



# 微机原理和接口技术

## 第三讲 微机的硬件结构2

---

# 提 纲

## 0. 基础知识

### 1. 8051微控制器的结构

### 2. 微控制器的工作原理

### 3. 存储器配置与地址空间

### 4. 特殊功能寄存器SFR (1)

### 5. 特殊功能寄存器SFR (2)

## 6. I/O端口结构与应用特点

### 7. 时钟与复位

### 8. 微控制器的工作方式

## 9. 8051微控制器的技术发展

# 提 纲

## 3. 存储器配置与地址空间



# 存储器配置与地址空间——存储器配置

## 1. 存储器的两种基本结构形式

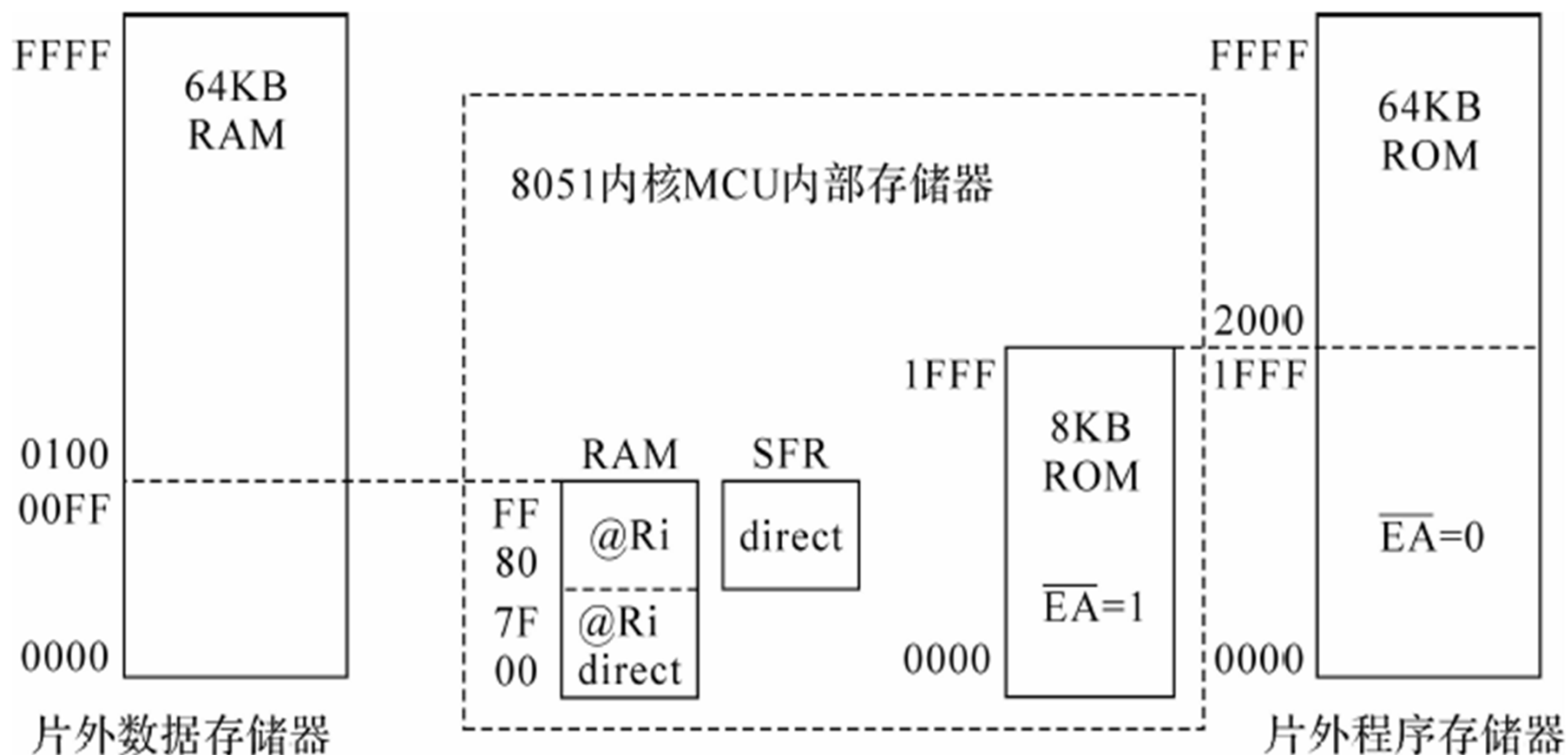
不同微控制器中存储器的用途是相同的，但结构与存储容量却不完全相同。微控制器中的存储器有两种基本结构形式：

- **冯·诺依曼（Von Neumann）结构：**也称普林斯顿（Princeton）结构。程序存储器和数据存储器共用一个逻辑空间，且它们是统一编址的。如16条地址线的寻址空间是64K，则ROM和RAM总共只有64K。（在通用微型计算机中广泛采用）
- **哈佛（Harvard）结构：**是程序存储器和数据存储器分别编址的结构。如16条地址线，可以分别寻址64K的ROM和64K的RAM。

# 存储器配置与地址空间——存储器配置

## 2. 8051存储器结构图

8051微控制器的存储器采用哈佛结构，ROM和RAM是分





# 存储器配置与地址空间——存储器配置

## 1. ROM空间配置

ROM的主要功能是存放程序和数据表格，以及掉电后不希望丢失的信息。在经典8051微控制器中，程序存储器可以分为内部和外部两部分：

- 内部8K空间： 0000H~1FFFH
- 外部64K空间： 0000H~FFFFH

注意：

- 内外部ROM是统一编址的，ROM总容量为64KB。所以内部8KB ROM和外部8KB ROM只能选用其一。

目前增强型的8051MCU，集成了16-64K的内部ROM，所以已不需外部扩展。



# 存储器配置与地址空间——存储器配置

## 2. ROM中的6个特殊单元

8051 MCU的ROM中有6个特殊单元，是设置的特定程序入口地址，1个复位入口和5个中断入口。所谓入口，是指一旦满足条件，PC的值自动变为这些入口地址，则CPU将自动转向这些ROM地址取指令执行程序。

名称	入口地址	意义
复位	0000H	系统复位后PC = 0000H
外部中断0	0003H	外部中断0响应时程序转向0003H
定时器T0溢出	000BH	T0中断响应时程序转向000BH
外部中断1	0013H	外部中断1响应时程序转向0013H
定时器T1溢出	001BH	T1中断响应时程序转向001BH
串行口中断	0023H	串行口中断响应时程序转向0023H



# 存储器配置与地址空间——存储器配置

## 1. 数据存储器RAM空间配置

RAM一般用于存放实时采集的数据、计算的中间结果、控制参数、需要传送和显示的数据等。

在8051微控制器中，RAM有内部和外部两个空间，内部具有256B的通用RAM，对于具有并行外扩功能的MCU，同时可以扩展64KB外部RAM。

- 内部RAM: 00H~FFH (256B)
- 外部RAM: 0000H~FFFFH (64KB)



