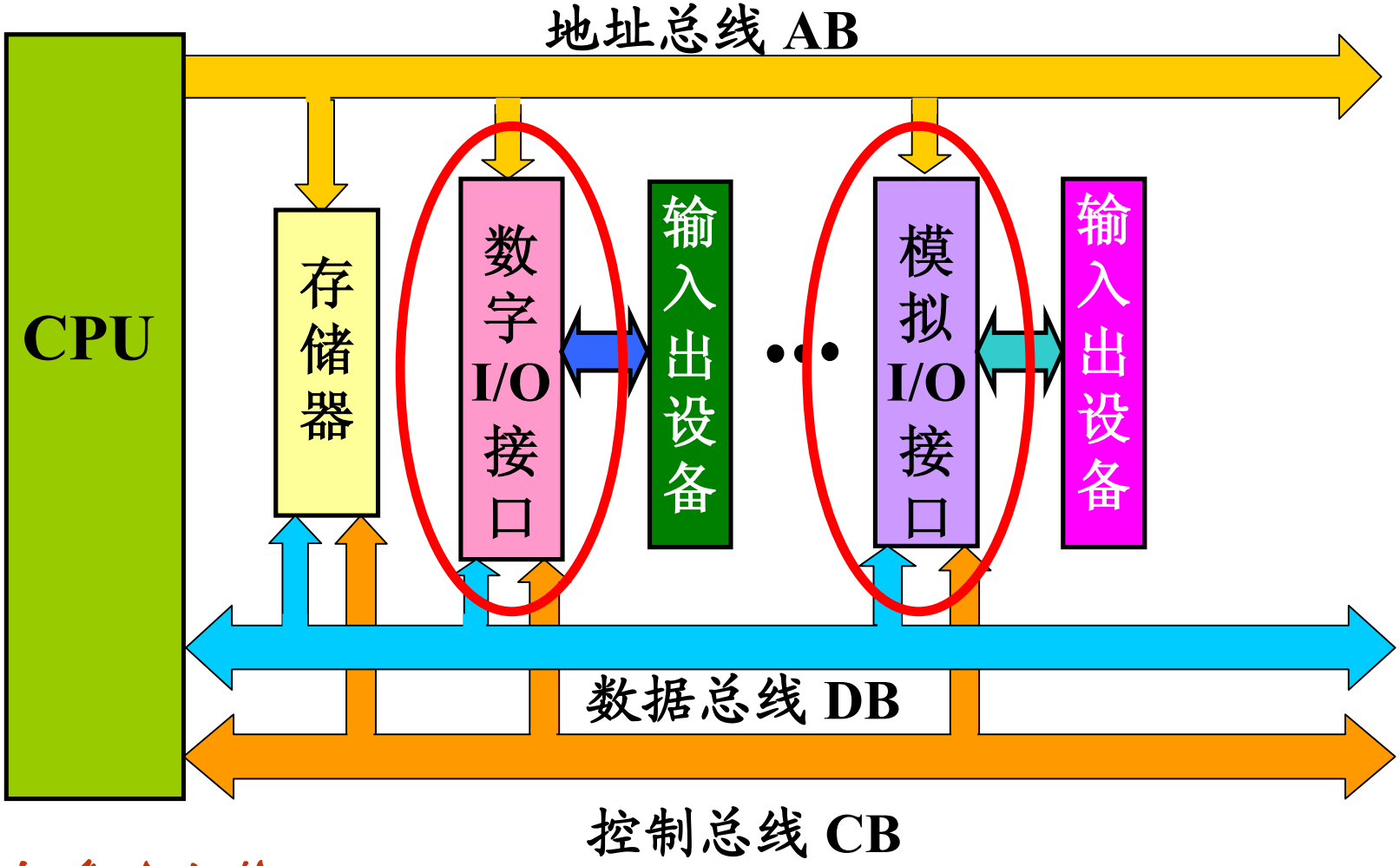
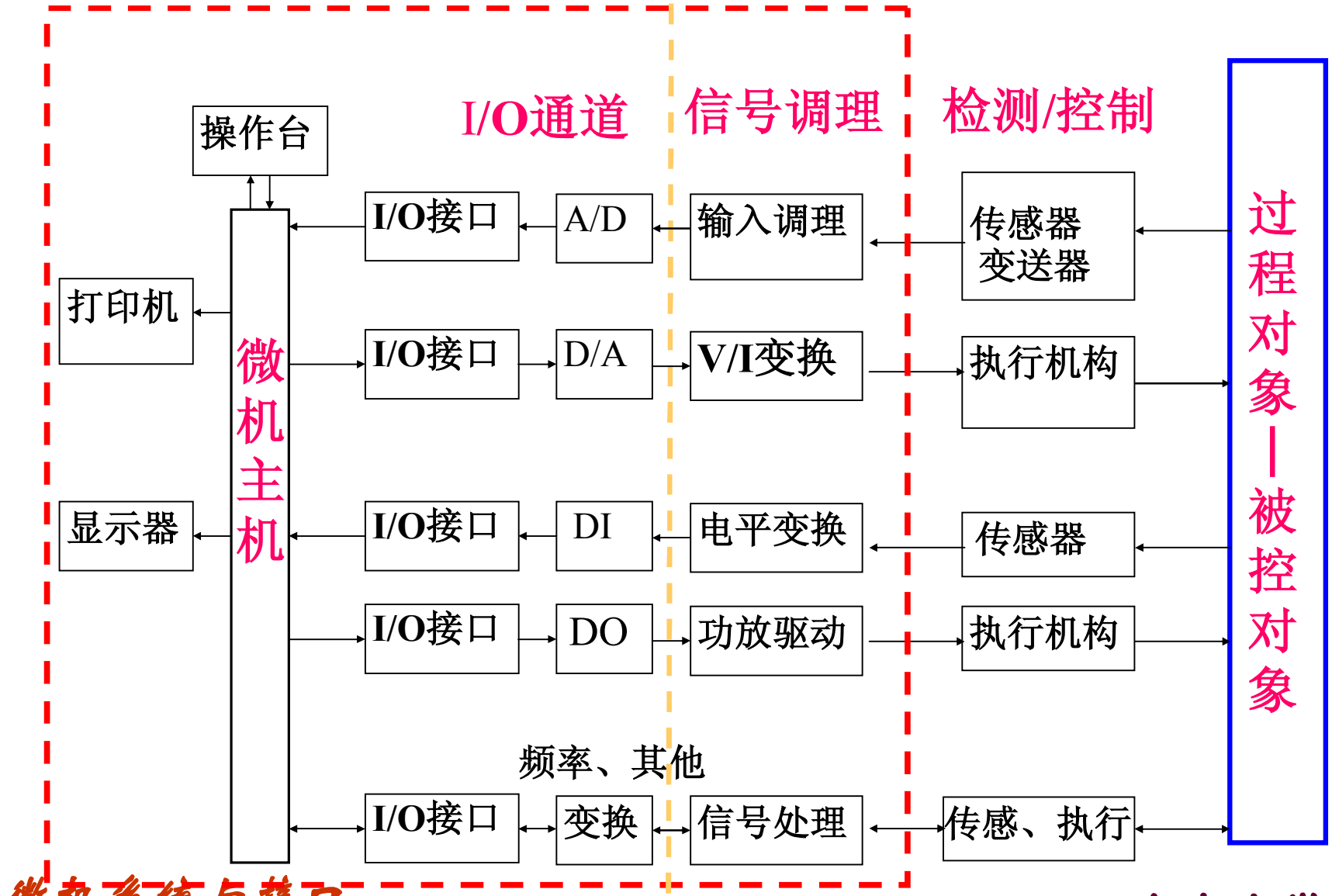
第五章　数字量输入输出接口
(Digital I/O Interface)



生产过程微机控制系统结构

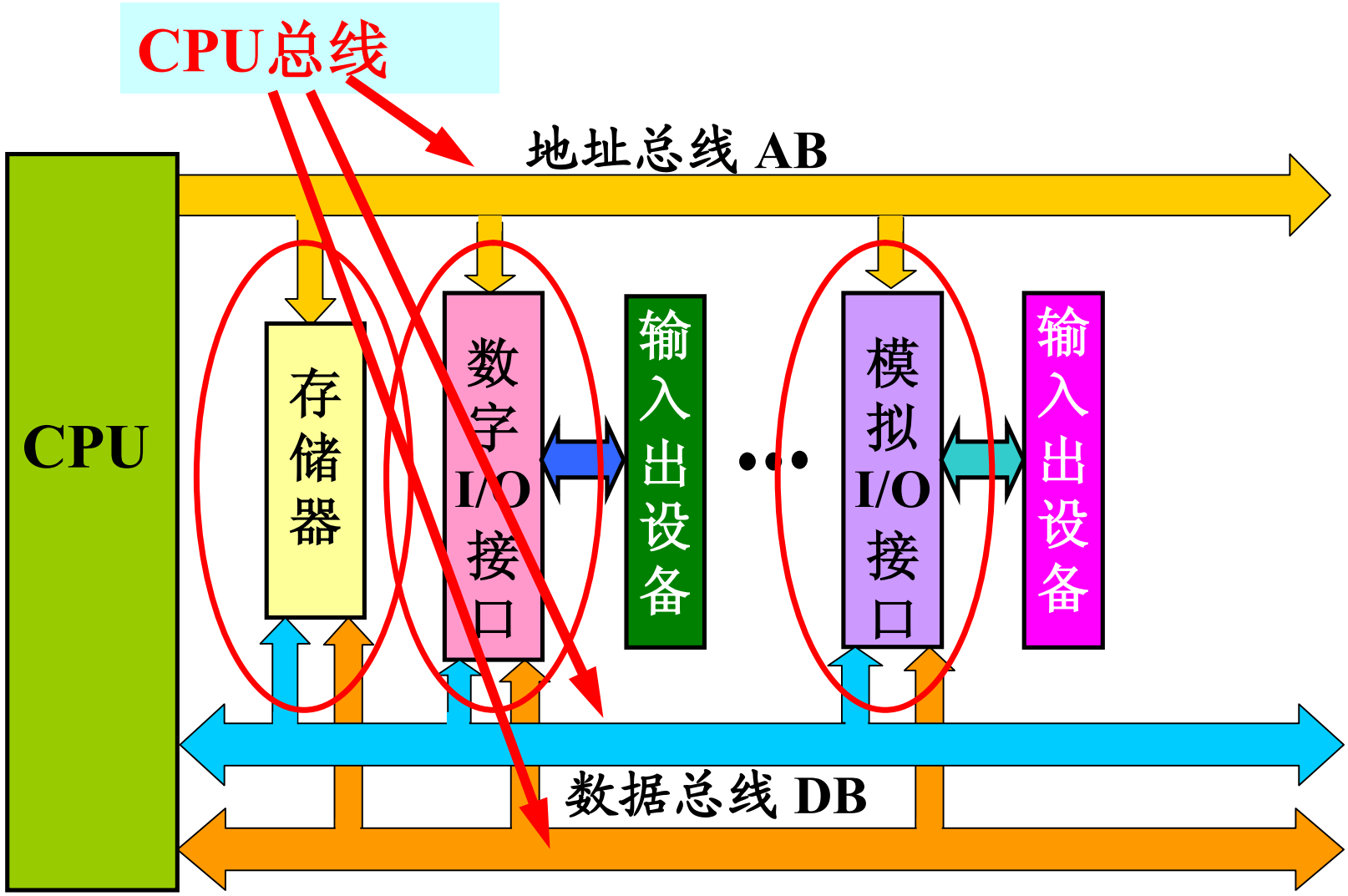


第五章　数字量输入输出接口

主　要　内　容

* 接口基本概念
* 接口电路（芯片）、端口地址
* 数据传送方式
* 总线及其接口
* 中断电路及其处理
* 定时/计数器电路与应用
* 并行接口电路与应用
* 串行接口电路与应用
* DMA电路与应用

