PRFM (register)

Prefetch Memory (register) 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 `PRFM` instruction is implementation defined. For more information, see *Prefetch memory*.

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

31 30	29 28 27	26	25 24	23 22	21	20 19 18 17 16	15 14 13	12	11 10	9 8 7 6 5	4 3 2 1 0
1 1	1 1 1	0	0 0	1 0	1	Rm	x 1 x	S	1 0	Rn	!= 11xxx
size				opc			option				Rt

```
PRFM (<prfop>|#<imm5>), [<Xn|SP>, (<Wm>|<Xm>){, <extend> {<amount>}}
```

```
if option<1> == '0' then UNDEFINED;     // sub-word index
ExtendType extend_type = DecodeRegExtend(option);
integer shift = if S == '1' then 3 else 0;
```

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. SLC When FEAT_PRFMSLC is implemented, system level cache, encoded in the "Rt<2:1>" field as 0b11. <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.
<Wm>	When option<0> is set to 0, is the 32-bit name of the general-purpose index register, encoded in the "Rm" field.

<Xm> When option<0> is set to 1, is the 64-bit name of the general-purpose index register, encoded in the "Rm" field.

<extend> Is the index extend/shift specifier, defaulting to LSL, and which must be omitted for the LSL option when <amount> is omitted. encoded in “option”:

option	<extend>
010	UXTW
011	LSL
110	SXTW
111	SXTX

<amount> Is the index shift amount, optional only when <extend> is not LSL. Where it is permitted to be optional, it defaults to #0. It is encoded in “S”:

S	<amount>
0	#0
1	#3

Shared Decode

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

Operation

```
bits(64) offset = ExtendReg(m, extend_type, shift, 64);
bits(64) address;

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

address = address + offset;

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