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

合肥工业大学 Hefei University of Technology

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

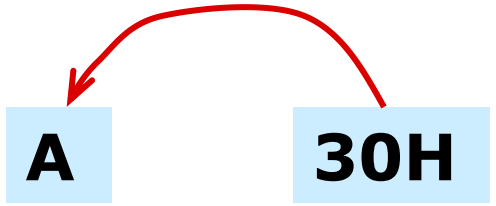
合肥工业大学 Hefei University of Technology

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

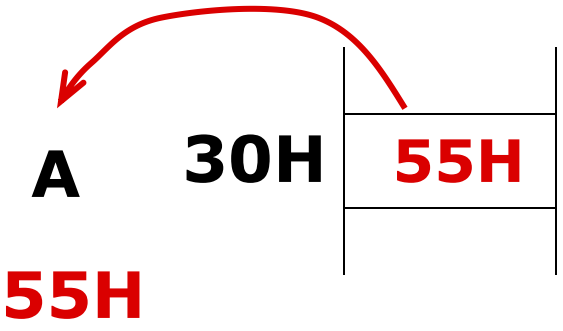


二、直接寻址

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

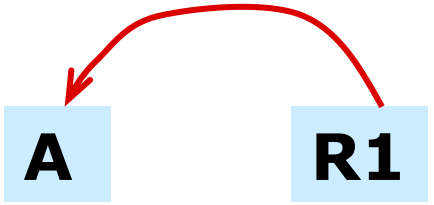


指令系统的寻址方式

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



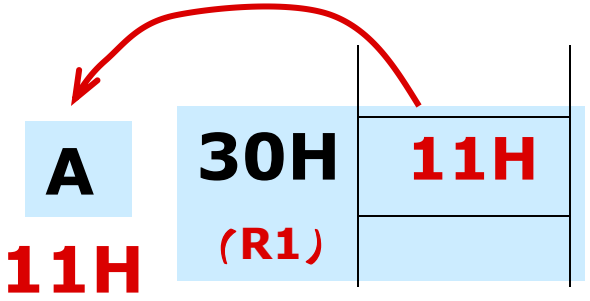
四、寄存器间接寻址

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

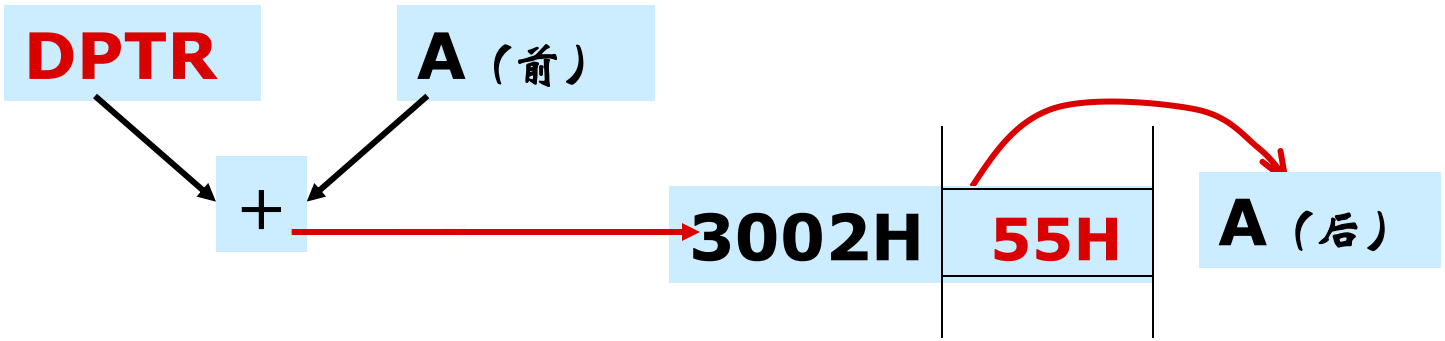


指令系统的寻址方式

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：

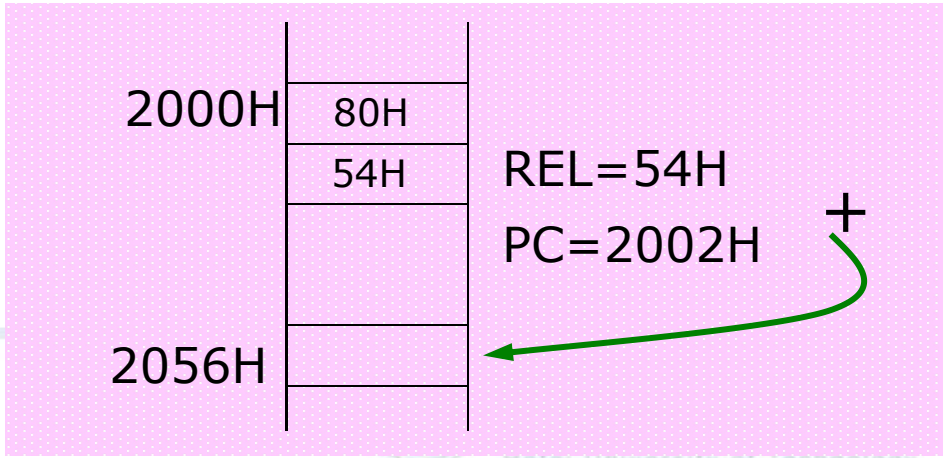
