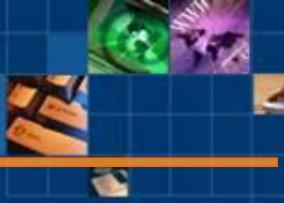




第三章 MCS-51单片机的 指令系统

3.1 指令系统概述



3.1.1 指令的概念

1. 机器码指令和汇编语言指令

- **指令**：指挥计算机工作的命令。一种计算机所能识别和执行的指令的集合，称为该计算机的指令系统。指令可分为**机器码指令**和**汇编语言指令**；
- **机器码指令**：用二进制代码表示的指令，能够直接被计算机硬件识别和执行。例如执行累加器A加1的操作的指令，二进制代码为00000100B，即04H；
- **汇编语言指令**：用助记符表示的指令。必须通过汇编程序将其转化为机器码指令才能供CPU执行。例如：

MOV A,#5FH ;将5FH送入A，机器码为74H,5FH



3.1.1 指令的概念

2. 汇编语言指令格式

[标号:] 操作码助记符 [目的操作数,][源操作数][;注释]

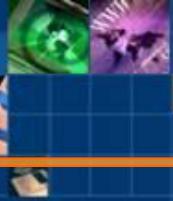
- 标号：指令的符号地址。通常代表一条机器指令代码的存储单元地址。标号可作为其它指令的操作数使用；
- 操作码助记符：用方便记忆的方式表示指令进行何种操作。



3.1.1 指令的概念

2. 汇编语言指令格式

- 操作数：指令操作的对象。分为目的操作数和源操作数，顺序上不可颠倒。操作数可以是数字（地址、数据），也可以是标号或寄存器名等，某些指令不需要操作数；
- 注释：为了加强程序的可读性，可增加适当的注释。注释不是程序代码，仅起说明作用。



3.1.1 指令的概念

2. 汇编语言指令格式

- MCS-51系列单片机使用42种助记符，有51种基本操作。通过助记符、指令中的源、目的操作数的不同组合，构成了MCS-51的111条指令；
- MCS-51指令系统中有一个位变量处理的指令子集，使其非常适合工业控制场合使用。

MCS-51
指令系统

按指令字节数分：
单字节49条； 双字节46条； 三字节16条

按指令执行周期分：
单周期57条； 二周期52条； 四周期2条



第三章 MCS-51单片机的 指令系统

3.2 寻址方式

单片机原理及系统设计



- MCS-51程序中所有的代码、数据、SFR等均有自己唯一的地址，组成指令的操作数指定了参与运算的数据、数据所在的存储单元、寄存器、I/O口地址等。指令中规定的寻找操作数的方式称为寻址方式；
- 寻址方式越多，计算机的功能就越强，灵活性就越大。MCS-51共有7种寻址方式：

单片机原理及系统设计



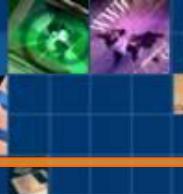
- 单片机各种寻址方式与可寻址存储空间及所使用的寄存器：

序号	寻址方式	各自可寻址的存储空间及使用的寄存器
1	立即寻址	存储在程序存储器中的立即数
2	寄存器寻址	R0 ~ R7、A、B、DPTR、C及存储器
3	间接寻址	片内和片外数据存储器空间，使用Ri、DPTR
4	直接寻址	内部RAM和特殊功能寄存器
5	变址寻址	程序存储器空间，使用A、DPTR或PC
6	相对寻址	程序存储器空间，使用PC
7	位寻址	Cy、片内RAM中的位空间和特殊功能寄存器中可位寻址的位



3.2.1 立即寻址

- 定义：操作数包含在指令字节中，即操作数以字节的形式存放在程序存储器中，这种方式称为立即寻址；
- 特点：指令中直接含有所需的操作数。该操作数可以是8位的，也可以是16位的，常常处在指令的第二字节和/或第三字节的位置上。立即数通常使用#data或#data16表示，在立即数前面加“#”标志，用以和直接寻址中的直接地址（direc或bit）相区别。



3.2.1 立即寻址

- 例如：

```
MOV DPTR, #8000H ;将立即数8000H存入DPTR寄存器中  
ADD A, #34H        ;把立即数34H与累加器A内容相加,  
                    ;结果存回累加器A中
```

