A64 -- SME Instructions (alphabetic order)

ADD (array accumulators): Add multi-vector to ZA array vector accumulators.

ADD (array results, multiple and single vector): Add replicated single vector to multi-vector with ZA array vector results.

ADD (array results, multiple vectors): Add multi-vector to multi-vector with ZA array vector results.

ADD (to vector): Add replicated single vector to multi-vector with multi-vector result.

ADDHA: Add horizontally vector elements to ZA tile.

ADDSPL: Add multiple of Streaming SVE predicate register size to scalar register.

ADDSVL: Add multiple of Streaming SVE vector register size to scalar register.

ADDVA: Add vertically vector elements to ZA tile.

BFADD: BFloat16 floating-point add multi-vector to ZA array vector accumulators.

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

BFCVT: Multi-vector floating-point convert from single-precision to packed BFloat16 format.

BFCVTN: Multi-vector floating-point convert from single-precision to interleaved BFloat16 format.

BFDOT (multiple and indexed vector): Multi-vector BFloat16 floating-point dot-product by indexed element.

BFDOT (multiple and single vector): Multi-vector BFloat16 floating-point dot-product by vector.

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

BFMAX (multiple and single vector): Multi-vector BFloat16 floating-point maximum by vector.

BFMAX (multiple vectors): Multi-vector BFloat16 floating-point maximum.

BFMAXNM (multiple and single vector): Multi-vector BFloat16 floating-point maximum number by vector.

BFMAXNM (multiple vectors): Multi-vector BFloat16 floating-point maximum number.

BFMIN (multiple and single vector): Multi-vector BFloat16 floating-point minimum by vector.

BFMIN (multiple vectors): Multi-vector BFloat16 floating-point minimum.

BFMINNM (multiple and single vector): Multi-vector BFloat16 floating-point minimum number by vector.

BFMINNM (multiple vectors): Multi-vector BFloat16 floating-point minimum number.

BFMLA (multiple and indexed vector): Multi-vector BFloat16 floating-point fused multiply-add by indexed element.

BFMLA (multiple and single vector): Multi-vector BFloat16 floating-point fused multiply-add by vector.

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

BFMLAL (multiple and indexed vector): Multi-vector BFloat16 floating-point multiply-add long by indexed element.

BFMLAL (multiple and single vector): Multi-vector BFloat16 floating-point multiply-add long by vector.

BFMLAL (multiple vectors): Multi-vector BFloat16 floating-point multiply-add long.

BFMLS (multiple and indexed vector): Multi-vector BFloat16 floating-point fused multiply-subtract by indexed element.

BFMLS (multiple and single vector): Multi-vector BFloat16 floating-point fused multiply-subtract by vector.

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

BFMLSL (multiple and indexed vector): Multi-vector BFloat16 floating-point multiply-subtract long by indexed element.

BFMLSL (multiple and single vector): Multi-vector BFloat16 floating-point multiply-subtract long by vector.

BFMLSL (multiple vectors): Multi-vector BFloat16 floating-point multiply-subtract long.

BFMOPA (non-widening): BFloat16 floating-point outer product and accumulate.

BFMOPA (widening): BFloat16 sum of outer products and accumulate.

BFMOPS (non-widening): BFloat16 floating-point outer product and subtract.

BFMOPS (widening): BFloat16 sum of outer products and subtract.

BFSUB: BFloat16 floating-point subtract multi-vector from ZA array vector accumulators.

BFVDOT: Multi-vector BFloat16 floating-point vertical dot-product by indexed element.

BMOPA: Bitwise exclusive NOR population count outer product and accumulate.

BMOPS: Bitwise exclusive NOR population count outer product and subtract.

FADD: Floating-point add multi-vector to ZA array vector accumulators.

FCLAMP: Multi-vector floating-point clamp to minimum/maximum number.

FCVT (narrowing): Multi-vector floating-point convert from single-precision to packed half-precision.

FCVT (widening): Multi-vector floating-point convert from half-precision to single-precision (in-order).

FCVTL: Multi-vector floating-point convert from half-precision to deinterleaved single-precision.

FCVTN: Multi-vector floating-point convert from single-precision to interleaved half-precision.

FCVTZS: Multi-vector floating-point convert to signed integer, rounding toward zero.

FCVTZU: Multi-vector floating-point convert to unsigned integer, rounding toward zero.

