Chapter C3
A64 Instruction Set Encoding

This chapter describes the A64 instruction set encoding. It contains an encoding index followed by a set of functional groups. Each group contains an alphabetical list of instructions that have similar function within the instruction set.

It contains the following sections:

C3.1 A64 instruction index by encoding

Table C3-1 A64 main encoding table

Instruction bits																						Encoding Group
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	
-	-	-	0	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	UNALLOCATED
-	-	-	1	0	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Data processing - immediate
-	-	-	1	0	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Branch, exception generation and system instructions
-	-	-	-	1	-	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Loads and stores
-	-	-	-	1	0	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Data processing - register
-	-	-	0	1	1	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Data processing - SIMD and floating point
-	-	-	1	1	1	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Data processing - SIMD and floating point

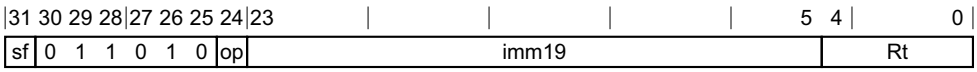
C3.2 Branches, exception generating and system instructions

This section describes the encoding of the instruction classes in the Branch, exception generation and system instruction group, and shows how each instruction class encodes the different instruction forms. For additional information on this functional group of instructions, see *Branches, Exception generating, and System instructions* on page C2-124.

Table C3-2 Encoding table for the Branches, Exception Generating and System instructions functional group

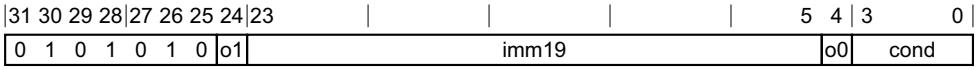
Instruction bits																						Instruction class
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	
-	0	0	1	0	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Unconditional branch (immediate)
-	0	1	1	0	1	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Compare & branch (immediate)
-	0	1	1	0	1	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Test & branch (immediate)
0	1	0	1	0	1	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Conditional branch (immediate)
1	1	0	1	0	1	0	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Exception generation
1	1	0	1	0	1	0	1	0	0	-	-	-	-	-	-	-	-	-	-	-	-	System
1	1	0	1	0	1	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Unconditional branch (register)

C3.2.1 Compare & branch (immediate)



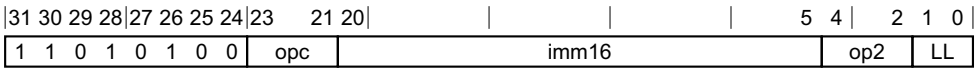
Decode fields		Instruction Page	Variant
sf	op		
0	0	CBZ	32-bit
0	1	CBNZ	32-bit
1	0	CBZ	64-bit
1	1	CBNZ	64-bit

C3.2.2 Conditional branch (immediate)



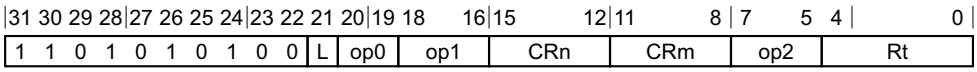
Decode fields		Instruction Page	Variant
o1	o0		
0	0	B.cond	-

C3.2.3 Exception generation



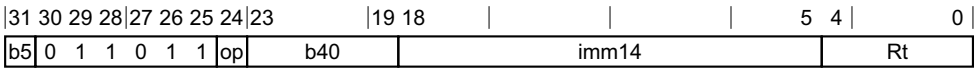
Decode fields			Instruction Page	Variant
opc	op2	LL		
000	000	01	SVC	-
000	000	10	HVC	-
000	000	11	SMC	-
001	000	00	BRK	-
010	000	00	HLT	-
101	000	01	DCPS1	-
101	000	10	DCPS2	-
101	000	11	DCPS3	-

C3.2.4 System



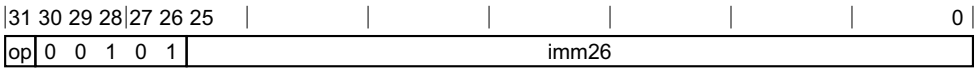
Decode fields						Instruction Page	Variant
L	op0	op1	CRn	op2	Rt		
0	00	-	0100	-	11111	MSR (immediate)	-
0	00	011	0010	-	11111	HINT	-
0	00	011	0011	010	11111	CLREX	-
0	00	011	0011	100	11111	DSB	-
0	00	011	0011	101	11111	DMB	-
0	00	011	0011	110	11111	ISB	-
0	01	-	-	-	-	SYS	-
0	1x	-	-	-	-	MSR (register)	-
1	01	-	-	-	-	SYSL	-
1	1x	-	-	-	-	MRS	-

C3.2.5 Test & branch (immediate)



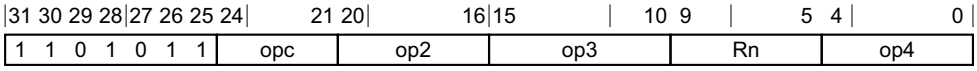
Decode fields		Instruction Page	Variant
op			
0		TBZ	-
1		TBNZ	-

C3.2.6 Unconditional branch (immediate)



Decode fields		Instruction Page	Variant
op			
0		B	-
1		BL	-

C3.2.7 Unconditional branch (register)



Decode fields					Instruction Page	Variant
opc	op2	op3	Rn	op4		
0000	11111	000000	-	00000	BR	-
0001	11111	000000	-	00000	BLR	-
0010	11111	000000	-	00000	RET	-
0100	11111	000000	11111	00000	ERET	-
0101	11111	000000	11111	00000	DRPS	-

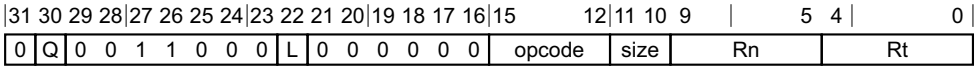
C3.3 Loads and stores

This section describes the encoding of the instruction classes in the Loads and stores instruction group, and shows how each instruction class encodes the different instruction forms. For additional information on this functional group of instructions, see *Loads and stores* on page C2-129.

Table C3-3 Encoding table for the Loads and Stores functional group

Instruction bits																						Instruction class
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	
-	-	0	0	1	0	0	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Load/store exclusive
-	-	0	1	1	-	0	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Load register (literal)
-	-	1	0	1	-	0	0	0	-	-	-	-	-	-	-	-	-	-	-	-	-	Load/store no-allocate pair (offset)
-	-	1	0	1	-	0	0	1	-	-	-	-	-	-	-	-	-	-	-	-	-	Load/store register pair (post-indexed)
-	-	1	0	1	-	0	1	0	-	-	-	-	-	-	-	-	-	-	-	-	-	Load/store register pair (offset)
-	-	1	0	1	-	0	1	1	-	-	-	-	-	-	-	-	-	-	-	-	-	Load/store register pair (pre-indexed)
-	-	1	1	1	-	0	0	-	-	0	-	-	-	-	-	-	-	-	-	0	0	Load/store register (unscaled immediate)
-	-	1	1	1	-	0	0	-	-	0	-	-	-	-	-	-	-	-	-	0	1	Load/store register (immediate post-indexed)
-	-	1	1	1	-	0	0	-	-	0	-	-	-	-	-	-	-	-	-	1	0	Load/store register (unprivileged)
-	-	1	1	1	-	0	0	-	-	0	-	-	-	-	-	-	-	-	-	1	1	Load/store register (immediate pre-indexed)
-	-	1	1	1	-	0	0	-	-	1	-	-	-	-	-	-	-	-	-	1	0	Load/store register (register offset)
-	-	1	1	1	-	0	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Load/store register (unsigned immediate)
0	-	0	0	1	1	0	0	0	-	0	0	0	0	0	0	-	-	-	-	-	-	AdvSIMD load/store multiple structures
0	-	0	0	1	1	0	0	1	-	0	-	-	-	-	-	-	-	-	-	-	-	AdvSIMD load/store multiple structures (post-indexed)
0	-	0	0	1	1	0	1	0	-	-	0	0	0	0	0	-	-	-	-	-	-	AdvSIMD load/store single structure
0	-	0	0	1	1	0	1	1	-	-	-	-	-	-	-	-	-	-	-	-	-	AdvSIMD load/store single structure (post-indexed)

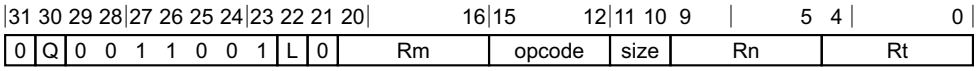
C3.3.1 AdvSIMD load/store multiple structures



Decode fields		Instruction Page	Variant
L	opcode		
0	0000	ST4 (multiple structures)	No offset
0	0010	ST1 (multiple structures)	Four registers
0	0100	ST3 (multiple structures)	No offset
0	0110	ST1 (multiple structures)	Three registers

Decode fields		Instruction Page	Variant
L	opcode		
0	0111	ST1 (multiple structures)	One register
0	1000	ST2 (multiple structures)	No offset
0	1010	ST1 (multiple structures)	Two registers
1	0000	LD4 (multiple structures)	No offset
1	0010	LD1 (multiple structures)	Four registers
1	0100	LD3 (multiple structures)	No offset
1	0110	LD1 (multiple structures)	Three registers
1	0111	LD1 (multiple structures)	One register
1	1000	LD2 (multiple structures)	No offset
1	1010	LD1 (multiple structures)	Two registers

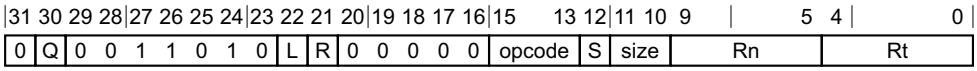
C3.3.2 AdvSIMD load/store multiple structures (post-indexed)



Decode fields			Instruction Page	Variant
L	Rm	opcode		
0	!= 11111	0000	ST4 (multiple structures)	Register offset
0	!= 11111	0010	ST1 (multiple structures)	Four registers, register offset
0	!= 11111	0100	ST3 (multiple structures)	Register offset
0	!= 11111	0110	ST1 (multiple structures)	Three registers, register offset
0	!= 11111	0111	ST1 (multiple structures)	One register, register offset
0	!= 11111	1000	ST2 (multiple structures)	Register offset
0	!= 11111	1010	ST1 (multiple structures)	Two registers, register offset
0	11111	0000	ST4 (multiple structures)	Immediate offset
0	11111	0010	ST1 (multiple structures)	Four registers, immediate offset
0	11111	0100	ST3 (multiple structures)	Immediate offset
0	11111	0110	ST1 (multiple structures)	Three registers, immediate offset
0	11111	0111	ST1 (multiple structures)	One register, immediate offset
0	11111	1000	ST2 (multiple structures)	Immediate offset
0	11111	1010	ST1 (multiple structures)	Two registers, immediate offset

Decode fields			Instruction Page	Variant
L	Rm	opcode		
1	!= 11111	0000	LD4 (multiple structures)	Register offset
1	!= 11111	0010	LD1 (multiple structures)	Four registers, register offset
1	!= 11111	0100	LD3 (multiple structures)	Register offset
1	!= 11111	0110	LD1 (multiple structures)	Three registers, register offset
1	!= 11111	0111	LD1 (multiple structures)	One register, register offset
1	!= 11111	1000	LD2 (multiple structures)	Register offset
1	!= 11111	1010	LD1 (multiple structures)	Two registers, register offset
1	11111	0000	LD4 (multiple structures)	Immediate offset
1	11111	0010	LD1 (multiple structures)	Four registers, immediate offset
1	11111	0100	LD3 (multiple structures)	Immediate offset
1	11111	0110	LD1 (multiple structures)	Three registers, immediate offset
1	11111	0111	LD1 (multiple structures)	One register, immediate offset
1	11111	1000	LD2 (multiple structures)	Immediate offset
1	11111	1010	LD1 (multiple structures)	Two registers, immediate offset

