## Arm 常用指令一览表 V1.1

作者: 李志勇

指令 功能		<del>_</del>		作者: 学志男
Mov t	指令	功能	实例	注释
Movt	Mov	给一个寄存器赋值	Mov r0, #10	R0 = 10
Movt r0, r1   注意: op2 不能大于 16 色。但可以不符合立即数规则,如 op2 可以是			Mov r0, r1	R0 = r1
Myn P	Movt	给一个寄存器的高 16 位赋值	Movt r0, #10	把一个 16 位的 op2 放到 r0 的[31-16],r0 的[15-0]不变
Mvn       把一个数值按位取反后赋值给一个。存得器       Mvn r0, r1       R0 = ~0xff         Add       计算两个数值的加法       Add r0, r0, #10       R0 = r0 + r0         Adc       带进位的加法       Adc r0, r0, #10       R0 = r0 + r1         Adc       带进位的加法       Adc r0, r0, #10       R0 = r0 + r1 + C         Sub       计算两个数值的减法       Sub r0, r0, #10       R0 = r0 - r1         Sbc       带借位的减法       Sbc r0, r0, #10       R0 = r0 - r1         Sbc r0, r0, r1       R0 = r0 - r1 - !C         Rsb       反转减法       Rsb r0, r0, r1       R0 = r1 - r0         Rsc       带借位的反转减法       Rsc r0, r0, r1       R0 = r1 - r0 - !C         Rsc       中借位的反转减法       Rsc r0, r0, r1       R0 = r1 - r0 - !C         Mu1       乘法       M0 + 0, r1, r2       R0 = r1 + r2 + r3         Mla       乘加       M1, r0, r1, r2, r3       R0 = r1 * r2 + r3         Mls       0rr r0, r0, r1, r2, r3       R0 = r3 - r1 * r2         Orr       按位或       0rr r0, r0, r1       R0 = r0   r1         Orr r0, r0, r1, r0, r1       R0 = r0   r1         Orr r0, r0, r1       R0 = r0   r1 </td <td></td> <td></td> <td>Movt r0, r1</td> <td>注意: op2 不能大于 16 位、但可以不符合立即数规则,如 op2 可以是</td>			Movt r0, r1	注意: op2 不能大于 16 位、但可以不符合立即数规则,如 op2 可以是
一个寄存器				0x1234, op1 必须是可读可写的
Add       计算两个数值的加法       Add r0, r0, #10 R0 = r0 + r1         Adc       带进位的加法       Adc r0, r0, #10 R0 = r0 + f0 + C R0 = r0 + f0 + C R0 = r0 + r1 + C         Sub       计算两个数值的减法       Sub r0, r0, #10 R0 = r0 - r1         Sub       Sub r0, r0, #10 Sub r0, r0, r1 R0 = r0 - r1         Sbc       带借位的减法       Sbc r0, r0, #10 R0 = r0 - r1 - !C         Rsb       反转减法       Rsb r0, r0, #1 R0 = r1 - r0 R0 = r1 - r0         Rsc r0, r0, r1 R0 = r1 - r0 - !C       Rsc r0, r0, r1 R0 = r1 - r0 - !C         Mu1 乘法       Mol r0, r1, r2 R0 = r1 * r2         Mu1 乘加       Mu1 r9, r1, r2, r3 R0 = r1 * r2 + r3         Mls r0, r1, r2, r3 R0 = r3 - r1 * r2         Orr 按位或       Orr r0, r0, r1 R0 = r0   r1         Orr r0, r0, #10 R0 = r0   r0         Eor r0, r0, r1 R0 = r0 r1         And r0, r0, r1 R0 = r0 & r1	Mvn	把一个数值按位取反后赋值给	Mvn r0, #0xff	
Add r0, r0, r1		一个寄存器	Mvn r0, r1	