目的地址 = PC当前值起始地址+rel

= 转移指令所在地址+转移指令字节数+rel

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

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。

合肥工业大学 Hefei University of Technology

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。

合肥工业大学 Hefei University of Technology

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
```

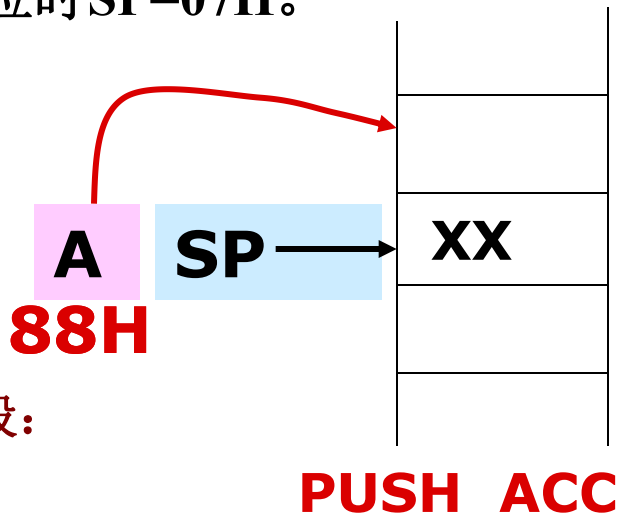
MCS-51指令系统介绍

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
```

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   $
```

合肥工业大学 Hefei University of Technology

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≤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

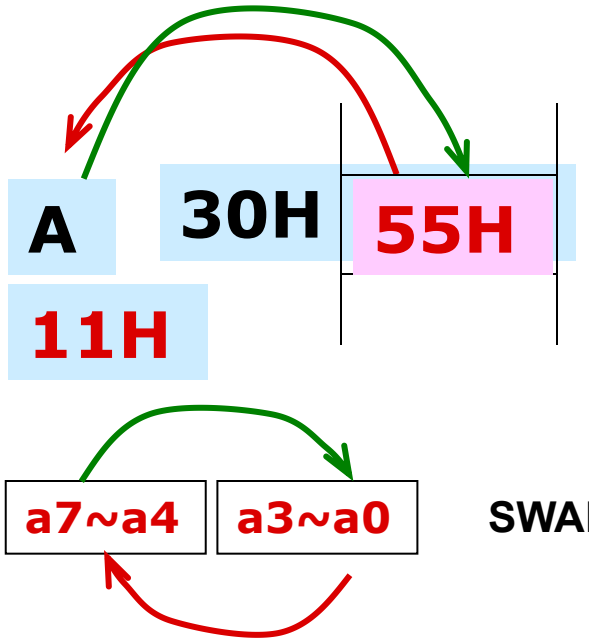
有3条XCH指令为整个字节相互交换，XCHD指令为低4位相互交换，SWAP指令为ACC中的高、低4位互换。

(1）字节交换

