

# ARM指令寻址方式

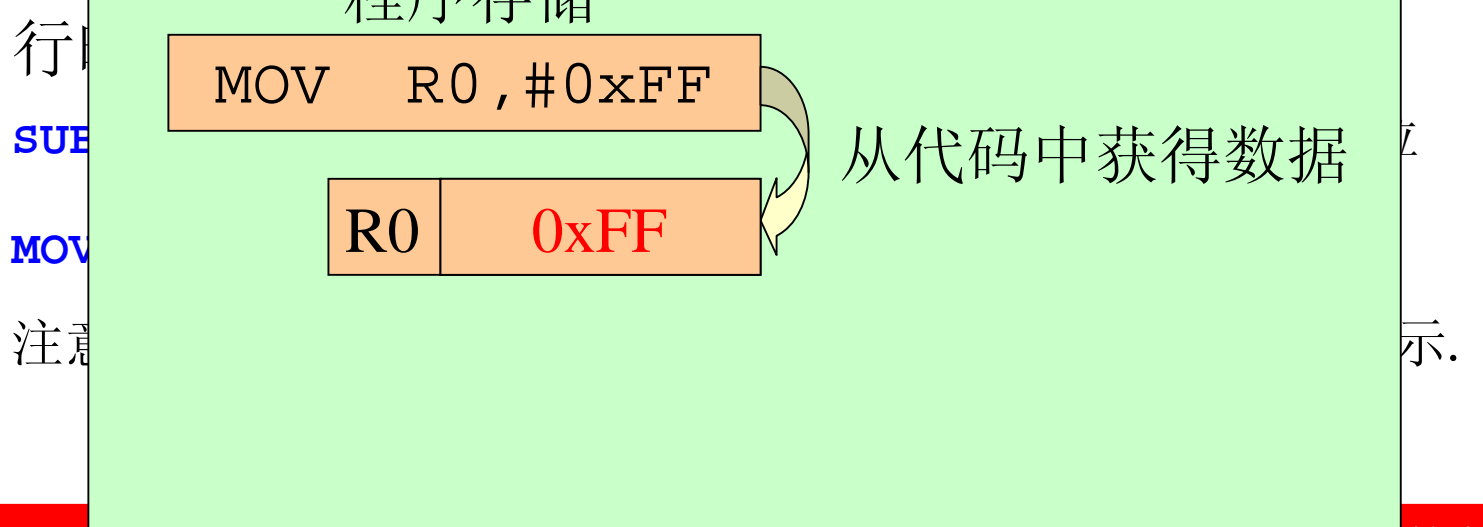
### 3.2 ARM指令的寻址方式

寻址方式是根据指令中给出的地址码字段来实现寻找真实操作数地址的方式。ARM处理器具有7种基本寻址方式。

- |            |          |
|------------|----------|
| 1.立即寻址;    | 2.寄存器寻址; |
| 3.寄存器间接寻址; | 4.基址寻址;  |
| 5.相对寻址;    | 6.堆栈寻址;  |
| 7.块拷贝寻址。   |          |

