

# 微机原理与接口技术

---

## 第二章 8086微处理器

## 第二章 8086微处理器

---

- 2.1 8086微处理器简介
- 2.2 8086的编程结构
- 2.3 8086的引脚信号定义
- 2.4 8086系统总线的形成
- 2.5 8086的总线操作时序
- 2.6 8086存储器和I/O组织

## 2.1 8086微处理器简介

---

- ❑ Intel公司推出的第三代CPU芯片
- ❑ 16位处理器，数据线：16位，地址线：20位
- ❑ 封装:双列直插,40管脚
- ❑ 时钟频率：5MHz~10MHz
- ❑ 供电：单一，+5V供电



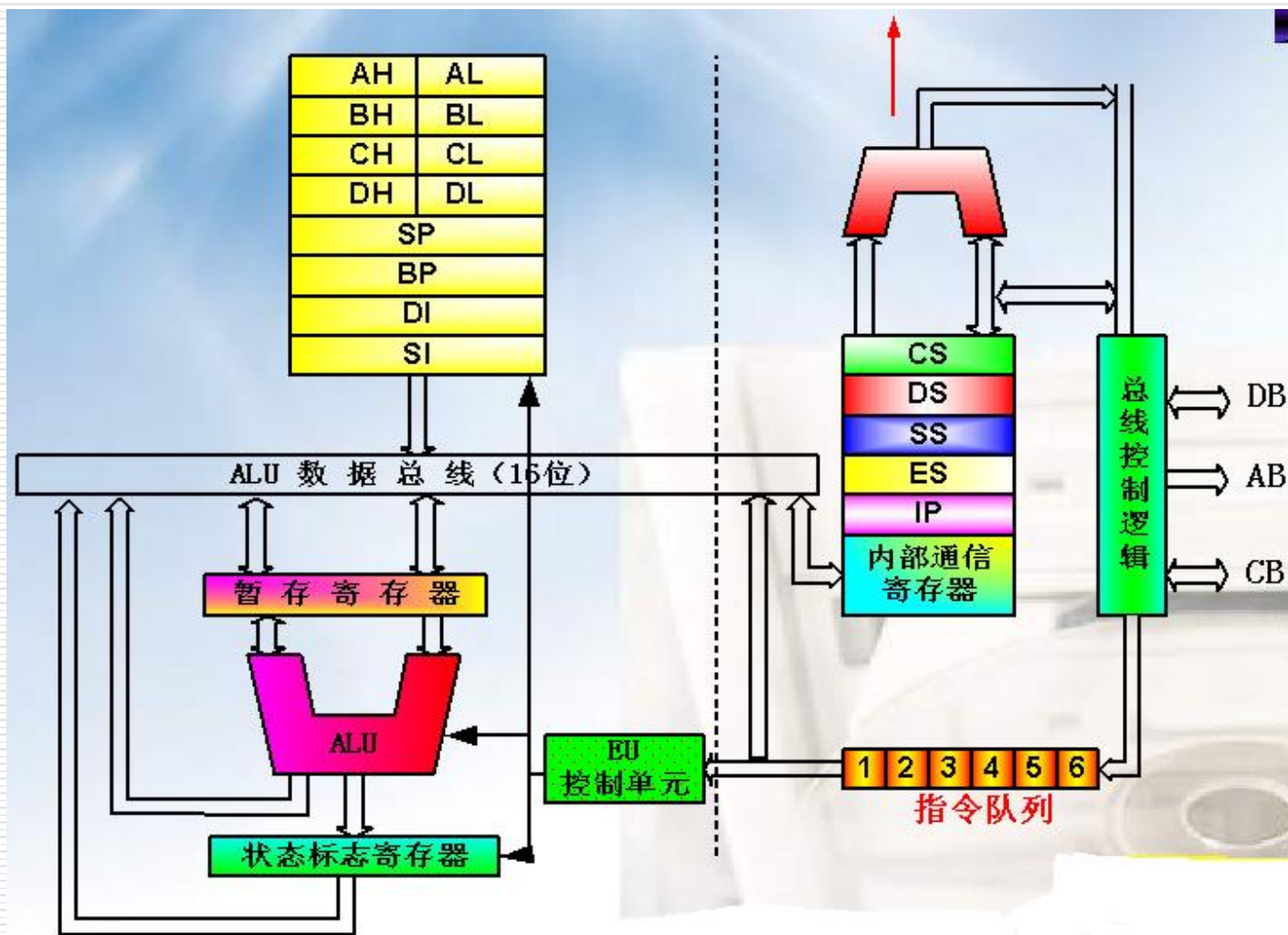
## 2.2 8086的编程结构-微处理器的编程结构

---

- 微处理器的编程结构是指从软件编程和使用者的角度看到的结构模型，亦可称为功能结构。例如：有哪些寄存器，如何操作？总线接口如何？ .....

