微机原理与接口技术

§7 微控制器IO扩展

主讲人：佘青山

Homepage: https://faculty.hdu.edu.cn/zdhxy/sqs/main.htm

Email: qsshe@hdu.edu.cn

Mob：13758167196

Office: 第二教研楼南楼308室

2024年11月14日

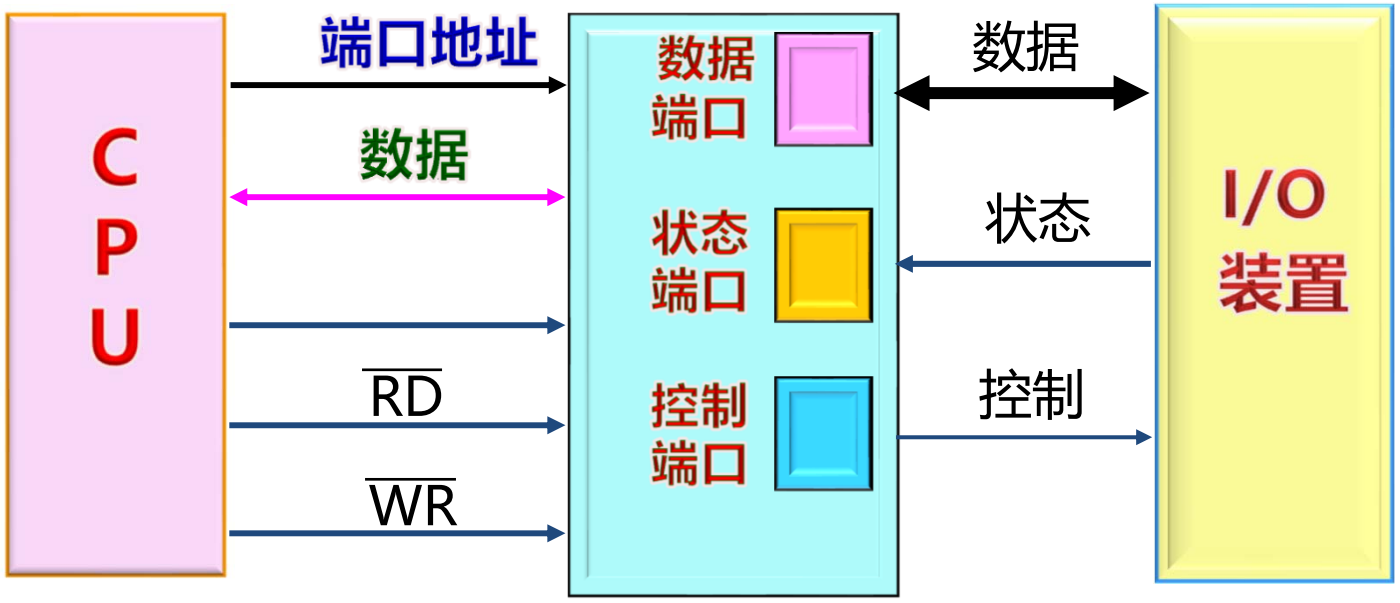
内容提要

1. I/O扩展概述
2. 扩展相关技术
3. I/O控制方式
4. I/O简单扩展
5. I/O扩展实例

基本概念

CPU的外部设备种类繁多，在电平，功率，速度，信息形式上与CPU有很大的差别。两者不能简单的连接。需要通过输入输出(I/O)接口传递信息。

I/O接口作为CPU与外设间的桥梁，完成CPU与外设的信息交换。信息可以分为状态、数据和控制信息。



7.1.1 为何要进行I/O扩展

CPU与外部设备交换信息的过程：在控制信号的作用下通过数据总线来完成的。外部设备种类繁多，它们对所传输的信息的要求也各不相同，计算机和外设之间的信息交换存在如下问题：

1.微控制器本身接口功能有限

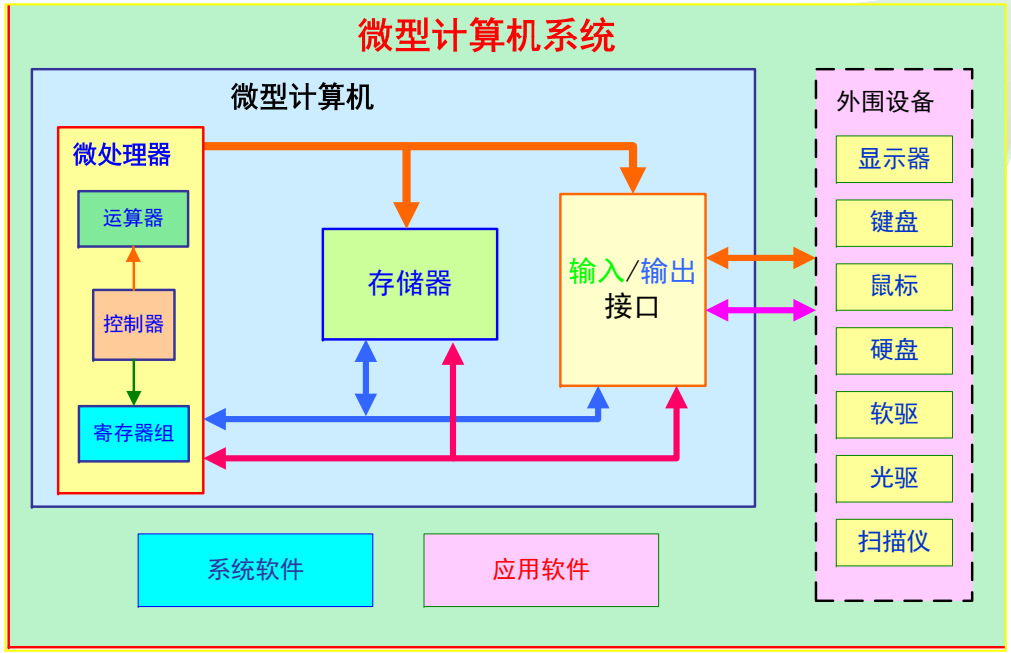
2. 微控制器控制应用中的复杂接口，并体现在以下几个方面：

（1）速度差异大

（2）设备种类繁多

（3）数据信号形式多种多样

- 信号电平不匹配
- 信号格式不匹配
- 时序不匹配



7.1.2 扩展I/O接口电路的功能

1）增加I/O引脚数量

MCS-51单片机的只有4个I/O口，并要用于扩展的数据和地址和第2功能，引脚不够。

2）速度协调

单片机的速度很高，外设的速度低，并且不同的外设速度差异甚大。

3）数据输出锁存

单片机的速度很高，数据在总线上时间很短，以至于外设来不及接收，信号就要消失，为此输出数据需要将数据先锁存起来

7.1.2 扩展I/O接口电路的功能

4）数据总线隔离

数据总线上连接多个数据源（输入设备）和多个负载（输出设备）。但同一时刻只能有一个原或负载与单片机之间进行数据传送，而其它不参与的设备在电性能上与总线隔开，这就是总线隔离功能。

为了实现总线隔离，接口电路需要提供三态缓冲电路，所谓的三态：低电平、高电平和高阻状态。

5）数据转换

外部设备种类繁多，不同设备之间的性能差异很大，信号形式也多种多样。有电压、电流、数字和模拟的，而单片机只有数字信号，如果外部设备需要的不是数字信号，就需要模数/转换或数/模转换电路。

6）增强驱动能力

通过接口电路可以为输出数据提供足够的驱动能力

内容提要

1. I/O扩展概述
2. 扩展相关技术
3. I/O控制方式
4. I/O简单扩展
5. I/O扩展实例

7.2.1 接口与端口

接口是指计算机与设备之间在数据传送方面的联系电路，负责两种之间的联系。

接口电路通常包括：

- **数据寄存器**：保存输入/输出**数据**
- **状态寄存器**：保存外界的**状态信息**
- **命令寄存器**：保存来自微控制器的**控制命令**等

