



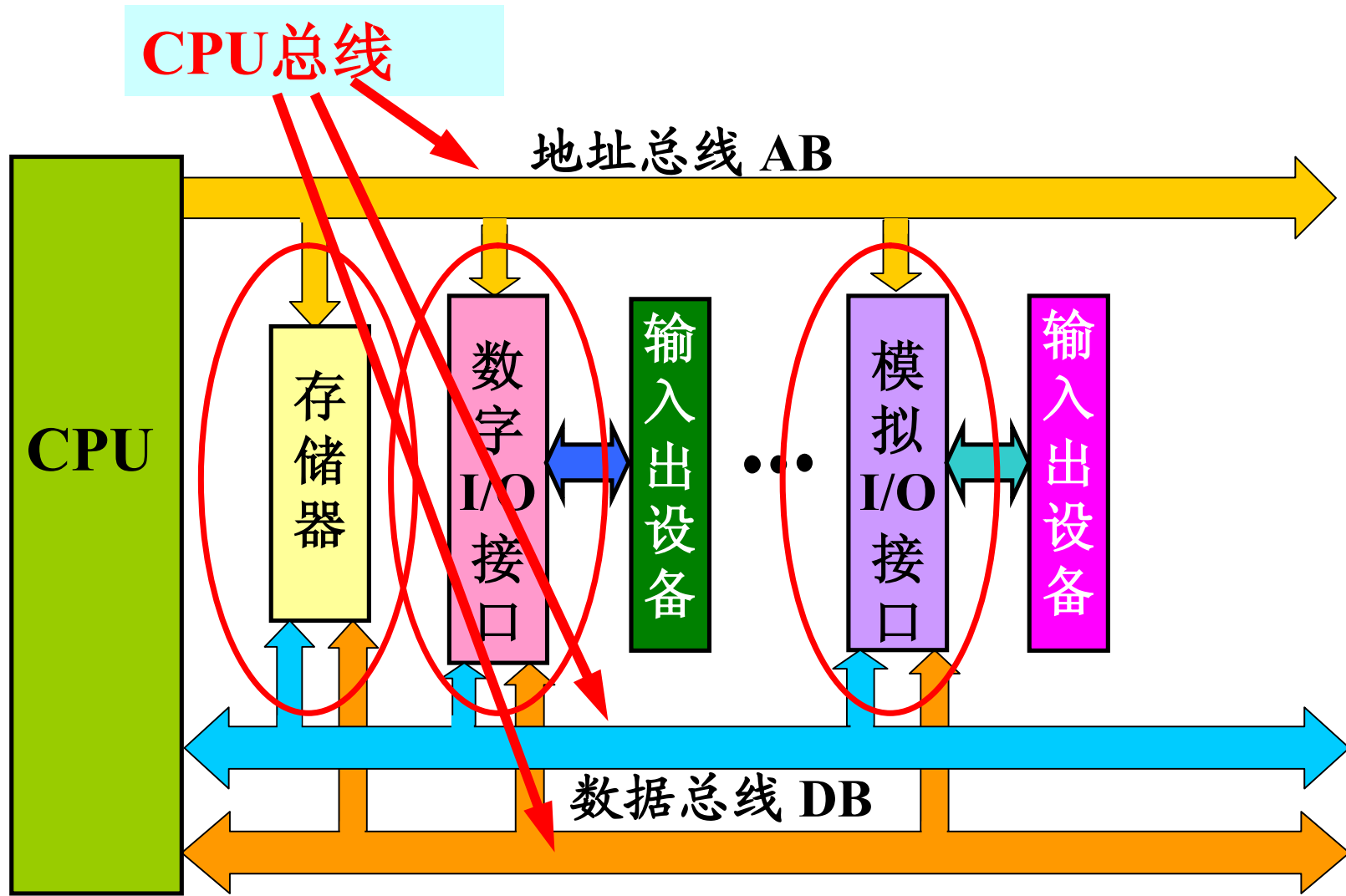
第五章 数字量输入输出接口

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





总线基本概念（续）

总线（BUS）

一组性质相似的各部件共用的信息线的集合。

分为：

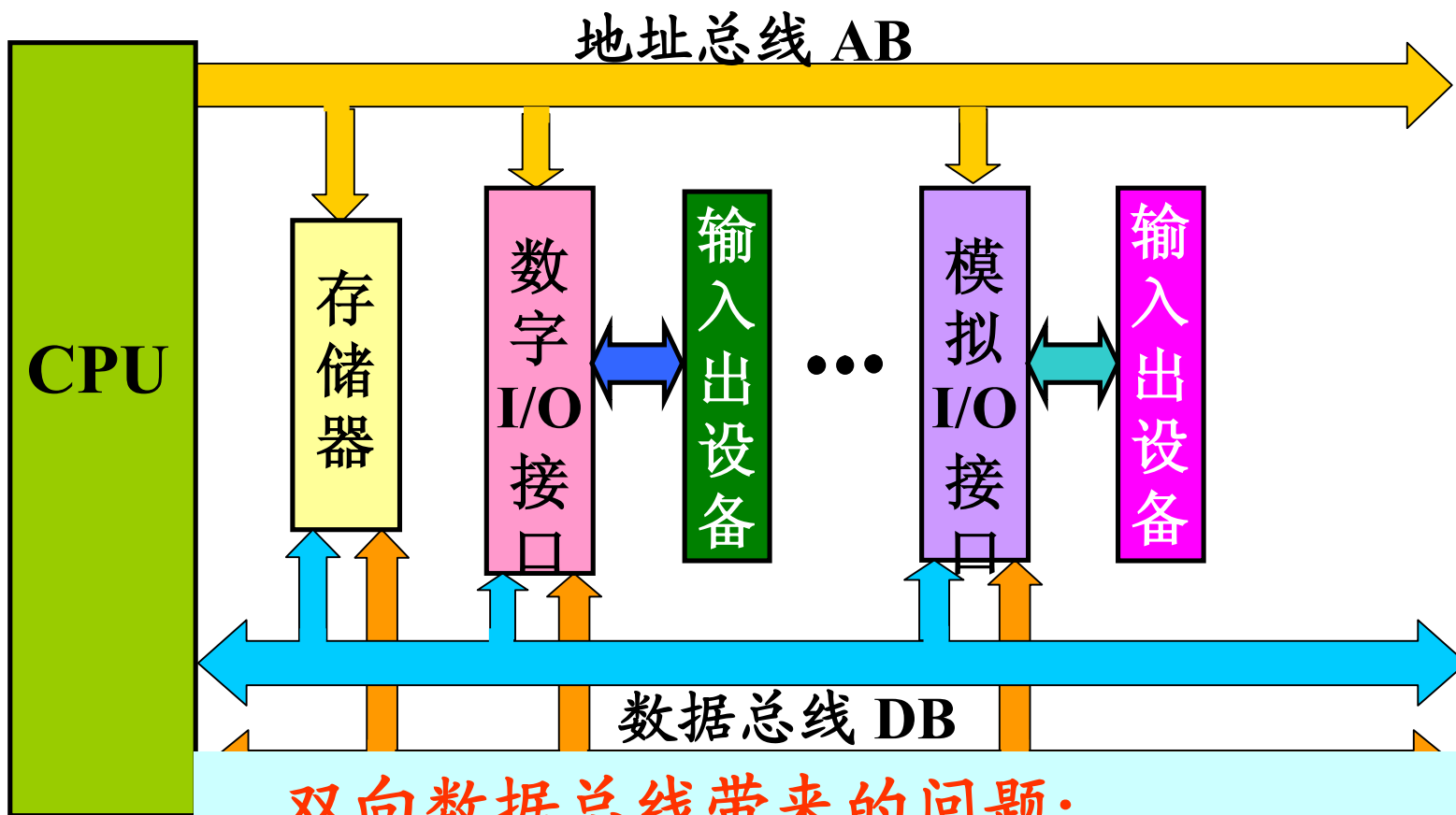
地址总线（AB）——单向

数据总线（DB）——双向

控制总线（CB）——每根线单向



总线基本概念 (续)



