(/)

(/) | Products (/product/) | Download (/download/) | Events (/events/) | Search Keil...

Support (/support/) | Videos (http://www2.keil.com/video)

Technical Support

Overview (/support/)

Search (/home/searchhelp)

Contact (/support/contact.asp)

Assistance Request (/support/request.asp)

Feedback (/support/feedback.asp)

On-Line Manuals

Product Manuals (/support/man/)

Document Conventions (/support/man/conventions.asp)

Assembler User Guide

Preface (armasm_deb1353593789871.htm)
Overview of the Assembler (armasm_dom1359731119400.htm)
Overview of the ARM Architecture (armasm_dom1359731124840.htm)
Structure of Assembly Language Modules (armasm_dom1359731140888.htm)
Writing ARM Assembly Language (armasm_dom1359731144635.htm)
Condition Codes (armasm_dom1359731158738.htm)
Using the Assembler (armasm_dom1359731163750.htm)
Symbols, Literals, Expressions, and Operators (armasm_dom1359731172022.htm)
VFP Programming (armasm_pge1423655975021.htm)
Assembler Command-line Options (armasm_dom1361289815333.htm)
ARM and Thumb Instructions (armasm_dom1361289850039.htm)
ARM and Thumb instruction summary (armasm_dom1361289850509.htm)
Instruction width specifiers (armasm_dom1361289851079.htm)
Flexible second operand (Operand2) (armasm_dom1361289851539.htm)
Syntax of Operand2 as a constant (armasm_dom1361289851958.htm)
Syntax of Operand2 as a register with optional shi (armasm_dom1361289852638.htm)
Shift operations (armasm_dom1361289852998.htm)
Saturating instructions (armasm_dom1361289860307.htm)
Condition code suffixes (armasm_dom1361289860997.htm)
ADC (armasm_dom1361289861367.htm)
ADD (armasm_dom1361289861747.htm)
ADR (PC-relative) (armasm_dom1361358903373.htm)
ADR (register-relative) (armasm_dom1361289862147.htm)
ADRL pseudo-instruction (armasm_dom1361289862667.htm)
AND (armasm_dom1361289863017.htm)
ASR (armasm_dom1361289863407.htm)
B (armasm_dom1361289863797.htm)
BFC (armasm_dom1361289864157.htm)
BFI (armasm_dom1361289864536.htm)
BIC (armasm_dom1361289864906.htm)
BKPT (armasm_dom1361289865326.htm)
BL (armasm_dom1361289865686.htm)

Home (/) / Assembler User Guide

(armasm_dom1361289851079.htm) (default.htm) (armasm_dom1361289850039.htm)

AF and Thumb instruction summary

Home (default.htm) » ARM and Thumb Instructions (armasm_dom1361289850039.htm) » ARM and Thu

10.1 ARM and Thumb instruction summary

Different ARM architectures support different sets of ARM and Thumb instructions.

The following table gives a summary of the availability of ARM and Thumb instructions in the ARM architecture:

Table 10-1 Summary of ARM and Thumb instructions

Mnemonic	Brief description	*Arch.* (armasm_dom1361289850509.htm#dom13612898
`ADC`	Add with Carry	All
`ADD`	Add	All
`ADR`	Load program or register-relative address (short range)	All
`ADRL` pseudo-instruction	Load program or register-relative address (medium range)	x6M
`AND`	Logical AND	All
`ASR`	Arithmetic Shift Right	All
`B`	Branch	All
`BFC`	Bit Field Clear	T2
`BFI`	Bit Field Insert	T2
`BIC`	Bit Clear	All
`BKPT`	Breakpoint	5
`BL`	Branch with Link	All
`BLX`	Branch with Link, change instruction set	T
`BX`	Branch, change instruction set	T
`BXJ`	Branch, change to Jazelle®	J, x7M
`CBZ, CBNZ`	Compare and Branch if {Non}Zero	T2
`CDP`	Coprocessor Data Processing operation	x6M
`CDP2`	Coprocessor Data Processing operation	5, x6M
`CLREX`	Clear Exclusive	K, x6M
`CLZ`	Count leading zeros	5, x6M
`CMN , CMP`	Compare Negative, Compare	All
`CPS`	Change Processor State	6

