汇编指令集

本节根据指令的功能来分，提供六张表来说明指令集的概况：

累加器、算数和逻辑指令（表 2）；

辅助寄存器和数据页指针指令（表 3）；

TREG、PREG 和乘法指令（表 4）；

转移指令（表 5）；

控制指令（表 6）；

I/O 和存储器操作（表 7）。

在每张表中，指令按字母顺序排列。执行每条指令所需要的周期数在表中给出，所有指令都假设从内部程序存储器和内部数据存储器中执行，指令的周期数适用于单指令执行，不适用于重复方式。编程时，用户必须对每条指令的寻址方式了解清楚，因此这里也在表中给出了每条指令的寻址方式。由于指令的操作码对用户编程没有多大指导意义，在这里就没有列出来。

为了参照起见，我们先定义这六张概述表的符号意义：

ACC 累加器。

AR　　辅助寄存器。

ARX 用于 LAR 和 SAR 指令的 3 位数据值，指定被操作的辅助寄存器。

BITX　　4 位数值，用于指定数据存储器数值中的哪一位将被 BIT 指令所测试。

CM　　2 位数值，CMPR 指令执行 CM 值所声明的比较：

若 CM=00，测试当前 AR=AR0 否；

若 CM=01，测试当前 AR<AR0 否；

若 CM=10，测试当前 AR>AR0 否；

若 CM=11，测试当前 AR≠AR0 否。

Shift　　4 位右移量。

TP　　用于条件执行指令的 2 位数值，代表如下 4 种条件：若 BIO 引脚为低，TP=00；若 TC 位=1，TP=01；若 TC 位=0，TP=10；无条件 TP=11。

表 2 累加器、算数和逻辑指令

助记符	指令功能	周期	寻址方式	指令说明
ABS	|（ACC）|→ACC	1		ACC 取绝对值
ADD	（ACC）+（数据存储器地址）× 2^{shift}→ACC （ACC）＋（数据存储器地址）× 2^{16}→ACC （ACC）＋k→ACC （ACC）+lk×2^{shift}→ACC	1 1 1 2	直接/间接 直接/间接 短立即数 长立即数	移位时低位填 0，若 SXM＝1,高位用符号扩展；若 SXM＝0，高位填 0；结果存在 ACC 中，该指令使 C=0。寻址短立即数时，加操作不受 SXM 的影响，且不能重复执行。
ADDC	（ACC）+（数据存储器地址）＋（C）→ACC	1	直接/间接	该指令抑制符号扩展
ADDS	（ACC）+（数据存储器地址）→ACC	1	直接/间接	该指令抑制符号扩展；无论 SXM 为何值，数据均作无符号 16 位数看待。当 SXM=0 和移位次数等于 0 时，ADDS 与 ADD 指令结果相同。
ADDT	（ACC）＋（数据存储器地址）左移（TREG（3:0））位→ACC（15:0）	1	直接/间接	被寻址的数据寄存器单元的内容左移并加到 ACC，移位次数由 TREG 的低 4 位确定，SXM 位的值控制移位时是否作符号扩展。
AND	（ACC（15:0））AND （数据存储器地址） →ACC（15:0） 0→ACC（31:16） （ACC（31:16））AND lk ×2^{shift}→ACC （ACC（31:16））AND lk ×2^{16}→ACC	1 2 2	直接/间接 长立即数 长立即数	使用直接/间接寻址，ACC 的低位字与数据存储器单元的值作与运算，结果存在 ACC 的低字，ACC 的高位字清 0。使用立即数寻址，长立即数可以左移，移位时，32 位中未被长立即操作数填充的位均被清零。
CMPL	（ACC）的逻辑反→ACC	1		累加器中的内容用其逻辑反取代
LACC	（数据存储器地址）×2^{shift} →ACC （数据存储器地址）×2^{16}→ACC lk×2^{shift} 位→ACC	1 1 2	直接/间接 直接/间接 长立即数	指定的数据存储器单元的内容或 16 位常数左移并加载到 ACC，低位填 0，若 SXM＝1，高位用符号位扩展；若 SXM＝0，则填 0。
LACL	（数据存储器地址）→ACC（15:0） 0→ACC（31:16） k→ACC（7:0） 0→ACC（31:16）	1 1	直接/间接 短立即数	指定的数据存储器单元的内容或用 0 扩展的 8 位常数左移并加载到 ACC 的低 16 位，ACC 的高位填 0。数据作为 16 位无符号数处理，无论 SXM 为何值，本指令都不进行符号扩展。
LACT	（数据存储器地址）左移 TREG（3:0）位→ACC（15:0）	1	直接/间接	将数据存储器单元的内容左移后加载到 ACC。左移位数由 TREG 的低 4 位确定。SXM 的值决定 ACC 高位是否符号扩展。
NEG	（ACC）×（－1）→ACC	1		将 ACC 的内容换成其相反数。当对 8000 0000h 作 NEG 操作时，OV 置 1。 若 OVM=1，ACC＝7FFFh，若 OVM＝0，则 ACC＝8000 0000h，只要 ACC 不等于 0 则 C＝0，当 ACC＝0，则 C=1。
NORM	若（ACC）=0，则 TC→1； 否则若（ACC（31））XOR（ACC（30））＝0， 则 TC→0，（ACC）×2 →ACC 按指定的方式修改 AR 否则 TC→1	1	间接	NORM 指令将 ACC 中的有符号数进行规格化。对定点数进行规格化即将其分成指数和尾数，为此对位 31 和位 30 作异或运算即可确定位 30 是数值还是符号。对当前 AR 的缺省修改方式是加 1。该指令对正、负二进制补码均适用。由于该指令在流水

