

第五章 数字量输入输出

主要内容:

§ 5.1 概述

1. I/O信号的形式
2. I/O接口的功能
3. I/O端口的编址
4. I/O的控制方式

§ 5.2 系统总线及简单接口

1. 总线标准
2. 总线周期
3. 简单的I/O接口
4. 总线信号与接口的连接

第五章 数字量输入输出

主要内容（续）：

几种重要接口及其它们的可编程接口控制器件

§ 5.3 中断控制与 Intel 8259A

§ 5.4 计数定时接口与 Intel 8253

§ 5.5 并行输入输出接口与 Ins 8255

§ 5.6 串行输入输出接口与 Ins 8250

§ 5.7 直接存储器存取控制与 Intel 8237

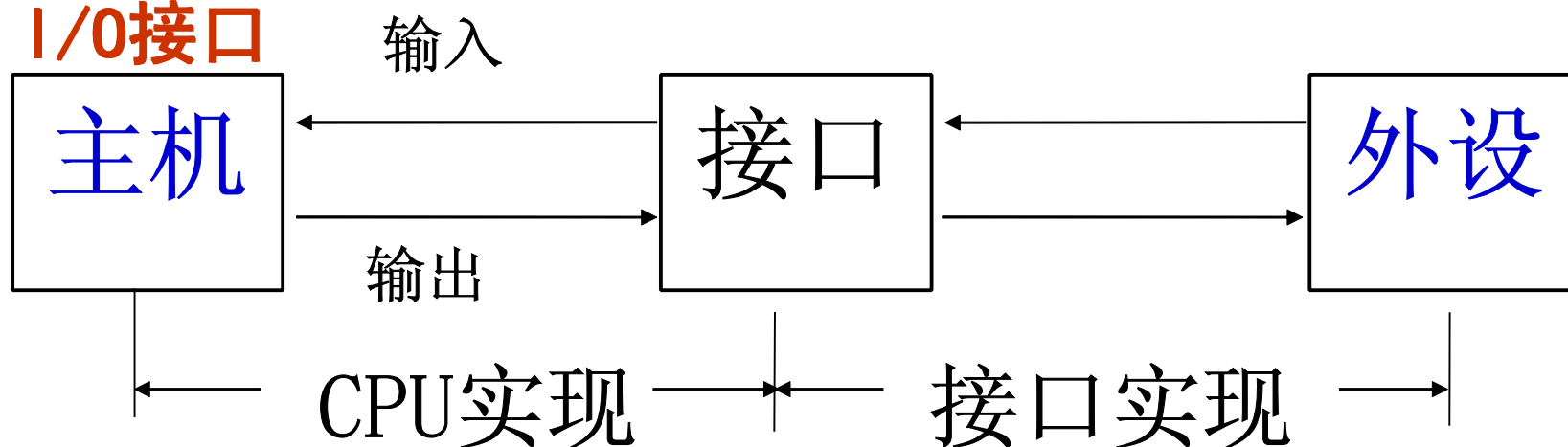
第五章 数字量输入输出

§ 5.1 概述

一、I/O信号

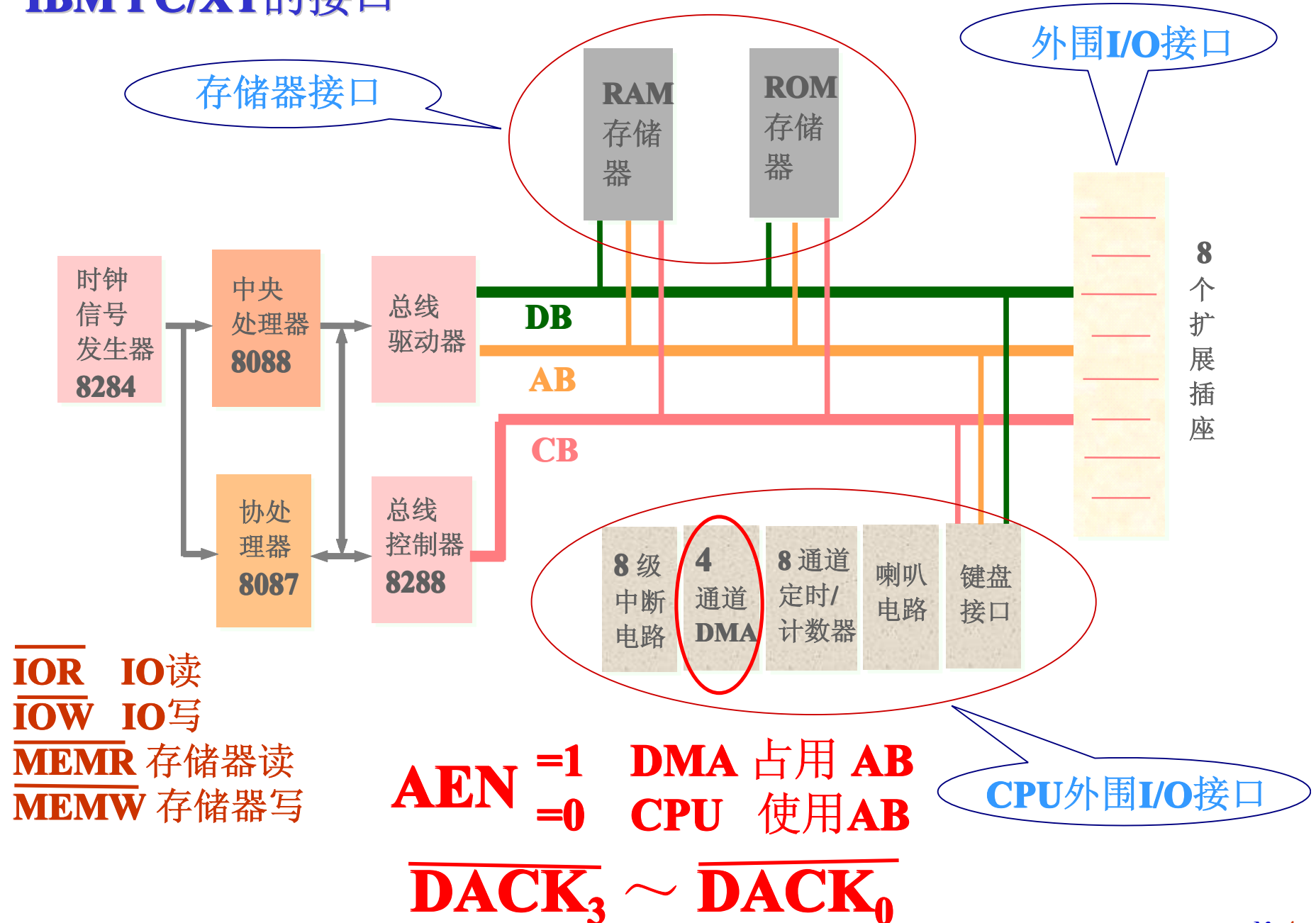
- 1. 开关信号 只有两状态0、1
- 2. 数字信号 多位二进制
- 3. 脉冲信号 0/1的跳变，用于选通或计数
- 4. 模拟信号 数值和时间上连续变化

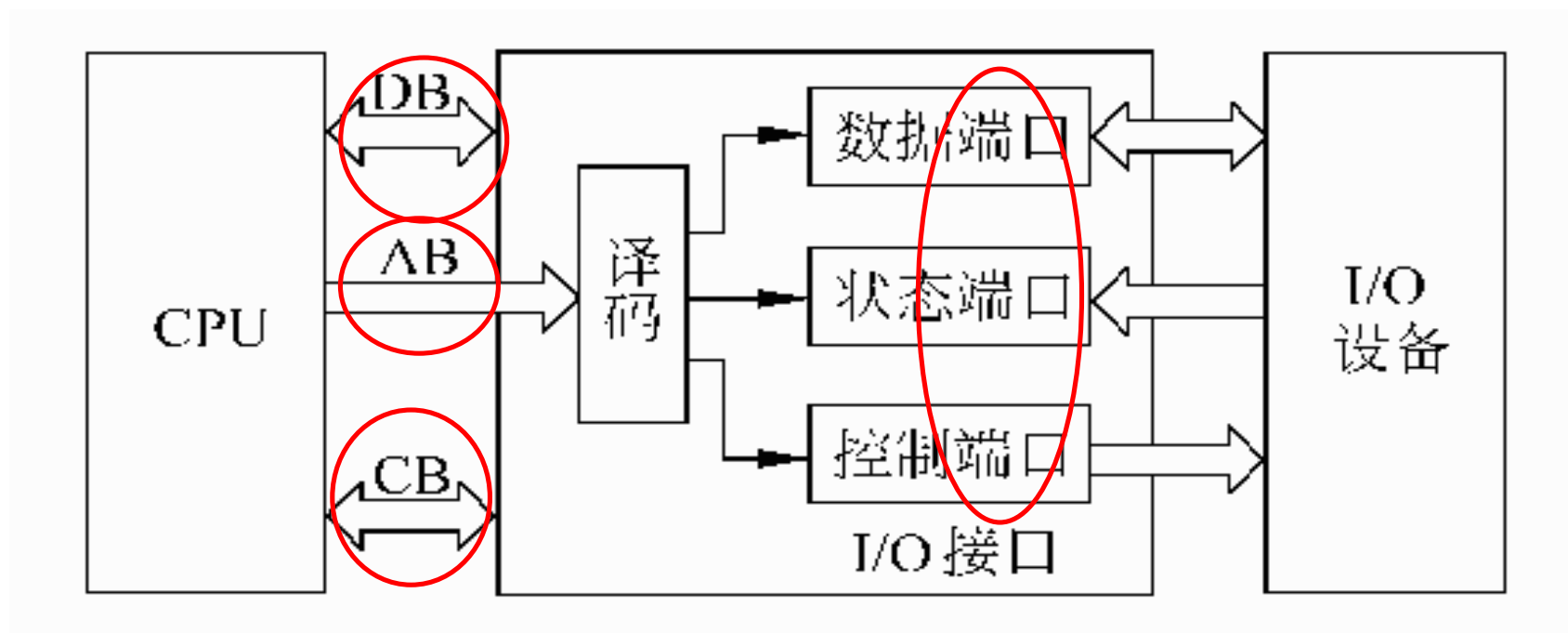
二、I/O接口



计算机由面向外设变为面向接口

IBM PC/XT的接口





P230 图 5.1 典型的I/O接口

不同端口应占用不同地址。

数据锁存及缓冲

数据、状态和控制这三类端口I/O的内容仅在外设表现不同作用，在CPU执行I/O指令时都是通过数据线传送。

地址译码

每一类端口可能不只一个。

控制逻辑

I/O端口

I/O端口：是I/O接口中的存储器，

CPU用I/O指令、按I/O地址进行读/写。

例： IN AL , 40H ; 从40H端口输入
 MOV DX , 200H
 OUT DX , AX ; 向200H、201H端口输出双字节

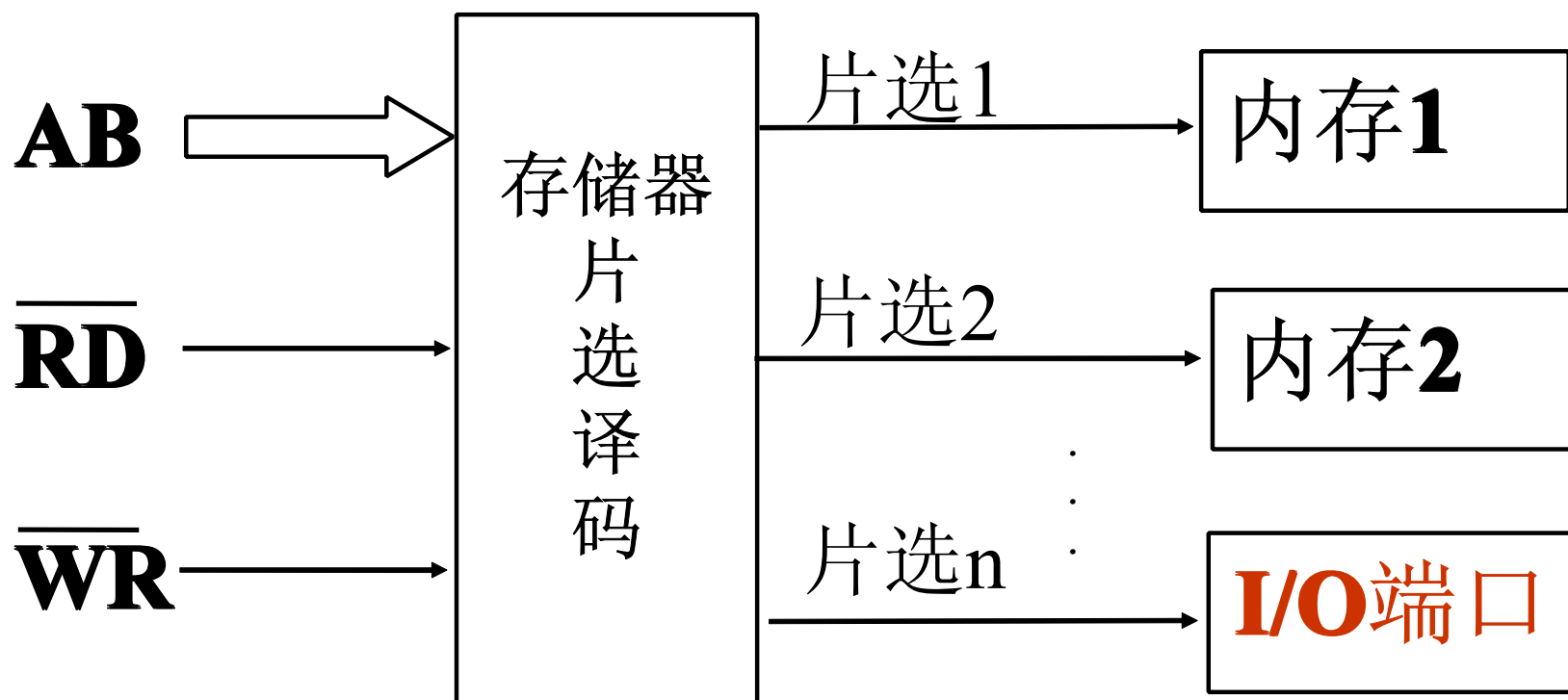
CPU使用16位I/O地址只能用DX寻址。

系统(IBM PC)限制：只使用10位地址表示I/O空间，

用户开发的I/O端口地址只能用200H以上。

三、I/O端口的编址

1. 存储器映象—I/O端口占用存储器的地址，
CPU用访问存储器的指令进行I/O，
控制总线上产生访问存储器的信号。



存储器映象方式是端口地址和存储器地址按照统一的方式编址。也叫做**统一编址方式**

其优点：

- ①**CPU**对外设（端口）操作可使用全部的存储器操作指令，故可用指令多，使用方便；
- ②内存和外设（端口）的地址分布空间是同一个；
- ③不需要专门的输入输出指令。

