第20讲 最后一课

关于考试

- 填空
  - 20空，20分
- 简答
  - 6题，18分
- 选择
  - 6题，12分
- 综合
  - 4题，50分

计算机基础

什么是计算机

- 计算机
  - Computer VS Calculator
  - 模拟人脑
  - 电脑：
    - 是一种利用电子学原理，根据一系列指令来对数据进行处理的机器。
    - 处理信息的工具。根据图灵机理论，一部具有最基本功能的计算机，应当能够完成任何其它计算机能做的事情。
    - 只要不考虑时间和存储因素，从个人数码助理（PDA）到超级计算机都应该可以完成同样的作业。

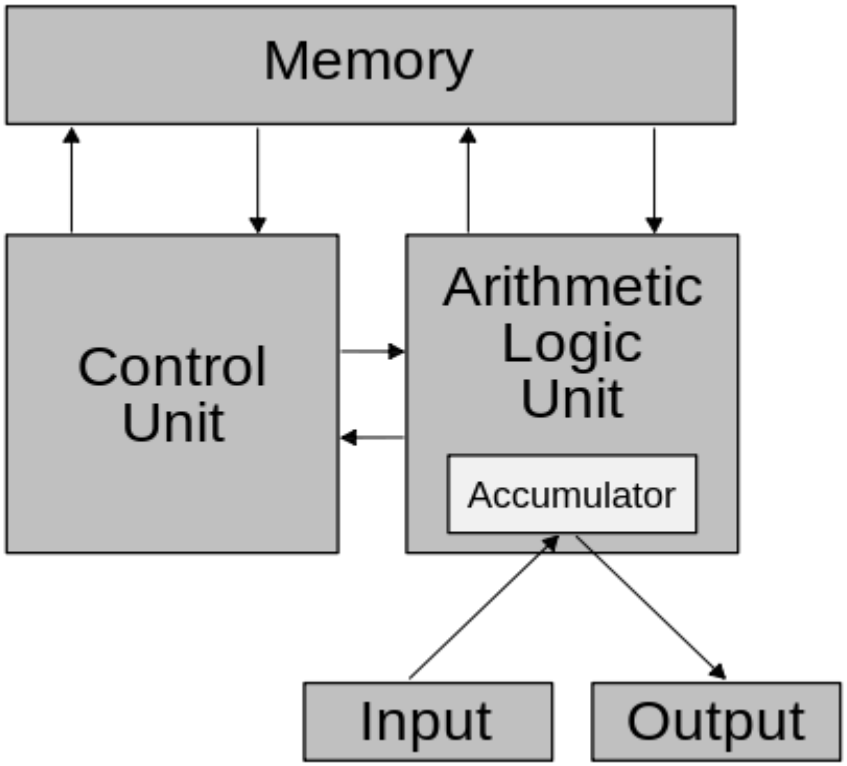
计算机的分类

- 个人计算机：台式计算机、便携计算机
  - 性能价格比
- 服务器（商用）
  - 高可靠性、吞吐率
  - 低时延
  - 可扩展
- 嵌入式计算机系统（专用）
  - 低成本，低功耗，小体积

冯·诺依曼体系结构

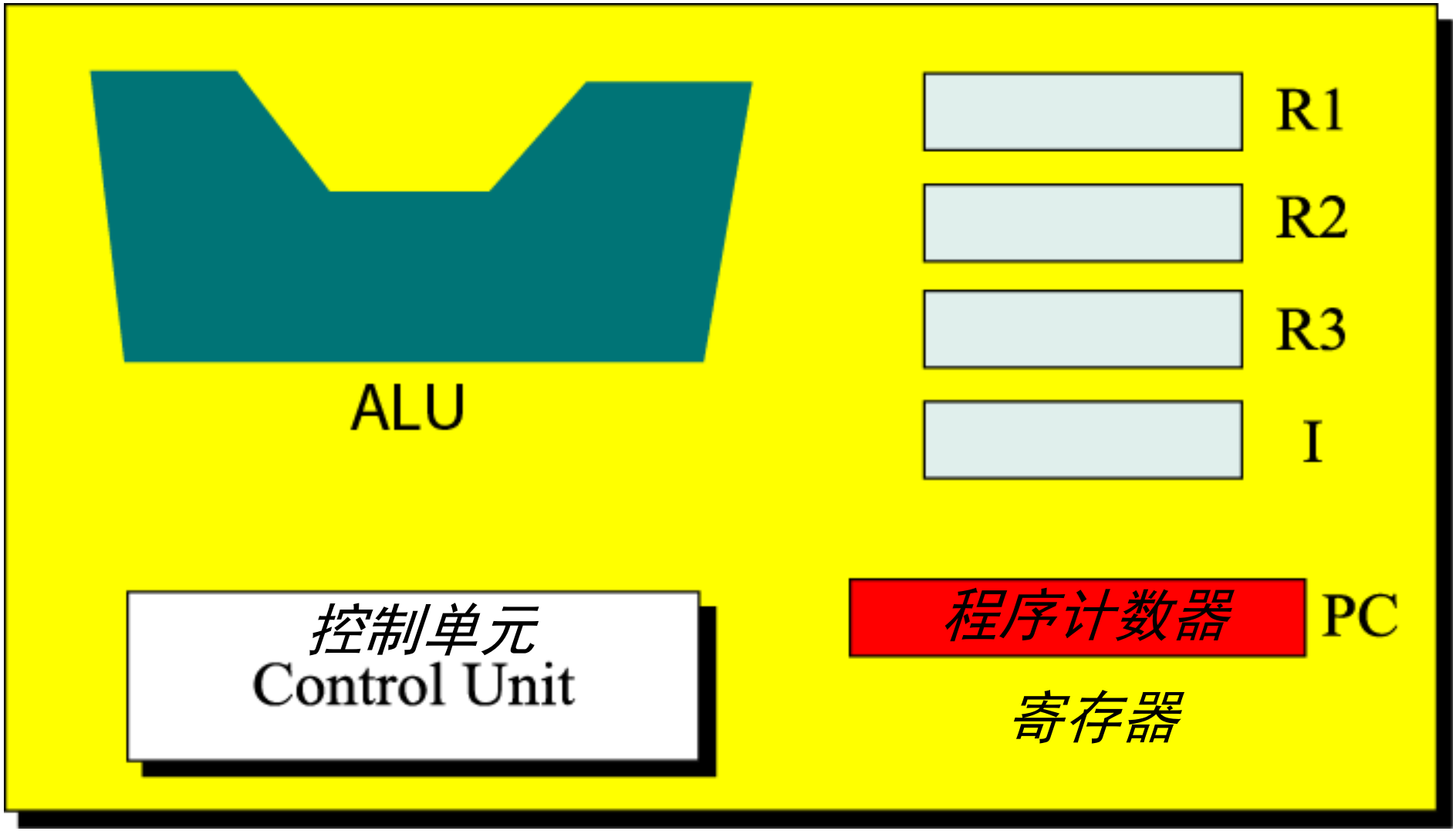
冯 诺依曼的三个重要设计思想

- 采用二进制数表示指令和数据;
- 将程序（由一系列指令组成）和数据存放在计算机的内存中，并让计算机自动执行
- 五大基本部件;



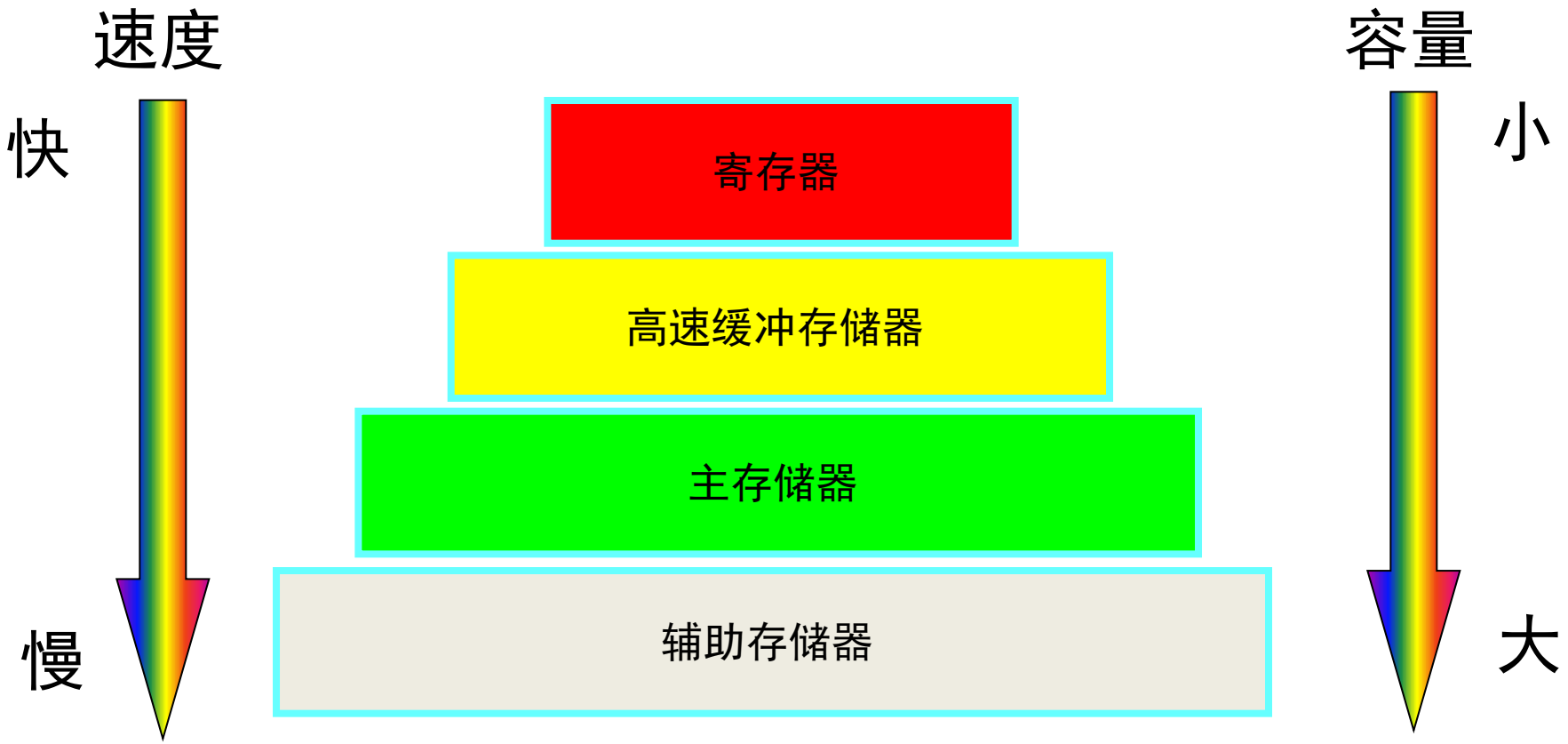
中央处理单元（CPU）

- CPU 包括三部分：算术逻辑单元 (ALU)、控制单元和寄存器。



3.存储器的层次结构

不同的存储器具有不同的存储容量、不同存取速度。



CPU

微处理器

- 处理器
  - 算术逻辑单元和控制逻辑单元
  - 可编程
  - 集成在一块或者数块集成电路之内
  - 物理上：一块芯片
- “微”
  - 当初各大芯片厂之工艺采用 1 微米的阶段，厂商在产品名称上用“微”字，强调他们很高科技。就如同现在的许多商业广告一样，很喜欢用“纳米”字眼。

CISC VS RISC

- CISC
  - 复杂指令集
    - 可以直接对应高级编程语言高级功能的复杂指令
  - 原因
    - 为了便于编写程序
    - 缺乏大容量的内存
    - 存储器访问速度慢
    - 只含有少量寄存器
  - “正交性”，为每个指令都提供所有的寻址模式，这给微处理器增加了一些复杂性
  - X86

- RISC
  - 精简指令集
  - 原因
    - 编译器的使用逐渐增多而汇编语言的使用相对减少，使得大多数正交寻址模式基本上已被程序员所忽略
    - 相比用更精简的一系列指令来完成同一个任务，用单一复杂指令甚至会更慢
    - 微处理器开始比内存运行得更快，需要有更多寄存器（以及后来的缓存）来支持更高频率的操作
  - ARM

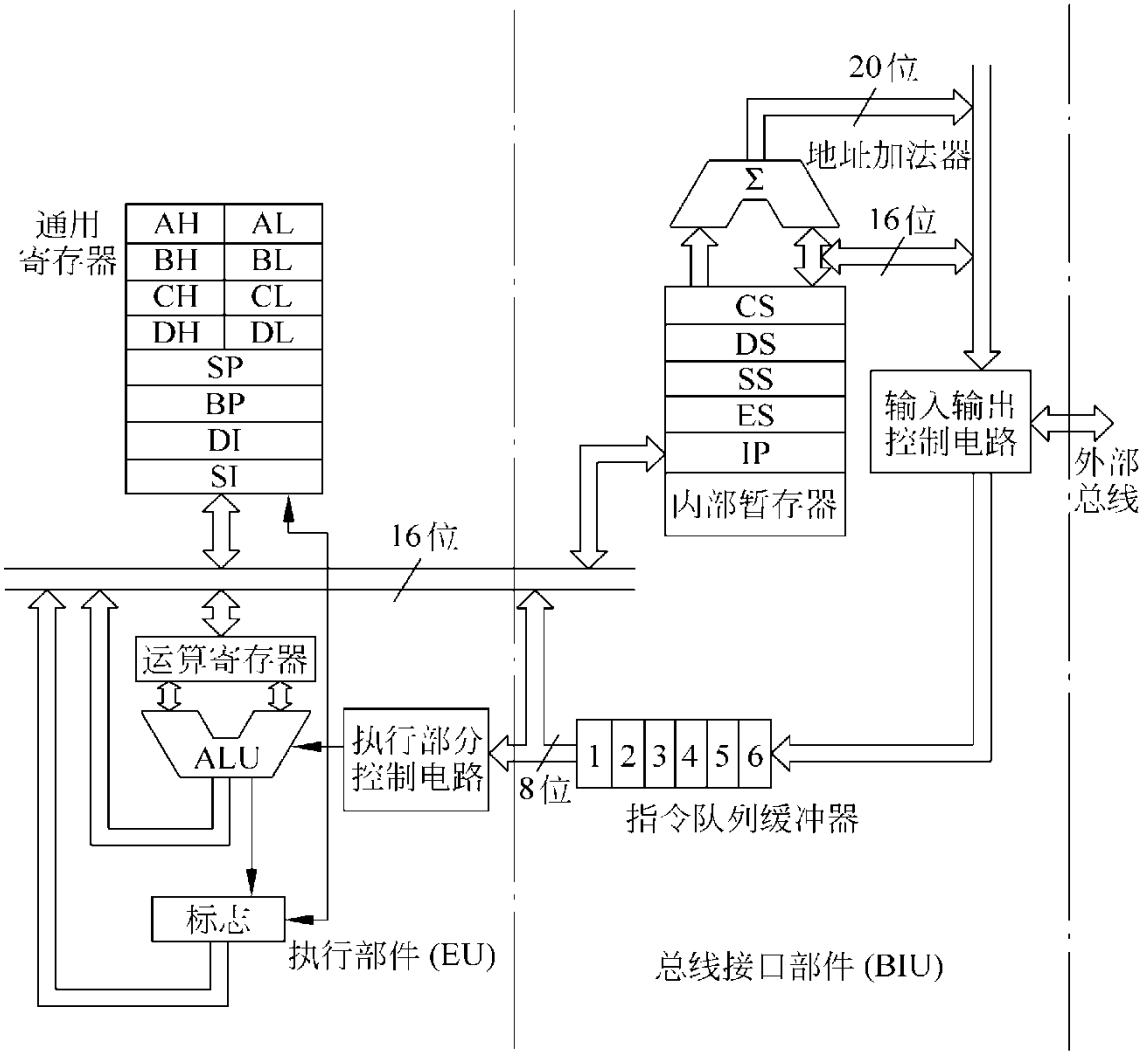
- 8086是16位处理器
- 80386→32位（1985）
- 2003 AMD → AMD64，Intel兼容的Intel 64
- 两者被称为x86-64或x64
- 不同于IA-64

- X86架构是可变指令长度的CISC
- 向前兼容
- 较新的架构改进
  - 将x86指令转化为更像RISC的微指令再予以执行
- 四种执行模式
  - 真实模式
  - 保护模式
  - 系统管理模式（如果说三种时，不答这个）
  - 虚拟86模式

8086编程结构

- **编程结构：**从程序员和使用者的角度看到的结构，与CPU内部的物理结构和实际布局是有区别的。
- **总线接口部件**（bus interface unit, BIU）
  - 负责与存储器、I/O端口传送数据。
- **执行部件**（execution unit, EU）
  - 负责指令的执行。

8086的编程结构


20位
地址加法器
Σ
16位
通用
寄存器
AH
AL
BH
BL
CH
CL
DH
DL
SP
BP
DI
SI
CS
DS
SS
ES
IP
内部暂存器
输入输出
控制电路
外部
总线
16位
运算寄存器
ALU
执行部分
控制电路
1
2
3
4
5
6
8位
指令队列缓冲器
标志
执行部件 (EU)
总线接口部件 (BIU)

8086工作模式

最小模式： 系统中只有一个8086微处理器，所有总线控制信号都由8086直接产生，系统中控制电路可以减小到最小；

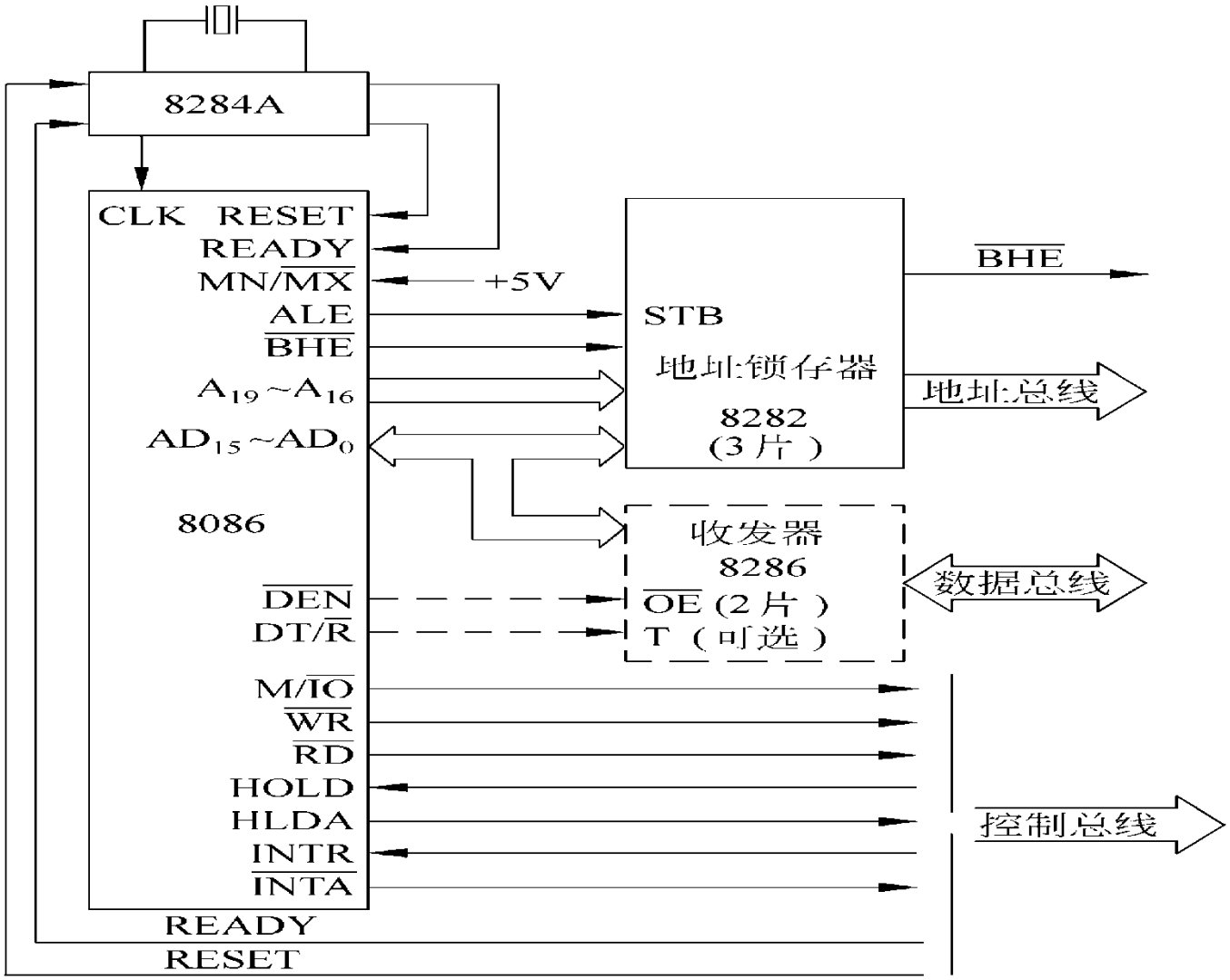
最大模式： 系统中包含两个或多个微处理器，其中一个主处理器是8086，其他的处理器称为协处理器，协助主处理器工作；

协处理器：

- 8087：专用于数值运算的处理器，用硬件方法完成整数、浮点数或超越函数的计算，提高系统的数值运算速度；
- 8089：专用于输入输出控制的处理器，直接为输入输出设备服务，使8086不再承担这类工作，提高主处理器效率。

✓8086在不同模式下，部分引脚功能定义不同；

8086最小模式信号连接



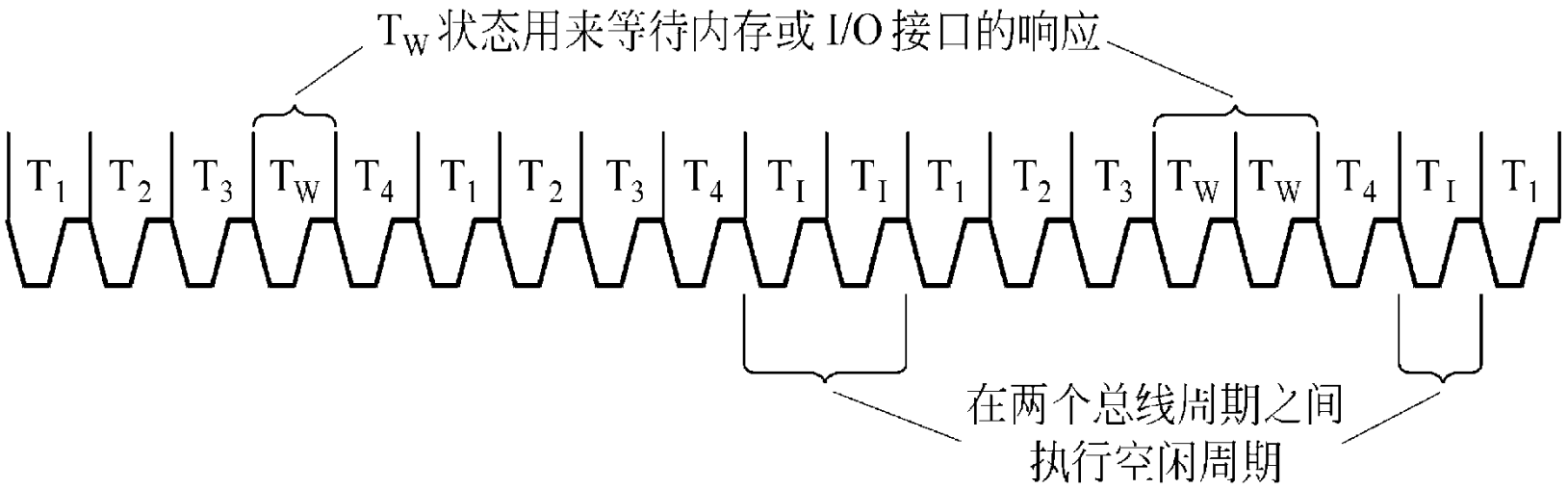
8086总线周期

时钟周期： 计算机主频决定的基本时间计量单位；

指令周期： 从取指令到执行指令完毕所需的时间；

总线周期： 从外部存储器或端口存取一次数据所需的时间；

8086总线周期由4个时钟周期组成，对应总线的4个状态分别为T1、T2、T3、T4；有时会插入等待状态Tw和空闲状态Ti；



8086总线周期

T1状态：CPU往地址/数据复用总线（AD）上发送地址信息，指出要寻址的存储单元或端口地址；

T2状态：CPU从总线上撤销地址，而使总线的低16位成高阻状态，为传输数据做准备。总线的高4位用来输出本总线周期状态信息；

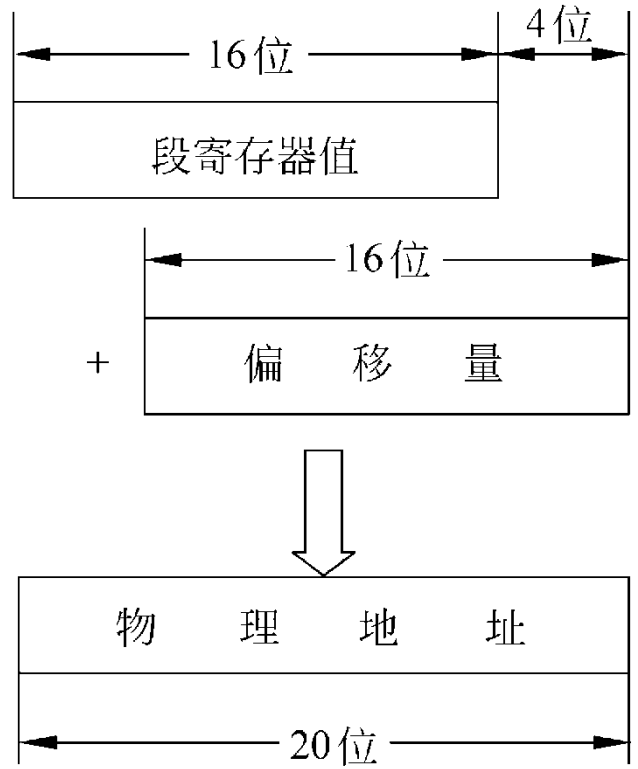
T3状态：地址/数据复用总线上出现CPU写出或从存储器、端口读入的数据；

T4状态：总线周期结束；

Tw状态：对于速度较慢的外设或存储器，不能及时配合CPU传送数据，这是外设可通过“READY”信号线在T3状态启动之前向CPU发一个“数据未准备好”信号，CPU会在T3之后插入一个或多个附加的时钟周期Tw，直到外设或存储器完成数据传输时，在”READY”上发出”准备好”信号。CPU据此信号自动脱离Tw状态，进入T4状态；

Ti状态：CPU不执行总线周期时，总线处于空闲状态；

8086的存储器编址



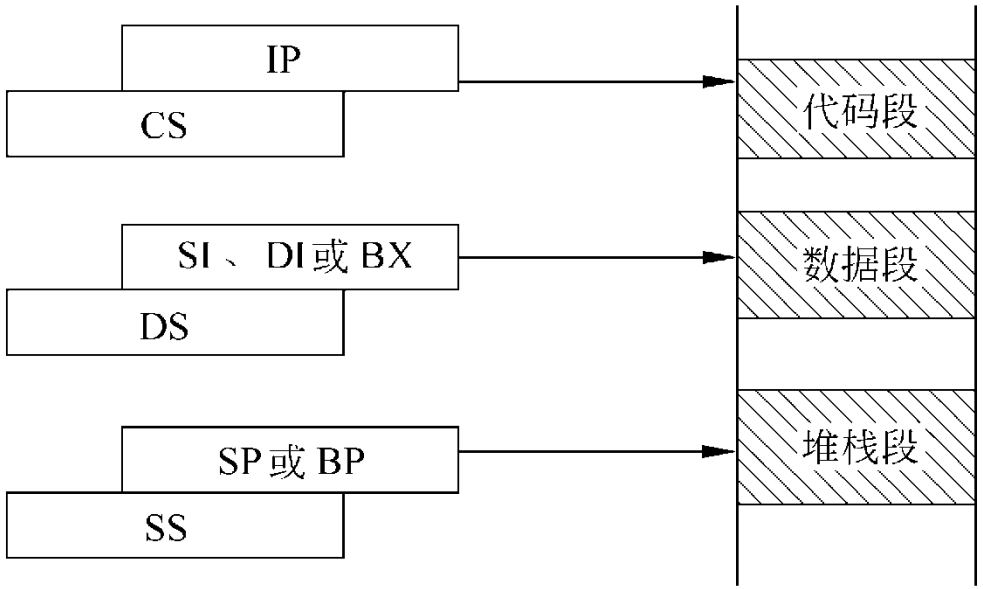
8086的存储器编址

物理地址的形成：

取指令：CS：IP

堆栈：SS：SP /BP

读写数据：DS：SI/DI/BX



8086的固定用途存储区

固定用途存储区

00000H－003FFH，1KB，中断向量表；

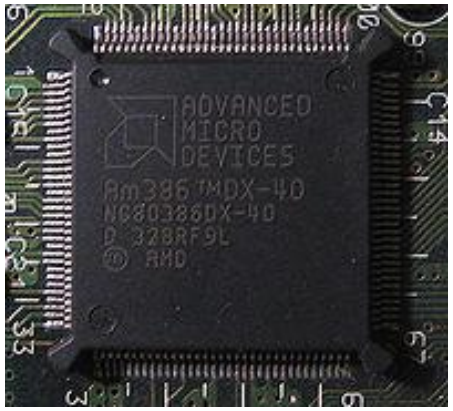
B0000H－B0F9FH， 4KB， 单色显示器显示缓冲区；

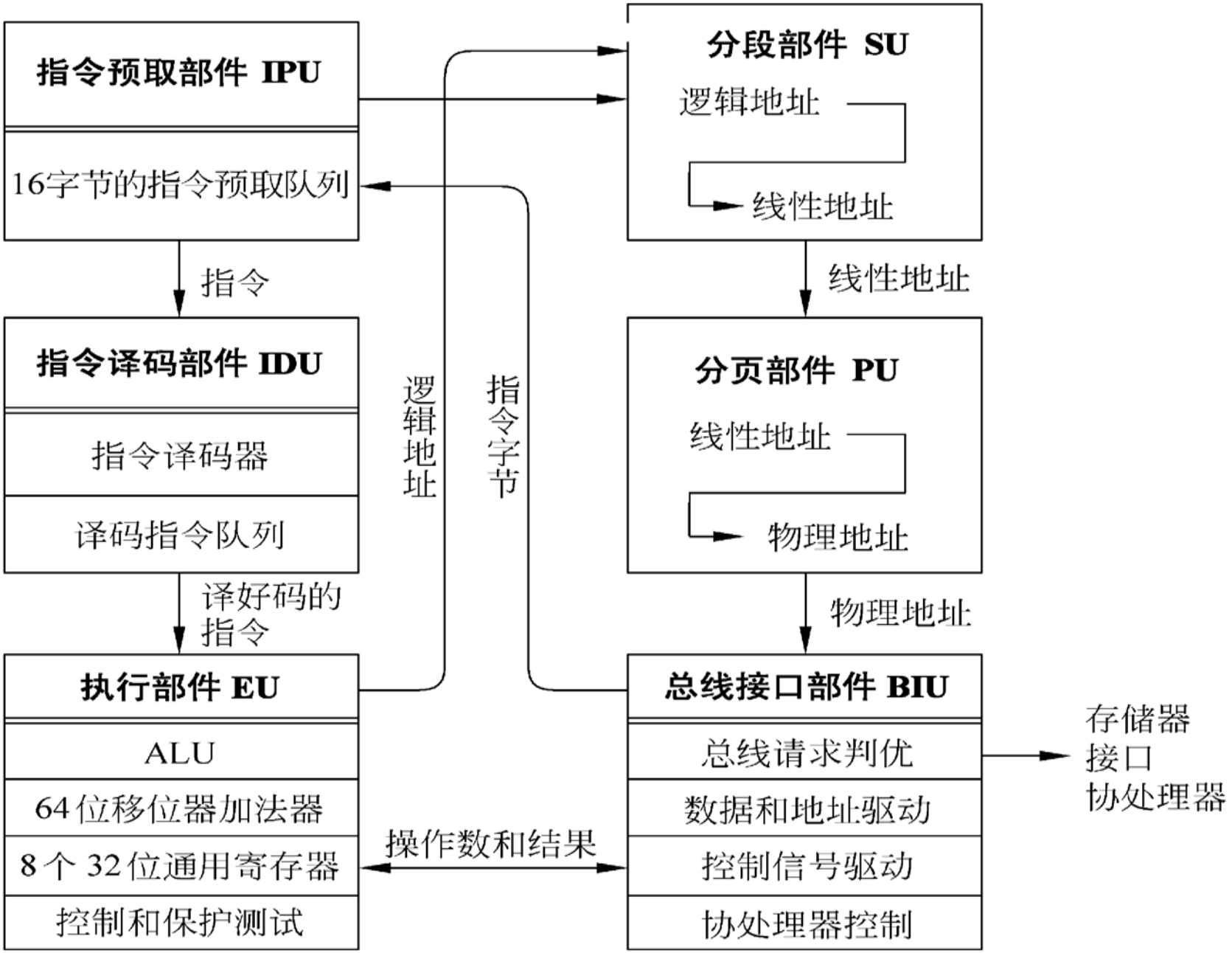
B8000H－BBF3FH，16KB，彩色显示器显示缓冲区；

FFFF0H－FFFFFH，16B，复位程序入口；

80386

- 革命性的一代
- 1985年推出的32位CPU；
- 兼容8086、80286CPU功能；
- 有32条地址线、32条数据线；
- 三种工作方式：实地址方式、保护方式和虚拟8086方式；
- 保护方式下，直接寻址4GB物理地址空间，虚拟存储空间为64TB；
- 采用分段部件、分页部件支持虚拟存储；



