

Top-level encodings for A64

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
op0		op1																													

Decode fields

op0 op1

Instruction details

op0	op1	Instruction details
0	0000	Reserved
1	0000	SME encodings
	0001	UNALLOCATED
	0010	SVE encodings
	0011	UNALLOCATED
	100x	Data Processing -- Immediate
	101x	Branches, Exception Generating and System instructions
	x1x0	Loads and Stores
	x101	Data Processing -- Register
	x111	Data Processing -- Scalar Floating-Point and Advanced SIMD

Reserved

These instructions are under the [top-level](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	op0	0000		op1																											

Decode fields

op0 op1

Instruction details

00	000000000	UDF
	!= 000000000	UNALLOCATED
!= 00		UNALLOCATED

SME encodings

These instructions are under the [top-level](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	op0	0000		op1																											

Decode fields

op0

op1

op2

Instruction details

0x	x10xxxxxxxxxxxxxx	xx1x	UNALLOCATED
0x	x11xxxxxxxxxxxxxx	x0xx	SME Outer Product - 64 bit
0x	x11xxxxxxxxxxxxxx	x1xx	UNALLOCATED
00	x0xxxxxxxxxxxxxx		UNALLOCATED
00	x10xxxxxxxxxxxxxx	x00x	SME FP Outer Product - 32 bit
00	x10xxxxxxxxxxxxxx	x10x	SME2 Outer Product - Misc
01	x10xxxxxxxxxxxxxx	xx0x	SME Integer Outer Product - 32 bit
01	00xxxxxxxxxxxxxx		SME2 Multi-vector - Memory (Contiguous)
01	10xxxxxxxxxxxxxx		SME2 Multi-vector - Memory (Strided)
10	0xx000x0xxxxxxxx	0xxx	SME Move into Array
10	0xx000x0xxxxxxxx	1xxx	UNALLOCATED
10	0xx000x1xxxxxxxx		SME Move from Array
10	0xx010xxxxxxxxxx	x0xx	SME Add Vector to Array
10	0xx010xxxxxxxxxx	x1xx	UNALLOCATED
10	0xx1xxxxxxxxxxxxx		UNALLOCATED
10	00x011xxxxxxxxxx		UNALLOCATED
10	0000010xxxxxxxxx		SME Zero
10	0000011xxxxxxxxx		SME2 Multiple Zero
10	0010010xxxxxxxxx		SME2 Zero Lookup Table
10	0010011xxxxxxxxx		SME2 Move Lookup Table
10	01x001xxxxxxxxxx		SME2 Expand Lookup Table (Contiguous)
10	010011xxxxxxxxxx		SME2 Expand Lookup Table (Non-contiguous)
10	011011xxxxxxxxxx		UNALLOCATED
10	1xx00xxxxxxxxxxx		SME2 Multi-vector - Indexed (One register)
10	1xx01xxxx0xxxxxx		SME2 Multi-vector - Indexed (Two registers)
10	1xx01xxxx1xxxxxx		SME2 Multi-vector - Indexed (Four registers)
10	1xx1xxxxx100xxx		SME2 Multi-vector - SVE Select
10	1xx1xxxxx110xxx		SME2 Multi-vector - SVE Constructive Binary
10	1xx1xxxxx111000		SME2 Multi-vector - SVE Constructive Unary
10	1xx1xxxxx111001		UNALLOCATED
10	1xx1xxxxx11101*		UNALLOCATED
10	1xx1xxxxx010110*		SME2 Multi-vector - Multiple Vectors SVE Destructive (Two registers)

10	1xx1xxxx010111*x xxx0	SME2 Multi-vector - Multiple Vectors SVE Destructive (Four registers)
10	1xx10xxxx10100*x	SME2 Multi-vector - Multiple and Single SVE Destructive (Two registers)
10	1xx10xxxx10101*x xxx0	SME2 Multi-vector - Multiple and Single SVE Destructive (Four registers)
10	1xx10xxx0101x1*x xxx1	UNALLOCATED
10	1xx10xxx01111xx	UNALLOCATED
10	1xx10xxx11x11xx	UNALLOCATED
10	1xx10xxx110101*x xxx1	UNALLOCATED
10	1xx11xxxx1111xx	UNALLOCATED
10	1xx11xxx01010xx	UNALLOCATED
10	1xx11xxx010111*x xxx1	UNALLOCATED
10	1xx11xxx1101xxx	UNALLOCATED
10	10x1xxxxx0xxxxxx	SME2 Multi-vector - Multiple and Single Array Vectors
10	11x1xxxx00xxxxxx	SME2 Multi-vector - Multiple Array Vectors (Two registers)
10	11x1xxxx10xxxxxx	SME2 Multi-vector - Multiple Array Vectors (Four registers)
11		SME Memory

SME Outer Product - 64 bit

These instructions are under [SME encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
10	op0	0000	op1	11	op2																									0	

Decode fields			Instruction details
op0	op1	op2	
0	0	0	SME FP64 outer product
0	0	1	UNALLOCATED
0	1		UNALLOCATED
1			SME Int16 outer product

SME FP64 outer product

These instructions are under [SME Outer Product - 64 bit](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	0	0	0	0	0	1	1	0	Zm	Pm	Pn										Zn	S	0	ZAda						

Decode fields	Instruction Details	Feature
S		
0	FMOPA (non-widening)	FEAT_SME_F64F64
1	FMOPS (non-widening)	FEAT_SME_F64F64

SME Int16 outer product

These instructions are under [SME Outer Product - 64 bit](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
1	0	1	0	0	0	0	u0	1	1	u1	Zm	Pm	Pn	Zn	S	0	ZAda																	

Decode fields			Instruction Details	Feature
u0	u1	S		
0	0	0	SMOPA (4-way)	FEAT_SME_I16I64
0	0	1	SMOPS (4-way)	FEAT_SME_I16I64
0	1	0	SUMOPA	FEAT_SME_I16I64
0	1	1	SUMOPS	FEAT_SME_I16I64
1	0	0	USMOPA	FEAT_SME_I16I64
1	0	1	USMOPS	FEAT_SME_I16I64
1	1	0	UMOPA (4-way)	FEAT_SME_I16I64
1	1	1	UMOPS (4-way)	FEAT_SME_I16I64

SME FP Outer Product - 32 bit

These instructions are under [SME encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1000000	op0	010	op1																											00	

Decode fields		Instruction details
op0	op1	
0	0	SME FP32 outer product
0	1	UNALLOCATED
1	0	SME BF16 widening outer product
1	1	SME FP16 widening outer product

SME FP32 outer product

These instructions are under [SME FP Outer Product - 32 bit](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
1	0	0	0	0	0	0	1	0	0	Zm	Pm	Pn	Zn	S	0	0	ZAda																

Decode fields
S

Instruction Details

Feature

0	FMOPA (non-widening)	FEAT_SME
1	FMOPS (non-widening)	FEAT_SME

SME BF16 widening outer product

These instructions are under [SME FP Outer Product - 32 bit](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
1	0	0	0	0	0	0	1	1	0	0	Zm	Pm	Pn	Zn	S	0	0	ZAda																

Decode fields
S

Instruction Details

Feature

0	BFMOPA (widening)	FEAT_SME
1	BFMOPS (widening)	FEAT_SME

SME FP16 widening outer product

These instructions are under [SME FP Outer Product - 32 bit](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
1	0	0	0	0	0	0	1	1	0	1	Zm	Pm	Pn	Zn	S	0	0	ZAda																

Decode fields
S

Instruction Details

Feature

0	FMOPA (widening)	FEAT_SME
1	FMOPS (widening)	FEAT_SME

SME2 Outer Product - Misc

These instructions are under [SME encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1000000	op0	10	op1																									10	op2		

Decode fields
op0 op1 op2

Instruction details

0	0		SME2 32-bit binary outer product
0	1		UNALLOCATED
1	0	0	SME2 FP16 non-widening outer product

1	1	0	SME2 BF16 non-widening outer product
1		1	UNALLOCATED

SME2 32-bit binary outer product

These instructions are under [SME2 Outer Product - Misc.](#)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
1	0	0	0	0	0	0	1	0	0	Zm	Pm	Pn	Zn	S	1	0	ZAda																	

Decode fields S	Instruction Details	Feature
0	BMOPA	FEAT_SME2
1	BMOPS	FEAT_SME2

SME2 FP16 non-widening outer product

These instructions are under [SME2 Outer Product - Misc.](#)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	0	0	0	0	0	1	1	0	0	Zm	Pm	Pn	Zn	S	1	0	0	ZAda												

Decode fields S	Instruction Details	Feature
0	FMOPA (non-widening)	FEAT_SME_F16F16
1	FMOPS (non-widening)	FEAT_SME_F16F16

SME2 BF16 non-widening outer product

These instructions are under [SME2 Outer Product - Misc.](#)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	0	0	0	0	0	1	1	0	1	Zm	Pm	Pn	Zn	S	1	0	0	ZAda												

Decode fields S	Instruction Details	Feature
0	BFMOPA (non-widening)	FEAT_SVE_B16B16
1	BFMOPS (non-widening)	FEAT_SVE_B16B16

SME Integer Outer Product - 32 bit

These instructions are under [SME encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1010000							10	op0																	op10						

Decode fields
op0 **op1**

Instruction details

0	1	SME2 Int16 two-way outer product
1	1	UNALLOCATED
	0	SME Int8 outer product

SME2 Int16 two-way outer product

These instructions are under [SME Integer Outer Product - 32 bit](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	1	0	0	0	0	u0	1	0	0		Zm		Pm		Pn		Zn		S	1	0	ZAda								

Decode fields
u0 **S**

Instruction Details

Feature

0	0	SMOPA (2-way)	FEAT_SME2
0	1	SMOPS (2-way)	FEAT_SME2
1	0	UMOPA (2-way)	FEAT_SME2
1	1	UMOPS (2-way)	FEAT_SME2

SME Int8 outer product

These instructions are under [SME Integer Outer Product - 32 bit](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	1	0	0	0	0	u0	1	0	u1		Zm		Pm		Pn		Zn		S	0	0	ZAda								

Decode fields
u0 **u1** **S**

Instruction Details

Feature

0	0	0	SMOPA (4-way)	FEAT_SME
0	0	1	SMOPS (4-way)	FEAT_SME
0	1	0	SUMOPA	FEAT_SME
0	1	1	SUMOPS	FEAT_SME
1	0	0	USMOPA	FEAT_SME
1	0	1	USMOPS	FEAT_SME
1	1	0	UMOPA (4-way)	FEAT_SME
1	1	1	UMOPS (4-way)	FEAT_SME

SME2 Multi-vector - Memory (Contiguous)

These instructions are under [SME encodings](#).

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	
101000000	op0

Decode fields			Instruction details
op0	op1	op2	
00x	0		SME2 multi-vec contiguous load (scalar plus scalar, two registers)
00x	1	0	SME2 multi-vec contiguous load (scalar plus scalar, four registers)
01x	0		SME2 multi-vec contiguous store (scalar plus scalar, two registers)
01x	1	0	SME2 multi-vec contiguous store (scalar plus scalar, four registers)
0xx	1	1	UNALLOCATED
100	0		SME2 multi-vec contiguous load (scalar plus immediate, two registers)
100	1	0	SME2 multi-vec contiguous load (scalar plus immediate, four registers)
110	0		SME2 multi-vec contiguous store (scalar plus immediate, two registers)
110	1	0	SME2 multi-vec contiguous store (scalar plus immediate, four registers)
1x0	1	1	UNALLOCATED
1x1			UNALLOCATED

SME2 multi-vec contiguous load (scalar plus scalar, two registers)

These instructions are under [SME2 Multi-vector - Memory \(Contiguous\)](#).

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	
1 0 1 0 0 0 0 0 0 0 Rm 0 msz PNg Rn Zt N	

Decode fields		Instruction Details	Feature
msz	N		
00	0	LD1B (scalar plus scalar, consecutive registers)	FEAT_SVE2p1
00	1	LDNT1B (scalar plus scalar, consecutive registers)	FEAT_SVE2p1
01	0	LD1H (scalar plus scalar, consecutive registers)	FEAT_SVE2p1
01	1	LDNT1H (scalar plus scalar, consecutive registers)	FEAT_SVE2p1

Decode fields		Instruction Details	Feature
msz	N		
10	0	LD1W (scalar plus scalar, consecutive registers)	FEAT_SVE2p1
10	1	LDNT1W (scalar plus scalar, consecutive registers)	FEAT_SVE2p1
11	0	LD1D (scalar plus scalar, consecutive registers)	FEAT_SVE2p1
11	1	LDNT1D (scalar plus scalar, consecutive registers)	FEAT_SVE2p1

SME2 multi-vec contiguous load (scalar plus scalar, four registers)

These instructions are under [SME2 Multi-vector - Memory \(Contiguous\)](#).

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
| 1 0 1 0 0 0 0 0 0 0 | Rm | 1 | msz | Png | Rn | Zt | 0 | N |

Decode fields		Instruction Details	Feature
msz	N		
00	0	LD1B (scalar plus scalar, consecutive registers)	FEAT_SVE2p1
00	1	LDNT1B (scalar plus scalar, consecutive registers)	FEAT_SVE2p1
01	0	LD1H (scalar plus scalar, consecutive registers)	FEAT_SVE2p1
01	1	LDNT1H (scalar plus scalar, consecutive registers)	FEAT_SVE2p1
10	0	LD1W (scalar plus scalar, consecutive registers)	FEAT_SVE2p1
10	1	LDNT1W (scalar plus scalar, consecutive registers)	FEAT_SVE2p1
11	0	LD1D (scalar plus scalar, consecutive registers)	FEAT_SVE2p1
11	1	LDNT1D (scalar plus scalar, consecutive registers)	FEAT_SVE2p1

SME2 multi-vec contiguous store (scalar plus scalar, two registers)

These instructions are under [SME2 Multi-vector - Memory \(Contiguous\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	1	0	0	0	0	0	0	0	1	Rm	0	msz	PNg							Rn		Zt	N							

Decode fields		Instruction Details	Feature
msz	N		
00	0	ST1B (scalar plus scalar, consecutive registers)	FEAT_SVE2p1
00	1	STNT1B (scalar plus scalar, consecutive registers)	FEAT_SVE2p1
01	0	ST1H (scalar plus scalar, consecutive registers)	FEAT_SVE2p1
01	1	STNT1H (scalar plus scalar, consecutive registers)	FEAT_SVE2p1
10	0	ST1W (scalar plus scalar, consecutive registers)	FEAT_SVE2p1
10	1	STNT1W (scalar plus scalar, consecutive registers)	FEAT_SVE2p1
11	0	ST1D (scalar plus scalar, consecutive registers)	FEAT_SVE2p1
11	1	STNT1D (scalar plus scalar, consecutive registers)	FEAT_SVE2p1

SME2 multi-vec contiguous store (scalar plus scalar, four registers)

These instructions are under [SME2 Multi-vector - Memory \(Contiguous\)](#).

Decode fields		Instruction Details	Feature
msz	N		
00	0	ST1B (scalar plus scalar, consecutive registers)	FEAT_SVE2p1
00	1	STNT1B (scalar plus scalar, consecutive registers)	FEAT_SVE2p1
01	0	ST1H (scalar plus scalar, consecutive registers)	FEAT_SVE2p1
01	1	STNT1H (scalar plus scalar, consecutive registers)	FEAT_SVE2p1
10	0	ST1W (scalar plus scalar, consecutive registers)	FEAT_SVE2p1
10	1	STNT1W (scalar plus scalar, consecutive registers)	FEAT_SVE2p1
11	0	ST1D (scalar plus scalar, consecutive registers)	FEAT_SVE2p1
11	1	STNT1D (scalar plus scalar, consecutive registers)	FEAT_SVE2p1

SME2 multi-vec contiguous load (scalar plus immediate, two registers)

These instructions are under [SME2 Multi-vector - Memory \(Contiguous\)](#).

Decode fields		Instruction Details	Feature
msz	N		
00	0	LD1B (scalar plus immediate, consecutive registers)	FEAT_SVE2p1
00	1	LDNT1B (scalar plus immediate, consecutive registers)	FEAT_SVE2p1
01	0	LD1H (scalar plus immediate, consecutive registers)	FEAT_SVE2p1
01	1	LDNT1H (scalar plus immediate, consecutive registers)	FEAT_SVE2p1
10	0	LD1W (scalar plus immediate, consecutive registers)	FEAT_SVE2p1
10	1	LDNT1W (scalar plus immediate, consecutive registers)	FEAT_SVE2p1
11	0	LD1D (scalar plus immediate, consecutive registers)	FEAT_SVE2p1
11	1	LDNT1D (scalar plus immediate, consecutive registers)	FEAT_SVE2p1

SME2 multi-vec contiguous load (scalar plus immediate, four registers)

These instructions are under [SME2 Multi-vector - Memory \(Contiguous\)](#).

Decode fields		Instruction Details	Feature
msz	N		
00	0	LD1B (scalar plus immediate, consecutive registers)	FEAT_SVE2p1
00	1	LDNT1B (scalar plus immediate, consecutive registers)	FEAT_SVE2p1
01	0	LD1H (scalar plus immediate, consecutive registers)	FEAT_SVE2p1
01	1	LDNT1H (scalar plus immediate, consecutive registers)	FEAT_SVE2p1
10	0	LD1W (scalar plus immediate, consecutive registers)	FEAT_SVE2p1
10	1	LDNT1W (scalar plus immediate, consecutive registers)	FEAT_SVE2p1
11	0	LD1D (scalar plus immediate, consecutive registers)	FEAT_SVE2p1
11	1	LDNT1D (scalar plus immediate, consecutive registers)	FEAT_SVE2p1

SME2 multi-vec contiguous store (scalar plus immediate, two registers)

These instructions are under [SME2 Multi-vector - Memory \(Contiguous\)](#).

Decode fields		Instruction Details	Feature
msz	N		
00	0	ST1B (scalar plus immediate, consecutive registers)	FEAT_SVE2p1
00	1	STNT1B (scalar plus immediate, consecutive registers)	FEAT_SVE2p1
01	0	ST1H (scalar plus immediate, consecutive registers)	FEAT_SVE2p1
01	1	STNT1H (scalar plus immediate, consecutive registers)	FEAT_SVE2p1
10	0	ST1W (scalar plus immediate, consecutive registers)	FEAT_SVE2p1
10	1	STNT1W (scalar plus immediate, consecutive registers)	FEAT_SVE2p1
11	0	ST1D (scalar plus immediate, consecutive registers)	FEAT_SVE2p1
11	1	STNT1D (scalar plus immediate, consecutive registers)	FEAT_SVE2p1

SME2 multi-vec contiguous store (scalar plus immediate, four registers)

These instructions are under [SME2 Multi-vector - Memory \(Contiguous\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	1	0	1	0	0	0	0	0	0	1	1	0	imm4	1	msz	PNG		Rn	Zt	O	N
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	------	---	-----	-----	--	----	----	---	---

Decode fields		Instruction Details		Feature
msz	N			
00	0	ST1B (scalar plus immediate, consecutive registers)		FEAT_SVE2p1
00	1	STNT1B (scalar plus immediate, consecutive registers)		FEAT_SVE2p1
01	0	ST1H (scalar plus immediate, consecutive registers)		FEAT_SVE2p1
01	1	STNT1H (scalar plus immediate, consecutive registers)		FEAT_SVE2p1
10	0	ST1W (scalar plus immediate, consecutive registers)		FEAT_SVE2p1
10	1	STNT1W (scalar plus immediate, consecutive registers)		FEAT_SVE2p1
11	0	ST1D (scalar plus immediate, consecutive registers)		FEAT_SVE2p1
11	1	STNT1D (scalar plus immediate, consecutive registers)		FEAT_SVE2p1

SME2 Multi-vector - Memory (Strided)

These instructions are under [SME encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Decode fields			Instruction details
op0	op1	op2	
00x	0		SME2 multi-vec non-contiguous load (scalar plus scalar, two registers)
00x	1	0	SME2 multi-vec non-contiguous load (scalar plus scalar, four registers)
01x	0		SME2 multi-vec non-contiguous store (scalar plus scalar, two registers)
01x	1	0	SME2 multi-vec non-contiguous store (scalar plus scalar, four registers)
0xx	1	1	UNALLOCATED
100	0		SME2 multi-vec non-contiguous load (scalar plus immediate, two registers)
100	1	0	SME2 multi-vec non-contiguous load (scalar plus immediate, four registers)
110	0		SME2 multi-vec non-contiguous store (scalar plus immediate, two registers)
110	1	0	SME2 multi-vec non-contiguous store (scalar plus immediate, four registers)

1x0	1	1	UNALLOCATED
1x1			UNALLOCATED

SME2 multi-vec non-contiguous load (scalar plus scalar, two registers)

These instructions are under [SME2 Multi-vector - Memory \(Strided\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	1	0	0	0	0	1	0	0	0	Rm	0	msz	PNg		Rn	Zth	N		Ztl											

Decode fields		Instruction Details			Feature
msz	N				
00	0	LD1B (scalar plus scalar, strided registers)			FEAT_SME2
00	1	LDNT1B (scalar plus scalar, strided registers)			FEAT_SME2
01	0	LD1H (scalar plus scalar, strided registers)			FEAT_SME2
01	1	LDNT1H (scalar plus scalar, strided registers)			FEAT_SME2
10	0	LD1W (scalar plus scalar, strided registers)			FEAT_SME2
10	1	LDNT1W (scalar plus scalar, strided registers)			FEAT_SME2
11	0	LD1D (scalar plus scalar, strided registers)			FEAT_SME2
11	1	LDNT1D (scalar plus scalar, strided registers)			FEAT_SME2

SME2 multi-vec non-contiguous load (scalar plus scalar, four registers)

These instructions are under [SME2 Multi-vector - Memory \(Strided\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	1	0	0	0	0	1	0	0	0	Rm	1	msz	PNg		Rn	Zth	N	0	Ztl											

Decode fields		Instruction Details			Feature
msz	N				
00	0	LD1B (scalar plus scalar, strided registers)			FEAT_SME2
00	1	LDNT1B (scalar plus scalar, strided registers)			FEAT_SME2
01	0	LD1H (scalar plus scalar, strided registers)			FEAT_SME2

Decode fields		Instruction Details		Feature
msz	N			
01	1	LDNT1H (scalar plus scalar, strided registers)		FEAT_SME2
10	0	LD1W (scalar plus scalar, strided registers)		FEAT_SME2
10	1	LDNT1W (scalar plus scalar, strided registers)		FEAT_SME2
11	0	LD1D (scalar plus scalar, strided registers)		FEAT_SME2
11	1	LDNT1D (scalar plus scalar, strided registers)		FEAT_SME2

SME2 multi-vec non-contiguous store (scalar plus scalar, two registers)

These instructions are under [SME2 Multi-vector - Memory \(Strided\)](#).

Decode fields		Instruction Details	Feature
msz	N		
00	0	ST1B (scalar plus scalar, strided registers)	FEAT_SME2
00	1	STNT1B (scalar plus scalar, strided registers)	FEAT_SME2
01	0	ST1H (scalar plus scalar, strided registers)	FEAT_SME2
01	1	STNT1H (scalar plus scalar, strided registers)	FEAT_SME2
10	0	ST1W (scalar plus scalar, strided registers)	FEAT_SME2
10	1	STNT1W (scalar plus scalar, strided registers)	FEAT_SME2
11	0	ST1D (scalar plus scalar, strided registers)	FEAT_SME2
11	1	STNT1D (scalar plus scalar, strided registers)	FEAT_SME2

SME2 multi-vec non-contiguous store (scalar plus scalar, four registers)

These instructions are under [SME2 Multi-vector - Memory \(Strided\)](#).

Decode fields		Instruction Details		Feature
msz	N			
00	0	ST1B (scalar plus scalar, strided registers)		FEAT_SME2
00	1	STNT1B (scalar plus scalar, strided registers)		FEAT_SME2
01	0	ST1H (scalar plus scalar, strided registers)		FEAT_SME2
01	1	STNT1H (scalar plus scalar, strided registers)		FEAT_SME2
10	0	ST1W (scalar plus scalar, strided registers)		FEAT_SME2
10	1	STNT1W (scalar plus scalar, strided registers)		FEAT_SME2
11	0	ST1D (scalar plus scalar, strided registers)		FEAT_SME2
11	1	STNT1D (scalar plus scalar, strided registers)		FEAT_SME2

SME2 multi-vec non-contiguous load (scalar plus immediate, two registers)

These instructions are under [SME2 Multi-vector - Memory \(Strided\)](#).

Decode fields		Instruction Details	Feature
msz	N		
00	0	LD1B (scalar plus immediate, strided registers)	FEAT_SME2
00	1	LDNT1B (scalar plus immediate, strided registers)	FEAT_SME2
01	0	LD1H (scalar plus immediate, strided registers)	FEAT_SME2
01	1	LDNT1H (scalar plus immediate, strided registers)	FEAT_SME2
10	0	LD1W (scalar plus immediate, strided registers)	FEAT_SME2
10	1	LDNT1W (scalar plus immediate, strided registers)	FEAT_SME2
11	0	LD1D (scalar plus immediate, strided registers)	FEAT_SME2
11	1	LDNT1D (scalar plus immediate, strided registers)	FEAT_SME2

SME2 multi-vec non-contiguous load (scalar plus immediate, four registers)

These instructions are under [SME2 Multi-vector - Memory \(Strided\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	1	0	1	0	0	0	0	1	0	1	0	0	imm4	1	msz	PNq		Rn	Zth	N	0	Ztl
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	------	---	-----	-----	--	----	-----	---	---	-----

Decode fields		Instruction Details	Feature
msz	N		
00	0	LD1B (scalar plus immediate, strided registers)	FEAT_SME2
00	1	LDNT1B (scalar plus immediate, strided registers)	FEAT_SME2
01	0	LD1H (scalar plus immediate, strided registers)	FEAT_SME2
01	1	LDNT1H (scalar plus immediate, strided registers)	FEAT_SME2
10	0	LD1W (scalar plus immediate, strided registers)	FEAT_SME2
10	1	LDNT1W (scalar plus immediate, strided registers)	FEAT_SME2
11	0	LD1D (scalar plus immediate, strided registers)	FEAT_SME2
11	1	LDNT1D (scalar plus immediate, strided registers)	FEAT_SME2

SME2 multi-vec non-contiguous store (scalar plus immediate, two registers)

These instructions are under [SME2 Multi-vector - Memory \(Strided\)](#).

Decode fields		Instruction Details	Feature
msz	N		
00	0	ST1B (scalar plus immediate, strided registers)	FEAT_SME2
00	1	STNT1B (scalar plus immediate, strided registers)	FEAT_SME2
01	0	ST1H (scalar plus immediate, strided registers)	FEAT_SME2
01	1	STNT1H (scalar plus immediate, strided registers)	FEAT_SME2
10	0	ST1W (scalar plus immediate, strided registers)	FEAT_SME2
10	1	STNT1W (scalar plus immediate, strided registers)	FEAT_SME2
11	0	ST1D (scalar plus immediate, strided registers)	FEAT_SME2
11	1	STNT1D (scalar plus immediate, strided registers)	FEAT_SME2

SME2 multi-vec non-contiguous store (scalar plus immediate, four registers)

These instructions are under [SME2 Multi-vector - Memory \(Strided\)](#).

Decode fields		Instruction Details						Feature
msz	N							
00	0	ST1B (scalar plus immediate, strided registers)						FEAT_SME2
00	1	STNT1B (scalar plus immediate, strided registers)						FEAT_SME2
01	0	ST1H (scalar plus immediate, strided registers)						FEAT_SME2
01	1	STNT1H (scalar plus immediate, strided registers)						FEAT_SME2
10	0	ST1W (scalar plus immediate, strided registers)						FEAT_SME2
10	1	STNT1W (scalar plus immediate, strided registers)						FEAT_SME2
11	0	ST1D (scalar plus immediate, strided registers)						FEAT_SME2
11	1	STNT1D (scalar plus immediate, strided registers)						FEAT_SME2

SME Move into Array

These instructions are under [SME encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Decode fields						Instruction details			Feature
op0	op1	op2	op3	op4	op5				
00	1	00	010	x0	0	MOVA (vector to array, two registers)			FEAT_SME2
00	1	00	011	00	0	MOVA (vector to array, four registers)			FEAT_SME2
00	1	00	0x0	x0	1	UNALLOCATED			-
00	1	00	0x0	x1		UNALLOCATED			-
00	1	00	0x1	00	1	UNALLOCATED			-
00	1	00	0x1	!= 00		UNALLOCATED			-
00	1	01	000	x0	1	UNALLOCATED			-
00	1	01	000	x1		UNALLOCATED			-
00	1	01	001	00	1	UNALLOCATED			-
00	1	01	001	!= 00		UNALLOCATED			-
00	1	01	01x			UNALLOCATED			-
!= 00	1	0x	000	x0	1	UNALLOCATED			-

!= 00	1	0x	000	x1		UNALLOCATED	-
!= 00	1	0x	001	00	1	UNALLOCATED	-
!= 00	1	0x	001	01		UNALLOCATED	-
!= 00	1	0x	001	1x		UNALLOCATED	-
!= 00	1	0x	01x			UNALLOCATED	-
	0					SME move vector to array	-
	1	0x	000	x0	0	SME2 move vector to tile, two registers	-
	1	0x	001	00	0	SME2 move vector to tile, four registers	-
	1	0x	1xx			UNALLOCATED	-
	1	1x				UNALLOCATED	-

SME move vector to array

These instructions are under [SME Move into Array](#).

Decode fields		Instruction Details	Feature
size	Q		
0x	1	UNALLOCATED	-
00	0	MOVA (vector to tile, single) " 8-bit	FEAT_SME
01	0	MOVA (vector to tile, single) " 16-bit	FEAT_SME
10	0	MOVA (vector to tile, single) " 32-bit	FEAT_SME
10	1	UNALLOCATED	-
11	0	MOVA (vector to tile, single) " 64-bit	FEAT_SME
11	1	MOVA (vector to tile, single) " 128-bit	FEAT_SME

SME2 move vector to tile, two registers

These instructions are under [SME Move into Array](#).

Decode fields size	Instruction Details	Feature
00	MOVA (vector to tile, two registers) â€” 8-bit	FEAT_SME2
01	MOVA (vector to tile, two registers) â€” 16-bit	FEAT_SME2
10	MOVA (vector to tile, two registers) â€” 32-bit	FEAT_SME2
11	MOVA (vector to tile, two registers) â€” 64-bit	FEAT_SME2

SME2 move vector to tile, four registers

These instructions are under [SME Move into Array](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	0	size	0	0	0	1	0	0	V	Rs	0	0	1	Zn	0	0	0	0	opc						

Decode fields size	Instruction Details	Feature
opc		
0x	UNALLOCATED	-
00	MOVA (vector to tile, four registers) â€” 8-bit	FEAT_SME2
01	MOVA (vector to tile, four registers) â€” 16-bit	FEAT_SME2
10	MOVA (vector to tile, four registers) â€” 32-bit	FEAT_SME2
10	UNALLOCATED	-
11	MOVA (vector to tile, four registers) â€” 64-bit	FEAT_SME2

SME Move from Array

These instructions are under [SME encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
11000000	op0	000	op1	1	op2			op3	op4																		op5				

op0	op1	op2	op3	op4	op5	Decode fields	Instruction details	Feature
00	1	00	010	00	x0		MOVA (array to vector, two registers)	FEAT_SME2

00	1	00	010	10	x0	MOVAZ (array to vector, two registers)	FEAT_SME2p1
00	1	00	011	00	00	MOVA (array to vector, four registers)	FEAT_SME2
00	1	00	011	10	00	MOVAZ (array to vector, four registers)	FEAT_SME2p1
00	1	00	0x0	x0	x1	UNALLOCATED	-
00	1	00	0x1	x0	!= 00	UNALLOCATED	-
00	1	00	0xx	x1		UNALLOCATED	-
00	1	01	000	x0	x1	UNALLOCATED	-
00	1	01	001	x0	!= 00	UNALLOCATED	-
00	1	01	00x	x1		UNALLOCATED	-
00	1	01	01x			UNALLOCATED	-
!= 00	1	0x	000	x0	x1	UNALLOCATED	-
!= 00	1	0x	001	x0	01	UNALLOCATED	-
!= 00	1	0x	001	x0	1x	UNALLOCATED	-
!= 00	1	0x	00x	x1		UNALLOCATED	-
!= 00	1	0x	01x			UNALLOCATED	-
	0		000	1x		SME zeroing move array to vector	-
	0		!= 000	1x		UNALLOCATED	-
	0			0x		SME move array to vector	-
	1	0x	000	00	x0	SME2 move tile to vector, two registers	-
	1	0x	000	10	x0	SME2 zeroing move tile to vector, two registers	-
	1	0x	001	00	00	SME2 move tile to vector, four registers	-
	1	0x	001	10	00	SME2 zeroing move tile to vector, four registers	-
	1	0x	1xx			UNALLOCATED	-
	1	1x				UNALLOCATED	-

SME zeroing move array to vector

These instructions are under [SME Move from Array](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	0	size	0	0	0	0	1	Q	V	Rs	0	0	0	1	opc		Zd								

Decode fields		Instruction Details			Feature
size	Q				
0x	1	UNALLOCATED			-
00	0	MOVAZ (tile to vector, single) â€” 8-bit			FEAT_SME2p1
01	0	MOVAZ (tile to vector, single) â€” 16-bit			FEAT_SME2p1
10	0	MOVAZ (tile to vector, single) â€” 32-bit			FEAT_SME2p1
10	1	UNALLOCATED			-
11	0	MOVAZ (tile to vector, single) â€” 64-bit			FEAT_SME2p1
11	1	MOVAZ (tile to vector, single) â€” 128-bit			FEAT_SME2p1

SME move array to vector

These instructions are under [SME Move from Array](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	0	size	0	0	0	0	1	Q	V	Rs	Pg	0	opc		Zd										

Decode fields		Instruction Details			Feature
size	Q				
0x	1	UNALLOCATED			-
00	0	MOVA (tile to vector, single) â€” 8-bit			FEAT_SME
01	0	MOVA (tile to vector, single) â€” 16-bit			FEAT_SME
10	0	MOVA (tile to vector, single) â€” 32-bit			FEAT_SME
10	1	UNALLOCATED			-
11	0	MOVA (tile to vector, single) â€” 64-bit			FEAT_SME
11	1	MOVA (tile to vector, single) â€” 128-bit			FEAT_SME

SME2 move tile to vector, two registers

These instructions are under [SME Move from Array](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	0	size	0	0	0	1	1	0	V	Rs	0	0	0	0	0	opc	Zd	0							

Decode fields size	Instruction Details	Feature
00	MOVA (tile to vector, two registers) â€” 8-bit	FEAT_SME2
01	MOVA (tile to vector, two registers) â€” 16-bit	FEAT_SME2
10	MOVA (tile to vector, two registers) â€” 32-bit	FEAT_SME2
11	MOVA (tile to vector, two registers) â€” 64-bit	FEAT_SME2

SME2 zeroing move tile to vector, two registers

These instructions are under [SME Move from Array](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	0	size	0	0	0	1	1	0	V	Rs	0	0	0	1	0	opc	Zd	0							

Decode fields size	Instruction Details	Feature
00	MOVAZ (tile to vector, two registers) â€” 8-bit	FEAT_SME2p1
01	MOVAZ (tile to vector, two registers) â€” 16-bit	FEAT_SME2p1
10	MOVAZ (tile to vector, two registers) â€” 32-bit	FEAT_SME2p1
11	MOVAZ (tile to vector, two registers) â€” 64-bit	FEAT_SME2p1

SME2 move tile to vector, four registers

These instructions are under [SME Move from Array](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	0	size	0	0	0	1	1	0	V	Rs	0	0	1	0	0	opc	Zd	0	0						

Decode fields		Instruction Details		Feature
size	opc			
0x	1xx	UNALLOCATED		-
00	0xx	MOVA (tile to vector, four registers) â€” 8-bit		FEAT_SME2
01	0xx	MOVA (tile to vector, four registers) â€” 16-bit		FEAT_SME2
10	0xx	MOVA (tile to vector, four registers) â€” 32-bit		FEAT_SME2
10	1xx	UNALLOCATED		-
11		MOVA (tile to vector, four registers) â€” 64-bit		FEAT_SME2

SME2 zeroing move tile to vector, four registers

These instructions are under [SME Move from Array](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	0	size	0	0	0	1	1	0	V	Rs	0	0	1	1	0	opc	Zd	0	0						

Decode fields		Instruction Details		Feature
size	opc			
0x	1xx	UNALLOCATED		-
00	0xx	MOVAZ (tile to vector, four registers) â€” 8-bit		FEAT_SME2p1
01	0xx	MOVAZ (tile to vector, four registers) â€” 16-bit		FEAT_SME2p1
10	0xx	MOVAZ (tile to vector, four registers) â€” 32-bit		FEAT_SME2p1
10	1xx	UNALLOCATED		-
11		MOVAZ (tile to vector, four registers) â€” 64-bit		FEAT_SME2p1

SME Add Vector to Array

These instructions are under [SME encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
11000000	op0	010	op1																					op20							

Decode fields			Instruction details	
op0	op1	op2		
0			UNALLOCATED	

1	00	0	SME add vector to array
1	00	1	UNALLOCATED
1	!= 00		UNALLOCATED

SME add vector to array

These instructions are under [SME Add Vector to Array](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	0	1	op	0	1	0	0	0	V	Pm	Pn	Zn	0	0	opc2										

Decode fields			Instruction Details	Feature
op	V	opc2		
0		1xx	UNALLOCATED	-
0	0	0xx	ADDHA â€“ 32-bit	FEAT_SME
0	1	0xx	ADDVA â€“ 32-bit	FEAT_SME
1	0		ADDHA â€“ 64-bit	FEAT_SME_I16I64
1	1		ADDVA â€“ 64-bit	FEAT_SME_I16I64

SME Zero

These instructions are under [SME encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1100000000000010																															

Decode fields		Instruction details	Feature
op0			
0000000000		ZERO (tile)	FEAT_SME
!= 0000000000		UNALLOCATED	-

SME2 Multiple Zero

These instructions are under [SME encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1100000000000011																															

Decode fields		Instruction details
op0		
0000000000		SME multiple vectors zero array
!= 0000000000		UNALLOCATED

SME multiple vectors zero array

These instructions are under [SME2 Multiple Zero](#).

Decode fields		Instruction Details	Feature
opc	opc2		
x1x	1xx	UNALLOCATED	-
000		<u>ZERO (single-vector)</u> â€” <u>two ZA single-vectors</u>	FEAT_SME2p1
001		<u>ZERO (double-vector)</u> â€” <u>one ZA double-vector</u>	FEAT_SME2p1
010	0xx	<u>ZERO (double-vector)</u> â€” <u>two ZA double-vectors</u>	FEAT_SME2p1
011	0xx	<u>ZERO (double-vector)</u> â€” <u>four ZA double-vectors</u>	FEAT_SME2p1
100		<u>ZERO (single-vector)</u> â€” <u>four ZA single-vectors</u>	FEAT_SME2p1
101	0xx	<u>ZERO (quad-vector)</u> â€” <u>one ZA quad-vector</u>	FEAT_SME2p1
101	1xx	UNALLOCATED	-
11x	01x	UNALLOCATED	-
110	00x	<u>ZERO (quad-vector)</u> â€” <u>two ZA quad-vectors</u>	FEAT_SME2p1
111	00x	<u>ZERO (quad-vector)</u> â€” <u>four ZA quad-vectors</u>	FEAT_SME2p1

SME2 Zero Lookup Table

These instructions are under [SME encodings](#).

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	11000000010010	op0	0
---	----------------	-----	---

Decode fields op0	Instruction details
00000000000000	SME2 zero lookup table
!= 00000000000000	UNALLOCATED

SME2 zero lookup table

These instructions are under [SME encodings](#).

Decode fields opc	Instruction Details	Feature
0000	UNALLOCATED	-
0001	ZERO (ZT0)	FEAT_SME2
001x	UNALLOCATED	-
01xx	UNALLOCATED	-
1xxx	UNALLOCATED	-

SME2 zero lookup table

These instructions are under [SME2 Zero Lookup Table](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	opc	

Decode fields opc	Instruction Details	Feature
0000	UNALLOCATED	-
0001	ZERO (ZT0)	FEAT_SME2
001x	UNALLOCATED	-
01xx	UNALLOCATED	-
1xxx	UNALLOCATED	-

SME2 Move Lookup Table

These instructions are under [SME encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																									op0	op1					

Decode fields op0 op1		Instruction details
0	00	SME2 move from lookup table
1	00	SME2 move into lookup table
	!= 00	UNALLOCATED

SME2 move from lookup table

These instructions are under [SME2 Move Lookup Table](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	imm3						opc				Rt				

Decode fields opc	Instruction Details	Feature
000xxxx	UNALLOCATED	-
0010xxx	UNALLOCATED	-
00110xx	UNALLOCATED	-
001110x	UNALLOCATED	-
0011110	UNALLOCATED	-
0011111	MOVT (ZT0 to scalar)	FEAT_SME2
01xxxxx	UNALLOCATED	-
1xxxxxx	UNALLOCATED	-

SME2 move into lookup table

These instructions are under [SME2 Move Lookup Table](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	0	1	0	0	1	1	1	0	0	imm3		opc		Rt											

Decode fields opc	Instruction Details	Feature
000xxxx	UNALLOCATED	-
0010xxx	UNALLOCATED	-
00110xx	UNALLOCATED	-
001110x	UNALLOCATED	-
0011110	UNALLOCATED	-
0011111	MOVT (scalar to ZT0)	FEAT_SME2
01xxxxx	UNALLOCATED	-
1xxxxxx	UNALLOCATED	-

SME2 Expand Lookup Table (Contiguous)

These instructions are under [SME encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
110000001	op0	001																										op2			

Decode fields			Instruction details
op0	op1	op2	
0	00		UNALLOCATED
0	10	00	SME2 lookup table expand four contiguous registers
0	10	!= 00	UNALLOCATED

0	x1	x0	SME2 lookup table expand two contiguous registers
0	x1	x1	UNALLOCATED
1			SME2 lookup table expand one register

SME2 lookup table expand four contiguous registers

These instructions are under [SME2 Expand Lookup Table \(Contiguous\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	0	1	0	0	0	1	opc	1	0	size	opc2	Zn		Zd	0	0									

Decode fields opc opc2		Instruction Details	Feature
00x		UNALLOCATED	-
01x	00	LUTI4 (four registers)	FEAT_SME2
01x	01	UNALLOCATED	-
01x	1x	UNALLOCATED	-
1xx	00	LUTI2 (four registers)	FEAT_SME2
1xx	01	UNALLOCATED	-
1xx	1x	UNALLOCATED	-

SME2 lookup table expand two contiguous registers

These instructions are under [SME2 Expand Lookup Table \(Contiguous\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	0	1	0	0	0	1	opc	1	size	opc2	Zn		Zd	0											

Decode fields opc opc2		Instruction Details	Feature
00xx		UNALLOCATED	-
01xx	00	LUTI4 (two registers)	FEAT_SME2
01xx	01	UNALLOCATED	-
01xx	1x	UNALLOCATED	-
1xxx	00	LUTI2 (two registers)	FEAT_SME2
1xxx	01	UNALLOCATED	-
1xxx	1x	UNALLOCATED	-

SME2 lookup table expand one register

These instructions are under [SME2 Expand Lookup Table \(Contiguous\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	1	0	0	1	opc	size	opc2	Zn		Zd														

Decode fields
opc opc2

Instruction Details

Feature

00xxxx		UNALLOCATED	-
01xxxx	00	LUTI4 (single)	FEAT_SME2
01xxxx	01	UNALLOCATED	-
01xxxx	1x	UNALLOCATED	-
1xxxxx	00	LUTI2 (single)	FEAT_SME2
1xxxxx	01	UNALLOCATED	-
1xxxxx	1x	UNALLOCATED	-

SME2 Expand Lookup Table (Non-contiguous)

These instructions are under [SME encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	0	1	0	0	1	1	opc	1	0	size	opc2	Zn		Zdh	0	0	Zdl								

Decode fields
op0 op1

Instruction details

00		UNALLOCATED
10	00	SME2 lookup table expand four non-contiguous registers
10	01	UNALLOCATED
!= 00	1x	UNALLOCATED
x1	0x	SME2 lookup table expand two non-contiguous registers

SME2 lookup table expand four non-contiguous registers

These instructions are under [SME2 Expand Lookup Table \(Non-contiguous\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	0	1	0	0	1	1	opc	1	0	size	opc2	Zn		Zdh	0	0	Zdl								

Decode fields
opc opc2

Instruction Details

Feature

00x		UNALLOCATED	-
01x	00	LUTI4 (four registers)	FEAT_SME2p1
01x	01	UNALLOCATED	-

Decode fields		Instruction Details		Feature
opc	opc2			
01x	1x	UNALLOCATED	-	
1xx	00	LUTI2 (four registers)	FEAT_SME2p1	
1xx	01	UNALLOCATED	-	
1xx	1x	UNALLOCATED	-	

SME2 lookup table expand two non-contiguous registers

These instructions are under [SME2 Expand Lookup Table \(Non-contiguous\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	0	1	0	0	1	1		opc	1	size	opc2	Zn	Zdh	0	Zdl										

Decode fields		Instruction Details		Feature
opc	opc2			
00xx		UNALLOCATED	-	
01xx	00	LUTI4 (two registers)	FEAT_SME2p1	
01xx	01	UNALLOCATED	-	
01xx	1x	UNALLOCATED	-	
1xxx	00	LUTI2 (two registers)	FEAT_SME2p1	
1xxx	01	UNALLOCATED	-	
1xxx	1x	UNALLOCATED	-	

SME2 Multi-vector - Indexed (One register)

These instructions are under [SME encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					11000001		op0	00											op1									op2			

Decode fields			Instruction details
op0	op1	op2	
00			SME2 multi-vec indexed long long MLA one source 32-bit
01			UNALLOCATED
10	0	0	SME2 multi-vec indexed long long MLA one source 64-bit
10	0	1	UNALLOCATED
10	1		SME2 multi-vec indexed long FMA one source
11	0		UNALLOCATED
11	1		SME2 multi-vec indexed long MLA one source

SME2 multi-vec indexed long long MLA one source 32-bit

These instructions are under [SME2 Multi-vector - Indexed \(One register\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	0	0	0	0	Zm	i4h	Rv	i4l			Zn	U	S	op	off2									

Decode fields			Instruction Details			Feature		
U	S	op						
	1	1	UNALLOCATED			-		
0	0	0	SMLALL (multiple and indexed vector)			FEAT_SME2		
0	0	1	USMLALL (multiple and indexed vector)			FEAT_SME2		
0	1	0	SMLSLL (multiple and indexed vector)			FEAT_SME2		
1	0	0	UMLALL (multiple and indexed vector)			FEAT_SME2		
1	0	1	SUMLALL (multiple and indexed vector)			FEAT_SME2		
1	1	0	UMLSLL (multiple and indexed vector)			FEAT_SME2		

SME2 multi-vec indexed long long MLA one source 64-bit

These instructions are under [SME2 Multi-vector - Indexed \(One register\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	1	0	0	0	Zm	i3h	Rv	0	i3l			Zn	U	S	0	off2								

Decode fields			Instruction Details			Feature		
U	S							
0	0		SMLALL (multiple and indexed vector)			FEAT_SME_I16I64		
0	1		SMLSLL (multiple and indexed vector)			FEAT_SME_I16I64		
1	0		UMLALL (multiple and indexed vector)			FEAT_SME_I16I64		
1	1		UMLSLL (multiple and indexed vector)			FEAT_SME_I16I64		

SME2 multi-vec indexed long FMA one source

These instructions are under [SME2 Multi-vector - Indexed \(One register\)](#).

Decode fields		Instruction Details	Feature
op	s		
0	0	FMLAL (multiple and indexed vector)	FEAT_SME2
0	1	FMLS (multiple and indexed vector)	FEAT_SME2
1	0	BFMLAL (multiple and indexed vector)	FEAT_SME2
1	1	BFMLS (multiple and indexed vector)	FEAT_SME2

SME2 multi-vec indexed long MLA one source

These instructions are under [SME2 Multi-vector - Indexed \(One register\)](#).

Decode fields		Instruction Details	Feature
U	S		
0	0	SMLAL (multiple and indexed vector)	FEAT_SME2
0	1	SMLS L (multiple and indexed vector)	FEAT_SME2
1	0	UMLAL (multiple and indexed vector)	FEAT_SME2
1	1	UMLSL (multiple and indexed vector)	FEAT_SME2

SME2 Multi-vector - Indexed (Two registers)

These instructions are under [SME encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
11000001	op0	01													0	op1										op2					

Decode fields			Instruction details
op0	op1	op2	
00	0x		SME2 multi-vec indexed long long MLA two sources 32-bit
00	1x		SME2 multi-vec ternary indexed two registers 16-bit

01			SME2 multi-vec ternary indexed two registers 32-bit
10	00	0	SME2 multi-vec indexed long long MLA two sources 64-bit
10	1x	0	SME2 multi-vec indexed long FMA two sources
11	00	0	SME2 multi-vec ternary indexed two registers 64-bit
11	1x	0	SME2 multi-vec indexed long MLA two sources
1x	00	1	UNALLOCATED
1x	01		UNALLOCATED
1x	1x	1	UNALLOCATED

SME2 multi-vec indexed long long MLA two sources 32-bit

These instructions are under [SME2 Multi-vector - Indexed \(Two registers\)](#).

Decode fields			Instruction Details	Feature
op	U	S		
0	0	0	SMLALL (multiple and indexed vector)	FEAT_SME2
0	0	1	SMLSLL (multiple and indexed vector)	FEAT_SME2
0	1	0	UMLALL (multiple and indexed vector)	FEAT_SME2
0	1	1	UMLSLL (multiple and indexed vector)	FEAT_SME2
1		1	UNALLOCATED	-
1	0	0	USMLALL (multiple and indexed vector)	FEAT_SME2
1	1	0	SUMLALL (multiple and indexed vector)	FEAT_SME2

SME2 multi-vec ternary indexed two registers 16-bit

These instructions are under [SME2 Multi-vector - Indexed \(Two registers\)](#).

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
| 1 1 0 0 0 0 1 0 0 0 1 | Zm | 0 | Rv | 1 | i3h | Zn | op | S | i3l | off3

Decode fields		Instruction Details	Feature
op	S		
0	0	FMLA (multiple and indexed vector)	FEAT_SME_F16F16
0	1	FMLS (multiple and indexed vector)	FEAT_SME_F16F16
1	0	BFMLA (multiple and indexed vector)	FEAT_SVE_B16B16
1	1	BFMLS (multiple and indexed vector)	FEAT_SVE_B16B16

SME2 multi-vec ternary indexed two registers 32-bit

These instructions are under [SME2 Multi-vector - Indexed \(Two registers\)](#).

