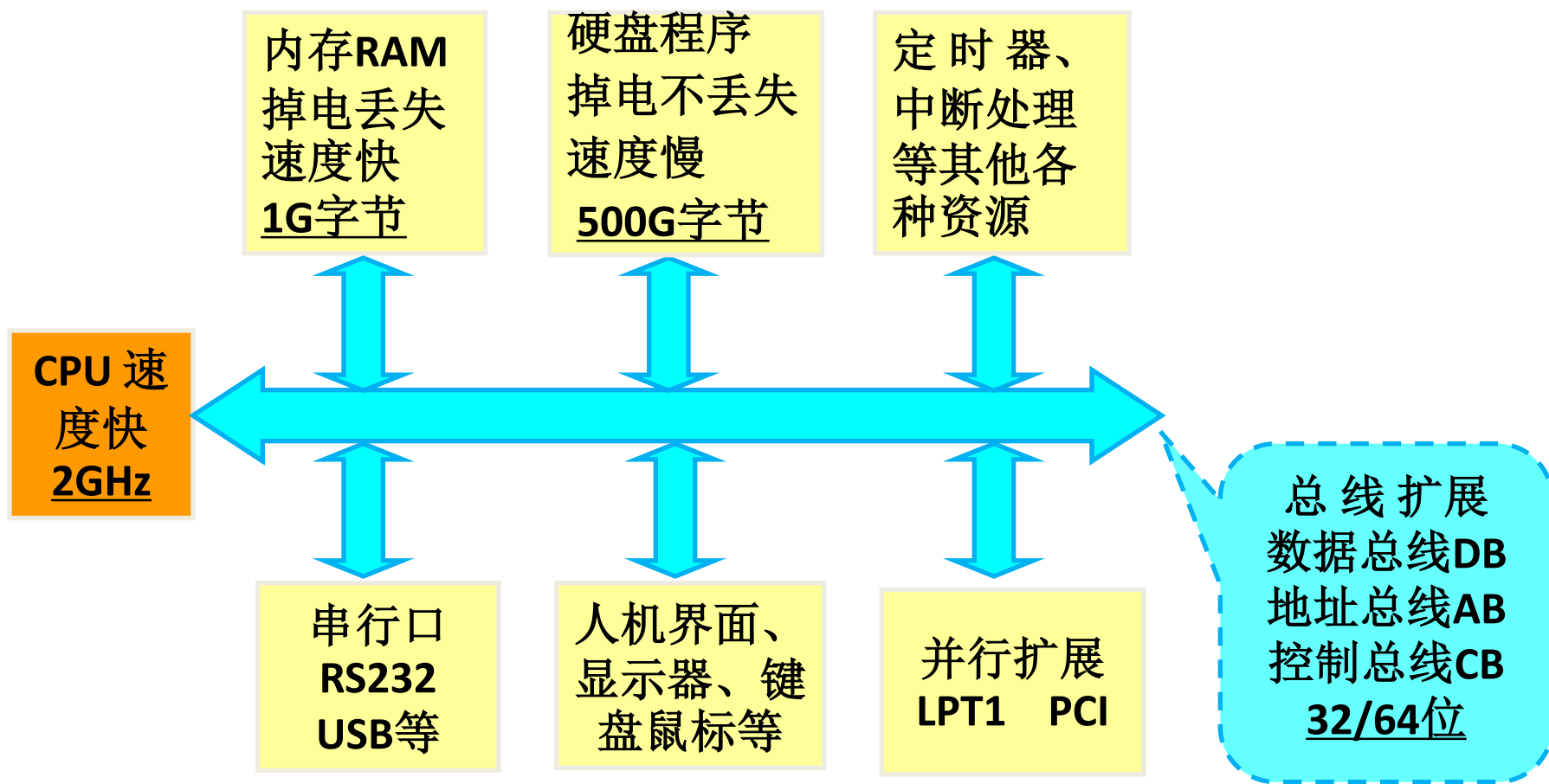
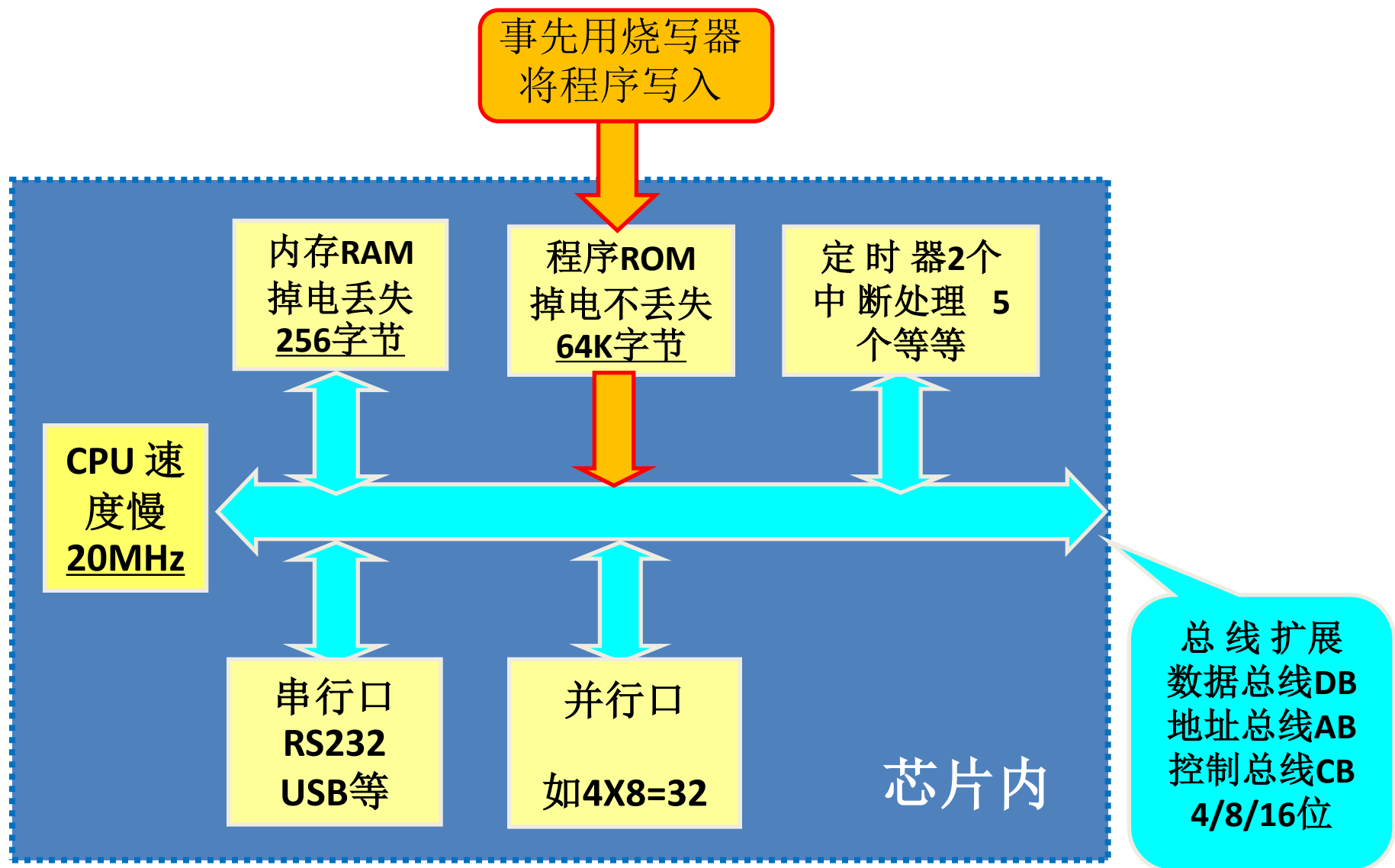


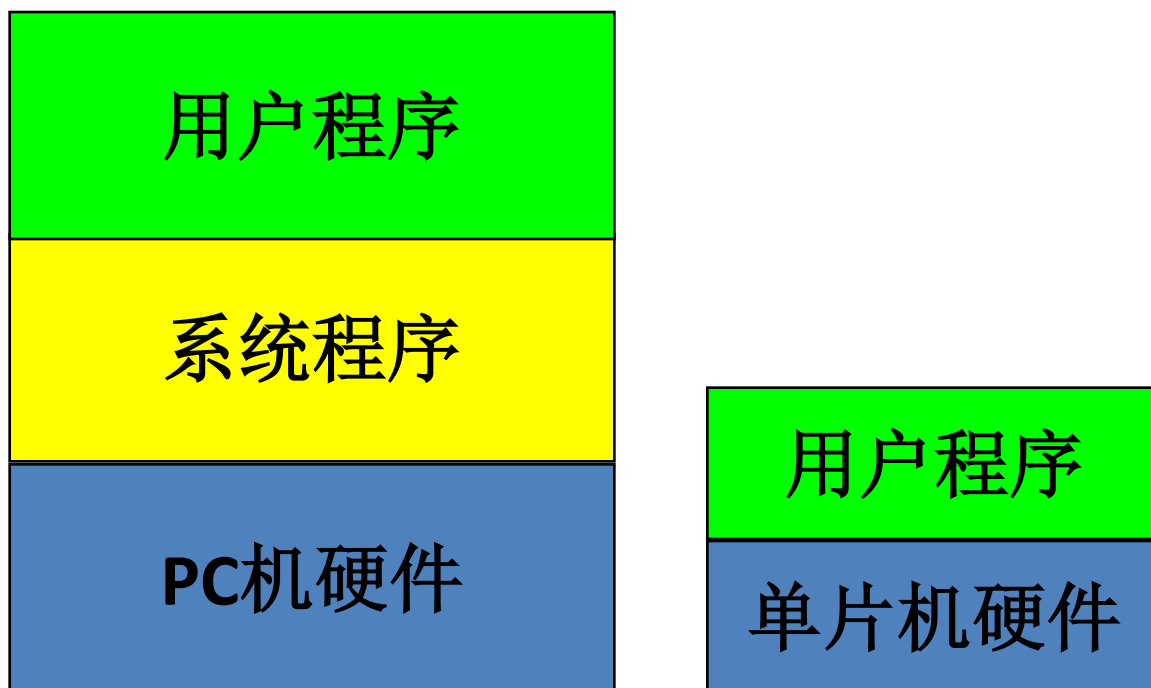
# 复习1 微型计算机硬件构成



## 复习2 单片机硬件构成



## 复习3 单片机和计算机的用户程序对比

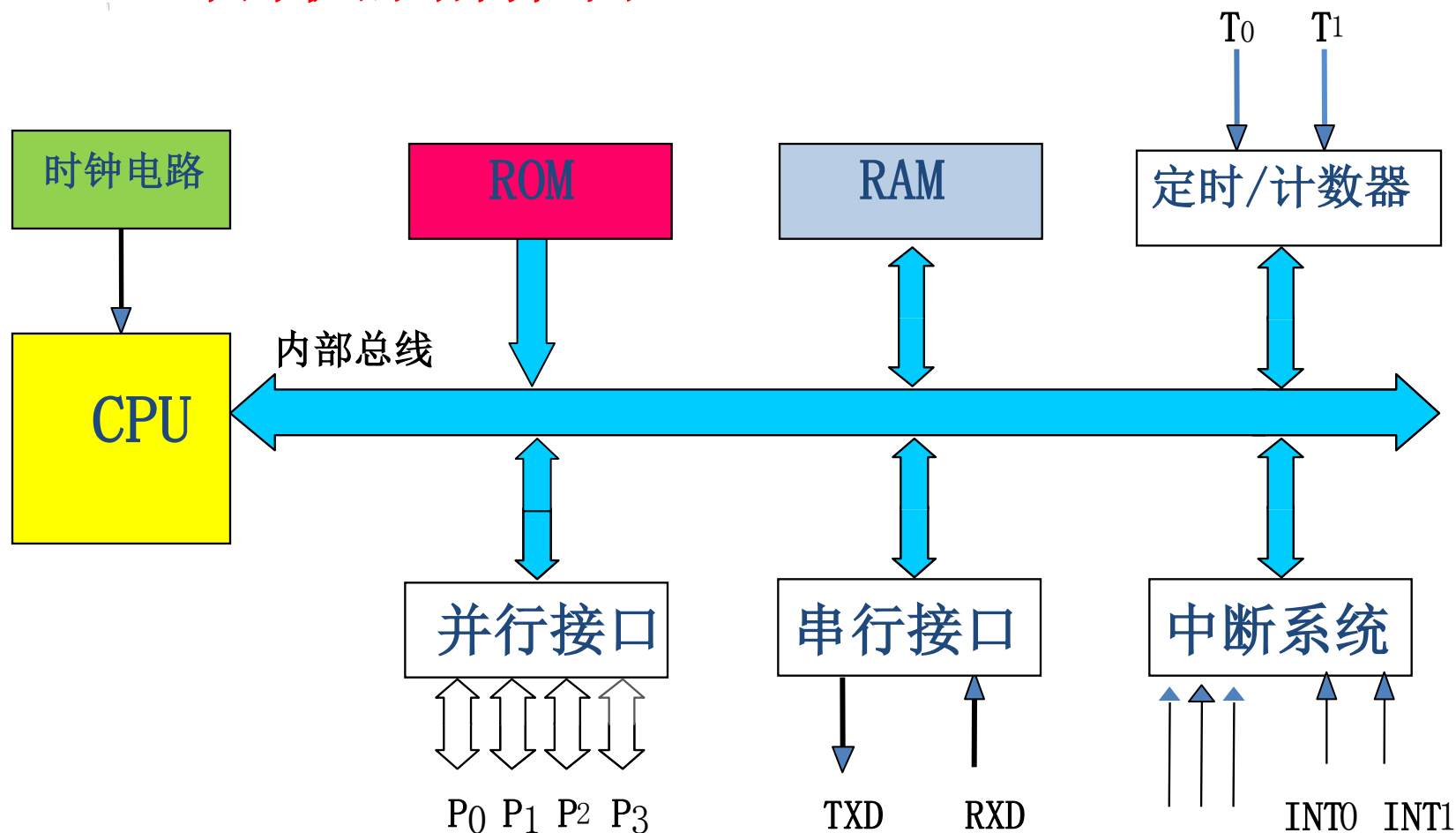


## 第2章 80C51 单片机的硬件结构

- 2.1 80C51单片机的逻辑结构及信号引脚
- 2.2 80C51单片机的内部存储器
- 2.3 80C51 单片机的并行I/O口
- 2.4 80C51 单片机的时钟与定时
- 2.5 80C51 单片机的系统复位
- 2.6 单片机低功耗工作模式

