



第7章 输入/输出技术

7.1 I/O接口概述

7.2 I/O端口

7.3 CPU与外设间的数据传送方式



7.1 I/O接口概述

一、I/O接口及接口技术的概念

接口：指**CPU**和存储器、外部设备或者两种外部设备之间，或者两种机器之间通过系统总线进行连接的逻辑部件（或电路），是**CPU**与外界进行信息交换的中转站。

接口技术：是专门研究**CPU**与外部设备之间的数据传送方式、接口电路工作原理和使用方法的一门技术，其采用硬件与软件相结合的方法，研究**CPU**如何与外部设备进行最佳耦合，以便在**CPU**与外部设备之间实现高效、可靠的信息交换。



7.1 I/O接口概述

CPU与外部设备之间交换的信息类型:

- **数据信息** 是**CPU**与外部设备之间交换最多的一类信息，微机中的数据通常为**8位**、**16位**或**32位**。
- **状态信息** 反映当前外部设备的工作状态，是**CPU**与外部设备之间进行信息交换时的联络信号。
- **控制信息** 是**CPU**对外部设备发出的控制命令，以设置外部设备的工作方式等。

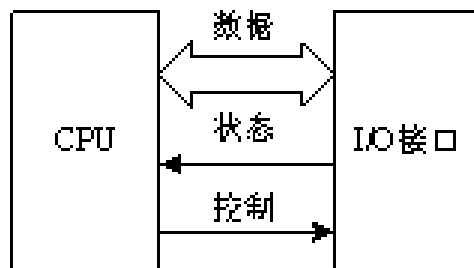


图7.1 CPU与外设之间传送的信息



7.1 I/O接口概述

二、I/O接口的主要功能

(1) 地址译码或设备选择功能

微机系统中通常会有多个外部设备同时与主机相连，**CPU**同一时刻只能与一个外部设备传送数据。**I/O**接口电路通过地址译码选择相应设备，只有被选中的设备才能与**CPU**进行数据交换或通信。

(2) 数据缓冲功能

解决**CPU**与外部设备间的速度不匹配等问题。接口电路中一般都设有数据寄存器或锁存器来缓冲数据信息，同时还提供“准备好”、“忙”、“闲”等状态信号。



7.1 I/O接口概述

(3) 输入/输出功能

外部设备通过**I/O**接口电路实现与**CPU**之间的信息交换，**CPU**通过向**I/O**接口写入命令控制其工作方式，通过读入命令可以随时监测、管理**I/O**接口和外部设备的工作状态。