Decode fields		Instruction Details	Feature
op	opc2		
0	000	FMLA (multiple and indexed vector)	FEAT_SME2
0	001	FVDOT	FEAT_SME2
0	010	FMLS (multiple and indexed vector)	FEAT_SME2
0	011	BFVTDOT	FEAT_SME2
0	1x1	UNALLOCATED	-
0	100	SVDOT (2-way)	FEAT_SME2
0	110	UVDOT (2-way)	FEAT_SME2
1	000	SDOT (2-way, multiple and indexed vector)	FEAT_SME2
1	001	FDOT (multiple and indexed vector)	FEAT_SME2
1	010	UDOT (2-way, multiple and indexed vector)	FEAT_SME2
1	011	BFDOT (multiple and indexed vector)	FEAT_SME2
1	100	SDOT (4-way, multiple and indexed vector)	FEAT_SME2
1	101	USDOT (multiple and indexed vector)	FEAT_SME2
1	110	UDOT (4-way, multiple and indexed vector)	FEAT_SME2
1	111	SUDOT (multiple and indexed vector)	FEAT_SME2

SME2 multi-vec indexed long long MLA two sources 64-bit

These instructions are under [SME2 Multi-vector - Indexed \(Two registers\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	1	0	0	1	Zm	0	Rv	0	0	i3h	Zn	0	U	S	i3l	o1								

Decode fields		Instruction Details										Feature									
U	S																				
0	0	SMLALL (multiple and indexed vector)										FEAT_SME_I16I64									
0	1	SMLSLL (multiple and indexed vector)										FEAT_SME_I16I64									
1	0	UMLALL (multiple and indexed vector)										FEAT_SME_I16I64									
1	1	UMLSL (multiple and indexed vector)										FEAT_SME_I16I64									

SME2 multi-vec indexed long FMA two sources

These instructions are under [SME2 Multi-vector - Indexed \(Two registers\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	1	0	0	1	Zm	0	Rv	1	i3h	Zn	0	op	S	i3l	off2									

Decode fields		Instruction Details										Feature									
op	S																				
0	0	FMLAL (multiple and indexed vector)										FEAT_SME2									
0	1	FMLSL (multiple and indexed vector)										FEAT_SME2									
1	0	BFMLAL (multiple and indexed vector)										FEAT_SME2									
1	1	BFMLSL (multiple and indexed vector)										FEAT_SME2									

SME2 multi-vec ternary indexed two registers 64-bit

These instructions are under [SME2 Multi-vector - Indexed \(Two registers\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	1	1	0	1	Zm	0	Rv	0	0	i1	Zn	0	opc		off3									

Decode fields	Instruction Details			Feature
opc				
00	FMLA (multiple and indexed vector)			FEAT_SME_F64F64
01	SDOT (4-way, multiple and indexed vector)			FEAT_SME_I16I64
10	FMLS (multiple and indexed vector)			FEAT_SME_F64F64
11	UDOT (4-way, multiple and indexed vector)			FEAT_SME_I16I64

SME2 multi-vec indexed long MLA two sources

These instructions are under [SME2 Multi-vector - Indexed \(Two registers\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
1	1	0	0	0	0	0	1	1	1	0	1	Zm	0	Rv	1	i3h	Zn	0	U	S	i3l	off2										

Decode fields	Instruction Details			Feature
U	S			
0	0	SMLAL (multiple and indexed vector)		
0	1	SMLSL (multiple and indexed vector)		
1	0	UMLAL (multiple and indexed vector)		
1	1	UMLSL (multiple and indexed vector)		

SME2 Multi-vector - Indexed (Four registers)

These instructions are under [SME encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
11000001			op0	01			1		op1		op2																						

Decode fields	Instruction details		
op0	op1	op2	
00	0x	0x	SME2 multi-vec indexed long long MLA four sources 32-bit
00	1x	0x	SME2 multi-vec ternary indexed four registers 16-bit
01		0x	SME2 multi-vec ternary indexed four registers 32-bit
0x		1x	UNALLOCATED

10	00	00	SME2 multi-vec indexed long long MLA four sources 64-bit
10	00	\neq 00	UNALLOCATED
10	01		UNALLOCATED
10	1x	00	SME2 multi-vec indexed long FMA four sources
10	1x	\neq 00	UNALLOCATED
11	0x	00	SME2 multi-vec ternary indexed four registers 64-bit
11	1x	00	SME2 multi-vec indexed long MLA four sources
11		\neq 00	UNALLOCATED

SME2 multi-vec indexed long long MLA four sources 32-bit

These instructions are under [SME2 Multi-vector - Indexed \(Four registers\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	0	0	0	1	Zm	1	Rv	0	i4h	Zn	0	op	U	S	i4l	o1								

Decode fields			Instruction Details			Feature
op	U	S				
0	0	0	SMLALL (multiple and indexed vector)			FEAT_SME2
0	0	1	SMLSLL (multiple and indexed vector)			FEAT_SME2
0	1	0	UMLALL (multiple and indexed vector)			FEAT_SME2
0	1	1	UMLSLL (multiple and indexed vector)			FEAT_SME2
1		1	UNALLOCATED			-
1	0	0	USMLALL (multiple and indexed vector)			FEAT_SME2
1	1	0	SUMLALL (multiple and indexed vector)			FEAT_SME2

SME2 multi-vec ternary indexed four registers 16-bit

These instructions are under [SME2 Multi-vector - Indexed \(Four registers\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	0	0	0	1	Zm	1	Rv	1	i3h	Zn	0	op	S	i3l	off3									

Decode fields		Instruction Details				Feature
op	S					
0	0	FMLA (multiple and indexed vector)				FEAT_SME_F16F16
0	1	FMLS (multiple and indexed vector)				FEAT_SME_F16F16
1	0	BFMLA (multiple and indexed vector)				FEAT_SVE_B16B16
1	1	BFMLS (multiple and indexed vector)				FEAT_SVE_B16B16

SME2 multi-vec ternary indexed four registers 32-bit

These instructions are under [SME2 Multi-vector - Indexed \(Four registers\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	0	1	0	1	Zm	1	Rv	op	i2	Zn	0	opc2	off3											

Decode fields		Instruction Details				Feature
op	opc2					
0	0x1	UNALLOCATED				-
0	000	FMLA (multiple and indexed vector)				FEAT_SME2
0	010	FMLS (multiple and indexed vector)				FEAT_SME2
0	100	SVDOT (4-way)				FEAT_SME2
0	101	USVDOT				FEAT_SME2
0	110	UVDOT (4-way)				FEAT_SME2
0	111	SUVDOT				FEAT_SME2
1	000	SDOT (2-way, multiple and indexed vector)				FEAT_SME2
1	001	FDOT (multiple and indexed vector)				FEAT_SME2
1	010	UDOT (2-way, multiple and indexed vector)				FEAT_SME2
1	011	BFDOT (multiple and indexed vector)				FEAT_SME2
1	100	SDOT (4-way, multiple and indexed vector)				FEAT_SME2
1	101	USDOT (multiple and indexed vector)				FEAT_SME2
1	110	UDOT (4-way, multiple and indexed vector)				FEAT_SME2
1	111	SUDOT (multiple and indexed vector)				FEAT_SME2

SME2 multi-vec indexed long long MLA four sources 64-bit

These instructions are under [SME2 Multi-vector - Indexed \(Four registers\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	1	0	0	1	Zm	1	Rv	0	0	i3h	Zn	0	0	U	S	i3l	o1							

Decode fields		Instruction Details												Feature															
U	S																												
0	0	SMLALL (multiple and indexed vector)												FEAT_SME_I16I64															
0	1	SMLSLL (multiple and indexed vector)												FEAT_SME_I16I64															
1	0	UMLALL (multiple and indexed vector)												FEAT_SME_I16I64															
1	1	UMLSL (multiple and indexed vector)												FEAT_SME_I16I64															

SME2 multi-vec indexed long FMA four sources

These instructions are under [SME2 Multi-vector - Indexed \(Four registers\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	1	0	0	1	Zm	1	Rv	1	i3h	Zn	0	0	op	S	i3l	off2								

Decode fields		Instruction Details												Feature															
op	S																												
0	0	FMLAL (multiple and indexed vector)												FEAT_SME2															
0	1	FMLS (multiple and indexed vector)												FEAT_SME2															
1	0	BFMLAL (multiple and indexed vector)												FEAT_SME2															
1	1	BFMLS (multiple and indexed vector)												FEAT_SME2															

SME2 multi-vec ternary indexed four registers 64-bit

These instructions are under [SME2 Multi-vector - Indexed \(Four registers\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	1	1	0	1	Zm	1	Rv	0	op	i1	Zn	0	0	opc2	off3									

Decode fields		Instruction Details			Feature
op	opc2				
0	00	FMLA (multiple and indexed vector)			FEAT_SME_F64F64
0	01	SDOT (4-way, multiple and indexed vector)			FEAT_SME_I16I64
0	10	FMLS (multiple and indexed vector)			FEAT_SME_F64F64
0	11	UDOT (4-way, multiple and indexed vector)			FEAT_SME_I16I64
1	x0	UNALLOCATED			-
1	01	SVDOT (4-way)			FEAT_SME_I16I64
1	11	UVDOT (4-way)			FEAT_SME_I16I64

SME2 multi-vec indexed long MLA four sources

These instructions are under [SME2 Multi-vector - Indexed \(Four registers\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	1	1	0	1	Zm	1	Rv	1	i3h	Zn	0	0	U	S	i3	off2								

Decode fields		Instruction Details			Feature
U	S				
0	0	SMLAL (multiple and indexed vector)			FEAT_SME2
0	1	SMLS L (multiple and indexed vector)			FEAT_SME2
1	0	UMLAL (multiple and indexed vector)			FEAT_SME2
1	1	UMLSL (multiple and indexed vector)			FEAT_SME2

SME2 Multi-vector - SVE Select

These instructions are under [SME encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
11000001			1		op0	100																	op1		op2						

Decode fields			Instruction details		Feature
op0	op1	op2			
01	00	00	SEL â€” Four registers		FEAT_SME2
01	00	!= 00	UNALLOCATED		-
01	!= 00		UNALLOCATED		-

11			UNALLOCATED	-
x0	x0	x0	SEL “ Two registers	FEAT_SME2
x0	x0	x1	UNALLOCATED	-
x0	x1		UNALLOCATED	-

SME2 Multi-vector - SVE Constructive Binary

These instructions are under [SME encodings](#).

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
11000001 op0 1 110 op1 0 op2

Decode fields			Instruction details
op0	op1	op2	
00	101		SME2 multi-vec quadwords ZIP two registers
01	101		UNALLOCATED
10	101		UNALLOCATED
11	101		SME2 multi-vec saturating shift right narrow two registers
	000		SME2 multi-vec FCLAMP two registers
	001		SME2 multi-vec CLAMP two registers
	010	0	SME2 multi-vec FCLAMP four registers
	011	0	SME2 multi-vec CLAMP four registers
	01x	1	UNALLOCATED
	100		SME2 multi-vec ZIP two registers
	11x		SME2 multi-vec saturating shift right narrow four registers

SME2 multi-vec quadwords ZIP two registers

These instructions are under [SME2 Multi-vector - SVE Constructive Binary](#).

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
1 1 0 0 0 0 0 1 0 0 1 Zm 1 1 0 1 0 1 Zn Zd op

Decode fields	Instruction Details	Feature
op		
0	ZIP (two registers)	FEAT_SME2
1	UZP (two registers)	FEAT_SME2

SME2 multi-vec saturating shift right narrow two registers

These instructions are under [SME2 Multi-vector - SVE Constructive Binary](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	1	1	1	op	imm4	1	1	0	1	0	1	Zn	U	Zd										

Decode fields		Instruction Details				Feature	
op	U						
0	0	SQRSHR (two registers)				FEAT_SME2	
0	1	UQRSHR (two registers)				FEAT_SME2	
1	0	SQRSHRU (two registers)				FEAT_SME2	
1	1	UNALLOCATED				-	

SME2 multi-vec FCLAMP two registers

These instructions are under [SME2 Multi-vector - SVE Constructive Binary](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	size	1	Zm	1	1	0	0	0	0	Zn		Zd	op											

Decode fields		Instruction Details				Feature	
size	op						
	1	UNALLOCATED				-	
!= 00	0	FCLAMP				FEAT_SME2	
00	0	BFCLAMP				FEAT_SVE_B16B16	

SME2 multi-vec CLAMP two registers

These instructions are under [SME2 Multi-vector - SVE Constructive Binary](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	size	1	Zm	1	1	0	0	0	1	Zn		Zd	U											

Decode fields		Instruction Details				Feature	
U							
0		SCLAMP				FEAT_SME2	
1		UCLAMP				FEAT_SME2	

SME2 multi-vec FCLAMP four registers

These instructions are under [SME2 Multi-vector - SVE Constructive Binary](#).

Decode fields		Instruction Details	Feature
size	op		
	1	UNALLOCATED	-
!= 00	0	FCLAMP	FEAT_SME2
00	0	BFCLAMP	FEAT_SVE_B16B16

SME2 multi-vec CLAMP four registers

These instructions are under [SME2 Multi-vector - SVE Constructive Binary](#).

Decode fields	Instruction Details	Feature
U	SCLAMP	FEAT_SME2
1	UCLAMP	FEAT_SME2

SME2 multi-vec ZIP two registers

These instructions are under [SME2 Multi-vector - SVE Constructive Binary](#).

Decode fields op	Instruction Details	Feature
0	ZIP (two registers)	FEAT_SME2
1	UZP (two registers)	FEAT_SME2

SME2 multi-vec saturating shift right narrow four registers

These instructions are under [SME2 Multi-vector - SVE Constructive Binary](#).

Decode fields			Instruction Details	Feature
N	op	U		
	1	1	UNALLOCATED	-
0	0	0	SQRSHR (four registers)	FEAT_SME2
0	0	1	UQRSHR (four registers)	FEAT_SME2
0	1	0	SQRSHRU (four registers)	FEAT_SME2
1	0	0	SQRSHRN	FEAT_SME2
1	0	1	UQRSHRN	FEAT_SME2
1	1	0	SQRSHRUN	FEAT_SME2

SME2 Multi-vector - SVE Constructive Unary

These instructions are under [SME encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
11000001		op0	1		op1		111000																op2			op3					

op0	Decode fields			Instruction details
	op1	op2	op3	
00	00001		x0	SME2 multi-vec FP to int convert two registers
00	00001		x1	UNALLOCATED
00	00010		x0	SME2 multi-vec int to FP two registers
00	00010		x1	UNALLOCATED
00	00100			UNALLOCATED
00	10001	0x	00	SME2 multi-vec FP to int convert four registers
00	10001	0x	!= 00	UNALLOCATED
00	10001	1x		UNALLOCATED
00	10010	0x	00	SME2 multi-vec int to FP four registers
00	10010	0x	!= 00	UNALLOCATED
00	10010	1x		UNALLOCATED
00	10111	00	x0	SME2 multi-vec quadwords ZIP four registers
00	1011x	00	x1	UNALLOCATED
00	1011x	!= 00		UNALLOCATED
00	10x00			UNALLOCATED
01	00001			UNALLOCATED
01	x0010			UNALLOCATED
0x	00000			SME2 multi-vec FP down convert two registers

0x	00011			SME2 multi-vec int down convert two registers
0x	0011x			UNALLOCATED
10	00000			SME2 multi-vec convert two registers
10	00001			UNALLOCATED
10	00x1x			UNALLOCATED
11	000xx			UNALLOCATED
11	0011x			UNALLOCATED
1x	10010			UNALLOCATED
!= 00	1000x			UNALLOCATED
!= 00	10110	00	x1	UNALLOCATED
!= 00	10110	01		UNALLOCATED
!= 00	10110	1x		UNALLOCATED
!= 00	10111			UNALLOCATED
!= 00	x0100			UNALLOCATED
	00101			SME2 multi-vec unpack two registers
	01xxx	x0	x0	SME2 multi-vec FRINT two registers
	01xxx	x0	x1	UNALLOCATED
	01xxx	x1		UNALLOCATED
	10011			SME2 multi-vec int down convert four registers
	10101	x0	0x	SME2 multi-vec unpack four registers
	10101	x0	1x	UNALLOCATED
	10101	x1		UNALLOCATED
	10110	00	x0	SME2 multi-vec ZIP four registers
	11xxx	00	00	SME2 multi-vec FRINT four registers
	11xxx	00	!= 00	UNALLOCATED
	11xxx	!= 00		UNALLOCATED

SME2 multi-vec FP to int convert two registers

These instructions are under [SME2 Multi-vector - SVE Constructive Unary](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	0	0	1	0	0	0	0	1	1	1	1	1	0	0	0	Zn	U	Zd	0					

Decode fields U	Instruction Details	Feature
0	FCVTZS	FEAT_SME2
1	FCVTZU	FEAT_SME2

SME2 multi-vec int to FP two registers

These instructions are under [SME2 Multi-vector - SVE Constructive Unary](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	0	0	1	0	0	0	1	0	1	1	1	0	0	0	Zn	U	Zd	0	0	0	0	0	0	

Decode fields U	Instruction Details	Feature
0	SCVT	FEAT_SME2
1	UCVT	FEAT_SME2

SME2 multi-vec FP to int convert four registers

These instructions are under [SME2 Multi-vector - SVE Constructive Unary](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	0	0	1	1	0	0	0	1	1	1	1	0	0	0	Zn	0	U	Zd	0	0	0	0	0	0

Decode fields U	Instruction Details	Feature
0	FCVTZS	FEAT_SME2
1	FCVTZU	FEAT_SME2

SME2 multi-vec int to FP four registers

These instructions are under [SME2 Multi-vector - SVE Constructive Unary](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	0	0	1	1	0	0	1	0	1	1	1	0	0	0	Zn	0	U	Zd	0	0	0	0	0	0

Decode fields U	Instruction Details	Feature
0	SCVT	FEAT_SME2
1	UCVT	FEAT_SME2

SME2 multi-vec quadwords ZIP four registers

These instructions are under [SME2 Multi-vector - SVE Constructive Unary](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	0	0	1	1	0	1	1	1	1	1	1	0	0	0	Zn	0	0	Zd	op	0				

Decode fields op	Instruction Details	Feature
0	ZIP (four registers)	FEAT_SME2
1	UZP (four registers)	FEAT_SME2

SME2 multi-vec FP down convert two registers

These instructions are under [SME2 Multi-vector - SVE Constructive Unary](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	0	op	1	0	0	0	0	1	1	1	1	0	0	0	Zn	N		Zd						

Decode fields op	N	Instruction Details	Feature
0	0	FCVT (narrowing)	FEAT_SME2
0	1	FCVTN	FEAT_SME2
1	0	BFCVT	FEAT_SME2
1	1	BFCVTN	FEAT_SME2

SME2 multi-vec int down convert two registers

These instructions are under [SME2 Multi-vector - SVE Constructive Unary](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	0	op	1	0	0	0	1	1	1	1	1	0	0	0	Zn	U		Zd						

Decode fields op	U	Instruction Details	Feature
0	0	SQCVT (two registers)	FEAT_SME2
0	1	UQCVT (two registers)	FEAT_SME2
1	0	SQCVTU (two registers)	FEAT_SME2
1	1	UNALLOCATED	-

SME2 multi-vec convert two registers

These instructions are under [SME2 Multi-vector - SVE Constructive Unary](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	1	0	1	0	0	0	0	1	1	1	0	0	0	Zn		Zd		L						

Decode fields L	Instruction Details	Feature
0	FCVT (widening)	FEAT_SME_F16F16
1	FCVTL	FEAT_SME_F16F16

SME2 multi-vec unpack two registers

These instructions are under [SME2 Multi-vector - SVE Constructive Unary](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	size	1	0	0	1	0	1	1	1	0	0	0	Zn		Zd		U							

Decode fields U	Instruction Details	Feature
0	SUNPK	FEAT_SME2
1	UUNPK	FEAT_SME2

SME2 multi-vec FRINT two registers

These instructions are under [SME2 Multi-vector - SVE Constructive Unary](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	size	1	0	1	opc	1	1	1	0	0	0	Zn	0	Zd	0									

Decode fields size	opc	Instruction Details	Feature
0x		UNALLOCATED	-
10	000	FRINTN	FEAT_SME2
10	001	FRINTP	FEAT_SME2
10	010	FRINTM	FEAT_SME2
10	011	UNALLOCATED	-
10	100	FRINTA	FEAT_SME2
10	101	UNALLOCATED	-
10	11x	UNALLOCATED	-
11		UNALLOCATED	-

SME2 multi-vec int down convert four registers

These instructions are under [SME2 Multi-vector - SVE Constructive Unary](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	sz	op	1	1	0	0	1	1	1	1	1	0	0	0	Zn	N	U		Zd					

Decode fields			Instruction Details												Feature																	
op	N	U																														
0	0	0	SQCVT (four registers)												FEAT_SME2																	
0	0	1	UQCVT (four registers)												FEAT_SME2																	
0	1	0	SQCVTN												FEAT_SME2																	
0	1	1	UQCVTN												FEAT_SME2																	
1		1	UNALLOCATED												-																	
1	0	0	SQCVTU (four registers)												FEAT_SME2																	
1	1	0	SQCVTUN												FEAT_SME2																	

SME2 multi-vec unpack four registers

These instructions are under [SME2 Multi-vector - SVE Constructive Unary](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	size	1	1	0	1	0	1	1	1	1	1	0	0	0	Zn	0	Zd	0	U					

Decode fields			Instruction Details												Feature																	
U																																
0			SUNPK												FEAT_SME2																	
1			UUNPK												FEAT_SME2																	

SME2 multi-vec ZIP four registers

These instructions are under [SME2 Multi-vector - SVE Constructive Unary](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	size	1	1	0	1	1	0	1	1	1	1	0	0	0	Zn	0	0	Zd	op	0				

Decode fields			Instruction Details												Feature																		
op																																	
0			ZIP (four registers)												FEAT_SME2																		
1			UZP (four registers)												FEAT_SME2																		

SME2 multi-vec FRINT four registers

These instructions are under [SME2 Multi-vector - SVE Constructive Unary](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	size	1	1	1	opc	1	1	1	0	0	0	Zn	0	0	Zd	0	0							

Decode fields		Instruction Details		Feature
size	opc			
0x		UNALLOCATED		-
10	000	FRINTN		FEAT_SME2
10	001	FRINTP		FEAT_SME2
10	010	FRINTM		FEAT_SME2
10	011	UNALLOCATED		-
10	100	FRINTA		FEAT_SME2
10	101	UNALLOCATED		-
10	11x	UNALLOCATED		-
11		UNALLOCATED		-

SME2 Multi-vector - Multiple Vectors SVE Destructive (Two registers)

These instructions are under [SME encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
11000001							1					010110		op0		op1		op2													

Decode fields			Instruction details
op0	op1	op2	
0	000	0x	SME2 multiple vectors int min/max two registers
0	010		SME2 multiple vectors FP min/max two registers
0	0x1		UNALLOCATED
0	10x		SME2 multiple vectors shift two registers
0	11x		UNALLOCATED
1	000	00	SME2 multi-vector signed saturating doubling multiply high two registers
1	000	01	UNALLOCATED
1	!= 000		UNALLOCATED
	000	1x	UNALLOCATED

SME2 multiple vectors int min/max two registers

These instructions are under [SME2 Multi-vector - Multiple Vectors SVE Destructive \(Two registers\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	size	1	Zm	0	1	0	1	1	0	0	0	0	0	0	0	1	0	opc	Zdn	U				

Decode fields		Instruction Details		Feature
op	U			
0	0	SMAX (multiple vectors)		FEAT_SME2
0	1	UMAX (multiple vectors)		FEAT_SME2
1	0	SMIN (multiple vectors)		FEAT_SME2
1	1	UMIN (multiple vectors)		FEAT_SME2

SME2 multiple vectors FP min/max two registers

These instructions are under [SME2 Multi-vector - Multiple Vectors SVE Destructive \(Two registers\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	size	1	Zm	0	1	0	1	1	0	0	0	1	0	0	0	1	0	opc	Zdn	o2				

Decode fields			Instruction Details		Feature
size	opc	o2			
	1x		UNALLOCATED		-
!= 00	00	0	FMAX (multiple vectors)		FEAT_SME2
!= 00	00	1	FMIN (multiple vectors)		FEAT_SME2
!= 00	01	0	FMAXNM (multiple vectors)		FEAT_SME2
!= 00	01	1	FMINNM (multiple vectors)		FEAT_SME2
00	00	0	BFMAX (multiple vectors)		FEAT_SVE_B16B16
00	00	1	BFMIN (multiple vectors)		FEAT_SVE_B16B16
00	01	0	BFMAXNM (multiple vectors)		FEAT_SVE_B16B16
00	01	1	BFMINNM (multiple vectors)		FEAT_SVE_B16B16

SME2 multiple vectors shift two registers

These instructions are under [SME2 Multi-vector - Multiple Vectors SVE Destructive \(Two registers\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	size	1	Zm	0	1	0	1	1	0	0	1	0	1	0	0	1	0	opc	Zdn	U				

Decode fields		Instruction Details		Feature
opc	U			
000		UNALLOCATED		-

Decode fields		Instruction Details		Feature
opc	U			
001	0	SRSHL (multiple vectors)		FEAT_SME2
001	1	URSHL (multiple vectors)		FEAT_SME2
01x		UNALLOCATED		-
1xx		UNALLOCATED		-

SME2 multi-vector signed saturating doubling multiply high two registers

These instructions are under [SME2 Multi-vector - Multiple Vectors SVE Destructive \(Two registers\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	size	1	Zm	0	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	Zdn	op		

Decode fields		Instruction Details		Feature
op				
0		SQDMULH (multiple vectors)		FEAT_SME2
1		UNALLOCATED		-

SME2 Multi-vector - Multiple Vectors SVE Destructive (Four registers)

These instructions are under [SME encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
11000001			1		op0	010111	op1	op2	op3		0																				

Decode fields				Instruction details
op0	op1	op2	op3	
0	0	000	0x	SME2 multiple vectors int min/max four registers
0	0	010		SME2 multiple vectors FP min/max four registers
0	0	0x1		UNALLOCATED
0	0	10x		SME2 multiple vectors shift four registers
0	0	11x		UNALLOCATED
0	1	000	00	SME2 multi-vector signed saturating doubling multiply high four registers
0	1	000	01	UNALLOCATED
0	1	!= 000		UNALLOCATED
0		000	1x	UNALLOCATED
1				UNALLOCATED

SME2 multiple vectors int min/max four registers

These instructions are under [SME2 Multi-vector - Multiple Vectors SVE Destructive \(Four registers\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	size	1	Zm	0	0	1	0	1	1	1	0	0	0	0	0	op	Zdn	0	U					

Decode fields		Instruction Details		Feature
op	U			
0	0	SMAX (multiple vectors)		FEAT_SME2
0	1	UMAX (multiple vectors)		FEAT_SME2
1	0	SMIN (multiple vectors)		FEAT_SME2
1	1	UMIN (multiple vectors)		FEAT_SME2

SME2 multiple vectors FP min/max four registers

These instructions are under [SME2 Multi-vector - Multiple Vectors SVE Destructive \(Four registers\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	size	1	Zm	0	0	1	0	1	1	1	0	0	1	0	opc	Zdn	0	o2						

Decode fields			Instruction Details		Feature
size	opc	o2			
	1x		UNALLOCATED		-
!= 00	00	0	FMAX (multiple vectors)		FEAT_SME2
!= 00	00	1	FMIN (multiple vectors)		FEAT_SME2
!= 00	01	0	FMAXNM (multiple vectors)		FEAT_SME2
!= 00	01	1	FMINNM (multiple vectors)		FEAT_SME2
00	00	0	BFMAX (multiple vectors)		FEAT_SVE_B16B16
00	00	1	BFMIN (multiple vectors)		FEAT_SVE_B16B16
00	01	0	BFMAXNM (multiple vectors)		FEAT_SVE_B16B16
00	01	1	BFMINNM (multiple vectors)		FEAT_SVE_B16B16

SME2 multiple vectors shift four registers

These instructions are under [SME2 Multi-vector - Multiple Vectors SVE Destructive \(Four registers\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	size	1	Zm	0	0	1	0	1	1	1	0	1	0	opc	Zdn	0	U							

Decode fields		Instruction Details	Feature
opc	U		
000		UNALLOCATED	-
001	0	SRSHL (multiple vectors)	FEAT_SME2
001	1	URSHL (multiple vectors)	FEAT_SME2
01x		UNALLOCATED	-
1xx		UNALLOCATED	-

SME2 multi-vector signed saturating doubling multiply high four registers

These instructions are under [SME2 Multi-vector - Multiple Vectors SVE Destructive \(Four registers\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	size	1	Zm	0	0	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	Zdn	0	op

Decode fields		Instruction Details	Feature
op			
0		SQDMULH (multiple vectors)	FEAT_SME2
1		UNALLOCATED	-

SME2 Multi-vector - Multiple and Single SVE Destructive (Two registers)

These instructions are under [SME encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
11000001			10								10100	op0	op1																		

Decode fields		Instruction details
op0	op1	
0	0000x	SME2 single-multi int min/max two registers
0	0100x	SME2 single-multi FP min/max two registers
0	0x!=00x	UNALLOCATED
0	10xxx	SME2 single-multi shift two registers
0	11000	SME2 single-multi add two registers
0	11!=000	UNALLOCATED
1	00000	SME2 single-multi signed saturating doubling multiply high two registers
1	!=00000	UNALLOCATED

SME2 single-multi int min/max two registers

These instructions are under [SME2 Multi-vector - Multiple and Single SVE Destructive \(Two registers\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	size	1	0	Zm	1	0	1	0	0	0	0	0	0	0	0	op	Zdn	U						

Decode fields		Instruction Details										Feature							
op	U																		
0	0	SMAX (multiple and single vector)										FEAT_SME2							
0	1	UMAX (multiple and single vector)										FEAT_SME2							
1	0	SMIN (multiple and single vector)										FEAT_SME2							
1	1	UMIN (multiple and single vector)										FEAT_SME2							

SME2 single-multi FP min/max two registers

These instructions are under [SME2 Multi-vector - Multiple and Single SVE Destructive \(Two registers\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	size	1	0	Zm	1	0	1	0	0	0	0	1	0	0	0	op	Zdn	o2						

Decode fields			Instruction Details										Feature							
size	op	o2																		
!= 00	0	0	FMAX (multiple and single vector)										FEAT_SME2							
!= 00	0	1	FMIN (multiple and single vector)										FEAT_SME2							
!= 00	1	0	FMAXNM (multiple and single vector)										FEAT_SME2							
!= 00	1	1	FMINNM (multiple and single vector)										FEAT_SME2							
00	0	0	BFMAX (multiple and single vector)										FEAT_SVE_B16B16							
00	0	1	BFMIN (multiple and single vector)										FEAT_SVE_B16B16							
00	1	0	BFMAXNM (multiple and single vector)										FEAT_SVE_B16B16							
00	1	1	BFMINNM (multiple and single vector)										FEAT_SVE_B16B16							

SME2 single-multi shift two registers

These instructions are under [SME2 Multi-vector - Multiple and Single SVE Destructive \(Two registers\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	size	1	0	Zm	1	0	1	0	0	0	1	0	opc	Zdn	U									

Decode fields		Instruction Details											Feature					
opc	U																	
000		UNALLOCATED											-					
001	0	SRSHL (multiple and single vector)											FEAT_SME2					
001	1	URSHL (multiple and single vector)											FEAT_SME2					
01x		UNALLOCATED											-					
1xx		UNALLOCATED											-					

SME2 single-multi add two registers

These instructions are under [SME2 Multi-vector - Multiple and Single SVE Destructive \(Two registers\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	size	1	0	Zm	1	0	1	0	0	0	1	1	0	0	0	Zdn	op							

Decode fields		Instruction Details											Feature					
op																		
0		ADD (to vector)											FEAT_SME2					
1		UNALLOCATED											-					

SME2 single-multi signed saturating doubling multiply high two registers

These instructions are under [SME2 Multi-vector - Multiple and Single SVE Destructive \(Two registers\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	size	1	0	Zm	1	0	1	0	0	1	0	0	0	0	0	Zdn	op							

Decode fields		Instruction Details											Feature					
op																		
0		SQDMULH (multiple and single vector)											FEAT_SME2					
1		UNALLOCATED											-					

SME2 Multi-vector - Multiple and Single SVE Destructive (Four registers)

These instructions are under [SME encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
11000001																10101	op0	op1												0	

Decode fields		Instruction details	
op0	op1		
0	0000x	SME2 single-multi int min/max four registers	
0	0100x	SME2 single-multi FP min/max four registers	
0	0x!=00x	UNALLOCATED	
0	10xxx	SME2 single-multi shift four registers	
0	11000	SME2 single-multi add four registers	
0	11!=000	UNALLOCATED	
1	00000	SME2 single-multi signed saturating doubling multiply high four registers	
1	!=00000	UNALLOCATED	

SME2 single-multi int min/max four registers

These instructions are under [SME2 Multi-vector - Multiple and Single SVE Destructive \(Four registers\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	size	1	0	Zm	1	0	1	0	1	0	0	0	0	0	op	Zdn	0	U						

Decode fields		Instruction Details		Feature
op	U			
0	0	SMAX (multiple and single vector)		FEAT_SME2
0	1	UMAX (multiple and single vector)		FEAT_SME2
1	0	SMIN (multiple and single vector)		FEAT_SME2
1	1	UMIN (multiple and single vector)		FEAT_SME2

SME2 single-multi FP min/max four registers

These instructions are under [SME2 Multi-vector - Multiple and Single SVE Destructive \(Four registers\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	size	1	0	Zm	1	0	1	0	1	0	0	1	0	0	op	Zdn	0	o2						

Decode fields			Instruction Details	Feature
size	op	o2		
!= 00	0	0	FMAX (multiple and single vector)	FEAT_SME2
!= 00	0	1	FMIN (multiple and single vector)	FEAT_SME2
!= 00	1	0	FMAXNM (multiple and single vector)	FEAT_SME2
!= 00	1	1	FMINNM (multiple and single vector)	FEAT_SME2
00	0	0	BFMAX (multiple and single vector)	FEAT_SVE_B16B16
00	0	1	BFMIN (multiple and single vector)	FEAT_SVE_B16B16
00	1	0	BFMAXNM (multiple and single vector)	FEAT_SVE_B16B16
00	1	1	BFMINNM (multiple and single vector)	FEAT_SVE_B16B16

SME2 single-multi shift four registers

These instructions are under [SME2 Multi-vector - Multiple and Single SVE Destructive \(Four registers\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	size	1	0	Zm	1	0	1	0	1	0	1	0	1	0	opc	Zdn	0	U						

Decode fields		Instruction Details	Feature
opc	U		
000		UNALLOCATED	-
001	0	SRSHL (multiple and single vector)	FEAT_SME2
001	1	URSHL (multiple and single vector)	FEAT_SME2
01x		UNALLOCATED	-
1xx		UNALLOCATED	-

SME2 single-multi add four registers

These instructions are under [SME2 Multi-vector - Multiple and Single SVE Destructive \(Four registers\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	size	1	0	Zm	1	0	1	0	1	0	1	1	0	0	0	Zdn	0	op						

Decode fields		Instruction Details	Feature
op			
0		ADD (to vector)	FEAT_SME2
1		UNALLOCATED	-

SME2 single-multi signed saturating doubling multiply high four registers

These instructions are under [SME2 Multi-vector - Multiple and Single SVE Destructive \(Four registers\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	size	1	0	Zm	1	0	1	0	1	1	0	0	0	0	0	Zdn	0	op						

Decode fields		Instruction Details	Feature
op			
0		SQDMULH (multiple and single vector)	FEAT_SME2
1		UNALLOCATED	-

SME2 Multi-vector - Multiple and Single Array Vectors

These instructions are under [SME encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
110000010	op0	op1	op1												0		op2									op3					

Decode fields				Instruction details
op0	op1	op2	op3	
0	0	010	xx0x	SME2 single-multi long FMA two sources
0	0	011		SME2 multiple and single vector long FMA one source
0	0	100		SME2 single-multi FP dot product two registers
0	0	101	x1xx	SME2 single-multi mixed dot product two registers
0	1	010	xx0x	SME2 single-multi long FMA four sources
0	1	100		SME2 single-multi FP dot product four registers
0	1	101	x1xx	SME2 single-multi mixed dot product four registers
1	0	010	xx0x	SME2 single-multi long MLA two sources

1	0	011		SME2 multiple and single vector long MLA one source
1	0	101	x1xx	SME2 single-multi two-way dot product two registers
1	1	010	xx0x	SME2 single-multi long MLA four sources
1	1	101	x1xx	SME2 single-multi two-way dot product four registers
1		100		UNALLOCATED
	0	000	xxx0	SME2 single-multi long long MLA two sources
	0	001		SME2 multiple and single vector long long FMA one source
	0	101	x0xx	SME2 single-multi four-way dot product two registers
	0	110	0xxx	SME2 single-multi ternary FP two registers
	0	110	1xxx	SME2 single-multi ternary int two registers
	0	111	0xxx	SME2 single-multi ternary FP16 two registers
	1	000	xxx0	SME2 single-multi long long MLA four sources
	1	0x1		UNALLOCATED
	1	101	x0xx	SME2 single-multi four-way dot product four registers
	1	110	0xxx	SME2 single-multi ternary FP four registers
	1	110	1xxx	SME2 single-multi ternary int four registers
	1	111	0xxx	SME2 single-multi ternary FP16 four registers
		000	xxx1	UNALLOCATED
		010	xx1x	UNALLOCATED
		111	1xxx	UNALLOCATED

SME2 single-multi long FMA two sources

These instructions are under [SME2 Multi-vector - Multiple and Single Array Vectors](#).

Decode fields		Instruction Details	Feature
op	S		
0	0	FMLAL (multiple and single vector)	FEAT_SME2
0	1	FMLS (multiple and single vector)	FEAT_SME2
1	0	BFMLAL (multiple and single vector)	FEAT_SME2
1	1	BFMLS (multiple and single vector)	FEAT_SME2

SME2 multiple and single vector long FMA one source

These instructions are under [SME2 Multi-vector - Multiple and Single Array Vectors](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	0	0	1	0	Zm	0	Rv	0	1	1					Zn	op	S	off3						

Decode fields		Instruction Details	Feature
op	S		
0	0	FMLAL (multiple and single vector)	FEAT_SME2
0	1	FMLS (multiple and single vector)	FEAT_SME2
1	0	BFMLAL (multiple and single vector)	FEAT_SME2
1	1	BFMLS (multiple and single vector)	FEAT_SME2

SME2 single-multi FP dot product two registers

These instructions are under [SME2 Multi-vector - Multiple and Single Array Vectors](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	0	0	1	0	Zm	0	Rv	1	0	0					Zn	opc	off3							

Decode fields		Instruction Details	Feature
opc			
x1		UNALLOCATED	-
00		FDOT (multiple and single vector)	FEAT_SME2
10		BFDOT (multiple and single vector)	FEAT_SME2

SME2 single-multi mixed dot product two registers

These instructions are under [SME2 Multi-vector - Multiple and Single Array Vectors](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	0	0	1	0	Zm	0	Rv	1	0	1	Zn	U	1	off3										

Decode fields		Instruction Details										Feature									
U																					
0		USDOT (multiple and single vector)										FEAT_SME2									
1		SUDOT (multiple and single vector)										FEAT_SME2									

SME2 single-multi long FMA four sources

These instructions are under [SME2 Multi-vector - Multiple and Single Array Vectors](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	0	0	1	1	Zm	0	Rv	0	1	0	Zn	op	S	0	off2									

Decode fields		Instruction Details										Feature									
op		S																			
0		FMLAL (multiple and single vector)										FEAT_SME2									
0		FMLS L (multiple and single vector)										FEAT_SME2									
1		BFMLAL (multiple and single vector)										FEAT_SME2									
1		BFMLS L (multiple and single vector)										FEAT_SME2									

SME2 single-multi FP dot product four registers

These instructions are under [SME2 Multi-vector - Multiple and Single Array Vectors](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	0	0	1	1	Zm	0	Rv	1	0	0	Zn	opc	off3											

Decode fields		Instruction Details										Feature									
opc																					
x1		UNALLOCATED										-									
00		FDOT (multiple and single vector)										FEAT_SME2									
10		BFDOT (multiple and single vector)										FEAT_SME2									

SME2 single-multi mixed dot product four registers

These instructions are under [SME2 Multi-vector - Multiple and Single Array Vectors](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	0	0	1	1	Zm	0	Rv	1	0	1	Zn	U	1	off3										

Decode fields	Instruction Details	Feature
U		
0	USDOT (multiple and single vector)	FEAT_SME2
1	SUDOT (multiple and single vector)	FEAT_SME2

SME2 single-multi long MLA two sources

These instructions are under [SME2 Multi-vector - Multiple and Single Array Vectors](#).

Decode fields		Instruction Details	Feature
U	S		
0	0	SMLAL (multiple and single vector)	FEAT_SME2
0	1	SMLS L (multiple and single vector)	FEAT_SME2
1	0	UMLAL (multiple and single vector)	FEAT_SME2
1	1	UMLSL (multiple and single vector)	FEAT_SME2

SME2 multiple and single vector long MLA one source

These instructions are under [SME2 Multi-vector - Multiple and Single Array Vectors](#).

Decode fields		Instruction Details	Feature
U	S		
0	0	SMLAL (multiple and single vector)	FEAT_SME2
0	1	SMLS L (multiple and single vector)	FEAT_SME2
1	0	UMLAL (multiple and single vector)	FEAT_SME2
1	1	UMLSL (multiple and single vector)	FEAT_SME2

SME2 single-multi two-way dot product two registers

These instructions are under [SME2 Multi-vector - Multiple and Single Array Vectors](#).

Decode fields	Instruction Details			Feature
U				
0	SDOT (2-way, multiple and single vector)			FEAT_SME2
1	UDOT (2-way, multiple and single vector)			FEAT_SME2

SME2 single-multi long MLA four sources

These instructions are under [SME2 Multi-vector - Multiple and Single Array Vectors](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
1	1	0	0	0	0	0	1	0	1	1	1	Zm	0	Rv	0	1	0	Zn	U	S	0	off2										

Decode fields	Instruction Details			Feature	
U	S				
0	0	SMLAL (multiple and single vector)			FEAT_SME2
0	1	SMLS L (multiple and single vector)			FEAT_SME2
1	0	UMLAL (multiple and single vector)			FEAT_SME2
1	1	UMLSL (multiple and single vector)			FEAT_SME2

SME2 single-multi two-way dot product four registers

These instructions are under [SME2 Multi-vector - Multiple and Single Array Vectors](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	0	1	1	1	Zm	0	Rv	1	0	1	Zn	U	1	off3										

Decode fields	Instruction Details			Feature
U				
0	SDOT (2-way, multiple and single vector)			FEAT_SME2
1	UDOT (2-way, multiple and single vector)			FEAT_SME2

SME2 single-multi long long MLA two sources

These instructions are under [SME2 Multi-vector - Multiple and Single Array Vectors](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
1	1	0	0	0	0	0	1	0	sz	1	0	Zm	0	Rv	0	0	0	Zn	U	S	op	0	01									

Decode fields				Instruction Details	Feature
sz	U	S	op		
	0	0	0	SMLALL (multiple and single vector)	FEAT_SME2
	0	1	0	SMMLSLL (multiple and single vector)	FEAT_SME2
	1	0	0	UMLALL (multiple and single vector)	FEAT_SME2
	1	1	0	UMLSLL (multiple and single vector)	FEAT_SME2
0		1	1	UNALLOCATED	-
0	0	0	1	USMLALL (multiple and single vector)	FEAT_SME2
0	1	0	1	SUMLALL (multiple and single vector)	FEAT_SME2
1			1	UNALLOCATED	-

SME2 multiple and single vector long long FMA one source

These instructions are under [SME2 Multi-vector - Multiple and Single Array Vectors](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
1	1	0	0	0	0	0	1	0	sz	1	0	Zm	0	Rv	0	0	1	Zn	U	S	op	off2										

Decode fields				Instruction Details	Feature
sz	U	S	op		
	0	0	0	SMLALL (multiple and single vector)	FEAT_SME2
	0	1	0	SMMLSLL (multiple and single vector)	FEAT_SME2
	1	0	0	UMLALL (multiple and single vector)	FEAT_SME2
	1	1	0	UMLSLL (multiple and single vector)	FEAT_SME2
0	0	0	1	USMLALL (multiple and single vector)	FEAT_SME2
0	0	1	1	UNALLOCATED	-
0	1		1	UNALLOCATED	-
1			1	UNALLOCATED	-

SME2 single-multi four-way dot product two registers

These instructions are under [SME2 Multi-vector - Multiple and Single Array Vectors](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	0	sz	1	0	Zm	0	Rv	1	0	1	Zn	U	0	off3										

Decode fields

Instruction Details

Feature

U

0	SDOT (4-way, multiple and single vector)	FEAT_SME2
1	UDOT (4-way, multiple and single vector)	FEAT_SME2

SME2 single-multi ternary FP two registers

These instructions are under [SME2 Multi-vector - Multiple and Single Array Vectors](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	0	sz	1	0	Zm	0	Rv	1	1	0	Zn	0	S	off3										

Decode fields

S

Instruction Details

Feature

0	FMLA (multiple and single vector)	FEAT_SME2
1	FMLS (multiple and single vector)	FEAT_SME2

SME2 single-multi ternary int two registers

These instructions are under [SME2 Multi-vector - Multiple and Single Array Vectors](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	0	sz	1	0	Zm	0	Rv	1	1	0	Zn	1	S	off3										

Decode fields

S

Instruction Details

Feature

0	ADD (array results, multiple and single vector)	FEAT_SME2
1	SUB (array results, multiple and single vector)	FEAT_SME2

SME2 single-multi ternary FP16 two registers

These instructions are under [SME2 Multi-vector - Multiple and Single Array Vectors](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	0	sz	1	0	Zm	0	Rv	1	1	1	Zn	0	S	off3										

Decode fields		Instruction Details			Feature
sz	S				
0	0	FMLA (multiple and single vector)			FEAT_SME_F16F16
0	1	FMLS (multiple and single vector)			FEAT_SME_F16F16
1	0	BFMLA (multiple and single vector)			FEAT_SVE_B16B16
1	1	BFMLS (multiple and single vector)			FEAT_SVE_B16B16

SME2 single-multi long long MLA four sources

These instructions are under [SME2 Multi-vector - Multiple and Single Array Vectors](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	0	sz	1	1	Zm	0	Rv	0	0	0	Zn	U	S	op	0	0	1							

Decode fields				Instruction Details			Feature
sz	U	S	op				
	0	0	0	SMLALL (multiple and single vector)			FEAT_SME2
	0	1	0	SMLSLL (multiple and single vector)			FEAT_SME2
	1	0	0	UMLALL (multiple and single vector)			FEAT_SME2
	1	1	0	UMLSLL (multiple and single vector)			FEAT_SME2
0		1	1	UNALLOCATED			-
0	0	0	1	USMLALL (multiple and single vector)			FEAT_SME2
0	1	0	1	SUMLALL (multiple and single vector)			FEAT_SME2
1			1	UNALLOCATED			-

SME2 single-multi four-way dot product four registers

These instructions are under [SME2 Multi-vector - Multiple and Single Array Vectors](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	0	sz	1	1	Zm	0	Rv	1	0	1	Zn	U	0	off3										

Decode fields

Instruction Details

Feature

U

0	SDOT (4-way, multiple and single vector)	FEAT_SME2
1	UDOT (4-way, multiple and single vector)	FEAT_SME2

SME2 single-multi ternary FP four registers

These instructions are under [SME2 Multi-vector - Multiple and Single Array Vectors](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	0	sz	1	1	Zm	0	Rv	1	1	0	Zn	0	S	off3										

Decode fields

S

Instruction Details

Feature

0	FMLA (multiple and single vector)	FEAT_SME2
1	FMLS (multiple and single vector)	FEAT_SME2

SME2 single-multi ternary int four registers

These instructions are under [SME2 Multi-vector - Multiple and Single Array Vectors](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	0	sz	1	1	Zm	0	Rv	1	1	0	Zn	1	S	off3										

Decode fields

S

Instruction Details

Feature

0	ADD (array results, multiple and single vector)	FEAT_SME2
1	SUB (array results, multiple and single vector)	FEAT_SME2

SME2 single-multi ternary FP16 four registers

These instructions are under [SME2 Multi-vector - Multiple and Single Array Vectors](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	0	sz	1	1		Zm	0	Rv	1	1	1		Zn		0	S		off3						

Decode fields		Instruction Details	Feature
sz	S		
0	0	FMLA (multiple and single vector)	FEAT_SME_F16F16
0	1	FMLS (multiple and single vector)	FEAT_SME_F16F16
1	0	BFMLA (multiple and single vector)	FEAT_SVE_B16B16
1	1	BFMLS (multiple and single vector)	FEAT_SVE_B16B16

SME2 Multi-vector - Multiple Array Vectors (Two registers)

These instructions are under [SME encodings](#).

Decode fields					Instruction details	Feature
op0	op1	op2	op3	op4		
0	00	x0	11x	1xxxx	UNALLOCATED	-
0			010	0xx0x	SME2 multiple vectors long FMA two sources	-
0			100	1x1xx	UNALLOCATED	-
0			100	xx0xx	SME2 multiple vectors FP dot product two registers	-
0			101	001xx	USDOT (multiple vectors) – Two ZA single-vectors	FEAT_SME2
0			101	011xx	UNALLOCATED	-
0			101	1xxxx	UNALLOCATED	-
1	00	x0	1xx	1xxxx	UNALLOCATED	-
1	00	x1	10x	1xxxx	UNALLOCATED	-
1	!= 00		10x	1xxxx	UNALLOCATED	-
1			010	0xx0x	SME2 multiple vectors long MLA two sources	-

1			100	0x0xx	UNALLOCATED	-
1			101	0x1xx	SME2 multiple vectors two-way dot product two registers	-
	00	00	111	00xxx	SME2 multiple vectors binary FP two registers	-
	00	00	111	01xxx	SME2 multiple vectors binary int two registers	-
	00	10	111	00xxx	SME2 multiple vectors binary FP16 two registers	-
	00	10	111	01xxx	UNALLOCATED	-
	00	x1	110	1xxxx	UNALLOCATED	-
	00	x1	111		UNALLOCATED	-
	!= 00		110	1xxxx	UNALLOCATED	-
	!= 00		111		UNALLOCATED	-
			000	0xxxx0	SME2 multiple vectors long long MLA two sources	-
			000	0xxx1	UNALLOCATED	-
			010	0xx1x	UNALLOCATED	-
			0x0	1xxxx	UNALLOCATED	-
			0x1		UNALLOCATED	-
			100	0x1xx	SME2 multiple vectors ternary FP16 two registers	-
			101	0x0xx	SME2 multiple vectors four-way dot product two registers	-
			110	00xxx	SME2 multiple vectors ternary FP two registers	-
			110	01xxx	SME2 multiple vectors ternary int two registers	-

SME2 multiple vectors long FMA two sources

These instructions are under [SME2 Multi-vector - Multiple Array Vectors \(Two registers\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	1	0	1	Zm	0	0	Rv	0	1	0	Zn	0	op	S	0	off2								

Decode fields		Instruction Details			Feature
op	S				
0	0	FMLAL (multiple vectors)			FEAT_SME2
0	1	FMLS (multiple vectors)			FEAT_SME2
1	0	BFMLAL (multiple vectors)			FEAT_SME2
1	1	BFMLS (multiple vectors)			FEAT_SME2

SME2 multiple vectors FP dot product two registers

These instructions are under [SME2 Multi-vector - Multiple Array Vectors \(Two registers\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	1	0	1	Zm	0	0	Rv	1	0	0	Zn	0	opc	0	off3									

Decode fields		Instruction Details			Feature
opc					
00		FDOT (multiple vectors)			FEAT_SME2
01		BFDOT (multiple vectors)			FEAT_SME2
1x		UNALLOCATED			-

SME2 multiple vectors long MLA two sources

These instructions are under [SME2 Multi-vector - Multiple Array Vectors \(Two registers\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	1	1	1	Zm	0	0	Rv	0	1	0	Zn	0	U	S	0	off2								

Decode fields		Instruction Details			Feature
U	S				
0	0	SMLAL (multiple vectors)			FEAT_SME2
0	1	SMLS (multiple vectors)			FEAT_SME2
1	0	UMLAL (multiple vectors)			FEAT_SME2
1	1	UMLS (multiple vectors)			FEAT_SME2

SME2 multiple vectors two-way dot product two registers

These instructions are under [SME2 Multi-vector - Multiple Array Vectors \(Two registers\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	1	1	1	Zm	0	0	Rv	1	0	1	Zn	0	U	1	off3									

Decode fields U	Instruction Details	Feature
0	SDOT (2-way, multiple vectors)	FEAT_SME2
1	UDOT (2-way, multiple vectors)	FEAT_SME2

SME2 multiple vectors binary FP two registers

These instructions are under [SME2 Multi-vector - Multiple Array Vectors \(Two registers\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	1	sz	1	0	0	0	0	0	Rv	1	1	1	Zm	0	0	S	off3							

Decode fields S	Instruction Details	Feature
0	FADD	FEAT_SME2
1	FSUB	FEAT_SME2

SME2 multiple vectors binary int two registers

These instructions are under [SME2 Multi-vector - Multiple Array Vectors \(Two registers\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	1	sz	1	0	0	0	0	0	Rv	1	1	1	Zm	0	1	S	off3							

Decode fields S	Instruction Details	Feature
0	ADD (array accumulators)	FEAT_SME2
1	SUB (array accumulators)	FEAT_SME2

SME2 multiple vectors binary FP16 two registers

These instructions are under [SME2 Multi-vector - Multiple Array Vectors \(Two registers\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	1	sz	1	0	0	1	0	0	0	Rv	1	1	1	Zm	0	0	S	off3						

Decode fields		Instruction Details		Feature
sz	S			
0	0	FADD		FEAT_SME_F16F16
0	1	FSUB		FEAT_SME_F16F16
1	0	BFADD		FEAT_SVE_B16B16
1	1	BFSUB		FEAT_SVE_B16B16

SME2 multiple vectors long long MLA two sources

These instructions are under [SME2 Multi-vector - Multiple Array Vectors \(Two registers\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	1	sz	1	Zm	0	0	Rv	0	0	0	Zn	0	U	S	op	0	0	1	0					

Decode fields				Instruction Details				Feature
sz	U	S	op					
	0	0	0	SMLALL (multiple vectors)				FEAT_SME2
	0	1	0	SMILSLL (multiple vectors)				FEAT_SME2
	1	0	0	UMLALL (multiple vectors)				FEAT_SME2
	1	1	0	UMLSLL (multiple vectors)				FEAT_SME2
0	0	0	1	USMLALL (multiple vectors)				FEAT_SME2
0	0	1	1	UNALLOCATED				-
0	1		1	UNALLOCATED				-
1			1	UNALLOCATED				-

SME2 multiple vectors ternary FP16 two registers

These instructions are under [SME2 Multi-vector - Multiple Array Vectors \(Two registers\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	1	sz	1	Zm	0	0	Rv	1	0	0	Zn	0	S	1	off3									

Decode fields		Instruction Details		Feature
sz	S			
0	0	FMLA (multiple vectors)		FEAT_SME_F16F16
0	1	FMLS (multiple vectors)		FEAT_SME_F16F16
1	0	BFMLA (multiple vectors)		FEAT_SVE_B16B16
1	1	BFMLS (multiple vectors)		FEAT_SVE_B16B16

SME2 multiple vectors four-way dot product two registers

These instructions are under [SME2 Multi-vector - Multiple Array Vectors \(Two registers\)](#).

Decode fields	Instruction Details	Feature
U		
0	SDOT (4-way, multiple vectors)	FEAT_SME2
1	UDOT (4-way, multiple vectors)	FEAT_SME2

SME2 multiple vectors ternary FP two registers

These instructions are under [SME2 Multi-vector - Multiple Array Vectors \(Two registers\)](#).

Decode fields	Instruction Details	Feature
S		
0	FMLA (multiple vectors)	FEAT_SME2
1	FMLS (multiple vectors)	FEAT_SME2

SME2 multiple vectors ternary int two registers

These instructions are under [SME2 Multi-vector - Multiple Array Vectors \(Two registers\)](#).

Decode fields	Instruction Details	Feature
S		
0	ADD (array results, multiple vectors)	FEAT_SME2
1	SUB (array results, multiple vectors)	FEAT_SME2

SME2 Multi-vector - Multiple Array Vectors (Four registers)

These instructions are under [SME encodings](#).