7.2.1 接口与端口

把接口电路中这些可编址、并能进行读或写操作的寄存器称之为端口（Port），或简称口。

一个接口电路中可能包括有多个口，例如数据口、状态口和命令口等，因此一个接口电路就对应（或包含）着多个口地址。但是，不是每个接口电路都具备上述功能。

7.2.2 数据总线隔离技术

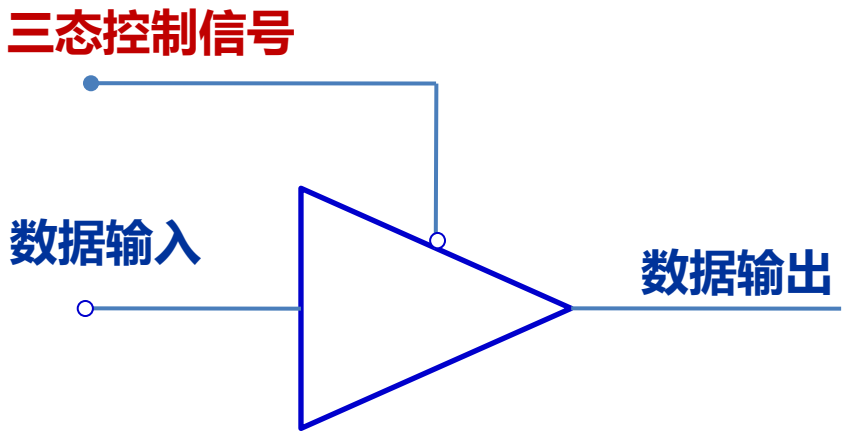
微控制器的I/O操作中，输入/输出的数据都要通过系统的数据总线进行传送，为了正确地进行数据的I/O传送，就必须解决数据总线的隔离问题。

总线隔离：数据传输设备在需要的时候能与数据总线接通，而在不需要的时候又能同数据总线隔开。

对于输出设备的接口电路，要提供锁存器，当允许接收输出数据时闩锁打开，当不允许接收输出数据时闩锁关闭。

而对于输入设备的接口电路，要使用三态缓冲电路。

7.2.2 数据总线隔离技术



三态缓冲控制逻辑

三态控制信号	工作状态	数据输入	输出端状态
1	高阻抗	0	高阻抗
		1	高阻抗
0	驱动	0	0
		1	1

单选题 1分

设置

下列功能中不是由I/O接口实现的是（）

A 速度协调

B 数据缓冲和锁存

C 信号（数据）转换

D 数据暂存

提交

多选题 4分

设置

接口电路通常包括如此哪些端口?

A 地址端口

B 数据端口

C 控制端口

D 状态端口

提交

7.2.3 微控制器IO编址技术

在计算机中，凡需进行读写操作的设备都存在着编址的问题。具体说在微控制器中有两个需要编址的子系统，一个是存储器，另一个就是接口电路。

存储器是对存储单元进行编址，而接口电路则是对其中的寄存器（口）进行编址。

7.2.3 微控制器IO编址技术

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编址共有两种方式：独立编址方式和统一编址方式。

1）独立编址方式

把I/O和存储器分开进行编址，亦即各编各的地址，这样在一个微控制器系统中就形成了两个独立的地址空间：存储器地址空间和I/O地址空间，

从而使存储器读写操作和I/O操作是针对两个不同存储空间的数据操作

7.2.3 微控制器IO编址技术

在独立编址方式的计算机指令系统中，除存储器读写指令之外，还有专门的I/O指令以进行数据输入/输出操作。此外，在硬件方面还需要定义一些专用信号，以便对存储器访问和I/O操作进行硬件控制。

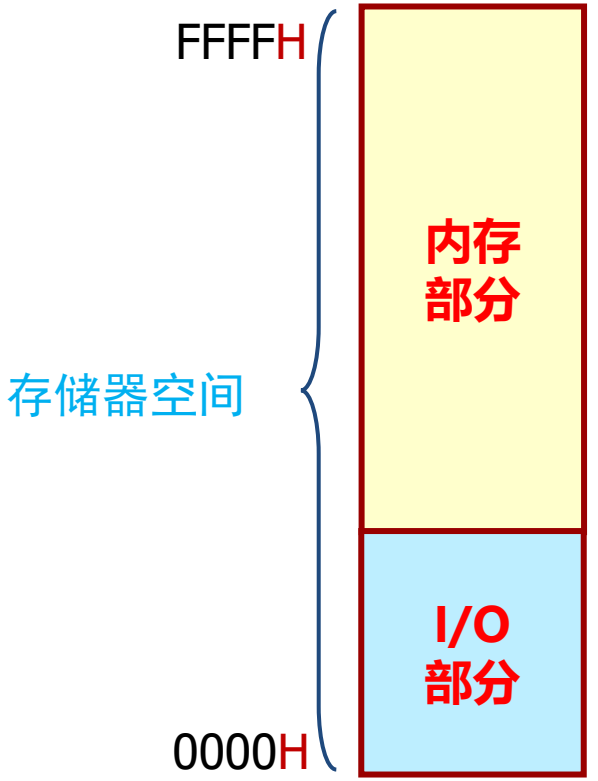
例如，8086CPU就有专门的输入和输出指令（IN指令和OUT指令）

独立编址的优点：I/O地址空间和存储器地址空间相互独立，界限分明。但为此却要在计算机中专门设置一套I/O指令和控制信号，从而增加了系统的开销。

2）统一编址方式

指I/O和存储器的统一，在这种编址方式中，把I/O接口中的寄存器（口）与存储器中的存储单元同等对待，统一进行编址。

为此也把这种编址称之为存储器映像编址方式。采用这种编址方式的计算机只有一个统一的地址空间，该地址空间既供存储器编址使用，也供I/O编址使用



7.2.3 微控制器IO编址技术

MCS-51微控制器使用统一编址方式，因此接口电路中的I/O地址与存储单元的地址长度相同。

统一编址方式的优点是不需要专门的I/O指令，而直接使用存储器指令进行I/O操作，不但简单、方便、功能强，而且I/O地址范围大。但这种编址方式，16位的口地址太长，会使地址译码变得复杂。此外，存储器指令比起专用的I/O指令来，指令长且执行速度慢。

内容提要

1. I/O扩展概述
2. 扩展相关技术
3. I/O控制方式
4. I/O简单扩展
5. I/O扩展实例

CPU与外设间的数据传送方式

在微型机系统中，可采用的输入输出方式主要有:

- 程序控制方式
  - 无条件传送方式
  - 条件传送方式（查询式传送方式）
- 中断方式
- 直接存储器存取(Direct Memory Access, DMA)方式