缺点：

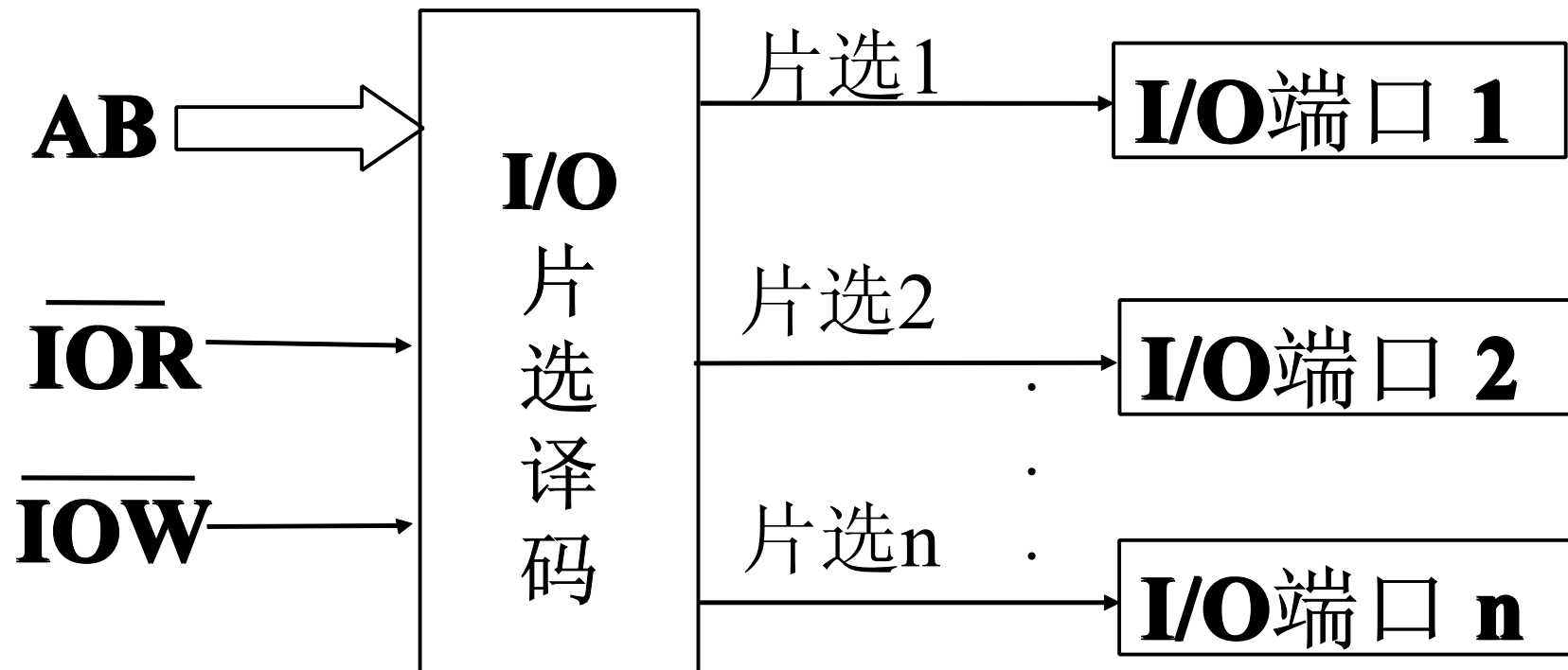
- ①外设占用了内存单元，使内存容量减少；
- ②访问内存指令长度一般比专用**I/O**指令长，从而指令周期较长，执行速度较慢。

2. I/O单独编址

I/O端口用单独的地址空间(单独的地址译码)

CPU用专门的I/O指令,

总线上产生I/O的控制信号。



I/O独立编址方式优点:

- ①**I/O**端口不占内存空间;
- ②由于**I/O**指令都是使用累加器 (**AX/AL**) 操作, 指令字节数少, 因此指令周期较短。

缺点是:

- ①要求**CPU**有**专用I/O指令**, 用于**CPU**与**I/O**端口间数据传输;
- ②用于**I/O**操作的指令少, 因此不够方便灵活。

四、I/O的控制方式

一次I/O数据传送如何启动，又如何完成。

1. 直接I/O(又称无条件I/O、同步I/O)

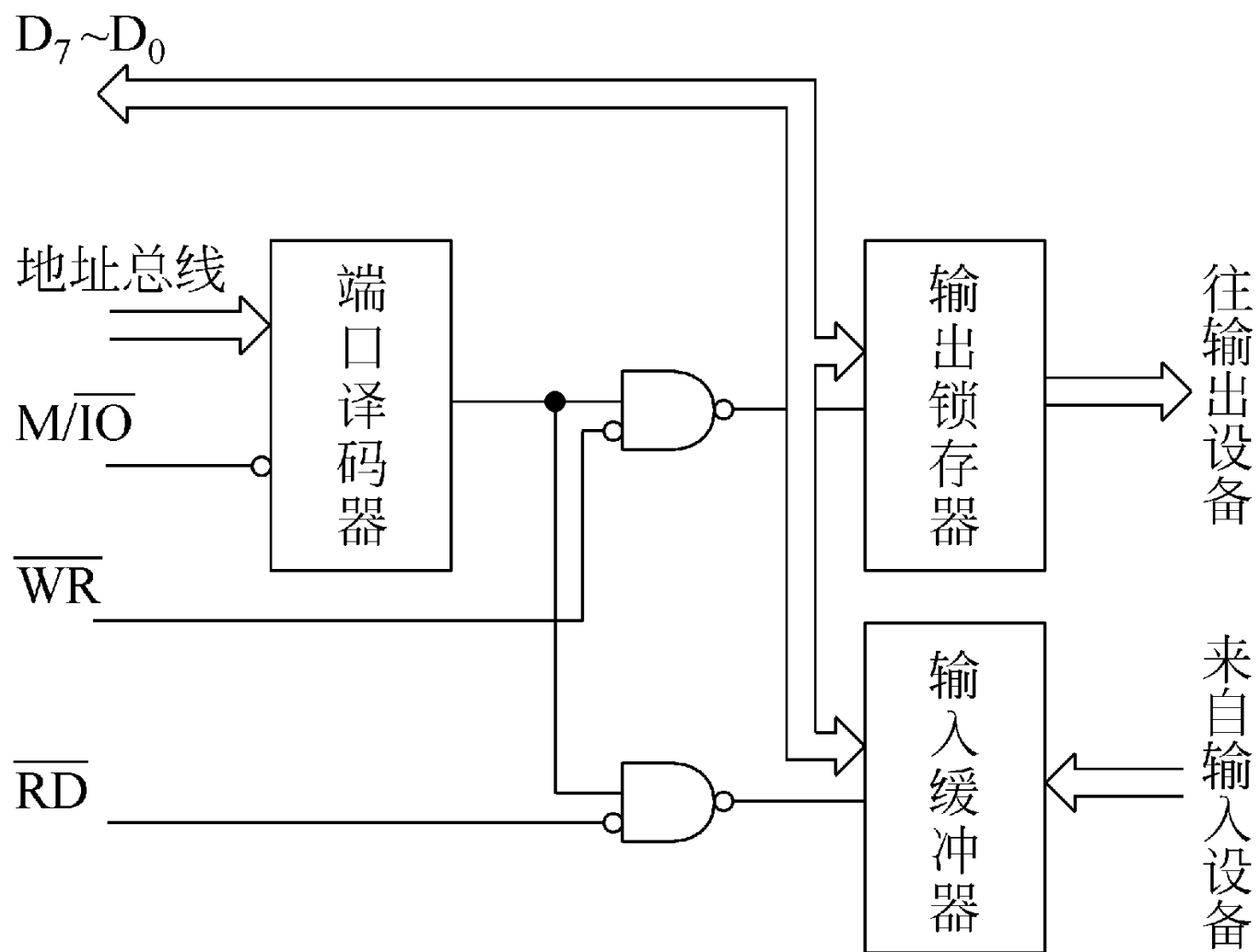
直接读写数据端口

不需要先查询状态条件

要求CPU和外设始终同步

一条I/O指令既启动又完成数据传送, 属程序控制I/O方式。

接口最简单，但使用有限制(即、同步)。



无条件传送方式接口电路

例如，

```
MAIN PROC near  
    ⋮  
    IN  AL, 30H  
    ⋮  
    OUT 2FH, AL  
    ⋮  
MAIN ENDP
```

这种传输方式下的程序设计较简单,不过要保证每次传送时,外设处于就绪状态，只用在对一些简单外设的操作，如开关、七段**LED**显示器等。

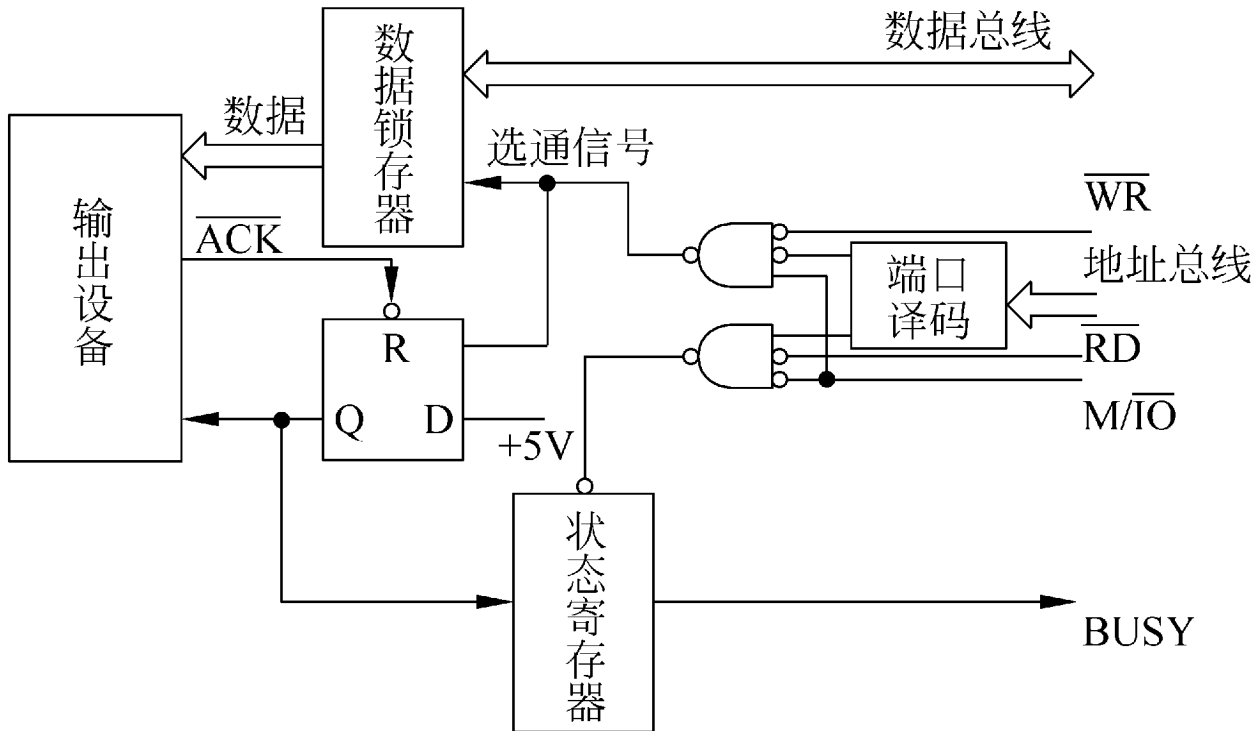
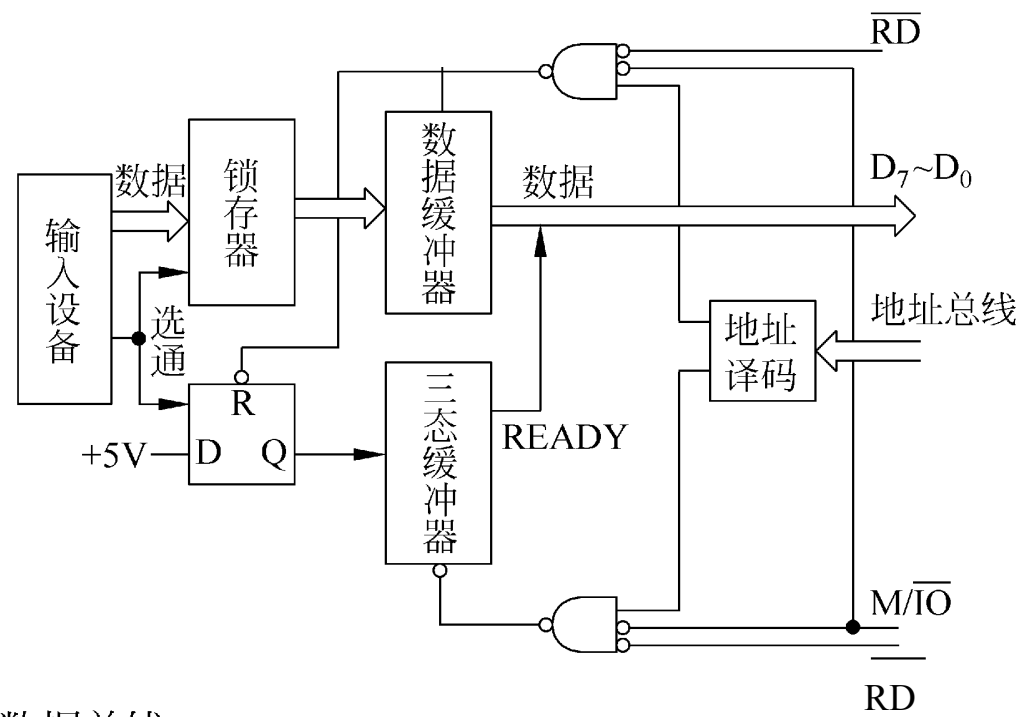
2.条件传送方式

条件传送也称为[查询式传送](#)。用条件传送方式时，**CPU**通过执行程序不断读取并测试外设的状态。如果外设处于准备好状态（输入设备）或者空闲状态（输出设备）时，则**CPU**执行输入指令或输出指令与外设交换信息。

对于条件传送来说，一个数据传送过程由**3**个环节组成：

- ①**CPU**从状态端口中读取状态字。
- ②**CPU**检测状态字的对应位是否满足“就绪”条件，如果不满足，则返①继续读取状态字。
- ③如状态字表明外设已处于“就绪”状态，则传送数据。执行**IO**传送指令

查询式输入的接口电路



查询式输出接口电路

例：查询式输入程序实例：从端口地址为 **60H** 的外设输入一个字节数据。因外设速度较慢，故采用查询方式，状态口为 **61H**，状态位为 **D₁** = **1** 表示 **READY**，程序如下；

XML: IN AL, 61H ; 读入状态字

TEST AL, 02H ; 测试状态位是否为1，不为1转

JZ XML ; XML

IN AL, 60H ; 已准备好，输入一个字节数据

例：查询式输出程序实例；**STATUS_PORT**表示状态端口地址，**READY_BIT**表示状态端口中的准备好位，**DATA_PORT**表示数据端口地址，**DATA**表示输出数据在存储器中存放的位置。

2. 查询方式程序

```
    mov  DX, STATUS_PORT    ; 为测试
Waiting: in  AL, DX
    test AL, READY_BIT      ; 测试“准备好”位
    jz   Waiting            ; 未准备好，查询
    mov  DX, DATA_PORT     ; 为输出数据
    mov  AL, DATA
    out  DX, AL
    mov  DX, STATUS_PORT    ; 为复位
    in   AL, DX
    and  AL, NOT READY_BIT  ; 清除“准备好”位
    out  DX, AL
    .....