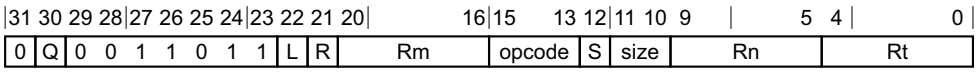
C3.3.3 AdvSIMD load/store single structure



Decode fields					Instruction Page	Variant
L	R	opcode	S	size		
0	0	000	-	-	ST1 (single structure)	8-bit
0	0	001	-	-	ST3 (single structure)	8-bit
0	0	010	-	x0	ST1 (single structure)	16-bit
0	0	011	-	x0	ST3 (single structure)	16-bit
0	0	100	-	00	ST1 (single structure)	32-bit
0	0	100	0	01	ST1 (single structure)	64-bit
0	0	101	-	00	ST3 (single structure)	32-bit
0	0	101	0	01	ST3 (single structure)	64-bit
0	1	000	-	-	ST2 (single structure)	8-bit
0	1	001	-	-	ST4 (single structure)	8-bit

Decode fields					Instruction Page	Variant
L	R	opcode	S	size		
0	1	010	-	x0	ST2 (single structure)	16-bit
0	1	011	-	x0	ST4 (single structure)	16-bit
0	1	100	-	00	ST2 (single structure)	32-bit
0	1	100	0	01	ST2 (single structure)	64-bit
0	1	101	-	00	ST4 (single structure)	32-bit
0	1	101	0	01	ST4 (single structure)	64-bit
1	0	000	-	-	LD1 (single structure)	8-bit
1	0	001	-	-	LD3 (single structure)	8-bit
1	0	010	-	x0	LD1 (single structure)	16-bit
1	0	011	-	x0	LD3 (single structure)	16-bit
1	0	100	-	00	LD1 (single structure)	32-bit
1	0	100	0	01	LD1 (single structure)	64-bit
1	0	101	-	00	LD3 (single structure)	32-bit
1	0	101	0	01	LD3 (single structure)	64-bit
1	0	110	0	-	LD1R	No offset
1	0	111	0	-	LD3R	No offset
1	1	000	-	-	LD2 (single structure)	8-bit
1	1	001	-	-	LD4 (single structure)	8-bit
1	1	010	-	x0	LD2 (single structure)	16-bit
1	1	011	-	x0	LD4 (single structure)	16-bit
1	1	100	-	00	LD2 (single structure)	32-bit
1	1	100	0	01	LD2 (single structure)	64-bit
1	1	101	-	00	LD4 (single structure)	32-bit
1	1	101	0	01	LD4 (single structure)	64-bit
1	1	110	0	-	LD2R	No offset
1	1	111	0	-	LD4R	No offset

C3.3.4 AdvSIMD load/store single structure (post-indexed)

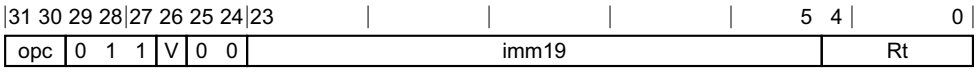


Decode fields						Instruction Page	Variant
L	R	Rm	opcode	S	size		
0	0	!= 11111	000	-	-	ST1 (single structure)	8-bit, register offset
0	0	!= 11111	001	-	-	ST3 (single structure)	8-bit, register offset
0	0	!= 11111	010	-	x0	ST1 (single structure)	16-bit, register offset
0	0	!= 11111	011	-	x0	ST3 (single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

Decode fields						Instruction Page	Variant
L	R	Rm	opcode	S	size		
0	1	11111	011	-	x0	ST4 (single structure)	16-bit, immediate offset
0	1	11111	100	-	00	ST2 (single structure)	32-bit, immediate offset
0	1	11111	100	0	01	ST2 (single structure)	64-bit, immediate offset
0	1	11111	101	-	00	ST4 (single structure)	32-bit, immediate offset
0	1	11111	101	0	01	ST4 (single structure)	64-bit, immediate offset
1	0	!= 11111	000	-	-	LD1 (single structure)	8-bit, register offset
1	0	!= 11111	001	-	-	LD3 (single structure)	8-bit, register offset
1	0	!= 11111	010	-	x0	LD1 (single structure)	16-bit, register offset
1	0	!= 11111	011	-	x0	LD3 (single structure)	16-bit, register offset
1	0	!= 11111	100	-	00	LD1 (single structure)	32-bit, register offset
1	0	!= 11111	100	0	01	LD1 (single structure)	64-bit, register offset
1	0	!= 11111	101	-	00	LD3 (single structure)	32-bit, register offset
1	0	!= 11111	101	0	01	LD3 (single structure)	64-bit, register offset
1	0	!= 11111	110	0	-	LD1R	Register offset
1	0	!= 11111	111	0	-	LD3R	Register offset
1	0	11111	000	-	-	LD1 (single structure)	8-bit, immediate offset
1	0	11111	001	-	-	LD3 (single structure)	8-bit, immediate offset
1	0	11111	010	-	x0	LD1 (single structure)	16-bit, immediate offset
1	0	11111	011	-	x0	LD3 (single structure)	16-bit, immediate offset
1	0	11111	100	-	00	LD1 (single structure)	32-bit, immediate offset
1	0	11111	100	0	01	LD1 (single structure)	64-bit, immediate offset
1	0	11111	101	-	00	LD3 (single structure)	32-bit, immediate offset
1	0	11111	101	0	01	LD3 (single structure)	64-bit, immediate offset
1	0	11111	110	0	-	LD1R	Immediate offset
1	0	11111	111	0	-	LD3R	Immediate offset
1	1	!= 11111	000	-	-	LD2 (single structure)	8-bit, register offset
1	1	!= 11111	001	-	-	LD4 (single structure)	8-bit, register offset
1	1	!= 11111	010	-	x0	LD2 (single structure)	16-bit, register offset
1	1	!= 11111	011	-	x0	LD4 (single structure)	16-bit, register offset
1	1	!= 11111	100	-	00	LD2 (single structure)	32-bit, register offset
1	1	!= 11111	100	0	01	LD2 (single structure)	64-bit, register offset
1	1	!= 11111	101	-	00	LD4 (single structure)	32-bit, register offset

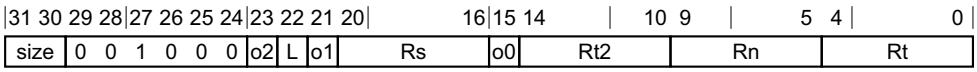
Decode fields						Instruction Page	Variant
L	R	Rm	opcode	S	size		
1	1	!= 11111	101	0	01	LD4 (single structure)	64-bit, register offset
1	1	!= 11111	110	0	-	LD2R	Register offset
1	1	!= 11111	111	0	-	LD4R	Register offset
1	1	11111	000	-	-	LD2 (single structure)	8-bit, immediate offset
1	1	11111	001	-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

C3.3.5 Load register (literal)



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

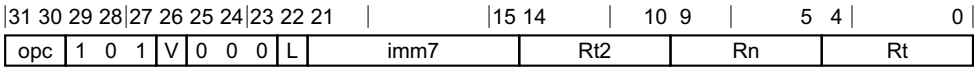
C3.3.6 Load/store exclusive



Decode fields					Instruction Page	Variant
size	o2	L	o1	o0		
00	0	0	0	0	STXRB	-
00	0	0	0	1	STLXRB	-
00	0	1	0	0	LDXRB	-
00	0	1	0	1	LDAXRB	-
00	1	0	0	1	STLRB	-
00	1	1	0	1	LDARB	-
01	0	0	0	0	STXRH	-
01	0	0	0	1	STLXRH	-
01	0	1	0	0	LDXRH	-
01	0	1	0	1	LDAXRH	-
01	1	0	0	1	STLRH	-
01	1	1	0	1	LDARH	-
10	0	0	0	0	STXR	32-bit
10	0	0	0	1	STLXR	32-bit
10	0	0	1	0	STXP	32-bit
10	0	0	1	1	STLXP	32-bit
10	0	1	0	0	LDXR	32-bit
10	0	1	0	1	LDAXR	32-bit
10	0	1	1	0	LDXP	32-bit
10	0	1	1	1	LDAXP	32-bit
10	1	0	0	1	STLR	32-bit
10	1	1	0	1	LDAR	32-bit
11	0	0	0	0	STXR	64-bit
11	0	0	0	1	STLXR	64-bit
11	0	0	1	0	STXP	64-bit
11	0	0	1	1	STLXP	64-bit
11	0	1	0	0	LDXR	64-bit

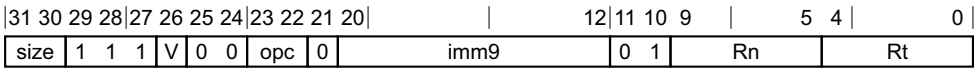
Decode fields					Instruction Page	Variant
size	o2	L	o1	o0		
11	0	1	0	1	LDAXR	64-bit
11	0	1	1	0	LDXP	64-bit
11	0	1	1	1	LDAXP	64-bit
11	1	0	0	1	STLR	64-bit
11	1	1	0	1	LDAR	64-bit

C3.3.7 Load/store no-allocate pair (offset)



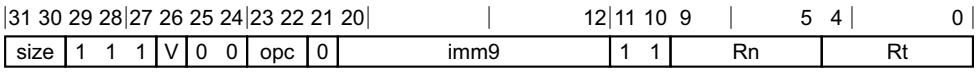
Decode fields			Instruction Page	Variant
opc	V	L		
00	0	0	STNP	32-bit
00	0	1	LDNP	32-bit
00	1	0	STNP (SIMD&FP)	32-bit
00	1	1	LDNP (SIMD&FP)	32-bit
01	1	0	STNP (SIMD&FP)	64-bit
01	1	1	LDNP (SIMD&FP)	64-bit
10	0	0	STNP	64-bit
10	0	1	LDNP	64-bit
10	1	0	STNP (SIMD&FP)	128-bit
10	1	1	LDNP (SIMD&FP)	128-bit

C3.3.8 Load/store register (immediate post-indexed)



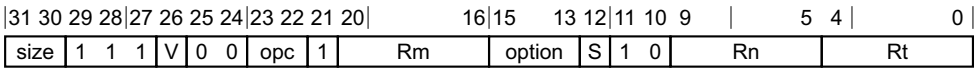
Decode fields			Instruction Page	Variant
size	V	opc		
00	0	00	STRB (immediate)	Post-index
00	0	01	LDRB (immediate)	Post-index
00	0	10	LDRSB (immediate)	64-bit
00	0	11	LDRSB (immediate)	32-bit
00	1	00	STR (immediate, SIMD&FP)	8-bit
00	1	01	LDR (immediate, SIMD&FP)	8-bit
00	1	10	STR (immediate, SIMD&FP)	128-bit
00	1	11	LDR (immediate, SIMD&FP)	128-bit
01	0	00	STRH (immediate)	Post-index
01	0	01	LDRH (immediate)	Post-index
01	0	10	LDRSH (immediate)	64-bit
01	0	11	LDRSH (immediate)	32-bit
01	1	00	STR (immediate, SIMD&FP)	16-bit
01	1	01	LDR (immediate, SIMD&FP)	16-bit
10	0	00	STR (immediate)	32-bit
10	0	01	LDR (immediate)	32-bit
10	0	10	LDRSW (immediate)	Post-index
10	1	00	STR (immediate, SIMD&FP)	32-bit
10	1	01	LDR (immediate, SIMD&FP)	32-bit
11	0	00	STR (immediate)	64-bit
11	0	01	LDR (immediate)	64-bit
11	1	00	STR (immediate, SIMD&FP)	64-bit
11	1	01	LDR (immediate, SIMD&FP)	64-bit

