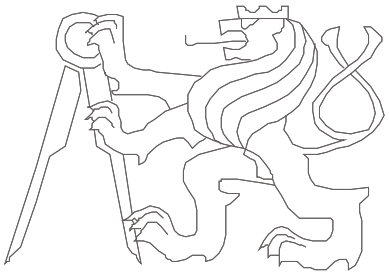
Advanced Computer Architectures

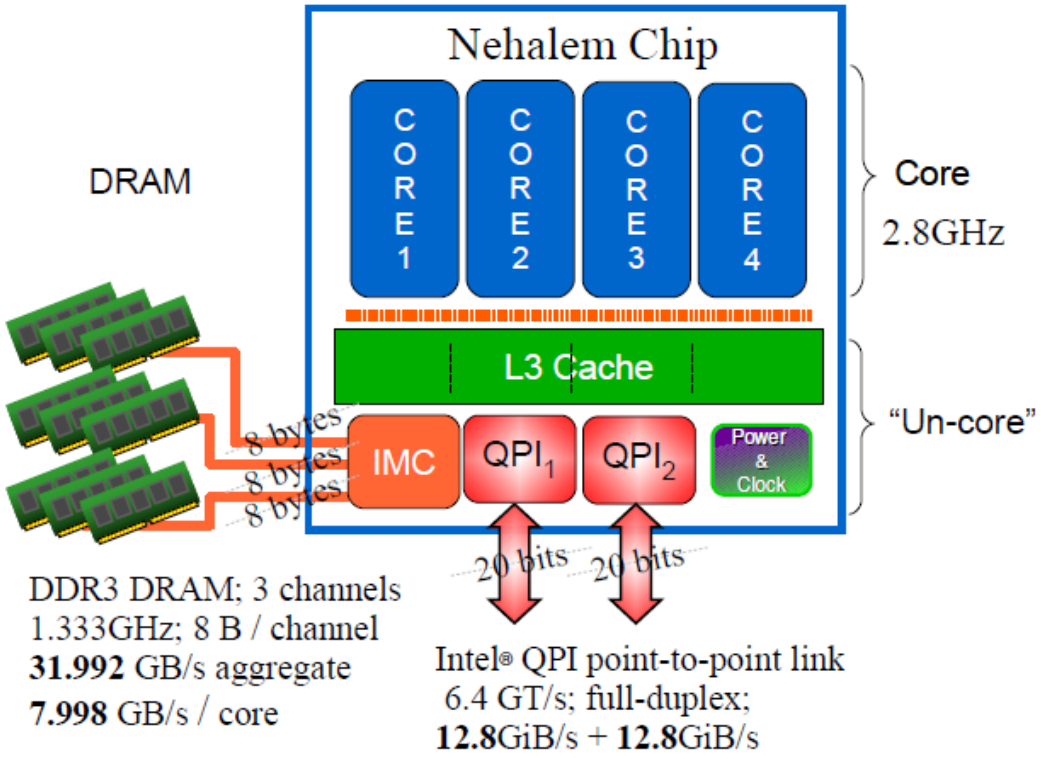
Theory in practice...

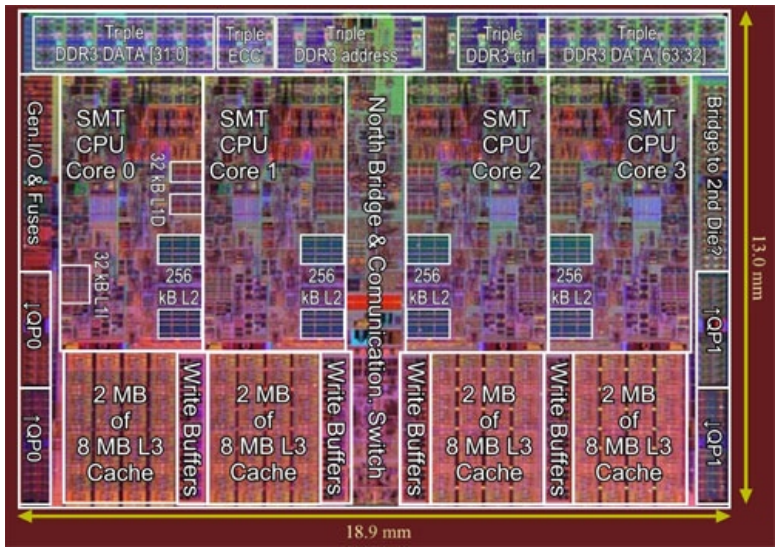
From Intel Nehalem Core i5 a Core i7 to AMD Zen



Czech Technical University in Prague, Faculty of Electrical Engineering
Slides authors: Pavel Píša, Michal Štepanovský

