Chapter C4
A64 Instruction Set Encoding

This chapter describes the encoding of the A64 instruction set. It contains the following section:

- *A64 instruction set encoding* on page C4-280.

In this chapter:

- In the decode tables, an entry of - for a field value means the value of the field does not affect the decoding.
- In the decode diagrams, a shaded field indicates that the bits in that field are not used in that level of decode.

C4.1 A64 instruction set encoding

The A64 instruction encoding is:

Table C4-1 Main encoding table for the A64 instruction set

Decode fields op0	Decode group or instruction page
0000	*Reserved* on page C4-280
0001	Unallocated.
0010	SVE Instructions. See *The Scalable Vector Extension (SVE)* on page A2-109
0011	Unallocated.
100x	*Data Processing -- Immediate* on page C4-280
101x	*Branches, Exception Generating and System instructions* on page C4-285
x1x0	*Loads and Stores* on page C4-295
x101	*Data Processing -- Register* on page C4-328
x111	*Data Processing -- Scalar Floating-Point and Advanced SIMD* on page C4-338

C4.1.1 Reserved

This section describes the encoding of the Reserved group. The encodings in this section are decoded from *A64 instruction set encoding*.

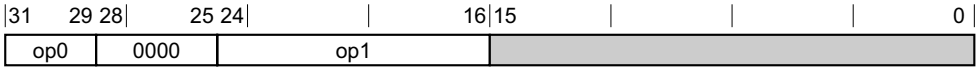


Table C4-2 Encoding table for the Reserved group

Decode fields op0	 op1	Decode group or instruction page
000	000000000	UDF
000	0001xxxxx	Unallocated.
!= 000	-	Unallocated.

C4.1.2 Data Processing -- Immediate

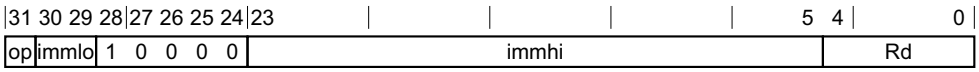
This section describes the encoding of the Data Processing -- Immediate group. The encodings in this section are decoded from *A64 instruction set encoding*.

Table C4-3 Encoding table for the Data Processing -- Immediate group

Decode fields op0	Decode group or instruction page
00x	*PC-rel. addressing*
010	*Add/subtract (immediate)*
011	*Add/subtract (immediate, with tags)* on page C4-282
100	*Logical (immediate)* on page C4-282
101	*Move wide (immediate)* on page C4-283
110	*Bitfield* on page C4-284
111	*Extract* on page C4-284

PC-rel. addressing

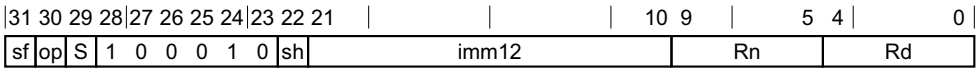
This section describes the encoding of the PC-rel. addressing instruction class. The encodings in this section are decoded from *Data Processing -- Immediate* on page C4-280.



Decode fields op	Instruction page
0	ADR
1	ADRP

Add/subtract (immediate)

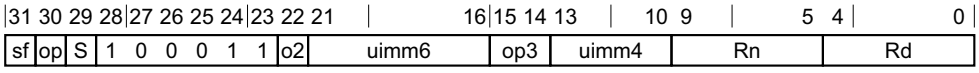
This section describes the encoding of the Add/subtract (immediate) instruction class. The encodings in this section are decoded from *Data Processing -- Immediate* on page C4-280.



Decode fields			Instruction page
sf	op	S	
0	0	0	ADD (immediate) - 32-bit variant
0	0	1	ADDS (immediate) - 32-bit variant
0	1	0	SUB (immediate) - 32-bit variant
0	1	1	SUBS (immediate) - 32-bit variant
1	0	0	ADD (immediate) - 64-bit variant
1	0	1	ADDS (immediate) - 64-bit variant
1	1	0	SUB (immediate) - 64-bit variant
1	1	1	SUBS (immediate) - 64-bit variant

Add/subtract (immediate, with tags)

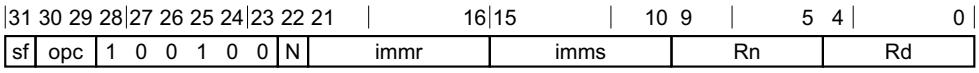
This section describes the encoding of the Add/subtract (immediate, with tags) instruction class. The encodings in this section are decoded from *Data Processing -- Immediate* on page C4-280.



Decode fields				Instruction page	Feature
sf	op	S	o2		
-	-	-	1	Unallocated.	-
0	-	-	0	Unallocated.	-
1	-	1	0	Unallocated.	-
1	0	0	0	ADDG	FEAT_MTE
1	1	0	0	SUBG	FEAT_MTE

Logical (immediate)

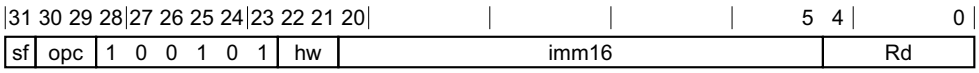
This section describes the encoding of the Logical (immediate) instruction class. The encodings in this section are decoded from *Data Processing -- Immediate* on page C4-280.



Decode fields			Instruction page
sf	opc	N	
0	-	1	Unallocated.
0	00	0	AND (immediate) - 32-bit variant
0	01	0	ORR (immediate) - 32-bit variant
0	10	0	EOR (immediate) - 32-bit variant
0	11	0	ANDS (immediate) - 32-bit variant
1	00	-	AND (immediate) - 64-bit variant
1	01	-	ORR (immediate) - 64-bit variant
1	10	-	EOR (immediate) - 64-bit variant
1	11	-	ANDS (immediate) - 64-bit variant

Move wide (immediate)

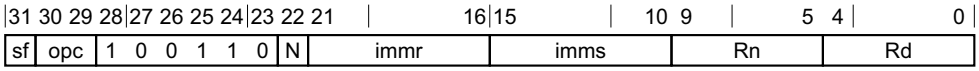
This section describes the encoding of the Move wide (immediate) instruction class. The encodings in this section are decoded from *Data Processing -- Immediate* on page C4-280.



Decode fields			Instruction page
sf	opc	hw	
-	01	-	Unallocated.
0	-	1x	Unallocated.
0	00	0x	MOVN - 32-bit variant
0	10	0x	MOVZ - 32-bit variant
0	11	0x	MOVK - 32-bit variant
1	00	-	MOVN - 64-bit variant
1	10	-	MOVZ - 64-bit variant
1	11	-	MOVK - 64-bit variant

Bitfield

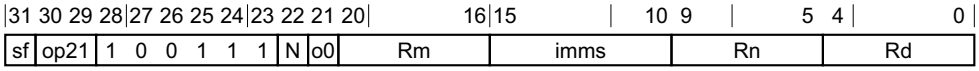
This section describes the encoding of the Bitfield instruction class. The encodings in this section are decoded from *Data Processing -- Immediate* on page C4-280.



Decode fields			Instruction page
sf	opc	N	
-	11	-	Unallocated.
0	-	1	Unallocated.
0	00	0	SBFM - 32-bit variant
0	01	0	BFM - 32-bit variant
0	10	0	UBFM - 32-bit variant
1	-	0	Unallocated.
1	00	1	SBFM - 64-bit variant
1	01	1	BFM - 64-bit variant
1	10	1	UBFM - 64-bit variant

Extract

This section describes the encoding of the Extract instruction class. The encodings in this section are decoded from *Data Processing -- Immediate* on page C4-280.



Decode fields					Instruction page
sf	op21	N	o0	imms	
-	x1	-	-	-	Unallocated.
-	00	-	1	-	Unallocated.
-	1x	-	-	-	Unallocated.
0	-	-	-	1xxxxx	Unallocated.
0	-	1	-	-	Unallocated.
0	00	0	0	0xxxxx	EXTR - 32-bit variant
1	-	0	-	-	Unallocated.
1	00	1	0	-	EXTR - 64-bit variant

C4.1.3 Branches, Exception Generating and System instructions

This section describes the encoding of the Branches, Exception Generating and System instructions group. The encodings in this section are decoded from *A64 instruction set encoding* on page C4-280.

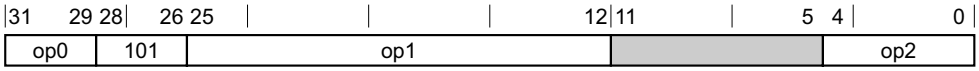


Table C4-4 Encoding table for the Branches, Exception Generating and System instructions group

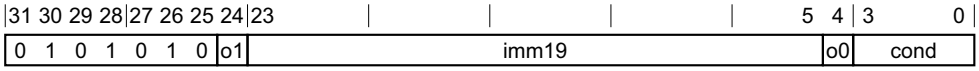
Decode fields			Decode group or instruction page
op0	op1	op2	
010	0xxxxxxxxxxxx	-	*Conditional branch (immediate)* on page C4-286
010	1xxxxxxxxxxxx	-	Unallocated.
110	00xxxxxxxxxxx	-	*Exception generation* on page C4-286
110	010000000x000x	-	Unallocated.
110	010000000x001x	-	Unallocated.
110	0100000010000x	-	Unallocated.
110	0100000010001x	-	Unallocated.
110	01000000110000	-	Unallocated.
110	01000000110001	-	*System instructions with register argument* on page C4-287
110	01000000110010	11111	*Hints* on page C4-288
110	01000000110010	!= 11111	Unallocated.
110	01000000110011	-	*Barriers* on page C4-289
110	01000001xx000x	-	Unallocated.
110	01000001xx001x	-	Unallocated.
110	0100000xxx0100	-	*PSTATE* on page C4-289
110	0100000xxx0101	-	Unallocated.
110	0100000xxx011x	-	Unallocated.
110	0100000xxx1xxx	-	Unallocated.
110	0100x01xxxxxxx	-	*System instructions* on page C4-290
110	0100x1xxxxxxxx	-	*System register move* on page C4-290
110	0101xxxxxxxxx	-	Unallocated.
110	011xxxxxxxxxx	-	Unallocated.
110	1xxxxxxxxxxxx	-	*Unconditional branch (register)* on page C4-291
x00	-	-	*Unconditional branch (immediate)* on page C4-294

Table C4-4 Encoding table for the Branches, Exception Generating and System instructions group (continued)

Decode fields			Decode group or instruction page
op0	op1	op2	
x01	0xxxxxxxxxxxx	-	*Compare and branch (immediate)* on page C4-294
x01	1xxxxxxxxxxxx	-	*Test and branch (immediate)* on page C4-294
x11	-	-	Unallocated.

Conditional branch (immediate)

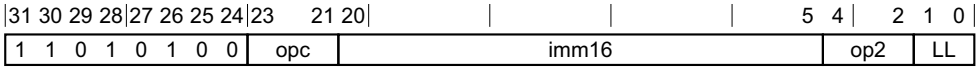
This section describes the encoding of the Conditional branch (immediate) instruction class. The encodings in this section are decoded from *Branches, Exception Generating and System instructions* on page C4-285.



Decode fields		Instruction page
o1	o0	
0	0	B.cond
0	1	Unallocated.
1	-	Unallocated.

Exception generation

This section describes the encoding of the Exception generation instruction class. The encodings in this section are decoded from *Branches, Exception Generating and System instructions* on page C4-285.

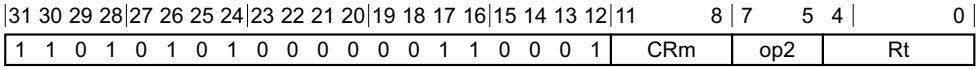


Decode fields			Instruction page
opc	op2	LL	
-	001	-	Unallocated.
-	01x	-	Unallocated.
-	1xx	-	Unallocated.
000	000	00	Unallocated.
000	000	01	SVC
000	000	10	HVC

Decode fields			Instruction page
opc	op2	LL	
000	000	11	SMC
001	000	x1	Unallocated.
001	000	00	BRK
001	000	1x	Unallocated.
010	000	x1	Unallocated.
010	000	00	HLT
010	000	1x	Unallocated.
011	000	01	Unallocated.
011	000	1x	Unallocated.
100	000	-	Unallocated.
101	000	00	Unallocated.
101	000	01	DCPS1
101	000	10	DCPS2
101	000	11	DCPS3
110	000	-	Unallocated.
111	000	-	Unallocated.

System instructions with register argument

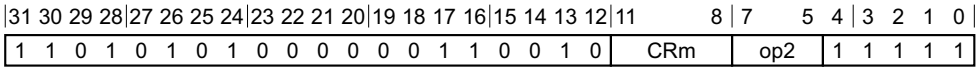
This section describes the encoding of the System instructions with register argument instruction class. The encodings in this section are decoded from *Branches, Exception Generating and System instructions* on page C4-285.



Decode fields		Instruction page	Feature
CRm	op2		
!= 0000	-	Unallocated.	-
0000	000	WFET	FEAT_WFxT
0000	001	WFIT	FEAT_WFxT
0000	01x	Unallocated.	-
0000	1xx	Unallocated.	-

Hints

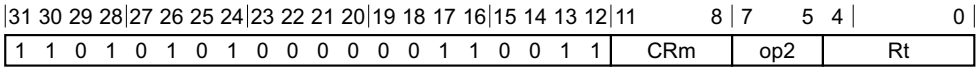
This section describes the encoding of the Hints instruction class. The encodings in this section are decoded from *Branches, Exception Generating and System instructions* on page C4-285.



Decode fields		Instruction page	Feature
CRm	op2		
-	-	HINT	-
0000	000	NOP	-
0000	001	YIELD	-
0000	010	WFE	-
0000	011	WFI	-
0000	100	SEV	-
0000	101	SEVL	-
0000	110	DGH	FEAT_DGH
0000	111	XPACD, XPACI, XPACLRI	FEAT_PAuth
0001	000	PACIA, PACIA1716, PACIASP, PACIAZ, PACIZA - PACIA1716 variant	FEAT_PAuth
0001	010	PACIB, PACIB1716, PACIBSP, PACIBZ, PACIZB - PACIB1716 variant	FEAT_PAuth
0001	100	AUTIA, AUTIA1716, AUTIASP, AUTIAZ, AUTIZA - AUTIA1716 variant	FEAT_PAuth
0001	110	AUTIB, AUTIB1716, AUTIBSP, AUTIBZ, AUTIZB - AUTIB1716 variant	FEAT_PAuth
0010	000	ESB	FEAT_RAS
0010	001	PSB CSYNC	FEAT_SPE
0010	010	TSB CSYNC	FEAT_TRF
0010	100	CSDB	-
0011	000	PACIA, PACIA1716, PACIASP, PACIAZ, PACIZA - PACIAZ variant	FEAT_PAuth
0011	001	PACIA, PACIA1716, PACIASP, PACIAZ, PACIZA - PACIASP variant	FEAT_PAuth
0011	010	PACIB, PACIB1716, PACIBSP, PACIBZ, PACIZB - PACIBZ variant	FEAT_PAuth
0011	011	PACIB, PACIB1716, PACIBSP, PACIBZ, PACIZB - PACIBSP variant	FEAT_PAuth
0011	100	AUTIA, AUTIA1716, AUTIASP, AUTIAZ, AUTIZA - AUTIAZ variant	FEAT_PAuth
0011	101	AUTIA, AUTIA1716, AUTIASP, AUTIAZ, AUTIZA - AUTIASP variant	FEAT_PAuth
0011	110	AUTIB, AUTIB1716, AUTIBSP, AUTIBZ, AUTIZB - AUTIBZ variant	FEAT_PAuth
0011	111	AUTIB, AUTIB1716, AUTIBSP, AUTIBZ, AUTIZB - AUTIBSP variant	FEAT_PAuth
0100	xx0	BTI	FEAT_BTI

Barriers

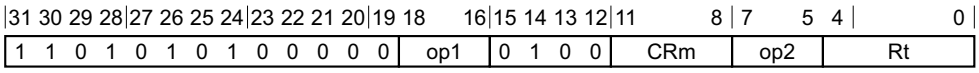
This section describes the encoding of the Barriers instruction class. The encodings in this section are decoded from *Branches, Exception Generating and System instructions* on page C4-285.



Decode fields			Instruction page	Feature
CRm	op2	Rt		
-	000	-	Unallocated.	-
-	001	!= 11111	Unallocated.	-
-	010	11111	CLREX	-
-	101	11111	DMB	-
-	110	11111	ISB	-
-	111	!= 11111	Unallocated.	-
-	111	11111	SB	-
xx0x	001	11111	Unallocated.	-
xx10	001	11111	DSB - Encoding	FEAT_XS
xx11	001	11111	Unallocated.	-
!= 0x00	100	11111	DSB - Encoding	-
0000	100	11111	SSBB	-
0001	011	-	Unallocated.	-
001x	011	-	Unallocated.	-
01xx	011	-	Unallocated.	-
0100	100	11111	PSSBB	-
1xxx	011	-	Unallocated.	-

PSTATE

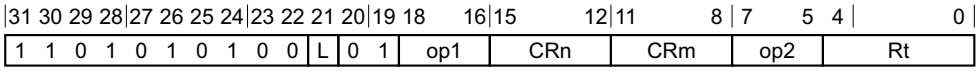
This section describes the encoding of the PSTATE instruction class. The encodings in this section are decoded from *Branches, Exception Generating and System instructions* on page C4-285.



Decode fields			Instruction page	Feature
op1	op2	Rt		
-	-	!= 11111	Unallocated.	-
-	-	11111	MSR (immediate)	-
000	000	11111	CFINV	FEAT_FlagM
000	001	11111	XAFLAG	FEAT_FlagM2
000	010	11111	AXFLAG	FEAT_FlagM2

System instructions

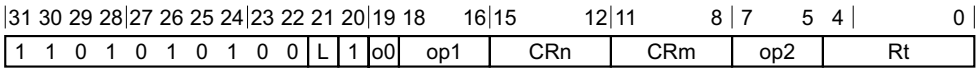
This section describes the encoding of the System instructions instruction class. The encodings in this section are decoded from *Branches, Exception Generating and System instructions* on page C4-285.



Decode fields	Instruction page
L	
0	SYS
1	SYSL

System register move

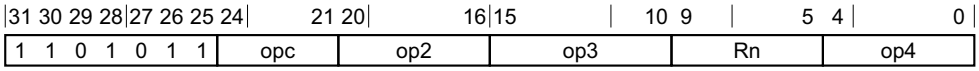
This section describes the encoding of the System register move instruction class. The encodings in this section are decoded from *Branches, Exception Generating and System instructions* on page C4-285.



Decode fields	Instruction page
L	
0	MSR (register)
1	MRS

Unconditional branch (register)

This section describes the encoding of the Unconditional branch (register) instruction class. The encodings in this section are decoded from *Branches, Exception Generating and System instructions* on page C4-285.



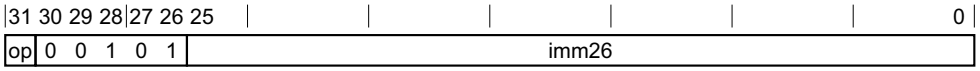
Decode fields					Instruction page	Feature
opc	op2	op3	Rn	op4		
-	!= 11111	-	-	-	Unallocated.	-
0000	11111	000000	-	!= 00000	Unallocated.	-
0000	11111	000000	-	00000	BR	-
0000	11111	000001	-	-	Unallocated.	-
0000	11111	000010	-	!= 11111	Unallocated.	-
0000	11111	000010	-	11111	BRAA, BRAAZ, BRAB, BRABZ - Key A, zero modifier variant	FEAT_PAuth
0000	11111	000011	-	!= 11111	Unallocated.	-
0000	11111	000011	-	11111	BRAA, BRAAZ, BRAB, BRABZ - Key B, zero modifier variant	FEAT_PAuth
0000	11111	0001xx	-	-	Unallocated.	-
0000	11111	001xxx	-	-	Unallocated.	-
0000	11111	01xxxx	-	-	Unallocated.	-
0000	11111	1xxxxx	-	-	Unallocated.	-
0001	11111	000000	-	!= 00000	Unallocated.	-
0001	11111	000000	-	00000	BLR	-
0001	11111	000001	-	-	Unallocated.	-
0001	11111	000010	-	!= 11111	Unallocated.	-
0001	11111	000010	-	11111	BLRAA, BLRAAZ, BLRAB, BLRABZ - Key A, zero modifier variant	FEAT_PAuth
0001	11111	000011	-	!= 11111	Unallocated.	-
0001	11111	000011	-	11111	BLRAA, BLRAAZ, BLRAB, BLRABZ - Key B, zero modifier variant	FEAT_PAuth
0001	11111	0001xx	-	-	Unallocated.	-
0001	11111	001xxx	-	-	Unallocated.	-
0001	11111	01xxxx	-	-	Unallocated.	-
0001	11111	1xxxxx	-	-	Unallocated.	-

Decode fields					Instruction page	Feature
opc	op2	op3	Rn	op4		
0010	11111	000000	-	!= 00000	Unallocated.	-
0010	11111	000000	-	00000	RET	-
0010	11111	000001	-	-	Unallocated.	-
0010	11111	000010	!= 11111	!= 11111	Unallocated.	-
0010	11111	000010	11111	11111	RETAA, RETAB - RETAA variant	FEAT_PAuth
0010	11111	000011	!= 11111	!= 11111	Unallocated.	-
0010	11111	000011	11111	11111	RETAA, RETAB - RETAB variant	FEAT_PAuth
0010	11111	0001xx	-	-	Unallocated.	-
0010	11111	001xxx	-	-	Unallocated.	-
0010	11111	01xxxx	-	-	Unallocated.	-
0010	11111	1xxxxx	-	-	Unallocated.	-
0011	11111	-	-	-	Unallocated.	-
0100	11111	000000	!= 11111	!= 00000	Unallocated.	-
0100	11111	000000	!= 11111	00000	Unallocated.	-
0100	11111	000000	11111	!= 00000	Unallocated.	-
0100	11111	000000	11111	00000	ERET	-
0100	11111	000001	-	-	Unallocated.	-
0100	11111	000010	!= 11111	!= 11111	Unallocated.	-
0100	11111	000010	!= 11111	11111	Unallocated.	-
0100	11111	000010	11111	!= 11111	Unallocated.	-
0100	11111	000010	11111	11111	ERETAA, ERETAB - ERETAA variant	FEAT_PAuth
0100	11111	000011	!= 11111	!= 11111	Unallocated.	-
0100	11111	000011	!= 11111	11111	Unallocated.	-
0100	11111	000011	11111	!= 11111	Unallocated.	-
0100	11111	000011	11111	11111	ERETAA, ERETAB - ERETAB variant	FEAT_PAuth
0100	11111	0001xx	-	-	Unallocated.	-
0100	11111	001xxx	-	-	Unallocated.	-
0100	11111	01xxxx	-	-	Unallocated.	-
0100	11111	1xxxxx	-	-	Unallocated.	-
0101	11111	!= 000000	-	-	Unallocated.	-
0101	11111	000000	!= 11111	!= 00000	Unallocated.	-
0101	11111	000000	!= 11111	00000	Unallocated.	-

Decode fields					Instruction page	Feature
opc	op2	op3	Rn	op4		
0101	11111	000000	11111	!= 00000	Unallocated.	-
0101	11111	000000	11111	00000	DRPS	-
011x	11111	-	-	-	Unallocated.	-
1000	11111	00000x	-	-	Unallocated.	-
1000	11111	000010	-	-	BRAA, BRAAZ, BRAB, BRABZ - Key A, register modifier variant	FEAT_PAuth
1000	11111	000011	-	-	BRAA, BRAAZ, BRAB, BRABZ - Key B, register modifier variant	FEAT_PAuth
1000	11111	0001xx	-	-	Unallocated.	-
1000	11111	001xxx	-	-	Unallocated.	-
1000	11111	01xxxx	-	-	Unallocated.	-
1000	11111	1xxxxx	-	-	Unallocated.	-
1001	11111	00000x	-	-	Unallocated.	-
1001	11111	000010	-	-	BLRAA, BLRAAZ, BLRAB, BLRABZ - Key A, register modifier variant	FEAT_PAuth
1001	11111	000011	-	-	BLRAA, BLRAAZ, BLRAB, BLRABZ - Key B, register modifier variant	FEAT_PAuth
1001	11111	0001xx	-	-	Unallocated.	-
1001	11111	001xxx	-	-	Unallocated.	-
1001	11111	01xxxx	-	-	Unallocated.	-
1001	11111	1xxxxx	-	-	Unallocated.	-
101x	11111	-	-	-	Unallocated.	-
11xx	11111	-	-	-	Unallocated.	-

Unconditional branch (immediate)

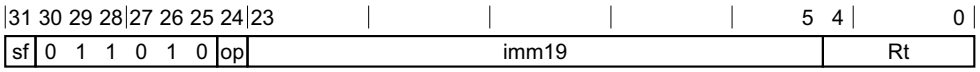
This section describes the encoding of the Unconditional branch (immediate) instruction class. The encodings in this section are decoded from *Branches, Exception Generating and System instructions* on page C4-285.



Decode fields	Instruction page
op	
0	B
1	BL

Compare and branch (immediate)

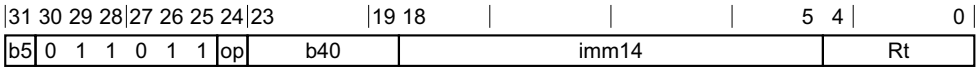
This section describes the encoding of the Compare and branch (immediate) instruction class. The encodings in this section are decoded from *Branches, Exception Generating and System instructions* on page C4-285.



Decode fields		Instruction page
sf	op	
0	0	CBZ - 32-bit variant
0	1	CBNZ - 32-bit variant
1	0	CBZ - 64-bit variant
1	1	CBNZ - 64-bit variant

Test and branch (immediate)

This section describes the encoding of the Test and branch (immediate) instruction class. The encodings in this section are decoded from *Branches, Exception Generating and System instructions* on page C4-285.



Decode fields	Instruction page
op	
0	TBZ
1	TBNZ

C4.1.4 Loads and Stores

This section describes the encoding of the Loads and Stores group. The encodings in this section are decoded from *A64 instruction set encoding* on page C4-280.

