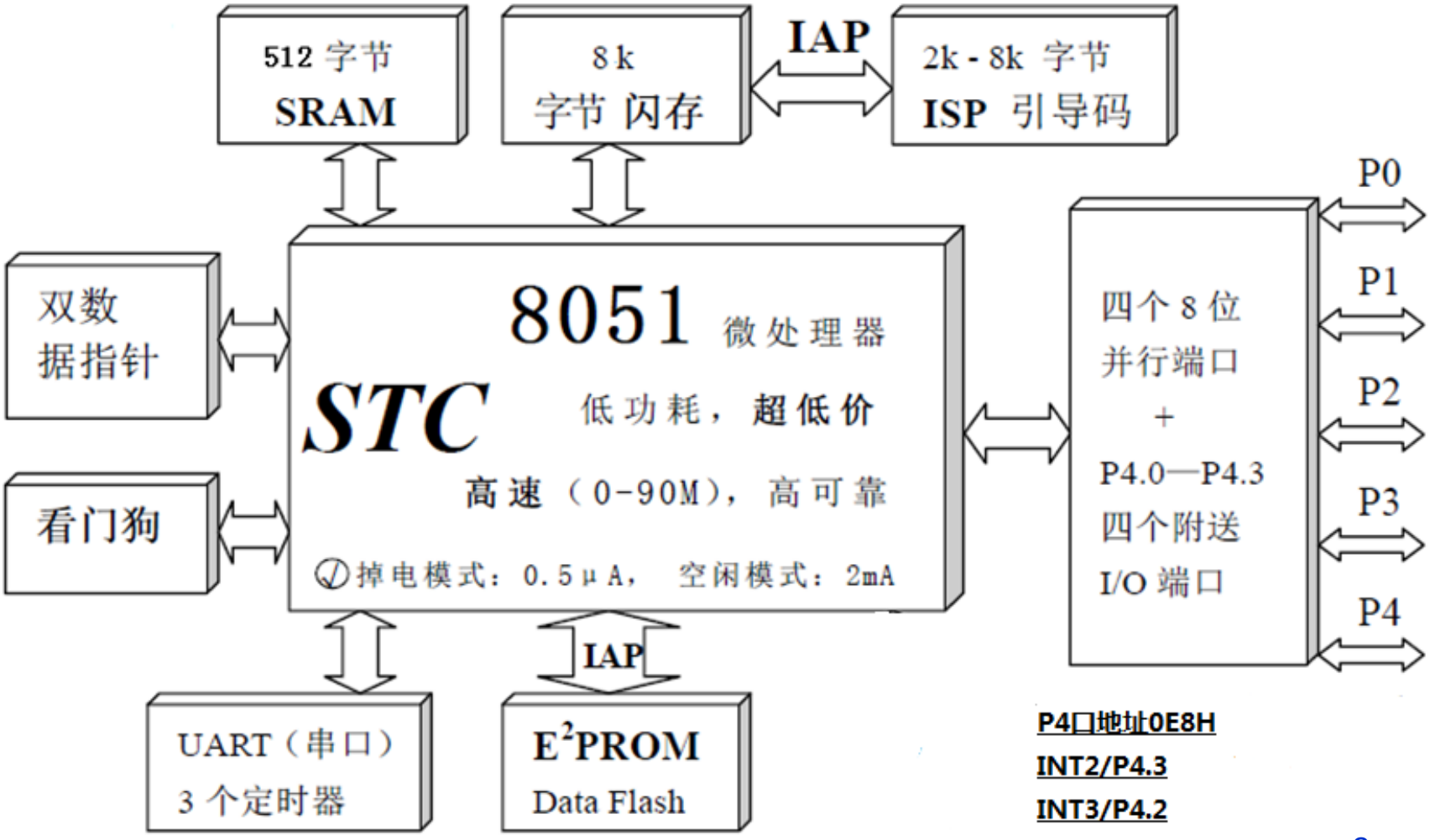
第三章 PART1-STC89C52指令系统

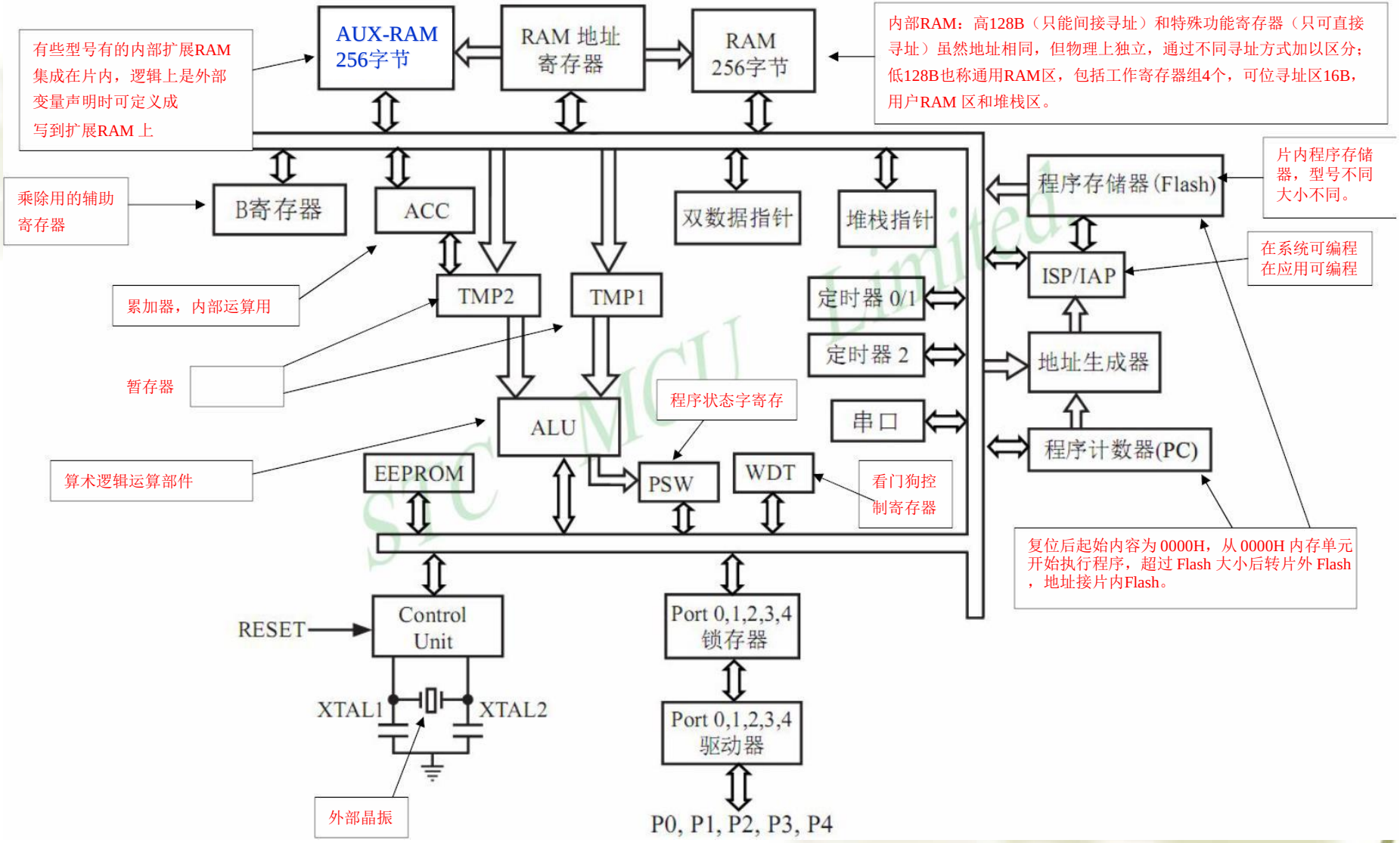
3.1 指令系统基本概念

3.2 STC89C52指令系统分类介绍

STC89C52单片机的硬件结构



P4口地址0E8H
INT2/P4.3
INT3/P4.2
P4口均可位寻址


有些型号有的内部扩展RAM集成在片内，逻辑上是外部变量声明时可定义成写到扩展RAM 上
AUX-RAM 256字节
RAM 地址寄存器
RAM 256字节
内部RAM：高128B（只能间接寻址）和特殊功能寄存器（只可直接寻址）虽然地址相同，但物理上独立，通过不同寻址方式加以区分；低128B也称通用RAM区，包括工作寄存器组4个，可位寻址区16B，用户RAM 区和堆栈区。
乘除用的辅助寄存器
B寄存器
ACC
双数据指针
堆栈指针
程序存储器(Flash)
片内程序存储器，型号不同大小不同。
累加器，内部运算用
TMP2
TMP1
定时器 0/1
ISP/IAP
在系统可编程
在应用可编程
暂存器
定时器 2
地址生成器
程序状态字寄存
ALU
串口
算术逻辑运算部件
EEPROM
PSW
WDT
程序计数器(PC)
看门狗控制寄存器
复位后起始内容为 0000H，从 0000H 内存单元开始执行程序，超过 Flash 大小后转片外 Flash，地址接片内Flash。
RESET
Control Unit
Port 0,1,2,3,4 锁存器
XTAL1
XTAL2
Port 0,1,2,3,4 驱动器
外部晶振
P0, P1, P2, P3, P4



内部RAM：高128B（只能间接寻址）和特殊功能寄存器（只可直接寻址）虽然地址相同，但物理上独立，通过不同寻址方式加以区分；低128B也称通用RAM区，包括工作寄存器组4个，可位寻址区16B，用户RAM 区和堆栈区。

第三章 PART1-STC89C52指令系统

3.1 指令系统基本概念

3.2 STC89C52指令系统分类介绍

3.1 指令系统基本概念

- STC89C52单片机所能执行的指令的集合称为指令系统。
- 指令的表示方法称为指令格式，一条指令通常由两部分组成，即操作码和地址码,地址码也被称为操作数。
- 操作数并不一定是一个具体的数据，也有可能仅指出到哪里取得数据的地址或符号。



- **STC89C52的基本指令共111条，下面是指令系统按空间和时间上的划分：**

按指令在程序存储器所占字节数		按指令的执行时间	
单字节指令 49 条	操作码、操作数同在一个字节	1 个机器周期	64 条
双字节指令 45 条	操作码、操作数各占一个字节	2 个机器周期	45 条
三字节指令 17 条	操作码占 1B，操作数占 2B	4 个机器周期	2 条（乘、除）

