



微型计算机原理与接口技术

总复习

Homepage: <https://auto.hdu.edu.cn/2019/0403/c3803a93084/page.htm>

Email: qsshe@hdu.edu.cn

Mob: 13758167196

Office: 第二教研楼南楼308室

There's always more to learn, and there are always better ways to do what you've done before.

—DONALD E. KNUTH

微机技术的两大分支

两大分支：通用微型计算机、嵌入式微型计算机 (embedded computer)

1. 通用微型计算机：冯·诺依曼的计算机体系结构

典型代表PC机。具有独立形态、功能通用的微型计算机。

组成结构：主机（CPU、存储器、I/O接口）+ 外设（输入设备和输出设备）+ 软件（操作系统、应用程序）

主要用途：科学计算、数据分析、图像处理、模拟仿真、人工智能、多媒体等。



微机技术的两大分支

2. 嵌入式微型计算机（嵌入式系统、单片微型计算机）

CPU+存储器（ROM/RAM）+输入输出（I/O）接口等 → 单芯片

特点：能够嵌入到对象体系中的专用计算机系统； **嵌入性、专用性、计算机。**

目的：面向测量、控制的应用，嵌入到对象体系中，实现对象体系的智能化。

32位微处理器：通常需要具有小型、用户可裁剪的操作系统（如linux、wince、 μ sDOS等），如ARM、PowerPC等。

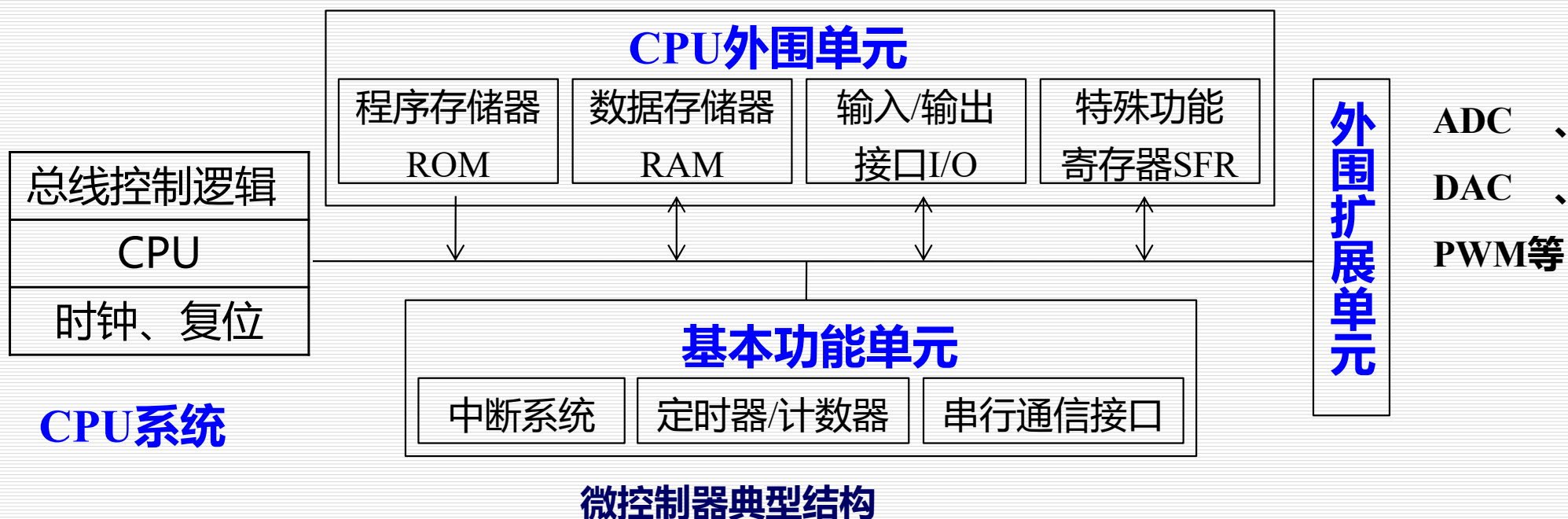
8位微控制器（Microcontroller Unit, MCU）：面向测控应用的单片微型计算机，是使用最广泛的嵌入式系统。



微机技术的两大分支

3. 微控制器典型组成结构

- **存储器结构**：哈佛结构（微控制器常用）、普林斯顿结构（通用微型计算机常用）。
- **指令集体系**：复杂指令集计算机（CISC）和精简指令集计算机（RISC）。



CPU系统 + CPU外围单元 + 基本功能单元 → 微控制器基核

微控制器基核 + 外围扩展单元 → 多种型号、多种功能的微控制器



微机技术的两大分支

4. 微控制器的性能与发展

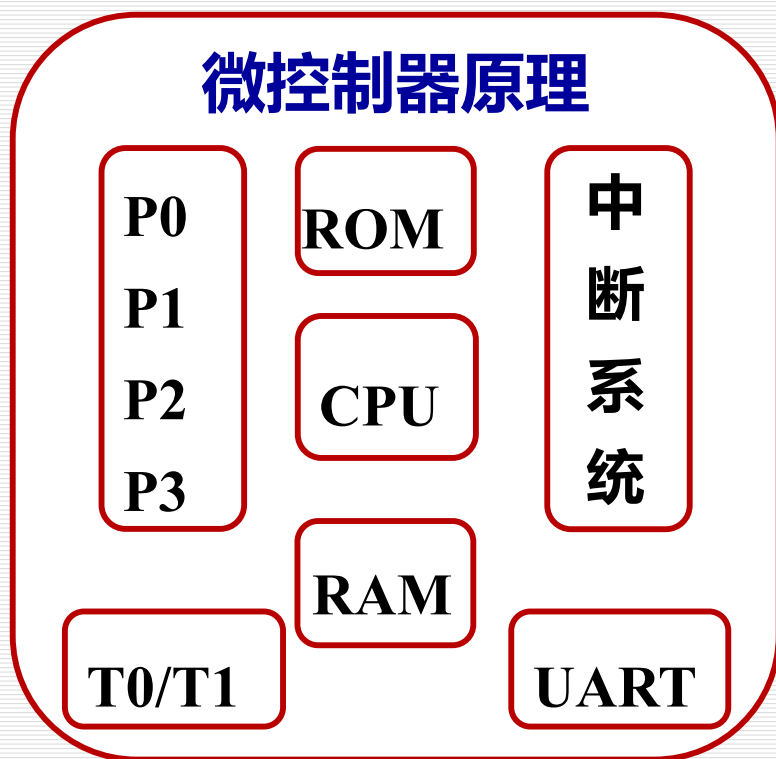
➤ 主要性能:

时钟频率、CPU位数、位处理器（能力）、指令系统、存储容量、I/O端口数量与性能、基本功能模块性能与数量（中断系统、定时器/计数器、串行接口（如UART、I²C、SPI））、**外围功能单元**（片内外围功能单元及性能，如ADC、DAC、PWM等）、**功耗、工作电压。**

➤ 发展：真正的芯片系统（System On Chip, SOC）

- **I/O性能增强**：**I/O的串行扩展**；**I/O端口的电路结构扩展**（结构形式：多样化，可编程；**电气特性**：推挽方式、开漏输出、弱上拉）；**功能I/O接口的增加**（传感器接口、功率驱动接口、通讯接口、高速I/O接口等）
- **性能多样化**：**高速**（时钟频率提高、RISC体系）、**高性能**（位处理能力、中断、复位等功能）、**低功耗、低电压、小体积、大容量、低价格，编程和调试开发环境的优化。**

