



河南大學  
Henan University

# 汇编语言与接口技术

## —— 第 7 章 串并行接口技术

---

主讲教师：舒高峰

电子邮箱：gaofeng.shu@henu.edu.cn

联系电话：13161693313

00

# 上节回顾-接口及其数据传输控制方式

## 简答题

- 接口的基本功能有哪些？
- 什么是端口？80x86 系统的端口编址方式是什么？

# 目录

---

- 01 定时/计数器 8253/8254
- 02 并行通信接口 8255A
- 03 串行通信芯片 8251A

# 01 | 定时/计数器-子目录

## 接口概述

- 定时/计数器的概念
- 8253A 可编程定时/计数器
- 8253A 控制字和工作方式
- 8253A 的编程应用



# 01 | 定时/计数器-概念

## 定时

- **日常生活**：日时钟、时时钟
- **微机系统**：系统时钟、DRAM 刷新定时、周期性控制信号

## 定时与计数的关系

- 相同点
  - 由数字电路中的**计数电路**构成
  - 定时就是采用时间单位进行计数
- 不同点
  - **定时**是对**固定频率**的脉冲信号进行计数
  - **计数**是对**随机性**的脉冲信号进行计数

# 01 | 定时/计数器-微机系统中的定时

## 内部定时

- CPU (主机) 本身工作的控制时序, 如**时钟周期**
- **由硬件结构确定**, 微机中所有操作都要严格按照此节拍来完成, 是**固定的**, 无法更改的
- 一般为 ns (纳秒) 级

## 外部定时

- **外设工作时**, 所需要的**时序关系**
  - 实时监控系统的定时中断、定时监测、定时扫描等
  - 对 I/O 设备运行速度和工作频率的**调整和控制**
- 一般定时为 ms (毫秒) 级

# 01 | 定时/计数器-外部定时方法-软件定时

## 软件定时

- **定时方法：**
  - 通过 CPU 执行一段**循环程序**，而产生**等待延时**
  - 需要按延时的时间来计算时间常数
- **优点：**
  - 节省硬件成本
- **缺点：**
  - **占用 CPU 的时间**，降低了 CPU 的工作效率
  - 软件延时的时间随主机频率不同而发生变化，**通用性差**
- 适用于**短时间或不频繁发生延时**的简单系统

# 01 | 定时/计数器-外部定时方法-软件定时

## 软件定时

- 延时程序:

```
1.      .....  
2.      MOV DX, 0FFFFH      ; 延时常数  
3. AGAIN: MOV CX, DX  
4. WAIT1: NOP  
5.      LOOP WAIT1  
6.      DEC DX  
7.      JNZ AGAIN  
8.      .....  
          }
```

; 也可单独作为一个延时程序, 延时时间较短

- 延时常数决定于:

- 需要的延时时间、系统主频、指令的执行时间



# 01 | 定时/计数器-外部定时方法-硬件定时

## 硬件定时

- **定时方法：**
  - 采用**可编程定时/计数器**或**单稳延时电路**产生定时或延时
- **特点：**
  - **不占用 CPU 时间**，提高了 CPU 的利用率
  - 可以产生精确的时间间隔，形成各种脉冲序列，**灵活性强**
  - 定时不受主机频率影响，定时程序**通用性强**
- 可用于**各种定时计数场合**

# 01 | 定时/计数器-子目录

## 接口概述

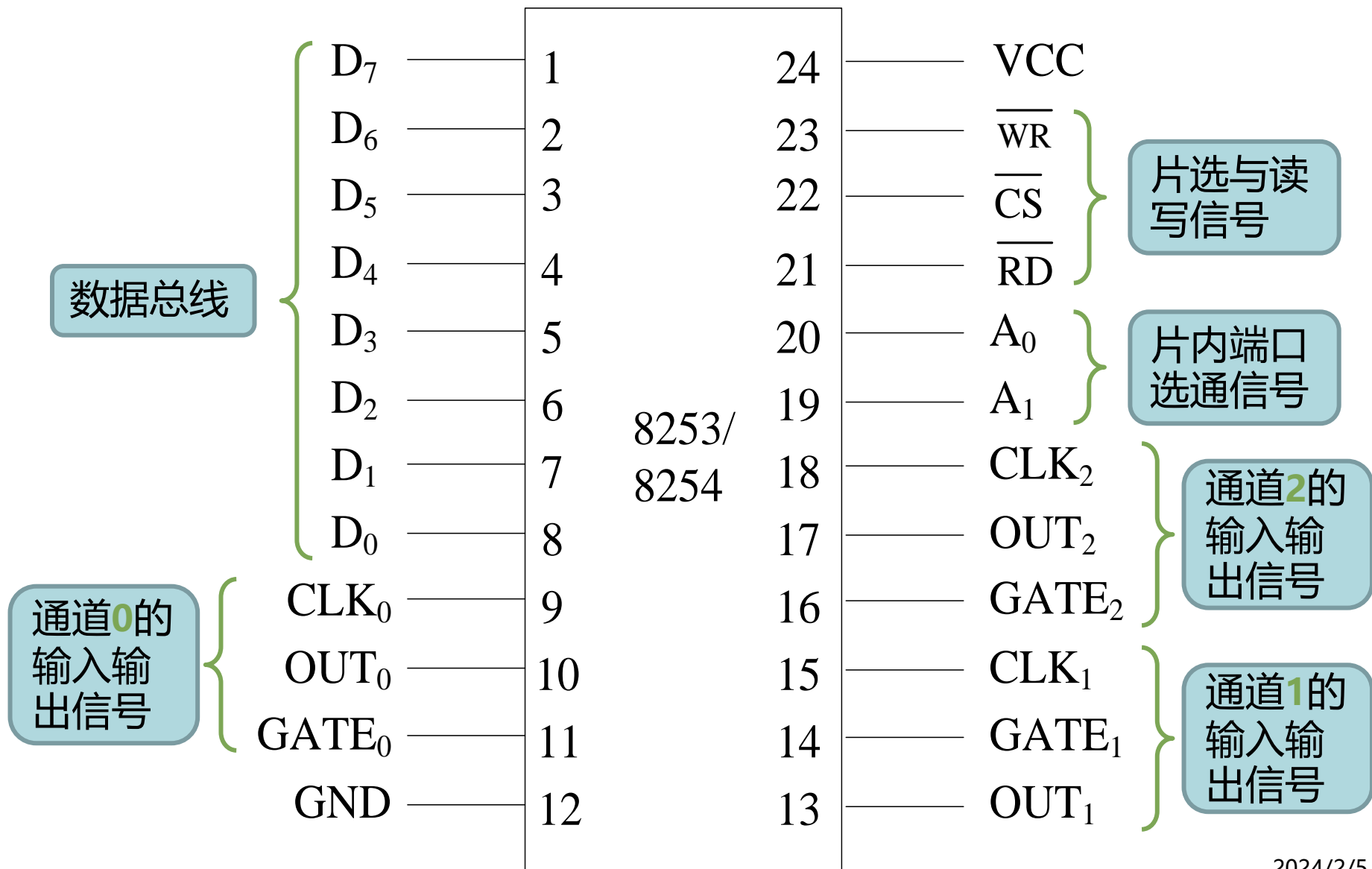
- 定时/计数器的概念
- 8253A 可编程定时/计数器
  - 8253A芯片的工作特点
  - 8253A芯片的外部引脚特性
  - 8253A的内部结构
  - 计数初值的设置
- 8253A 控制字和工作方式
- 8253A 的编程应用

# 01 | 定时/计数器-8253A的工作特点

## 工作特点

- 每片 8253A 上都有 3 个独立的 16 位的计数通道
  - 每个计数通道最大可计数值为 $2^{16}=65536$
- 每个通道计数均可编程设置为二进制或 BCD 码计数
  - BCD 码计数即十进制计数
- 每个计数通道都有 6 种工作方式，可以编程选择
  - 每种工作方式的输出波形、触发方式等不同
- 计数通道都是减法计数器
  - 计数/定时启动之前需要设置计数初值/定时常数

# 01 | 定时/计数器-8253A的外部引脚



# 01 | 定时/计数器-计数通道的外部信号

## 时钟输入信号 CLK

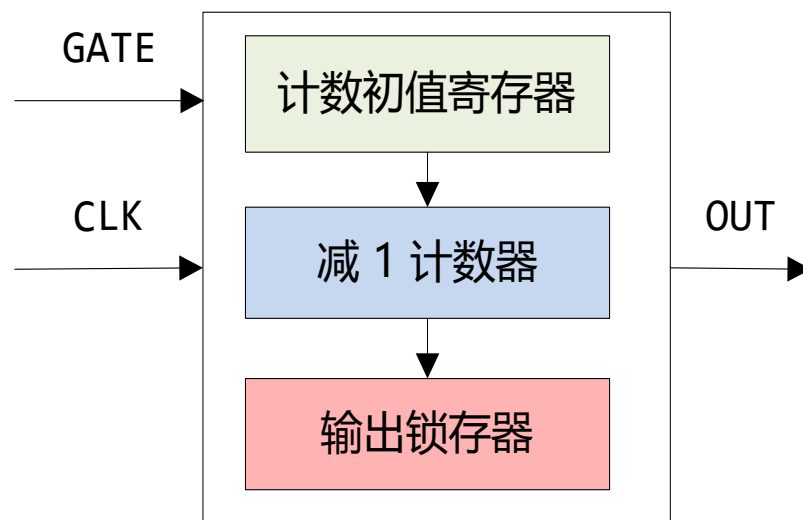
- 定时计数时，每经过一个脉冲/时钟周期，计数值减 1

## 门控输入信号 GATE

- 用于控制计数的启停
- 可有**开关作用**和**触发作用**

## 计数器输出信号 OUT

- 各种波形输出端
- 可供 CPU 检测状态，也可用于提供请求信号



# 01 | 定时/计数器-读写操作类型

## 8253A 读/写逻辑信号输入及功能

| $\overline{CS}$ | $\overline{RD}$ | $\overline{WR}$ | $A_1$ | $A_0$ | 功能             |
|-----------------|-----------------|-----------------|-------|-------|----------------|
| 0               | 1               | 0               | 0     | 0     | 设置计数器 0 初值     |
| 0               | 1               | 0               | 0     | 1     | 设置计数器 1 初值     |
| 0               | 1               | 0               | 1     | 0     | 设置计数器 2 初值     |
| 0               | 1               | 0               | 1     | 1     | 设置控制字, 以后为设置命令 |
| 0               | 0               | 1               | 0     | 0     | 从计数器 0 中读出计数值  |
| 0               | 0               | 1               | 0     | 1     | 从计数器 1 中读出计数值  |
| 0               | 0               | 1               | 1     | 0     | 从计数器 2 中读出计数值  |
| 0               | 0               | 1               | 1     | 1     | 高阻, 无操作        |
| 1               | X               | X               | X     | X     |                |

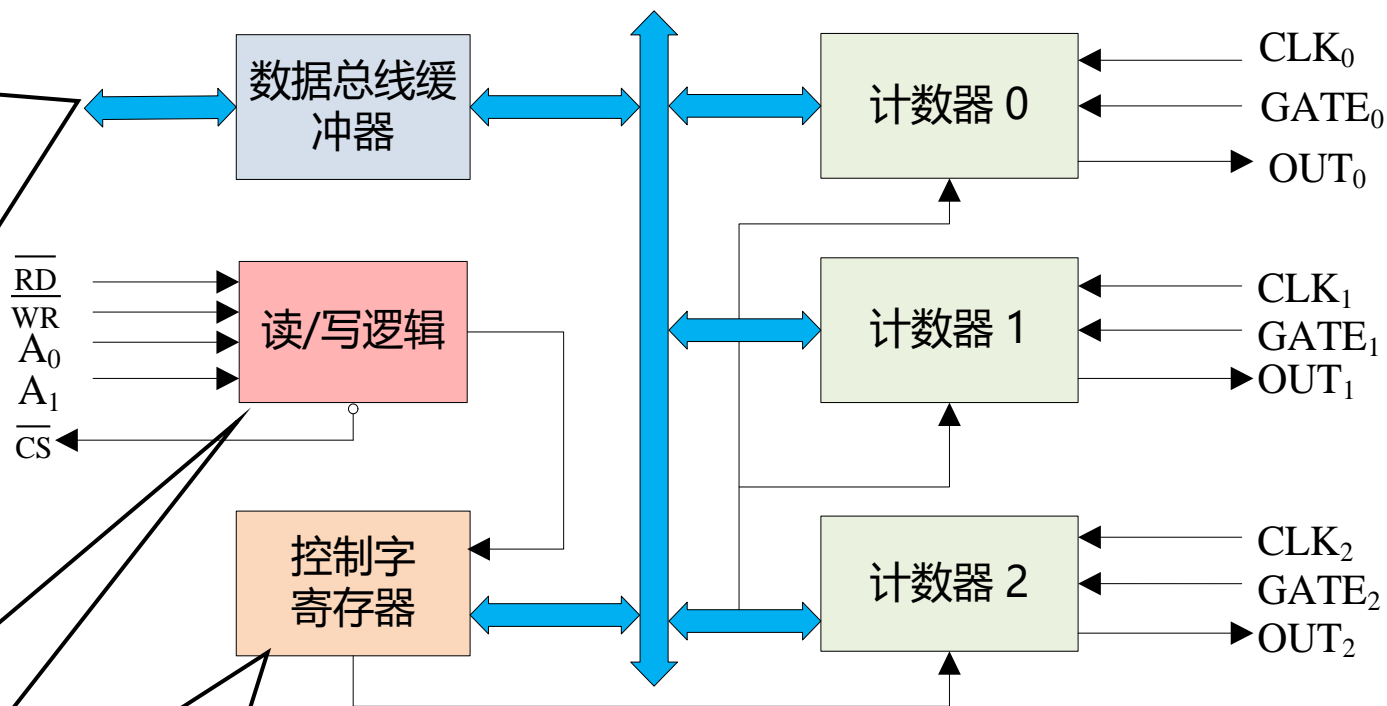
# 01 | 定时/计数器-8253A的内部结构

## ● 数据总线缓冲器:

- ✓ 三态双向 8 位寄存器
- ✓ 8253A 与 CPU 之间**数据交换**的部件

## ● 读/写控制逻辑:

- ✓ 接收 CPU 有关控制信号, 决定操作**对象**及**类型**

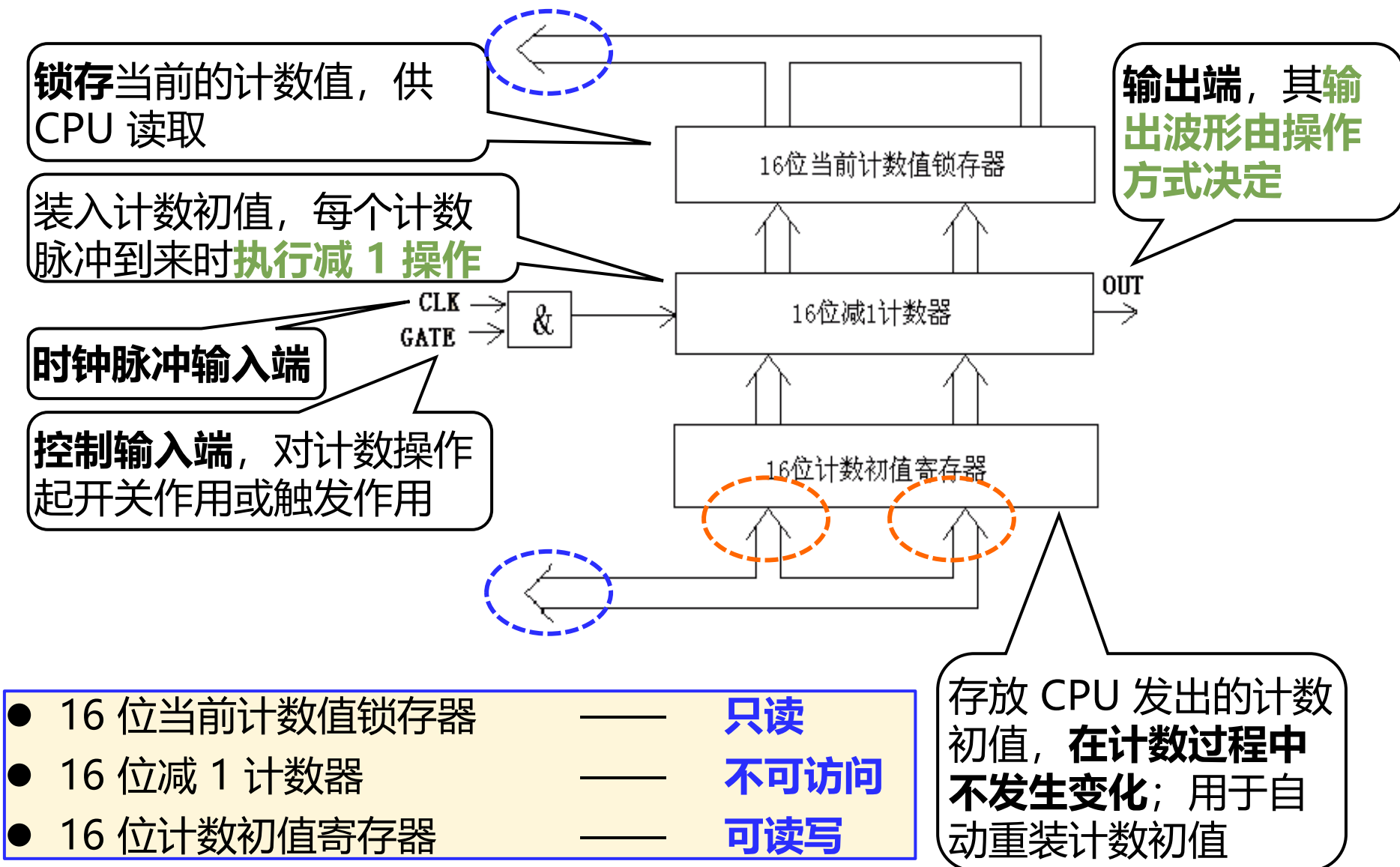


## ● 控制字寄存器:

- ✓ 接收 CPU 发出的控制字, 决定各计数器的**工作方式**
- ✓ 三个计数通道**共用一个命令端口**

- 计数通道为**数据端口**, 每个计数通道分别占用一个端口地址
  - (计数通道0-304H、计数通道1-305H、计数通道2-306H)
- 控制字寄存器为**命令端口**, 占用一个端口地址 (307H)
- 无**状态端口**

# 01 | 定时/计数器-计数通道的内部结构





# 01 | 定时/计数器-计数初值的设置

## 8253A 芯片的初始化步骤

1. 按照命令字格式写出满足当前操作的**命令字**，使用 I/O 指令将其写入**命令端口**
  - 写入命令字后，**计数初值寄存器清零**
2. 确定**计数初值** (定时常数)，使用 I/O 指令将初始值装入指定计数通道的初值寄存器
  - 计数初值会通过初值寄存器送入减 1 计数器
  - 当计数**条件满足**时，减 1 计数器即开始定时/计数

# 01 | 定时/计数器-计数初值的确定

## 按定时时间计算

- $t$  — 定时时间、 $T_{CLK}$  — 时钟周期

则计算公式为：
$$n = t / T_{CLK}$$

## 按脉冲序列频率计算

- $f_{CLK}$  — 输入时钟频率、 $f_{OUT}$  — 输出波形频率

则计算公式为：
$$n = f_{CLK} / f_{OUT}$$

## 计数初值的范围

- 二进制：00000 ~ 0FFFFH
- 十进制 (BCD 码)：0000 ~ 9999
- 其中，**最大值为 0**，因为计数器计数时先减 1 后判断

# 01 | 定时/计数器-计数初值/定时常数计算例子

假定输入时钟频率  $f_{CLK}=2\text{MHz}$

- 要求1: 定时 20 ms

$$\begin{aligned}n &= t/T_{CLK} = t \times f_{CLK} \\ &= 20 \times 10^{-3} \times 2 \times 10^6 = 40000\end{aligned}$$

- 要求2: 产生 1kHz 的方波

$$\begin{aligned}n &= f_{CLK}/f_{OUT} \\ &= 2 \times 10^6 / 1 \times 10^3 = 2000\end{aligned}$$

8253A定时/计数器是\_\_\_\_位的减1计数器，采用二进制计数时，写入计数初值为\_\_\_\_时计数值最大，最大值为\_\_\_\_。（ ）

- ☐ A 8, 0FFH, 255
- ☐ B 8, 0, 256
- ☒ C 16, 0, 65536
- ☐ D 16, 0FFFFH, 65535

# 01 | 定时/计数器-子目录

## 接口概述

- 定时/计数器的概念
- 8253A 可编程定时/计数器
  - 8253A芯片的工作特点
  - 8253A芯片的外部引脚特性
  - 8253A的内部结构
  - 计数初值的设置
- 8253A 控制字和工作方式
- 8253A 的编程应用

# 01 | 定时/计数器-8253A控制字

## 8253A 的控制字

- 由 8 位控制寄存器的值给出，每一位都预定不同的含义
- 控制字由 CPU 经过数据总线缓冲器写入，不同的控制字决定**相应的**计数器工作在不同**工作方式**下

## 8253A 的工作方式控制字的作用

- 初始化操作中使用
  - 按计数通道的使用**设置工作方式控制字**
  - 将计数初值**写入对应的计数通道**
- 读通道中的当前计数值
  - 使用工作方式控制字，**设置对应通道的锁存命令**
  - **读对应通道**，得到当前计数值

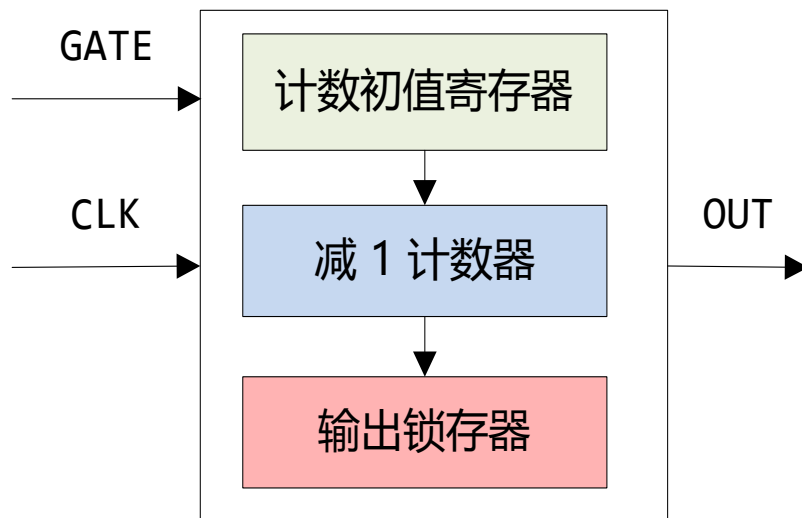
# 01 | 定时/计数器-8253A对计数通道的访问类型

## 写计数通道

- 写计数通道的**初值寄存器**

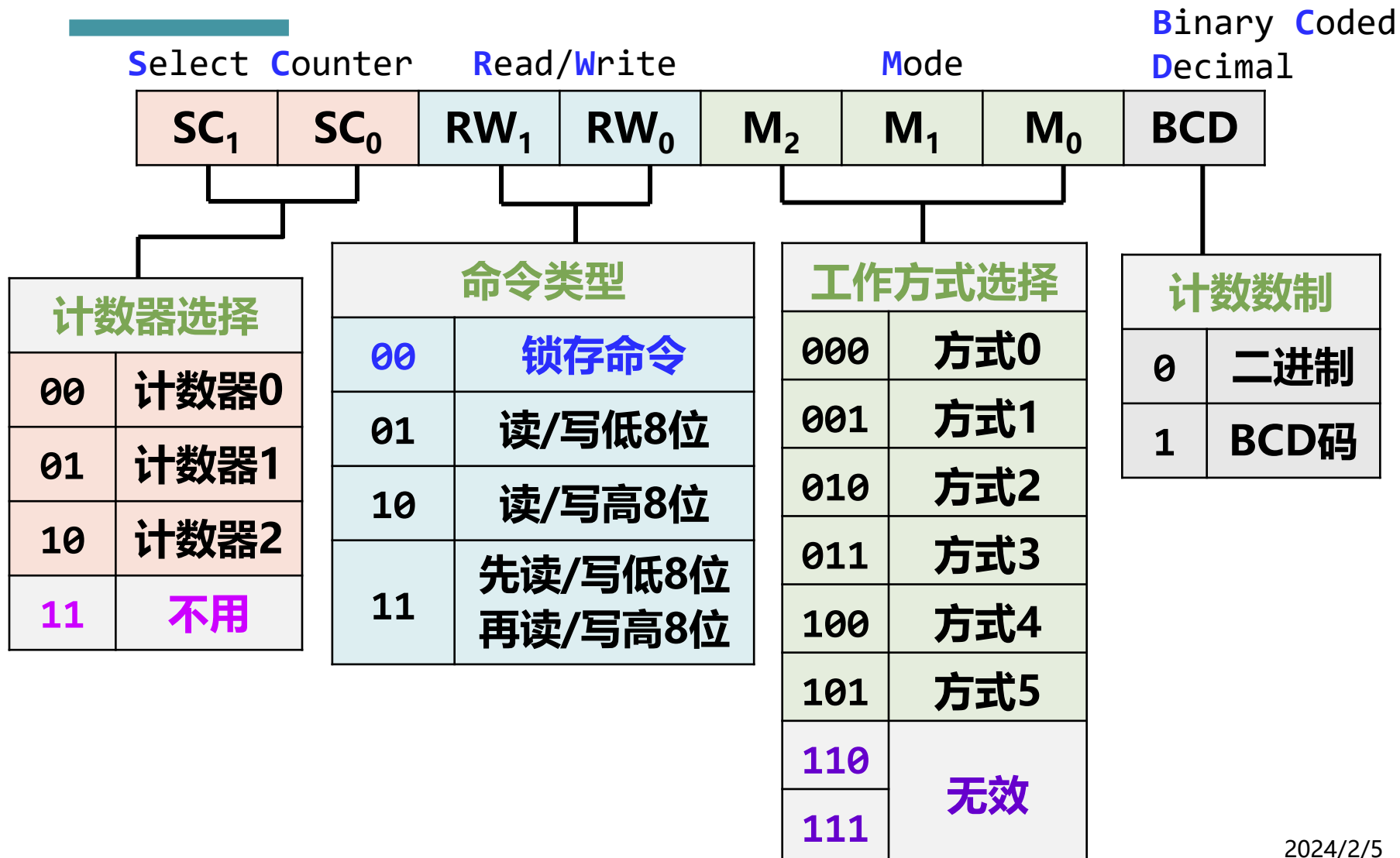
## 读计数通道

- 读计数通道的**初值寄存器**
  - 直接读取计数通道
- 读计数通道的**当前计数值**
  - 先使用**工作方式控制字**发锁存命令，再读取计数通道



# 01 | 定时/计数器-8253A控制字

## 8253A 的控制字





# 01 | 定时/计数器-8253A控制字举例1

示例1：选择计数通道 2，工作在方式 3 下，计数初值为 533H，采用二进制计数，试写出其初始化程序段

## ● 工作方式控制字

| SC <sub>1</sub> | SC <sub>0</sub> | RW <sub>1</sub> | RW <sub>0</sub> | M <sub>2</sub> | M <sub>1</sub> | M <sub>0</sub> | BCD |
|-----------------|-----------------|-----------------|-----------------|----------------|----------------|----------------|-----|
| 1               | 0               | 1               | 1               | 0              | 1              | 1              | 0   |

## ● 初始化程序段

### ■ 写入命令字

控制端口  
控制字

```
MOV DX, 307H
MOV AL, 0B6H
OUT DX, AL
```

### ■ 写入计数初值 (先低字节，再高字节)

```
MOV DX, 306H
```

计数通道 2 端口

```
MOV AL, 33H
```

计数初值低 8 位

```
OUT DX, AL
```

```
MOV AL, 05H
```

计数初值高 8 位

```
OUT DX, AL
```

# 01 | 定时/计数器-举例1初始化程序执行过程

示例1：选择计数通道 2，工作在方式 3 下，计数初值为 533H，采用二进制计数，试写出其初始化程序段

## ■ 写入命令字

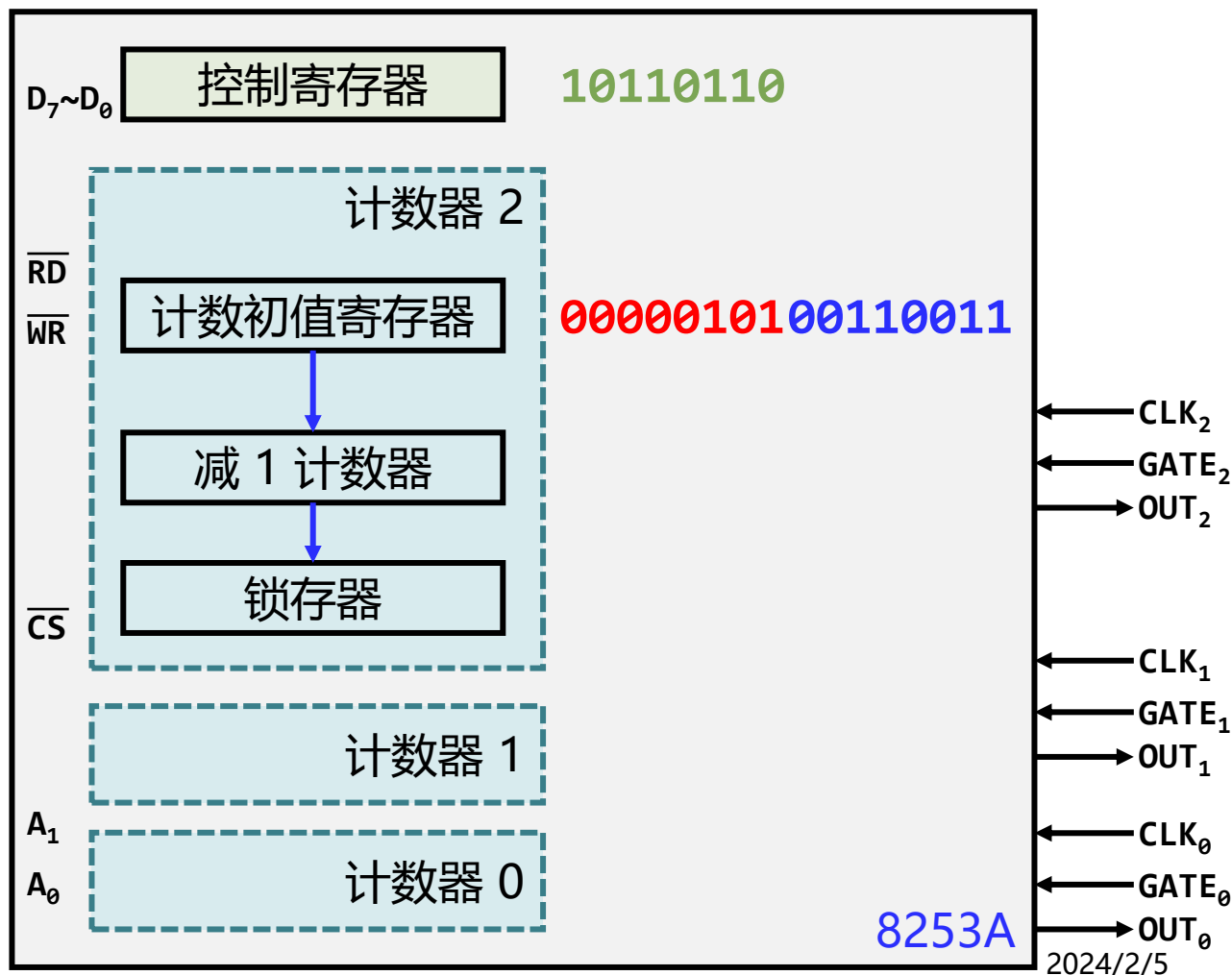
```
MOV DX, 307H
MOV AL, 0B6H
OUT DX, AL
```

## ■ 写入计数初值

```
MOV DX, 306H

MOV AL, 33H
OUT DX, AL

MOV AL, 05H
OUT DX, AL
```



# 01 | 定时/计数器-8253A控制字举例2

示例2：选择计数通道 1，工作在方式 0 下，计数初值为 128，采用二进制计数，试写出其初始化程序段

## ● 工作方式控制字

| SC <sub>1</sub> | SC <sub>0</sub> | RW <sub>1</sub> | RW <sub>0</sub> | M <sub>2</sub> | M <sub>1</sub> | M <sub>0</sub> | BCD |
|-----------------|-----------------|-----------------|-----------------|----------------|----------------|----------------|-----|
| 0               | 1               | 0               | 1               | 0              | 0              | 0              | 0   |

## ● 初始化程序段

### ■ 写入命令字

控制端口  
控制字

```
MOV DX, 307H
MOV AL, 50H
OUT DX, AL
```

### ■ 写入计数初值

计数通道 1 端口  
计数初值

```
MOV DX, 305H
MOV AL, 128
OUT DX, AL
```

# 01 | 定时/计数器-8253A控制字举例3

示例3：选择计数通道 0，工作在方式 1 下，计数初值为 3000，采用 BCD 码计数，试写出其初始化程序段

## ● 工作方式控制字

| SC <sub>1</sub> | SC <sub>0</sub> | RW <sub>1</sub> | RW <sub>0</sub> | M <sub>2</sub> | M <sub>1</sub> | M <sub>0</sub> | BCD |
|-----------------|-----------------|-----------------|-----------------|----------------|----------------|----------------|-----|
| 0               | 0               | 1               | 1               | 0              | 0              | 1              | 1   |