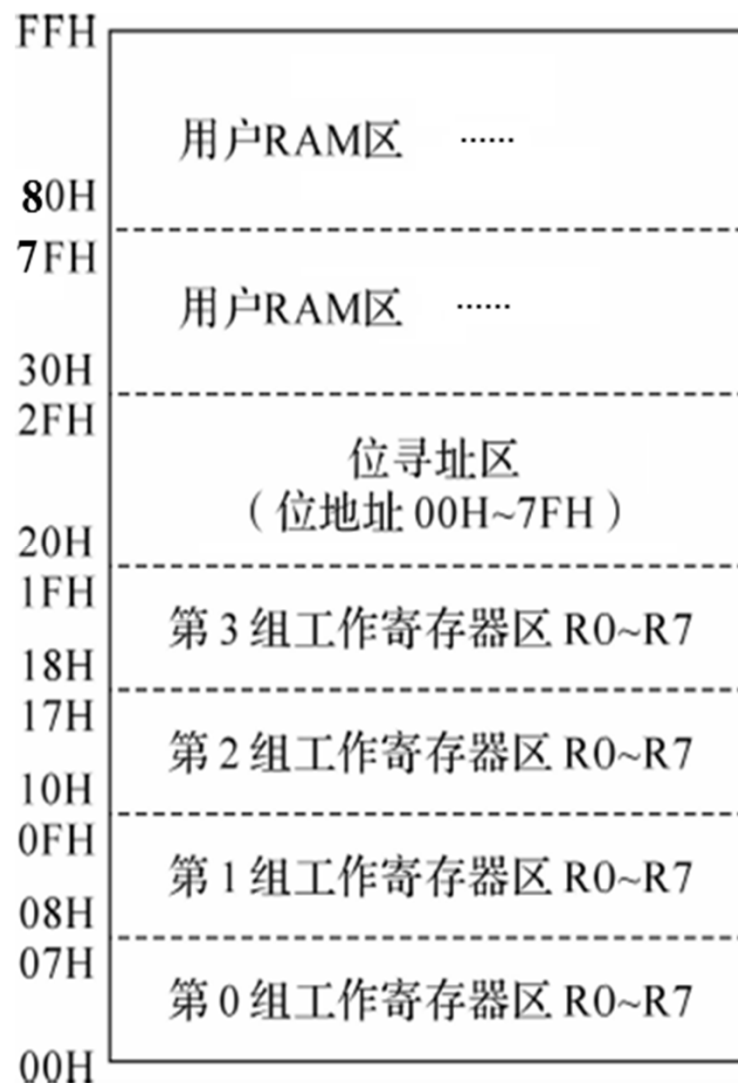
# 存储器配置与地址空间——存储器配置

## 2. 内部RAM配置

内部RAM中，低128B（00H-7FH）是基本数据存储器，可采用直接寻址、寄存器间接寻址、位寻址等多种寻址方式；高128B（80H-FFH）是扩展数据存储器，只能采用寄存器间接寻址方式。

内部RAM可划分为三块空间：

- （1）工作寄存器区
- （2）位寻址区
- （3）用户RAM区（包括堆栈）





## 存储器配置与地址空间——存储器配置

### (1) 工作寄存器区

工作寄存器区位于内部RAM的00H~1FH单元，共32字节，分成四组。每组8个字节，分别记作R0~R7。

- 工作寄存器0组，地址为00H~07H 对应：R0~R7
- 工作寄存器1组，地址为08H~0FH 对应：R0~R7
- 工作寄存器2组，地址为10H~17H 对应：R0~R7
- 工作寄存器3组，地址为18H~1FH 对应：R0~R7

(任一时刻，只能使用一个寄存器组；CPU复位后，默认选择第0组)

工作寄存器区是寄存器寻址区域，对该区域操作的指令数量最多，均为单周期指令，执行的速度最快。



## 存储器配置与地址空间——存储器配置

### (1) 工作寄存器区

选择四组工作寄存器区的哪一组作为R0~R7，由PSW寄存器中的两位RS1、RS0来确定。

RS1、RS0称为工作寄存器组选择位，两位确定4种选择。

不同组别工作寄存器对应的存储器地址

组	RS1	RS0	R0	R1	R2	R3	R4	R5	R6	R7
0	0	0	00H	01H	02H	03H	04H	05H	06H	07H
1	0	1	08H	09H	0AH	0BH	0CH	0DH	0EH	0FH
2	1	0	10H	11H	12H	13H	14H	15H	16H	17H
3	1	1	18H	19H	1AH	1BH	1CH	1DH	1EH	1FH



## 存储器配置与地址空间——存储器配置

### (1) 工作寄存器区

例如： 助记符          机器码  
指令    INC R7          0F

该指令的功能是将R7的内容增1，此时的R7对应于RAM的哪个单元？与此时的寄存器组有关。

RS1 RS0	寄存器组	R7地址	指令执行前	执行后
0 0	选用0组	R7为07H单元	(07H)=05H	(07H)= 06H
1 1	选用3组	R7为1FH单元	(1FH)=02H	(1FH)= 03H

# 存储器配置与地址空间——存储器配置

## (2) 位寻址区

内部RAM中的 20H~2FH，共16个单元是位寻址区。共有128位，位地址为00H~7FH。

可位寻址的16B内部RAM，既可进行字节寻址，又可进行位寻址。

可以使用位操作指令，如CLR，SETB等。

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									



## 存储器配置与地址空间——存储器配置

### (3) 用户RAM区

**用户RAM：**内部RAM的 30H~FFH，以及没有使用的工作寄存器区和位寻址区。通常作为数据缓冲区和堆栈区。

**数据缓冲区：**用来存放各种用户数据，如AD转换结果、键盘扫描码、参数设定值、数据处理结果、显示或通信缓冲区等等。

**堆栈区：**堆栈是一种具有特殊用途的存储区域，其作用是用于暂存数据和地址；在子程序和中断服务程序中，用于保护断点和保护现场。

- 8051微控制器的堆栈区必须开辟在内部通用RAM中。



# 存储器配置与地址空间——存储器配置

## 3. 外部数据存储器

- 外部RAM最多可以扩展64KB，地址范围为0000H~FFFFH。
- CPU访问内部RAM和外部RAM的指令不同，内部RAM（内存）的访问指令多、速度快，可采用直接寻址和寄存器间接寻址等方式。外部RAM访问指令少，速度也较慢，且只有一种寄存器间接寻址方式。