Table C4-5 Encoding table for the Loads and Stores group

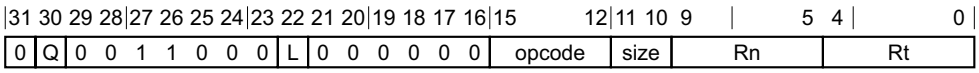
Decode fields					Decode group or instruction page
op0	op1	op2	op3	op4	
0x00	1	00	000000	-	*Advanced SIMD load/store multiple structures* on page C4-296
0x00	1	01	0xxxxx	-	*Advanced SIMD load/store multiple structures (post-indexed)* on page C4-297
0x00	1	0x	1xxxxx	-	Unallocated.
0x00	1	10	x00000	-	*Advanced SIMD load/store single structure* on page C4-298
0x00	1	11	-	-	*Advanced SIMD load/store single structure (post-indexed)* on page C4-301
0x00	1	x0	x1xxxx	-	Unallocated.
0x00	1	x0	xx1xxx	-	Unallocated.
0x00	1	x0	xxx1xx	-	Unallocated.
0x00	1	x0	xxxx1x	-	Unallocated.
0x00	1	x0	xxxxx1	-	Unallocated.
0x01	0	1x	1xxxxx	-	Unallocated.
1001	0	1x	1xxxxx	-	Unallocated.
1101	0	1x	1xxxxx	-	*Load/store memory tags* on page C4-305
1x00	1	-	-	-	Unallocated.
xx00	0	0x	-	-	*Load/store exclusive* on page C4-306
xx00	0	1x	-	-	Unallocated.
xx01	0	1x	0xxxxx	00	*LDAPR/STLR (unscaled immediate)* on page C4-308
xx01	1	1x	0xxxxx	00	Unallocated.
xx01	-	0x	-	-	*Load register (literal)* on page C4-309
xx10	-	00	-	-	*Load/store no-allocate pair (offset)* on page C4-309
xx10	-	01	-	-	*Load/store register pair (post-indexed)* on page C4-310
xx10	-	10	-	-	*Load/store register pair (offset)* on page C4-311
xx10	-	11	-	-	*Load/store register pair (pre-indexed)* on page C4-311
xx11	-	0x	0xxxxx	00	*Load/store register (unscaled immediate)* on page C4-312
xx11	-	0x	0xxxxx	01	*Load/store register (immediate post-indexed)* on page C4-313

Table C4-5 Encoding table for the Loads and Stores group (continued)

Decode fields					Decode group or instruction page
op0	op1	op2	op3	op4	
xx11	-	0x	0xxxxx	10	*Load/store register (unprivileged)* on page C4-314
xx11	-	0x	0xxxxx	11	*Load/store register (immediate pre-indexed)* on page C4-315
xx11	-	0x	1xxxxx	00	*Atomic memory operations* on page C4-316
xx11	-	0x	1xxxxx	10	*Load/store register (register offset)* on page C4-325
xx11	-	0x	1xxxxx	x1	*Load/store register (pac)* on page C4-326
xx11	-	1x	-	-	*Load/store register (unsigned immediate)* on page C4-326

Advanced SIMD load/store multiple structures

This section describes the encoding of the Advanced SIMD load/store multiple structures instruction class. The encodings in this section are decoded from *Loads and Stores* on page C4-295.

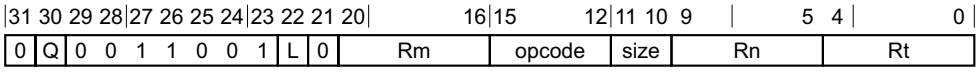


Decode fields		Instruction page
L	opcode	
0	0000	ST4 (multiple structures)
0	0001	Unallocated.
0	0010	ST1 (multiple structures) - Four registers variant
0	0011	Unallocated.
0	0100	ST3 (multiple structures)
0	0101	Unallocated.
0	0110	ST1 (multiple structures) - Three registers variant
0	0111	ST1 (multiple structures) - One register variant
0	1000	ST2 (multiple structures)
0	1001	Unallocated.
0	1010	ST1 (multiple structures) - Two registers variant
0	1011	Unallocated.
0	11xx	Unallocated.
1	0000	LD4 (multiple structures)
1	0001	Unallocated.
1	0010	LD1 (multiple structures) - Four registers variant

Decode fields		Instruction page
L	opcode	
1	0011	Unallocated.
1	0100	LD3 (multiple structures)
1	0101	Unallocated.
1	0110	LD1 (multiple structures) - Three registers variant
1	0111	LD1 (multiple structures) - One register variant
1	1000	LD2 (multiple structures)
1	1001	Unallocated.
1	1010	LD1 (multiple structures) - Two registers variant
1	1011	Unallocated.
1	11xx	Unallocated.

Advanced SIMD load/store multiple structures (post-indexed)

This section describes the encoding of the Advanced SIMD load/store multiple structures (post-indexed) instruction class. The encodings in this section are decoded from *Loads and Stores* on page C4-295.

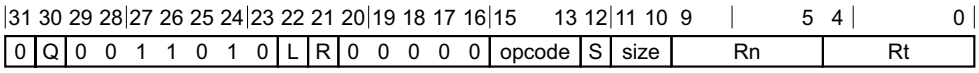


Decode fields			Instruction page
L	Rm	opcode	
0	-	0001	Unallocated.
0	-	0011	Unallocated.
0	-	0101	Unallocated.
0	-	1001	Unallocated.
0	-	1011	Unallocated.
0	-	11xx	Unallocated.
0	!= 11111	0000	ST4 (multiple structures) - Register offset variant
0	!= 11111	0010	ST1 (multiple structures) - Four registers, register offset variant
0	!= 11111	0100	ST3 (multiple structures) - Register offset variant
0	!= 11111	0110	ST1 (multiple structures) - Three registers, register offset variant
0	!= 11111	0111	ST1 (multiple structures) - One register, register offset variant
0	!= 11111	1000	ST2 (multiple structures) - Register offset variant
0	!= 11111	1010	ST1 (multiple structures) - Two registers, register offset variant

Decode fields			Instruction page
L	Rm	opcode	
0	11111	0000	ST4 (multiple structures) - Immediate offset variant
0	11111	0010	ST1 (multiple structures) - Four registers, immediate offset variant
0	11111	0100	ST3 (multiple structures) - Immediate offset variant
0	11111	0110	ST1 (multiple structures) - Three registers, immediate offset variant
0	11111	0111	ST1 (multiple structures) - One register, immediate offset variant
0	11111	1000	ST2 (multiple structures) - Immediate offset variant
0	11111	1010	ST1 (multiple structures) - Two registers, immediate offset variant
1	-	0001	Unallocated.
1	-	0011	Unallocated.
1	-	0101	Unallocated.
1	-	1001	Unallocated.
1	-	1011	Unallocated.
1	-	11xx	Unallocated.
1	!= 11111	0000	LD4 (multiple structures) - Register offset variant
1	!= 11111	0010	LD1 (multiple structures) - Four registers, register offset variant
1	!= 11111	0100	LD3 (multiple structures) - Register offset variant
1	!= 11111	0110	LD1 (multiple structures) - Three registers, register offset variant
1	!= 11111	0111	LD1 (multiple structures) - One register, register offset variant
1	!= 11111	1000	LD2 (multiple structures) - Register offset variant
1	!= 11111	1010	LD1 (multiple structures) - Two registers, register offset variant
1	11111	0000	LD4 (multiple structures) - Immediate offset variant
1	11111	0010	LD1 (multiple structures) - Four registers, immediate offset variant
1	11111	0100	LD3 (multiple structures) - Immediate offset variant
1	11111	0110	LD1 (multiple structures) - Three registers, immediate offset variant
1	11111	0111	LD1 (multiple structures) - One register, immediate offset variant
1	11111	1000	LD2 (multiple structures) - Immediate offset variant
1	11111	1010	LD1 (multiple structures) - Two registers, immediate offset variant

Advanced SIMD load/store single structure

This section describes the encoding of the Advanced SIMD load/store single structure instruction class. The encodings in this section are decoded from *Loads and Stores* on page C4-295.



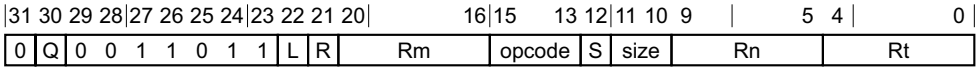
Decode fields					Instruction page
L	R	opcode	S	size	
0	-	11x	-	-	Unallocated.
0	0	000	-	-	ST1 (single structure) - 8-bit variant
0	0	001	-	-	ST3 (single structure) - 8-bit variant
0	0	010	-	x0	ST1 (single structure) - 16-bit variant
0	0	010	-	x1	Unallocated.
0	0	011	-	x0	ST3 (single structure) - 16-bit variant
0	0	011	-	x1	Unallocated.
0	0	100	-	00	ST1 (single structure) - 32-bit variant
0	0	100	-	1x	Unallocated.
0	0	100	0	01	ST1 (single structure) - 64-bit variant
0	0	100	1	01	Unallocated.
0	0	101	-	00	ST3 (single structure) - 32-bit variant
0	0	101	-	10	Unallocated.
0	0	101	0	01	ST3 (single structure) - 64-bit variant
0	0	101	0	11	Unallocated.
0	0	101	1	x1	Unallocated.
0	1	000	-	-	ST2 (single structure) - 8-bit variant
0	1	001	-	-	ST4 (single structure) - 8-bit variant
0	1	010	-	x0	ST2 (single structure) - 16-bit variant
0	1	010	-	x1	Unallocated.
0	1	011	-	x0	ST4 (single structure) - 16-bit variant
0	1	011	-	x1	Unallocated.
0	1	100	-	00	ST2 (single structure) - 32-bit variant
0	1	100	-	10	Unallocated.
0	1	100	0	01	ST2 (single structure) - 64-bit variant
0	1	100	0	11	Unallocated.
0	1	100	1	x1	Unallocated.
0	1	101	-	00	ST4 (single structure) - 32-bit variant

Decode fields					Instruction page
L	R	opcode	S	size	
0	1	101	-	10	Unallocated.
0	1	101	0	01	ST4 (single structure) - 64-bit variant
0	1	101	0	11	Unallocated.
0	1	101	1	x1	Unallocated.
1	0	000	-	-	LD1 (single structure) - 8-bit variant
1	0	001	-	-	LD3 (single structure) - 8-bit variant
1	0	010	-	x0	LD1 (single structure) - 16-bit variant
1	0	010	-	x1	Unallocated.
1	0	011	-	x0	LD3 (single structure) - 16-bit variant
1	0	011	-	x1	Unallocated.
1	0	100	-	00	LD1 (single structure) - 32-bit variant
1	0	100	-	1x	Unallocated.
1	0	100	0	01	LD1 (single structure) - 64-bit variant
1	0	100	1	01	Unallocated.
1	0	101	-	00	LD3 (single structure) - 32-bit variant
1	0	101	-	10	Unallocated.
1	0	101	0	01	LD3 (single structure) - 64-bit variant
1	0	101	0	11	Unallocated.
1	0	101	1	x1	Unallocated.
1	0	110	0	-	LD1R
1	0	110	1	-	Unallocated.
1	0	111	0	-	LD3R
1	0	111	1	-	Unallocated.
1	1	000	-	-	LD2 (single structure) - 8-bit variant
1	1	001	-	-	LD4 (single structure) - 8-bit variant
1	1	010	-	x0	LD2 (single structure) - 16-bit variant
1	1	010	-	x1	Unallocated.
1	1	011	-	x0	LD4 (single structure) - 16-bit variant
1	1	011	-	x1	Unallocated.
1	1	100	-	00	LD2 (single structure) - 32-bit variant
1	1	100	-	10	Unallocated.
1	1	100	0	01	LD2 (single structure) - 64-bit variant

Decode fields					Instruction page
L	R	opcode	S	size	
1	1	100	0	11	Unallocated.
1	1	100	1	x1	Unallocated.
1	1	101	-	00	LD4 (single structure) - 32-bit variant
1	1	101	-	10	Unallocated.
1	1	101	0	01	LD4 (single structure) - 64-bit variant
1	1	101	0	11	Unallocated.
1	1	101	1	x1	Unallocated.
1	1	110	0	-	LD2R
1	1	110	1	-	Unallocated.
1	1	111	0	-	LD4R
1	1	111	1	-	Unallocated.

Advanced SIMD load/store single structure (post-indexed)

This section describes the encoding of the Advanced SIMD load/store single structure (post-indexed) instruction class. The encodings in this section are decoded from *Loads and Stores* on page C4-295.



Decode fields						Instruction page
L	R	Rm	opcode	S	size	
0	-	-	11x	-	-	Unallocated.
0	0	-	010	-	x1	Unallocated.
0	0	-	011	-	x1	Unallocated.
0	0	-	100	-	1x	Unallocated.
0	0	-	100	1	01	Unallocated.
0	0	-	101	-	10	Unallocated.
0	0	-	101	0	11	Unallocated.
0	0	-	101	1	x1	Unallocated.
0	0	!= 11111	000	-	-	ST1 (single structure) - 8-bit, register offset variant
0	0	!= 11111	001	-	-	ST3 (single structure) - 8-bit, register offset variant
0	0	!= 11111	010	-	x0	ST1 (single structure) - 16-bit, register offset variant
0	0	!= 11111	011	-	x0	ST3 (single structure) - 16-bit, register offset variant

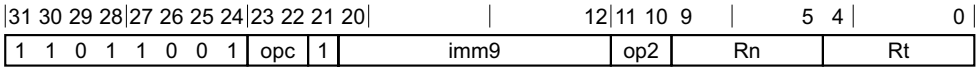
Decode fields						Instruction page
L	R	Rm	opcode	S	size	
0	0	!= 11111	100	-	00	ST1 (single structure) - 32-bit, register offset variant
0	0	!= 11111	100	0	01	ST1 (single structure) - 64-bit, register offset variant
0	0	!= 11111	101	-	00	ST3 (single structure) - 32-bit, register offset variant
0	0	!= 11111	101	0	01	ST3 (single structure) - 64-bit, register offset variant
0	0	11111	000	-	-	ST1 (single structure) - 8-bit, immediate offset variant
0	0	11111	001	-	-	ST3 (single structure) - 8-bit, immediate offset variant
0	0	11111	010	-	x0	ST1 (single structure) - 16-bit, immediate offset variant
0	0	11111	011	-	x0	ST3 (single structure) - 16-bit, immediate offset variant
0	0	11111	100	-	00	ST1 (single structure) - 32-bit, immediate offset variant
0	0	11111	100	0	01	ST1 (single structure) - 64-bit, immediate offset variant
0	0	11111	101	-	00	ST3 (single structure) - 32-bit, immediate offset variant
0	0	11111	101	0	01	ST3 (single structure) - 64-bit, immediate offset variant
0	1	-	010	-	x1	Unallocated.
0	1	-	011	-	x1	Unallocated.
0	1	-	100	-	10	Unallocated.
0	1	-	100	0	11	Unallocated.
0	1	-	100	1	x1	Unallocated.
0	1	-	101	-	10	Unallocated.
0	1	-	101	0	11	Unallocated.
0	1	-	101	1	x1	Unallocated.
0	1	!= 11111	000	-	-	ST2 (single structure) - 8-bit, register offset variant
0	1	!= 11111	001	-	-	ST4 (single structure) - 8-bit, register offset variant
0	1	!= 11111	010	-	x0	ST2 (single structure) - 16-bit, register offset variant
0	1	!= 11111	011	-	x0	ST4 (single structure) - 16-bit, register offset variant
0	1	!= 11111	100	-	00	ST2 (single structure) - 32-bit, register offset variant
0	1	!= 11111	100	0	01	ST2 (single structure) - 64-bit, register offset variant
0	1	!= 11111	101	-	00	ST4 (single structure) - 32-bit, register offset variant
0	1	!= 11111	101	0	01	ST4 (single structure) - 64-bit, register offset variant
0	1	11111	000	-	-	ST2 (single structure) - 8-bit, immediate offset variant
0	1	11111	001	-	-	ST4 (single structure) - 8-bit, immediate offset variant
0	1	11111	010	-	x0	ST2 (single structure) - 16-bit, immediate offset variant
0	1	11111	011	-	x0	ST4 (single structure) - 16-bit, immediate offset variant

Decode fields						Instruction page
L	R	Rm	opcode	S	size	
0	1	11111	100	-	00	ST2 (single structure) - 32-bit, immediate offset variant
0	1	11111	100	0	01	ST2 (single structure) - 64-bit, immediate offset variant
0	1	11111	101	-	00	ST4 (single structure) - 32-bit, immediate offset variant
0	1	11111	101	0	01	ST4 (single structure) - 64-bit, immediate offset variant
1	0	-	010	-	x1	Unallocated.
1	0	-	011	-	x1	Unallocated.
1	0	-	100	-	1x	Unallocated.
1	0	-	100	1	01	Unallocated.
1	0	-	101	-	10	Unallocated.
1	0	-	101	0	11	Unallocated.
1	0	-	101	1	x1	Unallocated.
1	0	-	110	1	-	Unallocated.
1	0	-	111	1	-	Unallocated.
1	0	!= 11111	000	-	-	LD1 (single structure) - 8-bit, register offset variant
1	0	!= 11111	001	-	-	LD3 (single structure) - 8-bit, register offset variant
1	0	!= 11111	010	-	x0	LD1 (single structure) - 16-bit, register offset variant
1	0	!= 11111	011	-	x0	LD3 (single structure) - 16-bit, register offset variant
1	0	!= 11111	100	-	00	LD1 (single structure) - 32-bit, register offset variant
1	0	!= 11111	100	0	01	LD1 (single structure) - 64-bit, register offset variant
1	0	!= 11111	101	-	00	LD3 (single structure) - 32-bit, register offset variant
1	0	!= 11111	101	0	01	LD3 (single structure) - 64-bit, register offset variant
1	0	!= 11111	110	0	-	LD1R - Register offset variant
1	0	!= 11111	111	0	-	LD3R - Register offset variant
1	0	11111	000	-	-	LD1 (single structure) - 8-bit, immediate offset variant
1	0	11111	001	-	-	LD3 (single structure) - 8-bit, immediate offset variant
1	0	11111	010	-	x0	LD1 (single structure) - 16-bit, immediate offset variant
1	0	11111	011	-	x0	LD3 (single structure) - 16-bit, immediate offset variant
1	0	11111	100	-	00	LD1 (single structure) - 32-bit, immediate offset variant
1	0	11111	100	0	01	LD1 (single structure) - 64-bit, immediate offset variant
1	0	11111	101	-	00	LD3 (single structure) - 32-bit, immediate offset variant
1	0	11111	101	0	01	LD3 (single structure) - 64-bit, immediate offset variant
1	0	11111	110	0	-	LD1R - Immediate offset variant

Decode fields						Instruction page
L	R	Rm	opcode	S	size	
1	0	11111	111	0	-	LD3R - Immediate offset variant
1	1	-	010	-	x1	Unallocated.
1	1	-	011	-	x1	Unallocated.
1	1	-	100	-	10	Unallocated.
1	1	-	100	0	11	Unallocated.
1	1	-	100	1	x1	Unallocated.
1	1	-	101	-	10	Unallocated.
1	1	-	101	0	11	Unallocated.
1	1	-	101	1	x1	Unallocated.
1	1	-	110	1	-	Unallocated.
1	1	-	111	1	-	Unallocated.
1	1	!= 11111	000	-	-	LD2 (single structure) - 8-bit, register offset variant
1	1	!= 11111	001	-	-	LD4 (single structure) - 8-bit, register offset variant
1	1	!= 11111	010	-	x0	LD2 (single structure) - 16-bit, register offset variant
1	1	!= 11111	011	-	x0	LD4 (single structure) - 16-bit, register offset variant
1	1	!= 11111	100	-	00	LD2 (single structure) - 32-bit, register offset variant
1	1	!= 11111	100	0	01	LD2 (single structure) - 64-bit, register offset variant
1	1	!= 11111	101	-	00	LD4 (single structure) - 32-bit, register offset variant
1	1	!= 11111	101	0	01	LD4 (single structure) - 64-bit, register offset variant
1	1	!= 11111	110	0	-	LD2R - Register offset variant
1	1	!= 11111	111	0	-	LD4R - Register offset variant
1	1	11111	000	-	-	LD2 (single structure) - 8-bit, immediate offset variant
1	1	11111	001	-	-	LD4 (single structure) - 8-bit, immediate offset variant
1	1	11111	010	-	x0	LD2 (single structure) - 16-bit, immediate offset variant
1	1	11111	011	-	x0	LD4 (single structure) - 16-bit, immediate offset variant
1	1	11111	100	-	00	LD2 (single structure) - 32-bit, immediate offset variant
1	1	11111	100	0	01	LD2 (single structure) - 64-bit, immediate offset variant
1	1	11111	101	-	00	LD4 (single structure) - 32-bit, immediate offset variant
1	1	11111	101	0	01	LD4 (single structure) - 64-bit, immediate offset variant
1	1	11111	110	0	-	LD2R - Immediate offset variant
1	1	11111	111	0	-	LD4R - Immediate offset variant

Load/store memory tags

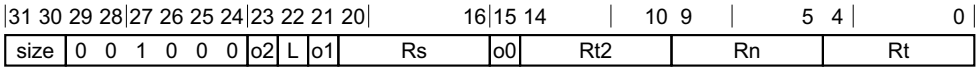
This section describes the encoding of the Load/store memory tags instruction class. The encodings in this section are decoded from *Loads and Stores* on page C4-295.



Decode fields			Instruction page	Feature
opc	imm9	op2		
00	-	01	STG - Encoding	FEAT_MTE
00	-	10	STG - Encoding	FEAT_MTE
00	-	11	STG - Encoding	FEAT_MTE
00	000000000	00	STZGM	FEAT_MTE2
01	-	00	LDG	FEAT_MTE
01	-	01	STZG - Encoding	FEAT_MTE
01	-	10	STZG - Encoding	FEAT_MTE
01	-	11	STZG - Encoding	FEAT_MTE
10	-	01	ST2G - Encoding	FEAT_MTE
10	-	10	ST2G - Encoding	FEAT_MTE
10	-	11	ST2G - Encoding	FEAT_MTE
10	!= 000000000	00	Unallocated.	-
10	000000000	00	STGM	FEAT_MTE2
11	-	01	STZ2G - Encoding	FEAT_MTE
11	-	10	STZ2G - Encoding	FEAT_MTE
11	-	11	STZ2G - Encoding	FEAT_MTE
11	!= 000000000	00	Unallocated.	-
11	000000000	00	LDGM	FEAT_MTE2

Load/store exclusive

This section describes the encoding of the Load/store exclusive instruction class. The encodings in this section are decoded from *Loads and Stores* on page C4-295.



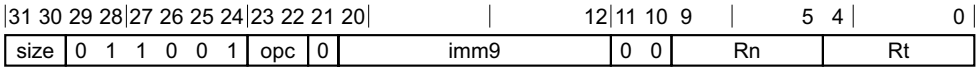
Decode fields						Instruction page	Feature
size	o2	L	o1	o0	Rt2		
-	1	-	1	-	!= 11111	Unallocated.	-
0x	0	-	1	-	!= 11111	Unallocated.	-
00	0	0	0	0	-	STXRB	-
00	0	0	0	1	-	STLXRB	-
00	0	0	1	0	11111	CASP, CASPA, CASPAL, CASPL - 32-bit CASP variant	FEAT_LSE
00	0	0	1	1	11111	CASP, CASPA, CASPAL, CASPL - 32-bit CASPL variant	FEAT_LSE
00	0	1	0	0	-	LDXRB	-
00	0	1	0	1	-	LDAXRB	-
00	0	1	1	0	11111	CASP, CASPA, CASPAL, CASPL - 32-bit CASPA variant	FEAT_LSE
00	0	1	1	1	11111	CASP, CASPA, CASPAL, CASPL - 32-bit CASPAL variant	FEAT_LSE
00	1	0	0	0	-	STLLRB	FEAT_LOR
00	1	0	0	1	-	STLRB	-
00	1	0	1	0	11111	CASB, CASAB, CASALB, CASLB - CASB variant	FEAT_LSE
00	1	0	1	1	11111	CASB, CASAB, CASALB, CASLB - CASLB variant	FEAT_LSE
00	1	1	0	0	-	LDLARB	FEAT_LOR
00	1	1	0	1	-	LDARB	-
00	1	1	1	0	11111	CASB, CASAB, CASALB, CASLB - CASAB variant	FEAT_LSE
00	1	1	1	1	11111	CASB, CASAB, CASALB, CASLB - CASALB variant	FEAT_LSE
01	0	0	0	0	-	STXRH	-
01	0	0	0	1	-	STLXRH	-
01	0	0	1	0	11111	CASP, CASPA, CASPAL, CASPL - 64-bit CASP variant	FEAT_LSE
01	0	0	1	1	11111	CASP, CASPA, CASPAL, CASPL - 64-bit CASPL variant	FEAT_LSE
01	0	1	0	0	-	LDXRH	-
01	0	1	0	1	-	LDAXRH	-
01	0	1	1	0	11111	CASP, CASPA, CASPAL, CASPL - 64-bit CASPA variant	FEAT_LSE
01	0	1	1	1	11111	CASP, CASPA, CASPAL, CASPL - 64-bit CASPAL variant	FEAT_LSE

Decode fields						Instruction page	Feature
size	o2	L	o1	o0	Rt2		
01	1	0	0	0	-	STLLRH	FEAT_LOR
01	1	0	0	1	-	STLRH	-
01	1	0	1	0	11111	CASH, CASAH, CASALH, CASLH - CASH variant	FEAT_LSE
01	1	0	1	1	11111	CASH, CASAH, CASALH, CASLH - CASLH variant	FEAT_LSE
01	1	1	0	0	-	LDLARH	FEAT_LOR
01	1	1	0	1	-	LDARH	-
01	1	1	1	0	11111	CASH, CASAH, CASALH, CASLH - CASAH variant	FEAT_LSE
01	1	1	1	1	11111	CASH, CASAH, CASALH, CASLH - CASALH variant	FEAT_LSE
10	0	0	0	0	-	STXR - 32-bit variant	-
10	0	0	0	1	-	STLXR - 32-bit variant	-
10	0	0	1	0	-	STXP - 32-bit variant	-
10	0	0	1	1	-	STLXP - 32-bit variant	-
10	0	1	0	0	-	LDXR - 32-bit variant	-
10	0	1	0	1	-	LDAXR - 32-bit variant	-
10	0	1	1	0	-	LDXP - 32-bit variant	-
10	0	1	1	1	-	LDAXP - 32-bit variant	-
10	1	0	0	0	-	STLLR - 32-bit variant	FEAT_LOR
10	1	0	0	1	-	STLR - 32-bit variant	-
10	1	0	1	0	11111	CAS, CASA, CASAL, CASL - 32-bit CAS variant	FEAT_LSE
10	1	0	1	1	11111	CAS, CASA, CASAL, CASL - 32-bit CASL variant	FEAT_LSE
10	1	1	0	0	-	LDLAR - 32-bit variant	FEAT_LOR
10	1	1	0	1	-	LDAR - 32-bit variant	-
10	1	1	1	0	11111	CAS, CASA, CASAL, CASL - 32-bit CASA variant	FEAT_LSE
10	1	1	1	1	11111	CAS, CASA, CASAL, CASL - 32-bit CASAL variant	FEAT_LSE
11	0	0	0	0	-	STXR - 64-bit variant	-
11	0	0	0	1	-	STLXR - 64-bit variant	-
11	0	0	1	0	-	STXP - 64-bit variant	-
11	0	0	1	1	-	STLXP - 64-bit variant	-
11	0	1	0	0	-	LDXR - 64-bit variant	-
11	0	1	0	1	-	LDAXR - 64-bit variant	-
11	0	1	1	0	-	LDXP - 64-bit variant	-
11	0	1	1	1	-	LDAXP - 64-bit variant	-

Decode fields						Instruction page	Feature
size	o2	L	o1	o0	Rt2		
11	1	0	0	0	-	STLLR - 64-bit variant	FEAT_LOR
11	1	0	0	1	-	STLR - 64-bit variant	-
11	1	0	1	0	11111	CAS, CASA, CASAL, CASL - 64-bit CAS variant	FEAT_LSE
11	1	0	1	1	11111	CAS, CASA, CASAL, CASL - 64-bit CASL variant	FEAT_LSE
11	1	1	0	0	-	LDLAR - 64-bit variant	FEAT_LOR
11	1	1	0	1	-	LDAR - 64-bit variant	-
11	1	1	1	0	11111	CAS, CASA, CASAL, CASL - 64-bit CASA variant	FEAT_LSE
11	1	1	1	1	11111	CAS, CASA, CASAL, CASL - 64-bit CASAL variant	FEAT_LSE

LDAPR/STLR (unscaled immediate)

This section describes the encoding of the LDAPR/STLR (unscaled immediate) instruction class. The encodings in this section are decoded from *Loads and Stores* on page C4-295.