### 3.2.1 寻址方式分类——立即寻址

立即寻址也称为立即数寻址，是一种特殊的寻址方式。操作数是直接通过指令给出，数据就包含在指令中。

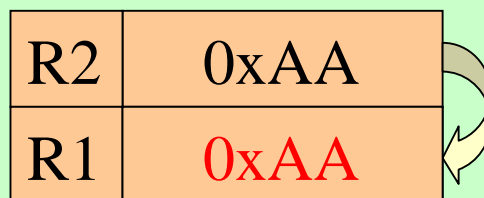


### 3.2.2 寻址方式分类——寄存器寻址

操作数的值在寄存器中，指令中的地址码字段指出的是寄存器编号，指令执行时直接取出寄存器值来操作。

MOV R1

SUB R0

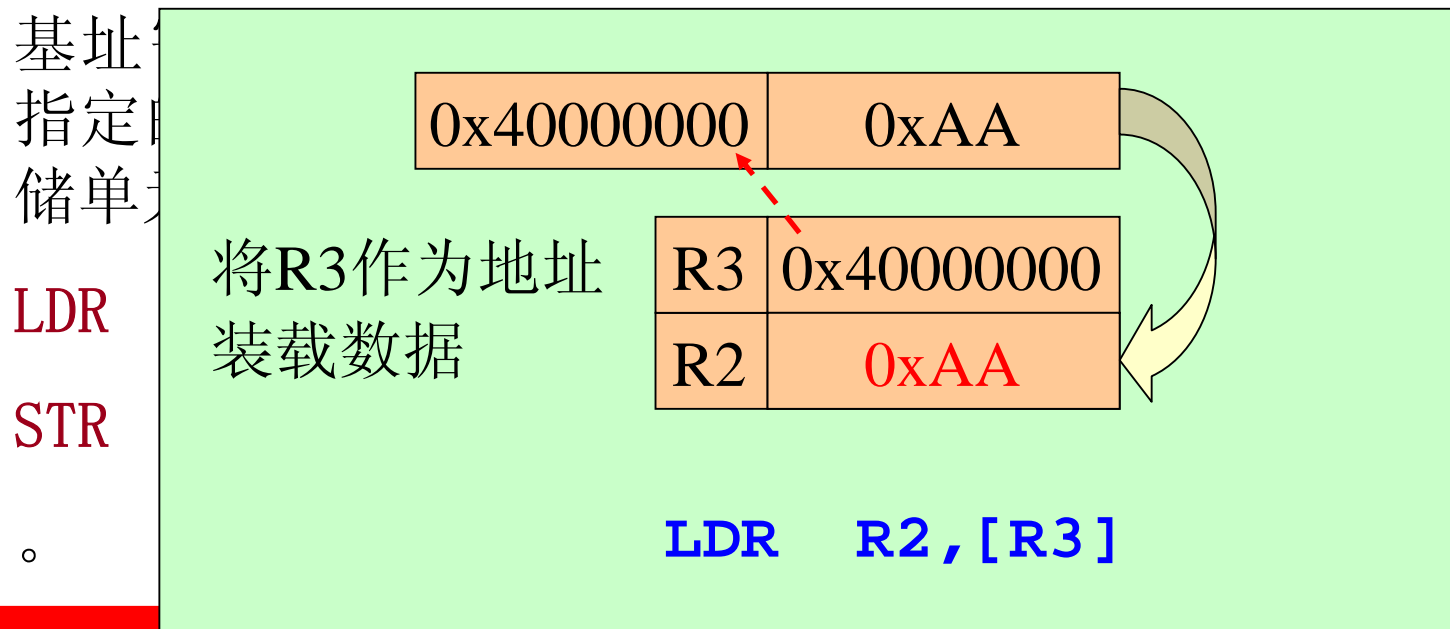


到R0

MOV R1, R2

### 3.2.3 寻址方式分类——寄存器间接寻址

ARM的数据传送指令都是基于寄存器间接寻址，即通过Load / Store完成对数据的传送操作。寄存器间接寻址利用一个寄存器的值(这个寄存器相当于指针的作用，在基址加变址的寻址方式中，它作为基址



### 3.2.4 寻址方式分类——基址寻址

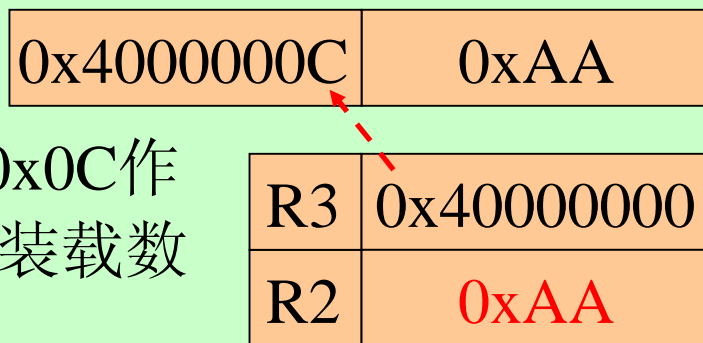
基址寻址就是将基址寄存器的内容与指令中给出的偏移量相加，形成操作数的有效地址。基址寻址用于访问基址附近的存储单元，常用于查表、数组操作、功能部件寄存器访问等。基址寻址指令举例如

LDR

STR R

LDR

将R3+0x0C作为地址装载数据



LDR R2, [R3, #0x0C]

## 3.2.4 寻址方式分类——基址寻址

前索引  
寻址**LDR R0,[R1,#4] ;R0←[R1+4]****LDR R0,[R1,#4]! ;R0←[R1+4], R1←R1+4**后索引  
寻址**LDR R0,[R1],#4 ;R0←[R1],R1←R1+4**

