Look Inside™

Intel® Architecture for HPC Developers

Presenter: Georg Zitzlsberger

Date: 09-07-2015

Agenda

- **Introduction**
- Intel® Architecture
  - Desktop, Mobile & Server
  - Intel® Xeon Phi™ Coprocessor
- Summary

Moore's "Law"

"The number of transistors on a chip will **double** approximately **every two years.**"
[Gordon Moore]

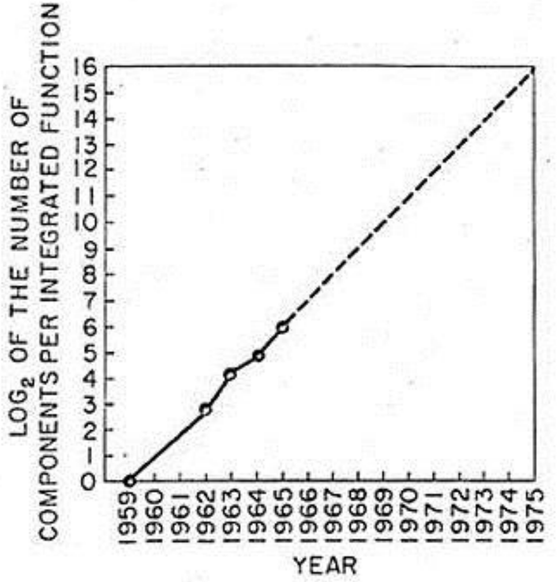
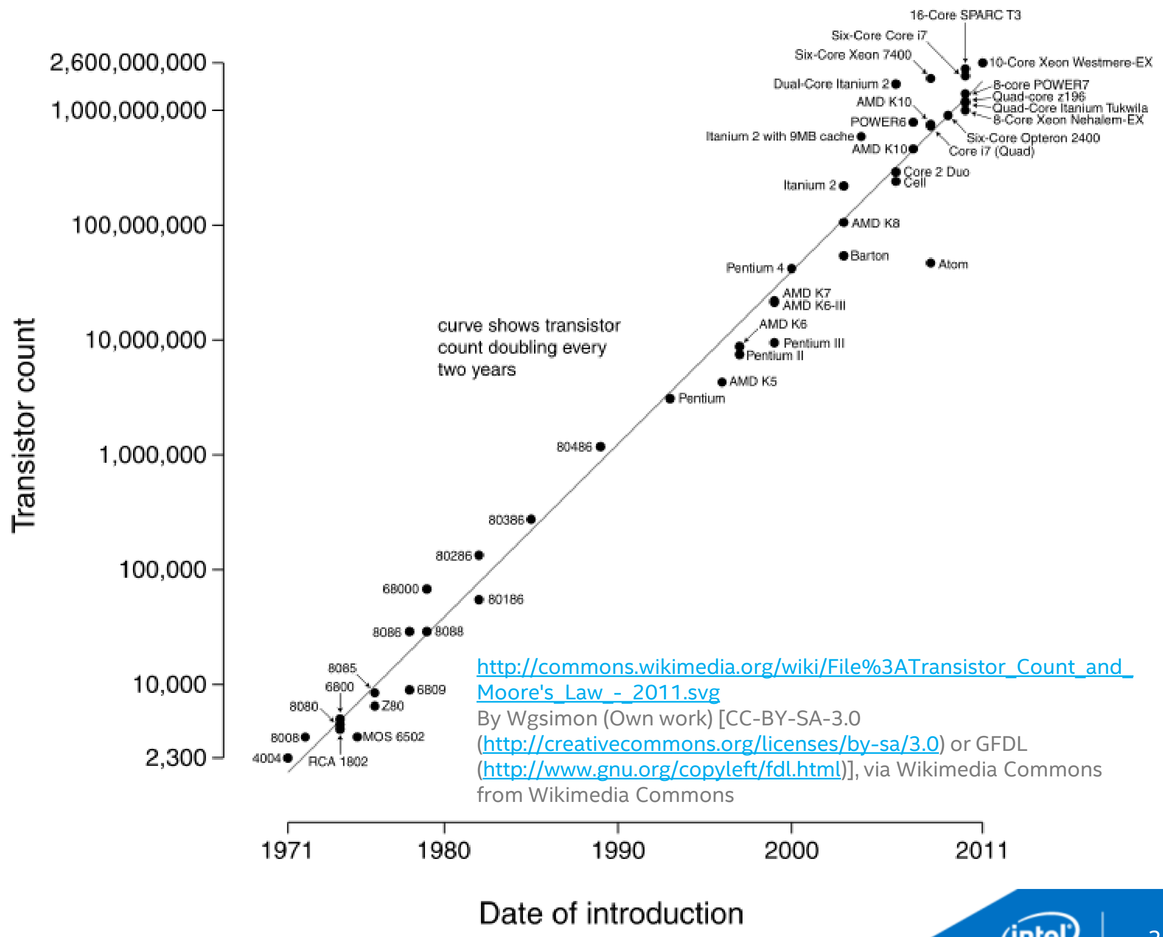


Fig. 2 Number of components per Integrated function for minimum cost per component extrapolated vs time.

Moore's Law graph, 1965

Microprocessor Transistor Counts 1971-2011 & Moore's Law



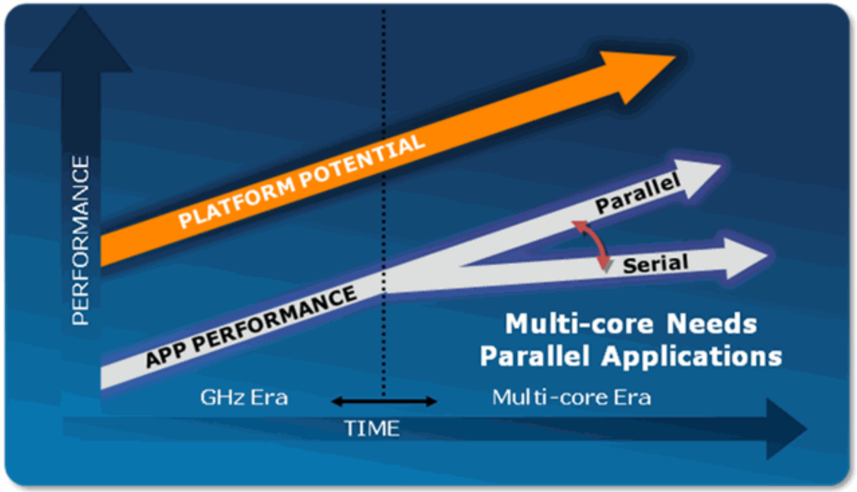
http://commons.wikimedia.org/wiki/File%3ATransistor_Count_and_Moore's_Law_-_2011.svg
By Wgsimon (Own work) [CC-BY-SA-3.0 (http://creativecommons.org/licenses/by-sa/3.0) or GFDL (http://www.gnu.org/copyleft/fdl.html)], via Wikimedia Commons from Wikimedia Commons

Parallelism

Problem:
Economical operation **frequency of (CMOS) transistors is limited**.
⇨ **No free lunch anymore!**

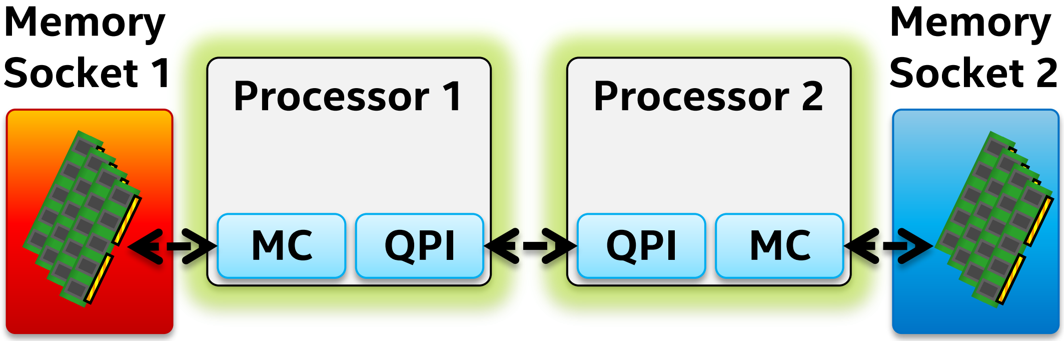
Solution:
More transistors allow more gates/logic on the same die space and power envelop, **improving parallelism**:

- **Thread level** parallelism (TLP):
  Multi- and many-core
- **Data level** parallelism (DLP):
  Wider vectors (SIMD)
- **Instruction level** parallelism (ILP):
  Microarchitecture improvements, e.g. threading, superscalarity, …



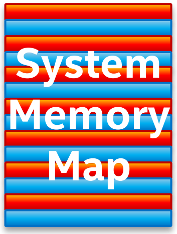
Processor Architecture Basics

UMA and NUMA



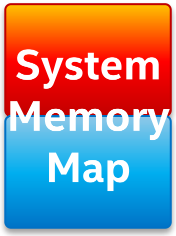
- **UMA (aka. non-NUMA):**
  - Uniform Memory Access (UMA)
  - Addresses interleaved across memory nodes **by cache line**
  - Accesses may or may not have to cross QPI link

  ⇨ **Provides good portable performance without tuning**



- **NUMA:**
  - Non-Uniform Memory Access (NUMA)
  - Addresses not interleaved across memory nodes by cache line
  - Each processor has direct access to contiguous block of memory

  ⇨ **Provides peek performance but requires special handling**



Processor Architecture Basics

UMA vs. NUMA

UMA (non-NUMA) is recommended:

- Avoid additional layer of complexity to tune for NUMA
- Portable NUMA-tuning is very difficult
- Future platforms based on Intel or non-Intel processors might require different NUMA tuning
- Own NUMA strategy might conflict with optimizations in 3rd party code or OS (scheduler)

Use NUMA if…

- Memory access latency and bandwidth is dominating bottleneck
- Developers are willing to deal with additional complexity
- System is dedicated to few applications worth the NUMA tuning
- Benchmark situation

Processor Architecture Basics

NUMA - Thread Affinity & Enumeration

Non-NUMA:
Thread affinity **might** be beneficial (e.g. cache locality) but not required

NUMA:
Thread affinity is **required**:

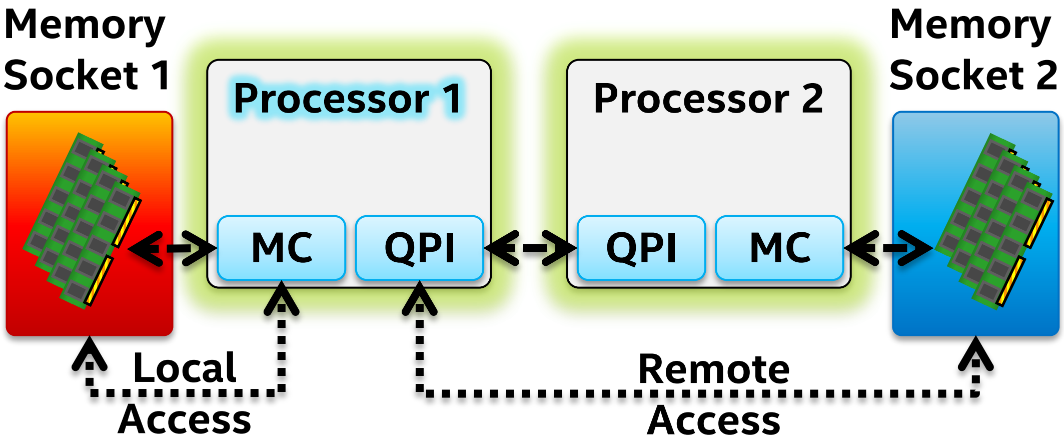
- Improve accesses to local memory vs. remote memory
- Ensure 3rd party components support affinity mapping, e.g.:
  - Intel® TBB via `set_affinity()`
  - Intel® OpenMP* via `$OMP_PLACES`
  - Intel® MPI via `$I_MPI_PIN_DOMAIN`
  - …
- Right way to get enumeration of cores:
  Intel® 64 Architecture Processor Topology Enumeration
  https://software.intel.com/en-us/articles/intel-64-architecture-processor-topology-enumeration

Processor Architecture Basics

NUMA - Memory, Bandwidth & Latency

Memory allocation:

- Differentiate: implicit vs. explicit memory allocation
- Explicit allocation with NUMA aware libraries, e.g. `libnuma` (Linux*)
- Bind **memory ⇔ (SW) thread**, and **(SW) thread ⇔ processor**
- More information on optimizing for performance: https://software.intel.com/de-de/articles/optimizing-applications-for-numa



Performance:

- Remote memory access **latency ~1.7x** greater than local memory
- Local memory **bandwidth** can be up to **~2x** greater than remote

NUMA Information

Get the NUMA configuration:
http://www.open-mpi.org/projects/hwloc/ or `numactl --hardware`

Documentation (`libnuma` & `numactl`):

http://halobates.de/numaapi3.pdf

`numactl`:

```
$ numactl --cpubind=0 --membind=0 <exe>
$ numactl --interleave=all <exe>
```

`libnuma`:

- Link with `–lnuma`
- Thread binding and preferred allocation:
  ```
  numa_run_on_node(node_id);
  numa_set_prefered(node_id);
  ```
- Allocation example:
  ```
  void *mem = numa_alloc_onnode(bytes, node_id);
  ```

Agenda

- Introduction
- **Intel® Architecture**
  - Desktop, Mobile & Server
  - Intel® Xeon Phi™ Coprocessor
- Summary

Desktop, Mobile & Server

“Big Core”

Desktop, Mobile & Server

Tick/Tock Model

Intel® Core™		2nd Generation Intel® Core™	3rd Generation Intel® Core™	4th Generation Intel® Core™
Nehalem (2008) New Microarchitecture **45nm**	**Westmere (2010)** New Process Technology **32nm**	**Sandy Bridge (2011)** New Microarchitecture **32nm**	**Ivy Bridge (2012)** New Process Technology **22nm**	**Haswell (2013)** New Microarchitecture **22nm**
Tock	**Tick**	**Tock**	**Tick**	**Tock**

