3.4 数据传送指令

数据传送指令共有29条。51单片机中的传送指令有从右向左传送数据的约定，即指令的右边操作数为源操作数，表达的是数据的来源；而左边操作数为目的操作数，表达的则是数据的去向。数据传送指令的特点为：把源操作数传送到目的操作数，指令执行后，源操作数不改变，目的操作数修改为源操作数。

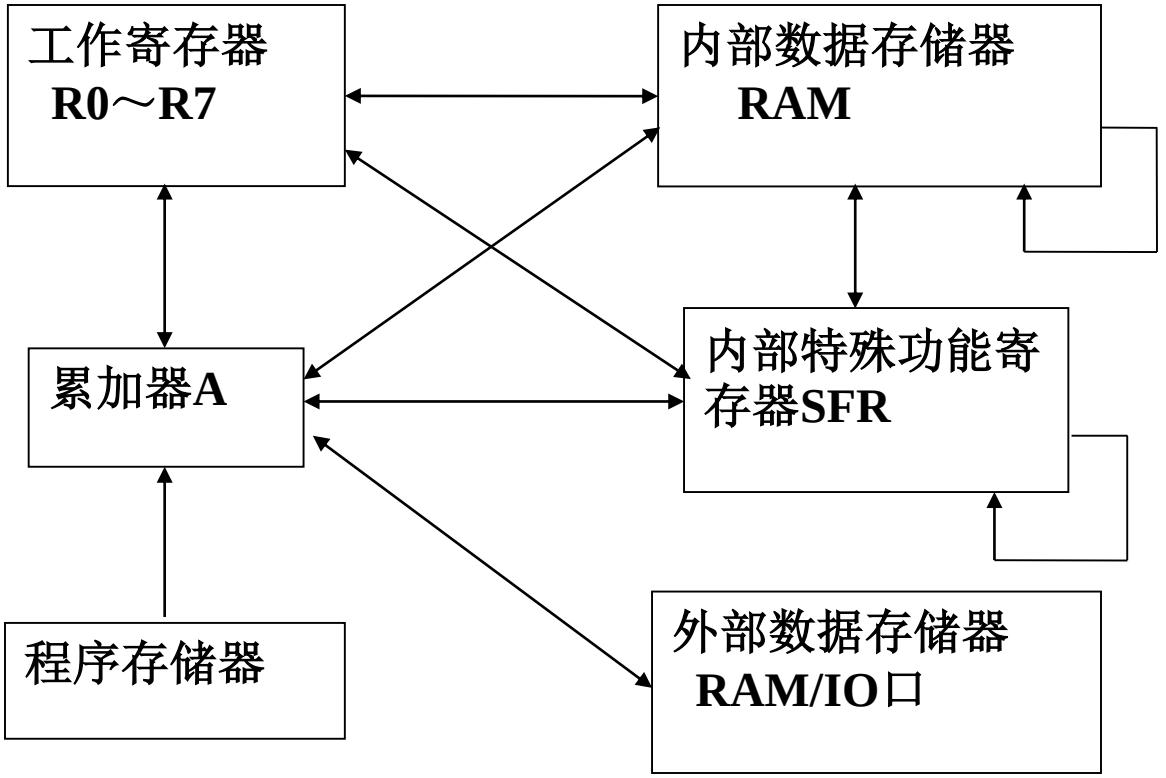
数据传送指令主要用于在单片机片内RAM和特殊功能寄存器SFR之间传送数据，也可以用于在累加器A和片外存储单元之间传送数据。交换指令也属于数据传送指令，是把两个地址单元中的内容相互交换。

数据传送类指令不影响标志位（进位标志CY、半进位标志AC和溢出标志OV），但当传送或交换数据后影响累加器A的值时，奇偶标志位P的值则按A的值重新设定。

传送类指令占有较大的比重。数据传送是进行数据处理的最基本的操作，可分为内部8位数据传送指令、16位数据传送指令、外部数据传送指令、程序存储器数据传送指令、交换指令和堆栈操作指令。

