

第2章 89S51单片机的硬件结构

第二章 89S51系列单片机的结构和原理

有关MCS-51机的性能特点，在第一章已作了简单介绍，本节将进一步详细介绍它的结构，使大家对它的性能有更深入的了解。

MCS-51的典型产品是8031(2)、8051(2)、8751(2)。8031无片内ROM，8051是ROM型单片机，内部有4KROM；8751片内有4K EPROM；89S51片内有4K Flash EPROM。除此以外，它们的内部结构及引脚完全相同。

学习要求：

- ☑ 掌握89S51单片机基本组成（内部资源）
- ☑ 掌握89S51单片机存储器的配置及特点；
- ☑ 熟练掌握21个特殊功能寄存器（SFR）的功能；
- ☑ 掌握各个引脚的功能，达到会应用的目的；
- ☑ 了解89S51单片机时序及单片机的工作过程,了解其工作方式，其中应掌握时钟周期和机器周期的计算。
- ☑ 理解单片机CPU的工作时序及相应的工作原理。

第二章 基本内容：

2.1 89S51单片机的片内结构

2.2 89S51的引脚

2.3 89S51 的CPU

2.4 89S51 存储器的结构

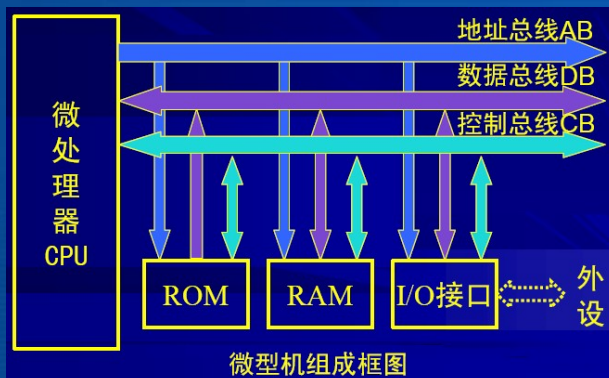
2.5 并行I/O端口

2.6 时钟电路与时序

2.7 复位操作和复位电路

2.1 89S51单片机的硬件结构

知识点回顾:

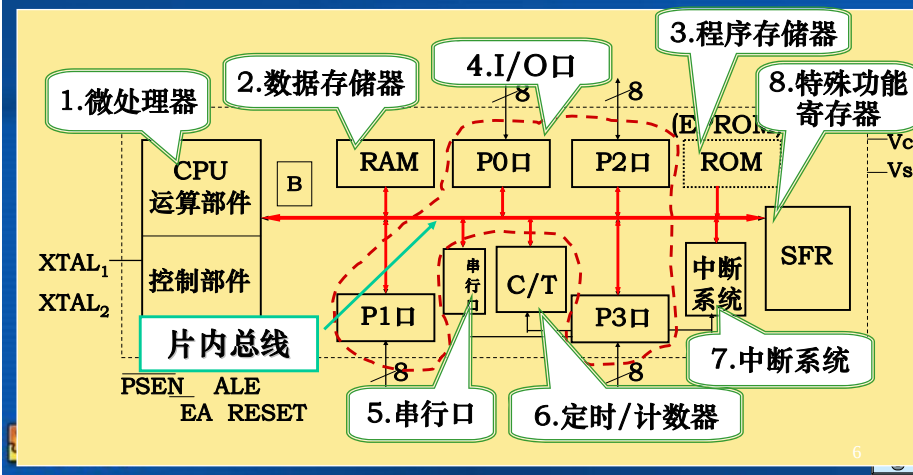


微机的硬件由CPU、存储器、输入/输出设备构成；
输入/输出设备通过输入/输出接口与系统相连；
(输入/输出接口简称I/O接口)
各部件通过总线连接。

5

2.1 89S51 单片机的硬件结构

- ◆ 按功能可分成8个部件，通过片内单一总线连接起来
- ◆ 控制方式：SFR对各功能部件集中控制



6

常用术语:

1. **位(Bit)**——能表示的最小数据单位，状态“0”和“1”。
2. **字节(Byte)**——一个8位二进制数称为一个字节。
字节是计算机处理数据的基本单位，即以字节为单位存储和解释信息。存储器中每个存储单元的大小就是一个字节。
3. **字(Word)**——一次可以处理或运算的一组二进制数，是计算中信息的基本单元。
4. **字长(Word Length)**——指字的二进制数的位数。

7



以8位字长为例：D7为符号位，D6~D0为数字位。
若字长为16位：D15为符号位，D14~D0为数字位。

8

各功能部件:

1.CPU (微处理器)

2.数据存储器 (RAM)

片内为128个字节 (52子系列的为256个字节)

3.程序存储器 (ROM/EPROM)

8031:无此部件;

8051:4K字节ROM;

8751:4K字节EPROM;

89S51/89C52/89C55:4K/8K/20K 字节闪存。

4. P1口、P2口、P3口、P0口: 为4个并行8位I/O口。

5. 串行口 1个全双工的异步串行口

(6) 2个可编程的16位定时器/计数器;

(7) 1个看门狗定时器;

(8) 中断系统有5个中断源, 对应5个中断向量;

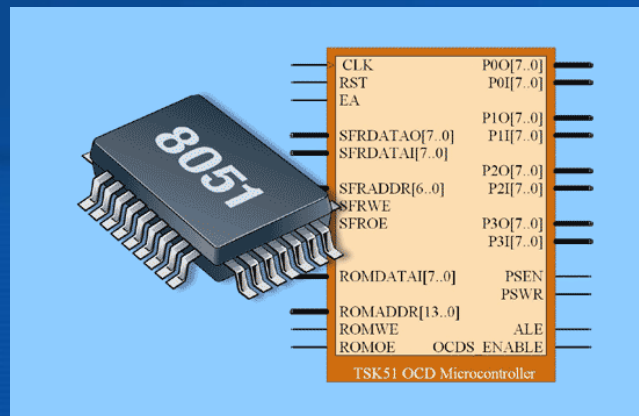
(9) 特殊功能寄存器 (SFR) 26个;

(10) 低功耗模式有**空闲模式**和**掉电模式**, 且具有掉电模式下

的中断恢复模式;

(11) 3个程序加密定位;

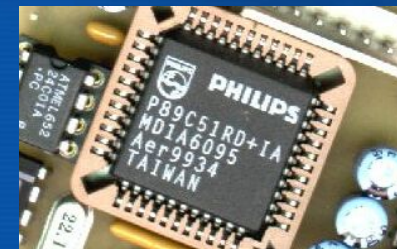
MCS-51的芯片示意图



MCS-51的芯片实物图

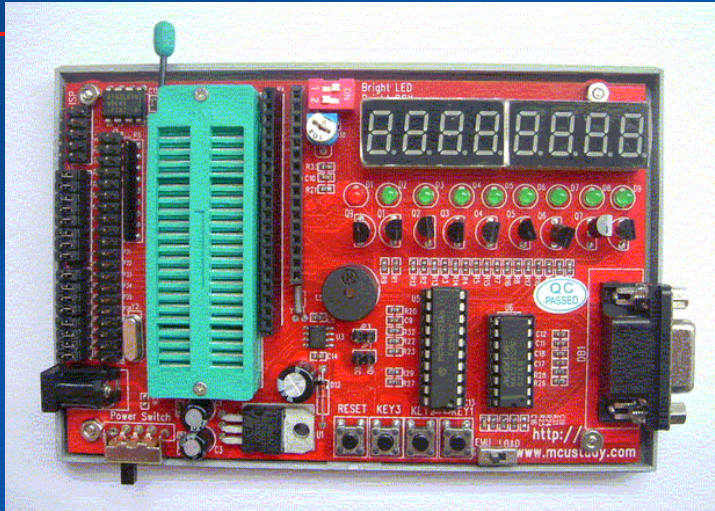


ATMEL公司的
AT89C52
8位Flash单片机
双列直插封装方式
(DIP)40引脚
6元/片



PHILIPS公司的P89C51
8位Flash单片机
—4k字节FLASH (89C51X)
—8k字节FLASH (89C52X)
方形封装方式

MCS-51的学习板



单片机学习套件配合软件可以直接对锁紧座的单片机进行烧写，编程和实验共用，烧写完毕即可运行，学习更方便，快捷。Easy51单片机学习套件提供了丰富的硬件资源和接口

13

第二章 基本内容:

2.1 89S51单片机的硬件结构

2.2 89S51的引脚

2.3 89S51的CPU

2.4 89S51存储器的结构

2.5 并行I/O端口

2.6 时钟电路与时序

2.7 复位操作和复位电路

14

2.2 89S51的引脚

40只引脚
双列直插
封装
(DIP)

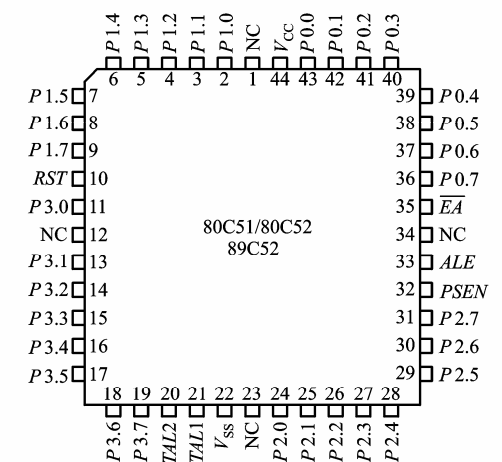


P1.0	1	40	V _{CC}
P1.1	2	39	P0.0
P1.2	3	38	P0.1
P1.3	4	37	P0.2
P1.4	5	36	P0.3
MOSI/P1.5	6	35	P0.4
MISO/P1.6	7	34	P0.5
SCK/P1.7	8	33	P0.6
RST	9	32	P0.7
RXD/P3.0	10	31	EA/V _{PP}
TXD/P3.1	11	30	ALE/PROG
INT0/P3.2	12	29	PSEN
INT1/P3.3	13	28	P2.7
T0/P3.4	14	27	P2.6
T1/P3.5	15	26	P2.5
WR/P3.6	16	25	P2.4
RD/P3.7	17	24	P2.3
XTAL2	18	23	P2.2
XTAL1	19	22	P2.1
V _{SS}	20	21	P2.0

15

2.2 89S51的引脚

44只引脚方形
封装方式 (4
只无用)

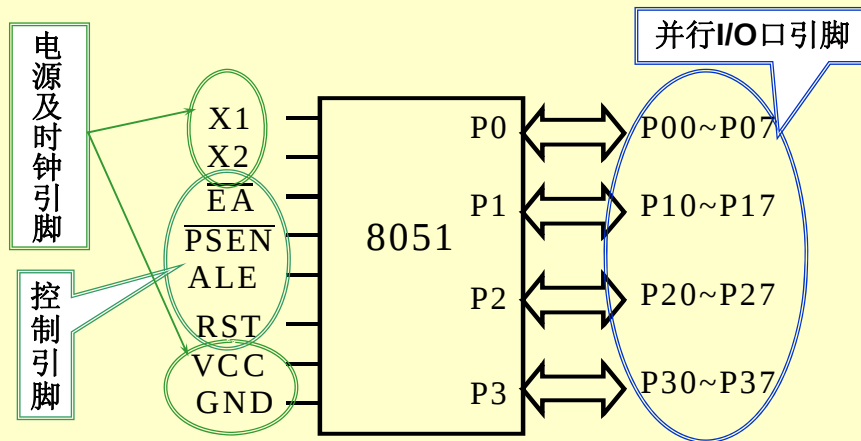


2-3

16

引脚逻辑图

- 89S51单片机为40条引脚双列直插式封装
- 引脚可分为三个部分



17

2.2.1 电源及时钟引脚

1. 电源引脚

- (1) **Vcc** (40脚) : +5V电源;
- (2) **Vss** (20脚) : 接地。

2. 时钟引脚

- (1) **XTAL1** (19脚) : 采用外接晶体振荡器时, 此引脚应接地。
- (2) **XTAL2** (18脚) : 接外部晶体的另一端。

18

2.2.2 控制引脚

- (1) **RST/VPD** (9脚) : 复位与备用电源
- (2) **ALE/PROG*** (30脚) :
第一功能**ALE** : 地址锁存允许
第二功能**PROG*** : 编程脉冲输入端。
- (3) **PSEN*** (29脚) : 读外部程序存储器的选通信号, 低有效。可以驱动8个LS型TTL负载。
- (4) **EA*/VPP** (31脚) : **EA***为内外程序存储器选择控制
EA*=1, 访问片内程序存储器,
EA*=0, 单片机则只访问外部程序存储器。
第二功能**Vpp**, 用于施加编程电压。

19

2.2.3 I/O口引脚

- (1) **P0口** : 双向8位三态I/O口, 地址总线(低8位)及数据总线分时复用口, 可驱动8个LS型TTL负载。
- (2) **P1口** : 8位准双向I/O口, 可驱动4个LS型TTL负载。
- (3) **P2口** : 8位准双向I/O口, 与地址总线(高8位)复用, 可驱动4个LS型TTL负载。
- (4) **P3口** : 8位准双向I/O口, 双功能复用口, 可驱动4个LS型TTL负载。

注意:准双向口与双向三态口的差别。

当3个准双向I/O口作输入口使用时, 要向该口先写“1”, 另外准双向I/O口无高阻的“浮空”状态。

2.2.3 I/O口引脚

表 2-1 P3 口的第二功能定义

引 脚	第 二 功 能	说 明
P3.0	RXD	串行数据输入口
P3.1	TXD	串行数据输出口
P3.2	INT0	外部中断 0 输入
P3.3	INT1	外部中断 1 输入
P3.4	T0	定时器 0 外部计数输入
P3.5	T1	定时器 1 外部计数输入
P3.6	WR	外部数据存储器写选通输出
P3.7	RD	外部数据存储器读选通输出