5th Generation Intel® Core™	Future		
Broadwell (2014) New Process Technology **14nm**	**Skylake** New Microarchitecture **14nm**	New Process Technology **11nm**	New Microarchitecture **11nm**
Tick	**Tock**	**Tick**	**Tock**

Tock: Innovate

Tick: Shrink

Desktop, Mobile & Server

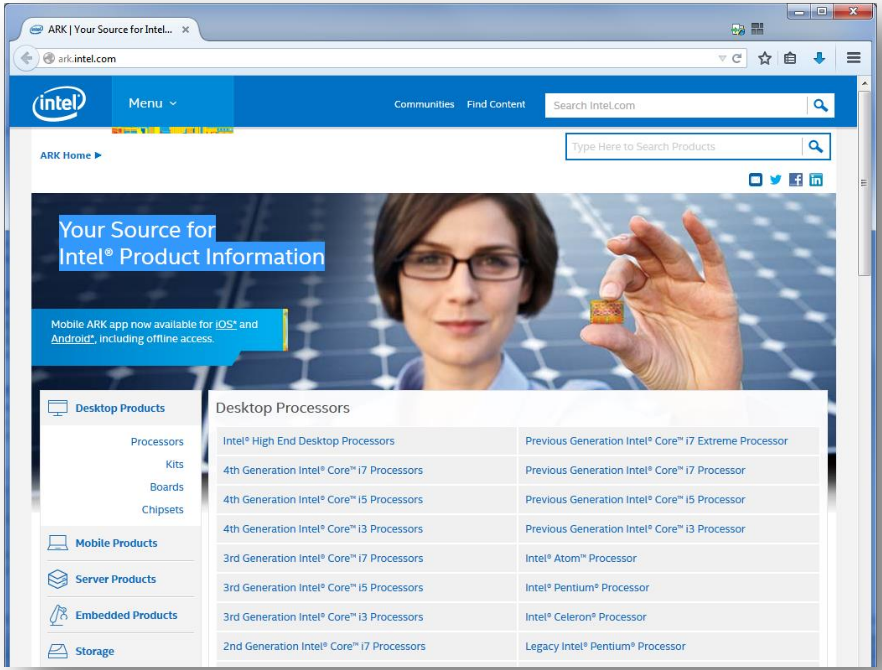
Your Source for Intel® Product Information

Naming schemes:

- Desktop & Mobile:
  - **Intel® Core™ i3/i5/i7** processor family
  - 5 generations, e.g.:
    4th Generation Intel® Core™ i5-4XXX
    5th Generation Intel® Core™ i5-5XXX
  - 1st generation starts with **“Nehalem”**

- Server:
  - **Intel® Xeon® E3/E5/E7** processor family
  - 3 generations, e.g.:
    Intel® Xeon® Processor E3-XXXX v3
  - 1st generation starts with **“Sandy Bridge”**

Information about available Intel products can be found here: http://ark.intel.com/



Your Source for Intel® Product Information

Mobile ARK app now available for iOS* and Android*, including offline access.

Desktop Processors

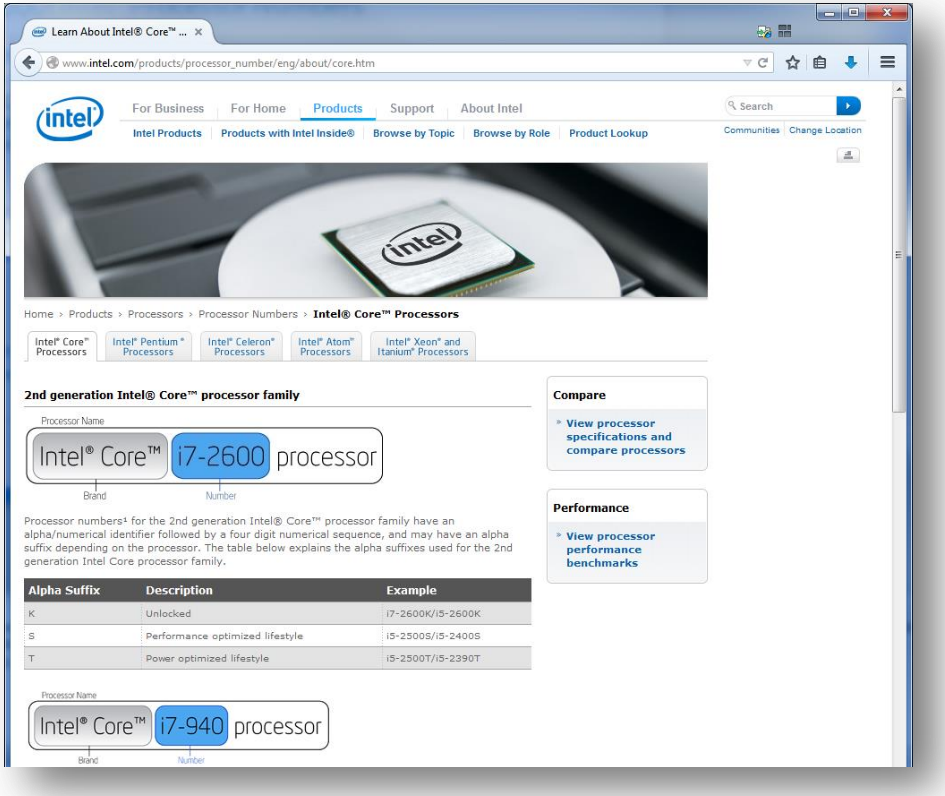
Intel® High End Desktop Processors	Previous Generation Intel® Core™ i7 Extreme Processor
4th Generation Intel® Core™ i7 Processors	Previous Generation Intel® Core™ i7 Processor
4th Generation Intel® Core™ i5 Processors	Previous Generation Intel® Core™ i5 Processor
4th Generation Intel® Core™ i3 Processors	Previous Generation Intel® Core™ i3 Processor
3rd Generation Intel® Core™ i7 Processors	Intel® Atom™ Processor
3rd Generation Intel® Core™ i5 Processors	Intel® Pentium® Processor
3rd Generation Intel® Core™ i3 Processors	Intel® Celeron® Processor
2nd Generation Intel® Core™ i7 Processors	Legacy Intel® Pentium® Processor

Desktop, Mobile & Server

Learn About Intel® Processor Numbers

Processor numbers follow specific scheme:

- Processor type
- Processor family/generation
- Product line/brand
- Power level
- Core count
- Multi-processor count
- Socket type
- …


Learn About Intel® Core™ ...
www.intel.com/products/processor_number/eng/about/core.htm
intel
For Business
For Home
Products
Support
About Intel
Search
Intel Products
Products with Intel Inside®
Browse by Topic
Browse by Role
Product Lookup
Communities
Change Location
Home › Products › Processors › Processor Numbers › Intel® Core™ Processors
Intel® Core™ Processors
Intel® Pentium® Processors
Intel® Celeron® Processors
Intel® Atom™ Processors
Intel® Xeon® and Itanium® Processors
Compare
View processor specifications and compare processors
Performance
View processor performance benchmarks
Processor Name
Intel® Core™ i7-2600 processor
Brand
Number
Processor Name
Intel® Core™ i7-940 processor
Brand
Number

2nd generation Intel® Core™ processor family

Processor numbers¹ for the 2nd generation Intel® Core™ processor family have an alpha/numerical identifier followed by a four digit numerical sequence, and may have an alpha suffix depending on the processor. The table below explains the alpha suffixes used for the 2nd generation Intel Core processor family.

Alpha Suffix	Description	Example
K	Unlocked	i7-2600K/i5-2600K
S	Performance optimized lifestyle	i5-2500S/i5-2400S
T	Power optimized lifestyle	i5-2500T/i5-2390T

Encoding of the processor numbers can be found here:
http://www.intel.com/products/processor_number/eng/

Desktop, Mobile & Server

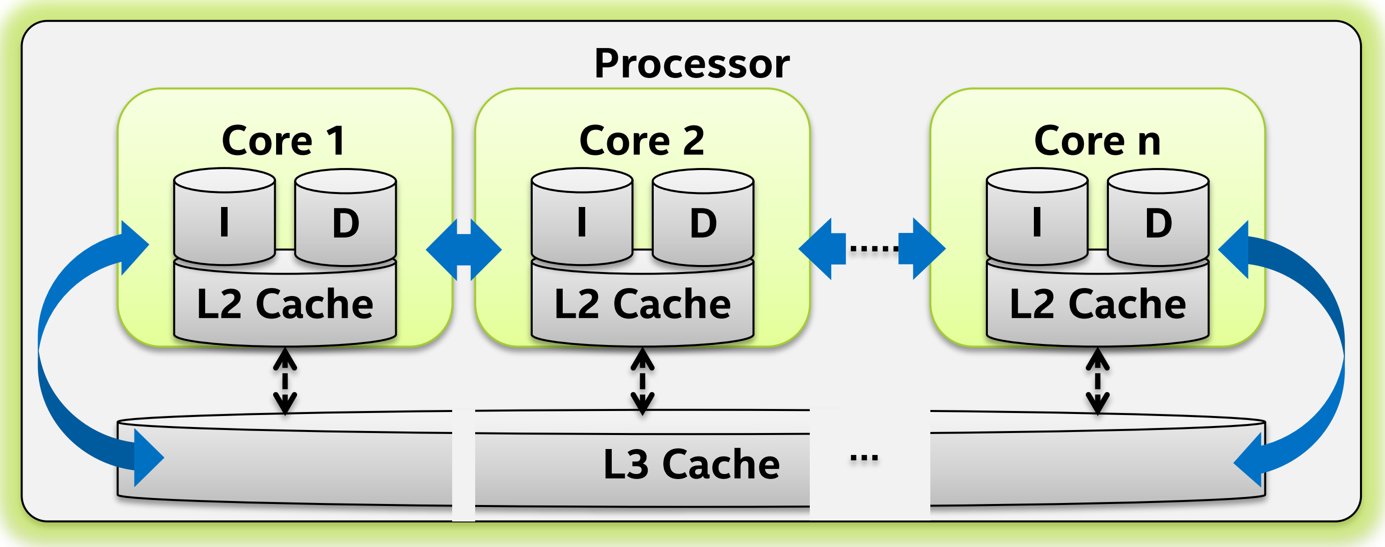
Characteristics

- Processor core:
  - **4 issue**
  - Superscalar **out-of-order** execution
  - Simultaneous multithreading:
    Intel® Hyper-Threading Technology with **2 HW threads per core**
- Multi-core:
  - Intel® Core™ processor family: up to 8 cores (desktop & mobile)
  - Intel® Xeon® processor family: up to 18 cores (server)
- Caches:
  - **Three level** cache hierarchy – L1/L2/L3 (Nehalem and later)
  - 64 byte cache line

Desktop, Mobile & Server

Caches

Cache hierarchy:



Level	Latency (cycles)	Bandwidth (per core per cycle)	Size
L1-D	4	2x 16 bytes	32KiB
L2 (unified)	12	1x 32 bytes	256KiB
L3 (LLC)	26-31	1x 32 bytes	varies (≥ 2MiB per core)
L2 and L1 D-Cache in other cores	43 (clean hit), 60 (dirty hit)		

Example for 4th Generation Intel® Core™

Optimization Notice

Desktop, Mobile & Server

Intel® Hyper-Threading Technology I

- 4 issue, superscalar, out-of-order processor:
  - 4 instructions are decoded to uops per cycle
  - Multiple uops are scheduled and executed in the backend
  - Backend-bound pipeline stalls likely (long 14 stage pipeline)

- Problem: How to increase instruction throughput?
  Solution: **Intel® Hyper-Threading Technology** (HT)
  - Simultaneous multi-threading (SMT)
  - **Two threads** per core

- Threads per core share the same resources - partitioned or duplicated:
  - L/S buffer and ROB
  - Scheduler (reservation station)
  - Execution Units
  - Caches
  - TLB

Desktop, Mobile & Server

Intel® Hyper-Threading Technology II

- Not shared are:
  - Registers
  - Architectural state
- Smaller extensions needed:
  - More uops in backend need to be handled
  - ROB needs to be increased
- Easy and efficient to implement:
  - Low die cost: Logic duplication is minimal
  - Easy to handle for a programmer (multiple SW threads)
  - Can be selectively used, depending on workload
  - Some workloads can benefit from SMT – just enable it
- More insights to Intel® Hyper-Threading Technology: https://software.intel.com/en-us/articles/performance-insights-to-intel-hyper-threading-technology

Desktop, Mobile & Server

New Instructions: 4th Generation Intel® Core™ (Haswell)

Group		Description	Count*
Intel® AVX2	SIMD Integer Instructions promoted to 256 bits	Adding vector integer operations to 256-bit	170 / 124
	Gather	Load elements using a vector of indices, vectorization enabler	
	Shuffling / Data Rearrangement	Blend, element shift and permute instructions	
FMA		Fused Multiply-Add operation forms (FMA-3)	96 / 60
Bit Manipulation and Cryptography		Improving performance of bit stream manipulation and decode, large integer arithmetic and hashes	15 / 15
~~TSX=RTM+HLE~~		~~Transactional Memory~~	~~4/4~~
Others		MOVBE: Load and Store of Big Endian forms INVPCID: Invalidate processor context ID	2 / 2

* Total instructions / different mnemonics

Desktop, Mobile & Server

Performance

- Following Moore's Law:

Microarchitecture	Instruction Set	SP FLOPs per Cycle per Core	DP FLOPs per Cycle per Core	L1 Cache Bandwidth (bytes/cycle)	L2 Cache Bandwidth (bytes/cycle)
Nehalem	SSE (128-bits)	8	4	32 (16B read + 16B write)	32
Sandy Bridge	Intel® AVX (256-bits)	16	8	48 (32B read + 16B write)	32
Haswell	Intel® AVX2 (256-bits)	32	16	96 (64B read + 32B write)	64