总线基本概念


CPU总线
地址总线 AB
CPU
存储器
数字I/O接口
输入出设备
模拟I/O接口
输入出设备
数据总线 DB
控制总线 CB

总线分类

CPU 片内总线

——CPU内部，各部件之间的连接总线

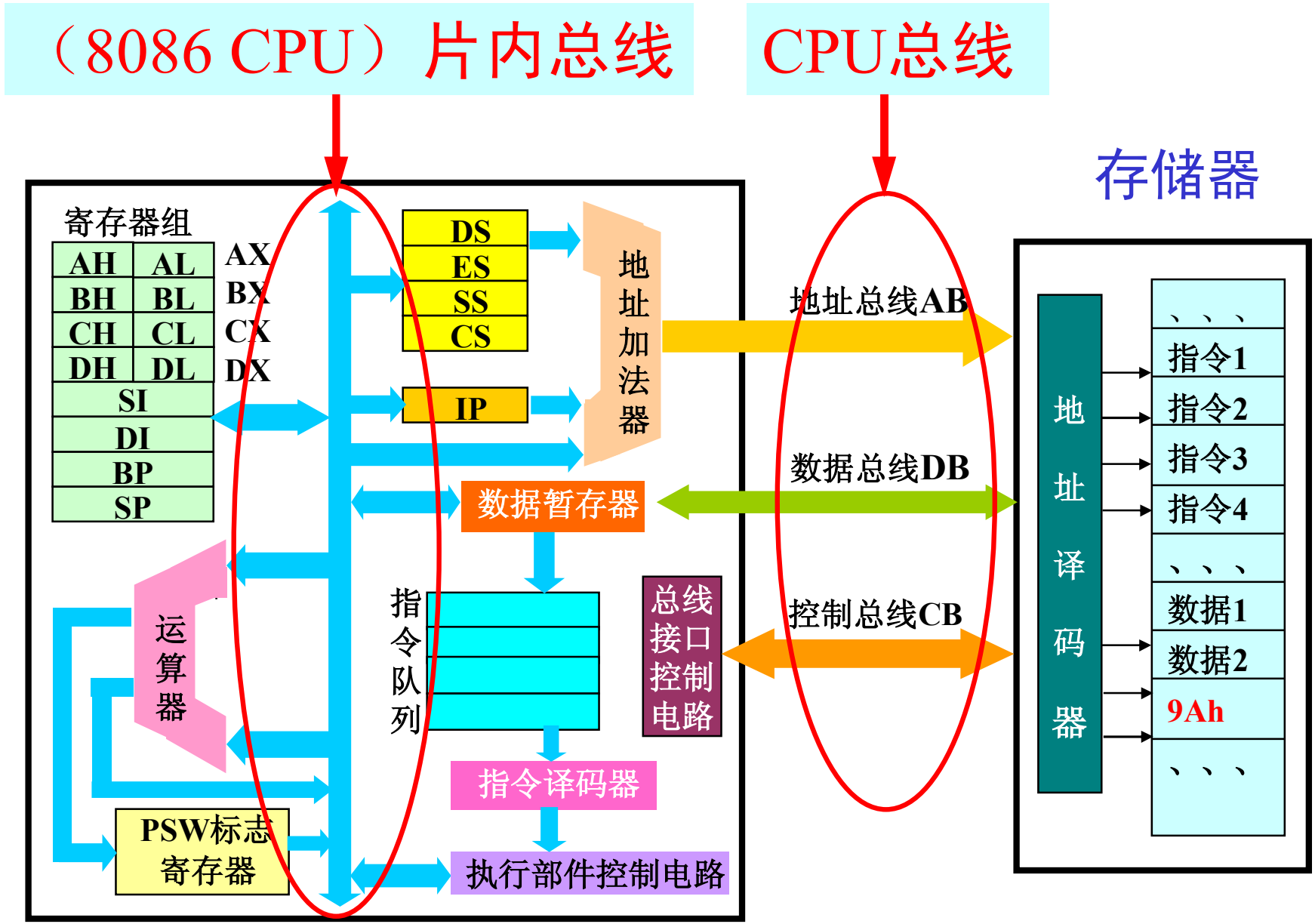
CPU总线

——计算机内部， CPU与各芯片的连接总线

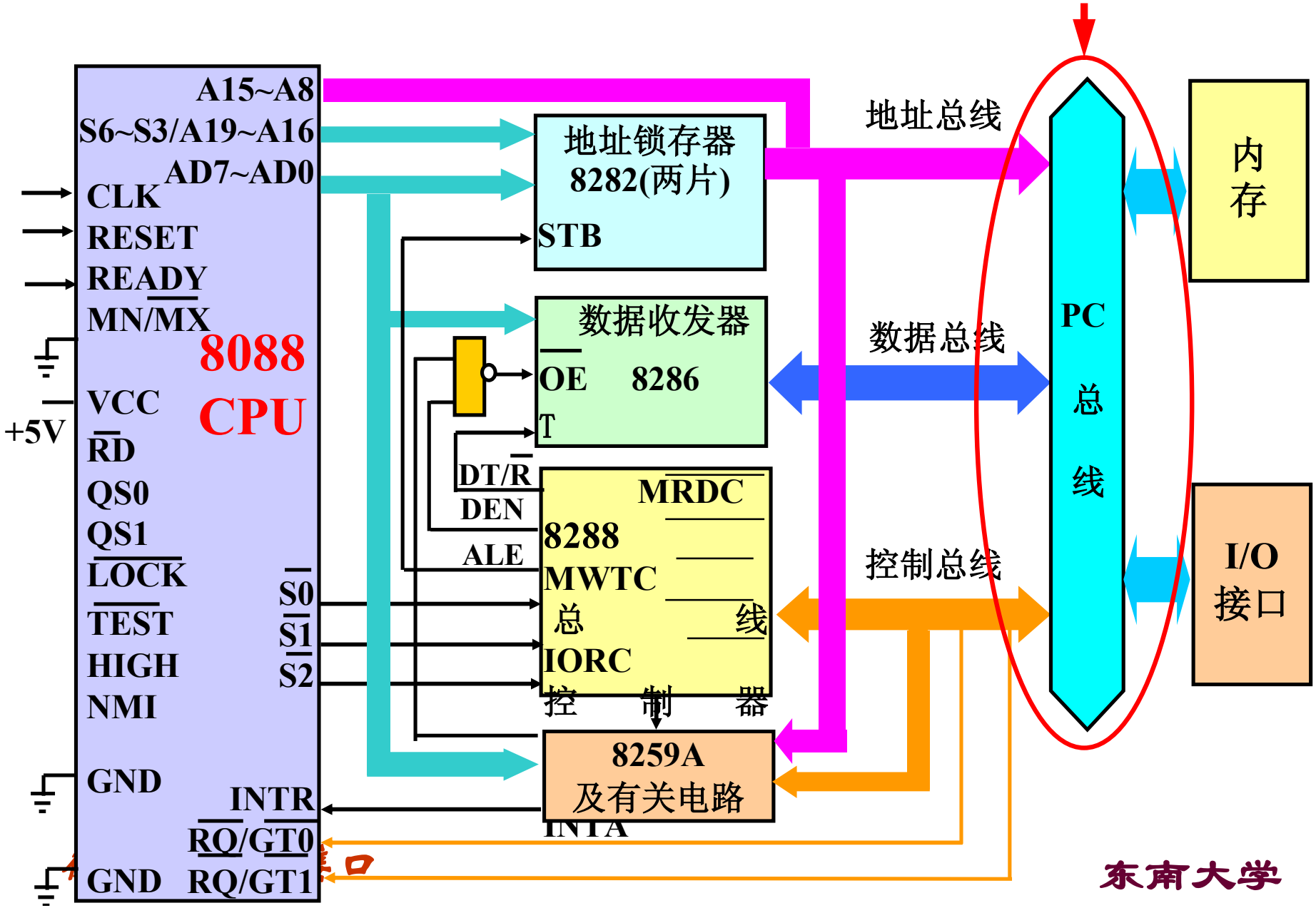
系统总线（ PC总线）

——计算机内部，插卡与插卡间的连接总线

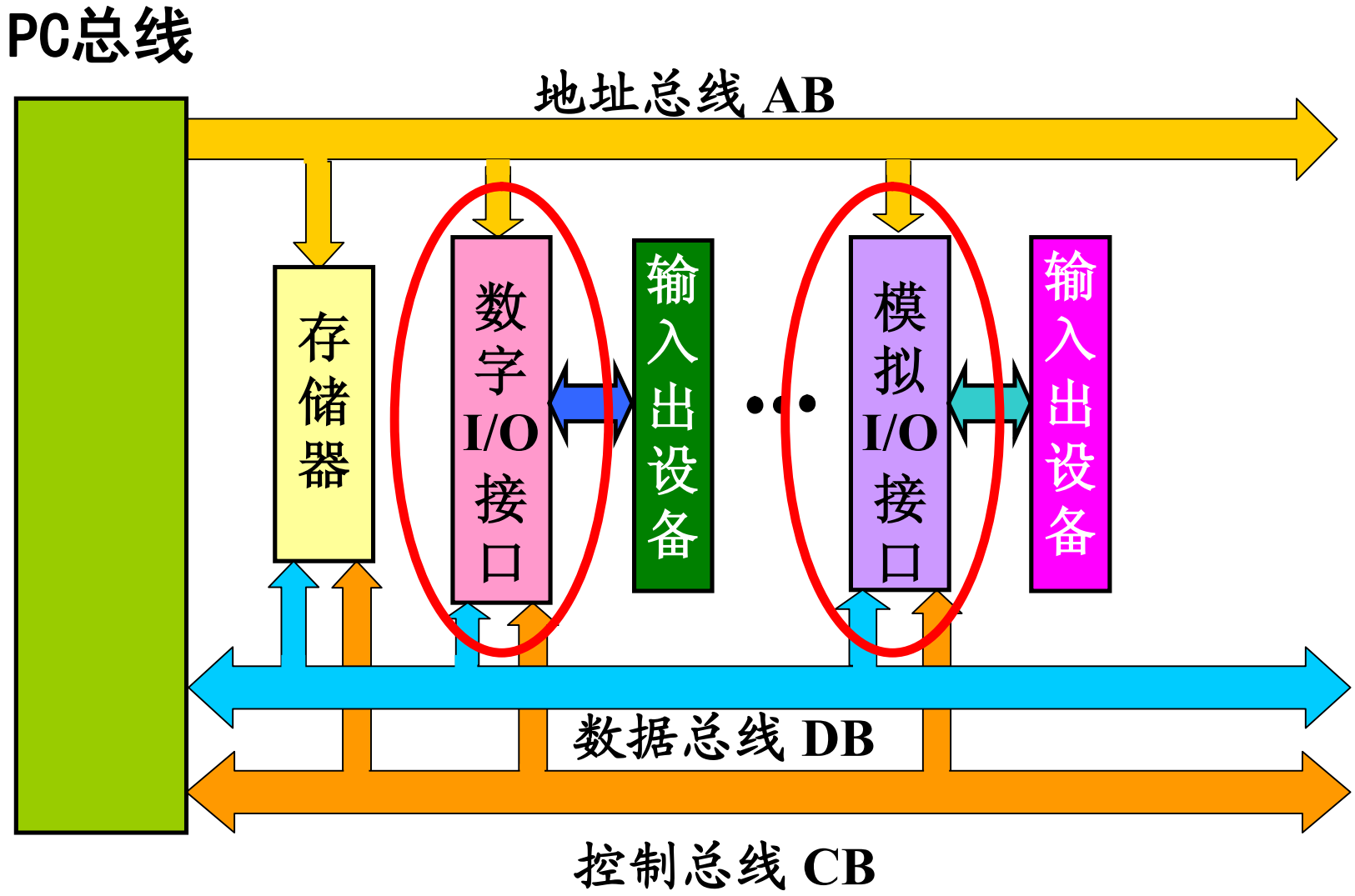
使计算机成为开放体系，实现技术的兼容与共享


（8086 CPU）片内总线
CPU总线
存储器
寄存器组
AH
AL
AX
BH
BL
BX
CH
CL
CX
DH
DL
DX
SI
DI
BP
SP
DS
ES
SS
CS
IP
地址加法器
数据暂存器
运算器
指令队列
总线接口控制电路
指令译码器
PSW标志寄存器
执行部件控制电路
地址总线AB
数据总线DB
控制总线CB
地址译码器
、、、
指令1
指令2
指令3
指令4
、、、
数据1
数据2
9Ah
、、、

8088(最大模式)结构下的PC总线-系统总线



微型计算机中I/O接口的作用



（广义的）接口和接口技术

接口(interface) ----- CPU、存储器、外设之间通过总线进行连接的电路部分，信息交换的中转：

MOV [BX],AL　MOV CX, [2000H]

INC WORD PTR [SI+100]

存储器接口

IN AL, 80H　OUT 40H, AL

IN AL, DX　OUT DX,AL

输入输出接口

I/O接口技术

研究CPU（PC）总线如何与外部世界进行最佳耦合与匹配，实现双方高效、可靠地交换信息的一门技术，体现软件、硬件结合，是微机应用的关键。

I/O接口功能

I/O接口：CPU控制外部设备的必经通道

实现：LSI/VLSI专用或通用接口芯片。

功能：

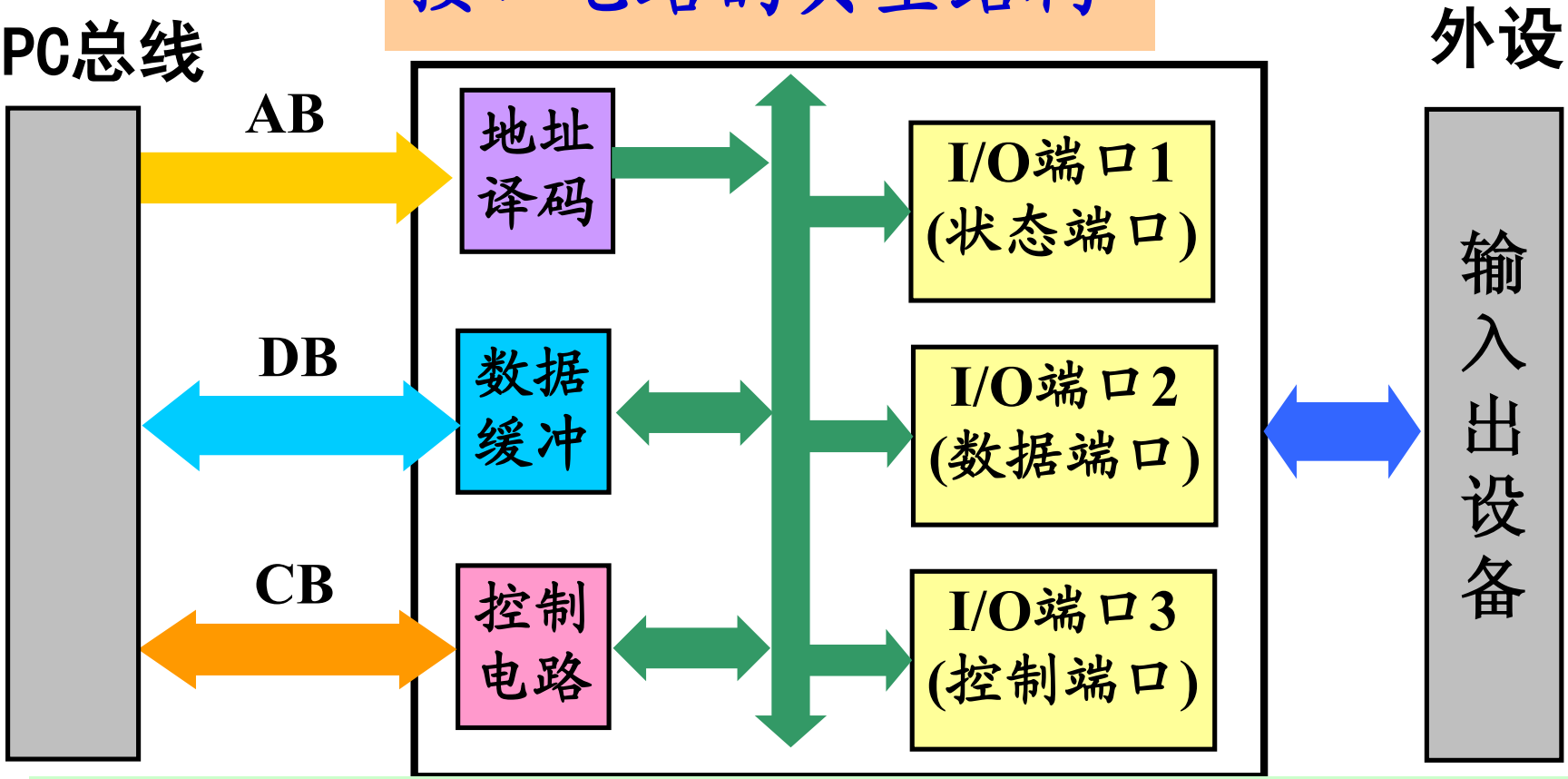
(1)地址译码I / O设备的选择:分时数据传送

(2)信息输入与输出（I/O接口按命令工作）

(3)数据的缓冲及锁存：时序匹配。

(4)信息的转换（设备的信息类型（数字、模拟量等）、电平（TTL电平、非TTL。电平等）及码制（二进制、十进制等）和信息格式（并行到串行或反之））。

接口电路的典型结构



- CPU对外设输入/输出的控制，通过对接口电路中各I/O端口的读/写操作完成。
- 端口→地址编号

I/O端口的编址方式1

端口与存储器分别独立编址（I/O映射方式）

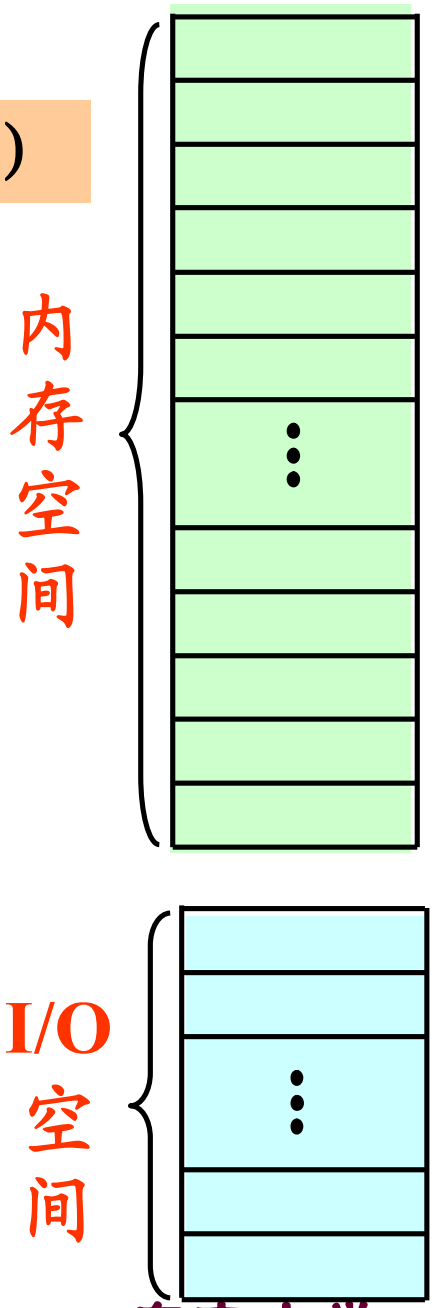
例 80X86 ,MCS96系列，Z80系列

特点：

- 端口与存储器分别独立编址
  端口不占用内存空间
- 设有专门的 I/O指令对端口进行读写，
  对内存操作的指令不能用于I/O端口

例

```
MOV     AL , [ 0040H ]   对内存操作
IN      AL ,40H          对端口操作
```



8088CPU:

采用I/O端口与存储器分别独立编址

可寻址2^{20}= 1M个内存单元

内存范围00000 ~0FFFFFH

内存单元的地址有多种寻址方式

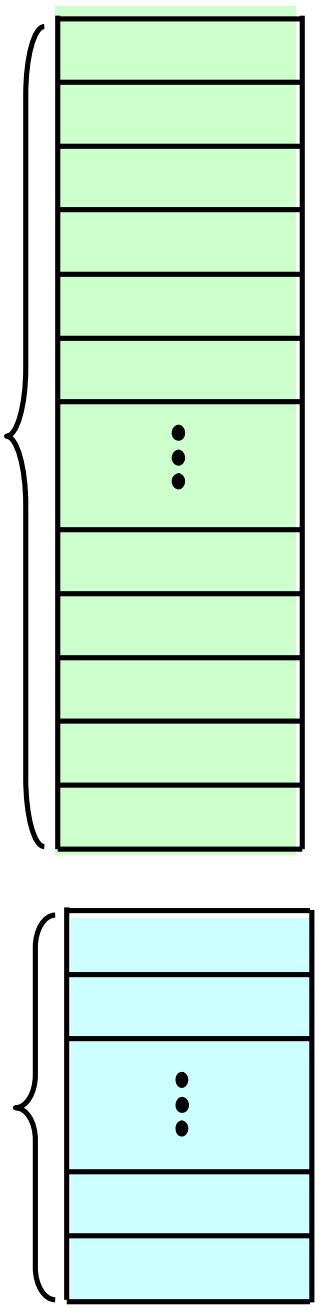
可寻址2^{16}= 64 K个I/O端口

I/O端口范围0000 ~ 0FFFFH

I/O端口的地址由

一个8位二进制数直接寻址

或DX寄存器间接寻址

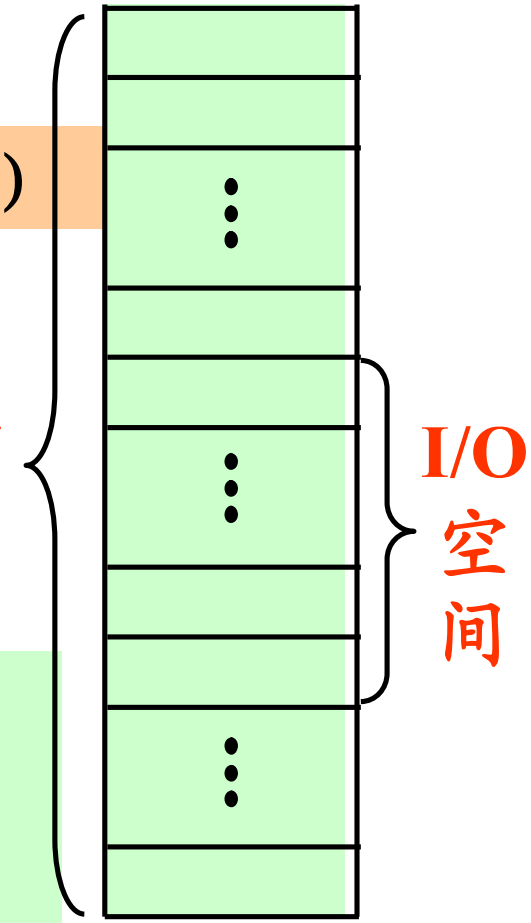


I/O端口的编址方式2

（端口与存储器统一编址存储器映射方式）

例 motorola的M6800系列

iMCS51系列



特点：

- I/O端口相当于内存的一部分，
  使内存容量减小
- 对I/O端口的读/写与对存储器的读/写相同，
  所有可对内存操作的指令对I/O端口均可使用
- 指令系统中不专设I/O指令。

8088的输入/输出指令

1. 输入指令IN

IN AL, port 或 IN AL, DX;

IN AX,port 或 IN AX,DX

(Port)→(AL),(Port+1)→(AH)

2. 输出指令OUT

OUT port, AL 或 OUT DX,AL;