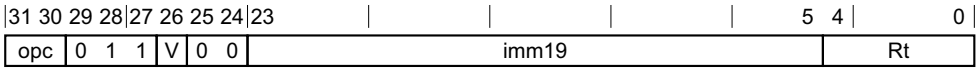


Decode fields		Instruction page	Feature
size	opc		
00	00	STLURB	FEAT_LRCPC2
00	01	LDAPURB	FEAT_LRCPC2
00	10	LDAPURSB - 64-bit variant	FEAT_LRCPC2
00	11	LDAPURSB - 32-bit variant	FEAT_LRCPC2
01	00	STLURH	FEAT_LRCPC2
01	01	LDAPURH	FEAT_LRCPC2
01	10	LDAPURSH - 64-bit variant	FEAT_LRCPC2
01	11	LDAPURSH - 32-bit variant	FEAT_LRCPC2
10	00	STLUR - 32-bit variant	FEAT_LRCPC2
10	01	LDAPUR - 32-bit variant	FEAT_LRCPC2
10	10	LDAPURSW	FEAT_LRCPC2
10	11	Unallocated.	-
11	00	STLUR - 64-bit variant	FEAT_LRCPC2

Decode fields		Instruction page	Feature
size	opc		
11	01	LDAPUR - 64-bit variant	FEAT_LRCPC2
11	10	Unallocated.	-
11	11	Unallocated.	-

Load register (literal)

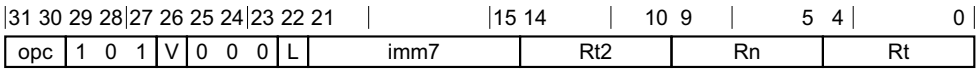
This section describes the encoding of the Load register (literal) instruction class. The encodings in this section are decoded from *Loads and Stores* on page C4-295.



Decode fields		Instruction page
opc	V	
00	0	LDR (literal) - 32-bit variant
00	1	LDR (literal, SIMD&FP) - 32-bit variant
01	0	LDR (literal) - 64-bit variant
01	1	LDR (literal, SIMD&FP) - 64-bit variant
10	0	LDRSW (literal)
10	1	LDR (literal, SIMD&FP) - 128-bit variant
11	0	PRFM (literal)
11	1	Unallocated.

Load/store no-allocate pair (offset)

This section describes the encoding of the Load/store no-allocate pair (offset) instruction class. The encodings in this section are decoded from *Loads and Stores* on page C4-295.

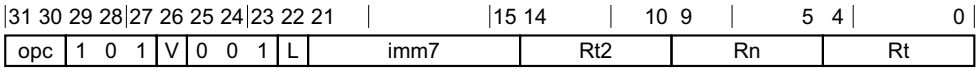


Decode fields			Instruction page
opc	V	L	
00	0	0	STNP - 32-bit variant
00	0	1	LDNP - 32-bit variant
00	1	0	STNP (SIMD&FP) - 32-bit variant

Decode fields			Instruction page
opc	V	L	
00	1	1	LDNP (SIMD&FP) - 32-bit variant
01	0	-	Unallocated.
01	1	0	STNP (SIMD&FP) - 64-bit variant
01	1	1	LDNP (SIMD&FP) - 64-bit variant
10	0	0	STNP - 64-bit variant
10	0	1	LDNP - 64-bit variant
10	1	0	STNP (SIMD&FP) - 128-bit variant
10	1	1	LDNP (SIMD&FP) - 128-bit variant
11	-	-	Unallocated.

Load/store register pair (post-indexed)

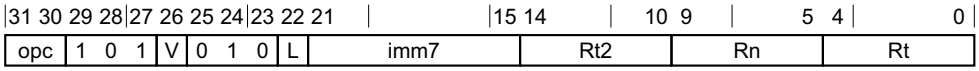
This section describes the encoding of the Load/store register pair (post-indexed) instruction class. The encodings in this section are decoded from *Loads and Stores* on page C4-295.



Decode fields			Instruction page	Feature
opc	V	L		
00	0	0	STP - 32-bit variant	-
00	0	1	LDP - 32-bit variant	-
00	1	0	STP (SIMD&FP) - 32-bit variant	-
00	1	1	LDP (SIMD&FP) - 32-bit variant	-
01	0	0	STGP	FEAT_MTE
01	0	1	LDPSW	-
01	1	0	STP (SIMD&FP) - 64-bit variant	-
01	1	1	LDP (SIMD&FP) - 64-bit variant	-
10	0	0	STP - 64-bit variant	-
10	0	1	LDP - 64-bit variant	-
10	1	0	STP (SIMD&FP) - 128-bit variant	-
10	1	1	LDP (SIMD&FP) - 128-bit variant	-
11	-	-	Unallocated.	-

Load/store register pair (offset)

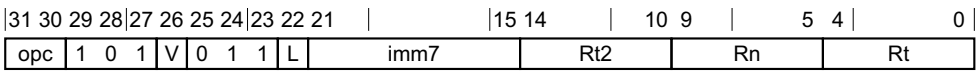
This section describes the encoding of the Load/store register pair (offset) instruction class. The encodings in this section are decoded from *Loads and Stores* on page C4-295.



Decode fields			Instruction page	Feature
opc	V	L		
00	0	0	STP - 32-bit variant	-
00	0	1	LDP - 32-bit variant	-
00	1	0	STP (SIMD&FP) - 32-bit variant	-
00	1	1	LDP (SIMD&FP) - 32-bit variant	-
01	0	0	STGP	FEAT_MTE
01	0	1	LDPSW	-
01	1	0	STP (SIMD&FP) - 64-bit variant	-
01	1	1	LDP (SIMD&FP) - 64-bit variant	-
10	0	0	STP - 64-bit variant	-
10	0	1	LDP - 64-bit variant	-
10	1	0	STP (SIMD&FP) - 128-bit variant	-
10	1	1	LDP (SIMD&FP) - 128-bit variant	-
11	-	-	Unallocated.	-

Load/store register pair (pre-indexed)

This section describes the encoding of the Load/store register pair (pre-indexed) instruction class. The encodings in this section are decoded from *Loads and Stores* on page C4-295.

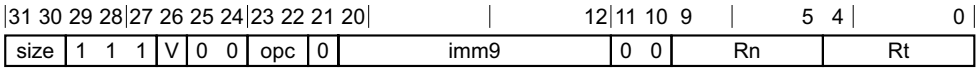


Decode fields			Instruction page	Feature
opc	V	L		
00	0	0	STP - 32-bit variant	-
00	0	1	LDP - 32-bit variant	-
00	1	0	STP (SIMD&FP) - 32-bit variant	-
00	1	1	LDP (SIMD&FP) - 32-bit variant	-

Decode fields			Instruction page	Feature
opc	V	L		
01	0	0	STGP	FEAT_MTE
01	0	1	LDPSW	-
01	1	0	STP (SIMD&FP) - 64-bit variant	-
01	1	1	LDP (SIMD&FP) - 64-bit variant	-
10	0	0	STP - 64-bit variant	-
10	0	1	LDP - 64-bit variant	-
10	1	0	STP (SIMD&FP) - 128-bit variant	-
10	1	1	LDP (SIMD&FP) - 128-bit variant	-
11	-	-	Unallocated.	-

Load/store register (unscaled immediate)

This section describes the encoding of the Load/store register (unscaled immediate) instruction class. The encodings in this section are decoded from *Loads and Stores* on page C4-295.

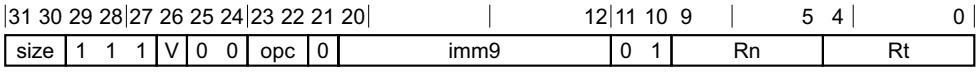


Decode fields			Instruction page
size	V	opc	
x1	1	1x	Unallocated.
00	0	00	STURB
00	0	01	LDURB
00	0	10	LDURSB - 64-bit variant
00	0	11	LDURSB - 32-bit variant
00	1	00	STUR (SIMD&FP) - 8-bit variant
00	1	01	LDUR (SIMD&FP) - 8-bit variant
00	1	10	STUR (SIMD&FP) - 128-bit variant
00	1	11	LDUR (SIMD&FP) - 128-bit variant
01	0	00	STURH
01	0	01	LDURH
01	0	10	LDURSH - 64-bit variant
01	0	11	LDURSH - 32-bit variant
01	1	00	STUR (SIMD&FP) - 16-bit variant

Decode fields			Instruction page
size	V	opc	
01	1	01	LDUR (SIMD&FP) - 16-bit variant
1x	0	11	Unallocated.
1x	1	1x	Unallocated.
10	0	00	STUR - 32-bit variant
10	0	01	LDUR - 32-bit variant
10	0	10	LDURSW
10	1	00	STUR (SIMD&FP) - 32-bit variant
10	1	01	LDUR (SIMD&FP) - 32-bit variant
11	0	00	STUR - 64-bit variant
11	0	01	LDUR - 64-bit variant
11	0	10	PRFUM
11	1	00	STUR (SIMD&FP) - 64-bit variant
11	1	01	LDUR (SIMD&FP) - 64-bit variant

Load/store register (immediate post-indexed)

This section describes the encoding of the Load/store register (immediate post-indexed) instruction class. The encodings in this section are decoded from *Loads and Stores* on page C4-295.

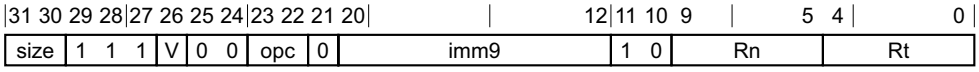


Decode fields			Instruction page
size	V	opc	
x1	1	1x	Unallocated.
00	0	00	STRB (immediate)
00	0	01	LDRB (immediate)
00	0	10	LDRSB (immediate) - 64-bit variant
00	0	11	LDRSB (immediate) - 32-bit variant
00	1	00	STR (immediate, SIMD&FP) - 8-bit variant
00	1	01	LDR (immediate, SIMD&FP) - 8-bit variant
00	1	10	STR (immediate, SIMD&FP) - 128-bit variant
00	1	11	LDR (immediate, SIMD&FP) - 128-bit variant
01	0	00	STRH (immediate)

Decode fields			Instruction page
size	V	opc	
01	0	01	LDRH (immediate)
01	0	10	LDRSH (immediate) - 64-bit variant
01	0	11	LDRSH (immediate) - 32-bit variant
01	1	00	STR (immediate, SIMD&FP) - 16-bit variant
01	1	01	LDR (immediate, SIMD&FP) - 16-bit variant
1x	0	11	Unallocated.
1x	1	1x	Unallocated.
10	0	00	STR (immediate) - 32-bit variant
10	0	01	LDR (immediate) - 32-bit variant
10	0	10	LDRSW (immediate)
10	1	00	STR (immediate, SIMD&FP) - 32-bit variant
10	1	01	LDR (immediate, SIMD&FP) - 32-bit variant
11	0	00	STR (immediate) - 64-bit variant
11	0	01	LDR (immediate) - 64-bit variant
11	0	10	Unallocated.
11	1	00	STR (immediate, SIMD&FP) - 64-bit variant
11	1	01	LDR (immediate, SIMD&FP) - 64-bit variant

Load/store register (unprivileged)

This section describes the encoding of the Load/store register (unprivileged) instruction class. The encodings in this section are decoded from *Loads and Stores* on page C4-295.

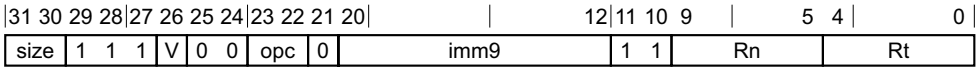


Decode fields			Instruction page
size	V	opc	
-	1	-	Unallocated.
00	0	00	STTRB
00	0	01	LDTRB
00	0	10	LDTRSB - 64-bit variant
00	0	11	LDTRSB - 32-bit variant
01	0	00	STTRH

Decode fields			Instruction page
size	V	opc	
01	0	01	LDTRH
01	0	10	LDTRSH - 64-bit variant
01	0	11	LDTRSH - 32-bit variant
1x	0	11	Unallocated.
10	0	00	STTR - 32-bit variant
10	0	01	LDTR - 32-bit variant
10	0	10	LDTRSW
11	0	00	STTR - 64-bit variant
11	0	01	LDTR - 64-bit variant
11	0	10	Unallocated.

Load/store register (immediate pre-indexed)

This section describes the encoding of the Load/store register (immediate pre-indexed) instruction class. The encodings in this section are decoded from *Loads and Stores* on page C4-295.

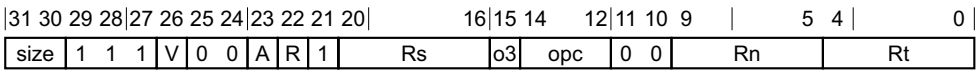


Decode fields			Instruction page
size	V	opc	
x1	1	1x	Unallocated.
00	0	00	STRB (immediate)
00	0	01	LDRB (immediate)
00	0	10	LDRSB (immediate) - 64-bit variant
00	0	11	LDRSB (immediate) - 32-bit variant
00	1	00	STR (immediate, SIMD&FP) - 8-bit variant
00	1	01	LDR (immediate, SIMD&FP) - 8-bit variant
00	1	10	STR (immediate, SIMD&FP) - 128-bit variant
00	1	11	LDR (immediate, SIMD&FP) - 128-bit variant
01	0	00	STRH (immediate)
01	0	01	LDRH (immediate)
01	0	10	LDRSH (immediate) - 64-bit variant
01	0	11	LDRSH (immediate) - 32-bit variant

Decode fields			Instruction page
size	V	opc	
01	1	00	STR (immediate, SIMD&FP) - 16-bit variant
01	1	01	LDR (immediate, SIMD&FP) - 16-bit variant
1x	0	11	Unallocated.
1x	1	1x	Unallocated.
10	0	00	STR (immediate) - 32-bit variant
10	0	01	LDR (immediate) - 32-bit variant
10	0	10	LDRSW (immediate)
10	1	00	STR (immediate, SIMD&FP) - 32-bit variant
10	1	01	LDR (immediate, SIMD&FP) - 32-bit variant
11	0	00	STR (immediate) - 64-bit variant
11	0	01	LDR (immediate) - 64-bit variant
11	0	10	Unallocated.
11	1	00	STR (immediate, SIMD&FP) - 64-bit variant
11	1	01	LDR (immediate, SIMD&FP) - 64-bit variant

Atomic memory operations

This section describes the encoding of the Atomic memory operations instruction class. The encodings in this section are decoded from *Loads and Stores* on page C4-295.



Decode fields							Instruction page	Feature
size	V	A	R	Rs	o3	opc		
-	0	-	-	-	1	11x	Unallocated.	-
-	0	0	-	-	1	100	Unallocated.	-
-	0	0	1	-	1	001	Unallocated.	-
-	0	0	1	-	1	010	Unallocated.	-
-	0	0	1	-	1	011	Unallocated.	-
-	0	0	1	-	1	101	Unallocated.	-
-	0	1	0	-	1	001	Unallocated.	-
-	0	1	0	-	1	010	Unallocated.	-
-	0	1	0	-	1	011	Unallocated.	-

Decode fields							Instruction page	Feature
size	V	A	R	Rs	o3	opc		
-	0	1	0	-	1	101	Unallocated.	-
-	0	1	1	-	1	001	Unallocated.	-
-	0	1	1	-	1	010	Unallocated.	-
-	0	1	1	-	1	011	Unallocated.	-
-	0	1	1	-	1	100	Unallocated.	-
-	0	1	1	-	1	101	Unallocated.	-
-	1	-	-	-	-	-	Unallocated.	-
00	0	0	0	-	0	000	LDADDB, LDADDAB, LDADDALB, LDADDLB - LDADDB variant	FEAT_LSE
00	0	0	0	-	0	001	LDCLRB, LDCLRAB, LDCLRALB, LDCLRLB - LDCLRB variant	FEAT_LSE
00	0	0	0	-	0	010	LDEORB, LDEORAB, LDEORALB, LDEORLB - LDEORB variant	FEAT_LSE
00	0	0	0	-	0	011	LDSETB, LDSETAB, LDSETALB, LDSETLB - LDSETB variant	FEAT_LSE
00	0	0	0	-	0	100	LDSMAXB, LDSMAXAB, LDSMAXALB, LDSMAXLB - LDSMAXB variant	FEAT_LSE
00	0	0	0	-	0	101	LDSMINB, LDSMINAB, LDSMINALB, LDSMINLB - LDSMINB variant	FEAT_LSE
00	0	0	0	-	0	110	LDUMAXB, LDUMAXAB, LDUMAXALB, LDUMAXLB - LDUMAXB variant	FEAT_LSE
00	0	0	0	-	0	111	LDUMINB, LDUMINAB, LDUMINALB, LDUMINLB - LDUMINB variant	FEAT_LSE
00	0	0	0	-	1	000	SWPB, SWPAB, SWPALB, SWPLB - SWPB variant	FEAT_LSE
00	0	0	0	-	1	001	Unallocated.	-
00	0	0	0	-	1	010	Unallocated.	-
00	0	0	0	-	1	011	Unallocated.	-
00	0	0	0	-	1	101	Unallocated.	-
00	0	0	1	-	0	000	LDADDB, LDADDAB, LDADDALB, LDADDLB - LDADDLB variant	FEAT_LSE
00	0	0	1	-	0	001	LDCLRB, LDCLRAB, LDCLRALB, LDCLRLB - LDCLRLB variant	FEAT_LSE
00	0	0	1	-	0	010	LDEORB, LDEORAB, LDEORALB, LDEORLB - LDEORLB variant	FEAT_LSE
00	0	0	1	-	0	011	LDSETB, LDSETAB, LDSETALB, LDSETLB - LDSETLB variant	FEAT_LSE
00	0	0	1	-	0	100	LDSMAXB, LDSMAXAB, LDSMAXALB, LDSMAXLB - LDSMAXLB variant	FEAT_LSE

Decode fields							Instruction page	Feature
size	V	A	R	Rs	o3	opc		
00	0	0	1	-	0	101	LDSMINB, LDSMINAB, LDSMINALB, LDSMINLB - LDSMINLB variant	FEAT_LSE
00	0	0	1	-	0	110	LDUMAXB, LDUMAXAB, LDUMAXALB, LDUMAXLB - LDUMAXLB variant	FEAT_LSE
00	0	0	1	-	0	111	LDUMINB, LDUMINAB, LDUMINALB, LDUMINLB - LDUMINLB variant	FEAT_LSE
00	0	0	1	-	1	000	SWPB, SWPAB, SWPALB, SWPLB - SWPLB variant	FEAT_LSE
00	0	1	0	-	0	000	LDADDB, LDADDAB, LDADDALB, LDADDLB - LDADDAB variant	FEAT_LSE
00	0	1	0	-	0	001	LDCLRB, LDCLRAB, LDCLRALB, LDCLRLB - LDCLRAB variant	FEAT_LSE
00	0	1	0	-	0	010	LDEORB, LDEORAB, LDEORALB, LDEORLB - LDEORAB variant	FEAT_LSE
00	0	1	0	-	0	011	LDSETB, LDSETAB, LDSETALB, LDSETLB - LDSETAB variant	FEAT_LSE
00	0	1	0	-	0	100	LDSMAXB, LDSMAXAB, LDSMAXALB, LDSMAXLB - LDSMAXAB variant	FEAT_LSE
00	0	1	0	-	0	101	LDSMINB, LDSMINAB, LDSMINALB, LDSMINLB - LDSMINAB variant	FEAT_LSE
00	0	1	0	-	0	110	LDUMAXB, LDUMAXAB, LDUMAXALB, LDUMAXLB - LDUMAXAB variant	FEAT_LSE
00	0	1	0	-	0	111	LDUMINB, LDUMINAB, LDUMINALB, LDUMINLB - LDUMINAB variant	FEAT_LSE
00	0	1	0	-	1	000	SWPB, SWPAB, SWPALB, SWPLB - SWPAB variant	FEAT_LSE
00	0	1	0	-	1	100	LDAPRB	FEAT_LRCPC
00	0	1	1	-	0	000	LDADDB, LDADDAB, LDADDALB, LDADDLB - LDADDALB variant	FEAT_LSE
00	0	1	1	-	0	001	LDCLRB, LDCLRAB, LDCLRALB, LDCLRLB - LDCLRALB variant	FEAT_LSE
00	0	1	1	-	0	010	LDEORB, LDEORAB, LDEORALB, LDEORLB - LDEORALB variant	FEAT_LSE
00	0	1	1	-	0	011	LDSETB, LDSETAB, LDSETALB, LDSETLB - LDSETALB variant	FEAT_LSE
00	0	1	1	-	0	100	LDSMAXB, LDSMAXAB, LDSMAXALB, LDSMAXLB - LDSMAXALB variant	FEAT_LSE
00	0	1	1	-	0	101	LDSMINB, LDSMINAB, LDSMINALB, LDSMINLB - LDSMINALB variant	FEAT_LSE
00	0	1	1	-	0	110	LDUMAXB, LDUMAXAB, LDUMAXALB, LDUMAXLB - LDUMAXALB variant	FEAT_LSE

Decode fields							Instruction page	Feature
size	V	A	R	Rs	o3	opc		
00	0	1	1	-	0	111	LDUMINB, LDUMINAB, LDUMINALB, LDUMINLB - LDUMINALB variant	FEAT_LSE
00	0	1	1	-	1	000	SWPB, SWPAB, SWPALB, SWPLB - SWPALB variant	FEAT_LSE
01	0	0	0	-	0	000	LDADDH, LDADDAH, LDADDALH, LDADDLH - LDADDH variant	FEAT_LSE
01	0	0	0	-	0	001	LDCLRH, LDCLRAH, LDCLRALH, LDCLRLH - LDCLRH variant	FEAT_LSE
01	0	0	0	-	0	010	LDEORH, LDEORAH, LDEORALH, LDEORLH - LDEORH variant	FEAT_LSE
01	0	0	0	-	0	011	LDSETH, LDSETAH, LDSETALH, LDSETLH - LDSETH variant	FEAT_LSE
01	0	0	0	-	0	100	LDSMAXH, LDSMAXAH, LDSMAXALH, LDSMAXLH - LDSMAXH variant	FEAT_LSE
01	0	0	0	-	0	101	LDSMINH, LDSMINAH, LDSMINALH, LDSMINLH - LDSMINH variant	FEAT_LSE
01	0	0	0	-	0	110	LDUMAXH, LDUMAXAH, LDUMAXALH, LDUMAXLH - LDUMAXH variant	FEAT_LSE
01	0	0	0	-	0	111	LDUMINH, LDUMINAH, LDUMINALH, LDUMINLH - LDUMINH variant	FEAT_LSE
01	0	0	0	-	1	000	SWPH, SWPAH, SWPALH, SWPLH - SWPH variant	FEAT_LSE
01	0	0	0	-	1	001	Unallocated.	-
01	0	0	0	-	1	010	Unallocated.	-
01	0	0	0	-	1	011	Unallocated.	-
01	0	0	0	-	1	101	Unallocated.	-
01	0	0	1	-	0	000	LDADDH, LDADDAH, LDADDALH, LDADDLH - LDADDLH variant	FEAT_LSE
01	0	0	1	-	0	001	LDCLRH, LDCLRAH, LDCLRALH, LDCLRLH - LDCLRLH variant	FEAT_LSE
01	0	0	1	-	0	010	LDEORH, LDEORAH, LDEORALH, LDEORLH - LDEORLH variant	FEAT_LSE
01	0	0	1	-	0	011	LDSETH, LDSETAH, LDSETALH, LDSETLH - LDSETLH variant	FEAT_LSE
01	0	0	1	-	0	100	LDSMAXH, LDSMAXAH, LDSMAXALH, LDSMAXLH - LDSMAXLH variant	FEAT_LSE
01	0	0	1	-	0	101	LDSMINH, LDSMINAH, LDSMINALH, LDSMINLH - LDSMINLH variant	FEAT_LSE
01	0	0	1	-	0	110	LDUMAXH, LDUMAXAH, LDUMAXALH, LDUMAXLH - LDUMAXLH variant	FEAT_LSE