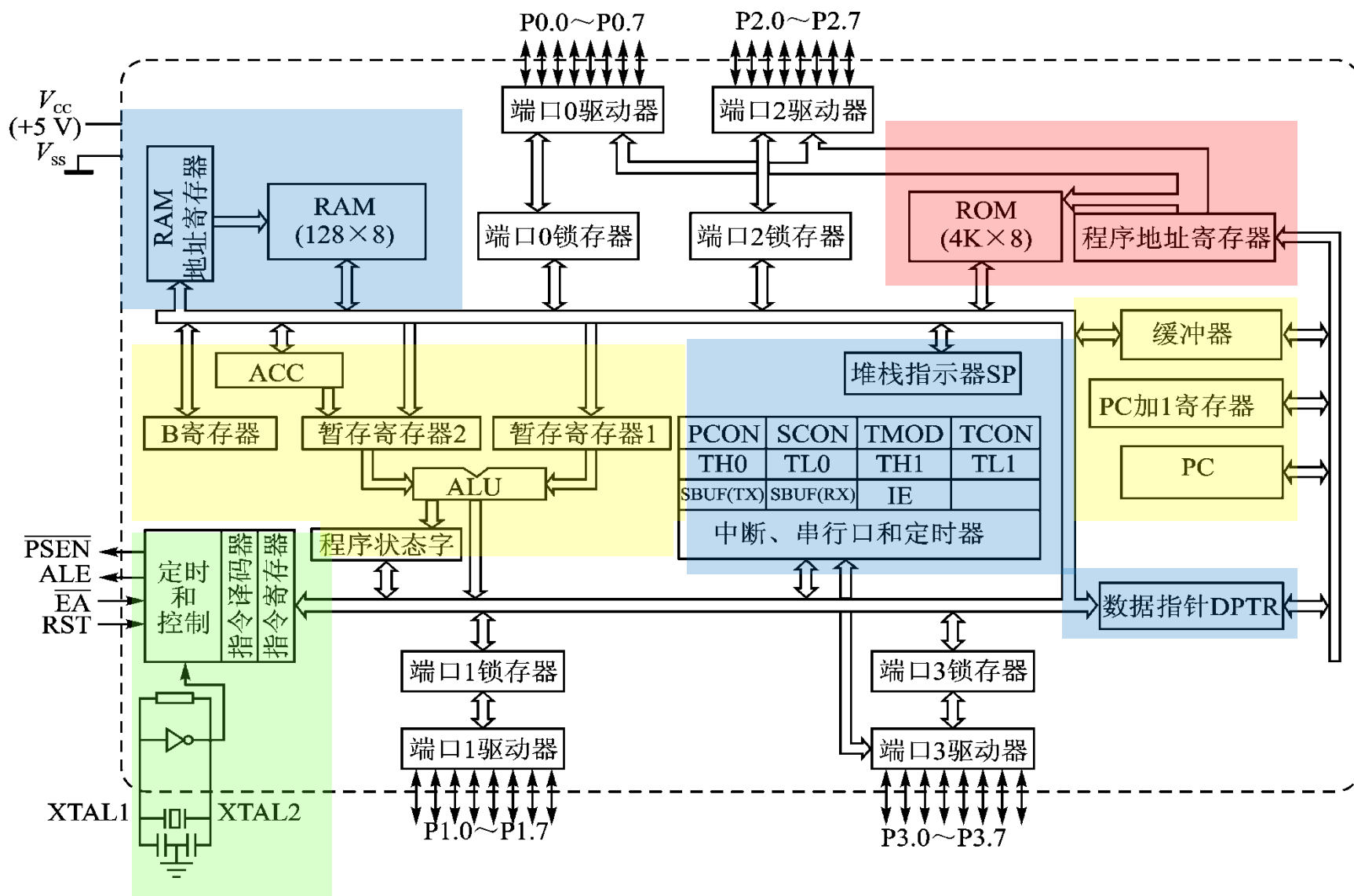
## 2.1 80C51单片机的逻辑结构和信号引脚

### 1、51单片机的结构框图

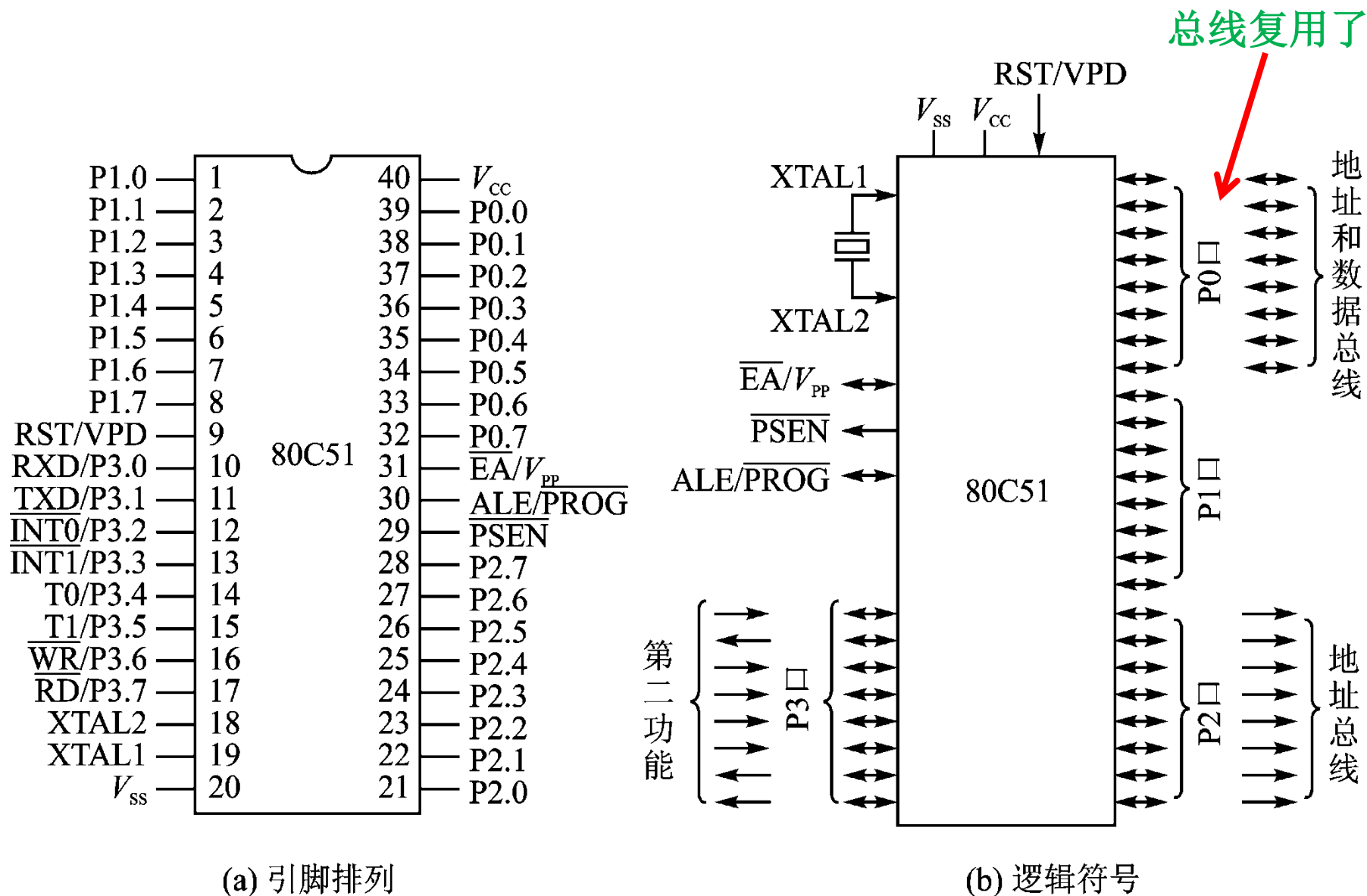


**MCS-51单片机功能模块框图**

## 2、51单片机芯片内逻辑结构图

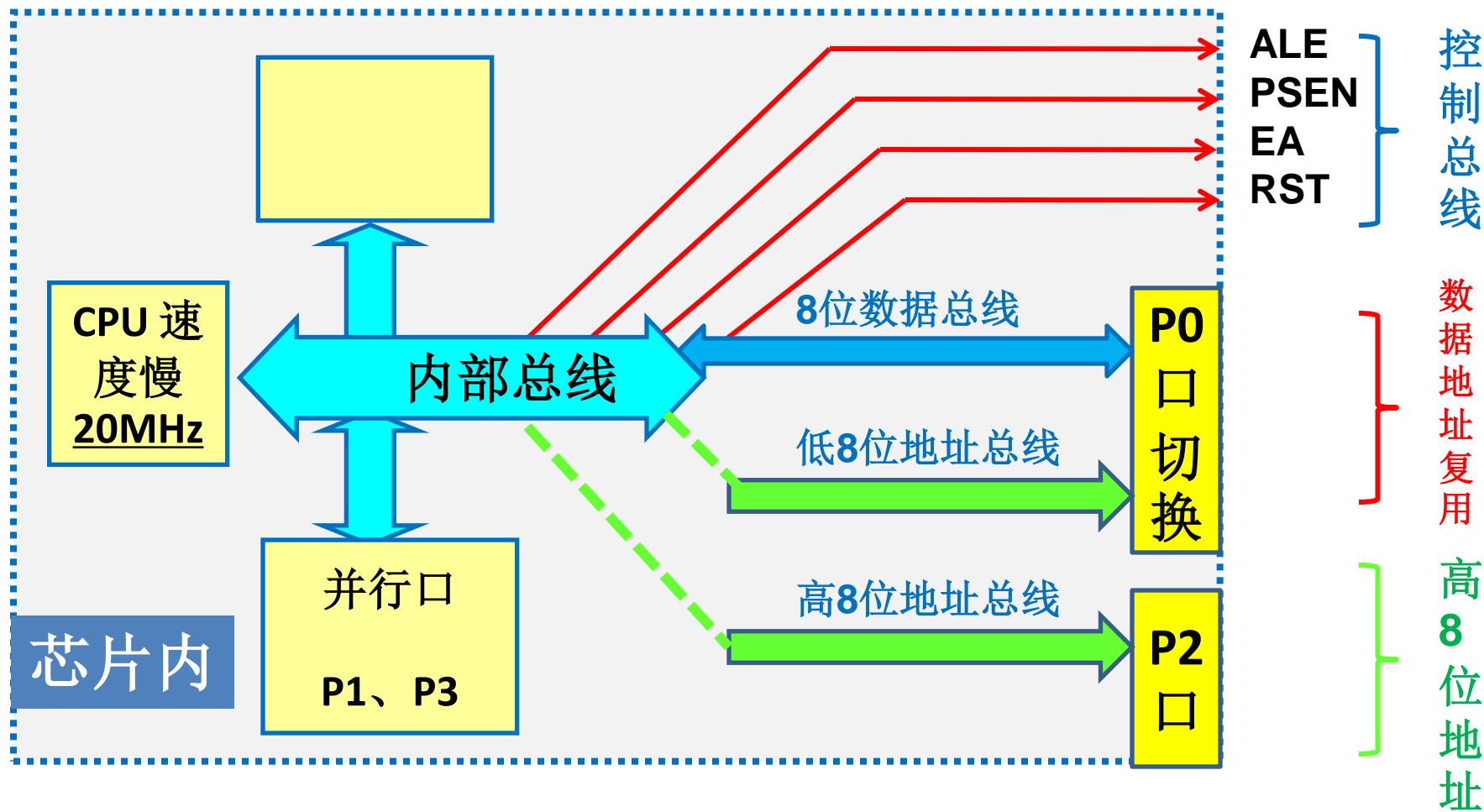


### 3、C51单片机的封装与信号引脚

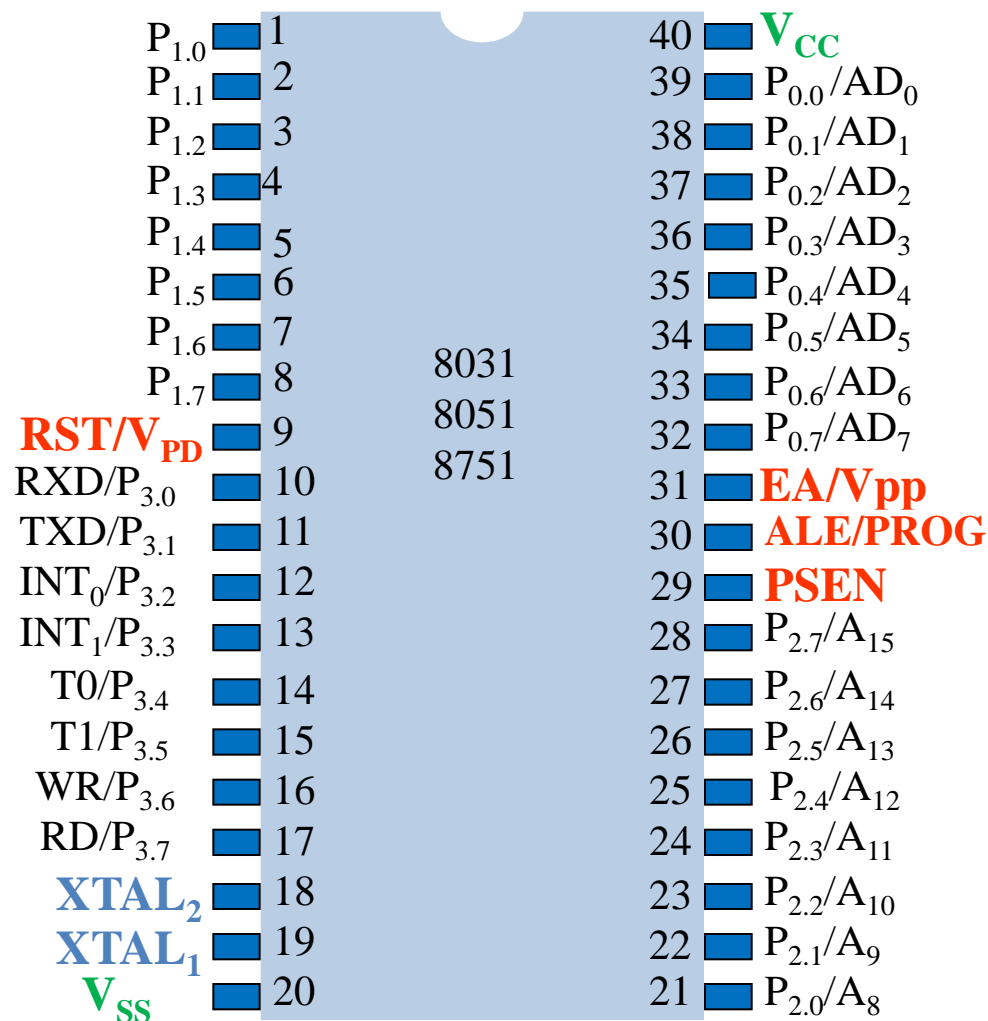


# 单片机内部总线复用示意

**C51**为节约IO口数量，内部总线除控制总线外没有单独占用口，而采用复用技术，即高8位地址通过**P2**口复用输出，低8位地址和8位数据通过**P0**口分时共用输出，如图所示。







单片机引脚图和实际芯片

## 4、MCS-51单片机引脚功能

- 1) 主电源引脚 $V_{CC}$ 和 $V_{SS}$  ( 2根)
- 2) 外接晶振引脚 $XTAL_1$ 和 $XTAL_2$  ( 2根)
- 3) 控制或其他电源复用引脚RST/  $V_{PD}$ 、  
ALE/PROG、PSEN、和EA/ $V_{PP}$  ( 4根)
- 4) 输入/输出引脚P0、P1、P2、P3 (32根)

## 5、控制引脚说明：

1) 地址锁存控制信号 **ALE**：访问片外ROM, RAM时，**ALE**用于控制把P0口输出的低8位地址送入锁存器锁存，实现低位地址和数据的分时传送。不访问片外存储器时，可做为外部时钟使用。

2) 外部程序存储器读选通信号  **$\overline{\text{PSEN}}$** ：CPU访问片外ROM时，使**PSEN**低电平有效，可实现片外ROM的读操作，其他情况下此引脚为高电平封锁状态。

3) 访问程序存储器控制信号  **$\overline{\text{EA}}$** ：当**EA**信号为低电平时，对ROM的读操作是针对外部ROM的，当**EA**信号为高电平时，对ROM的操作是从内部ROM开始，并可延续至外部ROM。

4) 复位信号 **RST**：复位即初始化，分自动上电复位和人工开关复位，复位时，产生一个延续2个机器周期以上的高电平为有效，实现单片机的复位操作。

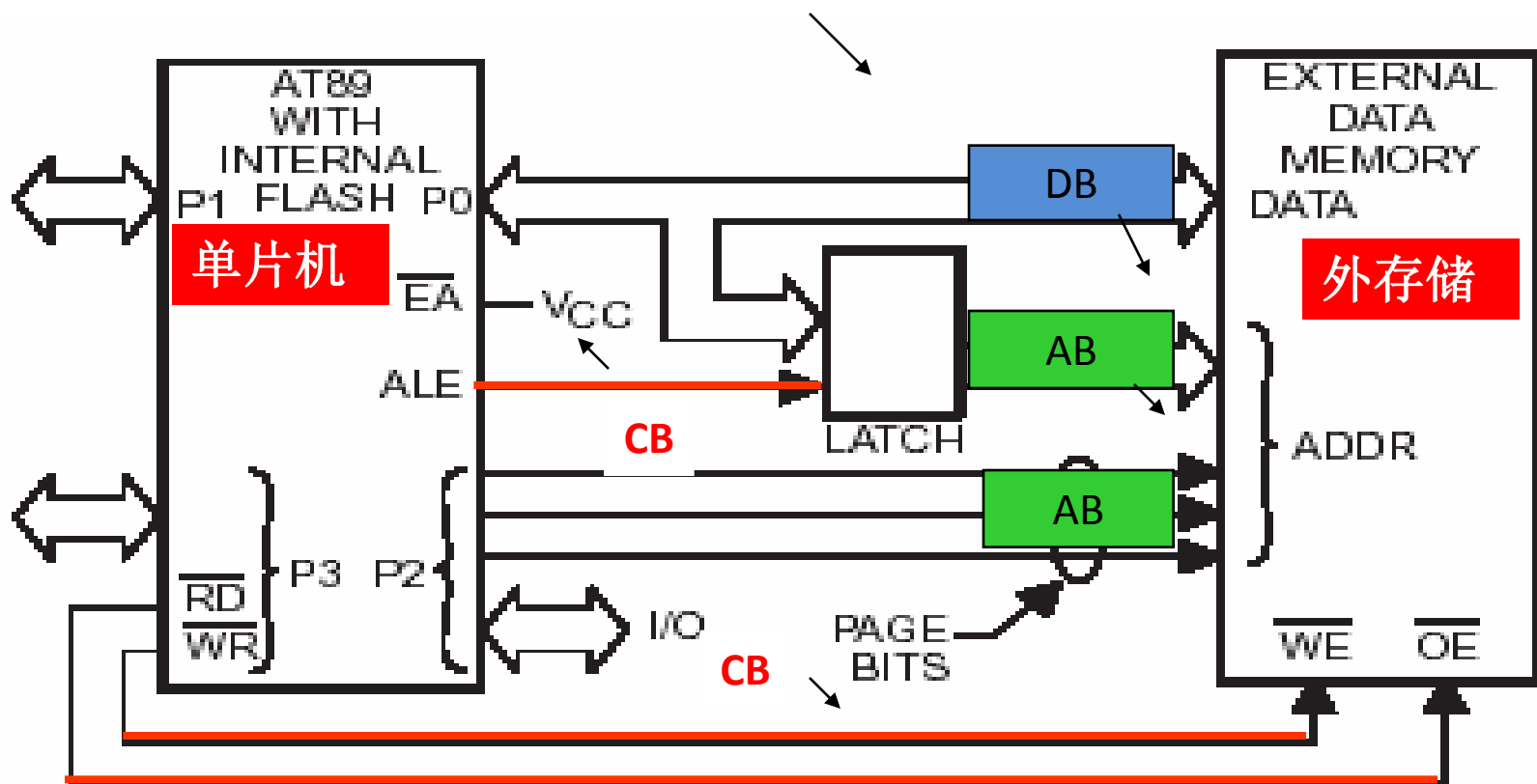
## 6、片外三总线

## 扩展外部总线时，采用锁存器，将P0的数据和地址总线分离

**AB: P0口经地址锁存后提供低8位地址， P2口直接提供高8位地址**

## DB: P0口提供8位数据

**CB: /EA, ALE, /PSEN, /RD, /WR, RST等**



## 7、P<sub>3</sub>口的第二功能表

I/O口	第二功能	注 释
P <sub>3.0</sub>	RXD	串行口数据接收端
P <sub>3.1</sub>	TXD	串行口数据发送端
P <sub>3.2</sub>	$\overline{\text{INT}}_0$	外部中断请求0
P <sub>3.3</sub>	$\overline{\text{INT}}_1$	外部中断请求1
P <sub>3.4</sub>	T <sub>0</sub>	定时/计数器0
P <sub>3.5</sub>	T <sub>1</sub>	定时/计数器1
P <sub>3.6</sub>	$\overline{\text{WR}}$	外部RAM写信号
P <sub>3.7</sub>	$\overline{\text{RD}}$	外部RAM读信号