- ADD A,#34H指令的执行过程如下页图所示：

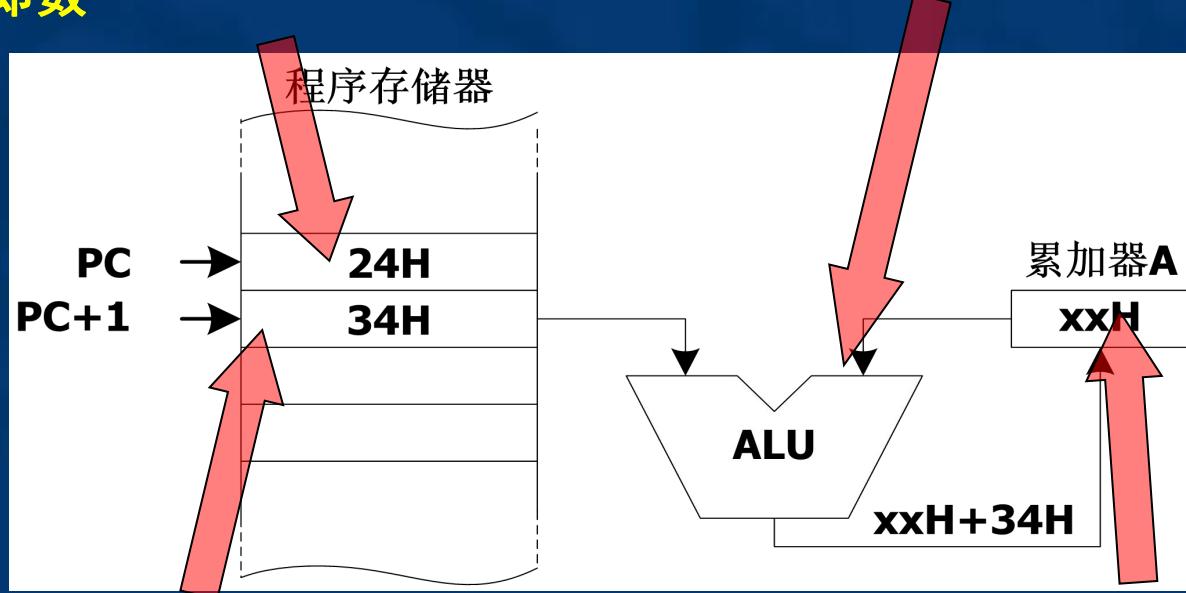


3.2.1 立即寻址

- ADD A,#34H指令的执行过程

指令执行到此处,指令代码24H意为
ADD A,#立即数

A中原来的内容送ALU



PC+1后, 从该地址取出
立即数送ALU准备与A运算

最后运算结果送回A



3.2.2 寄存器寻址

- 定义：操作数存放在MCS-51内部的某个工作寄存器Rn(R0~R7)或部分专用寄存器中，这种寻址方式称为寄存器寻址；
- 特点：由指令指出某一个寄存器的内容作为操作数。存放操作数的寄存器在指令代码中不占据单独的一个字节，而是嵌入（隐含）到操作码字节中。
- 例如：

INC R0 ;将R0中的内容加1

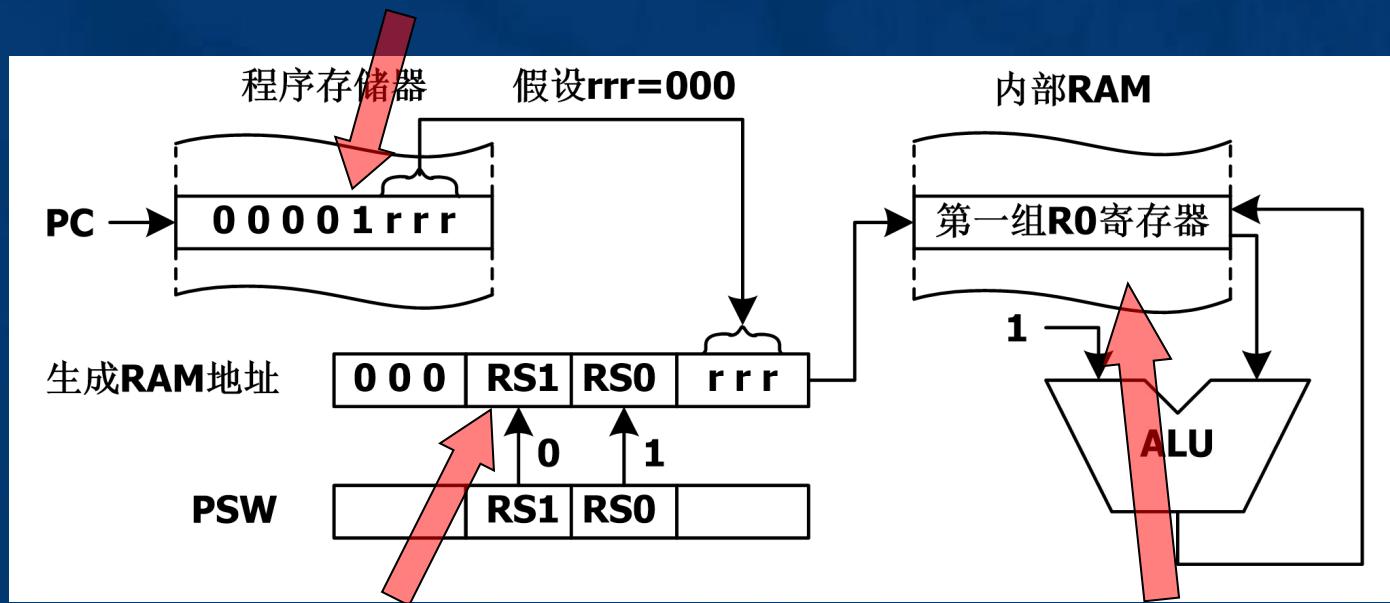


3.2.2 寄存器寻址

- INC Rn 指令的执行过程：

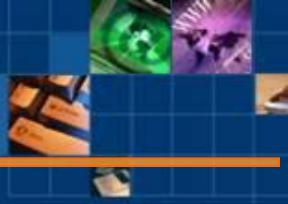
指令代码10~17H表示将R0~7内容加

1



取出指令代码低3位,和RS1,RS0
构成一个字节的内部RAM地址

将此单元内容
+1后送回



3.2.3 间接寻址

- **定义**: 又称寄存器间接寻址，由指令指出某一个寄存器的内容作为操作数的地址，称为间接寻址；
- **特点**: 指令中寄存器中存放的是操作数地址。间接寻址是一种二次寻找操作数地址的寻址方式，寄存器名前必须加前缀符号“@”；
- **寻址范围**: 内部RAM低128字节（只能使用R0或R1作间址寄存器）、外部RAM（只能使用DPTR作间址寄存器）。对于外部低256单元RAM的访问，除可以使用DPTR外，还可以使用R0或R1作间址寄存器。不能用于寻址特殊功能寄存器（SFR）。