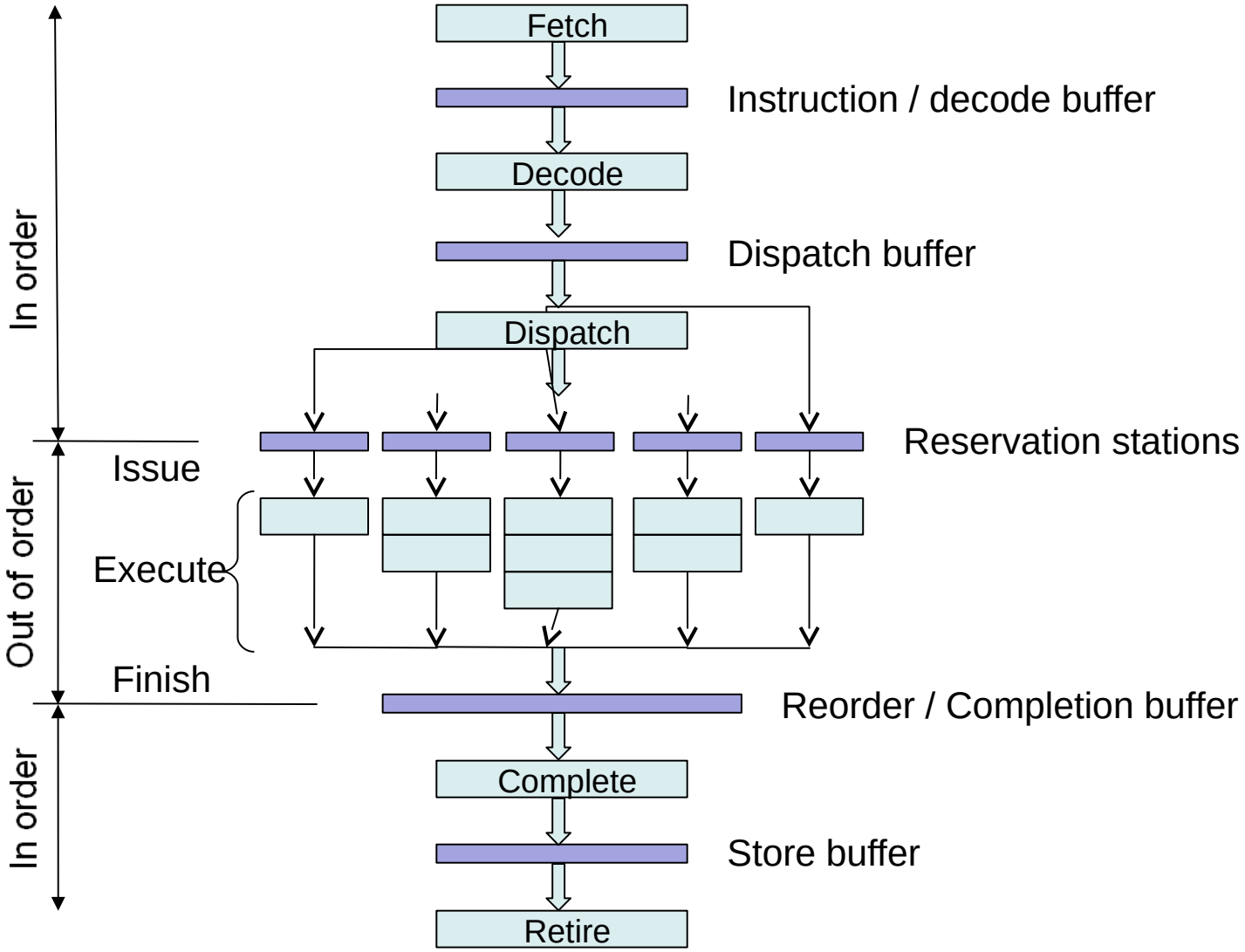
Intel Nehalem



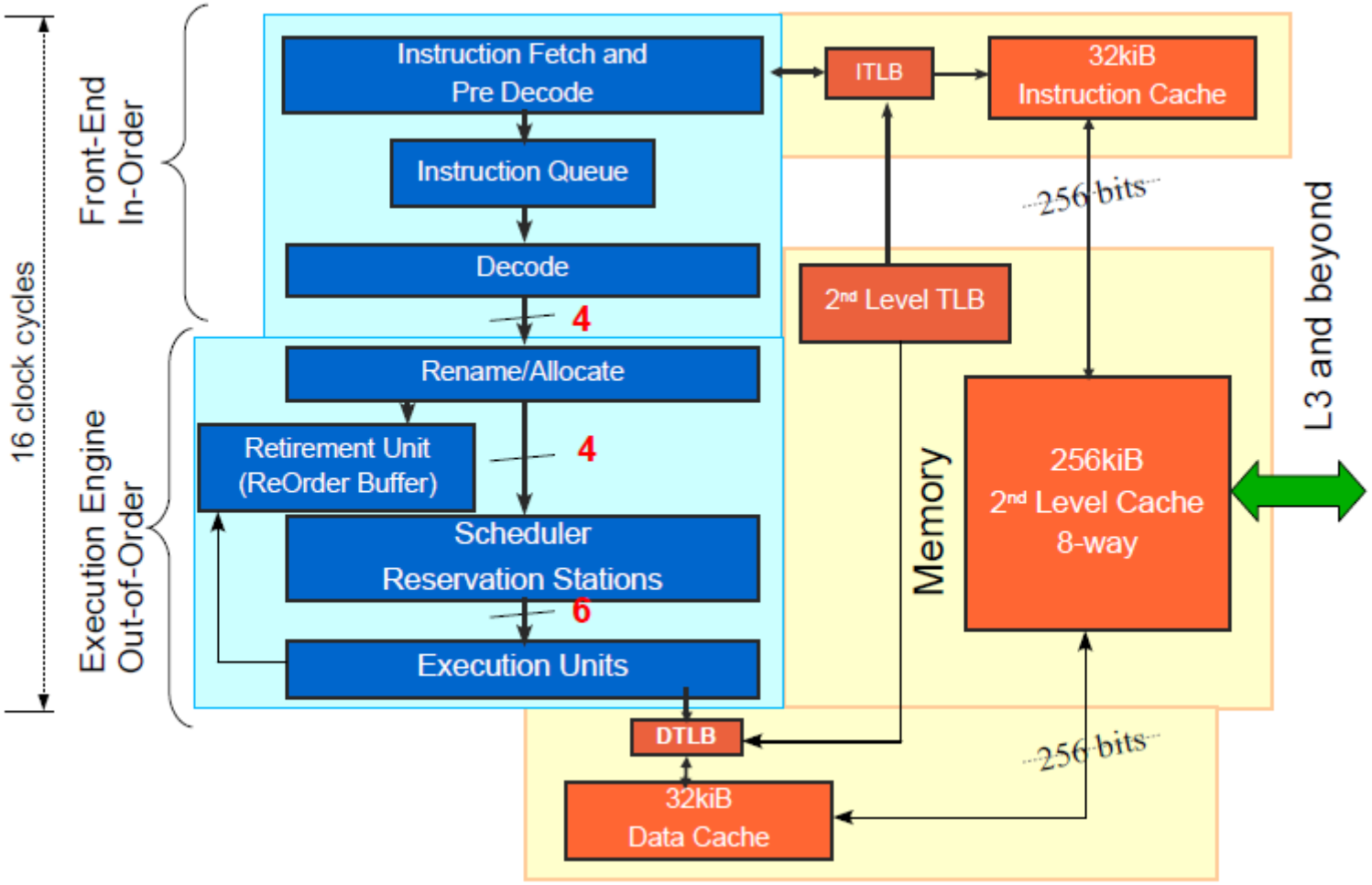


- core / un-core
- UIU: Un-Core Interface Unit (switch connecting the 4 cores to the 4 L3 cache segments, the IMC and QPI ports),
- IMC: 1 integrated memory controller with 3 DDR3 memory channels,
- QPI: 2 Quick-Path Interconnect ports
- Support circuits for cache coherency, power management, performance monitoring

Do you remember?

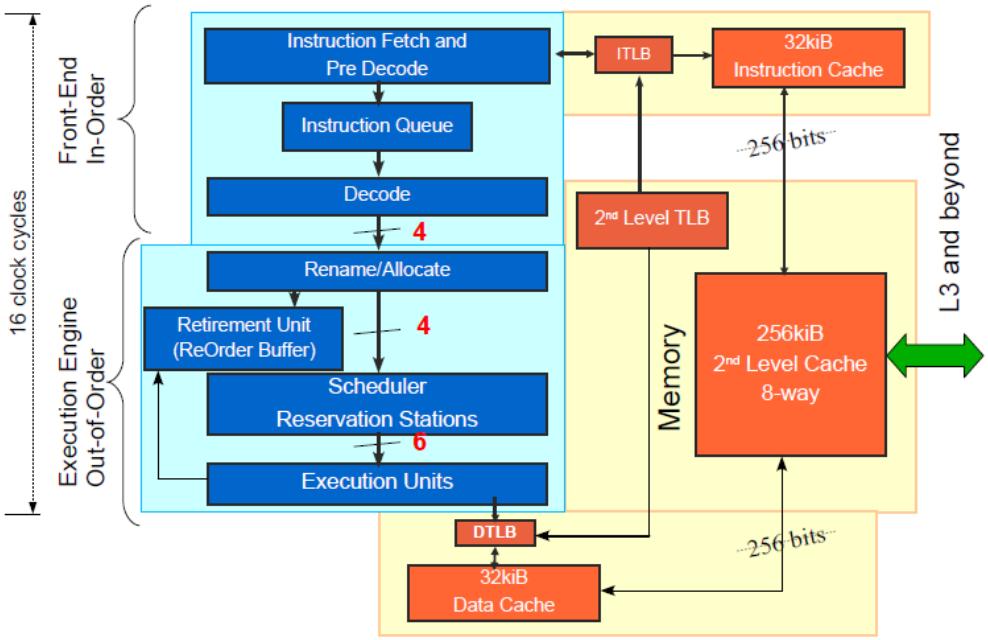


Nehalem Core Pipeline



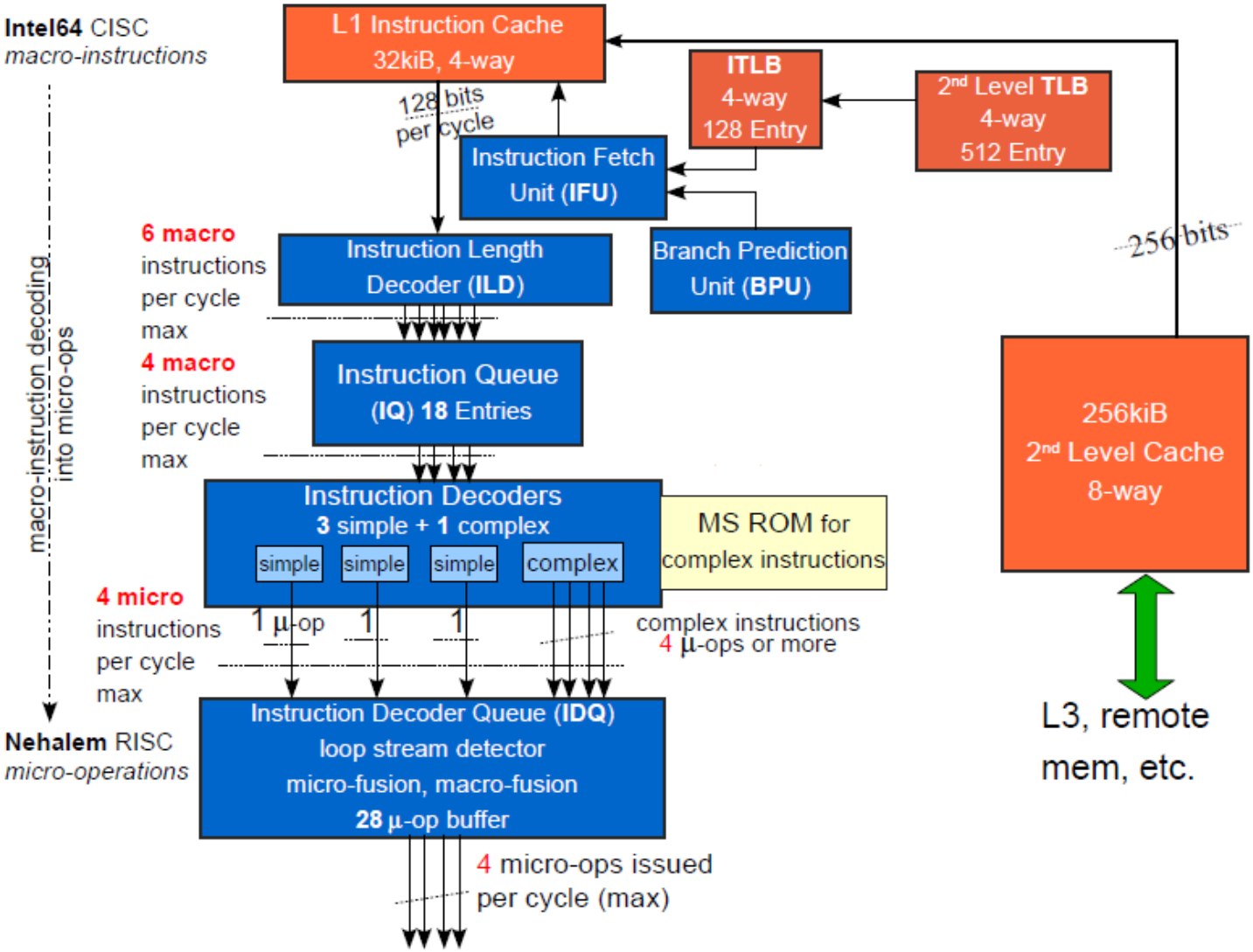
Nehalem Core Pipeline

- 14-stage core pipeline
- Front-End Pipeline (FEP)
  - translate macroinstructions to microinstructions
- Execution Engine (EE)
  - dynamic scheduling and dispatch of microinstructions
- Retirement Unit (RU)
  - architectural processor state is updated in original/program ord



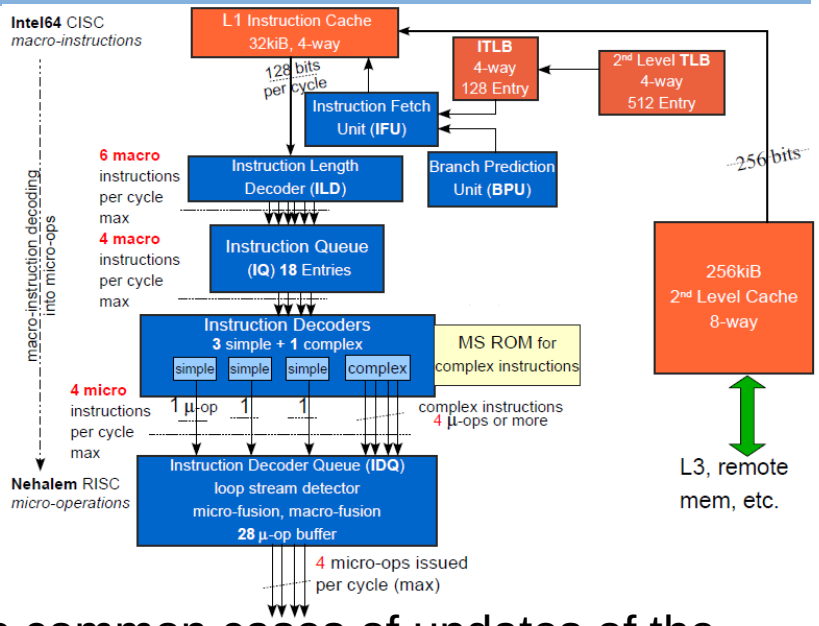
- TLB (Translation Lookaside Buffer) – realizes translation of virtual addresses to physical ones; caches entries from the page tables. Problem of the memory context switching overcome by in cooperation with … ASN (Address Space Number), (PGE flag), Huge Pages

