



# 第 5 章

## 微处理器外部特性与输入输出





# 5.1 处理器外部特性



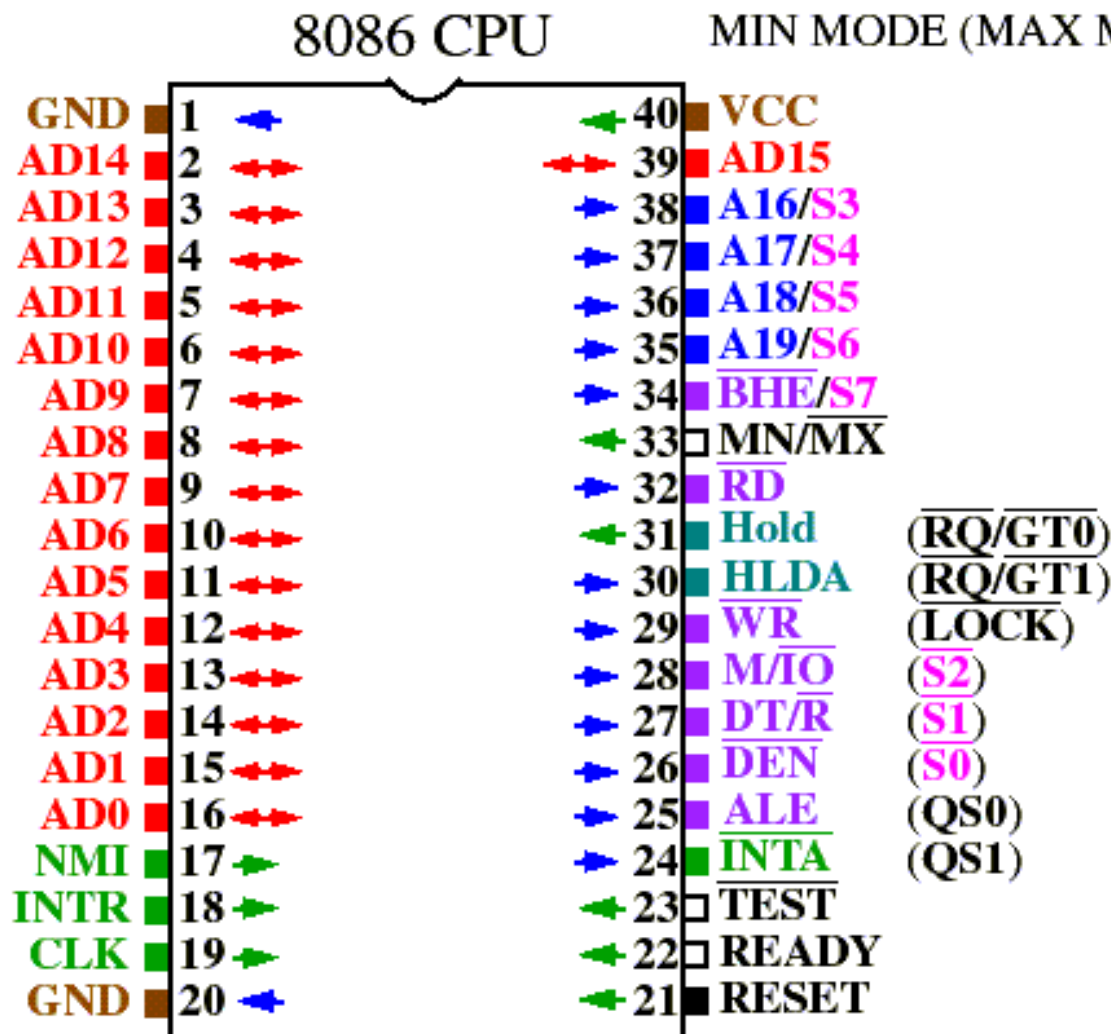
# 1. 管脚定义



# 8086/88管脚描述

8086: 16位微处理器,  
16位外部数据总线

8088: 16位微处理器,  
8位外部数据总线





# 8086/88管脚功能（1）

## ■ AD15-AD0

分时复用地址（ $ALE=1$ ） /数据总线（ $ALE=0$ ）

## ■ A19/S6-A16/S3

20位地址线的高4位或状态线S6-S3

## ■ $\overline{M}/\overline{IO}$

标识地址线用作存储器地址还是IO地址

## ■ $\overline{RD}$

读信号，低电平有效

## ■ $\overline{WR}$

写信号，低电平有效





# 分时复用的概念

- 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





# 8086/88管脚功能 (2)

## ■ ALE

地址锁存允许信号。高电平表示地址数据总线上传送的是地址

## ■ $\overline{\text{DT/R}}$

数据发送/接收控制信号

## ■ $\overline{\text{DEN}}$

数据允许信号

## ■ INTR

当为1且IF=1时，CPU将进行中断服务。在当前指令执行完后 $\overline{\text{INTA}}$ 变为低电平以响应中断。





# 8086/88管脚功能 (3)

## ■ $\overline{\text{INTA}}$

中断响应信号

## ■ NMI

不可屏蔽中断信号。上升沿有效

## ■ CLK

主时钟信号。 8088的工作时钟为5MHz

## ■ VCC

电源, +5V

## ■ GND

地线, 0V







# 8086/88管脚功能（4）

## ■ $\overline{\text{MN}}/\overline{\text{MX}}$

工作模式选择信号

## ■ $\overline{\text{BHE}}$

总线高字节有效。在读或写操作期间，使能数据总线的高8位

## ■ $\text{READY}$

准备好信号。用于在CPU的总线周期中插入等待状态。该信号由存储器或I/O设备产生。

## ■ $\text{RESET}$

复位信号。高电平持续4个时钟周期以上才有效





# 8086/88管脚功能 (5)

## ■ $\overline{\text{TEST}}$

测试信号。当CPU执行WAIT指令时，每5个CLK检测该管脚，若为1，则等待；否则执行下一条指令。当使用8087协处理器时，通过该管脚和WAIT指令，可使CPU与8087协调执行过程

## ■ HOLD (大模式: $\overline{\text{RQ}}/\overline{\text{GT0}}$ )

总线请求信号

## ■ HLDA (大模式: $\overline{\text{RQ}}/\overline{\text{GT1}}$ )

总线请求响应信号





# 8086/88管脚功能（6）

## ■ $\overline{\text{LOCK}}$

总线锁定信号。当在指令前加上 $\overline{\text{LOCK}}$ 前缀时，则在执行这条指令期间 $\text{LOCK}$ 保持有效，即在此指令执行期间，CPU封锁其它主控者使用总线

## ■ QS1和QS0

队列状态信号。允许从外部观察内部指令队列的状态





# 8086/88管脚功能（7）

■ S7, S6, S5, S4, S3,  $\overline{S2}$ ,  $\overline{S1}$ ,  $\overline{S0}$

S7为1, S6为0

S5等于IF

S4和S3指示在当前总线周期哪一个段正在被访问

S4 S3	功能
0 0	附加段
0 1	堆栈段
1 0	代码段或无（即I/O或中断过程）
1 1	数据段





# 8086/88管脚功能（8）

$\overline{S2}$ 、 $\overline{S1}$ 和 $\overline{S0}$ 指示当前总线周期的功能

$\overline{S2}\overline{S1}\overline{S0}$	功能
0 0 0	中断响应
0 0 1	读I/O端口
0 1 0	写I/O端口
0 1 1	Halt
1 0 0	访问代码
1 0 1	读存储器
1 1 0	写存储器
1 1 1	Passive（无总线周期）



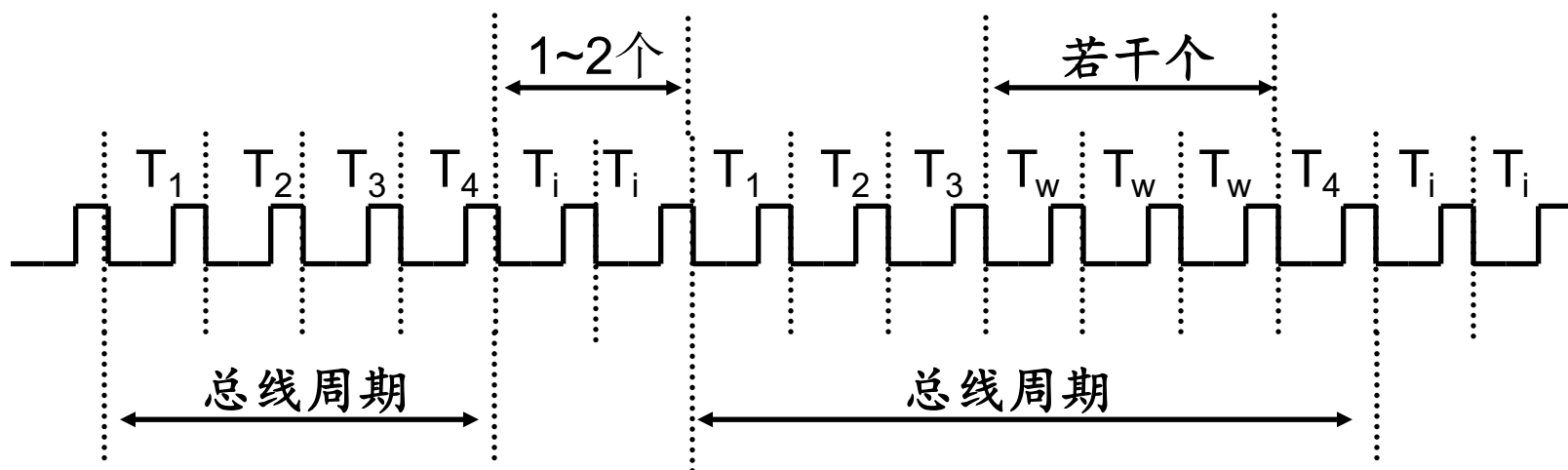


## 2. 总线周期及时序



# 基本概念

- 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。
- **总线周期** 是指CPU完成一次访问存储器 (或I/O端口) 的操作所需要的时间。
- 对于8086/88CPU, 基本总线周期包括4个时钟周期 ( $T_1 \sim T_4$ )。这个时钟周期也称为T状态。
- $T_i$  是在两个总线周期之间插入的空闲时钟周期





# 8086/88的总线周期

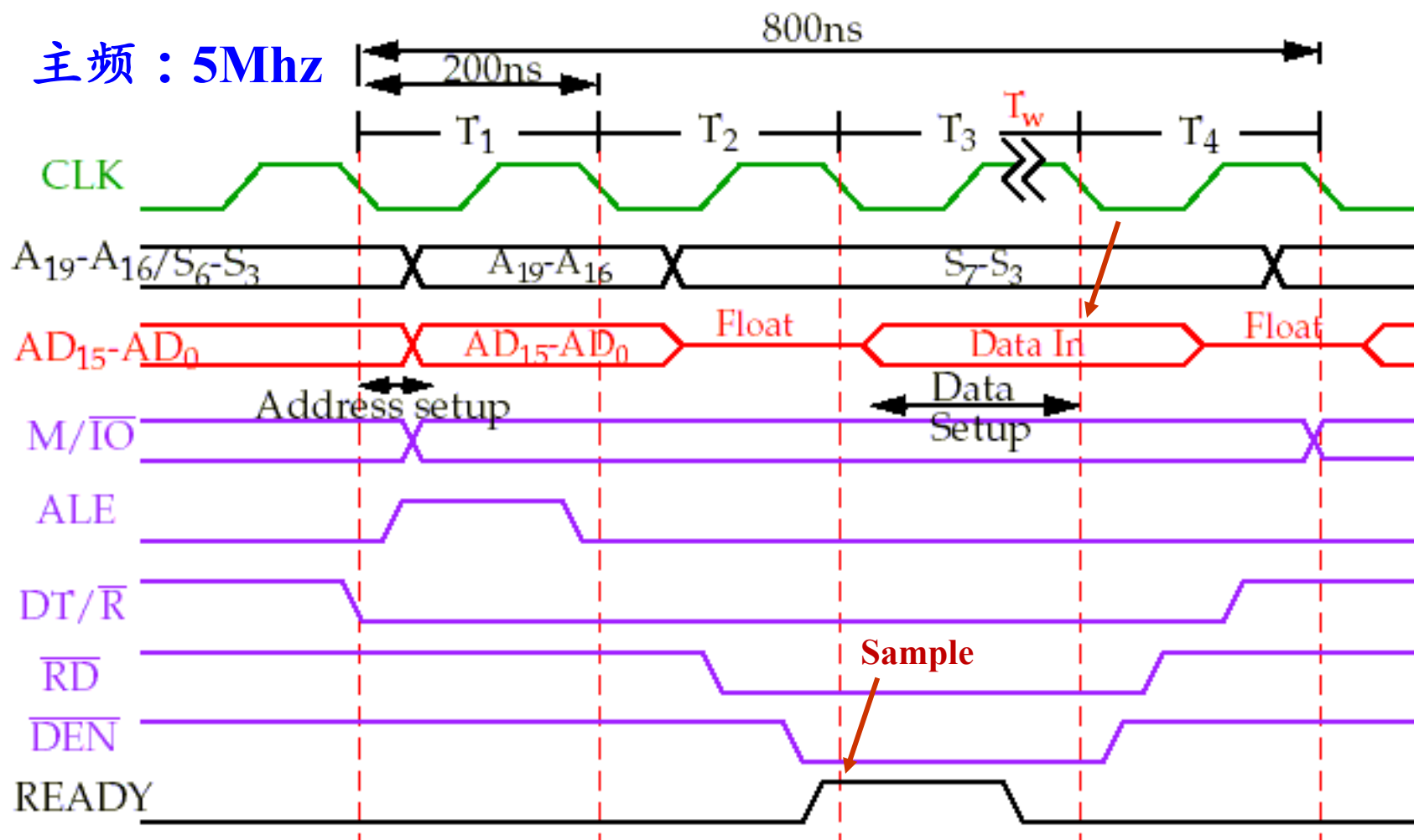
- 存储器写总线周期
- 存储器读总线周期
- I/O写总线周期
- I/O读总线周期
- 中断响应周期
- 总线请求及响应周期







# 8086读周期时序 (1)



Bus Timing for a Read Operation





# 8086读周期时序 (2)

## T1状态

- 8086把地址放在地址/数据总线上
- ALE锁存地址， $\overline{M}/\overline{IO}$ 和 $\overline{DT}/\overline{R}$ 指明存储器访问或I/O访问，确定数据传送方向。