课程内容组成结构



计算机的**工作过程就是执行程序的过程**，其需要硬件+软件的共同支持。运行不同的程序可以实现不同的功能。

第1章 微机技术概述

第2章 8051微控制器硬件结构

第5章 中断系统

第6章 定时器/计数器

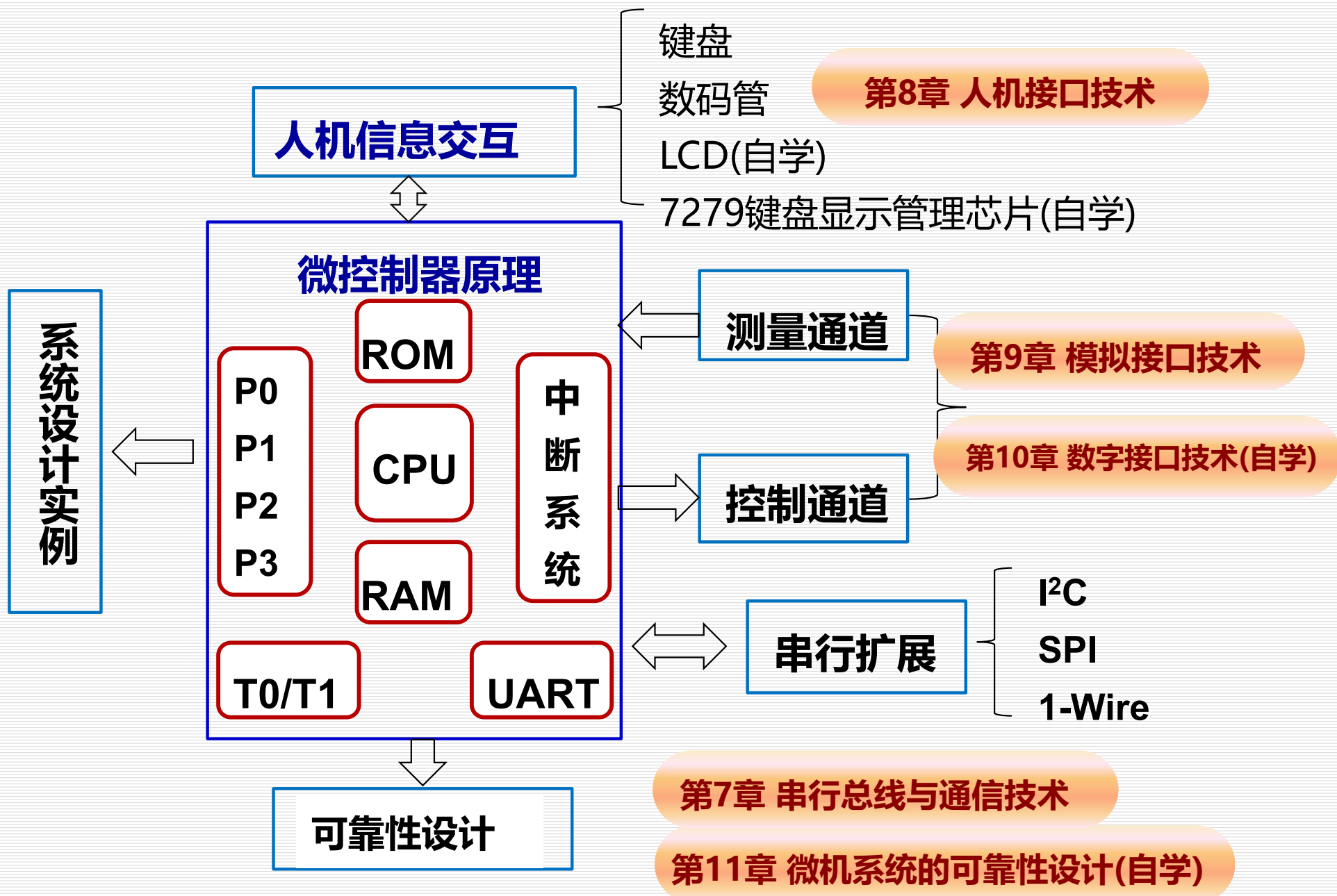
第7章 串行总线与通信技术-UART

第3章 8051指令系统与汇编程序设计

第4章 8051的C语言与程序设计

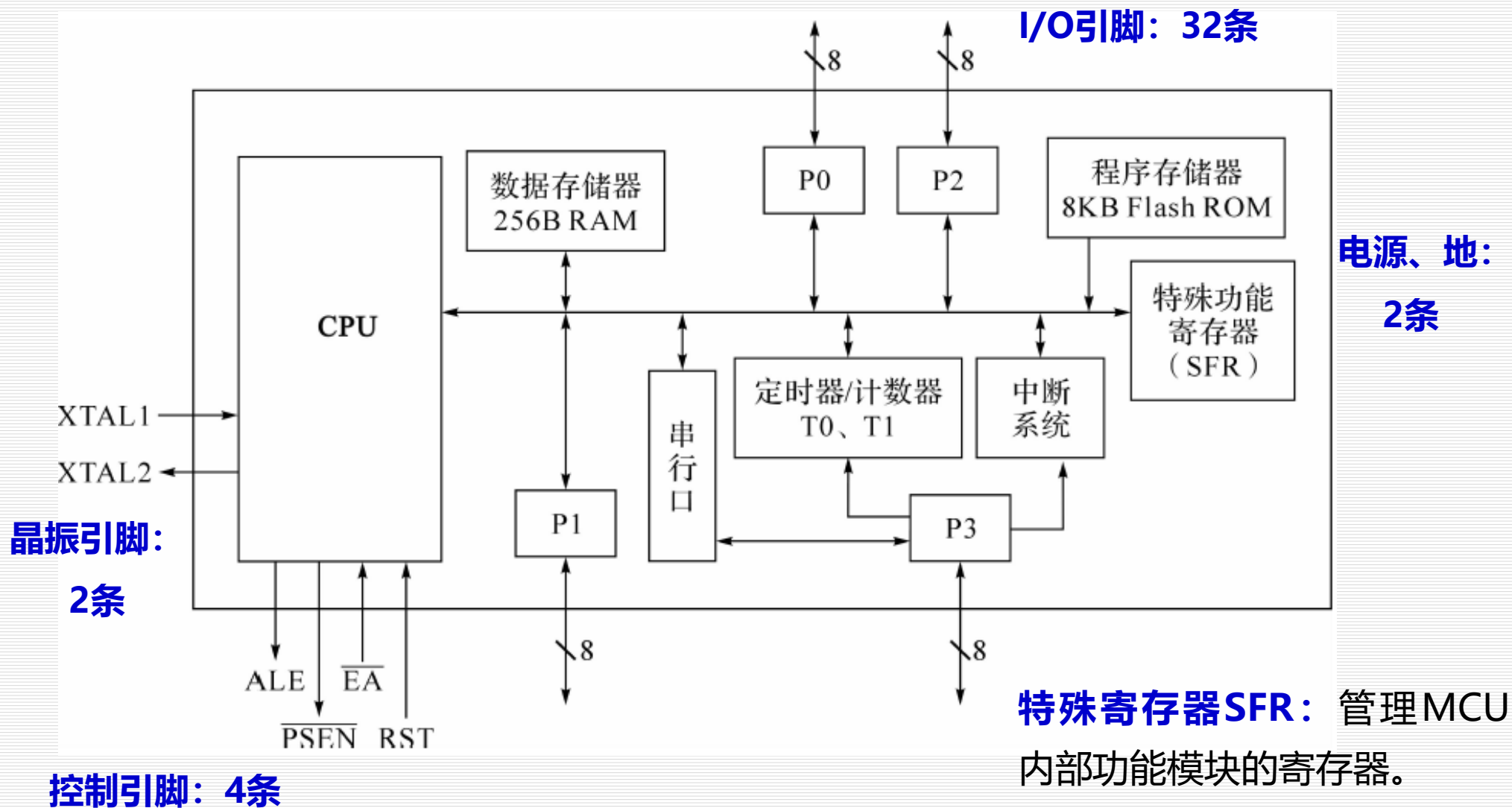
课程内容组成结构

第2章 微控制器应用系统设计(自学)



第2章 8051微控制器硬件结构

1. 8051微控制器的内部结构



指令系统: 111条指令。为复杂指令体系(CISC)。

第2章 8051微控制器硬件结构

2. 微控制器的工作原理

□ (1) CPU组成结构： 运算器+控制器。

➤ 运算器（算术逻辑单元ALU、暂存寄存器、布尔处理器、BCD码运算调整电路）：进行算术运算和逻辑操作的执行部件。

➤ 控制器（指令部件、时序部件和操作控制部件，PC指针）：指挥和控制微控制器工作的部件。

□ (2) CPU的工作过程：执行程序的过程。

➤ 程序：是完成特定功能的指令序列；对应的指令码存放在ROM。

➤ 指令：是微控制器指挥各功能部件工作的指示和命令。操作码+操作数。

➤ 指令执行过程：

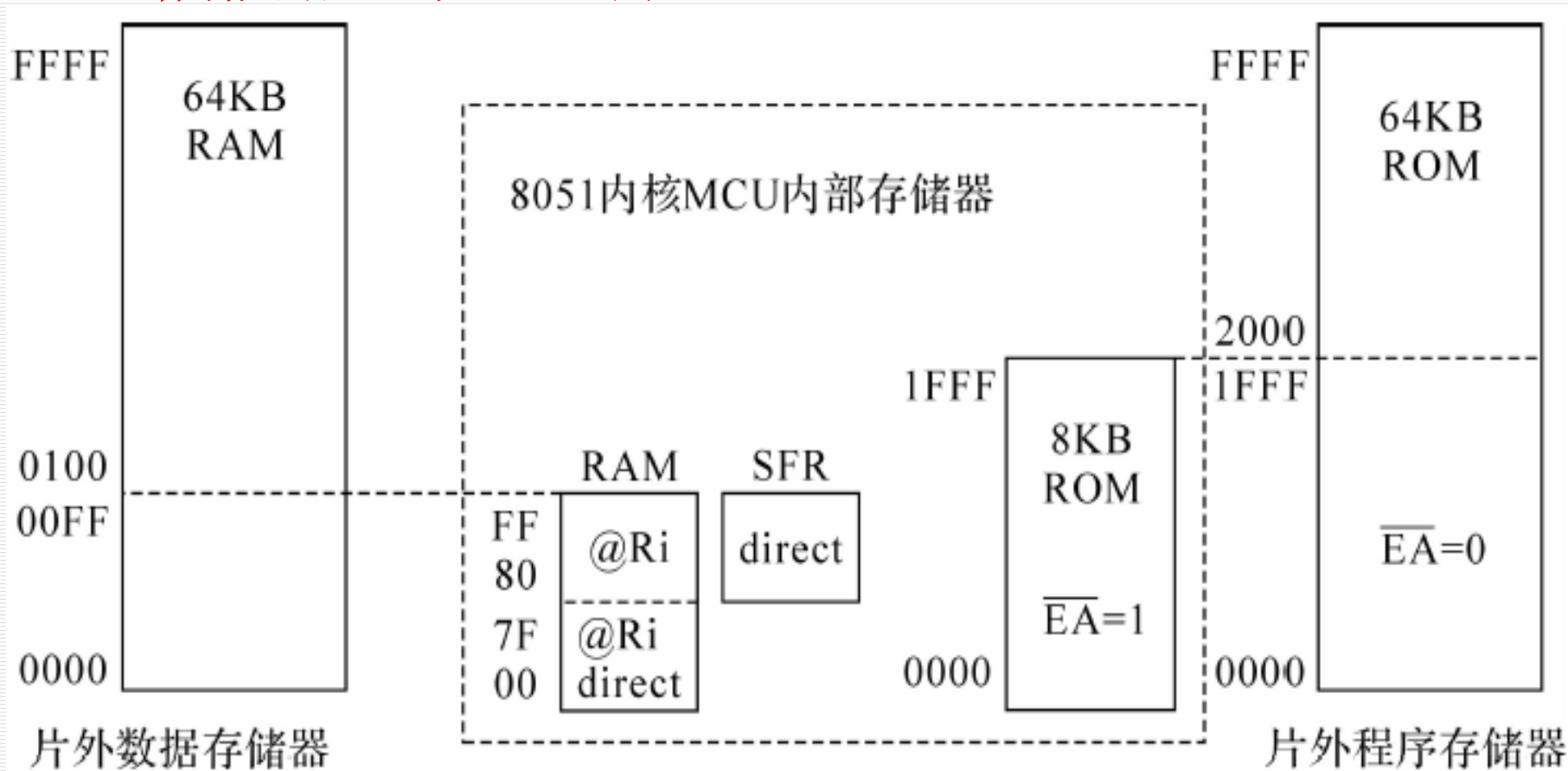
□ 取指令 → 分析指令 → 执行指令

□ （取指） （译码） （执行）



第2章 8051微控制器硬件结构

□ 3. 存储器配置与地址空间



对于MCS-51系列（8031除外）有4个物理上相互独立的存储空间：片内、外程序存储器，片内、外数据存储器。

可分为3个逻辑空间：

- (1) 片内外统一编址的64KB程序存储空间；(MOV_C)
- (2) 256B（51系列）的片内数据存储空间（其中包括特殊功能寄存器空间）；(MOV)
- (3) 64KB片外数据存储空间。(MOV_X)

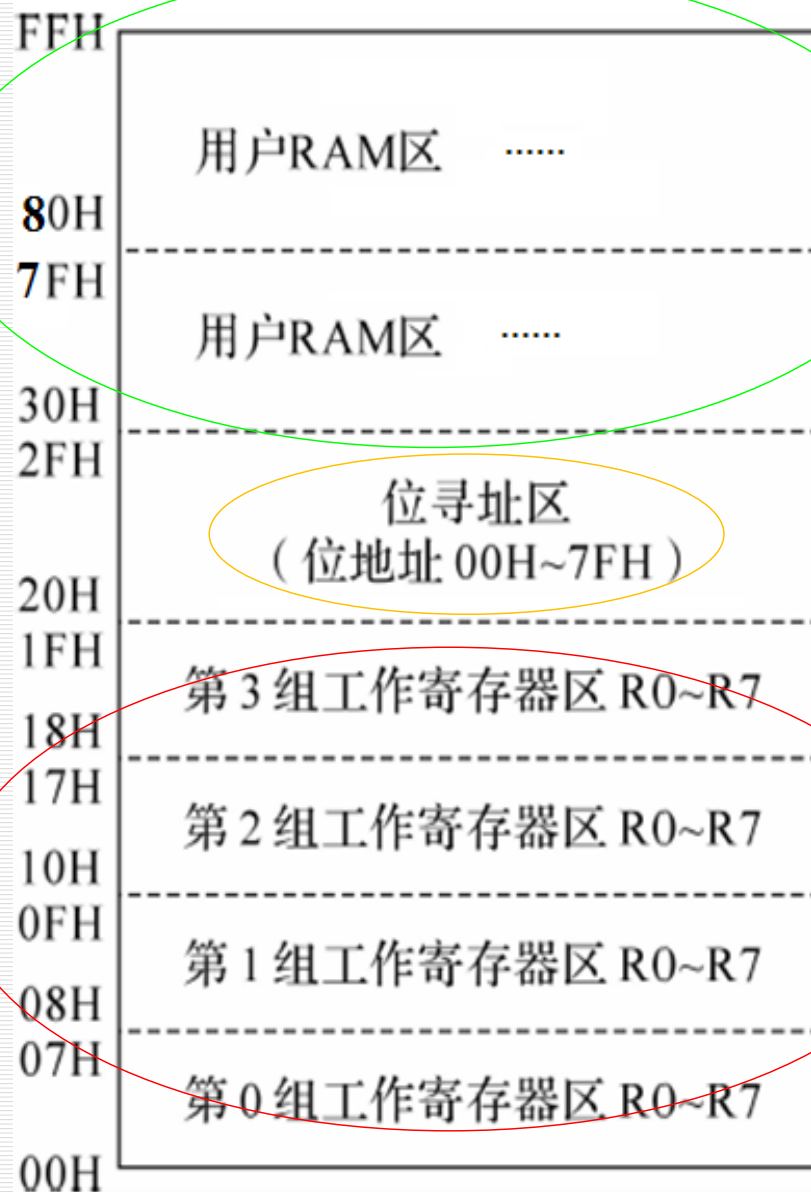
第2章 8051微控制器硬件结构

(1) 内部RAM

- **低128B (00H-7FH)** : 基本数据存储器; **寻址方式: 直接寻址、寄存器间接寻址。**
- **高128B (80H-FFH)** : 扩展数据存储器; **寻址方式: 寄存器间接寻址。**

内部RAM可划分为三块空间:

- 工作寄存器区
- 位寻址区
- 用户RAM区 (包括堆栈)



第2章 8051微控制器硬件结构

(2) 位寻址区

通用RAM中：20H ~ 2FH单元，共**128位**，**位地址为00H ~ 7FH**。

SFR区域中：字节地址的低位为0H或8H的SFR。

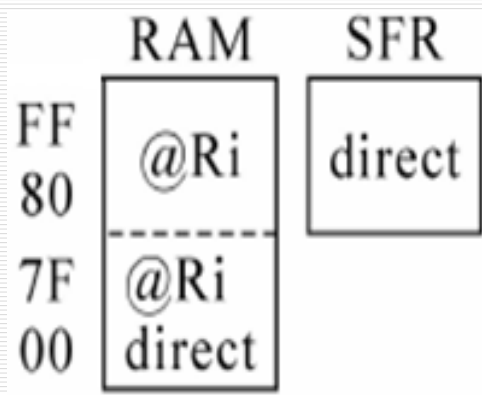
字节	MSB								LSB
FFH									
2FH	7F	7E	7D	7C	7B	7A	79	78	
2EH	77	76	75	74	73	72	71	70	
2DH	6F	6E	6D	6C	6B	6A	69	68	
2CH	67	66	65	64	63	62	61	60	
2BH	5F	5E	5D	5C	5B	5A	59	58	
2AH	57	56	55	54	53	52	51	50	
29H	4F	4E	4D	4C	4B	4A	49	48	
28H	47	46	45	44	43	42	41	40	
27H	3F	3E	3D	3C	3B	3A	39	38	
26H	37	36	35	34	33	32	31	30	
25H	2F	2E	2D	2C	2B	2A	29	28	
24H	27	26	25	24	23	22	21	20	
23H	1F	1E	1D	1C	1B	1A	19	18	
22H	17	16	15	14	13	12	11	10	
21H	0F	0E	0D	0C	0B	0A	09	08	
20H	07	06	05	04	03	02	01	00	
1FH	3 组								
18H									
17H	2 组								
10H									
0FH	1 组								
08H									
07H	0 组								
00H									

