本章分为七节，主要介绍：

3.1 指令格式及常用符号

3.1.1 机器指令的编码格式

一、单字节指令

1、8位编码仅为操作码：

位号	7 6 5 4 3 2 1 0	
字节	opcode	注：opcode 表示操作码。

如：**INC A。**该指令的编码为：**0000 0100B**，其十六进制表示为**04H**，累加器**A**隐含在操作码中。指令的功能是累加器**A**的内容加**1**。

注意：在指令中用“**A**”表示累加器，而用“**ACC**”表示累加器对应的地址（**E0H**）。

2．8位编码含有操作码和寄存器编码

位号	7 6 5 4 3 2 1 0	
字节	opcode | r r r	注：r r r 表示寄存器编码。

高5位为操作码，低3位为存放操作数的寄存器编码。如：MOV A，R0

编码为1110 1000B，其十六进制表示为E8H（低3位000为寄存器R0的编码）。功能是将当前工作寄存器R0中的数据传送到累加器A中。

二、双字节指令

位号	7 6 5 4 3 2 1 0	
字节	opcode	
	data 或 direct	注：data 和 direct 表示操作数或其地址。

第一字节表示操作码，第二个字节表示参与操作的数据或数据存放的地址。

如：MOV A，#50H

编码为0111 0100B，0101 0000B。其十六进制表示为74H，50H。功能是将立即数“50H”传送到累加器A中。

三、三字节指令

位号	7 6 5 4 3 2 1 0
字节	opcode
	data 或 direct
	data 或 direct

指令的第一字节表示该指令的操作码，后两个字节表示参与操作的数据或数据存放的地址。如：MOV 20H，#50H

编码为0111 0101B，0010 0000B，0101 0000B。其十六进制表示为75H，20H，50H。功能是将立即数“50H”传送到内部RAM 的20H单元中。

3.1.2 符号指令的格式

一般格式为：

操作助记符 [目的操作数][，源操作数][；注释]

多数指令为两操作数指令；当指令操作数隐含在操作助记符中时，在形式上这种指令无操作数；另有一些指令为单操作数指令或三操作数指令。指令的一般格式中使用了可选择符号“[ ]”，包含的内容因指令的不同可以有或无。

在两个操作数的指令中，通常目的操作数写在左边，源操作数写在右边。

如：ANL A，＃40H

功能是将立即数“40H”同累加器A中的数进行“与”操作，结果送回累加器。

ANL为“与”操作的助记符，立即数“40H”为源操作数，累加器A为目的操作数。

注：在指令中，多数情况下累加器用“A”表示，仅在直接寻址方式中，用“ACC”表示累加器在SFR区的具体地址E0H。

试比较，指令MOV A，＃30H的机器码为74H、30H；而指令MOV ACC，＃30H的机器码为75H、E0H、30H。

3.1.3 符号指令及其注释中常用的符号

Rn（n=0~7）--当前选中的工作寄存器组中的寄存器R0~R7之一；

Ri（i=0，1）--当前选中的工作寄存器组中的寄存器R0或R1；

@ ----------间址寄存器前缀；

#data ----------8位立即数；

#data16---------16位立即数；

direct--------片内低128个RAM单元地址及SFR地址（可用符号名称表示）；

addr11----------11位目的地址；

addr16----------16位目的地址；

rel----------补码形式表示的8位地址偏移量，值在－128～＋127范围内；

bit----------片内RAM位地址、SFR的位地址（可用符号名称表示）；

/----------位操作数的取反操作前缀；

（×）---------- 表示 × 地址单元或寄存器中的内容；

← ----------将箭头右边的内容送入箭头左边的单元中。

3.2 80C51的寻址方式

- 是寻找操作数或指令的地址的方式。
- 80C51的寻址方式有七种。即：寄存器寻址、直接寻址、寄存器间接寻址、立即寻址、基址寄存器加变址寄存器变址寻址、相对寻址和位寻址。

若不特别声明，我们后面提到的寻址方式均指源操作数的寻址方式。

表 3.1 寻址方式所对应的寄存器和存储空间

序号	寻址方式		寄存器或存储空间
1	基本方式	寄存器寻址	寄存器 R0~R7，A、AB、DPTR 和 C（布尔累加器）
2		直接寻址	片内 RAM 低 128 字节、SFR
3		寄存器间接寻址	片内 RAM（@R0，@R1，SP） 片外 RAM（@R0，@R1，@DPTR）
4		立即寻址	ROM
5	扩展方式	变址寻址	ROM（@A+DPTR，@A+PC）
6		相对寻址	ROM（PC 当前值的+127～−128 字节）
7		位寻址	可寻址位（内部 RAM20H~2FH 单元的位和部分 SFR 的位）

注：前 4 种寻址方式完成的是操作数的寻址，属于基本寻址方式；变址寻址实际上是间接寻址的推广；位寻址的实质是直接寻址；相对寻址是指令地址的寻址。

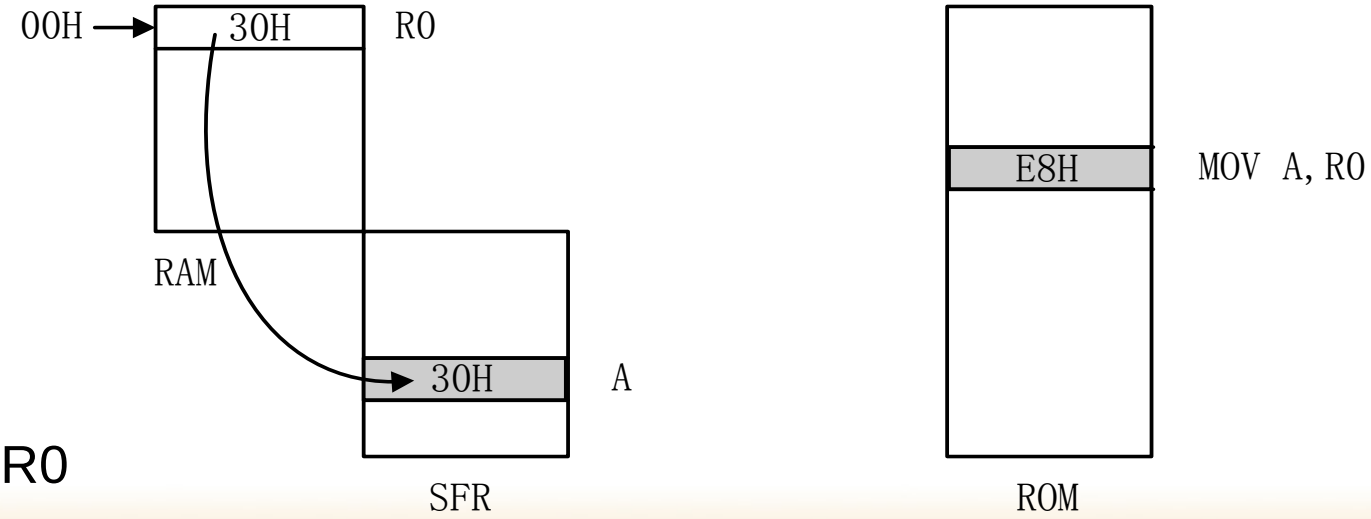
3.2.1 寄存器寻址

操作数存放在寄存器中，指令中直接给出该寄存器名称的寻址方式。可以获得较高的传送和运算速度。

寄存器可以是：R0~R7；A；B（以AB寄存器对形式出现）；DPTR。

如：

MOV A，R0



3.2.2 直接寻址

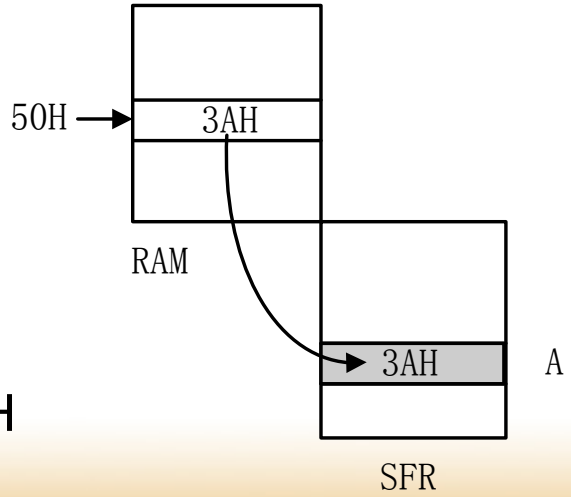
指令操作码之后的字节存放的是操作数的地址，操作数本身存放在该地址指示的存储单元中的寻址方式称为直接寻址。

