微机原理与接口技术

§3 MCS-51微控制器结构与原理

主讲人：佘青山、张卫

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

2022年9月5日

1、ROM形式：

8051	为掩膜型 ROM
8751	为EPROM型
8031	无ROM
AT89C51	为Flash型ROM

此外，AT89S51可以通过在线编程（In-System Programming, ISP）接口下载程序

2、RAM、ROM的容量

8051、8751、AT89C51

其内部用户数据存储器RAM为128个单元

其内部程序数据存储器ROM为4K

RAM——随机存取存储器（Random Access Memory），又分SRAM（静态）和DRAM（动态），也叫数据存储器

ROM——只读存储器（Read-Only Memory），也叫程序存储器

8052、8752、AT89C52其内部容量翻倍，此外中断源个数、计数器个数也有所不同。

3、工艺形式

有C——高速CHMOS、PP互补金属氧化物的HMOS工艺

无C——HMOS工艺

4、使用电压

有LV——可以使用3.7V

无LV——可以使用5V电源

注： LV代表低电压型

此外，还有常用的AT89C2051：

2K ROM、128个单元RAM、2个定时器、五个中断源

89C51微控制器典型产品资源配置

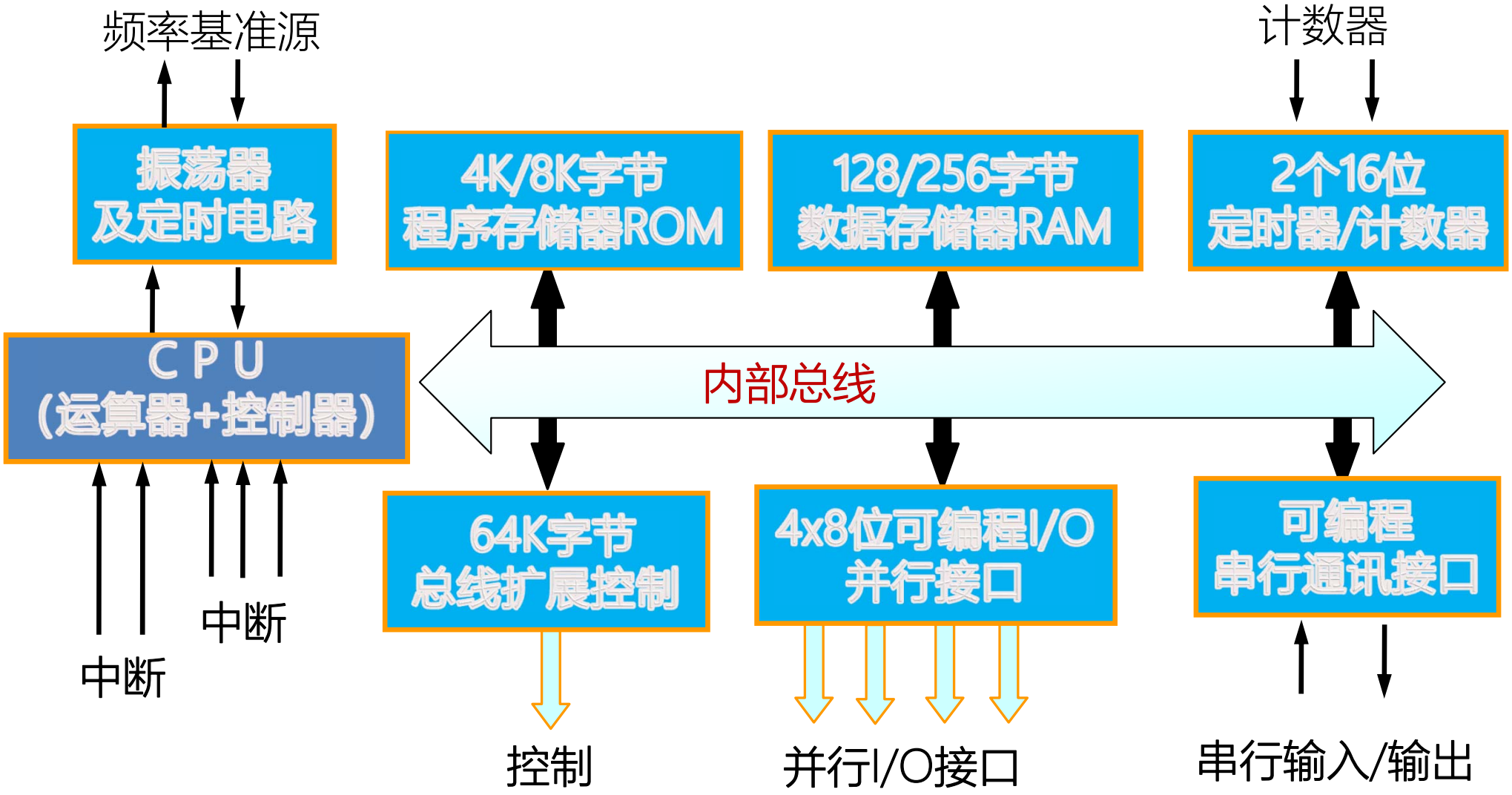
分类		芯片型号	存储器类型及字节数		片内其它功能单元数量			
			ROM	RAM	并行口	串行口	定时/计数器	中断源
总线型	基本型	80C31	无	128	4个	1个	2个	5个
		80C51	4K 掩膜	128	4个	1个	2个	5个
		87C51	4K EPROM	128	4个	1个	2个	5个
		89C51	**4K Flash**	128	4个	1个	2个	5个
	增强型	80C32	无	256	4个	1个	3个	6个
		80C52	8K 掩膜	256	4个	1个	3个	6个
		87C52	8K EPROM	256	4个	1个	3个	6个
		89C52	**8K Flash**	256	4个	1个	3个	6个
非总线型		89C2051	2K Flash	128	2个	1个	2个	5个
		89C4051	**4K Flash**	128	2个	1个	2个	5个

注：表中加黑的 ATMEL 公司 AT89 系列产品应用方便，应优先选用。

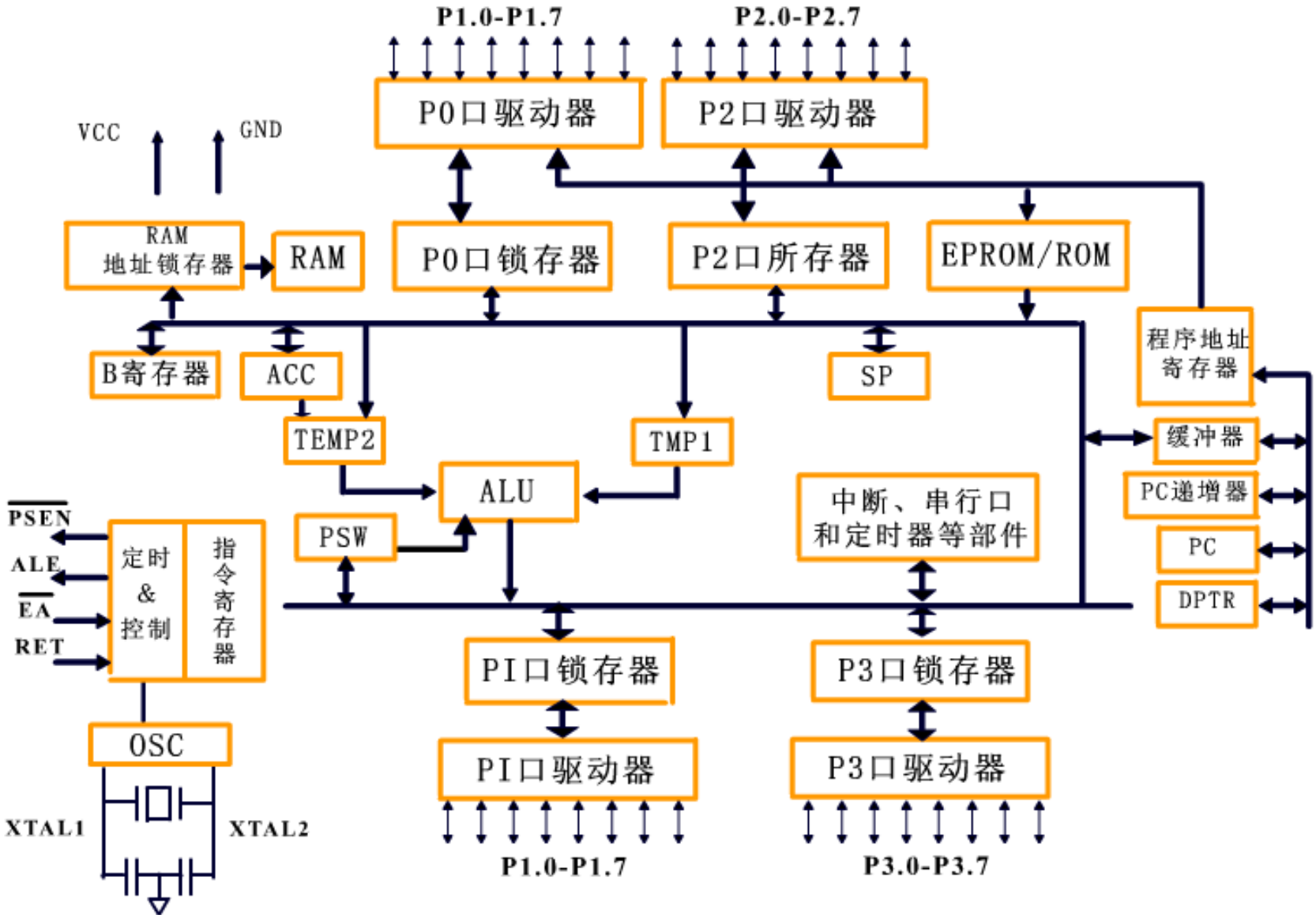
与80C51兼容的主要产品：

生产厂家	单片机型号
ATMEL 公司	AT89C5x 系列（89C51/89S51、89C52/89S52、89C55 等）
Philips（菲利浦）公司	80C51、8xC552 系列
Cygnal 公司	C80C51F 系列高速 SOC 单片机
LG 公司	GMS90/97 系列低价高速单片机
ADI 公司	ADμC8xx 系列高精度单片机
美国 Maxim 公司	DS89C420 高速（50MIPS）单片机系列
台湾华邦公司	W78C51、W77C51 系列高速低价单片机
AMD 公司	8-515/535 单片机
Siemens 公司	SAB80512 单片机

3.2 MCS-51微控制器内部结构



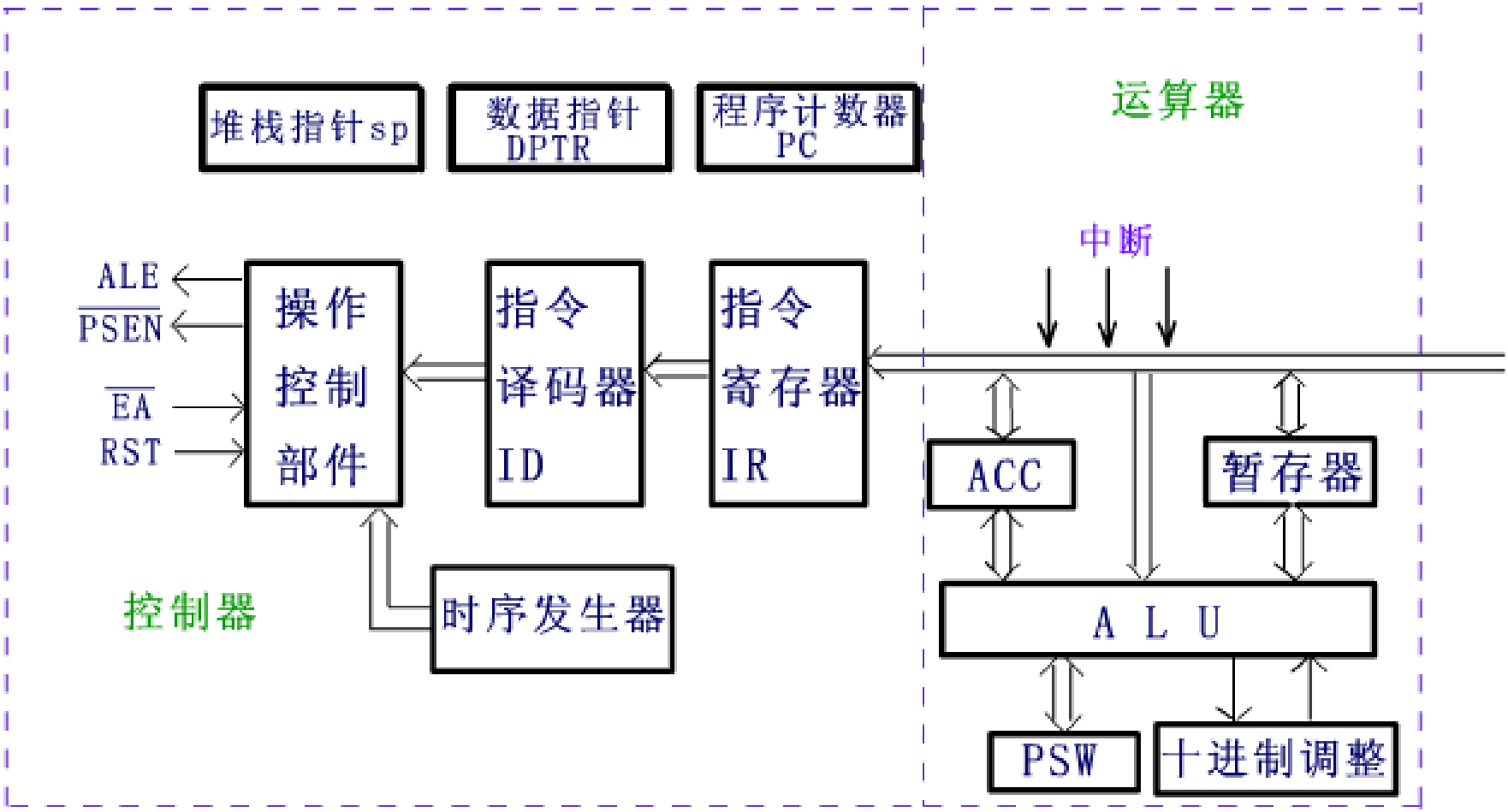
MCS-51微控制器内部结构



8051微控制器功能模块与特点

- 并行I/O口：4个8位I/O口：P0、P1、P2、P3，具有第二功能。
- 中断系统：具有5个中断源，2个中断优先权。
- 定时器/计数器：有2个16位的定时器/计数器，具有4种工作方式。
- 串行接口：1个全双工的串行口，用于微控制器与具有串行接口的外设进行异步串行通信，也可以扩展I/O接口
- 布尔处理器：具有较强的位寻址、位处理能力。
- 时钟电路：产生微控制器工作所需要的时钟脉冲，需要外接晶体振荡器和微调电容。
- 指令系统：有5大功能，111条指令，为复杂指令系统(CISC)。

CPU内部结构


堆栈指针sp
数据指针
DPTR
程序计数器
PC
运算器
中断
ALE
PSEN
EA
RST
操作
控制
部件
指令
译码器
ID
指令
寄存器
IR
ACC
暂存器
控制器
时序发生器
A L U
PSW
十进制调整

1.控制器

控制器由操作控制部件、时序发生器、指令寄存器IR、指令译码器ID、指令计数器PC等组成。

控制器是CPU的大脑中枢，是识别指令，控制计算机各部分工作的部件。包括控制取指令、译码和执行三个步骤的全部控制。

指令寄存器IR：用来存放当前正在执行的指令代码。

指令译码器ID：用来对指令代码进行分析、译码，根据指令译码的结果，输出相应的控制信号

1. 控制器（指令部件、时序部件、操作控制部件）

控制器是CPU的大脑中枢，其功能是从ROM中逐条读取指令，进行指令译码，并通过定时和控制电路，在规定的时刻发出执行指令操作所需的控制信号，使各部分按照一定的节拍协调工作，实现指令规定的功能。

控制器可归纳为指令部件、时序部件和操作控制部件三部分组成。

（1）指令部件

由16位程序计数器PC（Program Counter）、指令寄存器IR（Instruction Register）、指令译码器ID（Instruction Decode）等组成。

- 程序计数器 PC：16位的ROM指针，用于存放下一条取指指令的地址，寻址范围为64K；
- 指令寄存器IR： 存放当前指令的操作码，等待译码；
- 指令译码器ID： 对当前指令操作码进行解析，并通过控制电路产生执行该指令需要的控制信号，完成指令规定的操作。

（2）时序部件

时序部件由时钟电路和分频器组成，用于产生MCU运行程序时，操作控制部件所需的时序信号。包括CPU工作的时钟基准（称为**振荡周期**或**时钟周期**），以及状态周期、机器周期等信号。

（3）操作控制部件

操作控制部件为指令译码器的输出信号配上节拍电位和节拍脉冲，形成执行指令需要的操作控制序列信号，以完成规定的操作。

2. 运算器

运算器的任务是数据的处理和加工。由算术逻辑单元ALU、累加器ACC、暂存寄存器、程序状态寄存器PSW、布尔处理器、BCD码运算调整电路等通过内部总线连接而成。

- ALU (Arithmetic logic Unit)

完成算术运算及与、或 、非、异或等逻辑操作，并通过对运算结果的判断，影响程序状态寄存器PSW相关位的状态。

- 位处理器（布尔处理器）

能直接对位（bit）进行操作，操作空间是位寻址空间。位处理器中功能最强、使用最频繁的位是C，也称其为位累加器。

- 暂存寄存器

用于运算数据的暂时存放，该寄存器不能访问。

1. 程序与指令

程序存储执行： 计算机组成原理基础就是冯•诺依曼的体系结构，其基本设计思想就是**存储程序**和**程序控制**，即计算机的工作过程实质上是执行程序的过程。用户编写的**程序要预先存放在ROM中**，微控制器的工作过程就是从ROM中逐条取出指令并执行的过程。

程序： 完成一个特定功能的一系列指令集。

指令： 微控制器指挥各功能部件工作的指示和命令。指令是一组二进制数，其编码格式及功能、类别和数量因CPU的不同而不同，是芯片设计者设定的。

一条指令包括两部分内容：

- **操作码：** 指明指令的功能（即**做什么操作**）；
- **操作数：** 指明指令执行的数据或数据存放的地址（即**操作对象**）。

2. 指令样例

	助记符		机器码（十六进制）		机器码（二进制）	
			操作码	操作数		
①	ADD	A, #68H	24	68	00100100	01101000
②	MOV	A, #15H	74	15	01110100	00010101
③	SETB	P1.0	D2	90	11010010	10010000

执行的操作是：

① 将累加器A的内容与立即数68H相加，并把结果放回A中。

即（A）←（A）+ 68

② 将立即数15H赋给累加器A，执行后A中的内容为15H；即（A）←15H

③ 将P1口的D0位即P1.0置为1，执行后P1.0引脚变为高电平；即P1.0←1

3. 指令执行过程

计算机（微控制器）每执行一条指令，都可以分为三个阶段：

取指令 → 分析指令 → 执行指令

（取指） （译码） （执行）

- 读取指令：根据程序计数器（程序指针）PC中的值，从ROM读出现行指令，送到指令寄存器IR。
- 分析指令：由指令译码器对现行指令进行译码，分析该指令要求实现什么操作，如执行数据传送，还是加、减等运算等。
- 执行指令：取出操作数，由控制逻辑电路发出相应的控制信号，完成操作码规定的操作。

指令计数器PC（Program Counter）

指令计数器也叫指令指针，是一个独立的16位寄存器，不属于内部的特殊功能寄存器，因此不可寻址。

PC中存放的是下一条将要执行的指令的地址，即CPU是根据PC的值从程序存储器ROM中取得指令，同时程序计数器PC值自动加1，指向下一条指令地址。PC值的范围为0000H~FFFFH，即可寻址范围为64K。

微控制器的工作过程实质就是执行程序的过程。由于程序是要完成一个特殊功能的一系列指令的集合，因此执行程序的过程就是**逐条执行指令的过程**。

微控制器在前必须将程序按一定的顺序存放在程序存储器的存储单元中（这个过程也叫**程序烧入**）。微控制器工作时就是根据**PC指针逐条读取存放在ROM中的指令，并逐条译码并执行**，从而完成程序规定的任务。

举例说明程序执行过程：

例：MOV A,#09H　74H 09H ;把09H送到累加器A中

通常指令的格式是：（操作码）（操作数）

助记符：　　MOV　A，#09H；

机器码（十六进制）：74H　09H；

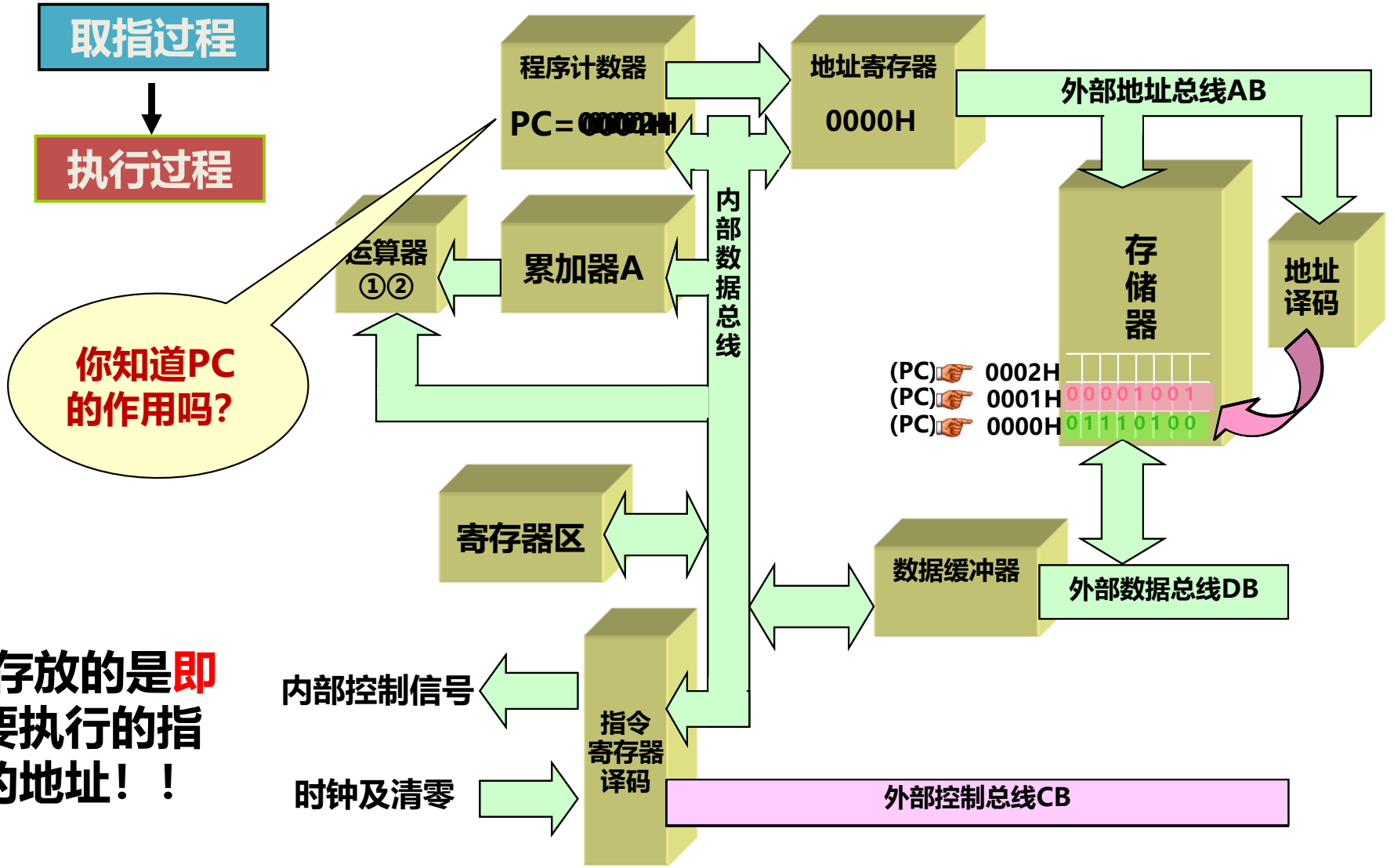
机器码（二进制）：　01110100　00001001B；

其所执行的操作是：　09→（A）

即：将数据09H数据送到累加器A中。

该指令的操作码是：74H；操作数是：09H

例: MOV A,#09H 74H 09H ;把09H送到累加器A中



PC存放的是即将要执行的指令的地址！！

单选题 1分

设置

(1) 8051 微控制器中的PC是不可寻址的。

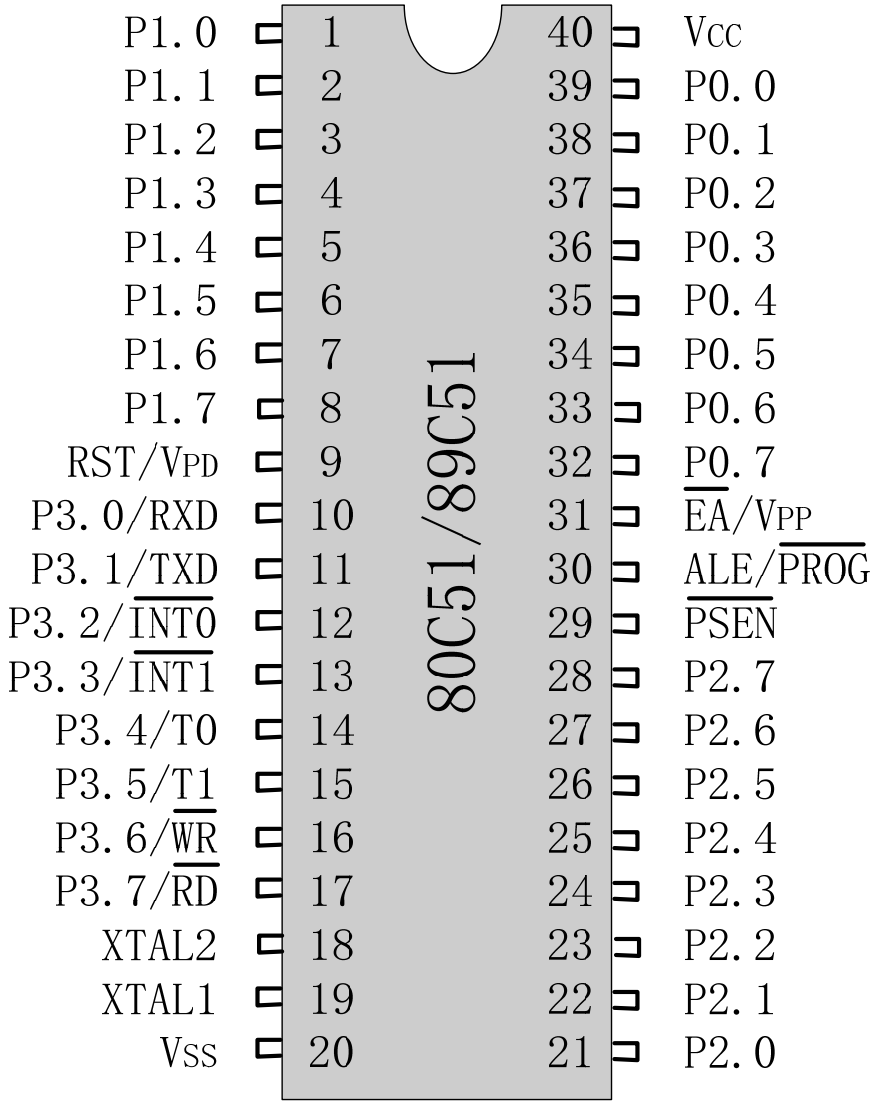
A √

B ×

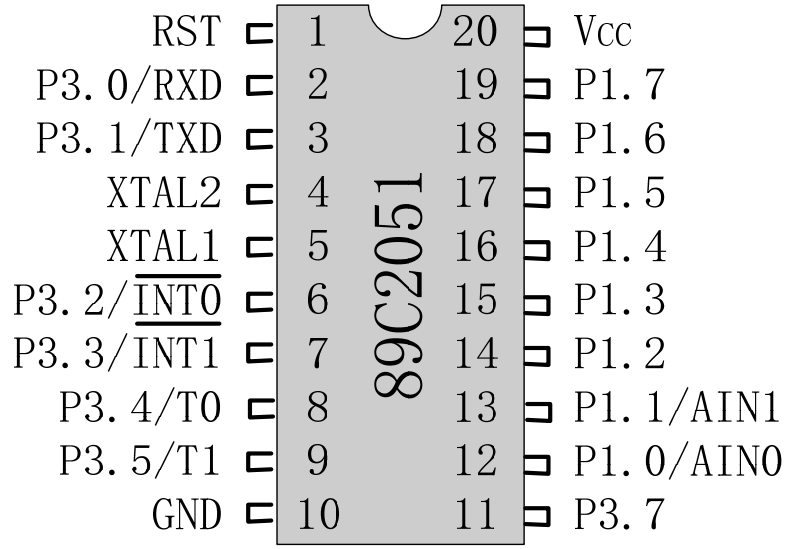
提交

3.4 MCS-51微控制器引脚

总线型



非总线型



注：类似的还有Philips公司的
87LPC64，20引脚
8XC748/750/（751），24引脚
8X749（752），28引脚
8XC754，28引脚
等等

