



计算机组织与体系结构

Computer Architectures

陆俊林



第九讲 输入输出接口

本讲要点

首先介绍输入输出电路的基本原理，其次讲解输入和输出数据的基本过程，然后比较并行接口和串行接口，最后结合芯片实例和应用讲解接口的编程。

阅读教材《微型计算机……》：10



主要内容

通过学习本课程
了解计算机的发展历程，理解计算机的组成原理，掌握计算机的设计方法



I 输入输出接口电路

II 输入和输出的过程

III 串行和并行的比较

IV 并口芯片的应用实例

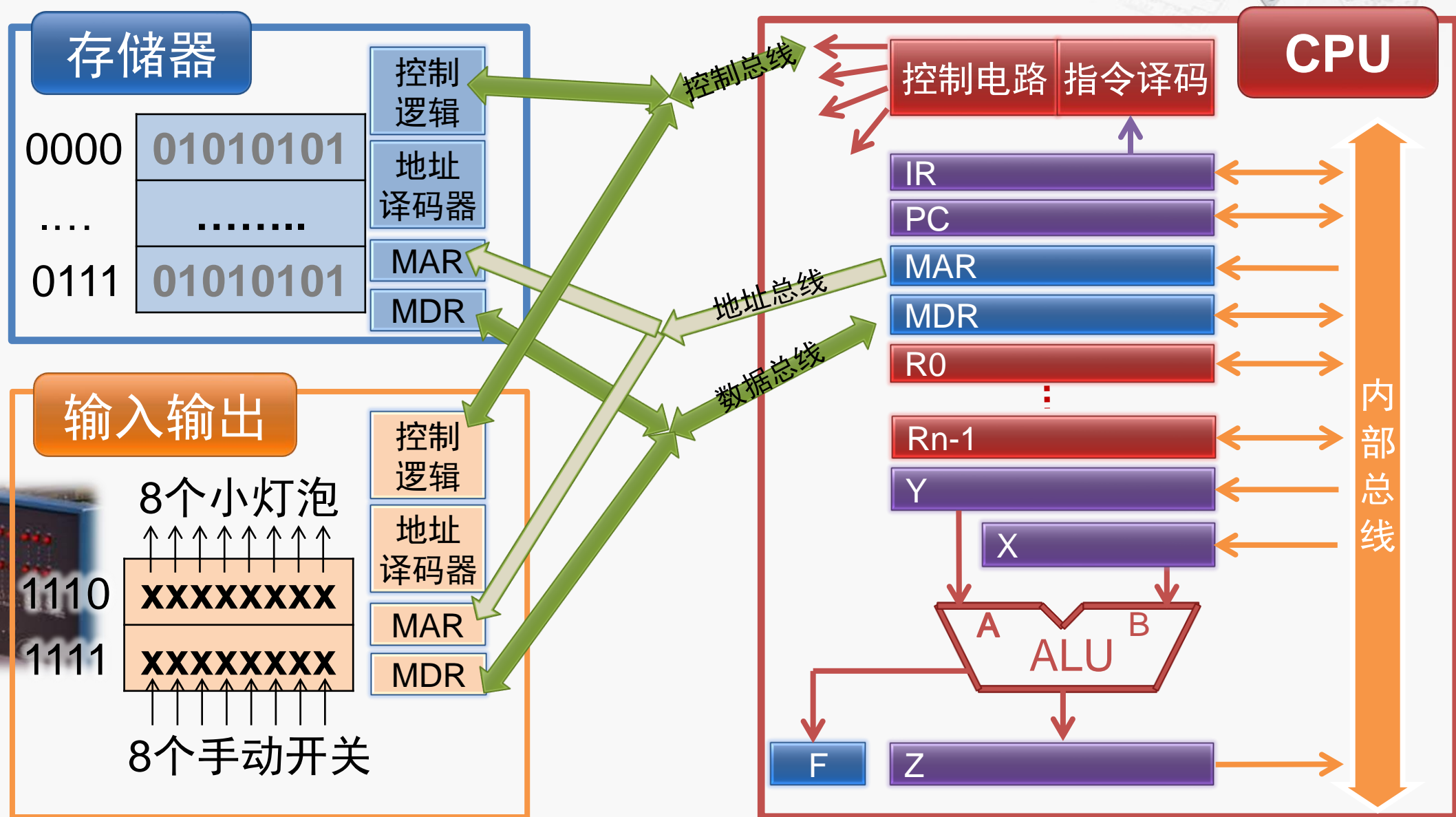


第一台微型计算机：Altair 8800

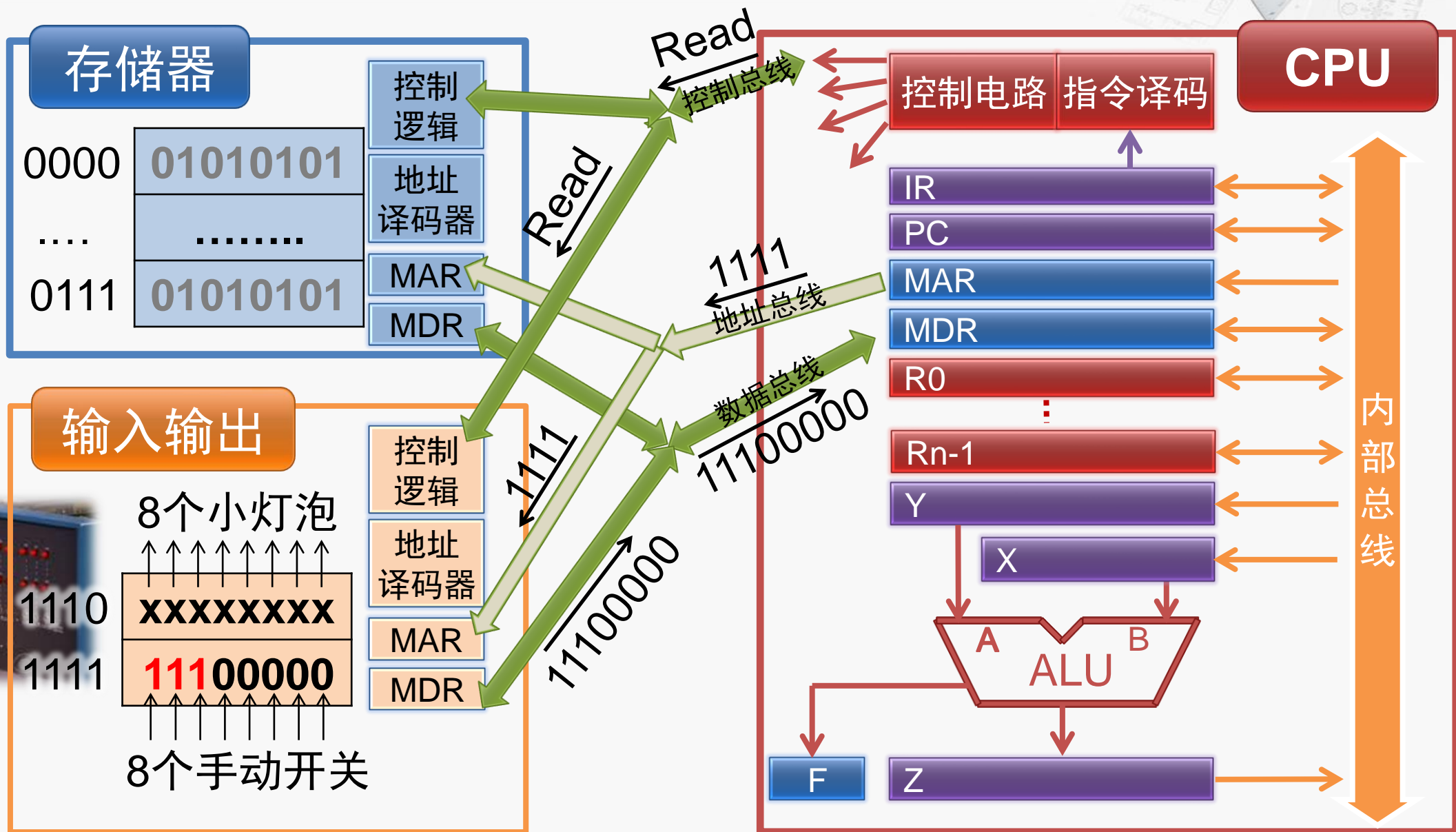
- 内部主要有两块集成电路：
 - Intel 8080微处理器，256字节存储器
- 面板上提供简单的输入输出
 - 输入：手动开关
 - 输出：小灯泡