内部通用RAM的位寻址空间

第2章 8051微控制器硬件结构

(3) 特殊功能寄存器SFR

- **MCU内部运算、状态寄存器等：**
A、B、PSW、SP、DPH、DPL
- **管理内部功能模块的寄存器：**
P0-P3, IP、IE、TCON、TMOD、TH0/TL0、TH1/TL1, SCON、SBUF、PCON



- **对于00H-7FH：**可运用直接寻址和寄存器间接寻址。
- **对于80H-FFH：**只能采用寄存器间接寻址。
- **对于地址范围同为80H-FFH的SFR：**只能采用直接寻址。

解决2个存储空间的地址重叠问题，避免存储单元访问的冲突，常用的办法是：**采用不同的寻址方式。**

序 号	符 号	地 址
1	B	F0H
2	A	E0H
3	PSW	D0H
4	IP	B8H
5	P3	B0H
6	IE	A8H
7	P2	A0H
8	SBUF	99H
9	SCON	98H
10	P1	90H
11	TH1	8DH
12	TH0	8CH
13	TL1	8BH
14	TL0	8AH
15	TMOD	89H
16	TCON	88H
17	PCON	87H
18	DPH	83H
19	DPL	82H
20	SP	81H
21	P0	80H

第2章 8051微控制器硬件结构

(3) 特殊功能寄存器SFR

程序状态字PSW (Program Status Word)

位地址	D7	D6	D5	D4	D3	D2	D1	D0
位符号	Cy	AC	F0	RS1	RS0	OV	F1	P

- **C (CY) : 进位标志，对于无符号数有意义。**
 - 在加、减法运算时，**发生进位或借位：即C=1，否则C=0。**
 - 对于加法，C=1表示结果超出了8位无符号数的最大值FFH (255) 。
- **OV: 溢出标志，对于带符号数有意义。反映结果是否超出8位带符号数的范围 (-128 ~ +127) 。**
 - OV=1: 溢出； OV=0: 没有溢出。**

实践：例题2-1、2-3



第2章 8051微控制器硬件结构

(3) 特殊功能寄存器SFR

堆栈指针SP (Stack Pointer)

8051 MCU的堆栈为向上生成的软件堆栈，必须开辟在内部通用RAM中。堆栈按照“先进后出”即“后进先出”的原则存取数据。

➤ **堆栈指针SP：**存放堆栈栈顶地址，总是指向堆栈顶部。

数据进栈，SP的内容增加；数据出栈，SP的内容减少。

➤ **堆栈的操作方式：** 指令方式 (PUSH、POP) ；

自动方式（即硬件自动完成，子程序调用和返回，中断响应和返回）。

➤ **堆栈的操作过程**

➤ **堆栈的深度**



第2章 8051微控制器硬件结构

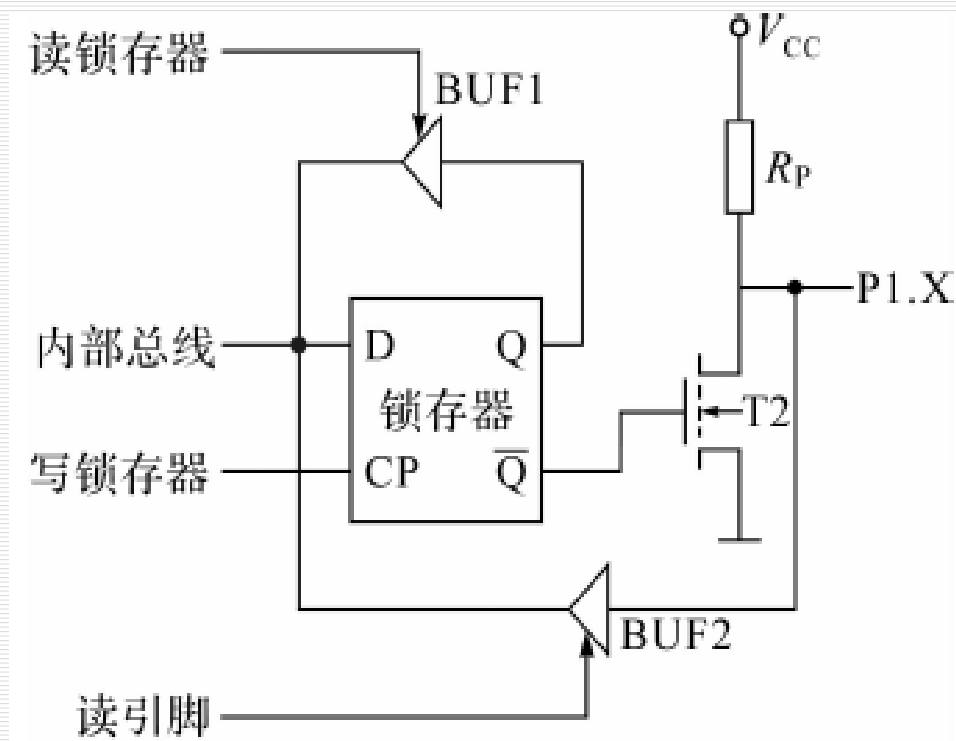
4. P0-P3端口结构与特点

端口内部结构：准双向I/O口结构

P0-P3端口的每一位，均有一个输出锁存D触发器、输出驱动电路组成；以及两个分别用于读锁存器数据和读引脚的三态输入缓冲器BUF1和BUF2。

输出时：向端口锁存器输出1或0，在引脚上出现相应的高低状态。

输入时：首先要向锁存器输出1，使T2处于截止状态。（准双向口的特点）



P1端口内部结构

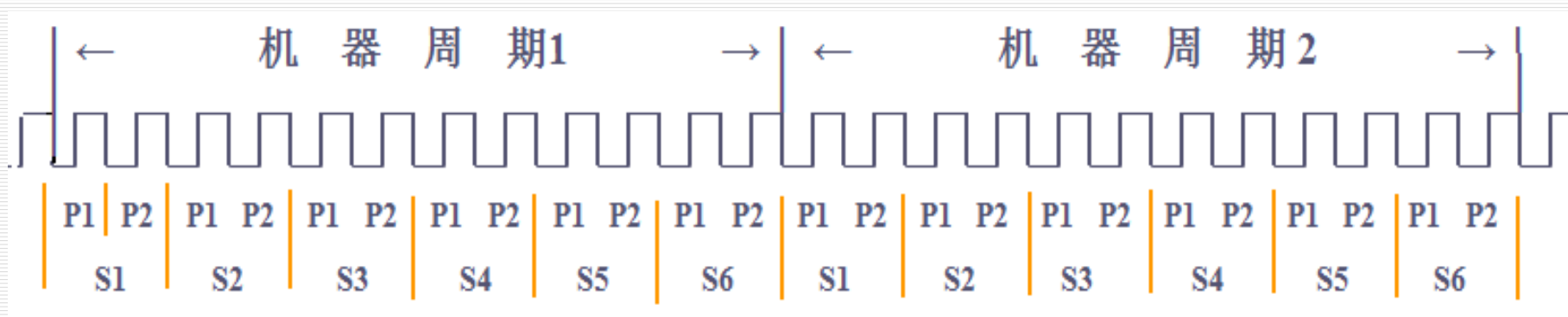
第2章 8051微控制器硬件结构

5. 时钟与工作周期

微控制器的内部电路在时钟信号控制下，严格按时序执行指令规定的操作。

8051 MCU的工作周期：

时钟周期（振荡周期）； 状态周期； 机器周期； 指令周期



1个机器周期 $T_M = 6$ 个状态周期 $S = 12$ 个振荡周期 P

1个指令周期 = 1~4 个 T_M

第2章 8051微控制器硬件结构

6. MCU的工作方式

低功耗方式、程序执行方式和复位方式。

➤ 低功耗方式：

- **休闲方式** (Idle) : CPU不工作。功能模块工作；中断或复位退出该方式。
- **掉电方式** (Power Down) : MCU和功能模块均不工作；用复位退出。

➤ 复位方式：微控制器的初始化操作。

两种复位方式：上电复位，按键复位

➤ 运行方式：执行程序方式；运用用户设计的程序。

课后习题：1、4、5、7、8、9、10、12



第3章 8051指令系统与汇编程序设计

1. 指令分类

(1) 按指令的长度（字节数）分类：

- 单字节指令：49条
- 双字节指令：45条
- 三字节指令：17条

(2) 按指令的执行时间(速度)分：

- 单机器周期指令：64条
- 双机器周期指令：45条
- 四机器周期指令：2条

(3) 按指令功能划分：

- 数据传送类指令：29条
- 算术运算类指令：24条
- 逻辑运算类指令：24条
- 控制转移类指令：17条
- 位操作类指令：17条

第3章 8051指令系统与汇编程序设计

□ 如何熟练掌握111条指令？ 在编写汇编程序时：

➤ 能够根据要实现的功能，想到应该使用的指令？

➤ 该指令的操作数能够是哪些？（别写出不存在的指令）

基本方法：

- 对于一个助记符（如MOV、ADD、SUBB、ANL等等）有多条指令的情况，其

基本特征：指令中的操作数是五种数据的组合。

- 记住特殊情况（多的指令或**没有的指令**）
- 对于一个助记符比较少的指令情况，通过**实践记忆**
- 多阅读程序，然后编写程序。如**例题3-14、3-15、3-16、3-24、3-28、3-30、3-32、3-35、3-36等**

五种数据（五种操作数）：A和四大天王

A, Rn, @Ri, direct, #data

第3章 8051指令系统与汇编程序设计

A, Rn, @Ri, direct, #data

指令的典型格式:

标号: 助记符 目的操作数, 源操作数 ;注释

(1) 数据传送指令: 29条

MOV指令: 16条 $4 * 4 + 1 + 1 - 1 - 1$

除了#data不能作为目的操作数, 其他4种操作数都可以 (存储数据)

-1: 没有 MOV Rn, @Ri

-1: 没有 MOV @Ri, Rn

工作寄存器组的门开不了两次

+1: 多了 MOV direct1, direct2

+1: 多了 MOV DPTR, #data16



第3章 8051指令系统与汇编程序设计

A, Rn, @Ri, direct, #data

(1) 数据传送指令：除MOV以外

MOVX (4条)：外部RAM操作指令；读写各2条；读写均只能通过A；寄存器间接寻址方式；

MOVC (2条)：ROM操作指令；2条查表指令，通过A；基址寄存器是DPTR和PC；

PUSH, POP：堆栈操作指令；操作数：direct

XCH (3条)：数据交换指令，3条；A为目的操作数；**没有XCH A, #data**

XCHD：只有XCHD A, @Ri

XCH A,Rn;

SWAP A

XCH A,@Ri;

XCH A,direct;

第3章 8051指令系统与汇编程序设计

A, Rn, @Ri, direct, #data

(2) 算术运算类指令：24条

ADD, ADDC, SUBB: 各4条, 目的操作数都是A;

INC 有5条, 5-1+1; **没有INC #data, 但有INC DPTR**

DEC 有4条, **没有DEC DPTR**

前3类指令要影响标志位; Cy是不来管你INC还是DEC的(INC、DEC不会改变Cy)。

另外: MUL, DIV, **DA A** (了解调整原则)