				线的第 4 阶段才完成对 AR 的操作，而其它指令在流水线的第 2 阶段完成对 AR 的操作，因此紧跟在 NORM 指令后的两条指令不能对 AR 值或者 ARP 值进行修改。否则 NORM 指令将不能操作正确的数据存储器单元。
OR	（ACC（15:0））OR （数据存储器地址）→ACC（15:0） （ACC（31:16））→ACC（31:16） （ACC）OR lk×2^{shift}→ACC （ACC）OR lk×2^{16}→ACC	1 2 2	直接/间接 长立即数 长立即数	ACC 的内容和被寻址的数据地址单元的值或左移后的立即数作或操作，SXM 位对 OR 指令没有影响。数据操作数没有被占用的位总是填 0。因此在直接或间接寻址或不移位的长立即数的情况下 ACC 的高位是不变的。当操作数是立即数且移位次数不等于 0 时，操作数左移低位补 0。
ROL	（ACC（31））→C （ACC（30:0））→ACC（31:1） C→ACC（0）	1		将 ACC 内容左移一位，进位位被移入最低位，最高位被移入进位位。
ROR	C→ACC（31） （ACC（31:1））→ACC（30:0） （ACC（0））→C	1		将 ACC 内容右移一位，进位位被移入最高位，最低位被移入进位位。
SACH	（（ACC）×2^{shift}）的高 16 位→数据存储器地址	1	直接/间接	SACH 将整个 ACC 的 32 位数左移 0-7 位，将移位后数值的高 16 位保存到数据存储器。
SACL	（（ACC）×2^{shift}）的低 16 位→数据存储器地址	1	直接/间接	SACH 将整个 ACC 的 32 位数左移 0-7 位，将移位后数值的低 16 位保存到数据存储器。
SFL	（ACC（31））→C （ACC（30:0））→ACC（31:1） 0→ACC（0）	1		SFL 指令使 ACC 的内容左移一位，ACC 的最低位填 0，最高位移入进位位 C。
SFR	若 SXM=0，则 0→ACC（31） 若 SXM=1，则 ACC（31）→ACC（31） （ACC（31:1））→ACC（30:0） （ACC（0））→C	1		SFR 指令将 ACC 的内容右移一位。若 SXM=1，则产生算数右移，ACC 的符号位不变，位 31 复制到位 30，位 0 移入进位位。若 SXM=0，则产生逻辑右移，ACC 的所有内容右移一位，位 0 移入进位位，位 31 填 0
SUB	（ACC）-（数据存储器地址）×2^{shift}→ACC （ACC）-（数据存储器地址）×2^{16}→ACC （ACC）-k→ACC （ACC）-lk×2^{shift}→ACC	1 1 1 2	直接/间接 直接/间接 短立即数 长立即数	在直接间接和长立即数寻址方式里被寻址的数据存储单元的内容或 16 位常数左移，然后累加器减去移位后的值。移位时低位填 0。高位扩展与否由 SXM 决定。当使用短立即数时累加器减去 8 位正常数。此时不可以指定移位值，减法不受 SXM 的影响，并且是不可重复的。通常情况下，若减法结果有借位则进位位清 0；若减法结果没有借位，则进位位置 1。但若移位次数为 16，则当减法结果有借位时进位位清 0，减法不产生借位时，进位位不变。
SUBB	（ACC）-（数据寄存器地址）-（C 的逻辑反） →ACC	1	直接/间接	ACC 的值减去被寻址的数据寄存单元的内容及进位位的逻辑反值，抑制符号扩展。若相减结果产生借位进位位清 0；若没有借位则进位位置 1。
SUBC		1	直接/间接	ACC 实现条件减，可用于除法：把 16 位的

	若（ACC）>=0 且（数据存储器地址）>=0 （ACC）-（数据寄存器地址）×2^{15}→ALU 输出。 若 ALU 输出>=0,则（ALU 输出）×2+1→ACC 否则（ACC）×2→ACC			正被除数放在 ACC 的低 16 位，ACC 的高位清 0；16 位的正除数放在数据存储单元中。执行 SUBC 指令 16 次。；最后一次 SUBC 指令完成后 ACC 的低 16 位是除数的商，ACC 的高 16 为是余数。若 ACC 和存储单元的内容为负，则不能用 SUBC 实现除法。若 16 位被除数的有效数字少于 16 位可将它放在 ACC 中并左移，左移的位数为前面非有效的 0 的个数，SUBC 指令执行的次数为 16 减去移位次数。执行该指令不会因正溢或负溢而饱和。
SUBS	（ACC）-（数据寄存器地址）→ACC	1	直接/间接	ACC 减去数据存储单元的内容，抑制符号扩展。不管 SXM 为何值，数据存储单元的内容都作为无符号处理。ACC 还是作为有符号数。若相减结果有借位清 0，若没有借位 C 置 1。
SUBT	ACC）-（（数据寄存器地址）×$2^{（TREG（3:0））}$→ACC 若 SXM=1，则（数据存储单元）用符号扩展 若 SXM=0，则（数据存储单元）不用符号扩展	1	直接/间接	ACC 减去左移后的数据存储单元的内容，移位次数由 TREG 的低 4 位确定。SXM 控制是否对 数据存储单元的内容作符号扩展。
XOR	ACC（15:0））XOR（数据寄存器地址）→ACC（15:0） （ACC（31:1））→ACC（31:16） 或（ACC（31:0））XOR 1k×2^{shift}→ACC（31:0） 或（ACC（31:0））XOR 1k×2^{16}→ACC（31:0）	1 2	直接/间接 长立即数	直接和间接寻址时 ACC 的低 16 位和被寻址的数据存储单元的内容作异或运算，结果存在 ACC 的低 16 位，ACC 的高 16 位不变。使用长立即数寻址时，长立即数左移，数据操作数没有占用的位填 0，然后和 ACC 中的 32 位数作异或运算。
ZALR	（数据存储器地址）→ACC（31:16） 8000h→ACC（15:0）	1	直接/间接	数据存储单元的内容送到 ACC 的高 16 位，ACC 的低 15 位清 0，第 15 位置 1。