3.4 数据传送指令

助记符8种：MOV、MOVX、MOVC、XCH、XCHD、SWAP、PUSH、POP。



数据传送操作

3.4 数据传送指令

- ◆源操作数可采用寄存器、寄存器间接、直接、立即、寄存器基址加变址等5种寻址方式。
- ◆目的操作数可以采用寄存器、寄存器间接、直接等3种寻址方式。

一般操作是把源操作数传送到目的操作数，指令执行后源操作数不变，目的操作数被修改为源操作数。

- ◆若要求进行数据传送时，目的操作数不变，则可以用交换指令。
- ◆数据传送类指令不影响标志位C、AC、OV。对于P标志一般不加以说明。

堆栈操作指令可以直接修改程序状态字PSW，这时可以使某些标志位发生改变。

3.4 数据传送指令

数据传送指令的特点：

⑴ 可以进行直接地址到直接地址的数据传送，能把一个并行I/O口中的内容传送到片内RAM单元中，而不必经过累加器或工作寄存器Rn。

⑵ 用R0和R1寄存器间址访问片外数据存储器256个字节址及片内RAM中的任一单元。用DPTR间址访问片外全部64KB的数据存储器或I/O。

⑶累加器A能对Rn寄存器寻址；能与特殊功能寄存器之间进行一个字节的数据传送；能对片内RAM直接寻址；能与片内RAM单元之间进行低半字节的数据交换。

⑷ 能用变址寻址方式访问程序存储器中的表格，将程序存储器单元中的固定常数或表格字节内容传送到累加器A中。

3.4 数据传送指令

3.4.1 内部数据传送指令

内部数据存储器RAM区是数据传送最活跃的区域，可用的指令数也最多，共有16条指令，指令操作码助记符主要为MOV。通用格式为：

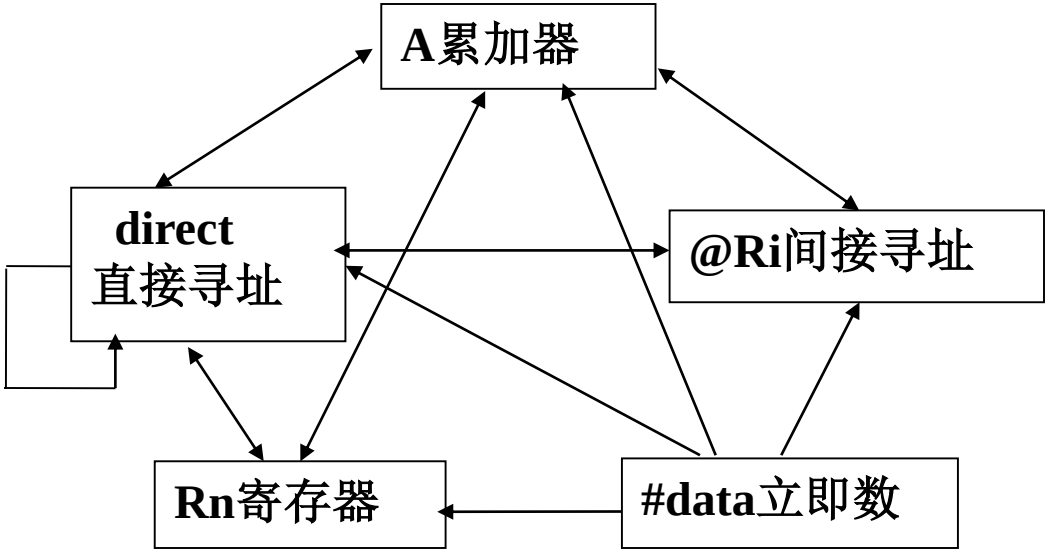
MOV＜目的操作数＞，＜源操作数＞

功能：把源操作数的内容送到目的操作数，而源操作数的内容不变。操作属于拷贝性质。不影响标志位。但当执行结果改变累加器A的值时，会使奇偶标志变化。

源操作数：累加器A，工作寄存器Rn，直接地址direct、间接寻址寄存器@Ri和立即数＃data等5种。

目的操作数：累加器A，工作寄存器Rn，直接地址direct和间接寻址寄存器@Ri等4种。

3.4 数据传送指令



内部RAM之间数据传递关系

在5种源操作数中，只有＃data不能用作目的操作数。可以用4种目的操作数为基础构造4类指令。相应的源操作数选择依据是：

•源操作数与目的操作数不相同（除direct外）；

•寄存器寻址及寄存器间接寻址间不相互传送。

传送指令是以累加器A为中心的总体结构。绝大部分传送操作均需通过A进行的。 下面以目的操作数分类，分别介绍。

3.4 数据传送指令

一、以累加器A为目的操作数的指令（4条）

MOV A，Rn ；（A）←（Rn），（n=0～7）

MOV A，direct ；（A）←（direct）

MOV A，@Ri ；（A）←（（Ri））（i=0、1）

MOV A，＃data ；（A）← data

这组指令的功能是把源操作数送入累加器A中。目的操作数都是累加器A，源操作数的寻址方式采用寄存器寻址、直接寻址、寄存器间接寻址和立即寻址四种基本寻址方式。

例：若（R1）= 20H，（20H）= 55H，执行指令MOV A，@R1 后，（A）= 55H。

3.4 数据传送指令

二、以工作寄存器Rn为目的操作数的指令（3条）

MOV Rn，A ；（Rn）←（A），（n=0～7）

MOV Rn，direct ；（Rn）←（direct ），（n=0～7 ）

MOV Rn，＃data ；（Rn）← data，（n=0～7）

这组指令的功能是把源操作数送入寄存器Rn中。都是以工作寄存器Rn为目的操作数，源操作数的寻址方式采用寄存器寻址、直接寻址和立即寻址。由于目的操作数为工作寄存器，所以源操作数不能是工作寄存器及其间址方式寻址。

例：若（50H）= 40H，执行指令 MOV R6，50H 后，（R6）= 40H。

3.4 数据传送指令