## T2状态

- 发出 $\overline{RD}$ 、 $\overline{WR}$ 和 $\overline{DEN}$ 信号

## T3状态

- 在T3的前沿采样READY信号，若为低电平，则在T3之后插入等待状态 $T_w$ （等价于T3）。





# 8086读周期时序 (3)

## T4状态

- T4的前沿采样数据总线。然后，所有总线信号变为无效，为下一总线周期做好准备。





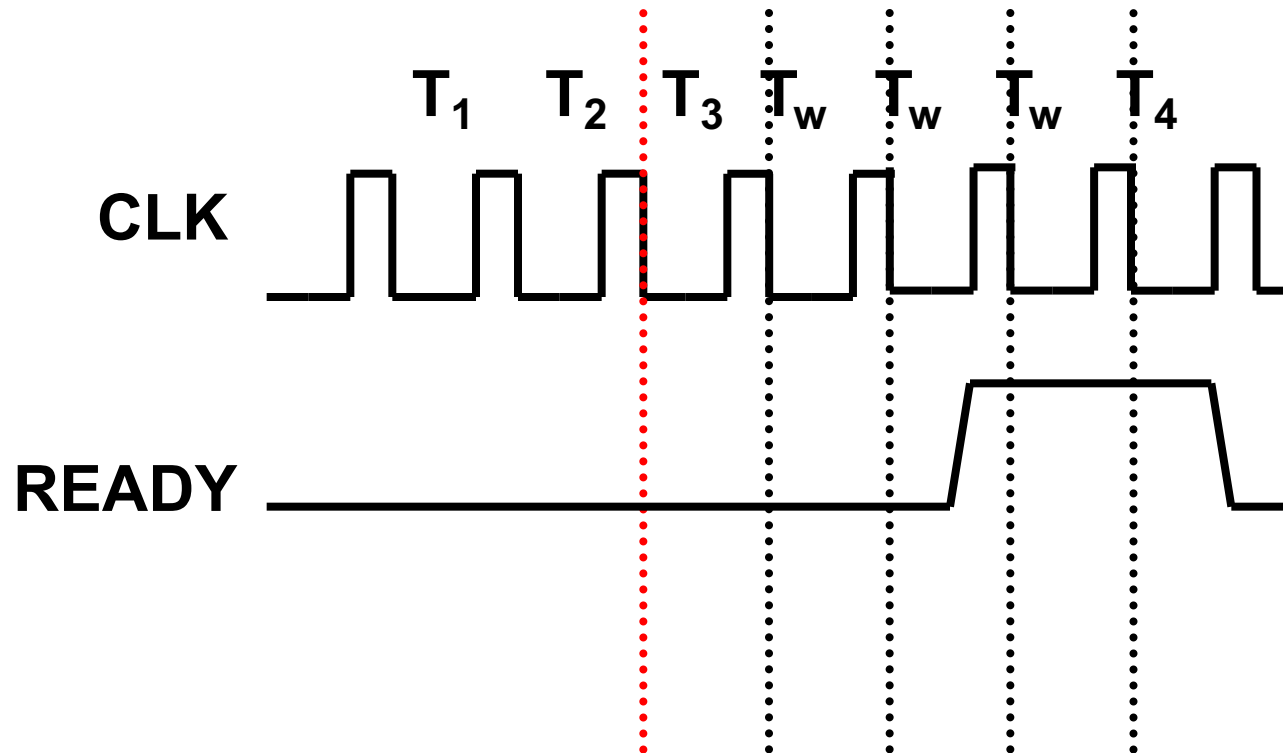
# READY信号线的使用 (1)

- 8086在访问慢速存储器或I/O设备时需要在T3和T4之间插入一个或多个等待状态 $T_w$ 。
- 存储器或I/O设备是通过READY信号来控制是否要插入 $T_w$ 状态
  1. 在T3的前沿检测READY引脚是否有效
  2. 如果READY无效，在T3和T4之间插入一个等效于T3的 $T_w$ ，转步骤1
  3. 如果READY有效，执行完该T状态，进入T4状态





# READY信号线的使用 (2)



前沿检测





### 3. 工作模式



# 8086CPU工作模式

两种工作模式：最大模式和最小模式。

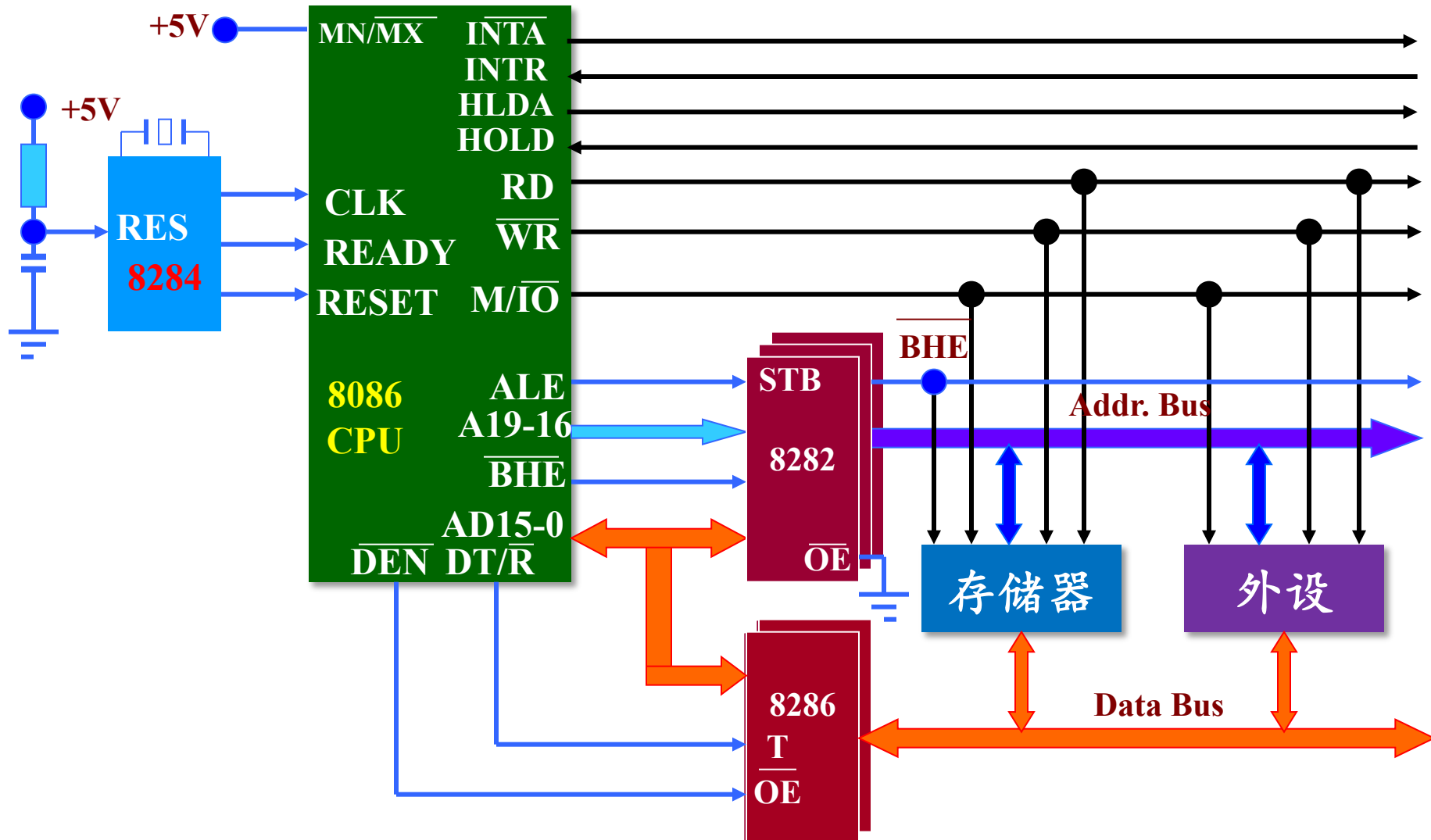
- **最小模式：**就是在系统中只有8086/8088一个处理器。在这种系统中所有的总线控制信号都直接由8086/8088产生，在这样系统中的总线控制逻辑电路被减到最少。
- **最大模式：**可包含两个或多个处理器，一个为主处理器，其他的是协处理器（协助主处理器工作）。和8086/8088CPU配合使用的协处理器主要有数值运算协处理器8087和输入/输出协处理器8089。

