A64 -- SVE Instructions (alphabetic order)

ABS: Absolute value (predicated).

ADCLB: Add with carry long (bottom).

ADCLT: Add with carry long (top).

ADD (immediate): Add immediate (unpredicated).

ADD (vectors, predicated): Add vectors (predicated).

ADD (vectors, unpredicated): Add vectors (unpredicated).

ADDHNB: Add narrow high part (bottom).

ADDHNT: Add narrow high part (top).

ADDP: Add pairwise.

ADDPL: Add multiple of predicate register size to scalar register.

ADDQV: Unsigned add reduction of quadword vector segments.

ADDVL: Add multiple of vector register size to scalar register.

ADR: Compute vector address.

AESD: AES single round decryption.

AESE: AES single round encryption.

AESIMC: AES inverse mix columns.

AESMC: AES mix columns.

AND (immediate): Bitwise AND with immediate (unpredicated).

AND (predicates): Bitwise AND predicates.

AND (vectors, predicated): Bitwise AND vectors (predicated).

AND (vectors, unpredicated): Bitwise AND vectors (unpredicated).

ANDQV: Bitwise AND reduction of quadword vector segments.

ANDS: Bitwise AND predicates, setting the condition flags.

ANDV: Bitwise AND reduction to scalar.

ASR (immediate, predicated): Arithmetic shift right by immediate (predicated).

ASR (immediate, unpredicated): Arithmetic shift right by immediate (unpredicated).

ASR (vectors): Arithmetic shift right by vector (predicated).

ASR (wide elements, predicated): Arithmetic shift right by 64-bit wide elements (predicated).

ASR (wide elements, unpredicated): Arithmetic shift right by 64-bit wide elements (unpredicated).

ASRD: Arithmetic shift right for divide by immediate (predicated).

ASRR: Reversed arithmetic shift right by vector (predicated).

BCAX: Bitwise clear and exclusive OR.

BDEP: Scatter lower bits into positions selected by bitmask.

BEXT: Gather lower bits from positions selected by bitmask.

BFADD (predicated): BFloat16 floating-point add vectors (predicated).

BFADD (unpredicated): BFloat16 floating-point add vectors (unpredicated).

BFCLAMP: BFloat16 floating-point clamp to minimum/maximum number.

BFCVT: Floating-point down convert to BFloat16 format (predicated).

BFCVTNT: Floating-point down convert and narrow to BFloat16 (top, predicated).

BFDOT (indexed): BFloat16 floating-point indexed dot product.

BFDOT (vectors): BFloat16 floating-point dot product.

BFMAX: BFloat16 floating-point maximum (predicated).

BFMAXNM: BFloat16 floating-point maximum number (predicated).

BFMIN: BFloat16 floating-point minimum (predicated).

BFMINNM: BFloat16 floating-point minimum number (predicated).

BFMLA (indexed): BFloat16 floating-point fused multiply-add vectors by indexed elements.

BFMLA (vectors): BFloat16 floating-point fused multiply-add vectors.

BFMLALB (indexed): BFloat16 floating-point multiply-add long to single-precision (bottom, indexed).

BFMLALB (vectors): BFloat16 floating-point multiply-add long to single-precision (bottom).

BFMLALT (indexed): BFloat16 floating-point multiply-add long to single-precision (top, indexed).

BFMLALT (vectors): BFloat16 floating-point multiply-add long to single-precision (top).

BFMLS (indexed): BFloat16 floating-point fused multiply-subtract vectors by indexed elements.

BFMLS (vectors): BFloat16 floating-point fused multiply-subtract vectors.

BFMLSLB (indexed): BFloat16 floating-point multiply-subtract long from single-precision (bottom, indexed).

BFMLSLB (vectors): BFloat16 floating-point multiply-subtract long from single-precision (bottom).

BFMLSLT (indexed): BFloat16 floating-point multiply-subtract long from single-precision (top, indexed).

BFMLSLT (vectors): BFloat16 floating-point multiply-subtract long from single-precision (top).

BFMMLA: BFloat16 floating-point matrix multiply-accumulate into 2Ã—2 matrices.

BFMUL (indexed): BFloat16 floating-point multiply vectors by indexed elements.

BFMUL (vectors, predicated): BFloat16 floating-point multiply vectors (predicated).

BFMUL (vectors, unpredicated): BFloat16 floating-point multiply vectors (unpredicated).

BFSUB (predicated): BFloat16 floating-point subtract vectors (predicated).

BFSUB (unpredicated): BFloat16 floating-point subtract vectors (unpredicated).

BGRP: Group bits to right or left as selected by bitmask.

BIC (immediate): Bitwise clear bits using immediate (unpredicated): an alias of AND (immediate).

BIC (predicates): Bitwise clear predicates.

BIC (vectors, predicated): Bitwise clear vectors (predicated).

BIC (vectors, unpredicated): Bitwise clear vectors (unpredicated).

BICS: Bitwise clear predicates, setting the condition flags.

BRKA: Break after first true condition.

BRKAS: Break after first true condition, setting the condition flags.

BRKB: Break before first true condition.

BRKBS: Break before first true condition, setting the condition flags.

BRKN: Propagate break to next partition.

BRKNS: Propagate break to next partition, setting the condition flags.

BRKPA: Break after first true condition, propagating from previous partition.

BRKPAS: Break after first true condition, propagating from previous partition and setting the condition flags.

BRKPB: Break before first true condition, propagating from previous partition.

BRKPBS: Break before first true condition, propagating from previous partition and setting the condition flags.

BSL: Bitwise select.

BSL1N: Bitwise select with first input inverted.

BSL2N: Bitwise select with second input inverted.

CADD: Complex integer add with rotate.

CDOT (indexed): Complex integer dot product (indexed).

CDOT (vectors): Complex integer dot product.

CLASTA (scalar): Conditionally extract element after last to general-purpose register.

CLASTA (SIMD&FP scalar): Conditionally extract element after last to SIMD&FP scalar register.

CLASTA (vectors): Conditionally extract element after last to vector register.

CLASTB (scalar): Conditionally extract last element to general-purpose register.

CLASTB (SIMD&FP scalar): Conditionally extract last element to SIMD&FP scalar register.

CLASTB (vectors): Conditionally extract last element to vector register.

CLS: Count leading sign bits (predicated).

CLZ: Count leading zero bits (predicated).

CMLA (indexed): Complex integer multiply-add with rotate (indexed).

CMLA (vectors): Complex integer multiply-add with rotate.

CMP<cc> (immediate): Compare vector to immediate.

CMP<cc> (vectors): Compare vectors.

CMP<cc> (wide elements): Compare vector to 64-bit wide elements.

CMPLE (vectors): Compare signed less than or equal to vector, setting the condition flags: an alias of CMP<cc> (vectors).

CMPLO (vectors): Compare unsigned lower than vector, setting the condition flags: an alias of CMP<cc> (vectors).

CMPLS (vectors): Compare unsigned lower or same as vector, setting the condition flags: an alias of CMP<cc> (vectors).

CMPLT (vectors): Compare signed less than vector, setting the condition flags: an alias of CMP<cc> (vectors).

CNOT: Logically invert boolean condition in vector (predicated).

CNT: Count non-zero bits (predicated).

CNTB, CNTD, CNTH, CNTW: Set scalar to multiple of predicate constraint element count.

CNTP (predicate as counter): Set scalar to count from predicate-as-counter.

CNTP (predicate): Set scalar to count of true predicate elements.

COMPACT: Shuffle active elements of vector to the right and fill with zero.

CPY (immediate, merging): Copy signed integer immediate to vector elements (merging).

CPY (immediate, zeroing): Copy signed integer immediate to vector elements (zeroing).

CPY (scalar): Copy general-purpose register to vector elements (predicated).

CPY (SIMD&FP scalar): Copy SIMD&FP scalar register to vector elements (predicated).

CTERMEQ, CTERMNE: Compare and terminate loop.

DECB, DECD, DECH, DECW (scalar): Decrement scalar by multiple of predicate constraint element count.

DECD, DECH, DECW (vector): Decrement vector by multiple of predicate constraint element count.

DECP (scalar): Decrement scalar by count of true predicate elements.

DECP (vector): Decrement vector by count of true predicate elements.

DUP (immediate): Broadcast signed immediate to vector elements (unpredicated).

DUP (indexed): Broadcast indexed element to vector (unpredicated).

DUP (scalar): Broadcast general-purpose register to vector elements (unpredicated).

DUPM: Broadcast logical bitmask immediate to vector (unpredicated).

DUPQ: Broadcast indexed element within each quadword vector segment (unpredicated).

EON: Bitwise exclusive OR with inverted immediate (unpredicated): an alias of EOR (immediate).

