第2章 微处理器功能结构

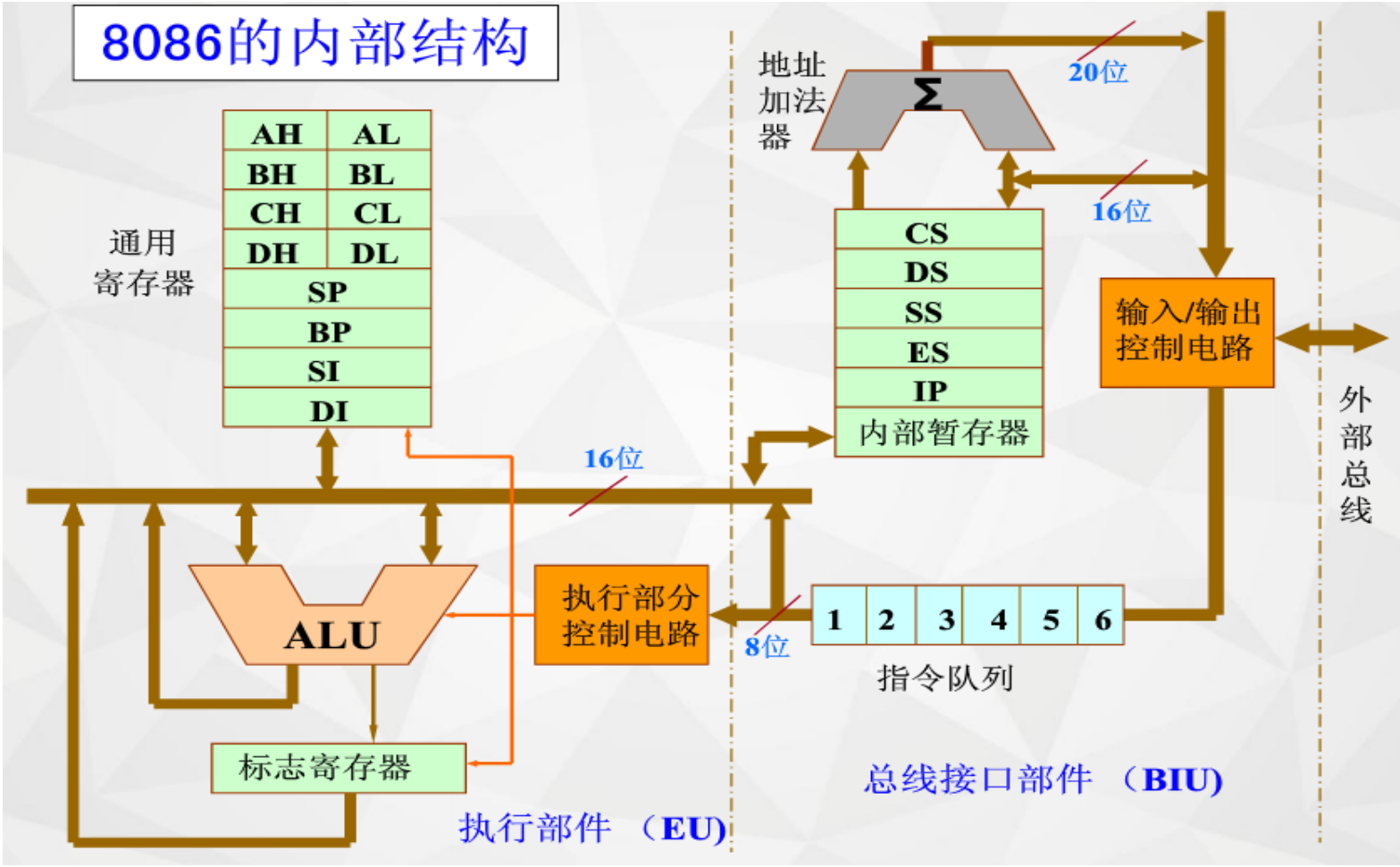
- 8086处理器
  - 微处理器的基本结构
  - 8086的功能结构
  - 8086的寄存器结构
- 8086存储器组织
  - 物理地址的确定
  - 段寄存器的使用

