

SEGGER Assembler

An assembler for
Arm microcontrollers

User Guide & Reference Manual

Document: UM20006
Software Version: 2.10 Beta
Revision: 0
Date: August 11, 2020



A product of SEGGER Microcontroller GmbH

www.segger.com

Disclaimer

Specifications written in this document are believed to be accurate, but are not guaranteed to be entirely free of error. The information in this manual is subject to change for functional or performance improvements without notice. Please make sure your manual is the latest edition. While the information herein is assumed to be accurate, SEGGER Microcontroller GmbH (SEGGER) assumes no responsibility for any errors or omissions. SEGGER makes and you receive no warranties or conditions, express, implied, statutory or in any communication with you. SEGGER specifically disclaims any implied warranty of merchantability or fitness for a particular purpose.

Copyright notice

You may not extract portions of this manual or modify the PDF file in any way without the prior written permission of SEGGER. The software described in this document is furnished under a license and may only be used or copied in accordance with the terms of such a license.

© 2018-2020 SEGGER Microcontroller GmbH, Monheim am Rhein / Germany

Trademarks

Names mentioned in this manual may be trademarks of their respective companies.

Brand and product names are trademarks or registered trademarks of their respective holders.

Contact address

SEGGER Microcontroller GmbH

Ecolab-Allee 5
D-40789 Monheim am Rhein

Germany

Tel. +49 2173-99312-0
Fax. +49 2173-99312-28
E-mail: support@segger.com*
Internet: www.segger.com

*By sending us an email your (personal) data will automatically be processed. For further information please refer to our privacy policy which is available at <https://www.segger.com/legal/privacy-policy/>.

Manual versions

This manual describes the current software version. If you find an error in the manual or a problem in the software, please report it to us and we will try to assist you as soon as possible.

Contact us for further information on topics or functions that are not yet documented.

Print date: August 11, 2020

Software	Revision	Date	By	Description
2.10	0	180514	PC	Initial release.
1.10	0	180427	PC	T32 + A32 internal release.
1.00	0	180410	PC	T32 internal release.

About this document

Assumptions

This document assumes that you already have a solid knowledge of the following:

- The software tools used for building your application (assembler, linker, C compiler).
- The C programming language.
- The target processor.
- DOS command line.

If you feel that your knowledge of C is not sufficient, we recommend *The C Programming Language* by Kernighan and Richie (ISBN 0--13--1103628), which describes the standard in C programming and, in newer editions, also covers the ANSI C standard.

How to use this manual

This manual describes all the functions and macros that the product offers. It assumes you have a working knowledge of the C language. Knowledge of assembly programming is not required.

Typographic conventions for syntax

This manual uses the following typographic conventions:

Style	Used for
Body	Body text.
Parameter	Parameters in API functions.
Sample	Sample code in program examples.
Sample comment	Comments in program examples.
User Input	Text entered at the keyboard by a user in a session transcript.
Secret Input	Text entered at the keyboard by a user, but not echoed (e.g. password entry), in a session transcript.
Reference	Reference to chapters, sections, tables and figures or other documents.
Emphasis	Very important sections.
SEGGER home page	A hyperlink to an external document or web site.

Table of contents

1	About the assembler	16
1.1	Introduction	17
1.1.1	What is the SEGGER Assembler?	17
1.1.2	Assembler features	17
2	Command-line options	18
2.1	Input file options	19
2.1.1	--include	20
2.1.2	--via	21
2.2	Output file options	22
2.2.1	--dependency-file	23
2.2.2	--keep-empty-sections	23
2.2.3	--no-keep-empty-sections	24
2.2.4	--keep-local-symbols	25
2.2.5	--no-keep-local-symbols	26
2.2.6	--mapping-symbols	27
2.2.7	--no-mapping-symbols	28
2.2.8	--output	29
2.2.9	--symbols	30
2.2.10	--no-symbols	31
2.2.11	--pad-sections	32
2.2.12	--no-pad-sections	33
2.3	Target selection options	34
2.3.1	--cpu=name	34
2.3.2	-marm	35
2.3.3	-mbig-endian	36
2.3.4	-mlittle-endian	37
2.3.5	-mthumb	38
2.4	Control options	39
2.4.1	--auto-import	39
2.4.2	--no-auto-import	40
2.4.3	--define	41
2.4.4	--input-syntax	42
2.5	List file options	43
2.5.1	--list-diagnostics	43
2.5.2	--no-list-diagnostics	44
2.5.3	--list-file	45
2.5.4	--list-form-feed	46
2.5.5	--no-list-form-feed	47
2.5.6	--list-html	48

2.5.7	--list-page-length	49
2.5.8	--list-symbols	50
2.5.9	--no-list-symbols	51
2.5.10	--list-text	52
2.6	Diagnostic options	53
2.6.1	--remarks	53
2.6.2	--remarks-are-errors	54
2.6.3	--remarks-are-warnings	55
2.6.4	--no-remarks	56
2.6.5	--silent	57
2.6.6	--verbose	58
2.6.7	--warnings	59
2.6.8	--warnings-are-errors	60
2.6.9	--no-warnings	60
3	Expressions	61
4	GNU syntax	62
4.1	Source file directives	63
4.1.1	.INCBIN	63
4.2	Mode directives	64
4.2.1	.ARM	64
4.2.2	.CODE	65
4.2.3	.THUMB	66
4.3	Section directives	67
4.3.1	.DATA	67
4.3.2	.BSS	68
4.3.3	.TEXT	69
4.3.4	.RODATA	70
4.4	Symbol directives	71
4.4.1	.GLOBL, .GLOBAL, .EXTERN	71
4.4.2	.LOCAL	72
4.4.3	.WEAK	73
4.4.4	.HIDDEN	74
4.4.5	.PROTECTED	75
4.4.6	.INTERNAL	76
4.5	Data allocation directives	77
4.5.1	.BYTE, .ASCII	77
4.5.2	.STRING, .ASCIZ	78
4.5.3	.2BYTE	79
4.5.4	.4BYTE	80
4.5.5	.HWORD, .SHORT	81
4.5.6	.WORD, .INT, .LONG	82
4.5.7	.INST	83
4.5.8	.INST.N	84
4.5.9	.INST.W	85
4.6	Section directives	86
4.6.1	.ALIGN	86
4.6.2	.P2ALIGN, .P2ALIGNW, .P2ALIGNL	87
4.7	Listing directives	88
4.7.1	.LIST	88
4.7.2	.NOLIST	89
4.7.3	.TITLE	90
4.7.4	.SUBTTL	91
5	Arm syntax	92
5.1	Mode directives	93
5.1.1	CODE16	93
5.1.2	CODE32	94

5.2	Listing directives	95
5.2.1	TTL	96
5.2.2	SUBT	97
6	IAR syntax	98
6.1	Mode directives	99
6.1.1	ARM, CODE32	99
6.1.2	CODE	100
6.1.3	THUMB, CODE16	101
6.2	Data allocation directives	102
6.2.1	DC8, DCB	102
6.2.2	DC16, DCW	103
6.2.3	DC24	104
6.2.4	DC32, DCD	105
6.2.5	DCI16, DCIW	106
6.2.6	DCI32, DCID	107
6.2.7	DS8, DS	108
6.2.8	DS16, DSW	109
6.2.9	DS24	110
6.2.10	DS32, DSD	111
7	Instructions	112
7.1	Alphabetical list of instructions	113
7.1.1	ADC	114
7.1.2	ADD	115
7.1.3	ADDW	117
7.1.4	AND	118
7.1.5	ASR	119
7.1.6	B	120
7.1.7	BCC	121
7.1.8	BCS	122
7.1.9	BEQ	123
7.1.10	BFC	124
7.1.11	BFI	125
7.1.12	BGE	126
7.1.13	BGT	127
7.1.14	BHI	128
7.1.15	BHS	129
7.1.16	BIC	130
7.1.17	BKPT	131
7.1.18	BL	132
7.1.19	BLE	133
7.1.20	BLO	134
7.1.21	BLS	135
7.1.22	BLT	136
7.1.23	BLX	137
7.1.24	BMI	138
7.1.25	BNE	139
7.1.26	BPL	140
7.1.27	BVC	141
7.1.28	BVS	142
7.1.29	BX	143
7.1.30	BXJ	144
7.1.31	CBNZ	145
7.1.32	CBZ	146
7.1.33	CDP	147
7.1.34	CDP2	148
7.1.35	CLREX	149
7.1.36	CLZ	150

7.1.37	CMN	151
7.1.38	CMP	152
7.1.39	CPS	153
7.1.40	CPSID	154
7.1.41	CPSIE	155
7.1.42	CPY	156
7.1.43	CRC32B	157
7.1.44	CRC32CB	158
7.1.45	CRC32CH	159
7.1.46	CRC32CW	160
7.1.47	CRC32H	161
7.1.48	CRC32W	162
7.1.49	DBG	163
7.1.50	DCPS1	164
7.1.51	DCPS2	165
7.1.52	DCPS3	166
7.1.53	DMB	167
7.1.54	DSB	168
7.1.55	EOR	169
7.1.56	ERET	170
7.1.57	ESB	171
7.1.58	FABSD	172
7.1.59	FABSS	173
7.1.60	FADDD	174
7.1.61	FADDS	175
7.1.62	FCMPD	176
7.1.63	FCMPED	177
7.1.64	FCMPES	178
7.1.65	FCMPS	179
7.1.66	FCVTDS	180
7.1.67	FCVTSD	181
7.1.68	FDIVD	182
7.1.69	FDIVS	183
7.1.70	FMDRR	184
7.1.71	FMRRD	185
7.1.72	FMRS	186
7.1.73	FMSR	187
7.1.74	FMSTAT	188
7.1.75	FMULD	189
7.1.76	FMULS	190
7.1.77	FNEGD	191
7.1.78	FNEGS	192
7.1.79	FSITOD	193
7.1.80	FSITOS	194
7.1.81	FSQRTD	195
7.1.82	FSQRTS	196
7.1.83	FSUBD	197
7.1.84	FSUBS	198
7.1.85	FTOSIZD	199
7.1.86	FTOSIZS	200
7.1.87	FTOUIZD	201
7.1.88	FTOUIZS	202
7.1.89	FUITOD	203
7.1.90	FUITOS	204
7.1.91	HLT	205
7.1.92	HVC	206
7.1.93	ISB	207
7.1.94	IT	208
7.1.95	ITE	209
7.1.96	ITEE	210

7.1.97	ITEEE	211
7.1.98	ITEET	212
7.1.99	ITET	213
7.1.100	ITETE	214
7.1.101	ITETT	215
7.1.102	ITT	216
7.1.103	ITTE	217
7.1.104	ITTEE	218
7.1.105	ITTET	219
7.1.106	ITTT	220
7.1.107	ITTTE	221
7.1.108	ITTTT	222
7.1.109	LDA	223
7.1.110	LDAB	224
7.1.111	LDAEX	225
7.1.112	LDAEXB	226
7.1.113	LDAEXD	227
7.1.114	LDAEXH	228
7.1.115	LDAH	229
7.1.116	LDM	230
7.1.117	LDMDA	231
7.1.118	LDMDB	232
7.1.119	LDMEA	233
7.1.120	LDMED	234
7.1.121	LDMFA	235
7.1.122	LDMFD	236
7.1.123	LDMIA	237
7.1.124	LDMIB	238
7.1.125	LDR	239
7.1.126	LDRB	240
7.1.127	LDRBT	241
7.1.128	LDRD	242
7.1.129	LDREX	243
7.1.130	LDREXB	244
7.1.131	LDREXD	245
7.1.132	LDREXH	246
7.1.133	LDRH	247
7.1.134	LDRHT	248
7.1.135	LDRSB	249
7.1.136	LDRSBT	250
7.1.137	LDRSH	251
7.1.138	LDRSHT	252
7.1.139	LDRT	253
7.1.140	LSL	254
7.1.141	LSR	255
7.1.142	MCR	256
7.1.143	MCR2	257
7.1.144	MCRR	258
7.1.145	MCRR2	259
7.1.146	MLA	260
7.1.147	MLS	261
7.1.148	MOV	262
7.1.149	MOV S	263
7.1.150	MOV T	264
7.1.151	MOV W	265
7.1.152	MRC	266
7.1.153	MRC2	267
7.1.154	MRRC	268
7.1.155	MRRC2	269
7.1.156	MRS	270

7.1.157	MSR	271
7.1.158	MUL	272
7.1.159	MVN	273
7.1.160	NEG	274
7.1.161	NOP	275
7.1.162	ORN	276
7.1.163	ORR	277
7.1.164	PKHBT	278
7.1.165	PKHTB	279
7.1.166	PLD	280
7.1.167	PLDW	281
7.1.168	PLI	282
7.1.169	POP	283
7.1.170	PUSH	284
7.1.171	QADD	285
7.1.172	QADD16	286
7.1.173	QADD8	287
7.1.174	QASX	288
7.1.175	QDADD	289
7.1.176	QDSUB	290
7.1.177	QSAX	291
7.1.178	QSUB	292
7.1.179	QSUB16	293
7.1.180	QSUB8	294
7.1.181	RBIT	295
7.1.182	REV	296
7.1.183	REV16	297
7.1.184	REVSH	298
7.1.185	RFE	299
7.1.186	RFEDA	300
7.1.187	RFEDB	301
7.1.188	RFEIA	302
7.1.189	RFEIB	303
7.1.190	ROR	304
7.1.191	RRX	305
7.1.192	RSB	306
7.1.193	RSC	307
7.1.194	SADD16	308
7.1.195	SADD8	309
7.1.196	SASX	310
7.1.197	SBC	311
7.1.198	SBFX	312
7.1.199	SDIV	313
7.1.200	SEL	314
7.1.201	SETEND	315
7.1.202	SETPAN	316
7.1.203	SEV	317
7.1.204	SEVL	318
7.1.205	SHADD16	319
7.1.206	SHADD8	320
7.1.207	SHASX	321
7.1.208	SHSAX	322
7.1.209	SHSUB16	323
7.1.210	SHSUB8	324
7.1.211	SMLABB	325
7.1.212	SMLABT	326
7.1.213	SMLAD	327
7.1.214	SMLADX	328
7.1.215	SMLAL	329
7.1.216	SMLALBB	330

7.1.217	SMLALBT	331
7.1.218	SMLALD	332
7.1.219	SMLALDX	333
7.1.220	SMLALTB	334
7.1.221	SMLALTT	335
7.1.222	SMLATB	336
7.1.223	SMLATT	337
7.1.224	SMLAWB	338
7.1.225	SMLAWT	339
7.1.226	SMLSD	340
7.1.227	SMLS DX	341
7.1.228	SMLS LD	342
7.1.229	SMLS DX	343
7.1.230	SMMLA	344
7.1.231	SMMLAR	345
7.1.232	SMMLS	346
7.1.233	SMMLSR	347
7.1.234	SMMUL	348
7.1.235	SMMULR	349
7.1.236	SMUAD	350
7.1.237	SMUADX	351
7.1.238	SMULBB	352
7.1.239	SMULBT	353
7.1.240	SMULL	354
7.1.241	SMULTB	355
7.1.242	SMULTT	356
7.1.243	SMULWB	357
7.1.244	SMULWT	358
7.1.245	SMUSD	359
7.1.246	SMUSDX	360
7.1.247	SRS	361
7.1.248	SRSDA	362
7.1.249	SRSDB	363
7.1.250	SRSIA	364
7.1.251	SRSIB	365
7.1.252	SSAT	366
7.1.253	SSAT16	367
7.1.254	SSAX	368
7.1.255	SSUB16	369
7.1.256	SSUB8	370
7.1.257	STL	371
7.1.258	STLB	372
7.1.259	STLH	373
7.1.260	STM	374
7.1.261	STMDA	375
7.1.262	STMDB	376
7.1.263	STMEA	377
7.1.264	STMED	378
7.1.265	STMFA	379
7.1.266	STMFD	380
7.1.267	STMIA	381
7.1.268	STMIB	382
7.1.269	STR	383
7.1.270	STRB	384
7.1.271	STRBT	385
7.1.272	STRD	386
7.1.273	STREX	387
7.1.274	STREXB	388
7.1.275	STREXD	389
7.1.276	STREXH	390

7.1.277	STRH	391
7.1.278	STRHT	392
7.1.279	STRT	393
7.1.280	SUB	394
7.1.281	SUBW	395
7.1.282	SVC	396
7.1.283	SXTAB	397
7.1.284	SXTAB16	398
7.1.285	SXTAH	399
7.1.286	SXTB	400
7.1.287	SXTB16	401
7.1.288	SXTH	402
7.1.289	TBB	403
7.1.290	TBH	404
7.1.291	TEQ	405
7.1.292	TST	406
7.1.293	TT	407
7.1.294	TTA	408
7.1.295	TTAT	409
7.1.296	TTT	410
7.1.297	UADD16	411
7.1.298	UADD8	412
7.1.299	UASX	413
7.1.300	UBFX	414
7.1.301	UDF	415
7.1.302	UDIV	416
7.1.303	UEXT16	417
7.1.304	UEXT8	418
7.1.305	UHADD16	419
7.1.306	UHADD8	420
7.1.307	UHASX	421
7.1.308	UHSAX	422
7.1.309	UHSUB16	423
7.1.310	UHSUB8	424
7.1.311	UMAAL	425
7.1.312	UMLAL	426
7.1.313	UMULL	427
7.1.314	UQADD16	428
7.1.315	UQADD8	429
7.1.316	UQASX	430
7.1.317	UQSAX	431
7.1.318	UQSUB16	432
7.1.319	UQSUB8	433
7.1.320	USAD8	434
7.1.321	USADA8	435
7.1.322	USAT	436
7.1.323	USAT16	437
7.1.324	USAX	438
7.1.325	USUB16	439
7.1.326	USUB8	440
7.1.327	UXTAB	441
7.1.328	UXTAB16	442
7.1.329	UXTAH	443
7.1.330	UXTB	444
7.1.331	UXTB16	445
7.1.332	UXTH	446
7.1.333	VABS	447
7.1.334	VADD	448
7.1.335	VCMP	449
7.1.336	VCMPE	450

7.1.337	VCVT	451
7.1.338	VCVTA	452
7.1.339	VCVTB	453
7.1.340	VCVTM	454
7.1.341	VCVTN	455
7.1.342	VCVTP	456
7.1.343	VCVTR	457
7.1.344	VCVTT	458
7.1.345	VDIV	459
7.1.346	VFMA	460
7.1.347	VFMS	461
7.1.348	VFNMA	462
7.1.349	VFNMS	463
7.1.350	VLDR	464
7.1.351	VMAXNM	465
7.1.352	VMINNM	466
7.1.353	VMLA	467
7.1.354	VMLS	468
7.1.355	VMOV	469
7.1.356	VMUL	470
7.1.357	VNEG	471
7.1.358	VNMLA	472
7.1.359	VNMLS	473
7.1.360	VNMUL	474
7.1.361	VRINTA	475
7.1.362	VRINTM	476
7.1.363	VRINTN	477
7.1.364	VRINTP	478
7.1.365	VRINTR	479
7.1.366	VRINTX	480
7.1.367	VRINTZ	481
7.1.368	VSELEQ	482
7.1.369	VSELGE	483
7.1.370	VSELGT	484
7.1.371	VSQRT	485
7.1.372	VSTR	486
7.1.373	VSUB	487
7.1.374	WFE	488
7.1.375	WFI	489
7.1.376	YIELD	490
7.2	Instructions by architecture	491
8	Differences between assemblers	501
8.1	Syntax and Encoding differences	502
8.1.1	MUL and MULS	502
8.1.2	MULS	502
9	Indexes	503
9.1	Subject index	504

Chapter 1

About the assembler

1.1 Introduction

This section presents an overview of the SEGGER Assembler and its capabilities.

1.1.1 What is the SEGGER Assembler?

The SEGGER Assembler is a fast assembler that processes Cortex-M assembly language source files. It is designed to be very flexible, yet simple to use.

The assembler accepts GNU, Arm, and IAR syntax source files and will assemble them to Arm ELF object files that can be linked by a standard Arm linker.

1.1.2 Assembler features

The SEGGER Assembler has the following features:

- Highly efficient and very fast to assemble.
- Supports GNU, Arm, and IAR syntax.

Chapter 2

Command-line options

Command line option naming is generally compatible with the following toolsets:

- GNU assembler `as`
- Arm assembler `armasm`
- IAR assembler `iasmarm`

2.1 Input file options

2.1.1 --include

Summary

Add include directory.

Syntax

`--include path`
`-Ipath`

Description

This option adds the path *path* to the end of the list of directories to search for included source files and included binary files.

2.1.2 --via

Synopsis

Read additional options and input files from file.

Syntax

```
--via filename  
--via=filename  
-f filename  
@filename
```

Description

This option reads the file *filename* for additional options and input files. Options are separated by spaces or newlines, and file names which contain special characters, such as spaces, must be enclosed in double quotation marks.

Notes

This option can only be provided on the command line and cannot appear in an indirect file.

2.2 Output file options

2.2.1 --dependency-file

Summary

Write dependencies to file.

Syntax

`--dependency-file filename`
`--MD filename`

Description

This option instructs the assembler to write the list of dependencies (included source files and included binary files) to the file *filename*.

2.2.2 --keep-empty-sections

Summary

Keep empty sections in object file.

Syntax

`--keep-empty-sections`

Description

This option instructs the assembler to emit empty sections that have been created during assembly. An empty section is a section that is created using a section directive but has no instructions or data allocated to it.

The default is to eliminate empty sections.

See also

`--no-keep-empty-sections` on page 24

2.2.3 --no-keep-empty-sections

Summary

Eliminate empty sections from object file.

Syntax

--no-keep-empty-sections

Description

This option instructs the assembler to eliminate empty sections that have been created during assembly. An empty section is a section that is created using a section directive but has no instructions or data allocated to it.

The default is to eliminate empty sections.

See also

--keep-empty-sections on page 23

2.2.4 --keep-local-symbols

Summary

Keep local symbol definitions in object file.

Syntax

--keep-local-symbols

Description

This option instructs the assembler to include local symbol definitions created during assembly in the object file's symbol table. Local symbols can be either an equate or a local label definition that is not exported as a global symbol.

The default is eliminate local symbols.

Note

Local symbols that start `".L"` are never emitted to the object even with **--keep-local-symbols** in effect.

See also

--no-keep-local-symbols on page 26

2.2.5 --no-keep-local-symbols

Summary

Do not keep local symbol definitions in object file.

Syntax

`--no-keep-local-symbols`

Description

This option instructs the assembler to exclude local symbol definitions created during assembly from the object file's symbol table. Local symbols can be either an equate or a local label definition that is not exported as a global symbol.

See also

--keep-local-symbols on page 25

2.2.6 --mapping-symbols

Summary

Include mapping symbols.

Syntax

`--mapping-symbols`

Description

This option includes mapping symbols in the Arm ELF file.

See also

--no-mapping-symbols on page 28

2.2.7 --no-mapping-symbols

Summary

Do not generate mapping symbols.

Syntax

`--no-mapping-symbols`

Description

This option produces an Arm ELF file that contains no code and data mapping symbols.

See also

--mapping-symbols on page 27

2.2.8 --output

Summary

Set output file name.

Syntax

`--output filename`

`-o filename`

`--output=filename`

`-o=filename`

Description

This option sets the Arm ELF output filename, typically with extension “o”.

2.2.9 --symbols

Summary

Include symbol table.

Syntax

`--symbols`

Description

This option includes the symbol table in the Arm ELF file. This does not automatically enable mapping symbol inclusion.

See also

`--no-symbols` on page 31

2.2.10 --no-symbols

Summary

Discard symbol table.

Syntax

`--no-symbols`

Description

This option eliminates all symbols from the Arm ELF file that are not required which includes the mapping symbols.

See also

`--symbols` on page 30

2.2.11 --pad-sections

Summary

Pad sections to alignment boundary.

Syntax

--pad-sections

Description

This option instructs the assembler to emit additional data to a section such that its size is a multiple of its alignment. Padding a section to a multiple of its alignment may waste space that a linker could otherwise utilize for section packing.

The default is not to pad sections.

See also

--no-pad-sections on page 33

2.2.12 --no-pad-sections

Summary

Do not pad sections to alignment boundary.

Syntax

--no-pad-sections

Description

This option instructs the assembler not to emit additional data to a section such that its size is a multiple of its alignment. Padding a section to a multiple of its alignment may waste space that a linker could otherwise utilize for section packing.

The default is not to pad sections.

See also

--pad-sections on page 32

2.3 Target selection options

2.3.1 --cpu=name

Summary

Set target core or architecture.

Syntax

--cpu=*name*
-cpu=*name*
-mcpu=*name*

Description

This option selects the target processor for the application and controls the construction of appropriate veneers when required.

The core names accepted are:

- cortex-m0, cortex-m0plus, cortex-m0+, cortex-m1, cortex-m3, cortex-m4, cortex-m7, cortex-m23, cortex-m33
- cortex-a5, cortex-a7, cortex-a8, cortex-a9, cortex-a12, cortex-a15, cortex-a17, cortex-a32, cortex-a35, cortex-a53, cortex-a55, cortex-a57, cortex-a72, cortex-a73, cortex-a75
- cortex-r4, cortex-r4f, cortex-r5, cortex-r7, cortex-r8, cortex-r52

The architecture names accepted are:

- 4t, 5t, 5te
- 6, 6j, 6k, 6kz, 6t2, 6z, 6zk, 6-m, 6s-m
- 7, 7ve, 7-a, 7-r, 7-m, 7e-m
- 8-a, 8.1-a, 8.2-a, 8.3-a, 8.4-a, 8-r, 8-m.base, 8-m.main

The default is --cpu=cortex-m0.

2.3.2 -marm

Summary

Select Arm instruction set.

Syntax

`-marm`

Description

This option sets the default assembler instruction set to Arm (A32). The default instruction set is the instruction set if the source file does not select a specific instruction set using assembler source directives.

See also

`-mthumb` on page 38

2.3.3 -mbig-endian

Summary

Select big-endian byte order.

Syntax

`-mbig-endian`
`-EB`

Description

This option selects big-endian byte ordering for data and instructions.

The default is little-endian byte ordering.

See also

-mlittle-endian on page 37

2.3.4 -mlittle-endian

Summary

Select little-endian byte order.

Syntax

`-mlittle-endian`
`-EL`

Description

This option selects little-endian byte ordering for data and instructions.

The default is little-endian byte ordering.

See also

-mbig-endian on page 36

2.3.5 -mthumb

Summary

Select Thumb instruction set.

Syntax

`-mthumb`

Description

This option sets the default assembler instruction set to Thumb (T32). The default instruction set is the instruction set if the source file does not select a specific instruction set using assembler source directives.

See also

`-marm` on page 35

2.4 Control options

2.4.1 --auto-import

Summary

Automatically import symbols.

Syntax

`--auto-import`

Description

This option instructs the assembler to automatically define a symbol as an external (imported) symbol if it is used and has not been declared with a different storage class.

See also

`--no-auto-import` on page 40

2.4.2 --no-auto-import

Summary

Do not automatically import symbols.

Syntax

`--no-auto-import`

Description

This option instructs the assembler not to automatically define as symbol as an external (imported) symbol and to require that all symbols are declared as local or external before their first use.

See also

--auto-import on page 39

2.4.3 --define

Summary

Define symbol.

Syntax

`--define symbol=value`

`-Dsymbol=value`

Description

This option defines the symbol *symbol* to the numeric value *value*.

2.4.4 --input-syntax

Summary

Set source file syntax.

Syntax

`--input-syntax=name`
`-fsyntax=name`

Description

This option selects the source file syntax. The syntaxes accepted by the assembler are:

- **arm** — Format is compatible with ARM Limited's `armasm` assembler.
- **iar** — Format is compatible with IAR's `iasmarm` assembler.
- **gnu** — Format is compatible with The Free Software Foundation's `gas` assembler.

There is one additional syntax identifier:

- **auto** — Source format is automatically derived from the input file.

The default is **auto** syntax where the assembler automatically determines the appropriate source file syntax using heuristics.

2.5 List file options

2.5.1 --list-diagnostics

Summary

Print diagnostics to listing.

Syntax

`--list-diagnostics`

Description

This option instructs the assembler to print all diagnostics at the end of the listing.

See also

--no-list-diagnostics on page 44

2.5.2 --no-list-diagnostics

Summary

Do not print diagnostics to listing.

Syntax

--no-list-diagnostics

Description

This option instructs the assembler not to print diagnostics at the end of the listing.

See also

--list-diagnostics on page 43

2.5.3 --list-file

Summary

Generate an assembler listing file.

Syntax

```
--list-file filename  
--list-file=filename  
-Lfilename
```

Description

Generates an assembler listing to the given filename.

2.5.4 --list-form-feed

Summary

Issue form feed between pages.

Syntax

`--list-form-feed`

Description

This option instructs the assembler to write a form feed between listing pages.

See also

--list-page-length on page 49, *--no-list-form-feed* on page 47

2.5.5 --no-list-form-feed

Summary

Do not issue form feed between pages.

Syntax

`--no-list-form-feed`

Description

This option inhibits the assembler writing a form feed between listing pages.

See also

--list-form-feed on page 46

2.5.6 --list-html

Summary

Format listing as an HTML document.

Syntax

`--list-html`

Description

This option instructs the assembler to format the assembler listing as an HTML document. In this mode, listing pagination controls are ignored.

See also

`--list-text` on page 52

2.5.7 --list-page-length

Summary

Set number of lines per page.

Syntax

`--list-page-length=value`

Description

This option sets the number of lines per page to *value*. If *value* is zero, which is the default, the listing is not divided into pages.

See also

`--list-form-feed` on page 46

2.5.8 --list-symbols

Summary

Print symbol table to listing.

Syntax

--list-symbols

Description

This option instructs the assembler to print a symbol table at the end of the listing.

See also

--no-list-symbols on page 51

2.5.9 --no-list-symbols

Summary

Do not print symbol table to listing.

Syntax

`--no-list-symbols`

Description

This option instructs the assembler not to print a symbol table at the end of the listing.

See also

--list-symbols on page 50

2.5.10 --list-text

Summary

Format listing as a text document.

Syntax

`--list-text`

Description

This option instructs the assembler to format the assembler listing as an HTML document. In this mode, listing pagination controls are effective.

This is the default.

See also

`--list-html` on page 48

2.6 Diagnostic options

2.6.1 --remarks

Summary

Issue remarks.

Syntax

`--remarks`

Description

This option instructs the assembler to issue remarks for potential issues during assembly. This is the default.

See also

--no-remarks on page 56, *--remarks-are-warnings* on page 55, *--remarks-are-errors* on page 54

2.6.2 --remarks-are-errors

Summary

Elevate remarks to errors.

Syntax

`--remarks-are-errors`

Description

This option elevates all remark diagnostics issued by the assembler to errors.

See also

--no-remarks on page 56, *--remarks* on page 53, , *--remarks-are-warnings* on page 55

2.6.3 --remarks-are-warnings

Summary

Elevate remarks to warnings.

Syntax

`--remarks-are-warnings`
`--remarks_are_warnings`

Description

This option elevates all remark diagnostics issued by the assembler to warnings.

See also

`--no-remarks` on page 56, `--remarks` on page 53, , `--remarks-are-errors` on page 54

2.6.4 --no-remarks

Summary

Suppress remarks.

Syntax

--no-remarks

Description

This option disables all remark diagnostics issued by the assembler. Although remarks are suppressed, the total number of remarks that are suppressed by the assembler is shown at the end of linking:

```
C:> segger-as --via=app.ind
Copyright (c) 2017-2018 SEGGER Microcontroller GmbH    www.segger.com
SEGGER Assembler 2.10 compiled Apr 11 2018 10:50:34

Assembly complete: 0 errors, 0 warnings, 2 remarks suppressed

C:> _
```

See also

--remarks-are-warnings on page 55, *--remarks-are-errors* on page 54

2.6.5 --silent

Summary

Do not show output.

Syntax

`--silent`

`-q`

Description

This option inhibits all assembler status messages; only diagnostic messages are shown.

See also

`--verbose` on page 58

2.6.6 --verbose

Summary

Increase verbosity.

Syntax

--verbose

-v

Description

This option increase the verbosity of the assembler by one level.

See also

--*silent* on page 57

2.6.7 --warnings

Summary

Issue warnings.

Syntax

`--warnings`

Description

This option instructs the assembler to issue warnings for dubious use or inputs. This is the default.

See also

`--no-warnings` on page 60, `--warnings-are-errors` on page 60

2.6.8 --warnings-are-errors

Summary

Elevate warnings to errors.

Syntax

`--warnings-are-errors`
`--fatal-warnings`

Description

This option elevates all warning diagnostics issued by the assembler to errors.

See also

`--no-warnings` on page 60, `--warnings` on page 59

2.6.9 --no-warnings

Summary

Suppress warnings.

Syntax

`--no-warnings`

Description

This option disables all warning diagnostics issued by the assembler. Although warnings are suppressed, the total number of warnings that are suppressed by the assembler is shown at the end of linking:

```
C:> segger-as --via=app.ind
SEGGER Macro Assembler V2.10 compiled Jul  9 2020 14:30:42
Copyright (c) 2018-2020 SEGGER Microcontroller GmbH    www.segger.com

Assembly complete: 0 errors, 1 warnings suppressed, 0 remarks

C:> _
```

See also

`--warnings` on page 59, `--warnings-are-errors` on page 60

Chapter 3

Expressions

GNU	IAR	ARM
(), prefix +, -, !, ~	(), prefix +, -, !, ~, LOW, HIGH, BYTE1, BYTE2, BYTE3, BYTE4, LWRD, HWRD, DATE, SFB, SFE, SIZEOF	(), prefix +, -, :NOT:, :LNOT:
*, /, %, <<, >>	*, /, %	*, /, %, :MOD:
, &, ^, !	+, -	<<, >>, :SHL:, :SHR:, :ROL:, :ROR:
+, -, =, <>, ≠, <, >, ≤, ≥	<<, >>	+, -, &, ^, :AND:, :OR:, :EOR:
&&,	&&, &	=, <>, ≠, /=, <, >, ≤, ≥
	, , ^, XOR	&&, , :LAND:, :LOR:, :LEOR:
	=, =, <>, ≠, <, >, ≤, ≥, UGT, ULT	

Chapter 4

GNU syntax

4.1 Source file directives

4.1.1 .INCBIN

Include binary file.

Syntax

`.INCBIN "filename"`

Description

Searches for the binary file *filename* using include paths. Inserts the binary file into the current section as data with alignment 1.

4.2 Mode directives

4.2.1 .ARM

Select Arm instruction set.

Syntax

.ARM

Description

Selects the Arm instruction set for subsequent instructions.

See also

.THUMB on page 66

4.2.3 .THUMB

Select Thumb instruction set.

Syntax

`.THUMB`

Description

Selects the Thumb instruction set for subsequent instructions.

See also

`.ARM` on page 64

4.3 Section directives

4.3.1 .DATA

Switch to data section

Syntax

.DATA

Description

Switches to the .data section.

4.3.2 .BSS

Switch to bss section

Syntax

.BSS

Description

Switches to the .bss section.

4.3.3 .TEXT

Switch to text section

Syntax

`.TEXT`

Description

Switches to the `.text` section.

4.3.4 .RODATA

Switch to data section

Syntax

.RODATA

Description

Switches to the .rodata section.

4.4 Symbol directives

4.4.1 .GLOBL, .GLOBAL, .EXTERN

Set external symbol binding.

Syntax

```
.GLOBL name, name...  
.GLOBAL name, name...  
.EXTERN name, name...
```

Description

Defines all symbols in the symbol list to have global binding so that they are externally visible.

4.4.2 .LOCAL

Set local symbol binding.

Syntax

`.LOCAL name, name...`

Description

Defines all symbols in the symbol list to have local binding so that they are not externally visible.

4.4.3 .WEAK

Set weak symbol binding.

Syntax

.WEAK *name*, *name*...

Description

Defines all symbols in the symbol list to have weak binding so that they can be overridden when linked.

4.4.4 .HIDDEN

Set symbol visibility to hidden.

Syntax

`.HIDDEN name, name...`

Description

Defines all symbols in the symbol list to have hidden visibility and is stored as **STV_HIDDEN** in the **st_other** member of the ELF symbol table entry.

4.4.5 .PROTECTED

Set symbol visibility to protected.

Syntax

`.PROTECTED name, name...`

Description

Defines all symbols in the symbol list to have protected visibility and is stored as **STV_PROTECTED** in the **st_other** member of the ELF symbol table entry.

4.4.6 .INTERNAL

Set symbol visibility to internal.

