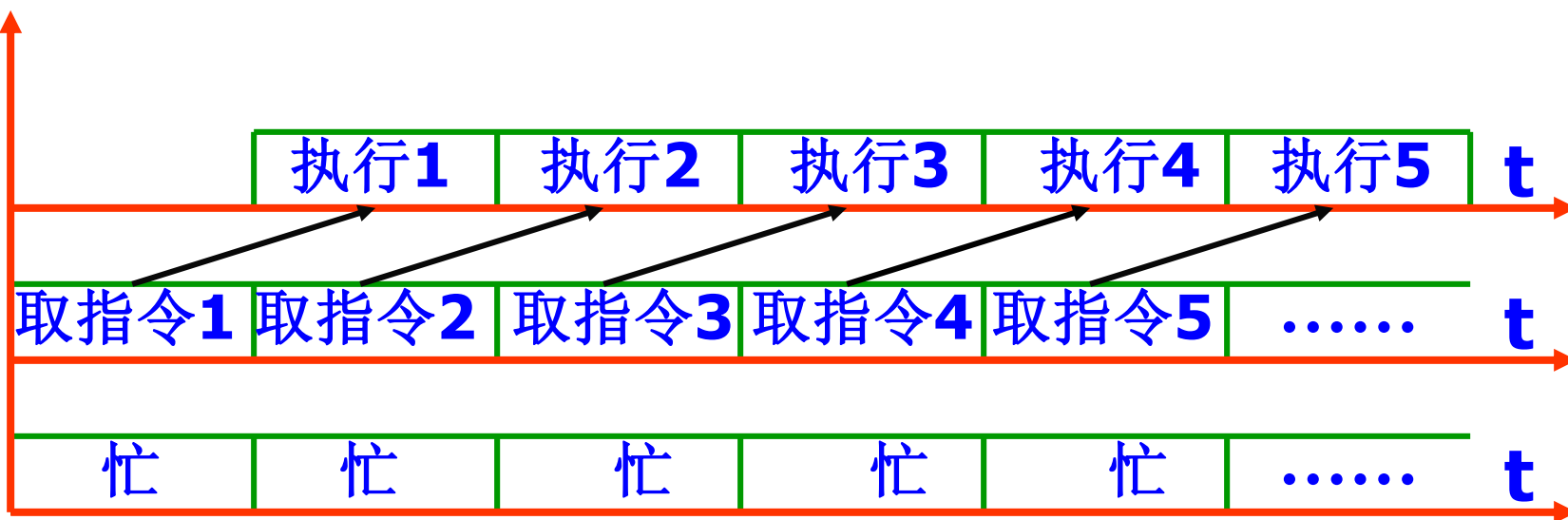
► EU主要负责从指令队列寄存器中获取指令，并对指令加以执行，完成指令所规定的操作。同时也负责算术/逻辑运算以及进行内存有效地址的计算等。



# 第2章 8086CPU结构与功能

## 2.3 微处理器的功能结构

► CPU程序执行过程：



8086/8088CPU执行程序的过程



## 2.3 微处理器的功能结构

EU和BIU单元执行过程中，应该满足规则：

1. 当指令队列寄存器中无指令时，EU处于等待状态；
2. 当指令队列中存满指令，而EU又没有访问存储器和I/O端口的需要，则BIU进入空闲状态；
3. 当指令队列中有两个空闲字节，则BIU自动执行取指令的总线周期；
4. 在EU执行指令时，需要访问存储器或I/O端口，如果这时BIU正在取指令，则应等待BIU完成取指令周期，然后BIU进入存储器和I/O端口访问周期；
5. 在EU执行转移、子程序调用或返回等指令时，自动清除指令队列的内容。

