



# Cortex™-M4 Technical Reference Manual

## ARM DDI 0439B

### Errata 01

This Errata document gives corrections and additions to the Cortex-M4 Technical Reference Manual (ARM DDI 0439B).

The number of cycles for the MUL and MLA instructions in Table 3-1 page 3-4 are incorrect. Each instruction takes one cycle to execute, not two cycles.

Table 3-1 Cortex-M4 instruction set summary lists all the correct cycle timings.

**Table 3-1 Cortex-M4 instruction set summary**

Operation	Description	Assembler	Cycles
Move	Register	MOV Rd, <op2>	1
	16-bit immediate	MOVW Rd, #<imm>	1
	Immediate into top	MOVT Rd, #<imm>	1
	To PC	MOV PC, Rm	1 + P
Add	Add	ADD Rd, Rn, <op2>	1
	Add to PC	ADD PC, PC, Rm	1 + P
	Add with carry	ADC Rd, Rn, <op2>	1
	Form address	ADR Rd, <label>	1
Subtract	Subtract	SUB Rd, Rn, <op2>	1
	Subtract with borrow	SBC Rd, Rn, <op2>	1
	Reverse	RSB Rd, Rn, <op2>	1
Multiply	Multiply	MUL Rd, Rn, Rm	1
	Multiply accumulate	MLA Rd, Rn, Rm	1
	Multiply subtract	MLS Rd, Rn, Rm	1
	Long signed	SMULL RdLo, RdHi, Rn, Rm	1
	Long unsigned	UMULL RdLo, RdHi, Rn, Rm	1
	Long signed accumulate	SMLAL RdLo, RdHi, Rn, Rm	1
	Long unsigned accumulate	UMLAL RdLo, RdHi, Rn, Rm	1

**Table 3-1 Cortex-M4 instruction set summary (continued)**

Operation	Description	Assembler	Cycles
Divide	Signed	SDIV Rd, Rn, Rm	2 to 12 <sup>a</sup>
	Unsigned	UDIV Rd, Rn, Rm	2 to 12 <sup>a</sup>
Saturate	Signed	SSAT Rd, #<imm>, <op2>	1
	Unsigned	USAT Rd, #<imm>, <op2>	1
Compare	Compare	CMP Rn, <op2>	1
	Negative	CMN Rn, <op2>	1
Logical	AND	AND Rd, Rn, <op2>	1
	Exclusive OR	EOR Rd, Rn, <op2>	1
	OR	ORR Rd, Rn, <op2>	1
	OR NOT	ORN Rd, Rn, <op2>	1
	Bit clear	BIC Rd, Rn, <op2>	1
	Move NOT	MVN Rd, <op2>	1
	AND test	TST Rn, <op2>	1
	Exclusive OR test	TEQ Rn, <op1>	
Shift	Logical shift left	LSL Rd, Rn, #<imm>	1
	Logical shift left	LSL Rd, Rn, Rs	1
	Logical shift right	LSR Rd, Rn, #<imm>	1
	Logical shift right	LSR Rd, Rn, Rs	1
	Arithmetic shift right	ASR Rd, Rn, #<imm>	1
	Arithmetic shift right	ASR Rd, Rn, Rs	1
Rotate	Rotate right	ROR Rd, Rn, #<imm>	1
	Rotate right	ROR Rd, Rn, Rs	1
	With extension	RRX Rd, Rn	1
Count	Leading zeroes	CLZ Rd, Rn	1