Decode fields						Instruction details	Feature
op0	op1	op2	op3	op4	op5		
0	00	x0	11x	01		UNALLOCATED	-
0		x0	010	00	xx0x	SME2 multiple vectors long FMA four sources	-
0		x0	100	01	x1xx	UNALLOCATED	-
0		x0	100	0x	x0xx	SME2 multiple vectors FP dot product four registers	-
0		x0	101	00	01xx	USDOT (multiple vectors) â€“ Four ZA single-vectors	FEAT_SME2
0		x0	101	00	11xx	UNALLOCATED	-
0		x0	101	01		UNALLOCATED	-
1	00	x0	1xx	01		UNALLOCATED	-
1	!= 00	x0	10x	01		UNALLOCATED	-
1		x0	010	00	xx0x	SME2 multiple vectors long MLA four sources	-
1		x0	100	00	x0xx	UNALLOCATED	-
1		x0	101	00	x1xx	SME2 multiple vectors two-way dot product four registers	-
	00	00	111	00	0xxx	SME2 multiple vectors binary FP four registers	-
	00	00	111	00	1xxx	SME2 multiple vectors binary int four registers	-
	00	10	111	00	0xxx	SME2 multiple vectors binary FP16 four registers	-
	00	10	111	00	1xxx	UNALLOCATED	-
	00	x0	1xx	1x		UNALLOCATED	-
	!= 00	x0	10x	1x		UNALLOCATED	-

	$\neq 00$	x0	110	01		UNALLOCATED	-
	$\neq 00$	x0	110	1x		UNALLOCATED	-
	$\neq 00$	x0	111			UNALLOCATED	-
		x0	000	00	xxx0	SME2 multiple vectors long long MLA four sources	-
		x0	000	00	xxx1	UNALLOCATED	-
		x0	010	00	xx1x	UNALLOCATED	-
		x0	0x0	$\neq 00$		UNALLOCATED	-
		x0	0x1			UNALLOCATED	-
		x0	100	00	x1xx	SME2 multiple vectors ternary FP16 four registers	-
		x0	101	00	x0xx	SME2 multiple vectors four-way dot product four registers	-
		x0	110	00	0xxx	SME2 multiple vectors ternary FP four registers	-
		x0	110	00	1xxx	SME2 multiple vectors ternary int four registers	-
		x1				UNALLOCATED	-

SME2 multiple vectors long FMA four sources

These instructions are under [SME2 Multi-vector - Multiple Array Vectors \(Four registers\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	1	0	1	Zm	0	1	0	Rv	0	1	0	Zn	0	0	op	S	0	off2						

Decode fields	Instruction Details		Feature
op	S		
0	0	FMLAL (multiple vectors)	FEAT_SME2
0	1	FMLS L (multiple vectors)	FEAT_SME2
1	0	BFMLAL (multiple vectors)	FEAT_SME2
1	1	BFMLS L (multiple vectors)	FEAT_SME2

SME2 multiple vectors FP dot product four registers

These instructions are under [SME2 Multi-vector - Multiple Array Vectors \(Four registers\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	1	0	1	Zm	0	1	0	Rv	1	0	0	Zn	0	opc	0	off3								

Decode fields opc	Instruction Details	Feature
00	FDOT (multiple vectors)	FEAT_SME2
01	BFDOT (multiple vectors)	FEAT_SME2
1x	UNALLOCATED	-

SME2 multiple vectors long MLA four sources

These instructions are under [SME2 Multi-vector - Multiple Array Vectors \(Four registers\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	1	1	1	Zm	0	1	0	Rv	0	1	0	Zn	0	0	U	S	0	off2						

Decode fields U	S	Instruction Details	Feature
0	0	SMLAL (multiple vectors)	FEAT_SME2
0	1	SMLS L (multiple vectors)	FEAT_SME2
1	0	UMLAL (multiple vectors)	FEAT_SME2
1	1	UMLSL (multiple vectors)	FEAT_SME2

SME2 multiple vectors two-way dot product four registers

These instructions are under [SME2 Multi-vector - Multiple Array Vectors \(Four registers\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	1	1	1	Zm	0	1	0	Rv	1	0	1	Zn	0	0	U	1	off3							

Decode fields U	Instruction Details	Feature
0	SDOT (2-way, multiple vectors)	FEAT_SME2
1	UDOT (2-way, multiple vectors)	FEAT_SME2

SME2 multiple vectors binary FP four registers

These instructions are under [SME2 Multi-vector - Multiple Array Vectors \(Four registers\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	1	sz	1	0	0	0	1	0	Rv	1	1	1	Zm	0	0	0	S	off3						

Decode fields
S

Instruction Details

Feature

0	FADD	FEAT_SME2
1	FSUB	FEAT_SME2

SME2 multiple vectors binary int four registers

These instructions are under [SME2 Multi-vector - Multiple Array Vectors \(Four registers\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	1	sz	1	0	0	0	0	1	0	Rv	1	1	1	Zm	0	0	1	S	off3					

Decode fields
S

Instruction Details

Feature

0	ADD (array accumulators)	FEAT_SME2
1	SUB (array accumulators)	FEAT_SME2

SME2 multiple vectors binary FP16 four registers

These instructions are under [SME2 Multi-vector - Multiple Array Vectors \(Four registers\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	1	sz	1	0	0	1	0	1	0	Rv	1	1	1	Zm	0	0	0	S	off3					

Decode fields
sz S

Instruction Details

Feature

0	0	FADD	FEAT_SME_F16F16
0	1	FSUB	FEAT_SME_F16F16
1	0	BFADD	FEAT_SVE_B16B16
1	1	BFSUB	FEAT_SVE_B16B16

SME2 multiple vectors long long MLA four sources

These instructions are under [SME2 Multi-vector - Multiple Array Vectors \(Four registers\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	1	sz	1	Zm	0	1	0	Rv	0	0	0	Zn	0	0	U	S	op	0	01					

Decode fields				Instruction Details	Feature
sz	U	S	op		
	0	0	0	SMLALL (multiple vectors)	FEAT_SME2
	0	1	0	SMILSLL (multiple vectors)	FEAT_SME2
	1	0	0	UMLALL (multiple vectors)	FEAT_SME2
	1	1	0	UMLSLL (multiple vectors)	FEAT_SME2
0	0	0	1	USMLALL (multiple vectors)	FEAT_SME2
0	0	1	1	UNALLOCATED	-
0	1		1	UNALLOCATED	-
1			1	UNALLOCATED	-

SME2 multiple vectors ternary FP16 four registers

These instructions are under [SME2 Multi-vector - Multiple Array Vectors \(Four registers\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	1	sz	1	Zm	0	1	0	Rv	1	0	0	Zn	0	0	S	1	off3							

Decode fields		Instruction Details	Feature
sz	S		
0	0	FMLA (multiple vectors)	FEAT_SME_F16F16
0	1	FMLS (multiple vectors)	FEAT_SME_F16F16
1	0	BFMLA (multiple vectors)	FEAT_SVE_B16B16
1	1	BFMLS (multiple vectors)	FEAT_SVE_B16B16

SME2 multiple vectors four-way dot product four registers

These instructions are under [SME2 Multi-vector - Multiple Array Vectors \(Four registers\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	1	sz	1	Zm	0	1	0	Rv	1	0	1	Zn	0	0	U	0	off3							

Decode fields		Instruction Details	Feature
U			
0		SDOT (4-way, multiple vectors)	FEAT_SME2
1		UDOT (4-way, multiple vectors)	FEAT_SME2

SME2 multiple vectors ternary FP four registers

These instructions are under [SME2 Multi-vector - Multiple Array Vectors \(Four registers\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	1	sz	1	Zm	0	1	0	Rv	1	1	0	Zn	0	0	0	S	off3							

Decode fields S	Instruction Details	Feature
0	FMLA (multiple vectors)	FEAT_SME2
1	FMLS (multiple vectors)	FEAT_SME2

SME2 multiple vectors ternary int four registers

These instructions are under [SME2 Multi-vector - Multiple Array Vectors \(Four registers\)](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	1	sz	1	Zm	0	1	0	Rv	1	1	0	Zn	0	0	1	S	off3							

Decode fields S	Instruction Details	Feature
0	ADD (array results, multiple vectors)	FEAT_SME2
1	SUB (array results, multiple vectors)	FEAT_SME2

SME Memory

These instructions are under [SME encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1110000	op0	op1	op2	op3																					op4						

op0	op1	op2	op3	op4	Decode fields	Instruction details	Feature
0xx0				0xx	SME load array vector (elements)	-	
0xx1				0xx	SME store array vector (elements)	-	
0xxx				1xx	UNALLOCATED	-	
100x	00000	0	xx000	0xx	SME save and restore array	-	
100x	00000	0	xx000	1xx	UNALLOCATED	-	
100x	00000	0	xx!=000		UNALLOCATED	-	

100x	\neq 00000	0			UNALLOCATED	-
100x		1	00000	000	SME2 lookup table load/store	-
100x		1	00000	\neq 000	UNALLOCATED	-
100x		1	\neq 00000		UNALLOCATED	-
101x					UNALLOCATED	-
110x					UNALLOCATED	-
1110				0xx	LD1Q	FEAT_SME
1111				0xx	ST1Q	FEAT_SME
111x				1xx	UNALLOCATED	-

SME load array vector (elements)

These instructions are under [SME Memory](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
1	1	1	0	0	0	0	msz	0	Rm	V	Rs	Pg		Rn	0	opc																

Decode fields

msz

Instruction Details

Feature

00	LD1B (scalar plus scalar, tile slice)	FEAT_SME
01	LD1H (scalar plus scalar, tile slice)	FEAT_SME
10	LD1W (scalar plus scalar, tile slice)	FEAT_SME
11	LD1D (scalar plus scalar, tile slice)	FEAT_SME

SME store array vector (elements)

These instructions are under [SME Memory](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
1	1	1	0	0	0	0	msz	1	Rm	V	Rs	Pg		Rn	0	opc																

Decode fields

msz

Instruction Details

Feature

00	ST1B (scalar plus scalar, tile slice)	FEAT_SME
01	ST1H (scalar plus scalar, tile slice)	FEAT_SME
10	ST1W (scalar plus scalar, tile slice)	FEAT_SME
11	ST1D (scalar plus scalar, tile slice)	FEAT_SME

SME save and restore array

These instructions are under [SME Memory](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	0	0	0	0	1	0	0	op	0	0	0	0	0	Rv	0	0	0	Rn	0	imm4									

Decode fields op	Instruction Details	Feature
0	LDR (vector)	FEAT_SME
1	STR (vector)	FEAT_SME

SME2 lookup table load/store

These instructions are under [SME Memory](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	0	0	0	0	1	0	0	opc	1	0	0	0	0	Rn	0	0	0	opc2											

Decode fields opc	Decode fields opc2	Instruction Details	Feature
x0xxxx		UNALLOCATED	-
x10xxx		UNALLOCATED	-
x110xx		UNALLOCATED	-
x1110x		UNALLOCATED	-
x11110		UNALLOCATED	-
x11111	01	UNALLOCATED	-
x11111	1x	UNALLOCATED	-
011111	00	LDR (ZT0)	FEAT_SME2
111111	00	STR (ZT0)	FEAT_SME2

SVE encodings

These instructions are under the [top-level](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
op0	0010		op1			op2			op3																						

Decode fields op0	Decode fields op1	Decode fields op2	Decode fields op3	Instruction details	Feature
000	0xx0xxxx	x1xxxx		SVE Integer Multiply-Add - Predicated	-
000	0xx0xxxx	000xxx		SVE Integer Binary Arithmetic - Predicated	-

000	0xx0xxxx	001xxx		SVE Integer Reduction	-
000	0xx0xxxx	100xxx		SVE Bitwise Shift - Predicated	-
000	0xx0xxxx	101xxx		SVE Integer Unary Arithmetic - Predicated	-
000	0xx1xxxx	000xxx		SVE integer add/ subtract vectors (unpredicated)	-
000	0xx1xxxx	001xxx		SVE Bitwise Logical - Unpredicated	-
000	0xx1xxxx	0100xx		SVE Index Generation	-
000	0xx1xxxx	0101xx		SVE Stack Allocation	-
000	0xx1xxxx	011xxx		SVE2 Integer Multiply - Unpredicated	-
000	0xx1xxxx	100xxx		SVE Bitwise Shift - Unpredicated	-
000	0xx1xxxx	1010xx		SVE address generation	-
000	0xx1xxxx	1011xx		SVE Integer Misc - Unpredicated	-
000	0xx1xxxx	11xxxx		SVE Element Count	-
000	1xx00xxx			SVE Bitwise Immediate	-
000	1xx01xxx			SVE Integer Wide Immediate - Predicated	-
000	1xx1xxxx	001000		DUP (indexed)	-
000	1xx1xxxx	001001		SVE Permute Vector - One Source Quadwords	-
000	1xx1xxxx	00101x		SVE table lookup (three sources)	-
000	1xx1xxxx	001100		TBL â€“ SVE	-
000	1xx1xxxx	001101		TBXQ	FEAT_SVE2p1
000	1xx1xxxx	001110		SVE Permute Vector - Unpredicated	-
000	1xx1xxxx	001111		UNALLOCATED	-
000	1xx1xxxx	010xxx		SVE Permute Predicate	-
000	1xx1xxxx	011xxx		SVE permute vector elements	-
000	1xx1xxxx	10xxxx		SVE Permute Vector - Predicated	-
000	1xx1xxxx	11xxxx		SEL (vectors)	-
000	10x1xxxx	000xxx		SVE Permute Vector - Extract	-
000	11x1xxxx	000xxx		SVE Permute Vector - Segments	-
001	0xx0xxxx			SVE Integer Compare - Vectors	-

001	0xx1xxxx			SVE integer compare with unsigned immediate	-
001	1xx0xxxx0xxxx			SVE integer compare with signed immediate	-
001	1xx00xxx01xxxx			SVE predicate logical operations	-
001	1xx00xxx11xxxx			SVE Propagate Break	-
001	1xx01xxx01xxxx			SVE Partition Break	-
001	1xx01xxx11xxxx			SVE Predicate Misc	-
001	1xx1xxxx00xxxx			SVE Integer Compare - Scalars	-
001	1xx1xxxx01xxxx	0		SVE broadcast predicate element	-
001	1xx1xxxx01xxxx	1		SVE Scalar Integer Compare - Predicate-as-counter	-
001	1xx1xxxx11xxxx			SVE Integer Wide Immediate - Unpredicated	-
001	1xx100xxx10xxxx			SVE Predicate Count	-
001	1xx101xx1000xx			SVE Inc/Dec by Predicate Count	-
001	1xx101xx1001xx			SVE Write FFR	-
001	1xx101xx101xxx			UNALLOCATED	-
001	1xx11xxx10xxxx			UNALLOCATED	-
010	0xx0xxxx0xxxxx			SVE Integer Multiply-Add - Unpredicated	-
010	0xx0xxxx10xxxx			SVE2 Integer - Predicated	-
010	0xx0xxxx11000x			SVE integer clamp	-
010	0xx0xxxx1101xx			UNALLOCATED	-
010	0xx0xxxx111xxx			SVE permute vector elements (quadwords)	-
010	0xx1xxxx			SVE Multiply - Indexed	-
010	0x10xxxx11001x			UNALLOCATED	-
010	0000xxxx11001x			SVE two-way dot product	-
010	0100xxxx11001x			SVE two-way dot product (indexed)	-
010	1xx0xxxx0xxxxx			SVE2 Widening Integer Arithmetic	-
010	1xx0xxxx10xxxx			SVE Misc	-
010	1xx0xxxx11xxxx			SVE2 Accumulate	-
010	1xx1xxxx0xxxxx			SVE2 Narrowing	-

010	1xx1xxxx 00xxx		SVE2 character match	-
010	1xx1xxxx 01xxx		SVE2 Histogram Computation - Segment	-
010	1xx1xxxx 10xxx		HISTCNT	-
010	1xx1xxxx 11xxx		SVE2 Crypto Extensions	-
011	0xx0xxxx 0xxxxx		FCMLA (vectors)	-
011	0xx00x1 xxxxxxxx		UNALLOCATED	-
011	0xx0000 000xxx		FCADD	-
011	0xx0000 010xxx		UNALLOCATED	-
011	0xx0000 011xxx		UNALLOCATED	-
011	0xx0000 1xxxxx		UNALLOCATED	-
011	0xx0010 000xxx		UNALLOCATED	-
011	0xx0010 010xxx		SVE floating-point convert precision odd elements	-
011	0xx0010 11xxxx		UNALLOCATED	-
011	0xx010xx 000xxx		SVE2 floating-point pairwise operations	-
011	0xx010xx 010xxx		SVE floating-point recursive reduction (quadwords)	-
011	0xx010xx 11xxxx		UNALLOCATED	-
011	0xx011xx 1xxxxx		UNALLOCATED	-
011	0xx1xxxx 0000xx		SVE floating-point multiply-add (indexed)	-
011	0xx1xxxx 0001xx		SVE floating-point complex multiply-add (indexed)	-
011	0xx1xxxx 0010x0		SVE floating-point multiply (indexed)	-
011	0xx1xxxx 001001		SVE FP clamp	-
011	0xx1xxxx 001011		UNALLOCATED	-
011	0xx1xxxx 0011xx		UNALLOCATED	-
011	0xx1xxxx 01x0xx		SVE Floating Point Widening Multiply-Add - Indexed	-
011	0xx1xxxx 01x1xx		UNALLOCATED	-
011	0xx1xxxx 10x00x		SVE Floating Point Widening Multiply-Add	-
011	0xx1xxxx 10x01x		UNALLOCATED	-
011	0xx1xxxx 10x1xx		UNALLOCATED	-
011	0xx1xxxx 110xxx		UNALLOCATED	-
011	0xx1xxxx 111000		UNALLOCATED	-

011	0xx1xxxx	111001		SVE floating point matrix multiply accumulate	-
011	0xx1xxxx	11101x		UNALLOCATED	-
011	0xx1xxxx	1111xx		UNALLOCATED	-
011	1xx0xxxx	1xxxxx		SVE floating-point compare vectors	-
011	1xx0xxxx	000xxx		SVE floating-point arithmetic (unpredicated)	-
011	1xx0xxxx	100xxx		SVE Floating Point Arithmetic - Predicated	-
011	1xx0xxxx	101xxx		SVE Floating Point Unary Operations - Predicated	-
011	1xx000x	001xxx		SVE floating-point recursive reduction	-
011	1xx001x	0010xx		UNALLOCATED	-
011	1xx001x	0011xx		SVE Floating Point Unary Operations - Unpredicated	-
011	1xx010x	001xxx		SVE Floating Point Compare - with Zero	-
011	1xx011x	001xxx		SVE Floating Point Accumulating Reduction	-
011	1xx1xxxx			SVE Floating Point Multiply-Add	-
100				SVE Memory - 32-bit Gather and Unsized Contiguous	-
101				SVE Memory - Contiguous Load	-
110				SVE Memory - 64-bit Gather	-
111		0x0xxx		SVE Memory - Contiguous Store and Unsized Contiguous	-
111		001xxx		SVE Memory - Non-temporal and Quadword Scatter Store	-
111		011xxx		SVE Memory - Non-temporal and Multi-register Contiguous Store	-

111		1x0xxx		SVE Memory - Scatter with Optional Sign Extend	-
111		101xxx		SVE Memory - Scatter	-
111		111xxx		SVE Memory - Contiguous Store with Immediate Offset	-

SVE Integer Multiply-Add - Predicated

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
00000100								0								op0	1																

Decode fields op0	Instruction details
0	SVE integer multiply-accumulate writing addend (predicated)
1	SVE integer multiply-add writing multiplicand (predicated)

SVE integer multiply-accumulate writing addend (predicated)

These instructions are under [SVE Integer Multiply-Add - Predicated](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	0	size	0		Zm	0	1	op	Pg		Zn		Zda												

Decode fields op	Instruction Details
0	MLA (vectors)
1	MLS (vectors)

SVE integer multiply-add writing multiplicand (predicated)

These instructions are under [SVE Integer Multiply-Add - Predicated](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	0	size	0		Zm	1	1	op	Pg		Za		Zdn												

Decode fields op	Instruction Details
0	MAD

Decode fields	Instruction Details
op	
1	<u>MSB</u>

SVE Integer Binary Arithmetic - Predicated

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
00000100				0 op0				000																							

Decode fields	Instruction details
op0	
00x	SVE integer add/subtract vectors (predicated)
01x	SVE integer min/max/difference (predicated)
100	SVE integer multiply vectors (predicated)
101	SVE integer divide vectors (predicated)
11x	SVE bitwise logical operations (predicated)

SVE integer add/subtract vectors (predicated)

These instructions are under [SVE Integer Binary Arithmetic - Predicated](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	size	0	0	0	opc	0	0	0	Pq	Zm	Zdn
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---	------	---	---	---	-----	---	---	---	----	----	-----

Decode fields	Instruction Details
opc	
000	ADD (vectors, predicated)
001	SUB (vectors, predicated)
010	UNALLOCATED
011	SUBR (vectors)
1xx	UNALLOCATED

SVE integer min/max/difference (predicated)

These instructions are under [SVE Integer Binary Arithmetic - Predicated](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	size	0	0	1	opc	U	0	0	0	Pa	Zm	Zdn
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---	------	---	---	---	-----	---	---	---	---	----	----	-----

Decode fields		Instruction Details
opc	U	
00	0	SMAX (vectors)
00	1	UMAX (vectors)
01	0	SMIN (vectors)
01	1	UMIN (vectors)
10	0	SABD
10	1	UABD
11		UNALLOCATED

SVE integer multiply vectors (predicated)

These instructions are under [SVE Integer Binary Arithmetic - Predicated](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	0	size	0	1	0	0	H	U	0	0	0	Pg		Zm		Zdn									

Decode fields		Instruction Details
H	U	
0	0	MUL (vectors, predicated)
0	1	UNALLOCATED
1	0	SMULH (predicated)
1	1	UMULH (predicated)

SVE integer divide vectors (predicated)

These instructions are under [SVE Integer Binary Arithmetic - Predicated](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	0	size	0	1	0	1	R	U	0	0	0	Pg		Zm		Zdn									

Decode fields		Instruction Details
R	U	
0	0	SDIV
0	1	UDIV
1	0	SDIVR
1	1	UDIVR

SVE bitwise logical operations (predicated)

These instructions are under [SVE Integer Binary Arithmetic - Predicated](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	0	size	0	1	1	opc	0	0	0	Pg		Zm		Zdn											

Decode fields opc	Instruction Details
000	ORR (vectors, predicated)
001	EOR (vectors, predicated)
010	AND (vectors, predicated)
011	BIC (vectors, predicated)
1xx	UNALLOCATED

SVE Integer Reduction

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	0	size	0	0	0	op0			001																

Decode fields op0	Instruction details
000	SVE integer add reduction (predicated)
001	SVE integer add reduction (quadwords)
010	SVE integer min/max reduction (predicated)
011	SVE integer min/max reduction (quadwords)
10x	SVE constructive prefix (predicated)
110	SVE bitwise logical reduction (predicated)
111	SVE bitwise logical reduction (quadwords)

SVE integer add reduction (predicated)

These instructions are under [SVE Integer Reduction](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	0	size	0	0	0	op	U	0	0	1	Pg		Zn		Vd										

Decode fields op	U	Instruction Details
0	0	SADDV
0	1	UADDV

Decode fields		Instruction Details																	
op	U																		
1		UNALLOCATED																	

SVE integer add reduction (quadwords)

These instructions are under [SVE Integer Reduction](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	0	size	0	0	0	1	op	U	0	0	1	Pg		Zn		Vd									

Decode fields		Instruction Details																		Feature				
op	U																							
0	0	UNALLOCATED																		-				
0	1	ADDQV																		FEAT_SVE2p1				
1		UNALLOCATED																		-				

SVE integer min/max reduction (predicated)

These instructions are under [SVE Integer Reduction](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	0	size	0	0	1	0	op	U	0	0	1	Pg		Zn		Vd									

Decode fields		Instruction Details																	
op	U																		
0	0	SMAXV																	
0	1	UMAXV																	
1	0	SMINV																	
1	1	UMINV																	

SVE integer min/max reduction (quadwords)

These instructions are under [SVE Integer Reduction](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	0	size	0	0	1	1	op	U	0	0	1	Pg		Zn		Vd									

Decode fields		Instruction Details																		Feature				
op	U																							
0	0	SMAXQV																		FEAT_SVE2p1				
0	1	UMAXQV																		FEAT_SVE2p1				
1	0	SMINQV																		FEAT_SVE2p1				

Decode fields		Instruction Details	Feature
op	U		
1	1	UMINQV	FEAT_SVE2p1

SVE constructive prefix (predicated)

These instructions are under [SVE Integer Reduction](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	size	0	1	0	opc	M	0	0	1	Pq		Zn		Zd
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---	------	---	---	---	-----	---	---	---	---	----	--	----	--	----

Decode fields	Instruction Details
opc	
00	MOVPRFX (predicated)
01	UNALLOCATED
1x	UNALLOCATED

SVE bitwise logical reduction (predicated)

These instructions are under [SVE Integer Reduction](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	size	0	1	1	0	opc	0	0	1	Pq		Zn		Vd
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---	------	---	---	---	---	-----	---	---	---	----	--	----	--	----

Decode fields	Instruction Details
opc	
00	ORV
01	EORV
10	ANDV
11	UNALLOCATED

SVE bitwise logical reduction (quadwords)

These instructions are under [SVE Integer Reduction](#).

Decode fields	Instruction Details	Feature
opc		
00	<u>ORQV</u>	FEAT_SVE2p1
01	<u>EORQV</u>	FEAT_SVE2p1
10	<u>ANDQV</u>	FEAT_SVE2p1

Decode fields		Instruction Details				Feature	
opc							
11		UNALLOCATED			-		

SVE Bitwise Shift - Predicated

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
00000100					0	op0					100																				

Decode fields		Instruction details									
opc	op0	SVE bitwise shift by immediate (predicated)									
0x		SVE bitwise shift by vector (predicated)									
10		SVE bitwise shift by wide elements (predicated)									

SVE bitwise shift by immediate (predicated)

These instructions are under [SVE Bitwise Shift - Predicated](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	0	tszh	0	0	opc	L	U	1	0	0	Pg	tszl	imm3	Zdn											

Decode fields			Instruction Details																													
opc	L	U																														
00	0	0	ASR (immediate, predicated)																													
00	0	1	LSR (immediate, predicated)																													
00	1	0	UNALLOCATED																													
00	1	1	LSL (immediate, predicated)																													
01	0	0	ASRD																													
01	0	1	UNALLOCATED																													
01	1	0	SQSHL (immediate)																													
01	1	1	UQSHL (immediate)																													
10			UNALLOCATED																													
11	0	0	SRSHR																													
11	0	1	URSHR																													
11	1	0	UNALLOCATED																													
11	1	1	SQSHLU																													

SVE bitwise shift by vector (predicated)

These instructions are under [SVE Bitwise Shift - Predicated](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	0	size	0	1	0	R	L	U	1	0	0	Pg		Zm		Zdn									

Decode fields **Instruction Details**
R L U

	1	0	UNALLOCATED
0	0	0	ASR (vectors)
0	0	1	LSR (vectors)
0	1	1	LSL (vectors)
1	0	0	ASRR
1	0	1	LSRR
1	1	1	LSLR

SVE bitwise shift by wide elements (predicated)

These instructions are under [SVE Bitwise Shift - Predicated](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	0	size	0	1	1	R	L	U	1	0	0	Pg		Zm		Zdn									

Decode fields **Instruction Details**
R L U

0	0	0	ASR (wide elements, predicated)
0	0	1	LSR (wide elements, predicated)
0	1	0	UNALLOCATED
0	1	1	LSL (wide elements, predicated)
1			UNALLOCATED

SVE Integer Unary Arithmetic - Predicated

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	0	0	op0			0	op0			101															

Decode fields **Instruction details**
op0

0x	UNALLOCATED
10	SVE integer unary operations (predicated)
11	SVE bitwise unary operations (predicated)

SVE integer unary operations (predicated)

These instructions are under [SVE Integer Unary Arithmetic - Predicated](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	size	0	1	0	opc	1	0	1	Pg		Zn		Zd
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---	------	---	---	---	-----	---	---	---	----	--	----	--	----

Decode fields	Instruction Details
opc	
000	<u>SXTB, SXTH, SXTW</u> â€” <u>SXTB</u>
001	<u>UXTB, UXTH, UXTW</u> â€” <u>UXTB</u>
010	<u>SXTB, SXTH, SXTW</u> â€” <u>SXTH</u>
011	<u>UXTB, UXTH, UXTW</u> â€” <u>UXTH</u>
100	<u>SXTB, SXTH, SXTW</u> â€” <u>SXTW</u>
101	<u>UXTB, UXTH, UXTW</u> â€” <u>UXTW</u>
110	<u>ABS</u>
111	<u>NEG</u>

SVE bitwise unary operations (predicated)

These instructions are under [SVE Integer Unary Arithmetic - Predicated](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	size	0	1	1	opc	1	0	1	Pq		Zn		Zd
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---	------	---	---	---	-----	---	---	---	----	--	----	--	----

Decode fields	Instruction Details
opc	
000	<u>CLS</u>
001	<u>CLZ</u>
010	<u>CNT</u>
011	<u>CNOT</u>
100	<u>FABS</u>
101	<u>FNEG</u>
110	<u>NOT (vector)</u>
111	UNALLOCATED

SVE integer add/subtract vectors (unpredicated)

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	size	1	Zm	0	0	0	opc	Zn	Zd
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---	------	---	----	---	---	---	-----	----	----

Decode fields**opc****Instruction Details**

000	ADD (vectors, unpredicated)
001	SUB (vectors, unpredicated)
01x	UNALLOCATED
100	SQADD (vectors, unpredicated)
101	UQADD (vectors, unpredicated)
110	SQSUB (vectors, unpredicated)
111	UQSUB (vectors, unpredicated)

SVE Bitwise Logical - UnpredicatedThese instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	0	opc	1	Zm	0	0	1	1	0	0	Zn	Zd													

Decode fields**op0****Instruction details**

0xx	UNALLOCATED
100	SVE bitwise logical operations (unpredicated)
101	XAR
11x	SVE2 bitwise ternary operations

SVE bitwise logical operations (unpredicated)These instructions are under [SVE Bitwise Logical - Unpredicated](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	0	opc	1	Zm	0	0	1	1	0	0	Zn	Zd													

Decode fields**opc****Instruction Details**

00	AND (vectors, unpredicated)
01	ORR (vectors, unpredicated)
10	EOR (vectors, unpredicated)
11	BIC (vectors, unpredicated)

SVE2 bitwise ternary operationsThese instructions are under [SVE Bitwise Logical - Unpredicated](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	0	opc	1	Zm	0	0	1	1	1	0	2	Zk	Zdn												

Decode fields

opc	o2
-----	----

Instruction Details

00	0	EOR3
00	1	BSL
01	0	BCAX
01	1	BSL1N
1x	0	UNALLOCATED
10	1	BSL2N
11	1	NBSL

SVE Index Generation

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
00000100							1					0100		op0																	

Decode fields

op0

Instruction details

00	INDEX (immediates)
01	INDEX (scalar, immediate)
10	INDEX (immediate, scalar)
11	INDEX (scalars)

SVE Stack Allocation

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
00000100				op0	1							0101		op1																	

Decode fields

op0	op1
-----	-----

Instruction details

0	0	SVE stack frame adjustment
0	1	Streaming SVE stack frame adjustment
1	0	SVE stack frame size
1	1	Streaming SVE stack frame size

SVE stack frame adjustment

These instructions are under [SVE Stack Allocation](#).

Decode fields	Instruction Details
op	

0	<u>ADDVL</u>
1	<u>ADDPL</u>

Streaming SVE stack frame adjustment

These instructions are under [SVE Stack Allocation](#).

Decode fields	Instruction Details	Feature
op		

0	<u>ADDSVL</u>	FEAT_SME
1	<u>ADDSPL</u>	FEAT_SME

SVE stack frame size

These instructions are under [SVE Stack Allocation](#).

Decode fields		Instruction Details
op	opc2	
0	0xxxx	UNALLOCATED
0	10xxx	UNALLOCATED
0	110xx	UNALLOCATED
0	1110x	UNALLOCATED
0	11110	UNALLOCATED
0	11111	<u>RDVL</u>
1		UNALLOCATED

Streaming SVE stack frame size

These instructions are under [SVE Stack Allocation](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	0	1	op	1		opc2		0	1	0	1	1		imm6						Rd					

Decode fields		Instruction Details	Feature
op	opc2		
0	0xxxx	UNALLOCATED	-
0	10xxx	UNALLOCATED	-
0	110xx	UNALLOCATED	-
0	1110x	UNALLOCATED	-
0	11110	UNALLOCATED	-
0	11111	RDSVL	FEAT_SME
1		UNALLOCATED	-

SVE2 Integer Multiply - Unpredicated

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Decode fields		Instruction details
op0		
0x		SVE2 integer multiply vectors (unpredicated)
10		SVE2 signed saturating doubling multiply high (unpredicated)
11		UNALLOCATED

SVE2 integer multiply vectors (unpredicated)

These instructions are under [SVE2 Integer Multiply - Unpredicated](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	0	size	1		Zm	0	1	1	0	opc		Zn		Zd											

Decode fields		Instruction Details
size	opc	
	00	MUL (vectors, unpredicated)
	10	SMULH (unpredicated)
	11	UMULH (unpredicated)
00	01	PMUL
01	01	UNALLOCATED
1x	01	UNALLOCATED

SVE2 signed saturating doubling multiply high (unpredicated)

These instructions are under [SVE2 Integer Multiply - Unpredicated](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	size	1	Zm	0	1	1	1	0	R	Zn	Zd
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---	------	---	----	---	---	---	---	---	---	----	----

Decode fields	Instruction Details
R	SODMULH (vectors)
0	SQRDMULH (vectors)
1	SQRDMULH (vectors)

SVE Bitwise Shift - Unpredicated

These instructions are under [SVE encodings](#).

Decode fields	Instruction details
op0	SVE bitwise shift by wide elements (unpredicated)
1	SVE bitwise shift by immediate (unpredicated)

SVE bitwise shift by wide elements (unpredicated)

These instructions are under [SVE Bitwise Shift - Unpredicated](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	size	1	Zm	1	0	0	0	opc	Zn	Zd
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---	------	---	----	---	---	---	---	-----	----	----

Decode fields	Instruction Details
opc	
00	ASR (wide elements, unpredicated)
01	LSR (wide elements, unpredicated)
10	UNALLOCATED
11	LSL (wide elements, unpredicated)

SVE bitwise shift by immediate (unpredicated)

These instructions are under [SVE Bitwise Shift - Unpredicated](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	1	0	0	tszh	1	tszl	imm3	1	0	0	1	opc	Zn		Zd
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	------	---	------	------	---	---	---	---	-----	----	--	----

Decode fields		Instruction Details
opc		
00		ASR (immediate, unpredicated)
01		LSR (immediate, unpredicated)
10		UNALLOCATED
11		LSL (immediate, unpredicated)

SVE address generation

These instructions are under [SVE encodings](#).

Decode fields	Instruction Details
opc	
00	ADR â€“ Unpacked 32-bit signed offsets
01	ADR â€“ Unpacked 32-bit unsigned offsets
1x	ADR â€“ Packed offsets

SVE Integer Misc - Unpredicated

These instructions are under [SVE encodings](#).

Decode fields	Instruction details
op0	
0x	SVE floating-point trig select coefficient
10	SVE floating-point exponential accelerator
11	SVE constructive prefix (unpredicated)

SVE floating-point trig select coefficient

These instructions are under [SVE Integer Misc - Unpredicated](#).

Decode fields		Instruction Details
opc		

0	FTSSEL	
1	UNALLOCATED	

SVE floating-point exponential accelerator

These instructions are under [SVE Integer Misc - Unpredicated](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	0	size	1	opc		1	0	1	1	1	0		Zn		Zd										

Decode fields		Instruction Details
opc		

00000	FEXPA	
00001	UNALLOCATED	
0001x	UNALLOCATED	
001xx	UNALLOCATED	
01xxx	UNALLOCATED	
1xxxx	UNALLOCATED	

SVE constructive prefix (unpredicated)

These instructions are under [SVE Integer Misc - Unpredicated](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	0	opc	1	opc2		1	0	1	1	1	1		Zn		Zd										

Decode fields		Instruction Details
opc	opc2	

00	00000	MOVPRFX (unpredicated)
00	00001	UNALLOCATED
00	0001x	UNALLOCATED
00	001xx	UNALLOCATED
00	01xxx	UNALLOCATED
00	1xxxx	UNALLOCATED
01		UNALLOCATED
1x		UNALLOCATED

SVE Element Count

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
00000100							1	op0				11		op1																	

Decode fields		Instruction details	
op0	op1		
0	00x	SVE saturating inc/dec vector by element count	
0	100	SVE element count	
0	101	UNALLOCATED	
1	000	SVE inc/dec vector by element count	
1	100	SVE inc/dec register by element count	
1	x01	UNALLOCATED	
	01x	UNALLOCATED	
	11x	SVE saturating inc/dec register by element count	

SVE saturating inc/dec vector by element count

These instructions are under [SVE Element Count](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	0	size	1	0	imm4	1	1	0	0	D	U	pattern		Zdn											

Decode fields			Instruction Details	
size	D	U		
00			UNALLOCATED	
01	0	0	SQINCH (vector)	
01	0	1	UQINCH (vector)	
01	1	0	SQDECW (vector)	
01	1	1	UQDECW (vector)	
10	0	0	SQINCW (vector)	
10	0	1	UQINCW (vector)	
10	1	0	SQDECW (vector)	
10	1	1	UQDECW (vector)	
11	0	0	SQINCD (vector)	
11	0	1	UQINCD (vector)	
11	1	0	SQDECW (vector)	
11	1	1	UQDECW (vector)	

SVE element count

These instructions are under [SVE Element Count](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	0	size	1	0	imm4	1	1	1	0	0	op	pattern	Rd												

Decode fields

size op

Instruction Details

	1	UNALLOCATED
00	0	CNTB, CNTD, CNTH, CNTW â€“ CNTB
01	0	CNTB, CNTD, CNTH, CNTW â€“ CNTH
10	0	CNTB, CNTD, CNTH, CNTW â€“ CNTW
11	0	CNTB, CNTD, CNTH, CNTW â€“ CNTD

SVE inc/dec vector by element count

These instructions are under [SVE Element Count](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	0	size	1	1	imm4	1	1	0	0	0	D	pattern	Zdn												

Decode fields

size D

Instruction Details

00		UNALLOCATED
01	0	INCD, INCH, INCW (vector) â€“ INCH
01	1	DECD, DECH, DECW (vector) â€“ DECH
10	0	INCD, INCH, INCW (vector) â€“ INCW
10	1	DECD, DECH, DECW (vector) â€“ DECW
11	0	INCD, INCH, INCW (vector) â€“ INCD
11	1	DECD, DECH, DECW (vector) â€“ DECD

SVE inc/dec register by element count

These instructions are under [SVE Element Count](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	0	size	1	1	imm4	1	1	1	0	0	D	pattern	Rdn												

Decode fields

size D

Instruction Details

00	0	INCB, INCD, INCH, INCW (scalar) â€“ INCB
00	1	DECB, DECD, DECH, DECW (scalar) â€“ DECB
01	0	INCB, INCD, INCH, INCW (scalar) â€“ INCH
01	1	DECB, DECD, DECH, DECW (scalar) â€“ DECH

Decode fields		Instruction Details
size	D	
10	0	INCB , INCD , INCH , INCW (scalar) â€” INCW
10	1	DECB , DECD , DECH , DECW (scalar) â€” DECW
11	0	INCB , INCD , INCH , INCW (scalar) â€” INCD
11	1	DECB , DECD , DECH , DECW (scalar) â€” DECD

SVE saturating inc/dec register by element count

These instructions are under [SVE Element Count](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	0	size	1	sf		imm4		1	1	1	1	D	U		pattern		Rdn								

Decode fields				Instruction Details
size	sf	D	U	
00	0	0	0	SQINCB â€” 32-bit
00	0	0	1	UQINCB â€” 32-bit
00	0	1	0	SQDECB â€” 32-bit
00	0	1	1	UQDECB â€” 32-bit
00	1	0	0	SQINCB â€” 64-bit
00	1	0	1	UQINCB â€” 64-bit
00	1	1	0	SQDECB â€” 64-bit
00	1	1	1	UQDECB â€” 64-bit
01	0	0	0	SQINCH (scalar) â€” 32-bit
01	0	0	1	UQINCH (scalar) â€” 32-bit
01	0	1	0	SQDECH (scalar) â€” 32-bit
01	0	1	1	UQDECH (scalar) â€” 32-bit
01	1	0	0	SQINCH (scalar) â€” 64-bit
01	1	0	1	UQINCH (scalar) â€” 64-bit
01	1	1	0	SQDECH (scalar) â€” 64-bit
01	1	1	1	UQDECH (scalar) â€” 64-bit
10	0	0	0	SQINCW (scalar) â€” 32-bit
10	0	0	1	UQINCW (scalar) â€” 32-bit
10	0	1	0	SQDECW (scalar) â€” 32-bit
10	0	1	1	UQDECW (scalar) â€” 32-bit
10	1	0	0	SQINCW (scalar) â€” 64-bit
10	1	0	1	UQINCW (scalar) â€” 64-bit
10	1	1	0	SQDECW (scalar) â€” 64-bit
10	1	1	1	UQDECW (scalar) â€” 64-bit
11	0	0	0	SQINCD (scalar) â€” 32-bit

Decode fields				Instruction Details
size	sf	D	U	
11	0	0	1	UQINCD (scalar) â€“ 32-bit
11	0	1	0	SQDECD (scalar) â€“ 32-bit
11	0	1	1	UQDECD (scalar) â€“ 32-bit
11	1	0	0	SQINCD (scalar) â€“ 64-bit
11	1	0	1	UQINCD (scalar) â€“ 64-bit
11	1	1	0	SQDECD (scalar) â€“ 64-bit
11	1	1	1	UQDECD (scalar) â€“ 64-bit

SVE Bitwise Immediate

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
00000101		op0	00	op1																											

Decode fields		Instruction details
op0	op1	
11	00	DUPM
!= 11	00	SVE bitwise logical with immediate (unpredicated)
	!= 00	UNALLOCATED

SVE bitwise logical with immediate (unpredicated)

These instructions are under [SVE Bitwise Immediate](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	1	!	=	11	0	0	0	0		imm13						Zdn									

The following constraints also apply to this encoding: opc != 11 && opc != 11

Decode fields		Instruction Details
opc		
00		ORR (immediate)
01		EOR (immediate)
10		AND (immediate)

SVE Integer Wide Immediate - Predicated

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
00000101								01									op0															

Decode fields	Instruction details
op0	

0xx	SVE copy integer immediate (predicated)
10x	UNALLOCATED
110	FCPY
111	UNALLOCATED

SVE copy integer immediate (predicated)

These instructions are under [SVE Integer Wide Immediate - Predicated](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	1	size	0	1	Pg	0	M	sh			imm8					Zd									

Decode fields	Instruction Details
M	

0	CPY (immediate, zeroing)
1	CPY (immediate, merging)

SVE Permute Vector - One Source Quadwords

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
00000101					op0	1	op1					001001																			

Decode fields	Instruction details	Feature
op0		
op1		

00		DUPQ	FEAT_SVE2p1
01	0	EXTQ	FEAT_SVE2p1
01	1	UNALLOCATED	-
1x		UNALLOCATED	-

SVE table lookup (three sources)

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	1	size	1	Zm	0	0	1	0	1	op	Zn	Zd													

Decode fields

op

0	TBL
1	TBX

Instruction Details

SVE Permute Vector - Unpredicated

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
00000101		1	op0	op1		001110	op2		op3																						

Decode fields

op0 op1 op2 op3

Instruction details

00	000			DUP (scalar)
00	100			INSR (scalar)
00	x10			UNALLOCATED
00	xx1			UNALLOCATED
01	xx0		0	SVE move predicate from vector
01	xx0		1	UNALLOCATED
01	xx1	0		SVE move predicate into vector
01	xx1	1		UNALLOCATED
10	0xx			SVE unpack vector elements
10	100			INSR (SIMD&FP scalar)
10	110			UNALLOCATED
10	1x1			UNALLOCATED
11	000			REV (vector)
11	!= 000			UNALLOCATED

SVE move predicate from vector

These instructions are under [SVE Permute Vector - Unpredicated](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	1	opc	1	0	1	opc2	0	0	0	1	1	1	0	Zn	0	Pd									

Decode fields

opc opc2

Instruction Details

Feature

00	00	UNALLOCATED	-
----	----	-------------	---

Decode fields		Instruction Details	Feature
opc	opc2		
00	01	PMOV (to predicate) â€“ byte	FEAT_SVE2p1
00	1x	PMOV (to predicate) â€“ halfword	FEAT_SVE2p1
01		PMOV (to predicate) â€“ word	FEAT_SVE2p1
1x		PMOV (to predicate) â€“ doubleword	FEAT_SVE2p1

SVE move predicate into vector

These instructions are under [SVE Permute Vector - Unpredicated](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	1	opc	1	0	1	opc2	1	0	0	1	1	1	0	0	0	Pn		Zd							

Decode fields		Instruction Details	Feature
opc	opc2		
00	00	UNALLOCATED	-
00	01	PMOV (to vector) \times byte	FEAT_SVE2p1
00	1x	PMOV (to vector) \times halfword	FEAT_SVE2p1
01		PMOV (to vector) \times word	FEAT_SVE2p1
1x		PMOV (to vector) \times doubleword	FEAT_SVE2p1

SVE unpack vector elements

These instructions are under [SVE Permute Vector - Unpredicated](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	1	size	1	1	0	0	U	H	0	0	1	1	1	0	Zn			Zd							

Decode fields		Instruction Details
U	H	
0	0	SUNPKHI , SUNPKLO â€” SUNPKLO
0	1	SUNPKHI , SUNPKLO â€” SUNPKHI
1	0	UUNPKHI , UUNPKLO â€” UUNPKLO
1	1	UUNPKHI , UUNPKLO â€” UUNPKHI

SVE Permute Predicate

These instructions are under [SVE encodings](#).

Decode fields				Instruction details
op0	op1	op2	op3	
00	1000x	0000	0	SVE unpack predicate elements
01	1000x	0000	0	UNALLOCATED
10	1000x	0000	0	UNALLOCATED
11	1000x	0000	0	UNALLOCATED
	0xxxx	xxx0	0	SVE permute predicate elements
	0xxxx	xxx1	0	UNALLOCATED
	10100	0000	0	REV (predicate)
	10101	0000	0	UNALLOCATED
	10x0x	1000	0	UNALLOCATED
	10x0x	x100	0	UNALLOCATED
	10x0x	xx10	0	UNALLOCATED
	10x0x	xxx1	0	UNALLOCATED
	10x1x		0	UNALLOCATED
	11xxx		0	UNALLOCATED
			1	UNALLOCATED

SVE unpack predicate elements

These instructions are under [SVE Permute Predicate](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	1	0	0	1	0	0	0	H	0	1	0	0	0	0	0	Pn	0	Pd							

Decode fields		Instruction Details
H		
0		PUNPKHI, PUNPKLO â€“ PUNPKLO
1		PUNPKHI, PUNPKLO â€“ PUNPKHI

SVE permute predicate elements

These instructions are under [SVE Permute Predicate](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	1	size	1	0	Pm	0	1	0	opc	H	0	Pn	0	Pd											

Decode fields		Instruction Details
opc		
00	0	ZIP1, ZIP2 (predicates) â€“ ZIP1
00	1	ZIP1, ZIP2 (predicates) â€“ ZIP2

Decode fields		Instruction Details
opc	H	
01	0	UZP1, UZP2 (predicates) â€“ UZP1
01	1	UZP1, UZP2 (predicates) â€“ UZP2
10	0	TRN1, TRN2 (predicates) â€“ TRN1
10	1	TRN1, TRN2 (predicates) â€“ TRN2
11		UNALLOCATED

SVE permute vector elements

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	1	size	1	Zm	0	1	1	opc		Zn		Zd													

Decode fields		Instruction Details
opc		
000		ZIP1, ZIP2 (vectors) â€“ ZIP1
001		ZIP1, ZIP2 (vectors) â€“ ZIP2
010		UZP1, UZP2 (vectors) â€“ UZP1
011		UZP1, UZP2 (vectors) â€“ UZP2
100		TRN1, TRN2 (vectors) â€“ TRN1
101		TRN1, TRN2 (vectors) â€“ TRN2
11x		UNALLOCATED

SVE Permute Vector - Predicated

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	1	op0	op1	op2	10	op3																			

Decode fields				Instruction details
op0	op1	op2	op3	
0	000	0	0	CPY (SIMD&FP scalar)
0	000	1	0	COMPACT
0	000		1	SVE extract element to general register
0	001		0	SVE extract element to SIMD&FP scalar register
0	01x		0	SVE reverse within elements
0	01x		1	UNALLOCATED
0	100	0	1	CPY (scalar)
0	100	1	1	UNALLOCATED

0	100		0	SVE conditionally broadcast element to vector
0	101		0	SVE conditionally extract element to SIMD&FP scalar
0	110	0	0	SPICE à€” Destructive
0	110	1	0	SPICE à€” Constructive
0	110		1	UNALLOCATED
0	111	0	0	SVE reverse doublewords
0	111	0	1	UNALLOCATED
0	111	1		UNALLOCATED
0	x01		1	UNALLOCATED
1	000		0	UNALLOCATED
1	000		1	SVE conditionally extract element to general register
1	!= 000			UNALLOCATED

SVE extract element to general register

These instructions are under [SVE Permute Vector - Predicated](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	1	size	1	0	0	0	0	B	1	0	1	Pg		Zn		Rd									

Decode fields B	Instruction Details
0	LASTA (scalar)
1	LASTB (scalar)

SVE extract element to SIMD&FP scalar register

These instructions are under [SVE Permute Vector - Predicated](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	1	size	1	0	0	0	1	B	1	0	0	Pg		Zn		Vd									

Decode fields B	Instruction Details
0	LASTA (SIMD&FP scalar)
1	LASTB (SIMD&FP scalar)

SVE reverse within elements

These instructions are under [SVE Permute Vector - Predicated](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	1	size	1	0	0	1	opc	1	0	0	Pg		Zn		Zd										

Decode fields

Instruction Details

opc

00	REV_B, REV_H, REV_W â€“ REV_B
01	REV_B, REV_H, REV_W â€“ REV_H
10	REV_B, REV_H, REV_W â€“ REV_W
11	RBIT

SVE conditionally broadcast element to vector

These instructions are under [SVE Permute Vector - Predicated](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	1	size	1	0	1	0	0	B	1	0	0	Pg		Zm		Zdn									

Decode fields

Instruction Details

B

0	CLASTA (vectors)
1	CLASTB (vectors)

SVE conditionally extract element to SIMD&FP scalar

These instructions are under [SVE Permute Vector - Predicated](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	1	size	1	0	1	0	1	B	1	0	0	Pg		Zm		Vdn									

Decode fields

Instruction Details

B

0	CLASTA (SIMD&FP scalar)
1	CLASTB (SIMD&FP scalar)

SVE reverse doublewords

These instructions are under [SVE Permute Vector - Predicated](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	1	size	1	0	1	1	1	0	1	0	0	Pg		Zn		Zd									

Decode fields size	Instruction Details	Feature
00	REVD	FEAT_SVE2p1
01	UNALLOCATED	-
1x	UNALLOCATED	-

SVE conditionally extract element to general register

These instructions are under [SVE Permute Vector - Predicated](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	1	size	1	1	0	0	0	B	1	0	1	Pg	Zm	Rdn											

Decode fields B	Instruction Details
0	CLASTA (scalar)
1	CLASTB (scalar)

SVE Permute Vector - Extract

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000001010	op0	1																													

Decode fields op0	Instruction details
0	EXT â€“ Destructive
1	EXT â€“ Constructive

SVE Permute Vector - Segments

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000001011	op0	1																													

Decode fields op0	Instruction details
0	SVE permute vector segments
1	UNALLOCATED

SVE permute vector segments

These instructions are under [SVE Permute Vector - Segments](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	1	1	0	1	Zm	0	0	0	opc	H		Zn		Zd											

Decode fields		Instruction Details		Feature
opc	H			
00	0	ZIP1, ZIP2 (vectors)	" ZIP1	FEAT_F64MM
00	1	ZIP1, ZIP2 (vectors)	" ZIP2	FEAT_F64MM
01	0	UZP1, UZP2 (vectors)	" UZP1	FEAT_F64MM
01	1	UZP1, UZP2 (vectors)	" UZP2	FEAT_F64MM
10		UNALLOCATED		-
11	0	TRN1, TRN2 (vectors)	" TRN1	FEAT_F64MM
11	1	TRN1, TRN2 (vectors)	" TRN2	FEAT_F64MM

SVE Integer Compare - Vectors

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
00100100											0				op0																

Decode fields		Instruction details	
op0			
0		SVE integer compare vectors	
1		SVE integer compare with wide elements	

SVE integer compare vectors

These instructions are under [SVE Integer Compare - Vectors](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	0	0	1	0	0	size	0	Zm	op	0	o2	Pg		Zn	ne	Pd													

Decode fields			Instruction Details	
op	o2	ne		
0	0	0	CMP<cc> (vectors)	" CMPHS
0	0	1	CMP<cc> (vectors)	" CMPI
0	1	0	CMP<cc> (wide elements)	" CMPEQ
0	1	1	CMP<cc> (wide elements)	" CMPNE
1	0	0	CMP<cc> (vectors)	" CMPGE
1	0	1	CMP<cc> (vectors)	" CMPGT
1	1	0	CMP<cc> (vectors)	" CMPEQ
1	1	1	CMP<cc> (vectors)	" CMPNE

SVE integer compare with wide elements

These instructions are under [SVE Integer Compare - Vectors](#).

Decode fields			Instruction Details
U	lt	ne	
0	0	0	CMP<cc> (wide elements) â€“ CMPGE
0	0	1	CMP<cc> (wide elements) â€“ CMPGT
0	1	0	CMP<cc> (wide elements) â€“ CMPLT
0	1	1	CMP<cc> (wide elements) â€“ CMPLE
1	0	0	CMP<cc> (wide elements) â€“ CMPHS
1	0	1	CMP<cc> (wide elements) â€“ CMPHI
1	1	0	CMP<cc> (wide elements) â€“ CMPLO
1	1	1	CMP<cc> (wide elements) â€“ CMPLS

SVE integer compare with unsigned immediate

These instructions are under [SVE encodings](#).

Decode fields		Instruction Details
lt	ne	
0	0	CMP<cc> (immediate) â€“ CMPHIS
0	1	CMP<cc> (immediate) â€“ CMPHI
1	0	CMP<cc> (immediate) â€“ CMPLO
1	1	CMP<cc> (immediate) â€“ CMPLS

SVE integer compare with signed immediate

These instructions are under [SVE encodings](#).

Decode fields			Instruction Details
op	o2	ne	
0	0	0	<u>CMP<cc></u> (immediate) → <u>CMPGE</u>

Decode fields			Instruction Details
op	o2	ne	
0	0	1	CMP<cc> (immediate) â€“ CMPGT
0	1	0	CMP<cc> (immediate) â€“ CMPLT
0	1	1	CMP<cc> (immediate) â€“ CMPLE
1	0	0	CMP<cc> (immediate) â€“ CMPEQ
1	0	1	CMP<cc> (immediate) â€“ CMPNE
1	1		UNALLOCATED

SVE predicate logical operations

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	0	0	1	0	1	op	S	0	0	Pm	0	1	Pg	o2	Pn	o3	Pd												

Decode fields				Instruction Details
op	S	o2	o3	
0	0	0	0	AND (predicates)
0	0	0	1	BIC (predicates)
0	0	1	0	EOR (predicates)
0	0	1	1	SEL (predicates)
0	1	0	0	ANDS
0	1	0	1	BICS
0	1	1	0	EORS
0	1	1	1	UNALLOCATED
1	0	0	0	ORR (predicates)
1	0	0	1	ORN (predicates)
1	0	1	0	NOR
1	0	1	1	NAND
1	1	0	0	ORRS
1	1	0	1	ORNS
1	1	1	0	NORS
1	1	1	1	NANDS

SVE Propagate Break

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
00100101			00			11		op0																							

Decode fields		Instruction details

op0	
0	SVE propagate break from previous partition
1	UNALLOCATED

SVE propagate break from previous partition

These instructions are under [SVE Propagate Break](#).