2.1 8086微处理器内部结构

- 总线接口单元 **BIU**
- 执行单元 **EU**
- 指令的执行过程

8086内部结构

分两部分：总线接口单元BIU（Bus Interface Unit）、执行单元EU（Execute Unit）



外部总线

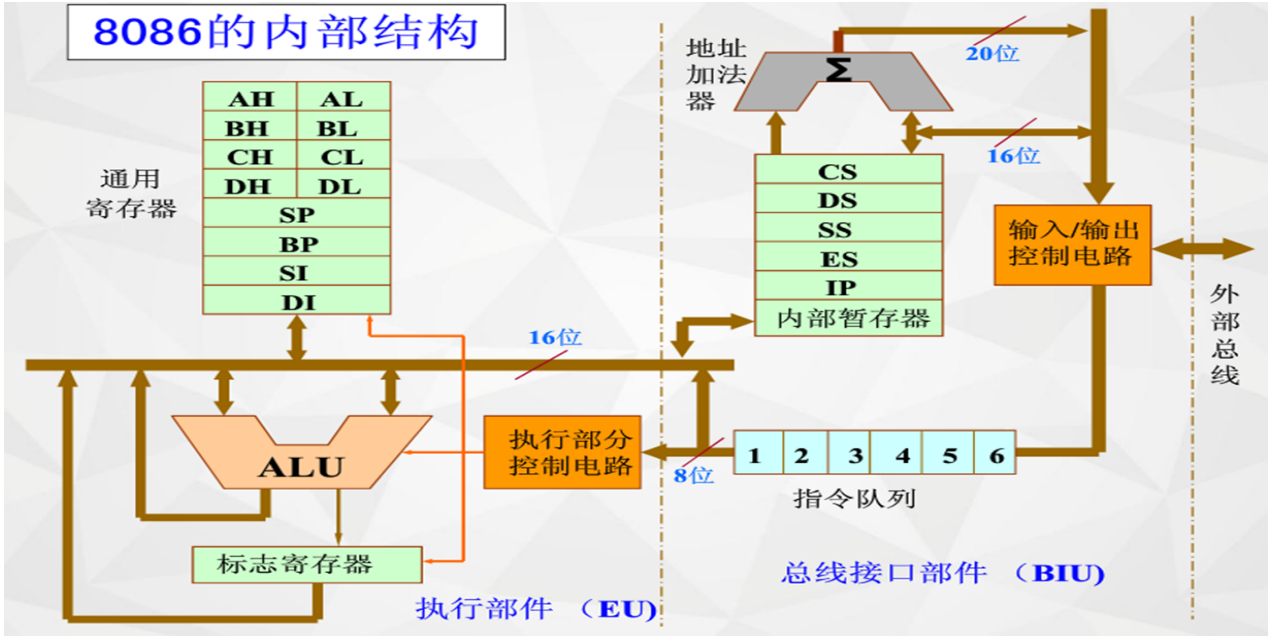
- 8086

  16bit数据总线和20bit地址总线，可寻址1M字节空间

- 8088

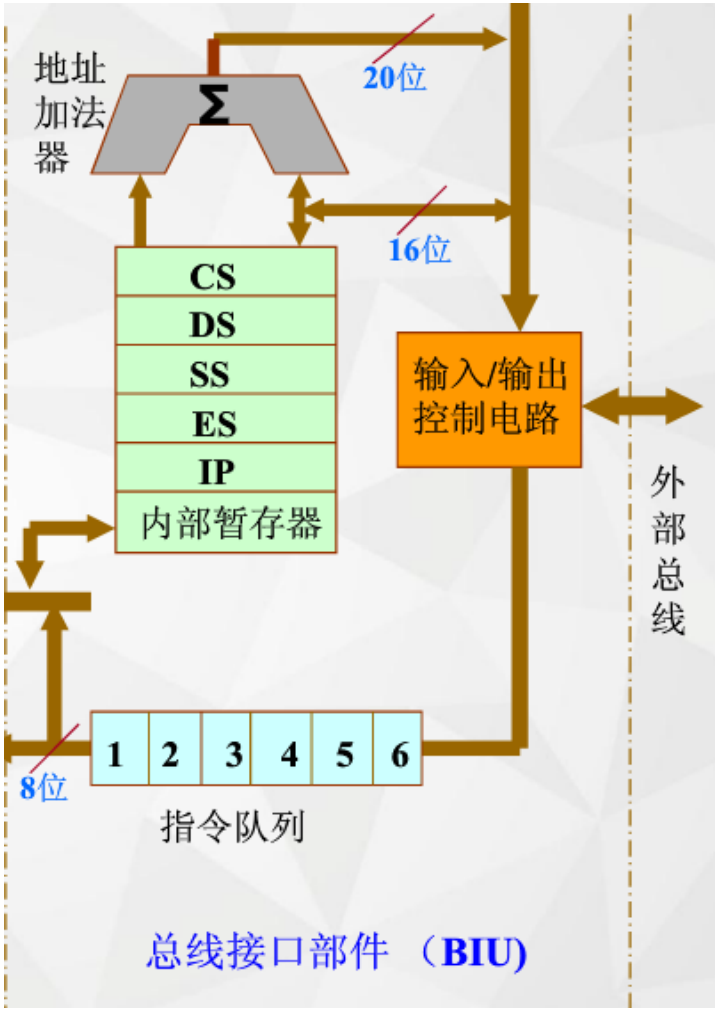
  8bit数据总线和20bit地址总线，可寻址1M字节空间

  但内部数据总线仍然是16bit



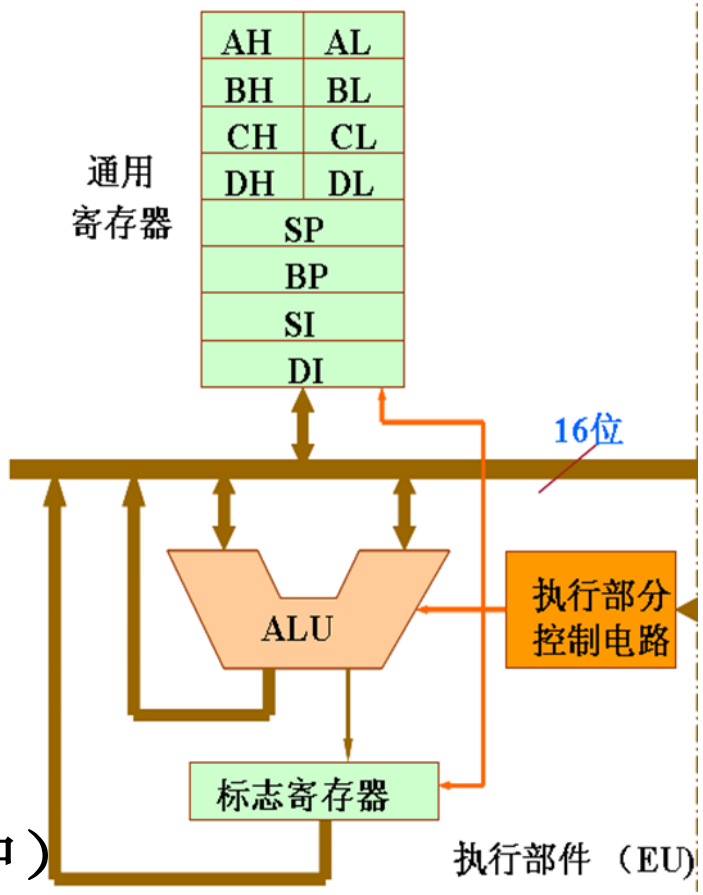
总线接口单元 BIU

- 组成：
  - 段寄存器 CS、DS、SS、ES
  - IP指令指针寄存器
  - 地址加法器Σ
  - 总线控制器
  - 指令队列(先进先出FIFO)
- 功能：
  - 执行所有总线操作（取指、取数）
  - 通过运算得到20位的物理地址
  - 预取指令
    - 队列长度：6个字节
    - 预取方法：闲两个字节就预取

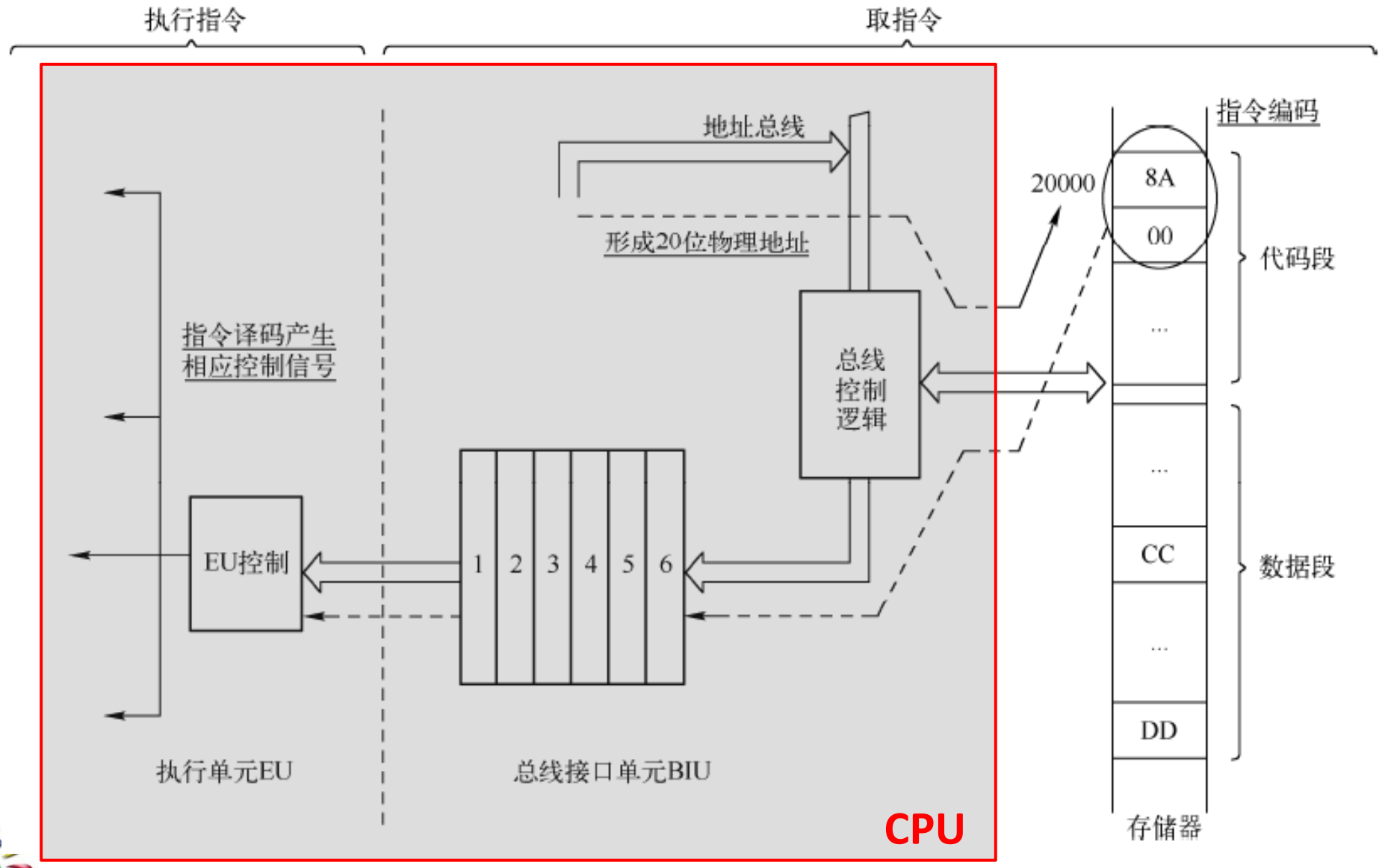


执行单元 EU

- 组成：
  - 8个通用寄存器 AX、BX、CX、DX、SP、BP、DI、SI
  - 运算器ALU
  - 指令操作控制电路
  - 1个状态标志寄存器
- 功能：（执行指令）
  - 指令译码
  - 指令执行
  - 暂存中间运算结果（在通用寄存器中）
  - 保存运算结果特征（在标志寄存器FLAGS中）



8086一条指令的执行过程



并行操作方式

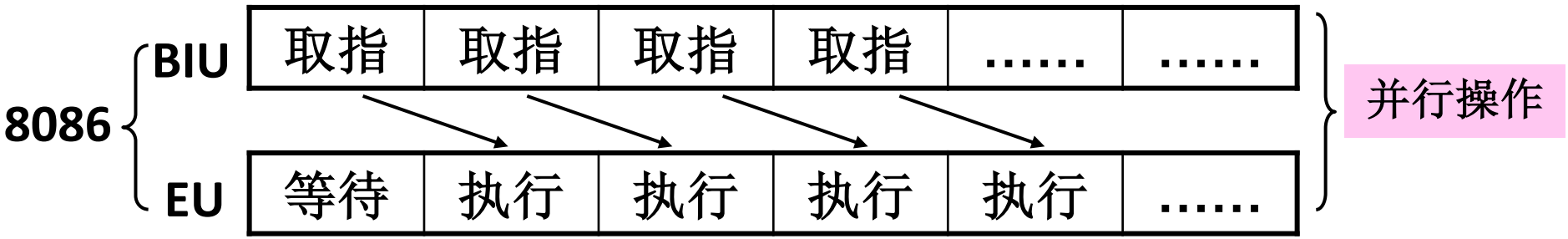
- 一条指令的执行过程（注：不同指令执行过程不同）

1　　2　　3　　4

取指、译码、执行、存结果

占用总线的操作①、④

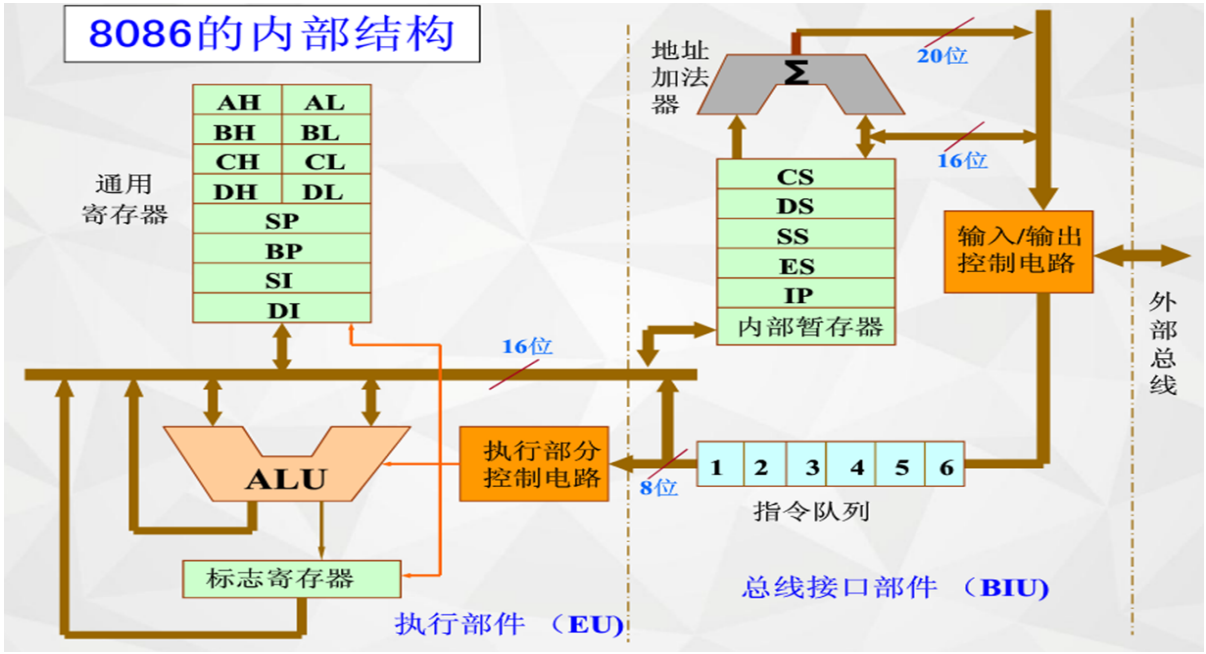
- **8086CPU采用并行执行方式，将指令执行的各步骤分配给CPU内的两个独立部件（EU和BIU）完成**