21

第二章 基本内容:

2.1 89S51单片机的硬件结构

2.2 89S51的引脚

2.3 89S51的CPU

2.4 89S51存储器的结构

2.5 并行I/O端口

2.6 时钟电路与时序

2.7 复位操作和复位电路



22



2.3 89S51的CPU



23



2.3 89S51的CPU

由**运算器**和**控制器**所构成

2.3.1 运算器

对操作数进行**算术**、**逻辑运算**和**位操作**。

1. 算术逻辑运算单元ALU

2. 累加器A

使用最频繁的寄存器，可写为**Acc**。A的作用：

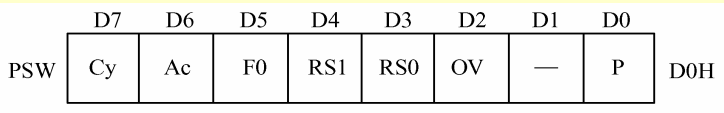
- (1) 是ALU单元的输入之一，又是运算结果存放单元。
- (2) 数据传送大多都通过累加器A。
- (3) A的进位标志**Cy**同时又是**位处理机**的**位累加器**。

24

3. 程序状态字寄存器PSW

PSW (Program Status Word) 位于片内特殊功能寄存器区，字节地址D0H。

包含了程序运行状态的信息，其中4位保存当前指令执行后的状态，供程序查询和判断。格式如下图。



PSW中各位功能:

(1) Cy (PSW. 7) 进位标志位

可写为C。在算术和逻辑运算时，若有进位/借位，Cy=1；否则，Cy=0。在位处理器中，它是位累加器。

(2) Ac (PSW. 6) 辅助进位标志位

在BCD码运算时，用作十进位调整。即当D3位向D4位产生进位或借位时，Ac=1；否则，Ac=0。

(3) F0 (PSW. 5) 用户设定标志位

由用户使用的一个状态标志位，可用指令来使它置“1”或清“0”，控制程序的流向。用户应充分利用。

(4) RS1、RS0 (PSW. 4、PSW. 3) 4组工作寄存器区选择

选择片内RAM区中的4组工作寄存器区中的某一组为当前工作寄存器区。如表2-2。

表2-2 RS1、RS0与4组工作寄存器区的对应关系

RS1	RS0	所选的4组寄存器
0	0	0区（内部RAM地址00H~07H）
0	1	1区（内部RAM地址08H~0FH）
1	0	2区（内部RAM地址10H~17H）
1	1	3区（内部RAM地址18H~1FH）

(5) OV (PSW. 2) 溢出标志位

当执行算术指令时，用来指示运算结果是否产生溢出。如果结果产生溢出，OV=1；否则，OV=0。

(6) PSW. 1位：保留位。

(7) P (PSW. 0) 奇偶标志位

指令执行后，累加器A中“1”的个数是奇数还是偶数。

P=1，A中“1”的个数为奇数。

P=0，A中“1”的个数为偶数。

此标志位对串行通信有重要意义，常用奇偶检验来检验数据串行传输的可靠性。

例

- 程序执行前, $F0=0$, $RS1RS0=00$, 执行

$\left\{ \begin{array}{l} \text{MOV A, \#0FH} \\ \text{ADD A, \#0F8H} \end{array} \right.$ 后, PSW中各位状态如何?

解:

0	0	0	0	1	1	1	1	(0FH)
+	1	1	1	1	0	0	0	(0F8H)
1	0	0	0	0	0	1	1	(A)

$1 = (Cy) \quad (Cs) = 1 \quad (Ac) = 1 \quad (A) = 07H$
 $(Ov) = (Cy) \oplus (Cs) = 0 \quad (P) = 1$
 $(PSW) = 11000001B = C1H$

2.3.2 控制器

发布操作命令的机构, 是计算机的指挥中心, 控制计算机的各部分协调工作, 用以自动执行程序。

2、控制器

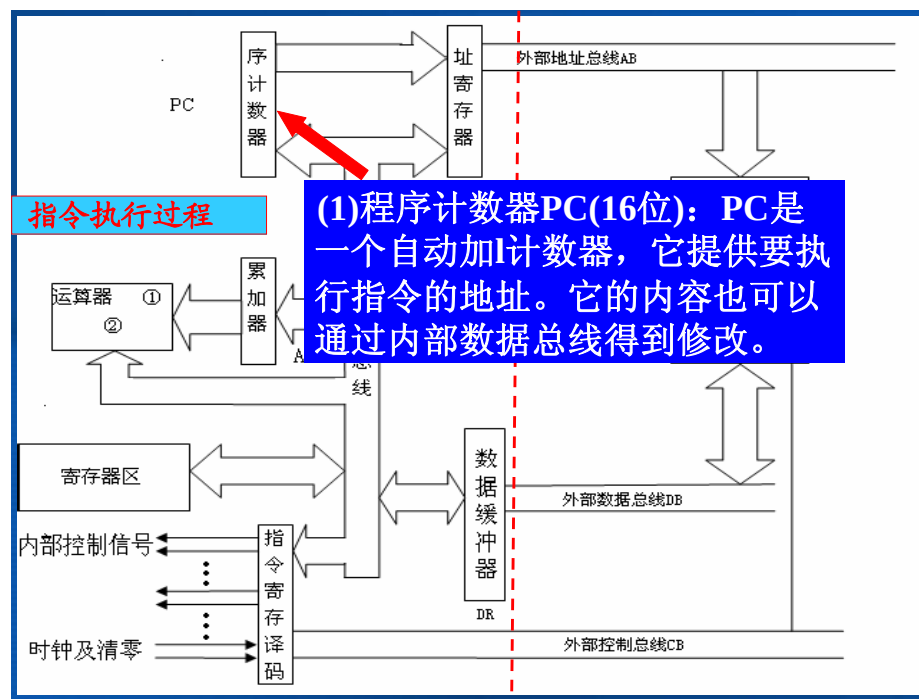
定时控制逻辑
 指令寄存器, 译码器
DPTR
程序计数器PC
堆栈指针SP
 RAM地址寄存器
 16位地址缓冲器等

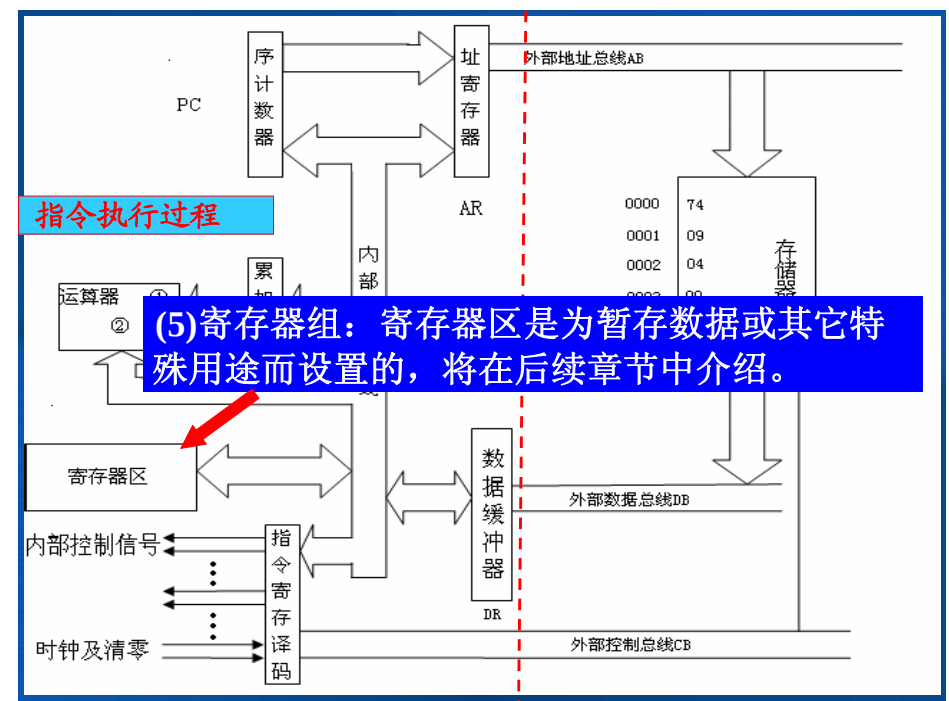
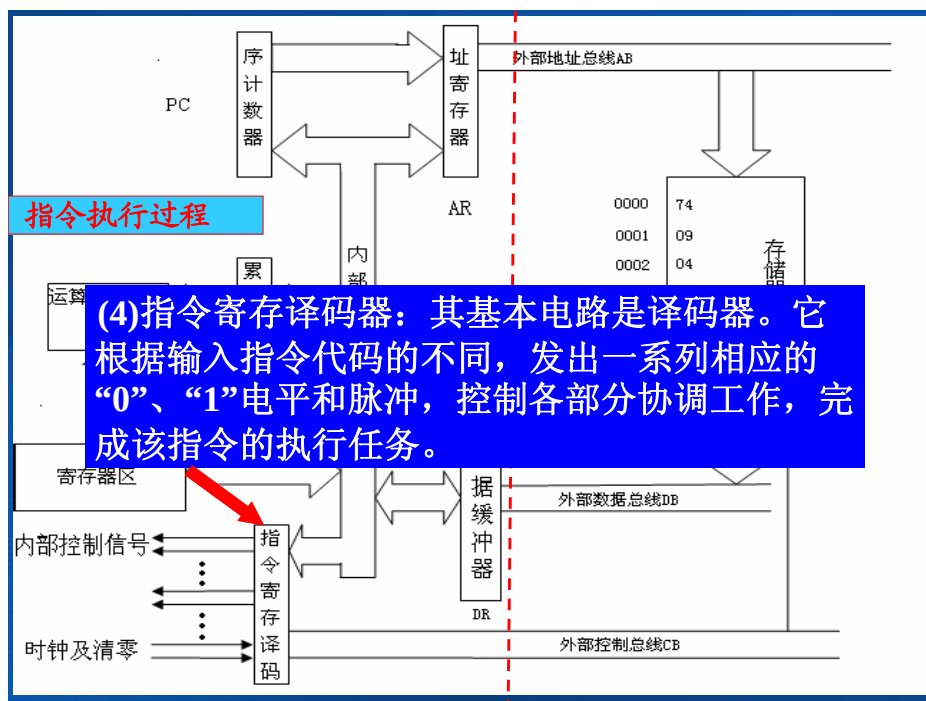
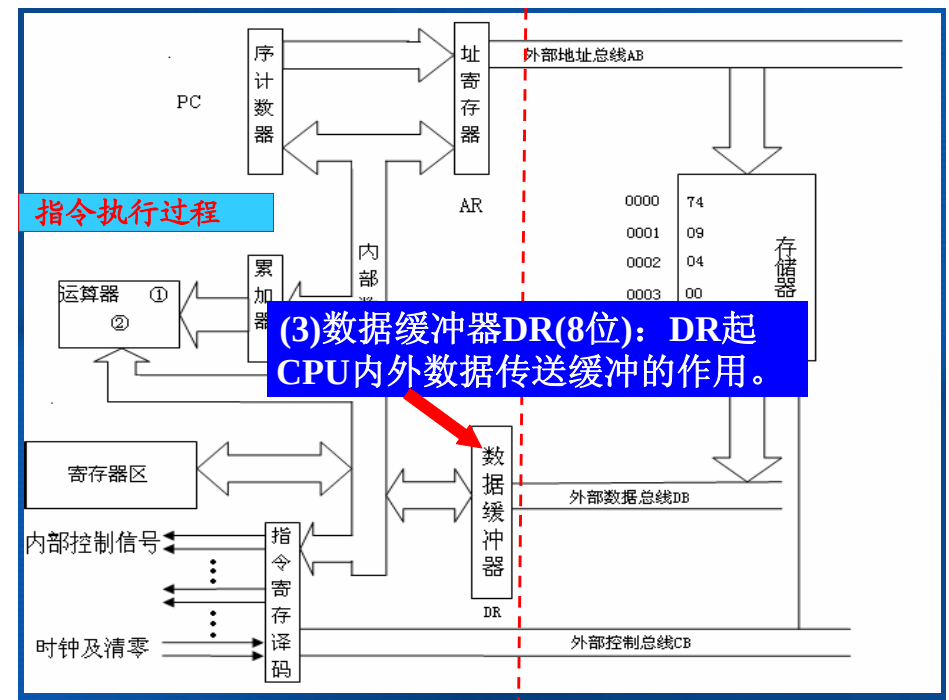
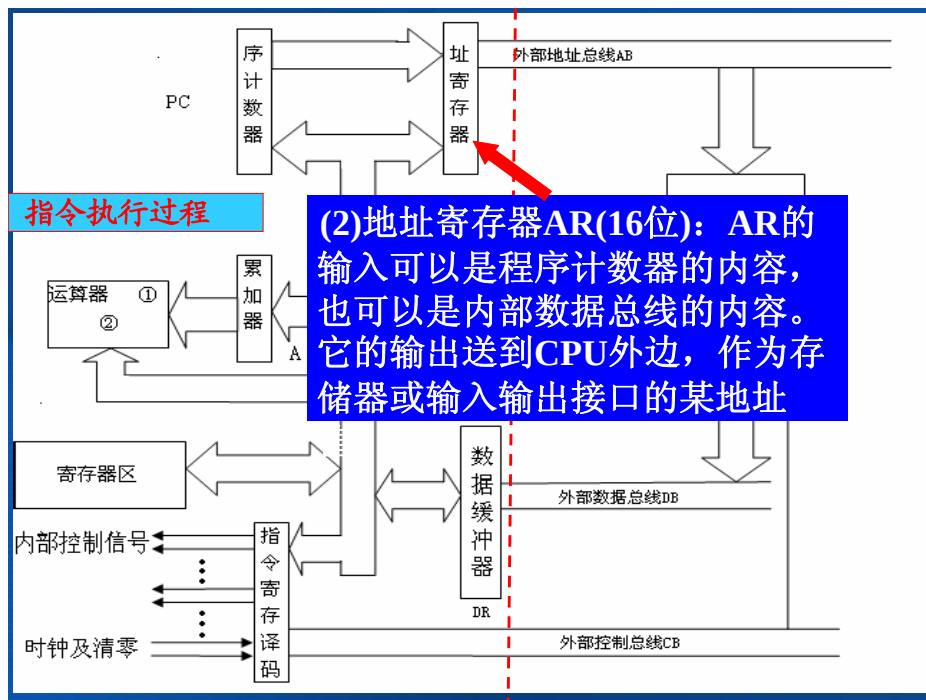
30

