



# 第四章 微机总线



### 学习重点

- ◆最小组态下的基本引脚
- ◆最小组态下的总线形成
- ◆最小组态下的总线时序

## 4.1 总线技术

### 4.1.1 总线类型

### 4.1.2 总线的数据传输

## 4.2 8088的引脚信号

◇ 处理器的外部特性表现在其引脚信号上，学习时请特别关注以下几个方面：

- (1) 引脚的功能
- (2) 信号的流向
- (3) 有效电平
- (4) 三态能力

指引脚信号的定义、作用；  
通常采用英文单词或其缩写表示

信号从芯片向外输出，  
起作用的逻辑电平  
高、低电平有效  
上升、下降边沿有效

输出正常的低电平、高电平外，还可以输出高阻的第三态

8088

# 引脚信号的标识方法

输入引脚

输出引脚

高电平有效或编码信号

HOLD

低电平有效

$\overline{\text{TEST}}$

上升沿有效

NMI

下降沿有效

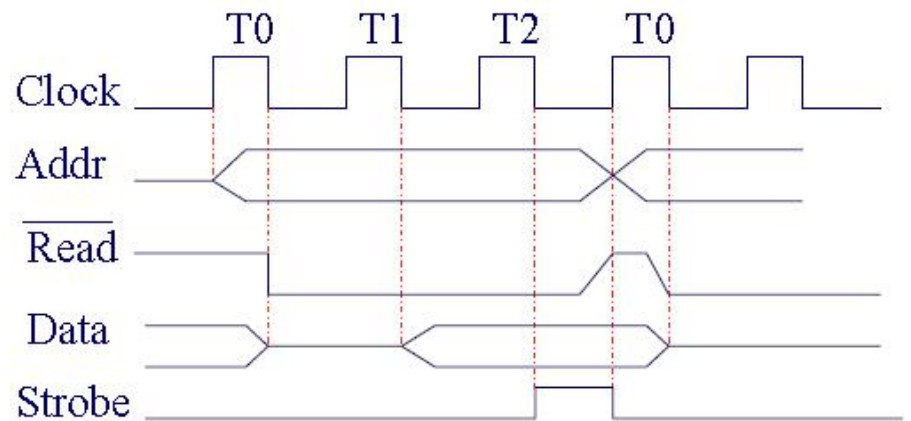
$\overline{\text{OE}}$

HLDA

高电平有效（或编码信号）

$\overline{\text{INTA}}$

低电平有效



## 4.2.1 8088的两种组态模式

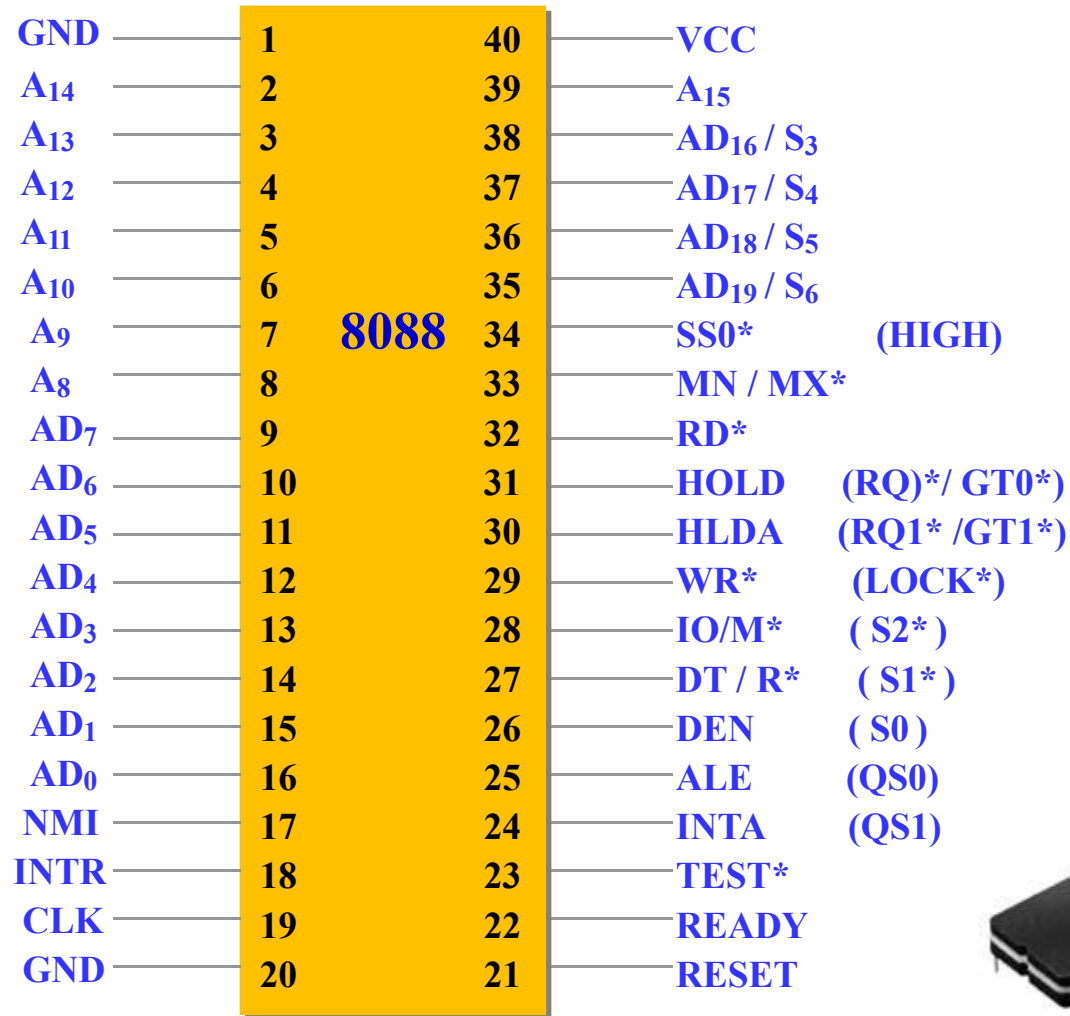
- ◇ 两种组态构成两种不同规模的应用系统
- ◇ 最小组态模式
  - ◆ 构成小规模的应用系统
  - ◆ 8088本身提供所有的系统总线信号
- ◇ 最大组态模式
  - ◆ 构成较大规模的应用系统，例如可以接入数值协处理器8087
  - ◆ 8088和总线控制器8288共同形成系统总线信号

### 4.1.1 8088的两种组态模式（续）

- ◇ 两种组态利用MN/MX\*引脚区别
  - ◆ MN/MX\*接高电平为最小组态模式
  - ◆ MN/MX\*接低电平为最大组态模式
- ◇ 两种组态下的内部操作并没有区别
  - ◆ IBM PC/XT采用最大组态
  - ◆ 课程中以最小组态进行基本原理的学习

通常在信号名称加上划线（如： $\overline{\text{MX}}$ ）或星号（如： $\text{MX}^*$ ）表示低电平有效

# 8088的引脚图





## 4.1.2 最小组态的引脚信号

分类学习这40个引脚（总线）信号

1. 数据和地址引脚
2. 读写控制引脚
3. 中断请求和响应引脚
4. 总线请求和响应引脚
5. 其它引脚



# 1. 数据和地址引脚

## $AD_7 \sim AD_0$ (Address/Data)

- ◇ 地址/数据分时复用引脚，双向、三态
- ◇ 在访问存储器或外设的总线操作周期中，这些引脚在第一个时钟周期输出存储器或I/O端口的低8位地址 $A_7 \sim A_0$
- ◇ 其他时间用于传送8位数据 $D_7 \sim D_0$

引脚图

最小系统

RAM读周期

## 1. 数据和地址引脚 (续1)

$A_{15} \sim A_8$  (Address)

- ◇ 中间8位地址引脚，输出、三态
- ◇ 这些引脚在访问存储器或外设时，提供全部20位地址中的中间8位地址 $A_{15} \sim A_8$

引脚图

最小系统

RAM读周期

## 1. 数据和地址引脚 (续2)

$A_{19}/S_6 \sim A_{16}/S_3$  (Address/Status)

- ◇ 地址/状态分时复用引脚，输出、三态
- ◇ 这些引脚在访问存储器的第一个时钟周期输出高4位地址  $A_{19} \sim A_{16}$
- ◇ 在访问外设的第一个时钟周期全部输出低电平无效
- ◇ 其他时间输出状态信号  $S_6 \sim S_3$

引脚图

最小系统

RAM读周期

## 状态信号S6 ~ S3

S6	未定义，始终为低电平	
S5	指示中断允许标志位IF的当前状态	
S3	S4	段寄存器使用状态
0	0	当前正在使用ES
0	1	当前正在使用SS
1	0	当前正在使用CS或未使用段寄存器
1	1	当前正在使用DS



## 2. 读写控制引脚

### ALE (Address Latch Enable #25)

- ◇ 地址锁存允许，输出、三态、高电平有效
- ◇ ALE引脚高有效时，表示复用引脚： $AD_7 \sim AD_0$ 和 $A_{19}/S_6 \sim A_{16}/S_3$ 正在传送地址信息
- ◇ 由于地址信息在这些复用引脚上出现的时间很短暂，所以系统可以利用ALE引脚将地址锁存起来

引脚图

最小系统

RAM读周期

## 2. 读写控制引脚 (续1)

### IO/M\* (Input and Output/Memory #28)

- ◇ I/O或存储器访问，输出、三态
- ◇ CPU访问I/O端口时，该引脚输出高电平，这时地址总线 $A_{15} \sim A_0$ 提供16位I/O口地址
- ◇ CPU访问存储器时，该引脚输出低电平，这时地址总线 $A_{19} \sim A_0$ 提供20位存储器地址

注：在8086上该引脚定义为M/IO\*。

引脚图

最小系统

RAM读周期

## 2. 读写控制引脚 (续2)

### WR\* (Write #29)

- ◇ 写控制，输出、三态、低电平有效
- ◇ 有效时，表示CPU正在写出数据给存储器或I/O端口

### RD\* (Read #32)

- ◇ 读控制，输出、三态、低电平有效
- ◇ 有效时，表示CPU正在从存储器或I/O端口读入数据

引脚图

最小系统

RAM读周期



## 2. 读写控制引脚 (续3)

- ◇ IO/M\*、WR\*和RD\*是最基本的控制信号
- ◇ 组合后，控制4种基本的总线周期

总线周期	IO/M*	WR*	RD*
存储器读	低	高	低
存储器写	低	低	高
I/O读	高	高	低
I/O写	高	低	高

## 第4章：2. 读写控制引脚（续4）

### READY(#22)

- ◇ 存储器或I/O口就绪，输入、高电平有效
- ◇ 总线操作周期中，CPU会测试该引脚
  - ◆ 如果测到高有效，CPU直接进入下一步
  - ◆ 如果测到无效，CPU将插入等待周期
- ◇ 等待周期中仍然要监测READY信号，确定是否继续插入等待周期

引脚图

最小系统

等待周期

## 2. 读写控制引脚 (续5)

### DEN\* (Data Enable) 26#

引脚图

- ◇ 数据允许，输出、三态、低电平有效
- ◇ 有效时，表示当前数据总线上正在传送数据，可利用他来控制对数据总线的驱动

### DT/R\* (Data Transmit/Receive) 27#

8286

- ◇ 数据发送/接收，输出、三态
- ◇ 该信号表明当前总线上数据的流向
  - ◆ 高电平时数据自CPU输出（发送）
  - ◆ 低电平时数据输入CPU（接收）

最小组态

## 2. 读写控制引脚 (续6)

### SS0\* (System Status 0) #34

- ◇ 最小组态模式下的状态输出信号
- ◇ 它与IO/M\*和DT/R\*一道，通过编码指示8088在最小组态下的8种工作状态。
  1. 取指
  2. 存储器读
  3. 存储器写
  4. 过渡状态
  5. 中断响应
  6. I/O读
  7. I/O写
  8. 暂停

注：8086通过M/IO\*，RD\*，WR\*，DT/R\*组合来决定总线周期的操作类型。

引脚图

最小组态

DT/R*	IO/M*	SS0*	总线操作
0	0	0	取指令
0	0	1	读存储器
0	1	0	中断相应
0	1	1	读I/O□
1	0	0	写存储器
1	0	1	无操作
1	1	0	写I/O□
1	1	1	暂停



# Intel 8088 datasheet

$\overline{\text{IOM}}$	$\overline{\text{DTR}}$	$\overline{\text{SSO}}$	Characteristics
1(HIGH)	0	0	Interrupt Acknowledge
1	0	1	Read IO Port
1	1	0	Write IO Port
1	1	1	Halt
0(LOW)	0	0	Code Access
0	0	1	Read Memory
0	1	0	Write Memory
0	1	1	Passive

Minimum  
mode

$\overline{\text{S2}}$	$\overline{\text{S1}}$	$\overline{\text{S0}}$	Characteristics
0(LOW)	0	0	Interrupt Acknowledge
0	0	1	Read IO Port
0	1	0	Write IO Port
0	1	1	Halt
1(HIGH)	0	0	Code Access
1	0	1	Read Memory
1	1	0	Write Memory
1	1	1	Passive

Maximum  
mode

### 3. 中断请求和响应引脚

#### INTR (Interrupt Request) #18

- ◇ 可屏蔽中断请求，输入、高电平有效
- ◇ 有效时，表示设备向CPU申请可屏蔽中断
- ◇ 该中断请求是否响应受控于IF（中断允许标志）、可以被屏蔽掉

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

[引脚图](#)[最小组态](#)

### 3. 中断请求和响应引脚 (续1)

#### INTA\* (Interrupt Acknowledge) 24#

- ◇ 可屏蔽中断响应，输出、低电平有效
- ◇ 有效时，表示来自INTR引脚的中断请求已被CPU响应，CPU进入中断响应周期
- ◇ 通常被用来作为中断类型码的读选通信号

引脚图

最小组态



### 3. 中断请求和响应引脚 (续2)

#### NMI (Non-Maskable Interrupt) 17#

[引脚图](#)[最小组态](#)

- ◇ 不可屏蔽中断请求，输入、上升沿有效
- ◇ 有效表示外界向CPU申请不可屏蔽中断
- ◇ 该中断请求不能被CPU屏蔽，所以优先级别高于INTR (可屏蔽中断)

主机与外设进行数据交换通常采用可屏蔽中断  
不可屏蔽中断通常用于处理掉电等系统故障

## 4. 总线请求和响应引脚

### HOLD #31

- ◇ 总线保持（即总线请求），输入、高电平有效
- ◇ 有效时，表示设备向**CPU**申请占有总线
- ◇ 该信号从有效回到无效时，表示设备对总线的使用已经结束，通知**CPU**收回对总线的控制权