## 2.2 8086的编程结构-8086的内部结构

- 8086的内部结构从功能上划分为两个功能单元：总线接口部件**BIU**和执行部件**EU**



## 2.2 8086的编程结构-8086的内部结构

### ❑ BIU：总线接口部件（单元）

控制完成全部总线操作，取指令、读数据和写运算结果。

### ❑ 20位地址加法器

### ❑ 4个16位段寄存器：

■ CS：代码段寄存器

■ DS：数据段寄存器

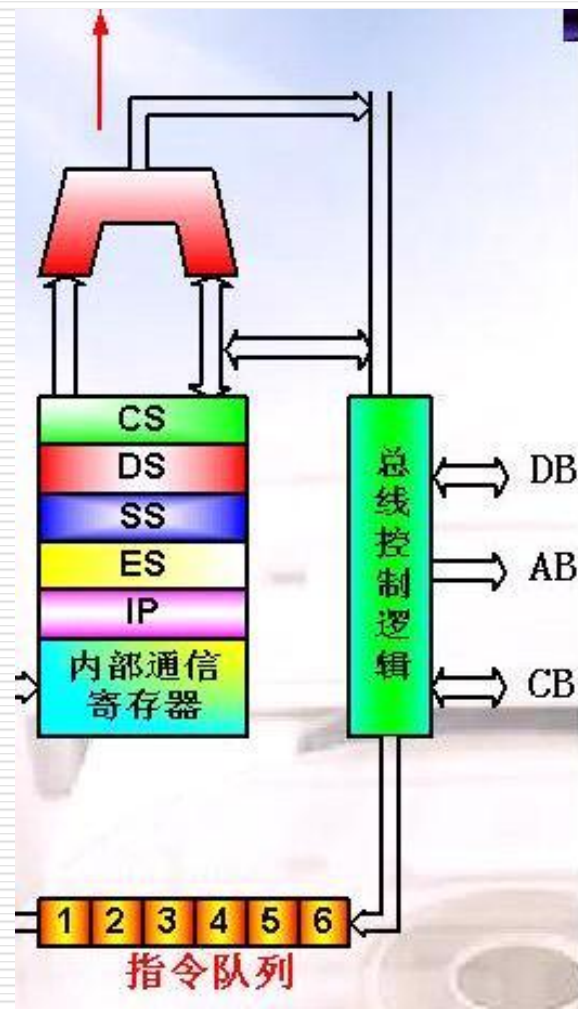
■ ES：扩展段寄存器

■ SS：堆栈段寄存器

### ❑ 1个16位指令指针寄存器：IP

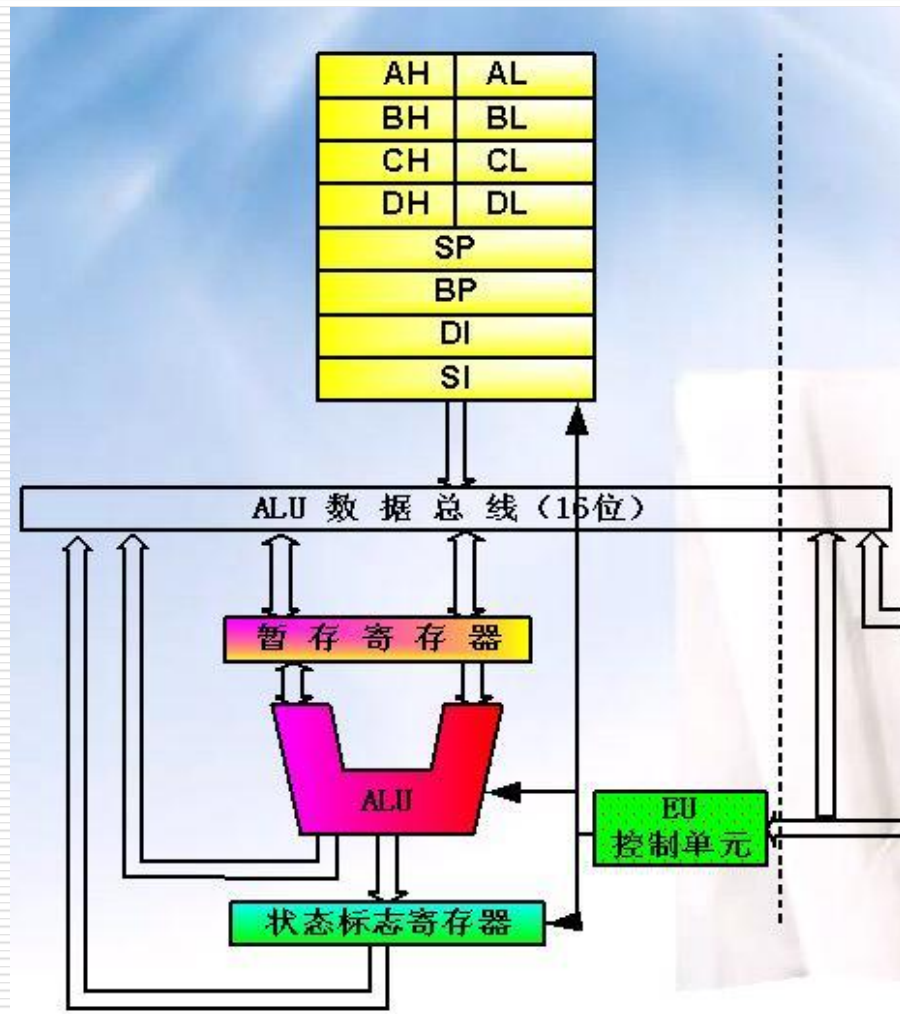
### ❑ 6字节的指令队列缓冲器

### ❑ 总线控制逻辑



## 2.2 8086的编程结构-8086的内部结构

- ❑ EU：执行部件（单元）  
分析指令，执行指令
- ❑ 1个16位ALU(算术逻辑单元)
- ❑ 8个通用寄存器
  - AX、BX、CX、DX
  - BP、SP、SI、DI
- ❑ 标志寄存器FLAGS
- ❑ EU控制系统



## 2.2 8086的编程结构—指令流水

---

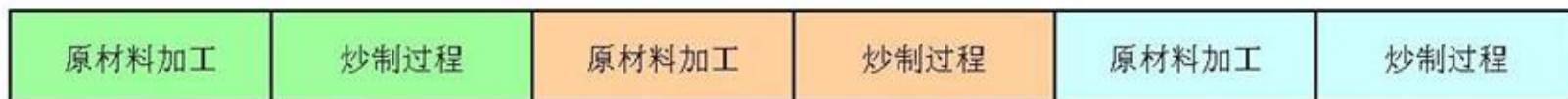
- 指令流水是一种实现多条指令重叠执行的重要技术，可有效提高总线传输效率和整个系统的执行速度。
- 指令流水的核心思想
  - 分级：将任务分解为多个子任务，子任务可由不同功能单元完成
  - 并行处理：同时对多个子任务进行处理



