# EL318: Lecture 11-1

# Evolution of iA32 processors

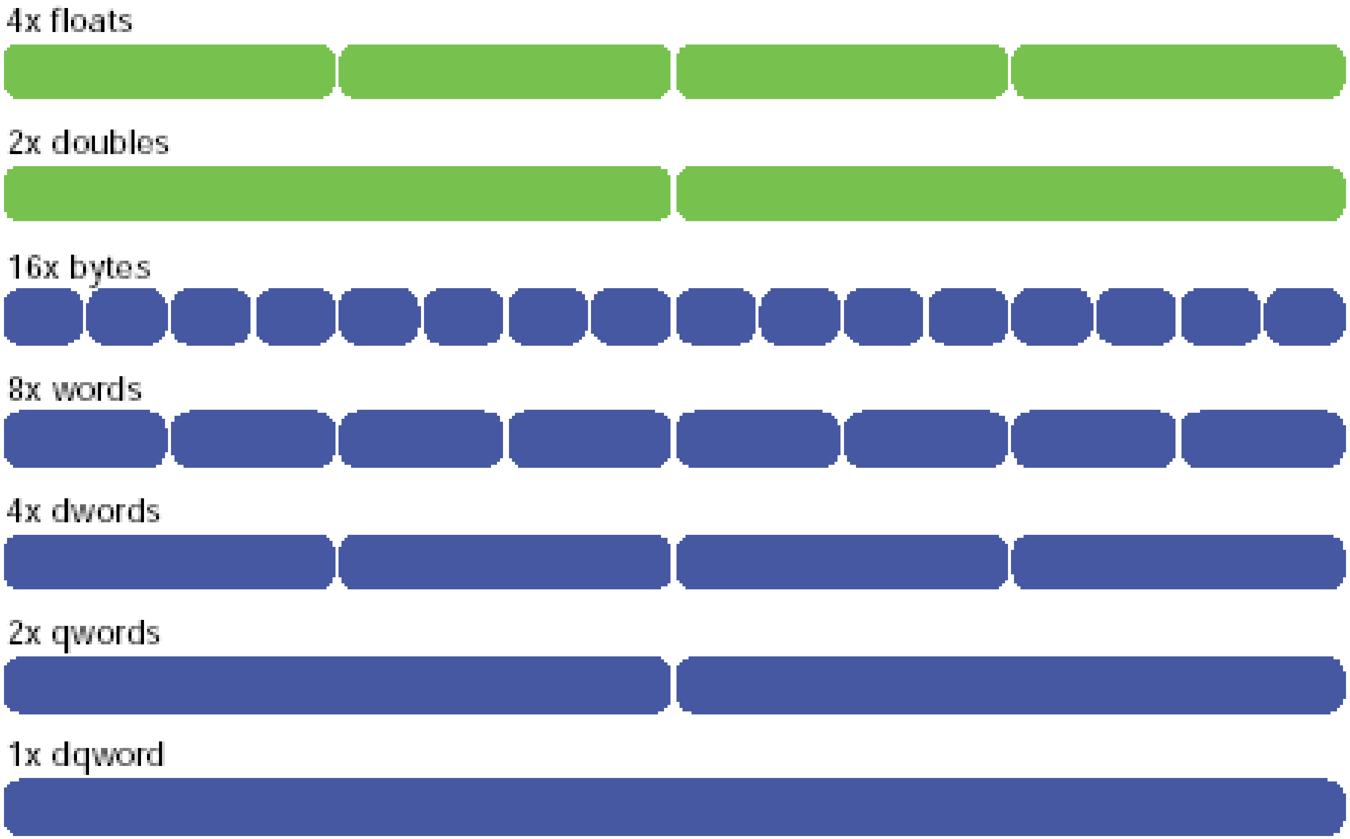
# Agenda

- Modern Floating Point processing: SSE and SSE2
- Hyperthreading
- Pentium III, Pentium 4 and Pentium M
- AMD64 and the K8
- Intel’s extension 64 bits.