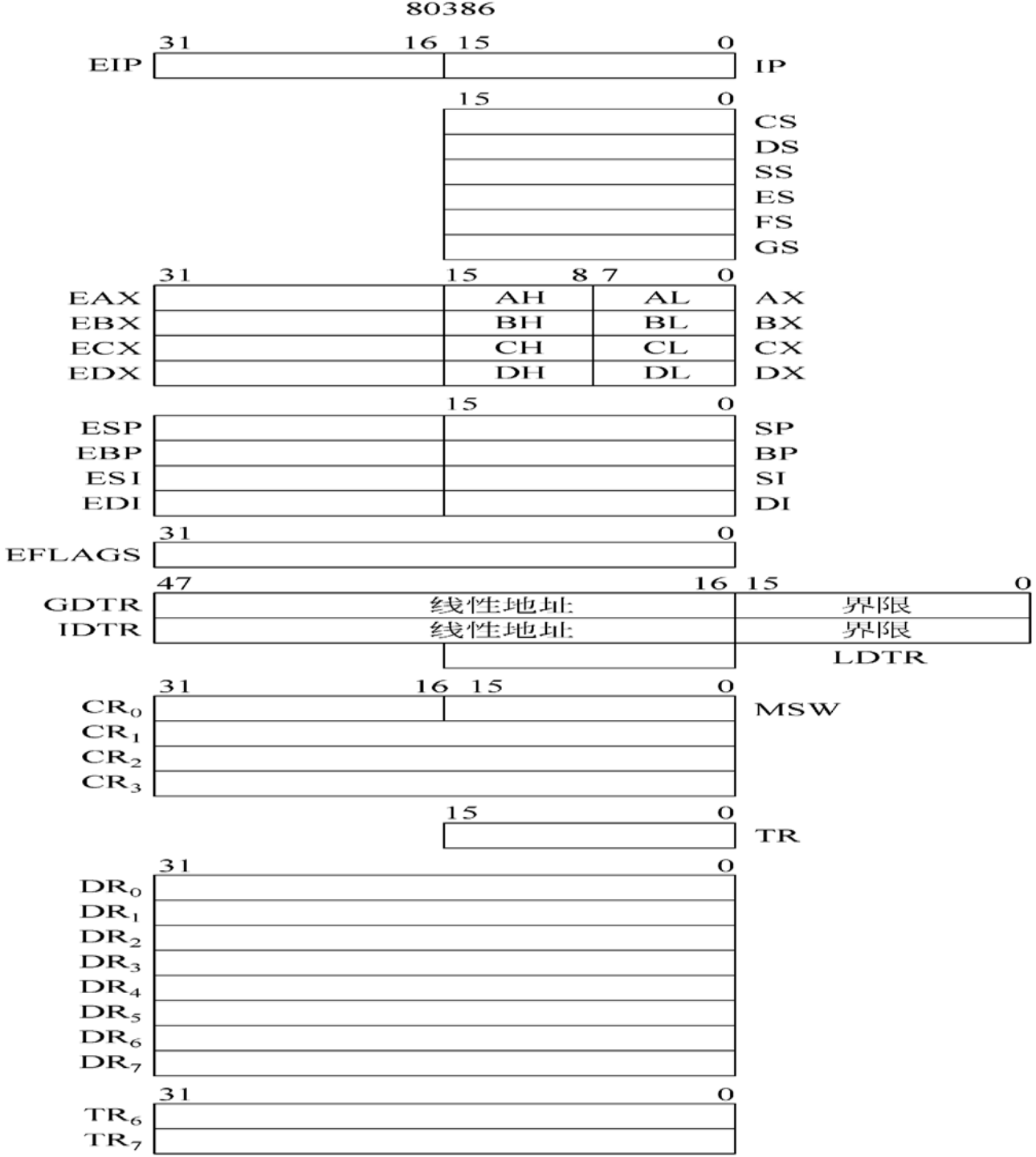

指令预取部件 IPU
16字节的指令预取队列
指令
指令译码部件 IDU
指令译码器
译码指令队列
译好码的指令
执行部件 EU
ALU
64位移位器加法器
8个32位通用寄存器
控制和保护测试
逻辑地址
指令字节
分段部件 SU
逻辑地址
线性地址
线性地址
分页部件 PU
线性地址
物理地址
物理地址
总线接口部件 BIU
总线请求判优
数据和地址驱动
控制信号驱动
协处理器控制
操作数和结果
存储器
接口
协处理器

- **6个组成部分**
  - ➢**CPU**
    - —指令预取部件(IPU)
    - —指令译码部件(IDU)
    - —执行部件(EU)
  - ➢存储器管理部件(MMU)
    - —分段部件(SU)
    - —分页部件(PU)
  - ➢总线接口部件(BIU)

80386CPU寄存器结构

80386CPU共有8类32个寄存器：

- 通用寄存器、
- 段寄存器、
- 指令指针
- 标志寄存器、
- 控制寄存器、
- 系统描述符表寄存器、
- 调试寄存器
- 测试寄存器。


80386
31
16
15
0
EIP
IP
CS
DS
SS
ES
FS
GS
EAX
EBX
ECX
EDX
AH
BH
CH
DH
AL
BL
CL
DL
AX
BX
CX
DX
ESP
EBP
ESI
EDI
SP
BP
SI
DI
EFLAGS
47
GDTR
IDTR
线性地址
界限
LDTR
MSW
TR
CR_0
CR_1
CR_2
CR_3
DR_0
DR_1
DR_2
DR_3
DR_4
DR_5
DR_6
DR_7
TR_6
TR_7

存储器

存储器主要性能指标

- 存储容量
  - ✓ 是指存储器芯片中所包含的存储单元（Memory cell）数。半导体存储单元通常以字节为单位，人们通常说的存储单元都是指的字节单元。
- 存取时间
  - ✓ 存储器的最重要的性能指标，是读写存储器中某一存储单元所需时间，一般指存储器接收到稳定地址信号到完成操作的时间。
- 功耗
- 可靠性
- 性价比

半导体存储器分类

- 按在系统中位置：内部存储器、外部存储器；
- 按制造工艺：双极型、MOS；
- 易失性：非易失性、易失性；
- 可读写性：只读存储器(ROM)、可读写存储器；
- 读写顺序：顺序读写存储器、随机存储器(RAM)；
- 动态/静态，异步/同步，串行/并行

只读存储器

- ✓**掩膜ROM:** mask programmed ROM;
- ✓**可编程ROM:** Programmable ROM, PROM;
- ✓**可擦除的PROM:** Erasable PROM, EPROM;
- ✓**电擦除的PROM:** Electrically Erasable PROM, E^2PROM/EEPROM;
- ✓**闪存FLASH**, NOR flash/NAND flash;

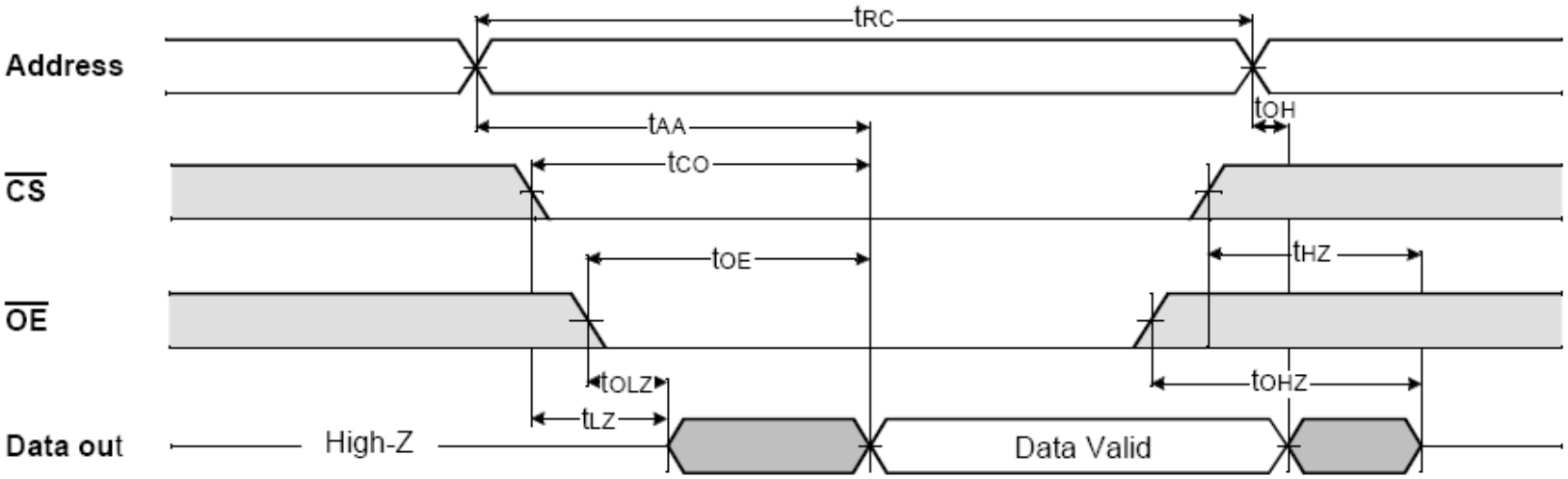
随机存取存储器RAM

- ✓**静态RAM：** Static RAM, SRAM；

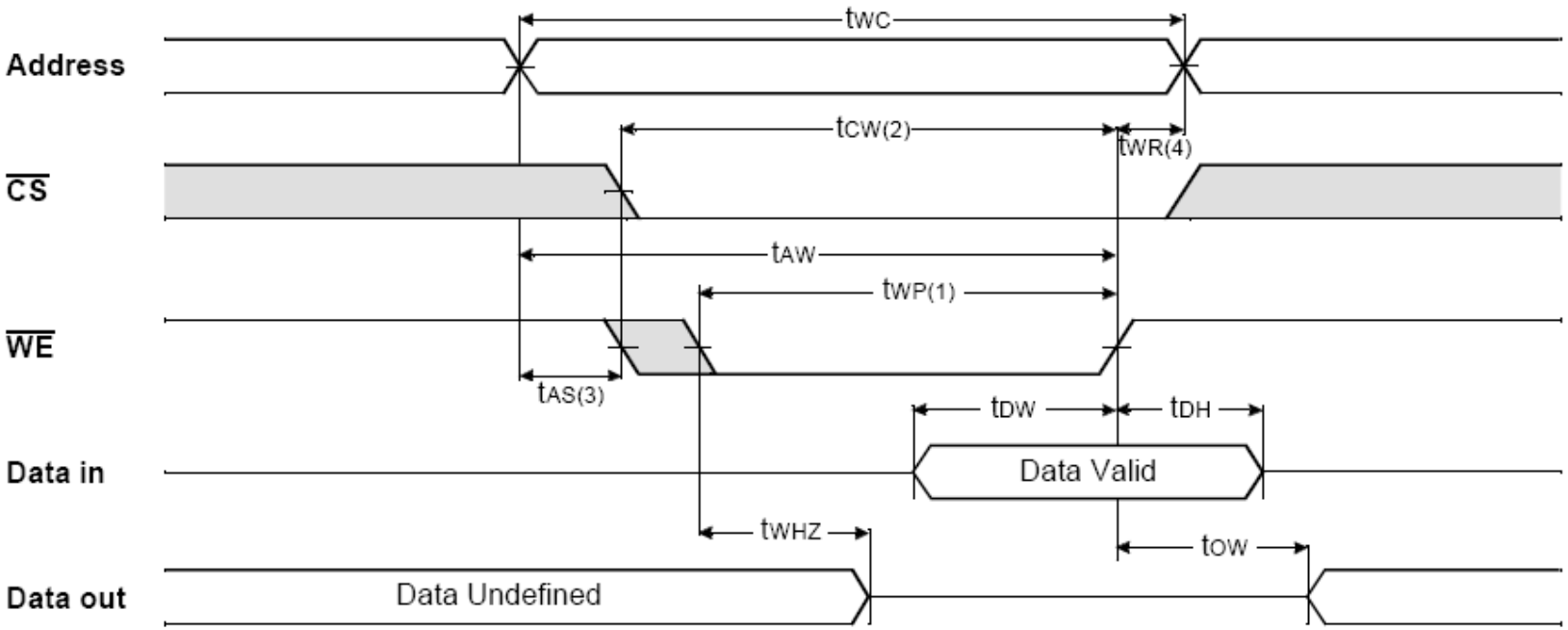
  异步静态RAM: asynchronous SRAM;
  同步静态RAM: synchronous SRAM;

- ✓**动态RAM：** Dynamic RAM, DRAM

SRAM读周期时序图


Address
CS
OE
Data out
tRC
tAA
tOH
tCO
tOE
tHZ
tOLZ
tLZ
tOHZ
High-Z
Data Valid

SRAM写周期时序图


Address
CS
WE
Data in
Data out
tWC
tCW(2)
tWR(4)
tAW
tWP(1)
tAS(3)
tDW
tDH
Data Valid
tWHZ
tOW
Data Undefined

存储器的组合与扩充

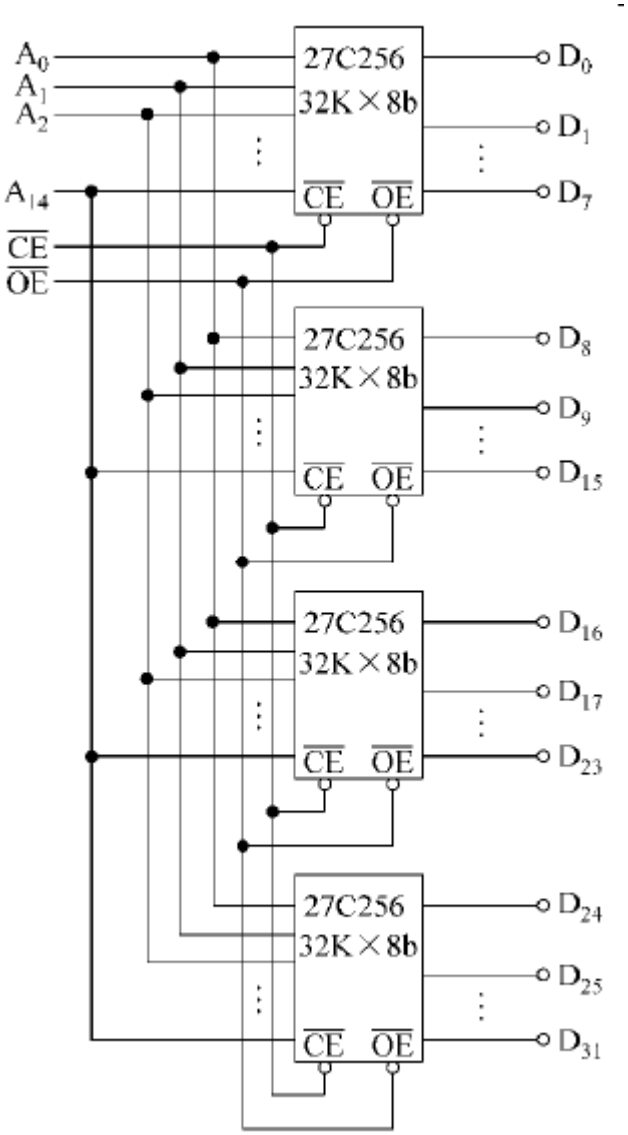
（1）存储宽度扩展；

（2）存储深度扩充；

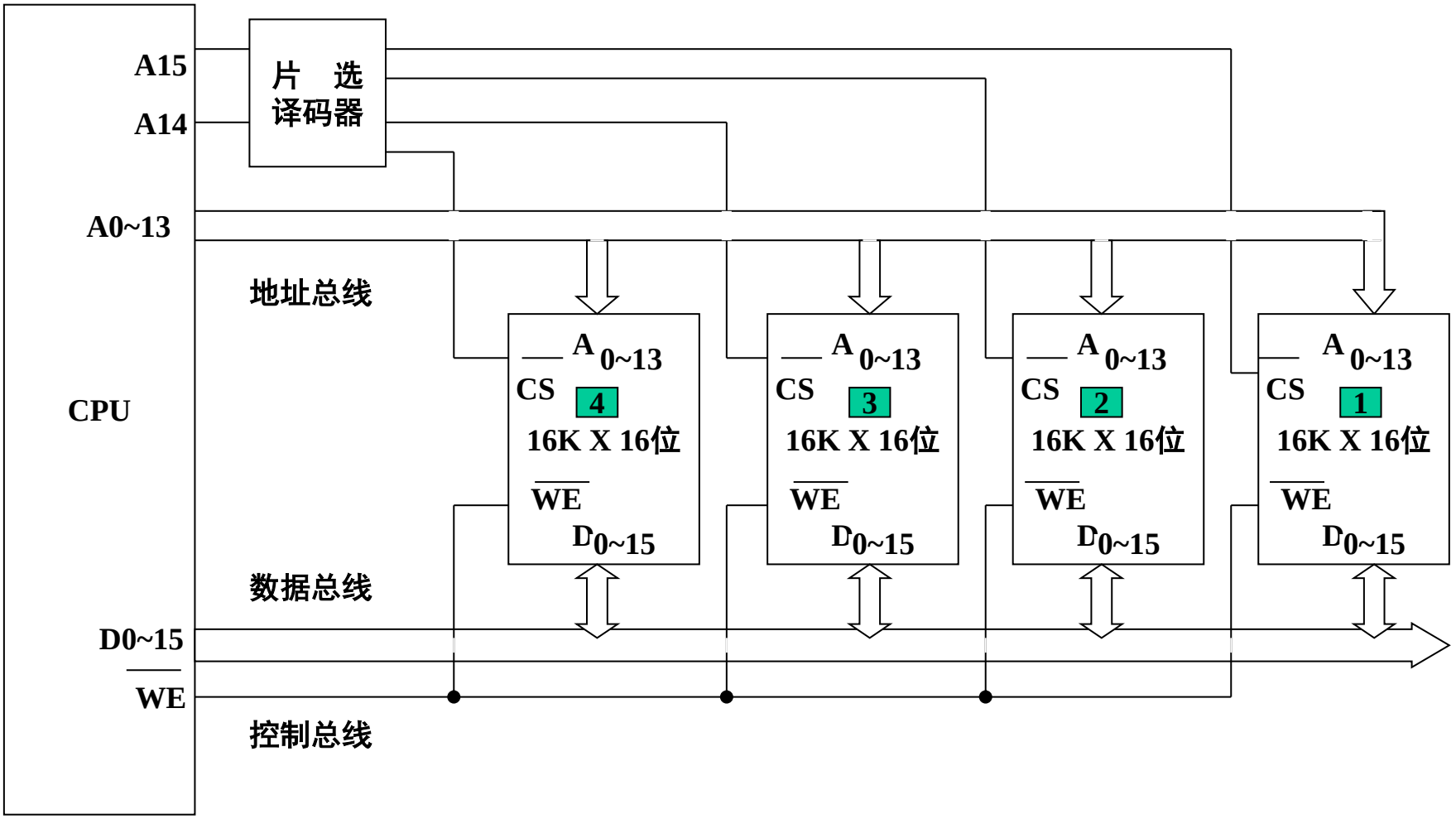
（3）16位和32位微机系统的内存组织；

涉及地址线、数据线和控制线的连接。

存储宽度扩展:位扩展


A0
A1
A2
A14
CE
OE
27C256
32K×8b
CE OE
D0
D1
D7
D8
D9
D15
D16
D17
D23
D24
D25
D31

存储深度扩充: 字扩展


A15
A14
片 选
译码器
A0~13
地址总线
CPU
CS
4
A 0~13
16K X 16位
WE
D0~15
CS
3
A 0~13
16K X 16位
WE
D0~15
CS
2
A 0~13
16K X 16位
WE
D0~15
CS
1
A 0~13
16K X 16位
WE
D0~15
数据总线
D0~15
WE
控制总线

地址译码

- 在微处理系统，存储器常常由多片组成，为了访问其中一个存储器，需要对系统中的高位地址进行译码产生片选信号，使选中的存储器可输出信号。

- 当某个存储器芯片的片选为无效电平时，它内部数据总线驱动器被关断，不会向数据总线输出数据。

- 也适用于IO端口的片选。
  - ✓线选法
  - ✓全译码法
  - ✓部分译码法
  - ✓混合译码法

地址译码

（1）线选法

- 直接用地址线作为片选信号，不需要片选译码器，利用片内地址之外的地址线选作为芯片的片选信号；
- 用在存储容量小、存储芯片也较小的系统中；
- 缺点1：整个存储器的地址常常不连续；
- 缺点2：同一单元可对应不同的地址，形成地址重叠；

地址译码

（2）全地址译码

- 除去用作片内译码的低位地址后，把全部高位地址进行译码来产生片选信号；
- 用在较大的系统中；
- 提供了对全部存储空间的寻址能力；
- 存储单元地址是唯一的、不存在地址重叠问题；
- 需要较多的译码逻辑；

地址译码

（3）部分地址译码

- 除去用作片内译码的低位地址后，将高位地址的一部分进行译码来产生片选信号；
- 它将存储器空间分成许多块，避免了部分译码不能充分利用存储空间的缺点。这些存储器块有时候被称为页；
- 应用举例：将具有64K存储空间分成16块，每块为4K字节，这样只需利用$A_{12}-A_{15}$四根高位地址线译码产生16个译码控制信号。使用块地址译码的优点是某一设备所占用的存储空间不超过一块；

地址译码的实现方法

（1）使用组合逻辑门电路实现地址译码

✓如与门、或门、与非门、或非门等；

✓使用灵活；

（2）使用集成译码器实现地址译码

✓如74LS138译码器；

✓集成度高；

有些情况下需要两种方法组合使用。

教材例题讲解

图5.33 全译码法8088与6264构成32KB存储空间

图5.34 部分译码法8088与6116构成8KB存储空间

图5.35 8088与ROM/RAM综合

图5.36 8088与ROM/RAM综合

图5.39 全译码法8086与6116构成8KB存储空间

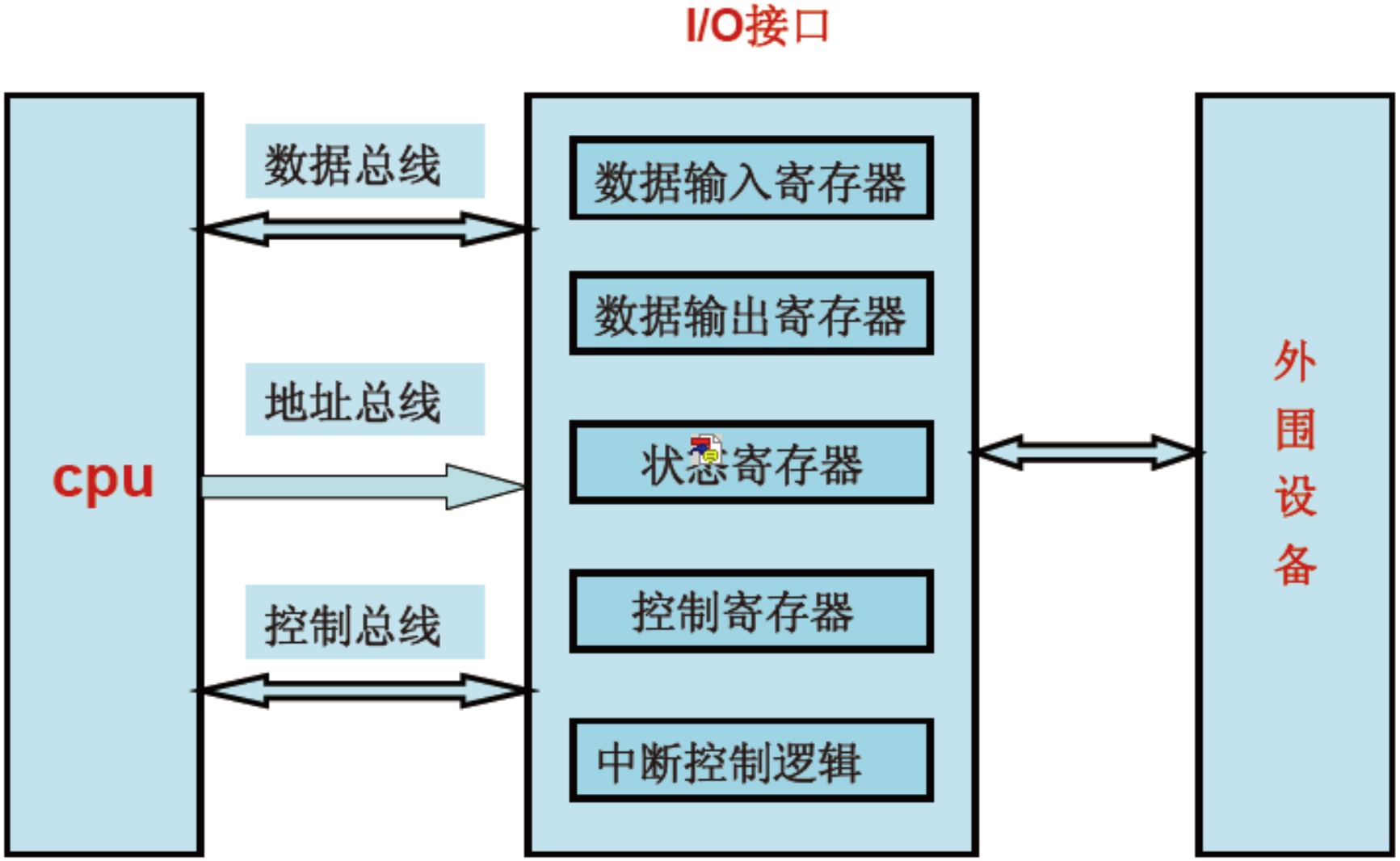
图5.40 8086与ROM/RAM综合

I/O接口

什么是接口

- I/O接口
  - 与CPU和I/O设备相连，实现CPU与外设之间数据传输的电路，在总线和外设之间实现相容性变换并提供数据缓冲能力。
  - 两个部分：
    - 对内：与总线相连，都很相似
    - 对外：与外设相连，差异较大

接口的基本结构


I/O接口
cpu
数据总线
地址总线
控制总线
数据输入寄存器
数据输出寄存器
状态寄存器
控制寄存器
中断控制逻辑
外围设备

接口的功能

- 数据缓冲
- 信息输入输出
- 信息格式转换
- 联络和中断管理
- 译码选址
- 电平转换
- 时序控制
- 可编程
- 错误检测

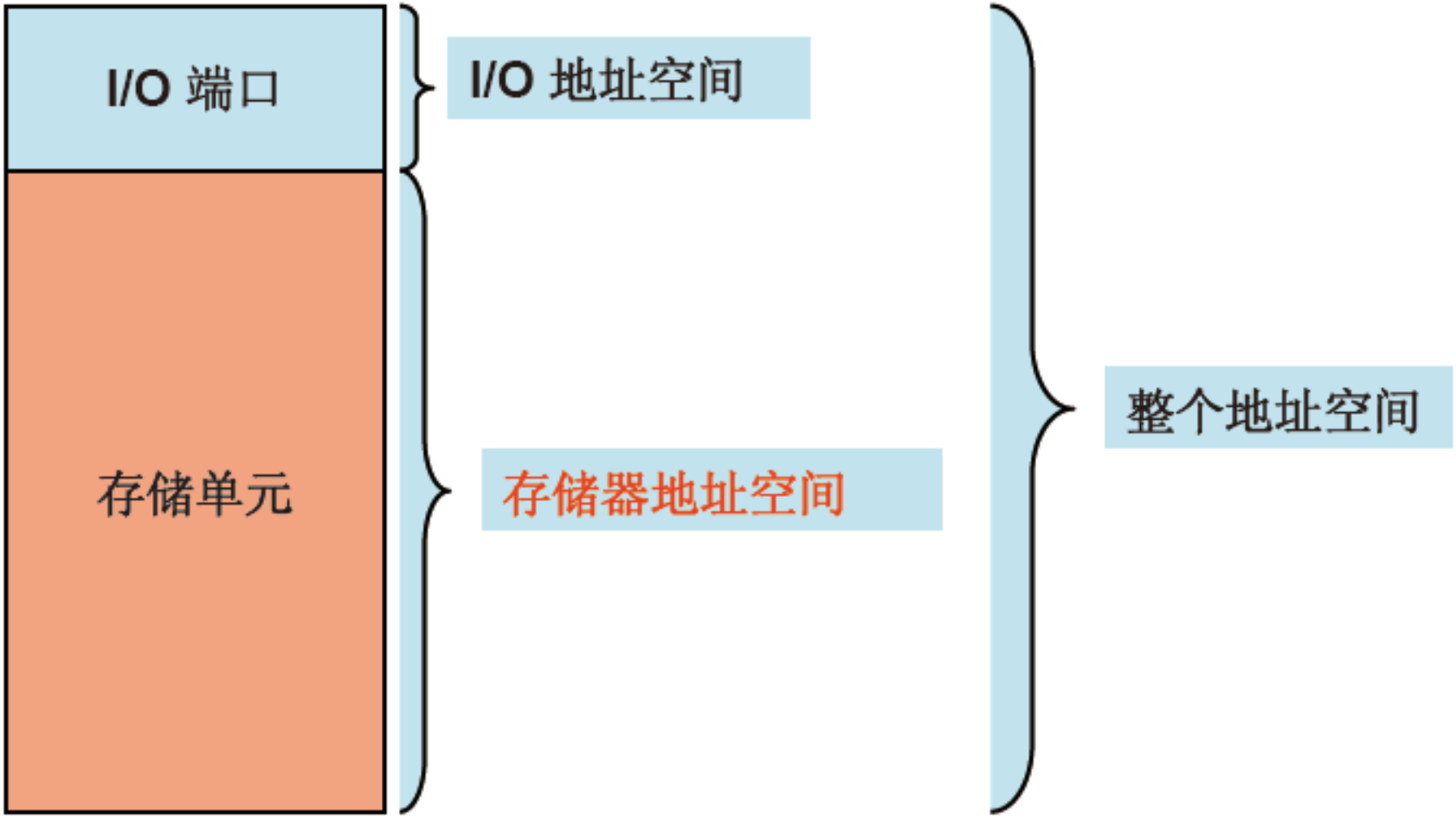
交换信息的类型

- 数据信息：
  - 数字量、模拟量、开关量；
- 状态信息：
  - 即反映外设当前工作状态的信息，输入装置是否准备好的信息；在输出时，输出装置是否空闲等状态信息；
- 控制信息：
  - 控制输入输出装置的启动或停止等。

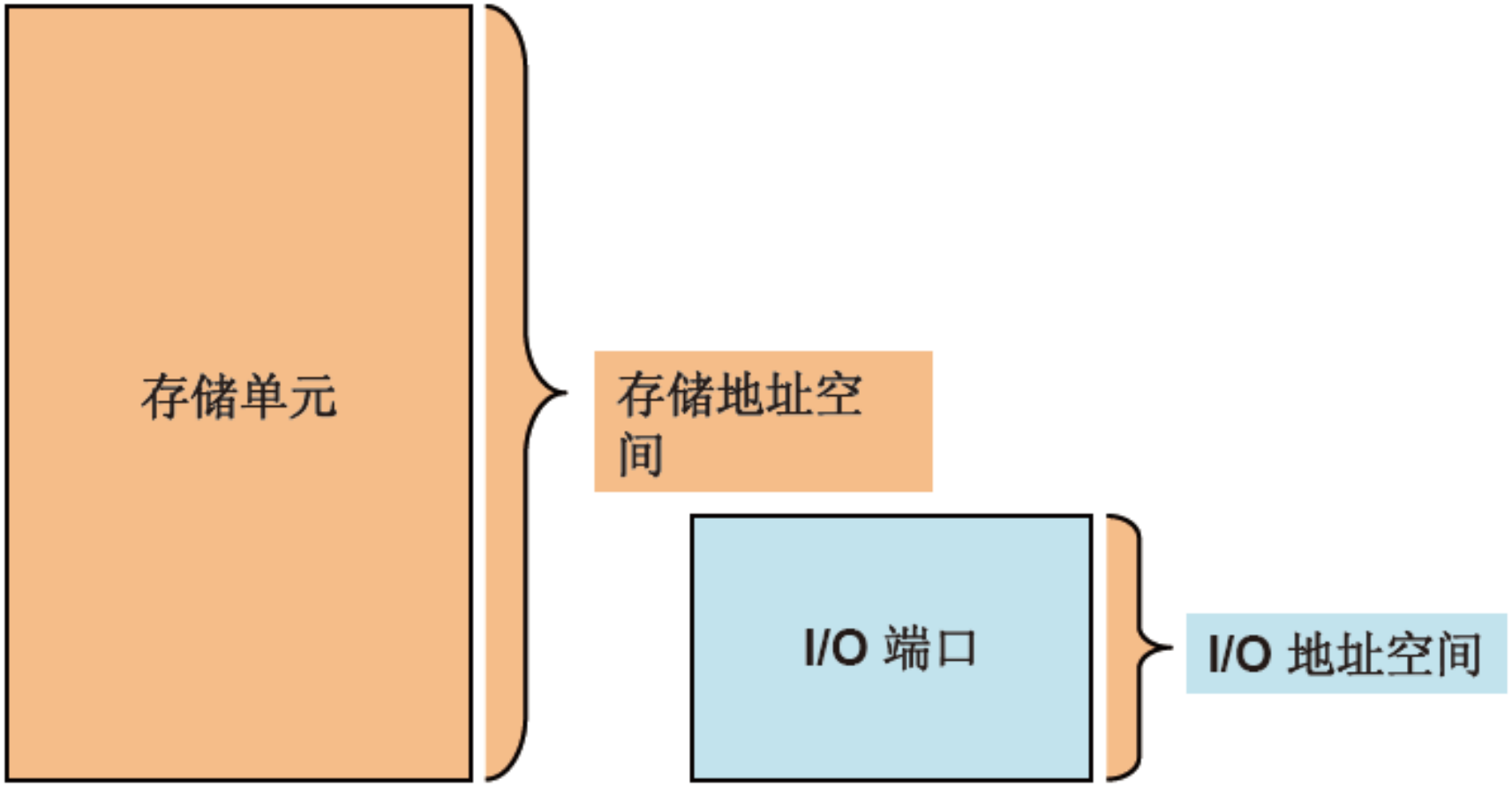
端口编址方式

- 输入输出接口包含一组称为I/O端口的寄存器。为了让CPU能够访问这些I/O端口，每个I/O端口都需有自己的端口地址(或端口号)。
- 在一个微型计算机系统中，如何编排这些I/O接口的端口地址，即所谓I/O端口的编址方式。
- 常见的I/O端口编址方式有两种:
  - 一种是I/O端口和存储器统一编址，也称存储器映像的I/O(Memory　Mapped I/O)方式；
  - 另一种是I/O端口和存储器分开编址，也称I/O映像的I/O(I/O Mapped I/O)方式。