Syntax

.INTERNAL *name, name...*

Description

Defines all symbols in the symbol list to have internal visibility and is stored as **STV_INTERNAL** in the **st_other** member of the ELF symbol table entry.

4.5 Data allocation directives

4.5.1 .BYTE, .ASCII

Allocate 8-bit data.

Syntax

`.BYTE expr, expr...`
`.ASCII expr, expr...`

Description

Place the initialized 8-bit values in the expression list into memory at the location counter.

Example

SEGGER Assembler V2.10 www.segger.com
GNU_BYTE.s - Assembled Thu Nov 7 11:46:23 2019

```
1                                     .DATA
2
3      00000000 4120736D      .BYTE "A small list of primes:"
      00000004 616C6C20
      00000008 6C697374
      0000000C 206F6620
      00000010 7072696D
      00000014 65733A
4      00000017 02030507      .BYTE 2, 3, 5, 7, 11, 13
      0000001B 0B0D
5
6                                     .END
```

ASSEMBLY COMPLETE

No errors, no warnings, no remarks

See also

`.STRING`, `.ASCIZ` on page 78

4.5.2 .STRING, .ASCIZ

Allocate string data.

Syntax

.STRING *expr*, *expr*...

.ASCIZ *expr*, *expr*...

Description

Place the initialized string values in the expression list into memory at the location counter, each string followed by a trailing zero byte.

Example

SEGGER Assembler V2.10 www.segger.com
GNU_STRING.s - Assembled Thu Nov 7 11:46:23 2019

```

1                                     .DATA
2
3                                     @ String is terminated by a zero byte
4      00000000 49742073      .STRING "It simply works!"
      00000004 696D706C
      00000008 7920776F
      0000000C 726B7321
      00000010 00

5
6                                     @ Each string is terminated by a zero byte
7      00000011 53454747      .STRING "SEGGER", "The Embedded Experts"
      00000015 45520054
      00000019 68652045
      0000001D 6D626564
      00000021 64656420
      00000025 45787065
      00000029 72747300

8
9                                     .END
```

ASSEMBLY COMPLETE

No errors, no warnings, no remarks

See also

.BYTE, *.ASCII* on page 77

4.5.3 .2BYTE

Allocate 16-bit data.

Syntax

.2BYTE *expr*, *expr*...

Description

Place the initialized 16-bit values in the expression list into memory at the location counter.

These directives do not force the location counter to be halfword aligned when placing the data, and no warning diagnostic is issued when the location counter is not correctly aligned.

Example

SEGGER Assembler V2.10 www.segger.com
GNU_2BYTE.s - Assembled Thu Nov 7 11:46:23 2019

```

1                               .DATA
2
3                               @ Data that is correctly aligned
4      00000000 DEADBEEF      .2BYTE 0xADDE, 0xEFBE, 0xDEAD, 0xBEEF
      00000004 ADDEEFBE
5
6                               @ Data that is not aligned on a halfword
7      00000008                .SPACE 1
8      00000009 DEADBEEF      .2BYTE 0xADDE, 0xEFBE, 0xDEAD, 0xBEEF
      0000000D ADDEEFBE
9
10                              .END
```

ASSEMBLY COMPLETE

No errors, no warnings, no remarks

See also

.HWORD, *.SHORT* on page 81

4.5.4 .4BYTE

Allocate 32-bit data.

Syntax

.4BYTE *expr*, *expr*...

Description

Place the initialized 32-bit values in the expression list into memory at the location counter.

These directives do not force the location counter to be word aligned when placing the data, and do not issue a warning diagnostic when the location counter is not correctly aligned.

Example

```
SEGGER Assembler V2.10    www.segger.com
GNU_4BYTE.s - Assembled Thu Nov  7 11:46:23 2019

1                          .DATA
2
3                          @ Data that is correctly aligned
4          00000000 DEADBEEF  .4BYTE 0xEFBEADDE, 0xDEADBEEF
5          00000004 EFBEADDE
6
7          00000008          @ Data that is not aligned on a word
8          00000009 DEADBEEF  .SPACE 1
9          0000000D EFBEADDE  .4BYTE 0xEFBEADDE, 0xDEADBEEF
10
11                         .END
```

ASSEMBLY COMPLETE

No errors, no warnings, no remarks

See also

.WORD, *.INT*, *.LONG* on page 82

4.5.5 .HWORD, .SHORT

Allocate 16-bit data.

Syntax

.HWORD *expr*, *expr*...

.SHORT *expr*, *expr*...

Description

Place the initialized 16-bit values in the expression list into memory at the location counter.

These directives do not force the location counter to be halfword aligned when placing the data, but a warning diagnostic is issued when the location counter is not correctly aligned.

Example

```
SEGGER Assembler V2.10      www.segger.com
GNU_HWORD.s - Assembled Thu Nov  7 11:46:23 2019
```

```

1                                .DATA
2
3                                @ Data that is correctly aligned
4      00000000 DEADBEEF        .HWORD 0xADDE, 0xEFBE, 0xDEAD, 0xBEEF
      00000004 ADDEEFBE
5
6                                @ Data that is not aligned on a halfword
7      00000008                .SPACE 1
8      00000009 DEADBEEF        .HWORD 0xADDE, 0xEFBE, 0xDEAD, 0xBEEF
      0000000D ADDEEFBE
9
10                               .END
```

DIAGNOSTICS

```
GNU_HWORD.s:8: warning: 16-bit data does not fall on natural alignment at address 00000009
GNU_HWORD.s:8: warning: 16-bit data does not fall on natural alignment at address 0000000B
GNU_HWORD.s:8: warning: 16-bit data does not fall on natural alignment at address 0000000D
GNU_HWORD.s:8: warning: 16-bit data does not fall on natural alignment at address 0000000F
```

ASSEMBLY COMPLETE

No errors, 4 warnings, no remarks

See also

.2BYTE on page 79

4.5.6 .WORD, .INT, .LONG

Allocate 32-bit data.

Syntax

.WORD *expr*, *expr*...

.INT *expr*, *expr*...

.LONG *expr*, *expr*...

Description

Place the initialized 32-bit values in the expression list into memory at the location counter.

These directives do not force the location counter to be word aligned when placing the data, but a warning diagnostic is issued when the location counter is not correctly aligned.

Example

SEGGER Assembler V2.10 www.segger.com

GNU_WORD.s - Assembled Thu Nov 7 11:46:23 2019

```

1                                .DATA
2
3                                @ Data that is correctly aligned
4      00000000 DEADBEEF        .WORD 0xEFBEADDE, 0xDEADBEEF
      00000004 EFBEADDE
5
6                                @ Data that is not aligned on a word
7      00000008                .SPACE 1
8      00000009 DEADBEEF        .WORD 0xEFBEADDE, 0xDEADBEEF
      0000000D EFBEADDE
9
10                               .END
```

DIAGNOSTICS

GNU_WORD.s:8: warning: 32-bit data does not fall on natural alignment at address 00000009

GNU_WORD.s:8: warning: 32-bit data does not fall on natural alignment at address 0000000D

ASSEMBLY COMPLETE

No errors, 2 warnings, no remarks

See also

.4BYTE on page 80

4.5.7 .INST

Allocate 16-bit or 32-bit instruction.

Syntax

`.INST expr, expr...`

Description

Place the initialized values in the expression list into memory at the location counter as instruction codes. For values that fit into 16 bits, a 16-bit instruction is generated, and for others a 32-bit instruction is generated.

4.5.8 .INST.N

Allocate 16-bit instruction.

Syntax

`.INST.N expr, expr...`

Description

Place the initialized 16-bit values in the expression list into memory at the location counter as instruction codes.

4.5.9 .INST.W

Allocate 32-bit instruction.

Syntax

.INST.W *expr*, *expr*...

Description

Place the initialized 32-bit values in the expression list into memory at the location counter as instruction codes.

4.6 Section directives

4.6.1 .ALIGN

Align location counter.

Syntax

.ALIGN *align-power*, *fill-value*, *max-bytes*

Description

The *align-power* expression, which must be absolute, is the power of two to align the location counter to. An *align-power* of 3, for instance, will ensure that the location counter is a multiple of 2^3 , i.e. 8.

The optional *fill-value* defines the filler byte to use when aligning the location counter. If omitted, a default value of is selected: the default value for code sections is a nop instruction and for other sections it is zero.

The optional *max-bytes* defines the maximum number of bytes inserted to align the location counter. If more than *max-bytes* bytes are required for alignment, no alignment is performed.

4.6.2 .P2ALIGN, .P2ALIGNW, .P2ALIGNL

Align location counter.

Syntax

```
.P2ALIGN align-power, fill-value, max-bytes  
.P2ALIGNW align-power, fill-value, max-bytes  
.P2ALIGNL align-power, fill-value, max-bytes
```

Description

The *align-power* expression, which must be absolute, is the power of two to align the location counter to. An *align-power* of 3, for instance, will ensure that the location counter is a multiple of 2^3 , i.e. 8.

The optional *fill-value* defines the filler value to use when aligning the location counter. If omitted, a default value of is selected: the default value for code sections is a nop instruction and for other sections it is zero. If it is specified, the fill value is a one-byte pattern for .P2ALIGN, a two-byte pattern for .P2ALIGNW, and a 4-byte pattern for .P2ALIGNL.

The optional *max-bytes* defines the maximum number of bytes inserted to align the location counter. If more than *max-bytes* bytes are required for alignment, no alignment is performed.

4.7 Listing directives

4.7.1 .LIST

Turn on listing.

Syntax

.LIST

Description

Each line subsequently assembled will be printed to the listing file.

4.7.2 .NOLIST

Turn off listing.

Syntax

.LIST

Description

Each line subsequently assembled will be not be printed to the listing file.

4.7.3 .TITLE

Set listing title.

Syntax

`.TITLE string-expr`

Description

Sets the title that is shown at the top of each page in the listing.

4.7.4 .SUBTTL

Set listing subtitle.

Syntax

`.SUBTTL string-expr`

Description

Sets the subtitle that is shown below the title at the top of each page in the listing.

Chapter 5

Arm syntax

5.1 Mode directives

5.1.1 CODE16

Select Thumb instruction set.

Syntax

CODE16

Description

Selects the Thumb instruction set for subsequent instructions.

Example

```
SEGGER Assembler V2.10    www.segger.com
ARM_CODE16.s - Assembled Thu Nov  7 11:46:23 2019

1                          AREA |.text|, CODE
2
3                          CODE16
4
5          00000000 0120      MOVS R0, #1
6          00000002 4018      ADDS R0, R0, R1
7
8                          END
```

ASSEMBLY COMPLETE

No errors, no warnings, no remarks

See also

CODE32 on page 94

5.1.2 CODE32

Select Arm instruction set.

Syntax

CODE32

Description

Selects the Arm instruction set for subsequent instructions.

Example

```
SEGGER Assembler V2.10    www.segger.com
ARM_CODE32.s - Assembled Thu Nov  7 11:46:23 2019

1                          AREA |.text|, CODE
2
3                          CODE32
4
5          00000000 0100B0E3    MOVS R0, #1
6          00000004 010090E0    ADDS R0, R0, R1
7
8                          END
```

ASSEMBLY COMPLETE

No errors, no warnings, no remarks

See also

CODE16 on page 93

5.2 Listing directives

5.2.1 TTL

Set listing title.

Syntax

`TTL open-string`

Description

Sets the title that is shown at the top of each page in the listing.

5.2.2 SUBT

Set listing subtitle.

Syntax

`SUBT open-string`

Description

Sets the subtitle that is shown below the title at the top of each page in the listing.

Chapter 6

IAR syntax

6.1 Mode directives

6.1.1 ARM, CODE32

Select Arm instruction set.

Syntax

ARM
CODE32

Description

Selects the Arm instruction set for subsequent instructions.

See also

THUMB, *CODE16* on page 101

6.1.2 CODE

Select default instruction set.

Syntax

`CODE`

Description

Selects the Arm or Thumb instruction set according to that specified on the command line by the `-mthumb` or `-marm` options. If no instruction set is specified on the command line, the instruction defaults to Thumb.

6.1.3 THUMB, CODE16

Select Thumb instruction set.

Syntax

THUMB
CODE16

Description

Selects the Thumb instruction set for subsequent instructions.

See also

ARM, CODE32 on page 99

6.2 Data allocation directives

6.2.1 DC8, DCB

Allocate 8-bit data.

Syntax

DC8 *expr, expr...*

DCB *expr, expr...*

Description

Place the initialized 8-bit values in the expression list into memory at the location counter.

Example

```

SEGGER Assembler V2.10      www.segger.com
IAR_DC8.s - Assembled Thu Nov  7 11:46:23 2019

1                               SECTION `.data`:DATA(2)
2
3      00000000 4120736D      DCB "A small list of primes:"
      00000004 616C6C20
      00000008 6C697374
      0000000C 206F6620
      00000010 7072696D
      00000014 65733A
4      00000017 02030507      DC8  2, 3, 5, 7, 11, 13
      0000001B 0B0D
5
6                               END

ASSEMBLY COMPLETE

No errors, no warnings, no remarks

```

6.2.2 DC16, DCW

Allocate 16-bit data.

Syntax

DC16 *expr, expr...*

DCW *expr, expr...*

Description

Place the initialized 16-bit values in the expression list into memory at the location counter.

6.2.3 DC24

Allocate 24-bit data.

Syntax

DC24 *expr, expr...*

Description

Place the initialized 32-bit values in the expression list into memory at the location counter.

6.2.4 DC32, DCD

Allocate 32-bit data.

Syntax

`DC32 expr, expr...`

`DCD expr, expr...`

Description

Place the initialized 32-bit values in the expression list into memory at the location counter.

6.2.5 DCI16, DCIW

Allocate 16-bit instruction.

Syntax

DCI16 *expr*, *expr*...

DCIW *expr*, *expr*...

Description

Place the initialized 16-bit values in the expression list into memory at the location counter as instruction codes.

6.2.6 DCI32, DCID

Allocate 32-bit instruction.

Syntax

DCI32 *expr, expr...*

DCID *expr, expr...*

Description

Place the initialized 32-bit values in the expression list into memory at the location counter as instruction codes.

6.2.7 DS8, DS

Allocate space for 8-bit data.

Syntax

DS8 *expr*

DS *expr*

Description

Allocate space for *expr* 8-bit bytes. For zero-data sections such as **.bss**, the space is allocated; for sections that have content, such as **.text** and **.data**, the allocated space is initialized with zero bytes.

6.2.8 DS16, DSW

Allocate space for 16-bit data.

Syntax

DS16 *expr*

DSW *expr*

Description

Allocate space for *expr* 16-bit elements. For zero-data sections such as **.bss**, the space is allocated; for sections that have content, such as **.text** and **.data**, the allocated space is initialized with zero bytes.

6.2.9 DS24

Allocate space for 24-bit data.

Syntax

DS24 *expr*

Description

Allocate space for *expr* 24-bit elements. For zero-data sections such as **.bss**, the space is allocated; for sections that have content, such as **.text** and **.data**, the allocated space is initialized with zero bytes.

6.2.10 DS32, DSD

Allocate space for 32-bit data.

Syntax

DS32 *expr*

DSD *expr*

Description

Allocate space for *expr* 32-bit elements. For zero-data sections such as **.bss**, the space is allocated; for sections that have content, such as **.text** and **.data**, the allocated space is initialized with zero bytes.

Chapter 7

Instructions

7.1 Alphabetical list of instructions

7.1.1 ADC

Description

Add with carry

Thumb syntax (16-bit)

ADCS *Rd, Rn* [1]

Thumb syntax (32-bit)

ADC *Rd, Rn, #const* [2]

ADC *Rd, Rn, Rm, shift* [3]

ADCS *Rd, Rn, #const* [2]

ADCS *Rd, Rn, Rm, shift* [3]

Arm syntax

ADC *Rd, Rn, #const* [4]

ADC *Rd, Rn, Rm, shift* [5]

ADCS *Rd, Rn, #const* [4]

ADCS *Rd, Rn, Rm, shift* [5]

Notes

[1] *Rd, Rn* must be **R0...R7**; ordering *Rd, Rn, Rd* is also permitted; ADC permitted in an IT block

[2] *Rd, Rn* ≠ **PC**; *Rd, Rn* ≠ **SP**; *const* is one of *\$00xy00xy*, *\$xy00xy00*, *\$xyxyxyxy*, *\$xy* **LSL** *0...24*

[3] *Rd, Rn, Rm* ≠ **PC**; *Rd, Rn, Rm* ≠ **SP**; *shift* is one of **LSL** *#0...31*, **LSR** *#1...32*, **ASR** *#1...32*, **RRX**

[4] *Rd, Rn* ≠ **PC**; *Rd, Rn* ≠ **SP**; *const* is *\$xy* **ROR** *2n*

[5] *Rd, Rn, Rm* ≠ **PC**; *Rd, Rn, Rm* ≠ **SP**; *shift* is one of **LSL** *#0...31/Rs*, **LSR** *#1...32/Rs*, **ASR** *#1...32/Rs*, **ROR** *#1...31/Rs*, **RRX**

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
ADC	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●

● Thumb mode only

● Arm and Thumb modes

7.1.2 ADD

Description

Add

Thumb syntax (16-bit)

ADD	<i>Rd, Rn</i>	[1]
ADD	SP, SP, #0...508	[2]
ADD	<i>Rd</i> , SP, #0...1020	[3]
ADD	<i>Rd</i> , PC, #0...1020	[3]
ADDS	<i>Rd, Rn, Rm</i>	[4]
ADDS	<i>Rd</i> , #0...255	[5]
ADDS	<i>Rd, Rn</i> , #0...7	[6]

Thumb syntax (32-bit)

ADD	<i>Rd, Rn</i> , #0...4095	[7]
ADD	<i>Rd, Rn</i> , #const	[8]
ADD	<i>Rd, Rn, Rm</i> , shift	[9]
ADDS	<i>Rd, Rn</i> , #const	[8]
ADDS	<i>Rd, Rn, Rm</i> , shift	[9]

Arm syntax

ADD	<i>Rd, Rn</i> , #const	[10]
ADD	<i>Rd, Rn, Rm</i> , shift	[11]
ADD	<i>Rd</i> , SP, <i>Rm</i> , shift	[12]
ADDS	<i>Rd, Rn</i> , #const	[13]
ADDS	<i>Rd, Rn, Rm</i> , shift	[11]
ADDS	<i>Rd</i> , SP, <i>Rm</i> , shift	[12]

Notes

- [1] ordering *Rd, Rn, Rd* is also permitted
- [2] offset a multiple of 4
- [3] *Rd* must be R0...R7; offset a multiple of 4
- [4] *Rd, Rn, Rm* must be R0...R7; ADD permitted in an IT block
- [5] *Rd* must be R0...R7; ADD permitted in an IT block
- [6] *Rd, Rn* must be R0...R7; ADD permitted in an IT block
- [7] *Rd* ≠ PC; *Rd* ≠ SP
- [8] *Rd, Rn* ≠ PC; *Rd, Rn* ≠ SP; *const* is one of \$00xy00xy, \$xy00xy00, \$xyxyxyxy, \$xy LSL 0...24
- [9] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP; *shift* is one of LSL #0...31, LSR #1...32, ASR #1...32, RRX
- [10] *Rd* ≠ SP; *const* is \$xy ROR 2n
- [11] *Rd, Rm* ≠ PC; *Rd, Rm* ≠ SP; *shift* is one of LSL #0...31/*Rs*, LSR #1...32/*Rs*, ASR #1...32/*Rs*, ROR #1...31/*Rs*, RRX
- [12] *Rd, Rm* ≠ PC; *Rm* ≠ SP; *shift* is one of LSL #0...31/*Rs*, LSR #1...32/*Rs*, ASR #1...32/*Rs*, ROR #1...31/*Rs*, RRX
- [13] *Rd* ≠ PC; *Rd* ≠ SP; *const* is \$xy ROR 2n

Availability

Instruction	8-M,Main	●
	8-M,Base	●
	7E-M	●
	7-M	●
	6S-M	●
	6-M	●
	8-R	●
	8.4-A	●
	8.3-A	●
	8.2-A	●
	8.1-A	●
	8-A	●
	7-A	●
	7-R	●
7VE	●	
6T2	●	
6ZK	●	
6Z	●	
6KZ	●	
6K	●	
6J	●	
6	●	
5TE	●	
5T	●	
4T	●	
ADD		

- Thumb mode only
- Arm and Thumb modes

7.1.3 ADDW

Description

Add

Thumb syntax (16-bit)

ADDW *Rd*, PC, #0...1020 [1]

Thumb syntax (32-bit)

ADDW *Rd*, *Rn*, #0...4095 [2]

Notes

[1] *Rd* must be R0...R7; offset a multiple of 4

[2] *Rd* ≠ PC; *Rd* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
ADDW										●	●	●	●	●	●	●	●	●	●			●	●		●

● Thumb mode only

7.1.4 AND

Description

Bitwise and

Thumb syntax (16-bit)

ANDS *Rd, Rn* [1]

Thumb syntax (32-bit)

AND *Rd, Rn, #const* [2]

AND *Rd, Rn, Rm, shift* [3]

ANDS *Rd, Rn, #const* [2]

ANDS *Rd, Rn, Rm, shift* [3]

Arm syntax

AND *Rd, Rn, #const* [4]

AND *Rd, Rn, Rm, shift* [5]

ANDS *Rd, Rn, #const* [4]

ANDS *Rd, Rn, Rm, shift* [5]

Notes

[1] *Rd, Rn* must be **R0...R7**; ordering *Rd, Rn, Rd* is also permitted; **AND** permitted in an IT block

[2] *Rd, Rn* ≠ **PC**; *Rd, Rn* ≠ **SP**; *const* is one of **\$00xy00xy**, **\$xy00xy00**, **\$xyxyxyxy**, **\$xy LSL #0...24**

[3] *Rd, Rn, Rm* ≠ **PC**; *Rd, Rn, Rm* ≠ **SP**; *shift* is one of **LSL #0...31**, **LSR #1...32**, **ASR #1...32**, **RRX**

[4] *Rd, Rn* ≠ **PC**; *Rd, Rn* ≠ **SP**; *const* is **\$xy ROR #2n**

[5] *Rd, Rn, Rm* ≠ **PC**; *Rd, Rn, Rm* ≠ **SP**; *shift* is one of **LSL #0...31/Rs**, **LSR #1...32/Rs**, **ASR #1...32/Rs**, **ROR #1...31/Rs**, **RRX**

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
AND	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●

● Thumb mode only

● Arm and Thumb modes

7.1.5 ASR

Description

Arithmetic shift right

Thumb syntax (16-bit)

ASRS *Rd, Rn* [1]
 ASRS *Rd, Rn, #1...32* [1]

Thumb syntax (32-bit)

ASR *Rd, Rn, Rm* [2]
 ASR *Rd, Rn, #1...32* [3]
 ASRS *Rd, Rn, Rm* [2]
 ASRS *Rd, Rn, #1...32* [3]

Arm syntax

ASR *Rd, Rn, Rm* [2]
 ASR *Rd, Rn, #1...32* [3]
 ASRS *Rd, Rn, Rm* [2]
 ASRS *Rd, Rn, #1...32* [3]

Notes

[1] *Rd, Rn* must be **R0...R7**; ASR permitted in an IT block

[2] *Rd, Rn, Rm* ≠ **PC**; *Rd, Rn, Rm* ≠ **SP**

[3] *Rd, Rn* ≠ **PC**; *Rd, Rn* ≠ **SP**

Availability

Instruction	8-M.Main	●
	8-M.Base	●
	7E-M	●
	7-M	●
	6S-M	●
	6-M	●
	8-R	●
	8.4-A	●
	8.3-A	●
	8.2-A	●
	8.1-A	●
	8-A	●
	7-A	●
7-R	●	
7VE	●	
6T2	●	
6ZK	●	
6Z	●	
6KZ	●	
6K	●	
6J	●	
6	●	
5TE	●	
5T	●	
4T	●	
ASR		

● Thumb mode only

● Arm and Thumb modes

7.1.10 BFC

Description

Bitfield clear

Thumb syntax (32-bit)

BFC *Rd, #lsb, #width* [1]

Arm syntax

BFC *Rd, #lsb, #width* [1]

Notes

[1] *Rd, Rn* ≠ PC; *Rd, Rn* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
BFC										●	●	●	●	●	●	●	●	●	●			●	●		●

- Thumb mode only
- Arm and Thumb modes

7.1.11 BFI

Description

Bitfield insert

Thumb syntax (32-bit)

BFI *Rd, Rn, #lsb, #width* [1]

Arm syntax

BFI *Rd, Rn, #lsb, #width* [1]

Notes

[1] *Rd, Rn* ≠ PC; *Rd, Rn* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
BFI										●	●	●	●	●	●	●	●	●	●			●	●		●

- Thumb mode only
- Arm and Thumb modes

7.1.16 BIC

Description

Bitwise clear

Thumb syntax (16-bit)

BICS *Rd, Rn* [1]

Thumb syntax (32-bit)

BIC *Rd, Rn, #const* [2]

BIC *Rd, Rn, Rm, shift* [3]

BICS *Rd, Rn, #const* [2]

BICS *Rd, Rn, Rm, shift* [3]

Arm syntax

BIC *Rd, Rn, #const* [4]

BIC *Rd, Rn, Rm, shift* [5]

BICS *Rd, Rn, #const* [4]

BICS *Rd, Rn, Rm, shift* [5]

Notes

[1] *Rd, Rn* must be **R0...R7**; **BIC** permitted in an IT block

[2] *Rd, Rn* ≠ **PC**; *Rd, Rn* ≠ **SP**; *const* is one of *\$00xy00xy*, *\$xy00xy00*, *\$xyxyxyxy*, *\$xy* **LSL** *0...24*

[3] *Rd, Rn, Rm* ≠ **PC**; *Rd, Rn, Rm* ≠ **SP**; *shift* is one of **LSL** *#0...31*, **LSR** *#1...32*, **ASR** *#1...32*, **RRX**

[4] *Rd, Rn* ≠ **PC**; *const* is *\$xy* **ROR** *2n*

[5] *Rd, Rn, Rm* ≠ **PC**; *shift* is one of **LSL** *#0...31/Rs*, **LSR** *#1...32/Rs*, **ASR** *#1...32/Rs*, **ROR** *#1...31/Rs*, **RRX**

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
BIC	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●

● Thumb mode only

● Arm and Thumb modes

7.1.17 BKPT

Description

Breakpoint

Thumb syntax (16-bit)

BKPT #0...255

Arm syntax

BKPT *Rd*, #0...65535

Availability

Instruction	8-M.Main	●
	8-M.Base	●
	7E-M	●
	7-M	●
	6S-M	●
	6-M	●
	8-R	●
	8.4-A	●
	8.3-A	●
	8.2-A	●
	8.1-A	●
	8-A	●
	7-A	●
	7-R	●
7VE	●	
6T2	●	
6ZK	●	
6Z	●	
6KZ	●	
6K	●	
6J	●	
6	●	
5TE	●	
5T	●	
4T		
BKPT		

- Thumb mode only
- Arm and Thumb modes

7.1.18 BL

Description

Branch link

Thumb syntax (32-bit)

BL *Label*

Arm syntax

BL *Label*

Availability

Instruction	8-M:Main	●
	8-M:Base	●
	7E-M	●
	7-M	●
	6S-M	●
	6-M	●
	8-R	●
	8.4-A	●
	8.3-A	●
	8.2-A	●
	8.1-A	●
	8-A	●
	7-A	●
	7-R	●
7VE	●	
6T2	●	
6ZK	●	
6Z	●	
6KZ	●	
6K	●	
6J	●	
6	●	
5TE	●	
5T	●	
4T	●	
BL		

- Thumb mode only
- Arm and Thumb modes

7.1.19 BLE

Description

Branch if less than or equal

Thumb syntax (16-bit)

BLE *Label* [1]

Thumb syntax (32-bit)

BLE *Label* [1]

Notes

[1] if in an IT block, can only appear as the last instruction

Availability

Instruction	8-M.Main	●
	8-M.Base	●
	7E-M	●
	7-M	●
	6S-M	●
	6-M	●
	8-R	●
	8.4-A	●
	8.3-A	●
	8.2-A	●
	8.1-A	●
	8-A	●
	7-A	●
	7-R	●
7VE	●	
6T2	●	
6ZK	●	
6Z	●	
6KZ	●	
6K	●	
6J	●	
6	●	
5TE	●	
5T	●	
4T	●	
BLE		

● Thumb mode only

7.1.20 BLO

Description

Branch if no carry

Thumb syntax (16-bit)

BLO *Label* [1]

Thumb syntax (32-bit)

BLO *Label* [1]

Notes

[1] if in an IT block, can only appear as the last instruction

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
BLO	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●

● Thumb mode only

7.1.23 BLX

Description

Branch link and exchange

Thumb syntax (16-bit)

BLX *Rd* [1]

Thumb syntax (32-bit)

BLX *Label*

Arm syntax

BLX *Rd* [1]
BLX *Label*

Notes

[1] *Rd* ≠ PC; *Rd* ≠ SP

Availability

Instruction	8-M.Main	●
	8-M.Base	●
	7E-M	●
	7-M	●
	6S-M	●
	6-M	●
	8-R	●
	8.4-A	●
	8.3-A	●
	8.2-A	●
	8.1-A	●
	8-A	●
	7-A	●
	7-R	●
7VE	●	
6T2	●	
6ZK	●	
6Z	●	
6KZ	●	
6K	●	
6J	●	
6	●	
5TE	●	
5T	●	
4T	●	
BLX		

- Thumb mode only
- Arm and Thumb modes

7.1.27 BVC

Description

Branch if no overflow

Thumb syntax (16-bit)

BVC *Label* [1]

Thumb syntax (32-bit)

BVC *Label* [1]

Notes

[1] if in an IT block, can only appear as the last instruction

Availability

Instruction	8-M.Main	●
	8-M.Base	●
	7E-M	●
	7-M	●
	6S-M	●
	6-M	●
	8-R	●
	8.4-A	●
	8.3-A	●
	8.2-A	●
	8.1-A	●
	8-A	●
	7-A	●
	7-R	●
	7VE	●
6T2	●	
6ZK	●	
6Z	●	
6KZ	●	
6K	●	
6J	●	
6	●	
5TE	●	
5T	●	
4T	●	
BVC		

● Thumb mode only

7.1.28 BVS

Description

Branch is overflow

Thumb syntax (16-bit)

BVS *Label* [1]

Thumb syntax (32-bit)

BVS *Label* [1]

Notes

[1] if in an IT block, can only appear as the last instruction

Availability

Instruction	8-M.Main	●
	8-M.Base	●
	7E-M	●
	7-M	●
	6S-M	●
	6-M	●
	8-R	●
	8.4-A	●
	8.3-A	●
	8.2-A	●
	8.1-A	●
	8-A	●
	7-A	●
7-R	●	
7VE	●	
6T2	●	
6ZK	●	
6Z	●	
6KZ	●	
6K	●	
6J	●	
6	●	
5TE	●	
5T	●	
4T	●	
	BVS	

● Thumb mode only

7.1.33 CDP

Description

Coprocessor data operation

Thumb syntax (32-bit)

CDP *Pn*, 0...15, *CRx*, *CRy*, *CRz* [1]

Arm syntax

CDP *Pn*, 0...15, *CRx*, *CRy*, *CRz* [1]

Notes

[1] constants may be prefixed by “#”

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
CDP										●	●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.35 CLREX

Description

Clear exclusive

Thumb syntax (32-bit)

CLREX

Arm syntax

CLREX

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
CLREX						●	●	●	●	●	●	●	●	●	●	●	●	●	●			●	●	●	●

- Arm mode only
- Thumb mode only
- Arm and Thumb modes

7.1.36 CLZ

Description

Count leading zeros

Thumb syntax (32-bit)

CLZ *Rd, Rn* [1]

Arm syntax

CLZ *Rd, Rn* [2]

Notes

- [1] *Rd, Rn* ≠ PC; *Rd, Rn* ≠ SP; only available when 'Relaxed UAL' enabled
- [2] *Rd, Rn* ≠ PC; only available when 'Relaxed UAL' enabled

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
CLZ		●	●							●	●	●	●	●	●	●	●	●	●			●	●		●

- Arm mode only
- Thumb mode only

7.1.38 CMP

Description

Compare

Thumb syntax (16-bit)

CMP	<i>Rn, Rm</i>	[1]
CMP	<i>Rn, #0...255</i>	[2]
CMPS	<i>Rd, Rn</i>	[3]
CMPS	<i>Rd, #0...255</i>	[4]

Thumb syntax (32-bit)

CMP	<i>Rn, #const</i>	[5]
CMP	<i>Rn, Rm, shift</i>	[6]
CMPS	<i>Rd, #const</i>	[7]
CMPS	<i>Rd, Rn, shift</i>	[8]

Arm syntax

CMP	<i>Rn, #const</i>	[9]
CMP	<i>Rn, Rm, shift</i>	[10]
CMPS	<i>Rd, #const</i>	[11]
CMPS	<i>Rd, Rn, shift</i>	[12]

Notes

- [1] *Rn, Rm* ≠ PC; *Rm* ≠ SP
- [2] *Rn* must be R0...R7
- [3] *Rd, Rn* ≠ PC; *Rn* ≠ SP
- [4] *Rd* must be R0...R7
- [5] *Rn* ≠ PC; *const* is one of \$00xy00xy, \$xy00xy00, \$xyxyxyxy, \$xy LSL 0...24
- [6] *Rn, Rm* ≠ PC; *Rm* ≠ SP; *shift* is one of LSL #0...31, LSR #1...32, ASR #1...32, RRX
- [7] *Rd* ≠ PC; *const* is one of \$00xy00xy, \$xy00xy00, \$xyxyxyxy, \$xy LSL 0...24
- [8] *Rd, Rn* ≠ PC; *Rn* ≠ SP; *shift* is one of LSL #0...31, LSR #1...32, ASR #1...32, RRX
- [9] *Rn* ≠ PC; *const* is \$xy ROR 2n
- [10] *Rn, Rm* ≠ PC; *Rm* ≠ SP; *shift* is one of LSL #0...31/*Rs*, LSR #1...32/*Rs*, ASR #1...32/*Rs*, ROR #1...31/*Rs*, RRX
- [11] *Rd* ≠ PC; *const* is \$xy ROR 2n
- [12] *Rd, Rn* ≠ PC; *Rn* ≠ SP; *shift* is one of LSL #0...31/*Rs*, LSR #1...32/*Rs*, ASR #1...32/*Rs*, ROR #1...31/*Rs*, RRX

Availability

Instruction	8-M.Main	●
	8-M.Base	●
	7E-M	●
	7-M	●
	6S-M	●
	6-M	●
	8-R	●
	8.4-A	●
	8.3-A	●
	8.2-A	●
	8.1-A	●
	8-A	●
	7-A	●
7-R	●	
7VE	●	
6T2	●	
6ZK	●	
6Z	●	
6KZ	●	
6K	●	
6J	●	
6	●	
5TE	●	
5T	●	
4T	●	
CMP		

- Thumb mode only
- Arm and Thumb modes

7.1.39 CPS

Description

Change processor state

Thumb syntax (32-bit)

CPS #0...31 [1]

Arm syntax

CPS #0...31 [1]

Notes

[1] not permitted in an IT block

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
CPS											•	•	•	•	•	•	•	•	•						

- Arm and Thumb modes

7.1.40 CPSID

Description

Change processor state

Thumb syntax (16-bit)

CPSID *flags* [1]

Thumb syntax (32-bit)

CPSID *flags* [1]

CPSID *flags*, #0...31 [1]

Arm syntax

CPSID *flags* [2]

CPSID *flags*, #0...31 [2]

Notes

[1] *flags* is any combination of 'A', 'I', and 'F'; not permitted in an IT block

[2] *flags* is any combination of 'A', 'I', and 'F'

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
CPSID				●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●

● Thumb mode only

● Arm and Thumb modes

7.1.41 CPSIE

Description

Change processor state

Thumb syntax (16-bit)

CPSIE *flags* [1]

Thumb syntax (32-bit)

CPSIE *flags* [1]

CPSIE *flags*, #0...31 [1]

Arm syntax

CPSIE *flags* [2]

CPSIE *flags*, #0...31 [2]

Notes

[1] *flags* is any combination of 'A', 'I', and 'F'; not permitted in an IT block

[2] *flags* is any combination of 'A', 'I', and 'F'