双向数据总线带来的问题：
多个输出挂在同一总线上，而正常情况下
BUS只能接通一组信息（数据）。
如何控制？

——三态缓冲器

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；

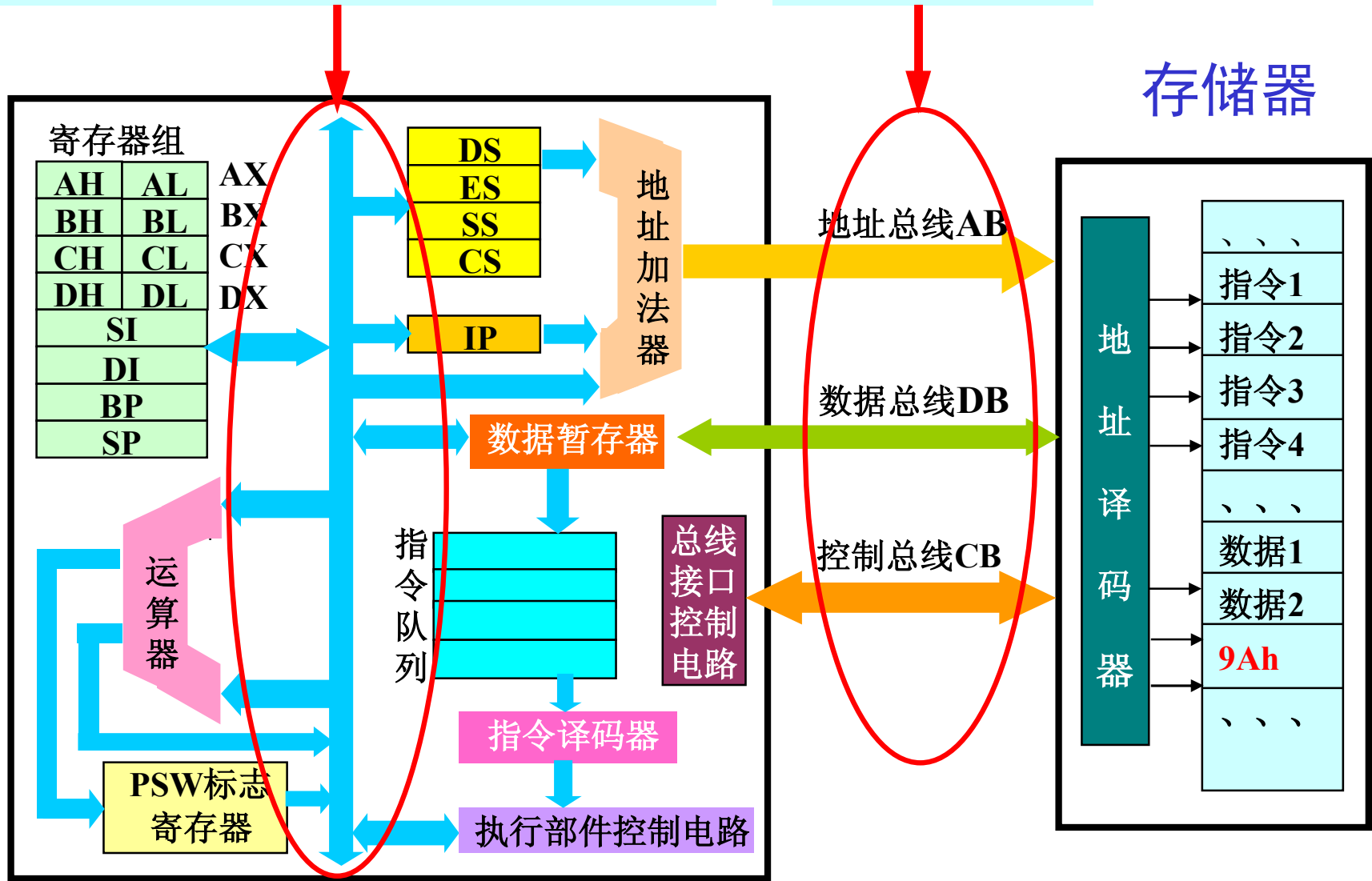
设备级总线

——计算机与外部设备之间的连接总线，
进一步使计算机真正成为开放体系



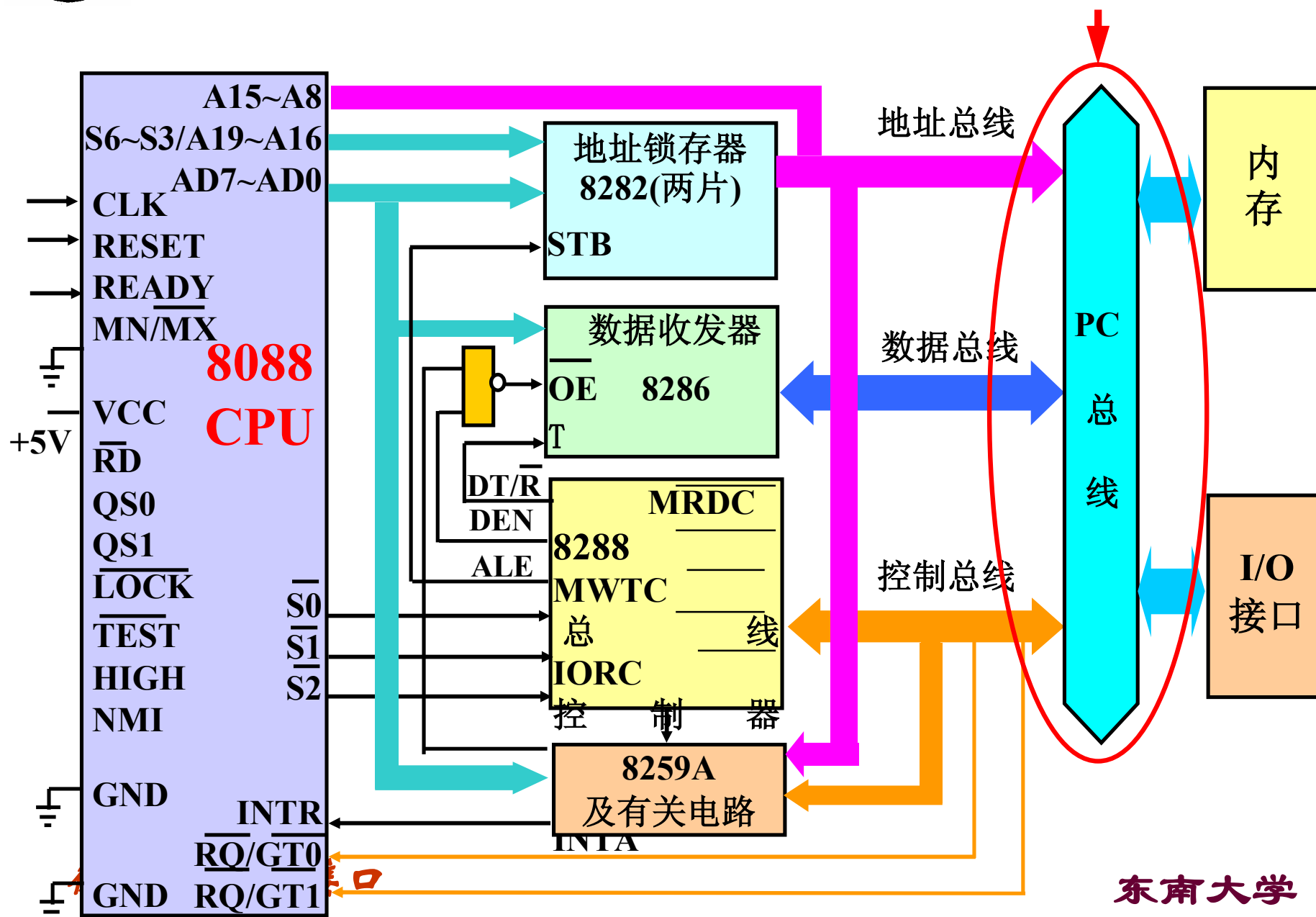
(8086 CPU) 片内总线

CPU总线





8088 (最大模式) 结构下的PC总线-系统总线



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；

设备级总线

——计算机与外部设备之间的连接总线，
进一步使计算机真正成为开放体系



系统总线——基本特性

同样分为:

地址总线 (AB) ——单向

数据总线 (DB) ——双向

控制总线 (CB) ——每根线单向

总线信号 (定义说明) ——P279

总线周期: 系统周期性操作

AB/DB/CB时序配合 (BUS Cycle time)。

(指令: 取指令、译码、执行操作 == 指令周期)

CPU控制: MEM读/写; I/O读写 (T_w)。

DMAC: MEM-MEM; MEM-I/O

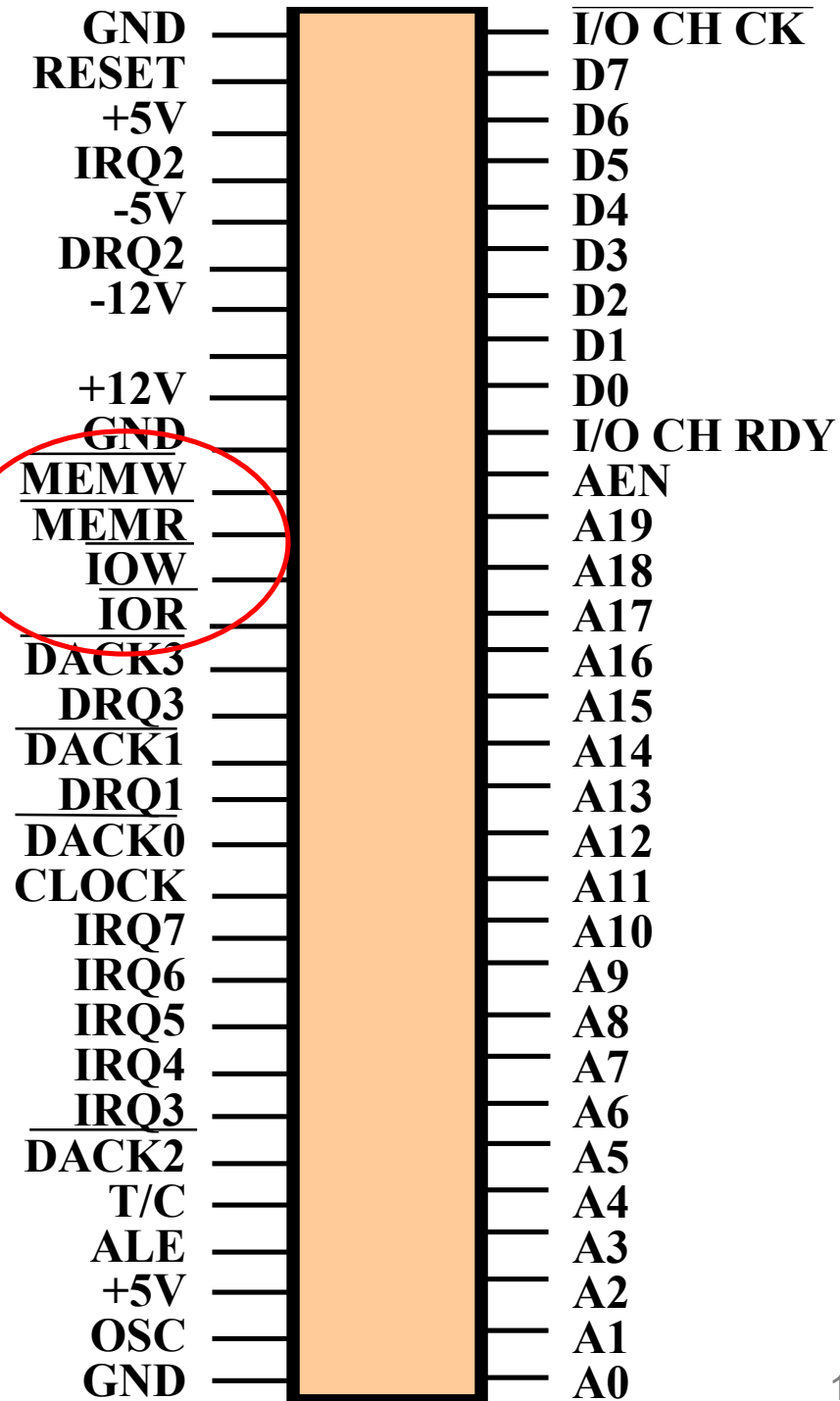
机电特性 (机械尺寸、电平、时序等)



系统 (PC) 总线标准

PC/XT总线

—— 62线 (8位数据总线)





系统总线（PC总线）标准（续）

PC/AT—ISA总线（**Industry Standard Architecture**）

62线与PC/XT总线兼容；+36线扩展部分，
（8位/16位数据总线）

EISA总线（**Extended Industry Standard Architecture**）

98线与PC/AT—ISA总线兼容；
插座加深，二层插芯，+100线插芯EISA专用信号，
（32位地址总线，寻址4G，16MB/s）

PC /104总线

与PC总线兼容，主要是小板结构，适用于工业控制



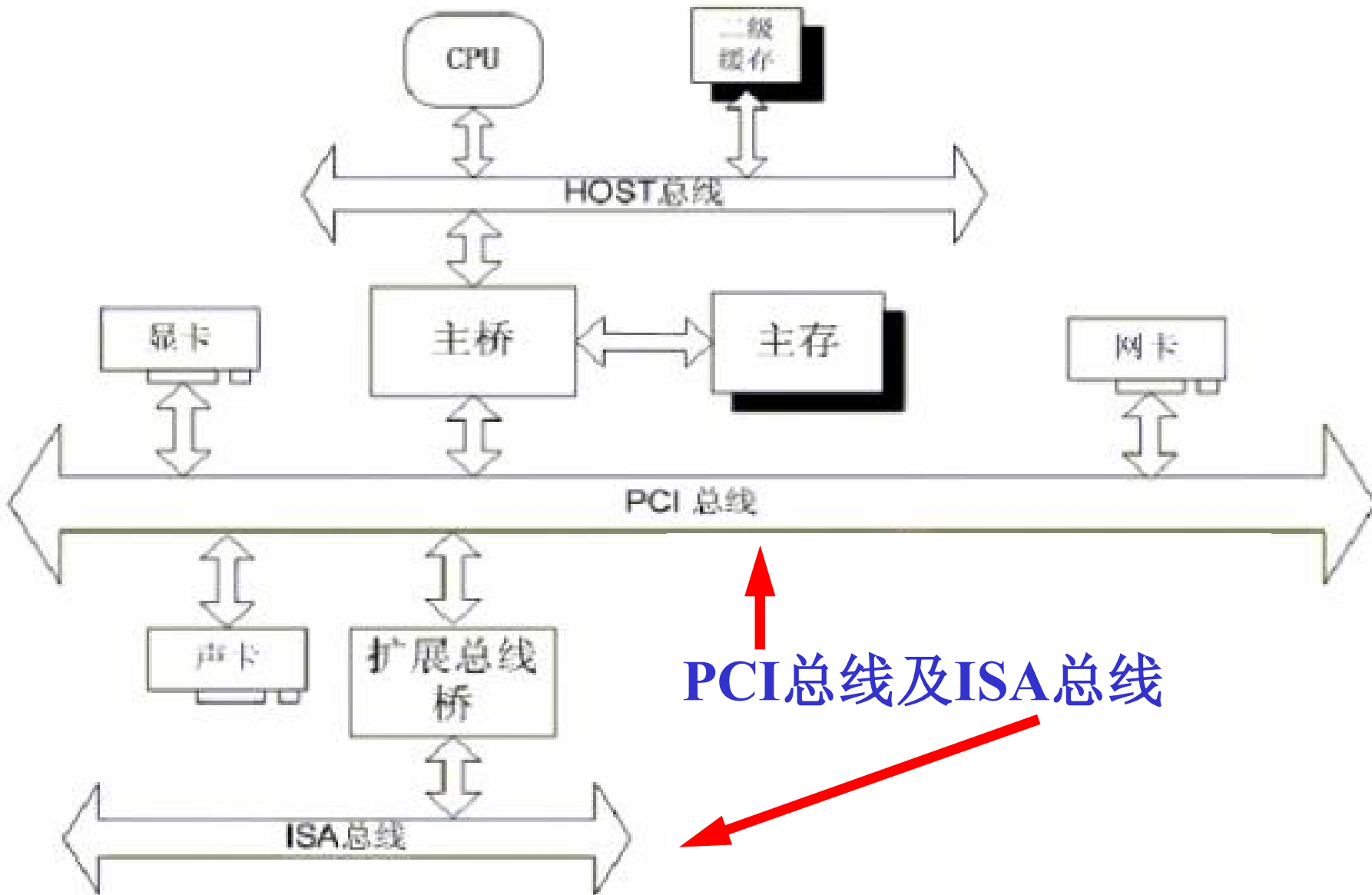
系统总线（PC总线）标准（续）

PCI总线(Peripheral Component Interconnect)