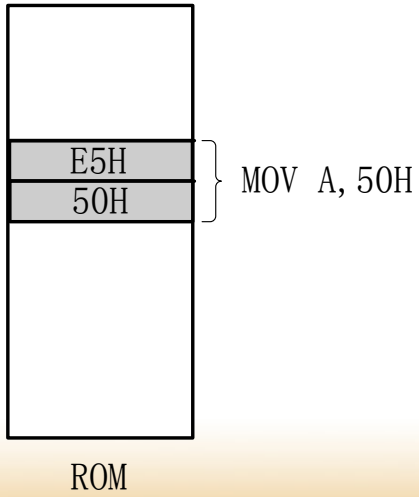
直接寻址中的SFR经常采用符号形式表示。

寻址空间为：片内RAM低128字节；SFR。

如：

MOV A，50H





3.2.3 寄存器间接寻址

寄存器中的内容为地址，从该地址去取操作数的寻址方式称为寄存器间接寻址 。

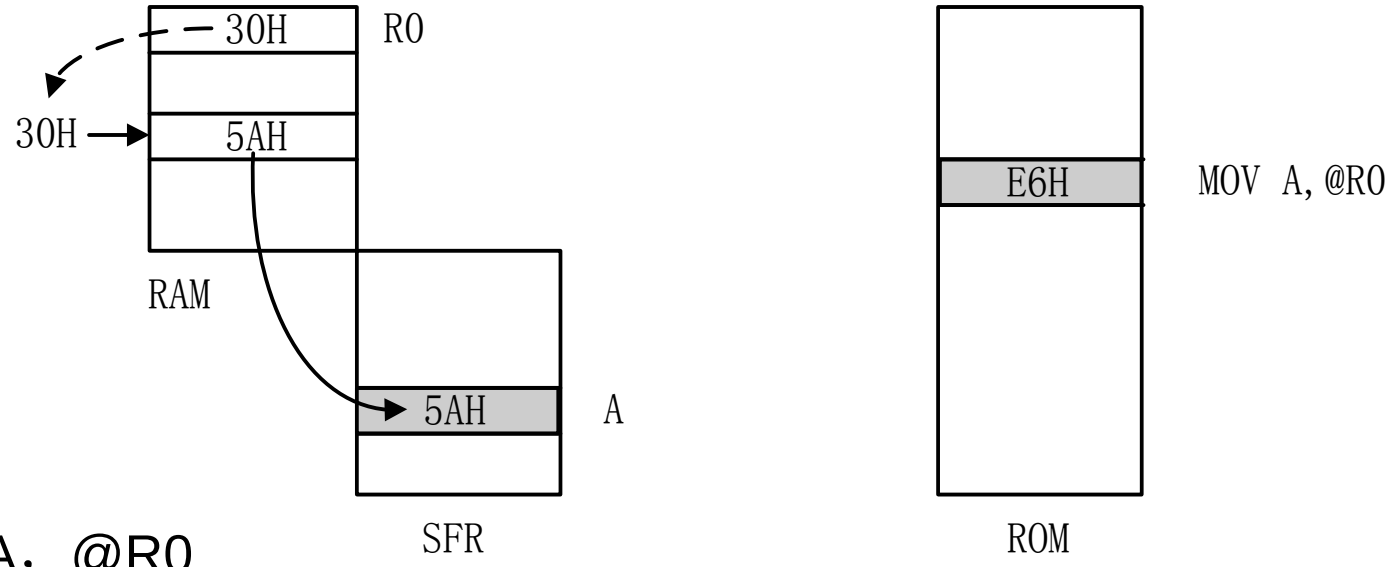
寻址的存储空间为片内RAM或片外RAM。

- 片内RAM的数据传送采用“MOV”类指令，间接寻址寄存器采用寄存器R0或R1（堆栈操作时采用SP）；

- 片外RAM的数据传送采用“MOVX”类指令，这时间接寻址寄存器有两种选择，一是采用R0和R1作间址寄存器，这时R0或R1提供低8位地址（外部RAM多于256字节采用页面方式访问时，可由P2口未使用的I/O引脚提供高位地址）；二是采用DPTR作为间址寄存器。

寄存器间接寻址对应的空间为：

- 片内RAM（采用@R0，@R1或SP）；
- 片外RAM（采用@R0，@R1或@DPTR）。



如：

MOV A，@R0

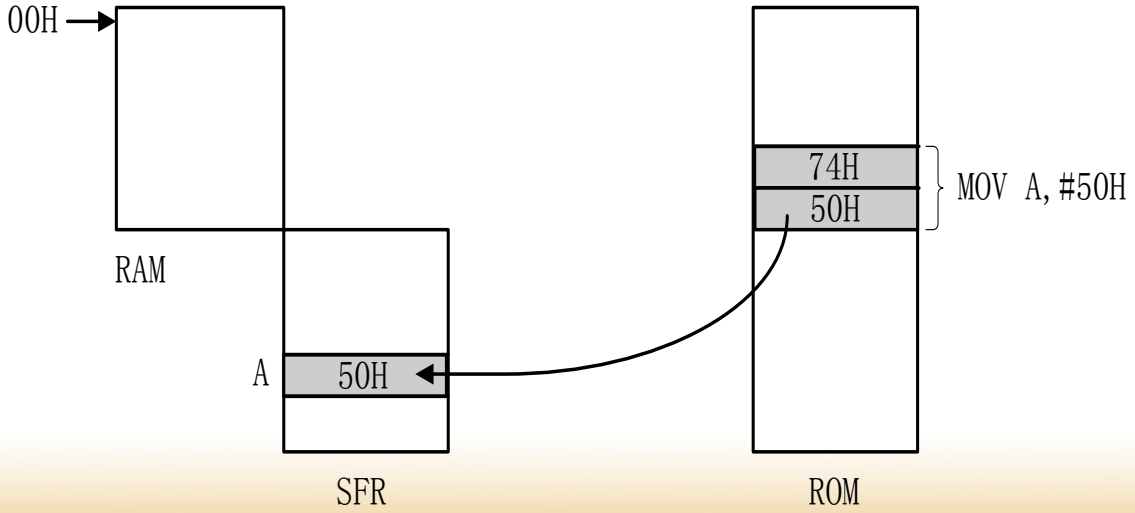
3.2.4 立即寻址

指令编码中直接给出操作数的寻址方式称为立即寻址。在这种寻址方式中，紧跟在操作码之后的操作数称为立即数。立即数可以为一个字节，也可以是两个字节，并要用符号“＃”来标识。由于立即数是一个常数，所以只能作为源操作数。