Availability

Instruction	8-M.Main	●
	8-M.Base	●
	7E-M	●
	7-M	●
	6S-M	●
	6-M	●
	8-R	●
	8.4-A	●
	8.3-A	●
	8.2-A	●
	8.1-A	●
	8-A	●
	7-A	●
	7-R	●
7VE	●	
6T2	●	
6ZK	●	
6Z	●	
6KZ	●	
6K	●	
6J	●	
6	●	
5TE		
5T		
4T		
	CPSIE	

● Thumb mode only

● Arm and Thumb modes

7.1.42 CPY

Description

Copy

Thumb syntax (16-bit)

CPY *Rd, Rn*

Arm syntax

CPY *Rd, Rn*

Availability

Instruction	8-M.Main	●
	8-M.Base	●
	7E-M	●
	7-M	●
	6S-M	●
	6-M	●
	8-R	●
	8.4-A	●
	8.3-A	●
	8.2-A	●
	8.1-A	●
	8-A	●
7-A	●	
7-R	●	
7VE	●	
6T2	●	
6ZK	●	
6Z	●	
6KZ	●	
6K	●	
6J	●	
6	●	
5TE	●	
5T	●	
4T	●	
CPY		

- Arm mode only
- Thumb mode only
- Arm and Thumb modes

7.1.43 CRC32B

Description

CRC32 checksum byte

Thumb syntax (32-bit)

CRC32B *Rd, Rn, Rm*

Arm syntax

CRC32B *Rd, Rn, Rm*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
CRC32B														•	•	•	•	•	•						

- Arm and Thumb modes

7.1.44 CRC32CB

Description

CRC32-C checksum byte

Thumb syntax (32-bit)

CRC32CB *Rd, Rn, Rm*

Arm syntax

CRC32CB *Rd, Rn, Rm*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
CRC32CB														•	•	•	•	•	•						

- Arm and Thumb modes

7.1.45 CRC32CH

Description

CRC32-C checksum halfword

Thumb syntax (32-bit)

CRC32CH *Rd, Rn, Rm*

Arm syntax

CRC32CH *Rd, Rn, Rm*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
CRC32CH														•	•	•	•	•	•						

- Arm and Thumb modes

7.1.46 CRC32CW

Description

CRC32-C checksum word

Thumb syntax (32-bit)

CRC32CW *Rd, Rn, Rm*

Arm syntax

CRC32CW *Rd, Rn, Rm*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
CRC32CW														•	•	•	•	•	•						

- Arm and Thumb modes

7.1.47 CRC32H

Description

CRC32 checksum halfword

Thumb syntax (32-bit)

CRC32H *Rd, Rn, Rm*

Arm syntax

CRC32H *Rd, Rn, Rm*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
CRC32H														•	•	•	•	•	•						

- Arm and Thumb modes

7.1.48 CRC32W

Description

CRC32 checksum word

Thumb syntax (32-bit)

CRC32W *Rd, Rn, Rm*

Arm syntax

CRC32W *Rd, Rn, Rm*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
CRC32W														•	•	•	•	•	•						

- Arm and Thumb modes

7.1.49 DBG

Description

Debug

Thumb syntax (32-bit)

DBG #0...15

Arm syntax

DBG #0...15

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
DBG				●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●	●		●

- Arm mode only
- Thumb mode only
- Arm and Thumb modes

7.1.50 DCPS1

Description

Debug switch to exception level 1

Thumb syntax (32-bit)

DCPS1

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
DCPS1														•	•	•	•	•	•						

- Thumb mode only

7.1.51 DCPS2

Description

Debug switch to exception level 2

Thumb syntax (32-bit)

DCPS2

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
DCPS2														●	●	●	●	●	●						

- Thumb mode only

7.1.52 DCPS3

Description

Debug switch to exception level 3

Thumb syntax (32-bit)

DCPS3

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
DCPS3														•	•	•	•	•	•						

- Thumb mode only

7.1.53 DMB

Description

Data memory barrier

Thumb syntax (32-bit)

DMB #0...15

Arm syntax

DMB #0...15

Availability

Instruction	8-M.Main	●
	8-M.Base	●
	7E-M	●
	7-M	●
	6S-M	●
	6-M	●
	8-R	●
	8.4-A	●
	8.3-A	●
	8.2-A	●
	8.1-A	●
	8-A	●
	7-A	●
	7-R	●
	7VE	●
	6T2	
6ZK		
6Z		
6KZ		
6K		
6J		
6		
5TE		
5T		
4T		

- Thumb mode only
- Arm and Thumb modes

7.1.54 DSB

Description

Data synchronization barrier

Thumb syntax (32-bit)

DSB #0...15

Arm syntax

DSB #0...15

Availability

Instruction	8-M.Main	●
	8-M.Base	●
	7E-M	●
	7-M	●
	6S-M	●
	6-M	●
	8-R	●
	8.4-A	●
	8.3-A	●
	8.2-A	●
	8.1-A	●
	8-A	●
	7-A	●
	7-R	●
	7VE	●
	6T2	
6ZK		
6Z		
6KZ		
6K		
6J		
6		
5TE		
5T		
4T		
DSB		

- Thumb mode only
- Arm and Thumb modes

7.1.55 EOR

Description

Bitwise exclusive-or

Thumb syntax (16-bit)

EORS *Rd, Rn* [1]

Thumb syntax (32-bit)

EOR *Rd, Rn, #const* [2]

EOR *Rd, Rn, Rm, shift* [3]

EORS *Rd, Rn, #const* [2]

EORS *Rd, Rn, Rm, shift* [3]

Arm syntax

EOR *Rd, Rn, #const* [4]

EOR *Rd, Rn, Rm, shift* [5]

EORS *Rd, Rn, #const* [4]

EORS *Rd, Rn, Rm, shift* [5]

Notes

[1] *Rd, Rn* must be **R0...R7**; ordering *Rd, Rn, Rd* is also permitted; EOR permitted in an IT block

[2] *Rd, Rn* ≠ **PC**; *Rd, Rn* ≠ **SP**; *const* is one of \$00xy00xy, \$xy00xy00, \$xyxyxyxy, \$xy **LSL** 0...24

[3] *Rd, Rn, Rm* ≠ **PC**; *Rd, Rn, Rm* ≠ **SP**; *shift* is one of **LSL** #0...31, **LSR** #1...32, **ASR** #1...32, **RRX**

[4] *Rd, Rn* ≠ **PC**; *Rd, Rn* ≠ **SP**; *const* is \$xy **ROR** 2n

[5] *Rd, Rn, Rm* ≠ **PC**; *Rd, Rn, Rm* ≠ **SP**; *shift* is one of **LSL** #0...31/*Rs*, **LSR** #1...32/*Rs*, **ASR** #1...32/*Rs*, **ROR** #1...31/*Rs*, **RRX**

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
EOR	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●

● Thumb mode only

● Arm and Thumb modes

7.1.56 ERET

Description

Exception return

Thumb syntax (32-bit)

ERET

Arm syntax

ERET

Availability

Instruction	8-M.Main	
	8-M.Base	
	7E-M	
	7-M	
	6S-M	
	6-M	
	8-R	
	8.4-A	
	8.3-A	
	8.2-A	
	8.1-A	
	8-A	
	7-A	
	7-R	
7VE	●	
6T2		
6ZK		
6Z		
6KZ		
6K		
6J		
6		
5TE		
5T		
4T		

- Arm and Thumb modes

7.1.57 ESB

Description

Exception synchronization barrier

Thumb syntax (32-bit)

ESB

Arm syntax

ESB

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
ESB																•	•	•	•						

- Arm and Thumb modes

7.1.58 FABSD

Description

Floating-point absolute

Thumb syntax (32-bit)

FABSD *Dd, Dn* [1]

Arm syntax

FABSD *Dd, Dn* [1]

Notes

[1] only available when 'Relaxed UAL' enabled

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
FABSD											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.59 FABSS

Description
Floating-point absolute

Thumb syntax (32-bit)

FABSS *Sd, Sn* [1]

Arm syntax

FABSS *Sd, Sn* [1]

Notes
[1] only available when 'Relaxed UAL' enabled

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
FABSS											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.62 FCMPD

Description

Floating-point compare

Thumb syntax (32-bit)

FCMPD *Dd, Dn* [1]

Arm syntax

FCMPD *Dd, Dn* [1]

Notes

[1] only available when 'Relaxed UAL' enabled

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
FCMPD											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.63 FCMPED

Description

Floating-point compare raising exception on NaN

Thumb syntax (32-bit)

FCMPED *Dd, Dn* [1]

Arm syntax

FCMPED *Dd, Dn* [1]

Notes

[1] only available when 'Relaxed UAL' enabled

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
FCMPED											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.64 FCMPEs

Description

Floating-point compare raising exception on NaN

Thumb syntax (32-bit)

FCMPES *Sd, Sn* [1]

Arm syntax

FCMPES *Sd, Sn* [1]

Notes

[1] only available when 'Relaxed UAL' enabled

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
FCMPES											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.65 FCMPS

Description

Floating-point compare

Thumb syntax (32-bit)

FCMPS *Sd, Sn* [1]

Arm syntax

FCMPS *Sd, Sn* [1]

Notes

[1] only available when 'Relaxed UAL' enabled

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
FCMPS											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.66 FCVTDS

Thumb syntax (32-bit)

FCVTDS *Dd, Sn* [1]

Arm syntax

FCVTDS *Dd, Sn* [1]

Notes

[1] only available when 'Relaxed UAL' enabled

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
FCVTDS											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.67 FCVTSD

Thumb syntax (32-bit)

FCVTSD *Sd, Dn* [1]

Arm syntax

FCVTSD *Sd, Dn* [1]

Notes

[1] only available when 'Relaxed UAL' enabled

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
FCVTSD											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.70 FMDRR

Thumb syntax (32-bit)

FMDRR *Dd, Rn, Rm* [1]

Notes

[1] only available when 'Relaxed UAL' enabled

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
FMDRR											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only

7.1.72 FMRS

Thumb syntax (32-bit)

FMRS *Rd, Sn* [1]

Notes

[1] only available when 'Relaxed UAL' enabled

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
FMRS											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only

7.1.73 FMSR

Thumb syntax (32-bit)

FMSR *Sd, Rn* [1]

Notes

[1] only available when 'Relaxed UAL' enabled

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
FMSR											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only

7.1.74 FMSTAT

Thumb syntax (32-bit)

FMSTAT

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
FMSTAT											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only

7.1.75 FMULD

Description

Floating-point multiply

Thumb syntax (32-bit)

FMULD *Dd, Dn, Dm* [1]

Arm syntax

FMULD *Dd, Dn, Dm* [1]

Notes

[1] only available when 'Relaxed UAL' enabled

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
FMULD											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.76 FMULS

Description

Floating-point multiply

Thumb syntax (32-bit)

FMULS *Sd, Sn, Sm* [1]

Arm syntax

FMULS *Sd, Sn, Sm* [1]

Notes

[1] only available when 'Relaxed UAL' enabled

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
FMULS											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.77 FNEGD

Description

Floating-point negate

Thumb syntax (32-bit)

FNEGD *Dd, Dn* [1]

Arm syntax

FNEGD *Dd, Dn* [1]

Notes

[1] only available when 'Relaxed UAL' enabled

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
FNEGD											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.79 FSITOD

Thumb syntax (32-bit)

FSITOD *Dd, Sn* [1]

Arm syntax

FSITOD *Dd, Sn* [1]

Notes

[1] only available when 'Relaxed UAL' enabled

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
FSITOD											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.80 FSITOS

Thumb syntax (32-bit)

FSITOS *Sd, Sn* [1]

Arm syntax

FSITOS *Sd, Sn* [1]

Notes

[1] only available when 'Relaxed UAL' enabled

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
FSITOS											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.81 FSQRTD

Description

Floating-point square root

Thumb syntax (32-bit)

FSQRTD *Dd, Dn* [1]

Arm syntax

FSQRTD *Dd, Dn* [1]

Notes

[1] only available when 'Relaxed UAL' enabled

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
FSQRTD											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.82 FSQRTS

Description

Floating-point square root

Thumb syntax (32-bit)

FSQRTS *Sd, Sn* [1]

Arm syntax

FSQRTS *Sd, Sn* [1]

Notes

[1] only available when 'Relaxed UAL' enabled

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
FSQRTS											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.86 FTOSIZS

Thumb syntax (32-bit)

FTOSIZS *Sd, Sn* [1]

Arm syntax

FTOSIZS *Sd, Sn* [1]

Notes

[1] only available when 'Relaxed UAL' enabled

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
FTOSIZS											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.87 FTOUIZD

Thumb syntax (32-bit)

FTOUIZD *Sd, Dn* [1]

Arm syntax

FTOUIZD *Sd, Dn* [1]

Notes

[1] only available when 'Relaxed UAL' enabled

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
FTOUIZD											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.89 FUITOD

Thumb syntax (32-bit)

FUITOD *Dd, Sn* [1]

Arm syntax

FUITOD *Dd, Sn* [1]

Notes

[1] only available when 'Relaxed UAL' enabled

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
FUITOD											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.90 FUITOS

Thumb syntax (32-bit)

FUITOS *Sd, Sn* [1]

Arm syntax

FUITOS *Sd, Sn* [1]

Notes

[1] only available when 'Relaxed UAL' enabled

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
FUITOS											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.91 HLT

Description

Halt

Thumb syntax (16-bit)

HLT #0...63

Arm syntax

HLT *Rd*, #0...65535

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
HLT														•	•	•	•	•	•						

- Arm and Thumb modes

7.1.92 HVC

Description

Hypervisor call

Thumb syntax (32-bit)

HVC #0...65535

Arm syntax

HVC *Rd*, #0...65535

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
HVC											•			•	•	•	•	•	•						

- Arm and Thumb modes

7.1.93 ISB

Description

Instruction synchronization barrier

Thumb syntax (32-bit)

ISB #0...15

Arm syntax

ISB #0...15

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
ISB											●	●	●	●	●	●	●	●	●	●	●	●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.95 ITE

Description

If-Then-Else

Thumb syntax (16-bit)

ITE *cond* [1]

Notes

[1] *cond* is one of EQ, NE, CS, HS, CC, LO, MI, PL, VS, VC, HI, LS, GE, LT, GT, LE, AL

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
ITE										●	●	●	●	●	●	●	●	●	●			●	●	●	●

● Thumb mode only

7.1.98 ITEET

Description
If-Then-Else-Else-Then

Thumb syntax (16-bit)

ITEET cond [1]

Notes
[1] cond is one of EQ, NE, CS, HS, CC, LO, MI, PL, VS, VC, HI, LS, GE, LT, GT, LE, AL

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
ITEET										●	●	●	●	●	●	●	●	●	●			●	●	●	●

● Thumb mode only

7.1.105 ITTET

Description

If-Then-Then-Else-Then

Thumb syntax (16-bit)

ITTET *cond* [1]

Notes

[1] *cond* is one of EQ, NE, CS, HS, CC, LO, MI, PL, VS, VC, HI, LS, GE, LT, GT, LE, AL

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
ITTET										●	●	●	●	●	●	●	●	●	●			●	●	●	●

● Thumb mode only

7.1.109 LDA

Description

Load-acquire word

Thumb syntax (32-bit)

LDA *Rd*, [*Rn*] [1]

Arm syntax

LDA *Rd*, [*Rn*] [1]

Notes

[1] *Rd*, *Rn* ≠ PC; *Rd* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
LDA														●	●	●	●	●	●					●	●

- Thumb mode only
- Arm and Thumb modes

7.1.110 LDAB

Description

Load-acquire byte

Thumb syntax (32-bit)

LDAB *Rd*, [*Rn*] [1]

Arm syntax

LDAB *Rd*, [*Rn*] [1]

Notes

[1] *Rd*, *Rn* ≠ PC; *Rd* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
LDAB														●	●	●	●	●	●					●	●

- Thumb mode only
- Arm and Thumb modes

7.1.112 LDAEXB

Description

Load-acquire byte exclusive

Thumb syntax (32-bit)

LDAEXB *Rd*, [*Rn*] [1]

Arm syntax

LDAEXB *Rd*, [*Rn*] [1]

Notes

[1] *Rd*, *Rn* ≠ PC; *Rd* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
LDAEXB														●	●	●	●	●	●					●	●

- Thumb mode only
- Arm and Thumb modes

7.1.113 LDAEXD

Description

Load-acquire doubleword exclusive

Thumb syntax (32-bit)

LDAEXD *Rd1*, *Rd2*, [*Rn*] [1]

Arm syntax

LDAEXD *Rd1*, *Rd2*, [*Rn*] [2]

Notes

[1] *Rd1*, *Rd2*, *Rn* ≠ PC; *Rd1* ≠ *Rd2*

[2] *Rd1*, *Rd2*, *Rn* ≠ PC; *Rd1* ≠ *Rd2*; *Rd1* must be an even-numbered register; *Rd2* must be the following odd-numbered register

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
LDAEXD														•	•	•	•	•	•						

- Arm and Thumb modes

7.1.115 LDAH

Description

Load-acquire halfword

Thumb syntax (32-bit)

LDAH *Rd*, [*Rn*] [1]

Arm syntax

LDAH *Rd*, [*Rn*] [1]

Notes

[1] *Rd*, *Rn* ≠ PC; *Rd* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
LDAH														●	●	●	●	●	●					●	●

- Thumb mode only
- Arm and Thumb modes

7.1.116 LDM

Description

Load multiple

Thumb syntax (16-bit)

LDM	SP!, {Rn, Rm...}	
LDM	Rd, {Rn, Rm...}	[1]
LDM	Rd, {Rn, Rm...}	[2]
LDM	Rd!, {Rn, Rm...}	[1]

Thumb syntax (32-bit)

LDM	Rd, {Rn, Rm...}	[2]
LDM	Rd!, {Rn, Rm...}	[2]

Arm syntax

LDM	Rd, {Rn, Rm...}	[2]
LDM	Rd!, {Rn, Rm...}	[2]

Notes

[1] *Rd* must be R0...R7; *Rd* ≠ PC

[2] *Rd* ≠ PC

Availability

Instruction	8-M.Main	●
	8-M.Base	●
	7E-M	●
	7-M	●
	6S-M	●
	6-M	●
	8-R	●
	8.4-A	●
	8.3-A	●
	8.2-A	●
	8.1-A	●
	8-A	●
	7-A	●
	7-R	●
7VE	●	
Instruction	6T2	●
	6ZK	●
	6Z	●
	6KZ	●
	6K	●
	6J	●
Instruction	6	●
	5TE	●
	5T	●
	4T	●
Instruction	LDM	

● Thumb mode only

● Arm and Thumb modes

7.1.117 LDMDA

Description

Load multiple decrement after

Thumb syntax (16-bit)

LDMDA *Rd*, {*Rn*, *Rm*...}

Thumb syntax (32-bit)

LDMDA *Rd*, {*Rn*, *Rm*...}

LDMDA *Rd*!, {*Rn*, *Rm*...} [1]

Arm syntax

LDMDA *Rd*, {*Rn*, *Rm*...} [1]

LDMDA *Rd*!, {*Rn*, *Rm*...} [1]

Notes

[1] *Rd* ≠ PC

Availability

Instruction	8-M.Main	●
	8-M.Base	●
	7E-M	●
	7-M	●
	6S-M	●
	6-M	●
	8-R	●
	8.4-A	●
	8.3-A	●
	8.2-A	●
	8.1-A	●
	8-A	●
	7-A	●
7-R	●	
7VE	●	
6T2	●	
6ZK	●	
6Z	●	
6KZ	●	
6K	●	
6J	●	
6	●	
5TE	●	
5T	●	
4T	●	
LDMDA		

● Thumb mode only

● Arm and Thumb modes

7.1.118 LDMDB

Description

Load multiple decrement before

Thumb syntax (32-bit)

LDMDB *Rd*, {*Rn*, *Rm*...} [1]
 LDMDB *Rd*!, {*Rn*, *Rm*...} [1]

Arm syntax

LDMDB *Rd*, {*Rn*, *Rm*...} [1]
 LDMDB *Rd*!, {*Rn*, *Rm*...} [1]

Notes

[1] *Rd* ≠ PC

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
LDMDB	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●	●		●

- Arm mode only
- Thumb mode only
- Arm and Thumb modes

7.1.119 LDMEA

Description

Load multiple empty ascending

Thumb syntax (32-bit)

LDMEA *Rd*, {*Rn*, *Rm*...} [1]
 LDMEA *Rd*!, {*Rn*, *Rm*...} [1]

Arm syntax

LDMEA *Rd*, {*Rn*, *Rm*...} [1]
 LDMEA *Rd*!, {*Rn*, *Rm*...} [1]

Notes

[1] *Rd* ≠ PC

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
LDMEA	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●	●		●

- Arm mode only
- Thumb mode only
- Arm and Thumb modes

7.1.120 LDMED

Description

Load multiple empty descending

Thumb syntax (16-bit)

LDMED *Rd*, {*Rn*, *Rm*...}

Thumb syntax (32-bit)

LDMED *Rd*, {*Rn*, *Rm*...}

LDMED *Rd*!, {*Rn*, *Rm*...} [1]

Arm syntax

LDMED *Rd*, {*Rn*, *Rm*...} [1]

LDMED *Rd*!, {*Rn*, *Rm*...} [1]

Notes

[1] *Rd* ≠ PC

Availability

Instruction	8-M.Main	●
	8-M.Base	●
	7E-M	●
	7-M	●
	6S-M	●
	6-M	●
	8-R	●
	8.4-A	●
	8.3-A	●
	8.2-A	●
	8.1-A	●
	8-A	●
	7-A	●
7-R	●	
7VE	●	
6T2	●	
6ZK	●	
6Z	●	
6KZ	●	
6K	●	
6J	●	
6	●	
5TE	●	
5T	●	
4T	●	
LDMED		

● Thumb mode only

● Arm and Thumb modes

7.1.121 LDMFA

Description

Load multiple full ascending

Thumb syntax (16-bit)

LDMFA *Rd*, {*Rn*, *Rm*...}

Thumb syntax (32-bit)

LDMFA *Rd*, {*Rn*, *Rm*...}

LDMFA *Rd*!, {*Rn*, *Rm*...} [1]

Arm syntax

LDMFA *Rd*, {*Rn*, *Rm*...} [1]

LDMFA *Rd*!, {*Rn*, *Rm*...} [1]

Notes

[1] *Rd* ≠ PC

Availability

Instruction	8-M.Main	●
	8-M.Base	●
	7E-M	●
	7-M	●
	6S-M	●
	6-M	●
	8-R	●
	8.4-A	●
	8.3-A	●
	8.2-A	●
	8.1-A	●
	8-A	●
	7-A	●
	7-R	●
7VE	●	
6T2	●	
6ZK	●	
6Z	●	
6KZ	●	
6K	●	
6J	●	
6	●	
5TE	●	
5T	●	
4T	●	
LDMFA		

● Thumb mode only

● Arm and Thumb modes

7.1.122 LDMFD

Description

Load multiple full descending

Thumb syntax (16-bit)

LDMFD	SP!, {Rn, Rm...}	
LDMFD	Rd, {Rn, Rm...}	[1]
LDMFD	Rd, {Rn, Rm...}	[2]
LDMFD	Rd!, {Rn, Rm...}	[1]

Thumb syntax (32-bit)

LDMFD	Rd, {Rn, Rm...}	[2]
LDMFD	Rd!, {Rn, Rm...}	[2]

Arm syntax

LDMFD	Rd, {Rn, Rm...}	[2]
LDMFD	Rd!, {Rn, Rm...}	[2]

Notes

[1] *Rd* must be R0...R7; *Rd* ≠ PC

[2] *Rd* ≠ PC

Availability

Instruction	8-M.Main	●
	8-M.Base	●
	7E-M	●
	7-M	●
	6S-M	●
	6-M	●
	8-R	●
	8.4-A	●
	8.3-A	●
	8.2-A	●
	8.1-A	●
	8-A	●
	7-A	●
7-R	●	
7VE	●	
6T2	●	
6ZK	●	
6Z	●	
6KZ	●	
6K	●	
6J	●	
6	●	
5TE	●	
5T	●	
4T	●	
LDMFD		

● Thumb mode only

● Arm and Thumb modes

7.1.123 LDMIA

Description

Load multiple increment after

Thumb syntax (16-bit)

```
LDMIA    SP!, {Rn, Rm...}
LDMIA    Rd, {Rn, Rm...}      [1]
LDMIA    Rd, {Rn, Rm...}      [2]
LDMIA    Rd!, {Rn, Rm...}     [1]
```

Thumb syntax (32-bit)

```
LDMIA    Rd, {Rn, Rm...}      [2]
LDMIA    Rd!, {Rn, Rm...}     [2]
```

Arm syntax

```
LDMIA    Rd, {Rn, Rm...}      [2]
LDMIA    Rd!, {Rn, Rm...}     [2]
```

Notes

[1] *Rd* must be R0...R7; *Rd* ≠ PC

[2] *Rd* ≠ PC

Availability

Instruction	8-M.Main	●
	8-M.Base	●
	7E-M	●
	7-M	●
	6S-M	●
	6-M	●
	8-R	●
	8.4-A	●
	8.3-A	●
	8.2-A	●
	8.1-A	●
	8-A	●
	7-A	●
7-R	●	
7VE	●	
6T2	●	
6ZK	●	
6Z	●	
6KZ	●	
6K	●	
6J	●	
6	●	
5TE	●	
5T	●	
4T	●	
LDMIA		

● Thumb mode only

● Arm and Thumb modes

7.1.124 LDMIB

Description

Load multiple increment before

Thumb syntax (16-bit)

LDMIB *Rd*, {*Rn*, *Rm*...}

Thumb syntax (32-bit)

LDMIB *Rd*, {*Rn*, *Rm*...}

LDMIB *Rd*!, {*Rn*, *Rm*...} [1]

Arm syntax

LDMIB *Rd*, {*Rn*, *Rm*...} [1]

LDMIB *Rd*!, {*Rn*, *Rm*...} [1]

Notes

[1] *Rd* ≠ PC

Availability

Instruction	8-M.Main	●
	8-M.Base	●
	7E-M	●
	7-M	●
	6S-M	●
	6-M	●
	8-R	●
	8.4-A	●
	8.3-A	●
	8.2-A	●
	8.1-A	●
	8-A	●
	7-A	●
7-R	●	
7VE	●	
6T2	●	
6ZK	●	
6Z	●	
6KZ	●	
6K	●	
6J	●	
6	●	
5TE	●	
5T	●	
4T	●	
LDMIB		

● Thumb mode only

● Arm and Thumb modes

7.1.125 LDR

Description

Load word

Thumb syntax (16-bit)

LDR	<i>Rd</i> , [<i>Rn</i> , #0...124]	[1]
LDR	<i>Rd</i> , [SP, #0...1020]	[2]
LDR	<i>Rd</i> , [PC, #0...1020]	[2]
LDR	<i>Rd</i> , [<i>Rn</i> , <i>Rm</i>]	[3]
LDR	<i>Rd</i> , # <i>Label</i>	

Thumb syntax (32-bit)

LDR	<i>Rd</i> , [<i>Rn</i> , #-255...4095]	
LDR	<i>Rd</i> , [PC, #-4095...4095]	
LDR	<i>Rd</i> , [<i>Rn</i> , #-255...255]!	[4]
LDR	<i>Rd</i> , [<i>Rn</i>], #-255...255	[4]
LDR	<i>Rd</i> , [<i>Rn</i> , <i>Rm</i>]	[5]
LDR	<i>Rd</i> , [<i>Rn</i> , <i>Rm</i> , LSL #1...3]	[6]
LDR	<i>Rd</i> , # <i>Label</i>	

Arm syntax

LDR	<i>Rd</i> , [<i>Rn</i> , #-4095...4095]	
LDR	<i>Rd</i> , [<i>Rn</i> , #-4095...4095]!	[4]
LDR	<i>Rd</i> , [<i>Rn</i>], #-4095...4095	[4]
LDR	<i>Rd</i> , [<i>Rn</i>], ± <i>Rm</i> , <i>shift</i>	[7]
LDR	<i>Rd</i> , [<i>Rn</i> , ± <i>Rm</i>]	[8]
LDR	<i>Rd</i> , [<i>Rn</i> , ± <i>Rm</i>]!	[8]

Notes

- [1] *Rd*, *Rn* must be R0...R7; offset a multiple of 4
- [2] offset a multiple of 4
- [3] *Rd*, *Rn*, *Rm* must be R0...R7
- [4] *Rn* ≠ PC; *Rd* ≠ *Rn*
- [5] *Rn*, *Rm* ≠ PC; *Rm* ≠ SP
- [6] *Rn*, *Rm* ≠ PC; *Rm* ≠ SP; LSL #0 is also permitted
- [7] *Rn*, *Rm* ≠ PC; *Rm* ≠ SP; *Rd* ≠ *Rn*
- [8] *Rm* ≠ PC; *Rm* ≠ SP

Availability

Instruction	8-M.Main	●
	8-M.Base	●
	7E-M	●
	7-M	●
	6S-M	●
	6-M	●
	8-R	●
	8.4-A	●
	8.3-A	●
	8.2-A	●
	8.1-A	●
	8-A	●
	7-A	●
	7-R	●
7VE	●	
6T2	●	
6ZK	●	
6Z	●	
6KZ	●	
6K	●	
6J	●	
6	●	
5TE	●	
5T	●	
4T	●	

LDR	●
-----	---

- Thumb mode only
- Arm and Thumb modes

7.1.126 LDRB

Description

Load byte

Thumb syntax (16-bit)

LDRB *Rd*, [*Rn*, #0...31] [1]
 LDRB *Rd*, [*Rn*, *Rm*] [2]

Thumb syntax (32-bit)

LDRB *Rd*, [*Rn*, #-255...4095] [3]
 LDRB *Rd*, [PC, #-4095...4095] [3]
 LDRB *Rd*, [*Rn*, #-255...255]! [4]
 LDRB *Rd*, [*Rn*], #-255...255 [4]
 LDRB *Rd*, [*Rn*, *Rm*, LSL #1...3] [5]

Arm syntax

LDRB *Rd*, [*Rn*, #-4095...4095] [3]
 LDRB *Rd*, [*Rn*, #-4095...4095]! [4]
 LDRB *Rd*, [*Rn*], #-4095...4095 [4]
 LDRB *Rd*, [*Rn*], ±*Rm*, *shift* [6]
 LDRB *Rd*, [*Rn*, ±*Rm*] [7]
 LDRB *Rd*, [*Rn*, ±*Rm*]! [6]

Notes

- [1] *Rd*, *Rn* must be R0...R7; *Rd* ≠ PC; *Rd* ≠ SP
- [2] *Rd*, *Rn*, *Rm* must be R0...R7
- [3] *Rd* ≠ PC; *Rd* ≠ SP
- [4] *Rd*, *Rn* ≠ PC; *Rd* ≠ SP; *Rd* ≠ *Rn*
- [5] *Rd*, *Rn*, *Rm* ≠ PC; *Rd*, *Rm* ≠ SP; LSL #0 is also permitted
- [6] *Rd*, *Rn*, *Rm* ≠ PC; *Rd*, *Rm* ≠ SP; *Rd* ≠ *Rn*
- [7] *Rd*, *Rm* ≠ PC; *Rd*, *Rm* ≠ SP

Availability

Instruction	8-M.Main	●
	8-M.Base	●
	7E-M	●
	7-M	●
	6S-M	●
	6-M	●
	8-R	●
	8.4-A	●
	8.3-A	●
	8.2-A	●
	8.1-A	●
	8-A	●
	7-A	●
	7-R	●
7VE	●	
6T2	●	
6ZK	●	
6Z	●	
6KZ	●	
6K	●	
6J	●	
6	●	
5TE	●	
5T	●	
4T	●	
LDRB		

- Thumb mode only
- Arm and Thumb modes

7.1.127 LDRBT

Description

Load byte, user mode

Thumb syntax (32-bit)

LDRBT *Rd*, [*Rn*, #0...255] [1]

Arm syntax

LDRBT *Rd*, [*Rn*], #-4095...4095 [2]

LDRBT *Rd*, [*Rn*], ±*Rm*, *shift* [3]

Notes

[1] *Rd*, *Rn* ≠ PC; *Rd* ≠ SP

[2] *Rd*, *Rn* ≠ PC; *Rd* ≠ SP; *Rd* ≠ *Rn*

[3] *Rd*, *Rn*, *Rm* ≠ PC; *Rd*, *Rm* ≠ SP; *Rd* ≠ *Rn*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
LDRBT	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●	●		●

● Arm mode only

● Thumb mode only

● Arm and Thumb modes

7.1.128 LDRD

Description

Load doubleword

Thumb syntax (32-bit)

```
LDRD    Rd1, Rd2, [Rn, #-1020...1020]    [1]
LDRD    Rd1, Rd2, [Rn, #-1020...1020]!    [2]
LDRD    Rd1, Rd2, [Rn], #-1020...1020    [2]
```

Arm syntax

```
LDRD    Rd1, Rd2, [Rn, #-255...255]        [3]
LDRD    Rd1, Rd2, [Rn, #-255...255]!        [4]
LDRD    Rd1, Rd2, [Rn], #-255...255        [4]
LDRD    Rd1, Rd2, [Rn, ±Rm]                [5]
LDRD    Rd1, Rd2, [Rn, ±Rm]!                [6]
LDRD    Rd1, Rd2, [Rn], ±Rm                [6]
```

Notes

- [1] $Rd1, Rd2 \neq PC$; $Rd1, Rd2 \neq SP$; $Rd1 \neq Rd2$; offset a multiple of 4
- [2] $Rd1, Rd2, Rn \neq PC$; $Rd1, Rd2 \neq SP$; $Rd1 \neq Rd2 \neq Rn$; offset a multiple of 4
- [3] $Rd1, Rd2 \neq PC$; $Rd2 \neq SP$; $Rd1 \neq Rd2$; $Rd1$ must be an even-numbered register; $Rd2$ must be the following odd-numbered register
- [4] $Rd1, Rd2, Rn \neq PC$; $Rd2 \neq SP$; $Rd1 \neq Rd2 \neq Rn$; $Rd1$ must be an even-numbered register; $Rd2$ must be the following odd-numbered register
- [5] $Rd1, Rd2, Rm \neq PC$; $Rd2, Rm \neq SP$; $Rd1 \neq Rd2 \neq Rm$; $Rd1$ must be an even-numbered register; $Rd2$ must be the following odd-numbered register
- [6] $Rd1, Rd2, Rn, Rm \neq PC$; $Rd2, Rm \neq SP$; $Rd1 \neq Rd2 \neq Rn$; $Rd1 \neq Rd2 \neq Rm$; $Rd1$ must be an even-numbered register; $Rd2$ must be the following odd-numbered register

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
LDRD			•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•			•	•		•

- Arm mode only
- Thumb mode only
- Arm and Thumb modes

7.1.129 LDREX

Description

Load word exclusive

Thumb syntax (32-bit)

LDREX *Rd*, [*Rn*, #0...1020] [1]

Arm syntax

LDREX *Rd*, [*Rn*] [2]

Notes

[1] *Rd*, *Rn* ≠ PC; *Rd* ≠ SP; offset a multiple of 4

[2] *Rd*, *Rn* ≠ PC; *Rd* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
LDREX				●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●	●	●	●

● Arm mode only

● Thumb mode only

● Arm and Thumb modes

7.1.130 LDREXB

Description

Load byte exclusive

Thumb syntax (32-bit)

LDREXB *Rd*, [*Rn*] [1]

Arm syntax

LDREXB *Rd*, [*Rn*] [1]

Notes

[1] *Rd*, *Rn* ≠ PC; *Rd* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
LDREXB				●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●	●	●	●

- Arm mode only
- Thumb mode only
- Arm and Thumb modes

7.1.131 LDREXD

Description

Load doubleword exclusive

Thumb syntax (32-bit)

LDREXD *Rd1*, *Rd2*, [*Rn*] [1]

Arm syntax

LDREXD *Rd1*, *Rd2*, [*Rn*] [2]

Notes

- [1] *Rd1*, *Rd2*, *Rn* ≠ PC; *Rd1*, *Rd2* ≠ SP; *Rd1* ≠ *Rd2*
[2] *Rd1*, *Rd2*, *Rn* ≠ PC; *Rd1*, *Rd2* ≠ SP; *Rd1* ≠ *Rd2*; *Rd1* must be an even-numbered register; *Rd2* must be the following odd-numbered register

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
LDREXD											•	•	•	•	•	•	•	•	•						

- Arm and Thumb modes

7.1.132 LDREXH

Description

Load halfword exclusive

Thumb syntax (32-bit)

LDREXH *Rd*, [*Rn*] [1]

Arm syntax

LDREXH *Rd*, [*Rn*] [1]