## ● 初始化程序段

### ■ 写入命令字

控制端口  
控制字

```
MOV DX, 307H
MOV AL, 33H
OUT DX, AL
```

### ■ 写入计数初值 (先低字节，再高字节)

```
MOV DX, 304H

MOV AX, 3000
OUT DX, AL
MOV AL, AH
OUT DX, AL
```

计数通道 0 端口

计数初值低 8 位

计数初值高 8 位

# 01 | 定时/计数器-8253A控制字举例4

示例4：读取 1 号计数器的当前计数值，并判断是否为全 1。  
假定初始化的工作方式控制字的 RW=11

- 读当前计数值操作：
  - 先发**锁存命令**，锁存当前计数值
  - 再用**输入指令**从计数器的当前计数值锁存器中**读取数据**

- 命令字：

| SC <sub>1</sub> | SC <sub>0</sub> | RW <sub>1</sub> | RW <sub>0</sub> | M <sub>2</sub> | M <sub>1</sub> | M <sub>0</sub> | BCD |
|-----------------|-----------------|-----------------|-----------------|----------------|----------------|----------------|-----|
| 0               | 1               | 0               | 0               | 0              | 0              | 0              | 0   |

- 初始化程序段

- 写入**命令字**

```
MOV DX, 307H
MOV AL, 40H
OUT DX, AL
```

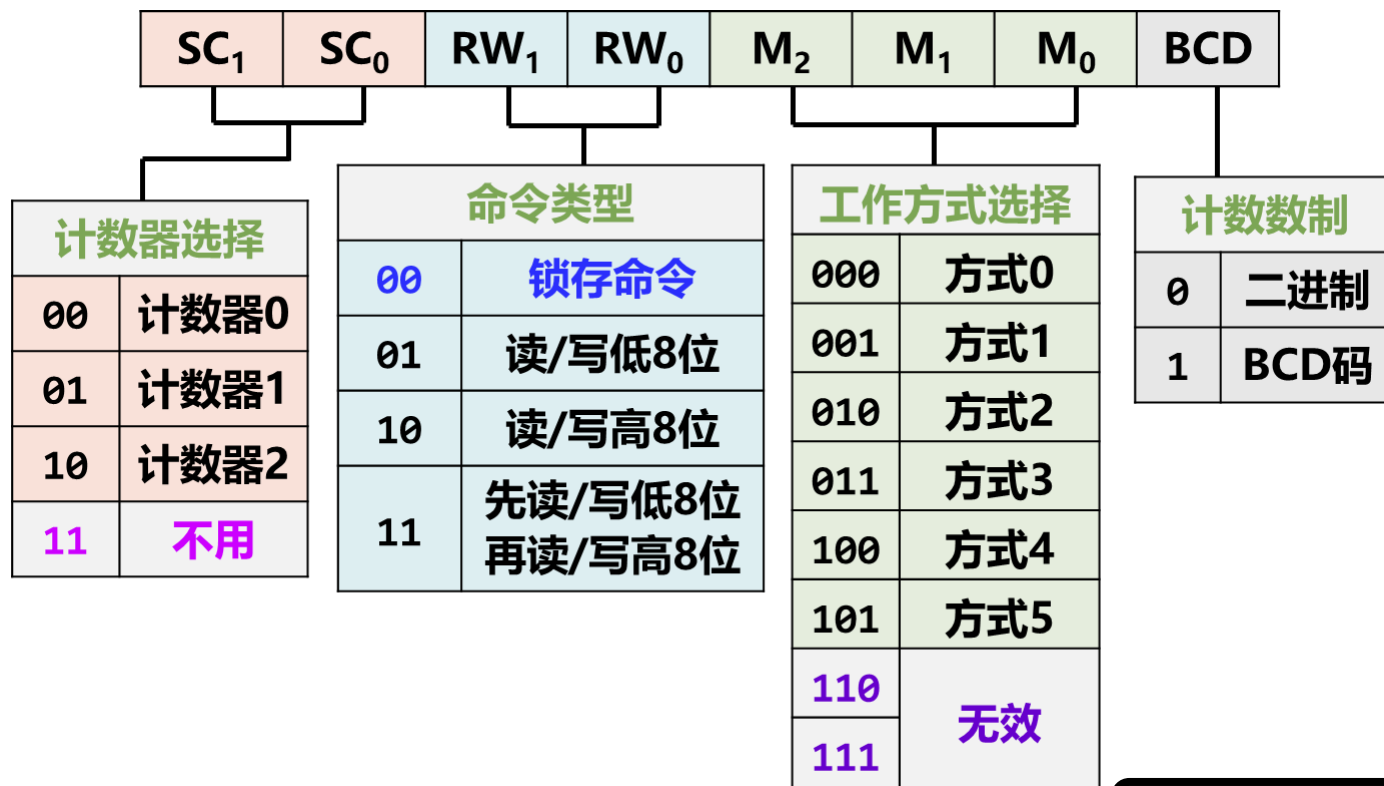
- 读**计数值**

```
MOV DX, 305H
IN AL, DX
MOV AH, AL
IN AL, DX
XCHG AL, AH
```

- 判断处理

```
CMP AX, 0FFFFH
JE YES
.....
.....
```

8253A芯片选择计数通道 1，工作在方式 2 下，计数初值为 18，采用二进制计数，则其控制字应为 [填空1]



# 01 | 定时/计数器-8253A的工作方式

## 工作方式

- 8253A 共有 **6 种工作方式**，三个计数通道都可以工作在这 6 种工作方式下

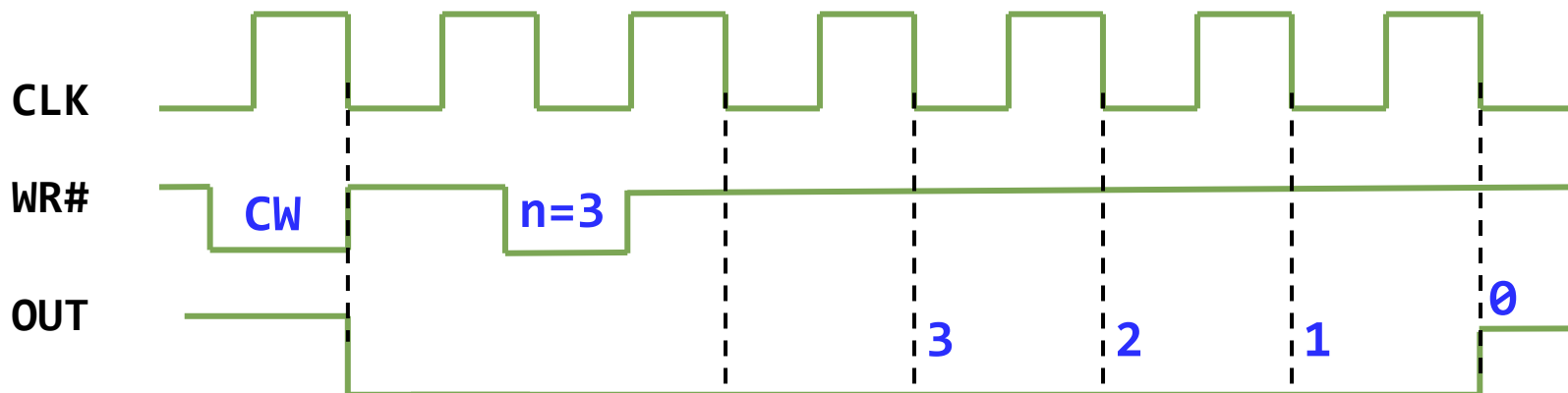
## 学习要点

- **输出波形**
- **计数的触发方式**
  - 软件触发、硬件触发、软硬触发
- 计数过程中，**门控信号 GATE** 的变化对计数的影响
  - 不影响、停止计数、暂停计数、重新计数
- **重装计数初值后**，对计数过程的影响
  - 不影响、立刻生效、下次计数生效

# 01 | 定时/计数器-方式0-计数结束产生正脉冲

## 工作方式 0

- 控制字设置计数器为方式 0 后，输出端 OUT 立即变为**低电平**
- 写入计数初值后，经过一个时钟周期，将计数初值送入减 1 计数器**开始计数**，OUT 输出维持低电平
- 直至**计数值变为 0**，OUT 输出端变为**高电平**，并维持到下次使用计数器

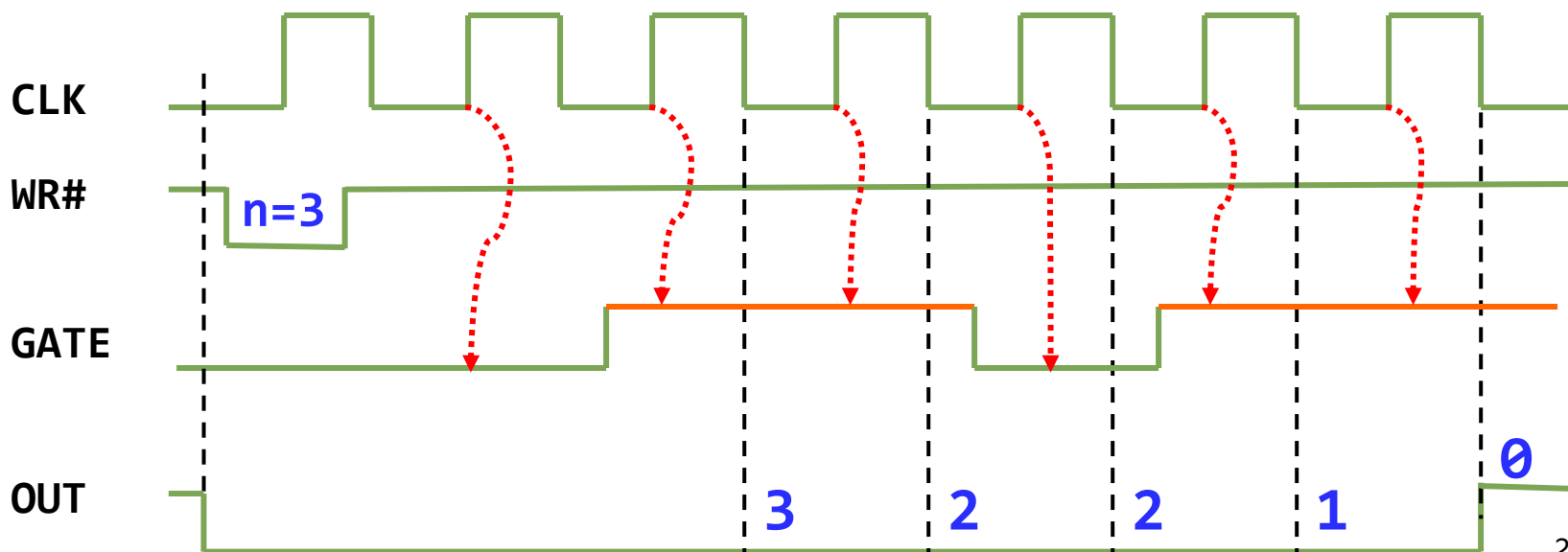




# 01 | 定时/计数器-方式0-计数结束产生正脉冲

## 门控信号的作用

- 门控信号 GATE 对计数起**开关作用**
  - 即当 GATE=1 时，允许计数；当 GATE=0 时，禁止计数
- 计数过程中，GATE 变为**低电平**，**暂停计数**；当 GATE 再次变为**高电平**时，从中止计数处**继续**开始计数
- GATE 的变化并不影响 OUT 端的输出状态

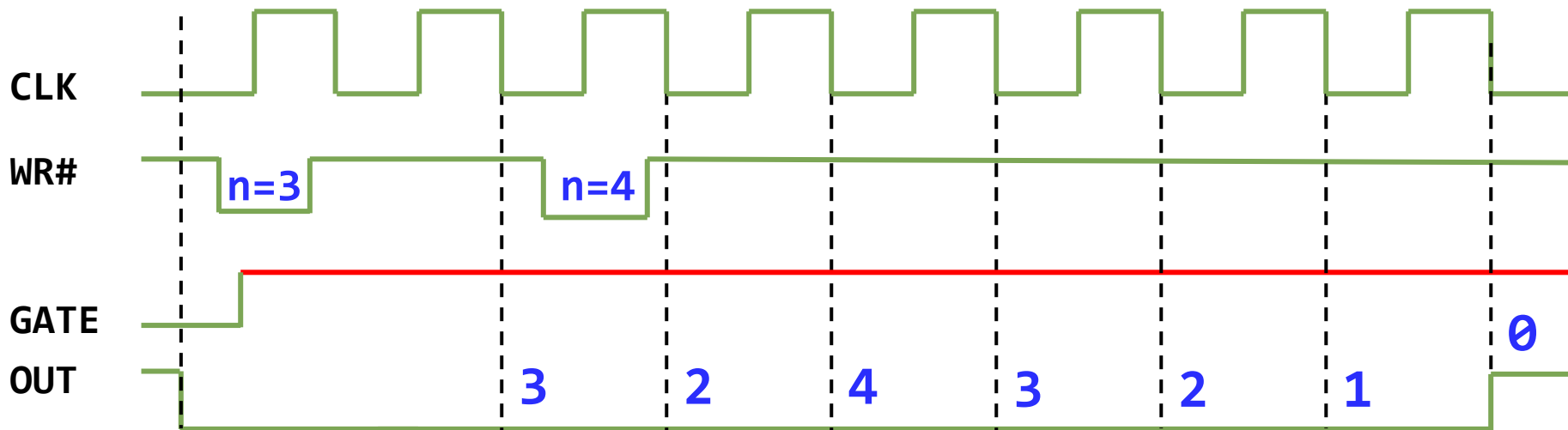


# 01 | 定时/计数器-方式0-计数结束产生正脉冲

## 改变计数初值的影响

- 计数过程中，若计数初值发生变化，则计数器会在下一时钟周期将新的计数值写入减 1 计数器，并重新开始计数

■ 方式 0 是软件触发（写计数初值的 OUT 指令）的计数方式

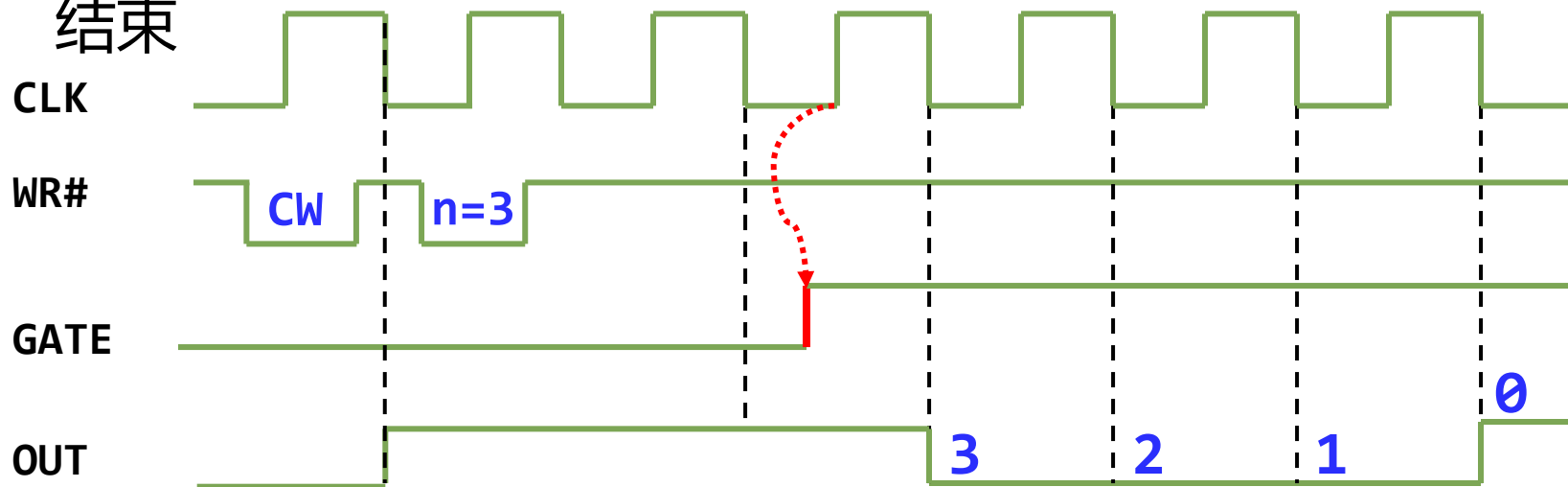


- 方式 0 的计数初值只有一**次有效**
  - 若要再次计数，即使计数值不变，也需重新写入计数初值

# 01 定时/计数器-方式1-可重复触发的单稳态脉冲

## 工作方式 1：硬件可重复触发的单脉冲发生器

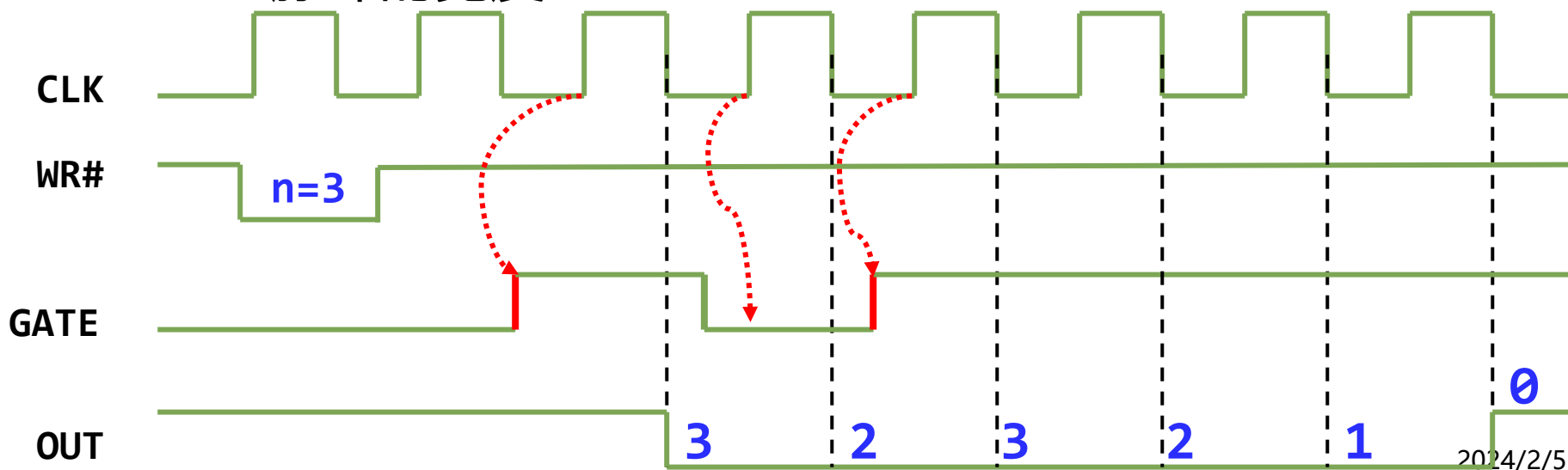
- 当命令字设定方式 1 后，输出端 OUT 以**高电平**为起始电平
- 写入计数初值后，经过一个时钟周期，将计数初值送入减 1 计数器，但**并不开始计数**
- 当门控信号 **GATE 出现上升沿**，开始计数；计数开始的同时，**OUT 输出端变为低电平**，作为单脉冲的开始
- 直到计数值为 0，**OUT 输出变回高电平**，计数结束，即单脉冲结束



# 01 定时/计数器-方式1-可重复触发的单稳态脉冲

## 门控信号的作用

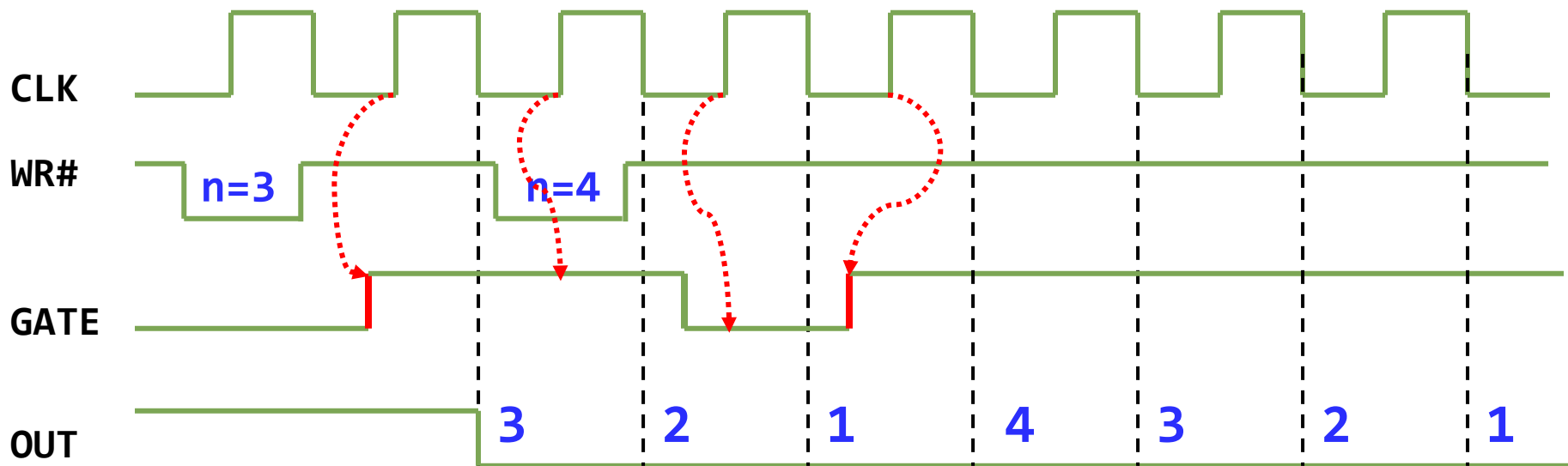
- GATE 信号对计数过程起**触发作用**
  - GATE 的**上升沿**会使计数过程重新开始；但其高、低电平及下降沿不影响计数。
  - 实质上，GATE 的上升沿使计数初值寄存器将当前的**计数初值再次装入减 1 计数器**中
- GATE 变化不影响 OUT 的输出状态，但会**影响**输出单脉冲的宽度



# 01 定时/计数器-方式1-可重复触发的单稳态脉冲

## 改变计数初值的影响

- 计数初值**多次有效**，即该方式下计数过程可以重复被触发
- 计数过程中，若计数初值发生变化，则**当 GATE 再次出现上升沿时，立即以新的计数初值开始计数**；否则，计数初值的改变不会影响当前的计数过程



# 01 | 定时/计数器-方式0与方式1的比较

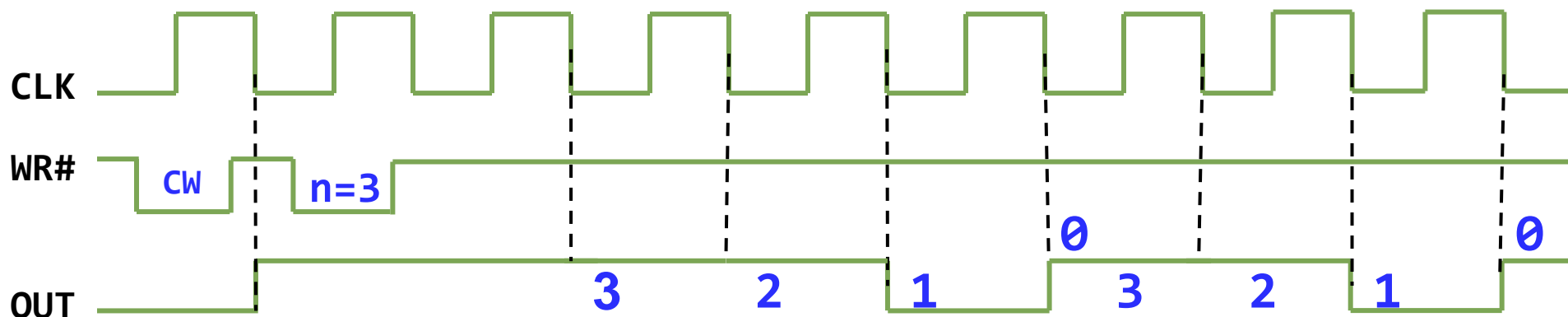
## 方式 0 与方式 1 的比较

- 输出波形大致相同
  - 在一次计数过程中输出**单次负脉冲**
- 触发方式不同
  - 方式 0 — 软件触发
  - 方式 1 — 硬件触发
- 方式 0 主要使用**计数结束的上升沿脉冲**作为请求信号使用；方式 1 主要用作**定时器**使用
  - 方式 1 是由硬件触发计数的，因此**方式 1 输出的负脉冲宽度是可以由计数初值精确计算的**，而方式 0 不可

# 01 | 定时/计数器-方式2-分频器

## 工作方式 2

- 当命令字设置方式 2 后，OUT 输出端的初始状态为**高电平**
- 装入计数初值后，**开始计数**，输出端仍维持高电平
- 直到计数值为 1 时，OUT 输出**一个时钟周期的低电平**，一次计数结束，输出恢复高电平
- 然后**自动重装**计数初值，再次进行同样的计数，从而产生连续的脉冲序列

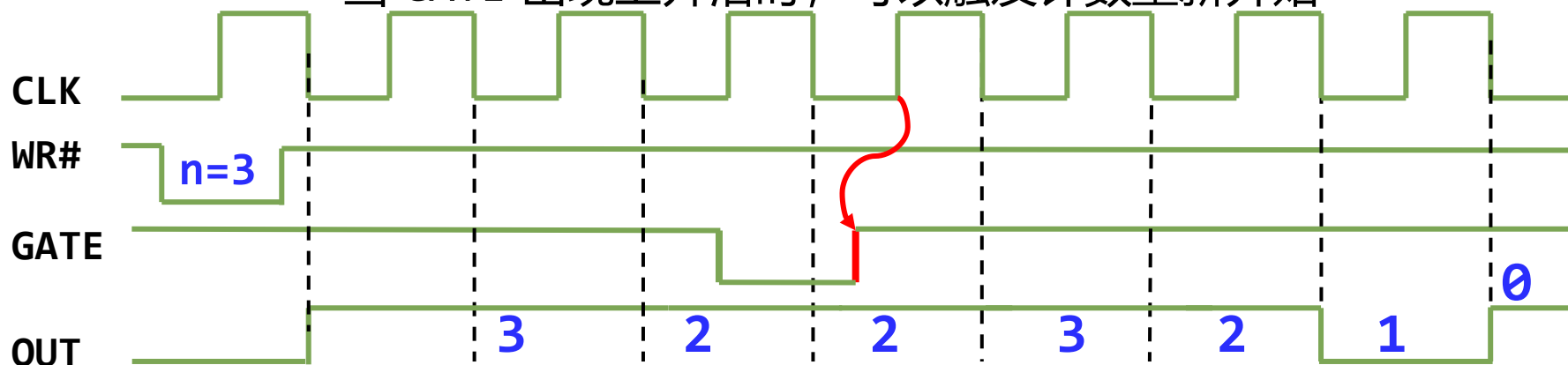


# 01 | 定时/计数器-方式2-分频器

## 门控信号的作用 和 改变计数初值的影响

- 门控信号 GATE 对计数过程有**开关作用**，也有**触发作用**

- 当 GATE=1 时，允许计数；当 GATE=0 时，禁止计数
- 当 GATE 出现上升沿时，可以触发计数重新开始



- 计数过程中，仅计数初值发生变化
  - 若**没有 GATE 上升沿**，则不影响**本次的**计数过程；只是在当前计数归 0，自动装入新的计数初值开始新的计数
  - 若**出现 GATE 上升沿**，则在**下个时钟周期立刻装入**新的计数初值开始计数

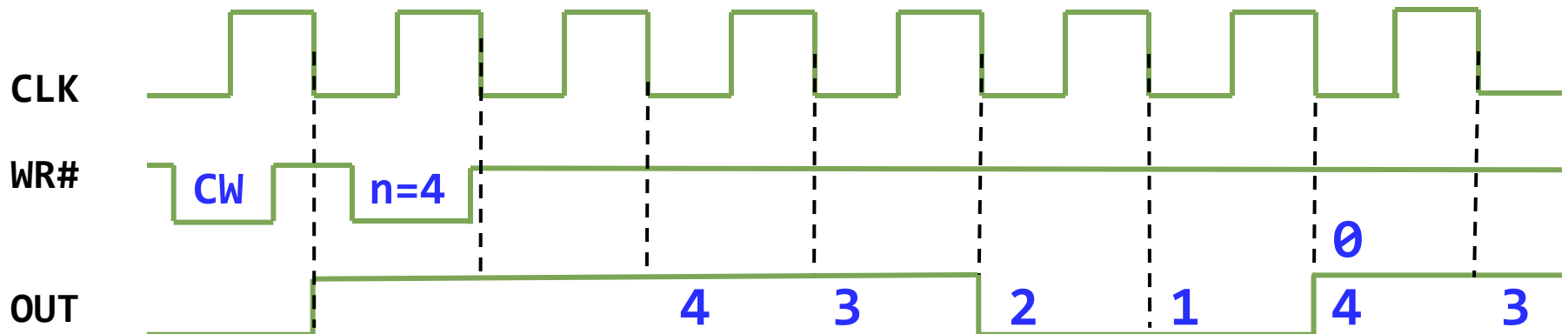


# 01 | 定时/计数器-方式3-方波发生器

## 工作方式 3

输出方波或基本对称的矩形波，常作为某些慢速外设的时钟输入信号

- 当命令字设置方式 3 后，OUT 输出端的初始状态为**高电平**
- 装入计数初值后，开始计数，输出端仍维持高电平；
- 当**计数到一半**时，OUT 输出变为**低电平**，直到计数值为 0，恢复高电平的输出
- 然后**自动重装**计数初值，再次进行同样的计数，从而产生连续的方波序列



# 01 | 定时/计数器-方式3-方波发生器

## 门控信号的作用 和 改变计数初值的影响

- 门控信号 GATE 对计数过程起开关作用和触发作用，同方式 2
- 计数初值的一半：
  - 当计数初值为**偶数**时，计数到  $n/2$ ，输出端 OUT 发生负跳变，得到占空比为 1:1 的方波
  - 当计数初值为**奇数**时，计数到  $(n+1)/2$ ，输出端 OUT 发生负跳变，得到占空比为  $(n+1)/2 : (n-1)/2$  的矩形波
- 计数过程中写入新的计数初值，则会在**半周期结束**或**门控脉冲触发**时，重新装入新值，开始计数

# 01 | 定时/计数器-方式2和方式3的比较

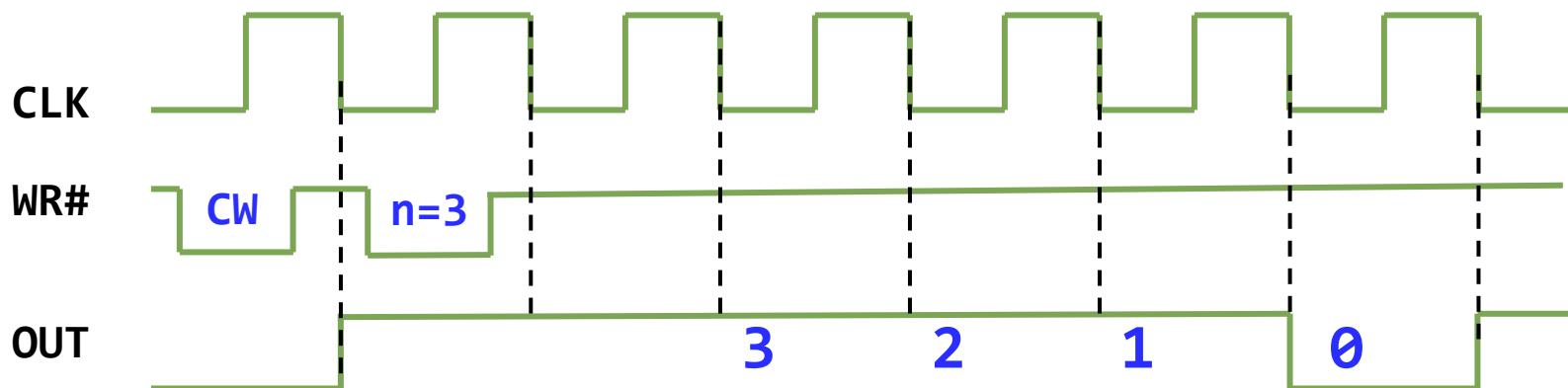
## 方式 2 和方式 3 的比较

- 方式 2 主要用于产生**周期性负脉冲**  
方式 3 主要用于产生**方波或矩形波**
- 方式 2 和方式 3 的计数触发方式均为**软硬件触发**
  - **软件**触发方法 —— 重新写入计数初值
    - ◆ 在一个完整的计数周期 (方式2) 或半周期 (方式3) 输出完成后, 按照新的计数初值开始计数
  - **硬件**触发方法 —— 门控信号出现上升沿
    - ◆ 立即 (下个时钟周期) 触发新的计数过程, 即输出脉冲序列的频率随 GATE 信号的变化而随时发生改变

# 01 定时/计数器-方式4-软件触发的单稳态脉冲

## 工作方式 4：软件触发选通的单脉冲发生器

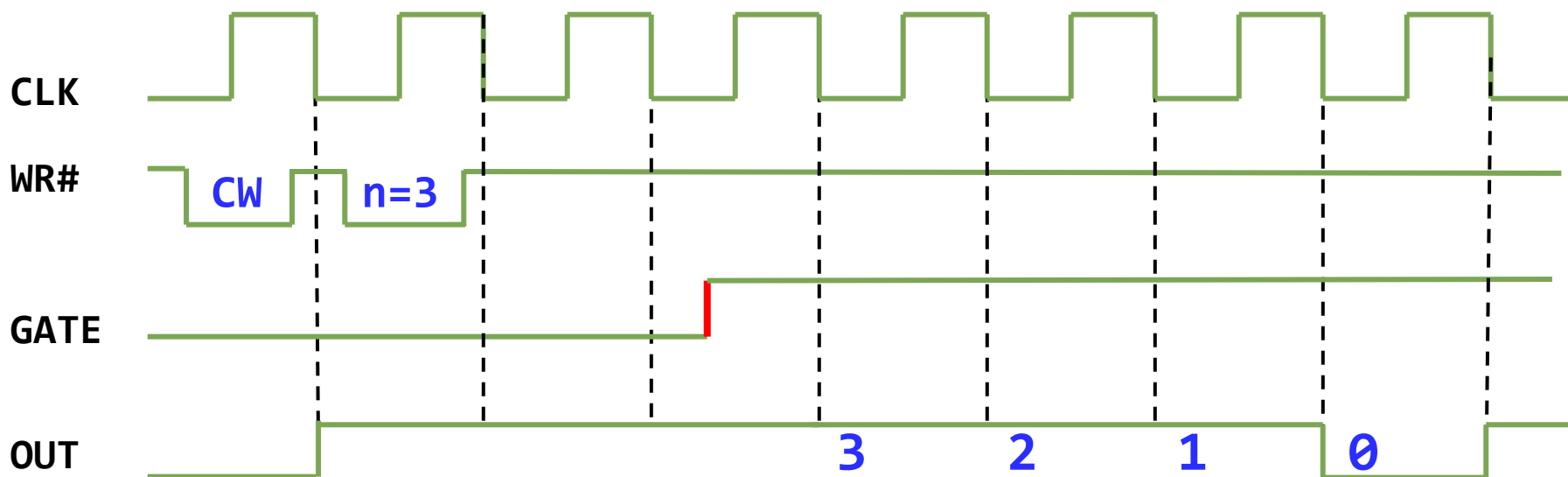
- 当用命令字设置方式 4 后，OUT 输出端的初始状态为高电平
- 装入计数初值后，即开始计数
- 在计数过程中，输出端始终维持高电平
- 当计数为 0，OUT 输出一个时钟周期的负脉冲，结束计数