对于内部RAM 256B与外部RAM低256B的地址重叠问题，解决办法是：访问内部、外部RAM，采用的指令不同。

# 提 纲

## 4. 特殊功能寄存器SFR (1)





# 特殊功能寄存器SFR(1)

## 1. SFR简介

特殊功能寄存器SFR (Special Function Register)，也称专用寄存器。

用于管理和控制MCU内部硬件功能模块（如定时器/计数器、串行口、中断系统等）的寄存器，用来存放功能模块的控制命令、状态或数据。

### **8051微控制器的SFR:**

- 21个SFR，离散分布于80H~FFH的专用寄存器区，未定义的访问无效。
- 除程序计数器PC指针和R0~R7工作寄存器外，其余所有定义的寄存器都属SFR。
- 有些SFR可以位寻址，能位寻址的单元一定能字节寻址。



# 特殊功能寄存器SFR(1)

## 2.SFR定义与分布

序 号	符 号	地 址	名称和作用		位寻址
1	B	F0H	称为 B 的一个寄存器(乘除指令中用)		✓
2	A	E0H	Accumulator	累加器	✓
3	PSW	D0H	Program Status Word	程序状态字	✓
4	IP	B8H	Interrupt Priority	中断优先级控制寄存器	✓
5	P3	B0H	Port 3	并行口 P3	✓
6	IE	A8H	Interrupt Enable	中断允许控制寄存器	✓
7	P2	A0H	Port 2	并行口 P2	✓
8	SBUF	99H	Serial Data Buffer	串行口数据寄存器	
9	SCON	98H	Serial Control	串行口控制寄存器	✓
10	P1	90H	Port 1	并行口 P1	✓
11	TH1	8DH	Timer 1 High Byte	定时器 1 高 8 位	
12	TH0	8CH	Timer 0 High Byte	定时器 0 高 8 位	
13	TL1	8BH	Timer 1 Low Byte	定时器 1 低 8 位	
14	TL0	8AH	Timer 0 Low Byte	定时器 0 低 8 位	
15	TMOD	89H	Timer Mode	定时器/计数器方式寄存器	
16	TCON	88H	Timer Control	定时器/计数器控制寄存器	✓
17	PCON	87H	Power Control	电源控制寄存器	
18	DPH	83H	Data Pointer High Byte	数据指针 DPTR 高 8 位	
19	DPL	82H	Data Pointer Low Byte	数据指针 DPTR 低 8 位	
20	SP	81H	Stack Pointer	堆栈指针	
21	P0	80H	Port 0	并行口 P0	✓



# 特殊功能寄存器SFR(1)

## 3. SFR的位寻址空间

字节地址的低位为0H或8H的SFR，是可位寻址的SFR。定义了83位。

通用RAM中的位寻址区和SFR中的位寻址区，构成了8051微控制器的位寻址空间。

符 号	寄存器名	位符号地址和物理地址								字节地址
		D7	D6	D5	D4	D3	D2	D1	D0	
B	B 寄存器	F7H	F6H	F5H	F4H	F3H	F2H	F1H	F0H	F0H
		B. 7	B. 6	B. 5	B. 4	B. 3	B. 2	B. 1	B. 0	
ACC	累加器	E7H	E6H	E5H	E4H	E3H	E2H	E1H	E0H	E0H
		ACC. 7	ACC. 6	ACC. 5	ACC. 4	ACC. 3	ACC. 2	ACC. 1	ACC. 0	
PSW	程序状态字	D7H	D6H	D5H	D4H	D3H	D2H	D1H	D0H	D0H
		Cy	AC	F0	RS1	RS0	OV	F1	P	
IP	中断优先级寄存器	BFH	BEH	BDH	BDH	BBH	BAH	B9H	B8H	B8H
		—	—	—	PS	PT1	PX1	PT0	PX0	
P3	P3 口	B7H	B6H	B5H	B4H	B3H	B2H	B1H	B0H	B0H
		P3. 7	P3. 6	P3. 5	P3. 4	P3. 3	P3. 2	P3. 1	P3. 0	
IE	中断允许寄存器	AFH	AEH	ADH	ACH	ABH	AAH	A9H	A8H	A8H
		EA	—	—	ES	ET1	EX1	ET0	EX0	
P2	P2 口	A7H	A6H	A5H	A4H	A3H	A2H	A1H	A0H	A0H
		P2. 7	P2. 6	P2. 5	P2. 4	P2. 3	P2. 2	P2. 1	P2. 0	
SCON	串行口控制寄存器	9FH	9EH	9EH	9CH	9BH	9AH	99H	98H	98H
		SM0	SM1	SM2	REN	TB8	RB8	TI	RI	
P1	P1 口	97H	96H	95H	94H	93H	92H	91H	90H	90H
		P1. 7	P1. 6	P1. 5	P1. 4	P1. 3	P1. 2	P1. 1	P1. 0	
TCON	定时器控制寄存器	8FH	8EH	8DH	8CH	8BH	8AH	89H	88H	88H
		TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	
P0	P0 口	87H	86H	85H	84H	83H	82H	81H	80H	80H
		P0. 7	P0. 6	P0. 5	P0. 4	P0. 3	P0. 2	P0. 1	P0. 0	



## 特殊功能寄存器SFR(1)

### 4. 程序计数器PC

也称为程序指针或PC指针，具有如下特点：

- PC是一个16位的专用寄存器，存放ROM的地址，因此成为程序指针，其寻址范围为0~64KB。
- PC存放的是下一条要执行的指令地址。复位后PC的内容为0000H，表示CPU将从ROM的0000H单元取指令执行。也即PC指向哪里，CPU就从其指向的ROM单元取指令执行程序。
- PC不属于特殊功能寄存器。因此不占用SFR地址空间，是不可寻址的，在程序中不能直接访问。
- PC可以通过LJMP、SJMP等转移指令来间接修改PC的值。



# 特殊功能寄存器SFR(1)

## 5. 特殊功能寄存器介绍

### (1) 累加器A

累加器A（或ACC）是CPU中使用最频繁8位专用寄存器。在算术、逻辑类操作时，ALU的一个输入来自A，运算结果也大多保存于A。

A的字节地址是E0H，可位寻址。

位地址	E7	E6	E5	E4	E3	E2	E1	E0
位符号	Acc.7	Acc.6	Acc.5	Acc.4	Acc.3	Acc.2	Acc.1	Acc.0



## 特殊功能寄存器SFR(1)

### 5. 特殊功能寄存器介绍

#### (2) B寄存器

B寄存器是一个8位寄存器，一般用于乘除指令中：

MUL AB ;  $A*B=BA$ ;

DIV AB ;  $A/B=\text{商}A\ldots\ldots\text{余数}B$

**B的字节地址是F0H；可位寻址，位地址为 F0H~F7H。**

在其它情况下，B寄存器可以作为内部RAM中的一个单元来使用。

## 特殊功能寄存器SFR(1)

### 5. 特殊功能寄存器介绍

#### (3) 程序状态字PSW (Program Status Word)

PSW 用来存放程序状态信息，表征指令执行后的状态，供程序查询和判别之用。字节地址：D0H；位地址为：D0H-D7H

位地址	D7	D6	D5	D4	D3	D2	D1	D0
位符号	Cy	AC	F0	RS1	RS0	OV	F1	P
注释	Carry	Assistant Carry	Flag 0	Register bank Selector bit 1	Register bank Selector bit0	Overflow	Flag 1	Parity Flag

