

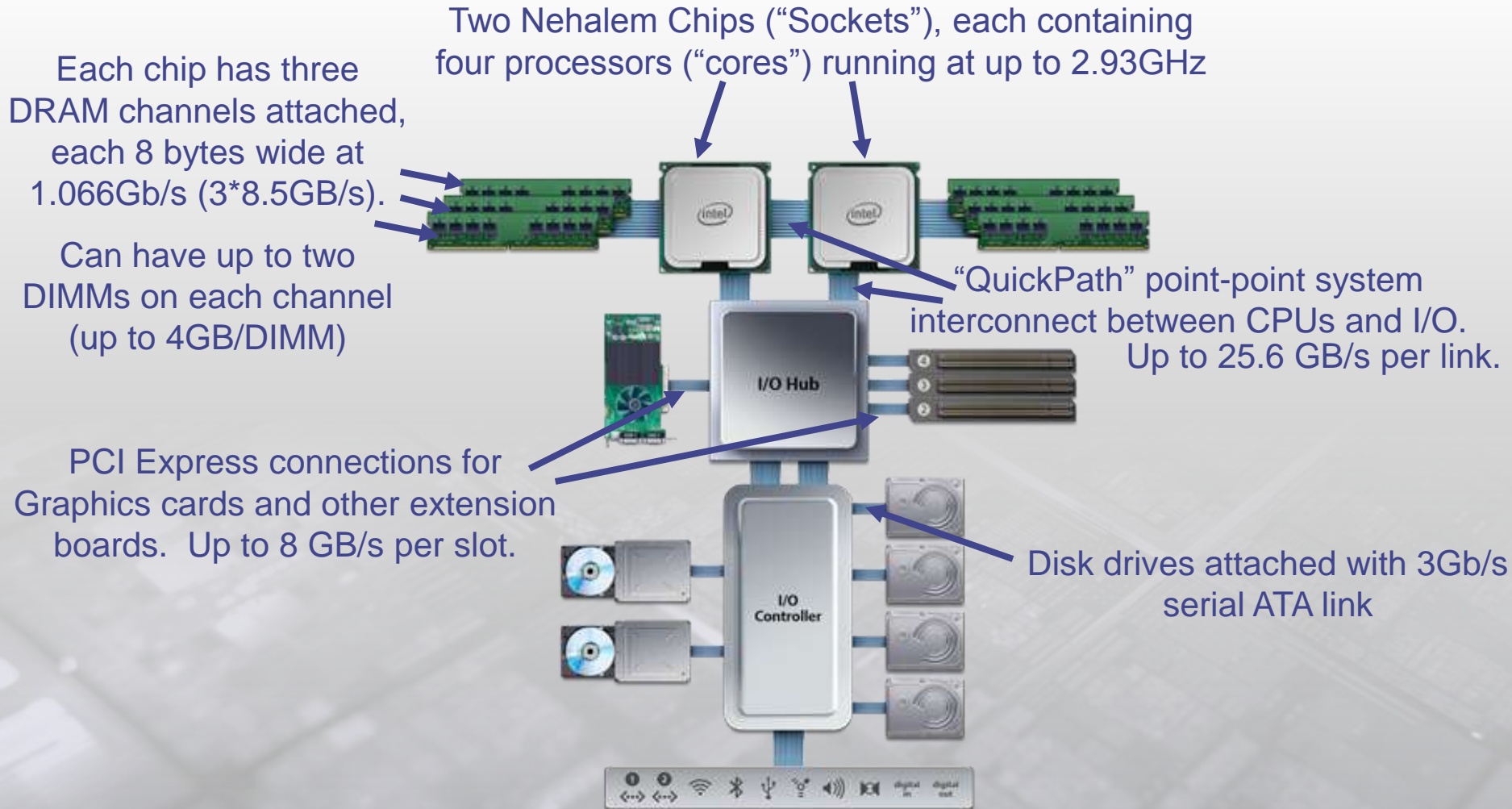
Putting it all together: Intel Nehalem

<http://www.realworldtech.com/page.cfm?ArticleID=RWT040208182719>

Intel Nehalem

- Review entire term by looking at most recent microprocessor from Intel
- Nehalem is code name for microarchitecture at heart of Core i7 and Xeon 5500/5600 series server chips
- First released at end of 2008

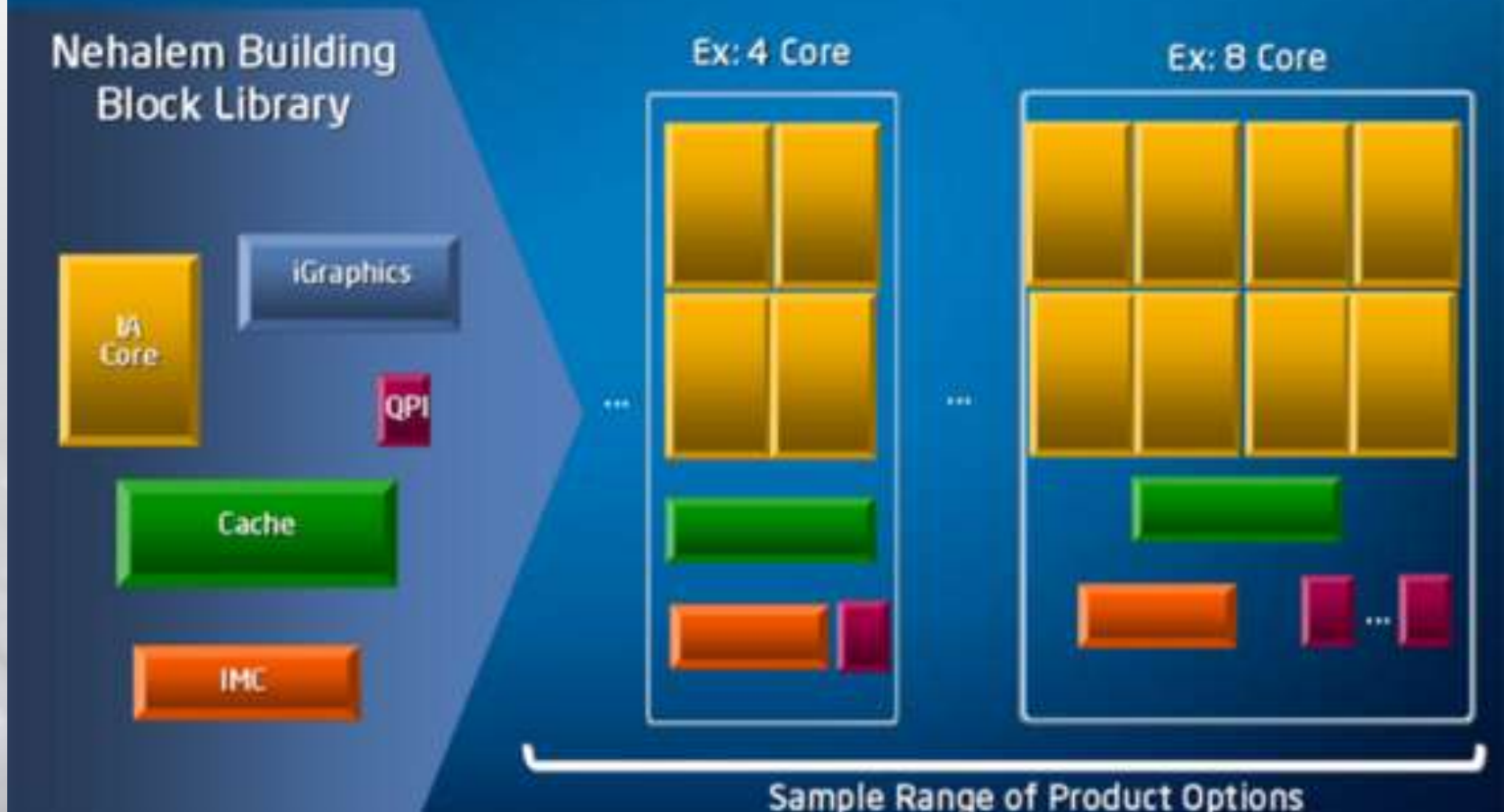
Nehalem System Example: Apple Mac Pro Desktop 2009



