3.6 逻辑运算指令

逻辑运算指令共24条，包括与、或、异或、清零、求反和左、右移位等逻辑指令。

按操作数也可分为单、双操作数两种。逻辑运算指令只按位进行逻辑运算，当以累加器A为目的操作数时，对P标志有影响，但对AC、OV及CY没有影响。

循环移位指令是对累加器A的循环移位操作，包括左、右方向以及带与不带进位位等移位方式，移位操作时，带进位的循环移位对CY和P标志有影响。

助记符9种：ANL、ORL、XRL、RL、RLC、RR、RRC、CPL、CLR。

3.6 逻辑运算指令

3.6.1 累加器A的逻辑操作指令

累加器A操作指令共有6条，可以实现将累加器A中的内容进行取反，清零，循环左、右移位、带CY循环左、右移位。

一、累加器清“0”（ CLR， Clear 1条）

CLR A ；（A）←0

将累加器A清“0”，不影响CY、AC、OV等标志。

例 若（A）=A5H，执行指令 CLR A 之后，（A）=00H。

二、累加器按位取反指令（CPL，Complment 1条）

CPL A ；(A) ←（/A）

将累加器A的每一位逻辑取反，不影响CY、AC、OV等标志。

要点分析：

逻辑运算是按位进行的，累加器的按位取反实际上是逻辑非运算。

例 设（A）＝21H（0010 0001B），执行指令：CPL A

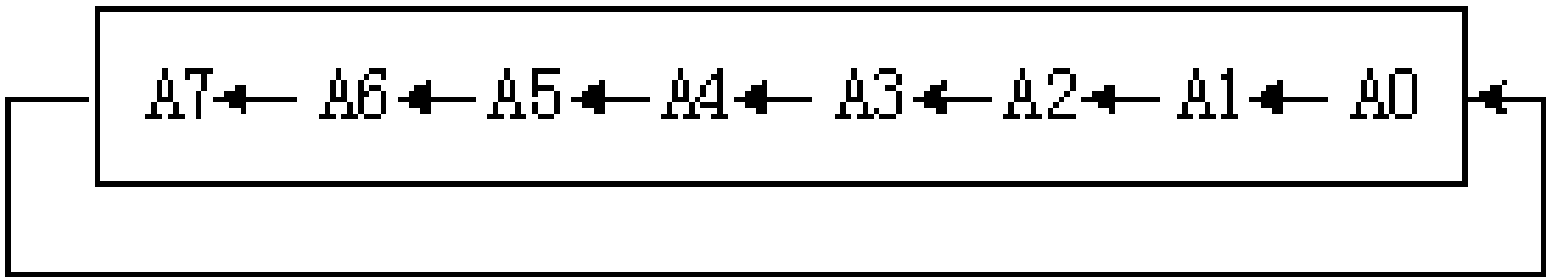
执行结果：（A）＝DEH（1101 1110B）。

3.6 逻辑运算指令

三、左环移指令（RL，Rotate Left 1条）

RL A；An+1←An，n=0～6，A0←A7

将累加器A的内容向左环移一位，位7循环移入位0。不影响标志。



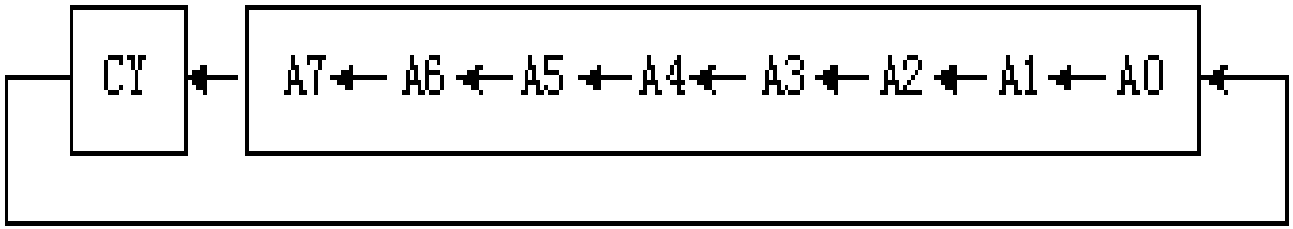
例 设（A）＝3AH（00111010B），执行RL A指令，执行结果：（A）＝74H（01110l00B）。