三、以直接地址为目的操作数的指令（5条）

MOV direct，A ；（direct）←（A）

MOV direct，Rn ；（direct）←（Rn），（n=0～7）

MOV directl，direct2 ；（direct1）←（direct2）

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

MOV direct，＃data ；（direct）←data

这组指令的功能是把源操作数的内容送入由直接地址指出的存储单元。

direct为8位直接地址，可寻址0～255个单元，对51单片机可直接寻址内部RAM 0～127个地址单元和128～255地址的特殊功能寄存器。目的操作数都是直接寻址单元，源操作数采用寄存器寻址、直接寻址、寄存器间接寻址和立即寻址。

例：若（R1）=50H,（50H）=18H，执行指令MOV 40H，@R1 后，（40H）=18H。

3.4 数据传送指令

四、以寄存器间接寻址的单元为目的操作数的指令（3条）

MOV @Ri，A ；（（Ri））←（A）

MOV @Ri，direct ；（（Ri））←（direct）

MOV @Ri，＃data ；（（Ri））←data

这组指令的功能是把源操作数的内容送入由 **R0**或**R1**的内容为地址所指的内部 **RAM**中的存储单元。目的操作数都是间接寻址单元，源操作数可采用寄存器寻址、直接寻址和立即寻址方式。因目的操作数采用寄存器间接寻址，故源操作数不能是寄存器及其间址寻址。**Ri**由操作码字节的最低位来选定是**R0**还是**R1**寄存器，间址是以**Ri**的内容作为操作数的地址来进行寻址的。

例：若（**R1**）=30H，（**A**）=20H，执行指令

MOV @R1，A 后，（**30H**）=20H。

3.4 数据传送指令

内部8位数据传送指令小结：

- 直接寻址direct单元在编程时就已明确，而间接寻址单元是在程序进行中明确的，间接寻址空间和直接寻址空间范围相同，均为0～255个单元地址。
- 立即数＃data为一常数，它是不带符号的8位二进制数。

例如，MOV A，80H ；表示把片内RAM中地址为80H单元（即P0口）中的内容送A。

MOV 80H，＃88H；把立即数88H送到片内RAM中的80H地址单元中去。

MOV 80H，0E0H ；表示把E0H单元的内容送到80H单元中去。这是片内数据存储单元中的直接地址单元之间数据的直接传送 。

又如，下列指令是非法的、错误的。

× MOV Rn，@Ri ；寄存器不能同寄存器间址互传数据

× MOV #data，A ；立即数不能作目标操作数

3.4 数据传送指令

内部8位数据传送指令综合举例

例3-1 将片内RAM的15H单元的内容0A7H送55H单元。

解法1： MOV 55H，15H

解法2： MOV R6，15H
MOV 55H，R6

解法3： MOV R1，＃15H
MOV 55H，@R1

解法4： MOV A，15H
MOV 55H，A

例3-2 若（R0）=30H，片内RAM（30H）=57H，片内RAM（40H）=7FH，试比较：

MOV A，R0 和 MOV A，@R0

MOV A，#40H 和 MOV A，40H

执行后的结果。

解：它们的执行结果为：

MOV A，R0 ； （A）=30H

MOV A，@R0 ； （A）=57H

MOV A，#40H ； （A）=40H

MOV A，40H ； （A）=7FH

3.4 数据传送指令

例3-3 内部RAM中（70H）=60H，（60H）=20H，若P1口输入的数据为#0B7H，执行下列程序段

```
MOV  R0，#70H
MOV  A，@R0
MOV  R1，#60H
MOV  B，  @R1
MOV  @R0，P1
MOV   P3，P1
```