表 3 辅助寄存器指令

助记符	指令功能	周期	寻址方式	指令说明
ADRK	（当前 AR）+k→当前 AR	1		8 位立即数按右对齐方式与当前辅助寄存器值相加。
BANZ	若（当前 AR）≠ 0， 则操作数→PC 否则（PC）+2→PC 按指令要求的方式修改（当前 AR）和（ARP）	4 2	间接	若当前辅助寄存器的内容不为 0，程序控制转到指定的程序存储器地址；否则执行下一条指令。对当前 AR 缺省的修改方式是减 1。在进入循环前将辅助寄存器（作循环计数器）初始化为 N-1,可执行 N 次循环。
CMPR	（当前 AR）与（AR0）比较,并把结果放在状态寄存器 ST1 的 TC 位	1	直接/间接	该指令完成由操作数指定的比较； 若 CM==00,测试是否（当前 AR）=（AR0）； 若 CM==01,测试是否（当前 AR）<（AR0）； 若 CM==10,测试是否（当前 AR）>（AR0）； 若 CM==11,测试是否（当前 AR）=（AR0）； 若条件为真,TC 位置 1;否则 TC 位清 0。
LAR	（数据存储器地址）→ARx k→ARx 1k→ARx	1 1 2	直接/间接 短立即数 长立即数	指定数据寄存器单元中的内容。8 位常数或 16 位常数加载到指定的辅助寄存器。不论 SXM 为何值,所指定的常数均为无符号整数。
MAR	按指令指定的方式修改（当前 AR）和（ARP）	1	间接	直接寻址方式下 MAR 指令的作用同 NOP 指令。间接寻址方式下可以修改辅助寄存器的值和 ARP 的值,对寄存器的引用不起作用。
SAR	（ARx）→数据寄存器地址	1	直接/间接	将指定的辅助寄存器（ARx）的内容拷贝到指定的数据存储单元。
SBRK	（当前 AR）-K→当前 AR	1		当前辅助寄存器（由 ARP 指定的那个）的内容减去 8 位立即数，结果存在当前辅助寄存器中。

表 4 TREG、PREG 和乘法指令

助记符	指令功能	周期	寻址方式	指令说明
APAC	（ACC）+移位后的（PREG）→ACC	1		按 ST1 寄存器中 PM 状态位指定的方式将 PREG 的内容进行移位，把移位后的值与 ACC 的内容相加。状态寄存器中的 SXM 位对 APAC 指令无影响。
LPH	（数据寄存器地址）→PREG（31:16）	1	直接/间接	被寻址的数据寄存单元的内容加载到 PREG 的高 16 位。PREG 的低 16 位不变。
LT	（数据寄存器地址）→TREG	1	直接/间接	被寻址的数据寄存单元的内容加载到 TREG。
LTA	（数据寄存器地址）→TREG （ACC）+移位后的（PREG）→ACC	1	直接/间接	指令的数据寄存单元的内容加载到 TREG。按 PM 状态位指定的方式对乘积寄存器的内容进行移位，并把移位后的值与 ACC 相加，结果放在 ACC 中。若相加结果有进位则 C=1，否则 C=0。
LTD	（数据寄存器地址）→TREG （数据寄存器地址）→数据寄存器地址+1 （ACC）+移位后的（PREG）→ACC	1	直接/间接	指令将数据寄存单元的内容加载到 TREG。按 PM 状态位指定的方式对乘积寄存器的内容进行移位，并把移位后的值与 ACC 相加，结果放在 ACC 中。指定的数据存储单元的内容拷贝到地址加 1 的数据存储单元。该指令移动数据的功能不能用于外部数据寄存器或存储器映射的寄存器。
LTP	（数据寄存器地址）→TREG	1	直接/间接	指令的数据寄存单元的内容加载到 TREG。按