Front-End Pipeline (FEP)



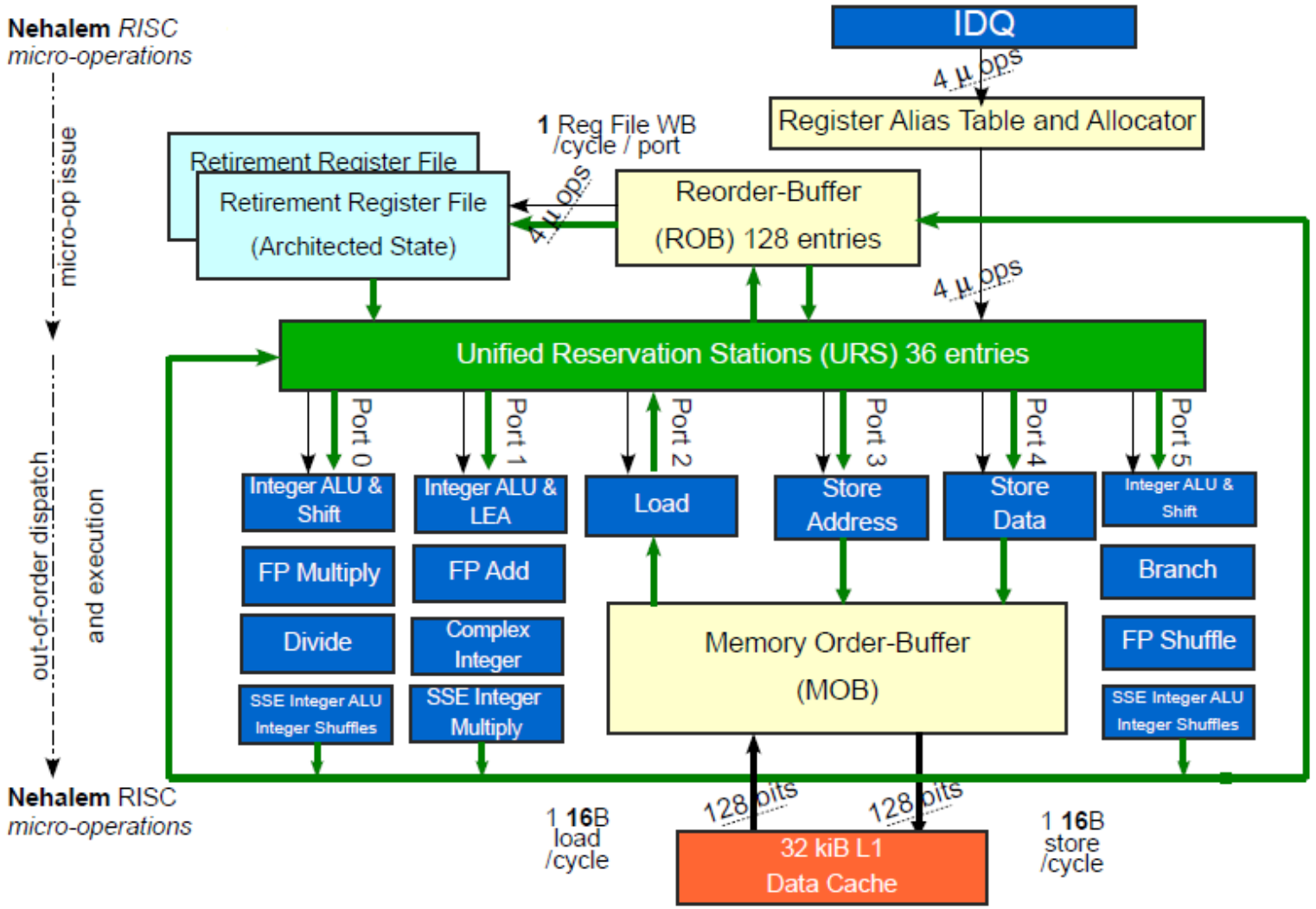
Front-End Pipeline (FEP) – Some Remarks

- every IDU (Instruction Decoding Unit) supports common cases of instructions flow (mikro-fusion, macro-fusion, stack pointer tracking)
- LSD (**Loop Stream Detection**) reduces penalties caused by unaligned instructions placement, LCP (Length Changing Prefixes) overhead, power consumption



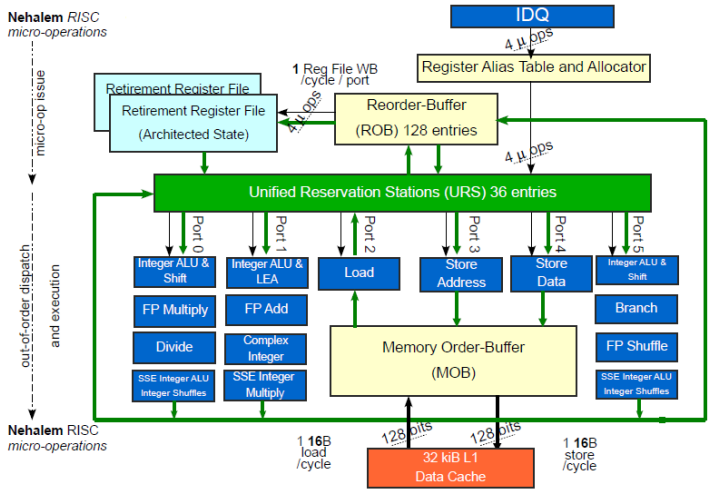
- SPT (**Stack Pointer Tracking**) implements common cases of updates of the Stack Pointer Register which are results of Stack manipulation instructions (push, pop, call, leave, ret) – these instructions required multiple micro-ops in the previous generations, it is single mikro-op now.
- **Mikro-fusion** coneverts multiple micro-operations of the same instruction ito single more complex fused microinstruction – reduces total „bit-toggling“
- **Macro-fusion** collects following microinstructions into single one (logic comparison, test + conditional branch)

Execution Engine (EE)



Execution Engine (EE) – Remarks

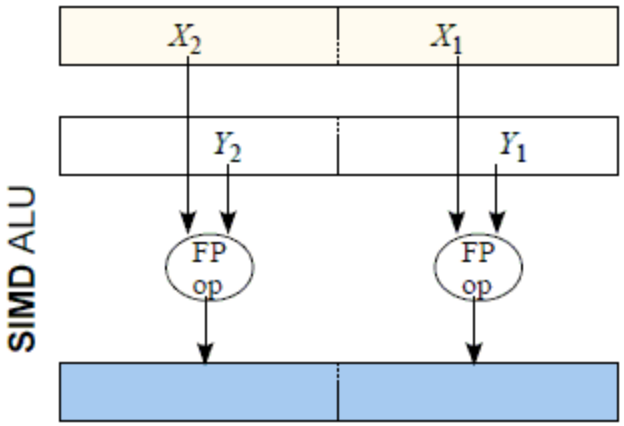
- MOB (Memory Order Buffer) supports speculative (out of order) reads and writes and ensures, that memory writes are realized in the original order with corresponding data
- Register renaming
- Bypass network – forwarding



- Execution Units – 6 operations (3 memory, 3 arithmetic/logic) in the single cycle:
- **Port 0** supports: Integer ALU and Shift Units, Integer SIMD ALU and SIMD shuffle, Single precision FP MUL, double precision FP MUL, FP MUL (x87), FP/SIMD/SSE2 Move and Logic and FP Shuffle, DIV/SQRT
- **Port 1** supports: Integer ALU, integer LEA and integer MUL, Integer SIMD MUL, integer SIMD shift, PSAD and string compare, and FP ADD
- **Port 2** Integer loads, **Port 3** Store address, **Port 4** Store data
- **Port 5** supports: Integer ALU and Shift Units, jump, Integer SIMD ALU and SIMD shuffle, FP/SIMD/SSE2 Move and Logic