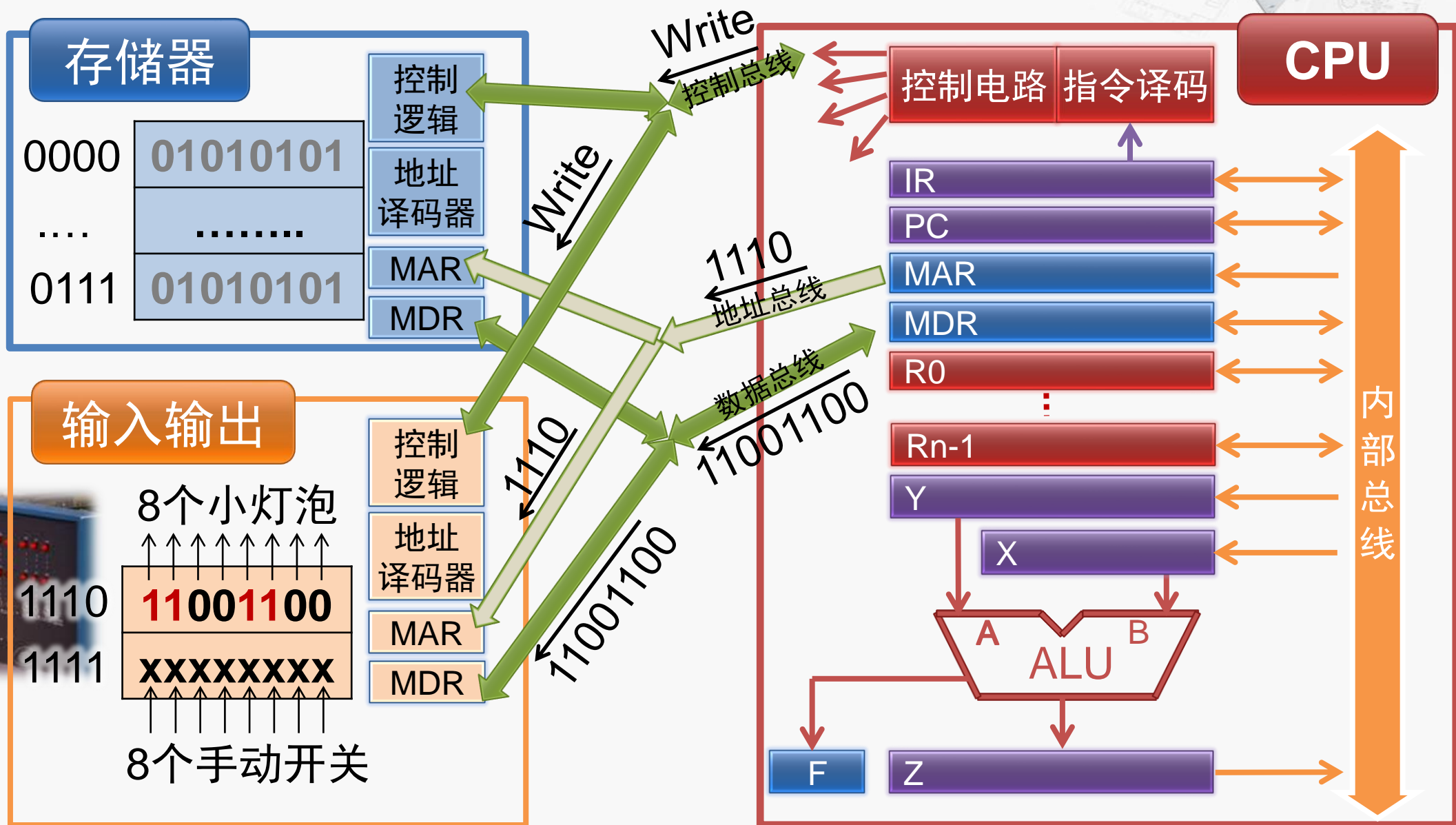
带有简单输入输出设备的模型机



“输入”的简单场景



“输出”的简单场景

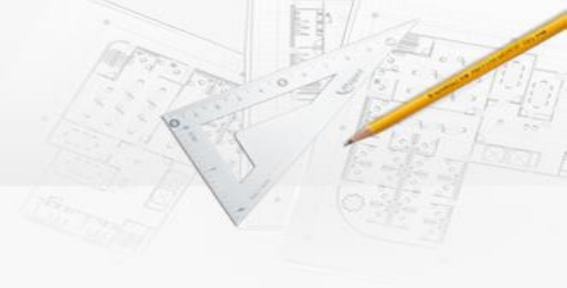


I/O接口的产生原因



1. CPU和外设之间的速度差距
 2. 外设处理的信息格式和接口信号形式多样
 - 串行、并行
 - 数字、模拟
 - 标准逻辑电平、非标准逻辑电平
- 🎯 现代微型计算机中设置了I/O接口电路，作为CPU和外设之间传送数据的转接站

I/O接口的基本功能



1. 数据缓冲

- 解决CPU和外设之间的速度差距

2. 提供联络信息

- 协调与同步数据交换过程

3. 信号与信息格式的转换

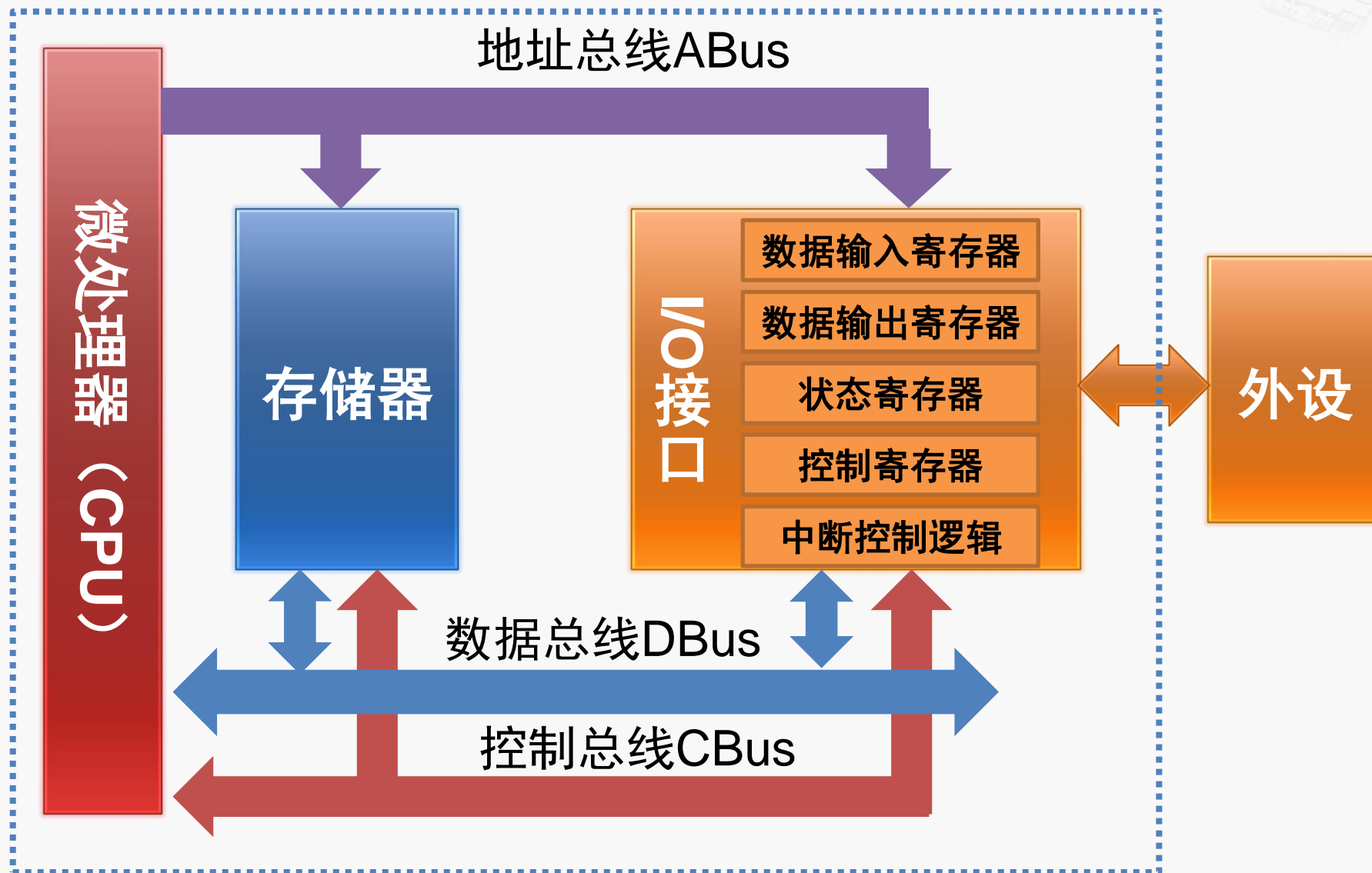
- 模/数、数/模转换，串/并、并/串转换，电平转换

4. 设备选择

5. 中断管理

6. 可编程功能

输入输出接口（I/O接口）的基本结构



主要内容

通过学习本课程
了解计算机的发展历程，理解计算机的组成原理，掌握计算机的设计方法

I 输入输出接口电路



II 输入和输出的过程

III 串行和并行的比较

IV 并口芯片的应用实例



I/O端口及其编址方式

I/O端口

- I/O接口内部包含一组称为I/O端口的寄存器
- 每个I/O端口都需有自己的端口地址（或称端口号），以便CPU访问

I/O端口的编址方式

- 在计算机系统中，如何编排I/O接口的端口地址？



常见的I/O端口编址方式



④ I/O端口和存储器分开编址

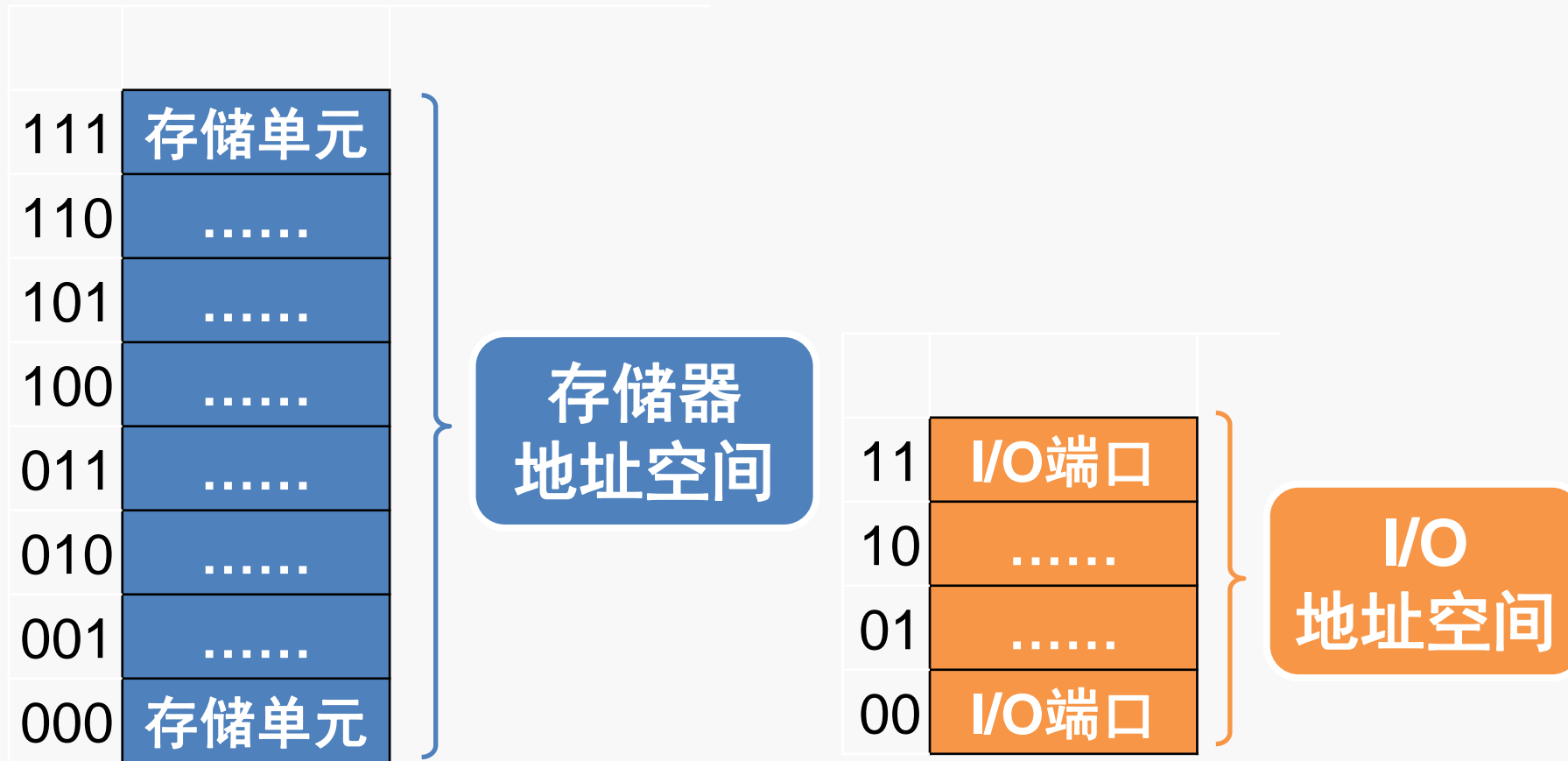
- I/O映像的I/O方式，I/O Mapped I/O
- x86体系结构采用该方式

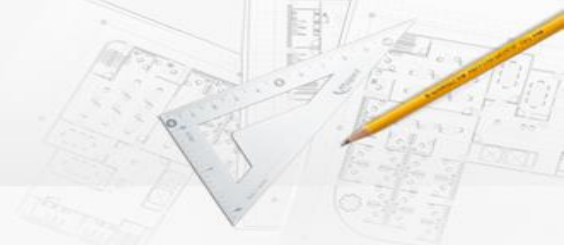
④ I/O端口和存储器统一编址

- 存储器映像的I/O方式，Memory Mapped I/O
- ARM、MIPS、PowerPC等体系结构采用该方式

I/O端口和存储器分开编址

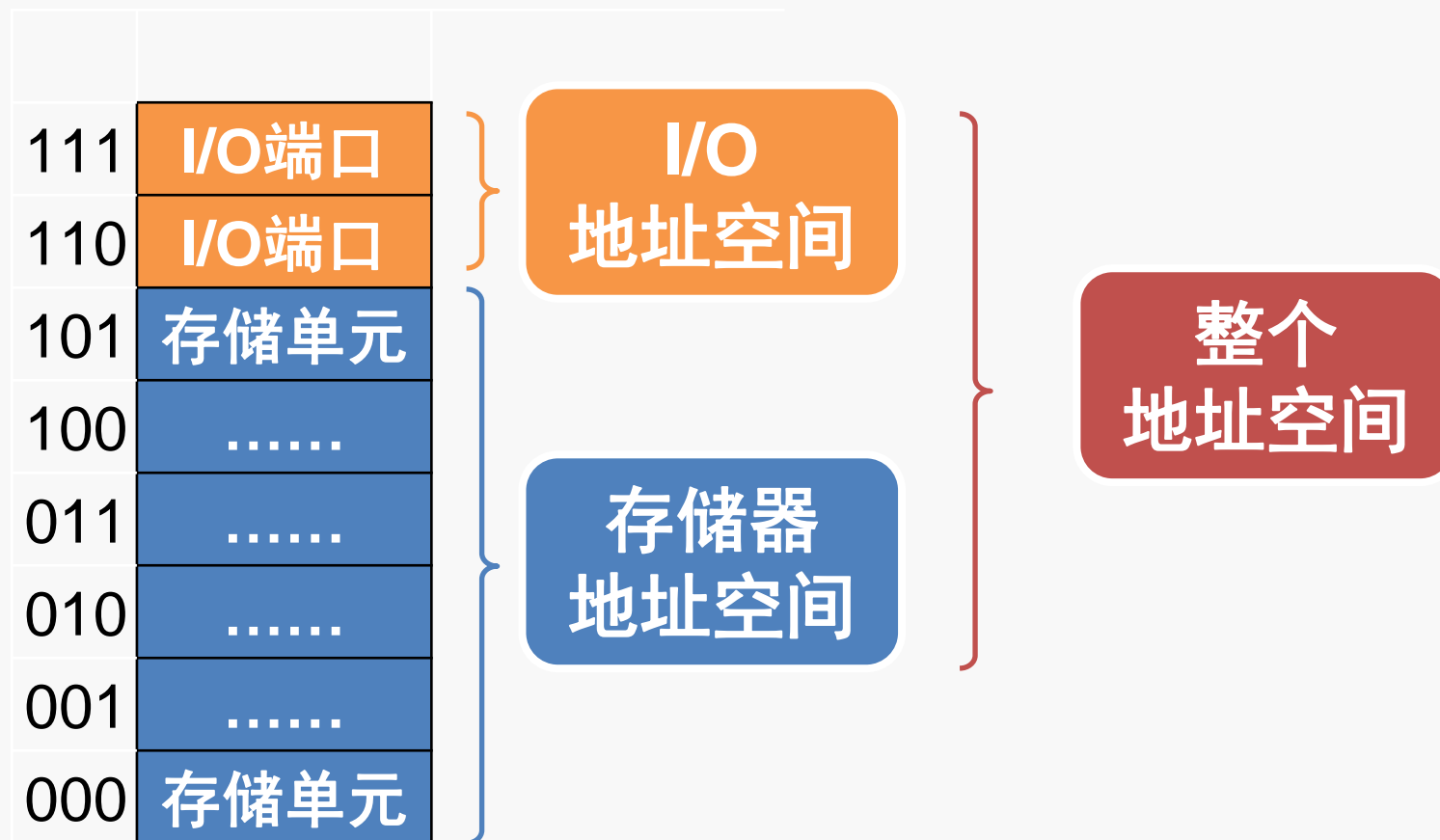
一个分开编址的地址空间划分示例如下





I/O端口和存储器统一编址

- ④ 假设地址宽度为3，一个统一编址的地址空间划分示例如下



统一编址的特点



④ 优点

- 可以用访问存储器的指令来访问I/O端口，访问存储器的指令功能比较齐全，可以实现直接对I/O端口内的数据进行处理
- 可以将CPU中的I/O操作与访问存储器操作统一设计为一套控制逻辑，简化内部结构，同时减少CPU的引脚数目

④ 缺点

- 由于I/O端口占用了一部分存储器地址空间，因而使用户的存储地址空间相对减小
- 由于利用访问存储器的指令来进行I/O操作，指令的长度通常比单独I/O指令要长，因而指令的执行时间也较长

分开编址的特点



④ 优点

- I/O端口不占用存储器地址，不会减少用户的存储器地址空间
- I/O指令编码短，执行速度快
- I/O指令的地址码较短，地址译码方便
- 采用单独的I/O指令，使程序中I/O操作和其他操作层次清晰，便于理解

④ 缺点

-

I/O指令说明

IN指令（输入）

- 格式：IN AC, PORT
- 操作：把外设端口的内容输入到AL或AX

OUT指令（输出）

- 格式：OUT PORT, AC
- 操作：把AL或AX的内容输出到外设端口





IN/OUT指令的寻址方式和示例

❶ 端口地址为0~255

- 直接寻址：用一个字节立即数指定端口地址
- 间接寻址：用DX的内容指定端口地址

❷ 端口地址大于255

- 间接寻址：用DX的内容指定端口地址

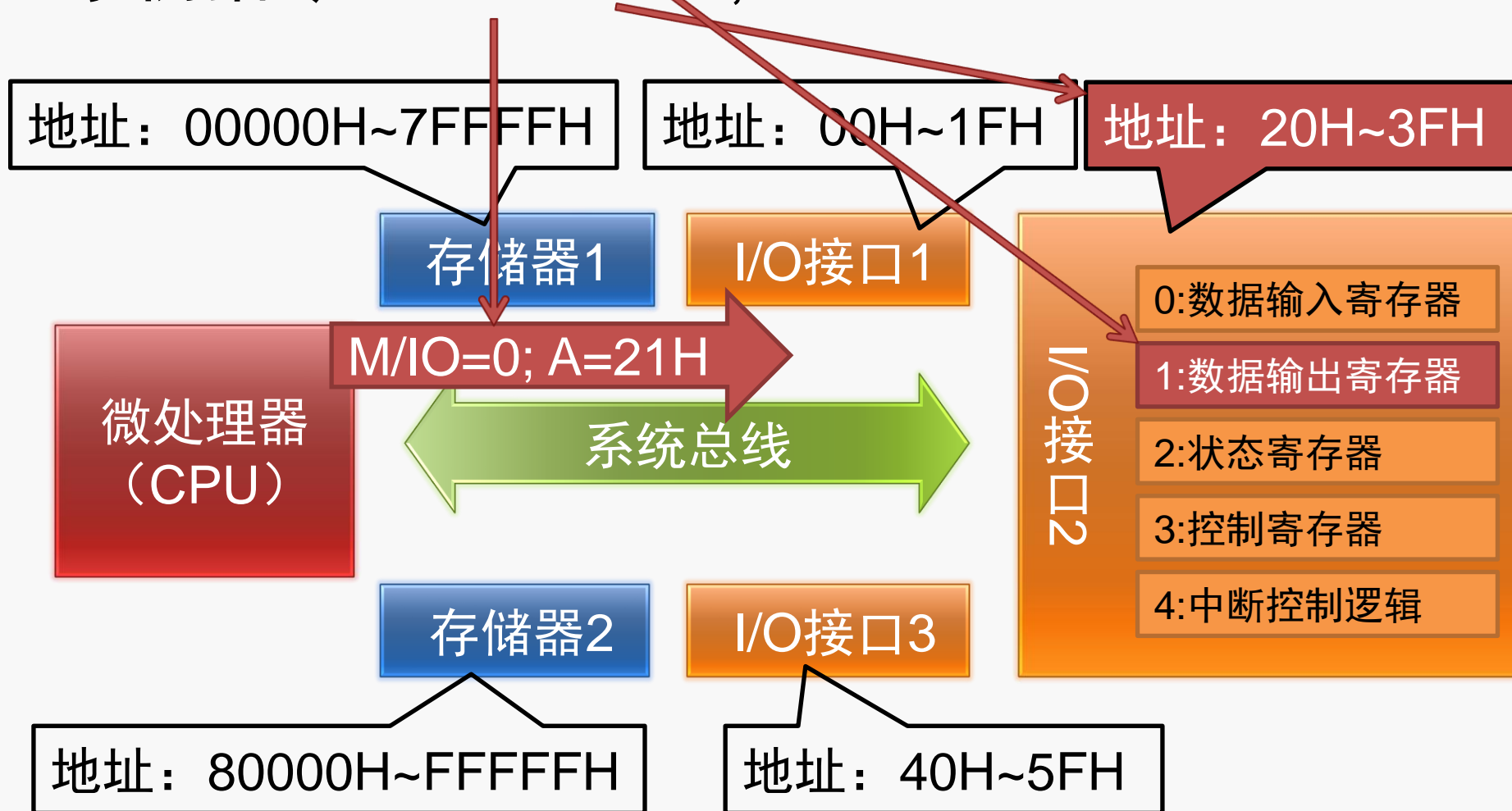
	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
IN：直接寻址	1 1 1 0 0 1 0 w	port
IN：间接寻址	1 1 1 0 1 1 0 w	
OUT：直接寻址	1 1 1 0 0 1 1 w	port
OUT：间接寻址	1 1 1 0 1 1 1 w	

```
IN    AL, 80H
IN    AX, 80H
OUT   80H, AL
OUT   80H, AX
```

```
MOV   DX, 288
IN     AL, DX
IN     AX, DX
OUT    DX, AL
OUT    DX, AX
```