3.6 逻辑运算指令

四、带进位左环移指令（RLC，Rotate Left with Carry 1条）

RLC A ；An+1←An n=0～6，A0←CY，CY←A7

将累加器A的内容和进位标志CY一起向左环移一位，ACC.7移入进位位CY，CY移入ACC.0。不影响 CY之外的标志位。



例 设（A）＝3AH（00111010B），（CY）＝1，执行RLC A指令，（A）=75H（01110101B），（CY）=0。

有时“累加器A内容乘2”的任务可以利用指令RLC A方便地完成。

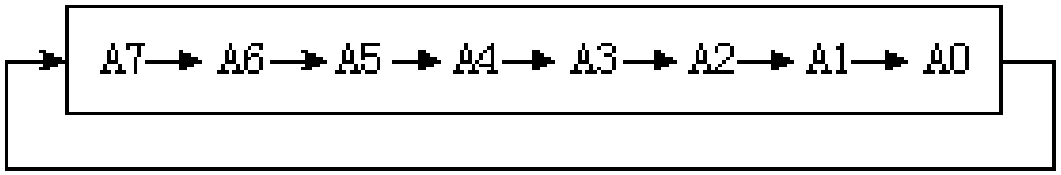
例 若（A）= BDH = 1011 1101B，（CY）＝0。执行指令RLC A后，（CY）=1，（A）= 0111 1010B = 7AH，（CY）＝1。 结果为：17AH（378）＝2×BDH（189）。

3.6 逻辑运算指令

五、右环移指令 （RR，Rotate Right 1条）

RR A； An←An+1，n＝0～6，A7←A0

将累加器A的内容向右环移一位，ACC.0循环移入ACC.7。不影响标志。

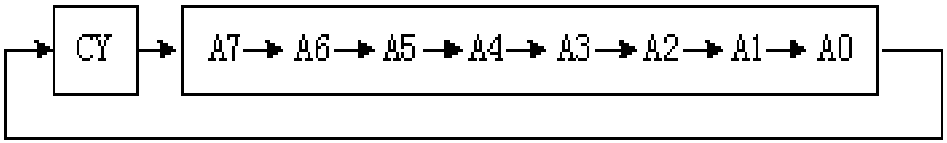


例 设（A）=A6H（10100110），执行RR A指令，执行结果= 53H（01010011B）。

六、带进位右环移指令（RRC，Rotate Right with Carry 1条）

RRC A； An←An+1，n＝0～6，A7←C，C←A0

将累加器A的内容和进位标志CY一起向右环移一位，ACC.0移入进位位CY，CY移入ACC.7。不影响 CY之外的标志位。



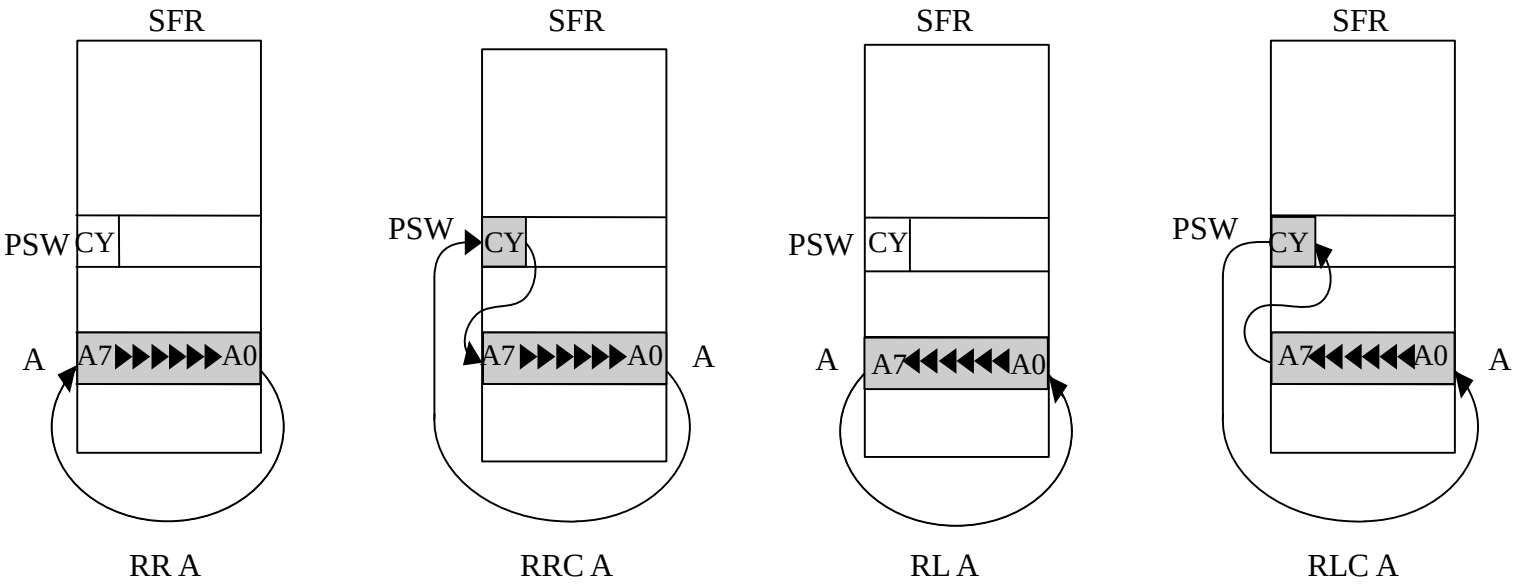
例 设（A）=B4H（10110100B ），（CY）=1，执行RRC A指令，执行结果为：（A）=DAH（11011010B），（CY）=0。

