<u>Base</u>	SIMD&FP	<u>SVE</u>	<u>SME</u>	Index by
<u>Instructions</u>	<u>Instructions</u>	<u>Instructions</u>	<u>Instructions</u>	<b>Encoding</b>

## **SUBP**

Subtract Pointer subtracts the 56-bit address held in the second source register from the 56-bit address held in the first source register, sign-extends the result to 64-bits, and writes the result to the destination register.

## Integer (FEAT MTE)

```
SUBP <Xd>, <Xn SP>, <Xm SP>

if !IsFeatureImplemented(FEAT_MTE) then UNDEFINED;
integer d = UInt(Xd);
integer n = UInt(Xn);
integer m = UInt(Xm);
```

## **Assembler Symbols**

```
<Xd> Is the 64-bit name of the general-purpose destination register, encoded in the "Xd" field.
<Xn|SP> Is the 64-bit name of the first source general-purpose register or stack pointer, encoded in the "Xn" field.
<Xm|SP> Is the 64-bit name of the second general-purpose source register or stack pointer, encoded in the "Xm" field.
```

## Operation

```
bits(64) operand1 = if n == 31 then SP[] else X[n, 64];
bits(64) operand2 = if m == 31 then SP[] else X[m, 64];
operand1 = SignExtend(operand1<55:0>, 64);
operand2 = SignExtend(operand2<55:0>, 64);
bits(64) result;
operand2 = NOT(operand2);
(result, -) = AddWithCarry(operand1, operand2, '1');
X[d, 64] = result;
```

BaseSIMD&FPSVESMEIndex byInstructionsInstructionsInstructionsEncoding

 $Internal\ version\ only: is a\ v33.64,\ AdvSIMD\ v29.12,\ pseudocode\ no\_diffs\_2023\_09\_RC2,\ sve\ v2023-06\_rel\ ;\ Build\ timestamp:\ 2023-09-18T17:56$ 

Sh Pseu

Pseu

Copyright © 2010-2023 Arm Limited or it	s affiliates. All rights reserved. This document is Non-Confidential.