

Vector Floating Point Instruction Set

Quick Reference Card

Key to Tables				
{cond}	See Table Condition Field		Fd, Fn, Fm	Sd, Sn, Sm (single precision), or Dd, Dn, Dm (double precision).
<S/D>	S (single precision) or D (double precision).		{E}	E : raise exception on any NaN. Without E : raise exception only on signaling NaNs.
<S/D/X>	As above, or X (unspecified precision).		{Z}	Round towards zero. Overrides FPSCR rounding mode.
<VFPsysreg>	FPSCR or FPSID.		<VFPregs>	A comma separated list of <i>consecutive</i> VFP registers, enclosed in braces ({ and }).
§	2: VFPv2 and above. 3: VFPv3 and above.		<BinLoc>	Number of integer bits in fixed-point number, 0-16 or 0-31.
			<type>	SH, SL, UH, or UL, for Signed or Unsigned, Halfword (16-bit) or Longword (32-bit).

Operation		§	Assembler	Exceptions	Action	Notes
Vector arithmetic	Multiply		FMUL<S/D>{cond} Fd, Fn, Fm	IO, OF, UF, IX	Fd := Fn * Fm	
	and negate		FNMUL<S/D>{cond} Fd, Fn, Fm	IO, OF, UF, IX	Fd := – (Fn * Fm)	
	and accumulate		FMAC<S/D>{cond} Fd, Fn, Fm	IO, OF, UF, IX	Fd := Fd + (Fn * Fm)	
	negate and accumulate		FNMAC<S/D>{cond} Fd, Fn, Fm	IO, OF, UF, IX	Fd := Fd – (Fn * Fm)	
	and subtract		FMSC<S/D>{cond} Fd, Fn, Fm	IO, OF, UF, IX	Fd := – Fd + (Fn * Fm)	
	negate and subtract		FNMSC<S/D>{cond} Fd, Fn, Fm	IO, OF, UF, IX	Fd := – Fd – (Fn * Fm)	
	Add		FADD<S/D>{cond} Fd, Fn, Fm	IO, OF, IX	Fd := Fn + Fm	
	Subtract		FSUB<S/D>{cond} Fd, Fn, Fm	IO, OF, IX	Fd := Fn – Fm	
	Divide		FDIV<S/D>{cond} Fd, Fn, Fm	IO, DZ, OF, UF, IX	Fd := Fn / Fm	
	Copy		FCPY<S/D>{cond} Fd, Fm		Fd := Fm	
	Absolute		FABS<S/D>{cond} Fd, Fm		Fd := abs(Fm)	
Scalar compare	Two values		FCMP{E}<S/D>{cond} Fd, Fm	IO	Set FPSCR flags on Fd – Fm	Use FMSTAT to transfer flags.
	Value with zero		FCMP{E}Z<S/D>{cond} Fd	IO	Set FPSCR flags on Fd – 0	Use FMSTAT to transfer flags.
Scalar convert	Single to double		FCVTDS{cond} Dd, Sm	IO	Dd := convertStoD(Sm)	
	Double to single		FCVTSd{cond} Sd, Dm	IO, OF, UF, IX	Sd := convertDtoS(Dm)	
	Unsigned integer to float		FUITO<S/D>{cond} Fd, Sm	IX	Fd := convertUtoF(Sm)	
	Signed integer to float		FSITO<S/D>{cond} Fd, Sm	IX	Fd := convertSIttoF(Sm)	
	Fixed-point to float	3	F<type>TO<S/D>{<cond>} Fd, #<BinLoc>	IO, IX	Fd := convert<type>toF(Fd)	Source is in bottom 16 or 32 bits of Fd.
	Float to unsigned integer		FTOUI{Z}<S/D>{cond} Sd, Fm	IO, IX	Sd := convertFtoUI(Fm)	
	Float to signed integer		FTOSI{Z}<S/D>{cond} Sd, Fm	IO, IX	Sd := convertFtoSI(Fm)	
Insert constant	Float to fixed-point	3	FTO<type><S/D>{<cond>} Fd, #<BinLoc>	IO, IX	Fd := convertFto<type>(Fd)	Destination is bottom 16 or 32 bits of Fd.
	Insert constant in register	3	FCONST{cond} Fd, #<imm8>		Fd := convertIMtoF(imm8)	TBD
Transfer registers	ARM [®] to single		FMSR{cond} Sn, Rd		Sn := Rd	
	Single to ARM		FMRs{cond} Rd, Sn		Rd := Sn	
	Two ARM to two singles	2	FMSRR{cond} {Sn,Sm}, Rd, Rn		Sn := Rd, Sm := Rn	
	Two singles to two ARM	2	FMRRS{cond} Rd, Rn, {Sn,Sm}		Rd := Sn, Rn := Sm	
	Two ARM to double	2	FMDRR{cond} Dn, Rd, Rn		Dn[31:0] := Rd, Dn[63:32] := Rn	
	Double to two ARM	2	FMRRD{cond} Rd, Rn, Dn		Rd := Dn[31:0], Rn := Dn[63:32]	
	ARM to lower half of double		FMDLR{cond} Dn, Rd		Dn[31:0] := Rd	Use with FMDHR.
	Lower half of double to ARM		FMRDL{cond} Rd, Dn		Rd := Dn[31:0]	Use with FMRDH.
	ARM to upper half of double		FMDHR{cond} Dn, Rd		Dn[63:32] := Rd	Use with FMDLR.
	Upper half of double to ARM		FMRDH{cond} Rd, Dn		Rd := Dn[63:32]	Use with FMRDL.
	ARM to VFP system register		FMXR{cond} <VFPsysreg>, Rd		VFPsysreg := Rd	Stalls ARM until all VFP ops complete.
	VFP system register to ARM		FMRX{cond} Rd, <VFPsysreg>		Rd := VFPsysreg	Stalls ARM until all VFP ops complete.
	FPSCR flags to CPSR		FMSTAT{cond}		CPSR flags := FPSCR flags	Equivalent to FMRX R15, FPSCR