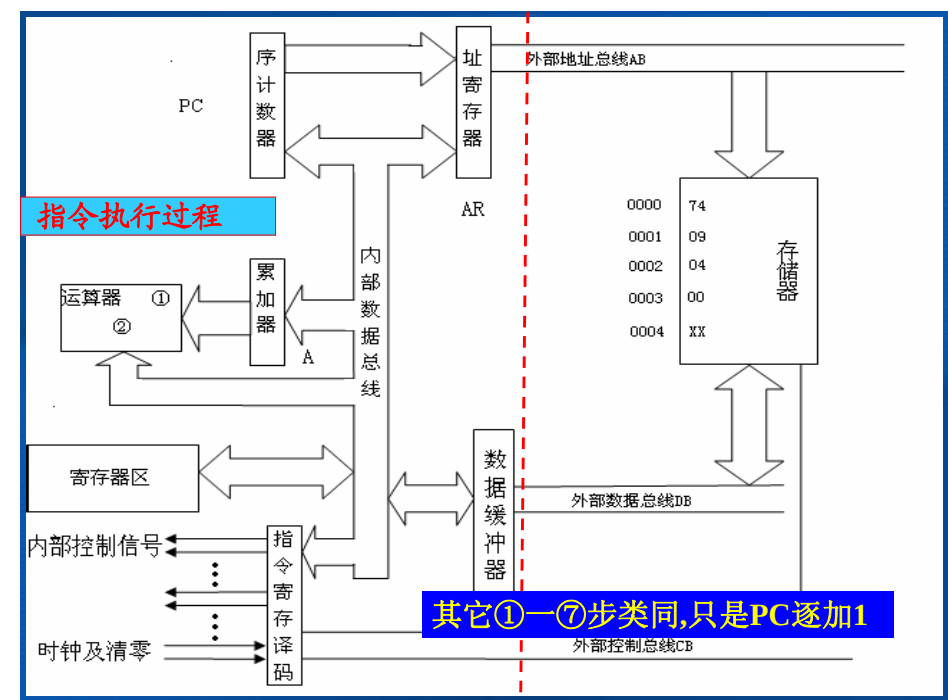
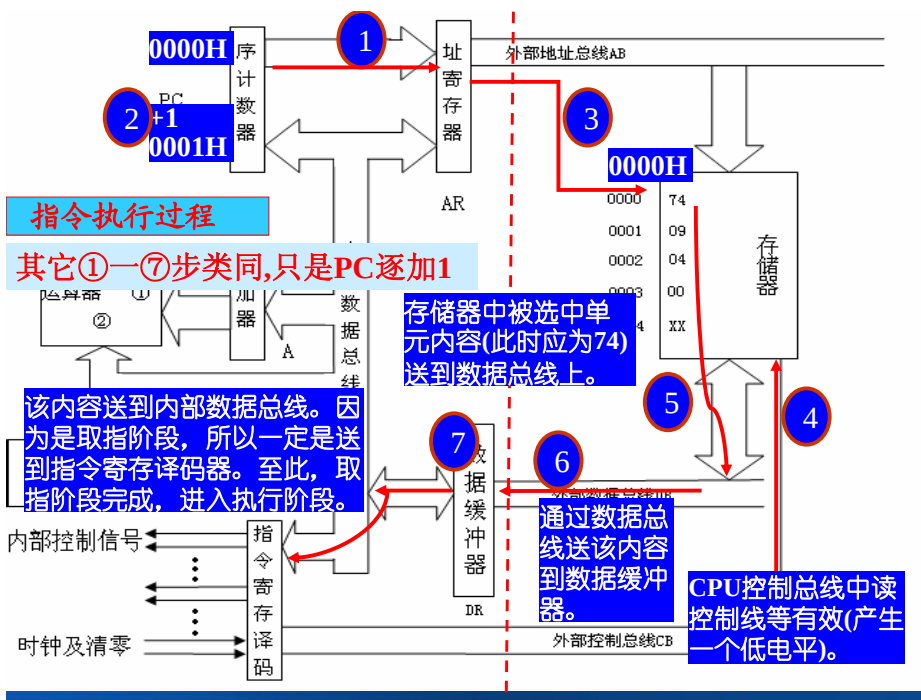
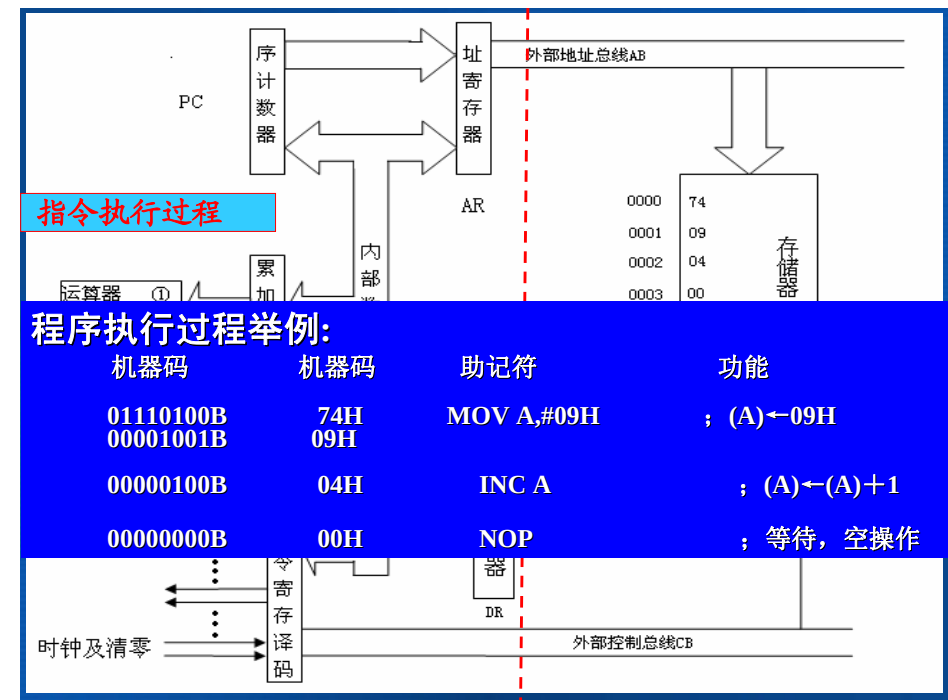
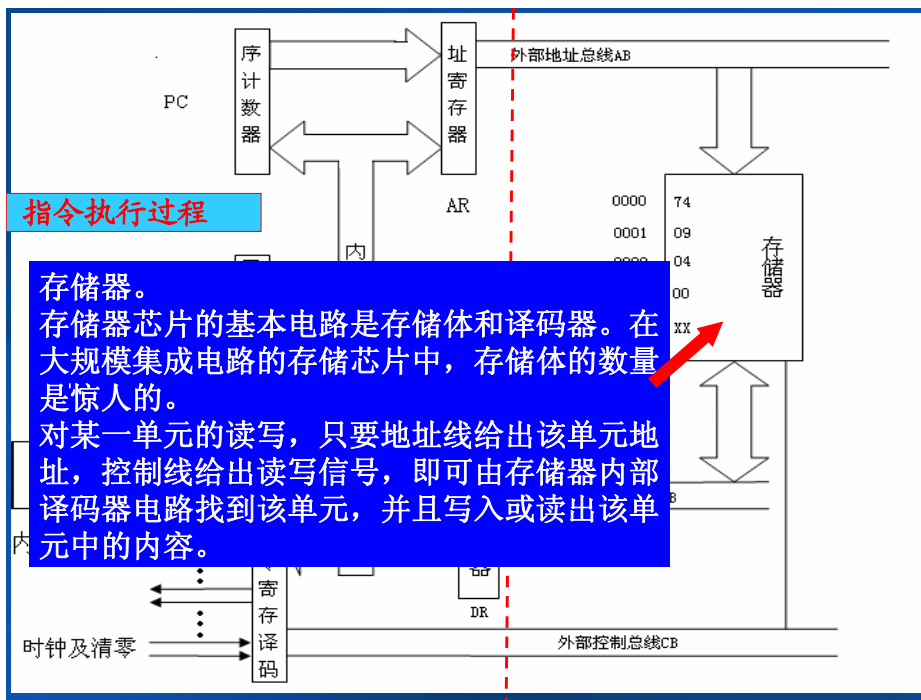
2.3.2 控制器

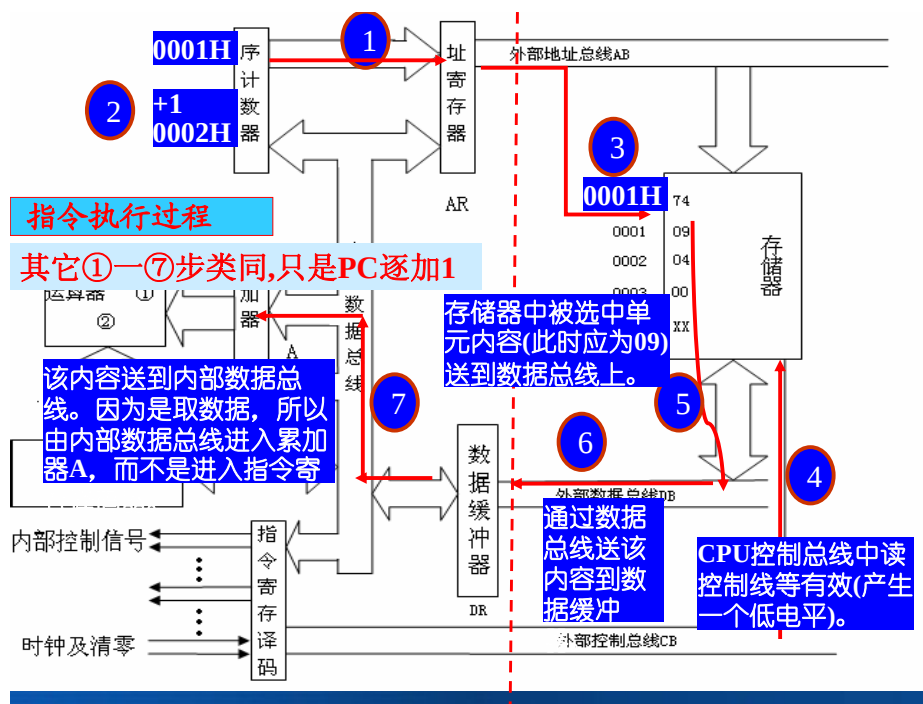
扩展知识: 指令的执行过程

31









2.3.2 控制器

1. 程序计数器 (PC) 16位计数器 (重要)

PC是程序的字节地址计数器, 其内容是将要执行的下一条指令的地址, 寻址范围达64KB。

PC 有自动加1功能, 从而实现程序的顺序执行。可以通过转移、调用、返回等指令改变其内容, 以实现程序的转移。

42

2.3.2 控制器

1. 程序计数器 (PC) 16位计数器 (重要)

PC是程序的字节地址计数器, 其内容是将要执行的下一条指令的地址, 寻址范围达64KB。

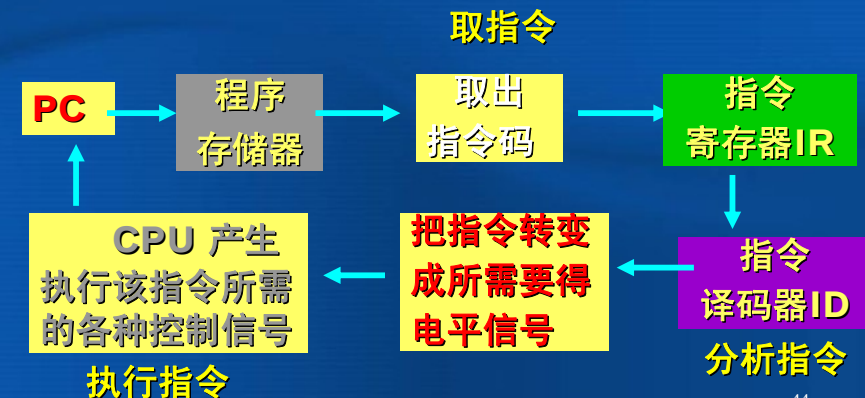
基本工作方式:

- (1) 程序计数器自动加1
- (2) 执行有条件或无条件转移指令时, 程序计数器将被置入新的数值, 从而使程序的流向发生变化。
- (3) 执行子程序调用或中断调用时完成下列操作:
 - ① PC的当前值保护
 - ② 将子程序入口地址或中断向量的地址送入PC