第3章 8051指令系统与汇编程序设计

A, Rn, @Ri, direct, #data

(3) 逻辑运算类指令：24条

➤ 与 (ANL)、或 (ORL)、异或 (XRL)：各6条；

A 为目的操作数4条，A与四大天王；

多了direct为目的操作数（只能跟A和#data），这是为什么？（难记）

➤ 循环移位指令：仅对A，4条（RR/RL、RRC/RLC）；

这个奇葩功能是用来干什么的？（对I/O进行循环操作）

➤ CLR, CPL：仅对A，2条；

A是专业杂技演员（SWAP、DA、CLR、CPL、RR/RL、RRC/RLC、JZ/JNZ）

C是业余杂技演员，白天在PSW工作。

第3章 8051指令系统与汇编程序设计

(4) 控制转移指令：17条

➤ 带着PUSH和POP的飞：LCALL、ACALL、RET、RETI

用PUSH 保存PC，修改PC后起飞；用POP恢复PC，PC起飞。

➤ 直接飞：LJMP、AJMP、SJMP、JMP：飞的范围不一样；

JMP @A+DPTR和MOVC是远房亲戚。

➤ 需要点火的飞：

判零： JZ/JNZ, JC/JNC和JB/JNB不用说了，知道有这些指令与功能就行（杂技演员、位）。

比较： CJNE, 4条（A与direct、#data；Rn、@Ri与#data）；**影响C**

计数： DJNZ, 2条（Rn、direct），为什么everywhere的A不在里面？

空操作： NOP

第3章 8051指令系统与汇编程序设计

(4) 位操作指令：17条

➤ 需要点火的飞：

位转移： JZ/JNZ, **JC/JNC**、**JB/JNB**、**JBC**就不说了，知道有这些指令与功能就行（杂技演员、位）。

JBC这个指令，有什么用吗？

位传送： **MOV (2条)** , C和bit 互为目的和源操作数；

位状态设置： **SETB**、**CLR**、**CPL (各2条)** , C与bit；

位逻辑： **ANL**、**ORL (各2条)** , C与bit、/bit, 结果为C。

课后习题： 1、2、8、13、14、15

程序阅读



第5章 中断系统

CPU与功能模块、外设交换信息的主要方式：**中断方式与查询方式。**

查询方式：

- CPU不断查询外设状况
- 限制处理其他事务的能力

中断方式：

- 快速响应处理突发事件
- 多任务

什么是中断？

是**硬件改变CPU程序运行方向**的一种技术，使得CPU能够及时响应和处理**紧急事件**。

中断的作用

- 分时操作：CPU与外部模块同步工作
- 实时处理：外部模块的操作请求，以及故障处理等。



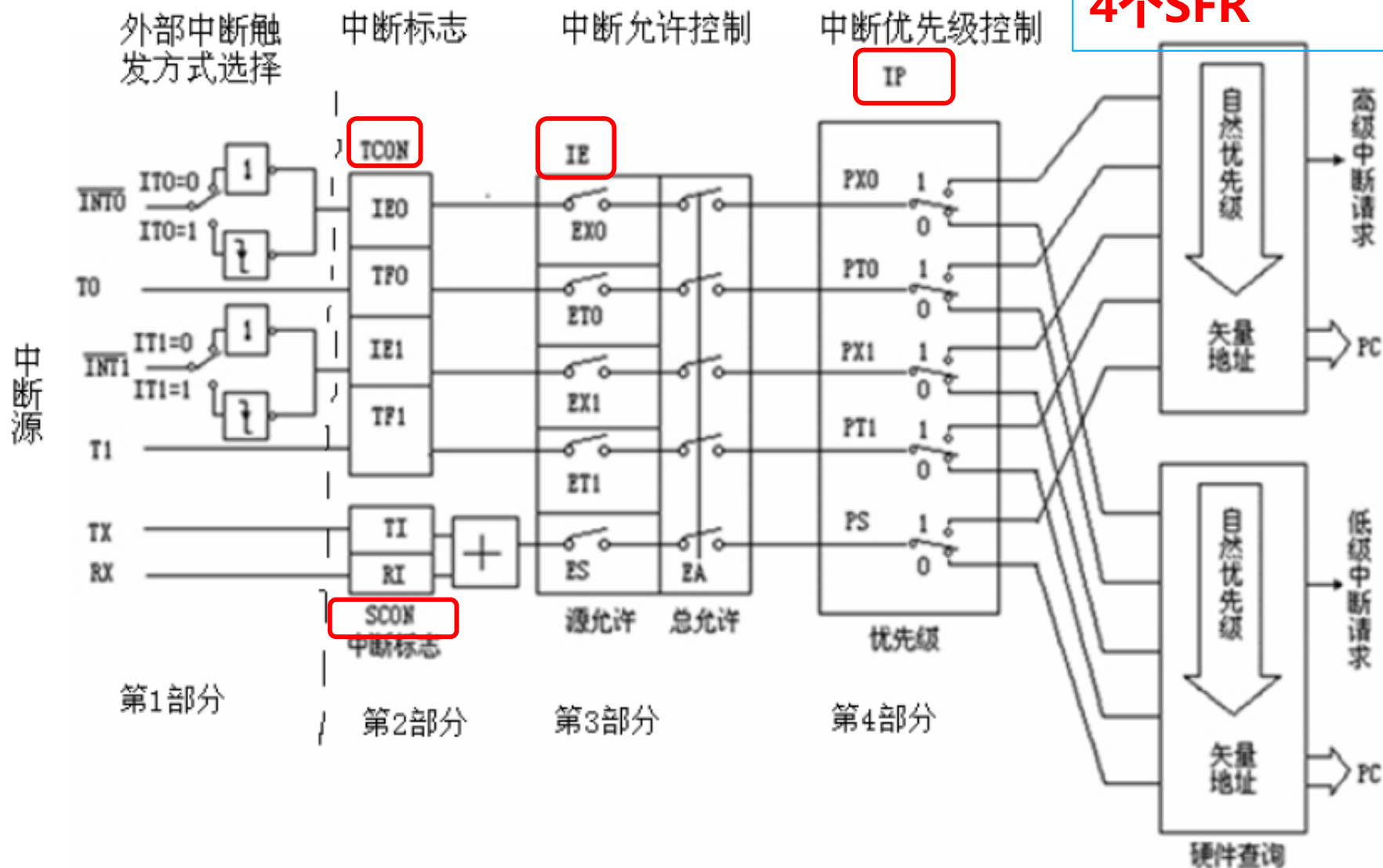
第5章 中断系统

1. 中断系统的结构

5个中断源

2个中断优先级

4个SFR

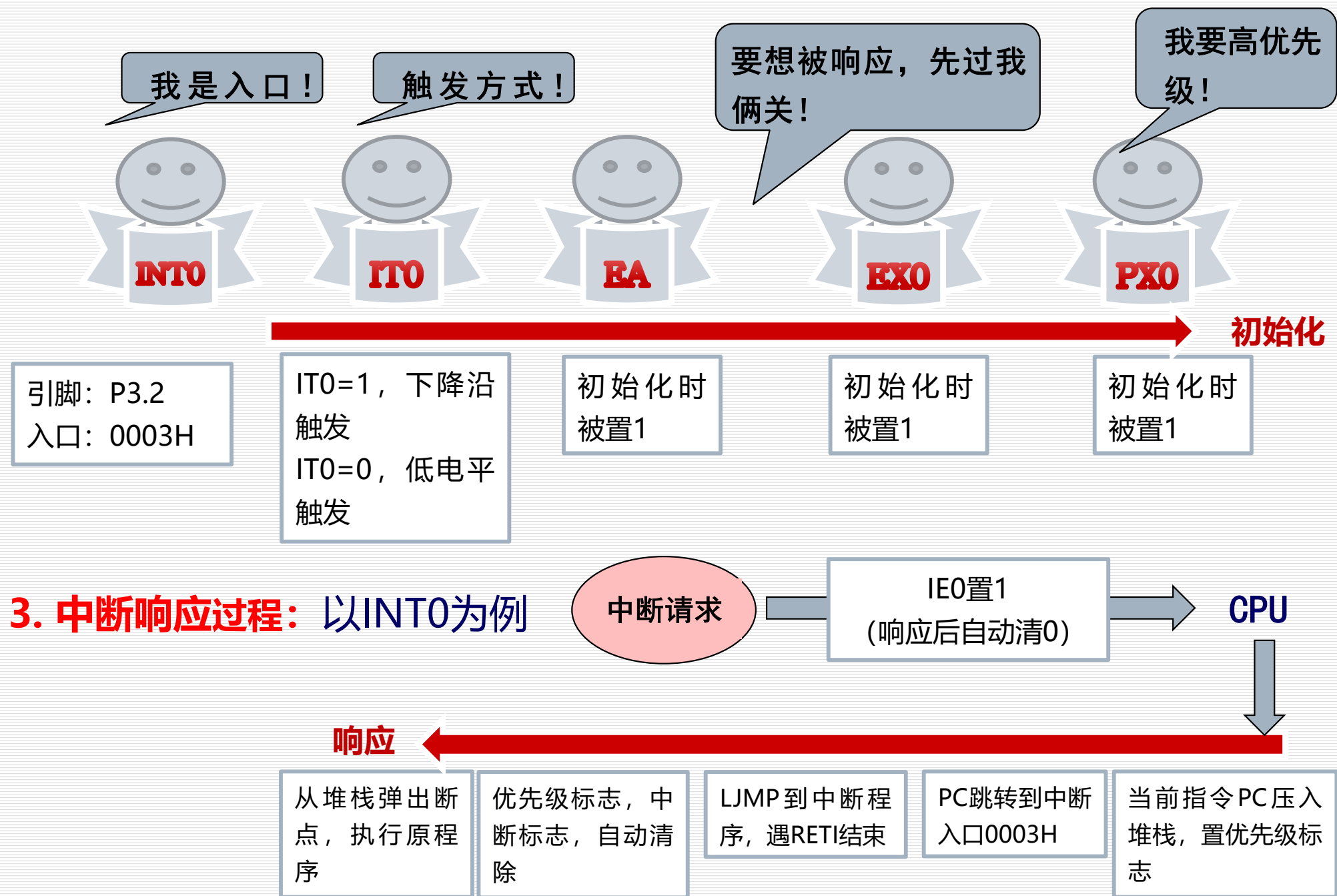


第5章 中断系统

2. 中断标志的产生和清除

中断名称	中断入口	中断源	中断标志	中断标志建立条件	标志清除
外部中断0	0003H	INT0	IE0	符合触发条件时	硬件自动清除（下降沿） 引脚变为高电平（低电平）
外部中断1	000BH	INT1	IE1		
外部中断触发方式：下降沿、低电平；触发方式选择：IT0/IT1					
T0中断	0013H	T0	TF0	T0溢出	中断方式：硬件自动清除 查询方式：软件清除
T1中断	001BH	T1	TF1	T1溢出	
串行口中断	0023H	TX	TI	发送完一帧数据	软件清除
		RX	RI	接收到一帧数据	

第5章 中断系统



第5章 中断系统

4. CPU响应中断的条件

3个基本条件	3个特殊条件
• 中断源发出中断请求	• 无同级或高级中断正在服务
• CPU中断允许位置位, 即 EA=1	• 正在执行的是现行指令的最后一个机器 周期
• 申请中断的中断源允许位置位	• 若执行的是RETI或对IE、IP操作指令, 则要执行完毕该指令的下一条指令

关于中断嵌套

- 高级中断能打断低级中断;
- 低级不能中断高级或同级中断;
- 对于同时请求的高、低级中断请求, 响应高级别;
- 对于同时请求的同级别中断请求, 按自然优先顺序响应。