统一编址方式



单独编址



CPU和外设之间的数据传送方式

主机与外设之间传送数据的方式大致可分为如下几种：

（1）程序方式

分为：无条件传送和条件传送方式（查询方式）

（2）中断传送方式

（3）直接数据传送方式（DMA）

（2）中断传送方式

使用查询方式，CPU必须检测接口电路的状态寄存器，如果设备未准备好，CPU就要不断地查询，降低了CPU的运行效率；

中断方式：当外设作好传送准备后，主动向CPU请求中断，CPU响应中断后在中断处理程序中与外设交换数据。若外设未准备好，CPU可以执行其他程序，提高了CPU的利用率；

每条指令完成后，CPU均可响应中断，因此当设备准备好时，可及时与CPU交换数据，提高了实时性。

（3）DMA传送方式

DMA=Direct Memory Access－直接存储器访问

对于高速外设（如磁盘、高速A/D），中断方式不能满足数据传输速度的要求；

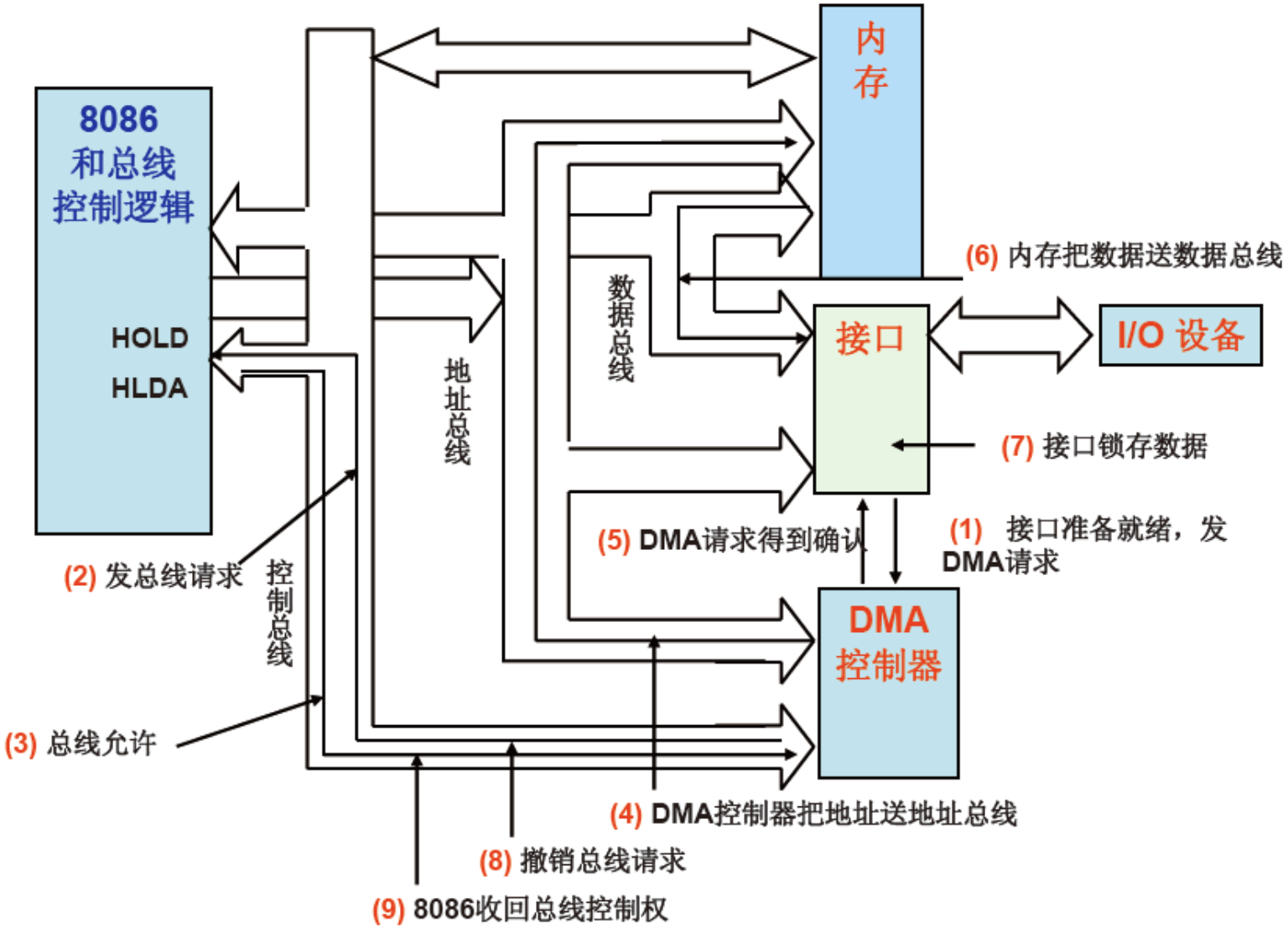
DMA方式是一种由专门的硬件电路执行I/O的数据传送方式，它可以让外设接口直接与内存进行高速的数据传送，而不必经过CPU。这种专门的硬件电路称为DMA控制器，简称DMAC。

DMA控制器工作方式

- 单字节传输方式
  - 在单字节传输方式下，DMA控制器每次请求总线只传送一个字节数据，传送完后即释放总线控制权。
  - 在此方式下，总线控制权处于CPU与DMA控制器交替控制之中，其间，总线控制权经过多次交换。
- 块传输方式(也称成组传输方式)
  - 块传输方式是指DMA控制器每次请求总线即连续传送一个数据块，待整个数据块全部传送完成后再释放总线控制权。

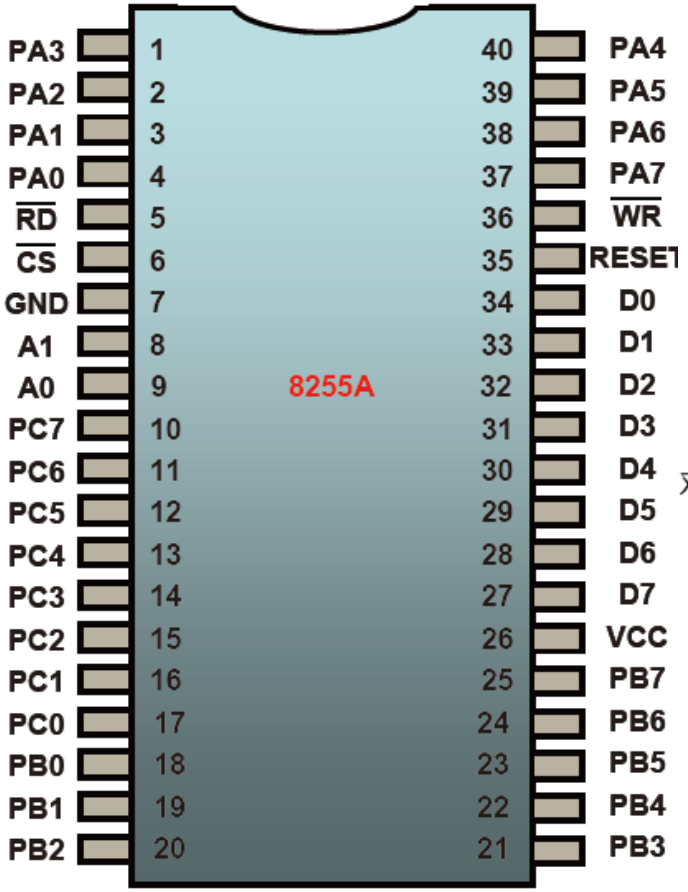
- 请求传输方式
  - 每传输完一个字节，DMA控制器都要检测由I/O接口发来的“DMA请求”信号是否仍然有效，如果该信号仍有效，则继续进行DMA传输；
  - 否则，就暂停传输，交还总线控制权给CPU，直至“DMA请求”信号再次变为有效，数据块传输则从刚才暂停的那一点继续进行下去。

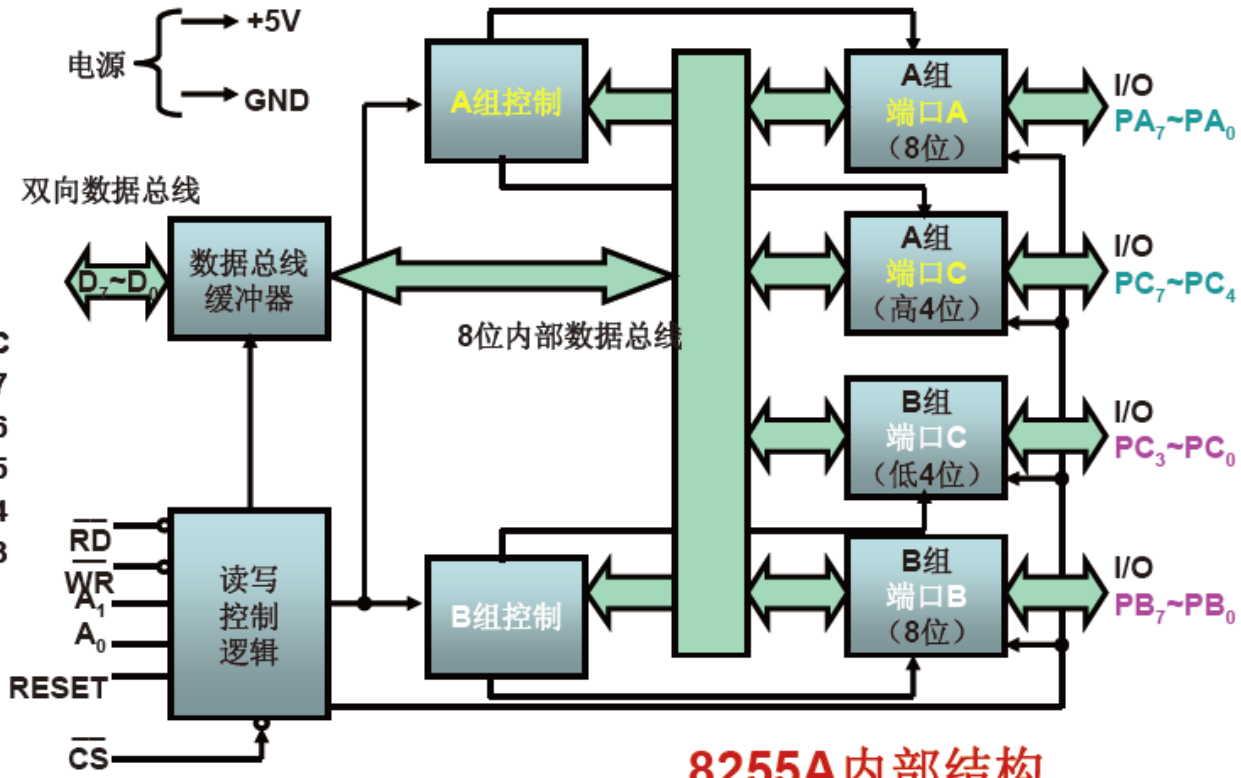
DMA工作过程


8086
和总线
控制逻辑
HOLD
HLDA
内
存
数据总线
地址总线
接口
I/O 设备
(6) 内存把数据送数据总线
(7) 接口锁存数据
(1) 接口准备就绪，发DMA请求
(5) DMA请求得到确认
DMA
控制器
(2) 发总线请求
控制总线
(3) 总线允许
(4) DMA控制器把地址送地址总线
(8) 撤销总线请求
(9) 8086收回总线控制权

8255A

8255A结构及功能





8255A内部结构

8255A端口选择和基本操作

A1	A0	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	输入操作（读）
0	0	0	1	0	端口A→数据总线
0	1	0	1	0	端口B→数据总线
1	0	0	1	0	端口C→数据总线
					输出操作（写）
0	0	1	0	0	数据总线→端口A
0	1	1	0	0	数据总线→端口B
1	0	1	0	0	数据总线→端口C
1	1	1	0	0	数据总线→控制字寄存器
					无操作情况
X	X	X	X	1	数据总线为三态（高阻）
1	1	0	1	0	非法操作
X	X	1	1	0	数据总线为三态（高阻）

7.1.6 8255A应用举例

应用1：多片8255的连接

某微机系统有两片8255芯片：J1和J2；

J1、J2的A_1/A_0分别和系统地址总线的A_2/A_1连接；

其它控制信号分别连在一起，然后与系统相关信号连接；

系统靠片选信号来区分对J1和J2的访问；

要求：J1、J2工作方式为

J1：端口A，方式0，输出

端口B，方式0，输入

端口C高4位，输出

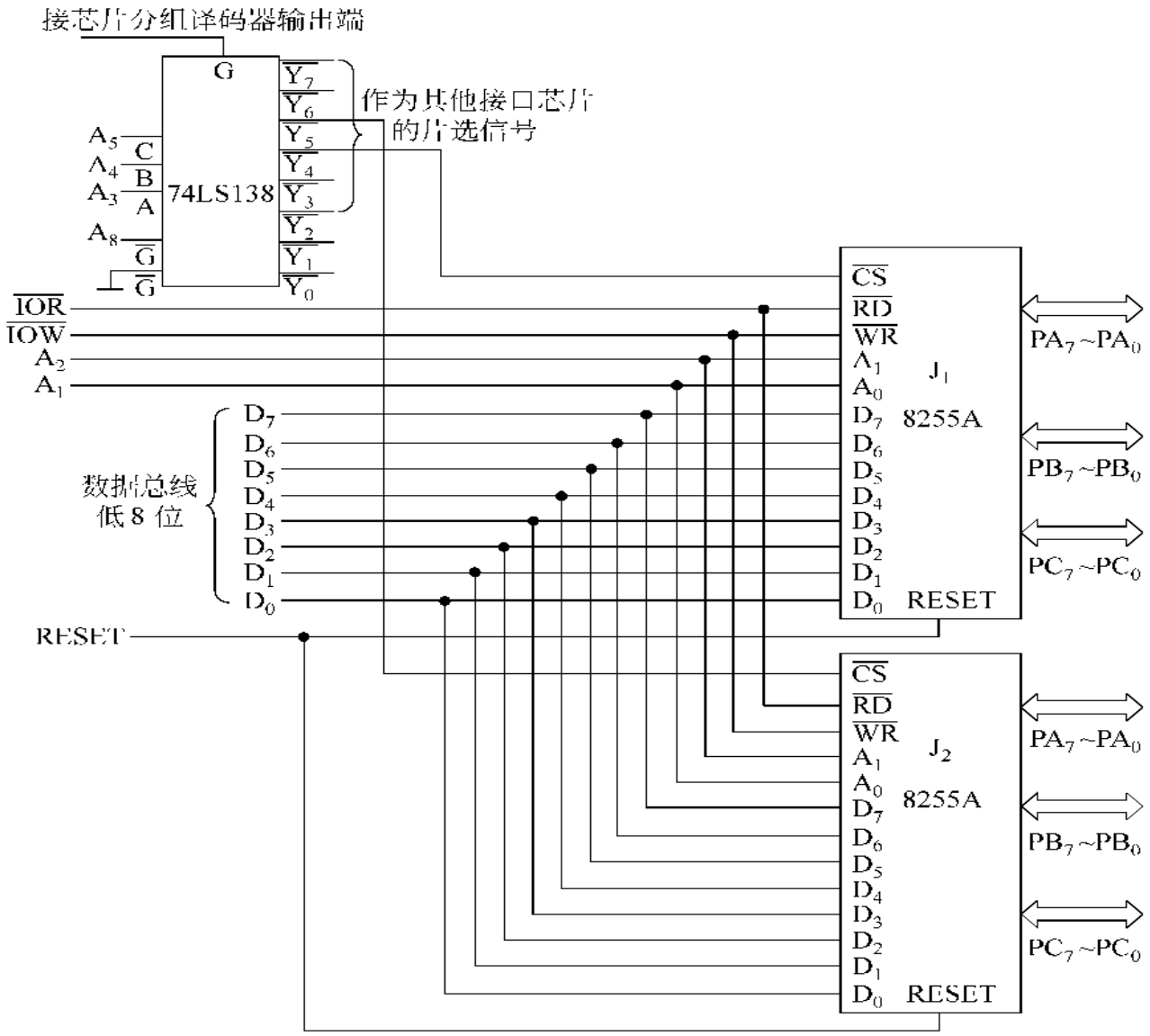
端口C低4位，输入

J2：端口A，方式0，输入

端口B，方式1，输出

端口C高4位，输出

应用1：多片8255的连接



多片8255的连接

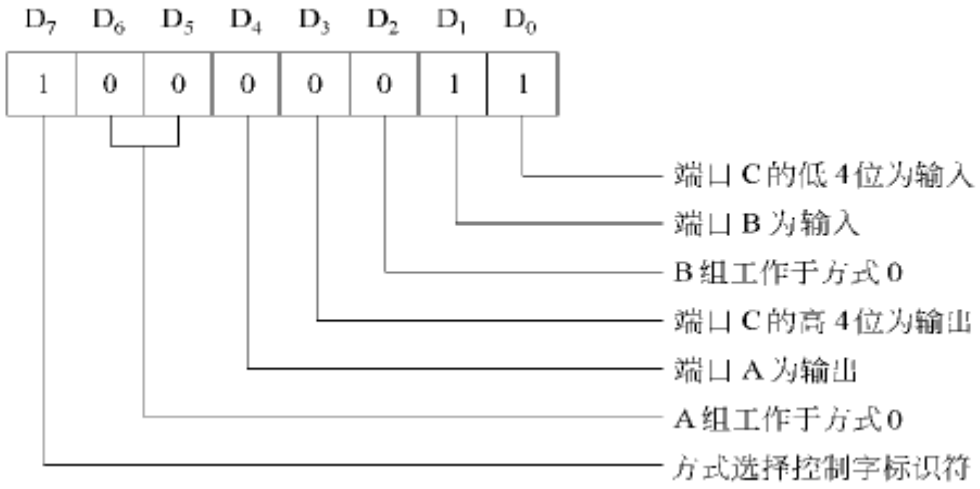
J1、J2端口地址：

芯　片	端口名称	地址（十六进制）	芯　片	端口名称	地址（十六进制）
	端口A	00E0		端口A	00E8
J_1	端口B	00E2	J_2	端口B	00EA
（8255A）	端口C	00E4	（8255A）	端口C	00EC
	控制口	00E6		控制口	00EE

一个端口可以有多个地址；

多片8255的连接

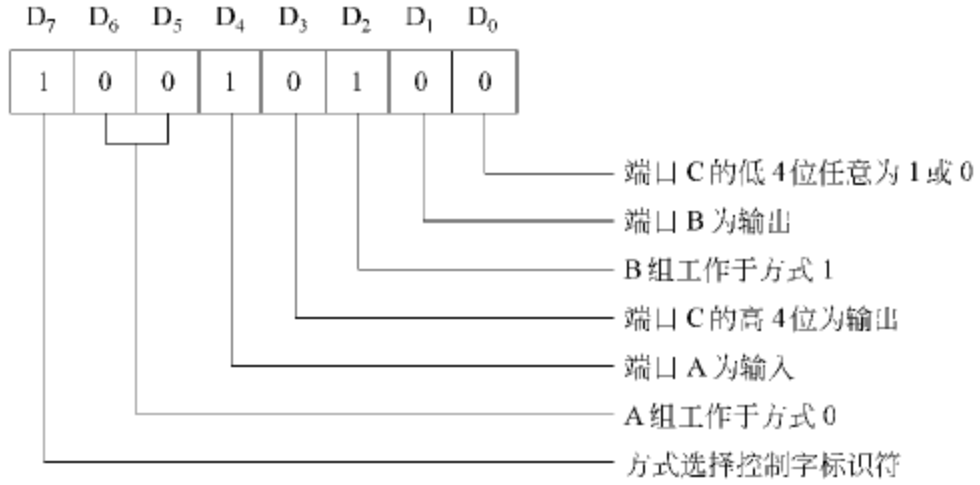
J1方式控制字：



方式设置指令：

```
MOV   AL,83H
MOV   DX,00E6H
OUT   DX,AL
```

J2方式控制字：

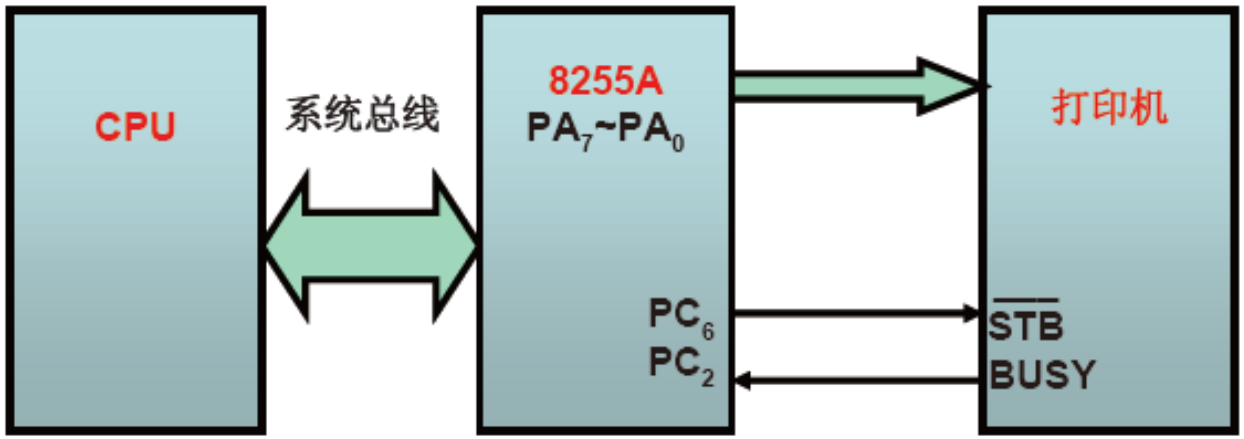



```
MOV   AL,94H
MOV   DX,00EEH
OUT   DX,AL
```

应用2：8255工作于方式0

8255A工作于方式0，用8255A作为以查询方式工作的打印机接口

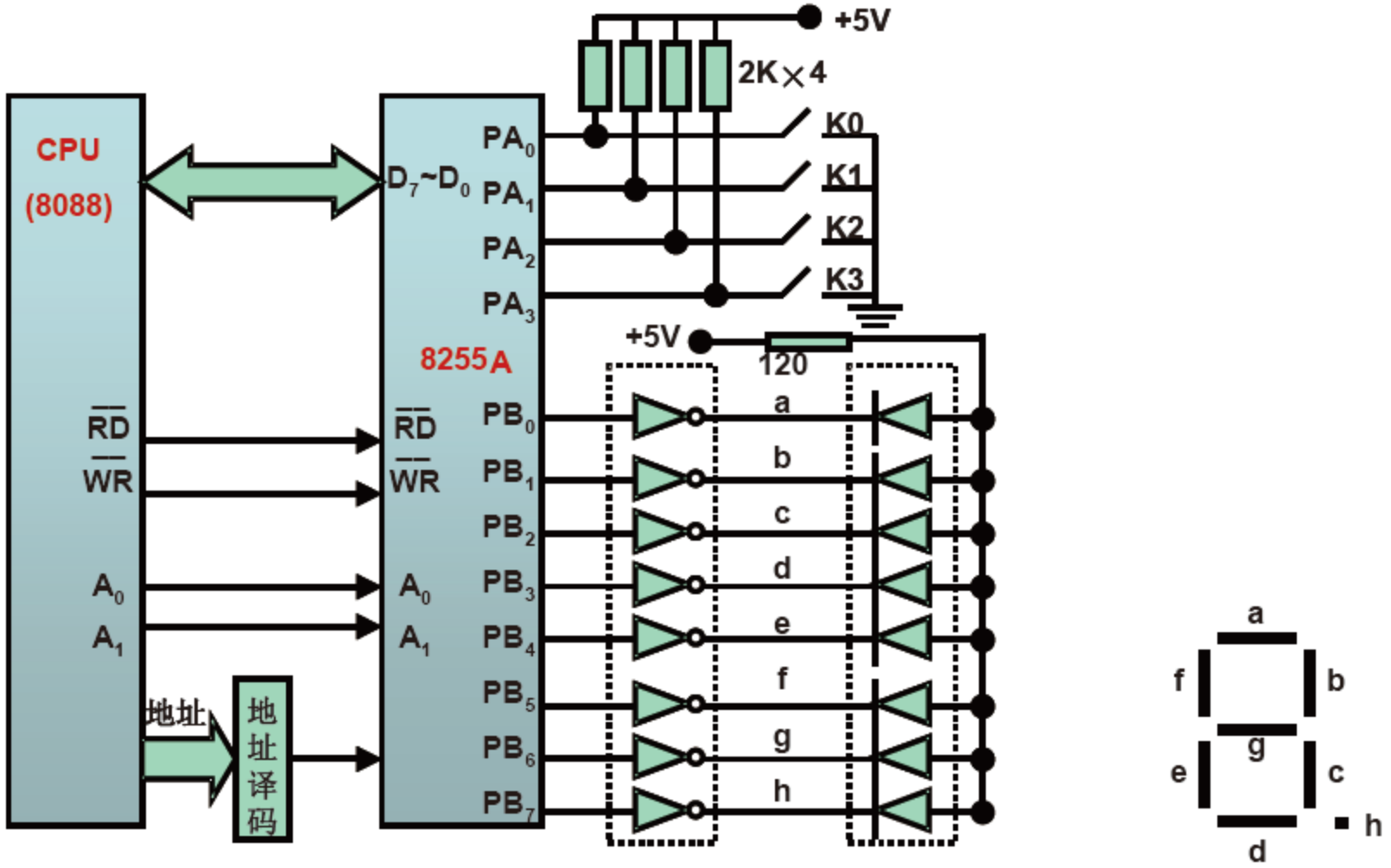
- 设打印字符存放在内存2000H单元。
- 8255A的端口地址为：端口A－D0H；端口B－D1H；端口C－D2H；控制口－D3H



应用3：

8255A工作于方式0，利用8255A将外设开关的二进制状态从端口A输入，经程序转换为对应的LED段选码(字形码) 后，再从端口B输出到LED显示器。

设8255A的端口地址为：端口A－－D0H，
端口B－－D1H，端口C－－D2H，
控制口－－D3H。



（a）连线图

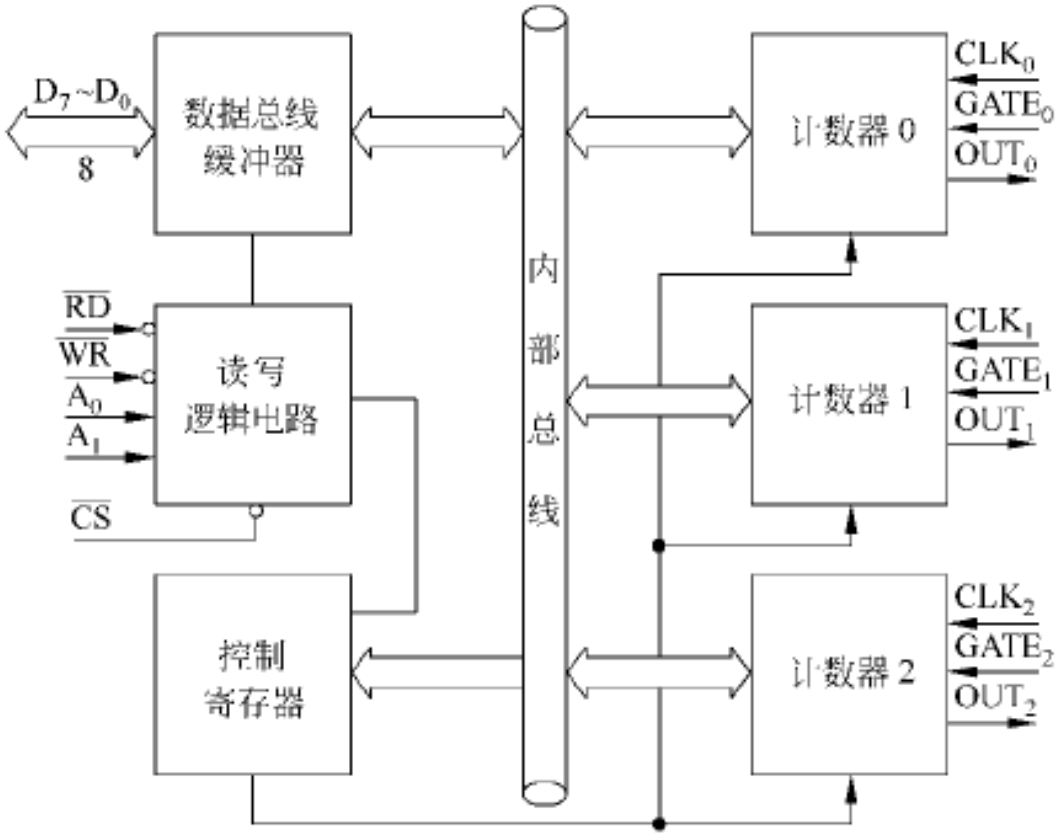
(b)LED显示器

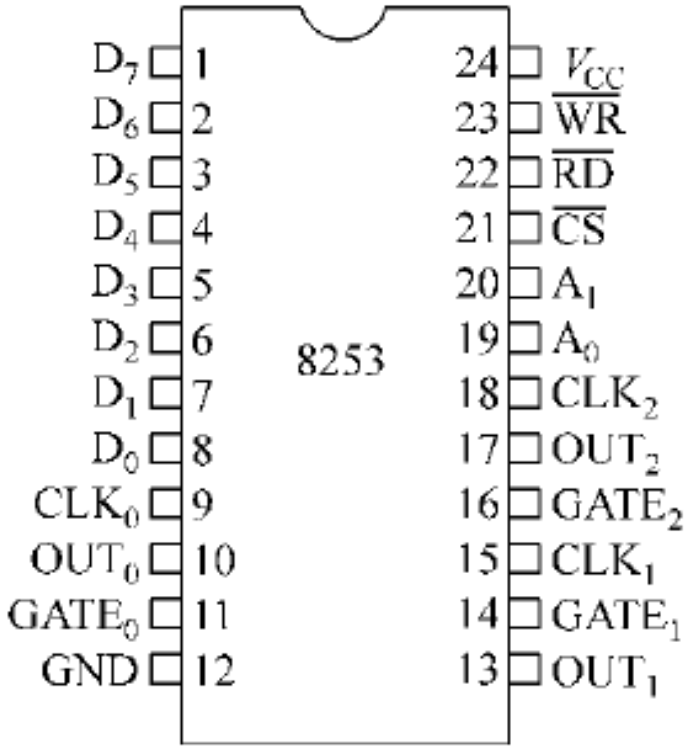
8253

计时方法

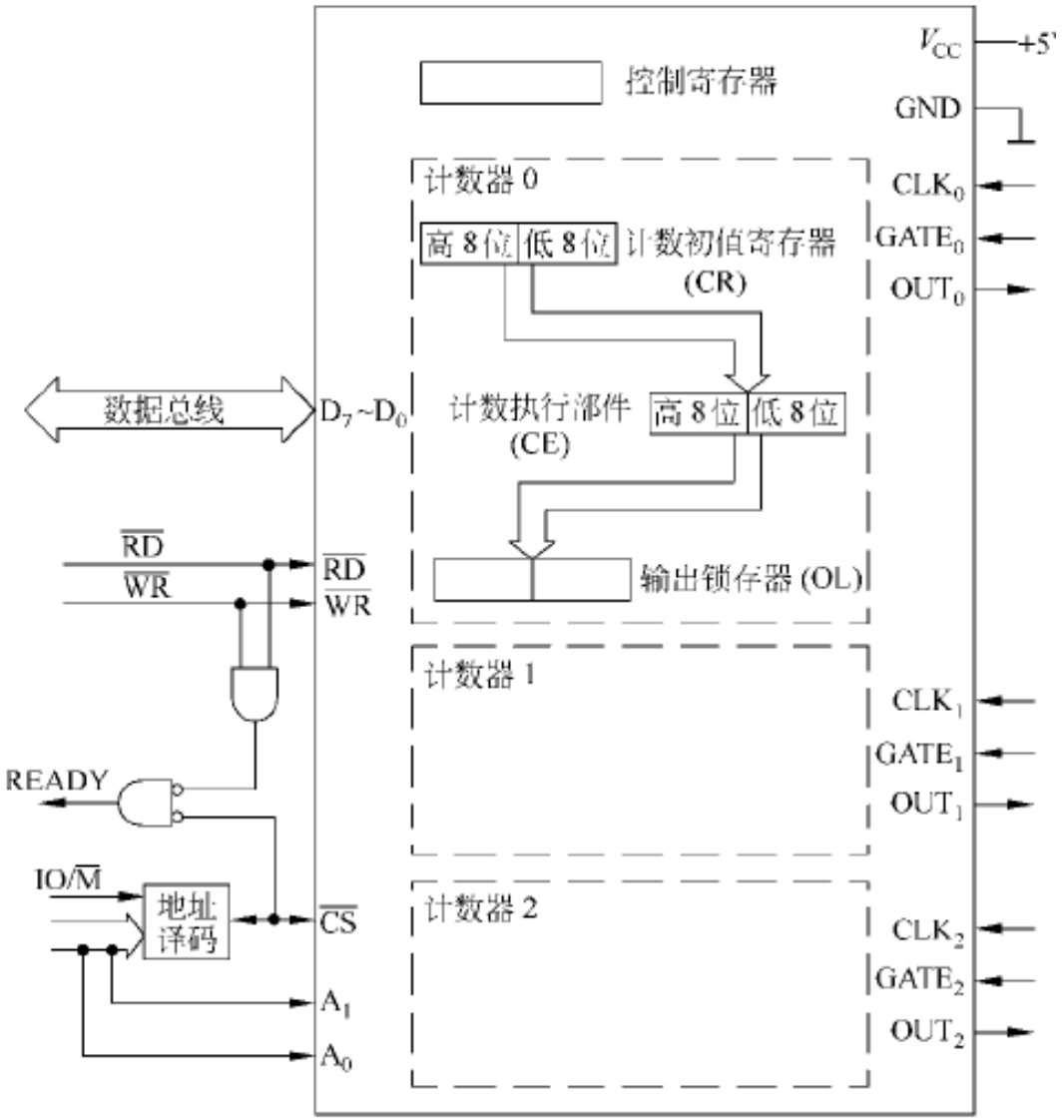
- 软件计时
  - 由CPU执行指令序列所花费的时间来构成一定的时间间隔，从而达到定时的目的。
- 硬件计时
  - 用专门的多谐振荡器或单稳态触发器
  - 不可编程
    - 用电路、改变定时需改变硬件
  - 可编程
    - 可用软件的方法(通过初始化编程)设定或调整定时范围，常用芯片Intel 8253/8254

