

第五章 输入输出和中断系统



信息科学与工程学院自动化系



本章重点及要求

- 1 输入/输出的基本概念
- 2 输入/输出数据的传送方式
- 3 中断技术
- 4 AT89C52的中断系统
- 5 学会AT89C52中断程序设计及应用
- 6 学会与中断有关的硬件连线



第五章 输入/输出和中断

5.1 输入/输出的基本概念

5.2 输入/输出数据的传送方式

5.3 中断技术

5.4 AT89C52的中断系统

5.5 AT89C52对外部中断源的扩展





5. 1 输入/输出的基本概念

5. 1. 1 输入/输出设备

5. 1. 2 输入/输出传送的信息

5. 1. 3 输入/输出接口的作用及其类型

5. 1. 4 I/O端口的编址方式



5. 1. 1 输入/输出设备

❓ 输入设备：

❓ 定义：往主机传送信息的设备

❓ 常用的输入设备：

键盘、鼠标、扫描仪、读卡机等

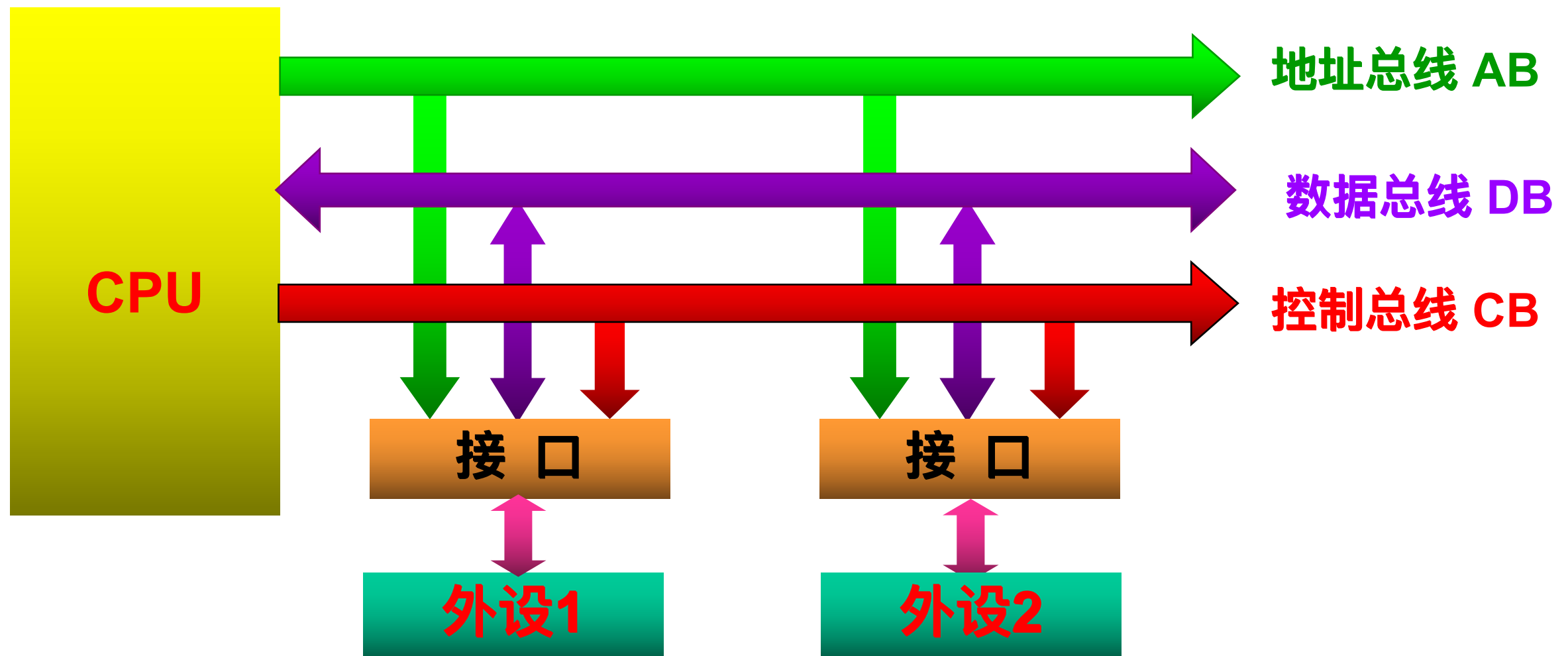
❓ 输出设备：

❓ 定义：接收主机信息并送出信息呈现给人们的设备

❓ 常用的输出设备：

显示器、打印机、绘图仪等

外设与CPU是通过接口连接



5. 1. 2 输入/输出传送的信息

❓ 数据信息

❓ 数字量

❓ 模拟量

❓ 开关量

❓ 状态信息

表征外设状态

❓ 控制信息

控制外设启停

二进制形式表示的数或以ASCII码表示的数或字符。

模拟量必须先经过A/D转换才能输入计算机，计算机的控制输出也必须先经过D/A转换才能控制执行机构。

只要用一位二进制数即可表示的量，如电机的运转与停止，开关的合与断，阀门的打开和关闭等。

在输入时，输入装置的信息是否准备好（Ready）；

在输出时，输出装置是否有空（Empty），若输出装置正在输出信息，则以Busy指示。



5. 1. 3 输入/输出接口的作用及其类型

计算机/单片机与外设之间起桥梁作用的
电路或部件——接口(interface)。
协调两者间的差异。

- 速度不一致
- 数据性质不同
- 数据协议/格式不同
- 所需功率/电平不匹配
- 所需测控点数量不匹配



接口电路的功能

❑ 变换作用

在CPU与外设之间进行信息类型、电平、传送接收方式变换

❑ 输出数据锁存

CPU速度快，数据在总线上保留时间短暂，无法满足慢速外设的数据接收，产生丢数现象

❑ 输入数据三态缓冲(隔离)

多个输入设备和数据源，防止冲突，只允许当前进行数据传输的数据源使用总线，其余处于隔离状态

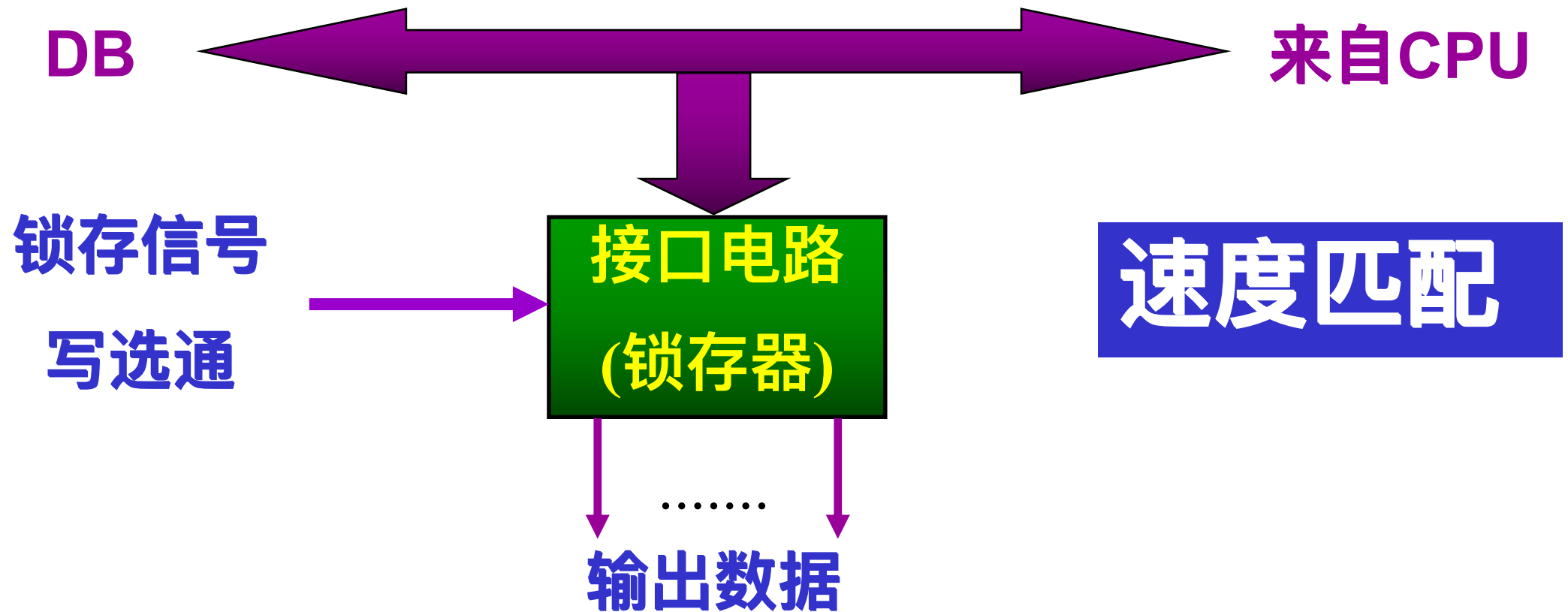
❑ 联络作用

在CPU与外设之间进行联络，Ready, Busy



输出接口电路——锁存作用

输出接口中必须含锁存器

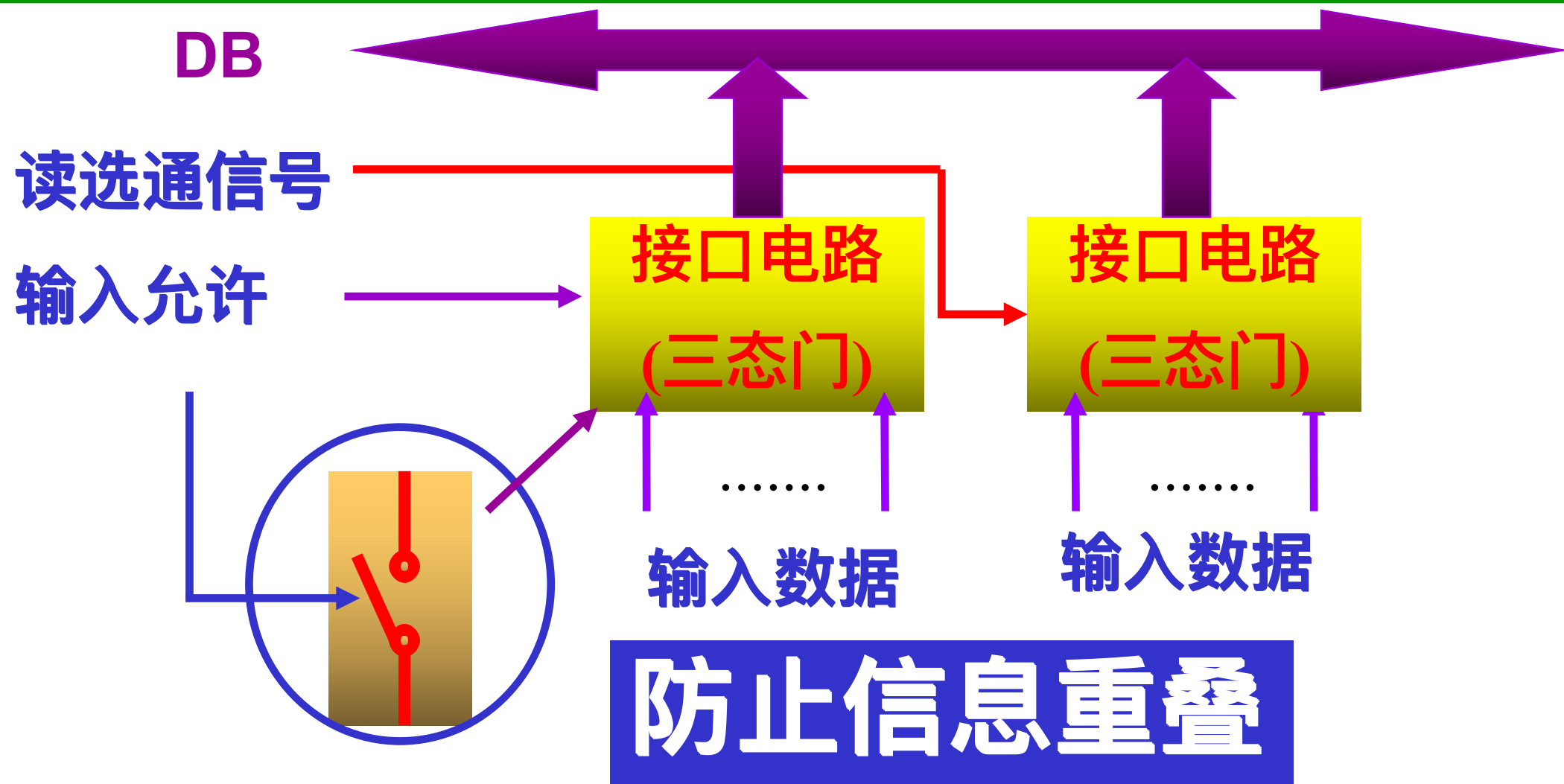


输出接口中需设置数据锁存器，将CPU输出的数据先锁存，再由外设慢慢处理；在锁存器的接收端，设置由锁存信号（写选通）控制的接收门电路，在CPU的控制下，确定是否向该接口电路发送数据。