47脚单/49主控+51脚可选，

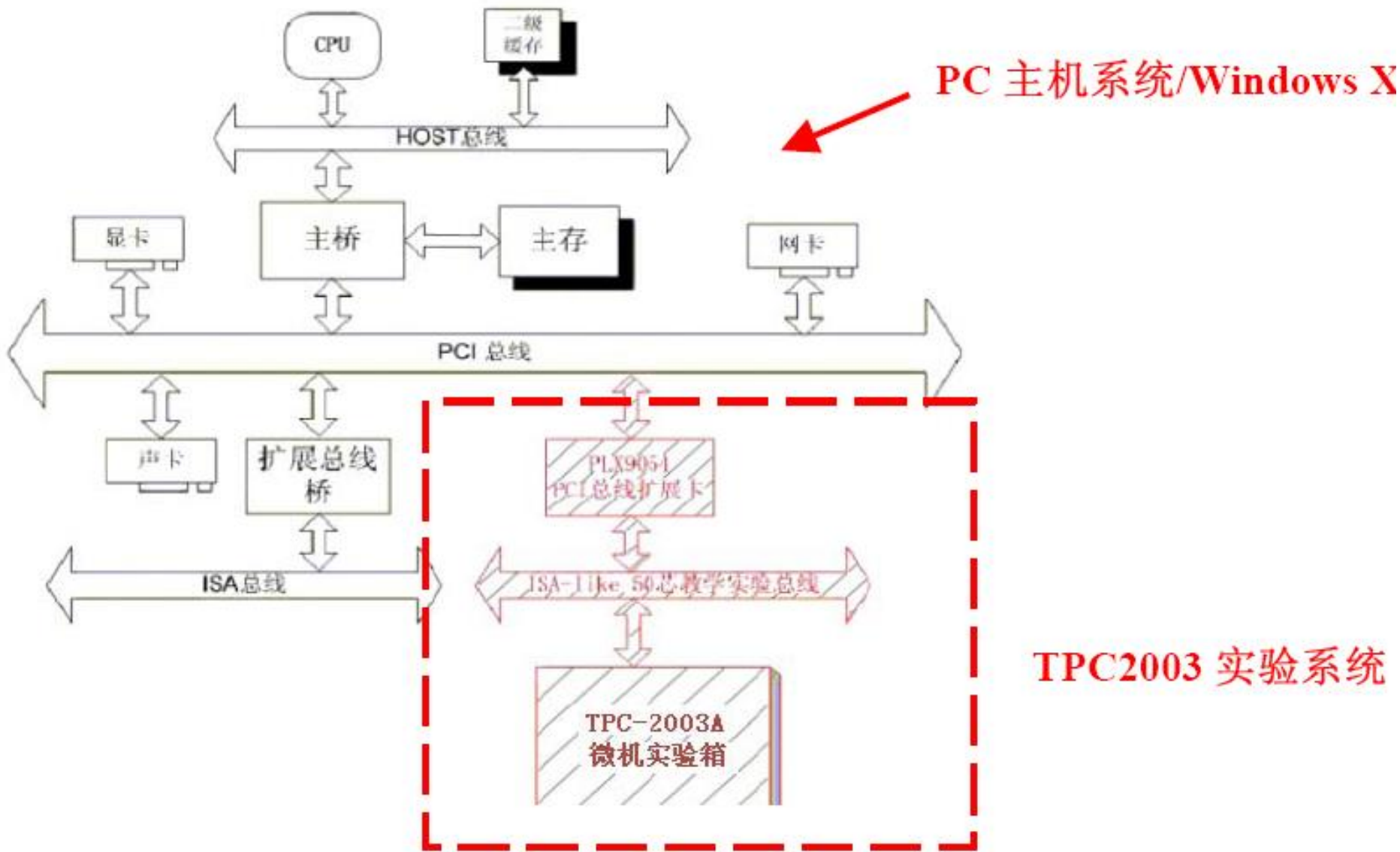
（32位/64位数据总线，数据、地址线复用，
速率：32位时132MB/s，62位时267MB/s）