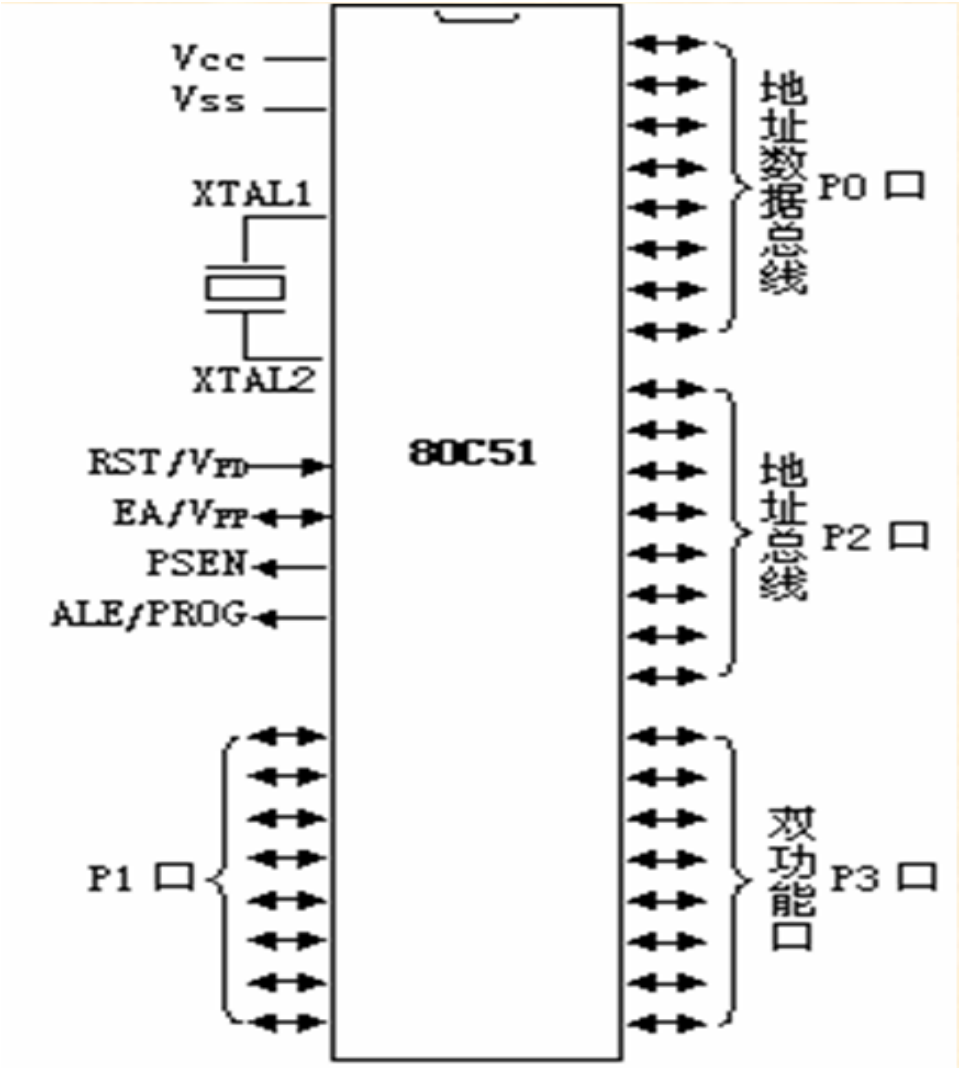
40条引脚可分为4组：

(1) 电源、地： 2条

(2) 时钟电路： 2条

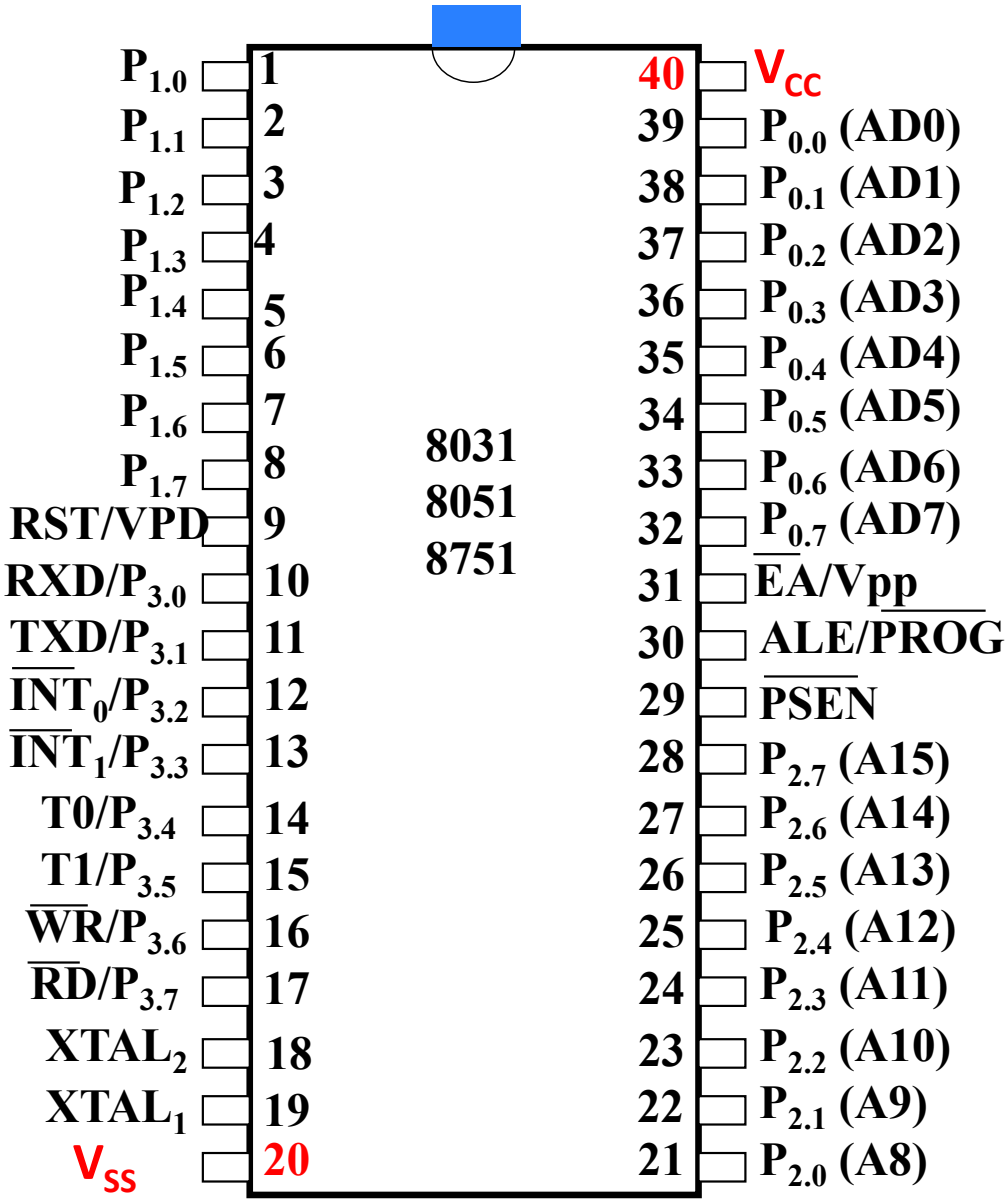
(3) 控制线： 4条

(4) I/O口线： 32条

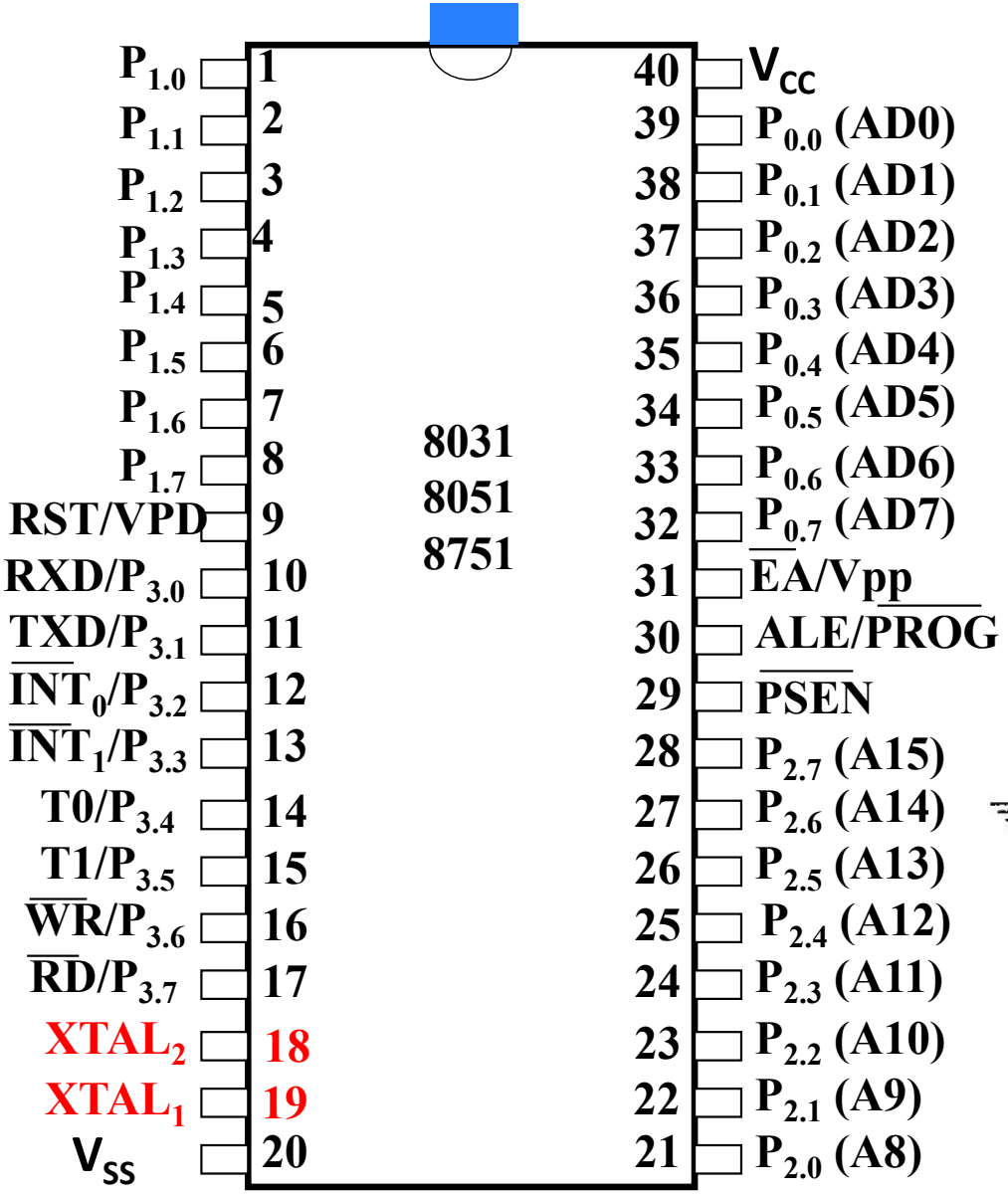


●电源引脚

- **Vcc（Pin40）：正电源引脚。正电源接4.0～5.0V电压，正常工作电压为+5V。**
- **Vss或GND（Pin20）：接地引脚。**

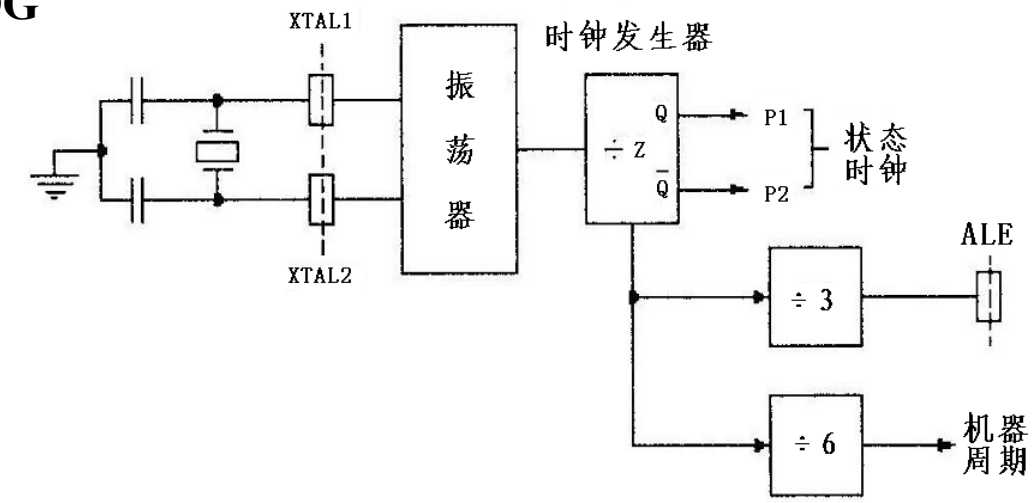


●时钟引脚

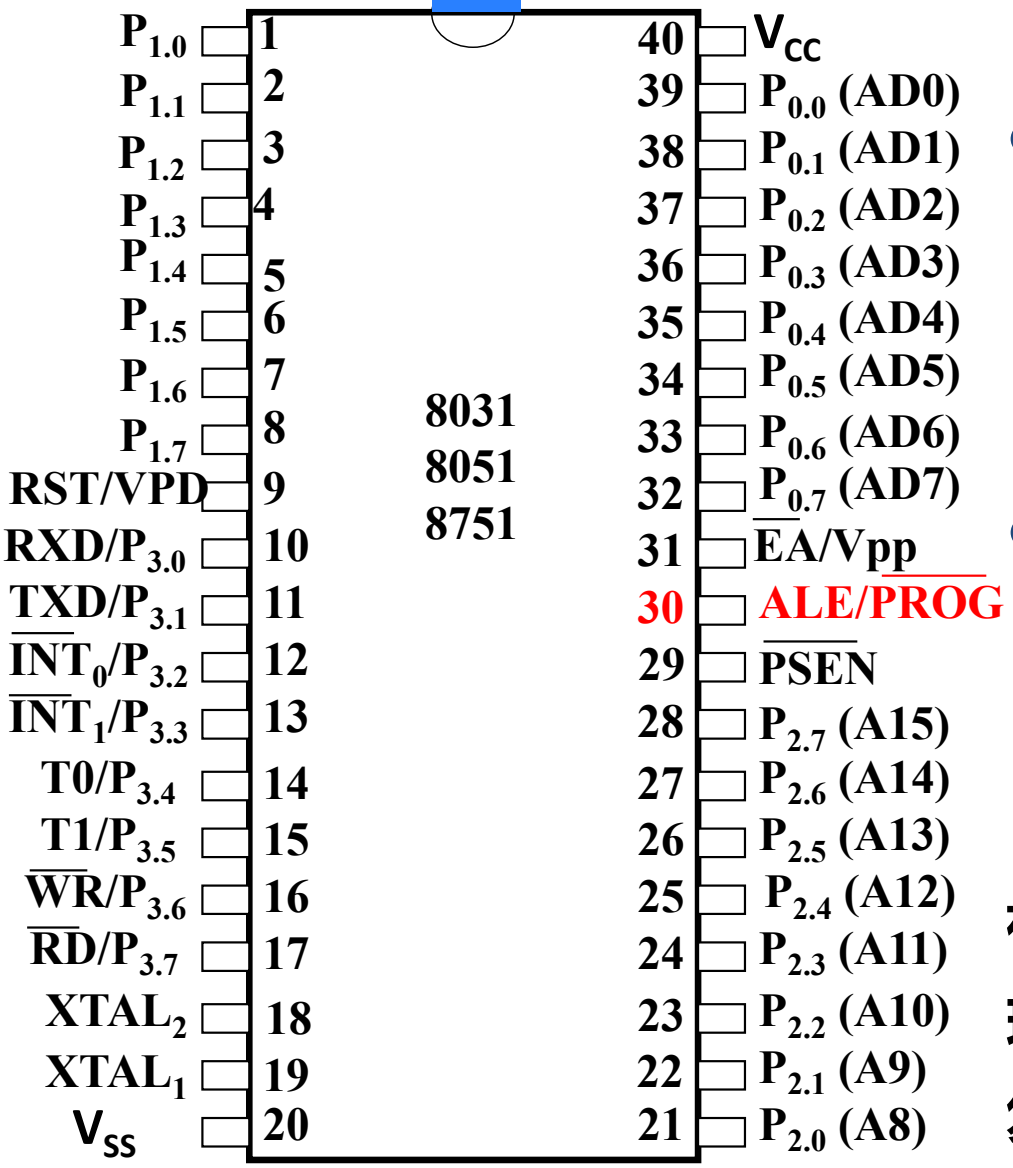


•**XTAL1（Pin19）**：用作片内振荡电路的输入端。

•**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。



●ALE / $\overline{PROG}$引脚

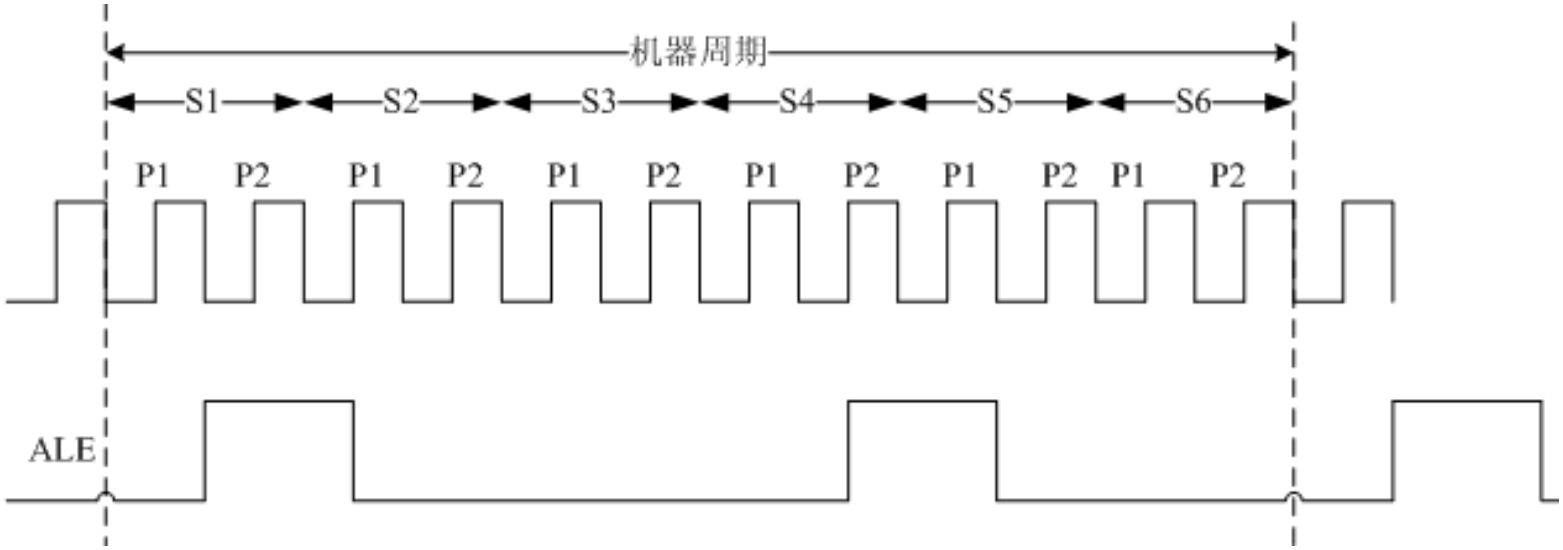


- ALE/$\overline{PROG}$引脚（Pin30）具有两种功能，可以作为地址锁存使能端（Address Latch Enable）和编程脉冲输入端
- 当作为地址锁存使能端时为ALE。当微控制器访问外部存储器时，地址锁存允许信号输出端。