Notes

[1] *Rd*, *Rn* ≠ PC; *Rd* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
LDREXH				●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●	●	●	●

- Arm mode only
- Thumb mode only
- Arm and Thumb modes

7.1.133 LDRH

Description

Load halfword

Thumb syntax (16-bit)

LDRH *Rd*, [*Rn*, #0...62] [1]
 LDRH *Rd*, [*Rn*, *Rm*] [2]

Thumb syntax (32-bit)

LDRH *Rd*, [*Rn*, #-255...4095] [3]
 LDRH *Rd*, [PC, #-4095...4095] [4]
 LDRH *Rd*, [*Rn*, #-255...255]! [5]
 LDRH *Rd*, [*Rn*], #-255...255 [5]
 LDRH *Rd*, [*Rn*, *Rm*] [6]
 LDRH *Rd*, [*Rn*, *Rm*, LSL #1...3] [7]

Arm syntax

LDRH *Rd*, [*Rn*, #-255...255] [4]
 LDRH *Rd*, [*Rn*, #-255...255]! [5]
 LDRH *Rd*, [*Rn*], #-255...255 [5]
 LDRH *Rd*, [*Rn*, ±*Rm*] [8]
 LDRH *Rd*, [*Rn*, ±*Rm*]! [9]
 LDRH *Rd*, [*Rn*], ±*Rm* [9]

Notes

- [1] *Rd*, *Rn* must be R0...R7; offset a multiple of 2
- [2] *Rd*, *Rn*, *Rm* must be R0...R7
- [3] *Rd*, *Rn* ≠ PC; *Rd* ≠ SP
- [4] *Rd* ≠ PC; *Rd* ≠ SP
- [5] *Rd*, *Rn* ≠ PC; *Rd* ≠ SP; *Rd* ≠ *Rn*
- [6] *Rd*, *Rn*, *Rm* ≠ PC; *Rd*, *Rm* ≠ SP
- [7] *Rd*, *Rn*, *Rm* ≠ PC; *Rd*, *Rm* ≠ SP; LSL #0 is also permitted
- [8] *Rd*, *Rm* ≠ PC; *Rd*, *Rm* ≠ SP
- [9] *Rd*, *Rn*, *Rm* ≠ PC; *Rd*, *Rm* ≠ SP; *Rd* ≠ *Rn*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
LDRH	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.134 LDRHT

Description

Load halfword, user mode

Thumb syntax (32-bit)

LDRHT *Rd*, [*Rn*, #0...255] [1]

Arm syntax

LDRHT *Rd*, [*Rn*], #-255...255 [2]

LDRHT *Rd*, [*Rn*], ±*Rm* [3]

Notes

[1] *Rd*, *Rn* ≠ PC; *Rd* ≠ SP

[2] *Rd*, *Rn* ≠ PC; *Rd* ≠ SP; *Rd* ≠ *Rn*

[3] *Rd*, *Rn*, *Rm* ≠ PC; *Rd*, *Rm* ≠ SP; *Rd* ≠ *Rn*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
LDRHT										•	•	•	•	•	•	•	•	•	•			•	•		•

• Thumb mode only

• Arm and Thumb modes

7.1.135 LDRSB

Description

Load sign-extended byte

Thumb syntax (16-bit)

LDRSB *Rd*, [*Rn*, *Rm*] [1]

Thumb syntax (32-bit)

LDRSB *Rd*, [*Rn*, #-255...4095] [2]

LDRSB *Rd*, [*PC*, #-4095...4095] [3]

LDRSB *Rd*, [*Rn*, #-255...255]! [4]

LDRSB *Rd*, [*Rn*], #-255...255 [4]

LDRSB *Rd*, [*Rn*, *Rm*, LSL #1...3] [5]

Arm syntax

LDRSB *Rd*, [*Rn*, #-255...255] [3]

LDRSB *Rd*, [*Rn*, #-255...255]! [4]

LDRSB *Rd*, [*Rn*], #-255...255 [4]

LDRSB *Rd*, [*Rn*, ±*Rm*] [6]

LDRSB *Rd*, [*Rn*, ±*Rm*]! [7]

LDRSB *Rd*, [*Rn*], ±*Rm* [7]

Notes

[1] *Rd*, *Rn*, *Rm* must be R0...R7

[2] *Rd*, *Rn* ≠ PC; *Rd* ≠ SP

[3] *Rd* ≠ PC; *Rd* ≠ SP

[4] *Rd*, *Rn* ≠ PC; *Rd* ≠ SP; *Rd* ≠ *Rn*

[5] *Rd*, *Rn*, *Rm* ≠ PC; *Rd*, *Rm* ≠ SP; LSL #0 is also permitted

[6] *Rd*, *Rm* ≠ PC; *Rd*, *Rm* ≠ SP

[7] *Rd*, *Rn*, *Rm* ≠ PC; *Rd*, *Rm* ≠ SP; *Rd* ≠ *Rn*

Availability

Instruction	8-M.Main	●
	8-M.Base	●
	7E-M	●
	7-M	●
	6S-M	●
	6-M	●
	8-R	●
	8.4-A	●
	8.3-A	●
	8.2-A	●
	8.1-A	●
	8-A	●
	7-A	●
	7-R	●
7VE	●	
6T2	●	
6ZK	●	
6Z	●	
6KZ	●	
6K	●	
6J	●	
6	●	
5TE	●	
5T	●	
4T	●	
LDRSB		

● Thumb mode only

● Arm and Thumb modes

7.1.136 LDRSBT

Description

Load sign-extended byte, user mode

Thumb syntax (32-bit)

LDRSBT *Rd*, [*Rn*, #0...255] [1]

Arm syntax

LDRSBT *Rd*, [*Rn*], #-255...255 [2]

LDRSBT *Rd*, [*Rn*], ±*Rm* [3]

Notes

[1] *Rd*, *Rn* ≠ PC; *Rd* ≠ SP

[2] *Rd*, *Rn* ≠ PC; *Rd* ≠ SP; *Rd* ≠ *Rn*

[3] *Rd*, *Rn*, *Rm* ≠ PC; *Rd*, *Rm* ≠ SP; *Rd* ≠ *Rn*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
LDRSBT										•	•	•	•	•	•	•	•	•	•			•	•		•

• Thumb mode only

• Arm and Thumb modes

7.1.137 LDRSH

Description

Load sign-extended halfword

Thumb syntax (16-bit)

LDRSH *Rd*, [*Rn*, *Rm*] [1]

Thumb syntax (32-bit)

LDRSH *Rd*, [*Rn*, #-255...4095] [2]

LDRSH *Rd*, [*PC*, #-4095...4095] [3]

LDRSH *Rd*, [*Rn*, #-255...255]! [2]

LDRSH *Rd*, [*Rn*], #-255...255 [2]

LDRSH *Rd*, [*Rn*, *Rm*, LSL #1...3] [4]

Arm syntax

LDRSH *Rd*, [*Rn*, #-255...255] [3]

LDRSH *Rd*, [*Rn*, #-255...255]! [2]

LDRSH *Rd*, [*Rn*], #-255...255 [2]

LDRSH *Rd*, [*Rn*, ±*Rm*] [5]

LDRSH *Rd*, [*Rn*, ±*Rm*]! [6]

LDRSH *Rd*, [*Rn*], ±*Rm* [6]

Notes

[1] *Rd*, *Rn*, *Rm* must be R0...R7

[2] *Rd*, *Rn* ≠ PC; *Rd* ≠ SP; *Rd* ≠ *Rn*

[3] *Rd* ≠ PC; *Rd* ≠ SP

[4] *Rd*, *Rn*, *Rm* ≠ PC; *Rd*, *Rm* ≠ SP; LSL #0 is also permitted

[5] *Rd*, *Rm* ≠ PC; *Rd*, *Rm* ≠ SP

[6] *Rd*, *Rn*, *Rm* ≠ PC; *Rd*, *Rm* ≠ SP; *Rd* ≠ *Rn*

Availability

Instruction	8-M.Main	●
	8-M.Base	●
	7E-M	●
	7-M	●
	6S-M	●
	6-M	●
	8-R	●
	8.4-A	●
	8.3-A	●
	8.2-A	●
	8.1-A	●
	8-A	●
	7-A	●
7-R	●	
7VE	●	
6T2	●	
6ZK	●	
6Z	●	
6KZ	●	
6K	●	
6J	●	
6	●	
5TE	●	
5T	●	
4T	●	
LDRSH		

● Thumb mode only

● Arm and Thumb modes

7.1.138 LDRSHT

Description

Load sign-extended halfword, user mode

Thumb syntax (32-bit)

LDRSHT *Rd*, [*Rn*, #0...255] [1]

Arm syntax

LDRSHT *Rd*, [*Rn*], #-255...255 [2]

LDRSHT *Rd*, [*Rn*], ±*Rm* [3]

Notes

[1] *Rd*, *Rn* ≠ PC; *Rd* ≠ SP

[2] *Rd*, *Rn* ≠ PC; *Rd* ≠ SP; *Rd* ≠ *Rn*

[3] *Rd*, *Rn*, *Rm* ≠ PC; *Rd*, *Rm* ≠ SP; *Rd* ≠ *Rn*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
LDRSHT										•	•	•	•	•	•	•	•	•	•			•	•		•

• Thumb mode only

• Arm and Thumb modes

7.1.139 LDRT

Description

Load word, user mode

Thumb syntax (32-bit)

LDRT *Rd*, [*Rn*, #0...255] [1]

Arm syntax

LDRT *Rd*, [*Rn*], #-4095...4095 [2]

LDRT *Rd*, [*Rn*], ±*Rm*, *shift* [3]

Notes

[1] *Rd*, *Rn* ≠ PC; *Rd* ≠ SP

[2] *Rd*, *Rn* ≠ PC; *Rd* ≠ SP; *Rd* ≠ *Rn*

[3] *Rd*, *Rn*, *Rm* ≠ PC; *Rd*, *Rm* ≠ SP; *Rd* ≠ *Rn*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
LDRT	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●	●		●

● Arm mode only

● Thumb mode only

● Arm and Thumb modes

7.1.140 LSL

Description

Logical shift left

Thumb syntax (16-bit)

LSLS *Rd, Rn* [1]
 LSLS *Rd, Rn, #0...31* [1]

Thumb syntax (32-bit)

LSL *Rd, Rn, Rm* [2]
 LSL *Rd, Rn, #0...31* [3]
 LSLS *Rd, Rn, Rm* [2]
 LSLS *Rd, Rn, #0...31* [3]

Arm syntax

LSL *Rd, Rn, Rm* [2]
 LSL *Rd, Rn, #0...31* [3]
 LSLS *Rd, Rn, Rm* [2]
 LSLS *Rd, Rn, #0...31* [3]

Notes

[1] *Rd, Rn* must be **R0...R7**; **LSL** permitted in an IT block

[2] *Rd, Rn, Rm* ≠ **PC**; *Rd, Rn, Rm* ≠ **SP**

[3] *Rd, Rn* ≠ **PC**; *Rd, Rn* ≠ **SP**

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
LSL	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●

● Thumb mode only

● Arm and Thumb modes

7.1.141 LSR

Description

Logical shift right

Thumb syntax (16-bit)

LSRS *Rd, Rn* [1]
 LSRS *Rd, Rn, #1...32* [1]

Thumb syntax (32-bit)

LSR *Rd, Rn, Rm* [2]
 LSR *Rd, Rn, #1...32* [3]
 LSRS *Rd, Rn, Rm* [2]
 LSRS *Rd, Rn, #1...32* [3]

Arm syntax

LSR *Rd, Rn, Rm* [2]
 LSR *Rd, Rn, #1...32* [3]
 LSRS *Rd, Rn, Rm* [2]
 LSRS *Rd, Rn, #1...32* [3]

Notes

[1] *Rd, Rn* must be **R0...R7**; LSR permitted in an IT block

[2] *Rd, Rn, Rm* ≠ **PC**; *Rd, Rn, Rm* ≠ **SP**

[3] *Rd, Rn* ≠ **PC**; *Rd, Rn* ≠ **SP**

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
LSR	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●

● Thumb mode only

● Arm and Thumb modes

7.1.142 MCR

Description

Move register to coprocessor

Thumb syntax (32-bit)

MCR *Pn*, 0...7, *Rn*, *CRx*, *CRy*{, 0...7} [1]

Arm syntax

MCR *Pn*, 0...7, *Rn*, *CRx*, *CRy*{, 0...7} [1]

Notes

[1] *Rm* ≠ PC; *Rm* ≠ SP; constants may be prefixed by “#”

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
MCR	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●	●		●

- Arm mode only
- Thumb mode only
- Arm and Thumb modes

7.1.143 MCR2

Description

Move register to coprocessor

Thumb syntax (32-bit)

MCR2 *Pn*, 0...7, *Rn*, *CRx*, *CRy*{, 0...7} [1]

Arm syntax

MCR2 *Pn*, 0...7, *Rn*, *CRx*, *CRy*{, 0...7} [1]

Notes

[1] *Rm* ≠ PC; *Rm* ≠ SP; constants may be prefixed by “#”

Availability

Instruction	4T	5T	5TE	6	6J	6K	6K2	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
MCR2		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•			•	•		•

- Arm mode only
- Thumb mode only
- Arm and Thumb modes

7.1.144 MCRR

Description

Move registers to coprocessor

Thumb syntax (32-bit)

MCRR *Pn, 0...15, Rn, Rm, CRx, CRy* [1]

Arm syntax

MCRR *Pn, 0...15, Rn, Rm, CRx, CRy* [1]

Notes

[1] constants may be prefixed by “#”

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
MCRR	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●	●		●

- Arm mode only
- Thumb mode only
- Arm and Thumb modes

7.1.145 MCRR2

Description

Move registers to coprocessor

Thumb syntax (32-bit)

MCRR2 *Pn*, 0...15, *Rn*, *Rm*, *CRx*, *CRy* [1]

Arm syntax

MCRR2 *Pn*, 0...15, *Rn*, *Rm*, *CRx*, *CRy* [1]

Notes

[1] *Rn* ≠ PC; *Rn* ≠ SP; constants may be prefixed by “#”

Availability

Instruction	4T	5T	5TE	6	6J	6K	6K2	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
MCRR2										●	●	●	●	●	●	●	●	●	●			●	●		●

- Thumb mode only
- Arm and Thumb modes

7.1.146 MLA

Description

Multiply and add

Thumb syntax (32-bit)

MLA Rd, Rn, Rm, Ra [1]

Arm syntax

MLA Rd, Rn, Rm, Ra [2]

MLAS Rd, Rn, Rm, Ra [2]

Notes

[1] $Rd, Rn, Rm, Ra \neq PC$; $Rd, Rn, Rm, Ra \neq SP$

[2] $Rd, Rn, Rm, Ra \neq PC$; $Rd \neq Rn$

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
MLA	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●	●		●

● Arm mode only

● Thumb mode only

7.1.147 MLS

Description

Multiply and subtract

Thumb syntax (32-bit)

MLS *Rd, Rn, Rm, Ra* [1]

Arm syntax

MLS *Rd, Rn, Rm, Ra* [1]

Notes

[1] *Rd, Rn, Rm, Ra* ≠ PC; *Rd, Rn, Rm, Ra* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
MLS										●	●	●	●	●	●	●	●	●	●			●	●		●

- Thumb mode only
- Arm and Thumb modes

7.1.148 MOV

Description

Move

Thumb syntax (16-bit)

```
MOV      Rd, Rn
MOVS     Rd, Rn           [1]
MOVS     Rd, #0           [2]
MOVS     Rd, Rn, shift   [3]
```

Thumb syntax (32-bit)

```
MOV      Rd, #0...65535   [4]
MOV      Rd, #const       [5]
MOV      Rd, Rn, shift    [6]
MOVS     Rd, #const       [5]
MOVS     Rd, Rn, shift    [7]
```

Arm syntax

```
MOV      Rd, #0...65535   [4]
MOV      Rd, #const       [8]
MOV      Rd, Rn, shift    [9]
MOVS     Rd, #const       [8]
MOVS     Rd, Rn, shift    [10]
```

Notes

- [1] *Rd, Rn* must be **R0...R7**; MOV permitted in an IT block
- [2] *Rd* must be **R0...R7**; MOV permitted in an IT block
- [3] *shift* is one of **LSL #0...31, LSR #1...32, ASR #1...32, RRX**; MOV permitted in an IT block
- [4] *Rd* ≠ **PC**; *Rd* ≠ **SP**
- [5] *Rd* ≠ **PC**; *Rd* ≠ **SP**; *const* is one of **\$00xy00xy, \$xy00xy00, \$xyxyxyxy, \$xy LSL 0...24**
- [6] *Rd, Rn* ≠ **PC**; *shift* is one of **LSL #0...31, LSR #1...32, ASR #1...32, RRX**
- [7] *Rd, Rn* ≠ **PC**; *Rd, Rn* ≠ **SP**; *shift* is one of **LSL #0...31, LSR #1...32, ASR #1...32, RRX**
- [8] *Rd* ≠ **PC**; *Rd* ≠ **SP**; *const* is **\$xy ROR 2n**
- [9] *shift* is one of **LSL #0...31/RS, LSR #1...32/RS, ASR #1...32/RS, ROR #1...31/RS, RRX**
- [10] *Rd, Rn* ≠ **PC**; *Rd, Rn* ≠ **SP**; *shift* is one of **LSL #0...31/RS, LSR #1...32/RS, ASR #1...32/RS, ROR #1...31/RS, RRX**

Availability

Instruction	8-M.Main	●
	8-M.Base	●
	7E-M	●
	7-M	●
	6S-M	●
	6-M	●
	8-R	●
	8.4-A	●
	8.3-A	●
	8.2-A	●
	8.1-A	●
	8-A	●
	7-A	●
7-R	●	
7VE	●	
6T2	●	
6ZK	●	
6Z	●	
6KZ	●	
6K	●	
6J	●	
6	●	
5TE	●	
5T	●	
4T	●	
MOV		

- Thumb mode only
- Arm and Thumb modes

7.1.149 MOVS

Thumb syntax (16-bit)

MOVS	<i>Rd, Rn</i>	[1]
MOVS	<i>Rd, #0</i>	[2]
MOVS	<i>Rd, Rn, shift</i>	[3]

Thumb syntax (32-bit)

MOVS	<i>Rd, #const</i>	[4]
MOVS	<i>Rd, Rn, shift</i>	[5]

Arm syntax

MOVS	<i>Rd, #const</i>	[6]
MOVS	<i>Rd, Rn, shift</i>	[7]

Notes

- [1] *Rd, Rn* must be **R0...R7**; MOV permitted in an IT block
 [2] *Rd* must be **R0...R7**; MOV permitted in an IT block
 [3] *shift* is one of **LSL #0...31, LSR #1...32, ASR #1...32, RRX**; MOV permitted in an IT block
 [4] *Rd* ≠ **PC**; *Rd* ≠ **SP**; *const* is one of **\$00xy00xy, \$xy00xy00, \$xyxyxyxy, \$xy LSL 0...24**
 [5] *Rd, Rn* ≠ **PC**; *Rd, Rn* ≠ **SP**; *shift* is one of **LSL #0...31, LSR #1...32, ASR #1...32, RRX**
 [6] *Rd* ≠ **PC**; *Rd* ≠ **SP**; *const* is **\$xy ROR 2n**
 [7] *Rd, Rn* ≠ **PC**; *Rd, Rn* ≠ **SP**; *shift* is one of **LSL #0...31/Rs, LSR #1...32/Rs, ASR #1...32/Rs, ROR #1...31/Rs, RRX**

Availability

Instruction	8-M.Main	●
	8-M.Base	●
	7E-M	●
	7-M	●
	6S-M	●
	6-M	●
	8-R	●
	8.4-A	●
	8.3-A	●
	8.2-A	●
	8.1-A	●
	8-A	●
7-A	●	
7-R	●	
7VE	●	
6T2	●	
6ZK	●	
6Z	●	
6KZ	●	
6K	●	
6J	●	
6	●	
5TE	●	
5T	●	
4T	●	
MOVS		

- Thumb mode only
- Arm and Thumb modes

7.1.152 MRC

Description

Move coprocessor to register

Thumb syntax (32-bit)

MRC *Pn*, 0...7, *Rn*, *CRx*, *CRy*{, 0...7} [1]

Arm syntax

MRC *Pn*, 0...7, *Rn*, *CRx*, *CRy*{, 0...7} [2]

Notes

[1] *Rm* ≠ PC; *Rm* ≠ SP; constants may be prefixed by “#”

[2] constants may be prefixed by “#”

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
MRC	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●	●		●

● Arm mode only

● Thumb mode only

● Arm and Thumb modes

7.1.153 MRC2

Description

Move coprocessor to register

Thumb syntax (32-bit)

MRC2 *Pn*, 0...7, *Rn*, *CRx*, *CRy*{, 0...7} [1]

Arm syntax

MRC2 *Pn*, 0...7, *Rn*, *CRx*, *CRy*{, 0...7} [2]

Notes

[1] *Rm* ≠ **SP**; constants may be prefixed by “#”

[2] constants may be prefixed by “#”

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
MRC2		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•			•	•		•

• Arm mode only

• Thumb mode only

• Arm and Thumb modes

7.1.154 MRRC

Description

Move coprocessor to registers

Thumb syntax (32-bit)

MRRC *Pn, 0...15, Rn, Rm, CRx, CRy* [1]

Arm syntax

MRRC *Pn, 0...15, Rn, Rm, CRx, CRy* [1]

Notes

[1] constants may be prefixed by “#”

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
MRRC	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●	●		●

- Arm mode only
- Thumb mode only
- Arm and Thumb modes

7.1.155 MRRC2

Description

Move coprocessor to registers

Thumb syntax (32-bit)

MRRC2 *Pn, 0...15, Rn, Rm, CRx, CRy* [1]

Arm syntax

MRRC2 *Pn, 0...15, Rn, Rm, CRx, CRy* [1]

Notes

[1] *Rn* ≠ PC; *Rn* ≠ SP; constants may be prefixed by “#”

Availability

Instruction	4T	5T	5TE	6	6J	6K	6K2	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
MRRC2										●	●	●	●	●	●	●	●	●	●			●	●		●

- Thumb mode only
- Arm and Thumb modes

7.1.156 MRS

Description

Move system register to register

Thumb syntax (32-bit)

MRS *Rd*, *SysReg* [1]

Arm syntax

MRS *Rd*, *SysReg* [1]

Notes

[1] *Rd* ≠ PC; *Rd* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
MRS	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●	●	●	●

- Arm mode only
- Thumb mode only
- Arm and Thumb modes

7.1.157 MSR

Description

Move register to system register

Thumb syntax (32-bit)

MSR *SysReg*, *Rn* [1]

Arm syntax

MSR *SysReg*, *Rn* [1]

MSR *SysReg*, #0...4095

Notes

[1] *Rn* ≠ PC; *Rn* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
MSR	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●	●	●	●

- Arm mode only
- Thumb mode only
- Arm and Thumb modes

7.1.158 MUL

Description

Multiply

Thumb syntax (16-bit)

MULS *Rd, Rn* [1]

Thumb syntax (32-bit)

MUL *Rd, Rn, Rm* [2]

Arm syntax

MUL *Rd, Rn, Rm* [3]

MULS *Rd, Rn, Rm* [3]

Notes

[1] *Rd, Rn* must be **R0...R7**; ordering *Rd, Rn, Rd* is also permitted; MUL permitted in an IT block

[2] *Rd, Rn, Rm* ≠ **PC**; *Rd, Rn, Rm* ≠ **SP**

[3] *Rd, Rn, Rm* ≠ **PC**; *Rd* ≠ *Rn*

Availability

Instruction	8-M.Main	●
	8-M.Base	●
	7E-M	●
	7-M	●
	6S-M	●
	6-M	●
	8-R	●
	8.4-A	●
	8.3-A	●
	8.2-A	●
	8.1-A	●
	8-A	●
	7-A	●
	7-R	●
7VE	●	
6T2	●	
6ZK	●	
6Z	●	
6KZ	●	
6K	●	
6J	●	
6	●	
5TE	●	
5T	●	
4T	●	
MUL		

● Thumb mode only

● Arm and Thumb modes

7.1.159 MVN

Description

Move not

Thumb syntax (16-bit)

MVNS *Rd, Rn* [1]

Thumb syntax (32-bit)

MVN *Rd, #const* [2]

MVN *Rd, Rn, shift* [3]

MVNS *Rd, #const* [2]

MVNS *Rd, Rn, shift* [3]

Arm syntax

MVN *Rd, #const* [4]

MVN *Rd, Rn, shift* [5]

MVNS *Rd, #const* [4]

MVNS *Rd, Rn, shift* [5]

Notes

[1] *Rd, Rn* must be **R0...R7**; **MVN** permitted in an IT block

[2] *Rd* ≠ **PC**; *Rd* ≠ **SP**; *const* is one of *\$00xy00xy*, *\$xy00xy00*, *\$xyxyxyxy*, *\$xy LSL 0...24*

[3] *Rd, Rn* ≠ **PC**; *Rd, Rn* ≠ **SP**; *shift* is one of **LSL #0...31**, **LSR #1...32**, **ASR #1...32**, **RRX**

[4] *Rd, Rn* ≠ **PC**; *Rd, Rn* ≠ **SP**; *const* is *\$xy ROR 2n*

[5] *Rd, Rn* ≠ **PC**; *Rd, Rn* ≠ **SP**; *shift* is one of **LSL #0...31/Rs**, **LSR #1...32/Rs**, **ASR #1...32/Rs**, **ROR #1...31/Rs**, **RRX**

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
MVN	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●

● Thumb mode only

● Arm and Thumb modes

7.1.160 NEG

Description

Negate

Thumb syntax (16-bit)

NEGS *Rd, Rn* [1]

Thumb syntax (32-bit)

NEG *Rd, Rn* [2]

NEGS *Rd, Rn* [2]

Arm syntax

NEG *Rd, Rn* [2]

NEGS *Rd, Rn* [2]

Notes

[1] *Rd, Rn* must be **R0...R7**; NEG permitted in an IT block

[2] *Rd, Rn* ≠ **PC**; *Rd, Rn* ≠ **SP**

Availability

Instruction	8-M.Main	●
	8-M.Base	●
	7E-M	●
	7-M	●
	6S-M	●
	6-M	●
	8-R	●
	8.4-A	●
	8.3-A	●
	8.2-A	●
	8.1-A	●
	8-A	●
	7-A	●
7-R	●	
7VE	●	
6T2	●	
6ZK	●	
6Z	●	
6KZ	●	
6K	●	
6J	●	
6	●	
5TE	●	
5T	●	
4T	●	
NEG		

● Thumb mode only

● Arm and Thumb modes

7.1.161 NOP

Description

No operation

Thumb syntax (16-bit)

NOP

Thumb syntax (32-bit)

NOP

Arm syntax

NOP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
NOP	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●

- Arm mode only
- Thumb mode only
- Arm and Thumb modes

7.1.162 ORN

Description

Bitwise or-not

Thumb syntax (32-bit)

ORN	<i>Rd, Rn, #const</i>	[1]
ORN	<i>Rd, Rn, Rm, shift</i>	[2]
ORNS	<i>Rd, Rn, #const</i>	[1]
ORNS	<i>Rd, Rn, Rm, shift</i>	[2]

Notes

[1] *Rd, Rn* ≠ PC; *Rd, Rn* ≠ SP; *const* is one of \$00xy00xy, \$xy00xy00, \$xyxyxyxy, \$xy LSL 0...24

[2] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP; *shift* is one of LSL #0...31, LSR #1...32, ASR #1...32, RRX

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
ORN										●	●	●	●	●	●	●	●	●	●			●	●		●

● Thumb mode only

7.1.163 ORR

Description

Bitwise or

Thumb syntax (16-bit)

ORRS *Rd, Rn* [1]

Thumb syntax (32-bit)

ORR *Rd, Rn, #const* [2]

ORR *Rd, Rn, Rm, shift* [3]

ORRS *Rd, Rn, #const* [2]

ORRS *Rd, Rn, Rm, shift* [3]

Arm syntax

ORR *Rd, Rn, #const* [4]

ORR *Rd, Rn, Rm, shift* [5]

ORRS *Rd, Rn, #const* [4]

ORRS *Rd, Rn, Rm, shift* [5]

Notes

[1] *Rd, Rn* must be **R0...R7**; ordering *Rd, Rn, Rd* is also permitted; ORR permitted in an IT block

[2] *Rd, Rn* ≠ **PC**; *Rd, Rn* ≠ **SP**; *const* is one of *\$00xy00xy*, *\$xy00xy00*, *\$xyxyxyxy*, *\$xy LSL #0...24*

[3] *Rd, Rn, Rm* ≠ **PC**; *Rd, Rn, Rm* ≠ **SP**; *shift* is one of **LSL #0...31**, **LSR #1...32**, **ASR #1...32**, **RRX**

[4] *Rd, Rn* ≠ **PC**; *Rd, Rn* ≠ **SP**; *const* is *\$xy ROR #n*

[5] *Rd, Rn, Rm* ≠ **PC**; *Rd, Rn, Rm* ≠ **SP**; *shift* is one of **LSL #0...31/RS**, **LSR #1...32/RS**, **ASR #1...32/RS**, **ROR #1...31/RS**, **RRX**

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
ORR	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●

● Thumb mode only

● Arm and Thumb modes

7.1.164 PKHBT

Description

Pack halfword

Thumb syntax (32-bit)

PKHBT *Rd, Rn, Rm, shift* [1]

Arm syntax

PKHBT *Rd, Rn, Rm, shift* [2]

Notes

[1] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP; *shift* is one of LSL #0...31, LSR #1...32, ASR #1...32, RRX

[2] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP; *shift* is one of LSL #0...31/*Rs*, LSR #1...32/*Rs*, ASR #1...32/*Rs*, ROR #1...31/*Rs*, RRX

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
PKHBT											●	●	●	●	●	●	●	●	●				●		

- Thumb mode only
- Arm and Thumb modes

7.1.165 PKHTB

Description

Pack halfword

Thumb syntax (32-bit)

PKHTB *Rd, Rn, Rm, shift* [1]

Arm syntax

PKHTB *Rd, Rn, Rm, shift* [2]

Notes

[1] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP; *shift* is one of LSL #0...31, LSR #1...32, ASR #1...32, RRX

[2] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP; *shift* is one of LSL #0...31/*Rs*, LSR #1...32/*Rs*, ASR #1...32/*Rs*, ROR #1...31/*Rs*, RRX

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
PKHTB											●	●	●	●	●	●	●	●	●				●		

- Thumb mode only
- Arm and Thumb modes

7.1.166 PLD

Description

Preload data

Thumb syntax (32-bit)

```
PLD      [Rd, #-4095...4095]
PLD      Rd, [Rn, Rm, LSL #1...3]           [1]
```

Arm syntax

```
PLD      [Rd, #-4095...4095]
```

Notes

[1] *Rd* ≠ PC; LSL #0 is also permitted

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
PLD										●	●	●	●	●	●	●	●	●	●			●	●		●

- Thumb mode only
- Arm and Thumb modes

7.1.168 PLI

Description

Preload instruction

Thumb syntax (32-bit)

PLI $[Rd, \#-4095...4095]$

PLI $Rd, [Rn, Rm, LSL \#1...3]$ [1]

Arm syntax

PLI $[Rd, \#-4095...4095]$

Notes

[1] $Rd \neq PC$; $LSL \#0$ is also permitted

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
PLI										●	●	●	●	●	●	●	●	●	●			●	●		●

- Thumb mode only
- Arm and Thumb modes

7.1.169 POP

Description

Pop registers

Thumb syntax (16-bit)

POP $\{Rn, Rm...\}$

Thumb syntax (32-bit)

POP $\{Rn, Rm...\}$

Arm syntax

POP $\{Rn, Rm...\}$

Availability

Instruction	8-M.Main	●
	8-M.Base	●
	7E-M	●
	7-M	●
	6S-M	●
	6-M	●
	8-R	●
	8.4-A	●
	8.3-A	●
	8.2-A	●
	8.1-A	●
	8-A	●
	7-A	●
	7-R	●
	7VE	●
6T2	●	
6ZK	●	
6Z	●	
6KZ	●	
6K	●	
6J	●	
6	●	
5TE	●	
5T	●	
4T	●	
POP		

- Thumb mode only
- Arm and Thumb modes

7.1.170 PUSH

Description

Push registers

Thumb syntax (16-bit)

PUSH {*Rn*, *Rm...*}

Thumb syntax (32-bit)

PUSH {*Rn*, *Rm...*}

Arm syntax

PUSH {*Rn*, *Rm...*}

Availability

Instruction	8-M.Main	●
	8-M.Base	●
	7E-M	●
	7-M	●
	6S-M	●
	6-M	●
	8-R	●
	8.4-A	●
	8.3-A	●
	8.2-A	●
	8.1-A	●
	8-A	●
	7-A	●
	7-R	●
	7VE	●
6T2	●	
6ZK	●	
6Z	●	
6KZ	●	
6K	●	
6J	●	
6	●	
5TE	●	
5T	●	
4T	●	
	PUSH	

- Thumb mode only
- Arm and Thumb modes

- Arm mode only
- Thumb mode only
- Arm and Thumb modes

7.1.172 QADD16

Description

Signed parallel add, 16-bit, saturating

Thumb syntax (32-bit)

QADD16 *Rd, Rn, Rm* [1]

Arm syntax

QADD16 *Rd, Rn, Rm* [1]

Notes

[1] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
QADD16										•	•	•	•	•	•	•	•	•	•				•		

- Thumb mode only
- Arm and Thumb modes

7.1.173 QADD8

Description

Signed parallel add, 8-bit, saturating

Thumb syntax (32-bit)

QADD8 *Rd, Rn, Rm* [1]

Arm syntax

QADD8 *Rd, Rn, Rm* [1]

Notes

[1] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
QADD8										●	●	●	●	●	●	●	●	●	●				●		

- Thumb mode only
- Arm and Thumb modes

7.1.174 QASX

Description

Signed parallel add and subtract with exchange, 16-bit, saturating

Thumb syntax (32-bit)

QASX *Rd, Rn, Rm* [1]

Arm syntax

QASX *Rd, Rn, Rm* [1]

Notes

[1] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
QASX										●	●	●	●	●	●	●	●	●	●				●		

- Thumb mode only
- Arm and Thumb modes

7.1.175 QDADD

Description

Double and add, saturating

Thumb syntax (32-bit)

QDADD *Rd, Rn, Rm* [1]

Arm syntax

QDADD *Rd, Rn, Rm* [1]

Notes

[1] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
QDADD			•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•				•		

- Arm mode only
- Thumb mode only
- Arm and Thumb modes

7.1.176 QDSUB

Description

Double and subtract, saturating

Thumb syntax (32-bit)

QDSUB *Rd, Rn, Rm* [1]

Arm syntax

QDSUB *Rd, Rn, Rm* [1]

Notes

[1] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
QDSUB			•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•				•		

- Arm mode only
- Thumb mode only
- Arm and Thumb modes

7.1.177 QSAX

Description

Signed parallel subtract and add with exchange, 16-bit, saturating

Thumb syntax (32-bit)

QSAX *Rd, Rn, Rm* [1]

Arm syntax

QSAX *Rd, Rn, Rm* [1]

Notes

[1] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
QSAX										●	●	●	●	●	●	●	●	●	●				●		

- Thumb mode only
- Arm and Thumb modes

7.1.180 QSUB8

Description

Signed parallel subtract, 8-bit, saturating

Thumb syntax (32-bit)

QSUB8 *Rd, Rn, Rm* [1]

Arm syntax

QSUB8 *Rd, Rn, Rm* [1]

Notes

[1] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
QSUB8										•	•	•	•	•	•	•	•	•	•				•		

- Thumb mode only
- Arm and Thumb modes

7.1.181 RBIT

Description

Reverse bits

Thumb syntax (32-bit)

RBIT *Rd*, *Rn* [1]

Arm syntax

RBIT *Rd*, *Rn* [1]

Notes

[1] *Rd*, *Rn* ≠ PC; *Rd*, *Rn* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
RBIT										●	●	●	●	●	●	●	●	●	●			●	●		●

- Thumb mode only
- Arm and Thumb modes

7.1.182 REV

Description

Reverse byte order

Thumb syntax (16-bit)

REV *Rd, Rn* [1]

Thumb syntax (32-bit)

REV *Rd, Rn* [1]

Arm syntax

REV *Rd, Rn* [1]