```
XCH  A, Rn          ; (A)←→(Rn)
XCH  A, direct      ; (A)←→(direct)
XCH  A, @Ri         ; (A)←→((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）→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。

```
        0101 0011
+       1111 1100
-------------------
    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)  →A
ADDC   A, direct      ; (A)+(direct)+(Cy)  →A
ADDC   A, @Ri         ; (A)+((Ri))+(Cy)  →A
ADDC   A, #data       ; (A)+#data +(Cy) →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→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) →A
SUBB   A, direct      ;  (A)-(direct)-(Cy) →A
SUBB   A, @Ri         ;  (A)-((Ri))-(Cy) →A
SUBB   A, #data       ;  (A)-#data-(Cy) →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。

```
        1100 1001
        0101 0100
-                1
-------------------
        0111 0100
```

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)×(B) →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→A，余数→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。

8. 十进制调整指令

ADD、ADDC指令都是对8位二进制数进行加法运算，当两个BCD码数进行加法时，必须增加一条DA A指令（对其结果进行调整），否则结果就会出错。

DA A

指令的调整原则是：

- 若（A0~A3）>9或（Ac）=1，则低4位（A0~A3）+6调整；
- 若（A4~A7）>9或（Cy）=1，则高4位（A4～A7）+6调整；
- 若（A4~A7）=9且（A0~A3）>9，则高4位（A4～A7）和低4位（A0~A3）分别+6调整。

注意：1）DA指令只能跟在加法指令后面使用；
2）调整前参与运算的两数是BCD码数；
3）DA指令不能与减法指令配对使用。

8. 十进制调整指令

例：设（A）=38H（压缩BCD码数），执行如下指令：

```
ADD  A，#59H   ；实现38+59＝97的操作
     DA  A     ；调整
```

```
      0011  1000
      0101  1001
+     1001  0001
            0110
      1001  0111
```

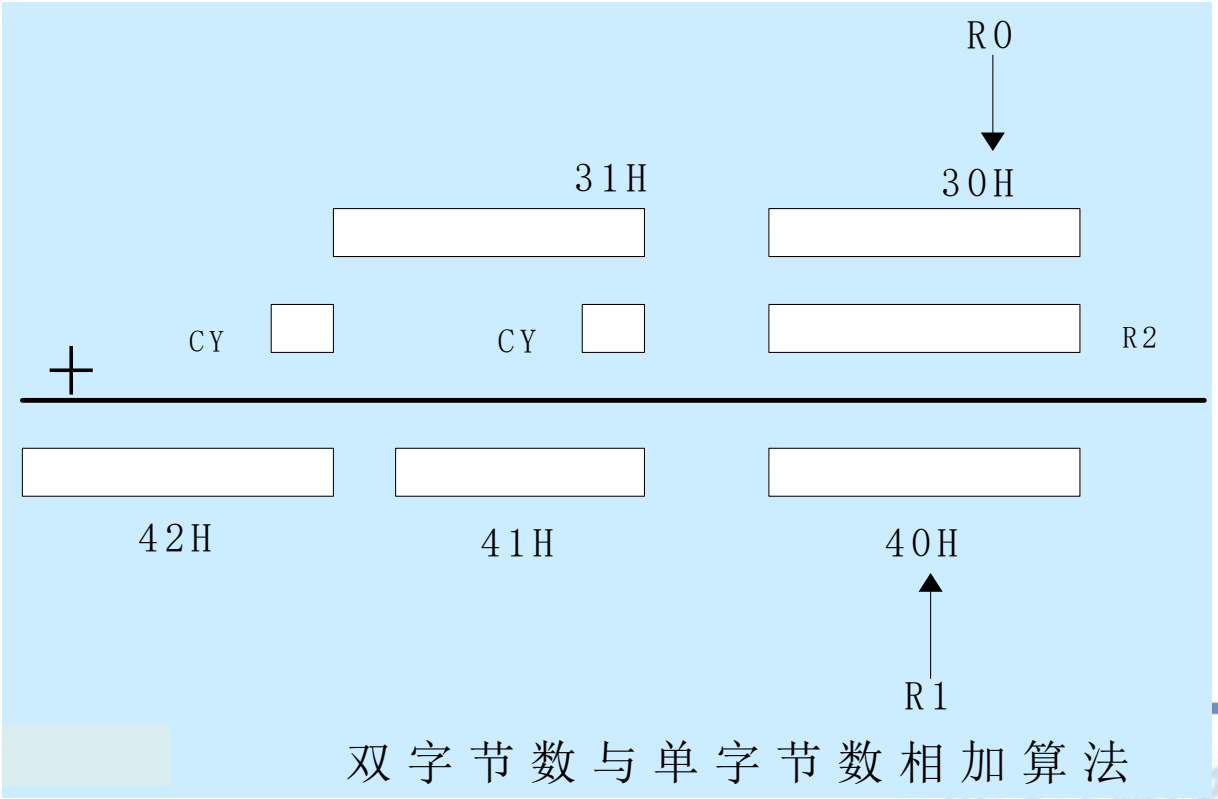
例：设（A）=85H（压缩BCD码数），执行如下指令：