3.2.3 间接寻址

- 例如：

MOV A, @R0

;将以R0的值为地址的内部
;RAM单元的内容送累加器A

MOVX A, @R0

;将以R0的值为地址的外部
;RAM单元的内容送累加器A

MOV DPTR, #8000H

;DPTR = 8000H

MOVX A, @DPTR

;外部RAM 8000H单元送A

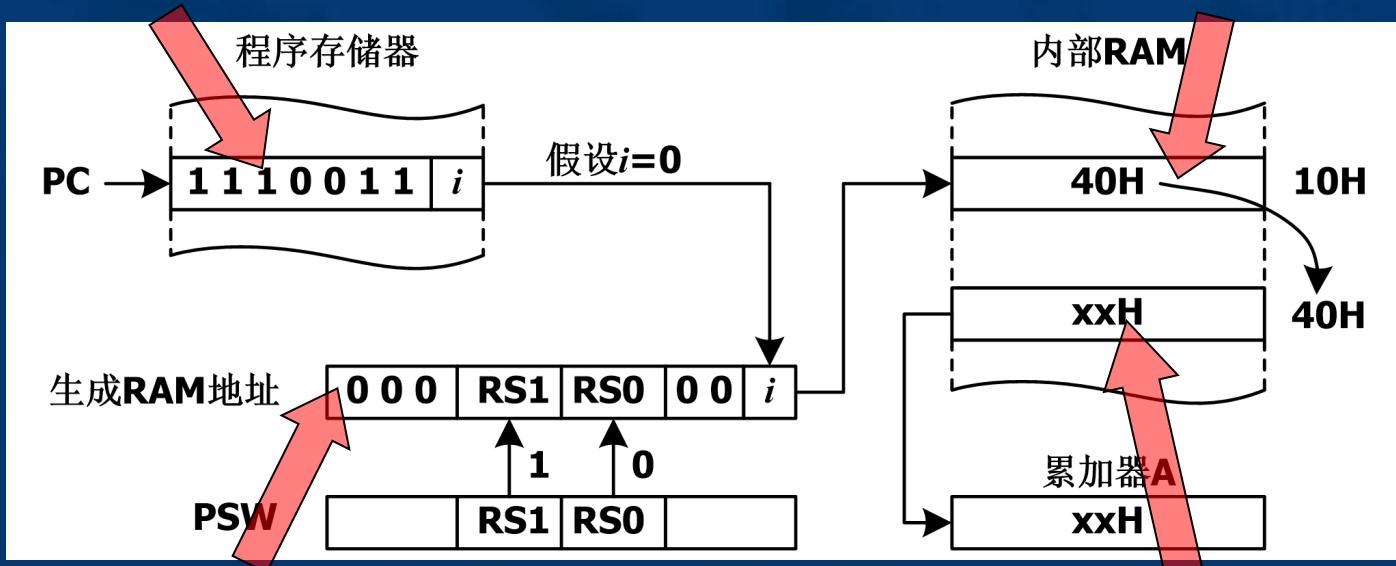


3.2.3 间接寻址

- MOV A, @Ri 指令的执行过程

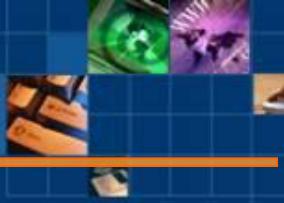
指令代码E6/E7H表示R0/R1间接寻址

访问10H单元，取出其内容，假设为40H



取出指令代码最低位，加上
RS1,RS0构成R0/R1的实际地址，
例中为10H

访问40H单元，取出其
内容，送入累加器A



3.2.4 直接寻址

- **定义**: 在指令中直接给出操作数所在的存储单元的地址;
- **特点**: 指令中含有操作数的地址。该地址指出了参与操作的数据所在的字节单元地址或位地址。CPU执行它们时便可根据直接地址找到所需要的操作数;
- **寻址范围**: 直接寻址方式中操作数的存储空间共有三种, 即:
片内RAM区、SFR和位地址空间。



3.2.4 直接寻址

- 例如：

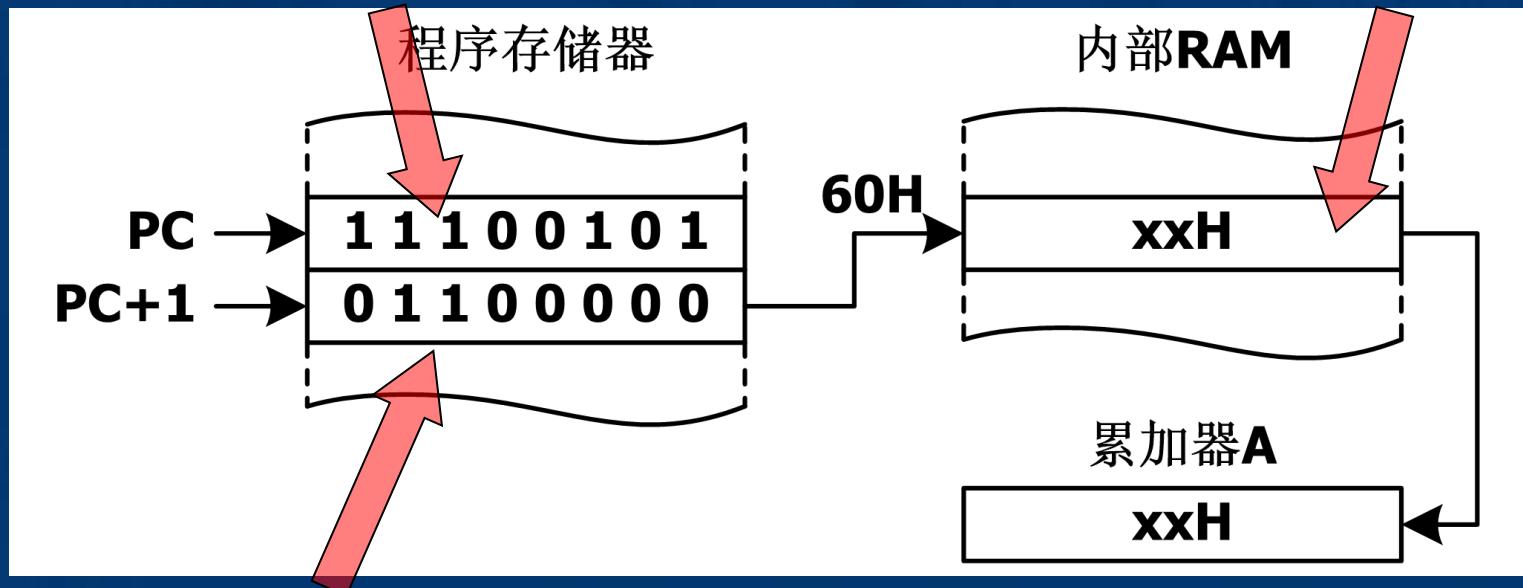
MOV A, 60H ; 内部RAM 60H单元内容送A
MOV C, 60H ; 位地址60H中的值送Cy
MOV P1, A ; A中内容送特殊功能寄存器P1