43

2. 指令译码器ID

当指令取出经指令寄存器IR送至指令译码器ID时, ID对该指令进行译码, 即把指令转变成所需的电平信号, CPU 根据ID输出的电平信号使定时控制电路定时地产生执行该指令所需的各种控制信号, 以使计算机能正确执行程序所要求的各种操作。



44

第二章 基本内容:

2.1 89S51单片机的片内结构

2.2 89S51的引脚

2.3 89S51 的CPU

2.4 89S51 存储器的结构

2.5 并行I/O端口

2.6 时钟电路与时序

2.7 复位操作和复位电路



45

1、存储器（类型、结构、重要指标）

类型：**ROM** **Read Only Memory** 正常工作时只能读不能写的存储器

不易失 PROM 可编程ROM, 厂家一次写入
EPROM 用户可编程可擦写ROM, 紫外线擦除器
EEPROM 电可擦写可编程ROM, 在线, 读快/写慢
Flash PEROM 闪速可编程可擦写ROM

RAM **Random Access Memory** 正常工作时即可读又可写的存储器

DRAM 动态RAM, 集成度高, 外加刷新电路
SRAM 静态RAM, 成本高、速度快
iRAM 全集成化RAM, DRAM+刷新电路
NVRAM SRAM+EEPROM, 不挥发即不易失 **易失**

ROM和RAM的主要区别:

1、断电后ROM内的信息不丢失

RAM 中的信息立即丢失

2、读/写方式不同

ROM采用特殊方式写入信息,

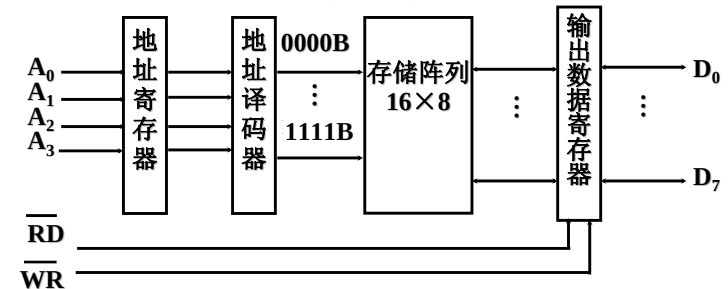
正常工作是只读方式

RAM正常工作既能读又能写

47

结构

图1-2 16×8bitRAM的内部结构框图



ROM和RAM芯片均有分四组引脚线: (ROM另有特殊的引脚线)

1、地址线 传送存储器的地址码, 其根数决定存储单元个数即字数

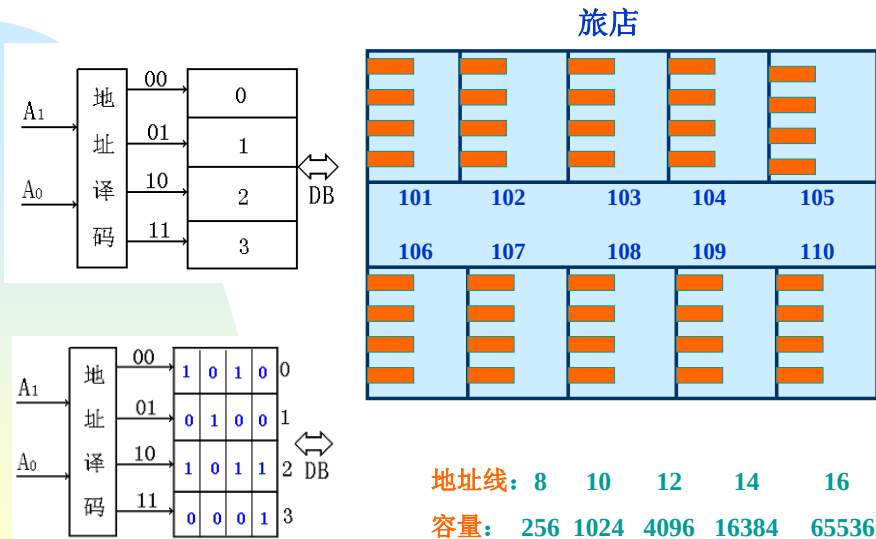
2、数据线 传送对某一单元进行读/写的数, 双向
决定一个单元内存储二进制数的位数, 即字长

3、控制线 传送读/写控制信号, 以控制读/写操作

4、电源线 +5V 和GND线

48

存储单元、单元位数、地址



49

存储器的两个重要指标

存储容量

$$\text{存储容量} = 2^{\text{地址线条数}} \times \text{数据线的条数bit}$$

$$= \text{字数 (存储单元个数)} \times \text{字长}$$

例: 芯片2732 4KB 即 $4\text{K} \times 8\text{bit} = 32\text{Kb}$

地址线 12根, 数据线 8根

芯片 2114 $1\text{K} \times 4\text{bit}$ 地址线 10根, 数据线 4根

问? 2764 8KB 地址线 ? 根, 数据线 ? 根

存取周期

存储器从接到存储单元地址开始,
到读出或写入数据为止所用的时间。



50

2.4 89S51的存储器

89S51单片机与一般微机的存储器配置方式很不相同。

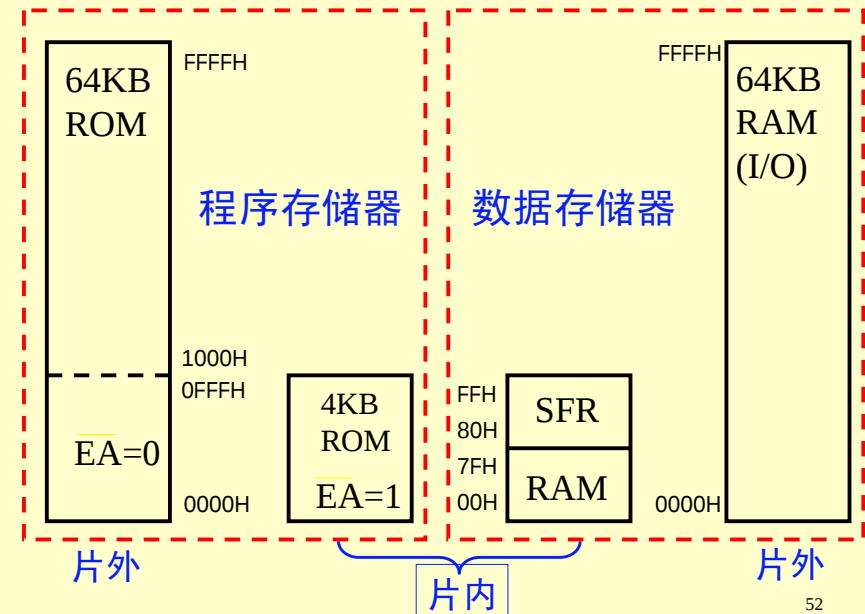
一般微机通常只有一个逻辑空间, 可以随意安排ROM或RAM。访问存储器时, 同一地址对应唯一的存储单元, 可以是ROM也可以是 RAM, 并用同类访问指令。

而89S51则不同:

89S51在物理结构上设计成程序存储器与数据存储器独立分开的哈佛结构:

- 片外数据存储器、片外程序存储器、
- 片内程序存储器4KB (ROM 0000H~0FFFH)
- 片内数据存储器128B (RAM 00H~7FH)

51



52

89S51在**逻辑**上，即从用户角度上89S51有三个存储空间：

片内外统一编址的程序存储器

片内外不统一编址的数据存储器

特殊功能寄存器（片内）

53

一、程序存储器及地址空间

作用——程序存储器用于存放编好的程序和表格常数。

①89S51片内有4K字节ROM，片外用16位地址线最多可扩展64K字节ROM，**两者是统一编址的**。

★如果**EA端保持高电平**，89S51执行片内前4KB ROM地址(0000H~0FFFH)中的程序。当寻址范围超过4KB(1000H~FFFFH)时，则从片外存储器取指令。

