

向量浮点指令集
快速参考卡

表关键字						
{C}	请参阅表 条件字段 F32（单精度）或 F64（双精度）。 单精度、双精度或半精度（F16）。 单精度或双精度浮点。 有符号或无符号整数。 FPSCR 或 FPSID。 2 VFPv2 及更高版本。3 VFPv3 及更高版本。3H VFPv3 及更高版本，带半精度扩展。	<P>	<fpconst>	+/- m * 2 ⁻ⁿ ，其中 m 和 n 为整数，16 <= m <= 31，0 <= n <= 7	Sd、Sn、Sm（单精度），或 Dd、Dn、Dm（双精度）。 E 出现任何 NaN 时都引发异常。无 E 仅当发送 NaN 信号时才引发异常。 使用 FPSCR 舍入模式。否则，向零舍入。 以逗号分隔的 连续 VFP 寄存器列表，括在大括号 { 和 } 内。 固定点数的小数位数，0-16 或 1-32。 S16、S32、U16 或 U32，有符号或无符号，16 位或 32 位。	
<P>			Fd, Fn, Fm	Sd、Sn、Sm（单精度），或 Dd、Dn、Dm（双精度）。		
S、D、H			{E}	E 出现任何 NaN 时都引发异常。无 E 仅当发送 NaN 信号时才引发异常。		
F			{R}	使用 FPSCR 舍入模式。否则，向零舍入。		
SI、UI			<VFPregs>	以逗号分隔的 连续 VFP 寄存器列表，括在大括号 { 和 } 内。		
<VFPsysreg>			<fbits>	固定点数的小数位数，0-16 或 1-32。		
\$			<type>	S16、S32、U16 或 U32，有符号或无符号，16 位或 32 位。		
运算		\$	汇编器	异常	操作	说明
向量运算	乘法		VMUL{C}.<P> Fd, Fn, Fm	IO, OF, UF, IX	Fd := Fn * Fm	
	乘法并求反		VNMUL{C}.<P> Fd, Fn, Fm	IO, OF, UF, IX	Fd := -(Fn * Fm)	
	并累加		VMLA{C}.<P> Fd, Fn, Fm	IO, OF, UF, IX	Fd := Fd + (Fn * Fm)	
	求反并累加		VMLS{C}.<P> Fd, Fn, Fm	IO, OF, UF, IX	Fd := Fd - (Fn * Fm)	
	乘减		VNMLS{C}.<P> Fd, Fn, Fm	IO, OF, UF, IX	Fd := -Fd + (Fn * Fm)	
	求反并相减		VNMLA{C}.<P> Fd, Fn, Fm	IO, OF, UF, IX	Fd := -Fd - (Fn * Fm)	
	加法		VADD{C}.<P> Fd, Fn, Fm	IO, OF, IX	Fd := Fn + Fm	
	减法		VSUB{C}.<P> Fd, Fn, Fm	IO, OF, IX	Fd := Fn - Fm	
	除法		VDIV{C}.<P> Fd, Fn, Fm	IO, DZ, OF, UF, IX	Fd := Fn / Fm	
	绝对值		VABS{C}.<P> Fd, Fm		Fd := abs(Fm)	
	求反		VNEG{C}.<P> Fd, Fm		Fd := -Fm	
	求平方根		VSQRT{C}.<P> Fd, Fm	IO, IX	Fd := sqrt(Fm)	
标量比较	两个值		VCMP{E}{C}.<P> Fd, Fm	IO	设置 Fd - Fm 的 FPSCR 标记	利用 VMRS APSR_nzcv, FPSCR 传送标记。
	值与零比较		VCMP{E}{C}.<P> Fd, #0.0	IO	设置 Fd - 0 的 FPSCR 标记	