- BLX (armasm_dom1361289866046.htm)
- BX (armasm_dom1361289866466.htm)
- BXJ (armasm_dom1361289866836.htm)
- CBZ and CBNZ (armasm_dom1361289867296.htm)
- CDP and CDP2 (armasm_dom1361289867676.htm)
- CLREX (armasm_dom1361289868046.htm)
- CLZ (armasm_dom1361289868426.htm)
- CMP and CMN (armasm_dom1361289868786.htm)
- CPS (armasm_dom1361289869176.htm)
- CPY pseudo-instruction (armasm_dom1361289869526.htm)
- DBG (armasm_dom1361289869916.htm)
- DMB (armasm_dom1361289870356.htm)
- DSB (armasm_dom1361289870725.htm)
- EOR (armasm_dom1361289871065.htm)
- ERET (armasm_dom1361289871515.htm)
- HVC (armasm_dom1387466087086.htm)
- ISB (armasm_dom1361289871865.htm)
- IT (armasm_dom1361289872225.htm)
- LDC and LDC2 (armasm_dom1361289872695.htm)
- LDM (armasm_dom1361289873045.htm)
- LDR (immediate offset) (armasm_dom1361289873425.htm)
- LDR (PC-relative) (armasm_dom1361289873875.htm)
- LDR (register offset) (armasm_dom1361289874275.htm)
- LDR (register-relative) (armasm_dom1361289874695.htm)
- LDR pseudo-instruction (armasm_dom1361289875065.htm)
- LDR, unprivileged (armasm_dom1361289875455.htm)
- LDREX (armasm_dom1361289875835.htm)
- LSL (armasm_dom1361289876185.htm)
- LSR (armasm_dom1361289876525.htm)
- MCR and MCR2 (armasm_dom1361289877204.htm)
- MCRR and MCRR2 (armasm_dom1361289877604.htm)
- MLA (armasm_dom1361289878324.htm)
- MLS (armasm_dom1361289878654.htm)
- MOV (armasm_dom1361289878994.htm)
- MOV32 pseudo-instruction (armasm_dom1361289879354.htm)
- MOVT (armasm_dom1361289879724.htm)
- MRC and MRC2 (armasm_dom1361289880404.htm)
- MRRC and MRRC2 (armasm_dom1361289880734.htm)
- MRS (PSR to general-purpose register) (armasm_dom1361289881054.htm)
- MRS (system coprocessor register to ARM register) (armasm_dom1361289881374.htm)
- MSR (ARM register to system coprocessor register) (armasm_dom1361289881714.htm)
- MSR (general-purpose register to PSR) (armasm_dom1361289882044.htm)
- MUL (armasm_dom1361289882394.htm)
- MVN (armasm_dom1361289882734.htm)
- NEG pseudo-instruction (armasm_dom1361289883063.htm)
- NOP

CPY pseudo-instruction	Copy	6
DBG	Debug	7
DMB	Data Memory Barrier	7, 6M
DSB	Data Synchronization Barrier	7, 6M
EOR	Exclusive OR	All
ERET	Exception Return	7VE
HVC	Hypervisor Call	7VE
ISB	Instruction Synchronization Barrier	7, 6M
IT	If-Then	T2
LDC	Load Coprocessor	x6M
LDC2	Load Coprocessor	5, x6M
LDM	Load Multiple registers	All
LDR	Load Register with word	All
LDR pseudo-instruction	Load Register pseudo-instruction	All
LDRB	Load Register with byte	All
LDRBT	Load Register with byte, user mode	x6M
LDRD	Load Registers with two words	5E, x6M
LDREX	Load Register Exclusive	6, x6M
LDREXB , LDREXH	Load Register Exclusive Byte, Halfword	K, x6M
LDREXD	Load Register Exclusive Doubleword	K, x7M
LDRH	Load Register with halfword	All
LDRHT	Load Register with halfword, user mode	T2
LDRSB	Load Register with signed byte	All
LDRSBT	Load Register with signed byte, user mode	T2
LDRSH	Load Register with signed halfword	All
LDRSHT	Load Register with signed halfword, user mode	T2
LDRT	Load Register with word, user mode	x6M
LSL	Logical Shift Left	All
LSR	Logical Shift Right	All
MCR	Move from Register to Coprocessor	x6M
MCR2	Move from Register to Coprocessor	5, x6M
MCRR	Move from Registers to Coprocessor	5E, x6M
MCRR2	Move from Registers to Coprocessor	6, x6M