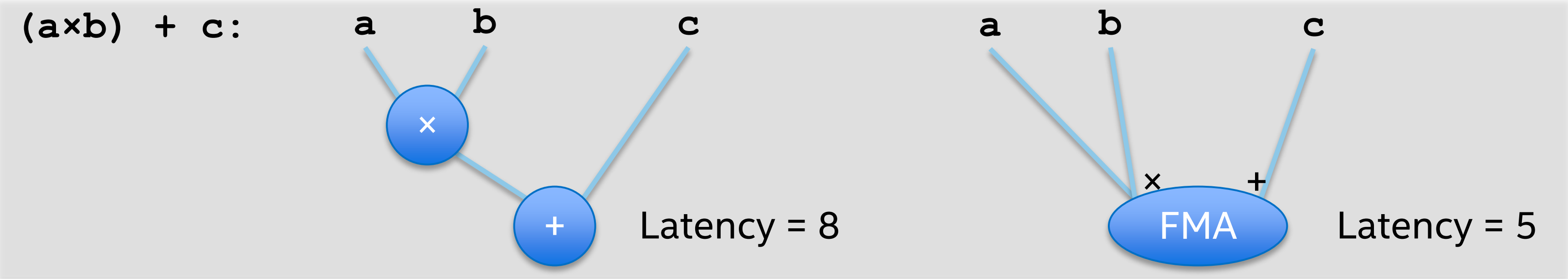
- Example of **theoretic peak** FLOP rates:
  - Intel® Core™ i7-2710QE (Sandy Bridge):
2.1 GHz * 16 SP FLOPs * 4 cores = **134.4 SP GFLOPs**

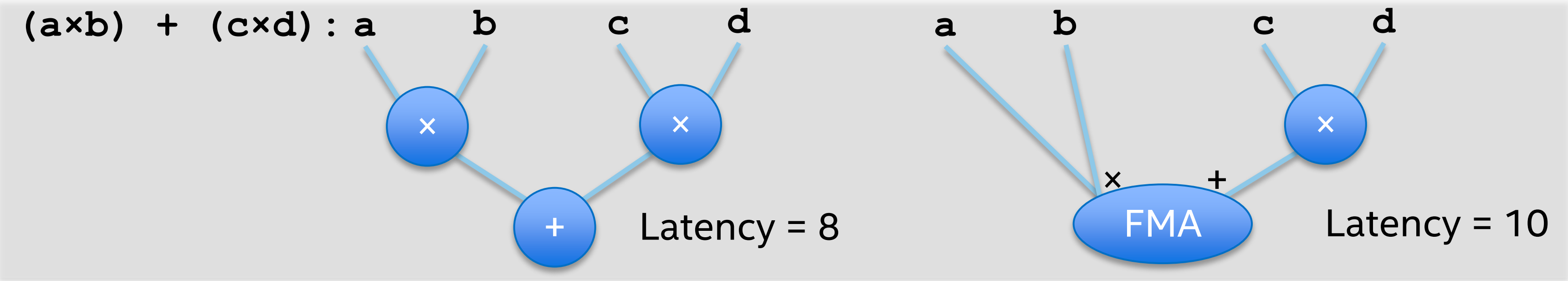
  - Intel® Core™ i7-4765T (Haswell):
2.0 GHz * 32 SP FLOPs * 4 cores = **256 SP GFLOPs**

Desktop, Mobile & Server

FMA Latency

- FMA latency better than combined multiply and add instruction:
  - Add latency: 3 cycles
  - Multiply and FMA latencies: 5 cycles
- But **not always optimal latency** with different combinations, e.g.:





⇨ **FMA can improve or reduce performance due to context!**

Desktop, Mobile & Server

Intel® Xeon® Processor: Ivy Bridge vs. Haswell

Feature	Ivy Bridge	Haswell
QPI Speed (GT/s)	6.4, 7.2 and 8.0	6.4, 8.0, 9.6
Cores	Up to 12	Up to 18
Last Level Cache (LLC)	Up to 30 MB	Up to 45 MB
Memory	DDR3-800/1066/1333/ 1600/1866	DDR4-1600/1866/2133
Max. Memory Bandwidth	59.7 GB/s	68 GB/s
Instruction Set Extension	Intel® AVX	Intel® AVX2

Desktop, Mobile & Server

New Instructions: 5th Generation Intel® Core™

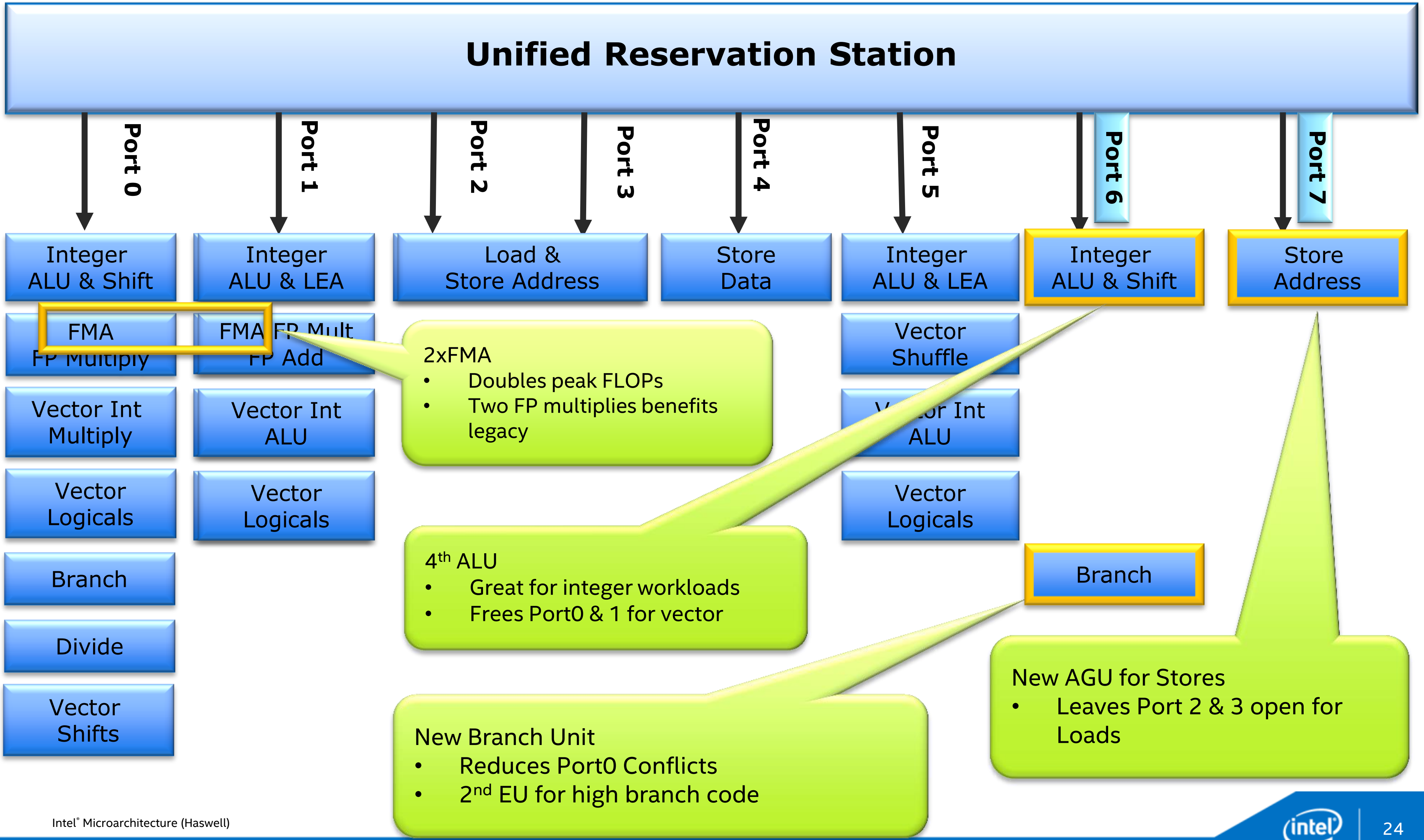
Instruction	Description
`RDSEED`	provide reliable seeds for pseudo-random number generator (PRNG)
`ADCX, ADOX`	large integer arithmetic addition
`PREFETCHW`	extending SW prefetch

Supported via intrinsics :

- Intel Compilers 13.0 Update 2 (and later)
- GNU GCC 4.8

4th Generation Intel® Core™ (Haswell)

Execution Unit Overview



Unified Reservation Station

Port 0
- Integer ALU & Shift
- FMA FP Multiply
- Vector Int Multiply
- Vector Logicals
- Branch
- Divide
- Vector Shifts

Port 1
- Integer ALU & LEA
- FMA FP Mult FP Add
- Vector Int ALU
- Vector Logicals

Port 2, Port 3
- Load & Store Address

Port 4
- Store Data

Port 5
- Integer ALU & LEA
- Vector Shuffle
- Vector Int ALU
- Vector Logicals

Port 6
- Integer ALU & Shift
- Branch

Port 7
- Store Address

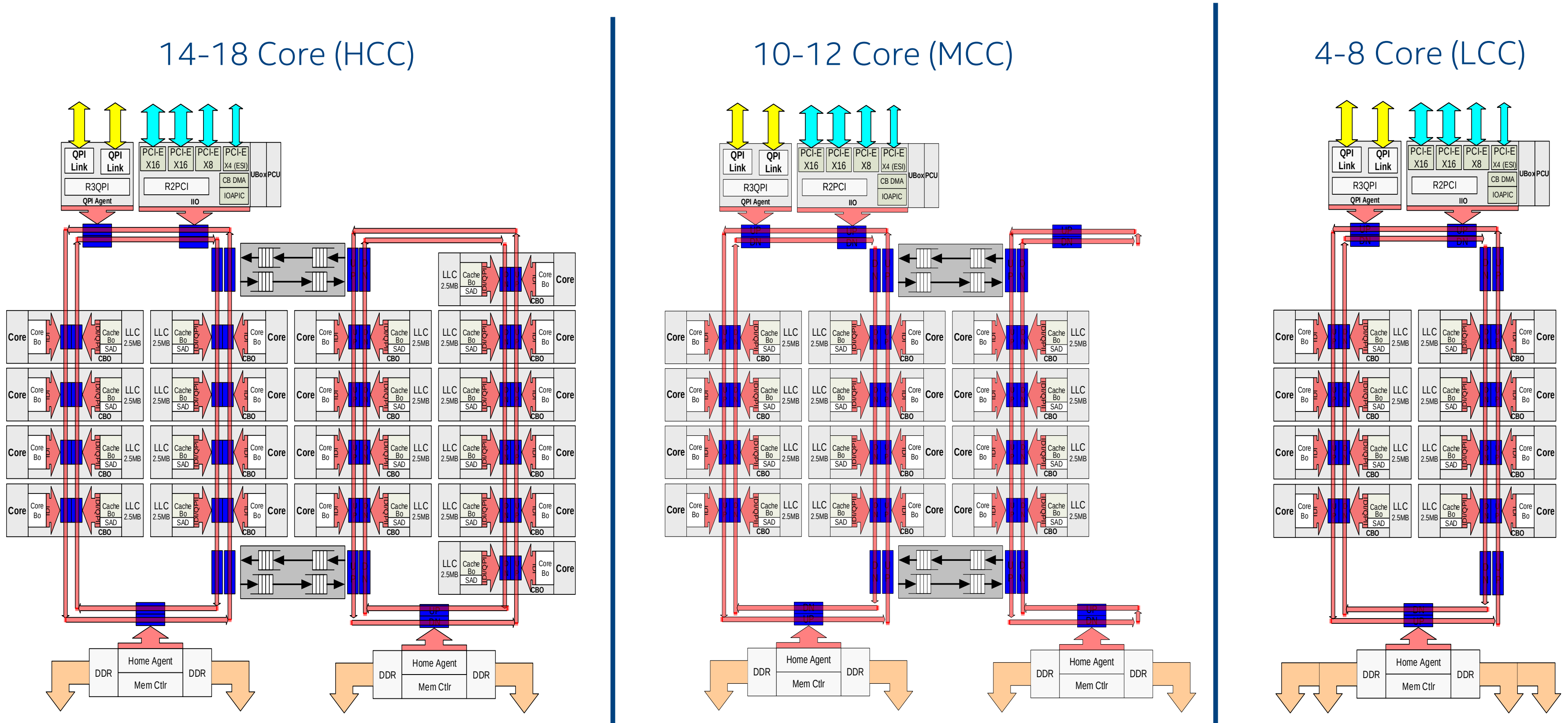
2xFMA
- Doubles peak FLOPs
- Two FP multiplies benefits legacy

4th ALU
- Great for integer workloads
- Frees Port0 & 1 for vector

New Branch Unit
- Reduces Port0 Conflicts
- 2nd EU for high branch code

New AGU for Stores
- Leaves Port 2 & 3 open for Loads

Haswell EP Die Configurations



Not representative of actual die-sizes, orientation and layouts – for informational use only.

Chop	Columns	Home Agents	Cores	Power (W)	Transitors (B)	Die Area (mm²)
HCC	4	2	14-18	110-145	5.69	662
MCC	3	2	6-12	65-160	3.84	492
LCC	2	1	4-8	55-140	2.60	354

Optimization Notice

Intel® Xeon® Processor E5-2600 v3 Product Family Snoop Modes

Each mode is configurable through BIOS settings

- Early Snoop Mode
  - Intel's BIOS default for HSW-EP
  - Same mode available on SNB-EP
  - Applications needing lowest memory latency or small chache-to-cache latency to **remote socket**
- Home Snoop Mode
  - Same mode available on IVB-EP*
  - Optimized for **NUMA** applications
- Cluster on Die Mode
  - New mode introduced on HSW-EP
  - Lowest memory latency and highest bandwidth; requires **NUMA** app.!