标量转换	单精度到双精度		VCVT{C}.F64.F32 Dd, Sm	IO	Dd := convertStoD(Sm)	源代码在 Fd 的低 16 位或 32 位中。 对象代码在 Fd 的低 16 位或 32 位中。 对象代码在 Sd 的高 16 位中 对象代码在 Sd 的低 16 位中 源代码在 Sm 的高 16 位中 源代码在 Sm 的低 16 位中
	双精度到单精度		VCVT{C}.F32.F64 Sd, Dm	IO, OF, UF, IX	Sd := convertDtoS(Dm)	
	无符号整数到浮点数		VCVT{C}.<P>.U32 Fd, Sm	IX	Fd := convertUtoF(Sm)	
	有符号整数到浮点数		VCVT{C}.<P>.S32 Fd, Sm	IX	Fd := convertSitoF(Sm)	
	浮点数到无符号整数		VCVT{R}{C}.U32.<P> Sd, Fm	IO, IX	Sd := convertFtoUI(Fm)	
	浮点数到有符号整数		VCVT{R}{C}.S32.<P> Sd, Fm	IO, IX	Sd := convertFtoSI(Fm)	
	定点到浮点	3	VCVT{C}.<P>.<type> Fd, Fd, #<fbits>	IO, IX	Fd := convert<type>toF(Fd)	
	浮点到定点	3	VCVT{C}.<type>.<P> Fd, Fd, #<fbits>	IO, IX	Fd := convertFto<type>(Fd)	
	单精度到半精度	3H	VCVTT{C}.F16.F32 Sd,Sm	ID, IO, OF, UF, IX	Sd:=convertStoH(Sm)	
	单精度到半精度	3H	VCVTB{C}.F16.F32 Sd,Sm	ID, IO, OF, UF, IX	Sd:=convertStoH(Sm)	
	半精度到单精度	3H	VCVTT{C}.F32.F16 Sd,Sm	ID, IO, OF, UF, IX	Sd:=convertHtoS(Sm)	
	半精度到单精度	3H	VCVTB{C}.F32.F16 Sd,Sm	ID, IO, OF, UF, IX	Sd:=convertHtoS(Sm)	
插入常数	将常数插入寄存器中	3	VMOV{C}.<P> Fd, #<fpconst>		Fd := <fpconst>	
传送寄存器	复制 VFP 寄存器		VMOV{C}.<P> Fd, Fm		Fd := Fm	Sm 必须为 S(n+1) Sm 必须为 S(n+1)
	ARM®到单精度		VMOV{C} Sn, Rd		Sn := Rd	
	单精度到 ARM		VMOV{C} Rd, Sn		Rd := Sn	
	两个 ARM 到两个单精度	2	VMOV{C} Sn, Sm, Rd, Rn		Sn := Rd, Sm := Rn	
	两个单精度到两个 ARM	2	VMOV{C} Rd, Rn, Sn, Sm		Rd := Sn, Rn := Sm	
	两个 ARM 到双精度	2	VMOV{C} Dm, Rd, Rn		Dm[31:0] := Rd, Dm[63:32] := Rn	
	双精度到两个 ARM	2	VMOV{C} Rd, Rn, Dm		Rd := Dm[31:0], Rn := Dm[63:32]	
	ARM 到双精度的低半字		VMOV{C} Dn[0], Rd		Dn[31:0] := Rd	
	双精度的低半字到 ARM		VMOV{C} Rd, Dn[0]		Rd := Dn[31:0]	

