

指令系统与指令格式

指令：指令是使计算机内部执行一定动作的一种控制命令。指令的集合构成指令系统。

程序：为完成某项特定任务的一组指令的集合。

计算机按程序一条一条地依次执行指令，从而完成指定任务。要让计算机完成各项任务，用户要设计各种程序。

指令格式：指令的表示方法称为指令格式。

- **操作码：**规定指令的操作功能
- **操作数：**指令操作的具体对象（地址、数据）

MCS-51的指令分类：

- | | |
|---------|----------|
| ■ 单字节指令 | ◆ 1个机器周期 |
| ■ 双字节指令 | ◆ 2个机器周期 |
| ■ 三字节指令 | ◆ 4个机器周期 |

指令系统的寻址方式

MCS-51单片机的指令系统共有111条指令，7种寻址方式。

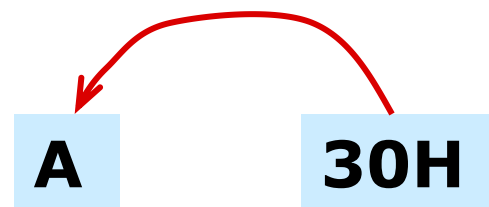
- 立即寻址方式
- 直接寻址方式
- 寄存器寻址方式
- 寄存器间接寻址方式
- 基址寄存器加变址寄存器间接寻址方式（变址寻址方式）
- 相对寻址方式
- 位寻址