即插即用（Plug & Play）功能



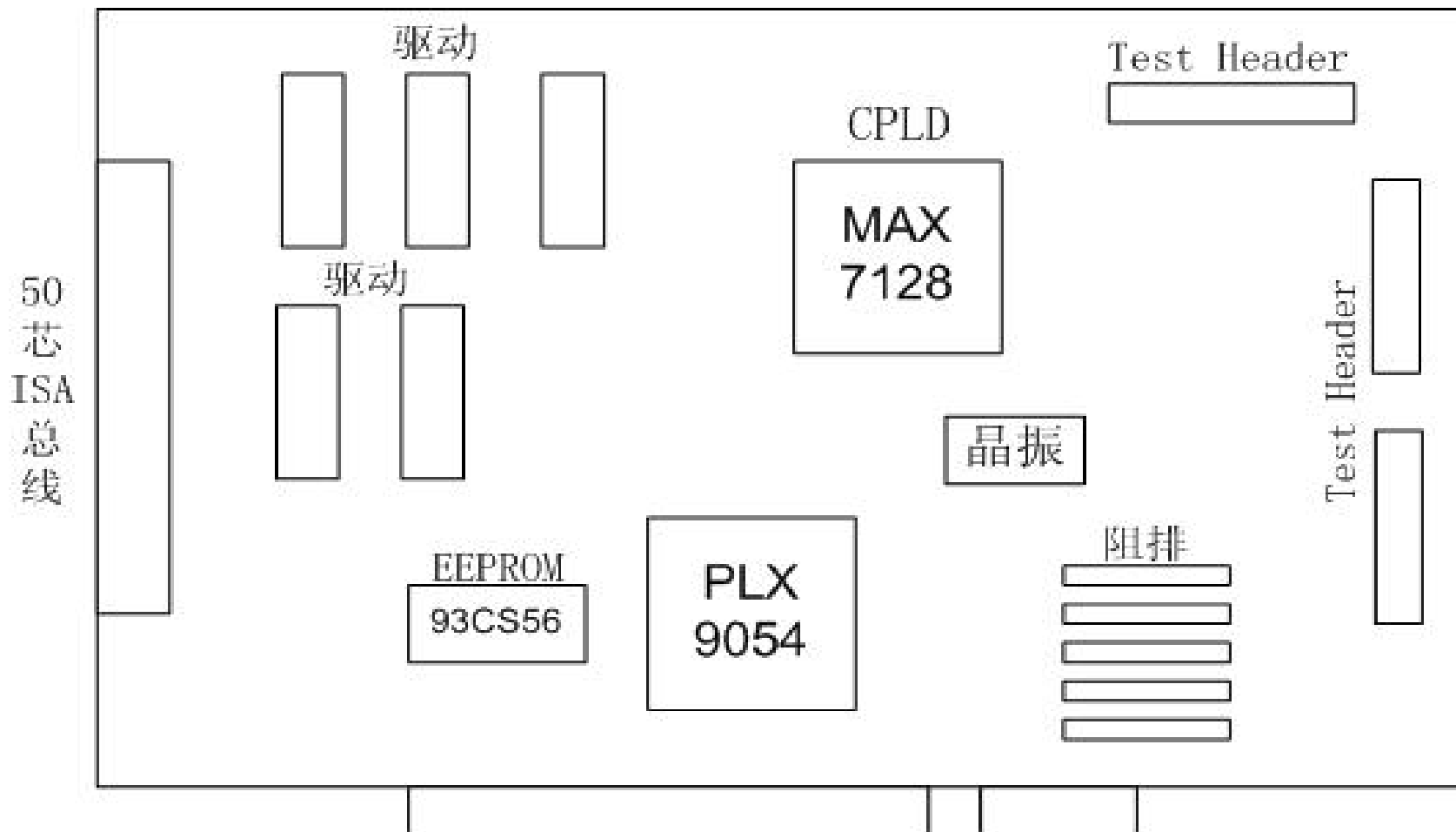


TPC2003A实验装置



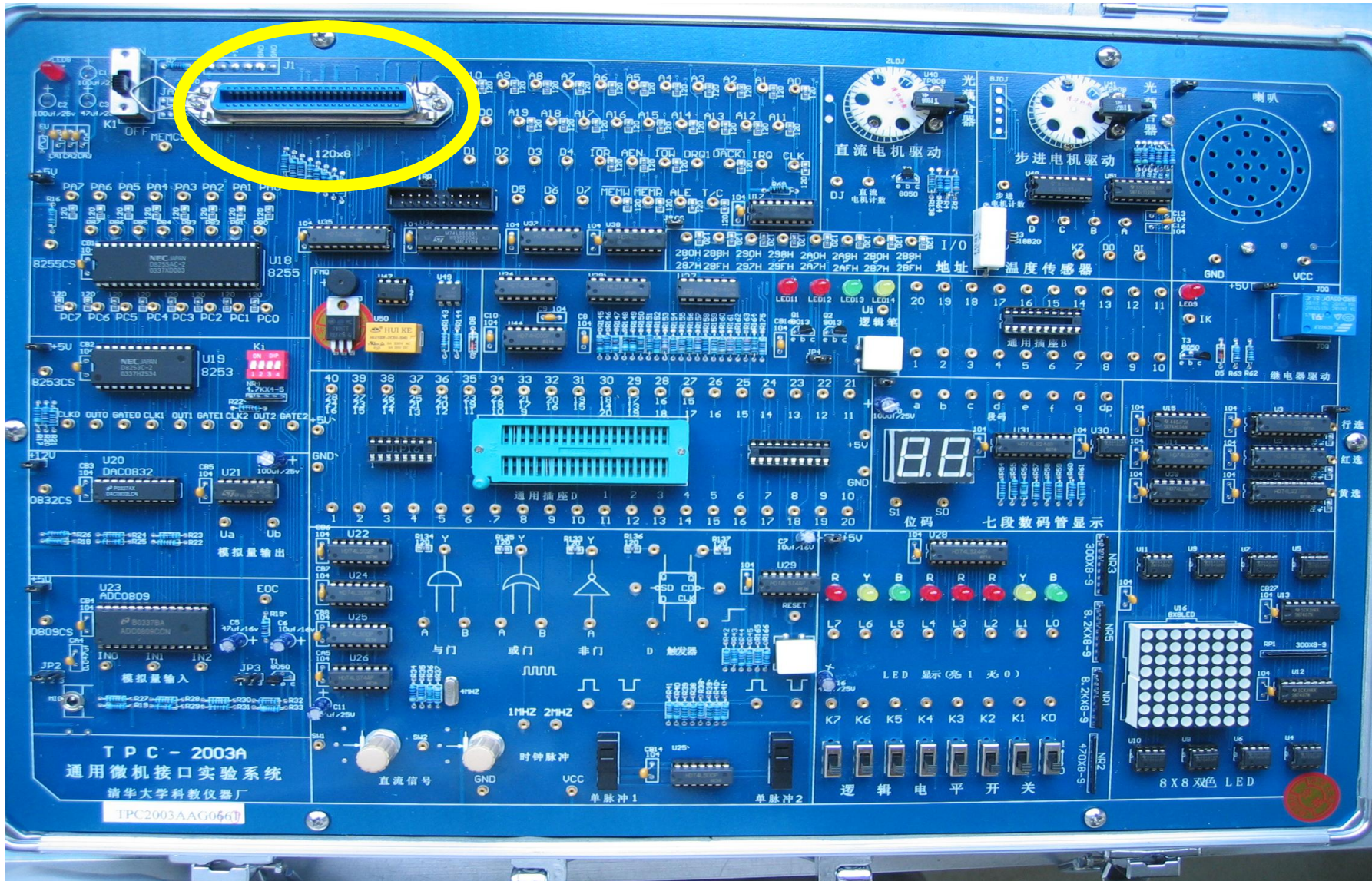


TPC2003A 实验装置—PCI总线延伸卡





TPC2003A 实验装置—接口实验箱





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



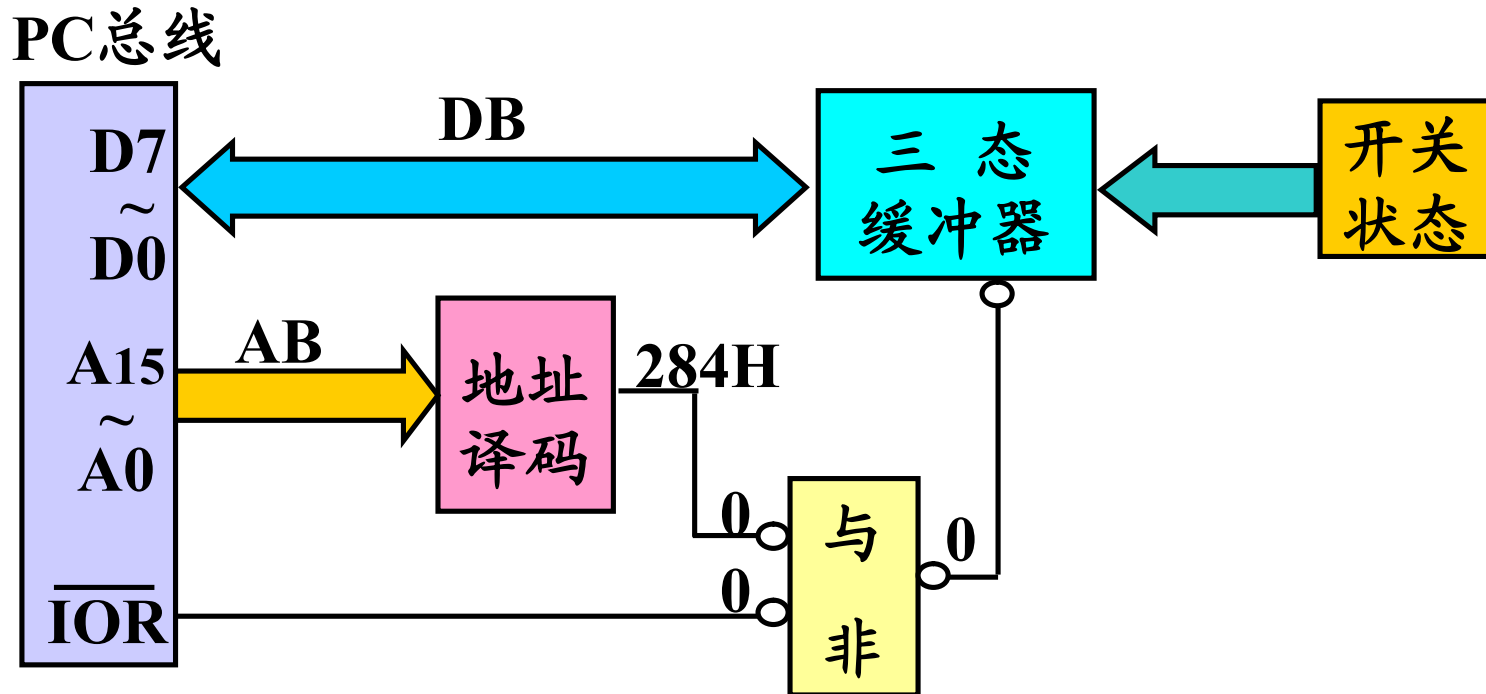
CPU（总线）与外设间的数据传送方式

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（**D**irect **M**emory **A**ccess）



无条件传送方式接口电路（输入）



该电路在CPU执行指令

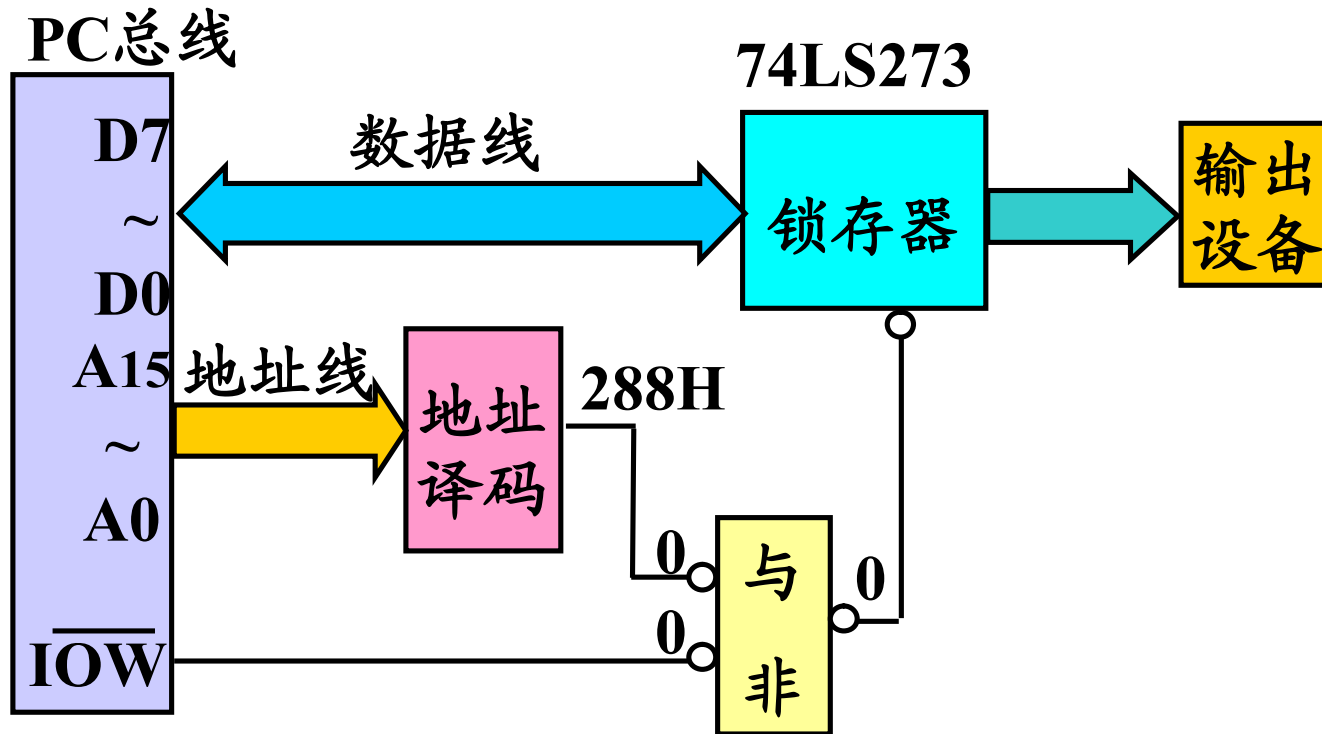
```
MOV DX, 284H
```

```
IN AL, DX
```

将输入设备的数据（开关状态）读入CPU内AL中



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)

● 特点

1. CPU不查询外设工作状态，
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

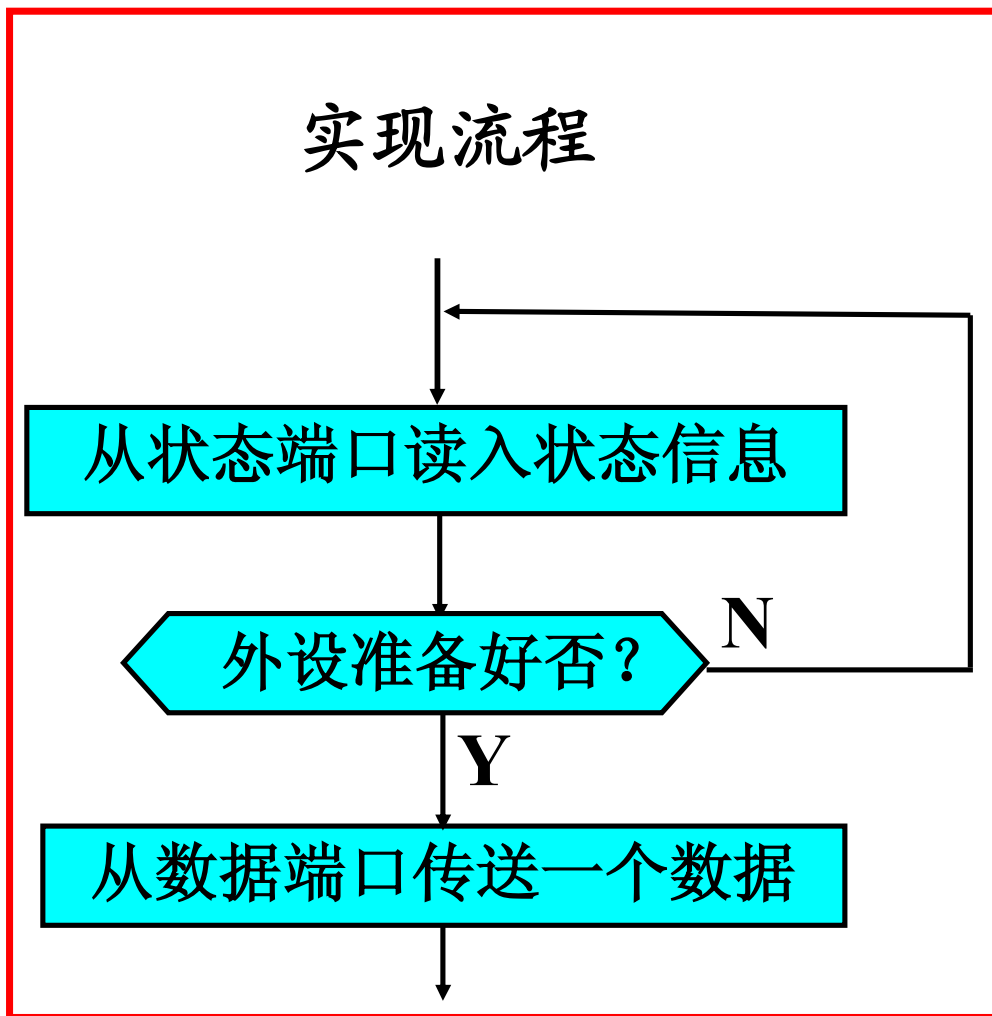


条件传送方式(查询传送方式)