OUT port, AX 或 OUT DX,AX

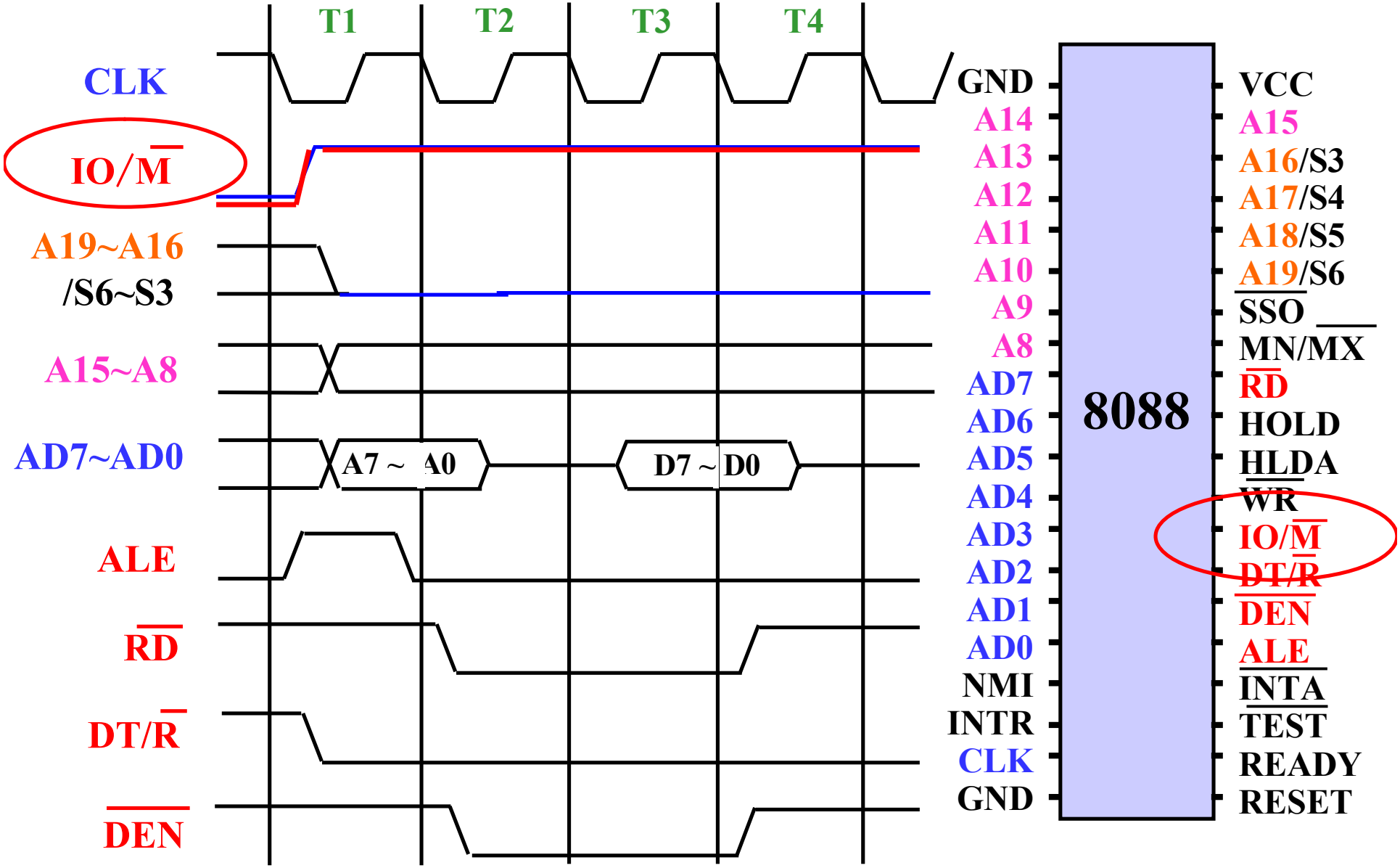
8088CPU，读、写I/O端口的时序

与读、写存储器的过程相似，不同之处：

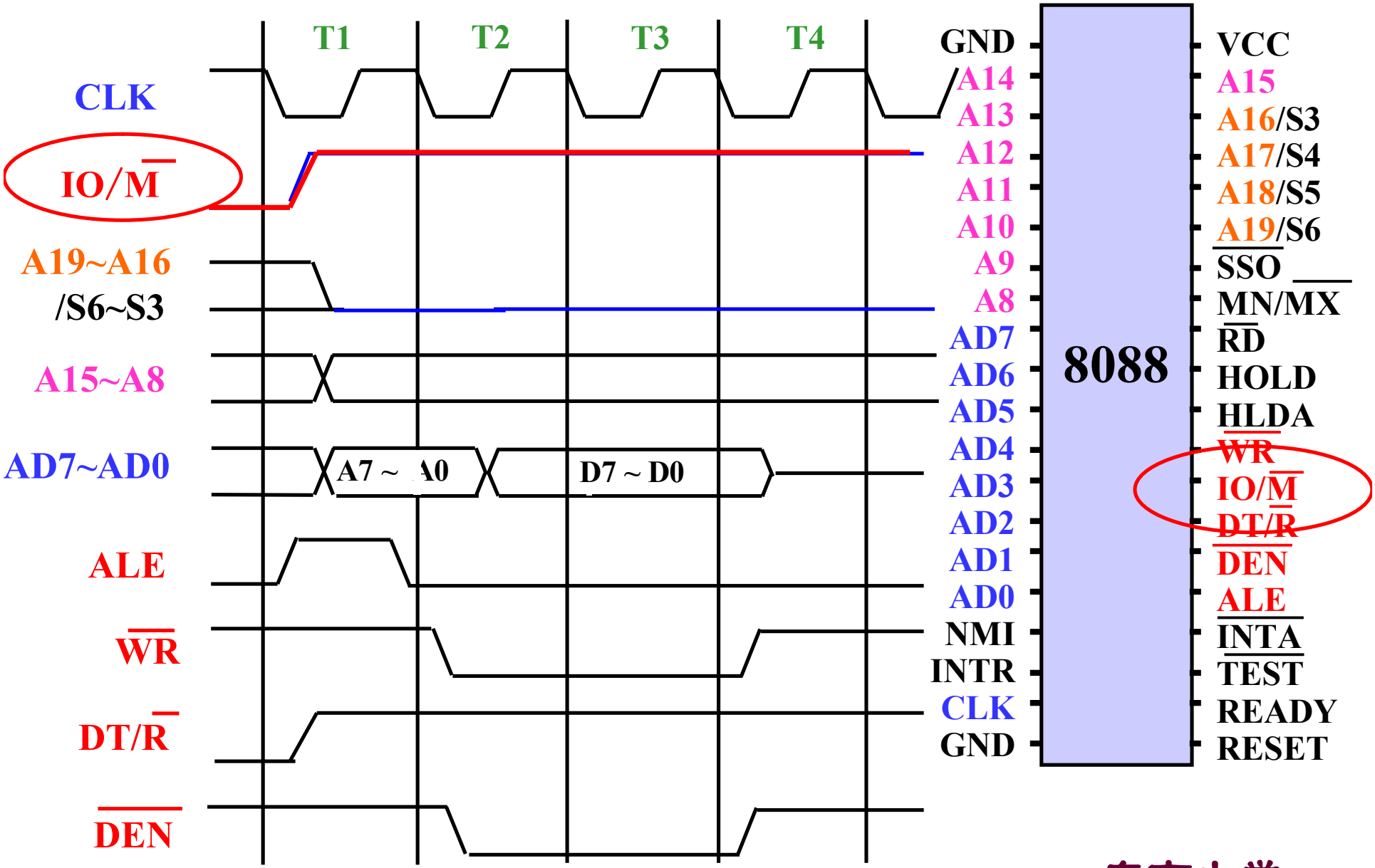
1. IO/$\overline{M}$变高，CPU操作I/O端口。
2. 端口的地址信号出现在A15~A0上，