前两种方式主要由软件实现，DMA方式主要由硬件实现。

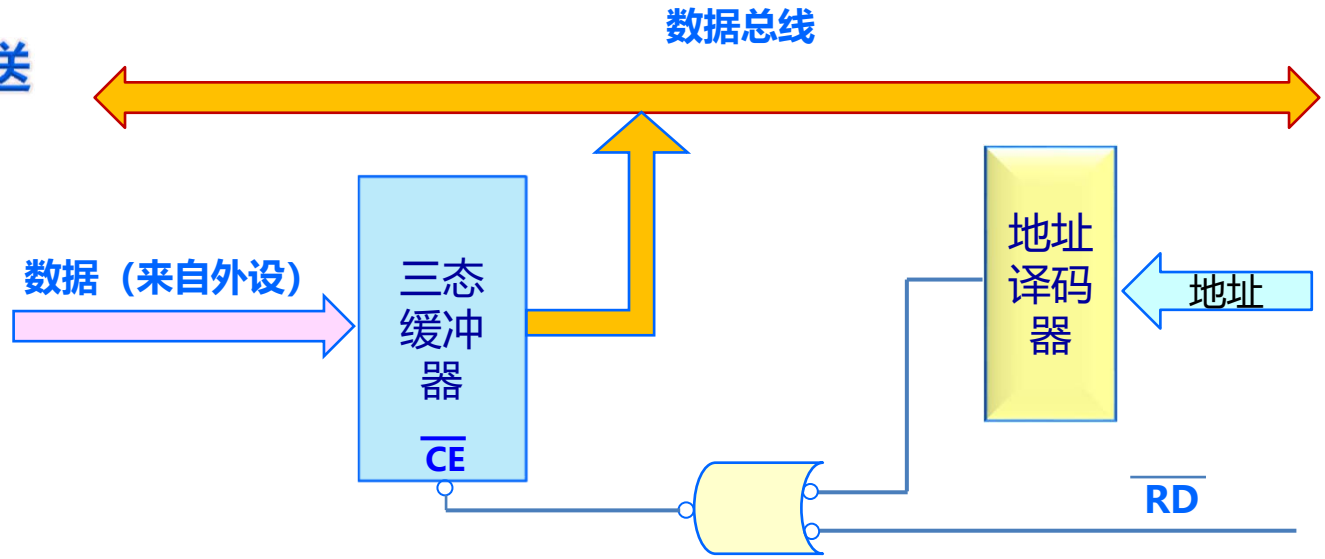
7.3.1 无条件传送方式

无条件传送方式也称为同步程序传送。只有那些能一直为数据I/O传送作好准备的设备，才能使用无条件传送方式。

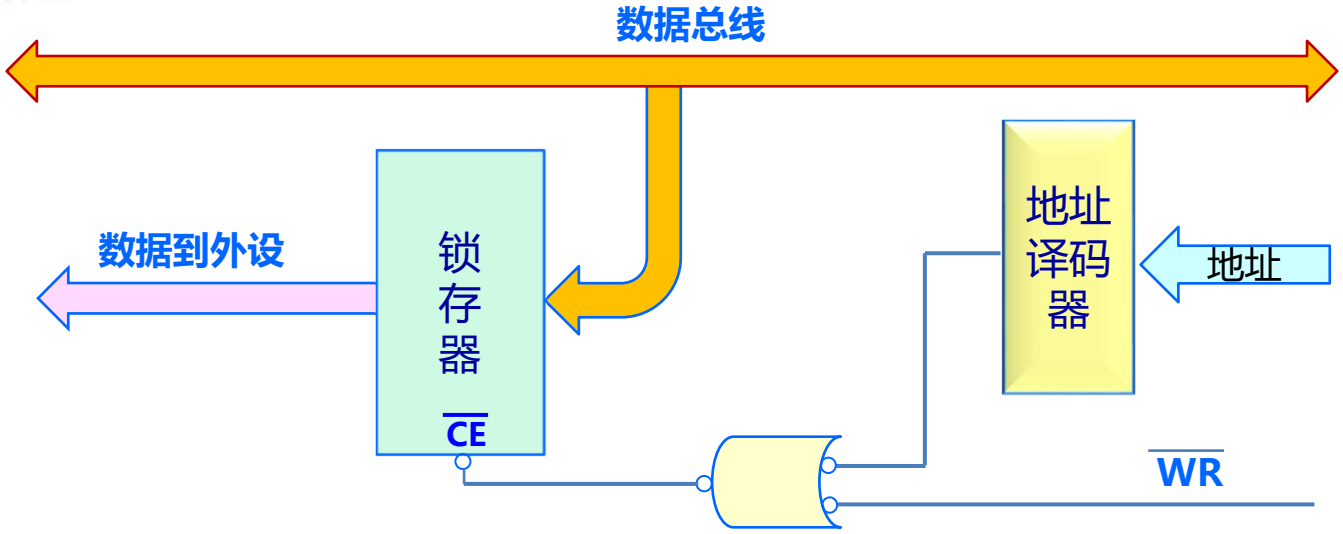
因为在进行I/O操作时，不需要测试设备的状态，可以根据需要随时进行数据传送操作。

7.3.1 无条件传送方式

(a) 无条件输入传送



(b) 无条件输出传送



7.3.1 无条件传送方式

无条件传送适用于以下两类设备的数据输入/输出：

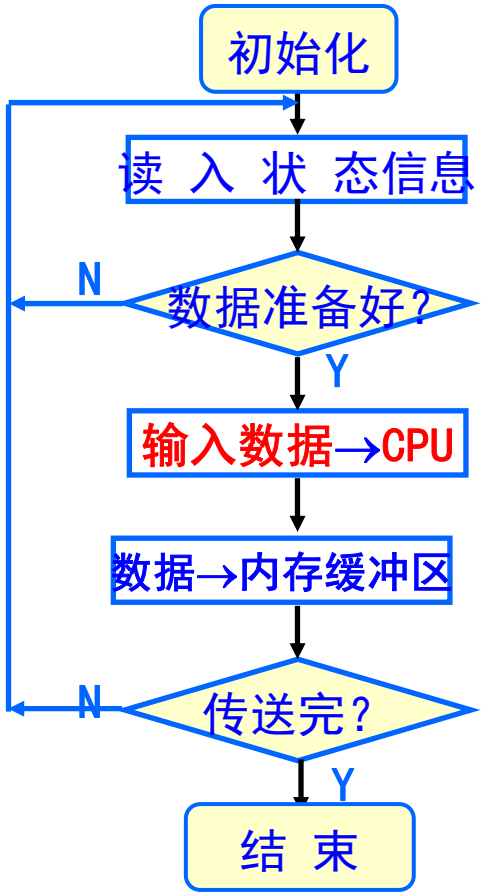
- 具有**常驻的或变化缓慢的数据信号的设备**。例如：机械开关、指示灯、发光二极管、数码管等。可以认为它们随时为数据输入/输出处于“准备好”状态。

- **工作速度非常快**，足以和微控制器同步工作的设备。例如数/模转换器（DAC），由于它是并行工作的，速度很快，因此微控制器可以随时向其传送数据，进行数/模转换。