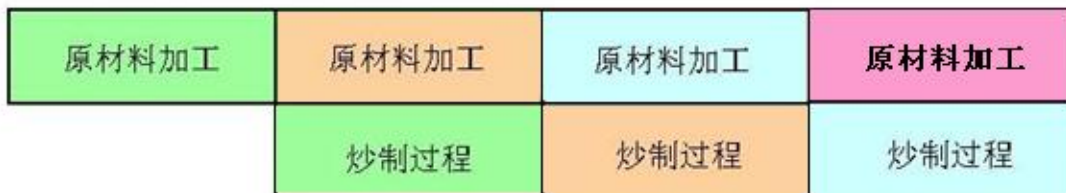
# 流水举例



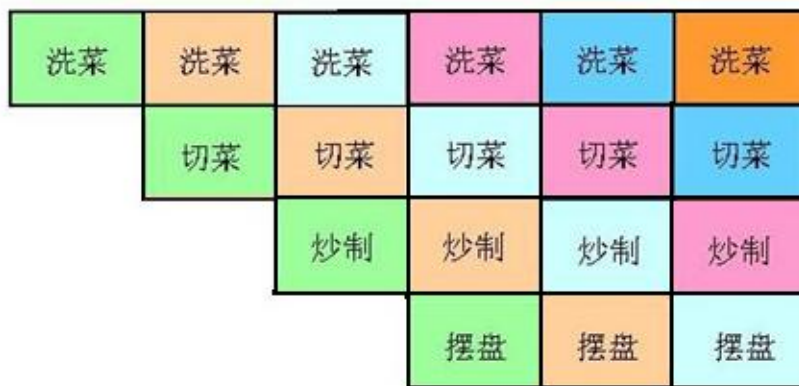
串行方式



两级流水



多极流水



## 2.2 8086的编程结构—BIU与EU的工作方式

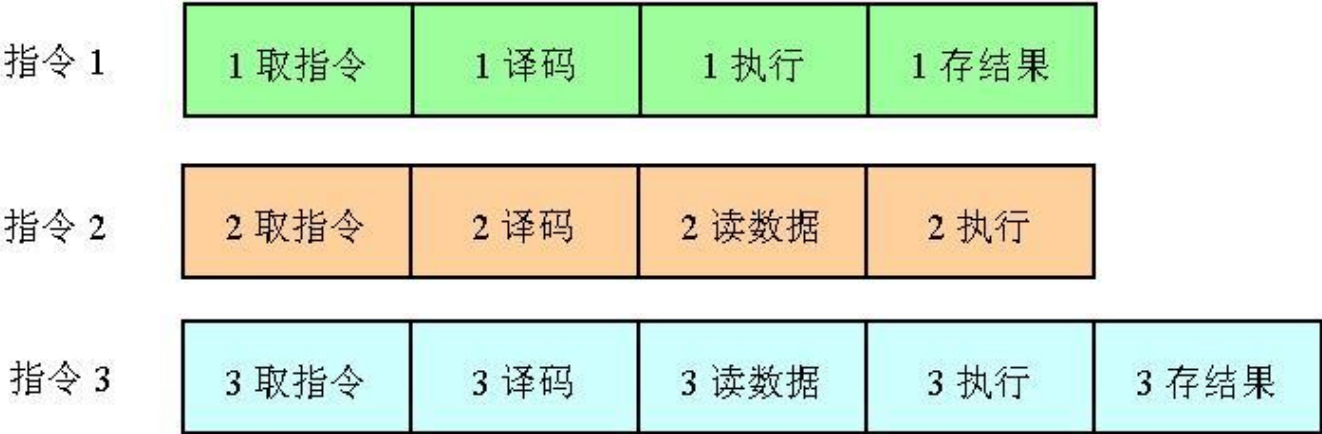
---



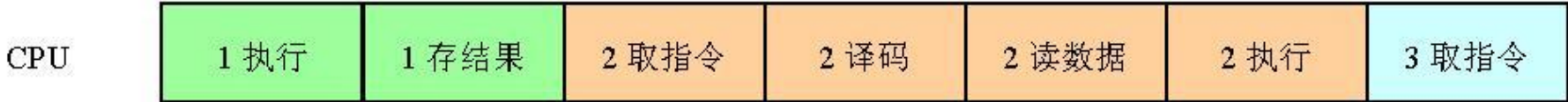
BIU与EU功能上相互独立，可并行进行工作

- BIU中的指令队列有2个或以上空字节时，BIU自动从存储单元中取指令，填充指令队列；
- EU执行完一条指令，立即从BIU指令队列首部取下一指令；
- EU在执行指令时，需要进行存储器或I/O访问时，向BIU发出总线访问请求；若这时BIU空闲，则立即进行总线访问；
- EU在执行转移、调用及返回指令时，指令队列自动清除，BIU根据目标地址重新取指令填充指令队列。

