

Vector Floating Point Instruction Set

Quick Reference Card

Key to Tables			
{C}	See Table Condition Field	<fpconst>	$\pm m * 2^{-n}$ where m and n are integers, $16 \leq m \leq 31$, $0 \leq n \leq 7$
<P>	F32 (single precision) or F64 (double precision).	Fd, Fn, Fm	Sd, Sn, Sm (single precision), or Dd, Dn, Dm (double precision).
S, D, H	Single, double, or half-precision (F16).	{E}	E : raise exception on any NaN. Without E : raise exception only on signaling NaNs.
F	Single or double-precision floating point.	{R}	Use FPSCR rounding mode. Otherwise, round towards zero.
SI, UI	Signed or unsigned integer.	<VFPregs>	A comma separated list of <i>consecutive</i> VFP registers, enclosed in braces ({ and }).
<VFPsysreg>	FPSCR or FPSID.	<fbits>	Number of fraction bits in fixed-point number, 0-16 or 1-32.
§	2: VFPv2 and above. 3: VFPv3 and above. 3H: VFPv3 and above with half-precision extension.	<type>	S16, S32, U16, or U32, for Signed or Unsigned, 16-bit or 32-bit.

Operation	§	Assembler	Exceptions	Action	Notes
Vector arithmetic	Multiply and negate and accumulate negate and accumulate and subtract negate and subtract Add Subtract Divide Absolute Negative Square root	VMUL{C}.<P> Fd, Fn, Fm VNML{C}.<P> Fd, Fn, Fm VMLA{C}.<P> Fd, Fn, Fm VMLS{C}.<P> Fd, Fn, Fm VNMLS{C}.<P> Fd, Fn, Fm VNMLA{C}.<P> Fd, Fn, Fm VADD{C}.<P> Fd, Fn, Fm VSUB{C}.<P> Fd, Fn, Fm VDIV{C}.<P> Fd, Fn, Fm VABS{C}.<P> Fd, Fm VNEG{C}.<P> Fd, Fm VSQRT{C}.<P> Fd, Fm	IO, OF, UF, IX IO, OF, UF, IX IO, OF, UF, IX IO, OF, UF, IX IO, OF, UF, IX IO, OF, UF, IX IO, OF, IX IO, OF, IX IO, DZ, OF, UF, IX IO, IX	Fd := Fn * Fm Fd := - (Fn * Fm) Fd := Fd + (Fn * Fm) Fd := Fd - (Fn * Fm) Fd := - Fd + (Fn * Fm) Fd := - Fd - (Fn * Fm) Fd := Fn + Fm Fd := Fn - Fm Fd := Fn / Fm Fd := abs(Fm) Fd := - Fm Fd := sqrt(Fm)	
Scalar compare	Two values Value with zero	VCMP{E}{C}.<P> Fd, Fm VCMP{E}{C}.<P> Fd, #0.0	IO IO	Set FPSCR flags on Fd - Fm Set FPSCR flags on Fd - 0	Use VMRS APSR_nzcv, FPSCR to transfer flags.
Scalar convert	Single to double Double to single Unsigned integer to float Signed integer to float Float to unsigned integer Float to signed integer Fixed-point to float Float to fixed-point Single to half-precision Single to half-precision Half to single-precision Half to single-precision	VCVT{C}.F64.F32 Dd, Sm VCVT{C}.F32.F64 Sd, Dm VCVT{C}.<P>.U32 Fd, Sm VCVT{C}.<P>.S32 Fd, Sm VCVT{R}{C}.U32.<P> Sd, Fm VCVT{R}{C}.S32.<P> Sd, Fm 3 VCVT{C}.<P>.<type> Fd, Fd, #<fbits> 3 VCVT{C}.<type>.<P> Fd, Fd, #<fbits> 3H VCVTT{C}.F16.F32 Sd, Sm 3H VCVTB{C}.F16.F32 Sd, Sm 3H VCVTT{C}.F32.F16 Sd, Sm 3H VCVTB{C}.F32.F16 Sd, Sm	IO IO, OF, UF, IX IX IX IO, IX IO, IX IO, IX IO, IX ID, IO, OF, UF, IX ID, IO, OF, UF, IX ID, IO, OF, UF, IX ID, IO, OF, UF, IX	Dd := convertStoD(Sm) Sd := convertDtoS(Dm) Fd := convertUtoF(Sm) Fd := convertStoF(Sm) Sd := convertFtoUI(Fm) Sd := convertFtoSI(Fm) Fd := convert<type>toF(Fd) Fd := convertFto<type>(Fd) Sd:=convertStoH(Sm) Sd:=convertStoH(Sm) Sd:=convertHtoS(Sm) Sd:=convertHtoS(Sm)	Source is in bottom 16 or 32 bits of Fd. Destination is bottom 16 or 32 bits of Fd. Destination is top 16 bits of Sd Destination is bottom 16 bits of Sd Source is top 16 bits of Sm Source is bottom 16 bits of Sm
Insert constant	Insert constant in register	3 VMOV{C}.<P> Fd, #<fpconst>		Fd := <fpconst>	
Transfer registers	Copy VFP register ARM® to single Single to ARM Two ARM to two singles Two singles to two ARM Two ARM to double Double to two ARM ARM to lower half of double Lower half of double to ARM	VMOV{C}.<P> Fd, Fm VMOV{C} Sn, Rd VMOV{C} Rd, Sn 2 VMOV{C} Sn, Sm, Rd, Rn 2 VMOV{C} Rd, Rn, Sn, Sm 2 VMOV{C} Dm, Rd, Rn 2 VMOV{C} Rd, Rn, Dm VMOV{C} Dn[0], Rd VMOV{C} Rd, Dn[0]		Fd := Fm Sn := Rd Rd := Sn Sn := Rd, Sm := Rn Rd := Sn, Rn := Sm Dm[31:0] := Rd, Dm[63:32] := Rn Rd := Dm[31:0], Rn := Dm[63:32] Dn[31:0] := Rd Rd := Dn[31:0]	Sm must be S(n+1) Sm must be S(n+1)