从上表我们可以看出，绝大多数指令占用单字节或双字节，并且都能在**1--2个机器周期**内完成。由此可见，**STC89C52**指令系统对存储空间和时间的利用率较高。

符号注释

在描述STC89C52指令系统的功能时，我们经常使用下面符号，其意义如下：

Rn（n＝0-7）

当前选中的寄存器区的8个工作寄存器R0--R7

Ri（i=0、1）

当前选中的寄存器区中可作地址寄存器的2个寄存器R0、R1。

Direct

直接地址，即8位的内部数据存储器单元或特殊功能寄存器的地址。

#data 指令中的8位常数

#data16 指令中的16位常数

Rel

8位的有符号偏移量。用于SJMP或所有的条件转移指令中。偏移量按相对于下一条指令的第一个字节地址与跳转后指令的第一个字节地址之差计算，在－128~＋127范围内取值。

DPTR

数据指针，可用作16位的地址寄存器

Bit

内部RAM或特殊功能寄存器中的直接寻址位地址

Cy

进位标志或进位位，也可作为位处理机中的1位累加器。

@

间址寄存器或基址寄存器的前缀。如@Ri，@A+DPTR

(X)　　X中的内容

((X))　由X寻址的单元中的内容

/

位操作数的前缀，表示对该位操作数取反，如/bit。

寻址方式

所谓寻址就是寻找确定参与操作的数的真正地址。一般来说，寻址方式越多则计算机的功能就越强，灵活性亦越大，指令系统也就越复杂。

STC89C52单片机的指令系统共有**7**种寻址方式。

寻址方式（分类）

1、 立即寻址

2、 直接寻址

3、 寄存器寻址

4、 寄存器间接寻址

5、 变址寻址

6、 相对寻址

7、 位寻址

1、立即寻址

指令中直接给出操作数。立即操作数用前面加有#号的8位或16位数来表示。 8位立即操作数占一个字节，16位立即操作数占二个字节。

例如：

```
MOV     A，# 60H    ；A←#60H
MOV     DPTR，# 3400H   DPTR←#3400H
MOV     30H，# 40H；(30H)←#40H
```

指令代码分别为：

74H 60H

90H 34H 00H

75H 30H 40H

2、直接寻址

指令直接给出操作数地址。占一个字节。可访问两种地址空间：

✓特殊功能寄存器地址空间

唯一能访问该空间的寻址方式。

✓片内内部RAM（128个字节：00H～7FH）

指令形如　　MOV　A，direct
指令代码：　11100101　direct

例：MOV　A，40H
指令代码　11100101 01000000
即为：E5H 40H

对于特殊功能寄存器，在助记符指令中可直接用符号来代替地址，如：

MOV　A ,P0

表示将**P0**口的内容传送给**A**。

思考:**MOV 50H, 40H** 占多少字节?

3、寄存器寻址

寻址的对象为选定寄存器区的8个工作寄存器R0--R7，寄存器A、B、DPTR、和Cy位。这种寻址方式中，被寻址的寄存器的内容就是操作数,一般和操作码共占一个字节.