各个部件等待时间减少，执行效率大大提高！

结论

- 指令预取队列的存在使EU和BIU两个部分可同时进行工作。
  - 提高了CPU的效率
  - 降低了对存储器存取速度的要求



指令执行过程举例

ADD AX，BX； AX←AX+BX

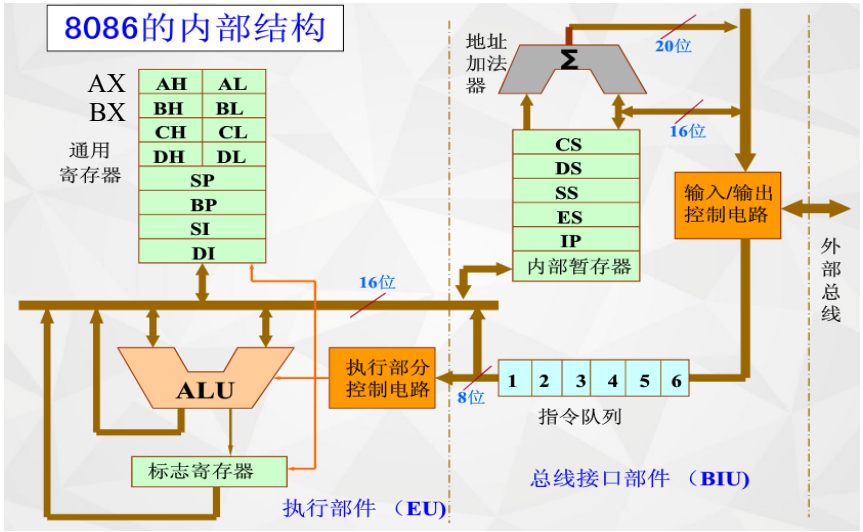
1、取指令 BIU用CS、IP内容生成20bit地址，从内存取出ADD指令送指令队列

2、执行指令

译码：EU控制器发出控制信号，将AX，BX寄存器的内容送到ALU的两个输入端

执行：ALU执行加法运算，置标志寄存器的相关位

写回：EU控制器发出控制信号，将加法结果送入AX



思考？

- **8086内部结构按功能可以分为（　）个部分，分别称为（　　）。**

- **20根地址线所能寻址的存储器地址范围是00000H~FFFFFH，14根地址线所能寻址的存储器地址范围是多少？**

  0000H~3FFFH

2.2 8086的寄存器结构

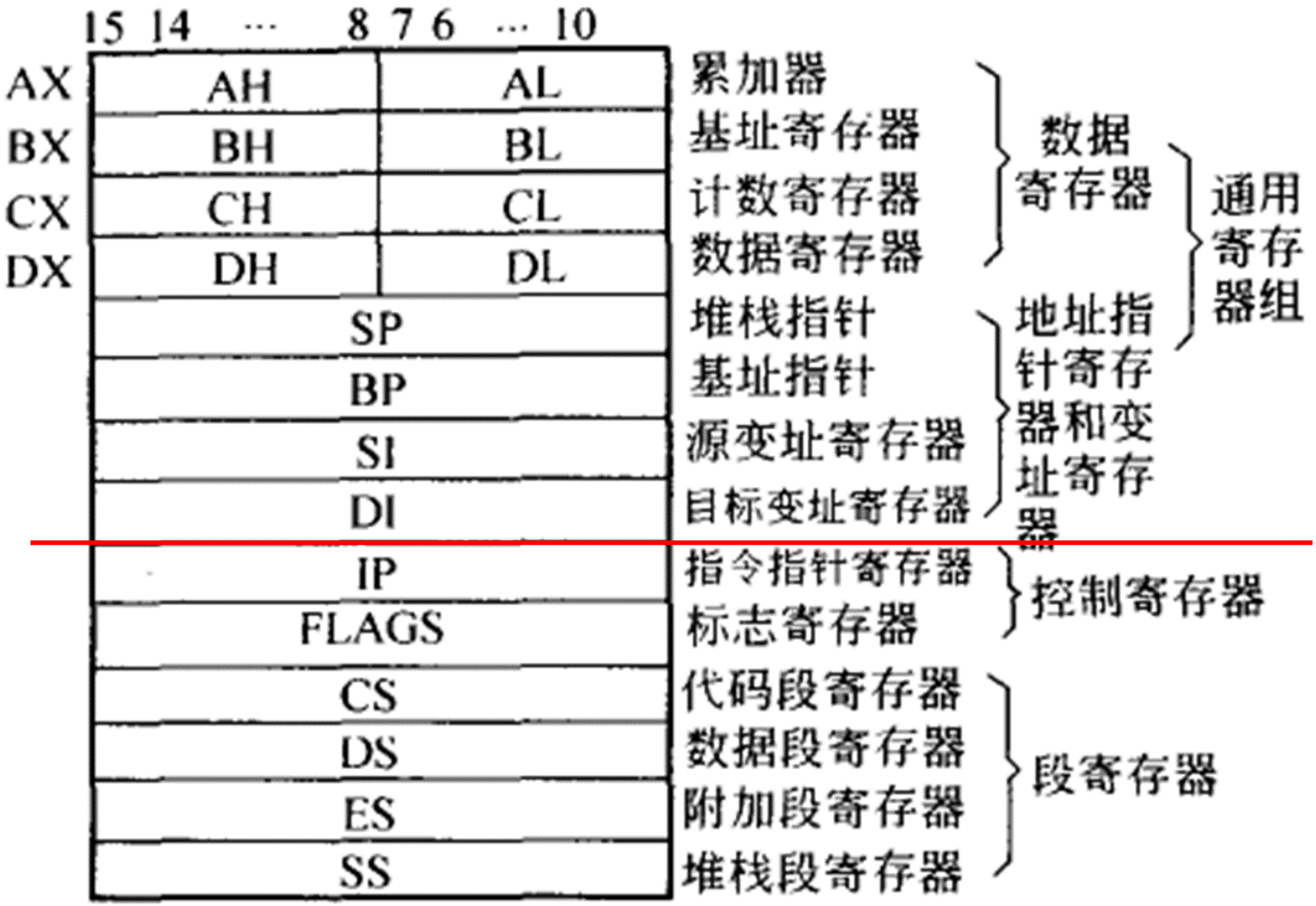
- 通用寄存器
  - 数据寄存器
  - 地址指针寄存器
  - 变址寄存器
- 段寄存器
- 指令指针寄存器
- 标志寄存器

寄存器分类（1）

- 共14个寄存器，每个寄存器16位，分三类
  - 8个通用寄存器
  - 4个段寄存器
  - 2个控制寄存器

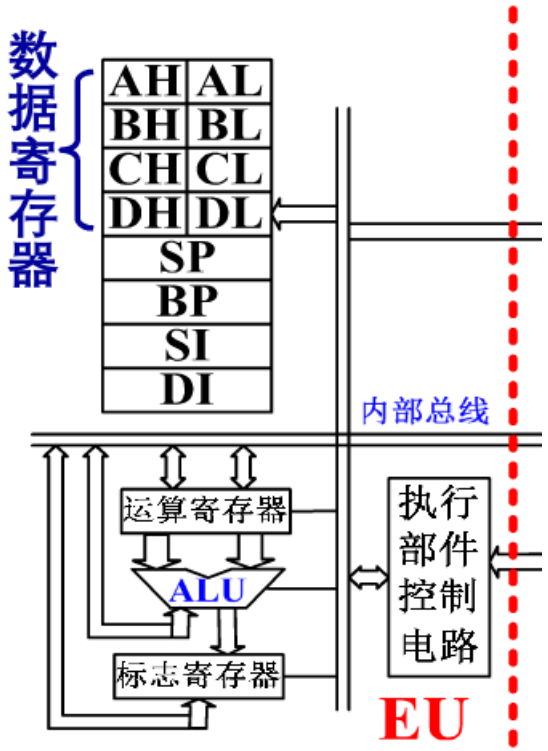
深入理解：每个寄存器中数据的含义

寄存器分类（2）


15 14 … 8 7 6 … 10
AX
AH
AL
BX
BH
BL
CX
CH
CL
DX
DH
DL
SP
BP
SI
DI
IP
FLAGS
CS
DS
ES
SS
累加器
基址寄存器
计数寄存器
数据寄存器
堆栈指针
基址指针
源变址寄存器
目标变址寄存器
指令指针寄存器
标志寄存器
代码段寄存器
数据段寄存器
附加段寄存器
堆栈段寄存器
数据寄存器
地址指针寄存器和变址寄存器
通用寄存器组
控制寄存器
段寄存器