3.2.4 直接寻址

- MOV A, direct 指令的执行过程

指令代码E5H表示将某存储单元内容送A

访问该地址，取出其内容送入累加器A



PC+1后取出该地址的内容，即为存储单元地址

3.2.5 变址寻址

- **定义**: 操作数存放在变址寄存器 (A) 和基址寄存器 (DPTR或PC) 相加形成的16位地址单元中。这种寻址方式称为基址加变址寄存器间接寻址，简称变址寻址；
- **特点**: 变址寻址方式是单字节指令，操作码中隐含作为基址寄存器使用的DPTR (或PC) 和作为变址用的A。在执行变址寻址指令时，MCS-51单片机先把基地址 (DPTR或PC 的内容) 和地址偏移量 (A中的内容) 相加，以形成操作数地址，再由操作数地址找到操作数；
- **寻址范围**: 只能对程序存储器 (ROM) 进行寻址，主要用于查表性质的访问。



3.2.5 变址寻址

- 注意：累加器A中存放的操作数地址相对基地址的偏移量的范围为00H ~ 0FFH(无符号数)；
- 例如：

(1) 以PC作为基址

MOVC A, @A+PC ;(PC) \leftarrow (PC)+1, (A) \leftarrow ((A)+(PC))

(2) 以DPTR作为基址

MOV DPTR, #2335H

MOV A, #10H

MOVC A, @A+DPTR ;(A) \leftarrow ((A)+(DPTR))