# Streaming SIMD Extensions

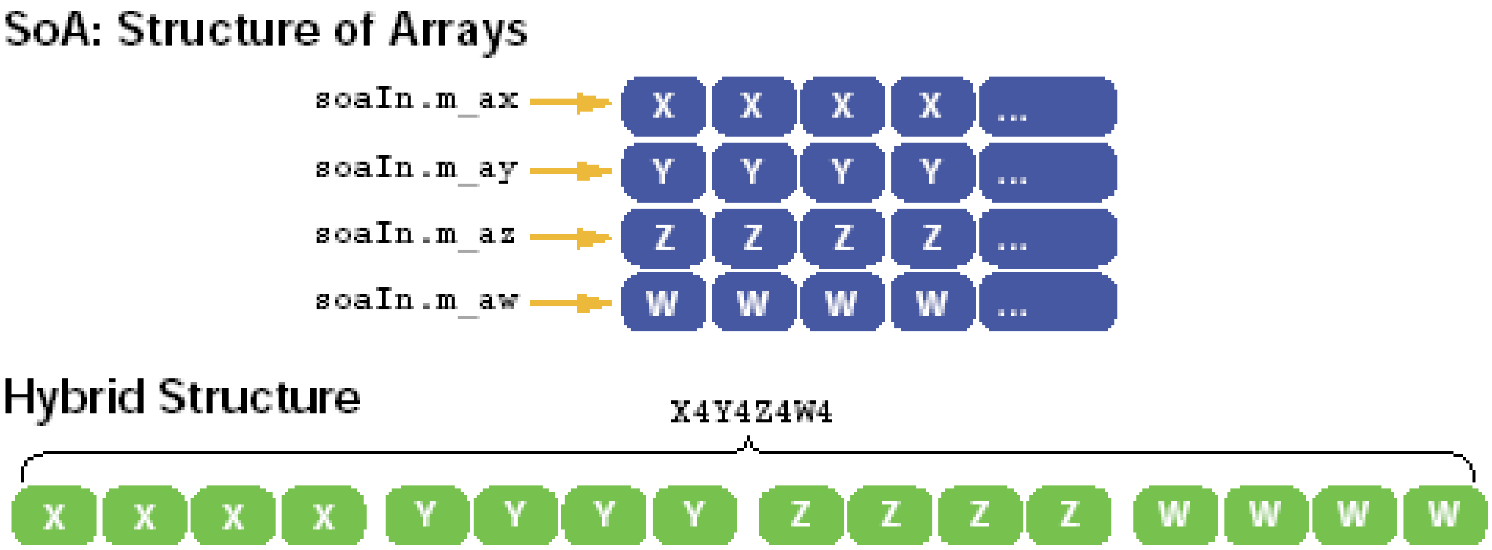
- Introduced with the Pentium III and extended to support additional types in the Pentium 4.
- A replacement for the stack-based 8087 instructions.
- SIMD was once a complete architecture, eg the ICL DAP, intended mainly for bit manipulation of images.
- Also supports prefetch and other instructions to imporve cache behaviour.