操作码的低3位指示所用的寄存器

例：单字节指令： MOV A, R_n (n=0~7)

指令代码：1110 1iii （iii=000～111）

即为：E8H-EFH

4、寄存器间接寻址

以寄存器中内容为地址，以该地址中内容为操作数的寻址方式。能用于寄存器间接寻址的寄存器有R0，R1，DPTR，SP。其中R0、R1必须是工作寄存器组中的寄存器。SP仅用于堆栈操作。

访问内部RAM或外部数据存储器的低256个字节时，可采用 R 0 或R 1 作为间址寄存器，通用形式为@Ri。

最低位是0还是1区分现用寄存器为R0还是R1。

MOV　A ,@Ri　(i=0、1)

例：MOV A, @R0 , 指令代码：1110 0110。

最低位是0还是1区分现用寄存器为R0还是R1。

MOVX A ,@Ri （i=0、1）

例：MOVX A,@R1 , 指令代码 :1110 0011。

例：假设该条指令存放在2030H,工作寄存器为第0组，(R1)=4FH。片外RAM (4FH)=30H。指令执行过程：



访问片外数据存储器还可用数据指针DPTR作为间址寄存器，可对整个外部数据存储器空间（64ＫB）寻址。

例： MOV DPTR ,#1234H

MOVX A ,@DPTR

执行PUSH和POP指令时，使用堆栈指针SP作间址寄存器来进行对栈区的间接寻址。

5. 基址寄存器加变址寄存器间址寻址

变址寻址只能对程序存储器中数据进行操作。由于程序存储器是只读的，因此变址寻址只有读操作而无写操作，此种寻址方式对查表访问特别有用。本寻址方式的指令只有三条：

MOVC A，@A+DPTR ；A←（A+DPTR）

MOVC A，@ A+PC ；A←（A+PC）

JMP @A+DPTR 无条件转移指令

基址寄存器：**DPTR或PC**

变址寄存器：累加器A

有效地址=（PC）+（A）

或有效地址=（DPTR）+（A）

6. 相对寻址

为解决程序转移而专门设置的。它以PC的当前值为基准，加上指令中给出的相对偏移量（rel）形成有效转移地址。偏移量占一个字节,但这里的PC的当前值是指执行完该指令后的PC值，即转移指令的PC值加上它的字节数。转移的目的地址用公式表示：

目的地址=转移指令所在地址+转移指令的字节数+rel

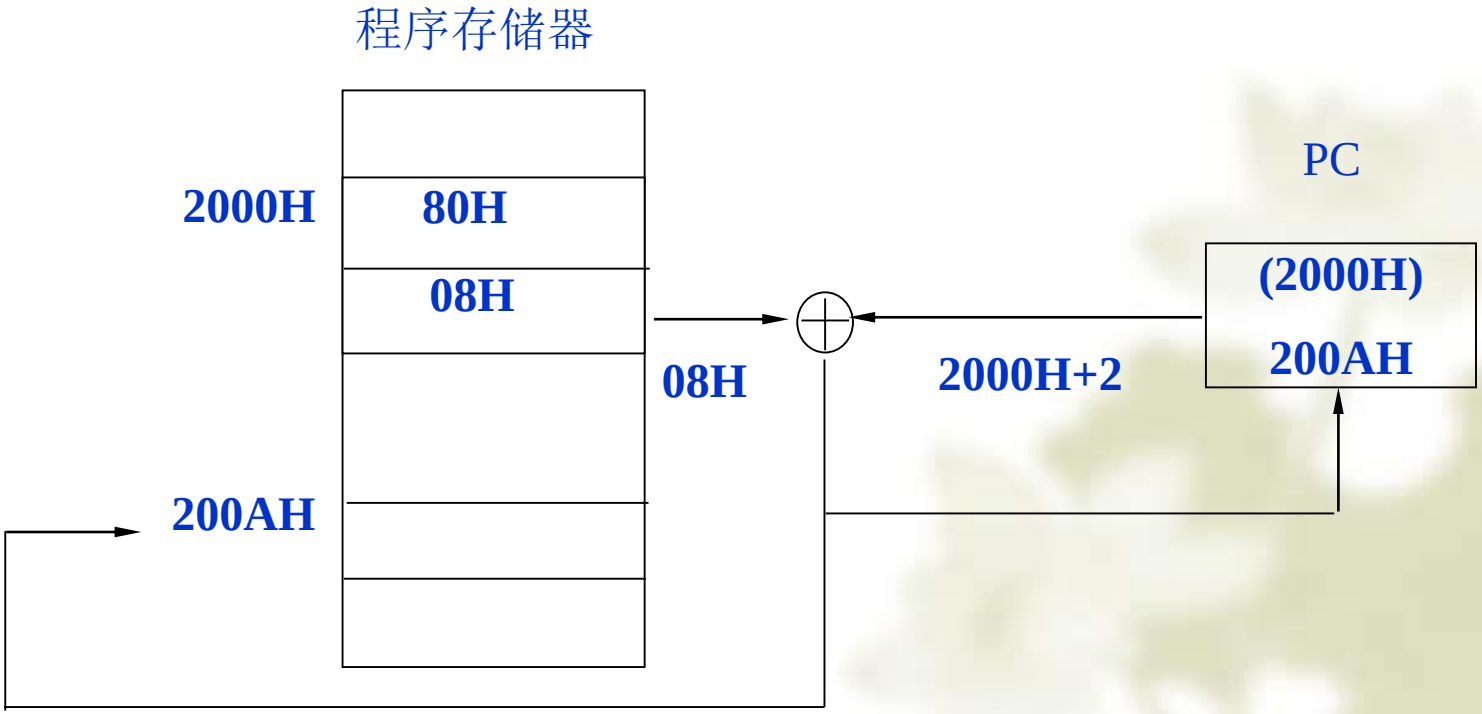
rel：1个有符号的8位二进制补码.

表示的范围是：-128 至 +127。

例如：SJMP 08H ；PC←PC+2+08

(指令的地址)=2000H

指令操作示意图如下图所示。



7. 位寻址

STC89C52有位处理的功能，可以对数据位进行操作，因此有相应的位寻址的方式。除C外,位寻址指令直接使用位地址，占一个字节。

例如：

MOV C, 40H ;把位40H的值送到进位位C。

位寻址的寻址范围包括两部分：一部分连续地分布在内部RAM20H-2FH中；另一部分布在可位寻址的特殊功能寄存器SFR中。

这些寻址位在指令中的表示方法不尽相同，有以下四种（以PSW位5为例）：

- ✓ 直接使用位寻址。 0D5H
- ✓ 位名称的表示方法。 F0标志位
- ✓ 单元地址加位数的表示方法。（0D0H）.5
- ✓ 特殊功能寄存器符号加位数的表示方法。PSW.5

序号	寻址方式	寻址空间
1	立即数寻址	程序存储器中的立即数
2	直接寻址	内部128字节RAM、特殊功能寄存器
3	寄存器寻址	R0～R7、A、B、C(位)、DPTR等
4	寄存器间接寻址	片内数据存储器、片外数据存储器
5	基址加变址寻址	读程序存储器固定数据和程序散转
6	相对寻址	程序存储器相对转移
7	位寻址	内部RAM中的可寻址位、SFR中的可寻址位