引脚图

最小组态

## 4. 总线请求和响应引脚 (续1)

### HLDA (HOLD Acknowledge) 30#

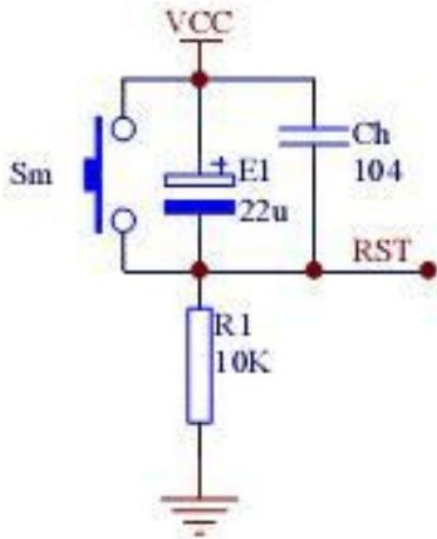
- ◇ 总线保持响应，输出、高电平有效
- ◇ 有效表示**CPU**已响应总线请求并已将总线释放
- ◇ 此时**CPU**的地址总线、数据总线及具有三态输出能力的控制总线将全面呈现高阻，使总线请求设备可以顺利接管总线
- ◇ 待到总线请求信号**HOLD**无效，总线响应信号**HLDA**也转为无效，**CPU**重新获得总线控制权

[引脚图](#)[最小组态](#)

## 5. 其它引脚

### RESET 21#

- ◇ 复位请求，输入、高电平有效
- ◇ 该信号有效，将使CPU回到其初始状态；当该信号再度返回无效时，CPU将重新开始工作
- ◇ 8088复位后CS = FFFFH、IP = 0000H，所以程序入口在物理地址FFFF0H



引脚图

最小组态

# 8086/8088复位时各寄存器值

03:4 PM 144

寄存器	值	寄存器	值
FLAGS	0000H	DS	0000H
IP	0000H	ES	0000H
CS	FFFFH	SS	0000H
指令队列	空	其余寄存器	0000H

## 5. 其它引脚 (续1)

### CLK (Clock) 19#

- ◇ 时钟输入
- ◇ 系统通过该引脚给CPU提供内部定时信号
  - ◆ 8088的标准工作时钟为5MHz
  - ◆ IBM PC/XT机的8088采用了4.77MHz的时钟，其时钟周期约为210ns

引脚图

最小组态

## 5. 其它引脚 (续2)

### Vcc

- ◇ 电源输入，向CPU提供 + 5V电源

### GND

- ◇ 接地，向CPU提供参考地电平

### MN/MX\* (Minimum/Maximum)

- ◇ 组态选择，输入
- ◇ 接高电平时，8088工作在最小组态；反之，8088工作在最大组态

引脚图

## 5. 其它引脚 (续3)

### TEST\* #23

- ◇ 测试，输入、低电平有效
- ◇ 使用协处理器8087时，通过该引脚和WAIT指令，可使8088与8087的操作保持同步

引脚图

最小组态



## “引脚” 小结

CPU引脚是系统总线的基本信号

可以分成三类信号

- ◇ 8位数据线： $D_0 \sim D_7$
- ◇ 20位地址线： $A_0 \sim A_{19}$
- ◇ 控制线：
  - ◆ ALE、IO/M\*、WR\*、RD\*、READY
  - ◆ INTR、INTA\*、NMI, HOLD、HLDA
  - ◆ RESET、CLK、Vcc、GND

## “引脚” 提问

### ◇ 提问之一

**CPU**引脚是如何与外部连接的呢？

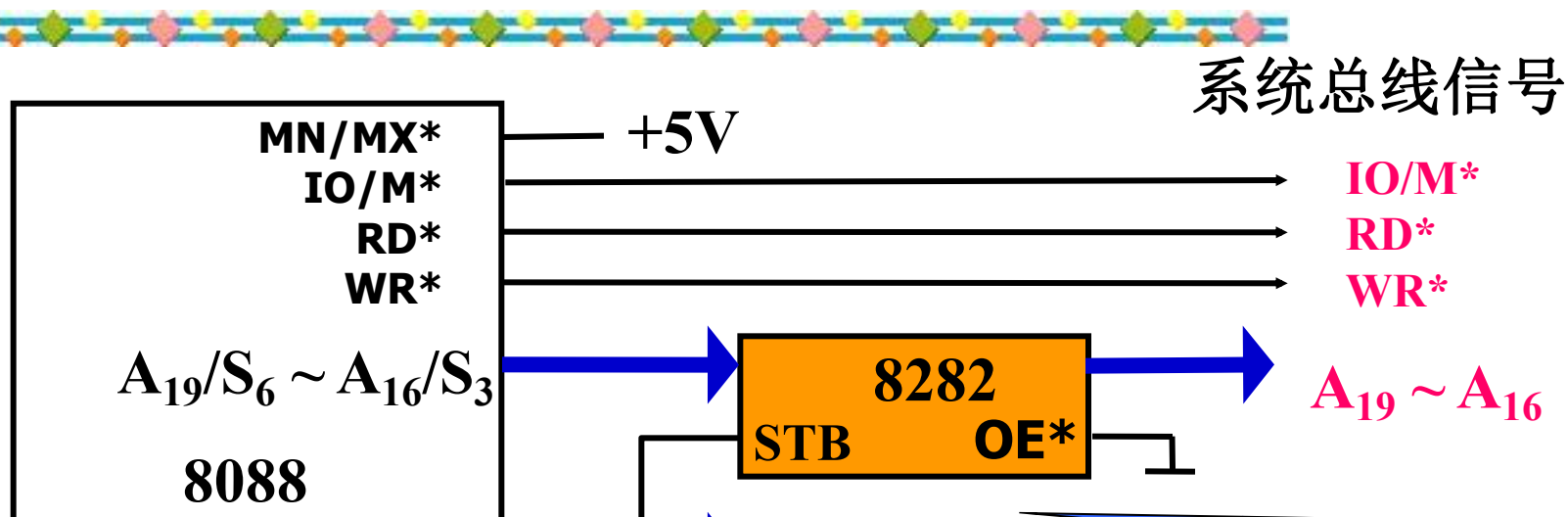
解答：总线形成（第4.3.1节）

### ◇ 提问之二

**CPU**引脚是如何相互配合，实现总线操作、控制系统工作的呢？

解答：总线时序（第4.4节）

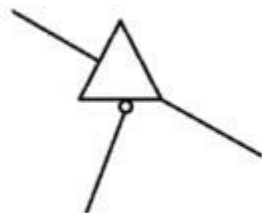
### 4.3.1 最小组态的总线形成



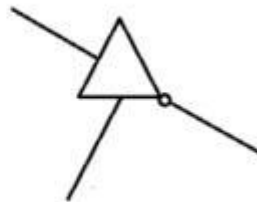
- (1) 20位地址总线——  
采用3个三态透明锁存器8282进行锁存和驱动
- (2) 8位数据总线——  
采用数据收发器8286进行驱动
- (3) 系统控制信号——  
由8088引脚直接提供

## 补充：三态门和D触发器

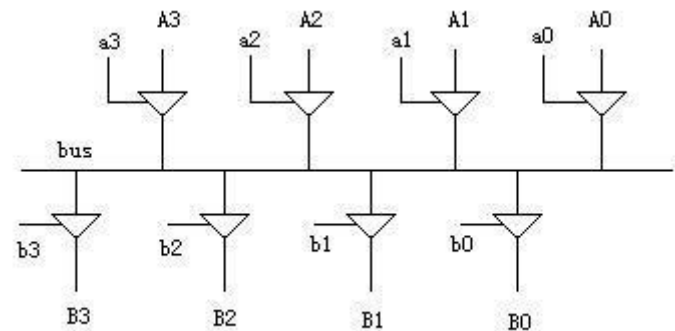
- ◇ 三态门和以D触发器形成的锁存器是微机接口电路中最常使用的两类逻辑电路
- ◇ 三态门的作用：功率放大、导通开关
- ◇ 器件共用总线时，一般使用三态电路：
  - ◆ 需要使用总线的时候打开三态门；
  - ◆ 不使用的时候关闭三态门，使之处于高阻
- ◇ D触发器的作用：信号保持，导通开关



低电平使能

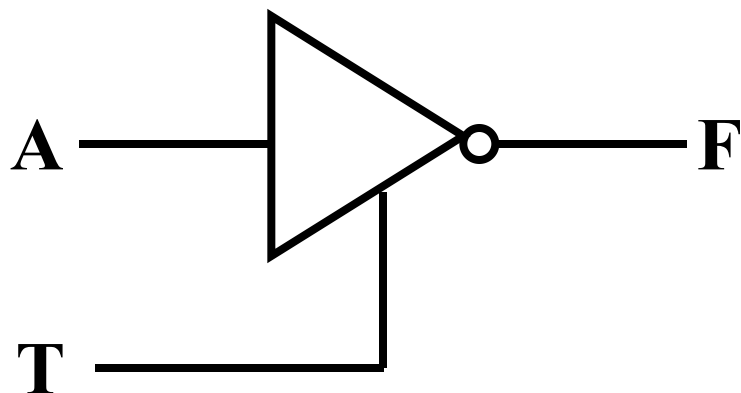


输出取反

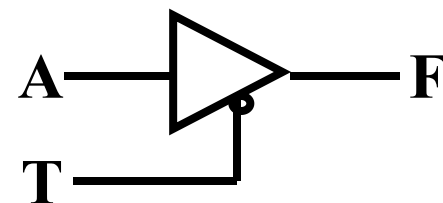
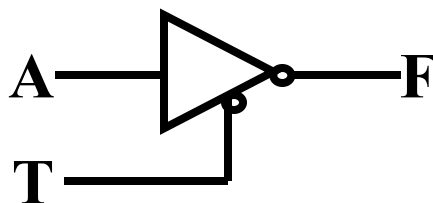
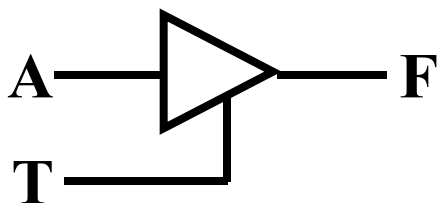


## 三态缓冲器（三态门）

三态门具有单向导通和三态的特性



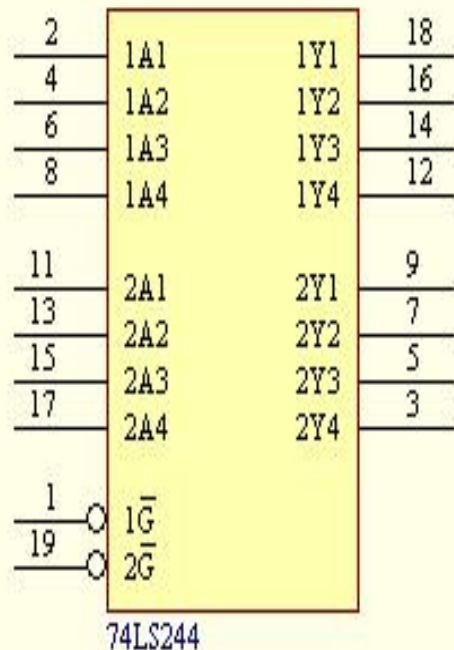
T为低电平时：  
输出为高阻抗（三态）  
T为高电平时：  
输出为输入的反相



○ 表示反相或低电平有效

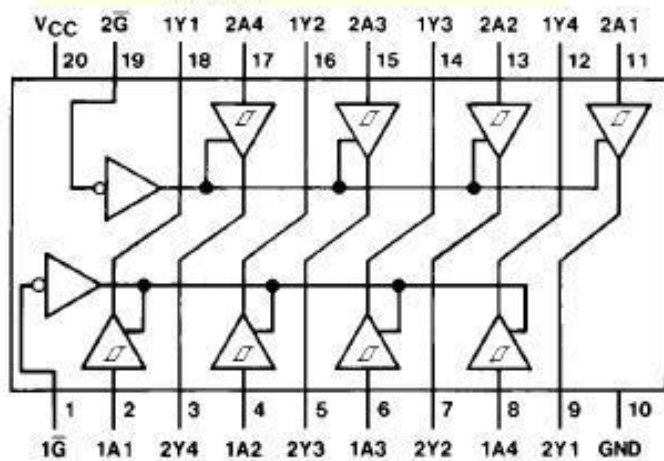


# 常用集成电路芯片 74LS244



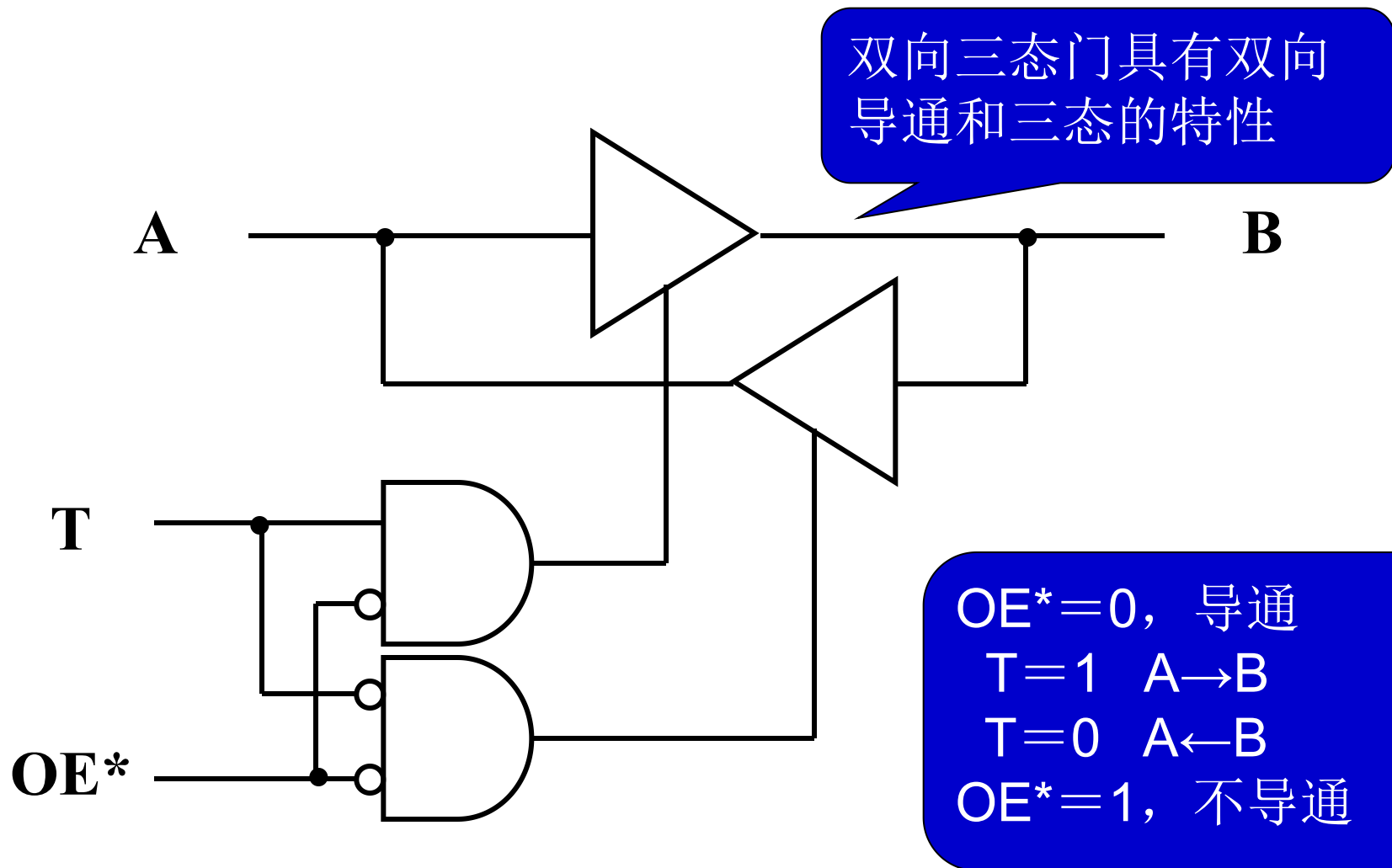
- 双4位单向缓冲器
- 分成4位的两组
- 每组的控制端连接在一起
- 控制端低电平有效
- 输出与输入同相