● 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

实现流程

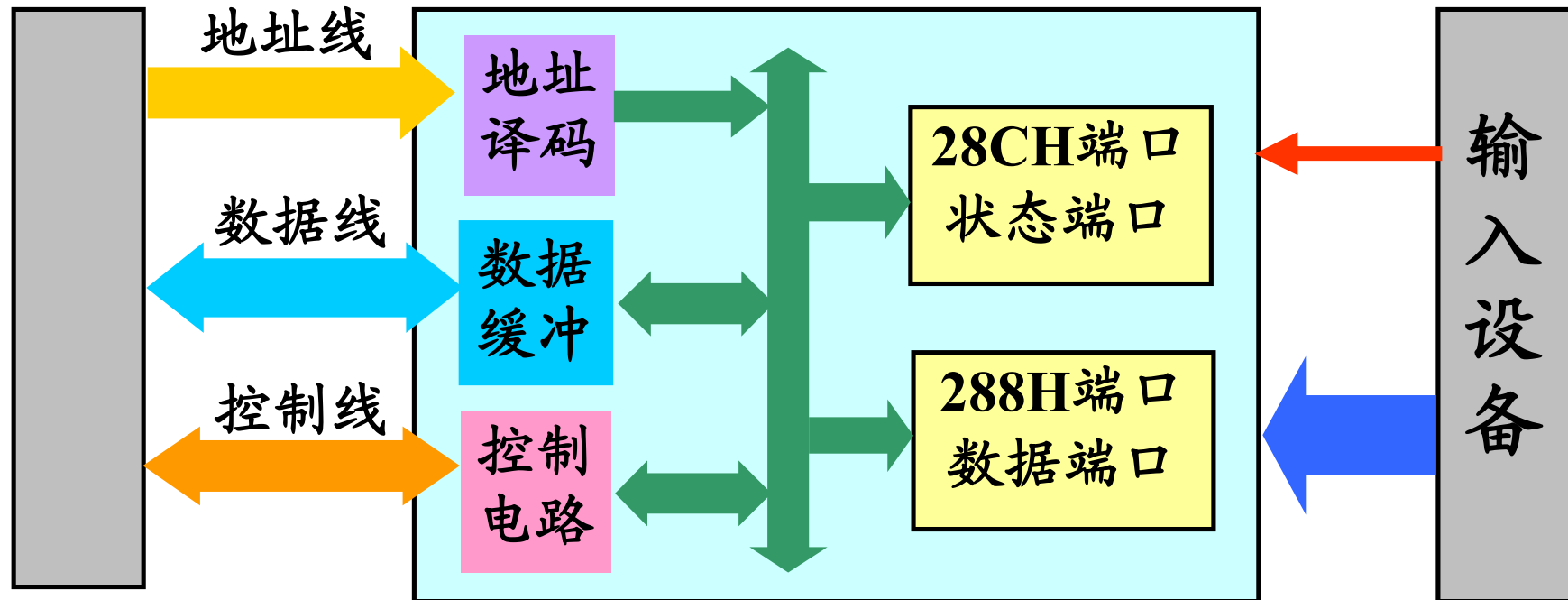




查询方式输入例

PC总线

外设



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。

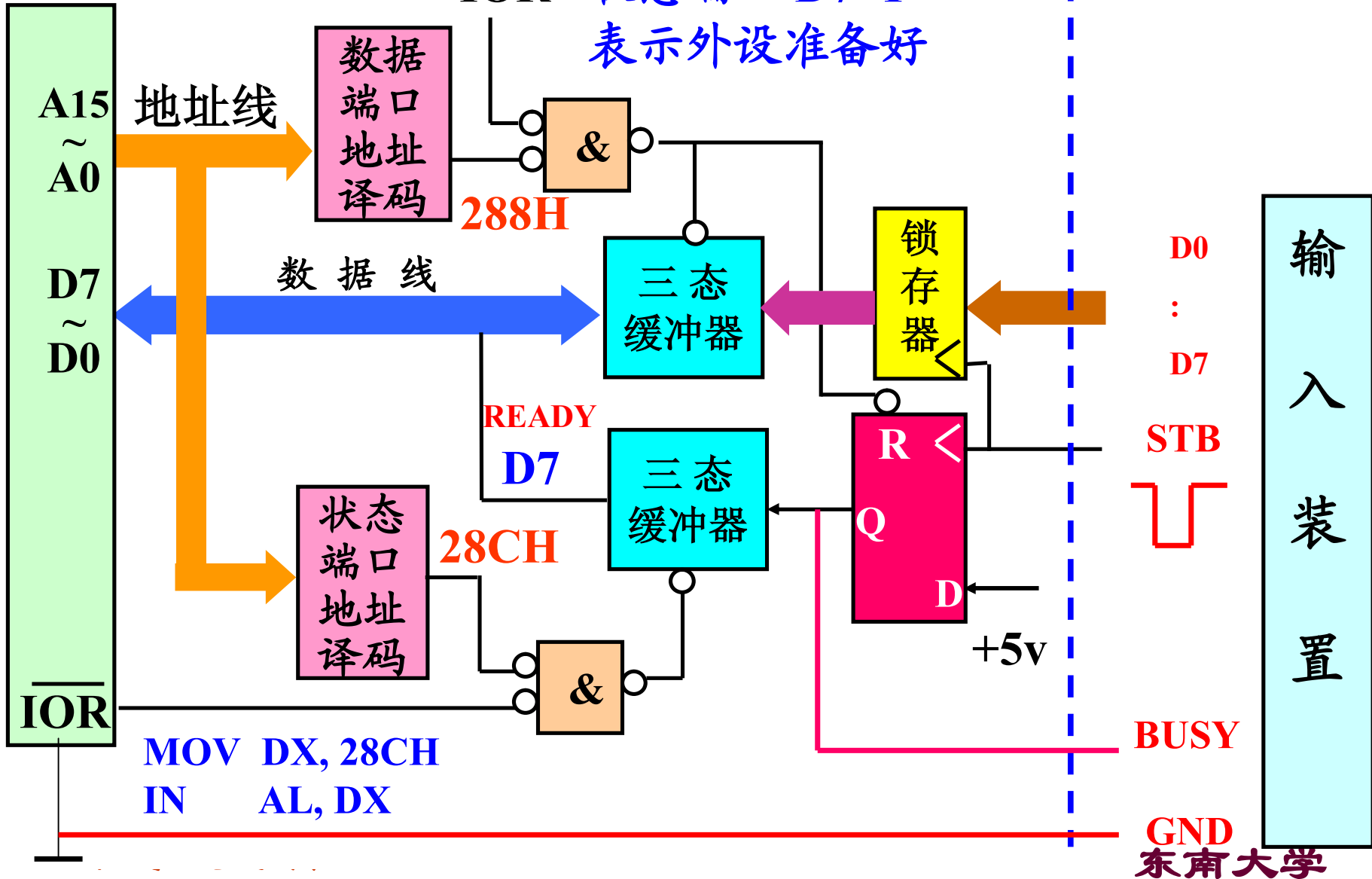


查询方式输入接口电路

MOV DX, 288H

PC总线 IN AL, DX

\overline{IOR} 状态端口 D7=1
表示外设准备好



MOV DX, 28CH
IN AL, DX



条件传送方式 (查询传送方式)

严重缺点:

CPU不断查询外设状态, 工作效率低

无条件传送方式 (同步传送方式)

严重缺点

CPU与外设必须同步, 否则出错!



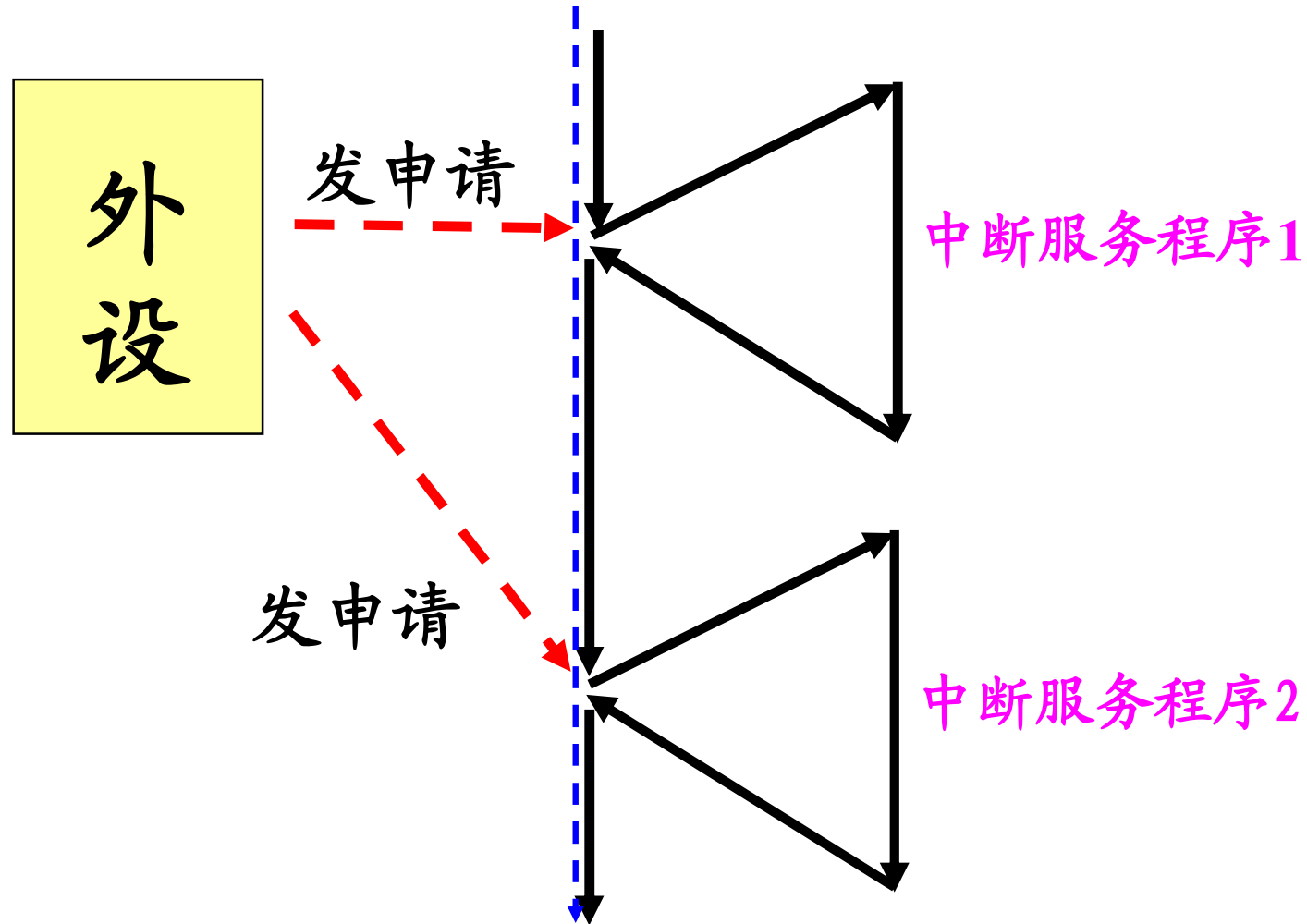
CPU（总线）与外设间的数据传送方式

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（**D**irect **M**emory **A**ccess）



中断方式下CPU执行程序流程



中断概念: CPU与外设交换信息的一种方式

==》硬件手段, 改变CPU执行程序的顺序 (程序流)



中断传送方式——具体过程

1. 当外设准备好，向CPU发出**中断请求**
2. CPU在满足响应中断条件下，发出**中断响应信号**；
3. CPU暂停当前的程序，转去执行**中断服务程序**，
完成与外设的数据传送；
4. CPU从**中断服务程序返回**，继续执行被中断的程序