   A19~A16全为低电平。

8088CPU最小模式下，I/O端口读周期时序

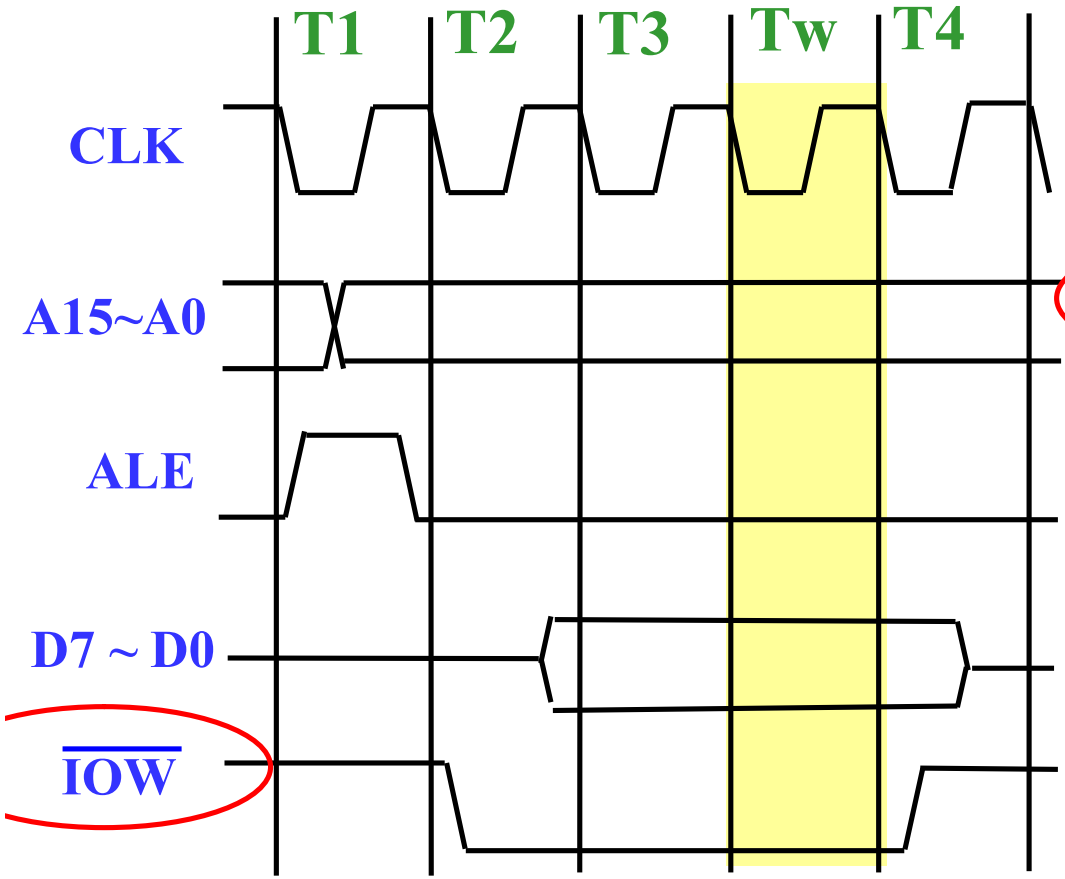


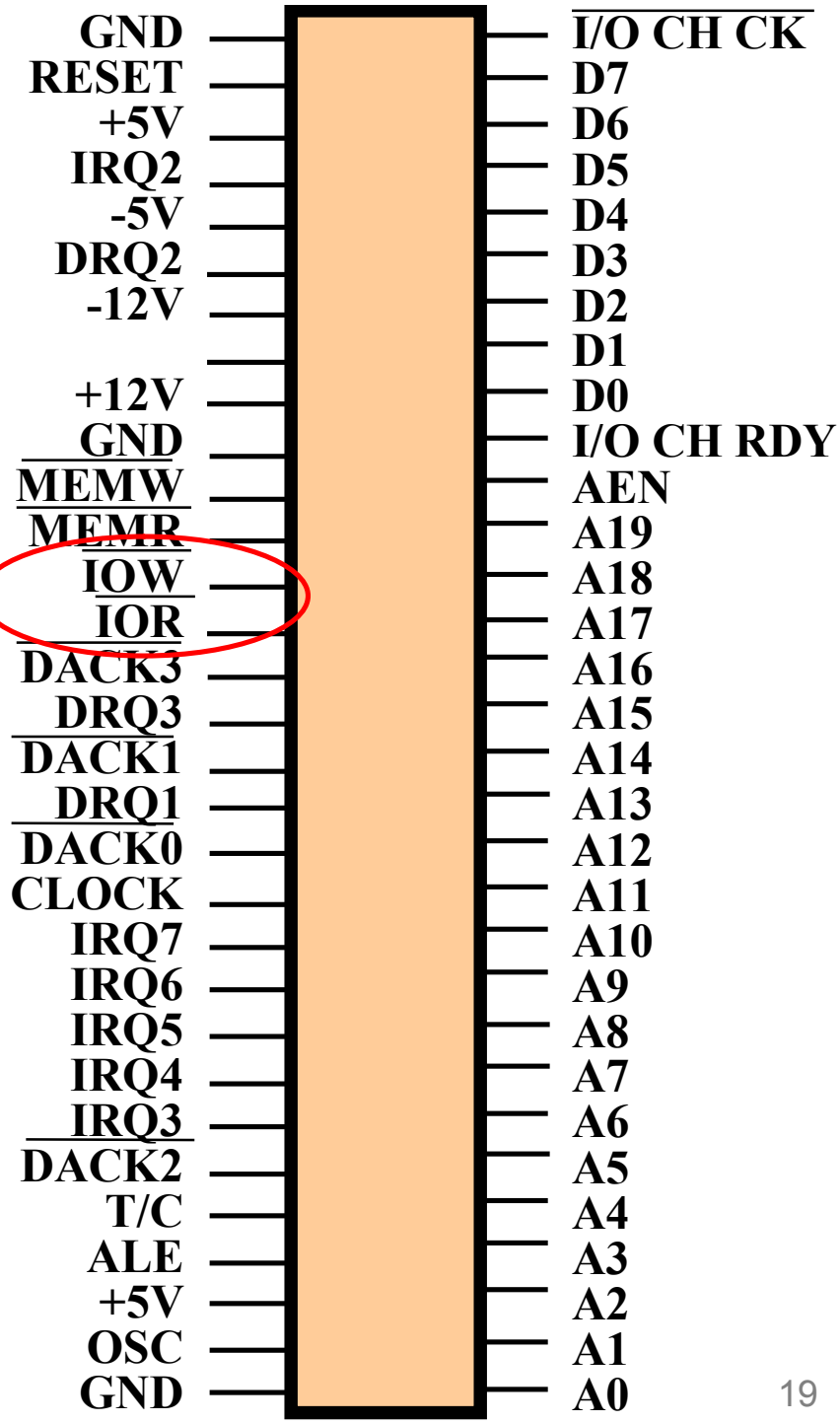
8088CPU最小模式下，I/O端口写周期时序



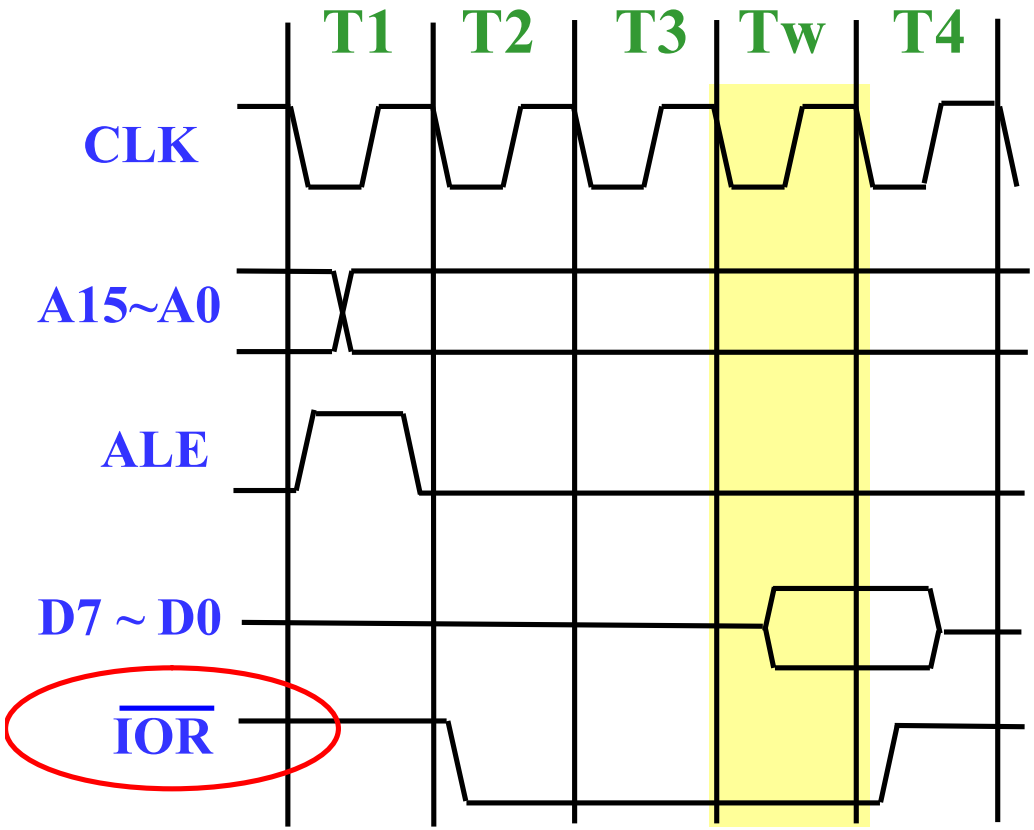
PC总线

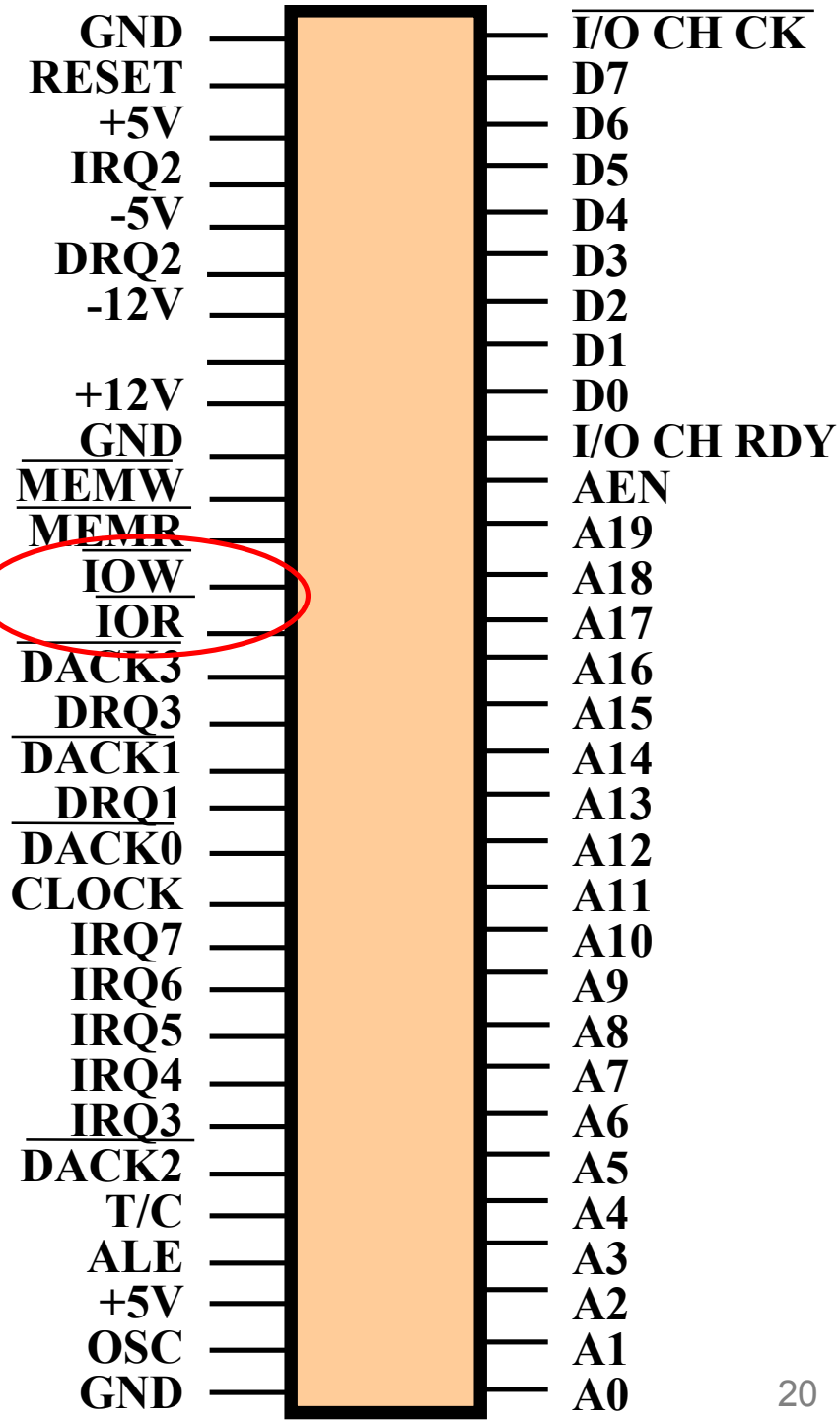
I/O端口写周期时序



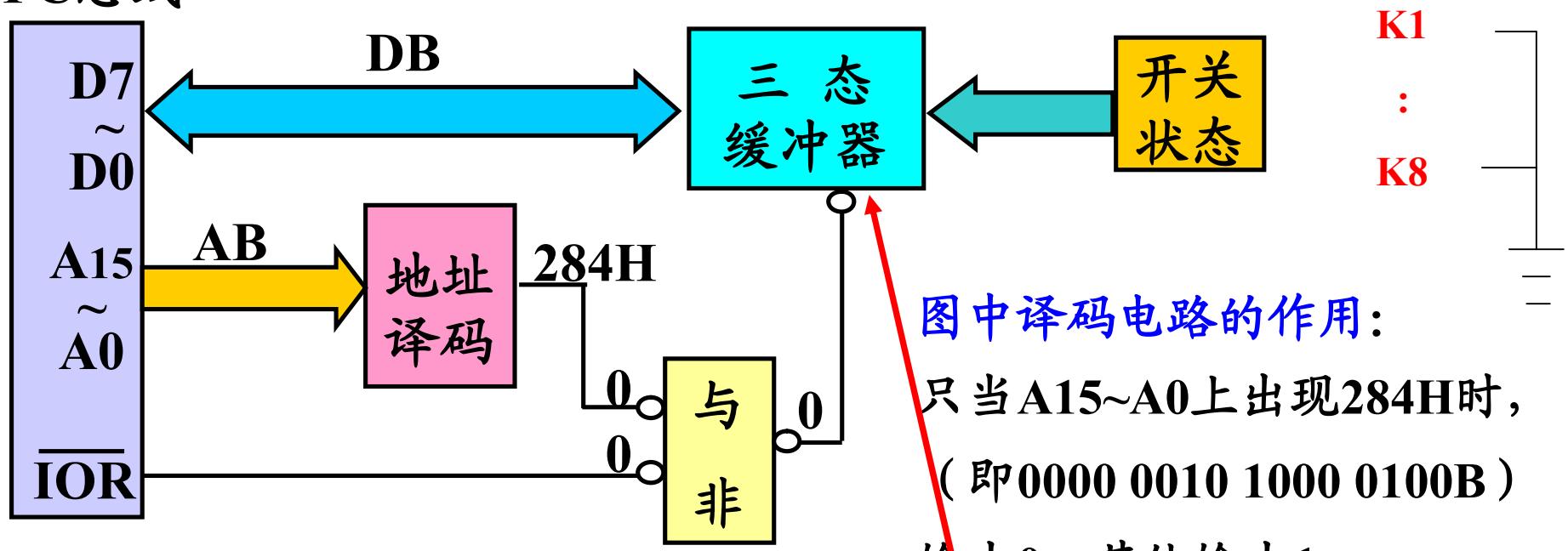


PC总线
I/O端口读周期时序





例　一个输入设备的简单接口电路



图中译码电路的作用：

只当A15~A0上出现284H时，

（即0000 0010 1000 0100B）

输出0，其他输出1。

该电路在CPU执行指令

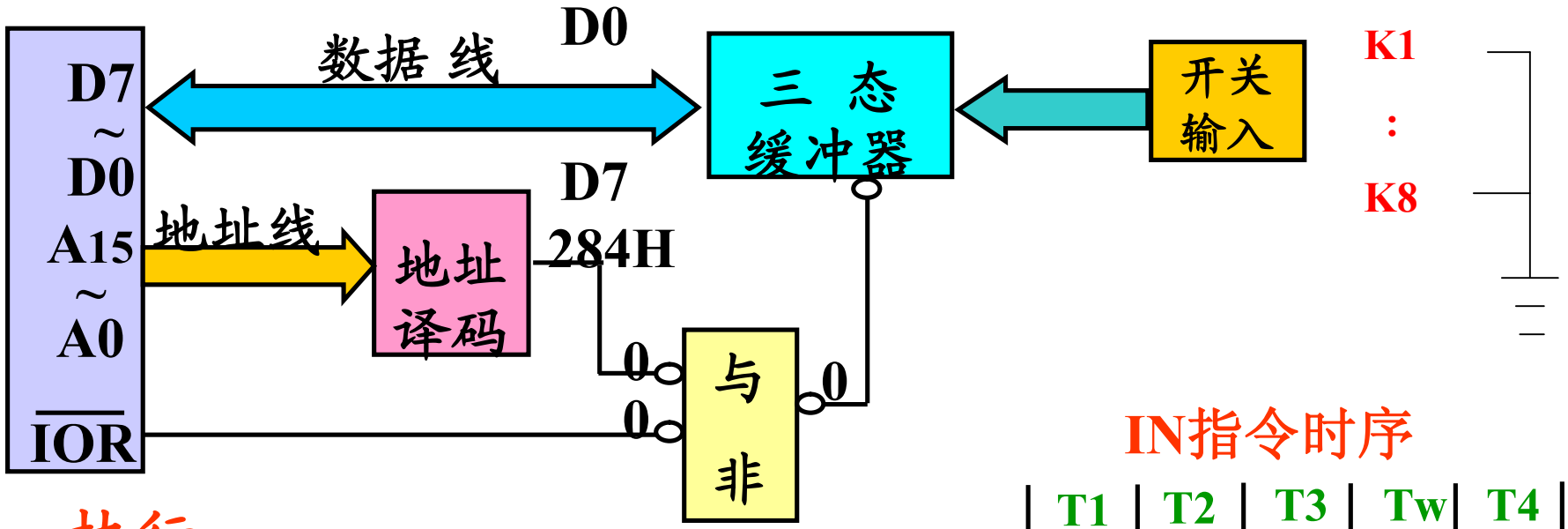
MOV DX, 284H

IN AL, DX

将输入设备的数据读入CPU内AL中

参考P290图5.11 74LS244G1/G2

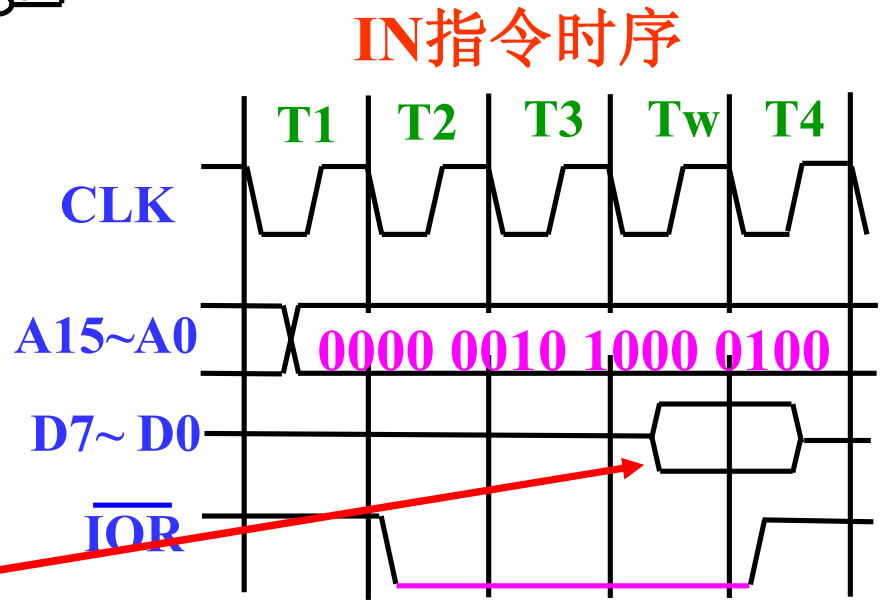
PC总线



执行:

MOV DX, 284H

IN AL, DX

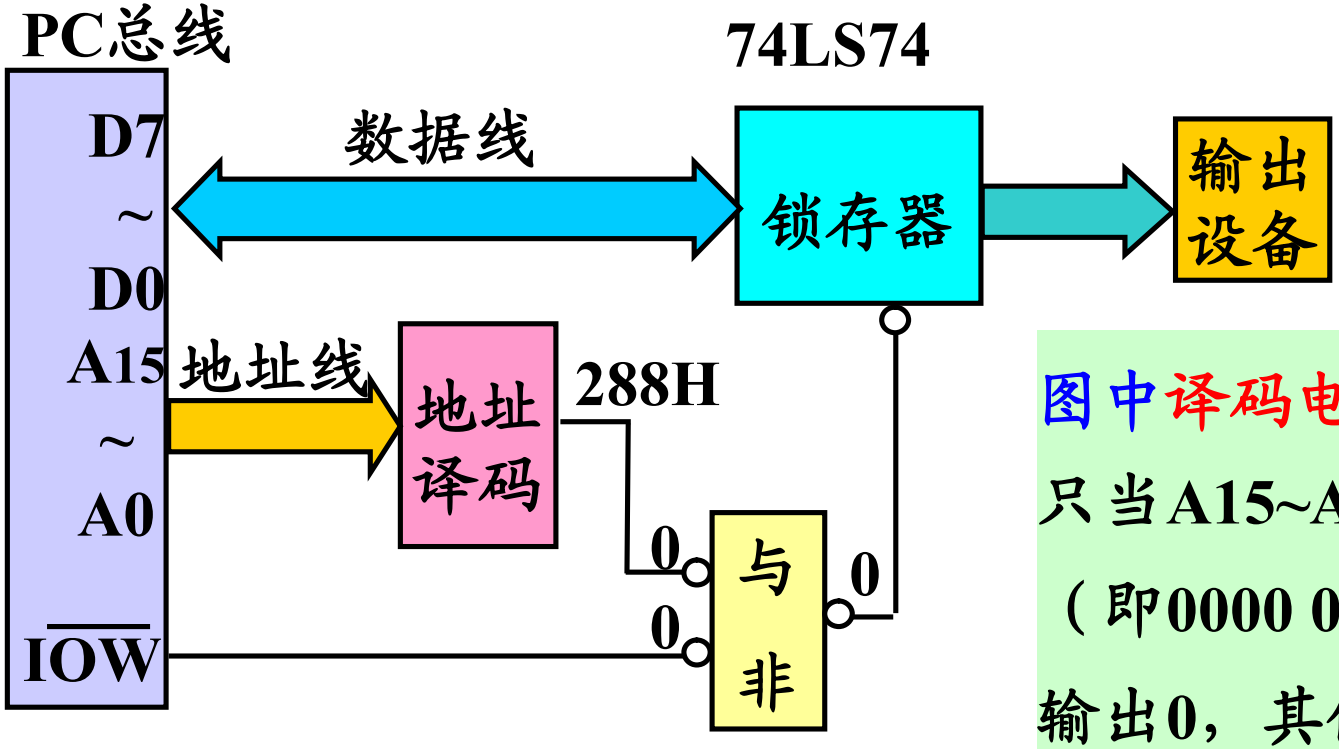


问题

P290图5.11 K1,K4,K7闭合时

DB=?, (AL)= 01101101B=6DH

例： 一个输出设备的简单接口电路



例:LED指示灯

图中译码电路的作用:

只当A15~A0上出现288H时，

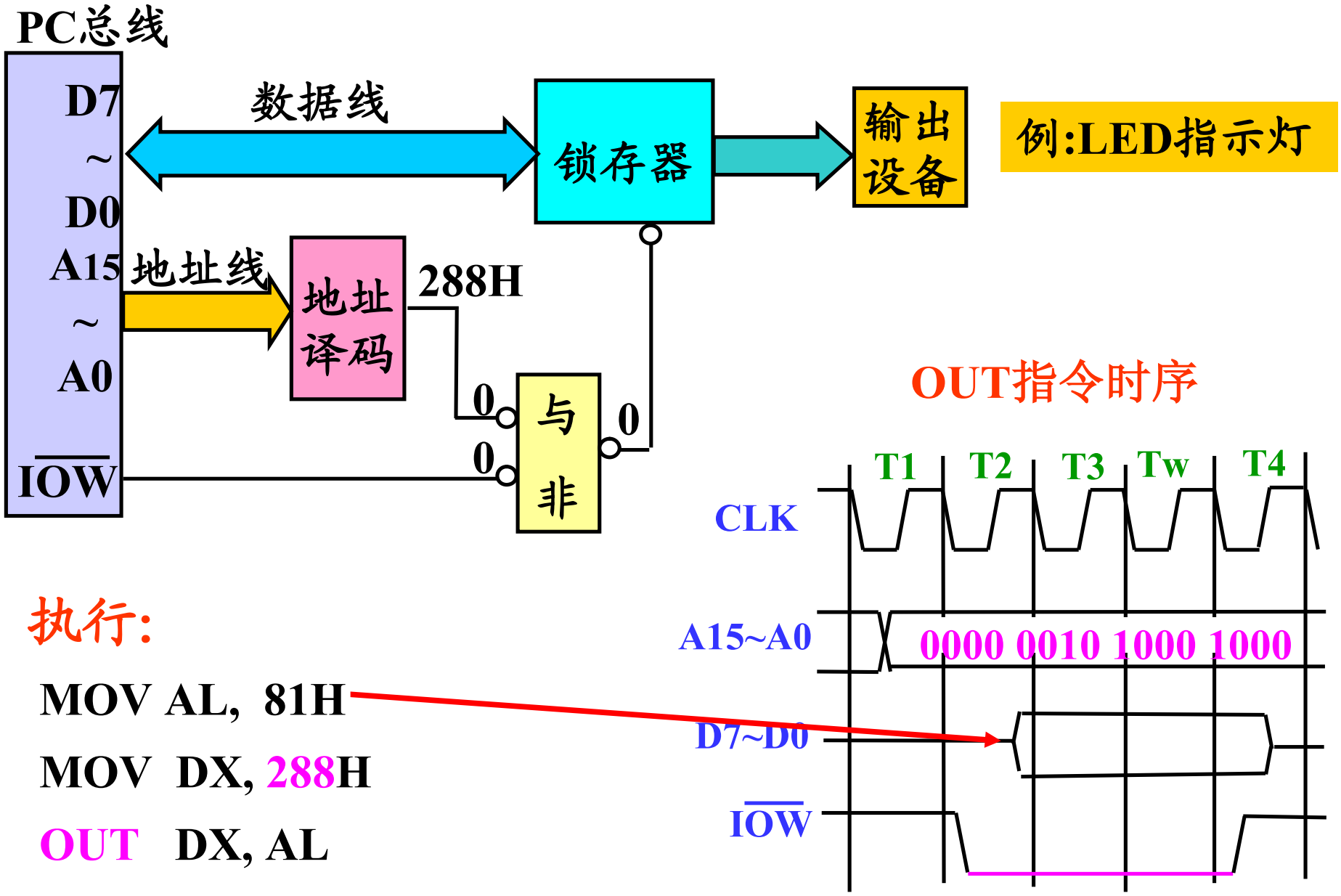
（即0000 0010 1000 1000B）

输出0，其他输出1。

该电路在CPU执行指令

```
MOV AL, 81H
MOV  DX, 288H
OUT   DX, AL
```

CPU内AL中的数据81H送至输出设备

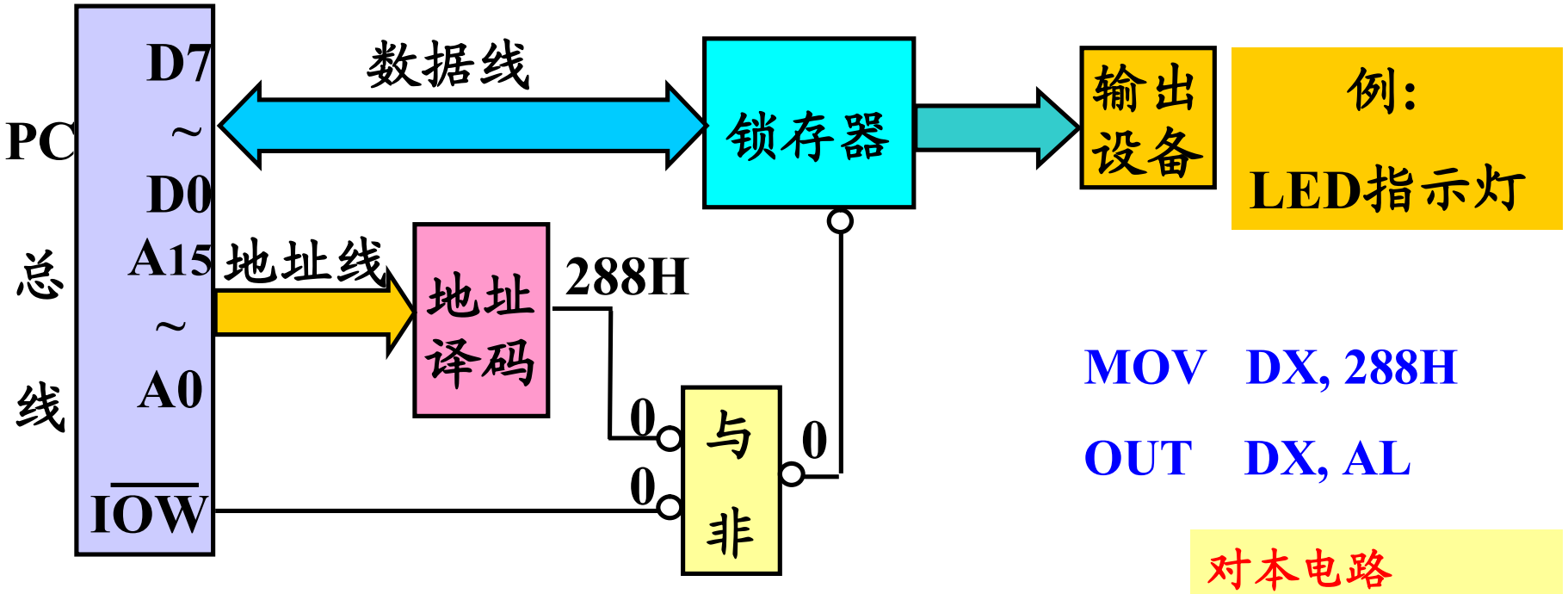


执行:

MOV AL, 81H

MOV DX, 288H

OUT DX, AL



对本电路

MOV DX,288H

IN AL,DX

结果如何?