后的结果如何？

解：运行结果为：（P3）= 0B7H，（70H）= 0B7H，（A）=60H，（B）=20H，（R1）=60H，（R0）=70H 。

例3-4 编写把30H单元和40H单元中的内容进行交换的程序。

解：分析：30H和40H单元中都装有数据，要想把其中的内容相交换必须寻求第三个存储单元对其中的一个数进行缓冲，这个存储单元若选为累加器A，则相应程序如下：

3.4 数据传送指令

```
MOV  A，30H   ；   （A）←（30H）
MOV  30H，40H ；   （30H）←（40H）
MOV  40H，A   ；   （40H）←（A）
```

例3-5 检查传送结果。已知内部RAM（10H）=00H，（30H）=40H，（40H）=10H，P1口为11001010B，分析指令执行后各单元内容。

```
MOV  R0，#30H    ；（R0）=30H
MOV  A，@R0      ；（A）=40H
MOV  R1，A       ；（R1）=40H
MOV  B，@R1      ；（B）=10H
MOV  @R1，P1     ；（40H）=11001010B
MOV  P2，P1      ；（P1）=（P2）=11001010B
MOV  10H，#20H   ；（10H）=20H
```

执行结果：（10H）=20H，（30H）=40H，（40H）=CAH，（P1）=（P2）=CAH，（A）=40H，（B）=10H，（R0）=30H，（R1）=40H。

3.4 数据传送指令

五、16位数据传送指令（1条）

MOV DPTR，＃data16 ；（DPTR）←data16

这条指令的功能是将源操作数data16（通常是地址常数）送入目的操作数DPTR中。即将16位立即数送入DPTR，高8位送入DPH，低8位送入DPL。源操作数的寻址方式为立即寻址。16位常数在指令的第二、第三字节中（第二字节为高位字节DPH，第三字节为低位字节DPL）。

例如，执行指令 MOV DPTR，#1234H 后，

（DPH）=12H，（DPL）=34H。又如，执行

MOV DPH，#35H

MOV DPL，#12H

就相当于执行了 MOV DPTR，#3512H。

3.4 数据传送指令

六、堆栈操作指令

堆栈是在内部RAM中按“后进先出”的规则组织的一片存储区。此区的一端固定，称为栈底；另一端是活动的，称为栈顶。栈顶的位置（地址）由栈指针SP指示（即SP的内容是栈顶的地址）。51单片机中，堆栈的生长方向是向上的（地址增大）。系统复位时，SP的内容为07H。通常用户应在系统初始化时对SP重新设置。SP的值越小，堆栈的深度越深。堆栈主要用于保护断点和恢复现场。堆栈操作有进栈和出栈操作，即压入和弹出数据。

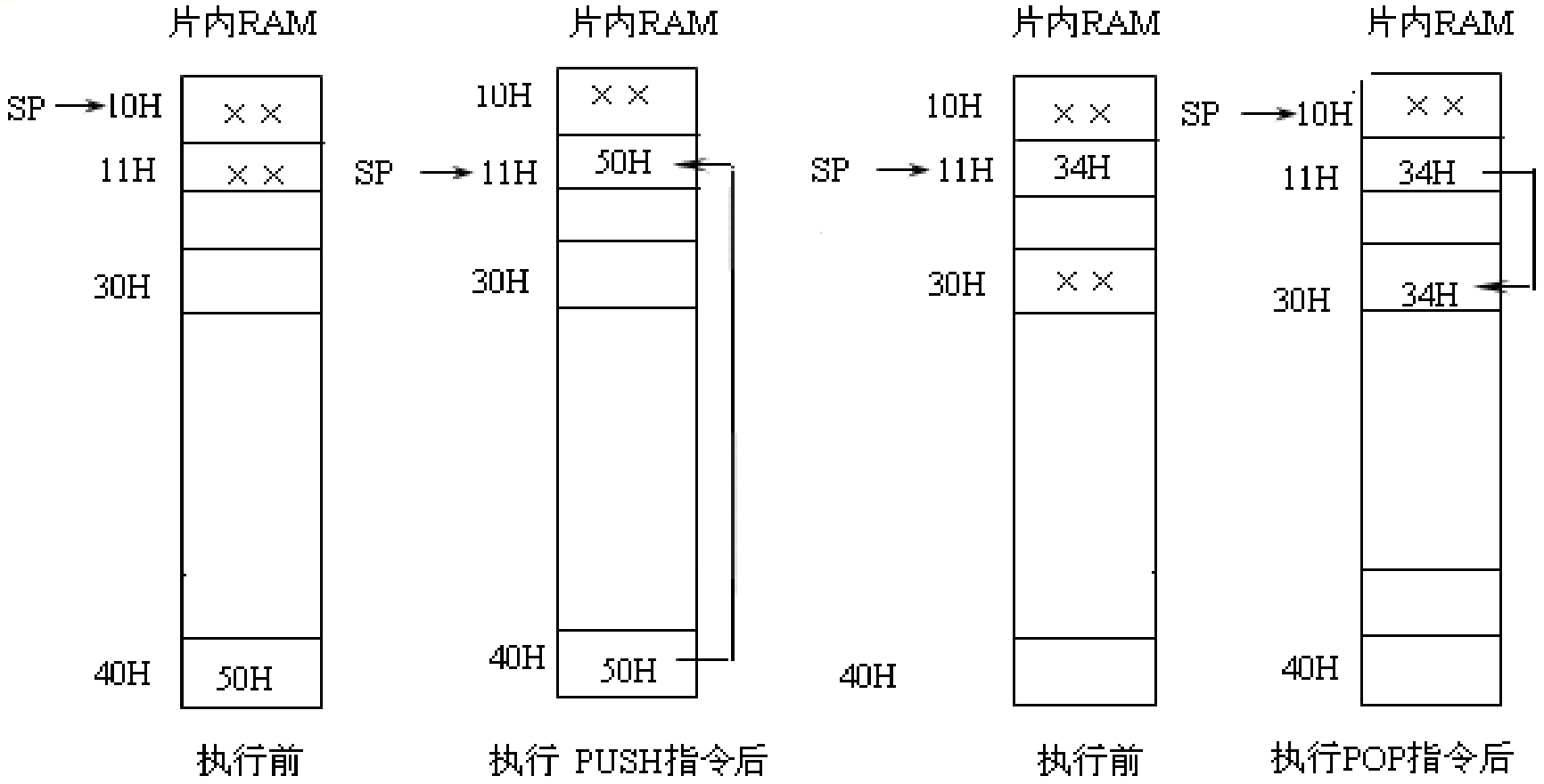
进栈指令　PUSH　direct ；SP ←（SP）＋ 1，（SP）←（direct）