Decode fields			Instruction Details
op	S	B	
0	0	0	BRKPA
0	0	1	BRKPB
0	1	0	BRKPAS
0	1	1	BRKPBS
1			UNALLOCATED

SVE Partition Break

These instructions are under [SVE encodings](#).

Decode fields				Instruction details
op0	op1	op2	op3	
0	1000	0	0	SVE propagate break to next partition
0	1000	0	1	UNALLOCATED
0	x000	1		UNALLOCATED
0	x1xx			UNALLOCATED
0	xx1x			UNALLOCATED
0	xxx1			UNALLOCATED
1	0000	1		UNALLOCATED
1	!= 0000			UNALLOCATED
	0000	0		SVE partition break condition

SVE propagate break to next partition

These instructions are under [SVE Partition Break](#).

Decode fields

S

Instruction Details

0	BRKN
1	BRKNS

SVE partition break condition

These instructions are under [SVE Partition Break](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	0	0	1	0	1	B	S	0	1	0	0	0	0	0	1	Pg	0	Pn	M	M	Pd								

Decode fields

B S M

Instruction Details

	1	1	UNALLOCATED
0	0		BRKA
0	1	0	BRKAS
1	0		BRKB
1	1	0	BRKBS

SVE Predicate Misc

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
00100101			01		op0		11		op1		op2		op3		op4																

Decode fields

op0

op1

op2

op3

op4

Instruction details

0000		x0		0	SVE predicate test
0100		x0		0	UNALLOCATED
0x10		x0		0	UNALLOCATED
0xx1		x0		0	UNALLOCATED
0xxx		x1		0	UNALLOCATED
1000	000	00		0	SVE predicate first active
1000	000	!= 00		0	UNALLOCATED
1000	100	10	0000	0	SVE predicate zero
1000	100	10	!= 0000	0	UNALLOCATED
1000	110	00		0	SVE predicate read from FFR (predicated)
1001	000	0x		0	UNALLOCATED

1001	000	10		0	PNEXT
1001	000	11		0	UNALLOCATED
1001	100	10		0	UNALLOCATED
1001	110	00	0000	0	SVE predicate read from FFR (unpredicated)
1001	110	00	!= 0000	0	UNALLOCATED
100x	010			0	UNALLOCATED
100x	100	0x		0	SVE predicate initialize
100x	100	11		0	UNALLOCATED
100x	110	!= 00		0	UNALLOCATED
100x	xx1			0	UNALLOCATED
110x				0	UNALLOCATED
1x1x				0	UNALLOCATED
				1	UNALLOCATED

SVE predicate test

These instructions are under [SVE Predicate Misc.](#)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	0	0	1	0	1	op	S	0	1	0	0	0	0	1	1	Pg	0	Pn	0	opc2									

Decode fields			Instruction Details
op	S	opc2	
0	0		UNALLOCATED
0	1	0000	PTEST
0	1	0001	UNALLOCATED
0	1	001x	UNALLOCATED
0	1	01xx	UNALLOCATED
0	1	1xxx	UNALLOCATED
1			UNALLOCATED

SVE predicate first active

These instructions are under [SVE Predicate Misc.](#)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	0	0	1	0	1	op	S	0	1	1	0	0	0	1	1	0	0	0	0	Pg	0	Pdn							

Decode fields		Instruction Details
op	S	
0	0	UNALLOCATED
0	1	PFIRST
1		UNALLOCATED

SVE predicate zero

These instructions are under [SVE Predicate Misc.](#)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	0	0	1	0	1	op	S	0	1	1	0	0	0	1	1	1	0	0	1	0	0	0	0	0	0	Pd			

Decode fields		Instruction Details
op	S	
0	0	PFALSE
0	1	UNALLOCATED
1		UNALLOCATED

SVE predicate read from FFR (predicated)

These instructions are under [SVE Predicate Misc.](#)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	0	0	1	0	1	op	S	0	1	1	0	0	0	1	1	1	0	0	0	0	Pg	0	Pd						

Decode fields		Instruction Details
op	S	
0	0	RDFFR (predicated)
0	1	RDFFRS
1		UNALLOCATED

SVE predicate read from FFR (unpredicated)

These instructions are under [SVE Predicate Misc.](#)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	0	0	1	0	1	op	S	0	1	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	Pd				

Decode fields		Instruction Details
op	S	
0	0	RDFFR (unpredicated)

Decode fields		Instruction Details
op	S	
0	1	UNALLOCATED
1		UNALLOCATED

SVE predicate initialize

These instructions are under [SVE Predicate Misc.](#)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	0	0	1	0	1	size	0	1	1	0	0	S	1	1	1	0	0	0	pattern	0	Pd								

Decode fields		Instruction Details
S		
0		PTRUE (predicate)
1		PTRUES

SVE Integer Compare - Scalars

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0	0	1	0	0	1	0	1			1				00	op0	op1														op2		

Decode fields			Instruction details
op0	op1	op2	
0x			SVE integer compare scalar count and limit
10	00	0000	SVE conditionally terminate scalars
10	00	!= 0000	UNALLOCATED
11	00		SVE pointer conflict compare
1x	!= 00		UNALLOCATED

SVE integer compare scalar count and limit

These instructions are under [SVE Integer Compare - Scalars](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0	0	1	0	0	1	0	1	size	1		Rm	0	0	0	sf	U	lt					Rn	eq	Pd								

Decode fields			Instruction Details
U	lt	eq	
0	0	0	WHILEGE (predicate)
0	0	1	WHILEGT (predicate)
0	1	0	WHILELT (predicate)

Decode fields			Instruction Details
U	lt	eq	
0	1	1	WHILELE (predicate)
1	0	0	WHILEHS (predicate)
1	0	1	WHILEHI (predicate)
1	1	0	WHILELO (predicate)
1	1	1	WHILELS (predicate)

SVE conditionally terminate scalars

These instructions are under [SVE Integer Compare - Scalars](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	0	0	1	0	1	op	sz	1	Rm	0	0	1	0	0	0	Rn	ne	0	0	0	0	0	0	0	0	0	0		

Decode fields		Instruction Details
op	ne	
0		UNALLOCATED
1	0	CTERMEQ, CTERMNE â€“ CTERMEQ
1	1	CTERMEQ, CTERMNE â€“ CTERMNE

SVE pointer conflict compare

These instructions are under [SVE Integer Compare - Scalars](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	0	0	1	0	1	size	1	Rm	0	0	1	1	0	0	Rn	rw	Pd												

Decode fields		Instruction Details
rw		
0		WHILEWR
1		WHILERW

SVE broadcast predicate element

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	0	0	1	0	1	i1	tszh	1	tszl	Rv	0	1	Pn	S	Pm	0	Pd												

Decode fields	Instruction Details	Feature
S		
0	PSEL	FEAT_SVE2p1

Decode fields		Instruction Details				Feature
S						
1		UNALLOCATED	-			

SVE Scalar Integer Compare - Predicate-as-counter

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
00100101				1		op0		01		op1													op2		1	op3					

op0	op1	op2	op3	Decode fields	Instruction details	Feature
00000	110			SVE extract mask predicate from predicate-as-counter	-	
00000	111	000000	0	PTRUE (predicate as counter)	FEAT_SVE2p1	
00000	111	000000	1	UNALLOCATED	-	
00000	111	!= 000000		UNALLOCATED	-	
!= 00000	11x			UNALLOCATED	-	
	01x			SVE integer compare scalar count and limit (predicate pair)	-	
	x0x			SVE integer compare scalar count and limit (predicate-as-counter)	-	

SVE extract mask predicate from predicate-as-counter

These instructions are under [SVE Scalar Integer Compare - Predicate-as-counter](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0 0 1 0 0 1 0 1	size	1	0	0	0	0	0	0	0	0	1	1	1	0		opc		PNn	1	Pd											

Decode fields	Instruction Details	Feature
opc		
0xx	PEXT (predicate)	FEAT_SVE2p1
10x	PEXT (predicate pair)	FEAT_SVE2p1
11x	UNALLOCATED	-

SVE integer compare scalar count and limit (predicate pair)

These instructions are under [SVE Scalar Integer Compare - Predicate-as-counter](#).

Decode fields			Instruction Details	Feature
U	lt	eq		
0	0	0	WHILEGE (predicate pair)	FEAT_SVE2p1
0	0	1	WHILEGT (predicate pair)	FEAT_SVE2p1
0	1	0	WHILELT (predicate pair)	FEAT_SVE2p1
0	1	1	WHILELE (predicate pair)	FEAT_SVE2p1
1	0	0	WHILEHS (predicate pair)	FEAT_SVE2p1
1	0	1	WHILEHI (predicate pair)	FEAT_SVE2p1
1	1	0	WHILELO (predicate pair)	FEAT_SVE2p1
1	1	1	WHILELS (predicate pair)	FEAT_SVE2p1

SVE integer compare scalar count and limit (predicate-as-counter)

These instructions are under [SVE Scalar Integer Compare - Predicate-as-counter](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	size	1	Rm	0	1	vl	0	U	lt	Rn	1	eq	PNd
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---	------	---	----	---	---	----	---	---	----	----	---	----	-----

Decode fields			Instruction Details	Feature
U	lt	eq		
0	0	0	WHILEGE (predicate as counter)	FEAT_SVE2p1
0	0	1	WHILEGT (predicate as counter)	FEAT_SVE2p1
0	1	0	WHILELT (predicate as counter)	FEAT_SVE2p1
0	1	1	WHILELE (predicate as counter)	FEAT_SVE2p1
1	0	0	WHILEHS (predicate as counter)	FEAT_SVE2p1
1	0	1	WHILEHI (predicate as counter)	FEAT_SVE2p1
1	1	0	WHILELO (predicate as counter)	FEAT_SVE2p1
1	1	1	WHILELS (predicate as counter)	FEAT_SVE2p1

SVE Integer Wide Immediate - Unpredicated

These instructions are under [SVE encodings](#).

Decode fields		Instruction details
op0	op1	
00		SVE integer add/subtract immediate (unpredicated)
01		SVE integer min/max immediate (unpredicated)
10		SVE integer multiply immediate (unpredicated)
11	0	SVE broadcast integer immediate (unpredicated)
11	1	SVE broadcast floating-point immediate (unpredicated)

SVE integer add/subtract immediate (unpredicated)

These instructions are under [SVE Integer Wide Immediate - Unpredicated](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	0	0	1	0	1	size	1	0	0	opc	1	1	sh		imm8		Zdn												

Decode fields		Instruction Details
opc		
000		ADD (immediate)
001		SUB (immediate)
010		UNALLOCATED
011		SUBR (immediate)
100		SQADD (immediate)
101		UQADD (immediate)
110		SQSUB (immediate)
111		UQSUB (immediate)

SVE integer min/max immediate (unpredicated)

These instructions are under [SVE Integer Wide Immediate - Unpredicated](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	0	0	1	0	1	size	1	0	1	opc	1	1	o2		imm8		Zdn												

Decode fields		Instruction Details
opc	o2	
0xx	1	UNALLOCATED
000	0	SMAX (immediate)

Decode fields		Instruction Details
opc	o2	
001	0	UMAX (immediate)
010	0	SMIN (immediate)
011	0	UMIN (immediate)
1xx		UNALLOCATED

SVE integer multiply immediate (unpredicated)

These instructions are under [SVE Integer Wide Immediate - Unpredicated](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0	0	1	0	0	1	0	1	size	1	1	0	opc	1	1	o2	imm8												Zdn				

Decode fields		Instruction Details
opc	o2	
000	0	MUL (immediate)
000	1	UNALLOCATED
001		UNALLOCATED
01x		UNALLOCATED
1xx		UNALLOCATED

SVE broadcast integer immediate (unpredicated)

These instructions are under [SVE Integer Wide Immediate - Unpredicated](#).

Decode fields	Instruction Details
opc	
00	DUP (immediate)
01	UNALLOCATED
1x	UNALLOCATED

SVE broadcast floating-point immediate (unpredicated)

These instructions are under [SVE Integer Wide Immediate - Unpredicated](#).

Decode fields		Instruction Details
opc	o2	
00	0	FDUP
00	1	UNALLOCATED
01		UNALLOCATED
1x		UNALLOCATED

SVE Predicate Count

These instructions are under [SVE encodings](#).

Decode fields		Instruction details
op0	op1	
000	1	SVE predicate count (predicate-as-counter)
!= 000	1	UNALLOCATED
	0	SVE predicate count

SVE predicate count (predicate-as-counter)

These instructions are under [SVE Predicate Count](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	0	0	1	0	1	size	1	0	0	opc	1	0	0	0	0	v1	1	8	PNn			Rd							

Decode fields	Instruction Details	Feature
opc		
000	CNTP (predicate as counter)	FEAT_SVE2p1
001	UNALLOCATED	-
01x	UNALLOCATED	-
1xx	UNALLOCATED	-

SVE predicate count

These instructions are under [SVE Predicate Count](#).

Decode fields	Instruction Details
opc	

000	CNTP (predicate)
001	UNALLOCATED
01x	UNALLOCATED
1xx	UNALLOCATED

SVE Inc/Dec by Predicate Count

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
00100101												101	op0		1000	op1															

Decode fields	Instruction details
op0	op1

Instruction details

0	0	SVE saturating inc/dec vector by predicate count
0	1	SVE saturating inc/dec register by predicate count
1	0	SVE inc/dec vector by predicate count
1	1	SVE inc/dec register by predicate count

SVE saturating inc/dec vector by predicate count

These instructions are under [SVE Inc/Dec by Predicate Count](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	0	0	1	0	1	size	1	0	1	0	D	U	1	0	0	0	0	opc	Pm		Zdn								

Decode fields	Instruction Details	
D	U	opc

Instruction Details

		01	UNALLOCATED
		1x	UNALLOCATED
0	0	00	SQINCP (vector)
0	1	00	UQINCP (vector)
1	0	00	SQDECP (vector)
1	1	00	UQDECP (vector)

SVE saturating inc/dec register by predicate count

These instructions are under [SVE Inc/Dec by Predicate Count](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	0	0	1	0	1	size	1	0	1	0	D	U	1	0	0	0	1	sf	op	Pm		Rdn							

Decode fields				Instruction Details														
D	U	sf	op															

			1	UNALLOCATED
0	0	0	0	SQINCP (scalar) â€“ 32-bit
0	0	1	0	SQINCP (scalar) â€“ 64-bit
0	1	0	0	UQINCP (scalar) â€“ 32-bit
0	1	1	0	UQINCP (scalar) â€“ 64-bit
1	0	0	0	SQDECP (scalar) â€“ 32-bit
1	0	1	0	SQDECP (scalar) â€“ 64-bit
1	1	0	0	UQDECP (scalar) â€“ 32-bit
1	1	1	0	UQDECP (scalar) â€“ 64-bit

SVE inc/dec vector by predicate count

These instructions are under [SVE Inc/Dec by Predicate Count](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	0	0	1	0	1	size	1	0	1	1	op	D	1	0	0	0	0	opc2	Pm		Zdn								

Decode fields			Instruction Details														
op	D	opc2															

0		01	UNALLOCATED
0		1x	UNALLOCATED
0	0	00	INCP (vector)
0	1	00	DECP (vector)
1			UNALLOCATED

SVE inc/dec register by predicate count

These instructions are under [SVE Inc/Dec by Predicate Count](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	0	0	1	0	1	size	1	0	1	1	op	D	1	0	0	0	1	opc2	Pm		Rdn								

Decode fields			Instruction Details														
op	D	opc2															

0		01	UNALLOCATED
0		1x	UNALLOCATED
0	0	00	INCP (scalar)

Decode fields			Instruction Details
op	D	opc2	
0	1	00	DECP (scalar)
1			UNALLOCATED

SVE Write FFR

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
00100101								101				op0				op1				1001				op2				op3				op4							

Decode fields					Instruction details
op0	op1	op2	op3	op4	
0	00	000		00000	SVE FFR write from predicate
1	00	000	0000	00000	SVE FFR initialise
1	00	000	1xxx	00000	UNALLOCATED
1	00	000	x1xx	00000	UNALLOCATED
1	00	000	xx1x	00000	UNALLOCATED
1	00	000	xxx1	00000	UNALLOCATED
	00	000		!= 00000	UNALLOCATED
	00	!= 000			UNALLOCATED
	!= 00				UNALLOCATED

SVE FFR write from predicate

These instructions are under [SVE Write FFR](#).

Decode fields	Instruction Details
opc	
00	WRFFR
01	UNALLOCATED
1x	UNALLOCATED

SVE FFR initialise

These instructions are under [SVE Write FFR](#).

Decode fields	Instruction Details
opc	
00	SETFFR
01	UNALLOCATED
1x	UNALLOCATED

SVE Integer Multiply-Add - Unpredicated

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
01000100								0				0		op0																		

Decode fields	Instruction details
op0	
0000x	SVE integer dot product (unpredicated)
0001x	SVE2 saturating multiply-add interleaved long
001xx	CDOT (vectors)
01xxx	SVE2 complex integer multiply-add
10xxx	SVE2 integer multiply-add long
110xx	SVE2 saturating multiply-add long
1110x	SVE2 saturating multiply-add high
11110	SVE mixed sign dot product
11111	UNALLOCATED

SVE integer dot product (unpredicated)

These instructions are under [SVE Integer Multiply-Add - Unpredicated](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	1	0	0	size	0	Zm	0	0	0	0	0	U	Zn		Zda												

Decode fields	Instruction Details
U	
0	SDOT (4-way, vectors)
1	UDOT (4-way, vectors)

SVE2 saturating multiply-add interleaved long

These instructions are under [SVE Integer Multiply-Add - Unpredicated](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	1	0	0	size	0	Zm	0	0	0	0	1	S	Zn	Zda													

Decode fields
S

Instruction Details

0	SQDMLALBT
1	SQDMLSLBT

SVE2 complex integer multiply-add

These instructions are under [SVE Integer Multiply-Add - Unpredicated](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	1	0	0	size	0	Zm	0	0	1	op	rot	Zn	Zda														

Decode fields
op

Instruction Details

0	CMLA (vectors)
1	QRDCMLAH (vectors)

SVE2 integer multiply-add long

These instructions are under [SVE Integer Multiply-Add - Unpredicated](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	1	0	0	size	0	Zm	0	1	0	S	U	T	Zn	Zda													

Decode fields
S U T

Instruction Details

0	0	0	SMLALB (vectors)
0	0	1	SMLALT (vectors)
0	1	0	UMLALB (vectors)
0	1	1	UMLALT (vectors)
1	0	0	SMLS LB (vectors)
1	0	1	SMLS LT (vectors)
1	1	0	UMLS LB (vectors)
1	1	1	UMLS LT (vectors)

SVE2 saturating multiply-add long

These instructions are under [SVE Integer Multiply-Add - Unpredicated](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	1	0	0	size	0	Zm	0	1	1	0	S	T	Zn	Zda													

Decode fields		Instruction Details
S	T	
0	0	SQDMLALB (vectors)
0	1	SQDMLALT (vectors)
1	0	SQDMLSLB (vectors)
1	1	SQDMLSLT (vectors)

SVE2 saturating multiply-add high

These instructions are under [SVE Integer Multiply-Add - Unpredicated](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	1	0	0	size	0	Zm	0	1	1	1	0	S	Zn	Zda													

Decode fields		Instruction Details
S		
0		QRDMLAH (vectors)
1		QRDMLSH (vectors)

SVE mixed sign dot product

These instructions are under [SVE Integer Multiply-Add - Unpredicated](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	1	0	0	size	0	Zm	0	1	1	1	1	0		Zn	Zda												

Decode fields	size	Instruction Details	Feature
0x		UNALLOCATED	-
10		USDOT (vectors)	FEAT_I8MM
11		UNALLOCATED	-

SVE2 Integer - Predicated

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
01000100		0	op0			10	op1																								

Decode fields	Instruction details

op0	op1	
0010	1	SVE2 integer pairwise add and accumulate long
0011	1	UNALLOCATED
011x	1	UNALLOCATED
0x0x	1	SVE2 integer unary operations (predicated)
0xxx	0	SVE2 saturating/rounding bitwise shift left (predicated)
10xx	0	SVE2 integer halving add/subtract (predicated)
10xx	1	SVE2 integer pairwise arithmetic
11xx	0	SVE2 saturating add/subtract
11xx	1	UNALLOCATED

SVE2 integer pairwise add and accumulate long

These instructions are under [SVE2 Integer - Predicated](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	1	0	0	size	0	0	0	1	0	U	1	0	1	Pg		Zn		Zda									

Decode fields	Instruction Details
U	
0	SADALP
1	UADALP

SVE2 integer unary operations (predicated)

These instructions are under [SVE2 Integer - Predicated](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	1	0	0	size	0	0	Q	0	opc	1	0	1	Pg		Zn		Zd										

Decode fields	Instruction Details	
Q		
0	1x	
0	00	URECPE
0	01	URSORTE
1	00	SQABS
1	01	SQNEG

SVE2 saturating/rounding bitwise shift left (predicated)

These instructions are under [SVE2 Integer - Predicated](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	1	0	0	size	0	0	Q	R	N	U	1	0	0	Pg		Zm		Zdn									

Decode fields				Instruction Details
Q	R	N	U	
0		0		UNALLOCATED
0	0	1	0	SRSHL
0	0	1	1	URSHL
0	1	1	0	SRSHLR
0	1	1	1	URSHLR
1	0	0	0	SQSHL (vectors)
1	0	0	1	UQSHL (vectors)
1	0	1	0	SQRSHL
1	0	1	1	UQRSHL
1	1	0	0	SQSHLR
1	1	0	1	UQSHLR
1	1	1	0	SQRSHLR
1	1	1	1	UQRSHLR

SVE2 integer halving add/subtract (predicated)

These instructions are under [SVE2 Integer - Predicated](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	1	0	0	size	0	1	0	R	S	U	1	0	0	Pg		Zm		Zdn									

Decode fields			Instruction Details
R	S	U	
0	0	0	SHADD
0	0	1	UHADD
0	1	0	SHSUB
0	1	1	UHSUB
1	0	0	SRHADD
1	0	1	URHADD
1	1	0	SHSUBR
1	1	1	UHSUBR

SVE2 integer pairwise arithmetic

These instructions are under [SVE2 Integer - Predicated](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	1	0	0	size	0	1	0	opc	U	1	0	1	Pg	Zm		Zdn											

Decode fields **Instruction Details**
opc **U**

00	0	UNALLOCATED
00	1	ADDP
01		UNALLOCATED
10	0	SMAXP
10	1	UMAXP
11	0	SMINP
11	1	UMINP

SVE2 saturating add/subtract

These instructions are under [SVE2 Integer - Predicated](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	1	0	0	size	0	1	1	op	S	U	1	0	0	Pg	Zm		Zdn										

Decode fields **Instruction Details**
op **S** **U**

0	0	0	SQADD (vectors, predicated)
0	0	1	UQADD (vectors, predicated)
0	1	0	SQSUB (vectors, predicated)
0	1	1	UQSUB (vectors, predicated)
1	0	0	SUQADD
1	0	1	USQADD
1	1	0	SQSUBR
1	1	1	UQSUBR

SVE integer clamp

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	1	0	0	size	0	Zm		1	1	0	0	0	U		Zn		Zd										

Decode fields **Instruction Details** **Feature**
U

0	SCLAMP	FEAT_SVE2p1
1	UCLAMP	FEAT_SVE2p1

SVE permute vector elements (quadwords)

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	1	0	0	size	0	Zm	1	1	1	opc	Zn		Zd														

Decode fields opc	Instruction Details	Feature
000	ZIPQ1	FEAT_SVE2p1
001	ZIPQ2	FEAT_SVE2p1
010	UZPQ1	FEAT_SVE2p1
011	UZPQ2	FEAT_SVE2p1
10x	UNALLOCATED	-
110	TBLQ	FEAT_SVE2p1
111	UNALLOCATED	-

SVE Multiply - Indexed

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
01000100		1												op0																	

Decode fields op0	Instruction details
00000x	SVE integer dot product (indexed)
00001x	SVE2 integer multiply-add (indexed)
00010x	SVE2 saturating multiply-add high (indexed)
00011x	SVE mixed sign dot product (indexed)
001xxx	SVE2 saturating multiply-add (indexed)
0100xx	SVE2 complex integer dot product (indexed)
0101xx	UNALLOCATED
0110xx	SVE2 complex integer multiply-add (indexed)
0111xx	SVE2 complex saturating multiply-add (indexed)
10xxxx	SVE2 integer multiply-add long (indexed)
110xxx	SVE2 integer multiply long (indexed)
1110xx	SVE2 saturating multiply (indexed)
11110x	SVE2 saturating multiply high (indexed)
111110	SVE2 integer multiply (indexed)
111111	UNALLOCATED

SVE integer dot product (indexed)

These instructions are under [SVE Multiply - Indexed](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	1	0	0	size	1		opc	0	0	0	0	0	U		Zn		Zda										

Decode fields

size U

Instruction Details

0x		UNALLOCATED
10	0	SDOT (4-way, indexed) â€“ 32-bit
10	1	UDOT (4-way, indexed) â€“ 32-bit
11	0	SDOT (4-way, indexed) â€“ 64-bit
11	1	UDOT (4-way, indexed) â€“ 64-bit

SVE2 integer multiply-add (indexed)

These instructions are under [SVE Multiply - Indexed](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	1	0	0	size	1		opc	0	0	0	0	1	S		Zn		Zda										

Decode fields

size S

Instruction Details

0x	0	MLA (indexed) â€“ 16-bit
0x	1	MLS (indexed) â€“ 16-bit
10	0	MLA (indexed) â€“ 32-bit
10	1	MLS (indexed) â€“ 32-bit
11	0	MLA (indexed) â€“ 64-bit
11	1	MLS (indexed) â€“ 64-bit

SVE2 saturating multiply-add high (indexed)

These instructions are under [SVE Multiply - Indexed](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	1	0	0	size	1		opc	0	0	0	1	0	S		Zn		Zda										

Decode fields

size S

Instruction Details

0x	0	SQRDMLAH (indexed) â€“ 16-bit
0x	1	SQRDMLSH (indexed) â€“ 16-bit
10	0	SQRDMLAH (indexed) â€“ 32-bit

Decode fields
size S

Instruction Details

10	1	SQRDMLSH (indexed) â€“ 32-bit
11	0	SQRDMLAH (indexed) â€“ 64-bit
11	1	SQRDMLSH (indexed) â€“ 64-bit

SVE mixed sign dot product (indexed)

These instructions are under [SVE Multiply - Indexed](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	1	0	0	size	1		opc		0	0	0	1	1	U		Zn		Zda									

Decode fields
size U

Instruction Details

Feature

0x		UNALLOCATED	-
10	0	USDOT (indexed)	FEAT_I8MM
10	1	SUDOT	FEAT_I8MM
11		UNALLOCATED	-

SVE2 saturating multiply-add (indexed)

These instructions are under [SVE Multiply - Indexed](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	1	0	0	size	1		opc		0	0	1	S	il	T		Zn		Zda									

Decode fields
size S T

Instruction Details

0x			UNALLOCATED
10	0	0	SQDMLALB (indexed) â€“ 32-bit
10	0	1	SQDMLALT (indexed) â€“ 32-bit
10	1	0	SQDMLSLB (indexed) â€“ 32-bit
10	1	1	SQDMLSLT (indexed) â€“ 32-bit
11	0	0	SQDMLALB (indexed) â€“ 64-bit
11	0	1	SQDMLALT (indexed) â€“ 64-bit
11	1	0	SQDMLSLB (indexed) â€“ 64-bit
11	1	1	SQDMLSLT (indexed) â€“ 64-bit

SVE2 complex integer dot product (indexed)

These instructions are under [SVE Multiply - Indexed](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	1	0	0	size	1		opc		0	1	0	0	rot		Zn		Zda										

Decode fields
size

Instruction Details

0x	UNALLOCATED
10	CDOT (indexed) â€“ 32-bit
11	CDOT (indexed) â€“ 64-bit

SVE2 complex integer multiply-add (indexed)

These instructions are under [SVE Multiply - Indexed](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	1	0	0	size	1		opc		0	1	1	0	rot		Zn		Zda										

Decode fields
size

Instruction Details

0x	UNALLOCATED
10	CMLA (indexed) â€“ 16-bit
11	CMLA (indexed) â€“ 32-bit

SVE2 complex saturating multiply-add (indexed)

These instructions are under [SVE Multiply - Indexed](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	1	0	0	size	1		opc		0	1	1	1	rot		Zn		Zda										

Decode fields
size

Instruction Details

0x	UNALLOCATED
10	SQRDCMLAH (indexed) â€“ 16-bit
11	SQRDCMLAH (indexed) â€“ 32-bit

SVE2 integer multiply-add long (indexed)

These instructions are under [SVE Multiply - Indexed](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	1	0	0	size	1		opc		1	0	S	U	il	T		Zn		Zda									

Decode fields				Instruction Details
size	S	U	T	
0x				UNALLOCATED
10	0	0	0	SMLALB (indexed) â€“ 32-bit
10	0	0	1	SMLALT (indexed) â€“ 32-bit
10	0	1	0	UMLALB (indexed) â€“ 32-bit
10	0	1	1	UMLALT (indexed) â€“ 32-bit
10	1	0	0	SMLS LB (indexed) â€“ 32-bit
10	1	0	1	SMLS LT (indexed) â€“ 32-bit
10	1	1	0	UMLS LB (indexed) â€“ 32-bit
10	1	1	1	UMLS LT (indexed) â€“ 32-bit
11	0	0	0	SMLALB (indexed) â€“ 64-bit
11	0	0	1	SMLALT (indexed) â€“ 64-bit
11	0	1	0	UMLALB (indexed) â€“ 64-bit
11	0	1	1	UMLALT (indexed) â€“ 64-bit
11	1	0	0	SMLS LB (indexed) â€“ 64-bit
11	1	0	1	SMLS LT (indexed) â€“ 64-bit
11	1	1	0	UMLS LB (indexed) â€“ 64-bit
11	1	1	1	UMLS LT (indexed) â€“ 64-bit

SVE2 integer multiply long (indexed)

These instructions are under [SVE Multiply - Indexed](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	1	0	0	size	1		opc		1	1	0	U	il	T		Zn		Zd									

Decode fields				Instruction Details
size	U	T		
0x				UNALLOCATED
10	0	0		SMULLB (indexed) â€“ 32-bit
10	0	1		SMULLT (indexed) â€“ 32-bit
10	1	0		UMULLB (indexed) â€“ 32-bit
10	1	1		UMULLT (indexed) â€“ 32-bit
11	0	0		SMULLB (indexed) â€“ 64-bit
11	0	1		SMULLT (indexed) â€“ 64-bit
11	1	0		UMULLB (indexed) â€“ 64-bit
11	1	1		UMULLT (indexed) â€“ 64-bit

SVE2 saturating multiply (indexed)

These instructions are under [SVE Multiply - Indexed](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	1	0	0	size	1		opc		1	1	1	0	il	T		Zn		Zd									

Decode fields		Instruction Details
size	T	
0x		UNALLOCATED
10	0	SQDMULLB (indexed) â€“ 32-bit
10	1	SQDMULLT (indexed) â€“ 32-bit
11	0	SQDMULLB (indexed) â€“ 64-bit
11	1	SQDMULLT (indexed) â€“ 64-bit

SVE2 saturating multiply high (indexed)

These instructions are under [SVE Multiply - Indexed](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	1	0	0	size	1		opc		1	1	1	0	R		Zn		Zd										

Decode fields		Instruction Details
size	R	
0x	0	SQDMULH (indexed) â€“ 16-bit
0x	1	SORDMULH (indexed) â€“ 16-bit
10	0	SQDMULH (indexed) â€“ 32-bit
10	1	SORDMULH (indexed) â€“ 32-bit
11	0	SQDMULH (indexed) â€“ 64-bit
11	1	SORDMULH (indexed) â€“ 64-bit

SVE2 integer multiply (indexed)

These instructions are under [SVE Multiply - Indexed](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	1	0	0	size	1		opc		1	1	1	1	0		Zn		Zd										

Decode fields		Instruction Details
size		
0x		MUL (indexed) â€“ 16-bit
10		MUL (indexed) â€“ 32-bit
11		MUL (indexed) â€“ 64-bit

SVE two-way dot product

These instructions are under [SVE encodings](#).

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
0 1 0 0 0 1 0 0 0 0 0 Zm 1 1 0 0 1 U Zn Zda

Decode fields U	Instruction Details	Feature
0	SDOT (2-way, vectors)	FEAT_SVE2p1
1	UDOT (2-way, vectors)	FEAT_SVE2p1

SVE two-way dot product (indexed)

These instructions are under [SVE encodings](#).

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
0 1 0 0 0 1 0 0 1 0 0 opc 1 1 0 0 1 U Zn Zda

Decode fields U	Instruction Details	Feature
0	SDOT (2-way, indexed)	FEAT_SVE2p1
1	UDOT (2-way, indexed)	FEAT_SVE2p1

SVE2 Widening Integer Arithmetic

These instructions are under [SVE encodings](#).

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
01000101 0 0 op0

Decode fields op0	Instruction details
0x	SVE2 integer add/subtract long
10	SVE2 integer add/subtract wide
11	SVE2 integer multiply long

SVE2 integer add/subtract long

These instructions are under [SVE2 Widening Integer Arithmetic](#).

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
0 1 0 0 0 1 0 1 size 0 Zm 0 0 op S U T Zn Zd

Decode fields				Instruction Details
op	S	U	T	
0	0	0	0	SADDLB
0	0	0	1	SADDLT
0	0	1	0	UADDLB
0	0	1	1	UADDLT
0	1	0	0	SSUBLB
0	1	0	1	SSUBLT
0	1	1	0	USUBLB
0	1	1	1	USUBLT
1	0			UNALLOCATED
1	1	0	0	SABDLB
1	1	0	1	SABDLT
1	1	1	0	UABDLB
1	1	1	1	UABDLT

SVE2 integer add/subtract wide

These instructions are under [SVE2 Widening Integer Arithmetic](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	size	0	Zm	0	1	0	S	U	T	Zn	Zd
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---	------	---	----	---	---	---	---	---	---	----	----

Decode fields			Instruction Details
S	U	T	
0	0	0	<u>SADDWB</u>
0	0	1	<u>SADDWT</u>
0	1	0	<u>UADDWB</u>
0	1	1	<u>UADDWT</u>
1	0	0	<u>SSUBWB</u>
1	0	1	<u>SSUBWT</u>
1	1	0	<u>USUBWB</u>
1	1	1	<u>USUBWT</u>

SVE2 integer multiply long

These instructions are under [SVE2 Widening Integer Arithmetic](#).

Decode fields				Instruction Details	Feature
size	op	U	T		
	0	0	0	SQDMULLB (vectors)	-
	0	0	1	SQDMULLT (vectors)	-
	1	0	0	SMULLB (vectors)	-
	1	0	1	SMULLT (vectors)	-
	1	1	0	UMULLB (vectors)	-
	1	1	1	UMULLT (vectors)	-
!= 00	0	1	0	PMULLB â€“ 16-bit or 64-bit elements	-
!= 00	0	1	1	PMULLT â€“ 16-bit or 64-bit elements	-
00	0	1	0	PMULLB â€“ 128-bit element	FEAT_SVE_PMULL128
00	0	1	1	PMULLT â€“ 128-bit element	FEAT_SVE_PMULL128

SVE Misc

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
01000101				opo				0								10				op1																			

Decode fields		Instruction details
op0	op1	
0	10xx	SVE2 bitwise shift left long
1	10xx	UNALLOCATED
	00xx	SVE2 integer add/subtract interleaved long
	010x	SVE2 bitwise exclusive-or interleaved
	0110	SVE integer matrix multiply accumulate
	0111	UNALLOCATED
	11xx	SVE2 bitwise permute

SVE2 bitwise shift left long

These instructions are under [SVE Misc.](#)

Decode fields		Instruction Details
U	T	
0	0	<u>SSHLLB</u>
0	1	<u>SSHLLT</u>
1	0	<u>USHLLB</u>
1	1	<u>USHLLT</u>

SVE2 integer add/subtract interleaved long

These instructions are under [SVE Misc.](#)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	size	0	Zm	1	0	0	0	S	tb	Zn	Zd
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---	------	---	----	---	---	---	---	---	----	----	----

Decode fields		Instruction Details
S	tb	
0	0	SADDLBT
0	1	UNALLOCATED
1	0	SSUBLBT
1	1	SSUBLTB

SVE2 bitwise exclusive-or interleaved

These instructions are under [SVE Misc.](#)

Decode fields	tb	Instruction Details
	0	EORBT
	1	EORTB

SVE integer matrix multiply accumulate

These instructions are under [SVE Misc.](#)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	0	1	0	0	0	1	0	1	uns	0	Zm	1	0	0	1	1	0	Zn	Zd
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	-----	---	----	---	---	---	---	---	---	----	----

Decode fields uns	Instruction Details	Feature
00	SMMLA	FEAT_I8MM
01	UNALLOCATED	-
10	USMMLA	FEAT_I8MM
11	UMMLA	FEAT_I8MM

SVE2 bitwise permute

These instructions are under [SVE Misc.](#)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	1	0	1	size	0	Zm	1	0	1	1	opc	Zn		Zd													

Decode fields opc	Instruction Details	Feature
00	BEXT	FEAT_SVE_BitPerm
01	BDEP	FEAT_SVE_BitPerm
10	BGRP	FEAT_SVE_BitPerm
11	UNALLOCATED	-

SVE2 Accumulate

These instructions are under [SVE encodings.](#)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
01000101		0	op0		11	op1																									

Decode fields		Instruction details
op0	op1	
0000	011	SVE2 complex integer add
!= 0000	011	UNALLOCATED
	00x	SVE2 integer absolute difference and accumulate long
	010	SVE2 integer add/subtract long with carry
	10x	SVE2 bitwise shift right and accumulate
	110	SVE2 bitwise shift and insert
	111	SVE2 integer absolute difference and accumulate

SVE2 complex integer add

These instructions are under [SVE2 Accumulate](#).

Decode fields	Instruction Details
op	
0	<u>CADD</u>
1	<u>SQCADD</u>

SVE2 integer absolute difference and accumulate long

These instructions are under [SVE2 Accumulate](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	size	0	Zm	1	1	0	0	U	T	Zn	Zda
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---	------	---	----	---	---	---	---	---	---	----	-----

Decode fields		Instruction Details
U	T	
0	0	<u>SABALB</u>
0	1	<u>SABALT</u>
1	0	<u>UABALB</u>
1	1	<u>UABALT</u>

SVE2 integer add/subtract long with carry

These instructions are under [SVE2 Accumulate](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	size	0	Zm	1	1	0	1	0	T	Zn	Zda
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---	------	---	----	---	---	---	---	---	---	----	-----

Decode fields		Instruction Details
size	T	
0x	0	<u>ADCLB</u>
0x	1	<u>ADCLT</u>
1x	0	<u>SBCLB</u>
1x	1	<u>SBCLT</u>

SVE2 bitwise shift right and accumulate

These instructions are under SVE2 Accumulate.

Decode fields Instruction Details

R	U	
0	0	SSRA
0	1	USRA
1	0	SRSRA
1	1	URSRA

SVE2 bitwise shift and insert

These instructions are under [SVE2 Accumulate](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	1	0	1	tszh	0	tszl	imm3	1	1	1	1	0	op		Zn		Zd										

Decode fields Instruction Details

op	
0	SRI
1	SLI

SVE2 integer absolute difference and accumulate

These instructions are under [SVE2 Accumulate](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	1	0	1	size	0		Zm	1	1	1	1	1	U		Zn		Zda										

Decode fields Instruction Details

U	
0	SABA
1	UABA

SVE2 Narrowing

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	1	0	1	op0	1		op1	op2	0	op3		op4		op5													

Decode fields

op0	op1	op2	op3	op4	op5	Instruction details
0	00	0	10			SVE2 saturating extract narrow

0	00	1	10	0	0	SME2 multi-vec extract narrow
0	00	1	10	0	1	UNALLOCATED
0	00	1	10	1		UNALLOCATED
0	!= 00		10			UNALLOCATED
0			0x			SVE2 bitwise shift right narrow
1			0x	0	0	SME2 multi-vec shift narrow
1			0x	0	1	UNALLOCATED
1			0x	1		UNALLOCATED
1			10			UNALLOCATED
			11			SVE2 integer add/ subtract narrow high part

SVE2 saturating extract narrow

These instructions are under [SVE2 Narrowing](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	1	0	1	0	tszh	1	tszl	0	0	0	0	1	0	opc	T		Zn		Zd								

Decode fields		Instruction Details	
opc	T		
00	0	SQXTNB	
00	1	SQXTNT	
01	0	UQXTNB	
01	1	UQXTNT	
10	0	SQXTUNB	
10	1	SQXTUNT	
11		UNALLOCATED	

SME2 multi-vec extract narrow

These instructions are under [SVE2 Narrowing](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	1	0	1	0	tszh	1	tszl	0	0	1	0	1	0	opc	0		Zn	0	Zd								

Decode fields			Instruction Details		Feature
tszh	tszl	opc			
0	0x		UNALLOCATED		-
0	10	00	SQCVTN		FEAT_SVE2p1

Decode fields			Instruction Details	Feature
tszh	tszl	opc		
0	10	01	UQCVTN	FEAT_SVE2p1
0	10	10	SQCVTUN	FEAT_SVE2p1
0	10	11	UNALLOCATED	-
0	11		UNALLOCATED	-
1			UNALLOCATED	-

SVE2 bitwise shift right narrow

These instructions are under [SVE2 Narrowing](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	0	1	0	0	0	1	0	1	0	tszh	1	tszl	imm3	0	0	op	U	R	T		Zn		Zd
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	------	---	------	------	---	---	----	---	---	---	--	----	--	----

Decode fields				Instruction Details
op	U	R	T	
0	0	0	0	SQSHRUNB
0	0	0	1	SQSHRUNT
0	0	1	0	SQRSHRUNB
0	0	1	1	SQRSHRUNT
0	1	0	0	SHRNB
0	1	0	1	SHRNT
0	1	1	0	RSHRNB
0	1	1	1	RSHRNT
1	0	0	0	SQSHRNB
1	0	0	1	SQSHRNT
1	0	1	0	SQRSHRNB
1	0	1	1	SQRSHRNT
1	1	0	0	UQSHRNB
1	1	0	1	UQSHRNT
1	1	1	0	UQRSHRNB
1	1	1	1	UQRSHRNT

SME2 multi-vec shift narrow

These instructions are under [SVE2 Narrowing](#).

Decode fields					Instruction Details	Feature
tszh	tszl	op	U	R		
0	0				UNALLOCATED	-
0	1	0	0	0	UNALLOCATED	-
0	1	0	0	1	SQRSHRUN	FEAT_SVE2p1
0	1	0	1		UNALLOCATED	-
0	1	1		0	UNALLOCATED	-
0	1	1	0	1	SQRSHRN	FEAT_SVE2p1
0	1	1	1	1	UQRSHRN	FEAT_SVE2p1
1					UNALLOCATED	-

SVE2 integer add/subtract narrow high part

These instructions are under [SVE2 Narrowing](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	1	0	1	size	1	Zm	0	1	1	S	R	T	Zn		Zd												

Decode fields			Instruction Details
S	R	T	
0	0	0	ADDHNB
0	0	1	ADDHNT
0	1	0	RADDHNB
0	1	1	RADDHNT
1	0	0	SUBHNB
1	0	1	SUBHNT
1	1	0	RSUBHNB
1	1	1	RSUBHNT

SVE2 character match

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	1	0	1	size	1	Zm	1	0	0	Pg		Zn	op	Pd													

Decode fields		Instruction Details
op		
0		MATCH
1		NMATCH

SVE2 Histogram Computation - Segment

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
01000101							1					101		op0																	

Decode fields op0	Instruction details
000	HISTSEG
!= 000	UNALLOCATED

SVE2 Crypto Extensions

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
01000101							1	op0	op1	111	op2			op3																	

op0	Decode fields			Instruction details									
	op1	op2	op3										
000	00	00	00000	SVE2 crypto unary operations									
000	00	00	!= 00000	UNALLOCATED									
000	00	x1		UNALLOCATED									
000	01	0x		UNALLOCATED									
000	01	11		UNALLOCATED									
000	1x	00		SVE2 crypto destructive binary operations									
000	1x	x1		UNALLOCATED									
!= 000		0x		UNALLOCATED									
!= 000		11		UNALLOCATED									
		10		SVE2 crypto constructive binary operations									

SVE2 crypto unary operations

These instructions are under [SVE2 Crypto Extensions](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	1	0	1	size	1	0	0	0	0	1	1	1	0	0	op	0	0	0	0	Zdn							

Decode fields		Instruction Details		Feature
size	op			
00	0	AESMC		FEAT_SVE_AES
00	1	AESIMC		FEAT_SVE_AES
01		UNALLOCATED		-
1x		UNALLOCATED		-

SVE2 crypto destructive binary operations

These instructions are under [SVE2 Crypto Extensions](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	1	0	1	size	1	0	0	0	1	op	1	1	1	0	0	o2	Zm		Zdn								

Decode fields			Instruction Details		Feature
size	op	o2			
00	0	0	AESE		FEAT_SVE_AES
00	0	1	AESD		FEAT_SVE_AES
00	1	0	SM4E		FEAT_SVE_SM4
00	1	1	UNALLOCATED		-
01			UNALLOCATED		-
1x			UNALLOCATED		-

SVE2 crypto constructive binary operations

These instructions are under [SVE2 Crypto Extensions](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	1	0	1	size	1		Zm		1	1	1	1	0	op		Zn		Zd									

Decode fields		Instruction Details		Feature
size	op			
00	0	SM4EKEY		FEAT_SVE_SM4
00	1	RAX1		FEAT_SVE_SHA3
01		UNALLOCATED		-
1x		UNALLOCATED		-

SVE floating-point convert precision odd elements

These instructions are under [SVE encodings](#).

Decode fields		Instruction Details	Feature
opc	opc2		
x0	11	UNALLOCATED	-
00	0x	UNALLOCATED	-
00	10	FCVTXNT	-
01		UNALLOCATED	-
10	00	FCVTNT " single-precision to half-precision	-
10	01	FCVTLT " half-precision to single-precision	-
10	10	BFCVTNT	FEAT_BF16
11	0x	UNALLOCATED	-
11	10	FCVTNT " double-precision to single-precision	-
11	11	FCVTLT " single-precision to double-precision	-

SVE2 floating-point pairwise operations

These instructions are under [SVE encodings](#).

Decode fields	Instruction Details
opc	
000	<u>FADDP</u>
001	UNALLOCATED
01x	UNALLOCATED
100	<u>FMAXNMP</u>
101	<u>FMINNMP</u>
110	<u>FMAXP</u>
111	<u>FMINP</u>

SVE floating-point recursive reduction (quadwords)

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	size	0	1	0	opc	1	0	1	Pg		Zn		Vd
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---	------	---	---	---	-----	---	---	---	----	--	----	--	----

Decode fields opc	Instruction Details	Feature
000	<u>FADDQV</u>	FEAT_SVE2p1
001	UNALLOCATED	-
01x	UNALLOCATED	-
100	<u>FMAXNMQV</u>	FEAT_SVE2p1
101	<u>FMINNMQV</u>	FEAT_SVE2p1
110	<u>FMAXQV</u>	FEAT_SVE2p1
111	<u>FMINQV</u>	FEAT_SVE2p1

SVE floating-point multiply-add (indexed)

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	size	1	opc	0	0	0	0	o2op	Zn	Zda
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---	------	---	-----	---	---	---	---	------	----	-----

Decode fields			Instruction Details	Feature
size	o2	op		
0x	0	0	FMLA (indexed) â€“ half-precision	-
0x	0	1	FMLS (indexed) â€“ half-precision	-
0x	1	0	BFMLA (indexed)	FEAT_SVE_B16B16
0x	1	1	BFMLS (indexed)	FEAT_SVE_B16B16
1x	1		UNALLOCATED	-
10	0	0	FMLA (indexed) â€“ single-precision	-
10	0	1	FMLS (indexed) â€“ single-precision	-
11	0	0	FMLA (indexed) â€“ double-precision	-
11	0	1	FMLS (indexed) â€“ double-precision	-

SVE floating-point complex multiply-add (indexed)

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	size	1	opc	0	0	0	1	rot	Zn		Zda
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---	------	---	-----	---	---	---	---	-----	----	--	-----

Decode fields	Instruction Details
size	
0x	UNALLOCATED
10	FCMLA (indexed) â€“ half-precision
11	FCMLA (indexed) â€“ single-precision

SVE floating-point multiply (indexed)

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	size	1	opc	0	0	1	0	02	0	0	Zn	Zd
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---	------	---	-----	---	---	---	---	----	---	---	----	----

Decode fields		Instruction Details	Feature
size	o2		
0x	0	FMUL (indexed) â€“ half-precision	-
0x	1	BFMUL (indexed)	FEAT_SVE_B16B16
1x	1	UNALLOCATED	-
10	0	FMUL (indexed) â€“ single-precision	-
11	0	FMUL (indexed) â€“ double-precision	-

SVE FP clamp

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	size	1	Zm	0	0	1	0	0	1	0	0	1	Zn	Zd
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---	------	---	----	---	---	---	---	---	---	---	---	---	----	----

Decode fields size	Instruction Details	Feature
!= 00	FCLAMP	FEAT_SVE2p1
00	BFCLAMP	FEAT_SVE_B16B16

SVE Floating Point Widening Multiply-Add - Indexed

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
01100100				opb	1					01				op	10																

Decode fields		Instruction details
op0	op1	
0	0	SVE BFLOAT16 floating-point dot product (indexed)
0	1	UNALLOCATED
1		SVE floating-point multiply-add long (indexed)

SVE BFLOAT16 floating-point dot product (indexed)

These instructions are under [SVE Floating Point Widening Multiply-Add - Indexed](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	0	0	1	0	0	0	op	1	i2	Zm	0	1	0	0	opc2	Zn		Zda											

Decode fields		Instruction Details	Feature
op	opc2		
	01	UNALLOCATED	-
	1x	UNALLOCATED	-
0	00	FDOT (indexed)	FEAT_SVE2p1
1	00	BFDOT (indexed)	FEAT_BF16

SVE floating-point multiply-add long (indexed)

These instructions are under [SVE Floating Point Widening Multiply-Add - Indexed](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	0	0	1	0	0	1	o2	1	i3h	Zm	0	1	op	0	i3l	T	Zn		Zda										

Decode fields			Instruction Details	Feature
o2	op	T		
0	0	0	FMLALB (indexed)	-
0	0	1	FMLALT (indexed)	-
0	1	0	FMLSLB (indexed)	-
0	1	1	FMLSLT (indexed)	-
1	0	0	BFMLALB (indexed)	FEAT_BF16
1	0	1	BFMLALT (indexed)	FEAT_BF16
1	1	0	BFMLSLB (indexed)	FEAT_SVE2p1
1	1	1	BFMLSLT (indexed)	FEAT_SVE2p1

SVE Floating Point Widening Multiply-Add

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
01100100		op0	1													10	op100															

Decode fields		Instruction details		
op0	op1	SVE BF16 floating-point dot product		
0	0	UNALLOCATED		
1		SVE floating-point multiply-add long		

SVE BF16 floating-point dot product

These instructions are under [SVE Floating Point Widening Multiply-Add](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	0	0	1	0	0	0	op1	Zm	1	0	0	0	0	o2	Zn	Zda													

Decode fields		Instruction Details		Feature
op	o2			
	1	UNALLOCATED		-
0	0	FDOT (vectors)		FEAT_SVE2p1
1	0	BFDOT (vectors)		FEAT_BF16

SVE floating-point multiply-add long

These instructions are under [SVE Floating Point Widening Multiply-Add](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	0	0	1	0	0	1	o21	Zm	1	0	op0	0	0	T	Zn	Zda													

Decode fields			Instruction Details		Feature
o2	op	T			
0	0	0	FMLALB (vectors)		-
0	0	1	FMLALT (vectors)		-
0	1	0	FMLSLB (vectors)		-
0	1	1	FMLSLT (vectors)		-
1	0	0	BFMLALB (vectors)		FEAT_BF16
1	0	1	BFMLALT (vectors)		FEAT_BF16
1	1	0	BFMLSLB (vectors)		FEAT_SVE2p1
1	1	1	BFMLSLT (vectors)		FEAT_SVE2p1

SVE floating point matrix multiply accumulate

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	0	0	1	0	0	opc	1	Zm		1	1	1	0	0	1	Zn		Zda											

Decode fields opc	Instruction Details	Feature
00	UNALLOCATED	-
01	BFMMLA	FEAT_BF16
10	FMMLA â€“ 32-bit element	FEAT_F32MM
11	FMMLA â€“ 64-bit element	FEAT_F64MM

SVE floating-point compare vectors

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	0	0	1	0	1	size	0	Zm	op	1	0	2	Pg		Zn	o3	Pd												

Decode fields op	o2	o3	Instruction Details
0	0	0	FCM<cc> (vectors) â€“ FCMGE
0	0	1	FCM<cc> (vectors) â€“ FCMGT
0	1	0	FCM<cc> (vectors) â€“ FCMEO
0	1	1	FCM<cc> (vectors) â€“ FCMNE
1	0	0	FCM<cc> (vectors) â€“ FCMUO
1	0	1	FAC<cc> â€“ FACGE
1	1	0	UNALLOCATED
1	1	1	FAC<cc> â€“ FACGT

SVE floating-point arithmetic (unpredicated)

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	0	0	1	0	1	size	0	Zm	0	0	0	opc		Zn		Zd													

Decode fields size	opc	Instruction Details	Feature
	011	FTSMUL	-

Decode fields		Instruction Details		Feature
size	opc			
	10x	UNALLOCATED		-
	110	FRECPS		-
	111	FRSQRTS		-
!= 00	000	FADD (vectors, unpredicated)		-
!= 00	001	FSUB (vectors, unpredicated)		-
!= 00	010	FMUL (vectors, unpredicated)		-
00	000	BFADD (unpredicated)		FEAT_SVE_B16B16
00	001	BFSUB (unpredicated)		FEAT_SVE_B16B16
00	010	BFMUL (vectors, unpredicated)		FEAT_SVE_B16B16

SVE Floating Point Arithmetic - Predicated

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
01100101				0		op0	100		op1		op2																				

Decode fields			Instruction details	
op0	op1	op2		
0x			SVE floating-point arithmetic (predicated)	
10	000		FTMAD	
10	!= 000		UNALLOCATED	
11		0000	SVE floating-point arithmetic with immediate (predicated)	
11		!= 0000	UNALLOCATED	

SVE floating-point arithmetic (predicated)

These instructions are under [SVE Floating Point Arithmetic - Predicated](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
0	1	1	0	0	1	0	1	size	0	0	opc	1	0	0	Pg	Zm	Zdn																	

Decode fields		Instruction Details		Feature
size	opc			
	0011	FSUBR (vectors)		-
	1000	FABD		-
	1001	FSCALE		-

Decode fields		Instruction Details	Feature
size	opc		
	1010	FMULX	-
	1011	UNALLOCATED	-
	1100	FDIVR	-
	1101	FDIV	-
	111x	UNALLOCATED	-
!= 00	0000	FADD (vectors, predicated)	-
!= 00	0001	FSUB (vectors, predicated)	-
!= 00	0010	FMUL (vectors, predicated)	-
!= 00	0100	FMAXNM (vectors)	-
!= 00	0101	FMINNM (vectors)	-
!= 00	0110	FMAX (vectors)	-
!= 00	0111	FMIN (vectors)	-
00	0000	BFADD (predicated)	FEAT_SVE_B16B16
00	0001	BFSUB (predicated)	FEAT_SVE_B16B16
00	0010	BFMUL (vectors, predicated)	FEAT_SVE_B16B16
00	0100	BFMAXNM	FEAT_SVE_B16B16
00	0101	BFMINNM	FEAT_SVE_B16B16
00	0110	BFMAX	FEAT_SVE_B16B16
00	0111	BFMIN	FEAT_SVE_B16B16

SVE floating-point arithmetic with immediate (predicated)

These instructions are under [SVE Floating Point Arithmetic - Predicated](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	0	0	1	0	1	size	0	1	1	opc	1	0	0	Pg	0	0	0	0	i1	Zdn									

Decode fields		Instruction Details
opc		
000		FADD (immediate)
001		FSUB (immediate)
010		FMUL (immediate)
011		FSUBR (immediate)
100		FMAXNM (immediate)
101		FMINNM (immediate)
110		FMAX (immediate)
111		FMIN (immediate)

SVE Floating Point Unary Operations - Predicated

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
01100101				0 op0				101																							

Decode fields	Instruction details
op0	
00x	SVE floating-point round to integral value
010	SVE floating-point convert precision
011	SVE floating-point unary operations
10x	SVE integer convert to floating-point
11x	SVE floating-point convert to integer

SVE floating-point round to integral value

These instructions are under [SVE Floating Point Unary Operations - Predicated](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	size	0	0	0	opc	1	0	1	Pq		Zn		Zd
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---	------	---	---	---	-----	---	---	---	----	--	----	--	----

Decode fields		Instruction Details
opc		
000	FRINT<r>	"nearest with ties to even"
001	FRINT<r>	"toward plus infinity"
010	FRINT<r>	"toward minus infinity"
011	FRINT<r>	"toward zero"
100	FRINT<r>	"nearest with ties to away"
101	UNALLOCATED	
110	FRINT<r>	"current mode signalling inexact"
111	FRINT<r>	"current mode"

SVE floating-point convert precision

These instructions are under [SVE Floating Point Unary Operations - Predicated](#).