3.6 逻辑运算指令

环移指令说明：

1）执行RL指令1次，相当于把原内容乘以2；执行RR指令1次，相当于把原内容除以2。

2）环移指令执行过程示意图。



3.6 逻辑运算指令

例 试用三种方法将累加器A中无符号数乘2。

方法1：

```
CLR C
RLC A
```

方法2：

```
CLR C
MOV R0，A
ADD A，R0
```

方法3：

```
MOV B，＃2
MUL AB
```

例 编程实现16位数的算术左移。设16位数存放在内部RAM 40H、41H 单元，低位在前。算术左移是指将操作数整体左移一位，最低位补充0。相当于完成对16位数的乘2操作。

解：需要带CY的循环左移，在第一次移位之前CY必须清0。程序如下：

```
CLR  C          ；CY清0
MOV  A，40H     ；取操作数低8位送A
RLC  A          ；低8位左移一位
MOV  40H，A     ；送回原单元保存
MOV  A，41H     ；指向高8位
RLC  A          ；高8位左移
MOV  41H，A     ；送回41H单元保存
```

3.6 逻辑运算指令

3.6.2 两个操作数的逻辑操作指令

两个操作数的逻辑运算指令共有18条，分为逻辑与指令、逻辑或指令和逻辑异或指令。

当需要只改变字节数据的某几位，而其余位不变时，不能使用直接传送方法，只能通过逻辑运算完成。

一、逻辑与指令（And on logical 6条）

运算规则为：0·0=0，0·1=0，1·0=0， 1·1=1。助记符为ANL，用符号“∧”表示。

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

3.6 逻辑运算指令

功能：将两个操作数的内容按位进行逻辑与操作，并将结果送回目的操作数的单元中。前4条指令的功能是把源操作数与累加器A的内容相与，结果送入累加器A中。后2条指令的功能是把源操作数与直接地址指示的单元内容相与，结果送入直接地址指示的单元。