指令系统的寻址方式

一、立即寻址

立即寻址是指在指令中直接给出其操作数，该操作数称为立即数。为了与直接寻址指令中的直接地址相区别，在立即数前面必需加上前缀“#”。

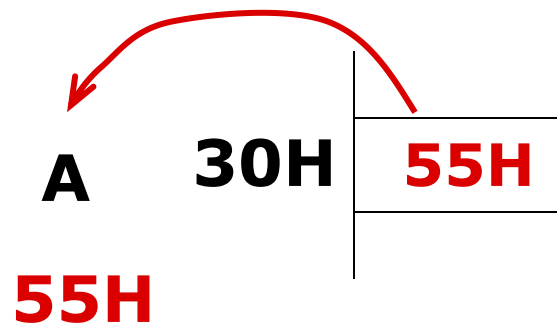
例如：MOV A, #30H



二、直接寻址

直接寻址是指在指令中直接给出存放数据的地址（注意：不是立即数,并且只限于片内RAM范围）。直接寻址只能访问特殊功能寄存器、内部数据存储器的低128个字节区域。

例如：MOV A, 30H



比较以上两指令的区别

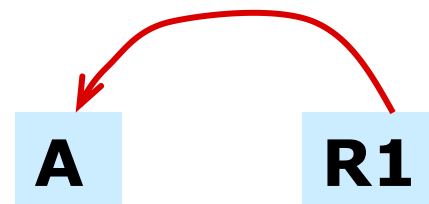
第3部分 MCS-51的指令系统

指令系统的寻址方式

三、寄存器寻址

寄存器寻址是指指令中的操作数为寄存器中的内容。

例如：MOV A, R1



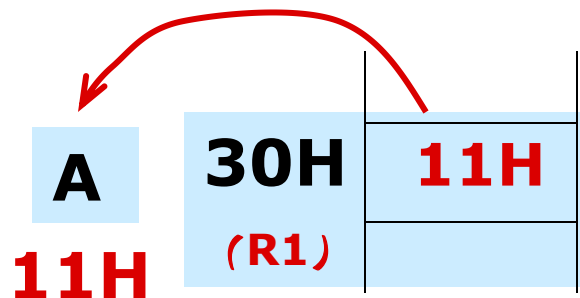
四、寄存器间接寻址

寄存器间接寻址是指指令中的操作数在寄存器的内容所指的地址单元中。

例如：MOV R1, #30H ; 把立即数30H送R1寄存器

MOV A, @R1 ; 把30H单元中的数送到A中

比较以上两指令的区别



指令系统的寻址方式

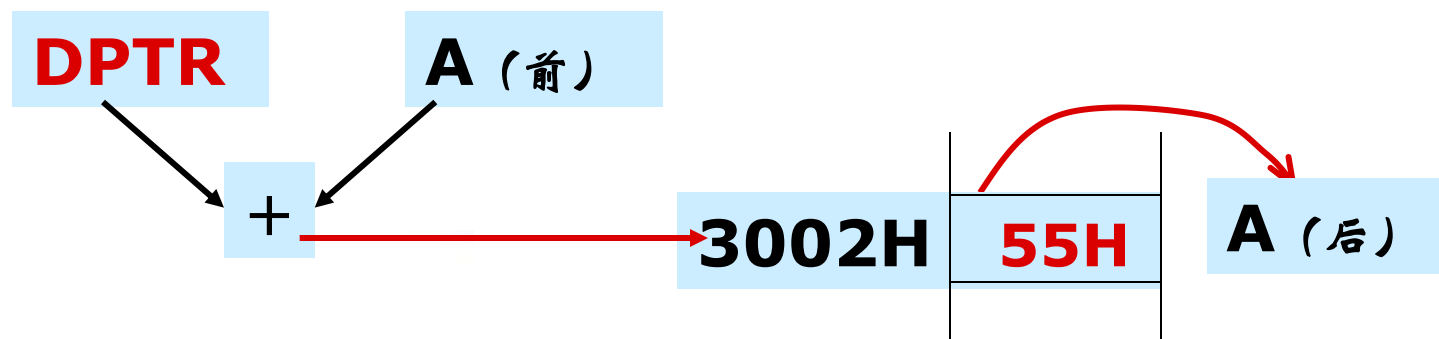
五、变址寻址

变址寻址用于访问程序存储器中的一个字节，该字节的地址是：基址寄存器（DPTR或PC）的内容与变址寄存器A中的内容之和。

例如： `MOV DPTR, #3000H` ； 立即数3000H送DPTR

`MOV A, #02H` ； 立即数02H送A

`MOVC A, @A+DPTR` ； 取ROM中3002H单元中的数送A



指令系统的寻址方式

六、相对寻址

以PC当前值为基准，加上相对偏移量rel形成转移地址

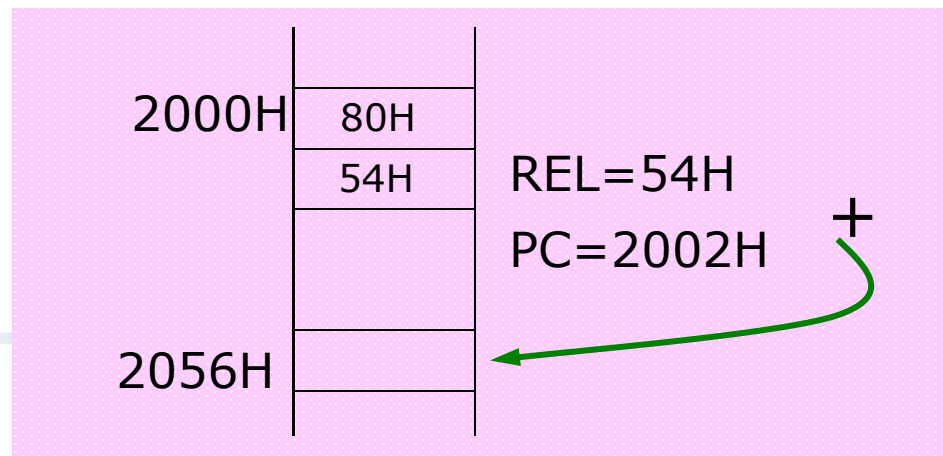
转移范围：以PC当前值起始地址，相对偏移在-128~+127字节单元之间。

相对寻址方式为相对转移指令所采用，转移的目的地址为：

$$\begin{aligned}\text{目的地址} &= \text{PC当前值起始地址} + \text{rel} \\ &= \text{转移指令所在地址} + \text{转移指令字节数} + \text{rel}\end{aligned}$$

例：2000H **SJMP LP**

2056H **LP:**



指令系统的寻址方式

七、位寻址

位寻址是指对片内RAM的位寻址区（20H~2FH）、可以位寻址的特殊功能寄存器（SFR）的各位，并进行位操作的寻址方式。例如：

MOV C, 00H	；把直接位地址00H位（20H单元中D0位）的值送C位。
MOV P1.3, C	；把C位中的值送P1口的D3位。
SETB 2AH.7	；把位寻址区2AH单元D7位（直接位地址为57H）置1。