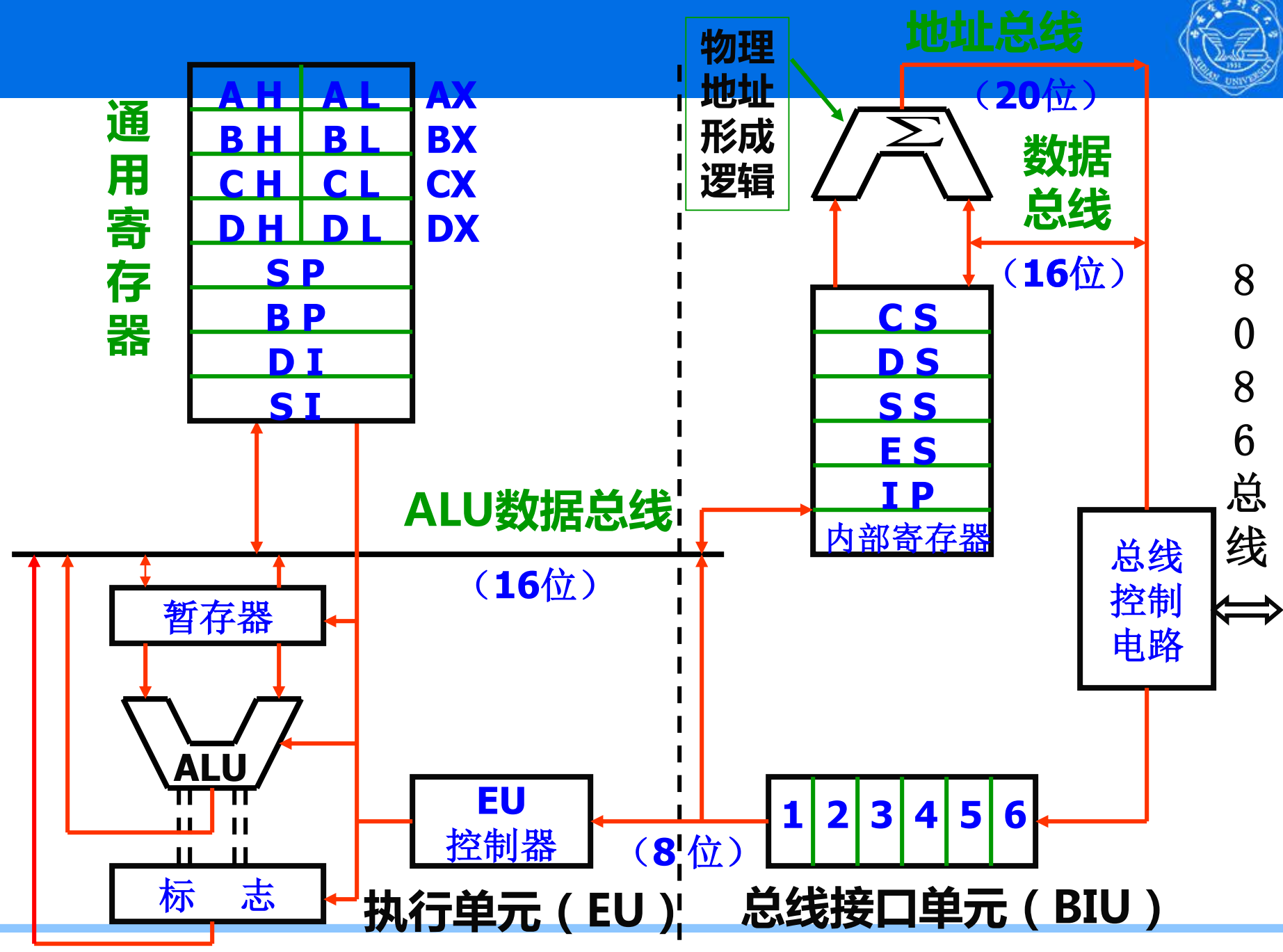
## 第2章 8086CPU结构与功能



8086/8088 CPU内部共有14个16位寄存器，用于提供运算，控制指令执行和对指令及操作数寻址。

14个寄存器按其用途可分为三大类：

- 通用寄存器（8个）
- 段寄存器（4个）
- 控制寄存器（2个）





## 2.4 微处理器的寄存器组织

### 一. 通用寄存器（8个）

8个16位通用寄存器分为两组：

- 数据寄存器（4个）
- 地址指针和变址寄存器（4个）



# 第2章 8086CPU结构与功能

## 2.4 微处理器的寄存器组织

### 1. 数据寄存器（4个）

AH	AL	AX
BH	BL	BX
CH	CL	CX
DH	DL	DX
8位		16位

累加器 **Accumulator**

基址寄存器（地址寄存器）  
**Base Register**

计数器 **Count Register**

数据寄存器 **Data Register**

存放数据



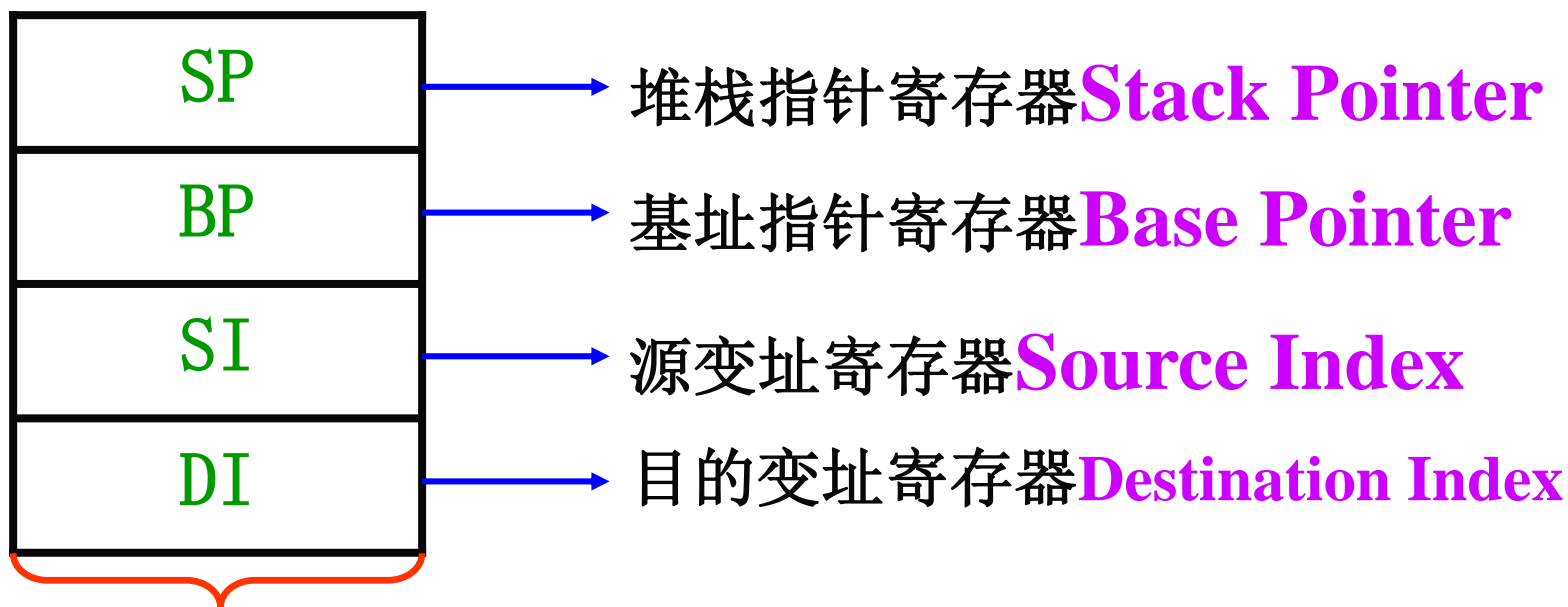


# 第2章 8086CPU结构与功能

## 2.4 微处理器的寄存器组织

### 2. 地址指针和变址寄存器（4个）

均为地址寄存器



均为16位，也能存放数据



# 第2章 8086CPU结构与功能

## 2.4 微处理器的寄存器组织

### 二. 段寄存器

在微机系统的内存中通常存放着三类信息：

- 代码（指令码）—— 指示CPU执行何种操作。
- 数据（数值、字符等）—— 程序处理的对象或结果。
- 堆栈信息 —— 被保存的返回地址和中间结果等。

代码段 数据段 堆栈段

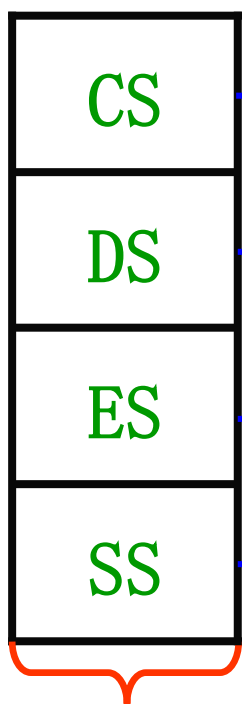


# 第2章 8086CPU结构与功能

## 2.4 微处理器的寄存器组织

8086 / 8088 CPU有4个段寄存器。

分别是：



CS

代码段寄存器。指向当前的代码段，指令由此段取出。Code Segment

DS

数据段寄存器。指向当前的数据段。Data Segment

ES

附加数据段寄存器。指向当前的附加数据段。Extra Segment

SS

堆栈段寄存器。指向当前的堆栈段。Stack Segment

16位



# 第2章 8086CPU结构与功能

## 2.4 微处理器的寄存器组织

### 三. 控制寄存器（2个）

#### 1. 指令指针寄存器(IP:Instruction Pointer )(16位)

指令指针寄存器相当于一般微处理器中的程序计数器(PC:Program Counter )。

➡ 它始终指向CPU下一条要取指令所在存储器单元的偏移地址(段地址由CS提供)。

➡ 用户不能更改IP的值,只有CPU执行转移指令,子程序调用指令和子程序返回指令以及中断处理时,IP才作相应的改变。



# 第2章 8086CPU结构与功能

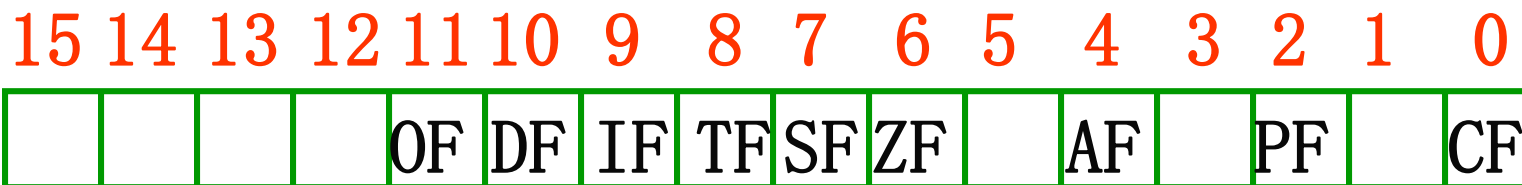
## 2.4 微处理器的寄存器组织

### 2.标志寄存器(FLAG)

标志寄存器相当于一般微处理器中的程序状态字寄存器(PSW)。16位,但有用的只有9位,其中:

{ 状态标志: CF, PF, AF, ZF, SF, OF, 共6位  
控制标志: TF, IF, DF, 共3位

如下图所示:





# 第2章 8086CPU结构与功能

## 2.4 微处理器的寄存器组织

### 状态标志

→ 反映的是ALU运算后结果的状态

- **CF(Carry Flag)进位标志**

如果加法时最高位(对字节操作是D7位, 对字操作是D15位)产生进位或减法时最高位产生借位, 则 $CF=1$ , 否则 $CF=0$ 。

- **AF(Auxiliary Carry Flag)辅助进位标志**

如果在加法时D3位有进位或减法时D3位有借位, 则 $AF=1$ , 否则 $AF=0$ 。这个标志位用于实现BCD码算术运算结果的调整。



## 2.4 微处理器的寄存器组织

- **ZF(Zero Flag)零标志位**

如果运算结果各位都为零，则 $ZF=1$ ，否则 $ZF=0$ 。

- **SF(Sign Flag)符号标志**

它总是和结果的最高位(字节操作时是D7，字操作时是D15)相同，因为在补码运算时最高位是符号位，所以运算结果为负时， $SF=1$ ，否则 $SF=0$ 。