**前索引寻址**是将基址与偏移量相加作为传送数据的地址，传送数据后自动将数据的地址传送给基址寄存器。

**后索引寻址**是将基址作为传送数据的地址，传送数据后自动将基址的内容与偏移量相加传送给基址寄存器。

**自动变址**惊叹号“!”表示在完成数据传送后将更新基址寄存器，更新的方式是每执行完一次操作，基址寄存器自动加上前变址的字节数。

### 3.2.5 寻址方式分类——相对寻址

相对寻址是基址寻址的一种变通。由程序计数器PC提供基准地址，指令中的地址码字段作为偏移量，两者相加后得到的地址即为操作数的有效地址。相对寻址指令举例如下：

	BL	SUBR1	; 调用到SUBR1子程序
	BEQ	LOOP	; 条件跳转到LOOP标号处
	...		
LOOP	MOV	R6, #1	
	...		
SUBR1	...		

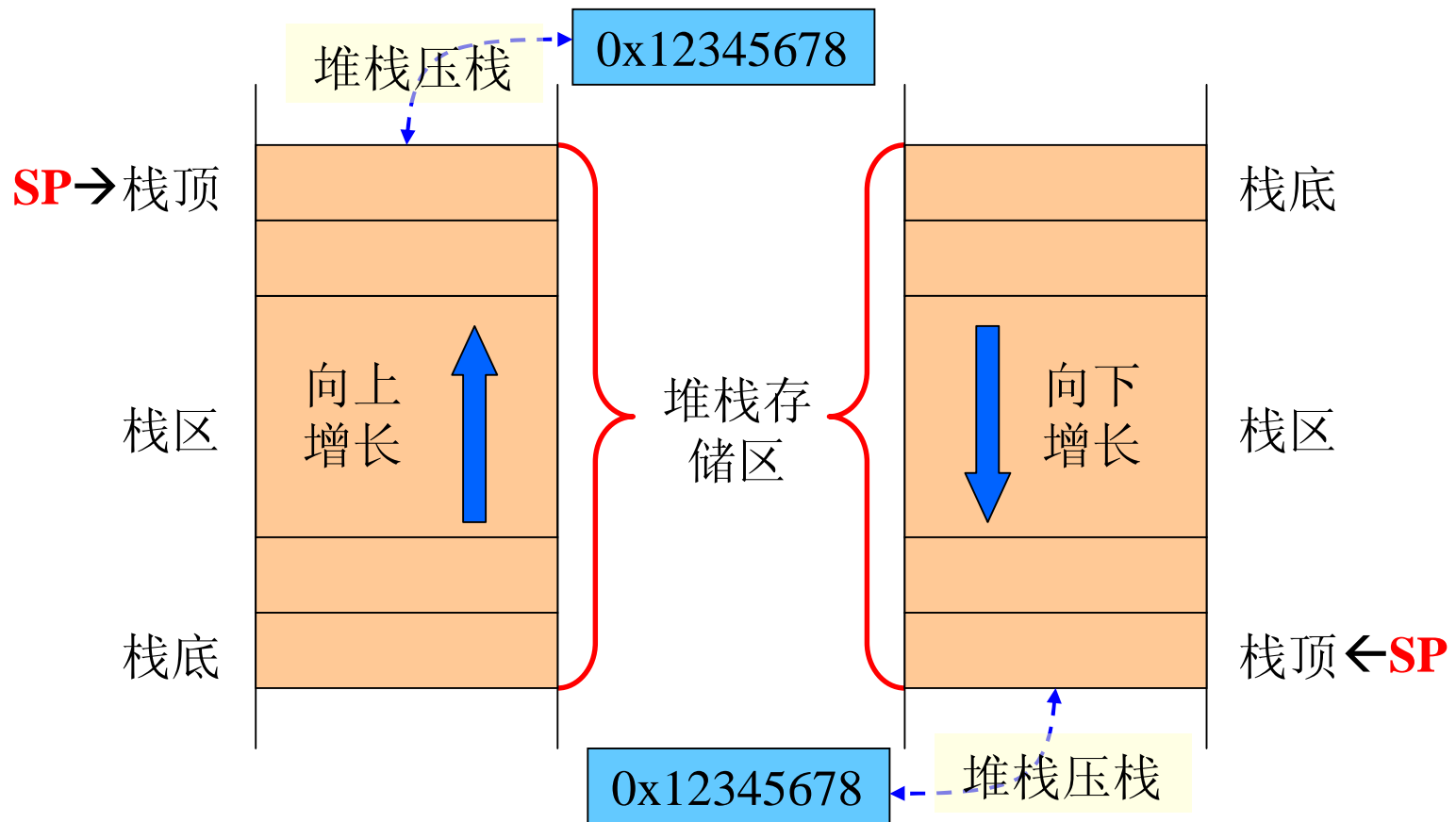


### 3.2.6 寻址方式分类——堆栈寻址

堆栈是一个按特定顺序进行存取的存储区，操作顺序为“后进先出”。堆栈寻址是隐含的，它使用一个专门的寄存器(堆栈指针)指向一块存储区域(堆栈)，指针所指向的存储单元即是堆栈的栈顶。存储器堆栈可分为两种：