# BIU与EU的工作方式举例



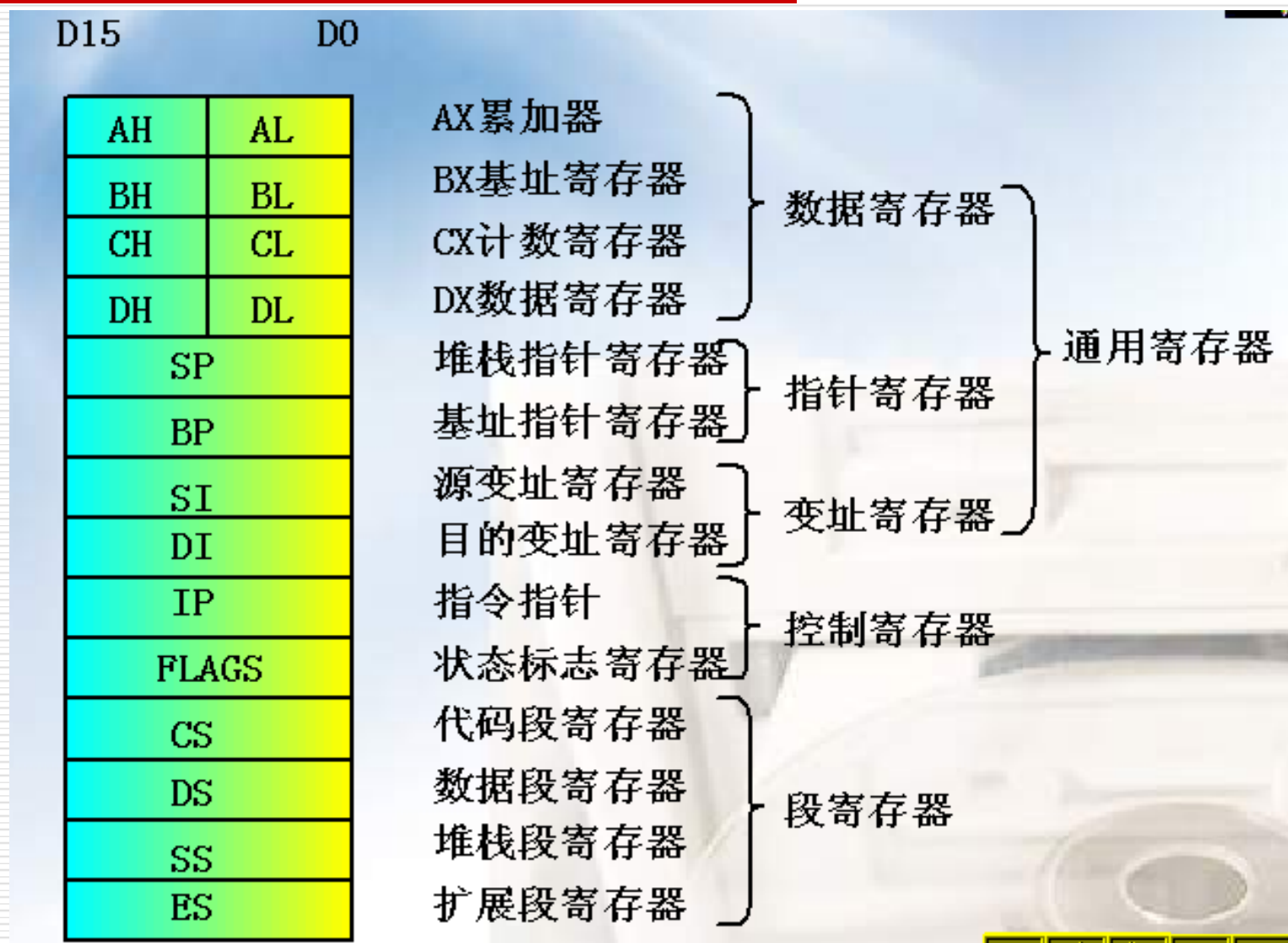
## 串行方式



## 流水方式



## 2.2 8086的编程结构-8086的内部寄存器



## 2.2 8086的编程结构-8086的内部寄存器

---

### □ 4个16位通用数据寄存器

- AX:累加器
- BX:基地址寄存器
- CX:计数寄存器
- DX:辅助累加器

### □ 作为1个16位寄存器使用

### □ 作为2个8位寄存器使用（例如，AH，AL）

### □ 隐含性质

## 2.2 8086的编程结构-8086的内部寄存器

---

### □ 2个16位指针寄存器

- SP:堆栈指针寄存器

- BP:基址指针寄存器

### □ 2个16位变址寄存器

- SI: 源变址寄存器

- DI:目的变址寄存器

### □ 只能作为16位寄存器使用

### □ 可存放数据，作为通用寄存器使用

## 2.2 8086的编程结构-8086的内部寄存器

□ 4个16位段寄存器 

■ CS: 代码段寄存器

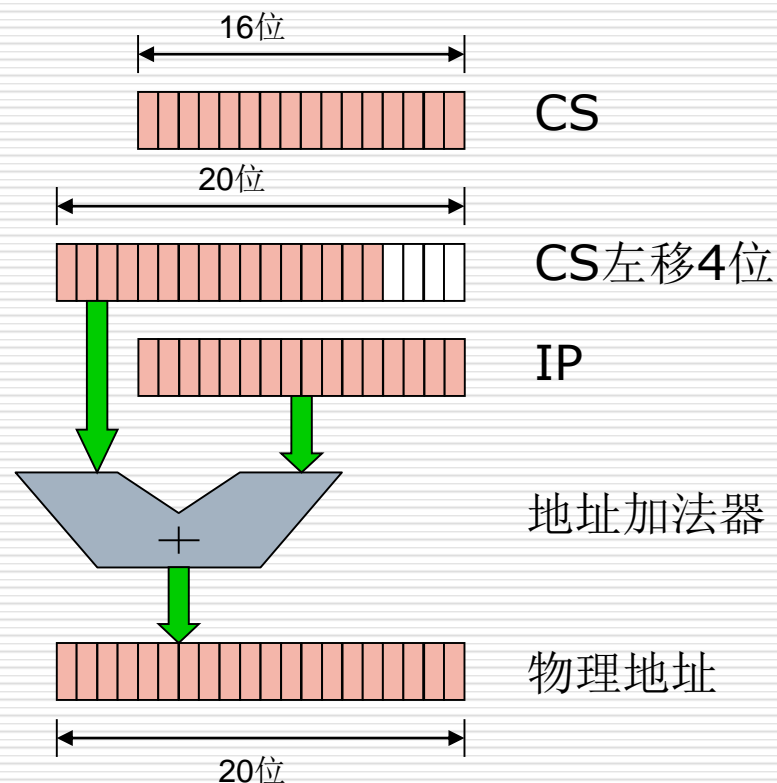
■ DS: 数据段寄存器

■ ES: 扩展段寄存器

■ SS: 堆栈段寄存器

□ 1个16位指令指针寄存器

■ IP



□ 段寄存器存放各段起始地址的高16位，称为段地址

□ IP存放下一条指令的偏移地址

## 2.2 8086的编程结构-8086的内部寄存器

D <sub>15</sub>	D <sub>14</sub>	D <sub>13</sub>	D <sub>12</sub>	D <sub>11</sub>	D <sub>10</sub>	D <sub>9</sub>	D <sub>8</sub>	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
				OF	DF	IF	TF	SF	ZF		AF		PF		CF

### □ 16位标志寄存器FLAGS

- 3个控制标志位：TF，IF，DF
- 6个状态标志位：CF，PF，AF，ZF，SF，OF

### □ 控制标志位

- TF：单步操作标志，用于控制单步中断
- IF：中断允许标志，IF=1，可响应可屏蔽中断
- DF：方向标志，控制串操作的步进方向，DF=1，串操作由高地址向低地址方向进行



## 2.2 8086的编程结构-8086的内部寄存器

### □ 标志寄存器的状态标志位

- CF: 进位标志, 加减法时最高位进位或借位, 该位置1
- OF: 溢出标志, 运算结果超出有符号数表示的范围时置1
- AF: 辅助进位标志, 加减法时低4位向高四位进位或借位, 该位置1
- ZF: 零标志, 运算结果为零, 该位置1
- SF: 符号标志, 与运算结果最高位相同
- PF: 奇偶标志, 运算结果低8位中1的个数为偶数时置1
- 例: 计算90+107

$$\begin{array}{r} 0101\ 1010\text{B} \\ + \underline{0110\ 1011\text{B}} \\ \hline 1100\ 0101\text{B} \end{array}$$