Decode fields							Instruction page	Feature
size	V	A	R	Rs	o3	opc		
01	0	0	1	-	0	111	LDUMINH, LDUMINAH, LDUMINALH, LDUMINLH - LDUMINLH variant	FEAT_LSE
01	0	0	1	-	1	000	SWPH, SWPAH, SWPALH, SWPLH - SWPLH variant	FEAT_LSE
01	0	1	0	-	0	000	LDADDH, LDADDAH, LDADDALH, LDADDLH - LDADDAH variant	FEAT_LSE
01	0	1	0	-	0	001	LDCLRH, LDCLRAH, LDCLRALH, LDCLRLH - LDCLRAH variant	FEAT_LSE
01	0	1	0	-	0	010	LDEORH, LDEORAH, LDEORALH, LDEORLH - LDEORAH variant	FEAT_LSE
01	0	1	0	-	0	011	LDSETH, LDSETAH, LDSETALH, LDSETLH - LDSETAH variant	FEAT_LSE
01	0	1	0	-	0	100	LDSMAXH, LDSMAXAH, LDSMAXALH, LDSMAXLH - LDSMAXAH variant	FEAT_LSE
01	0	1	0	-	0	101	LDSMINH, LDSMINAH, LDSMINALH, LDSMINLH - LDSMINAH variant	FEAT_LSE
01	0	1	0	-	0	110	LDUMAXH, LDUMAXAH, LDUMAXALH, LDUMAXLH - LDUMAXAH variant	FEAT_LSE
01	0	1	0	-	0	111	LDUMINH, LDUMINAH, LDUMINALH, LDUMINLH - LDUMINAH variant	FEAT_LSE
01	0	1	0	-	1	000	SWPH, SWPAH, SWPALH, SWPLH - SWPAH variant	FEAT_LSE
01	0	1	0	-	1	100	LDAPRH	FEAT_LRCPC
01	0	1	1	-	0	000	LDADDH, LDADDAH, LDADDALH, LDADDLH - LDADDALH variant	FEAT_LSE
01	0	1	1	-	0	001	LDCLRH, LDCLRAH, LDCLRALH, LDCLRLH - LDCLRALH variant	FEAT_LSE
01	0	1	1	-	0	010	LDEORH, LDEORAH, LDEORALH, LDEORLH - LDEORALH variant	FEAT_LSE
01	0	1	1	-	0	011	LDSETH, LDSETAH, LDSETALH, LDSETLH - LDSETALH variant	FEAT_LSE
01	0	1	1	-	0	100	LDSMAXH, LDSMAXAH, LDSMAXALH, LDSMAXLH - LDSMAXALH variant	FEAT_LSE
01	0	1	1	-	0	101	LDSMINH, LDSMINAH, LDSMINALH, LDSMINLH - LDSMINALH variant	FEAT_LSE
01	0	1	1	-	0	110	LDUMAXH, LDUMAXAH, LDUMAXALH, LDUMAXLH - LDUMAXALH variant	FEAT_LSE
01	0	1	1	-	0	111	LDUMINH, LDUMINAH, LDUMINALH, LDUMINLH - LDUMINALH variant	FEAT_LSE
01	0	1	1	-	1	000	SWPH, SWPAH, SWPALH, SWPLH - SWPALH variant	FEAT_LSE

Decode fields							Instruction page	Feature
size	V	A	R	Rs	o3	opc		
10	0	0	0	-	0	000	LDADD, LDADDA, LDADDAL, LDADDL - 32-bit LDADD variant	FEAT_LSE
10	0	0	0	-	0	001	LDCLR, LDCLRA, LDCLRAL, LDCLRL - 32-bit LDCLR variant	FEAT_LSE
10	0	0	0	-	0	010	LDEOR, LDEORA, LDEORAL, LDEORL - 32-bit LDEOR variant	FEAT_LSE
10	0	0	0	-	0	011	LDSET, LDSETA, LDSETAL, LDSETL - 32-bit LDSET variant	FEAT_LSE
10	0	0	0	-	0	100	LDSMAX, LDSMAXA, LDSMAXAL, LDSMAXL - 32-bit LDSMAX variant	FEAT_LSE
10	0	0	0	-	0	101	LDSMIN, LDSMINA, LDSMINAL, LDSMINL - 32-bit LDSMIN variant	FEAT_LSE
10	0	0	0	-	0	110	LDUMAX, LDUMAXA, LDUMAXAL, LDUMAXL - 32-bit LDUMAX variant	FEAT_LSE
10	0	0	0	-	0	111	LDUMIN, LDUMINA, LDUMINAL, LDUMINL - 32-bit LDUMIN variant	FEAT_LSE
10	0	0	0	-	1	000	SWP, SWPA, SWPAL, SWPL - 32-bit SWP variant	FEAT_LSE
10	0	0	0	-	1	001	Unallocated.	-
10	0	0	0	-	1	010	Unallocated.	-
10	0	0	0	-	1	011	Unallocated.	-
10	0	0	0	-	1	101	Unallocated.	-
10	0	0	1	-	0	000	LDADD, LDADDA, LDADDAL, LDADDL - 32-bit LDADDL variant	FEAT_LSE
10	0	0	1	-	0	001	LDCLR, LDCLRA, LDCLRAL, LDCLRL - 32-bit LDCLRL variant	FEAT_LSE
10	0	0	1	-	0	010	LDEOR, LDEORA, LDEORAL, LDEORL - 32-bit LDEORL variant	FEAT_LSE
10	0	0	1	-	0	011	LDSET, LDSETA, LDSETAL, LDSETL - 32-bit LDSETL variant	FEAT_LSE
10	0	0	1	-	0	100	LDSMAX, LDSMAXA, LDSMAXAL, LDSMAXL - 32-bit LDSMAXL variant	FEAT_LSE
10	0	0	1	-	0	101	LDSMIN, LDSMINA, LDSMINAL, LDSMINL - 32-bit LDSMINL variant	FEAT_LSE
10	0	0	1	-	0	110	LDUMAX, LDUMAXA, LDUMAXAL, LDUMAXL - 32-bit LDUMAXL variant	FEAT_LSE
10	0	0	1	-	0	111	LDUMIN, LDUMINA, LDUMINAL, LDUMINL - 32-bit LDUMINL variant	FEAT_LSE
10	0	0	1	-	1	000	SWP, SWPA, SWPAL, SWPL - 32-bit SWPL variant	FEAT_LSE
10	0	1	0	-	0	000	LDADD, LDADDA, LDADDAL, LDADDL - 32-bit LDADDA variant	FEAT_LSE
10	0	1	0	-	0	001	LDCLR, LDCLRA, LDCLRAL, LDCLRL - 32-bit LDCLRA variant	FEAT_LSE

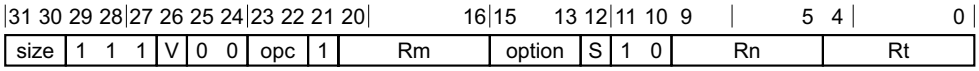
Decode fields							Instruction page	Feature
size	V	A	R	Rs	o3	opc		
10	0	1	0	-	0	010	LDEOR, LDEORA, LDEORAL, LDEORL - 32-bit LDEORA variant	FEAT_LSE
10	0	1	0	-	0	011	LDSET, LDSETA, LDSETAL, LDSETL - 32-bit LDSETA variant	FEAT_LSE
10	0	1	0	-	0	100	LDSMAX, LDSMAXA, LDSMAXAL, LDSMAXL - 32-bit LDSMAXA variant	FEAT_LSE
10	0	1	0	-	0	101	LDSMIN, LDSMINA, LDSMINAL, LDSMINL - 32-bit LDSMINA variant	FEAT_LSE
10	0	1	0	-	0	110	LDUMAX, LDUMAXA, LDUMAXAL, LDUMAXL - 32-bit LDUMAXA variant	FEAT_LSE
10	0	1	0	-	0	111	LDUMIN, LDUMINA, LDUMINAL, LDUMINL - 32-bit LDUMINA variant	FEAT_LSE
10	0	1	0	-	1	000	SWP, SWPA, SWPAL, SWPL - 32-bit SWPA variant	FEAT_LSE
10	0	1	0	-	1	100	LDAPR - 32-bit variant	FEAT_LRCPC
10	0	1	1	-	0	000	LDADD, LDADDA, LDADDAL, LDADDL - 32-bit LDADDAL variant	FEAT_LSE
10	0	1	1	-	0	001	LDCLR, LDCLRA, LDCLRAL, LDCLRL - 32-bit LDCLRAL variant	FEAT_LSE
10	0	1	1	-	0	010	LDEOR, LDEORA, LDEORAL, LDEORL - 32-bit LDEORAL variant	FEAT_LSE
10	0	1	1	-	0	011	LDSET, LDSETA, LDSETAL, LDSETL - 32-bit LDSETAL variant	FEAT_LSE
10	0	1	1	-	0	100	LDSMAX, LDSMAXA, LDSMAXAL, LDSMAXL - 32-bit LDSMAXAL variant	FEAT_LSE
10	0	1	1	-	0	101	LDSMIN, LDSMINA, LDSMINAL, LDSMINL - 32-bit LDSMINAL variant	FEAT_LSE
10	0	1	1	-	0	110	LDUMAX, LDUMAXA, LDUMAXAL, LDUMAXL - 32-bit LDUMAXAL variant	FEAT_LSE
10	0	1	1	-	0	111	LDUMIN, LDUMINA, LDUMINAL, LDUMINL - 32-bit LDUMINAL variant	FEAT_LSE
10	0	1	1	-	1	000	SWP, SWPA, SWPAL, SWPL - 32-bit SWPAL variant	FEAT_LSE
11	0	0	0	-	0	000	LDADD, LDADDA, LDADDAL, LDADDL - 64-bit LDADD variant	FEAT_LSE
11	0	0	0	-	0	001	LDCLR, LDCLRA, LDCLRAL, LDCLRL - 64-bit LDCLR variant	FEAT_LSE
11	0	0	0	-	0	010	LDEOR, LDEORA, LDEORAL, LDEORL - 64-bit LDEOR variant	FEAT_LSE

Decode fields							Instruction page	Feature
size	V	A	R	Rs	o3	opc		
11	0	0	0	-	0	011	LDSET, LDSETA, LDSETAL, LDSETL - 64-bit LDSET variant	FEAT_LSE
11	0	0	0	-	0	100	LDSMAX, LDSMAXA, LDSMAXAL, LDSMAXL - 64-bit LDSMAX variant	FEAT_LSE
11	0	0	0	-	0	101	LDSMIN, LDSMINA, LDSMINAL, LDSMINL - 64-bit LDSMIN variant	FEAT_LSE
11	0	0	0	-	0	110	LDUMAX, LDUMAXA, LDUMAXAL, LDUMAXL - 64-bit LDUMAX variant	FEAT_LSE
11	0	0	0	-	0	111	LDUMIN, LDUMINA, LDUMINAL, LDUMINL - 64-bit LDUMIN variant	FEAT_LSE
11	0	0	0	-	1	000	SWP, SWPA, SWPAL, SWPL - 64-bit SWP variant	FEAT_LSE
11	0	0	0	-	1	010	ST64BV0	FEAT_LS64_V
11	0	0	0	-	1	011	ST64BV	FEAT_LS64_V
11	0	0	0	11111	1	001	ST64B	FEAT_LS64
11	0	0	0	11111	1	101	LD64B	FEAT_LS64
11	0	0	1	-	0	000	LDADD, LDADDA, LDADDAL, LDADDL - 64-bit LDADDL variant	FEAT_LSE
11	0	0	1	-	0	001	LDCLR, LDCLRA, LDCLRAL, LDCLRL - 64-bit LDCLRL variant	FEAT_LSE
11	0	0	1	-	0	010	LDEOR, LDEORA, LDEORAL, LDEORL - 64-bit LDEORL variant	FEAT_LSE
11	0	0	1	-	0	011	LDSET, LDSETA, LDSETAL, LDSETL - 64-bit LDSETL variant	FEAT_LSE
11	0	0	1	-	0	100	LDSMAX, LDSMAXA, LDSMAXAL, LDSMAXL - 64-bit LDSMAXL variant	FEAT_LSE
11	0	0	1	-	0	101	LDSMIN, LDSMINA, LDSMINAL, LDSMINL - 64-bit LDSMINL variant	FEAT_LSE
11	0	0	1	-	0	110	LDUMAX, LDUMAXA, LDUMAXAL, LDUMAXL - 64-bit LDUMAXL variant	FEAT_LSE
11	0	0	1	-	0	111	LDUMIN, LDUMINA, LDUMINAL, LDUMINL - 64-bit LDUMINL variant	FEAT_LSE
11	0	0	1	-	1	000	SWP, SWPA, SWPAL, SWPL - 64-bit SWPL variant	FEAT_LSE

Decode fields							Instruction page	Feature
size	V	A	R	Rs	o3	opc		
11	0	1	0	-	0	000	LDADD, LDADDA, LDADDAL, LDADDL - 64-bit LDADDA variant	FEAT_LSE
11	0	1	0	-	0	001	LDCLR, LDCLRA, LDCLRAL, LDCLRL - 64-bit LDCLRA variant	FEAT_LSE
11	0	1	0	-	0	010	LDEOR, LDEORA, LDEORAL, LDEORL - 64-bit LDEORA variant	FEAT_LSE
11	0	1	0	-	0	011	LDSET, LDSETA, LDSETAL, LDSETL - 64-bit LDSETA variant	FEAT_LSE
11	0	1	0	-	0	100	LDSMAX, LDSMAXA, LDSMAXAL, LDSMAXL - 64-bit LDSMAXA variant	FEAT_LSE
11	0	1	0	-	0	101	LDSMIN, LDSMINA, LDSMINAL, LDSMINL - 64-bit LDSMINA variant	FEAT_LSE
11	0	1	0	-	0	110	LDUMAX, LDUMAXA, LDUMAXAL, LDUMAXL - 64-bit LDUMAXA variant	FEAT_LSE
11	0	1	0	-	0	111	LDUMIN, LDUMINA, LDUMINAL, LDUMINL - 64-bit LDUMINA variant	FEAT_LSE
11	0	1	0	-	1	000	SWP, SWPA, SWPAL, SWPL - 64-bit SWPA variant	FEAT_LSE
11	0	1	0	-	1	100	LDAPR - 64-bit variant	FEAT_LRCPC
11	0	1	1	-	0	000	LDADD, LDADDA, LDADDAL, LDADDL - 64-bit LDADDAL variant	FEAT_LSE
11	0	1	1	-	0	001	LDCLR, LDCLRA, LDCLRAL, LDCLRL - 64-bit LDCLRAL variant	FEAT_LSE
11	0	1	1	-	0	010	LDEOR, LDEORA, LDEORAL, LDEORL - 64-bit LDEORAL variant	FEAT_LSE
11	0	1	1	-	0	011	LDSET, LDSETA, LDSETAL, LDSETL - 64-bit LDSETAL variant	FEAT_LSE
11	0	1	1	-	0	100	LDSMAX, LDSMAXA, LDSMAXAL, LDSMAXL - 64-bit LDSMAXAL variant	FEAT_LSE
11	0	1	1	-	0	101	LDSMIN, LDSMINA, LDSMINAL, LDSMINL - 64-bit LDSMINAL variant	FEAT_LSE
11	0	1	1	-	0	110	LDUMAX, LDUMAXA, LDUMAXAL, LDUMAXL - 64-bit LDUMAXAL variant	FEAT_LSE
11	0	1	1	-	0	111	LDUMIN, LDUMINA, LDUMINAL, LDUMINL - 64-bit LDUMINAL variant	FEAT_LSE
11	0	1	1	-	1	000	SWP, SWPA, SWPAL, SWPL - 64-bit SWPAL variant	FEAT_LSE

Load/store register (register offset)

This section describes the encoding of the Load/store register (register offset) instruction class. The encodings in this section are decoded from *Loads and Stores* on page C4-295.

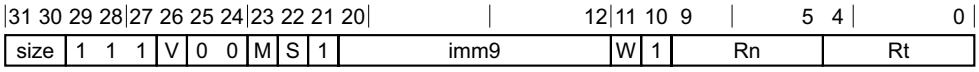


Decode fields				Instruction page
size	V	opc	option	
x1	1	1x	-	Unallocated.
00	0	00	!= 011	STRB (register) - Extended register variant
00	0	00	011	STRB (register) - Shifted register variant
00	0	01	!= 011	LDRB (register) - Extended register variant
00	0	01	011	LDRB (register) - Shifted register variant
00	0	10	!= 011	LDRSB (register) - 64-bit with extended register offset variant
00	0	10	011	LDRSB (register) - 64-bit with shifted register offset variant
00	0	11	!= 011	LDRSB (register) - 32-bit with extended register offset variant
00	0	11	011	LDRSB (register) - 32-bit with shifted register offset variant
00	1	00	!= 011	STR (register, SIMD&FP)
00	1	00	011	STR (register, SIMD&FP)
00	1	01	!= 011	LDR (register, SIMD&FP)
00	1	01	011	LDR (register, SIMD&FP)
00	1	10	-	STR (register, SIMD&FP)
00	1	11	-	LDR (register, SIMD&FP)
01	0	00	-	STRH (register)
01	0	01	-	LDRH (register)
01	0	10	-	LDRSH (register) - 64-bit variant
01	0	11	-	LDRSH (register) - 32-bit variant
01	1	00	-	STR (register, SIMD&FP)
01	1	01	-	LDR (register, SIMD&FP)
1x	0	11	-	Unallocated.
1x	1	1x	-	Unallocated.
10	0	00	-	STR (register) - 32-bit variant
10	0	01	-	LDR (register) - 32-bit variant
10	0	10	-	LDRSW (register)

Decode fields				Instruction page
size	V	opc	option	
10	1	00	-	STR (register, SIMD&FP)
10	1	01	-	LDR (register, SIMD&FP)
11	0	00	-	STR (register) - 64-bit variant
11	0	01	-	LDR (register) - 64-bit variant
11	0	10	-	PRFM (register)
11	1	00	-	STR (register, SIMD&FP)
11	1	01	-	LDR (register, SIMD&FP)

Load/store register (pac)

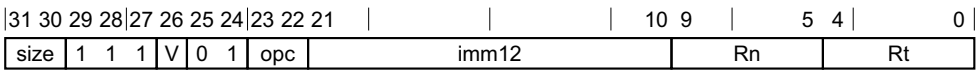
This section describes the encoding of the Load/store register (pac) instruction class. The encodings in this section are decoded from *Loads and Stores* on page C4-295.



Decode fields				Instruction page	Feature
size	V	M	W		
!= 11	-	-	-	Unallocated.	-
11	0	0	0	LDRAA, LDRAB - Key A, offset variant	FEAT_PAuth
11	0	0	1	LDRAA, LDRAB - Key A, pre-indexed variant	FEAT_PAuth
11	0	1	0	LDRAA, LDRAB - Key B, offset variant	FEAT_PAuth
11	0	1	1	LDRAA, LDRAB - Key B, pre-indexed variant	FEAT_PAuth
11	1	-	-	Unallocated.	-

Load/store register (unsigned immediate)

This section describes the encoding of the Load/store register (unsigned immediate) instruction class. The encodings in this section are decoded from *Loads and Stores* on page C4-295.



Decode fields			Instruction page
size	V	opc	
x1	1	1x	Unallocated.
00	0	00	STRB (immediate)
00	0	01	LDRB (immediate)
00	0	10	LDRSB (immediate) - 64-bit variant
00	0	11	LDRSB (immediate) - 32-bit variant
00	1	00	STR (immediate, SIMD&FP) - 8-bit variant
00	1	01	LDR (immediate, SIMD&FP) - 8-bit variant
00	1	10	STR (immediate, SIMD&FP) - 128-bit variant
00	1	11	LDR (immediate, SIMD&FP) - 128-bit variant
01	0	00	STRH (immediate)
01	0	01	LDRH (immediate)
01	0	10	LDRSH (immediate) - 64-bit variant
01	0	11	LDRSH (immediate) - 32-bit variant
01	1	00	STR (immediate, SIMD&FP) - 16-bit variant
01	1	01	LDR (immediate, SIMD&FP) - 16-bit variant
1x	0	11	Unallocated.
1x	1	1x	Unallocated.
10	0	00	STR (immediate) - 32-bit variant
10	0	01	LDR (immediate) - 32-bit variant
10	0	10	LDRSW (immediate)
10	1	00	STR (immediate, SIMD&FP) - 32-bit variant
10	1	01	LDR (immediate, SIMD&FP) - 32-bit variant
11	0	00	STR (immediate) - 64-bit variant
11	0	01	LDR (immediate) - 64-bit variant
11	0	10	PRFM (immediate)
11	1	00	STR (immediate, SIMD&FP) - 64-bit variant
11	1	01	LDR (immediate, SIMD&FP) - 64-bit variant

C4.1.5 Data Processing -- Register

This section describes the encoding of the Data Processing -- Register group. The encodings in this section are decoded from *A64 instruction set encoding* on page C4-280.

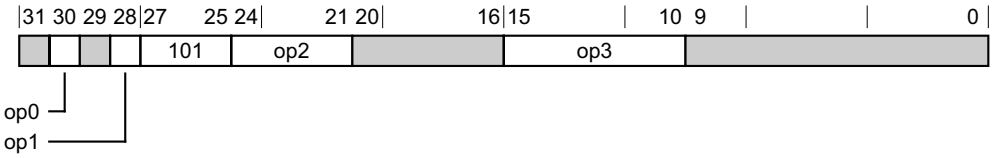
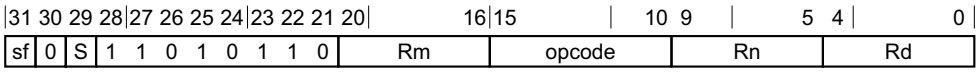


Table C4-6 Encoding table for the Data Processing -- Register group

Decode fields				Decode group or instruction page
op0	op1	op2	op3	
0	1	0110	-	*Data-processing (2 source)*
1	1	0110	-	*Data-processing (1 source)* on page C4-330
-	0	0xxx	-	*Logical (shifted register)* on page C4-332
-	0	1xx0	-	*Add/subtract (shifted register)* on page C4-333
-	0	1xx1	-	*Add/subtract (extended register)* on page C4-333
-	1	0000	000000	*Add/subtract (with carry)* on page C4-334
-	1	0000	000011	Unallocated.
-	1	0000	0001xx	Unallocated.
-	1	0000	001xxx	Unallocated.
-	1	0000	x00001	*Rotate right into flags* on page C4-334
-	1	0000	xx0010	*Evaluate into flags* on page C4-335
-	1	0010	xxxx0x	*Conditional compare (register)* on page C4-335
-	1	0010	xxxx1x	*Conditional compare (immediate)* on page C4-336
-	1	0100	-	*Conditional select* on page C4-336
-	1	0xx1	-	Unallocated.
-	1	1xxx	-	*Data-processing (3 source)* on page C4-337

Data-processing (2 source)

This section describes the encoding of the Data-processing (2 source) instruction class. The encodings in this section are decoded from *Data Processing -- Register*.

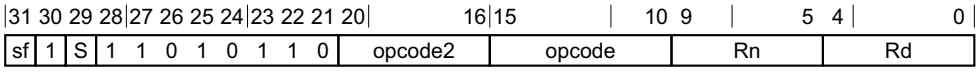


Decode fields			Instruction page	Feature
sf	S	opcode		
-	-	000001	Unallocated.	-
-	-	011xxx	Unallocated.	-
-	-	1xxxxx	Unallocated.	-
-	0	00011x	Unallocated.	-
-	0	001101	Unallocated.	-
-	0	00111x	Unallocated.	-
-	1	00001x	Unallocated.	-
-	1	0001xx	Unallocated.	-
-	1	001xxx	Unallocated.	-
-	1	01xxxx	Unallocated.	-
0	-	000000	Unallocated.	-
0	0	000010	UDIV - 32-bit variant	-
0	0	000011	SDIV - 32-bit variant	-
0	0	00010x	Unallocated.	-
0	0	001000	LSLV - 32-bit variant	-
0	0	001001	LSRV - 32-bit variant	-
0	0	001010	ASRV - 32-bit variant	-
0	0	001011	RORV - 32-bit variant	-
0	0	001100	Unallocated.	-
0	0	010x11	Unallocated.	-
0	0	010000	CRC32B, CRC32H, CRC32W, CRC32X - CRC32B variant	-
0	0	010001	CRC32B, CRC32H, CRC32W, CRC32X - CRC32H variant	-
0	0	010010	CRC32B, CRC32H, CRC32W, CRC32X - CRC32W variant	-
0	0	010100	CRC32CB, CRC32CH, CRC32CW, CRC32CX - CRC32CB variant	-
0	0	010101	CRC32CB, CRC32CH, CRC32CW, CRC32CX - CRC32CH variant	-
0	0	010110	CRC32CB, CRC32CH, CRC32CW, CRC32CX - CRC32CW variant	-
1	0	000000	SUBP	FEAT_MTE
1	0	000010	UDIV - 64-bit variant	-

Decode fields			Instruction page	Feature
sf	S	opcode		
1	0	000011	SDIV - 64-bit variant	-
1	0	000100	IRG	FEAT_MTE
1	0	000101	GMI	FEAT_MTE
1	0	001000	LSLV - 64-bit variant	-
1	0	001001	LSRV - 64-bit variant	-
1	0	001010	ASRV - 64-bit variant	-
1	0	001011	RORV - 64-bit variant	-
1	0	001100	PACGA	FEAT_PAuth
1	0	010xx0	Unallocated.	-
1	0	010x0x	Unallocated.	-
1	0	010011	CRC32B, CRC32H, CRC32W, CRC32X - CRC32X variant	-
1	0	010111	CRC32CB, CRC32CH, CRC32CW, CRC32CX - CRC32CX variant	-
1	1	000000	SUBPS	FEAT_MTE

Data-processing (1 source)

This section describes the encoding of the Data-processing (1 source) instruction class. The encodings in this section are decoded from *Data Processing -- Register* on page C4-328.