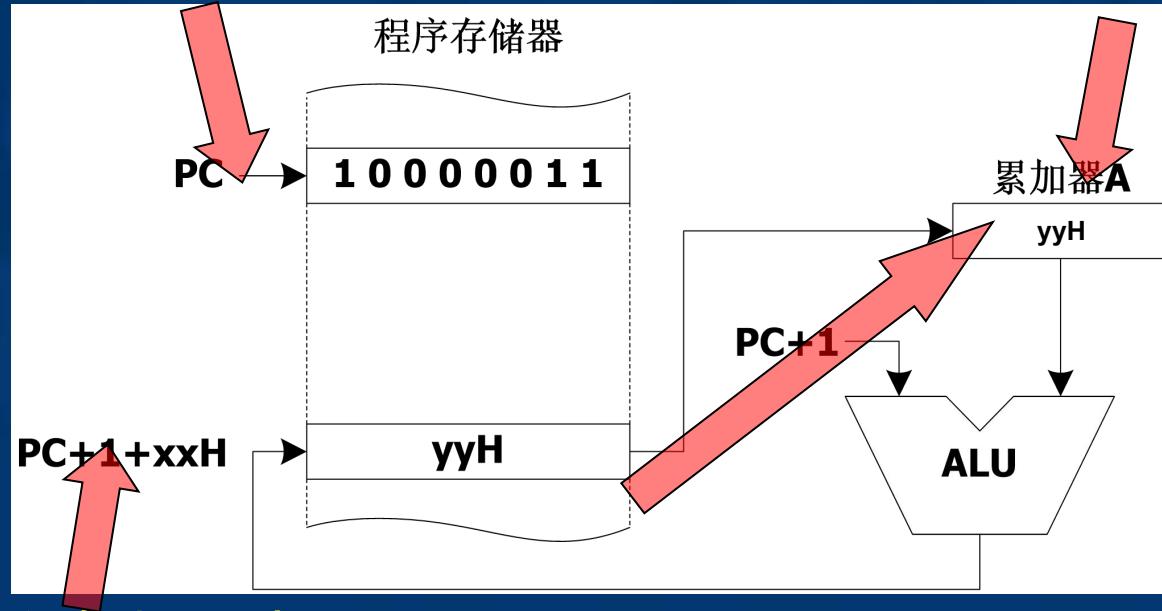
3.2.5 变址寻址

- MOVC A,@A+PC 指令的执行过程

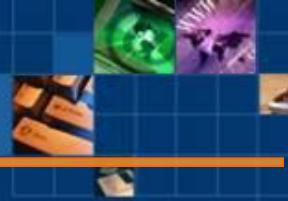
指令代码83H表示

MOVC A, @A+PC

将PC+1和累加器A中的
内容相加得到访问地址



根据得到的地址取出相
应单元内容,送入累加器A



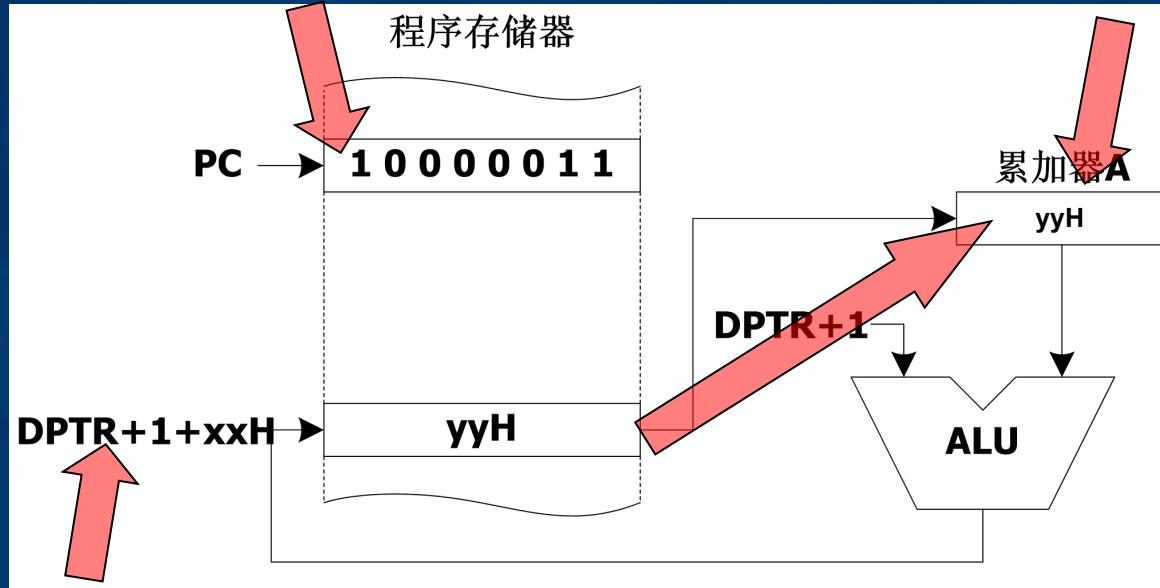
3.2.5 变址寻址

- MOVC A,@A+DPTR指令的执行过程

指令代码93H表示

MOVC A, @A+DPTR

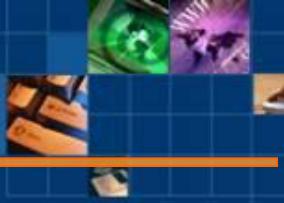
将DPTR和累加器A中的
内容相加得到访问地址



根据得到的地址取出相
应单元内容,送入累加器A

3.2.6 相对寻址

- **定义**：将PC的当前值（取出本条指令后的值）与指令第二个字节给出的偏移量（rel）相加，形成新的转移目标地址，称为相对寻址方式；
- **特点**：相对寻址方式是为实现程序的相对转移而设计的，其指令码中含有的相对地址偏移量可为正数或负数，能生成浮动代码；
- **寻址范围**：只能对程序存储器ROM进行寻址。相对地址偏移量（rel）是一个带符号的8位二进制补码，因此其取值范围为-128 ~ +127（以PC为中间点的256个字节范围）。

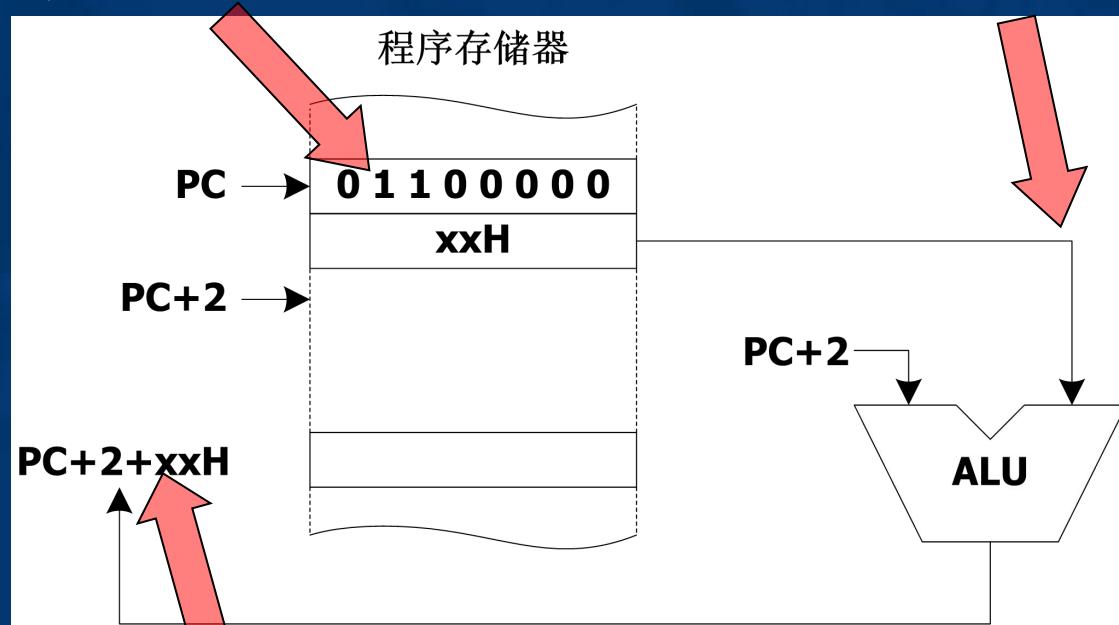


3.2.6 相对寻址

- JZ xxH 指令的执行过程

指令代码60H
表示JZ xxH

如果Z标志有效,则取出指令代码下一
字节的相对偏移量,将其和PC+2相加



相加后的结果送PC, 控制
程序转向该地址执行



3.2.7 位寻址

- **定义**: 指令中给出的操作数是一个可单独寻址的位地址, 这种寻址方式称为位寻址方式;
- **特点**: 位寻址是直接寻址方式的一种, 其特点是对8位二进制数中的某一位的地址进行操作;
- **寻址范围**: 片内RAM的位寻址区 (20H~2FH字节, 共128个位地址) 、部分SFR (对于标准的MCS-51, 其中有83位可以位寻址) 。



3.2.7 位寻址

- 位寻址指令举例：

(1) 直接使用位地址形式。如：

MOV 00H, C ;(00H)←(Cy)

(2) 字节地址加位序号的形式。如：

MOV 20H.0, C ;(20H.0)←(Cy)

(3) 位的符号地址(位名称)的形式。对于部分特殊功能寄存器，其各位均有一个特定的名字，可用来看访问该位。如：

ANL C, P ;(C)←(C) .AND. (P)

(4) 字节符号地址(字节名称)加位序号的形式。对于部分特殊功能寄存器(如PSW)，可以用其字节名称加位序号形式来访问某一位。如：

CPL PSW.0 ;P← .NOT. P



第三章 MCS-51单片机的 指令系统

3.3 指令分类介绍



● 符号使用约定

Rn

当前寄存器区的寄存器R0~R7

@Ri

通过R0、R1间接寻址片内RAM单元

direct

直接地址，一个内部RAM单元地址或一个
特殊功能寄存器(SFR,8 bit)