(4) 信息转换功能

外部设备所需信息格式往往与**CPU**的不一致，需要接口电路进行相应的信息格式变换。如正负逻辑关系转换、时序配合上的转换、电平匹配转换、串-并转换等。

7.1 I/O接口概述

三、I/O接口的基本结构与分类

1. I/O接口的基本结构

由端口寄存器和控制逻辑两大部分组成

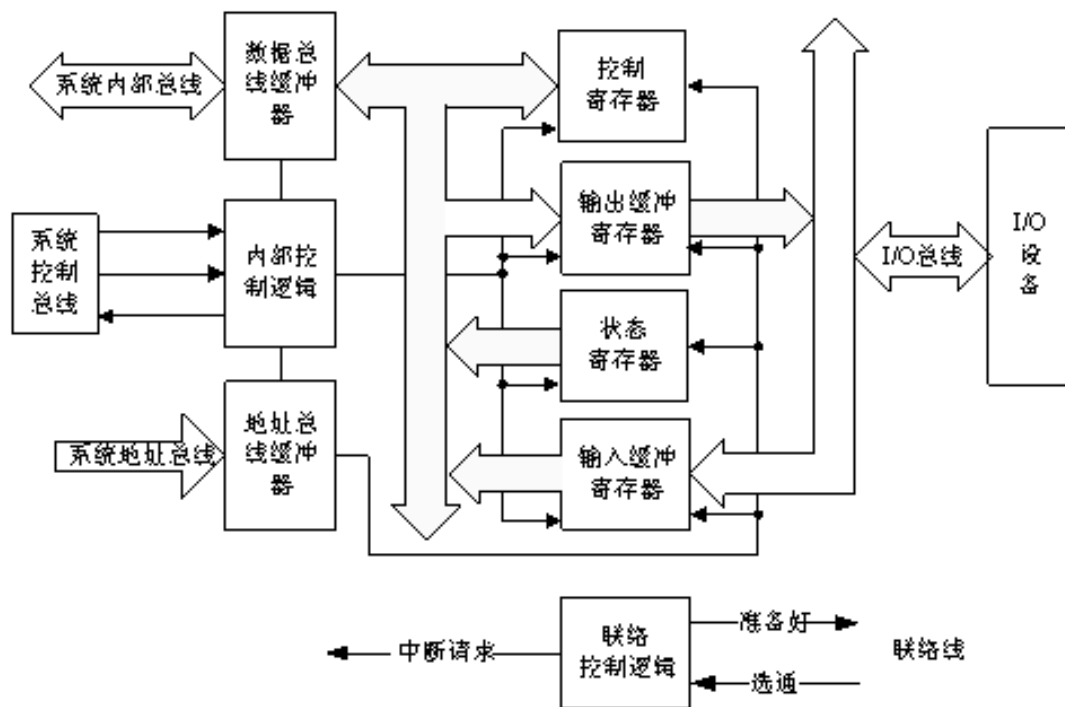


图7.2 接口电路的基本结构

7.1 I/O接口概述

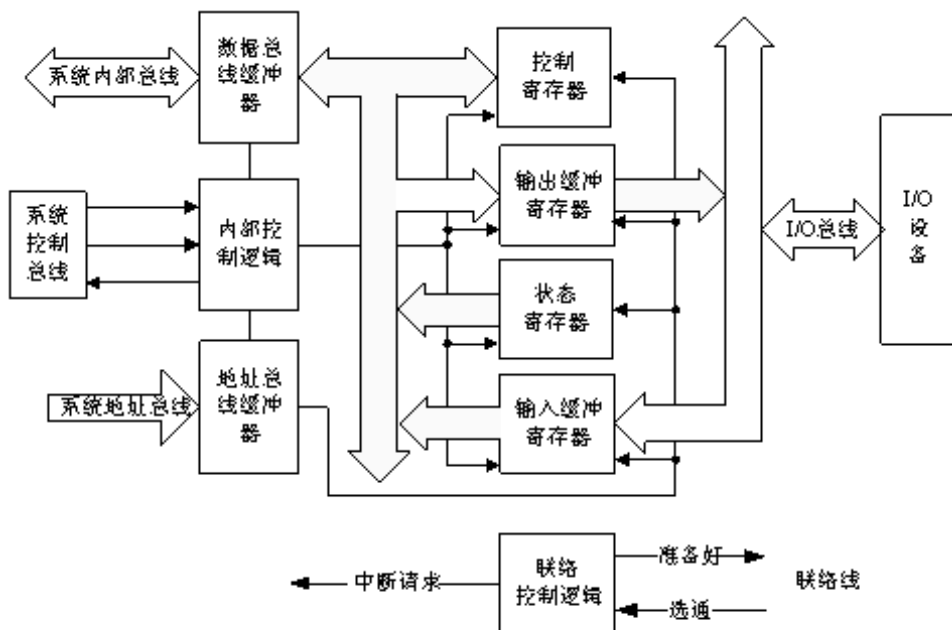


图7.2 接口电路的基本结构

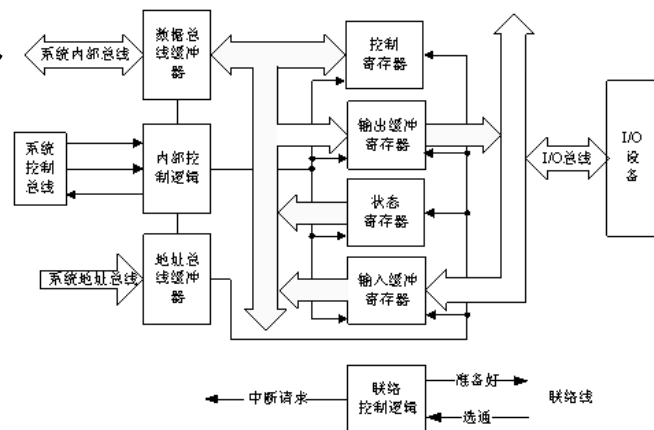
(1) 端口寄存器

- 数据缓冲寄存器：分为**输入缓冲寄存器**和**输出缓冲寄存器**，在高速**CPU**与低速外部设备之间起**协调、缓冲**作用。
- 控制寄存器：**只能写不能读**，存放**CPU**向外部设备发送的控制命令和工作方式命令字等。
- 状态寄存器：**只能读而不能写**，存放外部设备当前的工作状态信息，供**CPU**查询。

7.1 I/O接口概述

(2) 控制逻辑电路

- ❑ **数据总线缓冲器**：连接接口芯片内部的数据总线与系统总线
- ❑ **地址译码**：系统地址总线高位经片外的地址译码器译码后用来选择接口芯片，低位地址线在片内译码后选择接口芯片内部相应的端口寄存器，使CPU能够正确无误地与指定的外部设备完成相应的I/O操作。
- ❑ **内部控制逻辑**：接收来自系统的控制输入，产生接口电路内部的控制信号，实现系统控制总线与内部控制信号之间的转换。
- ❑ **联络控制逻辑**：接收来自CPU的有关控制信号，生成给外部设备的准备好信号和相应状态；接收外部设备的选通信号，产生相应状态标志和中断请求信号。





7.1 I/O接口概述

2. 接口分类

I/O接口电路从不同角度分为4种：

- ✓ 按数据传送方式，分为并行接口和串行接口；
- ✓ 按功能选择的灵活性，分为可编程接口和不可编程接口；
- ✓ 按通用性，分为通用接口和专用接口；
- ✓ 按数据控制方式，分为程序型接口和DMA型接口。

目前的I/O接口电路大多采用大规模、超大规模集成电路，并向智能化、系列化和一体化方向发展。



7.2 I/O端口

端口：每个接口电路内部所含的寄存器称为**I/O端口（PORT）**。

- 每个**端口**对应一个**I/O端口地址**；
- **数据端口**：数据缓冲寄存器，用来存储**CPU**与外部设备之间传送的**数据信息**；
- **状态端口**：状态寄存器，用来存放外部设备或接口部件的**状态**（**准备就绪位Ready**，**忙碌位Busy**，**错误位Error**）；
- **控制接口**：控制寄存器，用于存放**CPU**发出的命令，控制接口和外部设备的**动作**。



7.2 I/O端口

一、I/O端口的编址方式

1. **统一编址** 将I/O端口地址与存储单元一起编址。

□ I/O端口占用部分内存地址空间，I/O端口可看作是存储器的一部分。

□ I/O端口和存储单元具有不同的地址编号，存储器使用的指令都可以用于I/O端口，无需设置专门的I/O指令，但一条指令到底访问的是存储器还是I/O端口是由其**具体的地址编号**决定。

优点：指令类型多，功能齐全。

缺点：

存储器容量减少；
因而执行时间较长；
不易于程序的阅读和理解。



7.2 I/O端口

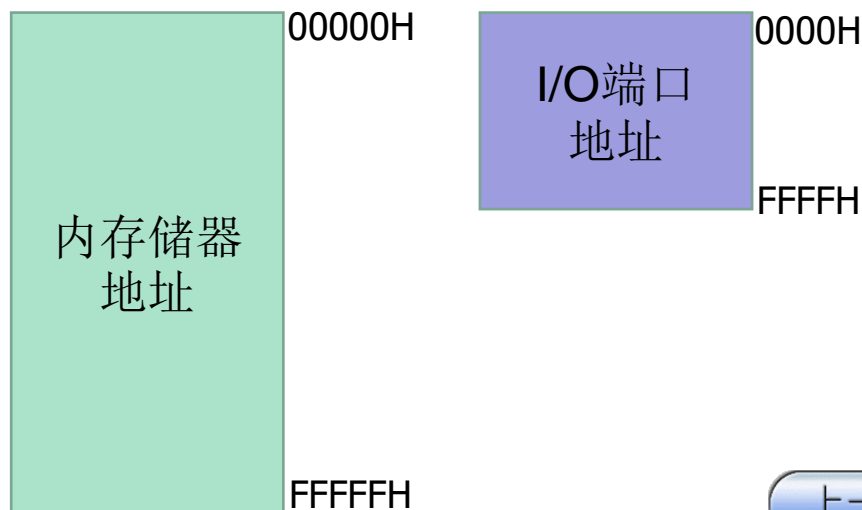
2. **独立编址** I/O端口地址和存储单元地址各自**独立**。

I/O端口采用8位/16位地址编址，**端口地址范围**为0~255（00H~FFH）/0~65535（0000H~FFFFH），I/O操作使用输入/输出指令**IN**和**OUT**。