立即寻址所对应的寻址空间为：ROM

如：

MOV A，＃50H



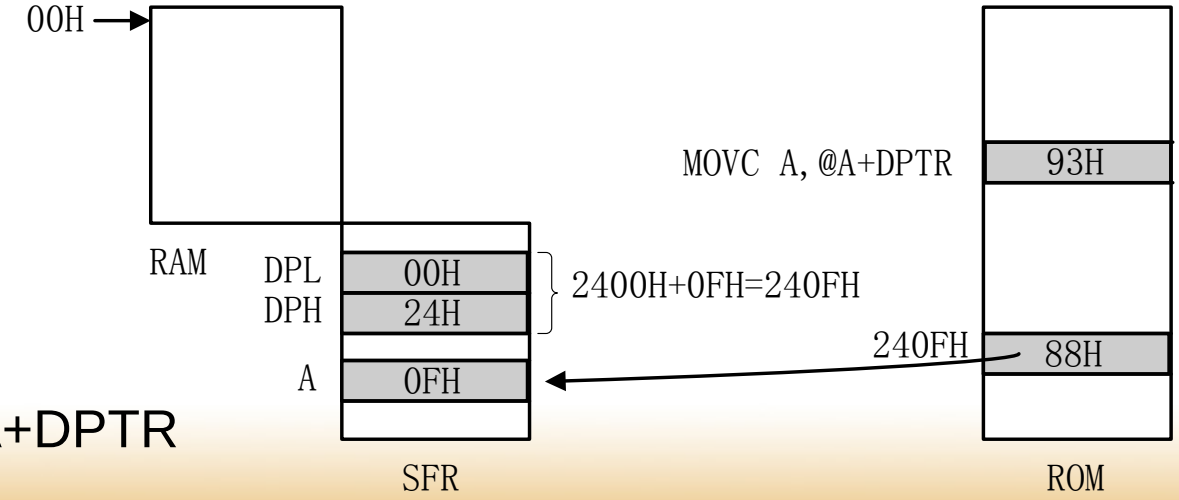
3.2.5 变址寻址

以一个基地址加上一个偏移量地址形成操作数地址的寻址方式称为变址寻址。在这种寻址方式中，以数据指针DPTR或程序计数器PC作为基址寄存器，累加器A作为偏移量寄存器，基址寄存器的内容与偏移量寄存器的内容之和作为操作数地址。

变址寻址所对应的寻址空间为：ROM

如：

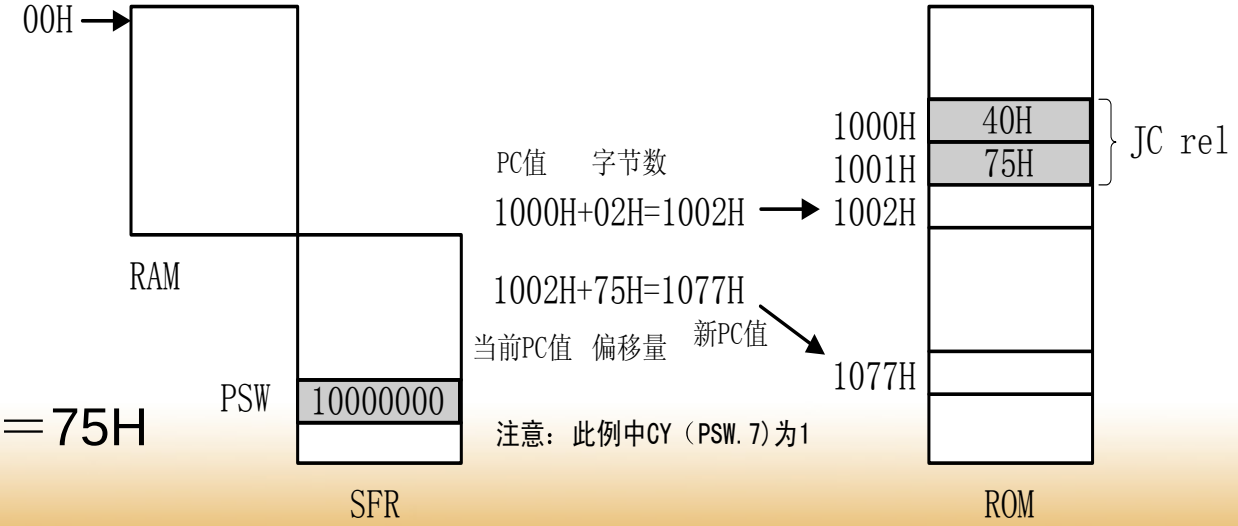
MOVC A，@A+DPTR



3.2.6 相对寻址

相对寻址是以程序计数器PC的当前值（指读出该2字节或3字节的跳转指令后，PC指向的下条指令的地址）为基准，加上指令中给出的相对偏移量 rel 形成目标地址的寻址方式 。

rel 是一个带符号的8位二进制数，取值范围是－128～＋127，以补码形式置于操作码之后存放。



如：

JC rel ； rel＝75H

3.2.7 位寻址

对位地址中的内容进行操作的寻址方式称为位寻址。采用位寻址指令的操作数是8位二进制数中的某一位。指令中给出的是位地址。位寻址方式实质属于位的直接寻址。

寻址空间为：片内RAM的20H~2FH单元中的128可寻址位；SFR的可寻址位。

习惯上，特殊功能寄存器的寻址位常用符号位地址表示。

如： CLR ACC.0

MOV 30H，C

3.3 数据传送类指令（29条）

- 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，这类指令一般不影响标志寄存器PSW的状态。

- 传送类指令可以分成两大类。一是采用MOV操作符，称为一般传送指令；二是采用非MOV操作符，称为特殊传送指令，如：MOVC、MOVX、PUSH、POP、XCH、XCHD及SWAP。

3.3.1 一般传送指令

编号	指令分类	指令及其注释	字节数	机器周期数
1	16 位传送	MOV DPTR, #data16 ; 16 位常数送 DPTR	3	2
2	A 为目的	MOV A, Rn ; Rn 的内容送 A	1	1
3		MOV A, direct ; direct 的内容送 A	2	1
4		MOV A, @Ri ; Ri 指示单元内容送 A	1	1
5		MOV A, #data ; 常数 data 送 A	2	1
6	Rn 为目的	MOV Rn, A ; A 的内容送 Rn	1	1
7		MOV Rn, direct ; direct 的内容送 Rn	2	2
8		MOV Rn, #data ; 常数 data 送 Rn	2	1
9	direct 为目的	MOV direct, A ; A 的内容送 direct	2	1
10		MOV direct, Rn ; Rn 的内容送 direct	2	2
11		MOV direct1, direct2 ; direct2 内容送 direct1	3	2
12		MOV direct, @Ri ; Ri 指示单元内容送 direct	2	2
13		MOV direct, #data ; 常数 data 送 direct	3	2
14	@Ri 为目的	MOV @Ri, A ; A 的内容送 Ri 指示单元	1	1
15		MOV @Ri, direct ; direct 的内容送 Ri 指示单元	2	2
16		MOV @Ri, #data ; 常数 data 送 Ri 指示单元	2	1

一、16位传送

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。源操作数的寻址方式为立即寻址。

例如：

执行指令 MOV DPTR，#1234H 后（DPH）=12H，（DPL）=34H。

二、8位传送

	目的字节	源字节
MOV	A	A
	Rn	Rn
	direct	direct
	@Ri	@Ri
	---	#data

在5种源字节中，只有＃data不能用作目的字节。所以可以用4种目的字节为基础构造4类指令。相应的源字节选择依据是：

- 源字节与目的字节不相同（除direct外）；
- 寄存器寻址与寄存器间接寻址间不相互传送。

1. 以A为目的

$$\text{MOV}\quad A,\begin{cases} Rn & ;\ A\leftarrow(Rn) \\ direct & ;\ A\leftarrow(direct) \\ @Ri & ;\ A\leftarrow((Ri)) \\ \#data & ;\ A\leftarrow data \end{cases}$$

•这组指令的功能是把源字节送入累加器中。源字节的寻址方式分别为直接寻址、寄存器间接寻址、寄存器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。

2．以Rn 为目的

$$\text{MOV}\quad \text{Rn，}\begin{cases} \text{A} & ;\ \text{Rn}\leftarrow(\text{A}) \\ \text{direct} & ;\ \text{Rn}\leftarrow(\text{direct}) \\ \text{\#data} & ;\ \text{Rn}\leftarrow \text{data} \end{cases}$$

•这组指令的功能是把源字节送入寄存器**Rn**中。源字节的寻址方式分别为立即寻址、直接寻址和寄存器寻址（由于目的字节为工作寄存器，所以源字节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. 以direct 为目的

MOV direct，
- A ； direct ←（A）
- Rn ； direct ←（Rn）
- direct1 ； direct ←（direct1）
- @Ri ； direct ←（（Ri））
- #data ； direct ← data

•这组指令的功能是把源字节送入direct中。源字节的寻址方式分别为立即寻址、直接寻址、寄存器间接寻址和寄存器寻址。

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

4. 以@Ri为目的

$$\text{MOV}\quad @Ri,\begin{cases} A & ; \quad (Ri)\leftarrow(A) \\ direct & ; \quad (Ri)\leftarrow(direct) \\ \#data & ; \quad (Ri)\leftarrow data \end{cases}$$