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

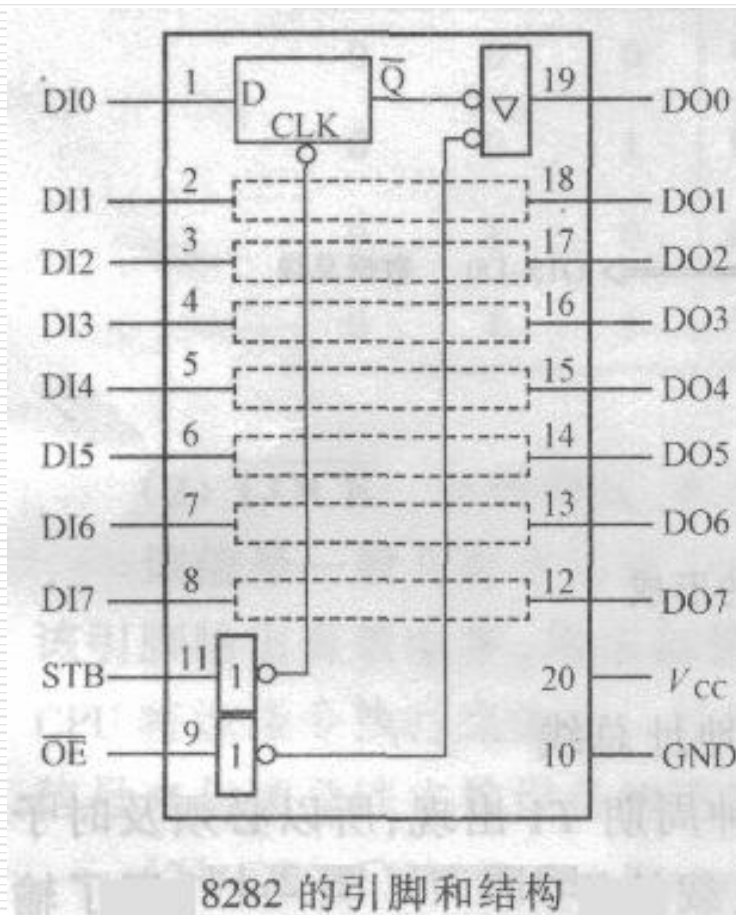
## 2.3 8086的引脚信号定义

---

- ❑ 总线操作的一般过程：CPU先送地址，地址用于选中总线操作的对象，然后进行读写操作
- ❑ 分时复用：同一引脚在不同的时刻可输出两种不同的信号。如地址/数据复用，在一个总线周期的不同状态周期，分别输出地址或数据信号。
- ❑ 地址锁存：由于地址/数据分时输出，需要外接地址锁存芯片对地址进行锁存。
- ❑ 总线驱动：当总线上挂载的负载较多时，需要外接总线驱动芯片提高总线的驱动能力。

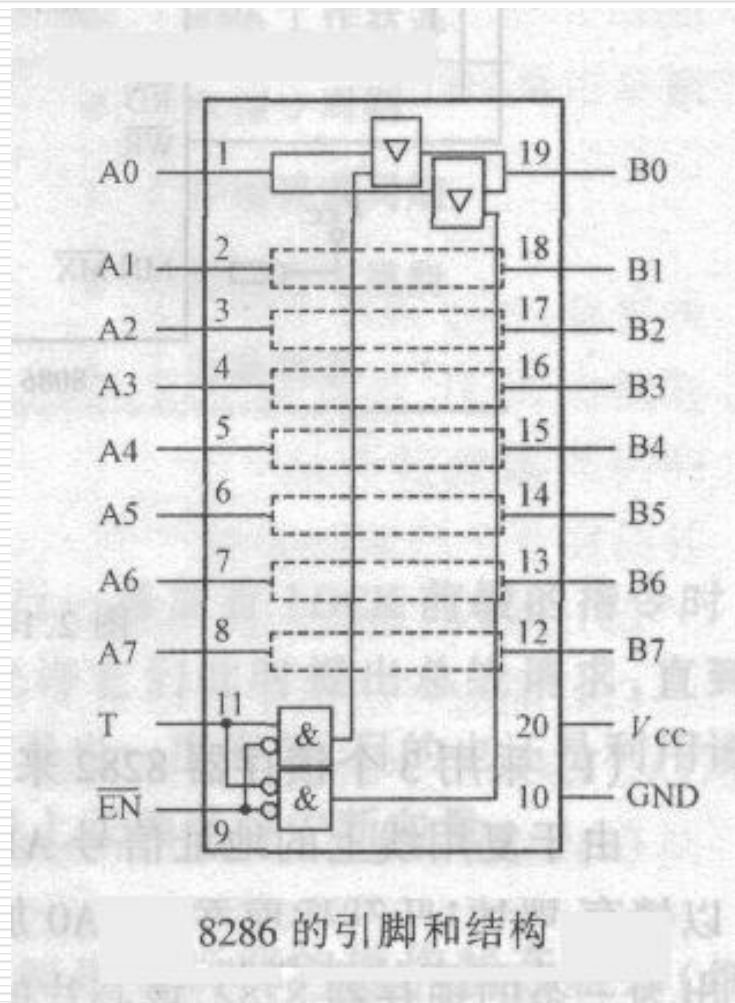
## 2.3 8086的引脚信号定义

- ❑ 地址锁存器：数据/地址线分时复用，CPU必须先将地址信息暂存起来，然后再传输数据。常使用8282或74LS373作为地址锁存器，下降沿触发；



## 2.3 8086的引脚信号定义

- 总线驱动器：提高总线的负载能力、抗干扰能力。常使用8286或74LS245增加数据总线的驱动能力



## 2.3 8086的引脚信号定义

---

### □ 8086的两种工作模式：

- 最小模式：系统只有1个8086处理器，产生所有控制信号；
- 最大模式：系统包含两个或多个微处理器，如8086、8087、8089等，存在总线争用问题，需8288总线控制器产生控制信号
- 两种模式下的引脚功能定义有所区别（第24~31引脚）