中断传送方式的特点

1. CPU和外设大部分时间处在并行工作状态,

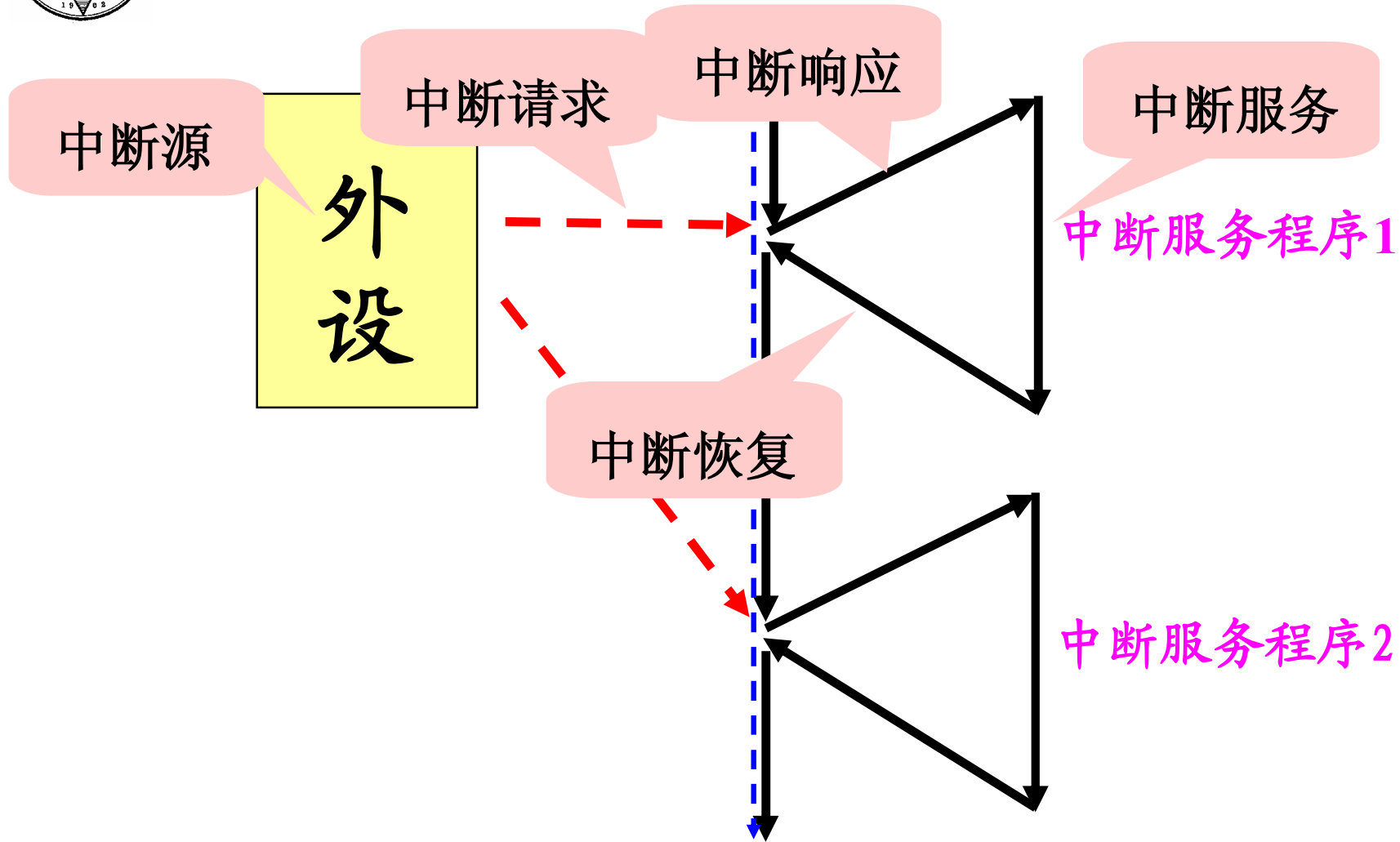
只在CPU响应外设的中断申请后,

进入I/O数据传送的过程

2. 中断传送方式极大地提高了CPU的效率



中断实现与控制



中断实现:

中断源---中断请求---中断响应---中断服务---中断恢复



8088 中断结构

• **软件中断——指令执行结果**

• 除零，单步，溢出

• 软件中断调用 (INT xxH)

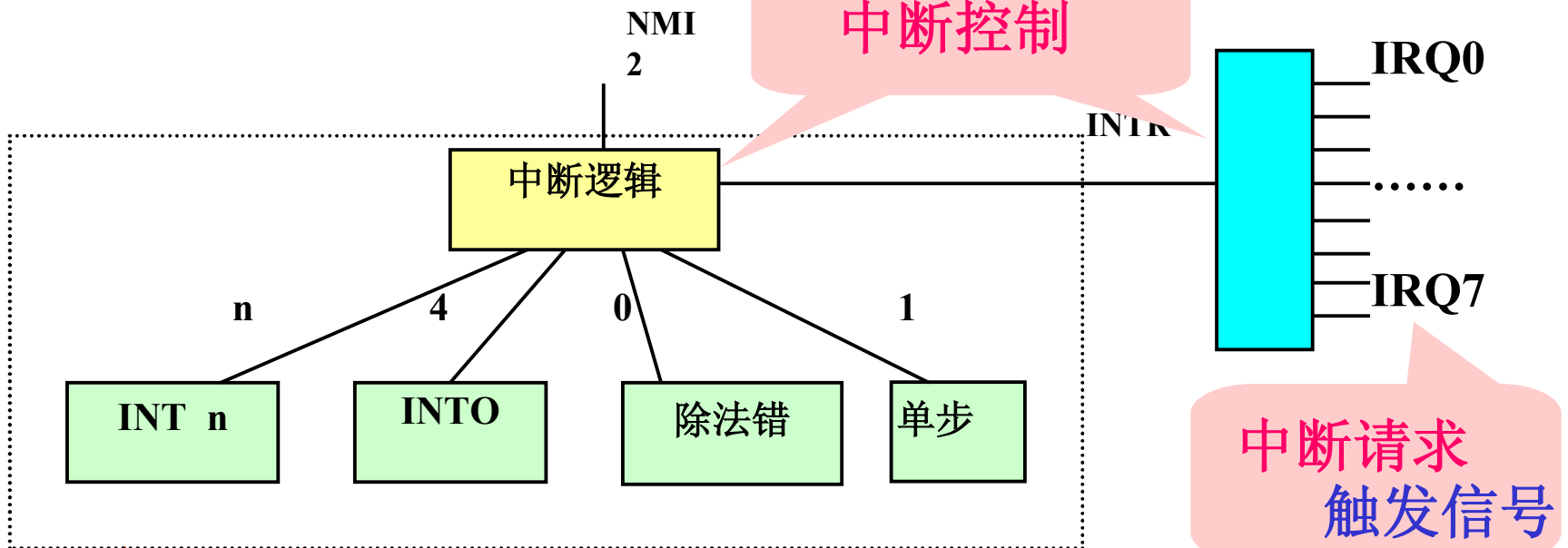
• **硬件中断——非指令执行**

• 非屏蔽中断NMI(Non-Maskable Interrupt)

• 可屏蔽中断INTR (IF=1)

内部中断

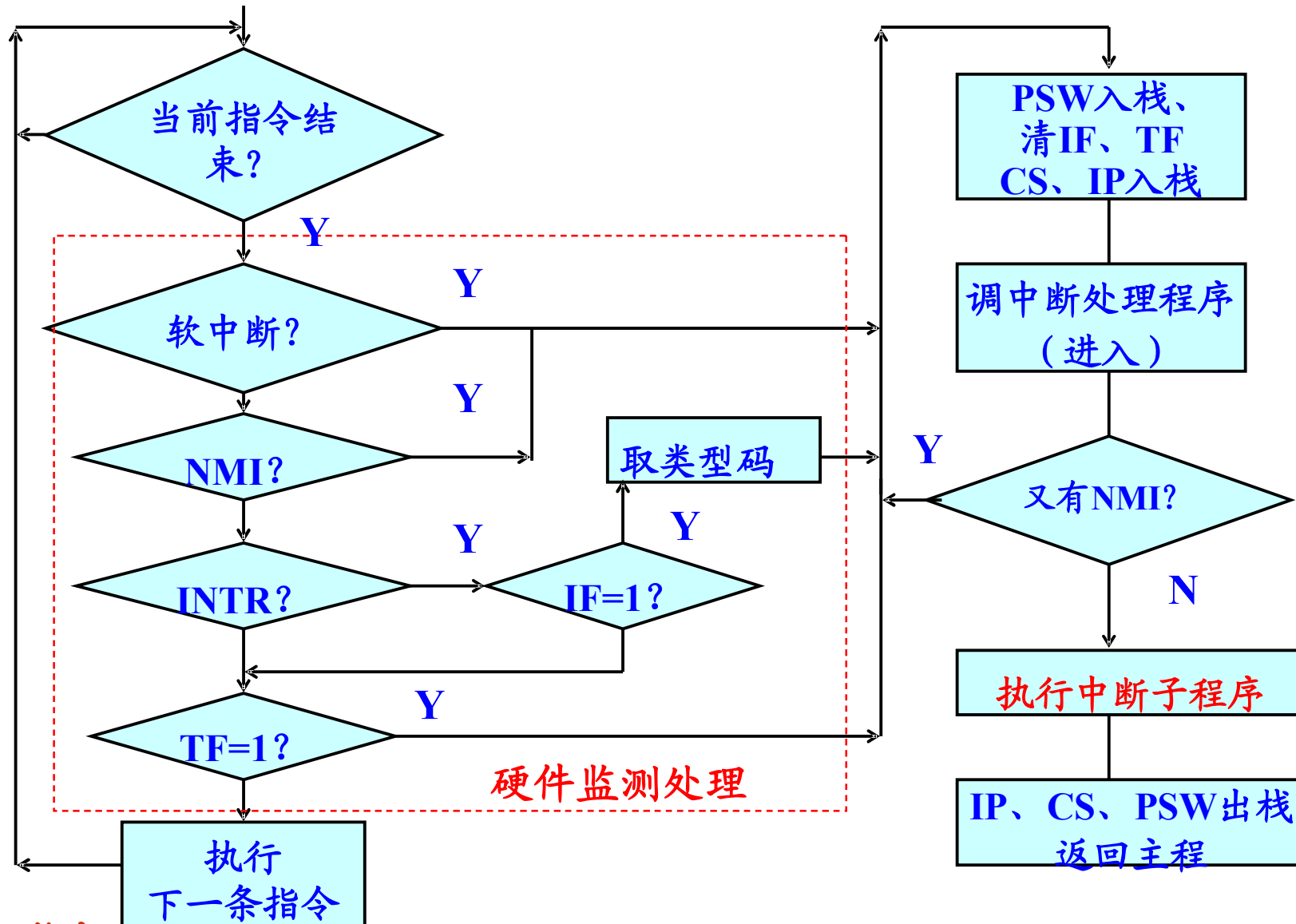
外部中断





CPU完整的中断响应和处理流程 (忽略单步中断处理)

详参考书P296





中断指令 Interrupt

INT n ; 软中断, n为中断类型号 ——形成中断向量

$(SP) \leftarrow (SP) - 2, ((SP) + 1, (SP)) \leftarrow (FLAGS);$

$(IF) \leftarrow 0 \quad (TF) \leftarrow 0$

$(SP) \leftarrow (SP) - 2, ((SP) + 1, (SP)) \leftarrow (CS); (CS) \leftarrow [0:n*4+2]$

$(SP) \leftarrow (SP) - 2, ((SP) + 1, (SP)) \leftarrow (IP); (IP) \leftarrow [0:n*4]$

IRET(Interrupt Return) ; 中断服务程序中最后一语句

$(IP) \leftarrow ((SP) + 1, (SP)), (SP) \leftarrow (SP) + 2$

$(CS) \leftarrow ((SP) + 1, (SP)), (SP) \leftarrow (SP) + 2$

$(FALGS) \leftarrow ((SP) + 1, (SP)) \leftarrow (IP), (SP) \leftarrow (SP) + 2$



中断向量（区）

0000:0~3FF



} 中断向量区

86系列PC:

4种中断: **NMI, INTR**, 内部中断、软中断

中断类型号 **n**: 0 ~ 255

——中断向量表 **4*n**: 0 - 3FFH

0:4*n存中断向量 → CS:IP



8086/8088 中断过程

保护断点

(F 入栈; IF=0 TF=0; CS、IP 入栈;

中断服务 (中断类型号: n)

(0: $4*n$ 取向量 \rightarrow CS:IP);

返回断点

(IRET: 依次弹出 IP、CS、F)

问题: 中断源如何识别?

多中断源如何协调?