```
ADD  A，#15H   ；实现85+15＝100的操作
     DA  A     ；调整
```

```
      1000  0101
      0001  0101
+     1001  1010
      0110  0110
    1 0000  0000
```

例：课本P54（旧版P41）的例子

例：在片内RAM 31H、30H中存有双字节数（高字节在31H、低字节在30H中），编程把该双字节数与R2中单字节数相加，和存在片内RAM 40H单元开始的空间中（低字节在先）。



双字节数与单字节数相加算法

```
MOV     R0，#30H        ；置被加数地址指针首址
MOV     R1，#40H        ；置和地址指针首址
MOV     A，  @R0        ；取被加数低字节
ADD     A，R2           ；低字节相加，并产生进位Cy
MOV     @R1，A          ；存和的低字节
INC     R0              ；地址指针增1，指向31H
INC     R1              ；地址指针增1，指向41H
MOV     A，@R0          ；取被加数的高字节
ADDC    A，#0           ；高字节与进位Cy相加，产生新的进位
MOV     @R1，A          ；存和中字节
INC     R1              ；地址指针增1，指向42H
MOV     A，#0           ；
ADDC     A，#0          ；把高位的进位Cy转到A中
 MOV    @R1，A          ；存和的高字节，和可能为三字节数
```

合肥工业大学 Hefei University of Technology

MCS-51指令系统介绍

三、逻辑运算指令

逻辑运算类指令包括求反与清除指令、循环移位指令、“与”操作指令 、“或”操作指令、“异或”操作指令。

特点是当A作目的操作数（第一操作数）时，影响P位；带进位的移位指令影响Cy、P位，其余都不影响PSW。

1. 清0指令