## 2.4 微处理器的寄存器组织

- **OF(Overflow Flag)溢出标志**

在加或减运算中结果超出8位或者16位有符号数所能表示的数值范围(-128 ~ +127或-32768 ~ +32767)时，产生溢出，OF=1，否则OF=0。

- **PF(Parity Flag)奇偶标志**

如果操作结果的**低8位中含有偶数个1**，PF=1，否则PF=0。



# 第2章 8086CPU结构与功能



## 2.4 微处理器的寄存器组织

例1: 若CPU执行5439H+476AH加法运算指令:

$$\begin{array}{r} \phantom{+} \phantom{0101} \phantom{0100} \phantom{0011} \phantom{1001} \text{B} \\ + \phantom{0101} \phantom{0100} \phantom{0011} \phantom{1001} \text{B} \\ \hline 1001 \phantom{0100} \phantom{0011} \phantom{1001} \phantom{0011} \text{B} \end{array}$$

那么, 指令执行后有:

SF=1, ZF=0, PF=1, AF=1, CF=0, OF=1

# 第2章 8086CPU结构与功能



## 2.4 微处理器的寄存器组织

例2: 若CPU执行543AH-FE00H减法运算指令;

$$\begin{array}{r} 0101 \ 0100 \ 0011 \ 1010\text{B} \\ - 1111 \ 1110 \ 0000 \ 0000\text{B} \\ \hline 0101 \ 0110 \ 0011 \ 1010\text{B} \end{array}$$

那么, 指令执行后有:

SF=0, ZF=0, PF=1, AF=0, CF=1, OF=0



# 第2章 8086CPU结构与功能

## 2.4 微处理器的寄存器组织

### 控制标志

→ 用来控制CPU的操作特征（运行状态）

- **DF(Direction Flag) 方向控制标志**

可由指令置1/清0 { STD; DF=1  
CLD; DF=0

在进行字符串操作时, CPU每执行一条串操作指令, 对源或(与)目的操作数的地址会自动进行一次调整, 其调整准则为

DF= { 0, 自动递增。  
1, 自动递减。



# 第2章 8086CPU结构与功能

## 2.4 微处理器的寄存器组织

- IF(Interrupt Enable Flag)

外部可屏蔽中断允许标志。

可由指令置1/清0:

{ STI; IF=1, CPU处于开中断状态。  
CLI; IF=0, CPU处于关中断状态。

当IF= { 1时, CPU能响应外部可屏蔽中断请求;  
0时, CPU不能响应外部可屏蔽中断请求。

**IF对外部非可屏蔽中断请求以及CPU内部的中断不起作用。**



# 第2章 8086CPU结构与功能

## 2.4 微处理器的寄存器组织

- **TF(Trap Flag)陷阱标志**

没有专门的置1/清0指令。

当 $TF=1$ 时，CPU每执行完一条指令便自动产生一个内部中断(类型为1)，转去执行一个中断服务程序，**可以借助中断服务程序来检查每条指令执行的情况，称为单步工作方式**，常用于程序的调试。



## 2.5 微处理器的存储器和I/O组织

### 一. 存储器地址空间和数据存储格式

- ◆ 8086/8088的存储器都是以字节(8位)为单位组织的。
- ◆ 有20条地址总线,  $2^{20}$  字节 ( 1MB ) 。
- ◆ 每个字节对应一个唯一的地址,
- ◆ 地址范围为  $0 \sim 2^{20} - 1$   
( 用16进制表示为 00000 ~ FFFFFH ), 如图所示。

## 第2章 8086CPU结构与功能

## 2.5 微处理器的存储器与I/O组织

[illegible]



# 第2章 8086CPU结构与功能

## 2.5 微处理器的存储器与I/O组织

- 存储器内两个连续的字节，定义为一个**字**，低字节（低8位）存放在低地址中，高字节（高8位）存放在高地址中。
- 各位的编号方法是最低位（LSB）为位0，一个字节中，最高位（MAS）编号为位7；一个字中最高位的编号为位15。