# 01 定时/计数器-方式5-硬件触发的单稳态脉冲

## 工作方式 5：硬件触发选通的单脉冲发生器

- 当命令字设置方式 5 后，OUT 输出端的初始状态为**高电平**
- 装入计数初值后，并不立即开始计数
- 当 **GATE 出现上升沿**时，开始减 1 计数，OUT 维持高电平
- 当**计数为 0** 时，OUT 输出**一个时钟周期的负脉冲**，结束计数



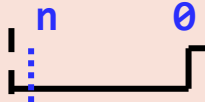

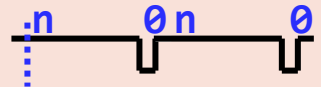
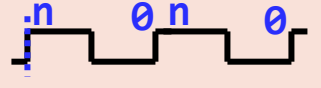


# 01 | 定时/计数器-方式4和方式5的比较

## 方式 4 和方式 5 的比较

- 输出波形完全相同
  - 一次计数得到**单次单拍负脉冲**
- 计数的触发方式不同
  - 方式 4 —— **软件**触发
    - ◆ 触发方式、GATE 和计数初值改变时，对计数的影响同**方式 0**
  - 方式 5 —— **硬件**触发
    - ◆ 触发方式、GATE 和计数初值改变时，对计数的影响同**方式 1**

# 01 | 定时/计数器-各种工作方式的比较1

## 各种工作方式的**功能**

| 方式 | 功能        | 输出波形描述  | 输出波形  |
|----|-----------|---------|---|
| 0  | 计数结束产生正脉冲 | 单次负脉冲   |    |
| 1  | 单脉冲发生器    | 单次负脉冲   |    |
| 2  | 分频器       | 连续脉冲波   |    |
| 3  | 方波发生器     | 连续方波    |   |
| 4  | 软件触发选通    | 单次单拍负脉冲 |  |
| 5  | 硬件触发选通    | 单次单拍负脉冲 |  |

# 01 | 定时/计数器-各种工作方式的比较2

## 计数器启动和重复计数的条件

| 方式 | 功能        | 启动条件  | 重复条件      |
|----|-----------|-------|-----------|
| 0  | 计数结束产生正脉冲 | 软件触发  | 写计数值      |
| 1  | 单脉冲发生器    | 硬件触发  | GATE 信号触发 |
| 2  | 分频器       | 软硬件触发 |           |
| 3  | 方波发生器     | 软硬件触发 |           |
| 4  | 软件触发选通    | 软件触发  | 写计数值      |
| 5  | 硬件触发选通    | 硬件触发  | GATE 信号触发 |



# 01 | 定时/计数器-各种工作方式的比较3

## 门控信号 GATE 的作用

| 方式 | GATE 信号的状态 |       |       |
|----|------------|-------|-------|
|    | 低电平或下降沿    | 上升沿   | 高电平   |
| 0  | 暂停计数       | 不影响计数 | 允许计数  |
| 1  | 不影响计数      | 重新计数  | 不影响计数 |
| 2  | 停止计数       | 重新计数  | 允许计数  |
| 3  | 停止计数       | 重新计数  | 允许计数  |
| 4  | 暂停计数       | 不影响计数 | 允许计数  |
| 5  | 不影响计数      | 重新计数  | 不影响计数 |

# 01 | 定时/计数器-各种工作方式的比较4

## 计数过程中改变计数初值的结果

| 方式 | 功能        | 改变计数初值                                |
|----|-----------|---------------------------------------|
| 0  | 计数结束产生正脉冲 | 立即有效                                  |
| 1  | 单脉冲发生器    | 门控信号触发后有效                             |
| 2  | 分频器       | 当前计数周期结束 (计数到1后) 或门控信号触发后有效           |
| 3  | 方波发生器     | 当前半个计数周期结束 (计数到一半或计数到 0 后) 或门控信号触发后有效 |
| 4  | 软件触发选通    | 立即有效                                  |
| 5  | 硬件触发选通    | 门控信号触发后有效                             |

能够输出方波的8253A工作方式是（ ）

- ☐ A 方式0
- ☐ B 方式1
- ☐ C 方式2
- ☒ D 方式3

# 01 | 定时/计数器-子目录

## 接口概述

- 定时/计数器的概念
- 8253A 可编程定时/计数器
  - 8253A芯片的工作特点
  - 8253A芯片的外部引脚特性
  - 8253A的内部结构
  - 计数初值的设置
- 8253A 控制字和工作方式
- 8253A 的编程应用
  - 微机系统中的应用
  - 两级计数例子
  - 波特率时钟发生器例子

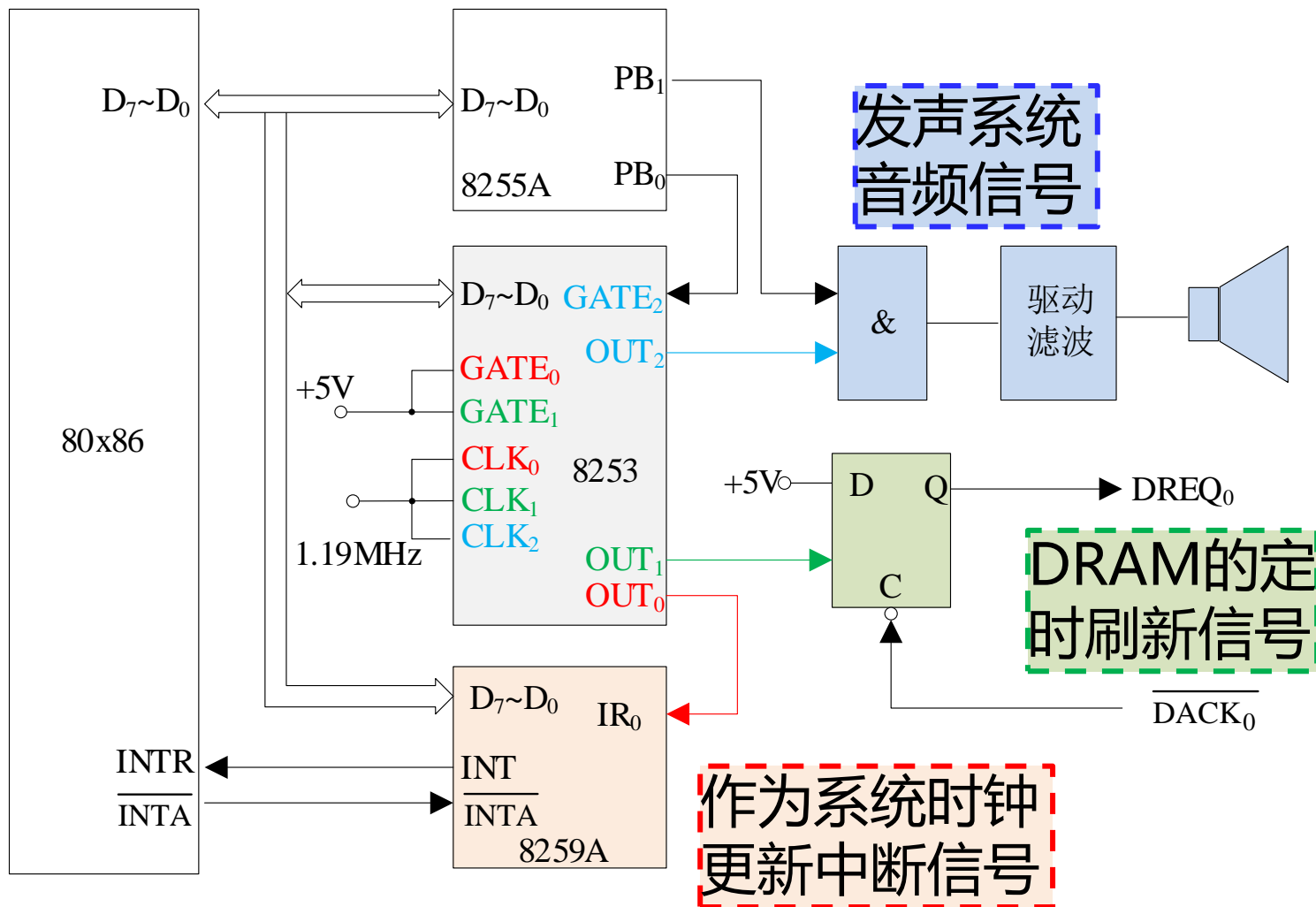
# 01 | 定时/计数器-应用1-微机系统中的应用

## 微机系统中的应用

- **通道 0** —— 系统时钟更新中断信号
  - 工作于**方式 3**，计数初值为 **0**
- **通道 1** —— 动态存储器的定时刷新信号
  - 工作于**方式 2**，计数初值为 **18**
- **通道 2** —— 发声系统音频信号
  - 工作于**方式 3**，计数初值为 **533H**

# 01 | 定时/计数器-应用1-微机系统中的应用

## PC 系列机中定时系统结构框图



# 01 | 定时/计数器-通道0-系统时钟更新中断信号

## 微机系统中的应用

- 工作于**方式 3**，计数初值为 **0**
- $GATE_0$  接 +5 V， $CLK_0$  输入时钟频率为 1.1931816 MHz
  - $OUT_0$  输出频率为 **18.2 Hz 的方波**
  - 输出脉冲周期= $1/18.2 \approx 55ms$
- $OUT_0$  连接 8259A 的  $IRQ_0$  上，**约 55ms 产生 1 次中断**，用于更新系统的日时钟计数
  - 系统直接调用固化在 BIOS 中的中断类型为 08H 的中断服务程序
- 初始化程序如下：

```
MOV AL, 00110110B
OUT 43H, AL
MOV AL, 0
OUT 40H, AL
OUT 40H, AL
```

| SC <sub>1</sub> | SC <sub>0</sub> | RW <sub>1</sub> | RW <sub>0</sub> | M <sub>2</sub> | M <sub>1</sub> | M <sub>0</sub> | BCD |
|-----------------|-----------------|-----------------|-----------------|----------------|----------------|----------------|-----|
| 0               | 0               | 1               | 1               | 0              | 1              | 1              | 0   |

# 01 定时/计数器-通道1-动态存储器的定时刷新信号

## 微机系统中的应用

- 工作于**方式 2**，计数初值为 **18**
- $GATE_1$  接 +5 V， $CLK_1$  输入时钟频率为 1.1931816 MHz
  - $OUT_1$  每隔 **15.08  $\mu$ s** 输出一个负脉冲
- $OUT_1$  连接 8237 的  $DREQ_0$  上
  - 8237 通道 0 用于控制 DRAM 的刷新
  - 8253A 的通道 1 定时产生 DRAM 的刷新请求信号，每个周期刷新存储器一行
- 初始化程序如下：

```
MOV AL, 01010100B
OUT 43H, AL
MOV AL, 18
OUT 41H, AL
```

| $SC_1$ | $SC_0$ | $RW_1$ | $RW_0$ | $M_2$ | $M_1$ | $M_0$ | BCD |
|--------|--------|--------|--------|-------|-------|-------|-----|
| 0      | 1      | 0      | 1      | 0     | 1     | 0     | 0   |



# 01 | 定时/计数器-通道2-发声系统音频信号

## 微机系统中的应用

- 工作于**方式 3**，计数初值为 **533H**
  - $GATE_2$  由 8255A 的  $PB_0$  位控制， $CLK_2$  输入时钟频率为 1.1931816 MHz
    - $OUT_2$  输出频率**约为 900 Hz** 的方波
  - 输出信号与 8255 的  $PB_1$  位（控制发声）共同经过滤波，作为系统扬声器发声的音频信号，进行提示和故障报警
- 初始化程序如下：

```
MOV AL, 10110110B
OUT 43H, AL
MOV AX, 533H
OUT 42H, AL
MOV AL, AH
OUT 42H, AL
```

| SC <sub>1</sub> | SC <sub>0</sub> | RW <sub>1</sub> | RW <sub>0</sub> | M <sub>2</sub> | M <sub>1</sub> | M <sub>0</sub> | BCD |
|-----------------|-----------------|-----------------|-----------------|----------------|----------------|----------------|-----|
| 1               | 0               | 1               | 1               | 0              | 1              | 1              | 0   |

# 01 | 定时/计数器-通道2-BIOS的初始化编程

## BIOS 中对计数通道 2 的初始化编程

```
BEEP PROC FAR
```

```
IN AL, 61H
```

```
MOV AH, AL
```

```
PUSH AX
```

读8255 PB  
口并保存

```
MOV AL, 10110110B
```

方式3写  
16位初值

```
OUT 43H, AL
```

```
MOV AX, 0533H
```

```
OUT 42H, AL
```

```
MOV AL, AH
```

```
OUT 42H, AL
```

```
POP AX
```

OUT<sub>2</sub>输出频率  
为1.1931816  
M/1331=896  
Hz的方波

```
OR AL, 03H
```

```
OUT 61H, AL
```

置PB<sub>0</sub>、PB<sub>1</sub>为  
1, 打开扬声器

```
NOP
```

```
L1: LOOP L1
```

由CX决定延时长短

```
AND AL, 0FDH
```

```
OUT 61H, AL
```

置PB<sub>0</sub>、PB<sub>1</sub>  
为0, 封锁  
OUT<sub>2</sub>输出

```
MOV AL, AH
```

```
OUT 61H, AL
```

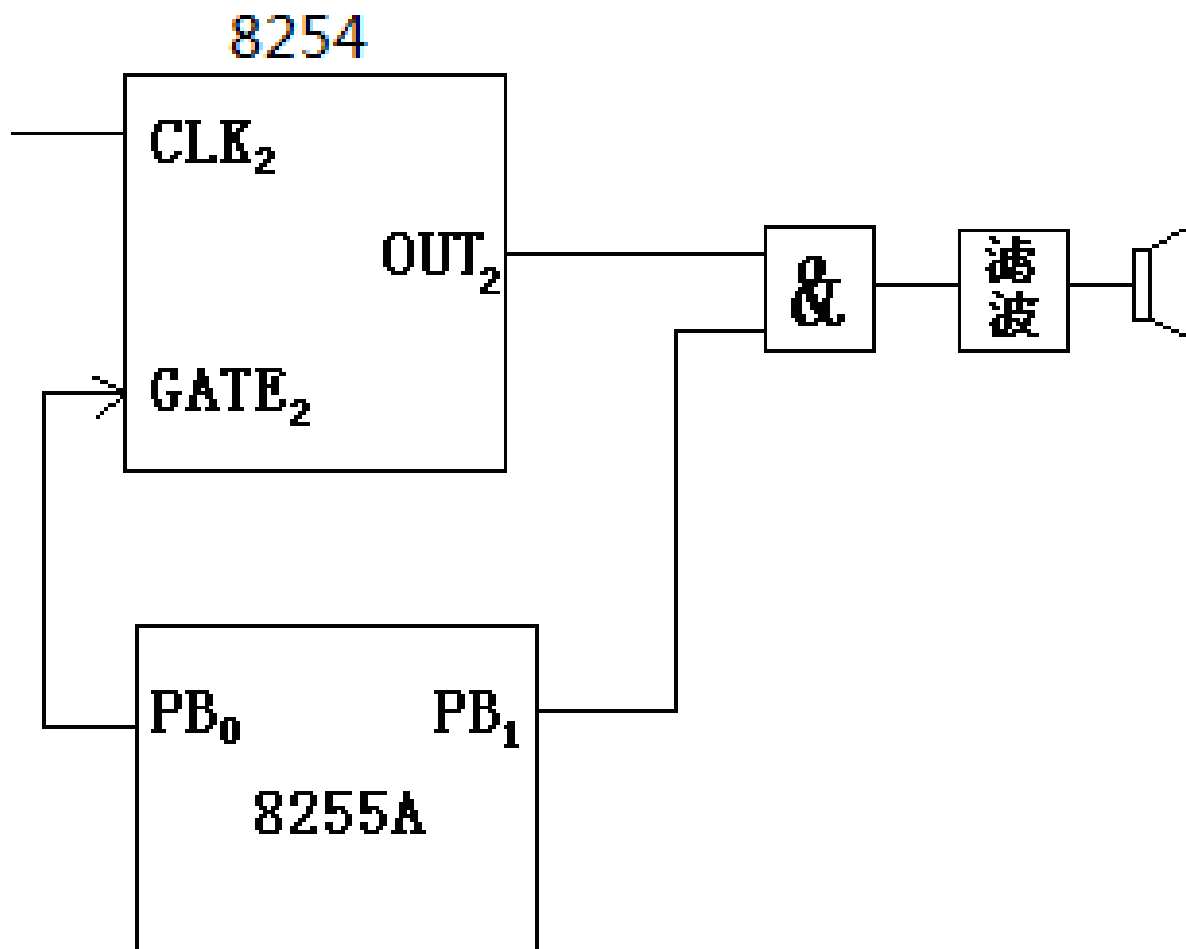
恢复原PB端  
口值

```
RET
```

```
BEEP ENDP
```

# 01 | 定时/计数器-通道2-发声系统连接图

## 发声系统连接图



# 01 | 定时/计数器-通道2-发声系统编程

## 使用入口参数 - 指定扬声器的发声频率和发声持续时间

; BIOS中对应程序  
; 功能: 按照指定的时间间隔发指定频率的声音  
; 入口参数: CX=指定频率, BX=指定时间  
; 返回: 无

SOUND PROC FAR

MOV AL, 10110110B

OUT 43H, AL

MOV AX, CX

OUT 42H, AL

MOV AL, AH

OUT 42H, AL

方式3写  
16位初值

向通道2写入指  
定频率(CX)

IN AL, 61H

MOV AH, AL

OR AL, 03H

OUT 61H, AL

读8255 PB口  
并保存

置PB<sub>0</sub>、PB<sub>1</sub>为  
1, 打开扬声器

L1: MOV CX, 5000H

L2: LOOP L2

DEC BX

JNZ L1

MOV AL, AH

OUT 61H, AL

RET

SOUND ENDP

指定时间  
延时

恢复原PB端  
口值

# 01 | 定时/计数器-子目录

## 接口概述

- 定时/计数器的概念
- 8253A 可编程定时/计数器
  - 8253A芯片的工作特点
  - 8253A芯片的外部引脚特性
  - 8253A的内部结构
  - 计数初值的设置
- 8253A 控制字和工作方式
- 8253A 的编程应用
  - 微机系统中的应用
  - 两级计数例子
  - 波特率时钟发生器例子

# 01 | 定时/计数器-应用2-两级计数

利用 8253A 设计一个定时器，每 5 秒钟输出一个负脉冲，  
设外部时钟频率为 2.5MHz

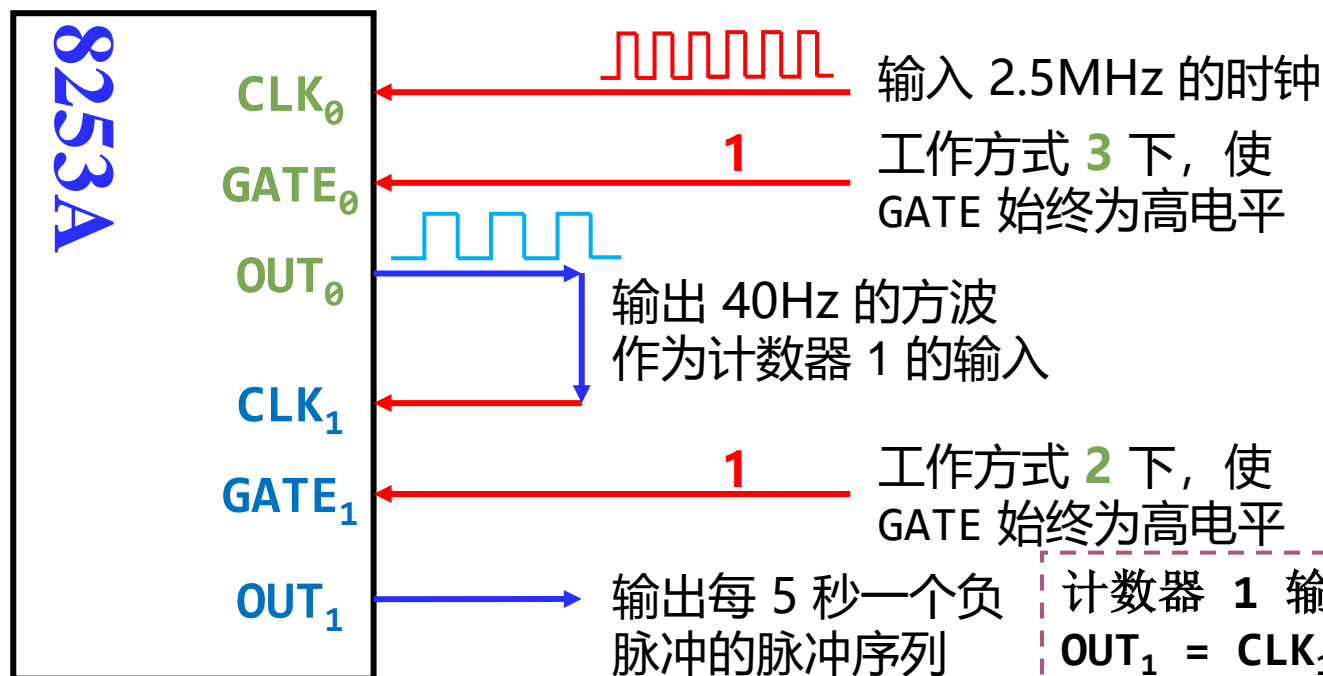
- 计数初值  $n_{\max}=65536$ 
  - $N = 5/T_{\text{CLK}} = 5 \times f_{\text{CLK}} = 5 \times 2.5 \times 10^6 = 12.5 \times 10^6$
  - 故采用**两级计数器**（两级计数初值分别为62500、200）
- 使用 8253A 的计数通道 0 和计数通道 1
  - **计数通道 0**：方式 3，产生频率低于外部时钟频率的**方波**
  - **计数通道 1**：方式 2，产生最终的**负脉冲序列**
- 8253A 的硬件连接
  - 通道 0 的 OUT 输出端作为通道 1 的 CLK 时钟输入
  - 两个通道的 GATE 端均接高电平

# 01 | 定时/计数器-两级计数-硬件连接图

利用 8253A 设计一个定时器，每 5 秒钟输出一个负脉冲，  
设外部时钟频率为 2.5MHz

## ● 硬件连接图

计数器 0 输出脉冲序列的频率  
$$\begin{aligned} \text{OUT}_0 &= \text{CLK}_0 / n_0 \\ &= 2.5\text{MHz} / 62500 \\ &= 40\text{Hz} \end{aligned}$$



计数器 1 输出脉冲序列的频率  
$$\begin{aligned} \text{OUT}_1 &= \text{CLK}_1 / n_1 \\ &= 40\text{Hz} / 200 \\ &= 0.2\text{Hz} \end{aligned}$$

# 01 | 定时/计数器-两级计数-计数初值和命令字

利用 8253A 设计一个定时器，每 5 秒钟输出一个负脉冲，  
设外部时钟频率为 2.5MHz

- 总计数初值  $n=12.5 \times 10^6$
- 采用两级计数时，通道 0、1 的计数初值  $n_0$ 、 $n_1$ 
  - $n = n_0 \times n_1$
  - 假定选择计数值为  $n_0=62500$ ； $n_1=200$

- 设置命令字：

- 计数通道 0
- 计数通道 1

| SC <sub>1</sub> | SC <sub>0</sub> | RW <sub>1</sub> | RW <sub>0</sub> | M <sub>2</sub> | M <sub>1</sub> | M <sub>0</sub> | BCD |
|-----------------|-----------------|-----------------|-----------------|----------------|----------------|----------------|-----|
| 0               | 0               | 1               | 1               | 0              | 1              | 1              | 0   |

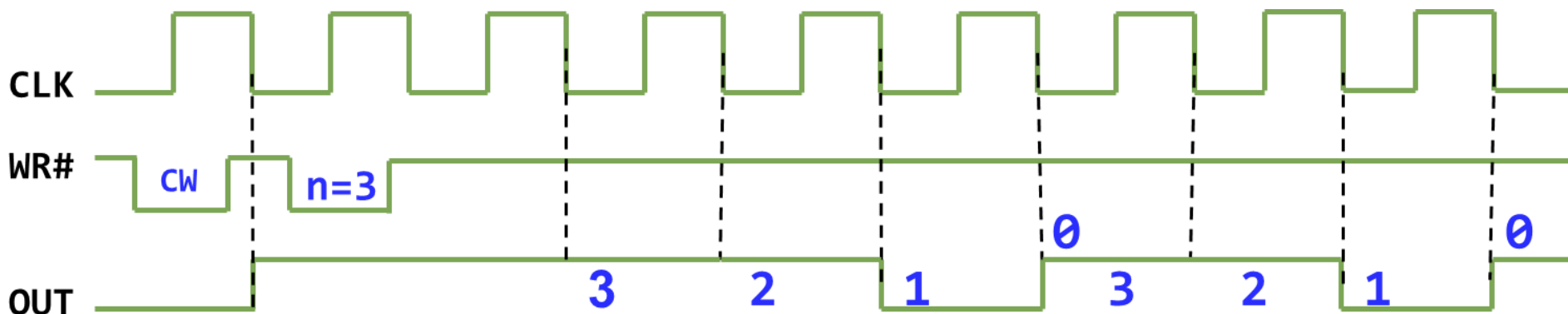
| SC <sub>1</sub> | SC <sub>0</sub> | RW <sub>1</sub> | RW <sub>0</sub> | M <sub>2</sub> | M <sub>1</sub> | M <sub>0</sub> | BCD |
|-----------------|-----------------|-----------------|-----------------|----------------|----------------|----------------|-----|
| 0               | 1               | 0               | 1               | 0              | 1              | 0              | 0   |



# 01 | 定时/计数器-两级计数-思考

本题中**输出波形**是否会因为采用的是**两级计数**或**单通道计数**而有所不同？有何不同？

- ✎ 其输出波形的**负脉冲宽度**会有不同
- ✎ 单通道计数 —— 输出负脉冲宽度为 **1 个时钟周期**
- ✎ 双通道计数 —— 输出负脉冲宽度由**第一级计数的输出**的**方波脉冲宽度**决定



# 01 | 定时/计数器-子目录

## 接口概述

- 定时/计数器的概念
- 8253A 可编程定时/计数器
  - 8253A芯片的工作特点
  - 8253A芯片的外部引脚特性
  - 8253A的内部结构
  - 计数初值的设置
- 8253A 控制字和工作方式
- 8253A 的编程应用
  - 微机系统中的应用
  - 两级计数例子
  - 波特率时钟发生器例子

# 01 | 定时/计数器-应用3-波特率时钟发生器

## 波特率时钟发生器的例子

- **要求：**设计一个波特率分为 8 档的波特率发生器，采用人机对话方式选择波特率，按 Esc 键退出
- **已知：**输入时钟为 1.19318MHz，波特率因子为 16，所需的 8 档输出波特率分别为 110b/s、150b/s、300b/s、600b/s、1200b/s、2400b/s、4800b/s、9600b/s

# 01 | 定时/计数器-应用3-波特率时钟发生器

## 定时常数的计算

- 利用 8253A 方式 3 产生指定波特率的方波，用于串行数据传送
- 定时常数:  $T_c = \text{CLK 频率} / (\text{Baud} \times \text{Factor})$ 
  - 波特率 110b/s,  $T_c = 1.19318 \times 10^6 / (110 \times 16) = 678$  2A6H
  - 波特率 150b/s,  $T_c = 1.19318 \times 10^6 / (150 \times 16) = 497$  1F1H
  - 波特率 300b/s,  $T_c = 1.19318 \times 10^6 / (300 \times 16) = 249$  0F9H
  - 波特率 600b/s,  $T_c = 1.19318 \times 10^6 / (600 \times 16) = 124$  07CH
  - ..... .....
  - 波特率 9600b/s,  $T_c = 1.19318 \times 10^6 / (9600 \times 16) = 8$  008H

# 01 | 定时/计数器-应用3-波特率时钟发生器

## 硬件设计

- 8253A 的计数通道产生所需的时钟
  - CLK 接**输入时钟**，OUT 得到**输出时钟**
  - GATE 接 8255A 的 **PC<sub>6</sub>**，控制 8253A 的工作

## 软件设计

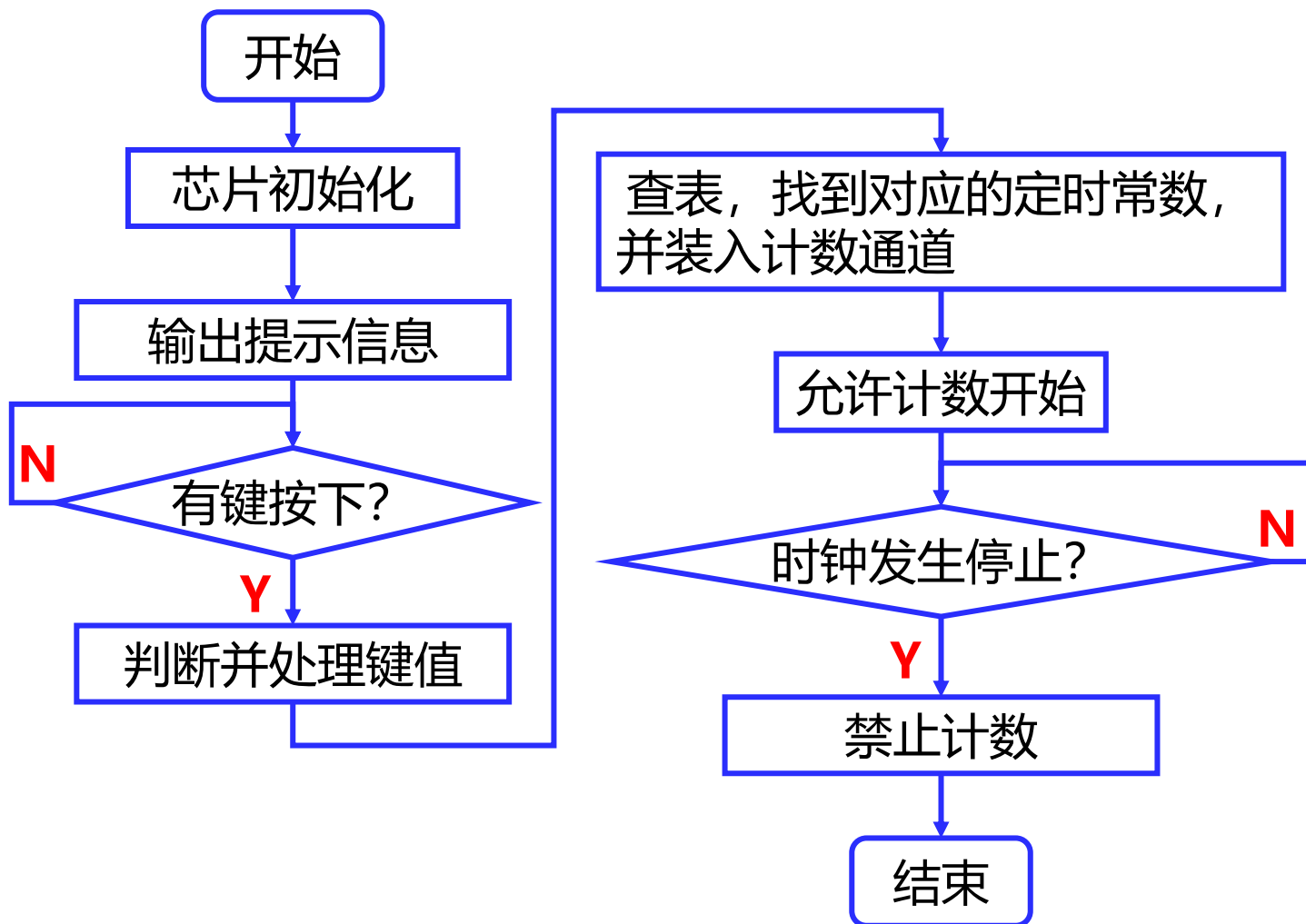
- **定时常数**事先设定，存于数据区
- 根据指定的波特率，**查表**选择定时常数，送入计数通道

↓ TC\_TABLE

|        |    |        |    |        |    |        |    |         |    |         |    |         |    |         |    |  |
|--------|----|--------|----|--------|----|--------|----|---------|----|---------|----|---------|----|---------|----|--|
| A6     | 02 | F1     | 01 | F9     | 00 | 7C     | 00 | 3E      | 00 | 1F      | 00 | 10      | 00 | 08      | 00 |  |
| 110b/s |    | 150b/s |    | 300b/s |    | 600b/s |    | 1200b/s |    | 2400b/s |    | 4800b/s |    | 9600b/s |    |  |

# 01 | 定时/计数器-应用3-波特率时钟发生器

## 程序流程图



请选择波特率:

0 - 110b/s

1 - 150b/s

2 - 300b/s

3 - 600b/s

4 - 1200b/s

5 - 2400b/s

6 - 4800b/s

7 - 9600b/s

# 01 | 定时/计数器-应用3-波特率时钟发生器

## 程序清单 (1/2)

```
;写8253A工作方式控制字
MOV DX, 307H
MOV AL, 10110110B
OUT DX, AL
;写8255A的工作方式控制字
MOV AL, 10000000B
OUT 63H, AL
;使PC6=0, 禁止计数
MOV AL, 00001100B
OUT 63H, AL
;输出字符串的DOS功能调用
LEA DX, INPUT_MESS
MOV AH, 09H
INT 21H
```

```
;等待键盘输入
INPUT: MOV AH, 08H
INT 21H
;判断按键是否合法
CMP AL, '0'
JB INPUT
CMP AL, '7'
JA INPUT
;按键值查表, 获取定时常数
SUB AL, 30H
MOV AH, 0
SHL AX, 1
LEA SI, TC_TABLE
ADD SI, AX
MOV AX, [SI]
```

# 01 | 定时/计数器-应用3-波特率时钟发生器

## 程序清单 (2/2)

```
    ;将定时常数写入计数通道
    MOV DX, 306H
    OUT DX, AL
    MOV AL, AH
    OUT DX, AL
    ;允许计数
    MOV AL, 00001101B
    OUT 63H, AL
    ;检测按键
LOP:  MOV AH, 0BH
      INT 21H
      CMP AL, 0
      JZ LOP
```

```
    ;有按键, 则判断
    MOV AH, 08H
    INT 21H
    CMP AL, 1BH ;ESC
    JE QUIT
    JMP LOP
    ;退出本次计数
QUIT: MOV AL, 00001100B
      OUT 63H, AL
      JMP INPUT
```



# 本节小结

---

- 熟悉8253的外部引脚，清楚各类的引脚的连接对象
- **掌握**8253的内部结构
- **掌握**8253的命令字和状态字，能够完成芯片的初始化编程

# 目录

---

- 01 定时/计数器 8253/8254
- 02 并行通信接口 8255A
- 03 串行通信芯片 8251A

## 02 | 上节回顾-定时/计数器8253A

### 简答题

- 8253A有几个端口？它们分别是什么？8253A计数通道的最大计数值是多少？计数初值如何计算？
- 8253A的工作方式有哪些？它们的触发方式是什么？GATE信号对计数的影响是什么？重新写入初值对计数的影响是什么？
- 请简述如何对8253A进行初始化。

# 02 | 并行接口-子目录

## 8255A 并行接口

- 并行接口的概念
- 8255A 可编程并行接口
  - 主要特性
  - 外部引脚
  - 内部结构
- 8255A 的控制字
- 8255A 的工作方式和编程
  - 基本输入/输出方式 (方式 0)
  - 选通输入/输出方式 (方式 1)
  - 双向输入/输出方式 (方式 2)
- 8255A 在微机中的应用



## 02 | 并行接口-并行通信的概念

### 并行通信

- 通信双方有**多条数据传送线**，二者之间的数据（一般以**字节**或**字**为单位）通过这些数据线同时传送



## 02 | 并行接口-并行通信的特点

### 优点

- 传输**速度快**，信息传输**效率高**

### 缺点

- 所用电缆多，**成本较高**
- 在长距离传输时，电缆间的**干扰大**，影响数据传送的正确性

### 适用场合

- 常用在**近距离**传送中使用，如微机系统中

# 02 | 并行接口 - 并行接口的概念

## 并行接口

- 可传送并行数据的接口电路

## 硬件结构

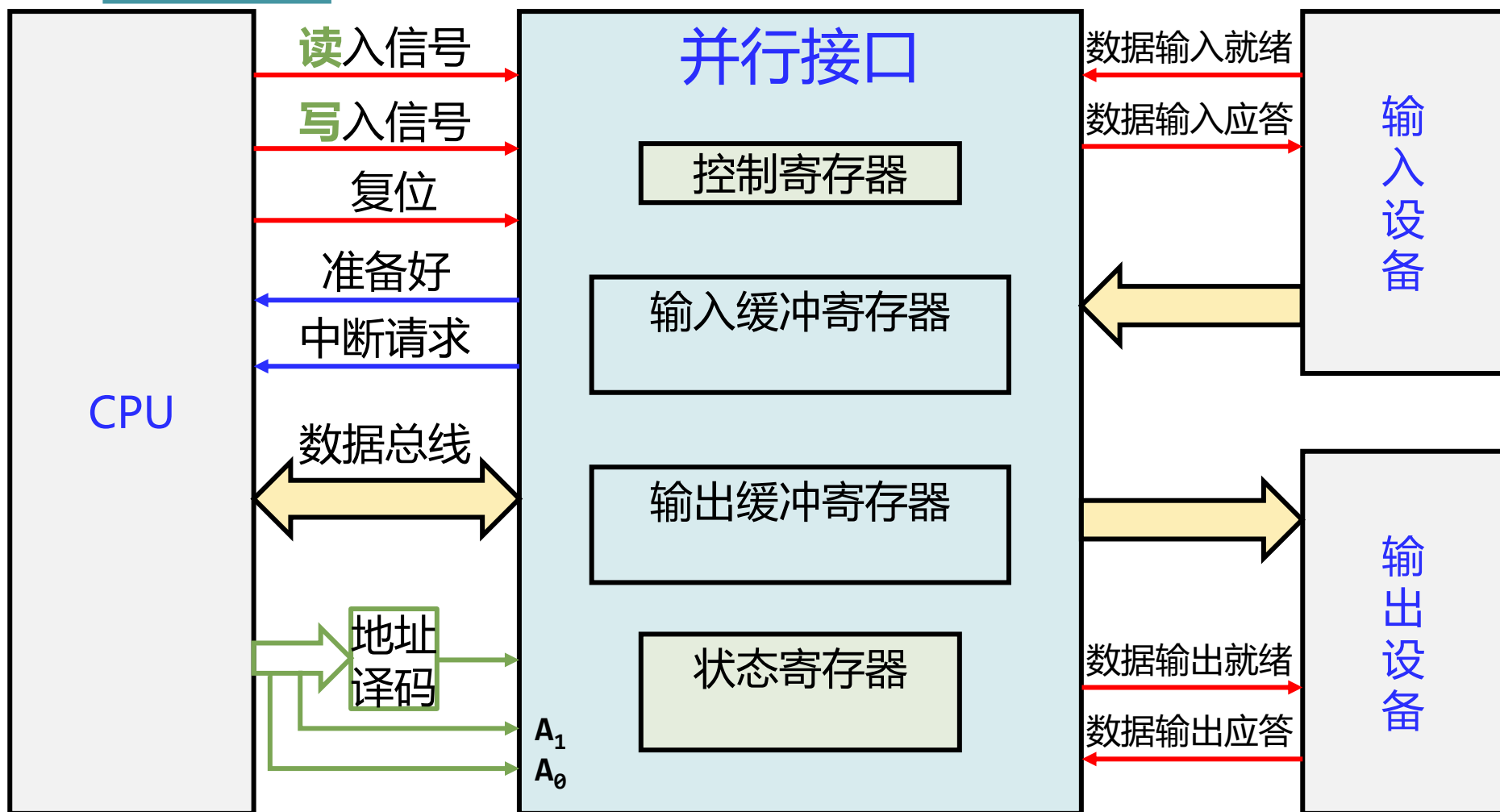
- 一个或一个以上具有**锁存**或**缓冲**的数据端口
- 与 **CPU** 交换数据的**控制**和**状态**联络信号及端口
- 与**外设**交换数据的**控制**和**状态**联络信号及端口

## 并行接口的类型和识别

- 并行接口可分为**简单接口**和**可编程接口**两种
- **CPU** 与任何接口都是并行的，关键是**接口**与**外设**之间的传送方式

# 02 | 并行接口 - 并行接口连接示意图

## 并行接口





## 02 | 并行接口-并行接口的功能

### 实现并行数据的无条件方式传送

- 通过并行接口内的数据端口实现直接的并行数据传送

### 实现并行数据的查询方式传送

- 并行接口中具有应答机制，在数据传送过程中，通过状态、控制信号与 CPU 和外设联络，以保证数据传送的有效进行

### 实现并行数据的中断方式传送

- 并行接口中具有一定的中断机制，可产生中断请求信号，作为外设中断源提交给 8259 处理，请求数据的传送操作
- 中断方式下的数据传送必须与 8259 配合

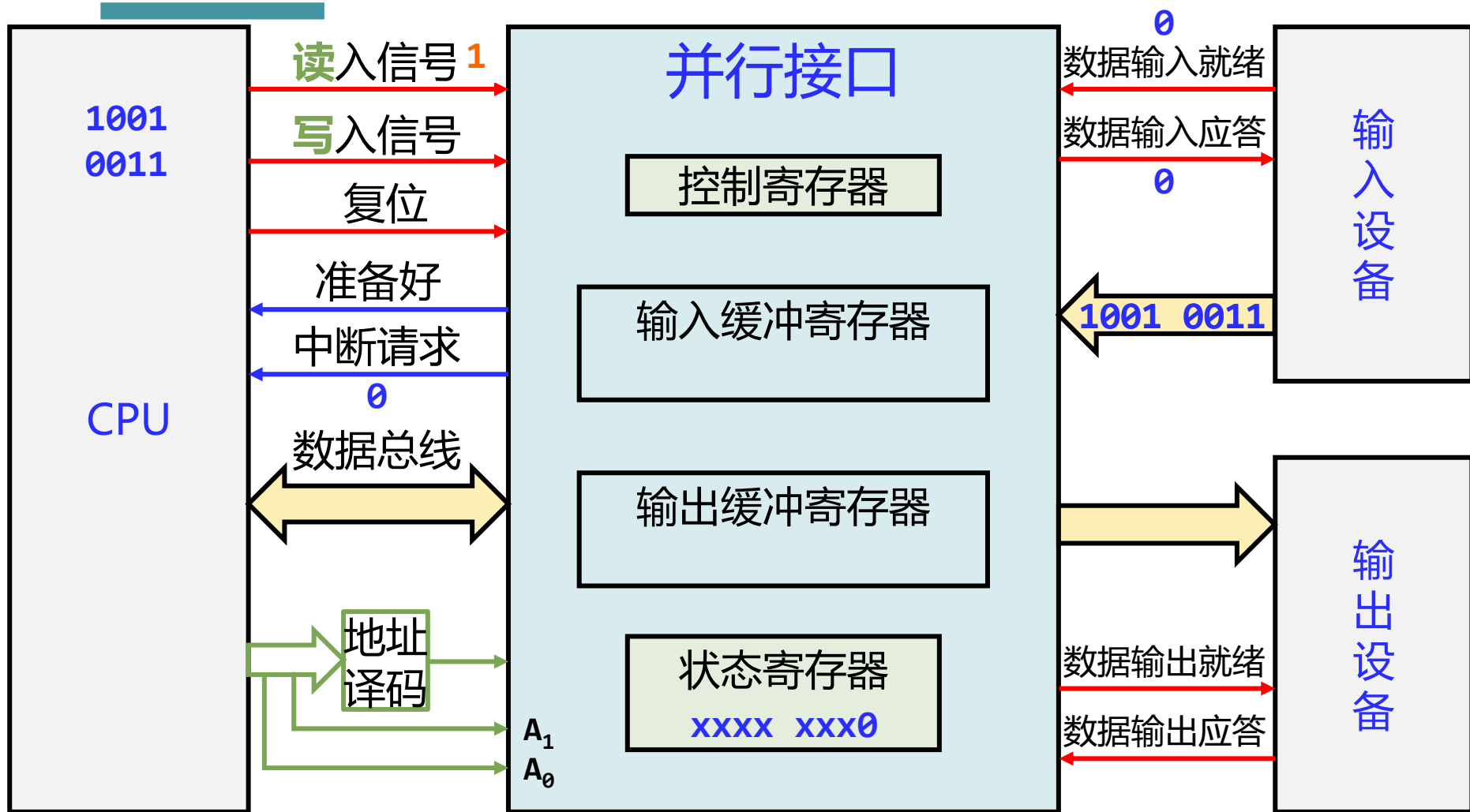
## 02 | 并行接口 - 并行接口的输入过程

### 并行接口的输入过程

1. 外设先将**数据送给接口**，并置“**数据输入就绪**”信号有效
2. 接口把接收到的数据输入缓冲器的同时，设置“**数据输入应答**”信号，作为对外设的响应
3. 外设接到此信号后，撤消数据和“**数据输入就绪**”信号
4. **数据到达接口**后，接口状态寄存器中设置“**输入准备好**”状态位，以便 CPU 查询，或由中断机制生成中断请求信号
5. CPU 查询到“**输入准备好**”的有效信号，或得到中断请求，进行数据输入处理时，**从接口中读取数据**，接口由该读取操作自动复位“**输入准备好**”状态位，结束一次数据输入操作

# 02 | 并行接口 - 数据输入过程示意图

## 数据输入过程



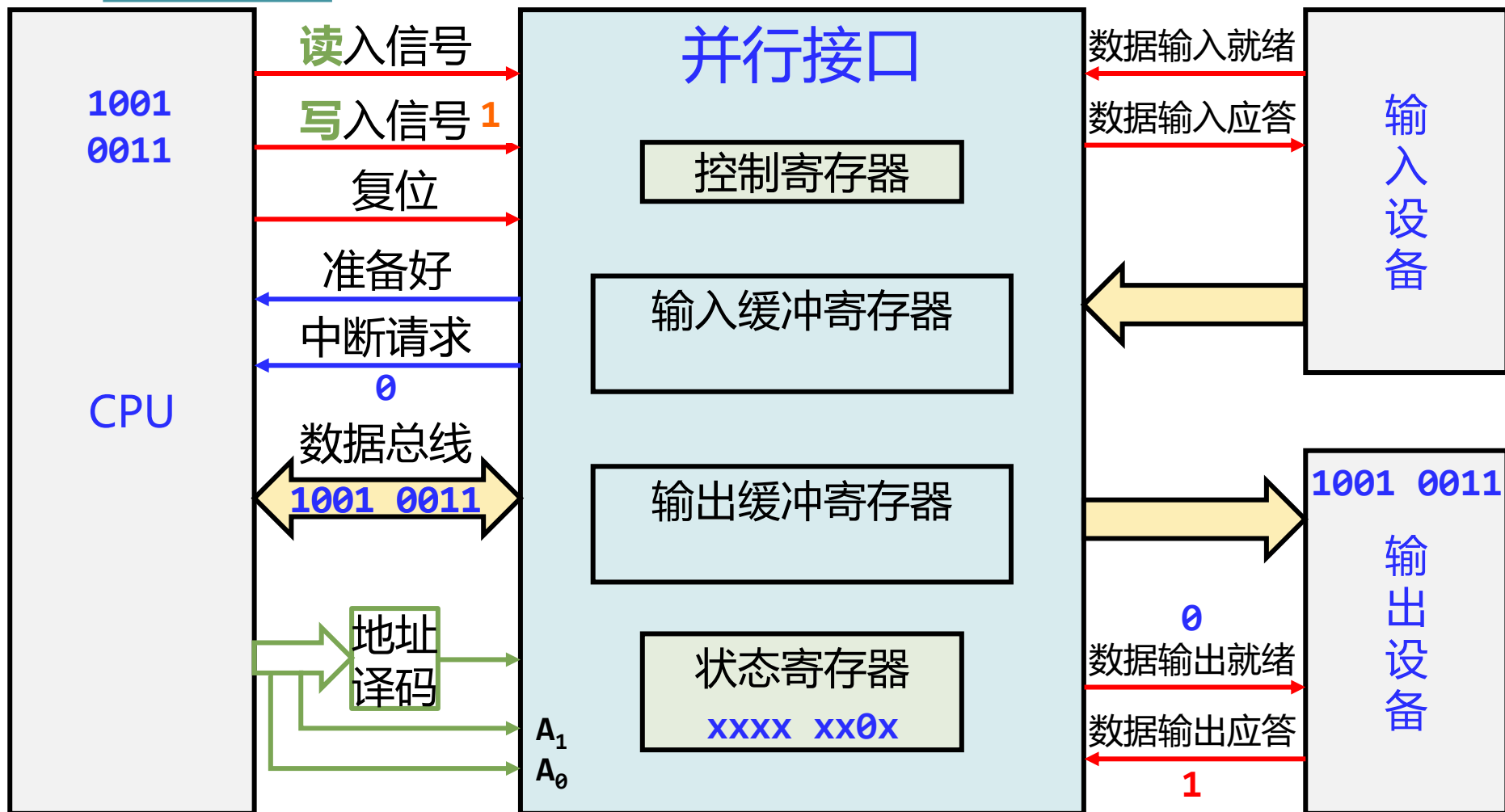
## 02 | 并行接口 - 并行接口的输出过程

### 并行接口的输出过程

1. 当接口状态寄存器的“**输出准备好**”位置位时，CPU 将**数据写入接口**的输出缓冲器
2. 执行写入操作后，接口自动清除“**输出准备好**”状态位，同时向外设发送“**数据输出就绪**”信号，并将**数据送到与外设连接的数据总线上**
3. 外设接收到“**数据输出就绪**”信号后，开始**接收数据**，完毕时向接口发回“**数据输出应答**”信号
4. 接口收到此信号后，置位状态寄存器中的“**输出准备好**”位，表示当前输出缓冲器中没有需要外设接收的新数据，等待 CPU 的下次输出操作

# 02 | 并行接口 - 数据输出过程示意图

## 数据输出过程



# 02 | 并行接口-子目录

## 8255A 并行接口

- 并行接口的概念
- 8255A 可编程并行接口
  - 主要特性
  - 外部引脚
  - 内部结构
- 8255A 的控制字
- 8255A 的工作方式和编程
  - 基本输入/输出方式（方式0）
  - 选通输入/输出方式（方式1）
  - 双向输入/输出方式（方式2）
- 8255A 在微机中的应用

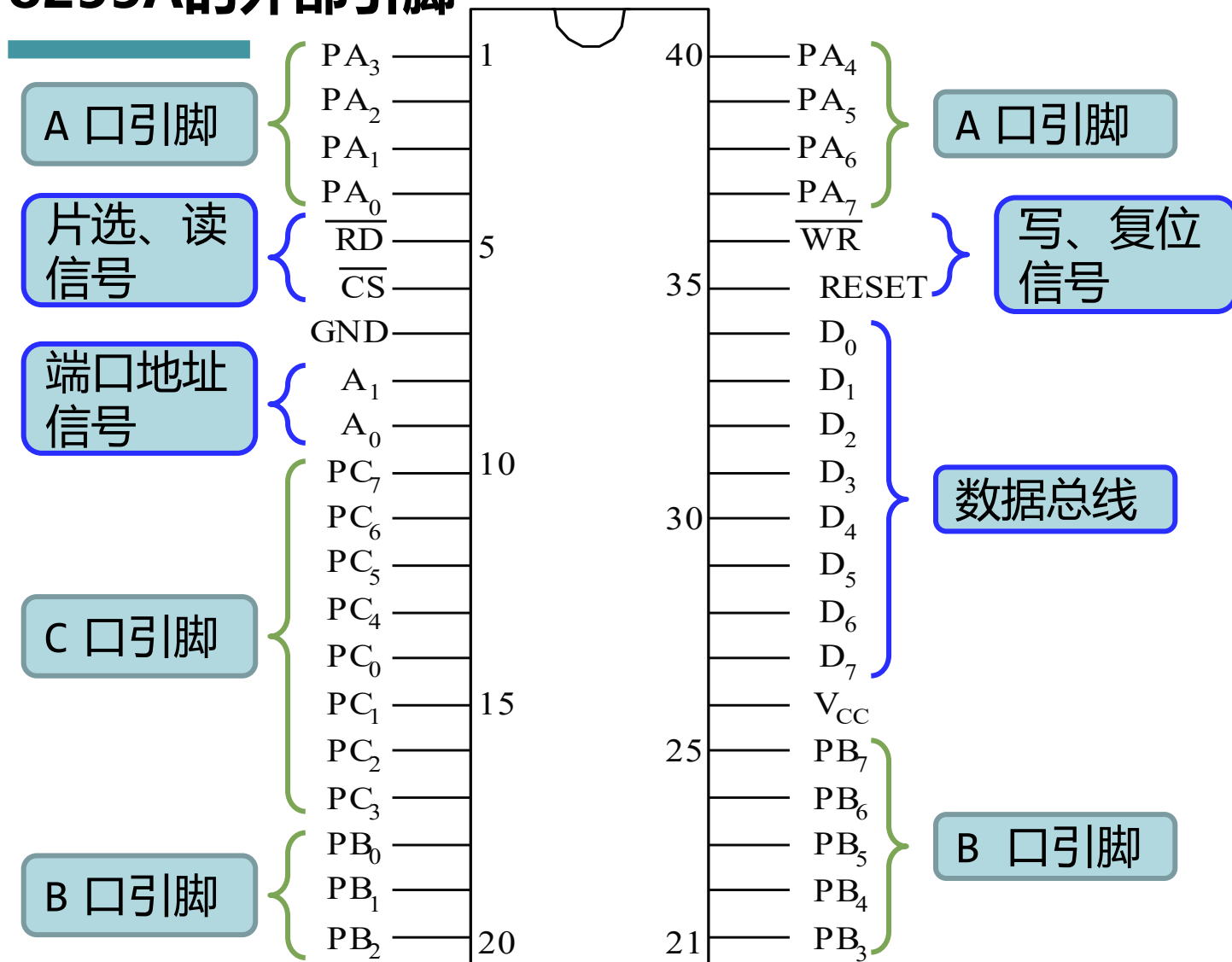
# 02 | 并行接口-8255A的主要特性

## 8255A的主要特性

- 具有三个 8 位的双向数据端口 (A 口、B 口、C 口)
- 具有 3 种工作方式，可通过编程选择
  - A 口 – 方式 0、1、2
  - B 口 – 方式 0、1
  - C 口 – 方式 0
- 支持无条件、程序查询、中断等数据传送方式
- 数据端口 C 口的使用较为特殊：
  - 既可作为一个 8 位端口，也可作为两个 4 位端口来使用
  - 既可作为数据端口，也可用作联络信号配合 A、B 口工作
  - 既可传送并行数据，也可单独按位控制，有专门的控制字

# 02 | 并行接口-8255A的外部引脚

## 8255A的外部引脚



Port A  
Port B  
Port C

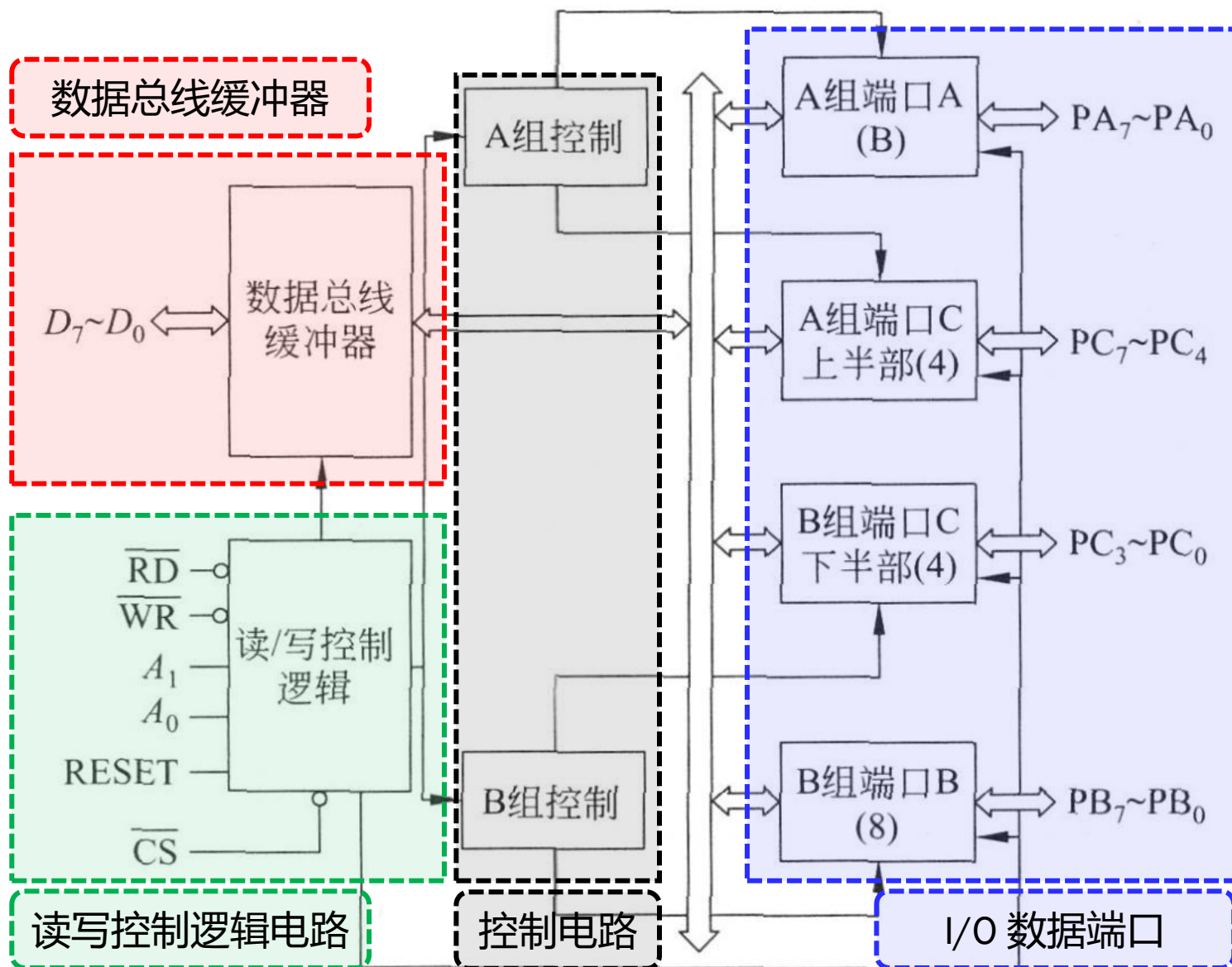


## 02 | 并行接口-8255A的外部引脚

### 8255A的外部引脚

- 面向 CPU 的引脚
  - RESET、CS#、RD#、WR#、A<sub>1</sub>、A<sub>0</sub>、D<sub>7</sub> ~ D<sub>0</sub>
  - 按下 RESET 后的复位状态为：  
片内所有寄存器清零，三个数据端口设为输入状态
- 面向外设的引脚
  - PA<sub>0</sub> ~ PA<sub>7</sub>、PB<sub>0</sub> ~ PB<sub>7</sub>、PC<sub>0</sub> ~ PC<sub>7</sub>

## 02 并行接口-8255A的内部结构1



## 02 | 并行接口-8255A的内部结构2

### 数据总线缓冲器

- 三态双向 **8 位**数据缓冲器
- 8255 **与 CPU 之间**的数据中转站
- 可传送的数据：
  - CPU 要写入 8255A 的**控制字**
  - CPU 与外设之间传送的**数据**
  - CPU 要获取的外设**状态字**

## 02 | 并行接口-8255A的内部结构3

### 读写控制逻辑电路

| CS# | RD# | WR# | A <sub>1</sub> A <sub>0</sub> | 执行的操作类型                   |
|-----|-----|-----|-------------------------------|---------------------------|
| 1   | x   | x   | x x                           | 未选中该 8255A 芯片             |
| 0   | 0   | 1   | 0 0                           | CPU 对 8255A 芯片的 A 端口执行读操作 |
| 0   | 1   | 0   | 0 0                           | CPU 对 8255A 芯片的 A 端口执行写操作 |
| 0   | 0   | 1   | 0 1                           | CPU 对 8255A 芯片的 B 端口执行读操作 |
| 0   | 1   | 0   | 0 1                           | CPU 对 8255A 芯片的 B 端口执行写操作 |
| 0   | 0   | 1   | 1 0                           | CPU 对 8255A 芯片的 C 端口执行读操作 |
| 0   | 1   | 0   | 1 0                           | CPU 对 8255A 芯片的 C 端口执行写操作 |
| 0   | 0   | 1   | 1 1                           | 无效                        |
| 0   | 1   | 0   | 1 1                           | CPU 对 8255A 芯片的命令端口执行写操作  |

## 02 | 并行接口-8255A的内部结构4

### A 组控制电路和 B 组控制电路

- 分组情况
  - A 组 —— A 口和 C 口的高 4 位 ( $PC_4 \sim PC_7$ )
  - B 组 —— B 口和 C 口的低 4 位 ( $PC_0 \sim PC_3$ )
- 控制电路的作用：
  - A、B 组控制电路接收来自读/写部件的命令和 CPU 写入命令端口的控制字，译码后控制对应端口的工作方式和读/写操作方式。
  - 对 C 口的按位操作也由此控制部件实现

## 02 | 并行接口-8255A的内部结构5

### 三个独立的输入/输出数据端口 (A、B、C口)

- 端口特点
  - 3 个 8 位数据端口，都由**输入寄存器**和**输出寄存器**构成
  - **A 口**的输入/输出寄存器均有锁存功能
  - **B、C 口**的输出寄存器具有锁存功能，但其输入寄存器不能锁存数据
- C 口的独特工作方式
  - A、B 口工作于方式 1、2 时，C 口的高/低 4 位分别作为 A、B 口在数据传输时的控制联络信号线
  - C 口的每一位可独立进行置/复位的操作
- A、B、C 口及命令口的端口地址分别为 60H ~ 63H

设8255A的端口地址为60H~63H，则控制字寄存器的地址为（ ）

- ☐ A 60H
- ☐ B 61H
- ☐ C 62H
- ☒ D 63H

# 02 | 并行接口-子目录

## 8255A 并行接口

- 并行接口的概念
- 8255A 可编程并行接口
  - 主要特性
  - 外部引脚
  - 内部结构
- 8255A 的控制字
- 8255A 的工作方式和编程
  - 基本输入/输出方式（方式0）
  - 选通输入/输出方式（方式1）
  - 双向输入/输出方式（方式2）
- 8255A 在微机中的应用



## 02 | 并行接口-8255A的控制字

### 8255A 的控制字

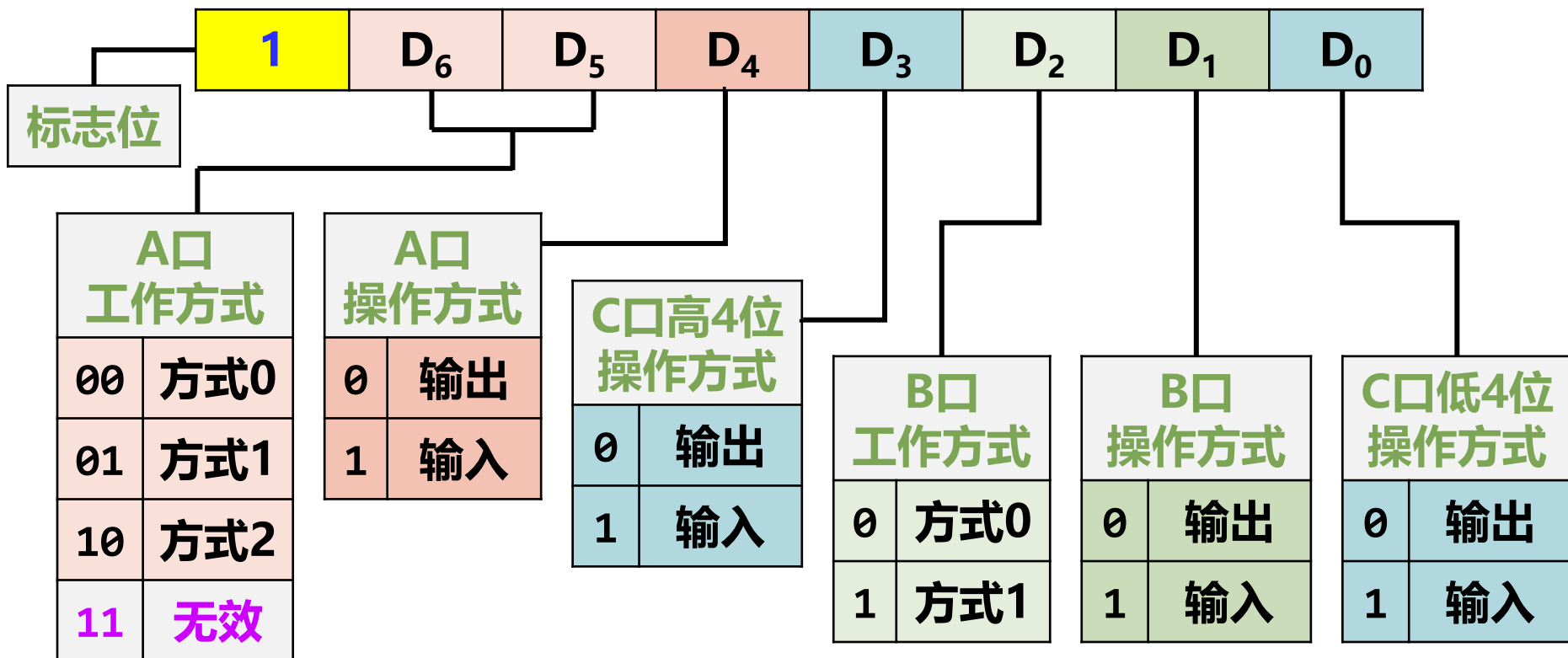
- **工作方式选择控制字**
  - 指定 8255A 的 3 个端口**工作方式**及该方式下的**操作类型**
- **C 口置/复位控制字**
  - 指定 C 口的某一位输出高电平或低电平

### CPU 对 8255A 的编程使用

- 向 8255A 的命令端口**写入工作方式控制字**，选择工作方式
- 若需要传送数据，则对 A、B、C 口进行**数据读写**
- 若需要 C 口发送命令，则可使用 **C 口置/复位控制字** 来设置