I/O指令的地址译码过程示例

示例指令：OUT 21H, AL



I/O接口示例：并行接口电路

并行接口电路是I/O接口的一种，一般有两种常见的物理实现形式：

- 1、独立的芯片，如Intel 8255A芯片；
- 2、包含在多功能的芯片中，如SuperIO芯片提供了并口、串口、键盘鼠标接口、风扇控制接口等



并行接口电路

0: 数据输入寄存器

1: 数据输出寄存器

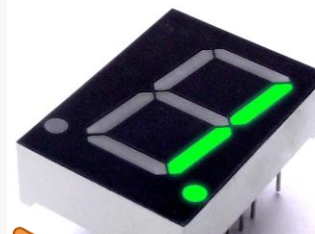
2: 状态寄存器

3: 控制寄存器

4: 中断控制逻辑



打印机

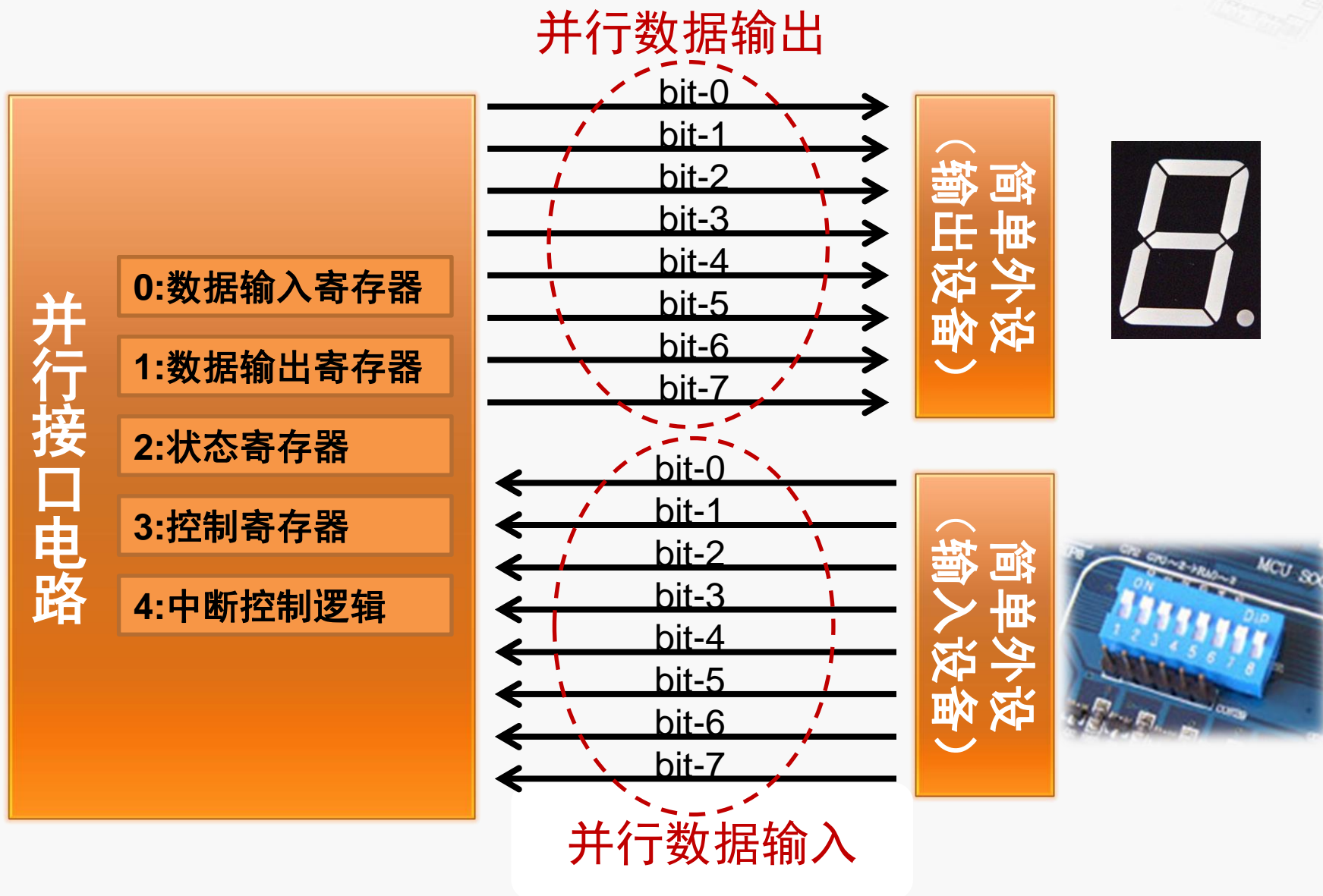


LED数码管

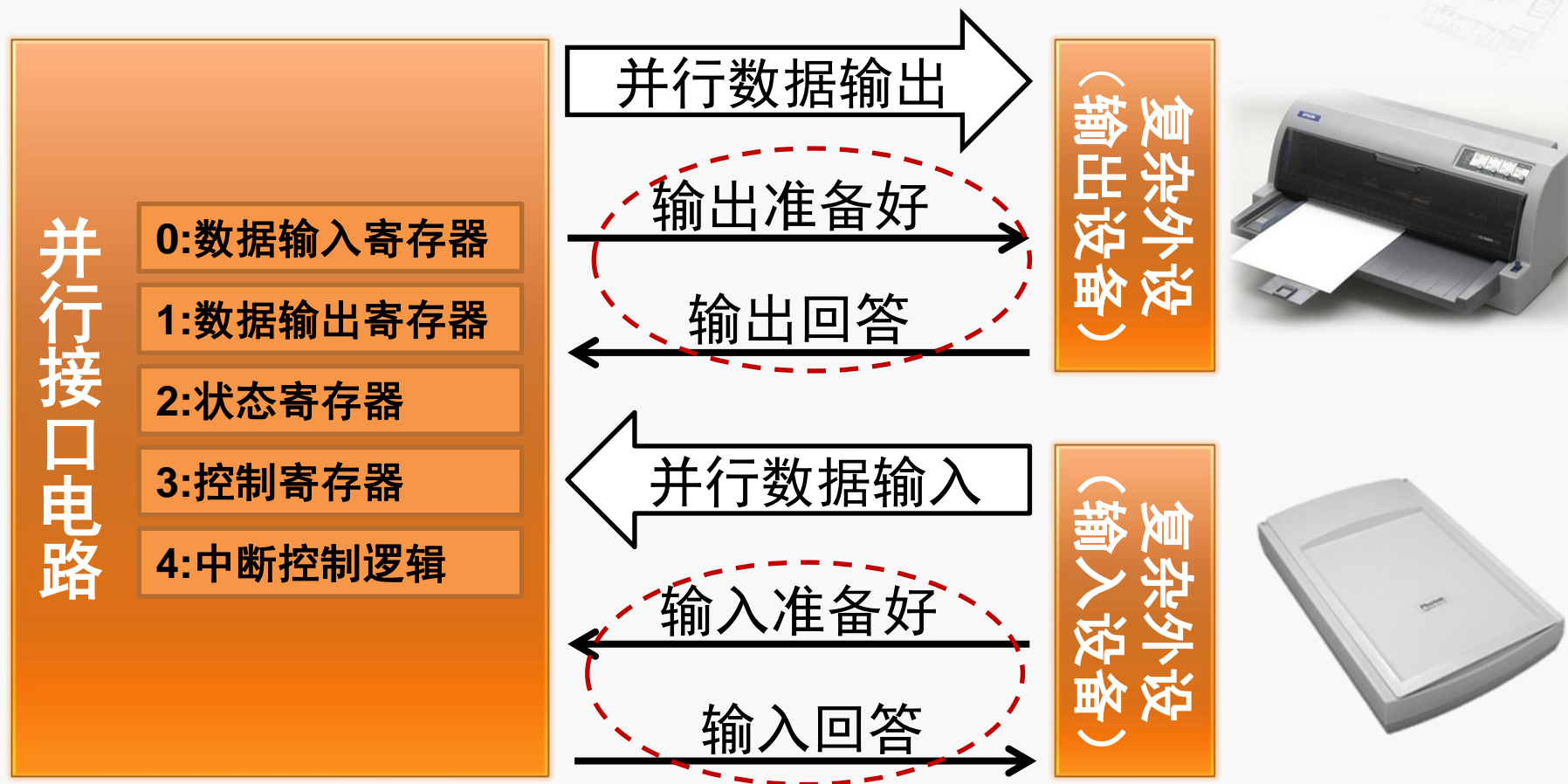


拨码开关

与简单外设的连接信号



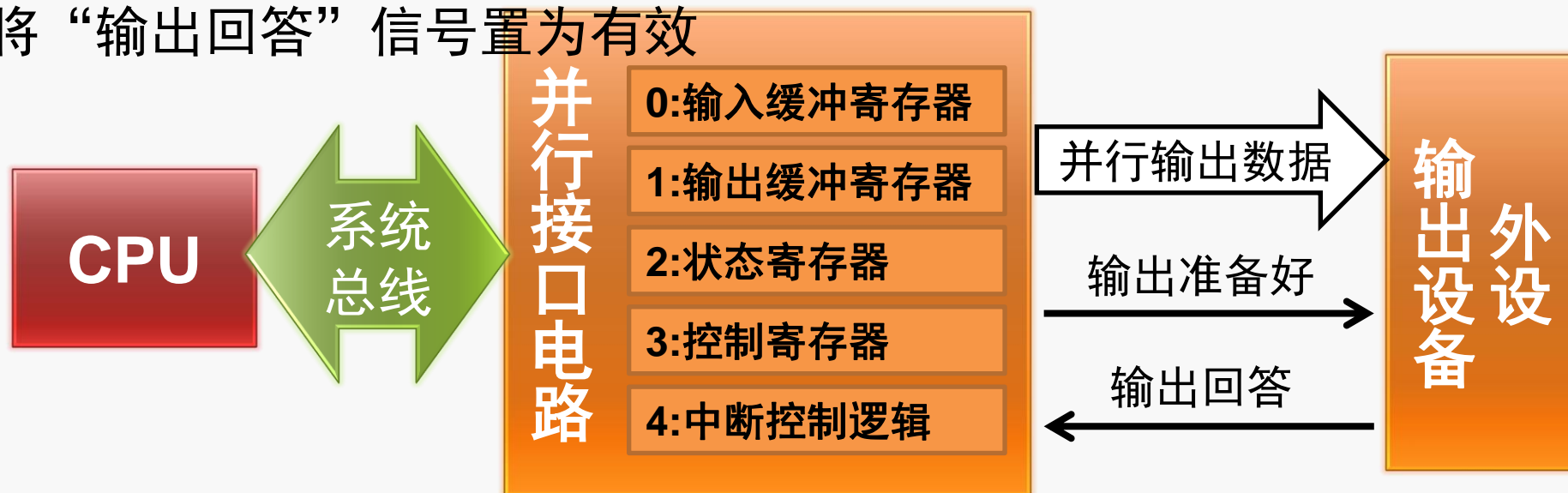
与复杂外设的连接信号



“握手（Handshaking）信号”：总是成对出现，在数据传送中起着定时协调与联络作用。采用“握手”方式的数据传送，每一过程必须都有应答，彼此进行确认。

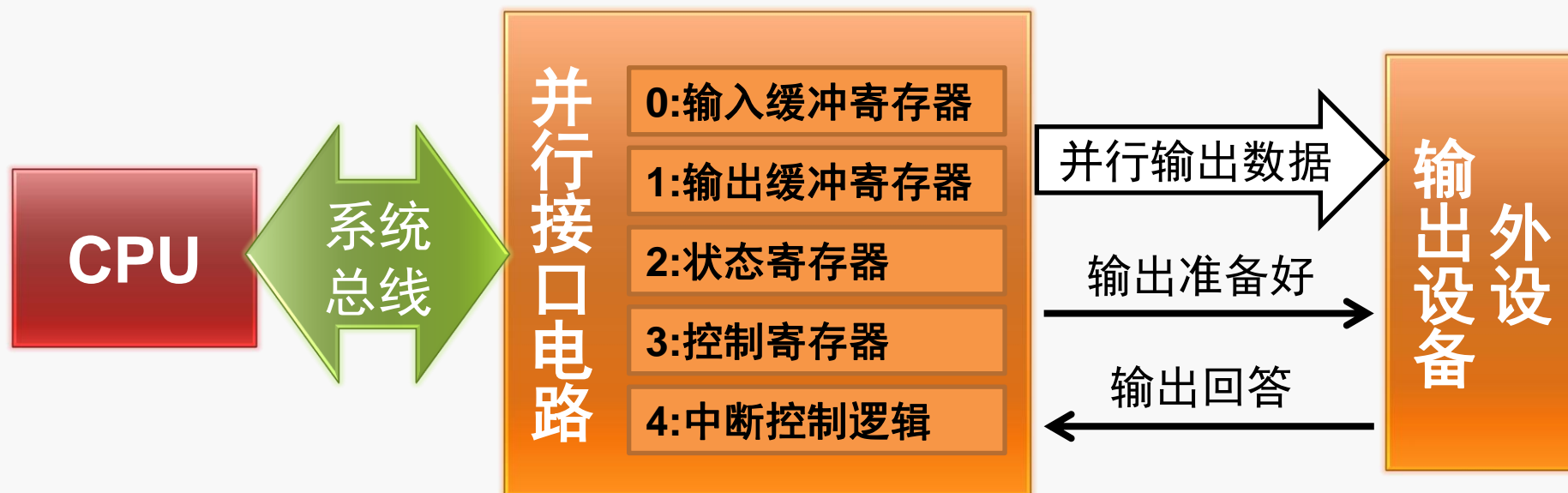
数据输出过程

- ① CPU执行OUT指令，将控制字写入接口的“控制寄存器”，从而设置接口的工作模式
- ② CPU执行OUT指令，将数据写到接口的“输出缓冲寄存器”
- ③ 接口将数据发到“并行数据输出”信号，并将“输出准备好”信号置为有效（亦可由CPU写控制字将该信号置为有效）
- ④ 外设发现“输出准备好”信号有效后，从“并行数据输出”信号接收数据，并将“输出回答”信号置为有效