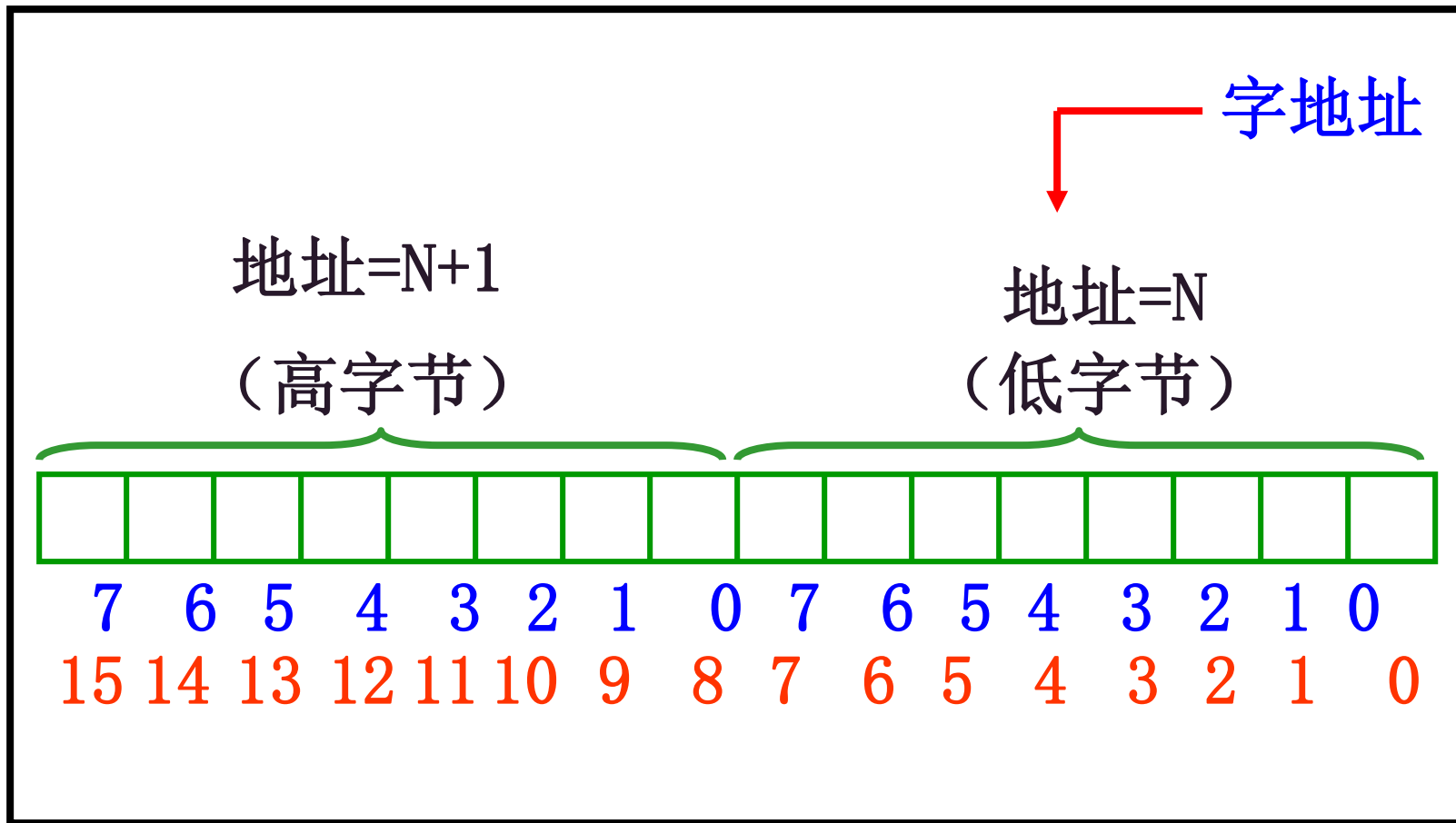
这些约定如图所示





# 第2章 8086CPU结构与功能

## 2.5 微处理器的存储器与I/O组织

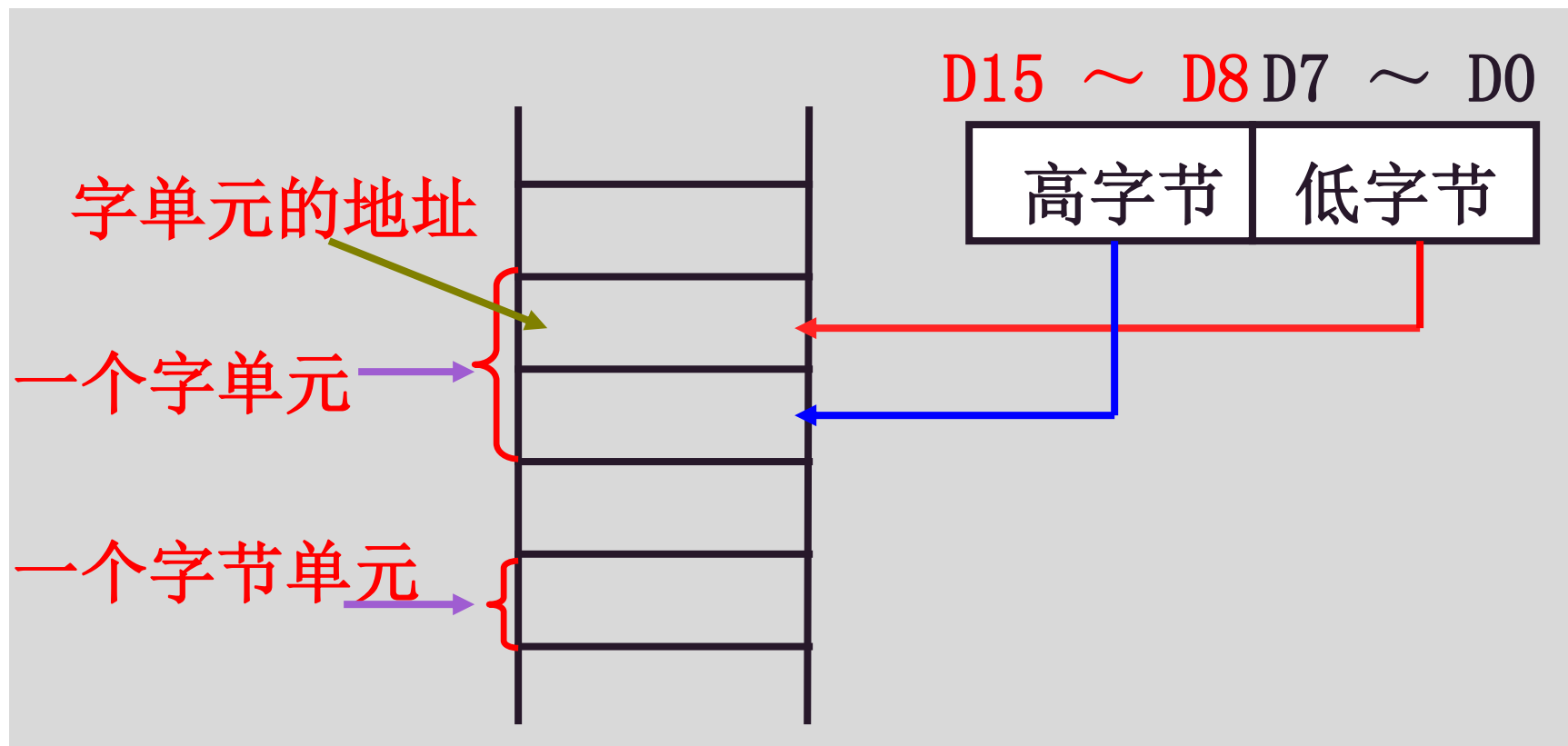




# 第2章 8086CPU结构与功能

## 2.5 微处理器的存储器与I/O组织

字数据在存储器中存放的格式如图所示



字数据在存储器中存放格式示意图



# 第2章 8086CPU结构与功能

## 2.5 微处理器的存储器与I/O组织

### 二. 存储器的分段和物理地址的形式

#### 1. 为什么要分段

从前面的介绍可知，

- 8086/8088CPU有20条地址线（A19~A0），能寻址外部存储空间为1MB；
- 在8088/8086CPU内部能**向存储器提供地址码的地址寄存器有六个**，均为16位，所以用这六个16位地址寄存器任意一个给外部存储器提供地址，只能提供64K个地址；
- 对1MB地址寻址不完；
- 这六个16位地址寄存器分别为：



# 第2章 8086CPU结构与功能

## 2.5 微处理器的存储器与I/O组织

BX	→	基址寄存器
BP	→	基址指针寄存器
SI	→	源变址寄存器