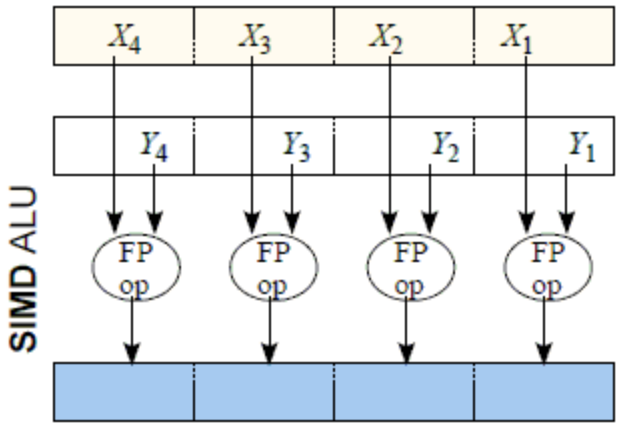
Execution Engine (EE) – Remarks

1 128 bit **XMM** data register ≡
2 packed DP FP quantities



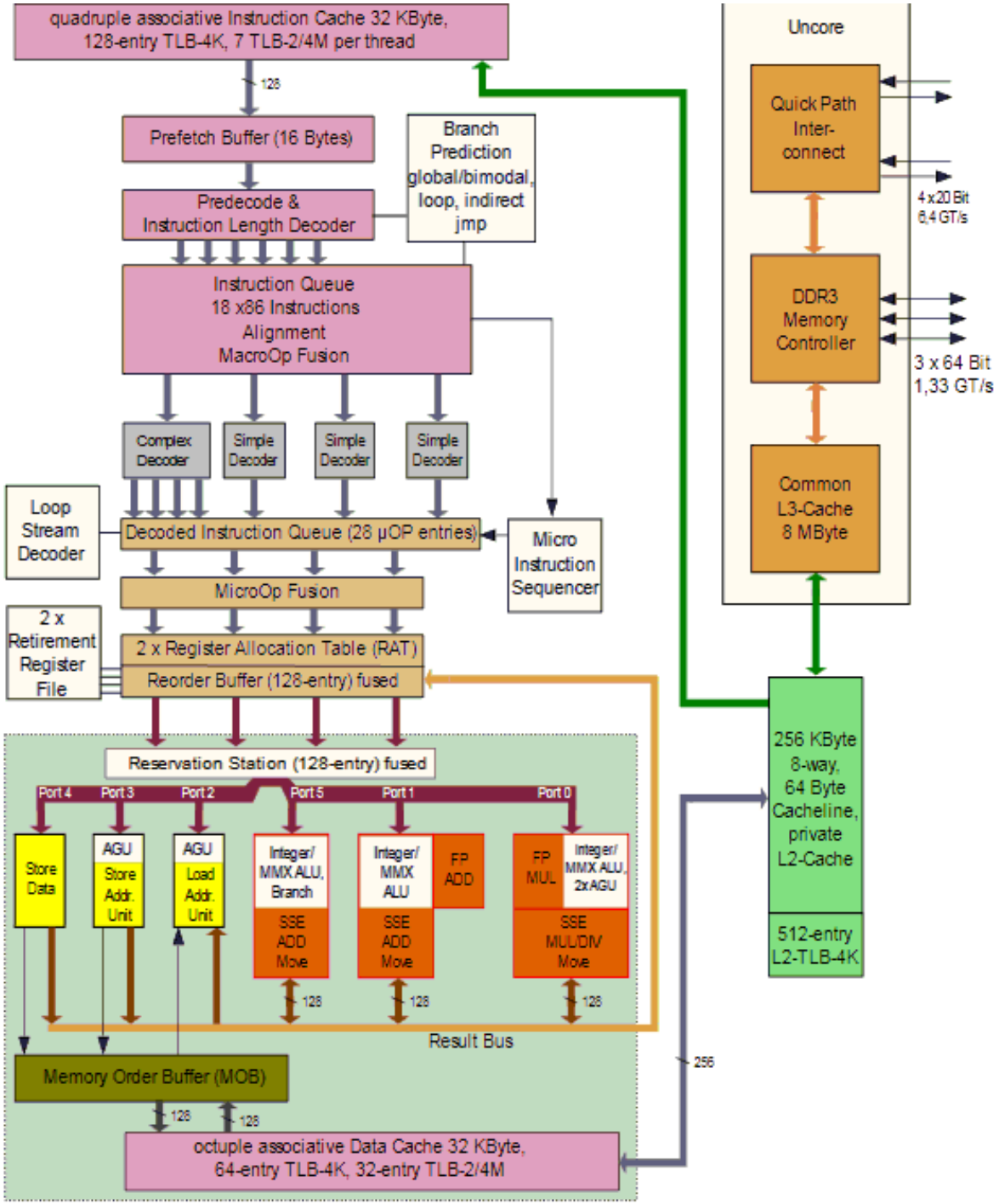
2 64-bit **Double**-Precision
SIMD FP operations
with **XMM** data registers
↔
2 DP FP ops / cycle / port

1 128 bit **XMM** data register ≡
4 packed SP FP quantities

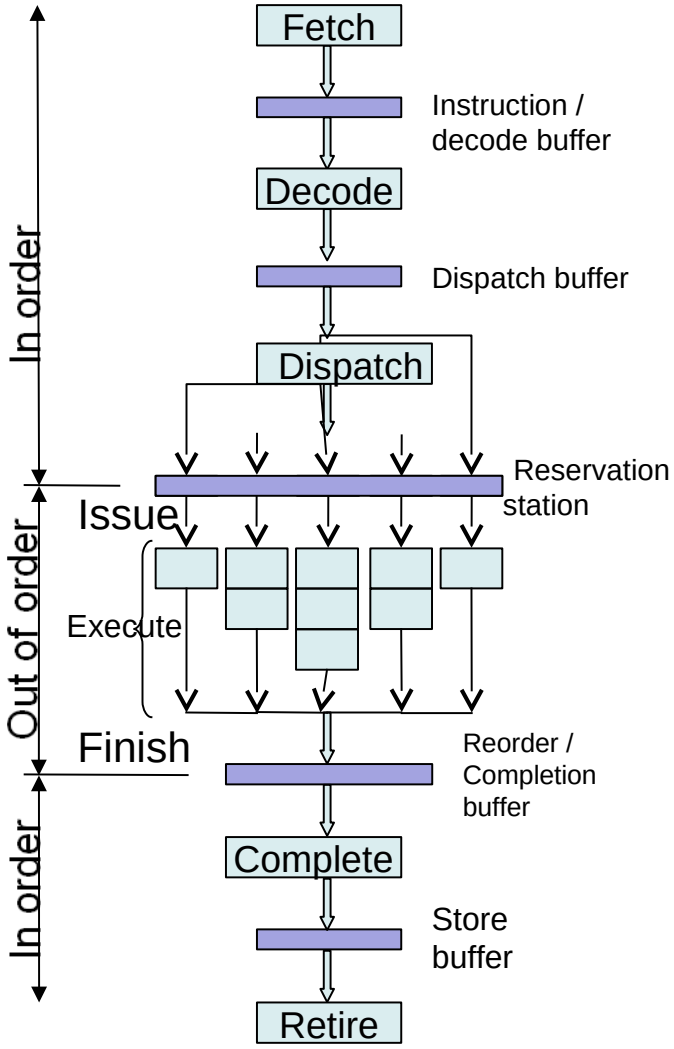


4 32-bit **Single**-Precision
SIMD FP operations
with **XMM** data registers
↔
4 SP FP ops / cycle / port

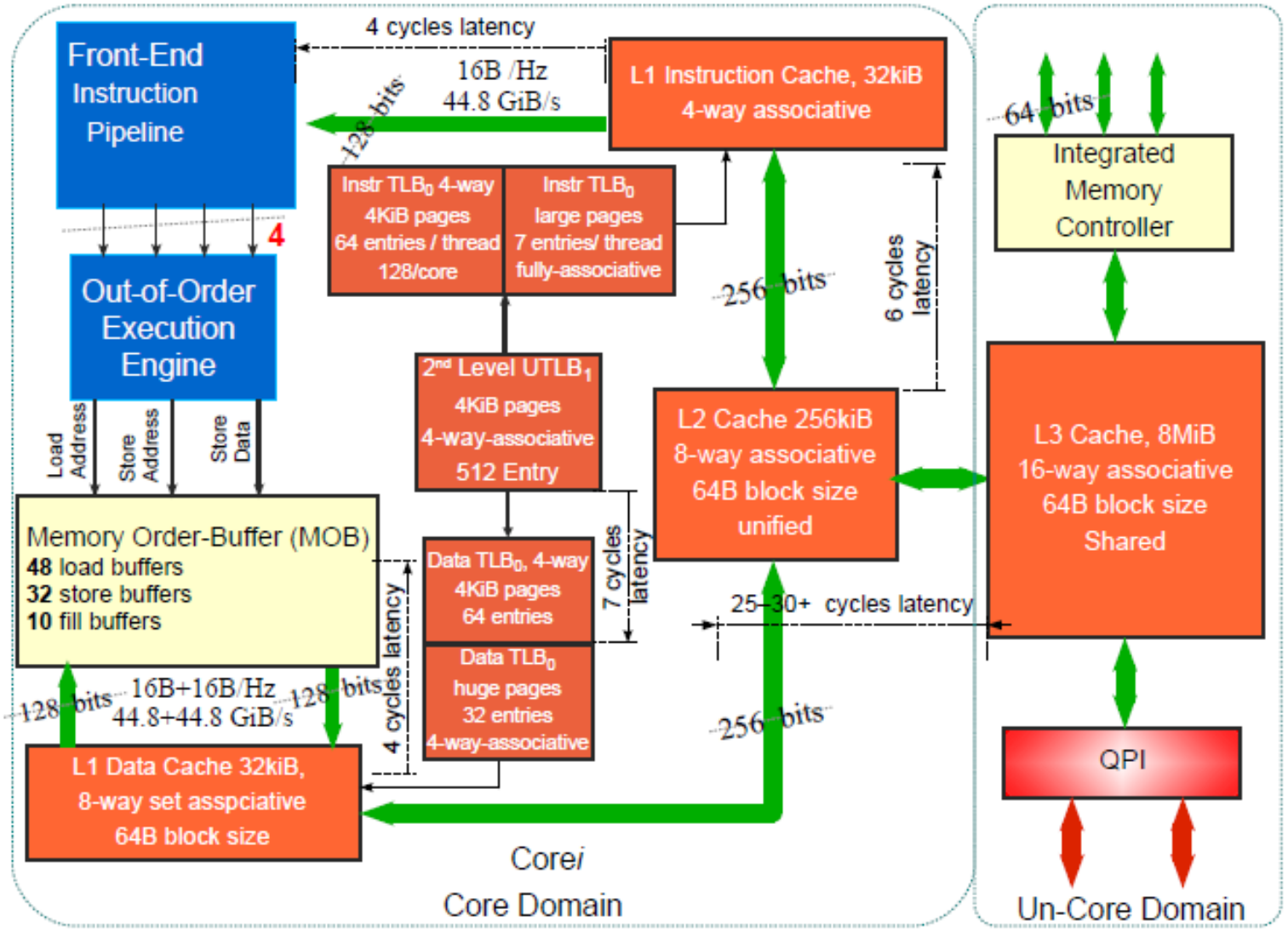
$$\begin{aligned}
\textbf{11.2}\ \text{Giga FLOPs/sec/core} &= 2.8\text{GHz} \times 4\text{FLOPs/Hz} \\
\textbf{44.8}\ \text{Giga FLOPs/sec/socket} &= 11.2\text{Giga FLOPs/sec/core} \times 4\text{cores} \\
\textbf{89.6}\ \text{Giga FLOPs/sec/node} &= 44.8\text{Giga FLOPs/sec/socket} \times 2\text{sockets},
\end{aligned}$$