# 02 | 并行接口-8255A控制字-工作方式选择

## 8255A 的工作方式控制字



## 02 | 并行接口-8255A工作方式控制字举例1

示例1：A 口工作于方式 1，输入口，C 口高 4 位为输出；B 口工作于方式 0，输出口，C 口低 4 位为输入

- 工作方式控制字

| 1 | D <sub>6</sub> | D <sub>5</sub> | D <sub>4</sub> | D <sub>3</sub> | D <sub>2</sub> | D <sub>1</sub> | D <sub>0</sub> |
|---|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| 1 | 0              | 1              | 1              | 0              | 0              | 0              | 1              |

- 初始化程序段

- 写入命令字

```
MOV AL, 0B1H  
OUT 63H, AL
```

## 02 | 并行接口-8255A工作方式控制字举例2

示例2：A 口工作于方式 2，输入口；B 口工作于方式 1，输出口

- 工作方式控制字

| 1 | D <sub>6</sub> | D <sub>5</sub> | D <sub>4</sub> | D <sub>3</sub> | D <sub>2</sub> | D <sub>1</sub> | D <sub>0</sub> |
|---|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| 1 | 1              | 0              | 1              | 0              | 1              | 0              | 0              |

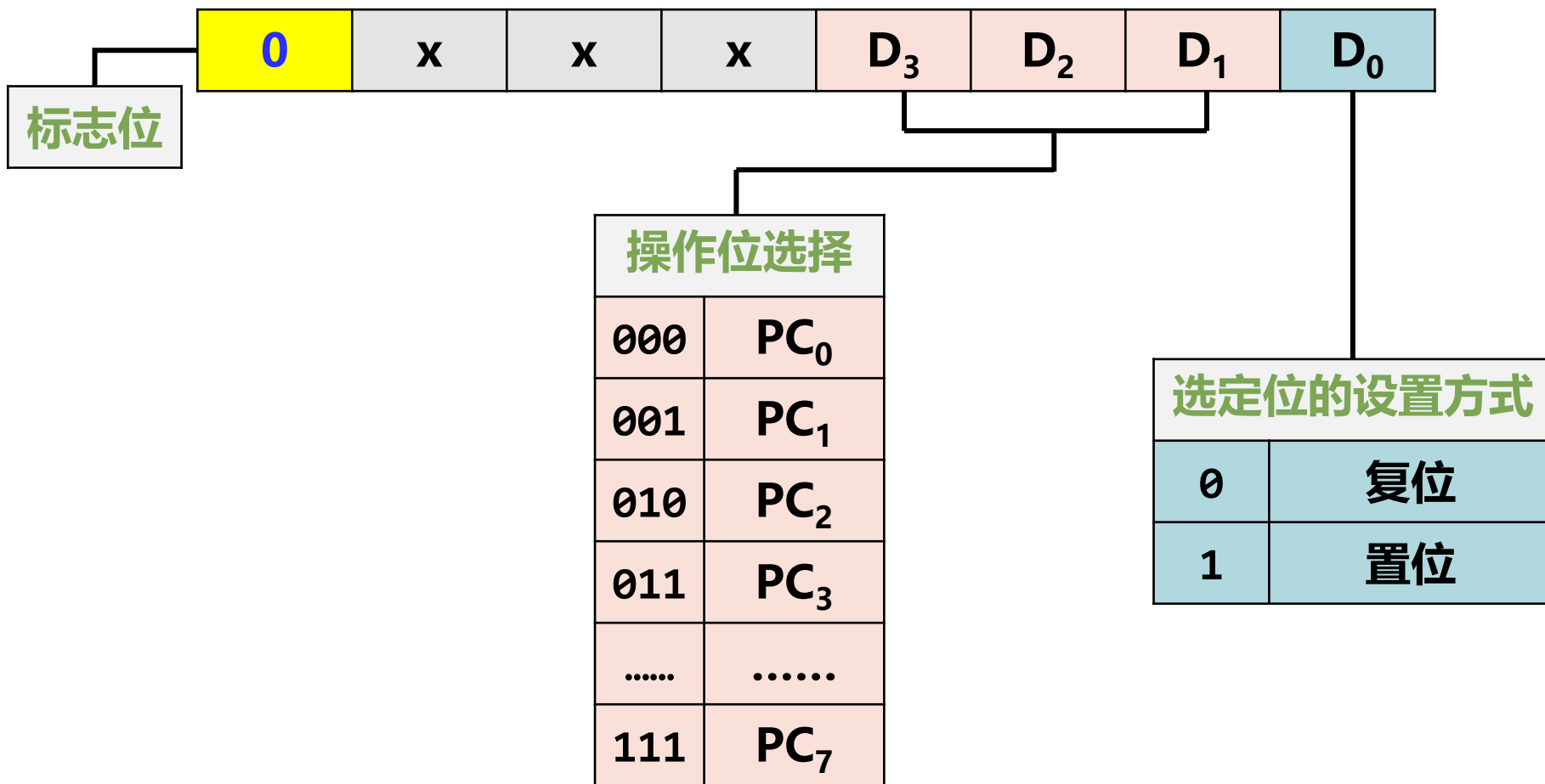
- 初始化程序段

- 写入命令字

```
MOV AL, 0D4H  
OUT 63H, AL
```

# 02 | 并行接口-8255A控制字-C口置复位

## 8255A 的 C 口置复位控制字



## 02 | 并行接口-C口置复位控制字举例1

示例1：使  $PC_2$  输出高电平，则相应的程序段应如何设置

- 工作方式控制字

| 0 | x | x | x | $D_3$ | $D_2$ | $D_1$ | $D_0$ |
|---|---|---|---|-------|-------|-------|-------|
| 0 | 0 | 0 | 0 | 0     | 1     | 0     | 1     |

- 初始化程序段

- 写入命令字

```
MOV AL, 05H  
OUT 63H, AL
```

## 02 | 并行接口-C口置复位控制字举例2

### 示例2：使 PC<sub>7</sub> 输出负脉冲，作为外设的选通信号

#### ● 程序段

```
      MOV AL, 00001111B  
      OUT 63H, AL  
      MOV CX, 50H  
L1:   LOOP L1  
      MOV AL, 00001110B  
      OUT 63H, AL  
      MOV CX, 5H  
L2:   LOOP L2  
      MOV AL, 00001111B  
      OUT 63H, AL
```

使PC<sub>7</sub>输出高电平

维持一段时间

使PC<sub>7</sub>输出低电平

维持一段时间（短）

使PC<sub>7</sub>输出高电平

## 02 | 并行接口-C口置复位控制字举例3

示例3：使 PC<sub>7</sub> 输出方波，经滤波放大后送喇叭发声

- 输出一个方波子程序：

```
SPK PROC
    MOV AL, 00001111B
    OUT 63H, AL
    CALL DELAY1
    MOV AL, 00001110B
    OUT 63H, AL
    CALL DELAY1
    RET
SPK ENDP
```

连续方波：

```
L: CALL SPK
    JMP L
```

可控连续方波：

```
L: CALL SPK
    .....
    .....
    Jcc L
```

条件转移指令



# 02 | 并行接口-应用控制字的注意要点

## 应用控制字的注意要点

- 写入时间：
  - **工作方式**控制字要在**使用 8255A 之前**写入其命令端口
  - **C 口置/复位**控制字在 8255A 工作的**任何时候**进行的
- 两个控制字写入**同一命令端口**，即 63H
  - 一定不能将置/复位控制字写入 C 口 (62H) 中
- 对 C 口的设置可以有两种方法：
  - 使用 **C 口置/复位控制字**，**每次设置一个** PC 位的输出状态
    - ◆ 无论初始化时该 PC 位的输入输出状态
  - **直接写 C 口**，**每次必须同时设置多个** (4 或 8) PC 位的输出状态
    - ◆ 初始化时，该 PC 位应为**输出状态**

使8255A的PC<sub>6</sub>输出高电平的C口置复位命令字为  
( )

- ☐ A 40H
- ☐ B 0FH
- ☒ C 0DH
- ☐ D 10H

# 02 | 并行接口-子目录

## 8255A 并行接口

- 并行接口的概念
- 8255A 可编程并行接口
- 8255A 的控制字
- 8255A 的工作方式和编程
  - 基本输入/输出方式（方式0）
  - 选通输入/输出方式（方式1）
  - 双向输入/输出方式（方式2）
- 8255A 在微机中的应用

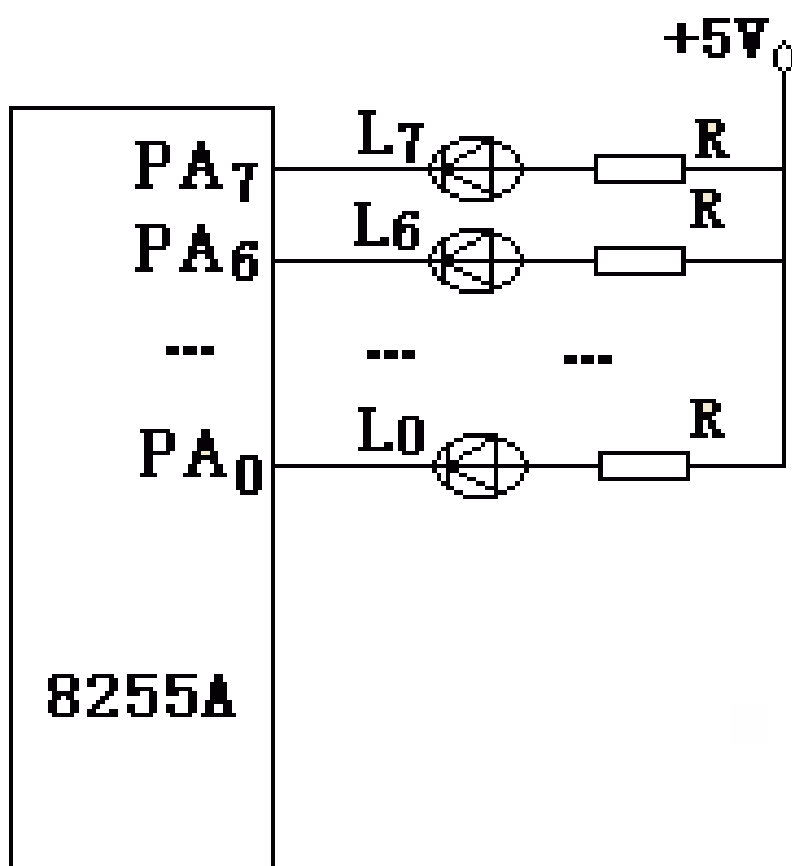
## 02 | 并行接口-方式0-基本输入/输出方式

### 方式 0 的工作特点

- 基本的输入/输出方式，此方式**没有固定的联络信号**，A、B、C 三个端口均可作为数据端口工作于此方式下
- **单向 I/O 方式**，即**每次初始化**后，指定端口 (两个 8 位端口和两个 4 位端口) **只能**作为输入端口或输出端口
- 方式 0 下的数据传送，可以是**无条件方式**，或**查询方式**
  - 方式 0 下的查询传送方式，需要使用 C 口中的某些 PC 位 **自定义**为所需的联络信号

## 02 | 并行接口-基本输入/输出方式举例1

示例1：利用 8255A 控制 8 个发光二极管的亮和灭，要求每隔 0.5s 依次点亮  $L_7 \sim L_0$  一次，周而复始，硬件连接如下图



- 发光二极管上有电流通过时，会被点亮
  - PA 端输出 1，二极管熄灭
  - PA 端输出 0，二极管点亮
- 编写循环程序，控制 PA<sub>7</sub>~PA<sub>0</sub> 每隔 0.5s 依次输出 0，而后变为 1，即可满足题目要求

## 02 | 并行接口-基本输入/输出方式举例1

示例1：利用 8255A 控制 8 个发光二极管的亮和灭，要求每隔 0.5s 依次点亮  $L_7 \sim L_0$  一次，周而复始，硬件连接如下图

- 8255A初始化控制字：

- A 口**方式 0**，**输出**端口，  
即控制字为 1**000** 0000

- 初始化程序段：

```
MOV AL, 80H
OUT 63H, AL
```

- A 口按要求输出数据程序段：

```
MOV AH, 7FH
LOP: MOV AL, AH
      OUT 60H, AL
      CALL DELAY1
      ROR AH, 1
      JMP LOP
```

控制数据

延时0.5s

## 02 | 并行接口 - 基本输入/输出方式举例2

示例2：利用 8255A 的 A 口方式 0 设计并行打印机接口。  
要求：CPU 采用**查询方式**将缓冲区 BUF 中的 256 个字符通过接口送到打印机打印

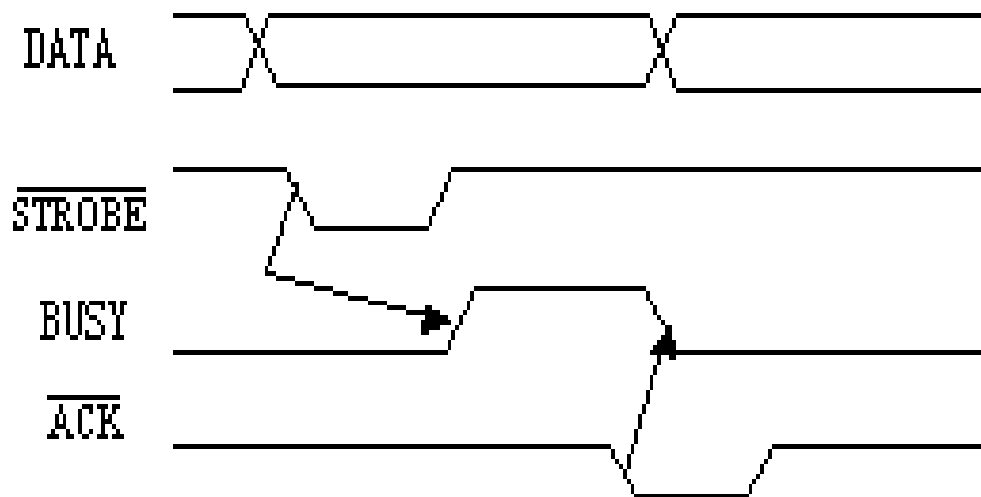
- 采用查询方式：
  - 打印在接口和打印机之间必须要有**联络控制信号**，以返回打印机**状态**，及**控制**打印机工作
    - ◆ 打印机的**控制信号和工作时序**
    - ◆ **接口电路和打印机之间的连接**，及信号控制的完成
- 打印内容：
  - 程序中 CPU 必须先将要打印的数据**从缓冲区 BUF 中逐个读出**，在查询到**打印机就绪状态**后送到接口中

## 02 | 并行接口-基本输入/输出方式举例2

### 示例2：利用 8255A 的 A 口方式 0 设计并行打印机接口 打印机控制信号和工作时序

#### ● 控制信号：

- STB#：输入，打印机选通信号
- BUSY：输出，表示忙状态信号
- ACK#：输出，打印机工作结束应答信号



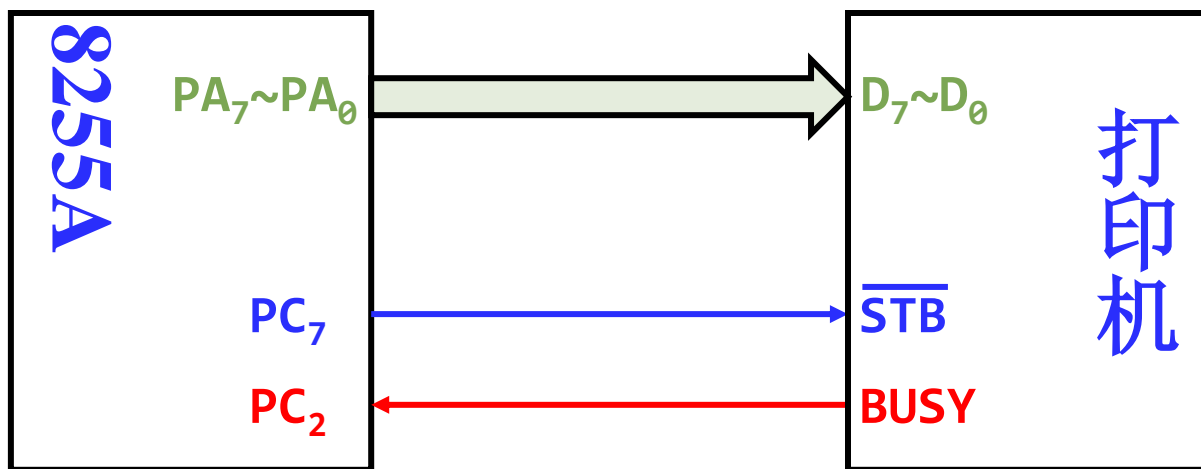
#### ● 工作时序：

- 需要打印数据时，CPU 先查看打印机的 BUSY 忙信号，若 **BUSY=0**，则将数据送数据总线
- 数据稳定后，发 **STB# 选通信号**，打印机即读入数据，同时使 **BUSY 信号有效**，通知主机停止送数
- 打印机处理完当前数据后，使 **ACK# 响应信号有效**，同时使 **BUSY 失效**，通知主机可以发下一个数据



## 02 | 并行接口-基本输入/输出方式举例2

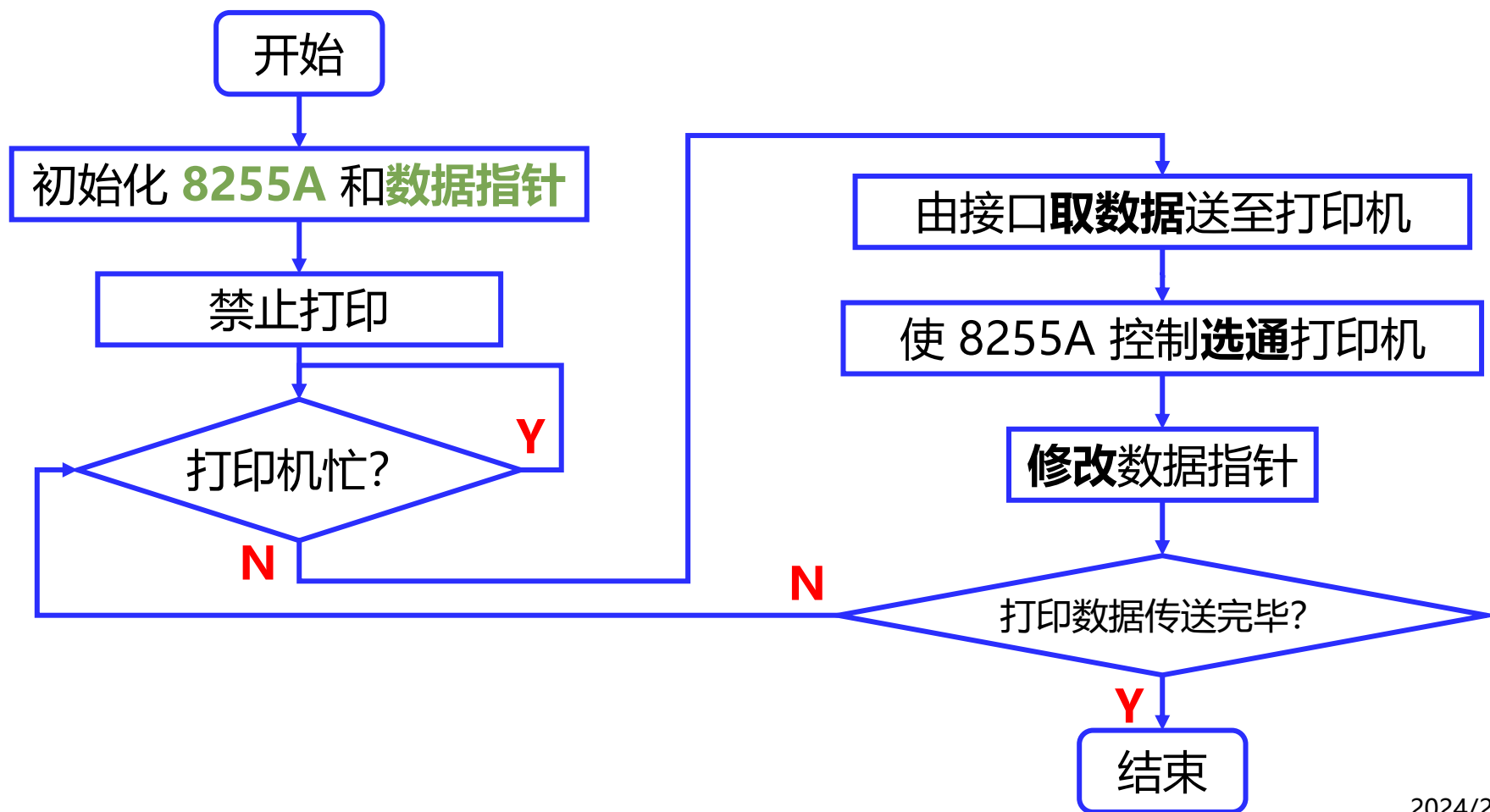
示例2：利用 8255A 的 A 口方式 0 设计并行打印机接口  
打印机与接口之间的连接图



- 思考：8255A 工作于方式 0 下，与打印机之间的联络信号如何设置？
  - ✎ 所需要的选通信号和状态信号对 8255A 来说分别是**输出**和**输入**的两种状态，可使用 **C 口作为联络信号**，并且把**高低 4 位**分别定义为输入和输出两种操作方式，以满足联络需要

## 02 | 并行接口-基本输入/输出方式举例2

示例2：利用 8255A 的 A 口方式 0 设计并行打印机接口  
程序流程图



## 02 | 并行接口-基本输入/输出方式举例2

### 示例2：利用 8255A 的 A 口方式 0 设计并行打印机接口 初始化程序段 (8255A)

- A 口作为数据输出口，工作于方式 0
- B 口没有使用
- C 口作为查询联络信号：
  - $PC_7$  —— 输出的选通信号
  - $PC_2$  —— 输入的状态信号
- 8255A 的工作方式控制字为

| 1 | D <sub>6</sub> | D <sub>5</sub> | D <sub>4</sub> | D <sub>3</sub> | D <sub>2</sub> | D <sub>1</sub> | D <sub>0</sub> |
|---|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| 1 | 0              | 0              | 0              | 0              | 0              | 0              | 1              |

- 8255A 的初始化程序段

```
MOV AL, 81H  
OUT 63H, AL
```

## 02 | 并行接口-基本输入/输出方式举例2

### 示例2：利用 8255A 的 A 口方式 0 设计并行打印机接口 初始化程序段 (数据指针)

- 使用间接寻址或相对寻址的方式对连续的存储空间寻址
- 打印 BUF 缓冲区中的 256 个字符，故取操作数的程序为循环次数已知的循环程序，需要设置循环计数器 CX

#### ; 间接寻址方式:

```
MOV SI, OFFSET BUF
; LEA SI, BUF ; 与上述指令等价
MOV CX, 256
AGAIN: .....
MOV AL, [SI]
        .....
INC SI
LOOP AGAIN
```

#### ; 相对寻址方式:

```
MOV SI, 0
MOV CX, 256
AGAIN: .....
MOV AL, BUF[SI]
        .....
INC SI
LOOP AGAIN
```

## 02 | 并行接口-基本输入/输出方式举例2

### 示例2：利用 8255A 的 A 口方式 0 设计并行打印机接口 打印的选通与禁止

- 打印机的选通信号 STB# 由 8255A 的 PC<sub>7</sub> 控制
  - PC<sub>7</sub> 输出**高电平**，禁止打印机工作

|   |   |   |   |                |                |                |                |
|---|---|---|---|----------------|----------------|----------------|----------------|
| 0 | x | x | x | D <sub>3</sub> | D <sub>2</sub> | D <sub>1</sub> | D <sub>0</sub> |
| 0 | 0 | 0 | 0 | 1              | 1              | 1              | 1              |

- PC<sub>7</sub> 输出**下降沿**，选通打印机工作

|   |   |   |   |                |                |                |                |
|---|---|---|---|----------------|----------------|----------------|----------------|
| 0 | x | x | x | D <sub>3</sub> | D <sub>2</sub> | D <sub>1</sub> | D <sub>0</sub> |
| 0 | 0 | 0 | 0 | 1              | 1              | 1              | 0              |

;**禁止打印**:

```
MOV AL, 0FH
OUT 63H, AL
```

;**选通打印**:

```
MOV AL, 0EH
OUT 63H, AL
NOP
NOP
MOV AL, 0FH
OUT 63H, AL
```

## 02 | 并行接口 - 基本输入/输出方式举例2

### 示例2：利用 8255A 的 A 口方式 0 设计并行打印机接口 “打印机忙”的判断和数据的输出打印

- 读取 8255A 的  $PC_2$  的状态，以判断下一步操作
  - 0 - 空闲，可以输出欲打印数据
  - 1 - 忙，上次数据尚未打印完毕，CPU 需等待
- 程序段

```
WAIT0:  IN AL, 62H    ;读C口
        AND AL, 04H   ;判断PC2状态
        JNZ WAIT0     ;若BUSY = 1, 则等待
        .....        ;否则, 取下一打印数据
        OUT 60H, AL    ;将数据由A口送至打印机
```

# 02 | 并行接口 - 基本输入/输出方式举例2

## 示例2：利用 8255A 的 A 口方式 0 设计并行打印机接口 8255A 方式 0 下的查询传送源程序段

打印机就绪后，CPU  
送出要打印的数据

```
.....  
MOV AL, 81H  
OUT 63H, AL  
MOV SI, 0  
MOV CX, 256  
MOV AL, 0FH  
OUT 63H, AL  
WAIT0: IN AL, 62H  
AND AL, 04H  
JNZ WAIT0
```

8255A初始化

指针初始化

PC<sub>7</sub>=1禁止打印

查询PC<sub>2</sub>的  
状态，决定  
是否打印

```
MOV AL, BUF[SI]  
OUT 60H, AL  
MOV AL, 0EH  
OUT 63H, AL  
NOP  
NOP  
MOV AL, 0FH  
OUT 63H, AL  
INC SI  
LOOP WAIT0  
.....
```

通过PC<sub>7</sub>选通打印机  
开始接收数据并打印

循环继续以查询方  
式输出数据并打印

# 02 | 并行接口-子目录

## 8255A 并行接口

- 并行接口的概念
- 8255A 可编程并行接口
- 8255A 的控制字
- 8255A 的工作方式和编程
  - 基本输入/输出方式 (方式0)
  - 选通输入/输出方式 (方式1)
  - 双向输入/输出方式 (方式2)
- 8255A 在微机中的应用



## 02 | 并行接口-方式1-选通输入/输出方式

### 方式 1 的工作特点

### 带选通的单向 I/O 方式

- 选通输入/输出方式，A、B 口可工作于方式 1 下，由指定的 PC 位作为固定的联络信号使用
  - 输入和输出操作所需的联络信号不同，但各操作的联络信号都存在着**固定的时序关系**，要严格遵循
- **单向 I/O 方式**，要改变输入/输出状态，必须重写工作方式控制字
- 方式 1 可使用**查询**或**中断**的方式传送数据

# 02 | 并行接口-方式1-输入操作的联络信号1

## 方式 1 输入操作的联络信号

Strobe  
Input Buffer Full

1. 选通信号 STB#：外设→8255A，低电平有效
  - 该信号有效期间，数据由外设传送至 8255A
  - 8255A 分别使用  $PC_4$ 、 $PC_2$  作为  $STB_A\#$  和  $STB_B\#$
2. 输入缓冲器满信号 IBF：由 8255A 输出，高电平有效
  - 该信号有效表示当前 8255A 的输入缓冲器中有新数据，等待 CPU 读取，并且禁止外设再次送入数据
  - 作用是：应答外设的 STB# 信号；或供 CPU 查询的状态位
  - 8255A 分别使用  $PC_5$ 、 $PC_1$  作为  $IBF_A$  和  $IBF_B$

# 02 | 并行接口-方式1-输入操作的联络信号2

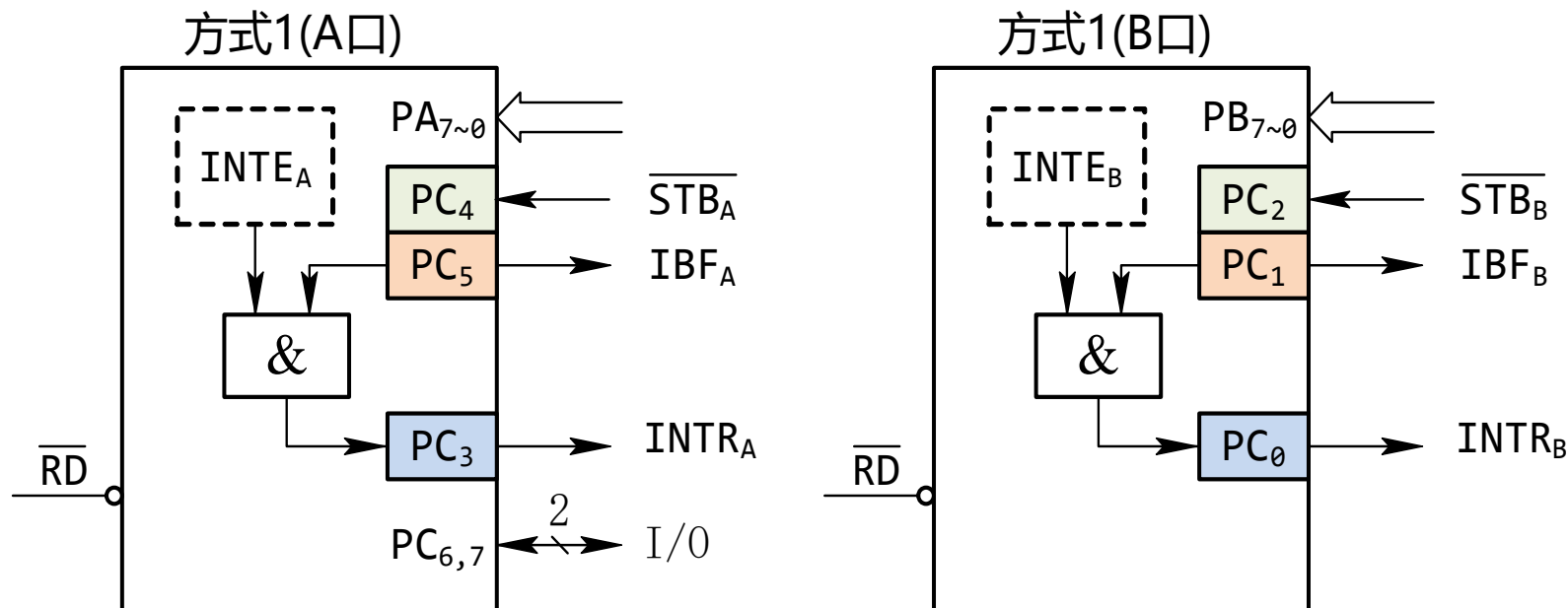
## 方式 1 输入操作的联络信号

Interrupt Request  
Interrupt Enable

3. 中断请求信号 INTR: **8255A → CPU, 高电平有效**
  - 该信号由 8255A 的内部控制逻辑产生, 用于中断传送方式下, 向 CPU 发出输入请求
  - 8255A 分别使用  $PC_3$ 、 $PC_0$  作为  $INTR_A$  和  $INTR_B$
4. 中断允许信号 INTE: **8255A 内部控制信号, 高电平有效**
  - $INTE=1$  时, 8255A 才有可能产生 INTR 请求; 否则不会产生 INTR 信号
  - 对 INTE 的**设置位同 STB# 信号位** ( $PC_4$ 、 $PC_2$ ), 属于 8255A 的**内部操作**, 不影响 STB# 信号的状态

## 02 | 并行接口-方式1-输入联络信号示意图

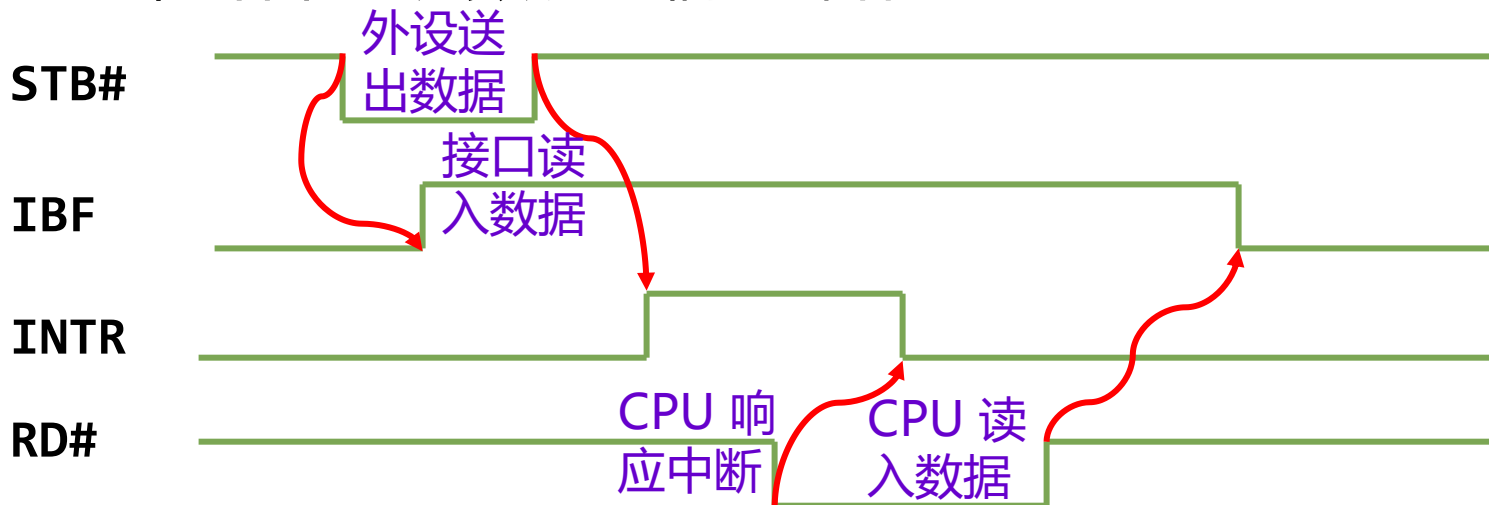
### 方式 1 输入联络信号示意图



- **中断机制**: 当  $STB\#=1$  (外设输入数据已结束)、 $IBF=1$  (输入缓冲器中有新数据)、 $INTE=1$  (允许中断) 时, 使  $INTR$  有效