**P3** 口线的第二功能都是单片机重要的控制信号。因此在使用时，总是按需要优先选用他的第二功能，剩下不用的才作为口线使用。

## 2.2 单片机内部存储器

### 一、 内部数据低128单元

#### 1、分类

##### 51子系列:

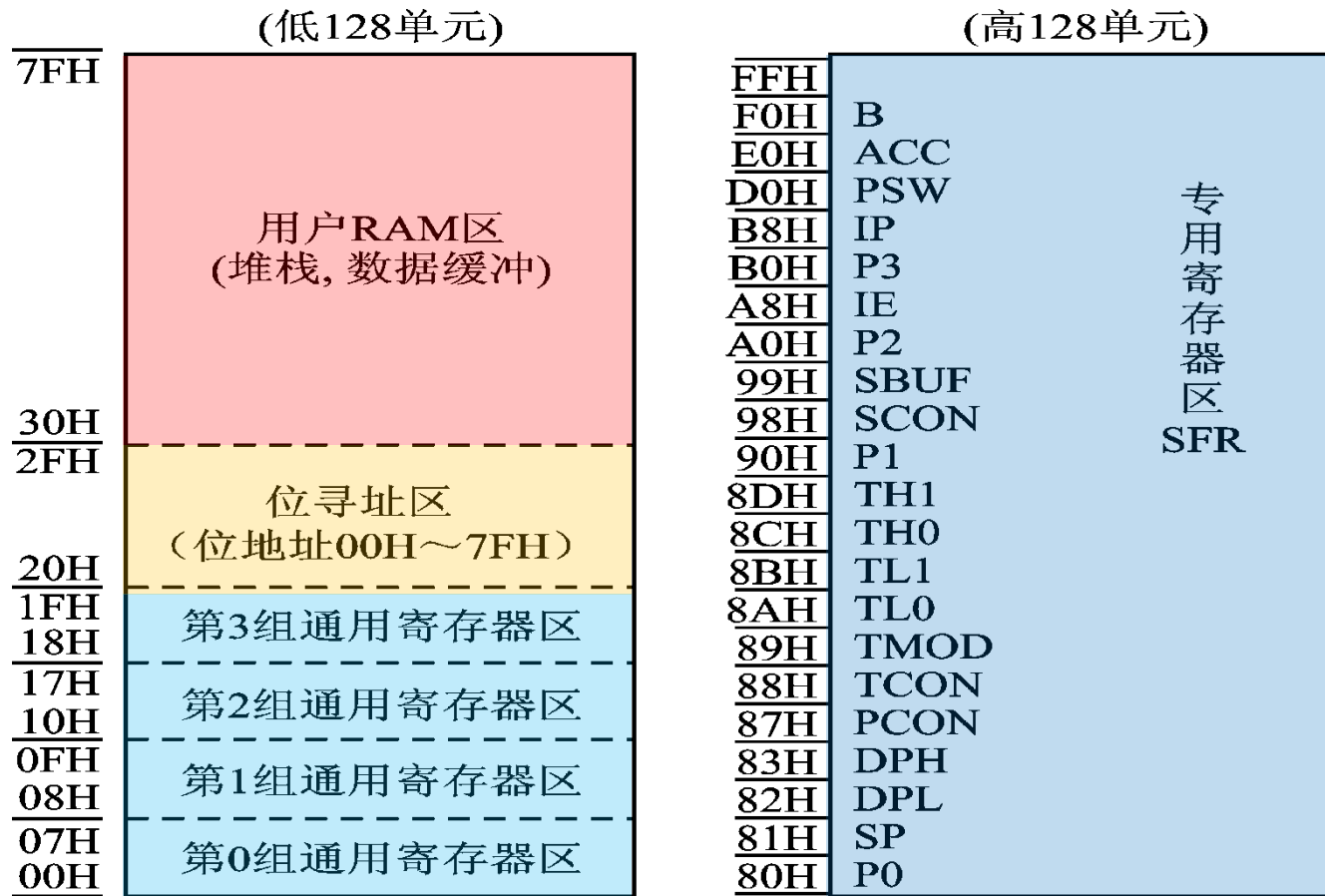
共256字节内部数据存储器，其中低128字节可以让用户使用，高128单元被专用寄存器（SFR）占用，不能让用户使用，所以常说的内部RAM区指的是低128单元；

##### 52子系列:

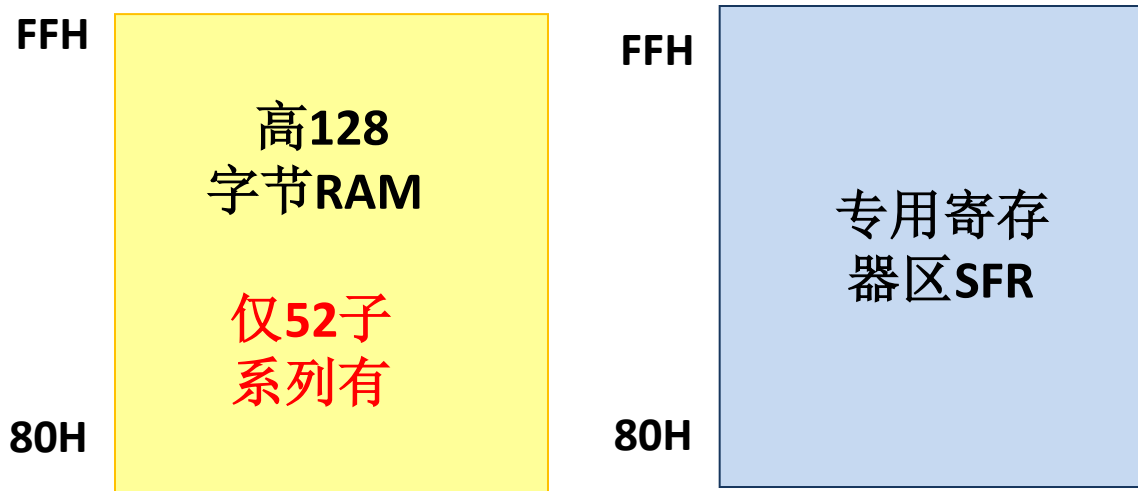
共有384字节内部数据存储器，低128字节RAM，高128字节RAM和128字节的专用寄存器区，常说的内部RAM为256字节。

对内部数据存储器的访问使用MOV指令。

# 51子系列内部数据存储器



## 52子系列内部数据存储器



NEXT





## 工作寄存器区的选择

内部RAM的00~1FH分为4个区，每个区有8个单元，分别用R0~R7来表示，选择哪个工作寄存器组是通过软件对程序状态字寄存器PSW的第3、4位进行设置实现的

程序状态字

位序	PSW.7	PSW.6	PSW.5	PSW.4	PSW.3	PSW.2	PSW.1	PSW.0
位标志	CY	AC	FO	RS1	RS0	OV	-	P

RS1	RS0	寄存器组	片内PAM地址	通用寄存器名称
0	0	0组	00H~07H	R0~R7
0	1	1组	08H~0FH	R0~R7
1	0	2组	10H~17H	R0~R7
1	1	3组	18H~1FH	R0~R7

### 3、位寻址区

位寻址: MOV 20H, C

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

字节寻址: MOV 20H,#30H

## 可位寻址的特殊功能寄存器

符号	单元地址	名称	位地址	
			符号	地址
* ACC	E0H	累加器	ACC.7~ACC.0	E7H~E0H
* B	F0H	乘法寄存器	B.7~B.0	F7H~F0H
* PSW	D0H	程序状态字	PSW.7~PSW.0	D7H~D0H
SP	81H	堆栈指针		
DPL	82H	数据存储器指针(低8		
DPH	83H	数据存储器指针(高8		
* IE	A8H	中断允许控制器	IE.7~IE.0	AFH~A8H
* IP	B8H	中断优先控制器	IP.7~IP.0	BFH~B8H
* P <sub>0</sub>	80H	通道0	P <sub>0.7</sub> ~P <sub>0.0</sub>	87H~80H
* P <sub>1</sub>	90H	通道1	P <sub>1.7</sub> ~P <sub>1.0</sub>	97H~90H
* P <sub>2</sub>	A0H	通道2	P <sub>2.7</sub> ~P <sub>2.0</sub>	A7H~A0H
* P <sub>3</sub>	B0H	通道3	P <sub>3.7</sub> ~P <sub>3.0</sub>	B7H~B0H
PCON	87H	电源控制及波特率选		
* SCON	98H	串行口控制		9FH~98H
SBUF	99H	串行数据缓冲器	SCON.7~SCON.0	
* TCON	88H	定时控制	CON.7~TCON.0	8FH~88H
TMOD	89H	定时器方式选择		
TL <sub>0</sub>	8AH	定时器0低8位		
TL <sub>1</sub>	8BH	定时器1低8位		
TH <sub>0</sub>	8CH	定时器0高8位		
TH <sub>1</sub>	8DH	定时器1高8位		

## 4、 数据缓冲区

数据缓冲区:内部RAM中30H-7FH为80个单元的数据缓冲区，这些单元只能按字节寻址。

## 二、 内部数据存储器高128单元(80H-0FFH)

22个专用寄存器离散分布在数据存储器高128字节中，因为存储器功能已专门规定，故称专用寄存器（SFR）。对没被专用寄存器使用的空闲地址的操作是无意义的

### 特殊功能寄存器SFR

- 1) 占用字节地址： 80H~FFH
- 2) 可位寻址寄存器 字节地址可被8整除
- 3) 专用寄存器：

A、B、PSW、DPTR、SP

I/O接口寄存器：

P0、P1、P2、P3、

SBUF、TMOD、TCON、SCON ...

F0H	F7	F6	F5	F4	F3	F2	F1	F0	B
E0H	E7	E6	E5	E4	E3	E2	E1	E0	ACC
D0H	D7	D6	D5	D4	D3	D2	D1	D0	PSW
C8H	CF	CE	CD	CC	CB	CA	C9	C8	T2CON
B8H	—	—	BD	BC	BB	BA	B9	B8	IP
B0H	B7	B6	B5	B4	B3	B2	B1	B0	P3
A8H	AF	—	AD	AC	AB	AA	A9	A8	IE
A0H	A7	A6	A5	A4	A3	A2	A1	A0	P2
98H	9F	9E	9D	9C	9B	9A	99	98	SCON
90H	97	96	95	94	93	92	91	90	P1
88H	8F	8E	8D	8C	8B	8A	89	88	TCON
80H	87	86	85	84	83	82	81	80	P0

# 1、相关专用寄存器简介

## 1) 累加器Acc (8位) A

需要ALU处理的数据和计算结果多数要经过A累加器，相当于十字路口的警察。累加器在算术运算中存放操作数和运算结果；在一些运算中作为源或目的操作数；在变址方式中作为变址寄存器。

**MOV A, #03**

**ADD A, #05H**

## 2) 寄存器B (8位)

与A累加器配合执行乘、除运算。也可用作通用寄存器。乘法时：B为乘数并存放乘积的高八位；除法时：B为除数并存放余数。

**MUL AB**

**DIV AB**

### 3) 程序状态字PSW(8位)

存放ALU运算过程的标志状态，程序分叉的判断依据，指令执行结果会自动影响到PSW的 **CY**、**AC**、**OV**、**P**位，可用指令设置的是**F0**、**RS1**、**RS0**位。

程序状态字								
位序	PSW.7	PSW.6	PSW.5	PSW.4	PSW.3	PSW.2	PSW.1	PSW.0
位标志	CY	AC	F0	RS1	RS0	OV	-	P

RS1	RS0	寄存器组	片内PAM地址	通用寄存器名称
0	0	0组	00H~07H	R0~R7
0	1	1组	08H~0FH	R0~R7
1	0	2组	10H~17H	R0~R7
1	1	3组	18H~1FH	R0~R7



## CY、AC、RS1、RS0、OV定义与举例

设程序执行前, F0=0 RS1,RS0=00,则执行下述程序后PSW  
的各位状态?

MOV A, #0FH

ADD A, #0F8H

	0 0 0 0	1 1 1 1	CY =1(A7有进位)	CS=1(A6有进位)	OV=CY⊕CS=1⊕1=0
+	1 1 1 1	1 0 0 0	AC=1(A3有进位),	F 0=0 , RS1,RS0=00	
<hr/>					
1	0 0 0 0	0 1 1 1	P =1(数据位有3个1, 不算进位的和1)		
CY	CS				

CY AC F0 RS1 RS0 OV -- P

执行后PSW各位的状态为: 1 1 0 0 0 0 x 1 B

用十六进制表示为: 0C1H或0C3H

#### 4) 数据指针DPTR(16位)

由高位字节DPH和低位字节DPL组成，主要用于存放片外存储器16位地址，作为片外存储器的指针。既可作为一个16位寄存器来使用，也可作为2个独立的8位寄存器DPH（DPTR高8位字节）和DPL（DPTR低8位字节）来使用。

名称	位数	范围
DPH:	8bit	0----255
DPL:	8bit	0----255
DPTR:	16bit	0----65535

## 2、专用寄存器的字节寻址和位寻址

如下图所示：\*可位寻址的特殊功能寄存器，51系列单片机，共有83个专用寄存器，但可位寻址的只有211位，无\*的只能字节寻址。

## 可位寻址的特殊功能寄存器

符号	单元地址	名称	位地址	
			符号	地址
* ACC	E0H	累加器	ACC.7~ACC.0	E7H~E0H
* B	F0H	乘法寄存器	B.7~B.0	F7H~F0H
* PSW	D0H	程序状态字	PSW.7~PSW.0	D7H~D0H
SP	81H	堆栈指针		
DPL	82H	数据存储器指针(低8		
DPH	83H	数据存储器指针(高8		
* IE	A8H	中断允许控制器	IE.7~IE.0	AFH~A8H
* IP	B8H	中断优先控制器	IP.7~IP.0	BFH~B8H
* P <sub>0</sub>	80H	通道0	P <sub>0.7</sub> ~P <sub>0.0</sub>	87H~80H
* P <sub>1</sub>	90H	通道1	P <sub>1.7</sub> ~P <sub>1.0</sub>	97H~90H
* P <sub>2</sub>	A0H	通道2	P <sub>2.7</sub> ~P <sub>2.0</sub>	A7H~A0H
* P <sub>3</sub>	B0H	通道3	P <sub>3.7</sub> ~P <sub>3.0</sub>	B7H~B0H
PCON	87H	电源控制及波特率选		
* SCON	98H	串行口控制		9FH~98H
SBUF	99H	串行数据缓冲器	SCON.7~SCON.0	
* TCON	88H	定时控制	CON.7~TCON.0	8FH~88H
TMOD	89H	定时器方式选择		
TL <sub>0</sub>	8AH	定时器0低8位		
TL <sub>1</sub>	8BH	定时器1低8位		
TH <sub>0</sub>	8CH	定时器0高8位		
TH <sub>1</sub>	8DH	定时器1高8位		