输入接口电路——隔离作用

输入接口中必须含三态门



外部设备和存储器都不允许长期占用数据总线，而仅允许被选中的设备在读写周期（RD或WR）时享用数据总线。

输入接口中使设备的信息仅在CPU发来的“允许信号（读选通）”有效期间将数据与总线接通，其它时间该接口与总线相连接的线处于高阻浮空状态，起到与总线隔离的作用。



I/O接口的类型

1、I/O接口的种类很多，有两种基本类型

- 串行I/O接口
- 并行I/O接口

2、不可编程接口

例：74LS273、74LS373、74LS138 、74LS244

3、可编程接口

例：8155、8255、8250、8253



5. 1. 4 I/O端口的编址方式

在计算机中, 凡需进行读写操作的设备都存在着编址问题。

具体说来在计算机中有两种需要编址的器件: 一种是存储器, 另一种就是接口电路。

存储器是对**存储单元**进行编址, 而接口电路则是对其中的**端口**进行编址。对端口编址是为I/O操作而进行的, 因此也称为**I/O编址**。

常用的I/O编址有两种方式:

独立编址方式和统一编址方式。

❑ 统一编址

❑ I/O端口与内存存储器采用一套地址，完全象存储器单元一样处理，使用访问存储器的指令。

如：movx 类指令

❑ 独立编址

❑ 采用一套与存储器不同的地址，利用/MREQ（存储器请求）和/IORQ（输入输出请求），CPU 有专门的I/O指令 如：IN，OUT

❑ MCS51系列采用统一编址

❑ 片内RAM 与P0~P3均用MOV指令

❑ 片外RAM与外扩口I/O口均用MOVX指令

接口 (Interface) 与端口 (Port)

I/O接口：单片机与外设间的I/O接口芯片。

一个I/O接口芯片可以有多个I/O端口。

I/O端口：简称口，指具有端口地址的寄存器或缓冲器，可被独立选通的口电路。

端口地址：端口在系统中被分配的惟一地址，简称口地址。

I/O端口编址是给所有I/O接口中的寄存器**编址**。



接口与端口

端口可分为：

输入端口——CPU从中读取外设的状态或数据信息

输出端口——CPU通过它输出控制信号或数据信息

状态端口——CPU从中读取外设的状态信息

控制端口——由CPU控制输出控制信号

数据端口 ——可以是输入数据端口或输出数据端口

I/O信息都是通过数据总线传递的。



5.2 输入/输出数据的传送方式

5. 2. 1 无条件传送方式 (又称同步传送)

5. 2. 2 查询式传送方式 (又称条件传送——异步传送)

5. 2. 3 中断传送方式

5. 2. 4 DMA传送方式



5. 2. 1 无条件传送方式（又称同步传送）

无条件传送方式

也称为同步程序传送, 在需要的时刻让CPU直接与外设进行输入/输出操作, CPU仅需要通过I/O指令即可由接口获取外设数据或为外设提供数据, 类似于CPU和存储器之间的数据传送。

只有那些一直为数据I/O传送作好准备的外部设备, 才能使用无条件传送方式。这种传送方式不需要测试外部设备的状态, 可以根据需要随时进行数据传送操作。

◆ 应用于定时为已知的
且 固定不变的低速I/O

◆ 无需等待的高速I/O

◆ 输入时:

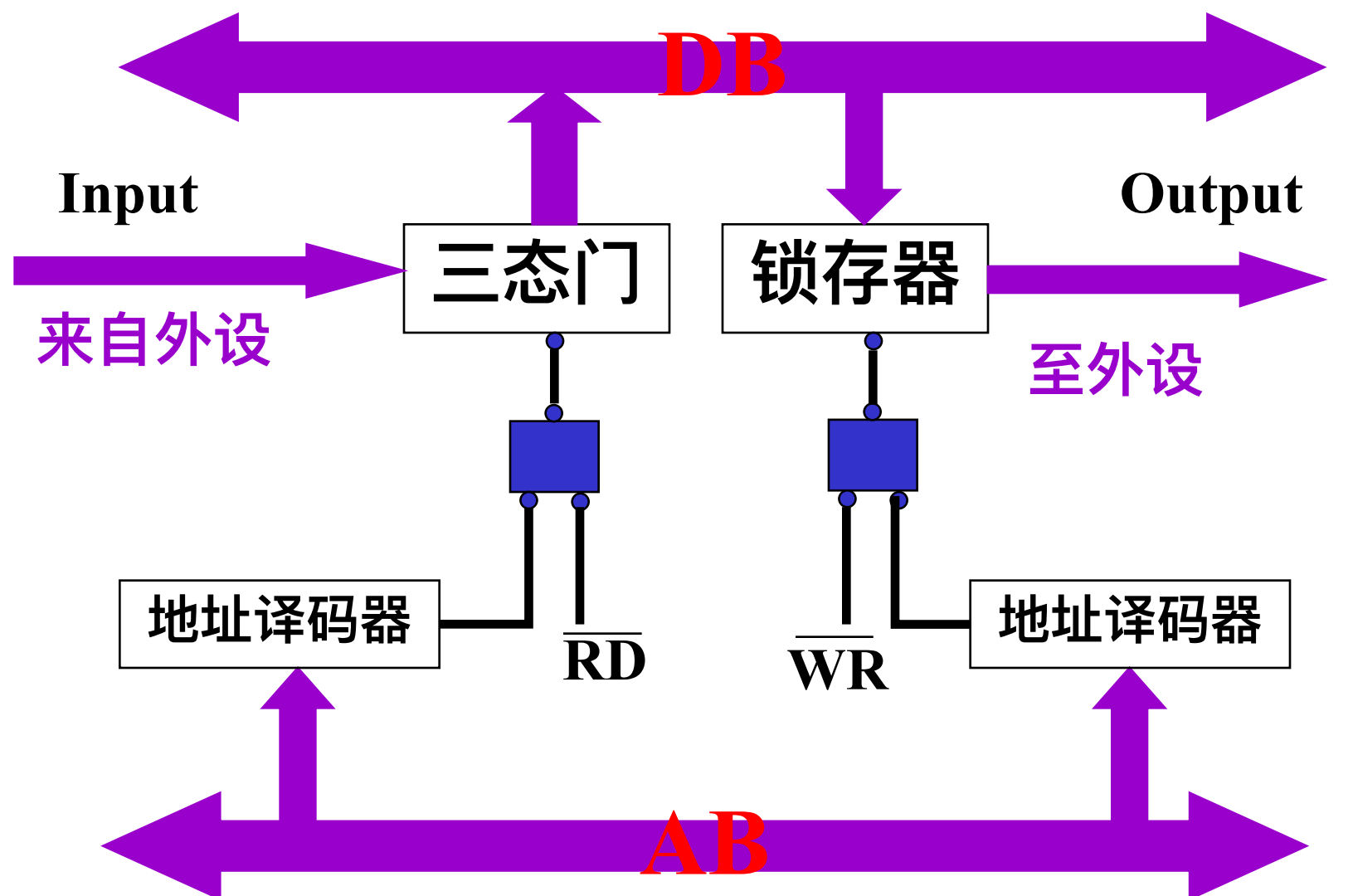
`mov DPTR, #PORT`

`movx A, @DPTR`

◆ 输出时:

`mov DPTR, #PORT`

`movx @DPTR, A`



Input:
译码器和/RD有效, 选通缓冲器, 外设数据经DB输入A;

Output:
译码器和/WR有效, 选通锁存器, 把DB的输出数据接收到锁存器;

无条件I/O举例——显示程序设计



5. 2. 2 查询式传送方式

(又称条件传送——异步传送)