## 02 | 并行接口-方式1-输入操作的工作时序

1. 外设将准备好的数据送至数据线上，并发有效的 **STB#** 信号
2. **STB#** 下降沿将数据送入 8255A 的输入缓冲器中，并 8255A 置 **IBF** 有效，禁止外设再次输入数据
3. **STB#** 自动维持一段时间后，其上升沿使 8255A 内部逻辑生成 **INTR** 中断请求信号，请求 CPU 执行中断，读入数据
4. CPU 在中断服务时执行读操作，**RD#** 前沿撤销 **INTR**，后沿撤销 **IBF**，结束一次数据的输入操作



# 02 | 并行接口-方式1-输出操作的联络信号1

## 方式 1 输出操作的联络信号

Output Buffer Full  
Acknowledge

1. 输出缓冲器满信号 OBF#：**由 8255A 输出，低电平有效**
  - 该信号有效，表示 8255A 的输出缓冲寄存器中有新数据等待外设读取
  - 用于**通知外设**取走新数据；或**供 CPU 查询状态**
  - 8255A 分别使用  $PC_7$ 、 $PC_1$  作为  $OBF_A\#$  和  $OBF_B\#$
2. 外部应答信号 ACK#：**外设 → 8255A，低电平有效**
  - 对 OBF# 的应答信号，8255A 收到该信号时，表示外设已从 8255A 中取走数据，结束本次输出操作
  - 8255A 分别使用  $PC_6$ 、 $PC_2$  作为  $ACK_A\#$  和  $ACK_B\#$

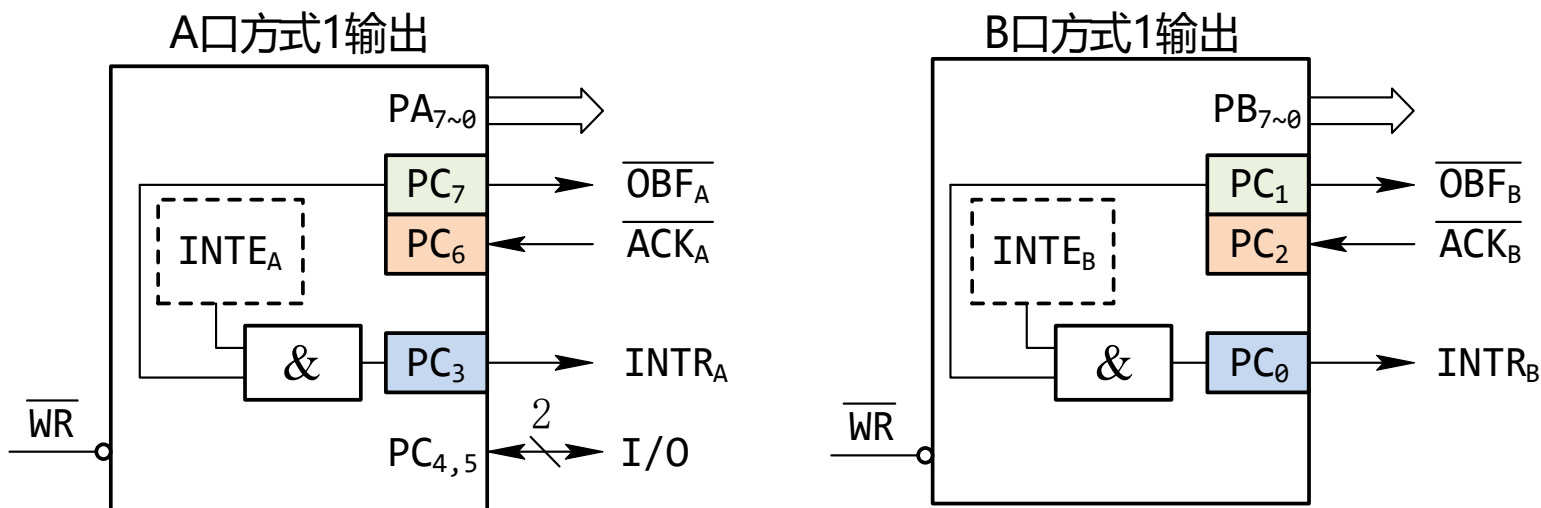
# 02 | 并行接口-方式1-输出操作的联络信号2

## 方式 1 输出操作的联络信号

3. 中断请求信号 INTR: **8255A → CPU, 高电平有效**
  - 该信号由 8255A 的内部控制逻辑产生, 用于中断传送方式下, 向 CPU 发出输出数据的请求
  - 8255A 分别使用  $PC_3$ 、 $PC_0$  作为  $INTR_A$  和  $INTR_B$
4. 中断允许信号 INTE: **8255A 内部控制信号, 高电平有效**
  - $INTE=1$  时, 8255A 才有可能产生 INTR 请求; 否则不会产生 INTR 信号
  - 对 INTE 的**设置位同 ACK# 信号位** ( $PC_6$ 、 $PC_2$ ), 属于 8255A 的**内部操作**, 不影响 ACK# 信号的状态

# 02 | 并行接口-方式1-输出联络信号示意图

## 方式 1 输出联络信号示意图

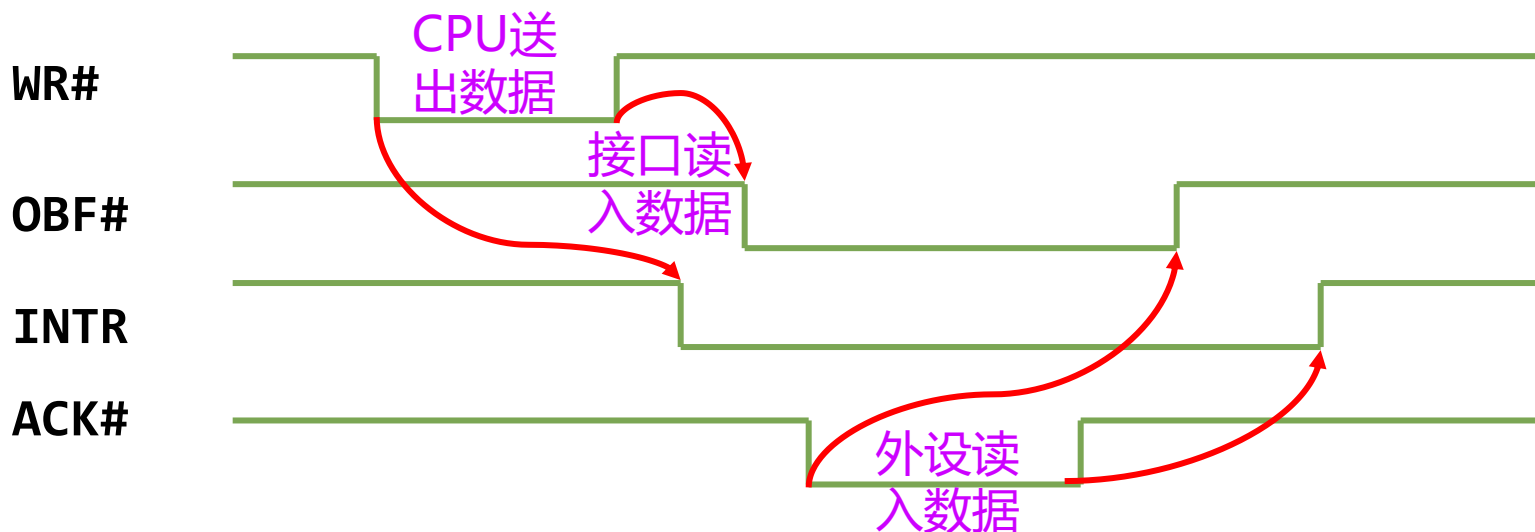


- **中断机制**: 当  $OBF\#=1$  (输出缓冲器空)、 $ACK\#=1$  (外设结束上一数据的输出操作)、 $INTE=1$  (允许中断)、 $WR\#=1$  (CPU 完成一次数据的写操作) 时,  $INTR$  有效



## 02 | 并行接口-方式1-输出操作的工作时序

1. CPU 输出数据到 8255A 的输出缓冲器, **WR# 有效**, 其前沿撤销 INTR 信号, **后沿设置 OBF# 信号**
2. 外设得到 OBF# 有效信号后, **读取数据**, 并发出 **ACK# 应答信号**, 通知 8255A
3. **ACK# 前沿撤销 OBF#**, 即表示输出缓冲器空, 可以**开始新的输出操作**, 后沿使 8255A 内部逻辑**产生新的中断请求 INTR**



## 02 | 并行接口-方式1-C口状态

### 方式 1 下的 C 口状态

- 方式 1 下, 某些 PC 位作为**固定**的联络信号配合 A、B 口工作
- 对于 C 口中不使用的其他位, 仍然可以作为数据端口

A、B 口**输入**操作时, C 口各位的状态

|     |     |                  |                   |                   |                   |                  |                   |
|-----|-----|------------------|-------------------|-------------------|-------------------|------------------|-------------------|
| I/O | I/O | IBF <sub>A</sub> | STB# <sub>A</sub> | INTR <sub>A</sub> | STB# <sub>B</sub> | IBF <sub>B</sub> | INTR <sub>B</sub> |
|-----|-----|------------------|-------------------|-------------------|-------------------|------------------|-------------------|

A、B 口**输出**操作时, C 口各位的状态

|                   |                   |     |     |                   |                   |                   |                   |
|-------------------|-------------------|-----|-----|-------------------|-------------------|-------------------|-------------------|
| OBF# <sub>A</sub> | ACK# <sub>A</sub> | I/O | I/O | INTR <sub>A</sub> | ACK# <sub>B</sub> | OBF# <sub>B</sub> | INTR <sub>B</sub> |
|-------------------|-------------------|-----|-----|-------------------|-------------------|-------------------|-------------------|

A 口**输入**、B 口**输出**操作时, C 口各位的状态

|     |     |                  |                   |                   |                   |                   |                   |
|-----|-----|------------------|-------------------|-------------------|-------------------|-------------------|-------------------|
| I/O | I/O | IBF <sub>A</sub> | STB# <sub>A</sub> | INTR <sub>A</sub> | ACK# <sub>B</sub> | OBF# <sub>B</sub> | INTR <sub>B</sub> |
|-----|-----|------------------|-------------------|-------------------|-------------------|-------------------|-------------------|

若8255A的B口工作于方式1下，则以下PC位中一定不能用于数据传送的是（ ）

☒ A  $PC_1$

☐ B  $PC_3$

☐ C  $PC_5$

☐ D  $PC_7$

## 02 | 并行接口-选通输入/输出方式举例

示例：利用 8255A 的 A 口方式 1 设计并行打印机接口

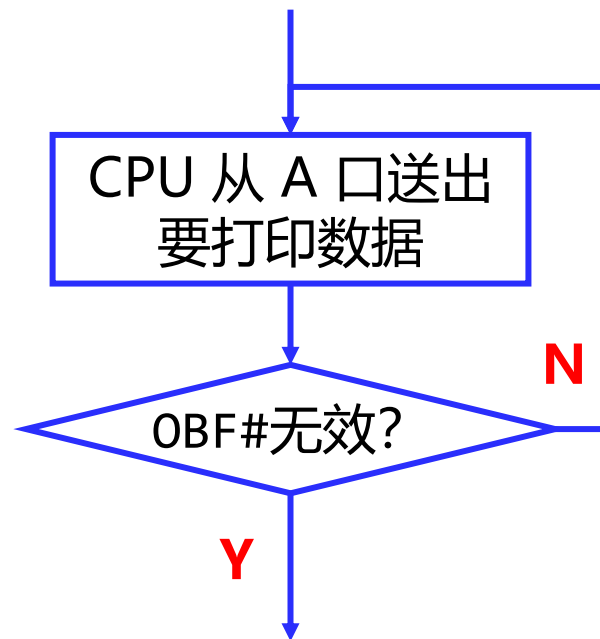
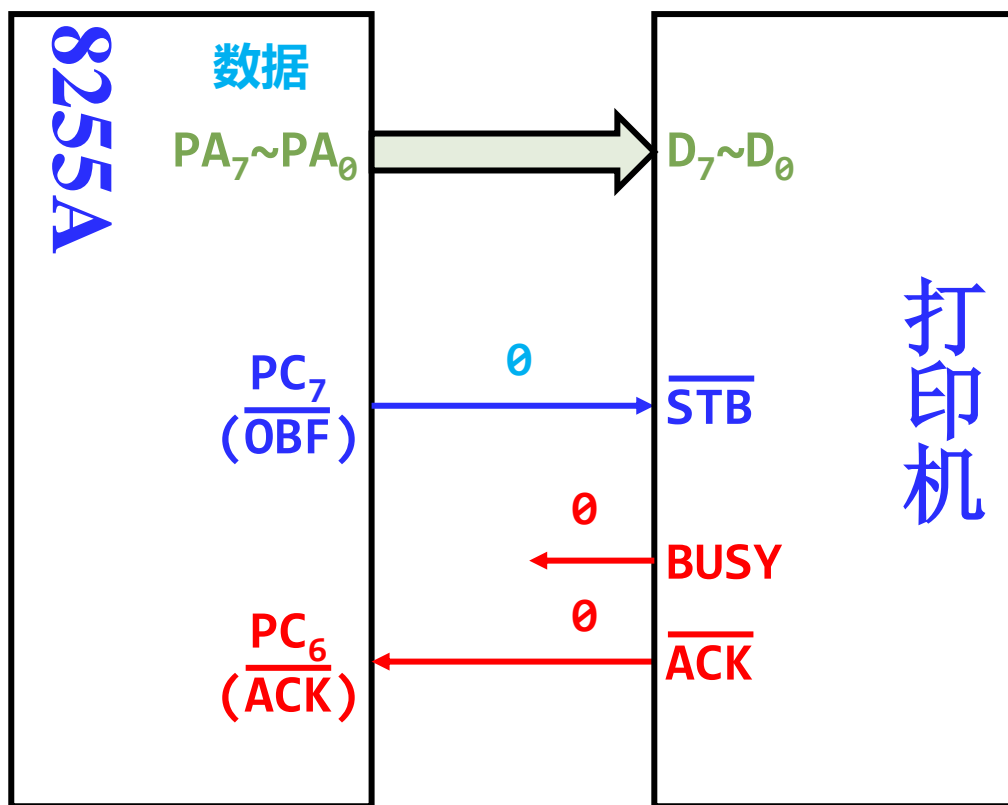
要求：CPU 采用查询方式将存放在缓冲区 BUF 的 256 个字符通过接口送到打印机打印

- A 口工作于方式 1，作为输出口，指定的联络信号
  - $OBF_A\#$  ( $PC_7$ )、 $ACK_A\#$  ( $PC_6$ )、 $INTR_A$  ( $PC_3$ )
- 打印机的联络信号
  - BUSY、STB#、ACK#
- 与方式 0 下的工作情况相似，只是在方式 1 下必须使用指定 PC 位的联络信号，而不能随意自定义
  - 方法 1：将 A 口的  $OBF\#$ 、 $ACK\#$  信号作为打印机的联络信号
  - 方法 2：自定义打印机的选通信号

## 02 | 并行接口-选通输入/输出方式举例

示例：利用 8255A 的 A 口方式 1 设计并行打印机接口  
打印机与接口之间的连接图

- 将 A 口的 OBF#、ACK# 信号作为打印机的联络信号



## 02 | 并行接口-选通输入/输出方式举例

示例：利用 8255A 的 A 口方式 1 设计并行打印机接口  
8255A 方式 1 下的查询传送源程序段

```
MOV AL, 10100000B
OUT 63H, AL
LEA SI, BUF
MOV CX, 256
NEXT: MOV AL, [SI]
      OUT 60H, AL
      INC SI
AGAIN: IN AL, 62H
      TEST AL, 80H
      JZ AGAIN
      LOOP NEXT
```

8255A初始化

指针初始化

CPU 输出数据  
到 8255A 的  
A 口

CPU 查询接口的输出状态  
(OBF#, PC<sub>7</sub>), 判断是否  
开始下一个数据的输出操作

# 02 | 并行接口-子目录

## 8255A 并行接口

- 并行接口的概念
- 8255A 可编程并行接口
- 8255A 的控制字
- 8255A 的工作方式和编程
  - 基本输入/输出方式（方式0）
  - 选通输入/输出方式（方式1）
  - 双向输入/输出方式（方式2）
- 8255A 在微机中的应用

## 02 | 并行接口-方式2-双向输入/输出方式

### 方式 2 的工作特点

带选通的双向 IO 方式

- 双向选通输入/输出方式，只有 A 口可以使用
- 设有专用的输入/输出联络信号线 ( $IBF_A$ 、 $STB\#_A$ 、 $OBF\#_A$ 、 $ACK\#_A$ 、 $INTR_A$ )，各联络信号的时序关系同方式 1
- 工作于方式 2 下 A 口既可以作数据输入端口，也可以作数据输出端口
- 可以用于查询和中断方式的数据传送



## 02 | 并行接口-方式2-联络信号及状态字

### 方式 2 下的联络信号及状态字

A 口工作于方式 2 下, C 口状态

|                   |                   |                  |                   |                   |     |     |     |
|-------------------|-------------------|------------------|-------------------|-------------------|-----|-----|-----|
| OBF# <sub>A</sub> | ACK# <sub>A</sub> | IBF <sub>A</sub> | STB# <sub>A</sub> | INTR <sub>A</sub> | I/O | I/O | I/O |
|-------------------|-------------------|------------------|-------------------|-------------------|-----|-----|-----|

A 口工作于方式 2 下, 控制字为

|   |   |   |    |    |   |   |   |
|---|---|---|----|----|---|---|---|
| 1 | 1 | 0 | 无效 | 无效 | * | * | * |
|---|---|---|----|----|---|---|---|

A口工作于方式 2 下, B口工作于方式 1 下, C口的状态

|                   |                   |                  |                   |                   |                   |                   |                   |
|-------------------|-------------------|------------------|-------------------|-------------------|-------------------|-------------------|-------------------|
| OBF# <sub>A</sub> | ACK# <sub>A</sub> | IBF <sub>A</sub> | STB# <sub>A</sub> | INTR <sub>A</sub> | STB# <sub>B</sub> | IBF# <sub>B</sub> | INTR <sub>B</sub> |
| OBF# <sub>A</sub> | ACK# <sub>A</sub> | IBF <sub>A</sub> | STB# <sub>A</sub> | INTR <sub>A</sub> | ACK# <sub>B</sub> | OBF# <sub>B</sub> | INTR <sub>B</sub> |

# 02 | 并行接口-子目录

## 8255A 并行接口

- 并行接口的概念
- 8255A 可编程并行接口
- 8255A 的控制字
- 8255A 的工作方式和编程
  - 基本输入/输出方式 (方式0)
  - 选通输入/输出方式 (方式1)
  - 双向输入/输出方式 (方式2)
- 8255A 在微机中的应用

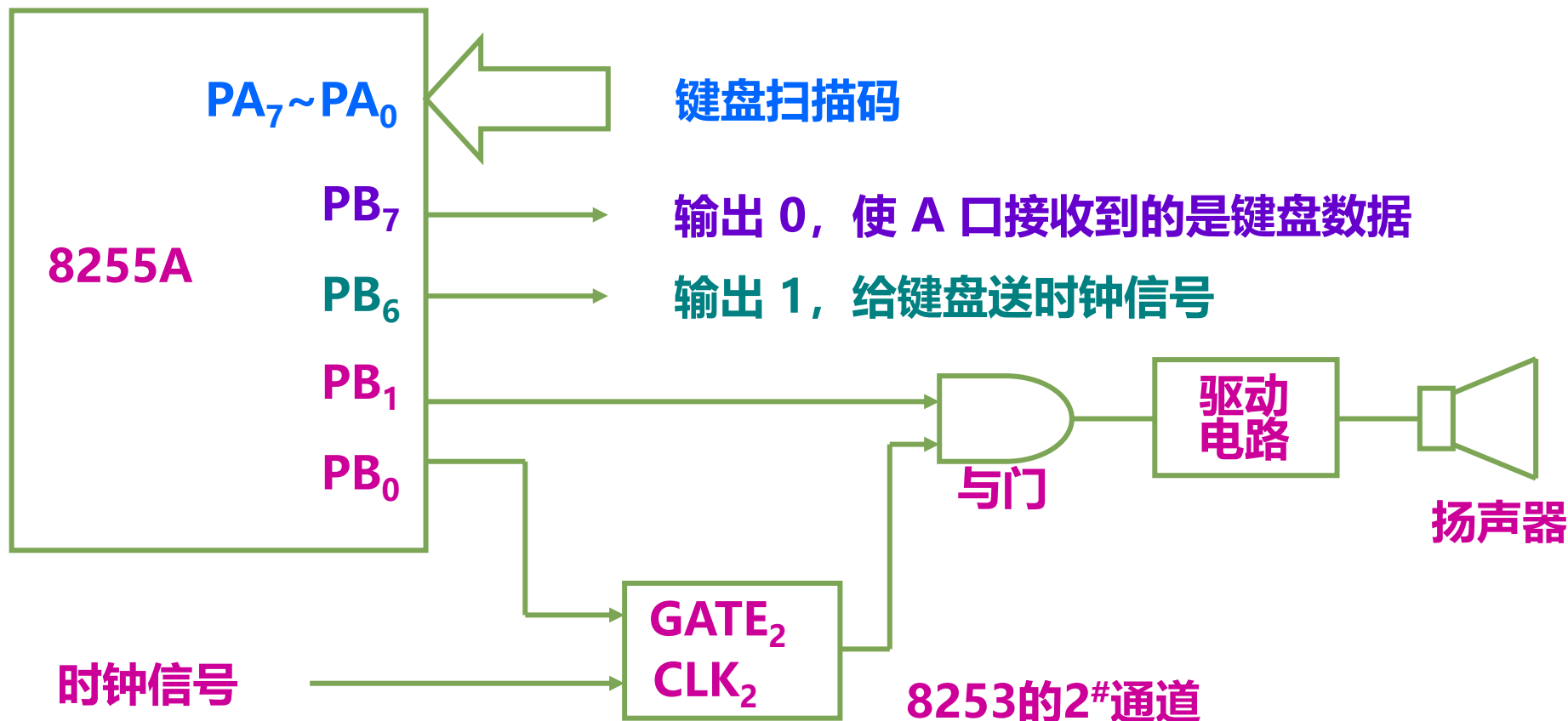
# 02 | 并行接口-8255A在微机系统中的应用

## 8255A 在 IBM PC/XT 机系统中的应用

- 使用一片 8255A 工作于方式 0 下
- A 口：输入/输出
  - 加电自检时为输出，输出当前检测部件的标志信号
  - 正常工作时为输入，用于读取键盘的扫描码
- B 口：输出一些控制信号
  - 键盘管理、RAM 和 I/O 通道检验、扬声器的控制等
- C 口：输入方式
  - 高 4 位为状态测试位，低 4 位读取系统配置开关状态

# 02 | 并行接口-8255A在微机系统中的应用

## 8255A 在 IBM PC/XT 机系统中的应用



# 本节小结

---

- 了解并行通信的基本概念及特点
- 熟悉8255A芯片的外部引脚，清楚各类的引脚的连接对象
- **掌握**8255A芯片的命令字和状态字，能够完成芯片的初始化编程
- **理解**8255A芯片的各种应用
  - 理解 8255A 在方式 0、方式 1 下的查询传送方式区别

# 目录

---

- 01 定时/计数器 8253/8254
- 02 并行通信接口 8255A
- 03 串行通信芯片 8251A

## 02 | 上节回顾-定时/计数器8255A

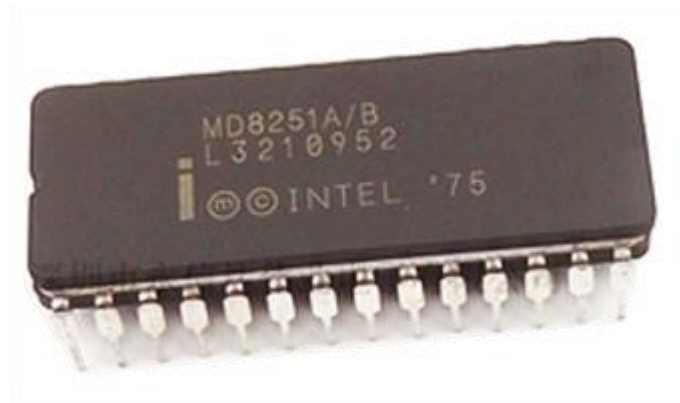
### 简答题

- 8255A有几个端口？它们分别是什么？
- 8255A的工作方式有哪些？它们的特点分别是什么？
- 请简述8255A工作方式控制字格式？如何对8255A进行初始化。

# 03 | 串行接口-子目录

## 8251A 串行接口

- 串行接口的概念
- 8251A 可编程串行接口
  - 8251A 的主要特性
  - 8251A 的外部引脚特性
  - 8251A 的内部结构
- 8251A 的编程及应用
  - 8251A 的命令字与状态字
  - 8251A 的初始化编程
  - 8251A 的应用举例

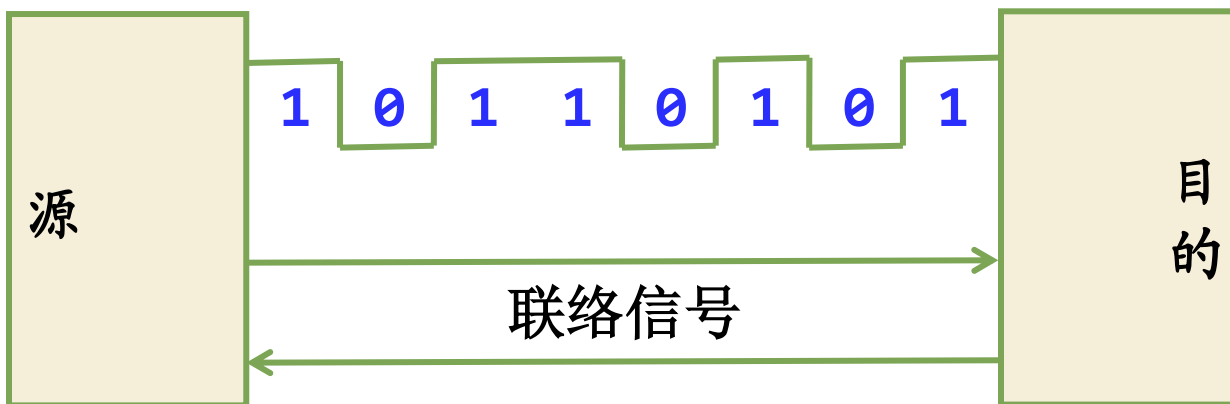




# 03 | 串行接口-串行接口相关概念

## 串行接口相关概念

- 串行通信和串行接口
- 串行数据传送方式：单工 (Simplex)、半双工 (Half Duplex)、全双工 (Full Duplex)、多工 (multiplex)
- 串行通信方式：异步通信、同步通信
- 数据传输率
- 信号的调制和解调



## 串行通信

- 使用**一根/对信息传输线**，将数据、控制、状态等信息按顺序逐位传送
- 串行传送的数据有**固定的格式**来区分信息类型
- 串行通信的双方要**约定通信传送的波特率**

## 串行通信的特点

- 串行通信适用于**远距离传送**
- 传送效率比并行通信**低**，但**容易进行差错控制**

## 串行接口

- 串行接口中需要进行数据的**串-并转换**
  - 利用接口内部的**移位逻辑**实现

## 03 | 串行接口 - 串行数据传送方式

### 单工 (Simplex)

- 收发双方由**一根单向的信息线**连接

### 半双工 (Half Duplex)

- 收发双方通过**一根双向的信息线**连接

### 全双工 (Full Duplex)

- 收发双方通过**两根单向信息线**连接

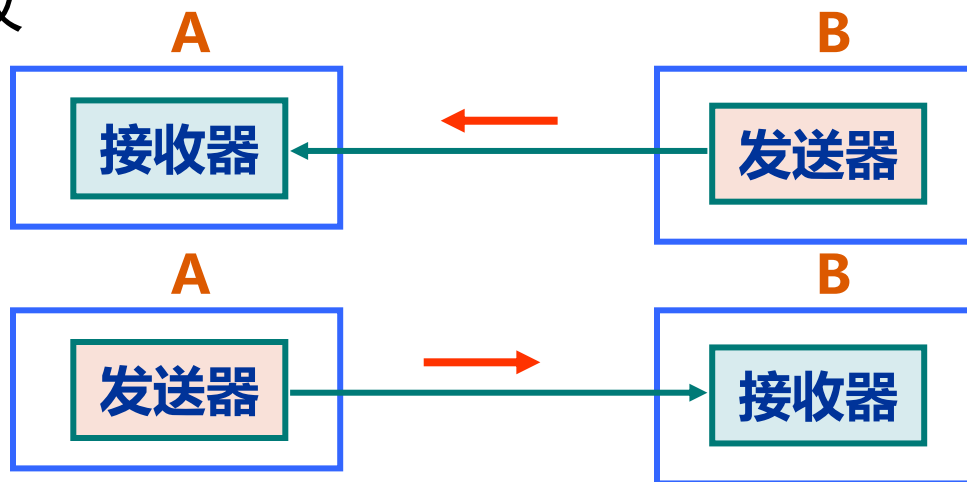
### 多工 (Multiplex)

- 通过将一个信道划分为**若干个频带或时间片**的复用技术，从而使多路信号同时共享信道

# 03 | 串行接口 - 单工数据传送方式

## 单工 (Simplex)

- 收发双方通过**一根单向的信息线**连接，**只能**进行数据的**发送或接收**，一旦连接确定数据的传送方向即不可更改



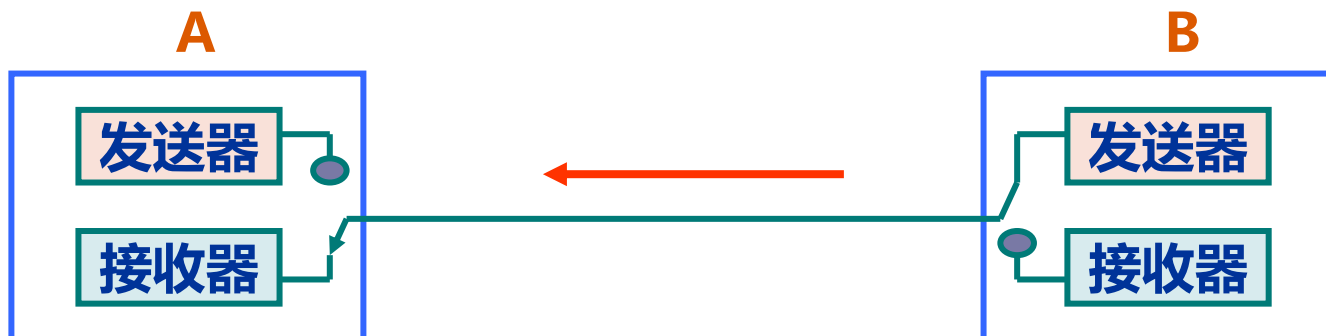
## 特点

- 两站点之间仅有一**根单向**的信息线
- 每个站点中仅需要一个发送器，或接收器

# 03 | 串行接口-半双工数据传送方式

## 半双工 (Half Duplex)

- 收发双方通过**一根双向的信息线**连接，**既可发送数据又可接收数据**，但通信双方不能同时收发数据



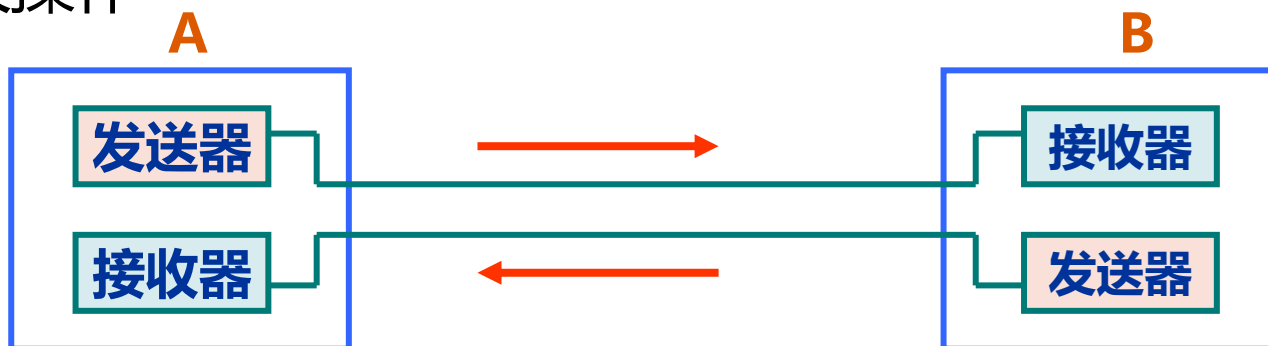
## 特点

- 两站点之间只有一根**双向**信息线
- 每端都有发送器和接收器，需有**收/发切换电子开关**
- 因有切换，会产生时间**延迟**

# 03 | 串行接口-全双工数据传送方式

## 全双工 (Full Duplex)

- 收发双方通过**两根单向信息线**连接，**分别**负责数据的**发送**和**接收**，通信双方都能在同一时刻进行发送和接收操作



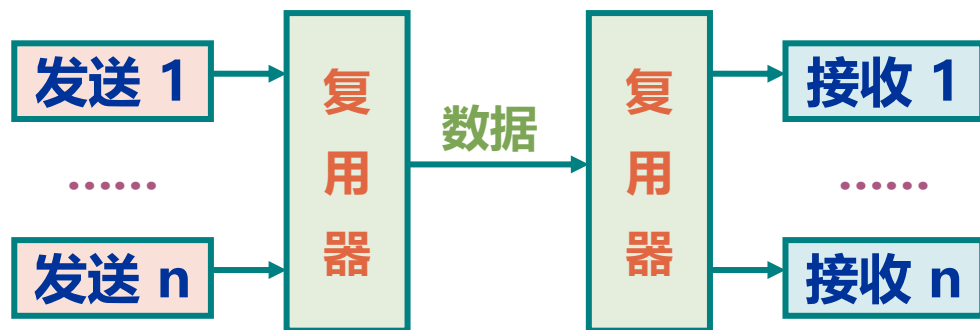
## 特点

- 每一端都有发送器和接收器
- 有**两条相互独立的单向信息**传送线

# 03 | 串行接口-多工数据传送方式

## 多工 (Multiplex)

- 通过将一个信道划分为**若干个频带或时间片**的复用技术，从而使多路信号同时**共享信道**



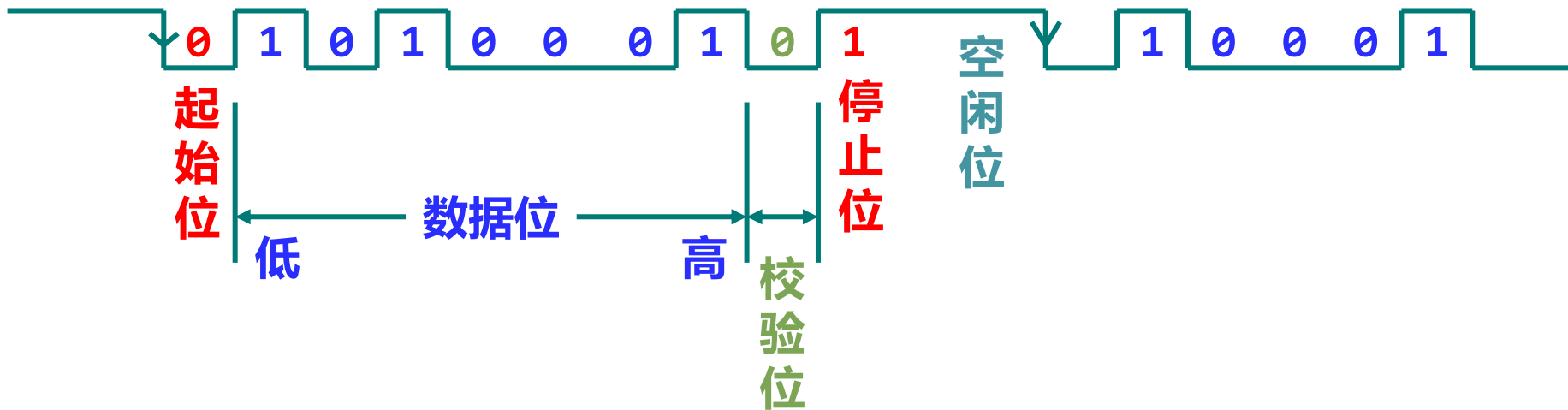
## 特点

- 在**一条线路**上传输不同的信号频率
- 使用**多路复用器或多路集中器**等专用的通信设备

# 03 | 串行接口-串行通信方式-异步通信

## 异步通信

- 以**字符**为单位传送数据，**字符与字符之间**是**异步**传送的，而**位与位之间**则是**同步**传送
- 数据格式：
  - 起始位 (1 位, 值恒为 0)、数据位 (5~8 位)、校验位 (可选)、停止位 (1 位、1.5 位、2 位可选, 值恒为 1)