优点:I/O端口地址不占用存储器空间，有专门的I/O指令，I/O指令短，执行速度快；

I/O指令与存储器访问指令区别明显，便于程序阅读和理解

缺点: 访问I/O端口的指令功能**较弱**，一些操作必须由I/O设备先输入到**CPU**的寄存器后才能进行。





7.2 I/O端口

二、I/O指令

1. 输入指令

(1) `IN AL/AX, PORT` ; 直接寻址的输入指令

功能：把8/16位数据直接由输入端口PORT（地址0~255）输入到AL/AX寄存器中。例如：

`IN AL, 0DFH` ; 从端口0DFH读一个字节到AL中

(2) `IN AL/AX, DX` ; 间接寻址的输入指令

功能：把8/16位数据由DX寄存器指定的端口地址输入到AL/AX寄存器中。例如：

`MOV DX, 300H`

`IN AX, DX` ; 从端口300H读一个字到AX中



7.2 I/O端口

2. 输出指令

(1) **OUT PORT,AL/AX** ;直接寻址的输出指令

功能：把AL/AX的数据直接输出到PORT指定的输出端口地址（0~255）。 例如：

OUT 80H,AL ;把AL的内容输出到80H端口中

(2) **OUT DX,AL/AX** ;间接寻址的输出指令

功能：把AL/AX的数据送到由DX寄存器指定的输出端口
例如：

MOV DX,310H
OUT DX,AL ;把AL的内容输出到端口310H中



7.2 I/O端口

三、I/O地址分配

I/O地址空间分为两部分：

- ✓ 供系统板上的I/O芯片使用的：定时/计数器、中断控制器、DMA控制器、并行接口等；
- ✓ 供I/O扩展槽上的接口控制卡使用的：软驱卡、硬驱卡、图形卡、声卡、打印卡、串行通信卡等。

表7-1 IBM-PC机端口地址分配

I/O接口名称	端口地址
DMA控制器1	000H~01FH
DMA控制器2	0C0H~0DFH
DMA页面寄存器	080H~09FH
中断控制器1	020H~03FH
中断控制器2	0A0H~0BFH
定时器	040H~05FH
键盘控制器	060H~06FH
RT/CMOS RAM	070H~07FH
协处理器	0F0H~0FFH

表7-2 IBM-PC/AT机端口地址分配

I/O接口名称	端口地址
游戏控制卡	200H~20FH
并行口控制卡1	370H~37FH
并行口控制卡2	270H~27FH
串行口控制卡1	3F8H~3FFH
串行口控制卡2	2F0H~2FFH
原型插件板（用户可用）	300H~31FH
同步通信卡1	3A0F~3AFH
同步通信卡2	380H~38FH
单显MDA	3B0H~3BFH
彩显CGA彩显EGA/VGA	3D0H~3DFH 3C0H~3CFH
硬驱控制卡	1F0H~1FFH
软驱控制卡	3F0H~3F7H



7.2 I/O端口

为了避免端口地址发生冲突，在使用和设计接口电路时，应遵循的**原则**：

- (1) 凡是**已被系统使用**的端口地址，不能再作为它用；
- (2) 凡是**被系统声明为保留**的地址，尽量不要作为它用。否则，可能与其它或未来的产品发生I/O端口地址重叠和冲突，从而造成与系统的不兼容；
- (3) **一般用户可使用300H~31FH地址**。同时，为了避免与其它用户开发的插板发生地址冲突，最好采用地址开关。



7.2 I/O端口

四、I/O端口地址译码

一般译码原则：

把地址分为两部分：

- ✓ 高位地址线与CPU的控制信号组合，通过译码产生I/O接口芯片的片选信号，实现片间寻址。片选地址线连接较复杂，通常按译码电路的形式分为固定式和可选式译码。
- ✓ 低位地址线直接连到I/O接口芯片，实现片内寻址，即访问片内寄存器。片内地址线的连接只需将芯片的地址输入线与系统地址总线的相应片内地址线一一相连即可。

7.2 I/O端口

1. 固定式端口地址译码

接口中用到的端口地址由硬件连线决定，不能更改。这种译码方式多用于不需改变端口地址的场合，当只需要一个端口地址时，可采用门电路构成译码电路。例如，“读2F8H端口”的电路。**AEN**参加译码，它对端口地址译码进行控制，**AEN=0**时，译码有效；**AEN=1**时，使译码无效。

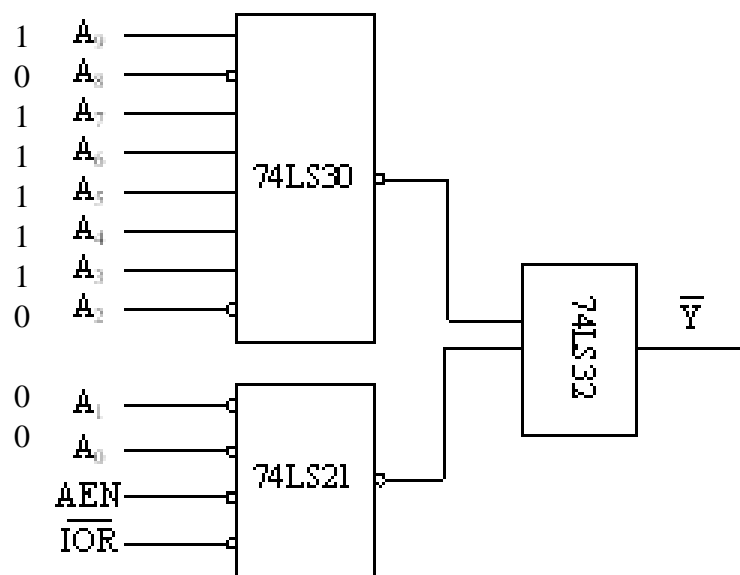
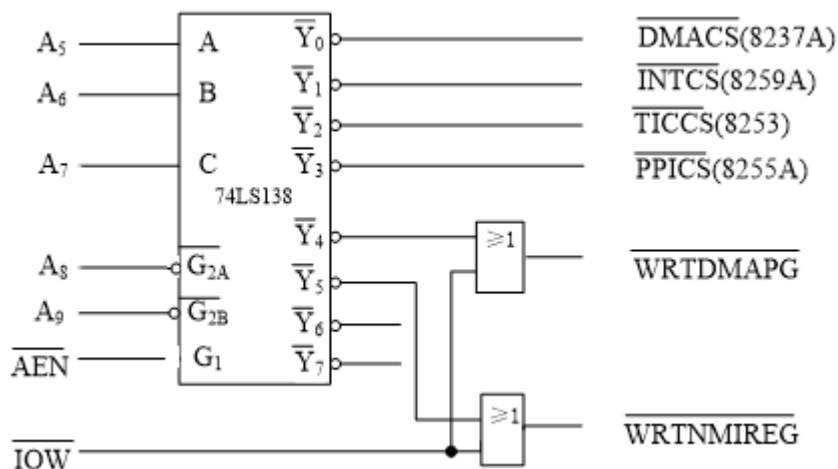