★当**EA端保持低电平**时，89S51的所有取指令操作均在片外程序存储器中进行，这时片外存储器可以从0000H开始编址。

54

②在程序存储器中，有6个单元具有特殊功能
0000H: 89S51复位后，PC=0000H，即程序从0000H

开始执行指令。

0003H: 外部中断0入口。

000BH: 定时器0溢出中断入口。

0013H: 外部中断1入口。

001BH: 定时器1溢出中断入口。

0023H: 串行口中断入口。

使用时，通常在这些入口地址处存放一条绝对跳转指令，使程序跳转到用户安排的中断程序起始地址，或者从0000H起始地址跳转到用户设计的初始程序上。

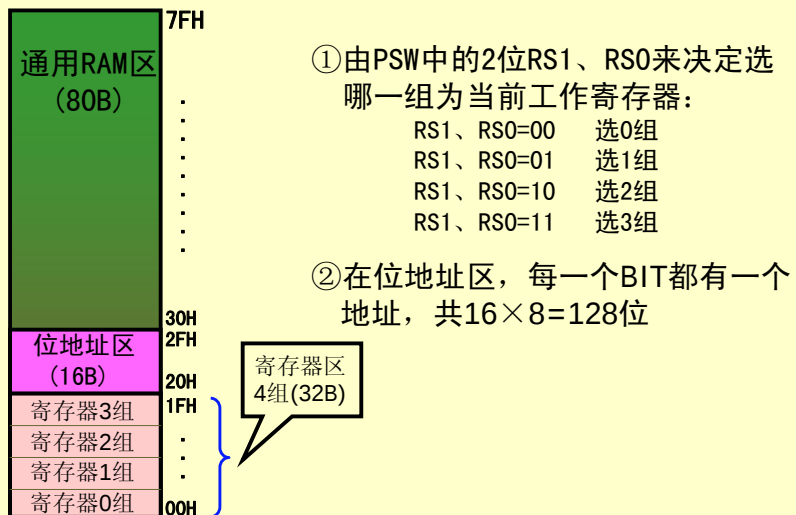
55

二、数据存储器及地址空间

数据存储器 { **片外RAM64KB**，地址范围0000H~FFFFH
片内RAM128B，地址范围00H~7FH

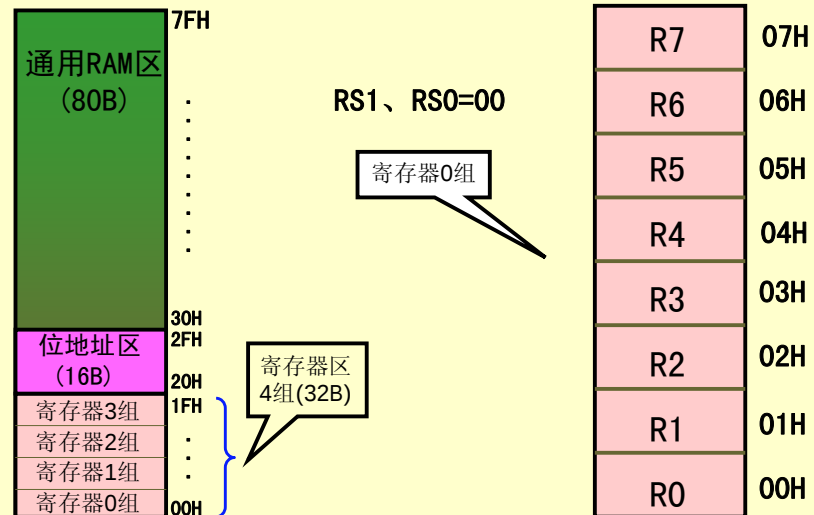
56

片内数据存储器空间分布图



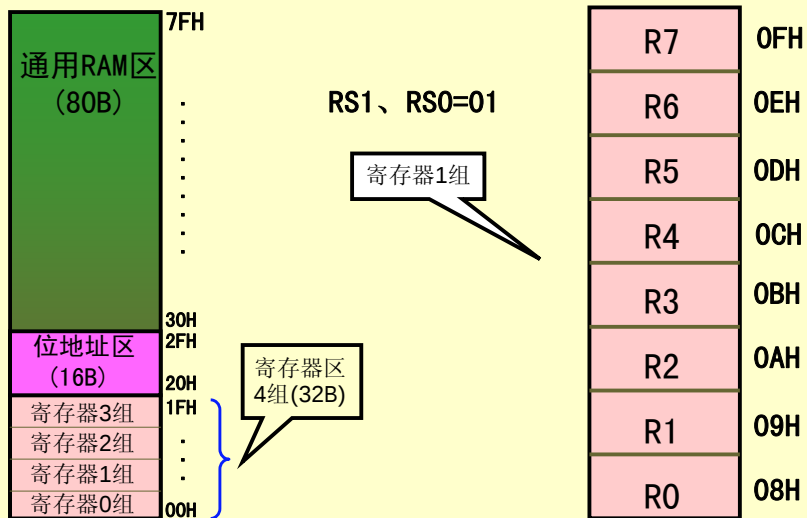
57

片内数据存储器空间分布图



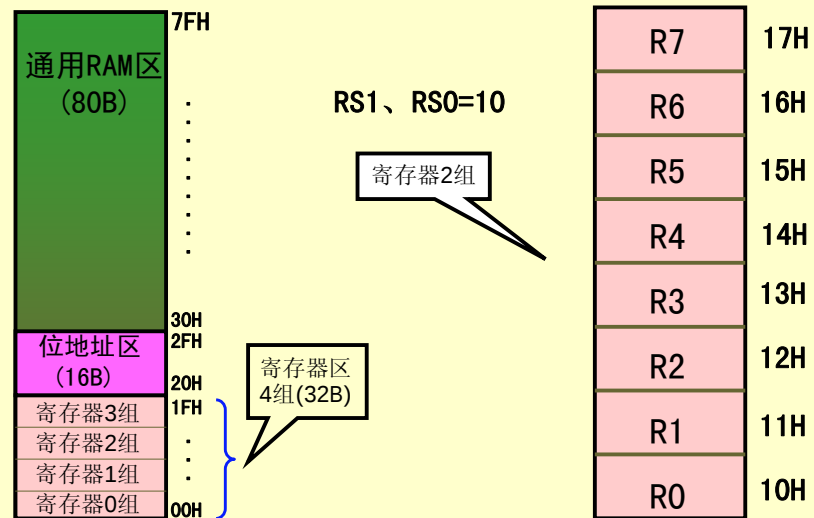
58

片内数据存储器空间分布图



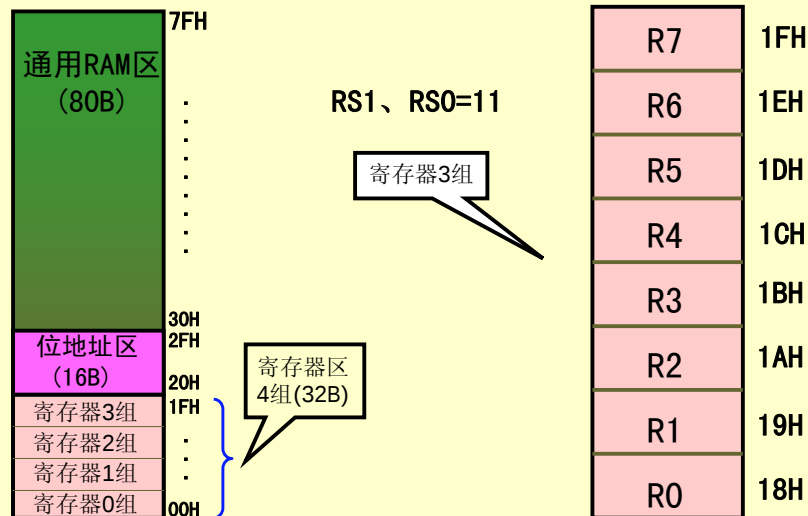
59

片内数据存储器空间分布图



60

片内数据存储器空间分布图



61

什么是工作寄存器呢？

让我们从现实生活中来找找答案。

如果出一道数学题： $123+567$,

再看下面一道题： $123+567+562$ ，要让你要上回答，就不这么容易了吧？

为了要得到最终结果，往往要做很多步的中间结果。这些中间结果要有个地方放才行，把它们放哪呢？放在前面提到过的ROM中可以吗？

显然不行，因为计算机要将结果写进去，而ROM是不可以写的，所以在单片机中另有一个区域称为RAM区（RAM是随机存取存储器的英文缩写），它可以将数据写进去。

特别地，在89S51单片机中，将RAM中分出一块区域，称为工作寄存器区。

62

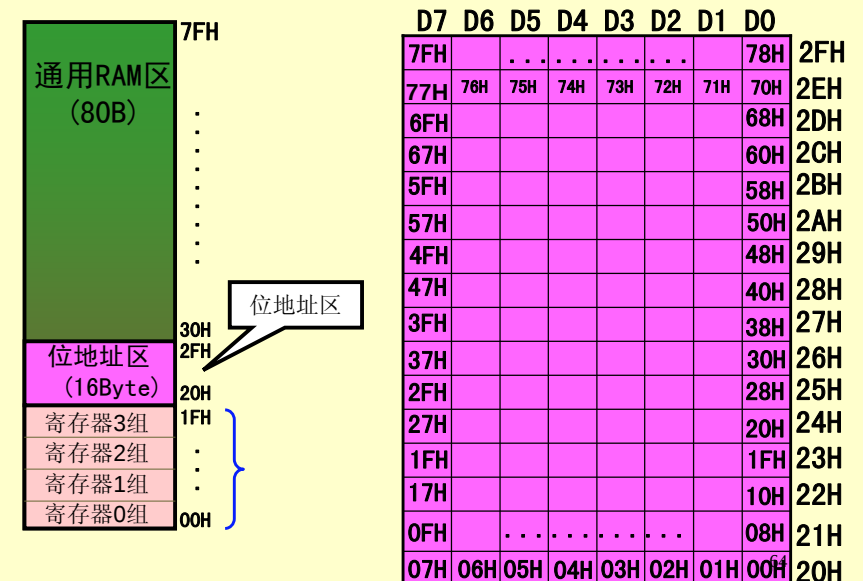
在某一时刻，只能选用一个寄存器组使用。其选择是通过软件对程序状态字（PSW）中的RS0、RS1二位的设置来实现的。设置RS0、RS1时，可以对PSW字节寻址，也可以位寻址方式，间接或直接修改RS0、RS1的内容。通常采用后者较方便。

表 2-1 RS1、RS0 与片内工作寄存器组的对应关系

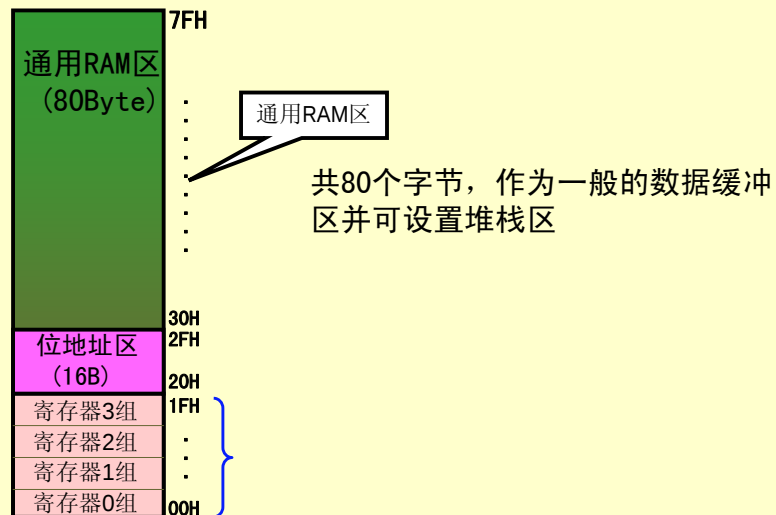
RS1	RS0	寄存器区	片内 RAM 地址	通用寄存器名称
0	0	0	00H~07H	R0~R7
0	1	1	08H~0FH	R0~R7
1	0	2	10H~17H	R0~R7
1	1	3	18H~1FH	R0~R7

63

片内数据存储器空间分布图



片内数据存储器空间分布图



65

2.4.3 特殊功能寄存器 (SFR)

CPU对各种功能部件的控制采用特殊功能寄存器集中控制方式，共26个。

与AT89C51相比，新增5个SFR：DP1L、DP1H、AUXR、AUXR1和WDTRST。

特殊功能寄存器映射在片内RAM 80H~FFH区域中。具体如下表所示。

66

表 2-4 SFR 的名称及其分布

序号	特殊功能寄存器符号	名称	字节地址	位地址	复位值
1	P0	P0口	80H	87H~80H	FFH
2	SP	堆栈指针	81H	—	07H
3	DP0L	数据指针DPTR0低字节	82H	—	00H
4	DP0H	数据指针DPTR0高字节	83H	—	00H
5	DP1L	数据指针DPTR1低字节	84H	—	00H
6	DP1H	数据指针DPTR1高字节	85H	—	00H
7	PCON	电源控制寄存器	87H	—	0××× 0000B
8	TCON	定时器/计数器控制寄存器	88H	8FH~88H	00H
9	TMOD	定时器/计数器方式控制	89H	—	00H
10	TL0	定时器/计数器0 (低字节)	8AH	—	00H
11	TL1	定时器/计数器1 (低字节)	8BH	—	00H
12	TH0	定时器/计数器0 (高字节)	8CH	—	00H

67

13	TH1	定时器/计数器1 (高字节)	8DH	—	00H
14	AUXR	辅助寄存器	8EH	—	×××0 0××0B
15	P1	P1口寄存器	90H	97H~90H	FFH
16	SCON	串行控制寄存器	98H	9FH~98H	00H
17	SBUF	串行发送数据缓冲器	99H	—	×××× ××××B
18	P2	P2口寄存器	A0H	A7H~A0H	FFH
19	AUXR1	辅助寄存器	A2H	—	×××× ×××0B
20	WDTRST	看门狗复位寄存器	A6H	—	×××× ××××B
21	IE	中断允许控制寄存器	A8H	AFH~A8H	0××0 0000B
22	P3	P3口寄存器	B0H	B7H~B0H	FFH
23	IP	中断优先级控制寄存器	B8H	BFH~B8H	××00 0000B
24	PSW	程序状态字寄存器	D0H	D7H~D0H	00H
25	A (或Acc)	累加器	E0H	E7H~E0H	00H
26	B	B寄存器	F0H	F7H~F0H	00H

68

特殊功能寄存器

SFR (Special Functional Register) 见书P21表2-4。下面简单介绍某些SFR，余下的SFR与片内外围部件密切相关，将在后续介绍片内外围部件时进行说明。

1. 堆栈指针SP

指示堆栈顶部在内部RAM块中的位置。

堆栈结构—**向上生长型**。单片机**复位**后，SP为**07H**，使得堆栈实际上从08H单元开始，由于08H~1FH单元分别属于1~3组的工作寄存器区，**最好在复位后把SP值改为60H或更大值**，避免堆栈与工作寄存器冲突。

堆栈主要是为**子程序调用和中断操作**而设。用于**保护断点和现场**。

(1) **保护断点**。无论子程序调用还是中断服务子程序调用，最终都要返回主程序。应预先先把主程序的断点在堆栈中保护起来，为程序正确返回做准备。

(2) **现场保护**。执行子程序或中断服务子程序时，要用到一些寄存器单元，会破坏原有内容。要把有关寄存器单元的内容送入堆栈，保存起来，即“现场保护”。

两种操作：**数据压入 (PUSH)** 堆栈，**数据弹出 (POP)** 堆栈。数据压入堆栈，SP自动加1；数据弹出堆栈，SP自动减1。

2. 寄存器B

为执行乘法和除法而设。在不执行乘、除法操作的情况下，可把它当作一个普通寄存器来使用。

乘法：两乘数分别在A、B中，执行乘法指令后，乘积在BA中

除法：被除数取自A，除数取自B，商存放在A中，余数存B中。

3. AUXR寄存器

AUXR是辅助寄存器，其格式如下图所示：

	D7	D6	D5	D4	D3	D2	D1	D0	
AUXR	-	-	-	WDIDLE	DISRTO	-	-	DISALE	8EH

其中：

DISALE：ALE的禁止/允许位。

0：ALE有效，发出脉冲；

1：ALE仅在执行MOVC和MOVX类指令时有效，不访问外部存储器时，ALE不输出脉冲信号；

DISRTO：禁止/允许WDT溢出时的复位输出。

0：WDT溢出时，在RST引脚输出一个高电平复位脉冲；

1：禁止WDT，RST引脚仅为输入脚。

WDIDLE：WDT在空闲模式下的禁止/允许位。

0：WDT在空闲模式下继续计数；

1：WDT在空闲模式下暂停计数。

4. 数据指针DPTR0和DPTR1

双数据指针寄存器，**便于访问数据存储器**。

DPTR0：AT89C51单片机**原有**的数据指针，

DPTR1：**新增加**的数据指针。

AUXR1的DPS位用于**选择两个数据指针**。

DPS=0时，选用DPTR0；

DPS=1时，选用DPTR1。

AT89S51复位时，默认选用DPTR0。

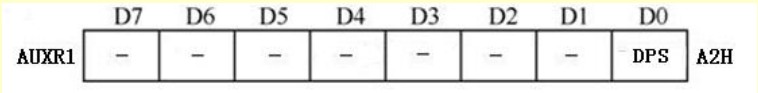
数据指针可作为一个16位寄存器来用，也可作为两个独立的8位寄存器DPOH（或DP1H）和DPOL（或DP1L）来用。

5. AUXR1寄存器

AUXR1是辅助寄存器，格式如图所示：

DPS：数据指针寄存器选择位。

- 0：选择数据指针寄存器DPTR0；
- 1：选择数据指针寄存器DPTR1。



6. 看门狗定时器WDT

包含1个14位计数器和看门狗定时器复位寄存器（WDTRST）。当CPU由于干扰，程序陷入死循环或跑飞状态时，WDT提供了一种使程序恢复正常运行的有效手段。

有关WDT的应用以及低功耗模式下运行的状态，将在本章2.8节介绍。

上面介绍的特殊功能寄存器，除SP和B以外，其余的均为AT89S51在AT89C51基础上新增加的SFR。

2.4.4 位地址空间

211个（128个+83个）寻址位。位地址范围为：00H~FFH。

内部RAM的可寻址位128个(字节地址20H~2FH)见表2-3（P24）。

特殊功能寄存器SFR为83个可寻址位，见表2-4（P24）。

表2-3 内部的可寻址位及位地址

字 节 地 址	位 地 址							
	D7	D6	D5	D4	D3	D2	D1	D0
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

表2-4 SFR中的位地址分布

特殊功能寄存器符号	位地址								字节地址
	D7	D6	D5	D4	D3	D2	D1	D0	
B	F7H	F6H	F5H	F4H	F3H	F2H	F1H	F0H	F0H
ACC	E7H	E6H	E5H	E4H	E3H	E2H	E1H	E0H	E0H
PSW	D7H	D6H	D5H	D4H	D3H	D2H	D1H	D0H	D0H
IP	—	—	—	BCH	BBH	BAH	B9H	B8H	B8H
P3	B7H	B6H	B5H	B4H	B3H	B2H	B1H	B0H	B0H
IE	AFH	—	—	ACH	ABH	AAH	A9H	A8H	A8H
P2	A7H	A6H	A5H	A4H	A3H	A2H	A1H	A0H	A0H
SCON	9FH	9EH	9DH	9CH	9BH	9AH	99H	98H	98H
P1	97H	96H	95H	94H	93H	92H	91H	90H	90H
TCN	8FH	8EH	8DH	8CH	8BH	8AH	89H	88H	88H
P0	87H	86H	85H	84H	83H	82H	81H	80H	80H