3.2 STC89C52指令系统分类介绍

数据传送类指令

算术运算类指令

逻辑运算与循环类指令

程序转移类指令

调用子程序及返回指令

位操作指令

空操作指令

数据传送类指令

------使用最频繁的指令

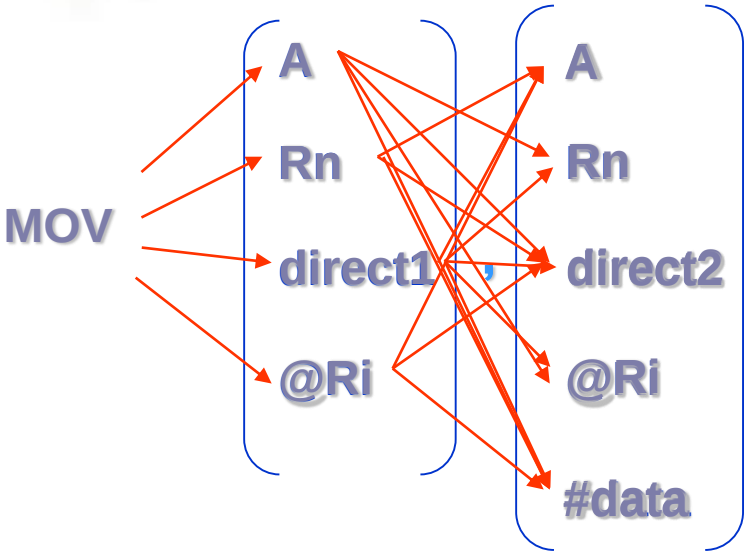
数据传送类指令共28条，是将源操作数送到目的操作数。指令执行后，源操作数不变，目的操作数被源操作数取代。数据传送类指令用到的助记符有**MOV、MOVX、MOVC、XCH、XCHD、SWAP、PUSH、POP** 8种。

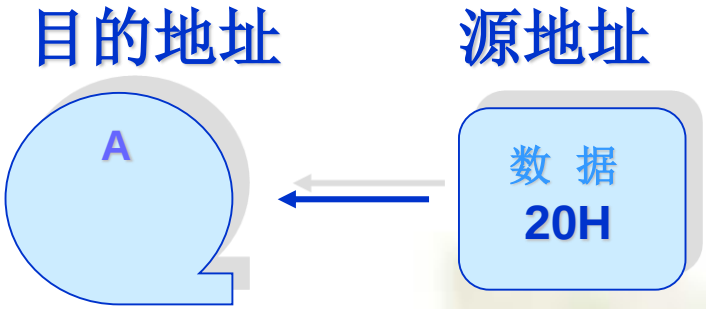
数据传送类指令不影响标志，这里所指的标志是指Cy、Ac和OV，但不包括检验累加器奇偶性的标志位P。

数据传送指令（28条）　使用最频繁的指令

——8位数据传送指令(15条)

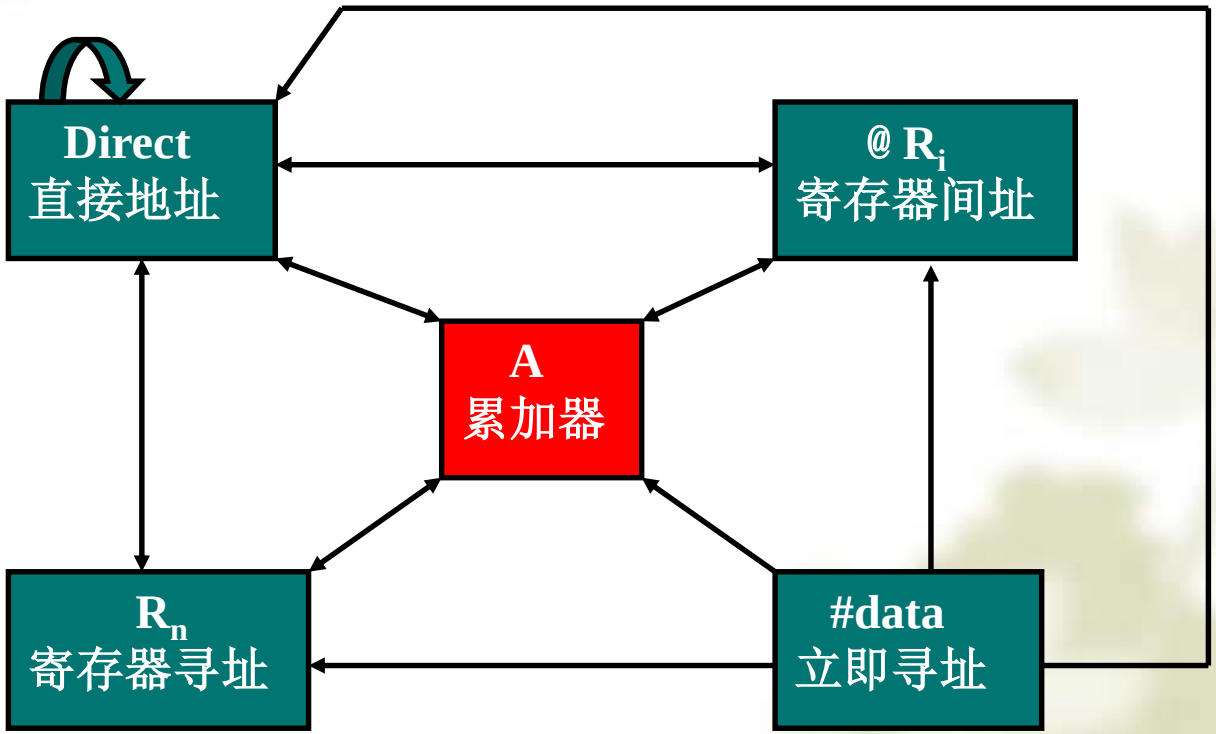
MOV　<目的操作数>,<源操作数>





MOV A，#20H

STC89C52单片机片内数据传送途径



以累加器为目的操作数的指令

MOV A，Rn ；A← Rn

MOV A，direct ；A←（direct）

MOV A，@Ri ；A←（Ri）

MOV A，#data ；A← #data

这组指令的功能是把源操作数的内容送入累加器A。

```
MOV  A,R6    ;(R6)→A
MOV  A,@R0   ;（（R0））→A
MOV  A,70H   ;（70H）→A
MOV  A,#78H  ;78H→A
```

寄存器寻址

间接寻址

直接寻址

立即寻址

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

这组指令的功能:把源操作数的内容送入当前工作寄存器区的**R0--R7**中的某一个寄存器.