FDOT (multiple and indexed vector): Multi-vector half-precision floating-point dot-product by indexed element.

FDOT (multiple and single vector): Multi-vector half-precision floating-point dot-product by vector.

FDOT (multiple vectors): Multi-vector half-precision floating-point dot-product.

FMAX (multiple and single vector): Multi-vector floating-point maximum by vector.

FMAX (multiple vectors): Multi-vector floating-point maximum.

FMAXNM (multiple and single vector): Multi-vector floating-point maximum number by vector.

FMAXNM (multiple vectors): Multi-vector floating-point maximum number.

FMIN (multiple and single vector): Multi-vector floating-point minimum by vector.

FMIN (multiple vectors): Multi-vector floating-point minimum.

FMINNM (multiple and single vector): Multi-vector floating-point minimum number by vector.

FMINNM (multiple vectors): Multi-vector floating-point minimum number.

FMLA (multiple and indexed vector): Multi-vector floating-point fused multiply-add by indexed element.

FMLA (multiple and single vector): Multi-vector floating-point fused multiply-add by vector.

FMLA (multiple vectors): Multi-vector floating-point fused multiply-add.

FMLAL (multiple and indexed vector): Multi-vector floating-point multiply-add long by indexed element.

FMLAL (multiple and single vector): Multi-vector floating-point multiply-add long by vector.

FMLAL (multiple vectors): Multi-vector floating-point multiply-add long.

FMLS (multiple and indexed vector): Multi-vector floating-point fused multiply-subtract by indexed element.

FMLS (multiple and single vector): Multi-vector floating-point fused multiply-subtract by vector.

FMLS (multiple vectors): Multi-vector floating-point fused multiply-subtract.

FMLSL (multiple and indexed vector): Multi-vector floating-point multiply-subtract long by indexed element.

FMLSL (multiple and single vector): Multi-vector floating-point multiply-subtract long by vector.

FMLSL (multiple vectors): Multi-vector floating-point multiply-subtract long.

FMOPA (non-widening): Floating-point outer product and accumulate.

FMOPA (widening): Half-precision floating-point sum of outer products and accumulate.

FMOPS (non-widening): Floating-point outer product and subtract.

FMOPS (widening): Half-precision floating-point sum of outer products and subtract.

FRINTA: Multi-vector floating-point round to integral value, to nearest with ties away from zero.

FRINTM: Multi-vector floating-point round to integral value, toward minus Infinity.

FRINTN: Multi-vector floating-point round to integral value, to nearest with ties to even.

FRINTP: Multi-vector floating-point round to integral value, toward plus Infinity.

FSUB: Floating-point subtract multi-vector from ZA array vector accumulators.

FVDOT: Multi-vector half-precision floating-point vertical dot-product by indexed element.

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

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

LD1B (scalar plus scalar, tile slice): Contiguous load of bytes to 8-bit element ZA tile slice.

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

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

LD1D (scalar plus scalar, tile slice): Contiguous load of doublewords to 64-bit element ZA tile slice.

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

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

LD1H (scalar plus scalar, tile slice): Contiguous load of halfwords to 16-bit element ZA tile slice.

LD1Q: Contiguous load of quadwords to 128-bit element ZA tile slice.

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

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

LD1W (scalar plus scalar, tile slice): Contiguous load of words to 32-bit element ZA tile slice.

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

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

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

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

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

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

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

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

LDR (vector): Load ZA array vector.

LDR (ZT0): Load ZT0 register.

LUTI2 (four registers): Lookup table read with 2-bit indexes.

LUTI2 (single): Lookup table read with 2-bit indexes.

LUTI2 (two registers): Lookup table read with 2-bit indexes.

LUTI4 (four registers): Lookup table read with 4-bit indexes.

LUTI4 (single): Lookup table read with 4-bit indexes.

LUTI4 (two registers): Lookup table read with 4-bit indexes.

MOV (array to vector, four registers): Move four ZA single-vector groups to four vector registers: an alias of MOVA (array to vector, four registers).

MOV (array to vector, two registers): Move two ZA single-vector groups to two vector registers: an alias of MOVA (array to vector, two registers).

MOV (tile to vector, four registers): Move four ZA tile slices to four vector registers: an alias of MOVA (tile to vector, four registers).