(armasm_dom1361289883473.htm)
ORN (Thumb only)
(armasm_dom1361289883823.htm)
ORR
(armasm_dom1361289884183.htm)
PKHBT and PKHTB
(armasm_dom1361289884513.htm)
PLD and PLI
(armasm_dom1361289884863.htm)
POP
(armasm_dom1361289885303.htm)
PUSH
(armasm_dom1361289885653.htm)
QADD
(armasm_dom1361289885963.htm)
QADD8
(armasm_dom1361289886293.htm)
QADD16
(armasm_dom1361289886623.htm)
QASX
(armasm_dom1361289886933.htm)
QDADD
(armasm_dom1361289887263.htm)
QDSUB
(armasm_dom1361289887583.htm)
QSAX
(armasm_dom1361289887943.htm)
QSUB
(armasm_dom1361289888283.htm)
QSUB8
(armasm_dom1361289888653.htm)
QSUB16
(armasm_dom1361289889003.htm)
RBIT
(armasm_dom1361289889382.htm)
REV
(armasm_dom1361289889712.htm)
REV16
(armasm_dom1361289890072.htm)
REVSH
(armasm_dom1361289890492.htm)
RFE
(armasm_dom1361289890882.htm)
ROR
(armasm_dom1361289891242.htm)
RRX
(armasm_dom1361289891582.htm)
RSB
(armasm_dom1361289891932.htm)
RSC
(armasm_dom1361289892272.htm)
SADD8
(armasm_dom1361289892652.htm)
SADD16
(armasm_dom1361289893012.htm)
SASX
(armasm_dom1361289893352.htm)
SBC
(armasm_dom1361289893722.htm)
SBFX
(armasm_dom1361289894042.htm)
SDIV
(armasm_dom1361289894402.htm)
SEL
(armasm_dom1361289894732.htm)
SETEND
(armasm_dom1361289895072.htm)
SEV
(armasm_dom1361289895451.htm)
SHADD8
(armasm_dom1361289895801.htm)
SHADD16
(armasm_dom1361289896191.htm)
SHASX
(armasm_dom1361289896561.htm)
SHSAX
(armasm_dom1361289896931.htm)
SHSUB8
(armasm_dom1361289897271.htm)
SHSUB16
(armasm_dom1361289897611.htm)
SMC
(armasm_dom1361289897931.htm)
SMLAxy
(armasm_dom1361289898261.htm)
SMLAD
(armasm_dom1361289898631.htm)
SMLAL
(armasm_dom1361289898991.htm)
SMLALD
(armasm_dom1361289899351.htm)
SMLALxy
(armasm_dom1361289899701.htm)
SMLAWy
(armasm_dom1361289900041.htm)