通用寄存器

- 所有通用寄存器都可用来存放数据，作为各种算术、逻辑运算的操作数使用
- 分为三类
  - 4个数据寄存器（AX，BX，CX，DX）
  - 2个地址指针寄存器（SP，BP）
  - 2个变址寄存器（SI，DI）



数据寄存器

- AX、BX、CX、DX寄存器可以8位或16位参与操作

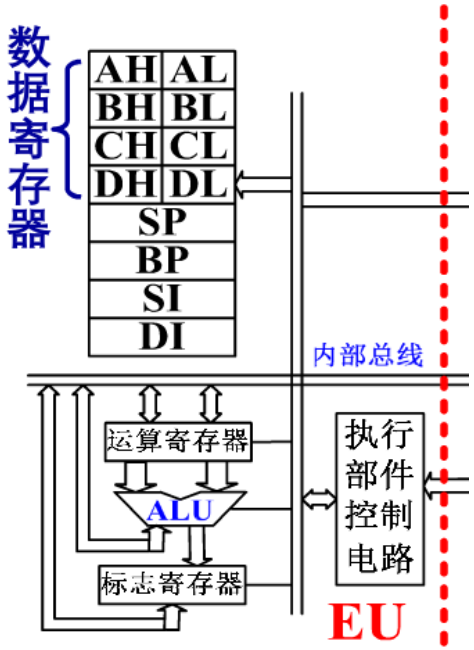
  - AX(Accumulator)： 使用频度最高，用于算术、逻辑运算以及与外设传送信息等
  - BX（Base address Register）：可用于存放存储器地址
  - CX（Counter）：循环、串操作等指令中的隐含计数器
  - DX（Data register）： 可存放双字长数据的高16位，或外设端口地址

符号	名称	高8位符号	低8位符号
AX	累加器	AH	AL
BX	基址寄存器	BH	BL
CX	计数寄存器	CH	CL
DX	数据寄存器	DH	DL

指针寄存器

- 只能以16位参与操作
- SP（Stack Pointer）：指示堆栈段栈顶的位置（偏移地址）
- BP（Base Pointer）：表示数据在堆栈段中的基地址

符号	名称
SP	堆栈指针寄存器
BP	基址指针寄存器



用BX表示所寻找的数据在数据段

用BP则表示数据在堆栈段

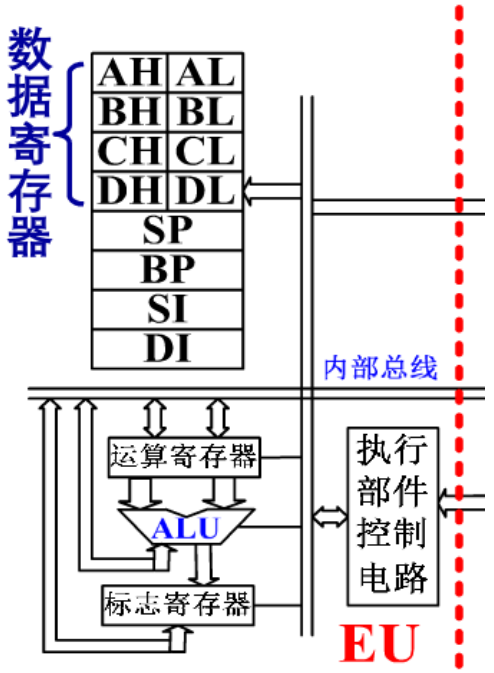
变址寄存器

- 只能以16位参与操作
- SI（Source Index）：常用于存储器变址寻址方式时提供地址；存放源数据的地址。
- DI（Destination Index）：在串操作类指令中，SI、DI有特殊用法；存放目的数据的地址。

SI → DI

source destination

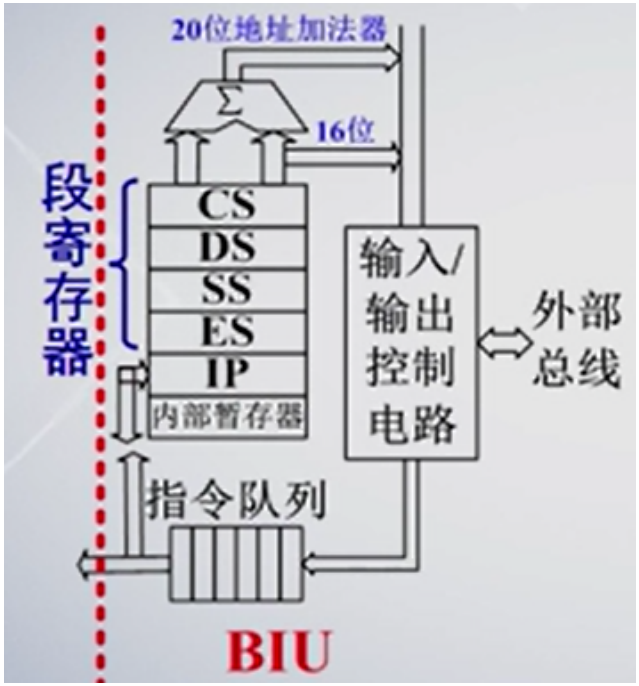
SI	源变址寄存器
DI	目的变址寄存器



段寄存器

- 四个段寄存器
  - CS(Code Segment)
  - DS(Data Segment)
  - ES(Extra Segment)
  - SS(Stack Segment)
- 每个段寄存器16位
- 用来存放相应逻辑段的段基地址
- DS、SS和ES的内容可由程序设置，CS的内容不能用程序设置

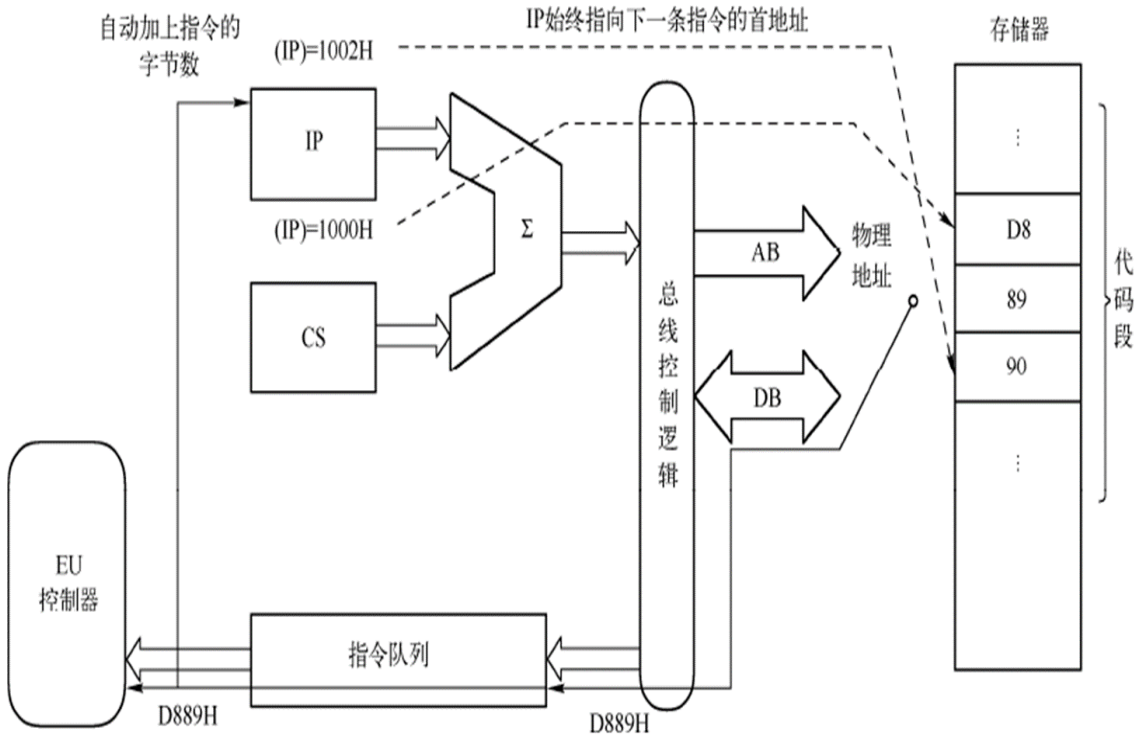
符号	名称
CS	代码段寄存器
DS	数据段寄存器
ES	附加段寄存器
SS	堆栈段寄存器

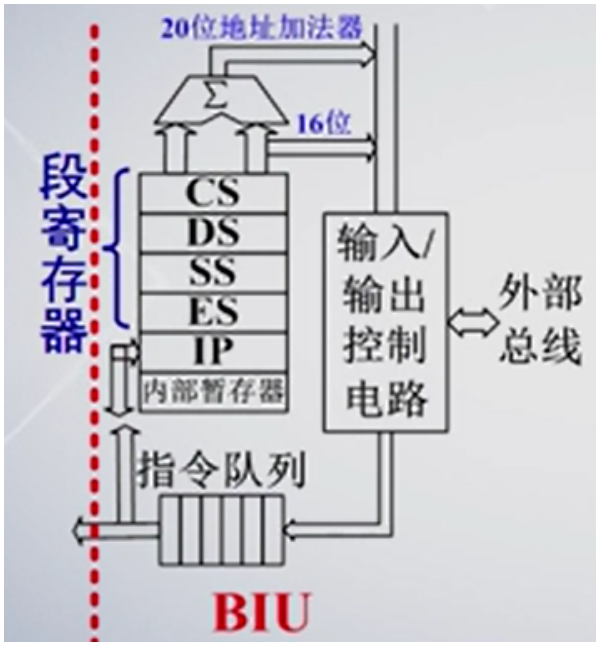


指令指针寄存器

- IP（Instruction Pointer）

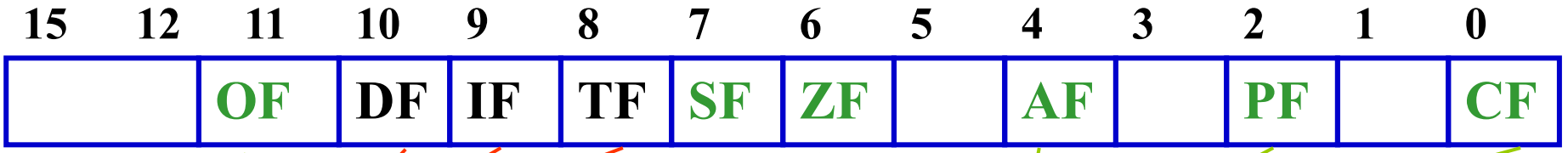
- 内容为下一条要执行指令的偏移地址。
- IP与CS寄存器联合确定下一条要执行指令的物理地址