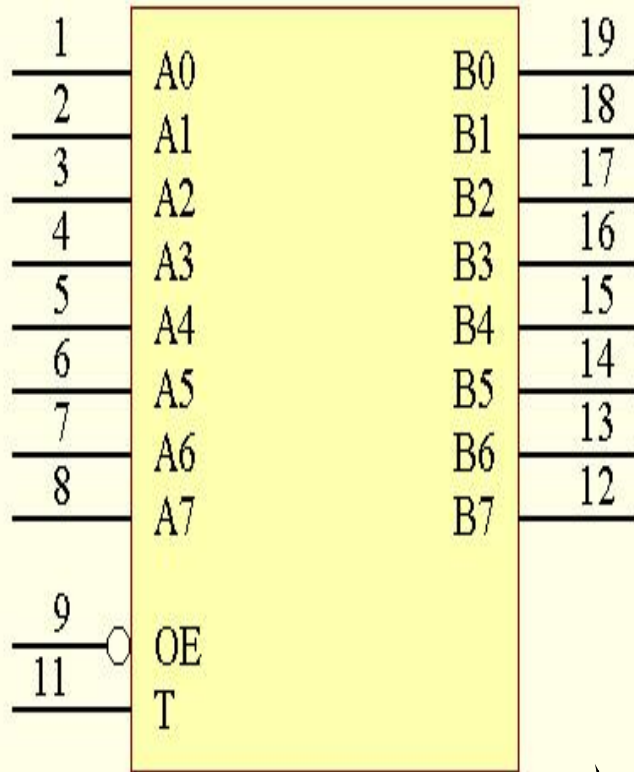
每一位都是一个三态门，  
每4个三态门的控制端连接在一起



# 双向三态缓冲器

03:41 PM 144





8286

## 8位双向缓冲器

- ▶ 控制端连接在一起，低电平有效
- ▶ 可以双向导通
- ▶ 输出与输入同相

$OE^*=0$ ，导通

$T=1$   $A \rightarrow B$

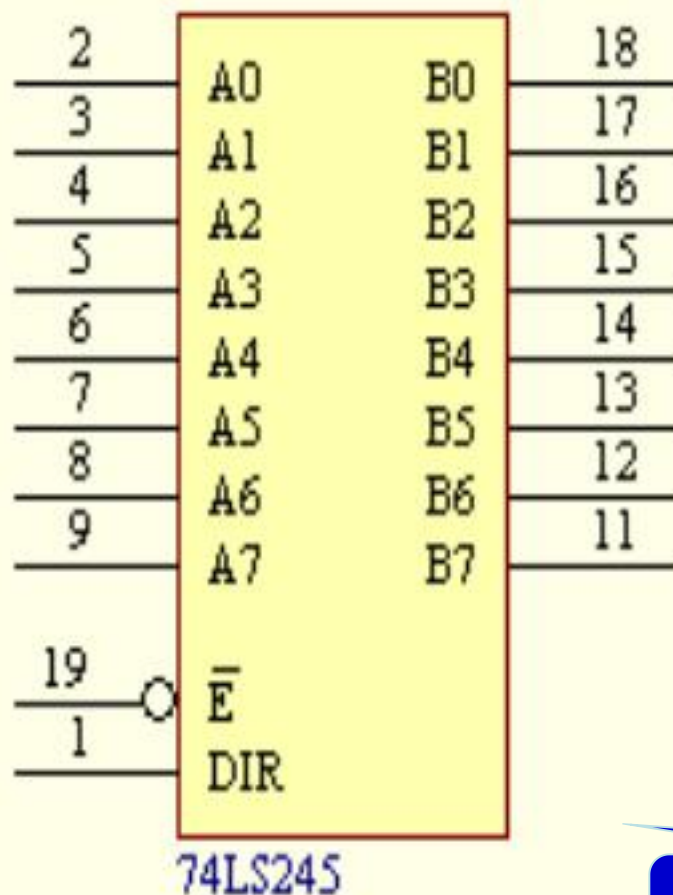
$T=0$   $A \leftarrow B$

$OE^*=1$ ，不导通

每一位都是一个双向三态门，  
8位具有共同的控制端



# 常用集成电路芯片 74LS245



## 8位双向缓冲器

- ▶ 控制端连接在一起，低电平有效
- ▶ 可以双向导通
- ▶ 输出与输入同相

$E^*=0$ ，导通

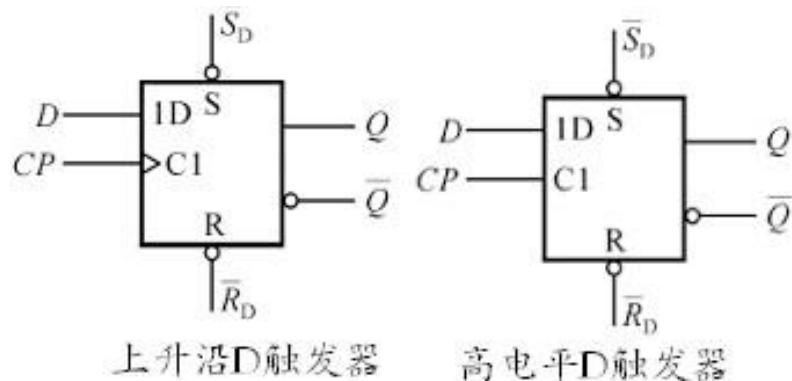
$DIR=1$   $A \rightarrow B$

$DIR=0$   $A \leftarrow B$

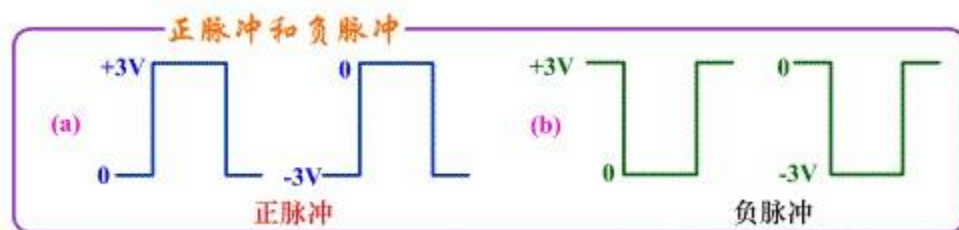
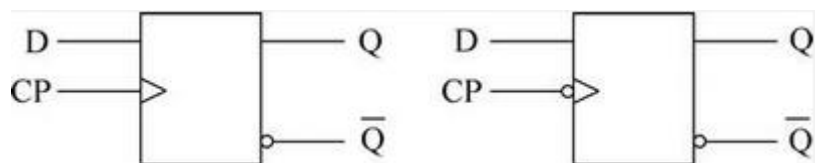
$E^*=1$ ，不导通

74LS245与Intel 8286功能一样

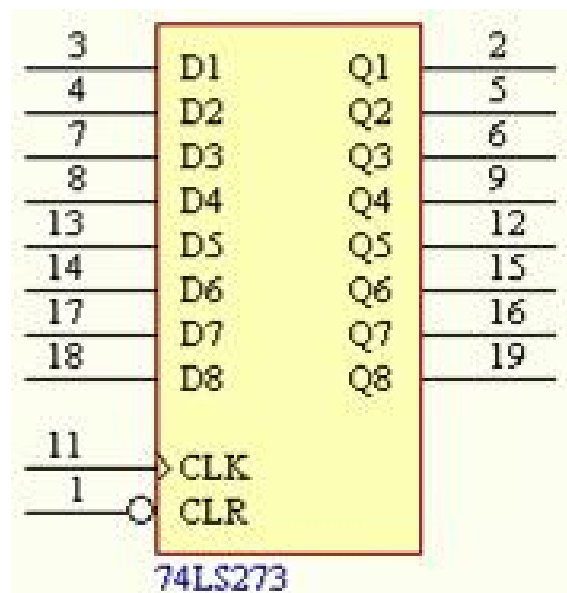
# D触发器



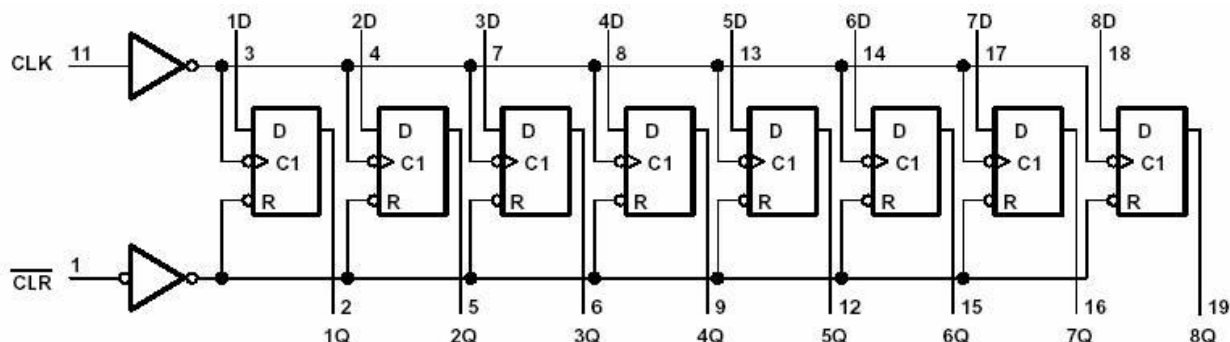
电平锁存：  
高电平通过，低电平锁存  
上升沿锁存：  
通常用负脉冲触发锁存