MOV (tile to vector, single): Move ZA tile slice to vector register: an alias of MOVA (tile to vector, single).

MOV (tile to vector, two registers): Move two ZA tile slices to two vector registers: an alias of MOVA (tile to vector, two registers).

MOV (vector to array, four registers): Move four vector registers to four ZA single-vector groups: an alias of MOVA (vector to array, four registers).

MOV (vector to array, two registers): Move two vector registers to two ZA single-vector groups: an alias of MOVA (vector to array, two registers).

MOV (vector to tile, four registers): Move four vector registers to four ZA tile slices: an alias of MOVA (vector to tile, four registers).

MOV (vector to tile, single): Move vector register to ZA tile slice: an alias of MOVA (vector to tile, single).

MOV (vector to tile, two registers): Move two vector registers to two ZA tile slices: an alias of MOVA (vector to tile, two registers).

MOVA (array to vector, four registers): Move four ZA single-vector groups to four vector registers.

MOVA (array to vector, two registers): Move two ZA single-vector groups to two vector registers.

MOVA (tile to vector, four registers): Move four ZA tile slices to four vector registers.

MOVA (tile to vector, single): Move ZA tile slice to vector register.

MOVA (tile to vector, two registers): Move two ZA tile slices to two vector registers.

MOVA (vector to array, four registers): Move four vector registers to four ZA single-vector groups.

MOVA (vector to array, two registers): Move two vector registers to two ZA single-vector groups.

MOVA (vector to tile, four registers): Move four vector registers to four ZA tile slices.

MOVA (vector to tile, single): Move vector register to ZA tile slice.

MOVA (vector to tile, two registers): Move two vector registers to two ZA tile slices.

MOVAZ (array to vector, four registers): Move and zero four ZA single-vector groups to vector registers.

MOVAZ (array to vector, two registers): Move and zero two ZA single-vector groups to vector registers.

MOVAZ (tile to vector, four registers): Move and zero four ZA tile slices to vector registers.

MOVAZ (tile to vector, single): Move and zero ZA tile slice to vector register.

MOVAZ (tile to vector, two registers): Move and zero two ZA tile slices to vector registers.

MOVT (scalar to ZT0): Move 8 bytes from general-purpose register to ZT0.

MOVT (ZT0 to scalar): Move 8 bytes from ZT0 to general-purpose register.

RDSVL: Read multiple of Streaming SVE vector register size to scalar register.

SCLAMP: Multi-vector signed clamp to minimum/maximum vector.

SCVTF: Multi-vector signed integer convert to floating-point.

SDOT (2-way, multiple and indexed vector): Multi-vector signed integer dot-product by indexed element.

SDOT (2-way, multiple and single vector): Multi-vector signed integer dot-product by vector.

SDOT (2-way, multiple vectors): Multi-vector signed integer dot-product.

SDOT (4-way, multiple and indexed vector): Multi-vector signed integer dot-product by indexed element.

SDOT (4-way, multiple and single vector): Multi-vector signed integer dot-product by vector.

SDOT (4-way, multiple vectors): Multi-vector signed integer dot-product.

SEL: Multi-vector conditionally select elements from two vectors.

SMAX (multiple and single vector): Multi-vector signed maximum by vector.

SMAX (multiple vectors): Multi-vector signed maximum.

SMIN (multiple and single vector): Multi-vector signed minimum by vector.

SMIN (multiple vectors): Multi-vector signed minimum.

SMLAL (multiple and indexed vector): Multi-vector signed integer multiply-add long by indexed element.

SMLAL (multiple and single vector): Multi-vector signed integer multiply-add long by vector.

SMLAL (multiple vectors): Multi-vector signed integer multiply-add long.

SMLALL (multiple and indexed vector): Multi-vector signed integer multiply-add long-long by indexed element.

SMLALL (multiple and single vector): Multi-vector signed integer multiply-add long-long by vector.

SMLALL (multiple vectors): Multi-vector signed integer multiply-add long-long.

SMLSL (multiple and indexed vector): Multi-vector signed integer multiply-subtract long by indexed element.

SMLSL (multiple and single vector): Multi-vector signed integer multiply-subtract long by vector.

SMLSL (multiple vectors): Multi-vector signed integer multiply-subtract long.

SMLSLL (multiple and indexed vector): Multi-vector signed integer multiply-subtract long-long by indexed element.