例 设（A）=A3H（1010 0011B），（R0）=AAH（1010 1010B），执行指令：ANL A，R0

执行结果：（A）= A2H（1010 0010B）。

例 （A）=FAH=11111010B，（R1）=7FH=01111111B，执行指令

ANL A，R1 ；（A）← 11111010∧01111111

结果为：（A）=01111010B=7AH。

逻辑“与”ANL指令常用于屏蔽（置0）字节中某些位。若清除某位，则用“0”和该位相与；若保留某位，则用“1”和该位相与。

例如要将一个字节中的高4位清0可用0FH进行“与”操作。

例 设P1=FFH，执行指令：ANL P1，#0F0H

执行结果：P1＝F0H，P1.7～P1.4位状态不变，P1.3～P1.0位被清除。

3.6 逻辑运算指令

二、逻辑或指令（ Or on Logical 6条）

助记符为ORL，用符号“∨”表示，运算规则为：0+0=0，0+1=1，1+0=1，1+1=1。

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

功能：将两个操作数的内容按位进行逻辑或操作，并将结果送回目的操作数的单元中。

前4条指令的功能是把源操作数与累加器A的内容相或，结果送入累加器A中。

后2条指令的功能是把源操作数与直接地址指示的单元内容相或，结果送入直接地址指示的单元。

3.6 逻辑运算指令

例 若（A）=C3H，（R0）=55H，执行指令ORL A，R0 之后，（A）=D7H。

例 设（A）＝A3H（10100011B），（R0）＝45H（0100 0101B）。执行指令：ORL A，R0，

执行结果：（A）＝E7H（1110 0111B）。

逻辑“或”指令将两个指定的操作数按位进行逻辑“或”操作。它常用来使字节中某些位置“1”，欲保留（不变）的位用“0”与该位相或，而欲置位的位则用“1”与该位相或。

例 若（A）=C0H，（R0）=3FH，（3F）=0FH，执行指令

ORL A，@R0 ；（A）←（A）∨（（R0））

结果为：（A）=CFH。

3.6 逻辑运算指令

三、逻辑异或指令（6条）

助记符为XRL，用符号“⊕”表示，运算规则为：0⊕0=0，1⊕1=0，0⊕1=1，1⊕0=1

XRL	A，Rn	；	（A）←（A）⊕（Rn）
XRL	A，drect	；	（A）←（A）⊕（direct）
XRL	A，@Ri	；	（A）←（A）⊕（（Ri））
XRL	A，#data	；	（A）←（A）⊕＃data
XRL	direct，A	；	（direct）←（direct）⊕（A）
XRL	direct，＃data	；	（direct）←（direct）⊕＃data

功能：将目的地址单元中的数和源地址单元中的数按“位”相“异或”，其结果放回目的地址单元中。

前4条指令的功能是把源操作数与累加器A的内容异或，结果送入累加器A中。 后2条指令的功能是把源操作数与直接地址指示的单元内容异或，结果送入直接地址指示的单元。

例 若（A）=C3H，（R0）=AAH，执行指令 XRL A，R0 之后，（A）=69H。

3.6 逻辑运算指令

例 设（A）=A3H（1010 0011B），（R0）=45H（0100 0l0lB）。执行指令为 XRL　A，R0

$$\begin{array}{rr} & 1\,0\,1\,0\,0\,0\,1\,1 \\ \oplus & 0\,1\,0\,0\,0\,1\,0\,1 \\ \hline & 1\,1\,1\,0\,0\,1\,1\,0 \end{array}$$

执行结果为（A）＝E6H（11100110B）。

逻辑“异或”指令常用来使字节中某些位进行取反操作，其它位保持不变。若某位需要取反则该位与“1”相异或；保留某位则该位与“0”相“异或”。利用“异或”指令对某单元自身“异或”，可以实现清零操作。也可以判两个数是否相等，若相等则结果为0。

例　若（A）=B5H=10110101B，执行下列指令：

```
XRL  A，＃0F0H ；A的高4位取反，低4位保留
MOV  30H，  A   ；（30H）←（A）= 45H
XRL  A，30H    ；自身异或使A清零
```

执行后结果：（A）=00H。

3.6 逻辑运算指令

逻辑“与”、“或”、“异或”共同点:

(1) 逻辑“与”ANL、“或”ORL、“异或”XRL运算指令除逻辑操作功能不同外，三者的寻址方式相同，指令字节数相同，机器周期数相同。

(2) ANL、 ORL、 XRL的后2条指令的目的操作数均为直接地址方式，可很方便地对内部RAM的00H～FFH任一单元或特殊功能寄存器的指定位进行清零、 置位、 取反、 保持等逻辑操作。

(3) ANL、 ORL、 XRL的前4条指令，其逻辑运算的目的操作数均在累加器A中，且逻辑运算结果保存在A中。

3.6 逻辑运算指令

逻辑运算指令综合举例：

例3-12 试分析下列程序执行结果

```
MOV A,＃0FFH   ；(A)＝0FFH
ANL P1，＃00H  ；SFR中P1口清零
ORL P1，＃55H  ；P1口内容为55H
XRL P1，A      ；P1口内容为0AAH
```

例3-13 根据累加器A中4～0位的状态，用逻辑与、 或指令控制P1口4～0位的状态， P1口的高3位保持不变。

解：

```
ANL  A，＃00011111B  ；屏蔽A的高3位
ANL  P1，＃11100000B ；保留P1的高3位
ORL  P1， A          ；使P1口4～0按A4～0置位
```