```



逻辑运算符

CPU反复**查询**接口**状态**，认为条件成熟才进行
传送

特点：增加状态端口，接口较简单，CPU效率低

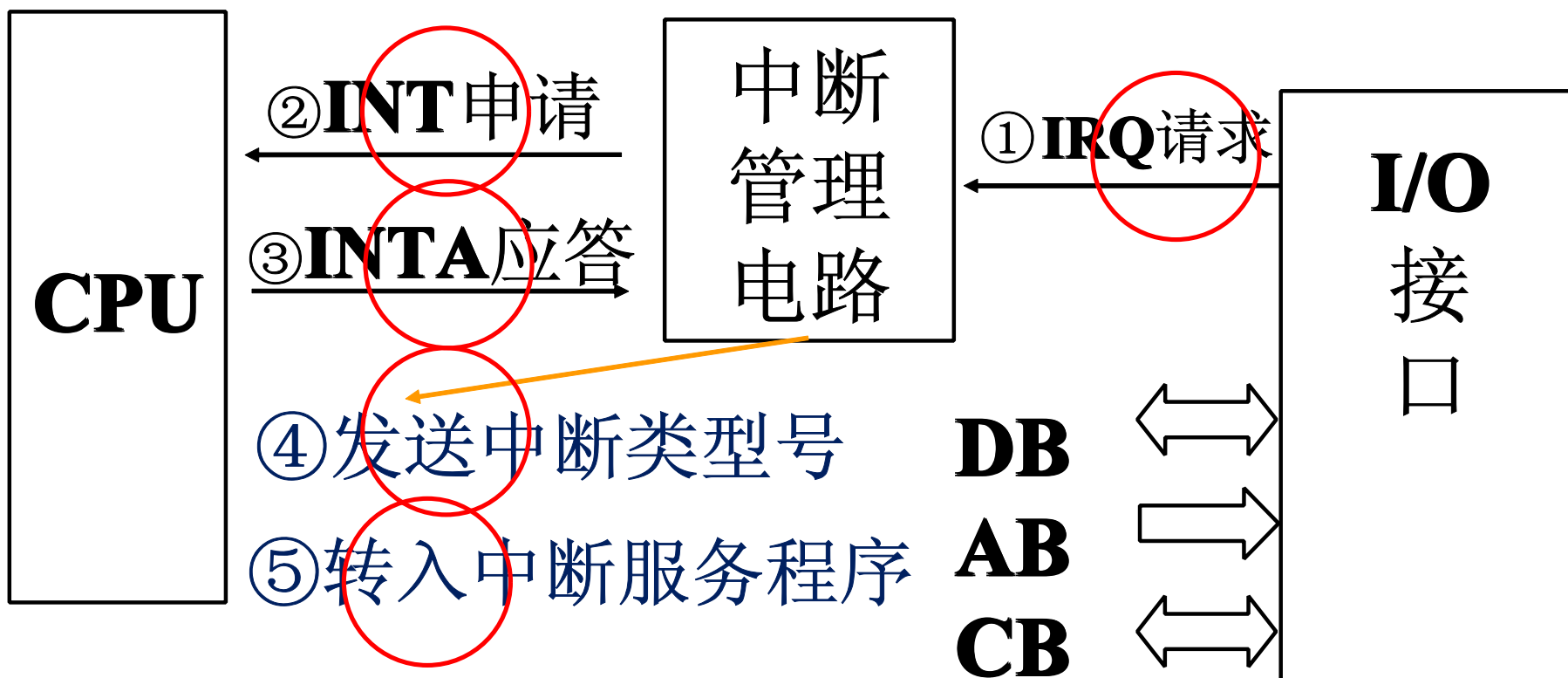
轮询：多个外设可以轮流查询适当提高效率。

查询方式由程序启动，又由程序完成，是典型的**程序控制I/O方式**。

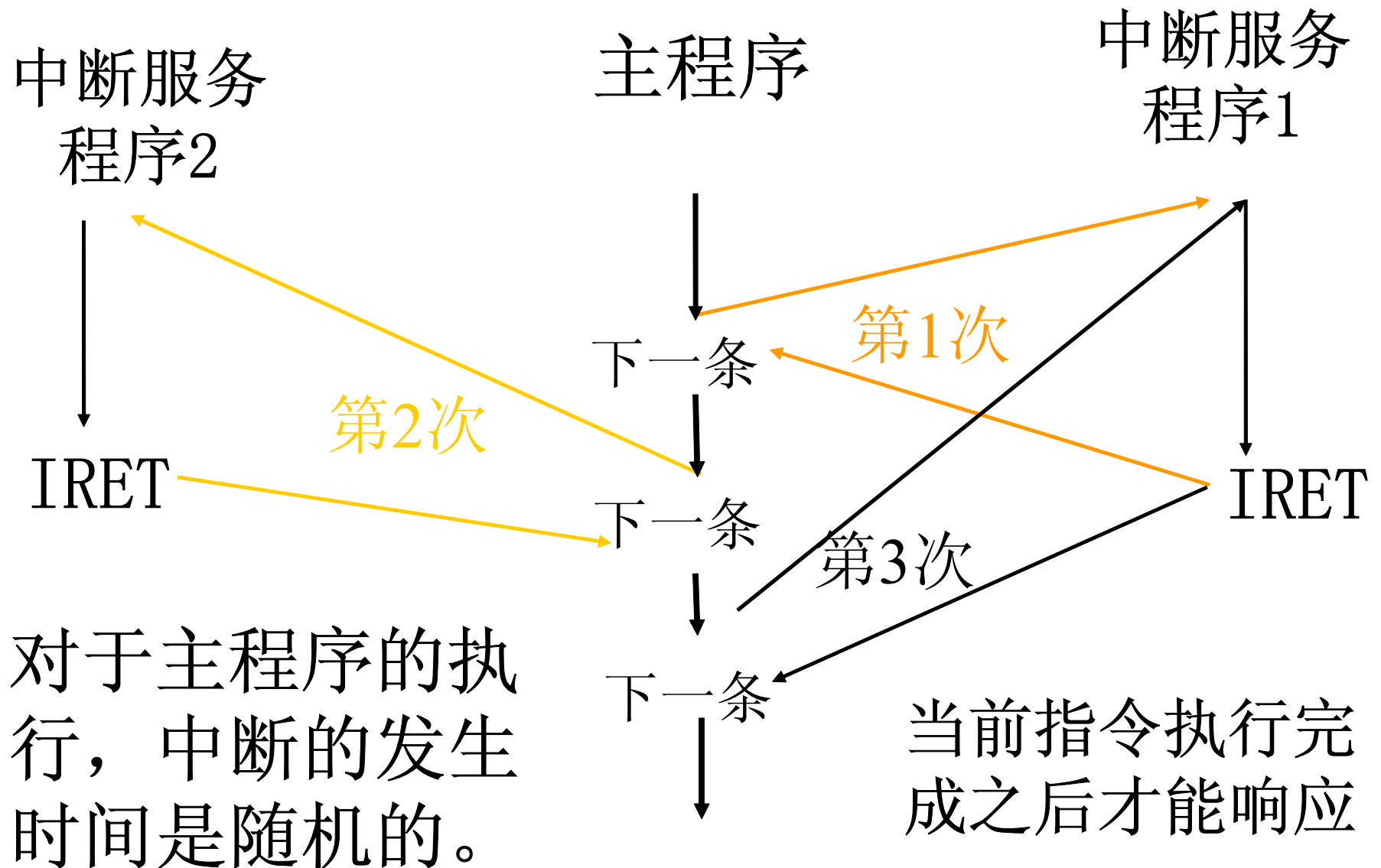
3. 中断方式

CPU和外设并行工作，接口主动提出请求，CPU响应后由**中断服务程序**完成I/O传送。

中断方式的接口及过程



中断方式的程序流程



中断方式特点

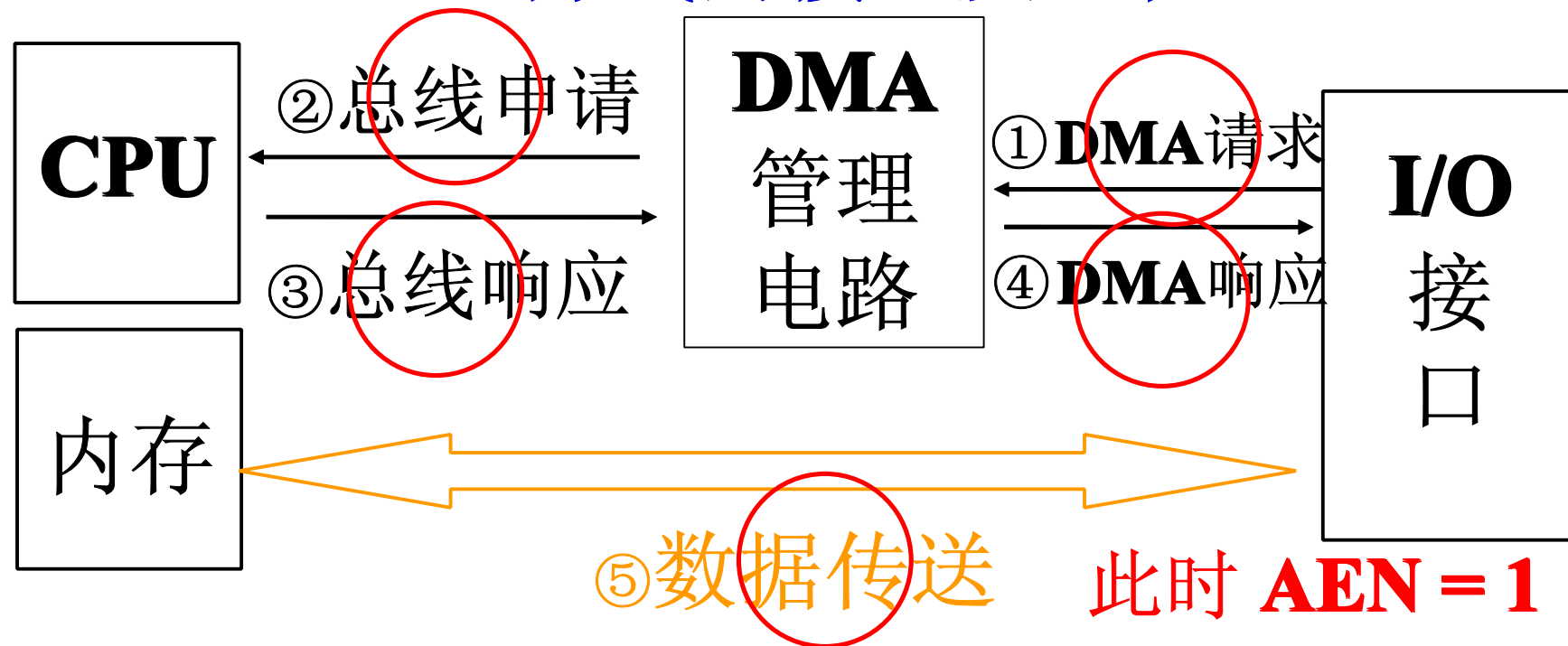
**CPU和外设并行工作，效率较高，
对接口响应较快，
接口较复杂（专用的中断管理电路
例如8259A）。**

**由中断服务(专用)程序完成数据传送。
属于程序控制I/O方式。
使用最广泛。**

4. DMA (直接存储器存取) 方式

CPU和外设并行工作，外设主动提出请求
CPU响应后由DMA控制电路接管总线，完成I/O传送。

DMA方式的接口及过程



DMA方式特点

CPU和外设并行工作效率最高

对外设响应最快(当前总线周期结束)

接口最复杂

(专用的**DMA**管理电路,例如**8237**)

在高速外设中广泛使用。

5. 其他方式

I/O处理机 (独立于CPU之外的协处理机)