Vector Floating Point Instruction Set
Quick Reference Card

Operation		§	Assembler	Action
Save VFP registers	Single		FST<S/D>{cond} Fd, [Rn{, #<immed>}]	[address] := Fd. Immediate range 0-1020, multiple of 4.
	Multiple, unindexed increment after decrement before		FSTMIA<S/D/X>{cond} Rn, <VFPregs> FSTMIA<S/D/X>{cond} Rn!, <VFPregs> FSTMDB<S/D/X>{cond} Rn!, <VFPregs>	Saves list of VFP registers, starting at address in Rn. synonym: FSTMEA (empty ascending) synonym: FSTMFD (full descending)
Load VFP registers	Single		FLD<S/D>{cond} Fd, [Rn{, #<immed>}]	Fd := [address]. Immediate range 0-1020, multiple of 4.
	Multiple, unindexed increment after decrement before		FLDMIA<S/D/X>{cond} Rn, <VFPregs> FLDMIA<S/D/X>{cond} Rn!, <VFPregs> FLDMDB<S/D/X>{cond} Rn!, <VFPregs>	Loads list of VFP registers, starting at address in Rn. synonym: FLDMFD (full descending) synonym: FLDMEA (empty ascending)

FPSCR format								Rounding		(Stride – 1)*3		Vector length – 1			Exception trap enable bits						Cumulative exception bits										
31	30	29	28				24	23	22	21	20		18	17	16				12	11	10	9	8				4	3	2	1	0
N	Z	C	V				FZ	RMODE		STRIDE			LEN						IXE	UFE	OFE	DZE	IOE				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.															

Register banks, and sector, scalar and mixed operations
If Fd is S0-S7, D0-D3, or D16-D19, operation is Scalar (regardless of vector length).
If Fd is S8-S31, D4-D15, or D20-D31, and Fm is S8-S31, D4-D15, or D20-D31, operation is Vector.
If Fd is S8-S31, D4-D15, or D20-D31, and Fm is S0-S7, D0-D3, or D16-D31, operation is Mixed (Fm scalar, others vector).
S0-S7 (or D0-D3), S8-S15 (D4-D7), S16-S23 (D8-D11), S24-S31 (D12-D15), D16-D19, D20-D23, D24-D27, D28-D31, each form a circulating bank of registers.

Exceptions	
IO	Invalid operation
OF	Overflow
UF	Underflow
IX	Inexact result
DZ	Division by zero

Condition Field						
Mnemonic	Description (VFP)	Description (ARM or Thumb®)		Mnemonic	Description (VFP)	Description (ARM or Thumb)
EQ	Equal	Equal		HI	Greater than, or unordered	Unsigned higher
NE	Not equal, or unordered	Not equal		LS	Less than or equal	Unsigned lower or same
CS / HS	Greater than or equal, or unordered	Carry Set / Unsigned higher or same		GE	Greater than or equal	Signed greater than or equal
CC / LO	Less than	Carry Clear / Unsigned lower		LT	Less than, or unordered	Signed less than
MI	Less than	Negative		GT	Greater than	Signed greater than
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 owned by ARM Limited. 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 0007A	Change Log		
Issue	Date	By	Change
A	November 2004	CKS	First Release