# 常用集成电路芯片 74LS273



具有异步清零的TTL上升沿锁存器



每一位都是一个D触发器，  
8个D触发器的控制端连接在一起

功能表

输 入			输 出
清零	时钟	D	
CLR	CLK	D	Q
L	X	X	L
H	↑	H	H
H	↑	L	L
H	L	X	Q <sub>0</sub>

H=高电平

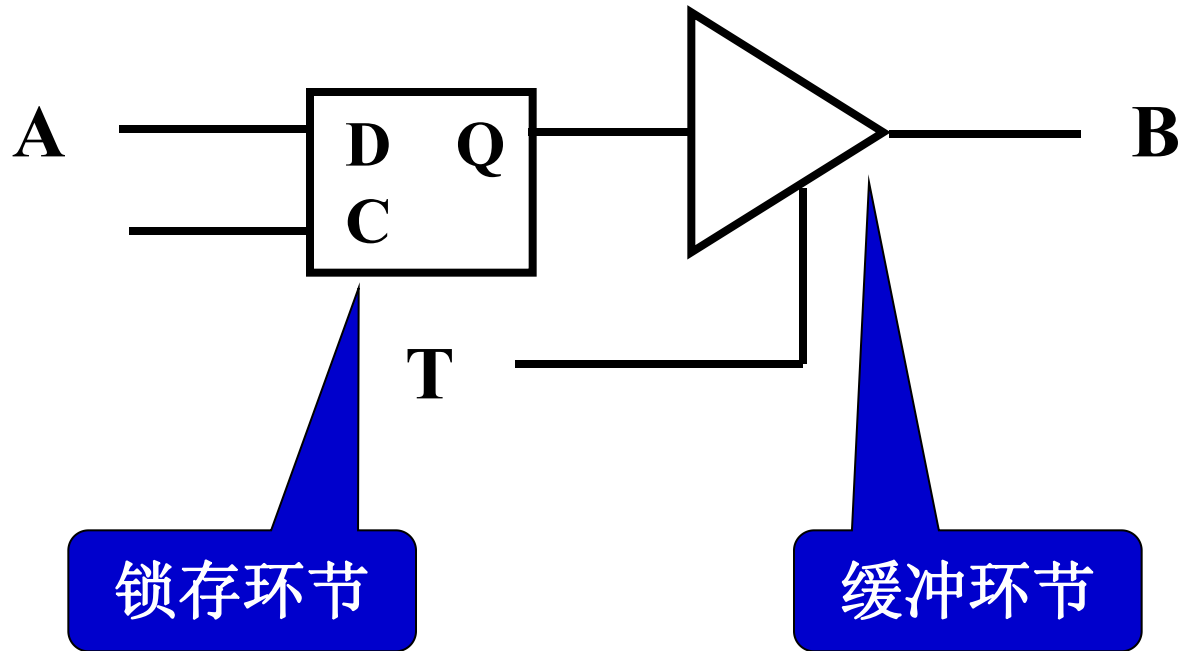
L=低电平

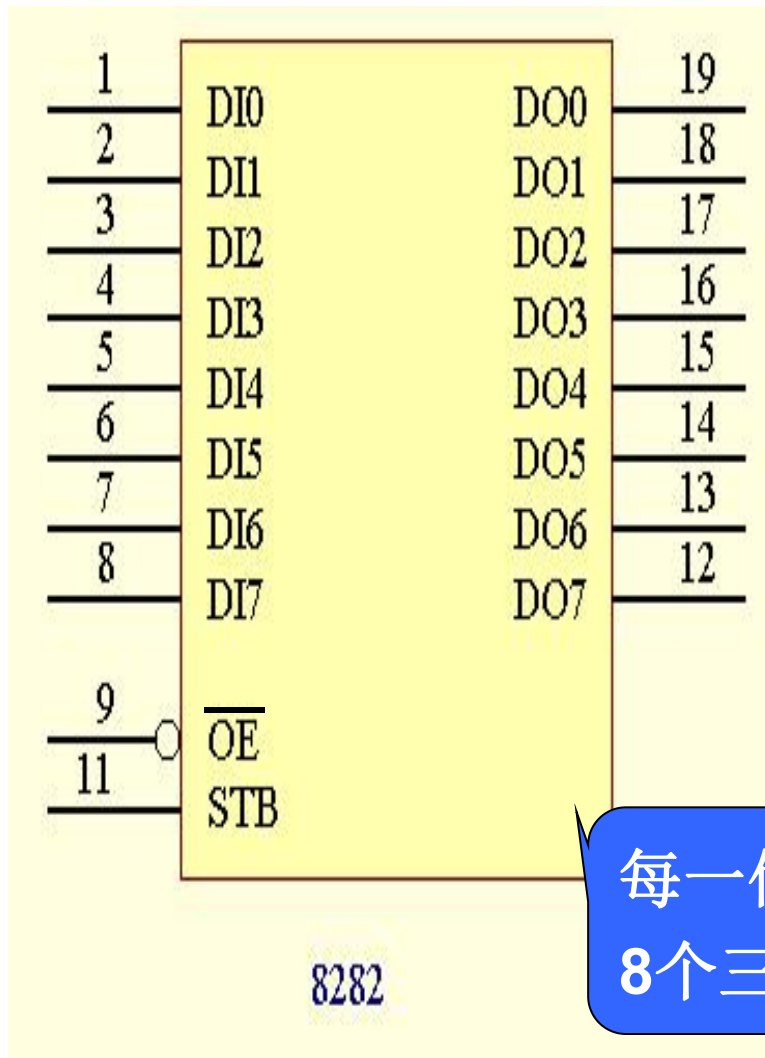
X=不定态

↑=从低电平转到高电平



## 三态缓冲锁存器（三态锁存器）





具有三态输出的  
TTL电平锁存器

STB 电平锁存引脚  
OE\* 输出允许引脚

每一位都是一个三态锁存器，  
8个三态锁存器的控制端连在一起

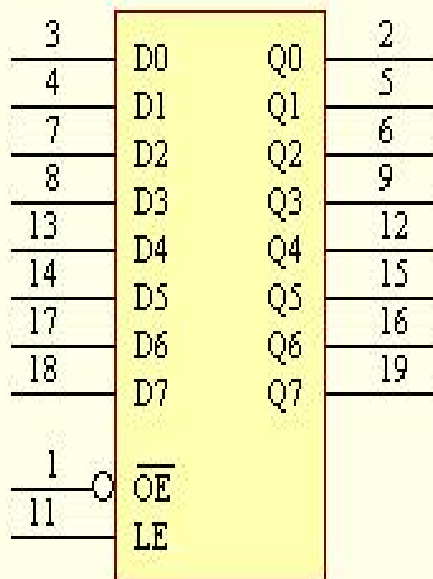
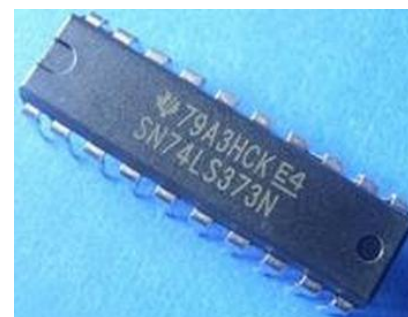
# 第4章：常用集成电路芯片 74LS373

74LS373与Intel 8282功能一样

具有三态输出的TTL电平锁存器

LE 电平锁存引脚

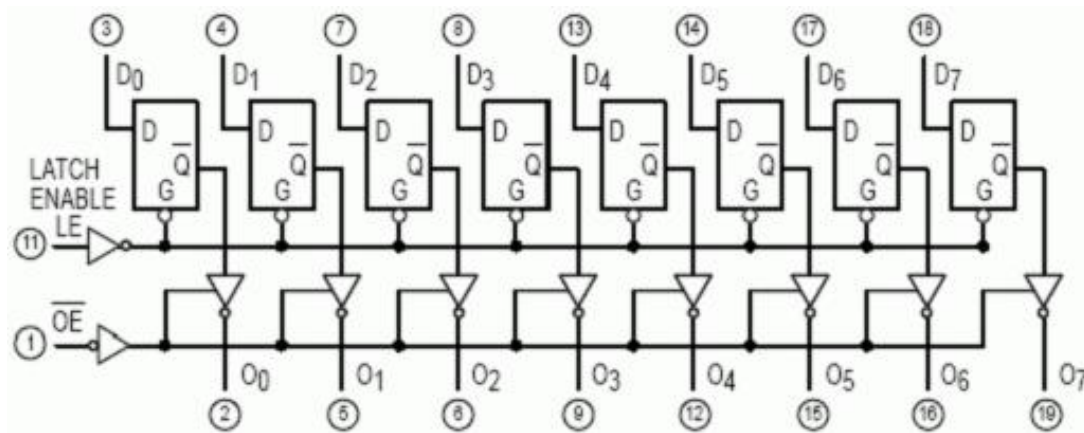
OE\* 输出允许引脚



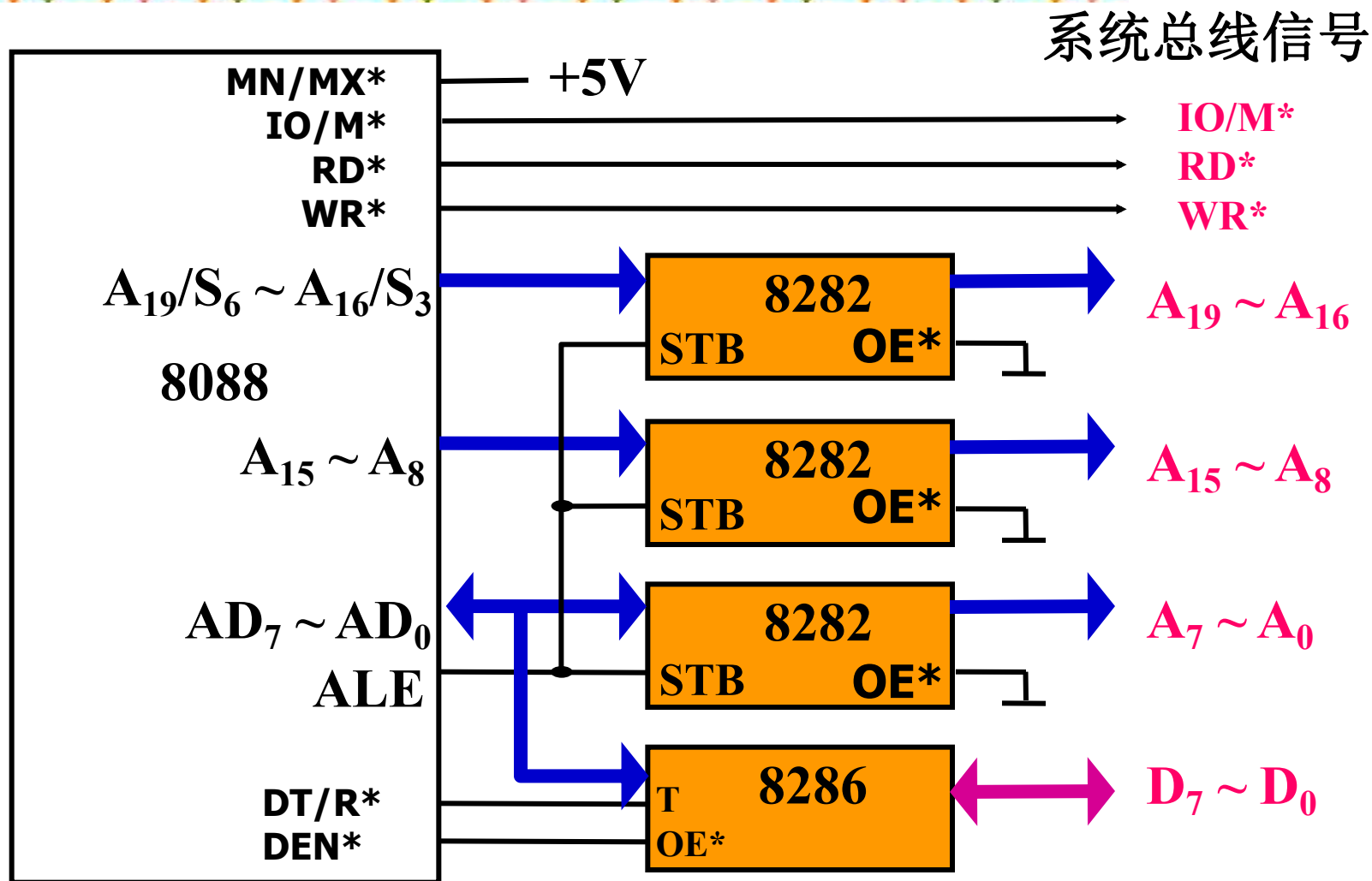
74LS373

DM74LS373

Output Control	Enable G	D	Output
L	H	H	H
L	H	L	L
L	L	X	$Q_0$
H	X	X	Z



### 4.3.1 最小组态的总线形成





## 1. 20位地址总线的形成

- ◇ 采用3个8282进行锁存和驱动
- ◇ Intel 8282是三态透明锁存器，类似有Intel 8283和通用数字集成电路芯片74373
- ◇ 三态输出：
  - ◆ 输出控制信号有效时，允许数据输出；
  - ◆ 输出控制信号无效时，不允许数据输出，呈高阻状态
- ◇ 透明：锁存器的输出能够跟随输入变化



## 2. 8位数据总线的形成

- ◇ 采用数据收发器8286进行双向驱动
- ◇ Intel 8286是8位三态双向缓冲器，类似功能的器件还有Intel 8287、通用数字集成电路74245等
- ◇ 另外，接口电路中也经常使用三态单向缓冲器，例如通用数字集成电路74244就是一个常用的双4位三态单向缓冲器

### 3. 系统控制信号的形成

- ◇ 由8088引脚直接提供
- ◇ 因为基本的控制信号8088引脚中都含有
- ◇ 例如：IO/M\*、WR\*、RD\*等
- ◇ 其它信号的情况看详图

## 4.4 8088的总线时序

- ◇ 时序（Timing）是指信号高低电平（有效或无效）变化及相互间的时间顺序关系
- ◇ CPU时序决定系统各部件间的同步和定时
- ◇ 总线时序描述CPU引脚如何实现总线操作

## 4.2 8088的总线时序 (续1)

- ◇ 总线操作是指CPU通过总线对外的各种操作
- ◇ 8088的总线操作主要有：
  - ◆ 存储器读、I/O读操作
  - ◆ 存储器写、I/O写操作
  - ◆ 中断响应操作
  - ◆ 总线请求及响应操作
  - ◆ CPU正在进行内部操作、并不进行实际对外操作的空闲状态Ti
- ◇ 描述总线操作的微处理器时序有三级
  - ◆ 指令周期 → 总线周期 → 时钟周期

## 4.2 8088的总线时序 (续2)

- ◇ 指令周期是指一条指令经取指、译码、读写操作数到执行完成的过程。
- ◇ 总线周期是指CPU通过总线操作与外部（存储器或I/O端口）进行一次数据交换的过程
- ◇ 若干总线周期组成一个指令周期
- ◇ 8088的基本总线周期需要4个时钟周期
  - ◆ 4个时钟周期编号为 $T_1$ 、 $T_2$ 、 $T_3$ 和 $T_4$
  - ◆ 总线周期中的时钟周期也被称作“T状态”
  - ◆ 时钟周期的时间长度就是时钟频率的倒数
- ◇ 当需要延长总线周期时插入等待状态 $T_w$
- ◇ CPU进行内部操作，没有对外操作时，其引脚就处于空闲状态 $T_i$

演示

## 4.2 8088的总线时序 (续3)

- ◇ 任何指令的取指阶段都需要存储器读总线周期，读取的内容是指令代码
- ◇ 任何一条以存储单元为源操作数的指令都将引起存储器读总线周期，任何一条以存储单元为目的的操作数的指令都将引起存储器写总线周期
- ◇ 执行IN指令出现I/O读总线周期，执行OUT指令出现I/O写总线周期
- ◇ CPU响应可屏蔽中断时生成中断响应总线周期

## 4.2 8088的总线时序 (续4)

- ◇ 总线操作中如何实现时序同步是关键
- ◇ CPU总线周期采用同步时序
  - ◆ 各部件都以系统时钟信号为基准
  - ◆ 当相互不能配合时，快速部件（CPU）插入等待状态等待慢速部件（I/O和存储器）
- ◇ CPU与外设接口常采用异步时序，它们通过应答联络信号实现同步操作