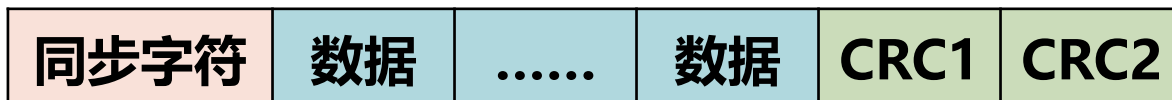
# 03 | 串行接口 - 串行通信方式 - 同步通信

## 同步通信

Cyclic Redundancy Check

- 以**数据块**为单位传送，每个数据块包括多个字符；字符内部的**位传送**和**字符之间的传送**都要**严格同步**
  - 同步方法有**外同步**和**内同步**两种
- 数据格式：
  - 单同步方式：只有一个同步字符
  - 双同步方式：有两个同步字符
  - 外同步方式：没有同步字符，靠**外部时钟**同步

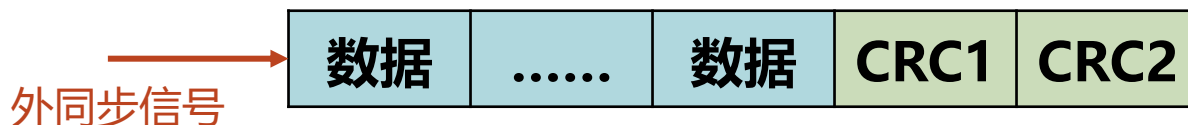
### 单同步方式



### 双同步方式



### 外同步方式



## 03 | 串行接口-异步通信与同步通信的主要区别

### 时钟要求

- 同步通信：发送与接收时钟频率**精确相等**
- 异步通信：发送与接收时钟频率**基本相等**即可

### 控制信息

- 同步通信：要求对**整个数据块**附加帧信息，用于**高速**数据链路
- 异步通信：要求对**每个数据字符**均附加帧信息，用于低速设备，**低速**传送

### 校验方式

- 同步通信：采用 **16 位 CRC 码**，可靠性**高**
- 异步通信：采用 **1 位奇偶校验**，可靠性相对**较低**

# 03 | 串行接口-数据传输率1

## 波特率

- 单位时间内传送二进制数据的位数，即串行通信的数据传输率

## 单位

- 1 波特 = 1 bit/s (位/秒) **两相调制**

## 常用的标准波特率

- 110、300、600、1200、2400、4800、9600、19200b/s
- 注意：波特率与字符传输率是不同的

## 03 | 串行接口-数据传输率2

### 波特率因子

- 每传送一位二进制数据所需要的**时钟脉冲个数**

### 波特率、波特率因子、时钟频率的关系

- **时钟频率 = 波特率 × 波特率因子**

### 示例

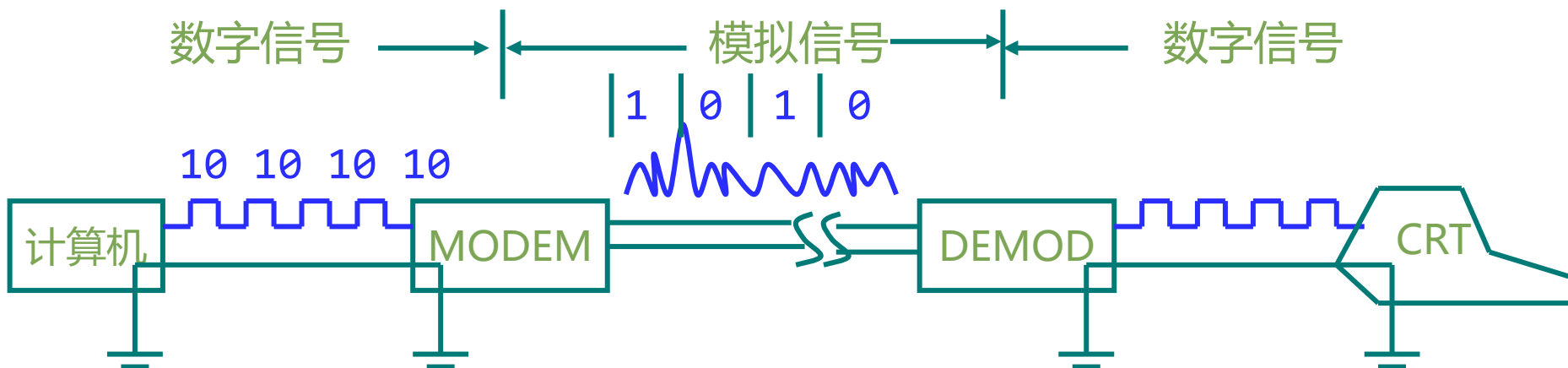
- 给定时钟频率  $f=110.2 \text{ kHz}$ ，若选择波特率因子为 16，则该串行通信的波特率为多少？
  - 波特率 =  $110.2 \times 10^3 / 16 = 1200 \text{ bps}$

# 03

## 串行接口-信号的调制与解调

### 调制与解调

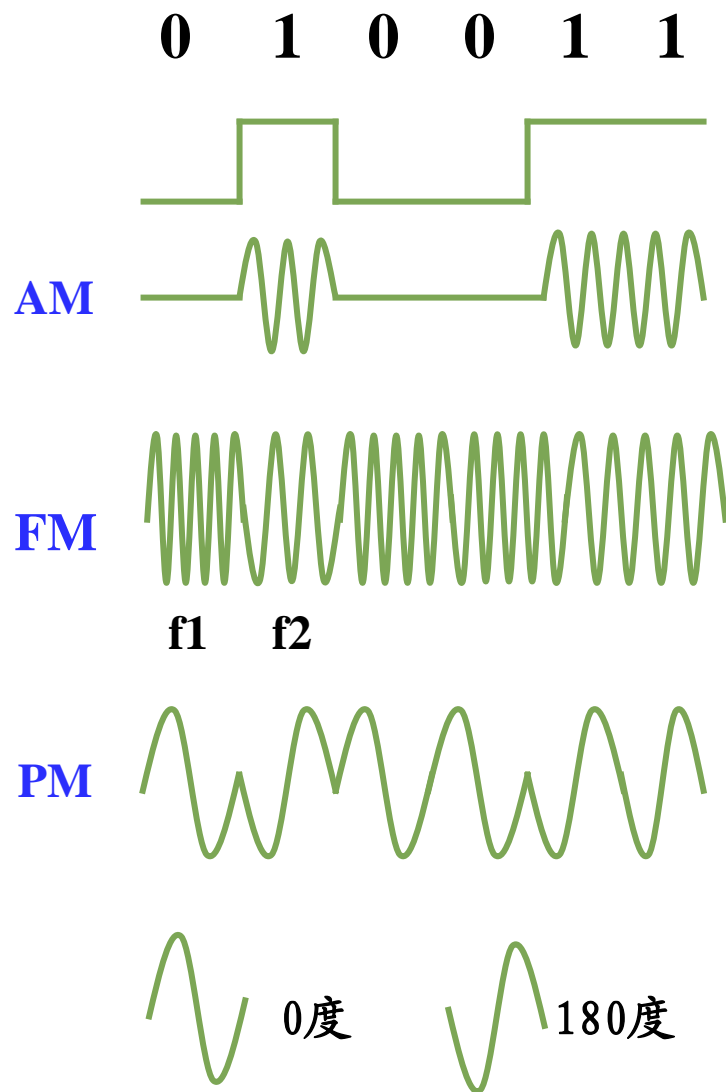
- **调制**就是用**基带信号**去控制**载波信号**的某个或几个**参量**的变化，将信息编码在其上，形成**已调信号**传输
  - **调制器** (Modulator) 是一个**波形变换器**，它将数字波形变换成适合于模拟信道传输的波形
- **解调**是调制的反过程，通过具体的方法从**已调信号**的**参量**变化中，恢复原始的**基带信号**
  - **解调器**是一个**波形识别器**，将模拟信号恢复成数字信号



# 03 | 串行接口-信号的调制与解调

## 信号调制方法

- 调幅 (AM)
  - 载波的**振幅**随数字信号而变化
  - 1 — 有载波; 0 — 无载波
- 调频 (FM)
  - 载波的**频率**随数字信号而变化
  - 0 —  $f_1$ ; 1 —  $f_2$
- 调相 (PM)
  - 载波的**初始相位**随数字信号而变化
  - 0 — 相位 0 度; 1 — 相位 180 度



# 03 | 串行接口-子目录

## 8251A 串行接口

- 串行接口的概念
- 8251A 可编程串行接口
  - 8251A 的主要特性
  - 8251A 的外部引脚特性
  - 8251A 的内部结构
- 8251A 的编程及应用
  - 8251A 的命令字与状态字
  - 8251A 的初始化编程
  - 8251A 的应用举例

# 03 | 串行接口-8251A的主要特性

## 8251A 的主要特性

- 8251A 是一个**全双工**双缓冲器的可编程串行接口芯片
- 同步方式
  - 波特率为 0~64 kbps, 可自动**插入/检测**同步字符
  - 可通过编程方式设置**数据格式**、**同步方式**、**同步字符**等
- 异步方式
  - 波特率为 0~110.2 kbps
  - 可通过编程方式设置**数据格式**、**波特率因子**
- 具有发送/接收缓冲器, 可以工作于**全双工**的传输方式
- 具有**出错检测**功能, 能检出**奇偶**、**超越**、**帧格式**等错误



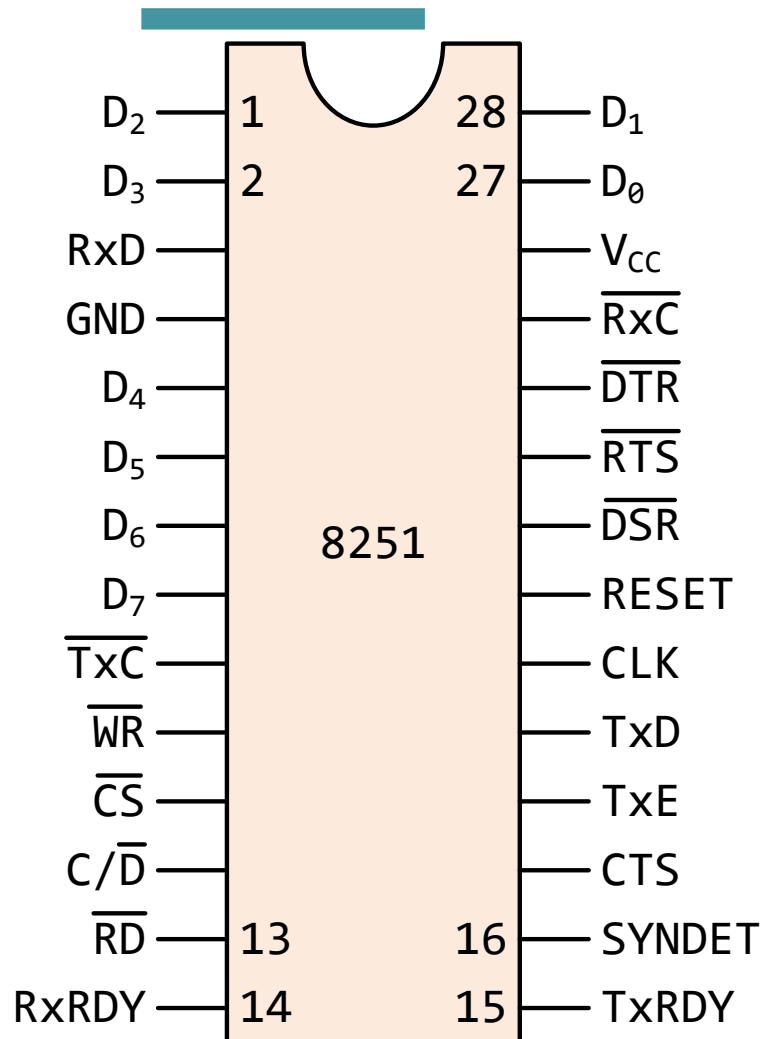
# 03 | 串行接口-子目录

## 8251A 串行接口

- 串行接口的概念
- 8251A 可编程串行接口
  - 8251A 的主要特性
  - 8251A 的外部引脚特性
  - 8251A 的内部结构
- 8251A 的编程及应用
  - 8251A 的命令字与状态字
  - 8251A 的初始化编程
  - 8251A 的应用举例

# 03 | 串行接口-8251A的外部引脚特性

## 8251A 的外部引脚



### 面向 CPU 的引脚:

D<sub>0</sub>~D<sub>7</sub>、RD#、WR#、CS#、RESET、  
C/D: Control/Data

### 状态信号:

TxRDY: Transmitter Ready  
TxE: Transmitter Empty  
RxRDY: Receiver Ready  
SYNDET: SYNC Detect

### 时钟信号:

CLK、  
TxC: Transmitter Clock  
RxC: Receiver Clock

### 调制解调信号:

DSR: Data Set Ready  
DTR: Data Terminal Ready  
RxD: Receive Data  
RTS: Request to Send  
CTS: Clear to Send  
TxD: Transmit Data

# 03 | 串行接口-8251A的外部引脚特性

## 面向 CPU 的引脚

C/D: Control/Data

- 三态双向数据总线  $D_0 \sim D_7$
- 读写信号 RD#、WR#
- 片选信号 CS#
- 复位信号 RESET:
  - 持续 6 个时钟周期的高电平，即使芯片处于待命状态
- 片内寻址的地址线 C/D#: (308H、309H)
  - 1 — 访问 8251A 的命令/状态端口 (写命令、读状态)
  - 0 — 访问 8251A 的数据端口 (读写数据)

## 03 | 串行接口-8251A的外部引脚-状态信号1

发送就绪信号 TxRDY: 高电平有效, 输出

TxRDY: Transmitter Ready  
CTS: Clear to Send

- 当 8251A 的**内部状态允许发送** (TxE 置位), **外设就绪** (CTS#引脚有效), 且**发送缓冲器空**时, 该信号有效
- 有效时, CPU 可将并行数据写入 8251 的发送缓冲器
- CPU 写入数据后, **该信号自动无效**
- 中断方式时, 可作为中断请求信号  
查询方式时, 可作为状态信号
- CPU 查询**状态字的 D<sub>0</sub> 位** (TxRDY 位) 来决定是否传送数据

## 03 | 串行接口-8251A的外部引脚-状态信号2

发送器空信号 TxE (TxEMPTY): 高电平有效, 输出

- 有效时, 8251A 发送器中的数据已送到外设
- **同步方式**下, 若 CPU 不能及时向 8251A 输出一个新的数据, 则该信号变为有效的高电平, 同时发送器在数据输出线上插入**同步字符**, 以填补传输空隙

### 注意区分

- TxRDY=1                      —— **发送缓冲器空**
- TxE=1                         —— **发送并传转换器空**

## 03 | 串行接口-8251A的外部引脚-状态信号3

接收就绪信号 RxRDY：高电平有效，输出

RxRDY: Receiver Ready

- 有效时，外设已将串行数据送入 8251A，并完成转换，等待 CPU 读取
- CPU 读取数据后，**该信号自动无效**
- 若 CPU 不能及时取走这个数据，新接收的数据将其覆盖，发生数据丢失，称为“**超越错**”，反映在状态字中
- 中断方式时，可作为中断请求信号  
查询方式时，可作为状态信号
- CPU 查询**状态字的 D<sub>1</sub> 位** (RxRDY位) 来决定是否传送数据

## 03 | 串行接口-8251A的外部引脚-状态信号4

双功能引脚 SYNDET/BD: 高电平有效

SYNDET: SYNC Detect  
BD: Break Detect

- 同步方式下，作为**同步检测信号 SYNDET**
  - **内同步**时，芯片内部电路搜索同步字符，找到则使该引脚输出高电平，同时设置状态寄存器
  - **外同步**时，片外检测电路检测到同步字符，从该引脚输入高电平，使 8251A 在下一个接收时钟 RxC# 开始装配字符
- 异步方式下，作为**中止符检测信号 BD**
  - 若 8251A 检测到**对方发送的中止符**，则从该引脚输出高电平，同时将设置状态寄存器

## 03 | 串行接口 - 8251A的外部引脚-时钟信号

### 接收器输入时钟 RxC

TxC: Transmitter Clock  
RxC: Receiver Clock

- **同步方式**下，接收方的 RxC 与发送方的 TxC 使用同一个时钟，且 RxC 的频率等于波特率
- **异步方式**下，RxC 的频率可设定为波特率的 1、16 或 64 倍

### 发送器输入时钟 TxC

- 与接收器时钟 RxC 的规定相同

### 工作时钟 CLK

- 为芯片内部电路提供定时，不用做收发数据
- 时钟周期范围：0.42  $\mu\text{s}$  ~ 1.35  $\mu\text{s}$



## 03 | 串行接口-8251A的外部引脚-调制解调信号1

**数据终端就绪信号 DTR#：低电平有效，8251A → 外设**

- 有效时，表示 8251A 准备就绪，**可以接收数据**

8251A  
接收

**数据装置就绪信号 DSR#：低电平有效，外设 → 8251A**

- 有效时，表示调制解调器准备就绪，**完成输入数据的转换**
- 作为 DTR# 的应答信号

**接收数据线 RxD：输入**

- 以上联络信号就绪后，通过 RxD **输入数据**，数据进入 8251A 之后，变为并行方式

|      |                     |
|------|---------------------|
| DSR: | Data Set Ready      |
| DTR: | Data Terminal Ready |
| RxD: | Receiver Data       |

## 03 | 串行接口-8251A的外部引脚-调制解调信号2

**请求发送信号 RTS#：低电平有效， 8251A → 外设**

- 有效时，8251A 已准备好发送数据
- 可由软件定义

8251A  
发送

**清除请求发射信号 CTS#：低电平有效， 外设 → 8251A**

- 有效时，表示调制解调器已准备好接收数据
- 作为 RTS# 的响应信号，通知 8251A 发送数据

**发送数据线 TxD：输出**

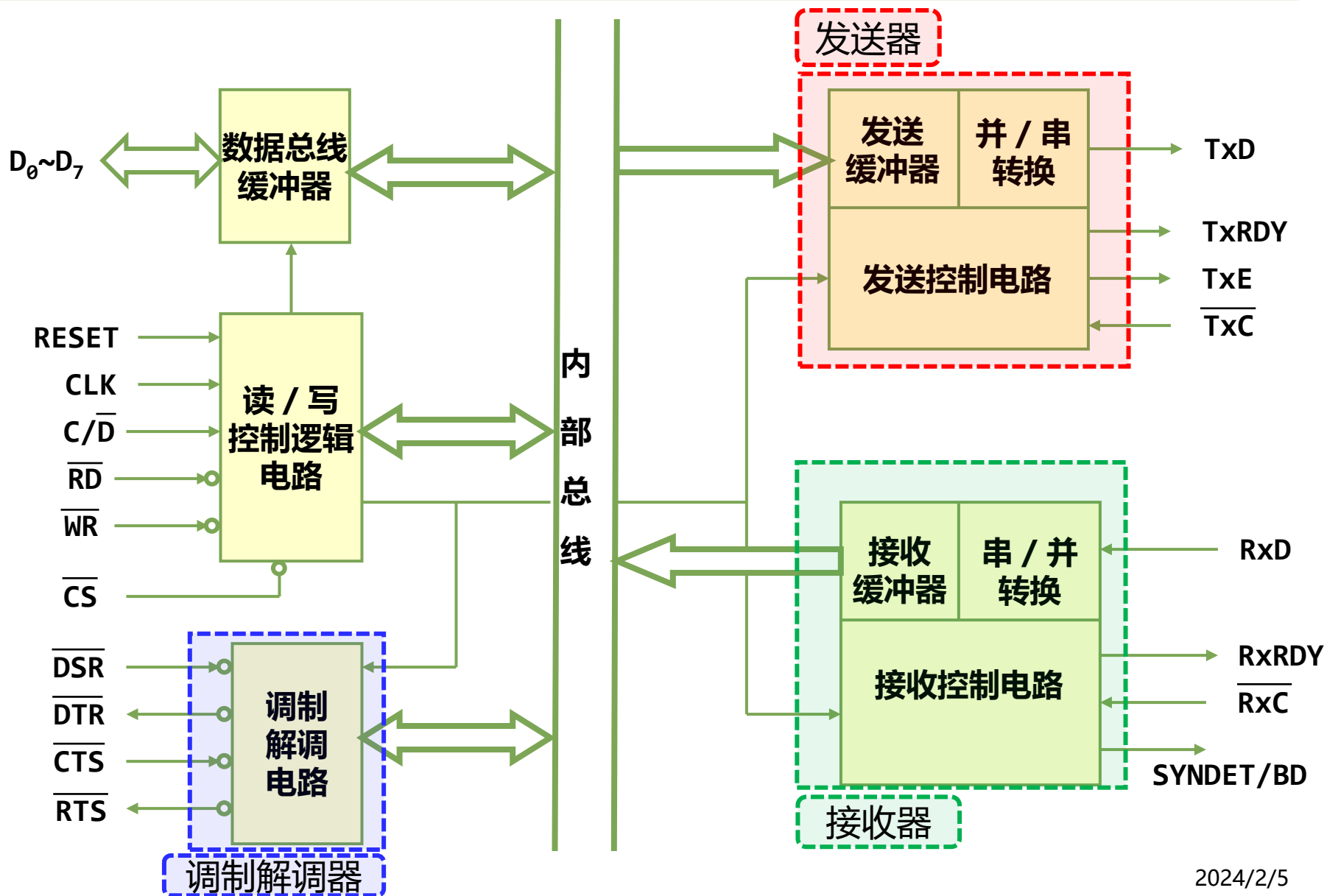
- 以上联络信号就绪后，通过 TxD 输出串行数据至外设

|      |                 |
|------|-----------------|
| RTS: | Request to Send |
| CTS: | Clear to Send   |
| TxD: | Transmit Data   |

## 8251A 串行接口

- 串行接口的概念
- 8251A 可编程串行接口
  - 8251A 的主要特性
  - 8251A 的外部引脚特性
  - 8251A 的内部结构
- 8251A 的编程及应用
  - 8251A 的命令字与状态字
  - 8251A 的初始化编程
  - 8251A 的应用举例

# 03 | 串行接口-8251A的内部结构



## 03 | 串行接口-8251A的内部结构-接收器

### 电路组成

- 接收缓冲器、接收移位寄存器（串-并转换电路）、接收控制电路
- 双缓冲结构

### 有关接收工作

- 当命令控制字中的“允许接收” RxE 和“数据终端准备好” DTR 有效时，接收器开始监视 RxD 上串行数据
- 接收数据对同步和异步两种方式采用不同的处理过程，并将接收到的串行数据转换成并行数据，存放在接收缓冲器中
- 接收缓冲器接收到数据，即设置“接收准备好” RxRDY 状态信号，通知 CPU 读取数据

## 03 | 串行接口-8251A的内部结构-发送器

### 电路组成

- 发送缓冲器、发送移位寄存器（并-串转换电路）、发送控制电路

### 有关发送工作

- TxRDY 有效时，CPU 将并行数据**写入发送数据缓冲器**
- 当**外设就绪**后，发回有效的 CTS# 信号，若命令控制字中的 TxEN 有效，则采用同步、异步方式将数据封装并发送
- 数据发送结束，使 TxE 引脚有效，CPU 可**再次写入下一个数据**

## 03 | 串行接口-8251A的内部结构-调制解调电路

### 作用

- 实现 8251A 与外界之间的联络控制

### 远距离传送时

- 提供与调制解调器的联络应答信号

### 近距离传送时

- 提供与外设的联络应答信号

## 8251A 串行接口

- 串行接口的概念
- 8251A 可编程串行接口
  - 8251A 的主要特性
  - 8251A 的外部引脚特性
  - 8251A 的内部结构
- 8251A 的编程及应用
  - 8251A 的命令字与状态字
  - 8251A 的初始化编程
  - 8251A 的应用举例



## 03 | 串行接口-8251的命令字与状态字

### 方式字

- 对 8251A 的**工作方式进行选择**，约定双方的通信方式，数据格式，传送速率等问题

### 命令字

- 确定 8251A 的**实际操作类型**，迫使 8251A 进入某种操作或工作状态，以便接收或发送数据

### 状态字

- 保存能反映 8251A **当前工作状态**的状态字，包括数据收发的就绪状态、同步状态、出错信息等内容

# 03 | 串行接口-方式控制字格式

## 方式控制字格式

Even Parity

Stop Bits

Parity Enable

Length

Baud Rate Factor



|    | 同步方式       | 异步方式        |
|----|------------|-------------|
| 00 | 内同步<br>双同步 | 无效          |
| 01 | 外同步<br>双同步 | 1位停止<br>位   |
| 10 | 内同步<br>单同步 | 1.5位停<br>止位 |
| 11 | 外同步<br>单同步 | 2位停止<br>位   |

| 校验选择 |     |
|------|-----|
| *0   | 无校验 |
| 01   | 奇校验 |
| 11   | 偶校验 |

| 数据长度 |     |
|------|-----|
| 00   | 5 位 |
| 01   | 6 位 |
| 10   | 7 位 |
| 11   | 8 位 |

| 同、异步方式 |                  |
|--------|------------------|
| 00     | 同步方式             |
| 01     | 异步方式<br>波特率因子为1  |
| 10     | 异步方式<br>波特率因子为16 |
| 11     | 异步方式<br>波特率因子为64 |

## 03 | 串行接口-方式控制字格式举例1

示例1：异步通信中，若帧数据格式为：字符长度 8 位，停止位 2 位，奇校验，波特率因数是 16

| S <sub>1</sub> | S <sub>0</sub> | EP | PEN | L <sub>1</sub> | L <sub>0</sub> | B <sub>1</sub> | B <sub>0</sub> |
|----------------|----------------|----|-----|----------------|----------------|----------------|----------------|
| 1              | 1              | 0  | 1   | 1              | 1              | 1              | 0              |

### ● 初始化程序段

```
MOV DX, 309H ; 8251命令端口地址
MOV AL, 0DEH ; 异步工作方式字
OUT DX, AL
```

## 03 | 串行接口-方式控制字格式举例2

示例2：同步通信中，若帧数据格式为：字符长度 8 位，双同步字符，内同步方式，奇校验

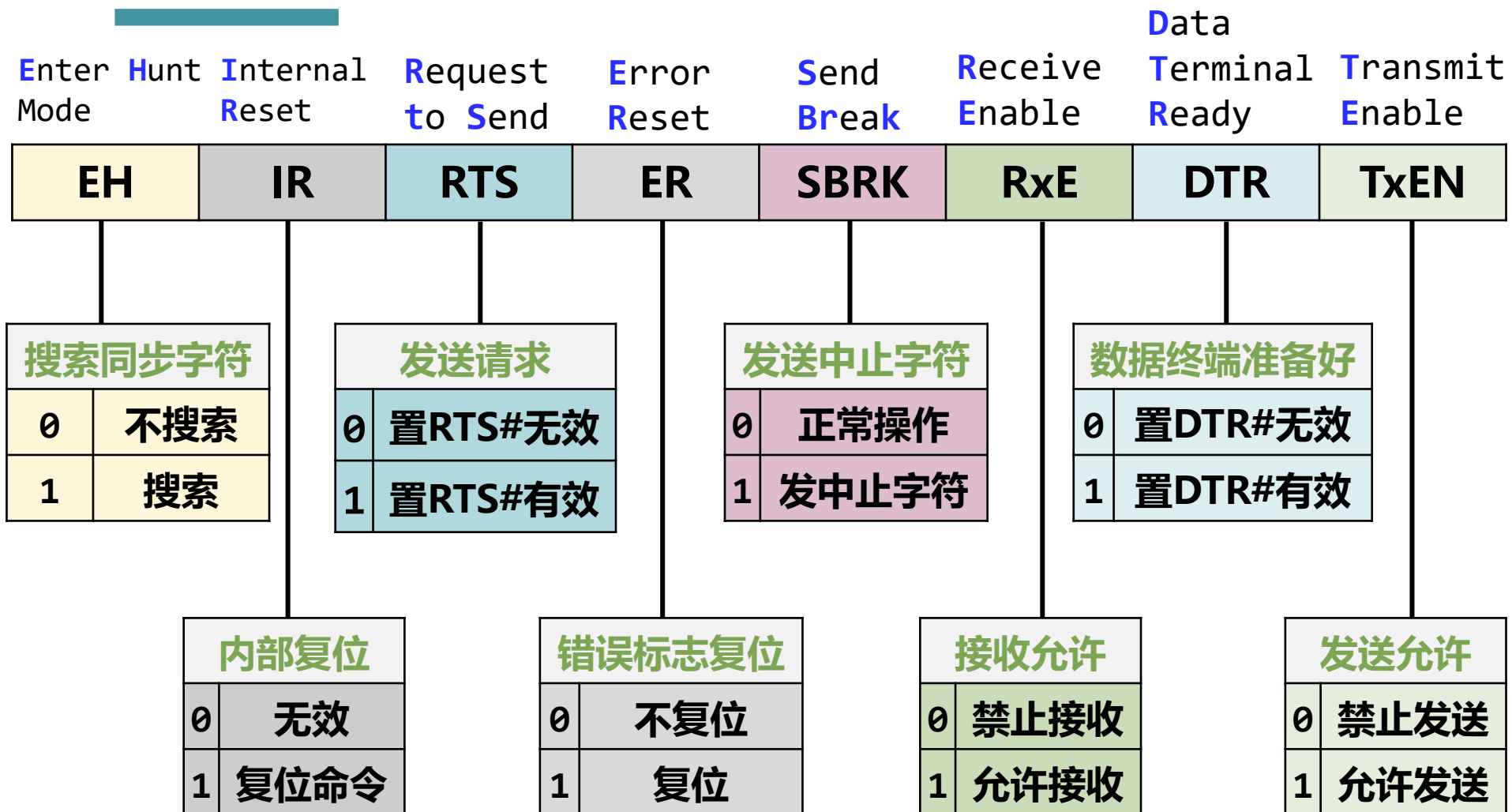
| S <sub>1</sub> | S <sub>0</sub> | EP | PEN | L <sub>1</sub> | L <sub>0</sub> | B <sub>1</sub> | B <sub>0</sub> |
|----------------|----------------|----|-----|----------------|----------------|----------------|----------------|
| 0              | 0              | 0  | 1   | 1              | 1              | 0              | 0              |

### ● 初始化程序段

```
MOV DX, 309H ; 8251命令端口地址
MOV AL, 1CH ; 同步工作方式字
OUT DX, AL
```

# 03 | 串行接口-命令字格式

## 命令控制字格式



## 03 | 串行接口-关于命令字

### D<sub>0</sub>位 (TxEN)、D<sub>2</sub>位 (RxE)

- 分别可作为发送和接收的**允许控制位**，0 — 禁止，1 — 允许

### D<sub>1</sub>位 (DTR)、D<sub>5</sub>位 (RTS )

- 设置对应引脚的状态，设置位状态与引脚状态相反

### D<sub>3</sub>位 (SBRK)

- 强迫 TxD 为低电平，**发送连续的空号** (中止符)