位地址的表示方法

位名称 例：Cy、RS0

寄存器名加序号 例：ACC.1、P0.3

字节地址加序号 例：20H.3

直接位地址 例：07H、65H

第3部分 MCS-51的指令系统

操作数的7种寻址方式和寻址的空间

寻址方式	相关寄存器	源操作数寻址的空间
立即寻址		程序存储器中（立即数存储于ROM中）
直接寻址		片内低128字节RAM和SFR
寄存器寻址	R0~R7, A, B、 DPTR	R0~R7, A, B, DPTR
寄存器间接寻址	@R0, @R1	片内RAM
	@R0, @R1, @DPTR	片外数据存储器
变址寻址	@A+PC, @A+DPTR	程序存储器
相对寻址	PC	程序存储器
位寻址	可位寻址的SFR	片内RAM20H~2FH、SFR中可寻址位

MCS-51指令系统介绍

MCS-51单片机的指令系统共分为五大类：数据传送类指令，算术运算类指令，逻辑运算类指令，控制转移类指令，位操作类指令。

指令格式如下：

操作码 [操作数1] [, 操作数2] [, 操作数3]

- 方括号“[]”内的字段表示可有可无。
- 操作码表示指令进行何种操作，即操作性质。一般为英语单词缩写。
- 操作数指出了参加操作的数据或数据存放的地址，即操作对象。它以一个或几个空格与操作码隔开；根据指令功能的不同，操作数可以有3个、2个、1个或没有，操作数之间以逗号“,”分开。



MCS-51指令系统介绍

MCS-51指令中，常用的符号：

Rn: 当前寄存器组的8个工作寄存器R0~R7。

Ri: 可用作间接寻址的工作寄存器，只能是R0、R1。

direct: 直接地址，即8位的内部数据存储器或者特殊功能寄存器的地址。

#data、#data16: 分别表示8位、16位立即数。

rel: 相对转移指令中的偏移量，为8位带符号数（补码形式）。

addr11、addr16: 分别表示11位、16位地址码。

bit: 片内RAM中（可位寻址）的位地址。

MCS-51指令系统介绍

MCS-51指令中，常用的符号：

DPTR： 数据指针，用作16位的地址寄存器。

A： 累加器A；ACC则表示累加器A的地址。

C或Cy： 进位标志位或位处理机中的累加器。

@： 间接寻址的前缀标志。

\$： 当前指令存放的地址。

(X)： X中的内容。

((X))： 由X间接寻址的单元中的内容。

→： 箭头右边的内容被箭头左边的内容所取代。

MCS-51指令系统介绍

一、数据传送类指令

特点：除第一操作数为A的指令影响标志P位之外，并不影响其他标志位。

1. 以累加器为目的操作数的指令

MOV A, Rn ; (Rn)→A, n=0~7

MOV A, @Ri ; ((Ri))→A, i=0,1

MOV A, direct ; (direct)→A

MOV A, #data ; #data→A

MCS-51指令系统介绍

2. 以 Rn 为目的操作数的指令

MOV Rn, A	; (A)→Rn, n=0~7
MOV Rn, direct	; (direct)→Rn, n=0~7
MOV Rn, #data	; #data→Rn, n=0~7

3. 以直接地址为目的操作数的指令

MOV direct, A	; (A)→direct
MOV direct, Rn	; (Rn)→direct, n=0~7
MOV direct1, direct2	; (direct2)→direct1
MOV direct, @Ri	; ((Ri))→direct, i=0,1
MOV direct, #data	; #data→direct