中断嵌套

响应条件

响应时间

第5章 中断系统

5. 汇编中断服务程序

- 断点自动保护;
- 保护现场和恢复现场：SFR和工作寄存器；3种方法
- 中断程序的安排：入口地址安排无条件转移指令

6. 中断程序设计

- 例题5-1、5-2

课后习题：1、4、11、12



第6章 定时器/计数器

1. 定时器/计数器的功能

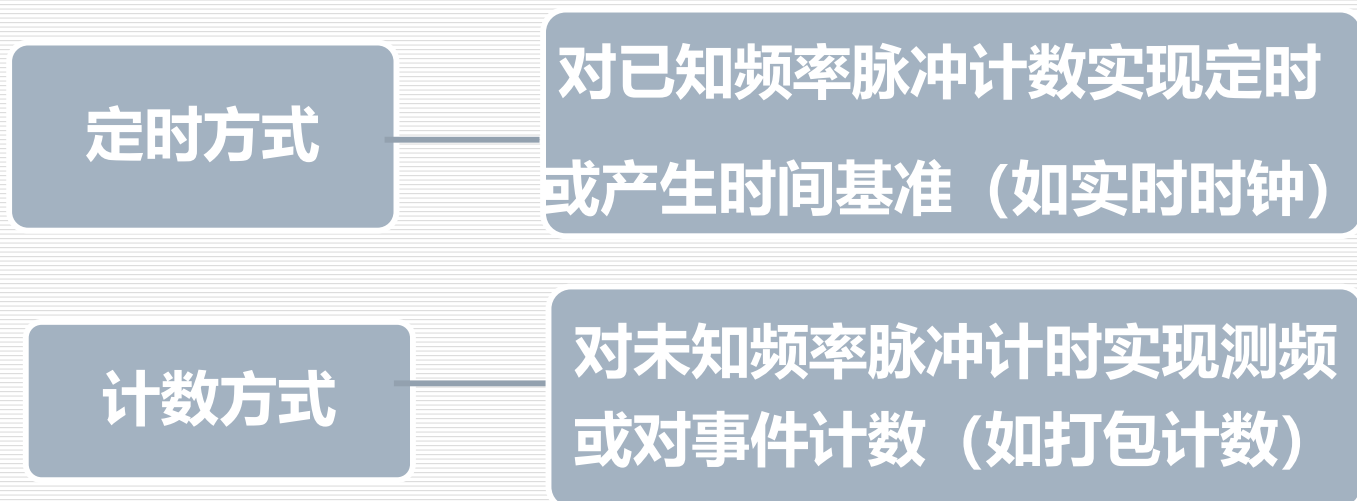
➤ 定时器（模式）：实现硬件的准确定时

- 微机系统的定时检测和控制；
- 测量事件的时间间隔（如脉冲宽度）；
- 为串行端口提供波特率时钟信号。

核心部件：**脉冲计数器**

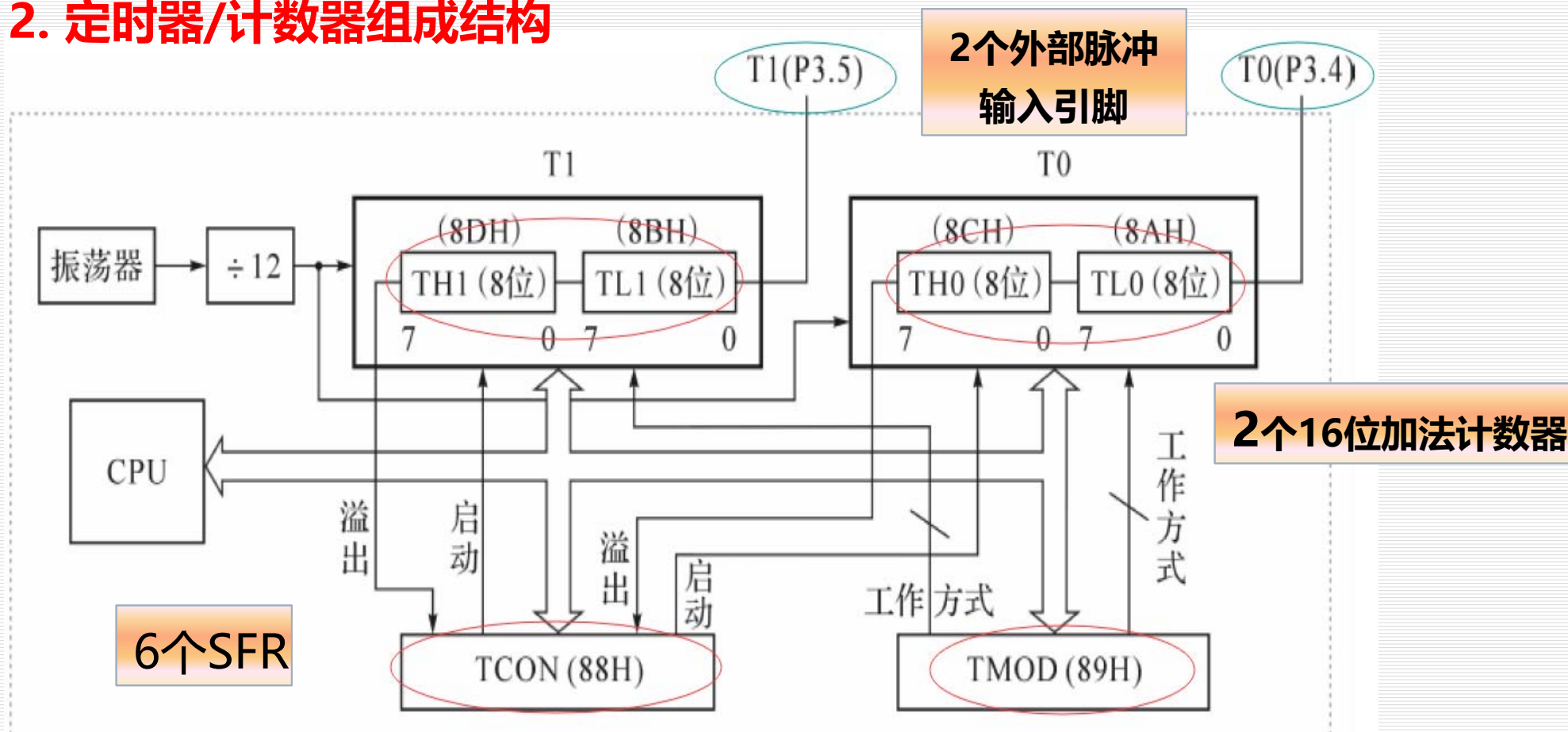
基本原理：对输入脉冲**进行
加减计数**

➤ 计数器（模式）：测定外部脉冲/事件发生的次数



第6章 定时器/计数器

2. 定时器/计数器组成结构

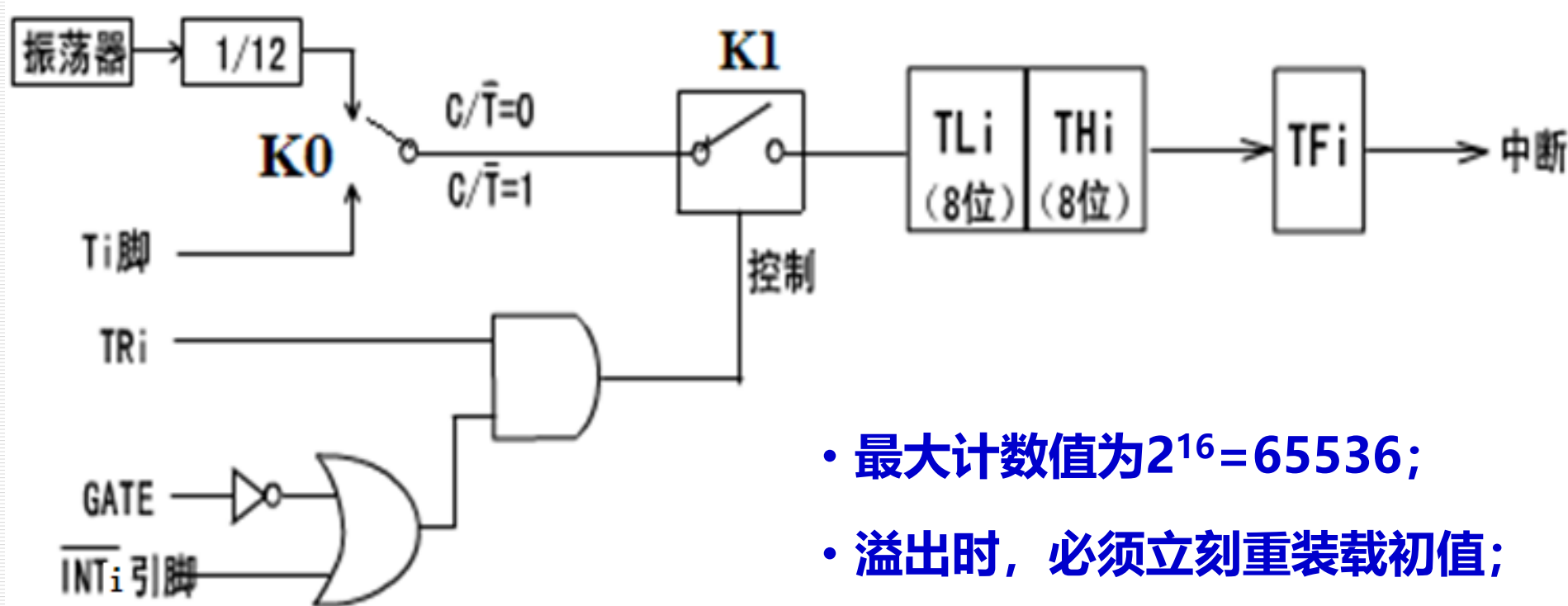


	地址	7	6	5	4	3	2	1	0
TMOD	89H	GATE	C/T	M1	M0	GATE	C/T	M1	M0
TCON	88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

第6章 定时器/计数器

3. 定时器/计数器工作方式

方式1：16位定时/计数方式



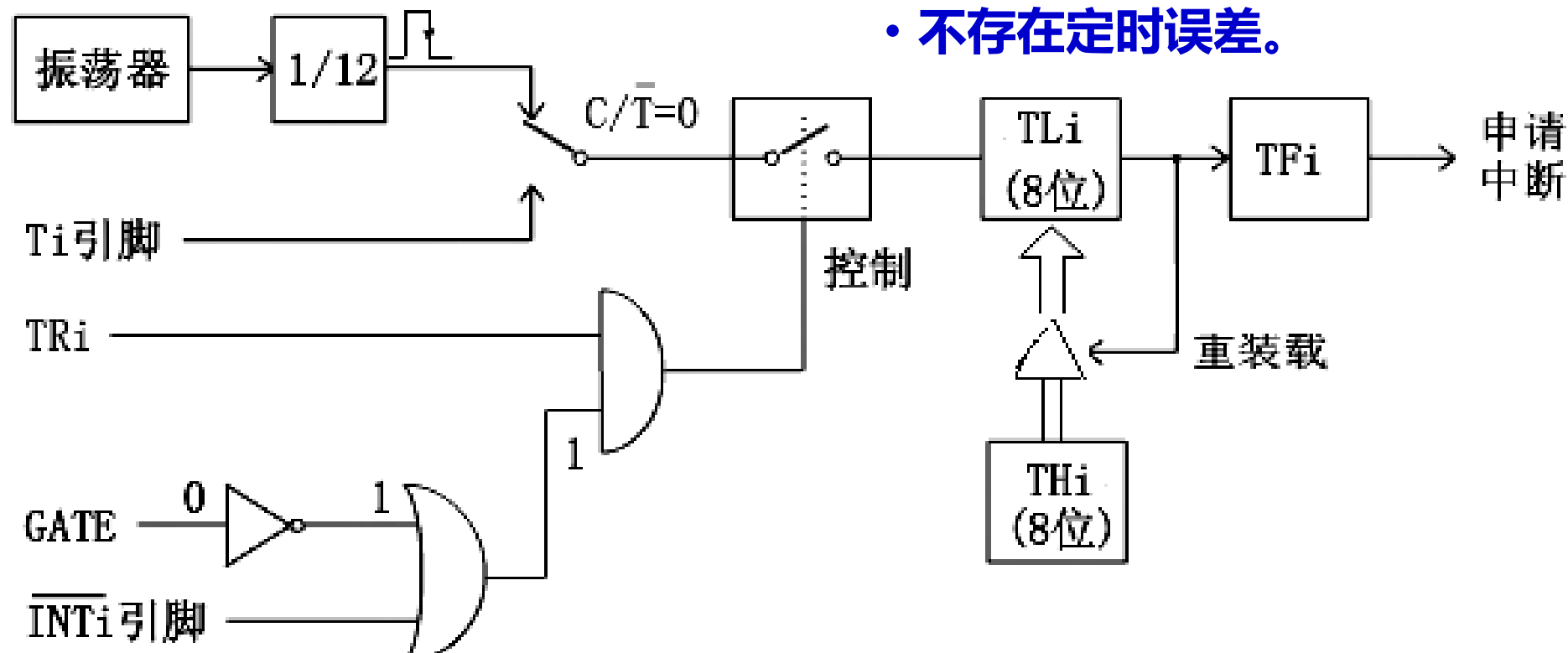
- 最大计数值为 $2^{16}=65536$;
- 溢出时，必须立刻重装载初值;
- 存在一定的定时/计数误差。

第6章 定时器/计数器

3. 定时器/计数器工作方式

方式2：8位初值重载方式

- 最大计数值为 $2^8=256$;
- 溢出时，自动重载初值;
- 不存在定时误差。



第6章 定时器/计数器

4. 定时器/计数器的初始化

1. 初始化步骤

- 确定工作方式，即设置方式寄存器TMOD。
- 预置定时初值/计数初值，并写入TH0、TL0或TH1、TL1。
- 中断设置（给IE赋值），允许或禁止定时器/计数器的中断。
- 启动定时器/计数器，令TCON中的TR0或TR1为“1”。

2. 定时/计数初值的确定

- 取决于需要定时（计数）值相对于工作方式最大计数值的补码
- 方式1最大计数值：65536；方式2最大计数值：256



第6章 定时器/计数器

5. 定时器/计数器的应用

定时方式的应用	<ul style="list-style-type: none">• 波形输出，如输出周期性方波• 产生等时间间隔定时信号，定时执行某功能程序等
计数方式的应用	外部事件统计，如流水线上工件的计数； 外部脉冲计数，当脉冲频率未知时，应采用工作方式1
脉冲宽度测量	结合GATE，测量外部高脉冲的宽度
外部扩展中断	方式2，定时初值为0FFH时，T0或T1引脚的下降沿即产生溢出中断
实时时钟设计	以16位工作方式定时50ms为基准，结合软件计数器，计算得到秒、分、时，实现时钟功能