DI	→	目的变址寄存器
SP	→	堆栈指针寄存器
IP	→	指令指针寄存器

为了使8088/8086CPU能寻址到外部存储器1MB空间中任何一个单元，8088/8086采用了地址分段方法（将1MB空间分成若干个逻辑段），从而将寻址范围扩大到了1MB。



## 2.5 微处理器的存储器与I/O组织

### 2. 物理地址(PA)的形成

段地址和段内偏移地址都是无符号的16位二进制数，常用4位十六进制数表示。这种方法表示的存储器单元的地址称为逻辑地址。如下图所示。

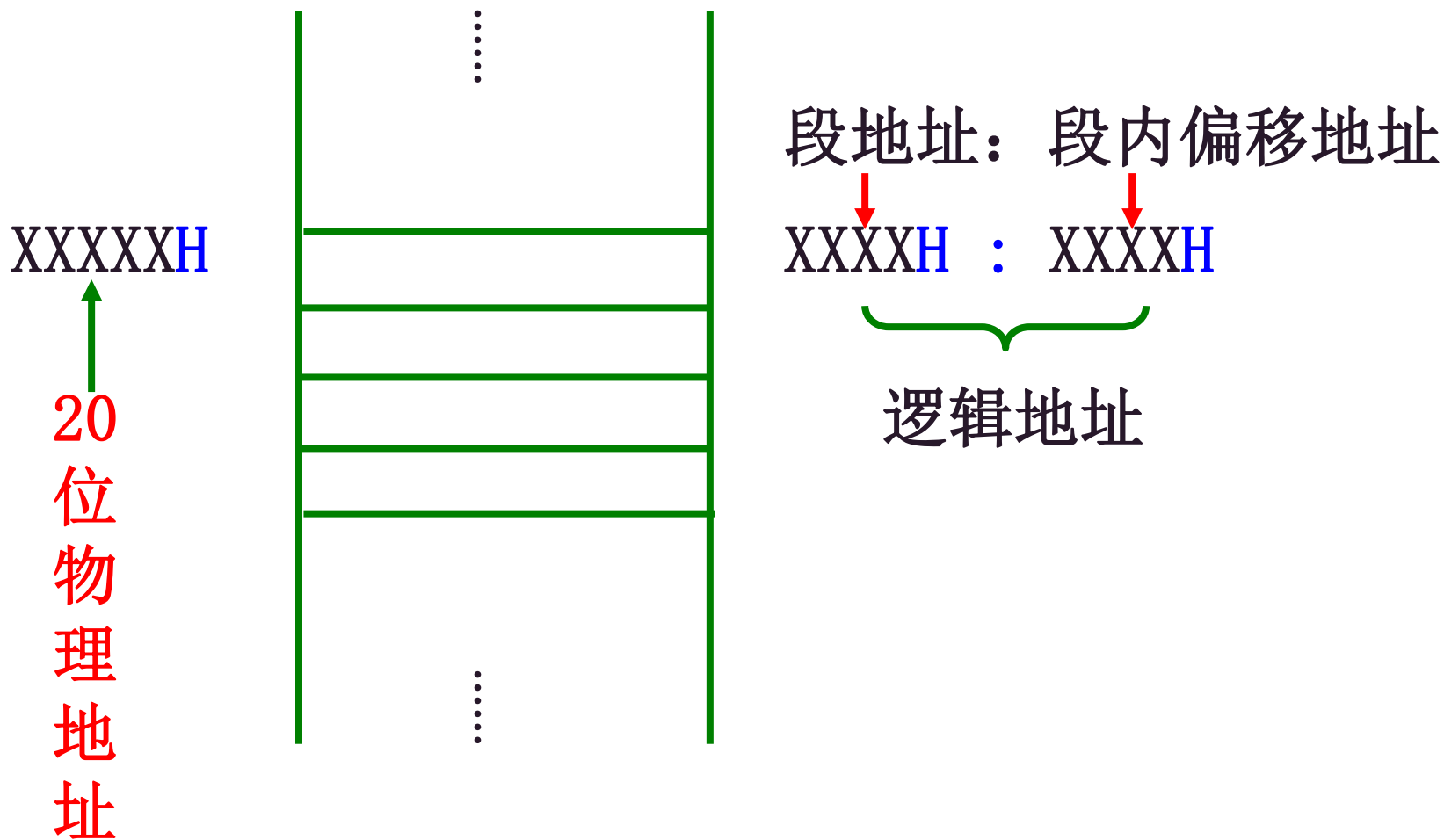
逻辑地址的表示格式为：

段地址：偏移地址



# 第2章 8086CPU结构与功能

## 2.5 微处理器的存储器与I/O组织





# 第2章 8086CPU结构与功能

## 2.5 微处理器的存储器与I/O组织

➡ 一个存储单元用逻辑地址表示后，CPU 对该单元的寻址就应提供两部分地址

┌ 段地址  
└ 段内有效地址

其中段地址由段寄存器提供：

- CS —— 提供当前**代码（程序）段**的段地址
- DS —— 提供当前**数据段**的段地址
- ES —— 提供当前**附加数据段**的段地址
- SS —— 提供当前**堆栈段**的段地址