## 4.4.1 最小组态的总线时序

本节展开微处理器最基本的4种总线周期。

- ◆ 存储器读总线周期
- ◆ 存储器写总线周期
- ◆ I/O读总线周期
- ◆ I/O写总线周期



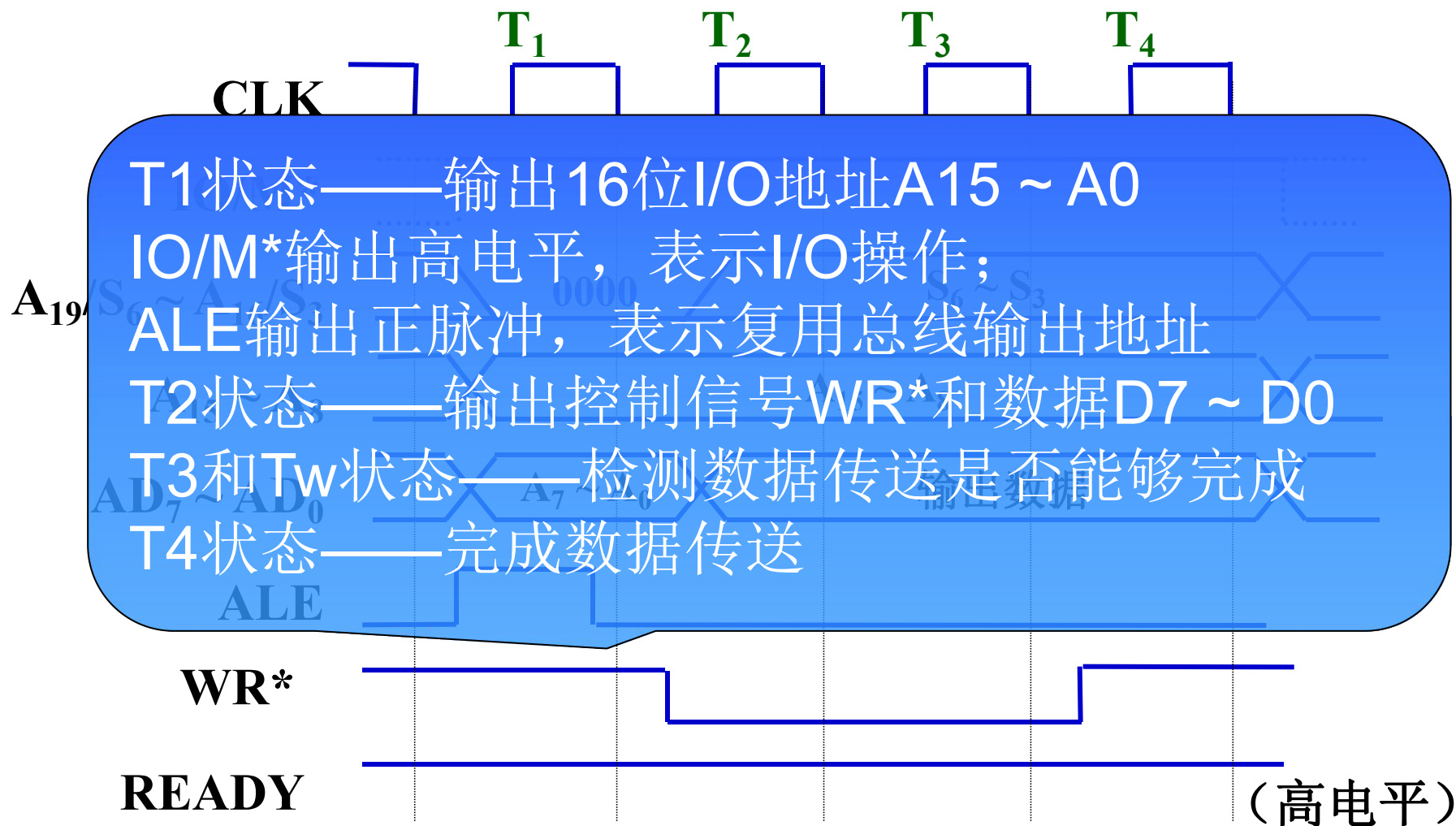
## 第4章：存储器写总线周期

03:4 PM 144



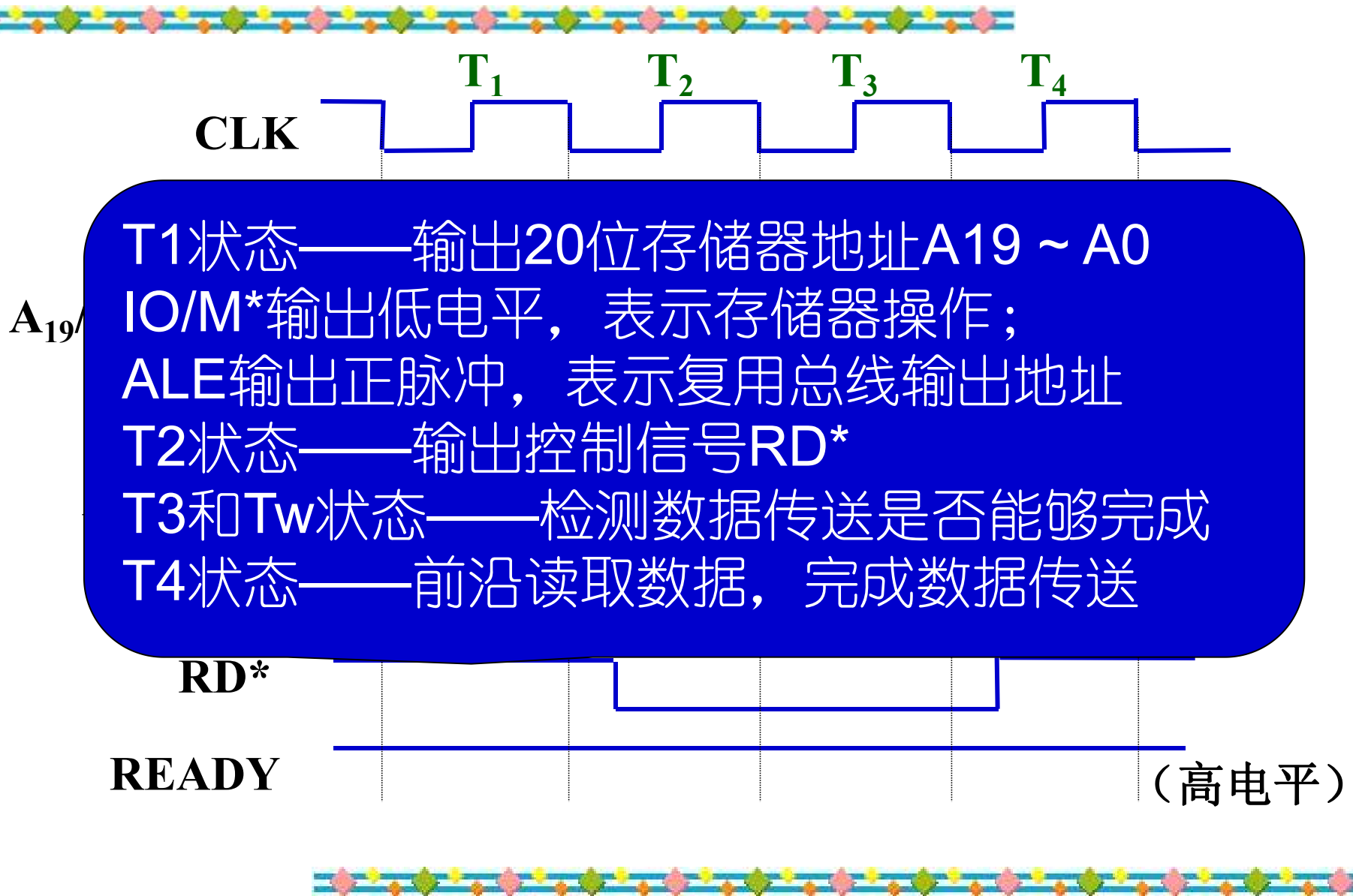
## 第4章：I/O写总线周期

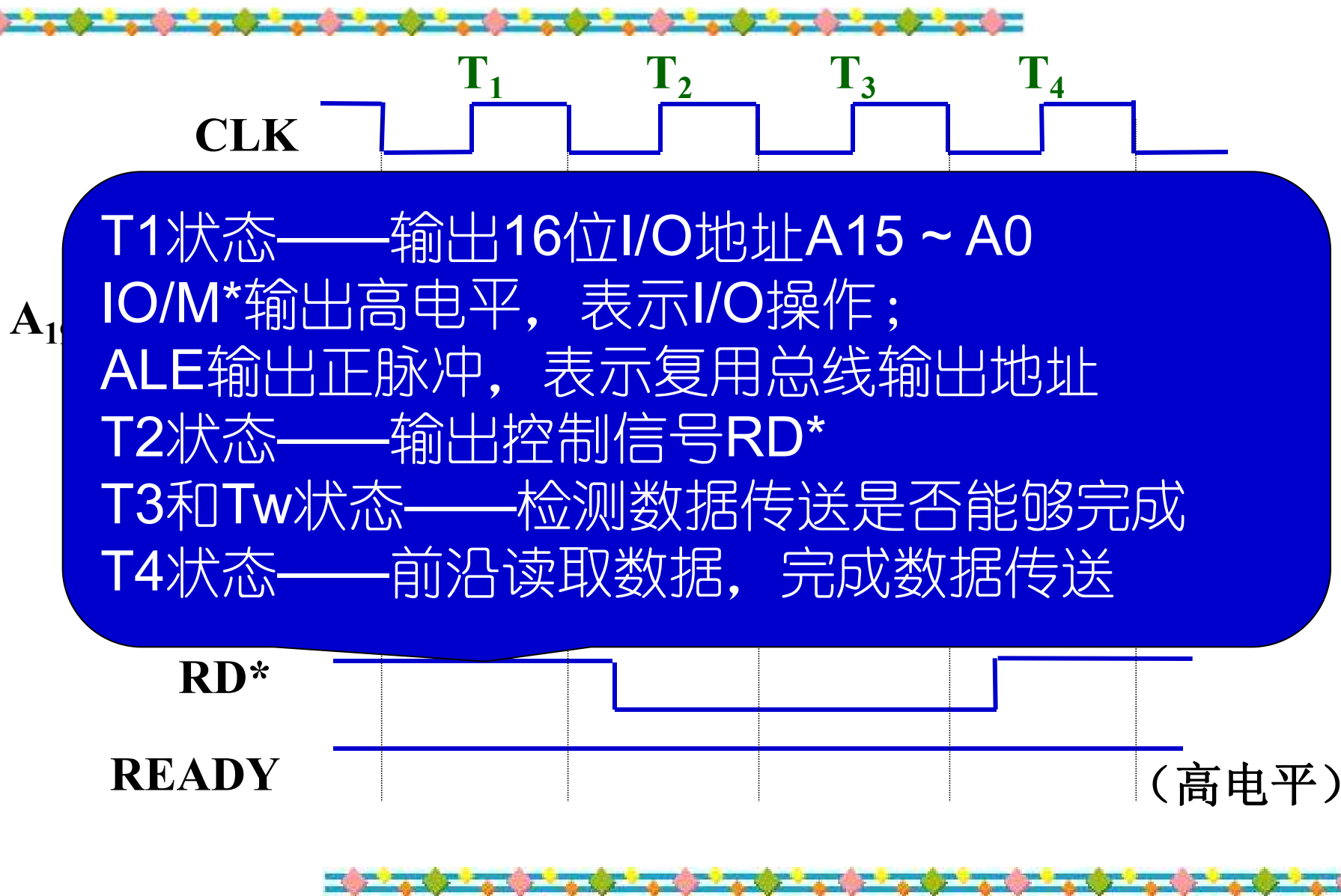
03:4 1PM 144



## 第4章：存储器读总线周期

03:41 PM 144





## 第4章：等待状态Tw

- ◇ 同步时序通过插入等待状态，来使速度差别较大的两部分保持同步
- ◇ 在读写总线周期中，判断是否插入Tw
  1. 在T3的前沿检测READY引脚是否有效
  2. 如果READY无效，在T3和它T4之间插入一个等效于T3的Tw，转1
  3. 如果READY有效，执行完该T状态，进入T4状态



演示

## 4.6 微机系统总线

- ◇ 微机系统采用总线结构。系统中主要部件通过系统总线相互连接、实现数据传输，并使微机系统具有组态灵活、易于扩展等诸多优点
- ◇ 广泛应用的总线都实现了标准化，便于互连各个部件时遵循共同的总线规范。接口的任一方只需要根据总线标准的要求来实现和完成接口的功能，而不必了解对方的接口方式

总线接口也是一种通用的接口技术

## 4.6.1 微机总线概述

- 
- ◇ 总线连接方法广泛用于微机系统的各个连接层次上
    - ◆ 大规模集成电路芯片内部（如微处理器的内部总线）
    - ◆ 主机板中微处理器、存储器及I/O接口电路之间，主机模板与各种接口模板之间
    - ◆ 微机系统之间以及微机系统与外部设备之间
- 

## (1) 芯片总线 (Chip Bus)

- ◇ 指大规模集成电路芯片内部，或一个较小系统中各种不同器件连接在一起的总线；用于芯片级互连
- ◇ 芯片总线也称为局部总线 (Local Bus)
- ◇ 微处理器的引脚信号就是芯片总线
- ◇ 微处理器内部的控制器、运算器、寄存器之间，还有系统主机板上CPU、存储器、接口电路等之间通常就是利用芯片级总线互连的

图示



## (2) 内总线 (Internal Bus)

- ◇ 微机系统中模板与模板间连接的总线，是微机系统所特有的总线；用于模板级互连
- ◇ 内总线也被称为板级总线或系统总线 (System Bus)
- ◇ 多数已实现标准化，例如STD总线、ISA总线等。
- ◇ 微机主板的各种扩展插槽多属于内总线

图示

### (3) 外总线 (External Bus)

- ◇ 微机系统之间或微机系统与其外设通信的总线，用于设备级互连
- ◇ 外总线过去又称为通信总线，主要指串行通信总线，例如RS-232
- ◇ 现在，外总线的意义常延伸为外设总线，主要用于连接各种外设
- ◇ 外总线种类较多，常与特定设备有关，例如Centronics并行打印机总线、IEEE 488智能仪器仪表并行总线（又称为GPIB总线）

图示

## 常见系统总线

- ◇ 在微机发展和应用中出现了许多种内、外总线标准
  - ◆ 第一个标准化的微机总线S-100总线
  - ◆ 面向工业控制领域的STD总线
  - ◆ 32位PC机上的ISA系统总线、EISA总线、VESA总线、PCI总线、USB总线等