# SSE2 data types



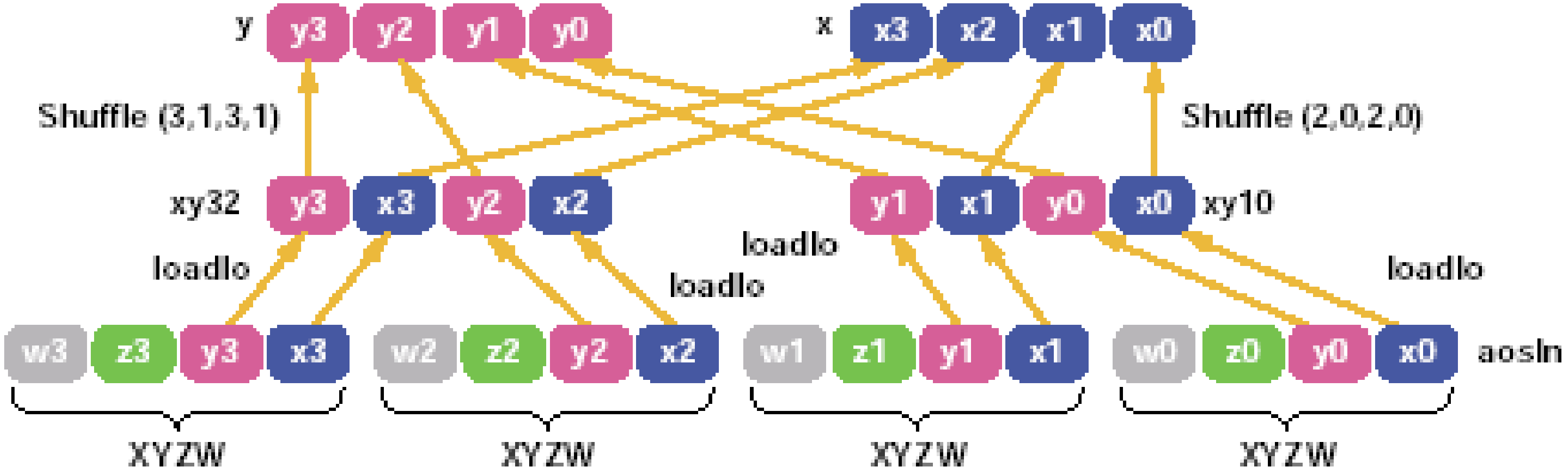
Anything that fits into 16 bytes

# SSE needs appropriate data organisation

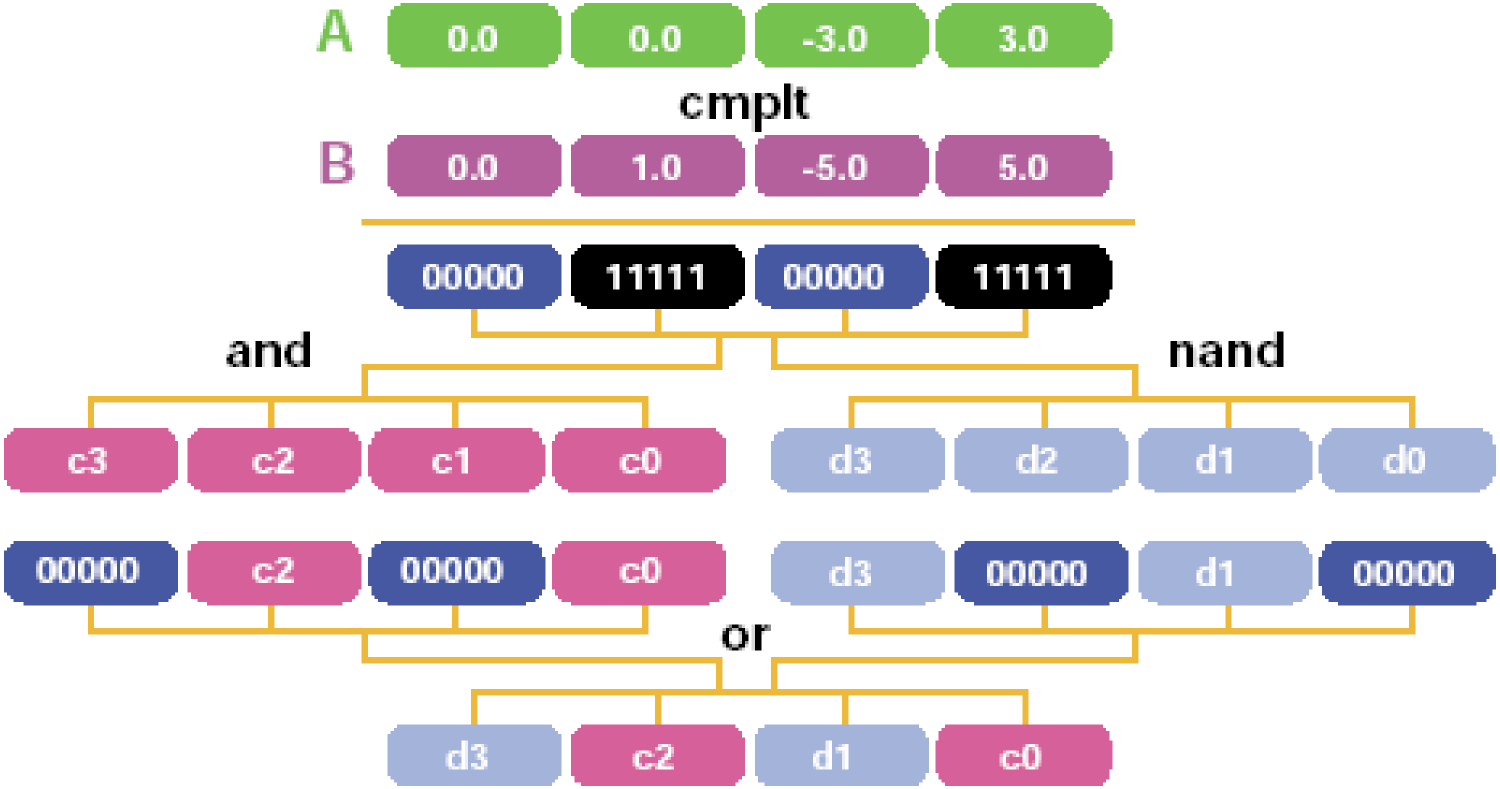


A Structure of arrays is better than an array of structures (hybrid structure)