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

这组指令的功能:把源操作数送入直接地址指出的存储单元。

以寄存器间接地址为目的操作数的指令

MOV @Ri,A ;(A) →((Ri)),i=0,1

MOV @Ri,direct ;direct→((Ri)),i=0,1

MOV @Ri,#data ;data→((Ri)),i=0,1

指令的功能:把源操作数内容送入R0或R1指出的存储单元。

例如：设（30H）=6FH，（R1）=40H，执行

MOV @R1，30H

30H单元中数据取出送入R1间接寻址的40H单元，（40H）=6FH。

16位数据传送指令 （1条）

MOV DPTR，#data16

指令功能:是把16位常数送入DPTR。这是整个指令系统唯一的一条16位数据传送指令，用来设置地址指针。地址指针DPTR由DPH和DPL组成。这条指令的执行结果把高8位立即数送入DPH,低8位立即数送入DPL。

堆栈操作指令

所谓堆栈是在片内RAM中按“先进后出”原则设置的专用存储区。数据的进栈出栈由指针SP统一管理。在指令系统中有两条用于数据操作的栈操作指令。

PUSH direct；SP←(SP+1)，(SP)←(direct)

指令功能：首先将栈指针SP加1，然后把直接地址指出的内容送到栈指针SP指示的内部RAM单元中。

POP direct；(direct)←(SP)，SP←SP-1

指令功能：栈指针SP指示的内部RAM单元内容送入直接地址指出的字节单元中，栈指针SP减1。

【例】将片内RAM 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   $
```

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    $
```

累加器A与外部数据存储器传送指令

指令功能:累加器A与外部RAM存储器或I/O的数据相互传送。

MOVX A,@DPTR ;((DPTR))→A, 读外部RAM/IO

MOVX @DPTR,A ;(A)→(DPTR)，写外部RAM/IO

采用16位的DPTR作间接寻址，可寻址整个64K片外数据存储器空间。高8位地址（DPH）由P2口输出，低8位地址DPL由P0口输出。

MOVX A,@Ri ;((Ri))→A， 读外部RAM/IO

MOVX @Ri,A ;(A)→((Ri))，写外部RAM/IO

采用Ri作间接寻址，可寻址片外256B的数据存储器。8位地址和数据均由P0口输出，可选用其他任何输出口线来输出高于8位的地址（一般选用P2口输出高8位的地址）。

查表指令

- **MOVC A,@A+PC**

这条指令以PC作基址寄存器，A的内容作为无符号整数和PC的内容相加后得到一个16位的地址，由该地址指出的程序存储单元的内容送达累加器A。

此指令的优点是不改变特殊功能寄存器及PC的状态，根据A的内容就可以取出表格中的常数。缺点是表格只能存放在该条查表指令后面的256个单元之内，表格的大小受到限制，而且表格只能被一段程序所利用。

MOVC A,@A+DPTR

这条指令以DPTR作为基址寄存器，A的内容作为无符号整数和DPTR的内容相加后得到一个16位的地址，由该地址指出的程序存储单元的内容送达累加器A。

例：（DPTR）=8100H （A)=40H 执行指令

MOVC A,@A+DPTR

结果将程序存储器中的8140H单元内容送入累加器A。

字节交换指令

XCH A,Rn ;(A)←→(Rn),n＝0～7

XCH A,direct ; (A)←→(direct)

XCH A,@Ri ;(A)←→((Ri)), i=0,1

指令功能:将累加器A的内容和源操作数的内容相互交换。

半字节交换指令

XCHD A，@Ri

指令功能：累加器的低4位与内部RAM低4位交换。

例：（R0）=60H,(60H)=3EH,(A)=59H

执行完XCHD A，@R0指令

则(A)=5EH，(60H)=39H。

累加器半字节交换指令

SWAP A

指令功能：累加器A的高半字节和低半字节互换。

例：（A）=0C5H,执行指令

SWAP A

结果：（A）=5CH

3.2 STC89C52指令系统分类介绍

数据传送类指令

算术运算类指令

逻辑运算与循环类指令

程序转移类指令

调用子程序及返回指令

位操作指令

空操作指令

算术运算类指令

算术运算类指令主要是对8位无符号数进行算术操作，包括加、减、加1、减1以及乘法和除法运算指令。这类指令会影响PSW的有关位，例如：加法、减法运算指令执行结果影响PSW的进位位CY、溢出位OV、半进位位AC和奇偶校验位P。对这一类指令要特别注意正确地判断结果对标志位的影响。

加法指令

ADD A,Rn　　；（A）+(Rn) →A ,n=0～7

ADD A,direct　；（A）+(direct) →A

ADD A,@Ri　　；（A）+(@Ri) →A, i=0,1

ADD A,#data　；（A）+data →A

这组指令的特点：一个加数总是来自累加器A，并且运算结果放在累加器A中。

使用加法指令时，对各个标志位的影响：

- ✓ 位7有进位，则置“1”进位标志Cy，否则清“0”Cy
- ✓ 位3有进位，置“1”辅助进位标志Ac，否则清“0”
- ✓ 如果位6有进位，而位7没有进位，或者位7有进位，而位6没有，则置“1” OV，否则清“0”OV。

注意：溢出标志位OV，只有在有符号数加法运算时才有意义。当2个有符号数相加时，OV=1，表示加法运算超出了累加器A所能表示的带符号数的有效范围（-128 ~ +127）

例：（A）＝53H,(R0)=FCH

执行指令 ADD A,R0

0101 0011

＋）1111 1100

1 0100 1111

结果：

(A)=4FH Cy=1

Ac=0 P=1

OV=0(位6、位7同时有进位)

例：（A）＝85H

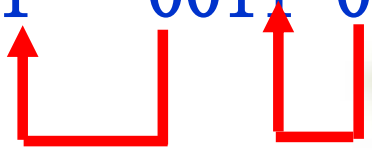
(R0)=20H,(20H)=AFH

执行指令 ADD A,@R0

1000 0101

＋）1010 1111

1 0011 0100



结果：

(A)=34H Cy=1

Ac=1 P=1

OV=1（位7有进位，位6无）

带进位加法指令

ADDC A,Rn ;（A）+(Rn)+ Cy →A ,n=0～7

ADDC A,direct ;（A）+(direct) + Cy →A

ADDC A,@Ri ; (A)+(@Ri) + Cy →A, i=0,1

ADDC A,#data ;（A）+data + Cy →A

这组指令的特点：进位标志位Cy参加运算。功能是同时把所指出的字节变量、进位标志与累加器A内容相加，结果留在累加器A中。

增量指令

INC A

INC Rn ；n＝0～7

INC @Ri ；i=0,1

INC DPTR（16位数加1指令）

INC direct

指令功能:把所指出的变量加1。

指令首先对低8位指针DPL的内容执行加1的操作，当产生溢出时，就对DPH的内容进行加1操作。

十进制调整指令

DA A

这条指令是对压缩BCD码加法运算时，跟在ADD和ADDC指令之后，用于对BCD码的加法运算结果自动进行修正，使其结果仍为BCD表达形式。

调整方法：

✓若 $(A_{0-3})>9$或 $(AC)=1$则 $(A_{0-3})+06H\rightarrow(A_{0-3})$

✓若 $(A_{4-7})>9$或 $(C_y)=1$则 $(A_{4-7})+06H\rightarrow(A_{4-7})$

例：（A）=56D（R5）=67D，56D+67D =123D

```
ADD  A，R5        0 1 0 1  0 1 1 0
DA   A       +    0 1 1 0  0 1 1 1
                  ---------------
                  1 0 1 1  1 1 0 1
调整：       +    0 1 1 0  0 1 1 0
                  ---------------
              1 0 0 1 0  0 0 1 1
```