C3.3.9 Load/store register (immediate pre-indexed)



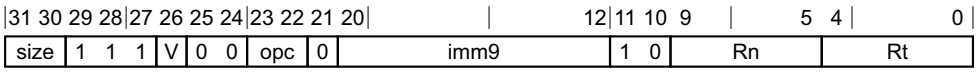
Decode fields			Instruction Page	Variant
size	V	opc		
00	0	00	STRB (immediate)	Pre-index
00	0	01	LDRB (immediate)	Pre-index
00	0	10	LDRSB (immediate)	64-bit
00	0	11	LDRSB (immediate)	32-bit
00	1	00	STR (immediate, SIMD&FP)	8-bit
00	1	01	LDR (immediate, SIMD&FP)	8-bit
00	1	10	STR (immediate, SIMD&FP)	128-bit
00	1	11	LDR (immediate, SIMD&FP)	128-bit
01	0	00	STRH (immediate)	Pre-index
01	0	01	LDRH (immediate)	Pre-index
01	0	10	LDRSH (immediate)	64-bit
01	0	11	LDRSH (immediate)	32-bit
01	1	00	STR (immediate, SIMD&FP)	16-bit
01	1	01	LDR (immediate, SIMD&FP)	16-bit
10	0	00	STR (immediate)	32-bit
10	0	01	LDR (immediate)	32-bit
10	0	10	LDRSW (immediate)	Pre-index
10	1	00	STR (immediate, SIMD&FP)	32-bit
10	1	01	LDR (immediate, SIMD&FP)	32-bit
11	0	00	STR (immediate)	64-bit
11	0	01	LDR (immediate)	64-bit
11	1	00	STR (immediate, SIMD&FP)	64-bit
11	1	01	LDR (immediate, SIMD&FP)	64-bit

C3.3.10 Load/store register (register offset)



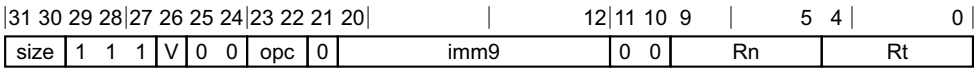
Decode fields				Instruction Page	Variant
size	V	opc	option		
00	0	00	-	STRB (register)	-
00	0	01	-	LDRB (register)	-
00	0	10	-	LDRSB (register)	64-bit
00	0	11	-	LDRSB (register)	32-bit
00	1	00	-	STR (register, SIMD&FP)	8-bit
00	1	01	-	LDR (register, SIMD&FP)	8-bit
00	1	10	-	STR (register, SIMD&FP)	128-bit
00	1	11	-	LDR (register, SIMD&FP)	128-bit
01	0	00	-	STRH (register)	-
01	0	01	-	LDRH (register)	-
01	0	10	-	LDRSH (register)	64-bit
01	0	11	-	LDRSH (register)	32-bit
01	1	00	-	STR (register, SIMD&FP)	16-bit
01	1	01	-	LDR (register, SIMD&FP)	16-bit
10	0	00	-	STR (register)	32-bit
10	0	01	-	LDR (register)	32-bit
10	0	10	-	LDRSW (register)	-
10	1	00	-	STR (register, SIMD&FP)	32-bit
10	1	01	-	LDR (register, SIMD&FP)	32-bit
11	0	00	-	STR (register)	64-bit
11	0	01	-	LDR (register)	64-bit
11	0	10	-	PRFM (register)	-
11	1	00	-	STR (register, SIMD&FP)	64-bit
11	1	01	-	LDR (register, SIMD&FP)	64-bit

C3.3.11 Load/store register (unprivileged)



Decode fields			Instruction Page	Variant
size	V	opc		
00	0	00	STTRB	-
00	0	01	LDTRB	-
00	0	10	LDTRSB	64-bit
00	0	11	LDTRSB	32-bit
01	0	00	STTRH	-
01	0	01	LDTRH	-
01	0	10	LDTRSH	64-bit
01	0	11	LDTRSH	32-bit
10	0	00	STTR	32-bit
10	0	01	LDTR	32-bit
10	0	10	LDTRSW	-
11	0	00	STTR	64-bit
11	0	01	LDTR	64-bit

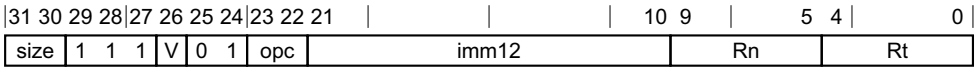
C3.3.12 Load/store register (unscaled immediate)



Decode fields			Instruction Page	Variant
size	V	opc		
00	0	00	STURB	-
00	0	01	LDURB	-
00	0	10	LDURSB	64-bit
00	0	11	LDURSB	32-bit
00	1	00	STUR (SIMD&FP)	8-bit
00	1	01	LDUR (SIMD&FP)	8-bit
00	1	10	STUR (SIMD&FP)	128-bit

Decode fields			Instruction Page	Variant
size	V	opc		
00	1	11	LDUR (SIMD&FP)	128-bit
01	0	00	STURH	-
01	0	01	LDURH	-
01	0	10	LDURSH	64-bit
01	0	11	LDURSH	32-bit
01	1	00	STUR (SIMD&FP)	16-bit
01	1	01	LDUR (SIMD&FP)	16-bit
10	0	00	STUR	32-bit
10	0	01	LDUR	32-bit
10	0	10	LDURSW	-
10	1	00	STUR (SIMD&FP)	32-bit
10	1	01	LDUR (SIMD&FP)	32-bit
11	0	00	STUR	64-bit
11	0	01	LDUR	64-bit
11	0	10	PRFUM	-
11	1	00	STUR (SIMD&FP)	64-bit
11	1	01	LDUR (SIMD&FP)	64-bit

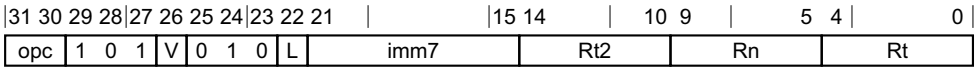
C3.3.13 Load/store register (unsigned immediate)



Decode fields			Instruction Page	Variant
size	V	opc		
00	0	00	STRB (immediate)	Unsigned offset
00	0	01	LDRB (immediate)	Unsigned offset
00	0	10	LDRSB (immediate)	64-bit
00	0	11	LDRSB (immediate)	32-bit
00	1	00	STR (immediate, SIMD&FP)	8-bit
00	1	01	LDR (immediate, SIMD&FP)	8-bit
00	1	10	STR (immediate, SIMD&FP)	128-bit

Decode fields			Instruction Page	Variant
size	V	opc		
00	1	11	LDR (immediate, SIMD&FP)	128-bit
01	0	00	STRH (immediate)	Unsigned offset
01	0	01	LDRH (immediate)	Unsigned offset
01	0	10	LDRSH (immediate)	64-bit
01	0	11	LDRSH (immediate)	32-bit
01	1	00	STR (immediate, SIMD&FP)	16-bit
01	1	01	LDR (immediate, SIMD&FP)	16-bit
10	0	00	STR (immediate)	32-bit
10	0	01	LDR (immediate)	32-bit
10	0	10	LDRSW (immediate)	Unsigned offset
10	1	00	STR (immediate, SIMD&FP)	32-bit
10	1	01	LDR (immediate, SIMD&FP)	32-bit
11	0	00	STR (immediate)	64-bit
11	0	01	LDR (immediate)	64-bit
11	0	10	PRFM (immediate)	-
11	1	00	STR (immediate, SIMD&FP)	64-bit
11	1	01	LDR (immediate, SIMD&FP)	64-bit

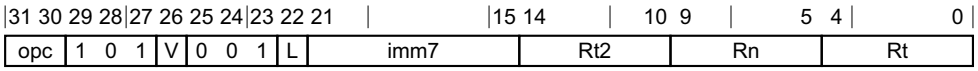
C3.3.14 Load/store register pair (offset)



Decode fields			Instruction Page	Variant
opc	V	L		
00	0	0	STP	32-bit
00	0	1	LDP	32-bit
00	1	0	STP (SIMD&FP)	32-bit
00	1	1	LDP (SIMD&FP)	32-bit
01	0	1	LDPSW	Signed offset
01	1	0	STP (SIMD&FP)	64-bit
01	1	1	LDP (SIMD&FP)	64-bit

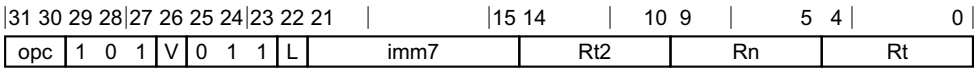
Decode fields			Instruction Page	Variant
opc	V	L		
10	0	0	STP	64-bit
10	0	1	LDP	64-bit
10	1	0	STP (SIMD&FP)	128-bit
10	1	1	LDP (SIMD&FP)	128-bit

C3.3.15 Load/store register pair (post-indexed)



Decode fields			Instruction Page	Variant
opc	V	L		
00	0	0	STP	32-bit
00	0	1	LDP	32-bit
00	1	0	STP (SIMD&FP)	32-bit
00	1	1	LDP (SIMD&FP)	32-bit
01	0	1	LDPSW	Post-index
01	1	0	STP (SIMD&FP)	64-bit
01	1	1	LDP (SIMD&FP)	64-bit
10	0	0	STP	64-bit
10	0	1	LDP	64-bit
10	1	0	STP (SIMD&FP)	128-bit
10	1	1	LDP (SIMD&FP)	128-bit

C3.3.16 Load/store register pair (pre-indexed)



Decode fields			Instruction Page	Variant
opc	V	L		
00	0	0	STP	32-bit
00	0	1	LDP	32-bit
00	1	0	STP (SIMD&FP)	32-bit
00	1	1	LDP (SIMD&FP)	32-bit
01	0	1	LDPSW	Pre-index
01	1	0	STP (SIMD&FP)	64-bit
01	1	1	LDP (SIMD&FP)	64-bit
10	0	0	STP	64-bit
10	0	1	LDP	64-bit
10	1	0	STP (SIMD&FP)	128-bit
10	1	1	LDP (SIMD&FP)	128-bit

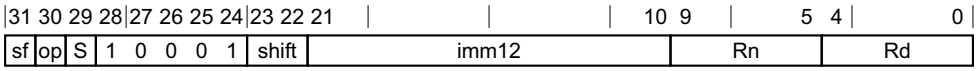
C3.4 Data processing - immediate

This section describes the encoding of the instruction classes in the Data processing (immediate) instruction group, and shows how each instruction class encodes the different instruction forms. For additional information on this functional group of instructions, see *Data processing - immediate* on page C2-140.