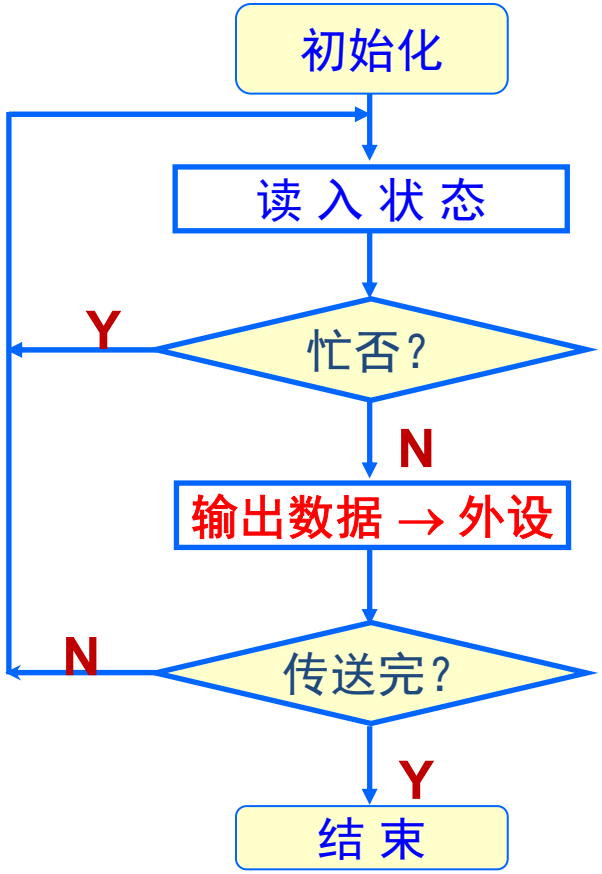
7.3.2 程序查询方式

只有在确认设备已“准备好”的情况下，MCU才能执行数据输入/输出操作。

需要由接口电路**提供设备状态**，并以软件方法进行状态测试(**软硬件方法结合的数据传送方式**)。



查询式输入流程



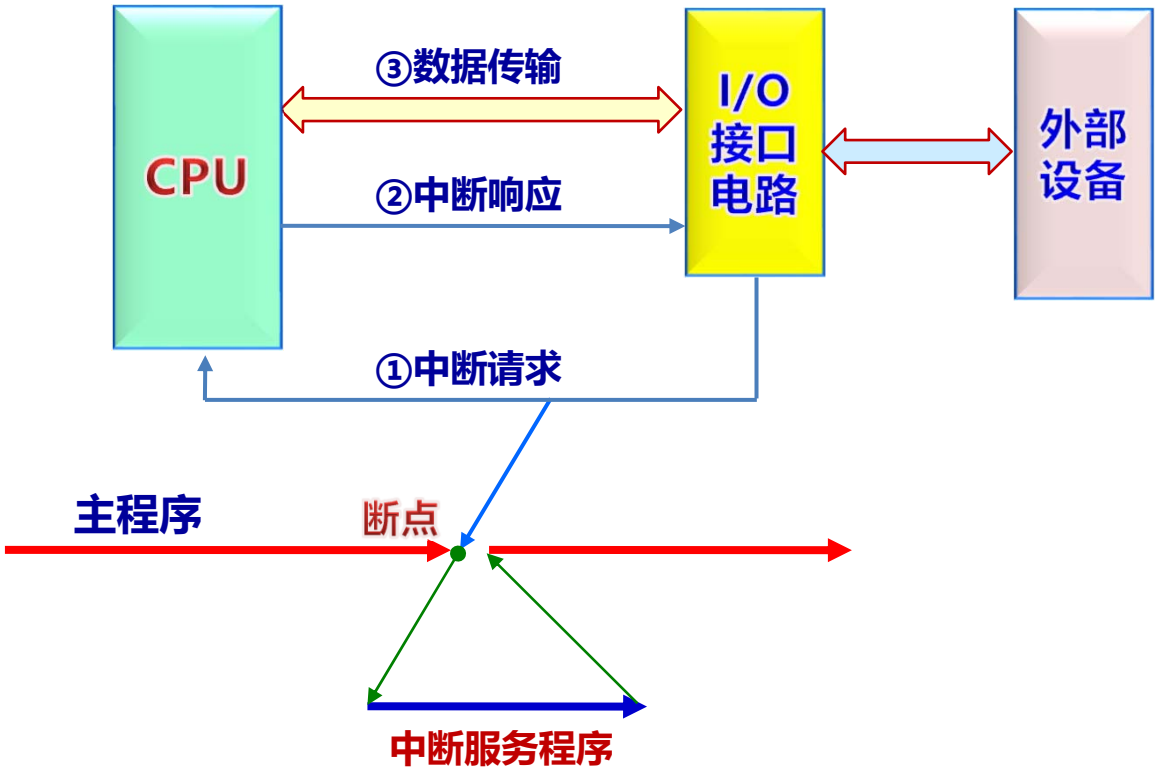
查询式输出流程

7.3.2 程序查询方式

程序查询方式，电路简单，查询软件也不复杂，而且通用性强，因此适用于各种设备的数据输入/输出传送。但是查询过程对微控制器来说毕竟是一个无用的开销，因此查询方式只能适用于单道作业、规模比较小的微控制器系统。

7.3.3 中断方式

中断方式又称**程序中断方式**，它与查询方式的主要区别在于**如何知道设备是否为数据传送作好了准备**，查询方式是微控制器的**主动形式**，而中断方式则是**被动形式**。微控制器等待通知（中断请求）。



7.3.3 中断方式

中断这种并行工作方式大大提高了微控制器系统的效率，所以在微控制器中被广泛采用。但中断请求是一种随机事件，为实现程序中断，对微控制器系统的硬件和软件都有较高的要求。