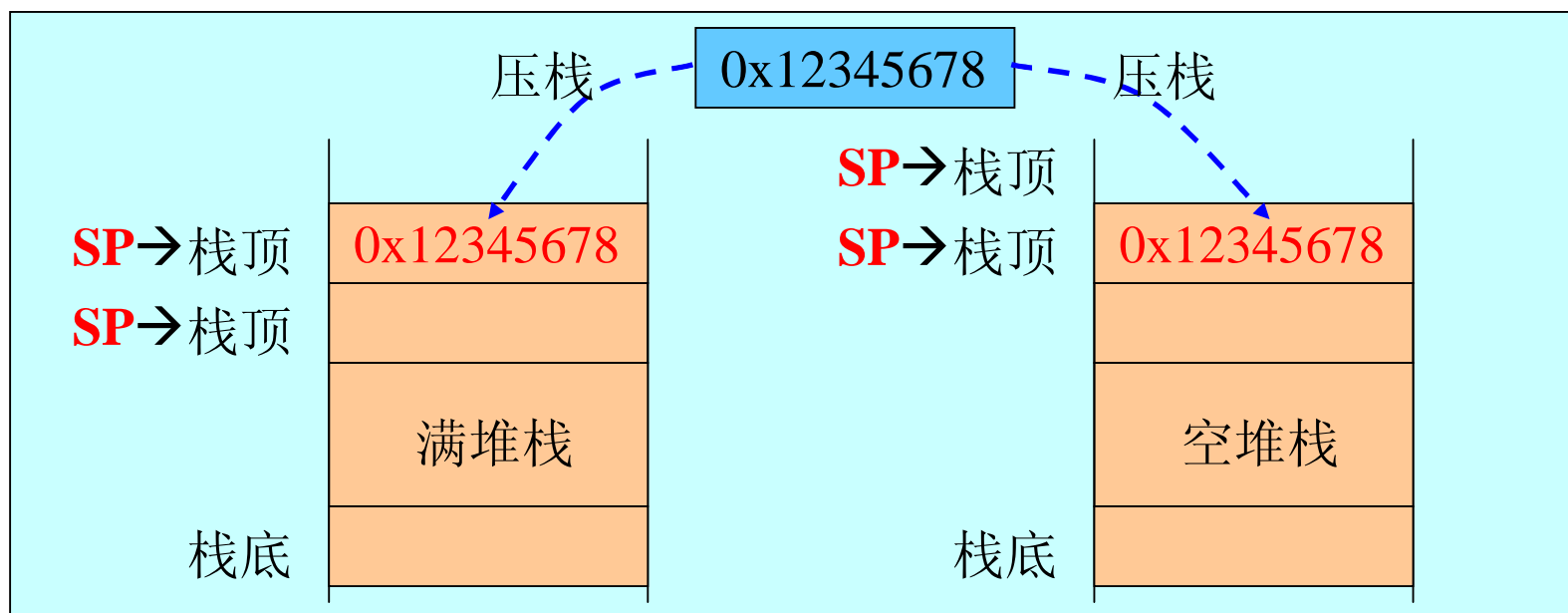
- 向上生长：向高地址方向生长，称为递增堆栈
- 向下生长：向低地址方向生长，称为递减堆栈

### 3.2.6 寻址方式分类——堆栈寻址



## 3.2.6 寻址方式分类——堆栈寻址

堆栈指针指向最后压入的堆栈的有效数据项，称为**满堆栈**；堆栈指针指向下一个待压入数据的空位置，称为**空堆栈**。



### 3.2.6 寻址方式分类——堆栈寻址

所以可以组合出四种类型的堆栈方式：

- **满递增（FA）**：堆栈向上增长，堆栈指针指向内含有效数据项的最高地址。指令如LDMFA、STMFA等；
- **空递增（EA）**：堆栈向上增长，堆栈指针指向堆栈上的第一个空位置。指令如LDMEA、STMEA等；
- **满递减（FD）**：堆栈向下增长，堆栈指针指向内含有效数据项的最低地址。指令如LDMFD、STMFD等；
- **空递减（ED）**：堆栈向下增长，堆栈指针向堆栈下的第一个空位置。指令如LDMED、STMED等。