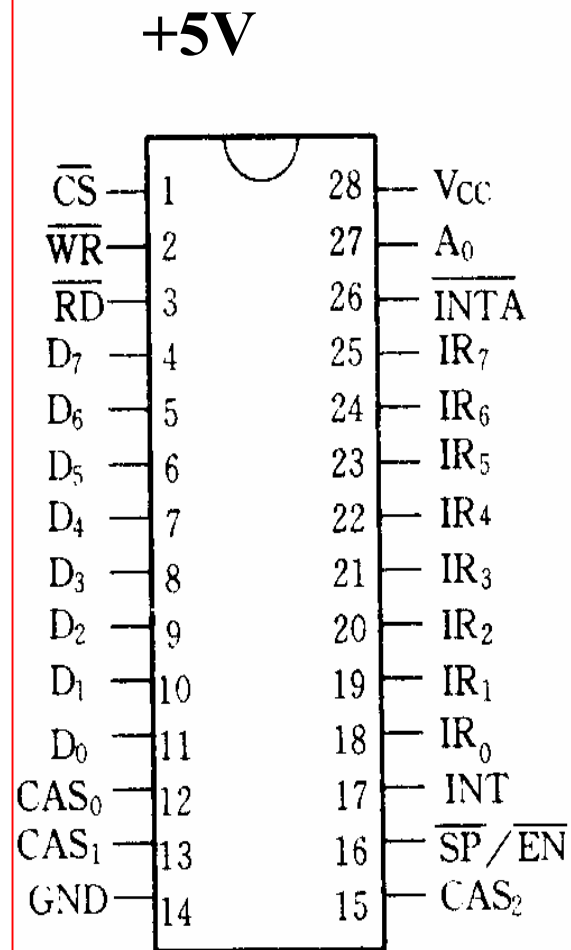
中断控制逻辑 \rightarrow 专用 IC i8259



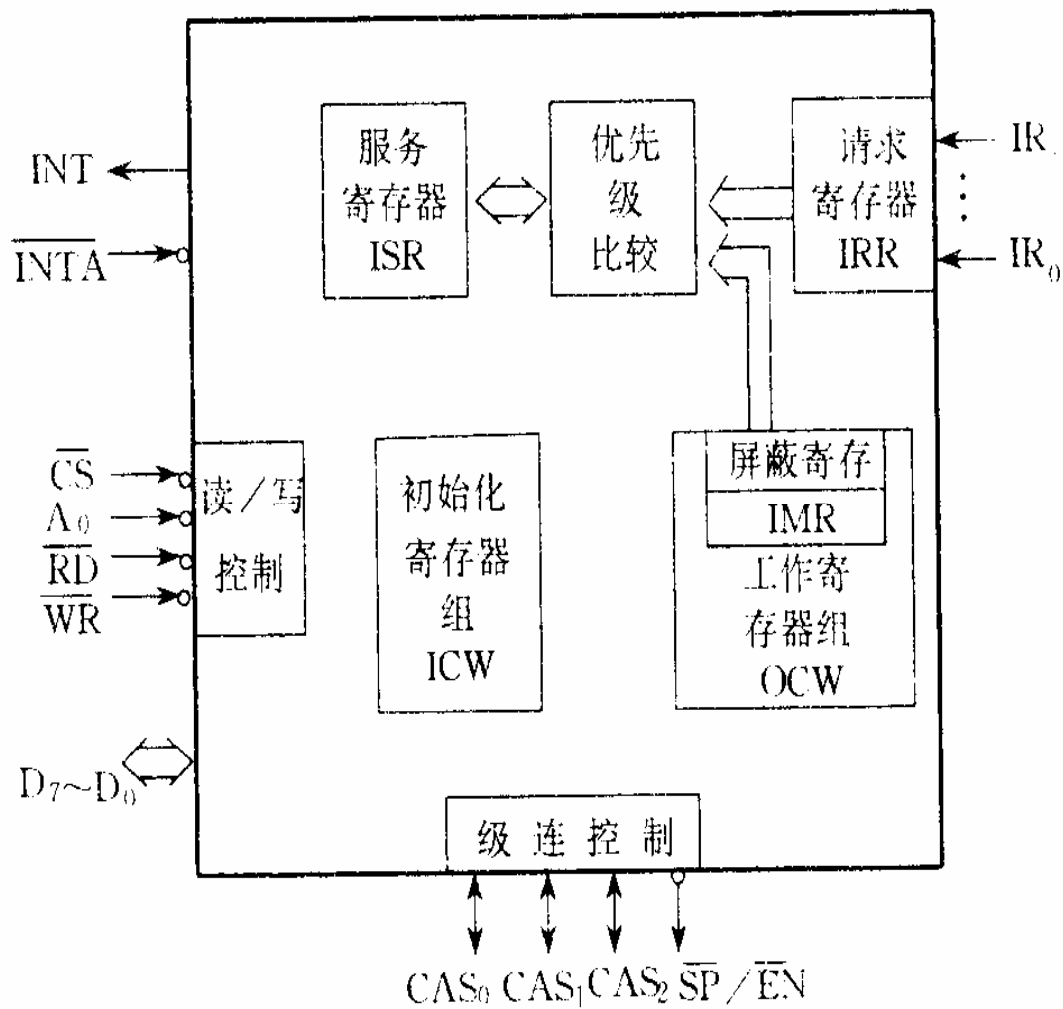
i8259A可编程中断控制器PIC

Top View

28PDIP



Function Diagram

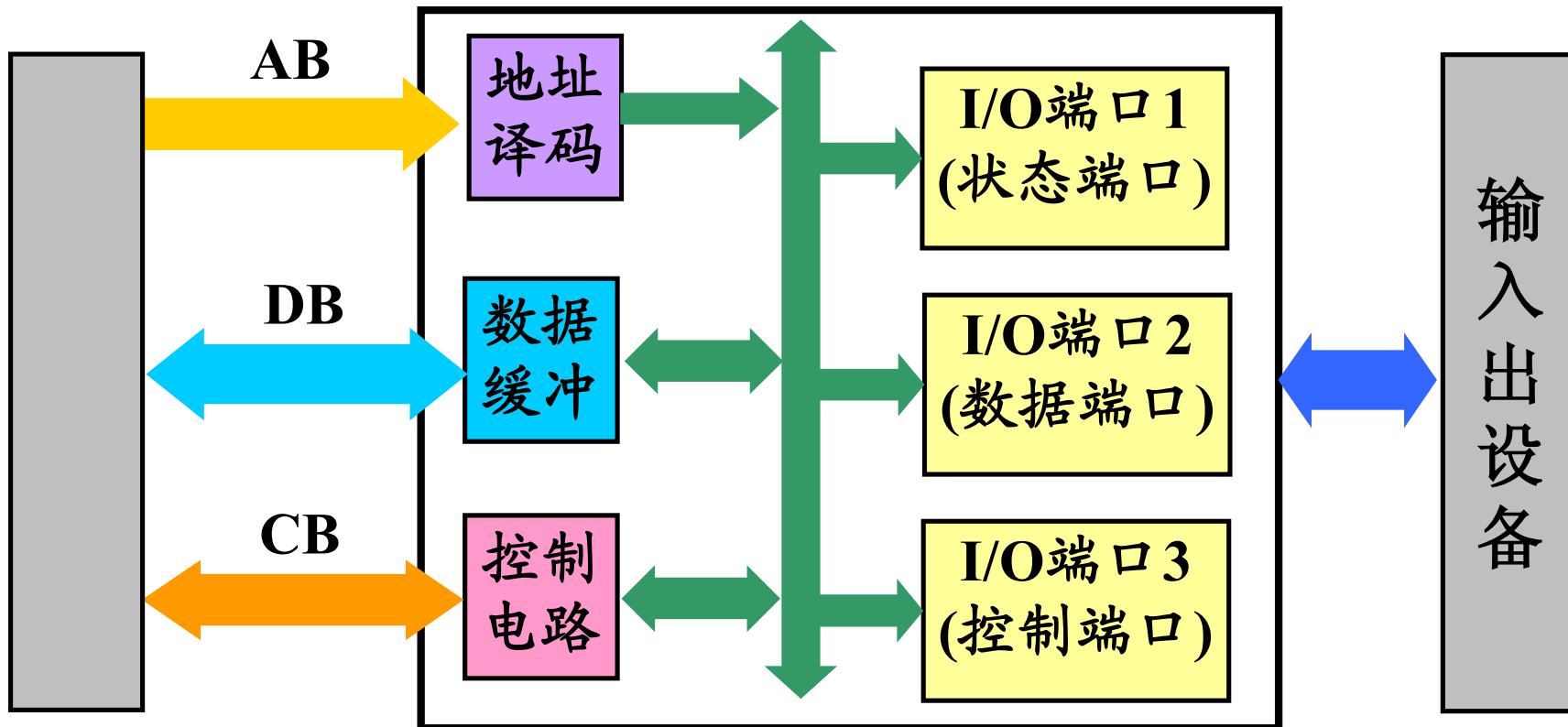




接口电路的典型结构

PC总线

外设





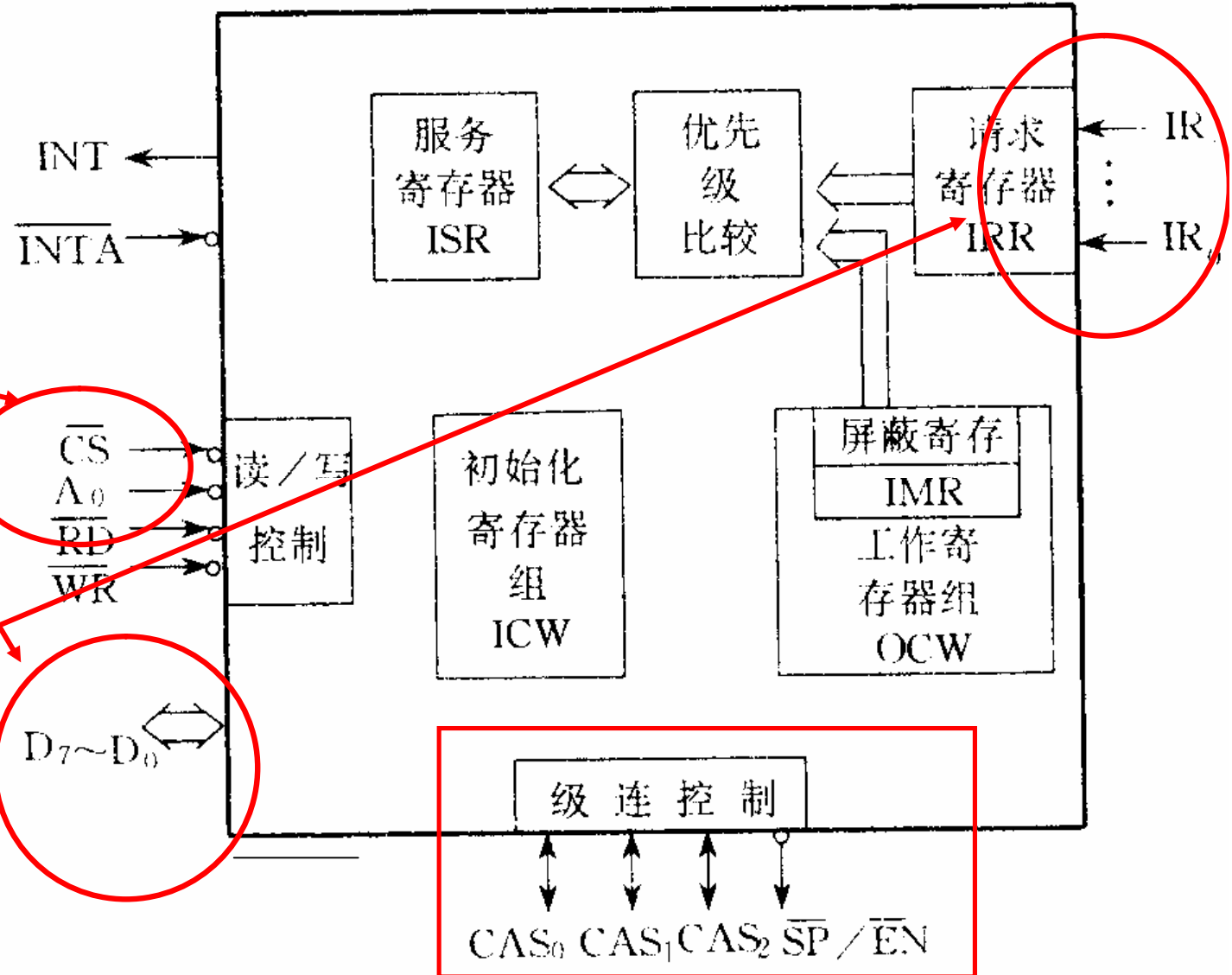
i8259A可编程中断控制器PIC

MPU接口:

D0~D7, /RD,
/WR, /CS,
A0:2个口地址, PC:20/21H

INT申请
/INTA应答
(允许)

IR0~IR7: 外部级中断源信号



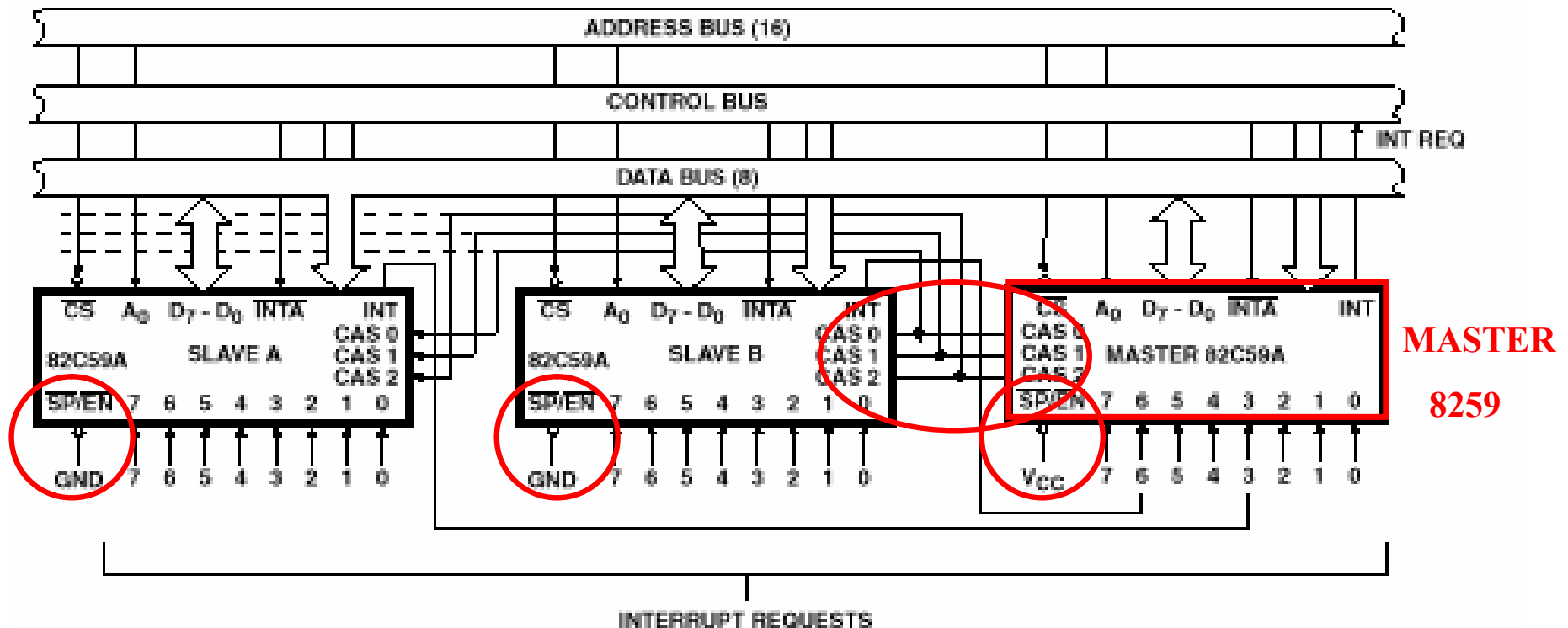


8259级联扩展

SP/EN和CAS0~CAS2 (级联)

MASTER (SP=1, IR0~7接各从片INT,
输出编码CAS0~2—8片)

Slave: SP=0, CAS入, 通常模式下确定





8259A编程

两步:

- 1、初始化编程——一次写入（编程）
初始化命令字ICW1~4， P300
- 2、工作编程——可多次写入（编程）
工作命令字OCW1~3， P303

寄存器的寻址与读写：（P300）

编程（写） ICW1~4； OCW1~3

初始化编程顺序：

ICW1 → ICW2

→ [多片时写ICW3] → [需要时写ICW4]



8259A初始化编程

ICW1: 标志 (口地址) $A0=0$

特征位 $D4=1$: 工作方式 (ICW标志 \rightarrow 初始化开始:
结果清中断屏蔽寄存器 $OCW1=0$; 优先级 $IR0$ (高) -- $IR7$
(低)

级联 (多片) 扩展位 $D1$: $D1=0$ 级联, 要送 $ICW3$
($=1$ 单片, 不送 $ICW3$);

中断控制标志位 $D0$: $D0=1$ 要送 $ICW4$ (不送时 $ICW4$
各位 $=0$);

触发电平位 $D3$: $D3$ (LTIM) $=1$ 高电平触发 $=0$ 上升
沿触发

可忽略位: $D7\sim D5$ ($A7\sim A5$), 8080/85用中断矢量;
 $D2$ ($ADI=1$ 间隔 $4/0=8$ —80/85)



8259A初始化编程(续)

ICW1→ICW2

ICW2: 标志(口地址) $A0=1$

可称为中断类型号(中断矢量)寄存器

中断类型号的高五位:

编程时写入D7~D3 (T7~T3)

中断类型号的低三位:

编程时不写D2~D0, 响应时由外设自动填入

中断源序号000B——111B

总的中断类型号: D7~D0

$\times \times \times \times \times 000B$ —— $\times \times \times \times \times 111B$

8个中断矢量的地址是连续的!



8259A初始化编程(续)

ICW1 → ICW2

→ ICW3 (此时: ICW1的D1=0级联, 要送ICW3)

ICW3: 级联命令字, 有主从片之分:

对MASTER主片:

(S7~S0) D7~D0对应8根中断请求线IR7~IR0

某根中断请求线IR接SLAVE从片, 则=1;

对SLAVE从片:

从片标识码 (ID2~ID0) D2~D0——对应8片从片

(中断响应时与CAS码比较, 相同为当前中断源)

(注: D7~D3=0)

注意: 每片(主从片)均应顺序写入ICW1 → ICW2 → ICW3

。



8259A初始化编程(续)

ICW1 → ICW2

→ [ICW3] → ICW4 (此时: ICW1的D0=1, 要送ICW4)

ICW4: 中断嵌套时的优先级控制

(D7~D5=0)

D4 (SFMN) =0 一般嵌套 (已服务SLAVE片屏蔽、高优先不能嵌套) D4=1特殊嵌套 (中断源屏蔽、可中断嵌套)。

D3 (BUF) 数据缓冲选择 (=0 无BUFFER; =1有三态BUFF, 单片8259A的DB→系统总线间加BUFFER由/SP/EN控制, 发送时/EN=0; D2=0SLAVE/=1MASTER;

D1 (AEOI) =1 自动EOI

0正常EOI---中断结束信号。

D0 (μPM) =0 -8080/85系列 =1 8086/88系列

例: PC—00001001B 88/86,非自动EOI



8259A工作编程——可多次编程

OCW1: 标志 (口地址) $A0=1$, 中断屏蔽字
每位对应1根中断请求线 (=1屏蔽, =0允许)

OCW2: 标志 (口地址) $A0=0$, 特征位 $D3=D4=0$
EOI (20H) 命定/中断优先权的旋转 (P304)

OCW3: 标志(口地址) $A0=0$, 特征位 $D3=1, D4=0$

辅助操作 (不常用)

特殊屏蔽、 $D2=1$ 查询方式 (读状态字) $\neq 0$ 非; $D1D0 = 10$ 随后读IRR ($D7 = 1$ 有效请求; $D2 \sim D0 = I R 0 \sim I R 7$ 之一编码), $D1D0=11$ 随后读ISR (中断服务R服务时相应位 = 1)。



PC/XT 中的8259A应用

对PC/XT系统总线（ P305 ）：

- 写： 20H: ICW1、 OCW2、 OCW3;
21H: ICW2、 ICW3, ICW4、 OCW1
- 读： 20H: 中断请求寄存器IRR、
中断服务寄存器ISR、
中断级编码（查询方式时）
注： 由写OCW3决定读何寄存器；
D1D0=10随后读IRR；
D1D0=11随后读ISR。
21H: 中断屏蔽寄存器IMR



PC/XT 中的8259A应用

对PC/XT系统总线（ P305 ）：

8259中断请求输入连接（ P460 ）

引脚	IRQ0	IRQ1	IRQ2	IRQ3	IRQ4	IRQ5	IRQ6	IRQ7
用途	定时器	键盘	保留	通信网	保留	硬盘	软盘	打印机
中断类型	8	9	0AH	0BH	0CH	0DH	0EH	0FH



PC/XT 中的8259A应用

(BIOS) 初始化编程 (P306)

```
MOV AL, 13H;  
OUT 20H, AL; (ICW1)  
MOV AL, 8;  
OUT 21, AL; (ICW2)  
MOV AL, 9;  
OUT 21H, AL; (ICW4)  
MOV AL, 0FFH;  
OUT 21H, AL; (OCW1)
```




可屏蔽中断应用

主程序要点:

为中断做准备:

保存系统中断向量、设置用户中断向量

设置设备中断屏蔽位

[屏蔽R地址 21H(在8259中) $D_i=1$ 屏蔽;
 $D_i=0$ 允许]

设置CPU中断允许位 **STI**

恢复系统中断向量设置



保护中断向量

保护原1CH矢量(P308)

Tvector_CS DW 0 ; 准备保护原矢量段地址

Tvector_IP DW 0 ; 准备保护原矢量偏移地址

intini PROC ;interrupt initializing

CLI ; 以下保护矢量

MOV AH,35H;

MOV AL,1CH; 原矢量中断类型码

INT 21H; 取原矢量

MOV Tvector_IP,BX; 保护原矢量偏移地址

MOV AX,ES;

MOV Tvector_CS,AX; 保护原矢量段地址



修改中断向量

——指向中断服务程序

修改1CH矢量——指向中断服务程序(P307)

PUSH DS ; 以下修改矢量

MOV DX,OFFSET INTR; 中断服务程序偏移地址

MOV AX,SEG INTR; 中断服务程序段地址

MOV DS,AX

MOV AH,25H;

MOV AL,1CH; 中断类型码

INT 21H; 修改矢量

POP DS



再次修改中断向量

——原中断向量恢复

再次修改1CH矢量——恢复原中断向量 (P308)

```
PUSH DS ; 以下修改矢量
MOV DX, Tvector_IP; 原偏移地址
MOV AX, Tvector_CS; 原段地址
MOV DS, AX
MOV AH, 25H;
MOV AL, 1CH; 中断类型码
INT 21H; 恢复原矢量
POP DS
```



可屏蔽中断应用（续）

中断服务子程序要点：

保存现场，R内容

如允许中断嵌套，开中断（STI）

处理中断

关中断（如允许中断嵌套）

送**中断结束命令EOI**给8259中断命令R

```
MOV AL, 20H
```

```
OUT 20H, AL
```

恢复现场，R内容

中断结束返回IRET



中断服务程序

关键:保护现场

MYINT PROC FAR

PUSH AX;

PUSH BX ;其他寄存器

.....

.....

POP BX

MOV AL, 20H; EOI

OUT 20H, AL;

POP AX;

IRET (注意结尾)

MYINT ENDP



中断 (Interrupt) 小结

(1) 中断概念: CPU与外设交换信息的一种方式

==》 硬件手段, 改变CPU执行程序的顺序 (程序流)

(2) 中断控制:

中断源---中断请求---中断响应---中断服务---中断恢复
(中断向量---优先级---中断屏蔽---中断嵌套)

(3) 中断控制器: i8259A, 每片直接管理8级外部中断,
级联扩展: 64级可编程

(4) 软件初始化: a. 芯片级 b. 系统级