	移位后的（PREG）→ACC			PM 状态位指定的方式对乘积寄存器的内容进行移位，并把移位后的值送到 ACC 中。
LTS	（数据寄存器地址）→TREG ACC-移位后的（PREG）→ACC	1	直接/间接	指令的数据寄存单元的内容加载到 TREG。按 PM 状态位指定的方式对乘积寄存器的内容进行移位，并用 ACC 的内容减去（PREG）移位后的值送到 ACC 中。若相减的结果产生了借位则进位位 C 清 0；否则进位位 C 置 1。
MAC	PC 加 1 即 PC=PC+1 （PC）→MSTACK 16 位程序存储器地址→PC （ACC）+移位后的（PREG）→ACC （数据存储器地址）→TREG （数据存储器地址）×（16 位程序存储器地址）→PREG 间接寻址时，按指定方式修改（当前 AR）和（ARP） 当（重复计数器）≠0 时， （重复计数器）-1→重复计数器。 （MSTACK）→PC	3	直接/间接	MAC 指令可以 1）按 PM 状态位指定的方式把先前的乘积移位,再与 ACC 的内容相加；2）把指定的数据存储单元的内容加载到 TREG；3）将数据存储单元的内容乘以指定的程序存储器地址中的内容。若程序存储器是片内 RAM 块 BO，则 CNF 位必须置 1。 当重复 MAC 指令时,每重复一次包含在 PC 中的程序存储器地址加 1。若使用间接寻址指定数据存储器地址则每次重复时就可以访问新的数据存储器地址；若使用直接寻址方式指定的数据存储器地址是常数，重复时不会对其进行修改。
MACD	操作与 MAC 相同，增加了（数据存储器地址）→数据存储器地址+1	3	直接/间接	该指令与 MAC 的不同之处在于片内增加了数据的移动
MPY	（TREG）×（数据存储器地址）→PREG（TREG）×k →PREG	1	直接/间接	TREG 的内容乘以被寻址的数据存储单元的内容；使用短立即数时 TREG 的内容乘以有符号的 13 为常数。短立即数值右对齐，相乘前无任 SXM 如何都将该常数用符号扩展。
助记符	**指令功能**	**周期**	**寻址方式**	**指令说明**
MPYA	ACC）+移位后的（PREG）→ACC （TREG）×（数据存储器地址）→PREG	1	直接/间接	TREG 的内容乘以被寻址的数据存储单元的内容；按 PM 状态位指定的方式对先前的乘积进行移位，并将移位后的值加到 ACC 中。
MPYS	（ACC）-移位后的（PREG）→ACC（TREG）×（数据存储器地址）→PREG	1	直接/间接	TREG 的内容乘以被寻址的数据存储单元的内容；按 PM 状态位指定的方式对先前的乘积进行移位，并从 ACC 中减去移位后的值，结果放在 ACC 中。
MPYU	无符号数（TREG）×无符号数（数据存储器地址）→PREG	1	直接/间接	TREG 的无符号数乘以被寻址的数据存储单元的无符号数；作无符号相乘不能使用 PM=3 这种移位模式，因为此时该移位器总要将 PREG 的值进行符号扩展。
PAC	移位后的（PREG）→ACC	1		按指定的方式将 PREG 的内容移位，并把移位后的结果加载到累加器。
SPAC	（ACC）-移位后的（PREG）→ACC	1		SPAC 不受 SXM 影响，PREG 的值总是要作符号扩展
SPH	（PREG）移位后的高 16 位→数据存储器地址	1	直接/间接	按 PM 指定的方式把 SPEG 的内容移位,移位后的高 16 位数值存到数据存储单元。若右移 6 位则高位用符号扩展，低位丢失;若左移则高位丢失而低位填 0。PREG 和 ACC 中的值都保持不变。
SPL	（PREG）移位后的低 16 位→数据存储器地址	1	直接/间接	按 PM 指定的方式把 SPEG 的内容移位,移位后的低 16 位数值存到数据存储单元。若右移 6 位则高位用符号扩展，低位丢失;若左移则高

				位丢失而低位填 0。PREG 和 ACC 中的值都保持不变。
SPM	constant→乘积移位模式（PM）位	1		指令字中的最低 2 位拷贝到状态寄存器 ST1 的乘积移位模式（PM）位。PM 控制 PREG 输出移位器的移位模式。
SQRA	（ACC）+（PREG）移位后的值→ACC （数据存储器地址）→TREG （TREG）×（数据存储器地址）→PREG	1	直接/间接	按 PM 位指定的方式把 PREG 的内容移位，并将其加到累加器。被寻址的数据存储单元的值加载到 TREG，计算其平方。
SQRS	（ACC）-（PREG）移位后的值→ACC （数据存储器地址）→TREG （TREG）×（数据存储器地址）→PREG	1	直接/间接	按 PM 位指定的方式把 PREG 的内容移位，累加器减去移位后的值。被寻址的数据存储单元的值加载到 TREG，计算其平方。