Table C3-4 Encoding table for the Data Processing - Immediate functional group

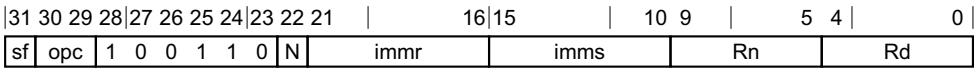
Instruction bits																						Instruction class
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	
-	-	-	1	0	0	0	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	PC-rel. addressing
-	-	-	1	0	0	0	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Add/subtract (immediate)
-	-	-	1	0	0	1	0	0	-	-	-	-	-	-	-	-	-	-	-	-	-	Logical (immediate)
-	-	-	1	0	0	1	0	1	-	-	-	-	-	-	-	-	-	-	-	-	-	Move wide (immediate)
-	-	-	1	0	0	1	1	0	-	-	-	-	-	-	-	-	-	-	-	-	-	Bitfield
-	-	-	1	0	0	1	1	1	-	-	-	-	-	-	-	-	-	-	-	-	-	Extract

C3.4.1 Add/subtract (immediate)



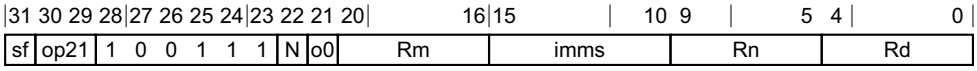
Decode fields				Instruction Page	Variant
sf	op	S	shift		
0	0	0	-	ADD (immediate)	32-bit
0	0	1	-	ADDS (immediate)	32-bit
0	1	0	-	SUB (immediate)	32-bit
0	1	1	-	SUBS (immediate)	32-bit
1	0	0	-	ADD (immediate)	64-bit
1	0	1	-	ADDS (immediate)	64-bit
1	1	0	-	SUB (immediate)	64-bit
1	1	1	-	SUBS (immediate)	64-bit

C3.4.2 Bitfield



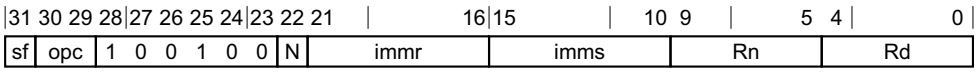
Decode fields			Instruction Page	Variant
sf	opc	N		
0	00	0	SBFM	32-bit
0	01	0	BFM	32-bit
0	10	0	UBFM	32-bit
1	00	1	SBFM	64-bit
1	01	1	BFM	64-bit
1	10	1	UBFM	64-bit

C3.4.3 Extract



Decode fields					Instruction Page	Variant
sf	op21	N	o0	imms		
0	00	0	0	0xxxxx	EXTR	32-bit
1	00	1	0	-	EXTR	64-bit

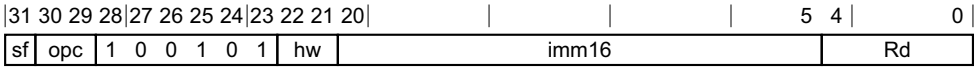
C3.4.4 Logical (immediate)



Decode fields			Instruction Page	Variant
sf	opc	N		
0	00	0	AND (immediate)	32-bit
0	01	0	ORR (immediate)	32-bit
0	10	0	EOR (immediate)	32-bit
0	11	0	ANDS (immediate)	32-bit

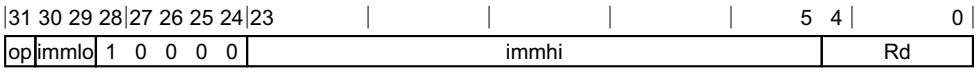
Decode fields			Instruction Page	Variant
sf	opc	N		
1	00	-	AND (immediate)	64-bit
1	01	-	ORR (immediate)	64-bit
1	10	-	EOR (immediate)	64-bit
1	11	-	ANDS (immediate)	64-bit

C3.4.5 Move wide (immediate)



Decode fields			Instruction Page	Variant
sf	opc	hw		
0	00	-	MOVN	32-bit
0	10	-	MOVZ	32-bit
0	11	-	MOVK	32-bit
1	00	-	MOVN	64-bit
1	10	-	MOVZ	64-bit
1	11	-	MOVK	64-bit

C3.4.6 PC-rel. addressing



Decode fields	Instruction Page	Variant
op		
0	ADR	-
1	ADRP	-

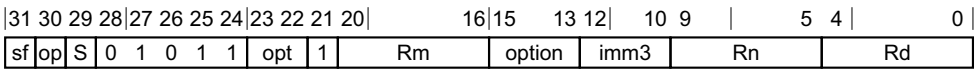
C3.5 Data processing - register

This section describes the encoding of the instruction classes in the Data processing (register) instruction group, and shows how each instruction class encodes the different instruction forms. For additional information on this functional group of instructions, see *Data processing - register* on page C2-145.

Table C3-5 Encoding table for the Data Processing - Register functional group

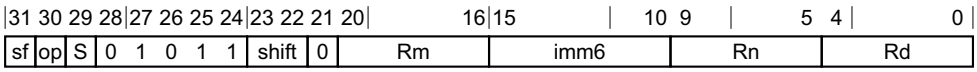
Instruction bits																						Instruction class
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	
-	-	-	0	1	0	1	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Logical (shifted register)
-	-	-	0	1	0	1	1	-	-	0	-	-	-	-	-	-	-	-	-	-	-	Add/subtract (shifted register)
-	-	-	0	1	0	1	1	-	-	1	-	-	-	-	-	-	-	-	-	-	-	Add/subtract (extended register)
-	-	-	1	1	0	1	0	0	0	0	-	-	-	-	-	-	-	-	-	-	-	Add/subtract (with carry)
-	-	-	1	1	0	1	0	0	1	0	-	-	-	-	-	-	-	-	-	0	-	Conditional compare (register)
-	-	-	1	1	0	1	0	0	1	0	-	-	-	-	-	-	-	-	-	1	-	Conditional compare (immediate)
-	-	-	1	1	0	1	0	1	0	0	-	-	-	-	-	-	-	-	-	-	-	Conditional select
-	-	-	1	1	0	1	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Data-processing (3 source)
-	0	-	1	1	0	1	0	1	1	0	-	-	-	-	-	-	-	-	-	-	-	Data-processing (2 source)
-	1	-	1	1	0	1	0	1	1	0	-	-	-	-	-	-	-	-	-	-	-	Data-processing (1 source)

C3.5.1 Add/subtract (extended register)



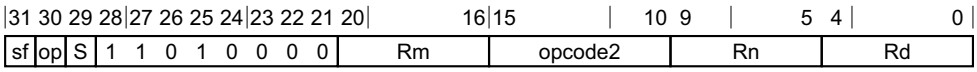
Decode fields					Instruction Page	Variant
sf	op	S	opt	imm3		
0	0	0	00	-	ADD (extended register)	32-bit
0	0	1	00	-	ADDS (extended register)	32-bit
0	1	0	00	-	SUB (extended register)	32-bit
0	1	1	00	-	SUBS (extended register)	32-bit
1	0	0	00	-	ADD (extended register)	64-bit
1	0	1	00	-	ADDS (extended register)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

C3.5.2 Add/subtract (shifted register)



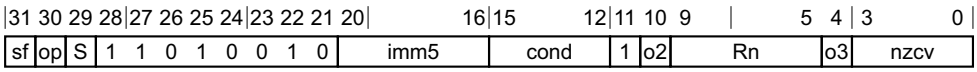
Decode fields					Instruction Page	Variant
sf	op	S	shift	imm6		
0	0	0	-	-	ADD (shifted register)	32-bit
0	0	1	-	-	ADDS (shifted register)	32-bit
0	1	0	-	-	SUB (shifted register)	32-bit
0	1	1	-	-	SUBS (shifted register)	32-bit
1	0	0	-	-	ADD (shifted register)	64-bit
1	0	1	-	-	ADDS (shifted register)	64-bit
1	1	0	-	-	SUB (shifted register)	64-bit
1	1	1	-	-	SUBS (shifted register)	64-bit

C3.5.3 Add/subtract (with carry)



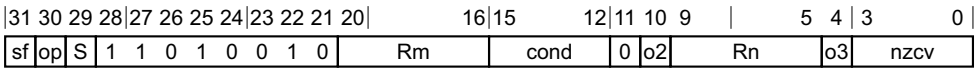
Decode fields				Instruction Page	Variant
sf	op	S	opcode2		
0	0	0	000000	ADC	32-bit
0	0	1	000000	ADCS	32-bit
0	1	0	000000	SBC	32-bit
0	1	1	000000	SBCS	32-bit
1	0	0	000000	ADC	64-bit
1	0	1	000000	ADCS	64-bit
1	1	0	000000	SBC	64-bit
1	1	1	000000	SBCS	64-bit

C3.5.4 Conditional compare (immediate)



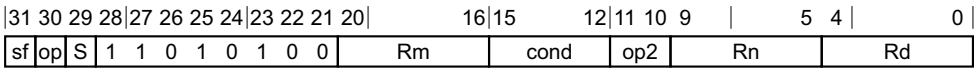
Decode fields					Instruction Page	Variant
sf	op	S	o2	o3		
0	0	1	0	0	CCMN (immediate)	32-bit
0	1	1	0	0	CCMP (immediate)	32-bit
1	0	1	0	0	CCMN (immediate)	64-bit
1	1	1	0	0	CCMP (immediate)	64-bit

C3.5.5 Conditional compare (register)



Decode fields					Instruction Page	Variant
sf	op	S	o2	o3		
0	0	1	0	0	CCMN (register)	32-bit
0	1	1	0	0	CCMP (register)	32-bit
1	0	1	0	0	CCMN (register)	64-bit
1	1	1	0	0	CCMP (register)	64-bit

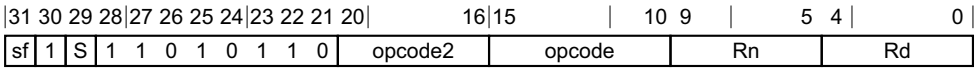
C3.5.6 Conditional select



Decode fields				Instruction Page	Variant
sf	op	S	op2		
0	0	0	00	CSEL	32-bit
0	0	0	01	CSINC	32-bit
0	1	0	00	CSINV	32-bit
0	1	0	01	CSNEG	32-bit

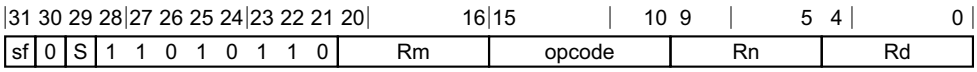
Decode fields				Instruction Page	Variant
sf	op	S	op2		
1	0	0	00	CSEL	64-bit
1	0	0	01	CSINC	64-bit
1	1	0	00	CSINV	64-bit
1	1	0	01	CSNEG	64-bit

C3.5.7 Data-processing (1 source)



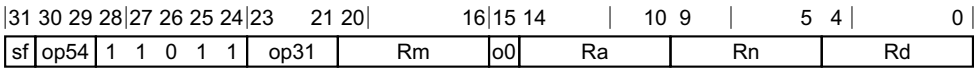
Decode fields				Instruction Page	Variant
sf	S	opcode2	opcode		
0	0	00000	000000	RBIT	32-bit
0	0	00000	000001	REV16	32-bit
0	0	00000	000010	REV	32-bit
0	0	00000	000100	CLZ	32-bit
0	0	00000	000101	CLS	32-bit
1	0	00000	000000	RBIT	64-bit
1	0	00000	000001	REV16	64-bit
1	0	00000	000010	REV32	-
1	0	00000	000011	REV	64-bit
1	0	00000	000100	CLZ	64-bit
1	0	00000	000101	CLS	64-bit

C3.5.8 Data-processing (2 source)