Theory and reality

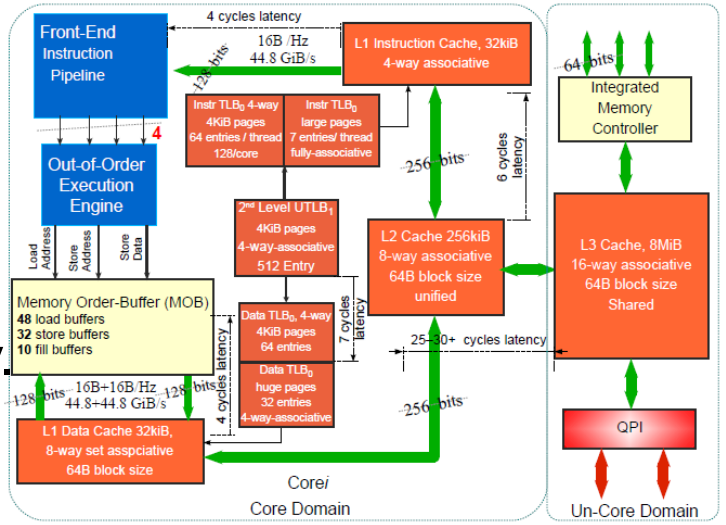


Memory Organization



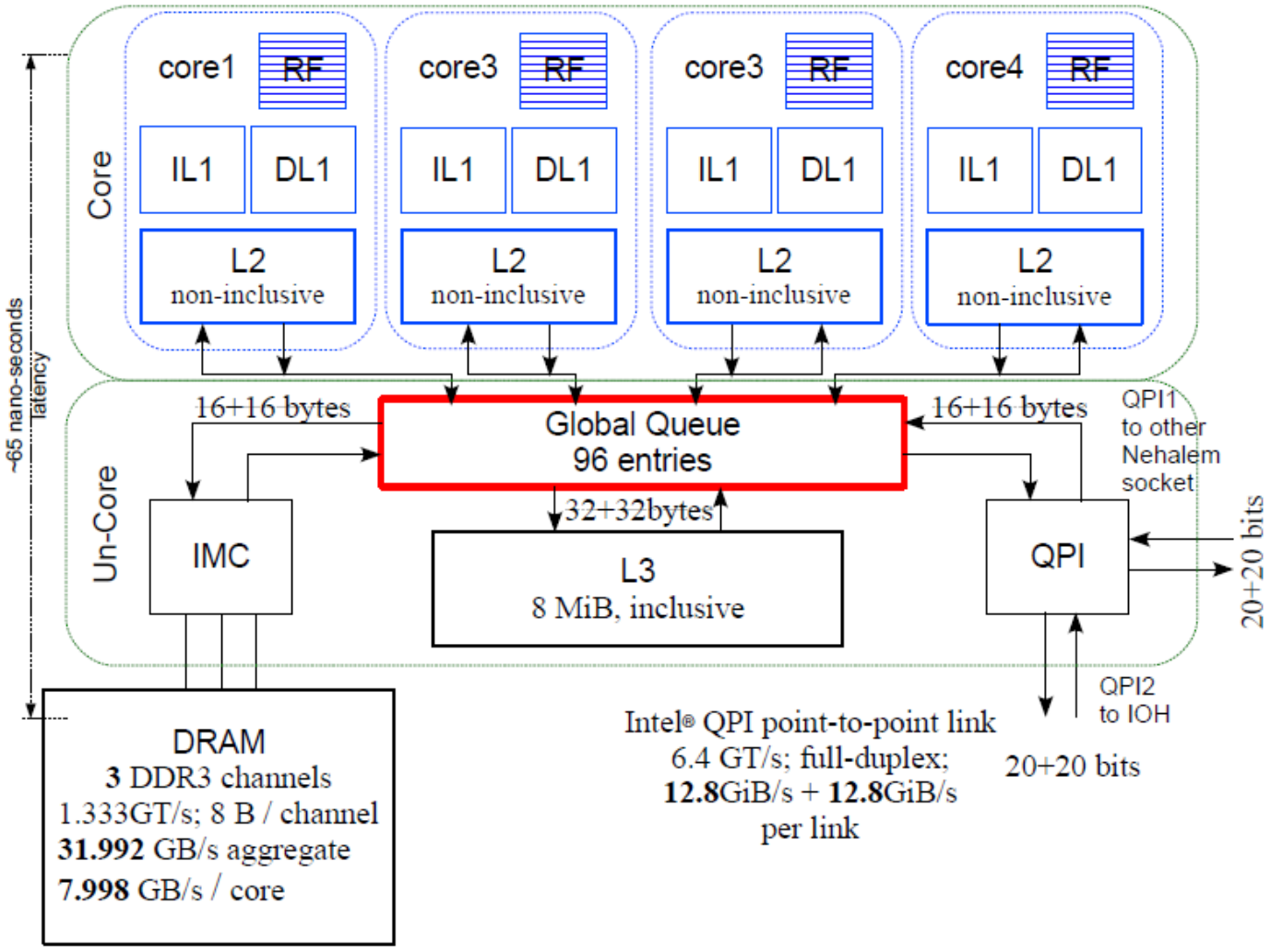
Memory Organization – Remarks

- Block size: 64B
- Processor reads reads whole cache line from main memory 64B, partial line fill or dirty are not supported
- L1 – Harvard. When SMT, shared by both h. threads, instruction – 4-way, data 8-way.
- L2 – unified, 8-way, non-inclusive, WB
- L3 – unified, 16-way, inclusive
  (cache line present in L1 or L2 is allocated/present in L3), WB



- **Store Buffers** – temporal store of the data for each memory write. No need to wait for write into cache or memory. Guarantees, that writes are in original program order and when required supports
  - discard in the case of exception, finish interrupt, serialization, lock,..
  - store forwarding
- Data prefetching to L1 caches (streaming prefetcher, strided prefetcher)
- Data prefetching to L2 – recognized access patters and fills L1 and L2