在保证CPU和外设协调传送的前提下,
尽量提高速度。

§ 5.2 系统总线及简单接口

总线：多个电路传送信号的公共通道

各电路有三态输出

**任何时候只能其中一个器件往传输线上输出
“0”或“1”（其他器件输出呈高阻态）**

不同范围的总线：

芯片级（芯片总线）如CPU引脚

**系统级（内总线） 如：PC/XT（8位数据总线），
ISA（16位），EISA（32位），PCI（64位）**

设备级（外总线） 如：RS232, GPIB, USB

IO接口是对系统总线的接口

1、 PC / XT系统总线标准

	B	A			B	A	
GND	1	1	$\overline{\text{I/O CH CK}}$ (I)	GND	1	1	$\overline{\text{I/O CH CK}}$ (I)
(O) RESET DRV	2	2	D ₇ (I/O)	(O) RESET DRV	2	2	SD ₇ (I/O)
+5Vdc	3	3	D ₆ (I/O)	+5Vdc	3	3	SD ₆ (I/O)
(I) IRQ ₂	4	4	D ₅ (I/O)	(I) IRQ ₂	4	4	SD ₅ (I/O)
-5Vdc	5	5	D ₄ (I/O)	-5Vdc	5	5	SD ₄ (I/O)
(I) DRQ ₂	6	6	D ₃ (I/O)	(I) DRQ ₂	6	6	SD ₃ (I/O)
-12Vdc	7	7	D ₂ (I/O)	-12Vdc	7	7	SD ₂ (I/O)
RESERVED	8	8	D ₁ (I/O)	(I) OWS	8	8	SD ₁ (I/O)
+12Vdc	9	9	D ₀ (I/O)	+12Vdc	9	9	SD ₀ (I/O)
GND	10	10	I/O CH RDY (I)	GND	10	10	I/O CH RDY (I)
(O) $\overline{\text{MEMW}}$	11	11	AEN (O)	(O) $\overline{\text{SMEMW}}$	11	11	AEN (O)
(O) $\overline{\text{MEMR}}$	12	12	Λ_{19} (O)	(O) $\overline{\text{SMEMR}}$	12	12	SA ₁₉ (O)
(O) $\overline{\text{IOW}}$	13	13	Λ_{18} (O)	(O) $\overline{\text{IOW}}$	13	13	SA ₁₈ (O)
(O) $\overline{\text{TOR}}$	14	14	Λ_{17} (O)	(O) $\overline{\text{TOR}}$	14	14	SA ₁₇ (O)
(O) $\overline{\text{DACK}}_3$	15	15	Λ_{16} (O)	(O) $\overline{\text{DACK}}_3$	15	15	SA ₁₆ (O)
(I) DRQ ₁	16	16	Λ_{15} (O)	(I) DRQ ₃	16	16	SA ₁₅ (O)
(O) $\overline{\text{DACK}}_1$	17	17	Λ_{14} (O)	(O) $\overline{\text{DACK}}_1$	17	17	SA ₁₄ (O)
(I) DRQ ₁	18	18	Λ_{13} (O)	(I) DRQ ₁	18	18	SA ₁₃ (O)
(O) $\overline{\text{DACK}}_0$	19	19	Λ_{12} (O)	(O) REFRESH	19	19	SA ₁₂ (O)
(O) CLK	20	20	Λ_{11} (O)	(O) CLK	20	20	SA ₁₁ (O)
(I) IRQ ₇	21	21	Λ_{10} (O)	(I) IRQ ₇	21	21	SA ₁₀ (O)
(I) IRQ ₆	22	22	Λ_9 (O)	(I) IRQ ₆	22	22	SA ₉ (O)
(I) IRQ ₅	23	23	Λ_8 (O)	(I) IRQ ₅	23	23	SA ₈ (O)
(I) IRQ ₄	24	24	Λ_7 (O)	(I) IRQ ₄	24	24	SA ₇ (O)
(I) IRQ ₃	25	25	Λ_6 (O)	(I) IRQ ₃	25	25	SA ₆ (O)
(O) $\overline{\text{DACK}}_2$	26	26	Λ_5 (O)	(O) $\overline{\text{DACK}}_2$	26	26	SA ₅ (O)
(O) T/C	27	27	Λ_4 (O)	(O) T/C	27	27	SA ₄ (O)
(O) ALE	28	28	Λ_3 (O)	(O) BALE	28	28	SA ₃ (O)
5Vdc	29	29	Λ_2 (O)	5Vdc	29	29	SA ₂ (O)
(O) OSC	30	30	Λ_1 (O)	(O) OSC	30	30	SA ₁ (O)
GND	31	31	Λ_0 (O)	GND	31	31	SA ₀ (O)

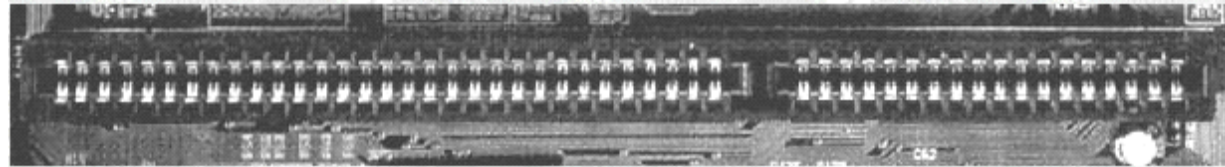
(a) PC/XT总线

P236 图5.2

(b) ISA 总线 和 XT兼容部分

EISA 总线 针对 386、486

Extended ISA (Industry Standard Architecture)—扩展工业标准体系结构
总线时钟频率**8.33MHz**，总线宽度**32位**，寻址能力**4GB**，
支持突发传送——带宽约**33MBps**。

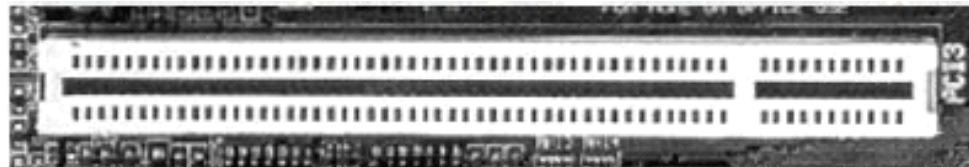


$2 \times 31 \times 2$

$2 \times 18 \times 2$

PCI总线 (**Peripheral Component Interconnect**) —外设部件互连标准

数据、地址总线**32位**，
可扩充到**64位**。



可进行突发(**burst**)式传输(成组传送: 地址 数据 数据 ...).

总线时钟频率**33MHZ**或**66MHZ**，
最高传输率可达**528MB/S**。

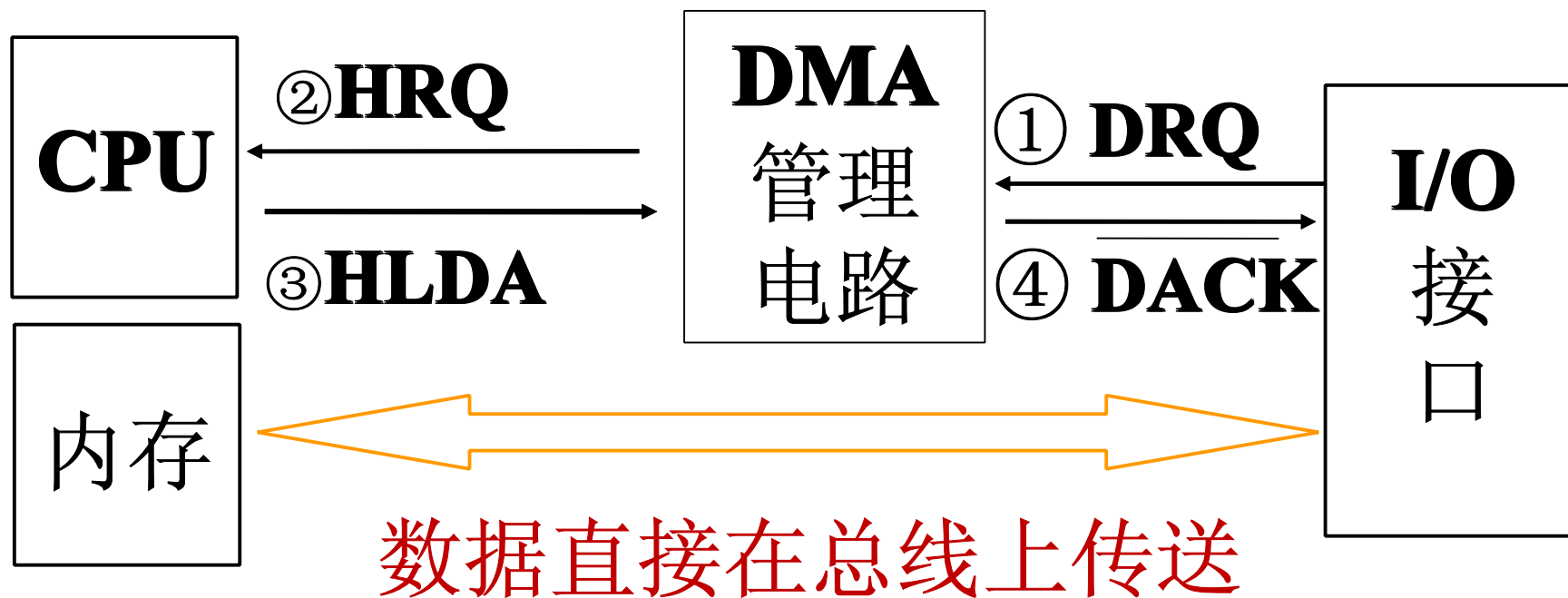
PCI卡内有设备信息寄存器组为系统提供卡的信息，
可实现即插即用 (**Plug & Play**) 。

1、 PC / XT系统总线标准

四方面信息： 引脚排列、 信号名称、
有效极性、 传送方向。

<u>IOR</u>	IO读	}	CPU 执行读写指令时 只能 4 选 1 有效。 因为只有一套地址总线
<u>IOW</u>	IO写		
<u>MEMR</u>	存储器读		
<u>MEMW</u>	存储器写		

DMA传送时读写信号两两同时有效



读内存 ($\overline{\text{MEMR}}$) 写接口 ($\overline{\text{IOW}}$) 同时

或者读接口 ($\overline{\text{IOR}}$) 写内存 ($\overline{\text{MEMW}}$) 同时

内存用地址信号选中

IO接口不用地址信号而用**DACK**信号锁定

AEN: DMA地址有效

$AEN=1$, DMA地址有效

$AEN=0$, CPU地址有效,

CPU执行指令时, $AEN=0$ ($\overline{AEN}=1$)