6. 应用举例

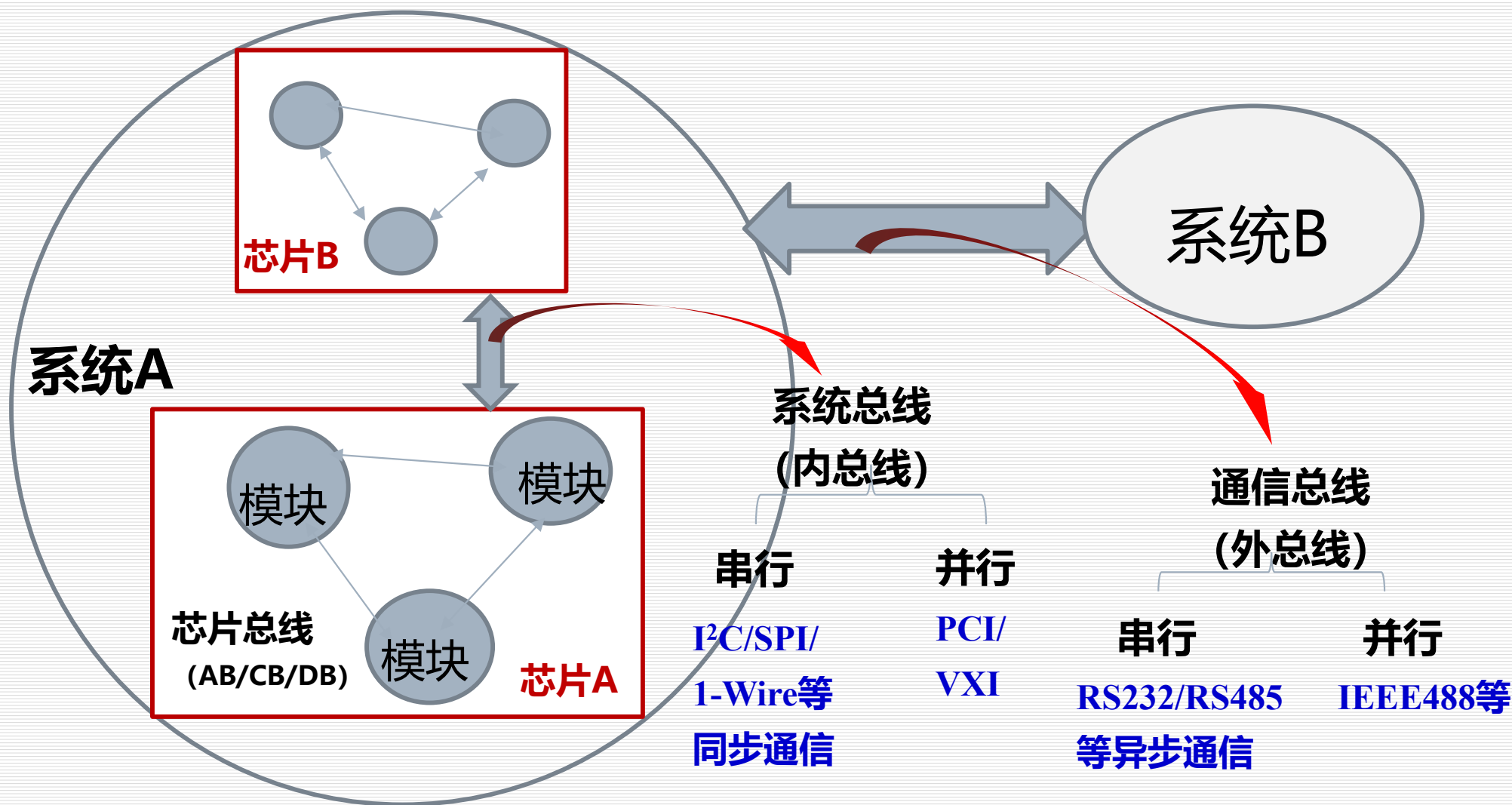
例题6-6、6-7

课后习题：1、3、4、7



第7章 串行总线与通信技术

1. 总线的概念与分类



第7章 串行总线与通信技术

2. 串行通信方式

异步通信：固定的帧格式



通信协议

对数据帧格式、波特率、校验方式、握手方式的约定。

同步通信：相同的时钟信号 校验方式

- 同步字符
- 数据块
- 校验字符

奇偶校验：字符校验；（只能检出奇数个错误）

累加和校验：对数据块；（检错能力有限）

循环冗余校验：对数据块（检错能力强，最常用）

第7章 串行总线与通信技术

例1：用奇校验传送33H和43H。

分析：奇校验即要求通过加上校验位（0或1），使得每帧数据中的“1”的个数为奇数个。

33H: 00110011 其中1的个数为4，所以校验位应为1；

43H: 01000011 其中1的个数为3，所以校验位应为0；

数据帧格式：每帧11位。

" 1	1	00110011	0 "	传输方向→
" 1	0	01000011	0 "	传输方向→
停	校	8位数据位	起	
止	验	(低位在前,	始	
位	位	即低位先发)	位	

第7章 串行总线与通信技术

串行通信的数据通路形式

(1) 单工 (simplex)形式——即单向传送

通信双方中一方固定为发送端，另一方则固定为接受端

只需一条数据线

(2) 全双工 (Full-duplex)形式——即双向传送

有两条数据线，双方均可作为接收端和发送端

(3) 半双工 (Half-duplex)形式——由单线双向

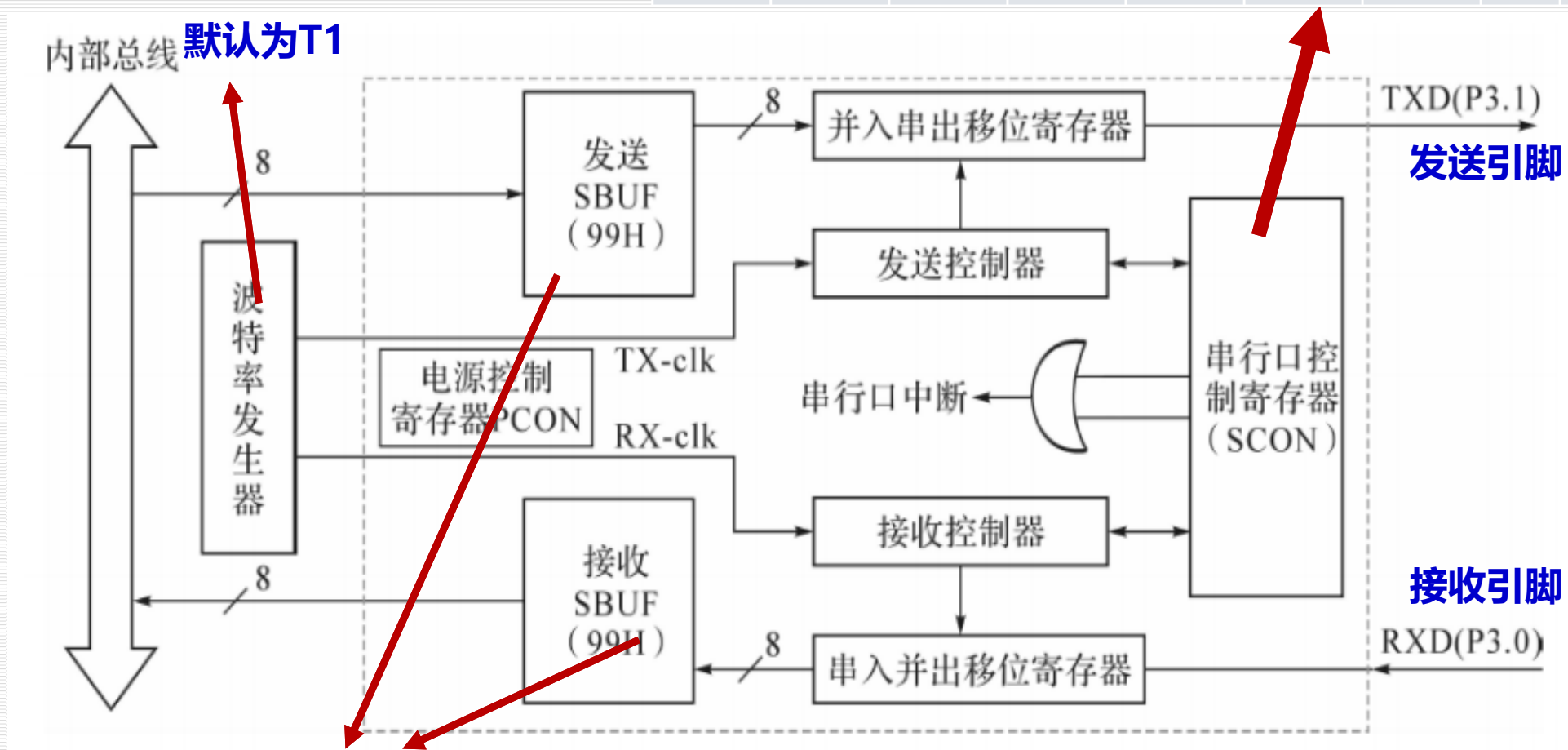
数据也是双向传送，但每次只能是其中一方发送，另一方接收数据



第7章 串行总线与通信技术

3. 8051微控制器的UART

位	7	6	5	4	3	2	1	0
符号	SM0	SM1	SM2	REN	TB8	RB8	TI	RI



物理上独立 (地址相同)

8051微控制器利用串行口发送一帧数据的过程为：(1) 用指令将待发送的数据写入SBUF；(2) UART模块自动从TXD引脚串行发送一帧数据；(3) 硬件自动将SCON的TI置1；(4) 用软件将TI清0

第7章 串行总线与通信技术

3. 8051微控制器的UART

SM0	SM1	工作方式	特点	波特率	备注
0	0	0	8位移位寄存器	$\frac{f_{osc}}{12}$	同步通信：扩展I/O端口 RXD：数据输入输出线DAT TXD：输出同步时钟信号CLK
0	1	1	10位UART	$\frac{2^{SMOD}}{32} \times \frac{f_{osc}}{12 \times (2^8 - X)}$	异步通信方式 1. 启动发送、发送数据： <ul style="list-style-type: none">• 数据写入发送SBUF。• 数据以波特率的速率从TXD逐位移出。 2. 启动接收、接收数据 <ul style="list-style-type: none">• 设置REN=1。• 以16倍波特率的速率检测RXD引脚电平，取中间7、8、9的结果作为该bit的接收值。
1	0	2	11位UART	$\frac{f_{osc}}{64} \text{ or } \frac{f_{osc}}{32}$	
1	1	3	11位UART	$\frac{2^{SMOD}}{32} \times \frac{f_{osc}}{12 \times (2^8 - X)}$	