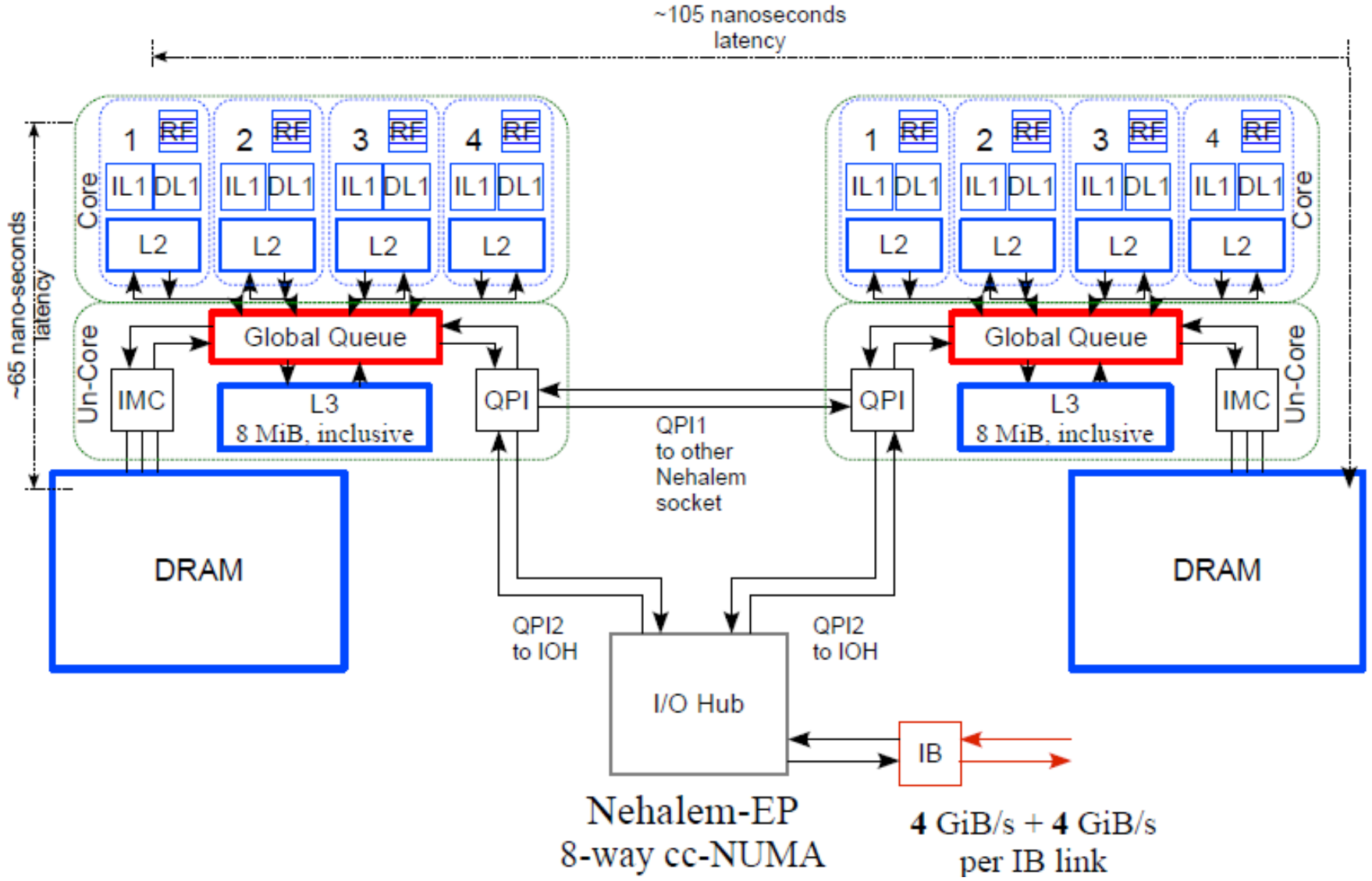
L3 – Shared Cache for All Cores



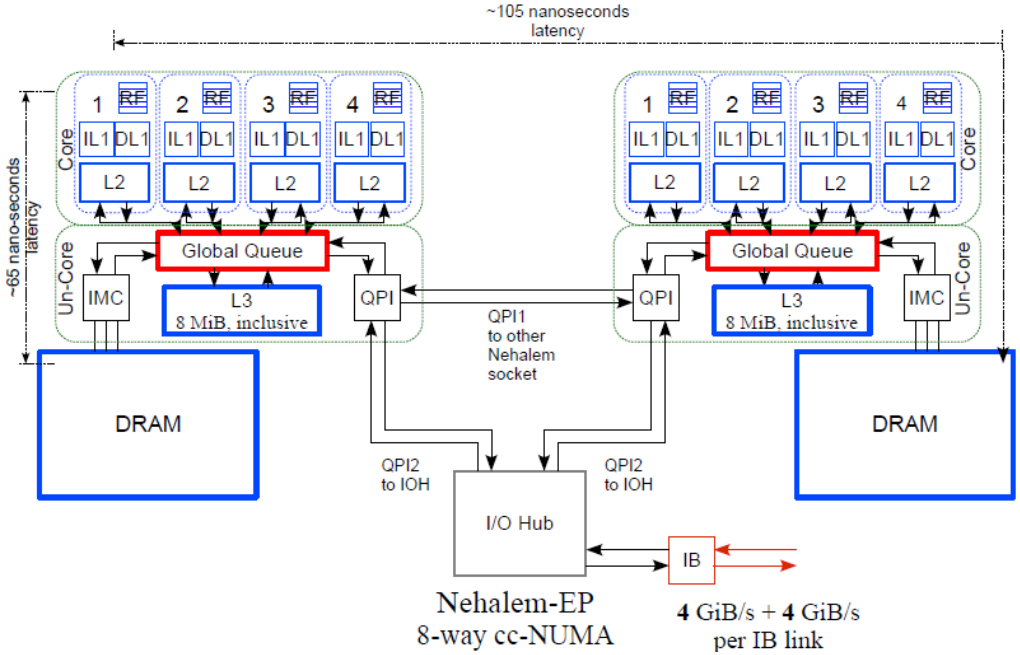
Global Queue

- GQ part of „uncore“ (L3 and memory controllers) and controls data flow
- Cache line requests on-chip cores, remote cores, or from I/O hub
- MESIF protocol – uses inclusive L3 cache
- Cache line requests from the on-chip four cores, from a remote chip or the I/O hub are handled by the Global Queue (GQ) which resides in the Uncore. The GQ buffers, schedules and manages the flow of data traffic through the uncore.
- Write Queue (WQ): 16-entry queue for memory write requests issued by local cores
- Load Queue (LQ): 32-entry queue for memory read requests from local cores
- QPI Queue (QQ): 12-entry queue for request received from QPI interfaces

Two Processors/Sockets



Two Processors/Sockets – Local RAM Access

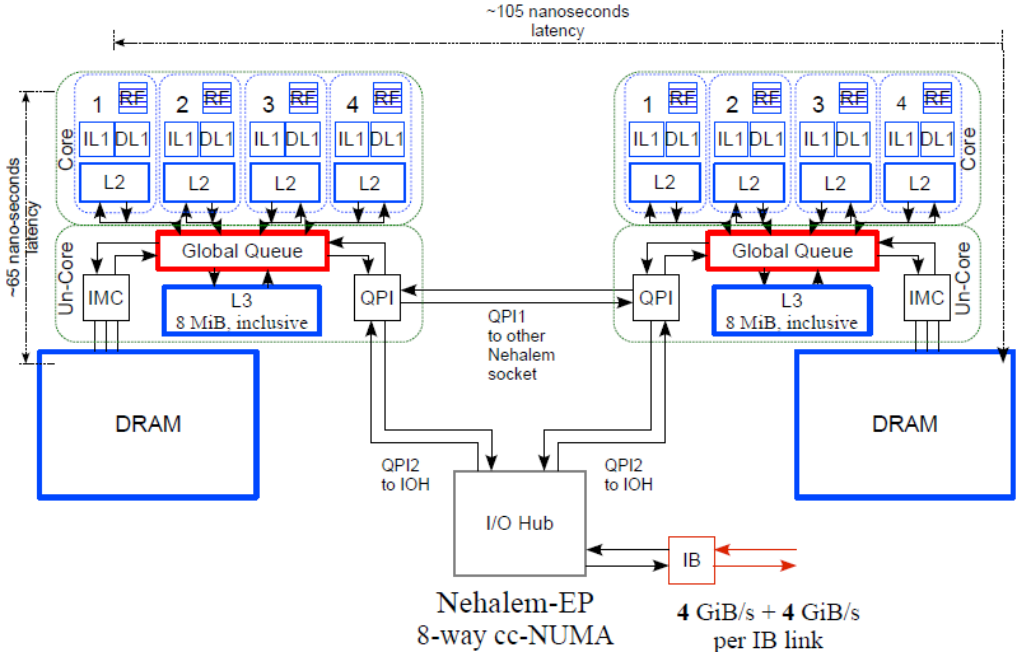


MESIF protocol

L3 cache – 4 bits

1. Processor 0 requests cache line, which is not present in any of L1, L2 or L3
   - Processor 0 requests data, which are located in its local DRAM,
   - Processor 0 snoops for data to be provided by Procesor 1
2. Response
   - local DRAM provides data,
   - Processor 1 provides response,
   - Processor 0 fills cache line

Two Processors/Sockets – Remote RAM Access



MESIF protokol

L3 cache – 4 bits

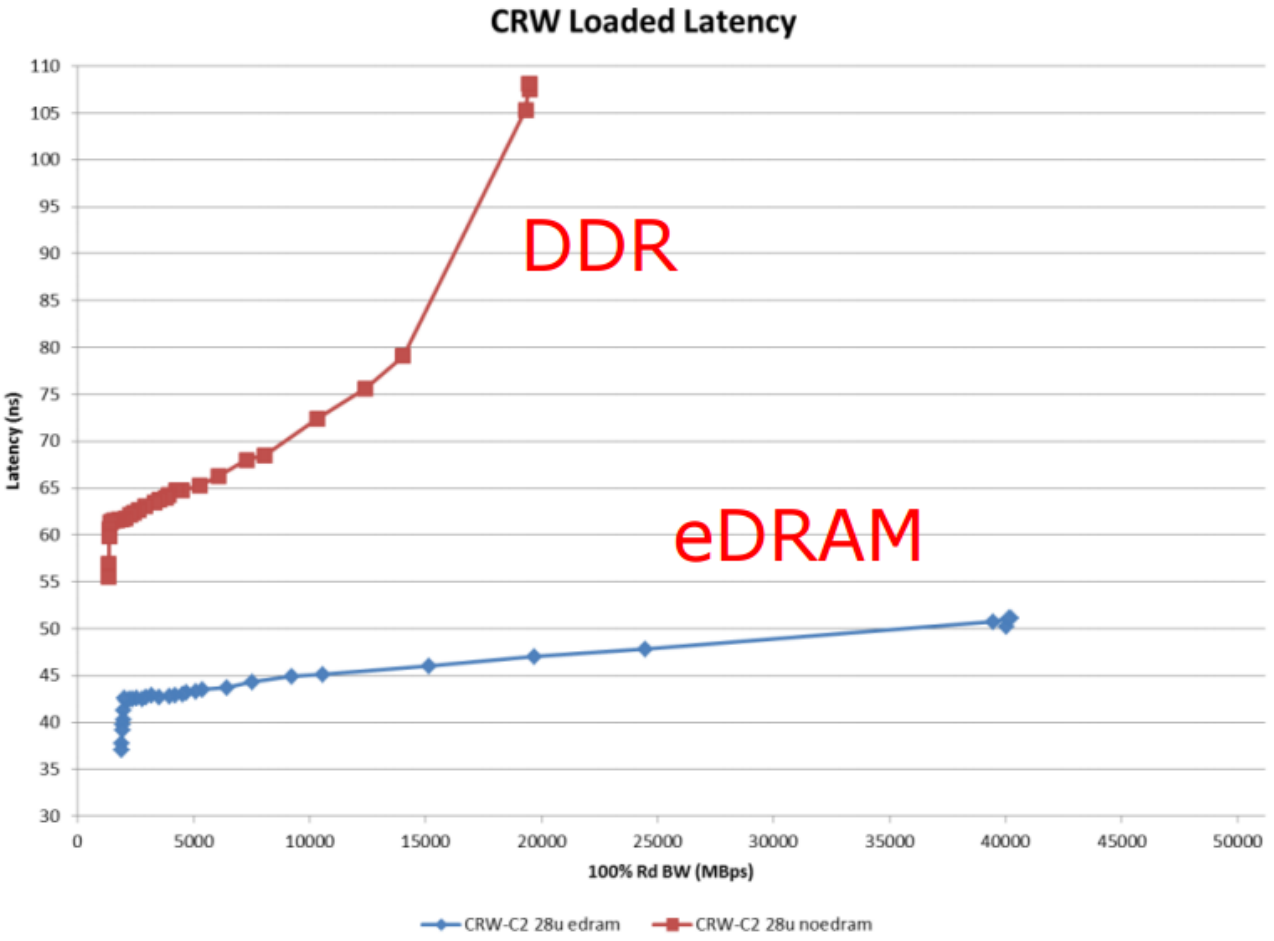
L3 Shared
L3 Modified

1. Processor 0 requests cache line, which is not presents neither L1, L2 or L3 cache
2. Request is sent by QPI to Processor 1
3. Processor 1 reacts (IMC generates request for DRAM, Processor 1 checks its its cache)
4. Response
   - Data sent to Processor 0 through QPI
   - Processor 0 fills cache and sets Shared or Exclusive state

Haswell (2013)

