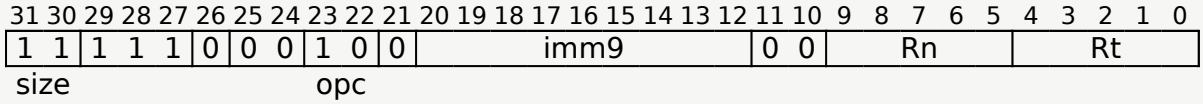
PRFUM

Prefetch Memory (unscaled offset) signals the memory system that data memory accesses from a specified address are likely to occur in the near future. The memory system can respond by taking actions that are expected to speed up the memory accesses when they do occur, such as preloading the cache line containing the specified address into one or more caches.

The effect of a `PRFUM` instruction is implementation defined. For more information, see *Prefetch memory*.

For information about memory accesses, see *Load/Store addressing modes*.



PRFUM (<prfop>|#<imm5>), [<Xn|SP>{, #<simm>}]

```
bits(64) offset = SignExtend(imm9, 64);
```

Assembler Symbols

<prfop>
Is the prefetch operation, defined as <type><target><policy>.

<type> is one of:

PLD
: Prefetch for load, encoded in the "Rt<4:3>" field as 0b00.

PLI
: Preload instructions, encoded in the "Rt<4:3>" field as 0b01.

PST
: Prefetch for store, encoded in the "Rt<4:3>" field as 0b10.

<target> is one of:

L1
: Level 1 cache, encoded in the "Rt<2:1>" field as 0b00.

L2
: Level 2 cache, encoded in the "Rt<2:1>" field as 0b01.

L3
: Level 3 cache, encoded in the "Rt<2:1>" field as 0b10.

<policy> is one of:

KEEP
: Retained or temporal prefetch, allocated in the cache normally. Encoded in the "Rt<0>" field as 0.

STRM
: Streaming or non-temporal prefetch, for data that is used only once. Encoded in the "Rt<0>" field as 1.

For more information on these prefetch operations, see *Prefetch memory*.

For other encodings of the "Rt" field, use <imm5>.

<imm5>
Is the prefetch operation encoding as an immediate, in the range 0 to 31, encoded in the "Rt" field.

This syntax is only for encodings that are not accessible using <prfop>.

<Xn|SP>
Is the 64-bit name of the general-purpose base register or stack pointer, encoded in the "Rn" field.

<simm>
Is the optional signed immediate byte offset, in the range -256 to 255, defaulting to 0 and encoded in the "imm9" field.

Shared Decode

```
integer n = UInt(Rn);
integer t = UInt(Rt);
```

Operation

```
bits(64) address;

if n == 31 then
    address = SP[];
else
    address = X[n, 64];

address = address + offset;

Prefetch(address, t<4:0>);
```

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