图7.3 门电路译码电路

7.2 I/O端口

当接口电路需要**多个端口地址**时，一般采用译码器进行译码。常用的译码器有**2-4译码器74LS139**、**3-8译码器74LS138**、**4-16译码器74LS154**等。



PC/XT 机的 I/O 端口分配表

分类	地址范围(H)	I/O 设备(端口)
系统板	000~01F(00~0F)	8237A-5 DMA 控制器
	020~03F(20~21)	8259A 中断控制器
	040~05F(40~43)	8253-5 计数器/定时器
	060~07F(60~63)	8255A-5 并行接口
	080~09F(80~83)	DMA 页寄存器
	0A0~0BF(A0)	NMI 屏蔽寄存器
	0C0~0DF	保留
	0E0~0FF	保留

$\overline{AEN}=1$ ，这时若 $A_9A_8=00$ ，则译码器选通，对输入 $A_7A_6A_5$ 译码，在 $\overline{Y_0}\sim\overline{Y_7}$ 中产生一个低电平输出信号，接到相应接口芯片的 \overline{CS} 端。

$A_4\sim A_0$ 用作对 I/O 接口芯片片内端口寻址

●使用低10位地址($A_9\sim A_0$)寻址I/O端口，I/O地址空间占1K。当 $A_9=0$ 时，寻址系统板上的512个端口；当 $A_9=1$ 时，寻址I/O通道上的512个端口。



7.2 I/O端口

2. 开关式可选地址译码

能适应不同场合的地址分配或需要为系统以后扩展留有余地时采用，可以通过开关使接口卡的I/O端口地址根据要求加以改变而无需改动线路。有两种电路结构形式。

(1) 直接使用拨动开关或跳接开关

(2) 使用地址开关加比较器

7.2 I/O端口

(1) 直接使用拨动开关或跳接开关

两个异步通信接口**COM1**和**COM2**的地址译码电路。当跳接开关**U15**接**4、8**两点时，地址范围为**3F8H~3FFH**，选中**COM1**为当前串行口适配器；当跳接开关**U15**接**2、6**两点时，地址范围为**2F8H~2FFH**，选中**COM2**为当前串行口适配器。