加电时，设置微处理器管脚 $\overline{MN}/\overline{MX}$ 电平的高低，可以控制微处理器工作在最小模式或最大模式



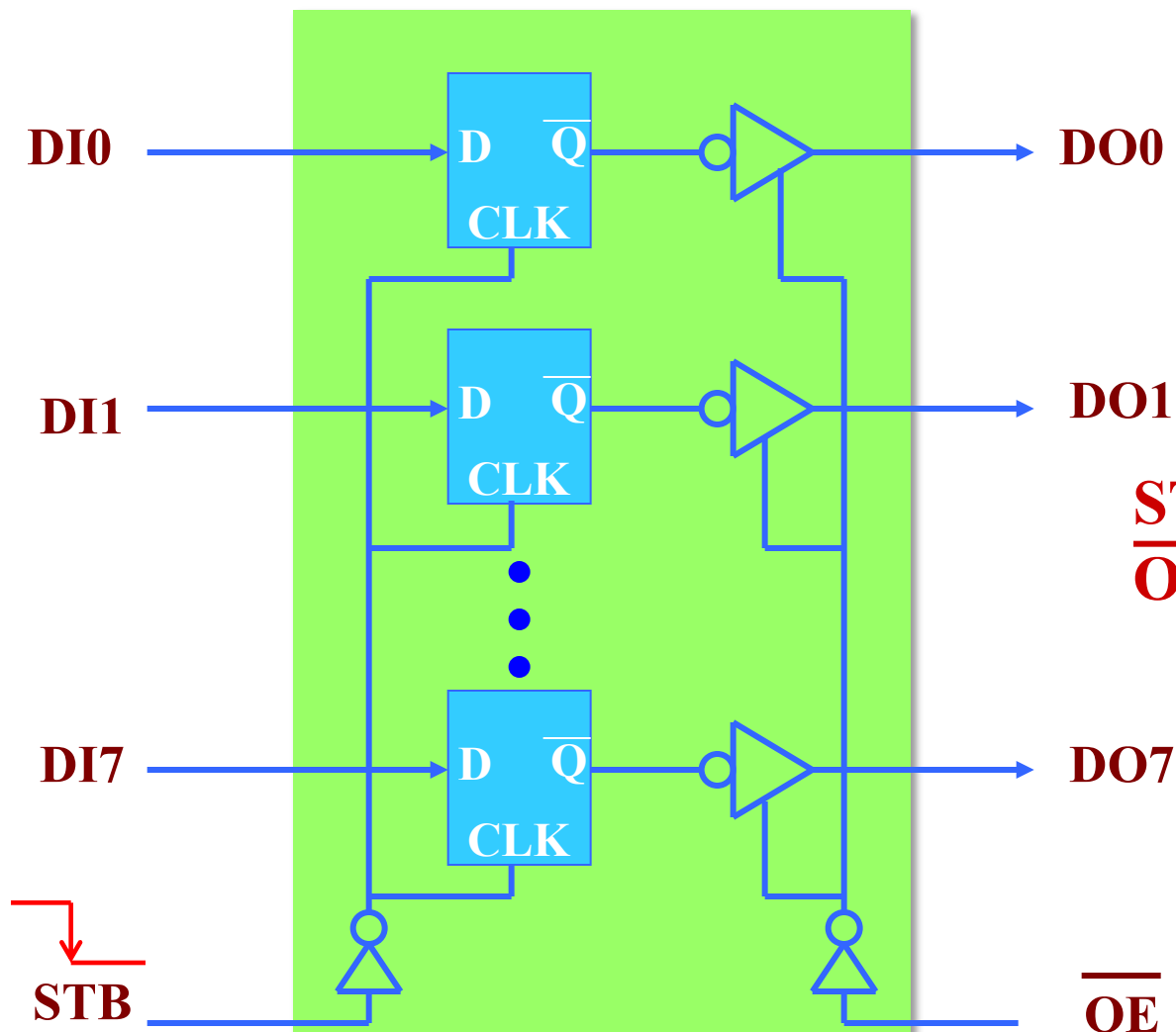


# 最小模式





# 带三态缓冲的8位数据锁存器8282

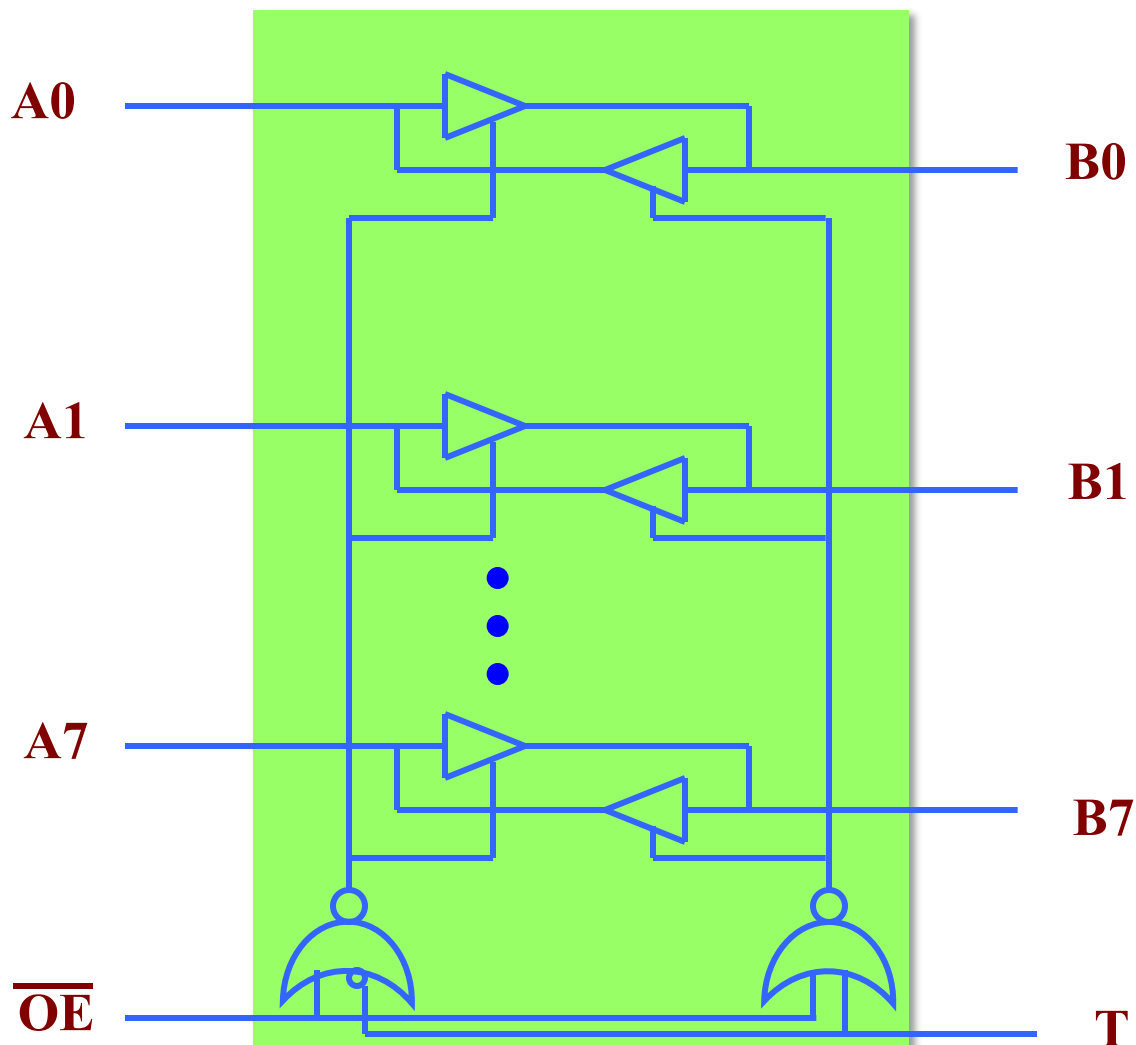


**STB:** 选通脉冲

**$\overline{OE}$ :** 为0时输出有效  
为1时输出为三态



# 带三态的8位双向数据缓冲器8286



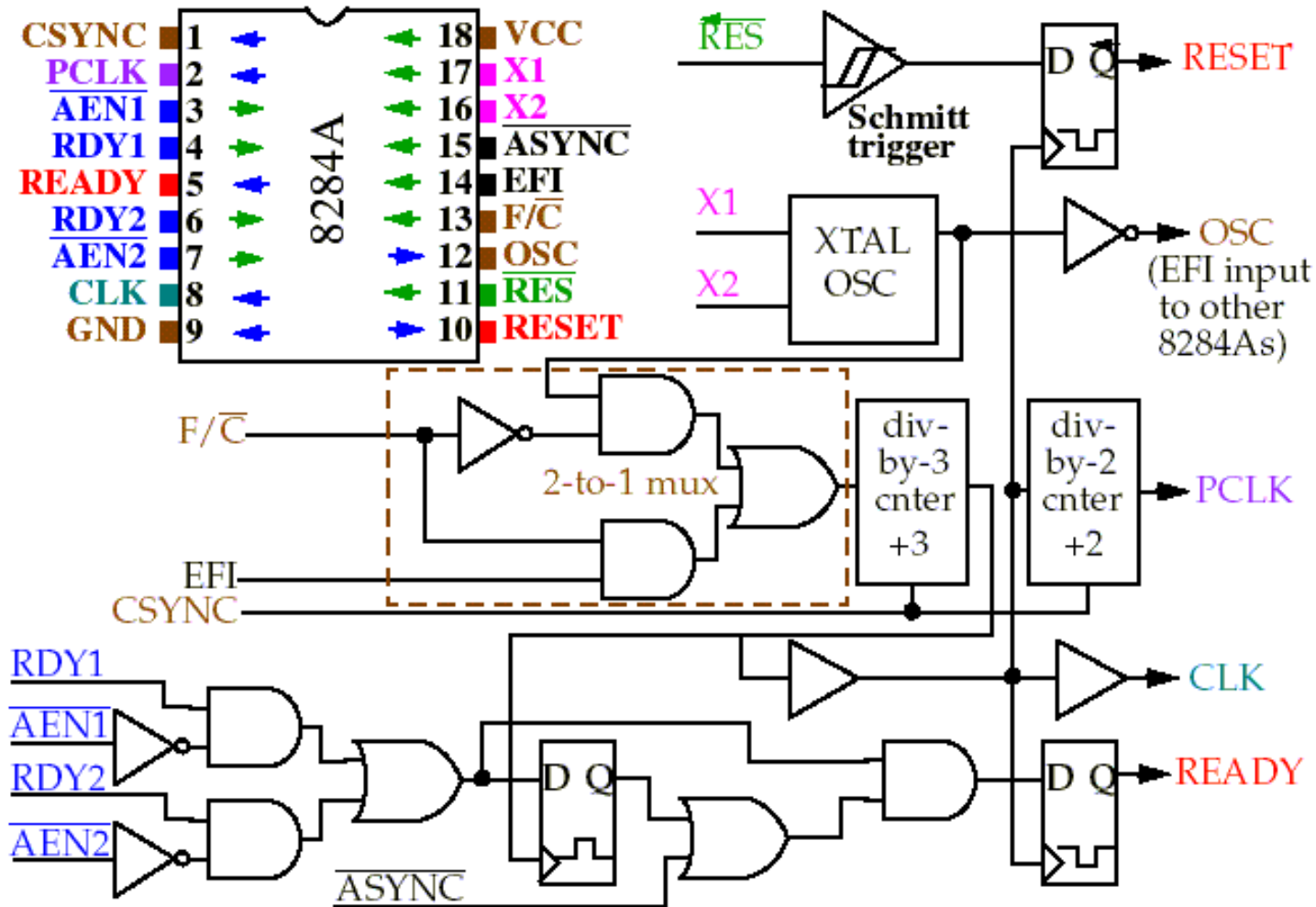
$\overline{OE}$	T	方向
0	1	A→B
0	0	B→A
1	1	三态





# 8284时钟生成器

8284A Clock Generator





# 最大模式

在最大模式下，下列管脚将被重新定义：

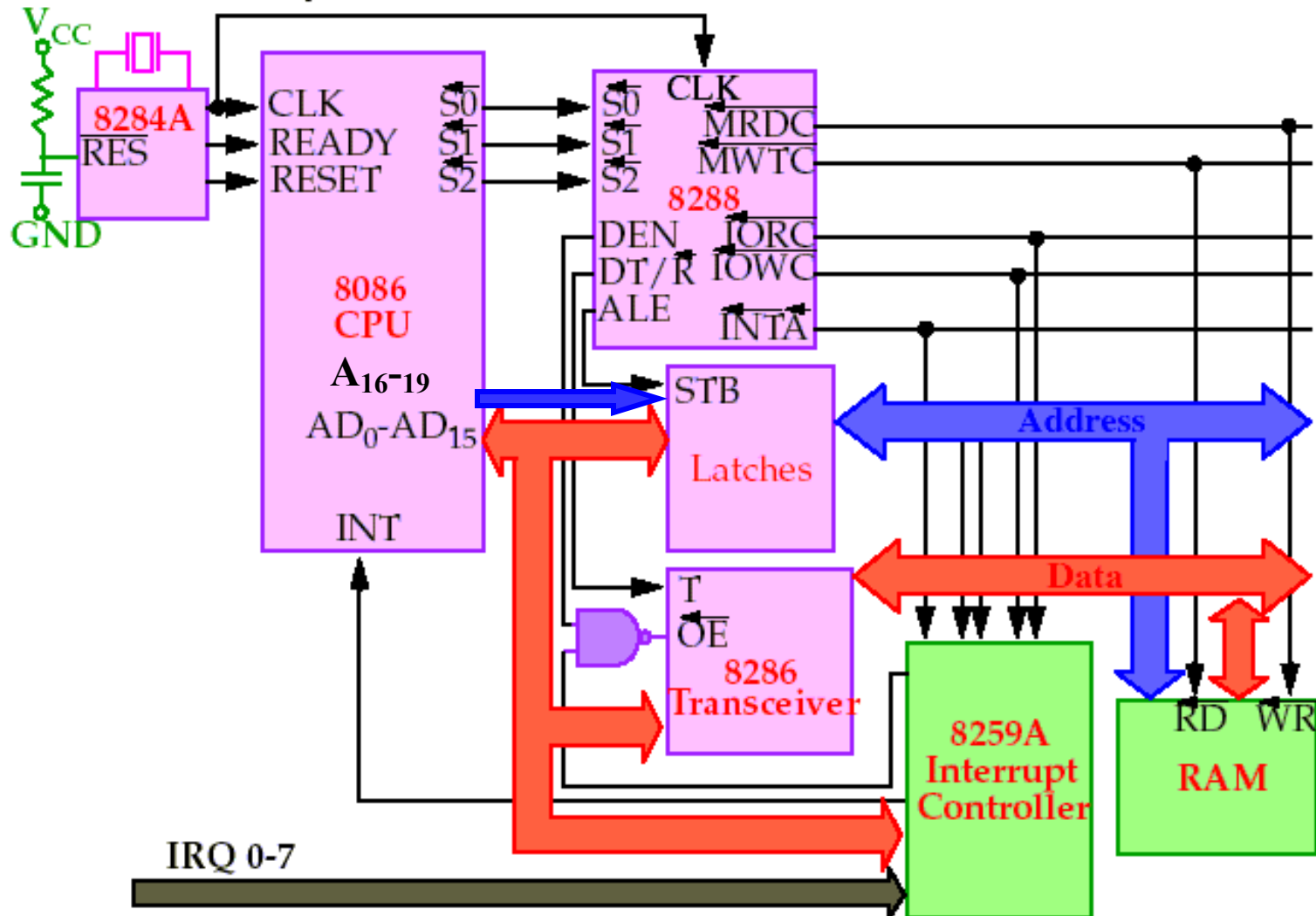
- **ALE**
- **$\overline{\text{WR}}$**
- **$\text{M}/\overline{\text{IO}}$**
- **$\text{DT}/\overline{\text{R}}$**
- **$\overline{\text{DEN}}$**
- **INTA**
- **HOLD**
- **HLDA**

这需要增加8288总线控制器来产生相应的控制信号。

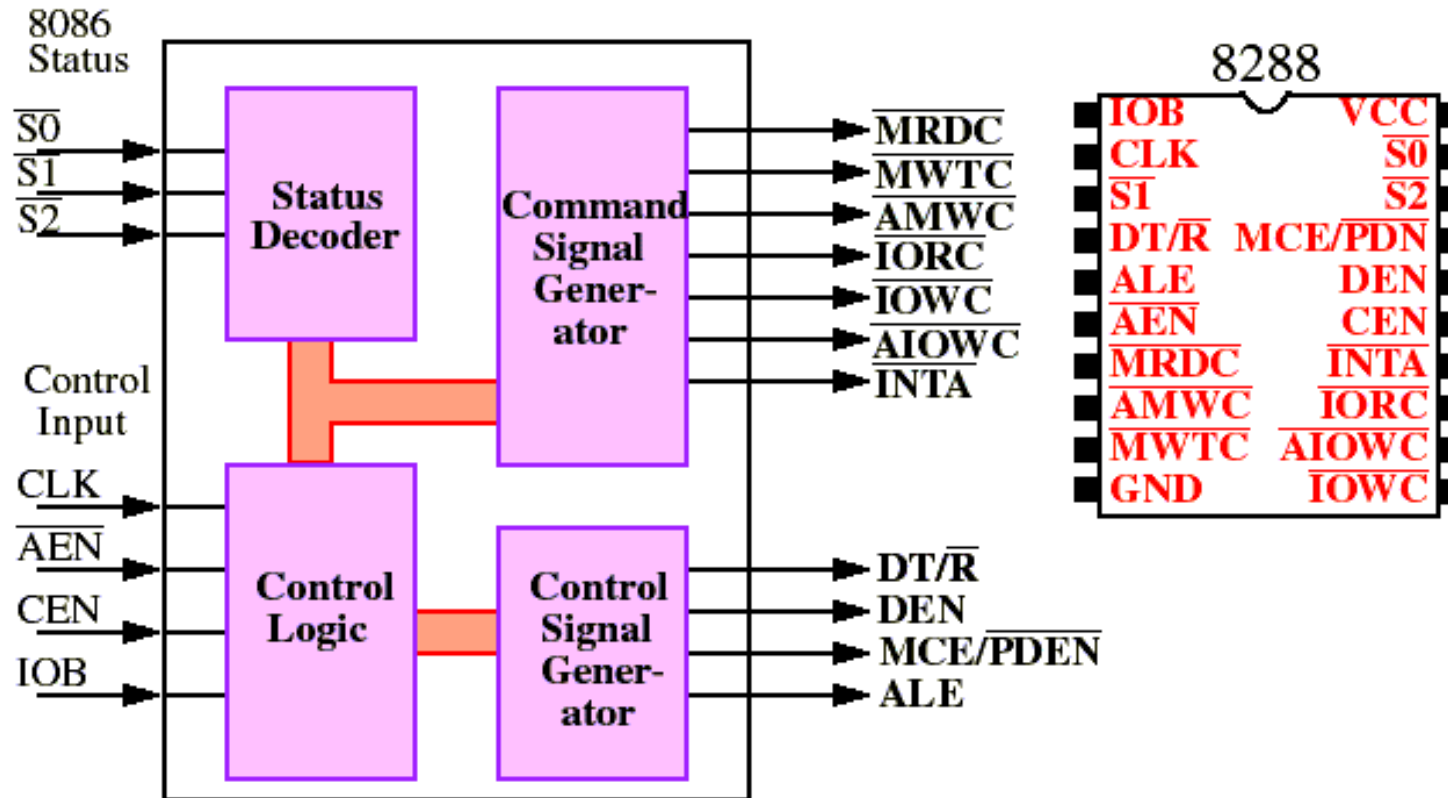




# 最大模式



# 8288总线控制器



Separate signals are used for I/O ( $\overline{IORC}$  and  $\overline{IOWC}$ ) and memory ( $\overline{MRDC}$  and  $\overline{MWTC}$ ).

Also provided are advanced memory ( $\overline{AIOWC}$ ) and I/O ( $\overline{AIOWC}$ ) write strobes plus  $\overline{INTA}$ .



