

# 三 ARM指令集

---

## 1.存储器访问(L/S)指令

### 常规

LDR R2,[R5] ;将R5为地址的存储单元中的数据加载到R2中。

STR R1,[R0,#0x04] ;将R1中的内容存放在以 R0 + 0x04 的内存地址单元中。

### 传送指令类型:

LDRB R3,[R2],#1 ;以R2为地址读取一个字节数据到R3中。

STRH R1, [R0,#2]! ;半字送达,传送R1中低两字节数据至R0+2为地址的存储单元, R0更新。

### 多寄存器补充:

在非用户或系统模式下, 可出现"^"后缀, 若LDM指令寄存器列表包含PC, 则会额外将SPSR拷贝给CPSR。

## 2.数据处理类指令

### 例:

```
MOV R1,R0 ;将R0的数据放进R1中;  
MOV R1,R0,LSL #3;将R0的数值成8赋值给R1;  
MVN R0,#0; 将立即数0取反传赋值给R0;
```

## 3.算术逻辑运算指令

### 例1: 64为整数加法

R0/R1与R2/R3分别放两个加数的低/高32位, R4/R5存放结果的低/高32位

ADDS R4,R0,R2 ;带S后缀结果会影响CPSR中的标志位C，如果不带S那么计算中的  
ADC R5, R1, R3;带进位的加法，C标志位参与运算。

## 例2: 64为整数减法

SUBS R4,R4,R2  
SBC R5,R1,R3

## 例3: 逆向减法

RSB R0,R1,R2 ; $R0 = R2 - R1$ 。  
RSC R0,R1,R2 ; $R0 = R2 - R1 - C$ 标志位的反码。

## 例4: 逆向减法

AND R0,R0,#3 ;保持R0的0位和1位，其余清0。  
ORR R0,R0,#3 ;置位R0的0位和1位，其余不变。  
EOR R0,R0,#3 ;反转R0的0位和1位，其余不变。  
BIC R0,R0,#3 ;清0 R0的0位和1位，其余不变。

## 例5: 比较指令

CMP R1,R0 ; $R1 - R0$ ,结果影响CPSR中的标志位,但是不保留运算的结果。  
CMN R0, #1 ;判断R0的数值是否是1的补码,如果是则Z置位。

## 例6: 测试指令

TST R1,#3 ;按位与,结果影响CPSR中的标志位。  
TEQ R1,R2 ;按位或,结果影响CPSR中的标志位。

## 例7: 乘法指令

**MUL:** 32位乘法。

**MLA:** 3位操作数,将操作数1与操作数2相乘,结果加第三个操作数,存入目的寄存器。

```
MAL Rd,Rm,Rs,Rn ;Rd = Rm * Rs + Rn。
```

**规则:** Rd和Rm不能是同一寄存器;

**形成两个矢量的标志量积例程:**

```
MOV R11 ,#20 ;计数即代表矢量维度
MOV R10,#0 ;初始化结果寄存器
LOOP:
    LDR R0,[R8],#4 ;读取矢量1指针
    LDR R1,[R9],#4 ;读取矢量2指针
    MAL R10,R0,R1,R10
    SUBS R11,R11,R10
    BNE LOOP
```

## 4.跳转指令

**含义:**跳转指令用于控制程序的走向,可以完成从当前指令向前或向后4字节的地址空间跳转,包括基本跳转指令,带返回跳转指令BL,带状态切换(ARM与Thumb之间)的跳转指令BX,带返回和状态切换的跳转指令BLX

**例:**

```
BL LABEL ;程序无条件跳转到LABEL处执行
          ;同时将PC数值存放到R14中

MOV LR,PC
B LABEL ;利用B指令也可以完成上面的操作
```

此外还有不受范围限制的跳转指令:

```
LDR PC, =LABEL
```

## 5.程序状态寄存器访问指令

CPSR[31:24]: \_f(标志域)

CPSR[23:16]: \_s(状态域)

CPSR[15:08]: \_x(扩展域)

CPSR[07:00]: \_c(控制域)

例:

清CPSR标志位

MSR R0, CPSR ;将CPSR的数值赋值给R0;MSR指令用于CPSR赋值给寄存器

BIC R0, R0, #0xF0000000 ;清高四位

MSR CPSR\_f, R0 ;将R0的数值赋给CPSR, 因为带了标志域, 则指令只会修改31-24

结尾:

初学ARM汇编将其分段整理成笔记供自己参考也供与大家学习, 如有错误请大佬们直言指出, 如果感觉有用那就点个赞留个言, 谢谢观众老爷们的赏脸。

若想获得上述内容的PDF版本移步到GitHub下载。

地址: <https://github.com/QianquanChina/Study-Notes>

-----缙缙