- Memory bandwidth & latency tradeoffs will vary across the 3 die configurations for each snoop mode
- Intel recommends exposing all snoop modes as BIOS options to the user

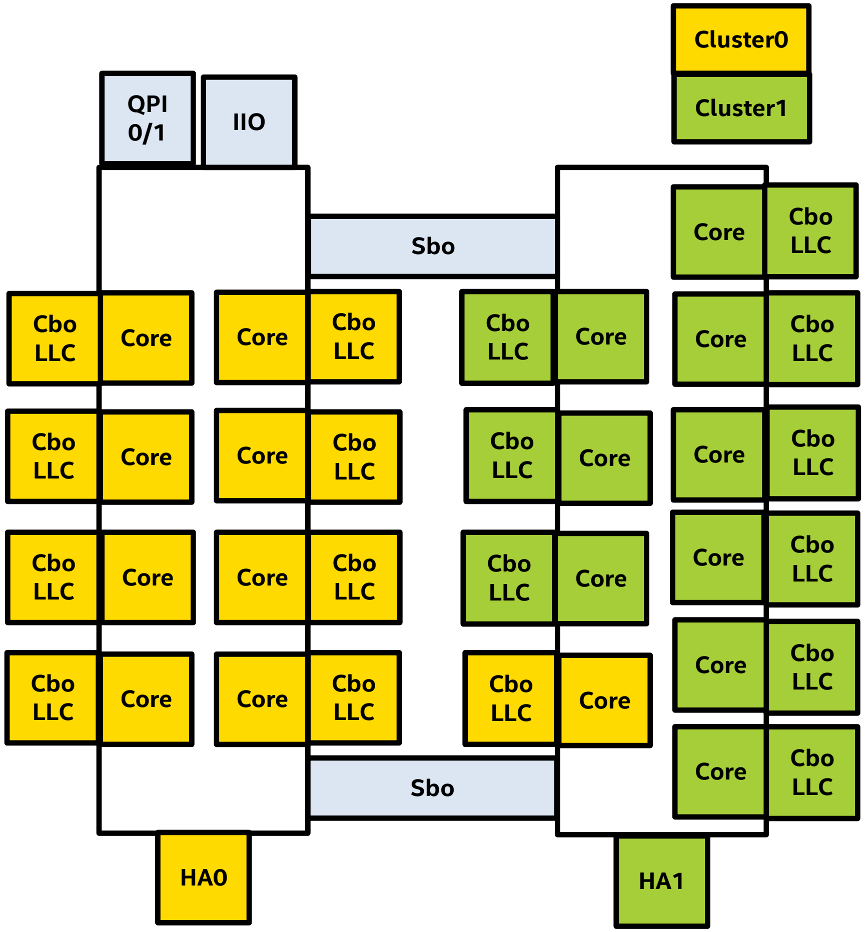
*Home Snoop mode is available on IVB-EP but is not the default setting

Optimization Notice

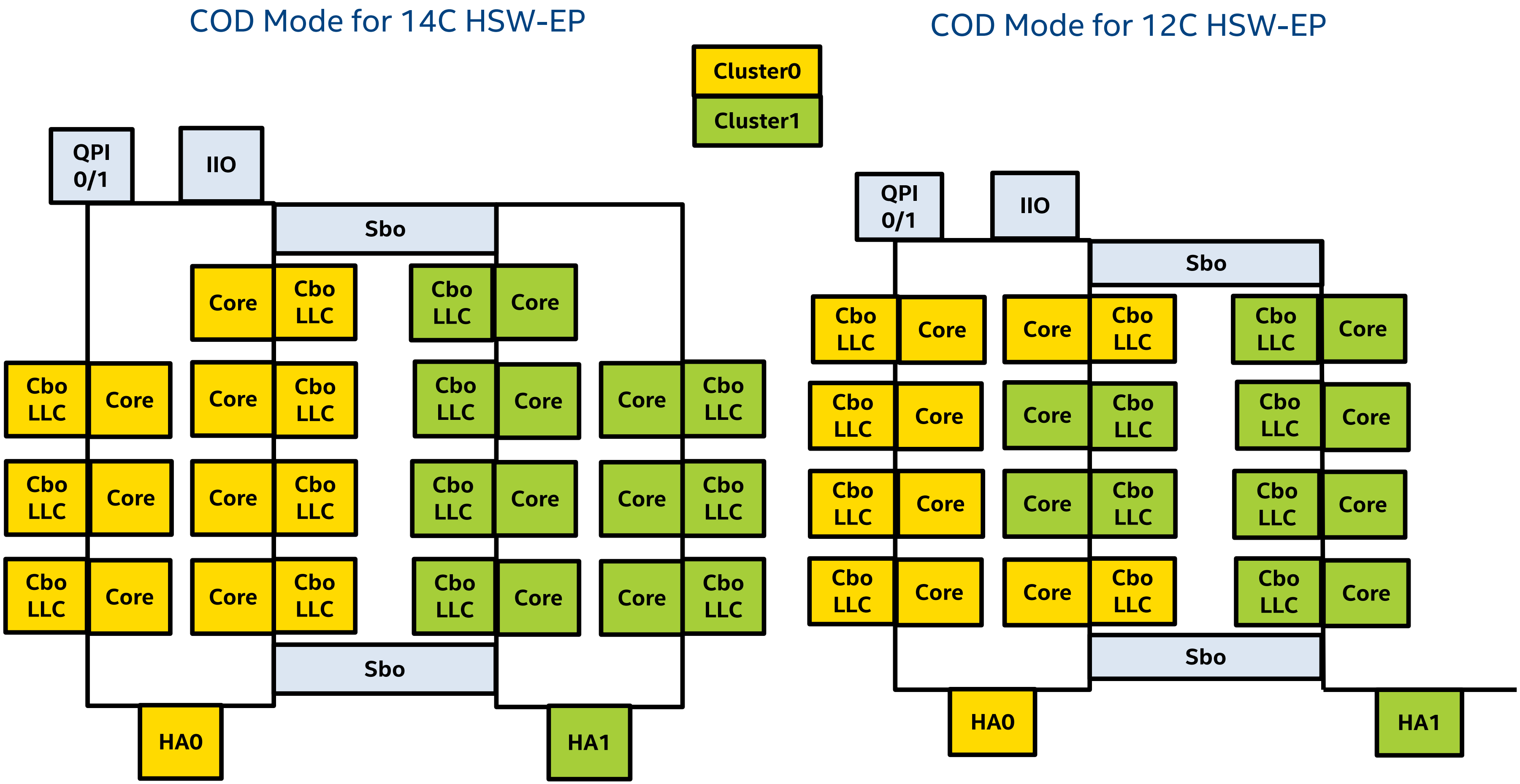
Cluster on Die (COD) Mode

- Supported on 2S HSW-EP SKUs with 2 Home Agents (10+ cores)
- Targeted at NUMA workloads where latency is more important than sharing data across Caching Agents (Cbo)
  - Reduces average LLC hit and local memory latencies
  - HA mostly sees requests from reduced set of threads which can lead to higher memory bandwidth
- OS/VMM own NUMA and process affinity decisions

COD Mode for 18C HSW-EP



Cluster on Die (COD) Mode



COD Mode for 14C HSW-EP

COD Mode for 12C HSW-EP

Feature Comparison across Intel® Xeon® Generations

	Intel® Xeon® Processor X5600 Series (Westmere-EP)	Intel® Xeon® Processor E5-2600 Product Family (Sandy Bridge-EP)	Intel® Xeon® Processor E5-2600 v2 Product Family (Ivy Bridge-EP)	Intel® Xeon® Processor E5-2600 v3 Product Family (Haswell-EP)
Essentials				
Launch Date	Q1'11	Q1'12	Q3'13	Q3'14
Maximum # of Cores	6	8	12	18
Maximum # of Threads	12	16	24	36
Last Level Cache (LLC)	Up to 12 MB	Up to 20 MB	Up to 30 MB	Up to 45 MB
Maximum QPI Bus Speed	6.4 GT/s	8 GT/s	8 GT/s	9.6 GT/s
Instruction Set Extensions	SSE4.2	Intel® AVX	Intel® AVX	Intel® AVX 2
Intel Process Technology	32 nm	32 nm	22 nm	22 nm
Intel® Turbo Boost Technology	1.0	2.0	2.0	2.0
Power Management	Same P-States for All Cores Fixed Uncore Frequency	Same P-States for All Cores Same Core and Uncore Frequency	Same P-States for All Cores Same Core and Uncore Frequency	Per Core P-States Independent Uncore Frequency Scaling
Memory Specifications				
Max Memory Size per Socket	288 GB	384 GB	768 GB	768 GB
Memory Types	DDR3 800/1066/1333 RDIMM/UDIMM	DDR3 800/1066/1333/1600 RDIMM/UDIMMs Quad Rank LRDIMM	DDR3 800/1066/1333/1600/1866 RDIMM/UDIMM Quad Rank LRDIMM	DDR4 1600/1866/2133 RDIMM Quad Rank LRDIMM
# of Memory Channels	3	4	4	4
Max # of DIMMs/Channel	3	3	3	3
Max # of DIMMs/Socket	9	12	12	12
Theoretical Max Memory Bandwidth per Socket	32 GB/s	51.2 GB/s	59.7 GB/s	68.2 GB/s
Expansion Options				
PCI Express Revision	2.0 (in Chipset)	3.0	3.0	3.0
Max # of PCI Express Lanes	N/A	40	40	40
PCI Express Configurations	N/A	x4, x8, x16	x4, x8, x16 x16 Non-Transparent Bridge	x4, x8, x16 x16 Non-Transparent Bridge

All comparisons based on a single socket.

System Configurations for HPC App.

	Intel® Xeon® E5-2697 v2	Intel® Xeon® E5-2667 v3		Intel® Xeon® E5-2680 v3		Intel® Xeon® E5-2695 v3		Intel® Xeon® E5-2697 v3		Intel® Xeon® E5-2699 v3	
Sockets / Cores	**2 x 12C**	**2 x 8C**		**2 x 12C**		**2 x 14C**		**2 x 14C**		**2 x 22C**	
Base Freq	**2.7GHz, C0**	**3.2GHz, R2**		**2.5GHz, M0**		**2.3GHz, C1**		**2.6GHz, C0**		**2.3GHz**	
Turbo Mode	**Enabled**	**Enabled**		**Enabled**		**Enabled**		**Enabled**		**Enabled**	
Memory	**64 GB DDR3-1867**	**128 GB DDR4-2133**		**128 GB DDR4-2133**		**128 GB DDR4-2133**		**128 GB DDR4-2133**		**128 GB DDR4-2133**	
Platform	**Romley-EP Server SDP**	**Wildcat Pass**		**Wildcat Pass**		**Wildcat Pass**		**Mayan City**		**Mayan City**	
	HT	**HT**	**Snoop Mode**	**HT**	**Snoop Mode**	**HT**	**Snoop Mode**	**HT**	**Snoop Mode**	**HT**	**Snoop Mode**
Computer Aided Engineering											
Finite Element Analysis	ON	ON	ES	OFF	COD	OFF	COD	ON	COD	ON	COD
Dynamic Analysis	ON	ON	ES	ON	ES	ON	ES	ON	HS	ON	HS
Computational Fluid Dynamics	ON	ON	ES	ON	COD	ON	COD	ON	COD	ON	COD
Multiphysics Simulation	ON	ON	ES	OFF	COD	OFF	COD	ON	COD	ON	COD
Crash Simulation	ON	ON	ES	ON	COD	ON	COD	ON	COD	ON	COD
Energy											
seis-kernel2 (Kirchhoff-PB) v1.1-12.0	ON	ON	ES	ON	COD	ON	COD	ON	COD	ON	COD
seis-kernel3 (TTI-T4-WG) v1.1-12.0	ON	OFF	ES	ON	COD	OFF	ES	OFF	HS	ON	HS
Financial Services											
binomialcpu v3.0-13.1.0_AVX / AVX2	ON	ON	ES	ON	COD	ON	COD	ON	HS	ON	HS
BlackScholes v5.0-13.1.1_AVX / AVX2	ON	ON	ES	ON	ES	ON	COD	ON	COD	ON	COD
FNR v4.0-12.0	ON	ON	ES	ON	COD	ON	ES	ON	COD	ON	ES
MonteCarlo v3.0-13.1.0-AVX / AVX2	ON	ON	ES	OFF	ES	ON	COD	OFF	COD	ON	COD
Life Sciences											
Amber v12-13.1.0_SSE4.2 / v12-14.0.0_AVX2	ON	ON	ES	OFF	ES	OFF	ES	ON	HS	ON	ES
Blast v2.2.28+_13.1.0_OPT2	ON	ON	ES	ON	ES	ON	COD	ON	COD	ON	COD
bowtie2 v2-2.1.0.0-13.1 / AVX2	ON	ON	ES	ON	COD	ON	COD	ON	HS	ON	ES
Gamess v01MAY2012-R1-12.1* / v01MAY2013.R1-14.0.1_AVX2	ON	ON	ES	ON	ES	ON	COD	ON	COD	ON	COD
Gaussian g09-D.01	ON	OFF	ES	OFF	COD	OFF	COD	OFF	COD	ON	HS
Gromacs v4.6.1-13.1.1_AVX	ON	ON	ES	ON	ES	ON	COD	ON	ES	ON	ES
NAMD v2.9-13.1.1_OPT3 / v2.9-14.0.0_AVX2	ON	ON	ES	OFF	ES	OFF	COD	ON	ES	ON	COD
MILC v7.7.8-13.1.1_OPT3 / v7.7.8-14.0.0_AVX2	ON	ON	ES	ON	COD	ON	COD	ON	COD	ON	COD
Numerical Weather											
HOMME v2841-20130227_AVX	ON	ON	ES	ON	COD	ON	COD	ON	COD	ON	COD
ROMS v3.0-12.0_AVX / v3.6.690-14.0.0_AVX2	ON	OFF	ES	OFF	COD	OFF	COD	ON	COD	ON	COD
WRF v3.1-11.1_AVX / v3.5-14.0.0_AVX2	ON	ON	ES	OFF	COD	OFF	COD	ON	COD	ON	COD