•这组指令的功能是把源字节送入**Ri**内容为地址的单元，源字节寻址方式为立即寻址、直接寻址和寄存器寻址（因目的字节采用寄存器间接寻址，故源字节不能是寄存器及其间址寻址）。

例：若（R1）=30H，（A）=20H，执行指令MOV @R1，A 后，（30H）=20H。

3.3.2 特殊传送指令

特殊传送指令的操作符为：

MOVC、MOVX、PUSH、POP、XCH、XCHD和 SWAP。

功能分别为：**ROM查表、外部RAM读写、堆栈操作和交换指令**

编号	指令分类	指令及其注释	字节数	机器周期数
17	ROM 查表	MOVC A，@A+DPTR ；DPTR 为基址、A 为偏移量	1	2
18		MOVC A，@A+PC ；PC 为基址、A 为偏移量	1	2
19	读片外 RAM	MOVX A，@DPTR ；片外 DPTR 指示单元送 A	1	2
20		MOVX A，@Ri ；片外 Ri 指示单元送 A	1	2
21	写片外 RAM	MOVX @DPTR，A ；A 内容送片外 DPTR 指示单元	1	2
22		MOVX @Ri，A ；A 内容送片外 Ri 指示单元	1	2
23	堆栈操作	PUSH direct ；将 direct 内容压入堆栈	2	2
24		POP direct ；堆栈中内容弹出到 direct 中	2	2
25	字节交换	XCH A，Rn ；Rn 内容与 A 内容交换	1	1
26		XCH A，direct ；direct 内容与 A 内容交换	2	1
27		XCH A，@Ri ；Ri 指示单元与 A 内容交换	1	1
28	半字节交换	XCHD A，@Ri ；Ri 指示单元与 A 低半字节交换	1	1
29	自交换	SWAP A ；A 的高 4 位、低 4 位自交换	1	1

一、ROM查表

1．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。

2．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。

二、读写片外RAM

1. 读片外RAM

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

第一条指令以16位DPTR为间址寄存器读片外RAM，可以寻址整个64K字节的片外RAM空间。指令执行时，在DPH中的高8位地址由P2口输出，在DPL中的低8位地址由P0口分时输出，并由ALE信号锁存在地址锁存器中。

第二条指令以R0或R1为间址寄存器，也可以读整个64K字节的片外RAM空间。指令执行时，低8位地址在R0或R1中由P0口分时输出，ALE信号将地址信息锁存在地址锁存器中（多于256字节的访问，高位地址由P2口提供）。

读片外RAM的MOVX操作，使P3.7引脚输出的信号选通片外RAM单元，相应单元的数据从P0口读入累加器中。

2．写片外RAM

MOVX　@DPTR，A　；（（DPTR））←（A）

MOVX　@Ri，A　　；（（Ri））←（A）

第一条指令以16位DPTR为间址寄存器写外部RAM，可以寻址整个64K字节的片外RAM空间。指令执行时，在DPH中高8位地址由P2口输出，在DPL中的低8位地址，由P0口分时输出，并由ALE信号锁存在地址锁存器中。

第二条指令以R0或R1为间址寄存器，也可以写整个64K字节的片外RAM空间。指令执行时，低8位地址在R0或R1中由P0口分时输出，ALE信号将地址信息锁存在地址锁存器中（多于256字节的访问，高位地址由P2口提供）。

写片外RAM的“MOVX”操作，使P3.6引脚的信号有效，累加器A的内容从P0口输出并写入选通的相应片外RAM单元。

(外部I/O口操作类同）

三、堆栈操作

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）。

在80C51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。

PUSH direct ；SP←（SP）＋1，（SP）←（direct）

POP direct ；direct←（（SP）），SP←（SP）－1

例：若（SP）=07H，（40H）=88H，执行指令 PUSH 40H 后，（SP）=08H，（08H）=88H。

四、数据交换

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 。

1. 字节交换

$$\text{XCH}\quad \text{A},\begin{cases} \text{Rn} & ;\quad (\text{A})\leftrightarrow(\text{Rn}) \\ \text{direct} & ;\quad (\text{A})\leftrightarrow(\text{direct}) \\ @\text{Ri} & ;\quad (\text{A})\leftrightarrow((\text{Ri})) \end{cases}$$

例：若（R0）=80H，（A）=20H。执行指令 XCH A，R0 后，（A）=80H，（R0）=20H。

2．半字节交换

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

SWAP A ；（$A_{3\sim0}$）↔（$A_{7\sim4}$）

XCHD指令的功能是间址操作数的低半字节与A的低半字节内容互换。

SWAP指令的功能是累加器 的高低4位互换。

例：若（R0）＝30H，（30H）＝67H，（A）＝20H。执行指令 XCHD A，@R0 指令后，（A）＝27H，（30H）＝60H。

若（A）＝30H，执行指令SWAP A后，（A）＝03H。

3.4 算术运算类指令（24条）

算术运算指令可以完成加、减、乘、除及加加1和减1等运算。这类指令多数以A为源操作数之一，同时又使A为目的操作数。

编号	指令分类	指令及其注释	字节	机器周期
30	不带进位加	ADD A，Rn ；Rn 和 A 的内容相加送 A	1	1
31		ADD A，direct ；direct 和 A 的内容相加送 A	2	1
32		ADD A，@Ri ；Ri 指示单元和 A 的内容相加送 A	1	1
33		ADD A，#data ；data 加上 A 的内容送 A	2	1
34	带进位加	ADDC A，Rn ；Rn、A 内容及进位位相加送 A	1	1
35		ADDC A，direct ；direct、A 内容及进位位相加送 A	2	1
36		ADDC A，@Ri ；Ri 指示单元、A 及进位位相加送 A	1	1
37		ADDC A，#data ；data、A 内容及进位位相加送 A	2	1

38	加 1	INC A ; A 的内容加 1 送 A	1	1
39		INC Rn ; Rn 内容加 1 送 Rn	1	1
40		INC direct ; direct 内容加 1 送 direct	2	1
41		INC @Ri ; Ri 指示单元内容加 1 送 Ri 指示单元	1	1
42		INC DPTR ; DPTR 内容加 1 送 DPTR	1	2
43	十进制调整	DA A ; 对 BCD 码加法结果调整	1	1
44	带借位减	SUBB A，Rn ; A 减 Rn 内容及进位位送 A	1	1
45		SUBB A，direct ; A 减 direct 内容及进位位送 A	2	1
46		SUBB A，@Ri ; A 减 Ri 指示单元内容及进位位送 A	1	1
47		SUBB A，#data ; A 减 data 及进位位送 A	2	1
48	减 1	DEC A ; A 的内容减 1 送 A	1	1
49		DEC Rn ; Rn 的内容减 1 送 Rn	1	1
50		DEC direct ; direct 内容减 1 送 direct	2	1
51		DEC @Ri ; Ri 指示单元内容减 1 送 Ri 指示单元	1	1
52	乘法	MUL AB ; A 乘以 B，结果高位在 B、低位在 A	1	4
53	除法	DIV AB ; A 除以 B，结果余数在 B、商在 A	1	4

进位（借位）标志CY为无符号整数的多字节加法、减法、移位等操作提供了方便；溢出标志OV可方便的控制补码运算；辅助进位标志AC用于BCD码运算。算术运算操作将影响PSW中的OV、CY、AC和P等。

指令 / 标志	ADD	ADDC	SUBB	DA	MUL	DIV
CY	√	√	√	√	0	0
AC	√	√	√	√	X	X
OV	√	√	√	X	√	√
P	√	√	√	√	√	√

注：符号√表示相应的指令操作影响标志；符号0表示相应的指令操作对该标志清0。符号X表示相应的指令操作不影响标志。另外，累加器加1（INC A）和减1(DEC A)指令影响P标志。

3.4.1 加法

一、不带进位加

$$\text{ADD}\quad A,\begin{cases} Rn & ;\ A\leftarrow(A)+(Rn) \\ direct & ;\ A\leftarrow(A)+(direct) \\ @Ri & ;\ A\leftarrow(A)+((Ri)) \\ \#data & ;\ A\leftarrow(A)+data \end{cases}$$