退栈指令　POP　direct ；direct←（（SP）），SP ←（SP）－1

功能：（1）PUSH称为压栈指令，将指定的直接寻址单元的内容压入堆栈。先将堆栈指针SP的内容+1，指向栈顶的一个单元，然后把指令指定的直接寻址单元内容送入该单元。

（2）POP称为出栈指令，它是将当前堆栈指针SP所指示的单元内容弹出到指定的内部RAM单元中，然后再将SP减1。

3.4 数据传送指令



指令PUSH 40H操作示意图　　指令POP 30H操作示意图

堆栈指令执行过程

3.4 数据传送指令

注意：堆栈操作的特点是“先进后出”，在使用时应注意指令顺序；进栈、出栈指令只能以直接寻址方式来取得操作数，不能用累加器A或工作寄存器Rn作为操作数。

例 若（SP）=07H，（40H）=88H，执行指令 PUSH 40H 后，

（SP）=08H，（08H）=88H。

例 中断响应时（SP）＝30H，DPTR的内容为0123H，执行入栈指令：

```
PUSH    DPL     ；DPL内容入栈
PUSH    DPH     ；DPH内容入栈
```

执行结果：第1条指令

（SP）+1＝31H→（SP），（DPL）＝23H→（31H）

第2条指令（SP）+1＝32H→（SP），（DPH）＝01H→（32H）

片内RAM中，

（31H）＝23H，（32H）＝01H，（SP）＝32H。

3.4 数据传送指令

例 设（SP）＝32H，片内RAM的30H~32H单元中的内容分别为20H，23H，01H，执行下列指令的结果怎样？

```
POP  DPH    ；（（SP））＝（32H）＝01H→DPH
                （SP）－1＝32H－1＝31H→SP
POP  DPL    ；（（SP））＝（31H）＝23H→DPL
                （SP）－1＝31H－1＝30H→SP
```

例 分析以下程序的运行结果

```
MOV  R2，＃05H
MOV  A，＃01H
PUSH  ACC    ；ACC表示累加器A的直接地址
PUSH  02H    ；02H表示R2的内部单元的直接地址
POP  ACC
POP  02H     ；02H表示R2的内部单元地址
```

结果：（R2）=01H，（A）=05H。也就是两者进行了数据交换。因此，使用堆栈时，入栈的顺序和出栈的顺序必须相反，才能保证数据被送回原位，即恢复现场。

3.4 数据传送指令

七、字节交换指令（XCH，Exchange 3条）

对于单一的MOV类指令，传送通常是单向的，即数据是从一处（源）到另一处（目的）的拷贝。而交换类指令完成的传送是双向的，是两字节间或两半字节间的双向交换 。数据交换指令分为字节交换指令和半字节交换指令。字节交换指令：