`MLA`	Multiply Accumulate	x6M
`MLS`	Multiply and Subtract	T2
`MOV`	Move	All
`MOVT`	Move Top	T2
`MOV32` pseudo-instruction	Move 32-bit immediate to register	T2
`MRC`	Move from Coprocessor to Register	x6M
`MRC2`	Move from Coprocessor to Register	5, x6M
`MRRC`	Move from Coprocessor to Registers	5E, x6M
`MRRC2`	Move from Coprocessor to Registers	6, x6M
`MRS`	Move from PSR to register	All
`MRS`	Move from system Coprocessor to Register	7R
`MSR`	Move from register to PSR	All
`MSR`	Move from Register to system Coprocessor	7R
`MUL`	Multiply	All
`MVN`	Move Not	All
`NEG` pseudo-instruction	Negate	All
`NOP`	No Operation	All
`ORN`	Logical OR NOT	T2
`ORR`	Logical OR	All
`PKHBT`, `PKHTB`	Pack Halfwords	6, 7EM
`PLD`	Preload Data	5E, x6M
`PLDW`	Preload Data with intent to Write	7MP
`PLI`	Preload Instruction	7
`POP`	POP registers from stack	All
`PUSH`	PUSH registers to stack	All
`QADD`	Signed saturating Add	5E, 7EM
`QADD8`	Signed saturating parallel byte-wise addition	6, 7EM
`QADD16`	Signed saturating parallel halfword-wise addition	6, 7EM
`QASX`	Signed saturating parallel add and subtract halfwords with exchange	6, 7EM
`QDADD`	Signed saturating Double and Add	5E, 7EM
`QDSUB`	Signed saturating Double and Subtract	5E, 7EM
`QSAX`	Signed saturating	6, 7EM

SMLSD (armasm_dom1361289900371.htm)
SMLSLD (armasm_dom1361289900721.htm)
SMMLA (armasm_dom1361289901101.htm)
SMMLS (armasm_dom1361289901461.htm)
SMMUL (armasm_dom1361289901780.htm)
SMUAD (armasm_dom1361289902130.htm)
SMULxy (armasm_dom1361289902460.htm)
SMULL (armasm_dom1361289902800.htm)
SMULWy (armasm_dom1361289903150.htm)
SMUSD (armasm_dom1361289903540.htm)
SRS (armasm_dom1361289903910.htm)
SSAT (armasm_dom1361289904320.htm)
SSAT16 (armasm_dom1361289904690.htm)
SSAX (armasm_dom1361289905040.htm)
SSUB8 (armasm_dom1361289905380.htm)
SSUB16 (armasm_dom1361289905720.htm)
STC and STC2 (armasm_dom1361289906120.htm)
STM (armasm_dom1361289906470.htm)
STR (immediate offset) (armasm_dom1361289906890.htm)
STR (register offset) (armasm_dom1361289907270.htm)
STR, unprivileged (armasm_dom1361289907680.htm)
STREX (armasm_dom1361289908049.htm)
SUB (armasm_dom1361289908389.htm)
SUBS pc, lr (armasm_dom1361289908769.htm)
SVC (armasm_dom1361289909139.htm)
SWP and SWPB (armasm_dom1361289909499.htm)
SXTAB (armasm_dom1361289909889.htm)
SXTAB16 (armasm_dom1361289910259.htm)
SXTAH (armasm_dom1361289910609.htm)
SXTB (armasm_dom1361289911009.htm)
SXTB16 (armasm_dom1361289911349.htm)
SXTH (armasm_dom1361289911659.htm)
SYS (armasm_dom1361289912019.htm)
TBB and TBH (armasm_dom1361289912399.htm)
TEQ (armasm_dom1361289912729.htm)
TST (armasm_dom1361289913099.htm)
UADD8 (armasm_dom1361289913539.htm)
UADD16 (armasm_dom1361289913919.htm)
UASX (armasm_dom1361289914278.htm)
UBFX (armasm_dom1361289914598.htm)
UDIV (armasm_dom1361289914938.htm)
UHADD8 (armasm_dom1361289915268.htm)
UHADD16 (armasm_dom1361289916088.htm)
UHASX (armasm_dom1361289916448.htm)
UHSAX (armasm_dom1361289916798.htm)
UHSUB8 (armasm_dom1361289917158.htm)
UHSUB16 (armasm_dom1361289917498.htm)