```
CLR     A       ；将累加器A的内容清0；
                ；不影响Cy、Ac、OV标志位。
```

2. 求反指令

```
CPL     A       ；对累加器A的内容各位求反；
                ；不影响标志位。
```

MCS-51指令系统介绍

3. 循环移位指令（仅对A有效）

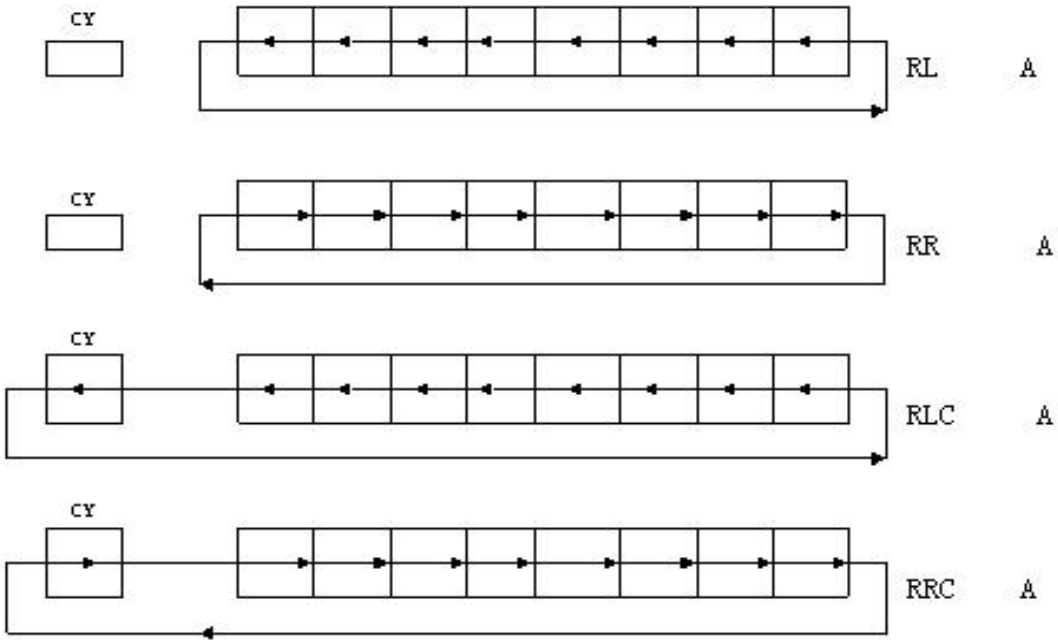
- 两条不带Cy位的逐位循环移位一次指令，不影响PSW。
- 两条带Cy位的逐位循环移位一次指令，影响Cy、P标志位。

RL A ；左移

RR A ；右移

RLC A ；带进位CY的左移

RRC A ；带进位CY的右移



MCS-51指令系统介绍

4. 逻辑与指令

两个操作数之间按位“与”操作，结果放到目的变量（第一操作数）中。

逻辑“与”指令格式

```
ANL  A，Rn              ；(A)∧(Rn)→A
ANL  A，direct
ANL  A，@Ri
ANL  A，#data
ANL  direct，A
ANL  direct，#data
```

```
  01100111
AND 11011100
------------
  01000100
```

- 例：将A中高位清 0 低 5 位不变.

  ANL A，#1FH

- 用法：屏蔽某些数位为0

MCS-51指令系统介绍

4. 逻辑或指令

两个操作数之间按位“或”操作，结果放到目的变量（第一操作数）中。

逻辑“或”指令格式

```
ORL  A，Rn              ；(A)∨(Rn)→A
ORL  A，direct
ORL  A，@Ri
ORL  A，#data
ORL  direct，A
ORL  direct，#data
```

```
  01000110
OR 01011010
-----------
  01011110
```

- 例：将A中高位置1低 5 位不变.

  ORL A，#0E0H

- 用法：屏蔽某些数位为1

MCS-51指令系统介绍

4. 逻辑异或指令

两个操作数之间按位“异或”操作，结果放到目的变量（第一操作数）中。

逻辑“异或”指令格式

XRL A，Rn ；(A)⊕(Rn)→A

XRL A，direct

XRL A，@Ri

XRL A，#data

XRL direct，A

XRL direct，#data

异或操作的用法：

某位用“0”异或不变；

用“1”异或该位取反。

也称为“指定位取反”。

例：将双字节数（R3）（R2）右移一位，最高位补0

```
CLR   C
MOV   A，R3
RRC   A
MOV   R3，A
MOV   A，R2
RRC   A
MOV   R2，A
```

例：将A的低位4位送到P1口的低位，P1高4位不变。

```
MOV  R0，A
ANL  A，＃0FH        ；A的低4位不变，高4位清零
ANL  P1，＃0F0H      ；P1口低4位清零，高4位保持不变
ORL  P1，A           ；A的低4位送P1，并保持P1高4位不变
MOV  A，R0
```

例：使P1口的低2位为0，高2位取反，其余位不变。