I/O片选译码电路仅在CPU执行IN/OUT指令时工作, 故加上 $AEN=0$ ($\overline{AEN}=1$) 选通。

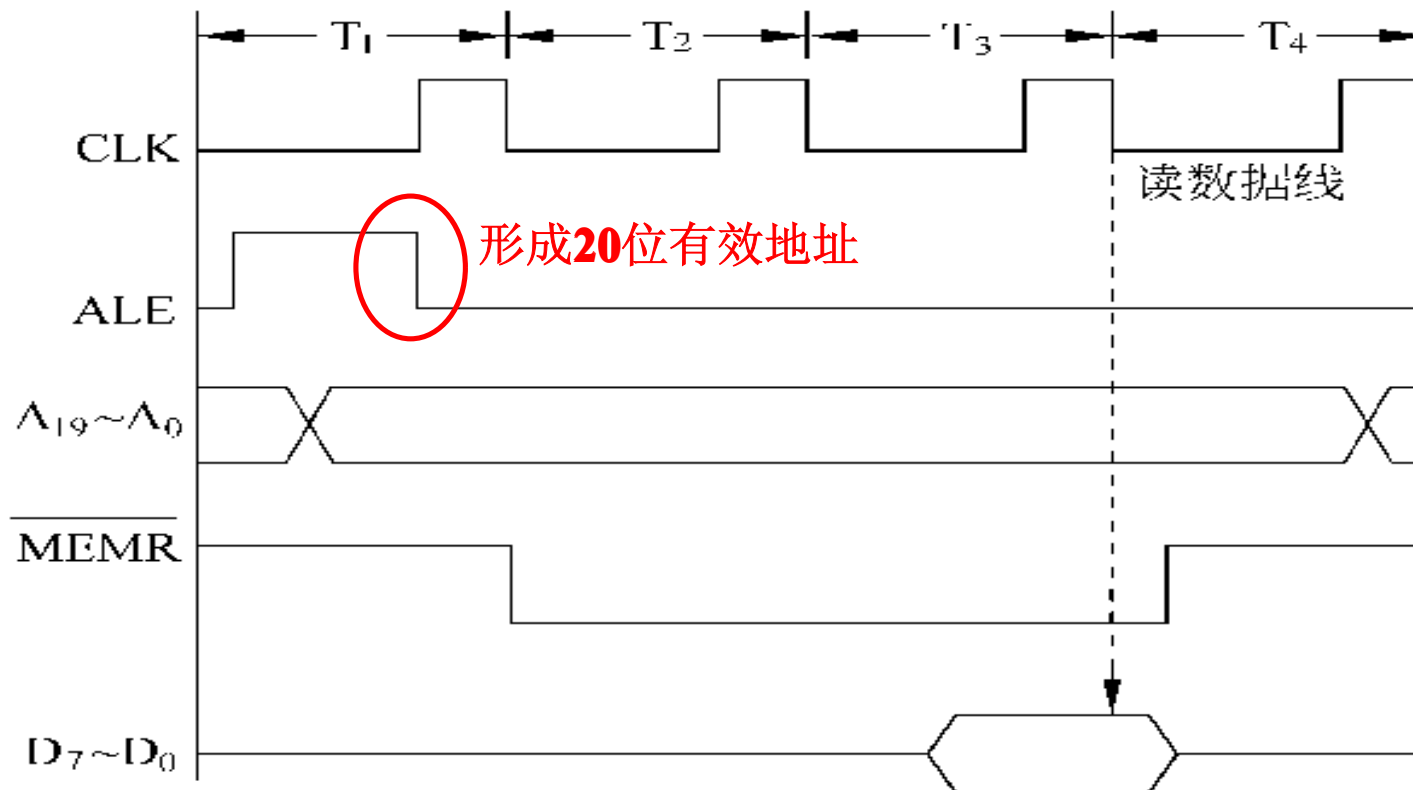
在DMA传送时I/O接口用DACK信号锁定,

I/O译码电路不工作, 地址总线信号用来选中直接访问的存储器。

2. 总线周期

指令执行过程中各总线信号的变化时序。

①CPU驱动的总线周期

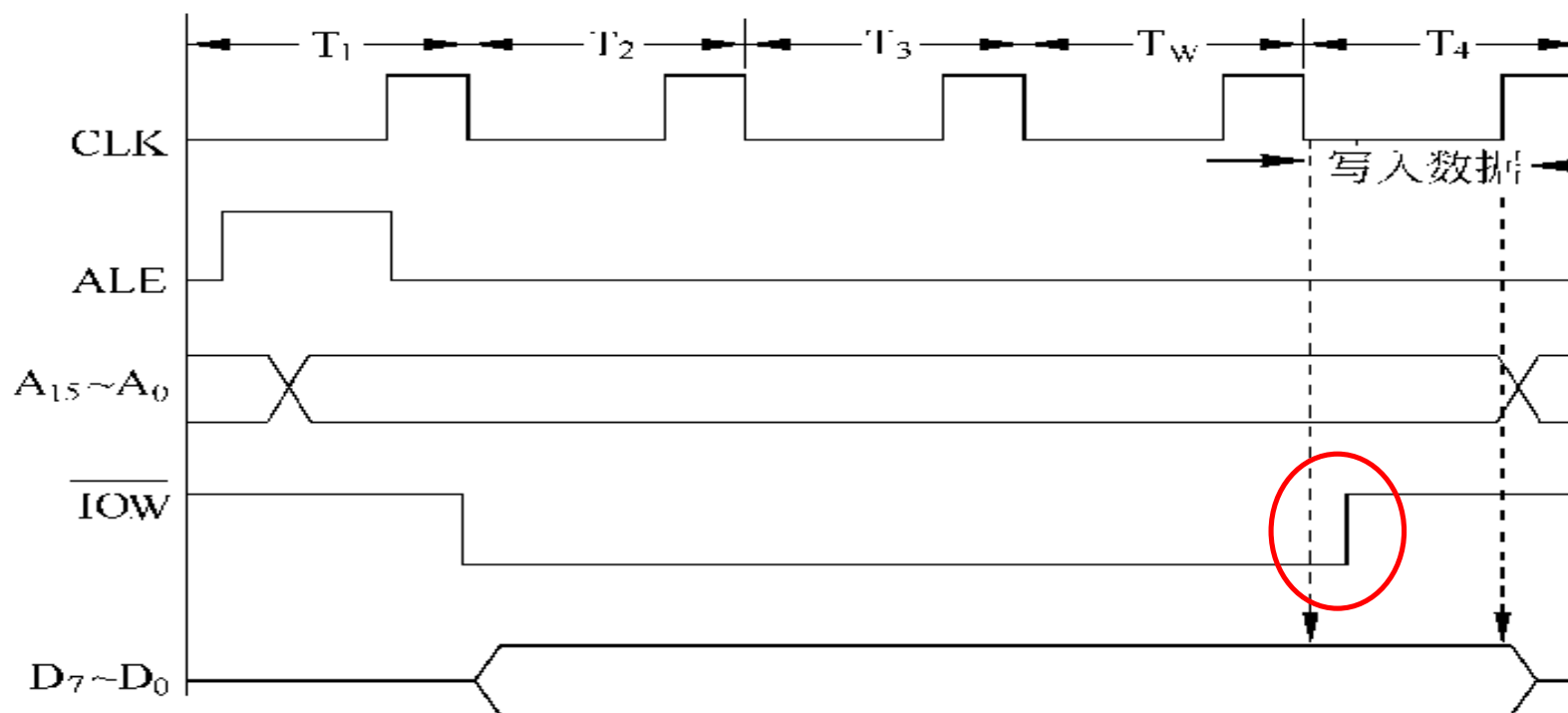


P238 图5.3 存储器读
P238 图5.3 存储器读

2. 总线周期

指令执行过程中各总线信号的变化时序。

①CPU驱动的总线周期



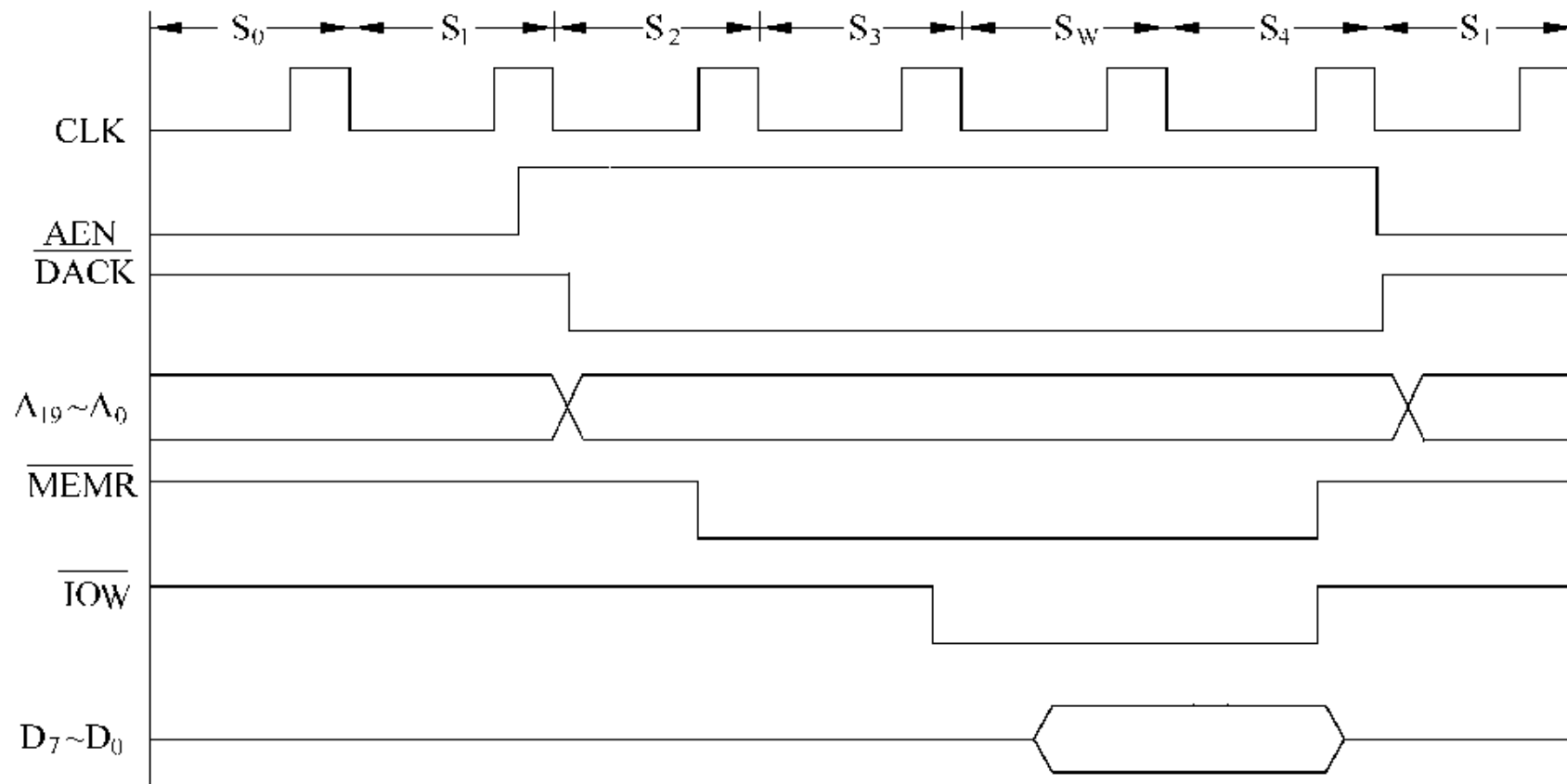
P240 图5.6 I/O 写

真正的读/写在读/写信号的后沿附近完成

2. 总线周期

② DMAC驱动的总线周期

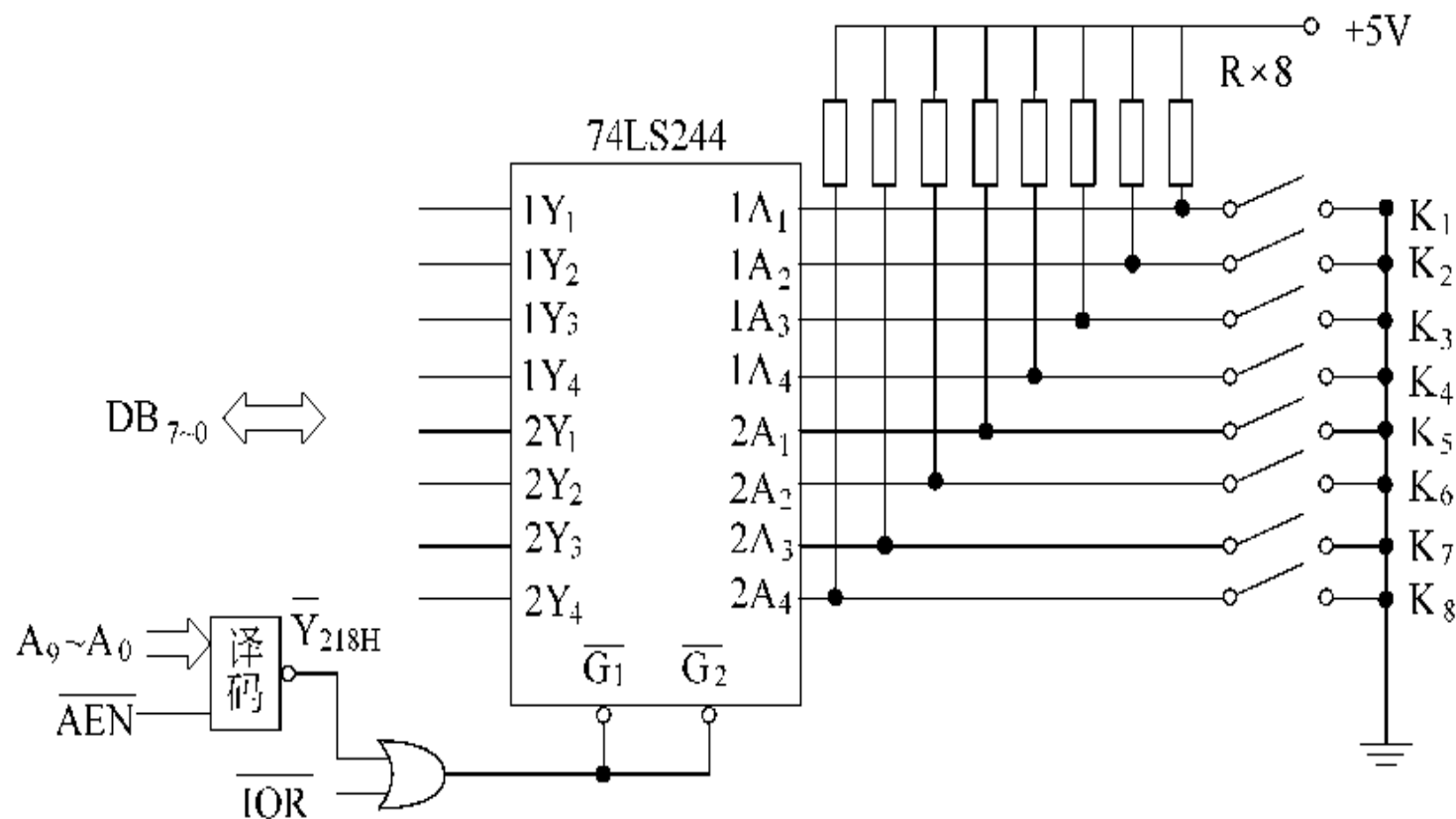
读存储器同时写I/O端口地址总线上是存储器地址



P241图5.7 DMA存储器读

3、简单的I/O接口

①三态缓冲器作输入接口



P247 图5.10 三态缓冲器用于输入接口

I/O指令中数据的宽度：指通过累加器所传输的数据的位数与指令中的累加器（**AL,AX,EAX**）有关，与端口地址无关

例如：

IN	AL, DX	； 输入 8 位数据
IN	AX, DX	； 输入 16 位数据
IN	EAX, DX	； 输入 32 位数据

IOR和**IOW**是**CPU**对**I/O**设备进行读/写的硬件上的控制信号，低电平有效。该信号为低，表示对外设进行读/写；为高，则不读/写。

在程序中：

执行**IN**指令，使**IOR**信号有效，完成读（输入）操作；
执行**OUT**指令，使**IOW**信号有效，完成写（输出）操作。

I/O端口地址译码

CPU通过**I/O**地址译码电路把来自地址总线上的地址代码翻译成为所要访问的端口

1. **I/O**地址译码的译码方法：全译码、部分译码和开关式译码

(1) 全译码方法

所有**I/O**地址线（**A0**~**A9**）全部作为译码电路的输入参加译码。一般在要求产生单个端口时采用。

(2) 部分译码方法

只有高位地址线参加译码，产生片选信号**CS**，而低位地址线不经过译码电路，直接引入接口芯片，作为寻址接口内的端口之用。一般在具有多个接口芯片的系统中采用。

IO地址译码方法（续）

（3）地址开关的译码方法

在部分译码方法的基础上，加上地址开关来改变端口地址。一般在要求**I/O**端口地址可以改变时采用。

因为地址开关不能直接接到系统地址线上，而必须通过某中介元件将地址开关的状态（**ON/OFF**）转移到地址线上去。

能够实现这种中介转移作用的有比较器、异或门等。

■ I/O地址译码方法（续）

2. 16位微机的I/O地址译码方法（部分译码方法）

把**10位I/O地址线**分为两部分：

- 1) 高位地址线，经译码电路产生**I/O接口芯片**的片选**CS**信号，实现系统中的片间寻址；
- 2) 低位地址线不参加译码，直接连到**I/O接口芯片**，进行**I/O接口芯片**的片内端口寻址，即寄存器寻址。

◆ 低位地址线的根数取决于接口中寄存器的个数：

并行接口芯片**82C55A**内部有**4**个寄存器，需要**2**根低位地址线

串行接口芯片**8251A**内部有**2**个寄存器，只需**1**根低位地址线

◆ 若从系统的角度来考虑，则低位地址线的根数应由系统中含有寄存器数目最多的接口芯片来决定。

■ I/O地址译码方法（续）

3. 32位微机的I/O地址译码方法

32位微处理器的**I/O**地址线仍然是**16位**

使用字节允许线**BE0~BE3**（**A0**和**A1**经过内部译码产生），来指定双字哪一个字节被选中。

- ◆ 采用部分译码方法，把高**14位I/O**地址线分为两部分：
 - 一部分是高位地址线，经译码电路产生**I/O**设备选中信号，实现设备寻址；
 - 另一部分是低位地址线，直接连到**I/O**设备接口芯片，进行**I/O**接口芯片内部寄存器寻址。