表 5 转移指令

助记符	指令功能	周期	指令说明
B	16 位程序存储器地址→PC 间接寻址	4	按指令要求的方式修改当前辅助寄存器和 ARP 的内容，把控制转到指令指定的程序存储器地址。
BACC	ACC（15:0）→PC	4	控制转换到 ACC 的低位字所指定的地址。
BANZ	若（当前 AR）≠0,则（16 位程序存储器地址）→PC 否则（PC）+2→PC	4 2	如果当前辅助寄存器的内容不为 0，程序控制转到指定的程序存储器地址；否则执行下一条指令。对当前 AR 缺省的修改方式是减 1。
助记符	**指令功能**	**周期**	**指令说明**
BCND	若条件都满足则（16 位程序存储器地址）→PC 否则（PC）+2→PC	4 2	如果指定的条件都满足则分支到指定的程序存储器地址。注意不是所有的条件组合都有意义。 条件 EQ 表示 ACC=0;NEQ 表示 ACC≠0; 条件 LT 表示 ACC<0;LEQ 表示 ACC<=0; 条件 GT 表示 ACC>0;GEQ 表示 ACC>=0; 条件 NC 表示 C=0;C 表示 C=1; 条件 NOV 表示 OV=0;OV 表示 OV=1; 条件 BIO 引脚为低表示 C=0;C 表示 C=1; 条件 NTC 表示 TC=0;TC 表示 TC=1; UNC 标志无条件。
CALA	PC+1→TOS ACC（15:0）→PC	4	当前程序计数器（PC）加 1 后压入栈顶（TOS）。再将累加器的低位字加载到 PC，程序从该地址继续运行。
CALL	PC+2→TOS （16 位程序存储器地址）→PC 按指令要求的方式修改（AR）和（ARP）	4	当前程序计数器（PC）加 2 后压入栈顶（TOS）。再将程序存储器地址的内容加载到 PC。 该指令为间接寻址方式。
CC	若条件都满足则 PC+2→TOS 且（16 位程序存储器地址）→PC 否则 PC+2	4 2	若指定的条件都满足则控制转到指定的程序存储器地址。同样应注意不是所有的条件组合都有意义。
INTR	（PC）+1→堆栈 相应的中断向量地址→PC	4	处理器有 32 个中断向量,每个 K（0-31）值代表一个中断向量。该指令是软件中断，它使程序控制转换到与 K 值对应的地址,有该地址引导到相应的中断服务程序。INTM 位和中断屏蔽位都不影响 INTR 指令。
NMI	（PC）+1→堆栈 24H→PC	4	该指令将程序计数器强置为不可屏蔽中断向量地址 24H。它与硬件不可屏蔽中断 NMI 的效果相同。

	1→INTM		
RET	TOS）→PC 堆栈上弹一级	4	子程序和中断服务程序以 RET 指令结束,使程序控制返回到调用程序或被中断的程序。
RETC	如果条件都满足则（TOS）→PC 同时堆栈上弹一级，否则继续	4 2	如果条件都满足则执行标准的返回但不是所有的条件组合都有意义。
TRAP	（PC）+1→堆栈 22h→PC	4	该指令是软件中断使控制转换到程序存储单元 22h，PC 加 1 后推入硬件堆栈。在程序空间 22h 可以存放分支命令使控制转到 TRAP 例程。TRAP 指令是不可屏蔽的。

注：对表中的条件指令，如果条件成立，则所需的指令周期为 4，如果条件不成立，则所需的周期为 2。

表 6 控制指令

助记符	指令功能	周期	寻址方式	指令说明
BIT	（数据 bit number（15-（bit code））→TC	1	直接/间接	该指令将数据存储单元中被指定位的值拷贝到状态寄存器 ST1 中的 TC 位。指令中 bit code 的值 0-15 分别对应数据存储器数值 bit number 的第 15-0 位。
BITT	（数据 bit number（15-TREG（3:0））→TC	1	直接/间接	该指令将数据存储单元中被指定位的值拷贝到状态寄存器 ST1 中的 TC 位。数据寄存器数值的 bit number 由 TREG 中低 4 位包含的 bit code 的值指定。TREG 的低 4 位与 bit number 的对应方法同 bit code 与 bit number 的对应相同。
CLRC	0→控制位	1		把指定的控制位清 0。也可用 LST 指令加载 ST0,ST1。控制位有：C，CNF，INTM，OVM，SXM，TC，XF。
助记符	**指令功能**	**周期**	**寻址方式**	**指令说明**
IDEL	PC 加 1,等待没被屏蔽的或不可屏蔽的硬件中断	2		该指令使程序停止运行,直至 CPU 接收到没被屏蔽的硬件中断（内部或外部的）,NMI 或复位。该指令使 C2xx 进入降功耗模式。C2xx 在进入降功耗模式前将 PC 加 1；在空闲状态中 PC 不变。即使 INTM 为 1 未被屏蔽的中断也可使 C2xx 退出空闲状态。当未被屏蔽的中断使 C2xx 退出空闲状态后，CPU 的工作取决于 INTM：若 INTM 为 0,程序分支到它所响应的中断服务程序。若 INTM 为 1,程序从 IDLE 后面的指令继续执行。因 NMI 或复位退出空闲状态时，则不管 INTM 为何值都执行相应的中断服务程序
LDP	（数据寄存器地址）的低 9 位→DP 或 k→DP	2 2	直接/间接 短立即数	被寻址的数据寄存单元的低 9 位或 9 位的立即数值加载到状态寄存器 ST0 中的 DP。也可以用 LST 指令加载 DP。
LST	（数据存储器地址）→状态寄存器 STm	2	直接/间接	用被寻址的数据寄存器的值加载状态寄存器但应注意：1）LST #0 操作不影响 ST1 寄存器中的 ARB 字段。2）LST #1 操作加载到 ARB 的值也加载到 ARP。3）状态寄存器中的保留位总是读 1,写入对其无影响。4）间接寻址时若操作数指定下一 AR 值则用所寻址的数据存储单元中的。高 3 位加载 ARP。子程序调用或中断后可用 LST 指令恢复状态寄存器。
NOP	PC 加 1 即 PC=PC+1	1		该指令只影响 PC，不作其它任何操作。它可用于建立流水线和延时。
POP	（TOS）→ACC（15:0） 0→ACC（31:16） 堆栈上弹一级	1	直接/间接	硬件堆栈是后进先出的 8 个单元。若弹出的次数多于 7 次则堆栈中的所有值将都相同。没有检查堆栈是否下溢的措施。
POPD	（TOS）→数据存储器	1	直接/间接	栈顶的值弹出并传送到指令指定的数据存储单元。其