Decode fields		Instruction Details	Feature
opc	opc2		
x0	11	UNALLOCATED	-
00	0x	UNALLOCATED	-
00	10	FCVTX	-
01		UNALLOCATED	-
10	00	FCVT " single-precision to half-precision	-
10	01	FCVT " half-precision to single-precision	-
10	10	BFCVT	FEAT_BF16
11	00	FCVT " double-precision to half-precision	-
11	01	FCVT " half-precision to double-precision	-
11	10	FCVT " double-precision to single-precision	-
11	11	FCVT " single-precision to double-precision	-

SVE floating-point unary operations

These instructions are under [SVE Floating Point Unary Operations - Predicated](#).

Decode fields	Instruction Details
opc	
00	FRECPX
01	FSQRT
1x	UNALLOCATED

SVE integer convert to floating-point

These instructions are under [SVE Floating Point Unary Operations - Predicated](#).

Decode fields			Instruction Details
opc	opc2	U	
00			UNALLOCATED
01	00		UNALLOCATED
01	01	0	SCVTF â€“ 16-bit to half-precision
01	01	1	UCVTF â€“ 16-bit to half-precision
01	10	0	SCVTF â€“ 32-bit to half-precision
01	10	1	UCVTF â€“ 32-bit to half-precision
01	11	0	SCVTF â€“ 64-bit to half-precision
01	11	1	UCVTF â€“ 64-bit to half-precision
10	0x		UNALLOCATED
10	10	0	SCVTF â€“ 32-bit to single-precision
10	10	1	UCVTF â€“ 32-bit to single-precision
10	11		UNALLOCATED
11	00	0	SCVTF â€“ 32-bit to double-precision
11	00	1	UCVTF â€“ 32-bit to double-precision
11	01		UNALLOCATED
11	10	0	SCVTF â€“ 64-bit to single-precision
11	10	1	UCVTF â€“ 64-bit to single-precision
11	11	0	SCVTF â€“ 64-bit to double-precision
11	11	1	UCVTF â€“ 64-bit to double-precision

SVE floating-point convert to integer

These instructions are under [SVE Floating Point Unary Operations - Predicated](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	0	0	1	0	1	opc	0	1	1	opc2	U	1	0	1	Pg		Zn		Zd										

Decode fields			Instruction Details
opc	opc2	U	
00		0	FLOGB
00		1	UNALLOCATED
01	00		UNALLOCATED
01	01	0	FCVTZS â€“ half-precision to 16-bit
01	01	1	FCVTZU â€“ half-precision to 16-bit
01	10	0	FCVTZS â€“ half-precision to 32-bit
01	10	1	FCVTZU â€“ half-precision to 32-bit
01	11	0	FCVTZS â€“ half-precision to 64-bit

Decode fields			Instruction Details
opc	opc2	U	
01	11	1	FCVTZU â€“ half-precision to 64-bit
10	0x		UNALLOCATED
10	10	0	FCVTZS â€“ single-precision to 32-bit
10	10	1	FCVTZU â€“ single-precision to 32-bit
10	11		UNALLOCATED
11	00	0	FCVTZS â€“ double-precision to 32-bit
11	00	1	FCVTZU â€“ double-precision to 32-bit
11	01		UNALLOCATED
11	10	0	FCVTZS â€“ single-precision to 64-bit
11	10	1	FCVTZU â€“ single-precision to 64-bit
11	11	0	FCVTZS â€“ double-precision to 64-bit
11	11	1	FCVTZU â€“ double-precision to 64-bit

SVE floating-point recursive reduction

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	0	0	1	0	1	size	0	0	0	opc	0	0	1	Pg		Zn		Vd											

Decode fields		Instruction Details
opc		
000		FADDV
001		UNALLOCATED
01x		UNALLOCATED
100		FMAXNMV
101		FMINNMV
110		FMAXV
111		FMINV

SVE Floating Point Unary Operations - Unpredicated

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0	1	1	0	0	1	0	1	001		op0		0011		op1																		

Decode fields		Instruction details
op0	op1	
0x		UNALLOCATED

10		UNALLOCATED
11	00	SVE floating-point reciprocal estimate (unpredicated)
11	!= 00	UNALLOCATED

SVE floating-point reciprocal estimate (unpredicated)

These instructions are under [SVE Floating Point Unary Operations - Unpredicated](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	0	0	1	0	1	size	0	0	1	1	1	op	0	0	1	1	0	0	Zn		Zd								

Decode fields op	Instruction Details
0	FRECPE
1	FRSQRTE

SVE Floating Point Compare - with Zero

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0	1	1	0	0	1	0	1	01100101		010	op0		001																			

Decode fields op0	Instruction details
0	SVE floating-point compare with zero
1	UNALLOCATED

SVE floating-point compare with zero

These instructions are under [SVE Floating Point Compare - with Zero](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	0	0	1	0	1	size	0	1	0	0	eq	lt	0	0	1	Pg		Zn	ne	Pd									

Decode fields eq	lt	ne	Instruction Details
0	0	0	FCM<cc> (zero) â€” FCMGE
0	0	1	FCM<cc> (zero) â€” FCMGT
0	1	0	FCM<cc> (zero) â€” FCMLT
0	1	1	FCM<cc> (zero) â€” FCMLE
1		1	UNALLOCATED

Decode fields			Instruction Details	
eq	lt	ne		
1	0	0	FCM<cc> (zero) → FCMEQ	
1	1	0	FCM<cc> (zero) → FCMNE	

SVE Floating Point Accumulating Reduction

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
01100101								011	op0			001																			

Decode fields		Instruction details	
op0			
0		SVE floating-point serial reduction (predicated)	
1		UNALLOCATED	

SVE floating-point serial reduction (predicated)

These instructions are under [SVE Floating Point Accumulating Reduction](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	0	0	1	0	1	size	0	1	1	0	opc	0	0	1	Pg		Zm		Vdn										

Decode fields		Instruction Details	
opc			
00		FADDA	
01		UNALLOCATED	
1x		UNALLOCATED	

SVE Floating Point Multiply-Add

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
01100101								1				op0																			

Decode fields		Instruction details	
op0			
0		SVE floating-point multiply-accumulate writing addend	
1		SVE floating-point multiply-accumulate writing multiplicand	

SVE floating-point multiply-accumulate writing addend

These instructions are under [SVE Floating Point Multiply-Add](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	0	0	1	0	1	size	1	Zm	0	opc	Pg		Zn		Zda														

Decode fields		Instruction Details		Feature
size	opc			
	10	FNMLA		-
	11	FNMLS		-
!= 00	00	FMLA (vectors)		-
!= 00	01	FMLS (vectors)		-
00	00	BFMLA (vectors)		FEAT_SVE_B16B16
00	01	BFMLS (vectors)		FEAT_SVE_B16B16

SVE floating-point multiply-accumulate writing multiplicand

These instructions are under [SVE Floating Point Multiply-Add](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	0	0	1	0	1	size	1	Za	1	opc	Pg		Zm		Zdn														

Decode fields		Instruction Details	
opc			
00		FMAD	
01		FMSB	
10		FNMAD	
11		FNMSB	

SVE Memory - 32-bit Gather and Unsized Contiguous

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1000010	op0	op1																								op2			op3		

Decode fields				Instruction details
op0	op1	op2	op3	
00	x1	0xx	0	SVE 32-bit gather prefetch (scalar plus 32-bit scaled offsets)
00	x1	0xx	1	UNALLOCATED
01	x1	0xx		SVE 32-bit gather load halfwords (scalar plus 32-bit scaled offsets)

10	x1	0xx		SVE 32-bit gather load words (scalar plus 32-bit scaled offsets)
11	0x	000	0	LDR (predicate)
11	0x	000	1	UNALLOCATED
11	0x	010		LDR (vector)
11	0x	0x1		UNALLOCATED
11	1x	0xx	0	SVE contiguous prefetch (scalar plus immediate)
11	1x	0xx	1	UNALLOCATED
!= 11	x0	0xx		SVE 32-bit gather load (scalar plus 32-bit unscaled offsets)
	00	10x		SVE2 32-bit gather non-temporal load (vector plus scalar)
	00	110	0	SVE contiguous prefetch (scalar plus scalar)
	00	111	0	SVE 32-bit gather prefetch (vector plus immediate)
	00	11x	1	UNALLOCATED
	01	1xx		SVE 32-bit gather load (vector plus immediate)
	1x	1xx		SVE load and broadcast element

SVE 32-bit gather prefetch (scalar plus 32-bit scaled offsets)

These instructions are under [SVE Memory - 32-bit Gather and Unsized Contiguous](#).

Decode fields	Instruction Details
msz	
00	PRFB (scalar plus vector)
01	PRFH (scalar plus vector)
10	PRFW (scalar plus vector)
11	PRFD (scalar plus vector)

SVE 32-bit gather load halfwords (scalar plus 32-bit scaled offsets)

These instructions are under [SVE Memory - 32-bit Gather and Unsized Contiguous](#).

Decode fields		Instruction Details
U	ff	
0	0	LD1SH (scalar plus vector)
0	1	LDFF1SH (scalar plus vector)
1	0	LD1H (scalar plus vector)
1	1	LDFF1H (scalar plus vector)

SVE 32-bit gather load words (scalar plus 32-bit scaled offsets)

These instructions are under [SVE Memory - 32-bit Gather and Unsized Contiguous](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	0	0	0	1	0	1	0	xs	1	Zm	0	U	ff	Pg		Rn		Zt												

Decode fields		Instruction Details
U	ff	
0		UNALLOCATED
1	0	LD1W (scalar plus vector)
1	1	LDFF1W (scalar plus vector)

SVE contiguous prefetch (scalar plus immediate)

These instructions are under [SVE Memory - 32-bit Gather and Unsized Contiguous](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	0	0	0	1	0	1	1	1	imm6	0	msz	Pg		Rn	0	prfop														

Decode fields		Instruction Details
msz		
00		PRFB (scalar plus immediate)
01		PRFH (scalar plus immediate)
10		PRFW (scalar plus immediate)
11		PRFD (scalar plus immediate)

SVE 32-bit gather load (scalar plus 32-bit unscaled offsets)

These instructions are under [SVE Memory - 32-bit Gather and Unsized Contiguous](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	0	0	0	1	0	!= 11	xs	0		Zm	0	U	ff	Pg		Rn		Zt										opc		

The following constraints also apply to this encoding: opc != 11 && opc != 11

Decode fields			Instruction Details														
opc	U	ff															
00	0	0	LD1SB (scalar plus vector)														
00	0	1	LDFF1SB (scalar plus vector)														
00	1	0	LD1B (scalar plus vector)														
00	1	1	LDFF1B (scalar plus vector)														
01	0	0	LD1SH (scalar plus vector)														
01	0	1	LDFF1SH (scalar plus vector)														
01	1	0	LD1H (scalar plus vector)														
01	1	1	LDFF1H (scalar plus vector)														
10	0		UNALLOCATED														
10	1	0	LD1W (scalar plus vector)														
10	1	1	LDFF1W (scalar plus vector)														

SVE2 32-bit gather non-temporal load (vector plus scalar)

These instructions are under [SVE Memory - 32-bit Gather and Unsized Contiguous](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	0	0	0	1	0	msz	0	0		Rm	1	0	U	Pg		Rn		Zn		Zt										

Decode fields			Instruction Details														
msz	U																
00	0		LDNT1SB														
00	1		LDNT1B (vector plus scalar)														
01	0		LDNT1SH														
01	1		LDNT1H (vector plus scalar)														
10	0		UNALLOCATED														
10	1		LDNT1W (vector plus scalar)														
11			UNALLOCATED														

SVE contiguous prefetch (scalar plus scalar)

These instructions are under [SVE Memory - 32-bit Gather and Unsized Contiguous](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	0	0	0	1	0	msz	0	0	Rm	1	1	0	Pg		Rn	0	prfop													

Decode fields	Instruction Details
msz	
00	PRFB (scalar plus scalar)
01	PRFH (scalar plus scalar)
10	PRFW (scalar plus scalar)
11	PRFD (scalar plus scalar)

SVE 32-bit gather prefetch (vector plus immediate)

These instructions are under [SVE Memory - 32-bit Gather and Unsized Contiguous](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	0	0	0	1	0	msz	0	0	imm5	1	1	1	Pg		Zn	0	prfop													

Decode fields	Instruction Details
msz	
00	PRFB (vector plus immediate)
01	PRFH (vector plus immediate)
10	PRFW (vector plus immediate)
11	PRFD (vector plus immediate)

SVE 32-bit gather load (vector plus immediate)

These instructions are under [SVE Memory - 32-bit Gather and Unsized Contiguous](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	0	0	0	1	0	msz	0	1	imm5	1	U	ff	Pg		Zn		Zt													

Decode fields	Instruction Details
msz	
U	
ff	
00	LD1SB (vector plus immediate)
00	LDFF1SB (vector plus immediate)
00	LD1B (vector plus immediate)
00	LDFF1B (vector plus immediate)
01	LD1SH (vector plus immediate)
01	LDFF1SH (vector plus immediate)

Decode fields			Instruction Details
msz	U	ff	
01	1	0	LD1H (vector plus immediate)
01	1	1	LDFF1H (vector plus immediate)
10	0		UNALLOCATED
10	1	0	LD1W (vector plus immediate)
10	1	1	LDFF1W (vector plus immediate)
11			UNALLOCATED

SVE load and broadcast element

These instructions are under [SVE Memory - 32-bit Gather and Unsized Contiguous](#).

```
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
| 1 0 0 0 0 1 0 | dtype: h   | imm6 | 1 | dtype: l | Pg |     | Rn |     | Zt |
```

Decode fields		Instruction Details
dtypeh	dtypel	
00	00	LD1RB â€“ 8-bit element
00	01	LD1RB â€“ 16-bit element
00	10	LD1RB â€“ 32-bit element
00	11	LD1RB â€“ 64-bit element
01	00	LD1RSW
01	01	LD1RH â€“ 16-bit element
01	10	LD1RH â€“ 32-bit element
01	11	LD1RH â€“ 64-bit element
10	00	LD1RSW â€“ 64-bit element
10	01	LD1RSW â€“ 32-bit element
10	10	LD1RW â€“ 32-bit element
10	11	LD1RW â€“ 64-bit element
11	00	LD1RSB â€“ 64-bit element
11	01	LD1RSB â€“ 32-bit element
11	10	LD1RSB â€“ 16-bit element
11	11	LD1RD

SVE Memory - Contiguous Load

These instructions are under [SVE encodings](#).

Decode fields			Instruction details
op0	op1	op2	
00	0	111	SVE contiguous non-temporal load (scalar plus immediate)
00	1	001	SVE contiguous load (quadwords, scalar plus immediate)
00	1	111	SVE load multiple structures (quadwords, scalar plus immediate)
00		100	SVE contiguous load (quadwords, scalar plus scalar)
00		110	SVE contiguous non-temporal load (scalar plus scalar)
01		100	SVE load multiple structures (quadwords, scalar plus scalar)
1x		100	UNALLOCATED
!= 00	0	111	SVE load multiple structures (scalar plus immediate)
!= 00	1	001	UNALLOCATED
!= 00	1	111	UNALLOCATED
!= 00		110	SVE load multiple structures (scalar plus scalar)
	0	001	SVE load and broadcast quadword (scalar plus immediate)
	0	101	SVE contiguous load (scalar plus immediate)
	1	101	SVE contiguous non-fault load (scalar plus immediate)
		000	SVE load and broadcast quadword (scalar plus scalar)
		010	SVE contiguous load (scalar plus scalar)
		011	SVE contiguous first-fault load (scalar plus scalar)

SVE contiguous non-temporal load (scalar plus immediate)

These instructions are under [SVE Memory - Contiguous Load](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	1	0	0	1	0	msz	0	0	0	imm4	1	1	1	Pg		Rn		Zt												

Decode fields		Instruction Details
msz		
00		LDNT1B (scalar plus immediate, single register)
01		LDNT1H (scalar plus immediate, single register)
10		LDNT1W (scalar plus immediate, single register)

Decode fields msz	Instruction Details
11	LDNT1D (scalar plus immediate, single register)

SVE contiguous load (quadwords, scalar plus immediate)

These instructions are under [SVE Memory - Contiguous Load](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	1	0	0	1	0	dtype=		0	0	1	imm4		0	0	1	Pg		Rn		Zt										

Decode fields	Instruction Details	Feature
dtype		
0x	UNALLOCATED	-
10	LD1W (scalar plus immediate, single register)	FEAT_SVE2p1
11	LD1D (scalar plus immediate, single register)	FEAT_SVE2p1

SVE load multiple structures (quadwords, scalar plus immediate)

These instructions are under [SVE Memory - Contiguous Load](#).

Decode fields	Instruction Details	Feature
num		
00	UNALLOCATED	-
01	LD2Q (scalar plus immediate)	FEAT_SVE2p1
10	LD3Q (scalar plus immediate)	FEAT_SVE2p1
11	LD4Q (scalar plus immediate)	FEAT_SVE2p1

SVE contiguous load (quadwords, scalar plus scalar)

These instructions are under [SVE Memory - Contiguous Load](#).

```
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
| 1 0 1 0 0 1 0 | dtype<=0 0 | Rm | 1 0 0 | Pg | 9 | Rn | Zt
```

Decode fields dtype	Instruction Details	Feature
0x	UNALLOCATED	-
10	LD1W (scalar plus scalar, single register)	FEAT_SVE2p1
11	LD1D (scalar plus scalar, single register)	FEAT_SVE2p1

SVE contiguous non-temporal load (scalar plus scalar)

These instructions are under [SVE Memory - Contiguous Load](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	1	0	0	1	0	msz	0	0	Rm	1	1	0	Pg		Rn		Zt													

Decode fields msz	Instruction Details
00	LDNT1B (scalar plus scalar, single register)
01	LDNT1H (scalar plus scalar, single register)
10	LDNT1W (scalar plus scalar, single register)
11	LDNT1D (scalar plus scalar, single register)

SVE load multiple structures (quadwords, scalar plus scalar)

These instructions are under [SVE Memory - Contiguous Load](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	1	0	0	1	0	num	0	1	Rm	1	0	0	Pg		Rn		Zt													

Decode fields num	Instruction Details	Feature
00	UNALLOCATED	-
01	LD2Q (scalar plus scalar)	FEAT_SVE2p1
10	LD3Q (scalar plus scalar)	FEAT_SVE2p1
11	LD4Q (scalar plus scalar)	FEAT_SVE2p1

SVE load multiple structures (scalar plus immediate)

These instructions are under [SVE Memory - Contiguous Load](#).

The following constraints also apply to this encoding: $\text{opc} \neq 00 \text{ \&& } \text{opc} \neq 00$

Decode fields		Instruction Details
msz	opc	
00	01	LD2B (scalar plus immediate)
00	10	LD3B (scalar plus immediate)
00	11	LD4B (scalar plus immediate)
01	01	LD2H (scalar plus immediate)
01	10	LD3H (scalar plus immediate)
01	11	LD4H (scalar plus immediate)
10	01	LD2W (scalar plus immediate)
10	10	LD3W (scalar plus immediate)
10	11	LD4W (scalar plus immediate)
11	01	LD2D (scalar plus immediate)
11	10	LD3D (scalar plus immediate)
11	11	LD4D (scalar plus immediate)

SVE load multiple structures (scalar plus scalar)

These instructions are under [SVE Memory - Contiguous Load](#).

The following constraints also apply to this encoding: $\text{opc} \neq 00 \text{ \&& } \text{opc} \neq 00$

Decode fields		Instruction Details
msz	opc	
00	01	LD2B (scalar plus scalar)
00	10	LD3B (scalar plus scalar)
00	11	LD4B (scalar plus scalar)
01	01	LD2H (scalar plus scalar)
01	10	LD3H (scalar plus scalar)
01	11	LD4H (scalar plus scalar)

Decode fields		Instruction Details	
msz	opc		
10	01	LD2W (scalar plus scalar)	
10	10	LD3W (scalar plus scalar)	
10	11	LD4W (scalar plus scalar)	
11	01	LD2D (scalar plus scalar)	
11	10	LD3D (scalar plus scalar)	
11	11	LD4D (scalar plus scalar)	

SVE load and broadcast quadword (scalar plus immediate)

These instructions are under [SVE Memory - Contiguous Load](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	1	0	0	1	0	msz	ssz	0	imm4	0	0	1	Pg		Rn		Zt													

Decode fields		Instruction Details		Feature
msz	ssz			
	1x	UNALLOCATED		-
00	00	LD1RQB (scalar plus immediate)		-
00	01	LD1ROB (scalar plus immediate)		FEAT_F64MM
01	00	LD1RQH (scalar plus immediate)		-
01	01	LD1ROH (scalar plus immediate)		FEAT_F64MM
10	00	LD1RQW (scalar plus immediate)		-
10	01	LD1ROW (scalar plus immediate)		FEAT_F64MM
11	00	LD1RQD (scalar plus immediate)		-
11	01	LD1ROD (scalar plus immediate)		FEAT_F64MM

SVE contiguous load (scalar plus immediate)

These instructions are under [SVE Memory - Contiguous Load](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	1	0	0	1	0	dtype	0	imm4	1	0	1	Pg		Rn		Zt														

Decode fields		Instruction Details
dtype		
0000		LD1B (scalar plus immediate, single register) â€” 8-bit element
0001		LD1B (scalar plus immediate, single register) â€” 16-bit element

Decode fields	Instruction Details
dtype	
0010	LD1B (scalar plus immediate, single register) â€“ 32-bit element
0011	LD1B (scalar plus immediate, single register) â€“ 64-bit element
0100	LD1SW (scalar plus immediate)
0101	LD1H (scalar plus immediate, single register) â€“ 16-bit element
0110	LD1H (scalar plus immediate, single register) â€“ 32-bit element
0111	LD1H (scalar plus immediate, single register) â€“ 64-bit element
1000	LD1SH (scalar plus immediate) â€“ 64-bit element
1001	LD1SH (scalar plus immediate) â€“ 32-bit element
1010	LD1W (scalar plus immediate, single register) â€“ 32-bit element
1011	LD1W (scalar plus immediate, single register) â€“ 64-bit element
1100	LD1SB (scalar plus immediate) â€“ 64-bit element
1101	LD1SB (scalar plus immediate) â€“ 32-bit element
1110	LD1SB (scalar plus immediate) â€“ 16-bit element
1111	LD1D (scalar plus immediate, single register)

SVE contiguous non-fault load (scalar plus immediate)

These instructions are under [SVE Memory - Contiguous Load](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	1	0	0	1	0																								Zt	

Decode fields	Instruction Details
dtype	
0000	LDNF1B â€“ 8-bit element
0001	LDNF1B â€“ 16-bit element
0010	LDNF1B â€“ 32-bit element
0011	LDNF1B â€“ 64-bit element
0100	LDNF1SW
0101	LDNF1H â€“ 16-bit element
0110	LDNF1H â€“ 32-bit element
0111	LDNF1H â€“ 64-bit element
1000	LDNF1SH â€“ 64-bit element
1001	LDNF1SH â€“ 32-bit element

Decode fields
dtype

Instruction Details

1010	LDNF1W â€“ 32-bit element
1011	LDNF1W â€“ 64-bit element
1100	LDNF1SB â€“ 64-bit element
1101	LDNF1SB â€“ 32-bit element
1110	LDNF1SB â€“ 16-bit element
1111	LDNF1D

SVE load and broadcast quadword (scalar plus scalar)

These instructions are under [SVE Memory - Contiguous Load](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	1	0	0	1	0	msz	ssz		Rm	0	0	0	Pg		Rn		Zt													

Decode fields
msz **ssz**

Instruction Details

Feature

	1x	UNALLOCATED	-
00	00	LD1RQB (scalar plus scalar)	-
00	01	LD1ROB (scalar plus scalar)	FEAT_F64MM
01	00	LD1RQH (scalar plus scalar)	-
01	01	LD1ROH (scalar plus scalar)	FEAT_F64MM
10	00	LD1RQW (scalar plus scalar)	-
10	01	LD1ROW (scalar plus scalar)	FEAT_F64MM
11	00	LD1RQD (scalar plus scalar)	-
11	01	LD1ROD (scalar plus scalar)	FEAT_F64MM

SVE contiguous load (scalar plus scalar)

These instructions are under [SVE Memory - Contiguous Load](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	1	0	0	1	0	dtype		Rm	0	1	0	Pg		Rn		Zt														

**Decode
fields**
dtype

Instruction Details

0000	LD1B (scalar plus scalar, single register) â€“ 8-bit element
0001	LD1B (scalar plus scalar, single register) â€“ 16-bit element

Decode fields		Instruction Details
dtype		
0010		LD1B (scalar plus scalar, single register) â€“ 32-bit element
0011		LD1B (scalar plus scalar, single register) â€“ 64-bit element
0100		LD1SW (scalar plus scalar)
0101		LD1H (scalar plus scalar, single register) â€“ 16-bit element
0110		LD1H (scalar plus scalar, single register) â€“ 32-bit element
0111		LD1H (scalar plus scalar, single register) â€“ 64-bit element
1000		LD1SH (scalar plus scalar) â€“ 64-bit element
1001		LD1SH (scalar plus scalar) â€“ 32-bit element
1010		LD1W (scalar plus scalar, single register) â€“ 32-bit element
1011		LD1W (scalar plus scalar, single register) â€“ 64-bit element
1100		LD1SB (scalar plus scalar) â€“ 64-bit element
1101		LD1SB (scalar plus scalar) â€“ 32-bit element
1110		LD1SB (scalar plus scalar) â€“ 16-bit element
1111		LD1D (scalar plus scalar, single register)

SVE contiguous first-fault load (scalar plus scalar)

These instructions are under [SVE Memory - Contiguous Load](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	1	0	0	1	0																								Zt	

Decode fields		Instruction Details
dtype		
0000		LDFF1B (scalar plus scalar) â€“ 8-bit element
0001		LDFF1B (scalar plus scalar) â€“ 16-bit element
0010		LDFF1B (scalar plus scalar) â€“ 32-bit element
0011		LDFF1B (scalar plus scalar) â€“ 64-bit element
0100		LDFF1SW (scalar plus scalar)
0101		LDFF1H (scalar plus scalar) â€“ 16-bit element
0110		LDFF1H (scalar plus scalar) â€“ 32-bit element
0111		LDFF1H (scalar plus scalar) â€“ 64-bit element
1000		LDFF1SH (scalar plus scalar) â€“ 64-bit element
1001		LDFF1SH (scalar plus scalar) â€“ 32-bit element

Decode fields		Instruction Details
dtype		
1010		LDFF1W (scalar plus scalar) â€“ 32-bit element
1011		LDFF1W (scalar plus scalar) â€“ 64-bit element
1100		LDFF1SB (scalar plus scalar) â€“ 64-bit element
1101		LDFF1SB (scalar plus scalar) â€“ 32-bit element
1110		LDFF1SB (scalar plus scalar) â€“ 16-bit element
1111		LDFF1D (scalar plus scalar)

SVE Memory - 64-bit Gather

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1100010		op0	op1																							op2				op3	

Decode fields				Instruction details	Feature
op0	op1	op2	op3		
00	00	101		LD1Q	FEAT_SVE2p1
00	01	0xx	1	UNALLOCATED	-
00	11	1xx	0	SVE 64-bit gather prefetch (scalar plus 64-bit scaled offsets)	-
00	11		1	UNALLOCATED	-
00	x1	0xx	0	SVE 64-bit gather prefetch (scalar plus unpacked 32-bit scaled offsets)	-
!= 00	00	101		UNALLOCATED	-
!= 00	11	1xx		SVE 64-bit gather load (scalar plus 64-bit scaled offsets)	-
!= 00	x1	0xx		SVE 64-bit gather load (scalar plus 32-bit unpacked scaled offsets)	-
	00	111	0	SVE 64-bit gather prefetch (vector plus immediate)	-
	00	111	1	UNALLOCATED	-
	00	1x0		SVE2 64-bit gather non-temporal load (vector plus scalar)	-
	01	1xx		SVE 64-bit gather load (vector plus immediate)	-

	10	1xx		SVE 64-bit gather load (scalar plus 64-bit unscaled offsets)	-
	x0	0xx		SVE 64-bit gather load (scalar plus unpacked 32- bit unscaled offsets)	-

SVE 64-bit gather prefetch (scalar plus 64-bit scaled offsets)

These instructions are under [SVE Memory - 64-bit Gather](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	1	0	0	0	1	1	Zm	1	msz	Pg		Rn	0	prfop													

Decode fields	Instruction Details
msz	
00	PRFB (scalar plus vector)
01	PRFH (scalar plus vector)
10	PRFW (scalar plus vector)
11	PRFD (scalar plus vector)

SVE 64-bit gather prefetch (scalar plus unpacked 32-bit scaled offsets)

These instructions are under [SVE Memory - 64-bit Gather](#).

Decode fields	Instruction Details
msz	
00	PRFB (scalar plus vector)
01	PRFH (scalar plus vector)
10	PRFW (scalar plus vector)
11	PRFD (scalar plus vector)

SVE 64-bit gather load (scalar plus 64-bit scaled offsets)

These instructions are under [SVE Memory - 64-bit Gather](#).

The following constraints also apply to this encoding: opc != 00 && opc != 00

Decode fields			Instruction Details
opc	U	ff	
01	0	0	LD1SH (scalar plus vector)
01	0	1	LDFF1SH (scalar plus vector)
01	1	0	LD1H (scalar plus vector)
01	1	1	LDFF1H (scalar plus vector)
10	0	0	LD1SW (scalar plus vector)
10	0	1	LDFF1SW (scalar plus vector)
10	1	0	LD1W (scalar plus vector)
10	1	1	LDFF1W (scalar plus vector)
11	0		UNALLOCATED
11	1	0	LD1D (scalar plus vector)
11	1	1	LDFF1D (scalar plus vector)

SVE 64-bit gather load (scalar plus 32-bit unpacked scaled offsets)

These instructions are under [SVE Memory - 64-bit Gather](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	1	0	!= 00	xs	1	Zm	0	U	ff	Pg		Rn		Zt													

The following constraints also apply to this encoding: opc != 00 && opc != 00

Decode fields			Instruction Details
opc	U	ff	
01	0	0	LD1SH (scalar plus vector)
01	0	1	LDFF1SH (scalar plus vector)
01	1	0	LD1H (scalar plus vector)
01	1	1	LDFF1H (scalar plus vector)
10	0	0	LD1SW (scalar plus vector)
10	0	1	LDFF1SW (scalar plus vector)
10	1	0	LD1W (scalar plus vector)
10	1	1	LDFF1W (scalar plus vector)
11	0		UNALLOCATED
11	1	0	LD1D (scalar plus vector)

Decode fields			Instruction Details
opc	U	ff	
11	1	1	LDFF1D (scalar plus vector)

SVE 64-bit gather prefetch (vector plus immediate)

These instructions are under [SVE Memory - 64-bit Gather](#).

Decode fields	Instruction Details
msz	
00	PRFB (vector plus immediate)
01	PRFH (vector plus immediate)
10	PRFW (vector plus immediate)
11	PRFD (vector plus immediate)

SVE2 64-bit gather non-temporal load (vector plus scalar)

These instructions are under [SVE Memory - 64-bit Gather](#).

Decode fields		Instruction Details
msz	U	
00	0	LDNT1SB
00	1	LDNT1B (vector plus scalar)
01	0	LDNT1SH
01	1	LDNT1H (vector plus scalar)
10	0	LDNT1SW
10	1	LDNT1W (vector plus scalar)
11	0	UNALLOCATED
11	1	LDNT1D (vector plus scalar)

SVE 64-bit gather load (vector plus immediate)

These instructions are under [SVE Memory - 64-bit Gather](#).

Decode fields			Instruction Details
msz	U	ff	
00	0	0	LD1SB (vector plus immediate)
00	0	1	LDFF1SB (vector plus immediate)
00	1	0	LD1B (vector plus immediate)
00	1	1	LDFF1B (vector plus immediate)
01	0	0	LD1SH (vector plus immediate)
01	0	1	LDFF1SH (vector plus immediate)
01	1	0	LD1H (vector plus immediate)
01	1	1	LDFF1H (vector plus immediate)
10	0	0	LD1SW (vector plus immediate)
10	0	1	LDFF1SW (vector plus immediate)
10	1	0	LD1W (vector plus immediate)
10	1	1	LDFF1W (vector plus immediate)
11	0		UNALLOCATED
11	1	0	LD1D (vector plus immediate)
11	1	1	LDFF1D (vector plus immediate)

SVE 64-bit gather load (scalar plus 64-bit unscaled offsets)

These instructions are under [SVE Memory - 64-bit Gather](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	1	0	msz	1	0	Zm	1	U	ff	Pg		Rn		Zt													

Decode fields			Instruction Details
msz	U	ff	
00	0	0	LD1SB (scalar plus vector)
00	0	1	LDFF1SB (scalar plus vector)
00	1	0	LD1B (scalar plus vector)
00	1	1	LDFF1B (scalar plus vector)
01	0	0	LD1SH (scalar plus vector)
01	0	1	LDFF1SH (scalar plus vector)
01	1	0	LD1H (scalar plus vector)
01	1	1	LDFF1H (scalar plus vector)
10	0	0	LD1SW (scalar plus vector)
10	0	1	LDFF1SW (scalar plus vector)
10	1	0	LD1W (scalar plus vector)
10	1	1	LDFF1W (scalar plus vector)
11	0		UNALLOCATED

Decode fields			Instruction Details
msz	U	ff	
11	1	0	LD1D (scalar plus vector)
11	1	1	LDFF1D (scalar plus vector)

SVE 64-bit gather load (scalar plus unpacked 32-bit unscaled offsets)

These instructions are under [SVE Memory - 64-bit Gather](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	1	0	msz	xs	0	Zm	0	U	ff	Pg		Rn		Zt													

Decode fields			Instruction Details
msz	U	ff	
00	0	0	LD1SB (scalar plus vector)
00	0	1	LDFF1SB (scalar plus vector)
00	1	0	LD1B (scalar plus vector)
00	1	1	LDFF1B (scalar plus vector)
01	0	0	LD1SH (scalar plus vector)
01	0	1	LDFF1SH (scalar plus vector)
01	1	0	LD1H (scalar plus vector)
01	1	1	LDFF1H (scalar plus vector)
10	0	0	LD1SW (scalar plus vector)
10	0	1	LDFF1SW (scalar plus vector)
10	1	0	LD1W (scalar plus vector)
10	1	1	LDFF1W (scalar plus vector)
11	0		UNALLOCATED
11	1	0	LD1D (scalar plus vector)
11	1	1	LDFF1D (scalar plus vector)

SVE Memory - Contiguous Store and Unsized Contiguous

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1110010		op0	op1			0	op2	0																			op3				

Decode fields				Instruction details
op0	op1	op2	op3	
0xx	00	0		SVE store multiple structures (quadwords, scalar plus immediate)
0xx	01	0		UNALLOCATED

0xx	1x	0		SVE store multiple structures (quadwords, scalar plus scalar)
10x		0		UNALLOCATED
110		0	0	STR (predicate)
110		0	1	UNALLOCATED
110		1		STR (vector)
111		0		UNALLOCATED
!= 110		1		SVE contiguous store (scalar plus scalar)

SVE store multiple structures (quadwords, scalar plus immediate)

These instructions are under [SVE Memory - Contiguous Store and Unsized Contiguous](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	0	0	1	0	0	num	0	0	imm4	0	0	0	Pg		Rn		Zt												

Decode fields num	Instruction Details	Feature
00	UNALLOCATED	-
01	ST2Q (scalar plus immediate)	FEAT_SVE2p1
10	ST3Q (scalar plus immediate)	FEAT_SVE2p1
11	ST4Q (scalar plus immediate)	FEAT_SVE2p1

SVE store multiple structures (quadwords, scalar plus scalar)

These instructions are under [SVE Memory - Contiguous Store and Unsized Contiguous](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	0	0	1	0	0	num	1		Rm	0	0	0	Pg		Rn		Zt												

Decode fields num	Instruction Details	Feature
00	UNALLOCATED	-
01	ST2Q (scalar plus scalar)	FEAT_SVE2p1
10	ST3Q (scalar plus scalar)	FEAT_SVE2p1
11	ST4Q (scalar plus scalar)	FEAT_SVE2p1

SVE contiguous store (scalar plus scalar)

These instructions are under [SVE Memory - Contiguous Store and Unsized Contiguous](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	0	0	1	0	!= 110	o2	Rm	0	1	0	Pg		Rn		Zt														

opc

The following constraints also apply to this encoding: opc != 110 && o2 != 110

Decode fields		Instruction Details		Feature
opc	o2			
00x		ST1B (scalar plus scalar, single register)		-
01x		ST1H (scalar plus scalar, single register)		-
100	0	ST1W (scalar plus scalar, single register) â€“ SVE2		FEAT_SVE2p1
101		ST1W (scalar plus scalar, single register) â€“ SVE		-
111	0	ST1D (scalar plus scalar, single register) â€“ SVE2		FEAT_SVE2p1
111	1	ST1D (scalar plus scalar, single register) â€“ SVE		-

SVE Memory - Non-temporal and Quadword Scatter Store

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1110010		op0	op1													001															

Decode fields		Instruction details		Feature
op0	op1			
000	1	ST1Q		FEAT_SVE2p1
!= 000	1	UNALLOCATED		-
xx0	0	SVE2 64-bit scatter non-temporal store (vector plus scalar)		-
xx1	0	SVE2 32-bit scatter non-temporal store (vector plus scalar)		-

SVE2 64-bit scatter non-temporal store (vector plus scalar)

These instructions are under [SVE Memory - Non-temporal and Quadword Scatter Store](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	0	0	1	0	msz	0	0	Rm	0	0	1	Pg	Zn															Zt	

Decode fields

msz

Instruction Details

00	STNT1B (vector plus scalar)
01	STNT1H (vector plus scalar)
10	STNT1W (vector plus scalar)
11	STNT1D (vector plus scalar)

SVE2 32-bit scatter non-temporal store (vector plus scalar)

These instructions are under [SVE Memory - Non-temporal and Quadword Scatter Store](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	0	0	1	0	msz	1	0	Rm	0	0	1	Pg	Zn															Zt	

Decode fields

msz

Instruction Details

00	STNT1B (vector plus scalar)
01	STNT1H (vector plus scalar)
10	STNT1W (vector plus scalar)
11	UNALLOCATED

SVE Memory - Non-temporal and Multi-register Contiguous Store

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1110010		op0													011																

Decode

fields

op0

Instruction details

00	SVE contiguous non-temporal store (scalar plus scalar)
!= 00	SVE store multiple structures (scalar plus scalar)

SVE contiguous non-temporal store (scalar plus scalar)

These instructions are under [SVE Memory - Non-temporal and Multi-register Contiguous Store](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	0	0	1	0	msz	0	0	Rm	0	1	1	Pg		Rn		Zt													

Decode fields	Instruction Details
msz	
00	STNT1B (scalar plus scalar, single register)
01	STNT1H (scalar plus scalar, single register)
10	STNT1W (scalar plus scalar, single register)
11	STNT1D (scalar plus scalar, single register)

SVE store multiple structures (scalar plus scalar)

These instructions are under [SVE Memory - Non-temporal and Multi-register Contiguous Store](#).

The following constraints also apply to this encoding: `opc != 00 && opc != 00`

Decode fields		Instruction Details
msz	opc	
00	01	ST2B (scalar plus scalar)
00	10	ST3B (scalar plus scalar)
00	11	ST4B (scalar plus scalar)
01	01	ST2H (scalar plus scalar)
01	10	ST3H (scalar plus scalar)
01	11	ST4H (scalar plus scalar)
10	01	ST2W (scalar plus scalar)
10	10	ST3W (scalar plus scalar)
10	11	ST4W (scalar plus scalar)
11	01	ST2D (scalar plus scalar)
11	10	ST3D (scalar plus scalar)
11	11	ST4D (scalar plus scalar)

SVE Memory - Scatter with Optional Sign Extend

These instructions are under [SVE encodings](#).

Decode fields
op0

Instruction details

00	SVE 64-bit scatter store (scalar plus unpacked 32-bit unscaled offsets)
01	SVE 64-bit scatter store (scalar plus unpacked 32-bit scaled offsets)
10	SVE 32-bit scatter store (scalar plus 32-bit unscaled offsets)
11	SVE 32-bit scatter store (scalar plus 32-bit scaled offsets)

SVE 64-bit scatter store (scalar plus unpacked 32-bit unscaled offsets)

These instructions are under [SVE Memory - Scatter with Optional Sign Extend](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
1	1	1	0	0	1	0	msz	0	0	Zm	1	xs	0	Pg		Rn		Zt																	

Decode fields
msz

Instruction Details

00	ST1B (scalar plus vector)
01	ST1H (scalar plus vector)
10	ST1W (scalar plus vector)
11	ST1D (scalar plus vector)

SVE 64-bit scatter store (scalar plus unpacked 32-bit scaled offsets)

These instructions are under [SVE Memory - Scatter with Optional Sign Extend](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
1	1	1	0	0	1	0	msz	0	1	Zm	1	xs	0	Pg		Rn		Zt																	

Decode fields
msz

Instruction Details

00	UNALLOCATED
01	ST1H (scalar plus vector)
10	ST1W (scalar plus vector)
11	ST1D (scalar plus vector)

SVE 32-bit scatter store (scalar plus 32-bit unscaled offsets)

These instructions are under [SVE Memory - Scatter with Optional Sign Extend](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	0	0	1	0	msz	1	0	Zm	1	xs	0	Pg		Rn		Zt													

Decode fields msz	Instruction Details
00	ST1B (scalar plus vector)
01	ST1H (scalar plus vector)
10	ST1W (scalar plus vector)
11	UNALLOCATED

SVE 32-bit scatter store (scalar plus 32-bit scaled offsets)

These instructions are under [SVE Memory - Scatter with Optional Sign Extend](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	0	0	1	0	msz	1	1	Zm	1	xs	0	Pg		Rn		Zt													

Decode fields msz	Instruction Details
00	UNALLOCATED
01	ST1H (scalar plus vector)
10	ST1W (scalar plus vector)
11	UNALLOCATED

SVE Memory - Scatter

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1110010		op0																													

Decode fields op0	Instruction details
00	SVE 64-bit scatter store (scalar plus 64-bit unscaled offsets)
01	SVE 64-bit scatter store (scalar plus 64-bit scaled offsets)
10	SVE 64-bit scatter store (vector plus immediate)

SVE 64-bit scatter store (scalar plus 64-bit unscaled offsets)

These instructions are under [SVE Memory - Scatter](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
1	1	1	0	0	1	0	msz	0	0	Zm	1	0	1	Pg	Rn		Zt																

Decode fields**msz****Instruction Details**

00	ST1B (scalar plus vector)
01	ST1H (scalar plus vector)
10	ST1W (scalar plus vector)
11	ST1D (scalar plus vector)

SVE 64-bit scatter store (scalar plus 64-bit scaled offsets)

These instructions are under [SVE Memory - Scatter](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
1	1	1	0	0	1	0	msz	0	1	Zm	1	0	1	Pg	Rn		Zt																

Decode fields**msz****Instruction Details**

00	UNALLOCATED
01	ST1H (scalar plus vector)
10	ST1W (scalar plus vector)
11	ST1D (scalar plus vector)

SVE 64-bit scatter store (vector plus immediate)

These instructions are under [SVE Memory - Scatter](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
1	1	1	0	0	1	0	msz	1	0	imm5	1	0	1	Pg	Zn		Zt																

Decode fields**msz****Instruction Details**

00	ST1B (vector plus immediate)
01	ST1H (vector plus immediate)
10	ST1W (vector plus immediate)
11	ST1D (vector plus immediate)

SVE 32-bit scatter store (vector plus immediate)

These instructions are under [SVE Memory - Scatter](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	0	0	1	0	msz	1	1		imm5	1	0	1	Pg		Zn		Zt												

Decode fields	Instruction Details
msz	
00	ST1B (vector plus immediate)
01	ST1H (vector plus immediate)
10	ST1W (vector plus immediate)
11	UNALLOCATED

SVE Memory - Contiguous Store with Immediate Offset

These instructions are under [SVE encodings](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1110010							op0	op1				111																			

Decode fields	Instruction details
op0	
op1	
00	SVE contiguous non-temporal store (scalar plus immediate)
!= 00	SVE store multiple structures (scalar plus immediate)
0	SVE contiguous store (scalar plus immediate)

SVE contiguous non-temporal store (scalar plus immediate)

These instructions are under [SVE Memory - Contiguous Store with Immediate Offset](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	0	0	1	0	msz	0	0	1	imm4	1	1	1	Pg		Rn		Zt												

Decode fields	Instruction Details
msz	
00	STNT1B (scalar plus immediate, single register)
01	STNT1H (scalar plus immediate, single register)
10	STNT1W (scalar plus immediate, single register)
11	STNT1D (scalar plus immediate, single register)

SVE store multiple structures (scalar plus immediate)

These instructions are under [SVE Memory - Contiguous Store with Immediate Offset](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	0	0	1	0	msz	opc	!= 00	1	imm4	1	1	1	Pg	Rn	Zt														

The following constraints also apply to this encoding: opc != 00 && opc != 00

Decode fields		Instruction Details
msz	opc	
00	01	ST2B (scalar plus immediate)
00	10	ST3B (scalar plus immediate)
00	11	ST4B (scalar plus immediate)
01	01	ST2H (scalar plus immediate)
01	10	ST3H (scalar plus immediate)
01	11	ST4H (scalar plus immediate)
10	01	ST2W (scalar plus immediate)
10	10	ST3W (scalar plus immediate)
10	11	ST4W (scalar plus immediate)
11	01	ST2D (scalar plus immediate)
11	10	ST3D (scalar plus immediate)
11	11	ST4D (scalar plus immediate)

SVE contiguous store (scalar plus immediate)

These instructions are under [SVE Memory - Contiguous Store with Immediate Offset](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	0	0	1	0	msz	opc	0	imm4	1	1	1	Pg	Rn	Zt															

Decode fields		Instruction Details	Feature
msz	opc		
00		ST1B (scalar plus immediate, single register)	-
01		ST1H (scalar plus immediate, single register)	-

Decode fields		Instruction Details	Feature
msz	opc		
10	00	ST1W (scalar plus immediate, single register) â€“ SVE2	FEAT_SVE2p1
10	01	UNALLOCATED	-
10	1x	ST1W (scalar plus immediate, single register) â€“ SVE	-
11	0x	UNALLOCATED	-
11	10	ST1D (scalar plus immediate, single register) â€“ SVE2	FEAT_SVE2p1
11	11	ST1D (scalar plus immediate, single register) â€“ SVE	-

Data Processing -- Immediate

These instructions are under the [top-level](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Decode fields	Instruction details
op0	
00xx	PC-rel. addressing
010x	Add/Subtract (immediate)
0110	Add/Subtract (immediate, with tags)
0111	Min/Max (immediate)
100x	Logical (immediate)
101x	Move wide (immediate)
110x	Bitfield
111x	Extract

PC-rel. addressing

These instructions are under [Data Processing -- Immediate](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
op	imm1	imm0	1	0	0	0	0	immhi																						Rd	

Decode fields	Instruction Details
op	
0	ADR
1	ADRP

Add/subtract (immediate)

These instructions are under [Data Processing -- Immediate](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
s	f	o	p	S	1	0	0	0	1	0	sh											imm12				Rn			Rd		

Decode fields			Instruction Details
sf	op	S	
0	0	0	<u>ADD (immediate)</u> â€“ 32-bit
0	0	1	<u>ADDS (immediate)</u> â€“ 32-bit
0	1	0	<u>SUB (immediate)</u> â€“ 32-bit
0	1	1	<u>SUBS (immediate)</u> â€“ 32-bit
1	0	0	<u>ADD (immediate)</u> â€“ 64-bit
1	0	1	<u>ADDS (immediate)</u> â€“ 64-bit
1	1	0	<u>SUB (immediate)</u> â€“ 64-bit
1	1	1	<u>SUBS (immediate)</u> â€“ 64-bit

Add/subtract (immediate, with tags)

These instructions are under [Data Processing -- Immediate](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	sf	op	S	1	0	0	0	1	1	0	ui	mm	6	op	3	ui	mm	4	Rn		Rd
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---	----	----	---	---	---	---	---	---	---	---	----	----	---	----	---	----	----	---	----	--	----

Decode fields			Instruction Details	Feature
sf	op	S		
0			UNALLOCATED	-
1		1	UNALLOCATED	-
1	0	0	ADDG	FEAT_MTE
1	1	0	SUBG	FEAT_MTE

Min/max (immediate)

These instructions are under Data Processing -- Immediate.