对于**C**、**AC**、**OV**、**P**，根据指令执行结果，由硬件置位或清0，称为状态位。对于**RS1**、**RS0**、**F1**、**F0**，根据需要使用，由指令设定，称为控制位。

**RS1**、**RS0**为工作寄存器组选择位，**F1**、**F0**由用户自定义使用。



## 特殊功能寄存器SFR(1)

### (3) 程序状态字PSW (C、AC、F0、F1、RS1、RS0、OV、P)

- **C (CY)** : 进位标志。在加、减法运算时, 若高位 (D7) 发生进位或借位则被置1 (即 $C=1$ ), 否则被清0 ( $C=0$ )。
- **AC**: 辅助进位标志。在加、减法运算时, 若低4位向高4位发生进位或借位则 $AC=1$ , 否则 $AC=0$ ; AC标志在十进制调整指令DAA中要用到。
- **F0、F1**: 软件标志。由软件置位或复位, 由用户定义使用。
- **RS1、RS0**: 工作寄存器组选择位。由软件置位或复位, 用来选择4组工作寄存器中的一组。





## 特殊功能寄存器SFR(1)

(3) 程序状态字PSW (C、AC、F0、F1、RS1、RS0、OV、P)

举例 (关于C、AC) :

- $86H + 68H = EEH$

$$\begin{array}{r} 10000110 \\ + 01101000 \\ \hline \end{array}$$

11101110      无进位, 无半进位, 所以,  $C=0$ ,  $AC=0$

- $9AH + 8DH = 127H$

$$\begin{array}{r} 10011010 \\ + 10001101 \\ \hline \end{array}$$

1,00100111

A中结果为27H, 有进位和半进位, 所以 $C=1$ ,  $AC=1$



## 特殊功能寄存器SFR(1)

### (3) 程序状态字PSW (C、AC、F0、F1、RS1、RS0、OV、P)

➤ **OV**: 溢出标志。对于带符号数而言, 反映运算结果是否溢出。

- **OV=1**: 溢出, 表示运算结果超出了A所能表示的带符号数的范围  
(-128~+127, 即8位带符号数的范围)。
- **OV=0**: 没有溢出。
- 对于乘法MUL, 当A、B两个乘数的积超过255时OV置位; 否则, OV=0。
- 对于除法DIV, 若除数为0时, OV=1; 否则, OV=0。



## 特殊功能寄存器SFR(1)

### (3) 程序状态字PSW (C、AC、F0、F1、RS1、RS0、OV、P)

判断OV的2种方法:

当加法或减法的运算结果超出-128 ~ +127这个范围时, OV=1,  
否则OV=0。

以下2种情况表示发生了溢出:

- 两个正数相加, 结果变成负数;
- 两个负数相加, 结果变成正数。



## 特殊功能寄存器SFR(1)

### (3) 程序状态字PSW (C、AC、F0、F1、RS1、RS0、OV、P)

**例1:** 两个正数相加超过了+127, 产生了溢出, A中的和变成了负数, 表示产生了溢出, 结果是错误的, 所以OV=1。

**例2:** 两个负数求加, 结果小于-128, A中的和变成了正数, 表示产生了溢出, 结果是错误的, 所以OV=1。

**例1**

$$\begin{array}{r} 01010111(+87) \\ +) 01111001(+121) \\ \hline \text{Cy}=0 \ 11010000(\text{结果为负}) \\ \text{C6}=1, \text{C7}=0 \rightarrow \text{OV}=1 \end{array}$$

**例2**

$$\begin{array}{r} 10001000(-120) \\ +) 10010111(-105) \\ \hline \text{Cy}=1 \ 00011111(\text{结果为正}) \\ \text{C6}=0, \text{C7}=1 \rightarrow \text{OV}=1 \end{array}$$



## 特殊功能寄存器SFR(1)

### (3) 程序状态字PSW (C、AC、F0、F1、RS1、RS0、OV、P)

➤ **P**: 奇偶标志。用以表示指令操作之后, 累加器A中1的个数的奇偶性。

- 若A中“1”的个数为奇数个, 则P=1;
- 若A中“1”的个数为偶数个, 则P=0;

只要A中的数据变化, P的状态亦随之变化。如:

MOV A, #35H (00110110); 则P=0;

ADD A, #14H; 则 (A) =49H (01000101), P=1

# 休息一下



# 提 纲

## 5. 特殊功能寄存器SFR (2)



## 特殊功能寄存器SFR (2)

### 5. 特殊功能寄存器介绍

#### (4) 堆栈指针SP (Stack Pointer)

##### ➤ 堆栈的概念:

堆栈是定义为特殊用途的存储区，主要功能是临时存放数据和地址，通常用于保护断点和保护现场。堆栈有二种形式，一是向上（向高地址）生成，二是向下（向低地址）生成。

8051 MCU的堆栈为满顶法向上生成的软件堆栈，其堆栈区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，最后进栈的数据最先被弹出。压入堆栈的数据总是保存在堆栈的顶部，从堆栈弹出的总是栈顶的数据。





## 特殊功能寄存器SFR (2)

### (4) 堆栈指针SP

#### ➤ 堆栈指针SP (Stack Pointer) :

存放堆栈栈顶地址的一个8位寄存器，地址为81H。

8051 MCU的堆栈是向上生成的：进栈时栈顶向高地址生长，SP的内容增加；出栈时栈顶向下回落，SP的内容减少。所以SP总是指向堆栈的栈顶。

#### ➤ 堆栈的设置：

8051微控制器的堆栈必须设在内部通用RAM中。MCU复位后SP内容为07H，即默认堆栈从08H单元开始。

由于08H-1FH单元为工作寄存器区，20-2FH为位寻址区，程序设计中很有可能要用到这些单元，所以通常是通过SP赋值而重新设置堆栈区域，使堆栈区设定在内部RAM的其它区域。