```
ANL  P1，#11111100B     ；先对低二位清0
XRL  P1，#11000000B     ；再对高二位取反
```

MCS-51指令系统介绍

四、控制转移类指令

控制程序转移类指令主要功能是控制程序转移到新的PC地址去执行。分为四大类：

无条件转移指令；条件转移指令；调用指令；返回指令。

1. 长转移指令

LJMP addr16　　；addr16→PC

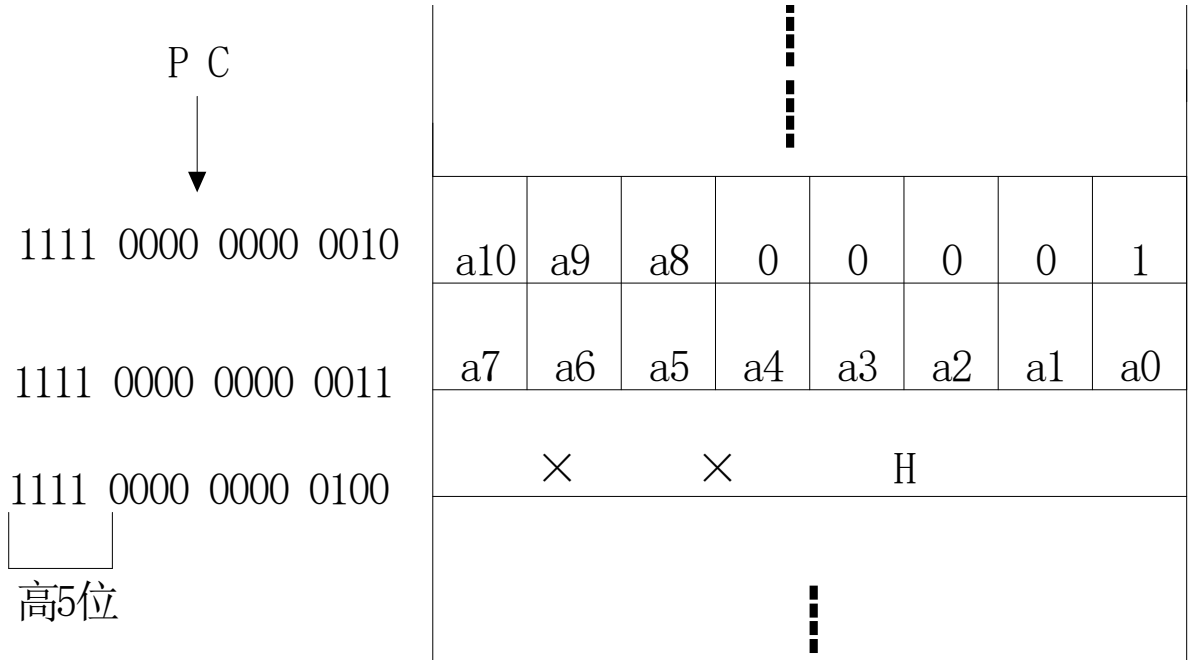
转移目标地址可以在64K程序存储器范围内的任何位置。

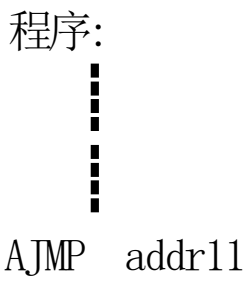
合肥工业大学 Hefei University of Technology

4. 绝对（短）转移指令

AJMP addr11　　；addr11→$PC_{10\sim0}$

指令中包含有11位的目的地址（a10 a9 a8 a7 a6 a5 a4 a3 a2 a1 a0）





- 转移范围2K字节
- PC 高5位不变
- 机器码为两字节
- a10 a9 a8 0 0 0 0 1
- a7----------a0

AJMP指令的转移目标地址必须与AJMP下一条指令的第一个字节在同一个2Kbyte区范围内，即转移目标地址必须与下一条指令的高5位地址一致。

合肥工业大学 Hefei University of Technology

MCS-51指令系统介绍

2. 相对转移指令

SJMP rel ； (PC)+rel →PC

- 指令执行时，把指令的有符号的偏移量rel加到PC上，计算出目标地址。因此跳转的目的地址可以在下一条指令为起点的（-128~+127）个字节的偏移范围之内。

3. 间接转移指令

JMP @A+DPTR；(A)+(DPTR) →PC

- 目的地址由指针DPTR作为基址，和变址A（无符号数）的内容之和形成 。
- 本指令不改变A和DPTR的内容，也不影响标志位。

MCS-51指令系统介绍

实际应用时，控制转移类指令中的addr16、addr11、rel一般用符号地址形式，由汇编程序自动生成目的地址或偏移量。通常不需要人工计算。

合肥工业大学 Hefei University of Technology

MCS-51指令系统介绍

5. 条件转移指令

条件转移指令都是依据某种条件成立才转移（不成立则继续顺序执行）的指令。此类指令均为相对寻址指令。

范围：以下一条指令的首地址为中心的-128~+127字节内。

- 累加器判零转