### 3、程序计数器PC (16位)

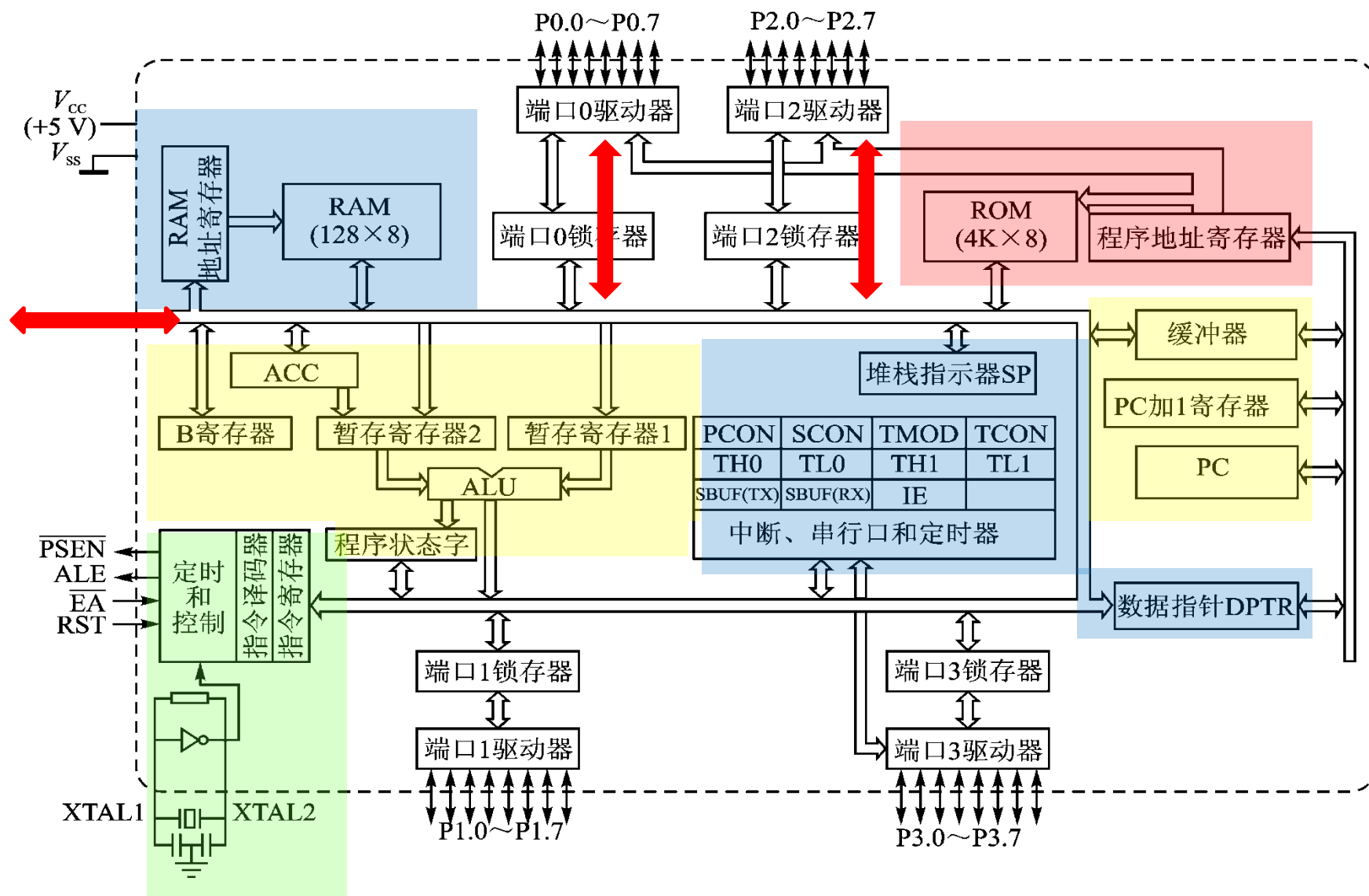
将要执行的指令地址。PC是唯一不可寻址的寄存器，用户不可对其进行读写。在执行转移、调用、返回指令时能自动改变其内容，以改变程序的执行顺序。

**复位时：** PC=0000H，程序从此单元开始执行，  
**访问范围：** 0000~FFFFH即64K。

## 作业：

- 1、简述MCS-51的低128字节RAM的分区情况和各自特点。
- 2、80C51单片机的PSW寄存器各位标志的意义如何？

# 复习1 单片机芯片内逻辑结构图和总线扩展

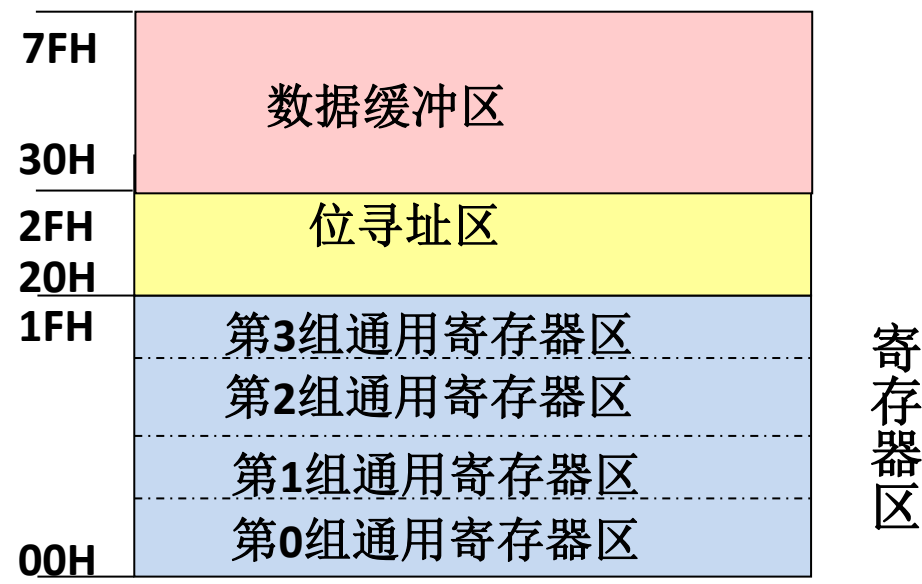
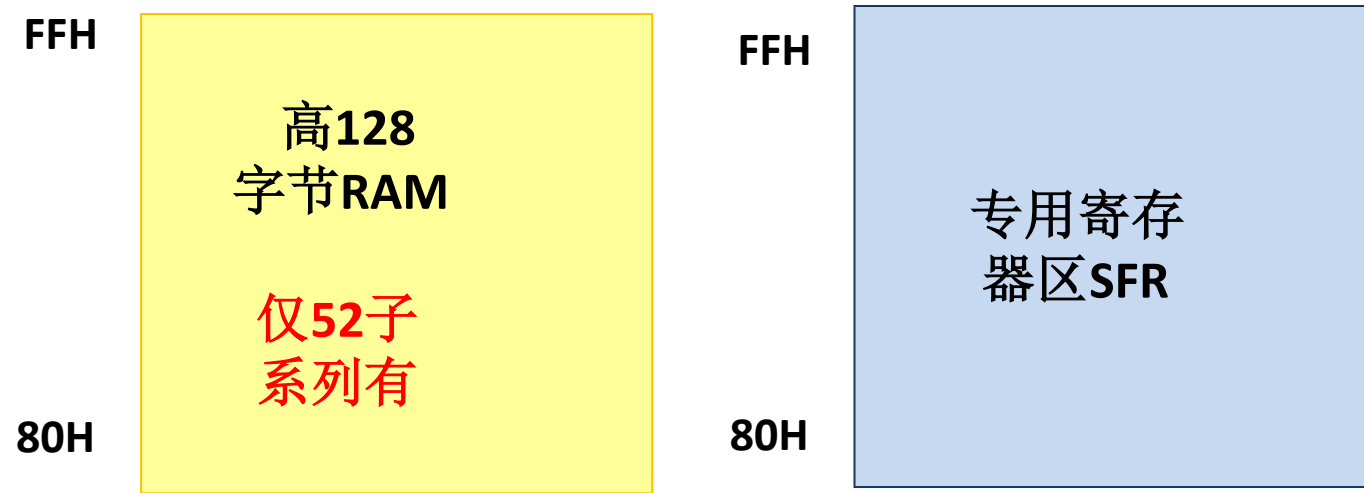


## 复习2： 控制引脚说明：

- 1 地址锁存控制信号 ALE：**访问片外ROM, RAM时，ALE用于控制把P0口输出的低8位地址送入锁存器锁存，实现低位地址和数据的分时传送。不访问片外存储器时，可做为外部时钟使用。
- 2 外部程序存储器读选通信号  $\overline{\text{PSEN}}$ ：**CPU访问片外ROM时，使PSEN低电平有效，可实现片外ROM的读操作，其他情况下此引脚为高电平封锁状态。
- 3 访问程序存储器控制信号  $\overline{\text{EA}}$ ：**当EA信号为低电平时，对ROM的读操作是针对外部ROM的，当EA信号为高电平时，对ROM的操作是从内部ROM开始，并可延续至外部ROM。
- 4 复位信号 RST：**复位即初始化，分自动上电复位和人工开关复位，复位时，产生一个延续2个机器周期以上的高电平为有效，实现单片机的复位操作。



# 复习3 内部数据存储器地址分配



## 复习4 程序状态字PSW(8位)

存放ALU运算过程的标志状态，用来存放程序状态信息，而且某些指令的执行结果会自动影响到PSW的有关标志位，有些标志位可用指令来设置。

程序状态字

位序	PSW.7	PSW.6	PSW.5	PSW.4	PSW.3	PSW.2	PSW.1	PSW.0
位标志	CY	AC	FO	RS1	RS0	OV	-	P

RS <sub>1</sub>	RS <sub>0</sub>	寄存器组	R <sub>0</sub> ~R <sub>7</sub> 地址
0	0	组 0	00~07H
0	1	组 1	08~0FH
1	0	组 2	10~17H
1	1	组 3	18~1FH

## 三、堆栈操作

### 1. 堆栈的功能

主要是为子程序调用和中断时保护断点和现场而设立，  
保护断点：存储16位的PC值，单片机自动将PC存RAM。  
保护现场：存储指定的寄存器的值，人工手动执行。

### 2. 堆栈的开辟

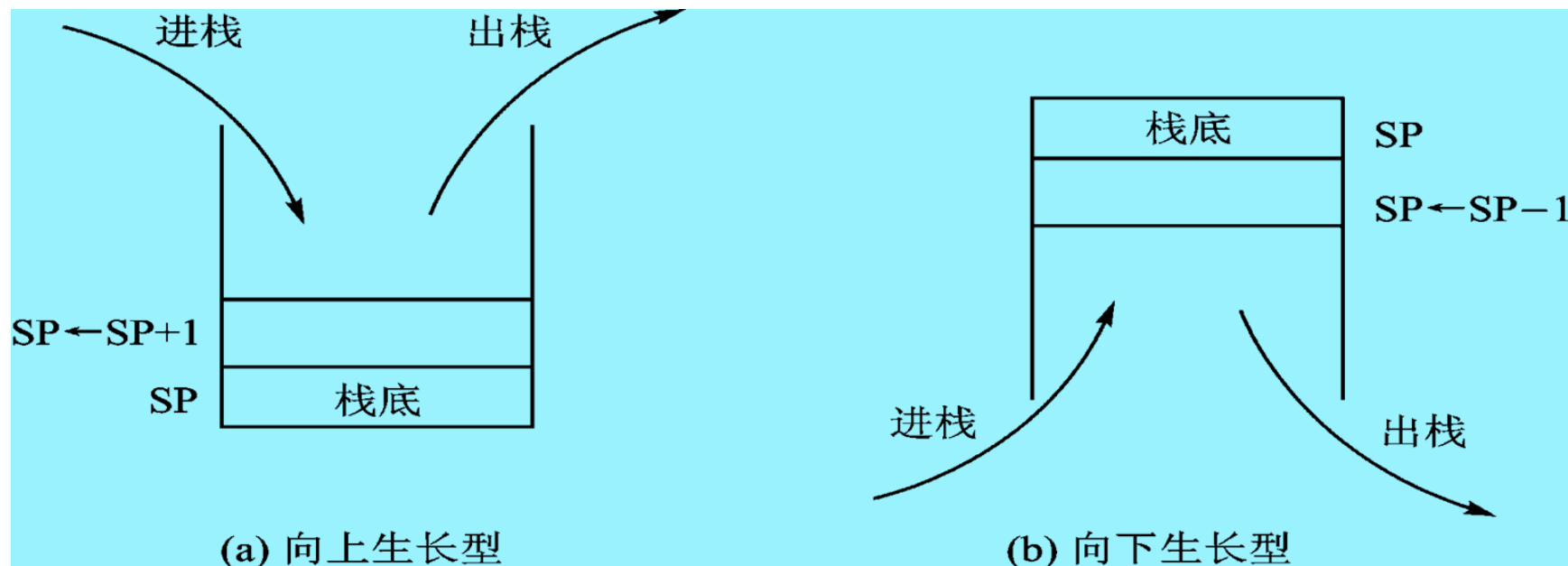
只能开辟在芯片的内部8位数据存储RAM中, 速度快。

### 3. 堆栈指针

堆栈有两种操作：进栈和出栈，都是对栈顶单元进行的，  
SP的内容就是堆栈栈顶的存储单元地址，指示栈顶位置。  
复位时 **SP=07H**，从**08H**单元堆。

## 4. 堆栈类型

### 两种不同类型的堆栈结构



- 1) 两种类型：向上生长型和向下生长型。
- 2) 51使用向上生长型堆栈，操作规则如下：  
进栈操作：先SP加1，后写入数据。  
出栈操作：先读出数据v，后SP减1。 均为先进后出。

## 5. 堆栈使用方式

两种：自动方式和指令方式。

- 1) **自动方式**：在调用子程序或中断时，断点地址自动进栈。程序返回时，断点地址再自动弹回PC。
- 2) **指令方式**：使用专用的堆栈操作指令，进行进出栈操作。进栈指令为PUSH，出栈指令为POP。

**例：**

MOV	A	,	#35H	
MOV	SP	,	#30H	
PUSH	ACC			; SP+1 → SP    #35H → (SP)
...	...	...	...	; 指令执行后 (31H) = 35H, SP = 31H
POP	ACC			; (SP) → A,    SP-1 → SP

指令执行后    A=35H, SP=30H

## 四、外部数据存储器

外部数据存储器一般由静态RAM构成，其容量大小由用户根据需要而定，由DPTR作为指针，高8位地址DPH通过P2口输出，低8位DPL地址通过P0口分时共用输出，共16位最大可扩展到 64 KBRAM，地址是 0000H~0FFFFH。

对外部数据存储器的访问使用MOVX指令

## 五、程序存储器

EA接高，从片内程序存储单元读起，如扩展了可延伸到片外  
EA接低，直接从片外程序存储单元读

片内外共有64KROM空间，地址0000H~0FFFFH  
PC作指针，其中特殊区间0000H-002AH功能如下：

- 0000H—0002H 系统的启动单元，单片机复位后，  
(PC) = 0000H，程序从0000H单元开始取指令。
- 0003H—000AH 外部中断0中断地址区。
- 000BH—0012H 定时/计数器0中断地址区。
- 0013H—001AH 外部中断1中断地址区。
- 001BH—0022H 定时/计数器1中断地址区。
- 0023H—002AH 串行中断地址区。

0FFFFH

程序  
区

0FFFH

002AH

引导  
区

0000H

对程序存储器的访问都是用MOV C指令

# 51单片机存储空间总结

**MCS-51在物理上有四个存储空间：**

- 1、片内程序存储器
- 2、片外程序存储器、
- 3、片内数据存储器
- 4、片外数据存储器。

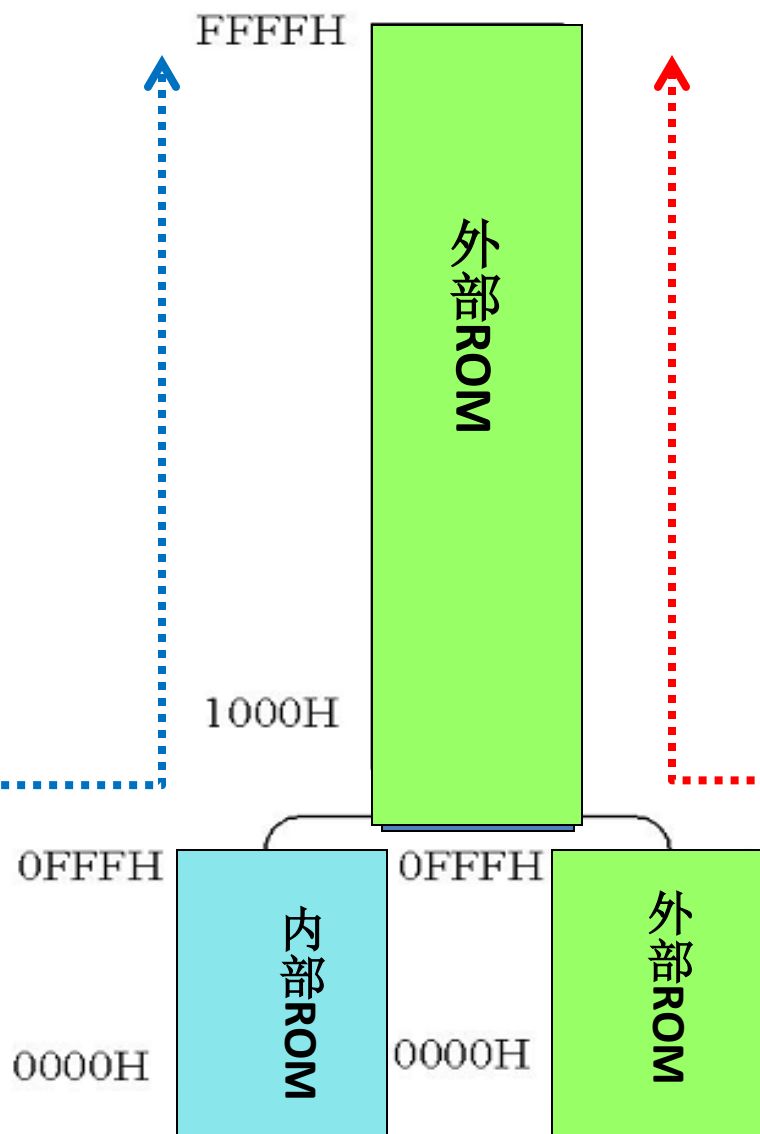
**从逻辑上划分有三个存储器地址空间：**

- 1、片内外统一编址的**64K**字节程序存储器，**PC**作为指针  
(**0000H~0FFFFH**)
- 2、内部**256**字节数据存储器地址空间  
(包括 **128**字节片内**RAM**和**128**字节的**SFR**)
- 3、外部**64K**字节数据存储器地址空间，**DPTR**作为指针  
(**0000H~0FFFFH**)