8253内部结构


D_7~D_0
8
数据总线
缓冲器
$\overline{RD}$
$\overline{WR}$
A_0
A_1
$\overline{CS}$
读写
逻辑电路
控制
寄存器
内
部
总
线
计数器 0
计数器 1
计数器 2
CLK_0
$GATE_0$
OUT_0
CLK_1
$GATE_1$
OUT_1
CLK_2
$GATE_2$
OUT_2


D_7 1
D_6 2
D_5 3
D_4 4
D_3 5
D_2 6
D_1 7
D_0 8
CLK_0 9
OUT_0 10
$GATE_0$ 11
GND 12
8253
24 V_{CC}
23 $\overline{WR}$
22 $\overline{RD}$
21 $\overline{CS}$
20 A_1
19 A_0
18 CLK_2
17 OUT_2
16 $GATE_2$
15 CLK_1
14 $GATE_1$
13 OUT_1

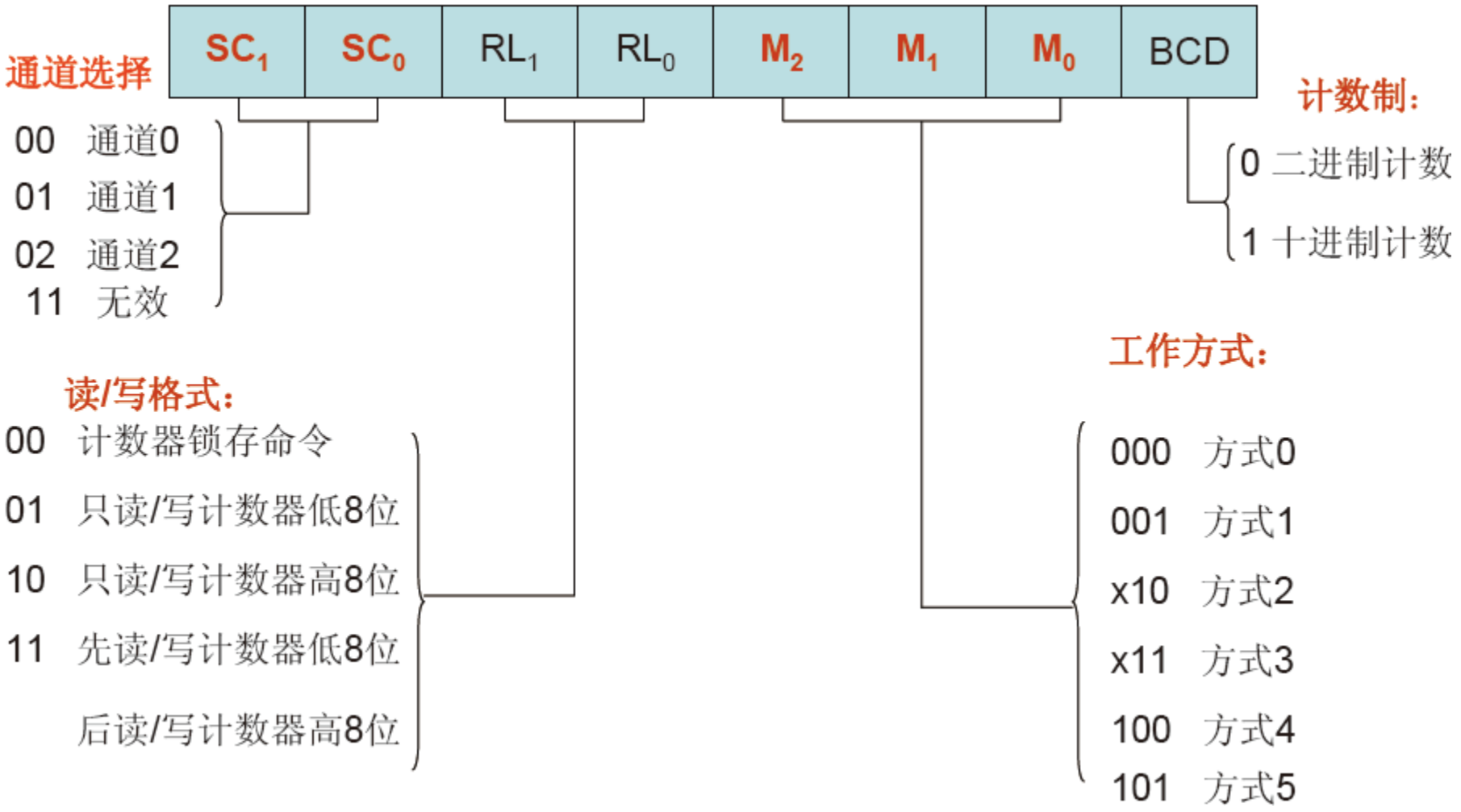
8253编程结构


V_{CC}
+5'
GND
控制寄存器
计数器 0
CLK_0
$GATE_0$
OUT_0
高 8 位
低 8 位
计数初值寄存器
(CR)
数据总线
D_7~D_0
计数执行部件
(CE)
高 8 位
低 8 位
$\overline{RD}$
$\overline{WR}$
输出锁存器 (OL)
计数器 1
CLK_1
$GATE_1$
OUT_1
READY
IO/$\overline{M}$
地址
译码
$\overline{CS}$
计数器 2
CLK_2
$GATE_2$
OUT_2
A_1
A_0

8253的读写逻辑

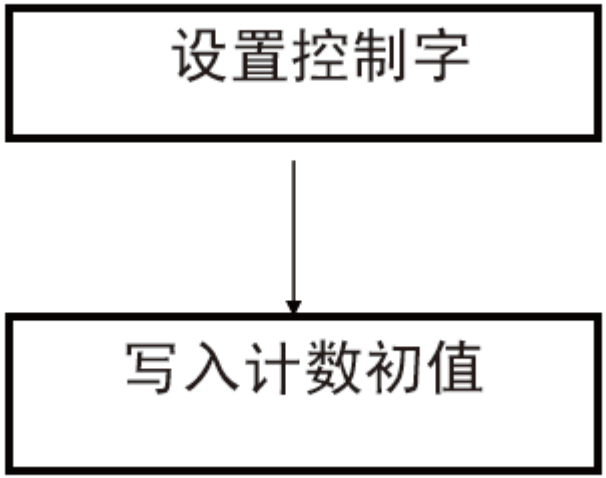
$\overline{CS}$	$\overline{RD}$	$\overline{WR}$	A1	A0	寄存器选择和操作
0	1	0	0	0	写通道0计数初值寄存器CR0
0	1	0	0	1	写通道1计数初值寄存器CR1
0	1	0	1	0	写通道2计数初值寄存器CR2
0	1	0	1	1	写控制寄存器
0	0	1	0	0	读通道0输出锁存器OL0
0	0	1	0	1	读通道1输出锁存器OL1
0	0	1	1	0	读通道2输出锁存器OL2

控制字


SC1 SC0 RL1 RL0 M2 M1 M0 BCD
通道选择
00 通道0
01 通道1
02 通道2
11 无效
计数制：
0 二进制计数
1 十进制计数
读/写格式：
00 计数器锁存命令
01 只读/写计数器低8位
10 只读/写计数器高8位
11 先读/写计数器低8位
后读/写计数器高8位
工作方式：
000 方式0
001 方式1
x10 方式2
x11 方式3
100 方式4
101 方式5

注意

- 必须先写控制字，再写初值
- 必须按控制字D5,D4位规定的格式进行写入。



8253工作方式总结

- 方式2(分频器)、方式4(软件触发选通)和方式5(硬件触发选通)，它们的输出波形相同，都是宽度为1个CLK周期的负脉冲。
  - 区别是，方式2是自动重复工作的，而方式4需由软件(设置计数值)触发启动，方式5需由门控GATE信号触发启动。
- 方式5(硬件触发选通)与方式1(硬件触发单稳)，触发信号相同，但输出波形不同.
  - 方式1输出为宽度是N个CLK周期的负脉冲(计数过程中输出为低);
  - 方式5输出为宽度是1个CLK周期的负脉冲(计数过程中输出为高)。

8253工作方式总结

- 在6种工作方式中，只有方式0在写入控制字后输出为低；其余5种方式，都是在写入控制字后输出为高。
- 6种工作方式中的任一种方式，只有在写入计数初值后才能开始计数。
  - 方式0、2、3、4都是写入计数初值后，计数过程就开始了。
  - 方式1、5在写入计数初值后，需由外部GATE信号的触发启动，才能开始计数过程。
- 6种工作方式中，只有方式2(分频器)和方式3(方波发生器)为自动重复工作方式，其他4种方式都是一次性计数，要继续工作需要重新启动。

例 1

若用8253的计数通道1，工作在方式0，按8位二进制计数，计数初值为128，则初始化编程如下：

(1) 确定通道控制字-------50H

(2) 8位计数初值----------80H

设8253的端口地址为48H～4BH，

例 2

若用通道0，工作在方式1，按十进制（BCD码）计数，计数初值为2010，则初始化编程如下：

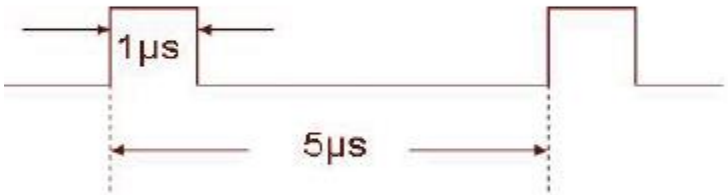
(1) 确定通道控制字------33H

(2) 计数初值低8位为10，高8位为20。

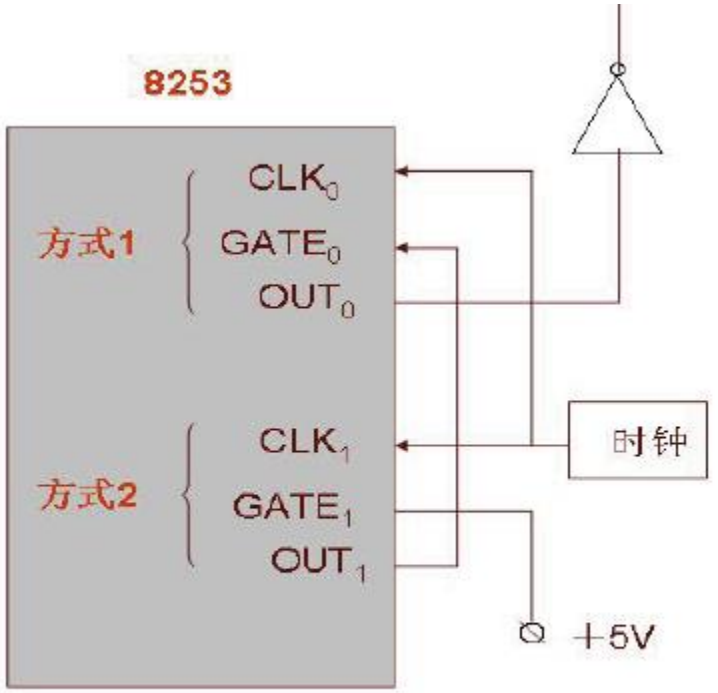
若8253的端口地址同例1

例 3

- 8253用作脉冲信号发生器。
- 可用8253产生如图 (a)所示的周期性脉冲信号，其重复周期为5μs，脉冲宽度为1μs。设CLK信号频率为2MHz。

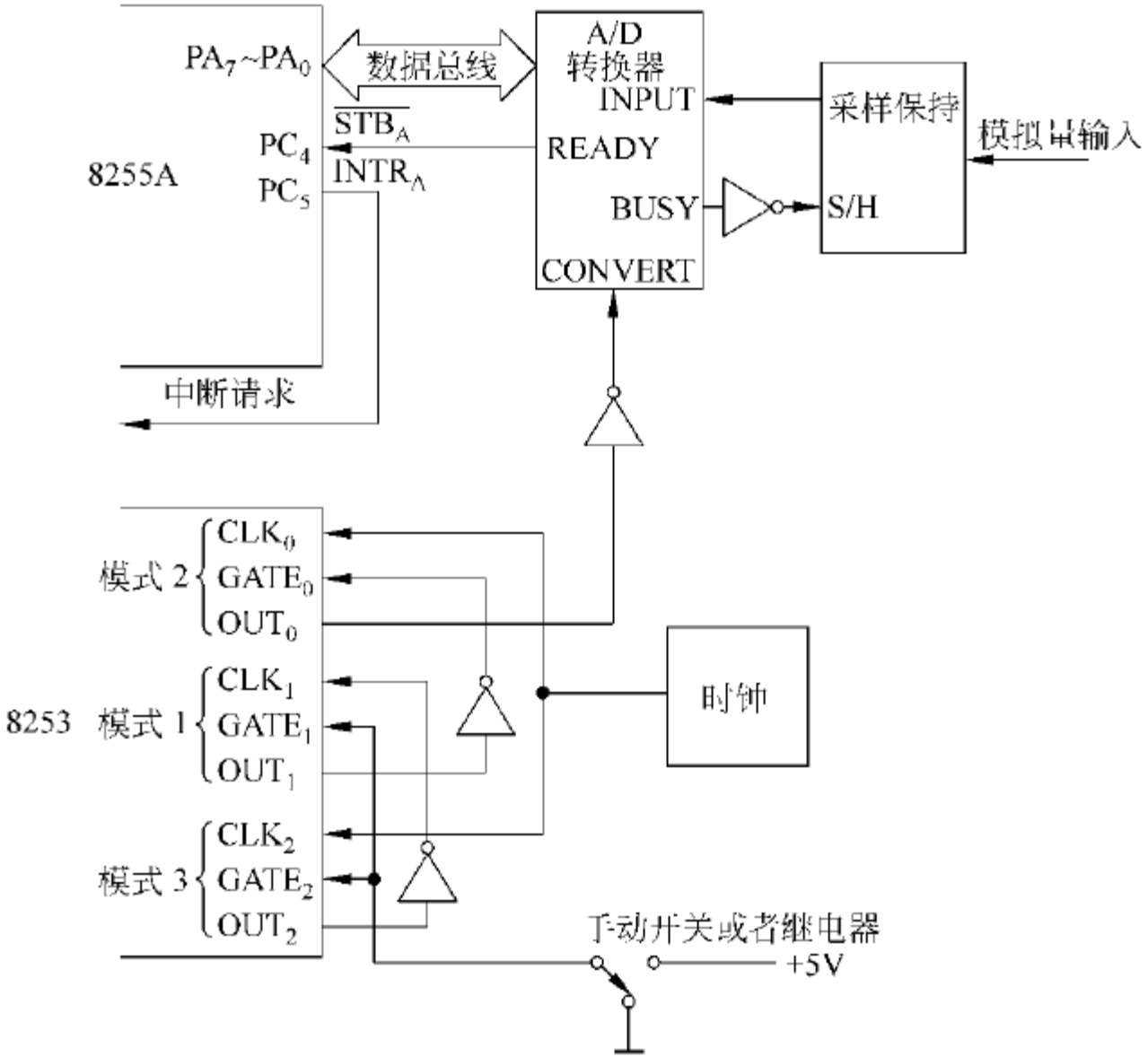


(a) 信号波形图



(b) 连接图

例4 用于信号采样

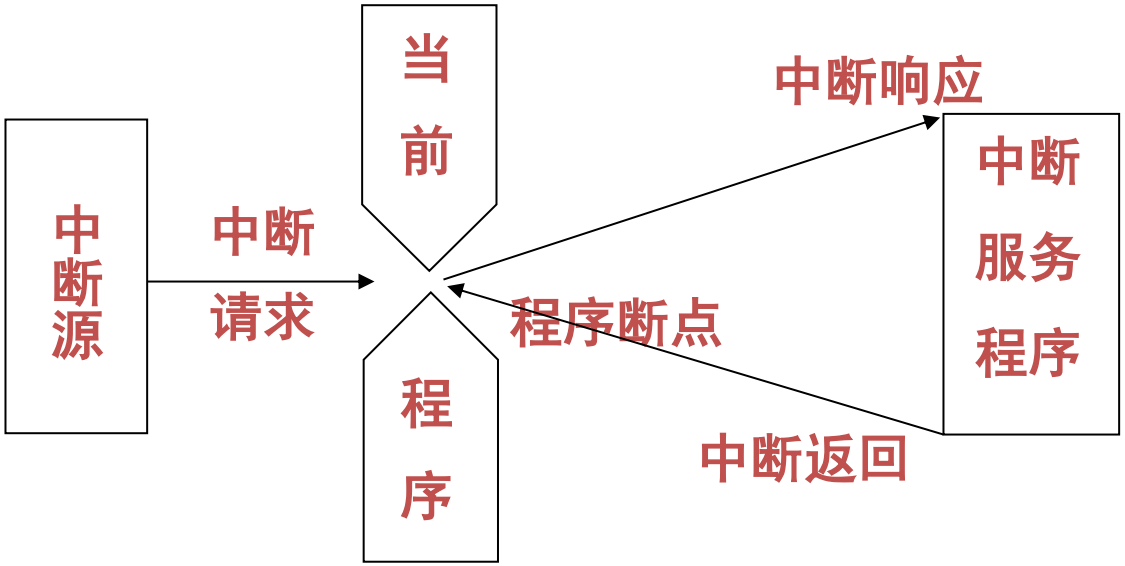

PA7~PA0
数据总线
A/D
转换器
INPUT
READY
BUSY
CONVERT
采样保持
S/H
模拟量输入
STBA
PC4
INTRA
PC5
8255A
中断请求
模式 2
CLK0
GATE0
OUT0
8253
模式 1
CLK1
GATE1
OUT1
模式 3
CLK2
GATE2
OUT2
时钟
手动开关或者继电器
+5V

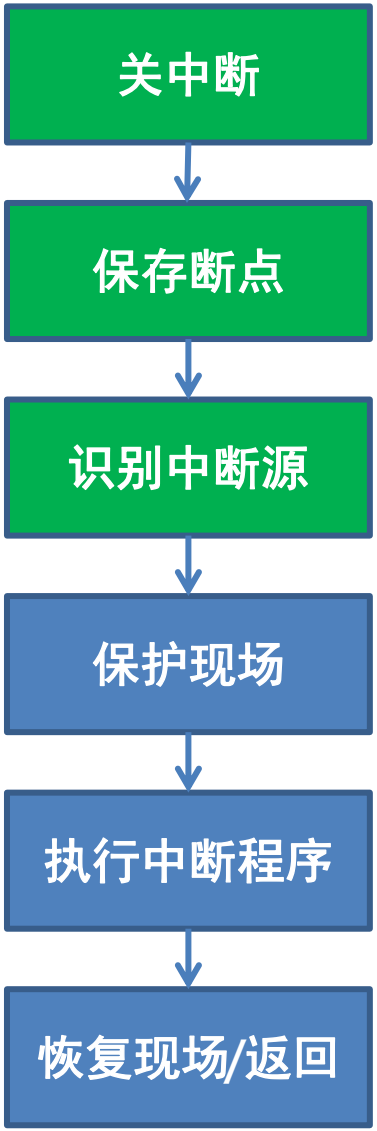
中断技术

中断

- 设计目标
  - 最初目标：解决高速CPU和低速I/O之间的问题，提高CPU的利用率
  - 使CPU具有实时响应和处理随机事件的能力
- 定义
  - 在程序运行时，系统外部、内部或现行程序本身若出现紧急事件，处理器必须立即强行中止现行程序的运行，改变机器的工作状态并启动相应的程序来处理这些事件，然后再恢复原来的程序运行，这一过程称为中断

中断处理过程

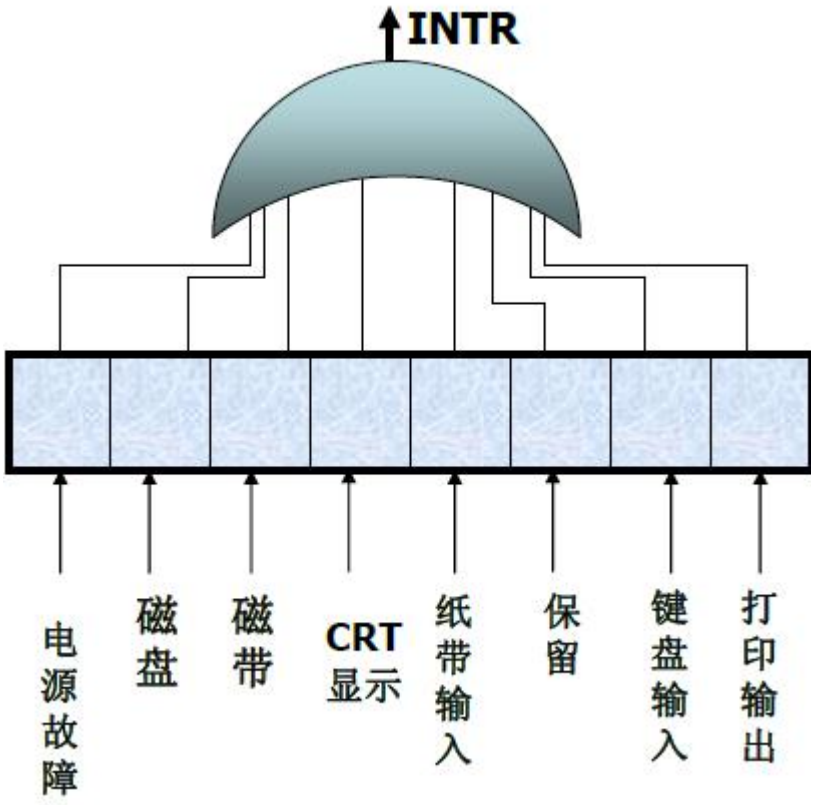

中断源
中断请求
当前程序
中断响应
中断服务程序
程序断点
中断返回

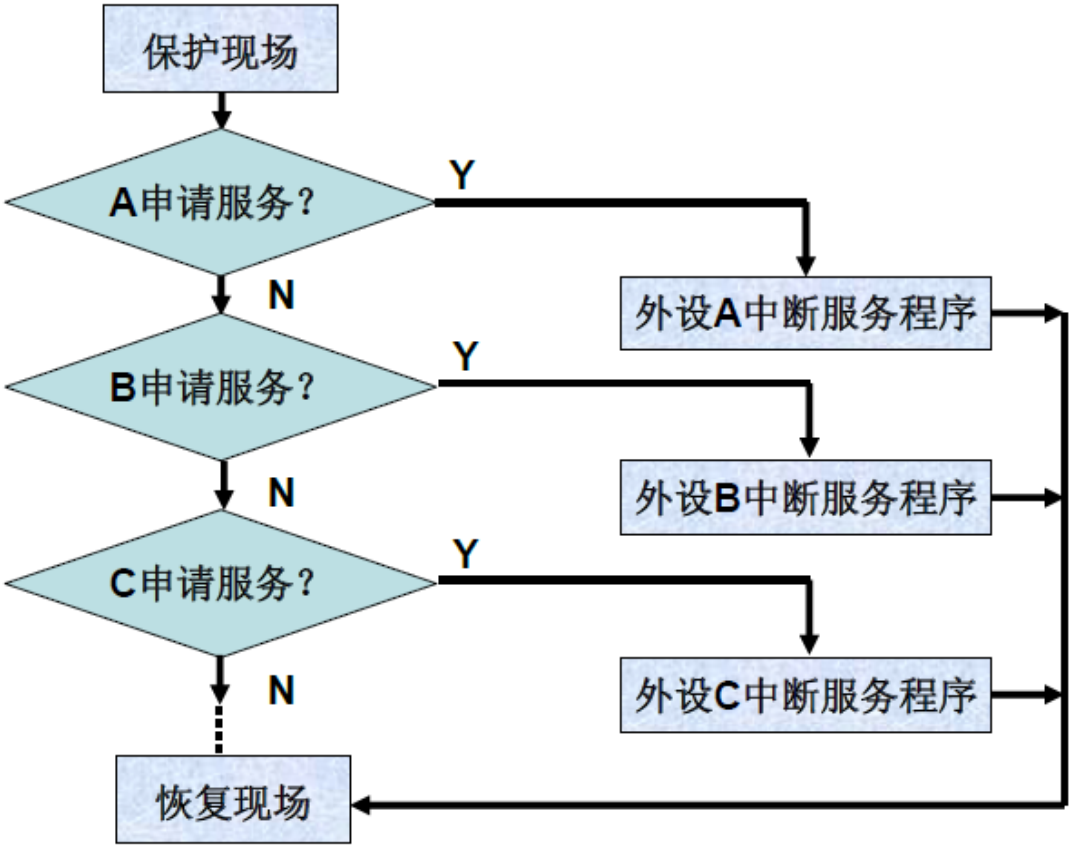

关中断
保存断点
识别中断源
保护现场
执行中断程序
恢复现场/返回

中断优先级和嵌套

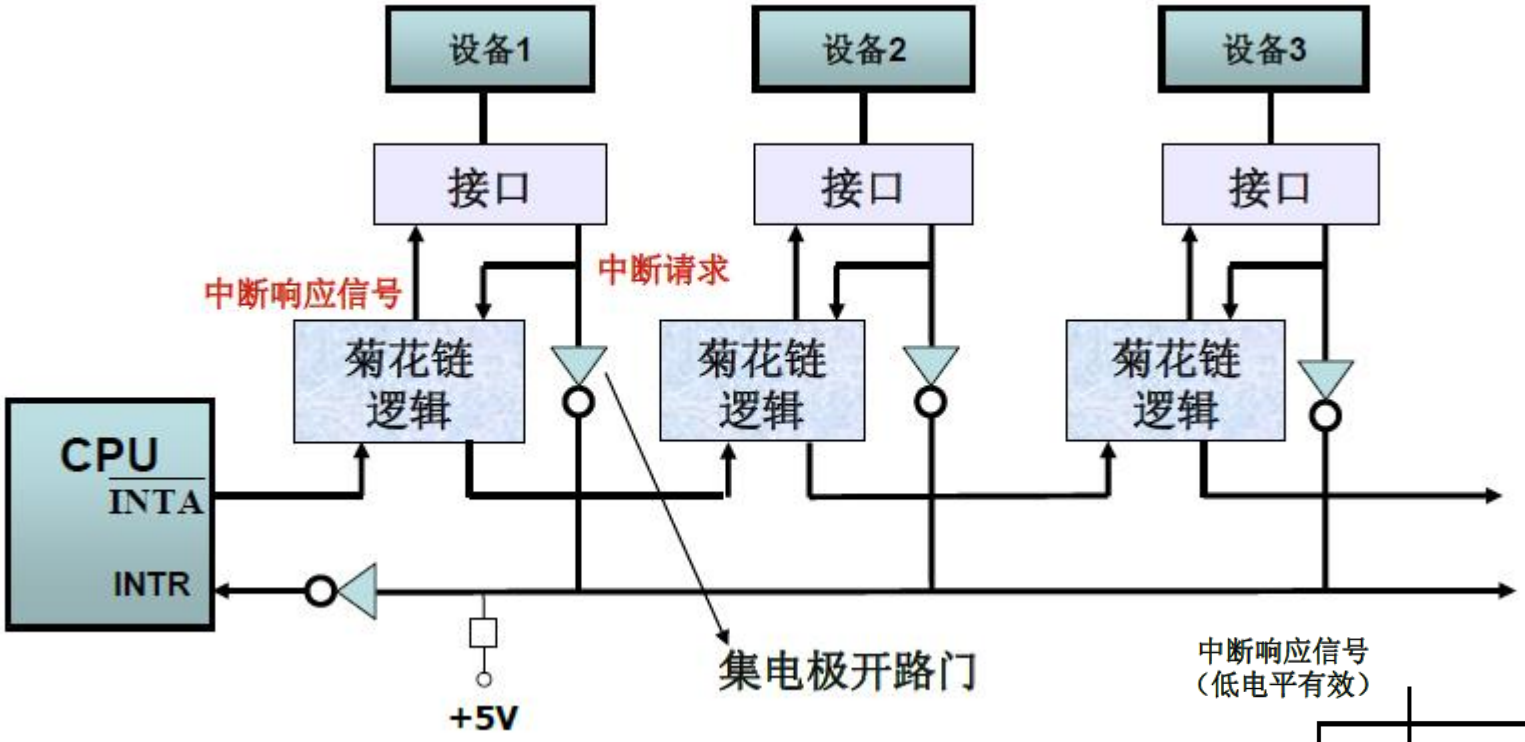
- 中断优先级
  - 在实际系统中，多个中断请求可能同时出现，但中断系统只能按一定的次序来响应和处理，这时CPU必须确定服务的次序，即根据中断源的重要性和实时性，照顾到操作系统处理的方便，对中断源的响应次序进行确定。
  - 这个响应次序称为中断优先级（priority）。
- 中断级依次降低：
  - 内部中断和异常＞软件中断＞外部非屏蔽中断＞外部可屏蔽中断