查询传送又称为**条件传送**, 即数据的传送是有条件的。

在输入/输出之前, 先要检测外设的状态, 以了解外设是否已为数据输入输出作好了准备, 只有在确认外设已“准备好”的情况下, CPU才能执行数据输入/输出操作。

通常把通过程序对外设状态的检测称之为“查询”, 所以这种有条件的传送方式又叫做**程序查询方式**。

❓ CPU查询外设状态信息(Ready,Busy), 条件 满足时, 进行数据传送

❓ 程序简单

❓ 高速CPU查询低速外设

❓ 浪费CPU时间, 效率低

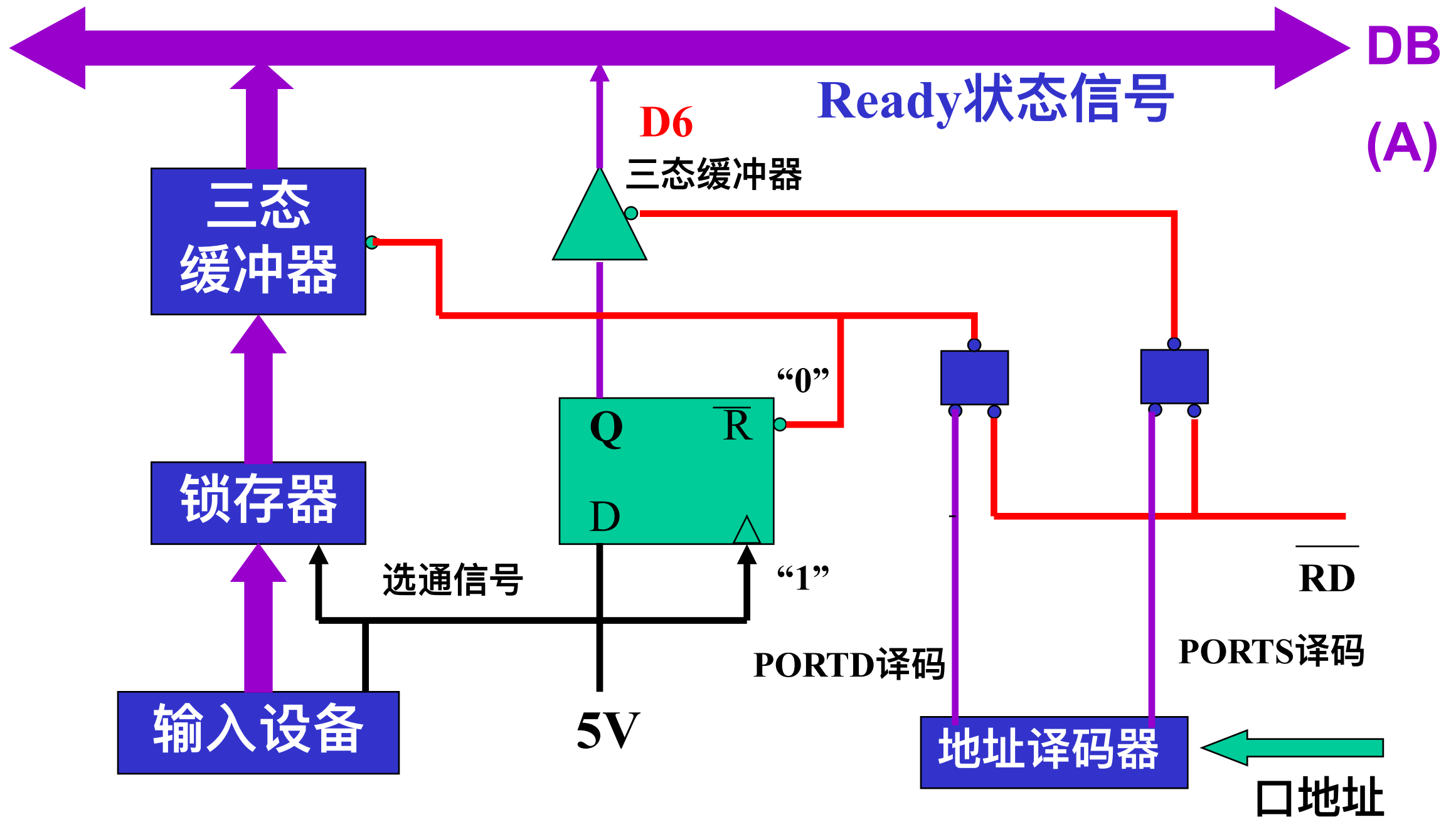
❓ 输入接口原理图

❓ 输出接口原理图

❓ 程序流程图



输入接口原理图



当外设准备好，置入状态信号 **Q=1** →

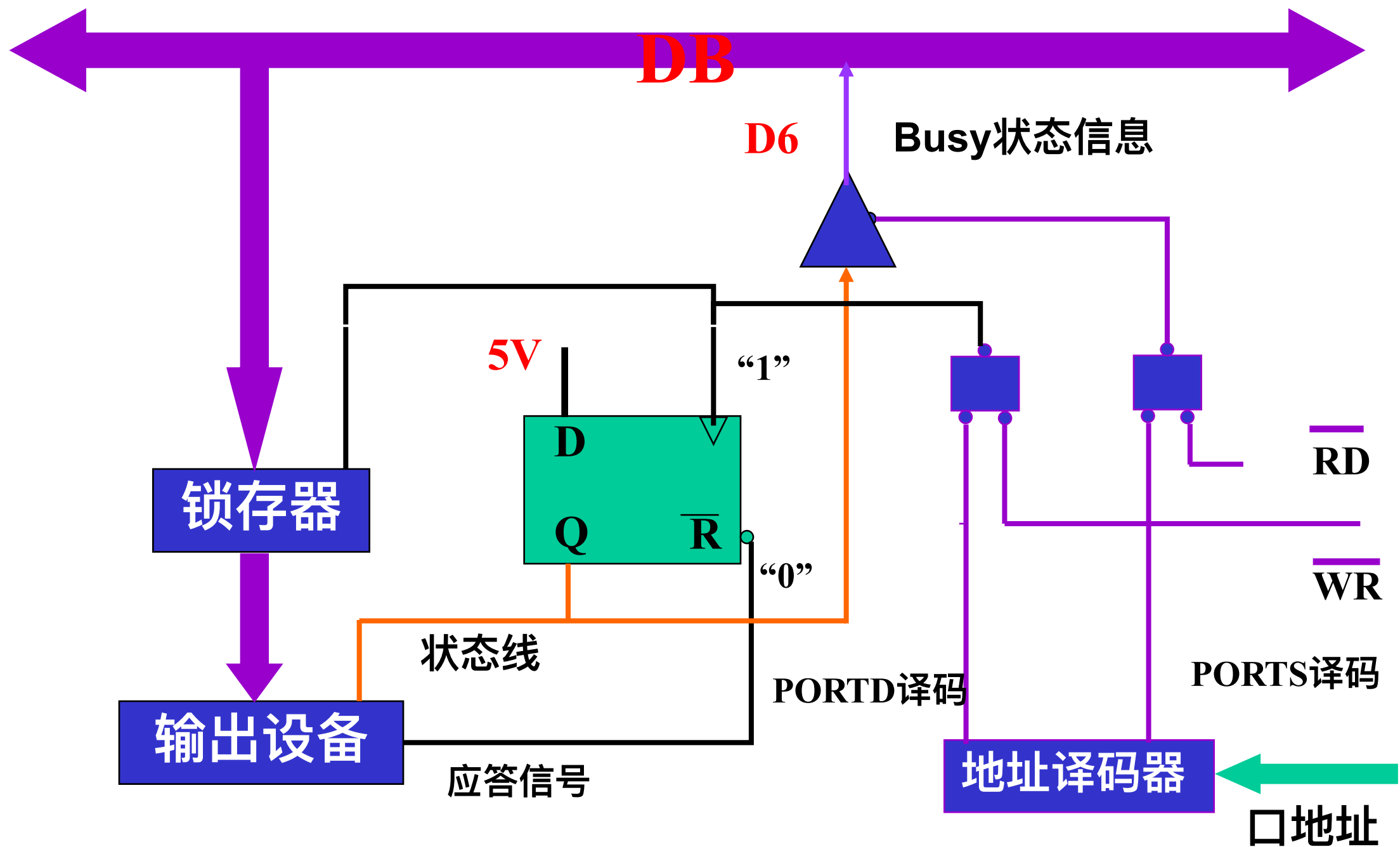
CPU读数据口时清掉状态信号

```
MOV DPTR, #PORTS
MOVX A, @DPTR
```

```
MOV DPTR, #PORTD
MOVX A, @DPTR
```



输出接口原理图



外设收到CPU数据后以“应答信号”清掉状态信号
(使状态触发器清0, 外设将数据取走)

CPU写入数据后, 置入状态信号 $Q=1$

→ `MOV DPTR, #PORTS`
`MOVX A, @DPTR`

← `MOV DPTR, #PORTD`
`MOVX @DPTR, A`



条件传送方式程序举例

Test: mov DPTR, #PORTS

movx A, @DPTR

anl A, #40H

jz Test ; (jnz Test)

mov DPTR, #PORTD

movx A, @DPTR ; 输入

(movx @DPTR, A ; 输出)

Ready=1

Busy=0

输入接口状态

准备好?

NO

YES

传送信息

(A)

D6

0/1

状态信号



5. 2. 3 中断传送方式

由于**查询传送方式**为CPU主动要求传送数据，而它又不能控制外设的工作速度，因此只能用等待的方式来解决配合的问题。

中断方式则是在外设为数据传送作好准备之后，就向CPU发出**中断请求信号**(相当于通知CPU)，CPU接收到中断请求信号之后立即作出响应，**暂停**正在执行的原程序(主程序)，而转去为外设的数据输入输出服务，待服务完之后，程序**返回**，CPU再继续执行被中断的原程序。