有效时输出一个高脉冲。ALE（地址锁存）的负跳变将低8位地址打入锁存。以实现P0口的8位数据线和低8位地址线的分时复用和隔离。

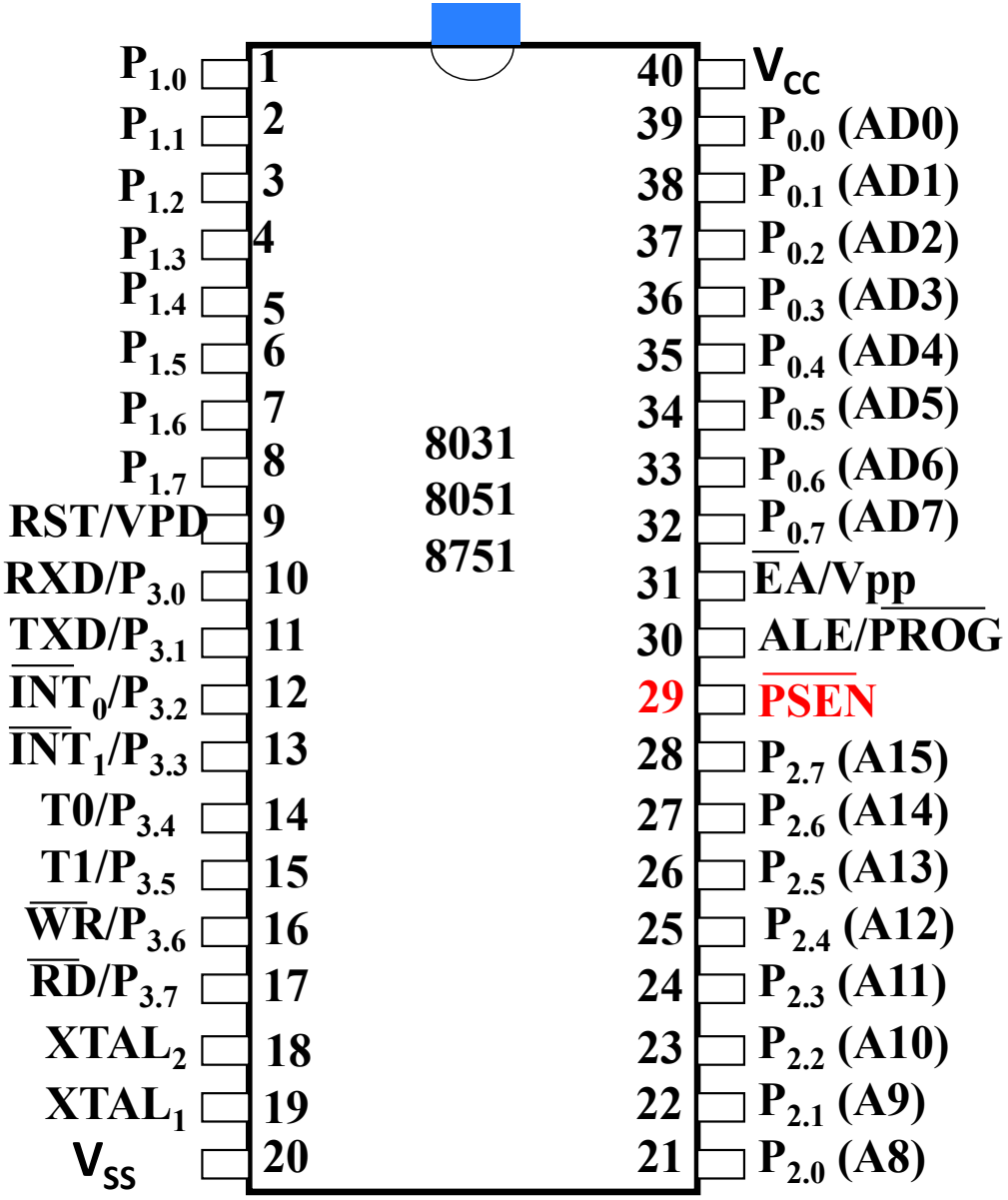
ALE / PROG引脚

- 当微控制器在非访问内部程序存储器时，ALE引脚将有**一个1/6振荡频率**的正脉冲信号输出，该信号可以用于外部计数或电路其他部分的时钟信号。



- 当作为编程脉冲输入端时，在进行程序下载时使用。

●/PSEN引脚

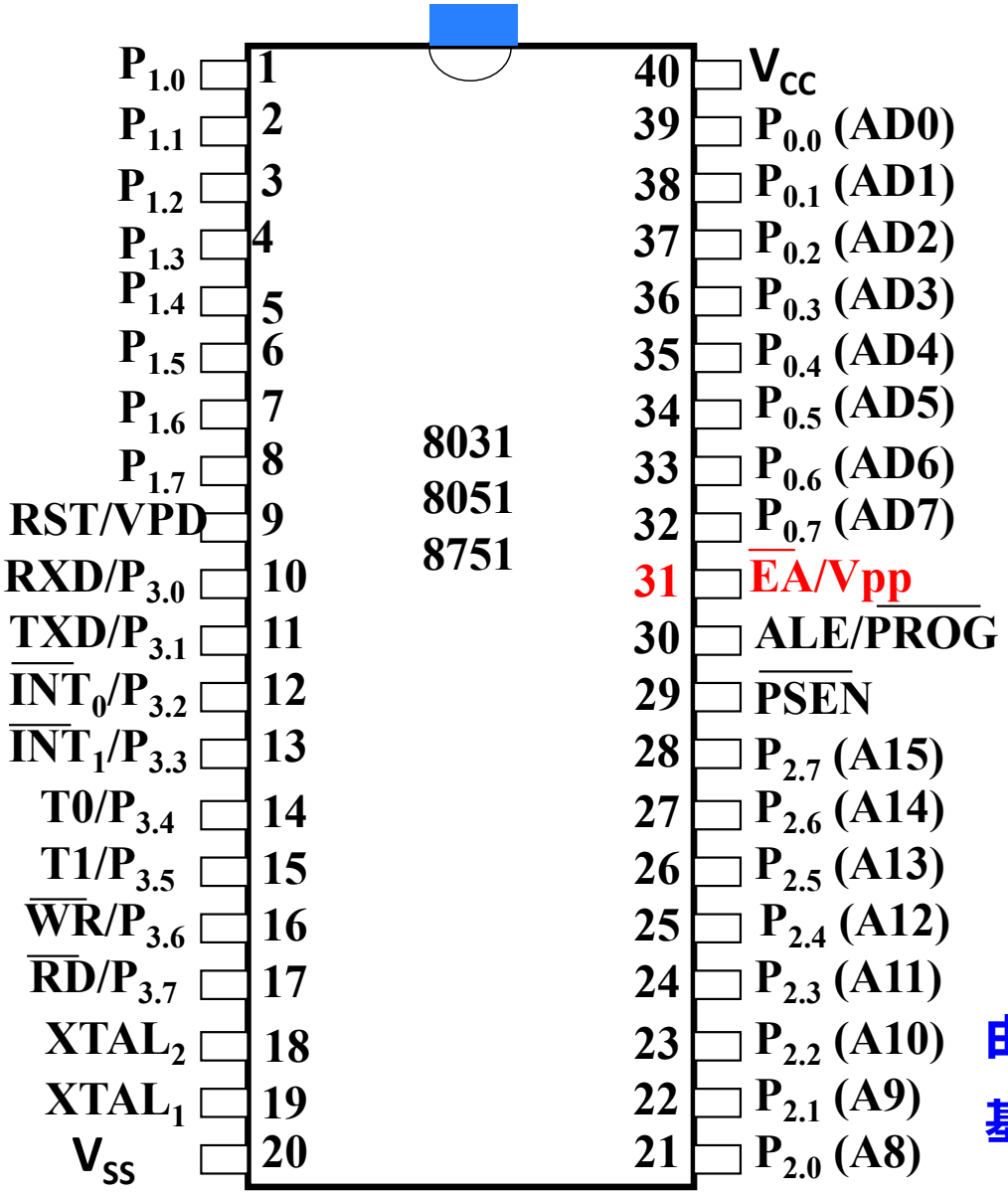


●$\overline{PSEN}$ (Program Store Enable, Pin29)

● 引脚是微控制器访问外部程序存储器的读选通信号(输出)，低电平有效。

3.4 MCS-51微控制器引脚

●$\overline{EA}$/VPP引脚



$\overline{EA}$/Vpp引脚（External Access Enable，Pin31）

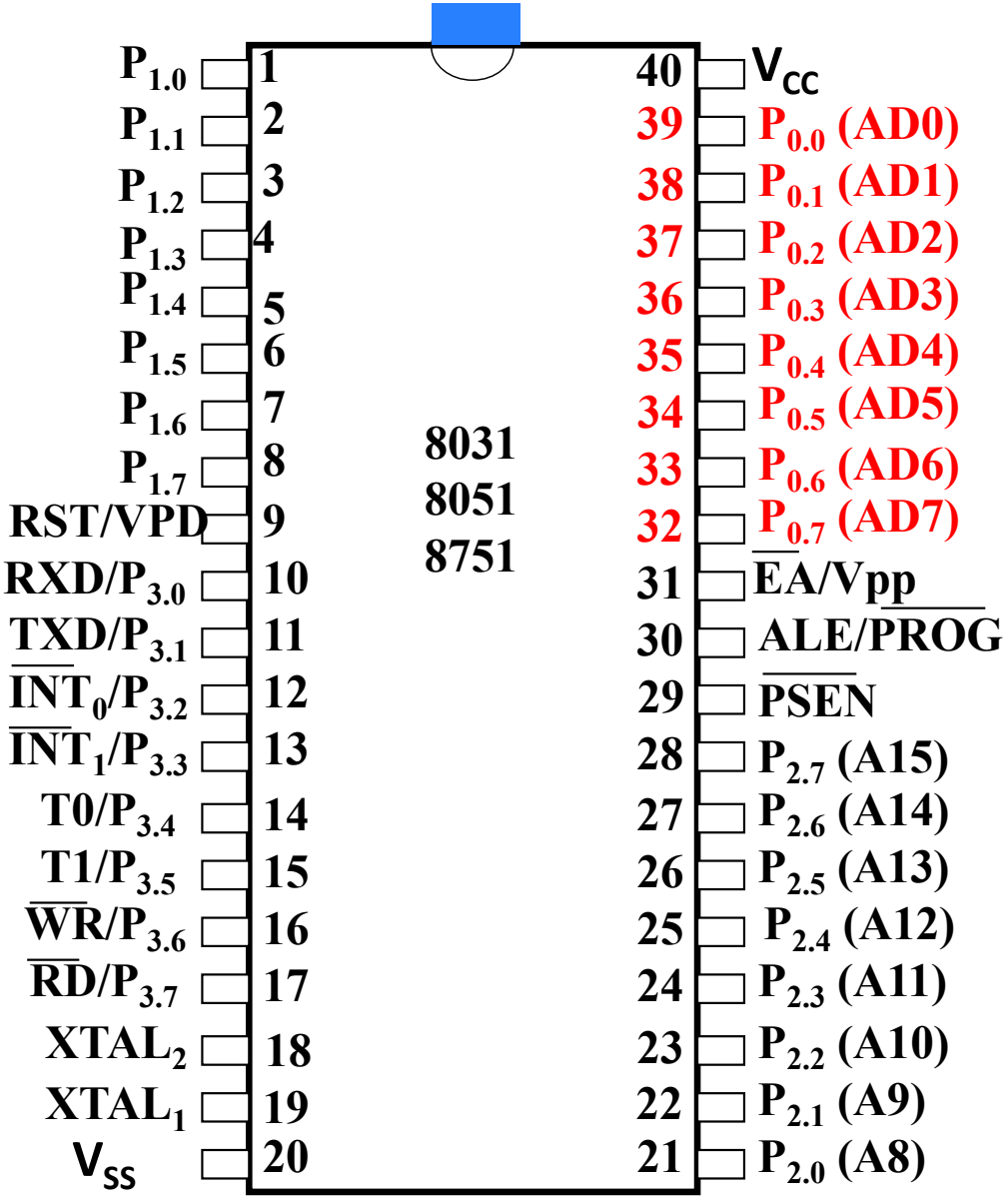
具有两种功能，访问内部或外部程序存储器选择信号和提供编程电压。

$\overline{EA}$ = 1 时，访问内部程序存储器，即内ROM

$\overline{EA}$ = 0 时，只访问外部程序存储器，即外ROM

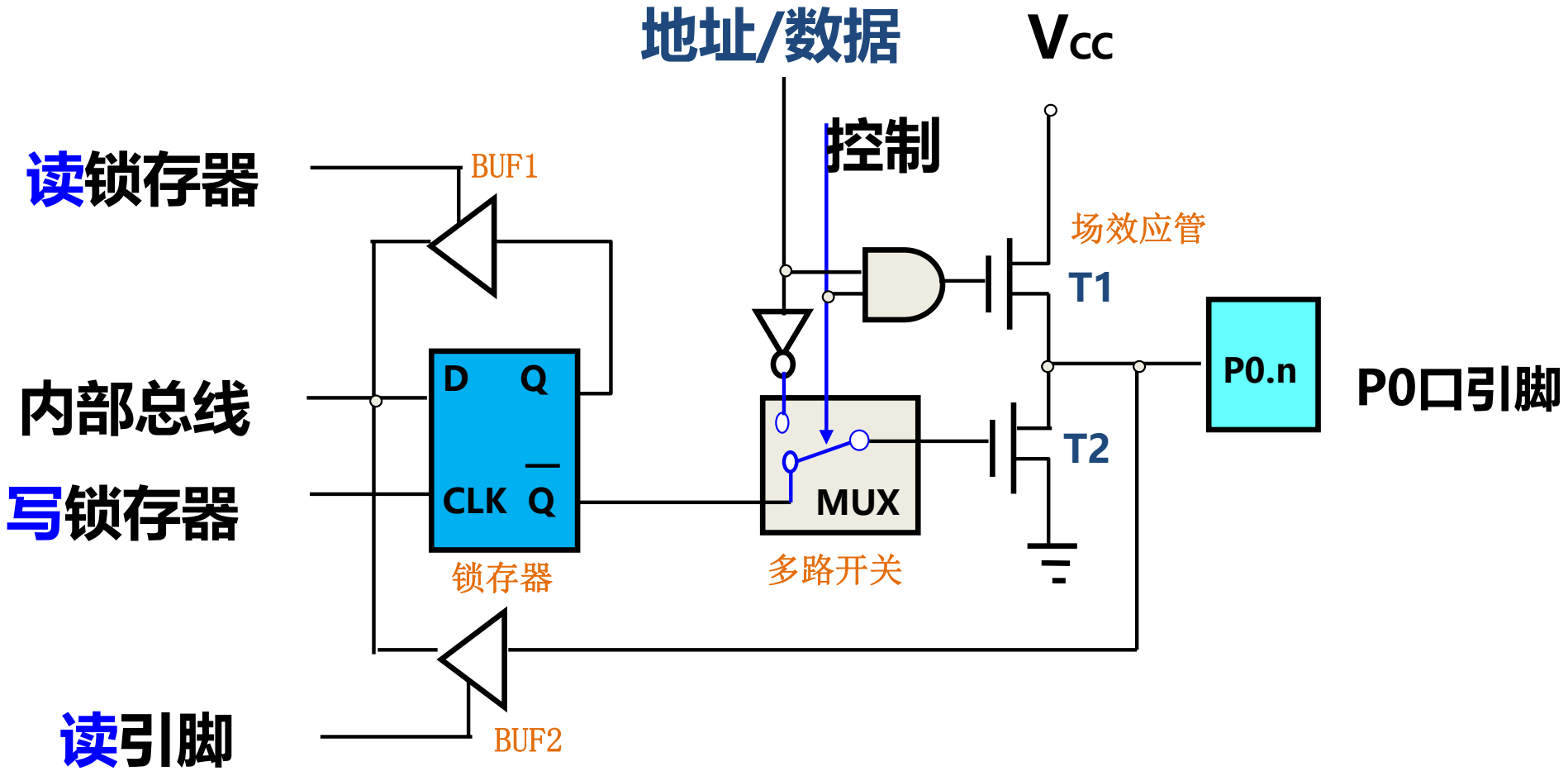
由于增强型8051 MCU集成了大容量ROM(16-64K)，基本不需要外扩，这三个控制信号已很少使用。

●P0端口



- P0.0 ~ P0.7，占据Pin39 ~ Pin32共8个引脚。
- P0端口具有两个功能，既可以用作**双向数据总线口**，也可以分时复用输出**低8位地址总线**。

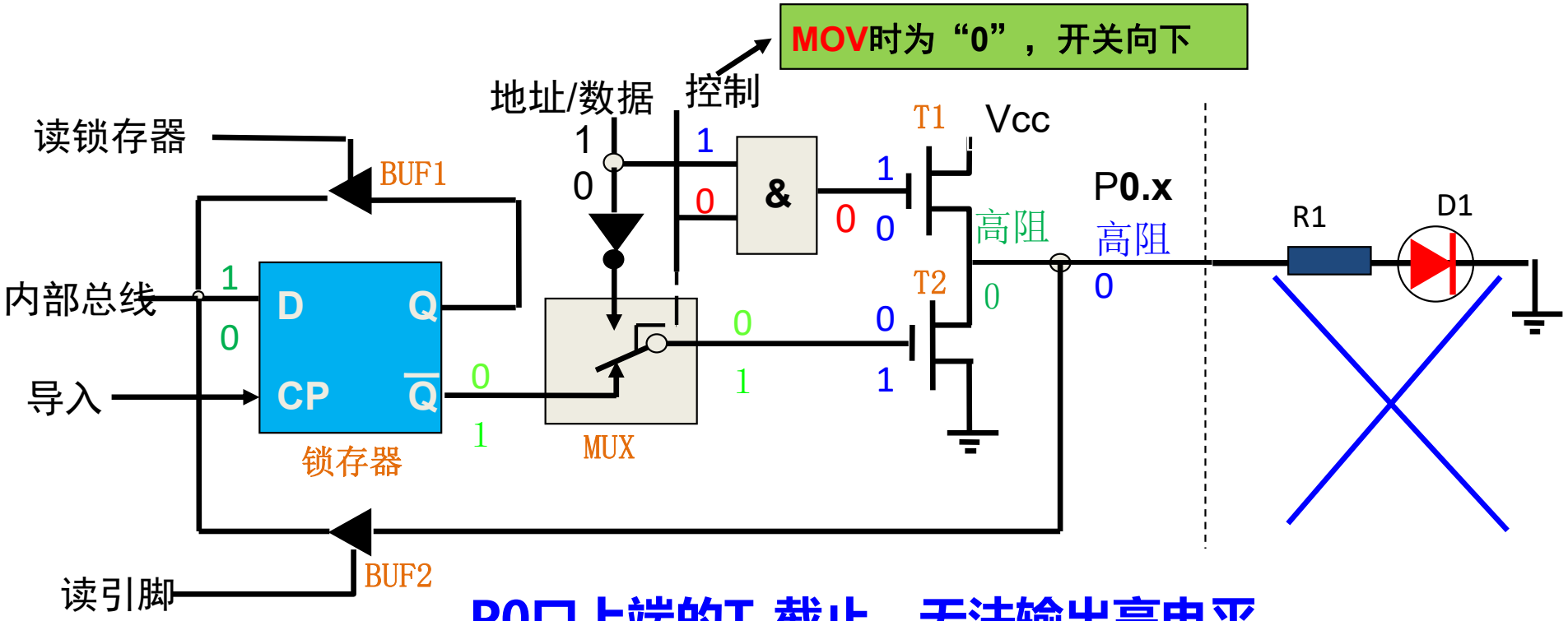
P0口内部结构


地址/数据
Vcc
控制
读锁存器
BUF1
场效应管
T1
内部总线
D
Q
P0.n
P0口引脚
写锁存器
CLK
Q
MUX
T2
锁存器
多路开关
读引脚
BUF2

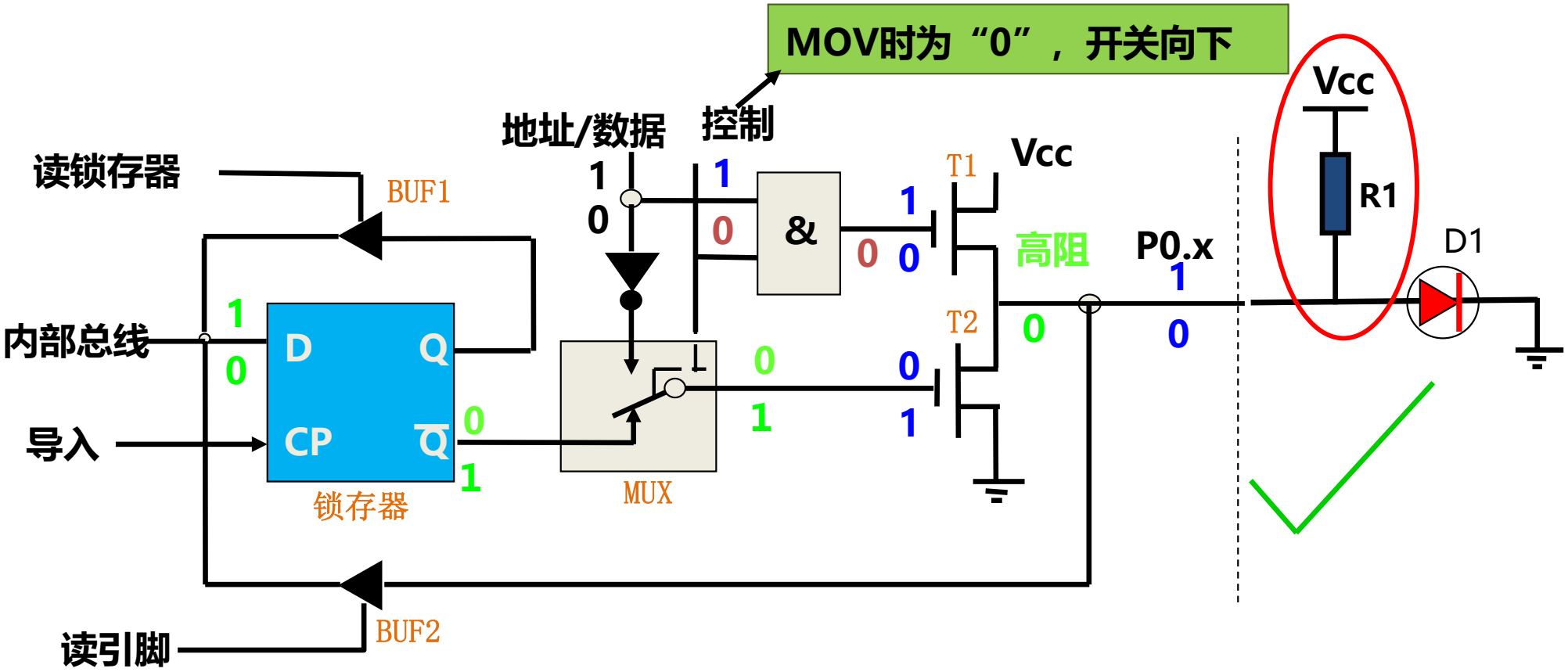
P0端口数据输出

MOV P0, A ➡ 将数据写入P0端口锁存器中

CPU发出控制电平“0”封锁“与”门，将输出上拉场效应管T_1截止，同时使多路开关MUX把锁存器与输出驱动电路接通。



P0口上端的T_1截止，无法输出高电平

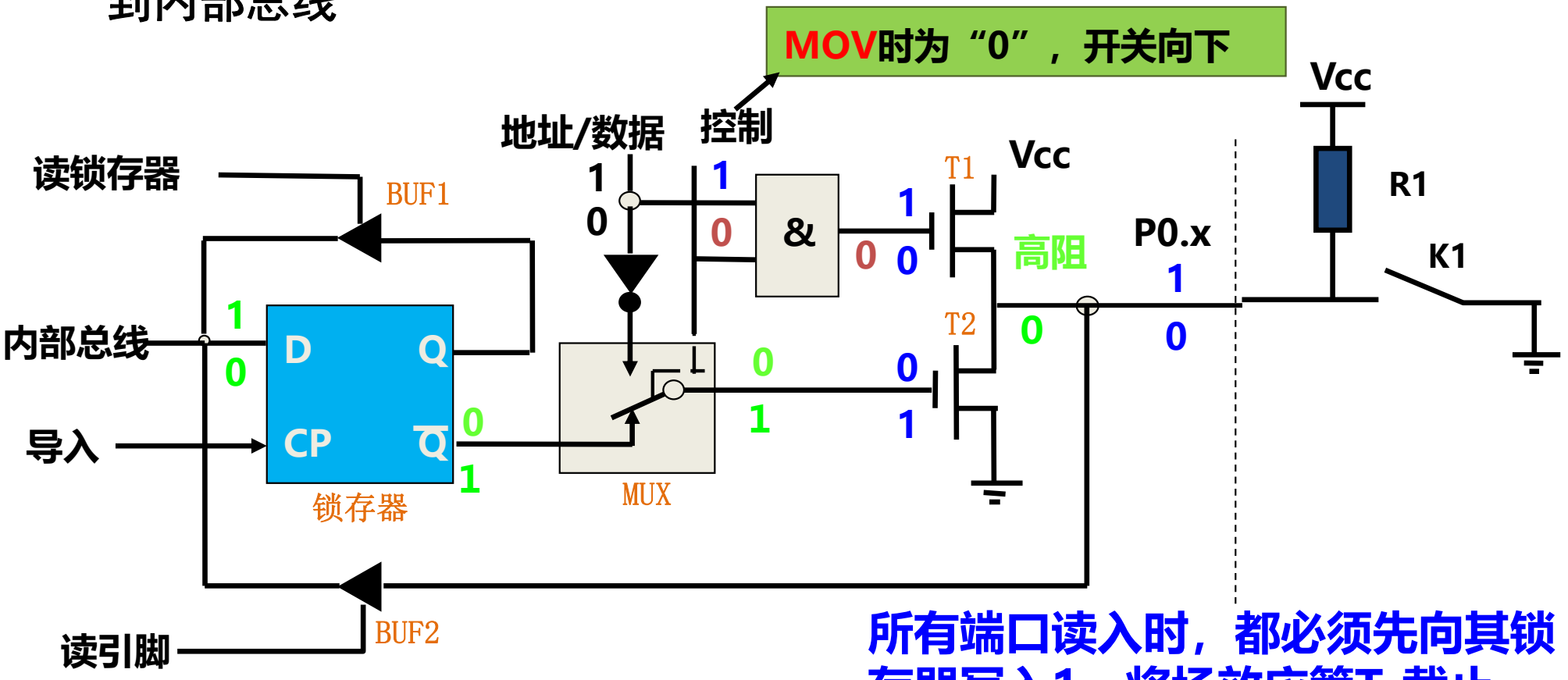


P0口做I/O口使用时，必须在外部加上拉电阻！

P0端口电平读入

MOV A, P0 ⟹ **读入P0口引脚电平!**

读指令把三态缓冲器BUF2打开，端口引脚上的数据经过缓冲器读入到内部总线



所有端口读入时，都必须先向其锁存器写入1，将场效应管T_2截止

T_2导通会发生什么问题呢? 无法得到引脚所接外设的高电平状态

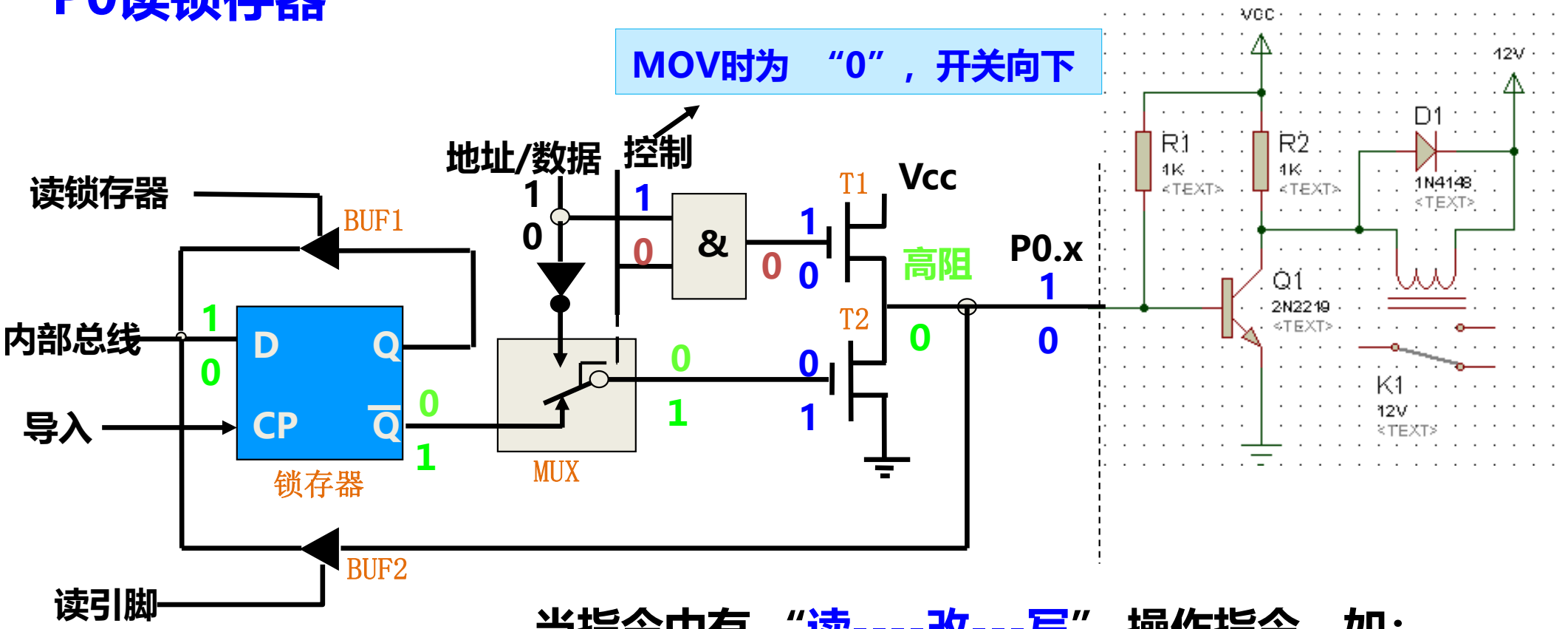
在读入端口数据时，由于输出驱动FET(场效应管)并接在引脚上，如果T_2导通，就会将输入的高电平拉成低电平，产生误读。