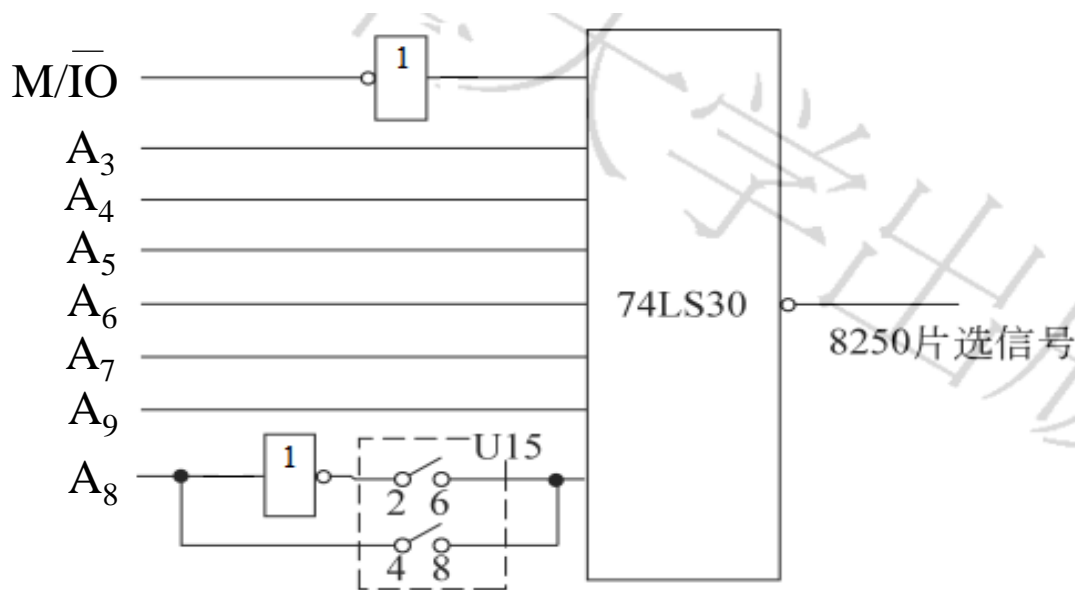


图 7.7 用跳线开关选择 I/O 端口的译码地址

7.2 I/O端口

(2) 使用地址开关加比较器

关键器件是比较器74LS688，有两组输入端，一个输出端。其规则为：

当 $P7 \sim P0 = Q7 \sim Q0$ 时，
输出低电平

当 $P7 \sim P0 \neq Q7 \sim Q0$ 时，
输出高电平

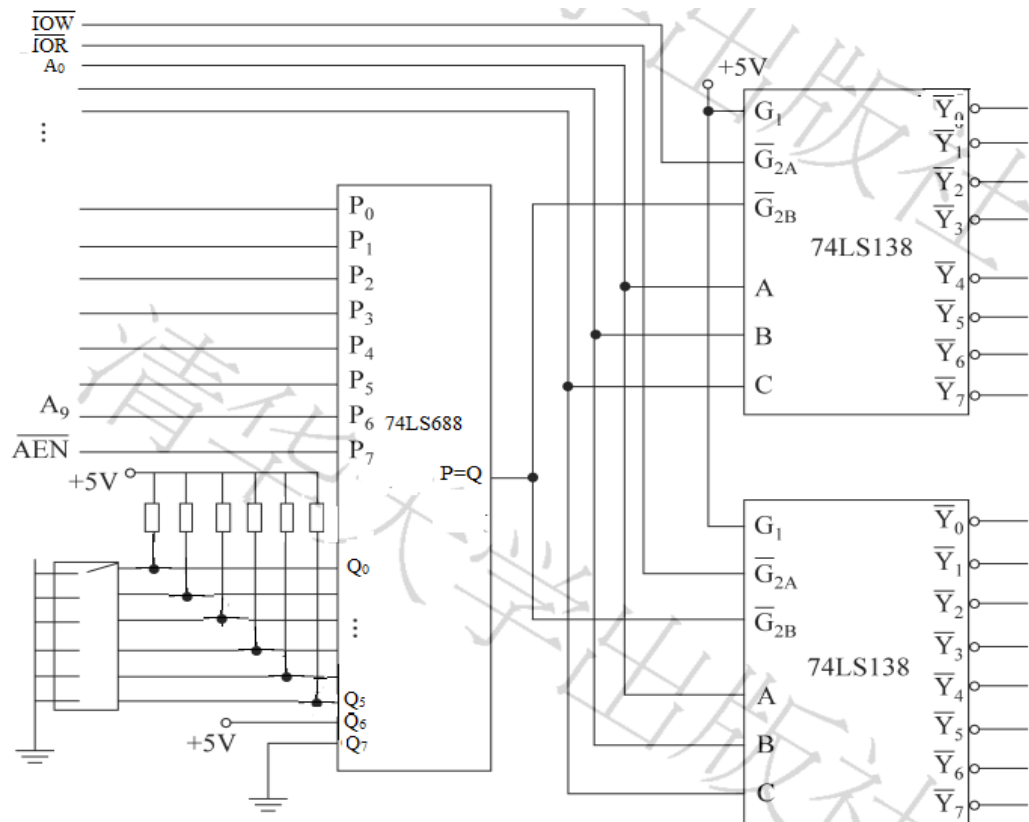


图 7.8 由比较器组成的可选式译码电路



7.3 CPU与外设间的数据传送方式

按照I/O接口电路复杂程度的演变顺序和外部设备与CPU并行工作的程度，CPU与外部设备之间的信息传送方式分为三种：

- ✓ 程序控制传送方式
- ✓ 中断传送方式
- ✓ DMA传送方式



7.3 CPU与外设间的数据传送方式

一、程序控制传送方式

由程序直接控制外部设备与CPU之间的数据传送过程。当需要进行数据传送时，通过I/O指令组成的程序段，直接控制外部设备工作。

特点：编写程序之前需预先知道何时进行这种数据交换工作。

根据外部设备的特点，程序控制传送方式又分为：

无条件传送方式

条件传送（程序查询传送）方式。

7.3 CPU与外设间的数据传送方式

1. 无条件传送方式

无条件传送（同步传送），是**最简单**的程序控制传送方式。当**CPU**能够确信外部设备已经准备就绪时，可不查询外部设备状态而**直接进行信息传送**。一般较少使用，只用于如开关、数码管等一些较简单的外部设备控制。

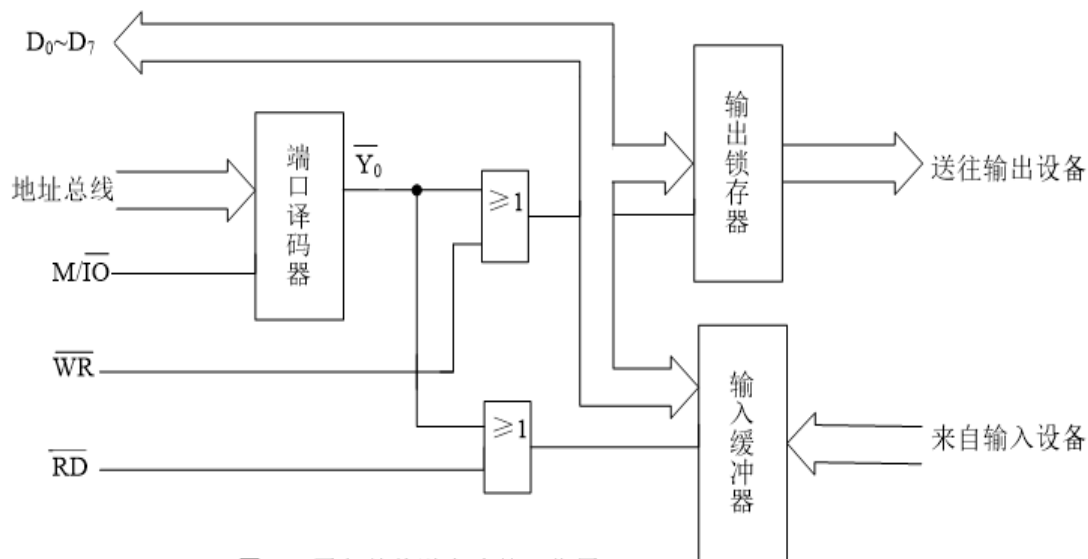
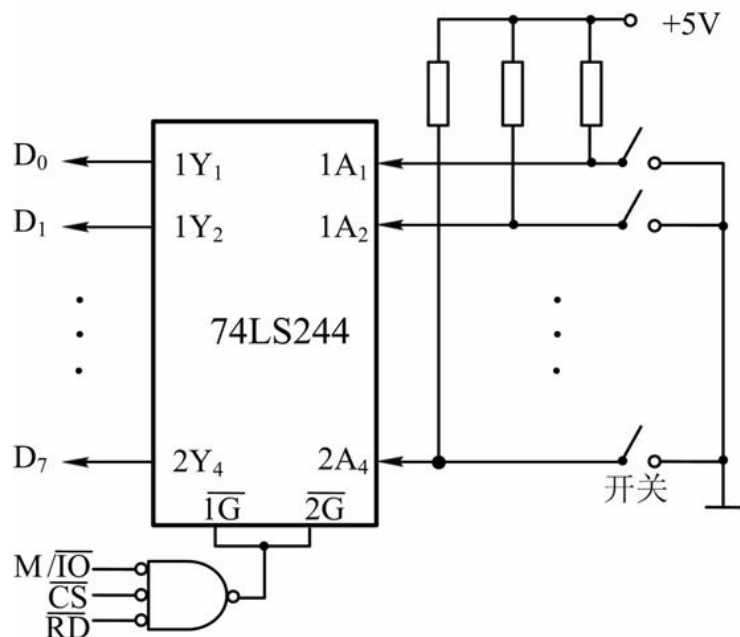