标志寄存器 Flag（1）

- 用于反映指令执行结果和处理器的工作状态
- 6个状态标志位（CF，SF，AF，PF，OF，ZF）
- 3个控制标志位（IF，TF，DF）



控制标志：由程序进行设置，用于控制处理器执行指令的方式

状态标志： 记录程序运行结果的状态信息，许多指令的执行都将相应地设置它们。

标志寄存器 Flag（2）

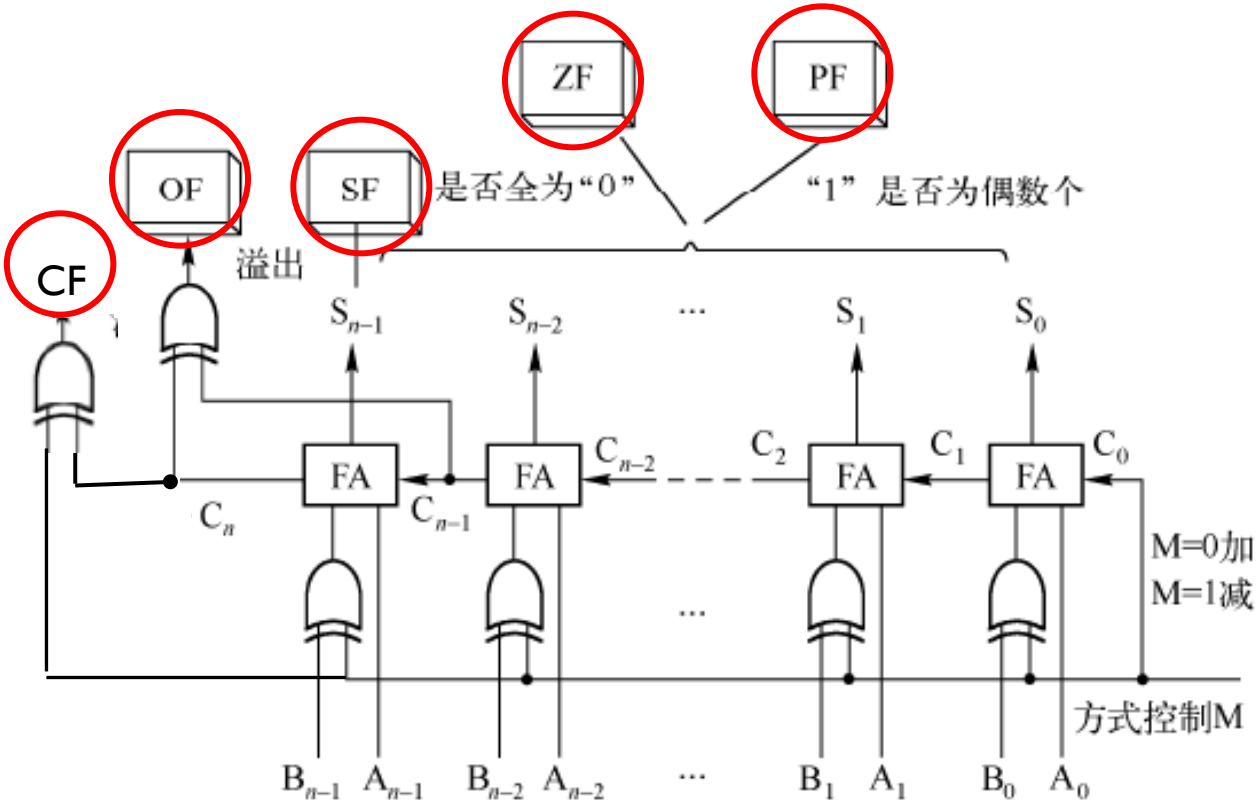
符号	名称	定义
CF	进位标志	运算中，最高位有进位或借位时CF=1，否则CF=0
PF	奇偶标志	运算结果低8位“1”个数为偶数时PF=1，否则PF=0
AF	辅助进位	D3有向D4进（借）位时AF=1，否则AF=0
ZF	零标志	运算结果每位均为“0”时ZF=1，否则ZF=0
SF	符号标志	运算结果的最高位为1时SF=1，否则SF=0
OF	溢出标志	运算 致补码溢出时OF=1，否则OF=0

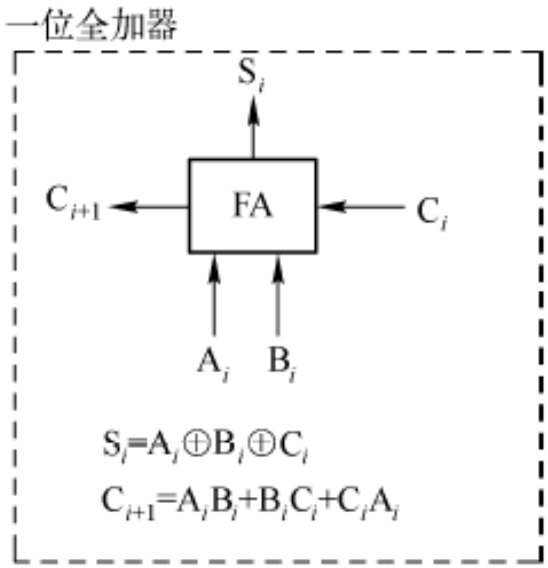
程序设计需要利用标志的状态

符号	名称	功能
TF	陷阱标志	TF=1将使CPU进入单步执行指令
IF	中断标志	IF=1允许CPU响应可屏蔽中断
DF	方向标志	DF=1将从高地址向低地址处理字符串

标志寄存器 Flag（3）

- n位串行进位加法器





例1

十六进制数0CCCCH与十六进制数5115H相加，请写出运算后六个标志状态位的值。

解：
```
  1100110011001100   (0CCCCH)
+ 0101000100010101   (5115H)
------------------
 1000111011100001
```

CF=1 ZF=0 SF=0

PF=1 AF=1 OF=0

例2

- 无符号数和有符号数加法示例：

表 2.1　无符号数和有符号数加法示例

无符号数加法		有符号数加法	
$\begin{array}{r} 00001001 \\ +\ 01111100 \\ \hline 10000101 \end{array}$	$\begin{array}{r} 00000010 \\ +\ 11111111 \\ \hline 00000001 \end{array}$	$\begin{array}{r} 00001001 \\ +\ 01111100 \\ \hline 10000101 \end{array}$	$\begin{array}{r} 00000010 \\ +\ 11111111 \\ \hline 00000001 \end{array}$
CF=0 OF=1	CF=1 OF=0	CF=0 OF=1	CF=1 OF=0
正确：9+124=133	错误：2+255=1	错误：9+124=−123	正确：2+(−1)=1