在端口进行输入操作前，应先向端口锁存器写“1”，使

MOV Pi, #0FFH ; i=0,1,2,3

T_2截止，引脚处于悬浮状态，变为高阻抗输入。这就是所谓的准双向口。

P0读锁存器



当指令中有 “读----改---写” 操作指令，如：

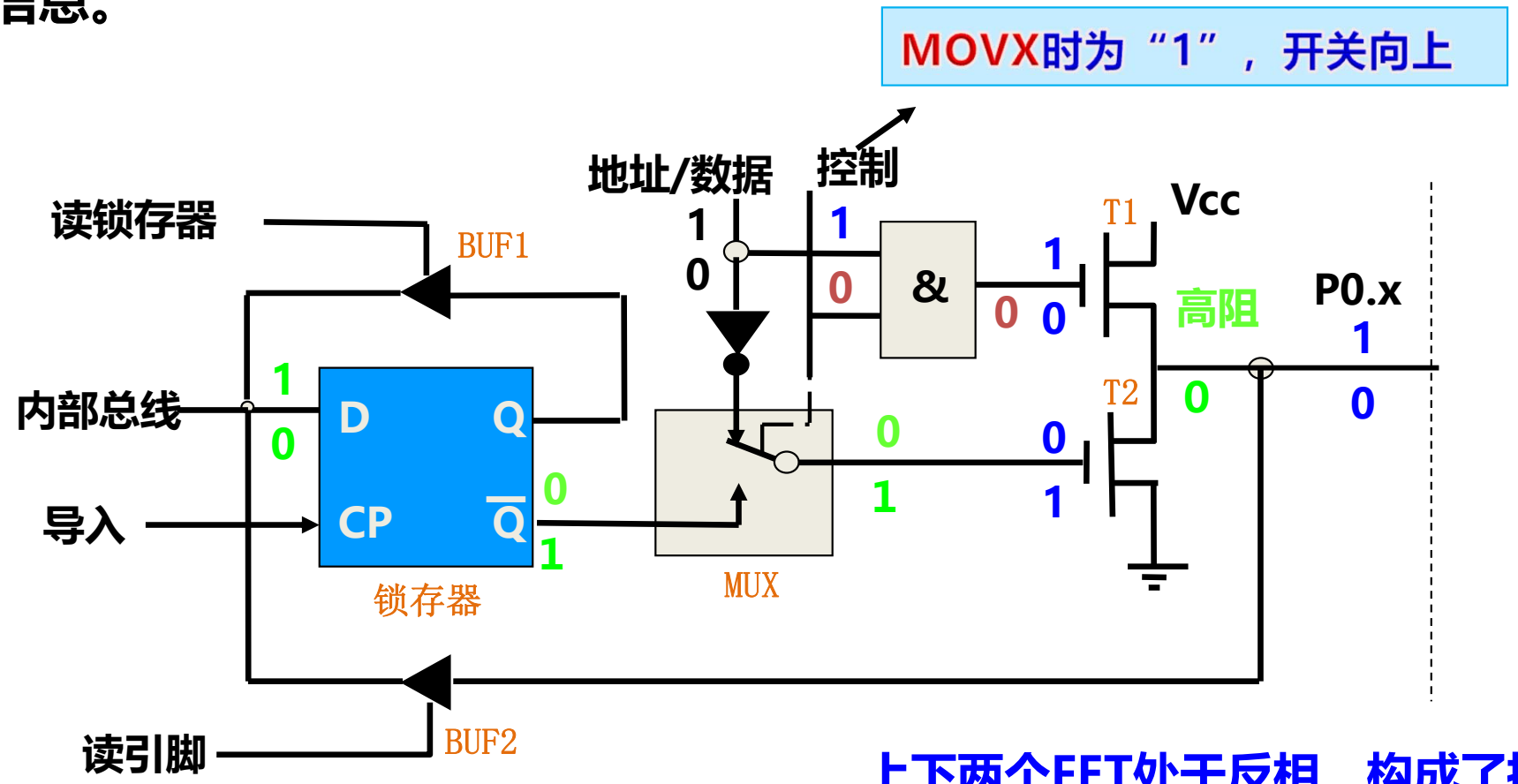
ORL P0, #0FH

指令的功能：先从P0口的锁存器中读取数据，再对数据进行或操作，然后将操作结果送回锁存器

**原因：如果此时该端口的负载恰是一个晶体管基极，且原端口输出值为1，那么导通了的PN结会把端口引脚高电平拉低；若此时直接读端口引脚信号，将会把原输出的“1”电平误读为“0”电平。现采用读输出锁存器代替读引脚，图中，上面的三态缓冲器就为读锁存器Q端信号而设，读输出锁存器可避免上述可能发生的错误。

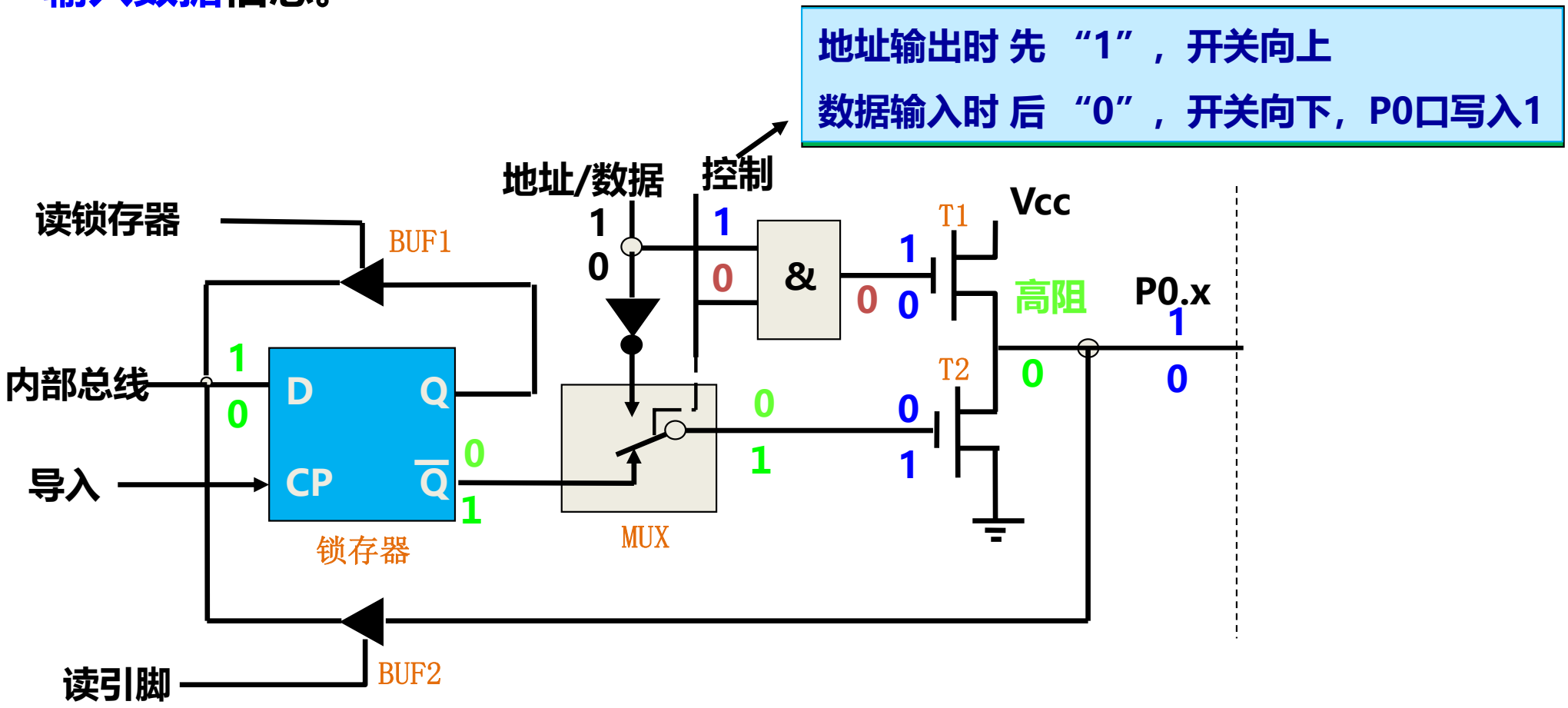
P0作为数据/地址总线口

在系统扩展时，P0端口作为地址/数据总线使用时，P0引脚输出地址/数据信息。



上下两个FET处于反相，构成了推拉式的输出电路，其负载能力大大增强。

在系统扩展时，P0端口作为地址/数据总线使用时，P0引脚输出地址/输入数据信息。



输入信号是从引脚通过输入缓冲器进入内部总线。

(2) P0口的第一功能是准双向I/O口，第二功能是分时复用的高8位地址线和8位数据线。

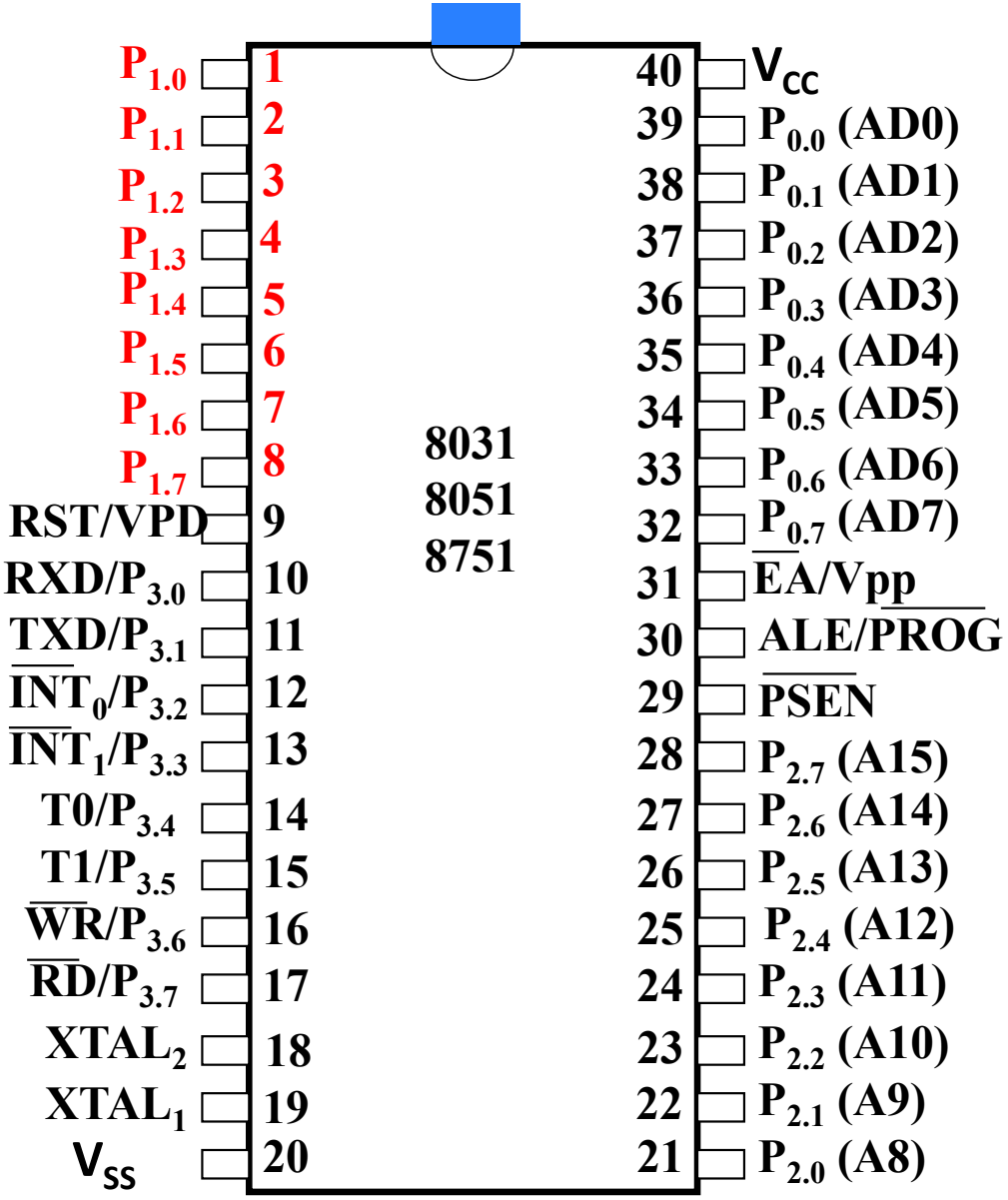
A √

B ×

在输入时，必须先向其锁存器输出1，使得输出驱动电路中的T_2截至，将端口设置为输入方式

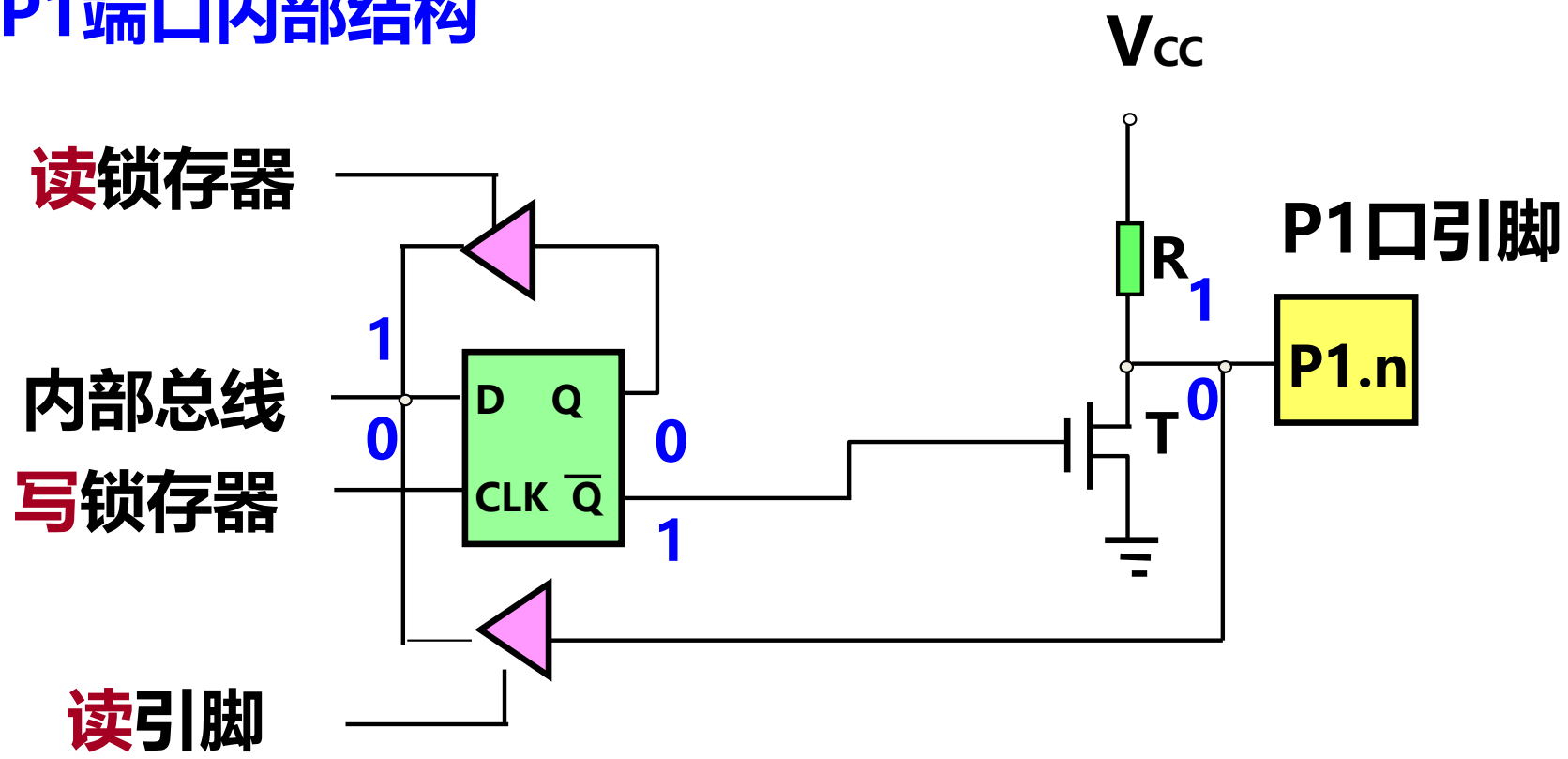
提交

●P1端口



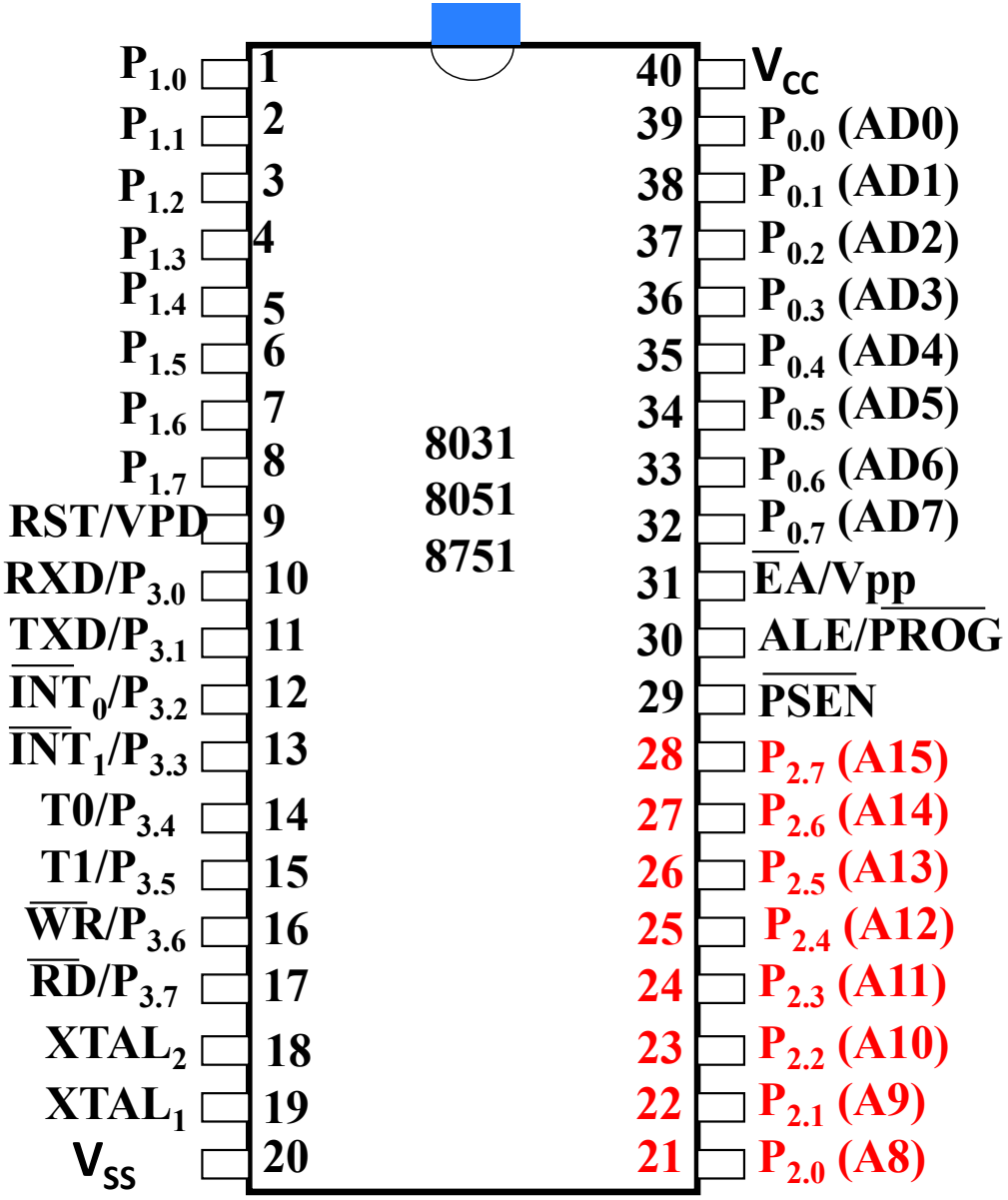
- P1.0～P1.7，占据Pin1～Pin8共8个引脚。
- P1端口一般只用作通用I/O端口。

P1端口内部结构



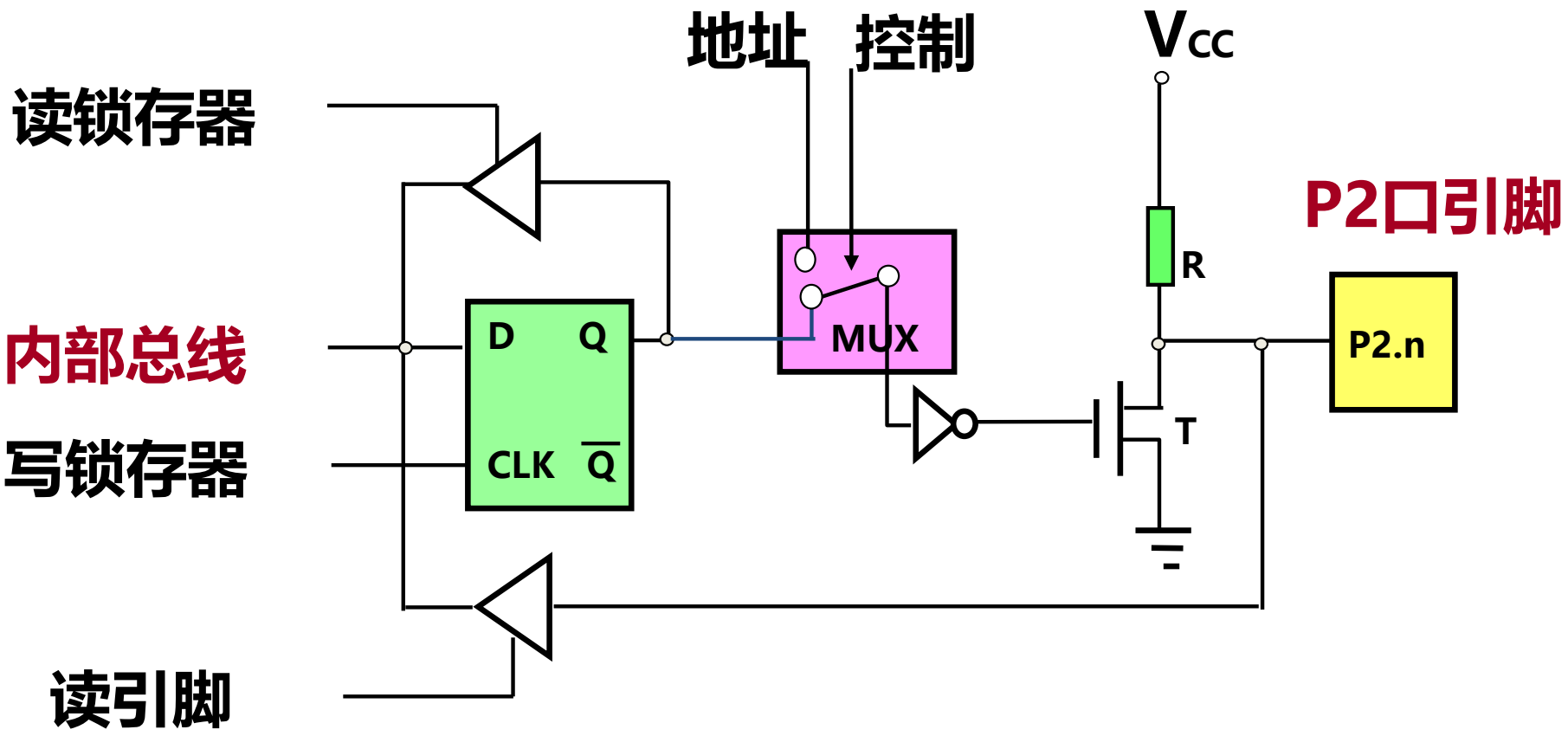
由于内部有上拉电阻，因此做I/O端口使用时不需要外部上拉电阻。

●P2端口



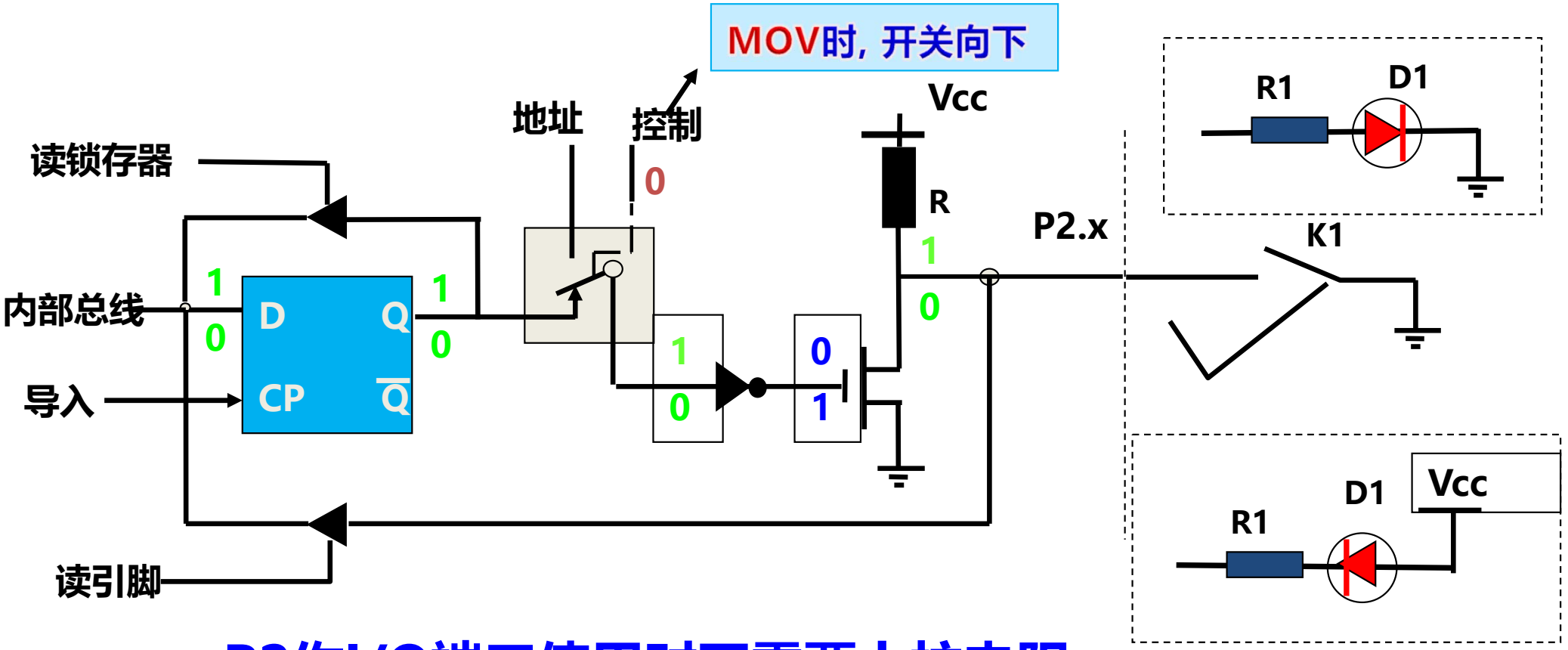
- P2.0~P3.7，占据Pin21~Pin28共8个引脚。
- P2端口可以用作通用I/O端口，或者在扩展外部存储器时用作高8位地址线。