结果：（A）=23H,Cy=1

在计算机中，遇到十进制调整指令时，中间结果的修正是由ALU硬件中的十进制修正电路自动进行时，用户不必考虑何时该加“6”，使用时只需在上述加法指令后面紧跟一条“DA A”指令即可。

注意：在STC89C52中，“DA A”指令不适用于减法指令。

带进位减法指令

SUBB A,Rn ；（A）-(Rn)- Cy →A ,n=0～7

SUBB A,direct ；（A）-(direct) - Cy →A

SUBB A,@Ri ; (A)-(@Ri) - Cy →A, i=0,1

SUBB A,#data ；（A）-data - Cy →A

这组带进位减法指令是从累加器A中减去指定的变量和进位标志，结果存在累加器中。

- 如果位7需借位则置位Cy，否则清“0”Cy；
- 如果位3需借位则置位Ac，否则清“0”Ac；
- 如果位6需借位而位7不需要借位，或位7需借位而位6不需要借位，则置位溢出标志位OV，否则清“0”OV。

例： (A) =C9H, (R2)=54H, Cy=1

执行指令 **SUBB A,R2**

	1100	1001
	0101	0100
−)		1
	0111	0100

结果为： (A) =74H, Cy=0, Ac=0, OV=1

减1指令

DEC A ;(A)-1→A

DEC Rn ;(Rn) -1→Rn ,n＝0～7

DEC direct ;(direct)-1→direct

DEC @Ri ;((Ri))-1→(Ri),i＝0,1

指令功能:是指定的变量减1。若原来为00H,减1后下溢为FFH。

乘法指令

MUL AB ； A*B →BA

这条指令的功能是把累加器A和寄存器B中的无符号8位整数相乘。其16位积的低位字节在累加器A中，高位字节在B中。如果积大于255，则置位溢出标志OV，否则清“0”OV。

注意：进位标志总是清“0”。

除法指令

DIV AB ;A/B → A(商)，余数→ B

指令的功能:把累加器A中8位无符号整数除以B中的8位无符号整数。商存放在累加器A中，余数在寄存器B中，清“0”Cy和溢出标志位OV。

注意：如果除数B的内容为0，则结果A,B中的内容不定，并置溢出标志位OV位1。

3.2 STC89C52指令系统分类介绍

数据传送类指令

算术运算类指令

逻辑运算与循环类指令

程序转移类指令

调用子程序及返回指令

位操作指令

空操作指令

逻辑运算与循环类指令

- ✓简单逻辑操作指令
- ✓左环移指令
- ✓右环移指令
- ✓带进位左环移指令
- ✓带进位右环移指令
- ✓逻辑与指令
- ✓逻辑或指令
- ✓逻辑异或指令