# 第2章 8086CPU结构与功能



## 2.5 微处理器的存储器与I/O组织

②段内偏移地址由下列地址寄存器提供：

BX  
BP  
SI  
DI

→ CPU 对存储器进行数据读/写操作时，由这些寄存器以某种寻址方式向存储器提供段内偏移地址。

SP → 堆栈操作时，提供堆栈段的段内偏移地址

IP → CPU 取指令时，由IP提供所取指令代码所在单元的偏移地址。



# 第2章 8086CPU结构与功能



## 2.5 微处理器的存储器与I/O组织

➡ 已知某存储单元的逻辑地址, 怎样求该单元的  
物理地址PA:

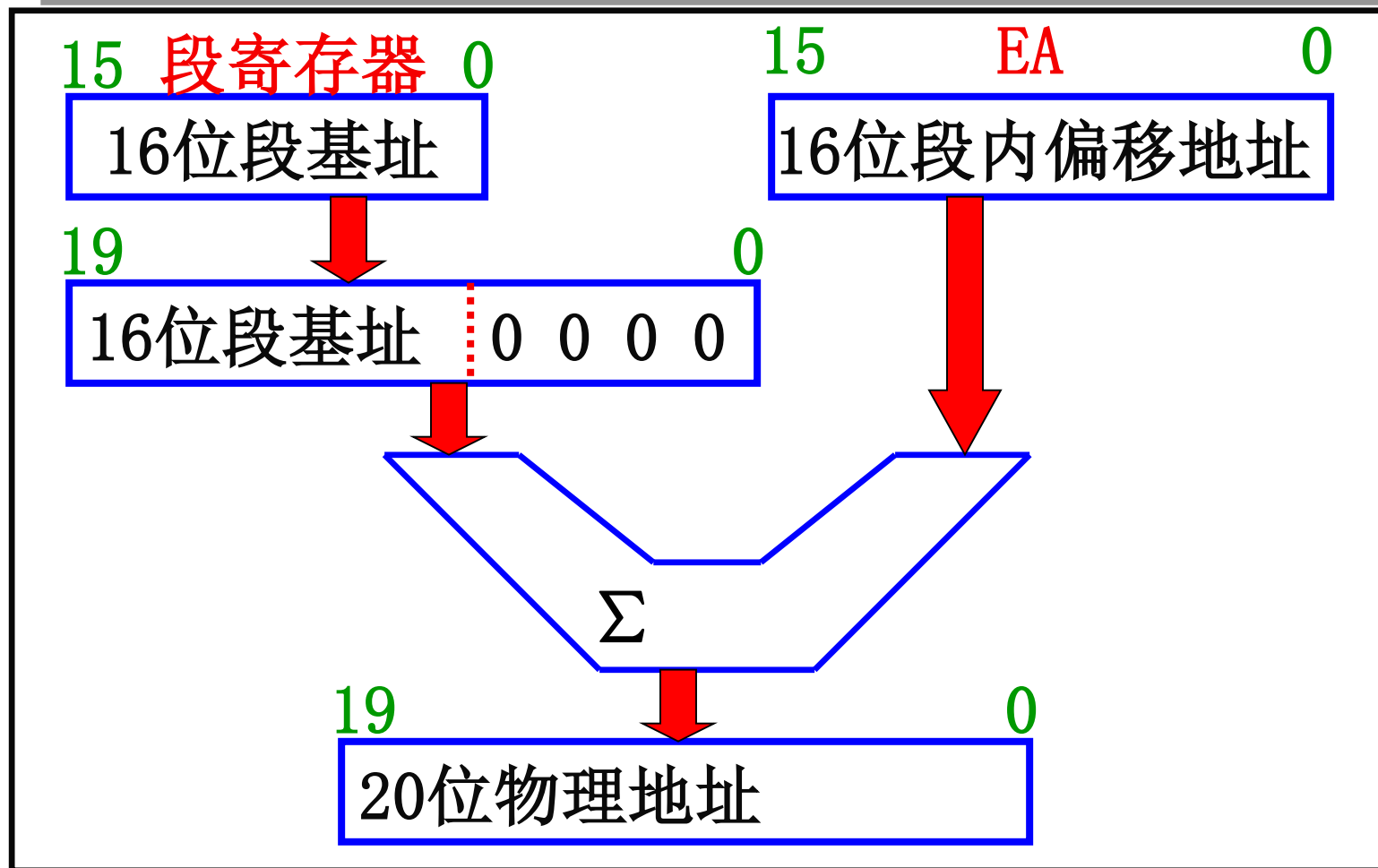
物理地址=段地址 ~~×~~10H+段内偏移地址

8086/8088 CPU中的BIU单元用来完成物理地址的计算, 其计算方法如图所示。



# 第2章 8086CPU结构与功能

## 2.5 微处理器的存储器与I/O组织



物理地址的形成



# 第2章 8086CPU结构与功能

## 2.5 微处理器的存储器与I/O组织

**例1：**某单元的逻辑地址为4B09H:5678H，则该存储单元的物理地址为：

$$\text{物理地址 (PA)} = \text{段地址} \times \cancel{10\text{H}} + \text{EA}$$

$$= 4\text{B}09\text{H} \times \cancel{10\text{H}} + 5678\text{H}$$

$$= 4\text{B}090\text{H} + 5678\text{H}$$

$$= 50708\text{H}$$



# 第2章 8086CPU结构与功能

## 2.5 微处理器的存储器与I/O组织

➡ 在访问存储器时，段地址总是由段寄存器提供的。8086/8088微处理器中有4个段寄存器（CS、DS、SS、ES），所以CPU可以通过这4个段寄存器来访问4个不同的段。