IO地址译码电路

1. I/O地址译码电路类型

固定式译码电路和开关式（可选）译码电路。

◆ 固定式端口地址译码

接口中用到的端口地址不能更改

缺点：不灵活，一旦地址与其他设备的地址发生冲突，就不能工作。一般由**IC**芯片或译码器组成

固定式端口地址译码

{ 单个端口地址译码
多个端口地址译码

IO地址译码电路（续）

◆ 开关式译码电路

I/O地址可以更改：若发生地址冲突，则可以通过地址开关来改变译码电路的I/O地址，避免冲突，设备仍能工作。

组成：由地址开关、译码器和比较器组成。如中小型IC器件、GAL/PAL器件和CPLD器件等。

IO地址译码电路（续）

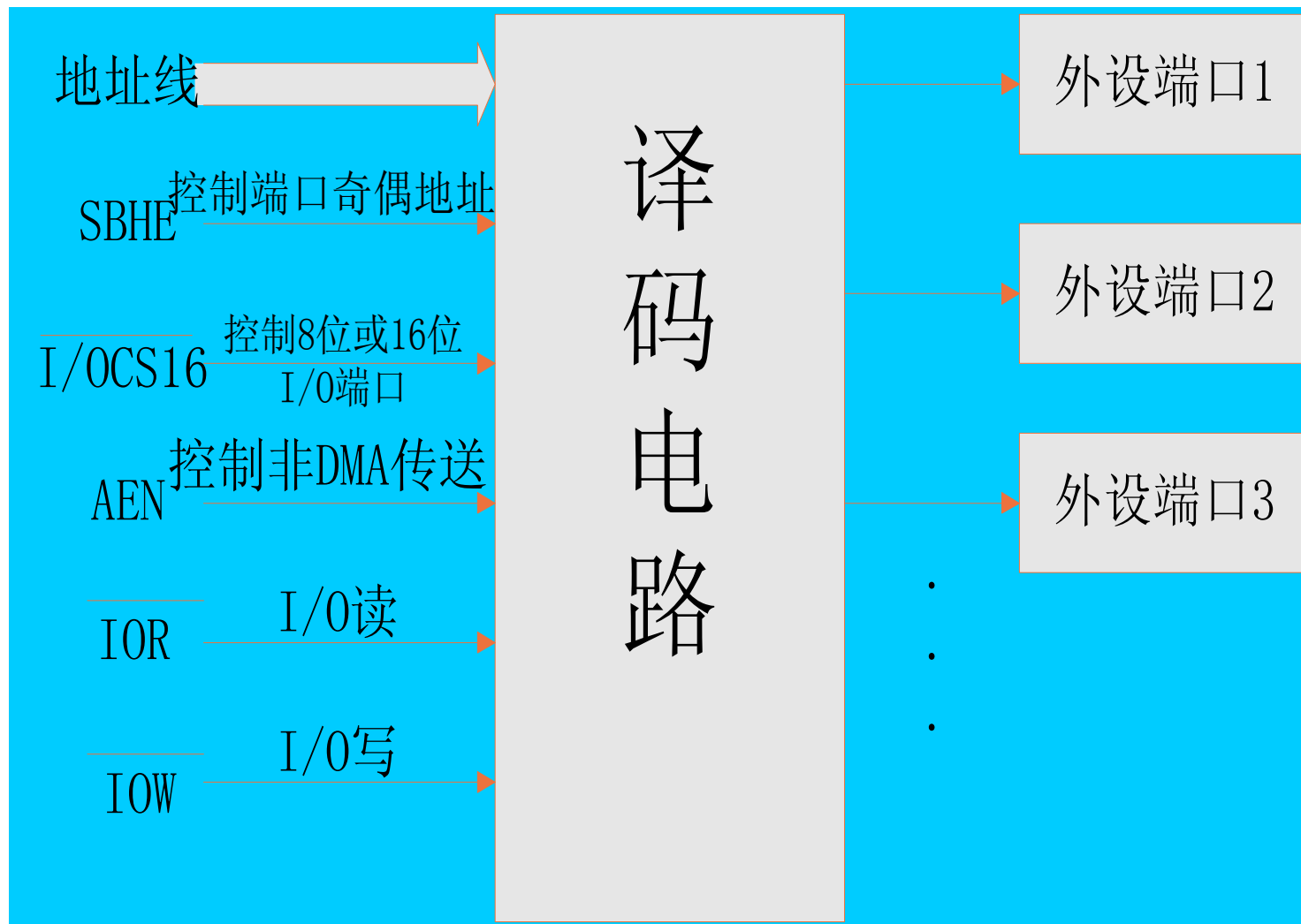
2. I/O地址译码电路的输入信号：

不仅仅与地址信号有关，而且与控制信号有关

I/O端口地址译码电路的作用：

把地址和控制信号进行逻辑组合，以产生对接口芯片的选择信号“0”

16位微机系统



IO地址译码电路（续）

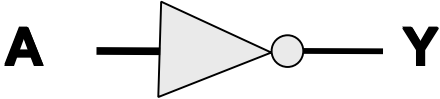

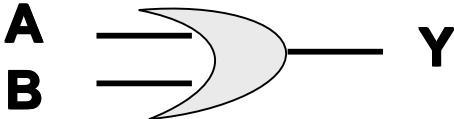
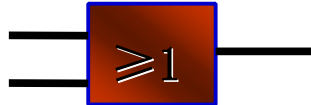
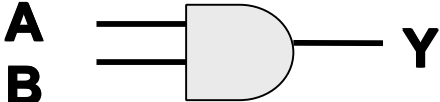

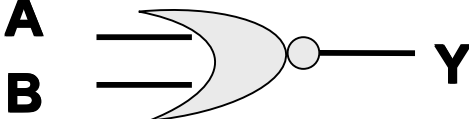
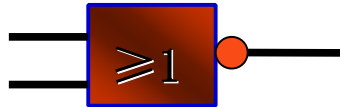
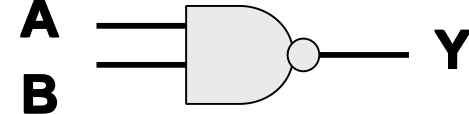
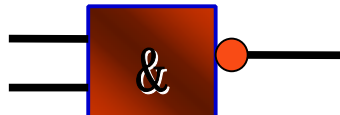
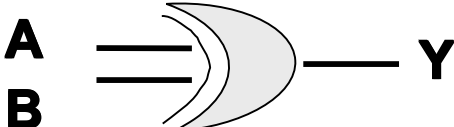
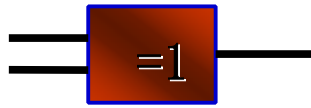
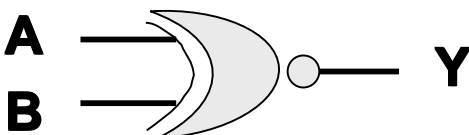
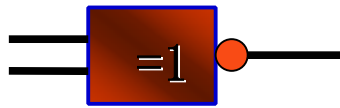
- 使用 $\overline{\text{M/IO}}=0$ 与 $\overline{\text{W/R}}$ 来激活I/O设备。
- 使用 $\overline{\text{IOR}}$ 和 $\overline{\text{IOW}}$ 来激活I/O设备执行一次读/写。

3. I/O地址译码电路的输出信号

是一根低电平输出线 \overline{CS}

- 当 \overline{CS} 有效,这根输出线所驱动的一个接口芯片被选中
- 该接口芯片内部的数据线与系统总线相连
- 其它未选中的外设接口芯片呈高阻状态
- **CPU**的数据就在被选中的外设之间传送

补充:

名称	老符号	新符号	表达式
非门			$Y=\bar{A}$
或门			$Y=A+B$
与门			$Y=A\cdot B$
或非门			$Y=\overline{A+B}$
与非门			$Y=\overline{A\cdot B}$
异或门			$Y=A\oplus B$
异或非门			$Y=\overline{A\oplus B}$

与门： 只要有一个输入为**0**，输出就为**0**

与非门： 只要有一个输入为**0**，输出就为**1**

或门： 只要有一个输入为**1**，输出就为**1**，
输入全为**0**，输出才为**0**。

IO地址译码电路设计的几个问题

1. 遵循I/O端口地址的选用原则

- ① 凡是被系统配置所占用了的地址一律不能使用。
- ② 原则上讲，未被占用的地址用户可以使用，但对计算机厂家申明保留的地址，不要使用，否则，会发生I/O端口地址重叠和冲突造成所开发的产品与系统不兼容而失去使用价值。
- ③ 一般用户可使用300H~31FH地址(留作实验卡用的)，但是最好采用地址开关(由于每个用户都可以使用，所以在用户可用的这段I/O地址范围内，为了避免与其他用户开发的插板发生地址冲突)

IO地址译码电路设计的几个问题

2. 正确选用地址译码方法

一般是单端口地址译码采用全译码法；多个接口地址译码采用部分译码法。

3. 灵活设计I/O地址译码电路

电路的组成，可以采用不同的元器件。

参加译码的地址信号和控制信号之间的逻辑组合可以不同

IC器件组成的IO地址译码电路

I/O译码电路设计包括：

- 采用不同元器件（IC电路、译码器、GAL器件）（21ic.com）
- 不同译码方法（全译码方法和部分译码方法）
- 不同电路类型（固定式、开关式）

例：单个端口地址译码电路设计

（1）要求：设计一个端口地址为2FBH的译码电路

(2) 分析

- 单个端口地址的译码电路，故采用全译码方法。
- **10**根地址线全部作为译码电路的输入线，参加译码
- 端口地址是**2FBH**可得**10**位输入地址线的取值：

地址线	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
2FBH	1	0	1	1	1	1	1	0	1	1

控制线：
 $\overline{AEN}=0$ 非**DMA**操作
 $\overline{IOR}=0$ 读
 $\overline{IOW}=0$ 写

在每条地址线的状态为上述值时，地址译码电路输出为**0**

(3) 设计

能够实现上述地址线取值的译码电路有很多种，一般采用门电路就可以实现，而且很方便。常用门电路有：与门、或门、非门及与非、或非等。

74LS20 双4输入与非门

74LS30 8输入与非门

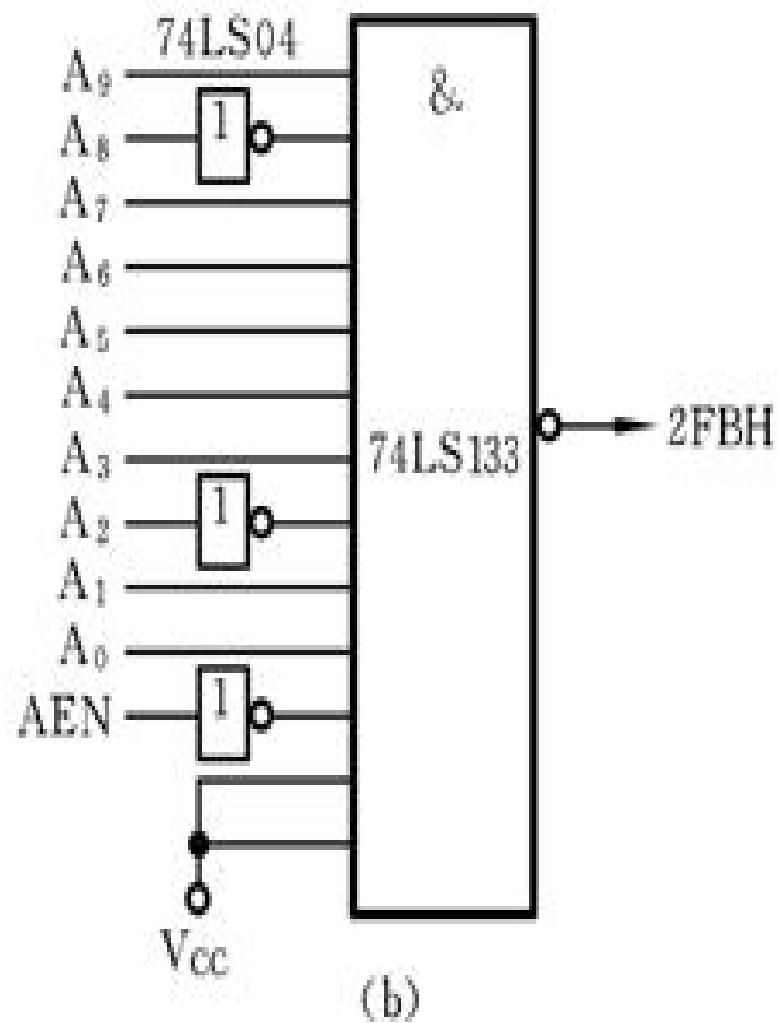
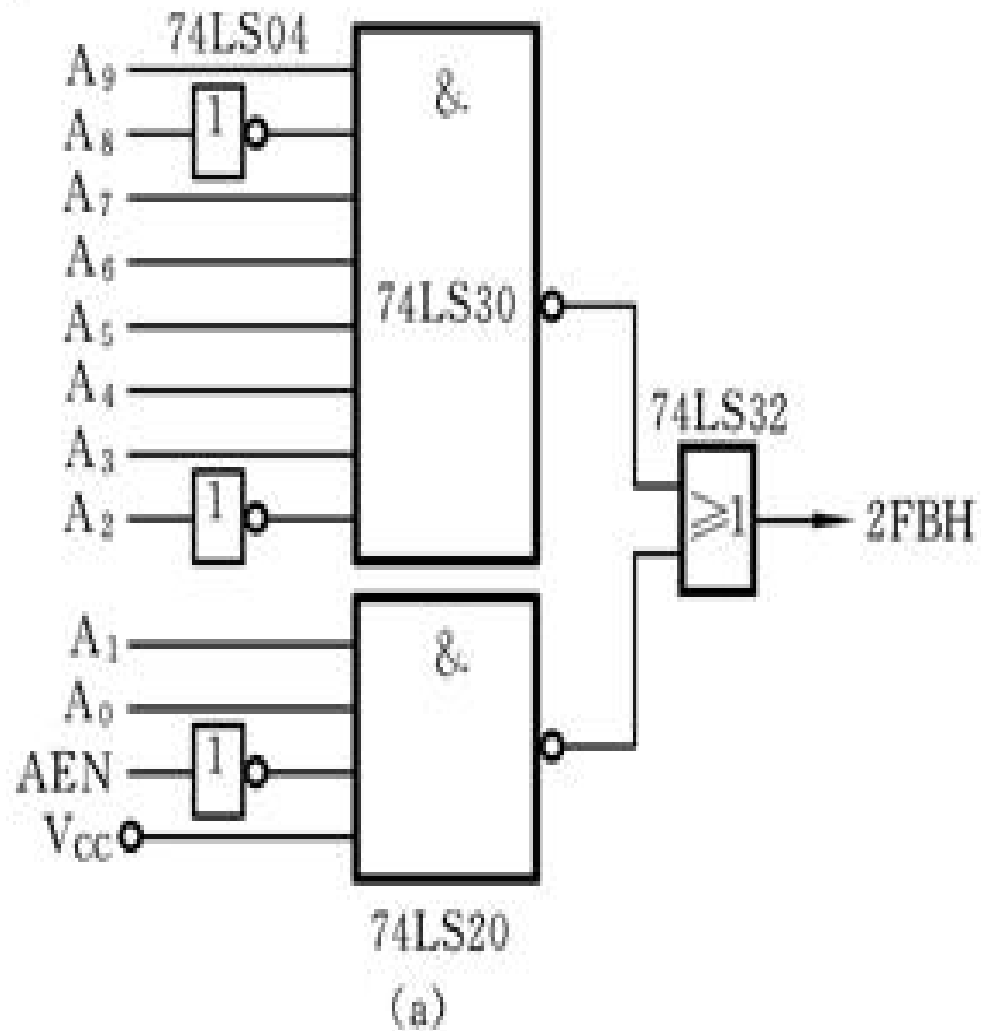
74LS32 四2输入或门