无符号数用CF判断有无溢出
有符号数用OF判断有无溢出

思考

- 8086的并行操作方式如何实现？
- 8086有多少个寄存器？每个寄存器多少位？
- 均属于执行部件（EU）的寄存器是（　）

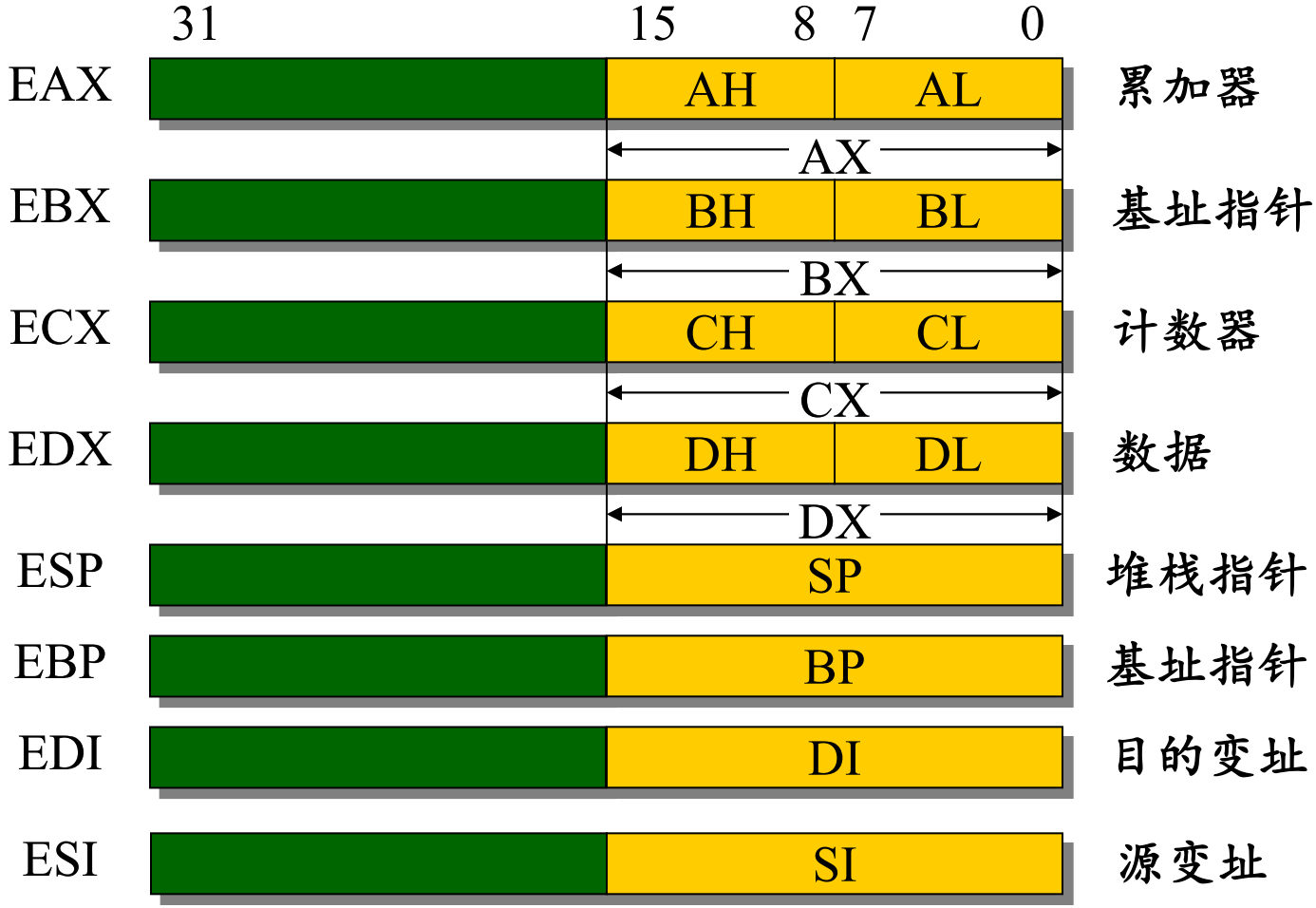
A SP、IP　　B DX、DS

C CX、CS　　D BX、BP

知识拓展

32位处理器（1）

- 8个通用寄存器扩展位32位，保持向后兼容



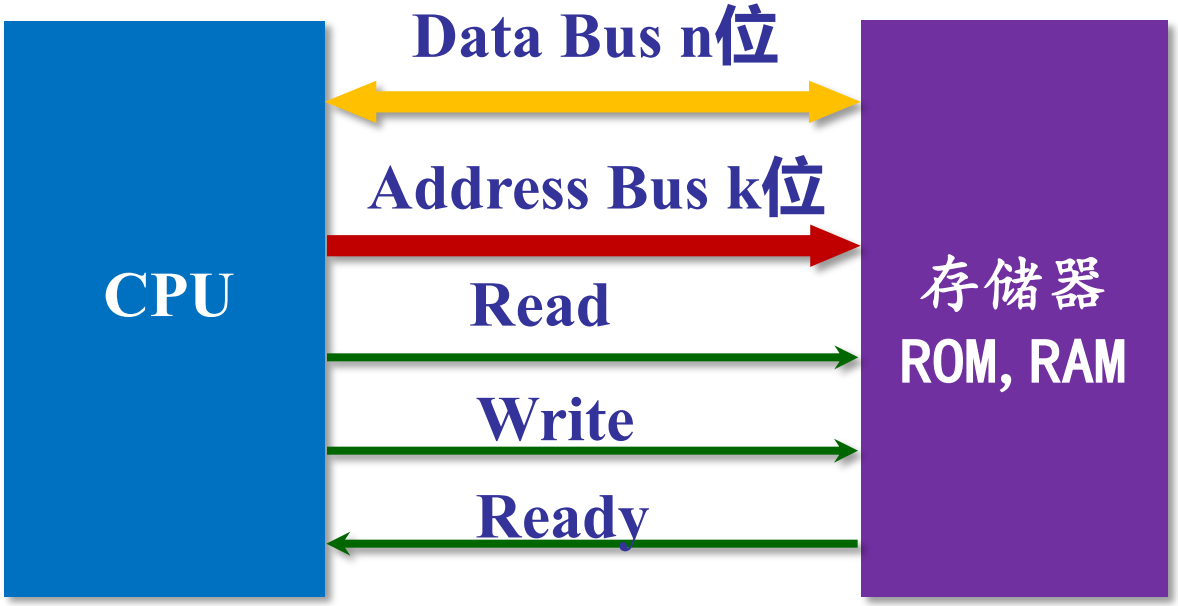
32位处理器（2）


31
0
CS
代码段
DS
数据段
ES
附加段
SS
堆栈段
FS
GS
EIP
IP
程序指针
47
16
15
0
基地址
限值
GDTR
基地址
限值
IDTR
LDTR
TR
控制调试寄存器
有很多…
31
0
CR0
CR1
CR2
CR3
CR4
标志寄存器
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
NT
IOPL
OF
DF
IF
TF
SF
ZF
AF
PF
CF
EFLAGS 31
22 21 20 19 18 17 16
ID
VIP
VIF
AC
VM
RF

2.3 8086存储器组织

- 内存地址空间和数据组织
- 逻辑地址和物理地址

基本概念（1）



对比：

寄存器：是微处理器芯片内部存放数据的存储单元，用名称区别

存储器：是微处理器外部存放程序和数据的空间，用地址寻址

- 物理存储器：处理器通过其总线可寻址的存储器
- 物理地址：为物理储存器的每个存储单元分配的唯一地址
- 物理地址空间：k条地址线，物理地址空间大小为$0 \sim 2^k - 1$

基本概念（2）

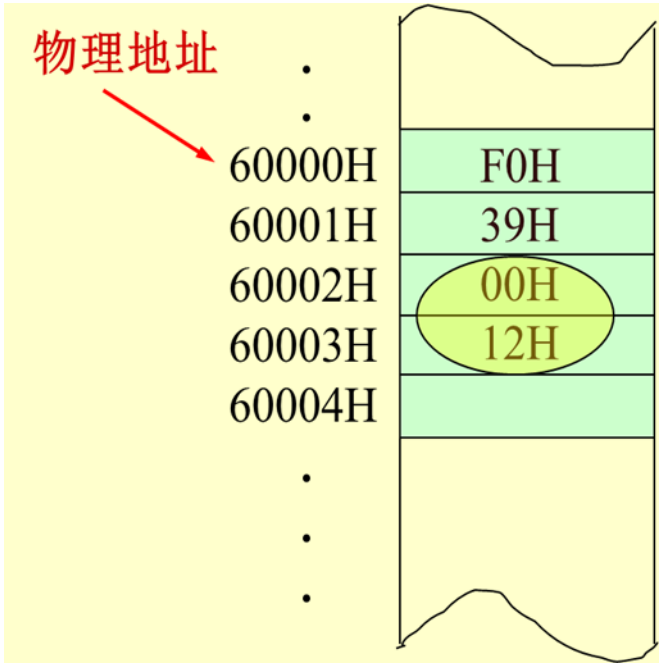
- 易失性半导体存储器统称为RAM
- 非易失性的半导体存储器统称为ROM
- RAM可以分为：
  - 静态RAM（SRAM）
  - 动态RAM（DRAM）
- ROM可以分为：
  - 掩膜ROM（MASK ROM）
  - 可编程ROM（PROM）
    - 紫外线擦除EPROM（UV EPROM）
    - 电擦除EPROM（EEPROM，E^2PROM）
    - 闪速存储器（FLASH ROM）

8086存储器组织

- 8086处理器地址线为20条
- 最大可寻址空间为2^{20}＝1MB
- 物理地址范围00000H~FFFFFH
- 16位数据的存储规则：

  低字节存入低地址

  高字节存入高地址



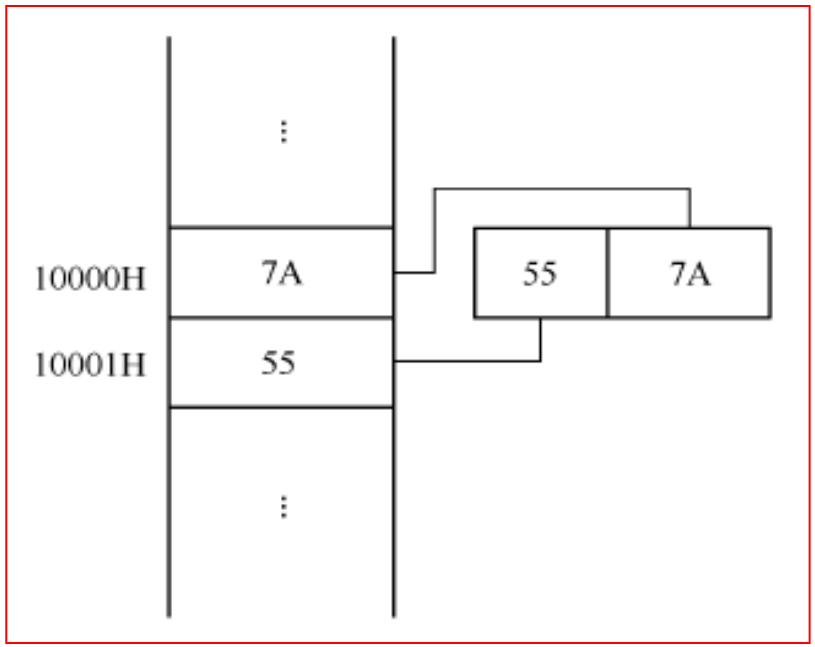
$D_{15\sim0}$= 1200H

[60003H] [60002H]