## 特殊功能寄存器SFR (2)

### (4) 堆栈指针SP

#### ➤ 堆栈的操作方式:

- 指令方式: 使用堆栈操作指令进行进栈、出栈操作, 以实现一些数据的暂存, 以及对现场的保护和恢复。

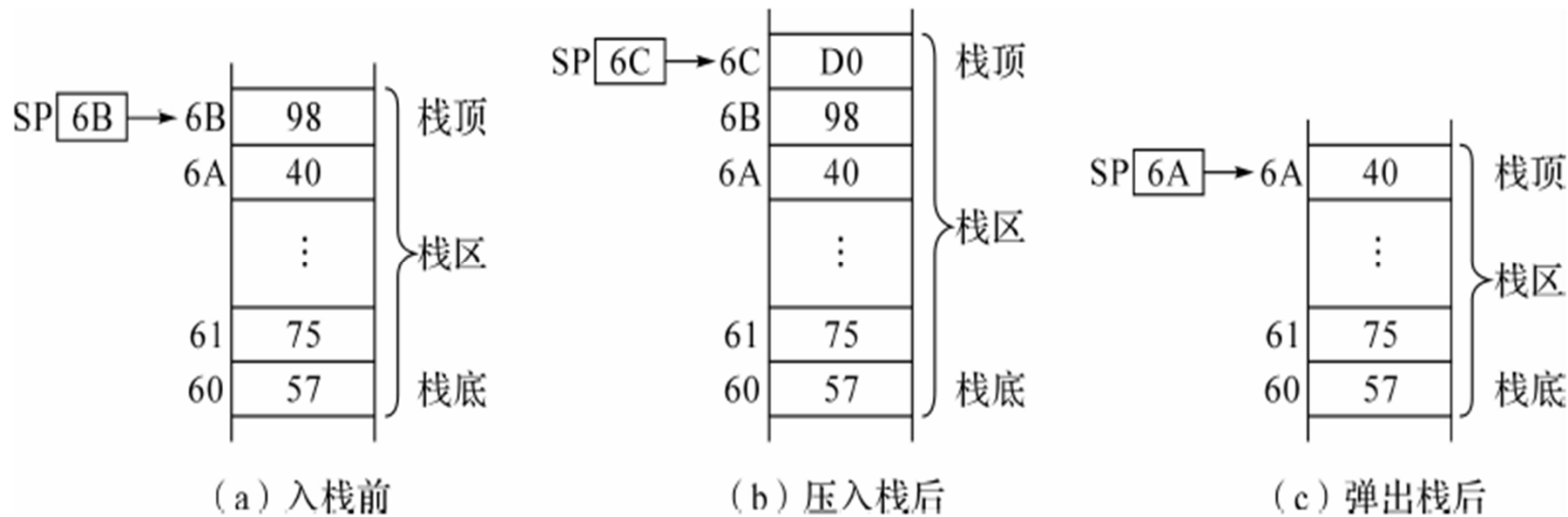
#### 堆栈操作指令:

- 进栈指令: PUSH direct (direct 范围: 00H-FFH)
- 出栈指令: POP direct
- 自动方式: 在调用子程序或发生中断时, 将子程序返回地址或中断的断点地址自动进栈保护; 在子程序和中断程序返回时, 调用地址或断点地址自动弹回PC。该操作由硬件自动完成。

## 特殊功能寄存器SFR (2)

### (4) 堆栈指针SP

#### ➤ 堆栈的操作过程:



栈底=60H（第一个进栈数据存放单元），压入数据，堆栈向上生长，图示栈顶是6BH，（SP）=6BH，即SP指向栈顶。

再压入1个数据D0H，压入6CH单元，栈顶变为6CH，（SP）=6CH。

弹出2个数据，栈顶变为6AH，（SP）也变为6AH。



## 特殊功能寄存器SFR (2)

### (4) 堆栈指针SP

#### ➤ 堆栈的深度:

子程序调用和中断都允许多级嵌套，现场保护需要使用堆栈，所以一定要保证堆栈有一定的深度，要避免堆栈溢出。

设置时要注意堆栈的深度，但不能超过内部RAM的空间。

例: (答案见后面)

设置堆栈指针 (SP) = 5FH, 堆栈从哪里开始?

如设置堆栈指针 (SP) = FFH, 堆栈从哪里开始? 是否可以?



## 特殊功能寄存器SFR (2)

### (4) 堆栈指针SP

#### ➤ 堆栈的深度:

子程序调用和中断都允许多级嵌套，现场保护需要使用堆栈，所以一定要保证堆栈有一定的深度，要避免堆栈溢出。

设置时要注意堆栈的深度，但不能超过内部RAM的空间。

#### 例: (答案)

设置堆栈指针 (SP) = 5FH, 堆栈从60H开始

如设置堆栈指针 (SP) = FFH, 堆栈溢出, 不可以!



## 特殊功能寄存器SFR (2)

### 5. 特殊功能寄存器介绍

#### (5) 数据指针DPTR (Data Pointer)

数据指针DPTR是一个16位的SFR。其功能是外部RAM（地址范围0000H-FFFFH）的地址指针，是存放外部RAM地址的16位寄存器。

DPTR由两个8位寄存器组成，高8位为DPH表示，低8位用DPL。

DPH地址：83H； DPL地址： 82H



## 特殊功能寄存器SFR (2)

### 5. 特殊功能寄存器介绍

#### (6) P0-P3端口寄存器

P0、P1、P2、P3：分别是I/O端口P0~P3的锁存器，地址分别为：80H、90H、A0H、B0H，可以位寻址。

对于端口即引脚的操作实际上是对这些寄存器的操作，其端口引脚与端口寄存器的位具有映射关系。

#### (7) 其它特殊功能寄存器

SBUF、IP、IE、TMOD、TCON、SCON、PCON等，将结合相关章节内容进行介绍。

微控制器复位后，除SP为07H，P0~P3为FFH外，其余均为0。



## 特殊功能寄存器SFR(2)

### 6. 内部RAM和SFR的寻址方式

- 对于**00H-7FH**存储空间：可运用直接寻址和寄存器间接寻址这两种寻址方式，对其进行访问。
- 对于**80H-FFH**存储空间：只能采用寄存器间接寻址方式进行访问。
- 对于地址范围同为**80H-FFH**的特殊功能寄存器（SFR）：只能采用直接寻址方式。

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。



# 提 纲

## 6. I/O端口结构与应用特点

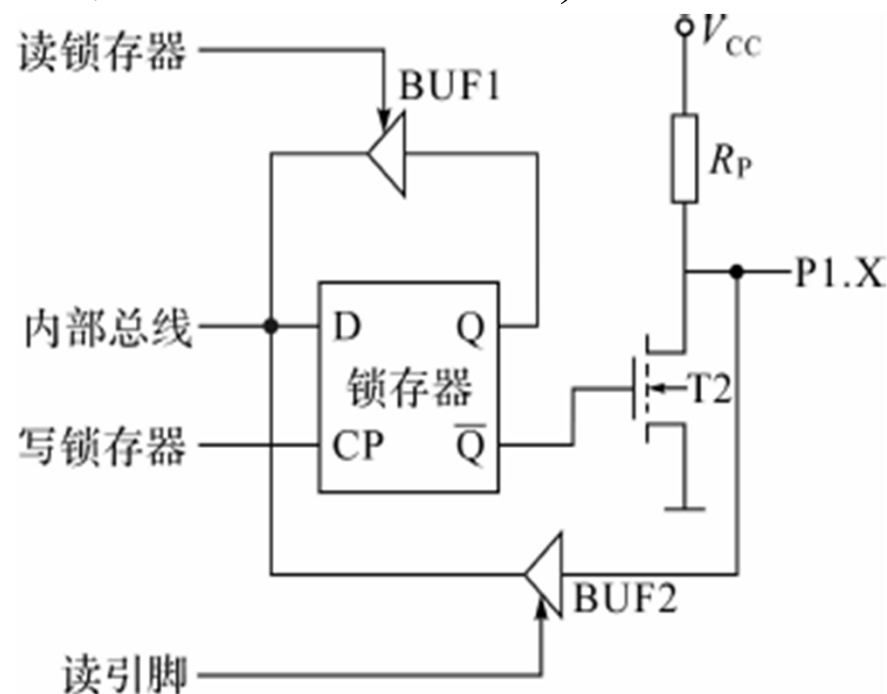
## I/O端口结构和应用特点——P0—P3端口的内部结构

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。(P0没有上拉电阻，要外接上拉电阻)