Notes

[1] *Rd, Rn* ≠ PC; *Rd, Rn* ≠ SP; only available when ‘Relaxed UAL’ enabled

Availability

Instruction	8-M.Main	●
	8-M.Base	●
	7E-M	●
	7-M	●
	6S-M	●
	6-M	●
	8-R	●
	8.4-A	●
	8.3-A	●
	8.2-A	●
	8.1-A	●
	8-A	●
	7-A	●
	7-R	●
7VE	●	
6T2	●	
6ZK	●	
6Z	●	
6KZ	●	
6K	●	
6J	●	
6	●	
5TE		
5T		
4T		
REV		

- Thumb mode only
- Arm and Thumb modes

7.1.183 REV16

Description

Reverse halfword byte order

Thumb syntax (16-bit)

REV16 *Rd, Rn* [1]

Thumb syntax (32-bit)

REV16 *Rd, Rn* [1]

Arm syntax

REV16 *Rd, Rn* [1]

Notes

[1] *Rd, Rn* ≠ PC; *Rd, Rn* ≠ SP; only available when 'Relaxed UAL' enabled

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
REV16				●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.186 RFEDA

Description

Return from exception, decrement after

Arm syntax

```
RFEDA    SP                               [1]
RFEDA    SP!                             [1]
```

Notes

[1] *Rd* ≠ PC

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
RFEDA				•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•						

- Arm mode only

7.1.188 RFEIA

Description

Return from exception, increment after

Thumb syntax (32-bit)

RFEIA SP [1]
RFEIA SP! [1]

Arm syntax

RFEIA SP [1]
RFEIA SP! [1]

Notes

[1] *Rd* ≠ PC

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
RFEIA				•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•						

- Arm and Thumb modes

7.1.189 RFEIB

Description

Return from exception, increment before

Arm syntax

```
RFEIB    SP                [1]
RFEIB    SP!               [1]
```

Notes

[1] *Rd* ≠ PC

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
RFEIB				•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•						

- Arm mode only

7.1.190 ROR

Description

Rotate right

Thumb syntax (16-bit)

RORS *Rd, Rn* [1]

Thumb syntax (32-bit)

ROR *Rd, Rn, Rm* [2]

ROR *Rd, Rn, #1...31* [3]

RORS *Rd, Rn, Rm* [2]

RORS *Rd, Rn, #1...31* [3]

Arm syntax

ROR *Rd, Rn, Rm* [2]

ROR *Rd, Rn, #1...31* [3]

RORS *Rd, Rn, Rm* [2]

RORS *Rd, Rn, #1...31* [3]

Notes

[1] *Rd, Rn* must be R0...R7; ROR permitted in an IT block

[2] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP

[3] *Rd, Rn* ≠ PC; *Rd, Rn* ≠ SP

Availability

Instruction	8-M.Main	●
	8-M.Base	●
	7E-M	●
	7-M	●
	6S-M	●
	6-M	●
	8-R	●
	8.4-A	●
	8.3-A	●
	8.2-A	●
	8.1-A	●
	8-A	●
	7-A	●
	7-R	●
7VE	●	
6T2	●	
6ZK	●	
6Z	●	
6KZ	●	
6K	●	
6J	●	
6	●	
5TE	●	
5T	●	
4T	●	
ROR		

● Thumb mode only

● Arm and Thumb modes

7.1.191 RRX

Description

Rotate right through carry

Thumb syntax (32-bit)

RRX *Rd, Rn* [1]
 RRXS *Rd, Rn* [1]

Arm syntax

RRX *Rd, Rn* [1]
 RRXS *Rd, Rn* [1]

Notes

[1] *Rd, Rn* ≠ PC; *Rd, Rn* ≠ SP; only available when 'Relaxed UAL' enabled

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
RRX	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●	●		●

- Arm mode only
- Thumb mode only
- Arm and Thumb modes

7.1.192 RSB

Description

Reverse subtract

Thumb syntax (16-bit)

RSBS *Rd, Rn, #0* [1]

Thumb syntax (32-bit)

RSB *Rd, Rn, #const* [2]

RSB *Rd, Rn, Rm, shift* [3]

RSBS *Rd, Rn, #const* [2]

RSBS *Rd, Rn, Rm, shift* [3]

Arm syntax

RSB *Rd, Rn, #const* [4]

RSB *Rd, Rn, Rm, shift* [5]

RSBS *Rd, Rn, #const* [4]

RSBS *Rd, Rn, Rm, shift* [5]

Notes

[1] *Rd, Rn* must be **R0...R7**; RSB permitted in an IT block

[2] *Rd, Rn* ≠ **PC**; *Rd, Rn* ≠ **SP**; *const* is one of **\$00xy00xy**, **\$xy00xy00**, **\$xyxyxyxy**, **\$xy LSL 0...24**

[3] *Rd, Rn, Rm* ≠ **PC**; *Rd, Rn, Rm* ≠ **SP**; *shift* is one of **LSL #0...31**, **LSR #1...32**, **ASR #1...32**, **RRX**

[4] *Rd, Rn* ≠ **PC**; *Rd, Rn* ≠ **SP**; *const* is **\$xy ROR 2n**

[5] *Rd, Rn, Rm* ≠ **PC**; *Rd, Rn, Rm* ≠ **SP**; *shift* is one of **LSL #0...31/*Rs***, **LSR #1...32/*Rs***, **ASR #1...32/*Rs***, **ROR #1...31/*Rs***, **RRX**

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
RSB	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●

● Thumb mode only

● Arm and Thumb modes

7.1.193 RSC

Description

Reverse subtract with carry

Arm syntax

RSC *Rd, Rn, #const* [1]
 RSC *Rd, Rn, Rm, shift* [2]
 RSCS *Rd, Rn, #const* [1]
 RSCS *Rd, Rn, Rm, shift* [2]

Notes

[1] *Rd, Rn* ≠ PC; *Rd, Rn* ≠ SP; *const* is \$xy ROR 2n

[2] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP; *shift* is one of LSL #0...31/*Rs*, LSR #1...32/*Rs*, ASR #1...32/*Rs*, ROR #1...31/*Rs*, RRX

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
RSC	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•						

- Arm mode only

7.1.194 SADD16

Description
Signed parallel add, 16-bit

Thumb syntax (32-bit)

SADD16 *Rd, Rn, Rm* [1]

Arm syntax

SADD16 *Rd, Rn, Rm* [1]

Notes
[1] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SADD16										●	●	●	●	●	●	●	●	●	●				●		

- Thumb mode only
- Arm and Thumb modes

7.1.195 SADD8

Description

Signed parallel add, 8-bit

Thumb syntax (32-bit)

SADD8 *Rd, Rn, Rm* [1]

Arm syntax

SADD8 *Rd, Rn, Rm* [1]

Notes

[1] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SADD8										•	•	•	•	•	•	•	•	•	•				•		

- Thumb mode only
- Arm and Thumb modes

7.1.196 SASX

Description

Signed parallel add and subtract with exchange, 16-bit

Thumb syntax (32-bit)

SASX *Rd, Rn, Rm* [1]

Arm syntax

SASX *Rd, Rn, Rm* [1]

Notes

[1] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SASX										●	●	●	●	●	●	●	●	●	●				●		

- Thumb mode only
- Arm and Thumb modes

7.1.197 SBC

Description

Subtract with carry

Thumb syntax (16-bit)

SBCS *Rd, Rn* [1]

Thumb syntax (32-bit)

SBC *Rd, Rn, #const* [2]

SBC *Rd, Rn, Rm, shift* [3]

SBCS *Rd, Rn, #const* [2]

SBCS *Rd, Rn, Rm, shift* [3]

Arm syntax

SBC *Rd, Rn, #const* [4]

SBC *Rd, Rn, Rm, shift* [5]

SBCS *Rd, Rn, #const* [4]

SBCS *Rd, Rn, Rm, shift* [5]

Notes

[1] *Rd, Rn* must be **R0...R7**; SBC permitted in an IT block

[2] *Rd, Rn* ≠ **PC**; *Rd, Rn* ≠ **SP**; *const* is one of *\$00xy00xy*, *\$xy00xy00*, *\$xyxyxyxy*, *\$xy* **LSL** 0...24

[3] *Rd, Rn, Rm* ≠ **PC**; *Rd, Rn, Rm* ≠ **SP**; *shift* is one of **LSL** #0...31, **LSR** #1...32, **ASR** #1...32, **RRX**

[4] *Rd, Rn* ≠ **PC**; *Rd, Rn* ≠ **SP**; *const* is *\$xy* **ROR** 2n

[5] *Rd, Rn, Rm* ≠ **PC**; *Rd, Rn, Rm* ≠ **SP**; *shift* is one of **LSL** #0...31/*Rs*, **LSR** #1...32/*Rs*, **ASR** #1...32/*Rs*, **ROR** #1...31/*Rs*, **RRX**

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SBC	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●

● Thumb mode only

● Arm and Thumb modes

7.1.199 SDIV

Description

Signed divide

Thumb syntax (32-bit)

SDIV *Rd, Rn, Rm* [1]

Arm syntax

SDIV *Rd, Rn, Rm* [1]

Notes

[1] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SDIV											●	●		●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.200 **SEL**

Description

Select bytes

Thumb syntax (32-bit)

SEL *Rd, Rn, Rm*

Arm syntax

SEL *Rd, Rn, Rm*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SEL										●	●	●	●	●	●	●	●	●	●				●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.203 SEV

Description

Signal event

Thumb syntax (16-bit)

SEV

Thumb syntax (32-bit)

SEV

Arm syntax

SEV

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SEV						•	•	•	•		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•

- Thumb mode only
- Arm and Thumb modes

7.1.204 SEVL

Description

Signal event, local

Thumb syntax (16-bit)

SEVL

Thumb syntax (32-bit)

SEVL

Arm syntax

SEVL

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SEVL														•	•	•	•	•	•						

- Arm and Thumb modes

7.1.205 SHADD16

Description

Signed parallel add, halved, 16-bit

Thumb syntax (32-bit)

SHADD16 *Rd, Rn, Rm* [1]

Arm syntax

SHADD16 *Rd, Rn, Rm* [1]

Notes

[1] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SHADD16										●	●	●	●	●	●	●	●	●	●				●		

- Thumb mode only
- Arm and Thumb modes

7.1.206 SHADD8

Description

Signed parallel add, halved, 8-bit

Thumb syntax (32-bit)

SHADD8 *Rd, Rn, Rm* [1]

Arm syntax

SHADD8 *Rd, Rn, Rm* [1]

Notes

[1] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SHADD8										●	●	●	●	●	●	●	●	●	●				●		

- Thumb mode only
- Arm and Thumb modes

7.1.207 SHASX

Description

Signed add and subtract with exchange, halved, 16-bit

Thumb syntax (32-bit)

SHASX *Rd, Rn, Rm* [1]

Arm syntax

SHASX *Rd, Rn, Rm* [1]

Notes

[1] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SHASX										●	●	●	●	●	●	●	●	●	●				●		

- Thumb mode only
- Arm and Thumb modes

7.1.208 SHSAX

Description

Signed subtract and add with exchange, halved, 16-bit

Thumb syntax (32-bit)

SHSAX *Rd, Rn, Rm* [1]

Arm syntax

SHSAX *Rd, Rn, Rm* [1]

Notes

[1] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SHSAX										●	●	●	●	●	●	●	●	●	●				●		

- Thumb mode only
- Arm and Thumb modes

7.1.209 SHSUB16

Description

Signed parallel subtract, halved, 16-bit

Thumb syntax (32-bit)

SHSUB16 *Rd, Rn, Rm* [1]

Arm syntax

SHSUB16 *Rd, Rn, Rm* [1]

Notes

[1] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SHSUB16										●	●	●	●	●	●	●	●	●	●				●		

- Thumb mode only
- Arm and Thumb modes

7.1.210 SHSUB8

Description

Signed parallel subtract, halved, 8-bit

Thumb syntax (32-bit)

SHSUB8 *Rd, Rn, Rm* [1]

Arm syntax

SHSUB8 *Rd, Rn, Rm* [1]

Notes

[1] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SHSUB8										•	•	•	•	•	•	•	•	•	•				•		

- Thumb mode only
- Arm and Thumb modes

7.1.211 SMLABB

Description

Signed halfword multiply, word accumulate

Thumb syntax (32-bit)

SMLABB *Rd, Rn, Rm, Ra* [1]

Arm syntax

SMLABB *Rd, Rn, Rm, Ra* [1]

Notes

[1] *Rd, Rn, Rm, Ra* ≠ PC; *Rd, Rn, Rm, Ra* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SMLABB										●	●	●	●	●	●	●	●	●	●				●		

- Thumb mode only
- Arm and Thumb modes

7.1.212 SMLABT

Description

Signed halfword multiply, word accumulate

Thumb syntax (32-bit)

SMLABT *Rd, Rn, Rm, Ra* [1]

Arm syntax

SMLABT *Rd, Rn, Rm, Ra* [1]

Notes

[1] *Rd, Rn, Rm, Ra* ≠ PC; *Rd, Rn, Rm, Ra* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SMLABT										•	•	•	•	•	•	•	•	•	•				•		

- Thumb mode only
- Arm and Thumb modes

7.1.213 SMLAD

Description

Dual 16-bit signed multiply-add

Thumb syntax (32-bit)

SMLAD *Rd, Rn, Rm, Ra* [1]

Arm syntax

SMLAD *Rd, Rn, Rm, Ra* [1]

Notes

[1] *Rd, Rn, Rm, Ra* ≠ PC; *Rd, Rn, Rm, Ra* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SMLAD										●	●	●	●	●	●	●	●	●	●				●		

- Thumb mode only
- Arm and Thumb modes

7.1.214 SMLADX

Description

Dual 16-bit signed multiply-add with exchange

Thumb syntax (32-bit)

SMLADX *Rd, Rn, Rm, Ra* [1]

Arm syntax

SMLADX *Rd, Rn, Rm, Ra* [1]

Notes

[1] *Rd, Rn, Rm, Ra* ≠ PC; *Rd, Rn, Rm, Ra* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SMLADX										●	●	●	●	●	●	●	●	●	●				●		

- Thumb mode only
- Arm and Thumb modes

7.1.216 SMLALBB

Description

Signed halfword multiply, doubleword accumulate

Thumb syntax (32-bit)

SMLALBB *Rd, Rn, Rm, Ra* [1]

Arm syntax

SMLALBB *Rd, Rn, Rm, Ra* [1]

Notes

[1] *Rd, Rn, Rm, Ra* ≠ PC; *Rd, Rn, Rm, Ra* ≠ SP; *Rd* ≠ *Rn*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SMLALBB										●	●	●	●	●	●	●	●	●	●				●		

- Thumb mode only
- Arm and Thumb modes

7.1.217 SMLALBT

Description

Signed halfword multiply, doubleword accumulate

Thumb syntax (32-bit)

SMLALBT *Rd, Rn, Rm, Ra* [1]

Arm syntax

SMLALBT *Rd, Rn, Rm, Ra* [1]

Notes

[1] *Rd, Rn, Rm, Ra* ≠ PC; *Rd, Rn, Rm, Ra* ≠ SP; *Rd* ≠ *Rn*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SMLALBT										●	●	●	●	●	●	●	●	●	●				●		

- Thumb mode only
- Arm and Thumb modes

7.1.218 SMLALD

Description

Dual 16-bit signed multiply-add, doubleword accumulate

Thumb syntax (32-bit)

SMLALD *Rd, Rn, Rm, Ra* [1]

Arm syntax

SMLALD *Rd, Rn, Rm, Ra* [1]

Notes

[1] *Rd, Rn, Rm, Ra* ≠ **PC**; *Rd, Rn, Rm, Ra* ≠ **SP**; *Rd* ≠ *Rn*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SMLALD										●	●	●	●	●	●	●	●	●	●				●		

- Arm mode only
- Thumb mode only
- Arm and Thumb modes

7.1.219 SMLALDX

Description

Dual 16-bit signed multiply-add with exchange, doubleword accumulate

Thumb syntax (32-bit)

SMLALDX *Rd, Rn, Rm, Ra* [1]

Arm syntax

SMLALDX *Rd, Rn, Rm, Ra* [1]

Notes

[1] *Rd, Rn, Rm, Ra* ≠ PC; *Rd, Rn, Rm, Ra* ≠ SP; *Rd* ≠ *Rn*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SMLALDX										●	●	●	●	●	●	●	●	●	●				●		

- Arm mode only
- Thumb mode only
- Arm and Thumb modes

7.1.220 SMLALTB

Description

Signed halfword multiply, doubleword accumulate

Thumb syntax (32-bit)

SMLALTB *Rd, Rn, Rm, Ra* [1]

Arm syntax

SMLALTB *Rd, Rn, Rm, Ra* [1]

Notes

[1] *Rd, Rn, Rm, Ra* ≠ PC; *Rd, Rn, Rm, Ra* ≠ SP; *Rd* ≠ *Rn*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SMLALTB										●	●	●	●	●	●	●	●	●	●				●		

- Thumb mode only
- Arm and Thumb modes

7.1.221 SMLALTT

Description

Signed halfword multiply, doubleword accumulate

Thumb syntax (32-bit)

SMLALTT *Rd, Rn, Rm, Ra* [1]

Arm syntax

SMLALTT *Rd, Rn, Rm, Ra* [1]

Notes

[1] *Rd, Rn, Rm, Ra* ≠ PC; *Rd, Rn, Rm, Ra* ≠ SP; *Rd* ≠ *Rn*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SMLALTT										●	●	●	●	●	●	●	●	●	●				●		

- Thumb mode only
- Arm and Thumb modes

7.1.222 SMLATB

Description

Signed halfword multiply, word accumulate

Thumb syntax (32-bit)

SMLATB *Rd, Rn, Rm, Ra* [1]

Arm syntax

SMLATB *Rd, Rn, Rm, Ra* [1]

Notes

[1] *Rd, Rn, Rm, Ra* ≠ PC; *Rd, Rn, Rm, Ra* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SMLATB										●	●	●	●	●	●	●	●	●	●				●		

- Thumb mode only
- Arm and Thumb modes

7.1.223 SMLATT

Description

Signed halfword multiply, word accumulate

Thumb syntax (32-bit)

SMLATT *Rd, Rn, Rm, Ra* [1]

Arm syntax

SMLATT *Rd, Rn, Rm, Ra* [1]

Notes

[1] *Rd, Rn, Rm, Ra* ≠ PC; *Rd, Rn, Rm, Ra* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SMLATT										●	●	●	●	●	●	●	●	●	●				●		

- Thumb mode only
- Arm and Thumb modes

7.1.224 SMLAWB

Description

Signed multiply-accumulate wide

Thumb syntax (32-bit)

SMLAWB *Rd, Rn, Rm, Ra* [1]

Arm syntax

SMLAWB *Rd, Rn, Rm, Ra* [1]

Notes

[1] *Rd, Rn, Rm, Ra* ≠ PC; *Rd, Rn, Rm, Ra* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SMLAWB										•	•	•	•	•	•	•	•	•	•				•		

- Thumb mode only
- Arm and Thumb modes

7.1.225 SMLAWT

Description

Signed multiply-accumulate wide

Thumb syntax (32-bit)

SMLAWT *Rd, Rn, Rm, Ra* [1]

Arm syntax

SMLAWT *Rd, Rn, Rm, Ra* [1]

Notes

[1] *Rd, Rn, Rm, Ra* ≠ PC; *Rd, Rn, Rm, Ra* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SMLAWT										●	●	●	●	●	●	●	●	●	●				●		

- Thumb mode only
- Arm and Thumb modes

7.1.226 SMLSD

Description

Dual 16-bit signed multiply-subtract

Thumb syntax (32-bit)

SMLSD *Rd, Rn, Rm, Ra* [1]

Arm syntax

SMLSD *Rd, Rn, Rm, Ra* [1]

Notes

[1] *Rd, Rn, Rm, Ra* ≠ **PC**; *Rd, Rn, Rm, Ra* ≠ **SP**

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SMLSD										•	•	•	•	•	•	•	•	•	•				•		

- Thumb mode only
- Arm and Thumb modes

7.1.227 SMLSDX

Description

Dual 16-bit signed multiply-subtract with exchange

Thumb syntax (32-bit)

SMLSDX *Rd, Rn, Rm, Ra* [1]

Arm syntax

SMLSDX *Rd, Rn, Rm, Ra* [1]

Notes

[1] *Rd, Rn, Rm, Ra* ≠ PC; *Rd, Rn, Rm, Ra* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SMLSDX										●	●	●	●	●	●	●	●	●	●				●		

- Thumb mode only
- Arm and Thumb modes

7.1.228 SMLS LD

Description

Dual 16-bit signed multiply-subtract, doubleword accumulate

Thumb syntax (32-bit)

SMLS LD *Rd, Rn, Rm, Ra* [1]

Arm syntax

SMLS LD *Rd, Rn, Rm, Ra* [1]

Notes

[1] *Rd, Rn, Rm, Ra* ≠ **PC**; *Rd, Rn, Rm, Ra* ≠ **SP**; *Rd* ≠ *Rn*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SMLS LD										•	•	•	•	•	•	•	•	•	•				•		

- Thumb mode only
- Arm and Thumb modes

7.1.229 SMLS�DX

Description

Dual 16-bit signed multiply-subtract with exchange, doubleword accumulate

Thumb syntax (32-bit)

SMLS�DX *Rd, Rn, Rm, Ra* [1]

Arm syntax

SMLS�DX *Rd, Rn, Rm, Ra* [1]

Notes

[1] *Rd, Rn, Rm, Ra* ≠ PC; *Rd, Rn, Rm, Ra* ≠ SP; *Rd* ≠ *Rn*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SMLS�DX										•	•	•	•	•	•	•	•	•	•				•		

- Thumb mode only
- Arm and Thumb modes

7.1.230 SMMLA

Description

Signed most-significant-word multiply-accumulate

Thumb syntax (32-bit)

SMMLA *Rd, Rn, Rm, Ra* [1]

Arm syntax

SMMLA *Rd, Rn, Rm, Ra* [1]

Notes

[1] *Rd, Rn, Rm, Ra* ≠ PC; *Rd, Rn, Rm, Ra* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SMMLA										•	•	•	•	•	•	•	•	•	•				•		

- Thumb mode only
- Arm and Thumb modes

7.1.231 SMMLAR

Description

Signed most-significant-word multiply-accumulate, rounded

Thumb syntax (32-bit)

SMMLAR *Rd, Rn, Rm, Ra* [1]

Arm syntax

SMMLAR *Rd, Rn, Rm, Ra* [1]

Notes

[1] *Rd, Rn, Rm, Ra* ≠ PC; *Rd, Rn, Rm, Ra* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SMMLAR										●	●	●	●	●	●	●	●	●	●				●		

- Thumb mode only
- Arm and Thumb modes

7.1.232 SMMLS

Description

Signed most-significant-word multiply-subtract

Thumb syntax (32-bit)

SMMLS *Rd, Rn, Rm, Ra* [1]

Arm syntax

SMMLS *Rd, Rn, Rm, Ra* [1]

Notes

[1] *Rd, Rn, Rm, Ra* ≠ PC; *Rd, Rn, Rm, Ra* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SMMLS										•	•	•	•	•	•	•	•	•	•				•		

- Thumb mode only
- Arm and Thumb modes

7.1.233 SMMLSR

Description

Signed most-significant-word multiply-subtract, rounded

Thumb syntax (32-bit)

SMMLSR *Rd, Rn, Rm, Ra* [1]

Arm syntax

SMMLSR *Rd, Rn, Rm, Ra* [1]

Notes

[1] *Rd, Rn, Rm, Ra* ≠ PC; *Rd, Rn, Rm, Ra* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SMMLSR										•	•	•	•	•	•	•	•	•	•				•		

- Thumb mode only
- Arm and Thumb modes

7.1.234 SMMUL

Description

Signed most-significant-word multiply

Thumb syntax (32-bit)

SMMUL *Rd, Rn, Rm* [1]

Arm syntax

SMMUL *Rd, Rn, Rm* [1]

Notes

[1] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SMMUL										●	●	●	●	●	●	●	●	●	●				●		

- Thumb mode only
- Arm and Thumb modes

7.1.235 SMMULR

Description

Signed most-significant-word multiply, rounded

Thumb syntax (32-bit)

SMMULR *Rd, Rn, Rm* [1]

Arm syntax

SMMULR *Rd, Rn, Rm* [1]

Notes

[1] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SMMULR										●	●	●	●	●	●	●	●	●	●				●		

- Thumb mode only
- Arm and Thumb modes

7.1.236 SMUAD

Description

Dual signed multiply-add

Thumb syntax (32-bit)

SMUAD *Rd, Rn, Rm* [1]

Arm syntax

SMUAD *Rd, Rn, Rm* [1]

Notes

[1] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SMUAD										•	•	•	•	•	•	•	•	•	•				•		

- Thumb mode only
- Arm and Thumb modes

7.1.237 SMUADX

Description

Dual signed multiply-add with exchange

Thumb syntax (32-bit)

SMUADX *Rd, Rn, Rm* [1]

Arm syntax

SMUADX *Rd, Rn, Rm* [1]

Notes

[1] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SMUADX										●	●	●	●	●	●	●	●	●	●				●		

- Thumb mode only
- Arm and Thumb modes

7.1.238 SMULBB

Description

Signed halfword multiply, bottom-bottom

Thumb syntax (32-bit)

SMULBB *Rd, Rn, Rm* [1]

Arm syntax

SMULBB *Rd, Rn, Rm* [1]

Notes

[1] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SMULBB			•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•				•		

- Arm mode only
- Thumb mode only
- Arm and Thumb modes

7.1.239 SMULBT

Description

Signed halfword multiply, bottom-top

Thumb syntax (32-bit)

SMULBT *Rd, Rn, Rm* [1]

Arm syntax

SMULBT *Rd, Rn, Rm* [1]

Notes

[1] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SMULBT			•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•				•		

- Arm mode only
- Thumb mode only
- Arm and Thumb modes

7.1.240 SMULL

Description

Signed multiply long

Thumb syntax (32-bit)

SMULL *Rd, Rn, Rm, Ra* [1]

Arm syntax

SMULL *Rd, Rn, Rm, Ra* [2]

SMULLS *Rd, Rn, Rm, Ra* [2]

Notes

[1] *Rd, Rn, Rm, Ra* ≠ PC; *Rd, Rn, Rm, Ra* ≠ SP; *Rd* ≠ *Rn*

[2] *Rd, Rn, Rm, Ra* ≠ PC; *Rd* ≠ *Rn* ≠ *Rm*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SMULL	●	●	●							●	●	●	●	●	●	●	●	●	●			●	●		●

- Arm mode only
- Thumb mode only

7.1.241 SMULTB

Description

Signed halfword multiply, top-bottom

Thumb syntax (32-bit)

SMULTB *Rd, Rn, Rm* [1]

Arm syntax

SMULTB *Rd, Rn, Rm* [1]

Notes

[1] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SMULTB			•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•				•		

- Arm mode only
- Thumb mode only
- Arm and Thumb modes

7.1.242 SMULTT

Description

Signed halfword multiply, top-top

Thumb syntax (32-bit)

SMULTT *Rd, Rn, Rm* [1]

Arm syntax

SMULTT *Rd, Rn, Rm* [1]

Notes

[1] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SMULTT			•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•				•		

- Arm mode only
- Thumb mode only
- Arm and Thumb modes

7.1.243 SMULWB

Description

Signed multiply wide

Thumb syntax (32-bit)

SMULWB *Rd, Rn, Rm* [1]

Arm syntax

SMULWB *Rd, Rn, Rm* [1]

Notes

[1] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SMULWB			•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•				•		

- Arm mode only
- Thumb mode only
- Arm and Thumb modes

7.1.245 SMUSD

Description
Dual signed multiply-subtract

Thumb syntax (32-bit)

SMUSD *Rd, Rn, Rm* [1]

Arm syntax

SMUSD *Rd, Rn, Rm* [1]

Notes
[1] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SMUSD										●	●	●	●	●	●	●	●	●	●				●		

- Thumb mode only
- Arm and Thumb modes

7.1.246 SMUSDX

Description

Dual signed multiply-subtract with exchange

Thumb syntax (32-bit)

SMUSDX *Rd, Rn, Rm* [1]

Arm syntax

SMUSDX *Rd, Rn, Rm* [1]

Notes

[1] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SMUSDX										●	●	●	●	●	●	●	●	●	●				●		

- Thumb mode only
- Arm and Thumb modes

7.1.247 SRS

Description
Store return state

Thumb syntax (32-bit)

SRS SP, #0...31
SRS SP!, #0...31

Arm syntax

SRS SP, #0...31
SRS SP!, #0...31

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SRS				•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•						

- Arm and Thumb modes

7.1.248 **SRSDA**

Description

Store return state, decrement after

Arm syntax

```
SRSDA    SP, #0...31
SRSDA    SP!, #0...31
```

Availability

Instruction	8-M>Main	
	8-M/Base	
	7E-M	
	7-M	
	6S-M	
	6-M	
	8-R	•
	8.4-A	•
	8.3-A	•
	8.2-A	•
	8.1-A	•
	8-A	•
	7-A	•
7-R	•	
7VE	•	
6T2	•	
6ZK	•	
6Z	•	
6KZ	•	
6K	•	
6J	•	
6	•	
5TE		
5T		
4T		

- Arm mode only

7.1.249 **SRSDB**

Description

Store return state, decrement before

Thumb syntax (32-bit)

```
SRSDB    SP, #0...31
SRSDB    SP!, #0...31
```

Arm syntax

```
SRSDB    SP, #0...31
SRSDB    SP!, #0...31
```

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SRSDB				•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•						

- Arm and Thumb modes

7.1.250 SRSIA

Description

Store return state, increment after

Thumb syntax (32-bit)

```
SRSIA    SP, #0...31
SRSIA    SP!, #0...31
```

Arm syntax

```
SRSIA    SP, #0...31
SRSIA    SP!, #0...31
```

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SRSIA				•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•						

- Arm and Thumb modes

7.1.251 SRSIB

Description

Store return state, increment before

Arm syntax

SRSIB SP, #0...31

SRSIB SP!, #0...31

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SRSIB				•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•						

- Arm mode only

7.1.252 SSAT

Description

Signed saturate

Thumb syntax (32-bit)

SSAT *Rd*, #1...32, *Rn* [1]

Arm syntax

SSAT *Rd*, #1...32, *Rn* [1]

Notes

[1] *Rd*, *Rm* ≠ PC; *Rd*, *Rm* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SSAT										●	●	●	●	●	●	●	●	●	●			●	●		●

- Thumb mode only
- Arm and Thumb modes

7.1.253 SSAT16

Description

Signed saturate, parallel halfwords

Thumb syntax (32-bit)

SSAT16 *Rd*, #1...16, *Rn* [1]

Arm syntax

SSAT16 *Rd*, #1...16, *Rn* [1]

Notes

[1] *Rd*, *Rm* ≠ PC; *Rd*, *Rm* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SSAT16										●	●	●	●	●	●	●	●	●	●				●		

- Thumb mode only
- Arm and Thumb modes

7.1.254 SSAX

Description

Signed parallel subtract and add with exchange, 16-bit

Thumb syntax (32-bit)

SSAX *Rd, Rn, Rm* [1]

Arm syntax

SSAX *Rd, Rn, Rm* [1]

Notes

[1] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SSAX										●	●	●	●	●	●	●	●	●	●				●		

- Thumb mode only
- Arm and Thumb modes

7.1.256 SSUB8

Description
Signed parallel subtract, 8-bit

Thumb syntax (32-bit)

SSUB8 *Rd, Rn, Rm* [1]

Arm syntax

SSUB8 *Rd, Rn, Rm* [1]

Notes
[1] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SSUB8										●	●	●	●	●	●	●	●	●	●				●		

- Thumb mode only
- Arm and Thumb modes

7.1.260 STM

Description

Store multiple

Thumb syntax (16-bit)

STM *Rd*, {*Rn*, *Rm*...} [1]
 STM *Rd*!, {*Rn*, *Rm*...}

Thumb syntax (32-bit)

STM *Rd*, {*Rn*, *Rm*...} [2]
 STM *Rd*!, {*Rn*, *Rm*...} [2]

Arm syntax

STM *Rd*, {*Rn*, *Rm*...} [2]
 STM *Rd*!, {*Rn*, *Rm*...} [2]

Notes

[1] *Rd* must be R0...R7; *Rd* ≠ PC

[2] *Rd* ≠ PC

Availability

Instruction	8-M.Main	●
	8-M.Base	●
	7E-M	●
	7-M	●
	6S-M	●
	6-M	●
	8-R	●
	8.4-A	●
	8.3-A	●
	8.2-A	●
	8.1-A	●
	8-A	●
	7-A	●
	7-R	●
7VE	●	
6T2	●	
6ZK	●	
6Z	●	
6KZ	●	
6K	●	
6J	●	
6	●	
5TE	●	
5T	●	
4T	●	
STM		

- Thumb mode only
- Arm and Thumb modes

7.1.261 STMDA

Description

Store multiple decrement after

Thumb syntax (16-bit)

STMDA *Rd*, {*Rn*, *Rm*...}

Thumb syntax (32-bit)

STMDA *Rd*, {*Rn*, *Rm*...} [1]

STMDA *Rd*!, {*Rn*, *Rm*...} [1]

Arm syntax

STMDA *Rd*, {*Rn*, *Rm*...} [1]

STMDA *Rd*!, {*Rn*, *Rm*...} [1]

Notes

[1] *Rd* ≠ PC

Availability

Instruction	8-M.Main	●
	8-M.Base	●
	7E-M	●
	7-M	●
	6S-M	●
	6-M	●
	8-R	●
	8.4-A	●
	8.3-A	●
	8.2-A	●
	8.1-A	●
	8-A	●
	7-A	●
	7-R	●
7VE	●	
Instruction	6T2	●
	6ZK	●
	6Z	●
	6KZ	●
	6K	●
	6J	●
Instruction	6	●
	5TE	●
	5T	●
	4T	●
STMDA		

● Thumb mode only

● Arm and Thumb modes

7.1.262 STMDB

Description

Store multiple decrement before

Thumb syntax (16-bit)

STMDB SP!, {*Rn*, *Rm*...}

Thumb syntax (32-bit)

STMDB *Rd*, {*Rn*, *Rm*...} [1]

STMDB *Rd*!, {*Rn*, *Rm*...} [1]

Arm syntax

STMDB *Rd*, {*Rn*, *Rm*...} [1]

STMDB *Rd*!, {*Rn*, *Rm*...} [1]

Notes

[1] *Rd* ≠ PC

Availability

Instruction	8-M.Main	●
	8-M.Base	●
	7E-M	●
	7-M	●
	6S-M	●
	6-M	●
	8-R	●
	8.4-A	●
	8.3-A	●
	8.2-A	●
	8.1-A	●
	8-A	●
	7-A	●
	7-R	●
7VE	●	
6T2	●	
6ZK	●	
6Z	●	
6KZ	●	
6K	●	
6J	●	
6	●	
5TE	●	
5T	●	
4T	●	
STMDB		

● Thumb mode only

● Arm and Thumb modes

7.1.263 STMEA

Description

Store multiple empty ascending

Thumb syntax (16-bit)

STMEA SP!, {Rn, Rm...}

Thumb syntax (32-bit)

STMEA Rd, {Rn, Rm...} [1]

STMEA Rd!, {Rn, Rm...} [1]

Arm syntax

STMEA Rd, {Rn, Rm...} [1]

STMEA Rd!, {Rn, Rm...} [1]

Notes

[1] Rd ≠ PC

Availability

Instruction	8-M.Main	●
	8-M.Base	●
	7E-M	●
	7-M	●
	6S-M	●
	6-M	●
	8-R	●
	8.4-A	●
	8.3-A	●
	8.2-A	●
	8.1-A	●
	8-A	●
	7-A	●
	7-R	●
7VE	●	
6T2	●	
6ZK	●	
6Z	●	
6KZ	●	
6K	●	
6J	●	
6	●	
5TE	●	
5T	●	
4T	●	

● Thumb mode only

● Arm and Thumb modes

7.1.264 STMED

Description

Store multiple empty descending

Thumb syntax (16-bit)

STMED *Rd*, {*Rn*, *Rm*...}

Thumb syntax (32-bit)

STMED *Rd*, {*Rn*, *Rm*...} [1]

Arm syntax

STMED *Rd*, {*Rn*, *Rm*...} [1]

STMED *Rd*!, {*Rn*, *Rm*...} [1]

Notes

[1] *Rd* ≠ PC

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
STMED	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.265 STMFA

Description

Store multiple full ascending

Thumb syntax (16-bit)

STMFA *Rd*, {*Rn*, *Rm*...}

Thumb syntax (32-bit)

STMFA *Rd*, {*Rn*, *Rm*...} [1]