[?] 中断：要求进行输入、输出的外设，发出就绪信号给CPU，作为中断请求，打断CPU正在进行的工作，即中断CPU正在执行的程序。

[?] 中断过程

[?] 中断方式与查询方式比较

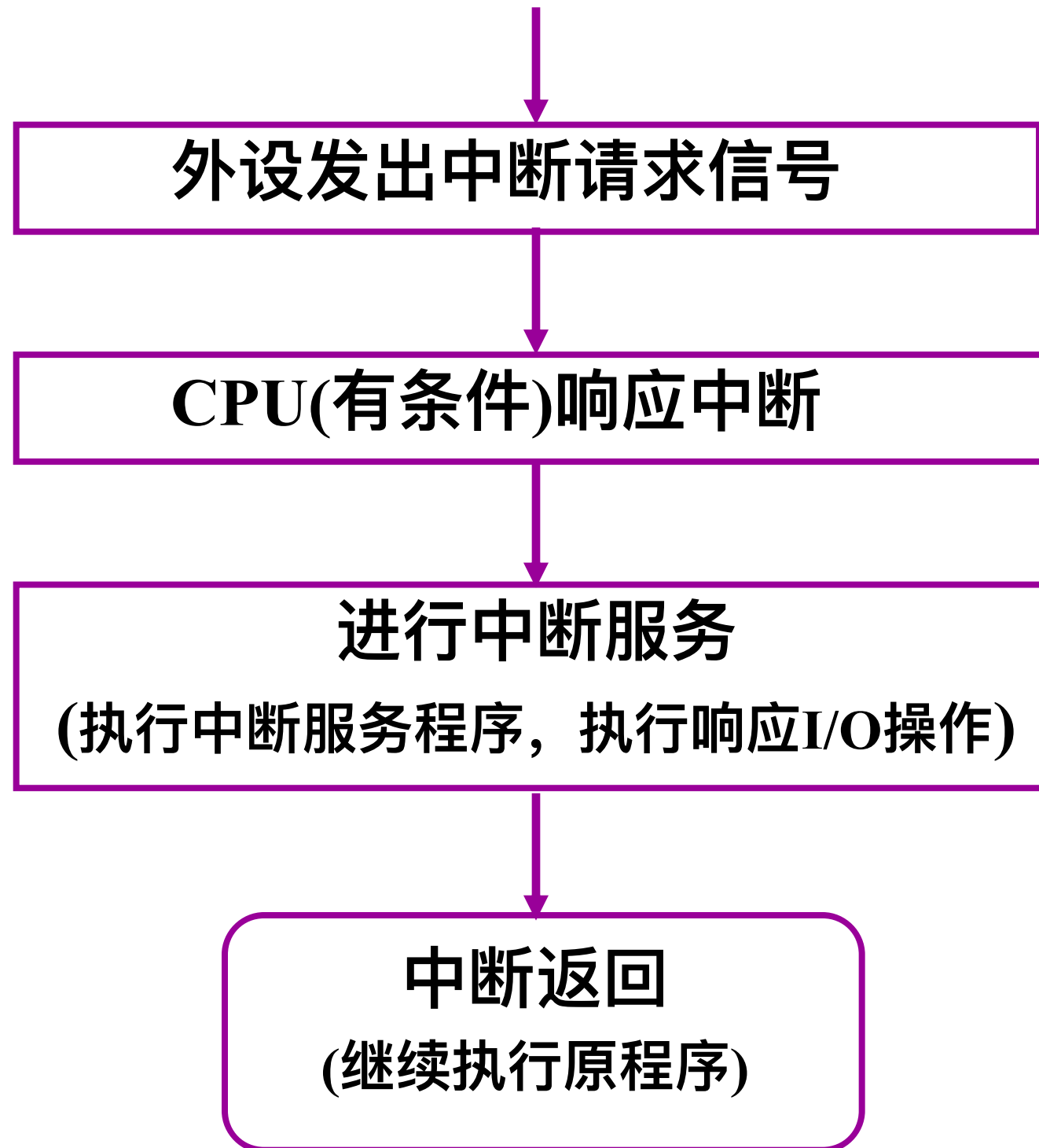
[?] 提高了CPU的效率

[?] CPU与外设可并行工作

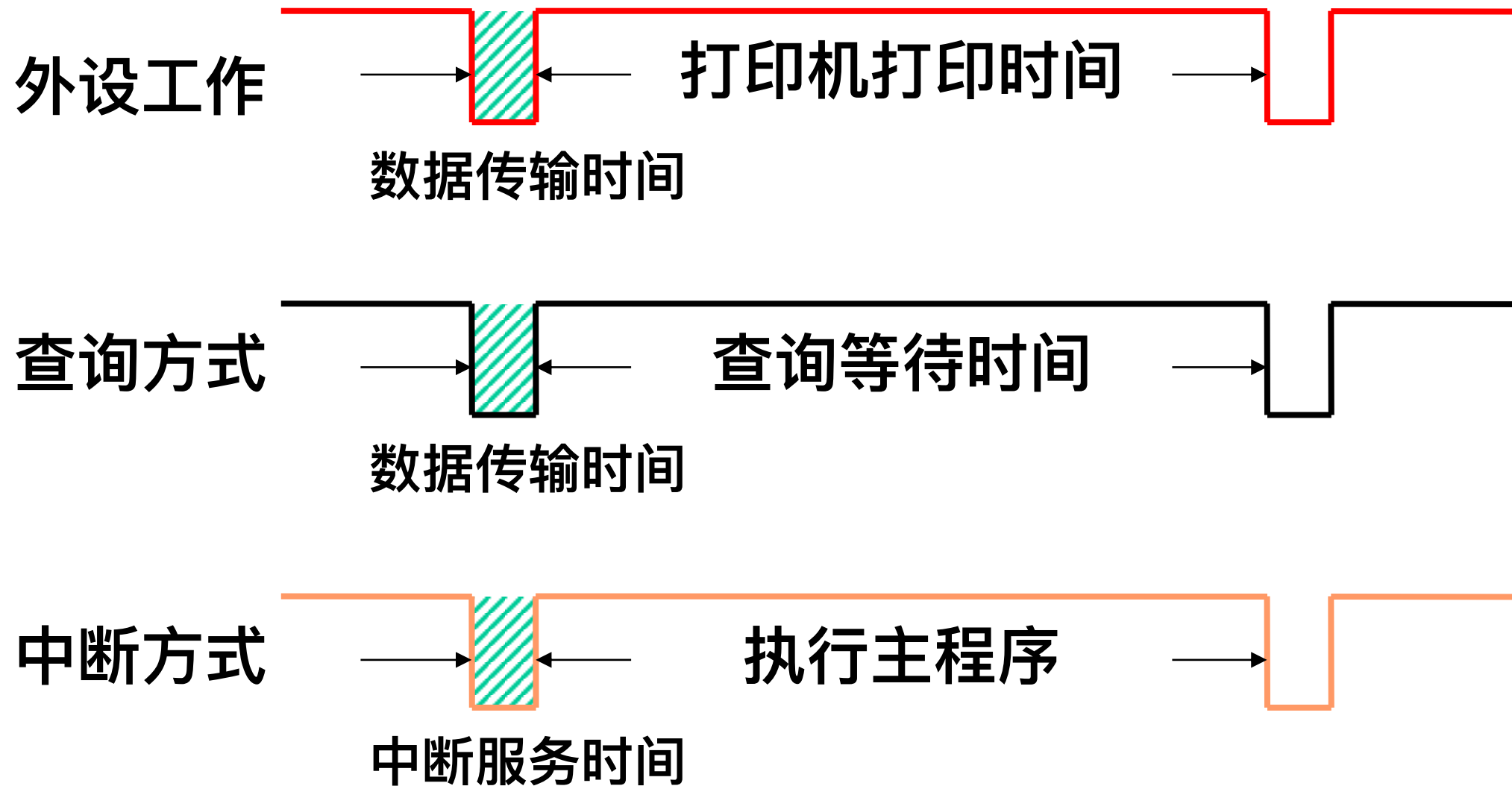
[?] CPU可及时响应外部事件



中断过程



中断方式与查询方式CPU占用时间比较



中断方式可使CPU和外设（多个外设）并行工作，当外设准备好后进入中断服务子程序，服务完后返回主程序，可以提高CPU的工作效率。



5. 2. 4 DMA传送方式

DMA (直接存储器存取)传送方式是CPU让出总线，使外设与存储器之间直接传送数据的方式。

❑ 利用DMAC(DMA控制器)实现成组、大批量的数据在内存和外设之间的快速传送。

❑ DMA过程:

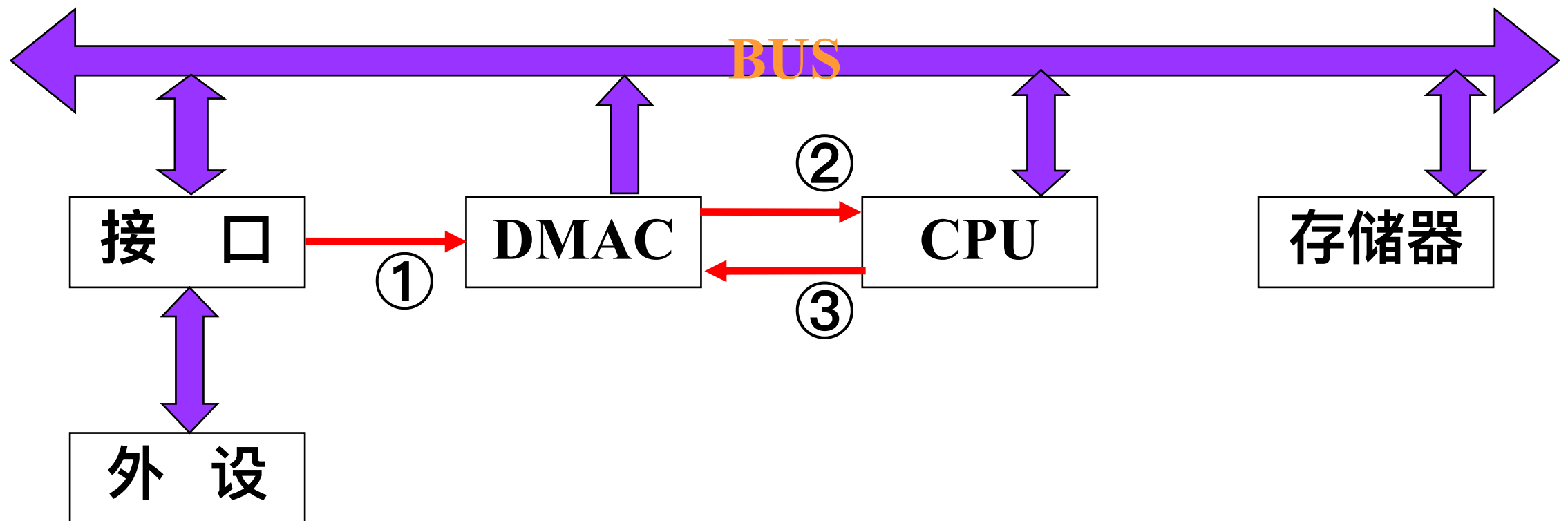
❑ 优点：速度快

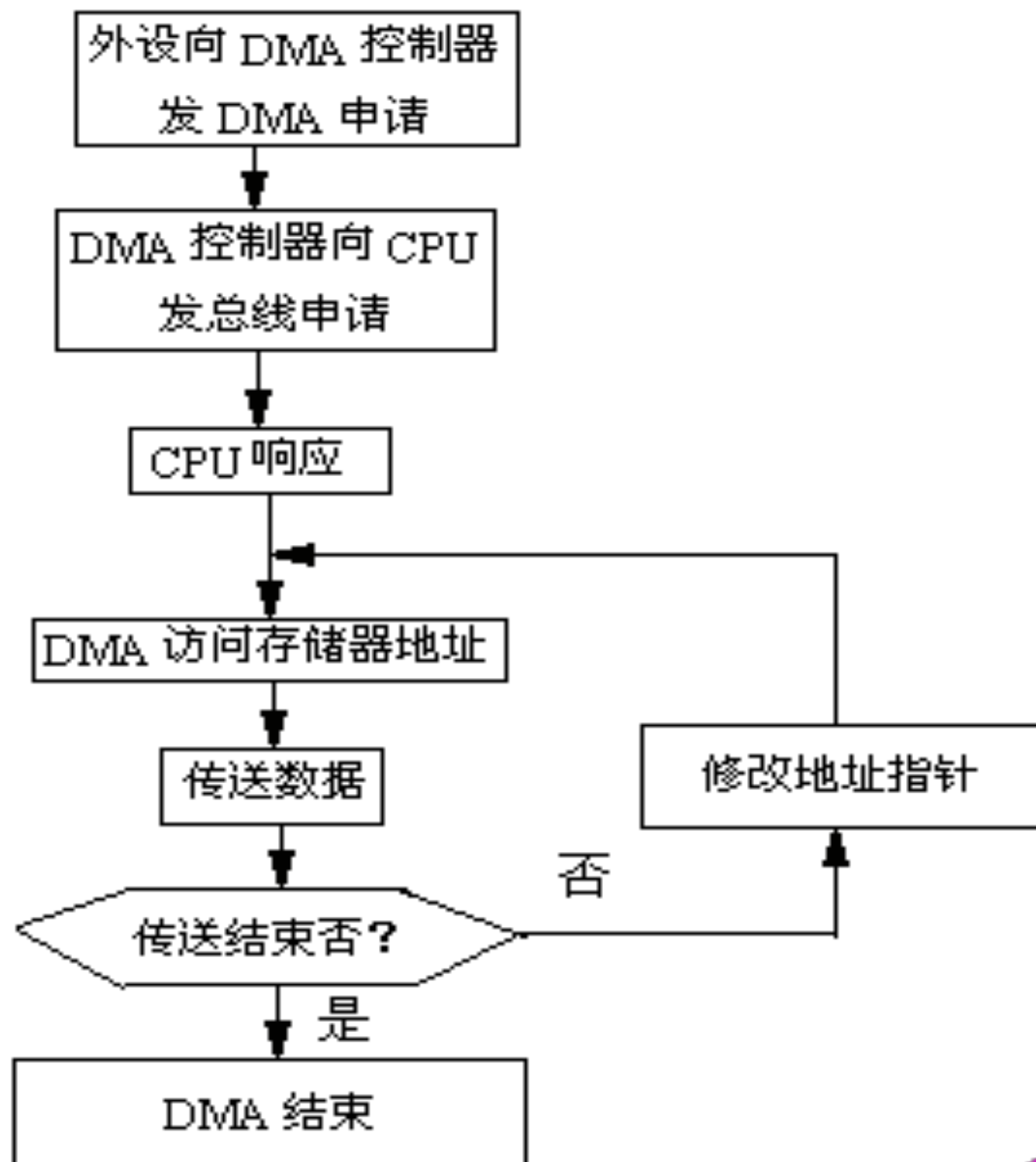
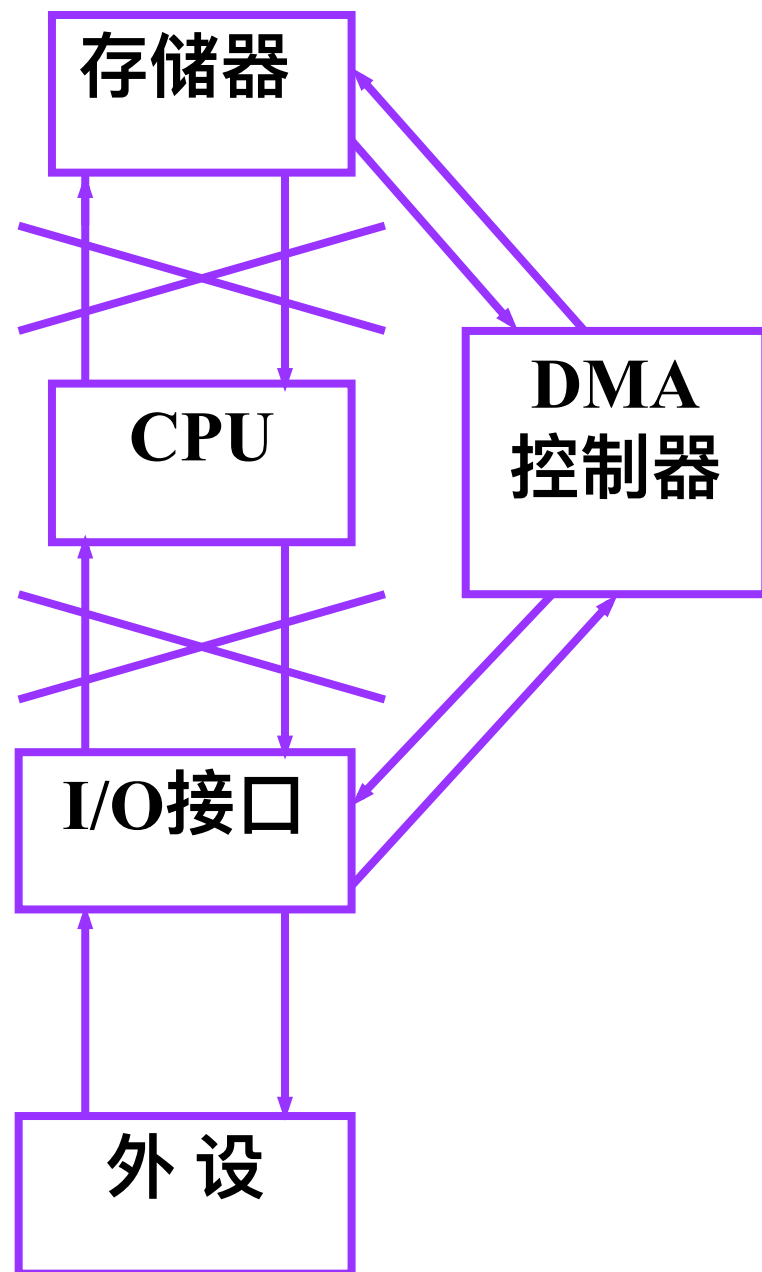
❑ 缺点：硬件复杂，成本增加



DMA过程

- ❑ 外设向DMAC发DMA申请
- ❑ DMAC向CPU发BUSRQ（总线请求）
- ❑ CPU发BUSAK响应，DMAC接管系统总线，进入DMA方式
- ❑ DMAC发存储器地址和读写信号，数据传送
- ❑ DMA控制结束，恢复CPU对总线控制