## 4. 存储器接口



# 8086存储器特点

- 数据总线16位，地址总线20位，可寻址空间为1M字节。
- 利用 $\overline{\text{BHE}}$ 信号线，可按字节或字寻址

$\overline{\text{BHE}}$	A0	总线使用情况
0	0	16 位
0	1	高 位
1	0	低 位
1	1	无效

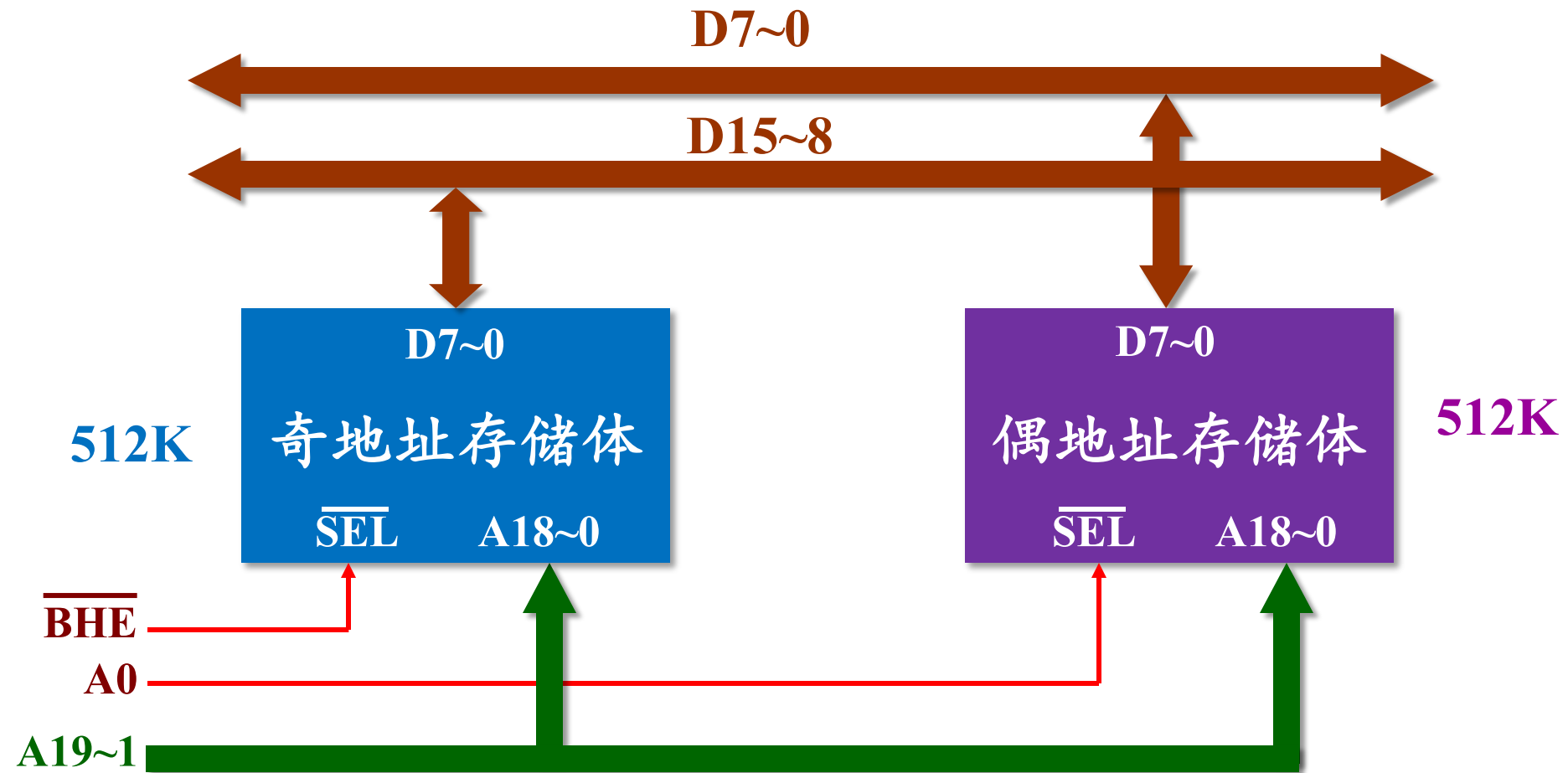
当一个字存放在奇地址开始的边界上，需要两个总线周期才能完成读或写的操作





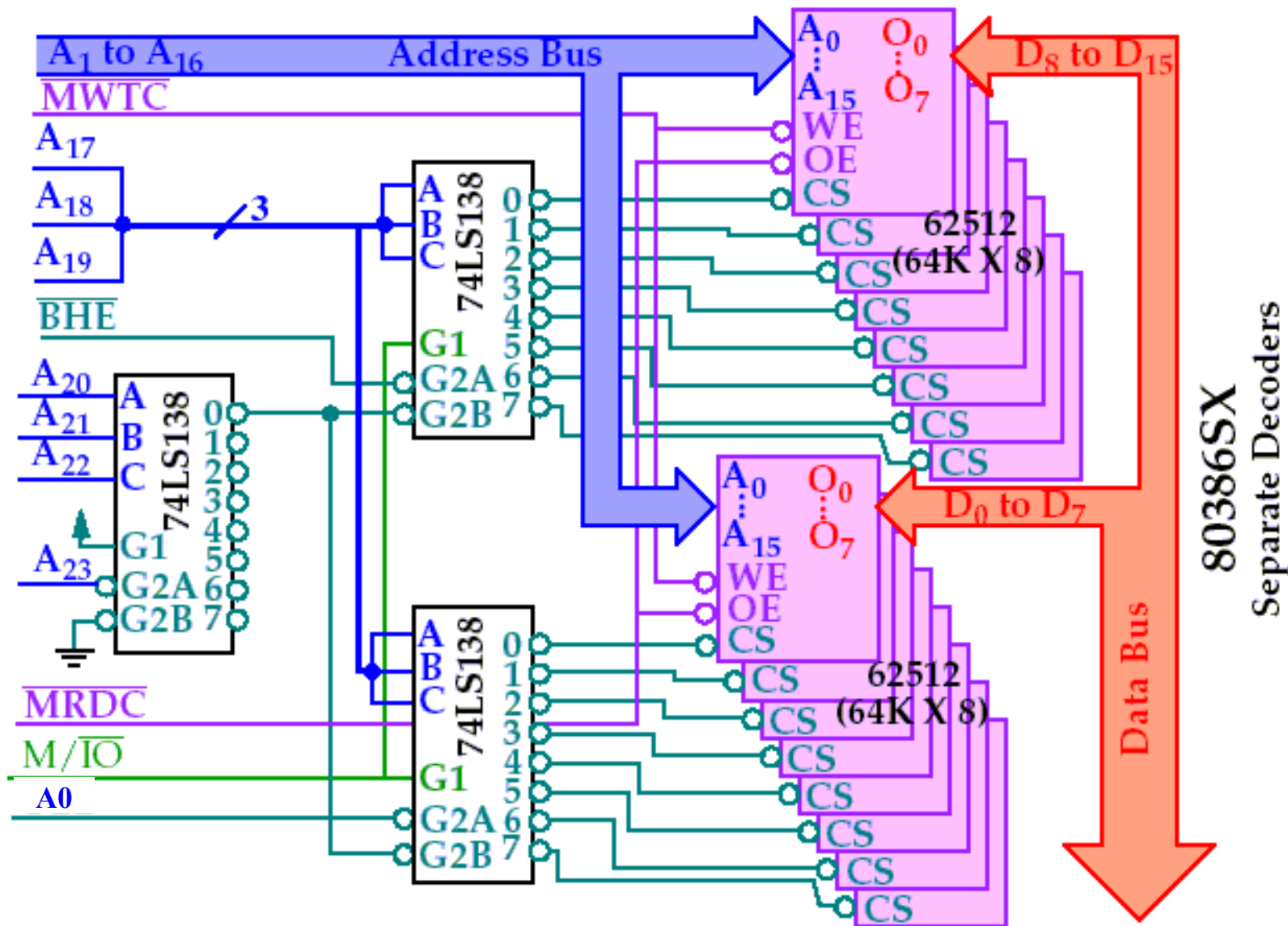


# 16位存储器接口框图





# 例：16位存储器接口





## 5.2 I/O接口概述



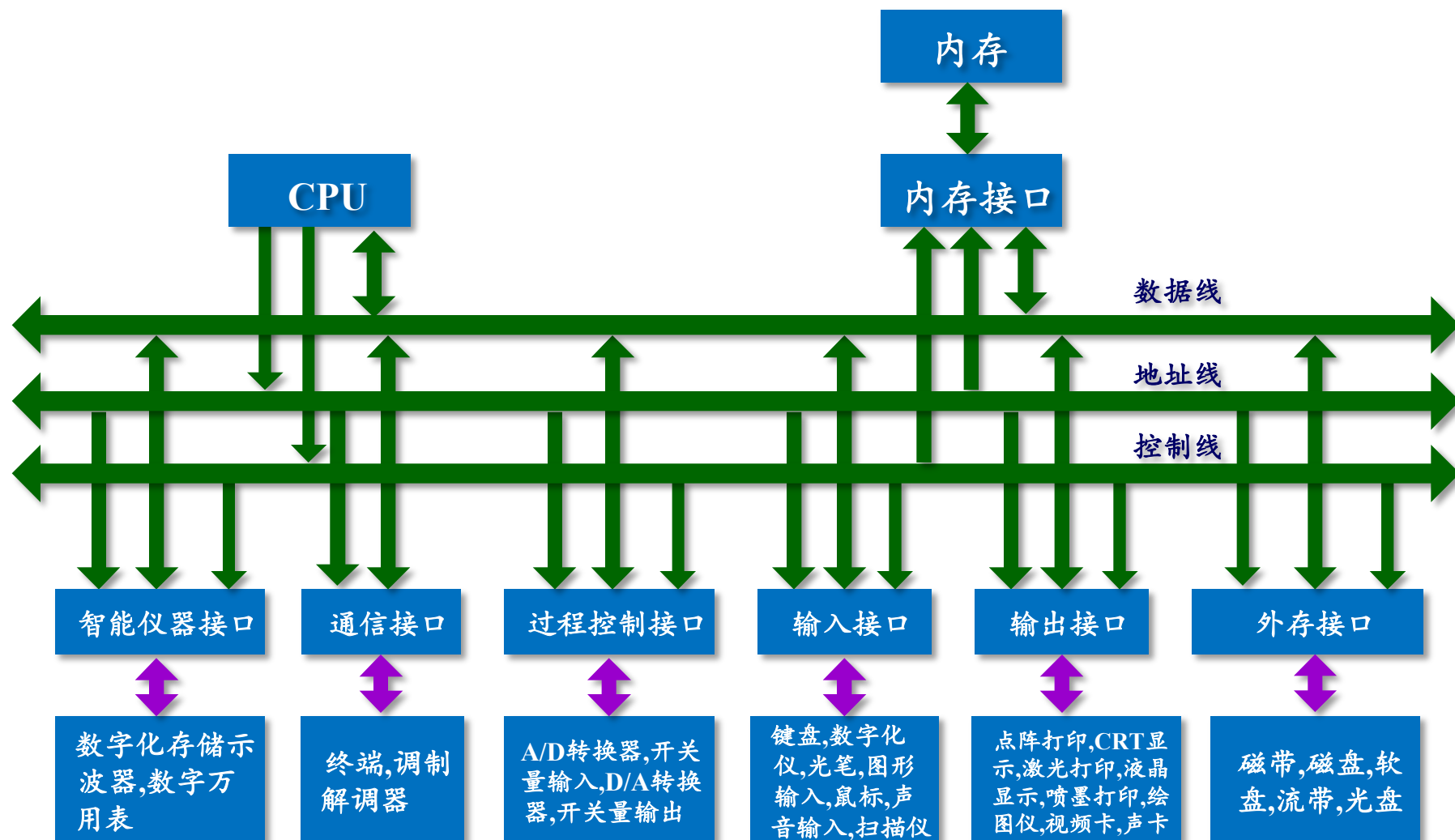


# 接口的基本概念

- 接口可以看成是两个系统或两个部件之间的**交接部分**，它既可以是两种硬设备之间的连接电路，也可以是两个软件之间的共同逻辑边界。
- **I/O接口指的是微处理器与外部世界的连接电路**（或部件）。在这里，外部世界指的是存储器、I/O设备、A/D和D/A转换器、控制设备等。



# 微机系统各类接口





# 接口功能（1）

- 执行CPU命令的功能
  - ◆ 根据CPU发来的命令，转换成控制信号控制I/O设备
- 联络功能（也称：返回外设状态的功能）
  - ◆ 设备发出“准备好”信号
  - ◆ 数据缓冲器发出“空”或“满”信号
- 数据缓冲与锁存功能
  - ◆ 对信号进行隔离、增强总线的驱动能力
  - ◆ 解决I/O设备与CPU速度不匹配的问题
- 寻址功能
  - ◆ 也称为设备选择功能

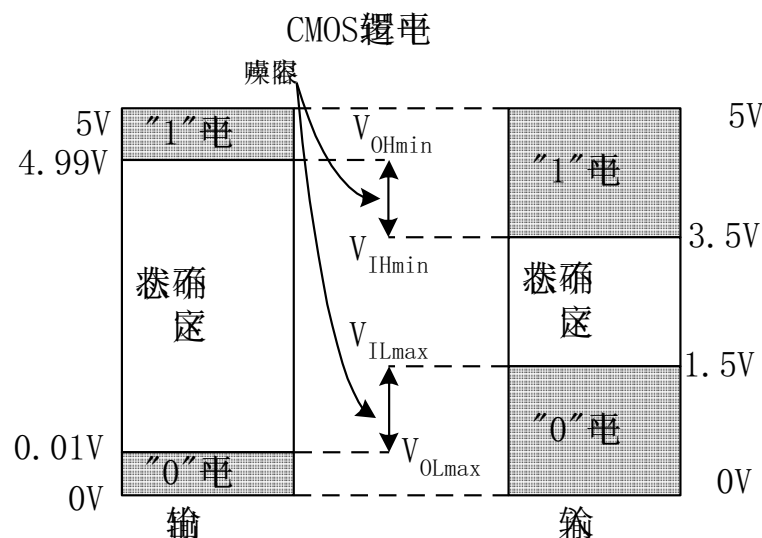
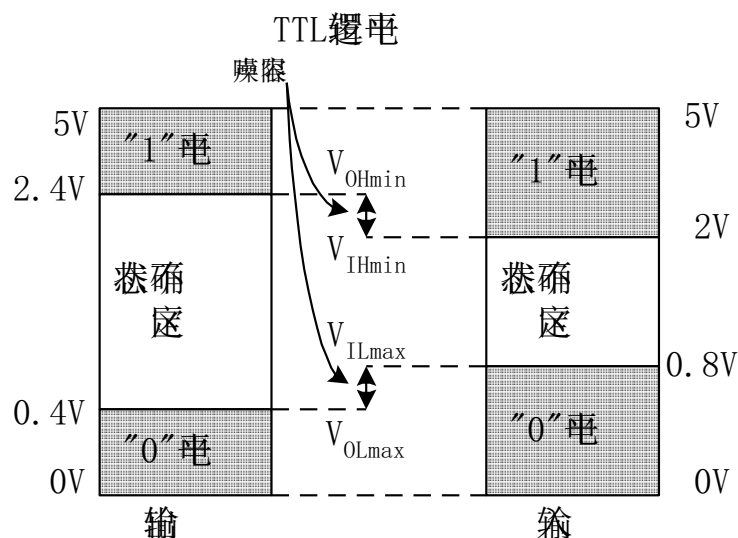


# 接口功能 (2)

## ■ 信号转换功能

◆ 时序配合、电平转换

◆ 例：TTL电路和CMOS电路的逻辑电平



◆ TTL—CMOS接口应有电平转换器

◆ CMOS—TTL驱动能力有限



# 接口功能（3）

## ■ 中断管理功能

- ◆ CPU与中低速外设有效地传递信息，使CPU与外设能并行工作

## ■ 数据转换功能

- ◆ 串 $\leftrightarrow$ 并变换、A/D、D/A转换

## ■ 可编程功能

- ◆ LSI接口芯片的可编程性

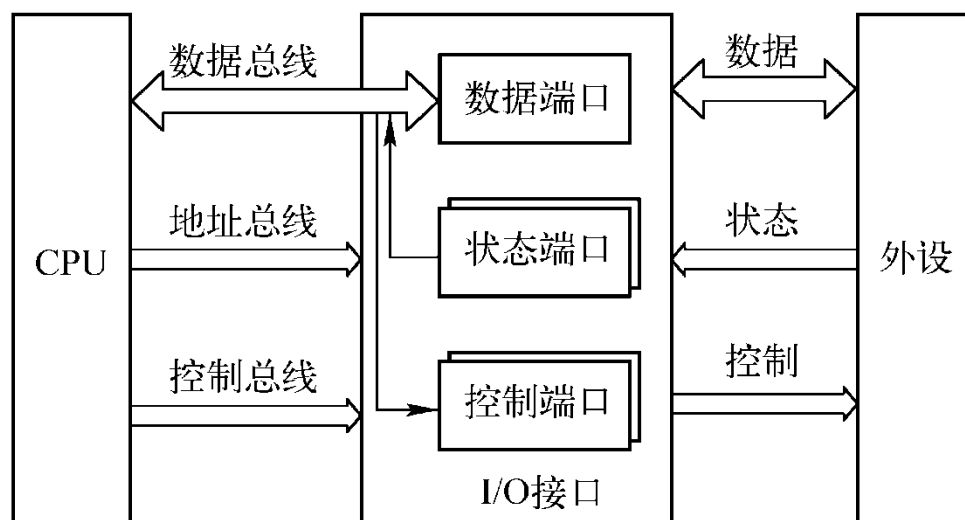






# I/O接口的基本结构

- CPU与外设之间传送三种信息：数据信息、状态信息和控制信息。
- 对于任何一种I/O接口，可抽象为一种I/O接口的基本结构
  - ◆ 数据端口
  - ◆ 状态端口
  - ◆ 控制端口





# 什么是I/O端口？

## ■ I/O端口

- ◆ 接口电路中CPU能直接访问的寄存器的地址
- ◆ 可读写，或只读，或只写
- ◆ 一个接口电路中可有多个端口

## ■ I/O操作

- ◆ 对接口电路中I/O端口的读/写





# 寻址方式

## ■ 统一编址方式（也称为存储器映射）

- ◆ 存储器地址与I/O端口地址共用一个地址空间。访问存储器与访问I/O端口使用同类型指令。I/O空间和存储器空间可以是交错的

## ■ 独立编址方式（也称为I/O映射）

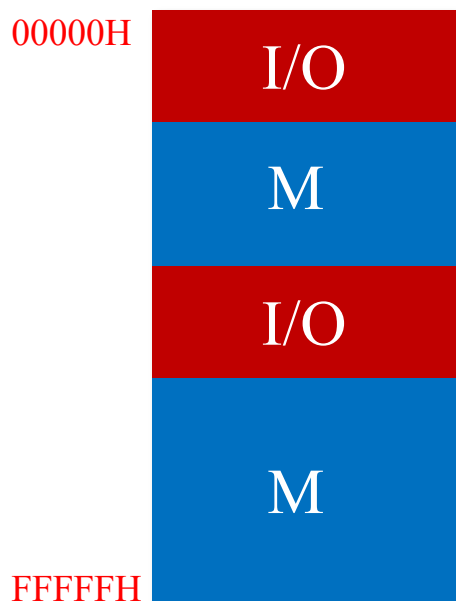
- ◆ 存储器地址与I/O端口地址各有自己独立的地址空间。有专门的I/O指令





# 统一编址方式

- 无专门的输入/输出指令
- 对I/O端口操作灵活（例：对端口内容进行算术逻辑运算、移位等）
- I/O端口占用存储器地址，对I/O操作执行时间长
- 程序的可读性差