## 2.3 8086的引脚信号定义-最小模式下的引脚定义

地址/数据（或状态）信号：

□ AD15~AD0:

■ 地址/数据，双向、三态；

□ A19/S6~A16/S3:

■ 地址/状态，输出、三态

□ S6: 指示是否与总线相连；

□ S5: 指示中断容许标志状态；

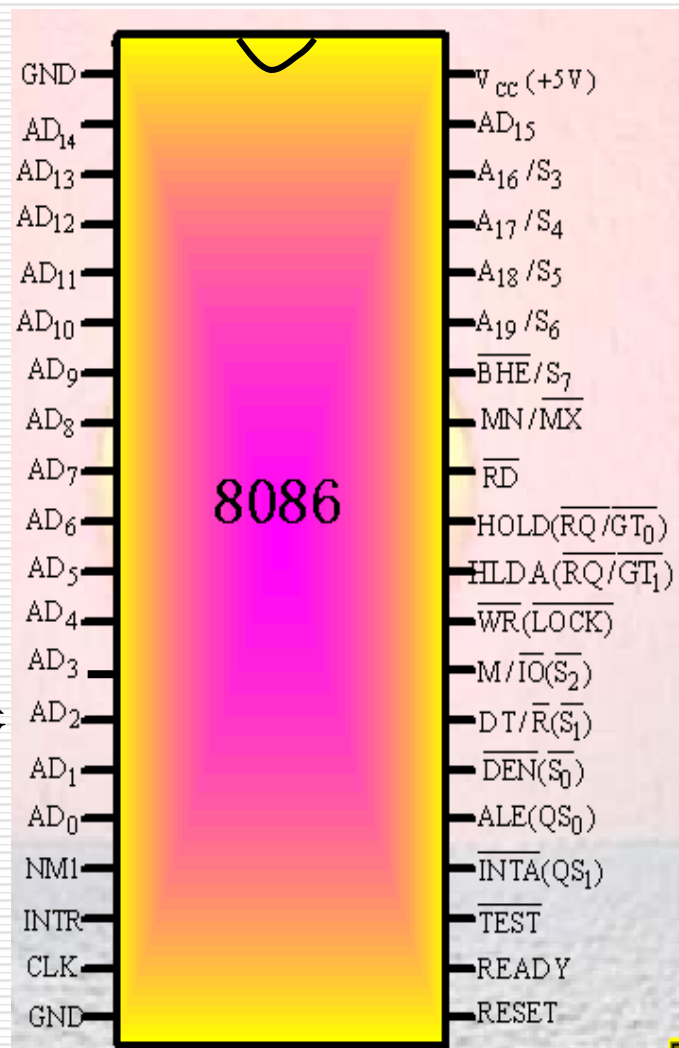
□ S4~S3: 指示当前使用的段寄存器

□  $\overline{\text{BHE}}/\text{S7}$ :

■ 数据总线高8位使能/状态S7，

■ 输出、三态

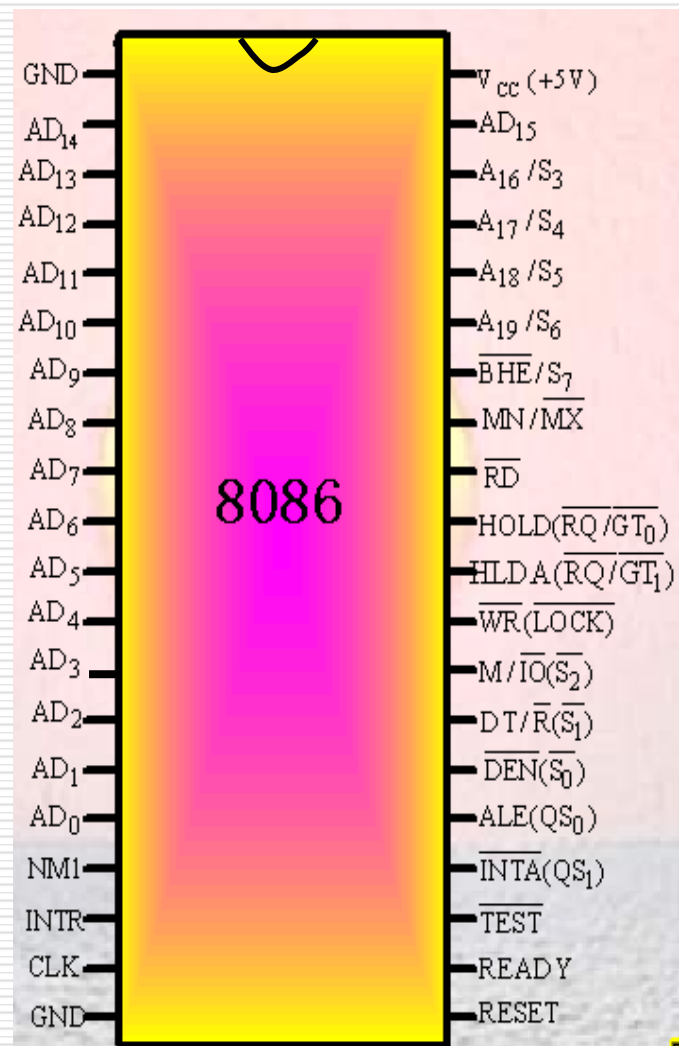
■  $\overline{\text{BHE}}=0$ ，AD15~AD8数据有效；



## 2.3 8086的引脚信号定义-最小模式下的引脚定义

控制信号与系统信号:

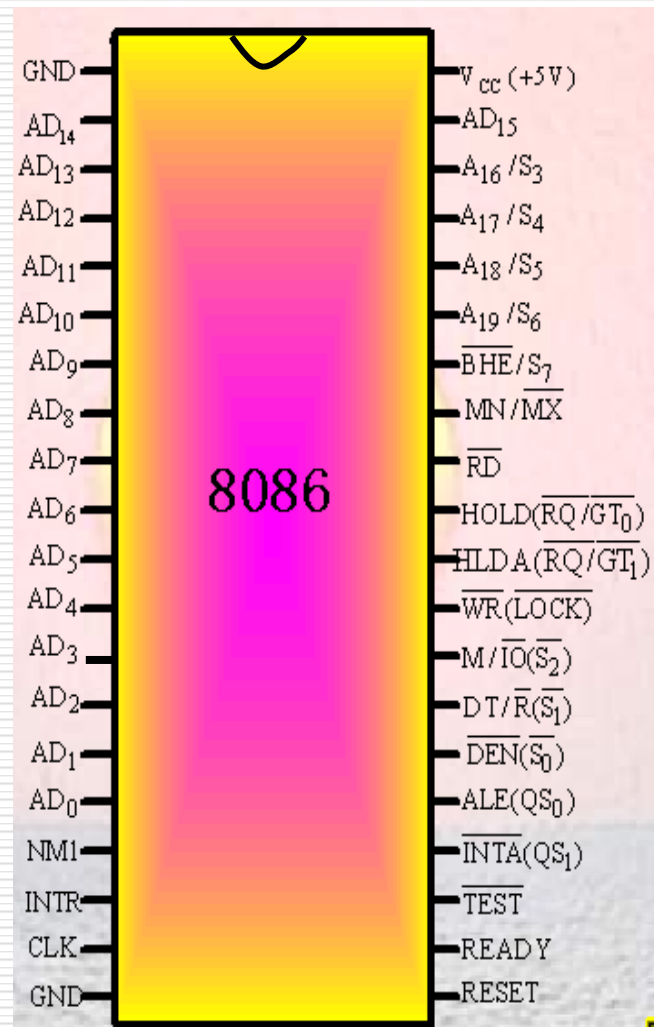
- ☐ ALE:
  - 地址锁存使能信号, 输出
  - 用于将地址锁存在地址锁存器中;
- ☐  $\overline{\text{DEN}}$ :
  - 数据使能信号, 输出
  - 用于控制数据总线驱动器;
- ☐  $\text{DT}/\overline{\text{R}}$ :
  - 数据总线驱动器流向控制, 输出;
- ☐  $\text{M}/\overline{\text{IO}}$ :
  - 存储器或I/O访问控制信号, 输出;
- ☐  $\overline{\text{RD}}$ :
  - 对存储器或I/O端口读信号, 输出;
- ☐  $\overline{\text{WR}}$ :
  - 对存储器或I/O端口写信号, 输出;



## 2.3 8086的引脚信号定义-最小模式下的引脚定义

控制信号与系统信号:

- ☐ **NMI:**
  - 非屏蔽中断请求, 输入
  - 当上升沿到达后, CPU响应中断请求;
- ☐ **INTR:**
  - 可屏蔽中断请求, 输入
  - 当高电平且IF=1, CPU响应中断请求;
- ☐  **$\overline{\text{INTA}}$ :**
  - CPU中断响应信号, 输出
- ☐ **HOLD:**
  - 外设总线请求, 输入, 高电平有效;
- ☐ **HLDA:**
  - 外设总线请求响应, 输出, 高电平有效;
- ☐ **MN/ $\overline{\text{MX}}$ :**
  - 最小/最大模式控制信号, 输入;





## 2.3 8086的引脚信号定义-最小模式下的引脚定义

控制信号与系统信号:

### ❑ RESET:

- 复位信号, 输入
- 清零 FLAGS、IP、DS、SS、ES 和指令队列, CS 设置为 FFFFH;

### ❑ READY:

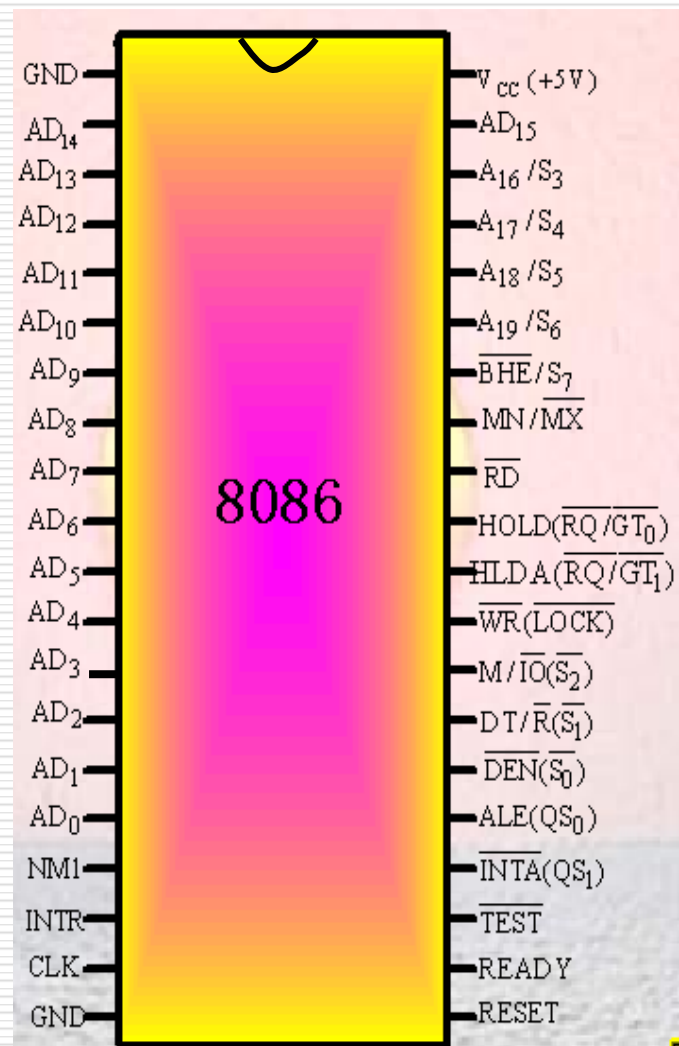
- 存储器或I/O准备好信号, 输入
- 用于CPU访问较慢的存储器或I/O时,
- 延长读写周期;

### ❑ $\overline{\text{TEST}}$ :

- 等待测试信号, 输入
- 与WAIT指令结合使用, 用于主处理器与协处理器的同步;

### ❑ CLK:

- CPU时钟信号, 输入, 占空比为1:3;