CY：和的**D7**位有进位时，（**CY**）**=1**；否则，（**CY**）**=0**。

AC：和的**D3**位有进位时，（**AC**）**=1**；否则，（**AC**）**=0**。

OV：和的**D7**、**D6**位只有一个有进位时，（**OV**）**=1**；溢出表示运算的结果超出了数值所允许的范围。如：两个正数相加结果为负数或两个负数相加结果为正数时属于错误结果，此时（**OV**）**=1**。

P：累加器**ACC**中“**1**”的个数为奇数时，（**P**）**=1**；为偶数时，（**P**）**=0**。

例 若（A）=84H，（30H）=8DH，执行指令 ADD A，30H 之后，由于：

（A）：	1 0 0 0　0 1 0 0
＋（30H）：	1 0 0 0　1 1 0 1
进位：　1	1　1
结果：	0 0 1　0 0 0 1

即：（A）=11H，（CY）=1，（AC）=1，（OV）=1（D7有进位，D6无进位），（P）=0。

二、带进位加

$$\text{ADDC}\quad A,\begin{cases} Rn & ;\ A\leftarrow(A)+(Rn)+(CY) \\ direct & ;\ A\leftarrow(A)+(direct)+(CY) \\ @Ri & ;\ A\leftarrow(A)+((Ri))+(CY) \\ \#data & ;\ A\leftarrow(A)+data+(CY) \end{cases}$$

指令的功能是把源操作数与累加器A的内容相加再与进位标志CY的值相加，结果送入目的操作数A中。

加的进位标志CY的值是在该指令执行之前已经存在的进位标志的值，而不是执行该指令过程中产生的进位 。

三、增1

INC
- A ; A←(A)+1
- Rn ; Rn←(Rn)+1
- direct ; direct←(direct)+1
- @Ri ; (Ri)←((Ri))+1
- DPTR ; DPTR←(DPTR)+1

指令的功能是把源操作数的内容加 1，结果再送回原单元。这些指令仅 INC A 影响P标志。其余指令都不影响标志位的状态。

四、十进制调整

DA A

指令的功能是对累加器A中刚进行的两个BCD码的加法的结果进行十进制调整。

两个压缩的BCD码按二进制相加后，必须经过调整方能得到正确的压缩BCD码的和。

调整要完成的任务是：

（1）当累加器A中的低4位数出现了非BCD码（1010~1111）或低4位产生进位（AC=1），则应在低4位加6调整，以产生低4位正确的BCD结果。

（2）当累加器A中的高4位数出现了非BCD码（1010~1111）或高4位产生进位（CY=1），则应在高4位加6调整，以产生高4位正确的BCD结果。

十进制调整指令执行后，PSW中的CY表示结果的百位值。

例 若（A）＝0101 0110B，表示的BCD码为，（R2）＝0110 0111B，表示的BCD码为，（CY）＝0。执行以下指令：

```
ADD  A，R2
DA   A
```

由于（A）＝0010 0011B，即，且（CY）＝1，即

（A）：	0 1 0 1　0 1 1 0
＋（R3）：	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
结果：1	0 0 1 0　0 0 1 1

结果为BCD数123。

应该注意，DA指令不能对减法进行十进制调整。

3.4.2 减法

一、带借位减

SUBB A，

Rn	；A←(A)–(Rn)–(CY)
direct	；A←(A)–(direct)–(CY)
@Ri	；A←(A)–((Ri))–(CY)
#data	；A←(A)–data –(CY)

CY：差的位7需借位时，（CY）=1；否则，（CY）=0。

AC：差的位3需借位时，（AC）=1；否则，（AC）=0。

OV：若位6有借位而位7无借位或位7有借位而位6无借位时，（OV）=1。

如要用此组指令完成不带借位减法，只需先清CY为 0。

例 若（A）=C9H，（R2）=54H，（CY）=1，执行指令 SUBB A，R2 之后，由于：

（A）：	1 1 0 0　1 0 0 1
－（CY）：	1
	1 1 0 0　1 0 0 0
－（R2）：	0 1 0 1　0 1 0 0
结果：	0 1 1 1　0 1 0 0

即：（A）=74H，（CY）=0，（AC）=1，（OV）=1（位6有借位，位7无借位），（P）=0。

二、减1

$$
\text{DEC}\begin{cases}
\text{A} & ;\ A \leftarrow (A)-1 \\
\text{Rn} & ;\ Rn \leftarrow (Rn)-1 \\
\text{direct} & ;\ direct \leftarrow (direct)-1 \\
\text{@Ri} & ;\ (Ri) \leftarrow ((Ri))-1
\end{cases}
$$

- 这组指令的功能是把操作数的内容减 1，结果再送回原单元。
- 这组指令仅 DEC A 影响P标志。其余指令都不影响标志位的状态。

3.4.3 乘法

MUL AB ；累加器A与B寄存器相乘

该指令的功能是将累加器A与寄存器B中的无符号8位二进制数相乘，乘积的低8位留在累加器A中，高8位存放在寄存器B中。

- 当乘积大于FFH时，溢出标志位（OV）=1。而标志CY总是被清0。

例 若（A）=50H，（B）=A0H，执行指令MUL AB 之后，（A）=00H，（B）=32H，（OV）=1，（CY）=0。

3.4.4 除法

DIV AB ；累加器A除以寄存器B

该指令的功能是将累加器A中的无符号8位二进制数除以寄存器B中的无符号8位二进制数，商的整数部分存放在累加器A中，余数部分存放在寄存器B中。

- 当除数为0时，则结果的A和B的内容不定，且溢出标志位（OV）=1。而标志CY总是被清0。

例 若（A）=FBH（251），（B）=12H（18），执行指令 DIV AB 之后，（A）=0DH，（B）=11H，（OV）=0，（CY）=0。

3.5 逻辑运算与循环类指令（24条）

逻辑运算指令可以完成与、或、异或、清0和取反操作，当以累加器A为目的操作数时，对P标志有影响；

循环指令是对累加器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；

累加器清0操作对P标志有影响。

编号	指令分类	指令及其注释	字节数	机器周期数
54	逻辑与	ANL direct，A ；direct、A 内容相与结果送 direct	2	1
55		ANL direct，#data ；direct 内容、data 相与结果送 direct	3	2
56		ANL A，Rn ；A、Rn 内容相与结果送 A	1	1
57		ANL A，direct ；A、direct 内容相与结果送 A	2	1
58		ANL A，@Ri ；A、Ri 指示单元内容相与结果送 A	1	1
59		ANL A，#data ；A 内容、data 相与结果送 A	2	1
60	逻辑或	ORL direct，A ；direct、A 内容相或结果送 direct	2	1
61		ORL direct，#data ；direct 内容、data 相或结果送 direct	3	2
62		ORL A，Rn ；A、Rn 内容相或结果送 A	1	1
63		ORL A，direct ；A、direct 内容相或结果送 A	2	1
64		ORL A，@Ri ；A、Ri 指示单元内容相或结果送 A	1	1
65		ORL A，#data ；A 内容、data 相或结果送 A	2	1

66	逻辑异或	XRL direct，A ；direct、A 内容异或结果送 direct	2	1
67		XRL direct，#data ；direct 内容、data 异或结果送 direct	3	2
68		XRL A，Rn ；A、Rn 内容异或结果送 A	1	1
69		XRL A，direct ；A、direct 内容异或结果送 A	2	1
70		XRL A，@Ri ；A、Ri 指示单元内容异或结果送 A	1	1
71		XRL A，#data ；A 内容、data 异或结果送 A	2	1
72	清 0 取反 移位	CLR A ；A 内容清 0	1	1
73		CPL A ；A 内容取反	1	1
74		RR A ；A 内容循环右移 1 位	1	1
75		RRC A ；A 内容带进位循环右移 1 位	1	1
76		RL A ；A 内容循环左移 1 位	1	1
77		RLC A ；A 内容带进位循环左移 1 位	1	1