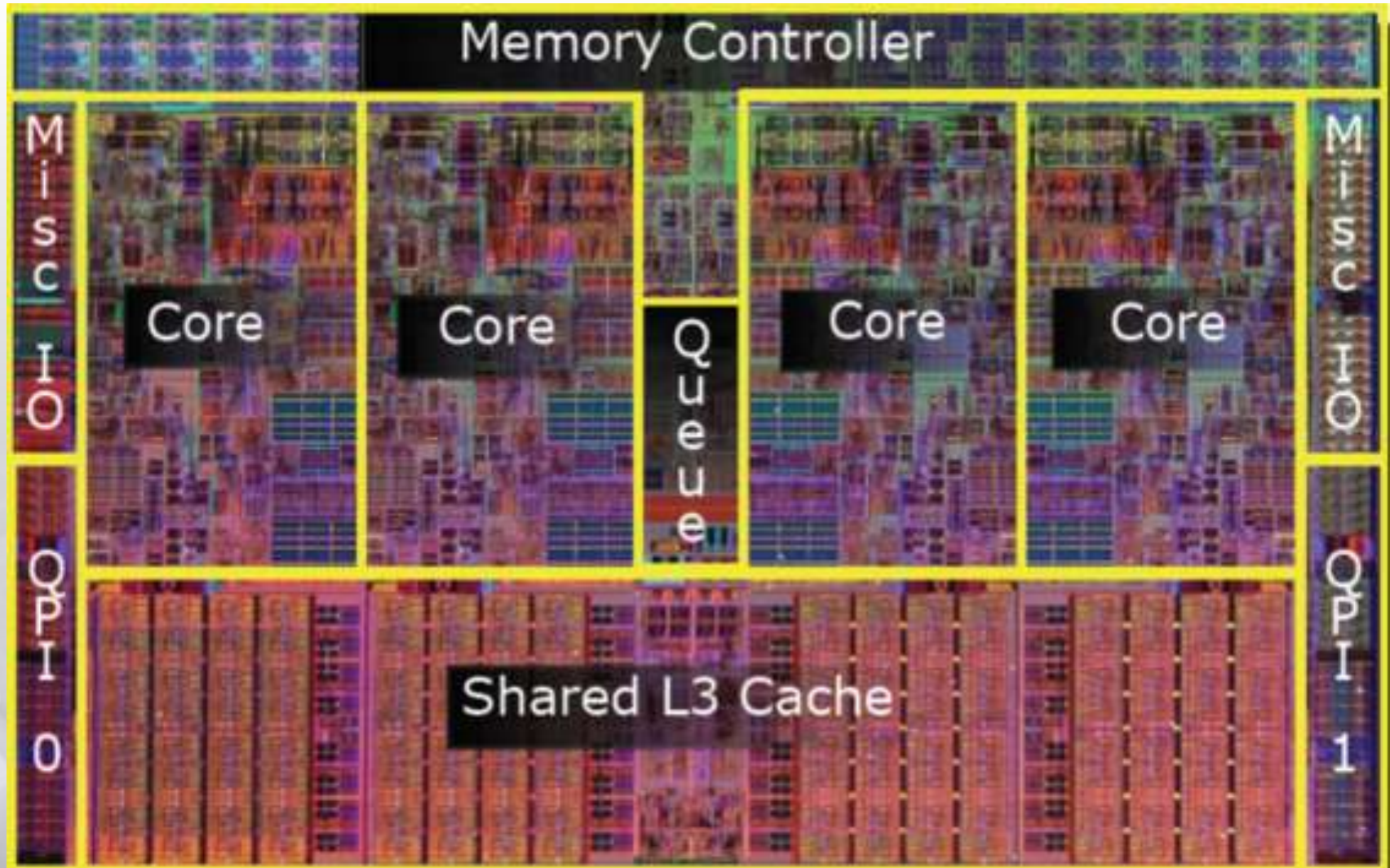
Slower peripherals (Ethernet, USB, Firewire, WiFi, Bluetooth, Audio)