# Data can be shuffled



# Avoid branches using bit masks

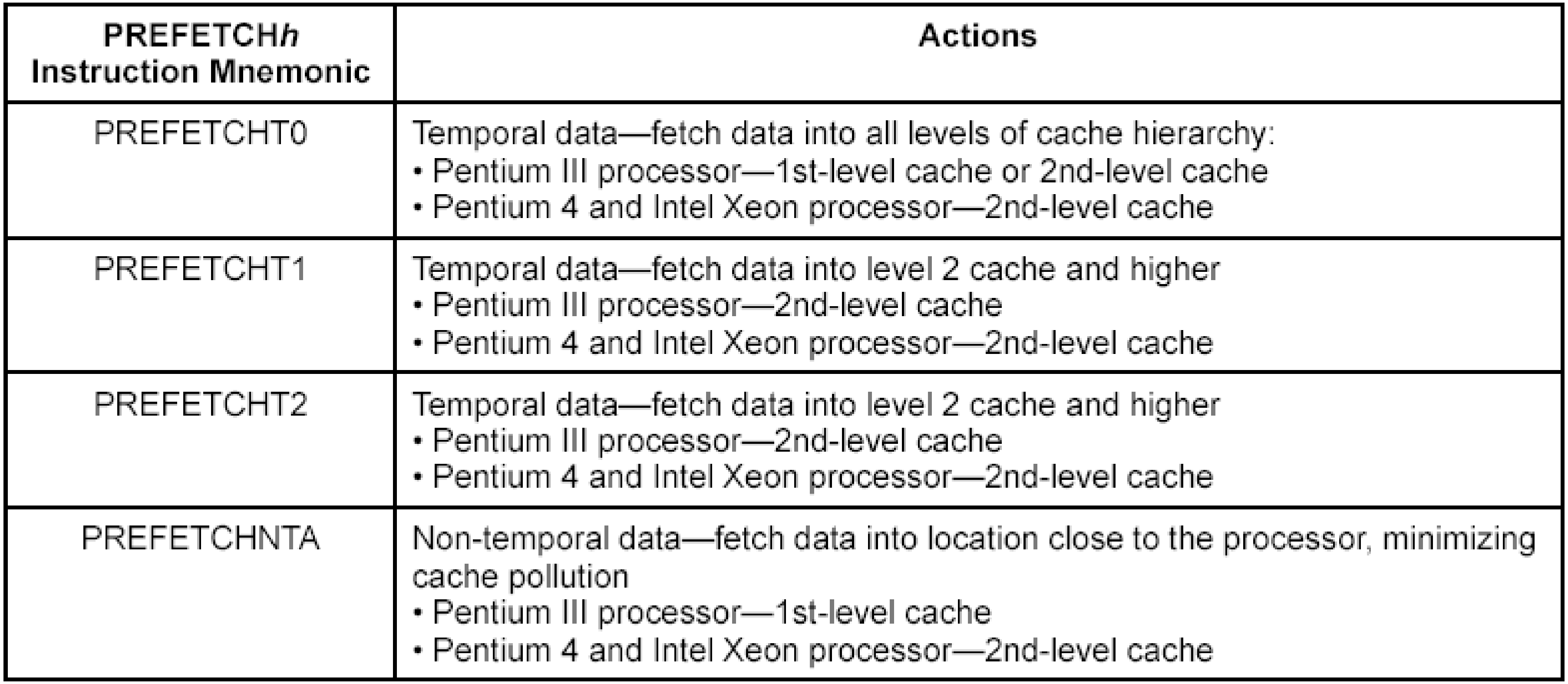


# Don’t pollute the cache

The SSE non-temporal store instructions minimize cache pollutions by treating the memory being accessed as the write combining (WC) type. If a program specifies a nontemporal store with one of these instructions and the destination region is mapped as cacheable memory, the processor will do the following:

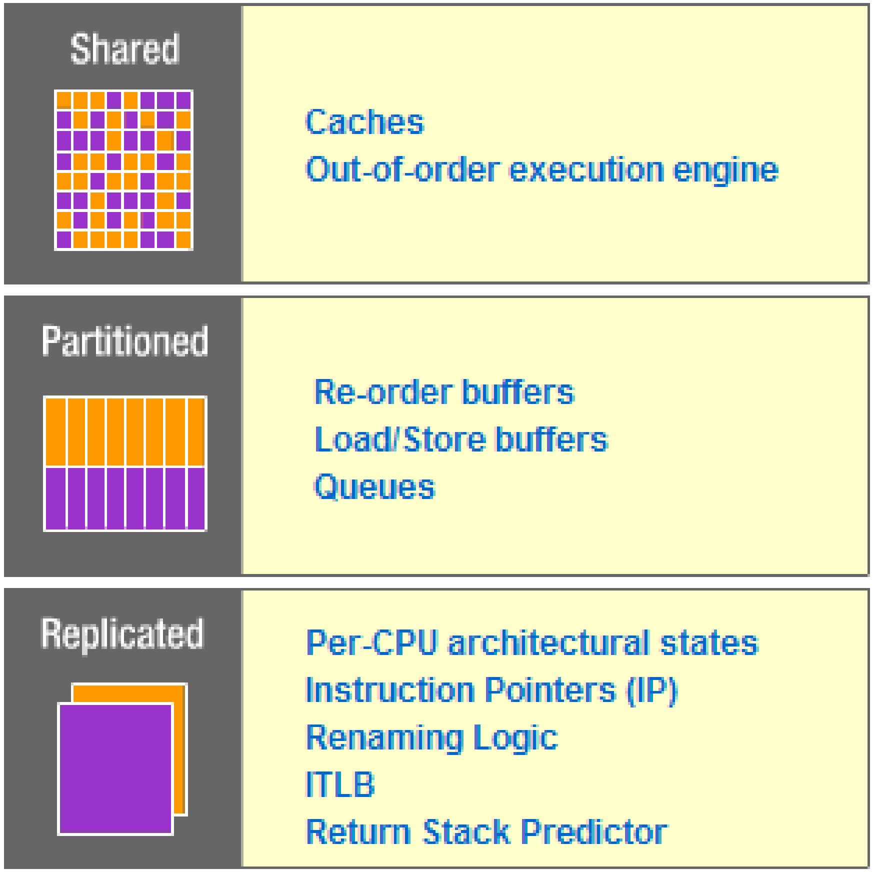
- If the memory location being written to is present in the cache hierarchy, the data in the caches is evicted.
- The non-temporal data is written to memory with weakly ordered semantics.

# Prefetch into the cache



| PREFETCH*h*<br>Instruction Mnemonic | Actions |
|---|---|
| PREFETCHT0 | Temporal data—fetch data into all levels of cache hierarchy:<br>• Pentium III processor—1st-level cache or 2nd-level cache<br>• Pentium 4 and Intel Xeon processor—2nd-level cache |
| PREFETCHT1 | Temporal data—fetch data into level 2 cache and higher<br>• Pentium III processor—2nd-level cache<br>• Pentium 4 and Intel Xeon processor—2nd-level cache |
| PREFETCHT2 | Temporal data—fetch data into level 2 cache and higher<br>• Pentium III processor—2nd-level cache<br>• Pentium 4 and Intel Xeon processor—2nd-level cache |
| PREFETCHNTA | Non-temporal data—fetch data into location close to the processor, minimizing cache pollution<br>• Pentium III processor—1st-level cache<br>• Pentium 4 and Intel Xeon processor—2nd-level cache |