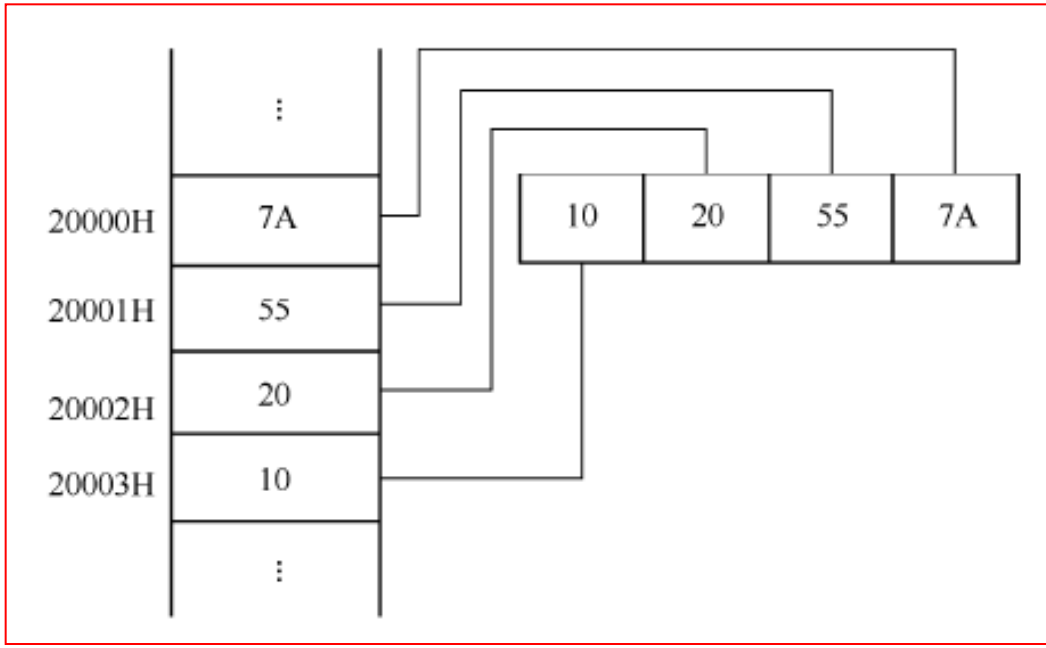
英特尔x86处理器这种数据存储方式称为小端方式。与之相反的称为大端方式

数据存储方式举例

字的存储

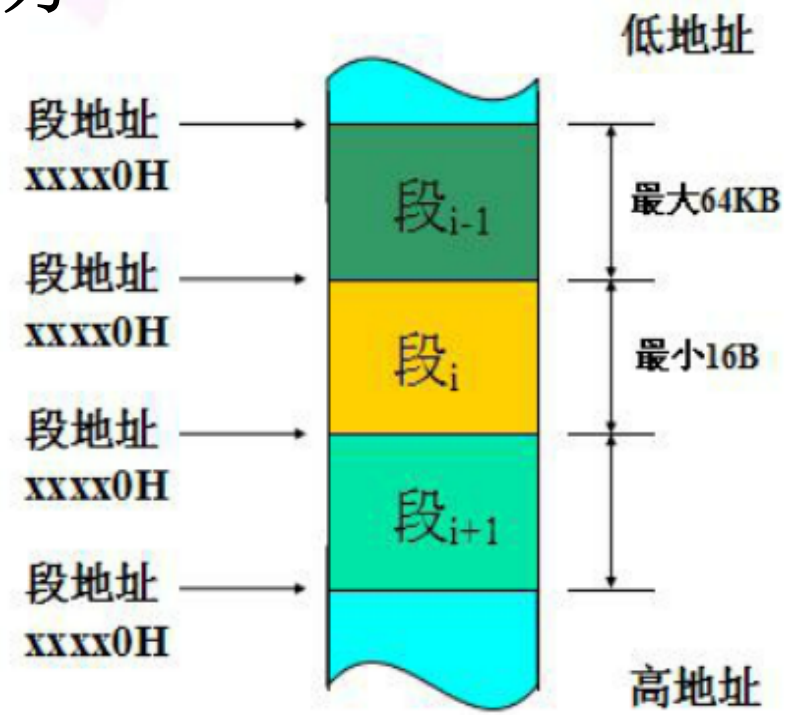


双字的存储



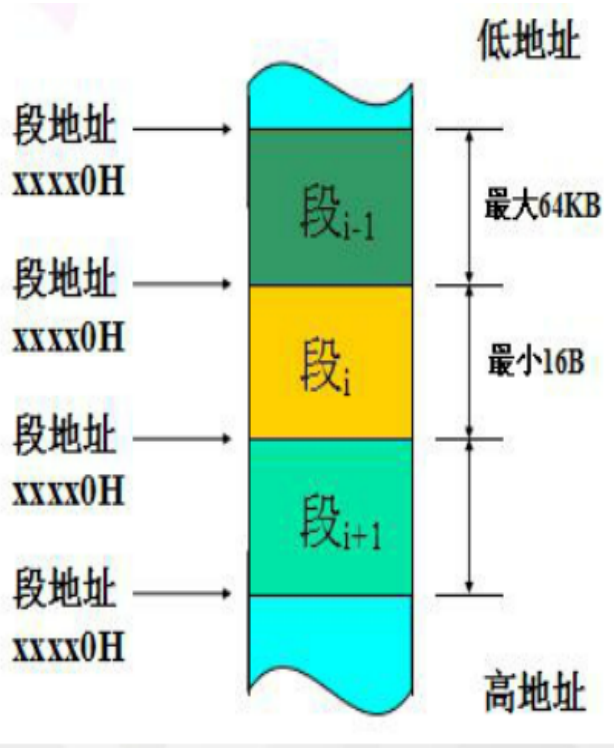
逻辑地址（1）

- 存储器分段
  - 问题：8086的内部总线和寄存器均为16位，如何指定20位地址？
  - 解决：存储器分段管理
- 内存分为多个逻辑段（Segment）
  - 段（基）地址：必须是16的倍数
  - 逻辑段最大空间：64K
- 段基地址
  - 段基址：XXXX0H
  - 目的是把XXXXH（16位）存放在段寄存器（CS、DS、ES和SS）中



逻辑地址（2）

- 偏移地址
  - 在段内相对于段基址的偏移值（距离）
  - 偏移地址（也称为段内地址）占16位



- 段基地址和偏移地址组成了逻辑地址记为：段地址:偏移地址

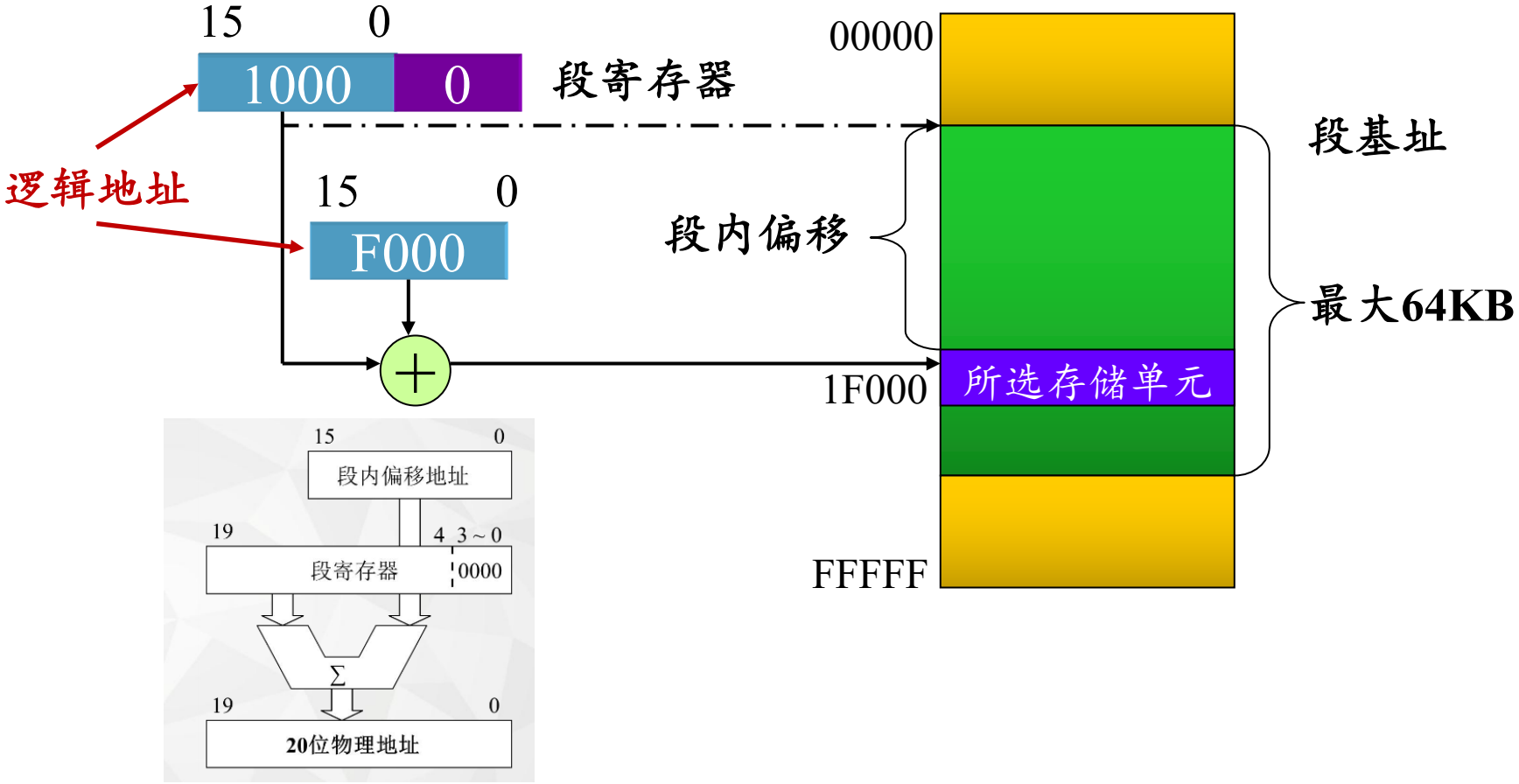
  6000:0002

- 逻辑地址是程序内部使用的地址，使程序员在编程时无需考虑物理存储器的情况

  逻辑地址 ⇒ 物理地址？

实模式存储器寻址

- 逻辑地址到物理地址的转换



20位物理地址 = 段地址（16位）×16 + 偏移地址（16位）

知识拓展 80x86存储器管理

- 实模式（Real-address mode）：8086处理器的存储器模式，使用20位物理地址、单任务工作方式、独占系统所有资源。存储器空间为00000H~FFFFFH。处理器上电后首先进入实模式