Adc       带进位的加法       Adc r0, r0, #10 Adc r0, r1       R0 = r0 + r1 + C         Sub       计算两个数值的减法       Sub r0, r0, #10 R0 = r0 - r1         Sub r0, r0, r1       R0 = r0 - r1         Sub r0, r0, r1       R0 = r0 - r1         Sbc r0, r0, #10 R0 = r0 - r1 - !C         Rsb Q转减法       Rsb r0, r0, #10 R0 = r1 - r0         Rsc 带借位的反转减法       Rsc r0, r0, #10 R0 = r1 - r0 - !C         Rsc 带借位的反转减法       Rsc r0, r0, #10 R0 = r1 - r0 - !C         Mu1 乘法       Mn1 r0, r1, r2       R0 = r1 * r2         Mla 乘加       Mn1 r0, r1, r2, r3       R0 = r1 * r2 + r3         Mls 死滅       Mn1 r0, r1, r2, r3       R0 = r3 - r1 * r2         Orr 按位或       Orr r0, r0, r1       R0 = r0   10         Eor       按位异或       Eor r0, r0, #10 R0 = r0   10         Eor r0, r0, r1       R0 = r0 * r1         And       按位与       And r0, r0, r1       R0 = r0 & r1	Add	计算两个数值的加法	Add r0, r0, #10	R0 = 10 + 10
Adc r0, r0, r1			Add r0, r0, r1	
Sub       计算两个数值的减法       Sub r0, r0, #10	Adc	带进位的加法	Adc r0, r0, #10	R0 = r0 + 10 + C
Sbc       带借位的减法       Sbc r0, r0, r1       R0 = r0 - r1         Sbc       带借位的减法       Sbc r0, r0, r1       R0 = r0 - r1 - !C         Rsb       反转减法       Rsb r0, r0, r1       R0 = r1 - r0         Rsc       带借位的反转减法       Rsc r0, r0, r1       R0 = r1 - r0 - !C         Rsc       Rsc r0, r0, #10       R0 = r1 - r0 - !C         Rsc r0, r0, #10       R0 = r1 - r0 - !C         Rsc r0, r0, #10       R0 = r1 - r2 - !C         Mul       乘法       Mul r0, r1, r2       R0 = r1 * r2         M1a       乘加       Mul r0, r1, r2, r3       R0 = r1 * r2 + r3         M1s       乘减       M1s r0, r1, r2, r3       R0 = r3 - r1 * r2         Orr       按位或       Orr r0, r0, r1       R0 = r0   r1         Orr r0, r0, #10       R0 = r0   10         Eor       按位异或       Eor r0, r0, #10       R0 = r0 ^ 10         Eor r0, r0, r1       R0 = r0 ^ r1         And       Y0与       And r0, r0, r1       R0 = r0 & r1				
Sbc       带借位的减法       Sbc r0, r0, #10       R0 = r0 - 10 - !C         Rsb       反转减法       Rsb r0, r0, r1       R0 = r0 - r1 - !C         Rsb       Rsb r0, r0, r1       R0 = r1 - r0         Rsc       带借位的反转减法       Rsc r0, r0, r1       R0 = r1 - r0 - !C         Rsc       Rsc r0, r0, #10       R0 = r1 - r0 - !C         Mul       乘法       Mol r0, r1, r2       R0 = r1 * r2         Mla       乘加       Mul r0, r1, r2, r3       R0 = r1 * r2 + r3         Mls       乘減       Mls r0, r1, r2, r3       R0 = r3 - r1 * r2         Orr       按位或       Orr r0, r0, r1       R0 = r0   r1         Orr r0, r0, #10       R0 = r0   10         Eor       按位异或       Eor r0, r0, #10       R0 = r0 ^ 10         Eor r0, r0, r1       R0 = r0 ^ r1         And       按位与       And r0, r0, r1       R0 = r0 & r1	Sub	计算两个数值的减法	Sub r0, r0, #10	R0 = r0 - 10
Sbc r0, r0, r1				
Rsb	Sbc	带借位的减法	Sbc r0, r0, #10	R0 = r0 - 10 - !C
Rsc 带借位的反转减法 Rsc r0, r0, #10 R0 = 10 - r0  Rsc 带借位的反转减法 Rsc r0, r0, r1 R0 = r1 - r0 - !C  Mul 乘法 Mul r0, r1, r2 R0 = r1 * r2  Mla 乘加 Mul r0, r1, r2, r3 R0 = r1 * r2 + r3  Mls 乘减 Mls r0, r1, r2, r3 R0 = r3 - r1 * r2  Orr 按位或 Orr r0, r0, r1 R0 = r0   r1  Orr r0, r0, #10 R0 = r0   10  Eor 按位异或 Eor r0, r0, r1 R0 = r0 ^ 10  Eor r0, r0, r1 R0 = r0 ^ 10  And r0, r0, r1 R0 = r0 & r1  R0 = r0 & r1  R0 = r0 & r1  R0 = r0 ~ r1				