XCH A，Rn ；（A）↔（Rn）

XCH A，@Ri ；（A）↔（（Ri））

XCH A，direct ；（A）↔（direct）

功能：将累加器A的内容与源操作数（Rn、direct或@Ri）所指定单元的内容相互交换。

例 若（R0）=80H，（A）=20H。执行指令 XCH A，R0 后，

（A）=80H，（R0）=20H。

例 设（R0）＝30H，（A）＝3FH，片内（30H）＝BBH。

执行指令 XCH A，@R0

执行结果 （A）=BBH，（30H）＝3FH。

3.4 数据传送指令

八、半字节交换指令（1条）

XCHD A, @Ri ；（A）$_{3\sim0}$↔（Ri）$_{3\sim0}$

功能：将Ri间接寻址单元的低4位内容与累加器A的低4位内容互换，而它们的高4位内容均不变。例 设（R0）＝20H，（A）＝36H（00110110B），（20H）＝75H（0111010lB）。

执行指令：XCHD A，@R0

结果为：（20H）＝01110110B＝76H，（A）＝00110101B＝35H。

例 设（A）=0ABH，（R0）=30H，（30H）=12H，执行指令

XCHD A，@R0后，（A）=A2H，（30H）=1B。

九、累加器A高低半字节交换指令（1条）

SWAP A ；（A）$_{7\sim4}$↔（A）$_{3\sim0}$

功能：将累加器A的高4位与低4位内容互换，不影响标志位。该操作也可看作是4位循环指令。

例 设（A）＝36H（0011 0110B）

执行指令： SWAP A

执行结果：（A）＝63H（0110 0011B）。

例 若（A）＝30H，执行指令SWAP A后，（A）＝03H。

3.4 数据传送指令

数据交换指令小结：

- 数据交换在内部RAM单元与累加器A之间进行。可以保存目的操作数。

  例 将片内RAM 60H单元与61H单元的数据交换

  不能用： XCH 60H，61H

  应该写成：MOV A，60H

  XCH A，61H

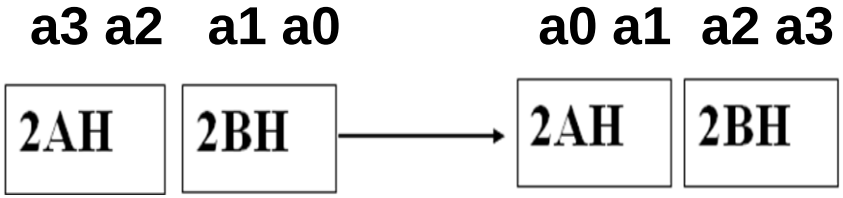
  MOV 60H，A

- XCHD和SWAP指令主要用于实现十六进制数或BCD码的数位交换。

3.4 数据传送指令

数据交换指令综合举例：

例3-6 将4位BCD码倒序。设内部数据存储器2AH、2BH单元连续存放有4位BCD码数符，试编一程序把4位BCD码数符倒序排列。



解：程序如下

```
        MOV   R0，#2AH
        MOV   R1，#2BH
        MOV   A，@R0      ；2AH单元内容送A
        SWAP  A           ；A的高4位与低4位交换（a2 a3）
        MOV   @R0，A
        MOV   A，@R1      ；2BH单元内容送A
        SWAP  A           ；A的高4位与低4位交换（a0 a1）
        XCH   A, @R0      ；2AH与2BH单元内容交换
        MOV   @R1， A
HERE:   SJMP  HERE
```

3.4 数据传送指令

例3-7 已知外部RAM 20H单元中有一个数X，内部RAM 20H单元中有一个数Y，试编出可以使它们互相交换的程序。

解：分析：本题是一个字节交换问题，故可以采用三条字节交换指令中的任何一条。若采用第三条字节交换指令，则相应程序为：

```
MOV   R1，#20H   ；（R1）←20H
MOVX  A，@R1     ；（A）←X
XCH   A，@R1     ；（20H）←X，（A）←Y
MOVX  @R1，A     ；Y→（20H）（片外RAM）
```

例3-8 已知50H中有一个0～9的数，请使用交换指令编程把它变成相应的ASCII码程序。

解：分析：0～9的ASCII码为30H～39H。进行比较后发现，二者之间仅相差30H，故可以利用半字节指令把0～9的数装配成相应的ASCII码。程序如下：

```
MOV   R0，  #50H  ；（R0）←50H
MOV    A，  #30H  ；（A）←30H
XCHD  A，  @R0    ；A中形成相应的ASCII码
MOV    @R0，A     ；ASCII码送回50H单元
```