EOR (immediate): Bitwise exclusive OR with immediate (unpredicated).

EOR (predicates): Bitwise exclusive OR predicates.

EOR (vectors, predicated): Bitwise exclusive OR vectors (predicated).

EOR (vectors, unpredicated): Bitwise exclusive OR vectors (unpredicated).

EOR3: Bitwise exclusive OR of three vectors.

EORBT: Interleaving exclusive OR (bottom, top).

EORQV: Bitwise exclusive OR reduction of quadword vector segments.

EORS: Bitwise exclusive OR predicates, setting the condition flags.

EORTB: Interleaving exclusive OR (top, bottom).

EORV: Bitwise exclusive OR reduction to scalar.

EXT: Extract vector from pair of vectors.

EXTQ: Extract vector segment from each pair of quadword vector segments.

FABD: Floating-point absolute difference (predicated).

FABS: Floating-point absolute value (predicated).

FAC<cc>: Floating-point absolute compare vectors.

FACLE: Floating-point absolute compare less than or equal: an alias of FAC<cc>.

FACLT: Floating-point absolute compare less than: an alias of FAC<cc>.

FADD (immediate): Floating-point add immediate (predicated).

FADD (vectors, predicated): Floating-point add vector (predicated).

FADD (vectors, unpredicated): Floating-point add vector (unpredicated).

FADDA: Floating-point add strictly-ordered reduction, accumulating in scalar.

FADDP: Floating-point add pairwise.

FADDQV: Floating-point add recursive reduction of quadword vector segments.

FADDV: Floating-point add recursive reduction to scalar.

FCADD: Floating-point complex add with rotate (predicated).

FCLAMP: Floating-point clamp to minimum/maximum number.

FCM<cc> (vectors): Floating-point compare vectors.

FCM<cc> (zero): Floating-point compare vector with zero.

FCMLA (indexed): Floating-point complex multiply-add by indexed values with rotate.

FCMLA (vectors): Floating-point complex multiply-add with rotate (predicated).

FCMLE (vectors): Floating-point compare less than or equal to vector: an alias of FCM<cc> (vectors).

FCMLT (vectors): Floating-point compare less than vector: an alias of FCM<cc> (vectors).

FCPY: Copy 8-bit floating-point immediate to vector elements (predicated).

FCVT: Floating-point convert precision (predicated).

FCVTLT: Floating-point up convert long (top, predicated).

FCVTNT: Floating-point down convert and narrow (top, predicated).

FCVTX: Floating-point down convert, rounding to odd (predicated).

FCVTXNT: Floating-point down convert, rounding to odd (top, predicated).

FCVTZS: Floating-point convert to signed integer, rounding toward zero (predicated).

FCVTZU: Floating-point convert to unsigned integer, rounding toward zero (predicated).

FDIV: Floating-point divide by vector (predicated).

FDIVR: Floating-point reversed divide by vector (predicated).

FDOT (indexed): Half-precision floating-point indexed dot product.

FDOT (vectors): Half-precision floating-point dot product.

FDUP: Broadcast 8-bit floating-point immediate to vector elements (unpredicated).

FEXPA: Floating-point exponential accelerator.

FLOGB: Floating-point base 2 logarithm as integer.

FMAD: Floating-point fused multiply-add vectors (predicated), writing multiplicand [Zdn = Za + Zdn * Zm].

FMAX (immediate): Floating-point maximum with immediate (predicated).

FMAX (vectors): Floating-point maximum (predicated).

FMAXNM (immediate): Floating-point maximum number with immediate (predicated).

FMAXNM (vectors): Floating-point maximum number (predicated).

FMAXNMP: Floating-point maximum number pairwise.

FMAXNMQV: Floating-point maximum number recursive reduction of quadword vector segments.

FMAXNMV: Floating-point maximum number recursive reduction to scalar.

FMAXP: Floating-point maximum pairwise.

FMAXQV: Floating-point maximum reduction of quadword vector segments.

FMAXV: Floating-point maximum recursive reduction to scalar.

FMIN (immediate): Floating-point minimum with immediate (predicated).

FMIN (vectors): Floating-point minimum (predicated).

FMINNM (immediate): Floating-point minimum number with immediate (predicated).

FMINNM (vectors): Floating-point minimum number (predicated).

FMINNMP: Floating-point minimum number pairwise.

FMINNMQV: Floating-point minimum number recursive reduction of quadword vector segments.

FMINNMV: Floating-point minimum number recursive reduction to scalar.

FMINP: Floating-point minimum pairwise.

FMINQV: Floating-point minimum recursive reduction of quadword vector segments.

FMINV: Floating-point minimum recursive reduction to scalar.

FMLA (indexed): Floating-point fused multiply-add by indexed elements (Zda = Zda + Zn * Zm[indexed]).

FMLA (vectors): Floating-point fused multiply-add vectors (predicated), writing addend [Zda = Zda + Zn * Zm].

FMLALB (indexed): Half-precision floating-point multiply-add long to single-precision (bottom, indexed).

FMLALB (vectors): Half-precision floating-point multiply-add long to single-precision (bottom).

FMLALT (indexed): Half-precision floating-point multiply-add long to single-precision (top, indexed).

FMLALT (vectors): Half-precision floating-point multiply-add long to single-precision (top).

FMLS (indexed): Floating-point fused multiply-subtract by indexed elements (Zda = Zda + -Zn * Zm[indexed]).

FMLS (vectors): Floating-point fused multiply-subtract vectors (predicated), writing addend [Zda = Zda + -Zn * Zm].

FMLSLB (indexed): Half-precision floating-point multiply-subtract long from single-precision (bottom, indexed).

FMLSLB (vectors): Half-precision floating-point multiply-subtract long from single-precision (bottom).

FMLSLT (indexed): Half-precision floating-point multiply-subtract long from single-precision (top, indexed).

FMLSLT (vectors): Half-precision floating-point multiply-subtract long from single-precision (top).

FMMLA: Floating-point matrix multiply-accumulate.

FMOV (immediate, predicated): Move 8-bit floating-point immediate to vector elements (predicated): an alias of FCPY.

FMOV (immediate, unpredicated): Move 8-bit floating-point immediate to vector elements (unpredicated): an alias of FDUP.

FMOV (zero, predicated): Move floating-point +0.0 to vector elements (predicated): an alias of CPY (immediate, merging).

FMOV (zero, unpredicated): Move floating-point +0.0 to vector elements (unpredicated): an alias of DUP (immediate).

FMSB: Floating-point fused multiply-subtract vectors (predicated), writing multiplicand [Zdn = Za + -Zdn * Zm].

FMUL (immediate): Floating-point multiply by immediate (predicated).

FMUL (indexed): Floating-point multiply by indexed elements.

FMUL (vectors, predicated): Floating-point multiply vectors (predicated).

FMUL (vectors, unpredicated): Floating-point multiply vectors (unpredicated).

FMULX: Floating-point multiply-extended vectors (predicated).

FNEG: Floating-point negate (predicated).

FNMAD: Floating-point negated fused multiply-add vectors (predicated), writing multiplicand [Zdn = -Za + -Zdn * Zm].

FNMLA: Floating-point negated fused multiply-add vectors (predicated), writing addend [Zda = -Zda + -Zn * Zm].

FNMLS: Floating-point negated fused multiply-subtract vectors (predicated), writing addend [Zda = -Zda + Zn * Zm].

FNMSB: Floating-point negated fused multiply-subtract vectors (predicated), writing multiplicand [Zdn = -Za + Zdn * Zm].

FRECPE: Floating-point reciprocal estimate (unpredicated).

FRECPS: Floating-point reciprocal step (unpredicated).

FRECPX: Floating-point reciprocal exponent (predicated).

FRINT<r>: Floating-point round to integral value (predicated).

FRSQRTE: Floating-point reciprocal square root estimate (unpredicated).

FRSQRTS: Floating-point reciprocal square root step (unpredicated).

FSCALE: Floating-point adjust exponent by vector (predicated).

FSQRT: Floating-point square root (predicated).

FSUB (immediate): Floating-point subtract immediate (predicated).

FSUB (vectors, predicated): Floating-point subtract vectors (predicated).

FSUB (vectors, unpredicated): Floating-point subtract vectors (unpredicated).

FSUBR (immediate): Floating-point reversed subtract from immediate (predicated).

FSUBR (vectors): Floating-point reversed subtract vectors (predicated).

FTMAD: Floating-point trigonometric multiply-add coefficient.

FTSMUL: Floating-point trigonometric starting value.

FTSSEL: Floating-point trigonometric select coefficient.

HISTCNT: Count matching elements in vector.

HISTSEG: Count matching elements in vector segments.

INCB, INCD, INCH, INCW (scalar): Increment scalar by multiple of predicate constraint element count.