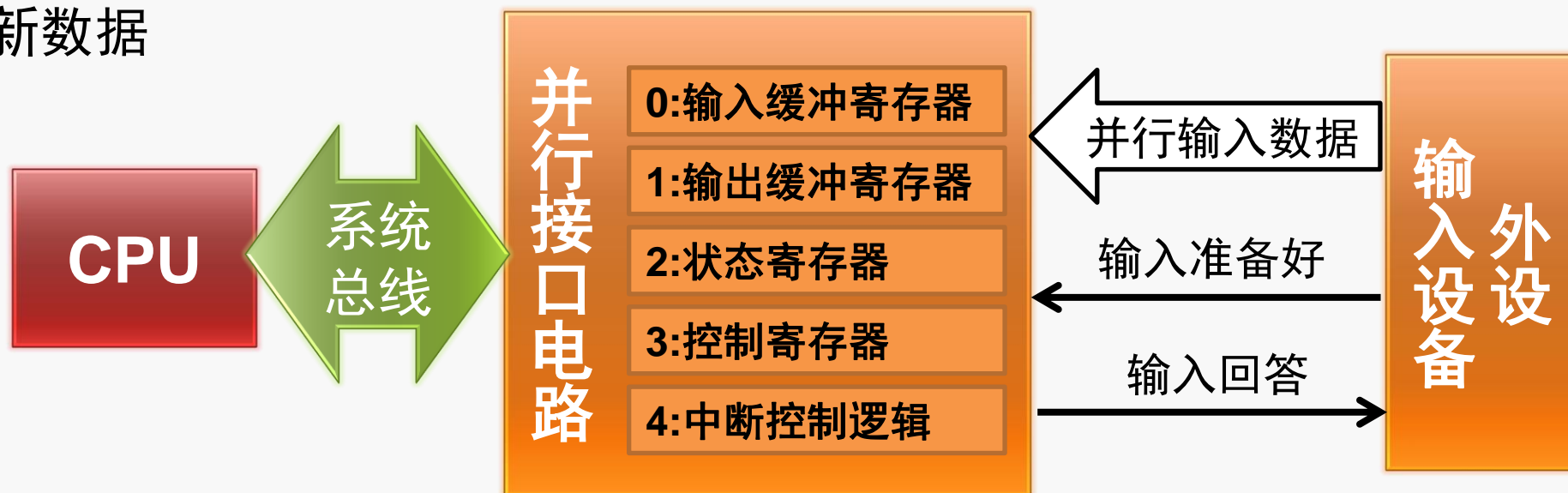
数据输出过程（续）

- ⑤ 接口发现“输出回答”信号有效后，将“状态寄存器”中的状态位“输出缓冲器空”置为有效
- ⑥ 在这个过程中，CPU反复执行IN指令从“状态寄存器”中读出状态字，直到发现“输出缓冲器空”，然后开始下一个输出过程，继续输出新数据
(注：此为程序控制方式，还可以采用中断控制方式)



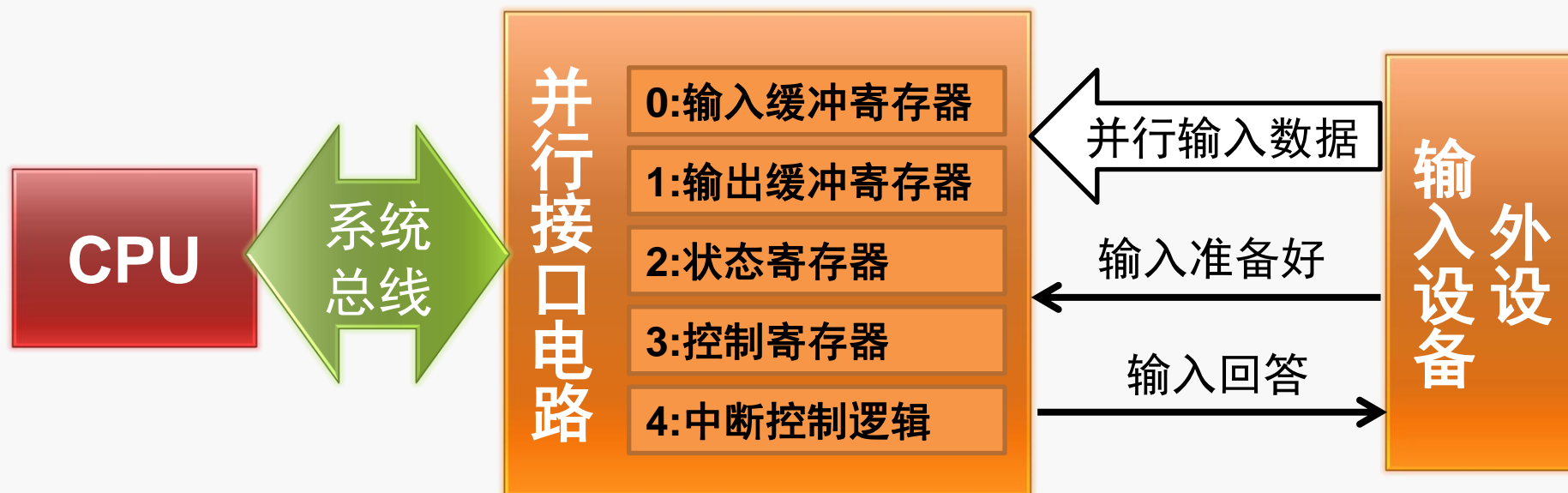
数据输入过程

- ① 系统初始化时，CPU执行OUT指令，将控制字写入接口的“控制寄存器”，设置接口的工作模式
- ② 外设将数据发到“并行数据输入”信号，并将“输入准备好”信号置为有效
- ③ 接口发现“输入准备好”信号有效后，从“并行数据输入”信号接收数据，放入“输入缓冲寄存器”，并将“输入回答”信号置为有效，阻止外设输入新数据



数据输入过程（续）

- ④ 接口将“状态寄存器”中的状态位“输入缓冲器满”置为有效
 - ⑤ 在上述过程中，CPU反复执行IN指令从“状态寄存器”中读出状态字，直到发现“输入缓冲器满”，然后执行IN指令，从“输入缓冲寄存器”中读出数据
 - ⑥ 接口将“输入回答”信号置为无效，等待外设输入新数据
- （注：此为程序控制方式，还可以采用中断控制方式）