# Hyperthreading



Shared

Caches
Out-of-order execution engine

Partitioned

Re-order buffers
Load/Store buffers
Queues

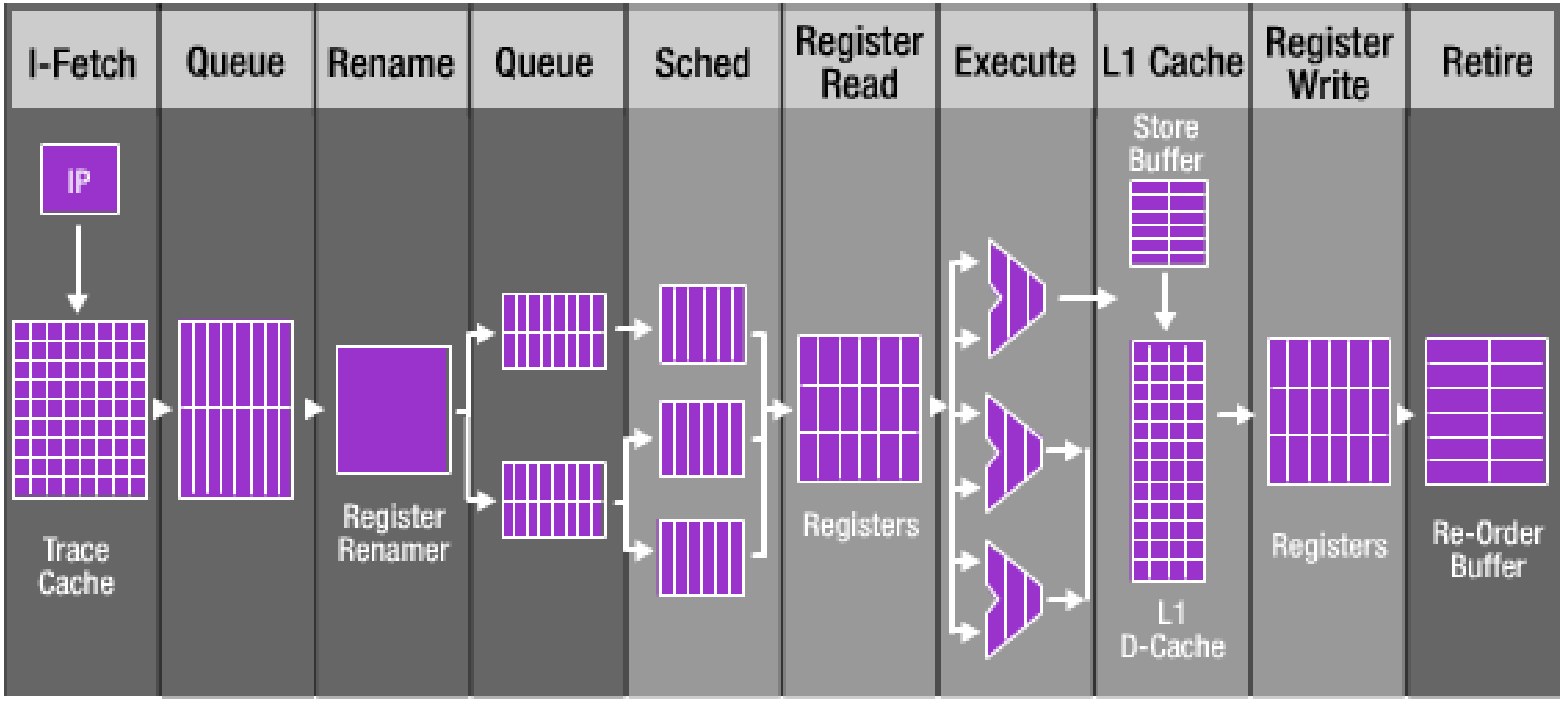
Replicated

Per-CPU architectural states
Instruction Pointers (IP)
Renaming Logic
ITLB
Return Stack Predictor

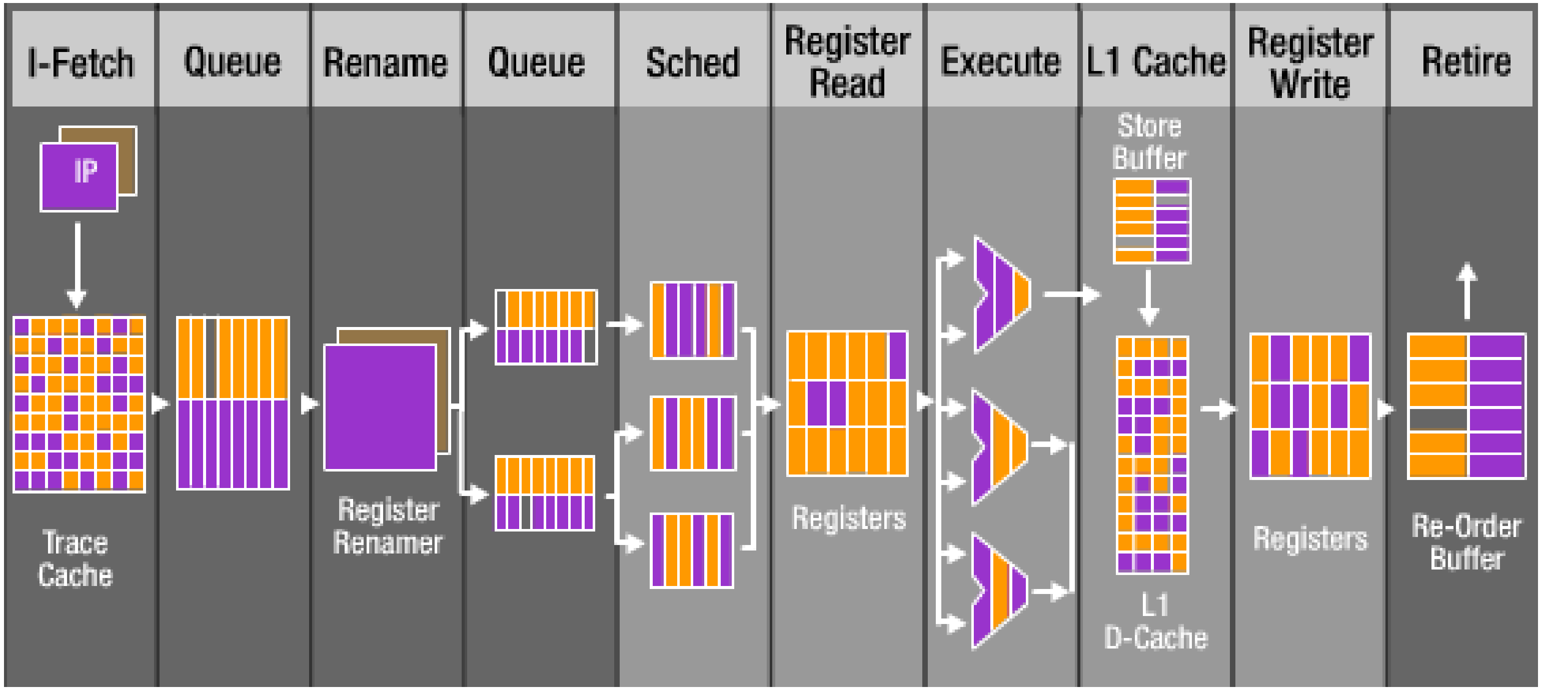
Hyperthreading (initially referred to as Simultaneous Multi-threading or SMT) allows for a single *physical* processor to appear to the operating system as two *logical* processors. The operating system doesn't know the difference and feeds threads to each as if they were indeed separate physical processors. Exploits the same *latency hiding* idea as the Denelcor HEP and Tera, but only two threads, not around a hundred.