INCD, INCH, INCW (vector): Increment vector by multiple of predicate constraint element count.

INCP (scalar): Increment scalar by count of true predicate elements.

INCP (vector): Increment vector by count of true predicate elements.

INDEX (immediate, scalar): Create index starting from immediate and incremented by general-purpose register.

INDEX (immediates): Create index starting from and incremented by immediate.

INDEX (scalar, immediate): Create index starting from general-purpose register and incremented by immediate.

INDEX (scalars): Create index starting from and incremented by general-purpose register.

INSR (scalar): Insert general-purpose register in shifted vector.

INSR (SIMD&FP scalar): Insert SIMD&FP scalar register in shifted vector.

LASTA (scalar): Extract element after last to general-purpose register.

LASTA (SIMD&FP scalar): Extract element after last to SIMD&FP scalar register.

LASTB (scalar): Extract last element to general-purpose register.

LASTB (SIMD&FP scalar): Extract last element to SIMD&FP scalar register.

LD1B (scalar plus immediate, consecutive registers): Contiguous load of bytes to multiple consecutive vectors (immediate index).

LD1B (scalar plus immediate, single register): Contiguous load unsigned bytes to vector (immediate index).

LD1B (scalar plus scalar, consecutive registers): Contiguous load of bytes to multiple consecutive vectors (scalar index).

LD1B (scalar plus scalar, single register): Contiguous load unsigned bytes to vector (scalar index).

LD1B (scalar plus vector): Gather load unsigned bytes to vector (vector index).

LD1B (vector plus immediate): Gather load unsigned bytes to vector (immediate index).

LD1D (scalar plus immediate, consecutive registers): Contiguous load of doublewords to multiple consecutive vectors (immediate index).

LD1D (scalar plus immediate, single register): Contiguous load unsigned doublewords to vector (immediate index).

LD1D (scalar plus scalar, consecutive registers): Contiguous load of doublewords to multiple consecutive vectors (scalar index).

LD1D (scalar plus scalar, single register): Contiguous load unsigned doublewords to vector (scalar index).

LD1D (scalar plus vector): Gather load doublewords to vector (vector index).

LD1D (vector plus immediate): Gather load doublewords to vector (immediate index).

LD1H (scalar plus immediate, consecutive registers): Contiguous load of halfwords to multiple consecutive vectors (immediate index).

LD1H (scalar plus immediate, single register): Contiguous load unsigned halfwords to vector (immediate index).

LD1H (scalar plus scalar, consecutive registers): Contiguous load of halfwords to multiple consecutive vectors (scalar index).

LD1H (scalar plus scalar, single register): Contiguous load unsigned halfwords to vector (scalar index).

LD1H (scalar plus vector): Gather load unsigned halfwords to vector (vector index).

LD1H (vector plus immediate): Gather load unsigned halfwords to vector (immediate index).

LD1Q: Gather load quadwords.

LD1RB: Load and broadcast unsigned byte to vector.

LD1RD: Load and broadcast doubleword to vector.

LD1RH: Load and broadcast unsigned halfword to vector.

LD1ROB (scalar plus immediate): Contiguous load and replicate thirty-two bytes (immediate index).

LD1ROB (scalar plus scalar): Contiguous load and replicate thirty-two bytes (scalar index).

LD1ROD (scalar plus immediate): Contiguous load and replicate four doublewords (immediate index).

LD1ROD (scalar plus scalar): Contiguous load and replicate four doublewords (scalar index).

LD1ROH (scalar plus immediate): Contiguous load and replicate sixteen halfwords (immediate index).

LD1ROH (scalar plus scalar): Contiguous load and replicate sixteen halfwords (scalar index).

LD1ROW (scalar plus immediate): Contiguous load and replicate eight words (immediate index).

LD1ROW (scalar plus scalar): Contiguous load and replicate eight words (scalar index).

LD1RQB (scalar plus immediate): Contiguous load and replicate sixteen bytes (immediate index).

LD1RQB (scalar plus scalar): Contiguous load and replicate sixteen bytes (scalar index).

LD1RQD (scalar plus immediate): Contiguous load and replicate two doublewords (immediate index).

LD1RQD (scalar plus scalar): Contiguous load and replicate two doublewords (scalar index).

LD1RQH (scalar plus immediate): Contiguous load and replicate eight halfwords (immediate index).

LD1RQH (scalar plus scalar): Contiguous load and replicate eight halfwords (scalar index).

LD1RQW (scalar plus immediate): Contiguous load and replicate four words (immediate index).

LD1RQW (scalar plus scalar): Contiguous load and replicate four words (scalar index).

LD1RSB: Load and broadcast signed byte to vector.

LD1RSH: Load and broadcast signed halfword to vector.

LD1RSW: Load and broadcast signed word to vector.

LD1RW: Load and broadcast unsigned word to vector.

LD1SB (scalar plus immediate): Contiguous load signed bytes to vector (immediate index).

LD1SB (scalar plus scalar): Contiguous load signed bytes to vector (scalar index).

LD1SB (scalar plus vector): Gather load signed bytes to vector (vector index).

LD1SB (vector plus immediate): Gather load signed bytes to vector (immediate index).

LD1SH (scalar plus immediate): Contiguous load signed halfwords to vector (immediate index).

LD1SH (scalar plus scalar): Contiguous load signed halfwords to vector (scalar index).

LD1SH (scalar plus vector): Gather load signed halfwords to vector (vector index).

LD1SH (vector plus immediate): Gather load signed halfwords to vector (immediate index).

LD1SW (scalar plus immediate): Contiguous load signed words to vector (immediate index).

LD1SW (scalar plus scalar): Contiguous load signed words to vector (scalar index).

LD1SW (scalar plus vector): Gather load signed words to vector (vector index).

LD1SW (vector plus immediate): Gather load signed words to vector (immediate index).

LD1W (scalar plus immediate, consecutive registers): Contiguous load of words to multiple consecutive vectors (immediate index).

LD1W (scalar plus immediate, single register): Contiguous load unsigned words to vector (immediate index).

LD1W (scalar plus scalar, consecutive registers): Contiguous load of words to multiple consecutive vectors (scalar index).

LD1W (scalar plus scalar, single register): Contiguous load unsigned words to vector (scalar index).

LD1W (scalar plus vector): Gather load unsigned words to vector (vector index).

LD1W (vector plus immediate): Gather load unsigned words to vector (immediate index).

LD2B (scalar plus immediate): Contiguous load two-byte structures to two vectors (immediate index).

LD2B (scalar plus scalar): Contiguous load two-byte structures to two vectors (scalar index).

LD2D (scalar plus immediate): Contiguous load two-doubleword structures to two vectors (immediate index).

LD2D (scalar plus scalar): Contiguous load two-doubleword structures to two vectors (scalar index).

LD2H (scalar plus immediate): Contiguous load two-halfword structures to two vectors (immediate index).

LD2H (scalar plus scalar): Contiguous load two-halfword structures to two vectors (scalar index).

LD2Q (scalar plus immediate): Contiguous load two-quadword structures to two vectors (immediate index).

LD2Q (scalar plus scalar): Contiguous load two-quadword structures to two vectors (scalar index).

LD2W (scalar plus immediate): Contiguous load two-word structures to two vectors (immediate index).

LD2W (scalar plus scalar): Contiguous load two-word structures to two vectors (scalar index).

LD3B (scalar plus immediate): Contiguous load three-byte structures to three vectors (immediate index).

LD3B (scalar plus scalar): Contiguous load three-byte structures to three vectors (scalar index).

LD3D (scalar plus immediate): Contiguous load three-doubleword structures to three vectors (immediate index).

LD3D (scalar plus scalar): Contiguous load three-doubleword structures to three vectors (scalar index).

LD3H (scalar plus immediate): Contiguous load three-halfword structures to three vectors (immediate index).

LD3H (scalar plus scalar): Contiguous load three-halfword structures to three vectors (scalar index).

LD3Q (scalar plus immediate): Contiguous load three-quadword structures to three vectors (immediate index).

LD3Q (scalar plus scalar): Contiguous load three-quadword structures to three vectors (scalar index).

LD3W (scalar plus immediate): Contiguous load three-word structures to three vectors (immediate index).

LD3W (scalar plus scalar): Contiguous load three-word structures to three vectors (scalar index).

LD4B (scalar plus immediate): Contiguous load four-byte structures to four vectors (immediate index).

LD4B (scalar plus scalar): Contiguous load four-byte structures to four vectors (scalar index).

LD4D (scalar plus immediate): Contiguous load four-doubleword structures to four vectors (immediate index).

LD4D (scalar plus scalar): Contiguous load four-doubleword structures to four vectors (scalar index).

LD4H (scalar plus immediate): Contiguous load four-halfword structures to four vectors (immediate index).