输出设备接口电路，从硬件上保证:

只在CPU执行从288H端口输出数据时，

锁存器处于触发状态，其输出随输入变化

而CPU执行其它指令时，

锁存器均处于锁存状态,其输出不随输入变化，

思考：其他的指令为什么不可以？

例：OUT 50H,AL； MOV [0288H],AL

I/O端口的译码

1. 译码电路的作用

2. 译码电路的构成与设计(与存储器译码相似)

3. 片内译码和片选译码

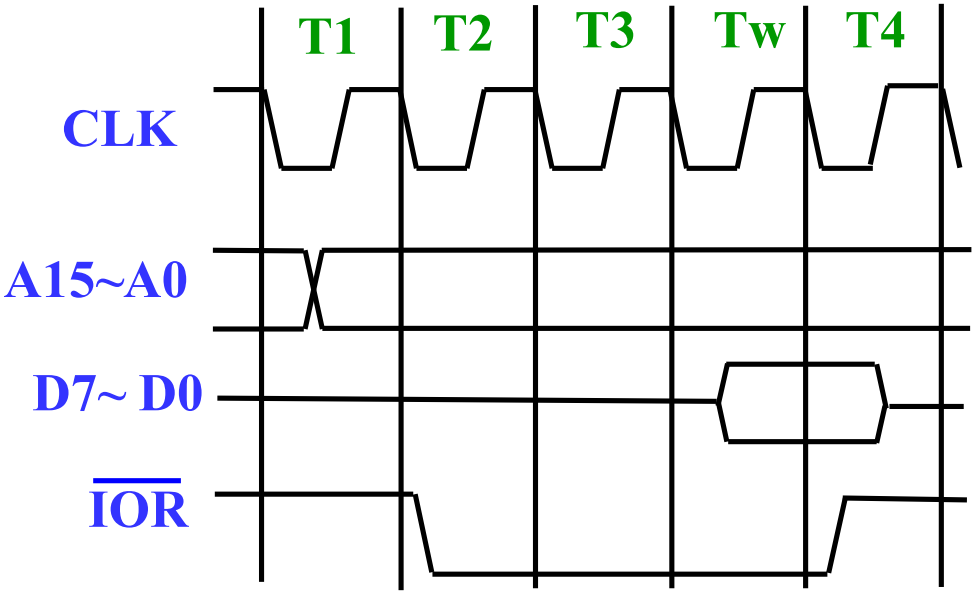
I/O译码电路的作用

将CPU执行IN/OUT指令发出的信号，

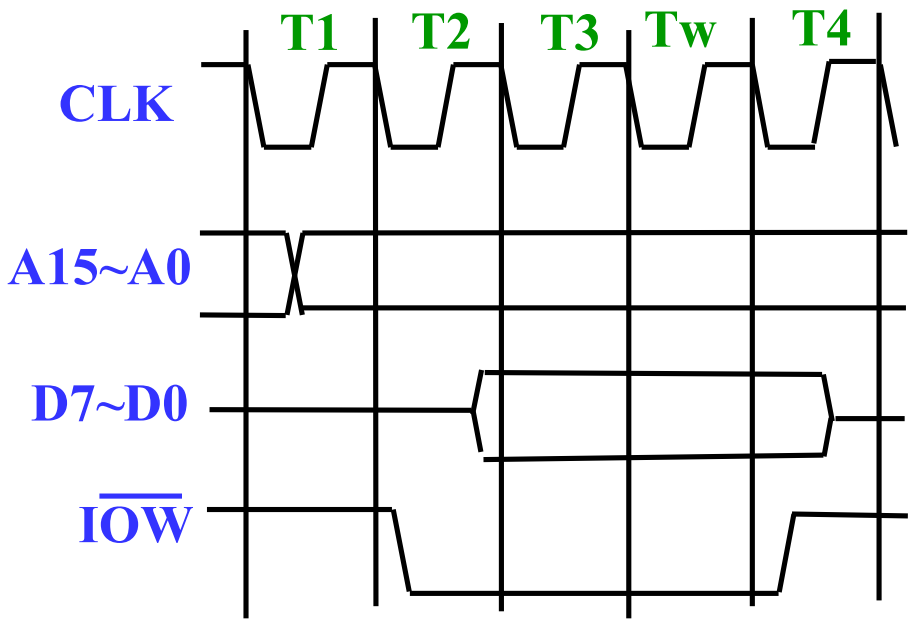
“翻译”成欲操作端口的选通信号，

此信号常作为接口内三态门或锁存器的控制信号，

接通或断开接口数据线与系统的连接。



IN指令时序



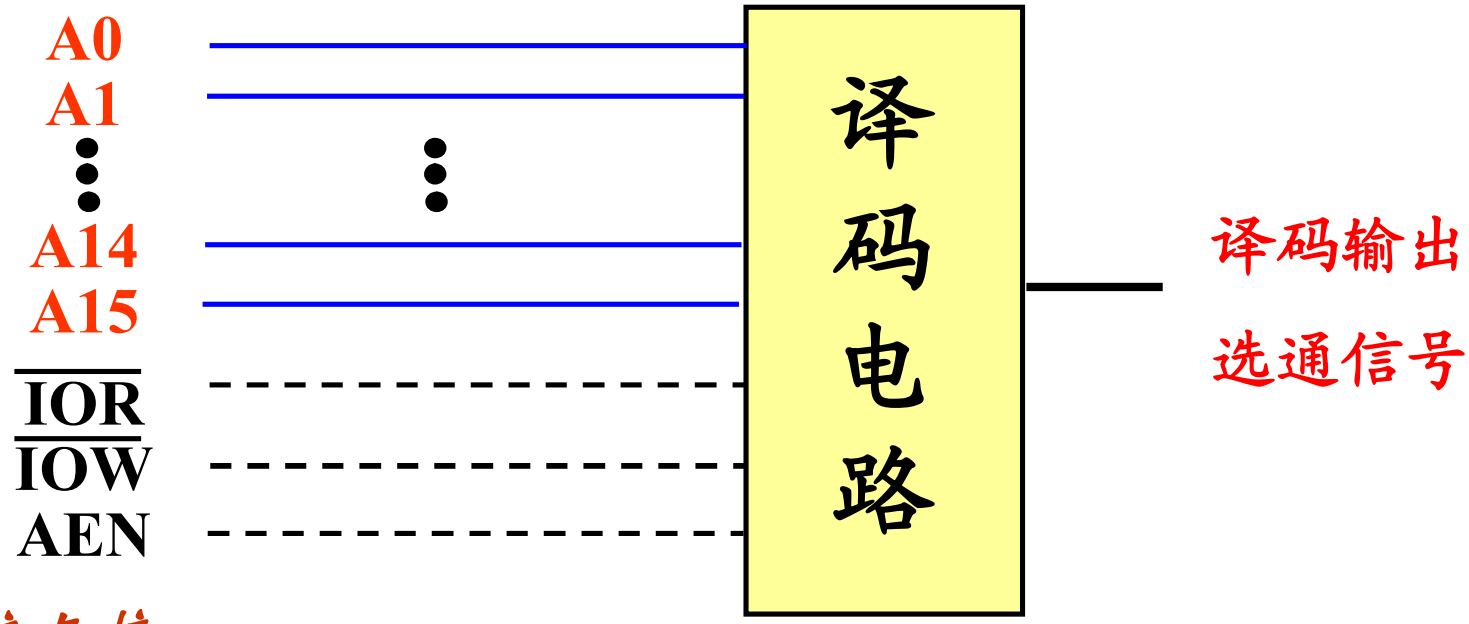
OUT指令时序

设计译码电路的方法

1. 根据端口地址确定地址信号A15~A0的条件取值，用门电路、译码器及组合、PLD/GAL实现满足条件情况的电路。

2. 除端口的地址信号参加译码外，控制信号$\overline{IOW}$、$\overline{IOR}$ ($IO/\overline{M}$、 AEN) 也可参加译码

3、端口的选通信号通常为低电平有效。



例： 设计端口地址为218H的译码电路

分析

CPU执行IN/OUT指令时，发出端口的地址信号

```
MOV   DX,  218H
IN    AL,  DX
```

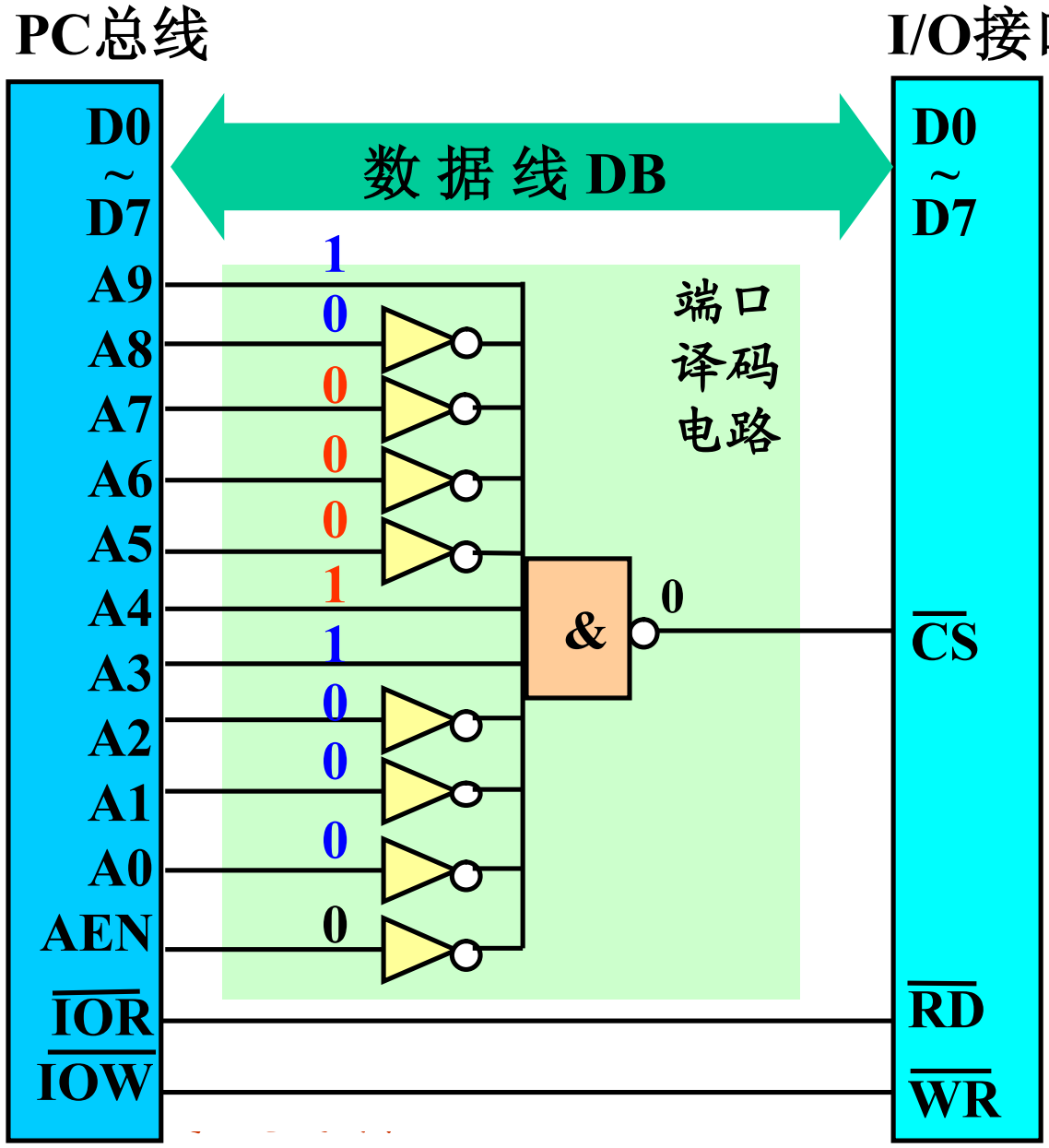
或 `OUT   DX, AL`

对应218H端口的地址信号为（只取A9~A0）：

A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	（地址信号）
1	0	0	0	0	1	1	0	0	0	B
2		1				8				H

只要满足此地址取值的译码电路均可

方法一、用门电路实现218H的地址译码



译码电路满足:

当地址信号A_9~A_0为:

A_9 A_8 A_7 A_6 A_5 A_4 A_3 A_2 A_1 A_0

1 0 0 0 0 1 1 0 0 0

即218H时，输出“0”，

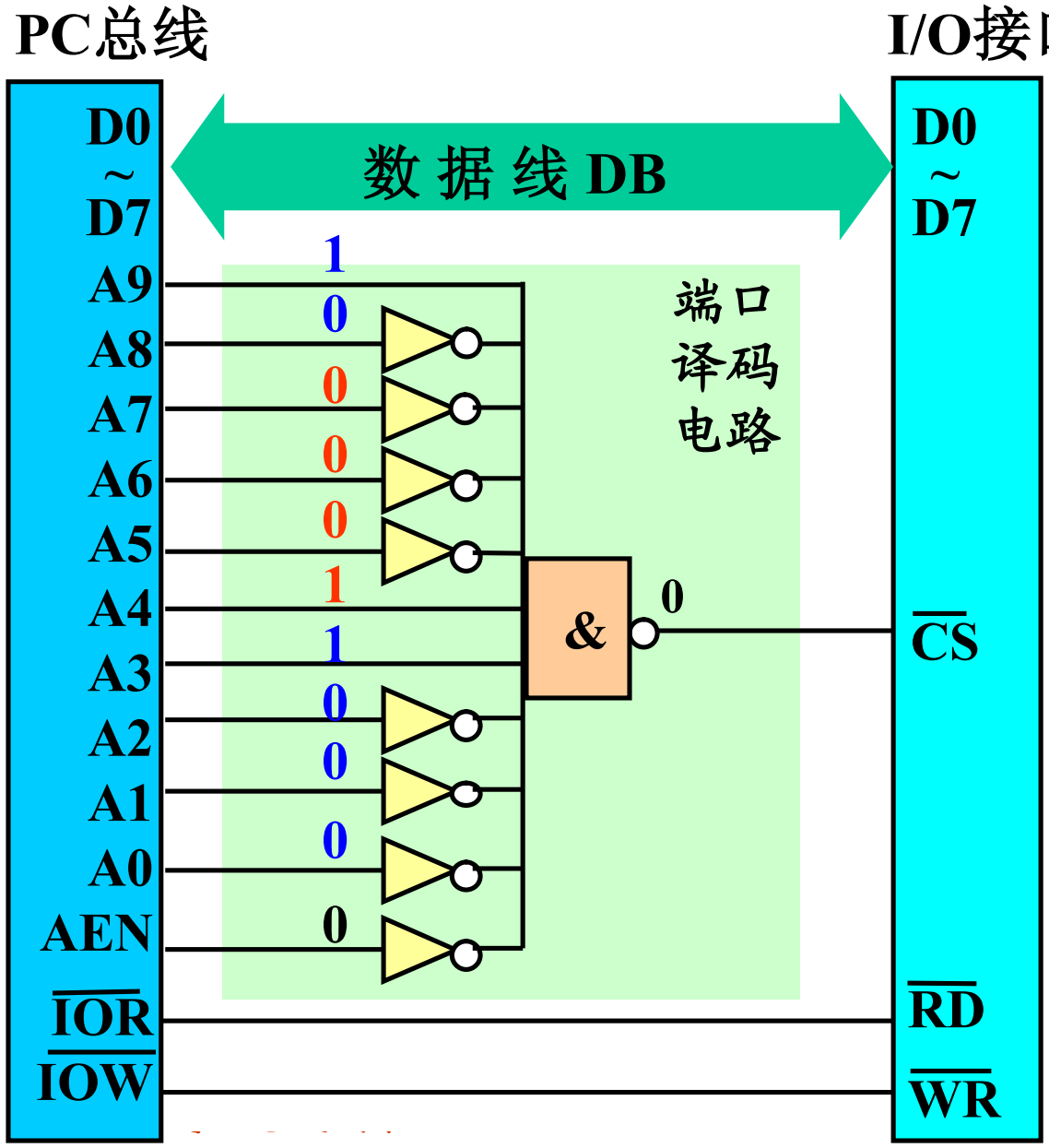
使I/O接口的$\overline{CS}$有效

否则输出“1”