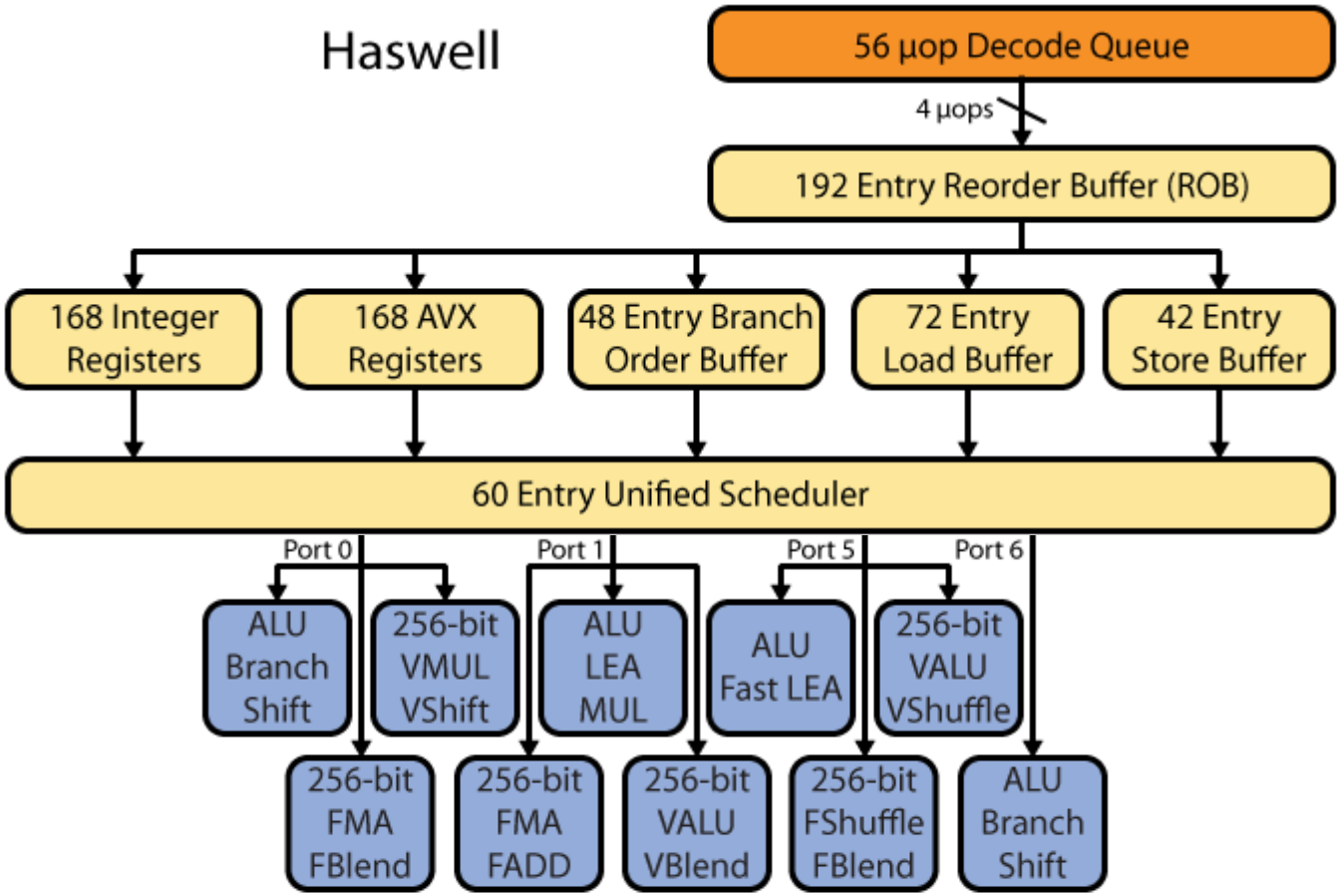
- Core i3, i5, and i7 fourth generation
- Introduces many additional instructions,
  example. Fused Multiply-Subtract:
  A = AxB – C
  C -= AxB
- Current trends: **System-on-a-Chip** (SoC) - CPU, GPU, Last Level Cache and system I/O on the single chip
- 128 MB L4 cache – eDRAM (Iris Pro model – integrated graphics accelerators)

eDRAM (embedded DRAM) and High Bandwidth Memory

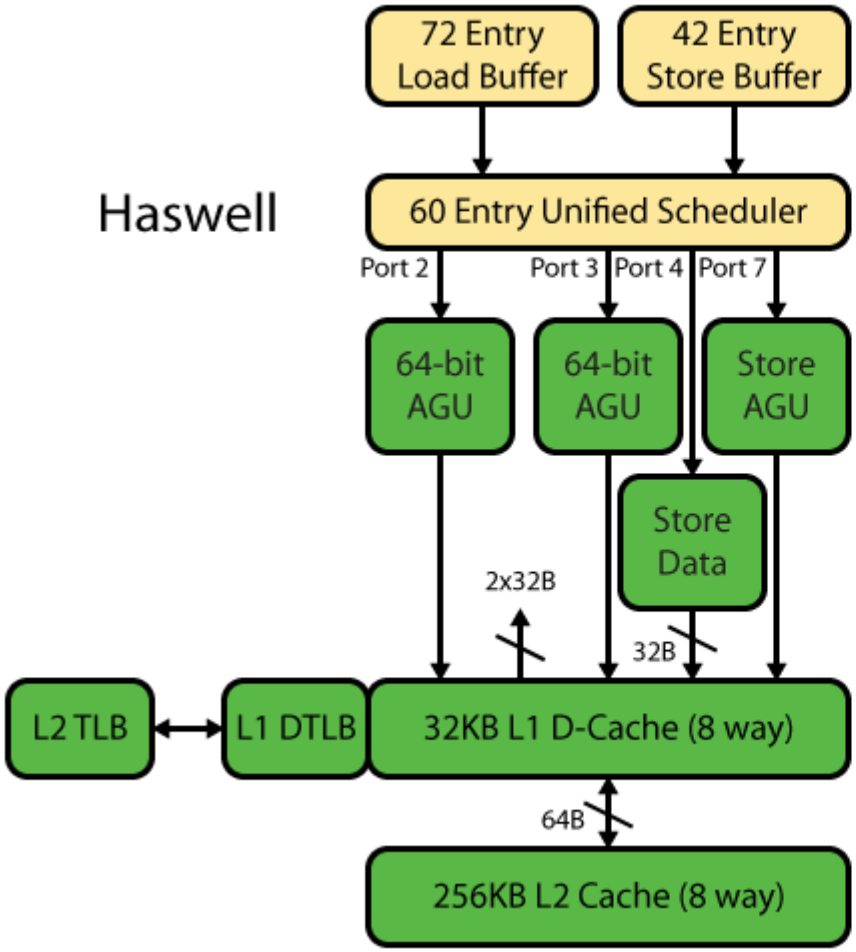


http://www.hotchips.org/wp-content/uploads/hc_archives/hc25/HC25.80-Processors2-epub/HC25.27.820-Haswell-Hammarlund-Intel.pdf

Haswell Out-of-Order Scheduling + Execution Units



Haswell – Memory Hierarchy



Intel Core Comparison

	Nehalem (2008)	Sandy Bridge (2011)	Haswell (2013)
L1 i-cache	32 KB, 4-way	32 KB, 8-way	32 KB, 8-way
Predecoded instructions …to Instruction Queue	6 instructions	6 instructions	6 instructions
Decoders number	3+1 complex	3+1 complex	3+1 complex
DecodeQ size, count of micro-op in	28	2x28	56
ROB entries (out-of-order window)	128	168	192
Unified reservation station entries / scheduler entries	36	54	60
In-flight loads	48	64	72
In-flight stores	32	36	42

Intel Core Microarchitecture Versions and Codenames

- Nehalem microarchitecture (1st generation)
  - Bloomfield (45 nm, 2008), Lynnfield" (45 nm, 2009)
- Westmere microarchitecture (1st generation)
  - Gulftown (32 nm, 2010)
- Sandy Bridge microarchitecture (2nd generation)
  - Sandy Bridge (32 nm, 2011), Sandy Bridge-E (32 nm, 2011)
- Ivy Bridge microarchitecture (3rd generation)
  - Ivy Bridge (22 nm, 2012), Ivy Bridge-E (22 nm, 2013)
- Haswell microarchitecture (4th generation)
  - Haswell-DT (quad-core, 22 nm, 2013), Haswell-H (MCP, quad-core, 22 nm, Haswell-E (22 nm 2014)
- Broadwell microarchitecture (5th generation)
  - Broadwell-H (quad-core, 14 nm, 2015), Broadwell-E (14 nm, 2016)
- Skylake microarchitecture (6th generation)
  - Skylake-S (quad-core, 14 nm, 2015), Skylake-H (quad-core, 14 nm, 2016), Skylake-X (14 nm, 2017)
- Kaby Lake microarchitecture (7th generation)
  - Kaby Lake-S (14 nm, 2017), Kaby Lake-X (14 nm, 2017)
- Coffee Lake microarchitecture (8th/9th generation)
  - Coffee Lake-" (14 nm, 2017)

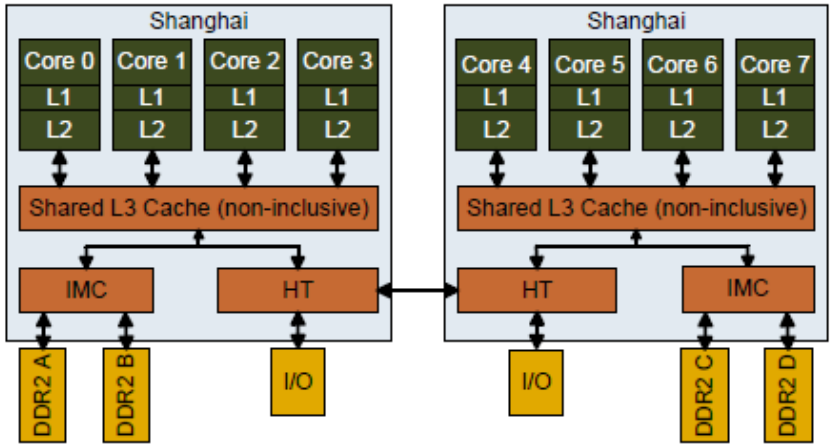
Zdroj: https://en.wikipedia.org/wiki/List_of_Intel_Core_i7_microprocessors

Intel Core Microarchitecture Versions and Codenames, cont.