MCS-51指令系统介绍

4. 以寄存器间接地址为目的的操作数的指令

MOV @Ri, A ; (A)→(Ri), i=0,1

MOV @Ri, direct ; (direct)→(Ri), i=0,1

MOV @Ri, #data ; #data→(Ri), i=0,1

5. 16位数据传送指令

MOV DPTR, #data16 ; #data16→DPTR

例如: **MOV 30H, #7AH** ; 将立即数7AH送片内RAM 30H单元中

MOV R0, #30H ; 将立即数30H送R0寄存器

MOV A, @R0 ; 将R0指定的30H中的数7AH送A中

MOV DPTR, #1000H ; 将1000H送DPTR寄存器



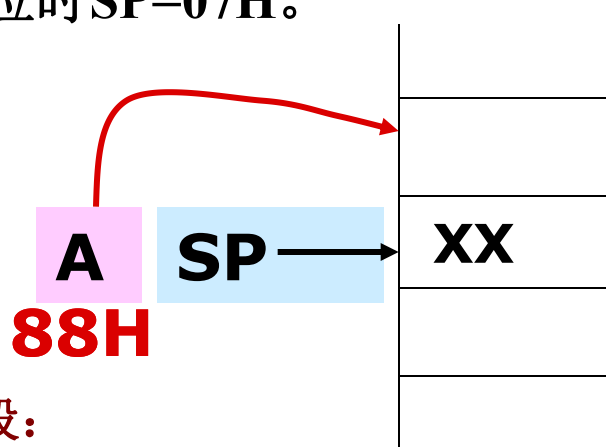
MCS-51指令系统介绍

6. 堆栈操作指令

堆栈是在内RAM开辟的一个数据的暂存空间，遵守“后进先出”原则操作，其地址指针为SP，它指出栈顶的位置，复位时SP=07H。

入栈: **PUSH direct** ; SP先加1，再将数据压栈。

出栈: **POP direct** ; 数据先出栈，再SP减1。



例如，已知 (A) = 44H, (30H) = 55H, 执行以下程序段：

```
MOV    SP, #5FH    ; 栈起点设置为5FH
PUSH   ACC          ; A中的44H压到60H中保存
PUSH   30H          ; 30H中的55H压到61H中保存
POP    30H          ; 把61H中的55H弹出到30H
POP    ACC          ; 把60H中的44H弹出到A中
```

PUSH ACC

MCS-51指令系统介绍

指令运用举例：

例：将片内RAM的 30H单元与 40H单元中的内容互换。

方法1（直接地址传送法）：

```
MOV  31H, 30H
MOV  30H, 40H
MOV  40H, 31H
SJMP  $
```

方法2（间接地址传送法）：

```
MOV  R0, #40H
MOV  R1, #30H
MOV  A, @R0
MOV  B, @R1
MOV  @R1, A
MOV  @R0, B
SJMP  $
```