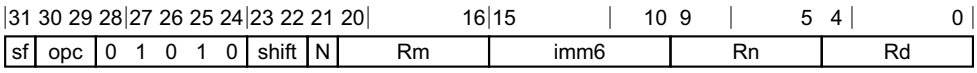
Decode fields			Instruction Page	Variant
sf	S	opcode		
0	0	000010	UDIV	32-bit
0	0	000011	SDIV	32-bit
0	0	001000	LSLV	32-bit
0	0	001001	LSRV	32-bit
0	0	001010	ASRV	32-bit
0	0	001011	RORV	32-bit
0	0	010000	CRC32B, CRC32H, CRC32W, CRC32X	CRC32B
0	0	010001	CRC32B, CRC32H, CRC32W, CRC32X	CRC32H
0	0	010010	CRC32B, CRC32H, CRC32W, CRC32X	CRC32W
0	0	010100	CRC32CB, CRC32CH, CRC32CW, CRC32CX	CRC32CB
0	0	010101	CRC32CB, CRC32CH, CRC32CW, CRC32CX	CRC32CH
0	0	010110	CRC32CB, CRC32CH, CRC32CW, CRC32CX	CRC32CW
1	0	000010	UDIV	64-bit
1	0	000011	SDIV	64-bit
1	0	001000	LSLV	64-bit
1	0	001001	LSRV	64-bit
1	0	001010	ASRV	64-bit
1	0	001011	RORV	64-bit
1	0	010011	CRC32B, CRC32H, CRC32W, CRC32X	CRC32X
1	0	010111	CRC32CB, CRC32CH, CRC32CW, CRC32CX	CRC32CX

C3.5.9 Data-processing (3 source)



Decode fields				Instruction Page	Variant
sf	op54	op31	o0		
0	00	000	0	MADD	32-bit
0	00	000	1	MSUB	32-bit
1	00	000	0	MADD	64-bit
1	00	000	1	MSUB	64-bit
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	-

C3.5.10 Logical (shifted register)



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

Decode fields				Instruction Page	Variant
sf	opc	N	imm6		
1	01	0	-	ORR (shifted register)	64-bit
1	01	1	-	ORN (shifted register)	64-bit
1	10	0	-	EOR (shifted register)	64-bit
1	10	1	-	EON (shifted register)	64-bit
1	11	0	-	ANDS (shifted register)	64-bit
1	11	1	-	BICS (shifted register)	64-bit

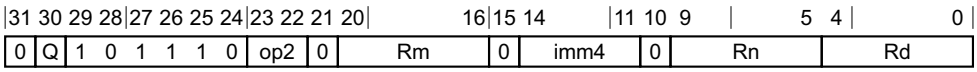
C3.6 Data processing - SIMD and floating point

This section describes the encoding of the instruction classes in the Data processing (SIMD and floating-point) instruction group, and shows how each instruction class encodes the different instruction forms. For additional information on this functional group of instructions, see *Data processing - SIMD and floating-point* on page C2-152.

Table C3-6 Encoding table for the Data Processing - Scalar Floating-Point and Advanced SIMD functional group

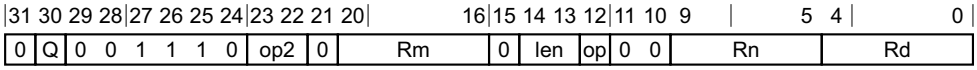
Instruction bits																						Instruction class
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	
-	0	-	1	1	1	1	0	-	-	0	-	-	-	-	-	-	-	-	-	-	-	Floating-point<->fixed-point conversions
-	0	-	1	1	1	1	0	-	-	1	-	-	-	-	-	-	-	-	-	0	1	Floating-point conditional compare
-	0	-	1	1	1	1	0	-	-	1	-	-	-	-	-	-	-	-	-	1	0	Floating-point data-processing (2 source)
-	0	-	1	1	1	1	0	-	-	1	-	-	-	-	-	-	-	-	-	1	1	Floating-point conditional select
-	0	-	1	1	1	1	0	-	-	1	-	-	-	-	-	-	-	-	1	0	0	Floating-point immediate
-	0	-	1	1	1	1	0	-	-	1	-	-	-	-	-	-	-	1	0	0	0	Floating-point compare
-	0	-	1	1	1	1	0	-	-	1	-	-	-	-	-	-	1	0	0	0	0	Floating-point data-processing (1 source)
-	0	-	1	1	1	1	0	-	-	1	-	-	-	-	-	0	0	0	0	0	0	Floating-point<->integer conversions
-	0	-	1	1	1	1	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Floating-point data-processing (3 source)
0	-	-	0	1	1	1	0	-	-	1	-	-	-	-	-	-	-	-	-	-	1	AdvSIMD three same
0	-	-	0	1	1	1	0	-	-	1	-	-	-	-	-	-	-	-	-	0	0	AdvSIMD three different
0	-	-	0	1	1	1	0	-	-	1	0	0	0	0	-	-	-	-	-	1	0	AdvSIMD two-reg misc
0	-	-	0	1	1	1	0	-	-	1	1	0	0	0	-	-	-	-	-	1	0	AdvSIMD across lanes
0	-	-	0	1	1	1	0	0	0	0	-	-	-	-	-	0	-	-	-	-	1	AdvSIMD copy
0	-	-	0	1	1	1	1	-	-	-	-	-	-	-	-	-	-	-	-	-	0	AdvSIMD vector x indexed element
0	-	-	0	1	1	1	1	0	0	0	0	0	-	-	-	-	-	-	-	-	1	AdvSIMD modified immediate
0	-	-	0	1	1	1	1	0	!= 0000				-	-	-	-	-	-	-	-	1	AdvSIMD shift by immediate
0	-	0	0	1	1	1	0	-	-	0	-	-	-	-	-	0	-	-	-	0	0	AdvSIMD TBL/TBX
0	-	0	0	1	1	1	0	-	-	0	-	-	-	-	-	0	-	-	-	1	0	AdvSIMD ZIP/UZP/TRN
0	-	1	0	1	1	1	0	-	-	0	-	-	-	-	-	0	-	-	-	-	0	AdvSIMD EXT
0	1	-	1	1	1	1	0	-	-	1	-	-	-	-	-	-	-	-	-	-	1	AdvSIMD scalar three same
0	1	-	1	1	1	1	0	-	-	1	-	-	-	-	-	-	-	-	-	0	0	AdvSIMD scalar three different
0	1	-	1	1	1	1	0	-	-	1	0	0	0	0	-	-	-	-	-	1	0	AdvSIMD scalar two-reg misc
0	1	-	1	1	1	1	0	-	-	1	1	0	0	0	-	-	-	-	-	1	0	AdvSIMD scalar pairwise
0	1	-	1	1	1	1	0	0	0	0	-	-	-	-	-	0	-	-	-	-	1	AdvSIMD scalar copy
0	1	-	1	1	1	1	1	-	-	-	-	-	-	-	-	-	-	-	-	-	0	AdvSIMD scalar x indexed element
0	1	-	1	1	1	1	1	0	-	-	-	-	-	-	-	-	-	-	-	-	1	AdvSIMD scalar shift by immediate
0	1	0	0	1	1	1	0	-	-	1	0	1	0	0	-	-	-	-	-	1	0	Crypto AES
0	1	0	1	1	1	1	0	-	-	0	-	-	-	-	-	0	-	-	-	0	0	Crypto three-reg SHA
0	1	0	1	1	1	1	0	-	-	1	0	1	0	0	-	-	-	-	-	1	0	Crypto two-reg SHA

C3.6.1 AdvSIMD EXT



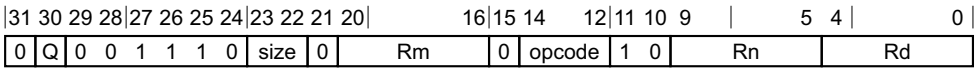
Decode fields	Instruction Page	Variant
op2		
00	EXT	-

C3.6.2 AdvSIMD TBL/TBX



Decode fields			Instruction Page	Variant
op2	len	op		
00	00	0	TBL	Single register table
00	00	1	TBX	Single register table
00	01	0	TBL	Two register table
00	01	1	TBX	Two register table
00	10	0	TBL	Three register table
00	10	1	TBX	Three register table
00	11	0	TBL	Four register table
00	11	1	TBX	Four register table

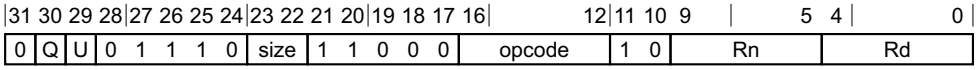
C3.6.3 AdvSIMD ZIP/UZP/TRN



Decode fields	Instruction Page	Variant
opcode		
001	UZP1	-
010	TRN1	-
011	ZIP1	-

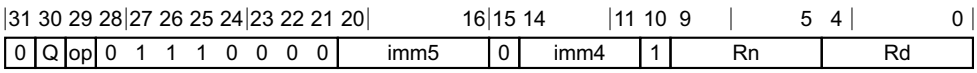
Decode fields		Instruction Page	Variant
opcode			
101		UZP2	-
110		TRN2	-
111		ZIP2	-

C3.6.4 AdvSIMD across lanes



Decode fields			Instruction Page	Variant
U	size	opcode		
0	-	00011	SADDLV	-
0	-	01010	SMAXV	-
0	-	11010	SMINV	-
0	-	11011	ADDV	-
1	-	00011	UADDLV	-
1	-	01010	UMAXV	-
1	-	11010	UMINV	-
1	0x	01100	FMAXNMV	-
1	0x	01111	FMAXV	-
1	1x	01100	FMINNMV	-
1	1x	01111	FMINV	-

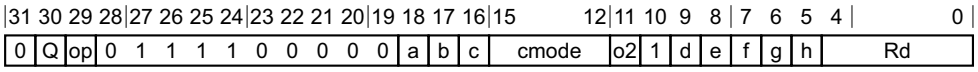
C3.6.5 AdvSIMD copy



Decode fields				Instruction Page	Variant
Q	op	imm5	imm4		
-	0	-	0000	DUP (element)	Vector
-	0	-	0001	DUP (general)	-
0	0	-	0101	SMOV	32-bit

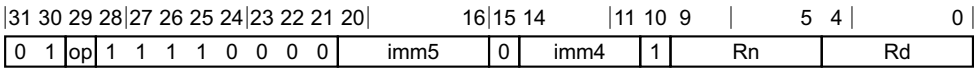
Decode fields				Instruction Page	Variant
Q	op	imm5	imm4		
0	0	-	0111	UMOV	32-bit
1	0	-	0011	INS (general)	-
1	0	-	0101	SMOV	64-bit
1	0	-	0111	UMOV	64-bit
1	1	-	-	INS (element)	-

C3.6.6 AdvSIMD modified immediate



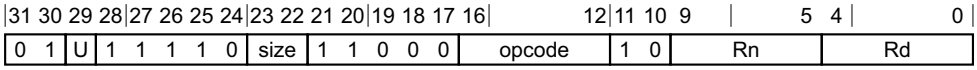
Decode fields				Instruction Page	Variant
Q	op	cmode	o2		
-	0	0xx0	0	MOVI	32-bit shifted immediate
-	0	0xx1	0	ORR (vector, immediate)	32-bit
-	0	10x0	0	MOVI	16-bit shifted immediate
-	0	10x1	0	ORR (vector, immediate)	16-bit
-	0	110x	0	MOVI	32-bit shifting ones
-	0	1110	0	MOVI	8-bit
-	0	1111	0	FMOV (vector, immediate)	Single-precision
-	1	0xx0	0	MVNI	32-bit shifted immediate
-	1	0xx1	0	BIC (vector, immediate)	32-bit
-	1	10x0	0	MVNI	16-bit shifted immediate
-	1	10x1	0	BIC (vector, immediate)	16-bit
-	1	110x	0	MVNI	32-bit shifting ones
0	1	1110	0	MOVI	64-bit scalar
1	1	1110	0	MOVI	64-bit vector
1	1	1111	0	FMOV (vector, immediate)	Double-precision