**Table 3-1 Cortex-M4 instruction set summary (continued)**

Operation	Description	Assembler	Cycles
Load	Word	LDR Rd, [Rn, <op2>]	2 <sup>b</sup>
	To PC	LDR PC, [Rn, <op2>]	2 <sup>b</sup> + P
	Halfword	LDRH Rd, [Rn, <op2>]	2 <sup>b</sup>
	Byte	LDRB Rd, [Rn, <op2>]	2 <sup>b</sup>
	Signed halfword	LDRSH Rd, [Rn, <op2>]	2 <sup>b</sup>
	Signed byte	LDRSB Rd, [Rn, <op2>]	2 <sup>b</sup>
	User word	LDRT Rd, [Rn, #<imm>]	2 <sup>b</sup>
	User halfword	LDRHT Rd, [Rn, #<imm>]	2 <sup>b</sup>
	User byte	LDRBT Rd, [Rn, #<imm>]	2 <sup>b</sup>
	User signed halfword	LDRSHT Rd, [Rn, #<imm>]	2 <sup>b</sup>
	User signed byte	LDRSBT Rd, [Rn, #<imm>]	2 <sup>b</sup>
	PC relative	LDR Rd, [PC, #<imm>]	2 <sup>b</sup>
	Doubleword	LDRD Rd, Rd, [Rn, #<imm>]	1 + N
	Multiple	LDM Rn, {<reglist>}	1 + N
	Multiple including PC	LDM Rn, {<reglist>, PC}	1 + N + P
Store	Word	STR Rd, [Rn, <op2>]	2 <sup>b</sup>
	Halfword	STRH Rd, [Rn, <op2>]	2 <sup>b</sup>
	Byte	STRB Rd, [Rn, <op2>]	2 <sup>b</sup>
	Signed halfword	STRSH Rd, [Rn, <op2>]	2 <sup>b</sup>
	Signed byte	STRSB Rd, [Rn, <op2>]	2 <sup>b</sup>
	User word	STRT Rd, [Rn, #<imm>]	2 <sup>b</sup>
	User halfword	STRHT Rd, [Rn, #<imm>]	2 <sup>b</sup>
	User byte	STRBT Rd, [Rn, #<imm>]	2 <sup>b</sup>
	User signed halfword	STRSHT Rd, [Rn, #<imm>]	2 <sup>b</sup>
	User signed byte	STRSBT Rd, [Rn, #<imm>]	2 <sup>b</sup>
	Doubleword	STRD Rd, Rd, [Rn, #<imm>]	1 + N
	Multiple	STM Rn, {<reglist>}	1 + N
Push	Push	PUSH {<reglist>}	1 + N
	Push with link register	PUSH {<reglist>, LR}	1 + N
Pop	Pop	POP {<reglist>}	1 + N
	Pop and return	POP {<reglist>, PC}	1 + N + P

**Table 3-1 Cortex-M4 instruction set summary (continued)**

Operation	Description	Assembler	Cycles
Semaphore	Load exclusive	LDREX Rd, [Rn, #<imm>]	2
	Load exclusive half	LDREXH Rd, [Rn]	2
	Load exclusive byte	LDREXB Rd, [Rn]	2
	Store exclusive	STREX Rd, Rt, [Rn, #<imm>]	2
	Store exclusive half	STREXH Rd, Rt, [Rn]	2
	Store exclusive byte	STREXB Rd, Rt, [Rn]	2
	Clear exclusive monitor	CLREX	1
Branch	Conditional	B<cc> <label>	1 or 1 + P <sup>c</sup>
	Unconditional	B <label>	1 + P
	With link	BL <label>	1 + P
	With exchange	BX Rm	1 + P
	With link and exchange	BLX Rm	1 + P
	Branch if zero	CBZ Rn, <label>	1 or 1 + P <sup>c</sup>
	Branch if non-zero	CBNZ Rn, <label>	1 or 1 + P <sup>c</sup>
	Byte table branch	TBB [Rn, Rm]	2 + P
	Halfword table branch	TBH [Rn, Rm, LSL#1]	2 + P
State change	Supervisor call	SVC #<imm>	-
	If-then-else	IT... <cond>	1 <sup>d</sup>
	Disable interrupts	CPSID <flags>	1 or 2
	Enable interrupts	CPSIE <flags>	1 or 2
	Read special register	MRS Rd, <specreg>	1 or 2
	Write special register	MSR <specreg>, Rn	1 or 2
	Breakpoint	BKPT #<imm>	-
Extend	Signed halfword to word	SXTH Rd, <op2>	1
	Signed byte to word	SXTB Rd, <op2>	1
	Unsigned halfword	UXTH Rd, <op2>	1
	Unsigned byte	UXTB Rd, <op2>	1
Bit field	Extract unsigned	UBFX Rd, Rn, #<imm>, #<imm>	1
	Extract signed	SBFX Rd, Rn, #<imm>, #<imm>	1
	Clear	BFC Rd, Rn, #<imm>, #<imm>	1
	Insert	BFI Rd, Rn, #<imm>, #<imm>	1

**Table 3-1 Cortex-M4 instruction set summary (continued)**

Operation	Description	Assembler	Cycles
Reverse	Bytes in word	REV Rd, Rm	1
	Bytes in both halfwords	REV16 Rd, Rm	1
	Signed bottom halfword	REVSH Rd, Rm	1
	Bits in word	RBIT Rd, Rm	1
Hint	Send event	SEV	1
	Wait for event	WFE	1 + W
	Wait for interrupt	WFI	1 + W
	No operation	NOP	1
Barriers	Instruction synchronization	ISB	1 + B
	Data memory	DMB	1 + B
	Data synchronization	DSB <flags>	1 + B

- Division operations use early termination to minimize the number of cycles required based on the number of leading ones and zeroes in the input operands.
- Neighboring load and store single instructions can pipeline their address and data phases. This enables these instructions to complete in a single execution cycle.
- Conditional branch completes in a single cycle if the branch is not taken.
- An IT instruction can be folded onto a preceding 16-bit Thumb instruction, enabling execution in zero cycles.