A feature of high-end Pentium 4s

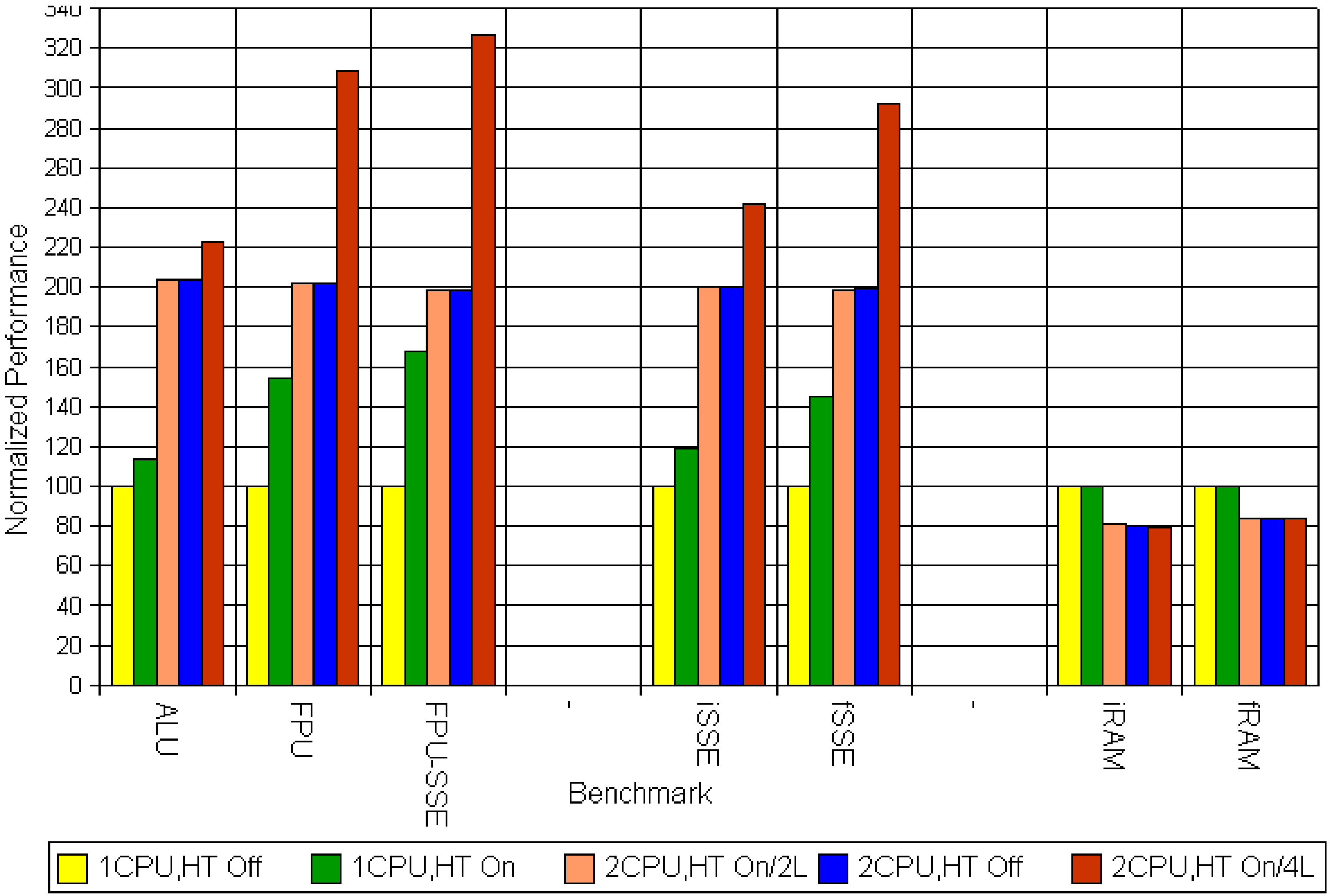
# Hyperthreading off



# Hyperthreading on



# Hyperthreading performance



# Hyperthreading performance

- Dell Precision 530 Workstation, with two 2.4GHz Xeon processors.