第7章 串行总线与通信技术

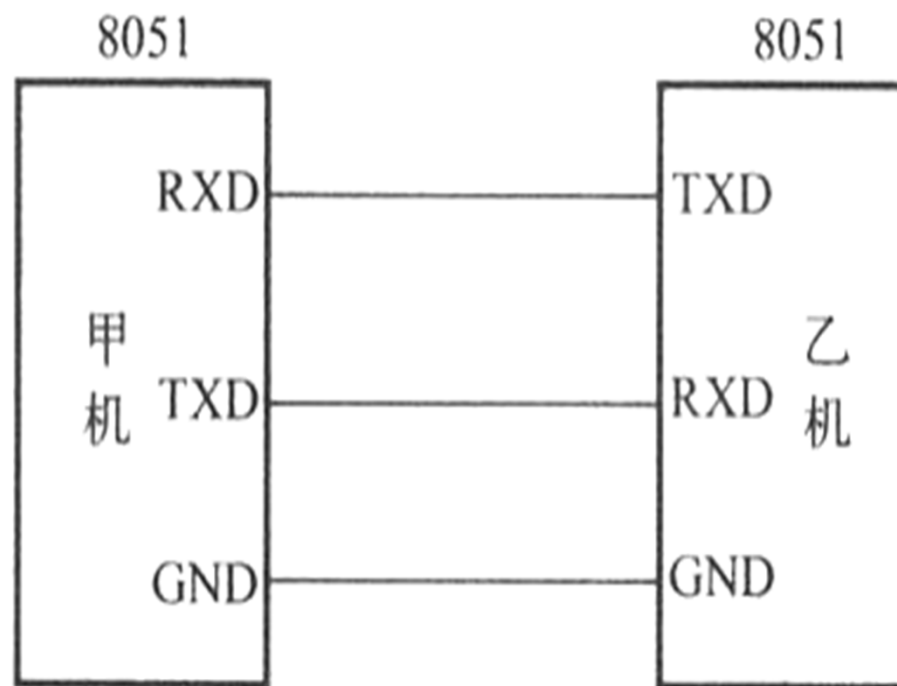
4. UART的应用

方式0：同步通信方式，用于扩展I/O接口；

- 连接并入串出移位寄存器如74HC165，扩展输入接口；
- 连接串入并出移位寄存器如74HC164，扩展输出接口。

方式1-3：异步通信方式，用于系统间的通信

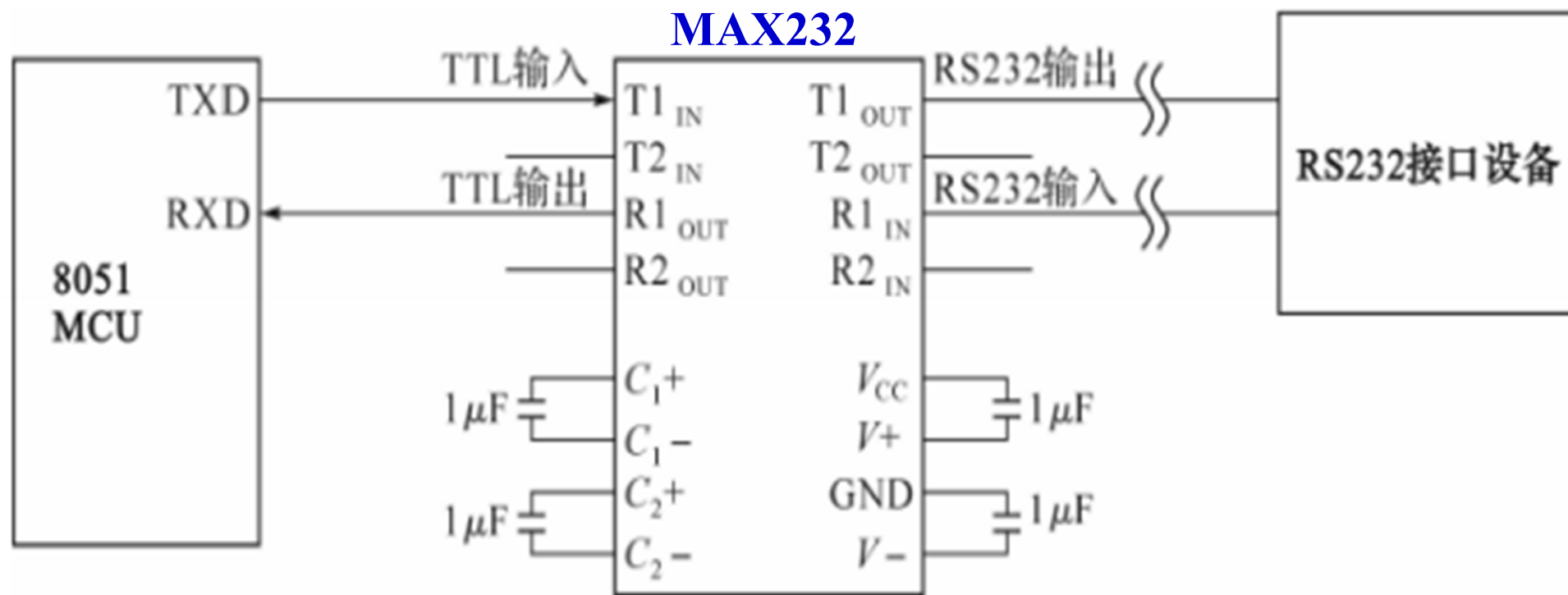
要求：波特率相同、数据帧格式相同



第7章 串行总线与通信技术

5. 串行通信技术

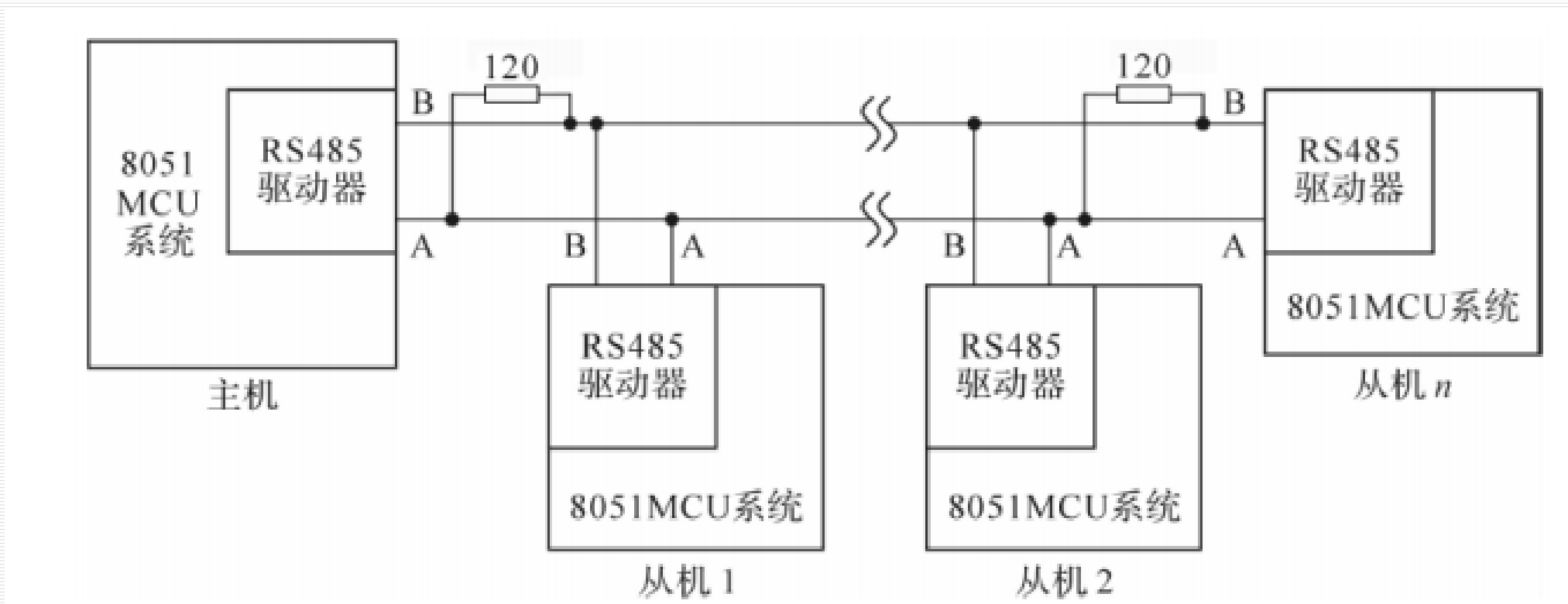
RS232通信：采用负逻辑电平；**MCU+电平转换芯片(如MAX232)：**实现TTL或CMOS电平到负逻辑电平的转换。



第7章 串行总线与通信技术

5. 串行通信技术

RS485通信：差分信号，（MCU+RS485驱动器）实现数字信号到差分信号的转换。



课后习题：1、4、9、10

第8章 人机接口技术

1. 键盘基础知识

(1) 键盘的三种工作方式

工作方式	工作原理	优缺点
编程扫描方式（查询方式）	CPU完成其他工作的空余时间扫描键盘。	CPU运行其他程序时不会响应按键操作。
定时扫描方式	通过定时器产生定时中断，CPU每隔T对键盘扫描一次， $T \leq 50\text{ms}$ 。	CPU常常处于空扫描状态，浪费资源。
中断工作方式	有键按下时产生外部中断请求，CPU响应中断后对键盘扫描。	不会空扫描，且每次响应迅速。

设计选择原则：及时响应 & 不能占用CPU过多时间

第8章 人机接口技术

1. 键盘基础知识

(2) 抖动、连击与重键

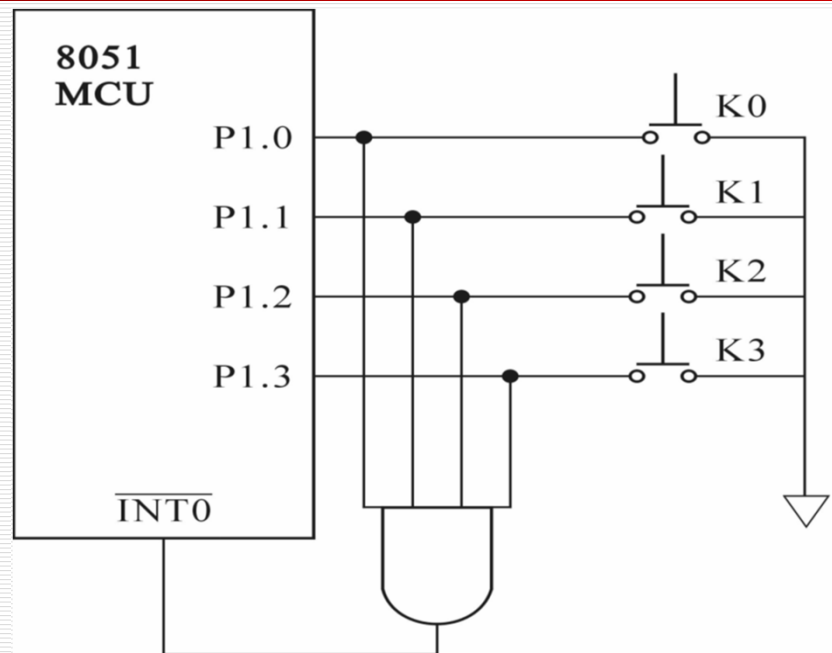
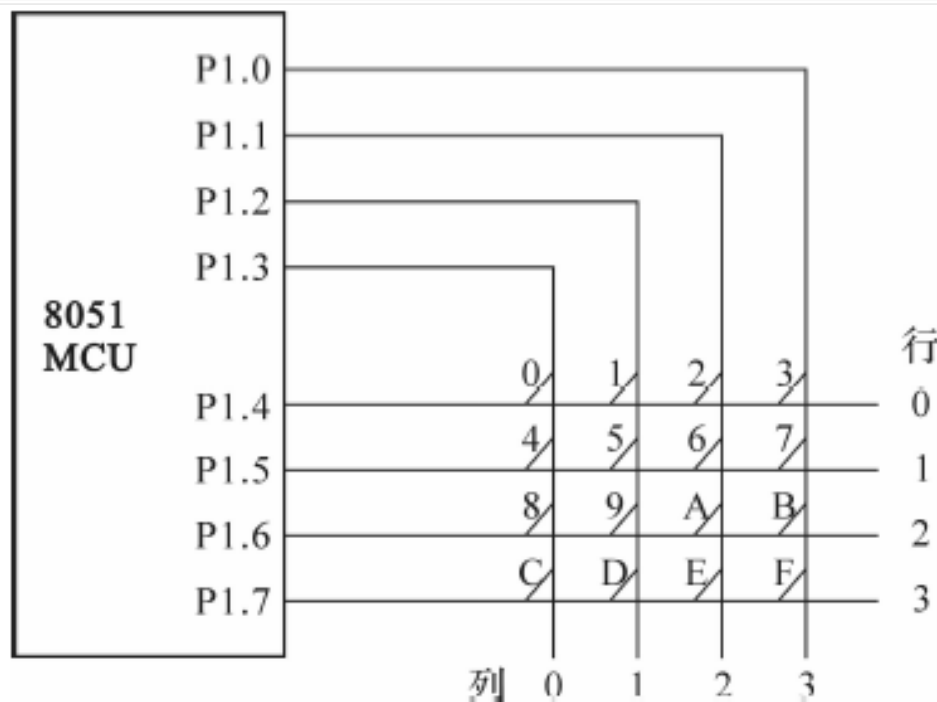
	产生原因	解决方法	
抖动	按键闭合和断开时不是立刻稳定地接通与断开	软件延时法： 检测到有键按下时，延时10ms去前沿抖动，再检测该键状态，状态不同则为干扰。有键松开时同理。	
连击	一次按键操作，多次响应	消除	执行程序后不断查询该键是否释放。
		利用	执行程序后并延时后重新读键。
重键	两个或多个键同时闭合	N键锁定	只把最后释放的键当作有效键。
		N键轮回	对所有按下的键依次产生键值。