	地址 堆栈上弹一级			它操作同 POP。没有检查堆栈是否下溢的措施。
PSHD	（数据存储器地址） →（TOS） 堆栈下压一级	1		将指令指定的数据存储单元的内容传送到栈顶。堆栈中低 7 个单元的值都下移一级。堆栈中最低一级单元的内容丢失。
PUSH	堆栈下压一级 ACC（15:0）→TOS	1		累加器的低位字节拷贝到硬件堆栈的栈顶。若弹出前压入的次数多于 8 次，则随着每次的压入堆栈的第一个值将丢失。
RPT	（数据存储器地址） →RPTC k→RPTC	1 1	直接/间接 短立即数	直接或间接寻址时,被寻址的数据存储单元的内容加载到重复计数器（RPTC）;使用短立即数寻址时,8 位立即数加载到 RPTC。RPT 后面的那条指令重复 n 次.n 为 RPTC 的初始值加 1。该指令是不可中断的，器件复位时 RPTC 被清 0。重复指令本身不能重复。
SETC	1→control bit	1		把指定的控制位置 1。也可用 LST 指令加载 ST0 和 ST1。
SPM	constant→乘积移位模式 PM 位 constant 为 0-3	1	直接/间接	指令字中的最低 2 位拷贝到状态寄存器 ST1 的乘积移位模式 PM 位（ST1 的 位 1 和位 0。
SST	（状态寄存器 TSm）→数据寄存器地址	1	直接/间接	直接寻址时,不论 ST0 中的数据页面指针 DP 为何值,指定的状态寄存器总是被保存到 0 页；不必修改 DP，执行该指令时处理器自动访问 0 页。使用间接寻址时，由所选用的辅助寄存器指定存储地址;因此状态寄存器的，值可以保存到数据寄存器的任何也面内。

表 7 I/0 和存储器指令

助记符	指令功能	周期	指令说明
BLDD	PC=PC+1 （PC）→MSTACK，lk→PC，（源）→目的 间接寻址时按指令指定的方式修改（AR）和（ARP） （PC）+1→PC 当（重复计数器）≠0 时 （重复计数器）-1→重复计数器（MSTACK）→PC	3	把“源”指定的数据存储器中的字拷贝到“目的”指定的数据存储单元。源和目的地址可由长立即数地址或数据存储器地址指定。但必须是：如果源地址为长立即数，则目的地址只能为直接或间接；如果源地址为直接或间接，则目的地址只能为长立即数。该指令不能用于存储器映射的寄存器。使用 RPT 指令重复 BLDD 操作期间中断被禁止。
BLPD	PC=PC+1 （PC）→MSTACK，pma→PC，（源）→目的 间接寻址时按指令指定的方式修改（AR）和（ARP （PC）+1→PC，当（重复计数器）≠0 时 （重复计数器）-1→重复计数器 （MSTACK）→PC	3	把’源’指定的程序存储器中的字拷贝到’目的’指定的数据存储单元。源空间的第一个字由长立即数指定.目的地址可由辅助寄存器或数据存储器地址指定。使用 RPT 指令重复 BLDD 操作期间中断被禁止。
DMOV	数据存储器地址）→数据存储器地址+1 直接或间接寻址	1	指定的数据存储单元的内容拷贝到地址加 1 的单元,拷贝时原来单元中的内容保持不变。该指令只能用于片内数据 RAM 块。可以工作于任何配置的 RAM 块但要求这些 RAM 块配置为数据存储器。数据的移动可以跨越块的边界进行。此功能不能用于外部数据存储器。若指令指定了外部存储器地址，该指令读指定的