SMLSLL (multiple and single vector): Multi-vector signed integer multiply-subtract long-long by vector.

SMLSLL (multiple vectors): Multi-vector signed integer multiply-subtract long-long.

SMOPA (2-way): Signed integer sum of outer products and accumulate.

SMOPA (4-way): Signed integer sum of outer products and accumulate.

SMOPS (2-way): Signed integer sum of outer products and subtract.

SMOPS (4-way): Signed integer sum of outer products and subtract.

SQCVT (four registers): Multi-vector signed saturating extract narrow.

SQCVT (two registers): Multi-vector signed saturating extract narrow.

SQCVTN: Multi-vector signed saturating extract narrow and interleave.

SQCVTU (four registers): Multi-vector signed saturating unsigned extract narrow.

SQCVTU (two registers): Multi-vector signed saturating unsigned extract narrow.

SQCVTUN: Multi-vector signed saturating unsigned extract narrow and interleave.

SQDMULH (multiple and single vector): Multi-vector signed saturating doubling multiply high by vector.

SQDMULH (multiple vectors): Multi-vector signed saturating doubling multiply high.

SQRSHR (four registers): Multi-vector signed saturating rounding shift right narrow by immediate.

SQRSHR (two registers): Multi-vector signed saturating rounding shift right narrow by immediate.

SQRSHRN: Multi-vector signed saturating rounding shift right narrow by immediate and interleave.

SQRSHRU (four registers): Multi-vector signed saturating rounding shift right unsigned narrow by immediate.

SQRSHRU (two registers): Multi-vector signed saturating rounding shift right unsigned narrow by immediate.

SQRSHRUN: Multi-vector signed saturating rounding shift right unsigned narrow by immediate and interleave.

SRSHL (multiple and single vector): Multi-vector signed rounding shift left by vector.

SRSHL (multiple vectors): Multi-vector signed rounding shift left.

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

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

ST1B (scalar plus scalar, tile slice): Contiguous store of bytes from 8-bit element ZA tile slice.

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

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

ST1D (scalar plus scalar, tile slice): Contiguous store of doublewords from 64-bit element ZA tile slice.

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

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

ST1H (scalar plus scalar, tile slice): Contiguous store of halfwords from 16-bit element ZA tile slice.

ST1Q: Contiguous store of quadwords from 128-bit element ZA tile slice.

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

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

ST1W (scalar plus scalar, tile slice): Contiguous store of words from 32-bit element ZA tile slice.

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

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

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

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

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

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

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

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

STR (vector): Store ZA array vector.

STR (ZT0): Store ZT0 register.

SUB (array accumulators): Subtract multi-vector from ZA array vector accumulators.

SUB (array results, multiple and single vector): Subtract replicated single vector from multi-vector with ZA array vector results.

SUB (array results, multiple vectors): Subtract multi-vector from multi-vector with ZA array vector results.

SUDOT (multiple and indexed vector): Multi-vector signed by unsigned integer dot-product by indexed elements.

SUDOT (multiple and single vector): Multi-vector signed by unsigned integer dot-product by vector.

SUMLALL (multiple and indexed vector): Multi-vector signed by unsigned integer multiply-add long-long by indexed element.

SUMLALL (multiple and single vector): Multi-vector signed by unsigned integer multiply-add long-long by vector.

SUMOPA: Signed by unsigned integer sum of outer products and accumulate.

SUMOPS: Signed by unsigned integer sum of outer products and subtract.

SUNPK: Unpack and sign-extend multi-vector elements.

SUVDOT: Multi-vector signed by unsigned integer vertical dot-product by indexed element.

SVDOT (2-way): Multi-vector signed integer vertical dot-product by indexed element.

SVDOT (4-way): Multi-vector signed integer vertical dot-product by indexed element.

UCLAMP: Multi-vector unsigned clamp to minimum/maximum vector.

UCVTF: Multi-vector unsigned integer convert to floating-point.

UDOT (2-way, multiple and indexed vector): Multi-vector unsigned integer dot-product by indexed element.

UDOT (2-way, multiple and single vector): Multi-vector unsigned integer dot-product by vector.

UDOT (2-way, multiple vectors): Multi-vector unsigned integer dot-product.