5.3 中断技术

5. 3. 1 中断的定义及必要性

5. 3. 2 中断源

5. 3. 3 中断的分类

5. 3. 4 中断系统的功能

5. 3. 5 中断处理过程



5. 3. 1 中断的定义及必要性

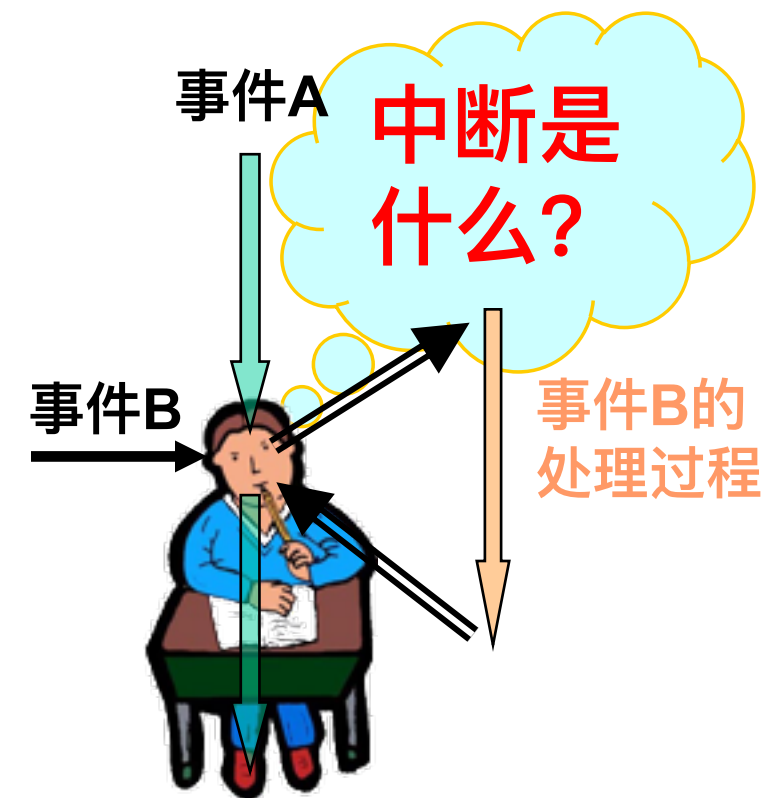
定义：所谓中断是指中央处理器CPU正在处理某件事的时候，外部发生了某一事件，请求CPU迅速处理（**中断发生**），CPU暂时中断当前的工作，转入处理所发生的事件（**中断响应和中断服务**），处理完后，再回到原来被中断的地方，继续原来的工作（**中断返回**）。这样的过程称为**中断**。

中断的概念

- CPU在处理某一事件A时，发生了另一事件B请求CPU迅速去处理（中断发生或中断请求）；
- CPU暂时中断当前的工作，转去处理事件B（中断响应和中断服务）；
- 待CPU将事件B处理完毕后，再回到原来事件A被中断的地方继续处理事件A（中断返回）。

中 断

中断请求→中断响应→中断服务→中断返回



必要性及应用

中断功能便于实现

1. 分时操作
2. 实时处理
3. 故障处理
4. 主机与外设之间的速度匹配

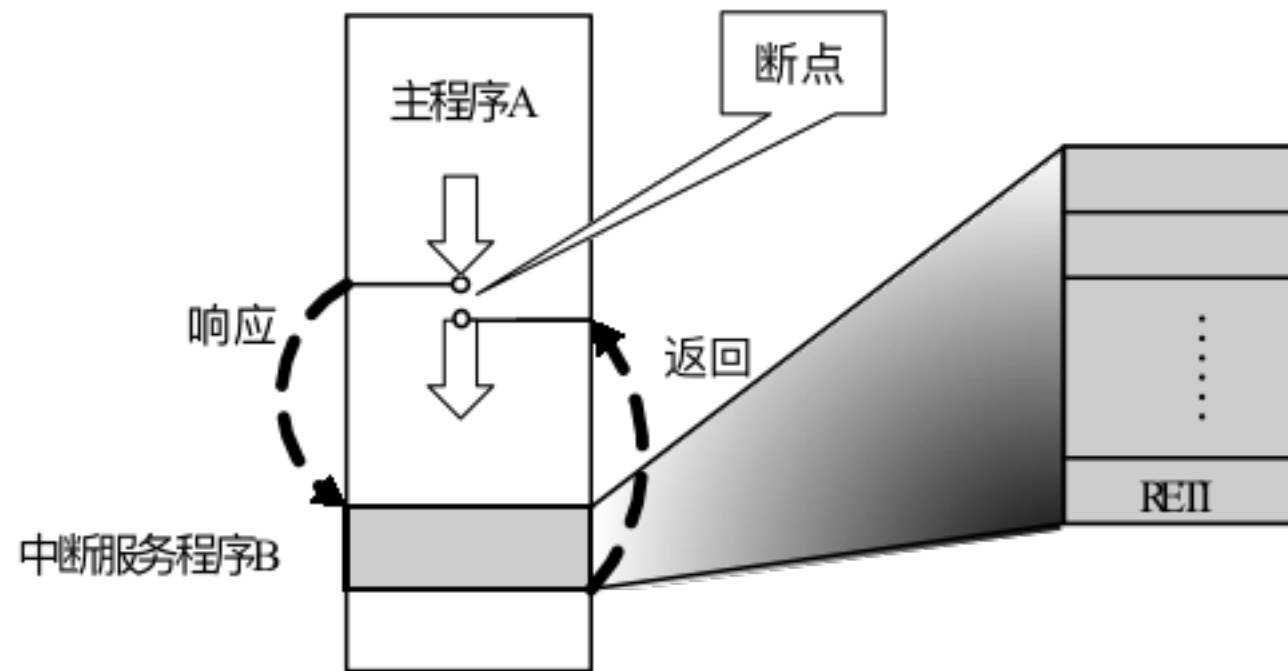
计算机在运行过程中，往往会出现事先预料不到的情况，或出现一些故障：如电源突跳，存储出错，运算溢出等等。计算机就可以利用中断系统自行处理，而不必停机或报告工作人员。

CPU和外设同时工作；CPU可以通过分时操作启动多个外设同时工作，统一管理。大大提高了CPU的利用率，也提高了输入、输出的速度。

当计算机用于实时控制时，中断是一个十分重要的功能。现场的各个参数、信息，需要的话可在任何时候发出中断申请，要求CPU处理；CPU就可以马上响应（若中断是开放的话）加以处理。这样的及时处理在查询的工作方式是做不到的。



5. 3. 2 中断源



引起CPU中断的根源，称为**中断源**。中断源向CPU提出的中断请求。CPU暂时中断原来的事务A，转去处理事件B。对事件B处理完毕后，再回到原来被中断的地方（即断点），称为中断返回。实现上述中断功能的部件称为中断系统（中断机构）。

中断源类型：



- ❑ 外设中断请求
- ❑ 实时时钟中断请求
- ❑ 控制对象中断源
- ❑ 故障引起的中断

强迫中断

引起的中断都是**随机**。

- ❑ 人为设置的中断

人为设置的中断，不是随机的，故称为**自愿中断**。



5. 3. 3 中断的分类

1. 可屏蔽中断 (Maskable Interrupt)

CPU对 INT 中断请求输入可以控制（屏蔽），通过中断控制指令；

2. 非屏蔽中断 (Non Maskable Interrupt)

CPU对 NMI 中断请求不能被用户用软件屏蔽，必须响应；

3. 软件中断 (Software Interrupt)

可以通过相应的中断指令使CPU响应。

中断类型	中断请求信号
可屏蔽中断	INT 中断请求输入线上输入
非屏蔽中断	NMI 中断请求输入线上输入
软件中断	用中断指令使CPU响应中断

中断类型	CPU响应中断的方式
可屏蔽中断	开中断指令 响应 关中断指令 屏蔽 INT 上来的低电平中断请求
非屏蔽中断	CPU必须予以响应，不能由软件指令屏蔽
软件中断	CPU只要执行这种指令，完成相应的中断功能