74LS04 六非门

74LS133是13输入端与非门

A9A8A7A6A5A4A3A2A1A0
2FBH 1 0 1 1 1 1 1 0 1 1

1



固定式单个端口地址译码电路

设 I/O 地址为 218H 时译码器输出选中此接口，

执行下列指令，可以读入 8 个开关状态：

```
MOV    DX, 218H
```

```
IN     AL, DX
```

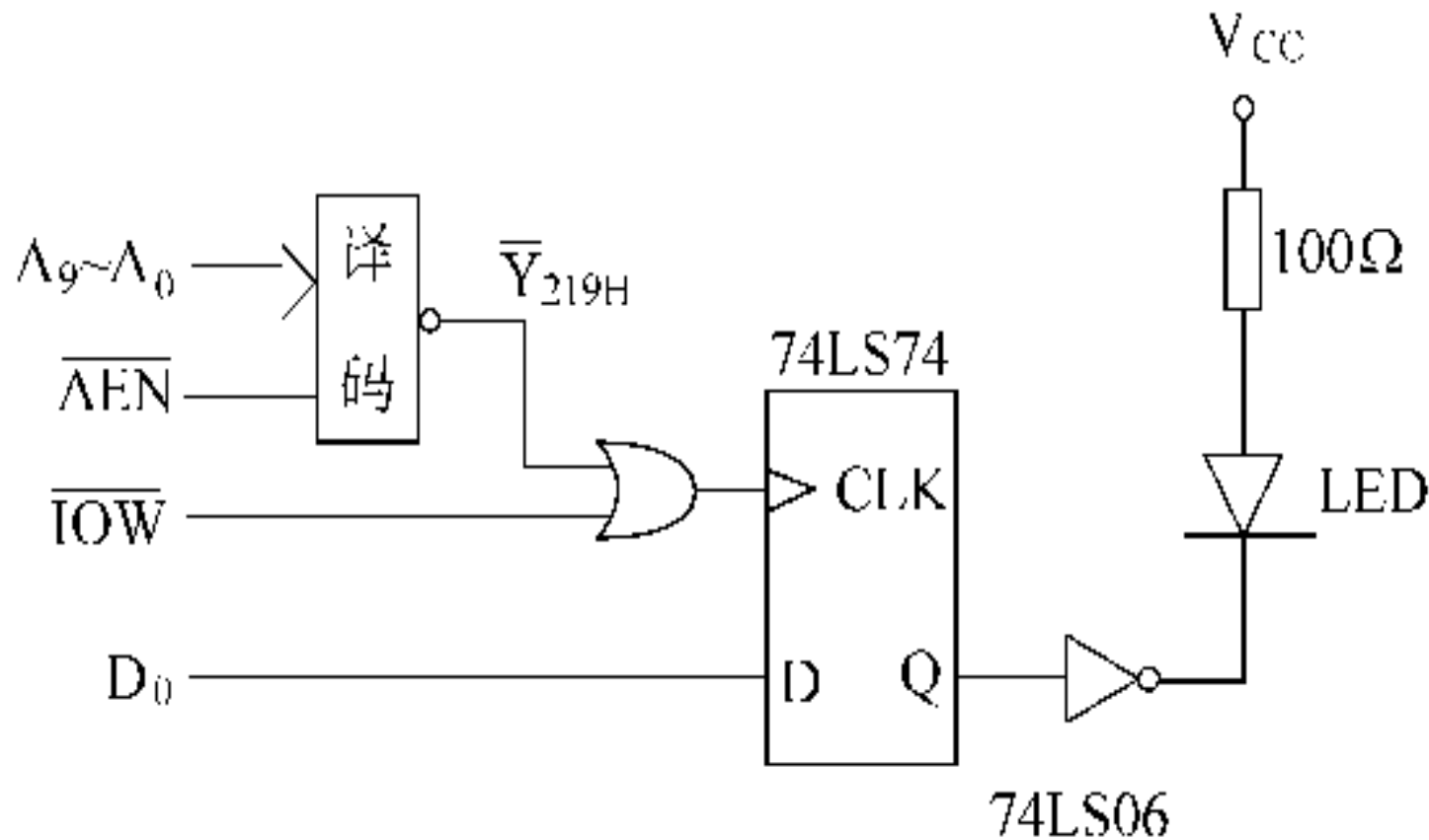
只有数据端口。
执行 **IN** 指令时开关正在变化如何保证输入的数据正确？

软件：重复多次 **IN** 的内容相等时（稳定）才接收。

硬件：增加状态端口，先查询状态稳定才输入数据。

3、简单的I/O接口

②锁存器作输出接口



P247 图5.11 D触发器用于输出接口

设I/O地址为219H时译码器输出选中接口
执行下列指令，AL的D₀位写入触发器，驱动LED：

```
MOV    DX, 219H
```

```
OUT    DX, AL
```

若连续输出如何保证数据都被可靠接收？

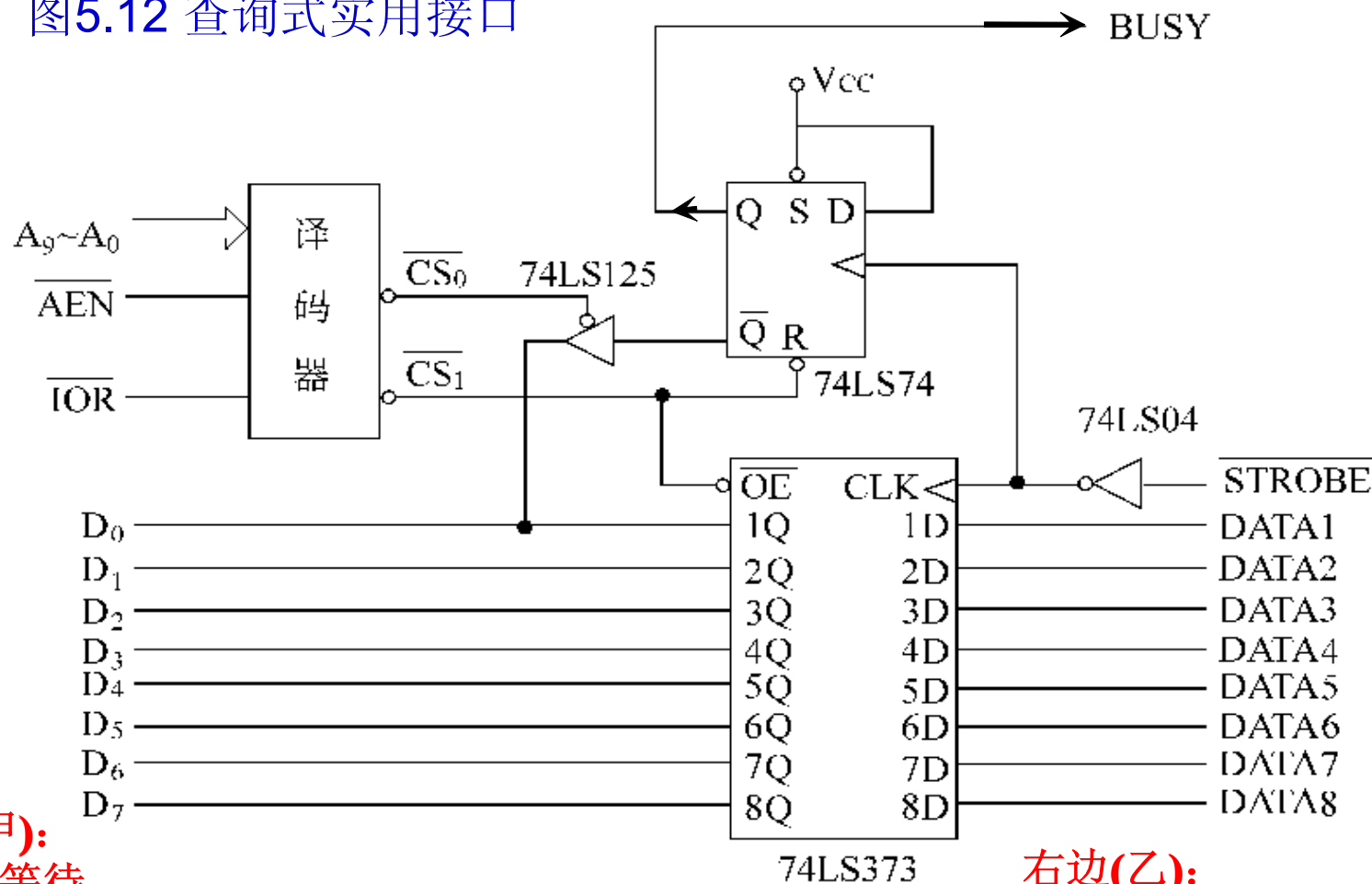
延时

增加状态接口

3、简单的I/O接口

③简单接口举例

P248 图5.12 查询式实用接口



左边(甲):

$D_0 = 1$ 等待

$D_0 = 0$ 接收数据
并使 **BUSY=0**

右边(乙):

BUSY = 1 等待

BUSY = 0 送出数据锁存到74LS373
并使 **BUSY = 1**

左边有主机，右边有外设

8位锁存缓冲器74LS373是数据端口，地址为 $\overline{CS_1}$

D触发器是状态端口， $Q=1$ 为准备好，地址为 $\overline{CS_0}$ ，状态位送到数据总线 D_0

状态位的清零也用 $\overline{CS_1}$ ，读数的同时清除状态位

计算机甲（左边）的查询方式输入

```
mov    DX, CS1  
in     AL, DX      ; 假读，清忙信号
```

```
mov    CX, DelayTime
```

```
Chks:  mov    DX, CS0  
       in     AL, DX  
       and    AL, 1  
       jz     Datin  
       loop   Chks
```