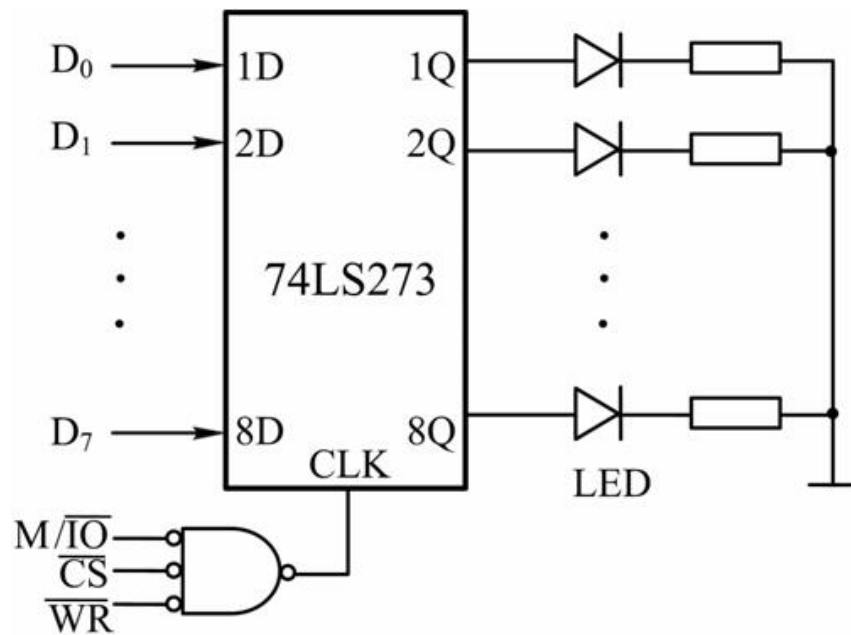
图 7.9 无条件传送方式的工作原理

7.3 CPU与外设间的数据传送方式

1. 无条件传送方式



- 最简单的输入端口：按键开关的状态输入口。
- 开关经1个8位三态缓冲器接到数据总线，用IN指令将开关状态读入缓冲器。
- 1-断开，0-闭合。
- 可随时读取开关状态。



- 最简单的输出口：LED的点亮和熄灭控制。
- 1个8位锁存器接8个LED显示器，用OUT指令将命令字送到LED，1-点亮，0-熄灭。
- 可随时改变LED状态。

7.3 CPU与外设间的数据传送方式

2. 条件传送方式

条件传送方式（程序查询）：**CPU**传送数据前，**不断读取并检测外部设备的状态**，只有外部设备状态信息满足条件时才进行数据传送，否则就一直等待。

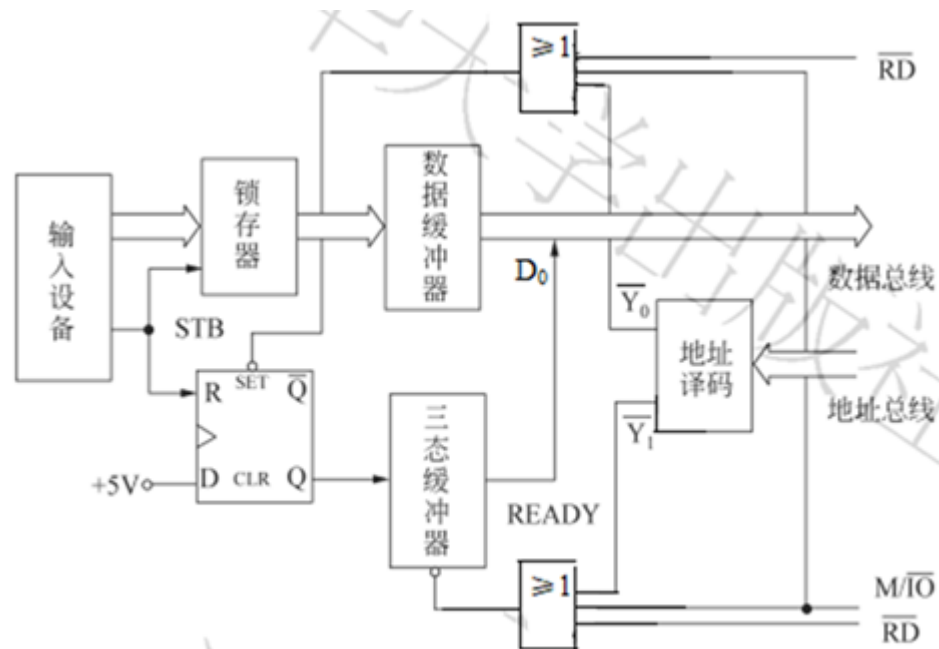


图 7.10 查询式输入接口电路

- 查询式输入接口电路
- 状态口由1个D触发器和1个三态门构成，执行IN指令选中状态口(\overline{Y}_1 有效)时，读取状态信息READY。READY=0，外设未就绪；READY=1，CPU可读外设。
- 输入数据口由1个8位锁存器和1个8位缓冲器构成，执行IN指令选中数据口(\overline{Y}_0 有效)时，从缓冲器读取数据信息。

```
READ_S: IN    AL, PORT_S ; 读入状态位
        TEST  AL, 01H    ; 数据准备好?
        JZ    READ_S     ; D0为0, 循环检测
        IN    AL, PORT_D
```