### D<sub>7</sub>位 (EH)

- 同步工作模式下，允许 8251A 搜索同步字符

## 03 | 串行接口-命令字应用举例1

### 示例1：使 8251A 内部复位

| EH | IR | RTS | ER | SBRK | RxE | DTR | TxEN |
|----|----|-----|----|------|-----|-----|------|
| 0  | 1  | 0   | 0  | 0    | 0   | 0   | 0    |

- 初始化程序段

```
MOV DX, 309H ; 8251命令端口地址
MOV AL, 40H
OUT DX, AL
```

- 8251A 初始化时，一定要在**写方式控制字之前**先使用复位命令对 8251A 进行内部的复位操作

## 03 | 串行接口-命令字应用举例2

### 示例2：使 8251A 同时允许接收和发送

| EH | IR | RTS | ER | SBRK | RxE | DTR | TxEN |
|----|----|-----|----|------|-----|-----|------|
| 0  | 0  | 0   | 0  | 0    | 1   | 0   | 1    |

- 初始化程序段

```
MOV DX, 309H ; 8251命令端口地址
MOV AL, 05H
OUT DX, AL
```



# 03 | 串行接口-状态字格式

## 状态控制字格式

|                |        |              |               |              |      |       |       |
|----------------|--------|--------------|---------------|--------------|------|-------|-------|
| Data Set Ready |        | 错误标志位        |               |              |      |       |       |
|                |        | Format Error | Overrun Error | Parity Error |      |       |       |
| DSR            | SYNDET | FE           | OE            | PE           | TxE  | RxRDY | TxRDY |
| 数据装置就绪         | 同步检出   | 格式错          | 超越错           | 奇偶错          | 发送器空 | 接收就绪  | 发送就绪  |

- 状态字是在 8251A 工作期间中自动设置的；对应位置 1，表示该状态有效
- FE、OE、PE 三个标志位的置位**并不会中止 8251A 的工作**，但可以使用命令控制字的 **ER 位**使之复位

# 03 | 串行接口-状态字应用举例1

## 示例1：查询 8251A 接收器是否准备好

| DSR    | SYNDET | FE  | OE  | PE  | TxE  | RxRDY | TxRDY |
|--------|--------|-----|-----|-----|------|-------|-------|
| 数据装置就绪 | 同步检出   | 格式错 | 超越错 | 奇偶错 | 发送器空 | 接收就绪  | 发送就绪  |

### ● 程序段

```
MOV DX, 309H ;状态口
L: IN AL, DX ;读状态字
AND AL, 02H ;查 D1=1? (RxRDY)
JZ L ;未准备好,则等待
MOV DX, 308H ;数据口
IN AL, DX
```

## 03 | 串行接口-状态字应用举例2

### 示例2：检查 8251A 接收的数据是否出错

| DSR    | SYNDET | FE  | OE  | PE  | TxE  | RxRDY | TxRDY |
|--------|--------|-----|-----|-----|------|-------|-------|
| 数据装置就绪 | 同步检出   | 格式错 | 超越错 | 奇偶错 | 发送器空 | 接收就绪  | 发送就绪  |

#### ● 程序段

```
MOV DX, 309H ;状态口
IN AL, DX
TEST AL, 38H ;检查FE, OE, PE三位
JNZ ERR ;若其中有一位为1, 则出错
```

# 03 | 串行接口-子目录

## 8251A 串行接口

- 串行接口的概念
- 8251A 可编程串行接口
  - 8251A 的主要特性
  - 8251A 的外部引脚特性
  - 8251A 的内部结构
- 8251A 的编程及应用
  - 8251A 的命令字与状态字
  - 8251A 的初始化编程
  - 8251A 的应用举例

## 03 | 串行接口-8251A的初始化编程

### 命令字类型

- 8251A 仅有一个控制端口 (309H)，且命令字没有标志位，故 8251A 根据控制字的**写入顺序**区分命令字的类型

### 异步方式下的初始化

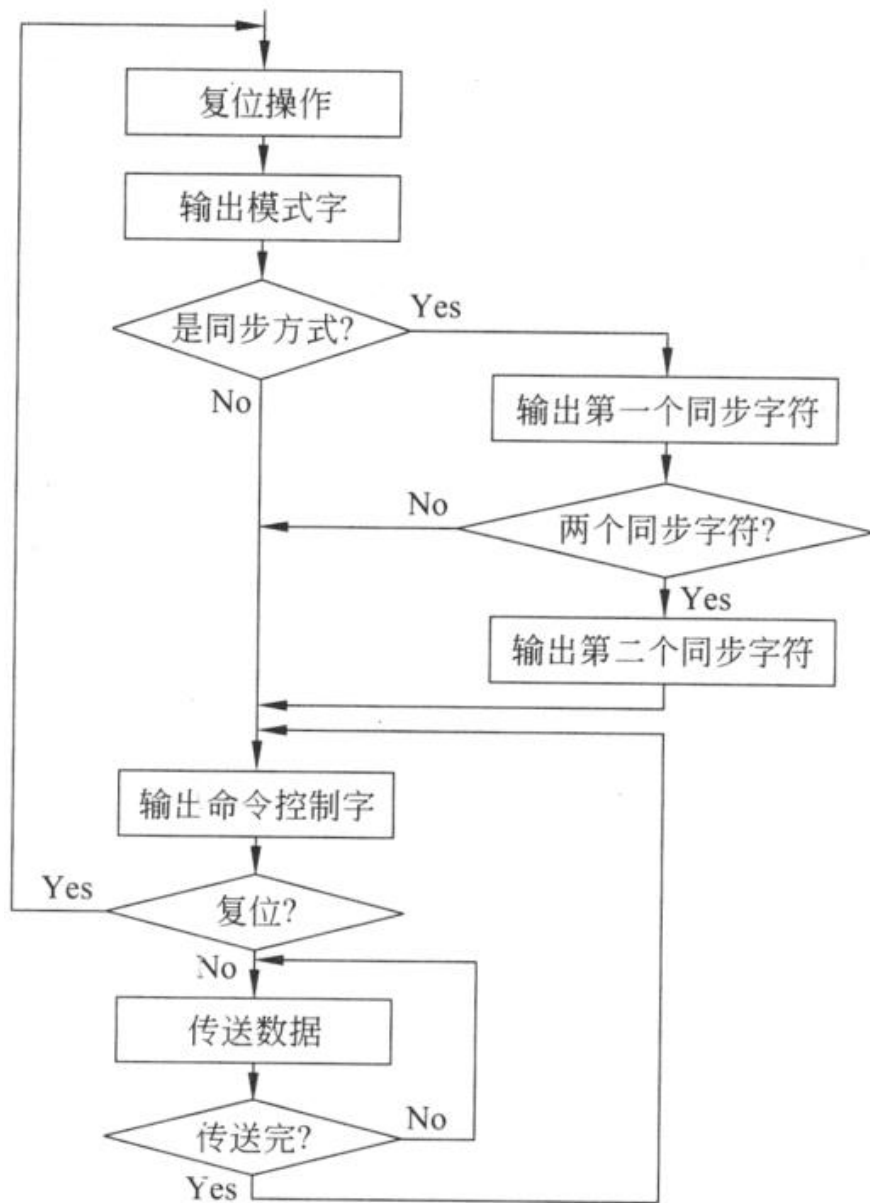
- 复位命令——方式控制字——命令控制字——数据收发

### 同步方式下的初始化

- 复位命令——方式控制字——**同步字符**——命令控制字——数据收发

# 03 | 串行接口-8251A的初始化编程

## 8251A 的初始化流程图



## 03 | 串行接口-异步方式下的初始化举例

示例：假设 8251A 工作于异步方式，波特率因子为 16，每字符 7 个数据位，偶校验，2 位停止位，工作于接收和发送状态，使 RTS# 和 DTR# 有效，试写出初始化程序段

### ● 方式控制字

| S <sub>1</sub> | S <sub>0</sub> | EP | PEN | L <sub>1</sub> | L <sub>0</sub> | B <sub>1</sub> | B <sub>0</sub> |
|----------------|----------------|----|-----|----------------|----------------|----------------|----------------|
| 1              | 1              | 1  | 1   | 1              | 0              | 1              | 0              |

0FAH

### ● 命令控制字

| EH | IR | RTS | ER | SBRK | RxE | DTR | TxEN |
|----|----|-----|----|------|-----|-----|------|
| 0  | 0  | 1   | 1  | 0    | 1   | 1   | 1    |

37H

Data

Enter Mode    Hunt    Internal Reset    Request to Send    Error Reset    Send Break    Receive Enable    Data Terminal Ready    Transmit Enable

## 03 | 串行接口-异步方式下的初始化举例

**示例：假设 8251A 工作于异步方式，波特率因子为 16，每字符 7 个数据位，偶校验，2 位停止位，工作于接收和发送状态，使 RTS# 和 DTR# 有效，试写出初始化程序段**

- 初始化程序段

```
MOV DX, 309H
MOV AL, 40H      ;复位命令
OUT DX, AL
MOV AL, 0FAH     ;方式命令字
OUT DX, AL
MOV AL, 37H      ;命令控制字
OUT DX, AL
```



## 03 | 串行接口-同步方式下的初始化举例

示例：假定 8251A 工作于同步方式，单同步字符 16H，内同步，偶校验，每字符 7 个数据位，试写出初始化程序段

- 方式控制字

| S <sub>1</sub> | S <sub>0</sub> | EP | PEN | L <sub>1</sub> | L <sub>0</sub> | B <sub>1</sub> | B <sub>0</sub> |      |
|----------------|----------------|----|-----|----------------|----------------|----------------|----------------|------|
| 1              | 0              | 1  | 1   | 1              | 0              | 0              | 0              | 0B8H |

- 命令控制字

| EH | IR | RTS | ER | SBRK | RxE | DTR | TxEN |      |
|----|----|-----|----|------|-----|-----|------|------|
| 1  | 0  | 1   | 1  | 0    | 1   | 1   | 1    | 0B7H |

Enter Mode    Hunt    Internal Reset    Request to Send    Error Reset    Send Break    Receive Enable    Data Terminal Ready    Transmit Enable

## 03 | 串行接口-异步方式下的初始化举例

示例：假定 8251A 工作于同步方式，单同步字符 16H，内同步，偶校验，每字符 7 个数据位，试写出初始化程序段

- 初始化程序段

```
MOV DX, 309H
MOV AL, 40H      ;复位命令
OUT DX, AL
MOV AL, 0B8H     ;方式命令字
OUT DX, AL
MOV AL, 16H      ;设置同步字符
OUT DX, AL
MOV AL, 0B7H     ;命令控制字
OUT DX, AL
```

# 03 | 串行接口-子目录

## 8251A 串行接口

- 串行接口的概念
- 8251A 可编程串行接口
  - 8251A 的主要特性
  - 8251A 的外部引脚特性
  - 8251A 的内部结构
- 8251A 的编程及应用
  - 8251A 的命令字与状态字
  - 8251A 的初始化编程
  - 8251A 的应用举例

## 03 | 串行接口-8251A的应用举例

### 8251A的数据传送可采用查询方式或中断方式

- **查询方式：先读取状态信息，再传送数据**
  - 状态寄存器  $D_0=1$  (TxRDY) — 8251A 发送就绪，CPU 可以写数据
  - 状态寄存器  $D_1=1$  (RxRDY) — 8251A 已接收一帧数据并完成串并转换，CPU 可以读入数据
- **中断方式：8251A 发硬件的请求信号给 CPU**
  - 8251A 使用 TxRDY、RxRDY 引脚作为发送、接收中断请求
  - 收发均采用中断方式时，TxRDY、RxRDY 通过或门与 INTR 相连
  - CPU 响应中断时，查询状态寄存器，区分是发送中断还是接收中断

# 03 | 串行接口-8251A的应用举例1

示例1：将 8251A 设置为异步传送方式，波特率因子为 64，采用偶校验，1 位停止位，7 位数据位。8251A 与外设有握手信号，采用查询方式发送数据

## ● 方式控制字

| S <sub>1</sub> | S <sub>0</sub> | EP | PEN | L <sub>1</sub> | L <sub>0</sub> | B <sub>1</sub> | B <sub>0</sub> |
|----------------|----------------|----|-----|----------------|----------------|----------------|----------------|
| 0              | 1              | 1  | 1   | 1              | 0              | 1              | 1              |

07BH

## ● 命令控制字

| EH | IR | RTS | ER | SBRK | RxE | DTR | TxEN |
|----|----|-----|----|------|-----|-----|------|
| 0  | 0  | 1   | 1  | 0    | 0   | 0   | 1    |

31H

Data

Enter Mode    Hunt    Internal Reset    Request to Send    Error Reset    Send Break    Receive Enable    Data Terminal Ready    Transmit Enable

## 03 | 串行接口-8251A的应用举例1

示例1：将 8251A 设置为异步传送方式，波特率因子为 64，采用偶校验，1 位停止位，7 位数据位。8251A 与外设有握手信号，采用查询方式发送数据

### ● 控制程序段

```
MOV DX, 309H ;控制口地址
MOV AL, 40H ;复位命令
OUT DX, AL
MOV AL, 7BH ;方式控制字
OUT DX, AL
MOV AL, 31H ;命令控制字
OUT DX, AL
```

```
WAIT0: IN AL, DX
      AND AL, 01H ;检查TxRDY=1?
      JZ WAIT0
      MOV DX, 308H
      MOV AL, ** ;输出数据
      OUT DX, AL
      JMP WAIT0
```

| DSR    | SYNDET | FE  | OE  | PE  | TxE  | RxRDY | TxRDY |
|--------|--------|-----|-----|-----|------|-------|-------|
| 数据装置就绪 | 同步检出   | 格式错 | 超越错 | 奇偶错 | 发送器空 | 接收就绪  | 发送就绪  |

## 03 | 串行接口-8251A的应用举例2

### 示例2：试为两台微机设计串行通信接口

- 要求：
  1. 甲机发送，乙机接收，均采用**查询方式**交换数据
  2. 采用**异步**工作方式，波特率因子为 64，8 位数据位，1 位停止位，偶校验
  3. 双方的 8251A **控制端口**为 20AH，**数据端口**为 208H
  4. 甲乙两机发送数据区为 SOUR，接收数据区为 DEST，数据块长度为 COUNT

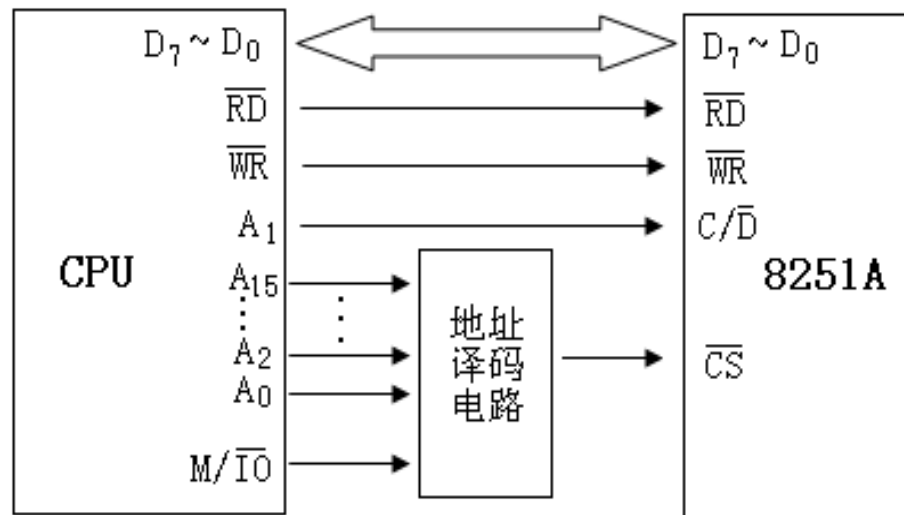
## 03 | 串行接口-8251A的应用举例2

### 8251A 的端口问题

- CPU 的  $A_1$  引脚接 8251A 的  $C/\overline{D}\#$  引脚

- $A_1=0$  时, 选择 8251A 的 **数据端口**

- $A_1=1$  时, 选择 8251A 的 **命令端口**

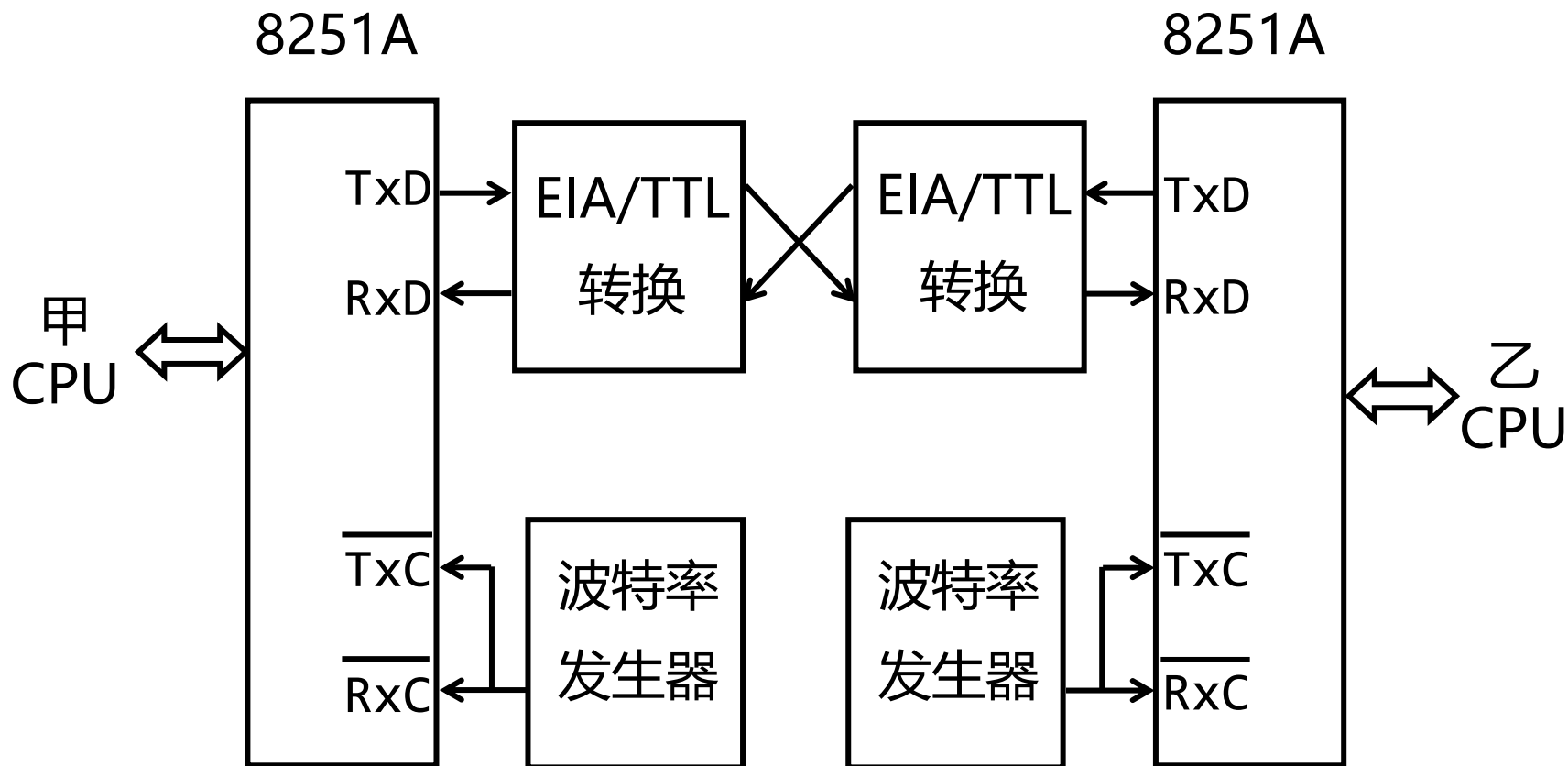


- 若 CPU 发出地址 **10 0000 1000** (**208H**) 时, 选中 8251A 的数据端口, 则其命令端口地址为 **10 0000 1010** (**20AH**)



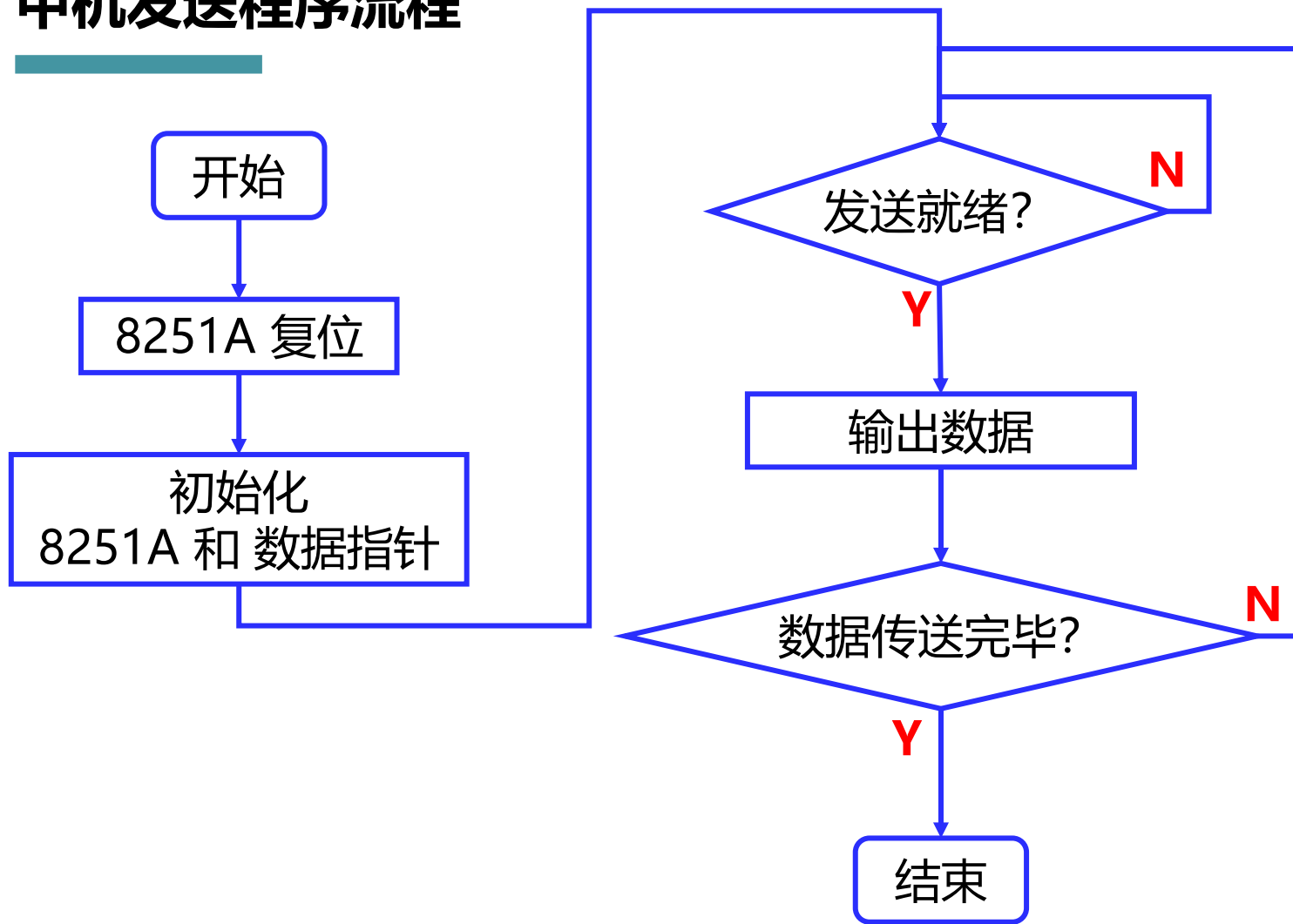
## 03 | 串行接口-8251A的应用举例2

### 甲乙双机通信串行连接图



# 03 | 串行接口-8251A的应用举例2

## 甲机发送程序流程



# 03 | 串行接口-8251A的应用举例2

## 甲机 8251A 命令字

采用**异步**工作方式，波特率因子为 64，8 位数据位，1 位停止位，偶校验

- 方式控制字

| S <sub>1</sub> | S <sub>0</sub> | EP | PEN | L <sub>1</sub> | L <sub>0</sub> | B <sub>1</sub> | B <sub>0</sub> |      |
|----------------|----------------|----|-----|----------------|----------------|----------------|----------------|------|
| 0              | 1              | 1  | 1   | 1              | 1              | 1              | 1              | 07FH |

- 命令控制字

| EH | IR | RTS | ER | SBRK | RxE | DTR | TxEN |      |
|----|----|-----|----|------|-----|-----|------|------|
| 0  | 0  | 1   | 1  | 0    | 0   | 0   | 1    | 031H |

Enter Mode

Hunt

Internal Reset

Request to Send

Error Reset

Send Break

Receive Enable

Data Terminal Ready

Transmit Enable

# 03 | 串行接口-8251A的应用举例2

## 甲机发送程序

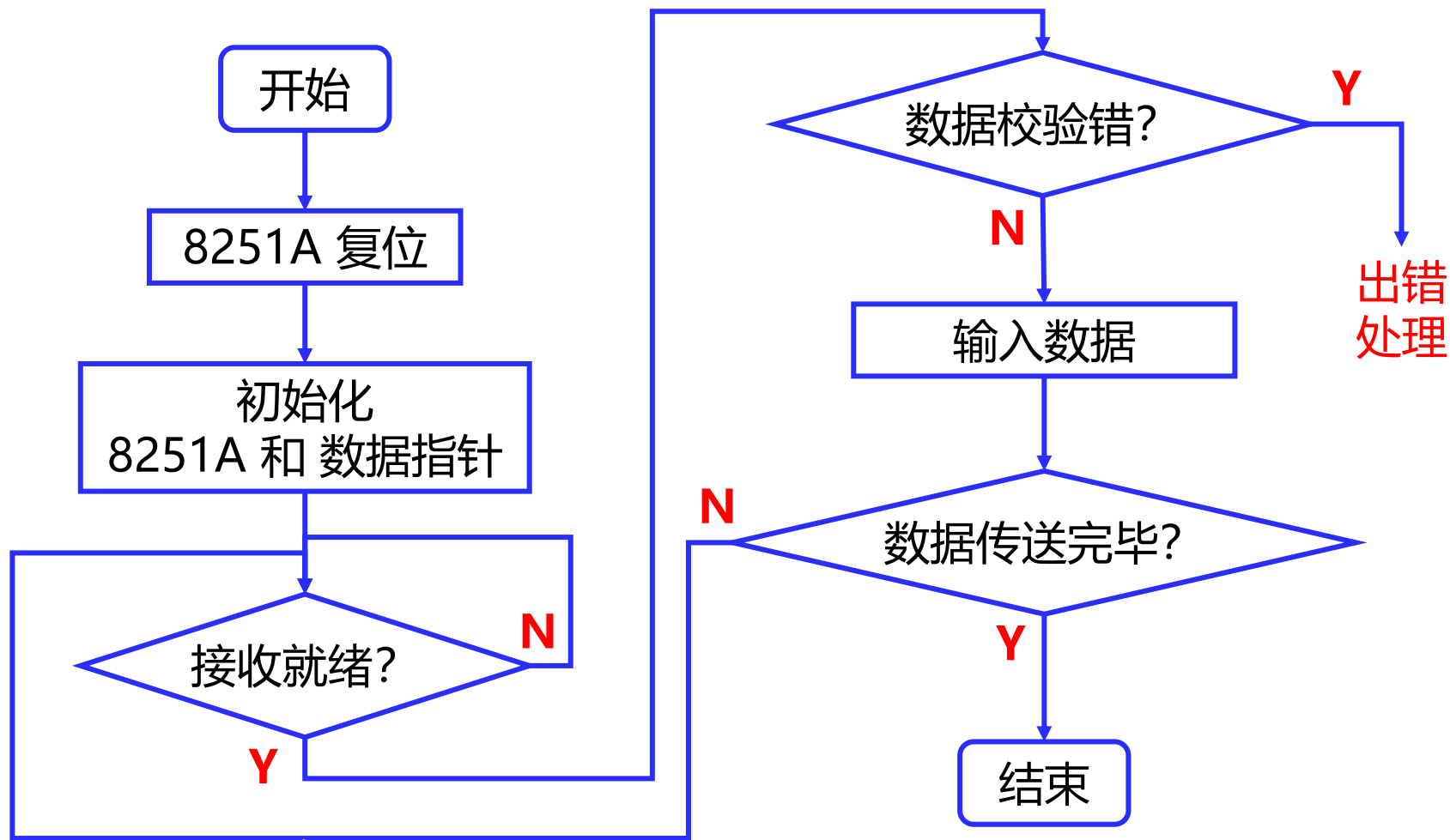
```
MOV DX, 20AH ;控制端口
MOV AL, 40H ;复位命令
OUT DX, AL
MOV AL, 7FH ;方式控制字
OUT DX, AL
MOV AL, 31H ;命令控制字
OUT DX, AL
MOV DI, offset SOUR
MOV CX, COUNT
```

```
NEXT: MOV DX, 20AH
      IN AL, DX
      AND AL, 01H ;发送就绪状态查询
      JZ NEXT
      MOV DX, 208H ;数据端口
      MOV AL, [DI]
      OUT DX, AL
      INC DI
      LOOP NEXT
```

| DSR        | SYNDET | FE  | OE  | PE  | TxE  | RxRDY | TxRDY |
|------------|--------|-----|-----|-----|------|-------|-------|
| 数据装<br>置就绪 | 同步检出   | 格式错 | 超越错 | 奇偶错 | 发送器空 | 接收就绪  | 发送就绪  |

# 03 | 串行接口-8251A的应用举例2

## 乙机接收程序流程



# 03 | 串行接口-8251A的应用举例2

## 乙机 8251A 命令字

采用异步工作方式，波特率因子为 64，8 位数据位，1 位停止位，偶校验

- 方式控制字

| S <sub>1</sub> | S <sub>0</sub> | EP | PEN | L <sub>1</sub> | L <sub>0</sub> | B <sub>1</sub> | B <sub>0</sub> |      |
|----------------|----------------|----|-----|----------------|----------------|----------------|----------------|------|
| 0              | 1              | 1  | 1   | 1              | 1              | 1              | 1              | 07FH |

- 命令控制字

| EH | IR | RTS | ER | SBRK | RxE | DTR | TxEN |      |
|----|----|-----|----|------|-----|-----|------|------|
| 0  | 0  | 0   | 1  | 0    | 1   | 1   | 0    | 016H |

Enter Mode

Hunt

Internal Reset

Request to Send

Error Reset

Send Break

Receive Enable

Data Terminal Ready

Transmit Enable

# 03 | 串行接口-8251A的应用举例2

## 乙机接收程序

```
MOV DX, 20AH ;控制端口
MOV AL, 40H ;复位命令
OUT DX, AL
MOV AL, 7FH ;方式控制字
OUT DX, AL
MOV AL, 16H ;命令控制字
OUT DX, AL
MOV DI, offset DEST
MOV CX, COUNT
```

```
NEXT: MOV DX, 20AH
      IN AL, DX
      AND AL, 02H ;接收就绪状态查询
      JZ NEXT
      TEST AL, 38H ;错误状态查询
      JNZ ERR
      MOV DX, 208H ;数据端口
      IN AL, DX
      MOV [DI], AL
      INC DI
      LOOP NEXT
```

| DSR        | SYNDET | FE  | OE  | PE  | TxE  | RxRDY | TxRDY |
|------------|--------|-----|-----|-----|------|-------|-------|
| 数据装<br>置就绪 | 同步检出   | 格式错 | 超越错 | 奇偶错 | 发送器空 | 接收就绪  | 发送就绪  |

# 本章小结

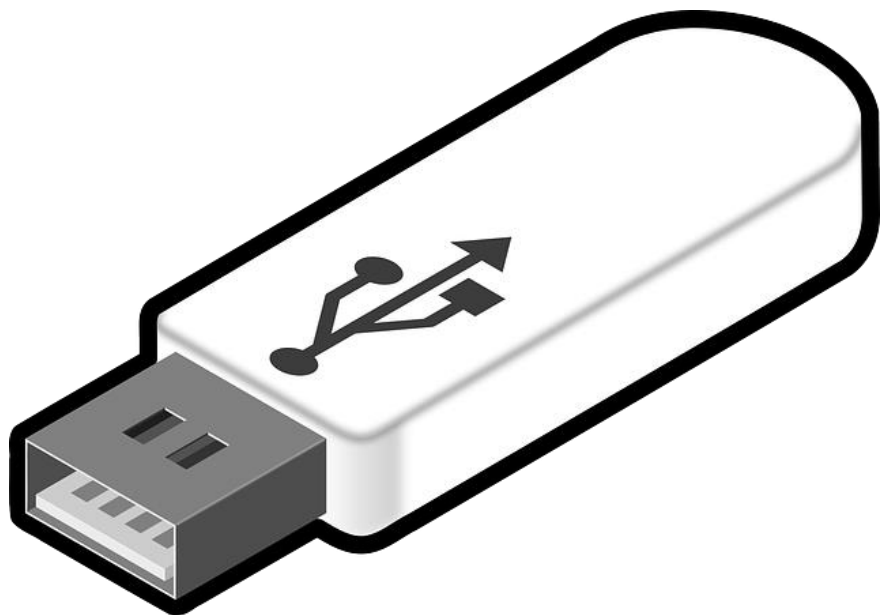
---

- 了解并行、串行通信的基本概念及特点
- 熟悉可编程接口芯片的外部引脚，清楚各类的引脚的连接对象
- **掌握**可编程接口芯片的关键内部结构
- **掌握**可编程接口芯片的命令字和状态字，能够完成芯片的初始化编程
- **理解**可编程接口芯片的各种应用
  - 理解 8255A 在方式 0、方式 1 下的查询传送方式区别
  - 理解 8251A 在查询传送方式下的发送与接收





河南大學  
Henan University



## Q&A

---

主讲教师：舒高峰

电子邮箱：[gaofeng.shu@henu.edu.cn](mailto:gaofeng.shu@henu.edu.cn)

联系电话：13161693313