RPRFM

Range Prefetch Memory signals the memory system that data memory accesses from a specified range of addresses are likely to occur in the near future. The instruction may also signal the memory system about the likelihood of data reuse of the specified range of addresses. The memory system can respond by taking actions that are expected to speed up the memory accesses when they do occur, such as prefetching locations within the specified address ranges into one or more caches. The memory system may also exploit the data reuse hints to decide whether to retain the data in other caches upon eviction from the innermost caches or to discard it.

The effect of an `RPRFM` instruction is implementation defined, but because these signals are only hints, the instruction cannot cause a synchronous Data Abort exception and is guaranteed not to access Device memory. It is valid for the PE to treat this instruction as a NOP.

An `RPRFM` instruction specifies the type of accesses and range of addresses using the following parameters:

- 'Type', in the <rprfop> operand opcode bits, specifies whether the prefetched data will be accessed by load or store instructions.
- 'Policy', in the <rprfop> operand opcode bits, specifies whether the data is likely to be reused or if it is a streaming, non-temporal prefetch. If a streaming prefetch is specified, then the 'ReuseDistance' parameter is ignored.
- 'BaseAddress', in the 64-bit base register, holds the initial block address for the accesses.
- 'ReuseDistance', in the metadata register bits[63:60], indicates the maximum number of bytes to be accessed by this PE before executing the next `RPRFM` instruction that specifies the same range. This includes the total number of bytes inside and outside of the range that will be accessed by the same PE. This parameter can be used to influence cache eviction and replacement policies, in order to retain the data in the most optimal levels of the memory hierarchy after each access. If software cannot easily determine the amount of other memory that will be accessed, these bits can be set to zero to indicate that 'ReuseDistance' is not known. Otherwise, these four bits encode decreasing powers of two in the range 512MiB (0b0001) to 32KiB (0b1111).
- 'Stride', in the metadata register bits[59:38], is a signed, two's complement integer encoding of the number of bytes to advance the block address after 'Length' bytes have been accessed, in the range -2MiB to +2MiB-1B. A negative value indicates that the block address is advanced in a descending direction.
- 'Count', in the metadata register bits[37:22], is an unsigned integer encoding of the number of blocks of data to be accessed minus 1, representing the range 1 to 65536 blocks. If 'Count' is 0, then the 'Stride' parameter is ignored and only a single block of contiguous

bytes from 'BaseAddress' to ('BaseAddress' + 'Length' - 1) is described.
- 'Length', in the metadata register bits[21:0], is a signed, two's complement integer encoding of the number of contiguous bytes to be accessed starting from the current block address, without changing the block address, in the range -2MiB to +2MiB-1B. A negative value indicates that the bytes are accessed in a descending direction.

Note

Software is expected to honor the parameters it provides to the `RPRFM` instruction, and the same PE should access all locations in the range, in the direction specified by the sign of the 'Length' and 'Stride' parameters. A range prefetch is considered active on a PE until all locations in the range have been accessed by the PE. A range prefetch might also be inactivated by the PE prior to completion, for example due to a software context switch or lack of hardware resources.

Software should not specify overlapping addresses in multiple active ranges. If a range is expected to be accessed by both load and store instructions (read-modify-write), then a single range with a 'Type' parameter of PST (prefetch for store) should be specified.

Integer
(FEAT_RPRFM)

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	1 1 x x x
size				opc			option				Rt

RPRFM (<rprfop>|#<imm6>), <Xm>, [<Xn|SP>]

```
bits(6) operation = option<2>:option<0>:S:Rt<2:0>;
integer n = UInt(Rn);
integer m = UInt(Rm);
```

Assembler Symbols

<rprfop>	Is the range prefetch operation, defined as <type><policy>. <type> is one of: PLD Prefetch for load, encoded in the "Rt<0>" field as 0. PST Prefetch for store, encoded in the "Rt<0>" field as 1. <policy> is one of: KEEP Retained or temporal prefetch, for data that is expected to be kept in caches to be accessed more than once, encoded in the "option<2>:option<0>:S:Rt<2:1>" fields as 0b00000. STRM Streaming or non-temporal prefetch, for data that is expected to be accessed once and not reused, encoded in the "option<2>:option<0>:S:Rt<2:1>" fields as 0b00010. For other encodings of the "option<2>:option<0>:S:Rt<2:0>" fields, use <imm6>.
<imm6>	Is the range prefetch operation encoding as an immediate, in the range 0 to 63, encoded in "option<2>:option<0>:S:Rt<2:0>". This syntax is only for encodings that are not representable using <rprfop>.
<Xm>	Is the 64-bit name of the general-purpose register that holds an encoding of the metadata, encoded in the "Rm" field.
<Xn|SP>	Is the 64-bit name of the general-purpose base register or stack pointer, encoded in the "Rn" field.

Operation

```
bits(64) address = if n == 31 then SP[] else X[n, 64];
bits(64) metadata = X[m, 64];
integer stride = SInt(metadata<59:38>);
integer count = UInt(metadata<37:22>) + 1;
integer length = SInt(metadata<21:0>);
integer reuse;

if metadata<63:60> == '0000' then
    reuse = -1;    // Not known
else
    reuse = 32768 << (15 - UInt(metadata<63:60>));

Hint_RangePrefetch(address, length, stride, count, reuse, operation);
```