# x86 Instruction Set Reference

# PREFETCHh

## Prefetch Data Into Caches

| Opcode | Mnemonic | Description |
|---|---|---|
| 0F 18 /1 | PREFETCHT0 m8 | Move data from m8 closer to the processor using T0 hint. |
| 0F 18 /2 | PREFETCHT1 m8 | Move data from m8 closer to the processor using T1 hint. |
| 0F 18 /3 | PREFETCHT2 m8 | Move data from m8 closer to the processor using T2 hint. |
| 0F 18 /0 | PREFETCHNTA m8 | Move data from m8 closer to the processor using NTA hint. |

### Description

Fetches the line of data from memory that contains the byte specified with the source operand to a location in the cache hierarchy spe
hint:

T0 (temporal data)-prefetch data into all levels of the cache hierarchy.
Pentium III processor-1st- or 2nd-level cache.
Pentium 4 and Intel Xeon processors-2nd-level cache.
T1 (temporal data with respect to first level cache)-prefetch data into level 2 cache and higher.
Pentium III processor-2nd-level cache.
Pentium 4 and Intel Xeon processors-2nd-level cache.
T2 (temporal data with respect to second level cache)-prefetch data into level 2 cache and higher.
Pentium III processor-2nd-level cache.
Pentium 4 and Intel Xeon processors-2nd-level cache.
NTA (non-temporal data with respect to all cache levels)-prefetch data into non-temporal cache structure and into a location close to the p
minimizing cache pollution.
Pentium III processor-1st-level cache - Pentium 4 and Intel Xeon processors-2nd-level cache The source operand is a byte memory locati
hints are encoded into the machine level instruction using bits 3 through 5 of the ModR/M byte. Use of any ModR/M value other than the s
lead to unpredictable behavior.) If the line selected is already present in the cache hierarchy at a level closer to the processor, no data mo
Prefetches from uncacheable or WC memory are ignored.

The PREFETCHh instruction is merely a hint and does not affect program behavior. If executed, this instruction moves data closer t
anticipation of future use.

The implementation of prefetch locality hints is implementation-dependent, and can be overloaded or ignored by a processor implementat
data prefetched is also processor implementation-dependent. It will, however, be a minimum of 32 bytes.

It should be noted that processors are free to speculatively fetch and cache data from system memory regions that are assigned a memor
speculative reads (that is, the WB, WC, and WT memory types). A PREFETCHh instruction is considered a hint to this speculative beha
speculative fetching can occur at any time and is not tied to instruction execution, a PREFETCHh instruction is not ordered with re
instructions (MFENCE, SFENCE, and LFENCE) or locked memory references. A PREFETCHh instruction is also unordered with res
instructions, other PREFETCHh instructions, or any other general instruction. It is ordered with respect to serializing instructions such as
OUT, and MOV CR.

### Operation

```
Fetch(m8);
```