主要内容

通过学习本课程
了解计算机的发展历程，理解计算机的组成原理，掌握计算机的设计方法

I 输入输出接口电路

II 输入和输出的过程

III 串行和并行的比较

IV 并口芯片的应用实例



串/并行通信的主要特点



计算机和数据通信系统中的基本数据传送方式

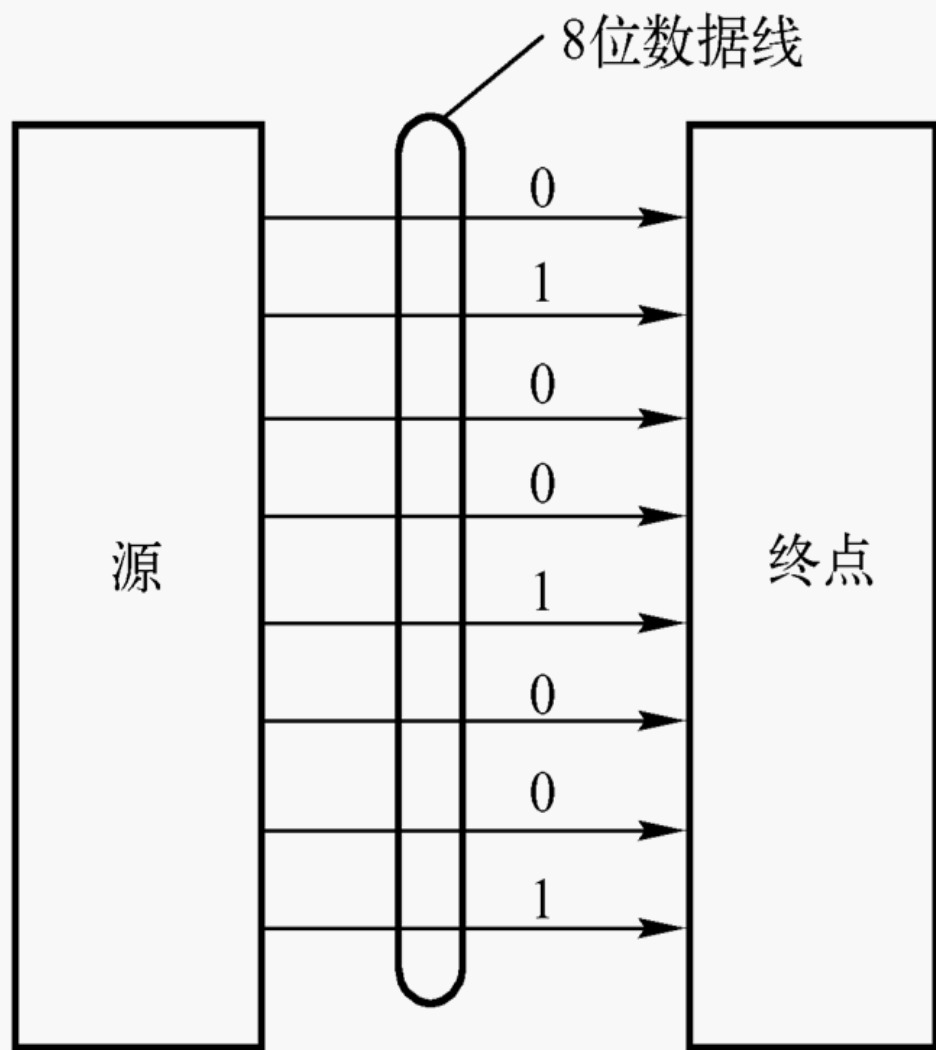
串行通信（串行数据传送）

- 数据在单条一位宽的传输线上按时间先后一位一位地进行传送

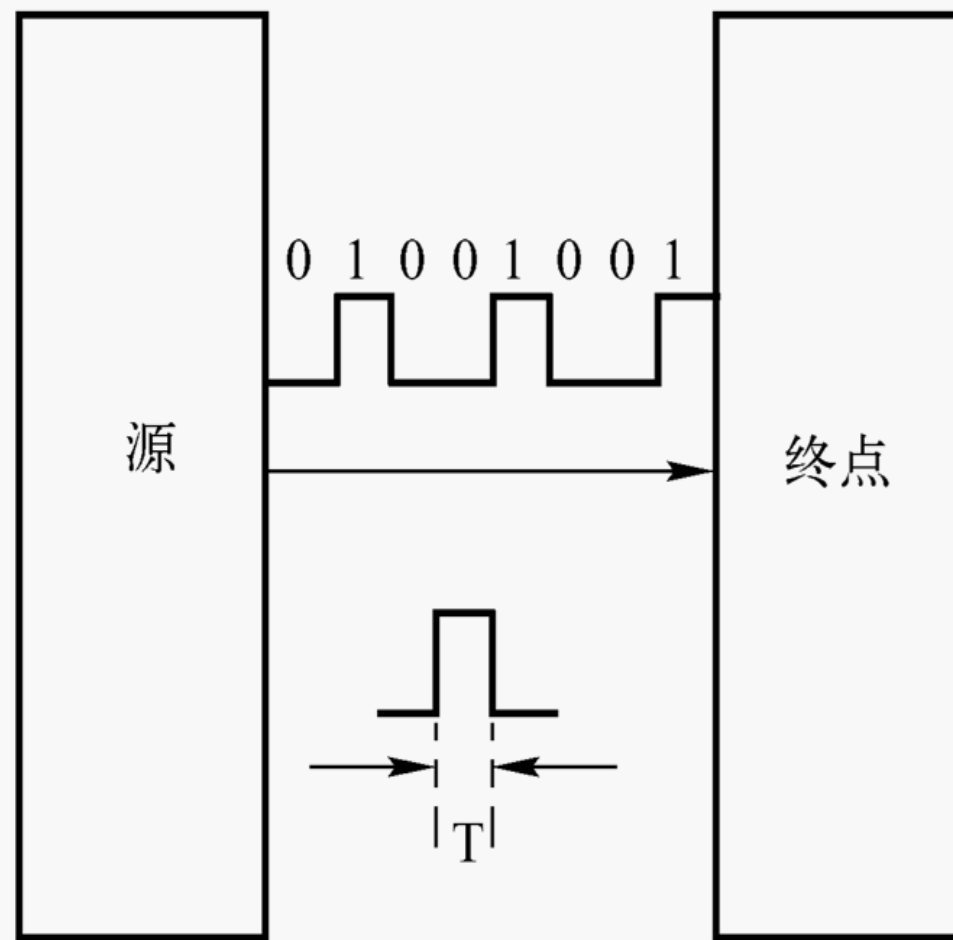
并行通信（并行数据传送）

- 数据在多位宽的传输线上各位同时进行传送

串/并行通信的基本原理



(a) 并行通信



(b) 串行通信

串/并行通信的比较



串行通信

传输线数量少

同频率下，数据传输率较低

需要经过复杂的串/并转换

避免了信号线之间的串扰

并行通信

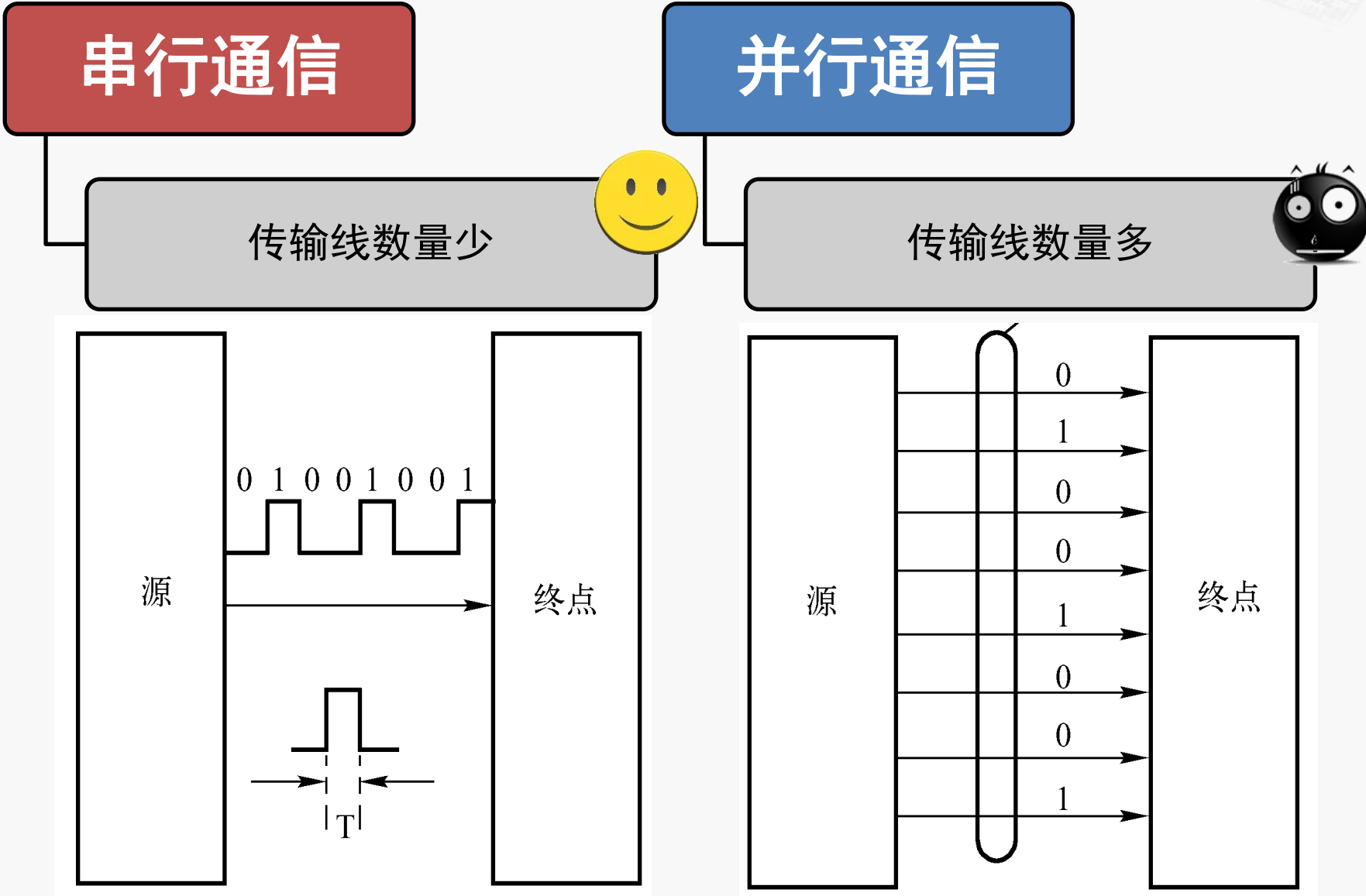
传输线数量多

同频率下，数据传输率较高

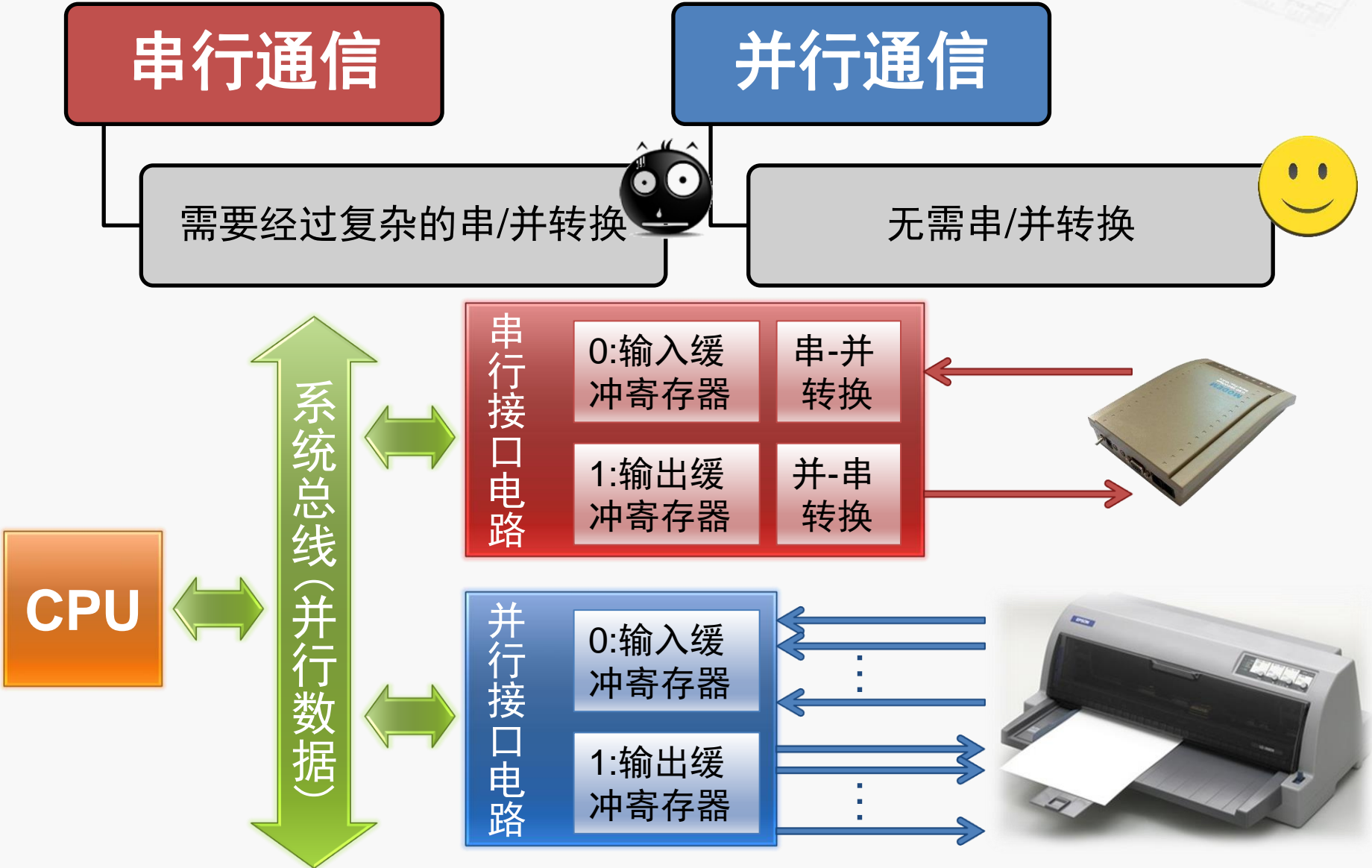
无需串/并转换

存在信号线之间的串扰

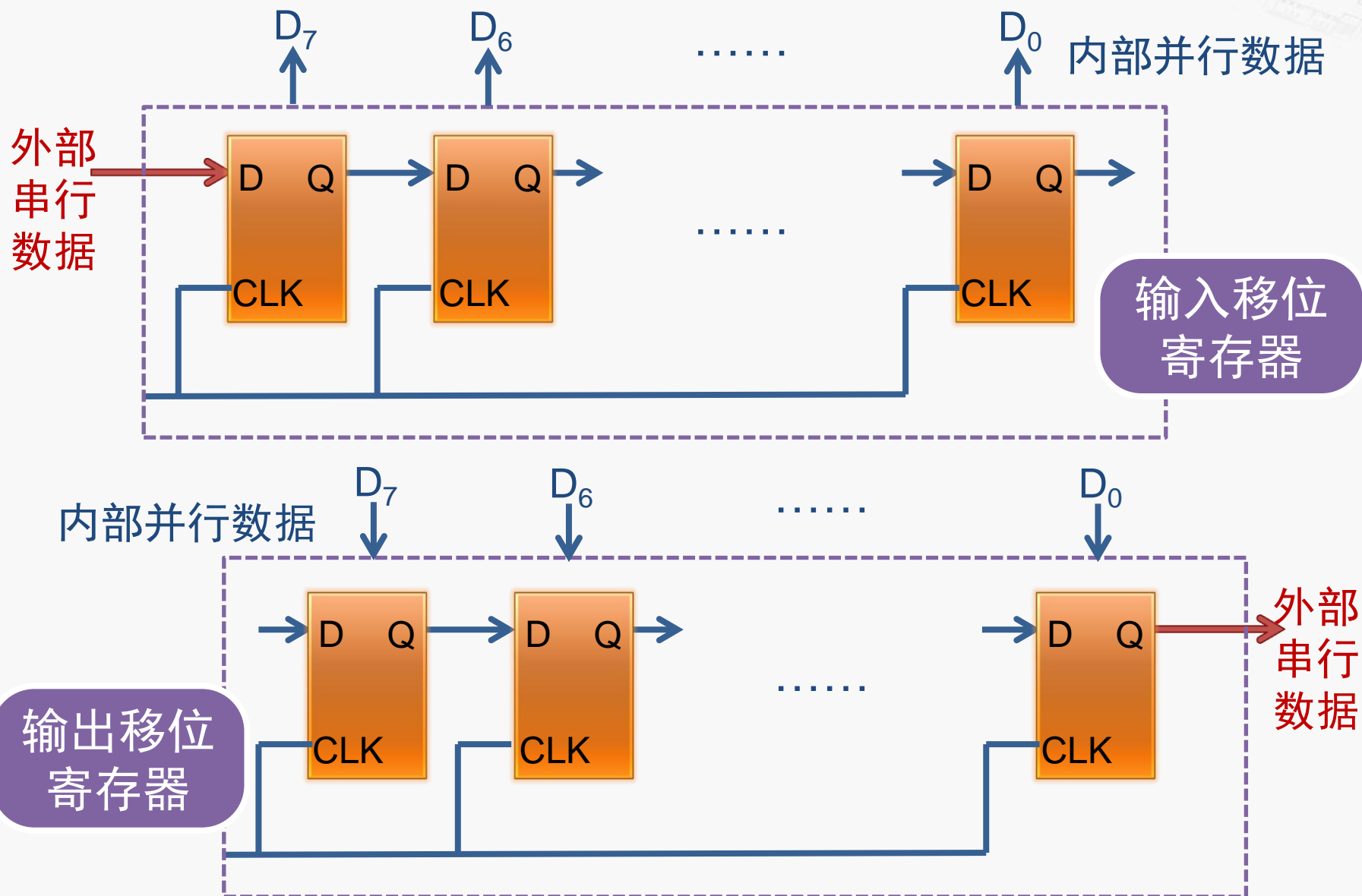
串/并行通信的比较 (1)



串/并行通信的比较（2）



串/并转换



串/并行通信的比较 (3)

串行通信

同频率下，数据传输率较低

串口：
9针RS-232接口

20KB/s

串口：
25针RS-232接口

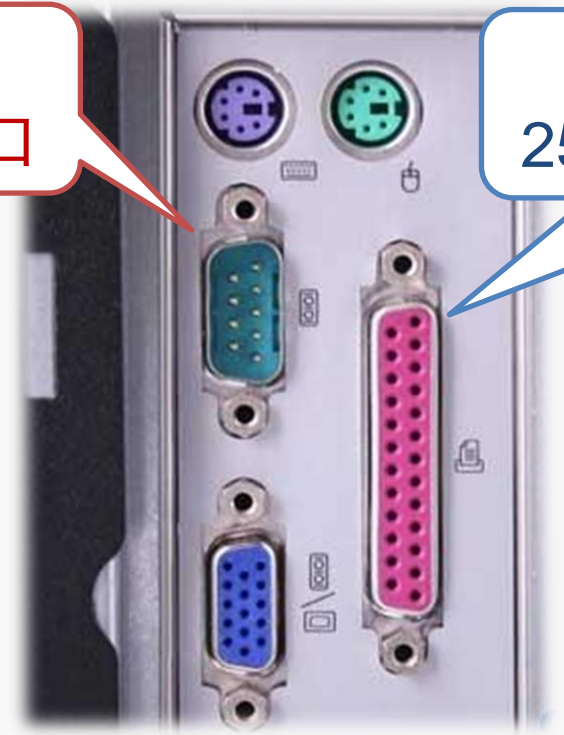


并行通信

同频率下，数据传输率较高

并口：
25针IEEE-1284接口

300KB/s



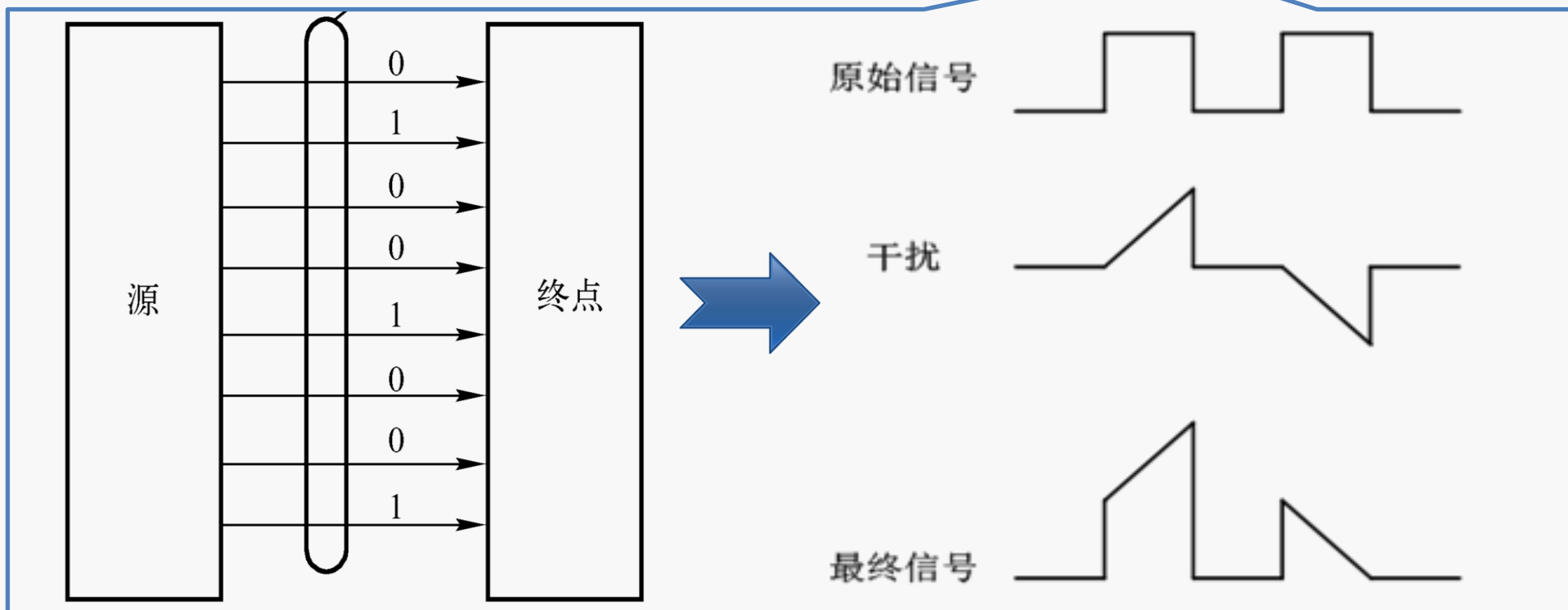
串/并行通信的比较 (4)