Building Blocks to support “Family” of processors

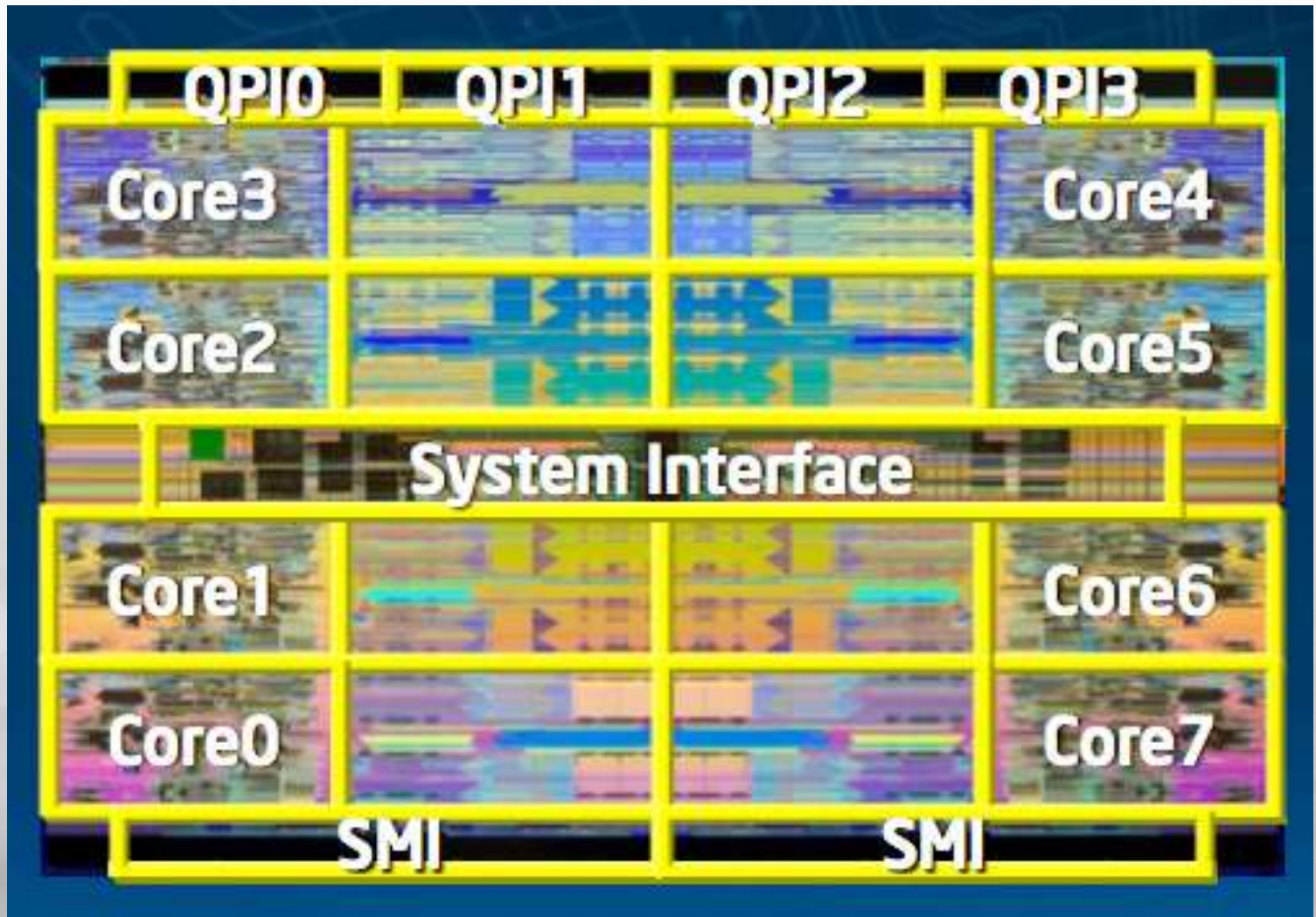
Nehalem Design Scalable Via Modularity



Nehalem Die Photo



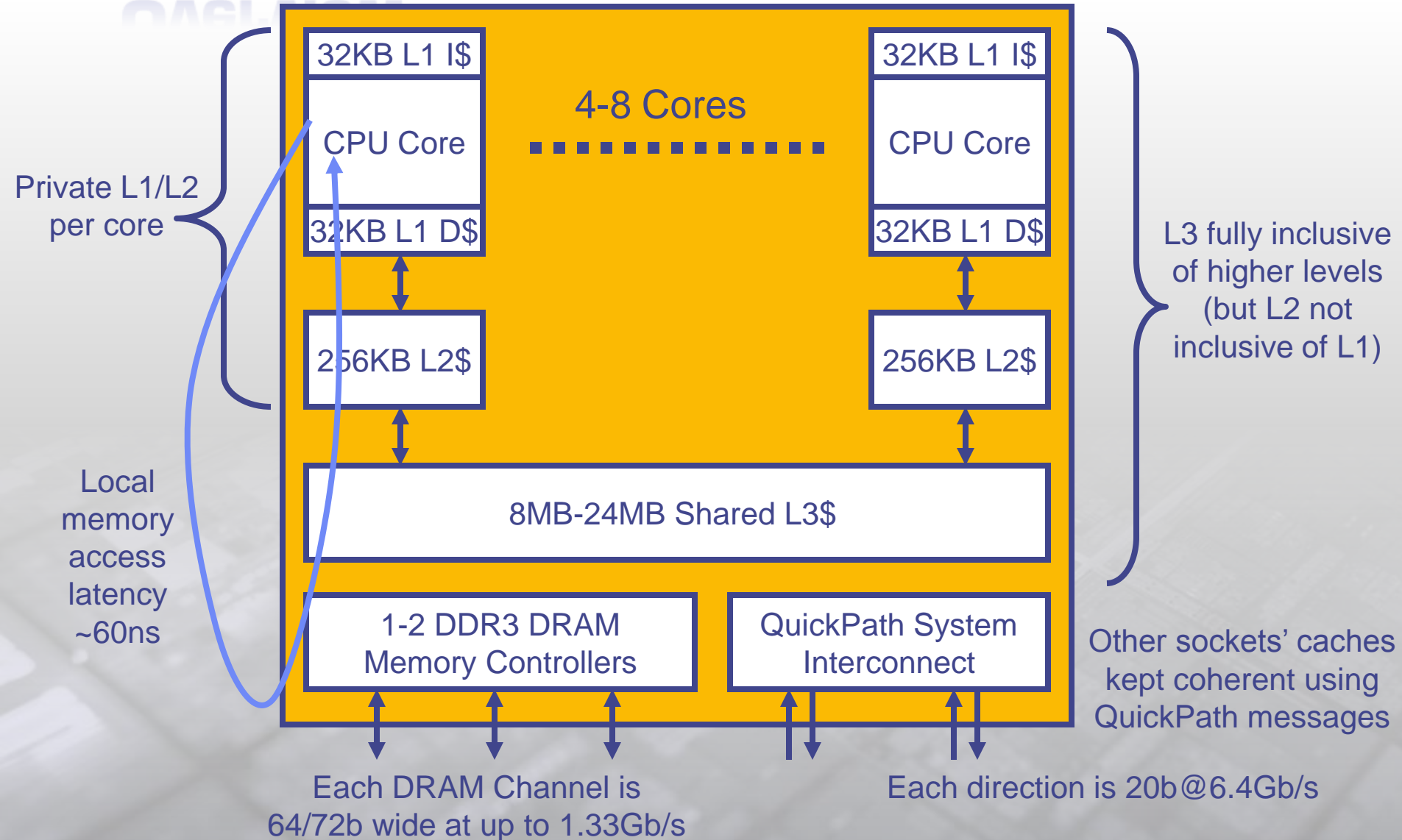
Nehalem-EX (8-core)





MEMORY SYSTEM

Nehalem Memory Hierarchy Overview



Cache Hierarchy Latencies

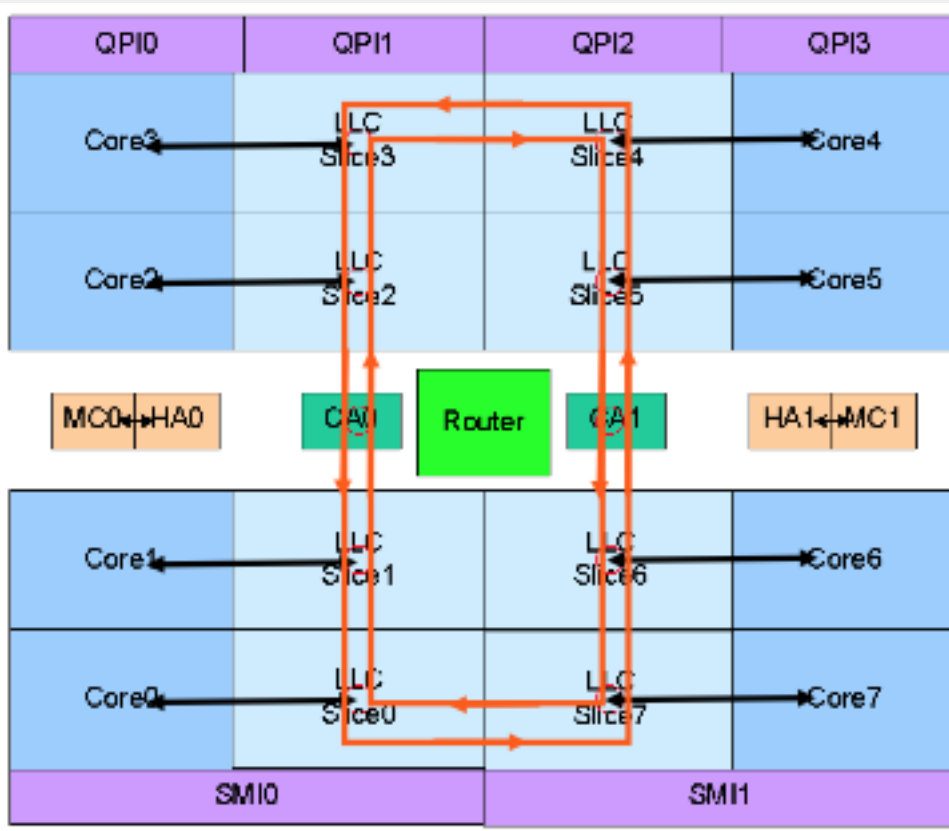
- L1 32KB 8-way, latency 4 cycles
- L2 256KB 8-way, latency <12 cycles
- L3 8MB shared, 16-way, latency 30-40 cycles (4 core system)
- L3 24 MB shared, 24-way, latency 30-60 cycles (4 core system)
- DRAM, latency ~180-200 cycles