MCS-51就是具有可屏蔽中断功能的一类CPU。



5. 3. 4 中断系统的功能

❓ 中断技术

—对中断全过程的分析、研究及实现的方法

❓ 中断系统

—包括中断源的产生、中断判优、中断响应、中断处理等实现这一全过程的硬件和软件。



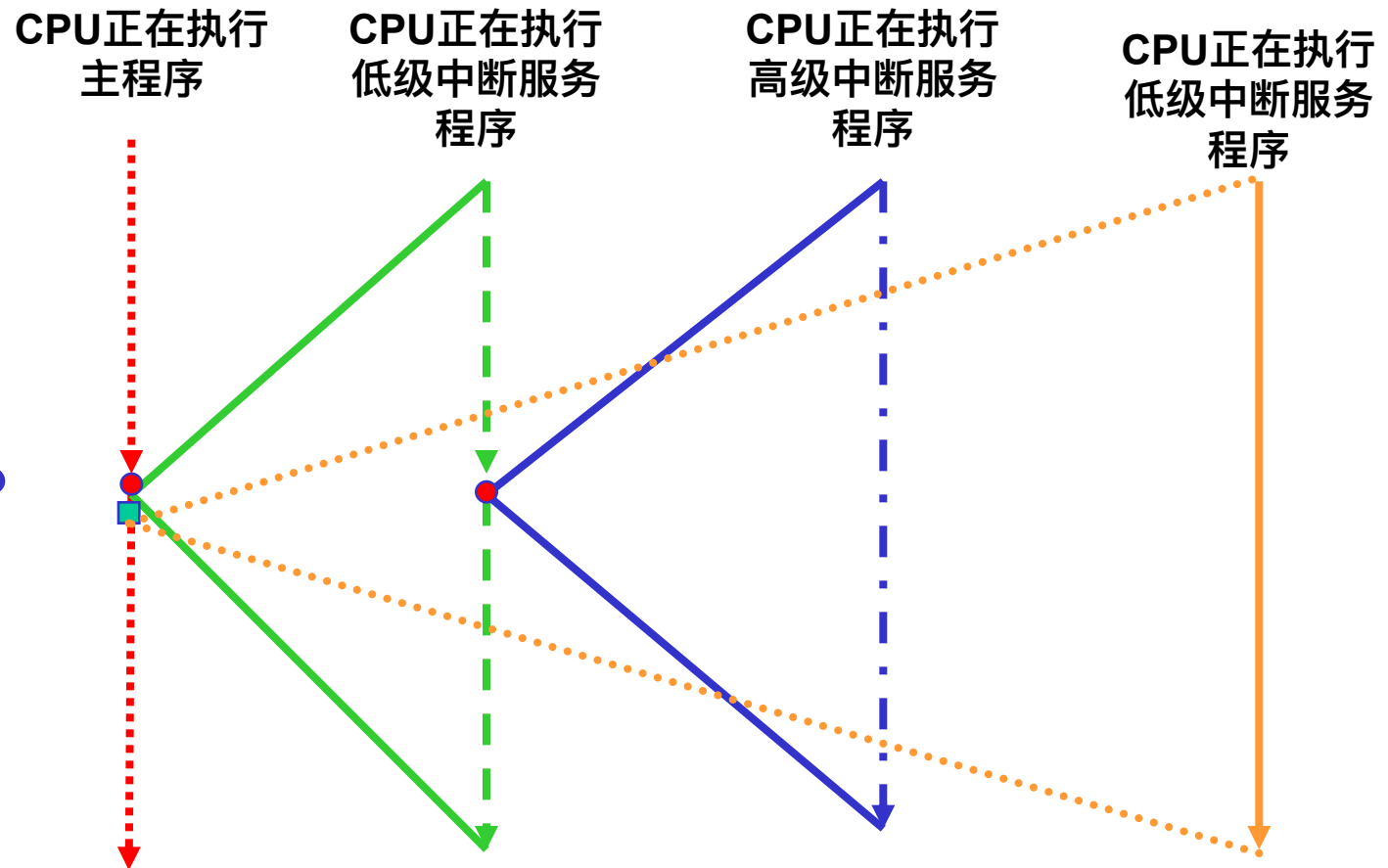
中断判优

?多中断源同时发出中断请求，CPU根据中断的优先级判断优先执行的中断请求。

?中断嵌套

条件：中断服务程序的开头设置一条开中断的指令；

中断优先权更高中断源的中断请求存在。



中断响应

保护断点

把当前PC的值，即被中断程序的下一条指令的地址保存到堆栈中。

转入中断服务子程序

把中断入口地址送入PC，以转入相应的中断服务程序。



中断处理

[?] 中断响应条件

计算机运行时，不是任何时刻都去响应中断请求，而是在中断响应条件满足后才会响应。

[?] CPU处于开中断状态：51内部有中断允许触发器EA
EA=0 禁止中断；EA=1 开放中断。

[?] 中断响应过程

[?] 中断响应：停止现行程序，
转向中断处理程序入口地址

[?] 关中断 （MCS-51响应中断后，不自动关中断）

[?] 保护断点 （自动将断点地址压入堆栈）

[?] 执行中断处理程序



执行中断处理程序

? 保护现场

- 保护中断服务程序中用到的寄存器和状态标志的内容

```
push PSW  
push ACC  
push DPH  
push DPL
```

? 中断服务

- 相应的中断源服务，完成一定的I/O操作

? 恢复现场

- 完成中断服务后，将保存在堆栈中的现场数据恢复

```
pop DPL  
pop DPH  
pop ACC  
pop PSW
```

? 开中断和中断返回

- RETI (中断返回指令)
- 恢复现场后的开中断是为了返回被中断的程序之后，CPU能响应新的中断。



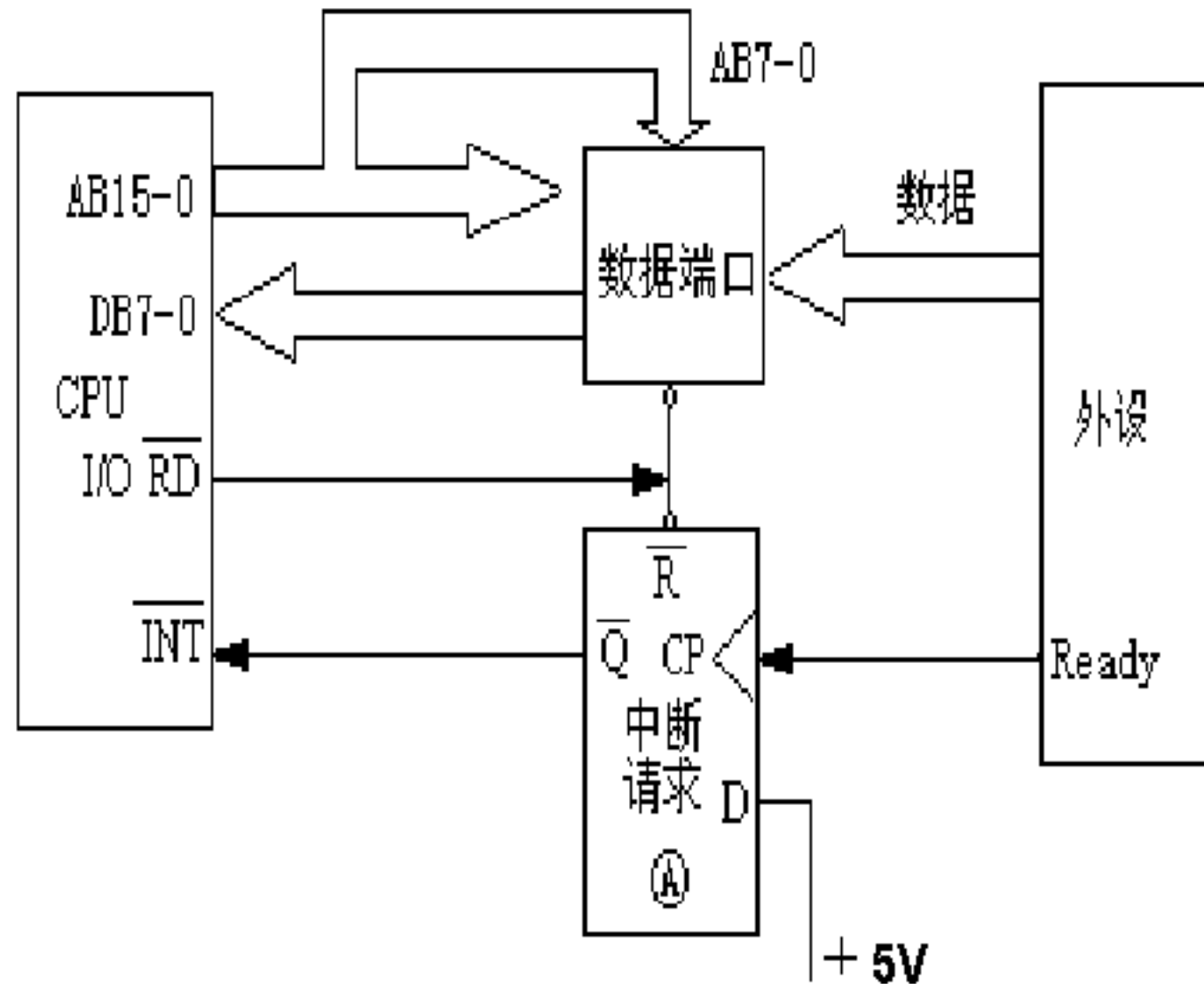
中断响应条件

CPU响应中断的条件主要有以下几点：

(1) 设置中断请求触发器 (又称中断标志触发器)

每个中断源需有一个中断请求触发器，保持中断请求信号，直至CPU响应这个中断后，才可清除中断请求。

当中断请求触发器为“1”状态时（ $Q=1$ ），表示有中断产生； $Q=0$ 表示没有中断产生。

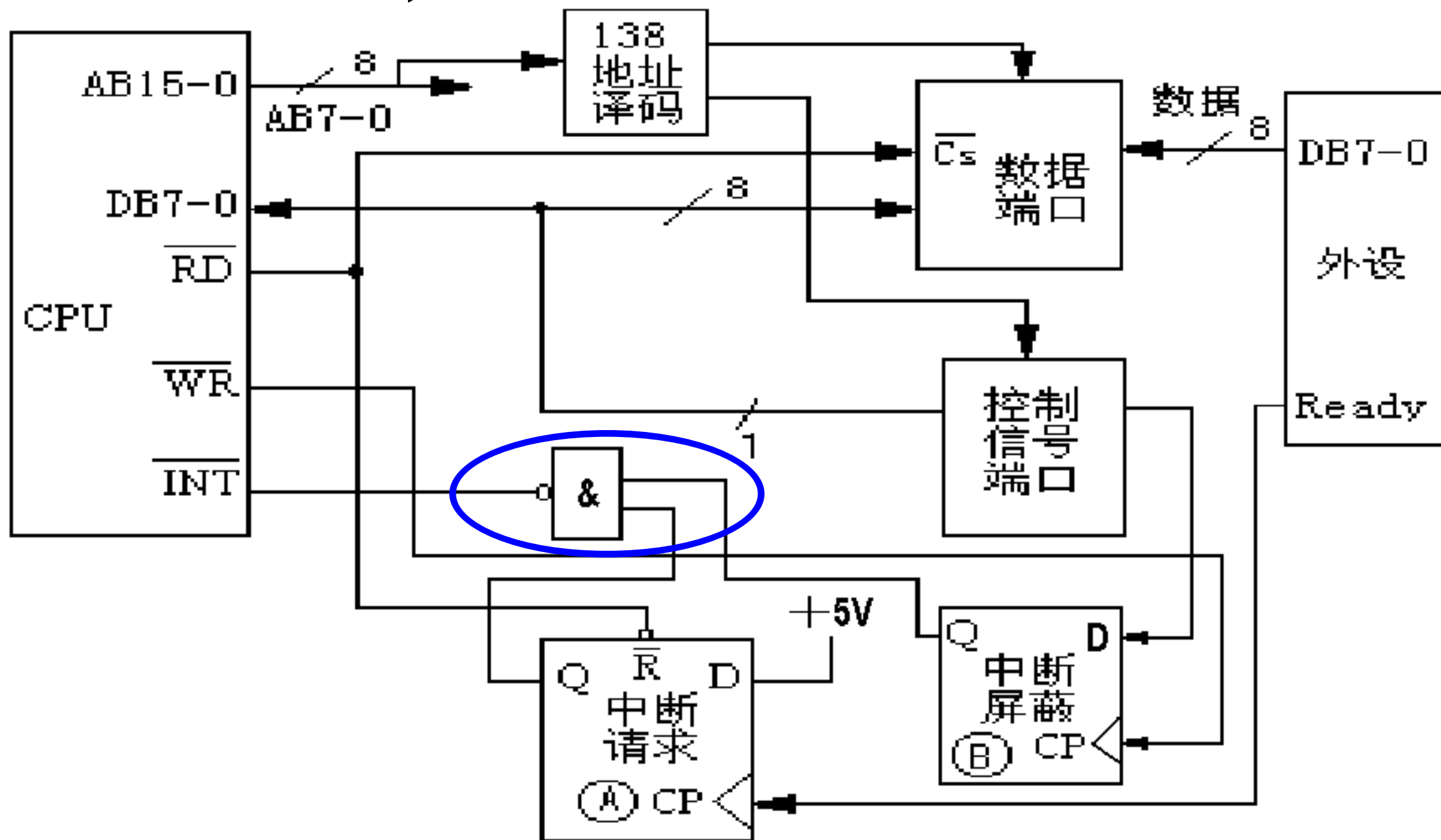


设置中断请求的情况

CPU 中也要有中断标志触发器，有中断请求则置1。

(2) 需设置中断屏蔽触发器(又称中断允许触发器)

每一个中断源的接口电路中需增加一个中断屏蔽触发器，只有当此触发器为“1”时，中断源的中断请求才能被送出至CPU。



(3) 中断是开放的

在CPU内部有一个中断允许触发器EA。只有当EA为“1”时（即中断开放时），CPU才能响应中断；若EA为“0”（即中断是关闭的），这时外部有中断请求，CPU不会响应，EA的状态是可以由指令来控制的（称为可编程的）。

(4) 需设中断优先级触发器

在CPU内部要设置中断优先级触发器，以实现嵌套；

在各中断源中也要设一个中断优先级触发器，由指令置位/复位，以表示本中断源在中断系统中的优先级。

(5) CPU在现行指令结束后响应中断

在开中断情况下，若中断源有中断请求，CPU也并不是立即响应，只是当正在执行的指令运行到最后一个机器周期T状态时，CPU才采样中断源。若发现有中断请求，则把内部的中断标志锁存器置“1”，然后下一个机器周期不进入取指周期，而进入中断周期。