串行通信

避免了信号线之间的串扰

并行通信

存在信号线之间的串扰



串/并行通信的比较

串行通信

传输线数量少



需要经过复杂的串/并转换



同频率下，数据传输率较低



避免了信号线之间的串扰



并行通信

传输线数量多



无需串/并转换



同频率下，数据传输率较高



存在信号线之间的串扰

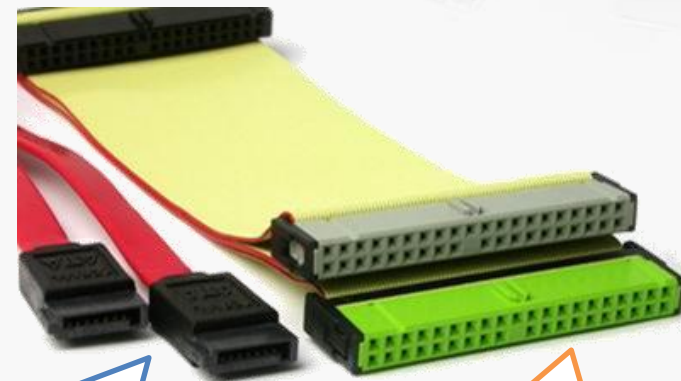


新型串行通信的兴起



IEEE 1394
50~400MB/s

USB
1.5~640MB/s



SATA
150~600MB/s

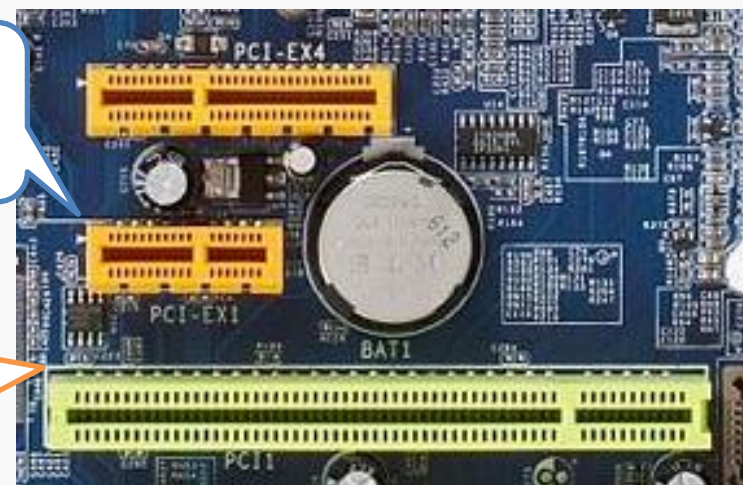
PATA
133MB/s

Intel QPI
34~96GB/s

AMD HT
41GB/s

PCI Express
500M~8GB/s

PCI
133MB/s



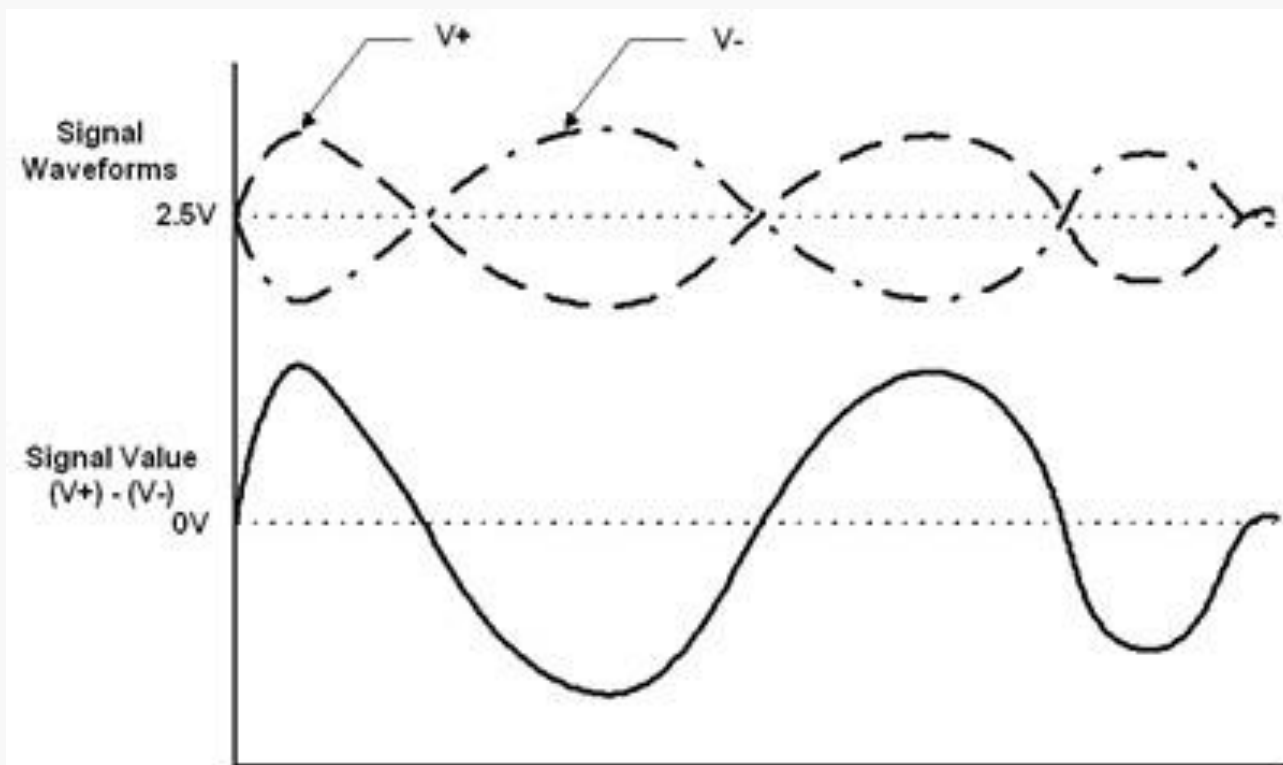
*上述数据传输率并非全面精确数值，仅供参考

串行通信兴起的技术背景



④ 差分信号传输技术

- 。发送端在两根线上发送振幅相等、相位相反的信号（即差分信号）
- 。信号接收端比较两个电压的差值，判断发送端发送的是逻辑0还是逻辑1



差分信号的抗干扰特点

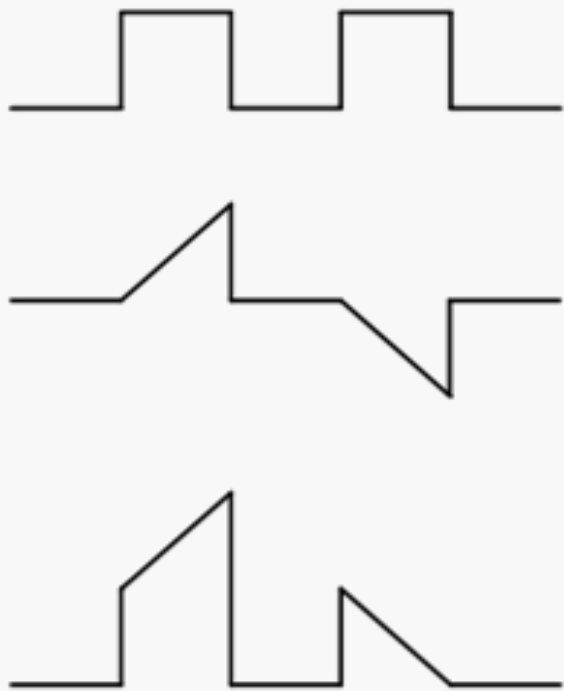
使用差分信号的传输

不使用差分信号的传输

原始信号

干扰

最终信号



原始信号

A+

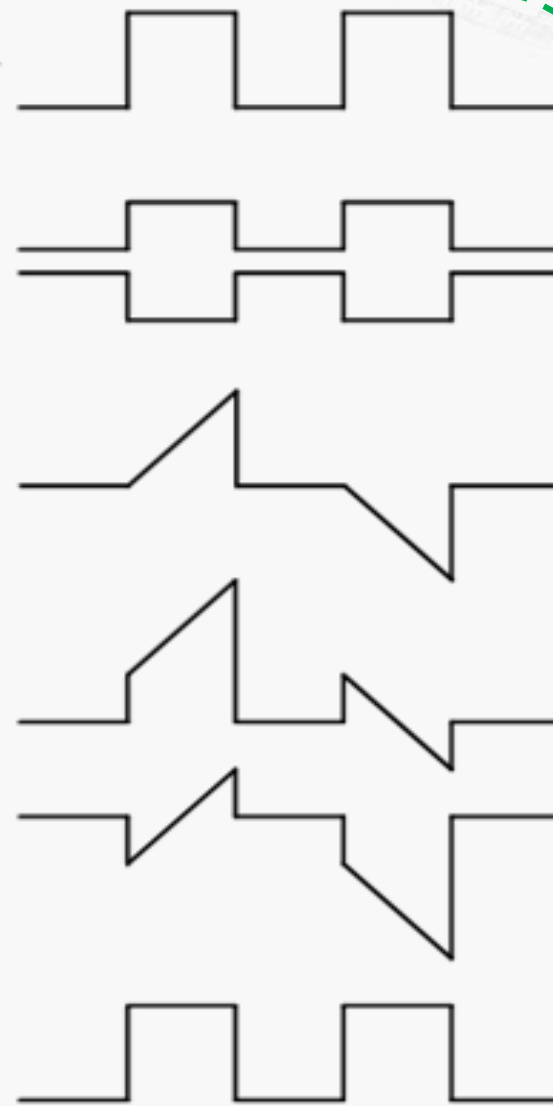
A-

干扰

A+

A-

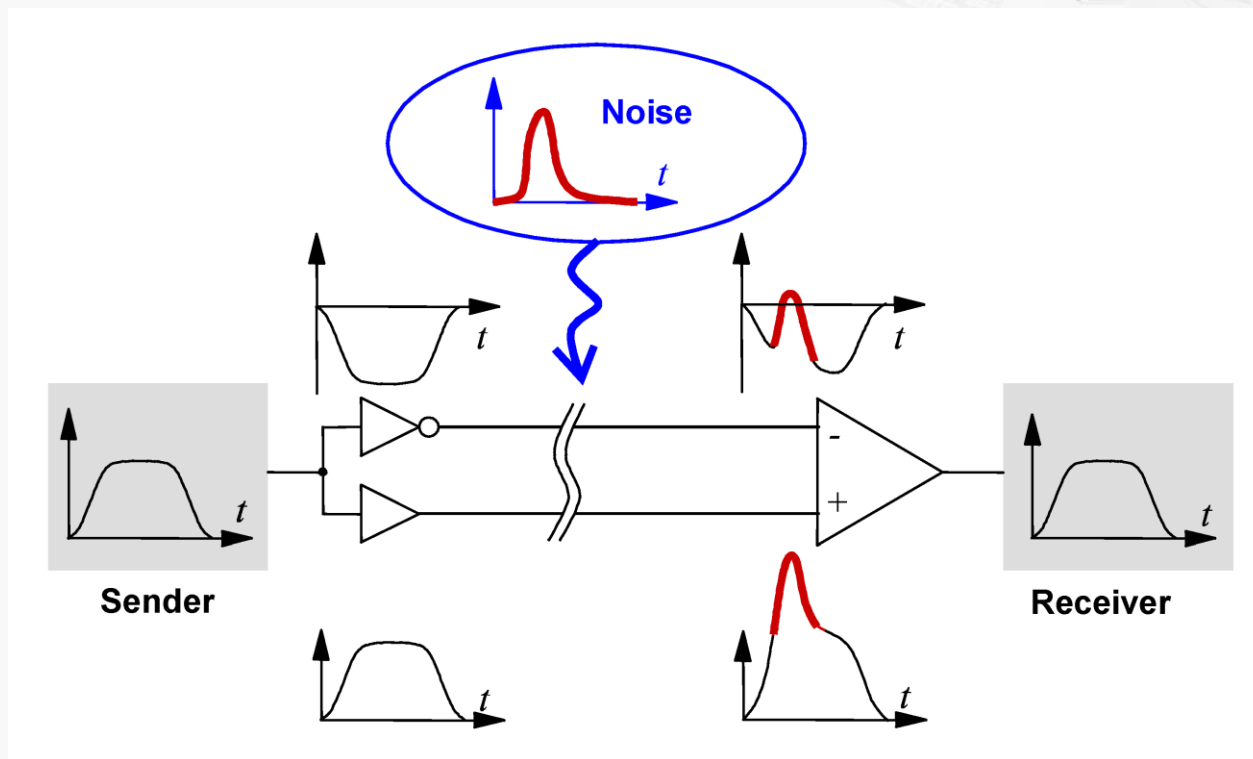
最终信号



差分信号传输技术的优缺点

④ 优点

- 抗干扰能力强
- 能有效抑制电磁干扰
- 时序定位准确



④ 缺点

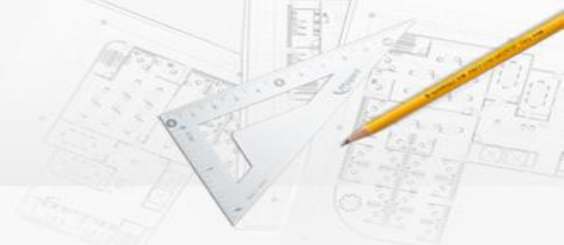
- 在电路板上，差分信号一定要走两根等长、等宽、紧密靠近且在同一层面的线，布线难度高

典型的串/并行通信接口

- ❶ 计算机系统中的“串口”和“并口”
 - “串口”常特指RS-232接口（亦称COM接口）
 - “并口”常特指IEEE-1284接口（亦称LPT接口）



“并口”（IEEE-1284）的发展历程



- 1960年代中期，Centronics公司设计了该接口，用于点阵行式打印机
- 1981年，IBM公司采用该接口，作为IBM PC的标配获得广泛应用，成为打印机的接口标准（直到USB接口的普及）
- 1991年，多家公司联合改进了该接口，实现更高速的双向通信



主要内容

通过学习本课程
了解计算机的发展历程，理解计算机的组成原理，掌握计算机的设计方法

I 输入输出接口电路

II 输入和输出的过程

III 串行和并行的比较

IV 并口芯片的应用实例



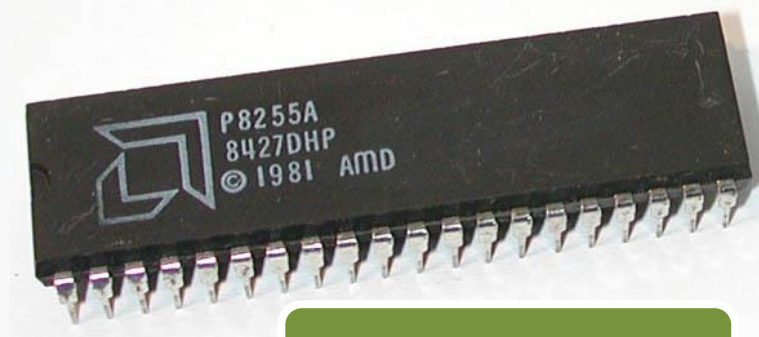
可编程并行接口电路8255A

Intel 8255A（及其它厂商的兼容芯片）

- 为Intel 8080和8085微机系统设计的通用可编程并行接口芯片，也可应用于其他微机系统之中
- 采用40脚双列直插封装，单一+5V电源，全部输入输出与TTL电平兼容
- 用8255A连接外部设备时，通常无需附加其它电路，使用方便



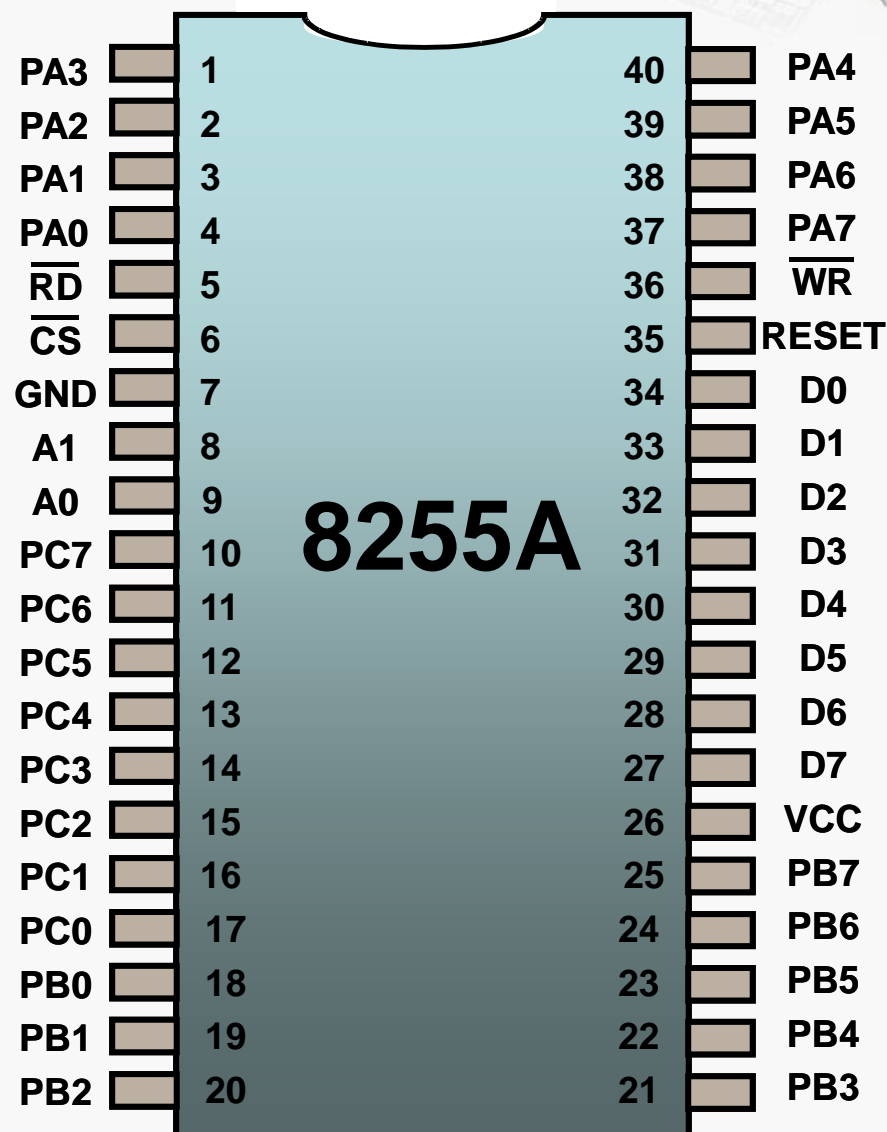
Intel 8255A



AMD 8255A

8255A的外部接口

- ❏ 电源与地线（2条）
 - Vcc, GND
- ❏ 与外设的连线（24条）
 - 端口A: PA₇~PA₀
 - 端口B: PB₇~PB₀
 - 端口C: PC₇~PC₀
- ❏ 与系统总线的连线（14条）
 - 复位: RESET
 - 数据: D₇~D₀
 - 地址: A₁、A₀
 - 控制: CS, RD, WR