使用MOVX语句

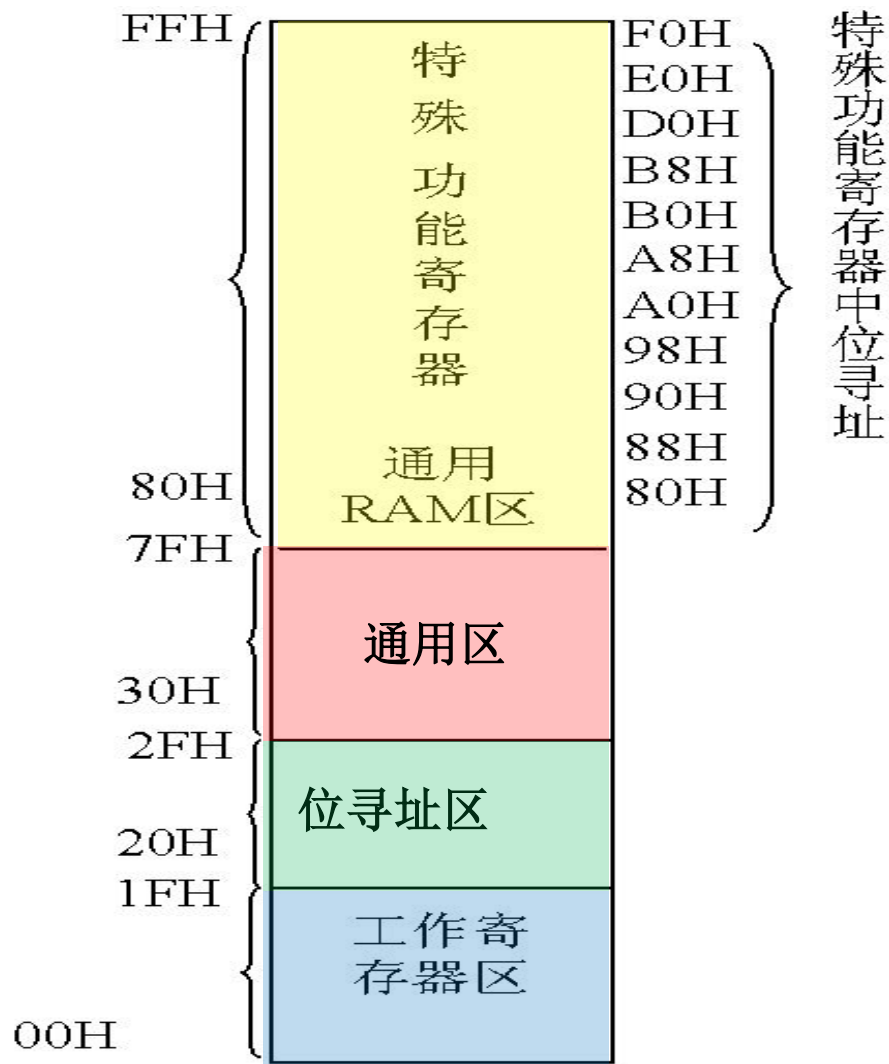
$\overline{EA}=1$



$\overline{EA}=0$

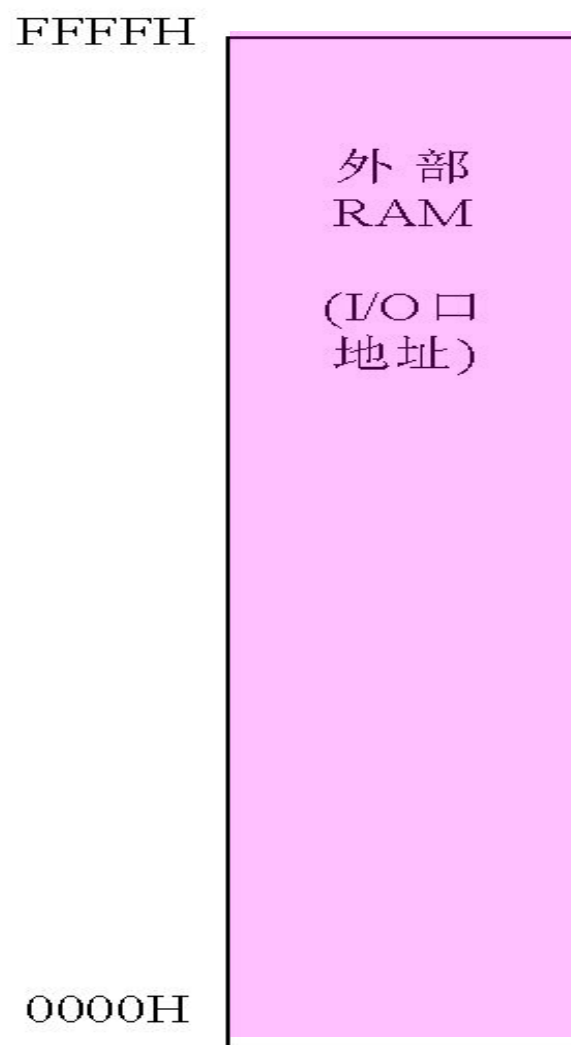
程序存储空间

## 使用MOV语句



内部数据存储器

## 使用MOVX语句



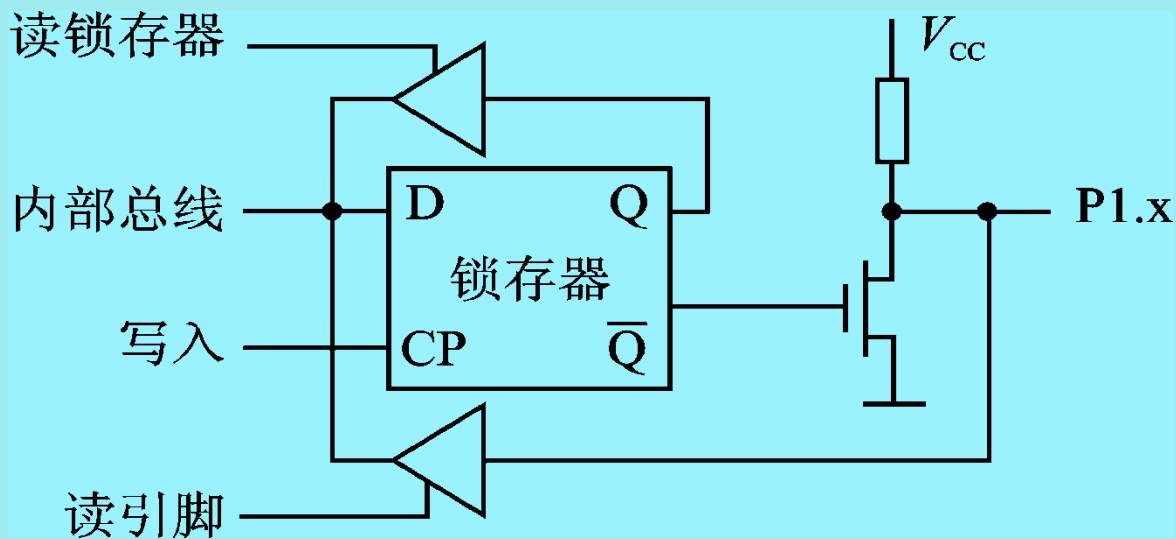
外部数据存储器

## 2.3 并行输入/输出端口结构（I/O口）

四个双向8位I/O口，共32根I/O口线，每个I/O线均由锁存器，输出电路和输入缓冲器组成。每个I/O既可作输入又可作输出，每一条口线可独立用作输入又可用作输出，作输出时可锁存数据，作输入时可缓冲数据。

## 1、P1口逻辑结构

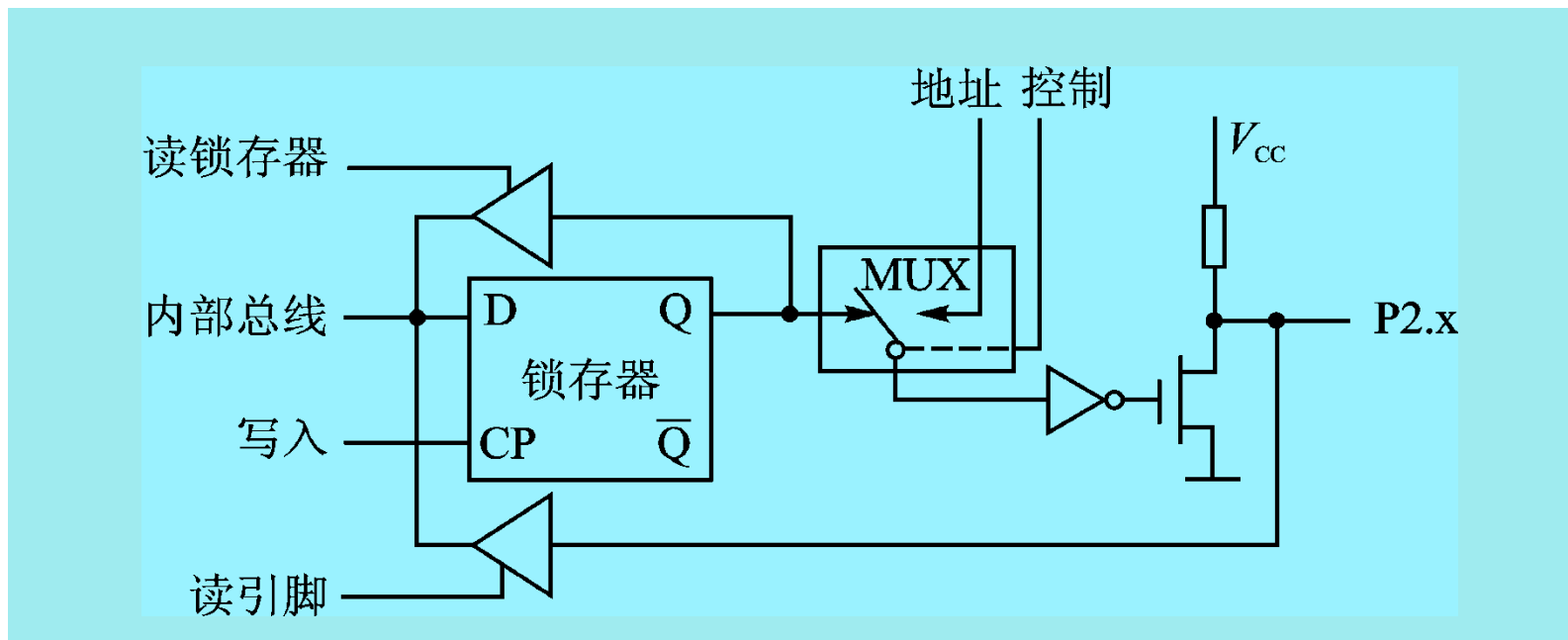
P1口地址为90H，位地址为90H-97H。只能作为通用数据I/O口使用，电路逻辑如下图：



- 1) 注意读引脚和读锁存器的区别。
- 2) 普通IO输入时，必须把P1相应位置1，使FET关断，引脚悬浮成为高阻，才能正确输入数据，称为准双向口。

## 2、P2口逻辑结构

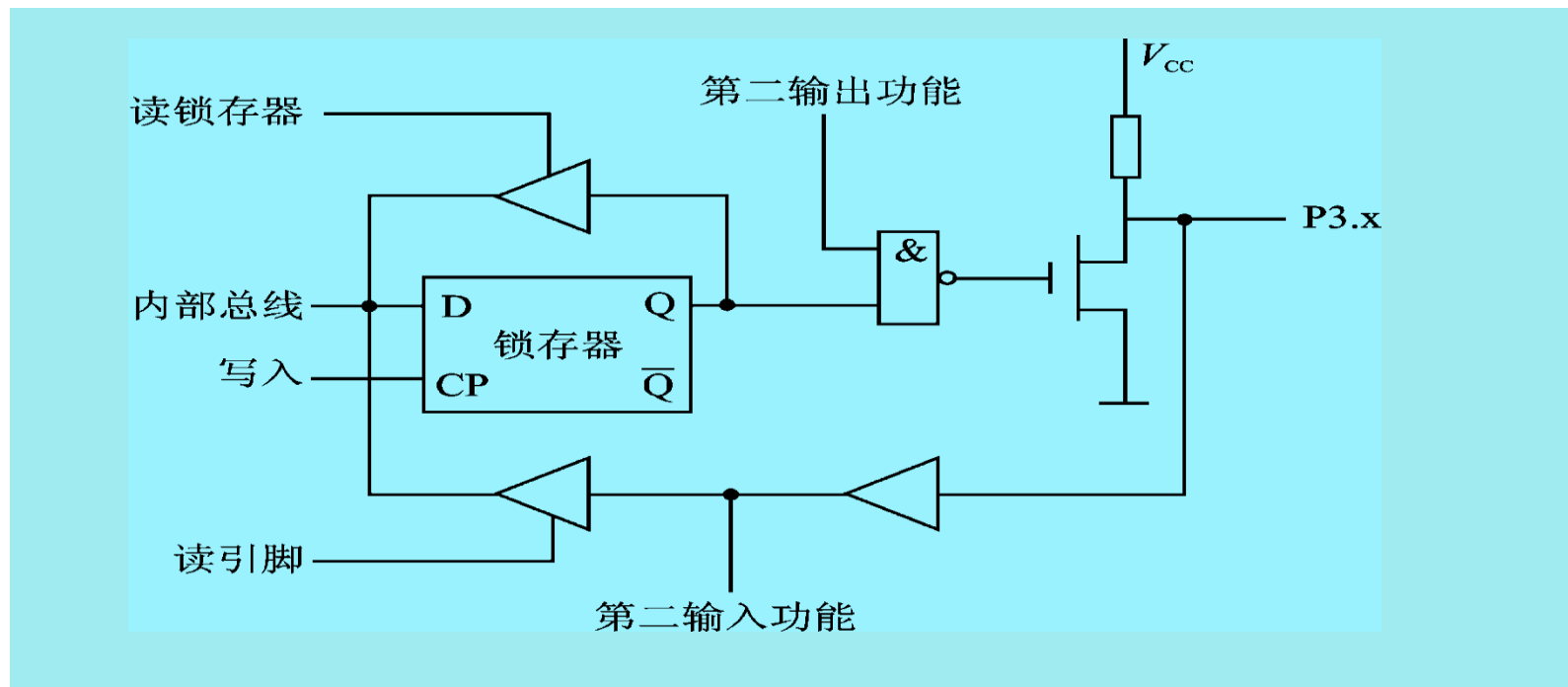
地址A0H，位地址A0H-A7H。可做通用I/O口，又可作高位地址



- 1) 普通IO输入时，必须把P2相应位置1，使FET关断，使引脚悬浮成为高阻，才能正确输入数据，称为准双向口。
- 2) MUX当程序中出现MOVX A, @DPTR (16bit) 语句时自动切向右方，输出高8位地址AB8~15；而执行MOVX A, @Ri (8bit)，时MUX接左边，仍作为普通IO口使用。