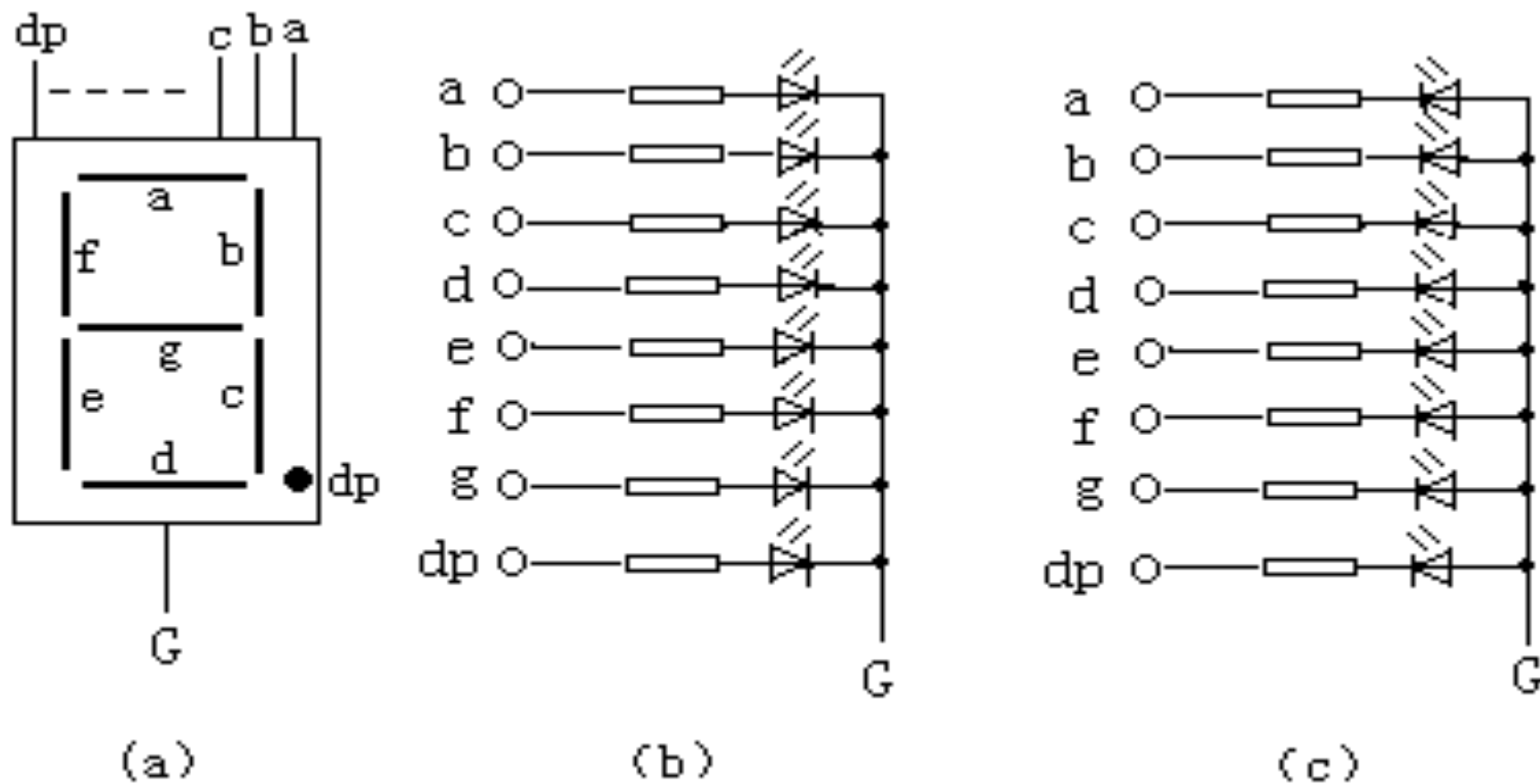
LED显示器程序设计_无条件I/O传送例1

- ❑ LED显示器结构与原理
- ❑ LED显示接口
- ❑ 显示程序设计

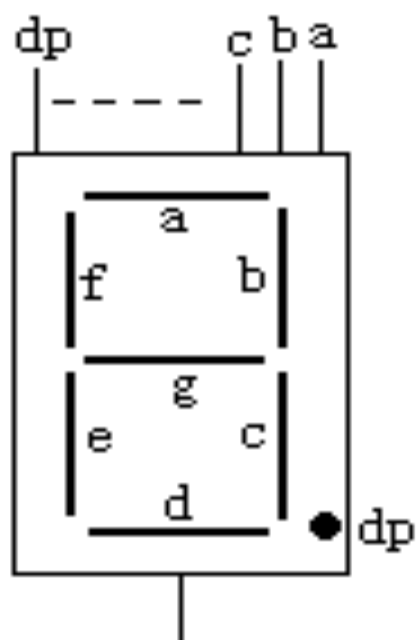


LED显示器结构与原理

LED显示器是由 8 个发光二极管显示字段的显示器件， 通过点亮不同的字段可显示0~9、A~F及小数点等字形。其外形结构与原理见图10-1所示。



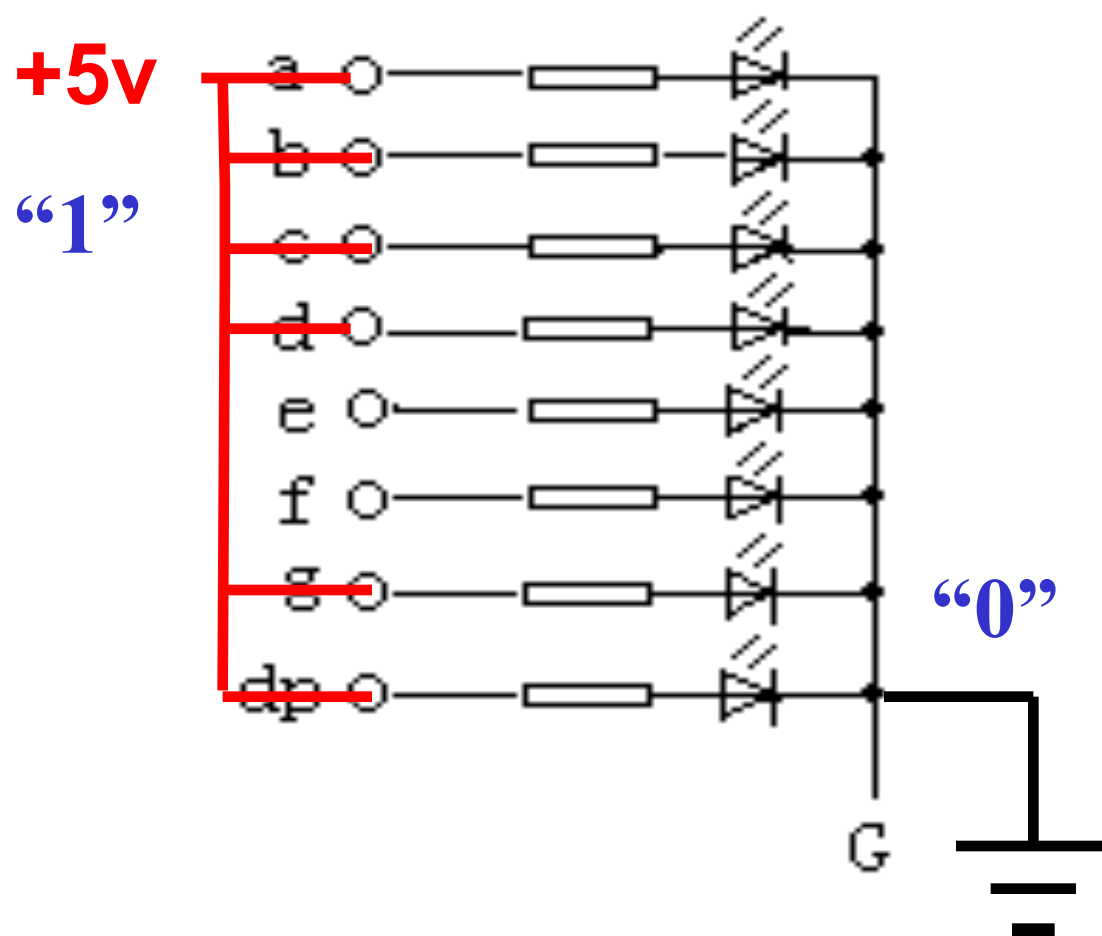
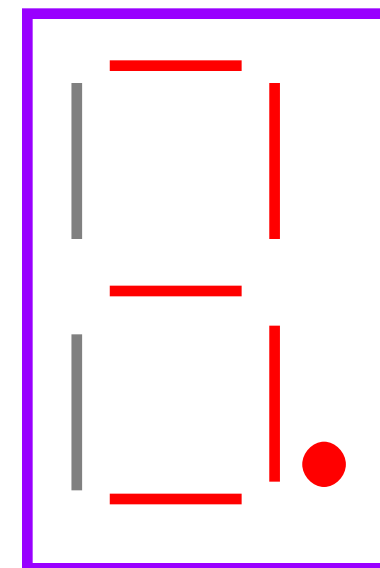
(a) LED管脚图； (b) 共阴极； (c) 共阳极



要使数码管显示指定的字符

对**共阴极**接法的电路：

- 1、要发光的二极管段加高电平
- 2、共阴极端接低电平



dp	g	f	e	d	c	b	a
d7	d6	d5	d4	d3	d2	d1	d0
1	1	0	0	1	1	1	1

0CFH

称0CFH为字形码（段码）

G端称为字位

通常显示字形码顺序排放，存放在固定区域，构成段码表，当要显示某字符时，可根据地址查表。

为使LED显示不同的符号或数字，要为LED提供段码（或称字形码）。

提供给LED显示器的段码正好是一个字节（八位）。

各段与字节中各位对应关系如下：

代码位	D7	D6	D5	D4	D3	D2	D1	D0
显示段	dp	g	f	e	d	c	b	a

8段LED的段码如下表：

字形码表



序号	显示字符	字形码	序号	显示字符	字形码	序号	显示字符	字形码
0	0	C0H	A	A	88H	14	y	91H
1	1	F9H	B	b	83H	15	o	A3H
2	2	A4H	C	C	C6H			
3	3	B0H	D	d	A1H			
4	4	99H	E	E	86H			
5	5	92H	F	F	8EH			
6	6	82H	10		FFH			
7	7	F8H	11	P.	0CH			
8	8	80H	12	H	7FH			
9	9	90H	13	.	BFH			

LED显示接口

LED显示器有静态显示与动态显示两种形式：

1、静态显示接口

常采用MC14495芯片作为LED的静态显示器接口，它是MOTOROLA公司生产CMOS BCD—七段十六进制锁存、译码驱动芯片，可以与LED显示器直接相连。

2. 动态显示接口

常用可编程并行接口芯片8155作为LED动态显示器的接口。通过8155接六只共阴极LED动态显示器的接口。PA口作为字位码口，PB口作为字形（字段）码口。

静态显示原理

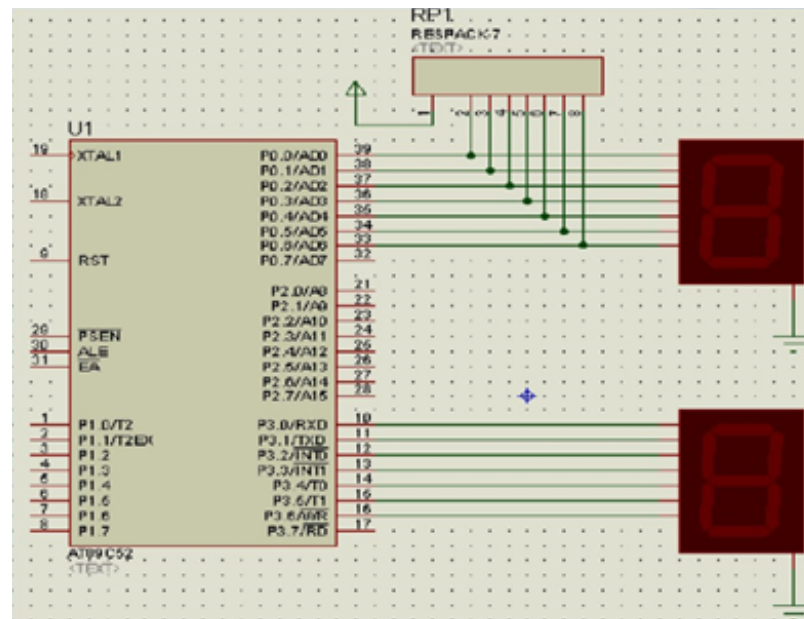
LED显示器各位的共阴极(或共阳极)连接在一起接地(或接+5V);每一位的段码线分别与一个8位并行I/O接口相连。