此外，由于在中断处理时常需现场保护和现场恢复，这对微控制器应用来说仍是一项较大的无用开销。

单选题 1分

设置

在MCS-51微控制器系统中，采用的统一编址方式

A 对

B 错

提交

单选题 1分

设置

数据输入/输出控制方式中，效率较高的是

A 无条件传送

B 有条件传送

C 查询

D 中断

提交

内容提要

1. I/O扩展概述
2. 扩展相关技术
3. I/O控制方式
4. I/O简单扩展
5. I/O扩展实例

7.4.1 IO简单扩展方法

扩展常用三种方法：

1、利用TTL、CMOS集成电路来扩展

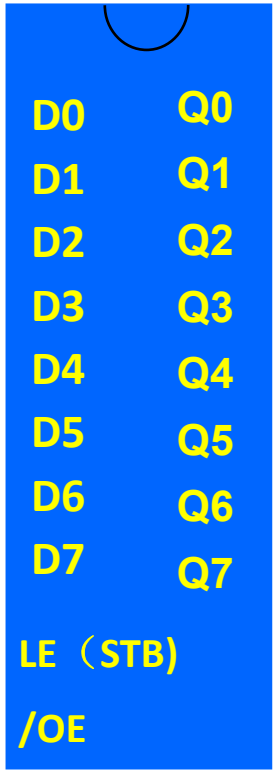
2、利用单片机串口扩展

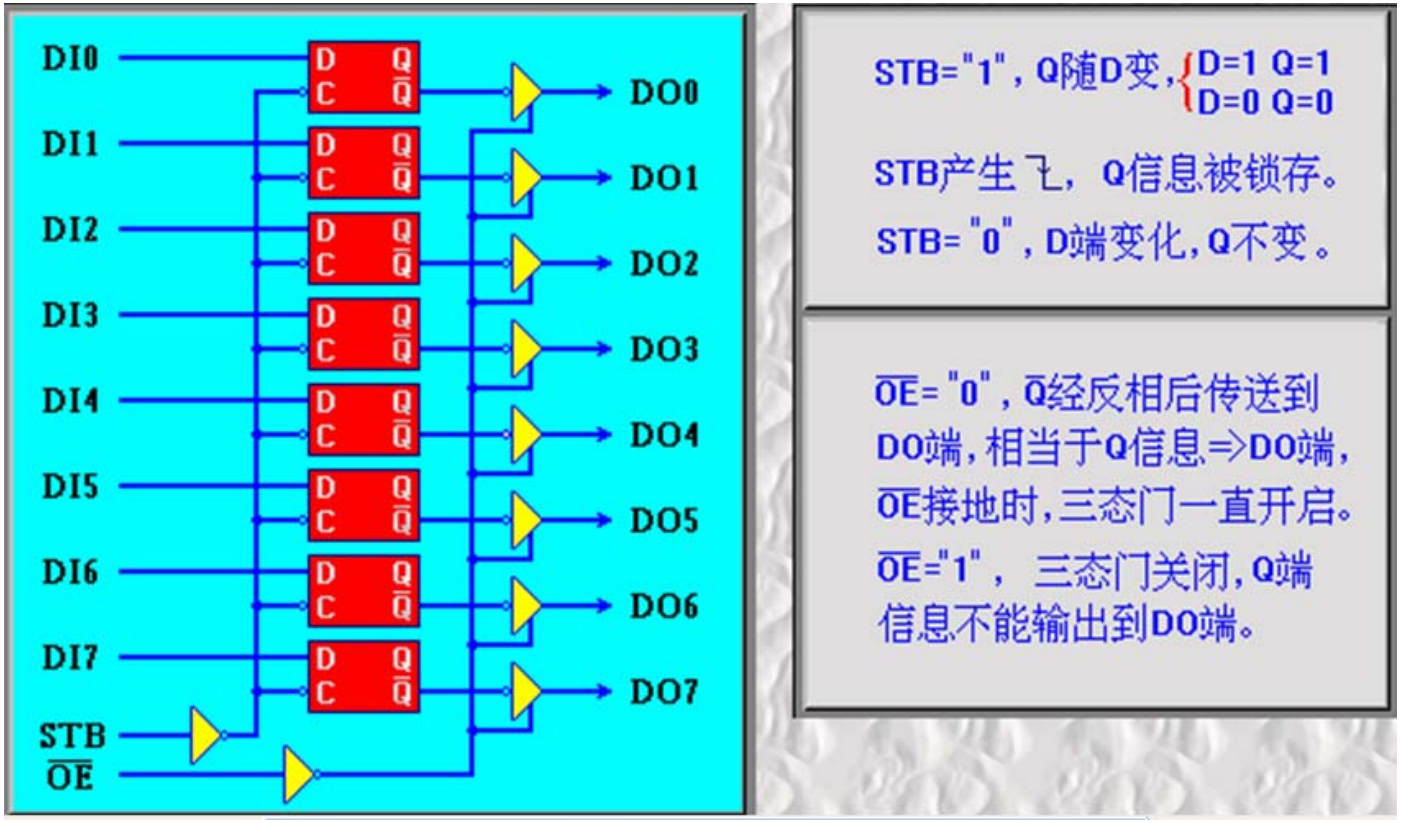
3、利用可编程并行接口芯片来扩展

这里主要介绍，简单I/O扩展，即利用TTL、CMOS集成电路中的锁存器或三态门芯片来扩展

7.4.1 IO简单扩展方法

74LS573锁存器

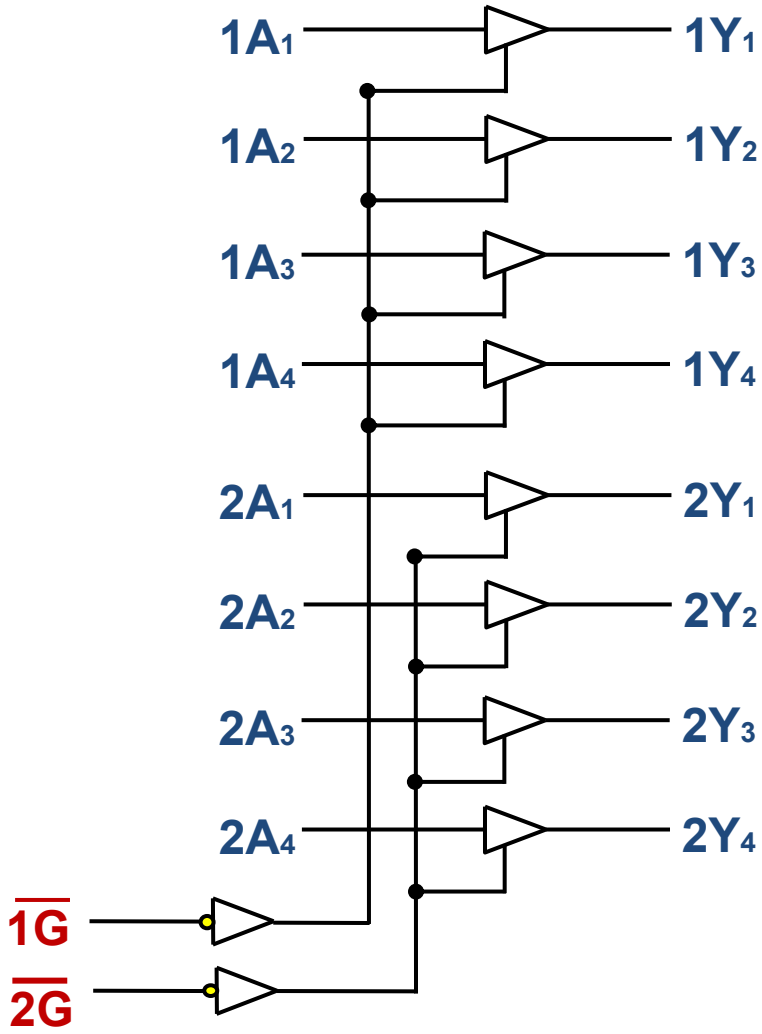


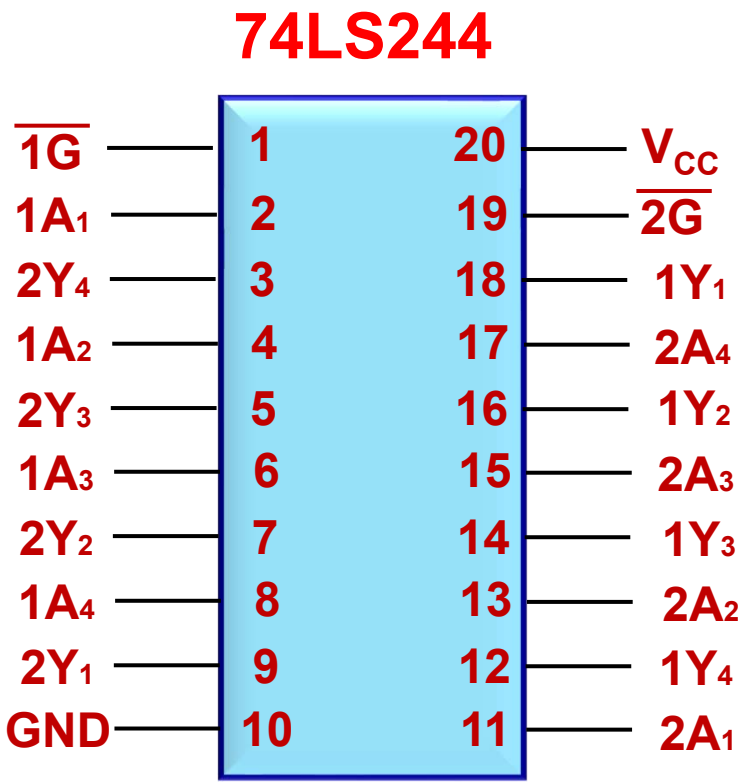


74LS573 既可以作为数据输出锁存，也可以作为数据输入缓冲器使用。只是两种使用过场合接线方法有所不同。

7.4.1 IO简单扩展方法

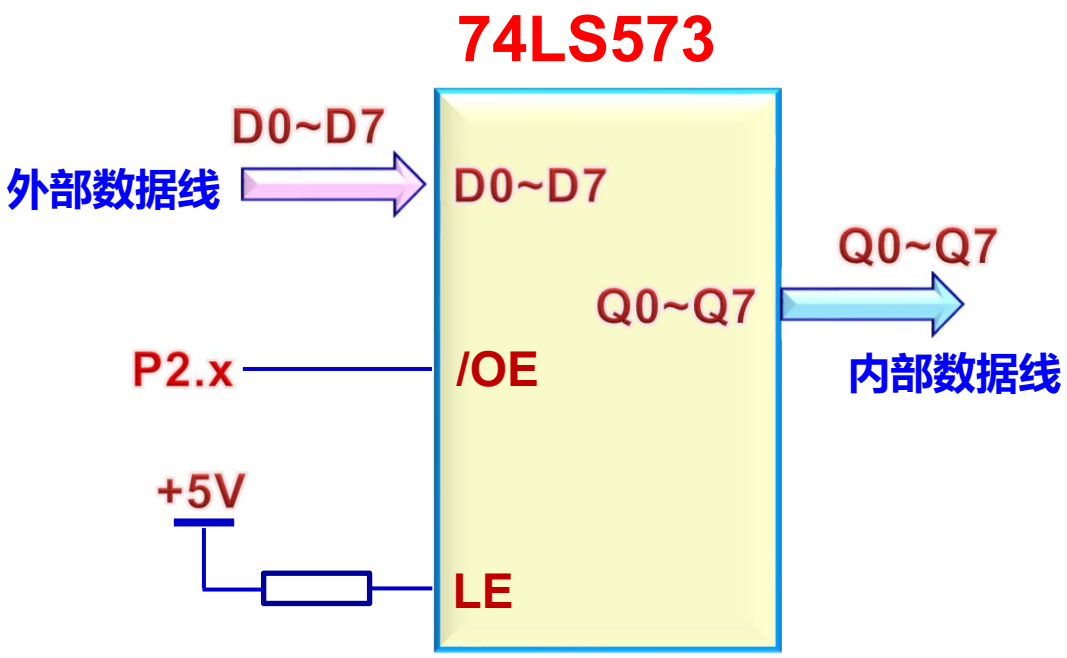
缓冲器74LS244芯片 (8路数据缓冲器)