可位寻址区 (20H—2FH)

这16个单元（共计128位）的每一位都有一个8位表示的位地址，位寻址范围为00H~7FH，如表2-3所示P24。

位寻址区的每一个单元既可作为一般RAM单元使用，进行字节操作，也可以对单元中的每一位进行位操作。

89S51布尔处理器的存储空间就是指这个位寻址空间。

D7 D6 D5 D4 D3 D2 D1 D0

--	--	--	--	--	--	--	--

47H 46H 45H 44H 43H 42H 41H 40H

位地址

28H 字节地址

78

28H 字节地址

D7 D6 D5 D4 D3 D2 D1 D0

0	1	0	1	1	0	1	0
---	---	---	---	---	---	---	---

47H 46H 45H 44H 43H 42H 41H 40H

字节操作

MOV 28H, #56H

SETB 43H

CLR 42H

位操作

79

2.4.5 外部数据存储器

最多可外扩64K字节的RAM或I/O。几点注意：

(1) 地址的重叠性

程序存储器与数据存储器全部64K字节地址空间重叠

(2) 程序存储器与数据存储器在使用上是严格区分的

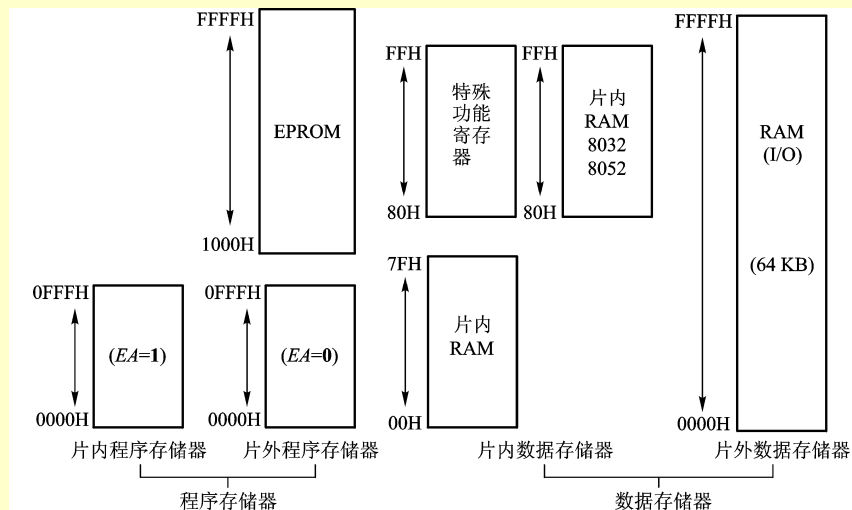
(3) 位地址空间共有两个区域

(4) 片外与片内数据存储器由指令来区分

(5) 片外数据存储区中，RAM与I/O端口统一编址。

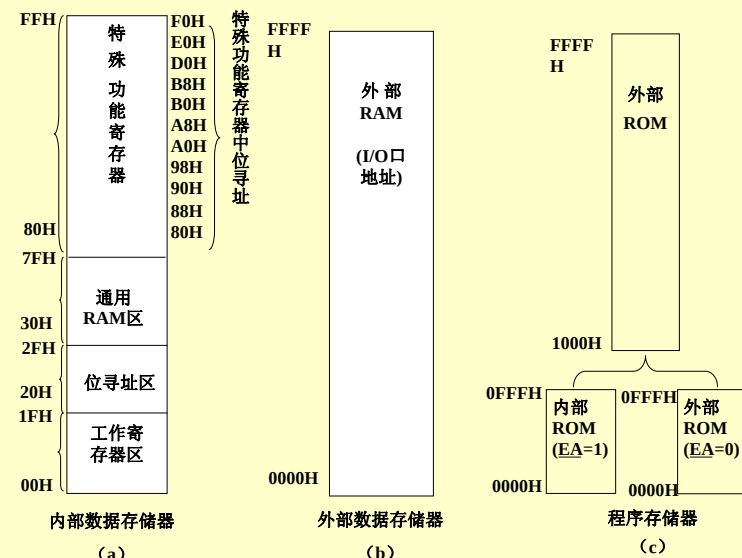
所有外围I/O端口的地址均占用RAM地址单元，使用与访问外部数据存储器相同的传送指令。

80



89S51单片机存储器空间分配

81



89S51单片机存储器空间分配

82

[返回](#)

第二章 基本内容:

2.1 89S51单片机的硬件结构

2.2 89S51的引脚

2.3 89S51的CPU

2.4 89S51存储器的结构

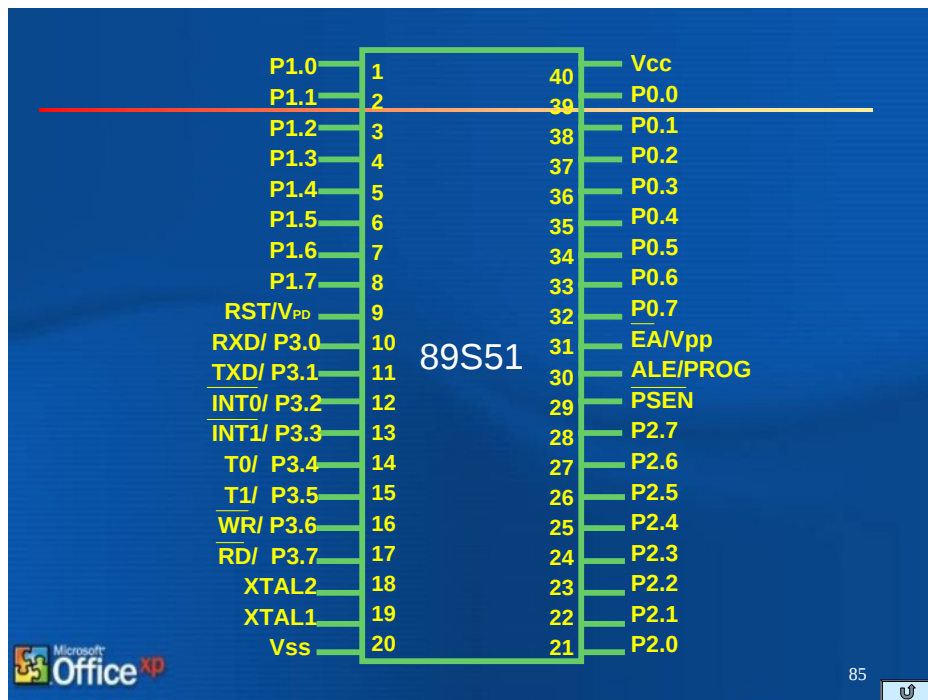
2.5 并行I/O端口

2.6 时钟电路与时序

2.7 复位操作和复位电路

§ 2.5 89S51单片机的并行端口结构与操作

89S51单片机有4个I/O端口，每个端口都是8位准双向口，共占32根引脚。每个端口都包括一个锁存器(即专用寄存器P0~P3)、一个输出驱动器和输入缓冲器。通常把4个端口笼统地表示为P0~P3。



85

在无片外扩展存储器的系统中，这4个端口的每一位都可以作为准双向通用I/O端口使用。在具有片外扩展存储器的系统中，P2口作为高8位地址线，P0口分时作为低8位地址线和双向数据总线。

89S51单片机4个I/O端口线路设计的非常巧妙，学习I/O端口逻辑电路，不但有利于正确合理地使用端口，而且会给设计单片机外围逻辑电路有所启发。

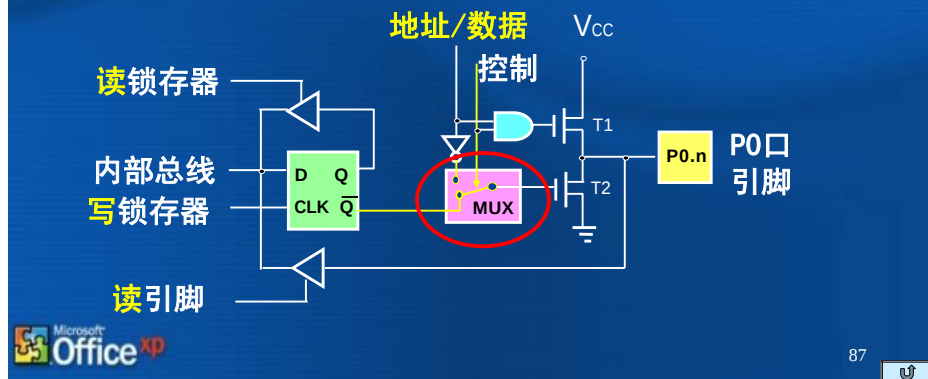
下面简单介绍一下输入/输出端口结构。

86

2.5.1 P0口和P2的结构

一、P0口的结构

下图为P0口的某位P0.n (n=0~7) 结构图，它由一个输出锁存器、两个三态输入缓冲器和输出驱动电路及控制电路组成。从图中可以看出，P0口既可以作为I/O用，也可以作为地址/数据线用。

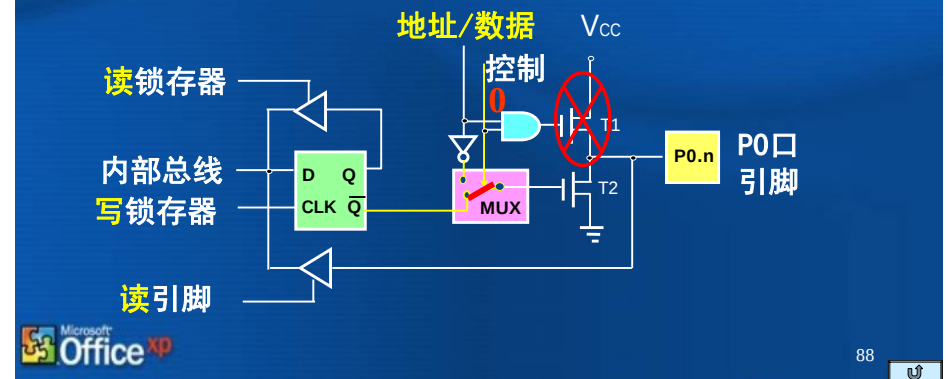


87

1、P0口作为普通I/O口

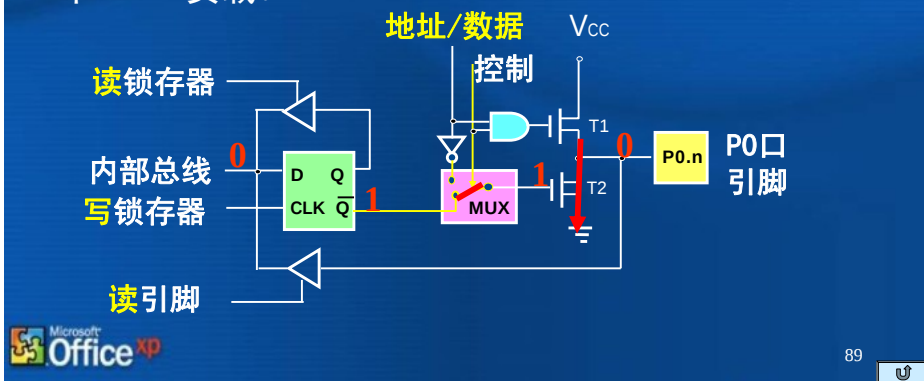
①输出时

CPU发出控制电平“0”封锁“与”门，将输出上拉场效应管T₁截止，同时使多路开关MUX把锁存器与输出



88

驱动场效应管 T_2 栅极接通。故内部总线与P0口同相。由于输出驱动级是漏极开路电路，若驱动NMOS或其它拉流负载时，需要外接上拉电阻。P0的输出级可驱动8个LSTTL负载。

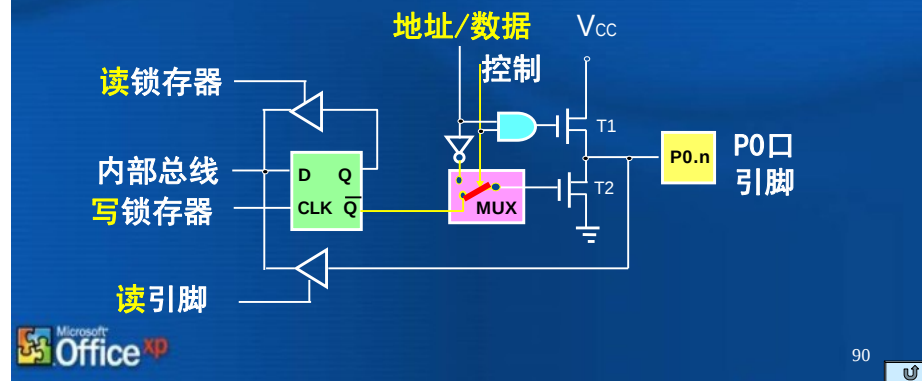


89

② 输入时——分读引脚或读锁存器

读引脚：由传送指令(MOV)实现；

下面一个缓冲器用于读端口引脚数据，当执行一条由端口输入的指令时，读脉冲把该三态缓冲器打开，这样端口引脚上的数据经过缓冲器读入到内部总线。



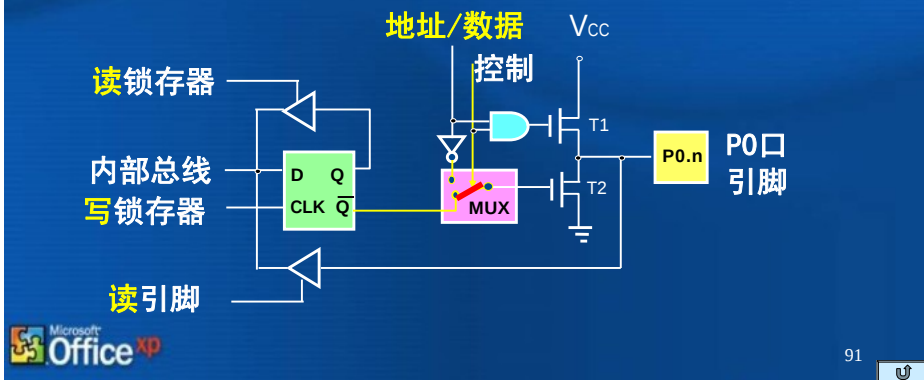
90

② 输入时——分读引脚或读锁存器

读锁存器：有些指令如：ANL P0, A称为“读-改-写”