Decode fields				Instruction Details	Feature
sf	op	S	opc		
	0		01xx	UNALLOCATED	-
	0		1xxx	UNALLOCATED	-

Decode fields				Instruction Details	Feature
sf	op	S	opc		
	0	1	00xx	UNALLOCATED	-
	1			UNALLOCATED	-
0	0	0	0000	SMAX (immediate) â€“ 32-bit	FEAT_CSSC
0	0	0	0001	UMAX (immediate) â€“ 32-bit	FEAT_CSSC
0	0	0	0010	SMIN (immediate) â€“ 32-bit	FEAT_CSSC
0	0	0	0011	UMIN (immediate) â€“ 32-bit	FEAT_CSSC
1	0	0	0000	SMAX (immediate) â€“ 64-bit	FEAT_CSSC
1	0	0	0001	UMAX (immediate) â€“ 64-bit	FEAT_CSSC
1	0	0	0010	SMIN (immediate) â€“ 64-bit	FEAT_CSSC
1	0	0	0011	UMIN (immediate) â€“ 64-bit	FEAT_CSSC

Logical (immediate)

These instructions are under [Data Processing -- Immediate](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
sf	opc	1	0	0	1	0	0	N																		Rn		Rd			

Decode fields			Instruction Details
sf	opc	N	
0		1	UNALLOCATED
0	00	0	AND (immediate) â€“ 32-bit
0	01	0	ORR (immediate) â€“ 32-bit
0	10	0	EOR (immediate) â€“ 32-bit
0	11	0	ANDS (immediate) â€“ 32-bit
1	00		AND (immediate) â€“ 64-bit
1	01		ORR (immediate) â€“ 64-bit
1	10		EOR (immediate) â€“ 64-bit
1	11		ANDS (immediate) â€“ 64-bit

Move wide (immediate)

These instructions are under [Data Processing -- Immediate](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
sf	opc	1	0	0	1	0	1	hw																		Rd					

Decode fields			Instruction Details
sf	opc	hw	
	01		UNALLOCATED

Decode fields			Instruction Details							
sf	opc	hw								
0		1x								UNALLOCATED
0	00	0x								MOVN â€“ 32-bit
0	10	0x								MOVZ â€“ 32-bit
0	11	0x								MOVK â€“ 32-bit
1	00									MOVN â€“ 64-bit
1	10									MOVZ â€“ 64-bit
1	11									MOVK â€“ 64-bit

Bitfield

These instructions are under [Data Processing -- Immediate](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
sf	opc	1	0	0	1	1	0	N																				Rn		Rd	

Decode fields			Instruction Details							
sf	opc	N								
	11									UNALLOCATED
0		1								UNALLOCATED
0	00	0								SBFM â€“ 32-bit
0	01	0								BFM â€“ 32-bit
0	10	0								UBFM â€“ 32-bit
1		0								UNALLOCATED
1	00	1								SBFM â€“ 64-bit
1	01	1								BFM â€“ 64-bit
1	10	1								UBFM â€“ 64-bit

Extract

These instructions are under [Data Processing -- Immediate](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
sf	op21	1	0	0	1	1	1	N	00																	Rm		imms		Rn		Rd	

Decode fields					Instruction Details					
sf	op21	N	o0	imms						
	x1									UNALLOCATED
	00		1							UNALLOCATED
	1x									UNALLOCATED
0				1xxxxx						UNALLOCATED

Decode fields					Instruction Details
sf	op21	N	o0	imms	
0		1			UNALLOCATED
0	00	0	0	0xxxxxx	EXTR “ 32-bit
1		0			UNALLOCATED
1	00	1	0		EXTR “ 64-bit

Branches, Exception Generating and System instructions

These instructions are under the [top-level](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
op0	101																												op2		

Decode fields			Instruction details
op0	op1	op2	
010	0xxxxxxxxxxxxxxx		Conditional branch (immediate)
110	00xxxxxxxxxxxxxx		Exception generation
110	01000000110001		System instructions with register argument
110	01000000110010	11111	Hints
110	01000000110011		Barriers
110	0100000xxx0100		PSTATE
110	0100100xxxxxxx		System with result
110	0100x01xxxxxxx		System instructions
110	0100x1xxxxxxx		System register move
110	0101x01xxxxxxx		System pair instructions
110	0101x1xxxxxxx		System register pair move
110	1xxxxxxxxxxxxxx		Unconditional branch (register)
x00			Unconditional branch (immediate)
x01	0xxxxxxxxxxxxxx		Compare and branch (immediate)
x01	1xxxxxxxxxxxxxx		Test and branch (immediate)

Conditional branch (immediate)

These instructions are under [Branches, Exception Generating and System instructions](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	1	0	1	0	01																			imm19	o0	cond			

Decode fields		Instruction Details	Feature
o1	o0		
0	0	B.cond	-
0	1	BC.cond	FEAT_HBC
1		UNALLOCATED	-

Exception generation

These instructions are under [Branches, Exception Generating and System instructions](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	1	0	1	0	0	opc																			op2	LL			

Decode fields			Instruction Details	Feature
opc	op2	LL		
	001		UNALLOCATED	-
	01x		UNALLOCATED	-
	1xx		UNALLOCATED	-
000	000	00	UNALLOCATED	-
000	000	01	SVC	-
000	000	10	HVC	-
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	00	TCANCEL	FEAT_TME
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

These instructions are under [Branches, Exception Generating and System instructions](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	1	0	1	0	1	0	0	0	0	0	1	1	0	0	0	1	CRm	op2	Rt										

Decode fields		Instruction Details		Feature
CRm	op2			
!= 0000		UNALLOCATED		-
0000	000	WFET		FEAT_WFxT
0000	001	WFIT		FEAT_WFxT
0000	01x	UNALLOCATED		-
0000	1xx	UNALLOCATED		-

Hints

These instructions are under [Branches, Exception Generating and System instructions](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	1	0	1	0	1	0	0	0	0	0	0	1	1	0	0	1	0	CRm	op2	1	1	1	1	1	1	1	1	1	

Decode fields		Instruction Details		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 â€” PACIA1716		FEAT_PAuth
0001	010	PACIB, PACIB1716, PACIBSP, PACIBZ, PACIZB â€” PACIB1716		FEAT_PAuth
0001	100	AUTIA, AUTIA1716, AUTIASP, AUTIAZ, AUTIZA â€” AUTIA1716		FEAT_PAuth
0001	110	AUTIB, AUTIB1716, AUTIBSP, AUTIBZ, AUTIZB â€” AUTIB1716		FEAT_PAuth

Decode fields		Instruction Details		Feature
CRm	op2			
0010	000	ESB		FEAT_RAS
0010	001	PSB CSYNC		FEAT_SPE
0010	010	TSB CSYNC		FEAT_TRF
0010	011	GCSB DSYNC		FEAT_GCS
0010	100	CSDB		-
0010	110	CLRBHB		FEAT_CLRBHB
0011	000	PACIA, PACIA1716, PACIASP, PACIAZ, PACIZA â€” PACIAZ		FEAT_PAuth
0011	001	PACIA, PACIA1716, PACIASP, PACIAZ, PACIZA â€” PACIASP		FEAT_PAuth
0011	010	PACIB, PACIB1716, PACIBSP, PACIBZ, PACIZB â€” PACIBZ		FEAT_PAuth
0011	011	PACIB, PACIB1716, PACIBSP, PACIBZ, PACIZB â€” PACIBSP		FEAT_PAuth
0011	100	AUTIA, AUTIA1716, AUTIASP, AUTIAZ, AUTIZA â€” AUTIAZ		FEAT_PAuth
0011	101	AUTIA, AUTIA1716, AUTIASP, AUTIAZ, AUTIZA â€” AUTIASP		FEAT_PAuth
0011	110	AUTIB, AUTIB1716, AUTIBSP, AUTIBZ, AUTIZB â€” AUTIBZ		FEAT_PAuth
0011	111	AUTIB, AUTIB1716, AUTIBSP, AUTIBZ, AUTIZB â€” AUTIBSP		FEAT_PAuth
0100	xx0	BTI		FEAT_BTI
0101	000	CHKFEAT		FEAT_CHK

Barriers

These instructions are under [Branches, Exception Generating and System instructions](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	1	0	1	0	1	0	0	0	0	0	0	0	1	1	0	0	1	1	CRm	op2	Rt								

Decode fields			Instruction Details		Feature
CRm	op2	Rt			
	000		UNALLOCATED		-
	001	!= 11111	UNALLOCATED		-
	010	11111	CLREX		-
	100	11111	DSB â€” memory barrier		-
	101	11111	DMB		-
	110	11111	ISB		-

Decode fields			Instruction Details		Feature
CRm	op2	Rt			
	111	$\neq 11111$	UNALLOCATED		-
	111	11111	SB		FEAT_SB
xx0x	001	11111	UNALLOCATED		-
xx10	001	11111	DSB â€“ Memory nXS barrier		FEAT_XS
xx11	001	11111	UNALLOCATED		-
0000	011	11111	TCOMMIT		FEAT_TME
0001	011		UNALLOCATED		-
001x	011		UNALLOCATED		-
01xx	011		UNALLOCATED		-
1xxx	011		UNALLOCATED		-

PSTATE

These instructions are under [Branches, Exception Generating and System instructions](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0	CRm	op2	Rt									

Decode fields			Instruction Details		Feature
op1	op2	Rt			
		$\neq 11111$	UNALLOCATED		-
		11111	MSR (immediate)		-
000	000	11111	CFINV		FEAT_FlagM
000	001	11111	XAFLAG		FEAT_FlagM2
000	010	11111	AXFLAG		FEAT_FlagM2

System with result

These instructions are under [Branches, Exception Generating and System instructions](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0	CRn	CRm	op2	Rt								

Decode fields				Instruction Details		Feature
op1	CRn	CRm	op2			
$\neq 011$				UNALLOCATED		-
011	$\neq 0011$			UNALLOCATED		-
011	0011		$\neq 011$	UNALLOCATED		-

op1	Decode fields			op2	Instruction Details	Feature
	CRn	CRm				
011	0011	!= 000x	011		UNALLOCATED	-
011	0011	0000	011		TSTART	FEAT_TME
011	0011	0001	011		TTEST	FEAT_TME

System instructions

These instructions are under [Branches, Exception Generating and System instructions](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	1	0	1	0	1	0	0	L	0	1	op1		CRn		CRm		op2		Rt										

Decode fields		Instruction Details
L		
0		SYS
1		SYSL

System register move

These instructions are under [Branches, Exception Generating and System instructions](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
1	1	0	1	0	1	0	1	0	0	L	1	0	0	op1		CRn		CRm		op2		Rt										

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

System pair instructions

These instructions are under [Branches, Exception Generating and System instructions](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	1	0	1	0	1	0	1	L	0	1	op1		CRn		CRm		op2		Rt										

Decode fields		Instruction Details	Feature
L			
0		SYSP	FEAT_SYSINSTR128
1		UNALLOCATED	-

System register pair move

These instructions are under [Branches, Exception Generating and System instructions](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	1	0	1	0	1	L	1	0	0	op1		CRn		CRm		op2		Rt											

Decode fields L	Instruction Details	Feature
0	MSRR	FEAT_SYSREG128
1	MRRS	FEAT_SYSREG128

Unconditional branch (register)

These instructions are under [Branches, Exception Generating and System instructions](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	1	0	1	1	opc		op2		op3		Rn		op4																

Decode fields opc	Decode fields op2	Decode fields op3	Rn	op4	Instruction Details	Feature
	!= 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 â€” key A, zero modifier	FEAT_PAuth
0000	11111	000011		!= 11111	UNALLOCATED	-
0000	11111	000011		11111	BRAA, BRAAZ, BRAB, BRABZ â€” key B, zero modifier	FEAT_PAuth
0000	11111	0001xx			UNALLOCATED	-
0000	11111	001xxx			UNALLOCATED	-
0000	11111	01xxxx			UNALLOCATED	-
0000	11111	1xxxxx			UNALLOCATED	-

Decode fields					Instruction Details	Feature
opc	op2	op3	Rn	op4		
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 â€” key A, zero modifier	FEAT_PAuth
0001	11111	000011		!= 11111	UNALLOCATED	-
0001	11111	000011		11111	BLRAA, BLRAAZ, BLRAB, BLRABZ â€” key B, zero modifier	FEAT_PAuth
0001	11111	0001xx			UNALLOCATED	-
0001	11111	001xxx			UNALLOCATED	-
0001	11111	01xxxx			UNALLOCATED	-
0001	11111	1xxxxx			UNALLOCATED	-
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	UNALLOCATED	-
0010	11111	000010	11111	!= 11111	UNALLOCATED	-
0010	11111	000010	11111	11111	RETAAC, RETAB â€” RETAAC	FEAT_PAuth
0010	11111	000011	!= 11111	!= 11111	UNALLOCATED	-
0010	11111	000011	!= 11111	11111	UNALLOCATED	-
0010	11111	000011	11111	!= 11111	UNALLOCATED	-
0010	11111	000011	11111	11111	RETAAC, RETAB â€” RETAB	FEAT_PAuth
0010	11111	0001xx			UNALLOCATED	-
0010	11111	001xxx			UNALLOCATED	-
0010	11111	01xxxx			UNALLOCATED	-
0010	11111	1xxxxx			UNALLOCATED	-

Decode fields					Instruction Details	Feature
opc	op2	op3	Rn	op4		
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 â€” ERETAA	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 â€” ERETAB	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	-
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 â€” key A, register modifier	FEAT_PAuth

Decode fields					Instruction Details	Feature
opc	op2	op3	Rn	op4		
1000	11111	000011			BRAA , BRAAZ , BRAB , BRABZ â€“ key B, register modifier	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 â€“ key A, register modifier	FEAT_PAuth
1001	11111	000011			BLRAA , BLRAAZ , BLRAB , BLRABZ â€“ key B, register modifier	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)

These instructions are under [Branches, Exception Generating and System instructions](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
op	0	0	1	0	1																												imm26

Decode fields		Instruction Details
op		
0		B
1		BL

Compare and branch (immediate)

These instructions are under [Branches, Exception Generating and System instructions](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
sf	0	1	1	0	1	0	op																						Rt		

Decode fields		Instruction Details
sf	op	

0	0	CBZ â€“ 32-bit
0	1	CBNZ â€“ 32-bit
1	0	CBZ â€“ 64-bit
1	1	CBNZ â€“ 64-bit

Test and branch (immediate)

These instructions are under [Branches, Exception Generating and System instructions](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
b5	0	1	1	0	1	1	op																						Rt		

Decode fields		Instruction Details
op		

0	TBZ
1	TBNZ

Loads and Stores

These instructions are under the [top-level](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
op0	1	op1	0																												

Decode fields			Instruction details
op0	op1	op2	
0x00	0	00xxxxxxxxxxxxxx	Compare and swap pair
0x00	1	00x000000xxxxxx	Advanced SIMD load/store multiple structures
0x00	1	00xxxxxx1xxxxxxx	UNALLOCATED
0x00	1	01x0xxxxxxxxxxxxx	Advanced SIMD load/store multiple structures (post-indexed)
0x00	1	0xx1xxxxxxxxxxxxx	UNALLOCATED
0x00	1	10xx0000xxxxxxx	Advanced SIMD load/store single structure
0x00	1	11xxxxxxxxxxxxxx	Advanced SIMD load/store single structure (post-indexed)
0x00	1	x0xx1xxxxxxxxxxx	UNALLOCATED
0x00	1	x0xx1xxxxxxxxxxx	UNALLOCATED

0x00	1	x0xxxx1xxxxxxxxx	UNALLOCATED
0x00	1	x0xxxx1xxxxxxxxx	UNALLOCATED
0x01	0	1xx1xxxx000010	RCW compare and swap
0x01	0	1xx1xxxx000011	RCW compare and swap pair
0x01	0	1xx1xxxxxxxxx00	128-bit atomic memory operations
1101	0	1000111110xxx11	GCS load/store
1101	0	1xx1xxxxxxxxxxxxx	Load/store memory tags
1x00	0	00x1xxxxxxxxxxxxx	Load/store exclusive pair
1x00	1		UNALLOCATED
xx00	0	00x0xxxxxxxxxxxxx	Load/store exclusive register
xx00	0	01x0xxxxxxxxxxxxx	Load/store ordered
xx00	0	01x1xxxxxxxxxxxxx	Compare and swap
xx01	0	10x0xxxxxxxxxx10	LDIAPP/STILP
xx01	0	11x000000000010	LDAPR/STLR (writeback)
xx01	0	1xx0xxxxxxxxxx00	LDAPR/STLR (unscaled immediate)
xx01	1	1xx0xxxxxxxxxx10	LDAPR/STLR (SIMD&FP)
xx01		0xxxxxxxxxxxxxx	Load register (literal)
xx01		1xx0xxxxxxxxxx01	Memory Copy and Memory Set
xx10		0xxxxxxxxxxxxxx	Load/store no-allocate pair (offset)
xx10		01xxxxxxxxxxxxxx	Load/store register pair (post-indexed)
xx10		10xxxxxxxxxxxxxx	Load/store register pair (offset)
xx10		11xxxxxxxxxxxxxx	Load/store register pair (pre-indexed)
xx11		0xx0xxxxxxxxxx00	Load/store register (unscaled immediate)
xx11		0xx0xxxxxxxxxx01	Load/store register (immediate post-indexed)
xx11		0xx0xxxxxxxxxx10	Load/store register (unprivileged)
xx11		0xx0xxxxxxxxxx11	Load/store register (immediate pre-indexed)
xx11		0xx1xxxxxxxxxx00	Atomic memory operations
xx11		0xx1xxxxxxxxxx10	Load/store register (register offset)
xx11		0xx1xxxxxxxxxx11	Load/store register (pac)
xx11		1xxxxxxxxxxxxxx	Load/store register (unsigned immediate)

Compare and swap pair

These instructions are under [Loads and Stores](#).

Decode fields				Instruction Details	Feature
sz	L	o0	Rt2		
			!= 11111	UNALLOCATED	-
0	0	0	11111	CASP, CASPA, CASPAL, CASPL â€” 32-bit CASP	FEAT_LSE
0	0	1	11111	CASP, CASPA, CASPAL, CASPL â€” 32-bit CASPL	FEAT_LSE
0	1	0	11111	CASP, CASPA, CASPAL, CASPL â€” 32-bit CASPA	FEAT_LSE
0	1	1	11111	CASP, CASPA, CASPAL, CASPL â€” 32-bit CASPAL	FEAT_LSE
1	0	0	11111	CASP, CASPA, CASPAL, CASPL â€” 64-bit CASP	FEAT_LSE
1	0	1	11111	CASP, CASPA, CASPAL, CASPL â€” 64-bit CASPL	FEAT_LSE
1	1	0	11111	CASP, CASPA, CASPAL, CASPL â€” 64-bit CASPA	FEAT_LSE
1	1	1	11111	CASP, CASPA, CASPAL, CASPL â€” 64-bit CASPAL	FEAT_LSE

Advanced SIMD load/store multiple structures

These instructions are under [Loads and Stores](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	Q	0	0	1	1	0	0	0	L	0	0	0	0	0	opcode	size	Rn	Rt													

Decode fields		Instruction Details
L	opcode	
0	0000	ST4 (multiple structures)
0	0001	UNALLOCATED
0	0010	ST1 (multiple structures) â€” four registers
0	0011	UNALLOCATED
0	0100	ST3 (multiple structures)
0	0101	UNALLOCATED
0	0110	ST1 (multiple structures) â€” three registers
0	0111	ST1 (multiple structures) â€” one register
0	1000	ST2 (multiple structures)
0	1001	UNALLOCATED
0	1010	ST1 (multiple structures) â€” two registers
0	1011	UNALLOCATED
0	11xx	UNALLOCATED

Decode fields		Instruction Details
L	opcode	
1	0000	LD4 (multiple structures)
1	0001	UNALLOCATED
1	0010	LD1 (multiple structures) â€“ four registers
1	0011	UNALLOCATED
1	0100	LD3 (multiple structures)
1	0101	UNALLOCATED
1	0110	LD1 (multiple structures) â€“ three registers
1	0111	LD1 (multiple structures) â€“ one register
1	1000	LD2 (multiple structures)
1	1001	UNALLOCATED
1	1010	LD1 (multiple structures) â€“ two registers
1	1011	UNALLOCATED
1	11xx	UNALLOCATED

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

These instructions are under [Loads and Stores](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	Q	0	0	1	1	0	0	1	L	0	Rm	opcode	size	Rn	Rt																

Decode fields			Instruction Details
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) â€“ register offset
0	!= 11111	0010	ST1 (multiple structures) â€“ four registers, register offset
0	!= 11111	0100	ST3 (multiple structures) â€“ register offset
0	!= 11111	0110	ST1 (multiple structures) â€“ three registers, register offset
0	!= 11111	0111	ST1 (multiple structures) â€“ one register, register offset
0	!= 11111	1000	ST2 (multiple structures) â€“ register offset

Decode fields			Instruction Details
L	Rm	opcode	
0	$\neq 11111$	1010	ST1 (multiple structures) â€“ two registers, register offset
0	11111	0000	ST4 (multiple structures) â€“ immediate offset
0	11111	0010	ST1 (multiple structures) â€“ four registers, immediate offset
0	11111	0100	ST3 (multiple structures) â€“ immediate offset
0	11111	0110	ST1 (multiple structures) â€“ three registers, immediate offset
0	11111	0111	ST1 (multiple structures) â€“ one register, immediate offset
0	11111	1000	ST2 (multiple structures) â€“ immediate offset
0	11111	1010	ST1 (multiple structures) â€“ two registers, immediate offset
1		0001	UNALLOCATED
1		0011	UNALLOCATED
1		0101	UNALLOCATED
1		1001	UNALLOCATED
1		1011	UNALLOCATED
1		11xx	UNALLOCATED
1	$\neq 11111$	0000	LD4 (multiple structures) â€“ register offset
1	$\neq 11111$	0010	LD1 (multiple structures) â€“ four registers, register offset
1	$\neq 11111$	0100	LD3 (multiple structures) â€“ register offset
1	$\neq 11111$	0110	LD1 (multiple structures) â€“ three registers, register offset
1	$\neq 11111$	0111	LD1 (multiple structures) â€“ one register, register offset
1	$\neq 11111$	1000	LD2 (multiple structures) â€“ register offset
1	$\neq 11111$	1010	LD1 (multiple structures) â€“ two registers, register offset
1	11111	0000	LD4 (multiple structures) â€“ immediate offset
1	11111	0010	LD1 (multiple structures) â€“ four registers, immediate offset
1	11111	0100	LD3 (multiple structures) â€“ immediate offset
1	11111	0110	LD1 (multiple structures) â€“ three registers, immediate offset

Decode fields			Instruction Details	
L	Rm	opcode		
1	11111	0111	LD1 (multiple structures) â€“ one register, immediate offset	
1	11111	1000	LD2 (multiple structures) â€“ immediate offset	
1	11111	1010	LD1 (multiple structures) â€“ two registers, immediate offset	

Advanced SIMD load/store single structure

These instructions are under [Loads and Stores](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	Q	0	0	1	1	0	1	0	L	R	0	0	0	0	o2	opcode	S	size		Rn		Rt									

L	R	Decode fields			Instruction Details		Feature
		o2	opcode	S	size		
	0	1	x00	0	00	UNALLOCATED	-
	0	1	x00	0	1x	UNALLOCATED	-
	0	1	x00	1		UNALLOCATED	-
	0	1	x01			UNALLOCATED	-
	0	1	x1x			UNALLOCATED	-
	1	1				UNALLOCATED	-
0		0	11x			UNALLOCATED	-
0	0	0	000			ST1 (single structure) â€“ 8-bit	-
0	0	0	001			ST3 (single structure) â€“ 8-bit	-
0	0	0	010		x0	ST1 (single structure) â€“ 16-bit	-
0	0	0	010		x1	UNALLOCATED	-
0	0	0	011		x0	ST3 (single structure) â€“ 16-bit	-
0	0	0	011		x1	UNALLOCATED	-
0	0	0	100		00	ST1 (single structure) â€“ 32-bit	-
0	0	0	100		1x	UNALLOCATED	-

Decode fields						Instruction Details	Feature
L	R	o2	opcode	S	size		
0	0	0	100	0	01	ST1 (single structure) â€“ 64-bit	-
0	0	0	100	1	01	UNALLOCATED	-
0	0	0	101		00	ST3 (single structure) â€“ 32-bit	-
0	0	0	101		10	UNALLOCATED	-
0	0	0	101	0	01	ST3 (single structure) â€“ 64-bit	-
0	0	0	101	0	11	UNALLOCATED	-
0	0	0	101	1	x1	UNALLOCATED	-
0	0	1	100	0	01	STL1 (SIMD&FP)	FEAT_LRCPC3
0	1	0	000			ST2 (single structure) â€“ 8-bit	-
0	1	0	001			ST4 (single structure) â€“ 8-bit	-
0	1	0	010		x0	ST2 (single structure) â€“ 16-bit	-
0	1	0	010		x1	UNALLOCATED	-
0	1	0	011		x0	ST4 (single structure) â€“ 16-bit	-
0	1	0	011		x1	UNALLOCATED	-
0	1	0	100		00	ST2 (single structure) â€“ 32-bit	-
0	1	0	100		10	UNALLOCATED	-
0	1	0	100	0	01	ST2 (single structure) â€“ 64-bit	-
0	1	0	100	0	11	UNALLOCATED	-
0	1	0	100	1	x1	UNALLOCATED	-
0	1	0	101		00	ST4 (single structure) â€“ 32-bit	-
0	1	0	101		10	UNALLOCATED	-
0	1	0	101	0	01	ST4 (single structure) â€“ 64-bit	-

Decode fields						Instruction Details	Feature
L	R	o2	opcode	S	size		
0	1	0	101	0	11	UNALLOCATED	-
0	1	0	101	1	x1	UNALLOCATED	-
1	0	0	000			LD1 (single structure) â€” 8-bit	-
1	0	0	001			LD3 (single structure) â€” 8-bit	-
1	0	0	010		x0	LD1 (single structure) â€” 16-bit	-
1	0	0	010		x1	UNALLOCATED	-
1	0	0	011		x0	LD3 (single structure) â€” 16-bit	-
1	0	0	011		x1	UNALLOCATED	-
1	0	0	100		00	LD1 (single structure) â€” 32-bit	-
1	0	0	100		1x	UNALLOCATED	-
1	0	0	100	0	01	LD1 (single structure) â€” 64-bit	-
1	0	0	100	1	01	UNALLOCATED	-
1	0	0	101		00	LD3 (single structure) â€” 32-bit	-
1	0	0	101		10	UNALLOCATED	-
1	0	0	101	0	01	LD3 (single structure) â€” 64-bit	-
1	0	0	101	0	11	UNALLOCATED	-
1	0	0	101	1	x1	UNALLOCATED	-
1	0	0	110	0		LD1R	-
1	0	0	110	1		UNALLOCATED	-
1	0	0	111	0		LD3R	-
1	0	0	111	1		UNALLOCATED	-
1	0	1	100	0	01	LDAP1 (SIMD&FP)	FEAT_LRCPC3
1	1	0	000			LD2 (single structure) â€” 8-bit	-

Decode fields					Instruction Details		Feature
L	R	o2	opcode	S	size		
1	1	0	001			LD4 (single structure) 8-bit	-
1	1	0	010		x0	LD2 (single structure) 16-bit	-
1	1	0	010		x1	UNALLOCATED	-
1	1	0	011		x0	LD4 (single structure) 16-bit	-
1	1	0	011		x1	UNALLOCATED	-
1	1	0	100		00	LD2 (single structure) 32-bit	-
1	1	0	100		10	UNALLOCATED	-
1	1	0	100	0	01	LD2 (single structure) 64-bit	-
1	1	0	100	0	11	UNALLOCATED	-
1	1	0	100	1	x1	UNALLOCATED	-
1	1	0	101		00	LD4 (single structure) 32-bit	-
1	1	0	101		10	UNALLOCATED	-
1	1	0	101	0	01	LD4 (single structure) 64-bit	-
1	1	0	101	0	11	UNALLOCATED	-
1	1	0	101	1	x1	UNALLOCATED	-
1	1	0	110	0		LD2R	-
1	1	0	110	1		UNALLOCATED	-
1	1	0	111	0		LD4R	-
1	1	0	111	1		UNALLOCATED	-

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

These instructions are under [Loads and Stores](#).

Decode fields						Instruction Details
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) â€“ 8-bit, register offset
0	0	!= 11111	001			ST3 (single structure) â€“ 8-bit, register offset
0	0	!= 11111	010		x0	ST1 (single structure) â€“ 16-bit, register offset
0	0	!= 11111	011		x0	ST3 (single structure) â€“ 16-bit, register offset
0	0	!= 11111	100		00	ST1 (single structure) â€“ 32-bit, register offset
0	0	!= 11111	100	0	01	ST1 (single structure) â€“ 64-bit, register offset
0	0	!= 11111	101		00	ST3 (single structure) â€“ 32-bit, register offset
0	0	!= 11111	101	0	01	ST3 (single structure) â€“ 64-bit, register offset
0	0	11111	000			ST1 (single structure) â€“ 8-bit, immediate offset
0	0	11111	001			ST3 (single structure) â€“ 8-bit, immediate offset
0	0	11111	010		x0	ST1 (single structure) â€“ 16-bit, immediate offset
0	0	11111	011		x0	ST3 (single structure) â€“ 16-bit, immediate offset
0	0	11111	100		00	ST1 (single structure) â€“ 32-bit, immediate offset
0	0	11111	100	0	01	ST1 (single structure) â€“ 64-bit, immediate offset
0	0	11111	101		00	ST3 (single structure) â€“ 32-bit, immediate offset
0	0	11111	101	0	01	ST3 (single structure) â€“ 64-bit, immediate offset
0	1		010		x1	UNALLOCATED
0	1		011		x1	UNALLOCATED

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

Decode fields						Instruction Details
L	R	Rm	opcode	S	size	
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) â€“ 8-bit, register offset
1	0	!= 11111	001			LD3 (single structure) â€“ 8-bit, register offset
1	0	!= 11111	010		x0	LD1 (single structure) â€“ 16-bit, register offset
1	0	!= 11111	011		x0	LD3 (single structure) â€“ 16-bit, register offset
1	0	!= 11111	100		00	LD1 (single structure) â€“ 32-bit, register offset
1	0	!= 11111	100	0	01	LD1 (single structure) â€“ 64-bit, register offset
1	0	!= 11111	101		00	LD3 (single structure) â€“ 32-bit, register offset
1	0	!= 11111	101	0	01	LD3 (single structure) â€“ 64-bit, register offset
1	0	!= 11111	110	0		LD1R â€“ register offset
1	0	!= 11111	111	0		LD3R â€“ register offset
1	0	11111	000			LD1 (single structure) â€“ 8-bit, immediate offset
1	0	11111	001			LD3 (single structure) â€“ 8-bit, immediate offset
1	0	11111	010		x0	LD1 (single structure) â€“ 16-bit, immediate offset
1	0	11111	011		x0	LD3 (single structure) â€“ 16-bit, immediate offset
1	0	11111	100		00	LD1 (single structure) â€“ 32-bit, immediate offset
1	0	11111	100	0	01	LD1 (single structure) â€“ 64-bit, immediate offset
1	0	11111	101		00	LD3 (single structure) â€“ 32-bit, immediate offset
1	0	11111	101	0	01	LD3 (single structure) â€“ 64-bit, immediate offset
1	0	11111	110	0		LD1R â€“ immediate offset
1	0	11111	111	0		LD3R â€“ immediate offset
1	1		010		x1	UNALLOCATED
1	1		011		x1	UNALLOCATED

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

Decode fields						Instruction Details	
L	R	Rm	opcode	S	size		
1	1	11111	110	0		LD2R	" immediate offset
1	1	11111	111	0		LD4R	" immediate offset

RCW compare and swap

These instructions are under [Loads and Stores](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	S	0	1	1	0	0	1	A	R	1		Rs	0	0	0	0	1	0	Rn		Rt										

Decode fields			Instruction Details		Feature
S	A	R			
0	0	0	RCWCAS , RCWCASA , RCWCASL , RCWCASAL " RCWCAS		FEAT_THE
0	0	1	RCWCAS , RCWCASA , RCWCASL , RCWCASAL " RCWCASL		FEAT_THE
0	1	0	RCWCAS , RCWCASA , RCWCASL , RCWCASAL " RCWCASA		FEAT_THE
0	1	1	RCWCAS , RCWCASA , RCWCASL , RCWCASAL " RCWCASAL		FEAT_THE
1	0	0	RCWSCAS , RCWSCASA , RCWSCASL , RCWSCASAL " RCWSCAS		FEAT_THE
1	0	1	RCWSCAS , RCWSCASA , RCWSCASL , RCWSCASAL " RCWSCASL		FEAT_THE
1	1	0	RCWSCAS , RCWSCASA , RCWSCASL , RCWSCASAL " RCWSCASA		FEAT_THE
1	1	1	RCWSCAS , RCWSCASA , RCWSCASL , RCWSCASAL " RCWSCASAL		FEAT_THE

RCW compare and swap pair

These instructions are under [Loads and Stores](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	S	0	1	1	0	0	1	A	R	1		Rs	0	0	0	0	1	1	Rn		Rt										

Decode fields			Instruction Details		Feature
S	A	R			
0	0	0	RCWCASP , RCWCASPA , RCWCASPL , RCWCASPAL " RCWCASP		FEAT_D128 && FEAT_THE

Decode fields			Instruction Details			Feature
S	A	R				
0	0	1	RCWCASP , RCWCASPA , RCWCASPL , RCWCASPAL â€” RCWCASPL			FEAT_D128 && FEAT_THE
0	1	0	RCWCASP , RCWCASPA , RCWCASPL , RCWCASPAL â€” RCWCASPA			FEAT_D128 && FEAT_THE
0	1	1	RCWCASP , RCWCASPA , RCWCASPL , RCWCASPAL â€” RCWCASPAL			FEAT_D128 && FEAT_THE
1	0	0	RCWSCASP , RCWSCASPA , RCWSCASPL , RCWSCASPAL â€” RCWSCASP			FEAT_D128 && FEAT_THE
1	0	1	RCWSCASP , RCWSCASPA , RCWSCASPL , RCWSCASPAL â€” RCWSCASPL			FEAT_D128 && FEAT_THE
1	1	0	RCWSCASP , RCWSCASPA , RCWSCASPL , RCWSCASPAL â€” RCWSCASPA			FEAT_D128 && FEAT_THE
1	1	1	RCWSCASP , RCWSCASPA , RCWSCASPL , RCWSCASPAL â€” RCWSCASPAL			FEAT_D128 && FEAT_THE

128-bit atomic memory operations

These instructions are under [Loads and Stores](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	S	0	1	1	0	0	1	A	R	1		Rt2	o3	opc	0	0		Rn		Rt											

Decode fields					Instruction Details			Feature
S	A	R	o3	opc				
0				1xx	UNALLOCATED			-
0			0	0x0	UNALLOCATED			-
0	0	0	0	001	LDCLRP , LDCLRPA , LDCLRPAL , LDCLRPL â€” LDCLRP			FEAT_LSE128
0	0	0	0	011	LDSETP , LDSETPA , LDSETPAL , LDSETPL â€” LDSETP			FEAT_LSE128
0	0	0	1	000	SWPP , SWPPA , SWPPAL , SWPPL â€” SWPP			FEAT_LSE128

Decode fields					Instruction Details	Feature
S	A	R	o3	opc		
0	0	0	1	001	RCWCLRP , RCWCLRPA , RCWCLRPL , RCWCLRPAL â€” RCWCLRP	FEAT_D128 && FEAT_THE
0	0	0	1	010	RCWSWPP , RCWSWPRA , RCWSWPPL , RCWSWPAL â€” RCWSWPP	FEAT_D128 && FEAT_THE
0	0	0	1	011	RCWSETP , RCWSETPA , RCWSETPL , RCWSETPAL â€” RCWSETP	FEAT_D128 && FEAT_THE
0	0	1	0	001	LDCLRPA , LDCLRPL , LDCLRPAL , LDCLRPL â€” LDCLRPL	FEAT_LSE128
0	0	1	0	011	LDSETP , LDSETPA , LDSETPAL , LDSETPPL â€” LDSETPL	FEAT_LSE128
0	0	1	1	000	SWPP , SWPPA , SWPPAL , SWPPL â€” SWPPL	FEAT_LSE128
0	0	1	1	001	RCWCLRP , RCWCLRPA , RCWCLRPL , RCWCLRPAL â€” RCWCLRPL	FEAT_D128 && FEAT_THE
0	0	1	1	010	RCWSWPP , RCWSWPRA , RCWSWPPL , RCWSWPAL â€” RCWSWPPL	FEAT_D128 && FEAT_THE
0	0	1	1	011	RCWSETP , RCWSETPA , RCWSETPL , RCWSETPAL â€” RCWSETPL	FEAT_D128 && FEAT_THE
0	1	0	0	001	LDCLRPA , LDCLRPL , LDCLRPAL , LDCLRPL â€” LDCLRPA	FEAT_LSE128
0	1	0	0	011	LDSETP , LDSETPA , LDSETPAL , LDSETPPL â€” LDSETPA	FEAT_LSE128
0	1	0	1	000	SWPP , SWPPA , SWPPAL , SWPPL â€” SWPPA	FEAT_LSE128
0	1	0	1	001	RCWCLRP , RCWCLRPA , RCWCLRPL , RCWCLRPAL â€” RCWCLRPA	FEAT_D128 && FEAT_THE

Decode fields					Instruction Details	Feature
S	A	R	o3	opc		
0	1	0	1	010	RCWSWPP , RCWSWPRA , RCWSWPPL , RCWSWPAL â€” RCWSWPAL	FEAT_D128 && FEAT_THE
0	1	0	1	011	RCWSETP , RCWSETPA , RCWSETPL , RCWSETPAL â€” RCWSETPA	FEAT_D128 && FEAT_THE
0	1	1	0	001	LDCLRPA , LDCLRPL , LDCLRPA , LDCLRPL â€” LDCLRPA	FEAT_LSE128
0	1	1	0	011	LDSETP , LDSETPA , LDSETPAL , LDSETPPL â€” LDSETPAL	FEAT_LSE128
0	1	1	1	000	SWPP , SWPPA , SWPPAL , SWPPL â€” SWPPAL	FEAT_LSE128
0	1	1	1	001	RCWCLRP , RCWCLRPA , RCWCLRPL , RCWCLRPAL â€” RCWCLRPAL	FEAT_D128 && FEAT_THE
0	1	1	1	010	RCWSWPP , RCWSWPRA , RCWSWPPL , RCWSWPAL â€” RCWSWPAL	FEAT_D128 && FEAT_THE
0	1	1	1	011	RCWSETP , RCWSETPA , RCWSETPL , RCWSETPAL â€” RCWSETPAL	FEAT_D128 && FEAT_THE
1			0		UNALLOCATED	-
1			1	000	UNALLOCATED	-
1			1	1xx	UNALLOCATED	-
1	0	0	1	001	RCWSCLRP , RCWSCLRPA , RCWSCLRPL , RCWSCLRPAL â€” RCWSCLRP	FEAT_D128 && FEAT_THE
1	0	0	1	010	RCWSSWPP , RCWSSWPRA , RCWSSWPPL , RCWSSWPAL â€” RCWSSWPP	FEAT_D128 && FEAT_THE
1	0	0	1	011	RCWSSETP , RCWSSETPA , RCWSSETPL , RCWSSETPAL â€” RCWSSETP	FEAT_D128 && FEAT_THE

Decode fields					Instruction Details	Feature
S	A	R	o3	opc		
1	0	1	1	001	RCWSCLRP , RCWSCLRPA , RCWSCLRPL , RCWSCLRPAL â€” RCWSCLRPL	FEAT_D128 && FEAT_THE
1	0	1	1	010	RCWSSWPP , RCWSSWPRA , RCWSSWPPL , RCWSSWPAL â€” RCWSSWPPL	FEAT_D128 && FEAT_THE
1	0	1	1	011	RCWSSETP , RCWSSETPA , RCWSSETPL , RCWSSETPAL â€” RCWSSETPL	FEAT_D128 && FEAT_THE
1	1	0	1	001	RCWSCLRP , RCWSCLRPA , RCWSCLRPL , RCWSCLRPAL â€” RCWSCLRPA	FEAT_D128 && FEAT_THE
1	1	0	1	010	RCWSSWPP , RCWSSWPRA , RCWSSWPPL , RCWSSWPAL â€” RCWSSWPRA	FEAT_D128 && FEAT_THE
1	1	0	1	011	RCWSSETP , RCWSSETPA , RCWSSETPL , RCWSSETPAL â€” RCWSSETPA	FEAT_D128 && FEAT_THE
1	1	1	1	001	RCWSCLRP , RCWSCLRPA , RCWSCLRPL , RCWSCLRPAL â€” RCWSCLRPA	FEAT_D128 && FEAT_THE
1	1	1	1	010	RCWSSWPP , RCWSSWPRA , RCWSSWPPL , RCWSSWPAL â€” RCWSSWPAL	FEAT_D128 && FEAT_THE
1	1	1	1	011	RCWSSETP , RCWSSETPA , RCWSSETPL , RCWSSETPAL â€” RCWSSETPA	FEAT_D128 && FEAT_THE

GCS load/store

These instructions are under [Loads and Stores](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	1	1	0	0	1	0	0	0	1	1	1	1	0	opc	1	1	Rn		Rt										

Decode fields opc	Instruction Details	Feature
000	GCSSTR	FEAT_GCS
001	GCSSTTR	FEAT_GCS
01x	UNALLOCATED	-
1xx	UNALLOCATED	-

Load/store memory tags

These instructions are under [Loads and Stores](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	1	1	0	0	1	opc	1						imm9		op2		Rn		Rt										

Decode fields opc	Instruction Details	Feature
imm9	op2	
00	STG " post-index	FEAT_MTE
00	STG " signed offset	FEAT_MTE
00	STG " pre-index	FEAT_MTE
00	STZGM	FEAT_MTE2
01	LDG	FEAT_MTE
01	STZG " post-index	FEAT_MTE
01	STZG " signed offset	FEAT_MTE
01	STZG " pre-index	FEAT_MTE
10	ST2G " post-index	FEAT_MTE
10	ST2G " signed offset	FEAT_MTE
10	ST2G " pre-index	FEAT_MTE
10	!= 000000000	UNALLOCATED
10	000000000	STGM
11	STZ2G " post-index	FEAT_MTE
11	STZ2G " signed offset	FEAT_MTE
11	STZ2G " pre-index	FEAT_MTE
11	!= 000000000	UNALLOCATED
11	000000000	LDGM

Load/store exclusive pair

These instructions are under [Loads and Stores](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	sz	0	0	1	0	0	0	0	L	1		Rs	o0		Rt2			Rn			Rt										

Decode fields			Instruction Details							
sz	L	o0								
0	0	0	STXP	â€”	32-bit					
0	0	1	STLXP	â€”	32-bit					
0	1	0	LDXP	â€”	32-bit					
0	1	1	LDAXP	â€”	32-bit					
1	0	0	STXP	â€”	64-bit					
1	0	1	STLXP	â€”	64-bit					
1	1	0	LDXP	â€”	64-bit					
1	1	1	LDAXP	â€”	64-bit					

Load/store exclusive register

These instructions are under [Loads and Stores](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
size	0	0	1	0	0	0	0	L	0		Rs	o0		Rt2			Rn			Rt											

Decode fields			Instruction Details							
size	L	o0								
00	0	0	STXRB							
00	0	1	STLXRB							
00	1	0	LDXRB							
00	1	1	LDAXRB							
01	0	0	STXRH							
01	0	1	STLXRH							
01	1	0	LDXRH							
01	1	1	LDAXRH							
10	0	0	STXR	â€”	32-bit					
10	0	1	STLXR	â€”	32-bit					
10	1	0	LDXR	â€”	32-bit					
10	1	1	LDAXR	â€”	32-bit					
11	0	0	STXR	â€”	64-bit					
11	0	1	STLXR	â€”	64-bit					
11	1	0	LDXR	â€”	64-bit					
11	1	1	LDAXR	â€”	64-bit					

Load/store ordered

These instructions are under [Loads and Stores](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
size	0	0	1	0	0	0	1	L	0		Rs	o0		Rt2			Rn		Rt												

Decode fields			Instruction Details	Feature
size	L	o0		
00	0	0	STLLRB	FEAT_LOR
00	0	1	STLRB	-
00	1	0	LDLARB	FEAT_LOR
00	1	1	LDARB	-
01	0	0	STLLRH	FEAT_LOR
01	0	1	STLRH	-
01	1	0	LDLARH	FEAT_LOR
01	1	1	LDARH	-
10	0	0	STLLR " 32-bit	FEAT_LOR
10	0	1	STLR " 32-bit	-
10	1	0	LDLAR " 32-bit	FEAT_LOR
10	1	1	LDAR " 32-bit	-
11	0	0	STLLR " 64-bit	FEAT_LOR
11	0	1	STLR " 64-bit	-
11	1	0	LDLAR " 64-bit	FEAT_LOR
11	1	1	LDAR " 64-bit	-

Compare and swap

These instructions are under [Loads and Stores](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
size	0	0	1	0	0	0	1	L	1		Rs	o0		Rt2			Rn		Rt												

Decode fields				Instruction Details	Feature
size	L	o0	Rt2		
			!= 11111	UNALLOCATED	-
00	0	0	11111	CASB , CASAB , CASALB , CASLB " CASB	FEAT_LSE
00	0	1	11111	CASB , CASAB , CASALB , CASLB " CASLB	FEAT_LSE
00	1	0	11111	CASB , CASAB , CASALB , CASLB " CASAB	FEAT_LSE

Decode fields				Instruction Details	Feature
size	L	o0	Rt2		
00	1	1	11111	CASB, CASAB, CASALB, CASLB â€“ CASALB	FEAT_LSE
01	0	0	11111	CASH, CASAH, CASALH, CASLH â€“ CASH	FEAT_LSE
01	0	1	11111	CASH, CASAH, CASALH, CASLH â€“ CASLH	FEAT_LSE
01	1	0	11111	CASH, CASAH, CASALH, CASLH â€“ CASAH	FEAT_LSE
01	1	1	11111	CASH, CASAH, CASALH, CASLH â€“ CASALH	FEAT_LSE
10	0	0	11111	CAS, CASA, CASAL, CASL â€“ 32-bit CAS	FEAT_LSE
10	0	1	11111	CAS, CASA, CASAL, CASL â€“ 32-bit CASL	FEAT_LSE
10	1	0	11111	CAS, CASA, CASAL, CASL â€“ 32-bit CASA	FEAT_LSE
10	1	1	11111	CAS, CASA, CASAL, CASL â€“ 32-bit CASAL	FEAT_LSE
11	0	0	11111	CAS, CASA, CASAL, CASL â€“ 64-bit CAS	FEAT_LSE
11	0	1	11111	CAS, CASA, CASAL, CASL â€“ 64-bit CASL	FEAT_LSE
11	1	0	11111	CAS, CASA, CASAL, CASL â€“ 64-bit CASA	FEAT_LSE
11	1	1	11111	CAS, CASA, CASAL, CASL â€“ 64-bit CASAL	FEAT_LSE

LDIAPP/STILP

These instructions are under [Loads and Stores](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
size	0	1	1	0	0	1	0	L	0	Rt2	opc2	1	0	Rn		Rt																

Decode fields				Instruction Details	Feature
size	L	opc2			
0x				UNALLOCATED	-
1x		001x		UNALLOCATED	-
1x		01xx		UNALLOCATED	-
1x		1xxx		UNALLOCATED	-
10	0	0000		STILP â€“ 32-bit pre-index	FEAT_LRCPC3
10	0	0001		STILP â€“ 32-bit	FEAT_LRCPC3
10	1	0000		LDIAPP â€“ 32-bit post-index	FEAT_LRCPC3

Decode fields			Instruction Details		Feature
size	L	opc2			
10	1	0001	LDIAPP	â€” 32-bit	FEAT_LRCPC3
11	0	0000	STILP	â€” 64-bit pre-index	FEAT_LRCPC3
11	0	0001	STILP	â€” 64-bit	FEAT_LRCPC3
11	1	0000	LDIAPP	â€” 64-bit post-index	FEAT_LRCPC3
11	1	0001	LDIAPP	â€” 64-bit	FEAT_LRCPC3

LDAPR/STLR (writeback)

These instructions are under [Loads and Stores](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
size	0	1	1	0	0	1	1	L	0	0	0	0	0	0	0	0	0	0	0	0	1	0	Rn		Rt							

Decode fields		Instruction Details		Feature
size	L			
0x		UNALLOCATED		-
10	0	STLR	â€” 32-bit	FEAT_LRCPC3
10	1	LDAPR	â€” 32-bit	FEAT_LRCPC3
11	0	STLR	â€” 64-bit	FEAT_LRCPC3
11	1	LDAPR	â€” 64-bit	FEAT_LRCPC3

LDAPR/STLR (unscaled immediate)

These instructions are under [Loads and Stores](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
size	0	1	1	0	0	1	opc	0											imm9	0	0	Rn		Rt							

Decode fields		Instruction Details		Feature
size	opc			
00	00	STLURB		FEAT_LRCPC2
00	01	LDAPURB		FEAT_LRCPC2
00	10	LDAPURSB	â€” 64-bit	FEAT_LRCPC2
00	11	LDAPURSB	â€” 32-bit	FEAT_LRCPC2
01	00	STLURH		FEAT_LRCPC2
01	01	LDAPURH		FEAT_LRCPC2
01	10	LDAPURSH	â€” 64-bit	FEAT_LRCPC2
01	11	LDAPURSH	â€” 32-bit	FEAT_LRCPC2
1x	11	UNALLOCATED		-
10	00	STLUR	â€” 32-bit	FEAT_LRCPC2

Decode fields		Instruction Details		Feature
size	opc			
10	01	LDAPUR â€“ 32-bit		FEAT_LRCPC2
10	10	LDAPURSW		FEAT_LRCPC2
11	00	STLUR â€“ 64-bit		FEAT_LRCPC2
11	01	LDAPUR â€“ 64-bit		FEAT_LRCPC2
11	10	UNALLOCATED		-

LDAPR/STLR (SIMD&FP)

These instructions are under [Loads and Stores](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
size	0	1	1	1	0	1	opc	0																						Rn		Rt

Decode fields		Instruction Details		Feature
size	opc			
00	00	STLUR (SIMD&FP) â€“ 8-bit		FEAT_LRCPC3
00	01	LDAPUR (SIMD&FP) â€“ 8-bit		FEAT_LRCPC3
00	10	STLUR (SIMD&FP) â€“ 128-bit		FEAT_LRCPC3
00	11	LDAPUR (SIMD&FP) â€“ 128-bit		FEAT_LRCPC3
01	00	STLUR (SIMD&FP) â€“ 16-bit		FEAT_LRCPC3
01	01	LDAPUR (SIMD&FP) â€“ 16-bit		FEAT_LRCPC3
01	1x	UNALLOCATED		-
1x	1x	UNALLOCATED		-
10	00	STLUR (SIMD&FP) â€“ 32-bit		FEAT_LRCPC3
10	01	LDAPUR (SIMD&FP) â€“ 32-bit		FEAT_LRCPC3
11	00	STLUR (SIMD&FP) â€“ 64-bit		FEAT_LRCPC3
11	01	LDAPUR (SIMD&FP) â€“ 64-bit		FEAT_LRCPC3

Load register (literal)

These instructions are under [Loads and Stores](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
opc	0	1	1	VR	0	0																							Rt		

Decode fields		Instruction Details	
opc	VR		
00	0	LDR (literal) â€“ 32-bit	
00	1	LDR (literal, SIMD&FP) â€“ 32-bit	
01	0	LDR (literal) â€“ 64-bit	

Decode fields		Instruction Details									
opc	VR										
01	1	LDR (literal, SIMD&FP) â€“ 64-bit									
10	0	LDRSW (literal)									
10	1	LDR (literal, SIMD&FP) â€“ 128-bit									
11	0	PRFM (literal)									
11	1	UNALLOCATED									

Memory Copy and Memory Set

These instructions are under [Loads and Stores](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
size	0	1	1	o0	0	1	op1	0		Rs		op2	0	1		Rn		Rd													

Decode fields			Instruction Details										Feature	
o0	op1	op2												
0	00	0000	CPYFP, CPYFM, CPYFE â€“ CPYFP										FEAT_MOPS	
0	00	0001	CPYFPWT, CPYFMWT, CPYFEWT â€“ CPYFPWT										FEAT_MOPS	
0	00	0010	CPYFPRT, CPYFMRT, CPYFERT â€“ CPYFPRT										FEAT_MOPS	
0	00	0011	CPYFPT, CPYFMT, CPYFET â€“ CPYFPT										FEAT_MOPS	
0	00	0100	CPYFPWN, CPYFMWN, CPYFEWN â€“ CPYFPWN										FEAT_MOPS	
0	00	0101	CPYFPWTWN, CPYFMWTWN, CPYFEWTWN â€“ CPYFPWTWN										FEAT_MOPS	
0	00	0110	CPYFPRTWN, CPYFMRTWN, CPYFERTWN â€“ CPYFPRTWN										FEAT_MOPS	
0	00	0111	CPYFPTWN, CPYFMTWN, CPYFETWN â€“ CPYFPTWN										FEAT_MOPS	
0	00	1000	CPYFPRN, CPYFMRN, CPYFERN â€“ CPYFPRN										FEAT_MOPS	
0	00	1001	CPYFPWTRN, CPYFMWTRN, CPYFEWTRN â€“ CPYFPWTRN										FEAT_MOPS	
0	00	1010	CPYFPRTRN, CPYFMRTRN, CPYFERTRN â€“ CPYFPRTRN										FEAT_MOPS	
0	00	1011	CPYFPTRN, CPYFMTRN, CPYFETRN â€“ CPYFPTRN										FEAT_MOPS	
0	00	1100	CPYFPN, CPYFMN, CPYFEN â€“ CPYFPN										FEAT_MOPS	
0	00	1101	CPYFPWTN, CPYFMWTN, CPYFEWTN â€“ CPYFPWTN										FEAT_MOPS	
0	00	1110	CPYFPRTN, CPYFMRTN, CPYFERTN â€“ CPYFPRTN										FEAT_MOPS	

Decode fields			Instruction Details	Feature
o0	op1	op2		
0	00	1111	CPYFPTN , CPYFMTN , CPYFETN â€” CPYFPTN	FEAT_MOPS
0	01	0000	CPYFP , CPYFM , CPYFE â€” CPYFM	FEAT_MOPS
0	01	0001	CPYFPWT , CPYFMWT , CPYFEWT â€” CPYFMWT	FEAT_MOPS
0	01	0010	CPYFPRT , CPYFMRT , CPYFERT â€” CPYFMRT	FEAT_MOPS
0	01	0011	CPYFPT , CPYFMT , CPYFET â€” CPYFMT	FEAT_MOPS
0	01	0100	CPYFPWN , CPYFMWN , CPYFEWN â€” CPYFMWN	FEAT_MOPS
0	01	0101	CPYFPWTWN , CPYFMWTWN , CPYFEWTWN â€” CPYFMWTWN	FEAT_MOPS
0	01	0110	CPYFPRTWN , CPYFMRTWN , CPYFERTWN â€” CPYFMRTWN	FEAT_MOPS
0	01	0111	CPYFPTWN , CPYFMTWN , CPYFETWN â€” CPYFMTWN	FEAT_MOPS
0	01	1000	CPYFPRN , CPYFMRN , CPYFERN â€” CPYFMRN	FEAT_MOPS
0	01	1001	CPYFPWTRN , CPYFMWTRN , CPYFEWTRN â€” CPYFMWTRN	FEAT_MOPS
0	01	1010	CPYFPRTRN , CPYFMRTRN , CPYFERTRN â€” CPYFMRTRN	FEAT_MOPS
0	01	1011	CPYFPTRN , CPYFMTRN , CPYFETRN â€” CPYFMTRN	FEAT_MOPS
0	01	1100	CPYFPN , CPYFMN , CPYFEN â€” CPYFMN	FEAT_MOPS
0	01	1101	CPYFPWTN , CPYFMWTN , CPYFEWTN â€” CPYFMWTN	FEAT_MOPS
0	01	1110	CPYFPRTN , CPYFMRDN , CPYFERTN â€” CPYFMRDN	FEAT_MOPS
0	01	1111	CPYFPTN , CPYFMTN , CPYFETN â€” CPYFMTN	FEAT_MOPS
0	10	0000	CPYFP , CPYFM , CPYFE â€” CPYFE	FEAT_MOPS
0	10	0001	CPYFPWT , CPYFMWT , CPYFEWT â€” CPYFEWT	FEAT_MOPS
0	10	0010	CPYFPRT , CPYFMRT , CPYFERT â€” CPYFERT	FEAT_MOPS
0	10	0011	CPYFPT , CPYFMT , CPYFET â€” CPYFET	FEAT_MOPS
0	10	0100	CPYFPWN , CPYFMWN , CPYFEWN â€” CPYFEWN	FEAT_MOPS
0	10	0101	CPYFPWTWN , CPYFMWTWN , CPYFEWTWN â€” CPYFEWTWN	FEAT_MOPS