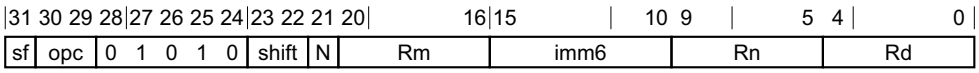
Decode fields					Instruction page	Feature
sf	S	opcode2	opcode	Rn		
-	-	-	1xxxxx	-	Unallocated.	-
-	-	xxx1x	-	-	Unallocated.	-
-	-	xx1xx	-	-	Unallocated.	-
-	-	x1xxx	-	-	Unallocated.	-
-	-	1xxxx	-	-	Unallocated.	-
-	0	00000	00011x	-	Unallocated.	-
-	0	00000	001xxx	-	Unallocated.	-
-	0	00000	01xxxx	-	Unallocated.	-
-	1	-	-	-	Unallocated.	-
0	-	00001	-	-	Unallocated.	-

Decode fields					Instruction page	Feature
sf	S	opcode2	opcode	Rn		
0	0	00000	000000	-	RBIT - 32-bit variant	-
0	0	00000	000001	-	REV16 - 32-bit variant	-
0	0	00000	000010	-	REV - 32-bit variant	-
0	0	00000	000011	-	Unallocated.	-
0	0	00000	000100	-	CLZ - 32-bit variant	-
0	0	00000	000101	-	CLS - 32-bit variant	-
1	0	00000	000000	-	RBIT - 64-bit variant	-
1	0	00000	000001	-	REV16 - 64-bit variant	-
1	0	00000	000010	-	REV32	-
1	0	00000	000011	-	REV - 64-bit variant	-
1	0	00000	000100	-	CLZ - 64-bit variant	-
1	0	00000	000101	-	CLS - 64-bit variant	-
1	0	00001	000000	-	PACIA, PACIA1716, PACIASP, PACIAZ, PACIZA - PACIA variant	FEAT_PAuth
1	0	00001	000001	-	PACIB, PACIB1716, PACIBSP, PACIBZ, PACIZB - PACIB variant	FEAT_PAuth
1	0	00001	000010	-	PACDA, PACDZA - PACDA variant	FEAT_PAuth
1	0	00001	000011	-	PACDB, PACDZB - PACDB variant	FEAT_PAuth
1	0	00001	000100	-	AUTIA, AUTIA1716, AUTIASP, AUTIAZ, AUTIZA - AUTIA variant	FEAT_PAuth
1	0	00001	000101	-	AUTIB, AUTIB1716, AUTIBSP, AUTIBZ, AUTIZB - AUTIB variant	FEAT_PAuth
1	0	00001	000110	-	AUTDA, AUTDZA - AUTDA variant	FEAT_PAuth
1	0	00001	000111	-	AUTDB, AUTDZB - AUTDB variant	FEAT_PAuth
1	0	00001	001000	11111	PACIA, PACIA1716, PACIASP, PACIAZ, PACIZA - PACIZA variant	FEAT_PAuth
1	0	00001	001001	11111	PACIB, PACIB1716, PACIBSP, PACIBZ, PACIZB - PACIZB variant	FEAT_PAuth
1	0	00001	001010	11111	PACDA, PACDZA - PACDZA variant	FEAT_PAuth
1	0	00001	001011	11111	PACDB, PACDZB - PACDZB variant	FEAT_PAuth
1	0	00001	001100	11111	AUTIA, AUTIA1716, AUTIASP, AUTIAZ, AUTIZA - AUTIZA variant	FEAT_PAuth
1	0	00001	001101	11111	AUTIB, AUTIB1716, AUTIBSP, AUTIBZ, AUTIZB - AUTIZB variant	FEAT_PAuth
1	0	00001	001110	11111	AUTDA, AUTDZA - AUTDZA variant	FEAT_PAuth

Decode fields					Instruction page	Feature
sf	S	opcode2	opcode	Rn		
1	0	00001	001111	11111	AUTDB, AUTDZB - AUTDZB variant	FEAT_PAuth
1	0	00001	010000	11111	XPACD, XPACI, XPACLRI - XPACI variant	FEAT_PAuth
1	0	00001	010001	11111	XPACD, XPACI, XPACLRI - XPACD variant	FEAT_PAuth
1	0	00001	01001x	-	Unallocated.	-
1	0	00001	0101xx	-	Unallocated.	-
1	0	00001	011xxx	-	Unallocated.	-

Logical (shifted register)

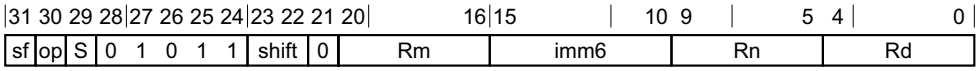
This section describes the encoding of the Logical (shifted register) instruction class. The encodings in this section are decoded from *Data Processing -- Register* on page C4-328.



Decode fields				Instruction page
sf	opc	N	imm6	
0	-	-	1xxxxx	Unallocated.
0	00	0	-	AND (shifted register) - 32-bit variant
0	00	1	-	BIC (shifted register) - 32-bit variant
0	01	0	-	ORR (shifted register) - 32-bit variant
0	01	1	-	ORN (shifted register) - 32-bit variant
0	10	0	-	EOR (shifted register) - 32-bit variant
0	10	1	-	EON (shifted register) - 32-bit variant
0	11	0	-	ANDS (shifted register) - 32-bit variant
0	11	1	-	BICS (shifted register) - 32-bit variant
1	00	0	-	AND (shifted register) - 64-bit variant
1	00	1	-	BIC (shifted register) - 64-bit variant
1	01	0	-	ORR (shifted register) - 64-bit variant
1	01	1	-	ORN (shifted register) - 64-bit variant
1	10	0	-	EOR (shifted register) - 64-bit variant
1	10	1	-	EON (shifted register) - 64-bit variant
1	11	0	-	ANDS (shifted register) - 64-bit variant
1	11	1	-	BICS (shifted register) - 64-bit variant

Add/subtract (shifted register)

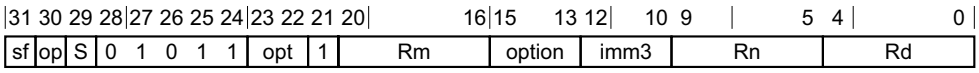
This section describes the encoding of the Add/subtract (shifted register) instruction class. The encodings in this section are decoded from *Data Processing -- Register* on page C4-328.



Decode fields					Instruction page
sf	op	S	shift	imm6	
-	-	-	11	-	Unallocated.
0	-	-	-	1xxxxx	Unallocated.
0	0	0	-	-	ADD (shifted register) - 32-bit variant
0	0	1	-	-	ADDS (shifted register) - 32-bit variant
0	1	0	-	-	SUB (shifted register) - 32-bit variant
0	1	1	-	-	SUBS (shifted register) - 32-bit variant
1	0	0	-	-	ADD (shifted register) - 64-bit variant
1	0	1	-	-	ADDS (shifted register) - 64-bit variant
1	1	0	-	-	SUB (shifted register) - 64-bit variant
1	1	1	-	-	SUBS (shifted register) - 64-bit variant

Add/subtract (extended register)

This section describes the encoding of the Add/subtract (extended register) instruction class. The encodings in this section are decoded from *Data Processing -- Register* on page C4-328.

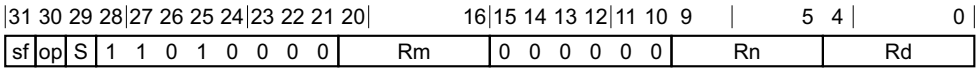


Decode fields					Instruction page
sf	op	S	opt	imm3	
-	-	-	-	1x1	Unallocated.
-	-	-	-	11x	Unallocated.
-	-	-	x1	-	Unallocated.
-	-	-	1x	-	Unallocated.
0	0	0	00	-	ADD (extended register) - 32-bit variant
0	0	1	00	-	ADDS (extended register) - 32-bit variant
0	1	0	00	-	SUB (extended register) - 32-bit variant

Decode fields					Instruction page
sf	op	S	opt	imm3	
0	1	1	00	-	SUBS (extended register) - 32-bit variant
1	0	0	00	-	ADD (extended register) - 64-bit variant
1	0	1	00	-	ADDS (extended register) - 64-bit variant
1	1	0	00	-	SUB (extended register) - 64-bit variant
1	1	1	00	-	SUBS (extended register) - 64-bit variant

Add/subtract (with carry)

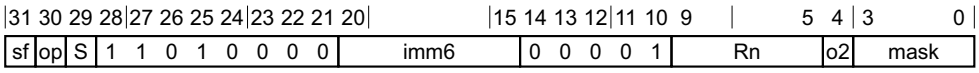
This section describes the encoding of the Add/subtract (with carry) instruction class. The encodings in this section are decoded from *Data Processing -- Register* on page C4-328.



Decode fields			Instruction page
sf	op	S	
0	0	0	ADC - 32-bit variant
0	0	1	ADCS - 32-bit variant
0	1	0	SBC - 32-bit variant
0	1	1	SBCS - 32-bit variant
1	0	0	ADC - 64-bit variant
1	0	1	ADCS - 64-bit variant
1	1	0	SBC - 64-bit variant
1	1	1	SBCS - 64-bit variant

Rotate right into flags

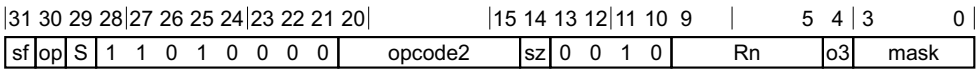
This section describes the encoding of the Rotate right into flags instruction class. The encodings in this section are decoded from *Data Processing -- Register* on page C4-328.



Decode fields				Instruction page	Feature
sf	op	S	o2		
0	-	-	-	Unallocated.	-
1	0	0	-	Unallocated.	-
1	0	1	0	RMIF	FEAT_FlagM
1	0	1	1	Unallocated.	-
1	1	-	-	Unallocated.	-

Evaluate into flags

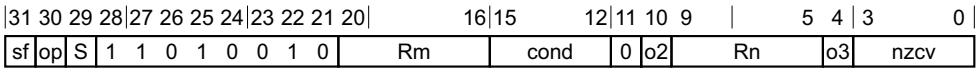
This section describes the encoding of the Evaluate into flags instruction class. The encodings in this section are decoded from *Data Processing -- Register* on page C4-328.



Decode fields							Instruction page	Feature
sf	op	S	opcode2	sz	o3	mask		
0	0	0	-	-	-	-	Unallocated.	-
0	0	1	!= 000000	-	-	-	Unallocated.	-
0	0	1	000000	-	0	!= 1101	Unallocated.	-
0	0	1	000000	-	1	-	Unallocated.	-
0	0	1	000000	0	0	1101	SETF8, SETF16 - SETF8 variant	FEAT_FlagM
0	0	1	000000	1	0	1101	SETF8, SETF16 - SETF16 variant	FEAT_FlagM
0	1	-	-	-	-	-	Unallocated.	-
1	-	-	-	-	-	-	Unallocated.	-

Conditional compare (register)

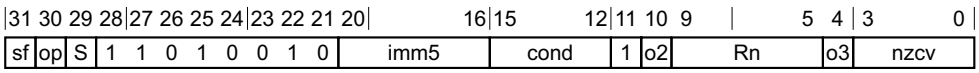
This section describes the encoding of the Conditional compare (register) instruction class. The encodings in this section are decoded from *Data Processing -- Register* on page C4-328.



Decode fields					Instruction page
sf	op	S	o2	o3	
-	-	-	-	1	Unallocated.
-	-	-	1	-	Unallocated.
-	-	0	-	-	Unallocated.
0	0	1	0	0	CCMN (register) - 32-bit variant
0	1	1	0	0	CCMP (register) - 32-bit variant
1	0	1	0	0	CCMN (register) - 64-bit variant
1	1	1	0	0	CCMP (register) - 64-bit variant

Conditional compare (immediate)

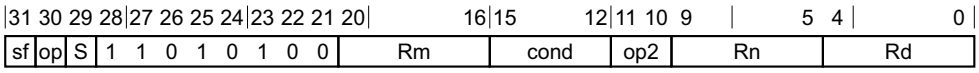
This section describes the encoding of the Conditional compare (immediate) instruction class. The encodings in this section are decoded from *Data Processing -- Register* on page C4-328.



Decode fields					Instruction page
sf	op	S	o2	o3	
-	-	-	-	1	Unallocated.
-	-	-	1	-	Unallocated.
-	-	0	-	-	Unallocated.
0	0	1	0	0	CCMN (immediate) - 32-bit variant
0	1	1	0	0	CCMP (immediate) - 32-bit variant
1	0	1	0	0	CCMN (immediate) - 64-bit variant
1	1	1	0	0	CCMP (immediate) - 64-bit variant

Conditional select

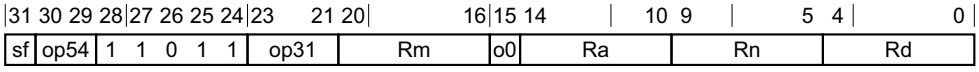
This section describes the encoding of the Conditional select instruction class. The encodings in this section are decoded from *Data Processing -- Register* on page C4-328.



Decode fields				Instruction page
sf	op	S	op2	
-	-	-	1x	Unallocated.
-	-	1	-	Unallocated.
0	0	0	00	CSEL - 32-bit variant
0	0	0	01	CSINC - 32-bit variant
0	1	0	00	CSINV - 32-bit variant
0	1	0	01	CSNEG - 32-bit variant
1	0	0	00	CSEL - 64-bit variant
1	0	0	01	CSINC - 64-bit variant
1	1	0	00	CSINV - 64-bit variant
1	1	0	01	CSNEG - 64-bit variant

Data-processing (3 source)

This section describes the encoding of the Data-processing (3 source) instruction class. The encodings in this section are decoded from *Data Processing -- Register* on page C4-328.



Decode fields				Instruction page
sf	op54	op31	o0	
-	00	010	1	Unallocated.
-	00	011	-	Unallocated.
-	00	100	-	Unallocated.
-	00	110	1	Unallocated.
-	00	111	-	Unallocated.
-	01	-	-	Unallocated.
-	1x	-	-	Unallocated.
0	00	000	0	MADD - 32-bit variant
0	00	000	1	MSUB - 32-bit variant

Decode fields				Instruction page
sf	op54	op31	o0	
0	00	001	0	Unallocated.
0	00	001	1	Unallocated.
0	00	010	0	Unallocated.
0	00	101	0	Unallocated.
0	00	101	1	Unallocated.
0	00	110	0	Unallocated.
1	00	000	0	MADD - 64-bit variant
1	00	000	1	MSUB - 64-bit variant
1	00	001	0	SMADDL
1	00	001	1	SMSUBL
1	00	010	0	SMULH
1	00	101	0	UMADDL
1	00	101	1	UMSUBL
1	00	110	0	UMULH

C4.1.6 Data Processing -- Scalar Floating-Point and Advanced SIMD

This section describes the encoding of the Data Processing -- Scalar Floating-Point and Advanced SIMD group. The encodings in this section are decoded from *A64 instruction set encoding* on page C4-280.

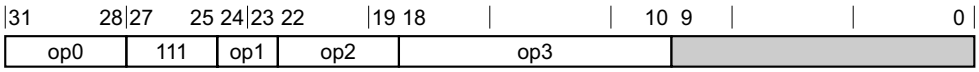


Table C4-7 Encoding table for the Data Processing -- Scalar Floating-Point and Advanced SIMD group

Decode fields				Decode group or instruction page	Feature
op0	op1	op2	op3		
0000	0x	x101	00xxxxx10	Unallocated.	-
0010	0x	x101	00xxxxx10	Unallocated.	-
0100	0x	x101	00xxxxx10	*Cryptographic AES* on page C4-341	-
0101	0x	x0xx	xxx0xxx00	*Cryptographic three-register SHA* on page C4-341	-
0101	0x	x0xx	xxx0xxx10	Unallocated.	-
0101	0x	x101	00xxxxx10	*Cryptographic two-register SHA* on page C4-342	-
0110	0x	x101	00xxxxx10	Unallocated.	-
0111	0x	x0xx	xxx0xxxx0	Unallocated.	-

Table C4-7 Encoding table for the Data Processing -- Scalar Floating-Point and Advanced SIMD group (continued)

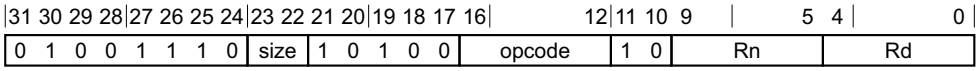
Decode fields				Decode group or instruction page	Feature
op0	op1	op2	op3		
0111	0x	x101	00xxxx10	Unallocated.	-
01x1	00	00xx	xxx0xxx1	*Advanced SIMD scalar copy* on page C4-343	-
01x1	01	00xx	xxx0xxx1	Unallocated.	-
01x1	0x	0111	00xxxx10	Unallocated.	-
01x1	0x	10xx	xxx00xx1	*Advanced SIMD scalar three same FP16* on page C4-343	-
01x1	0x	10xx	xxx01xx1	Unallocated.	-
01x1	0x	1111	00xxxx10	*Advanced SIMD scalar two-register miscellaneous FP16* on page C4-344	-
01x1	0x	x0xx	xxx1xxx0	Unallocated.	-
01x1	0x	x0xx	xxx1xxx1	*Advanced SIMD scalar three same extra* on page C4-345	-
01x1	0x	x100	00xxxx10	*Advanced SIMD scalar two-register miscellaneous* on page C4-346	-
01x1	0x	x110	00xxxx10	*Advanced SIMD scalar pairwise* on page C4-348	-
01x1	0x	x1xx	1xxxxx10	Unallocated.	-
01x1	0x	x1xx	x1xxxx10	Unallocated.	-
01x1	0x	x1xx	xxxxxx00	*Advanced SIMD scalar three different* on page C4-349	-
01x1	0x	x1xx	xxxxxxx1	*Advanced SIMD scalar three same* on page C4-349	-
01x1	10	-	xxxxxxx1	*Advanced SIMD scalar shift by immediate* on page C4-352	-
01x1	11	-	xxxxxxx1	Unallocated.	-
01x1	1x	-	xxxxxxx0	*Advanced SIMD scalar x indexed element* on page C4-353	-
0x00	0x	x0xx	xxx0xx00	*Advanced SIMD table lookup* on page C4-355	-
0x00	0x	x0xx	xxx0xx10	*Advanced SIMD permute* on page C4-355	-
0x10	0x	x0xx	xxx0xxx0	*Advanced SIMD extract* on page C4-356	-
0xx0	00	00xx	xxx0xxx1	*Advanced SIMD copy* on page C4-356	-
0xx0	01	00xx	xxx0xxx1	Unallocated.	-
0xx0	0x	0111	00xxxx10	Unallocated.	-
0xx0	0x	10xx	xxx00xx1	*Advanced SIMD three same (FP16)* on page C4-357	-
0xx0	0x	10xx	xxx01xx1	Unallocated.	-
0xx0	0x	1111	00xxxx10	*Advanced SIMD two-register miscellaneous (FP16)* on page C4-358	-
0xx0	0x	x0xx	xxx1xxx0	Unallocated.	-
0xx0	0x	x0xx	xxx1xxx1	*Advanced SIMD three-register extension* on page C4-360	-
0xx0	0x	x100	00xxxx10	*Advanced SIMD two-register miscellaneous* on page C4-361	-
0xx0	0x	x110	00xxxx10	*Advanced SIMD across lanes* on page C4-364	-

Table C4-7 Encoding table for the Data Processing -- Scalar Floating-Point and Advanced SIMD group (continued)

Decode fields				Decode group or instruction page	Feature
op0	op1	op2	op3		
0xx0	0x	x1xx	1xxxxxx10	Unallocated.	-
0xx0	0x	x1xx	x1xxxxx10	Unallocated.	-
0xx0	0x	x1xx	xxxxxxx00	*Advanced SIMD three different* on page C4-365	-
0xx0	0x	x1xx	xxxxxxxx1	*Advanced SIMD three same* on page C4-366	-
0xx0	10	0000	xxxxxxxx1	*Advanced SIMD modified immediate* on page C4-370	-
0xx0	10	!= 0000	xxxxxxxx1	*Advanced SIMD shift by immediate* on page C4-371	-
0xx0	11	-	xxxxxxxx1	Unallocated.	-
0xx0	1x	-	xxxxxxxx0	*Advanced SIMD vector x indexed element* on page C4-372	-
1100	00	10xx	xxx10xxxx	*Cryptographic three-register, imm2* on page C4-374	-
1100	00	11xx	xxx1x00xx	*Cryptographic three-register SHA 512* on page C4-375	-
1100	00	-	xxx0xxxxx	*Cryptographic four-register* on page C4-376	-
1100	01	00xx	-	XAR	FEAT_SHA3
1100	01	1000	0001000xx	*Cryptographic two-register SHA 512* on page C4-376	-
11x1	-	-	-	Unallocated.	-
1xx0	1x	-	-	Unallocated.	-
x0x1	0x	x0xx	-	*Conversion between floating-point and fixed-point* on page C4-377	-
x0x1	0x	x1xx	xxx000000	*Conversion between floating-point and integer* on page C4-379	-
x0x1	0x	x1xx	xxx100000	Unallocated.	-
x0x1	0x	x1xx	xxxx10000	*Floating-point data-processing (1 source)* on page C4-382	-
x0x1	0x	x1xx	xxxxx1000	*Floating-point compare* on page C4-385	-
x0x1	0x	x1xx	xxxxxx100	*Floating-point immediate* on page C4-386	-
x0x1	0x	x1xx	xxxxxxx01	*Floating-point conditional compare* on page C4-386	-
x0x1	0x	x1xx	xxxxxxx10	*Floating-point data-processing (2 source)* on page C4-387	-
x0x1	0x	x1xx	xxxxxxx11	*Floating-point conditional select* on page C4-388	-
x0x1	1x	-	-	*Floating-point data-processing (3 source)* on page C4-389	-

Cryptographic AES

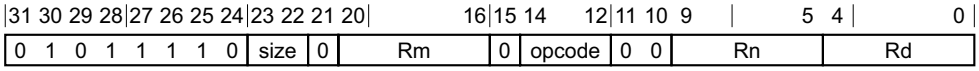
This section describes the encoding of the Cryptographic AES instruction class. The encodings in this section are decoded from *Data Processing -- Scalar Floating-Point and Advanced SIMD* on page C4-338.



Decode fields		Instruction page
size	opcode	
-	x1xxx	Unallocated.
-	000xx	Unallocated.
-	1xxxx	Unallocated.
x1	-	Unallocated.
00	00100	AESE
00	00101	AESD
00	00110	AESMC
00	00111	AESIMC
1x	-	Unallocated.

Cryptographic three-register SHA

This section describes the encoding of the Cryptographic three-register SHA instruction class. The encodings in this section are decoded from *Data Processing -- Scalar Floating-Point and Advanced SIMD* on page C4-338.

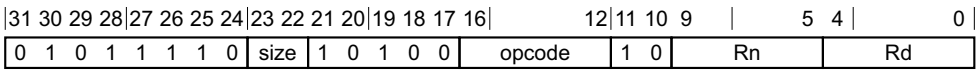


Decode fields		Instruction page
size	opcode	
-	111	Unallocated.
x1	-	Unallocated.
00	000	SHA1C
00	001	SHA1P
00	010	SHA1M
00	011	SHA1SU0
00	100	SHA256H

Decode fields		Instruction page
size	opcode	
00	101	SHA256H2
00	110	SHA256SU1
1x	-	Unallocated.

Cryptographic two-register SHA

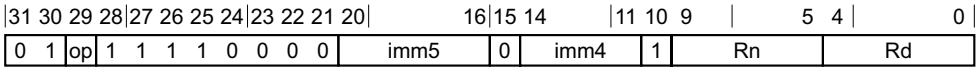
This section describes the encoding of the Cryptographic two-register SHA instruction class. The encodings in this section are decoded from *Data Processing -- Scalar Floating-Point and Advanced SIMD* on page C4-338.



Decode fields		Instruction page
size	opcode	
-	xx1xx	Unallocated.
-	x1xxx	Unallocated.
-	1xxxx	Unallocated.
x1	-	Unallocated.
00	00000	SHA1H
00	00001	SHA1SU1
00	00010	SHA256SU0
00	00011	Unallocated.
1x	-	Unallocated.

Advanced SIMD scalar copy

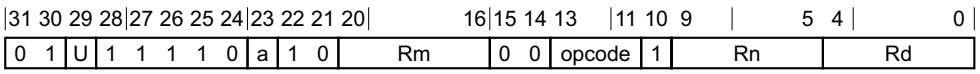
This section describes the encoding of the Advanced SIMD scalar copy instruction class. The encodings in this section are decoded from *Data Processing -- Scalar Floating-Point and Advanced SIMD* on page C4-338.



Decode fields		Instruction page
op	imm4	
0	xxx1	Unallocated.
0	xx1x	Unallocated.
0	x1xx	Unallocated.
0	0000	DUP (element)
0	1xxx	Unallocated.
1	-	Unallocated.

Advanced SIMD scalar three same FP16

This section describes the encoding of the Advanced SIMD scalar three same FP16 instruction class. The encodings in this section are decoded from *Data Processing -- Scalar Floating-Point and Advanced SIMD* on page C4-338.

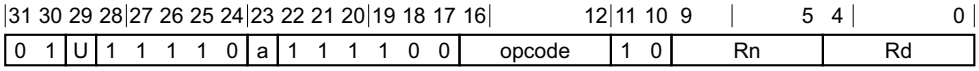


Decode fields			Instruction page	Feature
U	a	opcode		
-	-	110	Unallocated.	-
-	1	011	Unallocated.	-
0	0	011	FMULX	FEAT_FP16
0	0	100	FCMEQ (register)	FEAT_FP16
0	0	101	Unallocated.	-
0	0	111	FRECPS	FEAT_FP16
0	1	100	Unallocated.	-
0	1	101	Unallocated.	-
0	1	111	FRSQRTS	FEAT_FP16
1	0	011	Unallocated.	-
1	0	100	FCMGE (register)	FEAT_FP16

Decode fields			Instruction page	Feature
U	a	opcode		
1	0	101	FACGE	FEAT_FP16
1	0	111	Unallocated.	-
1	1	010	FABD	FEAT_FP16
1	1	100	FCMGT (register)	FEAT_FP16
1	1	101	FACGT	FEAT_FP16
1	1	111	Unallocated.	-

Advanced SIMD scalar two-register miscellaneous FP16

This section describes the encoding of the Advanced SIMD scalar two-register miscellaneous FP16 instruction class. The encodings in this section are decoded from *Data Processing -- Scalar Floating-Point and Advanced SIMD* on page C4-338.