输出时：当CPU通过内部总线向端口锁存器输出1或0时，通过输出驱动电路，端口相应引脚就会输出高电平或低电平。

输入时：即读引脚时，如果端口锁存器状态为0，则T2导通，引脚被钳位在“0”状态，导致无法得到端口引脚的高电平状态。（准双向口的特点）（读之前先写1）



P1端口内部结构



## I/O端口结构 and 应用特点——P0—P3端口的内部结构

### 1. P0端口 (P0.7~P0.0)

字节地址为80H，位地址为80H-87H。

- 第一功能：准双向I/O接口，做输出口使用时，需要外接上拉电阻。
- 第二功能：分时复用的8位数据线D7-D0和低8位地址线A7-A0，在扩展外部存储器或I/O接口时使用。

### 2. P1端口 (P1.7~P1.0)

字节地址为90H，位地址为90H-97H。

带有内部上拉电阻的8位准双向I/O口，无第二功能。

### 3. P2端口 (P2.7~P2.0)

字节地址为A0H，位地址为A0H-A7H。

- 第一功能：准双向I/O接口，带有内部上拉电阻。
- 第二功能：在扩展外部存储器时，用作高8位地址线A15-A8。



## I/O端口结构和应用特点——P0—P3端口的内部结构

### 4. P3端口（P3.7~P3.0）

字节地址为B0H，位地址为B0H-B7H；

第一功能：准双向I/O接口，带有内部上拉电阻。

第二功能：

端口引脚	第二功能	英文注释
P3.0	RXD（串行口输入）	Receive eXternal Data
P3.1	TXD（串行口输出）	Transmitted eXternal Data
P3.2	INT0（外部中断0输入）	Interrupt 0
P3.3	INT1（外部中断1输入）	Interrupt 1
P3.4	T0（定时器0计数输入）	Timer 0
P3.5	T1（定时器1计数输入）	Timer 1
P3.6	WR（外部RAM“写”选通）	Write
P3.7	RD（外部RAM“读”选通）	Read



# I/O端口结构和应用特点——P0—P3端口的内部结构

P0-P3端口既有共同点，也有差异点。使用时，应根据各端口的特点正确使用。

## 1. 准双向I/O口特性

- P0-P3端口均是准双向口，准双向口的输入操作和输出操作本质不同，输入操作是读引脚状态；输出操作是对口锁存器的写入操作。
- 在输出时，与真正双向口一样，CPU输出的“0”或“1”，通过输出驱动电路在引脚上表现为低电平或高电平。



# I/O端口结构 and 应用特点——P0—P3端口的内部结构

## 1. 准双向I/O口特性

➤ 在输入时，必须先向锁存器输出1，使得输出驱动电路中的T2处于截止状态，即将端口设置为输入方式。只有这样外部引脚的高、低电平状态才能被正确读入；反之若T2处于导通状态，则端口的引脚被钳位在0电平，而无法得到引脚所接外设的真实状态。

- 准双向口的输入操作：先向锁存器输出1，然后再输入引脚状态。

例：要将P1口状态读入A中，应执行以下两条指令

MOV P1, #0FFH ; P1口置为输入方式

MOV A, P1 ; 读P1口引脚状态



# I/O端口结构 and 应用特点——P0—P3端口的内部结构

## 2. I/O口的应用特性

- 4个端口的第一功能均为准双向I/O口。P0口在用作第一功能时，要外接上拉电阻；P1~P3口内部有上拉电阻。
- P0口的每一个引脚能驱动8个LSTTL输入端，而P1~P3口可驱动4个LSTTL输入端。

在全CMOS应用系统中，由于CMOS电路的输入驱动电流极微，因此通常不必考虑MCU I/O端口的驱动能力。只有在I/O端口作功率驱动，如驱动LED、可控硅、继电器时，才考虑I/O口的驱动能力。