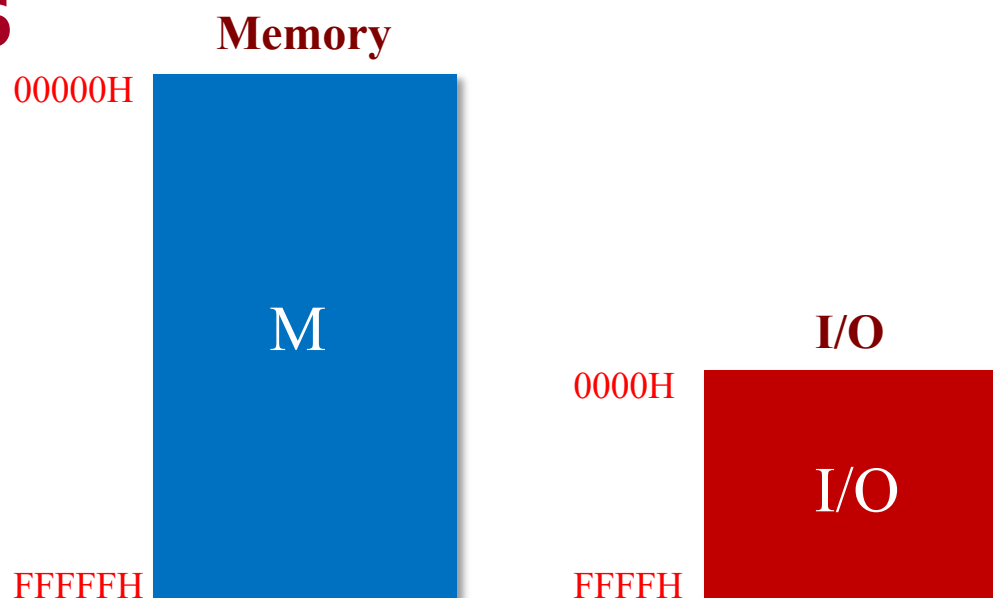




# 独立编址方式

- 有专门的输入/输出指令
- I/O端口不占用存储器地址
- I/O指令短，执行速度快，功能单一
- 程序的可读性强

例：8086





# 举例

## ■ 统一编址

**MOV A, (0100H) ; 0100H可以是I/O端口或内存地址**

## ■ 独立编址

**IN AL, 20H**

**OUT DX, AL**





# PC机系列I/O端口地址分配 (1)

## ■ 系统板上接口芯片的端口地址

I/O 芯片	端口地址
DMA 控制器 1	000~01FH
DMA 控制器 2	0C0~0DFH
DMA 页面寄存器	080~09FH
中断控制器 1	020~03FH
中断控制器 2	0A0~0BFH
定时器	040~05FH
并口	060~06FH
CMOS RAM	070~07FH
协处理器	0F0~0FFH





# PC机系列I/O端口地址分配 (2)

## ■ 扩展槽上接口控制卡的端口地址

I/O 接口名称	端口地址
游戏控制卡	200~20FH
并行口控制卡 1	370~37FH
并行口控制卡 2	270~27FH
串行口控制卡 1	3F8~3FFH
串行口控制卡 2	2F8~2FFH
用户实验卡	300~31FH
同步通信卡 1	3A0~3AFH
同步通信卡 2	380~38FH
单显 MDA	3B0~3BFH
彩显 CGA	3D0~3DFH
彩显 EGA/VGA	3C0~3CFH
硬驱控制卡	1F0~1FFH
软驱控制卡	3F0~3F7H
网卡	360~36FH







# I/O端口地址译码

- 译码电路的两种译码方式
  - ◆ 固定式端口地址译码
  - ◆ 可选式端口地址译码
- 译码电路的实现途径
  - ◆ 门电路译码
  - ◆ 译码器译码
  - ◆ GAL、FPGA器件实现译码



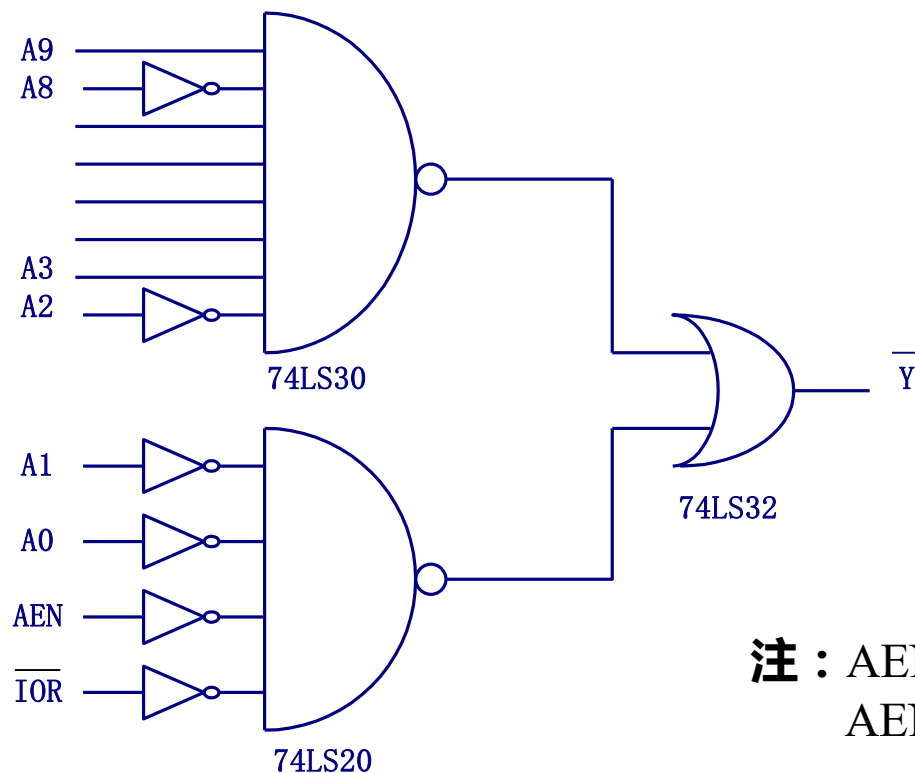


# 固定式端口地址译码 (1)

## ■ 读2F8H端口

$$\overline{Y} = \overline{IOR} + \overline{AEN} + \overline{A0} + \overline{A1} + \overline{A2} + \overline{A3} + \dots + \overline{A7} + \overline{A8} + \overline{A9}$$

$$= \overline{IOR} * \overline{AEN} * \overline{A0} * \overline{A1} + \overline{A2} * \overline{A3} * \dots * \overline{A7} * \overline{A8} * \overline{A9}$$



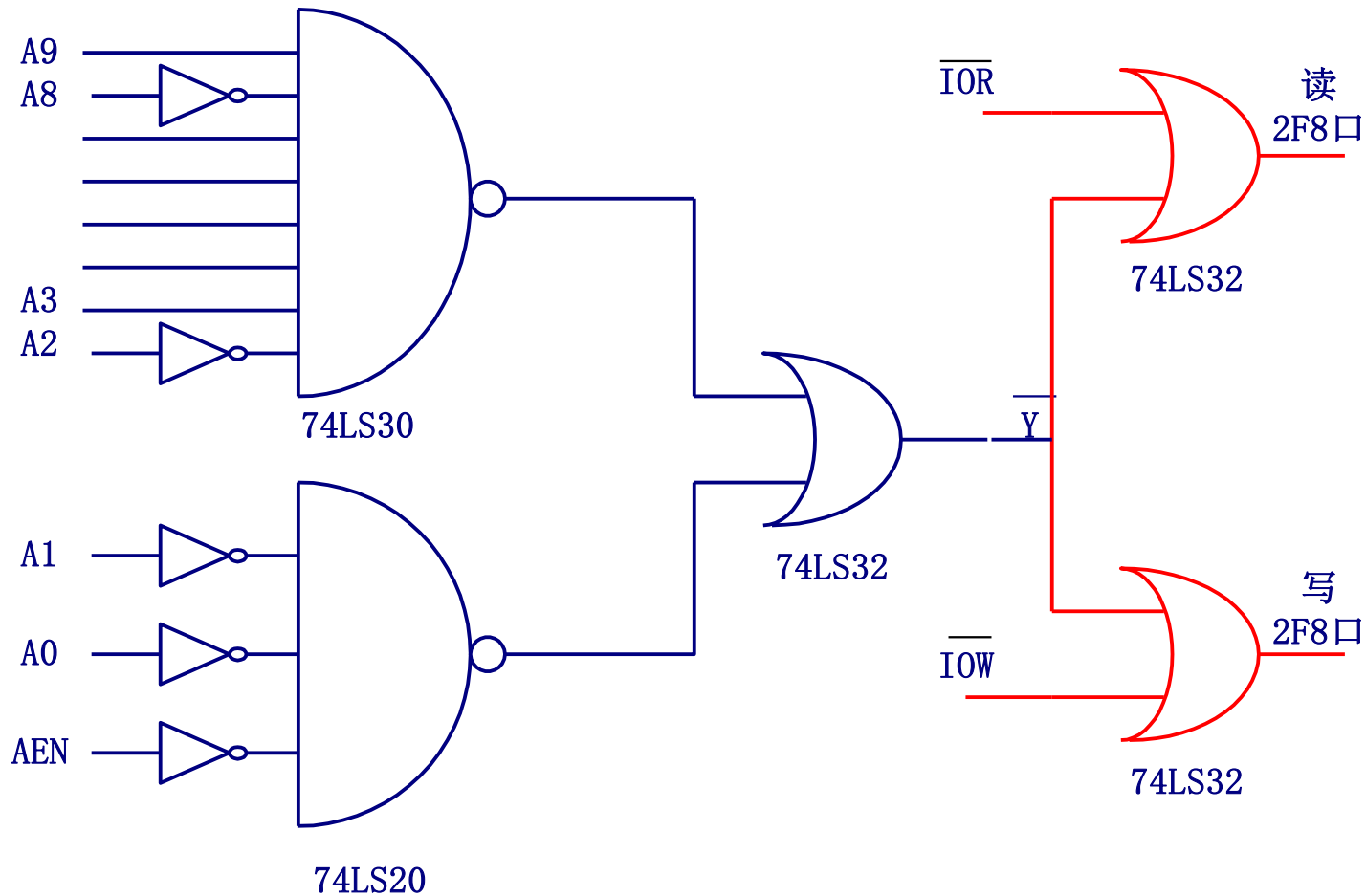
**注：** AEN=0 非DMA传送  
AEN=1 DMA传送





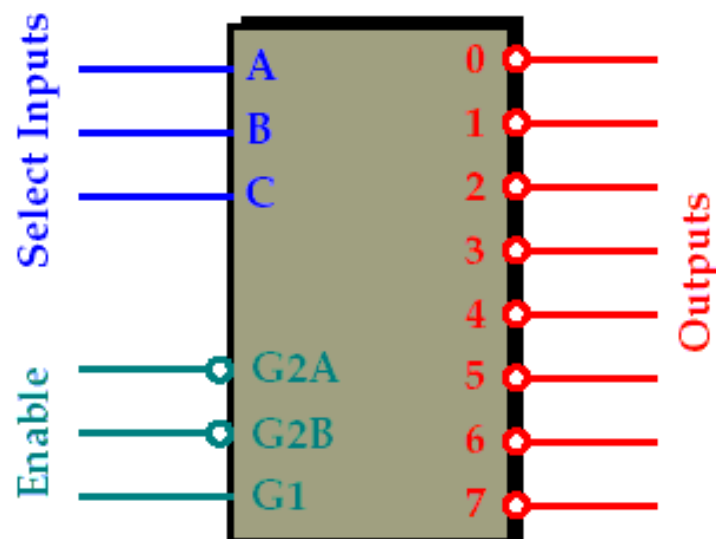
# 固定式端口地址译码 (2)

## ■ 读写2F8H端口



# 固定式端口地址译码 (3)

- 采用译码器可同时译出多个端口地址
  - ◆ 常用译码器有3-8译码器74LS138、4-16译码器74LS154，双2-4译码器74LS139等

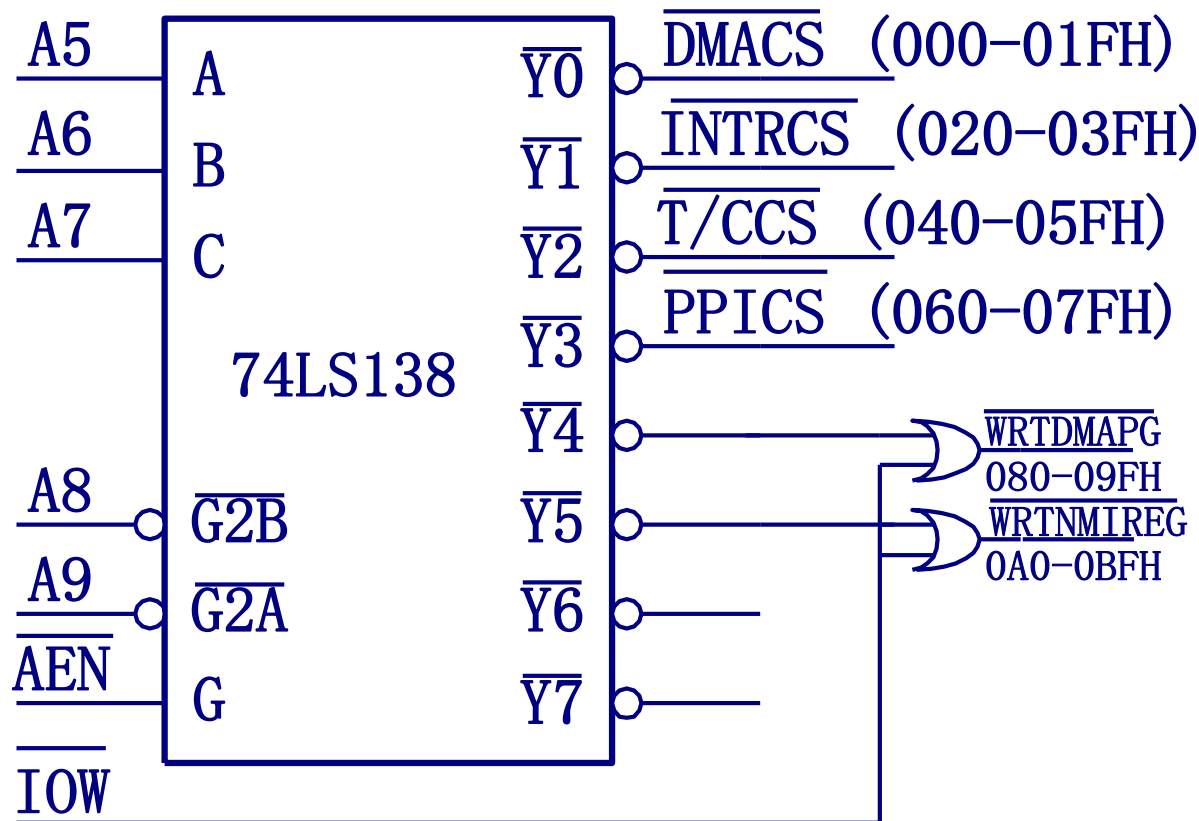


Inputs						Output							
Enable			Select										
G2A	G2B	G1	C	B	A	0	1	2	3	4	5	6	7
1	X	X	X	X	X	1	1	1	1	1	1	1	1
X	1	X	X	X	X	1	1	1	1	1	1	1	1
X	X	0	X	X	X	1	1	1	1	1	1	1	1
0	0	1	0	0	0	0	1	1	1	1	1	1	1
0	0	1	0	0	1	1	0	1	1	1	1	1	1
0	0	1	0	1	0	1	1	0	1	1	1	1	1
0	0	1	0	1	1	1	1	1	0	1	1	1	1
0	0	1	1	0	0	1	1	1	1	0	1	1	1
0	0	1	1	0	1	1	1	1	1	1	0	1	1
0	0	1	1	1	0	1	1	1	1	1	1	0	1
0	0	1	1	1	1	1	1	1	1	1	1	1	0