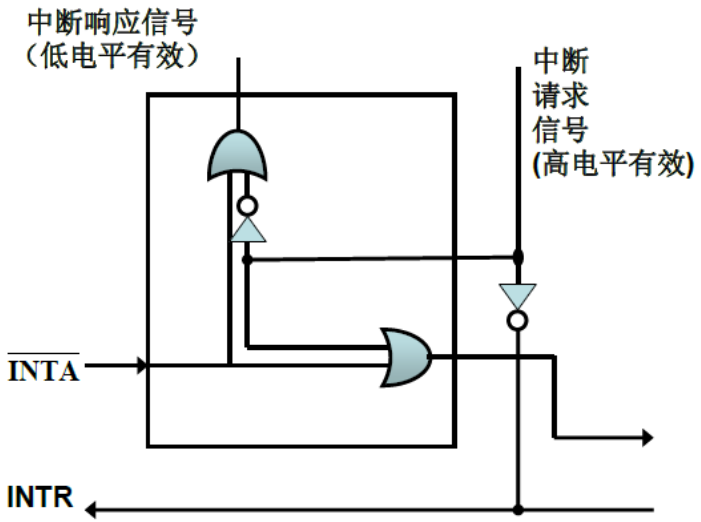
软件查询确定中断优先级



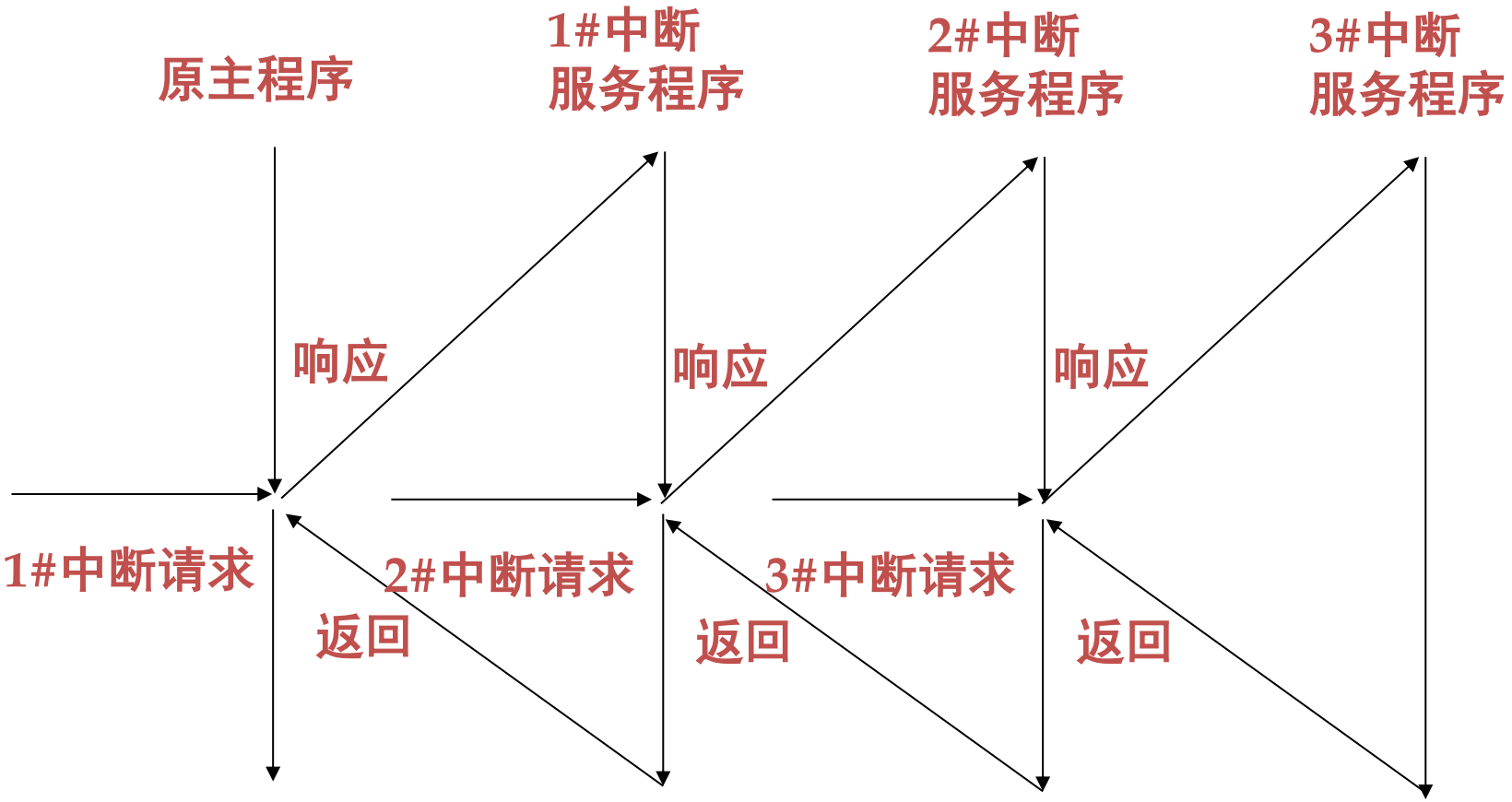


菊花链排队

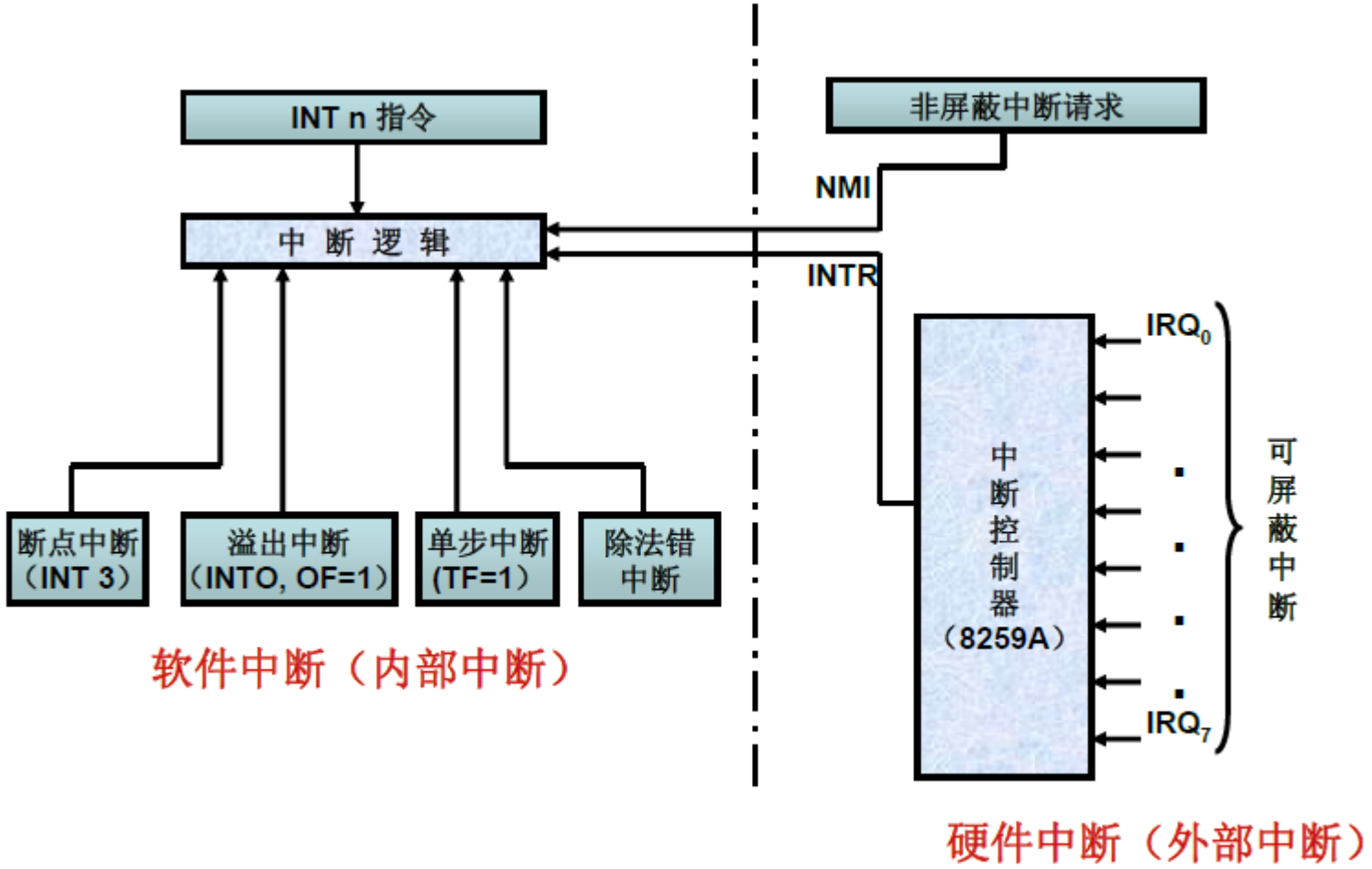

设备1
设备2
设备3
接口
接口
接口
中断响应信号
中断请求
菊花链
逻辑
菊花链
逻辑
菊花链
逻辑
CPU
INTA
INTR
+5V
集电极开路门


中断响应信号
（低电平有效）
中断
请求
信号
(高电平有效)
INTA
INTR

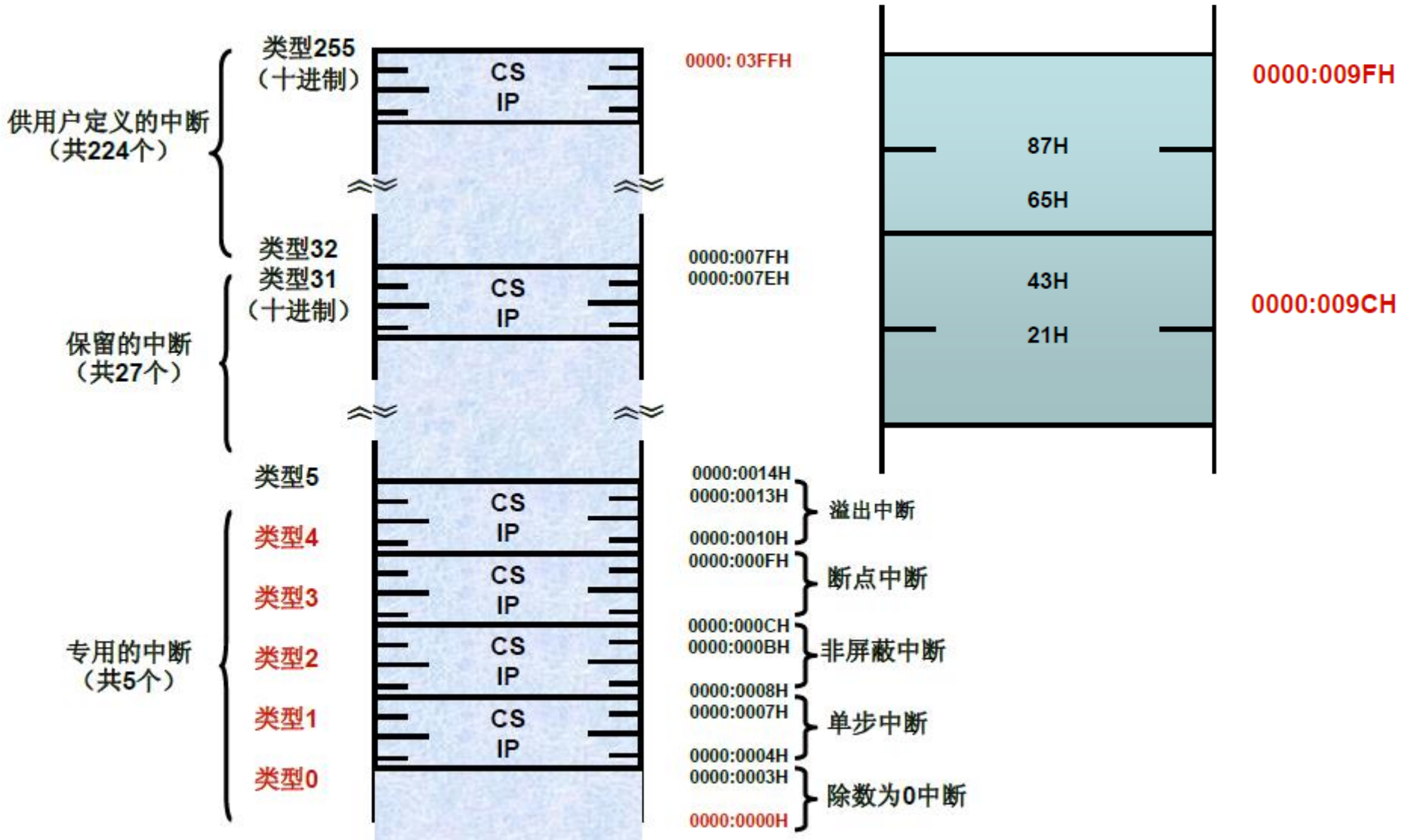
中断嵌套


原主程序
1#中断
服务程序
2#中断
服务程序
3#中断
服务程序
响应
响应
响应
1#中断请求
2#中断请求
3#中断请求
返回
返回
返回

8086实模式中断和处理



中断向量表


类型255
（十进制）
CS
IP
0000: 03FFH
供用户定义的中断
（共224个）
类型32
类型31
（十进制）
CS
IP
0000:007FH
0000:007EH
保留的中断
（共27个）
类型5
CS
IP
0000:0014H
0000:0013H
溢出中断
0000:0010H
类型4
CS
IP
0000:000FH
断点中断
类型3
CS
IP
0000:000CH
0000:000BH
非屏蔽中断
类型2
CS
IP
0000:0008H
0000:0007H
单步中断
类型1
CS
IP
0000:0004H
0000:0003H
除数为0中断
类型0
0000:0000H
专用的中断
（共5个）
0000:009FH
87H
65H
43H
21H
0000:009CH

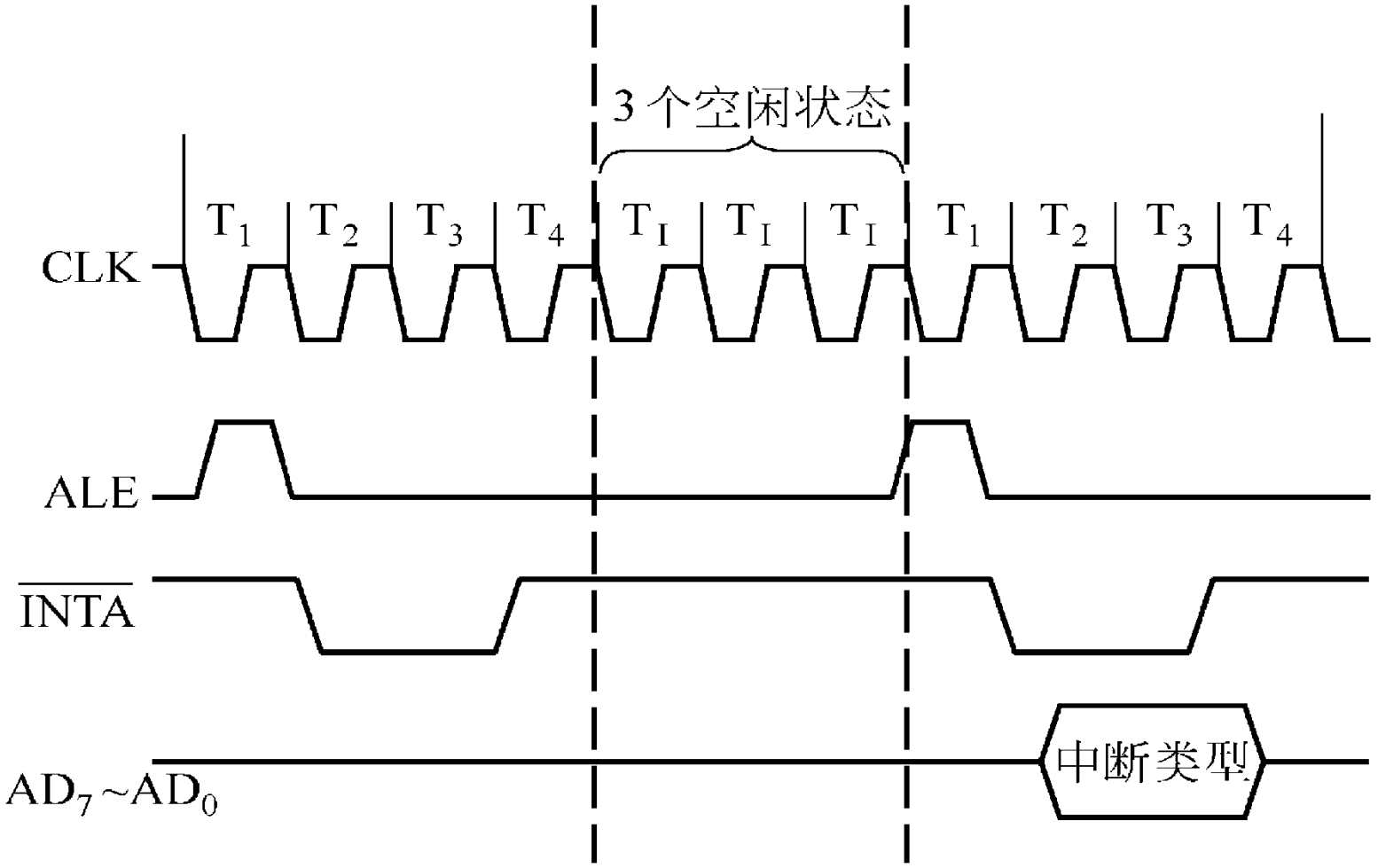
外部中断

- 由外部的中断请求信号启动的中断，称为外部中断,也称硬件中断。
- 80x86 CPU为外部中断提供两条引线，即NMI和INTR，用来输入中断请求信号。

非屏蔽中断

- 从NMI引脚进入的中断为非屏蔽中断，它不受中断允许标志IF的影响。
- 非屏蔽中断的类型码为2，因此，非屏蔽中断处理子程序的入口地址存放在08H、09H、0AH和0BH这4个字节单元中。

可屏蔽中断



内部中断/软中断

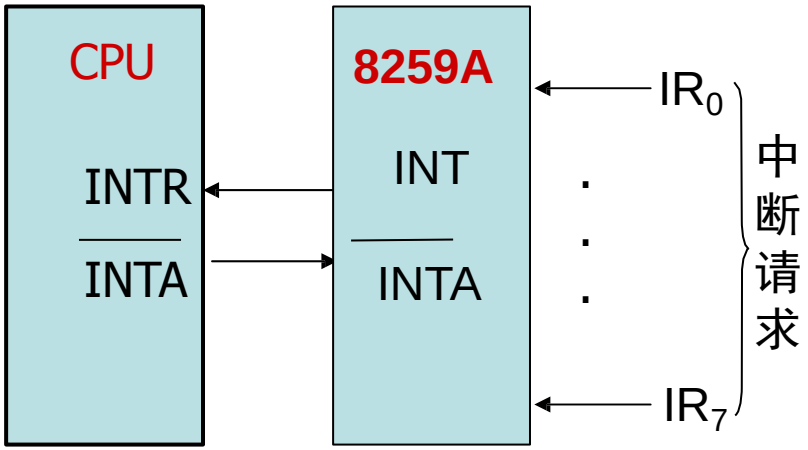
- 内部中断也称软件中断。
- 它是由于CPU执行了INT n(含INT 3)、INTO指令，或者由于除法出错以及进行单步操作所引起的中断，主要包括INT n指令中断、断点中断、溢出中断、除法错中断以及单步中断。

8259A及其应用

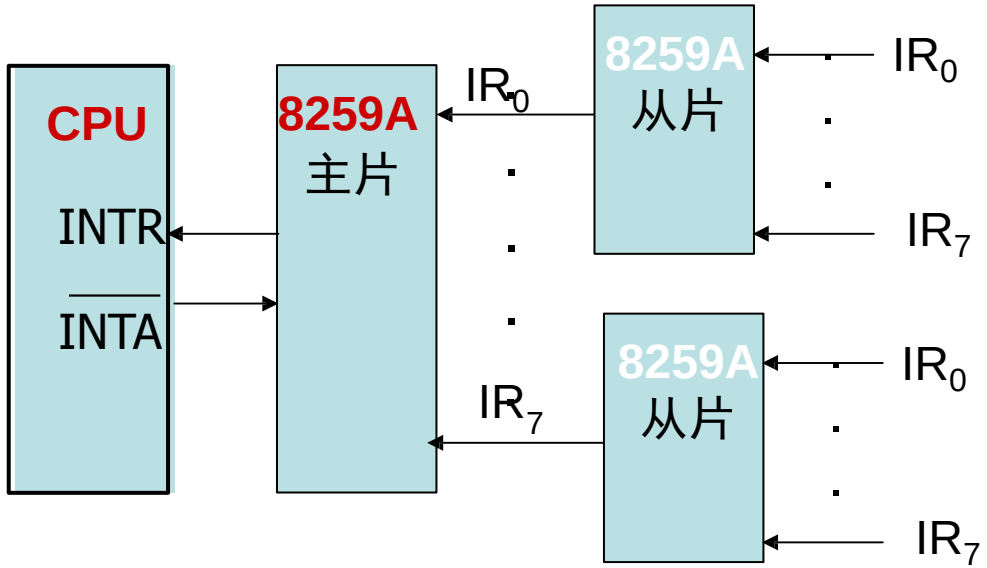
中断控制器8259A的功能与结构

8259A的功能

- 管理和控制80x86的外部中断请求
- 实现优先级判决
- 提供中断类型码
- 屏蔽中 断输入



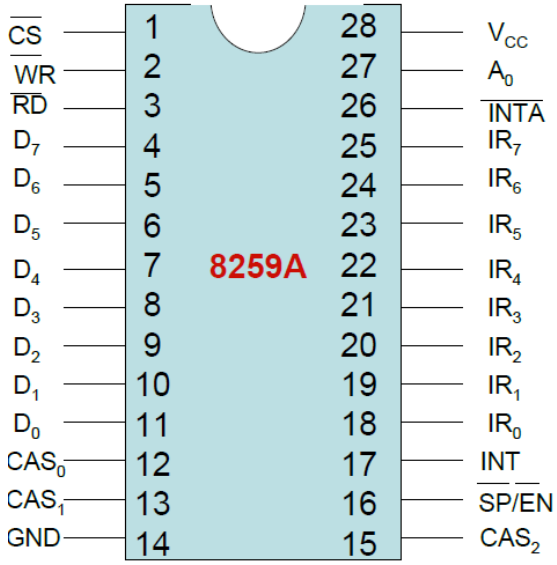
单片

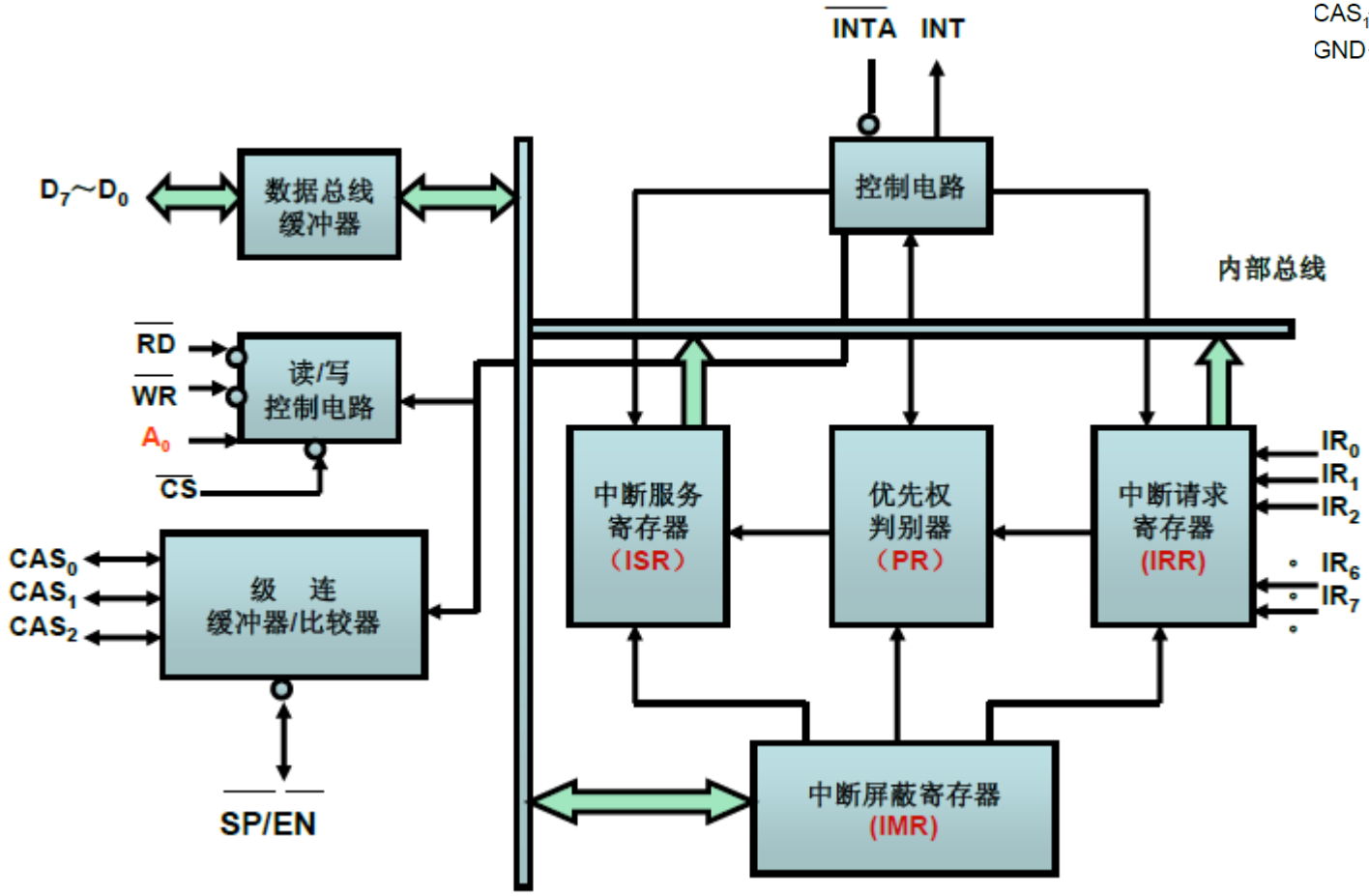


级联

8259A的结构

- 封装形式　28脚双列直插





7个寄存器的寻址问题:

规定：A_0

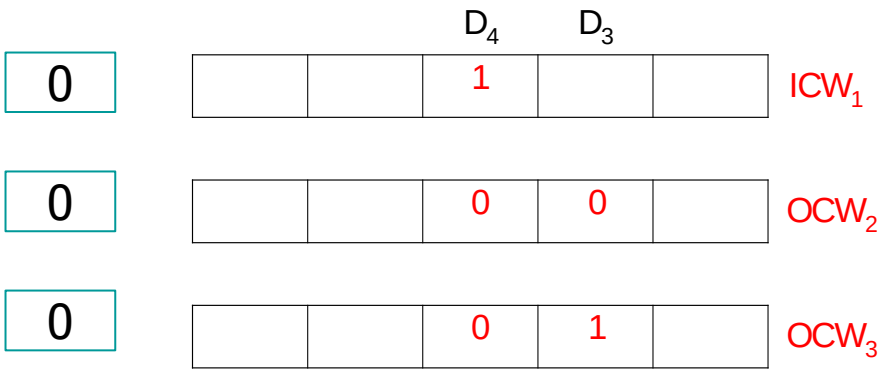
0	ICW_1:用偶地址写入，且D_4=1
1	ICW_2
1	ICW_3
1	ICW_4

} 紧接着ICW_1，用奇地址写入

1	OCW_1:也用奇地址写入，但不紧跟ICW_1
0	OCW_2
0	OCW_3

} 也用偶地址写入，但D_4=0

即：



•采用了专门的"标识位"，以节省输入地址的引脚数(仅用了A_0)

18.2 8259A的工作方式

1. 设置优先级的方式

(1) 全嵌套方式(FULLY NESTED MODE)

- 也称固定优先级方式。在这种方式下，由IR_i引入的中 断请求具有固定的优先级—IR_0(最高)—>IR_7(最低).

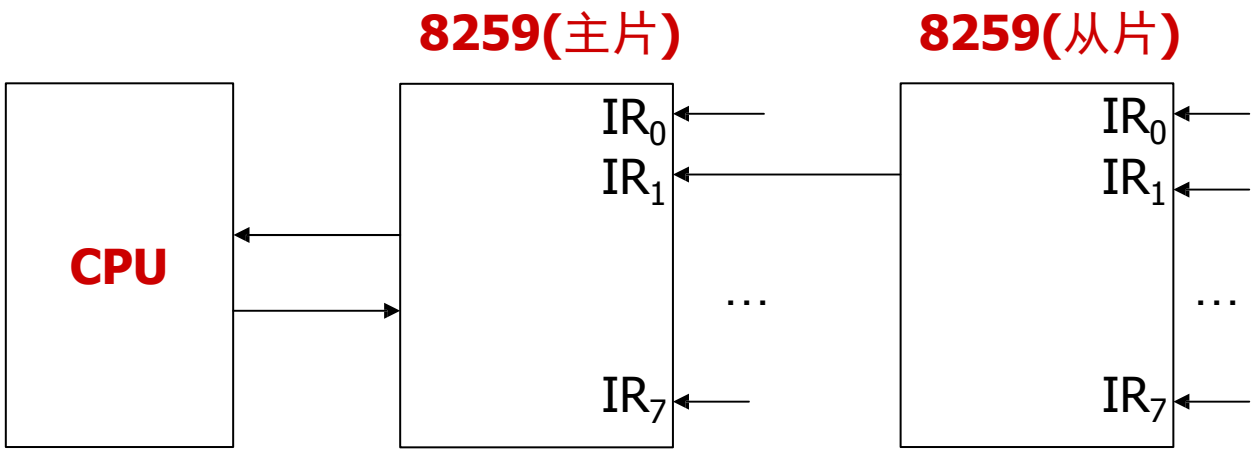
- 当一个中断请求被响应时， ISR中的对应位IS_n被置 “1”，8259A把中断类型码放到数据总线上，然后，进 入中断服务程序。

- 一般情况下(除了"中断自动结束"方式外)，在CPU发出 中断结束命令(EOI)前，此对应位一直保持为"1"—封锁 同级或低级的中断响应，但并不禁止比本级优先级高 的中断响应—实现中断"嵌套"。

(2) 特殊全嵌套方式

(SPECIAL FULLY NESTED MODE－SFNM)

- 在处理某一级中断时，不但允许优先级更高的中断请求进入，也允许同级的中断请求进入。
- 用于主从结构的8259系统中，将主片设置为“特殊全嵌套方式”。
- 通过ICW_4的“SFNM”位可以设置此种方式。



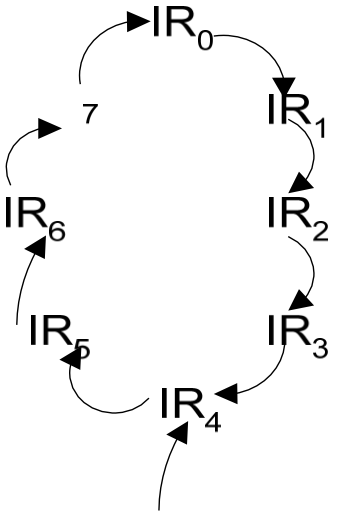
(3) 优先级自动循环方式(AUTOMATIC ROTATION)

- 优先级是循环变化的(不希望有固定的优先级差别)—
  —个设备的中断服务完成后，其优先级自动降为最 低，而将最高优先级赋给原来比它低一级的中断请求。
  - 开始时，优先级队列还是：IR_0->IR_7 ，若此时出现了IR_0请求，响应IR_0并处理完成后，队列变为：

IR_1，IR_2，IR_3，IR_4，IR_5，IR_6，IR_7，IR_0。

若又出现了IR_4请求，处理完IR_4后，队列变为：

- 系统中是否采用"自动循环优先级"，由操作命令字OCW_2来设定。



(4)优先级特殊循环方式(SPECIFIC ROTATION)

- 与“优先级自动循环方式”相比，只有一点不同，即可以设置开始的最低优先级。
- 例如，设定IR_4为最低优先级，那么IR_5就是最高优先级，其余各级按循环方法类推。

2. 屏蔽中断源的方式(中断屏蔽方式)

– 普通屏蔽方式

- 通过对中断屏蔽寄存器(IMR)的设定，实现对相应位为“1”的中断请求的屏蔽。
- 可通过OCW_1使IMR的一位或几位置“1”。

– *特殊屏蔽方式(SPECIAL MASK MODE)

- 实现：

输出OCW_3(ESMM=1,SMM=1)
输出OCW_1(使IMR对应于本级的位为“1”)　} 设置

……
……　} “中断级无效”

输出OCW_1(使IMR对应于本级的位为“0”)
输出OCW_3(ESMM=1，SMM=0)　} 撤销

3. 中断结束方式(END OF INTERRUPT－EOI)

(1) 中断自动结束方式(AUTOMATIC－AEOI方式)

在第二个 $\overline{\text{INTA}}$ 后沿，即完成把对应的ISR位复位。

- 在中断处理过程中，8259A中就没有“正在处 理”的标识。此时，若有中断请求出现，且IF＝1，则 无论其优先级如何(比本级高、低或相同)，都将得到响应。
- 尤其是当某一中断请求信号被CPU响应后，如不及时撤销，就会再次被响应—“二次中断”。
- **AEOI方式**适合于中断请求信号的持续时间有一定限制以及不出现中断嵌套的场合。
- 通过**ICW_4**可以设置AEOI方式**(AEOI=1)**。

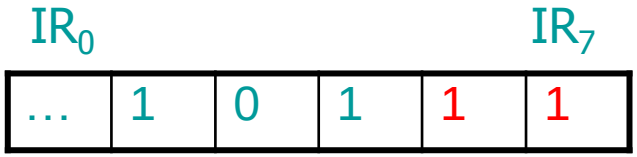
(2)一般(常规)中断结束方式

- 适合于全嵌套方式。
- 实现：在中断服务程序结束时，向8259A发常规中断结束命令(OCW_2:EOI=1,SL=0,R=0)