STMFA *Rd*!, {*Rn*, *Rm*...} [1]

Arm syntax

STMFA *Rd*, {*Rn*, *Rm*...} [1]

STMFA *Rd*!, {*Rn*, *Rm*...} [1]

Notes

[1] *Rd* ≠ PC

Availability

Instruction	8-M.Main	●
	8-M.Base	●
	7E-M	●
	7-M	●
	6S-M	●
	6-M	●
	8-R	●
	8.4-A	●
	8.3-A	●
	8.2-A	●
	8.1-A	●
	8-A	●
	7-A	●
	7-R	●
7VE	●	
6T2	●	
6ZK	●	
6Z	●	
6KZ	●	
6K	●	
6J	●	
6	●	
5TE	●	
5T	●	
4T	●	
STMFA		

● Thumb mode only

● Arm and Thumb modes

7.1.266 STMFD

Description

Store multiple full descending

Thumb syntax (16-bit)

STMFD {*Rn*, *Rm*...}
 STMFD *Rd*, {*Rn*, *Rm*...}
 STMFD *Rd*!, {*Rn*, *Rm*...} [1]

Thumb syntax (32-bit)

STMFD *Rd*, {*Rn*, *Rm*...} [2]
 STMFD *Rd*!, {*Rn*, *Rm*...} [2]

Arm syntax

STMFD *Rd*, {*Rn*, *Rm*...} [2]
 STMFD *Rd*!, {*Rn*, *Rm*...} [2]

Notes

[1] *Rd* must be R0...R7; *Rd* ≠ PC

[2] *Rd* ≠ PC

Availability

Instruction	8-M.Main	●
	8-M.Base	●
	7E-M	●
	7-M	●
	6S-M	●
	6-M	●
	8-R	●
	8.4-A	●
	8.3-A	●
	8.2-A	●
	8.1-A	●
	8-A	●
	7-A	●
7-R	●	
7VE	●	
6T2	●	
6ZK	●	
6Z	●	
6KZ	●	
6K	●	
6J	●	
6	●	
5TE	●	
5T	●	
4T	●	
STMFD		

● Thumb mode only

● Arm and Thumb modes

7.1.267 STMIA

Description

Store multiple increment after

Thumb syntax (16-bit)

STMIA *Rd*, {*Rn*, *Rm*...} [1]
 STMIA *Rd*!, {*Rn*, *Rm*...}

Thumb syntax (32-bit)

STMIA *Rd*, {*Rn*, *Rm*...} [2]
 STMIA *Rd*!, {*Rn*, *Rm*...} [2]

Arm syntax

STMIA *Rd*, {*Rn*, *Rm*...} [2]
 STMIA *Rd*!, {*Rn*, *Rm*...} [2]

Notes

[1] *Rd* must be R0...R7; *Rd* ≠ PC

[2] *Rd* ≠ PC

Availability

Instruction	8-M.Main	●
	8-M.Base	●
	7E-M	●
	7-M	●
	6S-M	●
	6-M	●
	8-R	●
	8.4-A	●
	8.3-A	●
	8.2-A	●
	8.1-A	●
	8-A	●
	7-A	●
	7-R	●
7VE	●	
6T2	●	
6ZK	●	
6Z	●	
6KZ	●	
6K	●	
6J	●	
6	●	
5TE	●	
5T	●	
4T	●	
STMIA		

- Thumb mode only
- Arm and Thumb modes

7.1.268 STMIB

Description

Store multiple increment before

Thumb syntax (16-bit)

STMIB *Rd*, {*Rn*, *Rm*...}

Thumb syntax (32-bit)

STMIB *Rd*, {*Rn*, *Rm*...} [1]

STMIB *Rd*!, {*Rn*, *Rm*...} [1]

Arm syntax

STMIB *Rd*, {*Rn*, *Rm*...} [1]

STMIB *Rd*!, {*Rn*, *Rm*...} [1]

Notes

[1] *Rd* ≠ PC

Availability

Instruction	8-M.Main	●
	8-M.Base	●
	7E-M	●
	7-M	●
	6S-M	●
	6-M	●
	8-R	●
	8.4-A	●
	8.3-A	●
	8.2-A	●
	8.1-A	●
	8-A	●
	7-A	●
	7-R	●
7VE	●	
6T2	●	
6ZK	●	
6Z	●	
6KZ	●	
6K	●	
6J	●	
6	●	
5TE	●	
5T	●	
4T	●	
STMIB		

- Thumb mode only
- Arm and Thumb modes

7.1.269 STR

Description

Store word

Thumb syntax (16-bit)

STR	<i>Rt</i> , [<i>Rn</i> , #0...124]	[1]
STR	<i>Rt</i> , [SP, #0...1020]	[2]
STR	<i>Rt</i> , [<i>Rn</i> , <i>Rm</i>]	[3]

Thumb syntax (32-bit)

STR	<i>Rt</i> , [<i>Rn</i> , #-255...4095]	[4]
STR	<i>Rt</i> , [<i>Rn</i> , #-255...255]!	[5]
STR	<i>Rt</i> , [<i>Rn</i>], #-255...255	[5]
STR	<i>Rt</i> , [<i>Rn</i> , <i>Rm</i> , LSL #1...3]	[6]

Arm syntax

STR	<i>Rt</i> , [<i>Rn</i> , #-4095...4095]	[5]
STR	<i>Rt</i> , [<i>Rn</i> , #-4095...4095]!	[5]
STR	<i>Rt</i> , [<i>Rn</i>], #-4095...4095	[5]
STR	<i>Rt</i> , [<i>Rn</i>], $\pm Rm$, <i>shift</i>	[7]
STR	<i>Rt</i> , [<i>Rn</i> , $\pm Rm$]	[7]
STR	<i>Rt</i> , [<i>Rn</i> , $\pm Rm$]!	[7]

Notes

- [1] *Rt*, *Rn* must be R0...R7; offset a multiple of 4
- [2] offset a multiple of 4
- [3] *Rt*, *Rn*, *Rm* must be R0...R7
- [4] *Rt*, *Rn* \neq PC
- [5] *Rt*, *Rn* \neq PC; *Rt* \neq *Rn*
- [6] *Rt*, *Rn*, *Rm* \neq PC; *Rm* \neq SP; LSL #0 is also permitted
- [7] *Rt*, *Rn*, *Rm* \neq PC; *Rm* \neq SP; *Rt* \neq *Rn*

Availability

Instruction	8-M.Main	●
	8-M.Base	●
	7E-M	●
	7-M	●
	6S-M	●
	6-M	●
	8-R	●
	8.4-A	●
	8.3-A	●
	8.2-A	●
	8.1-A	●
	8-A	●
	7-A	●
	7-R	●
7VE	●	
6T2	●	
6ZK	●	
6Z	●	
6KZ	●	
6K	●	
6J	●	
6	●	
5TE	●	
5T	●	
4T	●	
STR		

- Thumb mode only
- Arm and Thumb modes

7.1.270 STRB

Description

Store byte

Thumb syntax (16-bit)

STRB *Rt*, [*Rn*, #0...31] [1]
 STRB *Rt*, [*Rn*, *Rm*] [2]

Thumb syntax (32-bit)

STRB *Rt*, [*Rn*, #-255...4095] [3]
 STRB *Rt*, [*Rn*, #-255...255]! [4]
 STRB *Rt*, [*Rn*], #-255...255 [4]
 STRB *Rt*, [*Rn*, *Rm*, LSL #1...3] [5]

Arm syntax

STRB *Rt*, [*Rn*, #-4095...4095] [3]
 STRB *Rt*, [*Rn*, #-4095...4095]! [4]
 STRB *Rt*, [*Rn*], #-4095...4095 [4]
 STRB *Rt*, [*Rn*], ±*Rm*, *shift* [6]
 STRB *Rt*, [*Rn*, ±*Rm*] [7]
 STRB *Rt*, [*Rn*, ±*Rm*]! [6]

Notes

- [1] *Rt*, *Rn* must be R0...R7
- [2] *Rt*, *Rn*, *Rm* must be R0...R7
- [3] *Rt*, *Rn* ≠ PC; *Rt* ≠ SP
- [4] *Rt*, *Rn* ≠ PC; *Rt* ≠ SP; *Rt* ≠ *Rn*
- [5] *Rt*, *Rn*, *Rm* ≠ PC; *Rt*, *Rm* ≠ SP; LSL #0 is also permitted
- [6] *Rt*, *Rn*, *Rm* ≠ PC; *Rt*, *Rm* ≠ SP; *Rt* ≠ *Rn*
- [7] *Rt*, *Rn*, *Rm* ≠ PC; *Rt*, *Rm* ≠ SP

Availability

Instruction	8-M.Main	●
	8-M.Base	●
	7E-M	●
	7-M	●
	6S-M	●
	6-M	●
	8-R	●
	8.4-A	●
	8.3-A	●
	8.2-A	●
	8.1-A	●
	8-A	●
	7-A	●
	7-R	●
7VE	●	
6T2	●	
6ZK	●	
6Z	●	
6KZ	●	
6K	●	
6J	●	
6	●	
5TE	●	
5T	●	
4T	●	
STRB		

- Thumb mode only
- Arm and Thumb modes

7.1.271 STRBT

Description

Store byte, user mode

Thumb syntax (32-bit)

STRBT *Rt*, [*Rn*, #0...255] [1]

Arm syntax

STRBT *Rt*, [*Rn*], #-4095...4095 [2]

STRBT *Rt*, [*Rn*], ±*Rm*, *shift* [3]

Notes

[1] *Rt*, *Rn* ≠ PC; *Rt* ≠ SP

[2] *Rt*, *Rn* ≠ PC; *Rt* ≠ SP; *Rt* ≠ *Rn*

[3] *Rt*, *Rn*, *Rm* ≠ PC; *Rt*, *Rm* ≠ SP; *Rt* ≠ *Rn*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
STRBT	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●	●		●

● Arm mode only

● Thumb mode only

● Arm and Thumb modes

7.1.272 STRD

Description

Store doubleword

Thumb syntax (32-bit)

```
STRD    Rd1, Rd2, [Rn, #-1020...1020]    [1]
STRD    Rd1, Rd2, [Rn, #-1020...1020]!    [2]
STRD    Rd1, Rd2, [Rn], #-1020...1020    [2]
```

Arm syntax

```
STRD    Rd1, Rd2, [Rn, #-255...255]        [3]
STRD    Rd1, Rd2, [Rn, #-255...255]!        [4]
STRD    Rd1, Rd2, [Rn], #-255...255        [3]
STRD    Rd1, Rd2, [Rn, ±Rm]                [5]
STRD    Rd1, Rd2, [Rn, ±Rm]!                [6]
STRD    Rd1, Rd2, [Rn], ±Rm                [6]
```

Notes

- [1] *Rd1*, *Rd2*, *Rn* ≠ PC; *Rd1*, *Rd2* ≠ SP; offset a multiple of 4
 [2] *Rd1*, *Rd2*, *Rn* ≠ PC; *Rd1*, *Rd2* ≠ SP; *Rd1* ≠ *Rn*; *Rd2* ≠ *Rn*; offset a multiple of 4
 [3] *Rd1*, *Rd2*, *Rn* ≠ PC; *Rd2* ≠ SP; *Rd1* must be an even-numbered register; *Rd2* must be the following odd-numbered register
 [4] *Rd1*, *Rd2*, *Rn* ≠ PC; *Rd2* ≠ SP; *Rd1* ≠ *Rn*; *Rd2* ≠ *Rn*; *Rd1* must be an even-numbered register; *Rd2* must be the following odd-numbered register
 [5] *Rd1*, *Rd2*, *Rn*, *Rm* ≠ PC; *Rd2*, *Rm* ≠ SP; *Rd1* ≠ *Rd2*; *Rd1* must be an even-numbered register; *Rd2* must be the following odd-numbered register
 [6] *Rd1*, *Rd2*, *Rn*, *Rm* ≠ PC; *Rd2*, *Rm* ≠ SP; *Rd1* ≠ *Rd2* ≠ *Rm*; *Rd1* must be an even-numbered register; *Rd2* must be the following odd-numbered register

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
STRD			●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●	●		●

- Arm mode only
- Thumb mode only
- Arm and Thumb modes

7.1.273 STREX

Description

Store word exclusive

Thumb syntax (32-bit)

STREX *Rd1*, *Rd2*, [*Rn*, #0...1020] [1]

Arm syntax

STREX *Rd1*, *Rd2*, [*Rn*] [2]

Notes

[1] *Rd1*, *Rd2*, *Rn* ≠ **PC**; *Rd1*, *Rd2* ≠ **SP**; *Rd1* ≠ *Rd2*; *Rd1* ≠ *Rn*; offset a multiple of 4

[2] *Rd1*, *Rd2*, *Rn* ≠ **PC**; *Rd1*, *Rd2* ≠ **SP**; *Rd1* ≠ *Rd2*; *Rd1* ≠ *Rn*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
STREX										●	●	●	●	●	●	●	●	●	●			●	●	●	●

● Thumb mode only

● Arm and Thumb modes

7.1.275 STREXD

Description

Store doubleword exclusive

Thumb syntax (32-bit)

STREXD *Rd, Rt1, Rt2, [Rn]* [1]

Arm syntax

STREXD *Rd, Rt1, Rt2, [Rn]* [2]

Notes

[1] *Rd, Rt1, Rt2, Rn* ≠ PC; *Rd, Rt1, Rt2* ≠ SP; *Rd* ≠ *Rt1*; *Rd* ≠ *Rt2*; *Rd* ≠ *Rn*

[2] *Rd, Rt1, Rt2, Rn* ≠ PC; *Rd, Rt1, Rt2* ≠ SP; *Rd* ≠ *Rt1*; *Rd* ≠ *Rt2*; *Rd* ≠ *Rn*; *Rd* must be an even-numbered register; *Rt1* must be the following odd-numbered register

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
STREXD										•	•	•	•	•	•	•	•	•	•						

- Arm and Thumb modes

7.1.277 STRH

Description

Store halfword

Thumb syntax (16-bit)

STRH *Rt*, [*Rn*, #0...62] [1]
 STRH *Rt*, [*Rn*, *Rm*] [2]

Thumb syntax (32-bit)

STRH *Rt*, [*Rn*, #-255...4095] [3]
 STRH *Rt*, [*Rn*, #-255...255]! [4]
 STRH *Rt*, [*Rn*], #-255...255 [4]
 STRH *Rt*, [*Rn*, *Rm*, LSL #1...3] [5]

Arm syntax

STRH *Rt*, [*Rn*, #-255...255] [3]
 STRH *Rt*, [*Rn*, #-255...255]! [4]
 STRH *Rt*, [*Rn*], #-255...255 [4]
 STRH *Rt*, [*Rn*, ±*Rm*] [6]
 STRH *Rt*, [*Rn*, ±*Rm*]! [7]
 STRH *Rt*, [*Rn*], ±*Rm* [7]

Notes

- [1] *Rt*, *Rn* must be R0...R7; offset a multiple of 2
- [2] *Rt*, *Rn*, *Rm* must be R0...R7
- [3] *Rt*, *Rn* ≠ PC; *Rt* ≠ SP
- [4] *Rt*, *Rn* ≠ PC; *Rt* ≠ SP; *Rt* ≠ *Rn*
- [5] *Rt*, *Rn*, *Rm* ≠ PC; *Rt*, *Rm* ≠ SP; LSL #0 is also permitted
- [6] *Rt*, *Rn*, *Rm* ≠ PC; *Rt*, *Rm* ≠ SP
- [7] *Rt*, *Rn*, *Rm* ≠ PC; *Rt*, *Rm* ≠ SP; *Rt* ≠ *Rn*

Availability

Instruction	8-M.Main	●
	8-M.Base	●
	7E-M	●
	7-M	●
	6S-M	●
	6-M	●
	8-R	●
	8.4-A	●
	8.3-A	●
	8.2-A	●
	8.1-A	●
	8-A	●
	7-A	●
	7-R	●
7VE	●	
6T2	●	
6ZK	●	
6Z	●	
6KZ	●	
6K	●	
6J	●	
6	●	
5TE	●	
5T	●	
4T	●	
STRH		

- Thumb mode only
- Arm and Thumb modes

7.1.278 STRHT

Description

Store halfword, user mode

Thumb syntax (32-bit)

STRHT *Rt*, [*Rn*, #0...255] [1]

Arm syntax

STRHT *Rt*, [*Rn*], #-255...255 [2]

STRHT *Rt*, [*Rn*], ±*Rm* [3]

Notes

[1] *Rt*, *Rn* ≠ PC; *Rt* ≠ SP

[2] *Rt*, *Rn* ≠ PC; *Rt* ≠ SP; *Rt* ≠ *Rn*

[3] *Rt*, *Rn*, *Rm* ≠ PC; *Rt*, *Rm* ≠ SP; *Rt* ≠ *Rn*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
STRHT										•	•	•	•	•	•	•	•	•	•			•	•		•

• Thumb mode only

• Arm and Thumb modes

7.1.279 STRT

Description

Store word, user mode

Thumb syntax (32-bit)

STRT *Rt*, [*Rn*, #0...255] [1]

Arm syntax

STRT *Rt*, [*Rn*], #-4095...4095 [2]

STRT *Rt*, [*Rn*], ±*Rm*, *shift* [3]

Notes

[1] *Rt*, *Rn* ≠ PC; *Rt* ≠ SP

[2] *Rt*, *Rn* ≠ PC; *Rt* ≠ SP; *Rt* ≠ *Rn*

[3] *Rt*, *Rn*, *Rm* ≠ PC; *Rt*, *Rm* ≠ SP; *Rt* ≠ *Rn*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
STRT	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●	●		●

● Arm mode only

● Thumb mode only

● Arm and Thumb modes

7.1.280 SUB

Description

Subtract

Thumb syntax (16-bit)

SUB	SP, SP, #0...508	[1]
SUB	Rd, SP, #0	[2]
SUB	Rd, PC, #0	[2]
SUBS	Rd, Rn, Rm	[3]
SUBS	Rd, #0...255	[4]
SUBS	Rd, Rn, #0...7	[5]

Thumb syntax (32-bit)

SUB	Rd, Rn, #0...4095	[6]
SUB	Rd, Rn, #const	[7]
SUB	Rd, Rn, Rm, shift	[8]
SUBS	Rd, Rn, #const	[7]
SUBS	Rd, Rn, Rm, shift	[8]

Arm syntax

SUB	Rd, Rn, #const	[9]
SUB	Rd, Rn, Rm, shift	[10]
SUB	Rd, SP, Rm, shift	[11]
SUBS	Rd, Rn, #const	[12]
SUBS	Rd, Rn, Rm, shift	[10]
SUBS	Rd, SP, Rm, shift	[11]

Notes

- [1] offset a multiple of 4
- [2] *Rd* must be **R0...R7**
- [3] *Rd, Rn, Rm* must be **R0...R7**; SUB permitted in an IT block
- [4] *Rd* must be **R0...R7**; SUB permitted in an IT block
- [5] *Rd, Rn* must be **R0...R7**; SUB permitted in an IT block
- [6] *Rd* ≠ **PC**; *Rd* ≠ **SP**
- [7] *Rd, Rn* ≠ **PC**; *Rd, Rn* ≠ **SP**; *const* is one of \$00xy00xy, \$xy00xy00, \$xyxyxyxy, \$xy **LSL** 0...24
- [8] *Rd, Rn, Rm* ≠ **PC**; *Rd, Rn, Rm* ≠ **SP**; *shift* is one of **LSL** #0...31, **LSR** #1...32, **ASR** #1...32, **RRX**
- [9] *Rd* ≠ **SP**; *const* is \$xy **ROR** 2n
- [10] *Rd, Rn, Rm* ≠ **PC**; *Rd, Rm* ≠ **SP**; *shift* is one of **LSL** #0...31/*Rs*, **LSR** #1...32/*Rs*, **ASR** #1...32/*Rs*, **ROR** #1...31/*Rs*, **RRX**
- [11] *Rd, Rm* ≠ **PC**; *Rm* ≠ **SP**; *shift* is one of **LSL** #0...31/*Rs*, **LSR** #1...32/*Rs*, **ASR** #1...32/*Rs*, **ROR** #1...31/*Rs*, **RRX**
- [12] *Rd* ≠ **PC**; *Rd* ≠ **SP**; *const* is \$xy **ROR** 2n

Availability

Instruction	8-M.Main	●
	8-M.Base	●
	7E-M	●
	7-M	●
	6S-M	●
	6-M	●
	8-R	●
	8.4-A	●
	8.3-A	●
	8.2-A	●
	8.1-A	●
	8-A	●
	7-A	●
	7-R	●
	7VE	●
6T2	●	
6ZK	●	
6Z	●	
6KZ	●	
6K	●	
6J	●	
6	●	
5TE	●	
5T	●	
4T	●	
	SUB	

- Thumb mode only
- Arm and Thumb modes

7.1.282 SVC

Description

Supervisor call

Thumb syntax (16-bit)

SVC #0...255

Arm syntax

SVC 0...16777215

Availability

Instruction	8-M.Main	●
	8-M.Base	●
	7E-M	●
	7-M	●
	6S-M	●
	6-M	
	8-R	●
	8.4-A	●
	8.3-A	●
	8.2-A	●
	8.1-A	●
	8-A	●
	7-A	●
	7-R	●
7VE	●	
6T2	●	
6ZK	●	
6Z	●	
6KZ	●	
6K	●	
6J	●	
6	●	
5TE	●	
5T	●	
4T	●	
SVC		

- Thumb mode only
- Arm and Thumb modes

7.1.283 SXTAB

Description

Sign-extend byte, accumulate

Thumb syntax (32-bit)

SXTAB *Rd, Rn, Rm*{, ROR #*imm*} [1]

Arm syntax

SXTAB *Rd, Rn, Rm*{, ROR #*imm*} [1]

Notes

[1] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP; *imm* is one of 0, 8, 16, or 24

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SXTAB										●	●	●	●	●	●	●	●	●	●				●		

- Thumb mode only
- Arm and Thumb modes

7.1.284 SXTAB16

Description

Sign-extend bytes, 16-bit parallel accumulate

Thumb syntax (32-bit)

SXTAB16 *Rd, Rn, Rm*{, ROR #*imm*} [1]

Arm syntax

SXTAB16 *Rd, Rn, Rm*{, ROR #*imm*} [1]

Notes

[1] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP; *imm* is one of 0, 8, 16, or 24

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SXTAB16										●	●	●	●	●	●	●	●	●	●				●		

- Thumb mode only
- Arm and Thumb modes

7.1.287 SXTB16

Description

Sign-extend bytes, 16-bit parallel store

Thumb syntax (32-bit)

SXTB16 *Rd*, *Rn*{, **ROR** #*imm*} [1]

Arm syntax

SXTB16 *Rd*, *Rn*{, **ROR** #*imm*} [1]

Notes

[1] *Rd*, *Rn* ≠ **PC**; *Rd*, *Rn* ≠ **SP**; *imm* is one of 0, 8, 16, or 24

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SXTB16										●	●	●	●	●	●	●	●	●	●				●		

- Thumb mode only
- Arm and Thumb modes

7.1.288 SXTB

Description

Sign-extend halfword

Thumb syntax (16-bit)

SXTB *Rd*, *Rn*

Thumb syntax (32-bit)

SXTB *Rd*, *Rn*{, ROR #*imm*} [1]

Arm syntax

SXTB *Rd*, *Rn*{, ROR #*imm*} [1]

Notes

[1] *Rd*, *Rn* ≠ PC; *Rd*, *Rn* ≠ SP; *imm* is one of 0, 8, 16, or 24

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SXTB				●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.289 TBB

Description

Table branch, byte

Thumb syntax (32-bit)

TBB [Rd, Rn]

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
TBB										●	●	●	●	●	●	●	●	●	●			●	●		●

- Thumb mode only

7.1.290 TBH

Description

Table branch, halfword

Thumb syntax (32-bit)

TBH [Rd, Rn, LSL #1]

Availability

Instruction	8-M:Main	●
	8-M:Base	
	7E-M	●
	7-M	●
	6S-M	
	6-M	
	8-R	●
	8.4-A	●
	8.3-A	●
	8.2-A	●
	8.1-A	●
	8-A	●
	7-A	●
	7-R	●
	7VE	●
	6T2	●
6ZK		
6Z		
6KZ		
6K		
6J		
6		
5TE		
5T		
4T		

- Thumb mode only

7.1.291 TEQ

Description

Test equivalence

Thumb syntax (32-bit)

TEQ	<i>Rn</i> , # <i>const</i>	[1]
TEQ	<i>Rn</i> , <i>Rm</i> , <i>shift</i>	[2]
TEQS	<i>Rd</i> , # <i>const</i>	[3]
TEQS	<i>Rd</i> , <i>Rn</i> , <i>shift</i>	[4]

Arm syntax

TEQ	<i>Rn</i> , # <i>const</i>	[5]
TEQ	<i>Rn</i> , <i>Rm</i> , <i>shift</i>	[6]
TEQS	<i>Rd</i> , # <i>const</i>	[7]
TEQS	<i>Rd</i> , <i>Rn</i> , <i>shift</i>	[8]

Notes

- [1] *Rn* ≠ PC; *Rn* ≠ SP; *const* is one of \$00xy00xy, \$xy00xy00, \$xyxyxyxy, \$xy LSL 0...24
- [2] *Rn*, *Rm* ≠ PC; *Rn*, *Rm* ≠ SP; *shift* is one of LSL #0...31, LSR #1...32, ASR #1...32, RRX
- [3] *Rd* ≠ PC; *Rd* ≠ SP; *const* is one of \$00xy00xy, \$xy00xy00, \$xyxyxyxy, \$xy LSL 0...24
- [4] *Rd*, *Rn* ≠ PC; *Rd*, *Rn* ≠ SP; *shift* is one of LSL #0...31, LSR #1...32, ASR #1...32, RRX
- [5] *Rn* ≠ PC; *Rn* ≠ SP; *const* is \$xy ROR 2n
- [6] *Rn*, *Rm* ≠ PC; *Rn*, *Rm* ≠ SP; *shift* is one of LSL #0...31/*Rs*, LSR #1...32/*Rs*, ASR #1...32/*Rs*, ROR #1...31/*Rs*, RRX
- [7] *Rd* ≠ PC; *Rd* ≠ SP; *const* is \$xy ROR 2n
- [8] *Rd*, *Rn* ≠ PC; *Rd*, *Rn* ≠ SP; *shift* is one of LSL #0...31/*Rs*, LSR #1...32/*Rs*, ASR #1...32/*Rs*, ROR #1...31/*Rs*, RRX

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
TEQ	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●	●		●

- Arm mode only
- Thumb mode only
- Arm and Thumb modes

7.1.292 TST

Description

Test

Thumb syntax (16-bit)

TST *Rn*, *Rm* [1]
TSTS *Rd*, *Rn* [2]

Thumb syntax (32-bit)

TST *Rn*, #*const* [3]
TST *Rn*, *Rm*, *shift* [4]
TSTS *Rd*, #*const* [5]
TSTS *Rd*, *Rn*, *shift* [6]

Arm syntax

TST *Rn*, #*const* [7]
TST *Rn*, *Rm*, *shift* [8]
TSTS *Rd*, #*const* [9]
TSTS *Rd*, *Rn*, *shift* [10]

Notes

- [1] *Rn*, *Rm* must be R0...R7
[2] *Rd*, *Rn* must be R0...R7
[3] *Rn* ≠ PC; *Rn* ≠ SP; *const* is one of \$00xy00xy, \$xy00xy00, \$xyxyxyxy, \$xy LSL 0...24
[4] *Rn*, *Rm* ≠ PC; *Rn*, *Rm* ≠ SP; *shift* is one of LSL #0...31, LSR #1...32, ASR #1...32, RRX
[5] *Rd* ≠ PC; *Rd* ≠ SP; *const* is one of \$00xy00xy, \$xy00xy00, \$xyxyxyxy, \$xy LSL 0...24
[6] *Rd*, *Rn* ≠ PC; *Rd*, *Rn* ≠ SP; *shift* is one of LSL #0...31, LSR #1...32, ASR #1...32, RRX
[7] *Rn* ≠ PC; *Rn* ≠ SP; *const* is \$xy ROR 2n
[8] *Rn*, *Rm* ≠ PC; *Rn*, *Rm* ≠ SP; *shift* is one of LSL #0...31/*Rs*, LSR #1...32/*Rs*, ASR #1...32/*Rs*, ROR #1...31/*Rs*, RRX
[9] *Rd* ≠ PC; *Rd* ≠ SP; *const* is \$xy ROR 2n
[10] *Rd*, *Rn* ≠ PC; *Rd*, *Rn* ≠ SP; *shift* is one of LSL #0...31/*Rs*, LSR #1...32/*Rs*, ASR #1...32/*Rs*, ROR #1...31/*Rs*, RRX

Availability

Instruction	8-M.Main	●
	8-M.Base	●
	7E-M	●
	7-M	●
	6S-M	●
	6-M	●
	8-R	●
	8.4-A	●
	8.3-A	●
	8.2-A	●
	8.1-A	●
	8-A	●
	7-A	●
	7-R	●
7VE	●	
6T2	●	
6ZK	●	
6Z	●	
6KZ	●	
6K	●	
6J	●	
6	●	
5TE	●	
5T	●	
4T	●	
TST		

- Thumb mode only
- Arm and Thumb modes

7.1.293 TT

Description

Test target

Thumb syntax (32-bit)

TT *Rd, Rn* [1]

Notes

[1] *Rd, Rn* ≠ PC; *Rd* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
TT																								●	●

- Thumb mode only

7.1.295 TTAT

Description

Test target

Thumb syntax (32-bit)

TTAT *Rd, Rn* [1]

Notes

[1] *Rd, Rn* ≠ PC; *Rd* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
TTAT																								●	●

● Thumb mode only

7.1.297 UADD16

Description

Unsigned parallel add, 16-bit

Thumb syntax (32-bit)

UADD16 *Rd, Rn, Rm* [1]

Arm syntax

UADD16 *Rd, Rn, Rm* [1]

Notes

[1] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
UADD16										●	●	●	●	●	●	●	●	●	●				●		

- Thumb mode only
- Arm and Thumb modes

7.1.298 UADD8

Description

Unsigned parallel add, 8-bit

Thumb syntax (32-bit)

UADD8 *Rd, Rn, Rm* [1]

Arm syntax

UADD8 *Rd, Rn, Rm* [1]

Notes

[1] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
UADD8										●	●	●	●	●	●	●	●	●	●				●		

- Thumb mode only
- Arm and Thumb modes

7.1.299 UASX

Description

Unsigned add and subtract with exchange, 16-bit

Thumb syntax (32-bit)

UASX *Rd, Rn, Rm* [1]

Arm syntax

UASX *Rd, Rn, Rm* [1]

Notes

[1] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
UASX										•	•	•	•	•	•	•	•	•	•				•		

- Thumb mode only
- Arm and Thumb modes

7.1.300 UBFX

Description

Bitfield extract, unsigned

Thumb syntax (32-bit)

UBFX *Rd, Rn, #lsb, #width* [1]

Arm syntax

UBFX *Rd, Rn, #lsb, #width* [1]

Notes

[1] *Rd, Rn* ≠ PC; *Rd, Rn* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
UBFX										●	●	●	●	●	●	●	●	●	●			●	●		●

- Thumb mode only
- Arm and Thumb modes

7.1.301 UDF

Description

Undefined

Thumb syntax (16-bit)

UDF #0...255

Thumb syntax (32-bit)

UDF #0...65535

Arm syntax

UDF *Rd*, #0...65535

Availability

Instruction	UDF
	4T
	5T
	5TE
	6
	6J
	6K
	6KZ
	6Z
	6ZK
	6T2
	7VE
	7-R
	7-A
	8-A
8.1-A	
8.2-A	
8.3-A	
8.4-A	
8-R	
6-M	
6S-M	
7-M	
7E-M	
8-M.Base	
8-M.Main	

- Thumb mode only
- Arm and Thumb modes

7.1.302 UDIV

Description

Unsigned divide

Thumb syntax (32-bit)

UDIV *Rd, Rn, Rm* [1]

Arm syntax

UDIV *Rd, Rn, Rm* [1]

Notes

[1] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
UDIV											●	●		●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.303 UEXT16

Description

Zero-extend halfword

Thumb syntax (16-bit)

UEXT16 *Rd*, *Rn* [1]

Thumb syntax (32-bit)

UEXT16 *Rd*, *Rn*{, ROR #*imm*} [2]

Arm syntax

UEXT16 *Rd*, *Rn*{, ROR #*imm*} [2]

Notes

[1] only available when 'Relaxed UAL' enabled

[2] *Rd*, *Rn* ≠ PC; *Rd*, *Rn* ≠ SP; only available when 'Relaxed UAL' enabled; *imm* is one of 0, 8, 16, or 24

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
UEXT16				●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●

● Thumb mode only

● Arm and Thumb modes

7.1.304 UEXT8

Description

Zero-extend byte

Thumb syntax (16-bit)

UEXT8 *Rd, Rn* [1]

Thumb syntax (32-bit)

UEXT8 *Rd, Rn{, ROR #imm}* [2]

Arm syntax

UEXT8 *Rd, Rn{, ROR #imm}* [2]

Notes

[1] only available when 'Relaxed UAL' enabled

[2] *Rd, Rn* ≠ PC; *Rd, Rn* ≠ SP; only available when 'Relaxed UAL' enabled; *imm* is one of 0, 8, 16, or 24

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
UEXT8				●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●

● Thumb mode only

● Arm and Thumb modes

7.1.305 UHADD16

Description

Unsigned parallel add, halved, 16-bit

Thumb syntax (32-bit)

UHADD16 *Rd, Rn, Rm* [1]

Arm syntax

UHADD16 *Rd, Rn, Rm* [1]

Notes

[1] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
UHADD16										●	●	●	●	●	●	●	●	●	●				●		

- Thumb mode only
- Arm and Thumb modes

7.1.306 UHADD8

Description

Unsigned parallel add, halved, 8-bit

Thumb syntax (32-bit)

UHADD8 *Rd, Rn, Rm* [1]

Arm syntax

UHADD8 *Rd, Rn, Rm* [1]

Notes

[1] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
UHADD8										•	•	•	•	•	•	•	•	•	•				•		

- Thumb mode only
- Arm and Thumb modes

7.1.307 UHASX

Description

Unsigned add and subtract with exchange, halved, 16-bit

Thumb syntax (32-bit)

UHASX *Rd, Rn, Rm* [1]

Arm syntax

UHASX *Rd, Rn, Rm* [1]

Notes

[1] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
UHASX										●	●	●	●	●	●	●	●	●	●				●		

- Thumb mode only
- Arm and Thumb modes

7.1.308 UHSAX

Description

Unsigned subtract and add with exchange, halved, 16-bit

Thumb syntax (32-bit)

UHSAX *Rd, Rn, Rm* [1]

Arm syntax

UHSAX *Rd, Rn, Rm* [1]

Notes

[1] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
UHSAX										●	●	●	●	●	●	●	●	●	●				●		

- Thumb mode only
- Arm and Thumb modes

7.1.309 UHSUB16

Description

Unsigned parallel subtract, halved, 16-bit

Thumb syntax (32-bit)

UHSUB16 *Rd, Rn, Rm* [1]

Arm syntax

UHSUB16 *Rd, Rn, Rm* [1]

Notes

[1] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
UHSUB16										●	●	●	●	●	●	●	●	●	●				●		

- Thumb mode only
- Arm and Thumb modes

7.1.310 UHSUB8

Description

Unsigned parallel subtract, halved, 8-bit

Thumb syntax (32-bit)

UHSUB8 *Rd, Rn, Rm* [1]

Arm syntax

UHSUB8 *Rd, Rn, Rm* [1]

Notes

[1] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
UHSUB8										•	•	•	•	•	•	•	•	•	•				•		

- Thumb mode only
- Arm and Thumb modes

7.1.311 UMAAL

Description

Unsigned multiply-accumulate-accumulate long

Thumb syntax (32-bit)

UMAAL *Rd, Rn, Rm, Ra* [1]

Arm syntax

UMAAL *Rd, Rn, Rm, Ra* [1]

Notes

[1] *Rd, Rn, Rm, Ra* ≠ PC; *Rd, Rn, Rm, Ra* ≠ SP; *Rd* ≠ *Rn*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
UMAAL										●	●	●	●	●	●	●	●	●	●				●		

- Thumb mode only
- Arm and Thumb modes

7.1.312 UMLAL

Description

Unsigned multiply-accumulate long

Thumb syntax (32-bit)