# 固定式端口地址译码 (4)

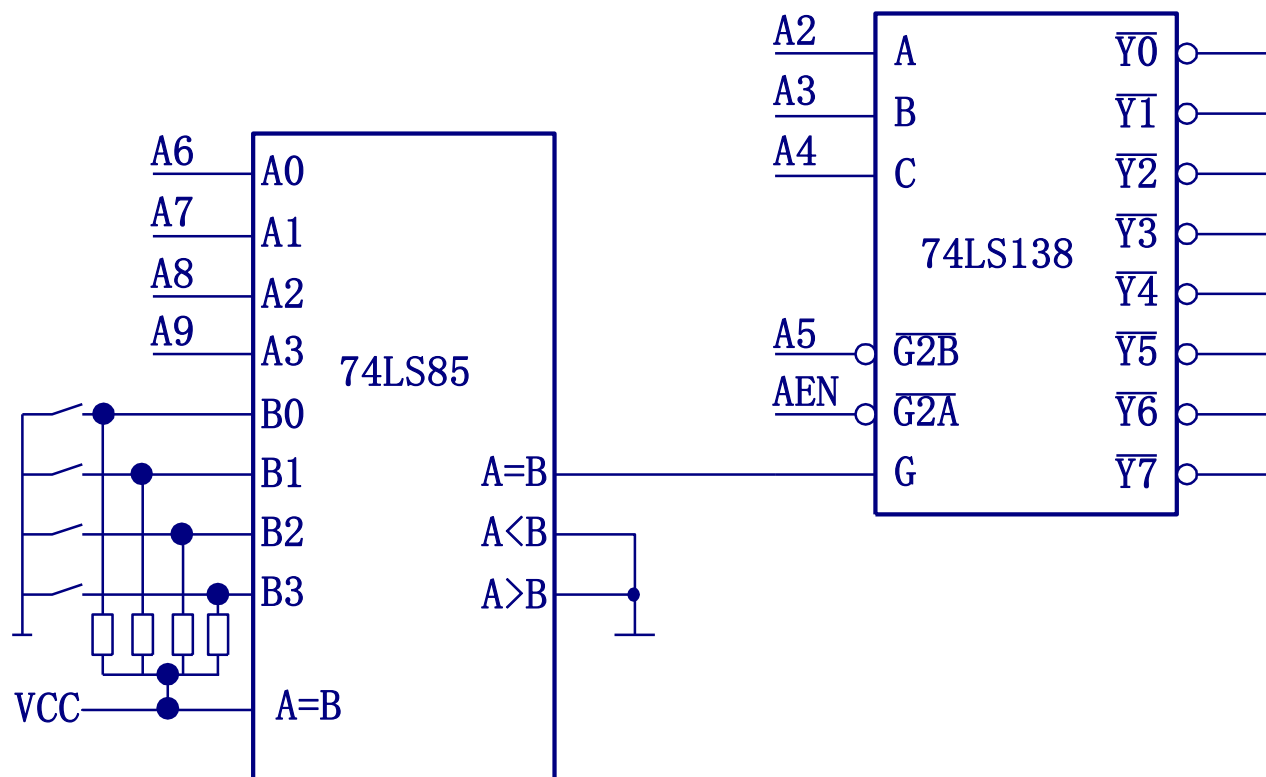
- 例：采用74LS138译码器实现PC机系统板上的端口地址译码





# 可选式端口地址译码 (1)

## 1. 用比较器和开关实现



$B_{0-3} \neq A_{0-3} \Rightarrow \text{输出0}$   
 $B_{0-3} = A_{0-3} \Rightarrow \text{输出1}$

A0-A9的变化:

	000	
	001	
	010	
xxxx 0	011	xx
	100	
	101	
	110	
	111	





# 可选式端口地址译码 (2)

该译码器可译出**16组**不同的地址空间，每组有**32个**地址，即：每组有**8个子组**，每个子组有**4个地址**

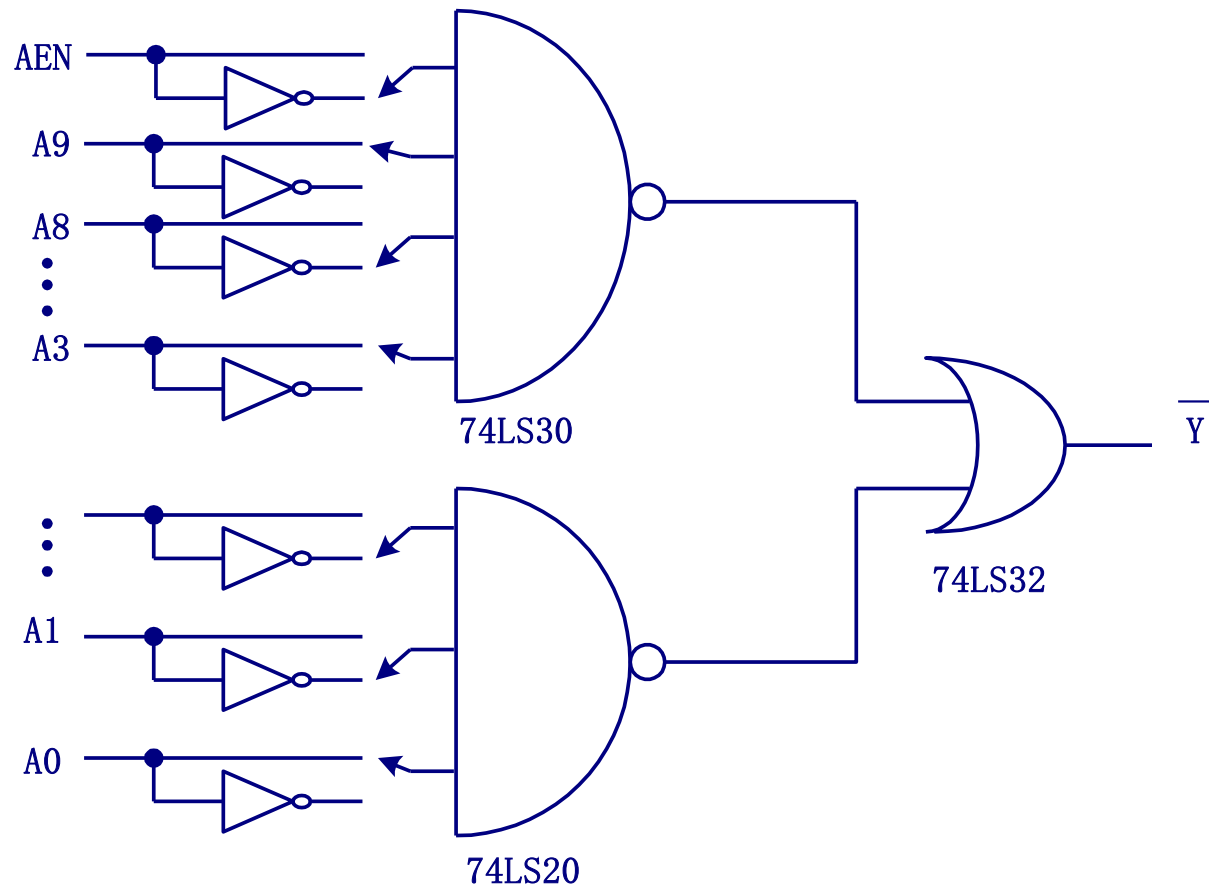
- 译码器可译出的最小地址为：**0**
- 译码器可译出的最大地址为：**3DFH**





# 可选式端口地址译码 (3)

## 2. 用跳接开关实现







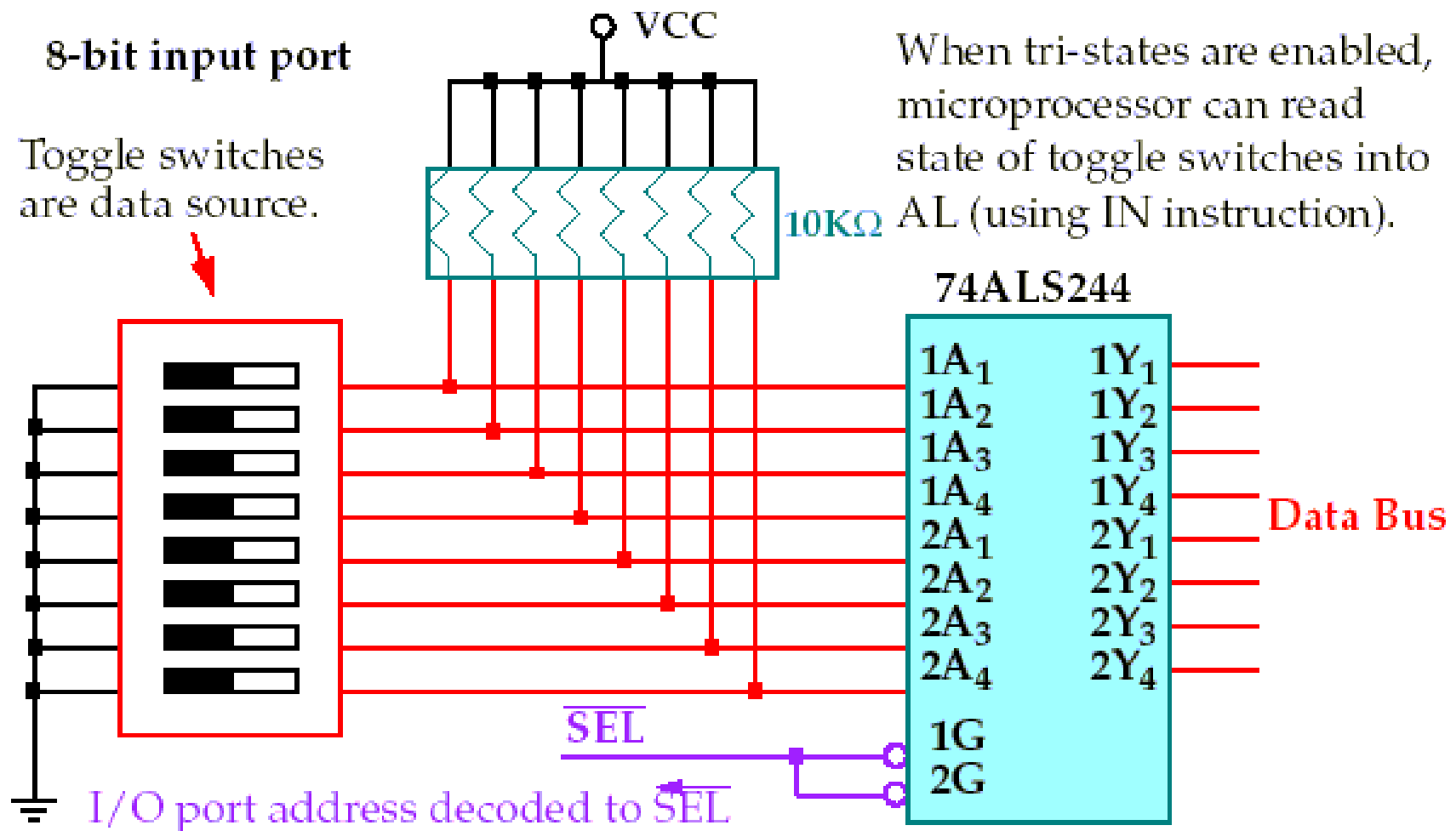
# 基本输入/输出接口

- 基本输入接口（至微处理器）是由一组三态缓冲器构成。
- 基本输出接口（自微处理器）是由一组锁存器构成。





# 基本输入接口





# 复习：三态门和D触发器

三态门和以D触发器形成的锁存器是接口电路中最常使用的两类逻辑电路

## ■ 三态门：功率放大、导通开关

通常**共用**总线的器件需要使用三态电路：

- ◆ 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

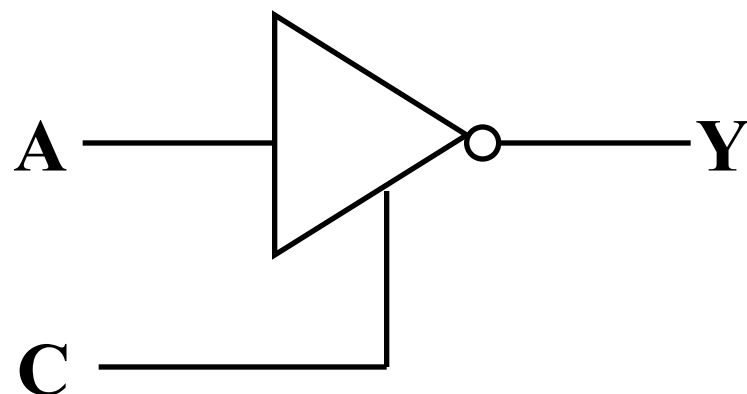
## ■ D触发器：信号保持，也可用作导通开关



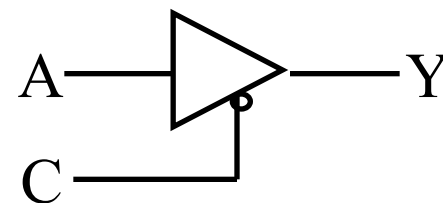
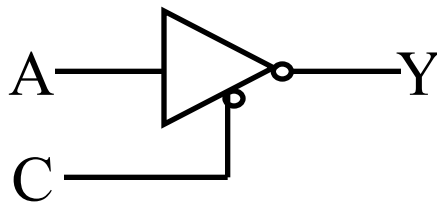
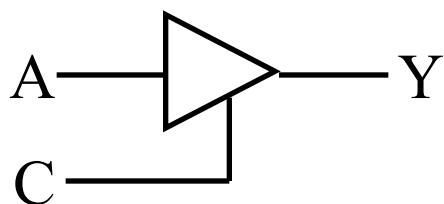