# 提 纲

## 7. 时钟与复位



## 时钟和复位——时钟电路与时序

时钟是微控制器内部CPU和各功能模块，协调工作需要的同步时钟信号和基本时序信号。

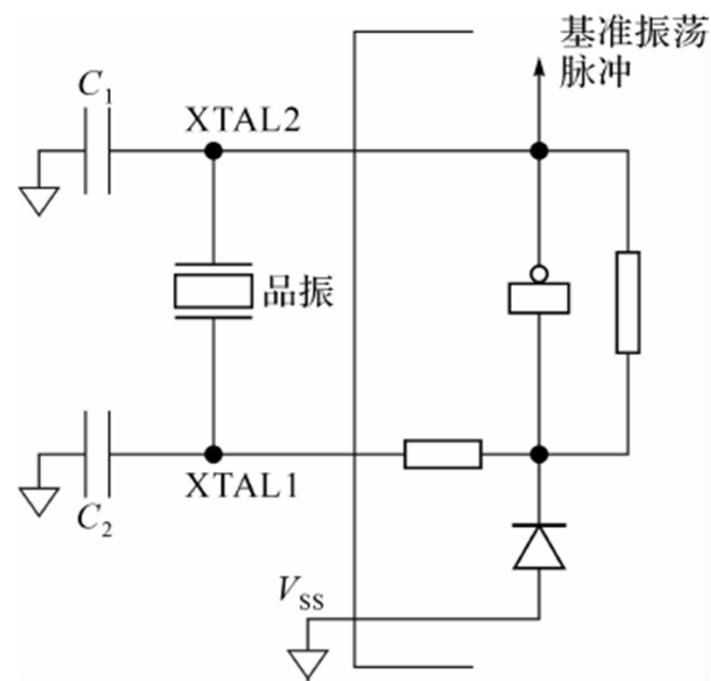
### 1. 时钟电路

8051 MCU内部具有时钟电路，在引脚XTAL1、XTAL2外接晶体振荡器、电容，就会产生MCU工作所需要的时钟信号。

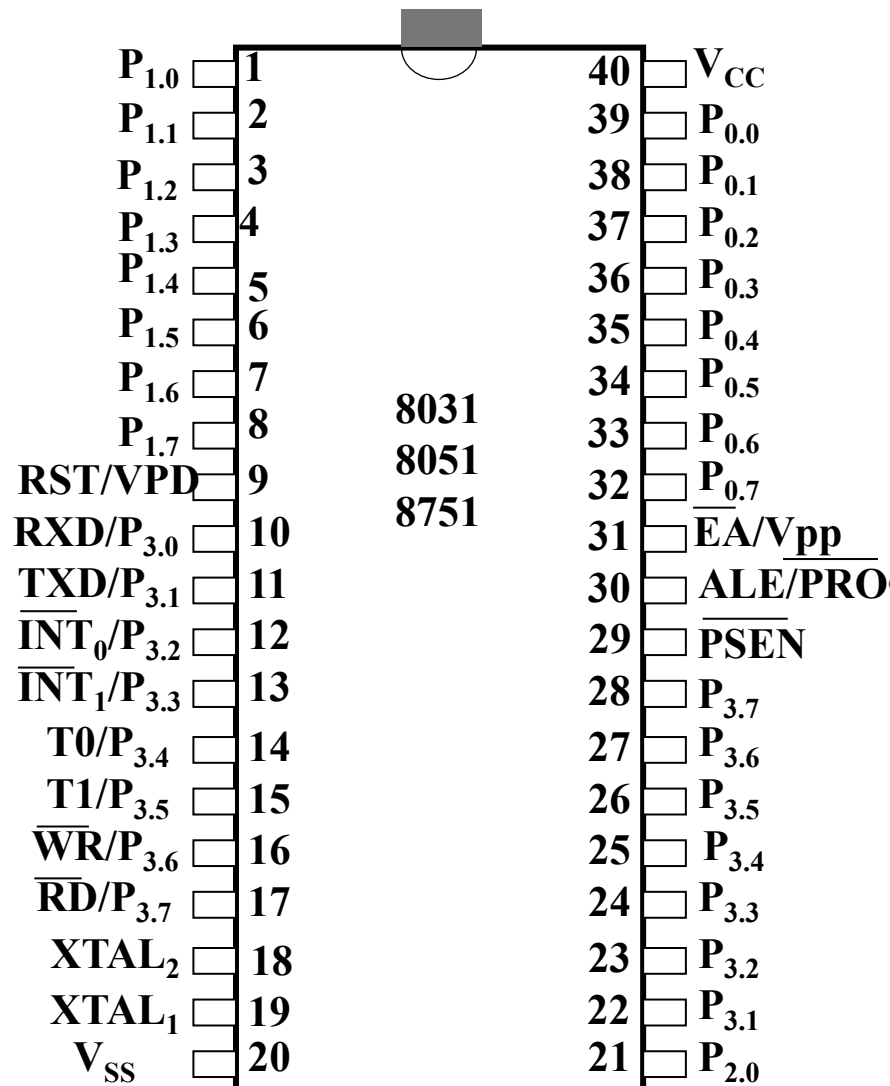
外接晶振的频率决定MCU的时钟频率。

时钟频率=外接晶振频率

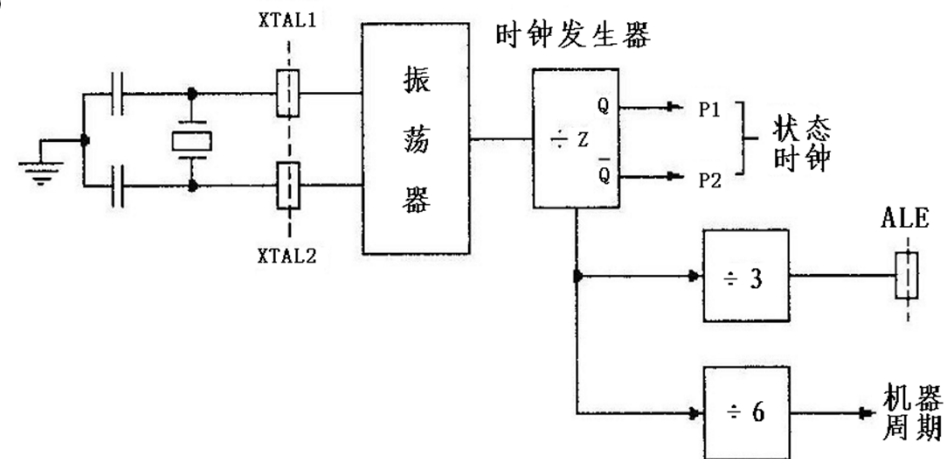
电容C1和C2对振荡频率起稳定微调作用。  
C1、C2必须相等，一般取30pf左右。



# 时钟和复位——时钟电路与时序



- XTAL1 (Pin19)：用作片内振荡电路的输入端。
- XTAL2 (Pin18)：用作片内振荡电路的输出端或者外部时钟源的输入引脚。





# 时钟和复位——时钟电路与时序

## 2. 时序与工作周期

微控制器的时序就是CPU和功能模块工作时，各控制信号之间的时间顺序关系。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微控制器规定了几种工作周期：

- 时钟周期（振荡周期）
- 状态周期
- 机器周期
- 指令周期



## 时钟和复位——时钟电路与时序

### 3. 工作周期（时钟周期、状态周期、机器周期、指令周期）

#### ➤ 时钟周期 $T_0$

也称为振荡周期，是外接晶振频率的倒数；是微控制器中最基本、最小的时间单位。若振荡源频率为 $f_{osc}$ ，则振荡周期为 $T_0 = 1 / f_{osc}$ ；一个振荡周期定义为一个节拍P。

若晶振频率 $f_{osc}$ 为6MHz，则时钟周期为 $1 / f_{osc}$ 即 $1/6 \mu s$ ；若晶振频率为 $f_{osc}$ 为12MHz，则时钟周期为 $1 / f_{osc}$ 即 $1/12 \mu s$ 。