UMLAL *Rd, Rn, Rm, Ra* [1]

Arm syntax

UMLAL *Rd, Rn, Rm, Ra* [2]

UMLALS *Rd, Rn, Rm, Ra* [2]

Notes

[1] *Rd, Rn, Rm, Ra* ≠ PC; *Rd, Rn, Rm, Ra* ≠ SP; *Rd* ≠ *Rn*

[2] *Rd, Rn, Rm, Ra* ≠ PC; *Rd* ≠ *Rn* ≠ *Rm*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
UMLAL	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●	●		●

- Arm mode only
- Thumb mode only

7.1.313 UMULL

Description

Unsigned multiply long

Thumb syntax (32-bit)

UMULL *Rd, Rn, Rm, Ra* [1]

Arm syntax

UMULL *Rd, Rn, Rm, Ra* [2]

UMULLS *Rd, Rn, Rm, Ra* [2]

Notes

[1] *Rd, Rn, Rm, Ra* ≠ PC; *Rd, Rn, Rm, Ra* ≠ SP; *Rd* ≠ *Rn*

[2] *Rd, Rn, Rm, Ra* ≠ PC; *Rd* ≠ *Rn* ≠ *Rm*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
UMULL	●	●	●							●	●	●	●	●	●	●	●	●	●			●	●		●

● Arm mode only

● Thumb mode only

7.1.314 UQADD16

Description

Unsigned parallel add, 16-bit, saturating

Thumb syntax (32-bit)

UQADD16 *Rd, Rn, Rm* [1]

Arm syntax

UQADD16 *Rd, Rn, Rm* [1]

Notes

[1] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
UQADD16										●	●	●	●	●	●	●	●	●	●				●		

- Thumb mode only
- Arm and Thumb modes

7.1.315 UQADD8

Description

Unsigned parallel add, 8-bit, saturating

Thumb syntax (32-bit)

UQADD8 *Rd, Rn, Rm* [1]

Arm syntax

UQADD8 *Rd, Rn, Rm* [1]

Notes

[1] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
UQADD8										•	•	•	•	•	•	•	•	•	•				•		

- Thumb mode only
- Arm and Thumb modes

7.1.316 UQASX

Description

Unsigned parallel add and subtract with exchange, saturating

Thumb syntax (32-bit)

UQASX *Rd, Rn, Rm* [1]

Arm syntax

UQASX *Rd, Rn, Rm* [1]

Notes

[1] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
UQASX										•	•	•	•	•	•	•	•	•	•				•		

- Thumb mode only
- Arm and Thumb modes

7.1.317 UQSAX

Description

Unsigned parallel subtract and add with exchange, saturating

Thumb syntax (32-bit)

UQSAX *Rd, Rn, Rm* [1]

Arm syntax

UQSAX *Rd, Rn, Rm* [1]

Notes

[1] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
UQSAX										•	•	•	•	•	•	•	•	•	•				•		

- Thumb mode only
- Arm and Thumb modes

7.1.318 UQSUB16

Description

Unsigned parallel subtract, 16-bit, saturating

Thumb syntax (32-bit)

UQSUB16 *Rd, Rn, Rm* [1]

Arm syntax

UQSUB16 *Rd, Rn, Rm* [1]

Notes

[1] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
UQSUB16										●	●	●	●	●	●	●	●	●	●				●		

- Thumb mode only
- Arm and Thumb modes

7.1.319 UQSUB8

Description

Unsigned parallel subtract, 8-bit, saturating

Thumb syntax (32-bit)

UQSUB8 *Rd, Rn, Rm* [1]

Arm syntax

UQSUB8 *Rd, Rn, Rm* [1]

Notes

[1] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
UQSUB8										●	●	●	●	●	●	●	●	●	●				●		

- Thumb mode only
- Arm and Thumb modes

7.1.320 USAD8

Description

Unsigned parallel sum of absolute differences, 8-bit

Thumb syntax (32-bit)

USAD8 *Rd, Rn, Rm* [1]

Arm syntax

USAD8 *Rd, Rn, Rm* [1]

Notes

[1] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
USAD8										●	●	●	●	●	●	●	●	●	●				●		

- Thumb mode only
- Arm and Thumb modes

7.1.321 USADA8

Description

Unsigned parallel sum of absolute differences and accumulate, 8-bit

Thumb syntax (32-bit)

USADA8 *Rd, Rn, Rm, Ra* [1]

Arm syntax

USADA8 *Rd, Rn, Rm, Ra* [1]

Notes

[1] *Rd, Rn, Rm, Ra* ≠ PC; *Rd, Rn, Rm, Ra* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
USADA8										•	•	•	•	•	•	•	•	•	•				•		

- Thumb mode only
- Arm and Thumb modes

7.1.322 USAT

Description

Unsigned saturate

Thumb syntax (32-bit)

USAT *Rd*, #0...31, *Rn* [1]

Arm syntax

USAT *Rd*, #0...31, *Rn* [1]

Notes

[1] *Rd*, *Rm* ≠ PC; *Rd*, *Rm* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
USAT										●	●	●	●	●	●	●	●	●	●			●	●		●

- Thumb mode only
- Arm and Thumb modes

7.1.323 USAT16

Description

Unsigned saturate, parallel halfwords

Thumb syntax (32-bit)

USAT16 *Rd*, #0...15, *Rn* [1]

Arm syntax

USAT16 *Rd*, #0...15, *Rn* [1]

Notes

[1] *Rd*, *Rm* ≠ PC; *Rd*, *Rm* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
USAT16										●	●	●	●	●	●	●	●	●	●				●		

- Thumb mode only
- Arm and Thumb modes

7.1.324 USAX

Description

Unsigned parallel subtract and add with exchange, 16-bit, saturating

Thumb syntax (32-bit)

USAX *Rd, Rn, Rm* [1]

Arm syntax

USAX *Rd, Rn, Rm* [1]

Notes

[1] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
USAX										•	•	•	•	•	•	•	•	•	•				•		

- Thumb mode only
- Arm and Thumb modes

7.1.325 USUB16

Description

Unsigned parallel subtract, 16-bit

Thumb syntax (32-bit)

USUB16 *Rd, Rn, Rm* [1]

Arm syntax

USUB16 *Rd, Rn, Rm* [1]

Notes

[1] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
USUB16										●	●	●	●	●	●	●	●	●	●				●		

- Thumb mode only
- Arm and Thumb modes

7.1.327 UXTAB

Description

Zero-extend byte, accumulate

Thumb syntax (32-bit)

UXTAB *Rd, Rn, Rm*{, ROR #*imm*} [1]

Arm syntax

UXTAB *Rd, Rn, Rm*{, ROR #*imm*} [1]

Notes

[1] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP; *imm* is one of 0, 8, 16, or 24

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
UXTAB										●	●	●	●	●	●	●	●	●	●				●		

- Thumb mode only
- Arm and Thumb modes

7.1.328 UXTAB16

Description

Zero-extend bytes, 16-bit parallel accumulate

Thumb syntax (32-bit)

UXTAB16 *Rd, Rn, Rm{, ROR #imm}* [1]

Arm syntax

UXTAB16 *Rd, Rn, Rm{, ROR #imm}* [1]

Notes

[1] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP; *imm* is one of 0, 8, 16, or 24

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
UXTAB16										●	●	●	●	●	●	●	●	●	●				●		

- Thumb mode only
- Arm and Thumb modes

7.1.329 UXTAH

Description

Zero-extend halfword, accumulate

Thumb syntax (32-bit)

UXTAH *Rd, Rn, Rm*{, ROR #*imm*} [1]

Arm syntax

UXTAH *Rd, Rn, Rm*{, ROR #*imm*} [1]

Notes

[1] *Rd, Rn, Rm* ≠ PC; *Rd, Rn, Rm* ≠ SP; *imm* is one of 0, 8, 16, or 24

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
UXTAH										●	●	●	●	●	●	●	●	●	●				●		

- Thumb mode only
- Arm and Thumb modes

7.1.330 UXTB

Description

Zero-extend byte

Thumb syntax (16-bit)

UXTB *Rd*, *Rn*

Thumb syntax (32-bit)

UXTB *Rd*, *Rn*{, ROR #*imm*} [1]

Arm syntax

UXTB *Rd*, *Rn*{, ROR #*imm*} [1]

Notes

[1] *Rd*, *Rn* ≠ PC; *Rd*, *Rn* ≠ SP; *imm* is one of 0, 8, 16, or 24

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
UXTB				●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.332 UXTH

Description

Zero-extend halfword

Thumb syntax (16-bit)

UXTH *Rd*, *Rn*

Thumb syntax (32-bit)

UXTH *Rd*, *Rn*{, ROR #*imm*} [1]

Arm syntax

UXTH *Rd*, *Rn*{, ROR #*imm*} [1]

Notes

[1] *Rd*, *Rn* ≠ PC; *Rd*, *Rn* ≠ SP; *imm* is one of 0, 8, 16, or 24

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
UXTH				●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.333 VABS

Description

Floating-point absolute

Thumb syntax (32-bit)

VABS.F32 *Sd, Sn*

VABS.F64 *Dd, Dn*

Arm syntax

VABS.F32 *Sd, Sn*

VABS.F64 *Dd, Dn*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
VABS											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.334 VADD

Description

Floating-point add

Thumb syntax (32-bit)

VADD.F32 *Sd, Sn, Sm*
VADD.F64 *Dd, Dn, Dm*

Arm syntax

VADD.F32 *Sd, Sn, Sm*
VADD.F64 *Dd, Dn, Dm*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
VADD											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.335 VCMP

Description

Floating-point compare

Thumb syntax (32-bit)

VCMP.F32 *Sd, Sn*
VCMP.F64 *Dd, Dn*

Arm syntax

VCMP.F32 *Sd, Sn*
VCMP.F64 *Dd, Dn*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
VCMP											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.336 VCMPE

Description

Floating-point compare raising exception on NaN

Thumb syntax (32-bit)

VCMPE.F32 *Sd, Sn*
VCMPE.F64 *Dd, Dn*

Arm syntax

VCMPE.F32 *Sd, Sn*
VCMPE.F64 *Dd, Dn*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
VCMPE											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.337 VCVT

Description

Floating-point convert

Thumb syntax (32-bit)

```
VCVT.U32.F32    Sd, Sn
VCVT.S32.F32    Sd, Sn
VCVT.F32.U32    Sd, Sn
VCVT.F32.S32    Sd, Sn
VCVT.F32.F64    Sd, Dn
VCVT.U32.F64    Sd, Dn
VCVT.S32.F64    Sd, Dn
VCVT.F64.F32    Dd, Sn
VCVT.F64.U32    Dd, Sn
VCVT.F64.S32    Dd, Sn
```

Arm syntax

```
VCVT.U32.F32    Sd, Sn
VCVT.S32.F32    Sd, Sn
VCVT.F32.U32    Sd, Sn
VCVT.F32.S32    Sd, Sn
VCVT.F32.F64    Sd, Dn
VCVT.U32.F64    Sd, Dn
VCVT.S32.F64    Sd, Dn
VCVT.F64.F32    Dd, Sn
VCVT.F64.U32    Dd, Sn
VCVT.F64.S32    Dd, Sn
```

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
VCVT											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.338 VCVTA

Description

Floating-point convert to integer with rounding to nearest with ties away

Thumb syntax (32-bit)

VCVTA.U32.F32 *Sd, Sn*

VCVTA.S32.F32 *Sd, Sn*

VCVTA.U32.F64 *Sd, Dn*

VCVTA.S32.F64 *Sd, Dn*

Arm syntax

VCVTA.U32.F32 *Sd, Sn*

VCVTA.S32.F32 *Sd, Sn*

VCVTA.U32.F64 *Sd, Dn*

VCVTA.S32.F64 *Sd, Dn*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
VCVTA											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.339 VCVTB

Description

Floating-point convert bottom

Thumb syntax (32-bit)

VCVTB.F32.F16 *Sd, Sn*
 VCVTB.F16.F32 *Sd, Sn*
 VCVTB.F16.F64 *Sd, Dn*
 VCVTB.F64.F16 *Dd, Sn*

Arm syntax

VCVTB.F32.F16 *Sd, Sn*
 VCVTB.F16.F32 *Sd, Sn*
 VCVTB.F16.F64 *Sd, Dn*
 VCVTB.F64.F16 *Dd, Sn*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
VCVTB											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.340 VCVTM

Description

Floating-point convert to integer with rounding to minus infinity

Thumb syntax (32-bit)

VCVTM.U32.F32 *Sd, Sn*
 VCVTM.S32.F32 *Sd, Sn*
 VCVTM.U32.F64 *Sd, Dn*
 VCVTM.S32.F64 *Sd, Dn*

Arm syntax

VCVTM.U32.F32 *Sd, Sn*
 VCVTM.S32.F32 *Sd, Sn*
 VCVTM.U32.F64 *Sd, Dn*
 VCVTM.S32.F64 *Sd, Dn*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
VCVTM											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.341 VCVTN

Description

Floating-point convert to integer with rounding to nearest

Thumb syntax (32-bit)

VCVTN.U32.F32 *Sd, Sn*

VCVTN.S32.F32 *Sd, Sn*

VCVTN.U32.F64 *Sd, Dn*

VCVTN.S32.F64 *Sd, Dn*

Arm syntax

VCVTN.U32.F32 *Sd, Sn*

VCVTN.S32.F32 *Sd, Sn*

VCVTN.U32.F64 *Sd, Dn*

VCVTN.S32.F64 *Sd, Dn*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
VCVTN											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.342 VCVTP

Description

Floating-point convert to integer with rounding to positive infinity

Thumb syntax (32-bit)

VCVTP.U32.F32 *Sd, Sn*

VCVTP.S32.F32 *Sd, Sn*

VCVTP.U32.F64 *Sd, Dn*

VCVTP.S32.F64 *Sd, Dn*

Arm syntax

VCVTP.U32.F32 *Sd, Sn*

VCVTP.S32.F32 *Sd, Sn*

VCVTP.U32.F64 *Sd, Dn*

VCVTP.S32.F64 *Sd, Dn*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
VCVTP											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.343 VCVTR

Description

Floating-point convert to integer with rounding from FPSCR

Thumb syntax (32-bit)

VCVTR.U32.F32 *Sd, Sn*
 VCVTR.S32.F32 *Sd, Sn*
 VCVTR.U32.F64 *Sd, Dn*
 VCVTR.S32.F64 *Sd, Dn*

Arm syntax

VCVTR.U32.F32 *Sd, Sn*
 VCVTR.S32.F32 *Sd, Sn*
 VCVTR.U32.F64 *Sd, Dn*
 VCVTR.S32.F64 *Sd, Dn*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
VCVTR											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.344 VCVTT

Description

Floating-point convert top

Thumb syntax (32-bit)

VCVTT.F32.F16 *Sd, Sn*

VCVTT.F16.F32 *Sd, Sn*

VCVTT.F16.F64 *Sd, Dn*

VCVTT.F64.F16 *Dd, Sn*

Arm syntax

VCVTT.F32.F16 *Sd, Sn*

VCVTT.F16.F32 *Sd, Sn*

VCVTT.F16.F64 *Sd, Dn*

VCVTT.F64.F16 *Dd, Sn*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
VCVTT											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.345 VDIV

Description

Floating-point divide

Thumb syntax (32-bit)

VDIV.F32 *Sd, Sn, Sm*
VDIV.F64 *Dd, Dn, Dm*

Arm syntax

VDIV.F32 *Sd, Sn, Sm*
VDIV.F64 *Dd, Dn, Dm*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
VDIV											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.346 VFMA

Description

Floating-point fused multiply-accumulate

Thumb syntax (32-bit)

VFMA.F32 *Sd, Sn, Sm*
VFMA.F64 *Dd, Dn, Dm*

Arm syntax

VFMA.F32 *Sd, Sn, Sm*
VFMA.F64 *Dd, Dn, Dm*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
VFMA											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.347 VFMS

Description

Floating-point fused multiply-subtract

Thumb syntax (32-bit)

VFMS.F32 *Sd, Sn, Sm*
VFMS.F64 *Dd, Dn, Dm*

Arm syntax

VFMS.F32 *Sd, Sn, Sm*
VFMS.F64 *Dd, Dn, Dm*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
VFMS											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.348 VFNMA

Description

Floating-point fused negate-multiply-accumulate

Thumb syntax (32-bit)

VFNMA.F32 *Sd, Sn, Sm*
VFNMA.F64 *Dd, Dn, Dm*

Arm syntax

VFNMA.F32 *Sd, Sn, Sm*
VFNMA.F64 *Dd, Dn, Dm*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
VFNMA											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.349 VFNMS

Description

Floating-point fused negate-multiply-subtract

Thumb syntax (32-bit)

VFNMS.F32 *Sd, Sn, Sm*
VFNMS.F64 *Dd, Dn, Dm*

Arm syntax

VFNMS.F32 *Sd, Sn, Sm*
VFNMS.F64 *Dd, Dn, Dm*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
VFNMS											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.351 VMAXNM

Description

Floating-point maximum

Thumb syntax (32-bit)

VMAXNM.F32 *Sd, Sn, Sm*
VMAXNM.F64 *Dd, Dn, Dm*

Arm syntax

VMAXNM.F32 *Sd, Sn, Sm*
VMAXNM.F64 *Dd, Dn, Dm*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
VMAXNM											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.352 VMINNM

Description

Floating-point minimum

Thumb syntax (32-bit)

VMINNM.F32 *Sd, Sn, Sm*
VMINNM.F64 *Dd, Dn, Dm*

Arm syntax

VMINNM.F32 *Sd, Sn, Sm*
VMINNM.F64 *Dd, Dn, Dm*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
VMINNM											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.353 VMLA

Description

Floating-point multiply-accumulate

Thumb syntax (32-bit)

VMLA.F32 *Sd, Sn, Sm*

VMLA.F64 *Dd, Dn, Dm*

Arm syntax

VMLA.F32 *Sd, Sn, Sm*

VMLA.F64 *Dd, Dn, Dm*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
VMLA											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.354 VMLS

Description

Floating-point multiply-subtract

Thumb syntax (32-bit)

VMLS.F32 *Sd, Sn, Sm*

VMLS.F64 *Dd, Dn, Dm*

Arm syntax

VMLS.F32 *Sd, Sn, Sm*

VMLS.F64 *Dd, Dn, Dm*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
VMLS											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.355 VMOV

Description

Floating-point move

Thumb syntax (32-bit)

VMOV *Rd, Sn*

VMOV *Sd, Rn*

VMOV *Dd, Rn, Rm*

VMOV *Rd1, Rd2, Dn*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
VMOV											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only

7.1.356 VMUL

Description

Floating-point multiply

Thumb syntax (32-bit)

VMUL.F32 *Sd, Sn, Sm*
VMUL.F64 *Dd, Dn, Dm*

Arm syntax

VMUL.F32 *Sd, Sn, Sm*
VMUL.F64 *Dd, Dn, Dm*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
VMUL											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.357 VNEG

Description

Floating-point negate

Thumb syntax (32-bit)

VNEG.F32 *Sd, Sn*
VNEG.F64 *Dd, Dn*

Arm syntax

VNEG.F32 *Sd, Sn*
VNEG.F64 *Dd, Dn*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
VNEG											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.358 VNMLA

Description

Floating-point negate-multiply-accumulate

Thumb syntax (32-bit)

VNMLA.F32 *Sd, Sn, Sm*
VNMLA.F64 *Dd, Dn, Dm*

Arm syntax

VNMLA.F32 *Sd, Sn, Sm*
VNMLA.F64 *Dd, Dn, Dm*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
VNMLA											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.359 VNMLS

Description

Floating-point negate-multiply-subtract

Thumb syntax (32-bit)

VNMLS.F32 *Sd, Sn, Sm*

VNMLS.F64 *Dd, Dn, Dm*

Arm syntax

VNMLS.F32 *Sd, Sn, Sm*

VNMLS.F64 *Dd, Dn, Dm*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
VNMLS											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.360 VNMUL

Description

Floating-point negate-multiply

Thumb syntax (32-bit)

VNMUL.F32 *Sd, Sn, Sm*

VNMUL.F64 *Dd, Dn, Dm*

Arm syntax

VNMUL.F32 *Sd, Sn, Sm*

VNMUL.F64 *Dd, Dn, Dm*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
VNMUL											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.361 VRINTA

Description

Floating-point round to integer with ties away

Thumb syntax (32-bit)

VRINTA.F32.F32 *Sd, Sn*
VRINTA.F64.F64 *Dd, Dn*

Arm syntax

VRINTA.F32.F32 *Sd, Sn*
VRINTA.F64.F64 *Dd, Dn*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
VRINTA											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.362 VRINTM

Description

Floating-point round to integer towards minus infinity

Thumb syntax (32-bit)

VRINTM.F32.F32 *Sd, Sn*
VRINTM.F64.F64 *Dd, Dn*

Arm syntax

VRINTM.F32.F32 *Sd, Sn*
VRINTM.F64.F64 *Dd, Dn*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
VRINTM											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.363 VRINTN

Description

Floating-point round to integer with ties to even

Thumb syntax (32-bit)

VRINTN.F32.F32 *Sd, Sn*
VRINTN.F64.F64 *Dd, Dn*

Arm syntax

VRINTN.F32.F32 *Sd, Sn*
VRINTN.F64.F64 *Dd, Dn*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
VRINTN											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.364 VRINTP

Description

Floating-point round to integer towards positive infinity

Thumb syntax (32-bit)

VRINTP.F32.F32 *Sd, Sn*
VRINTP.F64.F64 *Dd, Dn*

Arm syntax

VRINTP.F32.F32 *Sd, Sn*
VRINTP.F64.F64 *Dd, Dn*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
VRINTP											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.365 VRINTR

Description

Floating-point round to integer with rounding from FPSCR

Thumb syntax (32-bit)

VRINTR.F32.F32 *Sd, Sn*
VRINTR.F64.F64 *Dd, Dn*

Arm syntax

VRINTR.F32.F32 *Sd, Sn*
VRINTR.F64.F64 *Dd, Dn*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
VRINTR											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.366 VRINTX

Description

Floating-point round to integer with rounding from FPSCR

Thumb syntax (32-bit)

VRINTX.F32.F32 *Sd, Sn*

VRINTX.F64.F64 *Dd, Dn*

Arm syntax

VRINTX.F32.F32 *Sd, Sn*

VRINTX.F64.F64 *Dd, Dn*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
VRINTX											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.367 VRINTZ

Description

Floating-point round to integer towards zero

Thumb syntax (32-bit)

VRINTZ.F32.F32 *Sd, Sn*

VRINTZ.F64.F64 *Dd, Dn*

Arm syntax

VRINTZ.F32.F32 *Sd, Sn*

VRINTZ.F64.F64 *Dd, Dn*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
VRINTZ											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.368 VSELEQ

Description

Floating-point conditional select

Thumb syntax (32-bit)

VSELEQ.F32Sd, Sn, Sm[1]

VSELEQ.F64Dd, Dn, Dm[1]

Arm syntax

VSELEQ.F32Sd, Sn, Sm

VSELEQ.F64Dd, Dn, Dm

Notes

[1] not permitted in an IT block

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
VSELEQ											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.369 VSELGE

Description

Floating-point conditional select

Thumb syntax (32-bit)

VSELGE.F32Sd, Sn, Sm[1]

VSELGE.F64Dd, Dn, Dm[1]

Arm syntax

VSELGE.F32Sd, Sn, Sm

VSELGE.F64Dd, Dn, Dm

Notes

[1] not permitted in an IT block

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
VSELGE											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.370 VSELGT

Description

Floating-point conditional select

Thumb syntax (32-bit)

VSELGT.F32 *Sd, Sn, Sm* [1]
 VSELGT.F64 *Dd, Dn, Dm* [1]

Arm syntax

VSELGT.F32 *Sd, Sn, Sm*
 VSELGT.F64 *Dd, Dn, Dm*

Notes

[1] not permitted in an IT block

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
VSELGT											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.371 VSQRT

Description
Floating-point square root

Thumb syntax (32-bit)

VSQRT.F32 *Sd, Sn*
VSQRT.F64 *Dd, Dn*

Arm syntax

VSQRT.F32 *Sd, Sn*
VSQRT.F64 *Dd, Dn*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
VSQRT											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.372 VSTR

Description

Floating-point store

Thumb syntax (32-bit)

VSTR.32Sn, [Rn, #-1020...1020][1]

VSTR.64Dn, [Rn, #-1020...1020][1]

Notes

[1] Rn ≠ PC; offset a multiple of 4

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
VSTR											●	●	●	●	●	●	●	●	●			●	●	●	●

● Thumb mode only

7.1.373 VSUB

Description
Floating-point subtract

Thumb syntax (32-bit)

VSUB.F32 *Sd, Sn, Sm*
VSUB.F64 *Dd, Dn, Dm*

Arm syntax

VSUB.F32 *Sd, Sn, Sm*
VSUB.F64 *Dd, Dn, Dm*

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
VSUB											●	●	●	●	●	●	●	●	●			●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.374 **WFE**

Description

Wait for event

Thumb syntax (16-bit)

WFE

Thumb syntax (32-bit)

WFE

Arm syntax

WFE

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
WFE						●	●	●	●		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.375 WFI

Description

Wait for interrupt

Thumb syntax (16-bit)

WFI

Thumb syntax (32-bit)

WFI

Arm syntax

WFI

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
WFI						●	●	●	●		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.1.376 YIELD

Description

Yield

Thumb syntax (16-bit)

YIELD

Thumb syntax (32-bit)

YIELD

Arm syntax

YIELD

Availability

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
YIELD						●	●	●	●		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●

- Thumb mode only
- Arm and Thumb modes

7.2 Instructions by architecture

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
ADC	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	
ADD	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	
ADDW										•	•	•	•	•	•	•	•	•	•			•		•	
AND	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	
ASR	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	
B	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	
BCC	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	
BCS	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	
BEQ	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	
BFC										•	•	•	•	•	•	•	•	•	•			•	•	•	
BFI										•	•	•	•	•	•	•	•	•	•			•	•	•	
BGE	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	
BGT	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	
BHI	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	
BHS	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	
BIC	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	
BKPT		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	
BL	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	
BLE	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	
BLO	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	
BLS	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	
BLT	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	
BLX	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	
BMI	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	
BNE	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	
BPL	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	
BVC	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	
BVS	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	
BX	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	
BXJ					•	•	•	•	•	•	•	•	•	•	•	•	•	•	•						
CBNZ										•	•	•	•	•	•	•	•	•	•			•	•	•	
CBZ										•	•	•	•	•	•	•	•	•	•			•	•	•	
CDP										•	•	•	•	•	•	•	•	•	•			•	•	•	
CDP2										•	•	•	•	•	•	•	•	•	•			•	•	•	
CLREX						•	•	•	•	•	•	•	•	•	•	•	•	•	•			•	•	•	
CLZ		•	•							•	•	•	•	•	•	•	•	•	•			•	•	•	
CMN	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	
CMP	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	
CPS											•	•	•	•	•	•	•	•	•						

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
CPSID				●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
CPSIE				●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
CPY	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
CRC32B														●	●	●	●	●	●						
CRC32CB														●	●	●	●	●	●						
CRC32CH														●	●	●	●	●	●						
CRC32CW														●	●	●	●	●	●						
CRC32H														●	●	●	●	●	●						
CRC32W														●	●	●	●	●	●						
DBG				●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●	●		●
DCPS1														●	●	●	●	●	●						
DCPS2														●	●	●	●	●	●						
DCPS3														●	●	●	●	●	●						
DMB											●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
DSB											●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
EOR	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
ERET											●														
ESB																●	●	●	●						
FABSD											●	●	●	●	●	●	●	●	●			●	●	●	●
FABSS											●	●	●	●	●	●	●	●	●			●	●	●	●
FADDD											●	●	●	●	●	●	●	●	●			●	●	●	●
FADDS											●	●	●	●	●	●	●	●	●			●	●	●	●
FCMPD											●	●	●	●	●	●	●	●	●			●	●	●	●
FCMPED											●	●	●	●	●	●	●	●	●			●	●	●	●
FCMPES											●	●	●	●	●	●	●	●	●			●	●	●	●
FCMPS											●	●	●	●	●	●	●	●	●			●	●	●	●
FCVTDS											●	●	●	●	●	●	●	●	●			●	●	●	●
FCVTSD											●	●	●	●	●	●	●	●	●			●	●	●	●
FDIVD											●	●	●	●	●	●	●	●	●			●	●	●	●
FDIVS											●	●	●	●	●	●	●	●	●			●	●	●	●
FMDRR											●	●	●	●	●	●	●	●	●			●	●	●	●
FMRRD											●	●	●	●	●	●	●	●	●			●	●	●	●
FMRS											●	●	●	●	●	●	●	●	●			●	●	●	●
FMSR											●	●	●	●	●	●	●	●	●			●	●	●	●
FMSTAT											●	●	●	●	●	●	●	●	●			●	●	●	●
FMULD											●	●	●	●	●	●	●	●	●			●	●	●	●
FMULS											●	●	●	●	●	●	●	●	●			●	●	●	●
FNEGD											●	●	●	●	●	●	●	●	●			●	●	●	●
FNEGS											●	●	●	●	●	●	●	●	●			●	●	●	●
FSITOD											●	●	●	●	●	●	●	●	●			●	●	●	●
FSITOS											●	●	●	●	●	●	●	●	●			●	●	●	●

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8-1-A	8-2-A	8-3-A	8-4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
FSQRTD											•	•	•	•	•	•	•	•	•			•	•	•	•
FSQRTS											•	•	•	•	•	•	•	•	•			•	•	•	•
FSUBD											•	•	•	•	•	•	•	•	•			•	•	•	•
FSUBS											•	•	•	•	•	•	•	•	•			•	•	•	•
FTOSIZD											•	•	•	•	•	•	•	•	•			•	•	•	•
FTOSIZS											•	•	•	•	•	•	•	•	•			•	•	•	•
FTOUIZD											•	•	•	•	•	•	•	•	•			•	•	•	•
FTOUIZS											•	•	•	•	•	•	•	•	•			•	•	•	•
FUITOD											•	•	•	•	•	•	•	•	•			•	•	•	•
FUITOS											•	•	•	•	•	•	•	•	•			•	•	•	•
HLT														•	•	•	•	•	•						
HVC											•			•	•	•	•	•	•						
ISB											•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
IT										•	•	•	•	•	•	•	•	•	•			•	•	•	•
ITE										•	•	•	•	•	•	•	•	•	•			•	•	•	•
ITEE										•	•	•	•	•	•	•	•	•	•			•	•	•	•
ITEEE										•	•	•	•	•	•	•	•	•	•			•	•	•	•
ITEET										•	•	•	•	•	•	•	•	•	•			•	•	•	•
ITET										•	•	•	•	•	•	•	•	•	•			•	•	•	•
ITETE										•	•	•	•	•	•	•	•	•	•			•	•	•	•
ITETT										•	•	•	•	•	•	•	•	•	•			•	•	•	•
ITT										•	•	•	•	•	•	•	•	•	•			•	•	•	•
ITTE										•	•	•	•	•	•	•	•	•	•			•	•	•	•
ITTEE										•	•	•	•	•	•	•	•	•	•			•	•	•	•
ITTET										•	•	•	•	•	•	•	•	•	•			•	•	•	•
ITTT										•	•	•	•	•	•	•	•	•	•			•	•	•	•
ITTTE										•	•	•	•	•	•	•	•	•	•			•	•	•	•
ITTTT										•	•	•	•	•	•	•	•	•	•			•	•	•	•
LDA														•	•	•	•	•	•					•	•
LDAB														•	•	•	•	•	•					•	•
LDAEX														•	•	•	•	•	•					•	•
LDAEXB														•	•	•	•	•	•					•	•
LDAEXD														•	•	•	•	•	•						
LDAEXH														•	•	•	•	•	•					•	•
LDAH														•	•	•	•	•	•					•	•
LDM	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
LDMDA	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
LDMDB	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•			•	•		•
LDMEA	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•			•	•		•
LDMED	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
LDMFA	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8-1-A	8-2-A	8-3-A	8-4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
LDMFD	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
LDMIA	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
LDMIB	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
LDR	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
LDRB	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
LDRBT	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●	●		●
LDRD			●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●	●	●
LDREX				●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●	●	●
LDREXB				●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●	●	●
LDREXD											●	●	●	●	●	●	●	●	●	●					
LDREXH				●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●	●	●
LDRH	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
LDRHT										●	●	●	●	●	●	●	●	●	●			●	●		●
LDRSB	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
LDRSBT										●	●	●	●	●	●	●	●	●	●			●	●		●
LDRSH	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
LDRSHT										●	●	●	●	●	●	●	●	●	●	●			●	●	●
LDRT	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●	●	●
LSL	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
LSR	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
MCR	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●	●	●
MCR2		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●	●	●
MCRR	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●	●	●
MCRR2										●	●	●	●	●	●	●	●	●	●	●			●	●	●
MLA	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●	●	●
MLS										●	●	●	●	●	●	●	●	●	●	●			●	●	●
MOV	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
MOVS	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
MOVT										●	●	●	●	●	●	●	●	●	●	●			●	●	●
MOVW										●	●	●	●	●	●	●	●	●	●	●			●	●	●
MRC	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●	●	●
MRC2		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●	●	●
MRRC	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●	●	●
MRRC2										●	●	●	●	●	●	●	●	●	●	●			●	●	●
MRS	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●	●	●
MSR	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●	●	●
MUL	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
MVN	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
NEG	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
NOP	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
ORN										●	●	●	●	●	●	●	●	●	●	●			●	●	●

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8-1-A	8-2-A	8-3-A	8-4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
ORR	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
PKHBT											•	•	•	•	•	•	•	•	•				•		
PKHTB											•	•	•	•	•	•	•	•	•				•		
PLD										•	•	•	•	•	•	•	•	•	•			•	•		•
PLDW										•	•	•	•	•	•	•	•	•	•			•	•		
PLI										•	•	•	•	•	•	•	•	•	•			•	•		•
POP	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
PUSH	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
QADD			•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•				•		
QADD16										•	•	•	•	•	•	•	•	•	•				•		
QADD8										•	•	•	•	•	•	•	•	•	•				•		
QASX										•	•	•	•	•	•	•	•	•	•				•		
QDADD			•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•				•		
QDSUB			•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•				•		
QSAX										•	•	•	•	•	•	•	•	•	•				•		
QSUB			•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•				•		
QSUB16										•	•	•	•	•	•	•	•	•	•				•		
QSUB8										•	•	•	•	•	•	•	•	•	•				•		
RBIT										•	•	•	•	•	•	•	•	•	•			•	•		•
REV				•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
REV16				•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
REVSH				•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
RFE				•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•						
RFEDA				•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•					
RFEDB				•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•						
RFEIA				•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•						
RFEIB				•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•						
ROR	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
RRX	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•			•	•		•
RSB	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
RSC	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•				
SADD16										•	•	•	•	•	•	•	•	•	•				•		
SADD8										•	•	•	•	•	•	•	•	•	•				•		
SASX										•	•	•	•	•	•	•	•	•	•				•		
SBC	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
SBFX										•	•	•	•	•	•	•	•	•	•			•	•		•
SDIV											•	•		•	•	•	•	•	•			•	•	•	•
SEL										•	•	•	•	•	•	•	•	•	•				•	•	•
SETEND				•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
SETPAN															•	•	•	•	•					•	•
SEV						•	•	•	•		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8-1-A	8-2-A	8-3-A	8-4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SEVL																									
SHADD16																									
SHADD8																									
SHASX																									
SHSAX																									
SHSUB16																									
SHSUB8																									
SMLABB																									
SMLABT																									
SMLAD																									
SMLADX																									
SMLAL																									
SMLALBB																									
SMLALBT																									
SMLALD																									
SMLALDX																									
SMLALTB																									
SMLALTT																									
SMLATB																									
SMLATT																									
SMLAWB																									
SMLAWT																									
SMLSD																									
SMLS DX																									
SMLS LD																									
SMLS LDX																									
SMMLA																									
SMMLAR																									
SMMLS																									
SMMLSR																									
SMMUL																									
SMMULR																									
SMUAD																									
SMUADX																									
SMULBB																									
SMULBT																									
SMULL																									
SMULTB																									
SMULTT																									
SMULWB																									
SMULWT																									