LD4H (scalar plus scalar): Contiguous load four-halfword structures to four vectors (scalar index).

LD4Q (scalar plus immediate): Contiguous load four-quadword structures to four vectors (immediate index).

LD4Q (scalar plus scalar): Contiguous load four-quadword structures to four vectors (scalar index).

LD4W (scalar plus immediate): Contiguous load four-word structures to four vectors (immediate index).

LD4W (scalar plus scalar): Contiguous load four-word structures to four vectors (scalar index).

LDFF1B (scalar plus scalar): Contiguous load first-fault unsigned bytes to vector (scalar index).

LDFF1B (scalar plus vector): Gather load first-fault unsigned bytes to vector (vector index).

LDFF1B (vector plus immediate): Gather load first-fault unsigned bytes to vector (immediate index).

LDFF1D (scalar plus scalar): Contiguous load first-fault doublewords to vector (scalar index).

LDFF1D (scalar plus vector): Gather load first-fault doublewords to vector (vector index).

LDFF1D (vector plus immediate): Gather load first-fault doublewords to vector (immediate index).

LDFF1H (scalar plus scalar): Contiguous load first-fault unsigned halfwords to vector (scalar index).

LDFF1H (scalar plus vector): Gather load first-fault unsigned halfwords to vector (vector index).

LDFF1H (vector plus immediate): Gather load first-fault unsigned halfwords to vector (immediate index).

LDFF1SB (scalar plus scalar): Contiguous load first-fault signed bytes to vector (scalar index).

LDFF1SB (scalar plus vector): Gather load first-fault signed bytes to vector (vector index).

LDFF1SB (vector plus immediate): Gather load first-fault signed bytes to vector (immediate index).

LDFF1SH (scalar plus scalar): Contiguous load first-fault signed halfwords to vector (scalar index).

LDFF1SH (scalar plus vector): Gather load first-fault signed halfwords to vector (vector index).

LDFF1SH (vector plus immediate): Gather load first-fault signed halfwords to vector (immediate index).

LDFF1SW (scalar plus scalar): Contiguous load first-fault signed words to vector (scalar index).

LDFF1SW (scalar plus vector): Gather load first-fault signed words to vector (vector index).

LDFF1SW (vector plus immediate): Gather load first-fault signed words to vector (immediate index).

LDFF1W (scalar plus scalar): Contiguous load first-fault unsigned words to vector (scalar index).

LDFF1W (scalar plus vector): Gather load first-fault unsigned words to vector (vector index).

LDFF1W (vector plus immediate): Gather load first-fault unsigned words to vector (immediate index).

LDNF1B: Contiguous load non-fault unsigned bytes to vector (immediate index).

LDNF1D: Contiguous load non-fault doublewords to vector (immediate index).

LDNF1H: Contiguous load non-fault unsigned halfwords to vector (immediate index).

LDNF1SB: Contiguous load non-fault signed bytes to vector (immediate index).

LDNF1SH: Contiguous load non-fault signed halfwords to vector (immediate index).

LDNF1SW: Contiguous load non-fault signed words to vector (immediate index).

LDNF1W: Contiguous load non-fault unsigned words to vector (immediate index).

LDNT1B (scalar plus immediate, consecutive registers): Contiguous load non-temporal of bytes to multiple consecutive vectors (immediate index).

LDNT1B (scalar plus immediate, single register): Contiguous load non-temporal bytes to vector (immediate index).

LDNT1B (scalar plus scalar, consecutive registers): Contiguous load non-temporal of bytes to multiple consecutive vectors (scalar index).

LDNT1B (scalar plus scalar, single register): Contiguous load non-temporal bytes to vector (scalar index).

LDNT1B (vector plus scalar): Gather load non-temporal unsigned bytes.

LDNT1D (scalar plus immediate, consecutive registers): Contiguous load non-temporal of doublewords to multiple consecutive vectors (immediate index).

LDNT1D (scalar plus immediate, single register): Contiguous load non-temporal doublewords to vector (immediate index).

LDNT1D (scalar plus scalar, consecutive registers): Contiguous load non-temporal of doublewords to multiple consecutive vectors (scalar index).

LDNT1D (scalar plus scalar, single register): Contiguous load non-temporal doublewords to vector (scalar index).

LDNT1D (vector plus scalar): Gather load non-temporal unsigned doublewords.

LDNT1H (scalar plus immediate, consecutive registers): Contiguous load non-temporal of halfwords to multiple consecutive vectors (immediate index).

LDNT1H (scalar plus immediate, single register): Contiguous load non-temporal halfwords to vector (immediate index).

LDNT1H (scalar plus scalar, consecutive registers): Contiguous load non-temporal of halfwords to multiple consecutive vectors (scalar index).

LDNT1H (scalar plus scalar, single register): Contiguous load non-temporal halfwords to vector (scalar index).

LDNT1H (vector plus scalar): Gather load non-temporal unsigned halfwords.

LDNT1SB: Gather load non-temporal signed bytes.

LDNT1SH: Gather load non-temporal signed halfwords.

LDNT1SW: Gather load non-temporal signed words.

LDNT1W (scalar plus immediate, consecutive registers): Contiguous load non-temporal of words to multiple consecutive vectors (immediate index).

LDNT1W (scalar plus immediate, single register): Contiguous load non-temporal words to vector (immediate index).

LDNT1W (scalar plus scalar, consecutive registers): Contiguous load non-temporal of words to multiple consecutive vectors (scalar index).

LDNT1W (scalar plus scalar, single register): Contiguous load non-temporal words to vector (scalar index).

LDNT1W (vector plus scalar): Gather load non-temporal unsigned words.

LDR (predicate): Load predicate register.

LDR (vector): Load vector register.

LSL (immediate, predicated): Logical shift left by immediate (predicated).

LSL (immediate, unpredicated): Logical shift left by immediate (unpredicated).

LSL (vectors): Logical shift left by vector (predicated).

LSL (wide elements, predicated): Logical shift left by 64-bit wide elements (predicated).

LSL (wide elements, unpredicated): Logical shift left by 64-bit wide elements (unpredicated).

LSLR: Reversed logical shift left by vector (predicated).

LSR (immediate, predicated): Logical shift right by immediate (predicated).

LSR (immediate, unpredicated): Logical shift right by immediate (unpredicated).

LSR (vectors): Logical shift right by vector (predicated).

LSR (wide elements, predicated): Logical shift right by 64-bit wide elements (predicated).

LSR (wide elements, unpredicated): Logical shift right by 64-bit wide elements (unpredicated).

LSRR: Reversed logical shift right by vector (predicated).

MAD: Multiply-add vectors (predicated), writing multiplicand [Zdn = Za + Zdn * Zm].

MATCH: Detect any matching elements, setting the condition flags.

MLA (indexed): Multiply-add to accumulator (indexed).

MLA (vectors): Multiply-add vectors (predicated), writing addend [Zda = Zda + Zn * Zm].

MLS (indexed): Multiply-subtract from accumulator (indexed).

MLS (vectors): Multiply-subtract vectors (predicated), writing addend [Zda = Zda - Zn * Zm].

MOV: Move logical bitmask immediate to vector (unpredicated): an alias of DUPM.

MOV: Move predicate (unpredicated): an alias of ORR (predicates).

MOV (immediate, predicated, merging): Move signed integer immediate to vector elements (merging): an alias of CPY (immediate, merging).

MOV (immediate, predicated, zeroing): Move signed integer immediate to vector elements (zeroing): an alias of CPY (immediate, zeroing).

MOV (immediate, unpredicated): Move signed immediate to vector elements (unpredicated): an alias of DUP (immediate).

MOV (predicate, predicated, merging): Move predicates (merging): an alias of SEL (predicates).

MOV (predicate, predicated, zeroing): Move predicates (zeroing): an alias of AND (predicates).

MOV (scalar, predicated): Move general-purpose register to vector elements (predicated): an alias of CPY (scalar).

MOV (scalar, unpredicated): Move general-purpose register to vector elements (unpredicated): an alias of DUP (scalar).

MOV (SIMD&FP scalar, predicated): Move SIMD&FP scalar register to vector elements (predicated): an alias of CPY (SIMD&FP scalar).

MOV (SIMD&FP scalar, unpredicated): Move indexed element or SIMD&FP scalar to vector (unpredicated): an alias of DUP (indexed).

MOV (vector, predicated): Move vector elements (predicated): an alias of SEL (vectors).

MOV (vector, unpredicated): Move vector register (unpredicated): an alias of ORR (vectors, unpredicated).

MOVPRFX (predicated): Move prefix (predicated).

MOVPRFX (unpredicated): Move prefix (unpredicated).