7.3 CPU与外设间的数据传送方式

2. 条件传送方式

条件传送方式（程序查询）：**CPU**传送数据前，**不断读取并检测外部设备的状态**，只有外部设备状态信息满足条件时才进行数据传送，否则就一直等待。

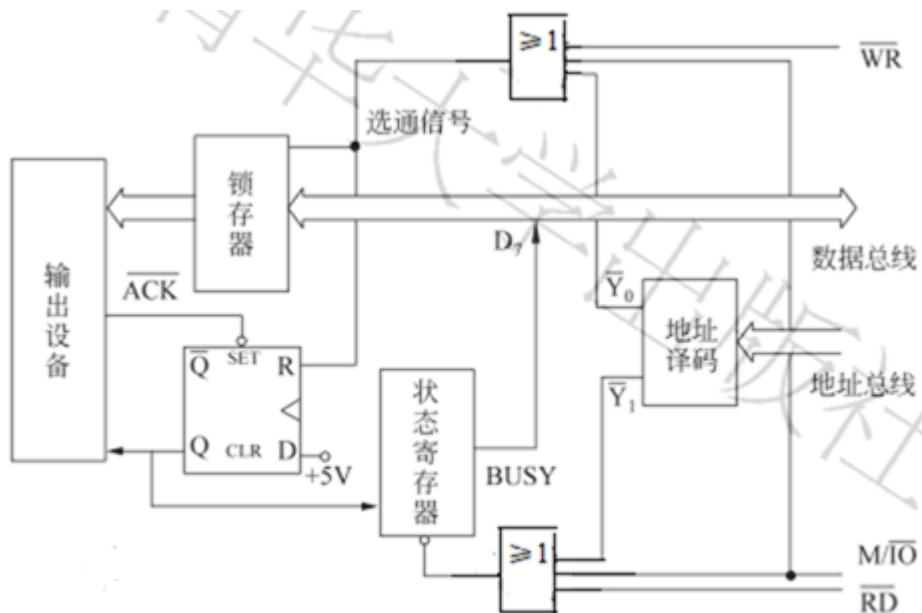


图 7.11 查询式输出接口电路

- 查询式输出电路和工作过程
 - CPU准备输出数据时先执行IN指令，它使状态口三态门开启，从数据总线D₇位读入BUSY状态。BUSY=1，外设忙；BUSY=0，CPU可向外设输出数据。
 - 外设不忙时，执行OUT指令选中数据口，锁存器选通，将数据送向外设。同时，其后沿还使D触发器翻转，置Q为高电平。
 - 输出设备取走数据后，送回一个应答信号，将D触发器清0。

7.3 CPU与外设间的数据传送方式

查询式输入/输出传送数据一般需要三个步骤

- (1) 读外部设备状态端口信息到CPU相应寄存器
- (2) 检测相应状态位，判断外部设备是否“准备就绪”
- (3) 如果已经“准备就绪”，则开始传送数据；否则重复执行(1)、(2)步，直到外部设备“准备就绪”

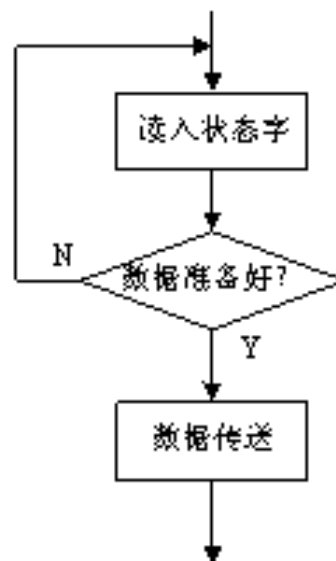


图7.10 查询式数据传送流程



7.3 CPU与外设间的数据传送方式

二、中断传送方式

基本思想：外部设备准备就绪（输入设备将数据准备好或输出设备可以接收数据）时，主动向**CPU**发出中断请求，使**CPU**中断当前正在执行的程序，转去执行输入/输出中断服务程序进行数据传送，传送完毕后再返回原来的断点处继续执行。