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8-1-A	8-2-A	8-3-A	8-4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SMUSD										•	•	•	•	•	•	•	•	•	•				•		
SMUSDx										•	•	•	•	•	•	•	•	•	•				•		
SRS				•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•						
SRSDA				•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•						
SRSDB				•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•						
SRSIA				•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•						
SRSIB				•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•						
SSAT										•	•	•	•	•	•	•	•	•	•			•	•		•
SSAT16										•	•	•	•	•	•	•	•	•	•				•		
SSAX										•	•	•	•	•	•	•	•	•	•				•		
SSUB16										•	•	•	•	•	•	•	•	•	•				•		
SSUB8										•	•	•	•	•	•	•	•	•	•				•		
STL														•	•	•	•	•	•					•	•
STLB														•	•	•	•	•	•					•	•
STLH														•	•	•	•	•	•					•	•
STM	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
STMDA	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
STMDB	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
STMEA	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
STMED	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
STMFA	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
STMFD	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
STMIA	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
STMIB	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
STR	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
STRB	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
STRBT	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•			•	•		•
STRD			•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•			•	•		•
STREX										•	•	•	•	•	•	•	•	•	•			•	•	•	•
STREXB										•	•	•	•	•	•	•	•	•	•			•	•	•	•
STREXD										•	•	•	•	•	•	•	•	•	•						
STREXH										•	•	•	•	•	•	•	•	•	•			•	•	•	•
STRH	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
STRHT										•	•	•	•	•	•	•	•	•	•			•	•		•
STRT	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•			•	•		•
SUB	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
SUBW										•	•	•	•	•	•	•	•	•	•			•	•		•
SVC	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	•	•	•
SXTAB										•	•	•	•	•	•	•	•	•	•				•		
SXTAB16										•	•	•	•	•	•	•	•	•	•				•		
SXTAH										•	•	•	•	•	•	•	•	•	•				•		

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
SXTB				●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	
SXTB16										●	●	●	●	●	●	●	●	●	●	●			●		
SXTH				●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	
TBB										●	●	●	●	●	●	●	●	●	●			●	●	●	
TBH										●	●	●	●	●	●	●	●	●	●			●	●	●	
TEQ	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●	●	●	
TST	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	
TT																							●	●	
TTA																							●	●	
TTAT																							●	●	
TTT																							●	●	
UADD16										●	●	●	●	●	●	●	●	●	●				●		
UADD8										●	●	●	●	●	●	●	●	●	●				●		
UASX										●	●	●	●	●	●	●	●	●	●				●		
UBFX										●	●	●	●	●	●	●	●	●	●			●	●	●	
UDF	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	
UDIV											●	●		●	●	●	●	●	●			●	●	●	
UEXT16				●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	
UEXT8				●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	
UHADD16										●	●	●	●	●	●	●	●	●	●				●		
UHADD8										●	●	●	●	●	●	●	●	●	●				●		
UHASX										●	●	●	●	●	●	●	●	●	●				●		
UHSAX										●	●	●	●	●	●	●	●	●	●				●		
UHSUB16										●	●	●	●	●	●	●	●	●	●				●		
UHSUB8										●	●	●	●	●	●	●	●	●	●				●		
UMAAL										●	●	●	●	●	●	●	●	●	●				●		
UMLAL	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●	●	●	
UMULL	●	●	●							●	●	●	●	●	●	●	●	●	●			●	●	●	
UQADD16										●	●	●	●	●	●	●	●	●	●				●		
UQADD8										●	●	●	●	●	●	●	●	●	●				●		
UQASX										●	●	●	●	●	●	●	●	●	●				●		
UQSAX										●	●	●	●	●	●	●	●	●	●				●		
UQSUB16										●	●	●	●	●	●	●	●	●	●				●		
UQSUB8										●	●	●	●	●	●	●	●	●	●				●		
USAD8										●	●	●	●	●	●	●	●	●	●				●		
USADA8										●	●	●	●	●	●	●	●	●	●				●		
USAT										●	●	●	●	●	●	●	●	●	●			●	●	●	
USAT16										●	●	●	●	●	●	●	●	●	●				●		
USAX										●	●	●	●	●	●	●	●	●	●				●		
USUB16										●	●	●	●	●	●	●	●	●	●				●		
USUB8										●	●	●	●	●	●	●	●	●	●				●		

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8-1-A	8-2-A	8-3-A	8-4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
UXTAB										•	•	•	•	•	•	•	•	•	•				•		
UXTAB16										•	•	•	•	•	•	•	•	•	•				•		
UXTAH										•	•	•	•	•	•	•	•	•	•				•		
UXTB				•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
UXTB16										•	•	•	•	•	•	•	•	•	•				•		
UXTH				•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
VABS											•	•	•	•	•	•	•	•	•			•	•	•	•
VADD											•	•	•	•	•	•	•	•	•			•	•	•	•
VCMP											•	•	•	•	•	•	•	•	•			•	•	•	•
VCMPPE											•	•	•	•	•	•	•	•	•			•	•	•	•
VCVT											•	•	•	•	•	•	•	•	•			•	•	•	•
VCVTA											•	•	•	•	•	•	•	•	•			•	•	•	•
VCVTB											•	•	•	•	•	•	•	•	•			•	•	•	•
VCVTM											•	•	•	•	•	•	•	•	•			•	•	•	•
VCVTN											•	•	•	•	•	•	•	•	•			•	•	•	•
VCVTP											•	•	•	•	•	•	•	•	•			•	•	•	•
VCVTR											•	•	•	•	•	•	•	•	•			•	•	•	•
VCVTT											•	•	•	•	•	•	•	•	•			•	•	•	•
VDIV											•	•	•	•	•	•	•	•	•			•	•	•	•
VFMA											•	•	•	•	•	•	•	•	•			•	•	•	•
VFMS											•	•	•	•	•	•	•	•	•			•	•	•	•
VFNMA											•	•	•	•	•	•	•	•	•			•	•	•	•
VFNMS											•	•	•	•	•	•	•	•	•			•	•	•	•
VLDR											•	•	•	•	•	•	•	•	•			•	•	•	•
VMAXNM											•	•	•	•	•	•	•	•	•			•	•	•	•
VMINNM											•	•	•	•	•	•	•	•	•			•	•	•	•
VMLA											•	•	•	•	•	•	•	•	•			•	•	•	•
VMLS											•	•	•	•	•	•	•	•	•			•	•	•	•
VMOV											•	•	•	•	•	•	•	•	•			•	•	•	•
VMUL											•	•	•	•	•	•	•	•	•			•	•	•	•
VNEG											•	•	•	•	•	•	•	•	•			•	•	•	•
VNMLA											•	•	•	•	•	•	•	•	•			•	•	•	•
VNMLS											•	•	•	•	•	•	•	•	•			•	•	•	•
VNMUL											•	•	•	•	•	•	•	•	•			•	•	•	•
VRINTA											•	•	•	•	•	•	•	•	•			•	•	•	•
VRINTM											•	•	•	•	•	•	•	•	•			•	•	•	•
VRINTN											•	•	•	•	•	•	•	•	•			•	•	•	•
VRINTP											•	•	•	•	•	•	•	•	•			•	•	•	•
VRINTR											•	•	•	•	•	•	•	•	•			•	•	•	•
VRINTX											•	•	•	•	•	•	•	•	•			•	•	•	•
VRINTZ											•	•	•	•	•	•	•	•	•			•	•	•	•

Instruction	4T	5T	5TE	6	6J	6K	6KZ	6Z	6ZK	6T2	7VE	7-R	7-A	8-A	8.1-A	8.2-A	8.3-A	8.4-A	8-R	6-M	6S-M	7-M	7E-M	8-M.Base	8-M.Main
VSELEQ											●	●	●	●	●	●	●	●	●			●	●	●	●
VSELGE											●	●	●	●	●	●	●	●	●			●	●	●	●
VSELGT											●	●	●	●	●	●	●	●	●			●	●	●	●
VSQRT											●	●	●	●	●	●	●	●	●			●	●	●	●
VSTR											●	●	●	●	●	●	●	●	●			●	●	●	●
VSUB											●	●	●	●	●	●	●	●	●			●	●	●	●
WFE						●	●	●	●		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
WFI						●	●	●	●		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
YIELD						●	●	●	●		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●

- Arm mode only
- Thumb mode only
- Arm and Thumb modes

Chapter 8

Differences between assemblers

8.1 Syntax and Encoding differences

The following sections document the differences in UAL input syntax and the binary encoding of UAL source code. This is particularly important if you expect identical instruction encodings between assemblers, especially when using the SEGGER assembler to assemble existing source code.

8.1.1 MUL and MULS

UAL allows a multiply instruction to be written as `MUL Rd, Rn`. When the only encoding of this UAL input is a three-operand instructions, `MUL Rd, Rn` must be interpreted as `MUL Rd, Rd, Rn`.

The following are the encodings of `MUL Rd, Rn` and `MULS Rd, Rn` where only three-operand encoding is possible. The SEGGER assembler follows Arm's encoding, whereas GNU deviates and IAR rejects correct UAL.

SEGGER	Arm	GNU	IAR
<code>MUL Rd, Rd, Rn</code>	<code>MUL Rd, Rd, Rn</code>	<code>MUL Rd, Rn, Rd</code>	Rejects
<code>MULS Rd, Rd, Rn</code>	<code>MULS Rd, Rd, Rn</code>	<code>MULS Rd, Rn, Rd</code>	Rejects

8.1.2 MULS

UAL allows a multiply instruction to be written as `MULS Rd, Rn`. The assembler can choose between two-operand form and three-operand form based on the registers `Rd` and `Rn`.

The following are the encodings of `MULS R1, R2`:

SEGGER	Arm	GNU	IAR
<code>MULS.N R1, R2</code>	Rejects	<code>MULS.N R1, R2</code>	<code>MULS.N R1, R2</code>

And `MULS R1, R9`:

SEGGER	Arm	GNU	IAR
<code>MULS.W R1, R1, R9</code>	<code>MULS.W R1, R1, R9</code>	<code>MULS.W R1, R9, R1</code>	<code>MULS.W R1, R1, R9</code>

Chapter 9

Indexes

9.1 Subject index

- .2BYTE (GNU directive), 79
- .4BYTE (GNU directive), 80
- ADC (instruction), 114
- ADD (instruction), 115
- ADDW (instruction), 117
- .ALIGN (GNU directive), 86
- AND (instruction), 118
- .ARM (GNU directive), 64
- ARM (IAR directive), 99
- .ASCII (GNU directive), 77
- .ASCIZ (GNU directive), 78
- ASR (instruction), 119
- auto-import (command line), 39
- B (instruction), 120
- BCC (instruction), 121
- BCS (instruction), 122
- BEQ (instruction), 123
- BFC (instruction), 124
- BFI (instruction), 125
- BGE (instruction), 126
- BGT (instruction), 127
- BHI (instruction), 128
- BHS (instruction), 129
- BIC (instruction), 130
- BKPT (instruction), 131
- BL (instruction), 132
- BLE (instruction), 133
- BLO (instruction), 134
- BLS (instruction), 135
- BLT (instruction), 136
- BLX (instruction), 137
- BMI (instruction), 138
- BNE (instruction), 139
- BPL (instruction), 140
- .BSS (GNU directive), 68
- BVC (instruction), 141
- BVS (instruction), 142
- BX (instruction), 143
- BXJ (instruction), 144
- .BYTE (GNU directive), 77
- CBNZ (instruction), 145
- CBZ (instruction), 146
- CDP (instruction), 147
- CDP2 (instruction), 148
- CLREX (instruction), 149
- CLZ (instruction), 150
- CMN (instruction), 151
- CMP (instruction), 152
- .CODE (GNU directive), 65
- CODE (IAR directive), 100
- CODE16 (Arm directive), 93
- CODE16 (IAR directive), 101
- CODE32 (Arm directive), 94
- CODE32 (IAR directive), 99
- command line,
 - auto-import, 39
 - cpu=name, 34
 - define, 41
 - dependency-file, 23
 - include, 20
 - input-syntax, 42
 - keep-empty-sections, 23
 - keep-local-symbols, 25
 - list-diagnostics, 43
 - list-file, 45
 - list-form-feed, 46
 - list-html, 48
 - list-page-length, 49
 - list-symbols, 50
 - list-text, 52
 - mapping-symbols, 27
 - marm, 35
 - mbig-endian, 36
 - mlittle-endian, 37
 - mthumb, 38
 - no-auto-import, 40
 - no-keep-empty-sections, 24
 - no-keep-local-symbols, 26
 - no-list-diagnostics, 44
 - no-list-form-feed, 47
 - no-list-symbols, 51
 - no-mapping-symbols, 28
 - no-pad-sections, 33
 - no-remarks, 56
 - no-symbols, 31
 - no-warnings, 60
 - output, 29
 - pad-sections, 32
 - remarks, 53
 - remarks-are-errors, 54
 - remarks-are-warnings, 55
 - silent, 57
 - symbols, 30
 - verbose, 58
 - via, 21
 - warnings, 59
 - warnings-are-errors, 60
- CPS (instruction), 153
- CPSID (instruction), 154
- CPSIE (instruction), 155
- cpu=name (command line), 34
- CPY (instruction), 156
- CRC32B (instruction), 157
- CRC32CB (instruction), 158
- CRC32CH (instruction), 159
- CRC32CW (instruction), 160
- CRC32H (instruction), 161
- CRC32W (instruction), 162
- .DATA (GNU directive), 67
- DBG (instruction), 163
- DC16 (IAR directive), 103
- DC24 (IAR directive), 104
- DC32 (IAR directive), 105
- DC8 (IAR directive), 102
- DCB (IAR directive), 102
- DCD (IAR directive), 105
- DCI16 (IAR directive), 106
- DCI32 (IAR directive), 107
- DCID (IAR directive), 107
- DCIW (IAR directive), 106
- DCPS1 (instruction), 164
- DCPS2 (instruction), 165
- DCPS3 (instruction), 166
- DCW (IAR directive), 103
- define (command line), 41
- dependency-file (command line), 23
- DMB (instruction), 167
- DS (IAR directive), 108
- DS16 (IAR directive), 109
- DS24 (IAR directive), 110
- DS32 (IAR directive), 111
- DS8 (IAR directive), 108
- DSB (instruction), 168
- DSD (IAR directive), 111
- DSW (IAR directive), 109
- EOR (instruction), 169
- ERET (instruction), 170
- ESB (instruction), 171
- .EXTERN (GNU directive), 71
- FABSD (instruction), 172
- FABSS (instruction), 173
- FADDD (instruction), 174
- FADDS (instruction), 175

- FCMPD (instruction), 176
- FCMPED (instruction), 177
- FCMPES (instruction), 178
- FCMPS (instruction), 179
- FCVTDS (instruction), 180
- FCVTS (instruction), 181
- FDIVD (instruction), 182
- FDIVS (instruction), 183
- FMDRR (instruction), 184
- FMRRD (instruction), 185
- FMRS (instruction), 186
- FMSR (instruction), 187
- FMSTAT (instruction), 188
- FMULD (instruction), 189
- FMULS (instruction), 190
- FNEGD (instruction), 191
- FNEGS (instruction), 192
- FSITOD (instruction), 193
- FSITOS (instruction), 194
- FSQRTD (instruction), 195
- FSQRTS (instruction), 196
- FSUBD (instruction), 197
- FSUBS (instruction), 198
- FTOSIZD (instruction), 199
- FTOSIZS (instruction), 200
- FTOUIZD (instruction), 201
- FTOUIZS (instruction), 202
- FUITOD (instruction), 203
- FUITOS (instruction), 204
- .GLOBAL (GNU directive), 71
- .GLOBL (GNU directive), 71
- GNU syntax,
 - .2BYTE, 79
 - .4BYTE, 80
 - .ALIGN, 86
 - .ARM, 64
 - .ASCII, 77
 - .ASCIZ, 78
 - .BSS, 68
 - .BYTE, 77
 - .CODE, 65
 - .DATA, 67
 - .EXTERN, 71
 - .GLOBAL, 71
 - .GLOBL, 71
 - .HIDDEN, 74
 - .HWORD, 81
 - .INCBIN, 63
 - .INST, 83
 - .INST.N, 84
 - .INST.W, 85
 - .INT, 82
 - .INTERNAL, 76
 - .LIST, 88
 - .LOCAL, 72
 - .LONG, 82
 - .NOLIST, 89
 - .P2ALIGN, 87
 - .P2ALIGNL, 87
 - .P2ALIGNW, 87
 - .PROTECTED, 75
 - .RODATA, 70
 - .SHORT, 81
 - .STRING, 78
 - .SUBTTL, 91
 - .TEXT, 69
 - .THUMB, 66
 - .TITLE, 90
 - .WEAK, 73
 - .WORD, 82
- .HIDDEN (GNU directive), 74
- HLT (instruction), 205
- HVC (instruction), 206
- .HWORD (GNU directive), 81
- IAR syntax,
 - ARM, 99
 - CODE, 100
 - CODE16, 93, 101
 - CODE32, 94, 99
 - DC16, 103
 - DC24, 104
 - DC32, 105
 - DC8, 102
 - DCB, 102
 - DCD, 105
 - DCI16, 106
 - DCI32, 107
 - DCID, 107
 - DCIW, 106
 - DCW, 103
 - DS, 108
 - DS16, 109
 - DS24, 110
 - DS32, 111
 - DS8, 108
 - DSI, 111
 - DSW, 109
 - SUBT, 97
 - THUMB, 101
 - TTL, 96
- .INCBIN (GNU directive), 63
- include (command line), 20
- input-syntax (command line), 42
- .INST (GNU directive), 83
- .INST.N (GNU directive), 84
- .INST.W (GNU directive), 85
- instruction,
 - ADC, 114
 - ADD, 115
 - ADDW, 117
 - AND, 118
 - ASR, 119
 - B, 120
 - BCC, 121
 - BCS, 122
 - BEQ, 123
 - BFC, 124
 - BFI, 125
 - BGE, 126
 - BGT, 127
 - BHI, 128
 - BHS, 129
 - BIC, 130
 - BKPT, 131
 - BL, 132
 - BLE, 133
 - BLO, 134
 - BLS, 135
 - BLT, 136
 - BLX, 137
 - BMI, 138
 - BNE, 139
 - BPL, 140
 - BVC, 141
 - BVS, 142
 - BX, 143
 - BXJ, 144
 - CBNZ, 145
 - CBZ, 146
 - CDP, 147
 - CDP2, 148
 - CLREX, 149
 - CLZ, 150
 - CMN, 151
 - CMP, 152
 - CPS, 153
 - CPSID, 154
 - CPSIE, 155
 - CPY, 156
 - CRC32B, 157

CRC32CB,	158	LDMED,	234
CRC32CH,	159	LDMFA,	235
CRC32CW,	160	LDMFD,	236
CRC32H,	161	LDMIA,	237
CRC32W,	162	LDMIB,	238
DBG,	163	LDR,	239
DCPS1,	164	LDRB,	240
DCPS2,	165	LDRBT,	241
DCPS3,	166	LDRD,	242
DMB,	167	LDREX,	243
DSB,	168	LDREXB,	244
EOR,	169	LDREXD,	245
ERET,	170	LDREXH,	246
ESB,	171	LDRH,	247
FABSD,	172	LDRHT,	248
FABSS,	173	LDRSB,	249
FADDD,	174	LDRSBT,	250
FADDS,	175	LDRSH,	251
FCMPD,	176	LDRSHT,	252
FCMPED,	177	LDRT,	253
FCMPES,	178	LSL,	254
FCMPS,	179	LSR,	255
FCVTDS,	180	MCR,	256
FCVTSD,	181	MCR2,	257
FDIVD,	182	MCRR,	258
FDIVS,	183	MCRR2,	259
FMDRR,	184	MLA,	260
FMRRD,	185	MLS,	261
FMRS,	186	MOV,	262
FMSR,	187	MOVS,	263
FMSTAT,	188	MOVT,	264
FMULD,	189	MOVW,	265
FMULS,	190	MRC,	266
FNEGD,	191	MRC2,	267
FNEGS,	192	MRRC,	268
FSITOD,	193	MRRC2,	269
FSITOS,	194	MRS,	270
FSQRTD,	195	MSR,	271
FSQRTS,	196	MUL,	272
FSUBD,	197	MVN,	273
FSUBS,	198	NEG,	274
FTOSIZD,	199	NOP,	275
FTOSIZS,	200	ORN,	276
FTOUIZD,	201	ORR,	277
FTOUIZS,	202	PKHBT,	278
FUITOD,	203	PKHTB,	279
FUITOS,	204	PLD,	280
HLT,	205	PLDW,	281
HVC,	206	PLI,	282
ISB,	207	POP,	283
IT,	208	PUSH,	284
ITE,	209	QADD,	285
ITEE,	210	QADD16,	286
ITEEE,	211	QADD8,	287
ITEET,	212	QASX,	288
ITET,	213	QDADD,	289
ITETE,	214	QDSUB,	290
ITETT,	215	QSAX,	291
ITT,	216	QSUB,	292
ITTE,	217	QSUB16,	293
ITTEE,	218	QSUB8,	294
ITTET,	219	RBIT,	295
ITTT,	220	REV,	296
ITTTTE,	221	REV16,	297
ITTTT,	222	REVSH,	298
LDA,	223	RFE,	299
LDAB,	224	RFEDA,	300
LDAEX,	225	RFEDB,	301
LDAEXB,	226	RFEIA,	302
LDAEXD,	227	RFEIB,	303
LDAEXH,	228	ROR,	304
LDAH,	229	RRX,	305
LDM,	230	RSB,	306
LDMDA,	231	RSC,	307
LDMDB,	232	SADD16,	308
LDMEA,	233	SADD8,	309

SASX, 310
 SBC, 311
 SBFX, 312
 SDIV, 313
 SEL, 314
 SETEND, 315
 SETPAN, 316
 SEV, 317
 SEVL, 318
 SHADD16, 319
 SHADD8, 320
 SHASX, 321
 SHSAX, 322
 SHSUB16, 323
 SHSUB8, 324
 SMLABB, 325
 SMLABT, 326
 SMLAD, 327
 SMLADX, 328
 SMLAL, 329
 SMLALBB, 330
 SMLALBT, 331
 SMLALD, 332
 SMLALDX, 333
 SMLALTB, 334
 SMLALTT, 335
 SMLATB, 336
 SMLATT, 337
 SMLAWB, 338
 SMLAWT, 339
 SMLSD, 340
 SMLSDX, 341
 SMLSLD, 342
 SMLSLDX, 343
 SMMLA, 344
 SMMLAR, 345
 SMMLS, 346
 SMMLSR, 347
 SMMUL, 348
 SMMULR, 349
 SMUAD, 350
 SMUADX, 351
 SMULBB, 352
 SMULBT, 353
 SMULL, 354
 SMULTB, 355
 SMULTT, 356
 SMULWB, 357
 SMULWT, 358
 SMUSD, 359
 SMUSDX, 360
 SRS, 361
 SRSDA, 362
 SRSDb, 363
 SRSIA, 364
 SRSIB, 365
 SSAT, 366
 SSAT16, 367
 SSAX, 368
 SSUB16, 369
 SSUB8, 370
 STL, 371
 STLB, 372
 STLH, 373
 STM, 374
 STMDA, 375
 STMDB, 376
 STMEA, 377
 STMED, 378
 STMFA, 379
 STMFD, 380
 STMIA, 381
 STMIB, 382
 STR, 383
 STRB, 384
 STRBT, 385
 STRD, 386
 STREX, 387
 STREXB, 388
 STREXD, 389
 STREXH, 390
 STRH, 391
 STRHT, 392
 STRT, 393
 SUB, 394
 SUBW, 395
 SVC, 396
 SXTAB, 397
 SXTAB16, 398
 SXTAH, 399
 SXTB, 400
 SXTB16, 401
 SXTH, 402
 TBB, 403
 TBH, 404
 TEQ, 405
 TST, 406
 TT, 407
 TTA, 408
 TTAT, 409
 TTT, 410
 UADD16, 411
 UADD8, 412
 UASX, 413
 UBFX, 414
 UDF, 415
 UDIV, 416
 UEXT16, 417
 UEXT8, 418
 UHADD16, 419
 UHADD8, 420
 UHASX, 421
 UHSAX, 422
 UHSUB16, 423
 UHSUB8, 424
 UMAAL, 425
 UMLAL, 426
 UMULL, 427
 UQADD16, 428
 UQADD8, 429
 UQASX, 430
 UQSAX, 431
 UQSUB16, 432
 UQSUB8, 433
 USAD8, 434
 USADA8, 435
 USAT, 436
 USAT16, 437
 USAX, 438
 USUB16, 439
 USUB8, 440
 UXTAB, 441
 UXTAB16, 442
 UXTAH, 443
 UXTB, 444
 UXTB16, 445
 UXTH, 446
 VABS, 447
 VADD, 448
 VCMPE, 449
 VCMPE, 450
 VCVT, 451
 VCVTA, 452
 VCVTB, 453
 VCVTM, 454
 VCVTN, 455
 VCVTP, 456
 VCVTR, 457
 VCVTT, 458
 VDIV, 459
 VFMA, 460
 VFMS, 461

- VFNMA, 462
- VFNMS, 463
- VLDR, 464
- VMAXNM, 465
- VMINNM, 466
- VMLA, 467
- VMLS, 468
- VMOV, 469
- VMUL, 470
- VNEG, 471
- VNMLA, 472
- VNMLS, 473
- VNMUL, 474
- VRINTA, 475
- VRINTM, 476
- VRINTN, 477
- VRINTP, 478
- VRINTR, 479
- VRINTX, 480
- VRINTZ, 481
- VSELEQ, 482
- VSELGE, 483
- VSELGT, 484
- VSQRT, 485
- VSTR, 486
- VSUB, 487
- WFE, 488
- WFI, 489
- YIELD, 490
- .INT (GNU directive), 82
- .INTERNAL (GNU directive), 76
- ISB (instruction), 207
- IT (instruction), 208
- ITE (instruction), 209
- ITEE (instruction), 210
- ITEEE (instruction), 211
- ITEET (instruction), 212
- ITET (instruction), 213
- ITETE (instruction), 214
- ITETT (instruction), 215
- ITT (instruction), 216
- ITTE (instruction), 217
- ITTEE (instruction), 218
- ITTET (instruction), 219
- ITTT (instruction), 220
- ITTTE (instruction), 221
- ITTTT (instruction), 222
- keep-empty-sections (command line), 23
- keep-local-symbols (command line), 25
- LDA (instruction), 223
- LDAB (instruction), 224
- LDAEX (instruction), 225
- LDAEXB (instruction), 226
- LDAEXD (instruction), 227
- LDAEXH (instruction), 228
- LDAH (instruction), 229
- LDM (instruction), 230
- LDMDA (instruction), 231
- LDMDB (instruction), 232
- LDMEA (instruction), 233
- LDMED (instruction), 234
- LDMFA (instruction), 235
- LDMFD (instruction), 236
- LDMIA (instruction), 237
- LDMIB (instruction), 238
- LDR (instruction), 239
- LDRB (instruction), 240
- LDRBT (instruction), 241
- LDRD (instruction), 242
- LDREX (instruction), 243
- LDREXB (instruction), 244
- LDREXD (instruction), 245
- LDREXH (instruction), 246
- LDRH (instruction), 247
- LDRHT (instruction), 248
- LDRSB (instruction), 249
- LDRSBT (instruction), 250
- LDRSH (instruction), 251
- LDRSHT (instruction), 252
- LDRT (instruction), 253
- .LIST (GNU directive), 88
- list-diagnostics (command line), 43
- list-file (command line), 45
- list-form-feed (command line), 46
- list-html (command line), 48
- list-page-length (command line), 49
- list-symbols (command line), 50
- list-text (command line), 52
- .LOCAL (GNU directive), 72
- .LONG (GNU directive), 82
- LSL (instruction), 254
- LSR (instruction), 255
- mapping-symbols (command line), 27
- marm (command line), 35
- mbig-endian (command line), 36
- MCR (instruction), 256
- MCR2 (instruction), 257
- MCRR (instruction), 258
- MCRR2 (instruction), 259
- MLA (instruction), 260
- mlittle-endian (command line), 37
- MLS (instruction), 261
- MOV (instruction), 262
- MOVS (instruction), 263
- MOVT (instruction), 264
- MOVW (instruction), 265
- MRC (instruction), 266
- MRC2 (instruction), 267
- MRRC (instruction), 268
- MRRC2 (instruction), 269
- MRS (instruction), 270
- MSR (instruction), 271
- mthumb (command line), 38
- MUL (instruction), 272
- MVN (instruction), 273
- NEG (instruction), 274
- no-auto-import (command line), 40
- no-keep-empty-sections (command line), 24
- no-keep-local-symbols (command line), 26
- no-list-diagnostics (command line), 44
- no-list-form-feed (command line), 47
- no-list-symbols (command line), 51
- no-mapping-symbols (command line), 28
- no-pad-sections (command line), 33
- no-remarks (command line), 56
- no-symbols (command line), 31
- no-warnings (command line), 60
- .NOLIST (GNU directive), 89
- NOP (instruction), 275
- ORN (instruction), 276
- ORR (instruction), 277
- output (command line), 29
- .P2ALIGN (GNU directive), 87
- .P2ALIGNL (GNU directive), 87
- .P2ALIGNW (GNU directive), 87
- pad-sections (command line), 32
- PKHBT (instruction), 278
- PKHTB (instruction), 279
- PLD (instruction), 280
- PLDW (instruction), 281
- PLI (instruction), 282
- POP (instruction), 283
- .PROTECTED (GNU directive), 75
- PUSH (instruction), 284
- QADD (instruction), 285
- QADD16 (instruction), 286
- QADD8 (instruction), 287

- QASX (instruction), 288
- QDADD (instruction), 289
- QDSUB (instruction), 290
- QSAX (instruction), 291
- QSUB (instruction), 292
- QSUB16 (instruction), 293
- QSUB8 (instruction), 294
- RBIT (instruction), 295
- remarks (command line), 53
- remarks-are-errors (command line), 54
- remarks-are-warnings (command line), 55
- REV (instruction), 296
- REV16 (instruction), 297
- REVSH (instruction), 298
- RFE (instruction), 299
- RFEDA (instruction), 300
- RFEDB (instruction), 301
- RFEIA (instruction), 302
- RFEIB (instruction), 303
- .RODATA (GNU directive), 70
- ROR (instruction), 304
- RRX (instruction), 305
- RSB (instruction), 306
- RSC (instruction), 307
- SADD16 (instruction), 308
- SADD8 (instruction), 309
- SASX (instruction), 310
- SBC (instruction), 311
- SBFX (instruction), 312
- SDIV (instruction), 313
- SEL (instruction), 314
- SETEND (instruction), 315
- SETPAN (instruction), 316
- SEV (instruction), 317
- SEVL (instruction), 318
- SHADD16 (instruction), 319
- SHADD8 (instruction), 320
- SHASX (instruction), 321
- .SHORT (GNU directive), 81
- SHSAX (instruction), 322
- SHSUB16 (instruction), 323
- SHSUB8 (instruction), 324
- silent (command line), 57
- SMLABB (instruction), 325
- SMLABT (instruction), 326
- SMLAD (instruction), 327
- SMLADX (instruction), 328
- SMLAL (instruction), 329
- SMLALBB (instruction), 330
- SMLALBT (instruction), 331
- SMLALD (instruction), 332
- SMLALDX (instruction), 333
- SMLALTB (instruction), 334
- SMLALTT (instruction), 335
- SMLATB (instruction), 336
- SMLATT (instruction), 337
- SMLAWB (instruction), 338
- SMLAWT (instruction), 339
- SMLSDB (instruction), 340
- SMLSDB (instruction), 341
- SMLSDB (instruction), 342
- SMLSDB (instruction), 343
- SMLSLDX (instruction), 344
- SMMLA (instruction), 345
- SMMLAR (instruction), 346
- SMMLS (instruction), 347
- SMMLSR (instruction), 348
- SMMUL (instruction), 349
- SMMULR (instruction), 350
- SMUAD (instruction), 351
- SMUADX (instruction), 352
- SMULBB (instruction), 353
- SMULBT (instruction), 354
- SMULL (instruction), 355
- SMULTB (instruction), 356
- SMULTT (instruction), 357
- SMULWB (instruction), 358
- SMULWT (instruction), 359
- SMUSD (instruction), 360
- SMUSD (instruction), 361
- SRSDA (instruction), 362
- SRSDB (instruction), 363
- SRSIA (instruction), 364
- SRSIB (instruction), 365
- SSAT (instruction), 366
- SSAT16 (instruction), 367
- SSAX (instruction), 368
- SSUB16 (instruction), 369
- SSUB8 (instruction), 370
- STL (instruction), 371
- STLB (instruction), 372
- STLH (instruction), 373
- STM (instruction), 374
- STMDA (instruction), 375
- STMDB (instruction), 376
- STMEA (instruction), 377
- STMED (instruction), 378
- STMFA (instruction), 379
- STMFD (instruction), 380
- STMIA (instruction), 381
- STMIB (instruction), 382
- STR (instruction), 383
- STRB (instruction), 384
- STRBT (instruction), 385
- STRD (instruction), 386
- STREX (instruction), 387
- STREXB (instruction), 388
- STREXD (instruction), 389
- STREXH (instruction), 390
- STRH (instruction), 391
- STRHT (instruction), 392
- .STRING (GNU directive), 78
- STRT (instruction), 393
- SUB (instruction), 394
- SUBT (Arm directive), 97
- .SUBTTL (GNU directive), 91
- SUBW (instruction), 395
- SVC (instruction), 396
- SXTAB (instruction), 397
- SXTAB16 (instruction), 398
- SXTAH (instruction), 399
- SXTB (instruction), 400
- SXTB16 (instruction), 401
- SXTH (instruction), 402
- symbols (command line), 30
- TBB (instruction), 403
- TBH (instruction), 404
- TEQ (instruction), 405
- .TEXT (GNU directive), 69
- .THUMB (GNU directive), 66
- THUMB (IAR directive), 101
- .TITLE (GNU directive), 90
- TST (instruction), 406
- TT (instruction), 407
- TTA (instruction), 408
- TTAT (instruction), 409
- TTL (Arm directive), 96
- TTT (instruction), 410
- UADD16 (instruction), 411
- UADD8 (instruction), 412
- UASX (instruction), 413
- UBFX (instruction), 414
- UDF (instruction), 415
- UDIV (instruction), 416
- UEXT16 (instruction), 417
- UEXT8 (instruction), 418
- UHADD16 (instruction), 419
- UHADD8 (instruction), 420
- UHASX (instruction), 421
- UHSAX (instruction), 422

UHSUB16 (instruction), 423
UHSUB8 (instruction), 424
UMAAL (instruction), 425
UMLAL (instruction), 426
UMULL (instruction), 427
UQADD16 (instruction), 428
UQADD8 (instruction), 429
UQASX (instruction), 430
UQSAX (instruction), 431
UQSUB16 (instruction), 432
UQSUB8 (instruction), 433
USAD8 (instruction), 434
USADA8 (instruction), 435
USAT (instruction), 436
USAT16 (instruction), 437
USAX (instruction), 438
USUB16 (instruction), 439
USUB8 (instruction), 440
UXTAB (instruction), 441
UXTAB16 (instruction), 442
UXTAH (instruction), 443
UXTB (instruction), 444
UXTB16 (instruction), 445
UXTH (instruction), 446

VABS (instruction), 447
VADD (instruction), 448
VCMP (instruction), 449
VCMPE (instruction), 450
VCVT (instruction), 451
VCVTA (instruction), 452
VCVTB (instruction), 453
VCVTM (instruction), 454
VCVTN (instruction), 455
VCVTP (instruction), 456
VCVTR (instruction), 457
VCVTT (instruction), 458
VDIV (instruction), 459
--verbose (command line), 58
VFMA (instruction), 460
VFMS (instruction), 461
VFNMA (instruction), 462
VFNMS (instruction), 463
--via (command line), 21
VLDLDR (instruction), 464
VMAXNM (instruction), 465
VMINNM (instruction), 466
VMLA (instruction), 467
VMLS (instruction), 468
VMOV (instruction), 469
VMUL (instruction), 470
VNEG (instruction), 471
VNMLA (instruction), 472
VNMLS (instruction), 473
VNMUL (instruction), 474
VRINTA (instruction), 475
VRINTM (instruction), 476
VRINTN (instruction), 477
VRINTP (instruction), 478
VRINTR (instruction), 479
VRINTX (instruction), 480
VRINTZ (instruction), 481
VSELEQ (instruction), 482
VSELGE (instruction), 483
VSELGT (instruction), 484
VSQRT (instruction), 485
VSTR (instruction), 486
VSUB (instruction), 487

--warnings (command line), 59
--warnings-are-errors (command line), 60
.WEAK (GNU directive), 73
WFE (instruction), 488
WFI (instruction), 489
.WORD (GNU directive), 82

YIELD (instruction), 490