简单逻辑操作指令

1） CLR A

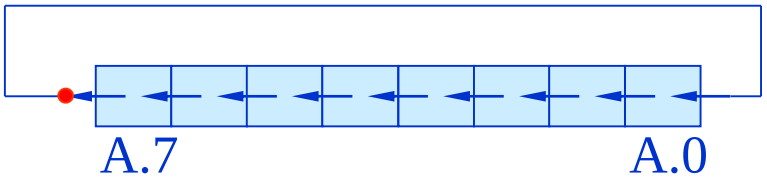
指令功能：累加器A清“0”。不影响Cy,Ac,OV等标志。

2） CPL A

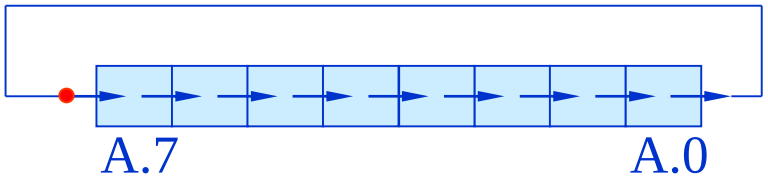
指令功能：累加器A按位取反，不影响标志。

循环移位指令

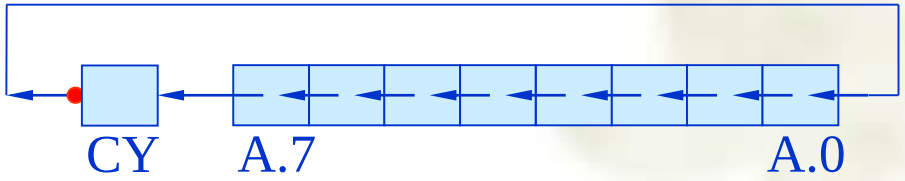
RL A



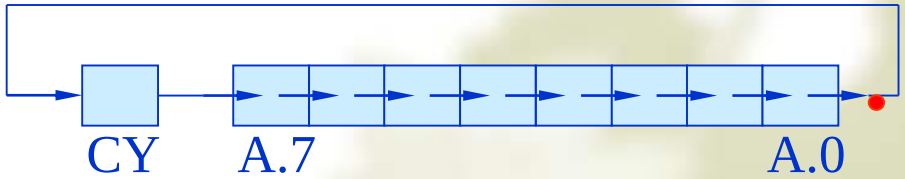
RR A



RLC A



RRC A



注意:执行带进位的循环移位指令之前，必须给CY置位或清零。

逻辑与指令

ANL A , Rn ;（A）∧（Rn）→A, n＝0～7

ANL A , direct ;（A）∧（direct）→A

ANL A , #data ;（A）∧data→A

ANL A , @Ri ;（A）∧（（Ri））→A

ANL direct , A ;（direct）∧A→direct

ANL direct , #data ;（direct）∧data→direct

指令功能：在指出的变量之间以位为基础进行逻辑与操作，结果存放到目的变量所在的寄存器或存储器中去。

逻辑或指令

ORL A，Rn ；（A）∨（Rn）→A，n＝0～7

ORL A，direct ；（A）∨（direct）→A

ORL A，#data ；（A）∨data→A

ORL A，@Ri ；（A）∨（（Ri））→A

ORL direct，A ；（direct）∨A→direct

ORL direct，#data ；（direct）∨data→direct

指令功能：在指出的变量之间执行以位为基础的逻辑“或”操作，结果存放到目的变量所在的寄存器或存储器中去。

逻辑异或指令

XRL A，Rn ；（A）+（Rn）→A ,n＝0～7

XRL A，direct ；（A）+（direct）→A

XRL A，#data ；（A）+data→A

XRL A，@Ri ；（A）+（（Ri））→A

XRL direct，A ；（direct）+A→direct

XRL direct，#data ；（direct）+data→direct

指令功能：在指出的变量之间执行以位为基础的逻辑异或操作，结果存放到目的变量所在的寄存器或存储器中去。

3.2 STC89C52指令系统分类介绍

数据传送类指令

算术运算类指令

逻辑运算与循环类指令

程序转移类指令

调用子程序及返回指令

位操作指令

空操作指令

程序转移类指令

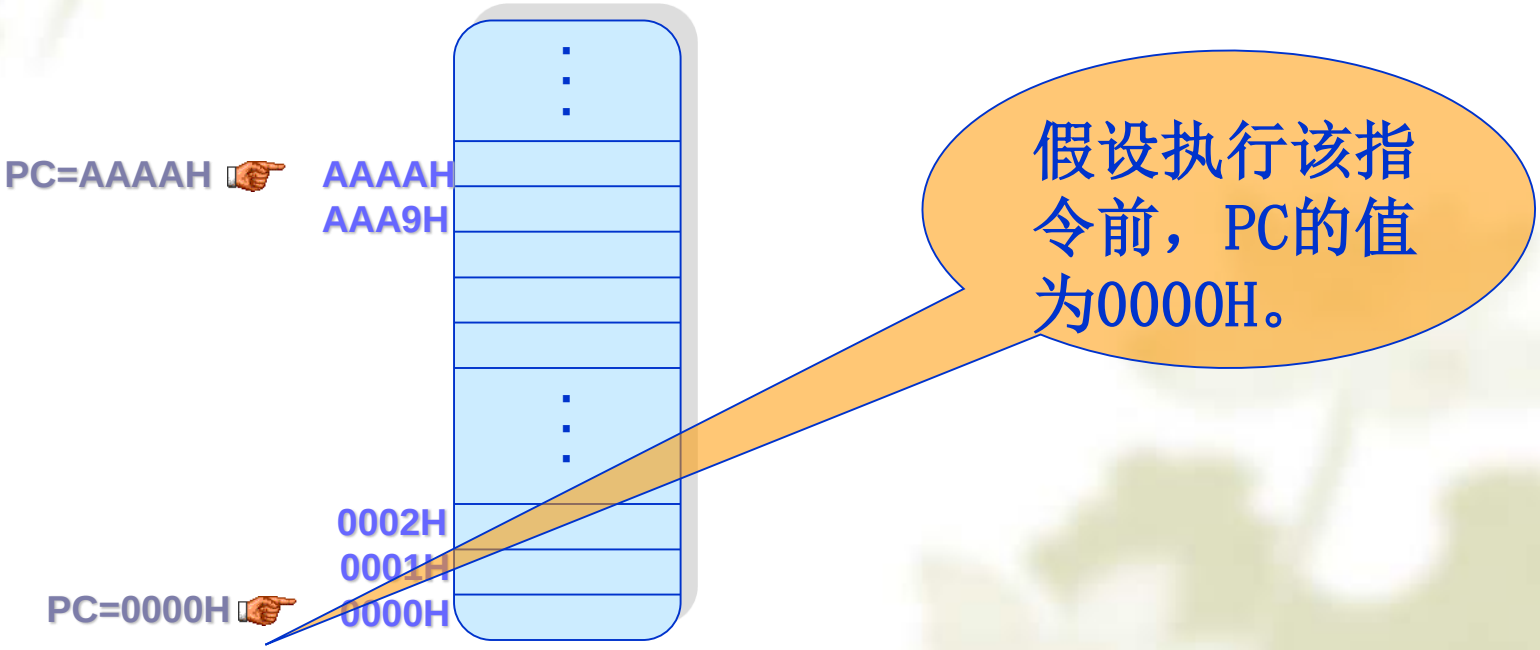
长跳转指令
绝对转移（短转移）指令
相对转移指令
间接跳转指令（散转指令）
条件转移指令
比较转移指令
减1不为0转移指令 （循环指令）

长跳转指令

LJMP addr16

指令中包含**16**位地址，所以转移的目标地址范围是程序存储器的**0000H--FFFFH**。指令执行时把指令的第二和第三字节分别装入**PC**的高位和低位字节中，无条件地转向指定地址。

LJMP　AAAAH ；AAAAH→PC



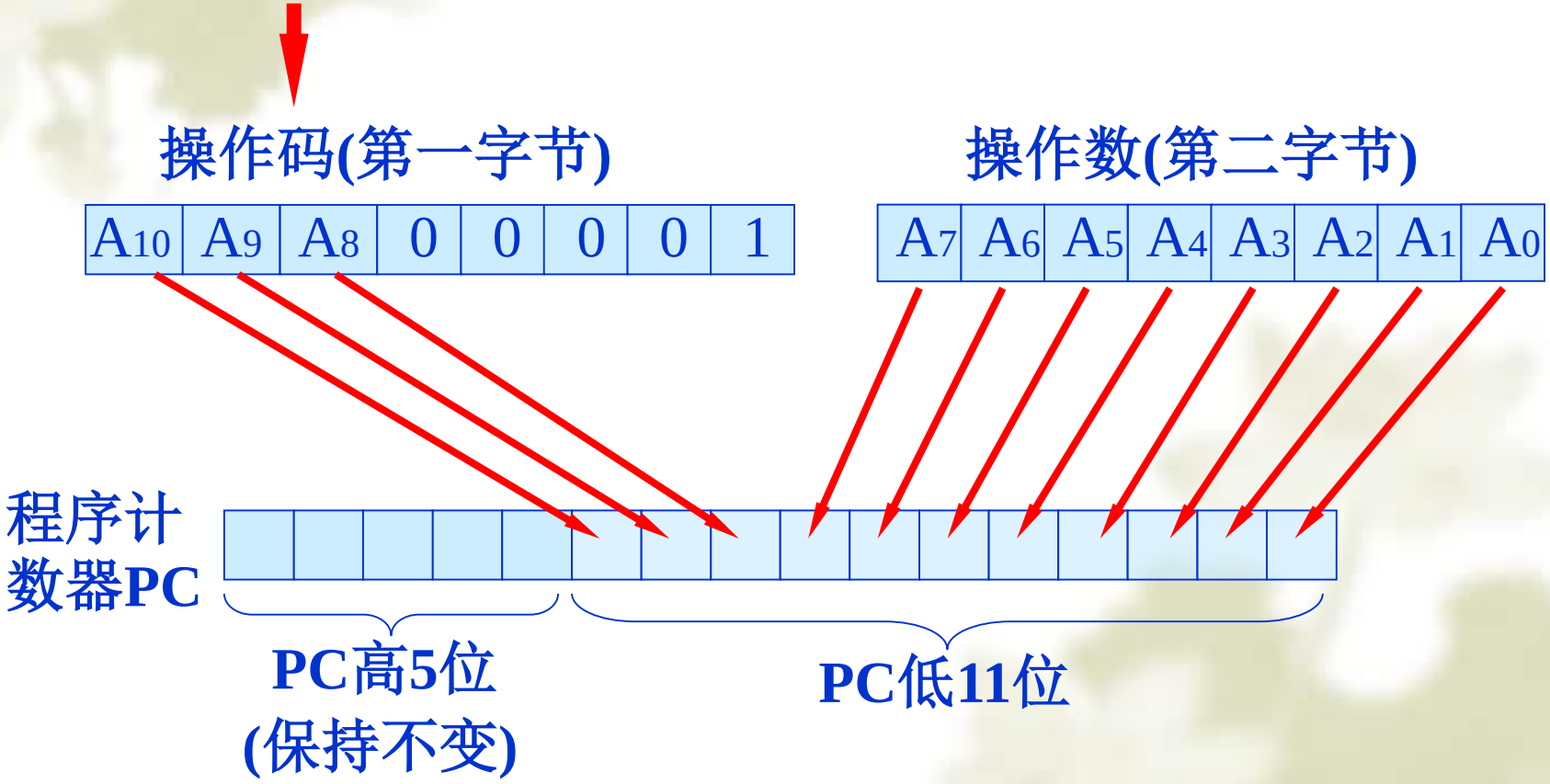
注意:该指令可以转移到64 KB程序存储器中的任意位置。

绝对转移（短转移）指令

AJMP addr11

指令中包含11位地址，因此这是2K字节范围内的跳转指令。执行该指令时，先将PC加2，然后把addr11送入PC.10～PC.0，PC.15～PC.11不变，程序转移到指定的地方。