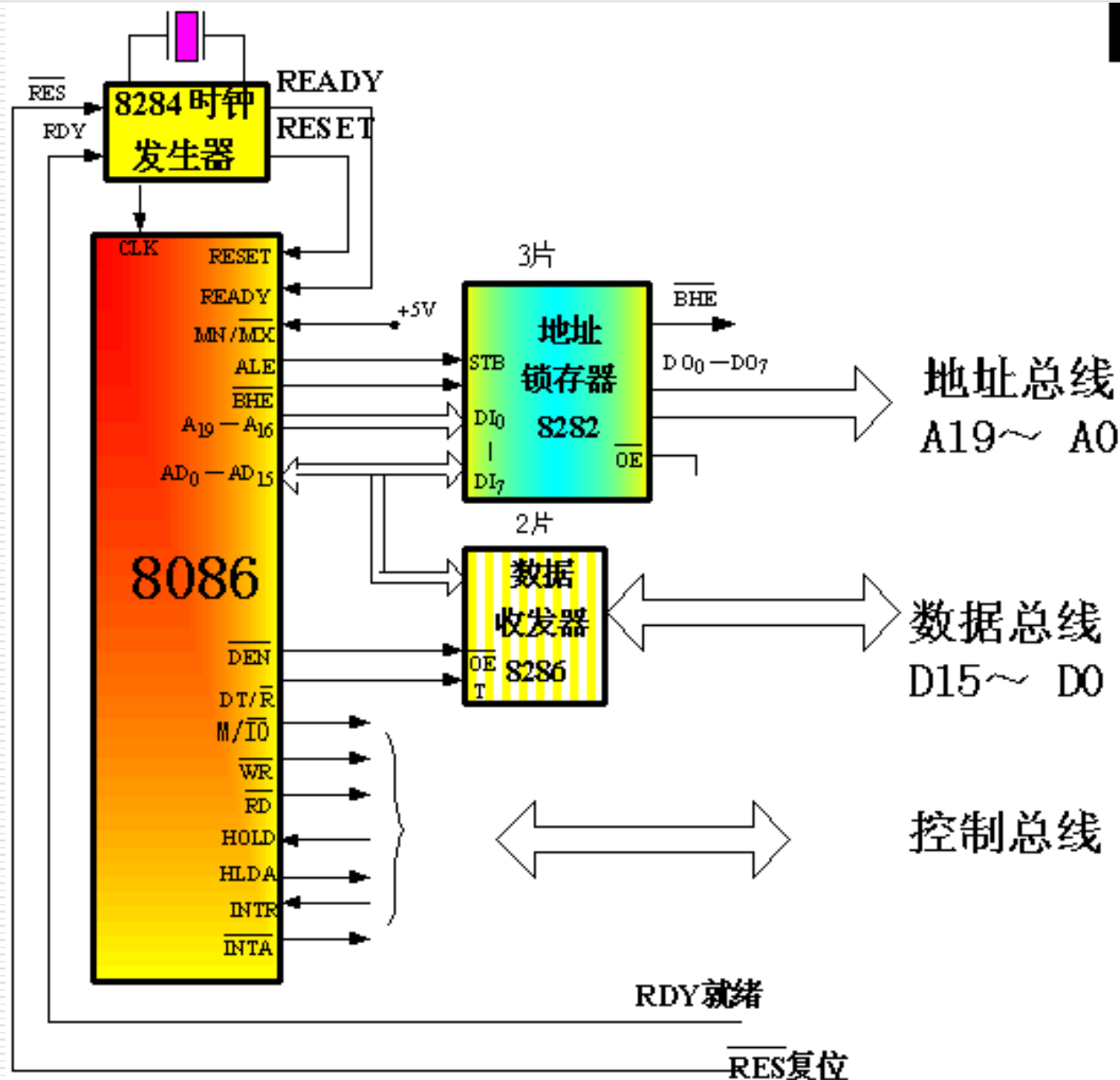
## 2.4 8086系统总线的形成-最小模式

---

所谓最小模式，就是系统中只有一个8086/8088微处理器，在这种情况下，所有的总线控制信号，都是直接由CPU产生的，系统中的总线控制逻辑电路被减到最少，该模式适用于小规模微机应用系统。

- ❑  $\overline{MN}/\overline{MX}$ 端接+5V，决定了工作模式；
- ❑ 有一片8284A，作为时钟信号发生器；
- ❑ 有三片8282或74LS273，用来作为地址信号的锁存器；
- ❑ 当系统中所连的存储器和外设端口较多时，需要增加数据总线的驱动能力，这时，需用2片8286/8287作为数据总线收发器。

## 2.4 8086系统总线的形成-最小模式系统配置





习题：

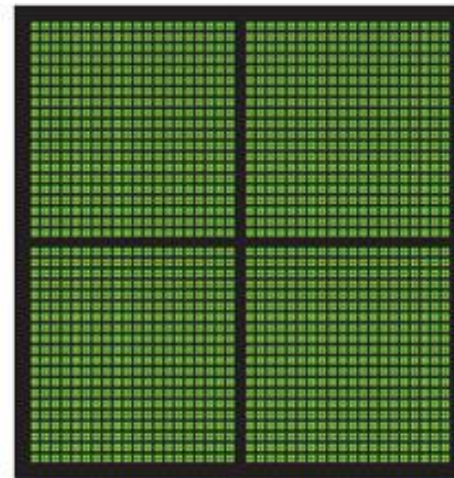
- 1、8086 CPU 内部分为哪两个部分？它们各自的组成和功能是什么？  
这种功能划分为什么可提高CPU的执行速度？
- 2、设  $A=0101010000111001$ ， $B=0100010101101010$ ，对A和B执行加法操作，根据运算结果判断8086 CPU标志寄存器的内容。
- 3、8086 CPU标志寄存器中的状态标志和控制标志有何不同？程序中如何利用这两种标志？
- 4、在一个程序段开始执行之前， $CS=0A8EH$ ， $IP=2A40H$ ，则该程序段的第一个字的物理地址是多少？
- 5、若最小模式下的8086引脚  $M/\overline{IO}=0$ 、 $DT/\overline{R}=1$ ，则表示CPU正在进行什么操作？
- 6、为什么8086 CPU采用地址/数据分时复用？组成最小模式系统时，如何实现该分时复用？

# CPU (Central Processing Unit) 与 GPU (Graphic Processing Unit)

---



CPU  
MULTIPLE CORES



GPU  
THOUSANDS OF CORES

# 8086存储器的分段结构