图示



本章到此结束  
谢谢

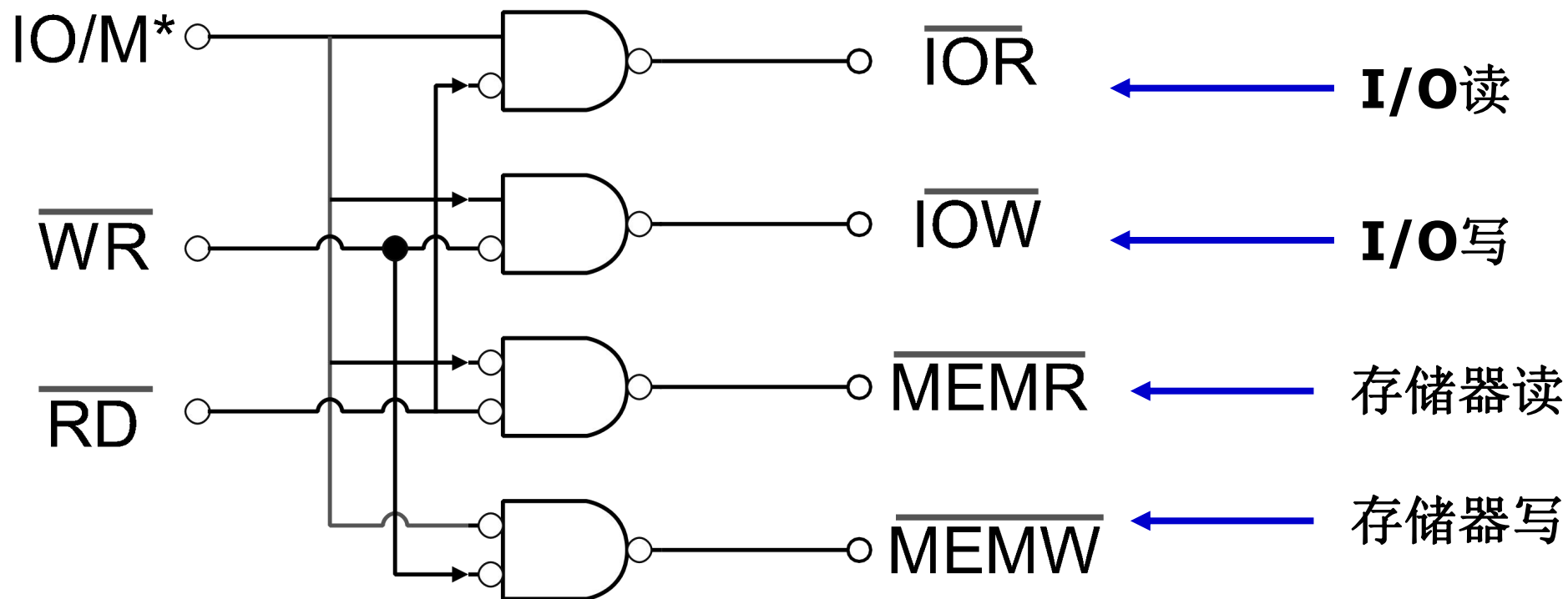


## 第4章：什么是分时复用？

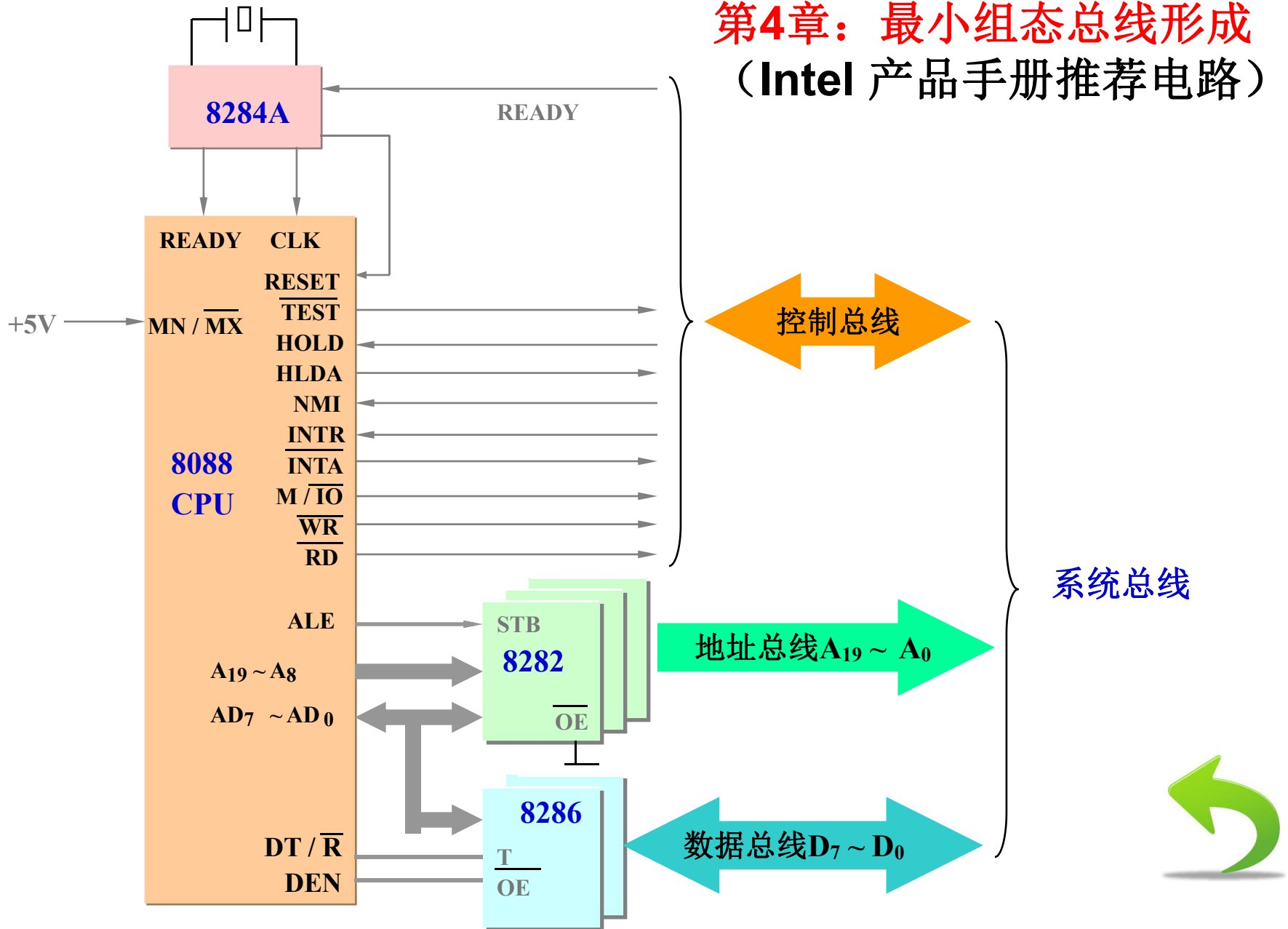
- ◇ 分时复用就是一个引脚在不同的时刻具有两个甚至多个作用
- ◇ 最常见的总线复用是数据和地址引脚复用
- ◇ 总线复用的目的是为了减少对外引脚个数
- ◇ 8088 /8086CPU的数据地址线采用了总线复用方法



## 第4章：基本控制信号的组合方法



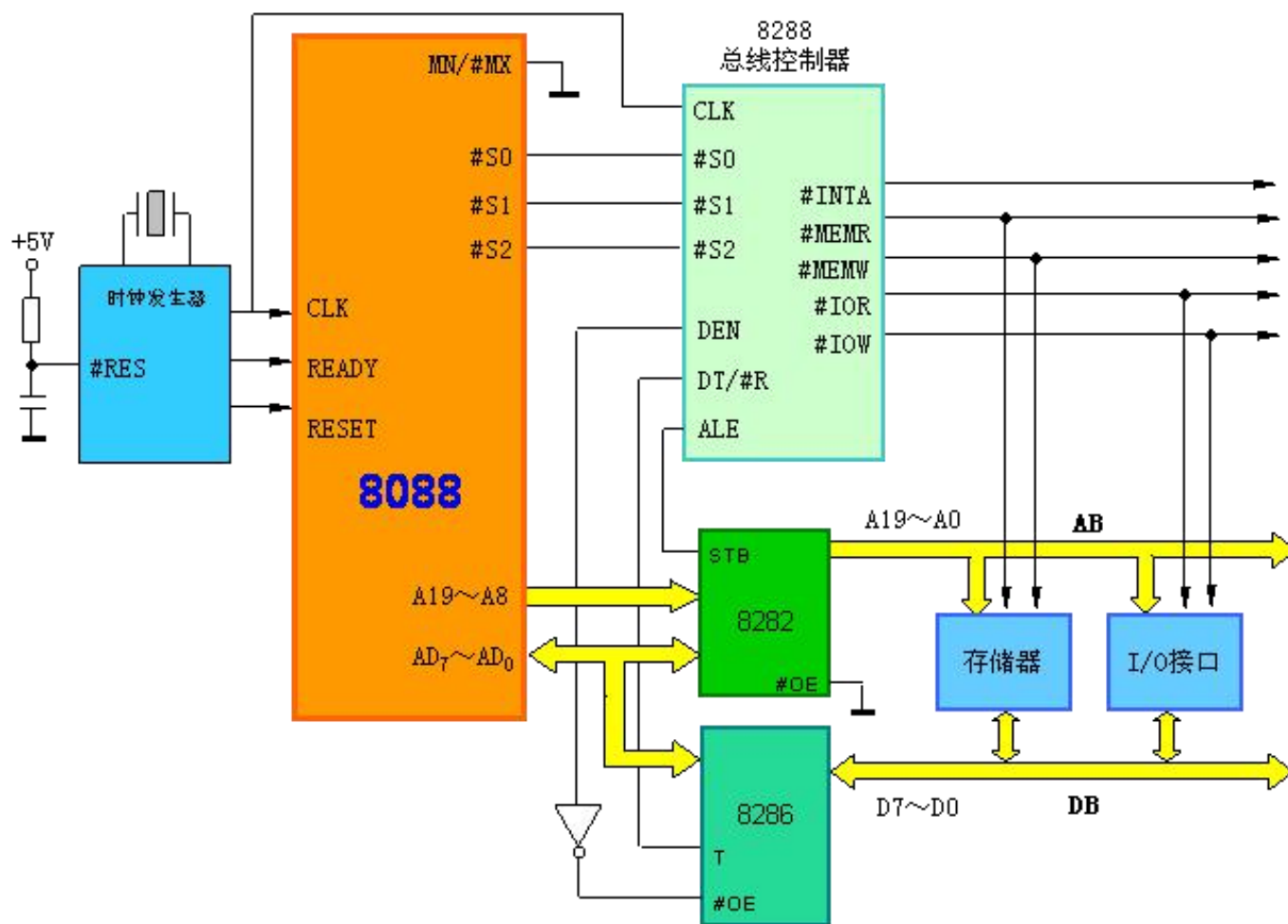
## 第4章：最小组态总线形成 (Intel 产品手册推荐电路)