AJMP addr11； PC+2→PC， addr11 →PC.10~PC.0



11位转移地址的形成示意图

相对转移指令

SJMP rel

这是无条件跳转指令，执行时在PC加2之后，把指令的有符号的偏移量rel(-128～+127，负数表示向后转移，正数向前转移)加到PC上，并计算出转向地址。

程序汇编时，直接写上要转向的目标地址标号就可以了。例如：

```
LOOP: MOV   A,R6
      …
      SJMP  LOOP
      …
```

手工汇编时，

计算目的地址值

目的地址值 = 本指令地址值+2+rel

计算偏移量

rel=目标地址-本指令地址值-2

例：　执行指令　　2000H　：　SJMP　07H

问执行后PC值的变化？

间接跳转指令（散转指令）

JMP @A+DPTR

指令功能:累加器中的8位无符号数与数据指针DPTR的16位数相加，结果作为下条指令的地址送入PC，不改变累加器和数据指针DPTR的内容，也不影响标志。

条件转移指令

条件转移指令是依某种特定条件转移的指令。条件满足时转移，不满足时顺序执行下一条指令。

JZ rel ；如果累加器为零则执行转移

JNZ rel；如果累加器不为零则执行转移

比较转移指令

CJNE A,direct,rel

CJNE A,#data,rel

CJNE Rn,#data,rel

CJNE @Ri,#data,rel

(PC)+3 →(PC)

若 (源地址)<(目的地址)

则 (PC)+REL →(PC)

且 0 → (CY)

若 (源地址)>(目的地址)

则 (PC)+REL →(PC)

且 1→ (CY)

若 (源地址)=(目的地址)

则顺序执行

减1不为0转移指令(循环指令)

这组指令把减1与条件转移两种功能结合在一起。执行时，将源操作数减1，结果回送到源操作数寄存器或存储器中去。

DJNZ Rn, rel ；n＝0~7

DJNZ direct,rel

在实际问题中，经常需要多次重复执行某段程序。在编程时，可以设置一个计数值，每执行一次这段程序，计算值减1，直至计数值减为0为止。

3.2 STC89C52指令系统分类介绍

数据传送类指令

算术运算类指令

逻辑运算与循环类指令

程序转移类指令

调用子程序及返回指令

位操作指令

空操作指令

调用子程序指令

（1） 短调用指令：**ACALL addrll**

PC+2→PC,SP+1→SP,

PC.7--PC.0→(SP), SP+1→SP

PC.15--PC.8→(SP), addr11→PC.10--PC.0

2KB范围内的调用子程序指令,其操作码的形成类似于“AJMP”指令。

（2）长调用指令：

LCALL addrl6 PC+3→PC,SP+1→SP,

PC.7～PC.0→(SP), SP+1→SP

PC.15～PC.8→(SP), addr16→PC

这条指令无条件的调用位于指定地址的子程序。

子程序返回指令

RET ； (SP) →PC.15～PC.8，SP-1→SP (SP) →PC.7～PC.0，SP-1→SP

指令功能:从堆栈中退出PC的高位和低位字节，把栈指针减2，从PC值开始继续执行程序。不影响任何标志。

中断返回指令

RETI

指令功能:和RET指令相似，不同的是清除STC89C52内部的中断状态标志。

3.2 STC89C52指令系统分类介绍

数据传送类指令

算术运算类指令

逻辑运算与循环类指令

程序转移类指令

调用子程序及返回指令

位操作指令

空操作指令

位操作指令

由于**STC89C52**单片机内部有一个位处理机,所以对位地址空间具有比较丰富的位操作指令。

- 数据位传送指令
- 位变量修改指令
- 位变量逻辑与指令
- 位变量逻辑或指令
- 条件转移类指令

数据位传送指令

MOV C , bit

MOV bit , C

指令功能：把由源操作数指出的位变量送到目的操作数指定的单元中去。

注意：一个操作数必须为进位标志(位累加器)。

例： 指令 MOV C , 06H

必须要注意，这里的06H是内部RAM20H字节位6的位地址。

位变量修改指令

CLR C ; 清C_y

CLR bit ; 清bit

CPL C ; C_y求反

CPL bit ; bit位求反

SETB C ; 置1C_y

SETB bit ; 置1 bit位

例：指令如下

CLR 27H ; 0→(24H).7

CPL 08H ; 将（21H）.0取反

SETB P1.7 ; 1→P1.7

STC89C52内部RAM的可寻址位及其位地址表

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

位变量逻辑与指令

ANL C,bit

ANL C,/bit

位变量逻辑或指令

ORL C,bit

ORL C,/bit

条件转移类指令

JC rel ；如果进位位C_y=1，则转移

JNC rel ；如果进位位C_y=0，则转移

JB bit, rel；如果直接寻址位bit=1，则转移

JNB bit, rel；如果直接寻址位bit=0，则转移

JBC bit, rel；如果直接寻址位bit=1，则转移，并清bit

3.2 STC89C52指令系统分类介绍

数据传送类指令

算术运算类指令

逻辑运算与循环类指令

程序转移类指令

调用子程序及返回指令

位操作指令

空操作指令

空操作指令

NOP

CPU不进行任何实际操作，只消耗1个机器周期的时间。只执行（PC+1）→PC操作。NOP指令常用于程序中的等待或时间的延迟。

若SP=60H，标号LABEL所在的地址为3456H。LCALL指令的地址为2000H，执行指令

2000H:　　LCALL　　LABEL

问堆栈指针SP和堆栈内容执行前后发生了什么变化？PC的值等于什么？如果将指令LCALL直接换成ACALL是否可以？

1）SP=SP+1=61H (61H)=PC的低字节=03H
　　SP=SP+1=62H(62H)=PC的高字节=20H
2）PC=3456H
3）不可以

重难点总结

1、内存的高128字节和SFR的访问。

2、内部RAM、外部扩展RAM、程序存储器访问方式。

3、工作寄存器区和内部RAM的关系。

4、七种寻址方式各自的用途。

5、位寻址区及位地址的表示方法。

6、PSW中Cy，Ac, OV, P的含义，OV的计算方法。

7、子程序调用和返回指令是对应的，涉及PC的保护和恢复。