每位可独立显示, 某一位的段码线上保持段码电平不变, 就能保持相应的显示字符。

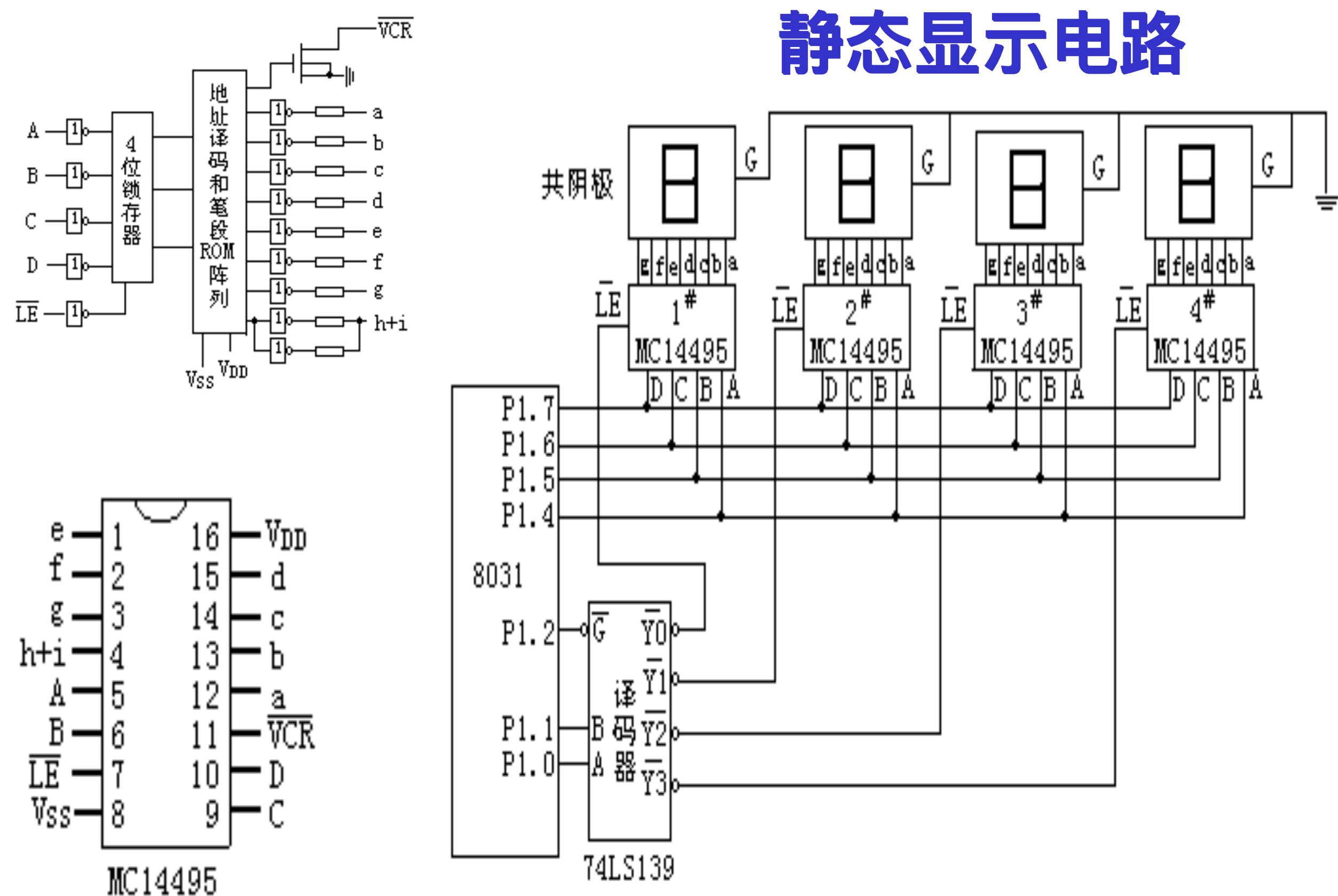
当显示器显示某一字符时，相应段的发光二极管恒定地导通或截止，并且显示器的各位可同时显示。静态显示时，较小的驱动电流就能得到较高的显示亮度。

优点：显示稳定，节省时间；

缺点：位数较多时，占用I/O线较多，硬件费用大。



静态显示电路



MC14495的作用是输入被显示字符的二进制码（或BCD码），并转换成相应的段码，送LED显示。

动态显示原理

所谓动态显示就是一位一位地轮流点亮各位显示器（称为动态扫描），对于每一位显示器来说，每隔一段时间点亮一次。显示器的亮度即与导通电流有关，也与点亮时间和间隔时间的比例有关。

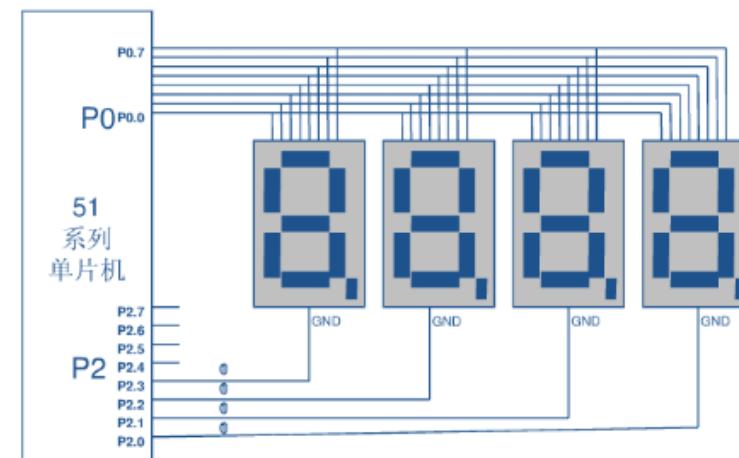
多个数码管显示时，依次循环点亮每个，利用视觉暂留看到整个显示内容。

动态显示原理

将所有LED显示器的段码线（a~g、dp）并联在一起，接一个8位I/O接口，各个显示器的公共端分别由相应的I/O接口线控制，形成分时选通。

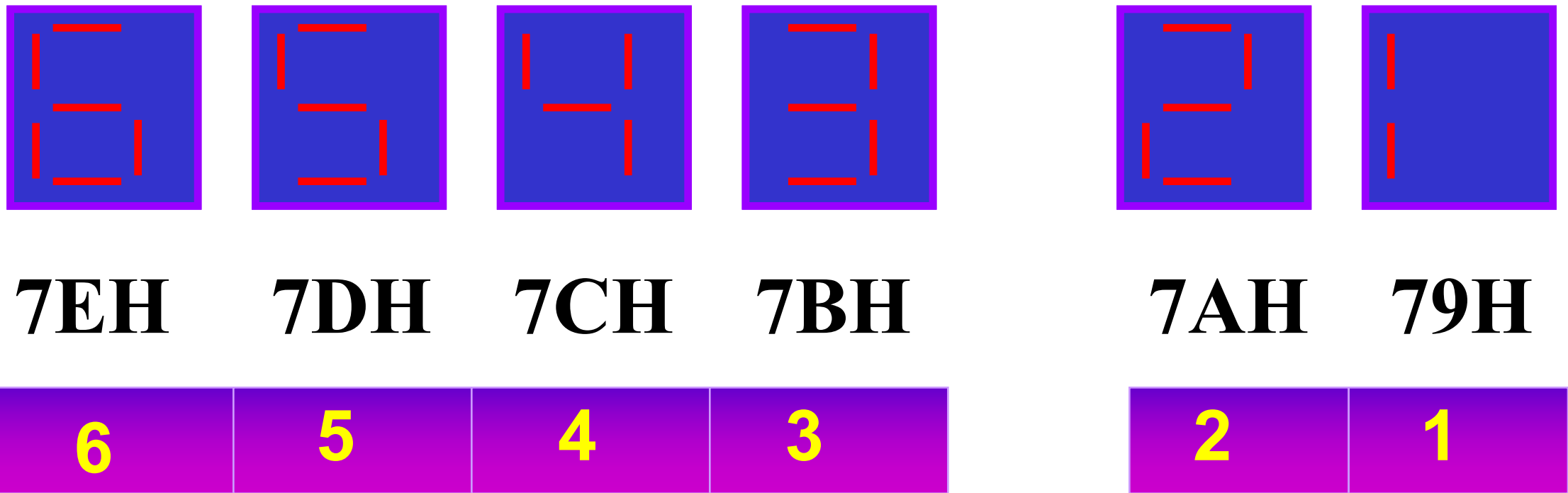
优点：占用I/O资源较少；

缺点：占用大量的单片机时间，降低了CPU的工作效率。（不断循环执行多路扫描显示程序才会有稳定地显示）





显示缓冲区——每个显示块对应一个存储单元，存放该块要显示的**字形码表中的序号或字形码**。



查字形码表——得到字符6的字形码82H——**送字形码口**

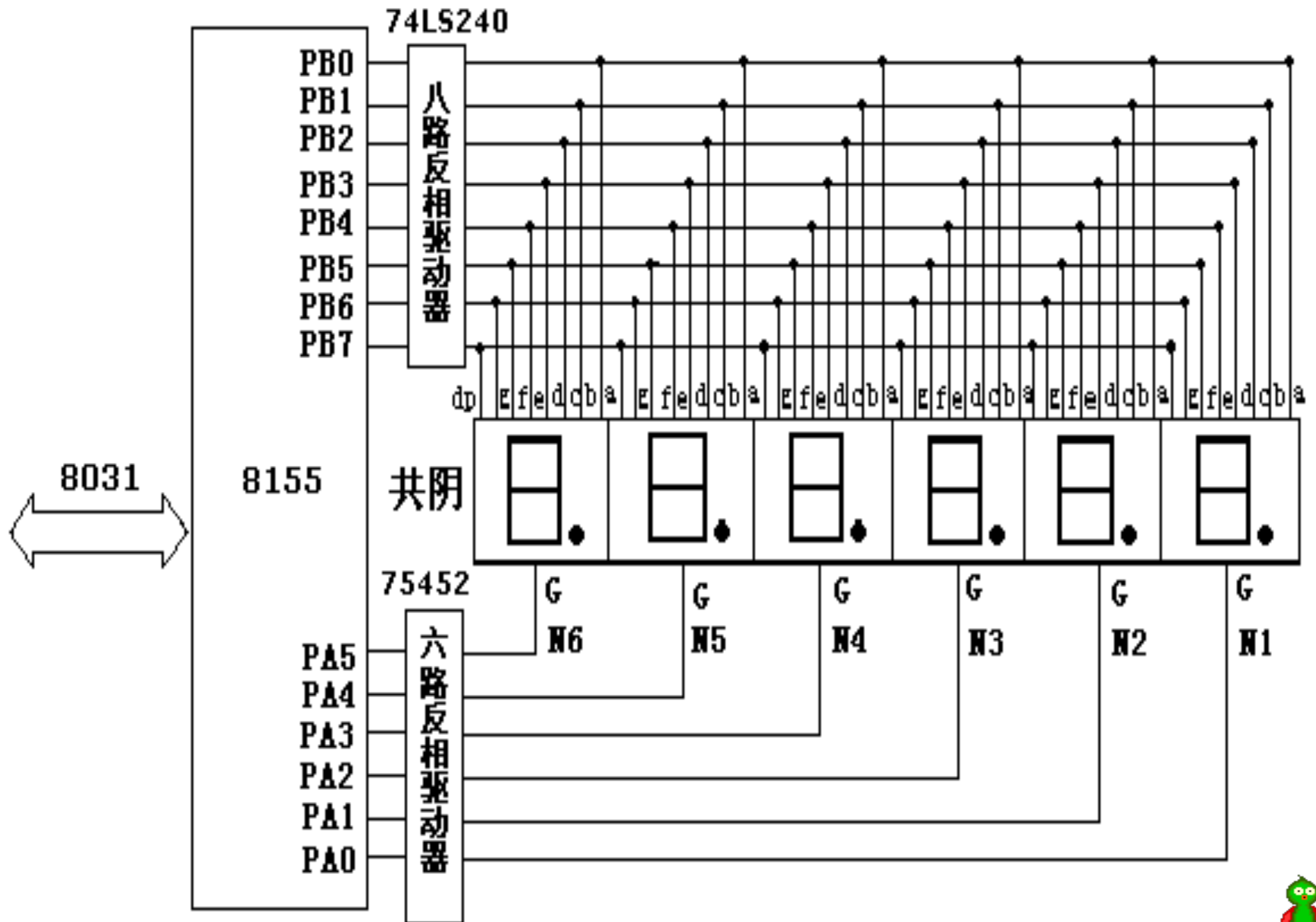
——**将20H送字位码口**——则该位显示6，其它位的显示类同，依次循环往复从低位到高位查字形码表，得到对应的字形码送字形码口，其块的字位码送字位码口，由于视觉残留，**六个不同的字符就同时显示出来。**

字位表

		N6	N5	N4	N3		N2	N1
未	用	20	10	08	04		02	01



8155动态显示电路



5.4 AT89C52的中断系统

AT89C52的中断系统的结构

5. 4. 1 AT89C52的中断源和中断标志

5. 4. 2 AT89C52对中断请求的控制

5. 4. 3 AT89C52对中断的响应和撤除

5. 4. 4 AT89C52中断系统的初始化及应用

中断程序设计举例