3.5.1 逻辑与

$$\text{ANL direct,}\begin{cases}\text{A} & ;\ \text{direct}\leftarrow(\text{direct})\wedge(\text{A})\\ \text{\#data} & ;\ \text{direct}\leftarrow(\text{direct})\wedge\text{data}\end{cases}$$

$$\text{ANL A,}\begin{cases}\text{Rn} & ;\ \text{A}\leftarrow(\text{A})\wedge(\text{Rn})\\ \text{direct} & ;\ \text{A}\leftarrow(\text{A})\wedge(\text{direct})\\ \text{@Ri} & ;\ \text{A}\leftarrow(\text{A})\wedge((\text{Ri}))\\ \text{\#data} & ;\ \text{A}\leftarrow(\text{A})\wedge\text{data}\end{cases}$$

前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。

后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。

例 若（A）=C3H，（R0）=AAH，执行指令 ANL A，R0 之后，（A）=82H。

3.5.2 逻辑或

$$\text{ORL}\quad \text{direct},\begin{cases} \text{A} & ;\ \text{direct}\leftarrow(\text{direct})\vee(\text{A}) \\ \text{\#data} & ;\ \text{direct}\leftarrow(\text{direct})\vee\text{data} \end{cases}$$

$$\text{ORL}\quad \text{A},\begin{cases} \text{Rn} & ;\ \text{A}\leftarrow(\text{A})\vee(\text{Rn}) \\ \text{direct} & ;\ \text{A}\leftarrow(\text{A})\vee(\text{direct}) \\ \text{@Ri} & ;\ \text{A}\leftarrow(\text{A})\vee((\text{Ri})) \\ \text{\#data} & ;\ \text{A}\leftarrow(\text{A})\vee\text{data} \end{cases}$$

前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。

后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。

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

3.5.3 逻辑异或

XRL direct，
- A ；direct←(direct)⊕(A)
- #data ；direct←(direct)⊕data

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

前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。

后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。

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

3.5.4 累加器清0和取反

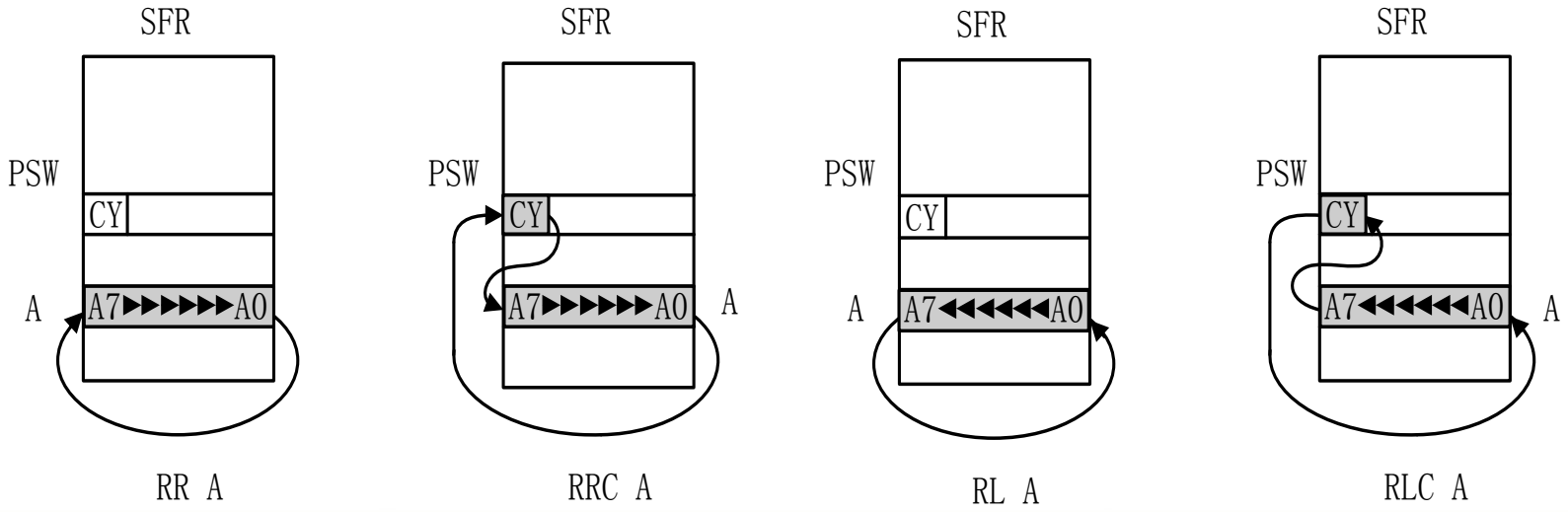
$$\left.\begin{array}{l}\text{CLR}\\ \text{CPL}\end{array}\right\}\ \text{A}\qquad \begin{array}{l};\ \text{A}\leftarrow 0\\ ;\ \text{A}\leftarrow \overline{\text{A}}\end{array}$$

这两条指令的功能分别是把累加器A的内容清 0和取反，结果仍在A中。

例 若（A）=A5H，执行指令 CLR A 之后，（A）=00H。

3.5.5 累加器循环移位

RR
RRC
RL
RLC
} A



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 控制转移类指令（17条）

通常情况下，程序的执行是顺序进行的，但也可以根据需要改变程序的执行顺序，这种情况称作程序转移。

控制程序的转移要利用转移指令。80C51的转移指令有无条件转移、条件转移及子程序调用与返回等。

编号	指令分类		指令及其注释	字节数	机器周期数
78	无条件转移	短转	AJMP addr11 ；程序转移到 addr11 指示的地址处	2	2
79		长转	LJMP addr16 ；程序转移到 addr16 指示的地址处	3	2
80		相对	SJMP rel ；程序转移到 rel 相对地址处	2	2
81		散转	JMP @A＋DPTR ；程序转移到变址指出的地址处	1	2
82	条件转移	判 0	JZ rel ；A 为 0，程序转到 rel 相对地址处	2	2
83			JNZ rel ；A 不为 0，程序转到 rel 相对地址处	2	2
84		比较不等	CJNE A，direct，rel ；A 与 direct 内容不等转	3	2
85			CJNE A，＃data，rel ；A 内容与 data 不等转	3	2
86			CJNE Rn，＃data，rel ；Rn 内容与 data 不等转	3	2
87			CJNE @Ri，＃data，rel；Ri 间址内容与 data 不等转	3	2
88		减 1 不为 0	DJNZ Rn，rel ；Rn 内容减 1 不为 0 转	2	2
89			DJNZ direct，rel ；direct 内容减 1 不为 0 转	3	2
90	子程序	调用	ACALL addr11 ；调用 addr11 处子程序	2	2
91			LCALL addr16 ；调用 addr16 处子程序	3	2
92		返回	RET ；子程序返回	1	2
93	中断返回		RETI ；中断返回	1	2
94	空操作		NOP ；空操作	1	1

3.6.1 无条件转移

一、 短跳转

AJMP addr11 ； PC ← （PC）+ 2，

； PC10~0 ← addr11

该指令执行时，先将PC的内容加2（这是PC指向的是AJMP的下一条指令），然后把指令中11位地址码传送到PC10~0，而PC15~11保持原内容不变。

- 在目标地址的11位中，前3位为页地址，后8位为页内地址（每页含256个单元）。

当前PC的高5位（即下条指令的存储地址的高5位）可以确定32个2KB段之一。所以，AJMP指令的转移范围为包含AJMP下条指令在内的2KB区间。

ROM 空间中 32 个 2K 地址范围

1	0000H~07FFH	9	4000H~47FFH	17	8000H~87FFH	25	C000H~C7FFH
2	0800H~0FFFH	10	4800H~4FFFH	18	8800H~8FFFH	26	C800H~CFFFH
3	1000H~17FFH	11	5000H~57FFH	19	9000H~97FFH	27	D000H~D7FFH
4	1800H~1FFFH	12	5800H~5FFFH	20	9800H~9FFFH	28	D800H~DFFFH
5	2000H~27FFH	13	6000H~67FFH	21	A000H~A7FFH	29	E000H~E7FFH
6	2800H~2FFFH	14	6800H~6FFFH	22	A800H~AFFFH	30	E800H~EFFFH
7	3000H~37FFH	15	7000H~77FFH	23	B000H~B7FFH	31	F000H~F7FFH
8	3800H~3FFFH	16	7800H~7FFFH	24	B800H~BFFFH	32	F800H~FFFFH