若上述程序执行前：（A）=B5H=10110101B，（P1）=6AH=01101010B，则执行程序后：

（A）=15H=00010101B，（P1）=75H=01110101B。

3.6 逻辑运算指令

例3-14 双字节数求补码。

解: 对于一个 16 位数, R3 存高 8 位, R2 存低 8 位, 求补结果仍存R3、 R2。 求补的参考程序如下:

```
MOV   A，  R2        ；低 8 位数送A
CPL   A              ；低 8 位数取反
ADD   A, ＃01H       ；加 1 得低 8 位数补码
MOV   R2，A          ；存补码低 8 位
MOV   A，R3          ；高 8 位数送A
CPL   A              ；高 8 位取反
ADDC  A，＃00H       ；加低 8 位进位
MOV   R3，A          ；存补码高 8 位
```

3.6 逻辑运算指令

例3-15 已知外部RAM 30H中有数AAH，现欲令它高4位不变和低4位取反，试编写相应程序。

解：完成本题有多种求解方法，现介绍其中两种。

（1）利用MOVX A，@Ri 类指令

```
ORG    0100H          ；定位程序的起始地址
MOV    R0，#30H       ；地址30H送R0
MOVX   A，@R0         ；（A）←AAH
XRL    A，#0FH        ；（A）←AAH⊕0FH=A5H
MOVX   @R0，A         ；送回30H单元
SJMP   $              ；等待
END                   ；汇编程序结束
```

（2）利用MOVX A， @DPTR 类指令

```
ORG    0200H          ；定位程序的起始地址
MOV    DPTR，#0030H   ；地址0030H送DPTR
MOVX   A，@DPTR       ；（A）←AAH
XRL    A，#0FH        ；（A）←AAH⊕0FH=A5H
MOVX   @DPTR，A       ；送回30H单元
SJMP   $              ；等待
END                   ；汇编程序结束
```

3.6 逻辑运算指令

例3-16 编写程序完成下列各题：

（1）选用工作寄存器组中0区为工作区。

（2）利用移位指令实现累加器A的内容乘6。

解：程序如下：

```
（1） ANL  PSW，#11100111B ；PSW的D4、D3位为00
（2） CLR   C
      RLC  A            ；左移一位，相当于乘2
      MOV  R0，A
      CLR  C
      RLC  A            ；再乘2，即乘4
      ADD  A，R0        ；乘2 + 乘4 = 乘6
```

3.6 逻辑运算指令

数据传送类指令中累加器A的内容半字节交换指令：

SWAP A，实际上相当于执行循环左移指令 4 次。该指令在BCD码的变换中很有用。

例3-17 将累加器A中压缩BCD码分为两个字节，形成非压缩BCD码，放入30H和31H单元中。

解：程序如下：

```
MOV  40H，A            ；保存A中的内容
ANL  A，#00001111B     ；清高4位，保留低4位
MOV  30H，A
MOV  A，40H            ；取原数据
ANL  A，#11110000B     ；保留高4位，清低4位
SWAP A
MOV  31H，A
```

逻辑或运算指令用做指定位强迫置位。给某些位置1，合并两个数中的“1”。利用“或”操作可进行数位的组合。

3.6 逻辑运算指令

例3-18 编写程序将累加器A中的低4位从P1口的低4位输出，P1口的高4位不变。

解：程序如下：

```
ANL   A，#00001111B
MOV   30H，A        ；保留A中的低4位
MOV   A，P1
ANL   A，#11110000B  ；P1的高4位不变
ORL   A，30H
MOV   P1，A
```

例3-19 在30H与31H单元有两个非压缩BCD码（高位在30H单元），编程将它们合并到30H单元以节省内存空间。

解：程序如下：

```
MOV   A，30H
SWAP  A             ；（A）7~4↔（30H）3~0
ORL   A，31H        ；合并为压缩BCD码
MOV   30H，A        ；回存到30H单元
```

3.6 逻辑运算指令

例3-20 数据的拆分与拼装。

要求：从（30H）=X7X6X5X4X3X2X1X0中取出高5位，从（31H）=Y7Y6Y5Y4Y3Y2Y1Y0中取出低3位，拼装后存入40H中，（40H）=Y2Y1Y0X7X6X5X4X3。

解：程序如下：

```
        ORG   0000H
        MOV  A，30H
        SWAP  A          ；X3X2X1X0  X7X6X5X4
        RL  A            ；X2X1X0  X7X6X5X4X3 左移了5位
        MOV  40H，A
        ANL  40H，#00011111B
        MOV  A，31H
        MOV  B，#20H
        MUL  AB          ； Y2Y1Y0 0 0 0 0 0左移了5位
        ANL  A，#11100000B
        ORL  40H，A
HERE：  AJMP  HERE
```

注：实现左移5位，采用了两种方法，即移位和乘法。