例：MOV AL,20H; **OCW_2=20H**

OUT 20H,AL; 端口地址**=20H(偶地址)**

- 在全嵌套方式下，ISR中最高优先级的置“1”位，正对应于当前正在处理的中断，将其清“0”，就完成了当前正在处理中断的结束操作。



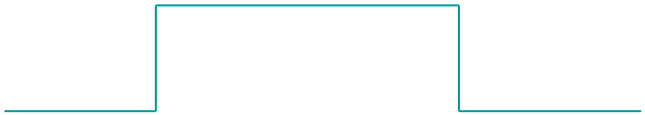
(3)特殊中断结束方式**(SPECIFIC EOI－SEOI)**

- 在非全嵌套方式下，无固定的优先级序列(使用设置优 先权命令或特殊屏蔽方式)，此时，根据**ISR**的内容就 无法确定刚刚所响应(处理)的中断。
- 这种情况下，就不能用上述的EOI方式进行中断结束处 理，而必须用特殊的中断结束命令SEOI－－用 OCW_2:EOI=1,SL=1,R=0,L_2～L_0.
- 由 L_2～L_0指定清除ISR中的哪一位。

4. 中断触发方式

- 电平触发方式： 由IR_i上的有效电平来触发"中断请求触 发器"。

- 边沿触发方式： 由IR_i上由低电平向高电平的跳变来触 发"中断请求触发器"。



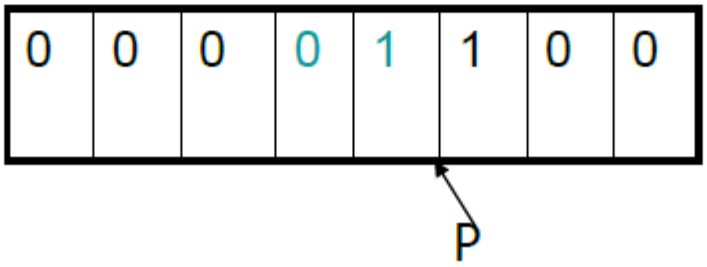
  - 由ICW_1的LTIM位可以设置中断触发方式。

5. 连接系统总线的方式

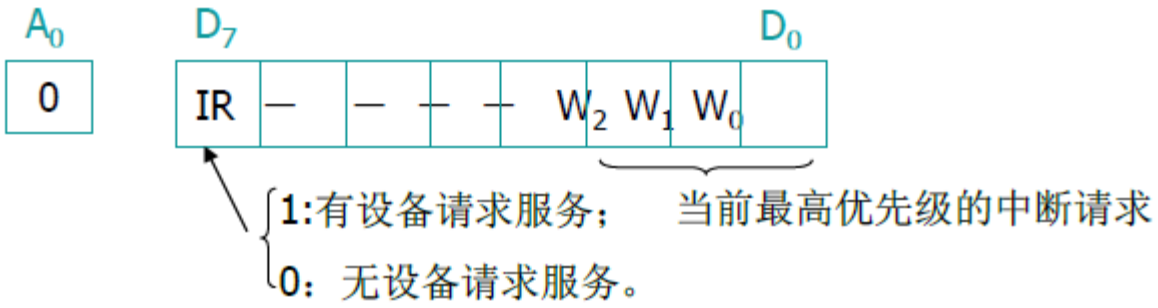
- 缓冲方式(ICW4的BUF=1)
  - 8259A通过总线驱动器(如8286)和数据总线相连。
  - 在缓冲方式下，8259A的 $\overline{SP}/\overline{EN}$ 作为输出($\overline{EN}$ 有效)，此时，由ICW4的M/S位来定义(标识)本8259A是主片还是从片。
- 非缓冲方式(ICW4的BUF=0)
  - 即8259A直接与数据总线相连
  - 在“非缓冲方式下”，8259A的 $\overline{SP}/\overline{EN}$ 作为输入($\overline{SP}$ 有效)
  - 此时，由 $\overline{SP}/\overline{EN}$ 端来标识本8259A是主片还是从片。
  - 在“非缓冲方式下”，ICW4的BUF＝0，M/S位无意义。

中断查询方式

- 特点：既有中断的特点，又有查询(Polling)的特点。
  - 外设仍然向8259A发中断请求信号，要求CPU服务。
  - CPU的IF=0，不响应外部的中断请求(对CPU的中断请求信号不起作用)
  - 此时，CPU需要用**软件查询方法**来确认中断源，从而实现对设备的服务
    - 先向8259A发查询命令(poll command)
      - OCW_3:



- 紧接着执行一条读指令(IN指令)，读出专门的**“中断状态字”**：

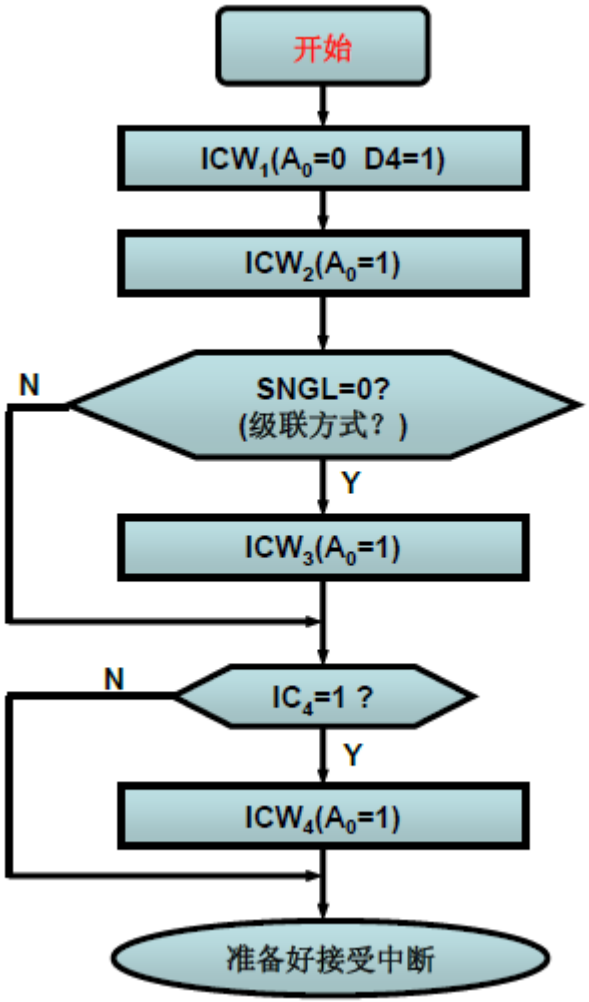


8259A的控制字及编程使用

- 8259A的控制寄存器可分为两组，一组是初始化命令字**ICW 1～ICW4**，另一组是操作命令字**OCW 1～OCW3**。

- 每片8259A具有两个内部端口地址，一个偶地址端口(A0 = 0)，一个奇地址端口(A0 = 1)，其他高位地址码由用户定义，用来产生8259A的片选信号。

8259A的初始化流程



2. 操作命令字

- 在8259A工作期间，可通过设置操作命令字来修改或控制8259A的工作方式。
- 需要说明的是，与初始化命令字ICW 1 ～ICW4需要按规定的顺序进行设置不同，操作命令字 **OCW1～OCW3** 的设置没有规定其先后顺序，使用时可根据需要灵活选择不同的操作命令字写入到8259A中。

8259A初始化编程举例

对8259A按下述要求进行初始化编程：

(1) 工作于80x86系统，单片8259A，边沿触发，缓冲方式

(2) 中断类型码初始值为08H，即IR0～IR７ 对应的中断类型码为08H～0FH。

(3) 中断结束时，用普通中断结束命令，固定优先级。

- 8259A的端口地址为20H，21H。

8259A的读出操作: 可以读出四个方面的内容

- 读“中断状态字”(“查询字”):
  - 先写入**P=1**的**OCW3**查询命令字
  - 用偶地址读**(IN AL,20H)**

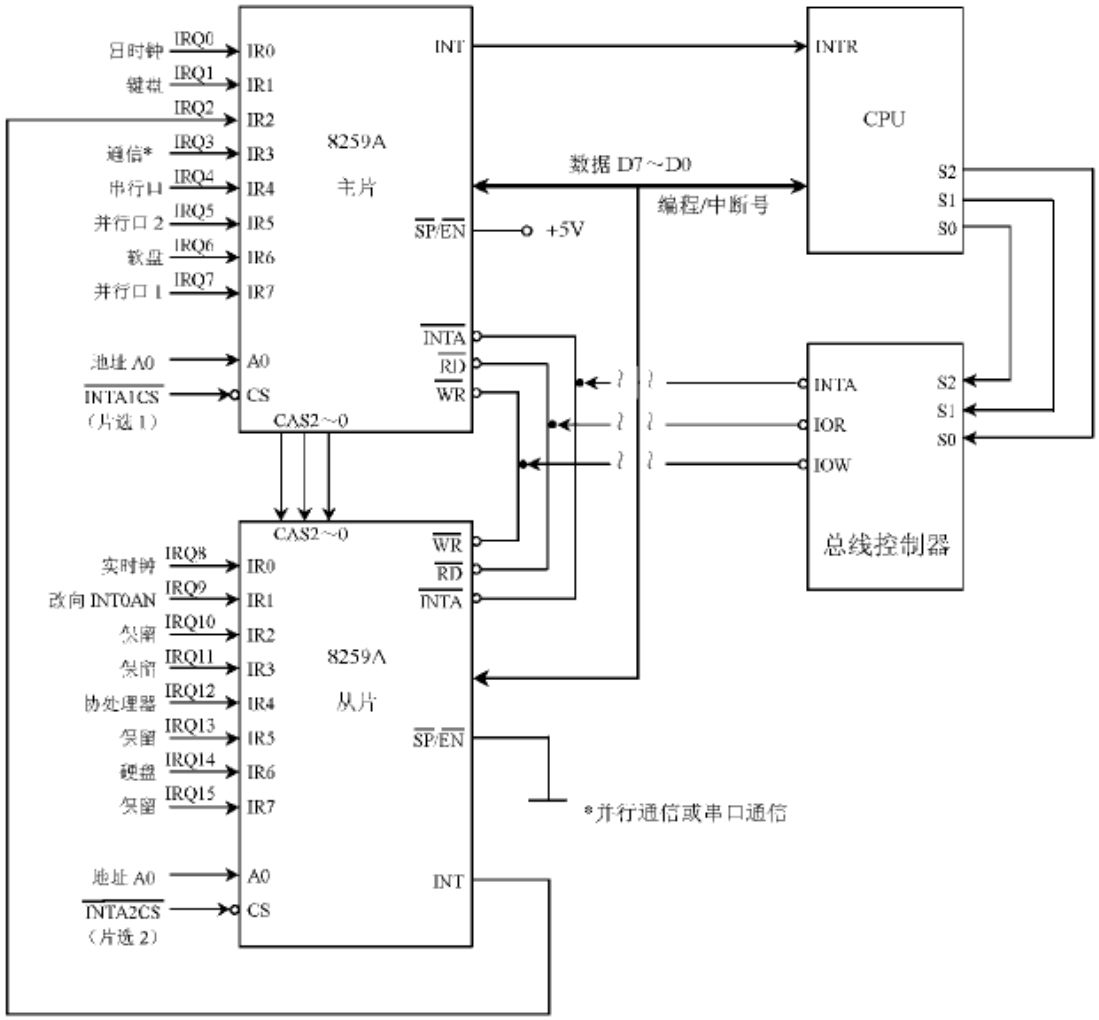
- 读**IRR**:
  - 先写入**OCW3(RR=1,RIS=0)**
  - 用偶地址读**(IN AL,20H)**

- 读**ISR:**
  - 先写入**OCW3(RR=1,RIS=1)**
  - 用偶地址读**(IN AL,20H)**

- 随时可用奇地址读**IMR**
- **(IN AL,21H)**

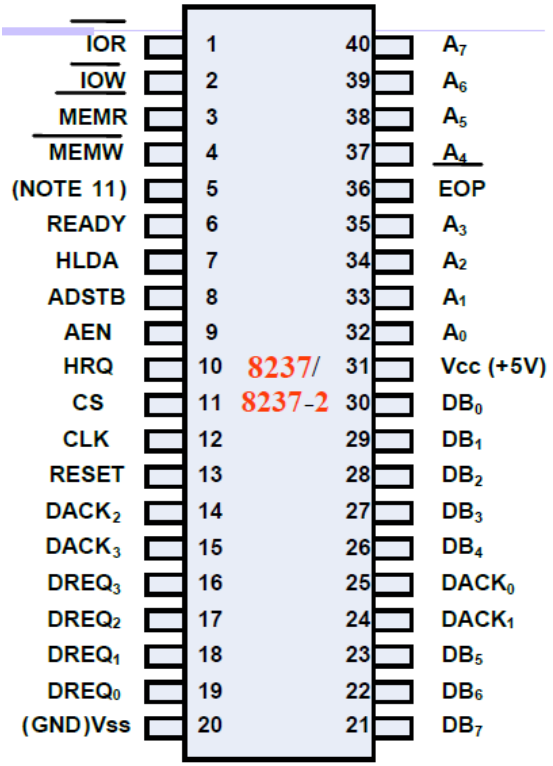
8259应用举例

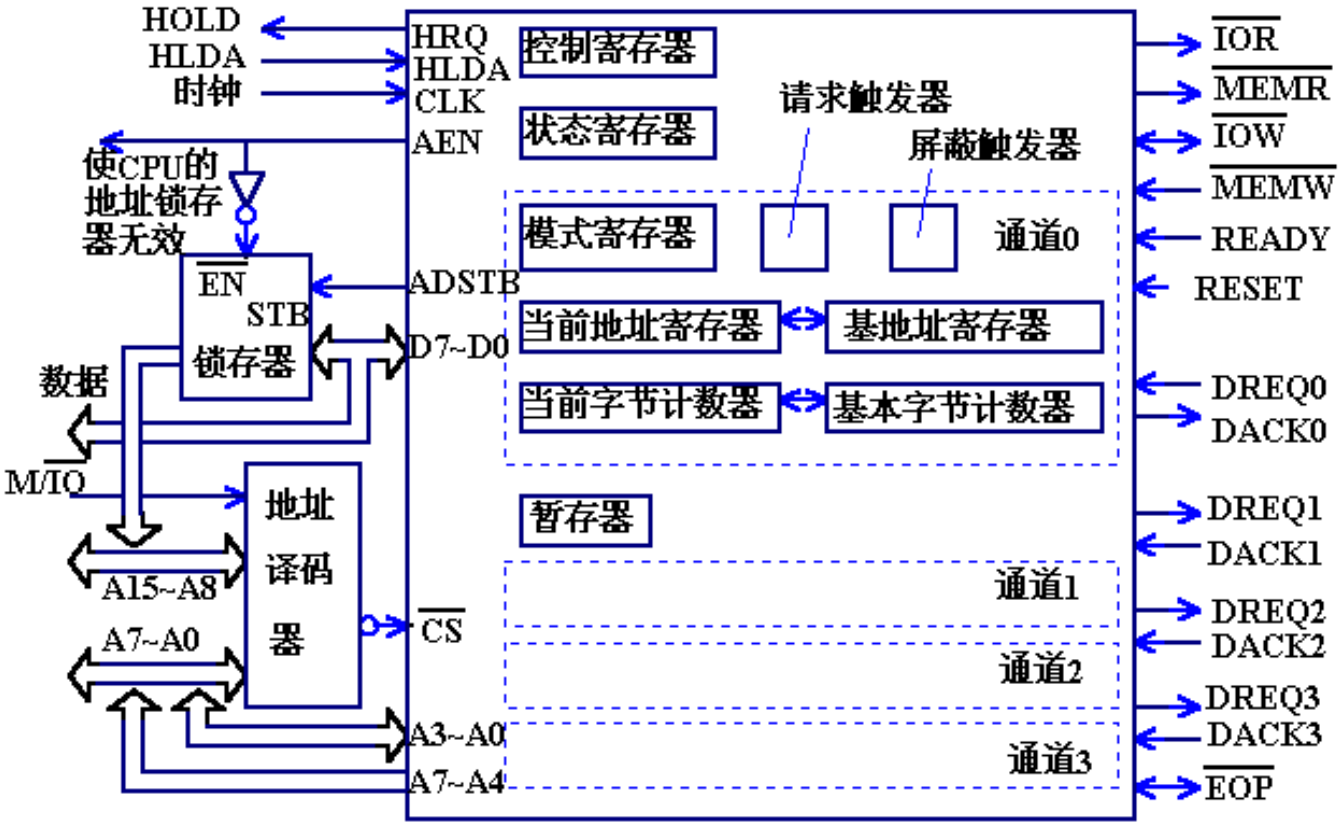
多片8259组成的主从式中断系统



DMAC 8237

Intel 8237

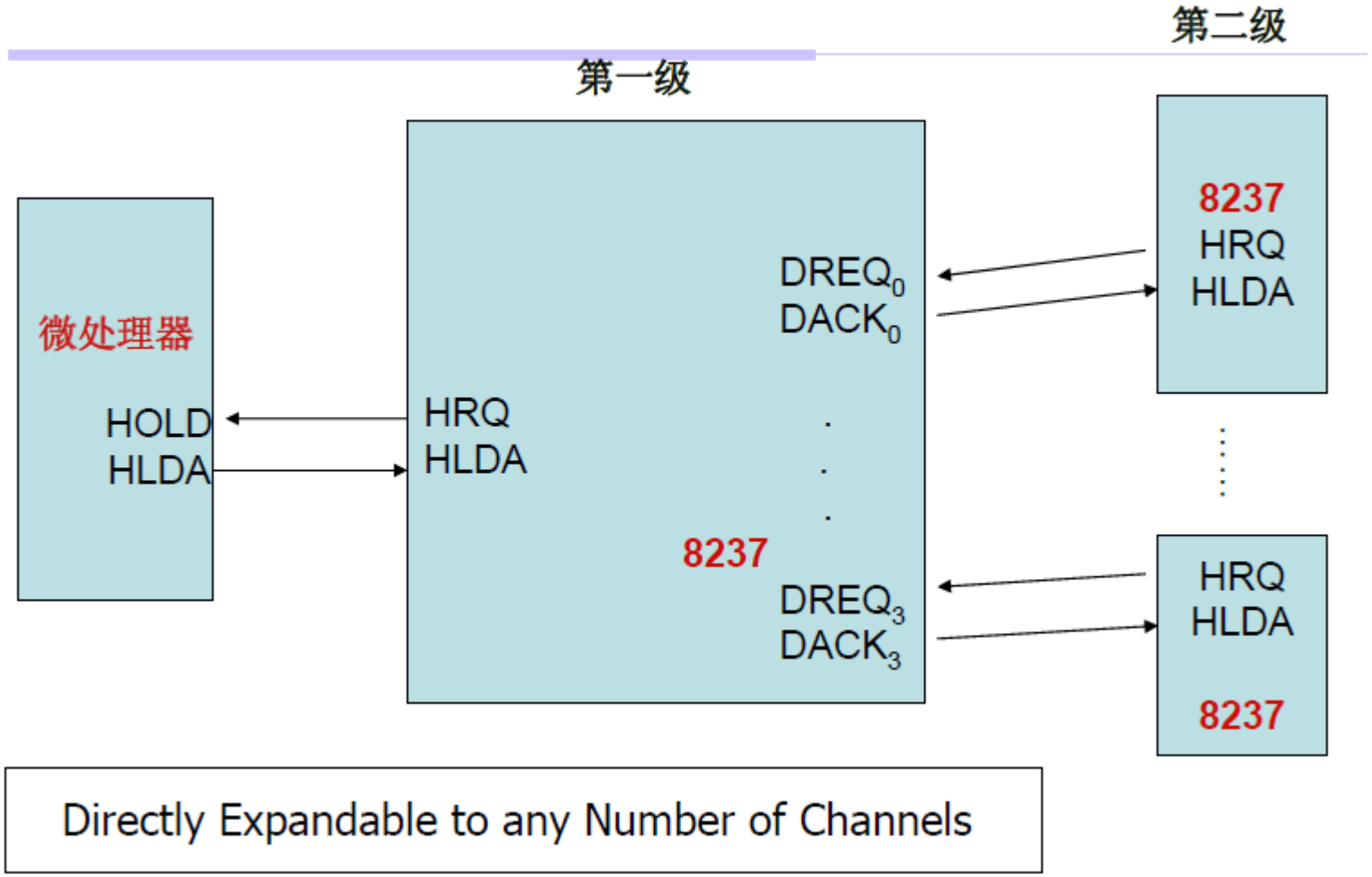

IOR 1
IOW 2
MEMR 3
MEMW 4
(NOTE 11) 5
READY 6
HLDA 7
ADSTB 8
AEN 9
HRQ 10
CS 11
CLK 12
RESET 13
DACK2 14
DACK3 15
DREQ3 16
DREQ2 17
DREQ1 18
DREQ0 19
(GND)Vss 20
8237/
8237-2
40 A7
39 A6
38 A5
37 A4
36 EOP
35 A3
34 A2
33 A1
32 A0
31 Vcc (+5V)
30 DB0
29 DB1
28 DB2
27 DB3
26 DB4
25 DACK0
24 DACK1
23 DB5
22 DB6
21 DB7


HOLD
HLDA
时钟
使CPU的地址锁存器无效
EN
STB
锁存器
数据
M/IO
A15~A8
A7~A0
地址译码器
HRQ
HLDA
CLK
AEN
ADSTB
D7~D0
CS
A3~A0
A7~A4
控制寄存器
状态寄存器
请求触发器
屏蔽触发器
模式寄存器
通道0
当前地址寄存器
基地址寄存器
当前字节计数器
基本字节计数器
暂存器
通道1
通道2
通道3
IOR
MEMR
IOW
MEMW
READY
RESET
DREQ0
DACK0
DREQ1
DACK1
DREQ2
DACK2
DREQ3
DACK3
EOP

8237工作模式

- 8237具有四种工作方式：
  - 单字节传送方式(single transfer mode)
  - 块传送方式(block transfer mode)
  - 请求传送方式(demand transfer mode)
  - 级联方式(cascade mode)

级联方式



Directly Expandable to any Number of Channels

8237内部寄存器

寄存器名称	位数	数量	CPU访问方式
基地址寄存器	16位	4	只写
基字节计数寄存器	16位	4	只写
当前地址寄存器	16位	4	可读可写
当前字节计数寄存器	16位	4	可读可写
临时地址寄存器	16位	4	不可访问
临时字节计数寄存器	16位	4	不可访问
命令寄存器	8位	1	只写
状态寄存器	8位	1	只读
暂存寄存器	8位	1	只读
模式寄存器	6位	4	只写
屏蔽寄存器	4位	1	只写
请求寄存器	4位	1	只写

8237的读写控制

A_3	A_2	A_1	A_0	$\overline{IOR}$	$\overline{IOW}$	命　　令
1	0	0	0	0	1	读状态寄存器
1	0	0	0	1	0	写控制寄存器
1	0	0	1	1	0	写 DMA 请求标志寄存器
1	0	1	0	1	0	写 DMA 屏蔽标志寄存器
1	0	1	1	1	0	写模式寄存器
1	1	0	0	1	0	清除先/后触发器
1	1	0	1	0	1	读暂存器
1	1	0	1	1	0	发复位命令
1	1	1	0	1	0	清除屏蔽标志
1	1	1	1	1	0	综合屏蔽命令

DMA 通道	基本地址寄存器和当前地址寄存器	基本字节计数器和当前字节计数器
通道 0	起始地址 +0	起始地址 +1
通道 1	起始地址 +2	起始地址 +3
通道 2	起始地址 +4	起始地址 +5
通道 3	起始地址 +6	起始地址 +7

页面寄存器

- 由于8237只能输出16位地址，所以在其控制下进行的DMA传送的最大寻址空间为216。
- 对于更大的DMA传送地址空间，则必须设法提供除此16位地址以外的高位地址。
- 系统中专门为每个DMA通道增设了一个4位I/O端口，在数据块传送之前可单独对其编程，用以提供高4位地址。
- 这个4位I/O端口也称DMA页面寄存器。

8237的编程步骤

(1) 输出主清除命令；
(2) 置页面寄存器；
(3) 写入基和当前地址寄存器；
(4) 写入基和当前字节计数寄存器；
(5) 写入模式寄存器；
(6) 写入命令寄存器；
(7) 写入屏蔽寄存器；
(8) 写入请求寄存器。

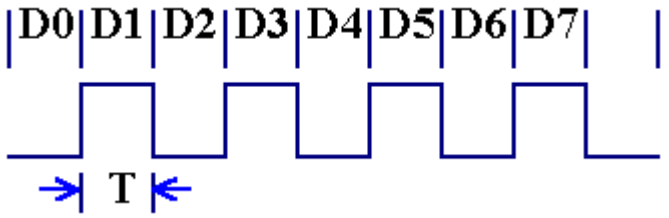
8237的编程

- 例：在IBM　PC系统中，试利用8237通道1，将内存8000H：0H开始的16K字节数据传送至磁盘(地址增量传送)。
  - 要求采用块传送方式，传送完不自动预置，DREQ和DACK均为高电平有效，固定优先级，普通时序，不扩展写信号。
  - 系统中8237的端口地址为00H～0FH。
  - 通道1“页面寄存器”的端口地址为83H。