Decode fields			Instruction Details	Feature
o0	op1	op2		
0	10	0110	CPYFPRTWN , CPYFMRTWN , CPYFERTWN â€“ CPYFERTWN	FEAT_MOPS
0	10	0111	CPYFPTWN , CPYFMTWN , CPYFETWN â€“ CPYFETWN	FEAT_MOPS
0	10	1000	CPYFPRN , CPYFMRN , CPYFERN â€“ CPYFERN	FEAT_MOPS
0	10	1001	CPYFPWTRN , CPYFMWTRN , CPYFEWTRN â€“ CPYFEWTRN	FEAT_MOPS
0	10	1010	CPYFPRTRN , CPYFMRTRN , CPYFERTRN â€“ CPYFERTRN	FEAT_MOPS
0	10	1011	CPYFPTRN , CPYFMTRN , CPYFETRN â€“ CPYFETRN	FEAT_MOPS
0	10	1100	CPYFPN , CPYFMN , CPYFEN â€“ CPYFEN	FEAT_MOPS
0	10	1101	CPYFPWTN , CPYFMWTN , CPYFEWTN â€“ CPYFEWTN	FEAT_MOPS
0	10	1110	CPYFPRTN , CPYFMRTN , CPYFERTN â€“ CPYFERTN	FEAT_MOPS
0	10	1111	CPYFPTN , CPYFMTN , CPYFETN â€“ CPYFETN	FEAT_MOPS
0	11	0000	SETP , SETM , SETE â€“ SETP	FEAT_MOPS
0	11	0001	SETPT , SETMT , SETET â€“ SETPT	FEAT_MOPS
0	11	0010	SETPN , SETMN , SETEN â€“ SETPN	FEAT_MOPS
0	11	0011	SETPTN , SETMTN , SETETN â€“ SETPTN	FEAT_MOPS
0	11	0100	SETP , SETM , SETE â€“ SETM	FEAT_MOPS
0	11	0101	SETPT , SETMT , SETET â€“ SETMT	FEAT_MOPS
0	11	0110	SETPN , SETMN , SETEN â€“ SETMN	FEAT_MOPS
0	11	0111	SETPTN , SETMTN , SETETN â€“ SETMTN	FEAT_MOPS
0	11	1000	SETP , SETM , SETE â€“ SETE	FEAT_MOPS
0	11	1001	SETPT , SETMT , SETET â€“ SETET	FEAT_MOPS
0	11	1010	SETPN , SETMN , SETEN â€“ SETEN	FEAT_MOPS
0	11	1011	SETPTN , SETMTN , SETETN â€“ SETETN	FEAT_MOPS
0	11	11xx	UNALLOCATED	-
1	00	0000	CPYP , CPYM , CPYE â€“ CPYP	FEAT_MOPS
1	00	0001	CPYPWT , CPYMWT , CPYEWT â€“ CPYPWT	FEAT_MOPS
1	00	0010	CPYPRT , CPYMRT , CPYERT â€“ CPYPRT	FEAT_MOPS

Decode fields			Instruction Details	Feature
o0	op1	op2		
1	00	0011	CPYPT , CPYMT , CPYET â€” CPYPT	FEAT_MOPS
1	00	0100	CPYPWN , CPYMWN , CPYEWN â€” CPYPWN	FEAT_MOPS
1	00	0101	CPYPWTWN , CPYMWTWN , CPYEWTWN â€” CPYPWTWN	FEAT_MOPS
1	00	0110	CPYPRTWN , CPYMRTWN , CPYERTWN â€” CPYPRTWN	FEAT_MOPS
1	00	0111	CPYPTWN , CPYMTWN , CPYETWN â€” CPYPTWN	FEAT_MOPS
1	00	1000	CPYPRN , CPYMRN , CPYERN â€” CPYPRN	FEAT_MOPS
1	00	1001	CPYPWTRN , CPYMWTRN , CPYEWTRN â€” CPYPWTRN	FEAT_MOPS
1	00	1010	CPYPRTRN , CPYMRTRN , CPYERTRN â€” CPYPRTRN	FEAT_MOPS
1	00	1011	CPYPTRN , CPYMTRN , CPYETRN â€” CPYPTRN	FEAT_MOPS
1	00	1100	CPYPN , CPYMN , CPYEN â€” CPYPN	FEAT_MOPS
1	00	1101	CPYPWTN , CPYMWTN , CPYEWTN â€” CPYPWTN	FEAT_MOPS
1	00	1110	CPYPRTN , CPYMRTRN , CPYERTRN â€” CPYPRTN	FEAT_MOPS
1	00	1111	CPYPTN , CPYMTN , CPYETN â€” CPYPTN	FEAT_MOPS
1	01	0000	CPYP , CPYM , CPYE â€” CPYM	FEAT_MOPS
1	01	0001	CPYPWT , CPYMWT , CPYEWT â€” CPYMWT	FEAT_MOPS
1	01	0010	CPYPRT , CPYMRTRT , CPYERT â€” CPYMRTRT	FEAT_MOPS
1	01	0011	CPYPT , CPYMT , CPYET â€” CPYMT	FEAT_MOPS
1	01	0100	CPYPWN , CPYMWN , CPYEWN â€” CPYMWN	FEAT_MOPS
1	01	0101	CPYPWTWN , CPYMWTWN , CPYEWTWN â€” CPYMWTWN	FEAT_MOPS
1	01	0110	CPYPRTWN , CPYMRTRN , CPYERTWN â€” CPYMRTRN	FEAT_MOPS
1	01	0111	CPYPTWN , CPYMTWN , CPYETWN â€” CPYMTWN	FEAT_MOPS
1	01	1000	CPYPRN , CPYMRN , CPYERN â€” CPYMRN	FEAT_MOPS
1	01	1001	CPYPWTRN , CPYMWTRN , CPYEWTRN â€” CPYMWTRN	FEAT_MOPS
1	01	1010	CPYPRTRN , CPYMRTRN , CPYERTRN â€” CPYMRTRN	FEAT_MOPS

Decode fields			Instruction Details	Feature
o0	op1	op2		
1	01	1011	CPYPTRN , CPYMTRN , CPYETRN â€” CPYMTRN	FEAT_MOPS
1	01	1100	CPYPN , CPYMN , CPYEN â€” CPYMN	FEAT_MOPS
1	01	1101	CPYPWTN , CPYMWTN , CPYEWTN â€” CPYMWTN	FEAT_MOPS
1	01	1110	CPYPRTN , CPYMRTN , CPYERTN â€” CPYMRTN	FEAT_MOPS
1	01	1111	CPYPTN , CPYMTN , CPYETN â€” CPYMTN	FEAT_MOPS
1	10	0000	CPYP , CPYM , CPYE â€” CPYE	FEAT_MOPS
1	10	0001	CPYPWT , CPYMWT , CPYEWT â€” CPYEWT	FEAT_MOPS
1	10	0010	CPYPRT , CPYMRT , CPYERT â€” CPYERT	FEAT_MOPS
1	10	0011	CPYPT , CPYMT , CPYET â€” CPYET	FEAT_MOPS
1	10	0100	CPYPWN , CPYMWN , CPYEWN â€” CPYEWN	FEAT_MOPS
1	10	0101	CPYPWTWN , CPYMWTWN , CPYEWTWN â€” CPYEWTWN	FEAT_MOPS
1	10	0110	CPYPRTWN , CPYMRTWN , CPYERTWN â€” CPYERTWN	FEAT_MOPS
1	10	0111	CPYPTWN , CPYMTWN , CPYETWN â€” CPYETWN	FEAT_MOPS
1	10	1000	CPYPRN , CPYMRN , CPYERN â€” CPYERN	FEAT_MOPS
1	10	1001	CPYPWTRN , CPYMWTRN , CPYEWTRN â€” CPYEWTRN	FEAT_MOPS
1	10	1010	CPYPRTRN , CPYMRTRN , CPYERTRN â€” CPYERTRN	FEAT_MOPS
1	10	1011	CPYPTRN , CPYMTRN , CPYETRN â€” CPYETRN	FEAT_MOPS
1	10	1100	CPYPN , CPYMN , CPYEN â€” CPYEN	FEAT_MOPS
1	10	1101	CPYPWTN , CPYMWTN , CPYEWTN â€” CPYEWTN	FEAT_MOPS
1	10	1110	CPYPRTN , CPYMRTN , CPYERTN â€” CPYERTN	FEAT_MOPS
1	10	1111	CPYPTN , CPYMTN , CPYETN â€” CPYETN	FEAT_MOPS
1	11	0000	SETGP , SETGM , SETGE â€” SETGP	FEAT_MOPS
1	11	0001	SETGPT , SETGMT , SETGET â€” SETGPT	FEAT_MOPS
1	11	0010	SETGPN , SETGMN , SETGEN â€” SETGPN	FEAT_MOPS

Decode fields			Instruction Details	Feature
o0	op1	op2		
1	11	0011	SETGPTN, SETGMTN, SETGETN â€” SETGPTN	FEAT_MOPS
1	11	0100	SETGP, SETGM, SETGE â€” SETGM	FEAT_MOPS
1	11	0101	SETGPT, SETGMT, SETGET â€” SETGMT	FEAT_MOPS
1	11	0110	SETGPN, SETGMN, SETGEN â€” SETGMN	FEAT_MOPS
1	11	0111	SETGPTN, SETGMTN, SETGETN â€” SETGMTN	FEAT_MOPS
1	11	1000	SETGP, SETGM, SETGE â€” SETGE	FEAT_MOPS
1	11	1001	SETGPT, SETGMT, SETGET â€” SETGET	FEAT_MOPS
1	11	1010	SETGPN, SETGMN, SETGEN â€” SETGEN	FEAT_MOPS
1	11	1011	SETGPTN, SETGMTN, SETGETN â€” SETGETN	FEAT_MOPS
1	11	11xx	UNALLOCATED	-

Load/store no-allocate pair (offset)

These instructions are under [Loads and Stores](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
opc	1	0	1	VR	0	0	0	L																			Rn		Rt		

Decode fields			Instruction Details
opc	VR	L	
00	0	0	STNP â€” 32-bit
00	0	1	LDNP â€” 32-bit
00	1	0	STNP (SIMD&FP) â€” 32-bit
00	1	1	LDNP (SIMD&FP) â€” 32-bit
01	0		UNALLOCATED
01	1	0	STNP (SIMD&FP) â€” 64-bit
01	1	1	LDNP (SIMD&FP) â€” 64-bit
10	0	0	STNP â€” 64-bit
10	0	1	LDNP â€” 64-bit
10	1	0	STNP (SIMD&FP) â€” 128-bit
10	1	1	LDNP (SIMD&FP) â€” 128-bit
11			UNALLOCATED

Load/store register pair (post-indexed)

These instructions are under [Loads and Stores](#).

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
opc	1	0	1	VR	0	0	1	L															Rt2		Rn		Rt					

Decode fields			Instruction Details			Feature		
opc	VR	L						
00	0	0	STP â€“ 32-bit				-	
00	0	1	LDP â€“ 32-bit				-	
00	1	0	STP (SIMD&FP) â€“ 32-bit				-	
00	1	1	LDP (SIMD&FP) â€“ 32-bit				-	
01	0	0	STGP				FEAT_MTE	
01	0	1	LDPSW				-	
01	1	0	STP (SIMD&FP) â€“ 64-bit				-	
01	1	1	LDP (SIMD&FP) â€“ 64-bit				-	
10	0	0	STP â€“ 64-bit				-	
10	0	1	LDP â€“ 64-bit				-	
10	1	0	STP (SIMD&FP) â€“ 128-bit				-	
10	1	1	LDP (SIMD&FP) â€“ 128-bit				-	
11			UNALLOCATED				-	

Load/store register pair (offset)

These instructions are under [Loads and Stores](#).

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
opc	1	0	1	VR	0	1	0	L														Rt2		Rn		Rt						

Decode fields			Instruction Details			Feature		
opc	VR	L						
00	0	0	STP â€“ 32-bit				-	
00	0	1	LDP â€“ 32-bit				-	
00	1	0	STP (SIMD&FP) â€“ 32-bit				-	
00	1	1	LDP (SIMD&FP) â€“ 32-bit				-	
01	0	0	STGP				FEAT_MTE	
01	0	1	LDPSW				-	
01	1	0	STP (SIMD&FP) â€“ 64-bit				-	
01	1	1	LDP (SIMD&FP) â€“ 64-bit				-	
10	0	0	STP â€“ 64-bit				-	
10	0	1	LDP â€“ 64-bit				-	

Decode fields			Instruction Details			Feature	
opc	VR	L					
10	1	0	STP (SIMD&FP)	â€”	128-bit	-	
10	1	1	LDP (SIMD&FP)	â€”	128-bit	-	
11			UNALLOCATED			-	

Load/store register pair (pre-indexed)

These instructions are under [Loads and Stores](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
opc	1	0	1	VR	0	1	1	L																						Rt	

Decode fields			Instruction Details			Feature	
opc	VR	L					
00	0	0	STP	â€”	32-bit	-	
00	0	1	LDP	â€”	32-bit	-	
00	1	0	STP (SIMD&FP)	â€”	32-bit	-	
00	1	1	LDP (SIMD&FP)	â€”	32-bit	-	
01	0	0	STGP			FEAT_MTE	
01	0	1	LDPSW			-	
01	1	0	STP (SIMD&FP)	â€”	64-bit	-	
01	1	1	LDP (SIMD&FP)	â€”	64-bit	-	
10	0	0	STP	â€”	64-bit	-	
10	0	1	LDP	â€”	64-bit	-	
10	1	0	STP (SIMD&FP)	â€”	128-bit	-	
10	1	1	LDP (SIMD&FP)	â€”	128-bit	-	
11			UNALLOCATED			-	

Load/store register (unscaled immediate)

These instructions are under [Loads and Stores](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
size	1	1	1	VR	0	0	opc	0																			Rn		Rt		

Decode fields			Instruction Details		
size	VR	opc			
x1	1	1x	UNALLOCATED		
00	0	00	STURB		
00	0	01	LDURB		
00	0	10	LDURSB â€” 64-bit		

Decode fields			Instruction Details	
size	VR	opc		
00	0	11	LDURSB â€” 32-bit	
00	1	00	STUR (SIMD&FP) â€” 8-bit	
00	1	01	LDUR (SIMD&FP) â€” 8-bit	
00	1	10	STUR (SIMD&FP) â€” 128-bit	
00	1	11	LDUR (SIMD&FP) â€” 128-bit	
01	0	00	STURH	
01	0	01	LDURH	
01	0	10	LDURSH â€” 64-bit	
01	0	11	LDURSH â€” 32-bit	
01	1	00	STUR (SIMD&FP) â€” 16-bit	
01	1	01	LDUR (SIMD&FP) â€” 16-bit	
1x	0	11	UNALLOCATED	
1x	1	1x	UNALLOCATED	
10	0	00	STUR â€” 32-bit	
10	0	01	LDUR â€” 32-bit	
10	0	10	LDURSW	
10	1	00	STUR (SIMD&FP) â€” 32-bit	
10	1	01	LDUR (SIMD&FP) â€” 32-bit	
11	0	00	STUR â€” 64-bit	
11	0	01	LDUR â€” 64-bit	
11	0	10	PRFUM	
11	1	00	STUR (SIMD&FP) â€” 64-bit	
11	1	01	LDUR (SIMD&FP) â€” 64-bit	

Load/store register (immediate post-indexed)

These instructions are under [Loads and Stores](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
size	1	1	1	VR	0	0	opc	0															imm9	0	1	Rn		Rt				

Decode fields			Instruction Details	
size	VR	opc		
x1	1	1x	UNALLOCATED	
00	0	00	STRB (immediate)	
00	0	01	LDRB (immediate)	
00	0	10	LDRSB (immediate) â€” 64-bit	
00	0	11	LDRSB (immediate) â€” 32-bit	
00	1	00	STR (immediate, SIMD&FP) â€” 8-bit	

Decode fields			Instruction Details									
size	VR	opc										
00	1	01	LDR (immediate, SIMD&FP)	â€”	8-bit							
00	1	10	STR (immediate, SIMD&FP)	â€”	128-bit							
00	1	11	LDR (immediate, SIMD&FP)	â€”	128-bit							
01	0	00	STRH (immediate)									
01	0	01	LDRH (immediate)									
01	0	10	LDRSH (immediate)	â€”	64-bit							
01	0	11	LDRSH (immediate)	â€”	32-bit							
01	1	00	STR (immediate, SIMD&FP)	â€”	16-bit							
01	1	01	LDR (immediate, SIMD&FP)	â€”	16-bit							
1x	0	11	UNALLOCATED									
1x	1	1x	UNALLOCATED									
10	0	00	STR (immediate)	â€”	32-bit							
10	0	01	LDR (immediate)	â€”	32-bit							
10	0	10	LDRSW (immediate)									
10	1	00	STR (immediate, SIMD&FP)	â€”	32-bit							
10	1	01	LDR (immediate, SIMD&FP)	â€”	32-bit							
11	0	00	STR (immediate)	â€”	64-bit							
11	0	01	LDR (immediate)	â€”	64-bit							
11	0	10	UNALLOCATED									
11	1	00	STR (immediate, SIMD&FP)	â€”	64-bit							
11	1	01	LDR (immediate, SIMD&FP)	â€”	64-bit							

Load/store register (unprivileged)

These instructions are under [Loads and Stores](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
size	1	1	1	VR	0	0	opc	0															imm9	1	0	Rn		Rt			

Decode fields			Instruction Details									
size	VR	opc										
		1										
			UNALLOCATED									
00	0	00	STTRB									
00	0	01	LDTRB									
00	0	10	LDTRSB	â€”	64-bit							
00	0	11	LDTRSB	â€”	32-bit							
01	0	00	STTRH									
01	0	01	LDTRH									
01	0	10	LDTRSH	â€”	64-bit							

Decode fields			Instruction Details
size	VR	opc	
01	0	11	LDTRSH â€“ 32-bit
1x	0	11	UNALLOCATED
10	0	00	STTR â€“ 32-bit
10	0	01	LDTR â€“ 32-bit
10	0	10	LDTRSW
11	0	00	STTR â€“ 64-bit
11	0	01	LDTR â€“ 64-bit
11	0	10	UNALLOCATED

Load/store register (immediate pre-indexed)

These instructions are under [Loads and Stores](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
size	1	1	1	VR	0	0	opc	0							imm9		1	1		Rn		Rt									

Decode fields			Instruction Details
size	VR	opc	
x1	1	1x	UNALLOCATED
00	0	00	STRB (immediate)
00	0	01	LDRB (immediate)
00	0	10	LDRSB (immediate) â€“ 64-bit
00	0	11	LDRSB (immediate) â€“ 32-bit
00	1	00	STR (immediate, SIMD&FP) â€“ 8-bit
00	1	01	LDR (immediate, SIMD&FP) â€“ 8-bit
00	1	10	STR (immediate, SIMD&FP) â€“ 128-bit
00	1	11	LDR (immediate, SIMD&FP) â€“ 128-bit
01	0	00	STRH (immediate)
01	0	01	LDRH (immediate)
01	0	10	LDRSH (immediate) â€“ 64-bit
01	0	11	LDRSH (immediate) â€“ 32-bit
01	1	00	STR (immediate, SIMD&FP) â€“ 16-bit
01	1	01	LDR (immediate, SIMD&FP) â€“ 16-bit
1x	0	11	UNALLOCATED
1x	1	1x	UNALLOCATED
10	0	00	STR (immediate) â€“ 32-bit
10	0	01	LDR (immediate) â€“ 32-bit
10	0	10	LDRSW (immediate)
10	1	00	STR (immediate, SIMD&FP) â€“ 32-bit

Decode fields			Instruction Details
size	VR	opc	
10	1	01	LDR (immediate, SIMD&FP) â€“ 32-bit
11	0	00	STR (immediate) â€“ 64-bit
11	0	01	LDR (immediate) â€“ 64-bit
11	0	10	UNALLOCATED
11	1	00	STR (immediate, SIMD&FP) â€“ 64-bit
11	1	01	LDR (immediate, SIMD&FP) â€“ 64-bit

Atomic memory operations

These instructions are under [Loads and Stores](#).

size	1	1	1	VR	0	0	A	R	1	Rs	o3	opc	0	0	Rn		Rt
------	---	---	---	----	---	---	---	---	---	----	----	-----	---	---	----	--	----

size	VR	Decode fields			o3	opc	Instruction Details		Feature
		A	R	Rs					
	0				1	11x	UNALLOCATED	-	
	0	0			1	100	UNALLOCATED	-	
	0	0	1		1	101	UNALLOCATED	-	
	0	1	0		1	101	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 â€” LDADDB	FEAT_LSE	
00	0	0	0		0	001	LDCLRB , LDCLRAB , LDCLRALB , LDCLRLB â€” LDCLRB	FEAT_LSE	
00	0	0	0		0	010	LDEORB , LDEORAB , LDEORALB , LDEORLB â€” LDEORB	FEAT_LSE	
00	0	0	0		0	011	LDSETB , LDSETAB , LDSETALB , LDSETLB â€” LDSETB	FEAT_LSE	

size	VR	Decode fields				o3	opc	Instruction Details	Feature
		A	R	Rs					
00	0	0	0			0	100	LDSMAXB , LDSMAXAB , LDSMAXALB , LDSMAXLB â€” LDSMAXB	FEAT_LSE
00	0	0	0			0	101	LDSMINB , LDSMINAB , LDSMINALB , LDSMINLB â€” LDSMINB	FEAT_LSE
00	0	0	0			0	110	LDUMAXB , LDUMAXAB , LDUMAXALB , LDUMAXLB â€” LDUMAXB	FEAT_LSE
00	0	0	0			0	111	LDUMINB , LDUMINAB , LDUMINALB , LDUMINLB â€” LDUMINB	FEAT_LSE
00	0	0	0			1	000	SWPB , SWPAB , SWPALB , SWPLB â€” SWPB	FEAT_LSE
00	0	0	0			1	001	RCWCLR , RCWCLRA , RCWCLRL , RCWCLRAL â€” RCWCLR	FEAT_THE
00	0	0	0			1	010	RCWSWP , RCWSWPA , RCWSWPL , RCWSWPAL â€” RCWSWP	FEAT_THE
00	0	0	0			1	011	RCWSET , RCWSETA , RCWSETL , RCWSETAL â€” RCWSET	FEAT_THE
00	0	0	0			1	101	UNALLOCATED	-
00	0	0	1			0	000	LDADDB , LDADDAB , LDADDALB , LDADDLB â€” LDADDLB	FEAT_LSE

size	VR	Decode fields				o3	opc	Instruction Details	Feature
		A	R	Rs					
00	0	0	1			0	001	LDCLRB , LDCLRAB , LDCLRALB , LDCLRLB â€” LDCLRLB	FEAT_LSE
00	0	0	1			0	010	LDEORB , LDEORAB , LDEORALB , LDEORLB â€” LDEORLB	FEAT_LSE
00	0	0	1			0	011	LDSETB , LDSETAB , LDSETALB , LDSETLB â€” LDSETLB	FEAT_LSE
00	0	0	1			0	100	LDSMAXB , LDSMAXAB , LDSMAXALB , LDSMAXLB â€” LDSMAXLB	FEAT_LSE
00	0	0	1			0	101	LDSMINB , LDSMINAB , LDSMINALB , LDSMINLB â€” LDSMINLB	FEAT_LSE
00	0	0	1			0	110	LDUMAXB , LDUMAXAB , LDUMAXALB , LDUMAXLB â€” LDUMAXLB	FEAT_LSE
00	0	0	1			0	111	LDUMINB , LDUMINAB , LDUMINALB , LDUMINLB â€” LDUMINLB	FEAT_LSE
00	0	0	1			1	000	SWPB , SWPAB , SWPALB , SWPLB â€” SWPLB	FEAT_LSE
00	0	0	1			1	001	RCWCLR , RCWCLRA , RCWCLRL , RCWCLRAL â€” RCWCLRL	FEAT_THE

size	VR	Decode fields				o3	opc	Instruction Details	Feature
		A	R	Rs					
00	0	0	1			1	010	RCWSWP , RCWSWPA , RCWSWPL , RCWSWPAL â€” RCWSWPL	FEAT_THE
00	0	0	1			1	011	RCWSET , RCWSETA , RCWSETL , RCWSETAL â€” RCWSETL	FEAT_THE
00	0	1	0			0	000	LDADDB , LDADDAB , LDADDALB , LDADDLB â€” LDADDAB	FEAT_LSE
00	0	1	0			0	001	LDCLRB , LDCLRAB , LDCLRALB , LDCLRLB â€” LDCLRAB	FEAT_LSE
00	0	1	0			0	010	LDEORB , LDEORAB , LDEORALB , LDEORLB â€” LDEORAB	FEAT_LSE
00	0	1	0			0	011	LDSETB , LDSETAB , LDSETALB , LDSETLB â€” LDSETAB	FEAT_LSE
00	0	1	0			0	100	LDSMAXB , LDSMAXAB , LDSMAXALB , LDSMAXLB â€” LDSMAXAB	FEAT_LSE
00	0	1	0			0	101	LDSMINB , LDSMINAB , LDSMINALB , LDSMINLB â€” LDSMINAB	FEAT_LSE
00	0	1	0			0	110	LDUMAXB , LDUMAXAB , LDUMAXALB , LDUMAXLB â€” LDUMAXAB	FEAT_LSE

size	VR	Decode fields				o3	opc	Instruction Details	Feature
		A	R	Rs					
00	0	1	0			0	111	LDUMINB , LDUMINAB , LDUMINALB , LDUMINLB â€” LDUMINAB	FEAT_LSE
00	0	1	0			1	000	SWPB , SWPAB , SWPALB , SWPLB â€” SWPAB	FEAT_LSE
00	0	1	0			1	001	RCWCLR , RCWCLRA , RCWCLRL , RCWCLRAL â€” RCWCLRA	FEAT_THE
00	0	1	0			1	010	RCWSWP , RCWSWPA , RCWSWPL , RCWSWPAL â€” RCWSWPA	FEAT_THE
00	0	1	0			1	011	RCWSET , RCWSETA , RCWSETL , RCWSETAL â€” RCWSETA	FEAT_THE
00	0	1	0			1	100	LDAPRB	FEAT_LRPCPC
00	0	1	1			0	000	LDADDB , LDADDAB , LDADDALB , LDADDLB â€” LDADDALB	FEAT_LSE
00	0	1	1			0	001	LDCLRB , LDCLRAB , LDCLRALB , LDCLRLB â€” LDCLRALB	FEAT_LSE
00	0	1	1			0	010	LDEORB , LDEORAB , LDEORALB , LDEORLB â€” LDEORALB	FEAT_LSE
00	0	1	1			0	011	LDSETB , LDSETAB , LDSETALB , LDSETLB â€” LDSETALB	FEAT_LSE

size	VR	Decode fields				o3	opc	Instruction Details	Feature
		A	R	Rs					
00	0	1	1			0	100	LDSMAXB , LDSMAXAB , LDSMAXALB , LDSMAXLB â€” LDSMAXALB	FEAT_LSE
00	0	1	1			0	101	LDSMINB , LDSMINAB , LDSMINALB , LDSMINLB â€” LDSMINALB	FEAT_LSE
00	0	1	1			0	110	LDUMAXB , LDUMAXAB , LDUMAXALB , LDUMAXLB â€” LDUMAXALB	FEAT_LSE
00	0	1	1			0	111	LDUMINB , LDUMINAB , LDUMINALB , LDUMINLB â€” LDUMINALB	FEAT_LSE
00	0	1	1			1	000	SWPB , SWPAB , SWPALB , SWPLB â€” SWPALB	FEAT_LSE
00	0	1	1			1	001	RCWCLR , RCWCLRA , RCWCLRL , RCWCLRAL â€” RCWCLRAL	FEAT_THE
00	0	1	1			1	010	RCWSWP , RCWSWPA , RCWSWPL , RCWSWPAL â€” RCWSWPAL	FEAT_THE
00	0	1	1			1	011	RCWSET , RCWSETA , RCWSETL , RCWSETAL â€” RCWSETAL	FEAT_THE
01	0	0	0			0	000	LDADDH , LDADDAH , LDADDALH , LDADDLH â€” LDADDH	FEAT_LSE

size	VR	Decode fields				o3	opc	Instruction Details	Feature
		A	R	Rs					
01	0	0	0			0	001	LDCLRH , LDCLRAH , LDCLRALH , LDCLRLH â€” LDCLRH	FEAT_LSE
01	0	0	0			0	010	LDEORH , LDEORAH , LDEORALH , LDEORLH â€” LDEORH	FEAT_LSE
01	0	0	0			0	011	LDSETH , LDSETAH , LDSETALH , LDSETLH â€” LDSETH	FEAT_LSE
01	0	0	0			0	100	LDSMAXH , LDSMAXAH , LDSMAXALH , LDSMAXLH â€” LDSMAXH	FEAT_LSE
01	0	0	0			0	101	LDSMINH , LDSMINAH , LDSMINALH , LDSMINLH â€” LDSMINH	FEAT_LSE
01	0	0	0			0	110	LDUMAXH , LDUMAXAH , LDUMAXALH , LDUMAXLH â€” LDUMAXH	FEAT_LSE
01	0	0	0			0	111	LDUMINH , LDUMINAH , LDUMINALH , LDUMINLH â€” LDUMINH	FEAT_LSE
01	0	0	0			1	000	SWPH , SWPAH , SWPALH , SWPLH â€” SWPH	FEAT_LSE
01	0	0	0			1	001	RCWSCLR , RCWSCLRA , RCWSCLRL , RCWSCLRAL â€” RCWSCLR	FEAT_THE

size	VR	Decode fields				o3	opc	Instruction Details	Feature
		A	R	Rs					
01	0	0	0			1	010	RCWSSWP , RCWSSWPA , RCWSSWPL , RCWSSWPAL â€” RCWSSWP	FEAT_THE
01	0	0	0			1	011	RCWSSET , RCWSSETA , RCWSSETL , RCWSSETAL â€” RCWSSET	FEAT_THE
01	0	0	0			1	101	UNALLOCATED	-
01	0	0	1			0	000	LDADDH , LDADDAH , LDADDALH , LDADDLH â€” LDADDLH	FEAT_LSE
01	0	0	1			0	001	LDCLRH , LDCLRAH , LDCLRALH , LDCLRLH â€” LDCLRLH	FEAT_LSE
01	0	0	1			0	010	LDEORH , LDEORAH , LDEORALH , LDEORLH â€” LDEORLH	FEAT_LSE
01	0	0	1			0	011	LDSETH , LDSETAH , LDSETALH , LDSETLH â€” LDSETLH	FEAT_LSE
01	0	0	1			0	100	LDSMAXH , LDSMAXAH , LDSMAXALH , LDSMAXLH â€” LDSMAXLH	FEAT_LSE
01	0	0	1			0	101	LDSMINH , LDSMINAH , LDSMINALH , LDSMINLH â€” LDSMINLH	FEAT_LSE
01	0	0	1			0	110	LDUMAXH , LDUMAXAH , LDUMAXALH , LDUMAXLH â€” LDUMAXLH	FEAT_LSE

size	VR	Decode fields				o3	opc	Instruction Details	Feature
		A	R	Rs					
01	0	0	1			0	111	LDUMINH , LDUMINAH , LDUMINALH , LDUMINLH â€” LDUMINLH	FEAT_LSE
01	0	0	1			1	000	SWPH , SWPAH , SWPALH , SWPLH â€” SWPLH	FEAT_LSE
01	0	0	1			1	001	RCWSCLR , RCWSCLRA , RCWSCLRL , RCWSCLRAL â€” RCWSCLRL	FEAT_THE
01	0	0	1			1	010	RCWSSWP , RCWSSWPA , RCWSSWPL , RCWSSWPAL â€” RCWSSWPL	FEAT_THE
01	0	0	1			1	011	RCWSSET , RCWSSETA , RCWSSETL , RCWSSETAL â€” RCWSSETL	FEAT_THE
01	0	1	0			0	000	LDADDH , LDADDAH , LDADDALH , LDADDLH â€” LDADDAH	FEAT_LSE
01	0	1	0			0	001	LDCLRH , LDCLRAH , LDCLRALH , LDCLRLH â€” LDCLRAH	FEAT_LSE
01	0	1	0			0	010	LDEORH , LDEORAH , LDEORALH , LDEORLH â€” LDEORAH	FEAT_LSE
01	0	1	0			0	011	LDSETH , LDSETAH , LDSETALH , LDSETLH â€” LDSETAH	FEAT_LSE

size	VR	Decode fields				o3	opc	Instruction Details	Feature
		A	R	Rs					
01	0	1	0			0	100	LDSMAXH , LDSMAXAH , LDSMAXALH , LDSMAXLH â€” LDSMAXAH	FEAT_LSE
01	0	1	0			0	101	LDSMINH , LDSMINAH , LDSMINALH , LDSMINLH â€” LDSMINAH	FEAT_LSE
01	0	1	0			0	110	LDUMAXH , LDUMAXAH , LDUMAXALH , LDUMAXLH â€” LDUMAXAH	FEAT_LSE
01	0	1	0			0	111	LDUMINH , LDUMINAH , LDUMINALH , LDUMINLH â€” LDUMINAH	FEAT_LSE
01	0	1	0			1	000	SWPH , SWPAH , SWPALH , SWPLH â€” SWPAH	FEAT_LSE
01	0	1	0			1	001	RCWSCLR , RCWSCLRA , RCWSCLRL , RCWSCLRAL â€” RCWSCLRA	FEAT_THE
01	0	1	0			1	010	RCWSSWP , RCWSSWPA , RCWSSWPL , RCWSSWPAL â€” RCWSSWPA	FEAT_THE
01	0	1	0			1	011	RCWSSET , RCWSSETA , RCWSSETL , RCWSSETAL â€” RCWSSETA	FEAT_THE
01	0	1	0			1	100	LDAPRH	FEAT_LRPCP
01	0	1	1			0	000	LDADDH , LDADDAH , LDADDALH , LDADDLH â€” LDADDALH	FEAT_LSE

size	VR	Decode fields				o3	opc	Instruction Details	Feature
		A	R	Rs					
01	0	1	1			0	001	LDCLRH , LDCLRAH , LDCLRALH , LDCLRLH â€” LDCLRALH	FEAT_LSE
01	0	1	1			0	010	LDEORH , LDEORAH , LDEORALH , LDEORLH â€” LDEORALH	FEAT_LSE
01	0	1	1			0	011	LDSETH , LDSETAH , LDSETALH , LDSETLH â€” LDSETALH	FEAT_LSE
01	0	1	1			0	100	LDSMAXH , LDSMAXAH , LDSMAXALH , LDSMAXLH â€” LDSMAXALH	FEAT_LSE
01	0	1	1			0	101	LDSMINH , LDSMINAH , LDSMINALH , LDSMINLH â€” LDSMINALH	FEAT_LSE
01	0	1	1			0	110	LDUMAXH , LDUMAXAH , LDUMAXALH , LDUMAXLH â€” LDUMAXALH	FEAT_LSE
01	0	1	1			0	111	LDUMINH , LDUMINAH , LDUMINALH , LDUMINLH â€” LDUMINALH	FEAT_LSE
01	0	1	1			1	000	SWPH , SWPAH , SWPALH , SWPLH â€” SWPALH	FEAT_LSE
01	0	1	1			1	001	RCWSCLR , RCWSCLRA , RCWSCLRL , RCWSCLRAL â€” RCWSCLRAL	FEAT_THE

size	VR	Decode fields				o3	opc	Instruction Details	Feature
		A	R	Rs					
01	0	1	1			1	010	RCWSSWP , RCWSSWPA , RCWSSWPL , RCWSSWPAL â€” RCWSSWPAL	FEAT_THE
01	0	1	1			1	011	RCWSSET , RCWSSETA , RCWSSETL , RCWSSETAL â€” RCWSSETAL	FEAT_THE
1x	0	0	1			1	001	UNALLOCATED	-
1x	0	0	1			1	010	UNALLOCATED	-
1x	0	0	1			1	011	UNALLOCATED	-
1x	0	1	0			1	001	UNALLOCATED	-
1x	0	1	0			1	010	UNALLOCATED	-
1x	0	1	0			1	011	UNALLOCATED	-
1x	0	1	1			1	001	UNALLOCATED	-
1x	0	1	1			1	010	UNALLOCATED	-
1x	0	1	1			1	011	UNALLOCATED	-
10	0	0	0			0	000	LDADD , LDADDA , LDADDAL , LDADDL â€” 32-bit LDADD	FEAT_LSE
10	0	0	0			0	001	LDCLR , LDCLRA , LDCLRAL , LDCLRL â€” 32-bit LDCLR	FEAT_LSE
10	0	0	0			0	010	LDEOR , LDEORA , LDEORAL , LDEORL â€” 32-bit LDEOR	FEAT_LSE
10	0	0	0			0	011	LDSET , LDSETA , LDSETAL , LDSETL â€” 32-bit LDSET	FEAT_LSE
10	0	0	0			0	100	LDSMAX , LDSMAXA , LDSMAXAL , LDSMAXL â€” 32-bit LDSMAX	FEAT_LSE

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

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

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

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

size	VR	Decode fields				o3	opc	Instruction Details	Feature
		A	R	Rs					
11	0	0	0			0	110	LDUMAX , LDUMAXA , LDUMAXAL , LDUMAXL â€“ 64-bit LDUMAX	FEAT_LSE
11	0	0	0			0	111	LDUMIN , LDUMINA , LDUMINAL , LDUMINL â€“ 64-bit LDUMIN	FEAT_LSE
11	0	0	0			1	000	SWP , SWPA , SWPAL , SWPL â€“ 64-bit SWP	FEAT_LSE
11	0	0	0			1	010	ST64BV0	FEAT_LS64_ACCDA
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 â€“ 64-bit LDADDL	FEAT_LSE
11	0	0	1			0	001	LDCLR , LDCLRA , LDCLRAL , LDCLRL â€“ 64-bit LDCLRL	FEAT_LSE
11	0	0	1			0	010	LDEOR , LDEORA , LDEORAL , LDEORL â€“ 64-bit LDEORL	FEAT_LSE
11	0	0	1			0	011	LDSET , LDSETA , LDSETAL , LDSETL â€“ 64-bit LDSETL	FEAT_LSE
11	0	0	1			0	100	LDSMAX , LDSMAXA , LDSMAXAL , LDSMAXL â€“ 64-bit LDSMAXL	FEAT_LSE
11	0	0	1			0	101	LDSMIN , LDSMINA , LDSMINAL , LDSMINL â€“ 64-bit LDSMINL	FEAT_LSE

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

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

size	VR	Decode fields					Instruction Details		Feature
		A	R	Rs	o3	opc			
11	0	1	1		0	111	LDUMIN , LDUMINA , LDUMINAL , LDUMINL â€” 64-bit LDUMINAL	FEAT_LSE	
11	0	1	1		1	000	SWP , SWPA , SWPAL , SWPL â€” 64-bit SWPAL	FEAT_LSE	

Load/store register (register offset)

These instructions are under [Loads and Stores](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
size	1	1	1	VR	0	0	opc	1	Rm		option	S	1	0		Rn		Rt													

size	VR	Decode fields			Instruction Details		Feature
		opc	option	Rt			
x1	1	1x			UNALLOCATED	-	
00	0	00	!= 011		STRB (register) â€” extended register	-	
00	0	00	011		STRB (register) â€” shifted register	-	
00	0	01	!= 011		LDRB (register) â€” extended register	-	
00	0	01	011		LDRB (register) â€” shifted register	-	
00	0	10	!= 011		LDRSB (register) â€” 64-bit with extended register offset	-	
00	0	10	011		LDRSB (register) â€” 64-bit with shifted register offset	-	
00	0	11	!= 011		LDRSB (register) â€” 32-bit with extended register offset	-	

size	VR	Decode fields			Rt	Instruction Details	Feature
		opc	option				
00	0	11	011			LDRSB (register) â€” 32-bit with shifted register offset	-
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) â€” 64-bit	-
01	0	11				LDRSH (register) â€” 32-bit	-
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) â€” 32-bit	-
10	0	01				LDR (register) â€” 32-bit	-
10	0	10				LDRSW (register)	-
10	1	00				STR (register, SIMD&FP)	-
10	1	01				LDR (register, SIMD&FP)	-
11	0	00				STR (register) â€” 64-bit	-
11	0	01				LDR (register) â€” 64-bit	-
11	0	10	x0x			UNALLOCATED	-
11	0	10	x1x	!= 11xxx		PRFM (register)	-
11	0	10	x1x	11xxx		RPRFM	FEAT_RPRFM

size	VR	opc	Decode fields		Rt	Instruction Details	Feature
			option				
11	1	00				STR (register, SIMD&FP)	-
11	1	01				LDR (register, SIMD&FP)	-

Load/store register (pac)

These instructions are under [Loads and Stores](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
size	1	1	1	VR	0	0	M	S	1																					Rt	

size	VR	Decode fields		W	Instruction Details	Feature
		M				
!= 11					UNALLOCATED	-
11	0	0	0		LDRAA, LDRAB "key A, offset	FEAT_PAuth
11	0	0	1		LDRAA, LDRAB "key A, pre-indexed	FEAT_PAuth
11	0	1	0		LDRAA, LDRAB "key B, offset	FEAT_PAuth
11	0	1	1		LDRAA, LDRAB "key B, pre-indexed	FEAT_PAuth
11	1				UNALLOCATED	-

Load/store register (unsigned immediate)

These instructions are under [Loads and Stores](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
size	1	1	1	VR	0	1	opc																					Rn		Rt	

size	VR	Decode fields		Instruction Details
		opc		
x1	1	1x		UNALLOCATED
00	0	00		STRB (immediate)
00	0	01		LDRB (immediate)
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 Details
size	VR	opc	
00	1	11	LDR (immediate, SIMD&FP) â€“ 128-bit
01	0	00	STRH (immediate)
01	0	01	LDRH (immediate)
01	0	10	LDRSH (immediate) â€“ 64-bit
01	0	11	LDRSH (immediate) â€“ 32-bit
01	1	00	STR (immediate, SIMD&FP) â€“ 16-bit
01	1	01	LDR (immediate, SIMD&FP) â€“ 16-bit
1x	0	11	UNALLOCATED
1x	1	1x	UNALLOCATED
10	0	00	STR (immediate) â€“ 32-bit
10	0	01	LDR (immediate) â€“ 32-bit
10	0	10	LDRSW (immediate)
10	1	00	STR (immediate, SIMD&FP) â€“ 32-bit
10	1	01	LDR (immediate, SIMD&FP) â€“ 32-bit
11	0	00	STR (immediate) â€“ 64-bit
11	0	01	LDR (immediate) â€“ 64-bit
11	0	10	PRFM (immediate)
11	1	00	STR (immediate, SIMD&FP) â€“ 64-bit
11	1	01	LDR (immediate, SIMD&FP) â€“ 64-bit

Data Processing -- Register

These instructions are under the [top-level](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
op0	op1	101		op2												op3																	

Decode fields				Instruction details
op0	op1	op2	op3	
0	1	0110		Data-processing (2 source)
1	1	0110		Data-processing (1 source)
	0	0xxx		Logical (shifted register)
	0	1xx0		Add/subtract (shifted register)
	0	1xx1		Add/subtract (extended register)
	1	0000	000000	Add/subtract (with carry)
	1	0000	x00001	Rotate right into flags
	1	0000	xx0010	Evaluate into flags
	1	0010	xxxx0x	Conditional compare (register)
	1	0010	xxxx1x	Conditional compare (immediate)
	1	0100		Conditional select

	1	1xxx		Data-processing (3 source)
--	---	------	--	--

Data-processing (2 source)

These instructions are under [Data Processing -- Register](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
sf	0	S	1	1	0	1	0	1	1	0	Rm	opcode	Rn	Rd																	

Decode fields			Instruction Details	Feature
sf	S	opcode		
		000001	UNALLOCATED	-
		1xxxxx	UNALLOCATED	-
	0	00011x	UNALLOCATED	-
	0	001101	UNALLOCATED	-
	0	00111x	UNALLOCATED	-
	0	0111xx	UNALLOCATED	-
	1	00001x	UNALLOCATED	-
	1	0001xx	UNALLOCATED	-
	1	001xxx	UNALLOCATED	-
	1	01xxxx	UNALLOCATED	-
0		000000	UNALLOCATED	-
0	0	000010	UDIV â€“ 32-bit	-
0	0	000011	SDIV â€“ 32-bit	-
0	0	00010x	UNALLOCATED	-
0	0	001000	LSLV â€“ 32-bit	-
0	0	001001	LSRV â€“ 32-bit	-
0	0	001010	ASRV â€“ 32-bit	-
0	0	001011	RORV â€“ 32-bit	-
0	0	001100	UNALLOCATED	-
0	0	010x11	UNALLOCATED	-
0	0	010000	CRC32B , CRC32H , CRC32W , CRC32X â€“ CRC32B	FEAT_CRC32
0	0	010001	CRC32B , CRC32H , CRC32W , CRC32X â€“ CRC32H	FEAT_CRC32
0	0	010010	CRC32B , CRC32H , CRC32W , CRC32X â€“ CRC32W	FEAT_CRC32
0	0	010100	CRC32CB , CRC32CH , CRC32CW , CRC32CX â€“ CRC32CB	FEAT_CRC32
0	0	010101	CRC32CB , CRC32CH , CRC32CW , CRC32CX â€“ CRC32CH	FEAT_CRC32
0	0	010110	CRC32CB , CRC32CH , CRC32CW , CRC32CX â€“ CRC32CW	FEAT_CRC32

Decode fields			Instruction Details	Feature
sf	S	opcode		
0	0	011000	SMAX (register) â€“ 32-bit	FEAT_CSSC
0	0	011001	UMAX (register) â€“ 32-bit	FEAT_CSSC
0	0	011010	SMIN (register) â€“ 32-bit	FEAT_CSSC
0	0	011011	UMIN (register) â€“ 32-bit	FEAT_CSSC
1	0	000000	SUBP	FEAT_MTE
1	0	000010	UDIV â€“ 64-bit	-
1	0	000011	SDIV â€“ 64-bit	-
1	0	000100	IRG	FEAT_MTE
1	0	000101	GMI	FEAT_MTE
1	0	001000	LSLV â€“ 64-bit	-
1	0	001001	LSRV â€“ 64-bit	-
1	0	001010	ASRV â€“ 64-bit	-
1	0	001011	RORV â€“ 64-bit	-
1	0	001100	PACGA	FEAT_PAuth
1	0	010xx0	UNALLOCATED	-
1	0	010x0x	UNALLOCATED	-
1	0	010011	CRC32B, CRC32H, CRC32W, CRC32X â€“ CRC32X	FEAT_CRC32
1	0	010111	CRC32CB, CRC32CH, CRC32CW, CRC32CX â€“ CRC32CX	FEAT_CRC32
1	0	011000	SMAX (register) â€“ 64-bit	FEAT_CSSC
1	0	011001	UMAX (register) â€“ 64-bit	FEAT_CSSC
1	0	011010	SMIN (register) â€“ 64-bit	FEAT_CSSC
1	0	011011	UMIN (register) â€“ 64-bit	FEAT_CSSC
1	1	000000	SUBPS	FEAT_MTE

Data-processing (1 source)

These instructions are under [Data Processing -- Register](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
sf	1	S	1	1	0	1	0	1	1	0	opcode2		opcode		Rn		Rd														

Decode fields			Instruction Details	Feature		
sf	S	opcode2	opcode	Rn		
			1xxxxx		UNALLOCATED	-
		xxx1x			UNALLOCATED	-
		xx1xx			UNALLOCATED	-
		x1xxx			UNALLOCATED	-
		1xxxx			UNALLOCATED	-

sf	S	Decode fields			Rn	Instruction Details	Feature
		opcode2	opcode				
	0	00000	001001			UNALLOCATED	-
	0	00000	00101x			UNALLOCATED	-
	0	00000	0011xx			UNALLOCATED	-
	0	00000	01xxxx			UNALLOCATED	-
	1					UNALLOCATED	-
	0		00001			UNALLOCATED	-
0	0	00000	000000			RBIT â€” 32-bit	-
0	0	00000	000001			REV16 â€” 32-bit	-
0	0	00000	000010			REV â€” 32-bit	-
0	0	00000	000011			UNALLOCATED	-
0	0	00000	000100			CLZ â€” 32-bit	-
0	0	00000	000101			CLS â€” 32-bit	-
0	0	00000	000110			CTZ â€” 32-bit	FEAT_CSSC
0	0	00000	000111			CNT â€” 32-bit	FEAT_CSSC
0	0	00000	001000			ABS â€” 32-bit	FEAT_CSSC
1	0	00000	000000			RBIT â€” 64-bit	-
1	0	00000	000001			REV16 â€” 64-bit	-
1	0	00000	000010			REV32	-
1	0	00000	000011			REV â€” 64-bit	-
1	0	00000	000100			CLZ â€” 64-bit	-
1	0	00000	000101			CLS â€” 64-bit	-
1	0	00000	000110			CTZ â€” 64-bit	FEAT_CSSC
1	0	00000	000111			CNT â€” 64-bit	FEAT_CSSC
1	0	00000	001000			ABS â€” 64-bit	FEAT_CSSC
1	0	00001	000000			PACIA , PACIA1716 , PACIASP , PACIAZ , PACIZA â€” PACIA	FEAT_PAuth
1	0	00001	000001			PACIB , PACIB1716 , PACIBSP , PACIBZ , PACIZB â€” PACIB	FEAT_PAuth
1	0	00001	000010			PACDA , PACDZA â€” PACDA	FEAT_PAuth
1	0	00001	000011			PACDB , PACDZB â€” PACDB	FEAT_PAuth

sf	S	Decode fields			Rn	Instruction Details	Feature
		opcode2	opcode				
1	0	00001	000100			AUTIA , AUTIA1716 , AUTIASP , AUTIAZ , AUTIZA â€” AUTIA	FEAT_PAuth
1	0	00001	000101			AUTIB , AUTIB1716 , AUTIBSP , AUTIBZ , AUTIZB â€” AUTIB	FEAT_PAuth
1	0	00001	000110			AUTDA , AUTDZA â€” AUTDA	FEAT_PAuth
1	0	00001	000111			AUTDB , AUTDZB â€” AUTDB	FEAT_PAuth
1	0	00001	001000	11111		PACIA , PACIA1716 , PACIASP , PACIAZ , PACIZA â€” PACIZA	FEAT_PAuth
1	0	00001	001001	11111		PACIB , PACIB1716 , PACIBSP , PACIBZ , PACIZB â€” PACIZB	FEAT_PAuth
1	0	00001	001010	11111		PACDA , PACDZA â€” PACDZA	FEAT_PAuth
1	0	00001	001011	11111		PACDB , PACDZB â€” PACDZB	FEAT_PAuth
1	0	00001	001100	11111		AUTIA , AUTIA1716 , AUTIASP , AUTIAZ , AUTIZA â€” AUTIZA	FEAT_PAuth
1	0	00001	001101	11111		AUTIB , AUTIB1716 , AUTIBSP , AUTIBZ , AUTIZB â€” AUTIZB	FEAT_PAuth
1	0	00001	001110	11111		AUTDA , AUTDZA â€” AUTDZA	FEAT_PAuth
1	0	00001	001111	11111		AUTDB , AUTDZB â€” AUTDZB	FEAT_PAuth
1	0	00001	010000	11111		XPACD , XPACL , XPACLRI â€” XPACI	FEAT_PAuth

Decode fields					Instruction Details	Feature
sf	S	opcode2	opcode	Rn		
1	0	00001	010001	11111	XPACD , XPACI , XPACLRI â€” XPACD	FEAT_PAuth
1	0	00001	01001x		UNALLOCATED	-
1	0	00001	0101xx		UNALLOCATED	-
1	0	00001	011xxx		UNALLOCATED	-

Logical (shifted register)

These instructions are under [Data Processing -- Register](#).

Decode fields				Instruction Details
sf	opc	N	imm6	
0			1xxxxx	UNALLOCATED
0	00	0		AND (shifted register) â€” 32-bit
0	00	1		BIC (shifted register) â€” 32-bit
0	01	0		ORR (shifted register) â€” 32-bit
0	01	1		ORN (shifted register) â€” 32-bit
0	10	0		EOR (shifted register) â€” 32-bit
0	10	1		EON (shifted register) â€” 32-bit
0	11	0		ANDS (shifted register) â€” 32-bit
0	11	1		BICS (shifted register) â€” 32-bit
1	00	0		AND (shifted register) â€” 64-bit
1	00	1		BIC (shifted register) â€” 64-bit
1	01	0		ORR (shifted register) â€” 64-bit
1	01	1		ORN (shifted register) â€” 64-bit
1	10	0		EOR (shifted register) â€” 64-bit
1	10	1		EON (shifted register) â€” 64-bit
1	11	0		ANDS (shifted register) â€” 64-bit
1	11	1		BICS (shifted register) â€” 64-bit

Add/subtract (shifted register)

These instructions are under [Data Processing -- Register](#).

Decode fields					Instruction Details
sf	op	S	shift	imm6	
			11		UNALLOCATED
0				1xxxxx	UNALLOCATED
0	0	0			ADD (shifted register) â€“ 32-bit
0	0	1			ADDS (shifted register) â€“ 32-bit
0	1	0			SUB (shifted register) â€“ 32-bit
0	1	1			SUBS (shifted register) â€“ 32-bit
1	0	0			ADD (shifted register) â€“ 64-bit
1	0	1			ADDS (shifted register) â€“ 64-bit
1	1	0			SUB (shifted register) â€“ 64-bit
1	1	1			SUBS (shifted register) â€“ 64-bit

Add/subtract (extended register)

These instructions are under [Data Processing -- Register](#).