使I/O接口的$\overline{CS}$无效

（未选中）

注意：译码中的地址重叠



高位地址线A15~A10
未参与译码！

使地址A15~A0满足：

× × × × × × 10 0001 1000

均能输出“0 ”低电平，

所以该电路使：

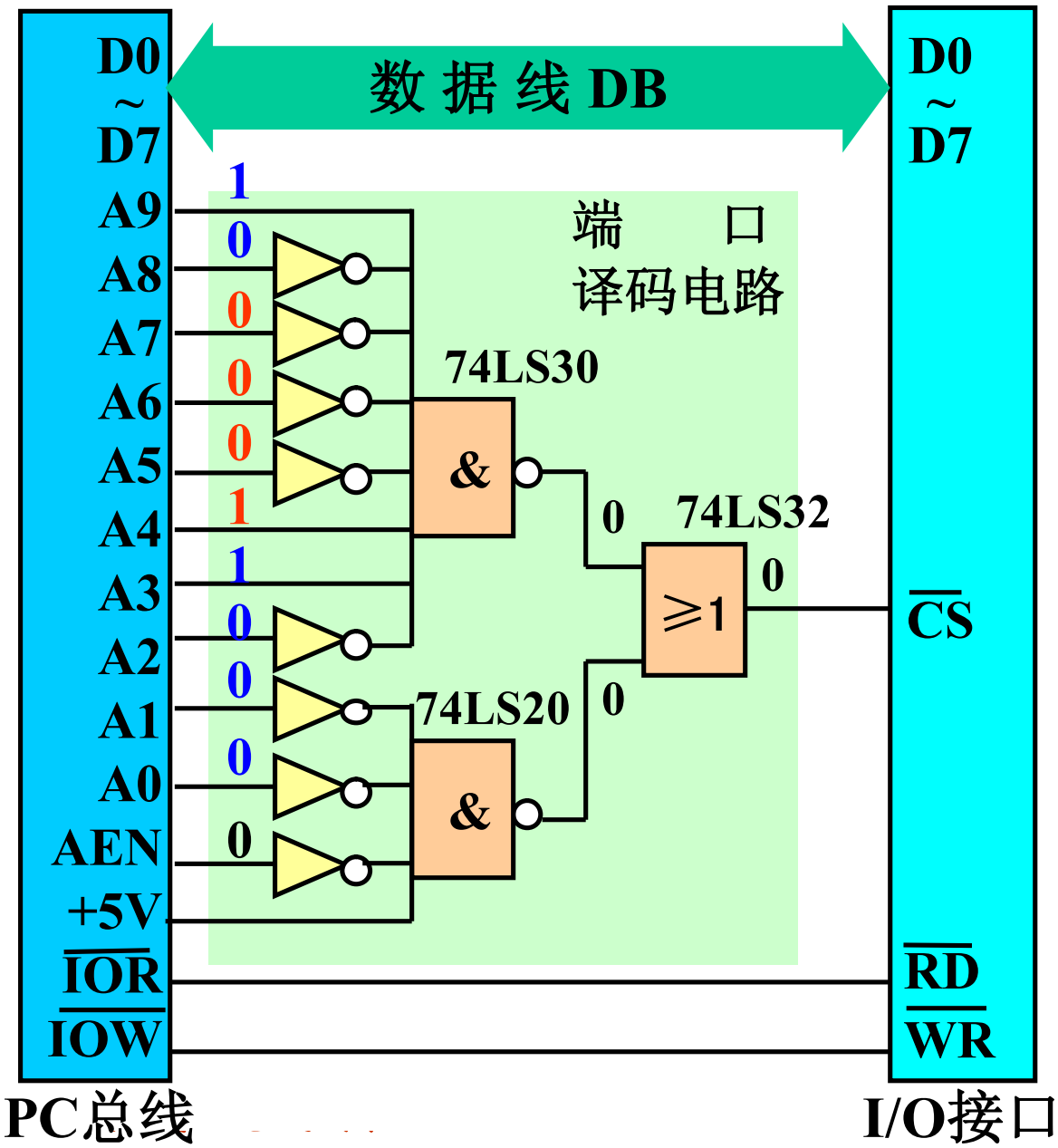
一个端口对应多个地址

共2^6=64个

218，618，

A18，E18　等等

（采用门电路的）译码电路的具体实现



74LS30为 8 输入与非门

74LS20为 4 输入与非门

74LS32为 2 输入或门

当地址信号为:

$A_9\ A_8\ A_7\ A_6\ A_5\ A_4\ A_3\ A_2\ A_1\ A_0$

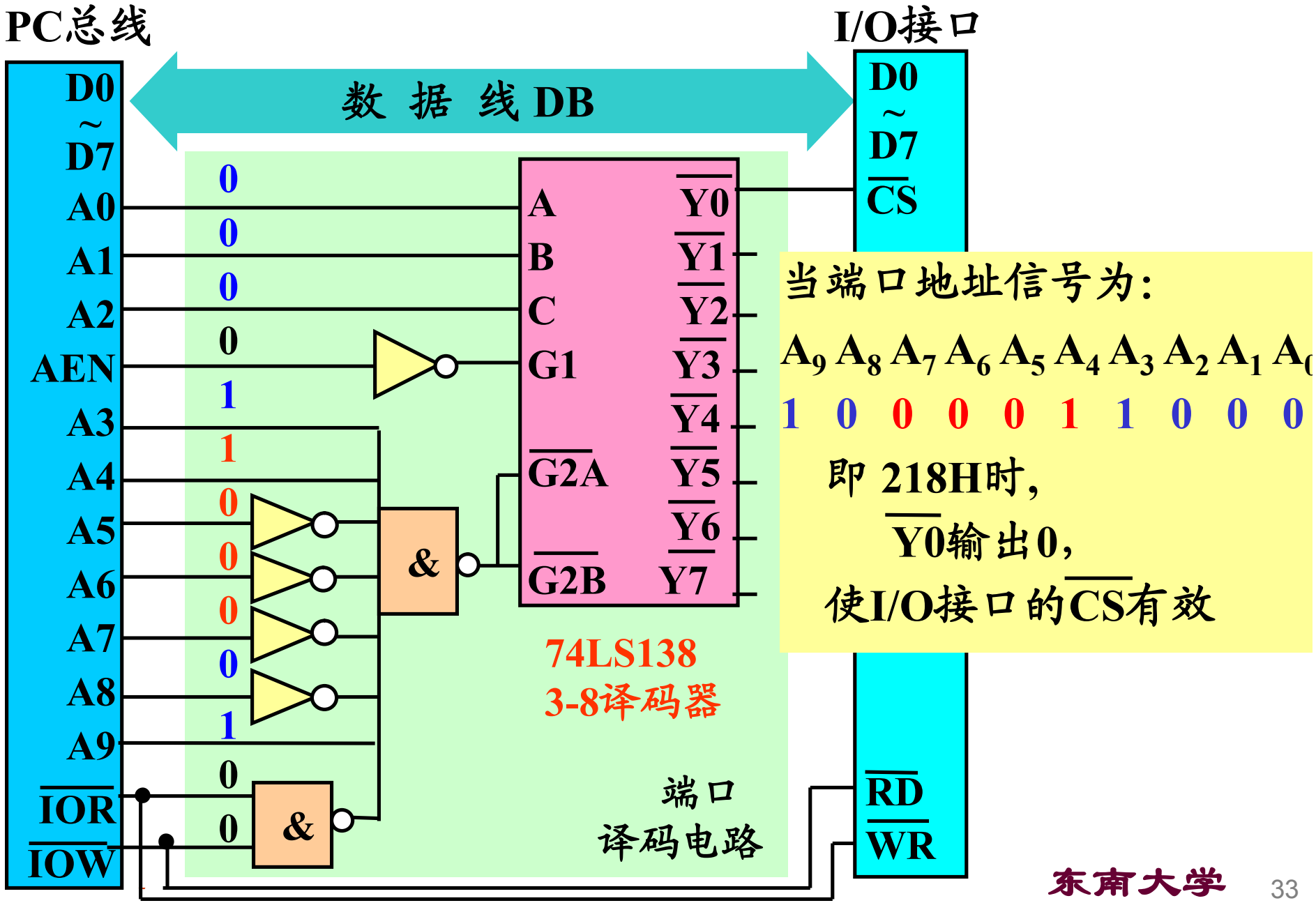
1 0 0 0 0 1 1 0 0 0

即地址为 218H

或门74LS32输出“0 ”，

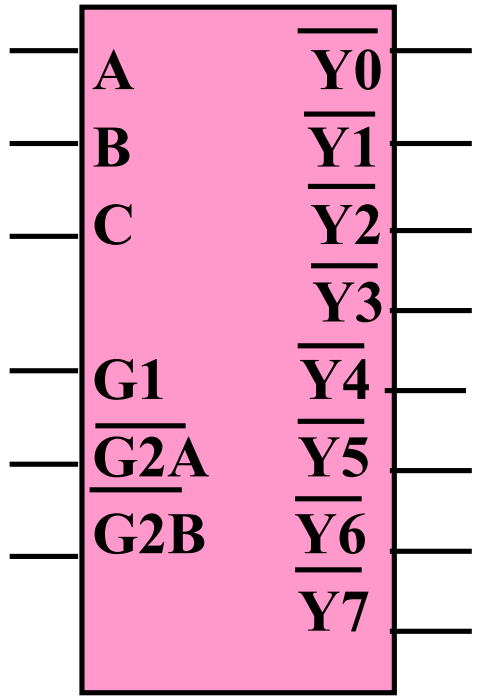
使I/O接口的$\overline{CS}$有效。

方法二、用译码器、门电路组合实现218H的地址译码



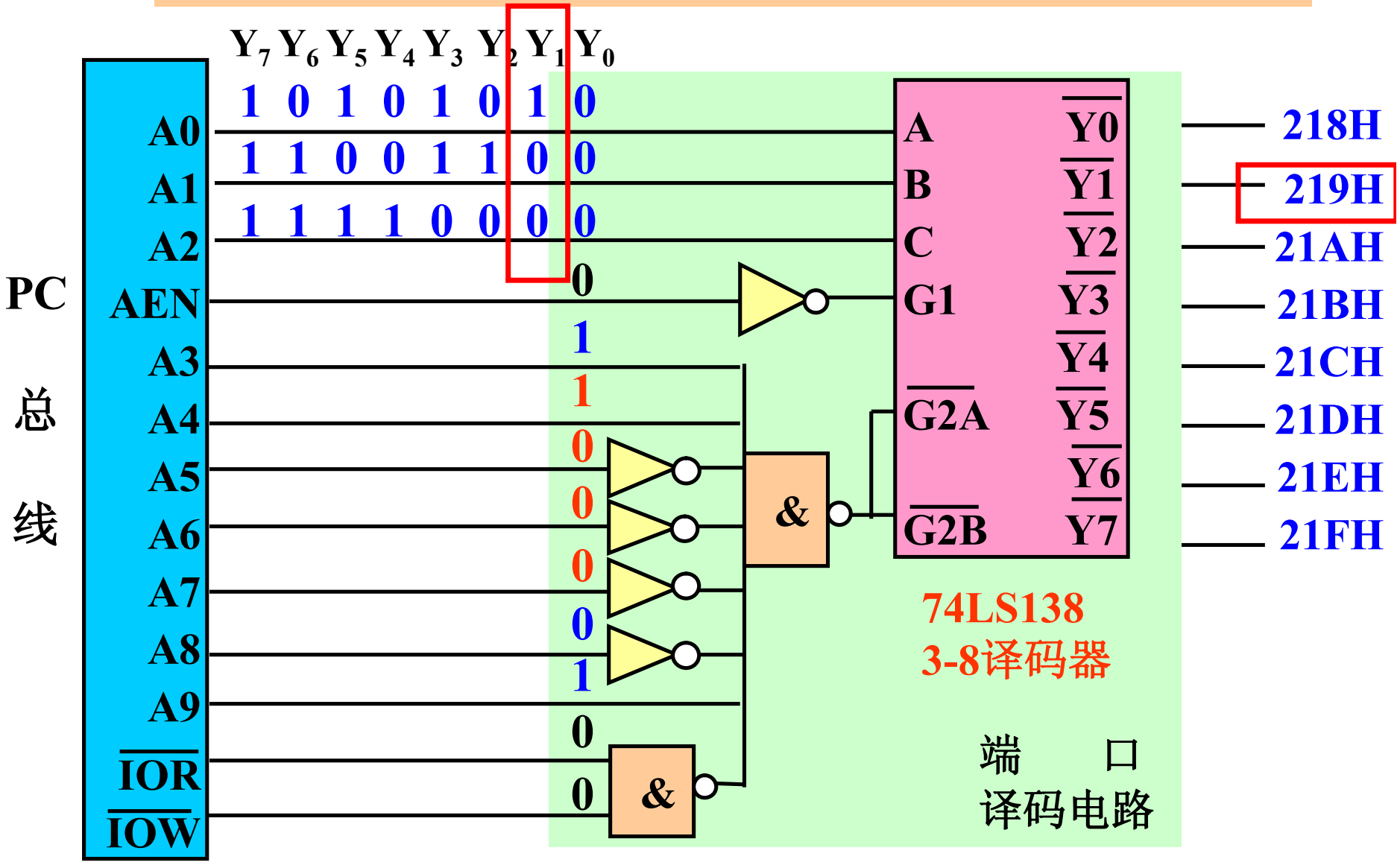
74LS138译码器功能表

使能输入			选择输入			$\overline{Y0}\sim\overline{Y7}$ 输出
G1	$\overline{G2A}$	$\overline{G2B}$	C	B	A	
1	0	0	0	0	0	$\overline{Y0}=0$，其余为1
1	0	0	0	0	1	$\overline{Y1}=0$，其余为1
1	0	0	0	1	0	$\overline{Y2}=0$，其余为1
1	0	0	0	1	1	$\overline{Y3}=0$，其余为1
1	0	0	1	0	0	$\overline{Y4}=0$，其余为1
1	0	0	1	0	1	$\overline{Y5}=0$，其余为1
1	0	0	1	1	0	$\overline{Y6}=0$，其余为1
1	0	0	1	1	1	$\overline{Y7}=0$，其余为1
0	×	×	×	×	×	全部为1
×	1	×	×	×	×	全部为1
×	×	1	×	×	×	全部为1

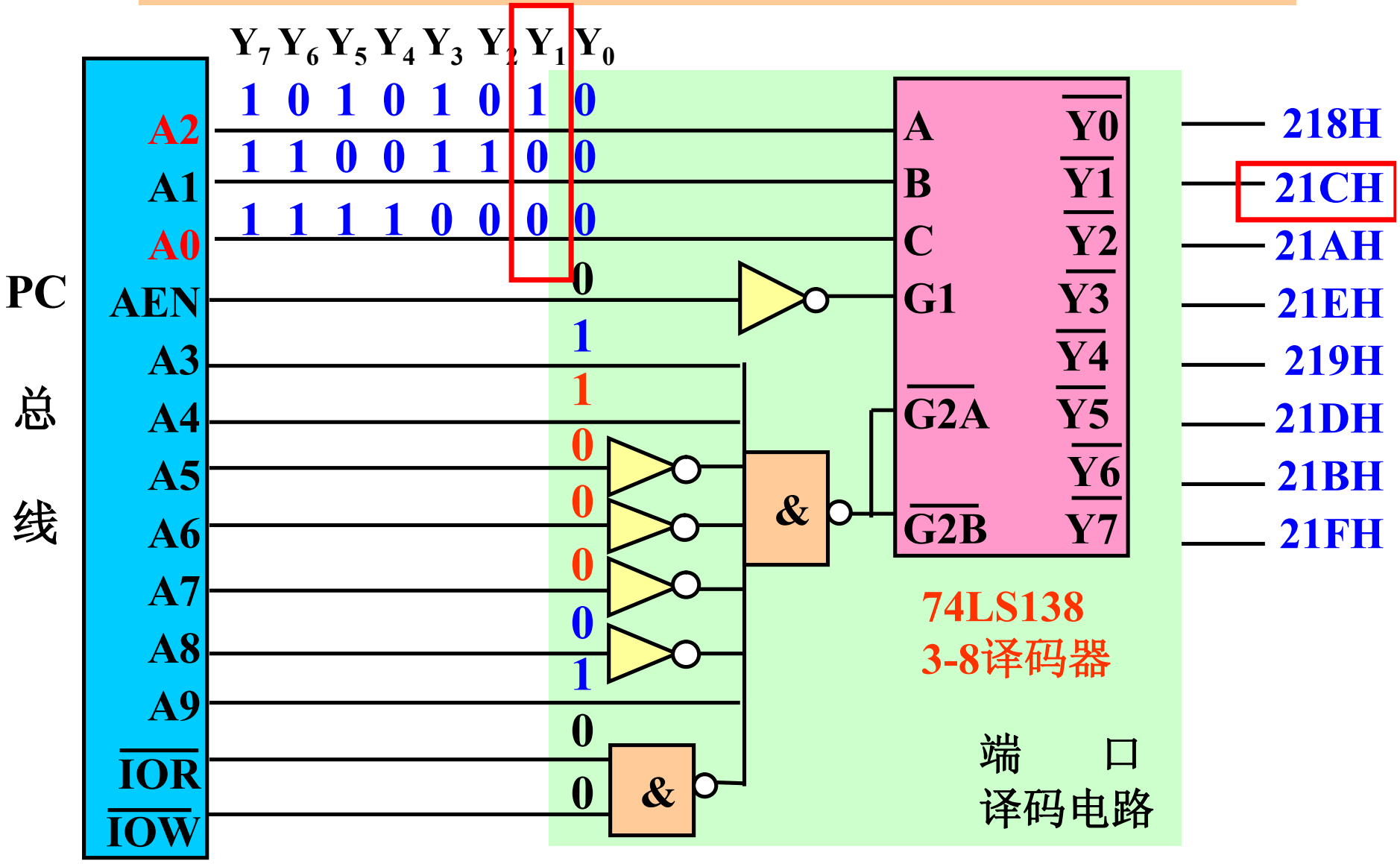


74LS138
3-8译码器

思考： $\overline{Y1}$~ $\overline{Y7}$ 对应的端口地址各是多少？



思考2：将A0与A2互换，$\overline{Y0}$-$\overline{Y7}$对应地址?