串行通信

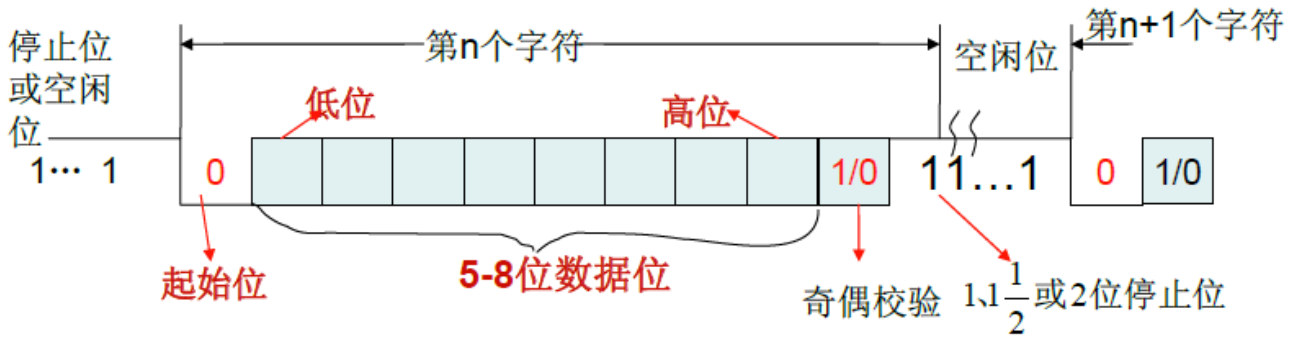
串行通信

- 串行通信特点
  - 将传输的数据分解成二进制位
  - 一条信号线
  - 按位顺序传送
  - 每位占规定的时间间隔
  - 适用于长距离通信

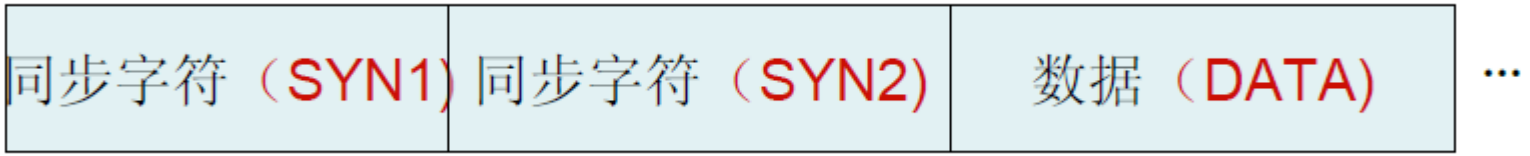


异步与同步

- 异步方式
  - 帧

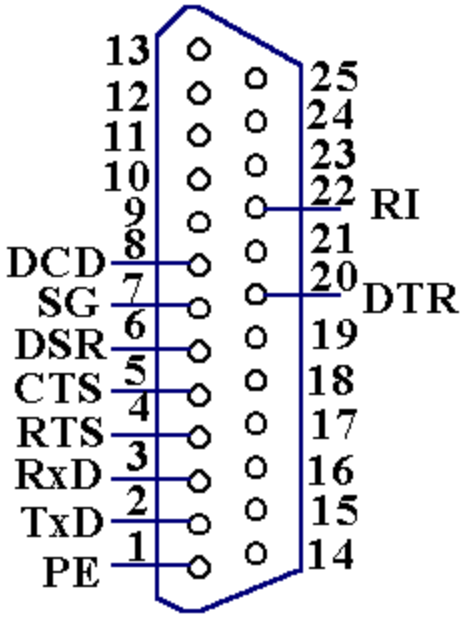


- 同步方式

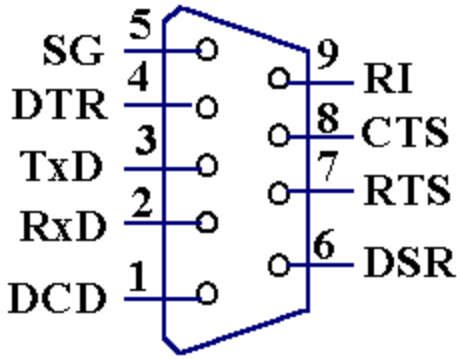


串行接口标准RS-232

- 两个问题
  - 计算机与外设共同遵守的约定
  - 按接口设计计算机与外设间的接口电路



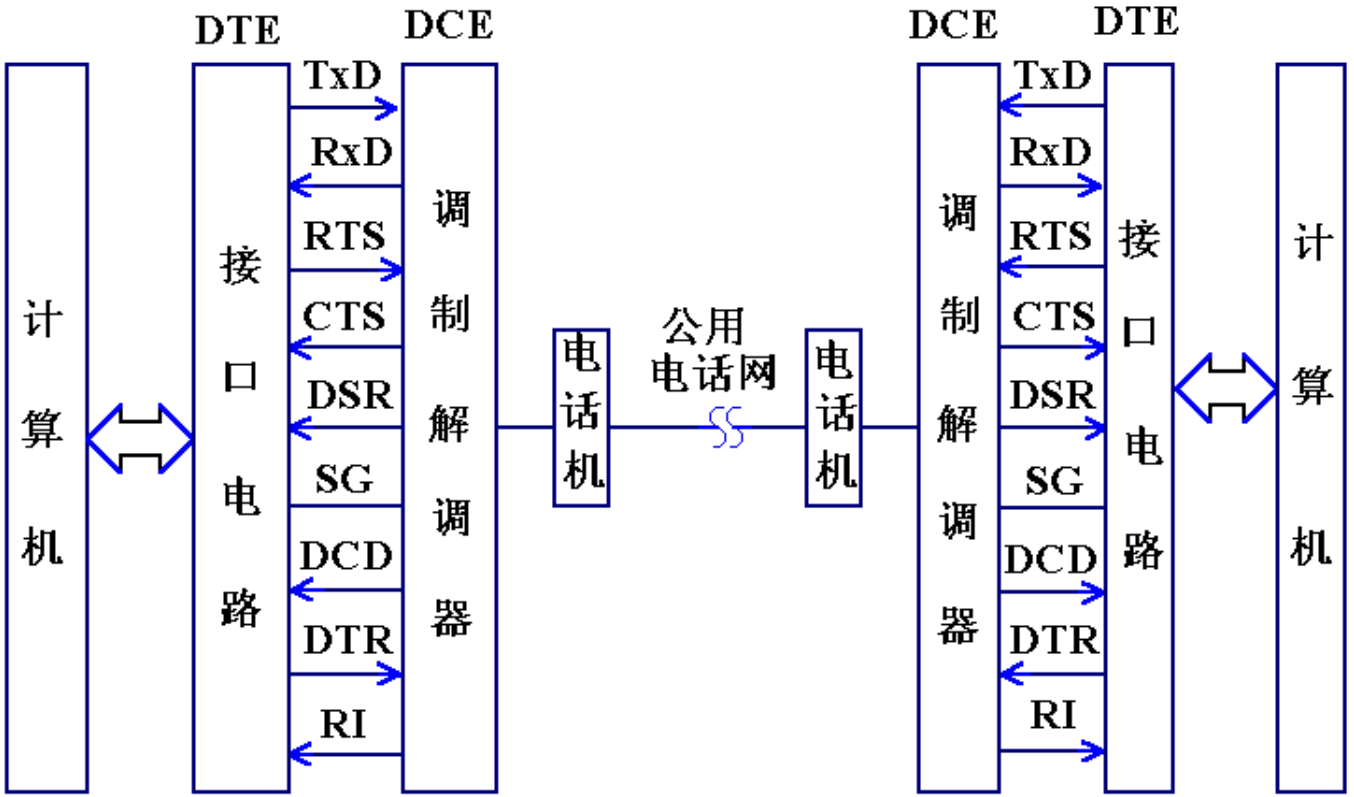
DB-25



DB-9

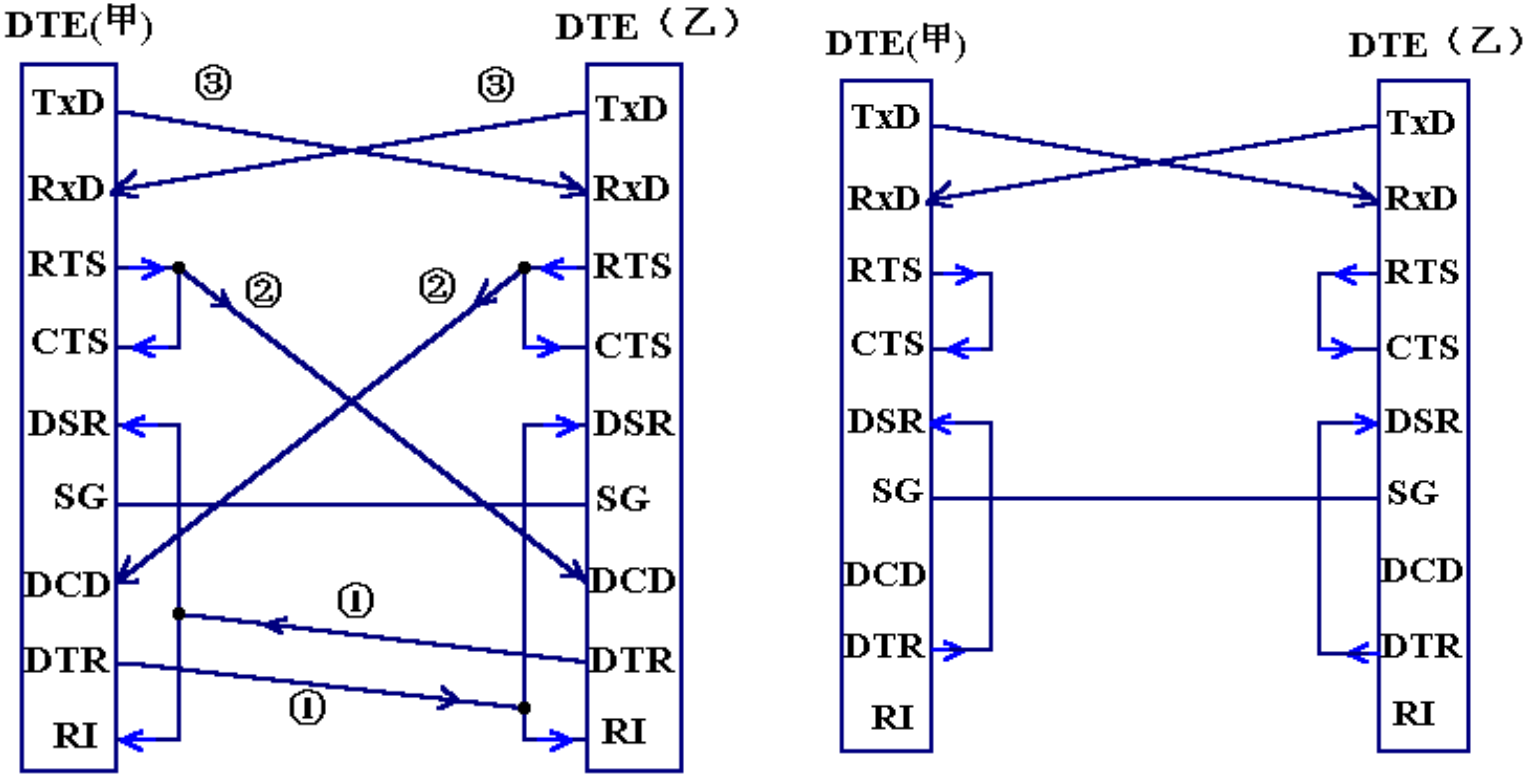
RS-232的连接方式

- 使用Modem

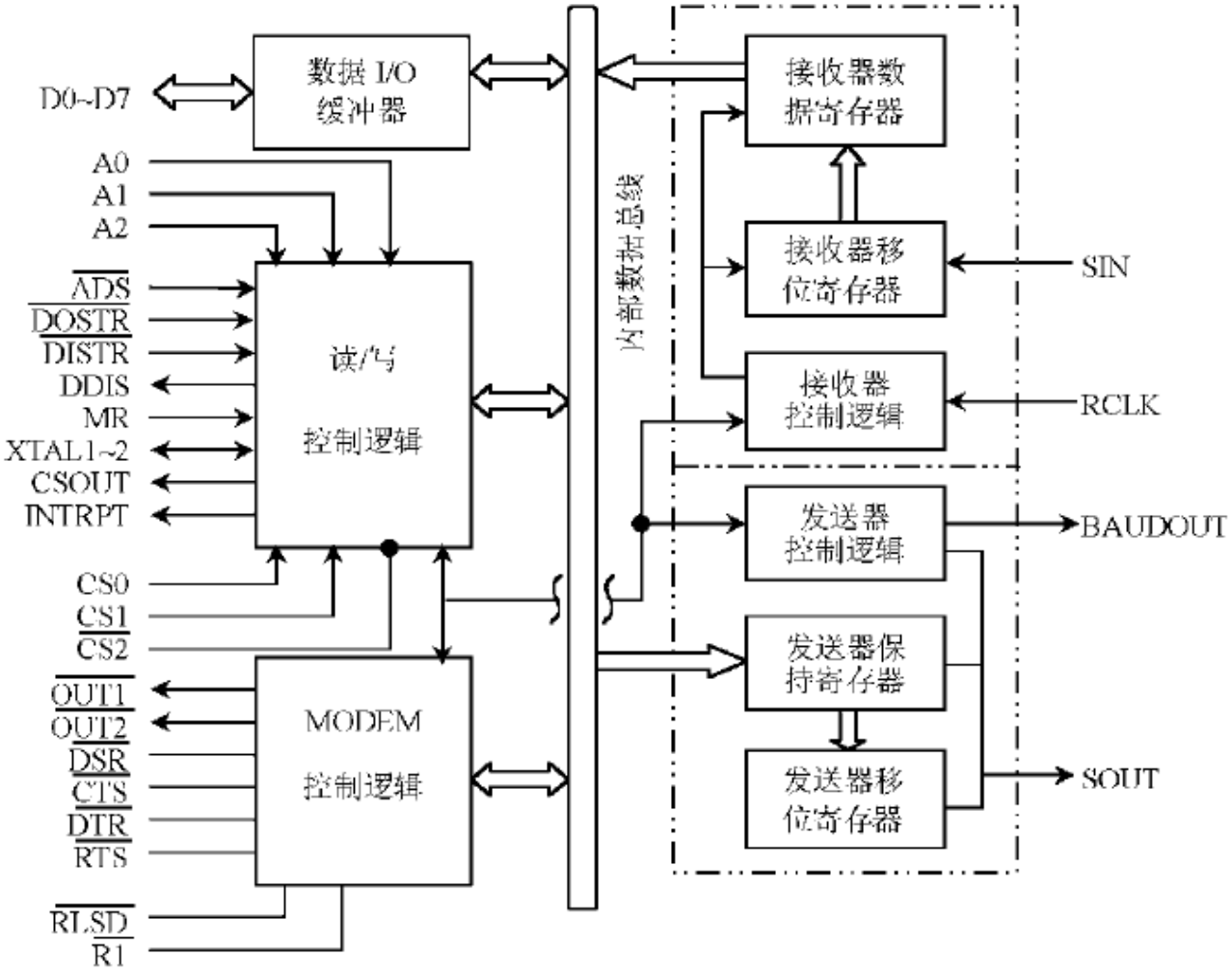


RS-232的连接方式

- 无Modem



8250功能与结构


D0~D7
数据 I/O
缓冲器
A0
A1
A2
ADS
DOSTR
DISTR
DDIS
MR
XTAL1~2
CSOUT
INTRPT
读/写
控制逻辑
CS0
CS1
CS2
OUT1
OUT2
DSR
CTS
DTR
RTS
MODEM
控制逻辑
RLSD
RI
内部数据总线
接收器数
据寄存器
接收器移
位寄存器
SIN
接收器
控制逻辑
RCLK
发送器
控制逻辑
BAUDOUT
发送器保
持寄存器
发送器移
位寄存器
SOUT

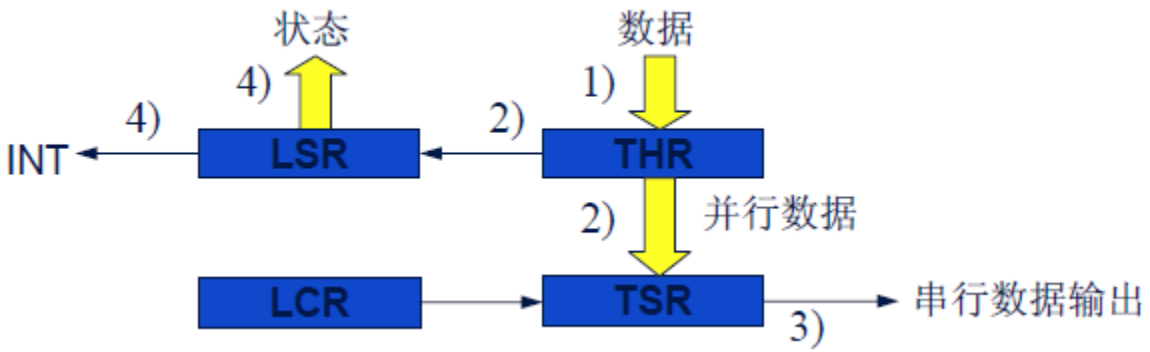
8250内部寄存器寻址

Register Addresses

DLAB	A_2	A_1	A_0	Register
0	0	0	0	Receiver Buffer (read), Transmitter Holding Register (write)
0	0	0	1	Interrupt Enable
X	0	1	0	Interrupt Identification (read only)
X	0	1	1	Line Control
X	1	0	0	MODEM Control
X	1	0	1	Line Status
X	1	1	0	MODEM Status
X	1	1	1	Scratch
1	0	0	0	Divisor Latch (least significant byte)
1	0	0	1	Divisor Latch (most significant byte)

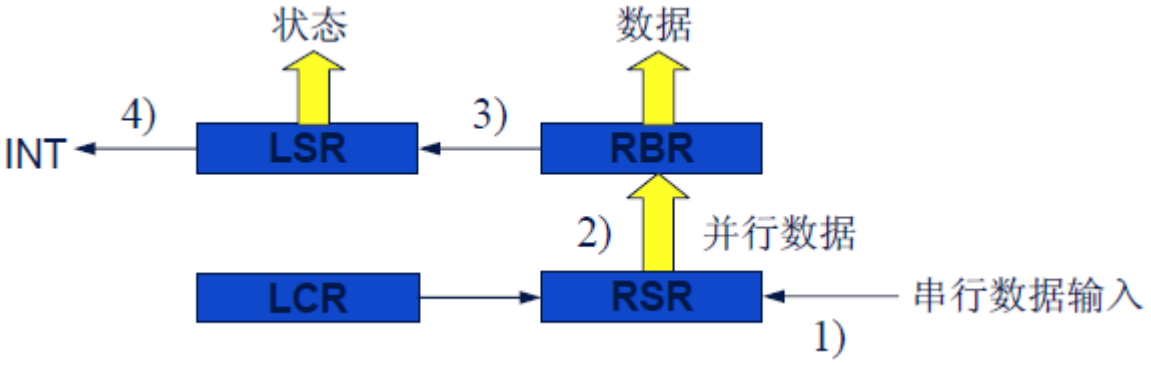
8250发送数据过程

- CPU(数据)→8250的THR ；
- TSR移空时，THR → TSR，LSR中“数据发送保持寄存器空” 状态位置位；
- TSR根据LCR中规定的格式从低到高逐位发送数据；
- LSR中“数据发送保持寄存器空” 状态位可用来产生中断，也可查询该状态位，以实现数据的连续发送。

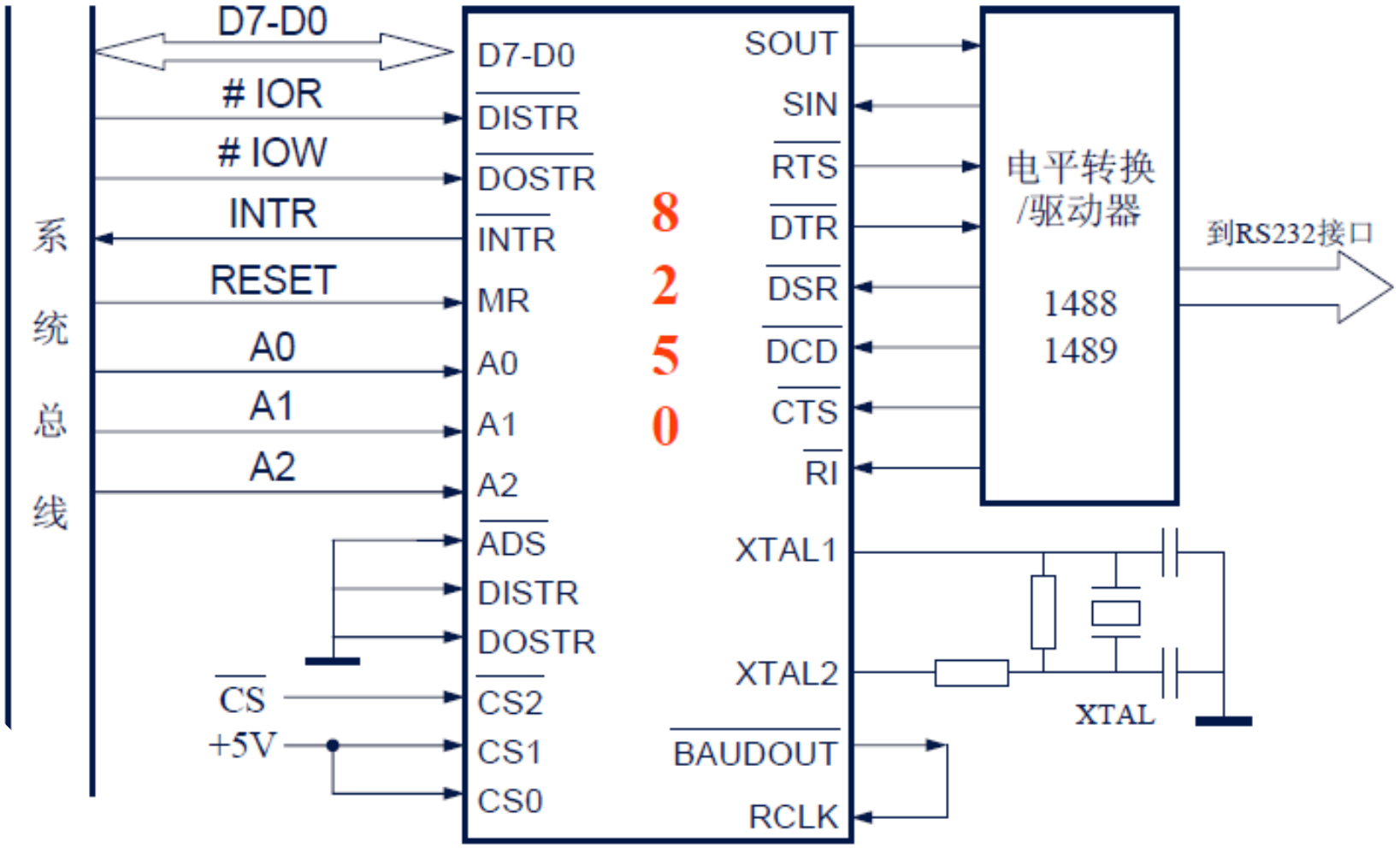


8250接收数据过程

- SIN引脚上的串行数据逐位进入RSR；
- RSR根据LSR中规定的数据位数确定是否收到了一个完整的数据，收到后将数据→RBR；
- RBR收到RSR的数据后，将LSR寄存器中“接收缓冲寄存器满”的状态位置位；
- LSR中“接收缓冲寄存器满”状态位可用来产生中断，也可查询该状态位，以实现数据的连续接收。



8250与8088的连接



初始化过程

- 确认串口地址
  - COM1、COM2
- 设置波特率
  - 依据波特率，查表得出分频次数，分别放入DLL和DLH
  - 先将LCR的D7置1，然后写入
- 设置通信格式
  - 初始化LCR
- 设置Modem控制寄存器
- 设置中断允许寄存器IER

例一

- 要求以9600波特率进行异步通讯，每字符8位，2个停止位，无校验，允许发送和接收中断。异步通讯的端口地址为2F8～2FFH

- 1.8432M=1843.2KHz/16/9600=12=000CH
- 要求波特率误差<5%

查询方式通信

- 查询方式通信编程
  - 读线路状态寄存器3FDH查相应状态位（D0和D5位）

发送程序：

```
TR: MOV DX, 3FDH
    IN AL, DX
    TEST AL, 20H     ;D5位是否为1
    JZ TR
    MOV AL, [SI]     ;从[SI]取出
    MOV DX, 3F8H     ;发送数据
    OUT DX, AL
```

接收程序：

```
RE: MOV DX, 3FDH
    IN AL, DX
    TEST AL, 1       ;D0位是否为1
    JZ RE
    MOV DX, 3F8H
    IN AL, DX
    MOV [DI], AL     ;读入数据存入[DI]中
```

中断方式

- 初始化，开放相应中断
- 发送：
  - 开发送缓冲器空中断，使IER的D1=1；
  - 发送第一字符，写入发送缓冲区
  - 发送完毕引起中断，在中断服务程序写入下一字符
- 接收
  - 开接收数据出错和数据就绪中断，IER的D2=1，D0=1
  - 有错，引起中断转出错处理程序
  - 无错，接收完毕引起中断，接收下一字符

初始化：

```
MOV     DX, 3FBH
MOV     AL, 80H         ;访问除数寄存器
OUT     DX, AL
MOV     DX, 3F8h
MOV     AX, 000CH
OUT     DX, AL          ;除数的低8位写入3F8H
INC     DX
MOV     AL, AH
OUT     DX, AL          ;除数的高8位写入3F9H
 MOV    AL, 0BH         ;00001011  8位数据位，1位停止位，奇校验
MOV     DX, 3FBH
OUT     DX, AL
MOV     AL, 3           ;设置中断允许寄存器，允许发送与接收中断请求
MOV     DX, 3F9H
OUT     DX, AL
STI
```

```
MOV    DI, RBUFPT       ;RBUFPT  DW  ?
MOV    SI, TBUFPT       ;TBUFPT  DW  ?
```

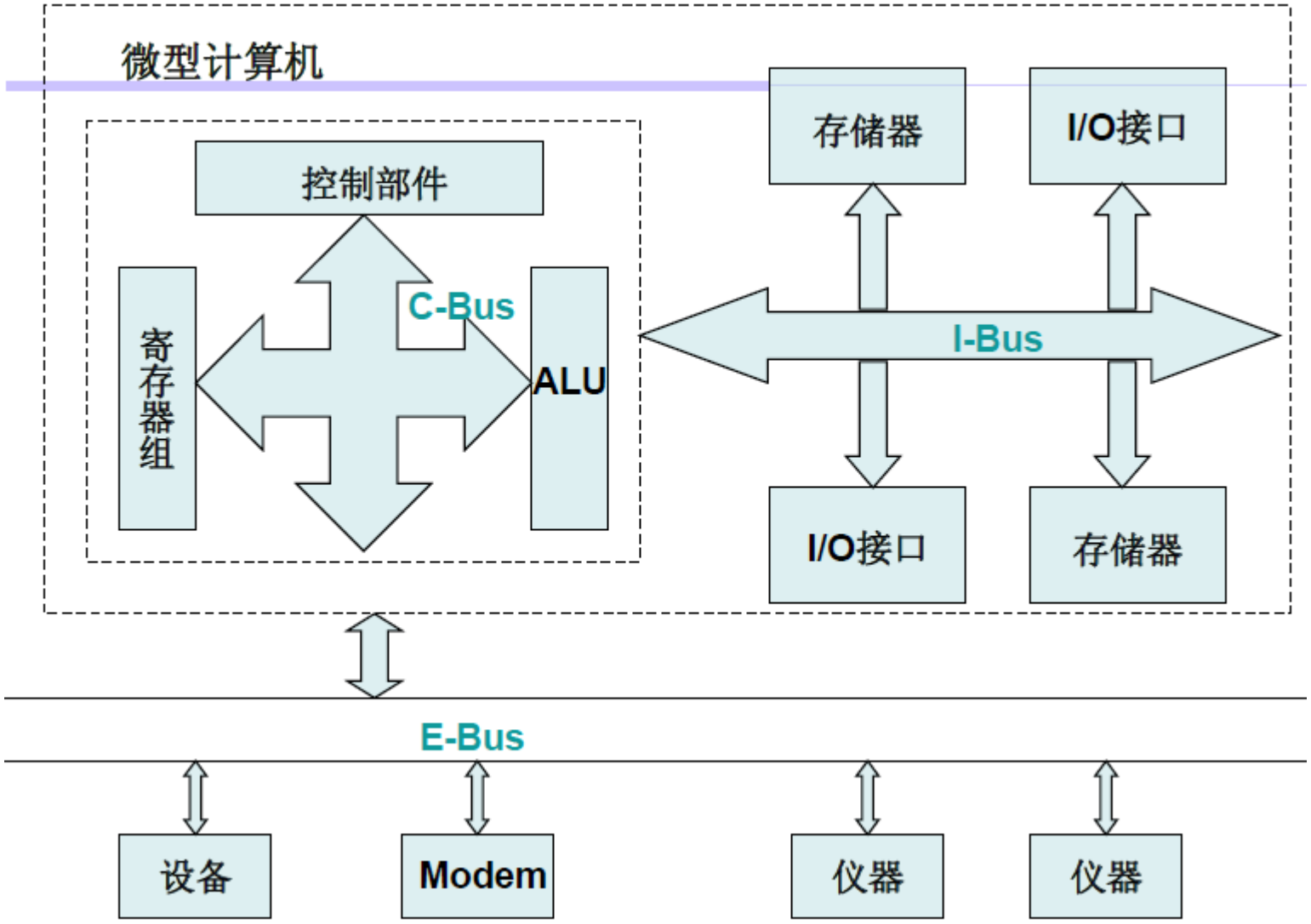
```
INTPRG:
        PUSH    AX
        PUSH    BX
        PUSH    DX
        PUSH    DS
        MOV     DX，3FD
        IN      AL，DX
        TEST    AL，1EH   ；检查线路状态寄
        JNZ     ERROR

        MOV     DX，3FAH
        IN      AL，DX        ；读IIR内容
        AND     AL, 7
        CMP     AL, 2
        JZ      TRAN          ；发送缓冲器空，转TRAN
        CMP     AL, 4
        JNZ     BACK
RECV:   MOV     DX，3F8H      ；否则，接收数据
        IN      AL，DX
        MOV     ［DI］，AL     ；存数据
        INC     DI
        JMP     BACK

TRAN:   MOV     DX，3F8H      ；发送数据
        MOV     AL，[SI］
        OUT     DX，AL
        INC     SI
BACK:   MOV     AL，20H       ；发中断结束命令EOI 给8259
        OUT     20H，AL
        JMP     ENDP
ERROR:  ……

ENDP:   POP     DS
        POP     DX
        POP     BX
        POP     AX
        STI
        IRET
```

总线

总线

- 总线
  - 计算机两个或两个以上的模块(部件或子系统)之间相互连接与通信的公共通路
  - 总线不仅仅是一组传输线，它还包括一套管理信息传输的规则(协议)
  - 在计算机系统中，总线可以看成一个具有独立功能的组成部件
  - 通常包括一组信号线
    - 数据线和地址线，控制时序和中断信号，电源线和地线，备用线

总线层次结构

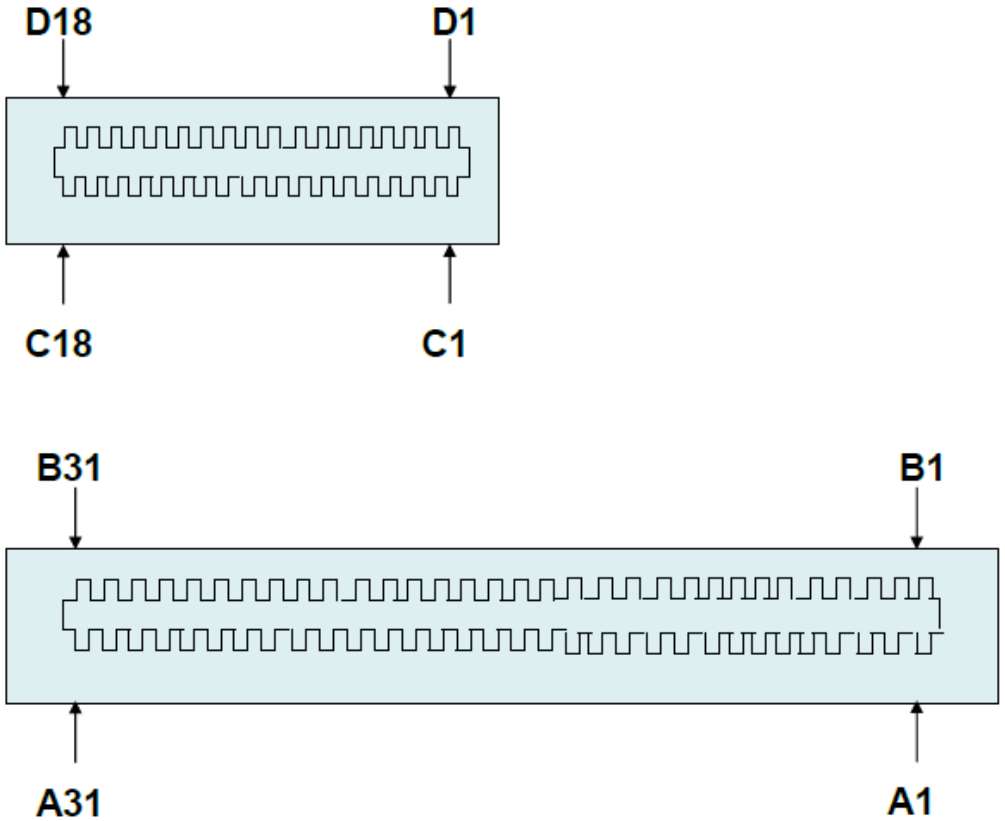

微型计算机
控制部件
寄存器组
C-Bus
ALU
存储器
I/O接口
I-Bus
I/O接口
存储器
E-Bus
设备
Modem
仪器
仪器

总线标准

- 总线标准必须有具体和明确的规范，通常包括：
  - 机械特性：规定模块插件的机械尺寸，总线插头、插座的规格及位置等；
  - 电气特性：规定总线信号的逻辑电平、噪声容限及负载能力等；
  - 功能特性：给出各总线信号的名称及功能定义；
  - 规程特性：对各总线信号的动作过程及时序关系进行说明。

ISA总线

- ISA总线(Industrial Standard Architecture)即AT总线，它是在8位的XT总线基础上扩展而成的16位的总线体系结构。



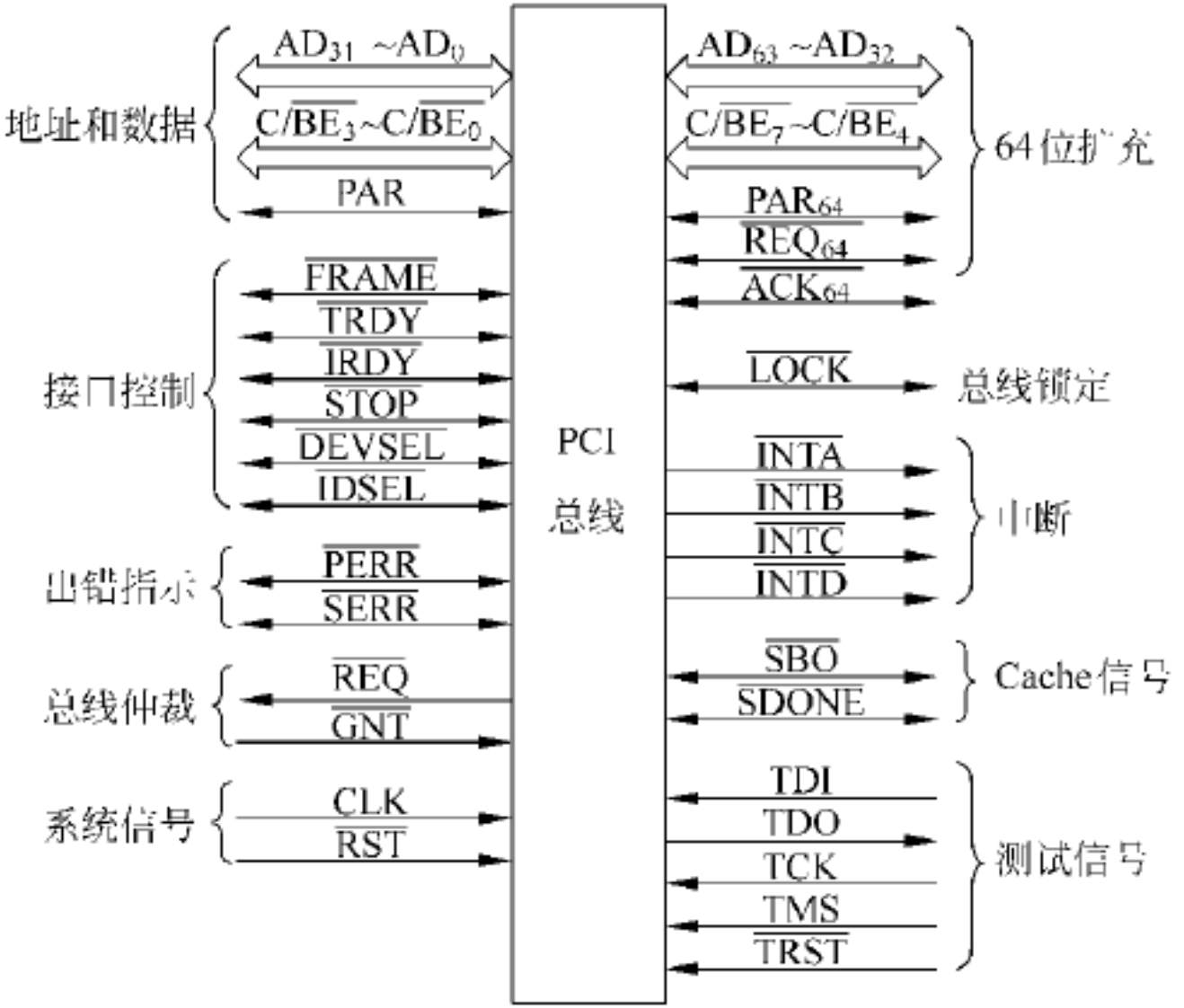
PCI总线

- PCI总线(Peripheral Component Interconnect，外围部件互连总线)
  - 1991年由Intel公司首先提出，并由PCI SIG(Special Interest Group)来发展和推广。
  - PCI SIG是一个包括Intel、IBM、Compaq、Apple和DEC等100多家公司在内的组织集团。
  - 1992年6月推出了PCI 1.0版
  - 1995年6月又推出了支持64位数据通路、66MHz工作频率的PCI 2.1版。
- 由于PCI总线先进的结构特性及其优异的性能，使之成为现代微机系统总线结构中的佼佼者

PCI的特点

- 高传输率
- 高效率
- 即插即用
- 独立于CPU
- 负载能力强、易于扩展

PCI 总线信号


地址和数据
AD31 ~AD0
C/BE3~C/BE0
PAR
接口控制
FRAME
TRDY
IRDY
STOP
DEVSEL
IDSEL
出错指示
PERR
SERR
总线仲裁
REQ
GNT
系统信号
CLK
RST
PCI
总线
AD63 ~AD32
C/BE7~C/BE4
PAR64
REQ64
ACK64
64位扩充
LOCK
总线锁定
INTA
INTB
INTC
INTD
中断
SBO
SDONE
Cache信号
TDI
TDO
TCK
TMS
TRST
测试信号

PCI操作特点

- 信号的变化在时钟下降沿，信号采样在时钟上升沿；
- 传输过程由主设备使FRAME#为有效开始，而FRAME#无效后进行最后一个数据传输；
- 每个数据传输过程由一个地址期和若干个数据期组成，地址期为一个时钟周期，数据期可插入等待，等待是由于主或从设备未准备好造成；

PCI的编址

三个地址空间

- I/O空间：位于主机系统
  - ✓正向译码：
  - ✓负向译码：其他设备未被选中时，由总线扩展桥发DEVSEL#信号来响应。
- 内存空间：位于主机系统
- 配置空间：分布于各个设备中
  - ✓为系统软件提供此设备的参数，以便安装驱动软件；
  - ✓提供设备的各种寄存器地址；
  - ✓由软件将其映射到统一的空间，用来支持即插即用功能。