Rsc 带借位的反转减法	Rsb	反转减法		
Rsc r0, r0, #10				
Mul       乘法       Mul $r0$ , $r1$ , $r2$ R0 = $r1 * r2$ Mla       乘加       Mul $r0$ , $r1$ , $r2$ , $r3$ R0 = $r1 * r2 + r3$ Mls       乘減       Mls $r0$ , $r1$ , $r2$ , $r3$ R0 = $r3 - r1 * r2$ Orr       接位或       Orr $r0$ , $r0$ , $r1$ R0 = $r0$   $r1$ Orr $r0$ , $r0$ , $r1$ R0 = $r0$   $r0$ Eor       接位异或       Eor $r0$ , $r0$ , $r1$ R0 = $r0$ ^ $r1$ And       按位与       And $r0$ , $r0$ , $r1$ R0 = $r0$ & $r1$	Rsc	带借位的反转减法		
Mla       乘加       Mul r0, r1, r2, r3       R0 = r1 * r2 + r3         Mls       乘減       Mls r0, r1, r2, r3       R0 = r3 - r1 * r2         Orr       按位或       Orr r0, r0, r1       R0 = r0   r1         Orr r0, r0, #10       R0 = r0   10         Eor       按位异或       Eor r0, r0, #10       R0 = r0 ^ 10         Eor r0, r0, r1       R0 = r0 ^ r1         And       按位与       And r0, r0, r1       R0 = r0 & r1				
Mls       乘減       Mls r0, r1, r2, r3       R0 = r3 - r1 * r2         Orr       按位或       Orr r0, r0, r1 Orr r0, r0, #10       R0 = r0   r1 R0 = r0   10         Eor       接位异或       Eor r0, r0, #10 R0 = r0 ^ 10 R0 = r0 ^ r1         And       接位与       And r0, r0, r1       R0 = r0 & r1         R0 = r0 * r1       R0 = r0 * r1				R0 = r1 * r2
Orr       接位或       Orr r0, r0, r1 Orr r0, r0, #10       R0 = r0   r1 R0 = r0   10         Eor       接位异或       Eor r0, r0, #10 R0 = r0 10 R0 = r0 r1         And       接位与       And r0, r0, r1 R0 = r0 & r1				
Eor       按位异或       Eor r0, r0, #10	Mls			R0 = r3 - r1 * r2
Eor       接位异或       Eor r0, r0, #10 Eor r0, r0, r1       R0 = r0 ^ 10 R0 = r0 ^ r1         And       接位与       And r0, r0, r1       R0 = r0 ^ r1         R0 = r0 ^ r1       R0 = r0 & r1	0rr	按位或		
Eor r0, r0, r1       R0 = r0 r1         And 按位与       And r0, r0, r1       R0 = r0 & r1         R0 = r0 & r1       R0 = r0 & r1				'
And 按位与 And r0, r0, r1 R0 = r0 & r1	Eor	按位异或	· · ·	
And r0, r0, $\#10$ $R0 = r0 \& 10$	And	按位与	' '	
			And r0, r0, #10	R0 = r0 & 10

Bic	位取反	Bic r0, r0, r1	$R0 = r0 & (^{\sim}r1)$
		Bic r0, r0, #10	$R0 = r0 & (^10)$
Lsr	逻辑右移	RO, 1sr r1	R0 >>> r1
		RO, 1sr #10	R0 >>> 1
Lsl	逻辑左移	RO, 1s1 r1	R0 <<< r1
		RO, 1s1 #1	R0 <<< 1
Asr	算术右移	RO, asr r1	$R0 \gg r1$
		RO, asr #1	R0 >> 1
Ror	循环右移	RO, ror r1	$(R0 \gg r1) \mid (R0 \ll (32 - r1))$
		R0, ror #1	(R0 >>> 1)  (R0 <<< (32 - 1))
Cmp	比较	Cmp r0, r1	RO - r1 影响 cpsr 标志位
		Cmp r0, #10	RO - 10 影响 cpsr 标志位
Teq	比较(按位异或)	Teq r0, r1	RO ^ r1 影响 cpsr 标志位
		Teq r0, #10	RO ^ 10 影响 cpsr 标志位
Tst	比较(按位与)	Tst r0, r1	RO & r1 影响 cpsr 标志位