```
JZ   rel        ；A=0， 转PC=PC（当前）+rel
JNZ  rel        ；A< >0转
```

- Cy位条件判转

```
JC   rel        ；Cy=1，转PC=PC（当前）+rel
JNC  rel        ；Cy=0 转
```

MCS-51指令系统介绍

- 可寻址位条件判转 （三字节）

```
JB  bit，rel        ；(bit)=1，转PC=PC（当前）+rel
JNB  bit，rel       ；(bit)=0 转
JBC  bit，rel       ；(bit)=1 转，并将该位清零
```

注意：上述几条指令都是三字节指令。

合肥工业大学 Hefei University of Technology

MCS-51指令系统介绍

6. 比较不相等转移指令

CJNE A，#data，rel

CJNE A，direct，rel

CJNE Rn，#data ，rel

CJNE @Ri，#data，rel

指令功能为两数比较不相等转移，操作过程为第1操作数减第2操作数。若<0，将Cy置“1”；否则Cy清“0”，但不改变原来的操作数。

范围：以下一条指令的首地址为起点的-128~+127字节内。

注意：参与比较的数为无符号数，CJNE是三字节指令。

MCS-51指令系统介绍

7. 减1不为0转移指令（循环控制指令）

DJNZ Rn, rel ； n=0~7

DJNZ direct, rel

指令功能是把源操作数（Rn或direct）减1，结果送回原操作数中，如果结果不为0就转到目的地去，否则继续下面一条指令执行。

主要应用在循环结构的编程中，预先把循环次数装入Rn或direct，可实现按次数循环执行。

合肥工业大学 Hefei University of Technology

例： 软延时子程序，假设一个机器周期为1us。

```
DELAY:    MOV   R5, #200      ；1个机器周期指令
D1:       MOV   R6, #250      ；1个机器周期指令
          DJNZ  R6, $         ；2个机器周期指令
          DJNZ  R5, D1        ；2个机器周期指令
          RET                 ；2个机器周期指令
```

问题：上述程序大约延时了多少时间？

例：根据A中的数是大于/等于/小于64H这三种情况去执行三种不同的处理程序。

```
CJNE  A，#64H，NEQ          ；不等则转到NEQ
EQ:     ┆                   ；执行相等的处理程序
        LMP  NEXT
NEQ：JNC      BIG           ；> 则转到BIG去执行程序
LOW:    ┆                   ；否则执行 < 的处理程序
        LMP  NEXT
BIG:    ┆                   ；执行 > 的处理程序
```

合肥工业大学 Hefei University of Technology

例：判正负数　要求从P1口输入一个数，若为正数将其存入20H单元，为负数则取反后存20H单元。

解：分析怎样判断数的正负？

D7=1 负数，D7=0 正数

程序如下：

```
        ORG   0030H
MAIN:   MOV  P1，#0FFH;
        MOV  A，P1
        JNB  ACC.7, STOR     ；正数
        CPL  A
STOR:   MOV  20H，A;
        SJMP  $
```

判断数的正负的其他方法：