7.4.2 简单输入接口扩展

1）输入接口直接控制法



74LS573作为输入缓冲器的I/O口直接控制方法

74LS573作为数据输入缓冲器时

- 锁存器使能端LE直接+5V，这样就可以确保内部触发器与外部数据保持随动变化
- 三态输出使能端/OE接控制线进行操作，需要时才打开。

例如：控制脚为P2.7，则：

```
CLR   p2.7   ;低电平时有效时，外部数据输入到内部数据线上
```

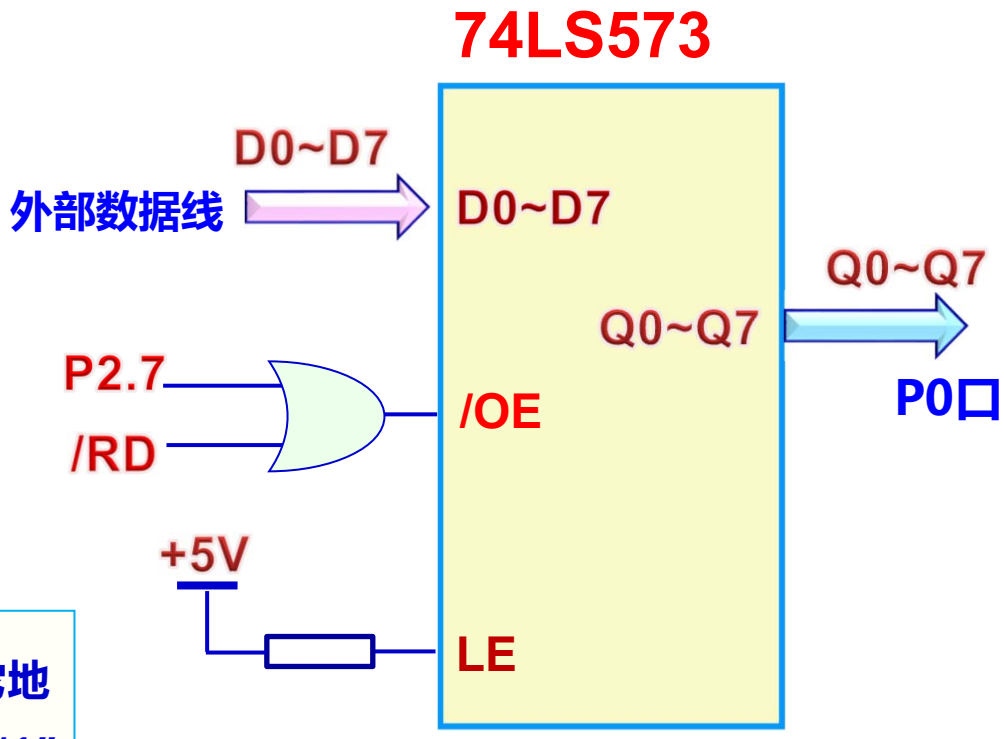
7.4.2 简单输入接口扩展

2）输入接口总线控制法

三态输出使能端/OE由微控制器P2口高位引脚与读引脚/RD联合实现控制。

例如：控制脚为P2.7，则通常需要如下指令：

```
MOV   DPTR,  #7FFFH ;假定其它地址线为“1”
MOVX  A,  @DPTR
```

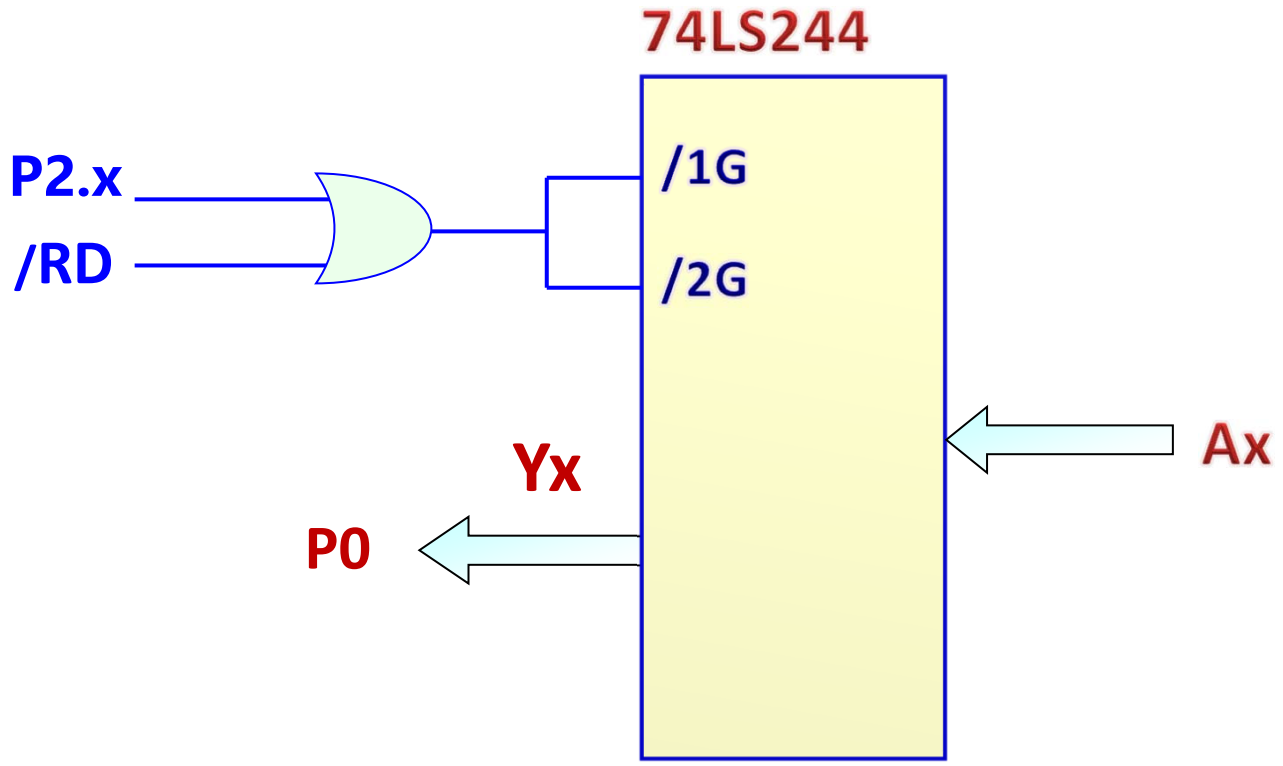


74LS573作为输入缓冲器的I/O口总线控制方法

该操作方法实际上就是利用P2口构成的I/O端口地址（统一编址方式线选法），并利用MOVX A，@DPTR 外部RAM 寻址方法直接从端口上读入数据。

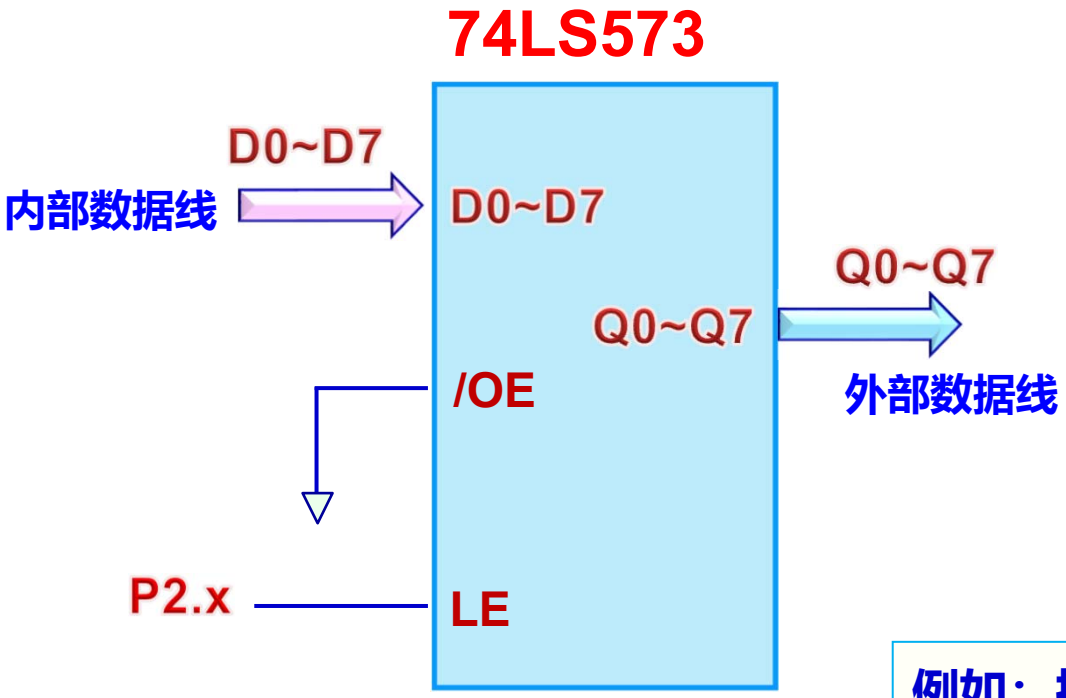
7.4.2 简单输入接口扩展

74LS244也可以作为I/O端口输入扩展



7.4.3 简单输出接口扩展

1）输出接口直接控制法



74LS573作为输出锁存器的I/O口直接控制方法

74LS573作为数据输出锁存时

- 三态输出使能端/OE直接接地，确保锁存器一直保持输出状态。
- 三态锁存端LE由某个引脚P2.x进行控制。

例如：控制脚为P2.7，则需要命令：

```
SETB   p2.7    ;高电平时有效时，外部数据输入到内部数据线上