P2端口内部结构



P2作为普通I/O端口

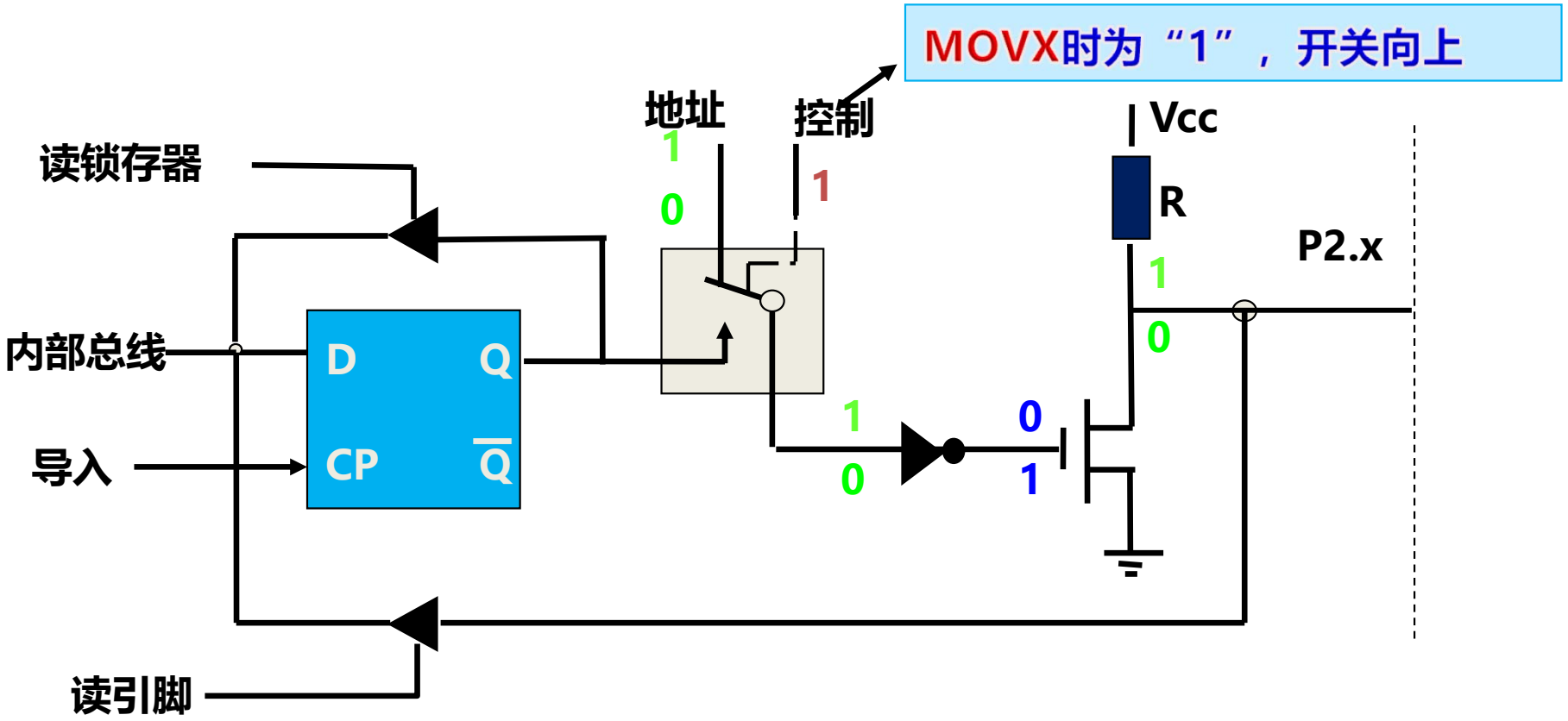
CPU发出控制电平“0”，使多路开关MUX倒向锁存器输出Q端，构成一个准双向口。其功能与P1相同。



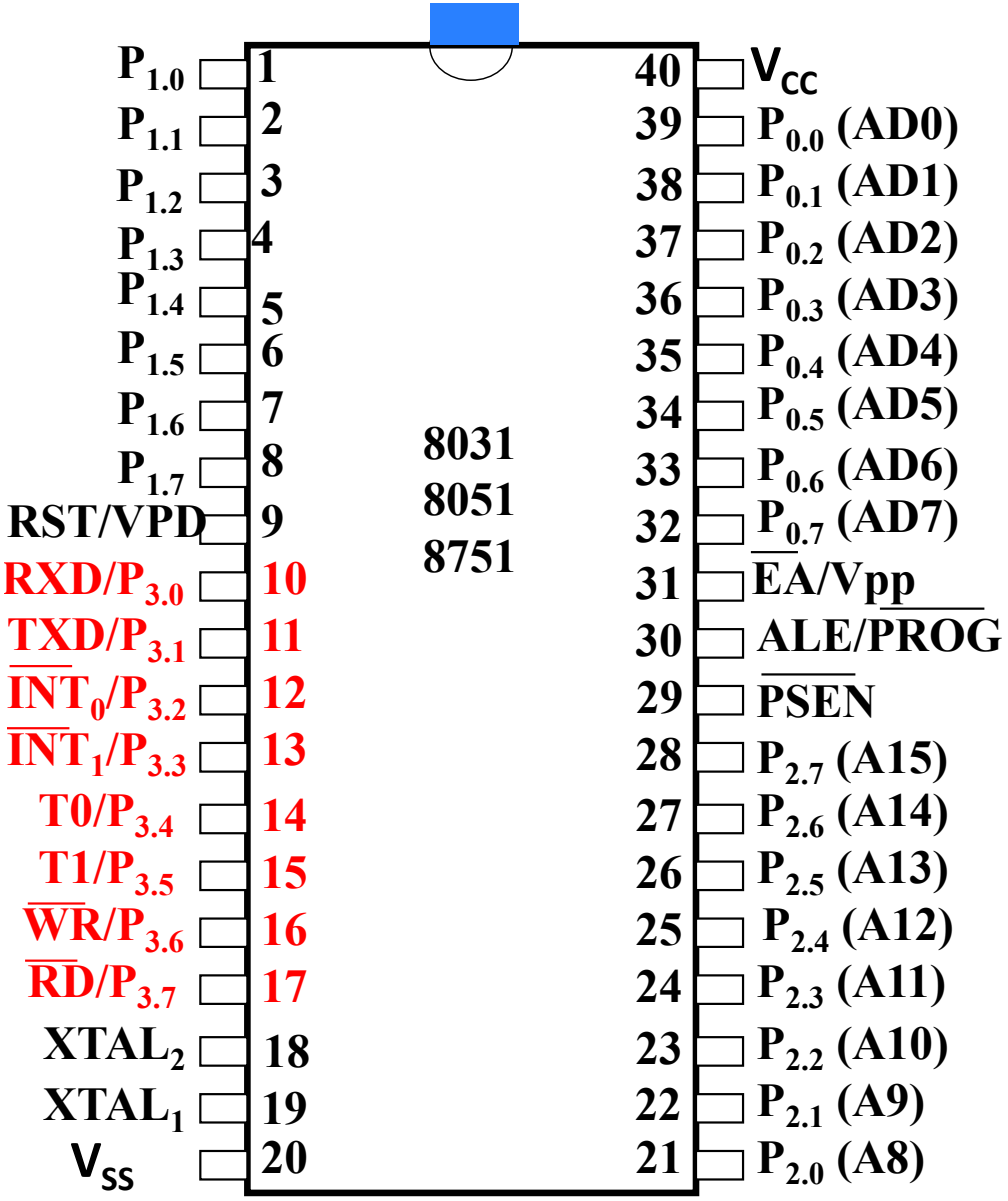
P2作I/O端口使用时不需要上拉电阻

P2作为地址线使用

CPU发出控制电平“1”，使多路开关MUX倒内部地址线。此时，P2输出高8位地址。



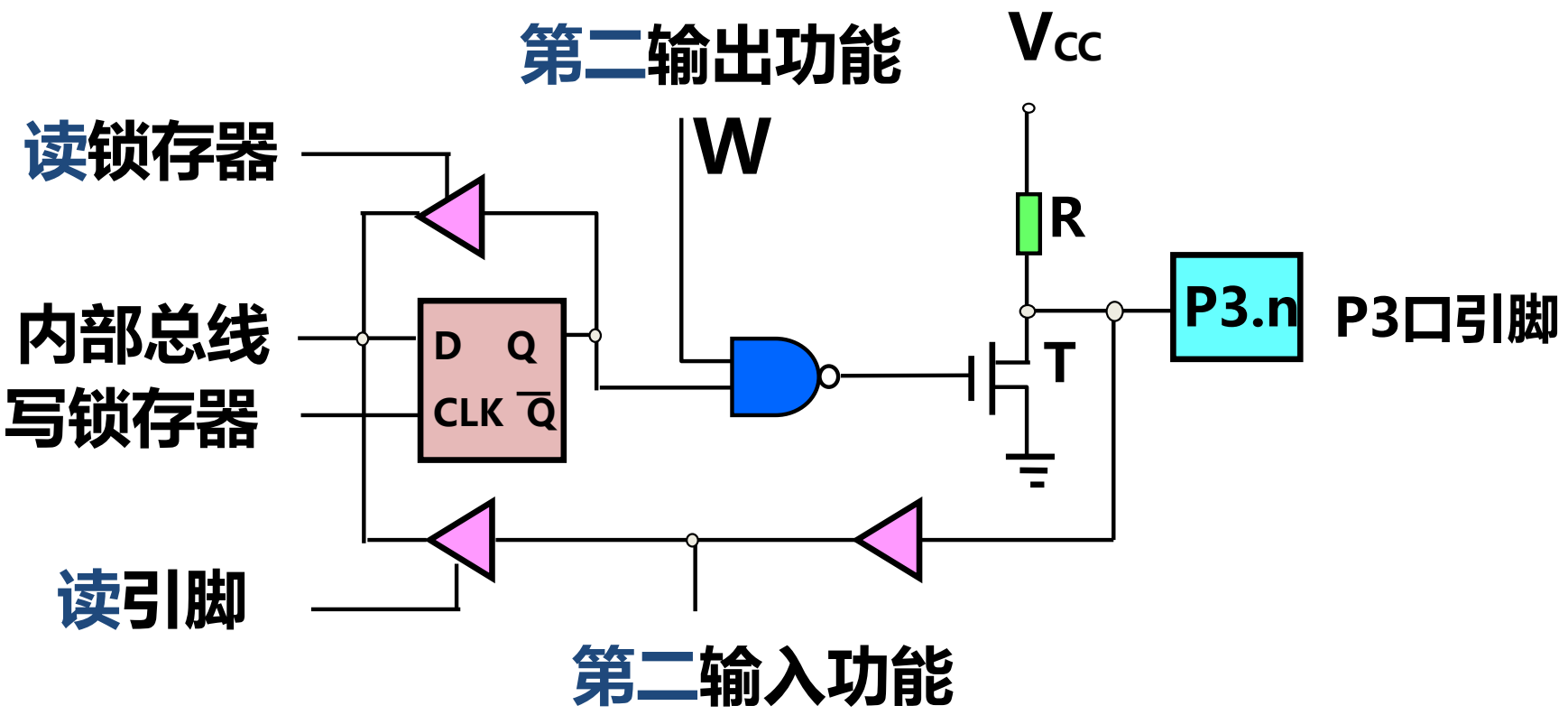
●P3端口



- P3端口即P3.0~P3.7，占据Pin10~Pin17共8个引脚。
- P3端口可以用作通用I/O端口，同时还具有特定的第二功能。

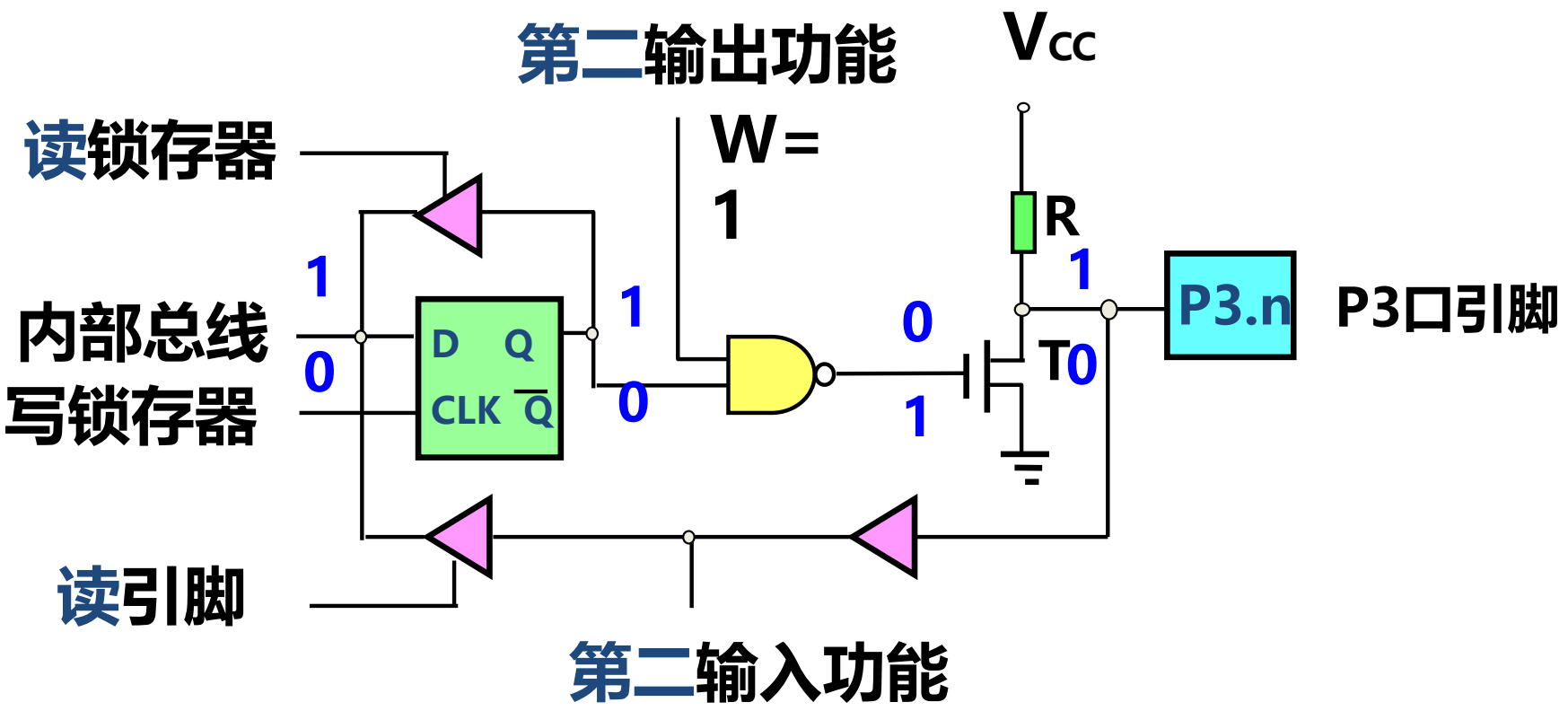
I/O 引脚	第二功能引脚名称	说明
P3.0	RXD	串行通信的数据接收端口
P3.1	TXD	串行通信的数据发送端口
P3.2	$\overline{INT0}$	外部中断 0 的请求端口
P3.3	$\overline{INT1}$	外部中断 1 的请求端口
P3.4	T0	定时/计数器 0 的外部事件计数输入端
P3.5	T1	定时/计数器 1 的外部事件计数输入端
P3.6	$\overline{RD}$	外部数据存储单元的写选通信号
P3.7	$\overline{WR}$	外部数据存储单元的读选通信号

P3端口内部结构



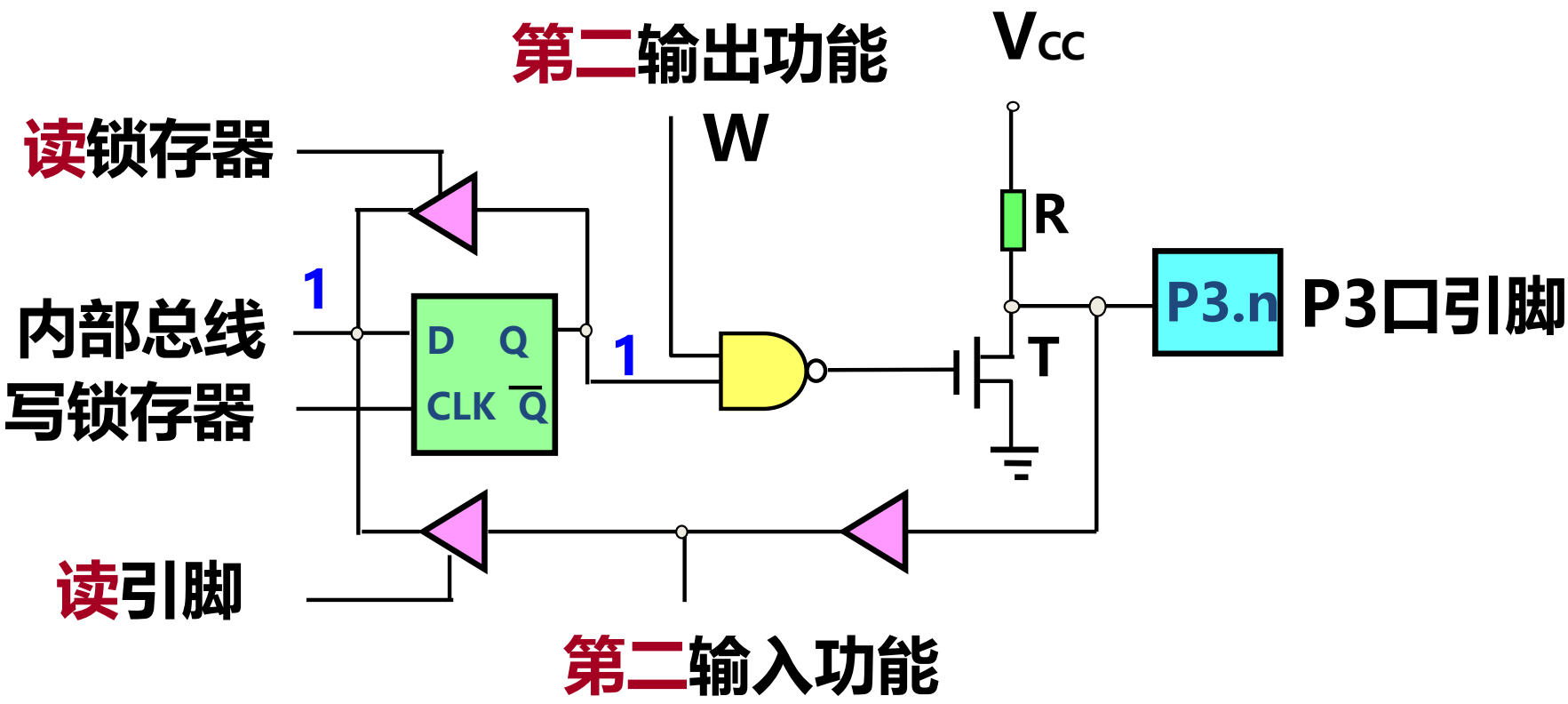
由于内部有上拉电阻，因此做I/O端口使用时不需要外部上拉电阻。

P3作为I/O端口使用



由于内部有上拉电阻，因此做I/O端口使用时不需要外部上拉电阻。

P3优先作为第二功能使用



P3口作为第二功能使用时，需要先向P3口写入1。

P0、P1、P2、P3的使用总结

1. 当系统无扩展时，P0、P2口均可以做为I/O口使用， 但P0口做I/O口使用时，必须在外接上拉电阻；

2. P1口一般作I/O使用；

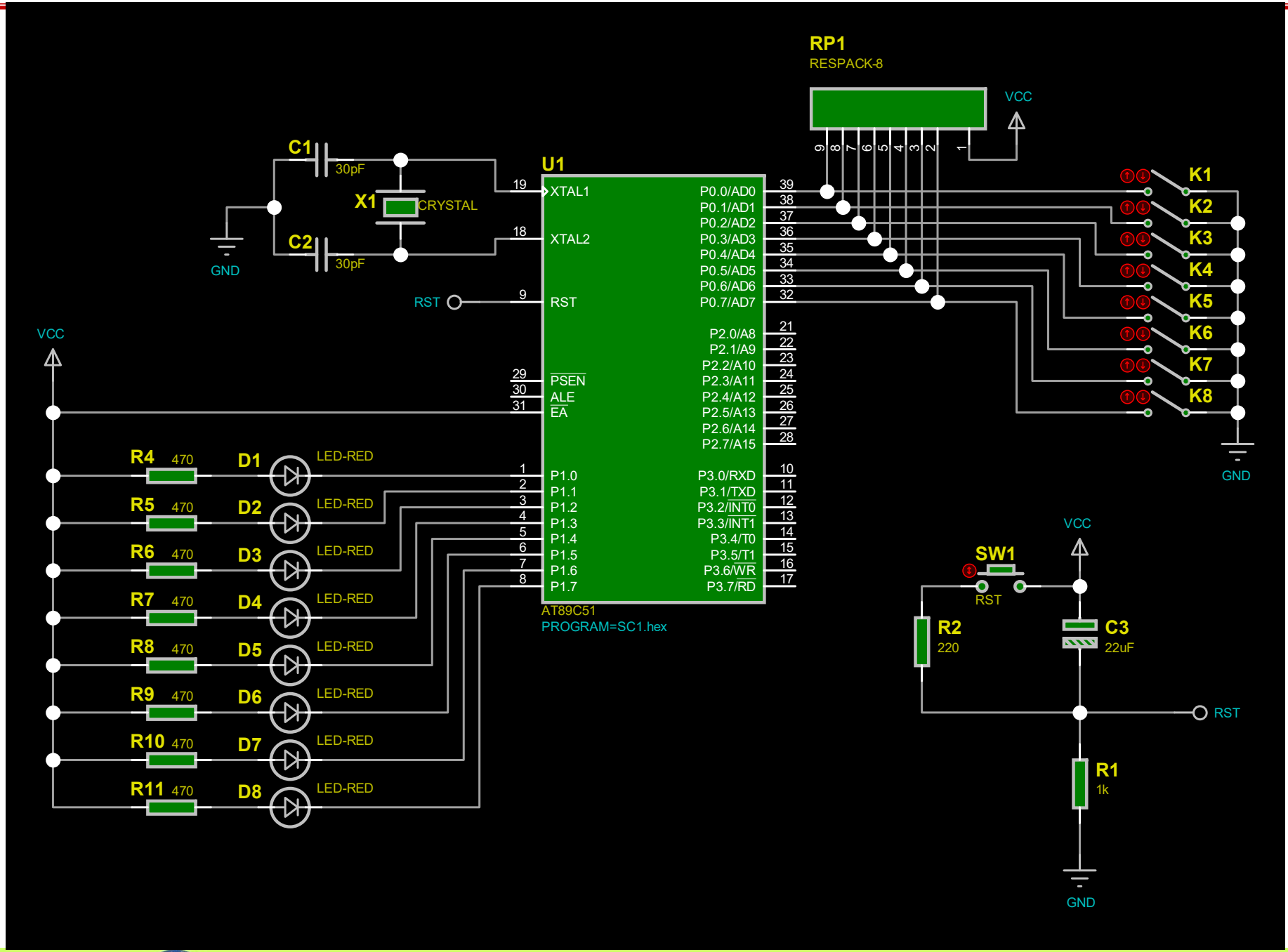
3. P3口优先使用第二功能，多余的才可用于I/O端口，当作第二功能使用时需事先向其锁存器输出1 ；