### 3、P3口逻辑结构

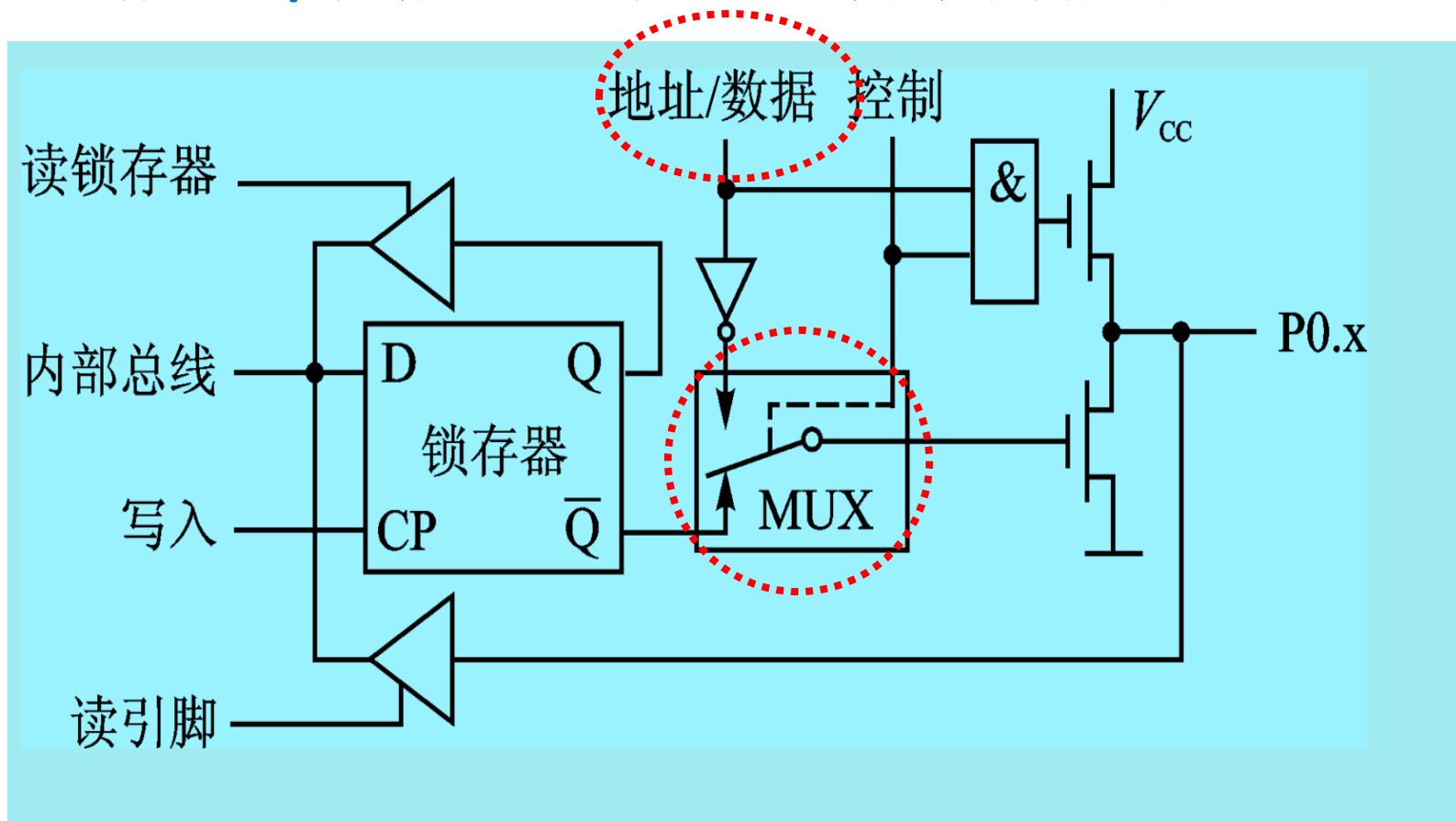
地址B0H，位地址B0H-B7H。主要做第二功能口。



- 1) 普通IO输入时，必须把P3相应位置1，使FET关断，使引脚悬浮成高阻，才能正确输入数据，称为准双向口。
- 2) 程序中使用第二功能语句时相应端口自动启动，此时应将第二功能口Q端置1，使FET仅受第二功能口控制。

## 4、P0口逻辑结构

P0口地址为80H，位地址为80H-87H。既可做一般I/O口，又可作地址/数据总线，某一位的内部结构如下图



## P0口特点

- 1) 注意读引脚和读锁存器的区别。
- 2) 输出时接上拉电阻。
- 3) P0口作输入时，应使输出两个驱动场效应管截止而呈现悬浮的高阻态，所以需先向锁存器写“1”，才能正确输入数据，称为准双向口。
- 4) 作地址（AB）和数据总线（DB）扩展时，分时输出A0~A7 和D0~D7, MUX当程序中出现MOVX语句时自动切向上方，扩展使用作为D0~D7 / A0~A7；复位后，不出现MOVX语句MUX自动切向下方则作为普通IO口。



## 5、I/O口电路小结

- 1) P0口：通用双向I/O口，或低8位地址/数据传送分时复用。
- 2) P1口：最简单的I/O通用双向接口。
- 3) P2口：通用双向I/O接口，或高8位地址总线AB8~15
- 4) P3口：通用双向并行接口和第二功能口：

串行接口引脚： TXD、RXD

中断输入引脚： INT0、INT1

定时器输入引脚： T0、 T1

读写控制线： RD、 WR

## 5) 每个I/O口均有两种读入方式（用命令区分）

读锁存器

**ORL P1, #0FH; 读锁存器**

（目的操作数必须是一个I/O口或I/O口的某一位）

读引脚

**MOV A, P1; 读引脚**

（源操作数是一个I/O口或I/O口的某一位）

**注意：** 由于是准双向口，读引脚时，需先向锁存器写“1”。  
系统复位时，所有口锁存器均置“1”。

## 作业:

- 1、80C51单片机的存储器地址空间如何划分？各地址空间的地址范围和容量是多少？分别用什么语句操作？
- 2、80C51单片机的P0~P3口在结构上有何不同？在使用上有何要求？

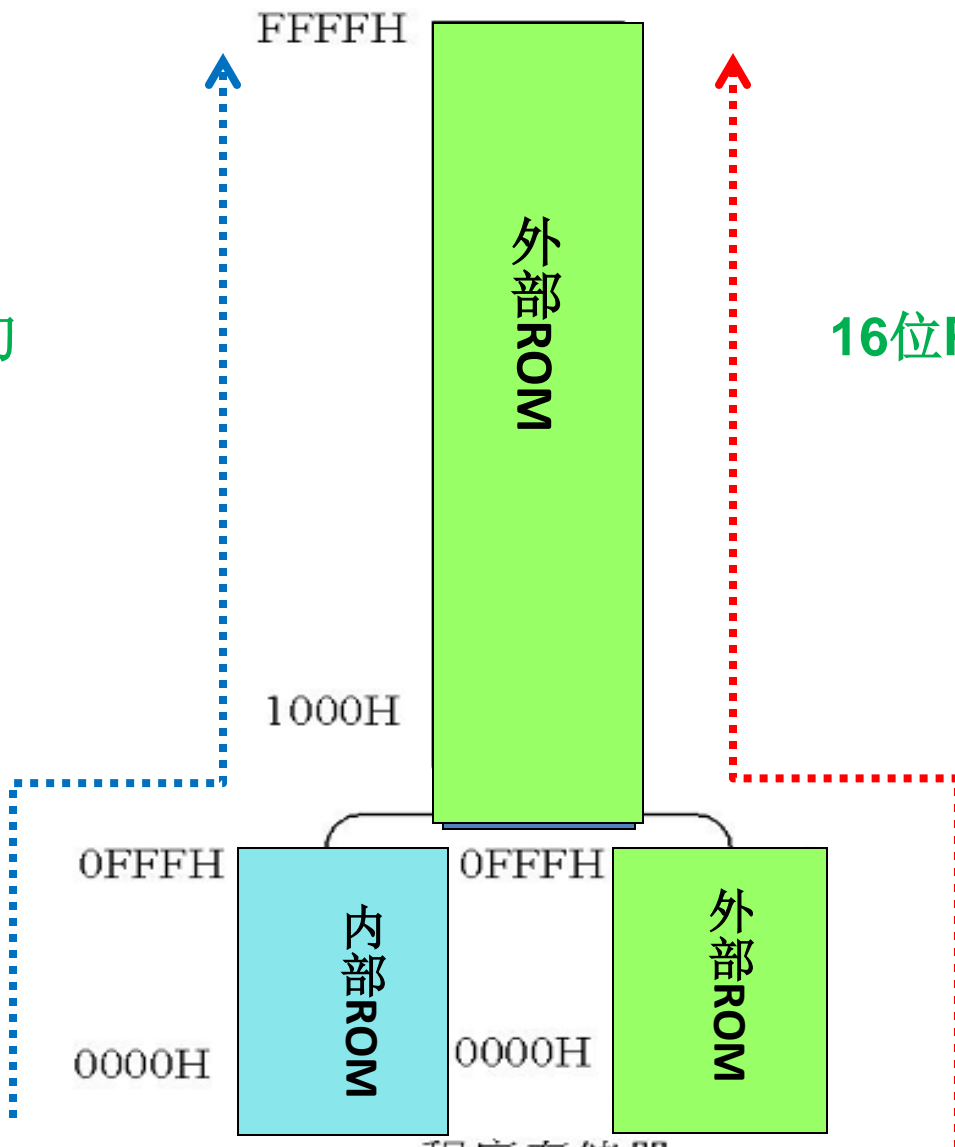
# 复习1

使用MOV<sub>VC</sub>语句

16位PC做指针

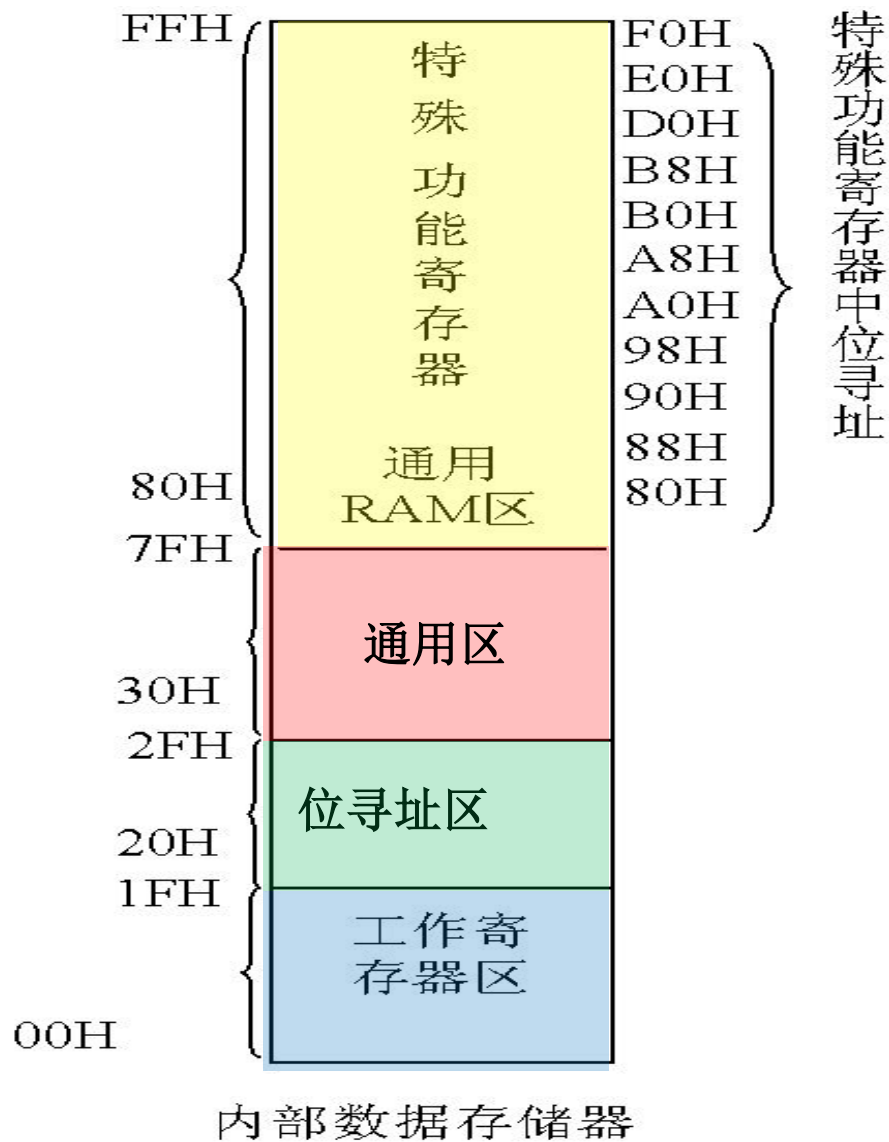
**/EA=1**

**/EA=0**

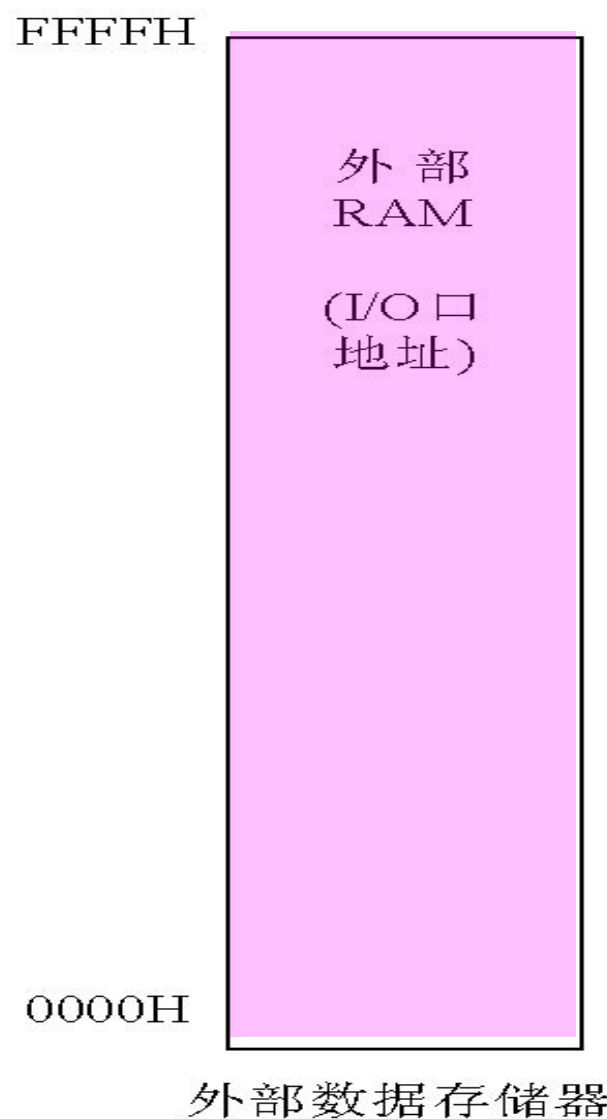


程序存储空间

## 复习2 使用MOV语句



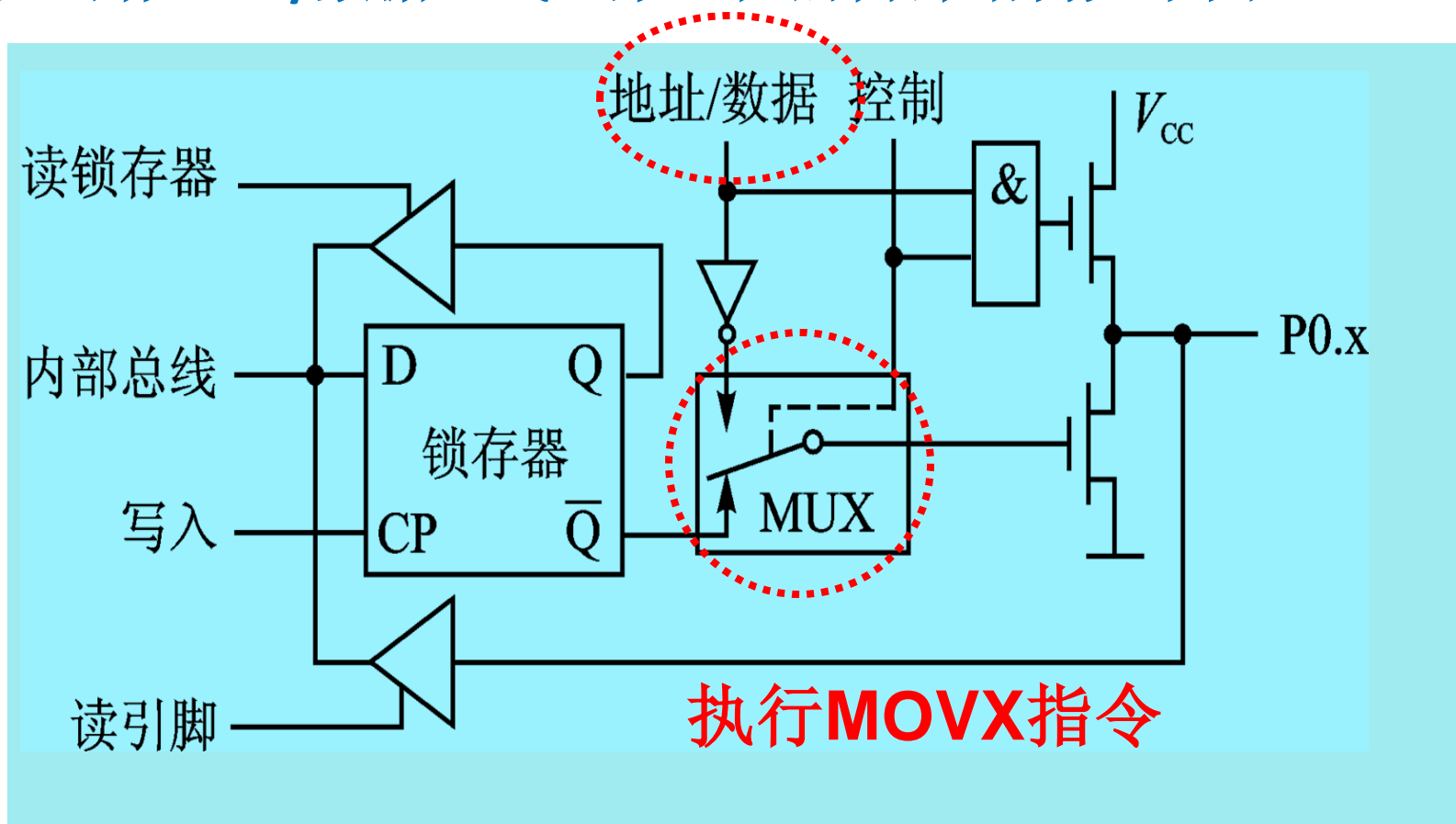
## 16位DPTR做指针 使用MOVX语句



## 复习3

### IO口逻辑结构

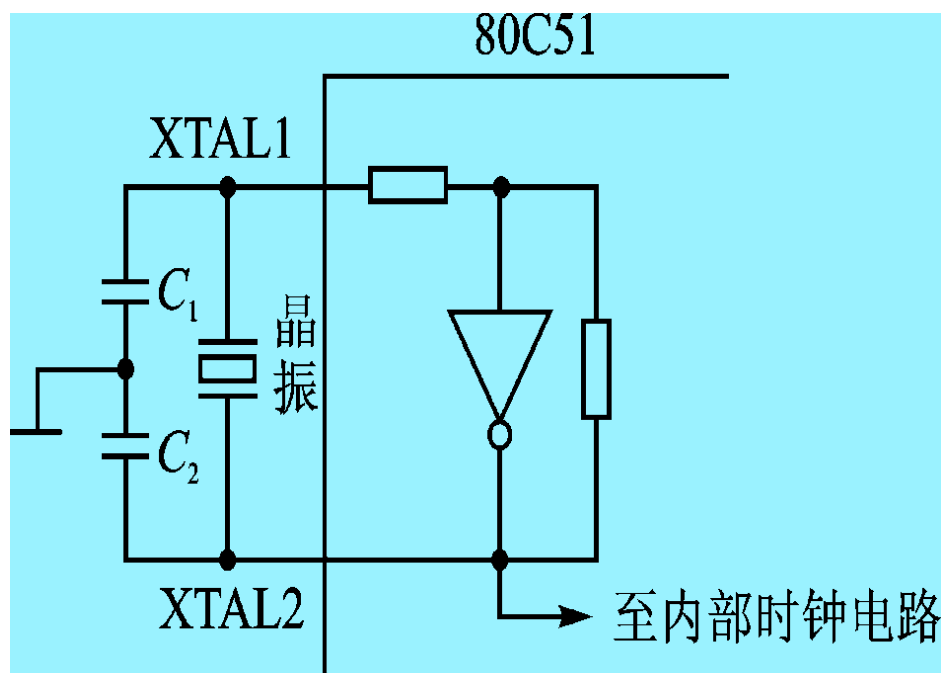
P0口地址为80H，位地址为80H-87H。既可做一般I/O口，又可作地址/数据总线，某一位的内部结构如下图