#data

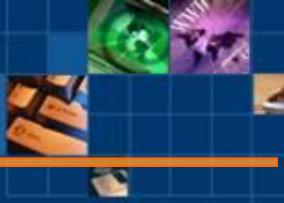
8位或16位常数，也称立即数

addr16

16位目的地址，供LCALL/LJMP指令使用

addr11

11位目的地址，供ACALL/AJMP指令使用



● 符号使用约定

rel 8位带符号偏移量(二进制补码表示), 用于
相对转移指令

bit 位地址

/ 位操作的取反前缀

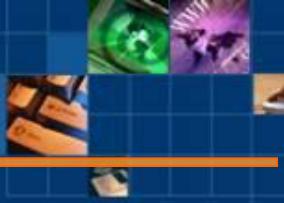
(x) 表示取x地址单元中的内容

((x)) 表示取x地址单元中的内容为地址单元的内容

\$ 当前指令的地址

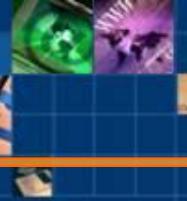
← 数据传输方向

↔ 数据交换



3.3.1 数据传送类指令

- 数据传送指令是MCS-51指令系统中最常用的指令，指令的执行是将源操作数传送到目的操作数，而源操作数保持不变。MCS-51的数据传送操作可以在内部数据存储器、外部数据存储器和程序存储器之间进行。



3.3.1 数据传送类指令

1. 内部数据传送指令

基本格式：

MOV 目的操作数, 源操作数

1) MOV A,<src>类指令

指令	操作说明	执行周期
MOV A,Rn	Rn (n=0~7) 中内容送A	1
MOV A,direct	内部RAM的direct单元中内容送A	
MOV A,@Ri	Ri (i=0,1)内容所对应的地址单元送A	
MOV A,#data	立即数data送A	



3.3.1 数据传送类指令

1. 内部数据传送指令

2) MOV <dest>, A类指令

指令	操作说明	执行周期
MOV Rn,A	A中内容送Rn (n=0~7)	
MOV direct,A	A中内容送内部RAM的direct单元	1
MOV @Ri,A	A送Ri (i=0,1)内容所对应的地址单元	



3.3.1 数据传送类指令

1. 内部数据传送指令

3) MOV <dest>, <src>类指令

指令	操作说明	执行周期
MOV Rn,direct	RAM中direct单元→Rn	2
MOV Rn,#data	立即数data→Rn	
MOV direct,Rn	Rn→RAM中direct单元	
MOV direct1,direct2	RAM中direct2单元→direct1	
MOV direct,@Ri	以Ri内容为地址的单元→direct单元	
MOV direct,#data	立即数data→direct单元	
MOV @Ri,direct	Direct单元→以Ri内容为地址的单元	
MOV@Ri,#data	立即数→以Ri内容为地址的单元	



3.3.1 数据传送类指令

1. 内部数据传送指令

4) 其它类数据传送指令

指令	操作说明	执行周期
MOV DPTR,#data16	16位立即数→DPTR	2
PUSH direct	SP=SP+1后，将direct单元内容放入堆栈	2
POP direct	从堆栈中弹出一个字节，放入direct单元	2
XCH A,Rn	Rn中内容和A中内容互换	1
XCH A,direct	Direct单元的内容和A互换	
XCH A,@Ri	以Ri内容为地址的单元和A互换	1
XCHD A,@Ri	以Ri内容为地址的单元内容的低4位和A的低4位互换	



3.3.1 数据传送类指令

2. 外部数据传送及查表指令

1) MOVX <dest>, <src>类指令

指令	操作说明	执行周期
MOVX A,@Ri	以Ri内容为地址的外部RAM单元内容送A	2
MOVX @Ri,A	A中内容送以Ri内容为地址的外部RAM单元	
MOVX A,@DPTR	以DPTR内容为地址的外部RAM单元内容送A	
MOVX @DPTR,A	A中内容送以DPTR内容为地址的外部RAM单元	



3.3.1 数据传送类指令

2. 外部数据传送及查表指令

2) MOVC A, @A+<基址寄存器>类指令

指令	操作说明	执行周期
MOVC A,@A+PC	$A \leftarrow ((A)+(PC))$	2
MOVC A,@A+DPTR	$A \leftarrow ((A)+(DPTR))$	



3.3.2 算术运算类指令

1. 加法指令

指令	操作说明	执行周期
ADD A,Rn	$(A) \leftarrow (A) + (R_n), n=0 \sim 7$	1
ADD A,direct	$(A) \leftarrow (A) + (\text{direct})$	
ADD A,@Ri	$(A) \leftarrow (A) + ((R_i))$	
ADD A,#data	$(A) \leftarrow (A) + \#data$	

- 上述指令对程序状态字PSW中标志位的影响如下：
 - 如果运算后D7位向上有进位时，进位标志Cy=1，否则Cy=0；
 - 如果运算后D3位向上有进位时，辅助进位标志AC=1，否则AC=0；
 - 相加结果送入A中后，将根据A中1的个数调整奇偶位P。



3.3.2 算术运算类指令

2. 带进位加法指令

指令	操作说明	执行周期
ADDC A,Rn	$(A) \leftarrow (A) + (C) + (Rn), n=0 \sim 7$	1
ADDC A,direct	$(A) \leftarrow (A) + (C) + (\text{direct})$	
ADDC A,@Ri	$(A) \leftarrow (A) + (C) + ((Ri))$	
ADDC A,#data	$(A) \leftarrow (A) + (C) + \#data$	