4. 所有端口在输入时，都必须先向其锁存器输出1。

   系统复位时，P0、P1、P2、P3均自动为FFH，所以其寄存器均为“1”。

注意：读端口与读引脚是不同的，端口操作实际上是对其锁存器操作，引脚操作时对引脚电平的操作。

3.4 MCS-51微控制器引脚



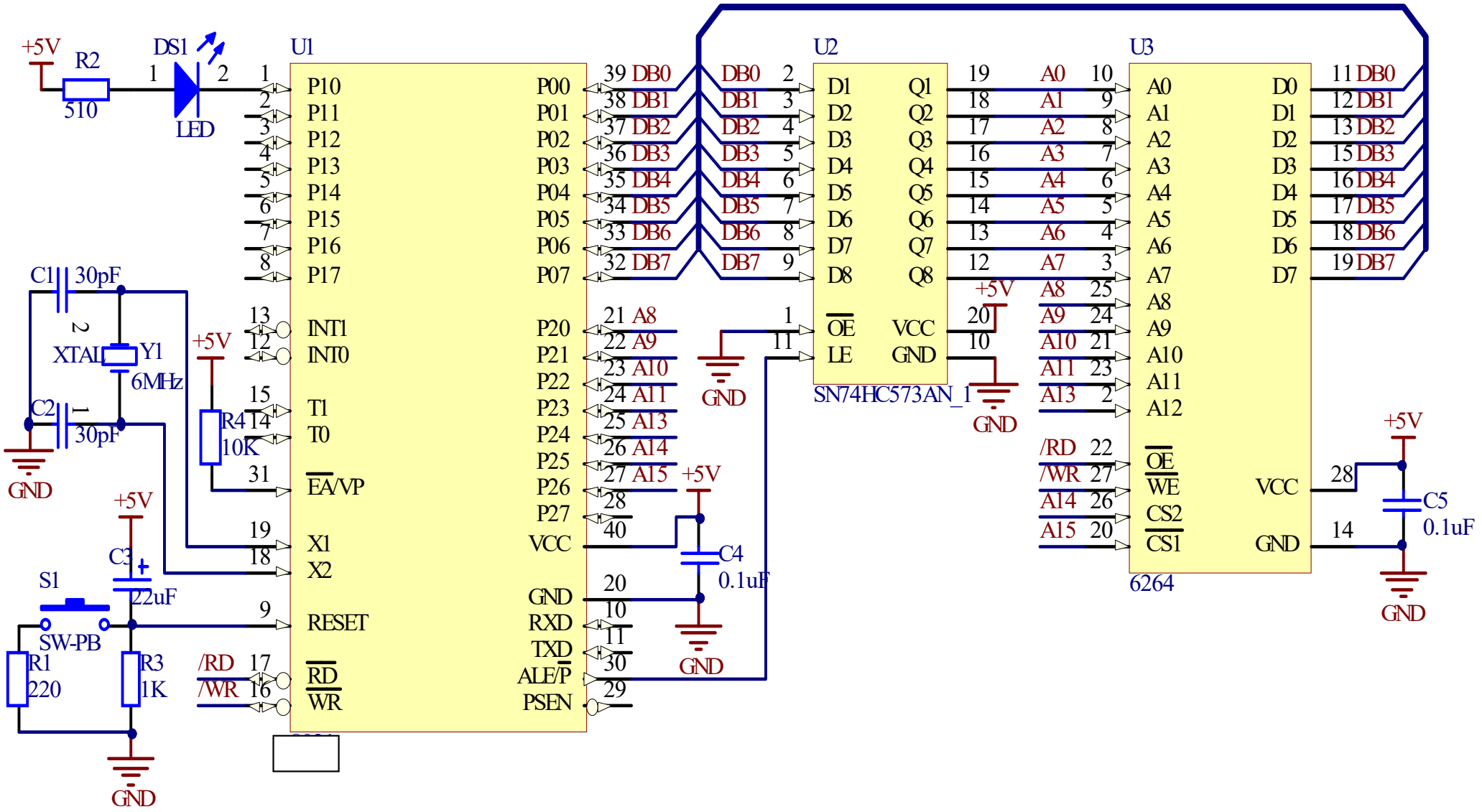
P0、P1、P2、P3的使用总结(续)

5. 当系统有扩展时，P0口首先为低8位地址和数据口使用；

6. 同样系统有扩展时，P2口首先考虑作为高8位地址使用；

7. 当P2口作为高8位地址使用时，即使没用全8位剩余的端口，一般也不允许用于I/O端口，若要使用，应该注意地址操作时对其影响；

AT89C51微控制器外部数据存储器扩展



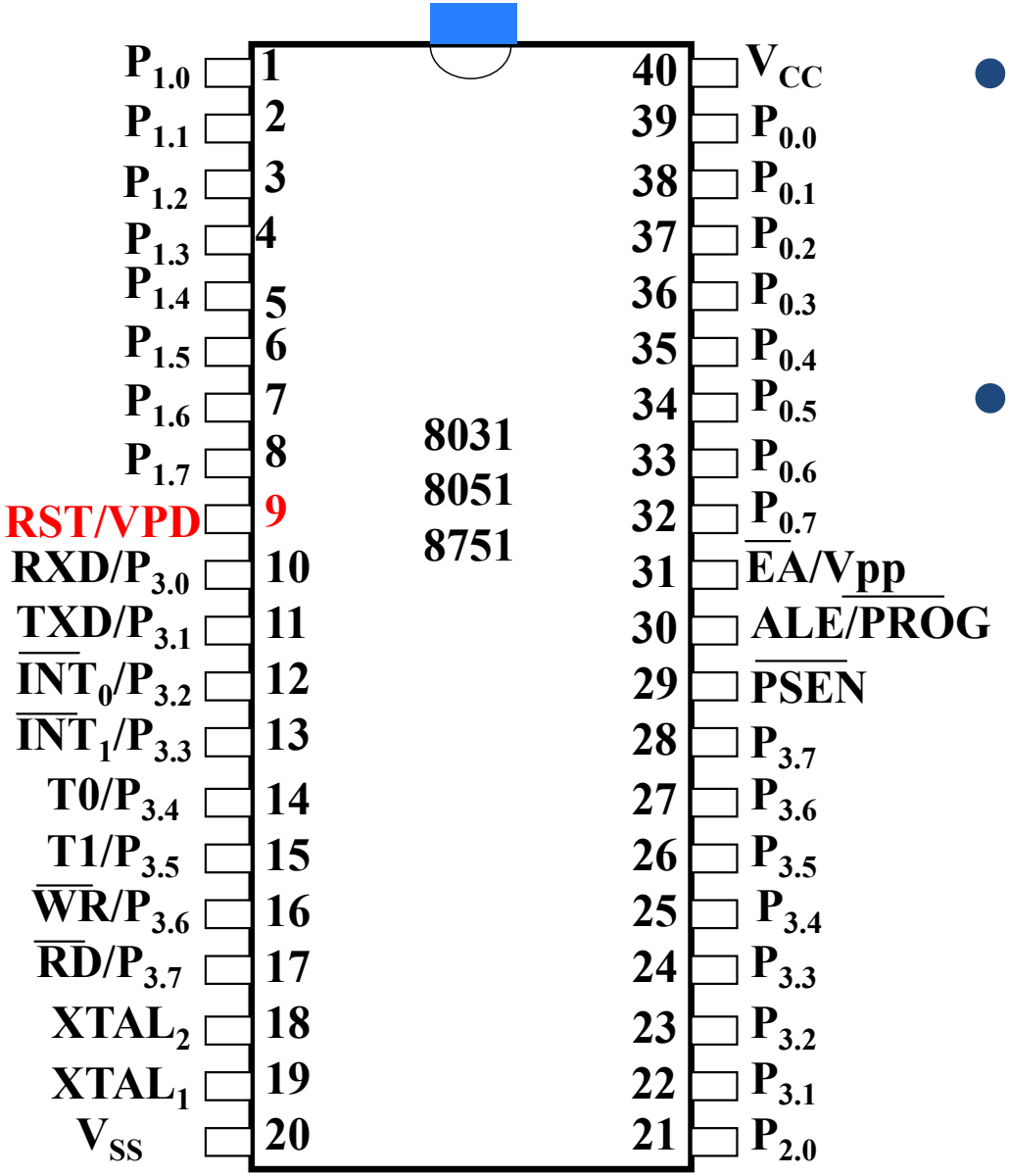
(3) 在8051微控制器中，为使准双向I/O口工作在输入方式，必须先向其输出1。

A √

B ×

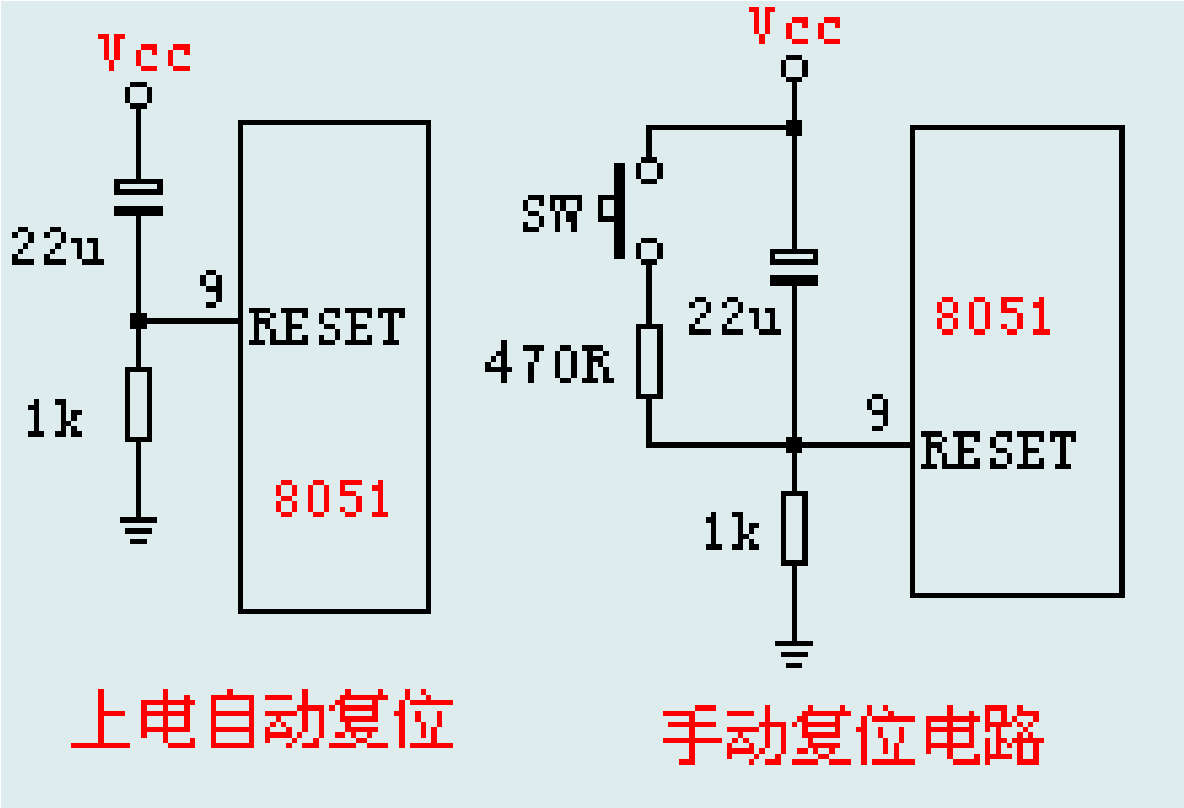
提交

●RST引脚

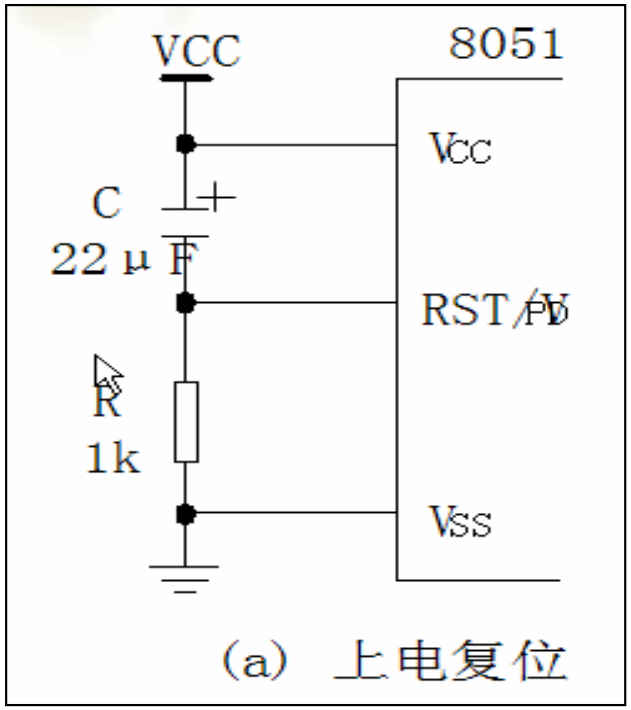


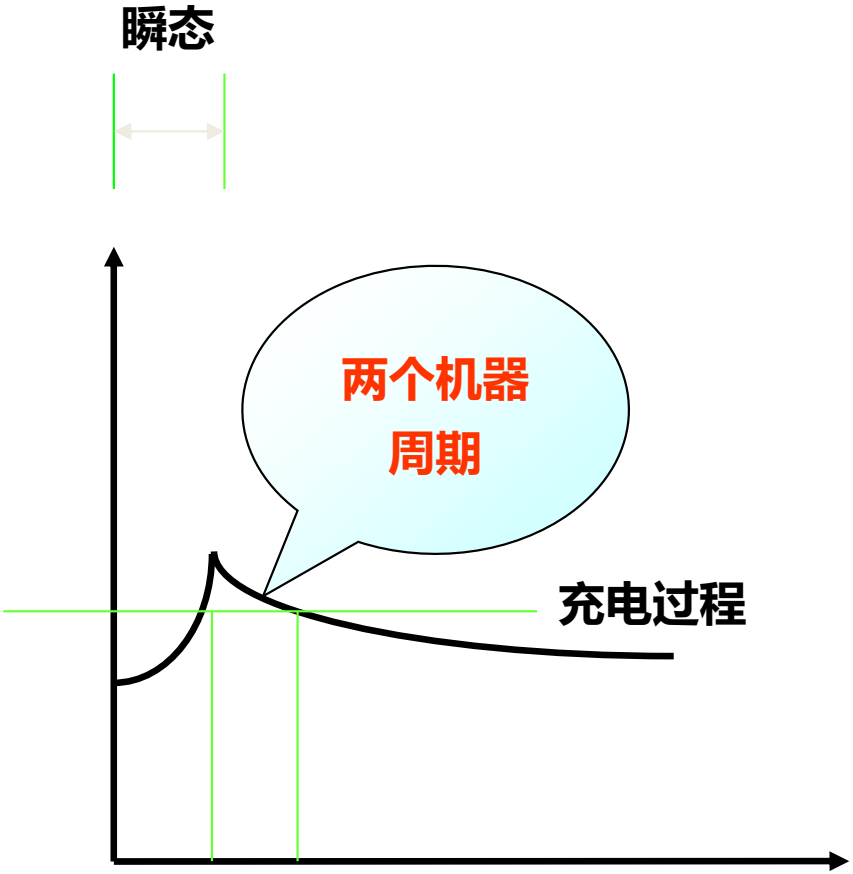
- 微控制器的复位引脚为RST（Pin9），微控制器内部CPU的复位信号便从这里输入。
- 微控制器复位完全通过RST引脚来完成，其基本原理是在微控制器的时钟振荡电路启动后，如果RST引脚外加**两个机器周期（即24个时钟振荡脉冲）以上的高电平**，微控制器便实现了复位。

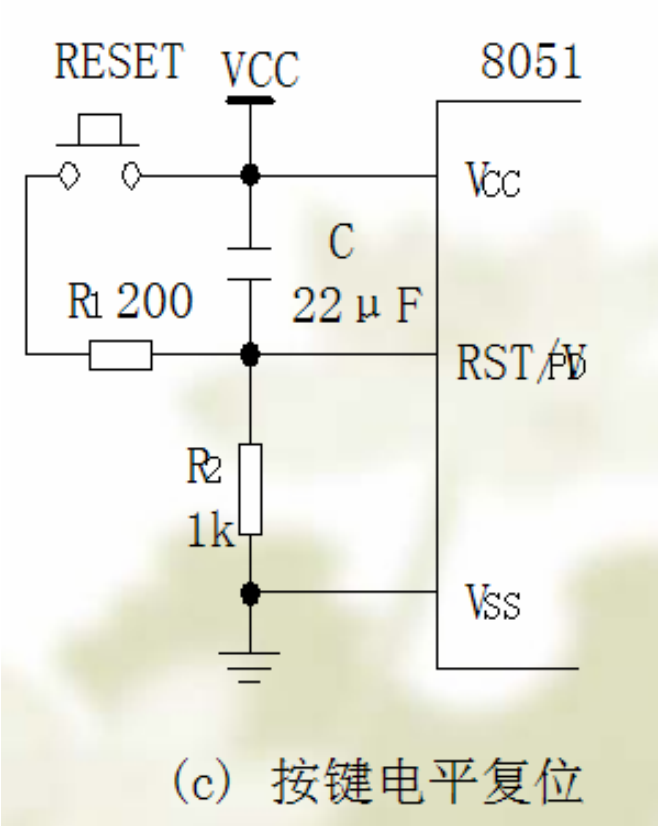
3.5 MCS-51微控制器系统复位



3.5 MCS-51微控制器系统复位







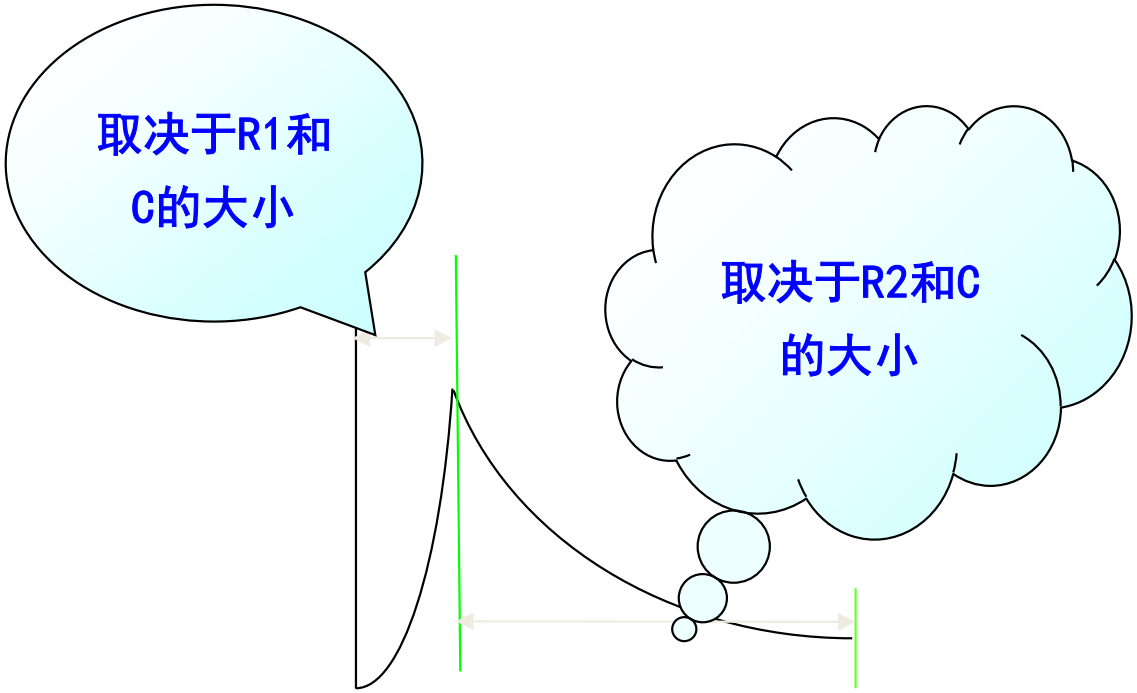
(c) 按键电平复位

a) 按下时：

电容先放电，电容电荷减少，但C正极电平不变，C负极从0上升（即RST电平）

b) 弹开后：

电容则充电



系统复位时程序指针PC=0000H，同时一些专用寄存器值自动复位

特殊功能寄存器	初始态	特殊功能寄存器	初始态
ACC	00H	B	00H
PSW	00H	SP	07H
DPH	00H	TH0	00H
DPL	00H	TL0	00H
IP	xxx00000B	TH1	00H
IE	0xx00000B	TL1	00H
TMOD	00H	TCON	00H
SCON	xxxxxxxxB	SBUF	00H
P0-P3	1111111B	PCON	0xxxxxxxB