  DOS系统要求实模式
- 保护模式（Protected mode）：为操作系统实现虚拟存储器系统、支持多任务等功能提供硬件支持。Windosw、macOS和Linux等操作系统均要求处理器运行于保护模式。

  由于课时限制，不在本课程范围
- 到目前为止，处理器加电后先进入实模式运行，然后切换到保护模式

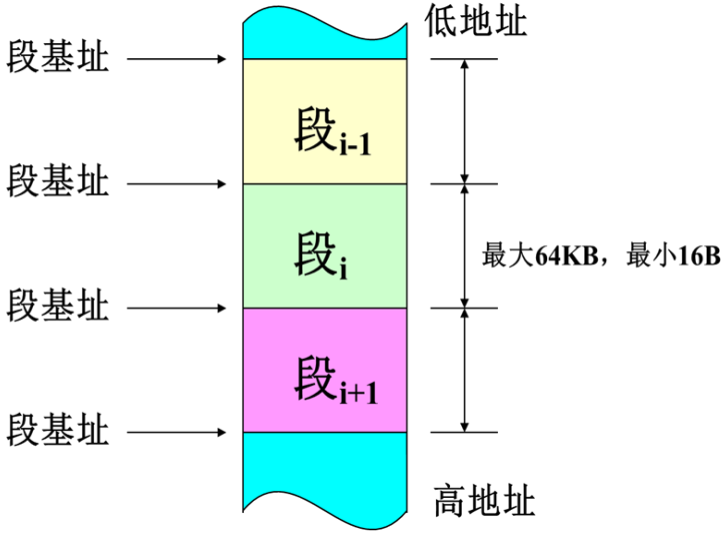
举例

已知CS=1055H，DS=250AH，ES=2EF0H，SS=8FF4H，DS所指示的段有一操作数，其偏移地址=0204H。

1）画出各段在内存中的分布。

2）指出各段的物理起始地址。

3）该操作数的物理地址=?



解答

已知CS=1055H，DS=250AH，ES=2EF0H，SS=8FF4H，DS所指示的段有一操作数，其偏移地址=0204H。

1）画出各段在内存中的分布。

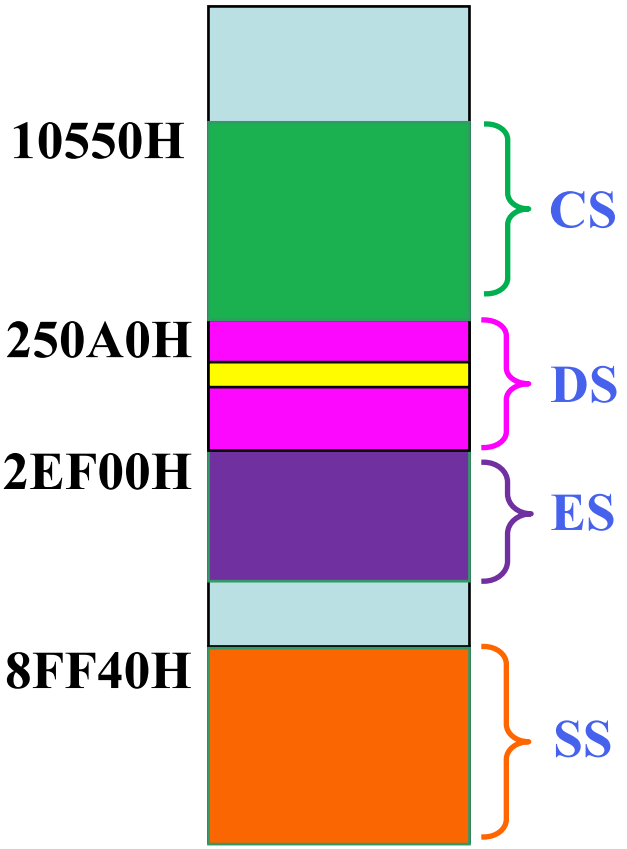
2）指出各段的物理起始地址。

3）该操作数的物理地址=?

解：

操作数的物理地址为：

250AH×10H+0204H = 252A4H



动手做

1、CPU从内存中物理地址为85004H、85005H读出16位数据存入寄存器DX中，问DX的内容是

A. (DX) = 7DA3H

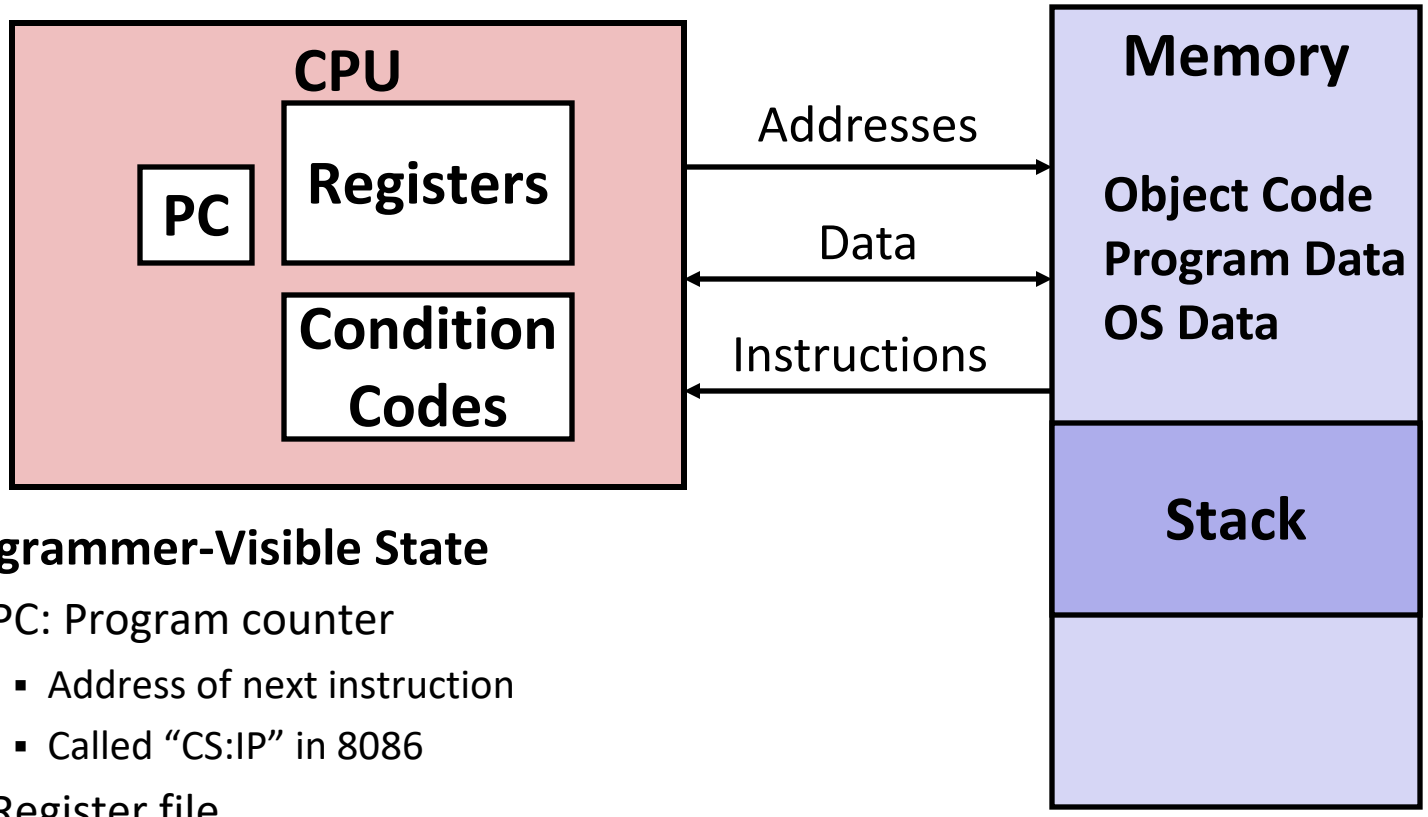
B. (DX) = A37DH

A4H	00000H
.....	
7DH	85004H
A3H	85005H
.....	
12H	FFFFDH
34H	FFFFEH
	FFFFFH

2、某数x在数据段偏移地址为FFFDH，该数据段的段地址为F000H，问数x在内存中的物理地址是

A. FFFDH　　B. 1EFFDH.　　C. FFFFDH

汇编程序员视角



- **Programmer-Visible State**
  - PC: Program counter
    - Address of next instruction
    - Called “CS:IP” in 8086
  - Register file
    - Heavily used program data
  - Condition codes(Flag)
    - Store status information about most recent arithmetic operation
    - Used for conditional branching

- **Memory**
  - Byte addressable array
  - Code, user data, (some) OS data
  - Includes stack used to support procedures

本章小结

- 8086内部结构
- 8086EU和BIU的功能
- 8086内部寄存器
- 8086存储器组织
- 逻辑地址、物理地址、地址转换

EMU 8086仿真

- 请点击观看这个短视频