与外设的连线



④ 端口A: $PA_7 \sim PA_0$; 端口B: $PB_7 \sim PB_0$

- 均为8位的端口，但端口A的功能更为丰富
- 可分别设定为输入端口或输出端口

④ 端口C: $PC_7 \sim PC_0$

- 分成两个4位的端口，可分别设定为输入端口或输出端口
- 通常不用于普通的数据传送，而是作为端口A和端口B的“握手”信号
 - 端口A与端口C的对应“握手”信号，合称为“A组”
 - 端口B与端口C的对应“握手”信号，合称为“B组”



与系统总线的连线（3）

④ **地址：** A_1 、 A_0 （又称端口选择信号）

- 8255A内部共有4个端口（即可编程寄存器）
- 当CS有效时，根据 A_1 、 A_0 选中其中一个端口

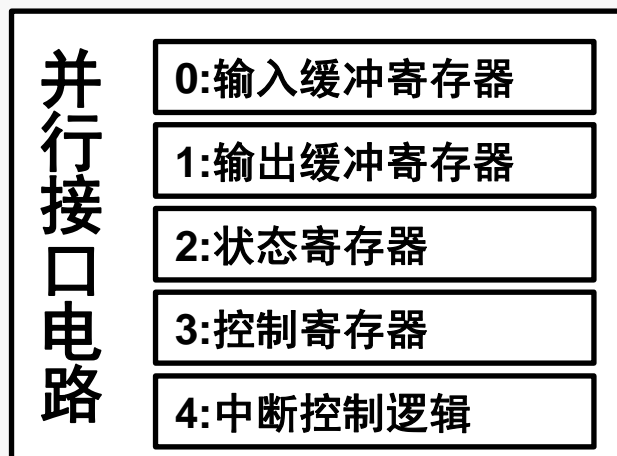
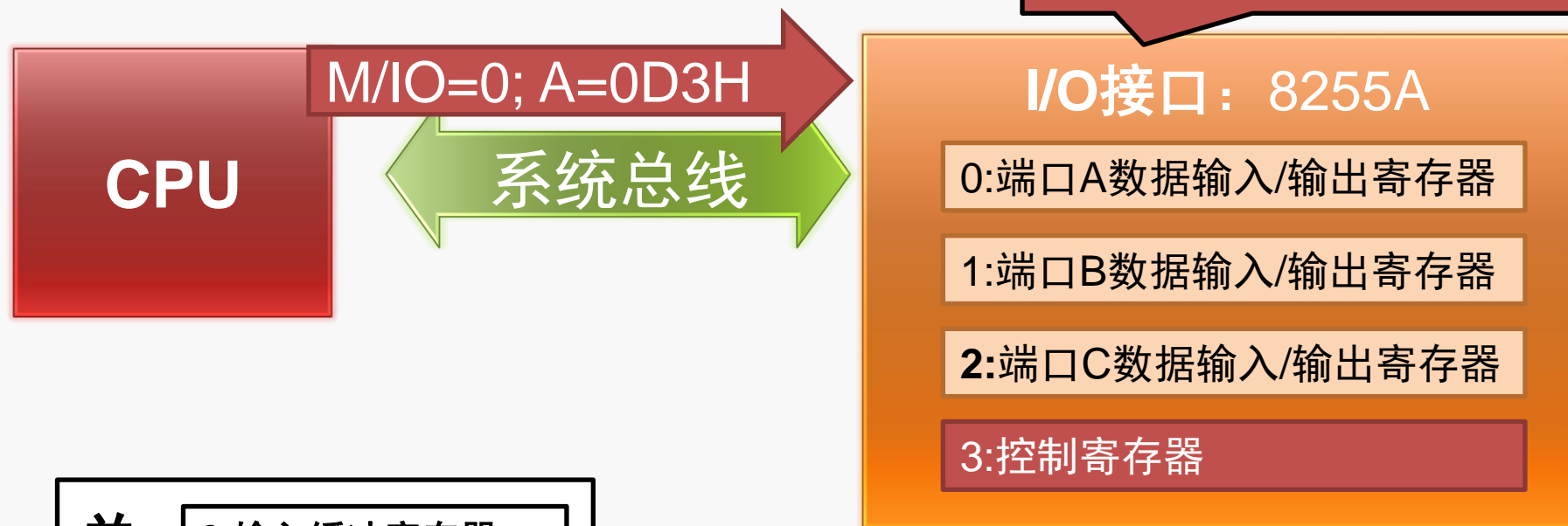
8255A内部端口

A_1	A_0	端口
0	0	端口A
0	1	端口B
1	0	端口C
1	1	控制端口

8255A的初始化编程：设置控制字

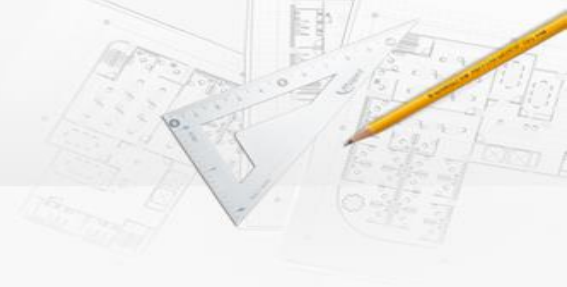
设置控制字：OUT 0D3H, AL

地址：0D0H~0D3H

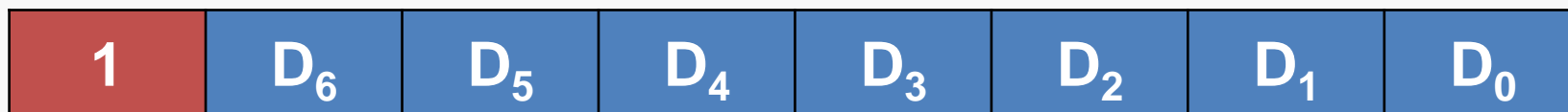


思考：
在内部寄存器设置上，并行接口电路的理想模型与具体实现（即8255A）的差别及原因

8255A的控制字



① 方式选择控制字



② 端口C按位置1/置0控制字



8255A的工作方式



方式0：基本输入/输出方式

- 单向输入/输出。没有规定专门的“握手”信号

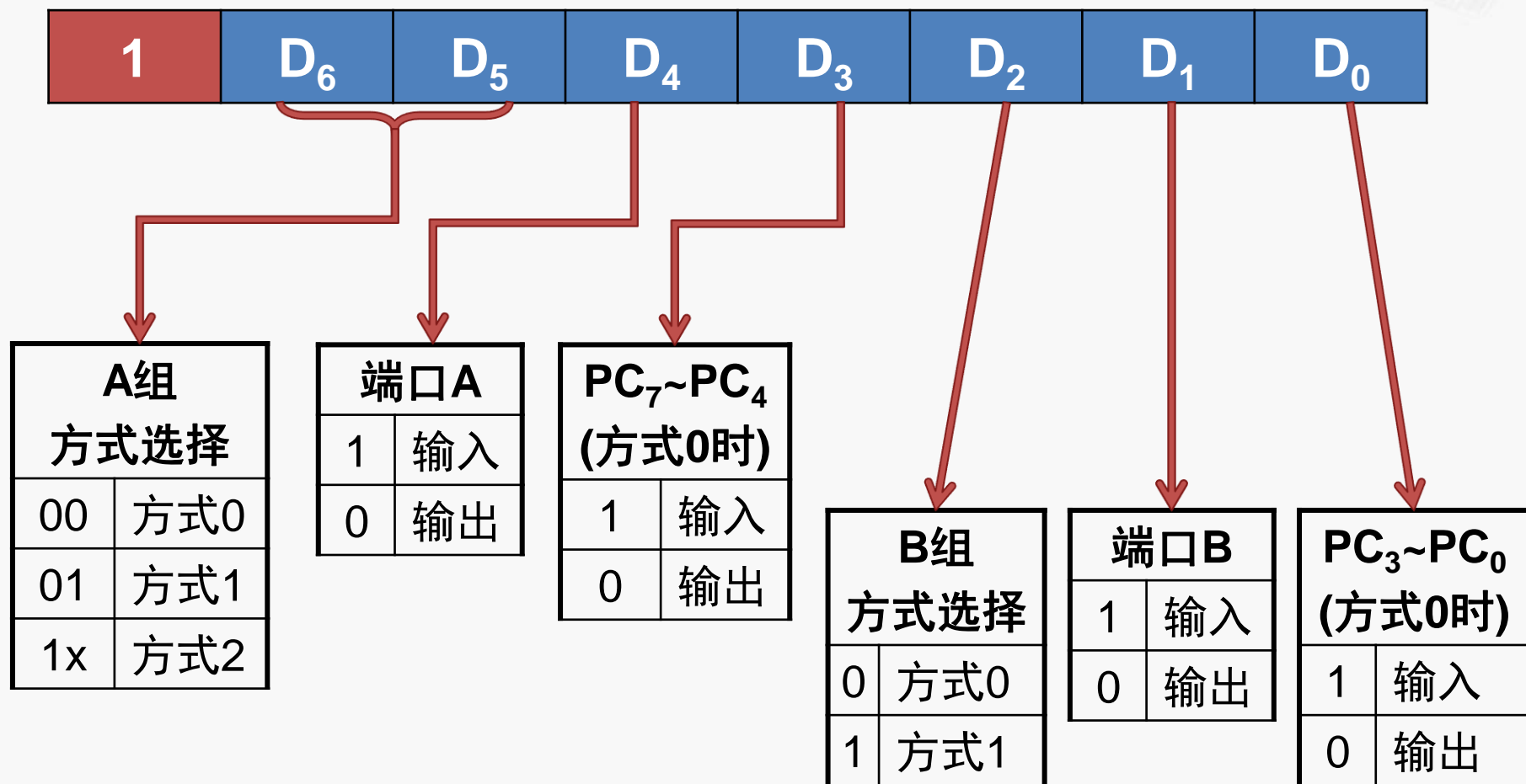
方式1：选通输入/输出方式

- 单向输入/输出。端口C专用于“握手”信号

方式2：双向传输方式

- 双向传输。端口C专用于“握手”信号

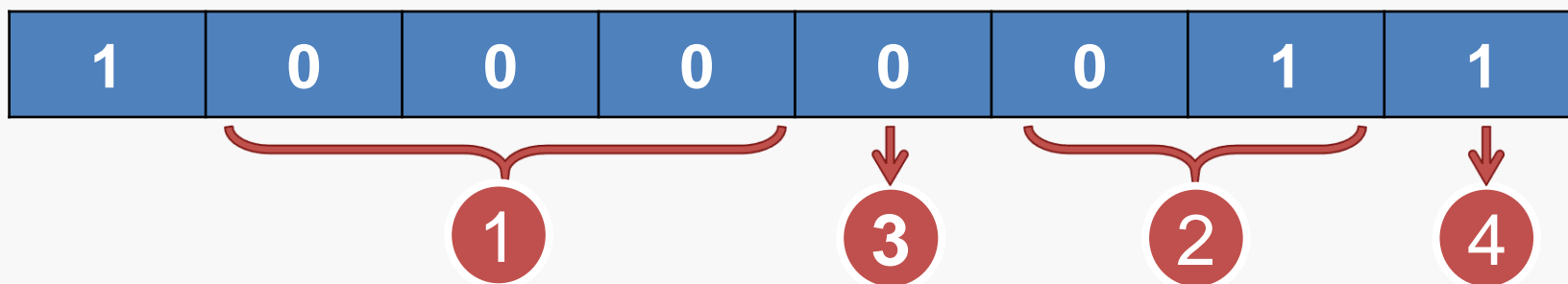
(1) 方式选择控制字



8255A的初始化编程示例

设：要求8255A的各个端口工作于如下方式

- ① 端口A——方式0，输出
- ② 端口B——方式0，输入
- ③ 端口C的高4位——方式0，输出
- ④ 端口C的低4位——方式0，输入



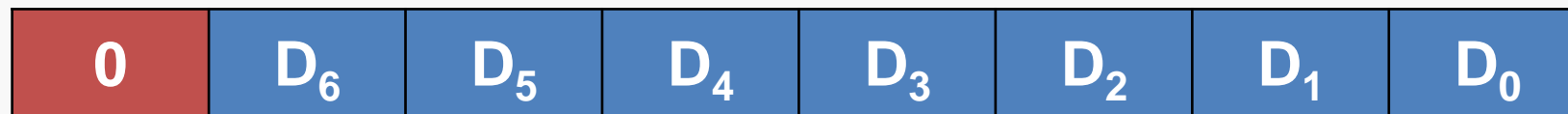
控制字为 10000011B = 83H

8255A的初始化编程示例（续）

🔍 若8255A控制端口的地址为0D3H，则初始化程序如下：

```
MOV    AL,    83H  
OUT    0D3H,  AL           ; 输出方式选择控制字
```

(2) 端口C按位置1/置0控制字



无关

“端口C按位置1 / 置0控制字”
是对端口C的操作控制信息，因此该控制字必须写入控制端口，
而不应写入端口C

位选择	
000	PC ₀
001	PC ₁
010	PC ₂
011	PC ₃
100	PC ₄
101	PC ₅
110	PC ₆
111	PC ₇

1	置1
0	置0

“端口C按位置1/置0”的编程示例

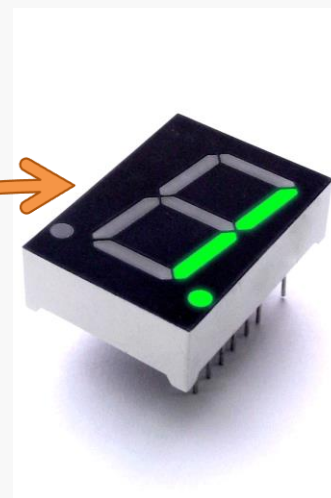
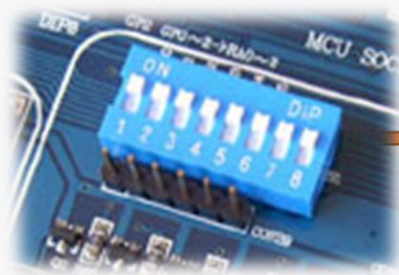


- ④ 设：要求对8255A端口C的PC₆位置0
- ④ 则：控制字应为0000**1100**B(0CH)
- ④ 若8255A控制端口的地址为0D3H，则程序如下：

```
MOV    AL,    0CH
OUT    0D3H,  AL           ; 写入控制字，对PC6位置0
```