			存储单元但不进行其它操作。数据移动对于实现 DSP 总的 Z^（-1）延时很有用处。
IN	PA→地址总线 A15-A0 数据总线 D15-D0→数据存储器地址 （PA）→数据存储器地址	2	该指令从 I/O 单元读入 16 位值,将其送到指定的数据存储单元。IS 线变低表示进行 I/O 访问。直接或间接寻址。
OUT	PA→地址总线 A15-A0 （数据存储器地址）→数据总线 D15-D0 数据存储器地址→（PA）	3	该指令将数据存储单元中的 16 位值写到指定的 I/O 端口。IS 线变低表示进行 I/O 访问。直接或间接寻址。
SPLK	lk→数据存储器地址，直接或间接寻址	2	将 16 位常数写到任意的数据存储单元。
TBLR	PC→PC+1 ，（PC）→MSTACK （ACC（15:0））→PC， （pma）→数据存储器地址 间接寻址时按指令指定的方式修改（AR）和（ARP） （PC）+1→PC， 当（重复计数器）≠0 时 （重复计数器）-1→重复计数器 （MSTACK）→PC	3	该指令将程序存储单元中的一个字传送到指令指定的数据存储单元。程序存储单元中的地址由累加器的低 16 位指定。该指令先从程序存储单元读出然后写入指定的数据存储单元。直接或间接寻址。
TBLW	PC→PC+1，（PC+1）→MSTACK （ACC（15:0））→PC+1 ， 数据存储器地址→（pma） 间接寻址时按指令指定的方式修改（AR）和（ARP） （PC）+1→PC ， 当（重复计数器）≠0 时 （重复计数器）-1→重复计数器 （MSTACK）→PC+1	3	该指令将数据存储单元中的一个字传送到指令指定的程序存储单元。数据存储单元的地址由指令指定,程序存储器地址由累加器的低 16 位确定。该指令先从数据存储单元读出然后写入指定的程序存储单元。直接或间接寻址。

3 典型指令说明

本节在指令集的基础上，以几个常用的指令为例，说明了在编程过程中怎样使用指令集中给出的指令。在举例时，直接寻址时一律认为 DP 指针已经指向要寻址的数据区，就不用再重新装载 DP，而间接寻址时则认为 ARP 已经指到当前 AR，从而也不用再单独声明当前 AR 的值。

1 对累加器的加操作 ADD 指令

ADD 指令执行的操作是将数据存储器单元的数 或立即数左移后加至累加器。移位时，低位填 0，高位在 SXM＝1 时为符号扩展，在 SXM＝0 时填 0。结果存在累加器中。寻址短立即数时，加操作不受 SXM 的影响，且不能重复执行。

```
ADD     5，2            ；（DP=4：200－280h）将数据存储器单元 205 的内容左移 2
                        ；位之后与 ACC 相加，结果存在 ACC
ADD     *+，2，AR0      ；（ARP=4，AR4=282）将数据存储器单元 282 的内容左移 2 位之
                        ；后加至 ACC，结果存在 ACC，指令执行后 AR4=283，ARP＝0
ADD     #2              ；短立即数 2 与 ACC 相加，结果存在 ACC
ADD     #1111h，2       ；长立即数 1111h 与 ACC 相加，结果存在 ACC
```

2 和累加器逻辑“与”操作指令 AND

AND 指令用来实现被寻址单元的内容和连接器的逻辑“与”操作，以及长立即数经过移位之后和连接器进行逻辑“与”操作。逻辑“与”操作之后的结果保存在累加器中。

```
AND     16              ；（DP＝4：200－27Fh）将数据存储器单元 210h 的内容与
                        ；ACC 的内容进行逻辑“与”操作，结果保留在 ACC 中。
AND     *               ；（ARP=0，AR0=301h）将数据存储器单元 301h 的内容与 ACC 的
                        ；内容进行逻辑“与”操作，结果保留在 ACC 中。
AND     #00FFh，4       ；将立即数＃0FFh 左移 4 位之后和 ACC 逻辑“与”，结果保留在
                        ；ACC 中。
```

4 条件转移指令 BCND

当所规定的条件符合时，控制转移到指定的程序存储器地址。

```
BCND    P1，LEQ         ；若 ACC 的内容小于等于零时，程序转到 P1 处开始执行。
```

5 位测试指令 BIT

该指令将数据存储器中的指定位的值复制到状态寄存器 ST1 的 TC 位。将该指令和 BCND 指令结合可判断指定位的状态，并根据该位的状态来控制程序的转移。

```
BIT     0h，15          ；（DP=6）测试 300h 处的最低有效位。
        BCND    P1，TC          ；若该位为 1，则程序转到 P1 处执行。
```

7 清除控制位指令 CLRC

CLRC 指令将指定的控制位清除为 0。指定的控制位为下述控制位之一：

C	状态寄存器 ST1 的进位位
CNF	状态寄存器 ST1 的 RAM 配置控制位
INTM	状态寄存器 ST0 的中断方式位
OVM	状态寄存器 ST0 的溢出方式位
SXM	状态寄存器 ST1 的符号扩展方式位
TC	状态寄存器 ST1 的测试/控制标志位
XF	状态寄存器 ST1 的 XF 引脚状态位

```
CLRC    TC              ；将 ST1 的 TC 位清零。
```

注：用 LST 指令也可装入 ST0 和 ST1 寄存器。

9 装载累加器的 LACC 指令

LACC 指令执行的操作是将指定的数据存储器单元的内容或一个 16 位常量左移后送入累加器。移位时，低位填 0，高位在 SXM＝1 时为符号扩展，在 SXM＝0 时填 0。

```
LACC    5，4            ；（DP=8：400－47fh）将数据存储器单元 405 的内容左移 4 位之
                        ；后送到 ACC。
LACC    *，4            ；（ARP=2，AR2=305h）将数据存储器单元 305 的内容左移 4 位之
                        ；后送到 ACC。
LACC    #1234h，2       ；将长立即数 1234h 左移 2 位之后送到 ACC。
```