AD/DA

信号转换

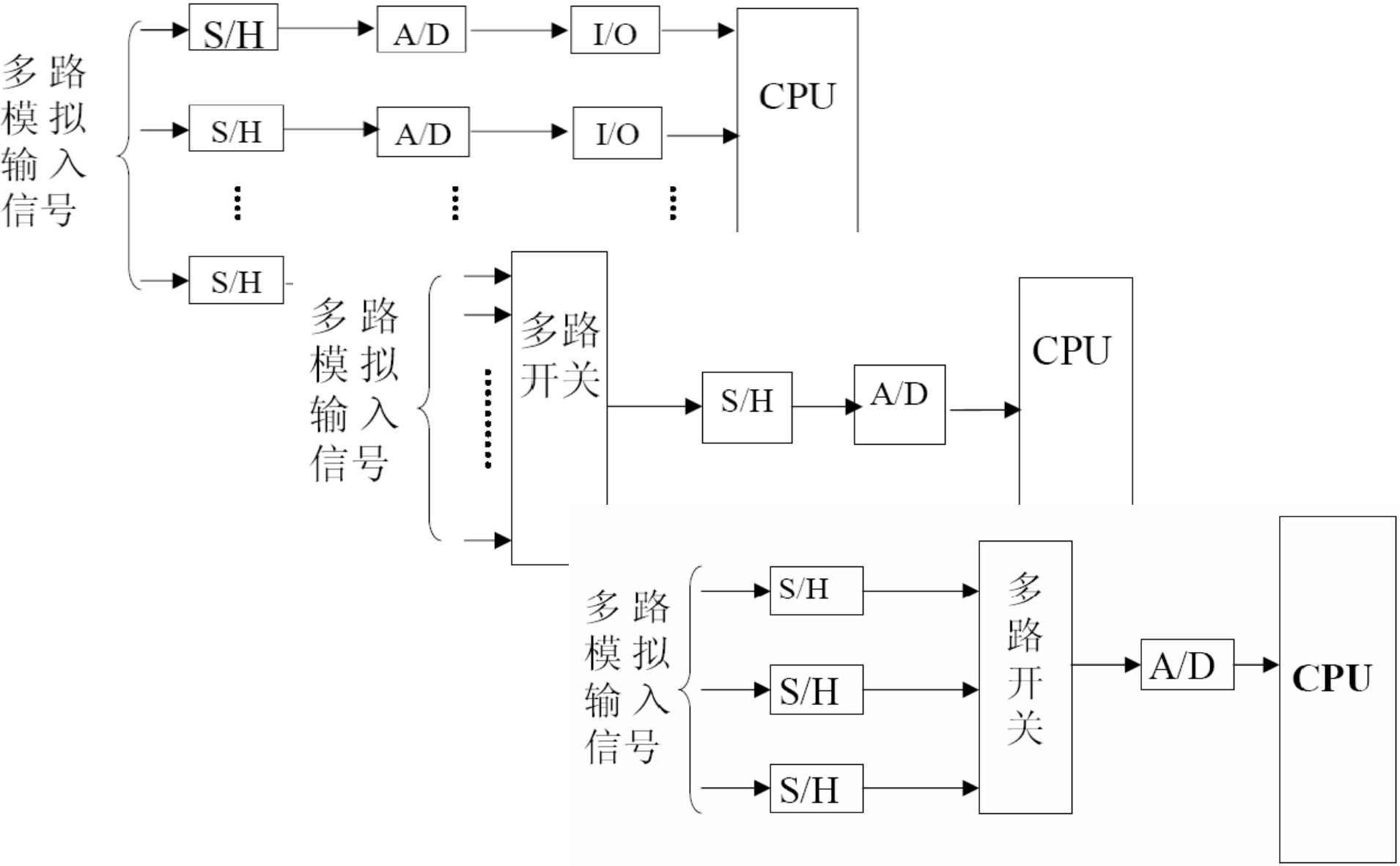
- AD/DA
  - AD：模拟量→数字量
  - DA：数字量→模拟量

- 数据采集
  - 从传感器或其它方式得到的模拟信号，经过必要的处理后转换成数字信号，以供存储、传输、处理和显示。
  - 模拟域到数字域间的接口，称为数据采集系统。

过程通道

- 模拟量输入通道
  - 传感器
  - 信号处理环节
  - 多路模拟开关
  - 采样保持器
  - 模数转换器（A/D）
- 模拟量输出通道
  - 数模转换器（D/A）
  - 多路模拟开关
  - 信号处理环节

数据采集系统的一般形态



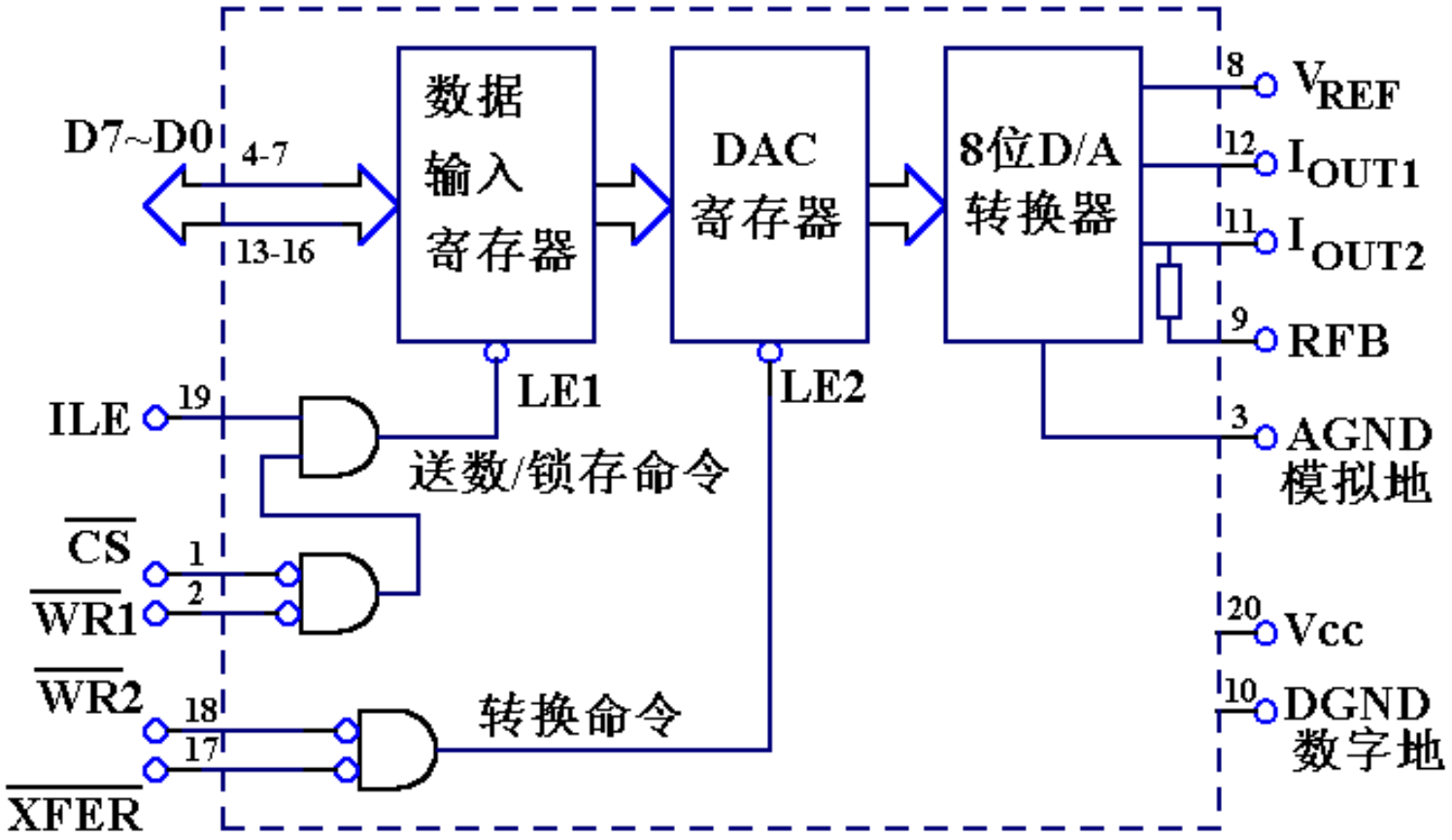
D/A接口电路设计

- D/A转换器原理
  - “按权展开、然后求和”
  - 电阻解码网络
  - 两种类型
    - 权电阻网络
    - T形电阻网络（R-2R梯形电阻网络）

D/A转换器主要技术指标

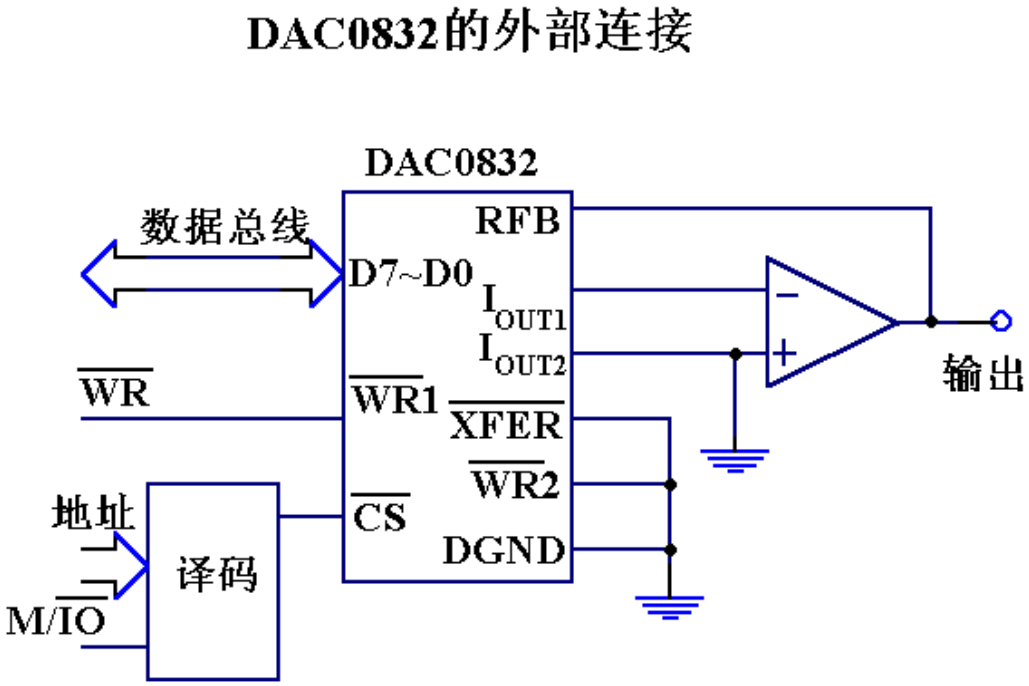
- 数模转换器的主要技术指标:
  - 分辨率
    - D/A转换器能够转换的二进制的位数
  - 转换时间
    - 数字量输入、转换、输出并稳定的时间，一般几百ns到几μs之间。
  - 转换精度
    - D/A转换器实际输出电压与理论值之间的误差。1/2（LSB）（LSB为最低有效位）
  - 线性度
    - 数字量变化时，D/A转换器输出的模拟量按比例关系变化的程度。

DAC0832


D7~D0
4-7
13-16
数据
输入
寄存器
DAC
寄存器
8位D/A
转换器
8
VREF
12
IOUT1
11
IOUT2
9
RFB
3
AGND
模拟地
20
Vcc
10
DGND
数字地
LE1
LE2
ILE
19
送数/锁存命令
CS
1
WR1
2
WR2
18
17
转换命令
XFER

- 内部有两级输入锁存器
  - 输入寄存器和DAC寄存器
- 有三种工作方式
  - 双缓冲工作方式
  - 单缓冲工作方式
  - 直通工作方式
- 双缓冲工作方式优点
  - 转换输出模拟信号的同时，输入新的数据，提高速度
  - 可实现多个模拟输出通道同步输出

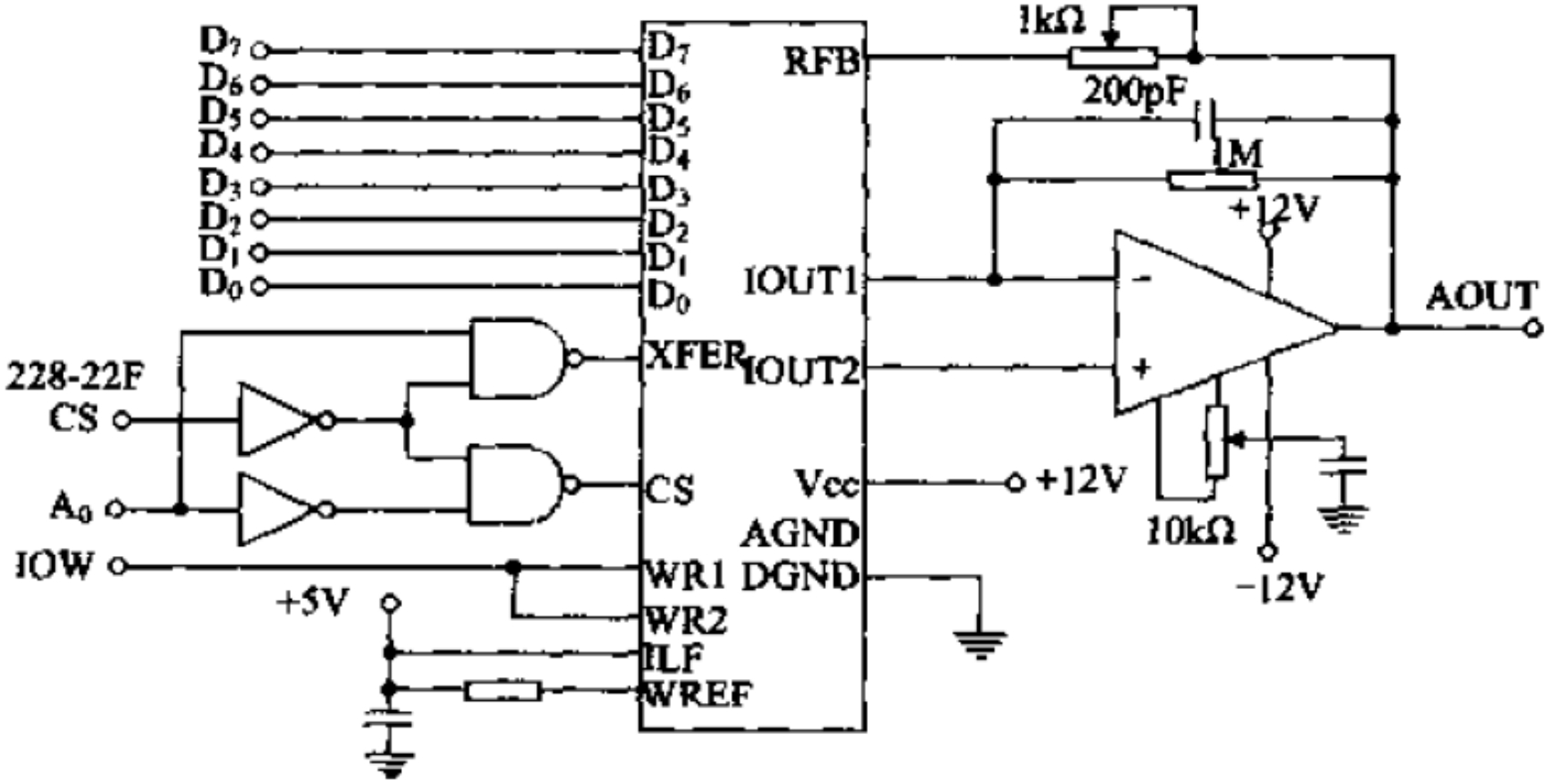
DAC0832单缓冲工作方式



设D/A转换端口号为PORTDA，设需转换的数据放在1000H单元，则D/A转换程序为：

```
MOV BX，1000H
MOV AL，[BX]
MOV DX，  PORTDA
OUT  DX，AL
```

DAC0832双缓冲工作方式



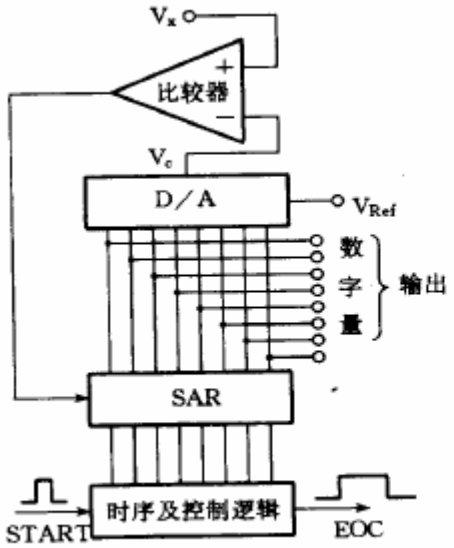
DAC0832双缓冲工作方式

设CS由A9~A1经译码产生，DAC的地址范围是228-22FH，实际只使用228H和229H两个地址。在CPU执行OUT指令时，若A0=0，DAC0832内部LE1有效，数据总线上的值（AL）送入数据输入寄存器; 若A0=1，DAC0832内部LE2有效，数据输入寄存器的值送DAC寄存器。

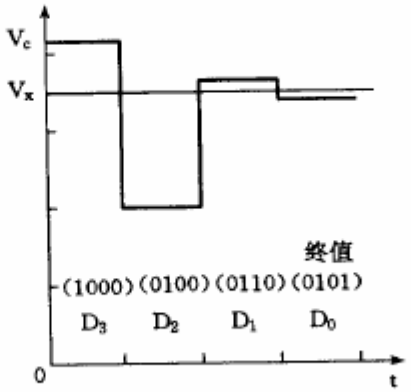
```
A0832   EQU 228H
MOV    DX，A0832+0      ；A0=0
OUT    DX，AL           ；AL的值为待转换的数字
MOV    DX，A0832+1      ；A0=1
OUT    DX，AL           ；AL的值无关
```

A/D接口电路设计

- 常用方式
  - 逐次逼近



（a）逐次逼近式A／D转换原理图



（b）逐次逼近过程原理图

逐次比较----DAC+比较器+SAR（逐次逼近寄存器）

AD转换器参数

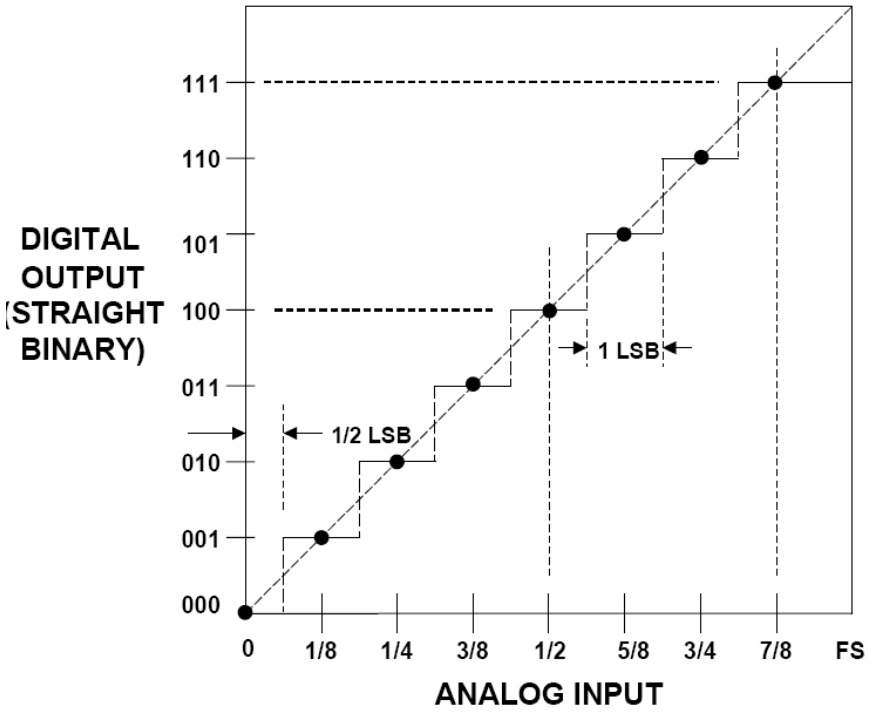
- 量程：输入模拟信号幅度范围，如1V，2V，5V；
- 带宽：输入模拟信号频率范围，如100MHz，1GHz；
- 转换速率：每秒能进行的转换次数，KHz，MHz ，1GHz ；
- 分辨率（Resolution）：能够分辨最小信号的能力，用ADC位数或每位对应的电压表示，能引起输出数字值发生一位变化的最小模拟输入变化。

例：位数8位，满量程5V，则其分辨率为8位，或 $5V/(2^8-1)=19.6mV$

AD转换器参数

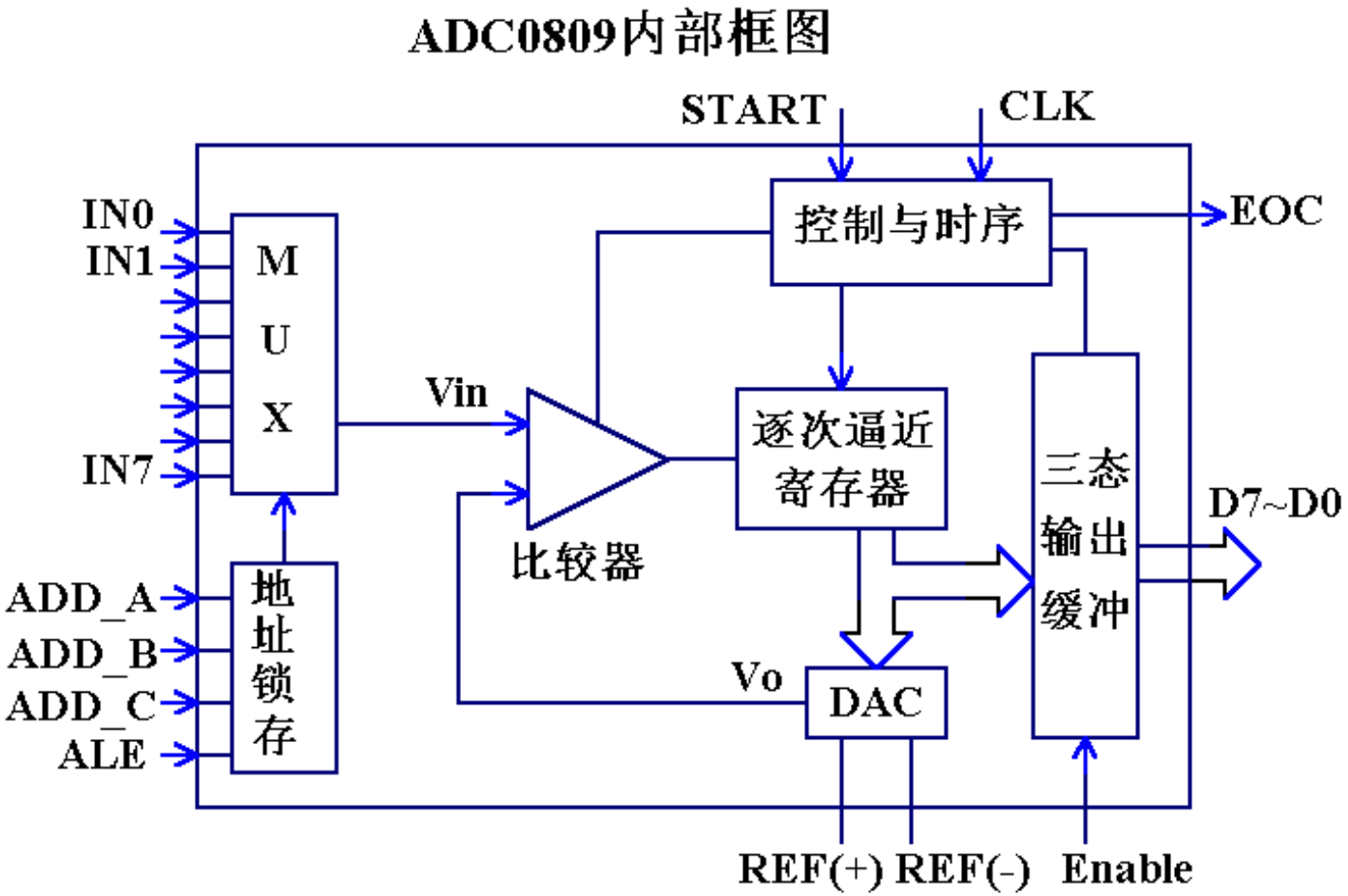
➢转换精度（Accuracy）：AD转换器实际输出值接近理想值的精确程度，用数字量的最低有效位LSB对应的模拟量△表示。

受分辨率限制，理想转换器输入在±0.5 △范围内变化时，输出对应同一个值，称其转换精度为±0LSB。与此相比较， 输入在±1△范围内变化时，输出对应同一个值，称其转换精度为±0.5LSB.

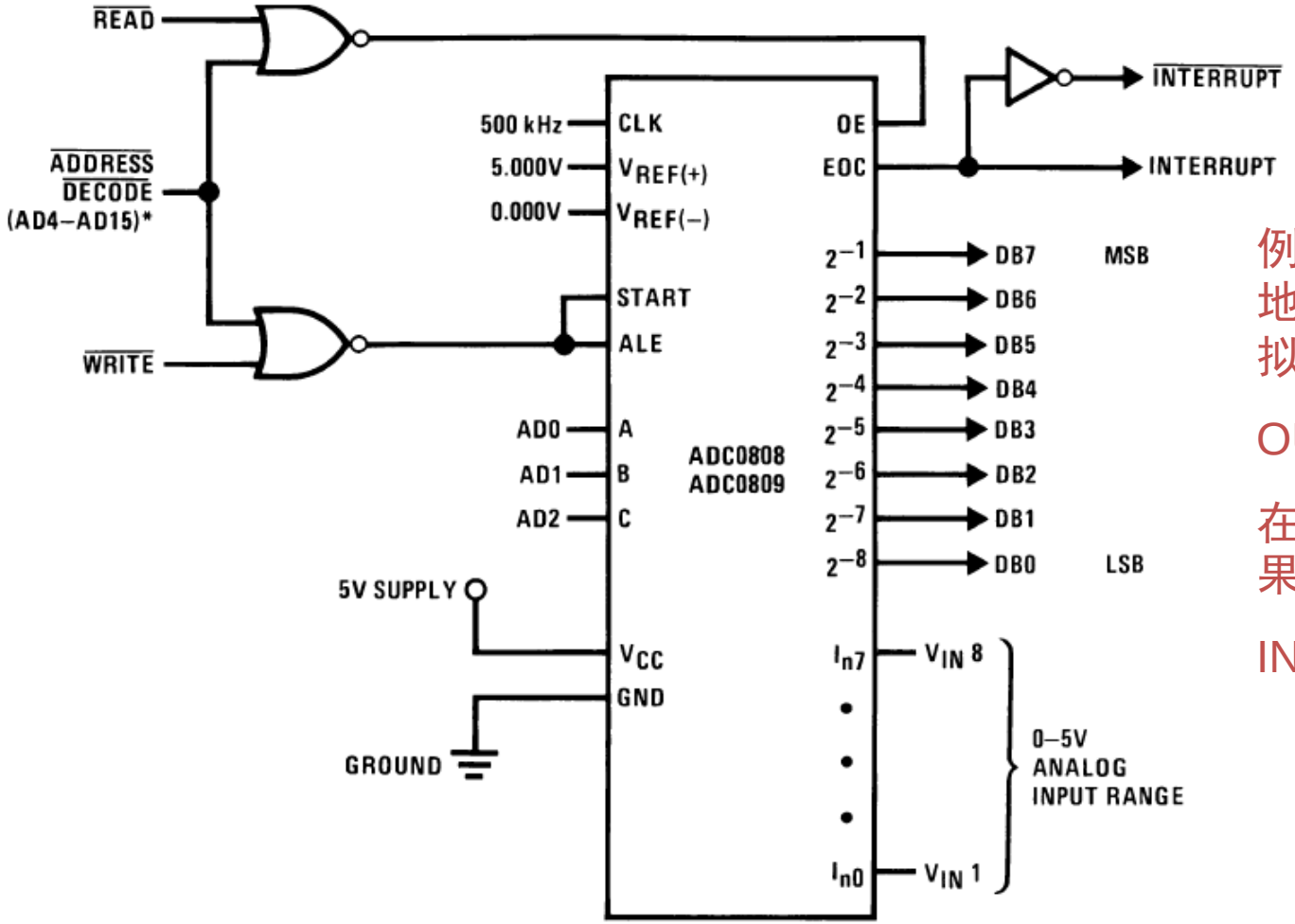


转换精度与分辨率有关系，但需注意精度和分辨率的区别，例如某ADC具有24位的分辨率，但其精度相当于16位，低8位表征随机噪声的影响。

ADC0809


ADC0809内部框图
START
CLK
IN0
IN1
IN7
M
U
X
控制与时序
EOC
Vin
逐次逼近
寄存器
三态
输出
缓冲
D7~D0
比较器
ADD_A
ADD_B
ADD_C
ALE
地
址
锁
存
Vo
DAC
REF(+)
REF(-)
Enable

ADC 0809与8088接口电路



例：设8路信号模拟输入地址为70～77H，对模拟通道2的转换命令为：

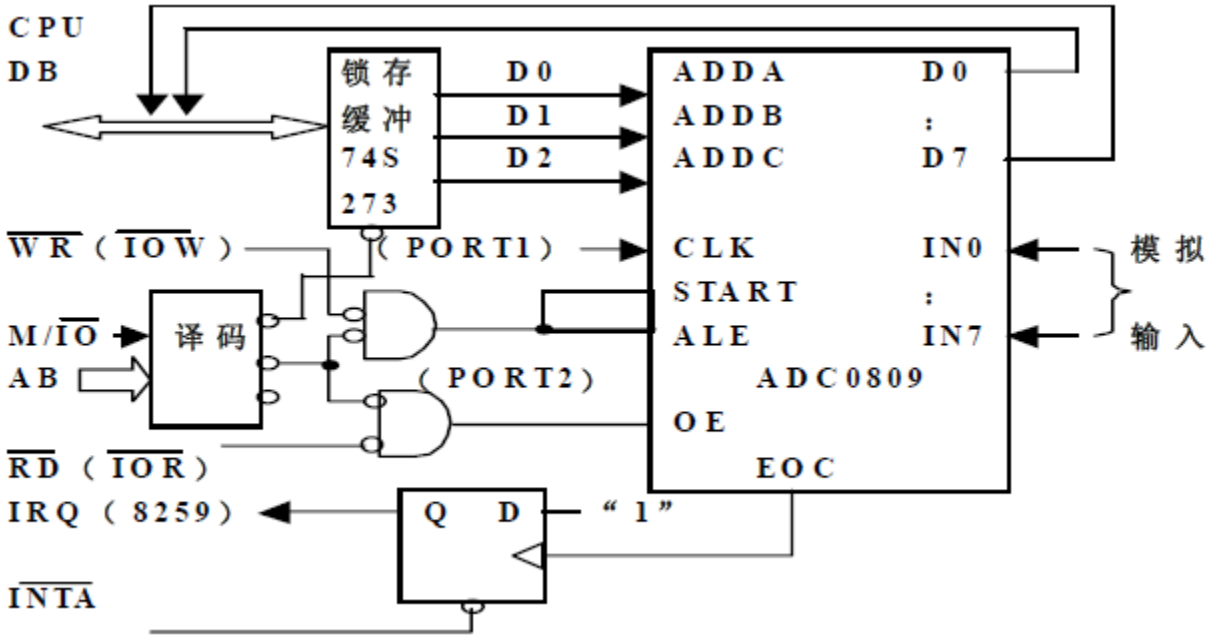
OUT 72H, AL

在中断程序中读转换结果的命令为：

IN AL, 72H

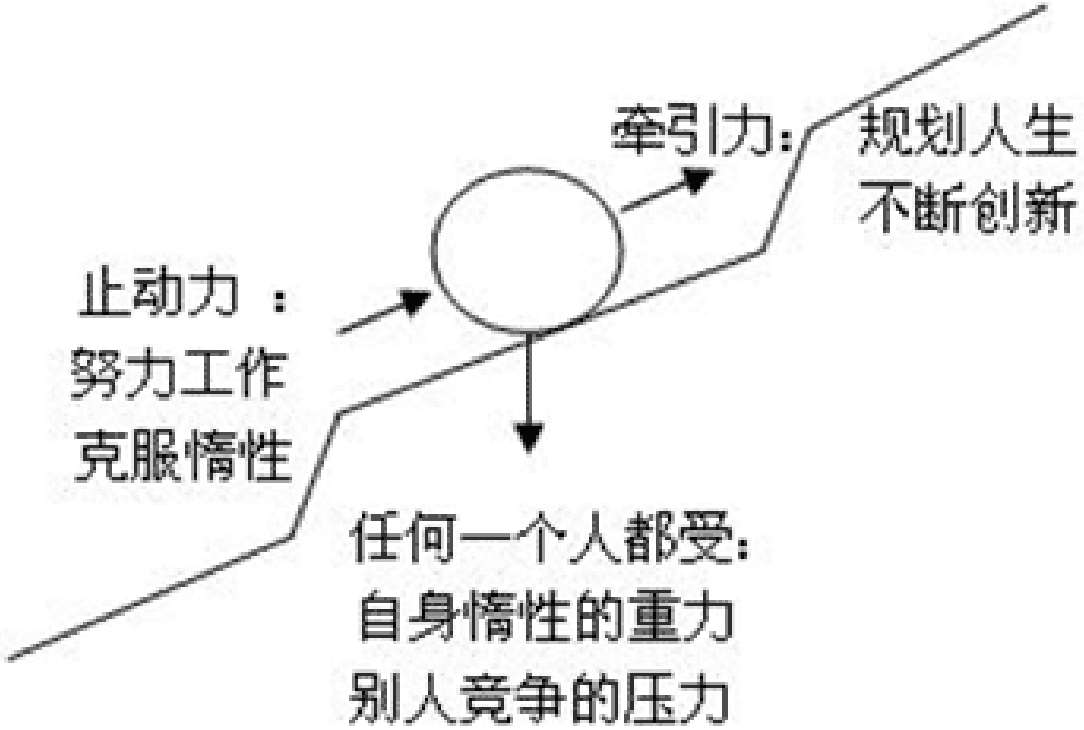
例一

- 设有8路外部模拟信号，现依次对它们进行A/D转换，并将转换结果存入BUFFER内存缓冲区，每个通道转换100次后结束。



自强不息

职业生涯的斜坡球理论：



人生的斜坡球理论图

有理想：追求自由、公平、正义
敢于追求一切美好的……

人性的光辉照亮雾霾的世界
祝大家都有光明的未来
和值得回味的人生