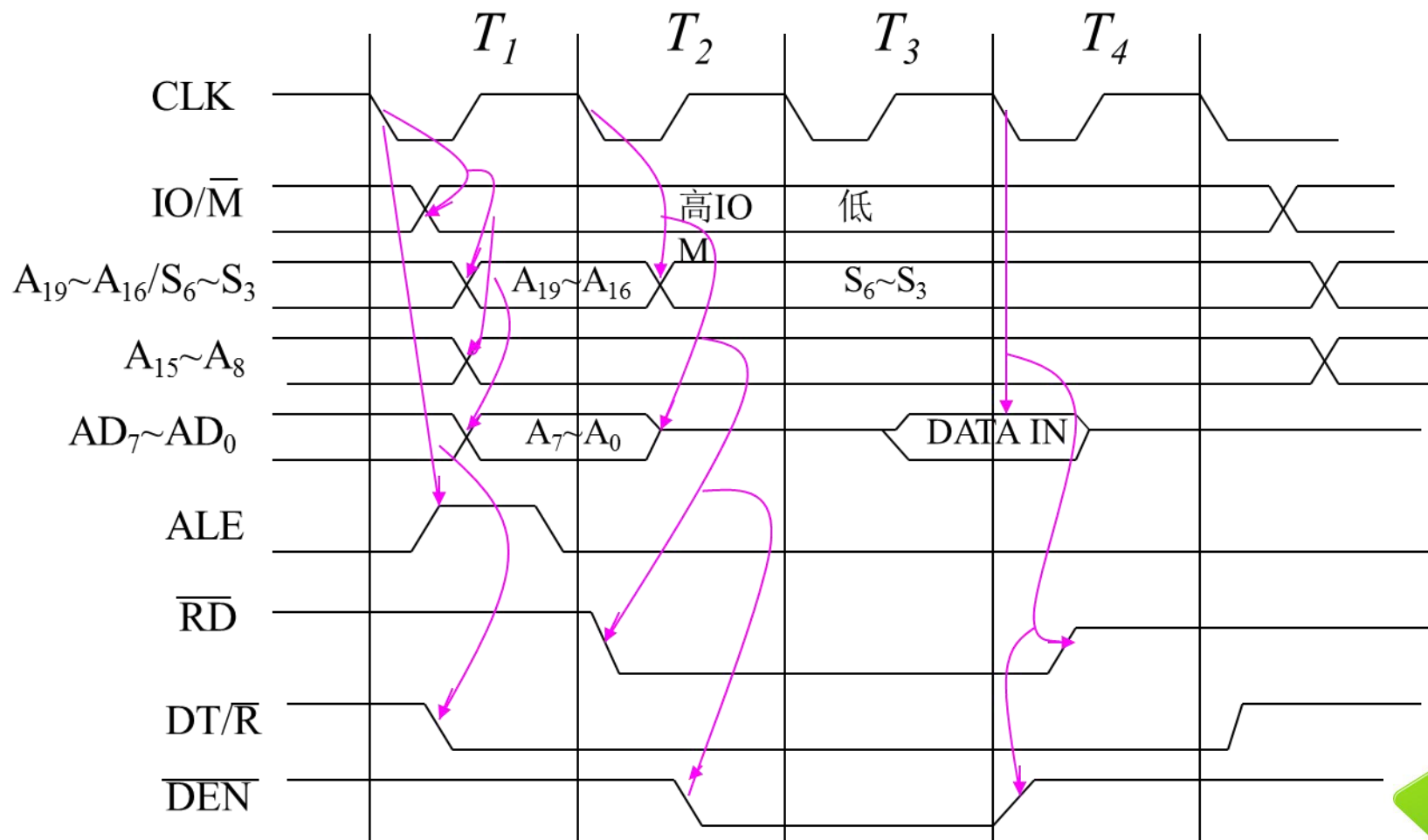


# 8088最大组态总线构成

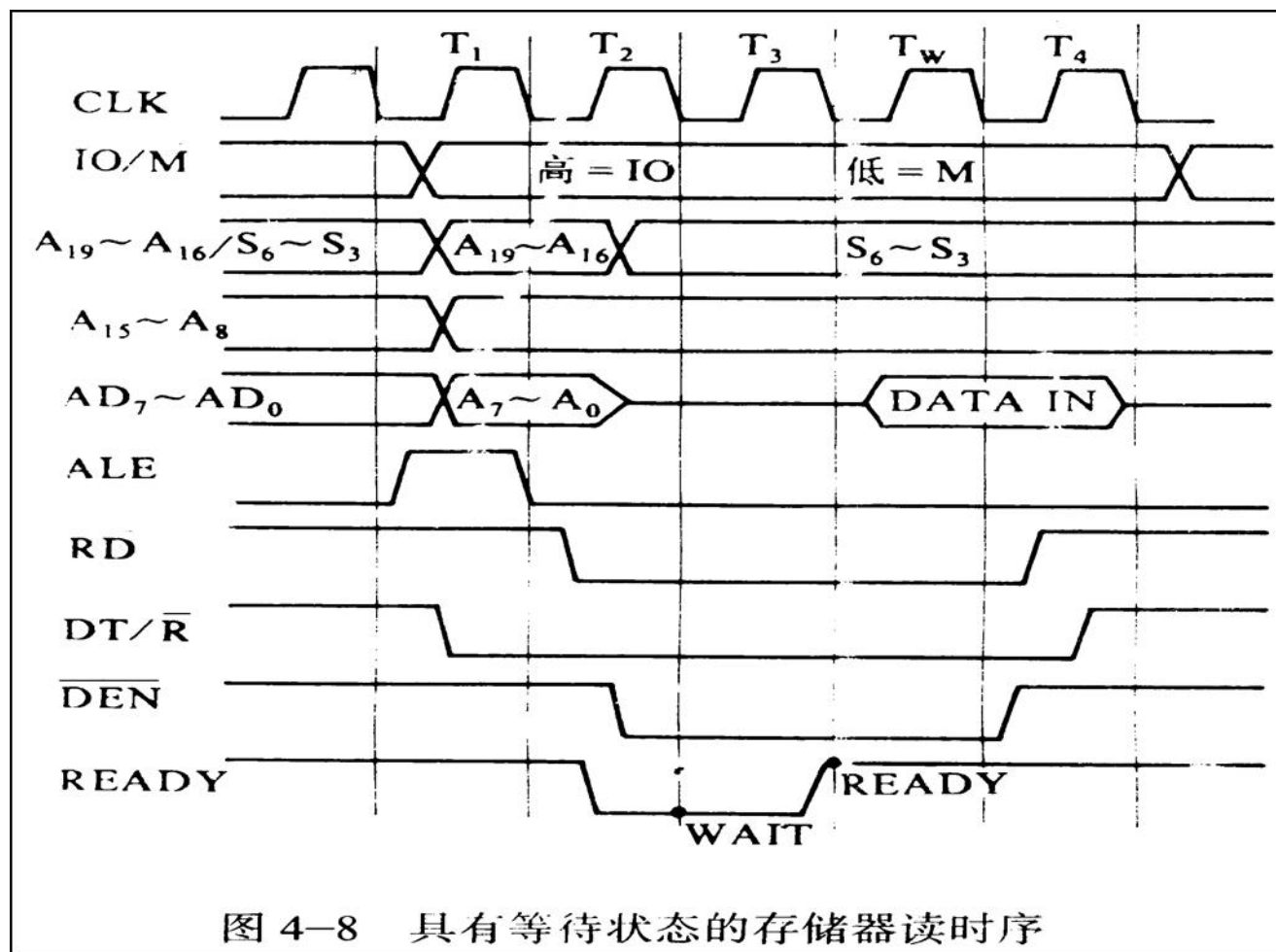


# 8088存储器读周期

0357 PM 136

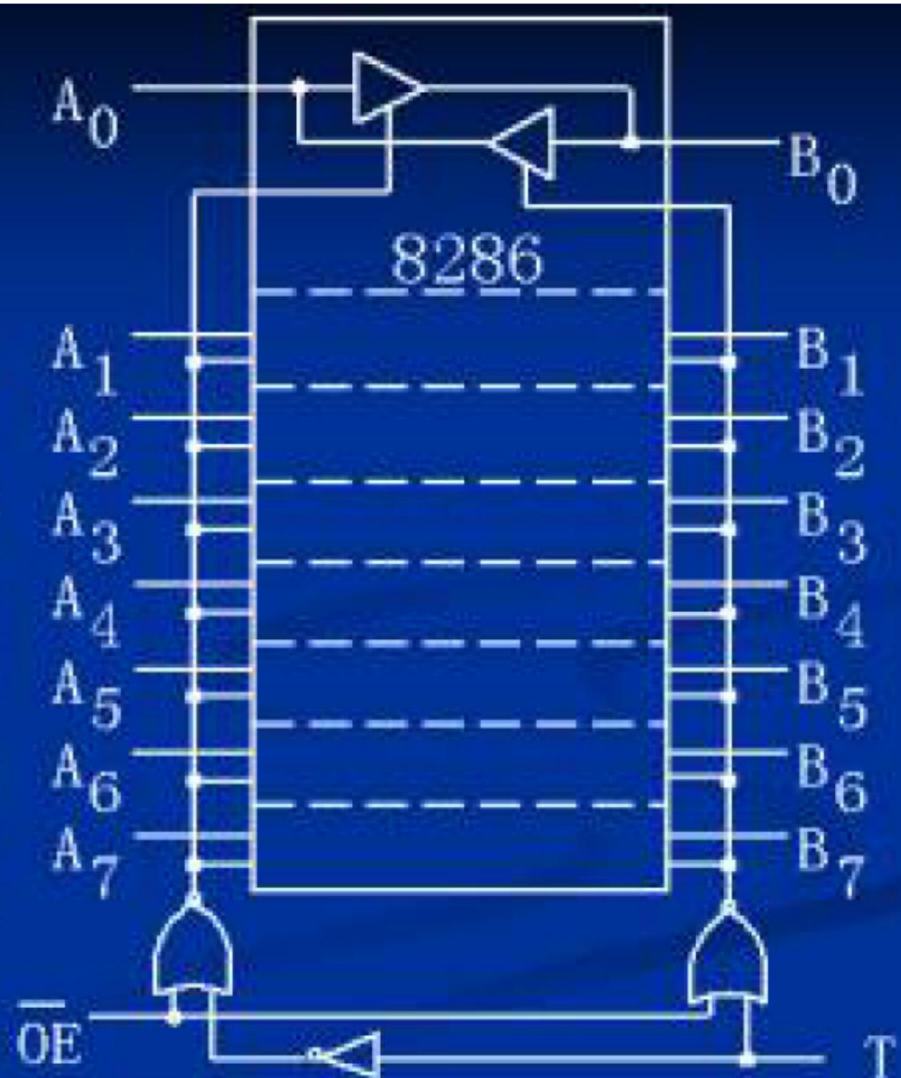


# 8088存储器读周期—等待周期



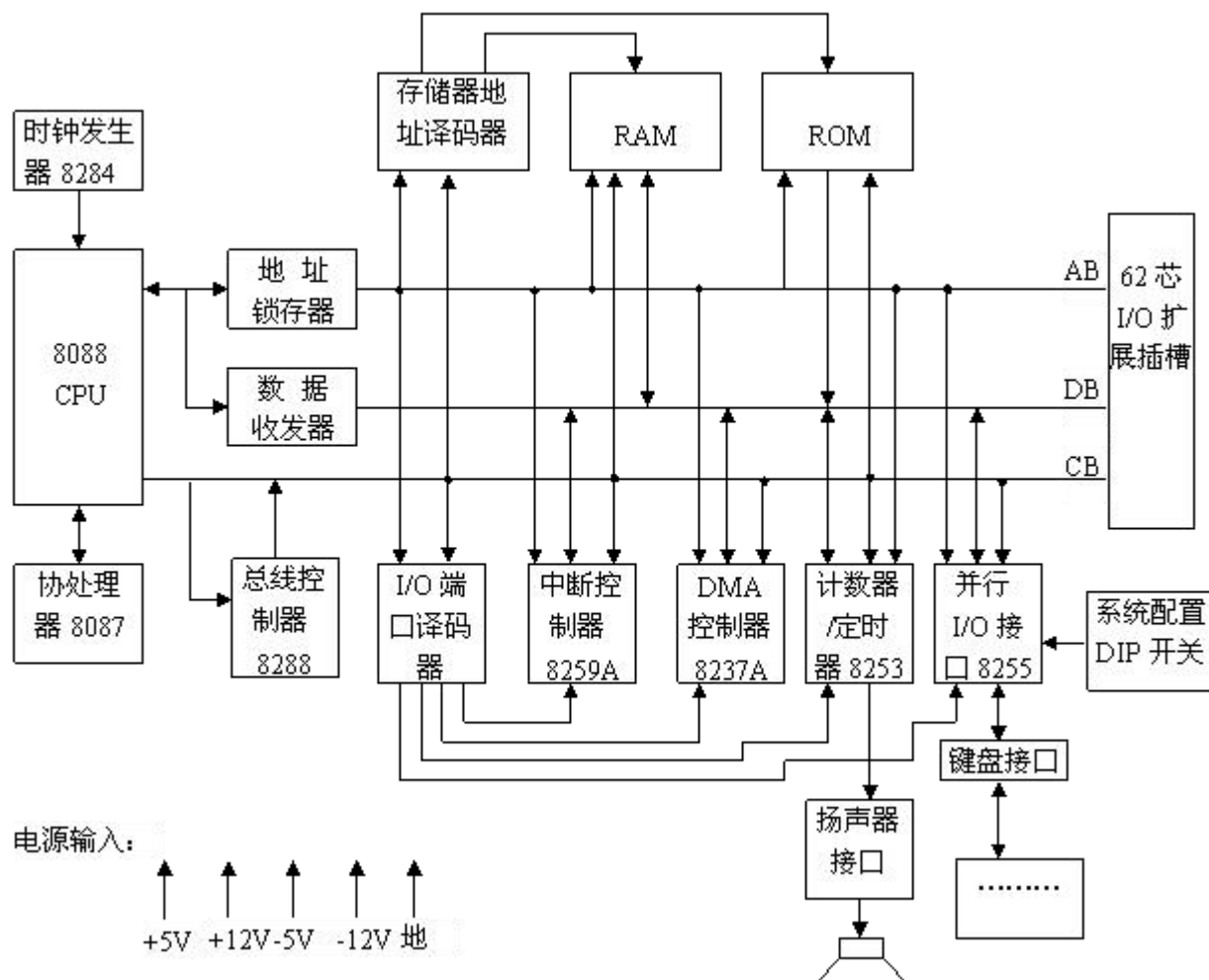
功能表

$\overline{OE}$	T	传送方向
0	1	$A_i \rightarrow B_i$
0	0	$B_i \rightarrow A_i$
1	1	高阻状态
1	0	高阻状态



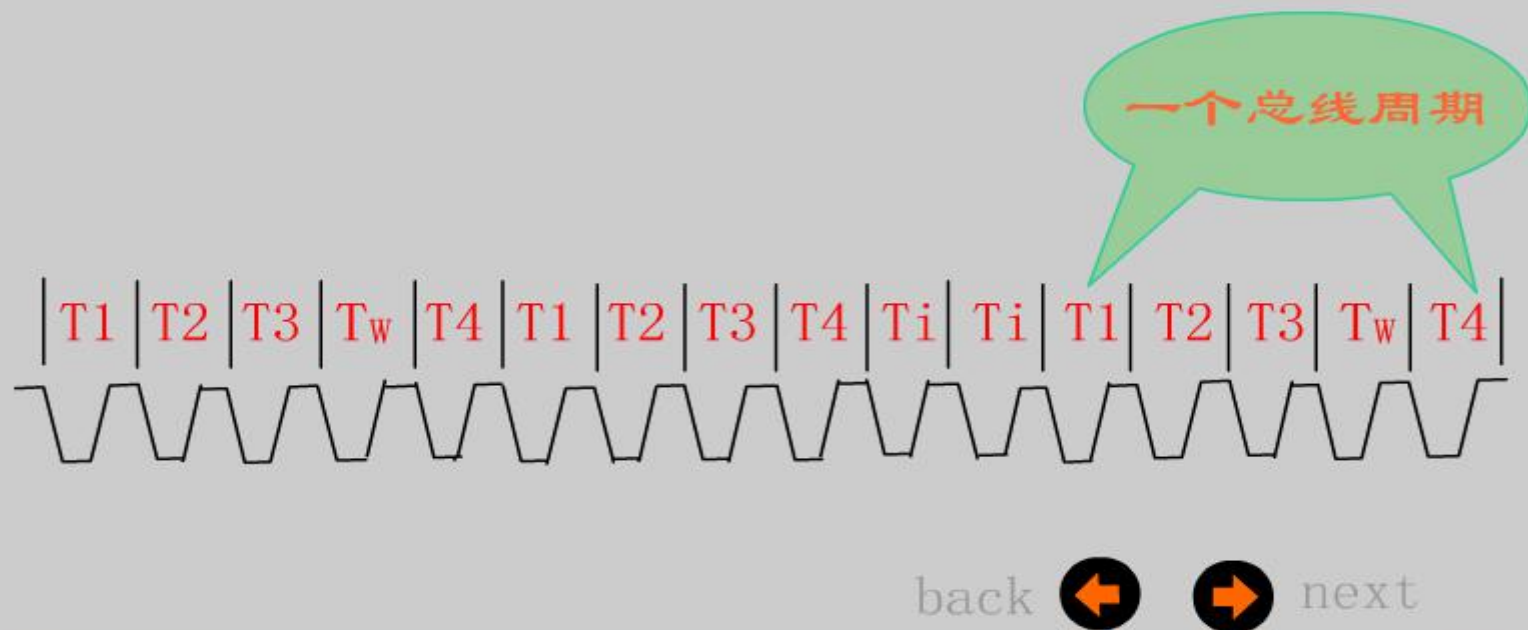
# IBM PC 结构

0357 PM 36





## 周期介绍



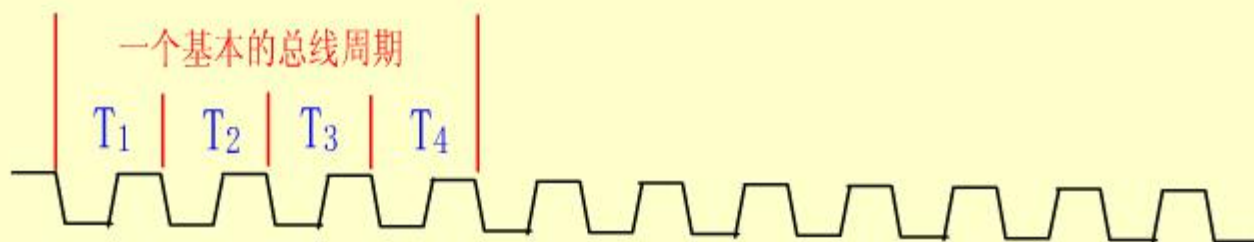
back



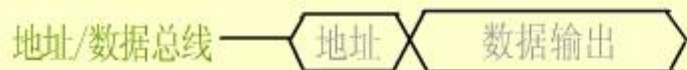
next

CPU与外部交换信息总是通过总线进行的，CPU的每一个这种信息输入输出过程所需要的时间称为总线周期，每当CPU要从存储器或输入输出口存取一个字节或字就需要一个总线周期。8086 CPU的总线周期至少由4个周期组成，分别以T1、T2、T3和T4表示