8255A应用需求

设计一个系统，使用四个拨码开关，控制一个LED显示管，用于显示十六进制数（0~9，A~F）



8255A的工作方式

方式0：基本输入/输出方式

- 单向输入/输出。没有规定专门的“握手”信号

方式1：选通输入/输出方式

- 单向输入/输出。端口C专用于“握手”信号

方式2：双向传输方式

- 双向传输。端口C专用于“握手”信号

方式0：基本输入/输出方式



- ④ 提供四个并行口，均可用作**输入或输出**
 - 两个8位口（端口A和端口B）
 - 两个4位口（端口C的 $PC_7 \sim PC_4$ 和 $PC_3 \sim PC_0$ ）
- ④ 没有规定专门的“握手”信号
- ④ 常用于与简单外设之间的数据传送
 - 向LED显示器的输出
 - 从二进制开关装置的输入

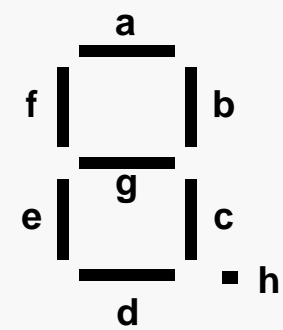
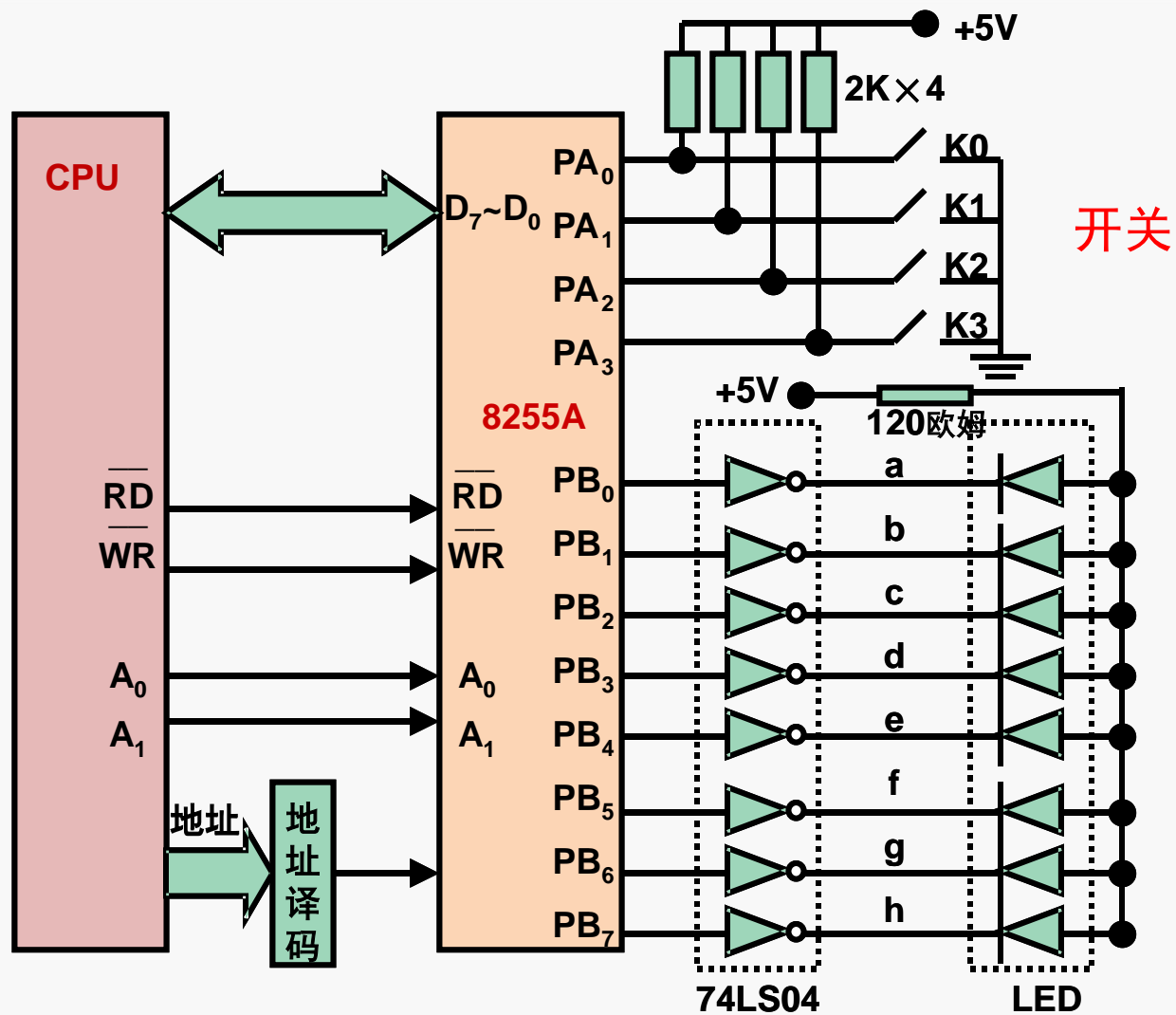
设计方案

④ 使用带8255A的计算机系统，要点如下

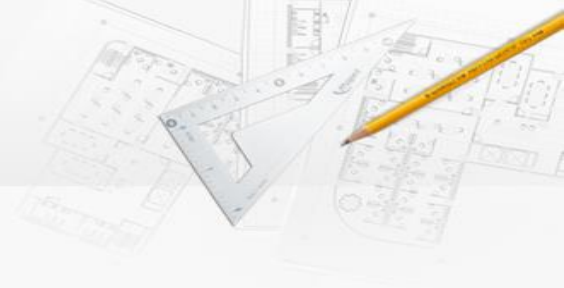
- ① 8255A工作于方式0
- ② 将拨码开关的二进制状态从端口A输入
- ③ 经程序转换为对应的LED段选码(字形码)
- ④ 从端口B输出到LED显示器



系统连线图



LED显示器



初始化及控制程序（概览）

```
DATA SEGMENT
    SSEGCODE DB 3FH, 06H, 5BH, 4FH, 66H, 6DH, 7DH, 07H
              DB 7FH, 67H, 77H, 7CH, 39H, 5EH, 79H, 71H
DATA ENDS
CODE SEGMENT
    ASSUME CS: CODE, DS: DATA
START: MOV AX, DATA
      MOV DS, AX
      MOV AL, 90H ; 设置方式选择控制字, A口工作于方式0输入, B口工作于方式0输出
      OUT 0D3H, AL

RDPORTA: IN AL, 0D0H ; 读A口
          AND AL, 0FH ; 取A口低4位
          MOV BX, OFFSET SSEGCODE ; 取LED段选码表首地址
          XLAT ; 查表, AL←(BX+AL)
          OUT 0D1H, AL ; 从B口输出LED段选码, 显示相应字形符号
          MOV AX, XXXXH ; 延时
          DELAY: DEC AX
                  JNZ DELAY
          MOV AH, 1 ; 判断是否有键按下
          INT 16H
          JZ RDPORTA ; 若无, 则继续读端口A
          MOV AH, 4CH ; 否则返回DOS
          INT 21H
CODE ENDS
      END START
```

设8255A的端口地址如下：

- 端口A： D0H
- 端口B： D1H
- 端口C： D2H
- 控制口： D3H

初始化及控制程序 (1)

DATA **SEGMENT**

```
SSEGCODE  DB  3FH, 06H, 5BH, 4FH, 66H, 6DH, 7DH, 07H
           DB  7FH, 67H, 77H, 7CH, 39H, 5EH, 79H, 71H
```

DATA **ENDS**

CODE **SEGMENT**

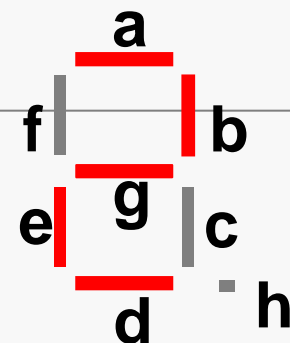
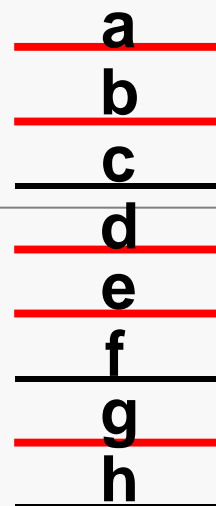
```
        ASSUME  CS: CODE, DS: DATA
```

```
START:  MOV    AX, DATA
```

```
        MOV    DS, AX
```

...

5BH=01011011b



LED显示器

初始化及控制程序（2）

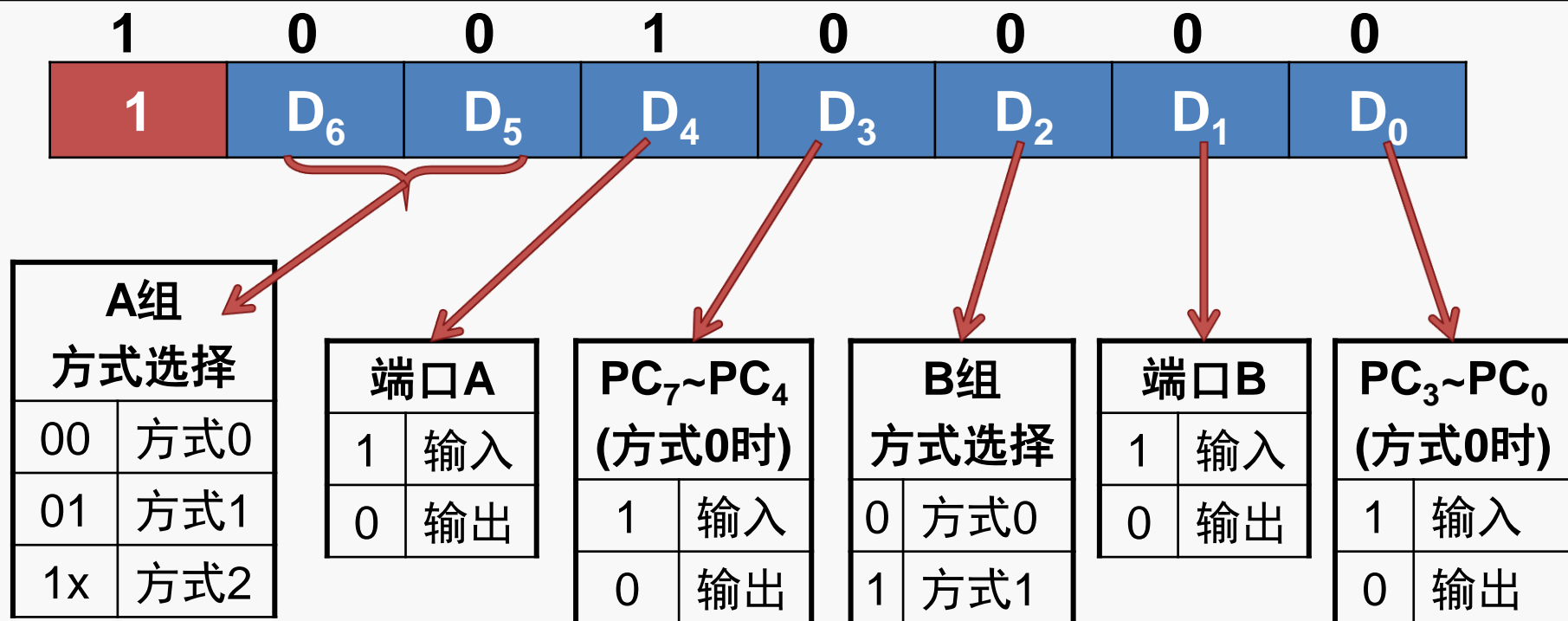
...

```
MOV AL, 90H
```

```
OUT 0D3H, AL
```

...

; 设置方式选择控制字,
; A口工作于方式0输入, B口工作于方式0输出



XLAT指令说明



XLAT指令（换码，查表）

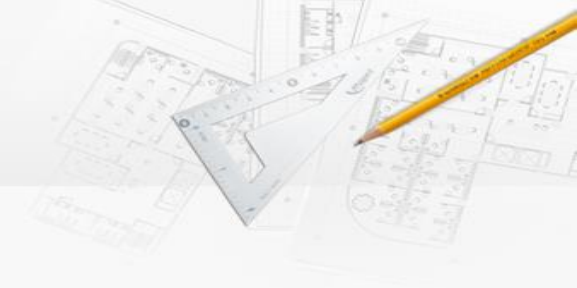
🕒 格式：XLAT

🕒 操作：

（事先在数据段中定义了一个字节型数据表）

- ① 从BX中取得数据表起始地址的偏移量
- ② 从AL中取得数据表项索引值
- ③ 在数据表中查得表项内容
- ④ 将查得的表项内容存入AL

XLAT指令示例



```
SSEGCODE  DB  3FH,  06H,  5BH,  4FH  
             DB  66H,  6DH,  7DH,  07H
```

...

```
MOV    BX, OFFSET SSEGCODE
```

...

MOV	AL, 2	→	02H	AL
XLAT		→	5BH	AL

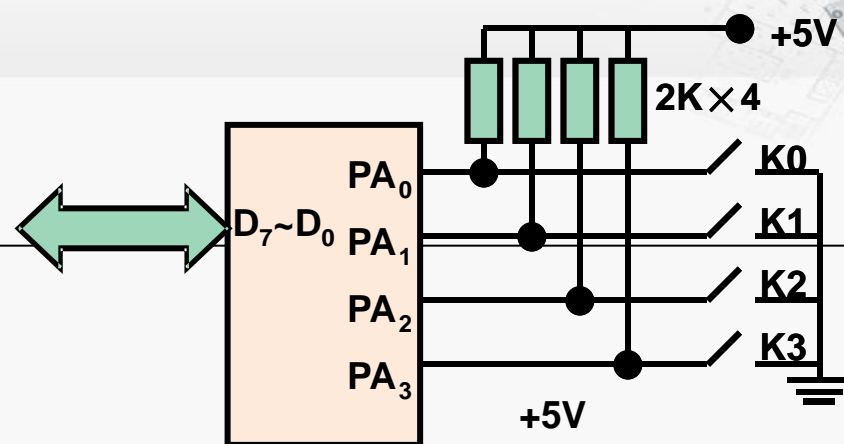
...

MOV	AL, 5	→	05H	AL
XLAT		→	6DH	AL

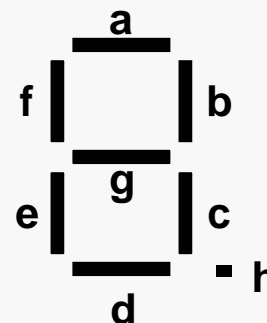
...

初始化及控制程序（3）

```
...
RDPORTA:
    IN    AL, 0D0H      ; 读A口
    AND   AL, 0FH       ; 取A口低4位
    MOV   BX, OFFSET SSEGCODE ; 取LED段选码表首地址
    XLAT                ; 查表, AL←(BX+AL)
    OUT   0D1H, AL      ; 从B口输出LED段选码, 显示字形符号
...
```



```
DATA SEGMENT
    SSEGCODE DB 3FH, 06H, 5BH, 4FH, 66H, 6DH, 7DH, 07H
              DB 7FH, 67H, 77H, 7CH, 39H, 5EH, 79H, 71H
DATA ENDS
```



LED显示器

初始化及控制程序（4）

```
...  
    MOV     AX, XXXXH           ; 通过循环实现延时  
DELAY: DEC     AX  
    JNZ     DELAY  
  
    MOV     AH, 1               ; 判断是否有键按下  
    INT     16H  
    JZ      RDPORTA             ; 若无按键，则继续读端口A  
    MOV     AH, 4CH             ; 否则，程序结束，返回操作系统  
    INT     21H  
  
CODE    ENDS  
        END  START
```

本讲到此结束，谢谢 欢迎继续学习本课程

计算机组织与体系结构 Computer Architectures
主讲：陆俊林