# 三态缓冲器

三态门具有单向导通和三态的特性



**C为低平时：**  
输出为高阻抗（三态）  
**C为高电平时：**  
输出为输入的反相

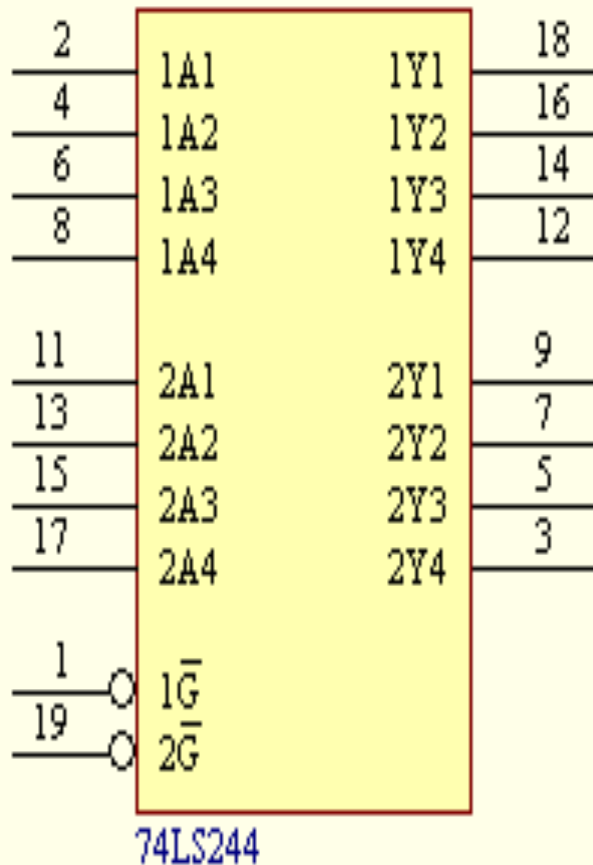


○ 表示反相或低电平有效





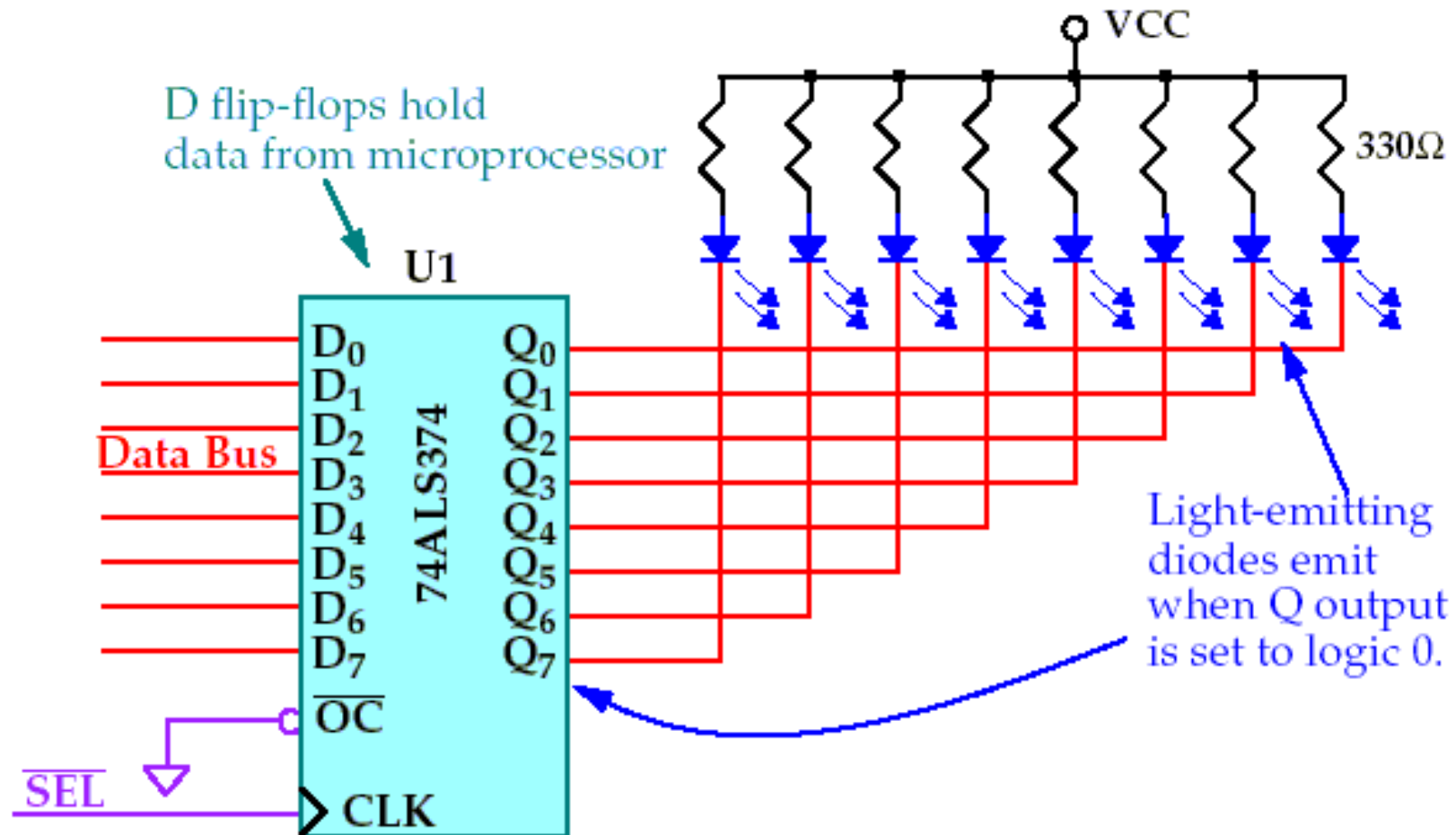
# 74LS244



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



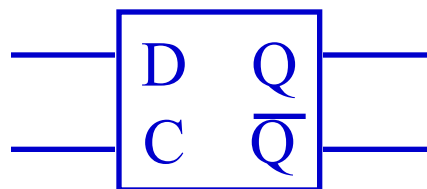
# 基本输出接口



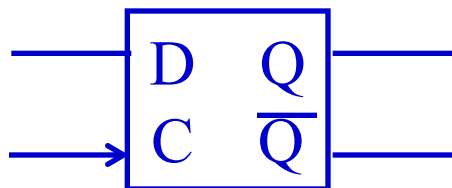
In this case, the data from the OUT instruction is latched using  $\overline{\text{SEL}}$ .



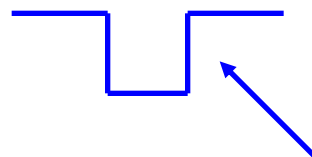
# D触发器



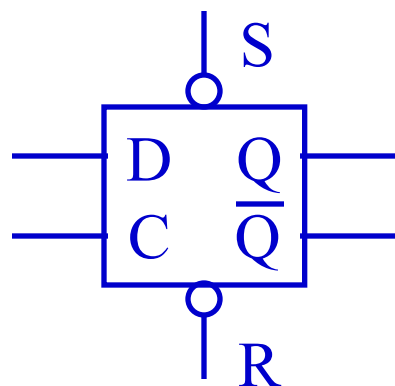
电平锁存：高电平通过，低电平锁存



上升沿锁存：通常用负脉冲触发锁存



负脉冲的上升沿

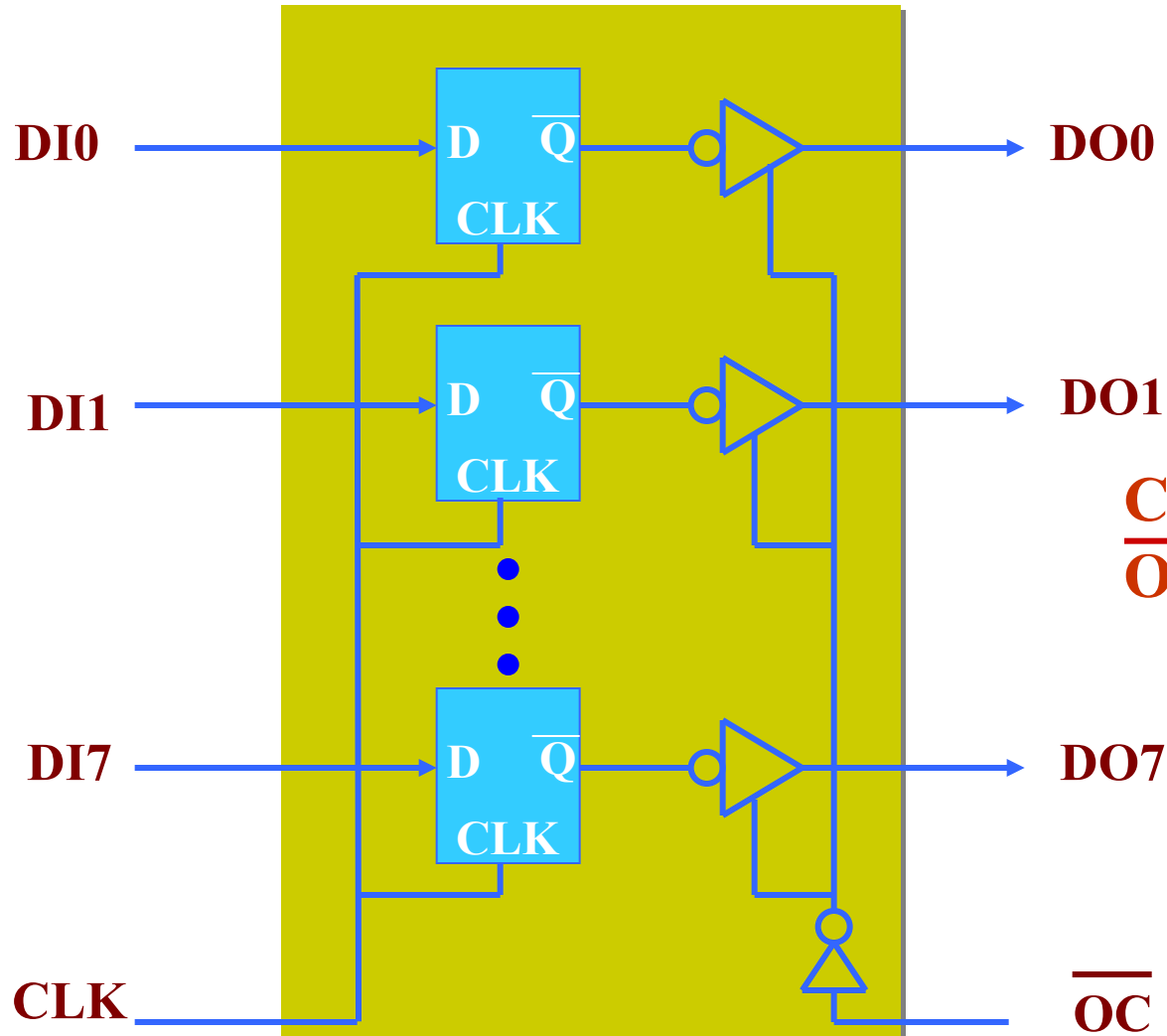


带有异步置位清零的  
电平控制的锁存器





# 74LS374



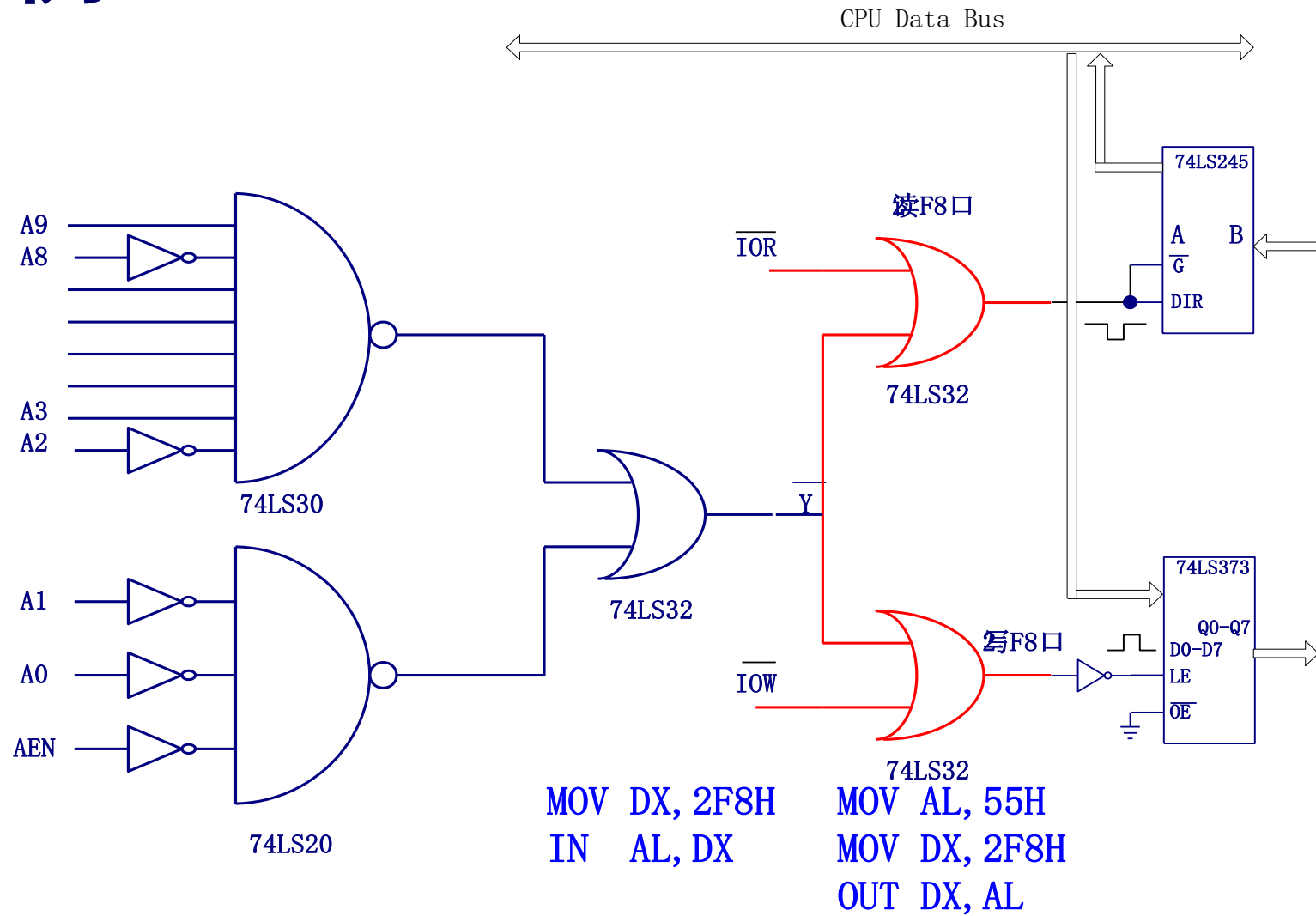
**CLK:** 上升沿锁存  
 **$\overline{OC}$ :** 为0时输出有效  
为1时输出为三态







# 举例





# 373与245真值表

## ■ LS373真值表

$\overline{\text{OE}}$	LE	D	Q
0	1	0	0
0	1	1	1
0	0	x	不变
1	x	x	高阻

## ■ LS245真值表

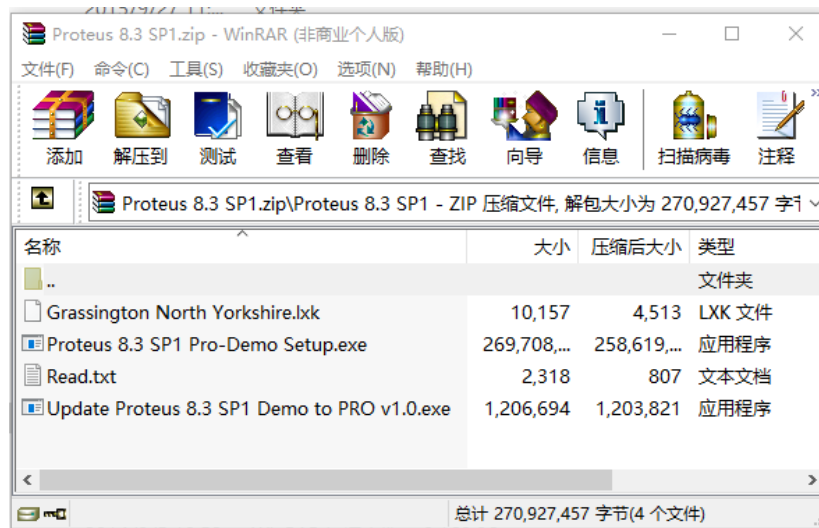
$\overline{\text{G}}$	DIR	D
0	1	$A \rightarrow B$
0	0	$B \rightarrow A$
1	x	三态