MOVS (predicated): Move predicates (zeroing), setting the condition flags: an alias of ANDS.

MOVS (unpredicated): Move predicate (unpredicated), setting the condition flags: an alias of ORRS.

MSB: Multiply-subtract vectors (predicated), writing multiplicand [Zdn = Za - Zdn * Zm].

MUL (immediate): Multiply by immediate (unpredicated).

MUL (indexed): Multiply (indexed).

MUL (vectors, predicated): Multiply vectors (predicated).

MUL (vectors, unpredicated): Multiply vectors (unpredicated).

NAND: Bitwise NAND predicates.

NANDS: Bitwise NAND predicates, setting the condition flags.

NBSL: Bitwise inverted select.

NEG: Negate (predicated).

NMATCH: Detect no matching elements, setting the condition flags.

NOR: Bitwise NOR predicates.

NORS: Bitwise NOR predicates, setting the condition flags.

NOT (predicate): Bitwise invert predicate: an alias of EOR (predicates).

NOT (vector): Bitwise invert vector (predicated).

NOTS: Bitwise invert predicate, setting the condition flags: an alias of EORS.

ORN (immediate): Bitwise inclusive OR with inverted immediate (unpredicated): an alias of ORR (immediate).

ORN (predicates): Bitwise inclusive OR inverted predicate.

ORNS: Bitwise inclusive OR inverted predicate, setting the condition flags.

ORQV: Bitwise inclusive OR reduction of quadword vector segments.

ORR (immediate): Bitwise inclusive OR with immediate (unpredicated).

ORR (predicates): Bitwise inclusive OR predicates.

ORR (vectors, predicated): Bitwise inclusive OR vectors (predicated).

ORR (vectors, unpredicated): Bitwise inclusive OR vectors (unpredicated).

ORRS: Bitwise inclusive OR predicates, setting the condition flags.

ORV: Bitwise inclusive OR reduction to scalar.

PEXT (predicate pair): Set pair of predicates from predicate-as-counter.

PEXT (predicate): Set predicate from predicate-as-counter.

PFALSE: Set all predicate elements to false.

PFIRST: Set the first active predicate element to true.

PMOV (to predicate): Move predicate from vector.

PMOV (to vector): Move predicate to vector.

PMUL: Polynomial multiply vectors (unpredicated).

PMULLB: Polynomial multiply long (bottom).

PMULLT: Polynomial multiply long (top).

PNEXT: Find next active predicate.

PRFB (scalar plus immediate): Contiguous prefetch bytes (immediate index).

PRFB (scalar plus scalar): Contiguous prefetch bytes (scalar index).

PRFB (scalar plus vector): Gather prefetch bytes (scalar plus vector).

PRFB (vector plus immediate): Gather prefetch bytes (vector plus immediate).

PRFD (scalar plus immediate): Contiguous prefetch doublewords (immediate index).

PRFD (scalar plus scalar): Contiguous prefetch doublewords (scalar index).

PRFD (scalar plus vector): Gather prefetch doublewords (scalar plus vector).

PRFD (vector plus immediate): Gather prefetch doublewords (vector plus immediate).

PRFH (scalar plus immediate): Contiguous prefetch halfwords (immediate index).

PRFH (scalar plus scalar): Contiguous prefetch halfwords (scalar index).

PRFH (scalar plus vector): Gather prefetch halfwords (scalar plus vector).

PRFH (vector plus immediate): Gather prefetch halfwords (vector plus immediate).

PRFW (scalar plus immediate): Contiguous prefetch words (immediate index).

PRFW (scalar plus scalar): Contiguous prefetch words (scalar index).

PRFW (scalar plus vector): Gather prefetch words (scalar plus vector).

PRFW (vector plus immediate): Gather prefetch words (vector plus immediate).

PSEL: Predicate select between predicate register or all-false.

PTEST: Set condition flags for predicate.

PTRUE (predicate as counter): Initialise predicate-as-counter to all active.

PTRUE (predicate): Initialise predicate from named constraint.

PTRUES: Initialise predicate from named constraint and set the condition flags.

PUNPKHI, PUNPKLO: Unpack and widen half of predicate.

RADDHNB: Rounding add narrow high part (bottom).

RADDHNT: Rounding add narrow high part (top).

RAX1: Bitwise rotate left by 1 and exclusive OR.

RBIT: Reverse bits (predicated).

RDFFR (predicated): Return predicate of succesfully loaded elements.

RDFFR (unpredicated): Read the first-fault register.

RDFFRS: Return predicate of succesfully loaded elements, setting the condition flags.

RDVL: Read multiple of vector register size to scalar register.

REV (predicate): Reverse all elements in a predicate.

REV (vector): Reverse all elements in a vector (unpredicated).

REVB, REVH, REVW: Reverse bytes / halfwords / words within elements (predicated).

REVD: Reverse 64-bit doublewords in elements (predicated).

RSHRNB: Rounding shift right narrow by immediate (bottom).

RSHRNT: Rounding shift right narrow by immediate (top).

RSUBHNB: Rounding subtract narrow high part (bottom).

RSUBHNT: Rounding subtract narrow high part (top).

SABA: Signed absolute difference and accumulate.

SABALB: Signed absolute difference and accumulate long (bottom).

SABALT: Signed absolute difference and accumulate long (top).

SABD: Signed absolute difference (predicated).

SABDLB: Signed absolute difference long (bottom).

SABDLT: Signed absolute difference long (top).

SADALP: Signed add and accumulate long pairwise.

SADDLB: Signed add long (bottom).

SADDLBT: Signed add long (bottom + top).

SADDLT: Signed add long (top).

SADDV: Signed add reduction to scalar.

SADDWB: Signed add wide (bottom).

SADDWT: Signed add wide (top).

SBCLB: Subtract with carry long (bottom).

SBCLT: Subtract with carry long (top).

SCLAMP: Signed clamp to minimum/maximum vector.

SCVTF: Signed integer convert to floating-point (predicated).

SDIV: Signed divide (predicated).

SDIVR: Signed reversed divide (predicated).

SDOT (2-way, indexed): Signed integer indexed dot product.

SDOT (2-way, vectors): Signed integer dot product.

SDOT (4-way, indexed): Signed integer indexed dot product.

SDOT (4-way, vectors): Signed integer dot product.

SEL (predicates): Conditionally select elements from two predicates.

SEL (vectors): Conditionally select elements from two vectors.

SETFFR: Initialise the first-fault register to all true.

SHADD: Signed halving addition.

SHRNB: Shift right narrow by immediate (bottom).

SHRNT: Shift right narrow by immediate (top).

SHSUB: Signed halving subtract.

SHSUBR: Signed halving subtract reversed vectors.

SLI: Shift left and insert (immediate).

SM4E: SM4 encryption and decryption.

SM4EKEY: SM4 key updates.

SMAX (immediate): Signed maximum with immediate (unpredicated).

SMAX (vectors): Signed maximum vectors (predicated).

SMAXP: Signed maximum pairwise.

SMAXQV: Signed maximum reduction of quadword vector segments.

SMAXV: Signed maximum reduction to scalar.

SMIN (immediate): Signed minimum with immediate (unpredicated).

SMIN (vectors): Signed minimum vectors (predicated).

SMINP: Signed minimum pairwise.

SMINQV: Signed minimum reduction of quadword vector segments.

SMINV: Signed minimum reduction to scalar.

SMLALB (indexed): Signed multiply-add long to accumulator (bottom, indexed).

SMLALB (vectors): Signed multiply-add long to accumulator (bottom).

SMLALT (indexed): Signed multiply-add long to accumulator (top, indexed).

SMLALT (vectors): Signed multiply-add long to accumulator (top).

SMLSLB (indexed): Signed multiply-subtract long from accumulator (bottom, indexed).

SMLSLB (vectors): Signed multiply-subtract long from accumulator (bottom).

SMLSLT (indexed): Signed multiply-subtract long from accumulator (top, indexed).

SMLSLT (vectors): Signed multiply-subtract long from accumulator (top).

SMMLA: Signed integer matrix multiply-accumulate.

SMULH (predicated): Signed multiply returning high half (predicated).

SMULH (unpredicated): Signed multiply returning high half (unpredicated).

SMULLB (indexed): Signed multiply long (bottom, indexed).

SMULLB (vectors): Signed multiply long (bottom).

SMULLT (indexed): Signed multiply long (top, indexed).

SMULLT (vectors): Signed multiply long (top).

SPLICE: Splice two vectors under predicate control.

SQABS: Signed saturating absolute value.

SQADD (immediate): Signed saturating add immediate (unpredicated).

SQADD (vectors, predicated): Signed saturating addition (predicated).

SQADD (vectors, unpredicated): Signed saturating add vectors (unpredicated).

SQCADD: Saturating complex integer add with rotate.