```

7.4.3 简单输出接口扩展

需要注意：

采用上述I/O直接控制时，在P口控制LE引脚时需要一个脉冲信号。

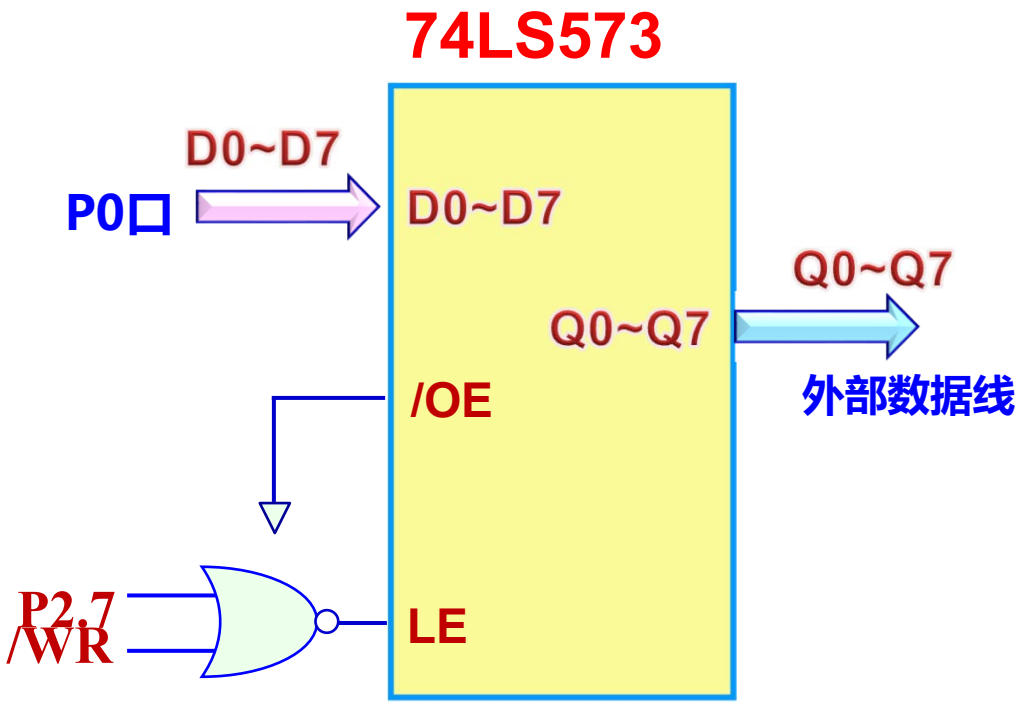
例如： 控制脚为**P2.7**，而由**P1口作为内部输入**，则通常需要**如下指令**：

```
MOV  P1, A
SETB  P2.7
NOP
CLR  P2.7
```

7.4.3 简单输出接口扩展

2）输出接口总线控制法

- 三态输出使能端/OE直接接地
- 锁存器LE由微控制器P2高位引脚与写控制引脚/WR联合实现控制。
- 例如：控制脚为P2.7，则通常需要如下指令：



74LS573作为输出锁存器的I/O口总线控制方法

```
MOV   DPTR,  #7FFFH  ;假定其它地址线为“1”
MOVX  @DPTR , A
```

该操作方法实际上就是利用P2 口构成的I/O 端口地址（统一编址方式线选法），并利用MOVX @DPTR，A将数据送到74LS573 进行锁存并保持输出。

内容提要

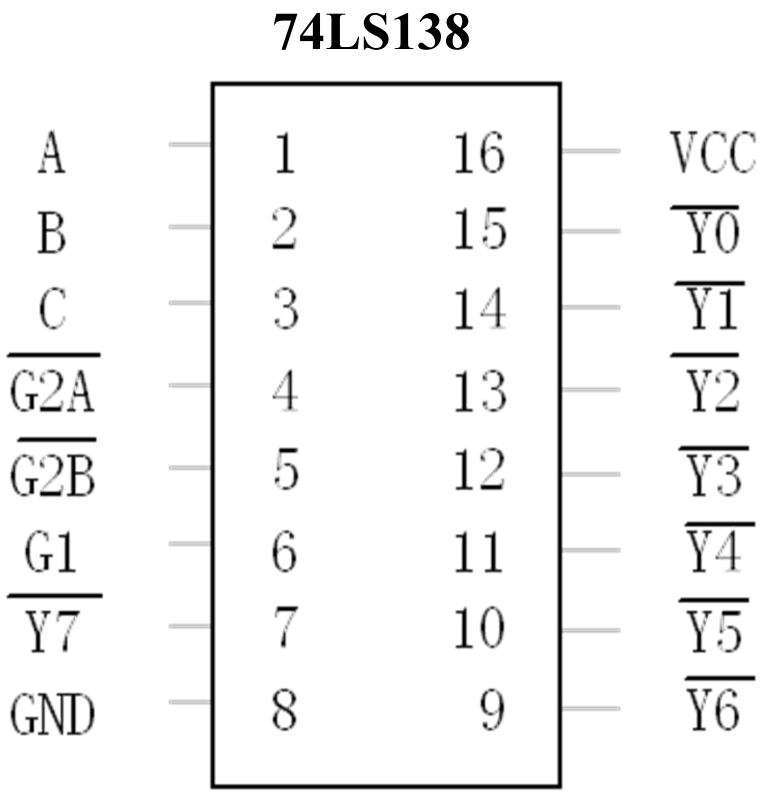
1. I/O扩展概述
2. 扩展相关技术
3. I/O控制方式
4. I/O简单扩展
5. I/O扩展实例

7.5 扩展实例

实例8

1. 利用AT89C51微控制器的P0口作开关量输入口，P2.4-P2.6接74LS138译码器的A~C端，P2.7接74LS138的门控端G1 (E1)，P0口作开关量输出口；
2. 当P0.x端开关闭合时，对应的P0.x口的LED发光二极管熄灭；当P0.x端开关断开时时，对应的P0.x口的LED发光二极管点亮，x=0,1,…,7。
3. 数据来自P0的扩展口，由74LS138译码器的Y0、Y1输出端和读、写引脚控制74LS373芯片的使能。