MCS-51指令系统介绍

指令运用举例：

例：将片内RAM的 30H单元与 40H单元中的内容互换。

方法3（字节交换传送法）：

```
MOV  A, 30H
XCH  A, 40H
MOV  30H, A
SJMP $
```

方法4（堆栈传送法）：

```
PUSH  30H
PUSH  40H
POP   30H
POP   40H
SJMP  $
```

MCS-51指令系统介绍

7. 累加器A与外部数据存储器传送指令

采用DPTR间接寻址；外部数据存储器的地址为16位（寻址空间64Kbyte）。采用Ri间接寻址，外部数据存储器的地址为8位（寻址空间256byte）。

(1) 64KB外部RAM单元与A之间的传送

MOVX A, @DPTR ; ((DPTR))→A, 读外部RAM或I/O口

MOVX @DPTR, A ; (A)→(DPTR), 写外部RAM或I/O口

(2) 外部RAM低256字节单元与A之间的传送

MOVX A, @Ri ; ((Ri))→A, 读外部RAM或I/O口

MOVX @Ri, A ; (A)→(Ri), 写外部RAM或I/O口



MCS-51指令系统介绍

例: **MOV R0, #80H**

MOVX A, @R0 ; 将外部RAM的80H单元内容→A

例: **MOV DPTR, #2000H**

MOVX A, @DPTR ; 将外部RAM中2000H单元内容→A

思考: 将片内RAM
50H单元内容 → 片
外RAM 5000H单元

MCS-51指令系统介绍

8. 查表指令

查表指令是单向指令，用于读取程序存储器中的数据表格。

MOVC A, @A+DPTR ; ((A)+(DPTR))→A

MOVC A, @A+PC ; ((A)+(PC))→A

; 以PC的当前值为基址，A为变址

例1：在程序存储器的ROM中从1000H开始存有5个字节数，编程将第2个字节数取出送片内RAM 30H单元中。程序段如下：

MOV DPTR, #1000H ; 置ROM地址指针（基址）DPTR

MOV A, #01H ; 表内序号送A（变址）

MOVC A, @A+DPTR ; 从ROM 1000H单元中取数送到A

MOV 30H, A ; 再存入内RAM 30H中

ORG 1000H ; 伪指令，定义数表起始地址

TAB: DB 55H, 67H, 9AH, ... ; 在ROM 1000H开始的空间中定义5个单字节数

MCS-51指令系统介绍

例2：设某数N已存于20H单元（ $N \leq 10$ ），查表求N平方值，存入21H单元。

MOV	A, 20H	; 取数N
ADD	A, #02H	; 增加TAB的偏移量
MOVC	A, @A+PC	; 查表
SJMP	Store	
TAB:	DB 00H, 01H, 04H, 09H,	; 定义数表
Store:	MOV 21H, A	; 结果存入21H单元

由于PC为程序计数器，总是指向下一条指令的地址，在执行第三条指令“**MOVC A, @A+PC**”前，应在A累加器中加上查表偏移量。

用DPTR查表时，表格可以放在ROM的64K范围；用**MOVC A, @A+PC**指令时则必须把表格就放在该条指令下面开始的255个字节的空间中。

MCS-51指令系统介绍

9. 字节/半字节交换指令指令

有3条XCH指令为整个字节相互交换，XCHD指令为低4位相互交换，SWAP指令为ACC中的高、低4位互换。

(1) 字节交换

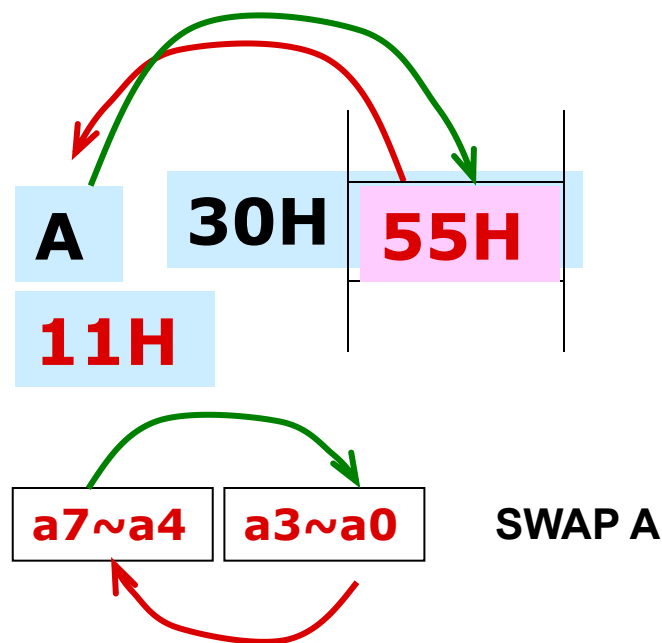
XCH A, Rn ; (A) \longleftrightarrow (Rn)
XCH A, direct ; (A) \longleftrightarrow (direct)
XCH A, @Ri ; (A) \longleftrightarrow ((Ri))

(2) 半字节交换

XCHD A, @Ri
SWAP A

例如：已知 (A)=34H, (R6)=29H,
执行以下指令后, (A)=?

XCH A, R6;
SWAP A



思考：用不同的方法实现片内RAM (20H) \rightarrow R1

MCS-51指令系统介绍

二、算术操作类指令

算术运算类指令有加、减、乘、除法指令、增1和减1指令、十进制调整指令，共24条。使用时应注意判断各种结果对哪些标志位（Cy、OV、Ac、P）产生影响。

1. 不带进位的加法指令 ----- 指令助记符为 **ADD**

ADD A, Rn	; (A)+(Rn) → A
ADD A, direct	; (A)+(direct) → A
ADD A, @Ri	; (A)+((Ri)) → A
ADD A, #data	; (A)+#data → A

MCS-51指令系统介绍

对标志位的影响:

- 如果位7有进位，则进位标志位Cy置“1”，否则Cy清“0”；
- 如果位3有进位，则辅助进位标志位Ac置“1”，否则Ac清“0”；
- 如果位6有进位而位7没有进位，或者位6没有进位而位7有进位，则溢出标志位OV置“1”，否则清“0”OV。