SQCVTN: Signed saturating extract narrow and interleave.

SQCVTUN: Signed saturating unsigned extract narrow and interleave.

SQDECB: Signed saturating decrement scalar by multiple of 8-bit predicate constraint element count.

SQDECD (scalar): Signed saturating decrement scalar by multiple of 64-bit predicate constraint element count.

SQDECD (vector): Signed saturating decrement vector by multiple of 64-bit predicate constraint element count.

SQDECH (scalar): Signed saturating decrement scalar by multiple of 16-bit predicate constraint element count.

SQDECH (vector): Signed saturating decrement vector by multiple of 16-bit predicate constraint element count.

SQDECP (scalar): Signed saturating decrement scalar by count of true predicate elements.

SQDECP (vector): Signed saturating decrement vector by count of true predicate elements.

SQDECW (scalar): Signed saturating decrement scalar by multiple of 32-bit predicate constraint element count.

SQDECW (vector): Signed saturating decrement vector by multiple of 32-bit predicate constraint element count.

SQDMLALB (indexed): Signed saturating doubling multiply-add long to accumulator (bottom, indexed).

SQDMLALB (vectors): Signed saturating doubling multiply-add long to accumulator (bottom).

SQDMLALBT: Signed saturating doubling multiply-add long to accumulator (bottom Ã— top).

SQDMLALT (indexed): Signed saturating doubling multiply-add long to accumulator (top, indexed).

SQDMLALT (vectors): Signed saturating doubling multiply-add long to accumulator (top).

SQDMLSLB (indexed): Signed saturating doubling multiply-subtract long from accumulator (bottom, indexed).

SQDMLSLB (vectors): Signed saturating doubling multiply-subtract long from accumulator (bottom).

SQDMLSLBT: Signed saturating doubling multiply-subtract long from accumulator (bottom Ã— top).

SQDMLSLT (indexed): Signed saturating doubling multiply-subtract long from accumulator (top, indexed).

SQDMLSLT (vectors): Signed saturating doubling multiply-subtract long from accumulator (top).

SQDMULH (indexed): Signed saturating doubling multiply high (indexed).

SQDMULH (vectors): Signed saturating doubling multiply high (unpredicated).

SQDMULLB (indexed): Signed saturating doubling multiply long (bottom, indexed).

SQDMULLB (vectors): Signed saturating doubling multiply long (bottom).

SQDMULLT (indexed): Signed saturating doubling multiply long (top, indexed).

SQDMULLT (vectors): Signed saturating doubling multiply long (top).

SQINCB: Signed saturating increment scalar by multiple of 8-bit predicate constraint element count.

SQINCD (scalar): Signed saturating increment scalar by multiple of 64-bit predicate constraint element count.

SQINCD (vector): Signed saturating increment vector by multiple of 64-bit predicate constraint element count.

SQINCH (scalar): Signed saturating increment scalar by multiple of 16-bit predicate constraint element count.

SQINCH (vector): Signed saturating increment vector by multiple of 16-bit predicate constraint element count.

SQINCP (scalar): Signed saturating increment scalar by count of true predicate elements.

SQINCP (vector): Signed saturating increment vector by count of true predicate elements.

SQINCW (scalar): Signed saturating increment scalar by multiple of 32-bit predicate constraint element count.

SQINCW (vector): Signed saturating increment vector by multiple of 32-bit predicate constraint element count.

SQNEG: Signed saturating negate.

SQRDCMLAH (indexed): Saturating rounding doubling complex integer multiply-add high with rotate (indexed).

SQRDCMLAH (vectors): Saturating rounding doubling complex integer multiply-add high with rotate.

SQRDMLAH (indexed): Signed saturating rounding doubling multiply-add high to accumulator (indexed).

SQRDMLAH (vectors): Signed saturating rounding doubling multiply-add high to accumulator (unpredicated).

SQRDMLSH (indexed): Signed saturating rounding doubling multiply-subtract high from accumulator (indexed).

SQRDMLSH (vectors): Signed saturating rounding doubling multiply-subtract high from accumulator (unpredicated).

SQRDMULH (indexed): Signed saturating rounding doubling multiply high (indexed).

SQRDMULH (vectors): Signed saturating rounding doubling multiply high (unpredicated).

SQRSHL: Signed saturating rounding shift left by vector (predicated).

SQRSHLR: Signed saturating rounding shift left reversed vectors (predicated).

SQRSHRN: Signed saturating rounding shift right narrow by immediate and interleave.

SQRSHRNB: Signed saturating rounding shift right narrow by immediate (bottom).

SQRSHRNT: Signed saturating rounding shift right narrow by immediate (top).

SQRSHRUN: Signed saturating rounding shift right unsigned narrow by immediate and interleave.

SQRSHRUNB: Signed saturating rounding shift right unsigned narrow by immediate (bottom).

SQRSHRUNT: Signed saturating rounding shift right unsigned narrow by immediate (top).

SQSHL (immediate): Signed saturating shift left by immediate.

SQSHL (vectors): Signed saturating shift left by vector (predicated).

SQSHLR: Signed saturating shift left reversed vectors (predicated).

SQSHLU: Signed saturating shift left unsigned by immediate.

SQSHRNB: Signed saturating shift right narrow by immediate (bottom).

SQSHRNT: Signed saturating shift right narrow by immediate (top).

SQSHRUNB: Signed saturating shift right unsigned narrow by immediate (bottom).

SQSHRUNT: Signed saturating shift right unsigned narrow by immediate (top).

SQSUB (immediate): Signed saturating subtract immediate (unpredicated).

SQSUB (vectors, predicated): Signed saturating subtraction (predicated).

SQSUB (vectors, unpredicated): Signed saturating subtract vectors (unpredicated).

SQSUBR: Signed saturating subtraction reversed vectors (predicated).

SQXTNB: Signed saturating extract narrow (bottom).

SQXTNT: Signed saturating extract narrow (top).

SQXTUNB: Signed saturating unsigned extract narrow (bottom).

SQXTUNT: Signed saturating unsigned extract narrow (top).

SRHADD: Signed rounding halving addition.

SRI: Shift right and insert (immediate).

SRSHL: Signed rounding shift left by vector (predicated).

SRSHLR: Signed rounding shift left reversed vectors (predicated).

SRSHR: Signed rounding shift right by immediate.

SRSRA: Signed rounding shift right and accumulate (immediate).

SSHLLB: Signed shift left long by immediate (bottom).

SSHLLT: Signed shift left long by immediate (top).

SSRA: Signed shift right and accumulate (immediate).

SSUBLB: Signed subtract long (bottom).

SSUBLBT: Signed subtract long (bottom - top).

SSUBLT: Signed subtract long (top).

SSUBLTB: Signed subtract long (top - bottom).

SSUBWB: Signed subtract wide (bottom).

SSUBWT: Signed subtract wide (top).

ST1B (scalar plus immediate, consecutive registers): Contiguous store of bytes from multiple consecutive vectors (immediate index).

ST1B (scalar plus immediate, single register): Contiguous store bytes from vector (immediate index).

ST1B (scalar plus scalar, consecutive registers): Contiguous store of bytes from multiple consecutive vectors (scalar index).

ST1B (scalar plus scalar, single register): Contiguous store bytes from vector (scalar index).

ST1B (scalar plus vector): Scatter store bytes from a vector (vector index).

ST1B (vector plus immediate): Scatter store bytes from a vector (immediate index).

ST1D (scalar plus immediate, consecutive registers): Contiguous store of doublewords from multiple consecutive vectors (immediate index).

ST1D (scalar plus immediate, single register): Contiguous store doublewords from vector (immediate index).

ST1D (scalar plus scalar, consecutive registers): Contiguous store of doublewords from multiple consecutive vectors (scalar index).

ST1D (scalar plus scalar, single register): Contiguous store doublewords from vector (scalar index).

ST1D (scalar plus vector): Scatter store doublewords from a vector (vector index).

ST1D (vector plus immediate): Scatter store doublewords from a vector (immediate index).

ST1H (scalar plus immediate, consecutive registers): Contiguous store of halfwords from multiple consecutive vectors (immediate index).

ST1H (scalar plus immediate, single register): Contiguous store halfwords from vector (immediate index).

ST1H (scalar plus scalar, consecutive registers): Contiguous store of halfwords from multiple consecutive vectors (scalar index).

ST1H (scalar plus scalar, single register): Contiguous store halfwords from vector (scalar index).

ST1H (scalar plus vector): Scatter store halfwords from a vector (vector index).

ST1H (vector plus immediate): Scatter store halfwords from a vector (immediate index).

ST1Q: Scatter store quadwords.

ST1W (scalar plus immediate, consecutive registers): Contiguous store of words from multiple consecutive vectors (immediate index).