Optimization Notice

Intel® Xeon Phi™ Coprocessor

High Performance Computing

Intel® Xeon Phi™ Coprocessor

Generations

Pre 2013:

"Knights Ferry"

Engineering prototype

2013:

Intel® Xeon Phi™ Coprocessor x100 Product Family

"Knights Corner"

- 22 nm process
- 57-61 cores
- 6-16 GB memory
- 1 TeraFLOPs DP peak

2H 2015:

Intel® Xeon Phi™ Coprocessor x200 Product Family

"Knights Landing"

- 14 nm process
- 60+ cores
- On package, high-bandwidth memory
- Processor & coprocessor
- +3 TeraFLOPs DP peak

In planning

Upcoming Generation of the Intel® MIC Architecture

"Future Knights"

Continued roadmap commitment

Announced at SC14:
"Knights Hill"

- 10 nm process
- 2nd generation Intel® Omni-Path Architecture

Optimization Notice

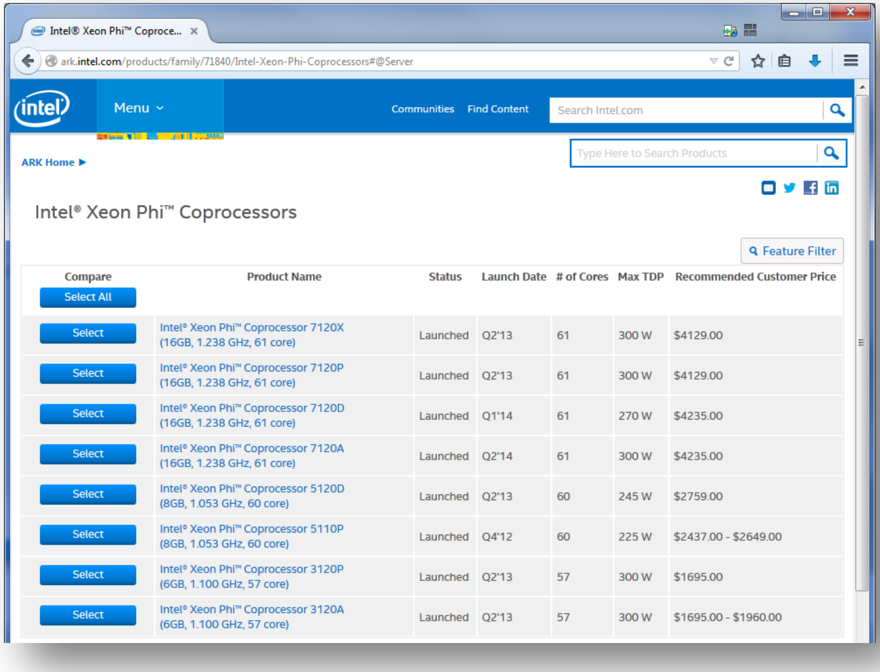
Intel® Xeon Phi™ Coprocessor

Your Source for Intel® Product Information

Family:

- Intel® Xeon Phi™ Coprocessor 3XXX:
  - Entry level
  - 57 cores
  - 6 GB memory
- Intel® Xeon Phi™ Coprocessor 5XXX:
  - Mid level
  - 60 cores
  - 8 GB memory
- Intel® Xeon Phi™ Coprocessor 7XXX:
  - Top level
  - 61 cores
  - 16 GB memory

Information about available Intel products can be found here: http://ark.intel.com/


Intel® Xeon Phi™ Coproce...
ark.intel.com/products/family/71840/Intel-Xeon-Phi-Coprocessors#@Server
Menu
Communities
Find Content
Search Intel.com
Type Here to Search Products
ARK Home ▶
Feature Filter

Intel® Xeon Phi™ Coprocessors

Compare Select All	Product Name	Status	Launch Date	# of Cores	Max TDP	Recommended Customer Price
Select	Intel® Xeon Phi™ Coprocessor 7120X (16GB, 1.238 GHz, 61 core)	Launched	Q2'13	61	300 W	$4129.00
Select	Intel® Xeon Phi™ Coprocessor 7120P (16GB, 1.238 GHz, 61 core)	Launched	Q2'13	61	300 W	$4129.00
Select	Intel® Xeon Phi™ Coprocessor 7120D (16GB, 1.238 GHz, 61 core)	Launched	Q1'14	61	270 W	$4235.00
Select	Intel® Xeon Phi™ Coprocessor 7120A (16GB, 1.238 GHz, 61 core)	Launched	Q2'14	61	300 W	$4235.00
Select	Intel® Xeon Phi™ Coprocessor 5120D (8GB, 1.053 GHz, 60 core)	Launched	Q2'13	60	245 W	$2759.00
Select	Intel® Xeon Phi™ Coprocessor 5110P (8GB, 1.053 GHz, 60 core)	Launched	Q4'12	60	225 W	$2437.00 - $2649.00
Select	Intel® Xeon Phi™ Coprocessor 3120P (6GB, 1.100 GHz, 57 core)	Launched	Q2'13	57	300 W	$1695.00
Select	Intel® Xeon Phi™ Coprocessor 3120A (6GB, 1.100 GHz, 57 core)	Launched	Q2'13	57	300 W	$1695.00 - $1960.00

Intel® Xeon Phi™ Coprocessor

Characteristics I

- Coprocessor core:
  - **2 issue** (with 2 cycle delay)
  - **In-order** execution
  - Simultaneous multithreading:
    4 HW threads per core
- Multi-core (**many-core**):
  - Up to 61 cores (57/60/61) per coprocessor
  - Up to 8 coprocessors already validated per host system (node)
- Caches:
  - **Two level** cache hierarchy – L1/L2
  - 64 byte cache line

Intel® Xeon Phi™ Coprocessor

Characteristics II

- Page sizes:
  - 4 kB
  - 64 kB
  - 2 MB
- Others:
  - Coprocessor(s) connected to node via PICe

Architecture:
https://software.intel.com/sites/default/files/article/393195/intel-xeon-phi-core-micro-architecture.pdf

Data sheet:
http://www.intel.com/content/dam/www/public/us/en/documents/datasheets/xeon-phi-coprocessor-datasheet.pdf

Intel® Xeon Phi™ Coprocessor

Intel® Xeon® Processor vs. Intel® Xeon Phi™ Coprocessor

Intel® Xeon® Processor:

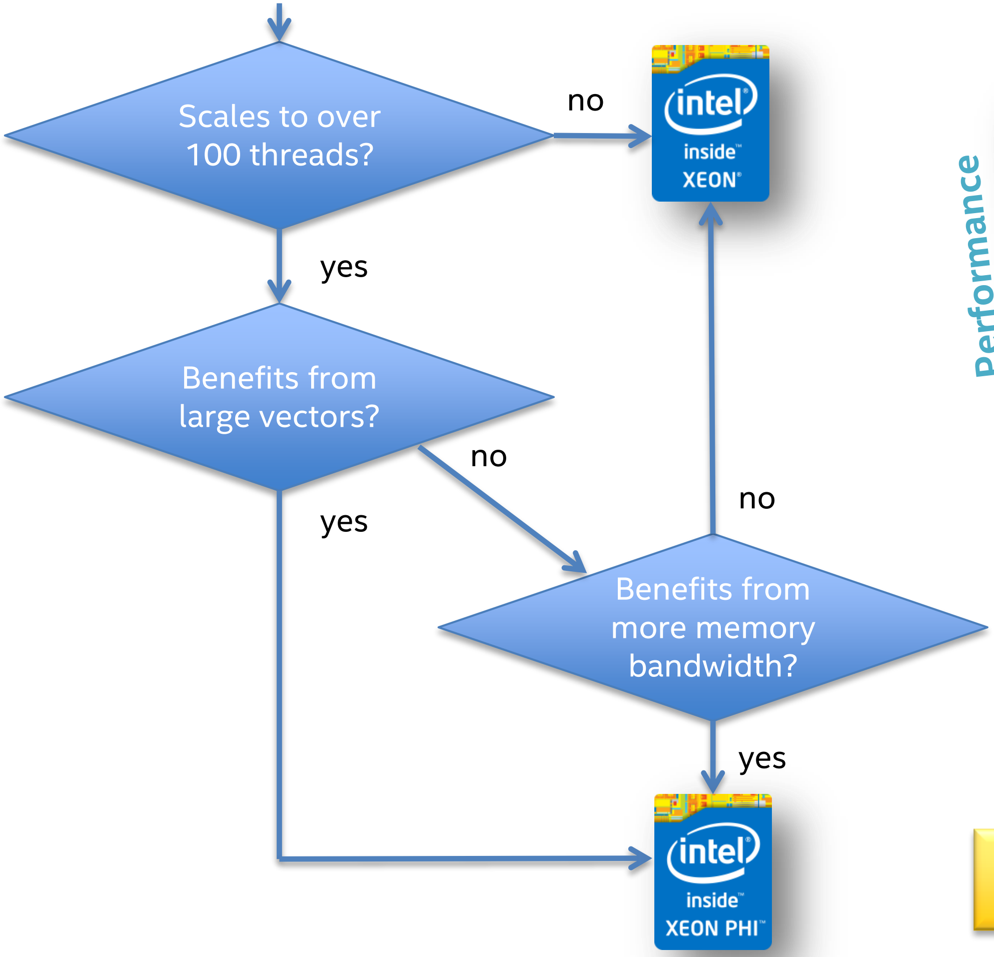
- Optimal for workloads with…
  - High single-thread performance
  - High memory capacity
- Core/memory connections via sockets and nodes
- Instruction set:
  - SIMD SSE 128-bit & AVX 256-bit
  - Gather, FMA, virtualization, AES, etc.

Intel® Xeon Phi™ Coprocessor:

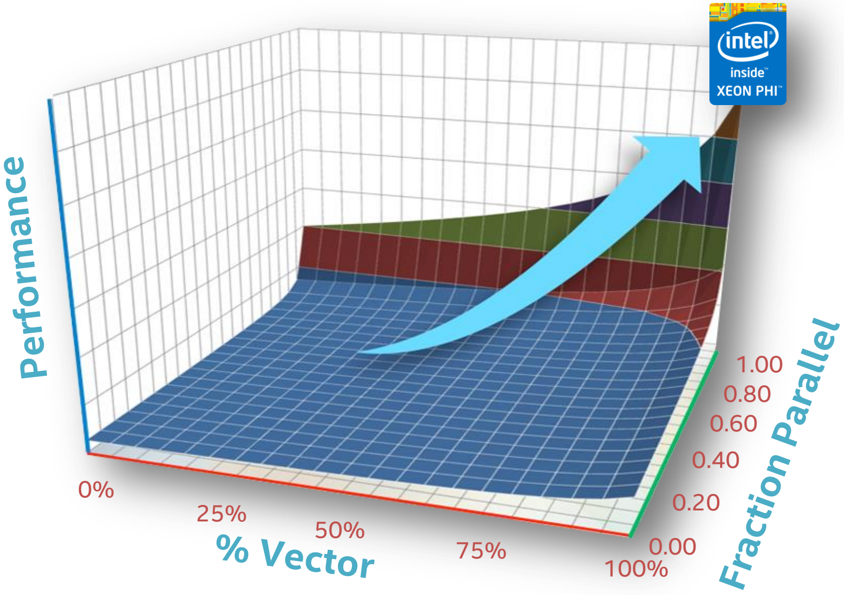
- Optimal for workloads with…
  - High parallelization
  - High memory bandwidth
- Up to 61 cores per die, connection via PCIe and nodes
- Instruction set:
  - SIMD 512-bit
  - Gather and scatter, FMA, masked instructions

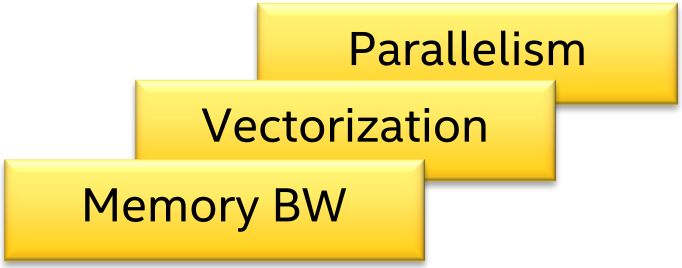
Intel® Xeon Phi™ Coprocessor

Domain



Scales to over 100 threads?
no
intel inside XEON®
yes
Benefits from large vectors?
no
yes
no
Benefits from more memory bandwidth?
yes
intel inside XEON PHI™