### 3.2.6 寻址方式分类——堆栈寻址

**STMFD SP! ,{R1-R7,LR}**

;将R1-R7入栈, 满递减堆栈

**LDMFD SP! ,{R1-R7,LR}**

;数据出栈, 放入R1-R7, LR寄存器,  
;满递减堆栈

### 3.2.7 寻址方式分类——块拷贝寻址

块拷贝是将寄存器内容复制到寄存器的地址所指示的存储器中,需要注意的是在存储第一个值之后存储器地址是增加还是减少. 增值类型如下:

IA---- 每次传送后, 地址加4

IB---- 每次传送前, 地址加4

DA—— 每次传送后, 地址减4

DB—— 每次传送前, 地址减4

### 3.2.7 寻址方式分类——块拷贝寻址

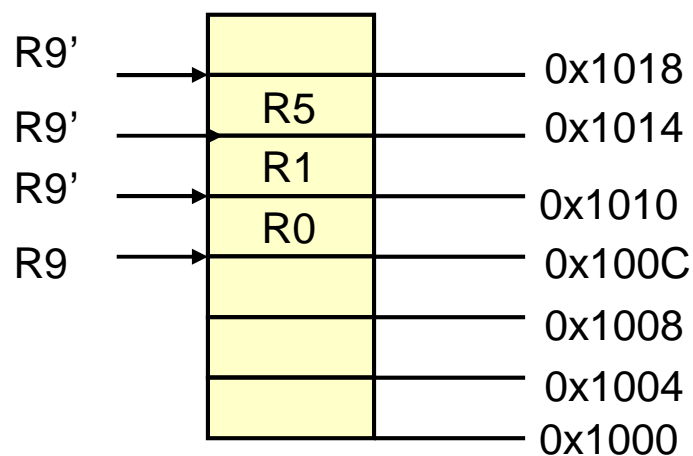
**STMIA R0!, {R1-R7}** ;将R1~R7的数据保存到存储器中, 存储指针在保  
;存第一个值之后增加, 增长方向为向上增长。

**STMIB R0!, {R1-R7}** ;将R1~R7的数据保存到存储器中, 存储指针在保  
;存第一个值之前增加, 增长方向为向上增长。

**STMDA R0!, {R1-R7}** ;将R1~R7的数据保存到存储器中, 存储指针在保存  
;第一个值之后增加,  
增长方向为相下增长。

**STMDB R0!, {R1-R7}** ;将R1~R7的数据保存到存储器中, 存储指针在保  
;存第一个值之前增  
加, 增长方向为相下增长。

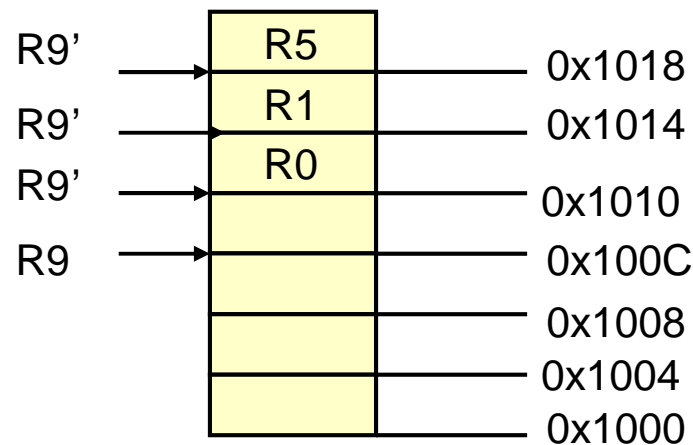
### 3.2.7 寻址方式分类——块拷贝寻址



**STMIA R9!, {R0, R1, R5}**



### 3.2.7 寻址方式分类——块拷贝寻址



**STMIB R9!, {R0, R1, R5}**

**注意：**在IA，IB方式下，寄存器存储的顺序是R0，R1，R5，而在DA，DB方式下，寄存器存储的顺序是R5，R1，R0。在这有一个约定：编号低的寄存器在存储数据或者加载数据时对应于存储器的低地址。也就是说，编号低的寄存器保存到存储器的最低地址或从最低地址取数。