- 外接晶振频率的高低，决定了CPU执行速度的快慢。
- 但振荡频率太高，会引入高频辐射。
- 不同型号MCU有不同的时钟频率范围要求。



## 时钟和复位——时钟电路与时序

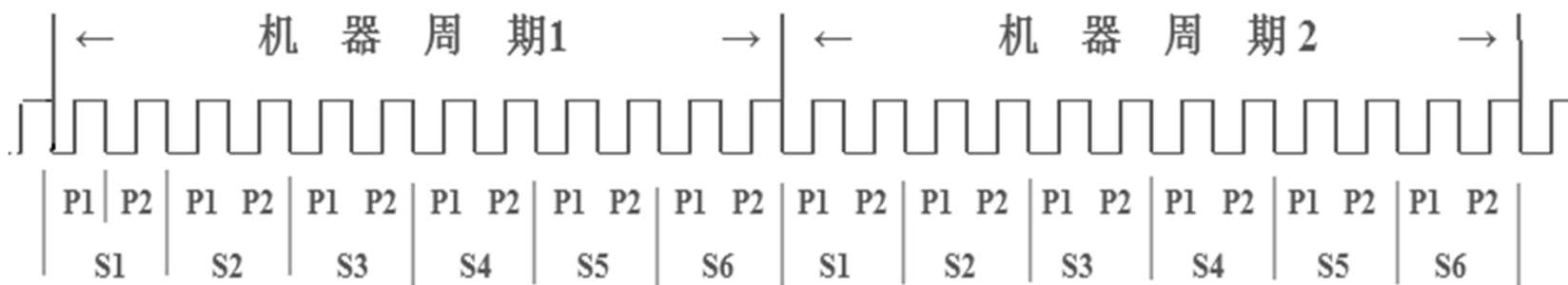
### 3. 工作周期（时钟周期、状态周期、机器周期、指令周期）

#### ➤ 状态周期S

是时钟周期的两倍  $S = 2T = 2/f_{osc}$ ，即由连续的两个节拍P1和P2组成。

#### ➤ 机器周期 $T_M$

CPU执行一个基本操作所需要的时间。一个机器周期由6个状态周期（S1-S6）即12个时钟周期组成。



1个机器周期 $T_M = 6$ 个状态周期 $S = 12$ 个振荡周期 $T_0$



## 时钟和复位——时钟电路与时序

### 3. 工作周期（时钟周期、状态周期、机器周期、指令周期）

#### ➤ 指令周期

执行一条指令所需要的时间，为指令周期。

通常每条指令的执行可划分为1-4个基本操作，完成一个基本操作所需时间为机器周期。所以指令周期有1-4个机器周期组成。



# 时钟和复位——复位与复位电路

复位是使计算机或微控制器退出死机或无效状态，重新开始工作的方法；是MCU的初始化操作。

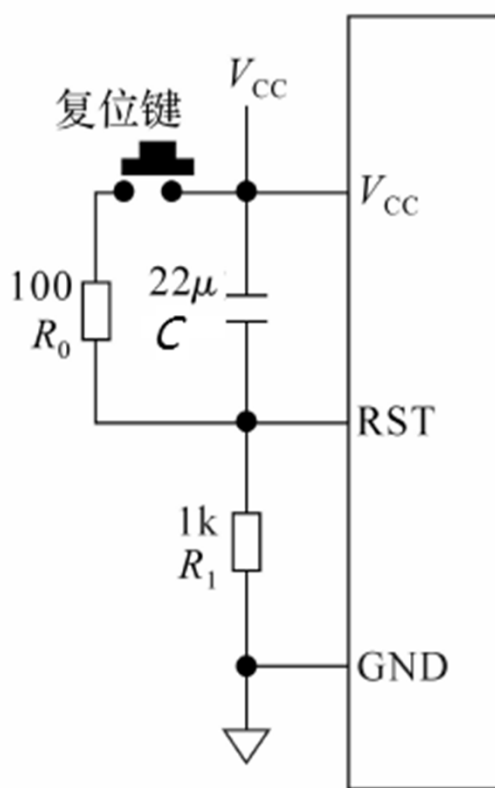
## 1. 复位电路

8051微控制器采用高电平复位。经典8051微控制器没有内置复位电路，因此需设计外部复位电路。

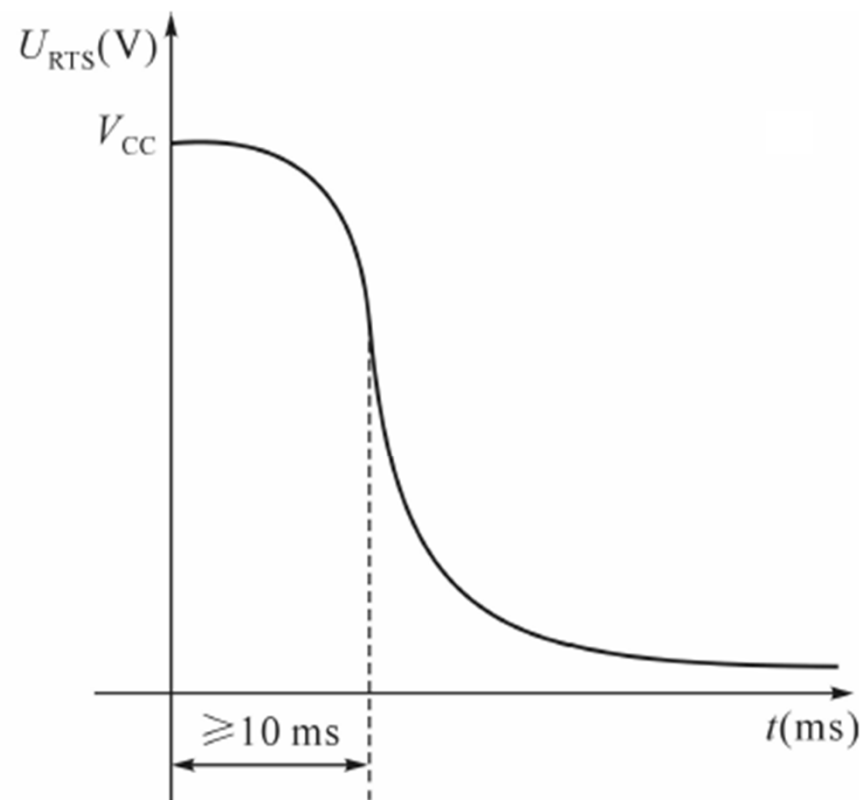
- 上电复位（也称为“冷启动”）：利用系统上电对C充电时，在RST引脚上产生高电平进行复位。复位信号（RST端的高电平）需维持10ms以上，才能实现可靠复位。
- 按键复位（也称为“热启动”）：按键复位时，RST端的高电平只要 $\geq 2$ 个机器周期，即可实现复位。

# 时钟和复位——时钟电路与时序

## 1. 复位电路



(a) 上电和按键复位电路



(b) 上电过程RST引脚电平变化





# 时钟和复位——时钟电路与时序

## 2. 复位状态

复位后，8051微控制器的初始化状态为：

- (PC) = 0000H;
- (SP) = 07H;
- (PSW) = 00H;
- P0~P3端口的锁存器为FFH，端口引脚全为1，处于可输入状态；
- 除上述SFR外，其余特殊功能寄存器SFR均为0；

RAM数据在热复位后保持不变，在上电复位后为随机数。



## 问题

1. RAM和SFR都有地址80H~7FH的部分，怎么回事？如何保证访问的时候不冲突？
2. RAM中的位寻址区可以字节寻址吗？
3. SP的内容是40H，那么堆栈从哪个地址开始？



## 问题

1. RAM和SFR都有地址80H~7FH的部分，怎么回事？如何保证访问的时候不冲突？

答：访问的寻址方式不一样，RAM的80H~7FH用寄存器间接寻址，SFR用直接寻址。

2. RAM中的位寻址区可以字节寻址吗？

答：可以。

3. SP的内容是40H，40H是什么存储器的地址？而堆栈实际从哪个地址开始？

答：40H是RAM地址，从41H开始，即下一个入堆栈的内容放置于41H单元。

# Thank you!