Nehalem Virtual Memory Details

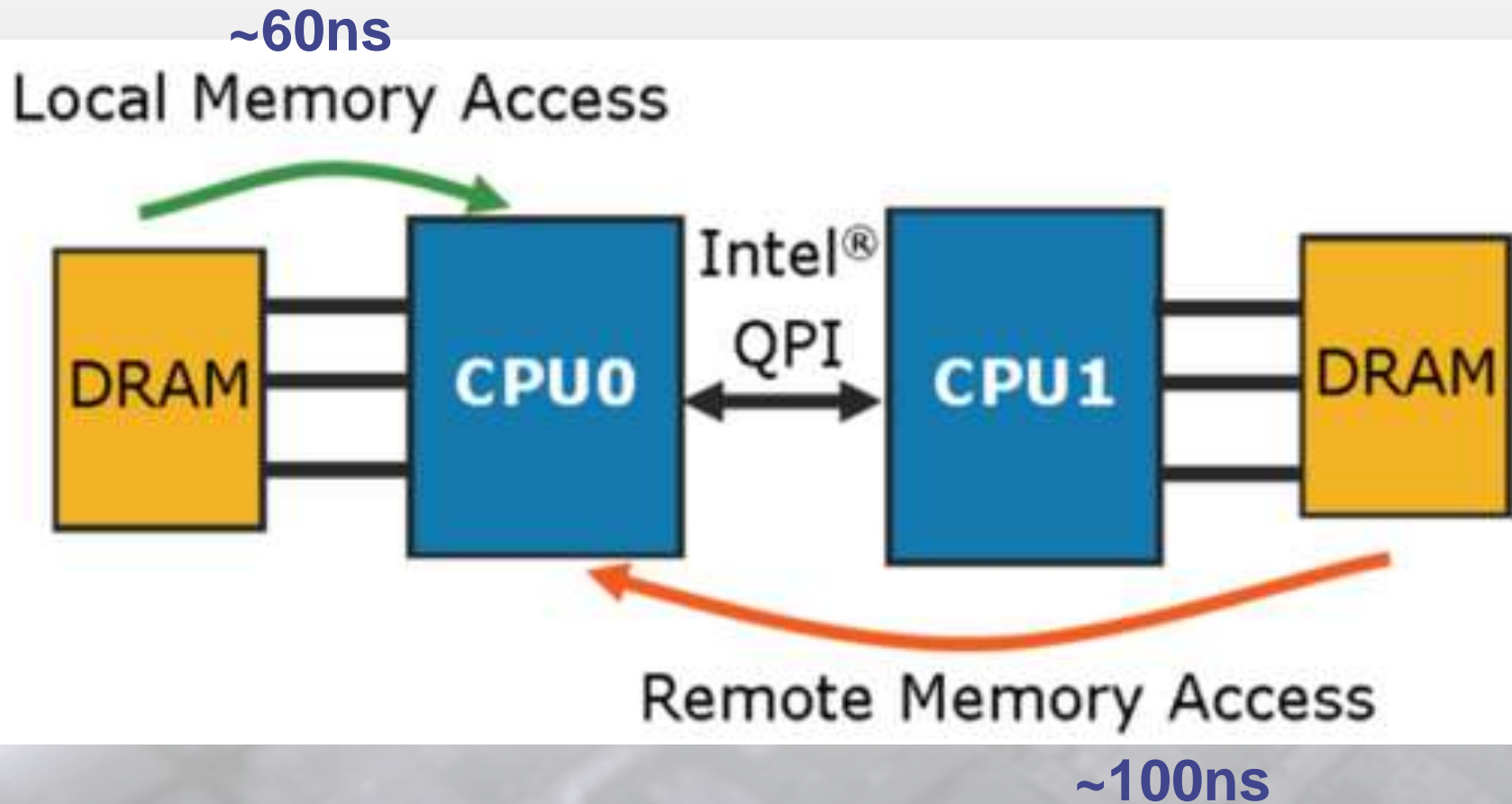
- Implements 48-bit virtual address space, 40-bit physical address space
- Two-level TLB
- I-TLB (L1) has shared 128 entries 4-way associative for 4KB pages, plus 7 dedicated fully-associative entries per SMT thread for large page (2/4MB) entries
- D-TLB (L1) has 64 entries for 4KB pages and 32 entries for 2/4MB pages, both 4-way associative, dynamically shared between SMT threads
- Unified L2 TLB has 512 entries for 4KB pages only, also 4-way associative

Cache coherency: on-chip

- on-chip snoop cache coherence
 - MESIF – (M)odified, (E)xclusive, (S)hared, (I)nvalid, (F)ordwarder
- Ring as interconnection network
 - Up to 250GB/sec