- 上述指令对PSW中标志位的影响和加法指令相同。



3.3.2 算术运算类指令

3. 加1减1类指令

指令	操作说明	执行周期
INC A	$(A) \leftarrow (A)+1$	1
INC Rn	$(R_n) \leftarrow (R_n)+1, n=0 \sim 7$	
INC direct	$(direct) \leftarrow (direct)+1$	
INC @Ri	$((R_i)) \leftarrow ((R_i))+1, i=0,1$	
INC DPTR	$(DPTR) \leftarrow (DPTR)+1$	
DECA	$(A) \leftarrow (A)-1$	
DEC Rn	$(R_n) \leftarrow (R_n)-1, n=0 \sim 7$	
DEC direct	$(direct) \leftarrow (direct)-1$	
DEC @Ri	$((R_i)) \leftarrow ((R_i))-1, i=0,1$	

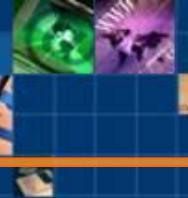


3.3.2 算术运算类指令

4. 带借位减法指令

指令	操作说明	执行周期
SUBB A,Rn	$(A) \leftarrow (A) - (C) - (Rn), n=0 \sim 7$	1
SUBB A,direct	$(A) \leftarrow (A) - (C) - (\text{direct})$	
SUBB A,@Ri	$(A) \leftarrow (A) - (C) - ((Ri))$	
SUBB A,#data	$(A) \leftarrow (A) - (C) - \#data$	

- 上述指令对程序状态字PSW中标志位的影响如下：
 - 当最高位D7有借位时，进位标志位Cy=1，否则Cy=0；
 - 当D3位有借位时，辅助进位标志位AC=1，否则AC=0；
 - 运算结束后，将根据累加器A中1的个数调整奇偶位P。



3.3.2 算术运算类指令

5. 乘除法指令

指令	操作说明	执行周期
MUL AB	A × B, 结果高8位送B, 低8位送A	4
DIV AB	A ÷ B, 商送A, 余数送B	

- 上述指令对程序状态字PSW中标志位的影响如下：
 - 乘除法运算后都将清零Cy；
 - 如果乘法结果大于0FFH, OV=1, 否则OV=0；
 - 如果除法运算中除数为0, OV=1, 否则OV=0；
 - 运算结束后, 将根据累加器A中1的个数调整奇偶位P。



3.3.2 算术运算类指令

6. 二-十进制调整指令

指令	操作说明	执行周期
DAA	如果A的低4位>9， 或辅助进位AC=1 则A的低4位加6 如果A的高4位>9， 或进位位C=1 则A的高4位加6	1



3.3.3 逻辑运算类指令

1. 对累加器的清零和取反指令

指令	操作说明	执行周期
CLR A	$(A) \leftarrow 0$	1
CPL A	$(A) \leftarrow \sim(A)$	

- 上述指令对程序状态字PSW中标志位的影响如下：
 - CLR A指令结果不影响Cy、AC及OV等标志，但P标志为0。
 - CPL A是将累加器A中的数据按位取反，不影响标志位。



3.3.3 逻辑运算类指令

2. 移位类指令

指令	操作说明	执行周期
RL A	$(An+1) \leftarrow (An), n=0 \sim 6; (A0) \leftarrow (A7)$	1
RLC A	$(An+1) \leftarrow (An), (A0) \leftarrow (C), (C) \leftarrow (A7)$	
RR A	$(An) \leftarrow (An+1), n=0 \sim 6; (A7) \leftarrow (A0)$	
RRC A	$(An) \leftarrow (An+1), (A7) \leftarrow (C), (C) \leftarrow (A0)$	
SWAP A	$(A3 \sim 0) \leftarrow \rightarrow (A7 \sim 4)$	



3.3.3 逻辑运算类指令

3. 逻辑与类指令

指令	操作说明	执行周期
ANL A,Rn	(A) \leftarrow (A) .AND. (Rn), n=0~7	1
ANL A,direct	(A) \leftarrow (A) .AND. (direct)	
ANL A,@Ri	(A) \leftarrow (A) .AND. ((Ri))	
ANL A,#data	(A) \leftarrow (A) .AND. #data	
ANL direct,A	(direct) \leftarrow (direct) .AND. (A)	
ANL direct,#data	(direct) \leftarrow (direct) .AND. #data	2



3.3.3 逻辑运算类指令

4. 逻辑或类指令

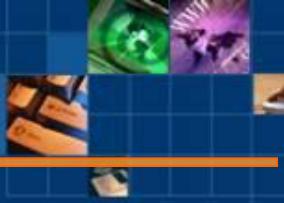
指令	操作说明	执行周期
ORL A,Rn	(A) \leftarrow (A) .OR. (Rn), n=0~7	1
ORL A,direct	(A) \leftarrow (A) .OR. (direct)	
ORL A,@Ri	(A) \leftarrow (A) .OR. ((Ri))	
ORL A,#data	(A) \leftarrow (A) .OR. #data	
ORL direct,A	(direct) \leftarrow (direct) .OR. (A)	
ORL direct,#data	(direct) \leftarrow (direct) .OR. #data	2



3.3.3 逻辑运算类指令

5. 逻辑异或类指令

指令	操作说明	执行周期
XRL A,Rn	(A) \leftarrow (A) .XOR. (Rn), n=0~7	1
XRL A,direct	(A) \leftarrow (A) .XOR. (direct)	
XRL A,@Ri	(A) \leftarrow (A) .XOR. ((Ri))	
XRL A,#data	(A) \leftarrow (A) .XOR. #data	
XRL direct,A	(direct) \leftarrow (direct) .XOR. (A)	
XRL direct,#data	(direct) \leftarrow (direct) .XOR. #data	2



3.3.4 控制转移类指令

- 控制转移类指令通过改变PC的值来实现程序的有条件、无条件地从当前位置转移到某个指定的位置去执行，从而实现程序流程(或执行方向)的改变；
- 控制转移类指令包括：无条件转移指令、条件转移指令、调用指令和返回指令；
- 调用指令和返回指令也属于无条件转移指令。