## 2.4 时钟电路及时序

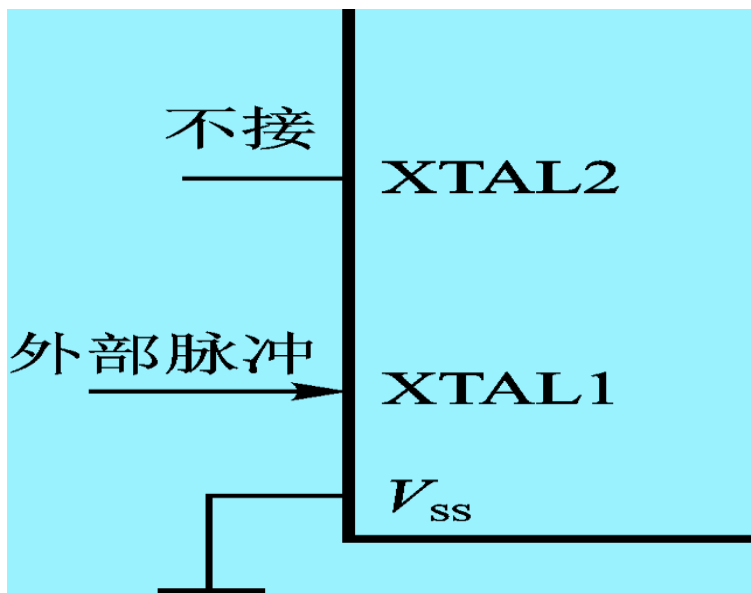
### 1、时钟电路

- 1) **内部振荡方式：** 引脚 $\text{XTAL}_1$ 和 $\text{XTAL}_2$ 分别接晶体振荡器或陶瓷谐振器，以及两个电容，就构成了内部自激振荡器。晶体振荡器频率即为系统的振荡频率。



通常为1.2MHz-38MHz，  
晶振频率越高，单片机  
运行速度就越快。

2) **外部振荡方式**：把外部已有的时钟信号引入单片机内，  
外部振荡方式常用于多块芯片同时工作，以便于同步。

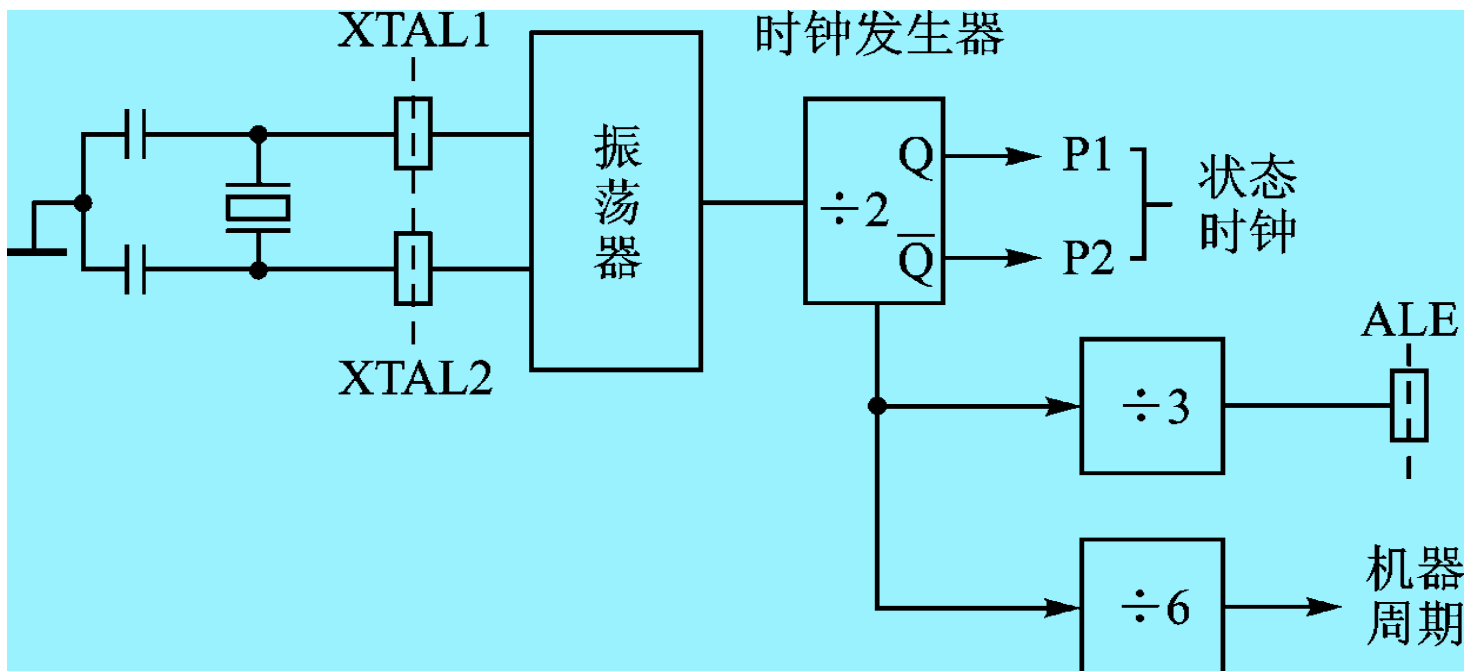


**HMOS型外部振荡方式**



## 2. 分频电路

振荡电路产生的振荡信号并不直接为单片机所用，进行分频，以得到单片机各种相关的时钟信号。

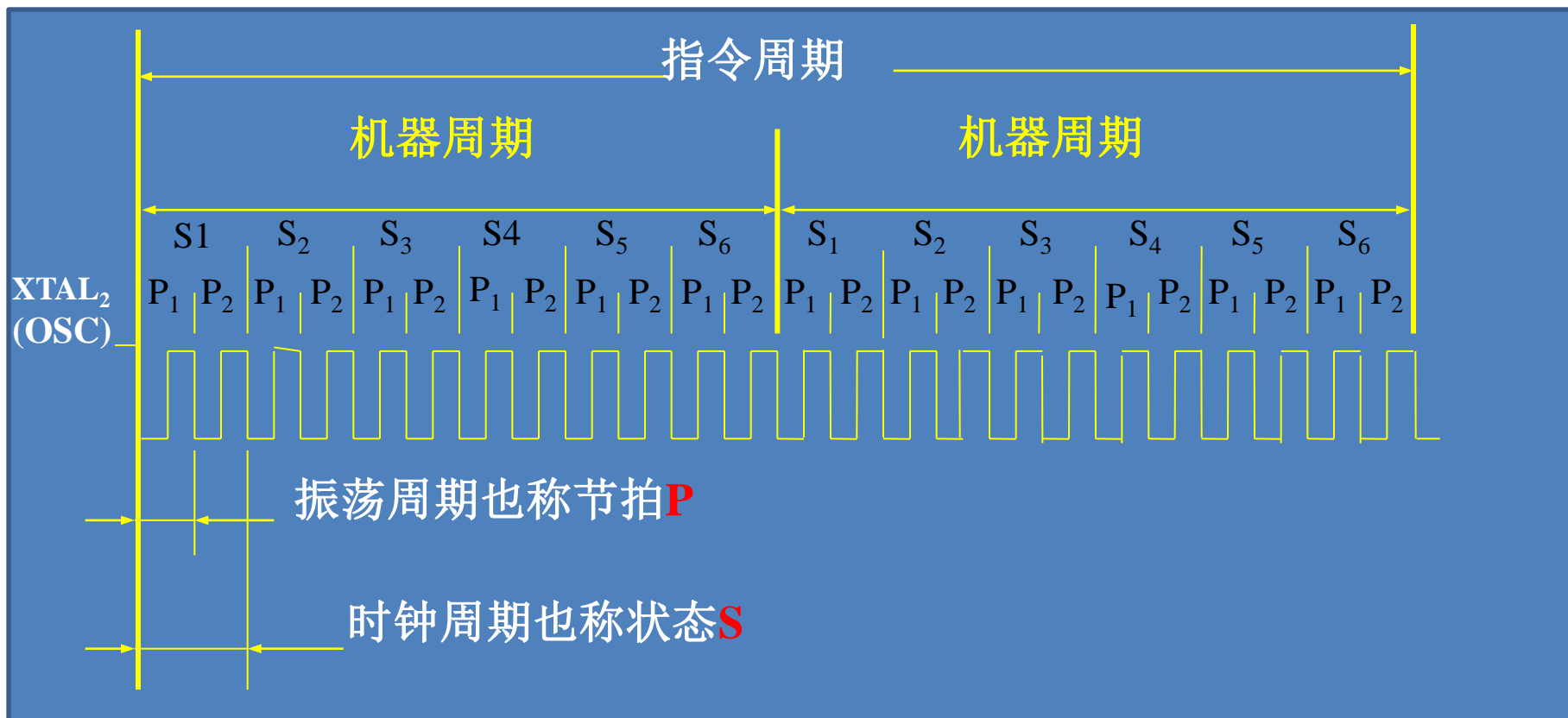


### 3、时序

时序研究的是指令执行中各信号之间的相互时间关系。下面是有关CPU时序的几个概念。

- 1) **振荡周期：** 振荡源的周期。也称拍节，用 $P$ 表示，如果为内部时钟方式，振荡周期即为石英晶体的振荡周期。
- 2) **时钟周期：** 是振荡源信号二分频，也称状态周期用 $S$ 表示，也就是一个时钟周期是振荡周期的2倍。
- 3) **机器周期：** 指完成一个基本操作所需时间，一个机器周期含6个时钟周期。
- 4) **指令周期：** 指CPU执行一条指令所需要的时间，通常含有1~4个机器周期。

## 振荡周期、时钟周期、机器周期、指令周期



若MCS-51单片机外接晶振为12MHz时，则单片机的四个周期的具体值为：

振荡周期 =  $1/12\text{MHz} = 1/12\mu\text{s} = 0.0833\mu\text{s}$       时钟周期 =  $1/6\mu\text{s} = 0.167\mu\text{s}$

机器周期 =  $1\mu\text{s}$

指令周期 =  $1 \sim 4\mu\text{s}$

OSC  
XTAL2

ALE

读操作码

04H

读下一个操作码(无效)

读下一指令

1字节

1周期指令

INC A

S1 S2 S3 S4 S5 S6

读操作码

24H

读第二字节

35H

读下一指令

2字节

1周期指令

S1 S2 S3 S4 S5 S6

读操作码

A3H

读下一个操作码(无效)

再读下一指令

1字节

2周期指令

INC DPTR

S1 S2 S3 S4 S5 S6 S1 S2 S3 S4 S5 S6

读操作码

E0H

读下一个操作码(无效)

无取指无ALE

无取指

再读下一指令

1字节

2周期指令

S1 S2 S3 S4 S5 S6 S1 S2 S3 S4 S5 S6

时序

MOVX A,@DPTR

## 2.5 单片机的复位

### 1、单片机为什么需要复位：

复位是单片机的初始化操作，使CPU和各个部件处在一个确定的初始状态，并从这个状态由用户程序开始引导工作。

### 2、什么时候需要复位：

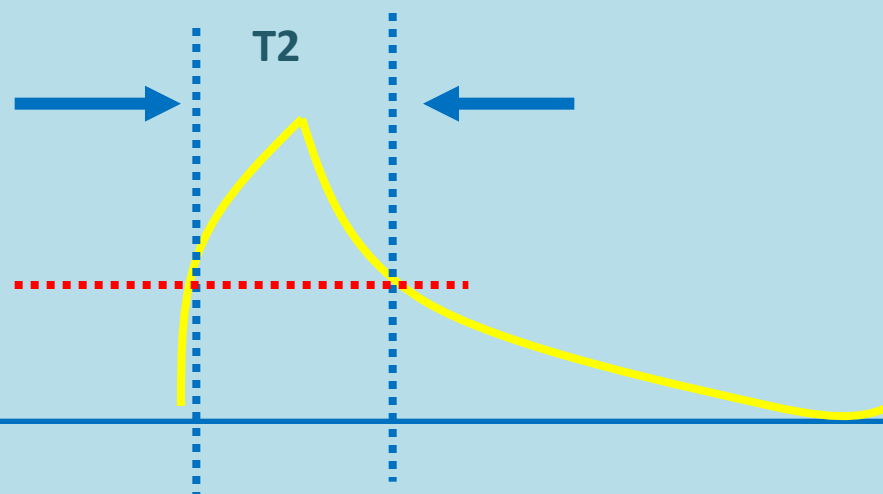
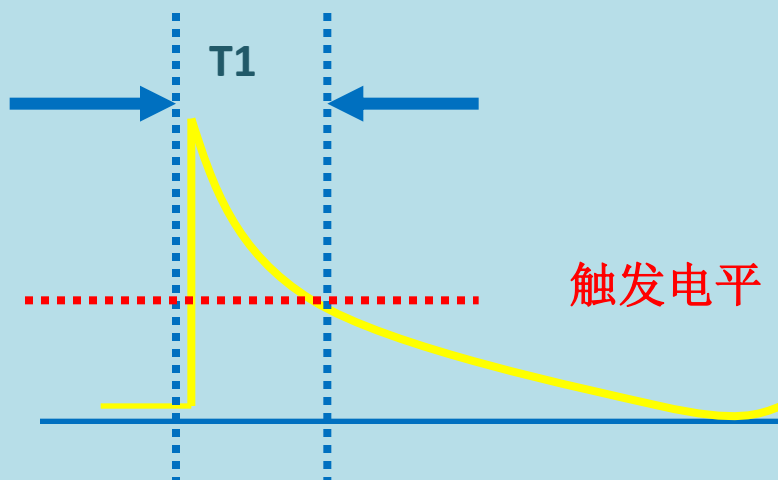
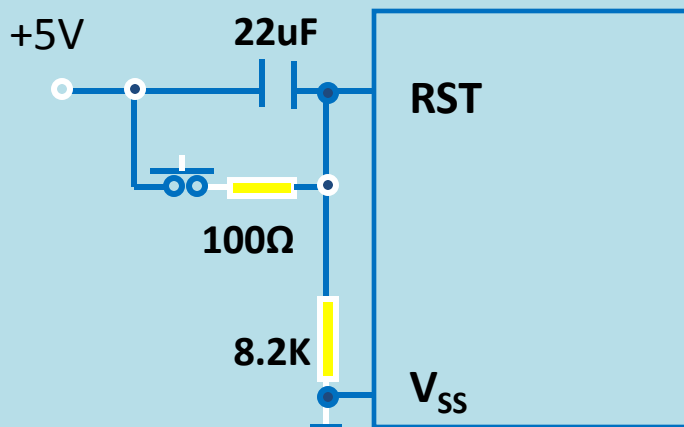
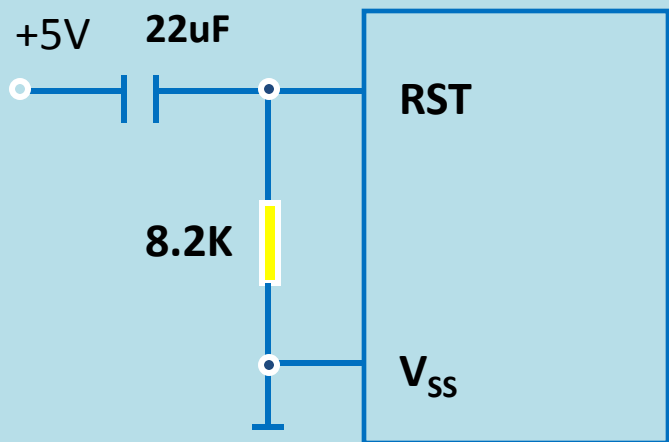
冷启动——当系统初始加电的时候——上电复位