3.4 数据传送指令

3.4.2 累加器A与外部数据存储器传送指令

在51系列单片机中，与外部存储器RAM或I/O端口之间进行数据交换的只可以是累加器A。即所有片外RAM或者I/O端口数据传送必须通过累加器A进行。指令助记符为MOVX，其中的X表示外部（External）。

```
MOVX  A，@Ri      ；（A）←（（Ri））
MOVX  @Ri，A      ；（（Ri））←（A）
MOVX  A，@DPTR ；（A）←（（DPTR））
MOVX  @DPTR，A ；（（DPTR））←（A）
```

要点分析：

（1）实现累加器A与外部数据存储器或I/O口之间传送一个字节数据的指令。

（2）要访问片外RAM，必须要知道RAM单元的地址，采用间接寻址方式访问外部数据存储器，有Ri和DPTR两种间接寻址方式。使用访问外部RAM或I/O口数据传送指令时，应当首先将要读或写的地址送入DPTR或Ri中，然后再用读或者写命令。

3.4 数据传送指令

（3）采用R0或R1作间址寄存器时，可寻址256个外部数据存储器单元，8位地址和数据均由P0口分时输入和输出。这时若要访问大于256个单元的片外RAM时，也可以由P2与R0或P2与R1组成16位地址指针，寻址外部RAM的64KB空间。

（4）采用16位DPTR作间址可寻址整个64KB片外数据存贮空间，低8位（DPL）由P0口进行分时使用，高8位（DPH）由P2口输出。

例 设工作寄存器R0的内容为12H，R1的内容为34H，片外RAM 34H单元的内容为56H。

执行指令：

MOVX A，@R1；片外（34H）＝56H→A

MOVX @R0，A；（A）→片外12H单元中

执行结果：片外RAM的（34H）＝56H，（12H）＝56H。

3.4 数据传送指令

累加器A与外部数据存储器传送指令综合举例：

例3-9 把内部RAM的20H单元内容送到外部RAM的800H单元中。

解：程序如下：

```
MOV   DPTR，#800H    ；外部数据存储器地址指针
MOV   R0，#20H       ；内部数据存储器地址指针
MOV   A，@R0         ；取内部RAM20H单元内容
MOVX  @DPTR，A       ；送外部RAM800H单元
```

3.4 数据传送指令

例3-10 将外部RAM中0010H单元中的内容送入外部RAM中2000H单元中。

解一：程序如下：

```
MOV  R0，＃10H
MOVX  A，@R0
MOV  DPTR，＃2000H
MOVX  @DPTR，A
```

解二：程序如下：

```
MOV   P2，＃00H
MOV   R0，＃10H
MOVX  A，@R0
MOV   DPTR，＃2000H
MOVX  @DPTR，A
```

例3-11 将外部存储器2000H单元的内容送入2100H单元。

解：程序如下：

```
MOV   DPTR，#2000H   ；
                （DPTR）← 2000H
MOVX  A，@DPTR       ；
                （A）←（（DPTR））
MOV   DPTR，#2100H   ；
                （DPTR）← 2100H
MOVX  @DPTR，A       ；
                （（DPTR））←（A）
```

3.4 数据传送指令

3.4.3 查表指令

这类指令共有两条，均属于变址寻址指令，因专门用于从ROM中查找数据而又称为查表指令。

指令助记符为：MOVC，其中的C表示代码（Code）。

一、DPTR内容为基址

MOVC A，@A+DPTR ；A←（（A）+（DPTR））

该指令首先执行16位无符号数加法，将获得的基址与变址之和作为16位的程序存储器地址，然后将该地址单元的内容传送到累加器A。指令执行后DPTR的内容不变。

二、PC内容为基址

MOVC A，@A+PC ；A←（（A）+（PC））

取出该单字节指令后PC的内容增1，以增1后的当前值去执行16位无符号数加法，将获得的基址与变址之和作为16位的程序存储器地址。然后将该地址单元的内容传送到累加器A。指令执行后PC的内容不变。