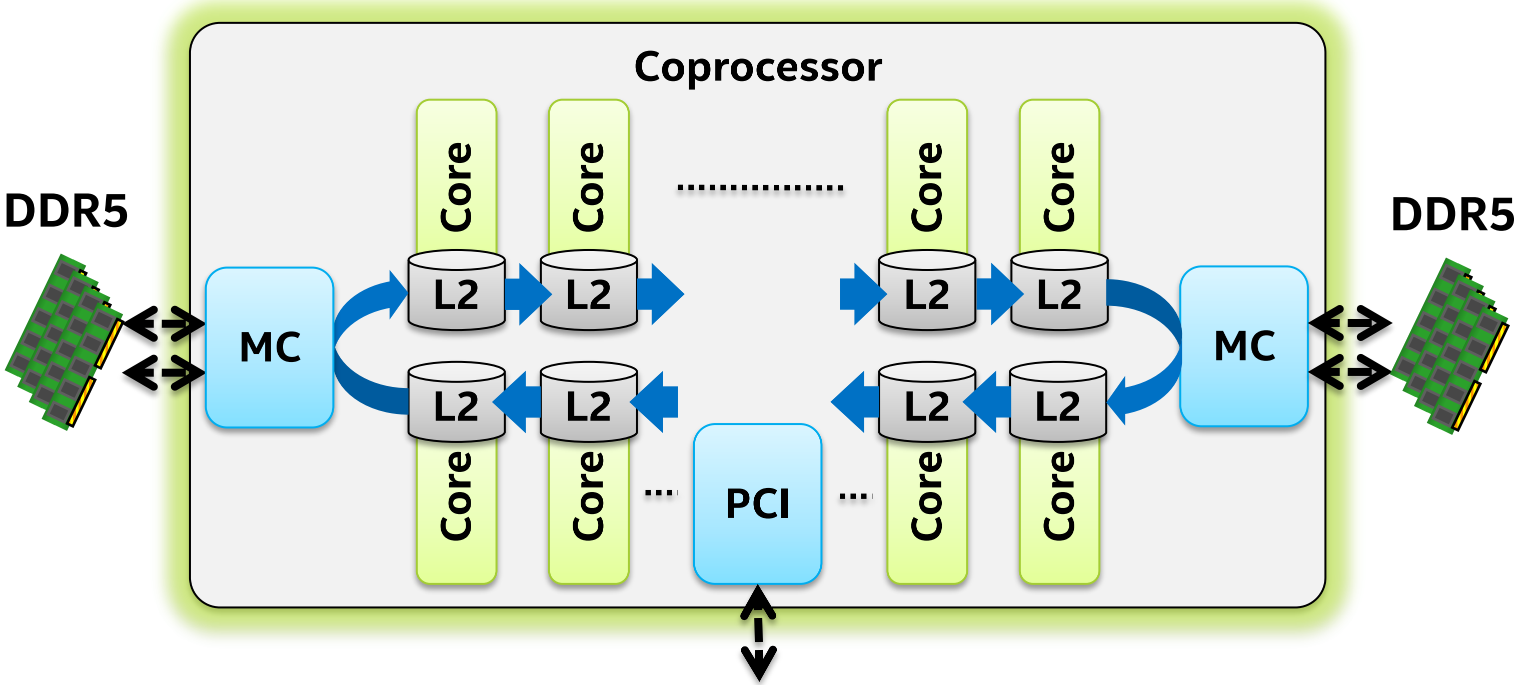




Intel® Xeon Phi™ coprocessor complements Intel® Xeon® processor!

Intel® Xeon Phi™ Coprocessor

Memory



- 57-61 cores
- 8-16 GB GDDR5 memory (ECC)
- PCIe Gen2 (client) x16 per direction
- 8 memory controllers (MC)
- 2 GDDR5 channels per MC
- Up to 5.5 GT/s per channel

Intel® Xeon Phi™ **theoretic** bandwidth:
8 MC * 2 channels * 5.5 GT/s * 4 byte = **352 GB/s**

Intel® Xeon Phi™ Coprocessor

Memory Considerations

Limitations of the theoretic memory bandwidth:

- HW related (signal noice, DDR5 overhead)

- Depending on the Intel® Xeon Phi™ coprocessor family:
  Low frequency models cannot saturate the bandwidth with loads/stores.

- ECC on/off

- Page size (4k by default)
  ⇨ Use 2MB pages (large pages)

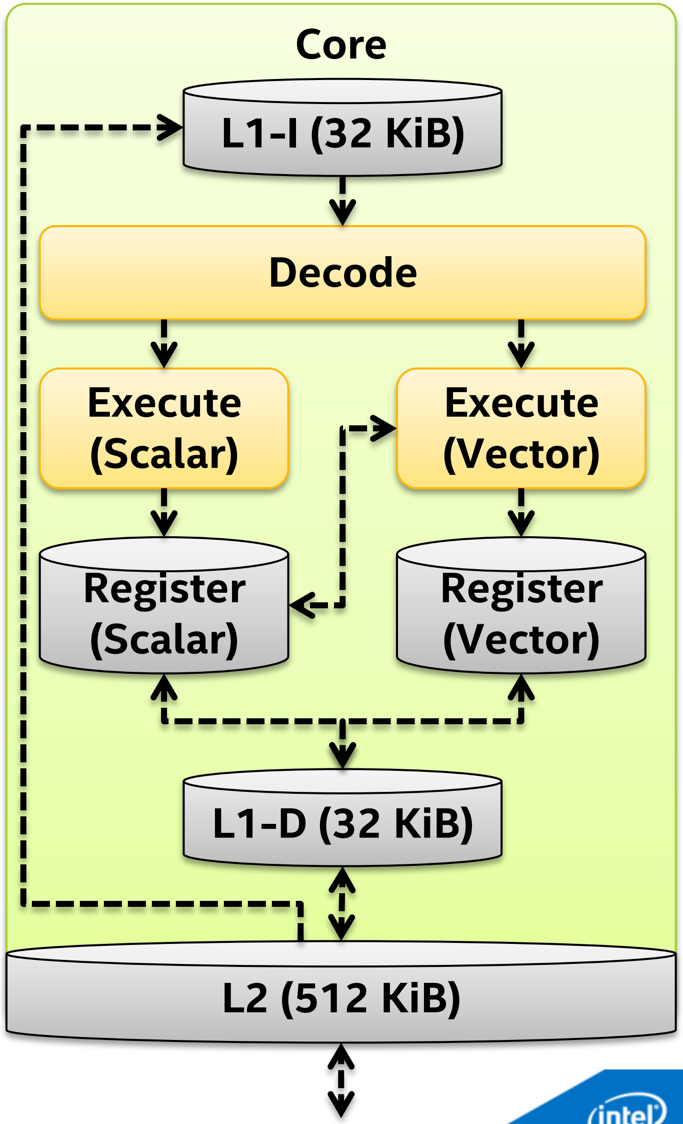
- Application not pure memory bound

How to benchmark using **Stream Triad**:
https://software.intel.com/en-us/articles/optimizing-memory-bandwidth-on-stream-triad

Intel® Xeon Phi™ Coprocessor

Architecture

- Pentium scalar instruction set with x87
- Extended with full 64 bit addressing
- Decoding:
  - In order-operation
  - 2 cycle decoder, 2 issue (1 scalar & 1 vector)
- Simultaneous multi-threading:
  - 4 HW threads per core
  - 2 instruction prefetch per HW thread
  - Round robin
- Instruction latencies:
  - Scalar: 1 cycle
  - Vector: 4 cycle (throughput of 1 cycle)
- Two pipelines:
  - Scalar (V pipeline)
  - Vector/Scalar (U pipeline)

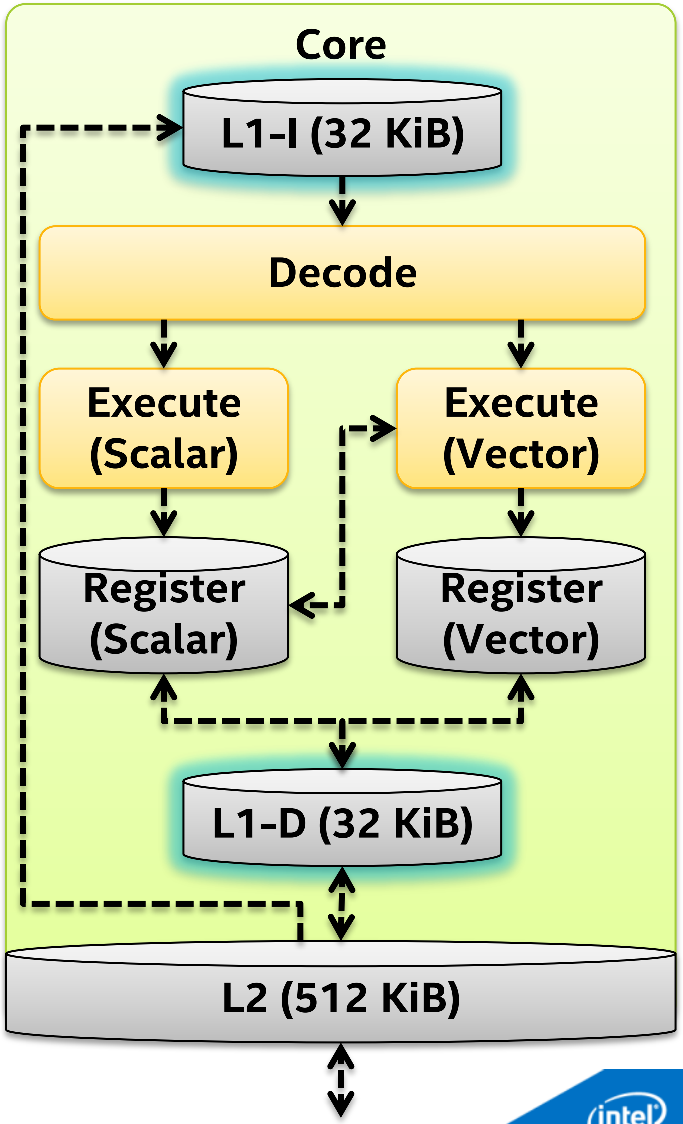


Optimization Notice

Intel® Xeon Phi™ Coprocessor

L1 Cache

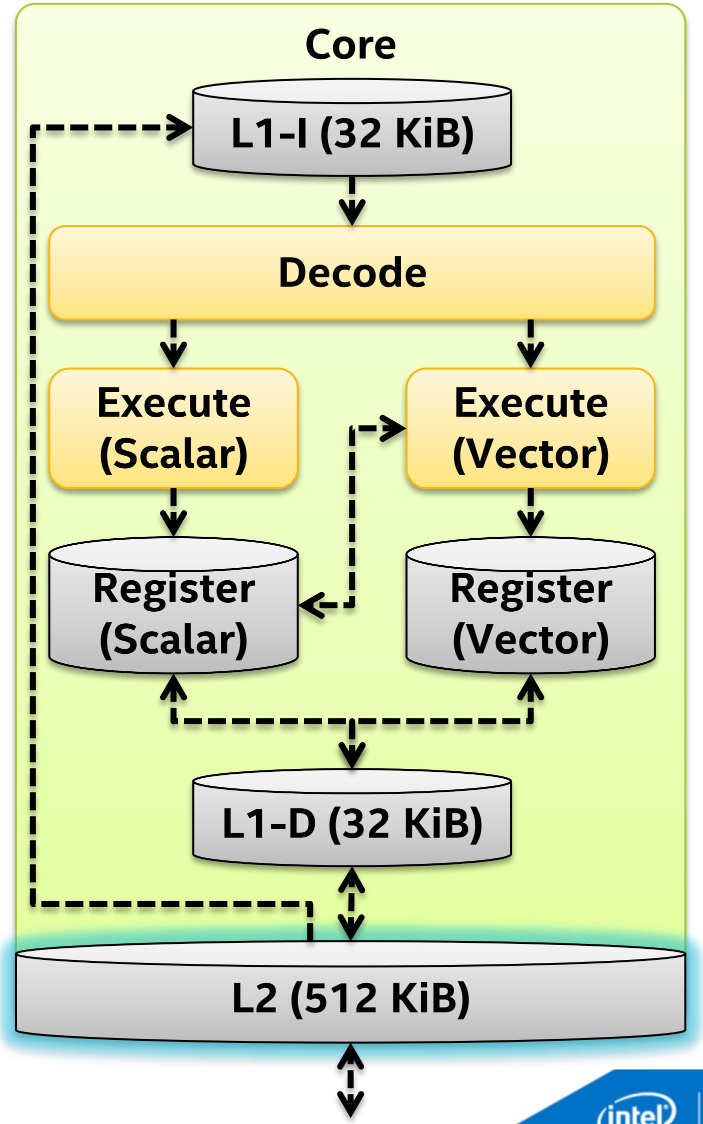
- Size:
  - 32 KiB I-cache per core
  - 32 KiB D-cache per core
- 8 way associative
- 64 byte cache line
- 3 cycle access latency (address generation)
- Up to 8 outstanding requests
- Fully coherent
- Inclusive (L2 contains L1 data)



Intel® Xeon Phi™ Coprocessor

L2 Cache

- Size:
  - 512 KiB unified per core
  - Total L2: 512 KiB x # cores (~30 MiB)
- 8 way associative
- 64 byte cache line
- 11 cycle access latency
- Up to 32 outstanding requests
- Streaming HW prefetcher
- Fully coherent
- Inclusive (L2 contains L1 data)