二、 长跳转

LJMP addr16 ； PC ← addr16

第一字节为操作码，该指令执行时，将指令的第二、三字节地址码分别装入指令计数器PC的高8位和低8位中，程序无条件地转移到指定的目标地址去执行。

LJMP提供的是16位地址，因此程序可以转向64KB的程序存储器地址空间的任何单元。

例 若标号“NEWADD”表示转移目标地址1234H。执行指令 LJMP NEWADD 时，两字节的目标地址将装入PC中，使程序转向目标地址 1234H 处运行。

三、 相对转移

SJMP rel ；PC ←（PC）+ 2，PC ←（PC）+ rel

第一字节为操作码，第二字节为相对偏移量 rel，rel 是一个带符号的偏移字节数（2的补码），取值范围为 ＋127 ～－128（00H～7FH对应表示0 ～ ＋127，80H～FFH对应表示－128～－1）。负数表示反向转移，正数表示正向转移。

rel 可以是一个转移目标地址的标号，由汇编程序在汇编过程中自动计算偏移地址，并填入指令代码中。在手工汇编时，可用转移目标地址减转移指令所在的源地址，再减转移指令字节数2得到偏移字节数rel。

例 若标号“NEWADD”表示转移目标地址0123H，PC的当前值为0100H。执行指令 SJMP NEWADD 后，程序将转向 0123H 处执行（此时re l= 0123H－（0100＋2）= 21H）。

四、 散转移

JMP @A+DPTR ；PC ←（PC）+ 1，PC ←（A）+（DPTR）

该指令具有散转功能，可以代替许多判别跳转指令。其转移地址由数据指针DPTR的16位数和累加器A的8位数进行无符号数相加形成，并直接装入PC。该指令执行时对标志位无影响。

例 有一段程序如下：

```
        MOV   DPTR，#TABLE
        JMP   @A+DPTR
TABLE：AJMP   ROUT0
        AJMP   ROUT1
        AJMP   ROUT2
        AJMP   ROUT3
```

•当（A）=00H时，程序将转到 ROUT0处执行；当（A）=02H时，程序将转到 ROUT1处执行；其余类推。

3.6.2 条件转移

一、 累加器判0转移

JZ rel ；若（A）=0，则 PC ← （PC）+rel

JNZ rel ；若（A）≠0，则 PC ← （PC）+rel

指令的功能是对累加器A的内容为 0 和不为 0 进行检测并转移。当不满足各自的条件时，程序继续往下执行。当各自的条件满足时，程序转向指定的目标地址。目标地址的计算与SJMP指令情况相同。指令执行时对标志位无影响。

例 若累加器A原始内容为00H，则：

```
JNZ  L1 ；由于A的内容为00H，所以程序往下执行
INC  A  ；

JNZ  L2 ；由于A的内容已不为0，所以程序转向L2 处执行
```

二、 比较不相等转移

CJNE A，direct，rel ； 若（A）≠（direct），则 PC ← （PC）+ rel

CJNE { A / Rn / @Ri } ,#data，rel

； 若（A）≠data，则 PC ← （PC）+ rel

； 若（Rn）≠data，则 PC ← （PC）+ rel

； 若（(Ri)）≠data，则 PC ← （PC）+ rel

这组指令的功能是对指定的目的字节和源字节进行比较，若它们的值不相等则转移，转移的目标地址为当前的**PC**值加**3**后，再加指令的第三字节偏移量**rel**；若目的字节的内容大于源字节的内容，则进位标志清**0**；若目的字节的内容小于源字节的内容，则进位标志置**1**；若目的字节的内容等于源字节的内容，程序将继续往下执行。

三、 减1不为0转移

```
DJNZ   Rn，rel      ；PC←（PC）+2，Rn←（Rn）−1
                    ；若（Rn）≠0，则 PC←（PC ）+ rel ，继续循环
                    ；若（Rn）= 0，则结束循环，程序往下执行
DJNZ   direct，rel  ；PC←（PC）+3，direct←（direct）−1
                    ；若（direct）≠0，则 PC←（PC ）+rel，继续循环
                    ；若（direct）=0，则结束循环，程序往下执行
```

这组指令每执行一次，便将目的操作数的循环控制单元的内容减1，并判其是否为 0。若不为0，则转移到目标地址继续循环；若为0，则结束循环，程序往下执行。

例　有一段程序如下：

```
        MOV  23H，#0AH
        CLR   A
LOOPX：ADD   A，23H
        DJNZ  23H，LOOPX
        SJMP  $
```

该程序执行后:

（A）=10+9+8+7+6+5+4+3+2+1=37H

3.6.3 调用与转移

一、 调用

ACALL addr11 ; PC ← (PC) +2, SP← (SP) +1, (SP) ← (PC_{7-0})

; SP← (SP) +1, (SP) ← (PC_{15-8}), PC_{10-0} ← addr11

LCALL addr16 ; PC ← (PC) +3, SP← (SP) +1, (SP) ← (PC_{7-0})

; SP← (SP) +1, (SP) ← (PC_{15-8}), PC← addr16

这两条指令可以实现子程序的短调用和长调用。目标地址的形成方式与**AJMP**和**LJMP**相似。这两条指令的执行不影响任何标志。

•ACALL指令执行时，被调用的子程序的首址必须设在包含当前指令（即调用指令的下一条指令）的第一个字节在内的2K字节范围内的程序存储器中。

•LCALL指令执行时，被调用的子程序的首址可以设在64K字节范围内的程序存储器空间的任何位置。

例 若（SP）=07H，标号“XADD”表示的实际地址为0345H，PC的当前值为0123H。执行指令 ACALL XADD 后，（PC）+2=0125H，其低8位的25H压入堆栈的08H单元，其高8位的01H压入堆栈的09H单元。（PC）=0345H，程序转向目标地址0345H 处执行。

二、 返回

RET ；$PC_{15-8} \leftarrow ((SP))$，$SP \leftarrow (SP) - 1$

；$PC_{7-0} \leftarrow ((SP))$，$SP \leftarrow (SP) - 1$

RETI ；$PC_{15-8} \leftarrow ((SP))$，$SP \leftarrow (SP) - 1$

；$PC_{7-0} \leftarrow ((SP))$，$SP \leftarrow (SP) - 1$

•**RET**指令的功能是从堆栈中弹出由调用指令压入堆栈保护的断点地址，并送入指令计数器**PC**，从而结束子程序的执行。程序返回到断点处继续执行。

•**RETI**指令是专用于中断服务程序返回的指令，除正确返回中断断点处执行主程序以外，并有清除内部相应的中断状态寄存器（以保证正确的中断逻辑）的功能。

3.6.4 空操作

NOP ；PC ← （PC）+ 1

•这条指令不产生任何控制操作，只是将程序计数器PC的内容加1。该指令在执行时间上要消耗1个机器周期，在存储空间上可以占用一个字节。因此，常用来实现较短时间的延时。

3.7 位操作类指令（17条）

位操作又称布尔操作，它是以位为单位进行的各种操作。位操作指令中的位地址有4 种表示形式：

- 直接地址方式（如，0D5H）；
- 点操作符方式（如，0D0H.5、PSW.5等）；
- 位名称方式（如，F0）；
- 伪指令定义方式（如，MYFLAG BIT F0）。