		Tst r0, #10	RO & 10 影响 cpsr 标志位
Mrs	读 cpsr	Mrs r0, cpsr	RO = cpsr
Msr	写 cpsr	Msr cpsr, r0	Cpsr = r0
Swi	软中断	Swi 10	产生软中断异常
Svc	等同于 swi	Svc 10	产生软中断异常
Ldr	把数据从内存加载的寄存器	Ldr r0, addr	RO = *addr
		ldr r0, =addr	R0 = addr
		ldr r1, [r0]	R1 = *r0
		ldr r1, [r0, #4]	R1 = *(r0 + 4)
		ldr r1, [r0, #4]!	R1 = *(r0 + 4); r0 += 4
		ldr rl, [r0], #4	R1 = *r0; r0 += 4
Str	把数据从寄存器保存的内存	Str r0, addr	*addr = r0
		Str r1, [r0]	*r0 = r1
		Str r1, [r0, #4]	*(r0 + 4) = r1
		Str r1, [r0, #4]!	*(r0 + 4) = r1; r0 += 4
		Str r1, [r0], #4	*r0 = r1; r0 += 4
Ldm	把数据从内存加载的寄存器	Ldmfd sp!, {r0-r12, 1r}	把寄存器的值放到慢递减栈中
Stm	把数据从寄存器保存的内存	Stmfd sp!, {r0-r12, 1r}	从慢递减栈中把值取到寄存器

Push	压栈	Push {r0-r12, 1r}	把寄存器的值放到慢递减栈中
Pop	出栈	Pop {r0-r12, 1r}	从慢递减栈中把值取到寄存器
b	跳转	B lable	跳到 lable 处执行
B1	跳转并保存返回地址	Bl lable	保存下一条指令的地址到 lr,并跳转到 lable 处执行
Bx	跳转(可切换状态)	Bx r0	跳转到 r0 所指的位置执行
Clz	计算一个数值高位零的个数	Clz r0, r1	计算 r1 中开头的零的个数,把计算结果放到 r0
Qadd	饱和加法	Qadd r0, r0, r1	运算结果的饱和到[0x80000000, 0x7fffffff]
Qadd8	饱和 8 位加法	Qadd8 r0, r0, r1	r1 和 r2 的每一个字节分别相加,饱和到[-2^7, 2^7-1]
			注意:每一个操作数都是寄存器,不影响Q位
Qadd16	饱和 16 位加法	Qadd16 r0, r0, r1	r1 和 r2 的每一个 16 位相加,饱和到[-2^15, 2^15-1]
			注意:每一个操作数都是寄存器,不影响Q位
Qsub	饱和减法	Qsub r0, r0, r1	运算的结果饱和到[0x80000000, 0x7fffffff]
Qsub16	饱和 16 位减法	Qsub16 r0, r0, r1	r1 和 r2 的每一个 16 位分别相减,饱和到[-2^15, 2^15-1]
			注意:每一个操作数都是寄存器 ,不影响 Q 位
Qsub8	饱和8位减法	Qsub8 r0, r0, r1	r1 和 r2 的每一个字节分别相减,饱和到[-2^7, 2^7-1]
			注意:每一个操作数都是寄存器 ,不影响 Q 位
Ssat	有符号饱和	Ssat r0, #sat, r1	1<=sat<=32
			把 r1 饱和到[-2^(sat-1), 2^(sat-1)-1], 结果放到 r0, 如果饱和会置
		4	位 Q
			注意: 把 r1 饱和到 sat 个位,饱和后符号不变
Ssat16	有符号 16 位饱和	Ssat16 r0, #sat, r1	1<=sat<=16
		• 7	把 r1 中的每一个 16 位饱和到[-2^(sat-1), 2^(sat-1)-1], 结果放到
		<b>Y</b>	r0,如果饱和会置位 Q
		AIY	注意: 把 r1 中的每一个 16 位饱和到 sat 个位,饱和后符号不变
Usat	无符号饱和	Usat r0, #sat, r1	0<=sat<=31
		<b>Y</b>	把 r1 饱和到[0, 2 <sup>^</sup> sat-1], 结果放到 r0, 如果饱和会置位 Q
			注意: 负数饱和到 0
Usat16	无符号 16 位饱和	Usat16 r0, #sat, r1	usat16 r0, #sat, r1
			0<=sat<=31
			把 r1 中的两个 16 位分别饱和到[0, 2 <sup>sat-1</sup> ], 结果放到 r0, 如果饱和
			会置位 Q
-			注意: 负数饱和到 0
Rev	大小端转换	Rev r0, r1	把 r1 进行大小端转换,结果放到 r0

Rev16	16 位大小端转换	Rev16 r0, r1	把 r1 的每个 16 位进行大小端转换,结果放到 r0
Revsh	16 位大小端转换并有符号扩展	Revsh r0, r1	把 r1 的低 16 位进行大小端转换,并扩展为一个 32 位的有符号数
Rbit	位反转	Rbit r0, r1	把 r1 进行位顺序翻转,结果放到 r0
Uxtb16	无符号 8 位扩展 16 位	Uxtb16 r0, r1	把 r1 中的两个 8 位数无符号扩展为两个 16 位数,结果放到 r0
Uxtb	无符号 8 位扩展 32 位	Uxtb8 r0, r1	把 r1 中的 8 位数无符号扩展为一个 32 位数,结果放到 r0
Uxth	无符号 16 位扩展 32 位	Uxth r0, r1	把 r1 中的 16 位数无符号扩展为一个 32 位数,结果放到 r0
Sxtb16	有符号 8 位扩展 16 位	Sxtb16 r0, r1	把 r1 中的两个 8 位数有符号扩展为两个 16 位数,结果放到 r0
Sxtb	有符号 8 位扩展 32 位	Sxtb8 r0, r1	把 r1 中的 8 位数有符号扩展为一个 32 位数,结果放到 r0
Sxth	有符号 16 位扩展 32 位	Sxth r0, r1	把 r1 中的 16 位数有符号扩展为一个 32 位数,结果放到 r0

注意: 这里并不是 arm 指令集的全部,只是 arm 指令集中比较常用的指令,如想查看全部指令集,请到 arm 官方网站自行下载: www. arm. com