热启动——当程序运行死机的时候——按钮复位

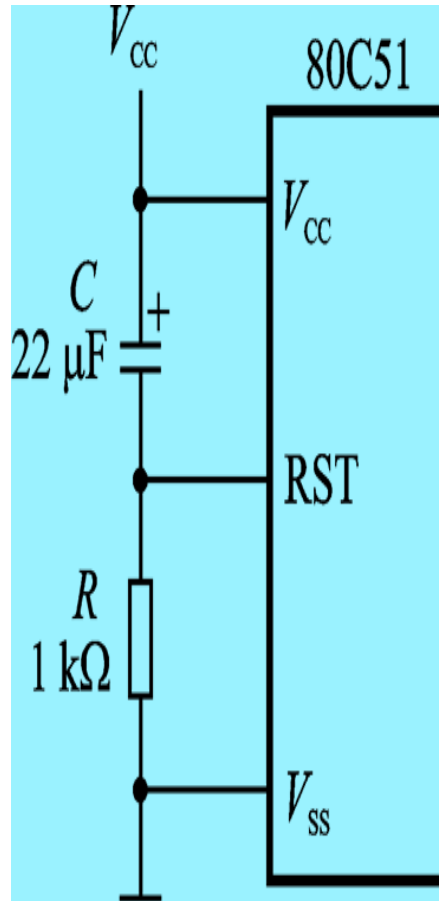
### 3、单片机复位的条件：

在振荡器运行时，RST端至少保持2个机器周期的高电平。

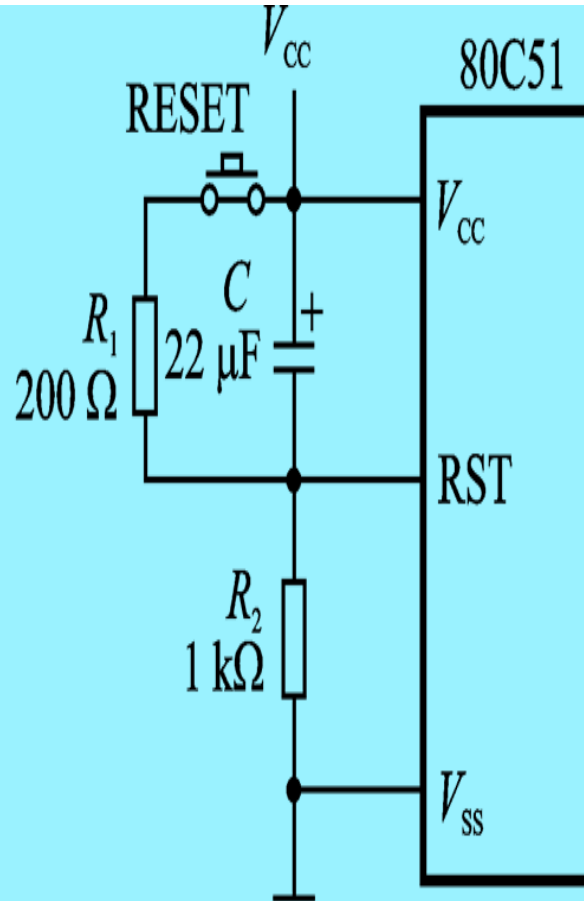
## 4、实现单片机复位的具体电路（复位方式）



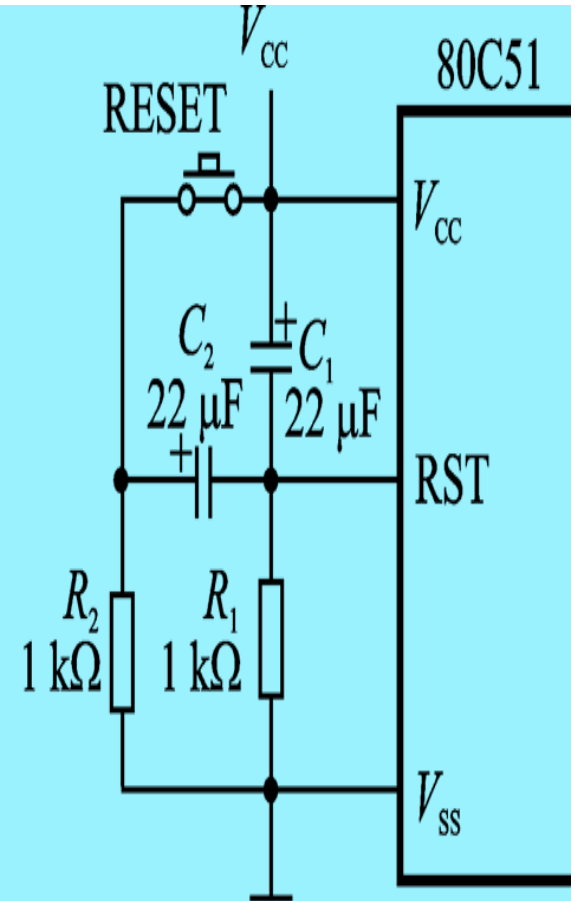
## 80C51 基本复位电路



(a) 上电复位



(b) 按键电平复位



(c) 按键脉冲复位



## 5、单片机复位后的初始状态:

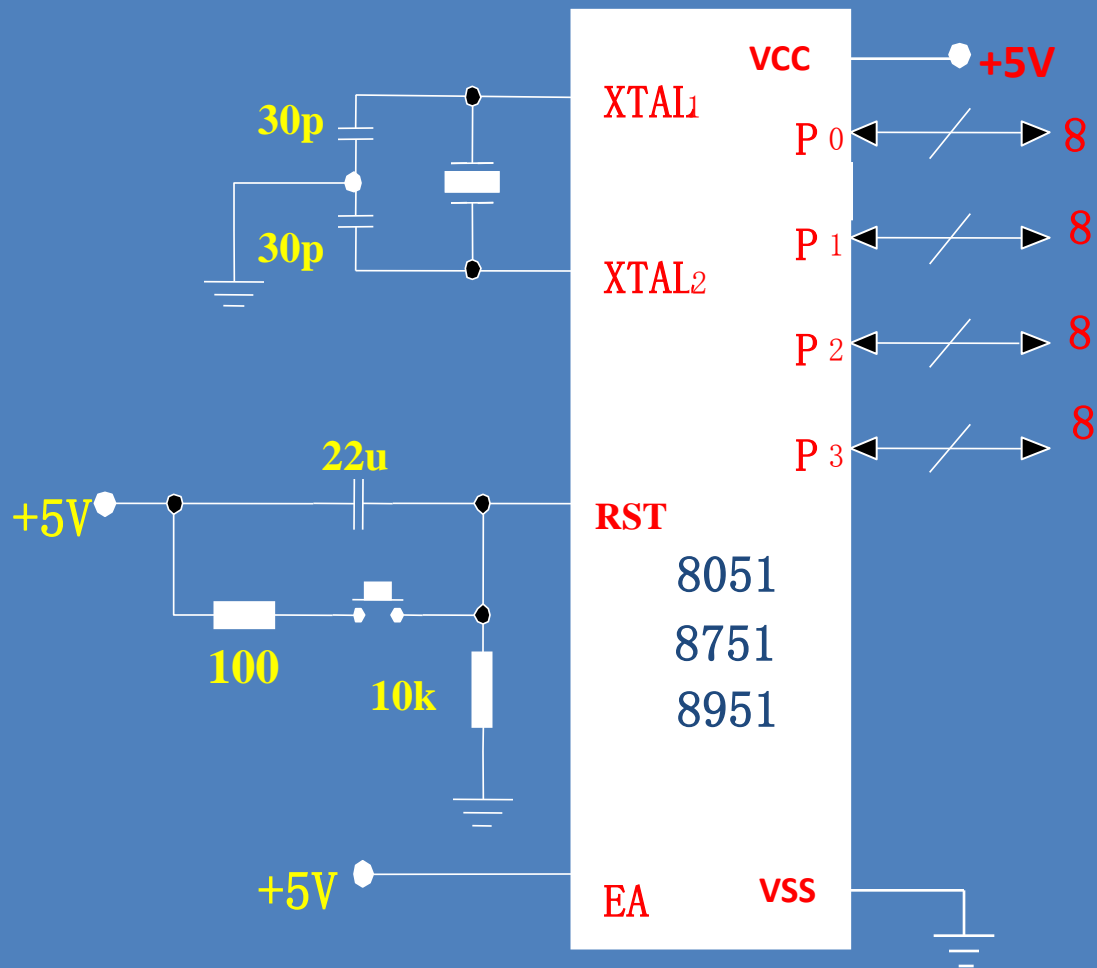
RAM 不复位

表 2-4 复位后各专用寄存器的状态

寄存器	内 容	寄存器	内 容
★ PC	✓ 0000H	T2CON	00H
ACC	00H	TH0	00H
B	00H	TL0	00H
★ PSW	00H	TH1	00H
★ SP	✓ 07H	TL1	00H
DPTR	0000H	TH2	00H
★ P0~P3	✓ 0FFH (30H输入)	TL2	00H
IP(8051)	×××00000B	RLDH	00H
(8052)	××000000B	RLDL	00H
IE(8051)	0××00000B	SCON	00H
(8052)	0×000000B	SBUF	不定
TMOD	00H	PCON(HMOS)	0×××××××B
TCON	00H	(CHMOS)	0×××0000B



## 6、MCS-51单片机最小应用系统



8051/8052最小应用系统

## 2.6 单片机的低功耗操作方式

### 1、为什么需要低功耗操作：

从便携等方面考虑，从芯片排列方面考虑，从可靠性方面考虑，从发展趋势方面考虑。

### 2、怎样进行低功耗操作：

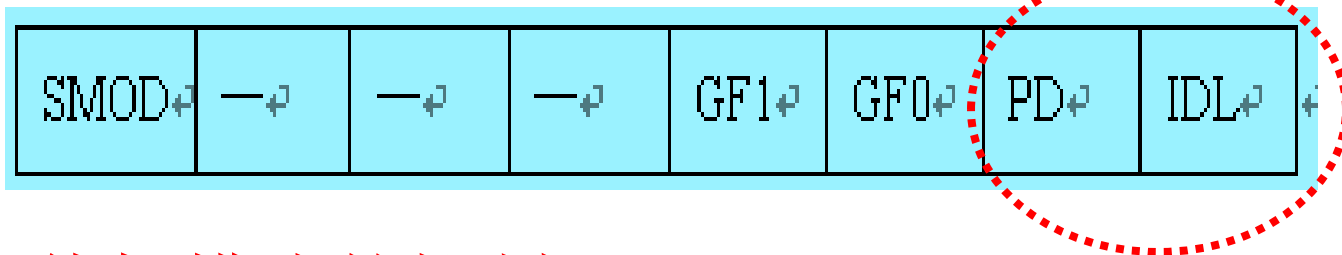
对PCON的相应位进行置高或置低。如  
**SETB IDL 等同于 SETB PCON.0**

### 3、低功耗工作模式及耗电量：

- 1) 待机方式：从正常工作24mA降为3.7mA。
- 2) 掉电方式：从待机3.7mA降为50uA。

## 4、待机模式（Idle Mode）：

- 1) 进入：将IDL置1进入待机工作状态。
- 2) 待机状态：提供给CPU的内部时钟信号被切断，但时钟信号仍提供给中断逻辑，定时器和串行口。CPU的全部状态在待机期间都保留起来，包括：堆栈指针SP、程序计数器PC、程序状态字PSW、累加器ACC以及所有的工作寄存器。
- 3) 恢复：可以通过被允许的中断或复位来终止待机方式。中断——结束待机，IDL自动被硬件清除；复位信号——使RST保持两个机器周期，“复位”也自动清除IDL



## 5、待机模式的控制

两种节电运行方式（待机方式和掉电方式），由PCON的有关位来控制。

- **SMOD**波特率倍增位，=1时，串口1，2，3方式，波特率加倍。
- **GF1，GF0**通用标志位。
- **PD** 掉电方式位，=1时，触发掉电工作方式。
- **IDL** 待机方式位，=1时，触发待机方式。
- **PD和IDL**同为1时，则触发掉电工作方式。

## 6、掉电模式（Power Down Mode）

- 1) 进入：执行一条把PD置1的指令；
- 2) 状态：片内振荡器停止工作，一切功能停止，只有RAM（片内）内容被保持。
- 3) 恢复：掉电方式只有硬件复位一种退出方式（并将重新定义所有的SFR，但不改变片内RAM的值），掉电方式时VCC可降到2V。

## 7、低功耗模式的应用

### 1) 降低功耗

待机模式和掉电模式主要是为降低功耗而设计的。

### 2) 抗电磁干扰

把待机模式作为抗电磁干扰措施。

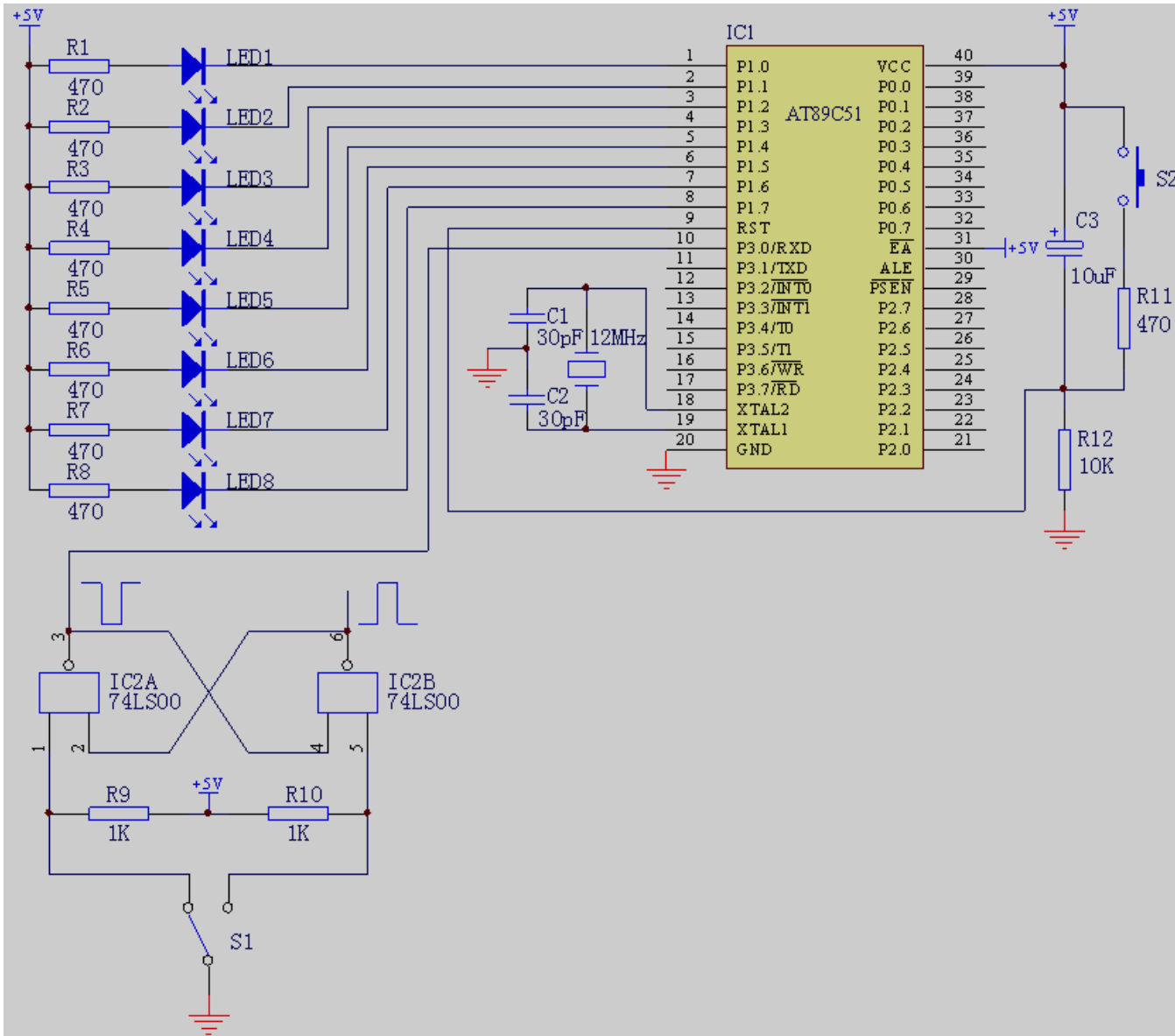
# 单片机应用实例

需求:

每按一次按键，8只发光二极管亮灯数据左移一位。

首先对单片机工作条件(电源、时钟和复位电路)进行设计，然后根据设计任务要求对I/O接口电路进行设计；

## 原理图及程序



## 源程序如下

```
ORG 0000H
```

MOV A, #0FEH

LP: MOV P1, A

JB P3.0, \$

JNB P3.0, \$

RL A

## SJMP LP

END

返回



## 作业：

- 1、如果80C51单片机晶振频率为24MHz，  
时钟周期、机器周期为多少？
- 2、80C51单片机复位后的状态如何？复位  
方法有几种？