读周期



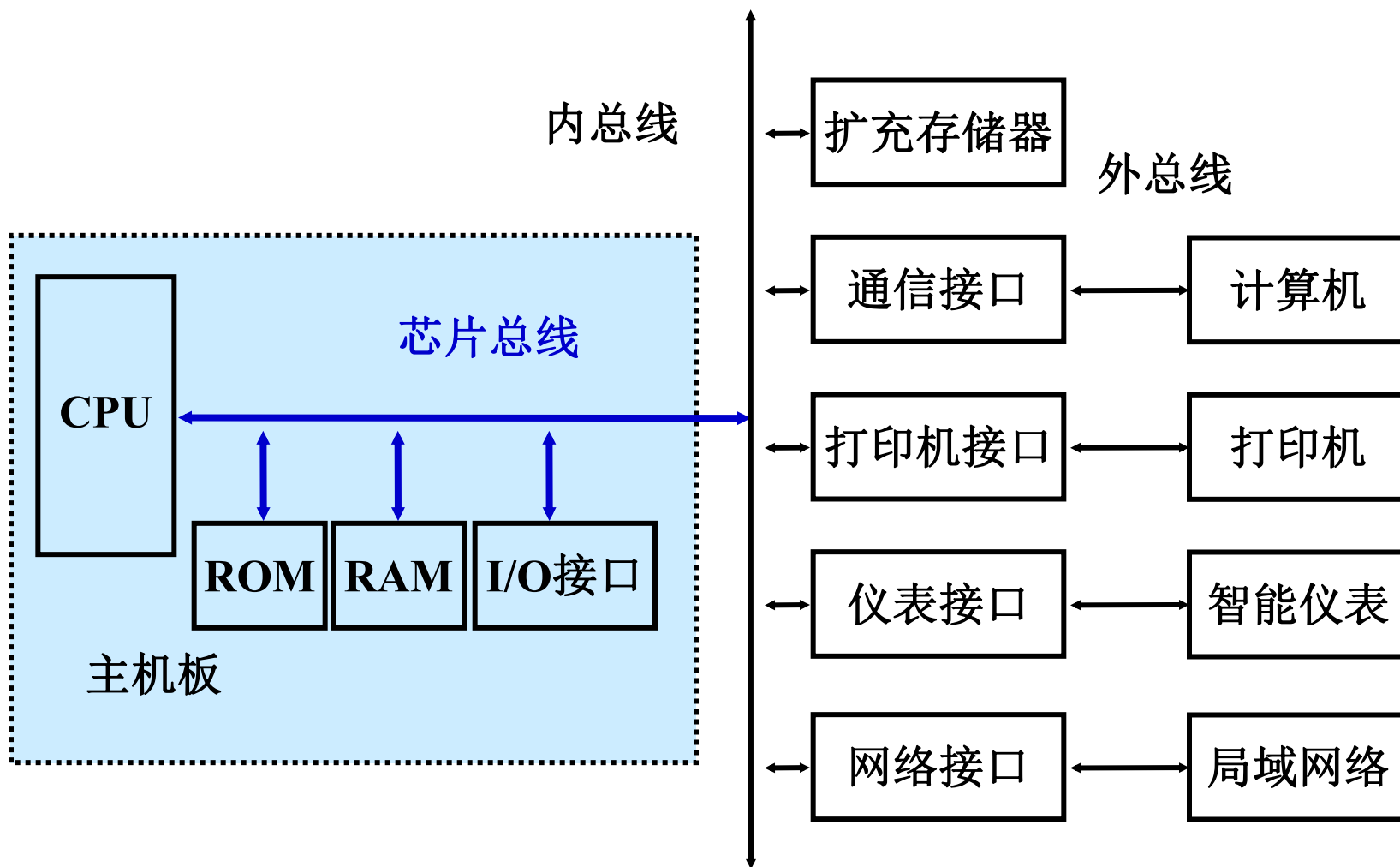
写周期

播放

停止

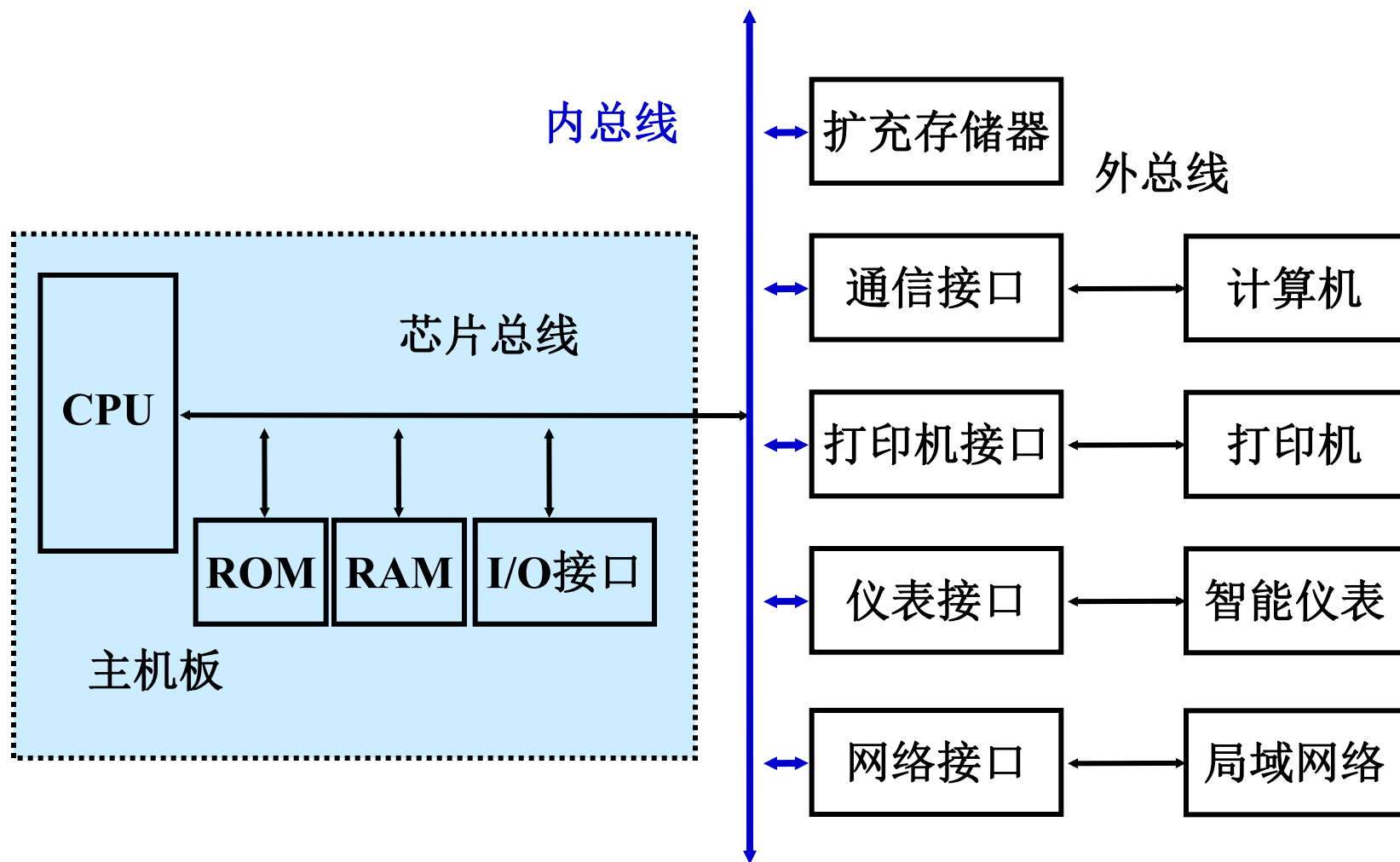


## 第4章：微机总线层次结构—芯片总线

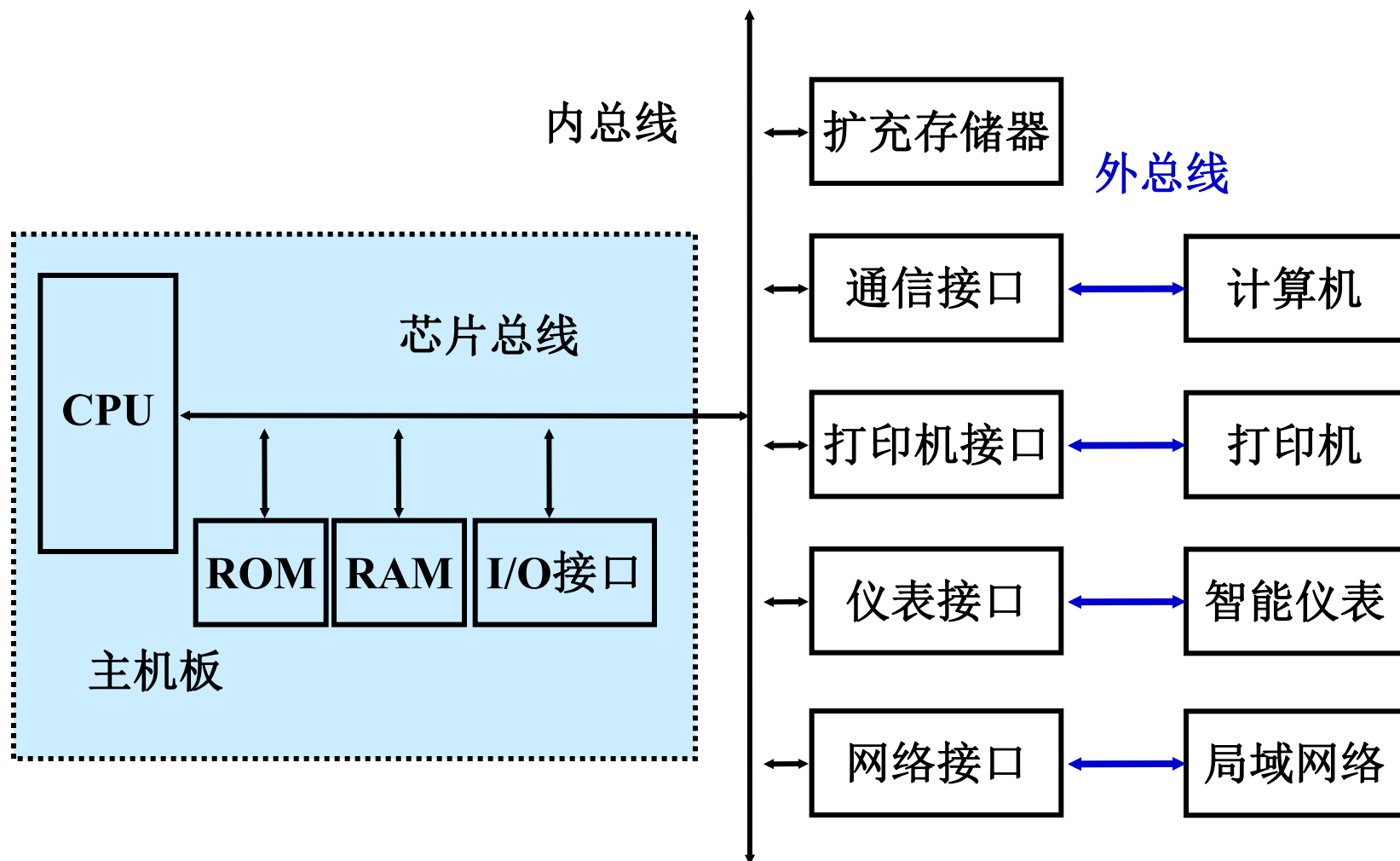




## 第4章：微机总线层次结构—内总线



## 第4章：微机总线层次结构—外总线

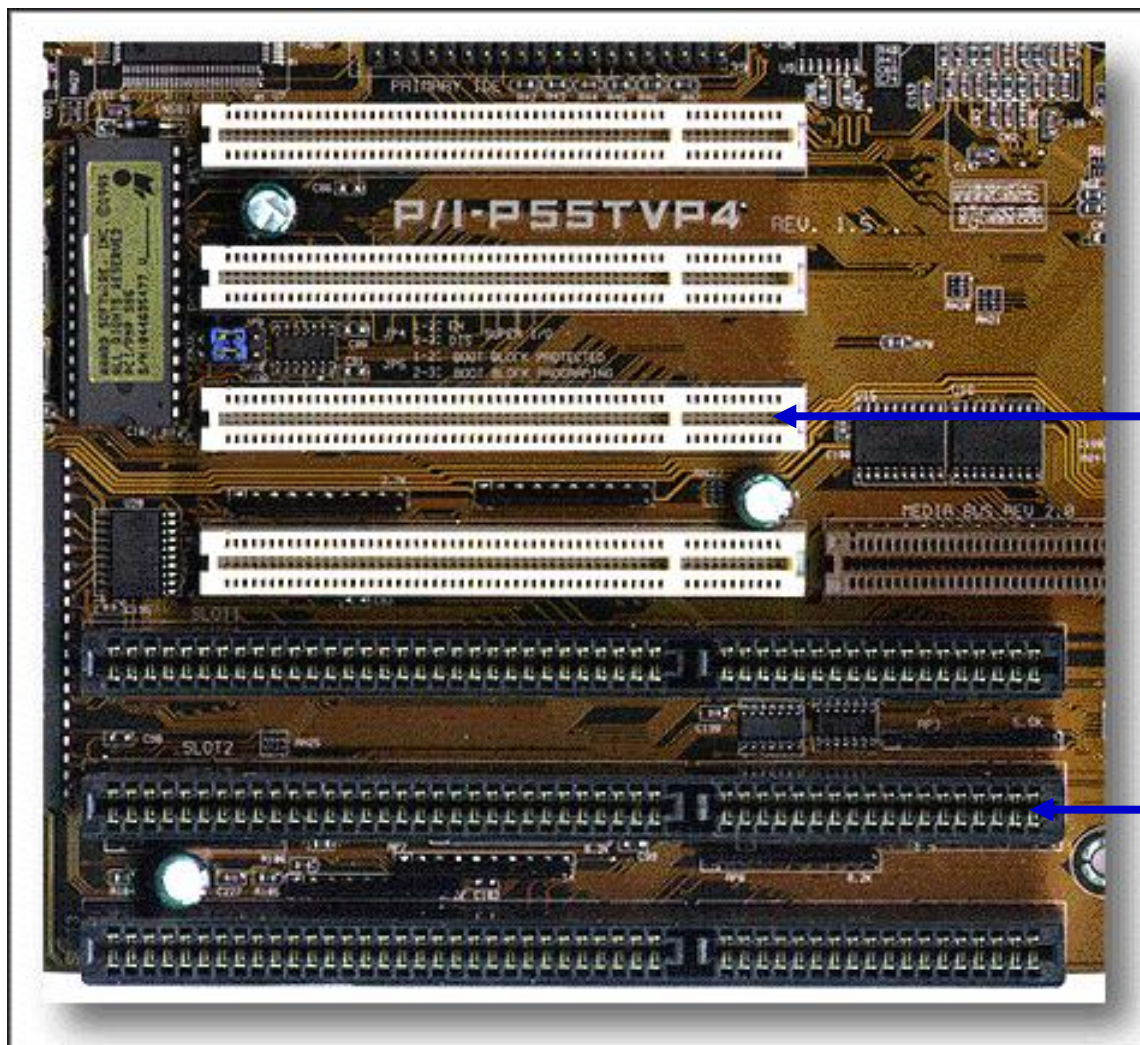


## 第4章：PC机上的总线

**USB总线**



**PCI总线**



**ISA总线**