C3.6.7 AdvSIMD scalar copy



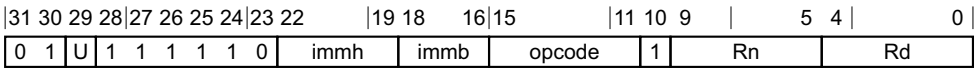
Decode fields			Instruction Page	Variant
op	imm5	imm4		
0	-	0000	DUP (element)	Scalar

C3.6.8 AdvSIMD scalar pairwise



Decode fields			Instruction Page	Variant
U	size	opcode		
0	-	11011	ADDP (scalar)	-
1	0x	01100	FMAXNMP (scalar)	-
1	0x	01101	FADDP (scalar)	-
1	0x	01111	FMAXP (scalar)	-
1	1x	01100	FMINNMP (scalar)	-
1	1x	01111	FMINP (scalar)	-

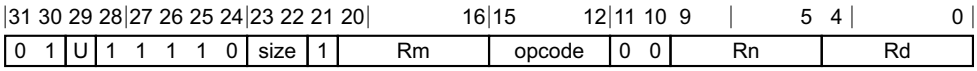
C3.6.9 AdvSIMD scalar shift by immediate



Decode fields			Instruction Page	Variant
U	immh	opcode		
0	!= 0000	00000	SSHR	Scalar
0	!= 0000	00010	SSRA	Scalar
0	!= 0000	00100	SRSHR	Scalar
0	!= 0000	00110	SRSRA	Scalar
0	!= 0000	01010	SHL	Scalar

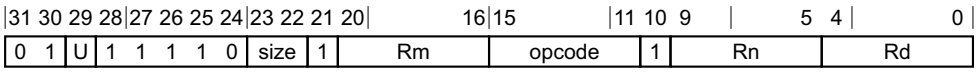
Decode fields			Instruction Page	Variant
U	immh	opcode		
0	!= 0000	01110	SQSHL (immediate)	Scalar
0	!= 0000	10010	SQSHRN, SQSHRN2	Scalar
0	!= 0000	10011	SQRSHRN, SQRSHRN2	Scalar
0	!= 0000	11100	SCVTF (vector, fixed-point)	Scalar
0	!= 0000	11111	FCVTZS (vector, fixed-point)	Scalar
1	!= 0000	00000	USHR	Scalar
1	!= 0000	00010	USRA	Scalar
1	!= 0000	00100	URSHR	Scalar
1	!= 0000	00110	URSRA	Scalar
1	!= 0000	01000	SRI	Scalar
1	!= 0000	01010	SLI	Scalar
1	!= 0000	01100	SQSHLU	Scalar
1	!= 0000	01110	UQSHL (immediate)	Scalar
1	!= 0000	10000	SQSHRUN, SQSHRUN2	Scalar
1	!= 0000	10001	SQRSHRUN, SQRSHRUN2	Scalar
1	!= 0000	10010	UQSHRN	Scalar
1	!= 0000	10011	UQRSHRN, UQRSHRN2	Scalar
1	!= 0000	11100	UCVTF (vector, fixed-point)	Scalar
1	!= 0000	11111	FCVTZU (vector, fixed-point)	Scalar

C3.6.10 AdvSIMD scalar three different



Decode fields		Instruction Page	Variant
U	opcode		
0	1001	SQDMLAL, SQDMLAL2 (vector)	Scalar
0	1011	SQDMLSL, SQDMLSL2 (vector)	Scalar
0	1101	SQDMULL, SQDMULL2 (vector)	Scalar

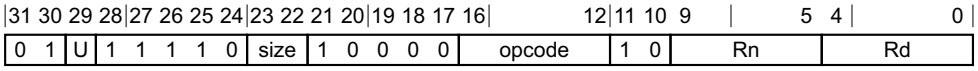
C3.6.11 AdvSIMD scalar three same



Decode fields			Instruction Page	Variant
U	size	opcode		
0	-	00001	SQADD	Scalar
0	-	00101	SQSUB	Scalar
0	-	00110	CMGT (register)	Scalar
0	-	00111	CMGE (register)	Scalar
0	-	01000	SSHL	Scalar
0	-	01001	SQSHL (register)	Scalar
0	-	01010	SRSHL	Scalar
0	-	01011	SQRSHL	Scalar
0	-	10000	ADD (vector)	Scalar
0	-	10001	CMTST	Scalar
0	-	10110	SQDMULH (vector)	Scalar
0	0x	11011	FMULX	Scalar
0	0x	11100	FCMEQ (register)	Scalar
0	0x	11111	FRECPS	Scalar
0	1x	11111	FRSQRTS	Scalar
1	-	00001	UQADD	Scalar
1	-	00101	UQSUB	Scalar
1	-	00110	CMHI (register)	Scalar
1	-	00111	CMHS (register)	Scalar
1	-	01000	USHL	Scalar
1	-	01001	UQSHL (register)	Scalar
1	-	01010	URSHL	Scalar
1	-	01011	UQRSHL	Scalar
1	-	10000	SUB (vector)	Scalar
1	-	10001	CMEQ (register)	Scalar
1	-	10110	SQRDMULH (vector)	Scalar
1	0x	11100	FCMGE (register)	Scalar

Decode fields			Instruction Page	Variant
U	size	opcode		
1	0x	11101	FACGE	Scalar
1	1x	11010	FABD	Scalar
1	1x	11100	FCMGT (register)	Scalar
1	1x	11101	FACGT	Scalar

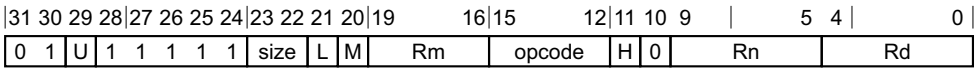
C3.6.12 AdvSIMD scalar two-reg misc



Decode fields			Instruction Page	Variant
U	size	opcode		
0	-	00011	SUQADD	Scalar
0	-	00111	SQABS	Scalar
0	-	01000	CMGT (zero)	Scalar
0	-	01001	CMEQ (zero)	Scalar
0	-	01010	CMLT (zero)	Scalar
0	-	01011	ABS	Scalar
0	-	10100	SQXTN, SQXTN2	Scalar
0	0x	11010	FCVTNS (vector)	Scalar
0	0x	11011	FCVTMS (vector)	Scalar
0	0x	11100	FCVTAS (vector)	Scalar
0	0x	11101	SCVTF (vector, integer)	Scalar
0	1x	01100	FCMGT (zero)	Scalar
0	1x	01101	FCMEQ (zero)	Scalar
0	1x	01110	FCMLT (zero)	Scalar
0	1x	11010	FCVTPS (vector)	Scalar
0	1x	11011	FCVTZS (vector, integer)	Scalar
0	1x	11101	FRECPE	Scalar
0	1x	11111	FRECPX	-
1	-	00011	USQADD	Scalar
1	-	00111	SQNEG	Scalar

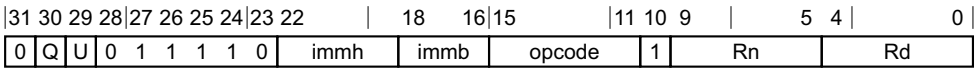
Decode fields			Instruction Page	Variant
U	size	opcode		
1	-	01000	CMGE (zero)	Scalar
1	-	01001	CMLE (zero)	Scalar
1	-	01011	NEG (vector)	Scalar
1	-	10010	SQXTUN, SQXTUN2	Scalar
1	-	10100	UQXTN, UQXTN2	Scalar
1	0x	10110	FCVTXN, FCVTXN2	Scalar
1	0x	11010	FCVTNU (vector)	Scalar
1	0x	11011	FCVTMU (vector)	Scalar
1	0x	11100	FCVTAU (vector)	Scalar
1	0x	11101	UCVTF (vector, integer)	Scalar
1	1x	01100	FCMGE (zero)	Scalar
1	1x	01101	FCMLE (zero)	Scalar
1	1x	11010	FCVTPU (vector)	Scalar
1	1x	11011	FCVTZU (vector, integer)	Scalar
1	1x	11101	FRSQRTE	Scalar

C3.6.13 AdvSIMD scalar x indexed element



Decode fields			Instruction Page	Variant
U	size	opcode		
0	-	0011	SQDMLAL, SQDMLAL2 (by element)	Scalar
0	-	0111	SQDMLSL, SQDMLSL2 (by element)	Scalar
0	-	1011	SQDMULL, SQDMULL2 (by element)	Scalar
0	-	1100	SQDMULH (by element)	Scalar
0	-	1101	SQRDMULH (by element)	Scalar
0	1x	0001	FMLA (by element)	Scalar
0	1x	0101	FMLS (by element)	Scalar
0	1x	1001	FMUL (by element)	Scalar
1	1x	1001	FMULX (by element)	Scalar

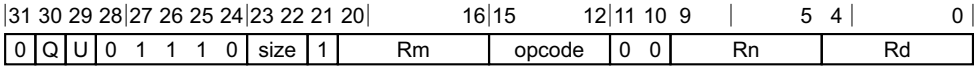
C3.6.14 AdvSIMD shift by immediate



Decode fields		Instruction Page	Variant
U	opcode		
0	00000	SSHR	Vector
0	00010	SSRA	Vector
0	00100	SRSHR	Vector
0	00110	SRSRA	Vector
0	01010	SHL	Vector
0	01110	SQSHL (immediate)	Vector
0	10000	SHRN, SHRN2	-
0	10001	RSHRN, RSHRN2	-
0	10010	SQSHRN, SQSHRN2	Vector
0	10011	SQRSHRN, SQRSHRN2	Vector
0	10100	SSHLL, SSHLL2	-
0	11100	SCVTF (vector, fixed-point)	Vector
0	11111	FCVTZS (vector, fixed-point)	Vector
1	00000	USHR	Vector
1	00010	USRA	Vector
1	00100	URSHR	Vector
1	00110	URSRA	Vector
1	01000	SRI	Vector
1	01010	SLI	Vector
1	01100	SQSHLU	Vector
1	01110	UQSHL (immediate)	Vector
1	10000	SQSHRUN, SQSHRUN2	Vector
1	10001	SQRSHRUN, SQRSHRUN2	Vector
1	10010	UQSHRN	Vector
1	10011	UQRSHRN, UQRSHRN2	Vector

Decode fields		Instruction Page	Variant
U	opcode		
1	10100	USHLL, USHLL2	-
1	11100	UCVTF (vector, fixed-point)	Vector
1	11111	FCVTZU (vector, fixed-point)	Vector

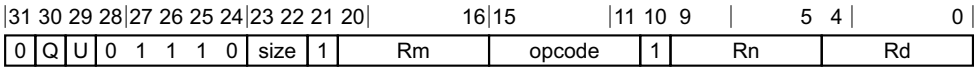
C3.6.15 AdvSIMD three different



Decode fields		Instruction Page	Variant
U	opcode		
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)	Vector
0	1010	SMLSL, SMLSL2 (vector)	-
0	1011	SQDMLSL, SQDMLSL2 (vector)	Vector
0	1100	SMULL, SMULL2 (vector)	-
0	1101	SQDMULL, SQDMULL2 (vector)	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	-

Decode fields		Instruction Page	Variant
U	opcode		
1	0110	RSUBHN, RSUBHN2	-
1	0111	UABDL, UABDL2	-
1	1000	UMLAL, UMLAL2 (vector)	-
1	1010	UMLSL, UMLSL2 (vector)	-
1	1100	UMULL, UMULL2 (vector)	-