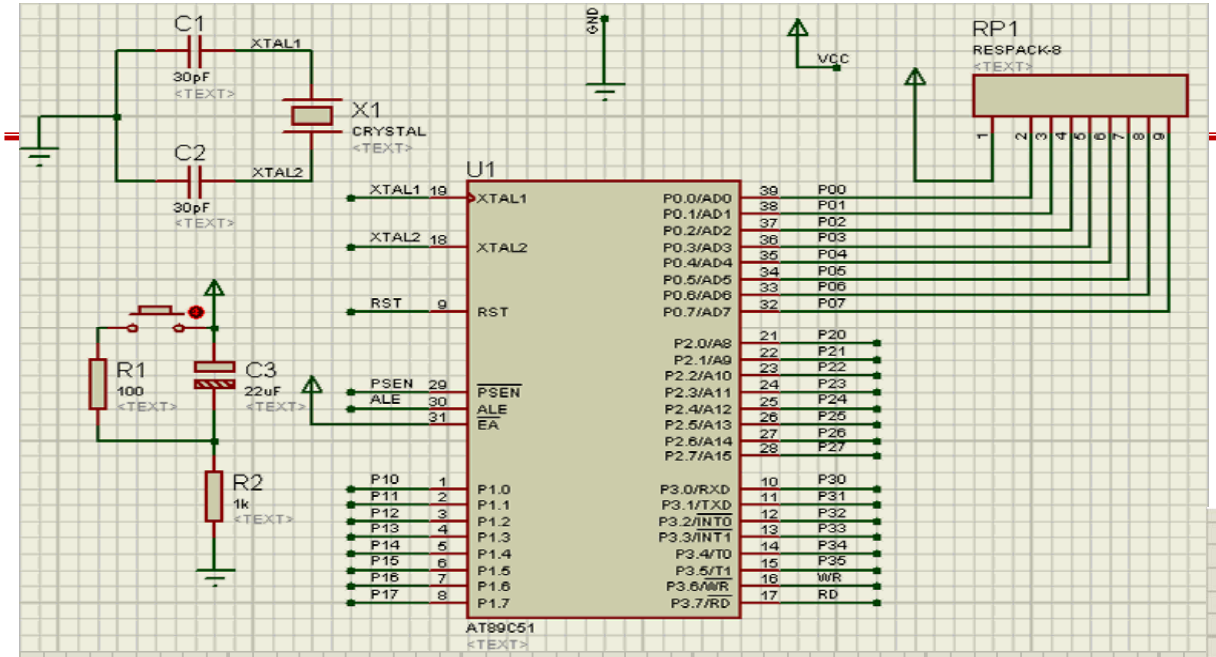
7.5 扩展实例

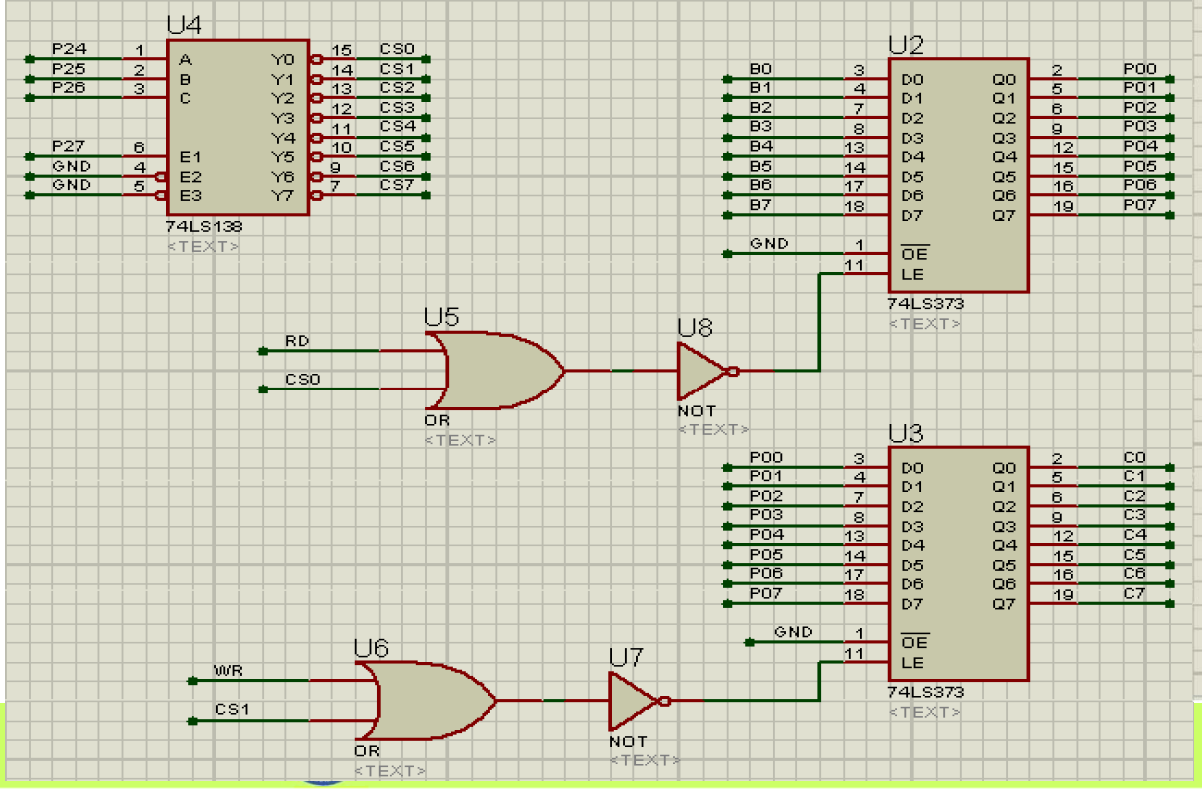


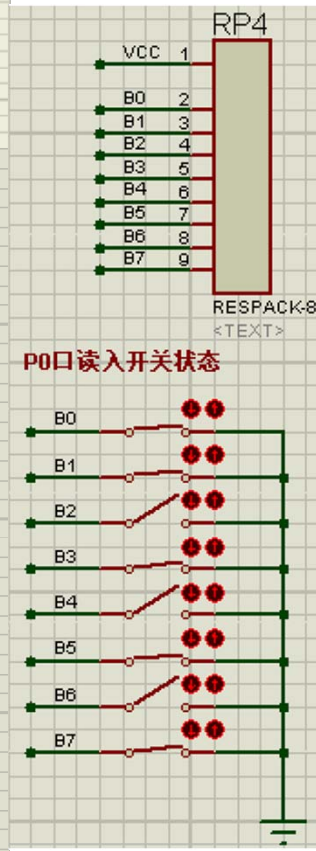
G_1	$\overline{G_{2A}}$	$\overline{G_{2B}}$	C	B	A	输出
1	0	0	0	0	0	$\overline{Y}_0$=0 其余为1
1	0	0	0	0	1	$\overline{Y}_1$=0 其余为1
1	0	0	0	1	0	$\overline{Y}_2$=0 其余为1
1	0	0	0	1	1	$\overline{Y}_3$=0 其余为1
1	0	0	1	0	0	$\overline{Y}_4$=0 其余为1
1	0	0	1	0	1	$\overline{Y}_5$=0 其余为1
1	0	0	1	1	0	$\overline{Y}_6$=0 其余为1
1	0	0	1	1	1	$\overline{Y}_7$=0 其余为1

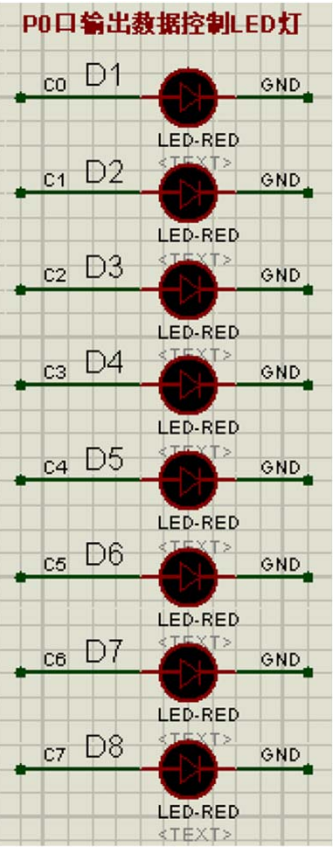
该操作方法利用P2 口构成的I/O 端口地址，并利用MOVX指令将数据送到74LS373 进行锁存并保持输出。

I/O 端口地址是多少？









7.5 扩展实例

程序：

```
        ORG    0000H
        SJMP   MAIN

        ORG    0030H
MAIN:   MOV    SP, #5FH
LOOP:   MOV    DPTR, #8FFFH    ；P0口读入开关状态。假定其他地址线为1
                               ；注：利用P2口构成的输入口地址可以是8000H~8FFFH地址中的任何一个
        MOVX   A, @DPTR
        NOP
        NOP
        MOV    DPTR, #9FFFH    ；P0口送出数据，控制LED灯。假定其他地址线为1
                               ；注：利用P2口构成的输出口地址可以是9000H~9FFFH地址中的任何一个
        MOVX   @DPTR, A
        SJMP   LOOP

        SJMP   $
        END
```

THE END

THANK YOU