	parallel subtract and add halfwords with exchange	
`QSUB`	Signed saturating Subtract	5E, 7EM
`QSUB8`	Signed saturating parallel byte-wise subtraction	6, 7EM
`QSUB16`	Signed saturating parallel halfword-wise subtraction	6, 7EM
`RBIT`	Reverse Bits	T2
`REV`	Reverse byte order in a word	6
`REV16`	Reverse byte order in two halfwords	6
`REVSH`	Reverse byte order in a halfword and sign extend	6
`RFE`	Return From Exception	T2, x7M
`ROR`	Rotate Right Register	All
`RRX`	Rotate Right with Extend	x6M
`RSB`	Reverse Subtract	All
`RSC`	Reverse Subtract with Carry	x7M
`SADD8`	Signed parallel byte-wise addition	6, 7EM
`SADD16`	Signed parallel halfword-wise addition	6, 7EM
`SASX`	Signed parallel add and subtract halfwords with exchange	6, 7EM
`SBC`	Subtract with Carry	All
`SBFX`	Signed Bit Field eXtract	T2
`SDIV`	Signed divide	7M, 7R
`SEL`	Select bytes according to APSR GE flags	6, 7EM
`SETEND`	Set Endianness for memory accesses	6, x7M
`SEV`	Set Event	K, 6M
`SHADD8`	Signed halving parallel byte-wise addition	6, 7EM
`SHADD16`	Signed halving parallel halfword-wise addition	6, 7EM
`SHASX`	Signed halving parallel add and subtract halfwords with exchange	6, 7EM
`SHSAX`	Signed halving parallel subtract and add halfwords with exchange	6, 7EM
`SHSUB8`	Signed halving parallel byte-wise subtraction	6, 7EM
`SHSUB16`	Signed halving parallel halfword-wise subtraction	6, 7EM
`SMC`	Secure Monitor Call	Z
`SMLAxy`	Signed Multiply with Accumulate (32 <= 16 x 16 + 32)	5E, 7EM

UMAAL (armasm_dom1361289917828.htm)
UMLAL (armasm_dom1361289918178.htm)
UMULL (armasm_dom1361289918548.htm)
UND pseudo-instruction (armasm_dom1361289918908.htm)
UQADD8 (armasm_dom1361289919288.htm)
UQADD16 (armasm_dom1361289919618.htm)
UQASX (armasm_dom1361289919958.htm)
UQSAX (armasm_dom1361289920278.htm)
UQSUB8 (armasm_dom1361289920727.htm)
UQSUB16 (armasm_dom1361289921077.htm)
USAD8 (armasm_dom1361289921417.htm)
USADA8 (armasm_dom1361289921757.htm)
USAT (armasm_dom1361289922077.htm)
USAT16 (armasm_dom1361289922427.htm)
USAX (armasm_dom1361289922797.htm)
USUB8 (armasm_dom1361289923147.htm)
USUB16 (armasm_dom1361289923497.htm)
UXTAB (armasm_dom1361289923867.htm)
UXTAB16 (armasm_dom1361289924207.htm)
UXTAH (armasm_dom1361289924617.htm)
UXTB (armasm_dom1361289924987.htm)
UXTB16 (armasm_dom1361289925377.htm)
UXTH (armasm_dom1361289925707.htm)
WFE (armasm_dom1361289926047.htm)
WFI (armasm_dom1361289926427.htm)
YIELD (armasm_dom1361289926796.htm)
VFP Instructions (armasm_pge1423738743329.htm)
Directives Reference (armasm_dom1361290000455.htm)
Via File Syntax (armasm_chr1359125030640.htm)