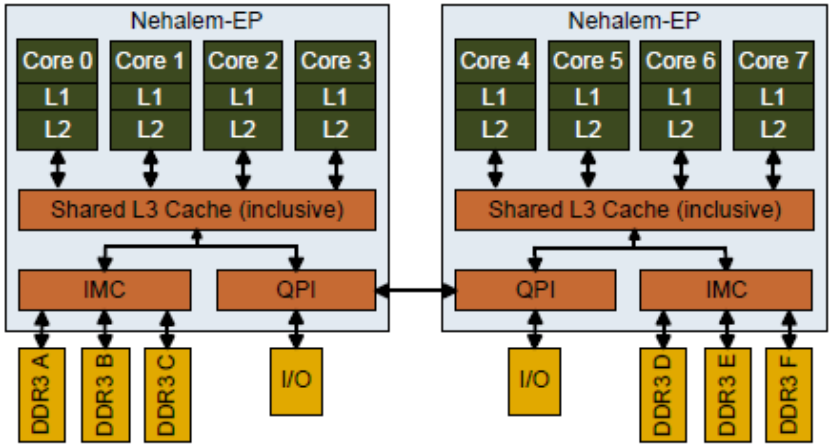
- Comet Lake microarchitecture (10th generation)
  - Comet Lake-S (14 nm, 2020)
- Cypress Cove microarchitecture (11th generation)
  - Rocket Lake-S (14 nm, 2021)
- Golden Cove + Gracemont microarchitecture (12th generation)
  - Alder Lake (Intel 7, 2021)

Zdroj: https://en.wikipedia.org/wiki/List_of_Intel_Core_i7_microprocessors

AMD (Shanghai) vs. Intel (Nehalem)

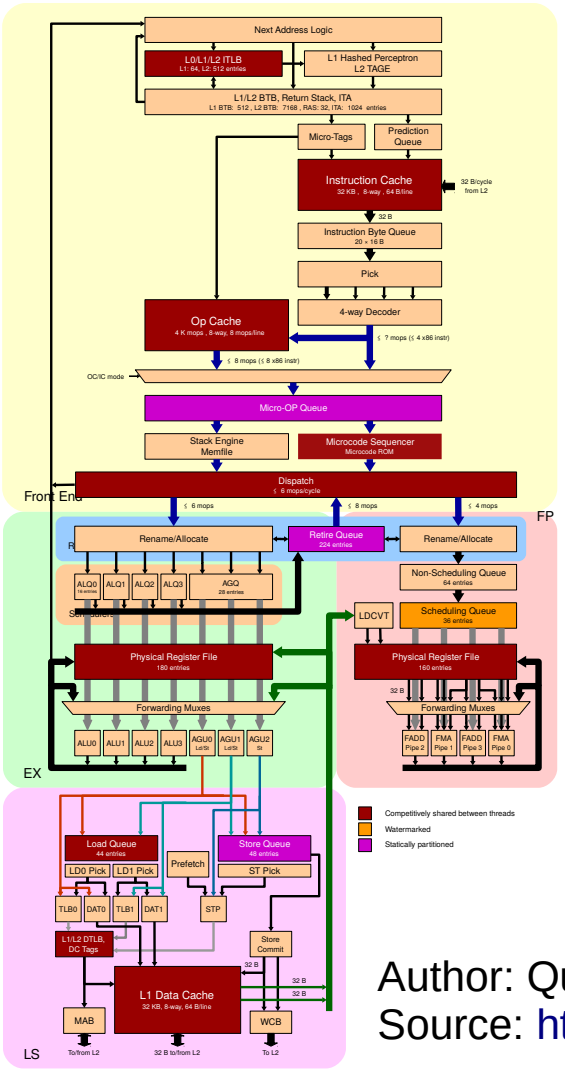


AMD



Intel

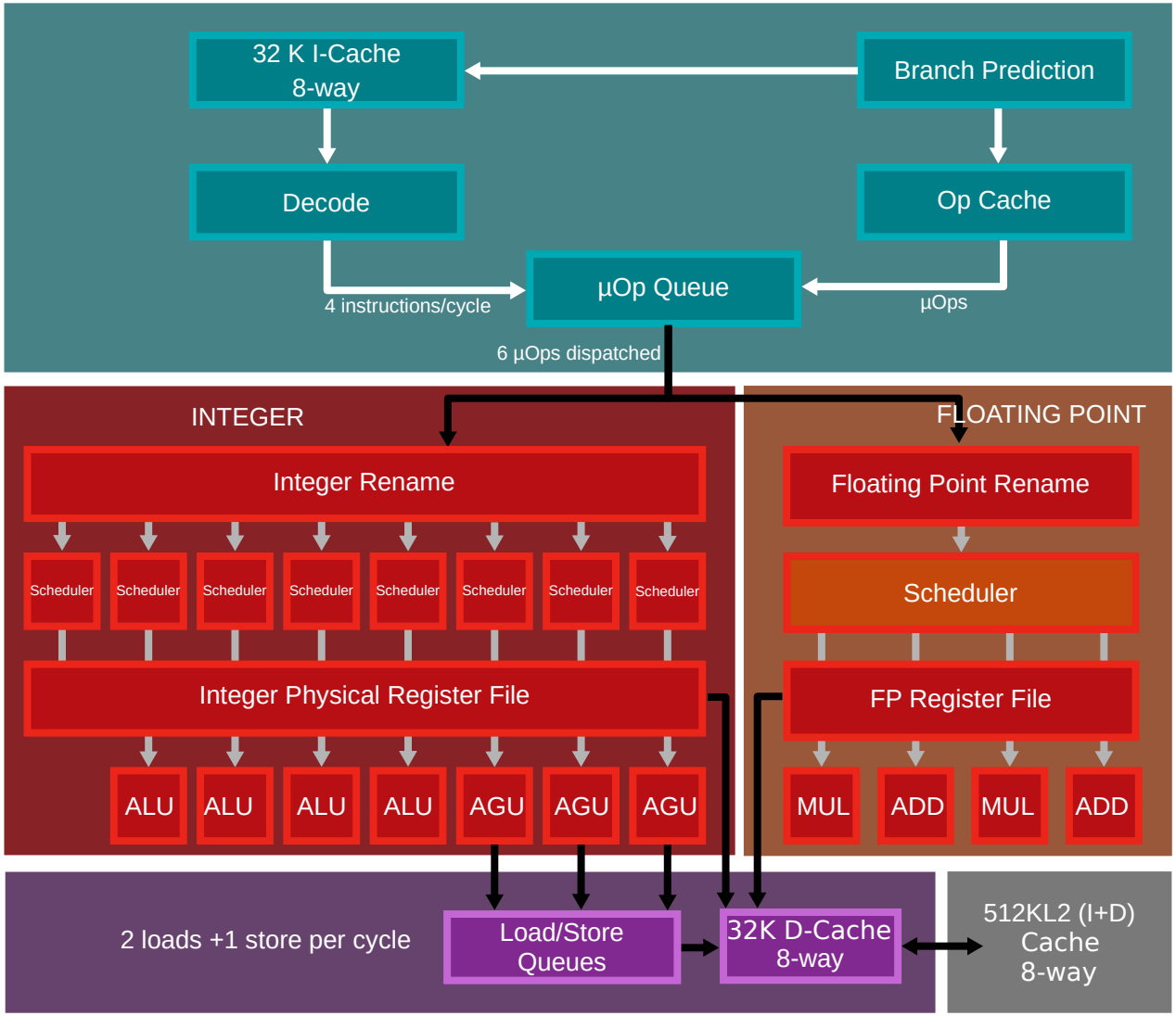
AMD Zen 2 - Microarchitecture



- 7 nm process (from 12 nm), I/O die utilizes 12 nm
- Core (8 cores on CPU chiplet), 6/8/4 µOPs in parallel
  - Frontend, µOP cache (4096 entries)
  - FPU, 256-bit Eus (256-bit FMAs) and LSU (2x256-bit L/S), 3 cycles DP vector mult latency
  - Integer, 180 registers, 3x AGU, scheduler (4x16 ALU + 1x28 AGU)
  - Reorder Buffer 224 entries
- Memory subsystem
  - L1 i-cache and d-cache, 32 KiB each, 8-way associative
  - L2 512 KiB per core, 8-way,
  - L2 DTLB 2048-entry
  - 48 entry store queue
- CCX
  - L3, slices, 2x 16 MiB
  - L3 latency (~40 cycles)
- In-silicon Spectre enhancements
- I/O, PCIe 4.0, Infinity Fabric 2, 25 GT/s

Author: QuietRub
Source: https://en.wikichip.org/wiki/amd/microarchitectures/zen_2

AMD Zen 2 - Microarchitecture



Alder Lake – Golden Cove

- High-performance "Performance-cores" (P-cores)
- Dedicated floating-point adders
- New 6-wide instruction decoder (older 4-wide) with the ability to fetch up to 32 bytes of instructions per cycle (older 16)
- 12 execution ports (up from 10)
- 512 reorder-buffer entries (up from 384)
- 6-wide µOP allocations (up from 5)
- µOP cache size increased to 4K entries (up from 2.25K)
- AVX-VNNI, a VEX-coded variant of AVX512-VNNI for 256-bit vectors
- AVX-512 (including FP16) is present but disabled by default to match E-cores
- ~18% IPC uplift

Source: Wikipedia.org

Alder Lake – Gracemont

- High-efficiency "Efficient-cores" (E-cores)
- E-cores are organized in 4-core modules; L2 cache is shared between E-cores within a module
- 256 reorder-buffer entries (up from 208 in Tremont)
- 17 execution ports (up from 12)
- AVX2, FMA and AVX-VNNI to catch up with P-cores
- Skylake-like IPC

Source: Wikipedia.org

Resources

Sources:

1. Thomadakis, M.E., The Architecture of the Nehalem Processor and Nehalem-EP SMP Platforms, Texas A&M University, 2011.
2. Nehalem (microarchitecture), http://en.wikipedia.org/wiki/Nehalem_%28microarchitecture%29
3. Trent Rolf, Cache Organization and Memory Management of the Intel Nehalem Computer Architecture, University of Utah Computer Engineering, Dec. 2009
4. Paul G. Howard, Next Generation Intel Microarchitecture Nehalem, Microway Inc., 2009
5. Intel QuickPath Interconnect, http://en.wikipedia.org/wiki/Intel_QuickPath_Interconnect
6. http://chip-architect.com/news/Nehalem_at_1st_glance_.jpg
7. http://www.realworldtech.com/haswell-cpu/
8. https://www.agner.org/optimize/