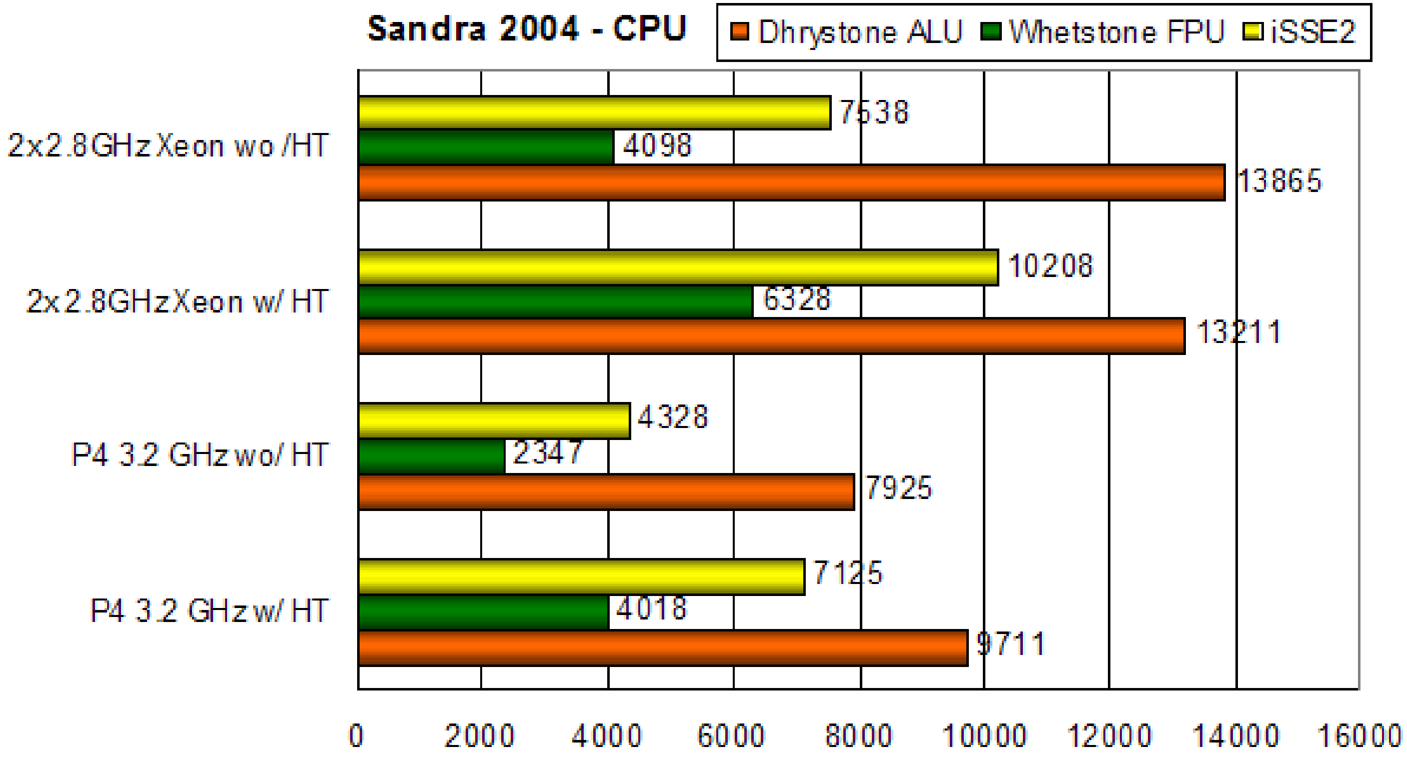
- CPU Arithmetic Benchmark
  - ALU Dhrystone ALU MIPS
  - FPU Whetstone FPU MFLOPS
  - FPU-SSE Whetstone iSSE2 MFLOPS

- CPU Multi-Media Benchmark
  - iSSE Integer iSSE2 it/s
  - fSSE Floating-Point iSSE2 it/s

- Memory Bandwidth Benchmark
  - iRAMRAM Bandwidth Int Buffered iSSE2 MB/s
  - fRAMRAM Bandwidth Float Buffered iSSE2 MB/s

http://home.insightbb.com/~george/Hyperthreading/Hyperthreading.html
http://www.sisoftware.net/

# Hyperthreading performance



http://www.2cpu.com/articles/42_3.html