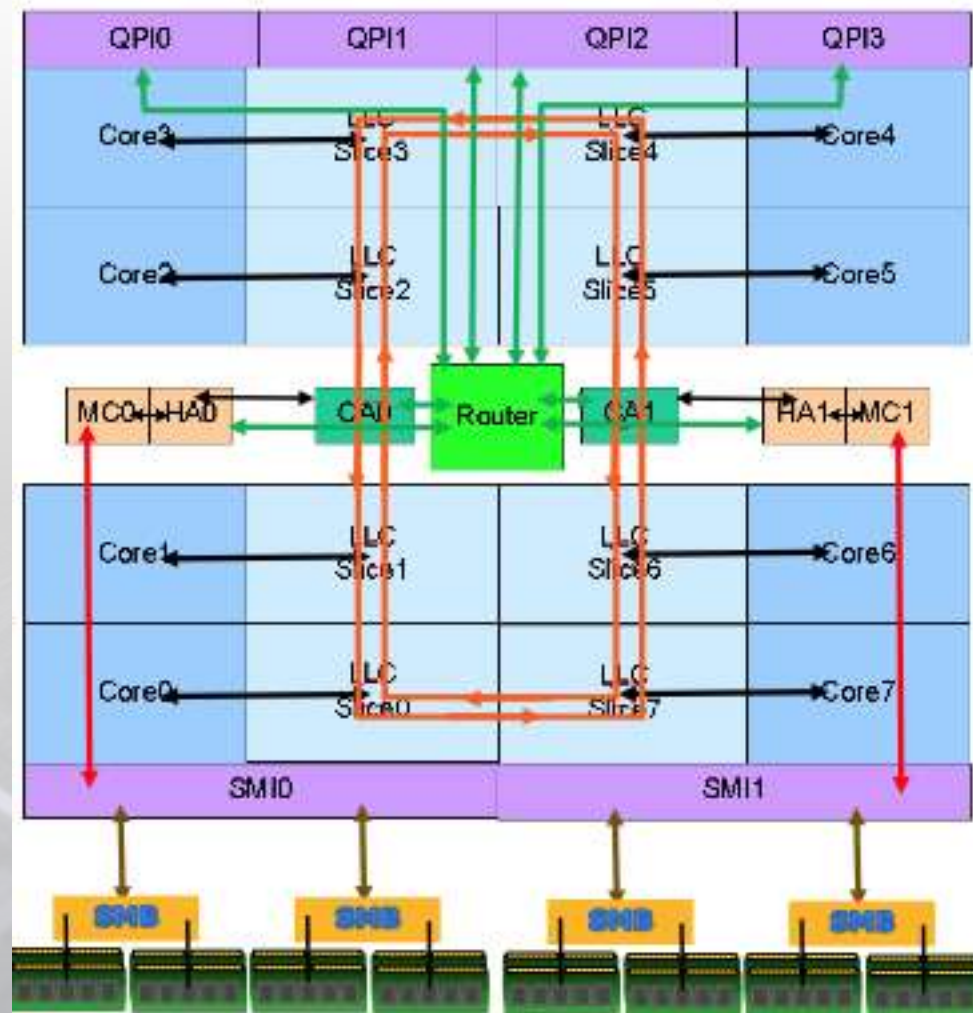


All Sockets can Access all Data



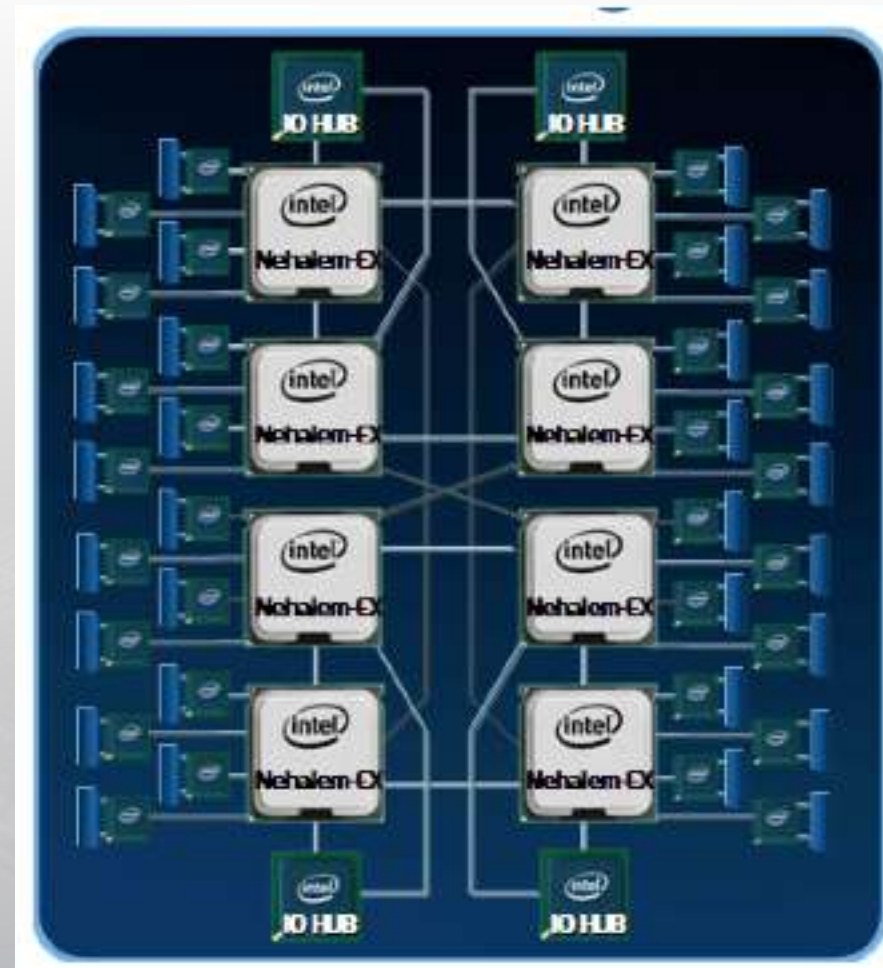
Remote cache requests

- Directory like scheme to keep shared-memory view
- The router can connect up to 4 QPI
 - Complex topologies: twisted hypercube



Large Scale systems

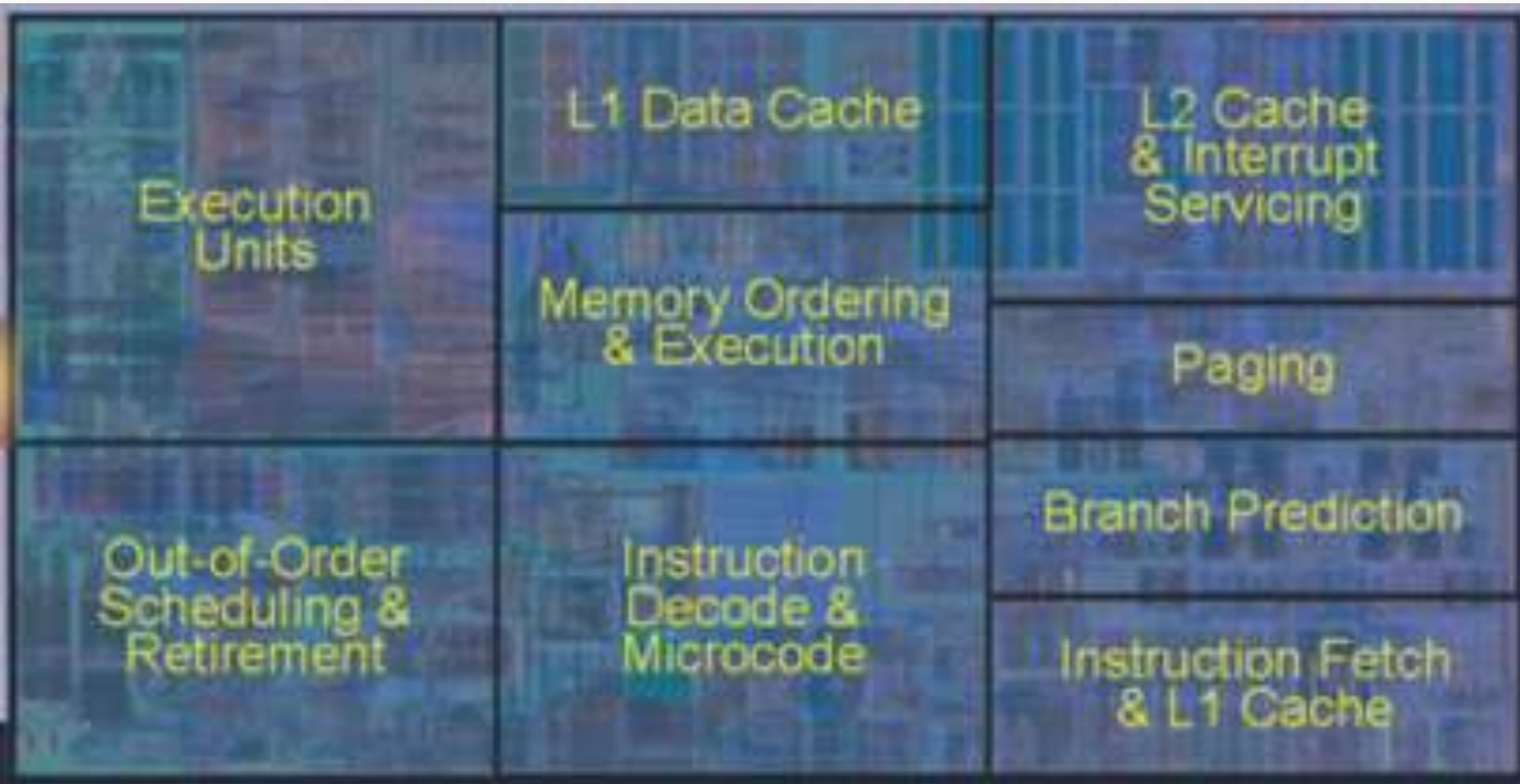
- Intel Architecture capable of QPI connected 8-Sockets / 128threads
- Max 2 QPI hops between two sockets