CPU（总线）与外设间的数据传送方式

问题的提出：

CPU与外设的工作速度不一致，尤其是当外设由其他CPU或时序电路控制时更突出，如何解决效率和可靠性？

数据传送控制：使两者高效、可靠地进行数据传送

四种传送方式

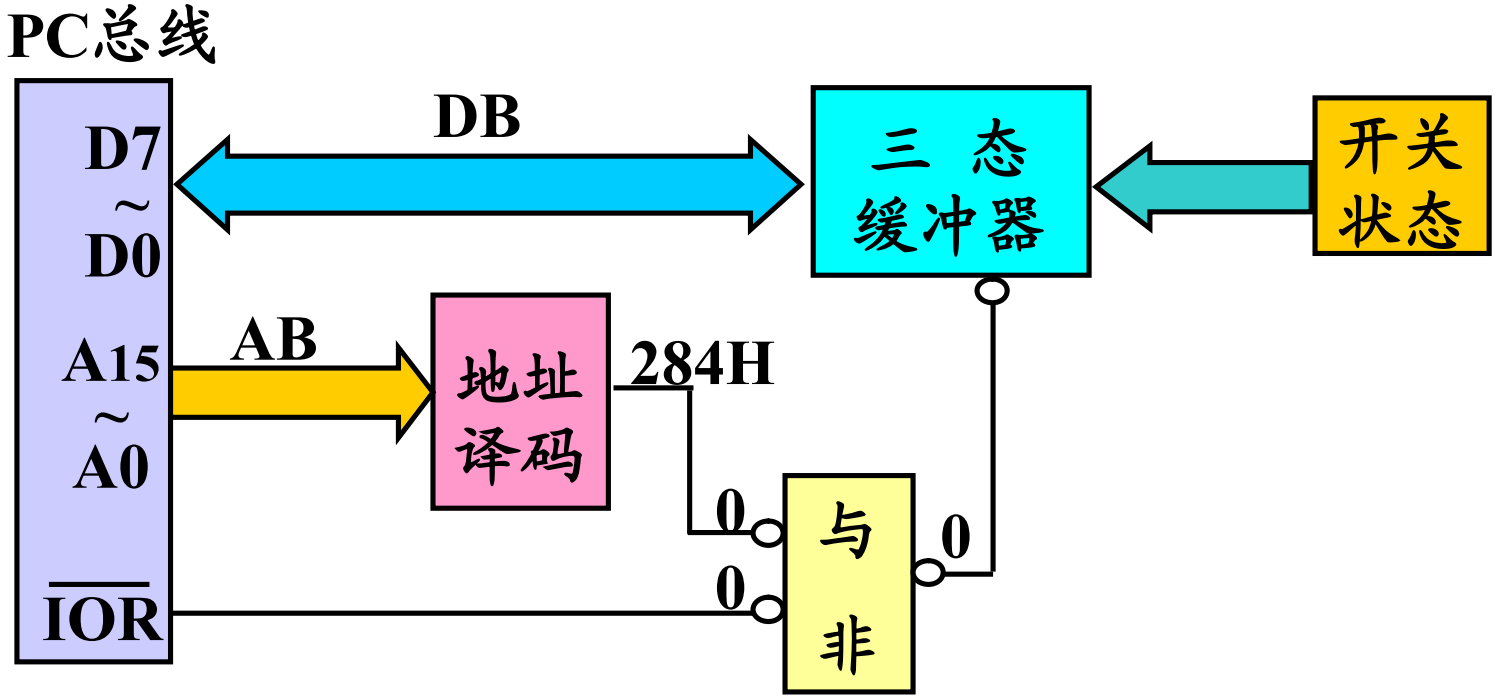
一、 无条件传送方式

二、 条件传送方式（查询方式）

三、 中断传送方式

四、 DMA传送方式（Direct Memory Access）

典型无条件传送方式接口电路（输入）



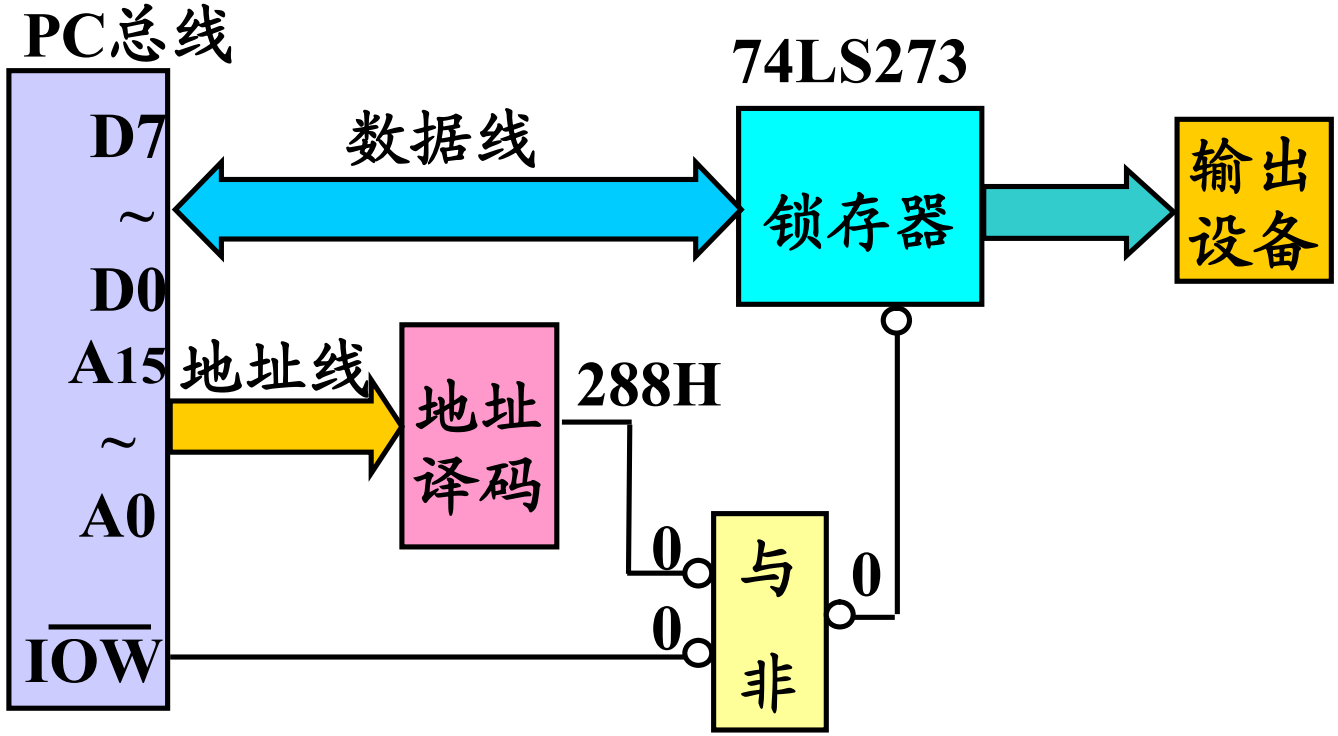
该电路在CPU执行指令

MOV DX, 284H

IN AL, DX

将输入设备的数据（开关状态）读入CPU内AL中

典型无条件传送方式接口电路（输出）



该电路在CPU执行指令

```
MOV  AL, 81H
MOV  DX, 288H
OUT  DX, AL
```

CPU内AL中的数据81H送至输出设备

无条件传送方式（同步传送方式）

- 实现方法

1、 CPU不查询外设工作状态，与外设速度的匹配通过在软件上延时完成，

2、在程序中直接用I/O指令，完成与外设的数据传送。

无条件传送方式（同步传送方式）

●特点

1. 适用于外设动作时间已知，

前提：CPU与外设进行数据传送时，外设保证已准备好。

2. 软硬件十分简单。

条件传送方式(查询传送方式)

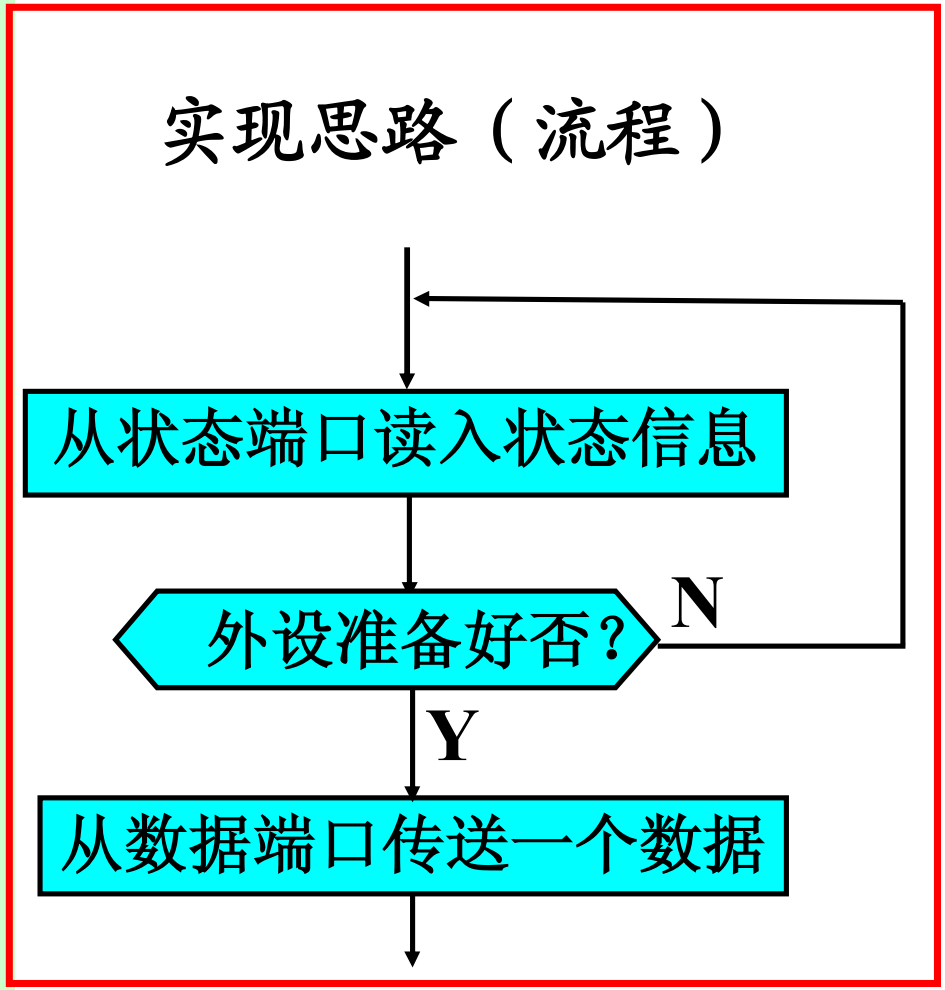
- **实现方法:**

  在与外设进行传送数据前，CPU先查询外设状态，

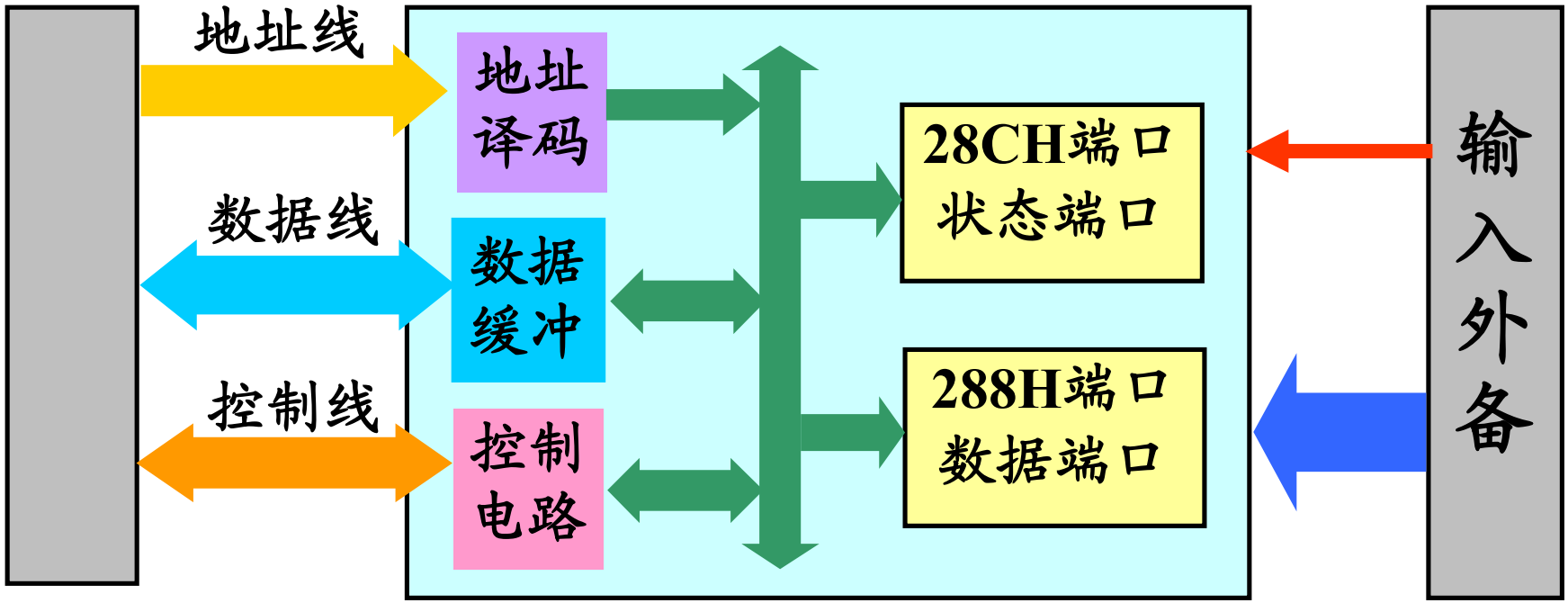
  当外设准备好后，才执行I/O指令，实现数据传送

- **特点:**

  1. CPU通过不断查询外设状态，实现与外设的速度匹配
  2. CPU的工作效率低



查询方式输入例



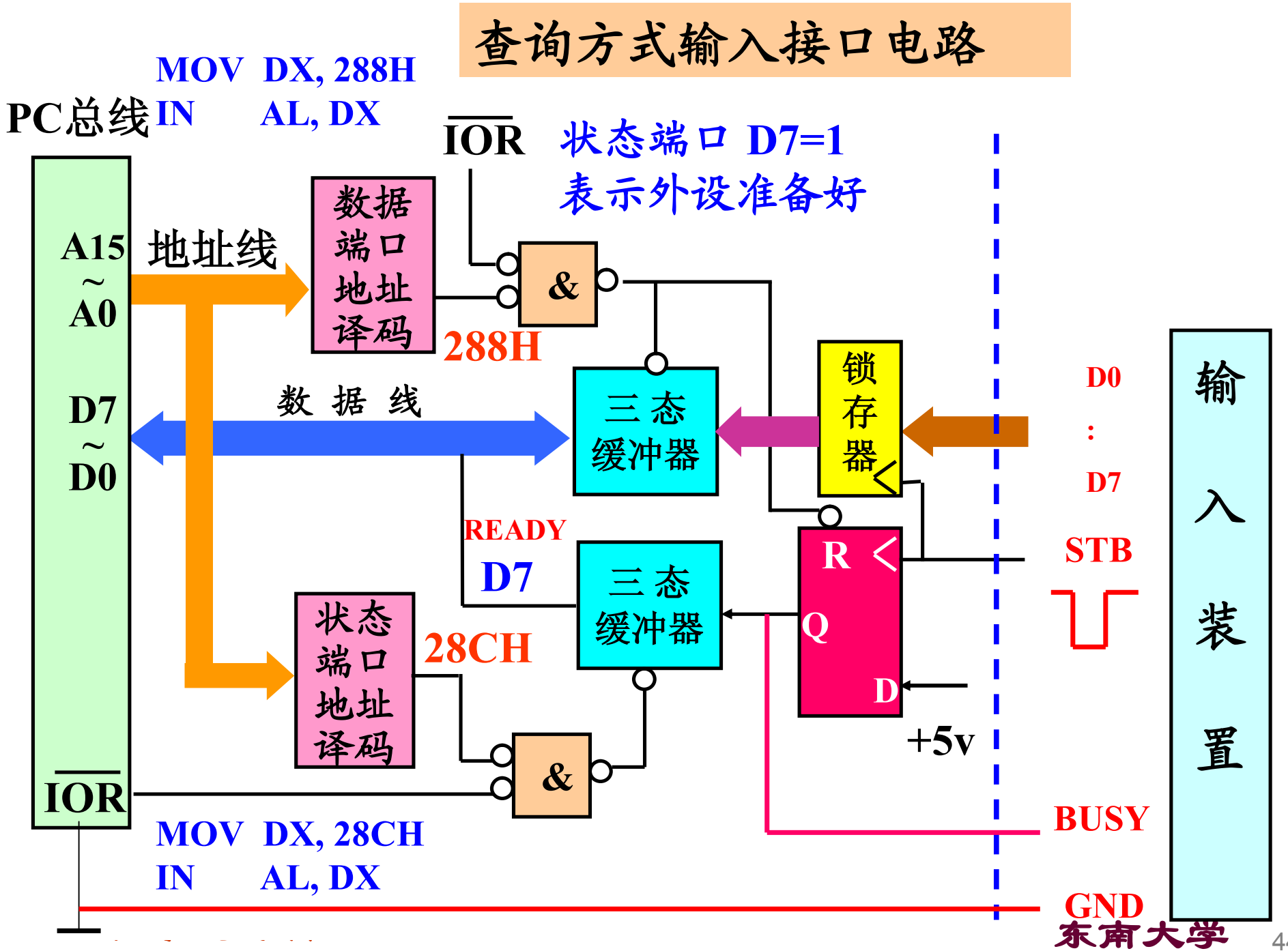
假设：　外设的状态端口为28C H，

其中D7=1时，表示外设数据准备好

外设的数据端口为288 H。

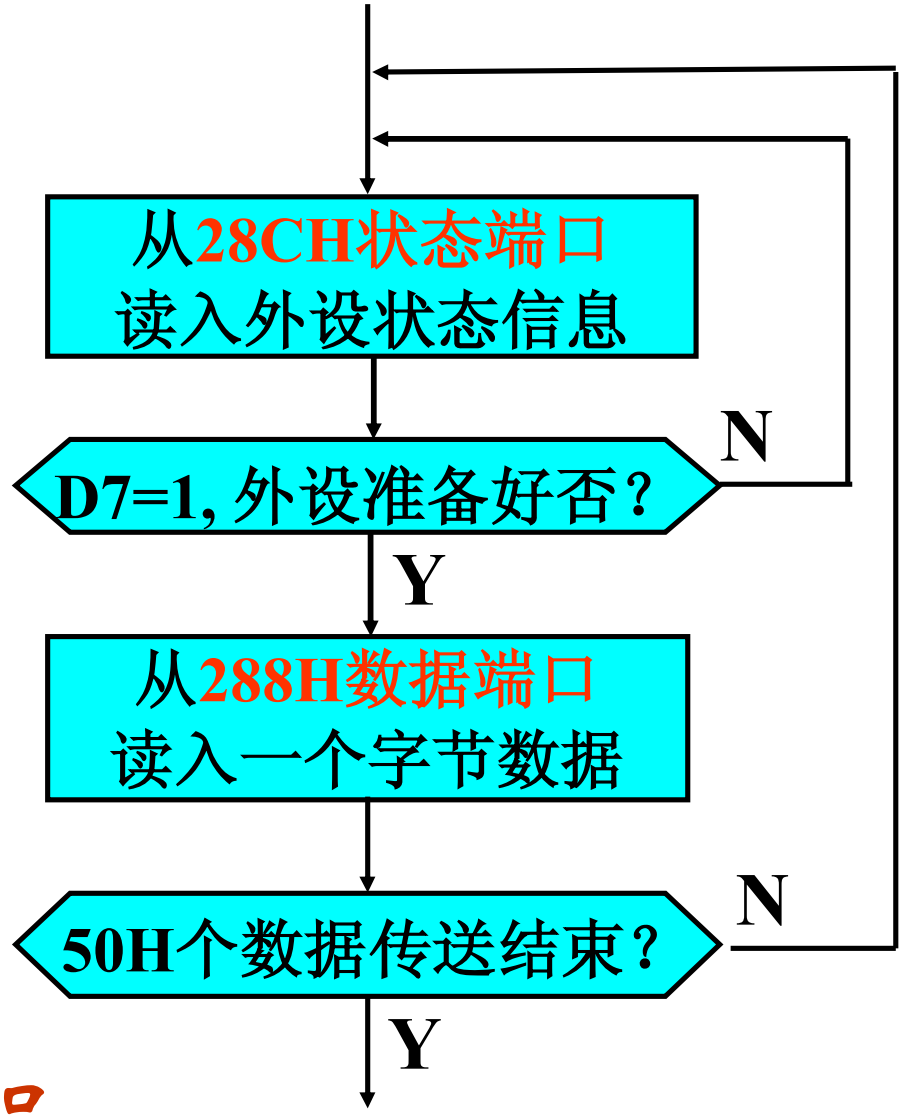
实现从外设读入50H个字节到内存缓冲区buffer中。

查询方式输入接口电路


MOV DX, 288H
IN AL, DX
PC总线
IOR
状态端口 D7=1
表示外设准备好
A15
~
A0
地址线
数据端口地址译码
&
288H
D7
~
D0
数 据 线
三 态缓冲器
锁存器
D0
:
D7
输入装置
READY
D7
三 态缓冲器
R
Q
D
STB
状态端口地址译码
28CH
&
+5v
IOR
BUSY
MOV DX, 28CH
IN AL, DX
GND