以上几种形式表示的都是PSW中的位5。

与字节操作指令中累加器ACC用字符“A”表示类似的是，在位操作指令中，位累加器要用字符“C”表示（注：在位操作指令中CY与具体的直接位地址D7H对应）。

编号	指令分类		指令及其注释	字节数	机器周期数
95	位传送		MOV bit，C ；CY 状态送入 bit 中	2	2
96			MOV C，bit ；bit 状态送入 CY 中	2	1
97	位设置	清 0	CLR C ；CY 状态清 0	1	1
98			CLR bit ；bit 状态清 0	2	1
99		置位	SETB C ；CY 状态置 1	1	1
100			SETB bit ；bit 状态置 1	2	1
101	位逻辑	位与	ANL C，bit ；CY 状态与 bit 状态相与结果送 CY	2	2
102			ANL C，/bit ；CY 状态与 bit 取反相与结果送 CY	2	2
103		位或	ORL C，bit ；CY 状态与 bit 状态相或结果送 CY	2	2
104			ORL C，/bit ；CY 状态与 bit 取反相或结果送 CY	2	2
105		取反	CPL C ；CY 状态取反	1	1
106			CPL bit ；bit 状态取反	2	1
107	位条件转移	判 CY	JC rel ；CY 为 0 转	2	2
108			JNC rel ；CY 不为 0 转	2	2
109		判 bit	JB bit，rel ；bit 位为 1 转	3	2
110			JBC bit，rel ；bit 位为 1 转，同时把 bit 位清 0	3	2
111			JNB bit，rel ；bit 位不为 1 转	3	2

3.7.1 位传送

MOV bit，C ；bit←（CY）

MOV C，bit ；CY ←（bit）

•这两条指令可以实现指定位地址中的内容与位累加器CY的内容的相互传送。

例 若（CY）=1，（P3）=1100 0101B，（P1）=0011 0101B。执行以下指令：

MOV P1.3，C

MOV C，P3.3

MOV P1.2，C

结果为：（CY）=0，P3的内容未变，P1的内容变为 0011 1001B。

3.7.2 位状态设置

一、位清0

$$\text{CLR}\begin{cases} \text{C} & ;\ \text{CY} \leftarrow 0 \\ \text{bit} & ;\ \text{bit} \leftarrow 0 \end{cases}$$

这两条指令可以实现位地址内容和位累加器内容的清0。

例 若（P1）=1001 1101B。执行指令CLR P1.3后，结果为：(P1)=1001 0101B。

二、位置位

$$\text{SETB}\ \begin{cases} \text{C} & ;\ \text{CY} \leftarrow 1 \\ \text{bit} & ;\ \text{bit} \leftarrow 1 \end{cases}$$

这两条指令可以实现地址内容和位累加器内容的置位。

例 若（P1）=1001 1100B。执行指令 SETB P1.0 后，（P1）＝1001 1101B。

3.7.3 位逻辑运算

一、位逻辑“与”

$$\text{ANL C,}\begin{cases}\text{bit} & ; \ CY \leftarrow (CY) \wedge \text{bit} \\ \text{/bit} & ; \ CY \leftarrow (CY) \wedge \overline{(\text{bit})}\end{cases}$$

这两条指令可以实现位地址单元内容或取反后的值与位累加器的内容“与”操作，操作的结果送位累加器C。

例 若（P1）=1001 1100B，（CY）＝1。执行指令ANL C，P1.0 后，结果为：P1 内容不变，而（CY）＝0。

二、位逻辑“或”

$$\text{ORL C,}\begin{cases}\text{bit} & ; \ CY \leftarrow (CY) \vee bit \\ \text{/bit} & ; \ CY \leftarrow (CY) \vee \overline{(bit)}\end{cases}$$

这两条指令可以实现位地址单元内容或取反后的值与位累加器的内容“或”操作，操作的结果送位累加器C。

三、位取反

$$\text{CPL}\begin{cases}\text{C} & ;\ \text{CY}\leftarrow\overline{(\text{CY})}\\ \text{bit} & ;\ \text{bit}\leftarrow\overline{(\text{bit})}\end{cases}$$

这两条指令可以实现位地址单元内容和位累加器内容的取反。

3.7.4 位判跳（条件转移）

一、判CY转移

JC rel ；若（CY）=1，PC←（PC）+2+rel，否则顺次执行。

JNC rel ；若（CY）=0，PC←（PC）+2+rel，否则顺次执行。

这两条指令的功能是对进位标志位CY进行检测，当（CY）=1（第一条指令）或（CY）=0（第二条指令），程序转向PC当前值与rel之和的目标地址去执行，否则程序将顺序执行。

二、判bit转移

JB	bit ,rel	;（bit）=1，PC←（PC）+3+rel，否则顺次执行。
JBC	bit ,rel	;（bit）=1，PC←（PC）+3+rel，并使 bit←0，否则顺次执行。
JNB	bit ,rel	;（bit）=0，PC←（PC）+3+rel，否则顺次执行。

这三条指令的功能是对指定位bit进行检测，当（bit）=1（第一和第二条指令）或（bit）=0（第三条指令），程序转向PC当前值与rel之和的目标地址去执行，否则程序将顺序执行。对于第二条指令，当条件满足时（指定位为1），还具有将该指定位清0的功能。

思考题与习题

1、80C51系列单片机的指令系统有何特点？

2、80C51单片机有哪几种寻址方式？各寻址方式所对应的寄存器或存储器空间如何？

3、访问特殊功能寄存器SFR可以采用哪些寻址方式？

4、访问内部RAM单元可以采用哪些寻址方式？

5、访问外部RAM单元可以采用哪些寻址方式？

6、访问外部程序存储器可以采用哪些寻址方式？

7、为什么说布尔处理功能是80C51单片机的重要特点？

8、对于80C52单片机内部RAM还存在高128字节，应采用何种方式访问？

9、试根据指令编码表写出下列指令的机器码。

（1）MOV A，#88H

（2）MOV R3，50H

（3）MOV P1.1，#55H

（4）ADD A，@R1

（5）SETB 12H

10、完成某种操作可以采用几条指令构成的指令序列实现，试写出完成以下每种操作的指令序列。

（1）将R0的内容传送到R1；

（2）内部RAM单元60H的内容传送到寄存器R2；

（3）外部RAM单元1000H的内容传送到内部RAM单元60H；

（4）外部RAM单元1000H的内容传送到寄存器R2；

（5）外部RAM单元1000H的内容传送到外部RAM单元2000H。

11、若（R1）=30H，（A）=40H，（30H）=60H，（40H）=08H。试分析执行下列程序段后上述各单元内容的变化。

```
MOV A，@R1
MOV @R1，40H
MOV 40H，A
MOV R1，#7FH
```

12、若（A）=E8H，（R0）=40H，（R1）=20H，（R4）=3AH，（40H）=2CH，（20）=0FH，试写出下列各指令独立执行后有关寄存器和存储单元的内容？若该指令影响标志位，试指出CY、AC、和OV的值。

（1）MOV A，@R0

（2）ANL 40H，#0FH

（3）ADD A，R4

（4）SWAP A

（5）DEC @R1

（6）XCHD A，@R1

13、若（50H）=40H，试写出执行以下程序段后累加器A、寄存器R0及内部RAM的40H、41H、42H单元中的内容各为多少？

```
MOV A，50H
MOV R0，A
MOV A，#00H
MOV @R0，A
MOV A，3BH
MOV 41H，A
MOV 42H，41H
```

14、试用位操作指令实现下列逻辑操作。要求不得改变未涉及的位的内容。

（1）使ACC.0置位；

（2）清除累加器高4位；

（3）清除ACC.3，ACC.4，ACC.5，ACC.6。

15、试编写程序，将内部RAM的20H、21H、22H三个连续单元的内容依次存入2FH、2EH和2DH单元。

16、试编写程序，完成两个16位数的减法：7F4DH－2B4EH，结果存入内部RAM的30H和31H单元，31H单元存差的高8位，30H单元存差的低8位。

17、试编写程序，将R1中的低4位数与R2中的高4位数合并成一个8位数，并将其存放在R1中。

18、试编写程序，将内部RAM的20H、21H单元的两个无符号数相乘，结果存放在R2、R3中，R2中存放高8位，R3中存放低8位。

19、若（CY）=1，（P1）=10100011B，（P3）=01101100B。试指出执行下列程序段后，CY、P1口及P3口内容的变化情况。

```
MOV  P1.3，C
MOV  P1.4，C
MOV  C，P1.6
MOV  P3.6，C
MOV  C，P1.0
MOV  P3.4，C
```

20、若单片机的主频为12MHz，试用循环转移指令编写延时20ms的延时子程序。并说明这种软件延时方式的优缺点。