(4) 8051微控制器复位后SP、PSW、P0的状态为：

A **00H、00H、00H**

B **00H、00H、0FFH**

C **07H、00H、0FFH**

D **07H、0FFH、00H**

提交

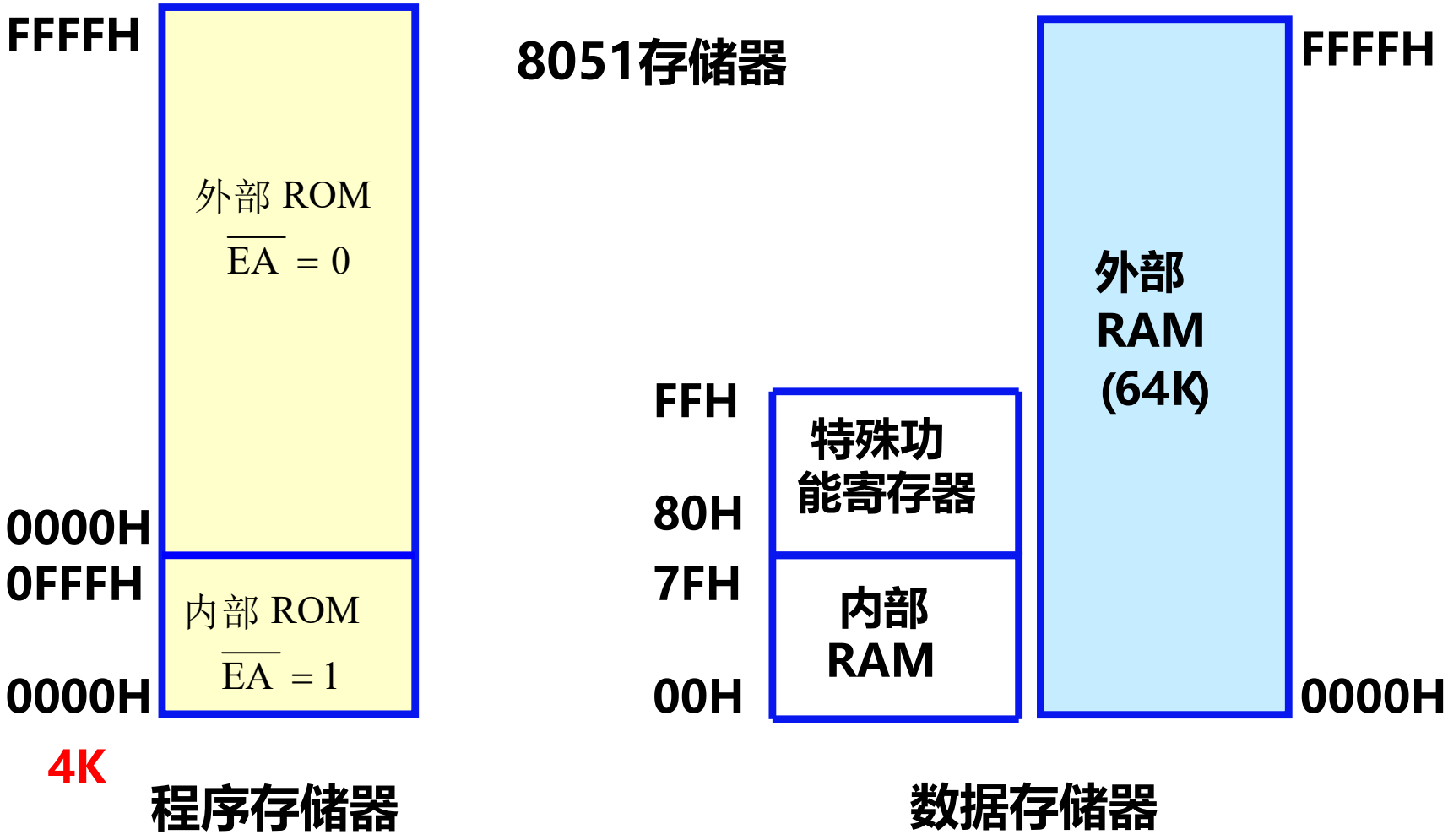
1．存储器的两种基本结构形式

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：**

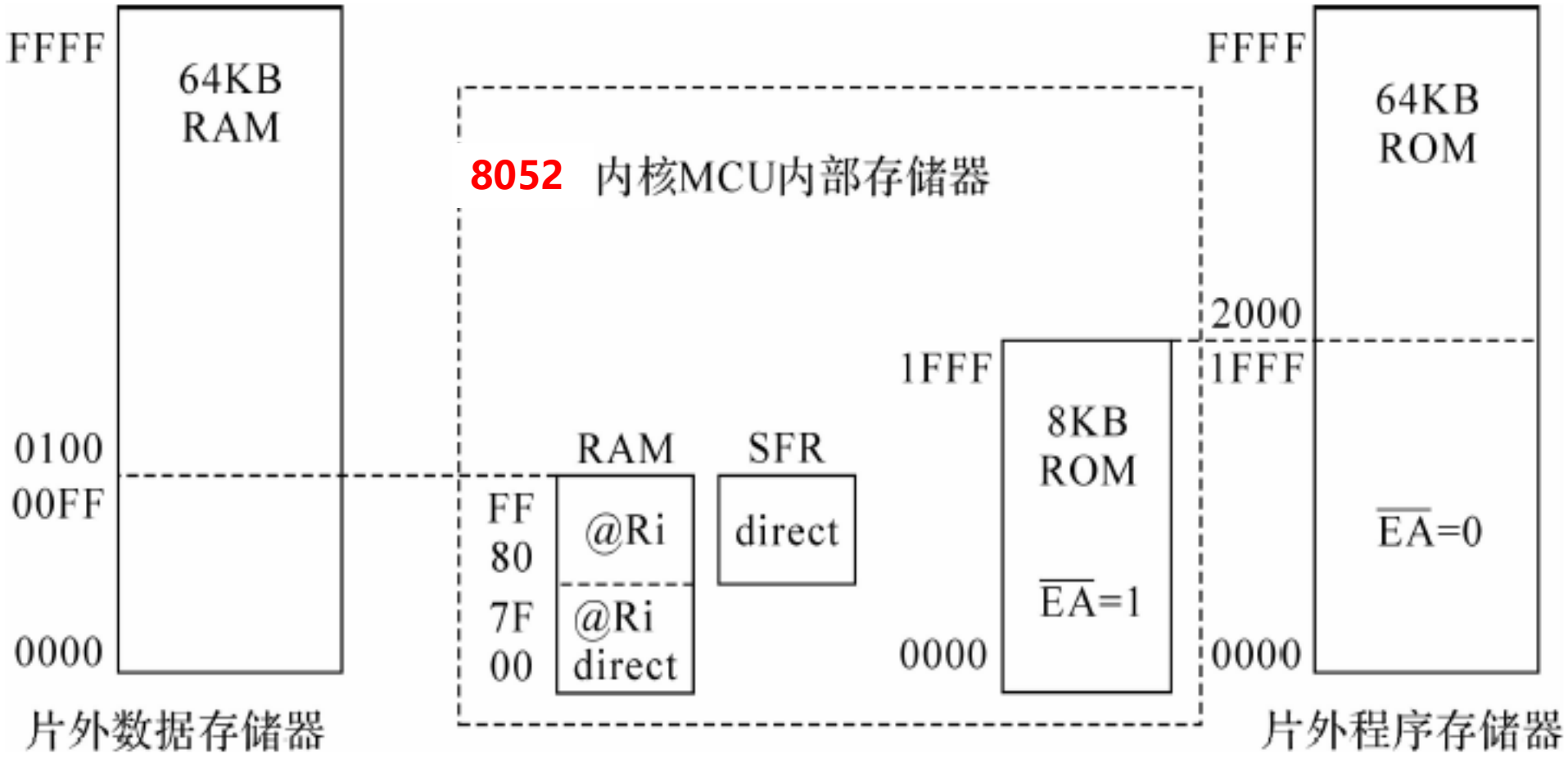
- **冯·诺依曼（Von Neumann）结构：也称普林斯顿（Princeton）结构。程序存储器和数据存储器共用一个逻辑空间，且它们是统一编址的。如16条地址线的寻址空间是64K，则ROM和RAM总共只有64K。（在通用微型计算机中广泛采用）**
- **哈佛（Harvard）结构：程序存储器和数据存储器分别编址。如16条地址线，可以分别寻址64K的ROM和64K的RAM。**

2. 8051存储器结构图

MCS-51微控制器的存储器采用哈佛结构，ROM和RAM是分开寻址的。



3.6 MCS-51微控制器存储器配置



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80C52型微控制器中，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：

口 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16K-64K的内部ROM，故不需外部扩展。

3.6.1 内部数据存储器空间配置

8051MCU存储空间与指令的关系

80C31 、80C51

80C52

间接寻址访问高128位RAM

MOV Ri，#8FH
MOV A，@Ri
MOV @Ri，A

使用 MOVX A，@DPTR

MOVX @DPTR, A

片内RAM寻址使用 MOV 指令

3.6.1 内部数据存储器空间配置

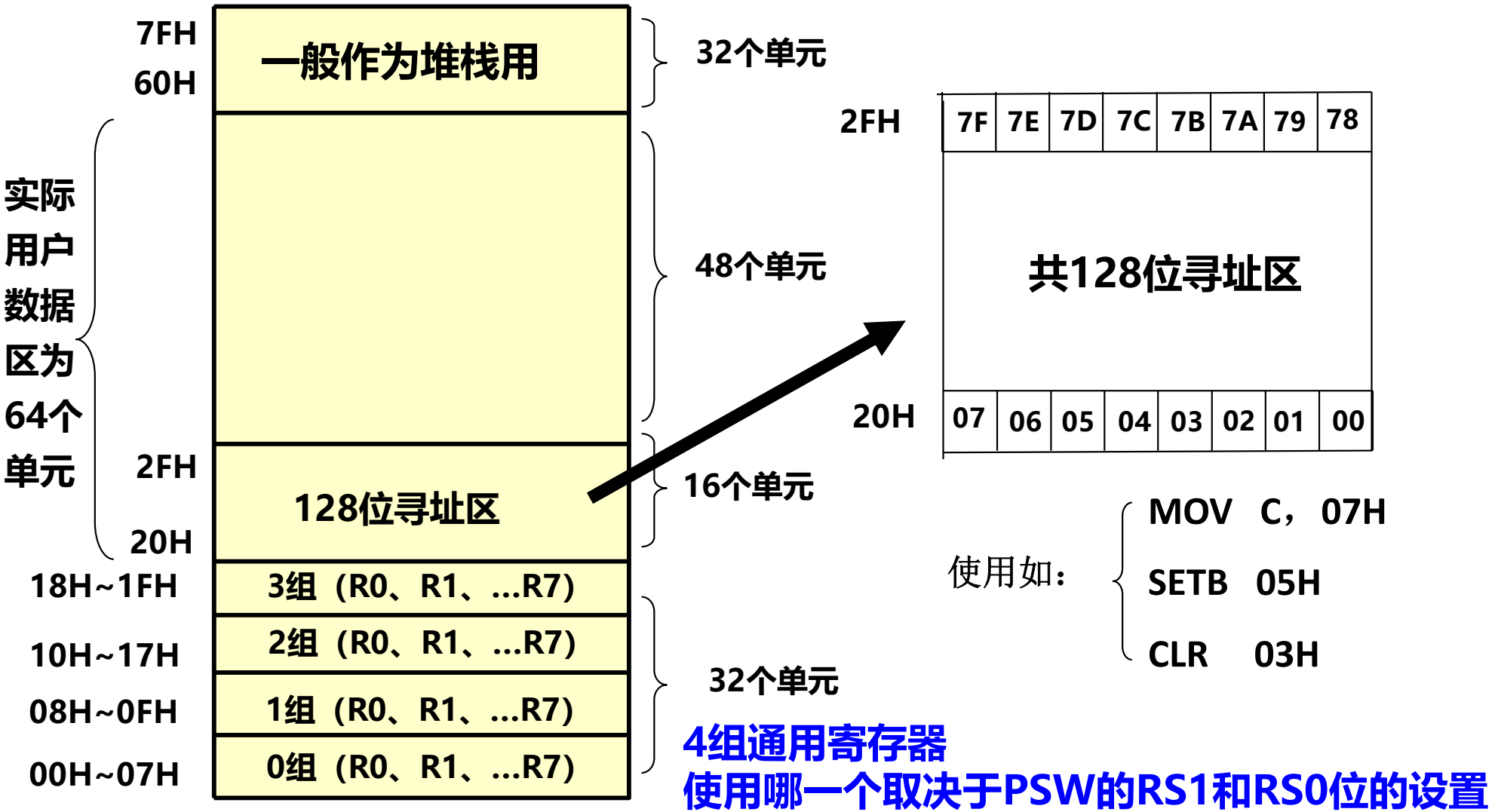
1、在微控制器中通常把RAM叫数据存储器。

2、80C51种内部RAM共有256个单元，其中低128单元也叫用户数据存储器，高128单元为专用寄存器区SFR（Special Function Register）

3、而80C52中，用户数据存储器又分为低128单元和高128单元，其中低128单元RAM与8031一样，而高128单元的地址与其专用寄存器区地址一样，但寻址方式不同，若使用直接寻址方式，则访问专用寄存器；使用间接寻址才可访问高128单元的用户数据存储器。

3.6.1 内部数据存储器空间配置

4、低128单元数据存储器又可分为三个区域



4组通用寄存器
使用哪一个取决于PSW的RS1和RS0位的设置

此外，在专用寄存器SFR中，地址值尾数为0或8的寄存器是可以位寻址的

（一）通用寄存器组区

1、工作寄存器区是指00H～1FH区，共分4个组，每组有8个单元，共32个内部RAM单元。

2、每次只能有1组作为工作寄存器使用，其它各组可以作为一般的数据缓冲区使用。

3、作为工作寄存器使用的8个单元，又称为R0～R7

4、程序状态字PSW中的PSW.3（RS0）和PSW.4（RS1）两位来选择哪一组作为工作寄存器使用。CPU通过软件修改PSW中RS0和RS1两位的状态，就可任选一个工作寄存器工作。

RS1、 RS0与片内工作寄存器组的对应关系

RS1	RS0	寄存器组	片内RAM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

工作寄存器和RAM地址对照表

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

（二）位寻址区

1、位寻址区是指 20H～2FH单元，共16个单元。

2、位寻址区的每1位都可由程序直接进行位处理。如：

```
MOV   C，07H
SETB  05H
CLR   03H
```

3、 位寻址区的16个单元（共计128位）的每1位都有一个8位表示的位地址，位地址范围为00H～7FH。

4、 同样，位寻址的RAM单元也可以按字节操作，作为一般的数据缓冲区。

(5) 下列8051微控制器内部单元中，既可位寻址又可字节寻址的单元是：

A **28H**

B **30H**

C **00H**

D **70H**

提交

（三）堆栈

堆栈是一种数据结构，就是只容许在其一端进行数据插入和删除。

数据写入堆栈称为插入运算PUSH，通常称入栈；

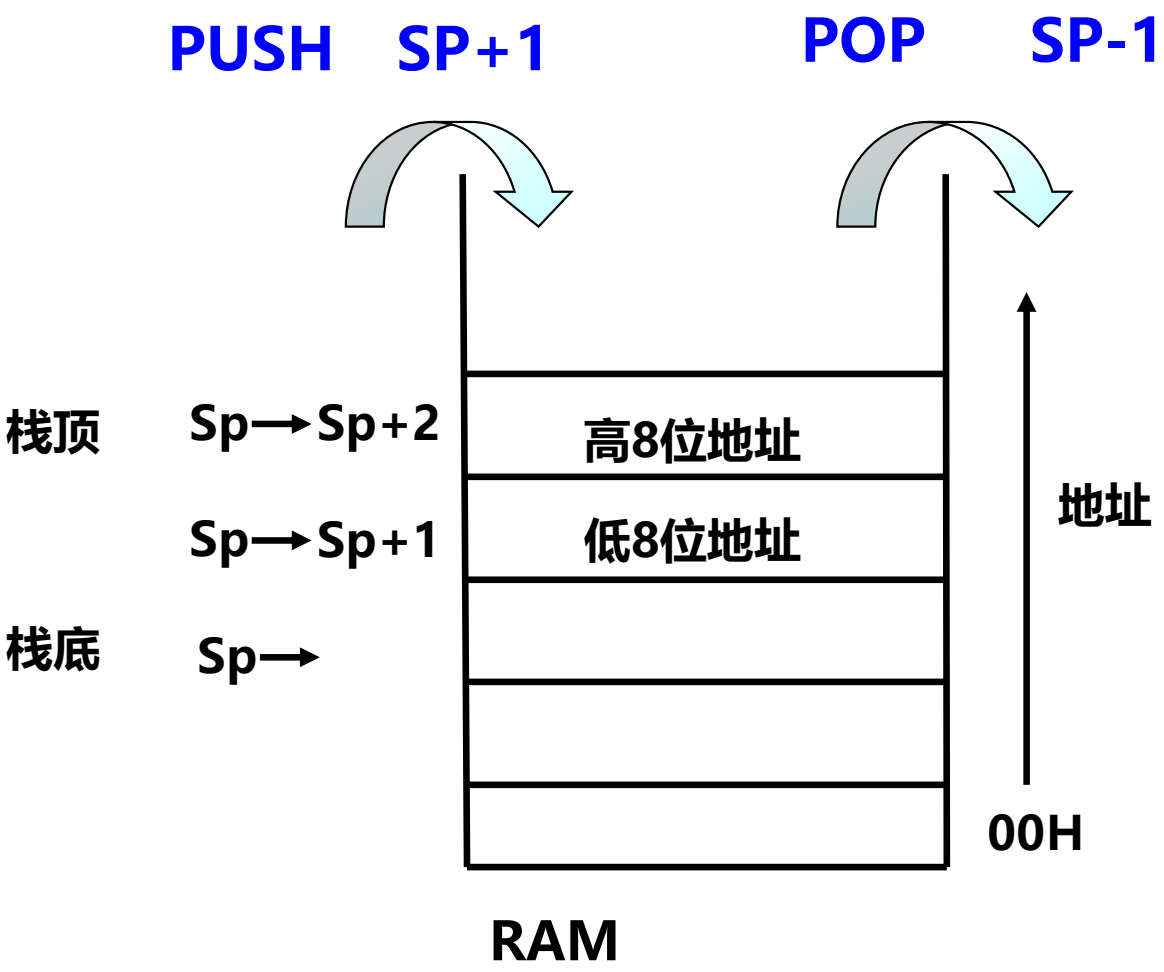
数据从堆栈读出称为复制运算POP，通常称出栈，但需要注意的是：

在微控制器中出栈后原先在堆栈中的数据并不被删除，仍在原位置保存！

在微控制器中，堆栈是在内RAM区专门开辟出来的按照“先进后出”原则进行数据存取的一块连续的存储区域

堆栈指针SP： SP用来存放堆栈栈顶的地址，8位寄存器

3.6.1 内部数据存储器空间配置



堆栈栈底：堆栈指针SP最先指向的存储单元

堆栈栈顶：最后推入堆栈的数据所在的存储单元，**堆栈指针SP**总是指向**栈顶**；

堆栈中没有数据时，栈顶和栈底二者重叠。

向堆栈推入数据后，栈顶向上生长，**堆栈指针SP** 也向上生长，**系统复位后，SP中的内容为07H** 。

堆栈的使用方法：

一种是自动方式：在调用子程序或中断时，返回地址（断点）自动进栈（先低位地址，后高位地址），程序返回时，断点自动弹回程序指针PC。

另一种是指令方式：直接使用专用的堆栈操作指令，进行进出栈操作。

进栈指令：PUSH

出栈指令：POP

进栈操作：先SP加1，后写入数据；

出栈操作：先读出数据，后SP减1

(6) 关于8051微控制器的堆栈操作，下列说法正确的是：

A 先入栈，再修改栈指针

B 先修改栈指针，再出栈

C 先修改栈指针，再入栈

D 以上都不对

提交

单选题 1分

设置

(7) MOV SP，#5FH指令是将堆栈空间设置到内部RAM 60H单元开始。

A √

B ×

提交

78

（四）专用寄存器SFR

专用寄存器位于高128单元，但在80C51微控制器中只使用了其中的21个单元。专用寄存器SFG均有名称。

分别是A，B，PSW，SP，DPH，DPL，P0，P1，P2，P3，IP，IE，TCON，TMOD，TH0，TL0，TH1，TL1，SCON，SBUF，PCON，详见书表2-5

在21个SFR中，地址最后位为0或8的SFR是可位寻址的共有11个SFR，但实际可寻址的只有83位，详见书表2-5

因此，在整个内部RAM空间共有83+128=211位可寻址

3.6.1 内部数据存储器空间配置

专用寄存器名称	符号	地址	位地址与位名称							
			D7	D6	D5	D4	D3	D2	D1	D0
P0 口	P0	80H	87	86	85	84	83	82	81	80
堆栈指针	SP	81H								
数据指针低字节	DPL DPTR DPH	82H								
数据指针高字节		83H								
定时器/计数器控制	TCON	88H	TF1 8F	TR1 8E	TF0 8D	TR0 8C	IE1 8B	IT1 8A	IE0 89	IT0 88
定时器/计数器方式控制	TMOD	89H	GATE	C/$\overline{T}$	M_1	M_0	GATE	C/$\overline{T}$	M_1	M_0
定时器/计数器 0 低字节	TL0	8AH								
定时器/计数器 1 低字节	TL1	8BH								
定时器/计数器 0 高字节	TH0	8CH								
定时器/计数器 1 高字节	TH1	8DH								
P1 口	P1	90H	97	96	95	94	93	92	91	90
电源控制	PCON	97H	SMOD	—	—	—	GF1	GF0	PD	IDL
串行控制	SCON	98H	SMO 9F	SM1 9E	SM2 9D	REN 9C	TB8 9B	RB8 9A	TI 99	RI 98

3.6.1 内部数据存储器空间配置

专用寄存器名称	符号	地址	位地址与位名称							
			D7	D6	D5	D4	D3	D2	D1	D0
串行数据缓冲器	SBUF	99H								
P2 口	P2	A0H	A7	A6	A5	A4	A3	A2	A1	A0
中断允许控制	IE	A8H	EA AF	—	ET2 AD	ES AC	ET1 AB	EX1 AA	ET0 A9	EX0 A8
P3 口	P3	B0H	B7	B6	B5	B4	B3	B2	B1	B0
中断优先级控制	IP	B8H	— —	— —	PT2 BD	PS BC	PT1 BB	PX1 BA	PT0 B9	PX0 B8
定时器/计数器 2 控制	T2CON *	C8H	TE2 CF	EXF2 CE	RCLK CD	TCLK CC	EXEN2 CB	TR2 CA	$C/\overline{T2}$ C9	$\overline{CP}/RL2$ C8
定时器/计数器 2 自动重装载低字节	RLDL *	CAH								
定时器/计数器 2 自动重装载高字节	RLDH *	CBH								
定时器/计数器 2 低字节	TL2 *	CCH								
定时器/计数器 2 高字节	TH2 *	CDH								
程序状态字	PSW	D0H	Cr D7	AC D6	F0 D5	RS1 D4	RS0 D3	OV D2	— D1	P D0
累加器	A	E0H	E7	E6	E5	E4	E3	E2	E1	E0
B 寄存器	B	F0H	F7	F6	F5	F4	F3	F2	F1	F0

（五）程序状态字PSW

CY	AC	F0	RS1	RS0	OV	F1	P

CY —— 进位/借位标志；位累加器。

AC —— 辅助进/借位标志；用于十进制调整。

F0 —— 用户定义标志位；软件置位/清零。

OV —— 溢出标志； 硬件置位/清零。

P —— 奇偶标志；A中1的个数为奇数，P = 1；否则 P = 0。

RS1、RS0 ——寄存器区选择控制位。

0 0： 0区 R0 ~ R7
0 1： 1区 R0 ~ R7
1 0： 2区 R0 ~ R7
1 1： 3区 R0 ~ R7

Cy(PSW.7)----溢出位　　在程序中用C表示

AC(PSW.6)----辅助进位　　在十进制数加减运算中，需要数据低4位是否有进位 即AC是否为1进行调整而用到

是自动进行的

- DA A 指令　　用于对BCD码十进制（加法）
- AC=1 时　　低4位加6修正　　A←(A)+06H
- Cy=1 时或高4位大于9,则高4位加6修正　　实际上　　A←(A)+60H
- Cy=1, AC=1　　则　　A←(A)+66H

F0(PSW.5)-------用户标志位.实际上20~2F单元中的128位均可作为用户标志位

RS1和RS0(PSW.4和PSW.3) 寄存器选择位

RS1	RS0	寄存器组	R0~R7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

OV(PSW.2)

加减运算时：第6位向第7位进位，第7位向第8位进位中有一次，则OV=1

无一次，则OV=0

3.6.1 内部数据存储器空间配置

除法运算时，　除数为0时OV=1

乘法时，　乘积超255，溢出时为OV =1

P(PSW.0)

每个指令周期由硬件根据累加器A中的“1”的个数进行置位或复位.

若1的个数为偶数　则P=0

若1的个数为奇数　则P=1

(8) 若RS1=1，RS0=0，则当前使用的工作寄存器组是:

A **第0组**

B **第1组**

C **第2组**

D **第3组**

提交

单选题 1分

设置

(9) 已知A的值为98H，将其与0FAH相加。则标志位Cy、AC、OV、P的值分别是:

A **0、0、0、1**

B **1、0、1、0**

C **1、1、1、1**

D **1、1、0、1**

提交

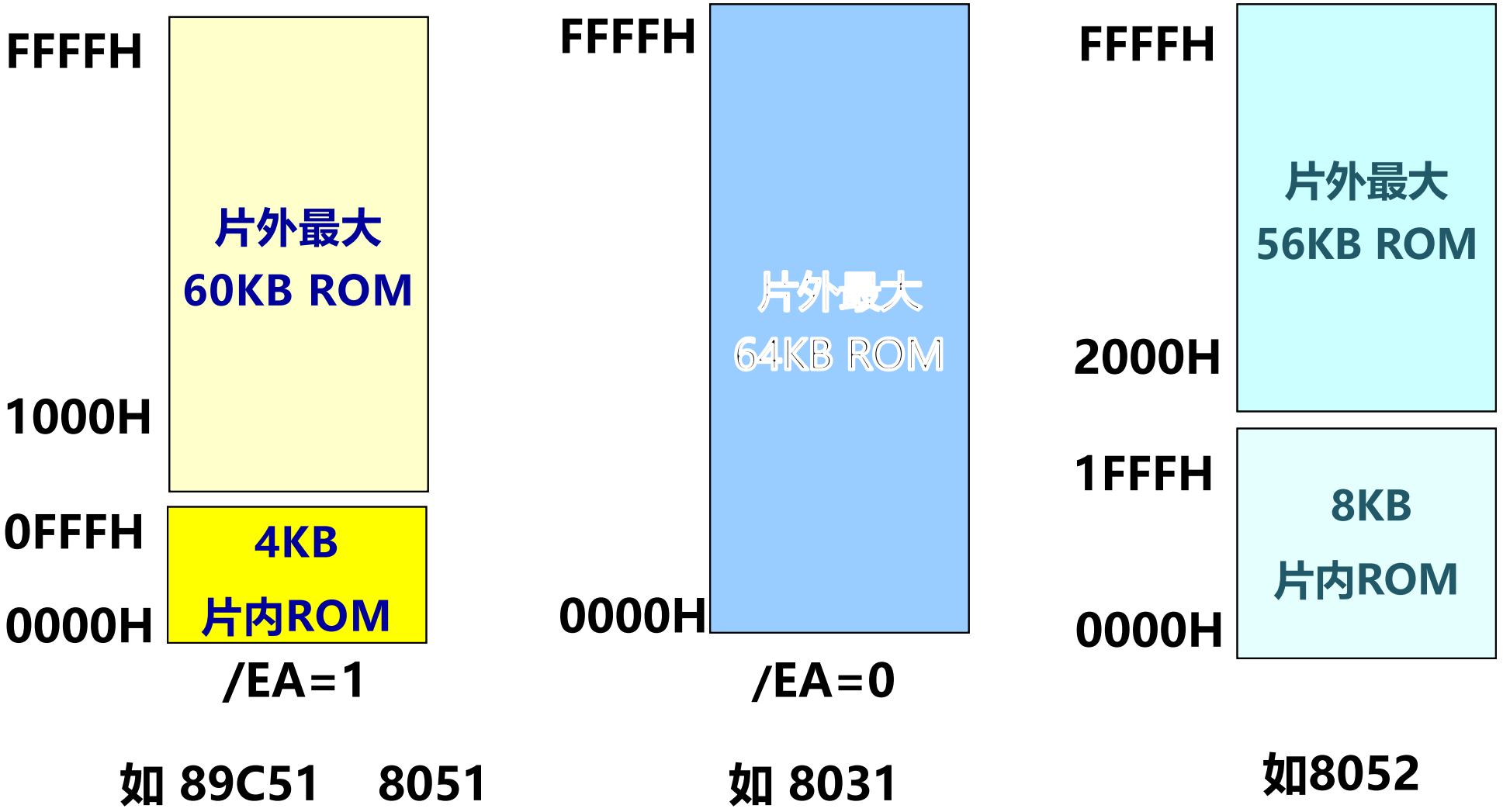
(七) 数据指针DPTR

1、数据指针DPTR是一个16位的专用寄存器，其高位字节寄存器用DPH表示，低位字节寄存器用DPL表示。

2、既可作为一个16位寄存器DPTR来处理，也可作为两个独立的8位寄存器DPH和DPL来处理。

3、DPTR 主要用来存放16位地址，当对64 KB外部数据存储器空间寻址时，作为间址寄存器用。在访问程序存储器时，用作基址寄存器。

1.程序存储器ROM


FFFFH
片外最大
60KB ROM
1000H
0FFFH
4KB
片内ROM
0000H
/EA=1
如 89C51 8051
FFFFH
片外最大
64KB ROM
0000H
/EA=0
如 8031
FFFFH
片外最大
56KB ROM
2000H
1FFFH
8KB
片内ROM
0000H
如8052