条件查询输入流程图

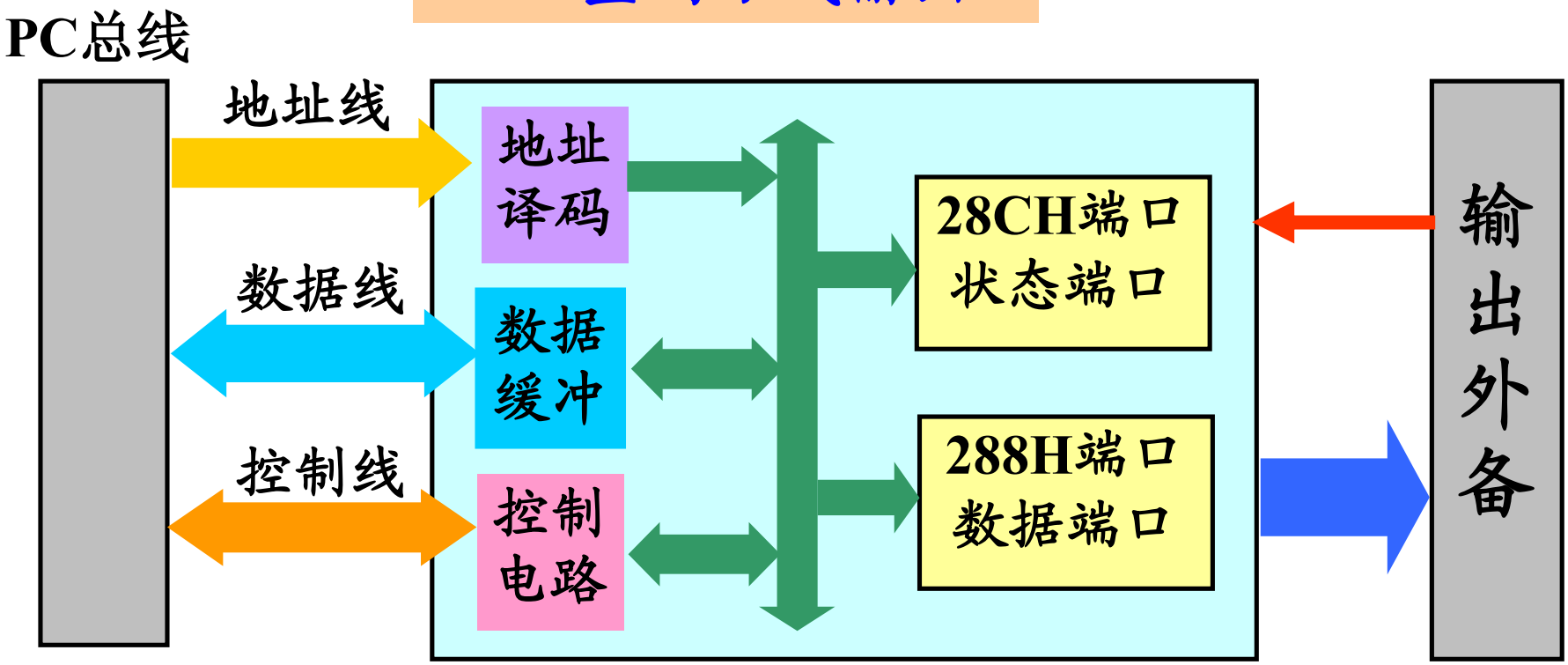
编程从外设读入50H个字节到内存缓冲区buffer中



查询方式输入程序片段

```
        MOV   AX,  SEG buffer   ;取缓冲区段地址
        MOV   DS,   AX
        LEA    DI,    buffer
        MOV   CX,   50H          ;传送个数
next:   MOV   DX, 28CH
ask:    IN      AL,  DX          ;从状态端口读入状态信息
        TEST   AL, 1000 0000B   ;80H, 检测D7位
        JZ       ask             ;D7=0,继续查询
        MOV   DX,  288H
        IN      AL,  DX          ;从数据端口读入数据
        MOV   [DI],  AL          ;送缓冲区
        INC     DI               ;修改缓冲区指针
        LOOP  next               ;传送下一个
        ….
```

查询方式输出



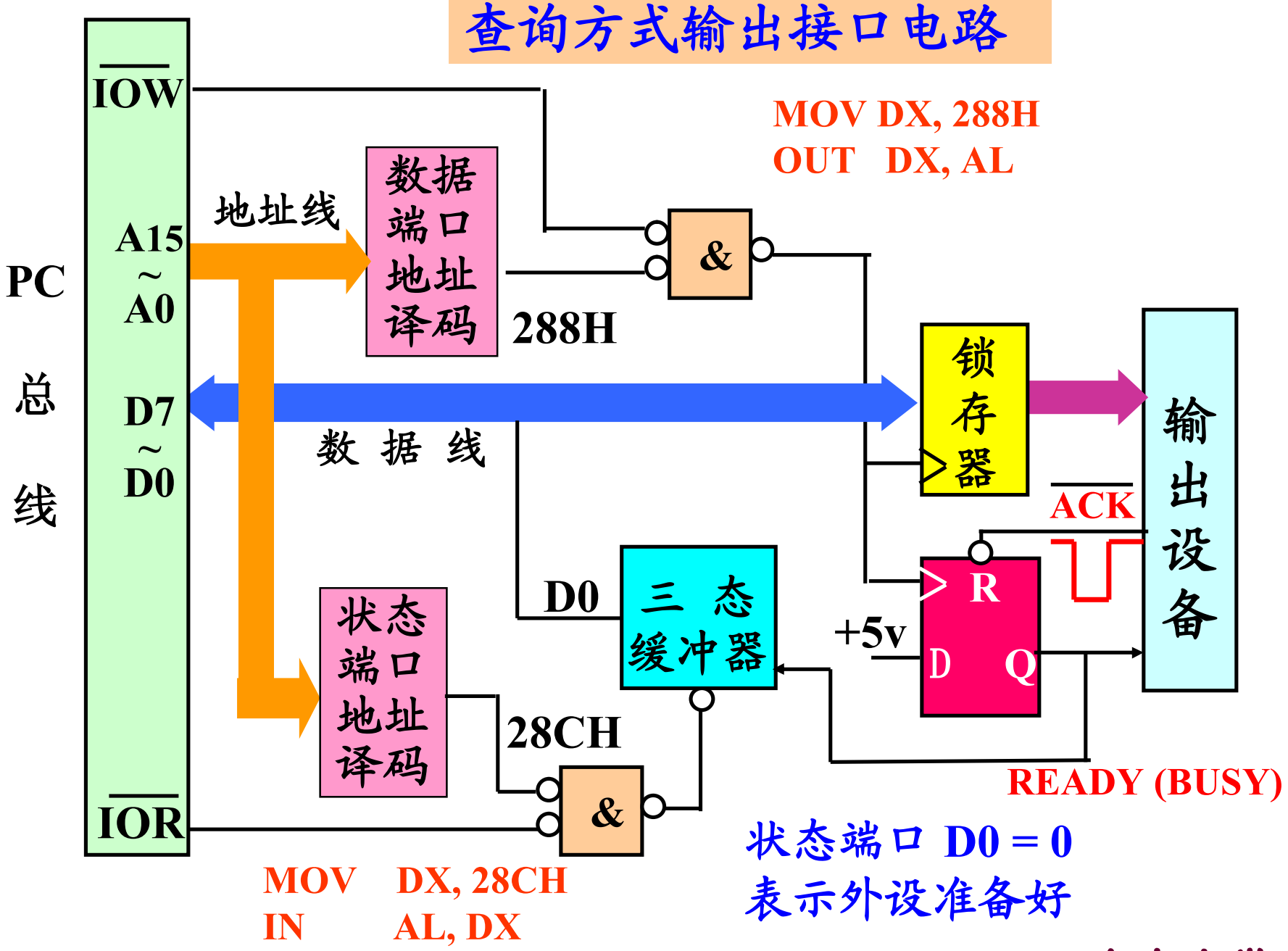
假设　外设的状态端口为28C H，

其中D0＝0时，表示外设准备好

外设的数据端口为288 H。

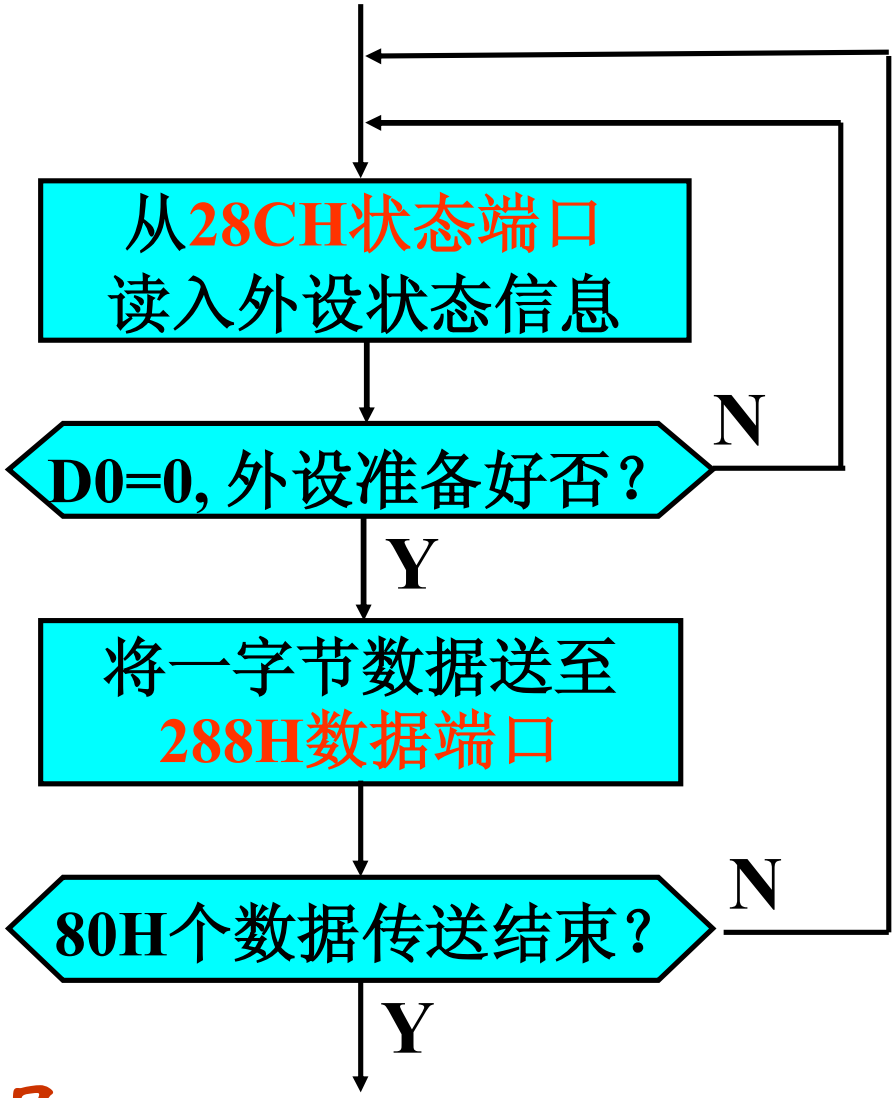
编程将缓冲区buffer的80H个字节输出到外设。

查询方式输出接口电路


PC 总 线
IOW
A15 ~ A0
D7 ~ D0
IOR
地址线
数据线
数据端口地址译码
288H
&
MOV DX, 288H
OUT DX, AL
锁存器
输出设备
ACK
R
D
Q
+5v
三 态缓冲器
D0
状态端口地址译码
28CH
&
READY (BUSY)
MOV DX, 28CH
IN AL, DX
状态端口 D0 = 0
表示外设准备好

条件查询输出流程图

编程将缓冲区buffer的80H个字节输出到外设



查询方式输出程序片断

```
        STATUS_PORT EQU  28CH
        DATA_PORT   EQU  288H
        MOV   AX,  SEG buffer   ;取缓冲区段地址
        MOV   DS,  AX
        LEA   SI,  buffer       ;取缓冲区首地址
        MOV   CX,  80H          ;传送个数
next:   MOV   DX,  STATUS_PORT
ask:    IN    AL,  DX           ;从状态端口读入状态信息
        TEST  AL,  0000 0001B   ;检测D0位
        JNZ   ask               ;D0≠0,继续查询
        MOV   AL,  [SI]         ;从缓冲区取数
        MOV   DX,  DATA_PORT
        OUT   DX,  AL           ;从数据端口输出数据
        INC   SI                ;修改缓冲区指针
        LOOP  next              ;输出下一个
        ....
```

CPU（总线）与外设间的数据传送方式

四种传送方式

一、 无条件传送方式

二、 条件传送方式(查询方式)

三、 中断传送方式

四、 DMA传送方式(Direct Memory Access)

DMA 传送方式(直接存储器存取方式)

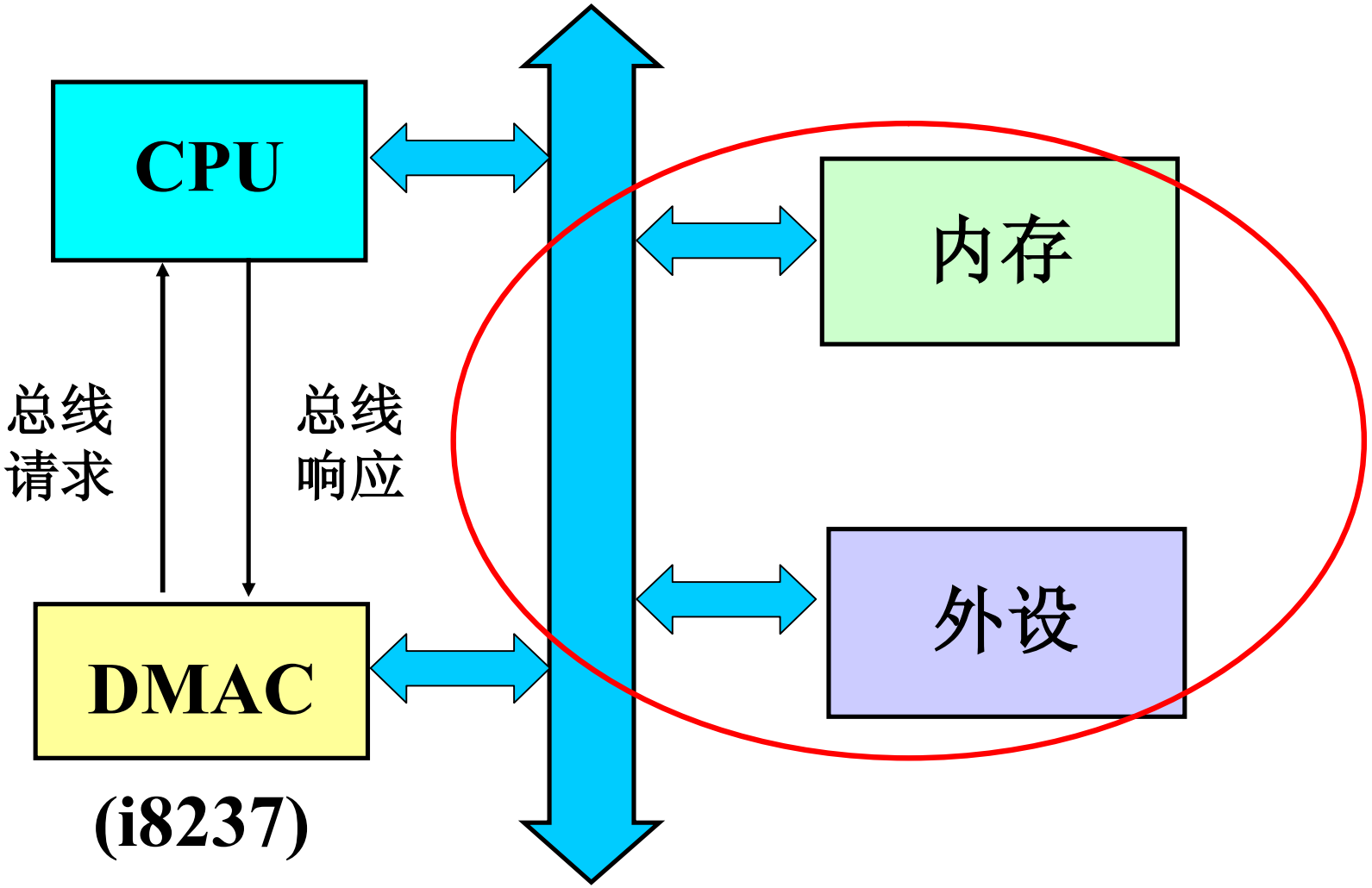
实现方法

1. 由专用接口芯片DMA控制器 (称DMAC) 控制传送过程,
2. 当外设需传送数据时，通过 DMAC向CPU发出总线请求;
3. CPU发出总线响应信号，释放总线;

——由CPU控制总线变为DMAC控制

4. DMAC接管总线，控制外设、内存之间直接数据传送

DMA 传送方式过程



DMA传送方式的特点

1. 外设和内存之间，直接进行数据传送，

不通过CPU, 传送效率高。

适用于在内存与高速外设、

或两个高速外设之间进行大批量数据传送。

2. 电路结构复杂，硬件开销较大。

本章后面详细介绍