`SMLAD`	Dual Signed Multiply Accumulate	6, 7EM
	(32 <= 32 + 16 x 16 + 16 x 16)	
`SMLAL`	Signed Multiply Accumulate (64 <= 64 + 32 x 32)	x6M
`SMLALxy`	Signed Multiply Accumulate (64 <= 64 + 16 x 16)	5E, 7EM
`SMLALD`	Dual Signed Multiply Accumulate Long	6, 7EM
	(64 <= 64 + 16 x 16 + 16 x 16)	
`SMLAWy`	Signed Multiply with Accumulate (32 <= 32 x 16 + 32)	5E, 7EM
`SMLSD`	Dual Signed Multiply Subtract Accumulate	6, 7EM
	(32 <= 32 + 16 x 16 – 16 x 16)	
`SMLSLD`	Dual Signed Multiply Subtract Accumulate Long	6, 7EM
	(64 <= 64 + 16 x 16 – 16 x 16)	
`SMMLA`	Signed top word Multiply with Accumulate (32 <= TopWord(32 x 32 + 32))	6, 7EM
`SMMLS`	Signed top word Multiply with Subtract (32 <= TopWord(32 - 32 x 32))	6, 7EM
`SMMUL`	Signed top word Multiply (32 <= TopWord(32 x 32))	6, 7EM
`SMUAD`, `SMUSD`	Dual Signed Multiply, and Add or Subtract products	6, 7EM
`SMULxy`	Signed Multiply (32 <= 16 x 16)	5E, 7EM
`SMULL`	Signed Multiply (64 <= 32 x 32)	x6M
`SMULWy`	Signed Multiply (32 <= 32 x 16)	5E, 7EM
`SRS`	Store Return State	T2, x7M
`SSAT`	Signed Saturate	6, x6M
`SSAT16`	Signed Saturate, parallel halfwords	6, 7EM
`SSAX`	Signed parallel subtract and add halfwords with exchange	6, 7EM
`SSUB8`	Signed parallel byte-wise subtraction	6, 7EM
`SSUB16`	Signed parallel halfword-wise subtraction	6, 7EM
`STC`	Store Coprocessor	x6M
`STC2`	Store Coprocessor	5, x6M
`STM`	Store Multiple registers	All
`STR`	Store Register with word	All

`STRB`	Store Register with byte	All
`STRBT`	Store Register with byte, user mode	x6M
`STRD`	Store Registers with two words	5E, x6M
`STREX`	Store Register Exclusive	6, x6M
`STREXB` , `STREXH`	Store Register Exclusive Byte, Halfword	K, x6M
`STREXD`	Store Register Exclusive Doubleword	K, x7M
`STRH`	Store Register with halfword	All
`STRHT`	Store Register with halfword, user mode	T2
`STRT`	Store Register with word, user mode	x6M
`SUB`	Subtract	All
`SUBS pc, lr`	Exception return, no stack	T2, x7M
`SVC` (formerly `SWI`)	SuperVisor Call	All
`SWP` , `SWPB`	Swap registers and memory (ARM only)	All, x7M
`SXTAB`	Sign extend Byte, with Addition	6, 7EM
`SXTAB16`	Sign extend two Bytes, with Addition	6, 7EM
`SXTAH`	Sign extend Halfword, with Addition	6, 7EM
`SXTB`	Sign extend Byte	6
`SXTH`	Sign extend Halfword	6
`SXTB16`	Sign extend two Bytes	6, 7EM
`SYS`	Execute system coprocessor instruction	7R
`TBB` , `TBH`	Table Branch Byte, Halfword	T2
`TEQ`	Test Equivalence	x6M
`TST`	Test	All
`UADD8`	Unsigned parallel byte-wise addition	6, 7EM
`UADD16`	Unsigned parallel halfword-wise addition	
`UASX`	Unsigned parallel add and subtract halfwords with exchange	
`UBFX`	Unsigned Bit Field eXtract	T2
`UDIV`	Unsigned divide	7M, 7R
`UHADD8`	Unsigned halving parallel byte-wise addition	6, 7EM
`UHADD16`	Unsigned halving parallel halfword-wise addition	6, 7EM
`UHASX`	Unsigned halving parallel add and subtract halfwords	6, 7EM