向量浮点指令集
快速参考卡

运算		§	汇编器	异常	操作	说明
传送寄存器 (续)	ARM 到双精度的高半字 双精度的高半字到 ARM ARM 到 VFP 系统寄存器 VFP 系统寄存器到 ARM FPSCR 标记到 APSR		VMOV{C} Dn[1], Rd VMOV{C} Rd, Dn[1] VMSR{C} <VFPsysreg>, Rd VMRS{C} Rd, <VFPsysreg> VMRS{C} APSR_nzcv, FPSCR		Dn[63:32] := Rd Rd := Dn[63:32] VFPsysreg := Rd Rd := VFPsysreg APSR flags := FPSCR flags	

运算		§	汇编器	同义词	操作
保存 VFP 寄存器	单个 单个, PC 相对的 多个, 无索引/之后增加 之前减小 推入堆栈		VSTR{C} Fd, [Rn{, #<immed>}] VSTR{C} Fd, <label> VSTM{C} Rn{!}, <VFPregs> VSTMDB{C} Rn!, <VFPregs> VPUSH{C} <VFPregs>	VSTMIA ,VSTMEA VSTMFD (满降序) VSTMFD SP!	[address] := Fd。立即数为 0-1020 范围内的 4 的倍数。 从 Rn 中的地址开始, 保存 VFP 寄存器列表。
加载 VFP 寄存器	单个 单个, PC 相对的 多个, 无索引/之后增加 之前减小 从堆栈中弹出		VLDR{C} Fd, [Rn{, #<immed>}] VLDR{C} Fd, <label> VLDM{C} Rn{!}, <VFPregs> VLDMDB{C} Rn!, <VFPregs> VPOP{C} <VFPregs>	VLDMIA ,VLDMFD VLDMEA (空升序) VLDM SP!	Fd := [address]。立即数为 0-1020 范围内的 4 的倍数。 从 Rn 中的地址开始, 加载 VFP 寄存器列表。

FPSCR 格式								舍入		(跨度 - 1) * 3			向量长度 - 1						异常陷阱启用位						累加异常位						
31	30	29	28	27	26	25	24	23	22	21	20		18	17	16	15			12	11	10	9	8	7			4	3	2	1	0
N	Z	C	V	QC	AHP	DB	FZ	RMODE		STRIDE			LEN			IDE			IXE	UFE	OFE	DZE	IOE	IDC			IXC	UFC	OFC	DZC	IOC
FZ 1 = 清零模式。								舍入: 0 = 舍入到最近, 1 = 向 +∞ 舍入, 2 = 向 -∞ 舍入, 3 = 向零舍入。								对于双精度操作数, (向量长度 * 跨度) 不能超过 4。(已弃用)															

条件字段						异常		
助记符	说明 (VFP)	说明 (ARM 或 Thumb®)		助记符	说明 (VFP)	说明 (ARM 或 Thumb)	ID	非标准输入
EQ	等于	等于		HI	大于或无序	无符号大于	IO	无效操作
NE	不等于或无序	不等于		LS	小于或等于	无符号小于或相同	OF	溢出
CS / HS	大于或等于或无序	进位设置/无符号大于或相同		GE	大于或等于	有符号大于或等于	UF	下溢
CC / LO	小于	进位清零/无符号小于		LT	小于或无序	有符号小于	IX	不精确的结果
MI	小于	求反		GT	大于	有符号大于	DZ	被零除
PL	大于或等于或无序	正数或零		LE	小于或等于或无序	有符号小于或等于		
VS	无序（至少一个非数字操作数）	溢出		AL	始终（通常省略）	始终（通常省略）		
VC	非无序的	无溢出						

所有权声明

除非本所有权声明在下面另有说明, 否则带有 ® 或 ™ 标记的词语和徽标是 ARM Limited 在欧盟和其他国家/地区的注册商标或商标。此处提及的其他品牌和名称可能是其各自所有者的商标。

除非事先得到版权所有人的书面许可, 否则不得以任何形式修改或复制本文档包含的部分或全部信息以及产品说明。

本文档描述的产品还将不断发展和完善。ARM Limited 将如实提供本文档所述产品的所有特性及其使用方法。但是, 所有暗示或明示的担保, 包括但不限于对特定用途适销性或适用性的担保, 均不包括在内。

本参考卡仅旨在帮助读者使用产品。对由于使用本参考卡中的任何信息, 或由于本参考卡的信息错误、遗漏, 以及产品的错误使用所造成的任何损失, ARM Limited 概不负责。

文档编号

ARM QRC 0007E

变更记录

发行号	日期	变更
A	2004 年 11 月	第一版
B	2005 年 5 月	RVCT 2.2 SP1 版
C	2006 年 3 月	RVCT 3.0 版
D	2007 年 3 月	RVCT 3.1 版
E	2008 年 9 月	RVCT 4.0 版