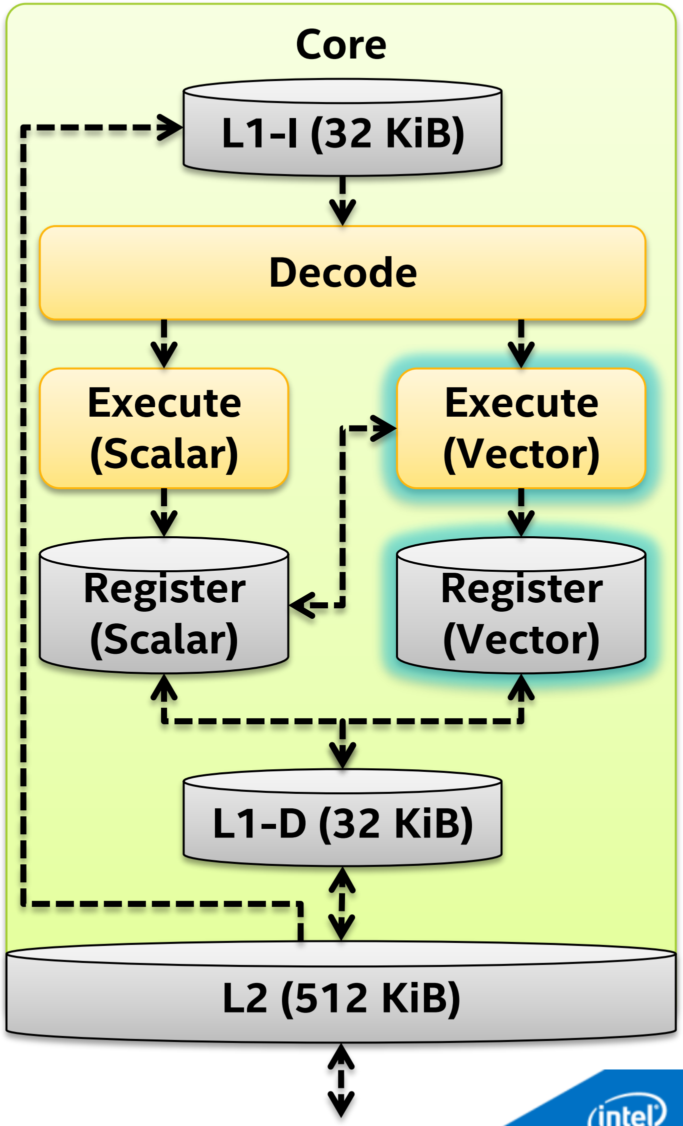


Optimization Notice

Intel® Xeon Phi™ Coprocessor

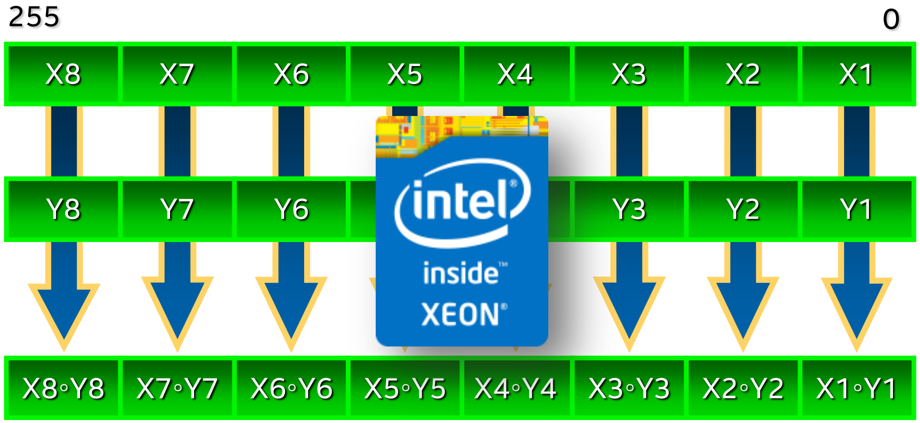
Vector Unit

- 32 512 bit vector registers per HW thread
- Each holds 16 SP FP or 8 DP FP
- ALUs support:
  - 32 bit integer/FP operations
  - 64 bit integer/FP/logic operations
- Ternary operations including Fused Multiply Add (FMA)
- Broadcast/swizzle support and 16 bit FP up-convert
- 8 vector mask registers for per lane conditional operations
- Most ops have a 4-cycle latency and 1-cycle throughput
- Mostly IEEE 754 2008 compliant
- Not supported:
  - MMX™ technology
  - Streaming SIMD Extensions (SSE)
  - Intel® Advanced Vector Extensions (Intel® AVX)



Intel® Xeon Phi™ Coprocessor

SIMD Vectors



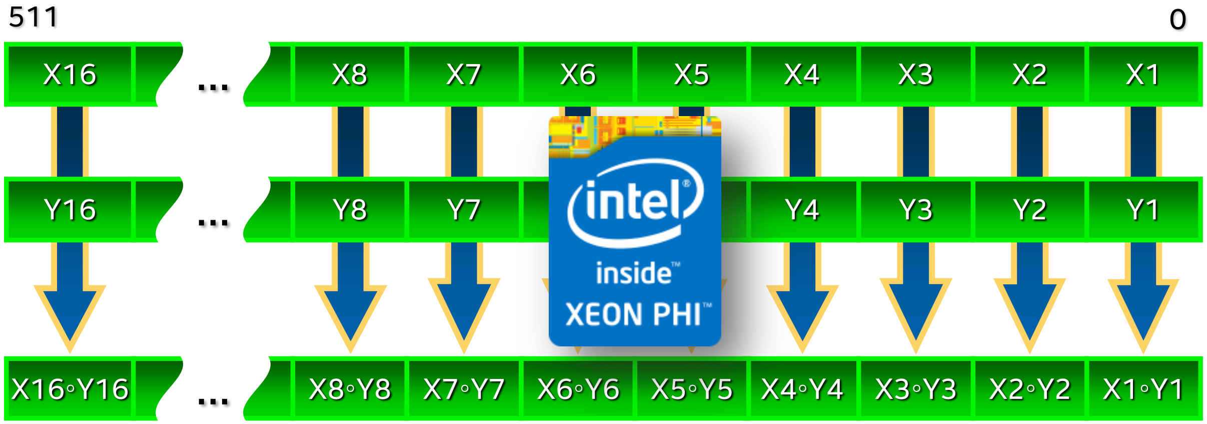
AVX

Vector size: **256 bit**

Data types:

- 8, 16, 32, 64 bit integer
- 32 and 64 bit float

VL: 4, 8, 16, 32



Intel® MIC Architecture

Vector size: **512 bit**

Data types:

- 32 bit integer
- 32 and 64 bit float

VL: 8, 16

Example of Intel® Xeon Phi™ coprocessor **theoretic peak** FLOP rate:
1.238 GHz * 16 DP FLOPs * 61 cores * = **1.208 TeraFLOPs**

Illustrations: Xi, Yi & results 32 bit integer

Intel® Xeon Phi™ Coprocessor

Instruction Set

- Pentium scalar instruction set
- 64 bit extensions (e.g. 64 bit registers **`rax`**, **`rbx`**, …)

- 512 bit SIMD vector registers:
  `zmm0` … **`zmm31`**

- Mask registers:
  `k0` … **`k7`** (**`k0`** is special, don't use)

- Backward compatibility to big core:
  - Missing SSE (128 bit) and AVX (256 bit) with "Knights Corner"
  - Compatibility with "Knights Landing"

- Legacy x87 also exists for scalar FP operations
  ⇨ **For good performance, don't use!**

Illustrations: Xi, Yi & results 32 bit integer

Future "Knights Landing"

NEW!
as of March 2015

PERFORMANCE

3+ TeraFLOPS of double-precision peak theoretical performance per single socket node[0]

INTEGRATION

Intel® Omni Scale™ fabric integration

High-performance on-package memory (MCDRAM)	Over 5x STREAM vs. DDR4[1] → Over 400 GB/s
	Up to 16GB at launch
	NUMA support
	Over 5x Energy Efficiency vs. GDDR5[2]
	Over 3x Density vs. GDDR5[2]
	In partnership with Micron Technology
	Flexible memory modes including cache and flat

SERVER PROCESSOR

Standalone bootable processor (running host OS) and a PCIe coprocessor (PCIe end-point device)

Platform memory: up to 384GB DDR4 using 6 channels

Reliability ("Intel server-class reliability")

Power Efficiency (Over 25% better than discrete coprocessor)[4] → Over 10 GF/W

Density (3+ KNL with fabric in 1U)[5]

Up to 36 lanes PCIe* Gen 3.0

MICROARCHITECTURE

Over 8 billion transistors per die based on Intel's 14 nanometer manufacturing technology

Binary compatible with Intel® Xeon® Processors with support for Intel® Advanced Vector Extensions 512 (Intel® AVX-512)[6]

3x Single-Thread Performance compared to Knights Corner[7]

60+ cores in a 2D Mesh architecture

2 cores per tile with 2 vector processing units (VPU) per core)

1MB L2 cache shared between 2 cores in a tile (cache-coherent)

"Based on Intel® Atom™ core (based on Silvermont microarchitecture) with many HPC enhancements"	4 Threads / Core
	2X Out-of-Order Buffer Depth[8]
	Gather/scatter in hardware
	Advanced Branch Prediction
	High cache bandwidth
	32KB Icache, Dcache
	2 x 64B Load ports in Dcache
	46/48 Physical/virtual address bits

Most of today's parallel optimizations carry forward to KNL

Multiple NUMA domain support per socket

All products, computer systems, dates and figures specified are preliminary based on current expectations, and are subject to change without notice. All projections are provided for informational purposes only. Any difference in system hardware or software design or configuration may affect actual performance.

[0] Over 3 Teraflops of peak theoretical double-precision performance is preliminary and based on current expecations of cores, clock frequency and floating point operations per cycle.

[1] Projected result based on internal Intel analysis of STREAM benchmark using a Knights Landing processor with 16GB of ultra high-bandwidth versus DDR4 memory with all channels populated.

[2] Projected result based on internal Intel analysis comparison of 16GB of ultra high-bandwidth memory to 16GB of GDDR5 memory used in the Intel® Xeon Phi™ coprocessor 7120P.

[3] Compared to 1st Generation Intel® Xeon Phi™ 7120P Coprocessor (formerly codenamed Knights Corner)

[4] Projected result based on internal Intel analysis using estimated performance and power consumption of a rack sized deployment of Intel® Xeon® processors and Knights Landing coprocessors as compared to a rack with KNL processors only

[5] Projected result based on internal Intel analysis comparing a discrete Knights Landing processor with integrated fabric to a discrete Intel fabric component card.

[6] Binary compatible with Intel® Xeon® Processors v3 (Haswell) with the exception of Intel® TSX (Transactionaly Synchronization Extensions)

[7] Projected peak theoretical single-thread performance relative to 1st Generation Intel® Xeon Phi™ Coprocessor 7120P

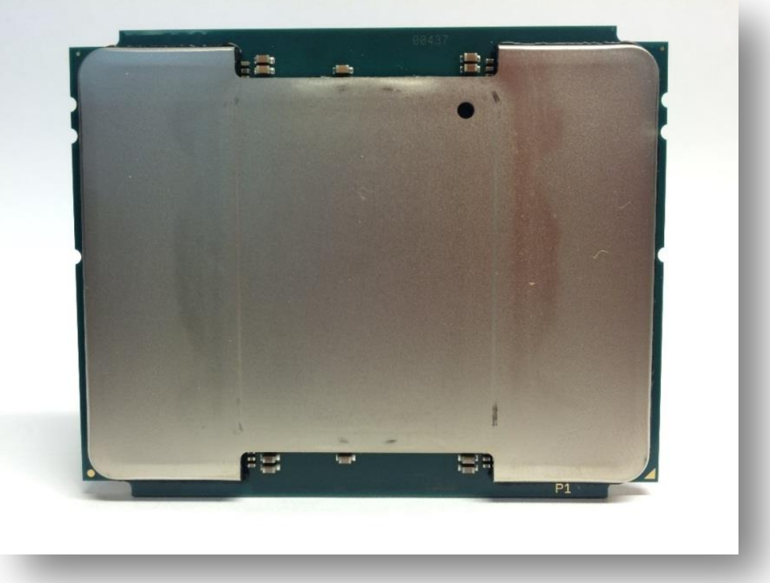
[8] Compared to the Intel® Atom™ core (base on Silvermont microarchitecture)

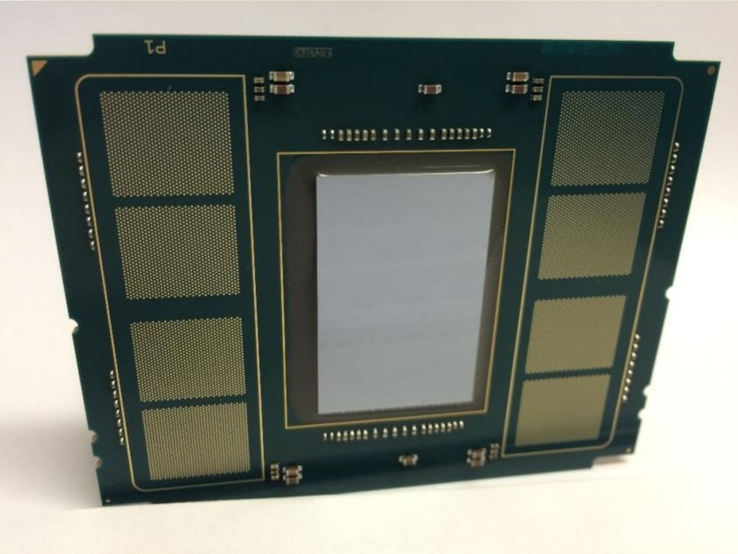
Future “Knights Landing” – cont’d

NEW!
as of March 2015

FUTURE	
Knights Hill is the codename for the 3rd generation of the Intel® Xeon Phi™ product family	Based on Intel’s 10 nanometer manufacturing technology
	Integrated 2nd generation Intel® Omni-Path Host Fabric Interface

AVAILABILITY
First commercial HPC systems in 2H’15
Knights Corner to Knights Landing upgrade program available today
Intel Adams Pass board (1U half-width) is custom designed for Knights Landing (KNL) and will be available to system integrators for KNL launch; the board is OCP Open Rack 1.0 compliant, features 6 ch native DDR4 (1866/2133/2400MHz) and 36 lanes of integrated PCIe* Gen 3 I/O