第8章 人机接口技术

2. 非编码式键盘与接口

(1) 独立式键盘

- 每个案件占用一根I/O口线
- 优点：结构简单，按键识别容易
- 缺点：I/O数量有限，适用于按键较少的系统



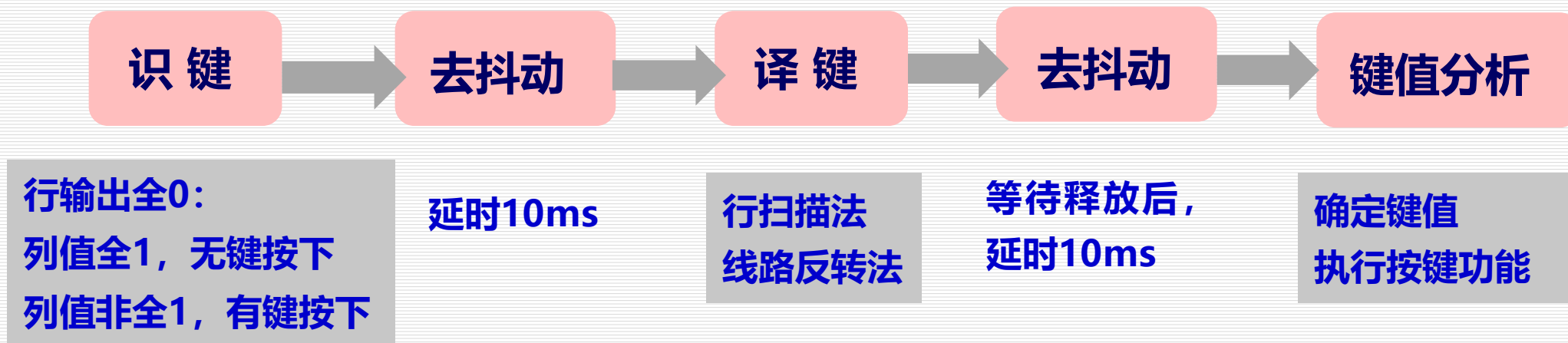
(2) 矩阵式键盘

- $m \times n$ 矩阵键盘只需要 $m+n$ 条口线
- 优点：节省I/O口线数量
- 缺点：扫描程序较为复杂

第8章 人机接口技术

2. 非编码式键盘与接口

(3) 矩阵式键盘扫描方法



➤ 行扫描法

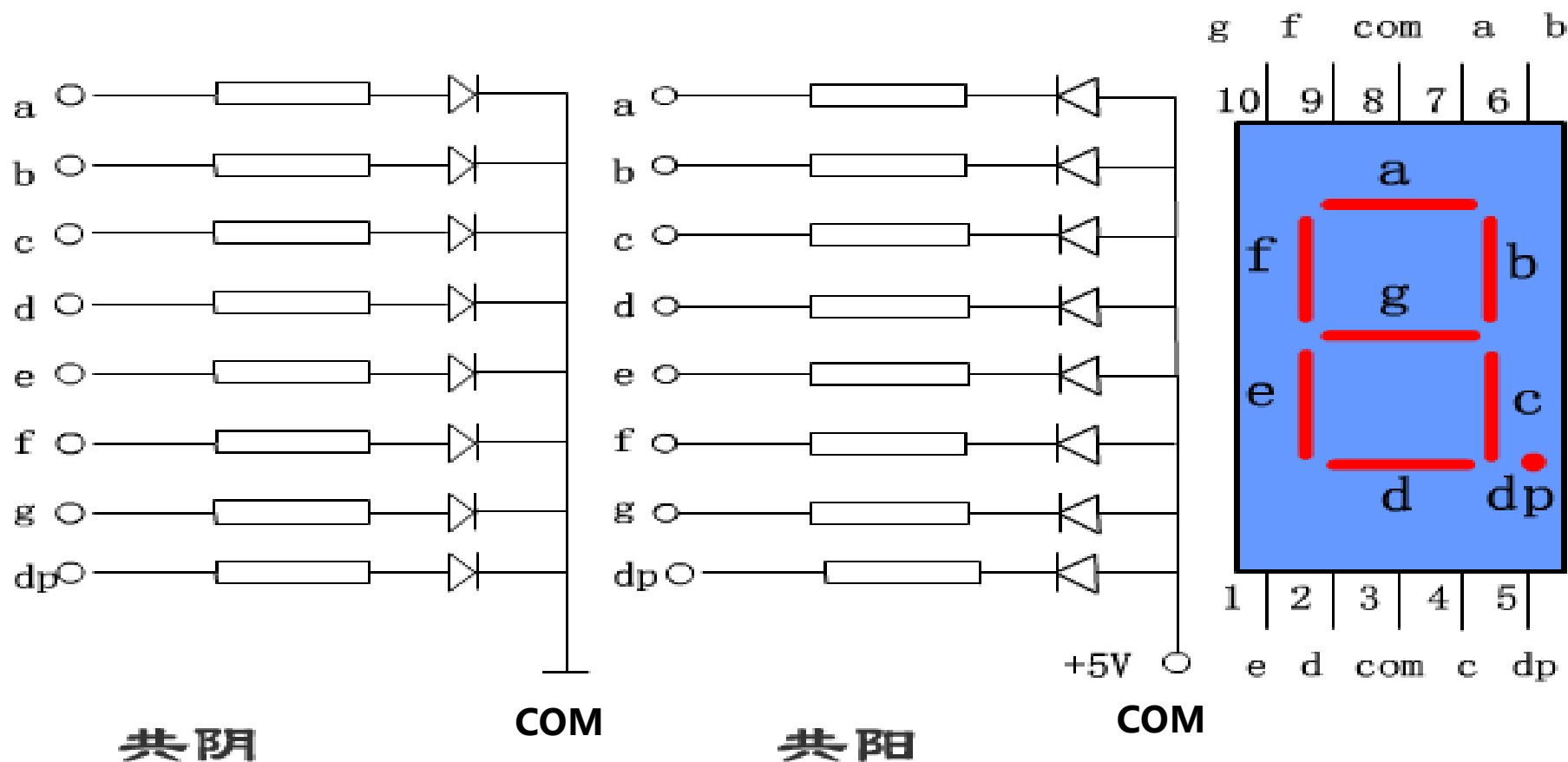
- **粗扫描:** 判别是否有键按下
- **细扫描:** 逐行扫描; 识别哪个键按下

➤ 线路反转法

- **第1步:** 行输出、列输入, 得到列值;
- **第2步:** 线路反转, 即列输出、行输入, 得到行状态;
- **第3步:** 形成特征码。

第8章 人机接口技术

3. 段码式LED（数码管）与显示接口



共阴数码管： COM端灌入电流（应接地或具有较大灌电流的口线），**段码端（阳极）输出电流。**

共阳数码管： COM端输出电流（应接电源或具有较大输出电流的口线），**段码端（阴极）灌入电流。**

第8章 人机接口技术

3. 段码式LED（数码管）与显示接口

静态接口、动态接口

	优点	缺点
静态显示	程序简单，显示稳定可靠。	位数较多时所需输出口较多； 串行扩展时芯片较多，更新速度慢。
动态显示	占用输出接口少。	需要定时对各个数码管进行显示扫描，占用CPU时间资源。位数过多时亮度较低。

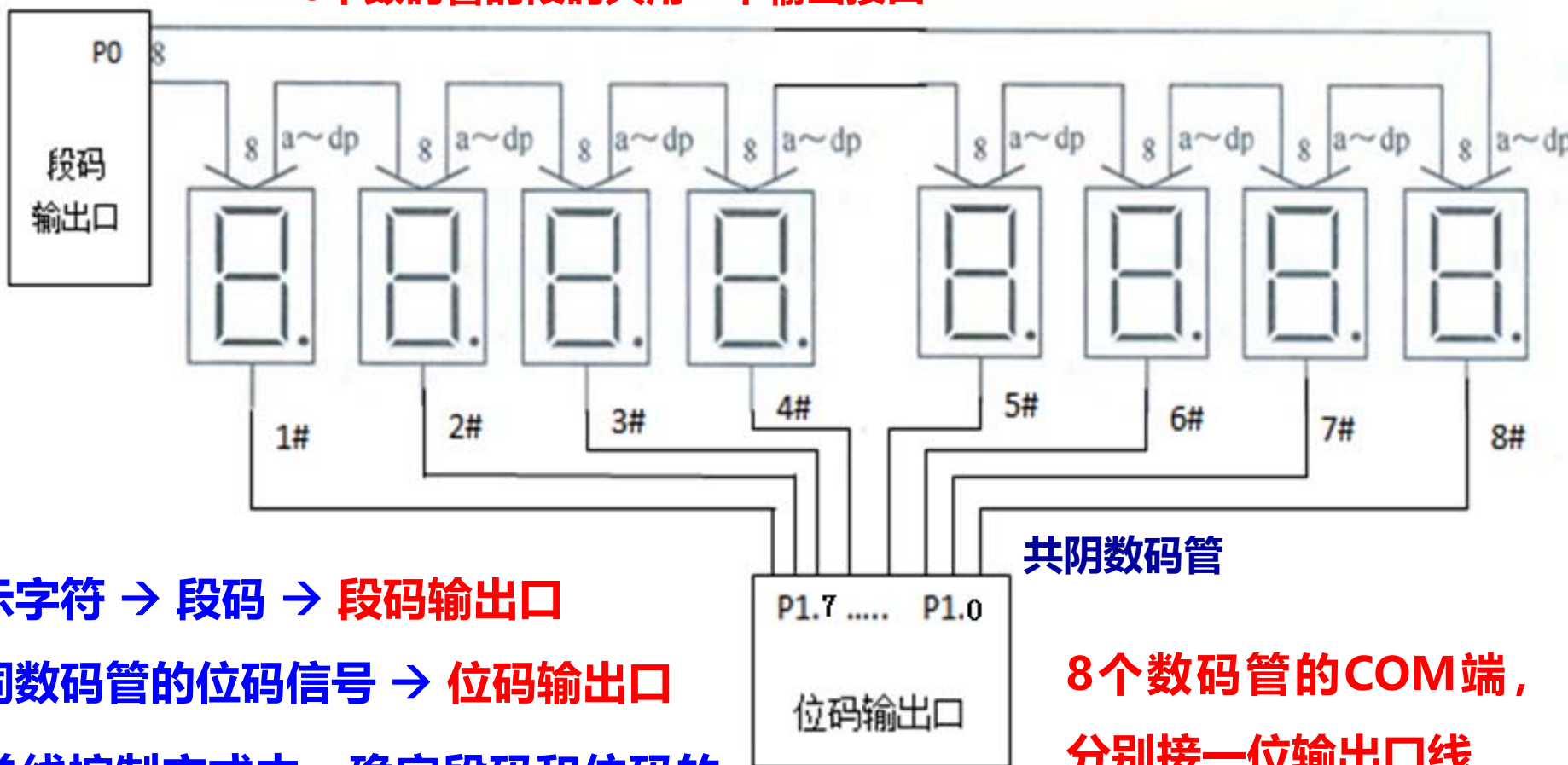


第8章 人机接口技术

3. 段码式LED（数码管）与显示接口

动态接口

8个数码管的段码共用一个输出接口



共阴数码管

显示字符 → 段码 → 段码输出口

不同数码管的位码信号 → 位码输出口

在总线控制方式中，确定段码和位码的映射地址

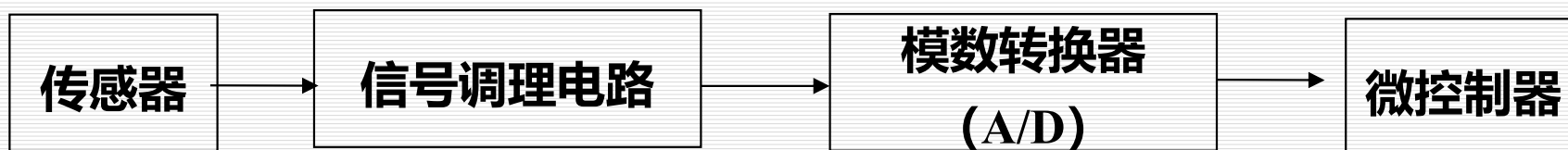
8个数码管的COM端，
分别接一位输出口线

4. 实验三：LED数码管显示实验

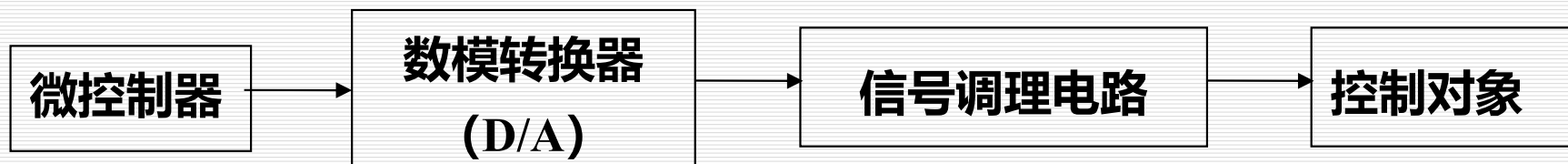
第9章 模拟接口技术

1. 模拟输入输出通道结构

(1) 模拟输入通道结构:



(2) 模拟输出通道结构:



第9章 模拟接口技术

2. ADC、DAC特性与指标

(1) A/D转换器

A/D转换器（ADC）是将**连续模拟信号量化**并编制成有限位**数字信号**的集成电路。

转换原理：双积分式、逐位逼近式、 $\Sigma-\Delta$ （和差）调制式、并行直接比较式

主要指标：

- **分辨率：**仅与器件的**位数**有关，增加转换器的位数是减少量化误差的**唯一**方法。
- **量化误差：**由**分辨率 Δ** 决定，数值是 $\pm 1/2\text{LSB}$ （最低有效位，即最小分辨电压）
- **转换精度：**ADC的实际输出结果与理论转换结果的偏差；与ADC器件的**材料、制作工艺**有关。

第9章 模拟接口技术

2. ADC、DAC特性与指标

(2) D/A转换器

D/A转换器 (DAC, Digital to Analog Converter) 是一种将数字信号转换成模拟信号的器件。

主要指标:

- **分辨率**: DAC输入数字量发生1bit变化引起的模拟量输出的变化, 用DAC的二进制位数表示。DAC的位数越多, 其分辨率越高。
- **转换精度**: DAC实际输出电压与理论转换电压之间的偏差。与DAC器件的材料、制作工艺有关。

课后习题: 1、3





考试时间：2024年1月18日

考试地点：6教中101

祝大家考试顺利

谢谢！

There's always more to learn, and there are always better ways to do what you've done before.

—DONALD E. KNUTH