ST1W (scalar plus immediate, single register): Contiguous store words from vector (immediate index).

ST1W (scalar plus scalar, consecutive registers): Contiguous store of words from multiple consecutive vectors (scalar index).

ST1W (scalar plus scalar, single register): Contiguous store words from vector (scalar index).

ST1W (scalar plus vector): Scatter store words from a vector (vector index).

ST1W (vector plus immediate): Scatter store words from a vector (immediate index).

ST2B (scalar plus immediate): Contiguous store two-byte structures from two vectors (immediate index).

ST2B (scalar plus scalar): Contiguous store two-byte structures from two vectors (scalar index).

ST2D (scalar plus immediate): Contiguous store two-doubleword structures from two vectors (immediate index).

ST2D (scalar plus scalar): Contiguous store two-doubleword structures from two vectors (scalar index).

ST2H (scalar plus immediate): Contiguous store two-halfword structures from two vectors (immediate index).

ST2H (scalar plus scalar): Contiguous store two-halfword structures from two vectors (scalar index).

ST2Q (scalar plus immediate): Contiguous store two-quadword structures from two vectors (immediate index).

ST2Q (scalar plus scalar): Contiguous store two-quadword structures from two vectors (scalar index).

ST2W (scalar plus immediate): Contiguous store two-word structures from two vectors (immediate index).

ST2W (scalar plus scalar): Contiguous store two-word structures from two vectors (scalar index).

ST3B (scalar plus immediate): Contiguous store three-byte structures from three vectors (immediate index).

ST3B (scalar plus scalar): Contiguous store three-byte structures from three vectors (scalar index).

ST3D (scalar plus immediate): Contiguous store three-doubleword structures from three vectors (immediate index).

ST3D (scalar plus scalar): Contiguous store three-doubleword structures from three vectors (scalar index).

ST3H (scalar plus immediate): Contiguous store three-halfword structures from three vectors (immediate index).

ST3H (scalar plus scalar): Contiguous store three-halfword structures from three vectors (scalar index).

ST3Q (scalar plus immediate): Contiguous store three-quadword structures from three vectors (immediate index).

ST3Q (scalar plus scalar): Contiguous store three-quadword structures from three vectors (scalar index).

ST3W (scalar plus immediate): Contiguous store three-word structures from three vectors (immediate index).

ST3W (scalar plus scalar): Contiguous store three-word structures from three vectors (scalar index).

ST4B (scalar plus immediate): Contiguous store four-byte structures from four vectors (immediate index).

ST4B (scalar plus scalar): Contiguous store four-byte structures from four vectors (scalar index).

ST4D (scalar plus immediate): Contiguous store four-doubleword structures from four vectors (immediate index).

ST4D (scalar plus scalar): Contiguous store four-doubleword structures from four vectors (scalar index).

ST4H (scalar plus immediate): Contiguous store four-halfword structures from four vectors (immediate index).

ST4H (scalar plus scalar): Contiguous store four-halfword structures from four vectors (scalar index).

ST4Q (scalar plus immediate): Contiguous store four-quadword structures from four vectors (immediate index).

ST4Q (scalar plus scalar): Contiguous store four-quadword structures from four vectors (scalar index).

ST4W (scalar plus immediate): Contiguous store four-word structures from four vectors (immediate index).

ST4W (scalar plus scalar): Contiguous store four-word structures from four vectors (scalar index).

STNT1B (scalar plus immediate, consecutive registers): Contiguous store non-temporal of bytes from multiple consecutive vectors (immediate index).

STNT1B (scalar plus immediate, single register): Contiguous store non-temporal bytes from vector (immediate index).

STNT1B (scalar plus scalar, consecutive registers): Contiguous store non-temporal of bytes from multiple consecutive vectors (scalar index).

STNT1B (scalar plus scalar, single register): Contiguous store non-temporal bytes from vector (scalar index).

STNT1B (vector plus scalar): Scatter store non-temporal bytes.

STNT1D (scalar plus immediate, consecutive registers): Contiguous store non-temporal of doublewords from multiple consecutive vectors (immediate index).

STNT1D (scalar plus immediate, single register): Contiguous store non-temporal doublewords from vector (immediate index).

STNT1D (scalar plus scalar, consecutive registers): Contiguous store non-temporal of doublewords from multiple consecutive vectors (scalar index).

STNT1D (scalar plus scalar, single register): Contiguous store non-temporal doublewords from vector (scalar index).

STNT1D (vector plus scalar): Scatter store non-temporal doublewords.

STNT1H (scalar plus immediate, consecutive registers): Contiguous store non-temporal of halfwords from multiple consecutive vectors (immediate index).

STNT1H (scalar plus immediate, single register): Contiguous store non-temporal halfwords from vector (immediate index).

STNT1H (scalar plus scalar, consecutive registers): Contiguous store non-temporal of halfwords from multiple consecutive vectors (scalar index).

STNT1H (scalar plus scalar, single register): Contiguous store non-temporal halfwords from vector (scalar index).

STNT1H (vector plus scalar): Scatter store non-temporal halfwords.

STNT1W (scalar plus immediate, consecutive registers): Contiguous store non-temporal of words from multiple consecutive vectors (immediate index).

STNT1W (scalar plus immediate, single register): Contiguous store non-temporal words from vector (immediate index).

STNT1W (scalar plus scalar, consecutive registers): Contiguous store non-temporal of words from multiple consecutive vectors (scalar index).

STNT1W (scalar plus scalar, single register): Contiguous store non-temporal words from vector (scalar index).

STNT1W (vector plus scalar): Scatter store non-temporal words.

STR (predicate): Store predicate register.

STR (vector): Store vector register.

SUB (immediate): Subtract immediate (unpredicated).

SUB (vectors, predicated): Subtract vectors (predicated).

SUB (vectors, unpredicated): Subtract vectors (unpredicated).

SUBHNB: Subtract narrow high part (bottom).

SUBHNT: Subtract narrow high part (top).

SUBR (immediate): Reversed subtract from immediate (unpredicated).

SUBR (vectors): Reversed subtract vectors (predicated).

SUDOT: Signed by unsigned integer indexed dot product.

SUNPKHI, SUNPKLO: Signed unpack and extend half of vector.

SUQADD: Signed saturating addition of unsigned value.

SXTB, SXTH, SXTW: Signed byte / halfword / word extend (predicated).

TBL: Programmable table lookup in one or two vector table (zeroing).

TBLQ: Programmable table lookup within each quadword vector segment (zeroing).

TBX: Programmable table lookup in single vector table (merging).

TBXQ: Programmable table lookup within each quadword vector segment (merging).

TRN1, TRN2 (predicates): Interleave even or odd elements from two predicates.

TRN1, TRN2 (vectors): Interleave even or odd elements from two vectors.

UABA: Unsigned absolute difference and accumulate.

UABALB: Unsigned absolute difference and accumulate long (bottom).

UABALT: Unsigned absolute difference and accumulate long (top).

UABD: Unsigned absolute difference (predicated).

UABDLB: Unsigned absolute difference long (bottom).

UABDLT: Unsigned absolute difference long (top).

UADALP: Unsigned add and accumulate long pairwise.

UADDLB: Unsigned add long (bottom).

UADDLT: Unsigned add long (top).

UADDV: Unsigned add reduction to scalar.

UADDWB: Unsigned add wide (bottom).

UADDWT: Unsigned add wide (top).

UCLAMP: Unsigned clamp to minimum/maximum vector.

UCVTF: Unsigned integer convert to floating-point (predicated).

UDIV: Unsigned divide (predicated).

UDIVR: Unsigned reversed divide (predicated).

UDOT (2-way, indexed): Unsigned integer indexed dot product.

UDOT (2-way, vectors): Unsigned integer dot product.

UDOT (4-way, indexed): Unsigned integer indexed dot product.

UDOT (4-way, vectors): Unsigned integer dot product.

UHADD: Unsigned halving addition.

UHSUB: Unsigned halving subtract.

UHSUBR: Unsigned halving subtract reversed vectors.

UMAX (immediate): Unsigned maximum with immediate (unpredicated).

UMAX (vectors): Unsigned maximum vectors (predicated).

UMAXP: Unsigned maximum pairwise.

UMAXQV: Unsigned maximum reduction of quadword vector segments.

UMAXV: Unsigned maximum reduction to scalar.

UMIN (immediate): Unsigned minimum with immediate (unpredicated).

UMIN (vectors): Unsigned minimum vectors (predicated).

UMINP: Unsigned minimum pairwise.

UMINQV: Unsigned minimum reduction of quadword vector segments.

UMINV: Unsigned minimum reduction to scalar.

UMLALB (indexed): Unsigned multiply-add long to accumulator (bottom, indexed).