C3.6.16 AdvSIMD three same

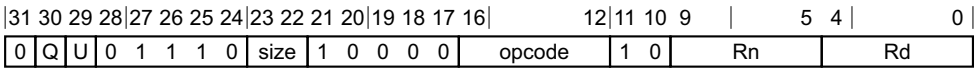


Decode fields			Instruction Page	Variant
U	size	opcode		
0	-	00000	SHADD	-
0	-	00001	SQADD	Vector
0	-	00010	SRHADD	-
0	-	00100	SHSUB	-
0	-	00101	SQSUB	Vector
0	-	00110	CMGT (register)	Vector
0	-	00111	CMGE (register)	Vector
0	-	01000	SSHL	Vector
0	-	01001	SQSHL (register)	Vector
0	-	01010	SRSHL	Vector
0	-	01011	SQRSHL	Vector
0	-	01100	SMAX	-
0	-	01101	SMIN	-
0	-	01110	SABD	-
0	-	01111	SABA	-
0	-	10000	ADD (vector)	Vector
0	-	10001	CMTST	Vector
0	-	10010	MLA (vector)	-
0	-	10011	MUL (vector)	-

Decode fields			Instruction Page	Variant
U	size	opcode		
0	-	10100	SMAXP	-
0	-	10101	SMINP	-
0	-	10110	SQDMULH (vector)	Vector
0	-	10111	ADDP (vector)	-
0	0x	11000	FMAXNM (vector)	-
0	0x	11001	FMLA (vector)	-
0	0x	11010	FADD (vector)	-
0	0x	11011	FMULX	Vector
0	0x	11100	FCMEQ (register)	Vector
0	0x	11110	FMAX (vector)	-
0	0x	11111	FRECPS	Vector
0	00	00011	AND (vector)	-
0	01	00011	BIC (vector, register)	-
0	1x	11000	FMINNM (vector)	-
0	1x	11001	FMLS (vector)	-
0	1x	11010	FSUB (vector)	-
0	1x	11110	FMIN (vector)	-
0	1x	11111	FRSQRTS	Vector
0	10	00011	ORR (vector, register)	-
0	11	00011	ORN (vector)	-
1	-	00000	UHADD	-
1	-	00001	UQADD	Vector
1	-	00010	URHADD	-
1	-	00100	UHSUB	-
1	-	00101	UQSUB	Vector
1	-	00110	CMHI (register)	Vector
1	-	00111	CMHS (register)	Vector
1	-	01000	USHL	Vector
1	-	01001	UQSHL (register)	Vector
1	-	01010	URSHL	Vector
1	-	01011	UQRSHL	Vector
1	-	01100	UMAX	-

Decode fields			Instruction Page	Variant
U	size	opcode		
1	-	01101	UMIN	-
1	-	01110	UABD	-
1	-	01111	UABA	-
1	-	10000	SUB (vector)	Vector
1	-	10001	CMEQ (register)	Vector
1	-	10010	MLS (vector)	-
1	-	10011	PMUL	-
1	-	10100	UMAXP	-
1	-	10101	UMINP	-
1	-	10110	SQRDMULH (vector)	Vector
1	0x	11000	FMAXNMP (vector)	-
1	0x	11010	FADDP (vector)	-
1	0x	11011	FMUL (vector)	-
1	0x	11100	FCMGE (register)	Vector
1	0x	11101	FACGE	Vector
1	0x	11110	FMAXP (vector)	-
1	0x	11111	FDIV (vector)	-
1	00	00011	EOR (vector)	-
1	01	00011	BSL	-
1	1x	11000	FMINNMP (vector)	-
1	1x	11010	FABD	Vector
1	1x	11100	FCMGT (register)	Vector
1	1x	11101	FACGT	Vector
1	1x	11110	FMINP (vector)	-
1	10	00011	BIT	-
1	11	00011	BIF	-

C3.6.17 AdvSIMD two-reg misc

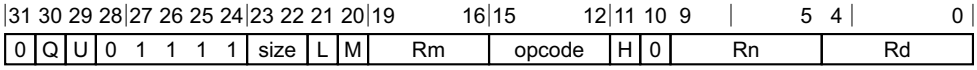


Decode fields			Instruction Page	Variant
U	size	opcode		
0	-	00000	REV64	-
0	-	00001	REV16 (vector)	-
0	-	00010	SADDLP	-
0	-	00011	SUQADD	Vector
0	-	00100	CLS (vector)	-
0	-	00101	CNT	-
0	-	00110	SADALP	-
0	-	00111	SQABS	Vector
0	-	01000	CMGT (zero)	Vector
0	-	01001	CMEQ (zero)	Vector
0	-	01010	CMLT (zero)	Vector
0	-	01011	ABS	Vector
0	-	10010	XTN, XTN2	-
0	-	10100	SQXTN, SQXTN2	Vector
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)	Vector
0	0x	11011	FCVTMS (vector)	Vector
0	0x	11100	FCVTAS (vector)	Vector
0	0x	11101	SCVTF (vector, integer)	Vector
0	1x	01100	FCMGT (zero)	Vector
0	1x	01101	FCMEQ (zero)	Vector
0	1x	01110	FCMLT (zero)	Vector
0	1x	01111	FABS (vector)	-
0	1x	11000	FRINTP (vector)	-

Decode fields			Instruction Page	Variant
U	size	opcode		
0	1x	11001	FRINTZ (vector)	-
0	1x	11010	FCVTPS (vector)	Vector
0	1x	11011	FCVTZS (vector, integer)	Vector
0	1x	11100	URECPE	-
0	1x	11101	FRECPE	Vector
1	-	00000	REV32 (vector)	-
1	-	00010	UADDLP	-
1	-	00011	USQADD	Vector
1	-	00100	CLZ (vector)	-
1	-	00110	UADALP	-
1	-	00111	SQNEG	Vector
1	-	01000	CMGE (zero)	Vector
1	-	01001	CMLE (zero)	Vector
1	-	01011	NEG (vector)	Vector
1	-	10010	SQXTUN, SQXTUN2	Vector
1	-	10011	SHLL, SHLL2	-
1	-	10100	UQXTN, UQXTN2	Vector
1	0x	10110	FCVTXN, FCVTXN2	Vector
1	0x	11000	FRINTA (vector)	-
1	0x	11001	FRINTX (vector)	-
1	0x	11010	FCVTNU (vector)	Vector
1	0x	11011	FCVTMU (vector)	Vector
1	0x	11100	FCVTAU (vector)	Vector
1	0x	11101	UCVTF (vector, integer)	Vector
1	00	00101	NOT	-
1	01	00101	RBIT (vector)	-
1	1x	01100	FCMGE (zero)	Vector
1	1x	01101	FCMLE (zero)	Vector
1	1x	01111	FNEG (vector)	-
1	1x	11001	FRINTI (vector)	-
1	1x	11010	FCVTPU (vector)	Vector
1	1x	11011	FCVTZU (vector, integer)	Vector

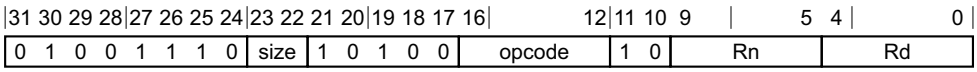
Decode fields			Instruction Page	Variant
U	size	opcode		
1	1x	11100	URSQRTE	-
1	1x	11101	FRSQRTE	Vector
1	1x	11111	FSQRT (vector)	-

C3.6.18 AdvSIMD vector x indexed element



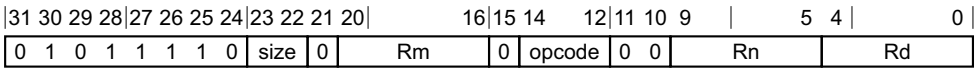
Decode fields			Instruction Page	Variant
U	size	opcode		
0	-	0010	SMLAL, SMLAL2 (by element)	-
0	-	0011	SQDMLAL, SQDMLAL2 (by element)	Vector
0	-	0110	SMLSL, SMLSL2 (by element)	-
0	-	0111	SQDMLSL, SQDMLSL2 (by element)	Vector
0	-	1000	MUL (by element)	-
0	-	1010	SMULL, SMULL2 (by element)	-
0	-	1011	SQDMULL, SQDMULL2 (by element)	Vector
0	-	1100	SQDMULH (by element)	Vector
0	-	1101	SQRDMULH (by element)	Vector
0	1x	0001	FMLA (by element)	Vector
0	1x	0101	FMLS (by element)	Vector
0	1x	1001	FMUL (by element)	Vector
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	1x	1001	FMULX (by element)	Vector

C3.6.19 Crypto AES



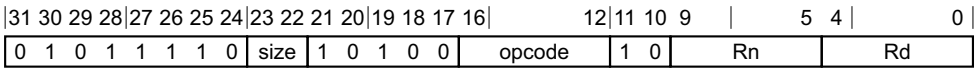
Decode fields		Instruction Page	Variant
size	opcode		
00	00100	AESE	-
00	00101	AESD	-
00	00110	AESMC	-
00	00111	AESIMC	-

C3.6.20 Crypto three-reg SHA



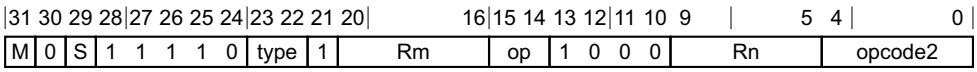
Decode fields		Instruction Page	Variant
size	opcode		
00	000	SHA1C	-
00	001	SHA1P	-
00	010	SHA1M	-
00	011	SHA1SU0	-
00	100	SHA256H	-
00	101	SHA256H2	-
00	110	SHA256SU1	-

C3.6.21 Crypto two-reg SHA



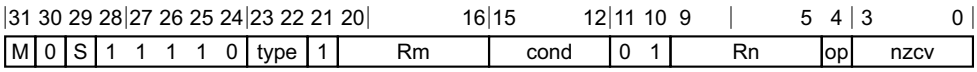
Decode fields		Instruction Page	Variant
size	opcode		
00	00000	SHA1H	-
00	00001	SHA1SU1	-
00	00010	SHA256SU0	-

C3.6.22 Floating-point compare



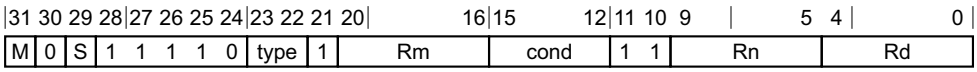
Decode fields					Instruction Page	Variant
M	S	type	op	opcode2		
0	0	00	00	00000	FCMP	Single-precision
0	0	00	00	01000	FCMP	Single-precision, zero
0	0	00	00	10000	FCMPE	Single-precision
0	0	00	00	11000	FCMPE	Single-precision, zero
0	0	01	00	00000	FCMP	Double-precision
0	0	01	00	01000	FCMP	Double-precision, zero
0	0	01	00	10000	FCMPE	Double-precision
0	0	01	00	11000	FCMPE	Double-precision, zero

C3.6.23 Floating-point conditional compare



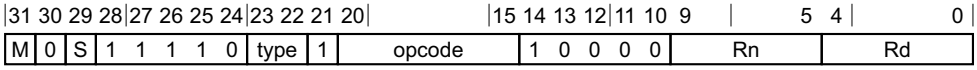
Decode fields				Instruction Page	Variant
M	S	type	op		
0	0	00	0	FCCMP	Single-precision
0	0	00	1	FCCMPE	Single-precision
0	0	01	0	FCCMP	Double-precision
0	0	01	1	FCCMPE	Double-precision

C3.6.24 Floating-point conditional select



Decode fields			Instruction Page	Variant
M	S	type		
0	0	00	FCSEL	Single-precision
0	0	01	FCSEL	Double-precision

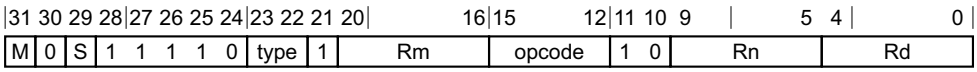
C3.6.25 Floating-point data-processing (1 source)