10 装载累加器低位并清零累加器高位指令 LACL

LACL 指令将被寻址数据存储器单元的内容或者被零扩展的 8 位常量装入累加器的低 16 位，累加器的高半部分填零。数据被作为无符号的 16 位处理，而非二进制补码。无论 SXM 为何状态，该指令的操作数抑制符号扩展。

```
LACL    #10h            ；将 10h 装载入 ACC。
LACL    1               ；（DP＝6：300h－37Fh）将数据存储器单元 301h 的内容装载入
                        ；ACC
LACL    *-，AR4         ；（ARP=0，AR0=301h，（301h）＝2）将数据存储器单元 301h 的
                        ；内容装载入 ACC，指令执行完后 AR0=302h，ARP=4。
```

11 修改辅助寄存器指令 MAR 和装载辅助寄存器指令 LAR

MAR 指令用来修改辅助寄存器 ARP 的值,该指令在直接寻址方式下相当于 NOP 指令。LAR 指令用来将数据存储器的值装载入辅助寄存器。LAR 和 SAR 指令可在子程序调用或中断处理时装载和存储辅助寄存器，从而实现在中断或子程序调用时上下文的保存。

```
MAR  *，AR1             ；向 ARP 装入 1。
MAR  *+，AR5；将当前辅助寄存器（AR1）增 1，并向 ARP 装入 5。
LAR         AR1，5H     ；（DP=4：200－280h）将数据存储器地址 205 的内容装入 AR1 寄
                        ；存器。
LAR         AR1，#50H   ；将短立即数 50h 装入 AR1 寄存器。
LAR         AR1，#1234H ；将长立即数 1234h 装入 AR1 寄存器。
```

12 装载数据页指针指令 LDP

该指令将被寻址数据存储器单元的 9 位最低有效位或 9 位立即数转入状态寄存器 ST0 的数据页指针 DP。DP 也可由 LST 指令装入。

```
LDP   5                 ；(DP=5：地址 280h－2FFh)。
```

17 重复执行下一条指令 RPT

若使用直接或间接寻址，则被寻址的数据存储器单元中的值送入重复计数器（RPTC）；若使用短立即数寻址，则 8 位立即数送入 RPTC。紧接 RPT 后的那条指令被执行 n 次，n 为 RPTC 初值加 1。由于在上下文切换时不能保存 RPTC 的值，所以重复循环被认为是多周期指令，它不能被中断。器件复位时，RPTC 被清零。

```
PRT     #20             ；执行 NOP 指令 21 次。
        NOP
```

18 移位并存储累加器高位指令 SACH

SACH 指令将整个累加器复制到输出移位寄存器中，然后全部 32 位数左移 0 至 7 位，再将移位后数值的高 16 位复制到数据存储器。在移位时，低位填零，高位丢失，累加器内容不变。

```
SACH    10h，1          ；（DP＝4：200h－27Fh）将 ACC 的左移一位，高 16 位存至数据
```

```
                                ；存储器单元 20Ah 中。
SACH    *+，AR2                 ；（ARP=1）将 ACC 的高 16 位存至 AR1 指向的数据存储器单
                                ；元，操作完成之后 ARP=2。
```

19 移位并存储累加器低位指令 SACL

SACL 指令将整个累加器复制到输出移位寄存器中，然后全部 32 位数左移 0 至 7 位，再将移位后数值的低 16 位复制到数据存储器。在移位时，低位填零，高位丢失，累加器内容不变。

```
SACL    10h，1                  ；（DP＝4：200h－27Fh）将 ACC 的左移一位，低 16 位存至数据
                                ；存储器单元 20Ah 中。
SACH    *+，AR2                 ；（ARP=1）将 ACC 的高 16 位存至 AR1 指向的数据存储器单
                                ；元，操作完成之后 ARP=2。
```

22 设置控制位指令 SETC

SETC 指令设置指定的控制位为 1。LST 指令也可用于装载 ST0 和 ST1 寄存器。指定的控制位为下述控制位之一：

C 状态寄存器 ST1 的进位位

CNF 状态寄存器 ST1 的 RAM 配置控制位

INTM 状态寄存器 ST0 的中断方式位

OVM 状态寄存器 ST0 的溢出方式位

SXM 状态寄存器 ST1 的符号扩展方式位

TC 状态寄存器 ST1 的测试/控制标志位

XF 状态寄存器 ST1 的 XF 引脚状态位

```
SETC    TC              ；将 ST1 的 TC 位置 1。
```

23 存储长立即数至数据存储器指令 SPLK

SPLK 指令将一个 16 位值写入任何一个数据存储器单元。在直接寻址方式下使用该指令对数据存储器单元赋值时，通常需要将数据页指针 DP 指向该数据存储器单元所在的数据页。

```
SPLK #30h，5             ；（DP=4）将 30h 存至数据存储器单元 205h 处。
SPLK #1122h，*+，AR4
                         指令执行前：ARP=0，AR0＝400h，（400h）＝0h，
                         指令执行后：ARP=4，AR0＝401h，（400h）＝1122h
```

由于篇幅有限，本节以几个比较典型的指令为例，介绍了编程过程中指令的使用方法，需要注意的是直接寻址时 DP 指针一定要定位到要访问的数据存储区，间接寻址时当前 ARP 必须设置正确，否则不能正确访问单元。关于 TMS320C24X 指令集每条指令的详细介绍请参考 TMS320C24x DSP Controllers Reference Guide CPU and Instruction Set。