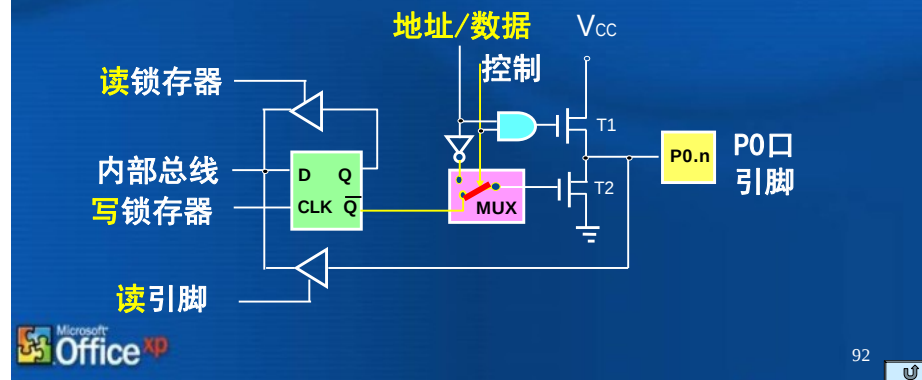
指令，需要读锁存器。

上面一个缓冲器用于读端口锁存器数据。



91

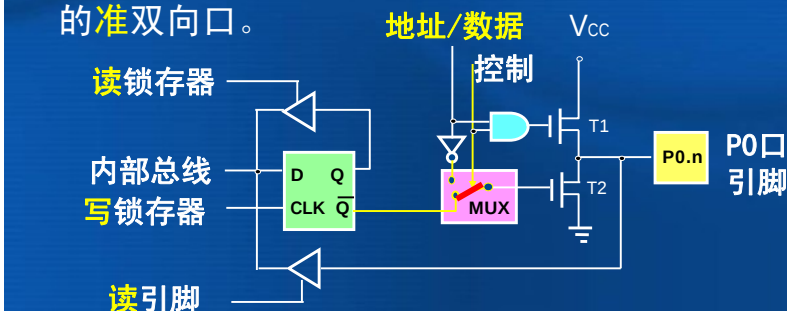
****原因：**如果此时该端口的负载恰是一个晶体管基极，且原端口输出值为1，那么导通了的PN结会把端口引脚高电平拉低；若此时直接读端口引脚信号，将会把原输出的“1”电平误读为“0”电平。现采用读输出锁存器代替读引脚，图中，上面的三态缓冲器就为读锁存器Q端信号而设，读输出锁存器可避免上述可能发生的错误。******



92

准双向口:

从图中可以看出, 在读入端口数据时, 由于输出驱动FET并接在引脚上, 如果T2导通, 就会将输入的高电平拉成低电平, 产生误读。所以在端口进行输入操作前, 应先向端口锁存器写“1”, 使T2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。

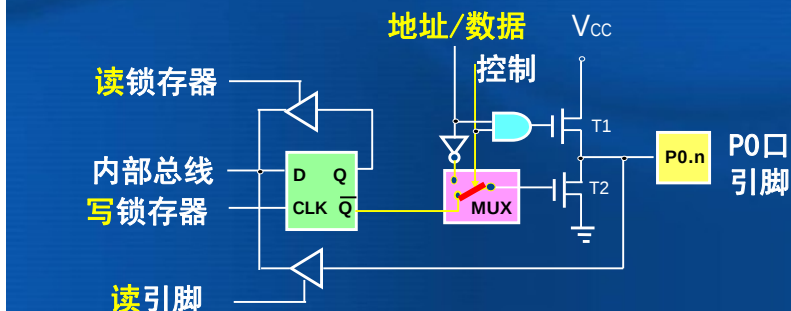


93

2、P0作为地址/数据总线

在系统扩展时, P0端口作为地址/数据总线使用时, 分为:

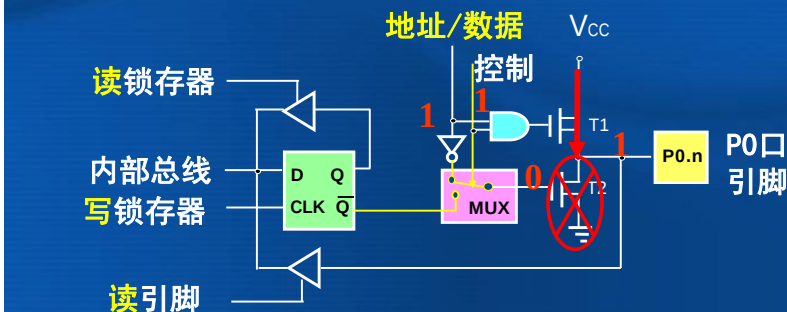
- P0引脚输出地址/数据信息。



94

2、P0作为地址/数据总线

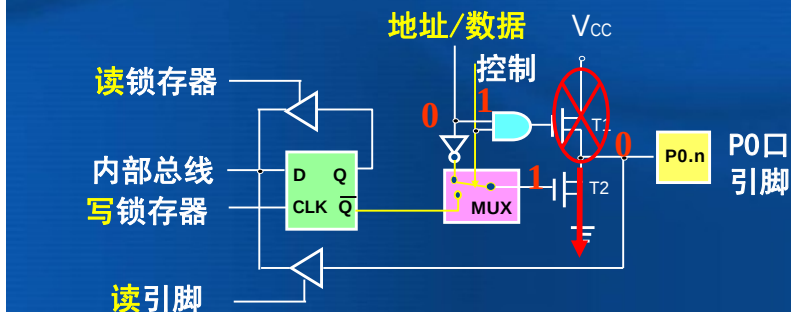
CPU发出控制电平“1”, 打开“与”门, 又使多路开关MUX把CPU的地址/数据总线与T2栅极反相接通, 输出地址或数据。由图上可以看出, 上下两个FET处于反相, 构成了推拉式的输出电路, 其负载能力大大增强。



95

2、P0作为地址/数据总线

CPU发出控制电平“1”, 打开“与”门, 又使多路开关MUX把CPU的地址/数据总线与T2栅极反相接通, 输出地址或数据。由图上可以看出, 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。



96

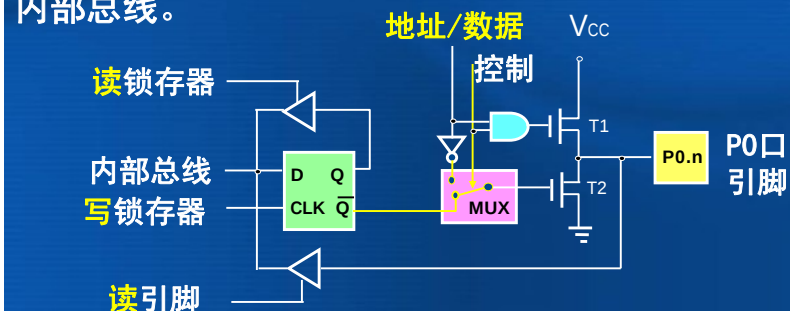
2、P0作为地址/数据总线

----真正的双向口

■ P0引脚输出地址/输入数据

输入信号是从引脚通过输入缓冲器进入内部总线。

此时，CPU自动使MUX向下，并向P0口写“1”，“读引脚”控制信号有效，下面的缓冲器打开，外部数据读入内部总线。

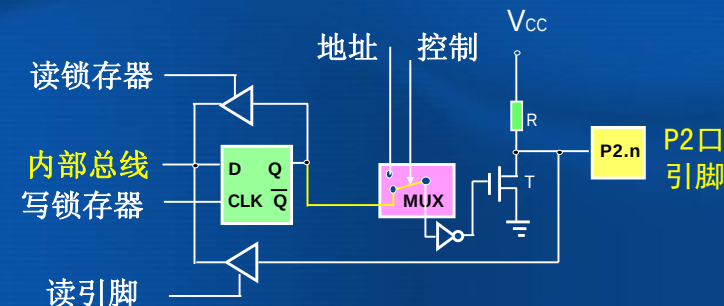


97

二、P2的内部结构

1. P2口作为普通I/O口

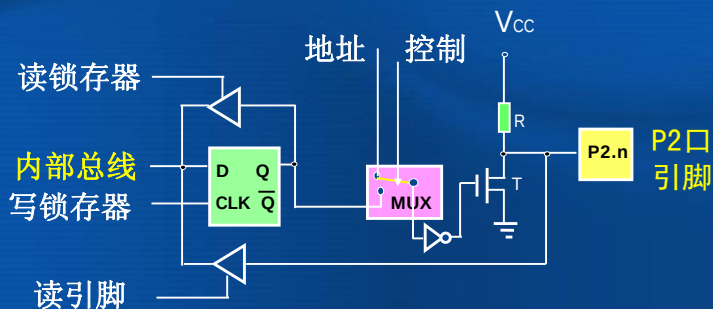
CPU发出控制电平“0”，使多路开关MUX倒向锁存器输出Q端，构成一个准双向口。其功能与P1相同。



98

2. P2口作为地址总线

在系统扩展片外程序存储器扩展数据存储器且容量超过256B（用MOVX @DPTR指令）时，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。

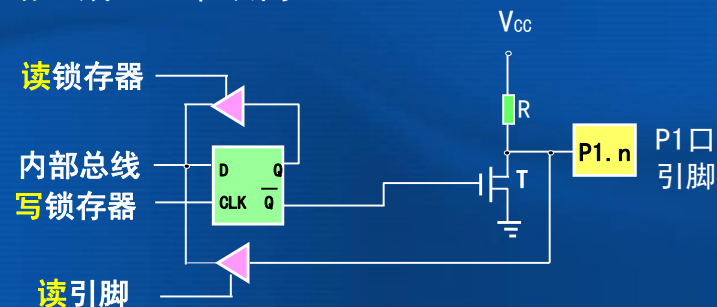


99

2.3.2 P1口、P3口的内部结构

①P1口的一位的结构

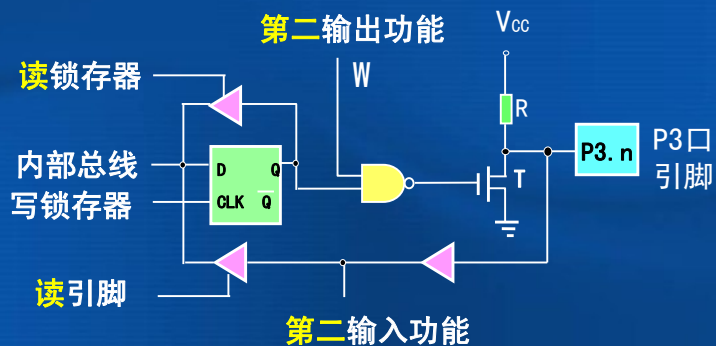
它由一个输出锁存器、两个三态输入缓冲器和输出驱动电路组成----准双向口。



100

②P3的内部结构

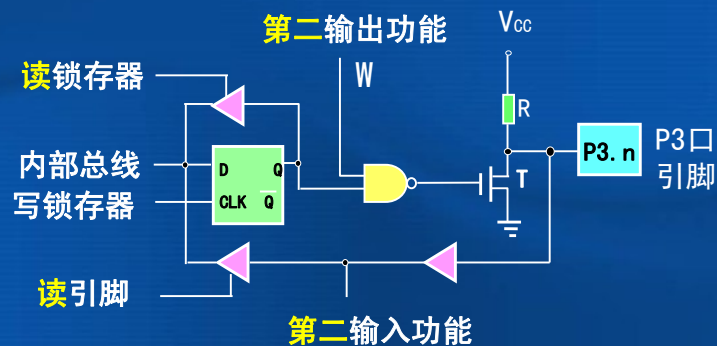
一、作为通用I/O口与P1口类似——准双向口(W=1)



②P3的内部结构

二、P3第二功能(Q=1)

此时引脚部分输入(Q=1、W=1)，部分输出(Q=1、W输出)。



P3第二功能各引脚功能定义:

- P3. 0: RXD 串行口输入
- P3. 1: TXD 串行口输出
- P3. 2: $\overline{\text{INT0}}$ 外部中断0输入
- P3. 3: $\overline{\text{INT1}}$ 外部中断1输入
- P3. 4: T0 定时器0外部输入
- P3. 5: T1 定时器1外部输入
- P3. 6: $\overline{\text{WR}}$ 外部写控制
- P3. 7: $\overline{\text{RD}}$ 外部读控制

2.5.5 P0~P3端口功能总结

使用中应注意的问题:

- P0~P3口都是并行I/O口，但P0口和P2口还可用来构建数据总线和地址总线，所以电路中有一个MUX，进行转换。
- 而P1口和P3口无构建系统的数据总线和地址总线的功能，因此，无需转接开关MUX。
- 只有P0口是一个真正的双向口，P1~P3口都是准双向口。

原因:P0口作数据总线使用时，为保证数据正确传送，需解决芯片内外的隔离问题，即只有在数据传送时芯片内外才接通；否则应处于隔离状态。为此，P0口的输出缓冲器应为三态门。

- **P3口具有第二功能。因此在P3口电路增加了第二功能控制逻辑。这是P3口与其它各口的不同之处。**

课堂复习

补充题:

1. 89S51单片机的哪些端口，有两种功能？分别是什么功能？
2. 在系统扩展片外程序存储器时，P2口是否可以再作为通用I/O口？

3. 某控制程序有如下片段:

```
MOV P1, #00H
```

```
⋮
```

```
(MOV P1, #0FFH)
```

```
MOV A, P1
```

执行后结果不正常，请问为什么会出现这种情况，如何修改程序，才能保证输入的数据正常？

§ 2.6 89S51单片机片外总线 and 复位

2.4.1 89S51单片机片外总线配置

地址总线:

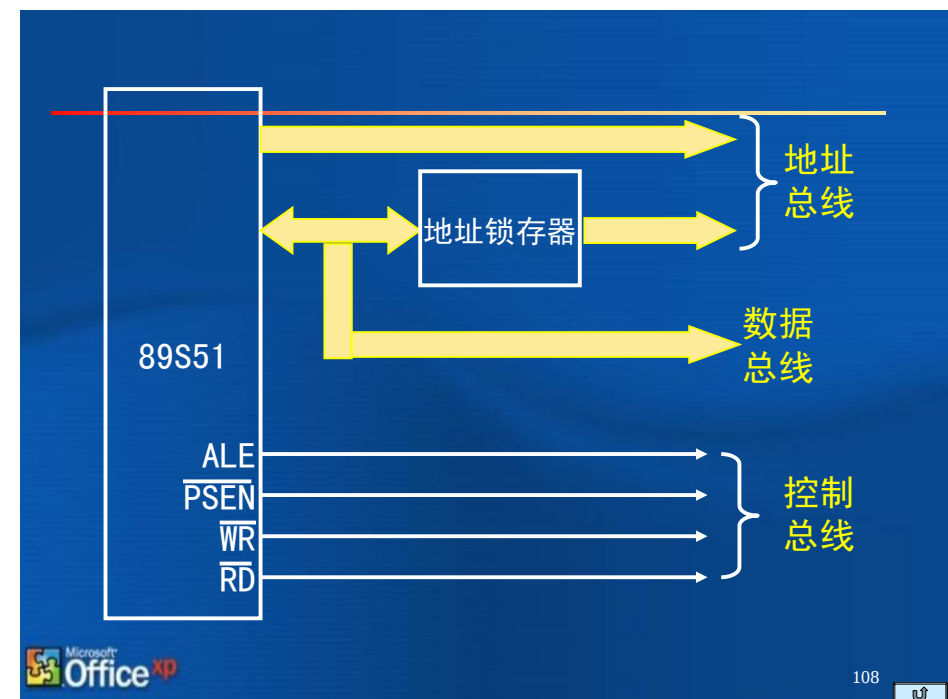
P2口（高8位A15~A8）和P0口（低8位A7~A0）
可寻址64KB。

数据总线:

P0口（D7~D0）

控制总线:

ALE、/PSEN、/WR、/RD



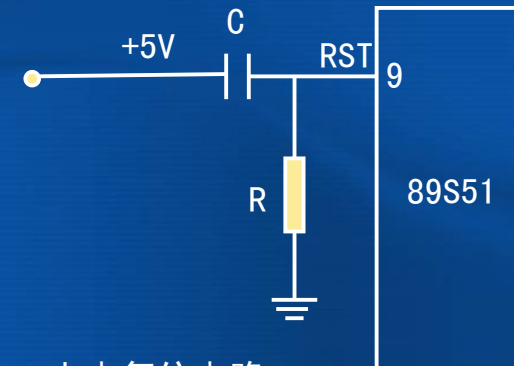
2.6.2 89S51单片机复位(RST高电平2个机器周期)

任何单片机在工作之前都要有个复位的过程，复位是什么意思呢？对单片机来说，是程序还没有开始执行，是在做准备工作——**初始状态**。显然，准备工作不需要太长的时间，复位需要不少于24个时钟周期的时间就可以了。

如何进行复位呢？只要在单片机的RST引脚上加上高电平，就可以了。为了达到这个要求，可以用很多种方法，见图。

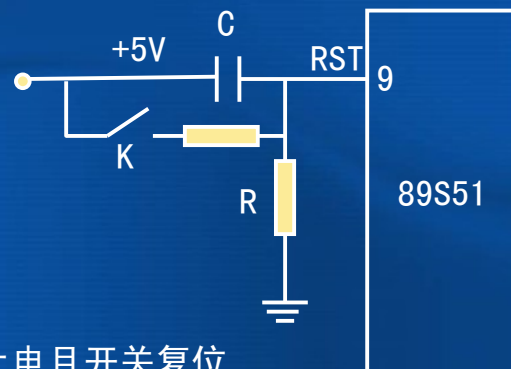
复位后，PC指向**0000H**，使单片机从起始地址**0000H**开始执行程序。

一、复位电路（两种）



上电复位电路

一、复位电路（两种）



上电且开关复位

二、复位后单片机的状态

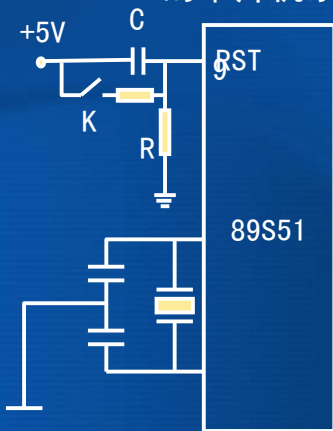
复位后各寄存器的状态

表2-7 复位时片内各寄存器的状态

寄存器	复位状态	寄存器	复位状态
PC	0000H	TMOD	00H
Acc	00H	TCON	00H
PSW	00H	TH0	00H
B	00H	TL0	00H
SP	07H	TH1	00H
DPTR	0000H	TL1	00H
P0~P3	FFH	SCON	00H
IP	×××0 0000B	SBUF	×××× ××××B
IE	0××0 0000B	PCON	0××× 0000B
DP0L	00H	<u>AUXR</u>	×××× 0××0 B
DP0H	00H	<u>AUXR1</u>	×××× ×××0 B
<u>DP1L</u>	00H	<u>WDTRST</u>	×××× ××××B
<u>DP1H</u>	00H		

2.4.3 89S51单片机最小系统

最小系统----最少外部电路条件下，可以独立工作的单片机系统。



113

§ 2.7 89S51单片机片时序

CPU时序及有关概念p29

计算机工作要有严格的时序。事实上，计算机更象一个大钟，什么时候分针动，什么时候秒针动，什么时候时针动，都有严格的规定，一点也不能乱。基本时序单位：

114

1. **振荡周期**：晶体振荡器的周期。

2. **状态周期**：振荡周期2分频（P1\P2），也称时钟周期。

3. **机器周期**：一个机器周期包括6个时钟周期。（新CPU不同）

4. **指令周期**：执行一条指令的时间。

INTEL对每一条指令都给出了它的指令周期数，89S51单片机的所有指令中，有一些完成得比较快，只要一个机器周期就行了，有一些完成得比较慢，得要2个机器周期，还有两条指令要4个机器周期才行。

115

2.7.2 机器周期、指令周期与指令时序

一、时钟周期

单片机的基本时间单位。

若时钟的晶体的振荡频率为 f_{osc} ，则时钟周期 $T_{osc}=1/f_{osc}$ 。如 $f_{osc}=6\text{MHz}$ ， $T_{osc}=166.7\text{ns}$ 。

二、机器周期

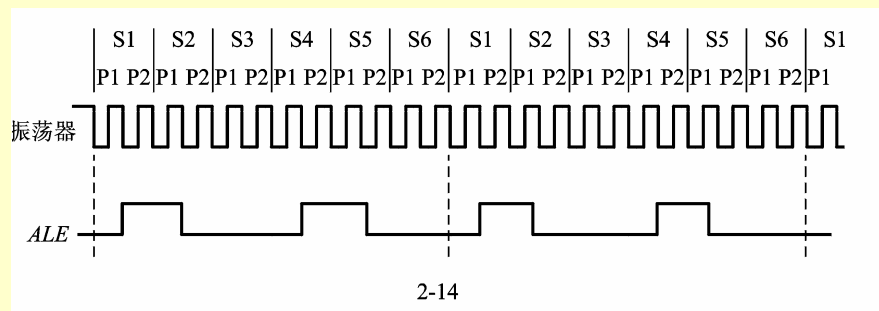
CPU完成一个基本操作所需要的时间。

执行一条指令分为几个机器周期。每个机器周期完成一个基本操作。89S51单片机每12个时钟周期为一个机器周期，

116

一个机器周期又分为6个状态：**S1~S6**。每个状态又分为两拍：**P1和P2**。因此，一个机器周期中的12个时钟周期表示为：

S1P1、S1P2、S2P1、S2P2、...、S6P2。



117

三、指令周期

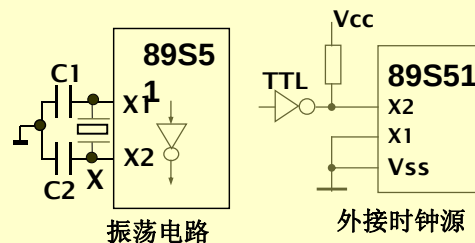
执行一条指令时，可分为**取指令阶段**和**指令执行阶段**。

- **取指令阶段**，PC中地址送到程序存储器，并从中取出需要执行指令的操作码和操作数。
- **指令执行阶段**，对指令操作码进行译码，以产生一系列控制信号完成指令的执行。
- **ALE信号是为地址锁存而定义的**，以时钟脉冲1/6的频率出现，在一个机器周期中，ALE信号两次有效（注意，在执行访问外部数据存储器的指令MOVX时，将会丢失一个ALE脉冲）

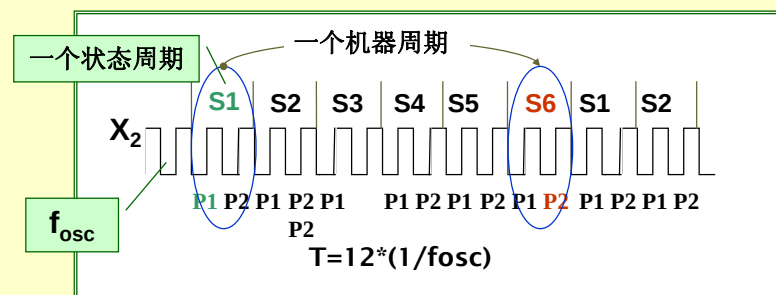
118

时钟电路

- 需外接晶振的频率 **1.2~12MHz**，C1和C2取 **30±10PF**



- **CPU的时序**（时钟周期、状态周期、机器周期）



119

若外接晶振为12MHz时，则单片机的四个周期的具体值为：

- ✓ 时钟周期 = $1/12\text{MHz} = 1/12 \mu\text{s} = \mathbf{0.0833 \mu\text{s}}$
- ✓ 状态周期 = $1/6 \mu\text{s} = \mathbf{0.167 \mu\text{s}}$
- ✓ 机器周期 = $\mathbf{1 \mu\text{s}}$
- ✓ 指令周期 = $\mathbf{1 \sim 4 \mu\text{s}}$

可用于计算指令、程序的执行时间，以及定时器的定时时间

120

§ 2.8 看门狗定时器（WDT）的使用

单片机系统受到干扰可能会引起程序“跑飞”或“死循环”，使系统失控。如果操作人员在场，可按人工复位按钮，强制系统复位。但操作人员不可能一直监视着系统，即使监视着系统，也往往是在引起不良后果之后才进行人工复位。

“看门狗”技术：使用一个定时器来不断计数，监视程序的运行。当看门狗启动运行后，为防止看门狗的不必要溢出，在程序正常运行过程中，应定期地把看门狗清0，以保证看门狗不溢出。



121



在C51语言编程中，若使用看门狗功能，由于头文件reg51.h中，并没有声明WDTRST寄存器，所以必须先声明WDTRST寄存器，如：

```
sfr WDTRST=0xa6
```

声明后可以用命令启动或复位看门狗。

```
WDTRST=0x1e;
```

```
WDTRST=0xe1;
```



122



【例】看门狗的使用举例。

```
#include<reg51.h>
```

```
sfr WDTRST=0xa6
```

```
main( )
```

```
{
```

```
.....
```

```
WDTRST=0x1e;           //启动看门狗运行
```

```
WDTRST=0xe1;
```

```
while(1)               //无限循环
```

```
{
```

```
WDTRST=0x1e; //清0并启动看门狗运行
```

```
WDTRST=0xe1;
```

```
.....;
```



123



上述程序在无限循环执行中运行，通过看门狗定时器可防止程序在执行过程中“跑飞”或“死循环”，因为只要程序一跑出while()循环，不执行复位看门狗的两条复位命令，看门狗定时器由于得不到及时复位，就会溢出使单片机复位，使程序从main()处开始重新运行。所以使用看门狗时一定要注意，一定要在看门狗启动后的16384μs（系统时钟为12MHz时）之内清0。



124