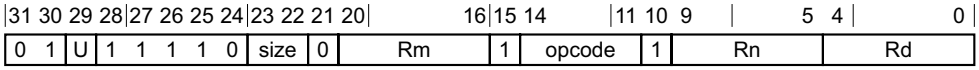


Decode fields			Instruction page	Feature
U	a	opcode		
-	-	00xxx	Unallocated.	-
-	-	010xx	Unallocated.	-
-	-	10xxx	Unallocated.	-
-	-	1100x	Unallocated.	-
-	-	11110	Unallocated.	-
-	0	011xx	Unallocated.	-
-	0	11111	Unallocated.	-
-	1	01111	Unallocated.	-
-	1	11100	Unallocated.	-
0	0	11010	FCVTNS (vector)	FEAT_FP16
0	0	11011	FCVTMS (vector)	FEAT_FP16
0	0	11100	FCVTAS (vector)	FEAT_FP16
0	0	11101	SCVTF (vector, integer)	FEAT_FP16
0	1	01100	FCMGT (zero)	FEAT_FP16
0	1	01101	FCMEQ (zero)	FEAT_FP16
0	1	01110	FCMLT (zero)	FEAT_FP16

Decode fields			Instruction page	Feature
U	a	opcode		
0	1	11010	FCVTPS (vector)	FEAT_FP16
0	1	11011	FCVTZS (vector, integer)	FEAT_FP16
0	1	11101	FRECPE	FEAT_FP16
0	1	11111	FRECPX	FEAT_FP16
1	0	11010	FCVTNU (vector)	FEAT_FP16
1	0	11011	FCVTMU (vector)	FEAT_FP16
1	0	11100	FCVTAU (vector)	FEAT_FP16
1	0	11101	UCVTF (vector, integer)	FEAT_FP16
1	1	01100	FCMGE (zero)	FEAT_FP16
1	1	01101	FCMLE (zero)	FEAT_FP16
1	1	01110	Unallocated.	-
1	1	11010	FCVTPU (vector)	FEAT_FP16
1	1	11011	FCVTZU (vector, integer)	FEAT_FP16
1	1	11101	FRSQRTE	FEAT_FP16
1	1	11111	Unallocated.	-

Advanced SIMD scalar three same extra

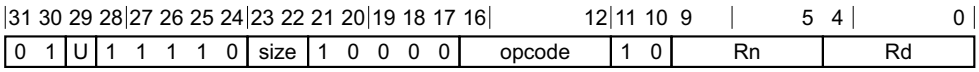
This section describes the encoding of the Advanced SIMD scalar three same extra instruction class. The encodings in this section are decoded from *Data Processing -- Scalar Floating-Point and Advanced SIMD* on page C4-338.



Decode fields		Instruction page	Feature
U	opcode		
-	001x	Unallocated.	-
-	01xx	Unallocated.	-
-	1xxx	Unallocated.	-
0	0000	Unallocated.	-
0	0001	Unallocated.	-
1	0000	SQRDMLAH (vector)	FEAT_RDM
1	0001	SQRDMLSH (vector)	FEAT_RDM

Advanced SIMD scalar two-register miscellaneous

This section describes the encoding of the Advanced SIMD scalar two-register miscellaneous instruction class. The encodings in this section are decoded from *Data Processing -- Scalar Floating-Point and Advanced SIMD* on page C4-338.

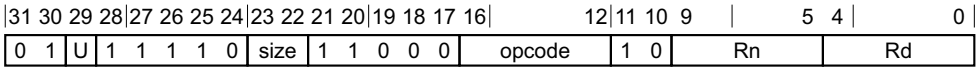


Decode fields			Instruction page
U	size	opcode	
-	-	0000x	Unallocated.
-	-	00010	Unallocated.
-	-	0010x	Unallocated.
-	-	00110	Unallocated.
-	-	01111	Unallocated.
-	-	1000x	Unallocated.
-	-	10011	Unallocated.
-	-	10101	Unallocated.
-	-	10111	Unallocated.
-	-	1100x	Unallocated.
-	-	11110	Unallocated.
-	0x	011xx	Unallocated.
-	0x	11111	Unallocated.
-	1x	10110	Unallocated.
-	1x	11100	Unallocated.
0	-	00011	SUQADD
0	-	00111	SQABS
0	-	01000	CMGT (zero)
0	-	01001	CMEQ (zero)
0	-	01010	CMLT (zero)
0	-	01011	ABS
0	-	10010	Unallocated.
0	-	10100	SQXTN, SQXTN2
0	0x	10110	Unallocated.
0	0x	11010	FCVTNS (vector)

Decode fields			Instruction page
U	size	opcode	
0	0x	11011	FCVTMS (vector)
0	0x	11100	FCVTAS (vector)
0	0x	11101	SCVTF (vector, integer)
0	1x	01100	FCMGT (zero)
0	1x	01101	FCMEQ (zero)
0	1x	01110	FCMLT (zero)
0	1x	11010	FCVTPS (vector)
0	1x	11011	FCVTZS (vector, integer)
0	1x	11101	FRECPE
0	1x	11111	FRECPX
1	-	00011	USQADD
1	-	00111	SQNEG
1	-	01000	CMGE (zero)
1	-	01001	CMLE (zero)
1	-	01010	Unallocated.
1	-	01011	NEG (vector)
1	-	10010	SQXTUN, SQXTUN2
1	-	10100	UQXTN, UQXTN2
1	0x	10110	FCVTXN, FCVTXN2
1	0x	11010	FCVTNU (vector)
1	0x	11011	FCVTMU (vector)
1	0x	11100	FCVTAU (vector)
1	0x	11101	UCVTF (vector, integer)
1	1x	01100	FCMGE (zero)
1	1x	01101	FCMLE (zero)
1	1x	01110	Unallocated.
1	1x	11010	FCVTPU (vector)
1	1x	11011	FCVTZU (vector, integer)
1	1x	11101	FRSQRTE
1	1x	11111	Unallocated.

Advanced SIMD scalar pairwise

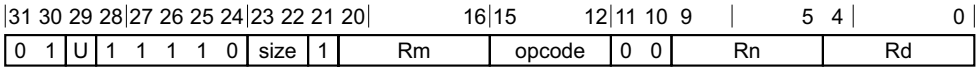
This section describes the encoding of the Advanced SIMD scalar pairwise instruction class. The encodings in this section are decoded from *Data Processing -- Scalar Floating-Point and Advanced SIMD* on page C4-338.



Decode fields			Instruction page	Feature
U	size	opcode		
-	-	00xxx	Unallocated.	-
-	-	010xx	Unallocated.	-
-	-	01110	Unallocated.	-
-	-	10xxx	Unallocated.	-
-	-	1100x	Unallocated.	-
-	-	11010	Unallocated.	-
-	-	111xx	Unallocated.	-
-	1x	01101	Unallocated.	-
0	-	11011	ADDP (scalar)	-
0	0x	01100	FMAXNMP (scalar) - Encoding	FEAT_FP16
0	0x	01101	FADDP (scalar) - Encoding	FEAT_FP16
0	0x	01111	FMAXP (scalar) - Encoding	FEAT_FP16
0	1x	01100	FMINNMP (scalar) - Encoding	FEAT_FP16
0	1x	01111	FMINP (scalar) - Encoding	FEAT_FP16
1	-	11011	Unallocated.	-
1	0x	01100	FMAXNMP (scalar) - Encoding	-
1	0x	01101	FADDP (scalar) - Encoding	-
1	0x	01111	FMAXP (scalar) - Encoding	-
1	1x	01100	FMINNMP (scalar) - Encoding	-
1	1x	01111	FMINP (scalar) - Encoding	-

Advanced SIMD scalar three different

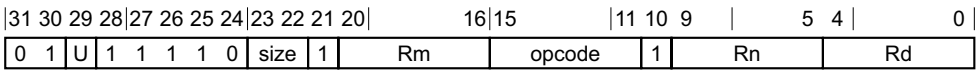
This section describes the encoding of the Advanced SIMD scalar three different instruction class. The encodings in this section are decoded from *Data Processing -- Scalar Floating-Point and Advanced SIMD* on page C4-338.



Decode fields		Instruction page
U	opcode	
-	00xx	Unallocated.
-	01xx	Unallocated.
-	1000	Unallocated.
-	1010	Unallocated.
-	1100	Unallocated.
-	111x	Unallocated.
0	1001	SQDMLAL, SQDMLAL2 (vector)
0	1011	SQDMLSL, SQDMLSL2 (vector)
0	1101	SQDMULL, SQDMULL2 (vector)
1	1001	Unallocated.
1	1011	Unallocated.
1	1101	Unallocated.

Advanced SIMD scalar three same

This section describes the encoding of the Advanced SIMD scalar three same instruction class. The encodings in this section are decoded from *Data Processing -- Scalar Floating-Point and Advanced SIMD* on page C4-338.



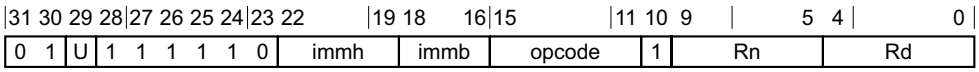
Decode fields			Instruction page
U	size	opcode	
-	-	00000	Unallocated.
-	-	0001x	Unallocated.
-	-	00100	Unallocated.
-	-	011xx	Unallocated.
-	-	1001x	Unallocated.

Decode fields			Instruction page
U	size	opcode	
-	1x	11011	Unallocated.
0	-	00001	SQADD
0	-	00101	SQSUB
0	-	00110	CMGT (register)
0	-	00111	CMGE (register)
0	-	01000	SSHL
0	-	01001	SQSHL (register)
0	-	01010	SRSHL
0	-	01011	SQRSHL
0	-	10000	ADD (vector)
0	-	10001	CMTST
0	-	10100	Unallocated.
0	-	10101	Unallocated.
0	-	10110	SQDMULH (vector)
0	-	10111	Unallocated.
0	0x	11000	Unallocated.
0	0x	11001	Unallocated.
0	0x	11010	Unallocated.
0	0x	11011	FMULX
0	0x	11100	FCMEQ (register)
0	0x	11101	Unallocated.
0	0x	11110	Unallocated.
0	0x	11111	FRECPS
0	1x	11000	Unallocated.
0	1x	11001	Unallocated.
0	1x	11010	Unallocated.
0	1x	11100	Unallocated.
0	1x	11101	Unallocated.
0	1x	11110	Unallocated.
0	1x	11111	FRSQRTS
1	-	00001	UQADD
1	-	00101	UQSUB

Decode fields			Instruction page
U	size	opcode	
1	-	00110	CMHI (register)
1	-	00111	CMHS (register)
1	-	01000	USHL
1	-	01001	UQSHL (register)
1	-	01010	URSHL
1	-	01011	UQRSHL
1	-	10000	SUB (vector)
1	-	10001	CMEQ (register)
1	-	10100	Unallocated.
1	-	10101	Unallocated.
1	-	10110	SQRDMULH (vector)
1	-	10111	Unallocated.
1	0x	11000	Unallocated.
1	0x	11001	Unallocated.
1	0x	11010	Unallocated.
1	0x	11011	Unallocated.
1	0x	11100	FCMGE (register)
1	0x	11101	FACGE
1	0x	11110	Unallocated.
1	0x	11111	Unallocated.
1	1x	11000	Unallocated.
1	1x	11001	Unallocated.
1	1x	11010	FABD
1	1x	11100	FCMGT (register)
1	1x	11101	FACGT
1	1x	11110	Unallocated.
1	1x	11111	Unallocated.

Advanced SIMD scalar shift by immediate

This section describes the encoding of the Advanced SIMD scalar shift by immediate instruction class. The encodings in this section are decoded from *Data Processing -- Scalar Floating-Point and Advanced SIMD* on page C4-338.

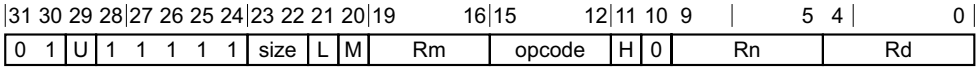


Decode fields			Instruction page
U	immh	opcode	
-	!= 0000	00001	Unallocated.
-	!= 0000	00011	Unallocated.
-	!= 0000	00101	Unallocated.
-	!= 0000	00111	Unallocated.
-	!= 0000	01001	Unallocated.
-	!= 0000	01011	Unallocated.
-	!= 0000	01101	Unallocated.
-	!= 0000	01111	Unallocated.
-	!= 0000	101xx	Unallocated.
-	!= 0000	110xx	Unallocated.
-	!= 0000	11101	Unallocated.
-	!= 0000	11110	Unallocated.
-	0000	-	Unallocated.
0	!= 0000	00000	SSHR
0	!= 0000	00010	SSRA
0	!= 0000	00100	SRSHR
0	!= 0000	00110	SRSRA
0	!= 0000	01000	Unallocated.
0	!= 0000	01010	SHL
0	!= 0000	01100	Unallocated.
0	!= 0000	01110	SQSHL (immediate)
0	!= 0000	10000	Unallocated.
0	!= 0000	10001	Unallocated.
0	!= 0000	10010	SQSHRN, SQSHRN2
0	!= 0000	10011	SQRSHRN, SQRSHRN2

Decode fields			Instruction page
U	immh	opcode	
0	!= 0000	11100	SCVTF (vector, fixed-point)
0	!= 0000	11111	FCVTZS (vector, fixed-point)
1	!= 0000	00000	USHR
1	!= 0000	00010	USRA
1	!= 0000	00100	URSHR
1	!= 0000	00110	URSRA
1	!= 0000	01000	SRI
1	!= 0000	01010	SLI
1	!= 0000	01100	SQSHLU
1	!= 0000	01110	UQSHL (immediate)
1	!= 0000	10000	SQSHRUN, SQSHRUN2
1	!= 0000	10001	SQRSHRUN, SQRSHRUN2
1	!= 0000	10010	UQSHRN, UQSHRN2
1	!= 0000	10011	UQRSHRN, UQRSHRN2
1	!= 0000	11100	UCVTF (vector, fixed-point)
1	!= 0000	11111	FCVTZU (vector, fixed-point)

Advanced SIMD scalar x indexed element

This section describes the encoding of the Advanced SIMD scalar x indexed element instruction class. The encodings in this section are decoded from *Data Processing -- Scalar Floating-Point and Advanced SIMD* on page C4-338.

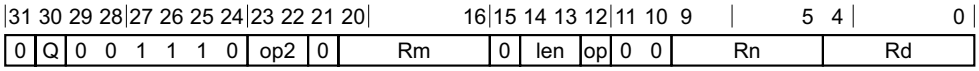


Decode fields			Instruction page	Feature
U	size	opcode		
-	-	0000	Unallocated.	-
-	-	0010	Unallocated.	-
-	-	0100	Unallocated.	-
-	-	0110	Unallocated.	-
-	-	1000	Unallocated.	-
-	-	1010	Unallocated.	-

Decode fields			Instruction page	Feature
U	size	opcode		
-	-	1110	Unallocated.	-
-	01	0001	Unallocated.	-
-	01	0101	Unallocated.	-
-	01	1001	Unallocated.	-
0	-	0011	SQDMLAL, SQDMLAL2 (by element)	-
0	-	0111	SQDMLSL, SQDMLSL2 (by element)	-
0	-	1011	SQDMULL, SQDMULL2 (by element)	-
0	-	1100	SQDMULH (by element)	-
0	-	1101	SQRDMULH (by element)	-
0	-	1111	Unallocated.	-
0	00	0001	FMLA (by element) - Encoding	FEAT_FP16
0	00	0101	FMLS (by element) - Encoding	FEAT_FP16
0	00	1001	FMUL (by element) - Encoding	FEAT_FP16
0	1x	0001	FMLA (by element) - Encoding	-
0	1x	0101	FMLS (by element) - Encoding	-
0	1x	1001	FMUL (by element) - Encoding	-
1	-	0011	Unallocated.	-
1	-	0111	Unallocated.	-
1	-	1011	Unallocated.	-
1	-	1100	Unallocated.	-
1	-	1101	SQRDMLAH (by element)	FEAT_RDM
1	-	1111	SQRDMLSH (by element)	FEAT_RDM
1	00	0001	Unallocated.	-
1	00	0101	Unallocated.	-
1	00	1001	FMULX (by element) - Encoding	FEAT_FP16
1	1x	0001	Unallocated.	-
1	1x	0101	Unallocated.	-
1	1x	1001	FMULX (by element) - Encoding	-

Advanced SIMD table lookup

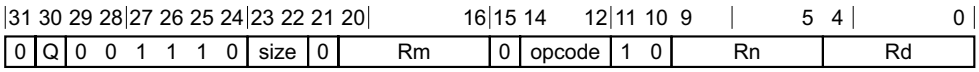
This section describes the encoding of the Advanced SIMD table lookup instruction class. The encodings in this section are decoded from *Data Processing -- Scalar Floating-Point and Advanced SIMD* on page C4-338.



Decode fields			Instruction page
op2	len	op	
x1	-	-	Unallocated.
00	00	0	TBL - Single register table variant
00	00	1	TBX - Single register table variant
00	01	0	TBL - Two register table variant
00	01	1	TBX - Two register table variant
00	10	0	TBL - Three register table variant
00	10	1	TBX - Three register table variant
00	11	0	TBL - Four register table variant
00	11	1	TBX - Four register table variant
1x	-	-	Unallocated.

Advanced SIMD permute

This section describes the encoding of the Advanced SIMD permute instruction class. The encodings in this section are decoded from *Data Processing -- Scalar Floating-Point and Advanced SIMD* on page C4-338.

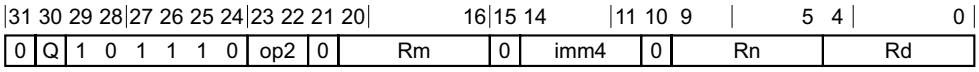


Decode fields	Instruction page
opcode	
000	Unallocated.
001	UZP1
010	TRN1
011	ZIP1
100	Unallocated.

Decode fields	Instruction page
opcode	
101	UZP2
110	TRN2
111	ZIP2

Advanced SIMD extract

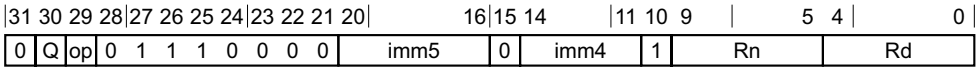
This section describes the encoding of the Advanced SIMD extract instruction class. The encodings in this section are decoded from *Data Processing -- Scalar Floating-Point and Advanced SIMD* on page C4-338.



Decode fields	Instruction page
op2	
x1	Unallocated.
00	EXT
1x	Unallocated.

Advanced SIMD copy

This section describes the encoding of the Advanced SIMD copy instruction class. The encodings in this section are decoded from *Data Processing -- Scalar Floating-Point and Advanced SIMD* on page C4-338.

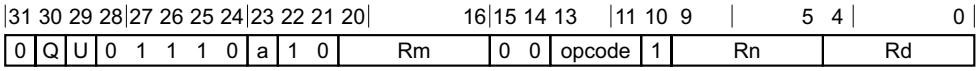


Decode fields				Instruction page
Q	op	imm5	imm4	
-	-	x0000	-	Unallocated.
-	0	-	0000	DUP (element)
-	0	-	0001	DUP (general)
-	0	-	0010	Unallocated.
-	0	-	0100	Unallocated.
-	0	-	0110	Unallocated.
-	0	-	1xxx	Unallocated.
0	0	-	0011	Unallocated.

Decode fields				Instruction page
Q	op	imm5	imm4	
0	0	-	0101	SMOV
0	0	-	0111	UMOV
0	1	-	-	Unallocated.
1	0	-	0011	INS (general)
1	0	-	0101	SMOV
1	0	x1000	0111	UMOV
1	1	-	-	INS (element)

Advanced SIMD three same (FP16)

This section describes the encoding of the Advanced SIMD three same (FP16) instruction class. The encodings in this section are decoded from *Data Processing -- Scalar Floating-Point and Advanced SIMD* on page C4-338.

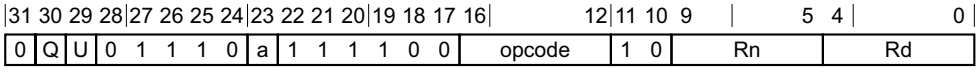


Decode fields			Instruction page	Feature
U	a	opcode		
0	0	000	FMAXNM (vector)	FEAT_FP16
0	0	001	FMLA (vector)	FEAT_FP16
0	0	010	FADD (vector)	FEAT_FP16
0	0	011	FMULX	FEAT_FP16
0	0	100	FCMEQ (register)	FEAT_FP16
0	0	101	Unallocated.	-
0	0	110	FMAX (vector)	FEAT_FP16
0	0	111	FRECPS	FEAT_FP16
0	1	000	FMINNM (vector)	FEAT_FP16
0	1	001	FMLS (vector)	FEAT_FP16
0	1	010	FSUB (vector)	FEAT_FP16
0	1	011	Unallocated.	-
0	1	100	Unallocated.	-
0	1	101	Unallocated.	-
0	1	110	FMIN (vector)	FEAT_FP16
0	1	111	FRSQRTS	FEAT_FP16

Decode fields			Instruction page	Feature
U	a	opcode		
1	0	000	FMAXNMP (vector)	FEAT_FP16
1	0	001	Unallocated.	-
1	0	010	FADDP (vector)	FEAT_FP16
1	0	011	FMUL (vector)	FEAT_FP16
1	0	100	FCMGE (register)	FEAT_FP16
1	0	101	FACGE	FEAT_FP16
1	0	110	FMAXP (vector)	FEAT_FP16
1	0	111	FDIV (vector)	FEAT_FP16
1	1	000	FMINNMP (vector)	FEAT_FP16
1	1	001	Unallocated.	-
1	1	010	FABD	FEAT_FP16
1	1	011	Unallocated.	-
1	1	100	FCMGT (register)	FEAT_FP16
1	1	101	FACGT	FEAT_FP16
1	1	110	FMINP (vector)	FEAT_FP16
1	1	111	Unallocated.	-

Advanced SIMD two-register miscellaneous (FP16)

This section describes the encoding of the Advanced SIMD two-register miscellaneous (FP16) instruction class. The encodings in this section are decoded from *Data Processing -- Scalar Floating-Point and Advanced SIMD* on page C4-338.



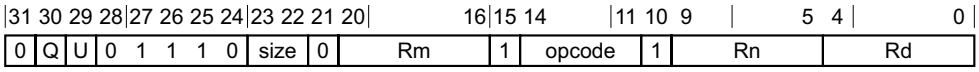
Decode fields			Instruction page	Feature
U	a	opcode		
-	-	00xxx	Unallocated.	-
-	-	010xx	Unallocated.	-
-	-	10xxx	Unallocated.	-
-	-	11110	Unallocated.	-
-	0	011xx	Unallocated.	-
-	0	11111	Unallocated.	-

Decode fields			Instruction page	Feature
U	a	opcode		
-	1	11100	Unallocated.	-
0	0	11000	FRINTN (vector)	FEAT_FP16
0	0	11001	FRINTM (vector)	FEAT_FP16
0	0	11010	FCVTNS (vector)	FEAT_FP16
0	0	11011	FCVTMS (vector)	FEAT_FP16
0	0	11100	FCVTAS (vector)	FEAT_FP16
0	0	11101	SCVTF (vector, integer)	FEAT_FP16
0	1	01100	FCMGT (zero)	FEAT_FP16
0	1	01101	FCMEQ (zero)	FEAT_FP16
0	1	01110	FCMLT (zero)	FEAT_FP16
0	1	01111	FABS (vector)	FEAT_FP16
0	1	11000	FRINTP (vector)	FEAT_FP16
0	1	11001	FRINTZ (vector)	FEAT_FP16
0	1	11010	FCVTPS (vector)	FEAT_FP16
0	1	11011	FCVTZS (vector, integer)	FEAT_FP16
0	1	11101	FRECPE	FEAT_FP16
0	1	11111	Unallocated.	-
1	0	11000	FRINTA (vector)	FEAT_FP16
1	0	11001	FRINTX (vector)	FEAT_FP16
1	0	11010	FCVTNU (vector)	FEAT_FP16
1	0	11011	FCVTMU (vector)	FEAT_FP16
1	0	11100	FCVTAU (vector)	FEAT_FP16
1	0	11101	UCVTF (vector, integer)	FEAT_FP16
1	1	01100	FCMGE (zero)	FEAT_FP16
1	1	01101	FCMLE (zero)	FEAT_FP16
1	1	01110	Unallocated.	-
1	1	01111	FNEG (vector)	FEAT_FP16
1	1	11000	Unallocated.	-
1	1	11001	FRINTI (vector)	FEAT_FP16
1	1	11010	FCVTPU (vector)	FEAT_FP16

Decode fields			Instruction page	Feature
U	a	opcode		
1	1	11011	FCVTZU (vector, integer)	FEAT_FP16
1	1	11101	FRSQRTE	FEAT_FP16
1	1	11111	FSQRT (vector)	FEAT_FP16

Advanced SIMD three-register extension

This section describes the encoding of the Advanced SIMD three-register extension instruction class. The encodings in this section are decoded from *Data Processing -- Scalar Floating-Point and Advanced SIMD* on page C4-338.

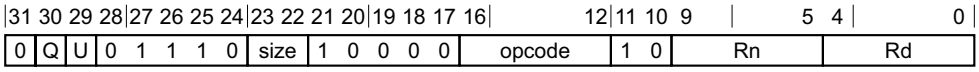


Decode fields				Instruction page	Feature
Q	U	size	opcode		
-	-	0x	0011	Unallocated.	-
-	-	11	0011	Unallocated.	-
-	0	-	0000	Unallocated.	-
-	0	-	0001	Unallocated.	-
-	0	-	0010	SDOT (vector)	FEAT_DotProd
-	0	-	1xxx	Unallocated.	-
-	0	10	0011	USDOT (vector)	FEAT_I8MM
-	1	-	0000	SQRDMLAH (vector)	FEAT_RDM
-	1	-	0001	SQRDMLSH (vector)	FEAT_RDM
-	1	-	0010	UDOT (vector)	FEAT_DotProd
-	1	-	10xx	FCMLA	FEAT_FCMA
-	1	-	11x0	FCADD	FEAT_FCMA
-	1	00	1101	Unallocated.	-
-	1	00	1111	Unallocated.	-
-	1	01	1111	BFDOT (vector)	FEAT_BF16
-	1	1x	1101	Unallocated.	-
-	1	10	0011	Unallocated.	-
-	1	10	1111	Unallocated.	-
-	1	11	1111	BFMLALB, BFMLALT (vector)	FEAT_BF16
0	-	-	01xx	Unallocated.	-

Decode fields				Instruction page	Feature
Q	U	size	opcode		
0	1	01	1101	Unallocated.	-
1	-	0x	01xx	Unallocated.	-
1	-	1x	011x	Unallocated.	-
1	0	10	0100	SMMLA (vector)	FEAT_I8MM
1	0	10	0101	USMMLA (vector)	FEAT_I8MM
1	1	01	1101	BFMMLA	FEAT_BF16
1	1	10	0100	UMMLA (vector)	FEAT_I8MM
1	1	10	0101	Unallocated.	-

Advanced SIMD two-register miscellaneous

This section describes the encoding of the Advanced SIMD two-register miscellaneous instruction class. The encodings in this section are decoded from *Data Processing -- Scalar Floating-Point and Advanced SIMD* on page C4-338.