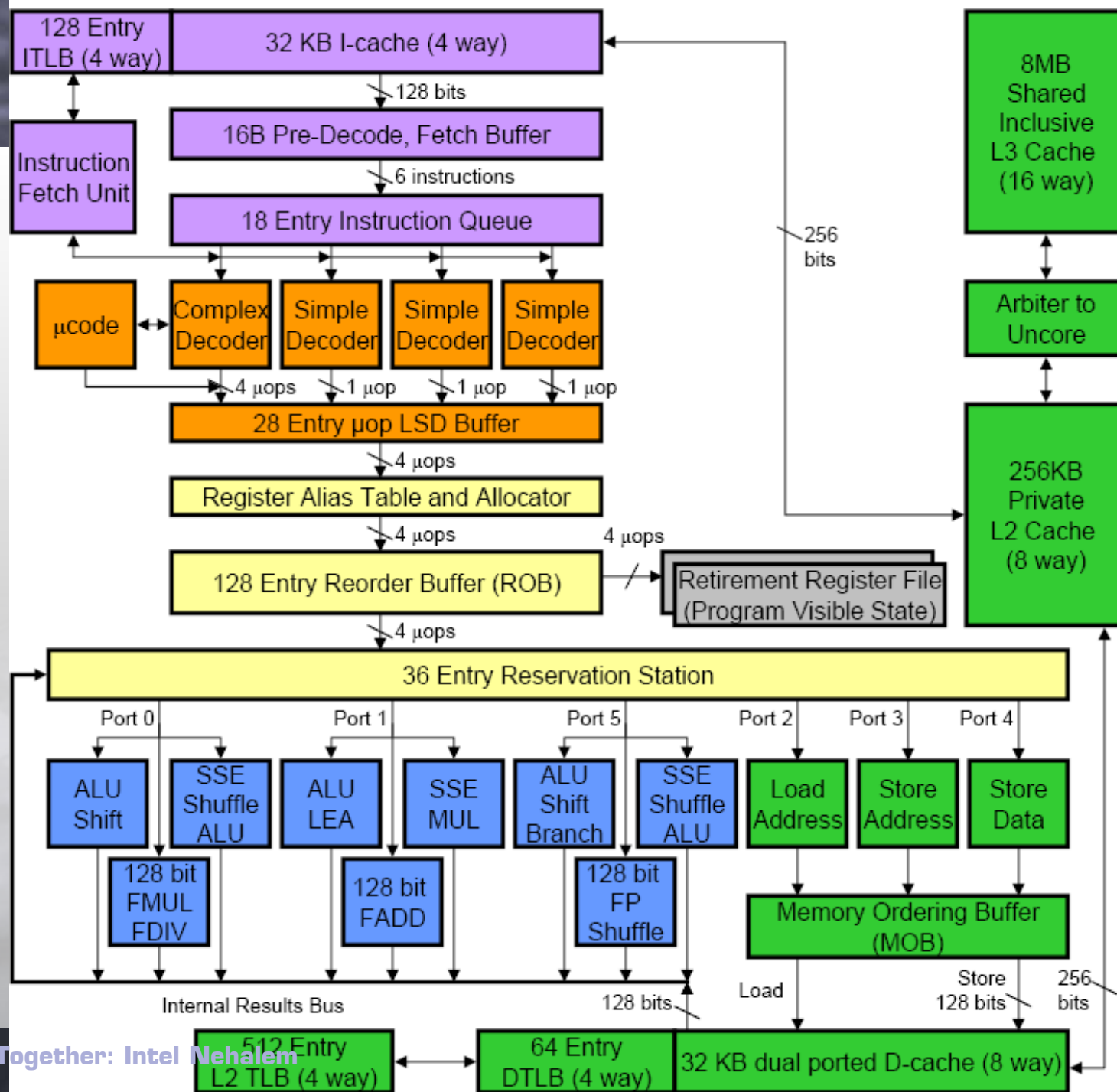




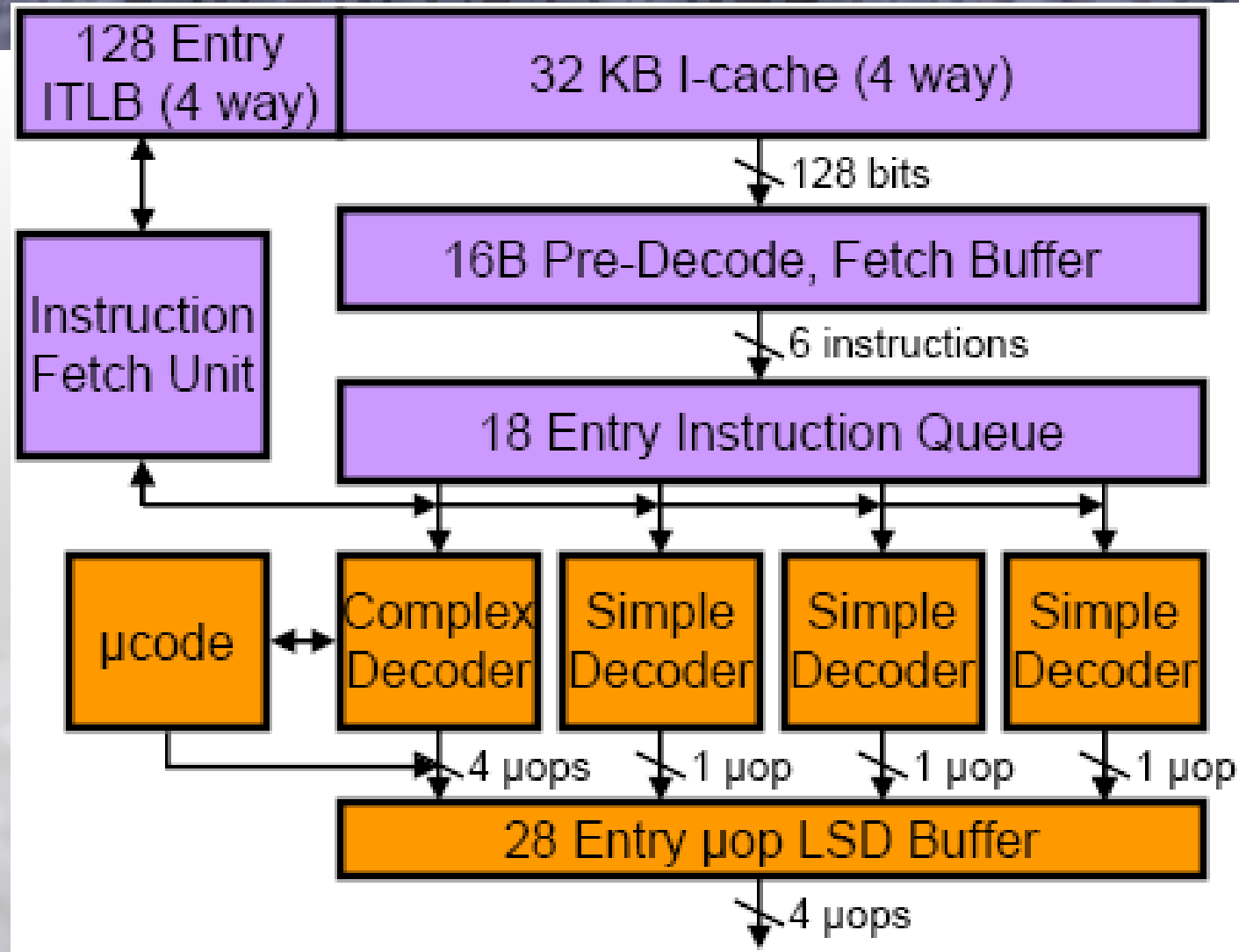
CORE ARCHITECTURE

Core Area Breakdown





Front-End Instruction Fetch & Decode



x86
instruction
bits

internal
 μ OP bits

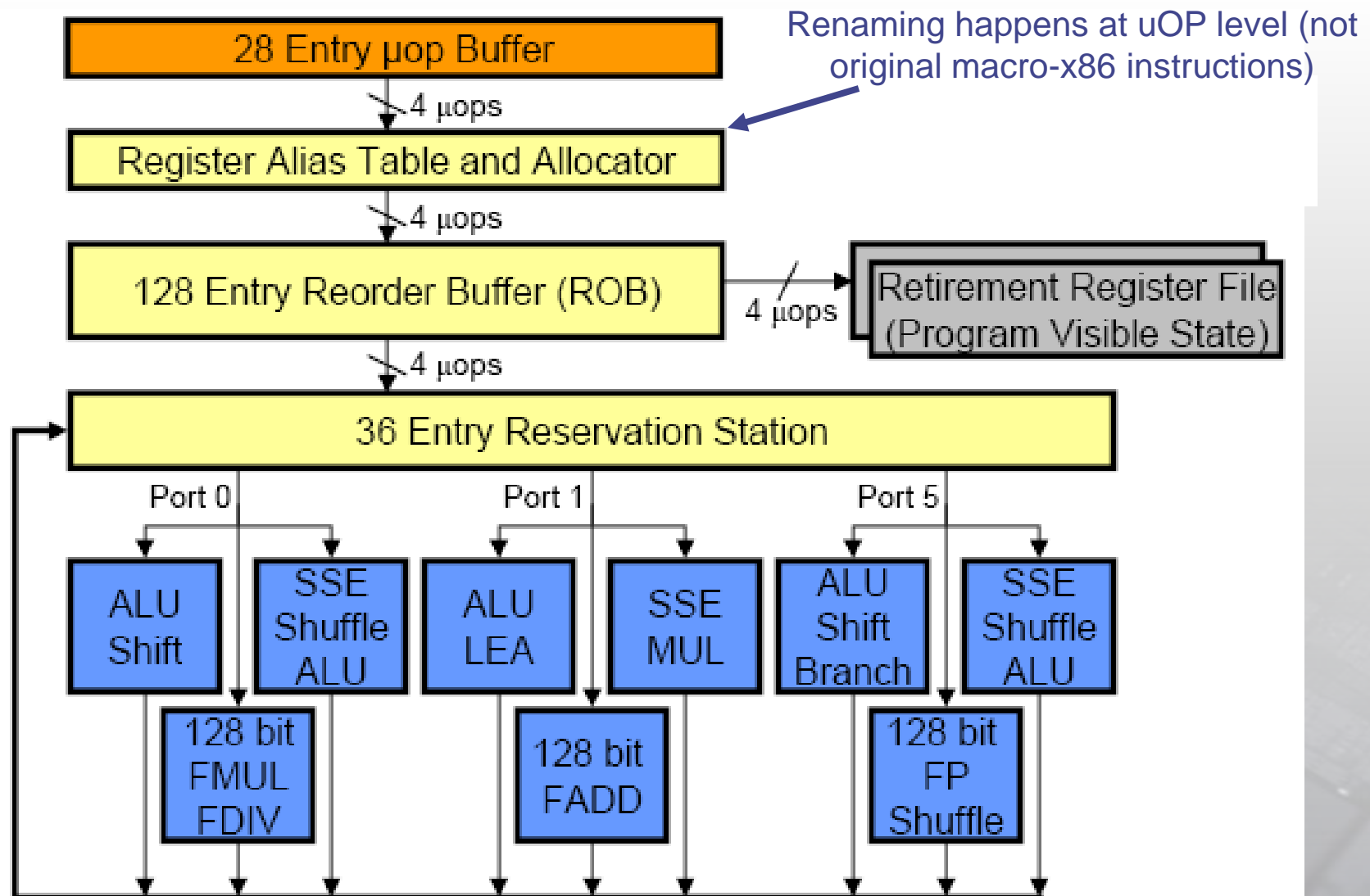
Front End I: Branch Prediction

- Part of instruction fetch unit
- Several different types of branch predictor
 - Details not public
- Loop count predictor
 - How many backwards taken branches before loop exit
 - (Also predictor for length of microcode loops, e.g., string move)
- Return Stack Buffer
 - Holds subroutine targets
 - Renames the stack buffer so that it is repaired after mispredicted returns