Decode fields					Instruction Details
sf	op	S	opt	imm3	
				1x1	UNALLOCATED
				11x	UNALLOCATED
			x1		UNALLOCATED
			1x		UNALLOCATED
0	0	0	00		ADD (extended register) â€“ 32-bit
0	0	1	00		ADDS (extended register) â€“ 32-bit
0	1	0	00		SUB (extended register) â€“ 32-bit
0	1	1	00		SUBS (extended register) â€“ 32-bit
1	0	0	00		ADD (extended register) â€“ 64-bit
1	0	1	00		ADDS (extended register) â€“ 64-bit
1	1	0	00		SUB (extended register) â€“ 64-bit
1	1	1	00		SUBS (extended register) â€“ 64-bit

Add/subtract (with carry)

These instructions are under [Data Processing -- Register](#).

Decode fields			Instruction Details
sf	op	S	
0	0	0	ADC â€“ 32-bit
0	0	1	ADCS â€“ 32-bit
0	1	0	SBC â€“ 32-bit
0	1	1	SBCS â€“ 32-bit
1	0	0	ADC â€“ 64-bit
1	0	1	ADCS â€“ 64-bit
1	1	0	SBC â€“ 64-bit
1	1	1	SBCS â€“ 64-bit

Rotate right into flags

These instructions are under [Data Processing -- Register](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
sf	op	S	1	1	0	1	0	0	0	0	imm6	0	0	0	0	1	Rn	o2	mask												

Decode fields				Instruction Details	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

These instructions are under [Data Processing -- Register](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
sf	op	S	1	1	0	1	0	0	0	0	opcode2	sz	0	0	1	0	Rn	o3	mask												

Decode fields			Instruction Details	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 â€“ SETF8	FEAT_FlagM

sf	op	S	Decode fields				Instruction Details		Feature
			opcode2	sz	o3	mask			
0	0	1	000000	1	0	1101	SETF8, SETF16 etc SETF16	FEAT_FlagM	
0	1						UNALLOCATED	-	
1							UNALLOCATED	-	

Conditional compare (register)

These instructions are under [Data Processing -- Register](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
sf	op	S	1	1	0	1	0	0	1	0	Rm	cond	0	o2	Rn	o3	nzcv																

sf	op	S	Decode fields		Instruction Details	
			o2	o3		
				1	UNALLOCATED	
				1	UNALLOCATED	
			0		UNALLOCATED	
0	0	1	0	0	CCMN (register) etc 32-bit	
0	1	1	0	0	CCMP (register) etc 32-bit	
1	0	1	0	0	CCMN (register) etc 64-bit	
1	1	1	0	0	CCMP (register) etc 64-bit	

Conditional compare (immediate)

These instructions are under [Data Processing -- Register](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
sf	op	S	1	1	0	1	0	0	1	0	imm5	cond	1	o2	Rn	o3	nzcv																

sf	op	S	Decode fields		Instruction Details	
			o2	o3		
				1	UNALLOCATED	
				1	UNALLOCATED	
			0		UNALLOCATED	
0	0	1	0	0	CCMN (immediate) etc 32-bit	
0	1	1	0	0	CCMP (immediate) etc 32-bit	
1	0	1	0	0	CCMN (immediate) etc 64-bit	
1	1	1	0	0	CCMP (immediate) etc 64-bit	

Conditional select

These instructions are under [Data Processing -- Register](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
sf	op	S	1	1	0	1	0	1	0	0	Rm		cond	op2		Rn		Rd													

Decode fields				Instruction Details
sf	op	S	op2	
			1x	UNALLOCATED
		1		UNALLOCATED
0	0	0	00	CSEL â€“ 32-bit
0	0	0	01	CSINC â€“ 32-bit
0	1	0	00	CSINV â€“ 32-bit
0	1	0	01	CSNEG â€“ 32-bit
1	0	0	00	CSEL â€“ 64-bit
1	0	0	01	CSINC â€“ 64-bit
1	1	0	00	CSINV â€“ 64-bit
1	1	0	01	CSNEG â€“ 64-bit

Data-processing (3 source)

These instructions are under [Data Processing -- Register](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
sf	op54	1	1	0	1	1	op31		Rm	00	Ra		Rn		Rd																

Decode fields				Instruction Details
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 â€“ 32-bit
0	00	000	1	MSUB â€“ 32-bit
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

sf	Decode fields			Instruction Details
	op54	op31	o0	
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

Data Processing -- Scalar Floating-Point and Advanced SIMD

These instructions are under the [top-level](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
op0	111	op1	op2																												

op0	Decode fields			Instruction details	Feature
	op1	op2	op3		
0000	0x	x101	00xxxxx10	UNALLOCATED	-
0010	0x	x101	00xxxxx10	UNALLOCATED	-
0100	0x	x101	00xxxxx10	Cryptographic AES	-
0101	0x	x0xx	xxx0xxx00	Cryptographic three-register SHA	-
0101	0x	x0xx	xxx0xxx10	UNALLOCATED	-
0101	0x	x101	00xxxxx10	Cryptographic two-register SHA	-
0110	0x	x101	00xxxxx10	UNALLOCATED	-
0111	0x	x0xx	xxx0xxxx0	UNALLOCATED	-
0111	0x	x101	00xxxxx10	UNALLOCATED	-
01x1	00	00xx	xxx0xxxx1	Advanced SIMD scalar copy	-
01x1	01	00xx	xxx0xxxx1	UNALLOCATED	-
01x1	0x	0111	00xxxxx10	UNALLOCATED	-
01x1	0x	10xx	xxx00xxx1	Advanced SIMD scalar three same FP16	-
01x1	0x	10xx	xxx01xxx1	UNALLOCATED	-
01x1	0x	1111	00xxxxx10	Advanced SIMD scalar two-register miscellaneous FP16	-
01x1	0x	x0xx	xxx1xxxx0	UNALLOCATED	-
01x1	0x	x0xx	xxx1xxxx1	Advanced SIMD scalar three same extra	-

01x1	0x	x100	00xxxxx10	Advanced SIMD scalar two-register miscellaneous	-
01x1	0x	x110	00xxxxx10	Advanced SIMD scalar pairwise	-
01x1	0x	x1xx	1xxxxxx10	UNALLOCATED	-
01x1	0x	x1xx	x1xxxxx10	UNALLOCATED	-
01x1	0x	x1xx	xxxxxxx00	Advanced SIMD scalar three different	-
01x1	0x	x1xx	xxxxxxxx1	Advanced SIMD scalar three same	-
01x1	10		xxxxxxxx1	Advanced SIMD scalar shift by immediate	-
01x1	11		xxxxxxxx1	UNALLOCATED	-
01x1	1x		xxxxxxxx0	Advanced SIMD scalar x indexed element	-
0x00	0x	x0xx	xxx0xxx00	Advanced SIMD table lookup	-
0x00	0x	x0xx	xxx0xxx10	Advanced SIMD permute	-
0x10	0x	x0xx	xxx0xxxx0	Advanced SIMD extract	-
0xx0	00	00xx	xxx0xxxx1	Advanced SIMD copy	-
0xx0	01	00xx	xxx0xxxx1	UNALLOCATED	-
0xx0	0x	0111	00xxxxx10	UNALLOCATED	-
0xx0	0x	10xx	xxx00xxx1	Advanced SIMD three same (FP16)	-
0xx0	0x	10xx	xxx01xxx1	UNALLOCATED	-
0xx0	0x	1111	00xxxxx10	Advanced SIMD two-register miscellaneous (FP16)	-
0xx0	0x	x0xx	xxx1xxxx0	UNALLOCATED	-
0xx0	0x	x0xx	xxx1xxxx1	Advanced SIMD three-register extension	-
0xx0	0x	x100	00xxxxx10	Advanced SIMD two-register miscellaneous	-
0xx0	0x	x110	00xxxxx10	Advanced SIMD across lanes	-
0xx0	0x	x1xx	1xxxxxx10	UNALLOCATED	-
0xx0	0x	x1xx	x1xxxxx10	UNALLOCATED	-
0xx0	0x	x1xx	xxxxxxxx00	Advanced SIMD three different	-
0xx0	0x	x1xx	xxxxxxxx1	Advanced SIMD three same	-
0xx0	10	0000	xxxxxxxx1	Advanced SIMD modified immediate	-

0xx0	10	\neq 0000	xxxxxxxx1	Advanced SIMD shift by immediate	-
0xx0	11		xxxxxxxx1	UNALLOCATED	-
0xx0	1x		xxxxxxxx0	Advanced SIMD vector x indexed element	-
1100	00	10xx	xxx10xxxx	Cryptographic three-register, imm2	-
1100	00	11xx	xxx1x00xx	Cryptographic three-register SHA 512	-
1100	00		xxx0xxxx	Cryptographic four-register	-
1100	01	00xx		XAR	FEAT_SHA3
1100	01	1000	0001000xx	Cryptographic two-register SHA 512	-
1xx0	1x			UNALLOCATED	-
x0x1	0x	x0xx		Conversion between floating-point and fixed-point	-
x0x1	0x	x1xx	xxx000000	Conversion between floating-point and integer	-
x0x1	0x	x1xx	xxxx10000	Floating-point data-processing (1 source)	-
x0x1	0x	x1xx	xxxxx1000	Floating-point compare	-
x0x1	0x	x1xx	xxxxxx100	Floating-point immediate	-
x0x1	0x	x1xx	xxxxxxxx01	Floating-point conditional compare	-
x0x1	0x	x1xx	xxxxxxxx10	Floating-point data-processing (2 source)	-
x0x1	0x	x1xx	xxxxxxxx11	Floating-point conditional select	-
x0x1	1x			Floating-point data-processing (3 source)	-

Cryptographic AES

These instructions are under [Data Processing -- Scalar Floating-Point and Advanced SIMD](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	1	1	1	0	size	1	0	1	0	0	opcode	1	0	Rn		Rd												

Decode fields
size **opcode**

Instruction Details

Feature

	x1xxx	UNALLOCATED	-
--	-------	-------------	---

Decode fields	Instruction Details	Feature	
size	opcode		
	000xx	UNALLOCATED	-
	1xxxx	UNALLOCATED	-
x1		UNALLOCATED	-
00	00100	AESE	FEAT_AES
00	00101	AESD	FEAT_AES
00	00110	AESMC	FEAT_AES
00	00111	AESIMC	FEAT_AES
1x		UNALLOCATED	-

Cryptographic three-register SHA

These instructions are under [Data Processing -- Scalar Floating-Point and Advanced SIMD](#).

Decode fields		Instruction Details	Feature
size	opcode		
	111	UNALLOCATED	-
x1		UNALLOCATED	-
00	000	SHA1C	FEAT_SHA1
00	001	SHA1P	FEAT_SHA1
00	010	SHA1M	FEAT_SHA1
00	011	SHA1SU0	FEAT_SHA1
00	100	SHA256H	FEAT_SHA256
00	101	SHA256H2	FEAT_SHA256
00	110	SHA256SU1	FEAT_SHA256
1x		UNALLOCATED	-

Cryptographic two-register SHA

These instructions are under [Data Processing -- Scalar Floating-Point](#) and [Advanced SIMD](#).

Decode fields		Instruction Details	Feature
size	opcode		
	xx1xx	UNALLOCATED	-
	x1xxx	UNALLOCATED	-
	1xxxx	UNALLOCATED	-
x1		UNALLOCATED	-
00	00000	SHA1H	FEAT_SHA1
00	00001	SHA1SU1	FEAT_SHA1
00	00010	SHA256SU0	FEAT_SHA256
00	00011	UNALLOCATED	-
1x		UNALLOCATED	-

Advanced SIMD scalar copy

These instructions are under [Data Processing -- Scalar Floating-Point and Advanced SIMD](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
0	1	op	1	1	1	1	0	0	0	0	imm5	0	imm4	1	Rn		Rd																

Decode fields		Instruction Details
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

These instructions are under [Data Processing -- Scalar Floating-Point and Advanced SIMD](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
0	1	U	1	1	1	1	0	a	1	0	Rm	0	0	opcode	1	Rn		Rd															

Decode fields			Instruction Details	Feature
U	a	opcode		
		110	UNALLOCATED	-
	1	011	UNALLOCATED	-

Decode fields			Instruction Details	Feature
U	a	opcode		
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
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

These instructions are under [Data Processing -- Scalar Floating-Point and Advanced SIMD](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	U	1	1	1	1	0	a	1	1	1	1	0	0	opcode	1	0	Rn		Rd											

Decode fields			Instruction Details	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

Decode fields			Instruction Details	Feature
U	a	opcode		
0	1	01100	FCMGT (zero)	FEAT_FP16
0	1	01101	FCMEQ (zero)	FEAT_FP16
0	1	01110	FCMLT (zero)	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	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

These instructions are under [Data Processing -- Scalar Floating-Point and Advanced SIMD](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0	1	U	1	1	1	1	0	size	0	Rm	1	opcode	1	Rn		Rd																

Decode fields			Instruction Details	Feature
U	a	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

These instructions are under [Data Processing -- Scalar Floating-Point and Advanced SIMD](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	U	1	1	1	1	0	size	1	0	0	0	0	opcode	1	0	Rn		Rd												

Decode fields			Instruction Details
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)
0	0x	11011	FCVTMS (vector)
0	0x	11100	FCVTAS (vector)
0	0x	11101	SCVTF (vector, integer)
0	1x	01100	FCMGT (zero)
0	1x	01101	FCMEO (zero)
0	1x	01110	FCMLT (zero)
0	1x	11010	FCVTPS (vector)
0	1x	11011	FCVTZS (vector, integer)

Decode fields			Instruction Details
U	size	opcode	
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

These instructions are under [Data Processing -- Scalar Floating-Point and Advanced SIMD](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	U	1	1	1	1	0	size	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Rn	Rd	

Decode fields			Instruction Details	Feature
U	size	opcode		
		00xxx	UNALLOCATED	-
		010xx	UNALLOCATED	-
		01110	UNALLOCATED	-
		10xxx	UNALLOCATED	-
		1100x	UNALLOCATED	-
		11010	UNALLOCATED	-

Decode fields			Instruction Details	Feature
U	size	opcode		
		111xx	UNALLOCATED	-
	1x	01101	UNALLOCATED	-
0		11011	ADDP (scalar)	-
0	0x	01100	FMAXNMP (scalar) â€“ half-precision	FEAT_FP16
0	0x	01101	FADDP (scalar) â€“ half-precision	FEAT_FP16
0	0x	01111	FMAXP (scalar) â€“ half-precision	FEAT_FP16
0	1x	01100	FMINNMP (scalar) â€“ half-precision	FEAT_FP16
0	1x	01111	FMINP (scalar) â€“ half-precision	FEAT_FP16
1		11011	UNALLOCATED	-
1	0x	01100	FMAXNMP (scalar) â€“ single-precision and double-precision	-
1	0x	01101	FADDP (scalar) â€“ single-precision and double-precision	-
1	0x	01111	FMAXP (scalar) â€“ single-precision and double-precision	-
1	1x	01100	FMINNMP (scalar) â€“ single-precision and double-precision	-
1	1x	01111	FMINP (scalar) â€“ single-precision and double-precision	-

Advanced SIMD scalar three different

These instructions are under [Data Processing -- Scalar Floating-Point and Advanced SIMD](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	U	1	1	1	1	0	size	1	Rm		opcode	0	0	Rn		Rd														

Decode fields			Instruction Details
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	

Decode fields		Instruction Details
U	opcode	
1	1011	UNALLOCATED
1	1101	UNALLOCATED

Advanced SIMD scalar three same

These instructions are under [Data Processing -- Scalar Floating-Point and Advanced SIMD](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	U	1	1	1	1	0	size	1	Rm		opcode	1	Rn		Rd															

Decode fields		Instruction Details	
U	size	opcode	
		00000	UNALLOCATED
		0001x	UNALLOCATED
		00100	UNALLOCATED
		011xx	UNALLOCATED
		1001x	UNALLOCATED
	1x	11011	UNALLOCATED
0		00001	SQADD
0		00101	SQSUB
0		00110	CMGT (register)
0		00111	CMGE (register)
0		01000	SSH
0		01001	SOSHL (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

Decode fields			Instruction Details
U	size	opcode	
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
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

These instructions are under [Data Processing -- Scalar Floating-Point and Advanced SIMD](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	U	1	1	1	1	0		immh		immb		opcode		1		Rn		Rd												

Decode fields			Instruction Details
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
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

Decode fields			Instruction Details
U	immh	opcode	
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

These instructions are under [Data Processing -- Scalar Floating-Point and Advanced SIMD](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	U	1	1	1	1	1	size	L	M	Rm	opcode	H	0	Rn		Rd														

Decode fields			Instruction Details	Feature
U	size	opcode		
		0000	UNALLOCATED	-
		0010	UNALLOCATED	-
		0100	UNALLOCATED	-
		0110	UNALLOCATED	-
		1000	UNALLOCATED	-
		1010	UNALLOCATED	-
		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	-

Decode fields			Instruction Details	Feature
U	size	opcode		
0	00	0001	FMLA (by element) â€“ half-precision	FEAT_FP16
0	00	0101	FMLS (by element) â€“ half-precision	FEAT_FP16
0	00	1001	FMUL (by element) â€“ half-precision	FEAT_FP16
0	1x	0001	FMLA (by element) â€“ single-precision and double-precision	-
0	1x	0101	FMLS (by element) â€“ single-precision and double-precision	-
0	1x	1001	FMUL (by element) â€“ single-precision and double-precision	-
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) â€“ half-precision	FEAT_FP16
1	1x	0001	UNALLOCATED	-
1	1x	0101	UNALLOCATED	-
1	1x	1001	FMULX (by element) â€“ single-precision and double-precision	-

Advanced SIMD table lookup

These instructions are under [Data Processing -- Scalar Floating-Point and Advanced SIMD](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	Q	0	0	1	1	1	0	op2	0	Rm	0	len	op	0	0	Rn	Rd														

Decode fields			Instruction Details
op2	len	op	
x1			UNALLOCATED
00	00	0	TBL â€“ single register table
00	00	1	TBX â€“ single register table
00	01	0	TBL â€“ two register table
00	01	1	TBX â€“ two register table

Decode fields			Instruction Details									
op2	len	op										
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								
1x			UNALLOCATED									

Advanced SIMD permute

These instructions are under [Data Processing -- Scalar Floating-Point and Advanced SIMD](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	Q	0	0	1	1	1	0	size	0	Rm	0	opcode	1	0		Rn		Rd													

Decode fields		Instruction Details									
opcode											
000		UNALLOCATED									
001		UZP1									
010		TRN1									
011		ZIP1									
100		UNALLOCATED									
101		UZP2									
110		TRN2									
111		ZIP2									

Advanced SIMD extract

These instructions are under [Data Processing -- Scalar Floating-Point and Advanced SIMD](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	Q	1	0	1	1	1	0	op2	0	Rm	0	imm4	0	Rn		Rd															

Decode fields		Instruction Details									
op2											
x1		UNALLOCATED									
00		EXT									
1x		UNALLOCATED									

Advanced SIMD copy

These instructions are under [Data Processing -- Scalar Floating-Point and Advanced SIMD](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0	Q	op	0	1	1	1	0	0	0	0	imm5	0	imm4	1	Rn		Rd															

Q	Decode fields			Instruction Details
	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
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)

These instructions are under [Data Processing -- Scalar Floating-Point and Advanced SIMD](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	Q	U	0	1	1	1	0	a	1	0	Rm	0	0	opcode	1	Rn		Rd													

U	a	Decode fields		Instruction Details	Feature
		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

Decode fields			Instruction Details	Feature
U	a	opcode		
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
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)

These instructions are under [Data Processing -- Scalar Floating-Point and Advanced SIMD](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	Q	U	0	1	1	1	0	a	1	1	1	1	0	0	opcode	1	0	Rn		Rd											

Decode fields			Instruction Details	Feature
U	a	opcode		
		00xxx	UNALLOCATED	-
		010xx	UNALLOCATED	-
		10xxx	UNALLOCATED	-
		11110	UNALLOCATED	-

Decode fields			Instruction Details	Feature
U	a	opcode		
	0	011xx	UNALLOCATED	-
	0	11111	UNALLOCATED	-
	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	FNNEG (vector)	FEAT_FP16
1	1	11000	UNALLOCATED	-
1	1	11001	FRINTI (vector)	FEAT_FP16
1	1	11010	FCVTPU (vector)	FEAT_FP16
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

These instructions are under [Data Processing -- Scalar Floating-Point and Advanced SIMD](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	Q	U	0	1	1	1	0	size	0	Rm	1	opcode	1	Rn	Rd																

Decode fields				Instruction Details	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	-
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

These instructions are under [Data Processing -- Scalar Floating-Point and Advanced SIMD](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	Q	U	0	1	1	1	0	size	1	0	0	0	0	opcode	1	0	Rn		Rd												

Decode fields			Instruction Details		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		-
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)	-	

Decode fields			Instruction Details	Feature
U	size	opcode		
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	-
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	-

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

Advanced SIMD across lanes

These instructions are under [Data Processing -- Scalar Floating-Point and Advanced SIMD](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	Q	U	0	1	1	1	0	size	1	1	0	0	0	0	opcode	1	0	Rn		Rd											

Decode fields			Instruction Details	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 â€“ half-precision	FEAT_FP16
0	00	01111	FMAXV â€“ half-precision	FEAT_FP16
0	01	01100	UNALLOCATED	-
0	01	01111	UNALLOCATED	-
0	10	01100	FMINNMV â€“ half-precision	FEAT_FP16

Decode fields			Instruction Details	Feature
U	size	opcode		
0	10	01111	FMINV â€“ half-precision	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 â€“ single-precision and double-precision	-
1	0x	01111	FMAXV â€“ single-precision and double-precision	-
1	1x	01100	FMINNMV â€“ single-precision and double-precision	-
1	1x	01111	FMINV â€“ single-precision and double-precision	-

Advanced SIMD three different

These instructions are under [Data Processing -- Scalar Floating-Point and Advanced SIMD](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0	Q	U	0	1	1	1	0	size	1	Rm	opcode	0	0	Rn	Rd																	

Decode fields		Instruction Details
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	SMLS1, SMLS12 (vector)
0	1011	SQDMLSL, SQDMLSL2 (vector)
0	1100	SMULL, SMULL2 (vector)
0	1101	SQDMULL, SQDMULL2 (vector)

Decode fields		Instruction Details
U	opcode	
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

These instructions are under [Data Processing -- Scalar Floating-Point and Advanced SIMD](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	Q	U	0	1	1	1	0	size	1	Rm	opcode	1	Rn	Rd																	

Decode fields			Instruction Details	Feature
U	size	opcode		
0		00000	SHADD	-
0		00001	SQADD	-
0		00010	SRHADD	-
0		00100	SHSUB	-
0		00101	SQSUB	-
0		00110	CMGT (register)	-
0		00111	CMGE (register)	-
0		01000	SSH	-
0		01001	SQSHL (register)	-
0		01010	SRSHL	-
0		01011	SQRSHL	-
0		01100	SMAX	-

Decode fields			Instruction Details	Feature
U	size	opcode		
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	FCMEO (register)	-
0	0x	11110	FMAX (vector)	-
0	0x	11111	FRECPS	-
0	00	00011	AND (vector)	-
0	00	11101	FMLAL, FMLAL2 (vector) â€“ FMLAL	FEAT_FHM
0	01	00011	BIC (vector, register)	-
0	01	11101	UNALLOCATED	-
0	1x	11000	FMINNM (vector)	-
0	1x	11001	FMLS (vector)	-
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	FMLS, FMLS2 (vector) â€“ FMLS	FEAT_FHM
0	11	00011	ORN (vector)	-
0	11	11101	UNALLOCATED	-
1		00000	UHADD	-
1		00001	UQADD	-
1		00010	URHADD	-
1		00100	UHSUB	-

Decode fields			Instruction Details	Feature
U	size	opcode		
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	SQRDMDLH (vector)	-
1		10111	UNALLOCATED	-
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) â€” FMLAL2	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	-

Decode fields			Instruction Details	Feature
U	size	opcode		
1	10	00011	BIT	-
1	10	11001	FMLS , FMLS2 (vector) â€“ FMLS2	FEAT_FHM
1	11	00011	BIF	-
1	11	11001	UNALLOCATED	-

Advanced SIMD modified immediate

These instructions are under [Data Processing -- Scalar Floating-Point and Advanced SIMD](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
0	Q	op	0	1	1	1	1	0	0	0	0	a	b	c	cmode	o2	1	d	e	f	g	h	Rd										

Decode fields			Instruction Details	Feature	
Q	op	cmode	o2		
	0	0xxx	1	UNALLOCATED	-
	0	0xx0	0	MOVI â€“ 32-bit shifted immediate	-
	0	0xx1	0	ORR (vector, immediate) â€“ 32-bit	-
	0	10xx	1	UNALLOCATED	-
	0	10x0	0	MOVI â€“ 16-bit shifted immediate	-
	0	10x1	0	ORR (vector, immediate) â€“ 16-bit	-
	0	110x	0	MOVI â€“ 32-bit shifting ones	-
	0	110x	1	UNALLOCATED	-
	0	1110	0	MOVI â€“ 8-bit	-
	0	1110	1	UNALLOCATED	-
	0	1111	0	FMOV (vector, immediate) â€“ single-precision	-
	0	1111	1	FMOV (vector, immediate) â€“ half-precision	FEAT_FP16
	1		1	UNALLOCATED	-
	1	0xx0	0	MVNI â€“ 32-bit shifted immediate	-
	1	0xx1	0	BIC (vector, immediate) â€“ 32-bit	-
	1	10x0	0	MVNI â€“ 16-bit shifted immediate	-
	1	10x1	0	BIC (vector, immediate) â€“ 16-bit	-

Decode fields				Instruction Details	Feature
Q	op	cmode	o2		
	1	110x	0	MVNI â€“ 32-bit shifting ones	-
0	1	1110	0	MOVI â€“ 64-bit scalar	-
0	1	1111	0	UNALLOCATED	-
1	1	1110	0	MOVI â€“ 64-bit vector	-
1	1	1111	0	FMOV (vector, immediate) â€“ double-precision	-

Advanced SIMD shift by immediate

These instructions are under [Data Processing -- Scalar Floating-Point and Advanced SIMD](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
0	Q	U	0	1	1	1	1	0	!= 0000	immh	immb	opcode	1	Rn		Rd																	

The following constraints also apply to this encoding: immh != 0000 && immh != 0000

Decode fields		Instruction Details
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

Decode fields		Instruction Details
U	opcode	
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
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

These instructions are under [Data Processing -- Scalar Floating-Point and Advanced SIMD](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
0	Q	U	0	1	1	1	1	size	L	M	Rm	opcode	H	0	Rn	Rd																		

Decode fields			Instruction Details	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)	-

Decode fields			Instruction Details	Feature
U	size	opcode		
0		0111	SQDMLSL, SQDMLSL2 (by element)	-
0		1000	MUL (by element)	-
0		1010	SMULL, SMULL2 (by element)	-
0		1011	SQDMULL, SQDMULL2 (by element)	-
0		1100	SQDMULH (by element)	-
0		1101	SORDMULH (by element)	-
0		1110	SDOT (by element)	FEAT_DotProd
0	0x	0000	UNALLOCATED	-
0	0x	0100	UNALLOCATED	-
0	00	0001	FMLA (by element) â€“ half-precision	FEAT_FP16
0	00	0101	FMLS (by element) â€“ half-precision	FEAT_FP16
0	00	1001	FMUL (by element) â€“ half-precision	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) â€“ single-precision and double-precision	-
0	1x	0101	FMLS (by element) â€“ single-precision and double-precision	-
0	1x	1001	FMUL (by element) â€“ single-precision and double-precision	-
0	10	0000	FMLAL, FMLAL2 (by element) â€“ FMLAL	FEAT_FHM
0	10	0100	FMLSL, FMLSL2 (by element) â€“ FMLSL	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)	-

Decode fields			Instruction Details	Feature
U	size	opcode		
1		1011	UNALLOCATED	-
1		1101	SQRDMLAH (by element)	FEAT_RDM
1		1110	UDOT (by element)	FEAT_DotProd
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) â€“ half-precision	FEAT_FP16
1	01	0xx1	FCMLA (by element)	FEAT_FCMA
1	1x	1001	FMULX (by element) â€“ single-precision and double-precision	-
1	10	0xx1	FCMLA (by element)	FEAT_FCMA
1	10	1000	FMLAL, FMLAL2 (by element) â€“ FMLAL2	FEAT_FHM
1	10	1100	FMLS, FMLS2 (by element) â€“ FMLS2	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

These instructions are under [Data Processing -- Scalar Floating-Point and Advanced SIMD](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	1	1	1	0	0	1	0	Rm	1	0	imm2	opcode	Rn		Rd													

Decode fields		Instruction Details	Feature
	opcode		
	00	SM3TT1A	FEAT_SM3
	01	SM3TT1B	FEAT_SM3
	10	SM3TT2A	FEAT_SM3

Decode fields opcode	Instruction Details	Feature
11	SM3TT2B	FEAT_SM3

Cryptographic three-register SHA 512

These instructions are under [Data Processing -- Scalar Floating-Point and Advanced SIMD](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
1	1	0	0	1	1	1	0	0	1	1	Rm	1	0	0	0	opcode	Rn											Rd				

Decode fields O opcode	Instruction Details	Feature
0	SHA512H	FEAT_SHA512
0	SHA512H2	FEAT_SHA512
0	SHA512SU1	FEAT_SHA512
0	RAX1	FEAT_SHA3
1	SM3PARTW1	FEAT_SM3
1	SM3PARTW2	FEAT_SM3
1	SM4EKEY	FEAT_SM4
1	UNALLOCATED	-

Cryptographic four-register

These instructions are under [Data Processing -- Scalar Floating-Point and Advanced SIMD](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	1	1	1	0	0	Op0		Rm	0		Ra		Rn		Rd													

Decode fields Op0	Instruction Details	Feature
00	FOR3	FEAT_SHA3
01	BCAX	FEAT_SHA3
10	SM3SS1	FEAT_SM3
11	UNALLOCATED	-

Cryptographic two-register SHA 512

These instructions are under [Data Processing -- Scalar Floating-Point and Advanced SIMD](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	1	1	1	0	1	1	0	0	0	0	0	0	1	0	0	0	0	opcode	Rn		Rd							

Decode fields opcode

Instruction Details

Feature

00	SHA512SU0	FEAT_SHA512
01	SM4E	FEAT_SM4
1x	UNALLOCATED	-

Conversion between floating-point and fixed-point

These instructions are under [Data Processing -- Scalar Floating-Point and Advanced SIMD](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
sf	0	S	1	1	1	1	0	ftype	0	rmode	opcode		scale			Rn		Rd													

Decode fields						Instruction Details	Feature
sf	S	ftype	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) â€“ 32-bit to single-precision	-
0	0	00	00	011		UCVTF (scalar, fixed-point) â€“ 32-bit to single-precision	-
0	0	00	11	000		FCVTZS (scalar, fixed-point) â€“ single-precision to 32-bit	-
0	0	00	11	001		FCVTZU (scalar, fixed-point) â€“ single-precision to 32-bit	-

Decode fields						Instruction Details	Feature
sf	S	fptype	rmode	opcode	scale		
0	0	01	00	010		SCVTF (scalar, fixed-point) â€“ 32-bit to double-precision	-
0	0	01	00	011		UCVTF (scalar, fixed-point) â€“ 32-bit to double-precision	-
0	0	01	11	000		FCVTZS (scalar, fixed-point) â€“ double-precision to 32-bit	-
0	0	01	11	001		FCVTZU (scalar, fixed-point) â€“ double-precision to 32-bit	-
0	0	11	00	010		SCVTF (scalar, fixed-point) â€“ 32-bit to half-precision	FEAT_FP16
0	0	11	00	011		UCVTF (scalar, fixed-point) â€“ 32-bit to half-precision	FEAT_FP16
0	0	11	11	000		FCVTZS (scalar, fixed-point) â€“ half-precision to 32-bit	FEAT_FP16
0	0	11	11	001		FCVTZU (scalar, fixed-point) â€“ half-precision to 32-bit	FEAT_FP16
1	0	00	00	010		SCVTF (scalar, fixed-point) â€“ 64-bit to single-precision	-
1	0	00	00	011		UCVTF (scalar, fixed-point) â€“ 64-bit to single-precision	-

Decode fields						Instruction Details	Feature
sf	S	ftype	rmode	opcode	scale		
1	0	00	11	000		FCVTZS (scalar, fixed-point) â€“ single-precision to 64-bit	-
1	0	00	11	001		FCVTZU (scalar, fixed-point) â€“ single-precision to 64-bit	-
1	0	01	00	010		SCVTF (scalar, fixed-point) â€“ 64-bit to double-precision	-
1	0	01	00	011		UCVTF (scalar, fixed-point) â€“ 64-bit to double-precision	-
1	0	01	11	000		FCVTZS (scalar, fixed-point) â€“ double-precision to 64-bit	-
1	0	01	11	001		FCVTZU (scalar, fixed-point) â€“ double-precision to 64-bit	-
1	0	11	00	010		SCVTF (scalar, fixed-point) â€“ 64-bit to half-precision	FEAT_FP16
1	0	11	00	011		UCVTF (scalar, fixed-point) â€“ 64-bit to half-precision	FEAT_FP16
1	0	11	11	000		FCVTZS (scalar, fixed-point) â€“ half-precision to 64-bit	FEAT_FP16
1	0	11	11	001		FCVTZU (scalar, fixed-point) â€“ half-precision to 64-bit	FEAT_FP16

Conversion between floating-point and integer

These instructions are under [Data Processing -- Scalar Floating-Point and Advanced SIMD](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
sf	0	S	1	1	1	1	0	ftype	1	rmode	opcode	0	0	0	0	0	0	Rn		Rd											

sf	S	Decode fields			opcode	Instruction Details		Feature
		ftype	rmode	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) â€” single-precision to 32-bit	-	
0	0	00	00		001	FCVTNU (scalar) â€” single-precision to 32-bit	-	
0	0	00	00		010	SCVTF (scalar, integer) â€” 32-bit to single-precision	-	
0	0	00	00		011	UCVTF (scalar, integer) â€” 32-bit to single-precision	-	
0	0	00	00		100	FCVTAS (scalar) â€” single-precision to 32-bit	-	
0	0	00	00		101	FCVTAU (scalar) â€” single-precision to 32-bit	-	
0	0	00	00		110	FMOV (general) â€” single-precision to 32-bit	-	

sf	S	Decode fields		opcode	Instruction Details	Feature
		ftype	rmode			
0	0	00	00	111	FMOV (general) â€” 32-bit to single-precision	-
0	0	00	01	000	FCVTPS (scalar) â€” single-precision to 32-bit	-
0	0	00	01	001	FCVTPU (scalar) â€” single-precision to 32-bit	-
0	0	00	1x	11x	UNALLOCATED	-
0	0	00	10	000	FCVTMS (scalar) â€” single-precision to 32-bit	-
0	0	00	10	001	FCVTMU (scalar) â€” single-precision to 32-bit	-
0	0	00	11	000	FCVTZS (scalar, integer) â€” single-precision to 32-bit	-
0	0	00	11	001	FCVTZU (scalar, integer) â€” single-precision to 32-bit	-
0	0	01	0x	11x	UNALLOCATED	-
0	0	01	00	000	FCVTNS (scalar) â€” double-precision to 32-bit	-
0	0	01	00	001	FCVTNU (scalar) â€” double-precision to 32-bit	-
0	0	01	00	010	SCVTF (scalar, integer) â€” 32-bit to double-precision	-
0	0	01	00	011	UCVTF (scalar, integer) â€” 32-bit to double-precision	-

sf	S	Decode fields		opcode	Instruction Details	Feature
		ftype	rmode			
0	0	01	00	100	FCVTAS (scalar) â€” double-precision to 32-bit	-
0	0	01	00	101	FCVTAU (scalar) â€” double-precision to 32-bit	-
0	0	01	01	000	FCVTPS (scalar) â€” double-precision to 32-bit	-
0	0	01	01	001	FCVTPU (scalar) â€” double-precision to 32-bit	-
0	0	01	10	000	FCVTMS (scalar) â€” double-precision to 32-bit	-
0	0	01	10	001	FCVTMU (scalar) â€” double-precision to 32-bit	-
0	0	01	10	11x	UNALLOCATED	-
0	0	01	11	000	FCVTZS (scalar, integer) â€” double-precision to 32-bit	-
0	0	01	11	001	FCVTZU (scalar, integer) â€” double-precision to 32-bit	-
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) â€” half-precision to 32-bit	FEAT_FP16
0	0	11	00	001	FCVTNU (scalar) â€” half-precision to 32-bit	FEAT_FP16

sf	S	Decode fields		opcode	Instruction Details	Feature
		fptype	rmode			
0	0	11	00	010	SCVTF (scalar integer) â€“ 32-bit to half-precision	FEAT_FP16
0	0	11	00	011	UCVTF (scalar integer) â€“ 32-bit to half-precision	FEAT_FP16
0	0	11	00	100	FCVTAS (scalar) â€“ half-precision to 32-bit	FEAT_FP16
0	0	11	00	101	FCVTAU (scalar) â€“ half-precision to 32-bit	FEAT_FP16
0	0	11	00	110	FMOV (general) â€“ half-precision to 32-bit	FEAT_FP16
0	0	11	00	111	FMOV (general) â€“ 32-bit to half-precision	FEAT_FP16
0	0	11	01	000	FCVTPS (scalar) â€“ half-precision to 32-bit	FEAT_FP16
0	0	11	01	001	FCVTPU (scalar) â€“ half-precision to 32-bit	FEAT_FP16
0	0	11	10	000	FCVTMS (scalar) â€“ half-precision to 32-bit	FEAT_FP16
0	0	11	10	001	FCVTMU (scalar) â€“ half-precision to 32-bit	FEAT_FP16
0	0	11	11	000	FCVTZS (scalar integer) â€“ half-precision to 32-bit	FEAT_FP16
0	0	11	11	001	FCVTZU (scalar integer) â€“ half-precision to 32-bit	FEAT_FP16

sf	S	Decode fields		opcode	Instruction Details	Feature
		fptype	rmode			
1	0	00		11x	UNALLOCATED	-
1	0	00	00	000	FCVTNS (scalar) â€” single-precision to 64-bit	-
1	0	00	00	001	FCVTNU (scalar) â€” single-precision to 64-bit	-
1	0	00	00	010	SCVTF (scalar, integer) â€” 64-bit to single-precision	-
1	0	00	00	011	UCVTF (scalar, integer) â€” 64-bit to single-precision	-
1	0	00	00	100	FCVTAS (scalar) â€” single-precision to 64-bit	-
1	0	00	00	101	FCVTAU (scalar) â€” single-precision to 64-bit	-
1	0	00	01	000	FCVTPS (scalar) â€” single-precision to 64-bit	-
1	0	00	01	001	FCVTPU (scalar) â€” single-precision to 64-bit	-
1	0	00	10	000	FCVTMS (scalar) â€” single-precision to 64-bit	-
1	0	00	10	001	FCVTMU (scalar) â€” single-precision to 64-bit	-
1	0	00	11	000	FCVTZS (scalar, integer) â€” single-precision to 64-bit	-

sf	S	Decode fields		opcode	Instruction Details	Feature
		ftype	rmode			
1	0	00	11	001	FCVTZU (scalar, integer) â€“ single-precision to 64-bit	-
1	0	01	x1	11x	UNALLOCATED	-
1	0	01	00	000	FCVTNS (scalar) â€“ double-precision to 64-bit	-
1	0	01	00	001	FCVTNU (scalar) â€“ double-precision to 64-bit	-
1	0	01	00	010	SCVTF (scalar, integer) â€“ 64-bit to double-precision	-
1	0	01	00	011	UCVTF (scalar, integer) â€“ 64-bit to double-precision	-
1	0	01	00	100	FCVTAS (scalar) â€“ double-precision to 64-bit	-
1	0	01	00	101	FCVTAU (scalar) â€“ double-precision to 64-bit	-
1	0	01	00	110	FMOV (general) â€“ double-precision to 64-bit	-
1	0	01	00	111	FMOV (general) â€“ 64-bit to double-precision	-
1	0	01	01	000	FCVTPS (scalar) â€“ double-precision to 64-bit	-
1	0	01	01	001	FCVTPU (scalar) â€“ double-precision to 64-bit	-
1	0	01	1x	11x	UNALLOCATED	-

sf	S	Decode fields		opcode	Instruction Details	Feature
		fptype	rmode			
1	0	01	10	000	FCVTMS (scalar) â€” double-precision to 64-bit	-
1	0	01	10	001	FCVTMU (scalar) â€” double-precision to 64-bit	-
1	0	01	11	000	FCVTZS (scalar, integer) â€” double-precision to 64-bit	-
1	0	01	11	001	FCVTZU (scalar, integer) â€” double-precision to 64-bit	-
1	0	10	x0	11x	UNALLOCATED	-
1	0	10	01	110	FMOV (general) â€” top half of 128-bit to 64-bit	-
1	0	10	01	111	FMOV (general) â€” 64-bit to top half of 128-bit	-
1	0	10	1x	11x	UNALLOCATED	-
1	0	11	00	000	FCVTNS (scalar) â€” half-precision to 64-bit	FEAT_FP16
1	0	11	00	001	FCVTNU (scalar) â€” half-precision to 64-bit	FEAT_FP16
1	0	11	00	010	SCVTF (scalar, integer) â€” 64-bit to half-precision	FEAT_FP16
1	0	11	00	011	UCVTF (scalar, integer) â€” 64-bit to half-precision	FEAT_FP16
1	0	11	00	100	FCVTAS (scalar) â€” half-precision to 64-bit	FEAT_FP16

sf	S	Decode fields			opcode	Instruction Details	Feature
		ftype	rmode				
1	0	11	00		101	FCVTAU (scalar) â€” half-precision to 64-bit	FEAT_FP16
1	0	11	00		110	FMOV (general) â€” half-precision to 64-bit	FEAT_FP16
1	0	11	00		111	FMOV (general) â€” 64-bit to half-precision	FEAT_FP16
1	0	11	01		000	FCVTPS (scalar) â€” half-precision to 64-bit	FEAT_FP16
1	0	11	01		001	FCVTPU (scalar) â€” half-precision to 64-bit	FEAT_FP16
1	0	11	10		000	FCVTMS (scalar) â€” half-precision to 64-bit	FEAT_FP16
1	0	11	10		001	FCVTMU (scalar) â€” half-precision to 64-bit	FEAT_FP16
1	0	11	11		000	FCVTZS (scalar, integer) â€” half-precision to 64-bit	FEAT_FP16
1	0	11	11		001	FCVTZU (scalar, integer) â€” half-precision to 64-bit	FEAT_FP16

Floating-point data-processing (1 source)

These instructions are under [Data Processing -- Scalar Floating-Point and Advanced SIMD](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
M	0	S	1	1	1	1	0	ftype	1		opcode					1	0	0	0	0	Rn				Rd						

Decode fields				Instruction Details	Feature
M	S	ftype	opcode		
			1xxxxxx	UNALLOCATED	-
	1			UNALLOCATED	-
0	0	00	000000	FMOV (register) â€“ single-precision	-
0	0	00	000001	FABS (scalar) â€“ single-precision	-
0	0	00	000010	FNNEG (scalar) â€“ single-precision	-
0	0	00	000011	FSQRT (scalar) â€“ single-precision	-
0	0	00	000100	UNALLOCATED	-
0	0	00	000101	FCVT â€“ single-precision to double-precision	-
0	0	00	000110	UNALLOCATED	-
0	0	00	000111	FCVT â€“ single-precision to half-precision	-
0	0	00	001000	FRINTN (scalar) â€“ single-precision	-
0	0	00	001001	FRINTP (scalar) â€“ single-precision	-
0	0	00	001010	FRINTM (scalar) â€“ single-precision	-
0	0	00	001011	FRINTZ (scalar) â€“ single-precision	-
0	0	00	001100	FRINTA (scalar) â€“ single-precision	-
0	0	00	001101	UNALLOCATED	-
0	0	00	001110	FRINTX (scalar) â€“ single-precision	-
0	0	00	001111	FRINTI (scalar) â€“ single-precision	-
0	0	00	010000	FRINT32Z (scalar) â€“ single-precision	FEAT_FRINTTS
0	0	00	010001	FRINT32X (scalar) â€“ single-precision	FEAT_FRINTTS
0	0	00	010010	FRINT64Z (scalar) â€“ single-precision	FEAT_FRINTTS
0	0	00	010011	FRINT64X (scalar) â€“ single-precision	FEAT_FRINTTS
0	0	00	0101xx	UNALLOCATED	-
0	0	00	011xxx	UNALLOCATED	-

M	S	Decode fields ftype	opcode	Instruction Details	Feature
0	0	01	000000	FMOV (register) â€“ double-precision	-
0	0	01	000001	FABS (scalar) â€“ double-precision	-
0	0	01	000010	FNNEG (scalar) â€“ double-precision	-
0	0	01	000011	FSQRT (scalar) â€“ double-precision	-
0	0	01	000100	FCVT â€“ double-precision to single-precision	-
0	0	01	000101	UNALLOCATED	-
0	0	01	000110	BFCVT	FEAT_BF16
0	0	01	000111	FCVT â€“ double-precision to half-precision	-
0	0	01	001000	FRINTN (scalar) â€“ double-precision	-
0	0	01	001001	FRINTP (scalar) â€“ double-precision	-
0	0	01	001010	FRINTM (scalar) â€“ double-precision	-
0	0	01	001011	FRINTZ (scalar) â€“ double-precision	-
0	0	01	001100	FRINTA (scalar) â€“ double-precision	-
0	0	01	001101	UNALLOCATED	-
0	0	01	001110	FRINTX (scalar) â€“ double-precision	-
0	0	01	001111	FRINTI (scalar) â€“ double-precision	-
0	0	01	010000	FRINT32Z (scalar) â€“ double-precision	FEAT_FRINTTS
0	0	01	010001	FRINT32X (scalar) â€“ double-precision	FEAT_FRINTTS
0	0	01	010010	FRINT64Z (scalar) â€“ double-precision	FEAT_FRINTTS
0	0	01	010011	FRINT64X (scalar) â€“ double-precision	FEAT_FRINTTS
0	0	01	0101xx	UNALLOCATED	-
0	0	01	011xxx	UNALLOCATED	-
0	0	10	0xxxxx	UNALLOCATED	-
0	0	11	000000	FMOV (register) â€“ half-precision	FEAT_FP16

M	S	Decode fields ftype	opcode	Instruction Details	Feature
0	0	11	000001	FABS (scalar) â€“ half-precision	FEAT_FP16
0	0	11	000010	FNNEG (scalar) â€“ half-precision	FEAT_FP16
0	0	11	000011	FSQRT (scalar) â€“ half-precision	FEAT_FP16
0	0	11	000100	FCVT â€“ half-precision to single-precision	-
0	0	11	000101	FCVT â€“ half-precision to double-precision	-
0	0	11	00011x	UNALLOCATED	-
0	0	11	001000	FRINTN (scalar) â€“ half-precision	FEAT_FP16
0	0	11	001001	FRINTP (scalar) â€“ half-precision	FEAT_FP16
0	0	11	001010	FRINTM (scalar) â€“ half-precision	FEAT_FP16
0	0	11	001011	FRINTZ (scalar) â€“ half-precision	FEAT_FP16
0	0	11	001100	FRINTA (scalar) â€“ half-precision	FEAT_FP16
0	0	11	001101	UNALLOCATED	-
0	0	11	001110	FRINTX (scalar) â€“ half-precision	FEAT_FP16
0	0	11	001111	FRINTI (scalar) â€“ half-precision	FEAT_FP16
0	0	11	01xxxx	UNALLOCATED	-
1				UNALLOCATED	-

Floating-point compare

These instructions are under [Data Processing -- Scalar Floating-Point and Advanced SIMD](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
M	0	S	1	1	1	1	0	ftype	1	Rm	op	1	0	0	0	Rn	opcode2																	

M	S	Decode fields ftype	op	opcode2	Instruction Details	Feature
				xxxx1	UNALLOCATED	-
				xxx1x	UNALLOCATED	-
				xx1xx	UNALLOCATED	-
			x1		UNALLOCATED	-

Decode fields					Instruction Details	Feature
M	S	ftype	op	opcode2		
			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

These instructions are under [Data Processing -- Scalar Floating-Point and Advanced SIMD](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
M	0	S	1	1	1	1	0	ftype	1							imm8			1	0	0		imm5			Rd						

Decode fields				Instruction Details	Feature
M	S	ftype	imm5		
			xxxx1	UNALLOCATED	-
			xxx1x	UNALLOCATED	-
			xx1xx	UNALLOCATED	-
			x1xxx	UNALLOCATED	-
			1xxxx	UNALLOCATED	-
		10		UNALLOCATED	-
		1		UNALLOCATED	-
0	0	00	00000	FMOV (scalar, immediate) â€” single-precision	-
0	0	01	00000	FMOV (scalar, immediate) â€” double-precision	-
0	0	11	00000	FMOV (scalar, immediate) â€” half-precision	FEAT_FP16

Decode fields				Instruction Details				Feature	
M	S	ftype	imm5						
1				UNALLOCATED				-	

Floating-point conditional compare

These instructions are under [Data Processing -- Scalar Floating-Point and Advanced SIMD](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
M	0	S	1	1	1	1	0	ftype	1	Rm		cond	0	1	Rn	op	nzcv														

Decode fields				Instruction Details				Feature	
M	S	ftype	op						
		10		UNALLOCATED				-	
		1		UNALLOCATED				-	
0	0	00	0	FCCMP â€“ single-precision				-	
0	0	00	1	FCCMPE â€“ single-precision				-	
0	0	01	0	FCCMP â€“ double-precision				-	
0	0	01	1	FCCMPE â€“ double-precision				-	
0	0	11	0	FCCMP â€“ half-precision				FEAT_FP16	
0	0	11	1	FCCMPE â€“ half-precision				FEAT_FP16	
1				UNALLOCATED				-	

Floating-point data-processing (2 source)

These instructions are under [Data Processing -- Scalar Floating-Point and Advanced SIMD](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
M	0	S	1	1	1	1	0	ftype	1	Rm		opcode	1	0	Rn		Rd														

Decode fields				Instruction Details				Feature	
M	S	ftype	opcode						
			1xx1	UNALLOCATED				-	
			1x1x	UNALLOCATED				-	
			11xx	UNALLOCATED				-	
		10		UNALLOCATED				-	
	1			UNALLOCATED				-	
0	0	00	0000	FMUL (scalar) â€“ single-precision				-	
0	0	00	0001	FDIV (scalar) â€“ single-precision				-	

M	Decode fields	Instruction Details	Feature	
S	ftype	opcode		
0	0	00	FADD (scalar) â€“ single-precision	-
0	0	00	FSUB (scalar) â€“ single-precision	-
0	0	00	FMAX (scalar) â€“ single-precision	-
0	0	00	FMIN (scalar) â€“ single-precision	-
0	0	00	FMAXNM (scalar) â€“ single-precision	-
0	0	00	FMINNM (scalar) â€“ single-precision	-
0	0	00	FNMUL (scalar) â€“ single-precision	-
0	0	01	FMUL (scalar) â€“ double-precision	-
0	0	01	FDIV (scalar) â€“ double-precision	-
0	0	01	FADD (scalar) â€“ double-precision	-
0	0	01	FSUB (scalar) â€“ double-precision	-
0	0	01	FMAX (scalar) â€“ double-precision	-
0	0	01	FMIN (scalar) â€“ double-precision	-
0	0	01	FMAXNM (scalar) â€“ double-precision	-
0	0	01	FMINNM (scalar) â€“ double-precision	-
0	0	01	FNMUL (scalar) â€“ double-precision	-
0	0	11	FMUL (scalar) â€“ half-precision	FEAT_FP16
0	0	11	FDIV (scalar) â€“ half-precision	FEAT_FP16
0	0	11	FADD (scalar) â€“ half-precision	FEAT_FP16
0	0	11	FSUB (scalar) â€“ half-precision	FEAT_FP16
0	0	11	FMAX (scalar) â€“ half-precision	FEAT_FP16
0	0	11	FMIN (scalar) â€“ half-precision	FEAT_FP16
0	0	11	FMAXNM (scalar) â€“ half-precision	FEAT_FP16

M	S	Decode fields		Instruction Details	Feature
		ftype	opcode		
0	0	11	0111	FMINNM (scalar) â€“ half-precision	FEAT_FP16
0	0	11	1000	FNMUL (scalar) â€“ half-precision	FEAT_FP16
1				UNALLOCATED	-

Floating-point conditional select

These instructions are under [Data Processing -- Scalar Floating-Point and Advanced SIMD](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
M	0	S	1	1	1	1	0	ftype	1	Rm		cond	1	1	Rn		Rd														

M	S	Decode fields		Instruction Details	Feature
		ftype			
		10		UNALLOCATED	-
	1			UNALLOCATED	-
0	0	00		FCSEL â€“ single-precision	-
0	0	01		FCSEL â€“ double-precision	-
0	0	11		FCSEL â€“ half-precision	FEAT_FP16
1				UNALLOCATED	-

Floating-point data-processing (3 source)

These instructions are under [Data Processing -- Scalar Floating-Point and Advanced SIMD](#).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
M	0	S	1	1	1	1	1	ftype	01	Rm	00	Ra		Rn		Rd															

M	S	Decode fields		Instruction Details	Feature
		ftype	o1		
		10		UNALLOCATED	-
	1			UNALLOCATED	-
0	0	00	0	FMADD â€“ single-precision	-
0	0	00	0	FMSUB â€“ single-precision	-
0	0	00	1	FNMADD â€“ single-precision	-

Decode fields					Instruction Details	Feature
M	S	fptype	o1	o0		
0	0	00	1	1	FNMSUB â€“ single-precision	-
0	0	01	0	0	FMADD â€“ double-precision	-
0	0	01	0	1	FMSUB â€“ double-precision	-
0	0	01	1	0	FNMADD â€“ double-precision	-
0	0	01	1	1	FNMSUB â€“ double-precision	-
0	0	11	0	0	FMADD â€“ half-precision	FEAT_FP16
0	0	11	0	1	FMSUB â€“ half-precision	FEAT_FP16
0	0	11	1	0	FNMADD â€“ half-precision	FEAT_FP16
0	0	11	1	1	FNMSUB â€“ half-precision	FEAT_FP16
1					UNALLOCATED	-

[Base Instructions](#)

[SIMD&FP Instructions](#)

[SVE Instructions](#)

[SME Instructions](#)

[Index by Encoding](#)

Internal version only: isa v33.64, AdvSIMD v29.12, pseudocode no_diffs_2023_09_RC2, sve v2023-06_rel ; Build timestamp: 2023-09-18T17:56

Copyright Â© 2010-2023 Arm Limited or its affiliates. All rights reserved. This document is Non-Confidential.

Sh
Pseu