UDOT (4-way, multiple and indexed vector): Multi-vector unsigned integer dot-product by indexed element.

UDOT (4-way, multiple and single vector): Multi-vector unsigned integer dot-product by vector.

UDOT (4-way, multiple vectors): Multi-vector unsigned integer dot-product.

UMAX (multiple and single vector): Multi-vector unsigned maximum by vector.

UMAX (multiple vectors): Multi-vector unsigned maximum.

UMIN (multiple and single vector): Multi-vector unsigned minimum by vector.

UMIN (multiple vectors): Multi-vector unsigned minimum.

UMLAL (multiple and indexed vector): Multi-vector unsigned integer multiply-add long by indexed element.

UMLAL (multiple and single vector): Multi-vector unsigned integer multiply-add long by vector.

UMLAL (multiple vectors): Multi-vector unsigned integer multiply-add long.

UMLALL (multiple and indexed vector): Multi-vector unsigned integer multiply-add long-long by indexed element.

UMLALL (multiple and single vector): Multi-vector unsigned integer multiply-add long-long by vector.

UMLALL (multiple vectors): Multi-vector unsigned integer multiply-add long-long.

UMLSL (multiple and indexed vector): Multi-vector unsigned integer multiply-subtract long by indexed element.

UMLSL (multiple and single vector): Multi-vector unsigned integer multiply-subtract long by vector.

UMLSL (multiple vectors): Multi-vector unsigned integer multiply-subtract long.

UMLSLL (multiple and indexed vector): Multi-vector unsigned integer multiply-subtract long-long by indexed element.

UMLSLL (multiple and single vector): Multi-vector unsigned integer multiply-subtract long-long by vector.

UMLSLL (multiple vectors): Multi-vector unsigned integer multiply-subtract long-long.

UMOPA (2-way): Unsigned integer sum of outer products and accumulate.

UMOPA (4-way): Unsigned integer sum of outer products and accumulate.

UMOPS (2-way): Unsigned integer sum of outer products and subtract.

UMOPS (4-way): Unsigned integer sum of outer products and subtract.

UQCVT (four registers): Multi-vector unsigned saturating extract narrow.

UQCVT (two registers): Multi-vector unsigned saturating extract narrow.

UQCVTN: Multi-vector unsigned saturating extract narrow and interleave.

UQRSHR (four registers): Multi-vector unsigned saturating rounding shift right narrow by immediate.

UQRSHR (two registers): Multi-vector unsigned saturating rounding shift right narrow by immediate.

UQRSHRN: Multi-vector unsigned saturating rounding shift right narrow by immediate and interleave.

URSHL (multiple and single vector): Multi-vector unsigned rounding shift left by vector.

URSHL (multiple vectors): Multi-vector unsigned rounding shift left.

USDOT (multiple and indexed vector): Multi-vector unsigned by signed integer dot-product by indexed element.

USDOT (multiple and single vector): Multi-vector unsigned by signed integer dot-product by vector.

USDOT (multiple vectors): Multi-vector unsigned by signed integer dot-product.

USMLALL (multiple and indexed vector): Multi-vector unsigned by signed integer multiply-add long-long by indexed element.

USMLALL (multiple and single vector): Multi-vector unsigned by signed integer multiply-add long-long by vector.

USMLALL (multiple vectors): Multi-vector unsigned by signed integer multiply-add long-long.

USMOPA: Unsigned by signed integer sum of outer products and accumulate.

USMOPS: Unsigned by signed integer sum of outer products and subtract.

USVDOT: Multi-vector unsigned by signed integer vertical dot-product by indexed element.

UUNPK: Unpack and zero-extend multi-vector elements.

UVDOT (2-way): Multi-vector unsigned integer vertical dot-product by indexed element.

UVDOT (4-way): Multi-vector unsigned integer vertical dot-product by indexed element.

UZP (four registers): Concatenate elements from four vectors.

UZP (two registers): Concatenate elements from two vectors.

ZERO (double-vector): Zero ZA double-vector groups.

ZERO (quad-vector): Zero ZA quad-vector groups.

ZERO (single-vector): Zero ZA single-vector groups.

ZERO (tile): Zero a list of 64-bit element ZA tiles.

ZERO (ZT0): Zero ZT0.

ZIP (four registers): Interleave elements from four vectors.

ZIP (two registers): Interleave elements from two vectors.

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