# 第2章 8086CPU结构与功能

## 2.5 微处理器的存储器与I/O组织

### 三. 信息的分段存储与段寄存器的关系

➡ 用8086/8088汇编语言编写程序时，要把程序中的不同信息安排在不同的段，用户源程序汇编后在存储器中存放是按照不同的信息放在不同的逻辑段。

{ 程序（代码）信息  
数据信息  
堆栈信息

# 第2章 8086CPU结构与功能



## 2.5 微处理器的存储器与I/O组织

**代码信息** —→ 存放在代码段，其地址由CS:IP提供。

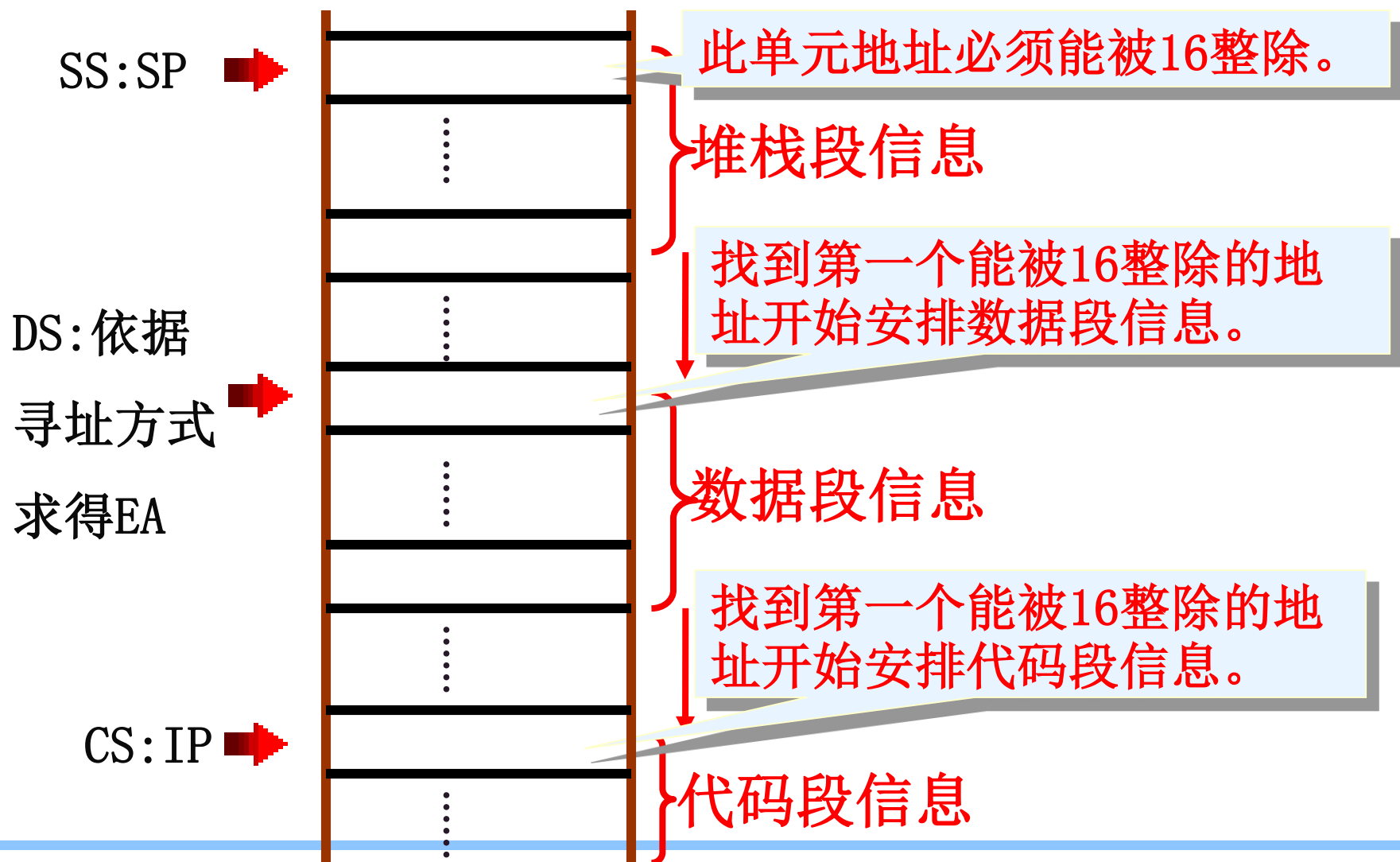
**堆栈信息** —→ 存放在堆栈段，其地址由SS:SP提供。

**数据信息** —→ 通常存放在数据段（段地址由DS提供）；也可以存放在附加数据段（段地址由ES提供），其段内偏移地址依据寻址方式的不同来求得。



# 第2章 8086CPU结构与功能

## 2.5 微处理器的存储器与I/O组织





# 第2章 8086CPU结构与功能

## 2.5 微处理器的存储器与I/O组织

### 3 . 分段讨论

- ➡ 1MB的存贮空间中,每个存贮单元的实际地址编码称为该单元的物理地址(用PA表示)。
- ➡ 把1MB的存贮空间划分成若干个逻辑段,每段最多64KB。
- ➡ 各逻辑段的起始地址必须能被16整除,即一个段的起始地址(20位物理地址)的低4位二进制码必须是0。





# 第2章 8086CPU结构与功能

## 2.5 微处理器的存储器与I/O组织

- ➡ 一个段的起始地址的高16位自然数为该段的段地址.显然,在1MB的存贮空间中,可以有  $2^{16}$  个段地址.每个相邻的两个段地址之间相隔16个存贮单元。
- ➡ 在一个段内的每个存贮单元,可以用相对于本段的起始地址的偏移量来表示,这个偏移量称为**段内偏移地址**,也称为**有效地址(EA)**。
- ➡ 段内偏移地址也用16位二进制编码表示.所以,在一个段内有  $2^{16} = 64\text{K}$  个偏移地址(即一个段最大为64KB)。