UMLALB (vectors): Unsigned multiply-add long to accumulator (bottom).

UMLALT (indexed): Unsigned multiply-add long to accumulator (top, indexed).

UMLALT (vectors): Unsigned multiply-add long to accumulator (top).

UMLSLB (indexed): Unsigned multiply-subtract long from accumulator (bottom, indexed).

UMLSLB (vectors): Unsigned multiply-subtract long from accumulator (bottom).

UMLSLT (indexed): Unsigned multiply-subtract long from accumulator (top, indexed).

UMLSLT (vectors): Unsigned multiply-subtract long from accumulator (top).

UMMLA: Unsigned integer matrix multiply-accumulate.

UMULH (predicated): Unsigned multiply returning high half (predicated).

UMULH (unpredicated): Unsigned multiply returning high half (unpredicated).

UMULLB (indexed): Unsigned multiply long (bottom, indexed).

UMULLB (vectors): Unsigned multiply long (bottom).

UMULLT (indexed): Unsigned multiply long (top, indexed).

UMULLT (vectors): Unsigned multiply long (top).

UQADD (immediate): Unsigned saturating add immediate (unpredicated).

UQADD (vectors, predicated): Unsigned saturating addition (predicated).

UQADD (vectors, unpredicated): Unsigned saturating add vectors (unpredicated).

UQCVTN: Unsigned saturating extract narrow and interleave.

UQDECB: Unsigned saturating decrement scalar by multiple of 8-bit predicate constraint element count.

UQDECD (scalar): Unsigned saturating decrement scalar by multiple of 64-bit predicate constraint element count.

UQDECD (vector): Unsigned saturating decrement vector by multiple of 64-bit predicate constraint element count.

UQDECH (scalar): Unsigned saturating decrement scalar by multiple of 16-bit predicate constraint element count.

UQDECH (vector): Unsigned saturating decrement vector by multiple of 16-bit predicate constraint element count.

UQDECP (scalar): Unsigned saturating decrement scalar by count of true predicate elements.

UQDECP (vector): Unsigned saturating decrement vector by count of true predicate elements.

UQDECW (scalar): Unsigned saturating decrement scalar by multiple of 32-bit predicate constraint element count.

UQDECW (vector): Unsigned saturating decrement vector by multiple of 32-bit predicate constraint element count.

UQINCB: Unsigned saturating increment scalar by multiple of 8-bit predicate constraint element count.

UQINCD (scalar): Unsigned saturating increment scalar by multiple of 64-bit predicate constraint element count.

UQINCD (vector): Unsigned saturating increment vector by multiple of 64-bit predicate constraint element count.

UQINCH (scalar): Unsigned saturating increment scalar by multiple of 16-bit predicate constraint element count.

UQINCH (vector): Unsigned saturating increment vector by multiple of 16-bit predicate constraint element count.

UQINCP (scalar): Unsigned saturating increment scalar by count of true predicate elements.

UQINCP (vector): Unsigned saturating increment vector by count of true predicate elements.

UQINCW (scalar): Unsigned saturating increment scalar by multiple of 32-bit predicate constraint element count.

UQINCW (vector): Unsigned saturating increment vector by multiple of 32-bit predicate constraint element count.

UQRSHL: Unsigned saturating rounding shift left by vector (predicated).

UQRSHLR: Unsigned saturating rounding shift left reversed vectors (predicated).

UQRSHRN: Unsigned saturating rounding shift right narrow by immediate and interleave.

UQRSHRNB: Unsigned saturating rounding shift right narrow by immediate (bottom).

UQRSHRNT: Unsigned saturating rounding shift right narrow by immediate (top).

UQSHL (immediate): Unsigned saturating shift left by immediate.

UQSHL (vectors): Unsigned saturating shift left by vector (predicated).

UQSHLR: Unsigned saturating shift left reversed vectors (predicated).

UQSHRNB: Unsigned saturating shift right narrow by immediate (bottom).

UQSHRNT: Unsigned saturating shift right narrow by immediate (top).

UQSUB (immediate): Unsigned saturating subtract immediate (unpredicated).

UQSUB (vectors, predicated): Unsigned saturating subtraction (predicated).

UQSUB (vectors, unpredicated): Unsigned saturating subtract vectors (unpredicated).

UQSUBR: Unsigned saturating subtraction reversed vectors (predicated).

UQXTNB: Unsigned saturating extract narrow (bottom).

UQXTNT: Unsigned saturating extract narrow (top).

URECPE: Unsigned reciprocal estimate (predicated).

URHADD: Unsigned rounding halving addition.

URSHL: Unsigned rounding shift left by vector (predicated).

URSHLR: Unsigned rounding shift left reversed vectors (predicated).

URSHR: Unsigned rounding shift right by immediate.

URSQRTE: Unsigned reciprocal square root estimate (predicated).

URSRA: Unsigned rounding shift right and accumulate (immediate).

USDOT (indexed): Unsigned by signed integer indexed dot product.

USDOT (vectors): Unsigned by signed integer dot product.

USHLLB: Unsigned shift left long by immediate (bottom).

USHLLT: Unsigned shift left long by immediate (top).

USMMLA: Unsigned by signed integer matrix multiply-accumulate.

USQADD: Unsigned saturating addition of signed value.

USRA: Unsigned shift right and accumulate (immediate).

USUBLB: Unsigned subtract long (bottom).

USUBLT: Unsigned subtract long (top).

USUBWB: Unsigned subtract wide (bottom).

USUBWT: Unsigned subtract wide (top).

UUNPKHI, UUNPKLO: Unsigned unpack and extend half of vector.

UXTB, UXTH, UXTW: Unsigned byte / halfword / word extend (predicated).

UZP1, UZP2 (predicates): Concatenate even or odd elements from two predicates.

UZP1, UZP2 (vectors): Concatenate even or odd elements from two vectors.

UZPQ1: Concatenate even elements within each pair of quadword vector segments.

UZPQ2: Concatenate odd elements within each pair of quadword vector segments.

WHILEGE (predicate as counter): While decrementing signed scalar greater than or equal to scalar (predicate-as-counter).

WHILEGE (predicate pair): While decrementing signed scalar greater than or equal to scalar (pair of predicates).

WHILEGE (predicate): While decrementing signed scalar greater than or equal to scalar.

WHILEGT (predicate as counter): While decrementing signed scalar greater than scalar (predicate-as-counter).

WHILEGT (predicate pair): While decrementing signed scalar greater than scalar (pair of predicates).

WHILEGT (predicate): While decrementing signed scalar greater than scalar.

WHILEHI (predicate as counter): While decrementing unsigned scalar higher than scalar (predicate-as-counter).

WHILEHI (predicate pair): While decrementing unsigned scalar higher than scalar (pair of predicates).

WHILEHI (predicate): While decrementing unsigned scalar higher than scalar.

WHILEHS (predicate as counter): While decrementing unsigned scalar higher or same as scalar (predicate-as-counter).

WHILEHS (predicate pair): While decrementing unsigned scalar higher or same as scalar (pair of predicates).

WHILEHS (predicate): While decrementing unsigned scalar higher or same as scalar.

WHILELE (predicate as counter): While incrementing signed scalar less than or equal to scalar (predicate-as-counter).

WHILELE (predicate pair): While incrementing signed scalar less than or equal to scalar (pair of predicates).

WHILELE (predicate): While incrementing signed scalar less than or equal to scalar.

WHILELO (predicate as counter): While incrementing unsigned scalar lower than scalar (predicate-as-counter).

WHILELO (predicate pair): While incrementing unsigned scalar lower than scalar (pair of predicates).

WHILELO (predicate): While incrementing unsigned scalar lower than scalar.

WHILELS (predicate as counter): While incrementing unsigned scalar lower or same as scalar (predicate-as-counter).

WHILELS (predicate pair): While incrementing unsigned scalar lower or same as scalar (pair of predicates).

WHILELS (predicate): While incrementing unsigned scalar lower or same as scalar.

WHILELT (predicate as counter): While incrementing signed scalar less than scalar (predicate-as-counter).

WHILELT (predicate pair): While incrementing signed scalar less than scalar (pair of predicates).

WHILELT (predicate): While incrementing signed scalar less than scalar.

WHILERW: While free of read-after-write conflicts.

WHILEWR: While free of write-after-read/write conflicts.

WRFFR: Write the first-fault register.

XAR: Bitwise exclusive OR and rotate right by immediate.

ZIP1, ZIP2 (predicates): Interleave elements from two half predicates.

ZIP1, ZIP2 (vectors): Interleave elements from two half vectors.

ZIPQ1: Interleave elements from low halves of each pair of quadword vector segments.

ZIPQ2: Interleave elements from high halves of each pair of quadword vector segments.

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.