Decode fields				Instruction Page	Variant
M	S	type	opcode		
0	0	00	000000	FMOV (register)	Single-precision
0	0	00	000001	FABS (scalar)	Single-precision
0	0	00	000010	FNEG (scalar)	Single-precision
0	0	00	000011	FSQRT (scalar)	Single-precision
0	0	00	000101	FCVT	Single-precision to double-precision
0	0	00	000111	FCVT	Single-precision to half-precision

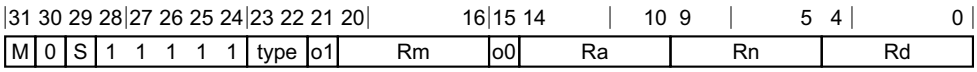
Decode fields				Instruction Page	Variant
M	S	type	opcode		
0	0	00	001000	FRINTN (scalar)	Single-precision
0	0	00	001001	FRINTP (scalar)	Single-precision
0	0	00	001010	FRINTM (scalar)	Single-precision
0	0	00	001011	FRINTZ (scalar)	Single-precision
0	0	00	001100	FRINTA (scalar)	Single-precision
0	0	00	001110	FRINTX (scalar)	Single-precision
0	0	00	001111	FRINTI (scalar)	Single-precision
0	0	01	000000	FMOV (register)	Double-precision
0	0	01	000001	FABS (scalar)	Double-precision
0	0	01	000010	FNEG (scalar)	Double-precision
0	0	01	000011	FSQRT (scalar)	Double-precision
0	0	01	000100	FCVT	Double-precision to single-precision
0	0	01	000111	FCVT	Double-precision to half-precision
0	0	01	001000	FRINTN (scalar)	Double-precision
0	0	01	001001	FRINTP (scalar)	Double-precision
0	0	01	001010	FRINTM (scalar)	Double-precision
0	0	01	001011	FRINTZ (scalar)	Double-precision
0	0	01	001100	FRINTA (scalar)	Double-precision
0	0	01	001110	FRINTX (scalar)	Double-precision
0	0	01	001111	FRINTI (scalar)	Double-precision
0	0	11	000100	FCVT	Half-precision to single-precision
0	0	11	000101	FCVT	Half-precision to double-precision

C3.6.26 Floating-point data-processing (2 source)



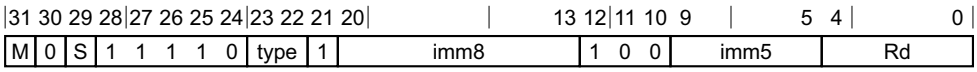
Decode fields				Instruction Page	Variant
M	S	type	opcode		
0	0	00	0000	FMUL (scalar)	Single-precision
0	0	00	0001	FDIV (scalar)	Single-precision
0	0	00	0010	FADD (scalar)	Single-precision
0	0	00	0011	FSUB (scalar)	Single-precision
0	0	00	0100	FMAX (scalar)	Single-precision
0	0	00	0101	FMIN (scalar)	Single-precision
0	0	00	0110	FMAXNM (scalar)	Single-precision
0	0	00	0111	FMINNM (scalar)	Single-precision
0	0	00	1000	FNMUL	Single-precision
0	0	01	0000	FMUL (scalar)	Double-precision
0	0	01	0001	FDIV (scalar)	Double-precision
0	0	01	0010	FADD (scalar)	Double-precision
0	0	01	0011	FSUB (scalar)	Double-precision
0	0	01	0100	FMAX (scalar)	Double-precision
0	0	01	0101	FMIN (scalar)	Double-precision
0	0	01	0110	FMAXNM (scalar)	Double-precision
0	0	01	0111	FMINNM (scalar)	Double-precision
0	0	01	1000	FNMUL	Double-precision

C3.6.27 Floating-point data-processing (3 source)



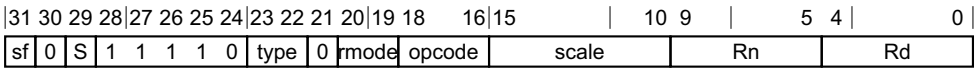
Decode fields					Instruction Page	Variant
M	S	type	o1	o0		
0	0	00	0	0	FMADD	Single-precision
0	0	00	0	1	FMSUB	Single-precision
0	0	00	1	0	FNMADD	Single-precision
0	0	00	1	1	FNMSUB	Single-precision
0	0	01	0	0	FMADD	Double-precision
0	0	01	0	1	FMSUB	Double-precision
0	0	01	1	0	FNMADD	Double-precision
0	0	01	1	1	FNMSUB	Double-precision

C3.6.28 Floating-point immediate



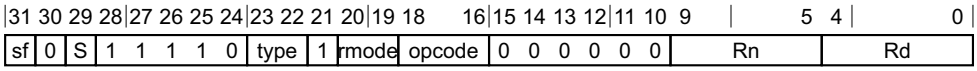
Decode fields				Instruction Page	Variant
M	S	type	imm5		
0	0	00	00000	FMOV (scalar, immediate)	Single-precision
0	0	01	00000	FMOV (scalar, immediate)	Double-precision

C3.6.29 Floating-point<->fixed-point conversions



Decode fields						Instruction Page	Variant
sf	S	type	rmode	opcode	scale		
0	0	00	00	010	-	SCVTF (scalar, fixed-point)	32-bit to single-precision
0	0	00	00	011	-	UCVTF (scalar, fixed-point)	32-bit to single-precision
0	0	00	11	000	-	FCVTZS (scalar, fixed-point)	Single-precision to 32-bit
0	0	00	11	001	-	FCVTZU (scalar, fixed-point)	Single-precision to 32-bit
0	0	01	00	010	-	SCVTF (scalar, fixed-point)	32-bit to double-precision
0	0	01	00	011	-	UCVTF (scalar, fixed-point)	32-bit to double-precision
0	0	01	11	000	-	FCVTZS (scalar, fixed-point)	Double-precision to 32-bit
0	0	01	11	001	-	FCVTZU (scalar, fixed-point)	Double-precision to 32-bit
1	0	00	00	010	-	SCVTF (scalar, fixed-point)	64-bit to single-precision
1	0	00	00	011	-	UCVTF (scalar, fixed-point)	64-bit to single-precision
1	0	00	11	000	-	FCVTZS (scalar, fixed-point)	Single-precision to 64-bit
1	0	00	11	001	-	FCVTZU (scalar, fixed-point)	Single-precision to 64-bit
1	0	01	00	010	-	SCVTF (scalar, fixed-point)	64-bit to double-precision
1	0	01	00	011	-	UCVTF (scalar, fixed-point)	64-bit to double-precision
1	0	01	11	000	-	FCVTZS (scalar, fixed-point)	Double-precision to 64-bit
1	0	01	11	001	-	FCVTZU (scalar, fixed-point)	Double-precision to 64-bit

C3.6.30 Floating-point<->integer conversions



Decode fields					Instruction Page	Variant
sf	S	type	rmode	opcode		
0	0	00	00	000	FCVTNS (scalar)	Single-precision to 32-bit
0	0	00	00	001	FCVTNU (scalar)	Single-precision to 32-bit
0	0	00	00	010	SCVTF (scalar, integer)	32-bit to single-precision
0	0	00	00	011	UCVTF (scalar, integer)	32-bit to single-precision

Decode fields					Instruction Page	Variant
sf	S	type	rmode	opcode		
0	0	00	00	100	FCVTAS (scalar)	Single-precision to 32-bit
0	0	00	00	101	FCVTAU (scalar)	Single-precision to 32-bit
0	0	00	00	110	FMOV (general)	Single-precision to 32-bit
0	0	00	00	111	FMOV (general)	32-bit to single-precision
0	0	00	01	000	FCVTPS (scalar)	Single-precision to 32-bit
0	0	00	01	001	FCVTPU (scalar)	Single-precision to 32-bit
0	0	00	10	000	FCVTMS (scalar)	Single-precision to 32-bit
0	0	00	10	001	FCVTMU (scalar)	Single-precision to 32-bit
0	0	00	11	000	FCVTZS (scalar, integer)	Single-precision to 32-bit
0	0	00	11	001	FCVTZU (scalar, integer)	Single-precision to 32-bit
0	0	01	00	000	FCVTNS (scalar)	Double-precision to 32-bit
0	0	01	00	001	FCVTNU (scalar)	Double-precision to 32-bit
0	0	01	00	010	SCVTF (scalar, integer)	32-bit to double-precision
0	0	01	00	011	UCVTF (scalar, integer)	32-bit to double-precision
0	0	01	00	100	FCVTAS (scalar)	Double-precision to 32-bit
0	0	01	00	101	FCVTAU (scalar)	Double-precision to 32-bit
0	0	01	01	000	FCVTPS (scalar)	Double-precision to 32-bit
0	0	01	01	001	FCVTPU (scalar)	Double-precision to 32-bit
0	0	01	10	000	FCVTMS (scalar)	Double-precision to 32-bit
0	0	01	10	001	FCVTMU (scalar)	Double-precision to 32-bit
0	0	01	11	000	FCVTZS (scalar, integer)	Double-precision to 32-bit
0	0	01	11	001	FCVTZU (scalar, integer)	Double-precision to 32-bit
1	0	00	00	000	FCVTNS (scalar)	Single-precision to 64-bit
1	0	00	00	001	FCVTNU (scalar)	Single-precision to 64-bit
1	0	00	00	010	SCVTF (scalar, integer)	64-bit to single-precision
1	0	00	00	011	UCVTF (scalar, integer)	64-bit to single-precision
1	0	00	00	100	FCVTAS (scalar)	Single-precision to 64-bit
1	0	00	00	101	FCVTAU (scalar)	Single-precision to 64-bit
1	0	00	01	000	FCVTPS (scalar)	Single-precision to 64-bit
1	0	00	01	001	FCVTPU (scalar)	Single-precision to 64-bit
1	0	00	10	000	FCVTMS (scalar)	Single-precision to 64-bit
1	0	00	10	001	FCVTMU (scalar)	Single-precision to 64-bit

Decode fields					Instruction Page	Variant
sf	S	type	rmode	opcode		
1	0	00	11	000	FCVTZS (scalar, integer)	Single-precision to 64-bit
1	0	00	11	001	FCVTZU (scalar, integer)	Single-precision to 64-bit
1	0	01	00	000	FCVTNS (scalar)	Double-precision to 64-bit
1	0	01	00	001	FCVTNU (scalar)	Double-precision to 64-bit
1	0	01	00	010	SCVTF (scalar, integer)	64-bit to double-precision
1	0	01	00	011	UCVTF (scalar, integer)	64-bit to double-precision
1	0	01	00	100	FCVTAS (scalar)	Double-precision to 64-bit
1	0	01	00	101	FCVTAU (scalar)	Double-precision to 64-bit
1	0	01	00	110	FMOV (general)	Double-precision to 64-bit
1	0	01	00	111	FMOV (general)	64-bit to double-precision
1	0	01	01	000	FCVTPS (scalar)	Double-precision to 64-bit
1	0	01	01	001	FCVTPU (scalar)	Double-precision to 64-bit
1	0	01	10	000	FCVTMS (scalar)	Double-precision to 64-bit
1	0	01	10	001	FCVTMU (scalar)	Double-precision to 64-bit
1	0	01	11	000	FCVTZS (scalar, integer)	Double-precision to 64-bit
1	0	01	11	001	FCVTZU (scalar, integer)	Double-precision to 64-bit
1	0	10	01	110	FMOV (general)	Top half of 128-bit to 64-bit
1	0	10	01	111	FMOV (general)	64-bit to top half of 128-bit