```
a）     RLC    A
        JNC    STOR
b）     ANL  A，#80H;
        JZ     STOR;
```

MCS-51指令系统介绍

8. 子程序调用指令

（1）长调用指令

LCALL addr16 ；(PC)+3 →PC，
；(SP)+1 →SP，($PC_{7\sim0}$) →(SP)，
；(SP)+1 →SP，($PC_{15\sim8}$) →(SP)，
；addr16 →PC

（2）绝对（短）调用指令

ACALL addr11 ；(PC)+2 →PC，
；(SP)+1 →SP，($PC_{7\sim0}$) →(SP)，
；(SP)+1 →SP，($PC_{15\sim8}$) →(SP) ，
；addr11 →$PC_{10\sim0}$

调用地址的形成过程同AJMP指令。

MCS-51指令系统介绍

9. 子程序返回指令

RET ；((SP))→$PC_{15\sim8}$，(SP)-1→SP

；((SP))→$PC_{7\sim0}$，(SP)-1→SP

从堆栈中弹出（断点）地址值给PC（先高后低，栈指针减2），使程序从该PC值处开始执行程序。不影响PSW。

10. 中断返回指令

RETI ；((SP))→$PC_{15\sim8}$，(SP)-1→SP

；((SP))→$PC_{7\sim0}$，(SP)-1→SP

注意：不能用RET指令代替RETI。

除具有RET指令的所有功能外，还将自动清除优先级寄存器的优先级状态。RETI指令用在中断服务子程序中，作最后一条返回指令。

MCS-51指令系统介绍

11. 空操作指令

NOP

指令不执行任何实际操作，执行时间为一个机器周期，占一个字节。

常用于程序中的等待或者时间延迟。

合肥工业大学 Hefei University of Technology

MCS-51指令系统介绍

五、位操作指令

1. 位传送指令

```
MOV   C，bit        ；  (bit) → Cy
MOV   bit，C        ；  (Cy) → bit
```

指令中必须有一个位操作数是布尔累加器C，另一个可以是任何直接可寻址的位。

```
例：MOV   P1.5，C        ；把Cy中的值送到P1.5口线输出
    MOV   C，06H         ；把(20H).6中的值送到Cy
```

MCS-51指令系统介绍

2. 位修改指令

```
CLR   C          ；将Cy清“0”
CLR   bit        ；将bit位清“0”
SETB  C          ；将Cy置“1”
SETB  bit        ；将bit位置“1”
CPL   C          ；Cy求反
CPL   bit        ；bit位求反
```

例：
```
CLR   27H        ；0 → 24H.7
CPL   08H        ；(21H.0)‾ → 21H.0
SETB  P1.7       ；1 → P1.7
```


Hefei University of Technology

MCS-51指令系统介绍

3. 位逻辑运算指令

这组指令的第一操作数必须是Cy，第二操作数是直接寻址位；两位逻辑运算的结果送回Cy中；式中的斜杠是位取反，但并不影响第二操作数本身的值。

```
ANL   C，bit        ；  (Cy) ∧ (bit) →Cy
ANL   C，/bit       ；  (Cy) ∧ /(bit) →Cy
ORL   C，bit        ；  (Cy) ∨ (bit) →Cy
ORL   C，/bit       ；  (Cy) ∨ /(bit) →Cy
```

作业3-2

第三部分 作业 2

一、判断以下指令是否正确

（1）MOV R0，25H　　（2）DW 0200H
（3）DEC DPTR　　（4）CPL P1
（5）MOVX A，@R0　　（6）DJNZ R0，#80H，LOOP
（7）MOV A，#DFH　　（8）DIV A，#05H
（9）POP R5　　（10）MOV C，P3.2

二、已知程序执行前有 A=01H，SP=6AH，（69H）=50H，（6AH）=80H。下述程序执行后，则 A=_____；SP= _____；（69H）=_____；（6AH）=_____；PC=_____。

```
POP    DPH
POP    DPL
MOV    DPTR，#3000H
RL     A
MOV    B，A
MOVC   A，@A+DPTR
PUSH       ACC
MOV    A，B
RL     A
MOVC       A，@A+DPTR
PUSH       ACC
RET
ORG    3000H
DB     11H，22H，33H，44H，55H，66H
```

三、假定(A)=57H，(R0)=63H，(63H)=0A1H，执行以下指令后，(A)=_____。

```
ANL        A，#63H
ORL        63H，A
XRL        A，@R0
CPL        A
```

四、编一段程序，将 4 个放在片内 30H~33H 存储单元中的单字节数进性求和，求和结果放在片内 41H 和 40H 单元，其中 41H 存放高位字节。

15:36:54