例：(A) =53H, (R0) =0FCH, 执行指令 ADD A, R0。

$$\begin{array}{r} 0101\ 0011 \\ + \quad 1111\ 1100 \\ \hline \end{array}$$

1← 0100 1111

标志位结果为：(A) =4FH, Cy=1, Ac=0, OV=0, P=1 (偶校验)。

MCS-51指令系统介绍

2. 带进位的加法指令 ----- 指令助记符为 **ADDC**

ADDC 比 **ADD** 多了加 **Cy** 位的值（之前指令留下的Cy值），主要用于多字节的加法运算，结果也送A。影响Ac、Cy、OV、P位。

ADDC	A, Rn	;	$(A) + (Rn) + (Cy) \rightarrow A$
ADDC	A, direct	;	$(A) + (\text{direct}) + (Cy) \rightarrow A$
ADDC	A, @Ri	;	$(A) + ((Ri)) + (Cy) \rightarrow A$
ADDC	A, #data	;	$(A) + \#data + (Cy) \rightarrow A$

对标志位的影响和**ADD**指令一样。

MCS-51指令系统介绍

3. 增1指令 ----- 指令助记符为 **INC**

指令的功能是将操作数中的内容加1。除对A操作影响P外不影响任何标志。

INC A ; $(A)+1 \rightarrow A$ 以下类同

INC Rn

INC direct

INC @Ri

INC DPTR

若原来的内容为FFH，则加1后变为00H。

MCS-51指令系统介绍

4. 带借位的减法指令 ----- 指令助记符为 **SUBB**

指令的功能是第一操作数A的内容减去第二操作数的内容，再减去Cy值，然后把结果存入A中，同时产生新的Cy、Ac、OV、P位的值。

SUBB A, Rn	； $(A) - (Rn) - (Cy) \rightarrow A$
SUBB A, direct	； $(A) - (\text{direct}) - (Cy) \rightarrow A$
SUBB A, @Ri	； $(A) - ((Ri)) - (Cy) \rightarrow A$
SUBB A, #data	； $(A) - \#data - (Cy) \rightarrow A$

注意：没有不带借位的减法指令。
单字节减法如何实现？

CLR C ； 将前次Cy清零
SUBB A, 30H

MCS-51指令系统介绍

对标志位的影响:

- 如果位7需借位, 则标志位Cy置“1”, 否则Cy清“0”;
- 如果位3需借位, 则标志位Ac置“1”, 否则Ac清“0”;
- 如果位6需借位而位7不需要借位, 或者位6不需借位而位7需借位, 则溢出标志位OV置“1”, 否则清“0”OV。

例: (A) = 0C9H, (R2) = 54H, Cy=1, 执行指令 SUBB A, R2。

$$\begin{array}{r} 1100\ 1001 \\ 0101\ 0100 \\ - \qquad \qquad 1 \\ \hline 0111\ 0100 \end{array}$$

标志位结果为: (A) = 74H, Cy=0, Ac=0, OV=1, P=0。

MCS-51指令系统介绍

5. 减1指令 ----- 指令助记符为 **DEC**

指令的功能是将操作数中的内容减1。除对A操作影响P外不影响任何标志。

DEC A ; (A)-1→A 以下类同

DEC Rn

DEC direct

DEC @Ri

注意：没有对DPTR的减1操作指令

若原来的内容为00H，则减1后变为FFH。

6. 乘法指令

MUL AB ; $(A) \times (B) \rightarrow BA$

说明：当积大于255（0FFH）时，即积的高字节B不为0时，置OV=1，否则OV=0；Cy位总是0，运算结果影响P标志位。

7. 除法指令

DIV AB ; $(A)/(B)$: 商 $\rightarrow A$, 余数 $\rightarrow B$

说明：无符号数相除，当除数（B）=0时，结果为无意义，并置OV=1；Cy位总是0，运算结果影响P标志位。

作业3-1

一、MC5-51共有哪几种寻址方式？各有什么特点？

二、假定累加器A中的内容为10H，执行指令

3200H: MOVC A, @A+PC

后，把程序存储器（ ）单元的内容送入累加器A中。

三、编写一段程序，将内部RAM的30H单元的内容传送到外部RAM的2000H单元中。

四、编写一段程序，将内部RAM的20H单元内容与累加器A内容相加，结果存放到20H单元。