Vector Floating Point Instruction Set

Quick Reference Card

Operation	§	Assembler	Exceptions	Action	Notes
Transfer registers (continued)	ARM to upper half of double Upper half of double to ARM ARM to VFP system register VFP system register to ARM FPSCR flags to 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	

Operation	§	Assembler	Synonyms	Action
Save VFP registers	Single Single, PC-relative Multiple, unindexed / increment after decrement before Push onto stack	VSTR{C} Fd, [Rn{, #<immed>}] VSTR{C} Fd, <label> VSTM{C} Rn{!}, <VFPregs> VSTMDB{C} Rn!, <VFPregs> VPUSH{C} <VFPregs>	VSTMIA, VSTMEA VSTMFD (full descending) VSTMFD SP!	[address] := Fd. Immediate range 0-1020, multiple of 4. Saves list of VFP registers, starting at address in Rn.
Load VFP registers	Single Single, PC-relative Multiple, unindexed / increment after decrement before Pop from stack	VLDR{C} Fd, [Rn{, #<immed>}] VLDR{C} Fd, <label> VLDM{C} Rn{!}, <VFPregs> VLDMDB{C} Rn!, <VFPregs> VPOP{C} <VFPregs>	VLDMIA, VLDMFD VLDMEA (empty ascending) VLDM SP!	Fd := [address]. Immediate range 0-1020, multiple of 4. Loads list of VFP registers, starting at address in Rn.

FPSCR format								Rounding		(Stride - 1)*3		Vector length - 1			Exception trap enable bits					Cumulative exception bits										
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 = flush to zero mode.								Rounding: 0 = round to nearest, 1 = towards +∞, 2 = towards -∞, 3 = towards zero.					(Vector length * Stride) must not exceed 4 for double precision operands. (Deprecated)																	

Condition Field						Exceptions	
Mnemonic	Description (VFP)	Description (ARM or Thumb®)	Mnemonic	Description (VFP)	Description (ARM or Thumb®)	ID	Input Denormal
EQ	Equal	Equal	HI	Greater than, or unordered	Unsigned higher	IO	Invalid operation
NE	Not equal, or unordered	Not equal	LS	Less than or equal	Unsigned lower or same	OF	Overflow
CS / HS	Greater than or equal, or unordered	Carry Set / Unsigned higher or same	GE	Greater than or equal	Signed greater than or equal	UF	Underflow
CC / LO	Less than	Carry Clear / Unsigned lower	LT	Less than, or unordered	Signed less than	IX	Inexact result
MI	Less than	Negative	GT	Greater than	Signed greater than	DZ	Division by zero
PL	Greater than or equal, or unordered	Positive or zero	LE	Less than or equal, or unordered	Signed less than or equal		
VS	Unordered (at least one NaN operand)	Overflow	AL	Always (normally omitted)	Always (normally omitted)		
VC	Not unordered	No overflow					

Proprietary Notice

Words and logos marked with ® or ™ are registered trademarks or trademarks of ARM Limited in the EU and other countries, except as otherwise stated below in this proprietary notice. Other brands and names mentioned herein may be the trademarks of their respective owners.

Neither the whole nor any part of the information contained in, or the product described in, this document may be adapted or reproduced in any material form except with the prior written permission of the copyright holder.

The product described in this document is subject to continuous developments and improvements. All particulars of the product and its use contained in this document are given by ARM in good faith. However, all warranties implied or expressed, including but not limited to implied warranties of merchantability, or fitness for purpose, are excluded.

This reference card is intended only to assist the reader in the use of the product. ARM Ltd shall not be liable for any loss or damage arising from the use of any information in this reference card, or any error or omission in such information, or any incorrect use of the product.

Document Number

ARM QRC 0007E

Change Log

Issue	Date	Change
A	Nov 2004	First Release
B	May 2005	Release for RVCT 2.2 SP1
C	March 2006	Release for RVCT 3.0
D	March 2007	Release for RVCT 3.1
E	Sept 2008	Release for RVCT 4.0