Decode fields			Instruction page	Feature
U	size	opcode		
-	-	1000x	Unallocated.	-
-	-	10101	Unallocated.	-
-	0x	011xx	Unallocated.	-
-	1x	10111	Unallocated.	-
-	1x	11110	Unallocated.	-
-	11	10110	Unallocated.	-
0	-	00000	REV64	-
0	-	00001	REV16 (vector)	-
0	-	00010	SADDLP	-
0	-	00011	SUQADD	-
0	-	00100	CLS (vector)	-
0	-	00101	CNT	-
0	-	00110	SADALP	-
0	-	00111	SQABS	-

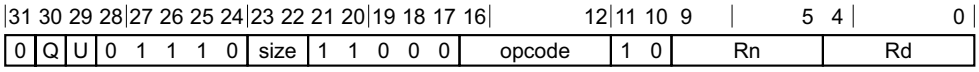
Decode fields			Instruction page	Feature
U	size	opcode		
0	-	01000	CMGT (zero)	-
0	-	01001	CMEQ (zero)	-
0	-	01010	CMLT (zero)	-
0	-	01011	ABS	-
0	-	10010	XTN, XTN2	-
0	-	10011	Unallocated.	-
0	-	10100	SQXTN, SQXTN2	-
0	0x	10110	FCVTN, FCVTN2	-
0	0x	10111	FCVTL, FCVTL2	-
0	0x	11000	FRINTN (vector)	-
0	0x	11001	FRINTM (vector)	-
0	0x	11010	FCVTNS (vector)	-
0	0x	11011	FCVTMS (vector)	-
0	0x	11100	FCVTAS (vector)	-
0	0x	11101	SCVTF (vector, integer)	-
0	0x	11110	FRINT32Z (vector)	FEAT_FRINTTS
0	0x	11111	FRINT64Z (vector)	FEAT_FRINTTS
0	1x	01100	FCMGT (zero)	-
0	1x	01101	FCMEQ (zero)	-
0	1x	01110	FCMLT (zero)	-
0	1x	01111	FABS (vector)	-
0	1x	11000	FRINTP (vector)	-
0	1x	11001	FRINTZ (vector)	-
0	1x	11010	FCVTPS (vector)	-
0	1x	11011	FCVTZS (vector, integer)	-
0	1x	11100	URECPE	-
0	1x	11101	FRECPE	-
0	1x	11111	Unallocated.	-
0	10	10110	BFCVTN, BFCVTN2	FEAT_BF16
1	-	00000	REV32 (vector)	-
1	-	00001	Unallocated.	-
1	-	00010	UADDLP	-

Decode fields			Instruction page	Feature
U	size	opcode		
1	-	00011	USQADD	-
1	-	00100	CLZ (vector)	-
1	-	00110	UADALP	-
1	-	00111	SQNEG	-
1	-	01000	CMGE (zero)	-
1	-	01001	CMLE (zero)	-
1	-	01010	Unallocated.	-
1	-	01011	NEG (vector)	-
1	-	10010	SQXTUN, SQXTUN2	-
1	-	10011	SHLL, SHLL2	-
1	-	10100	UQXTN, UQXTN2	-
1	0x	10110	FCVTXN, FCVTXN2	-
1	0x	10111	Unallocated.	-
1	0x	11000	FRINTA (vector)	-
1	0x	11001	FRINTX (vector)	-
1	0x	11010	FCVTNU (vector)	-
1	0x	11011	FCVTMU (vector)	-
1	0x	11100	FCVTAU (vector)	-
1	0x	11101	UCVTF (vector, integer)	-
1	0x	11110	FRINT32X (vector)	FEAT_FRINTTS
1	0x	11111	FRINT64X (vector)	FEAT_FRINTTS
1	00	00101	NOT	-
1	01	00101	RBIT (vector)	-
1	1x	00101	Unallocated.	-
1	1x	01100	FCMGE (zero)	-
1	1x	01101	FCMLE (zero)	-
1	1x	01110	Unallocated.	-
1	1x	01111	FNEG (vector)	-
1	1x	11000	Unallocated.	-
1	1x	11001	FRINTI (vector)	-
1	1x	11010	FCVTPU (vector)	-
1	1x	11011	FCVTZU (vector, integer)	-

Decode fields			Instruction page	Feature
U	size	opcode		
1	1x	11100	URSQRTE	-
1	1x	11101	FRSQRTE	-
1	1x	11111	FSQRT (vector)	-
1	10	10110	Unallocated.	-

Advanced SIMD across lanes

This section describes the encoding of the Advanced SIMD across lanes instruction class. The encodings in this section are decoded from *Data Processing -- Scalar Floating-Point and Advanced SIMD* on page C4-338.

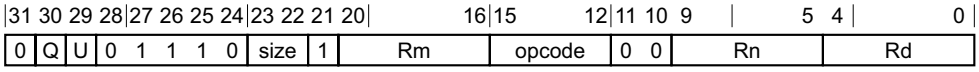


Decode fields			Instruction page	Feature
U	size	opcode		
-	-	0000x	Unallocated.	-
-	-	00010	Unallocated.	-
-	-	001xx	Unallocated.	-
-	-	0100x	Unallocated.	-
-	-	01011	Unallocated.	-
-	-	01101	Unallocated.	-
-	-	01110	Unallocated.	-
-	-	10xxx	Unallocated.	-
-	-	1100x	Unallocated.	-
-	-	111xx	Unallocated.	-
0	-	00011	SADDLV	-
0	-	01010	SMAXV	-
0	-	11010	SMINV	-
0	-	11011	ADDV	-
0	00	01100	FMAXNMV - Encoding	FEAT_FP16
0	00	01111	FMAXV - Encoding	FEAT_FP16
0	01	01100	Unallocated.	-
0	01	01111	Unallocated.	-
0	10	01100	FMINNMV - Encoding	FEAT_FP16

Decode fields			Instruction page	Feature
U	size	opcode		
0	10	01111	FMINV - Encoding	FEAT_FP16
0	11	01100	Unallocated.	-
0	11	01111	Unallocated.	-
1	-	00011	UADDLV	-
1	-	01010	UMAXV	-
1	-	11010	UMINV	-
1	-	11011	Unallocated.	-
1	0x	01100	FMAXNMV - Encoding	-
1	0x	01111	FMAXV - Encoding	-
1	1x	01100	FMINNMV - Encoding	-
1	1x	01111	FMINV - Encoding	-

Advanced SIMD three different

This section describes the encoding of the Advanced SIMD three different instruction class. The encodings in this section are decoded from *Data Processing -- Scalar Floating-Point and Advanced SIMD* on page C4-338.

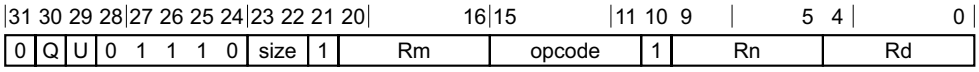


Decode fields		Instruction page
U	opcode	
-	1111	Unallocated.
0	0000	SADDL, SADDL2
0	0001	SADDW, SADDW2
0	0010	SSUBL, SSUBL2
0	0011	SSUBW, SSUBW2
0	0100	ADDHN, ADDHN2
0	0101	SABAL, SABAL2
0	0110	SUBHN, SUBHN2
0	0111	SABDL, SABDL2
0	1000	SMLAL, SMLAL2 (vector)
0	1001	SQDMLAL, SQDMLAL2 (vector)
0	1010	SMLSL, SMLSL2 (vector)

Decode fields		Instruction page
U	opcode	
0	1011	SQDMLSL, SQDMLSL2 (vector)
0	1100	SMULL, SMULL2 (vector)
0	1101	SQDMULL, SQDMULL2 (vector)
0	1110	PMULL, PMULL2
1	0000	UADDL, UADDL2
1	0001	UADDW, UADDW2
1	0010	USUBL, USUBL2
1	0011	USUBW, USUBW2
1	0100	RADDHN, RADDHN2
1	0101	UABAL, UABAL2
1	0110	RSUBHN, RSUBHN2
1	0111	UABDL, UABDL2
1	1000	UMLAL, UMLAL2 (vector)
1	1001	Unallocated.
1	1010	UMLSL, UMLSL2 (vector)
1	1011	Unallocated.
1	1100	UMULL, UMULL2 (vector)
1	1101	Unallocated.
1	1110	Unallocated.

Advanced SIMD three same

This section describes the encoding of the Advanced SIMD three same instruction class. The encodings in this section are decoded from *Data Processing -- Scalar Floating-Point and Advanced SIMD* on page C4-338.



Decode fields			Instruction page	Feature
U	size	opcode		
0	-	00000	SHADD	-
0	-	00001	SQADD	-
0	-	00010	SRHADD	-
0	-	00100	SHSUB	-

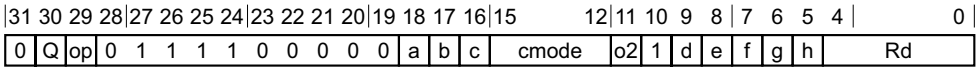
Decode fields			Instruction page	Feature
U	size	opcode		
0	-	00101	SQSUB	-
0	-	00110	CMGT (register)	-
0	-	00111	CMGE (register)	-
0	-	01000	SSHL	-
0	-	01001	SQSHL (register)	-
0	-	01010	SRSHL	-
0	-	01011	SQRSHL	-
0	-	01100	SMAX	-
0	-	01101	SMIN	-
0	-	01110	SABD	-
0	-	01111	SABA	-
0	-	10000	ADD (vector)	-
0	-	10001	CMTST	-
0	-	10010	MLA (vector)	-
0	-	10011	MUL (vector)	-
0	-	10100	SMAXP	-
0	-	10101	SMINP	-
0	-	10110	SQDMULH (vector)	-
0	-	10111	ADDP (vector)	-
0	0x	11000	FMAXNM (vector)	-
0	0x	11001	FMLA (vector)	-
0	0x	11010	FADD (vector)	-
0	0x	11011	FMULX	-
0	0x	11100	FCMEQ (register)	-
0	0x	11110	FMAX (vector)	-
0	0x	11111	FRECPS	-
0	00	00011	AND (vector)	-
0	00	11101	FMLAL, FMLAL2 (vector) - Encoding	FEAT_FHM
0	01	00011	BIC (vector, register)	-
0	01	11101	Unallocated.	-
0	1x	11000	FMINNM (vector)	-
0	1x	11001	FMLS (vector)	-

Decode fields			Instruction page	Feature
U	size	opcode		
0	1x	11010	FSUB (vector)	-
0	1x	11011	Unallocated.	-
0	1x	11100	Unallocated.	-
0	1x	11110	FMIN (vector)	-
0	1x	11111	FRSQRTS	-
0	10	00011	ORR (vector, register)	-
0	10	11101	FMLSL, FMLSL2 (vector) - Encoding	FEAT_FHM
0	11	00011	ORN (vector)	-
0	11	11101	Unallocated.	-
1	-	00000	UHADD	-
1	-	00001	UQADD	-
1	-	00010	URHADD	-
1	-	00100	UHSUB	-
1	-	00101	UQSUB	-
1	-	00110	CMHI (register)	-
1	-	00111	CMHS (register)	-
1	-	01000	USHL	-
1	-	01001	UQSHL (register)	-
1	-	01010	URSHL	-
1	-	01011	UQRSHL	-
1	-	01100	UMAX	-
1	-	01101	UMIN	-
1	-	01110	UABD	-
1	-	01111	UABA	-
1	-	10000	SUB (vector)	-
1	-	10001	CMEQ (register)	-
1	-	10010	MLS (vector)	-
1	-	10011	PMUL	-
1	-	10100	UMAXP	-
1	-	10101	UMINP	-
1	-	10110	SQRDMULH (vector)	-
1	-	10111	Unallocated.	-

Decode fields			Instruction page	Feature
U	size	opcode		
1	0x	11000	FMAXNMP (vector)	-
1	0x	11010	FADDP (vector)	-
1	0x	11011	FMUL (vector)	-
1	0x	11100	FCMGE (register)	-
1	0x	11101	FACGE	-
1	0x	11110	FMAXP (vector)	-
1	0x	11111	FDIV (vector)	-
1	00	00011	EOR (vector)	-
1	00	11001	FMLAL, FMLAL2 (vector) - Encoding	FEAT_FHM
1	01	00011	BSL	-
1	01	11001	Unallocated.	-
1	1x	11000	FMINNMP (vector)	-
1	1x	11010	FABD	-
1	1x	11011	Unallocated.	-
1	1x	11100	FCMGT (register)	-
1	1x	11101	FACGT	-
1	1x	11110	FMINP (vector)	-
1	1x	11111	Unallocated.	-
1	10	00011	BIT	-
1	10	11001	FMLSL, FMLSL2 (vector) - Encoding	FEAT_FHM
1	11	00011	BIF	-
1	11	11001	Unallocated.	-

Advanced SIMD modified immediate

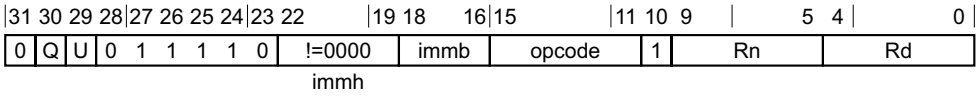
This section describes the encoding of the Advanced SIMD modified immediate instruction class. The encodings in this section are decoded from *Data Processing -- Scalar Floating-Point and Advanced SIMD* on page C4-338.



Decode fields				Instruction page	Feature
Q	op	cmode	o2		
-	0	0xxx	1	Unallocated.	-
-	0	0xx0	0	MOVI - 32-bit shifted immediate variant	-
-	0	0xx1	0	ORR (vector, immediate) - 32-bit variant	-
-	0	10xx	1	Unallocated.	-
-	0	10x0	0	MOVI - 16-bit shifted immediate variant	-
-	0	10x1	0	ORR (vector, immediate) - 16-bit variant	-
-	0	110x	0	MOVI - 32-bit shifting ones variant	-
-	0	110x	1	Unallocated.	-
-	0	1110	0	MOVI - 8-bit variant	-
-	0	1110	1	Unallocated.	-
-	0	1111	0	FMOV (vector, immediate) - Single-precision variant	-
-	0	1111	1	FMOV (vector, immediate) - Encoding	FEAT_FP16
-	1	-	1	Unallocated.	-
-	1	0xx0	0	MVNI - 32-bit shifted immediate variant	-
-	1	0xx1	0	BIC (vector, immediate) - 32-bit variant	-
-	1	10x0	0	MVNI - 16-bit shifted immediate variant	-
-	1	10x1	0	BIC (vector, immediate) - 16-bit variant	-
-	1	110x	0	MVNI - 32-bit shifting ones variant	-
0	1	1110	0	MOVI - 64-bit scalar variant	-
0	1	1111	0	Unallocated.	-
1	1	1110	0	MOVI - 64-bit vector variant	-
1	1	1111	0	FMOV (vector, immediate) - Double-precision variant	-

Advanced SIMD shift by immediate

This section describes the encoding of the Advanced SIMD shift by immediate instruction class. The encodings in this section are decoded from *Data Processing -- Scalar Floating-Point and Advanced SIMD* on page C4-338.

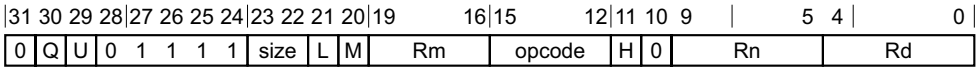


Decode fields		Instruction page
U	opcode	
-	00001	Unallocated.
-	00011	Unallocated.
-	00101	Unallocated.
-	00111	Unallocated.
-	01001	Unallocated.
-	01011	Unallocated.
-	01101	Unallocated.
-	01111	Unallocated.
-	10101	Unallocated.
-	1011x	Unallocated.
-	110xx	Unallocated.
-	11101	Unallocated.
-	11110	Unallocated.
0	00000	SSHR
0	00010	SSRA
0	00100	SRSHR
0	00110	SRSRA
0	01000	Unallocated.
0	01010	SHL
0	01100	Unallocated.
0	01110	SQSHL (immediate)
0	10000	SHRN, SHRN2
0	10001	RSHRN, RSHRN2
0	10010	SQSHRN, SQSHRN2
0	10011	SQRSHRN, SQRSHRN2
0	10100	SSHLL, SSHLL2

Decode fields		Instruction page
U	opcode	
0	11100	SCVTF (vector, fixed-point)
0	11111	FCVTZS (vector, fixed-point)
1	00000	USHR
1	00010	USRA
1	00100	URSHR
1	00110	URSRA
1	01000	SRI
1	01010	SLI
1	01100	SQSHLU
1	01110	UQSHL (immediate)
1	10000	SQSHRUN, SQSHRUN2
1	10001	SQRSHRUN, SQRSHRUN2
1	10010	UQSHRN, UQSHRN2
1	10011	UQRSHRN, UQRSHRN2
1	10100	USHLL, USHLL2
1	11100	UCVTF (vector, fixed-point)
1	11111	FCVTZU (vector, fixed-point)

Advanced SIMD vector x indexed element

This section describes the encoding of the Advanced SIMD vector x indexed element instruction class. The encodings in this section are decoded from *Data Processing -- Scalar Floating-Point and Advanced SIMD* on page C4-338.



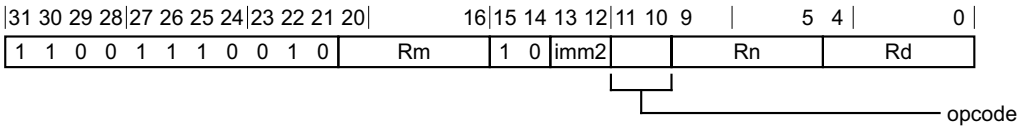
Decode fields			Instruction page	Feature
U	size	opcode		
-	01	1001	Unallocated.	-
0	-	0010	SMLAL, SMLAL2 (by element)	-
0	-	0011	SQDMLAL, SQDMLAL2 (by element)	-
0	-	0110	SMLSL, SMLSL2 (by element)	-
0	-	0111	SQDMLSL, SQDMLSL2 (by element)	-

Decode fields			Instruction page	Feature
U	size	opcode		
0	-	1000	MUL (by element)	-
0	-	1010	SMULL, SMULL2 (by element)	-
0	-	1011	SQDMULL, SQDMULL2 (by element)	-
0	-	1100	SQDMULH (by element)	-
0	-	1101	SQRDMULH (by element)	-
0	-	1110	SDOT (by element)	FEAT_DotProd
0	0x	0000	Unallocated.	-
0	0x	0100	Unallocated.	-
0	00	0001	FMLA (by element) - Encoding	FEAT_FP16
0	00	0101	FMLS (by element) - Encoding	FEAT_FP16
0	00	1001	FMUL (by element) - Encoding	FEAT_FP16
0	00	1111	SUDOT (by element)	FEAT_I8MM
0	01	0001	Unallocated.	-
0	01	0101	Unallocated.	-
0	01	1111	BFDOT (by element)	FEAT_BF16
0	1x	0001	FMLA (by element) - Encoding	-
0	1x	0101	FMLS (by element) - Encoding	-
0	1x	1001	FMUL (by element) - Encoding	-
0	10	0000	FMLAL, FMLAL2 (by element) - Encoding	FEAT_FHM
0	10	0100	FMLSL, FMLSL2 (by element) - Encoding	FEAT_FHM
0	10	1111	USDOT (by element)	FEAT_I8MM
0	11	0000	Unallocated.	-
0	11	0100	Unallocated.	-
0	11	1111	BFMLALB, BFMLALT (by element)	FEAT_BF16
1	-	0000	MLA (by element)	-
1	-	0010	UMLAL, UMLAL2 (by element)	-
1	-	0100	MLS (by element)	-
1	-	0110	UMLSL, UMLSL2 (by element)	-
1	-	1010	UMULL, UMULL2 (by element)	-
1	-	1011	Unallocated.	-
1	-	1101	SQRDMLAH (by element)	FEAT_RDM
1	-	1110	UDOT (by element)	FEAT_DotProd

Decode fields			Instruction page	Feature
U	size	opcode		
1	-	1111	SQRDMLSH (by element)	FEAT_RDM
1	0x	1000	Unallocated.	-
1	0x	1100	Unallocated.	-
1	00	0001	Unallocated.	-
1	00	0011	Unallocated.	-
1	00	0101	Unallocated.	-
1	00	0111	Unallocated.	-
1	00	1001	FMULX (by element) - Encoding	FEAT_FP16
1	01	0xx1	FCMLA (by element)	FEAT_FCMA
1	1x	1001	FMULX (by element) - Encoding	-
1	10	0xx1	FCMLA (by element)	FEAT_FCMA
1	10	1000	FMLAL, FMLAL2 (by element) - Encoding	FEAT_FHM
1	10	1100	FMLSL, FMLSL2 (by element) - Encoding	FEAT_FHM
1	11	0001	Unallocated.	-
1	11	0011	Unallocated.	-
1	11	0101	Unallocated.	-
1	11	0111	Unallocated.	-
1	11	1000	Unallocated.	-
1	11	1100	Unallocated.	-

Cryptographic three-register, imm2

This section describes the encoding of the Cryptographic three-register, imm2 instruction class. The encodings in this section are decoded from *Data Processing -- Scalar Floating-Point and Advanced SIMD* on page C4-338.

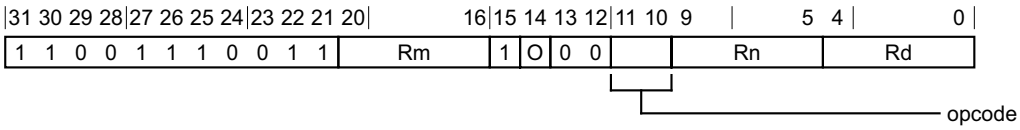


Decode fields	Instruction page	Feature
opcode		
00	SM3TT1A	FEAT_SM3

Decode fields	Instruction page	Feature
opcode		
01	SM3TT1B	FEAT_SM3
10	SM3TT2A	FEAT_SM3
11	SM3TT2B	FEAT_SM3

Cryptographic three-register SHA 512

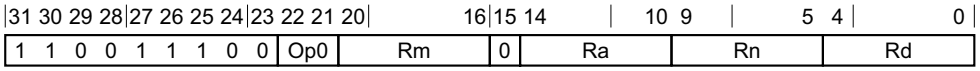
This section describes the encoding of the Cryptographic three-register SHA 512 instruction class. The encodings in this section are decoded from *Data Processing -- Scalar Floating-Point and Advanced SIMD* on page C4-338.



Decode fields		Instruction page	Feature
O	opcode		
0	00	SHA512H	FEAT_SHA512
0	01	SHA512H2	FEAT_SHA512
0	10	SHA512SU1	FEAT_SHA512
0	11	RAX1	FEAT_SHA3
1	00	SM3PARTW1	FEAT_SM3
1	01	SM3PARTW2	FEAT_SM3
1	10	SM4EKEY	FEAT_SM4
1	11	Unallocated.	-

Cryptographic four-register

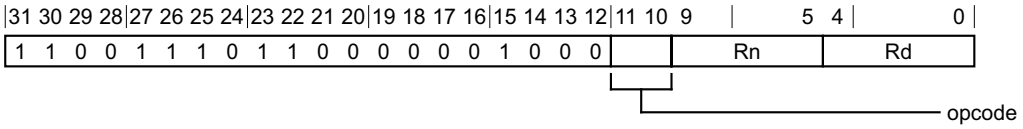
This section describes the encoding of the Cryptographic four-register instruction class. The encodings in this section are decoded from *Data Processing -- Scalar Floating-Point and Advanced SIMD* on page C4-338.



Decode fields	Instruction page	Feature
Op0		
00	EOR3	FEAT_SHA3
01	BCAX	FEAT_SHA3
10	SM3SS1	FEAT_SM3
11	Unallocated.	-

Cryptographic two-register SHA 512

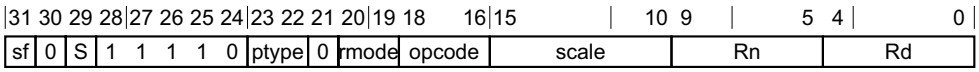
This section describes the encoding of the Cryptographic two-register SHA 512 instruction class. The encodings in this section are decoded from *Data Processing -- Scalar Floating-Point and Advanced SIMD* on page C4-338.



Decode fields	Instruction page	Feature
opcode		
00	SHA512SU0	FEAT_SHA512
01	SM4E	FEAT_SM4
1x	Unallocated.	-

Conversion between floating-point and fixed-point

This section describes the encoding of the Conversion between floating-point and fixed-point instruction class. The encodings in this section are decoded from *Data Processing -- Scalar Floating-Point and Advanced SIMD* on page C4-338.