3.3.4 控制转移类指令

1. 无条件转移指令

1) 无条件跳转类指令

指令	操作说明	执行周期
LJMP addr16	$(PC) \leftarrow addr0 \sim 15$	
AJMP addr11	$(PC) \leftarrow (PC) + 2$ $(PC0 \sim 10) \leftarrow addr0 \sim 10$	2
SJMP rel	$(PC) \leftarrow (PC) + 2$ $(PC) \leftarrow (PC) + rel$	
JMP @A+DPTR	$(PC) \leftarrow (A) + (DPTR)$	



3.3.4 控制转移类指令

1. 无条件转移指令

2) 调用类指令

指令	操作说明	执行周期
LCALL addr16	$(PC) \leftarrow PC + 3; (SP) \leftarrow (SP) + 1;$ $((SP)) \leftarrow (PC0 \sim 7); (SP) \leftarrow (SP) + 1;$ $((SP)) \leftarrow (PC8 \sim 15);$ $(PC) \leftarrow addr0 \sim 15$	2
ACALL addr11	$(PC) \leftarrow PC + 2; (SP) \leftarrow (SP) + 1;$ $((SP)) \leftarrow (PC0 \sim 7); (SP) \leftarrow (SP) + 1;$ $((SP)) \leftarrow (PC8 \sim 15);$ $(PC0 \sim 10) \leftarrow addr0 \sim 10$	2



3.3.4 控制转移类指令

1. 无条件转移指令

3) 返回类指令及空操作

指令	操作说明	执行周期
RET	$(PC8 \sim 15) \leftarrow ((SP)); (SP) \leftarrow (SP)-1;$ $(PC0 \sim 7) \leftarrow ((SP)); ((SP)) \leftarrow (SP)-1$	2
RETI	$(PC8 \sim 15) \leftarrow ((SP)); (SP) \leftarrow (SP)-1;$ $(PC0 \sim 7) \leftarrow ((SP)); ((SP)) \leftarrow (SP)-1$	
NOP	$(PC) \leftarrow (PC) + 1$	1



3.3.4 控制转移类指令

2. 条件转移指令

1) 判0转移类指令

指令	操作说明	执行周期
JZ rel	$(PC) \leftarrow (PC) + 2$ if $(A) == 0 : (PC) \leftarrow (PC) + rel$	2
JNZ rel	$(PC) \leftarrow (PC) + 2$ if $(A) != 0 : (PC) \leftarrow (PC) + rel$	



3.3.4 控制转移类指令

2. 条件转移指令

2) 比较转移类指令

格式为 CJNE OP1, OP2, rel

指令	操作说明	执行周期
CJNE A,direct,rel	$(PC) \leftarrow (PC) + 3$ if $OP1 \neq OP2$ then	
CJNE A,#data,rel	$(PC) \leftarrow (PC) + rel$ if $OP1 < OP2$ then	2
CJNE Rn,#data,rel	$C \leftarrow 1$ else $C \leftarrow 0$	
CJNE @Ri,#data,rel		



3.3.4 控制转移类指令

2. 条件转移指令

3) 循环转移类指令

格式为 DJNZ OP, rel

指令	操作说明	执行周期
DJNZ Rn,rel	$(PC) \leftarrow (PC) + 2$ $(OP) \leftarrow (OP) - 1$ if $OP \neq 0$ then $(PC) \leftarrow (PC) + rel$	2
DJNZ direct,rel		



3.3.5 布尔(位)处理类指令

- 位操作是以位 (bit) 为对象来进行运算和操作的。由于位变量的值只能为1或0，故位操作又称为布尔操作；
- MCS-51 CPU内部有一个功能强大的布尔处理器，它实际上是个一位的CPU。它以进位标志位Cy作为布尔累加器，以单片机内部RAM的可位寻址区和特殊功能寄存器中可位寻址寄存器的位为操作数；
- MCS-51单片机的位寻址空间可分为两个部分：
 - 内部RAM中地址为20H~2FH的16个可位寻址字节，共有128个位地址，位地址范围从00H~7FH;
 - SFR区中可被位寻址的SFR（字节地址可被8整除）中的每个位；



3.3.5 布尔(位)处理类指令

1. 位传送指令

指令	操作说明	执行周期
MOV C,bit	(C) \leftarrow (bit)	1
MOV bit,C	(bit) \leftarrow (C)	



3.3.5 布尔(位)处理类指令

2. 位修改指令

指令	操作说明	执行周期
CLR C	$(C) \leftarrow 0$	1
CLR bit	$(bit) \leftarrow 0$	
SETB C	$(C) \leftarrow 1$	
SETB bit	$(bit) \leftarrow 1$	
CPL C	$(C) \leftarrow !(C)$	
CPL bit	$(bit) \leftarrow !(bit)$	



3.3.5 布尔(位)处理类指令

3. 位逻辑运算指令

指令	操作说明	执行周期
ANL C,bit	$(C) \leftarrow (C) .AND. (bit)$	2
ANL C,/bit	$(C) \leftarrow (C) .AND. (!bit)$	
ORL C,bit	$(C) \leftarrow (C) .OR. (bit)$	
ORL C,/bit	$(C) \leftarrow (C) .OR. (!bit)$	



3.3.5 布尔(位)处理类指令

4. 布尔(位)条件转移指令

指令	操作说明	执行周期
JC rel	$(PC) \leftarrow (PC)+2$ if $(C) == 1$ then $(PC) \leftarrow (PC)+rel$	
JNC rel	$(PC) \leftarrow (PC)+2$ if $(C) == 0$ then $(PC) \leftarrow (PC)+rel$	
JB bit,rel	$(PC) \leftarrow (PC)+3$ if $(bit) == 1$ then $(PC) \leftarrow (PC)+rel$	2
JBC bit,rel	$(PC) \leftarrow (PC)+3$ if $(bit) == 1$ then $(bit) \leftarrow 0; (PC) \leftarrow (PC)+rel$	
JNB bit,rel	$(PC) \leftarrow (PC)+3$ if $(bit) == 0$ then $(PC) \leftarrow (PC)+rel$	