3.6.2 程序存储器空间配置

统一使用　　MOVC　A，@A+DPTR ⟶ 主要用与查表

MOVC　A，@A+PC

由于A最多为256，PC基址是固定的，DPTR是可以改变的

所以MOVC　　A，@A+DPTR更灵活

2、程序存储器的特别区域

0000H ~ 002AH （80C51微控制器）

0000H~0002H —— PC指针复位处

0003H~000AH —— 外部中断0中断地址区

000BH~0012H —— 定时器/计数器0中断区

0013H~001AH —— 外部中断1中断地址区

001BH~0022H —— 定时器/计数器1中断区

0023H~002AH —— 串行中断地址区

3.6.2 程序存储器空间配置

由于PC复位时指向0000H，同时从0003H开始又是中断区，此外，每个中断区也只有8个单元，也无法存放一般的中断程序，所以在这些地方往往放一条无条件转移指令。

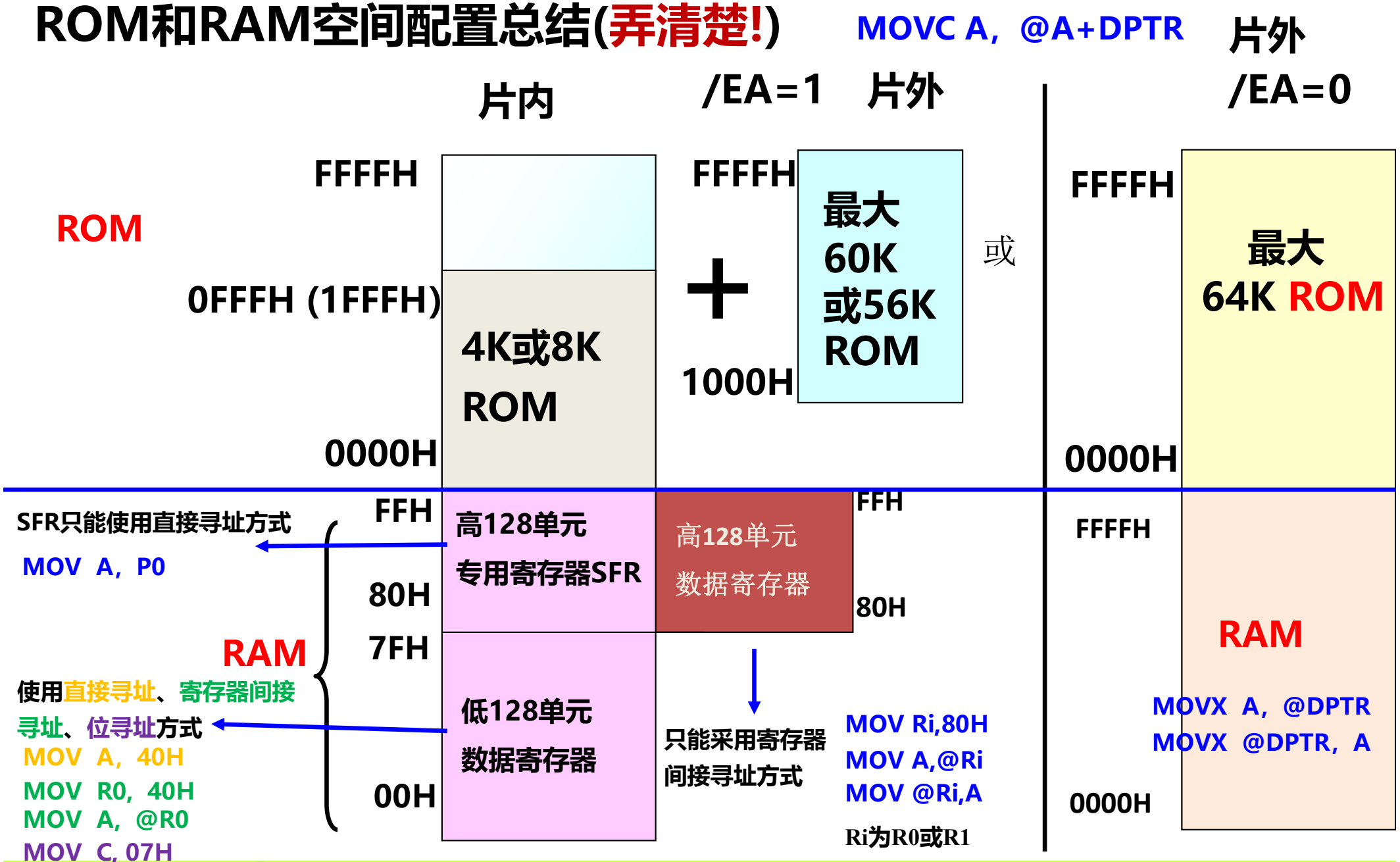
例如：程序中有2个中断服务程序，一个是定时中断服务程序，使用了定时器T0； 一个是串行中断服务程序，使用了串行口。则：

```
        ORG     0000H
        AJMP    MAIN

        ORG     000BH
        AJMP    D3SEC

        ORG     0023H
        AJMP    CHTX

        ORG     0100H
MAIN:   MOV     SP,  #60H
        …
```

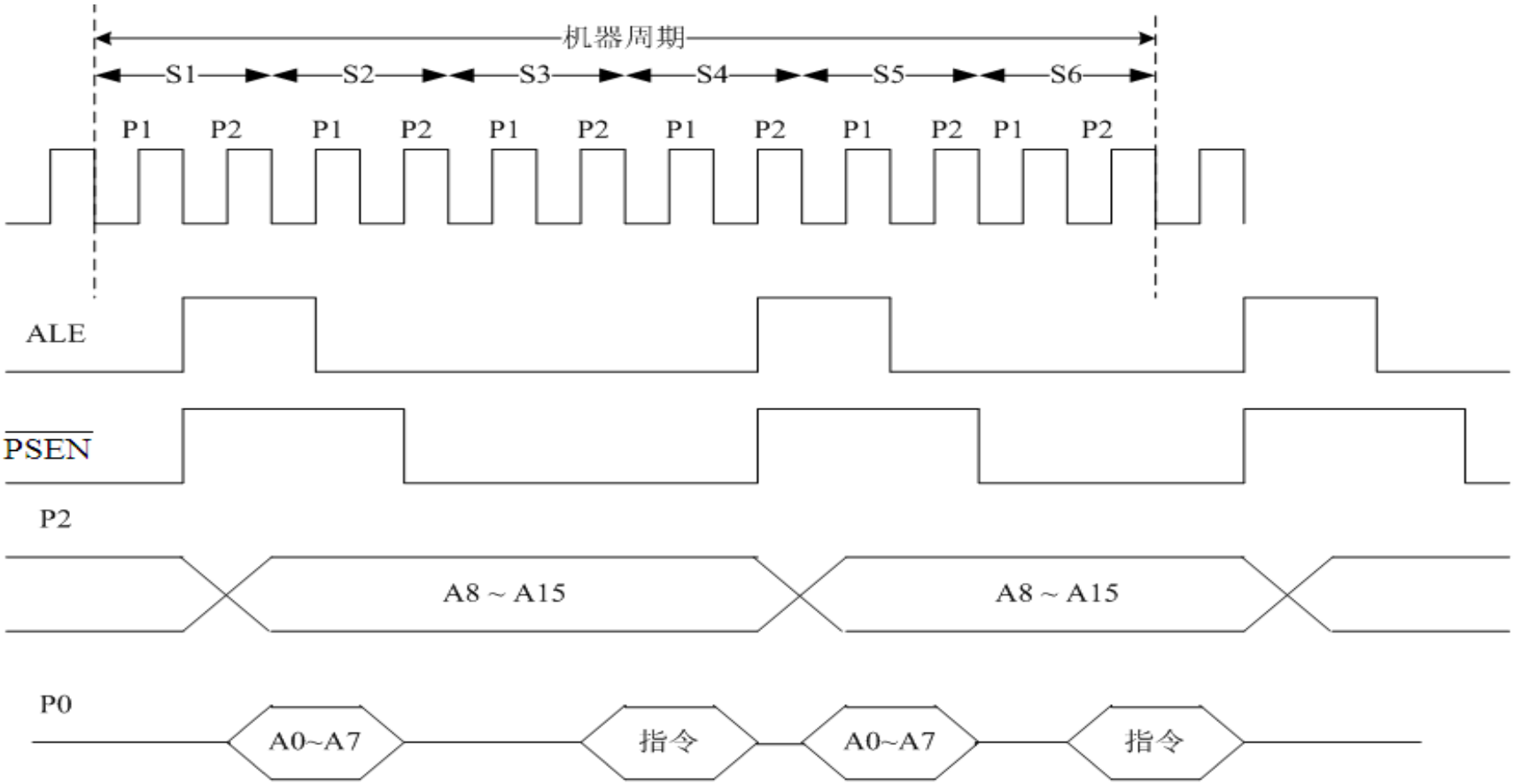
```
D3SEC:  PUSH    PSW
        PUSH    ACC
        …
        POP     ACC
        POP     PSW
        RETI

CHTX:   CLR     ES
        PUSH    PSW
        PUSH    ACC
        …
        SETB    ES
        POP     ACC
        POP     PSW
        RETI
```

3.6 MCS-51微控制器存储器配置

ROM和RAM空间配置总结(弄清楚!)

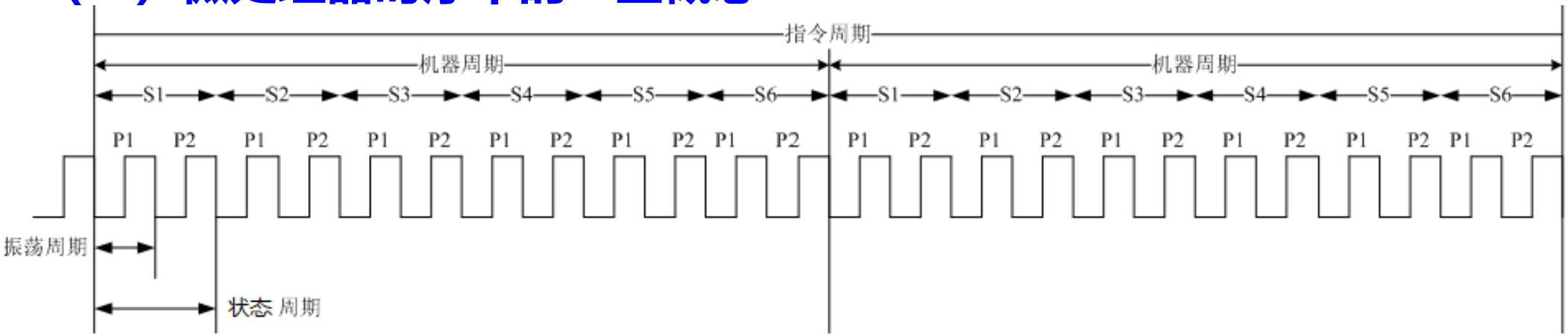


3.7 指令时序



所谓时序，是指MCU在执行指令过程中，其控制器所发出的一系列特定的控制信号在时间上的相互关系

（一）微处理器时序中的一些概念



●**时钟周期**：也称**振荡周期**，是单片机外部晶振的倒数。**是单片机CPU中最基本的时间单元**。一个时钟周期定义一个**节拍**，用P_i表示。

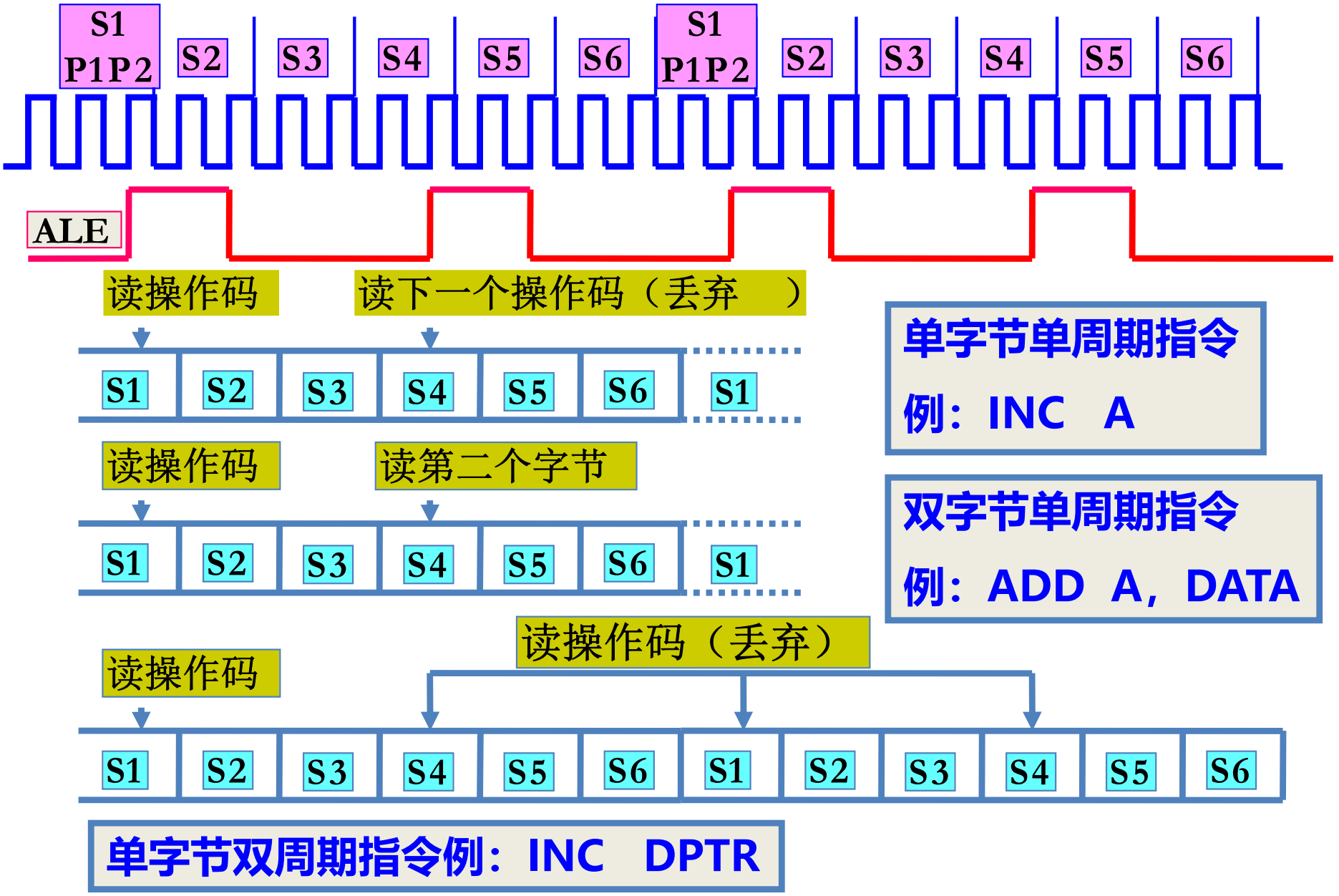
●**状态周期**：在一个状态周期内，CPU仅完成一个最基本的动作，用S_i表示。它是时钟周期的两倍S =2T=2/ fosc ，即由连续的**两个节拍**P1和P2组成。

●**机器周期**：由6个时钟周期（S1～S6）构成，也就是**12个节拍**组成，它是微控制器的基本操作周期。

●**指令周期**：即从取指到执行完，所需时间。

不同机器指令周期不一样；即使相同机器，不同的指令其指令周期也不一样。指令周期含若干机器周期（单、双、四周期）

3.7 指令时序



单字节单周期指令

例：INC A

双字节单周期指令

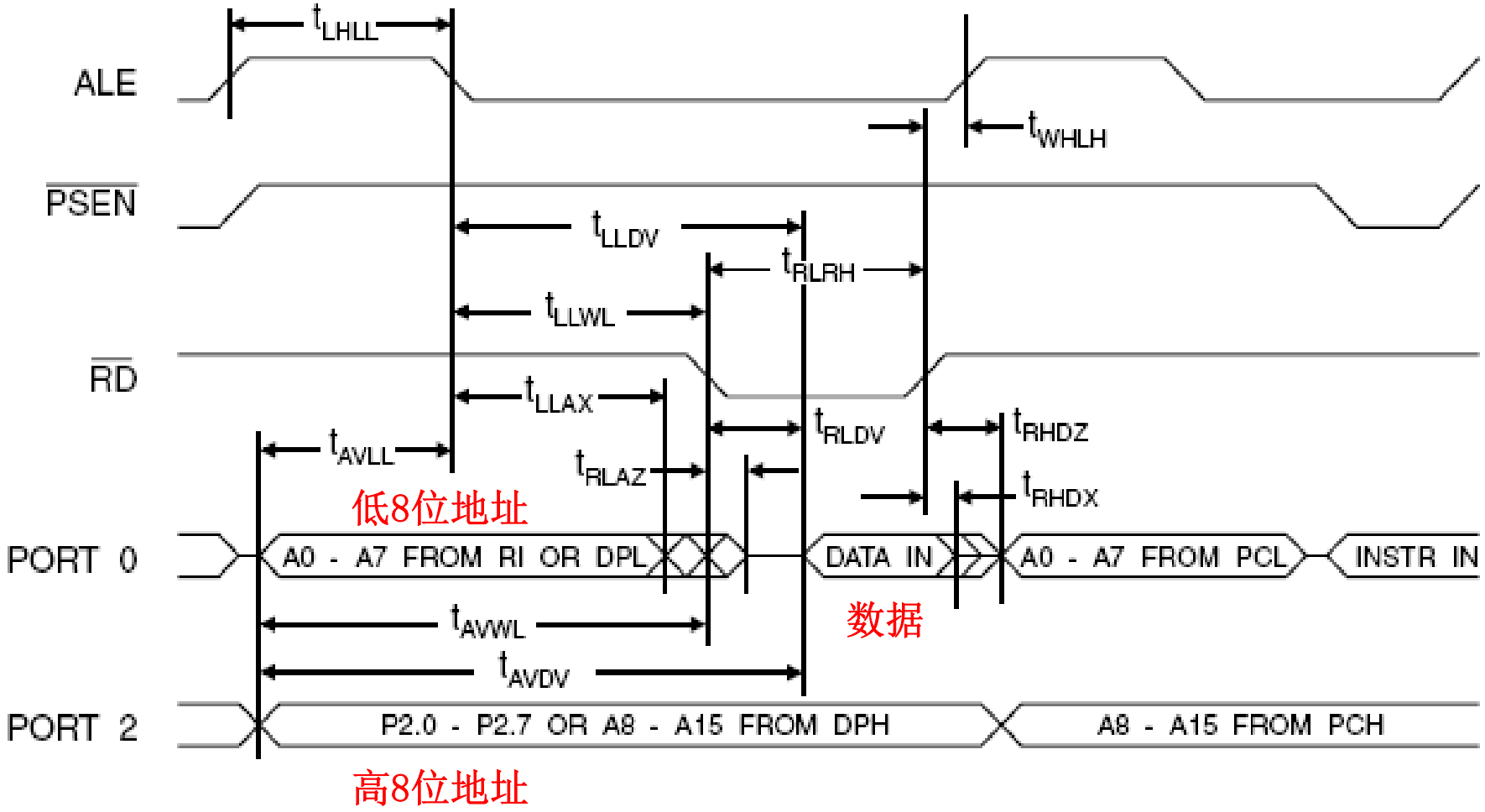
例：ADD A，DATA

外部数据存储器RAM读指令的时序

MOVX A, @DPTR

MOVX A, @Ri

External Data Memory Read Cycle

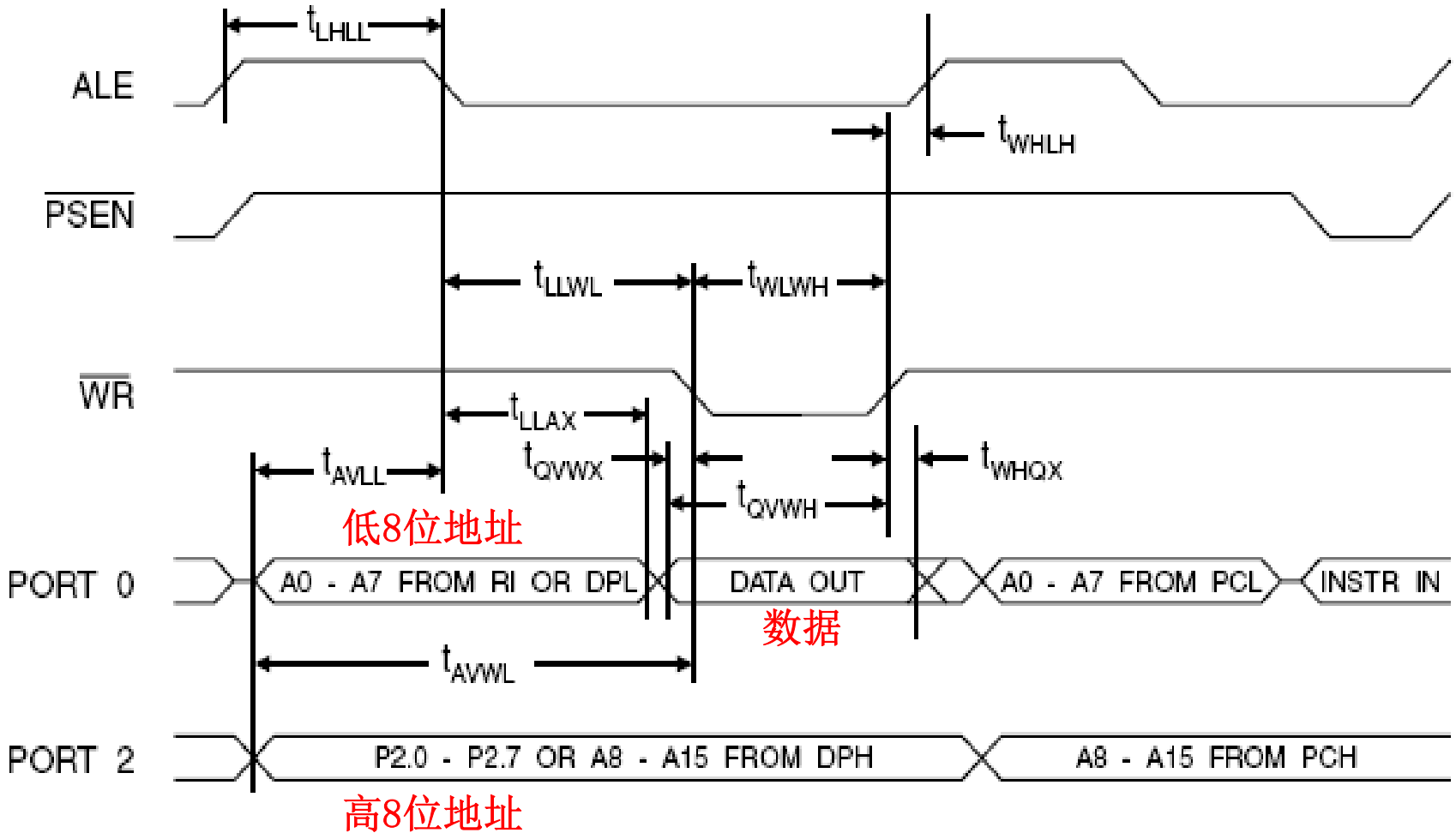


外部数据存储器RAM写指令的时序

MOVX @DPTR, A

MOVX @Ri, A

External Data Memory Write Cycle



(10) 8051 微控制器的一个机器周期由____个振荡周期组成，设系统晶振频率为6MHz，则一个机器周期的时间是_____。

A 6，2μs

B 12，2μs

C 12，1μs

D 6，1μs

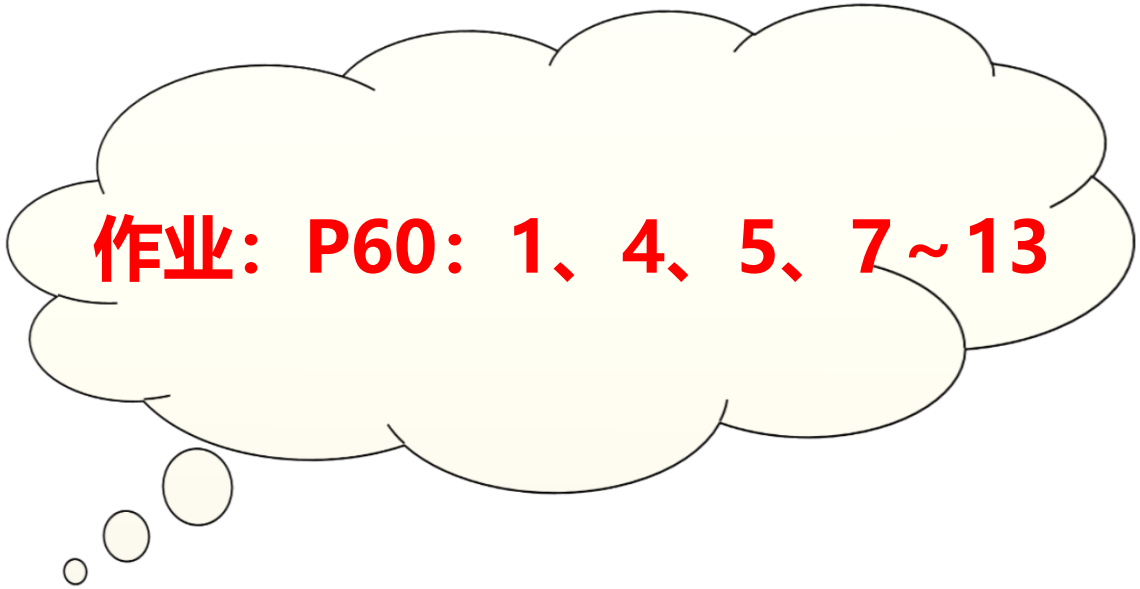
提交

3.8 作业与思考题

任务： 1.复习这章学习内容

2.完成下面的作业

3.预习第3章——MCS-51汇编指令



THE END

THANK YOU