Front End II: x86 Decoding

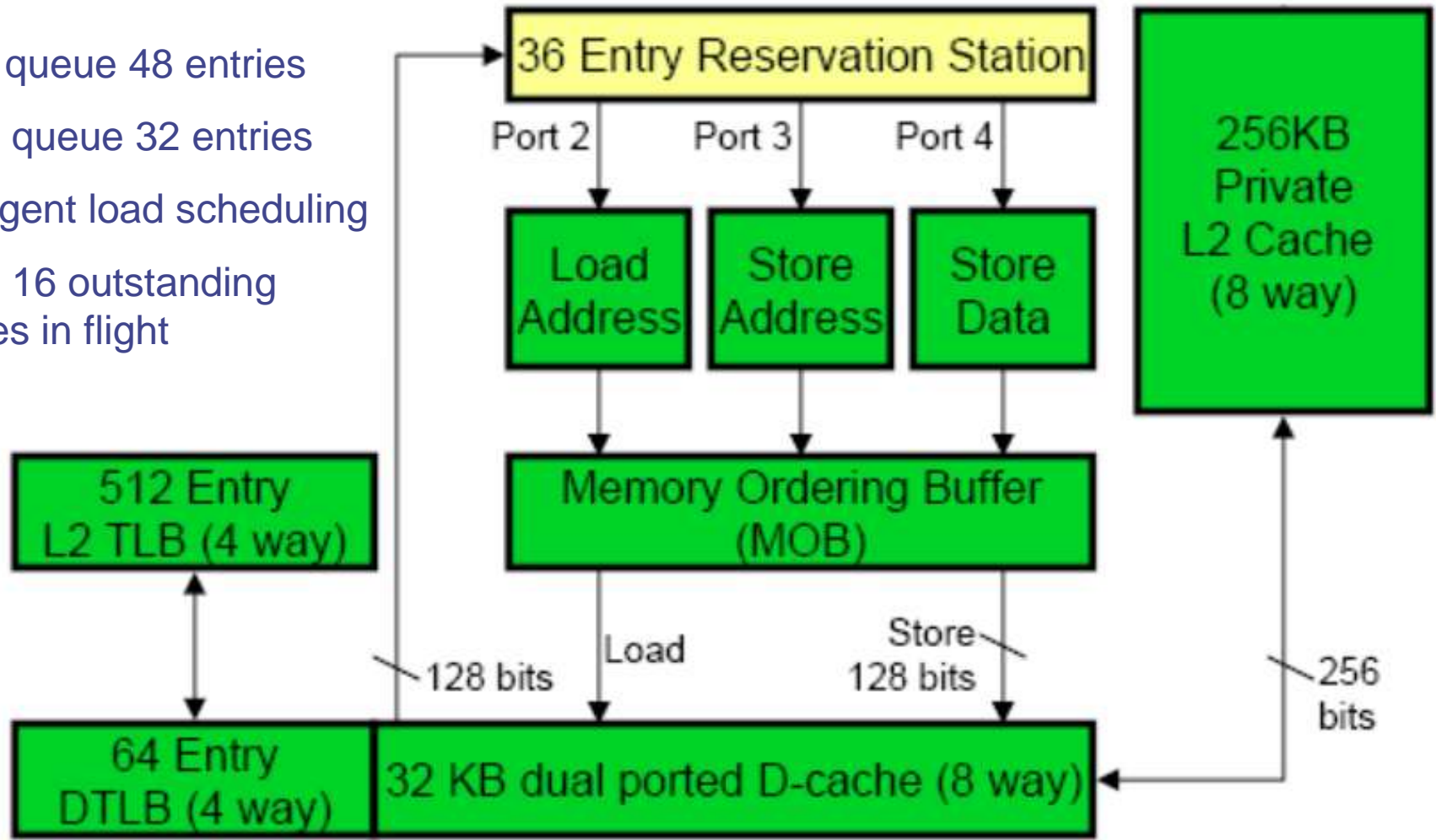
- Translate up to 4 x86 instructions into uOPS each cycle
- Only first x86 instruction in group can be complex (maps to 1-4 uOPS), rest must be simple (map to one uOP)
- Even more complex instructions, jump into microcode engine which spits out stream of uOPS

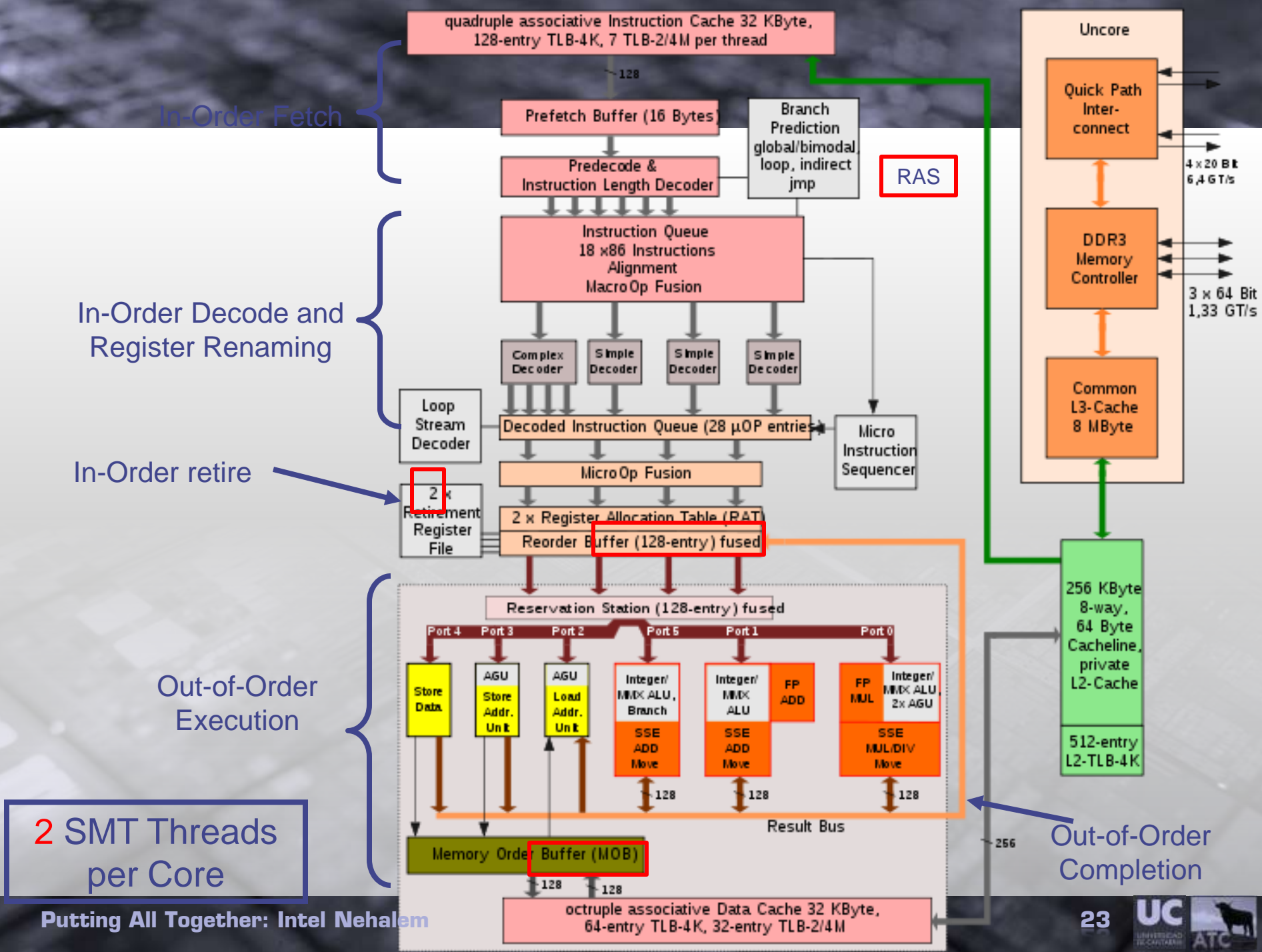
BE: Out-of-Order Execution Engine



LSQ

Load queue 48 entries
Store queue 32 entries
Intelligent load scheduling
Up to 16 outstanding misses in flight





SMT with OoO Execution Core

- Reorder buffer (remembers program order and exception status for in-order commit) has 128 entries divided statically and equally between both SMT threads
 - ICOUNT (guess?) fetch policy
 - (BRCOUNT, MISSCOUNT, etc..)
- Reservation stations (instructions waiting for operands for execution) have 36 entries competitively shared by SMT threads
- Load & Store Queues Statically partitioned and equally between both SMT threads

Final

- "Cualquier tecnología suficientemente avanzada es indistinguible de la magia" Arthur C. Clarke