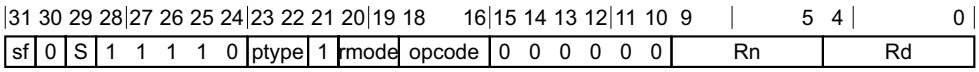


Decode fields						Instruction page	Feature
sf	S	ptype	rmode	opcode	scale		
-	-	-	-	1xx	-	Unallocated.	-
-	-	-	x0	00x	-	Unallocated.	-
-	-	-	x1	01x	-	Unallocated.	-
-	-	-	0x	00x	-	Unallocated.	-
-	-	-	1x	01x	-	Unallocated.	-
-	-	10	-	-	-	Unallocated.	-
-	1	-	-	-	-	Unallocated.	-
0	-	-	-	-	0xxxxx	Unallocated.	-
0	0	00	00	010	-	SCVTF (scalar, fixed-point) - 32-bit to single-precision variant	-
0	0	00	00	011	-	UCVTF (scalar, fixed-point) - 32-bit to single-precision variant	-
0	0	00	11	000	-	FCVTZS (scalar, fixed-point) - Single-precision to 32-bit variant	-
0	0	00	11	001	-	FCVTZU (scalar, fixed-point) - Single-precision to 32-bit variant	-
0	0	01	00	010	-	SCVTF (scalar, fixed-point) - 32-bit to double-precision variant	-
0	0	01	00	011	-	UCVTF (scalar, fixed-point) - 32-bit to double-precision variant	-
0	0	01	11	000	-	FCVTZS (scalar, fixed-point) - Double-precision to 32-bit variant	-
0	0	01	11	001	-	FCVTZU (scalar, fixed-point) - Double-precision to 32-bit variant	-
0	0	11	00	010	-	SCVTF (scalar, fixed-point) - 32-bit to half-precision variant	FEAT_FP16
0	0	11	00	011	-	UCVTF (scalar, fixed-point) - 32-bit to half-precision variant	FEAT_FP16

Decode fields						Instruction page	Feature
sf	S	ptype	rmode	opcode	scale		
0	0	11	11	000	-	FCVTZS (scalar, fixed-point) - Half-precision to 32-bit variant	FEAT_FP16
0	0	11	11	001	-	FCVTZU (scalar, fixed-point) - Half-precision to 32-bit variant	FEAT_FP16
1	0	00	00	010	-	SCVTF (scalar, fixed-point) - 64-bit to single-precision variant	-
1	0	00	00	011	-	UCVTF (scalar, fixed-point) - 64-bit to single-precision variant	-
1	0	00	11	000	-	FCVTZS (scalar, fixed-point) - Single-precision to 64-bit variant	-
1	0	00	11	001	-	FCVTZU (scalar, fixed-point) - Single-precision to 64-bit variant	-
1	0	01	00	010	-	SCVTF (scalar, fixed-point) - 64-bit to double-precision variant	-
1	0	01	00	011	-	UCVTF (scalar, fixed-point) - 64-bit to double-precision variant	-
1	0	01	11	000	-	FCVTZS (scalar, fixed-point) - Double-precision to 64-bit variant	-
1	0	01	11	001	-	FCVTZU (scalar, fixed-point) - Double-precision to 64-bit variant	-
1	0	11	00	010	-	SCVTF (scalar, fixed-point) - 64-bit to half-precision variant	FEAT_FP16
1	0	11	00	011	-	UCVTF (scalar, fixed-point) - 64-bit to half-precision variant	FEAT_FP16
1	0	11	11	000	-	FCVTZS (scalar, fixed-point) - Half-precision to 64-bit variant	FEAT_FP16
1	0	11	11	001	-	FCVTZU (scalar, fixed-point) - Half-precision to 64-bit variant	FEAT_FP16

Conversion between floating-point and integer

This section describes the encoding of the Conversion between floating-point and integer instruction class. The encodings in this section are decoded from *Data Processing -- Scalar Floating-Point and Advanced SIMD* on page C4-338.



Decode fields					Instruction page	Feature
sf	S	ptype	rmode	opcode		
-	-	-	x1	01x	Unallocated.	-
-	-	-	x1	10x	Unallocated.	-
-	-	-	1x	01x	Unallocated.	-
-	-	-	1x	10x	Unallocated.	-
-	0	10	-	0xx	Unallocated.	-
-	0	10	-	10x	Unallocated.	-
-	1	-	-	-	Unallocated.	-
0	0	00	x1	11x	Unallocated.	-
0	0	00	00	000	FCVTNS (scalar) - Single-precision to 32-bit variant	-
0	0	00	00	001	FCVTNU (scalar) - Single-precision to 32-bit variant	-
0	0	00	00	010	SCVTF (scalar, integer) - 32-bit to single-precision variant	-
0	0	00	00	011	UCVTF (scalar, integer) - 32-bit to single-precision variant	-
0	0	00	00	100	FCVTAS (scalar) - Single-precision to 32-bit variant	-
0	0	00	00	101	FCVTAU (scalar) - Single-precision to 32-bit variant	-
0	0	00	00	110	FMOV (general) - Single-precision to 32-bit variant	-
0	0	00	00	111	FMOV (general) - 32-bit to single-precision variant	-
0	0	00	01	000	FCVTPS (scalar) - Single-precision to 32-bit variant	-
0	0	00	01	001	FCVTPU (scalar) - Single-precision to 32-bit variant	-
0	0	00	1x	11x	Unallocated.	-
0	0	00	10	000	FCVTMS (scalar) - Single-precision to 32-bit variant	-
0	0	00	10	001	FCVTMU (scalar) - Single-precision to 32-bit variant	-
0	0	00	11	000	FCVTZS (scalar, integer) - Single-precision to 32-bit variant	-
0	0	00	11	001	FCVTZU (scalar, integer) - Single-precision to 32-bit variant	-
0	0	01	0x	11x	Unallocated.	-
0	0	01	00	000	FCVTNS (scalar) - Double-precision to 32-bit variant	-

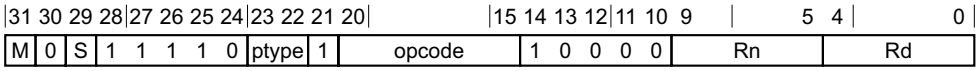
Decode fields					Instruction page	Feature
sf	S	ptype	rmode	opcode		
0	0	01	00	001	FCVTNU (scalar) - Double-precision to 32-bit variant	-
0	0	01	00	010	SCVTF (scalar, integer) - 32-bit to double-precision variant	-
0	0	01	00	011	UCVTF (scalar, integer) - 32-bit to double-precision variant	-
0	0	01	00	100	FCVTAS (scalar) - Double-precision to 32-bit variant	-
0	0	01	00	101	FCVTAU (scalar) - Double-precision to 32-bit variant	-
0	0	01	01	000	FCVTPS (scalar) - Double-precision to 32-bit variant	-
0	0	01	01	001	FCVTPU (scalar) - Double-precision to 32-bit variant	-
0	0	01	10	000	FCVTMS (scalar) - Double-precision to 32-bit variant	-
0	0	01	10	001	FCVTMU (scalar) - Double-precision to 32-bit variant	-
0	0	01	10	11x	Unallocated.	-
0	0	01	11	000	FCVTZS (scalar, integer) - Double-precision to 32-bit variant	-
0	0	01	11	001	FCVTZU (scalar, integer) - Double-precision to 32-bit variant	-
0	0	01	11	110	FJCVTZS	FEAT_JSCVT
0	0	01	11	111	Unallocated.	-
0	0	10	-	11x	Unallocated.	-
0	0	11	00	000	FCVTNS (scalar) - Half-precision to 32-bit variant	FEAT_FP16
0	0	11	00	001	FCVTNU (scalar) - Half-precision to 32-bit variant	FEAT_FP16
0	0	11	00	010	SCVTF (scalar, integer) - 32-bit to half-precision variant	FEAT_FP16
0	0	11	00	011	UCVTF (scalar, integer) - 32-bit to half-precision variant	FEAT_FP16
0	0	11	00	100	FCVTAS (scalar) - Half-precision to 32-bit variant	FEAT_FP16
0	0	11	00	101	FCVTAU (scalar) - Half-precision to 32-bit variant	FEAT_FP16
0	0	11	00	110	FMOV (general) - Half-precision to 32-bit variant	FEAT_FP16
0	0	11	00	111	FMOV (general) - 32-bit to half-precision variant	FEAT_FP16
0	0	11	01	000	FCVTPS (scalar) - Half-precision to 32-bit variant	FEAT_FP16
0	0	11	01	001	FCVTPU (scalar) - Half-precision to 32-bit variant	FEAT_FP16
0	0	11	10	000	FCVTMS (scalar) - Half-precision to 32-bit variant	FEAT_FP16
0	0	11	10	001	FCVTMU (scalar) - Half-precision to 32-bit variant	FEAT_FP16
0	0	11	11	000	FCVTZS (scalar, integer) - Half-precision to 32-bit variant	FEAT_FP16
0	0	11	11	001	FCVTZU (scalar, integer) - Half-precision to 32-bit variant	FEAT_FP16
1	0	00	-	11x	Unallocated.	-
1	0	00	00	000	FCVTNS (scalar) - Single-precision to 64-bit variant	-
1	0	00	00	001	FCVTNU (scalar) - Single-precision to 64-bit variant	-

Decode fields					Instruction page	Feature
sf	S	ptype	rmode	opcode		
1	0	00	00	010	SCVTF (scalar, integer) - 64-bit to single-precision variant	-
1	0	00	00	011	UCVTF (scalar, integer) - 64-bit to single-precision variant	-
1	0	00	00	100	FCVTAS (scalar) - Single-precision to 64-bit variant	-
1	0	00	00	101	FCVTAU (scalar) - Single-precision to 64-bit variant	-
1	0	00	01	000	FCVTPS (scalar) - Single-precision to 64-bit variant	-
1	0	00	01	001	FCVTPU (scalar) - Single-precision to 64-bit variant	-
1	0	00	10	000	FCVTMS (scalar) - Single-precision to 64-bit variant	-
1	0	00	10	001	FCVTMU (scalar) - Single-precision to 64-bit variant	-
1	0	00	11	000	FCVTZS (scalar, integer) - Single-precision to 64-bit variant	-
1	0	00	11	001	FCVTZU (scalar, integer) - Single-precision to 64-bit variant	-
1	0	01	x1	11x	Unallocated.	-
1	0	01	00	000	FCVTNS (scalar) - Double-precision to 64-bit variant	-
1	0	01	00	001	FCVTNU (scalar) - Double-precision to 64-bit variant	-
1	0	01	00	010	SCVTF (scalar, integer) - 64-bit to double-precision variant	-
1	0	01	00	011	UCVTF (scalar, integer) - 64-bit to double-precision variant	-
1	0	01	00	100	FCVTAS (scalar) - Double-precision to 64-bit variant	-
1	0	01	00	101	FCVTAU (scalar) - Double-precision to 64-bit variant	-
1	0	01	00	110	FMOV (general) - Double-precision to 64-bit variant	-
1	0	01	00	111	FMOV (general) - 64-bit to double-precision variant	-
1	0	01	01	000	FCVTPS (scalar) - Double-precision to 64-bit variant	-
1	0	01	01	001	FCVTPU (scalar) - Double-precision to 64-bit variant	-
1	0	01	1x	11x	Unallocated.	-
1	0	01	10	000	FCVTMS (scalar) - Double-precision to 64-bit variant	-
1	0	01	10	001	FCVTMU (scalar) - Double-precision to 64-bit variant	-
1	0	01	11	000	FCVTZS (scalar, integer) - Double-precision to 64-bit variant	-
1	0	01	11	001	FCVTZU (scalar, integer) - Double-precision to 64-bit variant	-
1	0	10	x0	11x	Unallocated.	-
1	0	10	01	110	FMOV (general) - Top half of 128-bit to 64-bit variant	-
1	0	10	01	111	FMOV (general) - 64-bit to top half of 128-bit variant	-
1	0	10	1x	11x	Unallocated.	-
1	0	11	00	000	FCVTNS (scalar) - Half-precision to 64-bit variant	FEAT_FP16
1	0	11	00	001	FCVTNU (scalar) - Half-precision to 64-bit variant	FEAT_FP16

Decode fields					Instruction page	Feature
sf	S	ptype	rmode	opcode		
1	0	11	00	010	SCVTF (scalar, integer) - 64-bit to half-precision variant	FEAT_FP16
1	0	11	00	011	UCVTF (scalar, integer) - 64-bit to half-precision variant	FEAT_FP16
1	0	11	00	100	FCVTAS (scalar) - Half-precision to 64-bit variant	FEAT_FP16
1	0	11	00	101	FCVTAU (scalar) - Half-precision to 64-bit variant	FEAT_FP16
1	0	11	00	110	FMOV (general) - Half-precision to 64-bit variant	FEAT_FP16
1	0	11	00	111	FMOV (general) - 64-bit to half-precision variant	FEAT_FP16
1	0	11	01	000	FCVTPS (scalar) - Half-precision to 64-bit variant	FEAT_FP16
1	0	11	01	001	FCVTPU (scalar) - Half-precision to 64-bit variant	FEAT_FP16
1	0	11	10	000	FCVTMS (scalar) - Half-precision to 64-bit variant	FEAT_FP16
1	0	11	10	001	FCVTMU (scalar) - Half-precision to 64-bit variant	FEAT_FP16
1	0	11	11	000	FCVTZS (scalar, integer) - Half-precision to 64-bit variant	FEAT_FP16
1	0	11	11	001	FCVTZU (scalar, integer) - Half-precision to 64-bit variant	FEAT_FP16

Floating-point data-processing (1 source)

This section describes the encoding of the Floating-point data-processing (1 source) instruction class. The encodings in this section are decoded from *Data Processing -- Scalar Floating-Point and Advanced SIMD* on page C4-338.



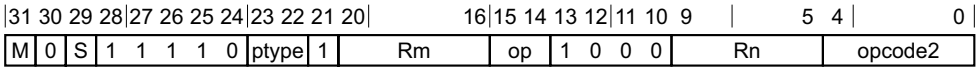
Decode fields				Instruction page	Feature
M	S	ptype	opcode		
-	-	-	1xxxxx	Unallocated.	-
-	1	-	-	Unallocated.	-
0	0	00	000000	FMOV (register) - Single-precision variant	-
0	0	00	000001	FABS (scalar) - Single-precision variant	-
0	0	00	000010	FNEG (scalar) - Single-precision variant	-
0	0	00	000011	FSQRT (scalar) - Single-precision variant	-
0	0	00	000100	Unallocated.	-
0	0	00	000101	FCVT - Single-precision to double-precision variant	-
0	0	00	000110	Unallocated.	-
0	0	00	000111	FCVT - Single-precision to half-precision variant	-

Decode fields				Instruction page	Feature
M	S	ptype	opcode		
0	0	00	001000	FRINTN (scalar) - Single-precision variant	-
0	0	00	001001	FRINTP (scalar) - Single-precision variant	-
0	0	00	001010	FRINTM (scalar) - Single-precision variant	-
0	0	00	001011	FRINTZ (scalar) - Single-precision variant	-
0	0	00	001100	FRINTA (scalar) - Single-precision variant	-
0	0	00	001101	Unallocated.	-
0	0	00	001110	FRINTX (scalar) - Single-precision variant	-
0	0	00	001111	FRINTI (scalar) - Single-precision variant	-
0	0	00	010000	FRINT32Z (scalar) - Single-precision variant	FEAT_FRINTTS
0	0	00	010001	FRINT32X (scalar) - Single-precision variant	FEAT_FRINTTS
0	0	00	010010	FRINT64Z (scalar) - Single-precision variant	FEAT_FRINTTS
0	0	00	010011	FRINT64X (scalar) - Single-precision variant	FEAT_FRINTTS
0	0	00	0101xx	Unallocated.	-
0	0	00	011xxx	Unallocated.	-
0	0	01	000000	FMOV (register) - Double-precision variant	-
0	0	01	000001	FABS (scalar) - Double-precision variant	-
0	0	01	000010	FNEG (scalar) - Double-precision variant	-
0	0	01	000011	FSQRT (scalar) - Double-precision variant	-
0	0	01	000100	FCVT - Double-precision to single-precision variant	-
0	0	01	000101	Unallocated.	-
0	0	01	000110	BFCVT	FEAT_BF16
0	0	01	000111	FCVT - Double-precision to half-precision variant	-
0	0	01	001000	FRINTN (scalar) - Double-precision variant	-
0	0	01	001001	FRINTP (scalar) - Double-precision variant	-
0	0	01	001010	FRINTM (scalar) - Double-precision variant	-
0	0	01	001011	FRINTZ (scalar) - Double-precision variant	-
0	0	01	001100	FRINTA (scalar) - Double-precision variant	-
0	0	01	001101	Unallocated.	-
0	0	01	001110	FRINTX (scalar) - Double-precision variant	-
0	0	01	001111	FRINTI (scalar) - Double-precision variant	-
0	0	01	010000	FRINT32Z (scalar) - Double-precision variant	FEAT_FRINTTS
0	0	01	010001	FRINT32X (scalar) - Double-precision variant	FEAT_FRINTTS

Decode fields				Instruction page	Feature
M	S	ptype	opcode		
0	0	01	010010	FRINT64Z (scalar) - Double-precision variant	FEAT_FRINTTS
0	0	01	010011	FRINT64X (scalar) - Double-precision variant	FEAT_FRINTTS
0	0	01	0101xx	Unallocated.	-
0	0	01	011xxx	Unallocated.	-
0	0	10	0xxxxx	Unallocated.	-
0	0	11	000000	FMOV (register) - Half-precision variant	FEAT_FP16
0	0	11	000001	FABS (scalar) - Half-precision variant	FEAT_FP16
0	0	11	000010	FNEG (scalar) - Half-precision variant	FEAT_FP16
0	0	11	000011	FSQRT (scalar) - Half-precision variant	FEAT_FP16
0	0	11	000100	FCVT - Half-precision to single-precision variant	-
0	0	11	000101	FCVT - Half-precision to double-precision variant	-
0	0	11	00011x	Unallocated.	-
0	0	11	001000	FRINTN (scalar) - Half-precision variant	FEAT_FP16
0	0	11	001001	FRINTP (scalar) - Half-precision variant	FEAT_FP16
0	0	11	001010	FRINTM (scalar) - Half-precision variant	FEAT_FP16
0	0	11	001011	FRINTZ (scalar) - Half-precision variant	FEAT_FP16
0	0	11	001100	FRINTA (scalar) - Half-precision variant	FEAT_FP16
0	0	11	001101	Unallocated.	-
0	0	11	001110	FRINTX (scalar) - Half-precision variant	FEAT_FP16
0	0	11	001111	FRINTI (scalar) - Half-precision variant	FEAT_FP16
0	0	11	01xxxx	Unallocated.	-
1	-	-	-	Unallocated.	-

Floating-point compare

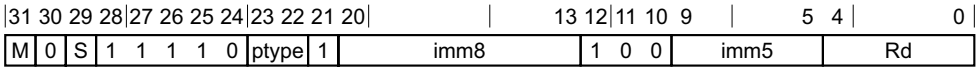
This section describes the encoding of the Floating-point compare instruction class. The encodings in this section are decoded from *Data Processing -- Scalar Floating-Point and Advanced SIMD* on page C4-338.



Decode fields					Instruction page	Feature
M	S	ptype	op	opcode2		
-	-	-	-	xxxx1	Unallocated.	-
-	-	-	-	xxx1x	Unallocated.	-
-	-	-	-	xx1xx	Unallocated.	-
-	-	-	x1	-	Unallocated.	-
-	-	-	1x	-	Unallocated.	-
-	-	10	-	-	Unallocated.	-
-	1	-	-	-	Unallocated.	-
0	0	00	00	00000	FCMP	-
0	0	00	00	01000	FCMP	-
0	0	00	00	10000	FCMPE	-
0	0	00	00	11000	FCMPE	-
0	0	01	00	00000	FCMP	-
0	0	01	00	01000	FCMP	-
0	0	01	00	10000	FCMPE	-
0	0	01	00	11000	FCMPE	-
0	0	11	00	00000	FCMP	FEAT_FP16
0	0	11	00	01000	FCMP	FEAT_FP16
0	0	11	00	10000	FCMPE	FEAT_FP16
0	0	11	00	11000	FCMPE	FEAT_FP16
1	-	-	-	-	Unallocated.	-

Floating-point immediate

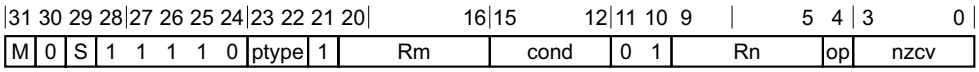
This section describes the encoding of the Floating-point immediate instruction class. The encodings in this section are decoded from *Data Processing -- Scalar Floating-Point and Advanced SIMD* on page C4-338.



Decode fields				Instruction page	Feature
M	S	ptype	imm5		
-	-	-	xxxx1	Unallocated.	-
-	-	-	xxx1x	Unallocated.	-
-	-	-	xx1xx	Unallocated.	-
-	-	-	x1xxx	Unallocated.	-
-	-	-	1xxxx	Unallocated.	-
-	-	10	-	Unallocated.	-
-	1	-	-	Unallocated.	-
0	0	00	00000	FMOV (scalar, immediate) - Single-precision variant	-
0	0	01	00000	FMOV (scalar, immediate) - Double-precision variant	-
0	0	11	00000	FMOV (scalar, immediate) - Half-precision variant	FEAT_FP16
1	-	-	-	Unallocated.	-

Floating-point conditional compare

This section describes the encoding of the Floating-point conditional compare instruction class. The encodings in this section are decoded from *Data Processing -- Scalar Floating-Point and Advanced SIMD* on page C4-338.

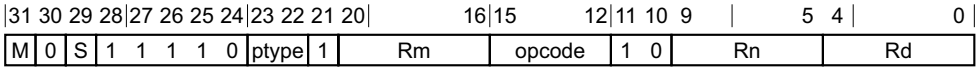


Decode fields				Instruction page	Feature
M	S	ptype	op		
-	-	10	-	Unallocated.	-
-	1	-	-	Unallocated.	-
0	0	00	0	FCCMP - Single-precision variant	-
0	0	00	1	FCCMPE - Single-precision variant	-
0	0	01	0	FCCMP - Double-precision variant	-
0	0	01	1	FCCMPE - Double-precision variant	-

Decode fields				Instruction page	Feature
M	S	ptype	op		
0	0	11	0	FCCMP - Half-precision variant	FEAT_FP16
0	0	11	1	FCCMPE - Half-precision variant	FEAT_FP16
1	-	-	-	Unallocated.	-

Floating-point data-processing (2 source)

This section describes the encoding of the Floating-point data-processing (2 source) instruction class. The encodings in this section are decoded from *Data Processing -- Scalar Floating-Point and Advanced SIMD* on page C4-338.

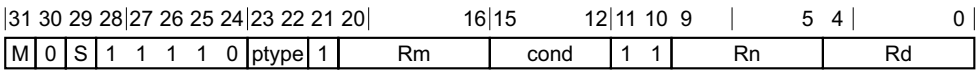


Decode fields				Instruction page	Feature
M	S	ptype	opcode		
-	-	-	1xx1	Unallocated.	-
-	-	-	1x1x	Unallocated.	-
-	-	-	11xx	Unallocated.	-
-	-	10	-	Unallocated.	-
-	1	-	-	Unallocated.	-
0	0	00	0000	FMUL (scalar) - Single-precision variant	-
0	0	00	0001	FDIV (scalar) - Single-precision variant	-
0	0	00	0010	FADD (scalar) - Single-precision variant	-
0	0	00	0011	FSUB (scalar) - Single-precision variant	-
0	0	00	0100	FMAX (scalar) - Single-precision variant	-
0	0	00	0101	FMIN (scalar) - Single-precision variant	-
0	0	00	0110	FMAXNM (scalar) - Single-precision variant	-
0	0	00	0111	FMINNM (scalar) - Single-precision variant	-
0	0	00	1000	FNMUL (scalar) - Single-precision variant	-
0	0	01	0000	FMUL (scalar) - Double-precision variant	-
0	0	01	0001	FDIV (scalar) - Double-precision variant	-
0	0	01	0010	FADD (scalar) - Double-precision variant	-
0	0	01	0011	FSUB (scalar) - Double-precision variant	-
0	0	01	0100	FMAX (scalar) - Double-precision variant	-

Decode fields				Instruction page	Feature
M	S	ptype	opcode		
0	0	01	0101	FMIN (scalar) - Double-precision variant	-
0	0	01	0110	FMAXNM (scalar) - Double-precision variant	-
0	0	01	0111	FMINNM (scalar) - Double-precision variant	-
0	0	01	1000	FNMUL (scalar) - Double-precision variant	-
0	0	11	0000	FMUL (scalar) - Half-precision variant	FEAT_FP16
0	0	11	0001	FDIV (scalar) - Half-precision variant	FEAT_FP16
0	0	11	0010	FADD (scalar) - Half-precision variant	FEAT_FP16
0	0	11	0011	FSUB (scalar) - Half-precision variant	FEAT_FP16
0	0	11	0100	FMAX (scalar) - Half-precision variant	FEAT_FP16
0	0	11	0101	FMIN (scalar) - Half-precision variant	FEAT_FP16
0	0	11	0110	FMAXNM (scalar) - Half-precision variant	FEAT_FP16
0	0	11	0111	FMINNM (scalar) - Half-precision variant	FEAT_FP16
0	0	11	1000	FNMUL (scalar) - Half-precision variant	FEAT_FP16
1	-	-	-	Unallocated.	-

Floating-point conditional select

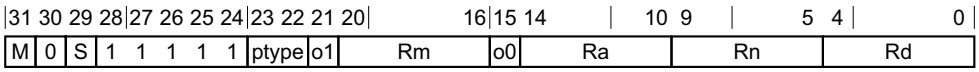
This section describes the encoding of the Floating-point conditional select instruction class. The encodings in this section are decoded from *Data Processing -- Scalar Floating-Point and Advanced SIMD* on page C4-338.



Decode fields			Instruction page	Feature
M	S	ptype		
-	-	10	Unallocated.	-
-	1	-	Unallocated.	-
0	0	00	FCSEL - Single-precision variant	-
0	0	01	FCSEL - Double-precision variant	-
0	0	11	FCSEL - Half-precision variant	FEAT_FP16
1	-	-	Unallocated.	-

Floating-point data-processing (3 source)

This section describes the encoding of the Floating-point data-processing (3 source) instruction class. The encodings in this section are decoded from *Data Processing -- Scalar Floating-Point and Advanced SIMD* on page C4-338.



Decode fields					Instruction page	Feature
M	S	ptype	o1	o0		
-	-	10	-	-	Unallocated.	-
-	1	-	-	-	Unallocated.	-
0	0	00	0	0	FMADD - Single-precision variant	-
0	0	00	0	1	FMSUB - Single-precision variant	-
0	0	00	1	0	FNMADD - Single-precision variant	-
0	0	00	1	1	FNMSUB - Single-precision variant	-
0	0	01	0	0	FMADD - Double-precision variant	-
0	0	01	0	1	FMSUB - Double-precision variant	-
0	0	01	1	0	FNMADD - Double-precision variant	-
0	0	01	1	1	FNMSUB - Double-precision variant	-
0	0	11	0	0	FMADD - Half-precision variant	FEAT_FP16
0	0	11	0	1	FMSUB - Half-precision variant	FEAT_FP16
0	0	11	1	0	FNMADD - Half-precision variant	FEAT_FP16
0	0	11	1	1	FNMSUB - Half-precision variant	FEAT_FP16
1	-	-	-	-	Unallocated.	-