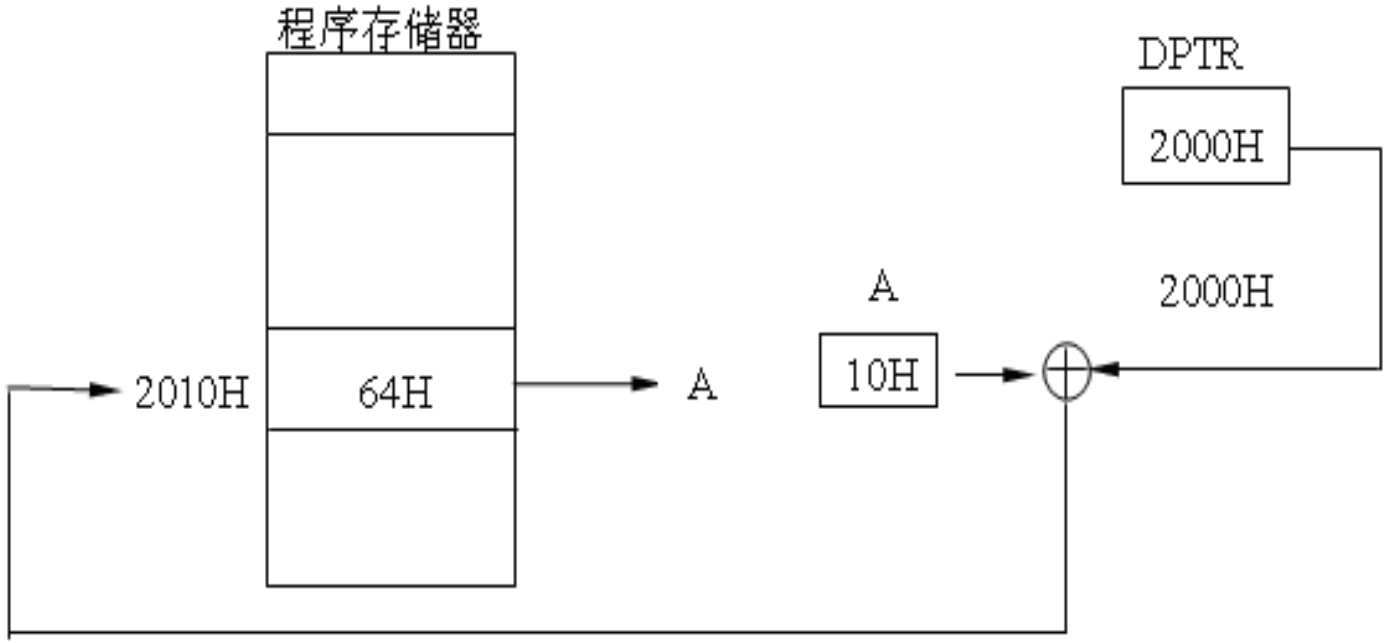
3.4 数据传送指令

要点分析：

1）累加器A为变址寄存器，而PC、DPTR为基址寄存器。 这两条指令寻址范围为64KB，指令首先执行16位无符号数的加法操作，获得基址与变址之和，“和”作为程序存储器的地址，该地址中的内容送入A中。两条MOVC是64KB存储空间内的查表指令，实现程序存储器到累加器的常数传送，每次传送一个字节。

2）DPTR为基址寄存器时，指令的执行结果只与指针DPTR及累加器A的内容有关，与该指令存放的地址无关。因此，表格的大小和位置可以在64KB程序存储器中任意安排，并且一个表格可以为各个程序块所共用。允许数表存放在程序存储器的任意单元，称为远程查表。

3.4 数据传送指令



指令执行示意图

3.4 数据传送指令

3）PC为基址寄存器时，数表只能放在当前PC往下的255个单元中，称为近程查表。由于PC的内容不能通过数据传送指令来改变，而且随该指令在程序中的位置变化而变化，因此在使用时需对变址寄存器A进行修正。编程时需计算A值与数表首址的偏移量。这条指令的优点是不改变特殊功能寄存器和PC的状态，只要根据A的内容就可以取出表格中的常数。缺点是表格只能放在该条查表指令后面的255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。

例 求平方数（远程查表法）

```
        MOV  DPTR，  #TABLE      ；指向表首址
        MOVC  A，@A+DPTR         ；查表得到平方数
        MOV  20H，A              ；存平方数
HERE:   SJMP  HERE
TABLE   DB  00H，01H，04H，09H   ；平方表0²～9²
        DB  16H，25H，36H
        DB  49H，64H，81H
```

3.4 数据传送指令

例 求平方数（近程查表 ）

```
        ADD  A，#rel          ；修正偏移量
        MOVC  A，@A+PC        ；查表得到平方数
        MOV  20H，A           ；存平方数
HERE：SJMP  HERE
TABLE DB  00H，01H，04H，09H，16H   ；平方表0²～9²
        DB  25H，36H，49H，64H，81H
```

注： rel=TABLE-（查表指令地址+1）；MOVC指令为单字节。