	with exchange	
`UHSAX`	Unsigned halving parallel subtract and add halfwords with exchange	6, 7EM
`UHSUB8`	Unsigned halving parallel byte-wise subtraction	6, 7EM
`UHSUB16`	Unsigned halving parallel halfword-wise subtraction	6, 7EM
`UMAAL`	Unsigned Multiply Accumulate Accumulate Long	6, 7EM
	(64 <= 32 + 32 + 32 x 32)	
`UMLAL`	Unsigned Multiply Accumulate	x6M
	(64 <= 32 x 32 + 64), (64 <= 32 x 32)	
`UMULL`	Unsigned Multiply	x6M
	(64 <= 32 x 32 + 64), (64 <= 32 x 32)	
`UQADD8`	Unsigned saturating parallel byte-wise addition	6, 7EM
`UQADD16`	Unsigned saturating parallel halfword-wise addition	6, 7EM
`UQASX`	Unsigned saturating parallel add and subtract halfwords with exchange	6, 7EM
`UQSAX`	Unsigned saturating parallel subtract and add halfwords with exchange	6, 7EM
`UQSUB8`	Unsigned saturating parallel byte-wise subtraction	6, 7EM
`UQSUB16`	Unsigned saturating parallel halfword-wise subtraction	6, 7EM
`USAD8`	Unsigned Sum of Absolute Differences	6, 7EM
`USADA8`	Accumulate Unsigned Sum of Absolute Differences	6, 7EM
`USAT`	Unsigned Saturate	6, x6M
`USAT16`	Unsigned Saturate, parallel halfwords	6, 7EM
`USAX`	Unsigned parallel subtract and add halfwords with exchange	6, 7EM
`USUB8`	Unsigned parallel byte-wise subtraction	6, 7EM
`USUB16`	Unsigned parallel halfword-wise subtraction	6, 7EM
`UXTAB`	Zero extend Byte with Addition	6, 7EM
`UXTAB16`	Zero extend two bytes with Addition	6, 7EM
`UXTAH`	Zero extend Halfword with Addition	6, 7EM
`UXTB`	Zero extend Byte	6

`UXTH`	Zero extend Halfword	6
`UXTB16`	Zero extend two bytes	6, 7EM
`V*`	VFP instructions	
`WFE`	Wait For Event	T2, 6M
`WFI`	Wait For Interrupt	T2, 6M
`YIELD`	Yield	T2, 6M

Entries in the Architecture column indicate that the instructions are available as follows:

All
All versions of the ARM architecture.

5
The ARMv5T*, ARMv6*, and ARMv7 architectures.

5E
The ARMv5TE, ARMv6*, and ARMv7 architectures.

6
The ARMv6* and ARMv7 architectures.

6M
The ARMv6-M and ARMv7 architectures.

x6M
Not available in the ARMv6-M architecture.

7
The ARMv7 architectures.

7M
The ARMv7-M architecture, including ARMv7E-M implementations.

x7M
Not available in the ARMv6-M or ARMv7-M architecture, or any ARMv7E-M implementation.

7EM
ARMv7E-M implementations but not in the ARMv7-M or ARMv6-M architecture.

7R
The ARMv7-R architecture.

7VE
The ARMv7 architectures that implement the Virtualization Extensions.

J
The ARMv5TEJ, ARMv6*, and ARMv7 architectures.

K
The ARMv6K, and ARMv7 architectures.

T
The ARMv4T, ARMv5T*, ARMv6*, and ARMv7 architectures.

T2
The ARMv6T2 and above architectures.

Z
If Security Extensions are implemented.

Products (/product/)

Development Tools
Arm (/Arm/)
C166 (/c166/)
C51 (/c51/)
C251 (/c251/)
µVision IDE and Debugger (/uvision/)

Hardware & Collateral
ULINK Debug Adaptors (/ulink/)
Evaluation Boards (/boards2/)
Product Brochures (/product/brochures.asp)
Device Database (/dd2/)
Distributors (/distis/)

Downloads (/download/)
MDK-Arm (/demo/eval/arm.htm)
C51 (/demo/eval/c51.htm)
C166 (/demo/eval/c166.htm)
C251 (/demo/eval/c251.htm)
File downloads (/download/file/)

Support (/support/)
Knowledgebase (/support/knowledgebase.asp)
Discussion Forum (/forum/)
Product Manuals (/support/man/)
Application Notes (/appnotes/)

Contact
Distributors (/d
Request a Quo
Sales Contacts

Cookie Settings (/company/cookiesettings) | Terms of Use (/company/terms) | Privacy (/company/privacy) | Accessibility (/company/accessibility) | Trademarks (https://www.arm.com/company/policies/trademarks) | Contact Us (/company/contact/) | Feedback (/support/feedback.asp)