特点：

- ✓ 在一定程度上实现了**CPU与外部设备的并行工作**
- ✓ 提高了**CPU的利用率**
- ✓ 实现了**外部设备的并行工作**。

7.3 CPU与外设间的数据传送方式

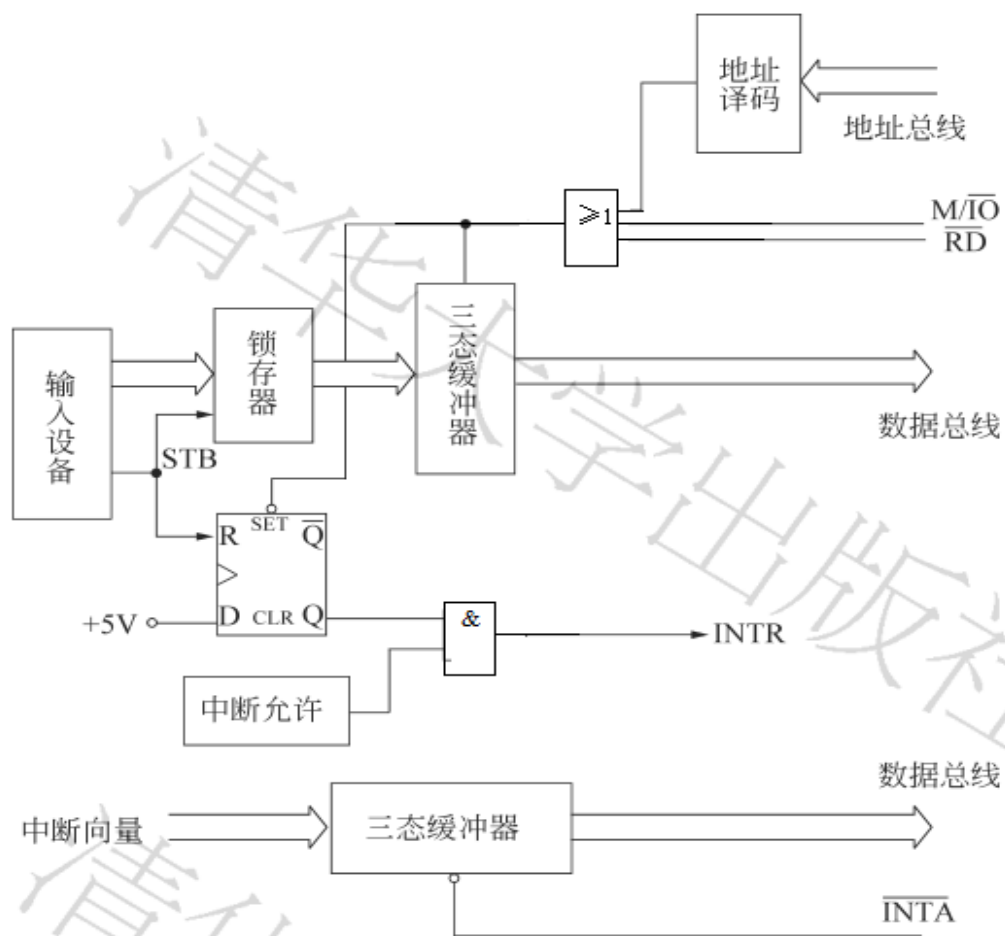


图 7.13 中断传送方式接口电路



7.3 CPU与外设间的数据传送方式

三、DMA传送方式

思想：

DMA（**Direct Memory Access**,直接存储器访问）传送方式，**CPU**不参与，在外部设备与存储器之间开辟直接的数据交换通路，由专门的硬件**DMAC**（**DMA**控制器）控制数据在内存与外设、外设与外设之间进行直接传送。

特点：

- （1）在硬件控制下完成数据传送，减轻了**CPU**的负担；
- （2）传送的速度上限取决于存储器的工作速度，大大提高了数据传送速率；
- （3）**DMAC**获得总线控制权，为系统的主控部件，**CPU**不再控制系统总线。

7.3 CPU与外设间的数据传送方式

DMAC只负责送出地址及控制信号，数据传送是直接接口和内存之间进行的，对于内存到内存之间的传送是先用一个**DMA**存储器读周期将数据从内存读出，放在**DMAC**中的内部数据暂存器中，再利用另一个**DMA**存储器写周期将数据写入内存指定位置。

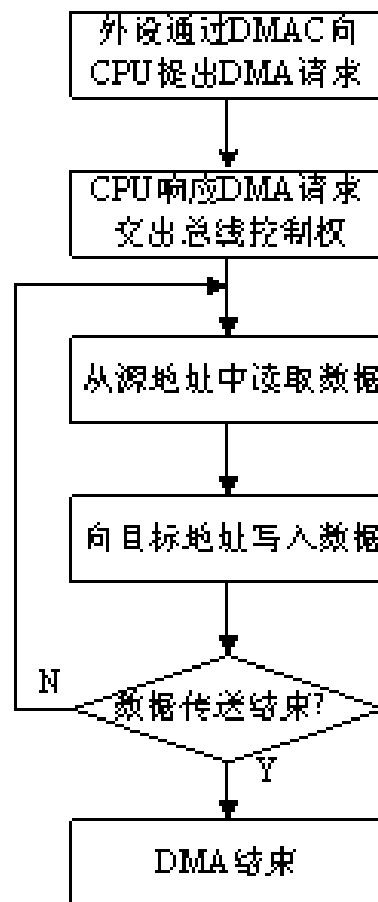
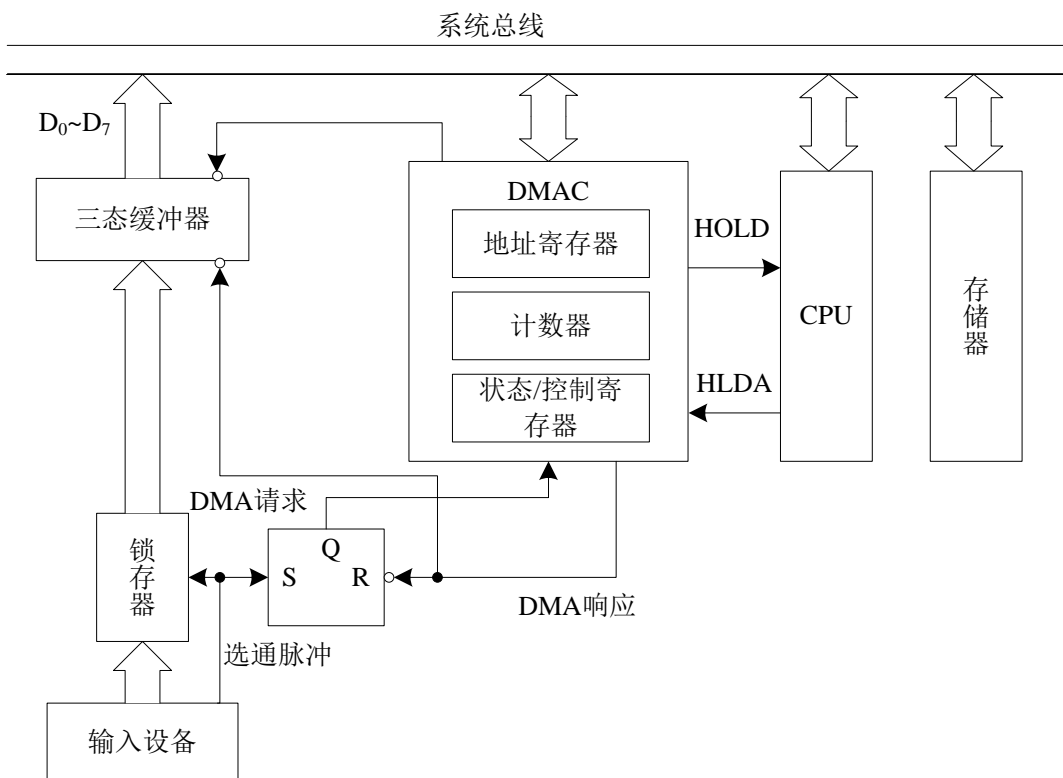


图7.13 DMA方式数据传送流程

7.3 CPU与外设间的数据传送方式



DMA传送方式输入接口电路

1. 输入设备准备好数据后，发选通信号，一方面把数据存入锁存器，另一方面把DMA请求触发器置1，向DMAC发出DMA请求信号。

2. DMAC收到DMA请求后向CPU发出HOLD信号，请求使用总线。

3. 当CPU完成当前总线周期后发出HLDA信号，同时放弃对总线的控制权。

4. DMAC接管总线，向输入设备发出DMA响应信号，进入DMA工作方式。

5. DMAC发出地址信号和相应的控制信号把外设输入的数据写入存储器，然后修改地址指针和字节计数器，待规定的数据传送完后，DMAC撤消向CPU的HOLD信号。

6. CPU检测到HOLD信号失效后也撤消HLDA信号，并在下一时钟周期重新接管总线。