# 第2章 8086CPU结构与功能

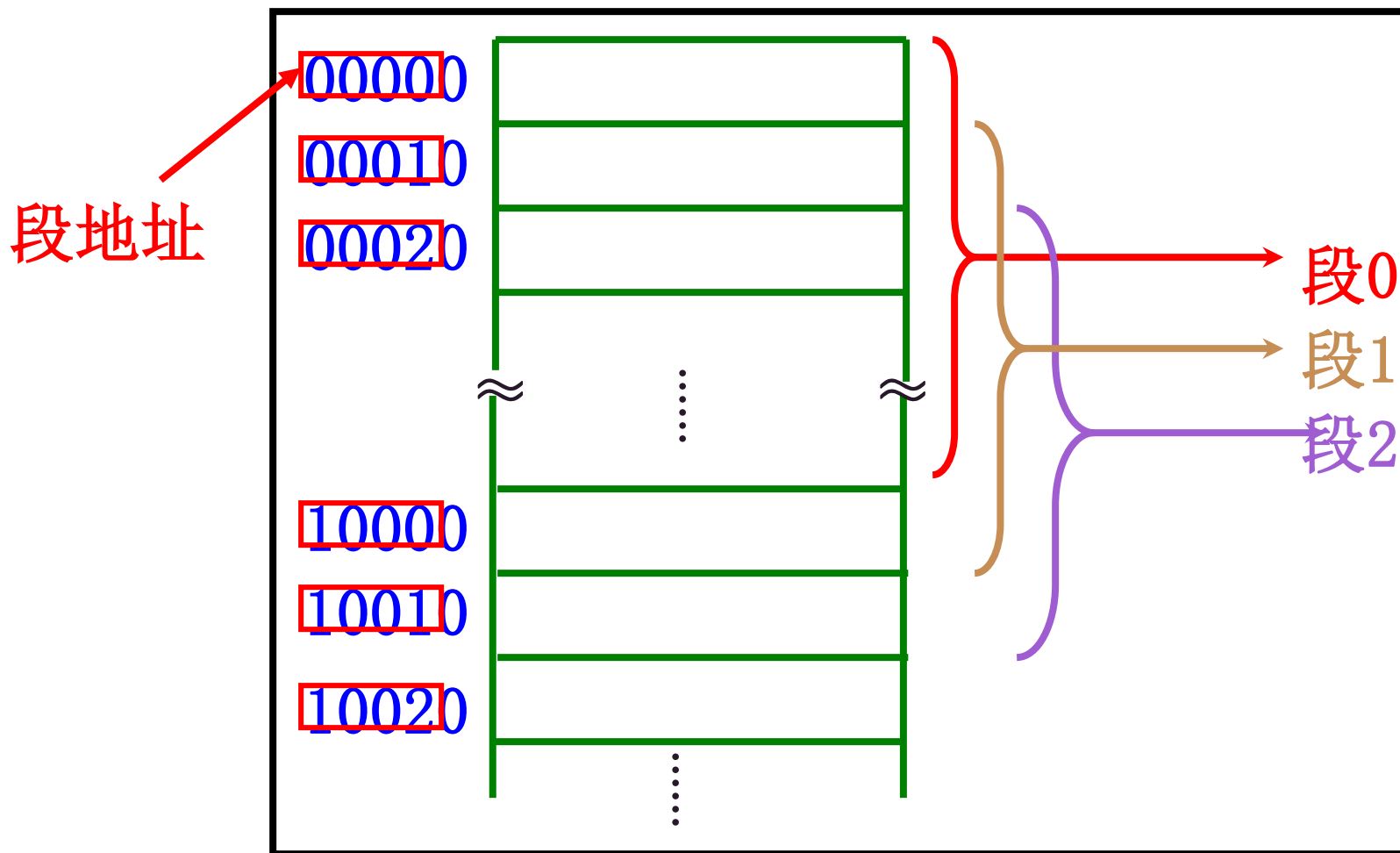
## 2.5 微处理器的存储器与I/O组织

- ➡ 在一个64KB的段内，每个偏移地址单元的段地址是相同的.所以段地址也称为段基址。
- ➡ 由于相邻两个段地址只相隔16个单元，所以段与段之间大部分空间互相覆盖(重叠)。
- ➡ 存储器段的划分与段的覆盖示意图如下图所示。



# 第2章 8086CPU结构与功能

## 2.5 微处理器的存储器与I/O组织



存储器段的划分与段的覆盖示意图



# 第2章 8086CPU结构与功能

## 2.5 微处理器的存储器与I/O组织

- I/O设备包括与外界通信和存储大容量信息用的各种外部设备。由于这些外部设备的复杂性和多样性，特别是速度比CPU低得多，因此I/O设备不能直接和总线相连接。
- I/O接口是保证信息和数据在CPU和I/O设备之间正常传送的电路。



# 第2章 8086CPU结构与功能

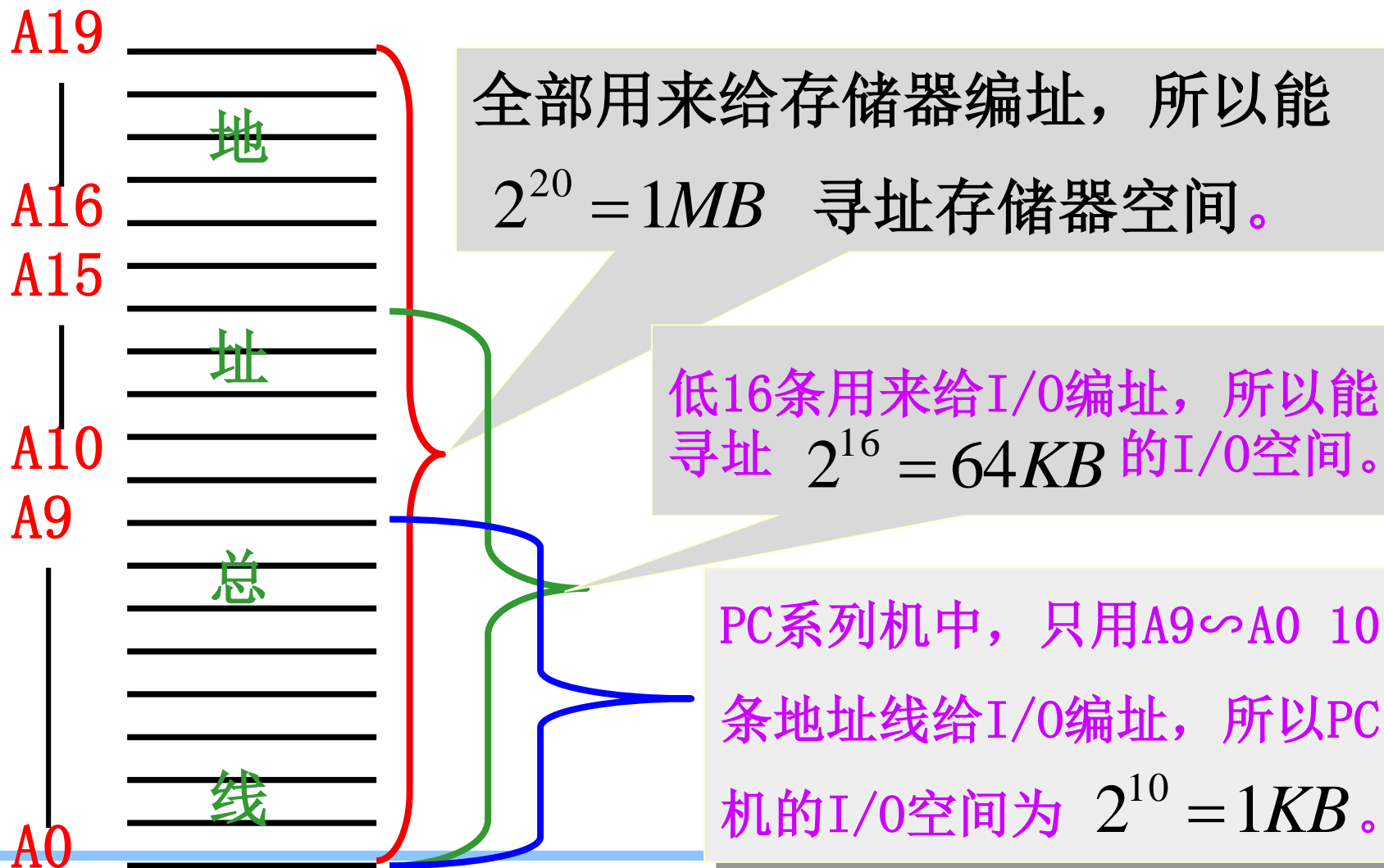
## 2.5 微处理器的存储器与I/O组织

- I/O接口与CPU之间的通信是利用称为I/O端口的寄存器来完成的，一个I/O端口有一个唯一的I/O地址与之对应。
- 8086/8088CPU共有20条地址线，对存储器和I/O端口的寻址采用独立编址的方式。



# 第2章 8086CPU结构与功能

## 2.5 微处理器的存储器与I/O组织



# 第2章 8086CPU结构与功能



## 作业