# Proteus EDA软件

- Proteus是一个功能众多的**EDA工具**，能实现从原理图输入、代码调试到微处理器与外围电路软硬件协同仿真的功能，并且能一键切换到PCB设计，真正实现从概念到产品的完整设计。是一个将电路仿真软件、PCB设计软件和虚拟系统模型仿真软件三合一的设计工具
- 安装版本 Proteus 8.3 SP1版本





# Proteus EDA软件

Proteus 8.3 SP1版本，安装步骤：

1. 安装Proteus 8.3 SP1 Pro-Demo Setup （以管理员身份运行）

2. 安装注册码：Grassington North Yorkshire.lxk

3. 选择Custom方式

4. 对于Windows x86-64位，安装到：

Installation Path- C:\Program Files (x86)\Labcenter Electronics\Proteus 8 Professional\  
Program Data - C:\Program Files (x86)\Labcenter Electronics\Proteus 8 Professional\

5. 运行Update Proteus 8.3 SP1 Demo to PRO v1.0，目标文件夹

C:\Program Files (x86)\Labcenter Electronics\Proteus 8 Professional\

6. 安装完成

创建或打开一个工程，选中Source Code选项卡：

点击System菜单下的Compilers Configuration，点击MASM32的Download按钮，下载汇编器







## 5.3

# 微机与外设的数据交换方式





# 三种交换方式

- 程序控制方式
  - ◆ 无条件传送方式
  - ◆ 查询方式
- 中断方式
- DMA方式

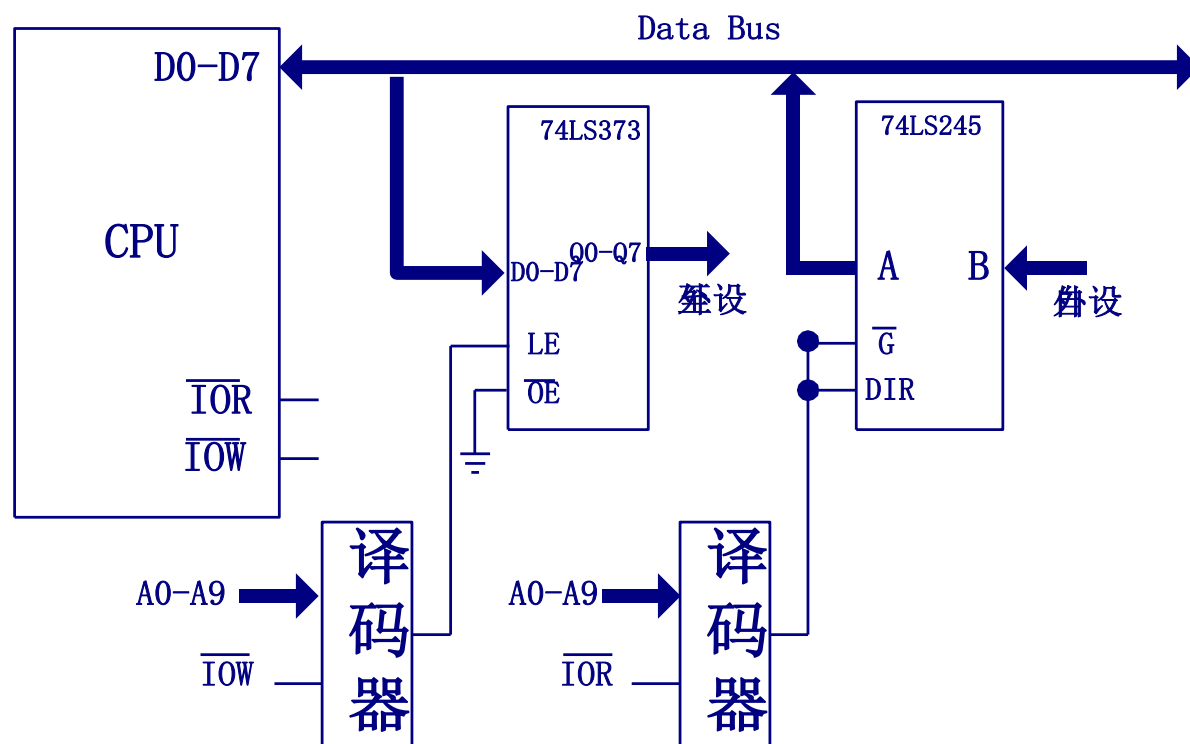




# 1. 程序控制方式 (1)

## 1) 无条件传送方式

- ◆ 输入时假设外设已准备好，输出时假设外设空闲
- ◆ 适用于简单外设的操作







# 1. 程序控制方式 (2)

## ■ LS373真值表

$\overline{\text{OE}}$	LE	D	Q
0	1	0	0
0	1	1	1
0	0	x	不变
1	x	x	高阻

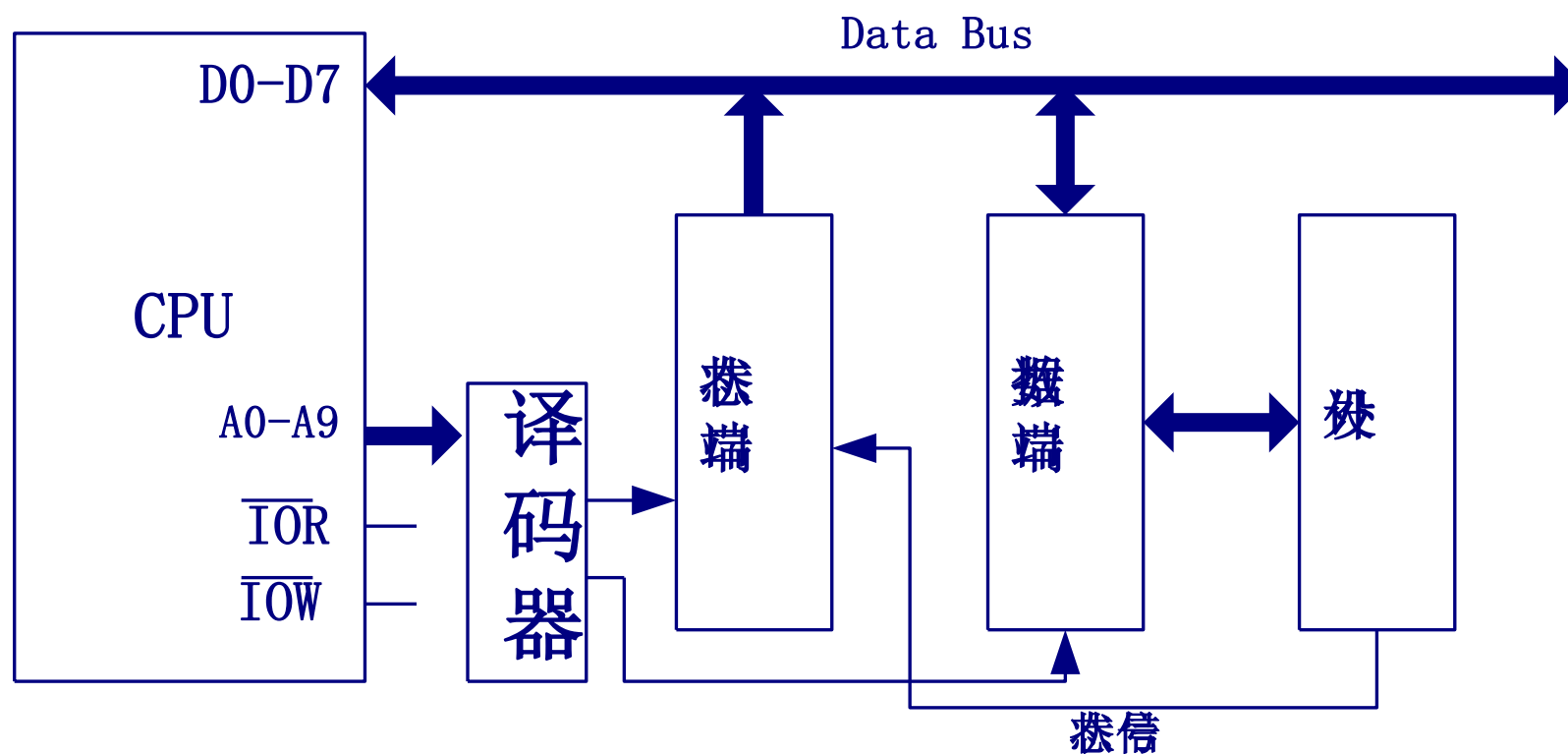
## ■ LS245真值表

$\overline{\text{G}}$	DIR	D
0	1	$A \rightarrow B$
0	0	$B \rightarrow A$
1	x	三态



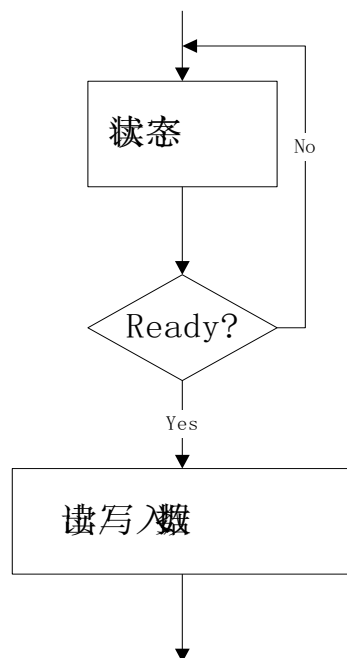
## 1. 程序控制方式 (3)

## 2) 查询方式





# 1. 程序控制方式 (4)



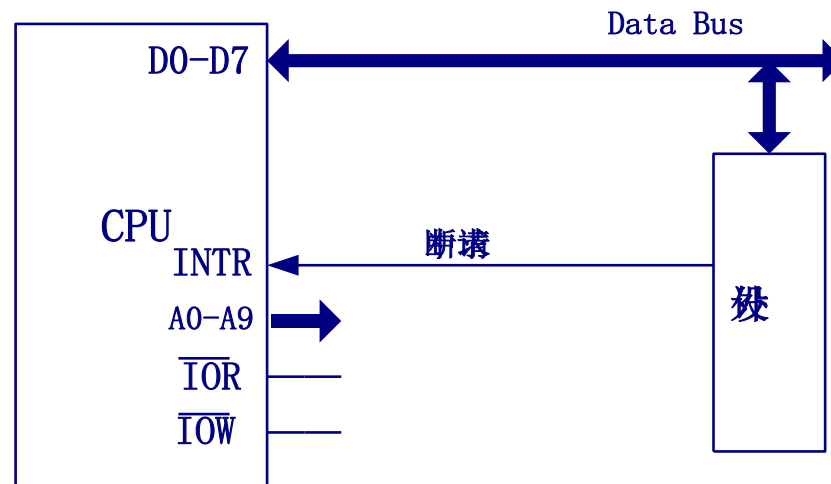
## ■ 特点

- ◆ 效率低，CPU与外设不能并行工作
- ◆ 接口电路简单，硬件开销小
- ◆ 适用于CPU负荷不太大、传送速度不高的系统





## 2. 中断方式



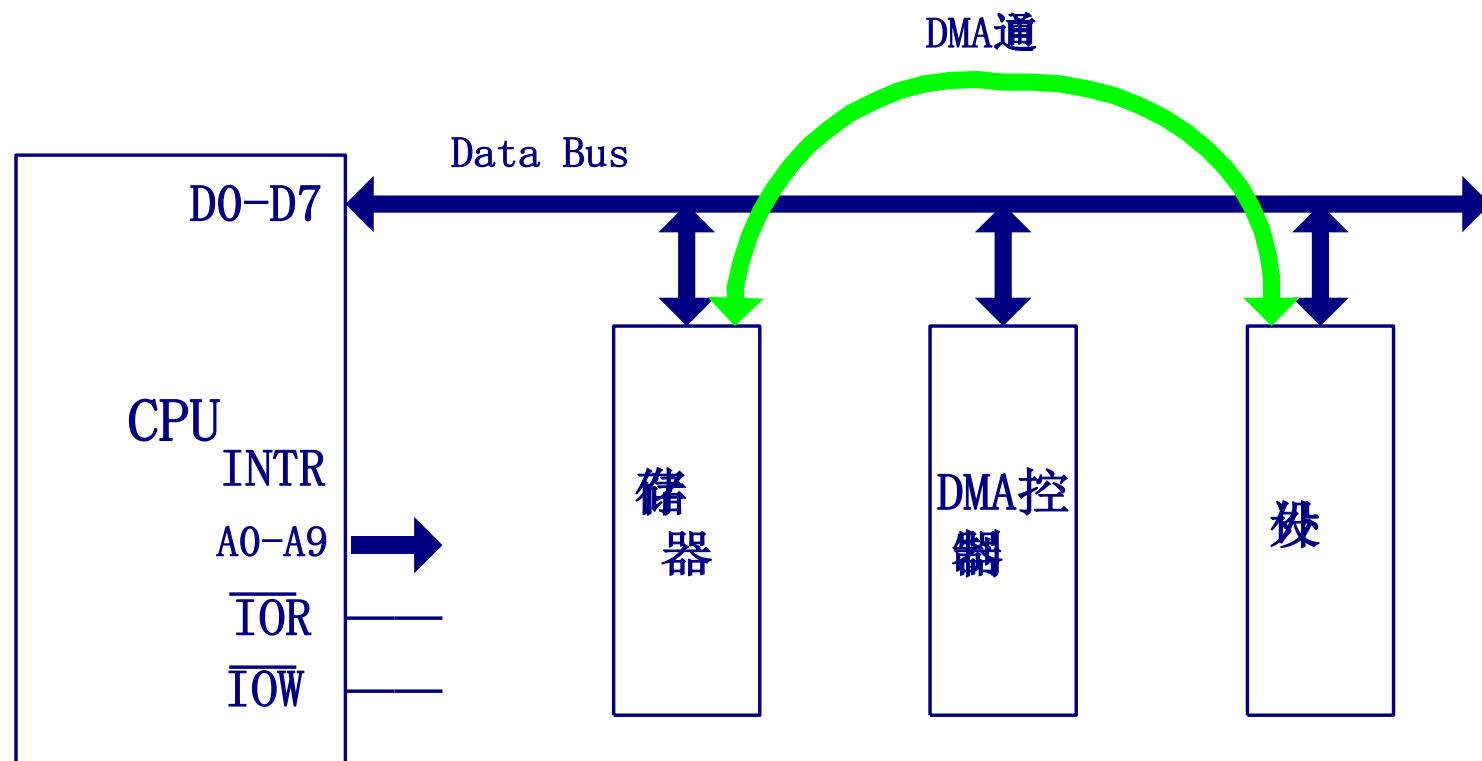
### ■ 特点

- ◆ CPU与外设可并行工作
- ◆ 接口需要中断控制逻辑支持
- ◆ 适用于CPU负荷较大、传送速度不太高的系统





# 3. 直接存储器存取 (DMA) (1)



**Direct Memory Access**





# 3. 直接存储器存取 (DMA) (2)

- 数据传送不经过CPU
- 需要DMA控制器、硬件开销大
- 适用于高速大量数据传送的系统