...

```
Datin: mov    DX, CS1  
       in     AL, DX      ; 输入数据
```

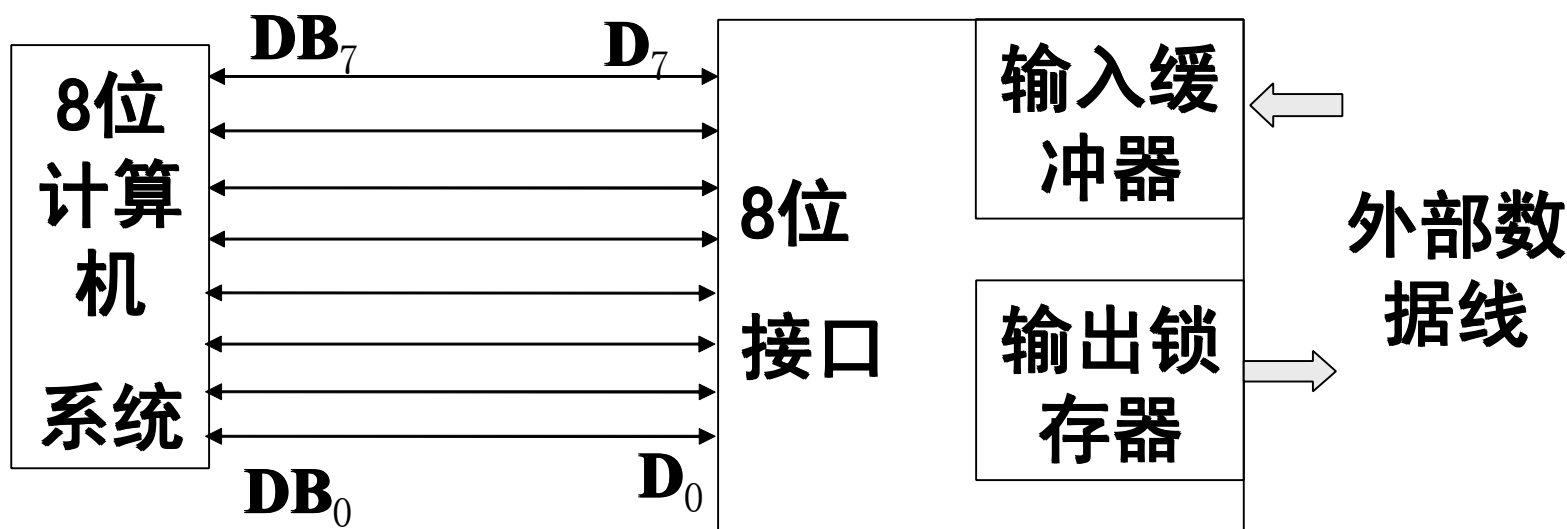
4 . 总线信号与接口的连接

1.数据信号的连接

系统的数据总线是双向三态，若接口的数据总线也是双向三态则可以直接相接：

输出数据一般要求有锁存，

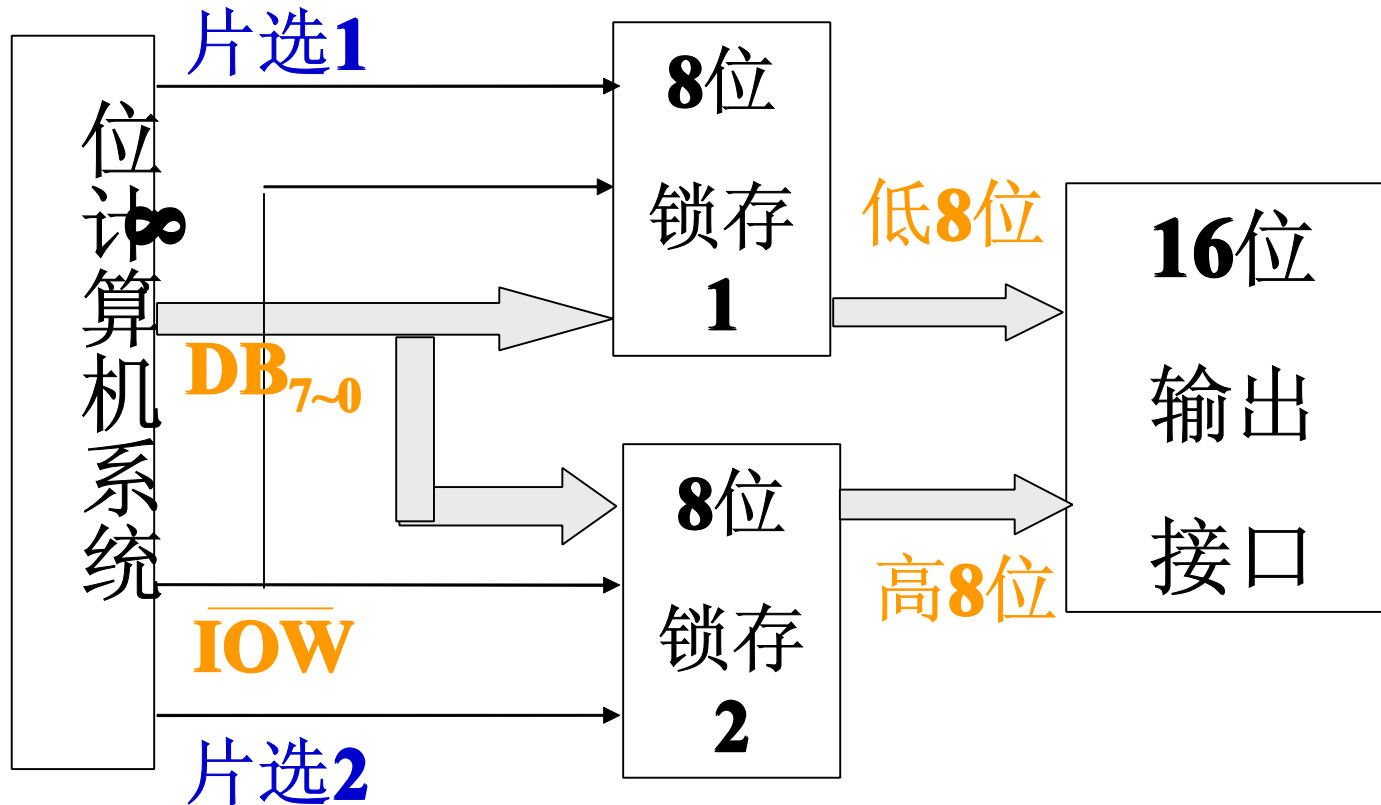
输入数据一般要求有三态缓冲。



8位系统对8位接口：按数据线位序对应相接 № 61

8位系统对16位接口：

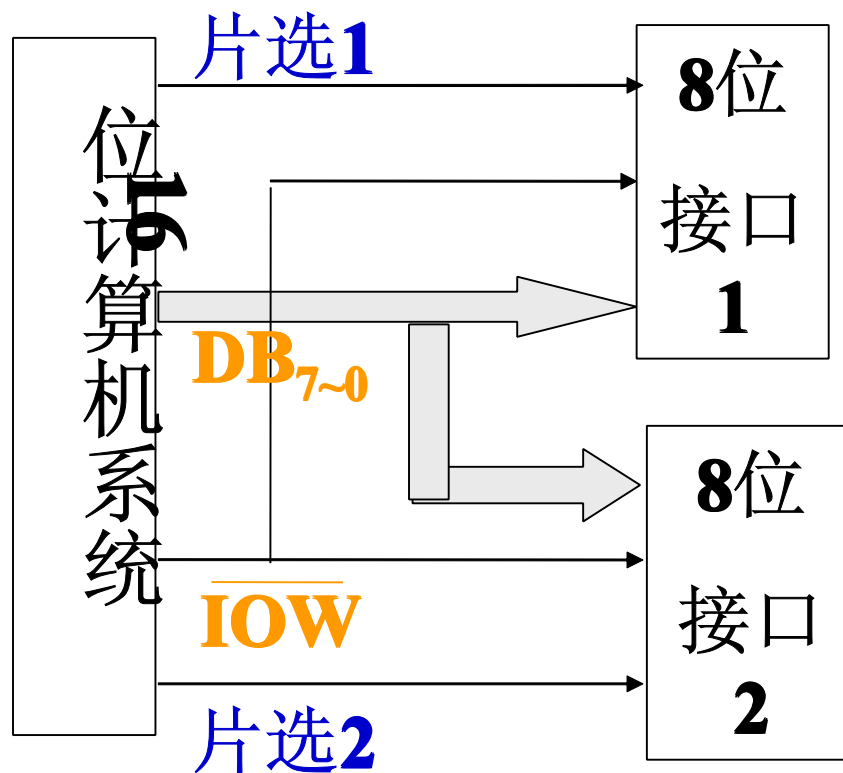
按高低字节分成两个8位端口分别进行I/O。



更多位接口用更多个8位端口

输入接口：数据传送方向相反。锁存器改成三态缓冲器，**IOW**改成**IOR**。

16位系统接8位接口：一般接数据总线低8位



建议**8位**接口地址都用偶地址，地址对准（**aligned**）能在一个总线周期完成**I/O**传送。

2.控制信号的连接

所有的**I/O**片选译码都要加**AEN**信号控制。

输入接口(三态缓冲器)用 **$\overline{\text{IOR}}$** 信号选通，

输入数据在 **$\overline{\text{IOR}}$** 信号的后沿附近采样。

输出接口(锁存器)用 **$\overline{\text{IOW}}$** 信号锁存，

输出数据应保持到 **$\overline{\text{IOW}}$** 信号的后沿以后消失

中断方式传送的接口还要有中断联络信号，

DMA方式传送的接口还有**DMA**联络信号。

3.地址信号的连接

片选地址：

取决于高位地址信号译码产生的片选信号。

会有重叠区（部分译码）

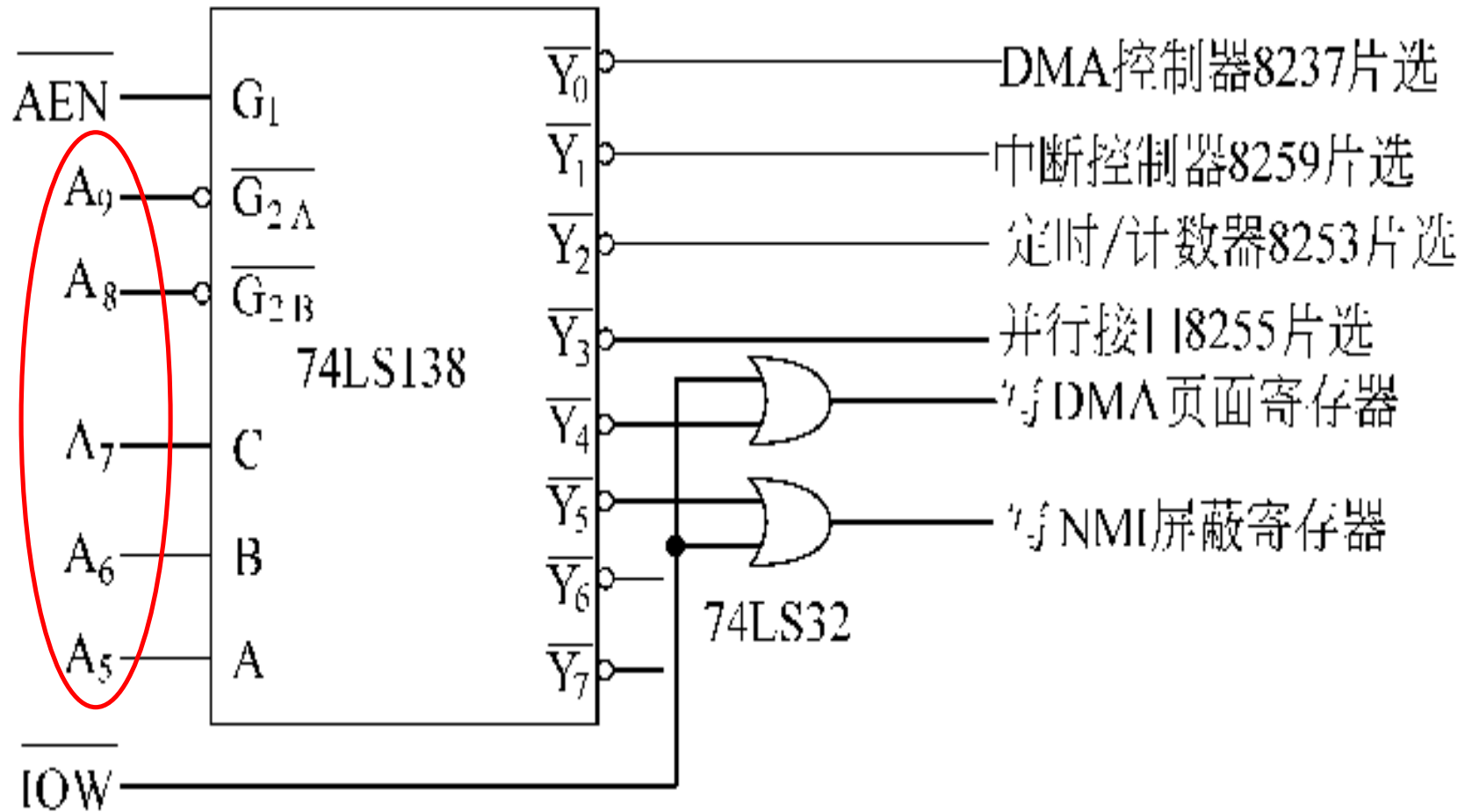
只是一个范围(低位地址信号到片上再译码)

片内地址：

接口芯片的地址引脚接收低位地址总线信号
在片内译码产生的地址。

片选地址和片内地址组合起来决定端口在系统中的物理地址（指令中使用的地址）。

IBM PC/XT 系统板 I/O 端口



00*** #####

P245图5.9

I / O 端口地址译码

I/O端口地址分配

64K IO地址映像图

16位的I/O地址线所形成的64K I/O端口地址空间整体映像：

1. 端口**0000H**和**03FFH**之间的I/O空间留给微机系统和**ISA总线**；
部分**I/O**端口分配给相应的设备作为专用的端口，因此是固定的**I/O**端口；
2. 位于**0400H**~**FFFFH**的I/O端口一般给用户应用、主板功能和**PCI**总线使用。
I/O端口由**PNP**即插即用机制和操作系统动态分配与管理

FFFF	I/O 扩展区域
0400	
03FF	COM ₁
03F8	
0357	软磁盘
03F0	
03EF	
03E0	
03DF	适配器
03D0	
03CF	
0380	
037F	LPT ₁
0378	
0377	
0330	
032F	硬磁盘
0320	
031F	
0300	
02FF	COM ₂
02F8	
02F7	
0064	
0063	8255(PPI)
0060	
005F	
0044	
0043	定时器
0040	
003F	
0024	
0023	中断控制器
0020	
001F	
0010	
000F	DMA 控制器
0000	

实模式下IO地址的分配

分两类：

1、系统板上的I/O地址：大多数是可编程的大规模集成电路，如定时/计数器，中断控制器，**DMA**控制器，并行接口等。

2、I/O扩展槽上的接口控制卡地址：由若干个集成电路按一定的逻辑组成的一个部件，如软驱卡、硬驱卡、图形卡、声卡、串行通信卡等。

PC机的I/O地址线只用了低10根A9~A0（高6位未用）
故地址范围0000H~03FFH（ $2^{10}=1024=1K$ 个端口）

实模式下IO地址的分配(续)

I/O 接口名称	端口地址
游戏控制卡	200~20FH
并行口控制卡 1	378~37FH
并行口控制卡 2	278~27FH
串行口控制卡 1	3F8~3FFH
串行口控制卡 2	2F8~2FFH
原型插件板(用户可用)	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	320~32FH
软驱控制卡	3F0~3F7H

系统机中的I / O端口地址 (P244 表5. 1)

系统机使用的I / O地址**1K** : 0~3FFH

主板上占用: 0~1FFH ($A_9=0$)

扩展槽用: 200~3FFH ($A_9=1$)

系统的I / O端口地址译码电路

(1) 是部分译码, A_{10} 以上、 A_4 以下未参加译码, 有地址重迭。

用户新加入接口地址为**400H**可否?

(2) 有AEN=0选通, 仅CPU执行指令时工作。

所有**I/O**地址译码必须用**AEN=0**选通

简单接口小结：

计算机和**I/O**设备的数据传送分成两级：

计算机 \longleftrightarrow 接口 \longleftrightarrow 外设

计算机的**I/O**是面向接口，

I/O指令的传送只做到端口为止。

为了保证数据可靠传送，计算机、接口、
外设三者之间要很好协调

计算机、接口之间的协调靠软硬件综合实现