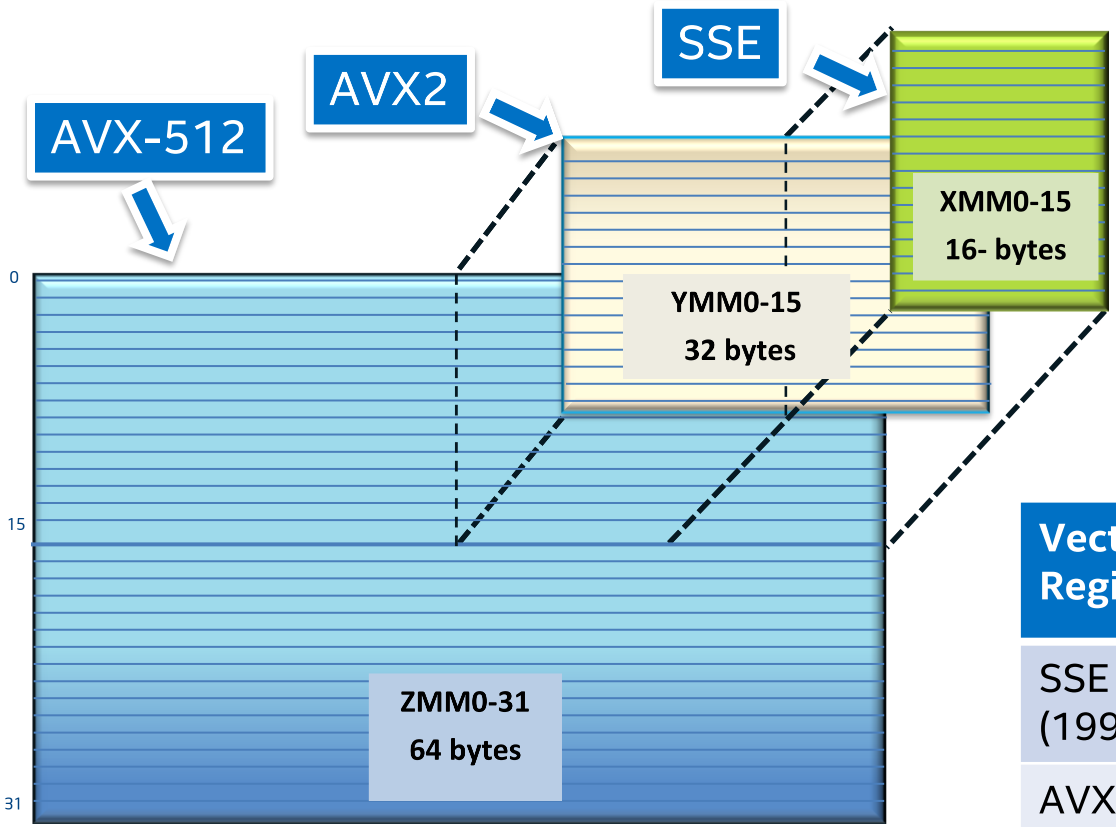




All products, computer systems, dates and figures specified are preliminary based on current expectations, and are subject to change without notice.
All projections are provided for informational purposes only. Any difference in system hardware or software design or configuration may affect actual performance.

Optimization Notice

AVX-512 - Greatly increased Register File



Vector Registers	IA32 (32bit)	Intel64 (64bit)
SSE (1999)	8 x 128bit	16 x 128bit
AVX and AVX-2 (2011 / 2013)	8 x 256bit	16 x 256bit
AVX-512 (2014 – KNL)	8 x 512bit	32 x 512bit

The Intel® AVX-512 Subsets [1]

AVX-512 F

AVX-512 F: 512-bit Foundation instructions common between MIC and Xeon

- ❑ Comprehensive vector extension for HPC and enterprise
- ❑ All the key AVX-512 features: masking, broadcast…
- ❑ 32-bit and 64-bit integer and floating-point instructions
- ❑ Promotion of many AVX and AVX2 instructions to AVX-512
- ❑ Many new instructions added to accelerate HPC workloads

AVX-512CD

AVX-512 CD (Conflict Detection instructions)

- ❑ Allow vectorization of loops with possible address conflict
- ❑ Will show up on Xeon

AVX-512ER

AVX-512PR

AVX-512 extensions for exponential and prefetch operations

- ❑ fast (28 bit) instructions for exponential and reciprocal and transcendentals (as well as RSQRT)
- ❑ New prefetch instructions: gather/scatter prefetches and PREFETCHWT1

Optimization Notice

The Intel® AVX-512 Subsets [2]

AVX-512DQ

AVX-512 Double and Quad word instructions

- ❑ All of (packed) 32bit/64 bit operations AVX-512F doesn't provide
- ❑ Close 64bit gaps like VPMULLQ : packed 64x64 ➔ 64
- ❑ Extend mask architecture to word and byte (to handle vectors)
- ❑ Packed/Scalar converts of signed/unsigned to SP/DP

AVX-512BW

AVX-512 Byte and Word instructions

- ❑ Extent packed (vector) instructions to byte and word (16 and 8 bit) data type
    - ❑MMX/SSE2/AVX2 re-promoted to AVX512 semantics
- ❑ Mask operations extended to 32/64 bits to adapt to number of objects in 512bit
- ❑ Permute architecture extended to words (VPERMW, VPERMI2W, …)

AVX-512VL

AVX-512 Vector Length extensions

- ❑ Vector length orthogonality
    - ❑Support for 128 and 256 bits instead of full 512 bit
- ❑ Not a new instruction set but an attribute of existing 512bit instructions

Other New Instructions

MPX

Intel® MPX – Intel Memory Protection Extension

❑Set of instructions to implement checking a pointer against its bounds
❑Pointer Checker support in HW (today a SW only solution of e.g. Intel Compilers)
❑Debug and security features

SHA

Intel® SHA – Intel Secure Hash Algorithm

❑ Fast implementation of cryptographic hashing algorithm as defined by NIST FIPS PUB 180

CLFLUSHOPT

Single Instruction – Flush a cache line

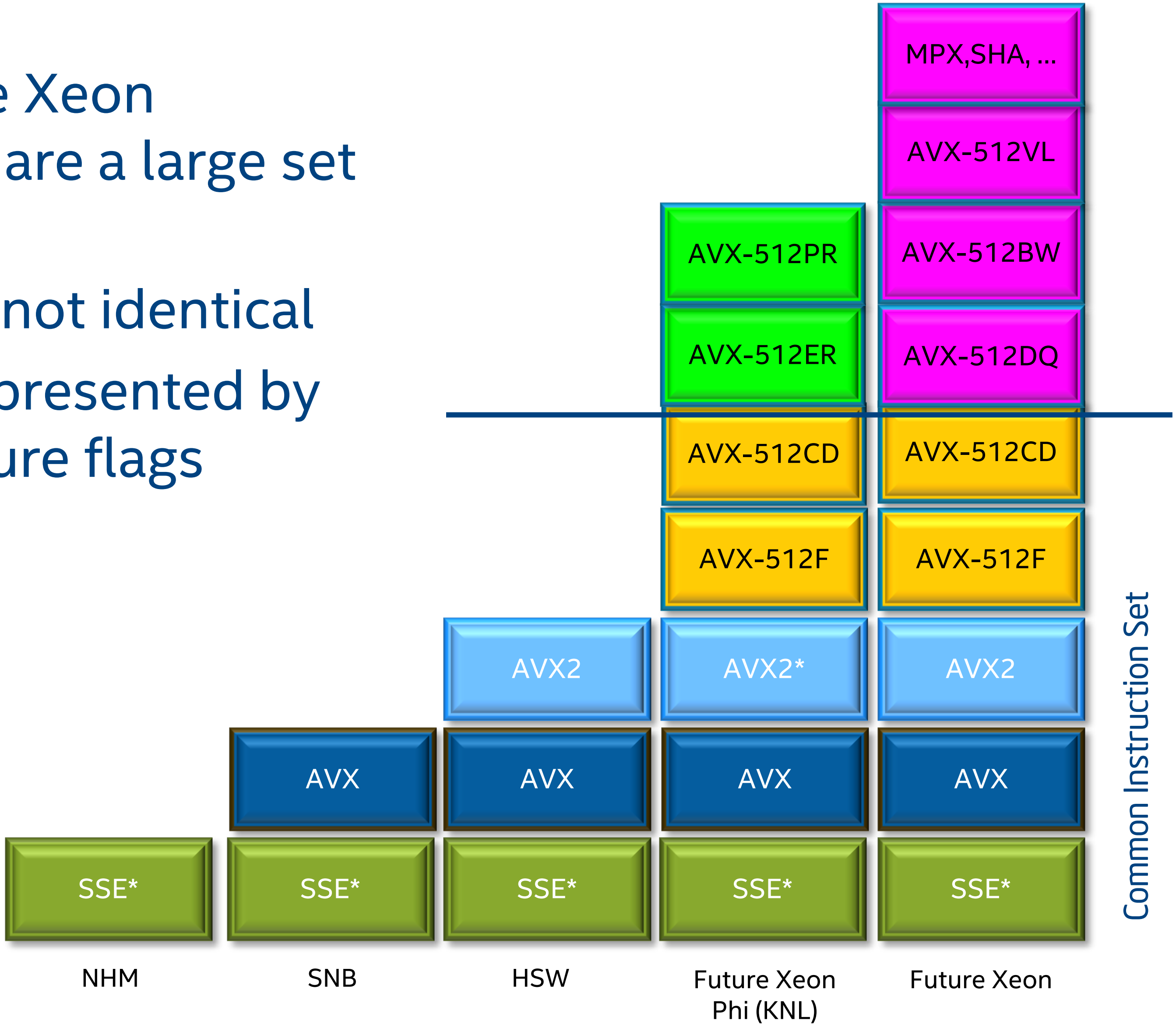
❑ needed for future memory technologies

XSAVE{S,C}

Save and restore extended processor state

AVX-512 – KNL and future XEON

- KNL and future Xeon architecture share a large set of instructions
  - but sets are not identical
- Subsets are represented by individual feature flags (CPUID)



Intel® Compiler Processor Switches

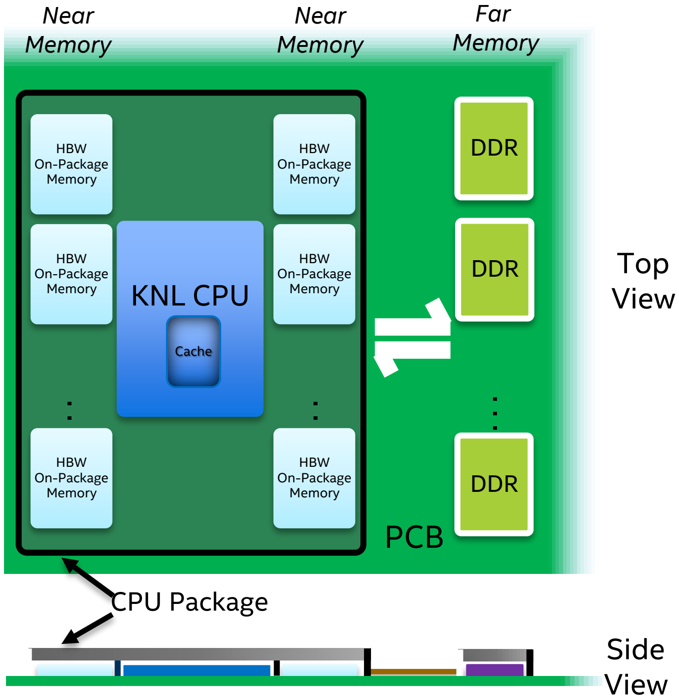
Switch	Description
`-xmic-avx512`	KNL only; already in 14.0
`-xcore-avx512`	Future XEON only, already in 15.0.1
`-xcommon-avx512`	AVX-512 subset common to both, already in 15.0.2
`-m,  -march,   /arch`	Not yet !
`-ax<…-avx512>`	Same as for “-x<…-avx512>”
`-mmic`	No – not for KNL

Knights Landing Integrated On-Package Memory

Cache Model Let the hardware automatically manage the integrated on-package memory as an "L3" cache between KNL CPU and external DDR

Flat Model Manually manage how your application uses the integrated on-package memory and external DDR for peak performance

Hybrid Model Harness the benefits of both cache and flat models by segmenting the integrated on-package memory



Maximizes performance through higher memory bandwidth and flexibility[1]

[1] As compared with Intel® Xeon Phi™ x100 Coprocessor Family

Diagram is for conceptual purposes only and only illustrates a CPU and memory – it is not to scale, and is not representative of actual component layout.

High Bandwidth On-Chip Memory API

- API is open-sourced (BSD licenses)
  - https://github.com/memkind
  - Uses jemalloc API underneath
    - http://www.canonware.com/jemalloc/
    - https://www.facebook.com/notes/facebook-engineering/scalable-memory-allocation-using-jemalloc/480222803919

Malloc replacement:

```
#include <memkind.h>

  hbw_check_available()
  hbw_malloc, _calloc, _realloc,… (memkind_t kind, …)
  hbw_free()
  hbw_posix_memalign()
  hbw_get_size(), _psize()

ld …  -ljemalloc -lnuma -lmemkind -lpthread
```

HBW API for Fortran, C++

Fortran:

!DIR$ ATTRIBUTES FASTMEM :: data_object1, data_object2

- All Fortran data types supported
- Global, local, stack or heap;   scalar, array, ...
- Support in compiler 15.0 update 1 and later versions

C++:

standard allocator replacement for e.g. STL like

```
#include <hbwmalloc.h>
std::vector<int, hbwmalloc::hbw_allocator>
```

Available already but not documented yet – working on documentation just now

Intel® Software Development Emulator (SDE)

Use Intel® Software Development Emulator (SDE) to test AVX-512 enabled code

- Will test instruction mix, not performance
- Does not emulate hardware (e.g. memory hierarchy) only ISA

Use the SDE to answer

- Is compiler generating Intel® AVX-512/KNL-ready code for my source code already ?
- How do I restructure my code so that Intel® AVX-512 code is generated?

Visit *Intel Xeon Phi Coprocessor code named “Knights Landing” - Application Readiness*

Agenda

- Introduction
- Intel® Architecture
  - Desktop, Mobile & Server
  - Intel® Xeon Phi™ Coprocessor
- **Summary**

Summary

- More cores to come in the future
- Make sure your application scales with more cores
- Identify whether new technology is applicable for you and use it
- Parallelism is not just adding cores…

Thank you!

Legal Disclaimer & Optimization Notice

INFORMATION IN THIS DOCUMENT IS PROVIDED "AS IS". NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO THIS INFORMATION INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products.

Optimization Notice

Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Notice revision #20110804

intel®