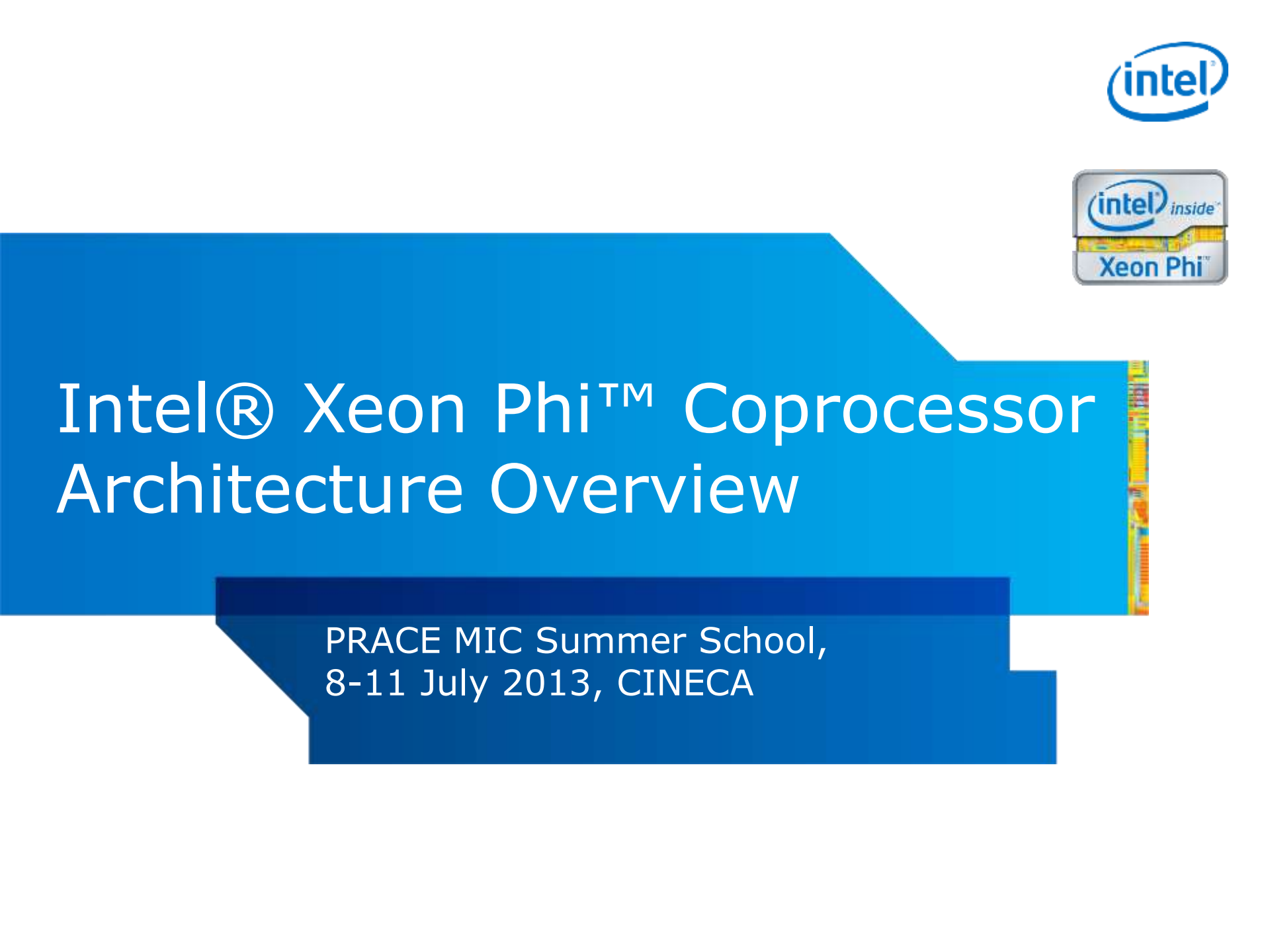


Intel® Xeon Phi™ Coprocessor Architecture Overview

PRACE MIC Summer School,
8-11 July 2013, CINECA

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

Agenda

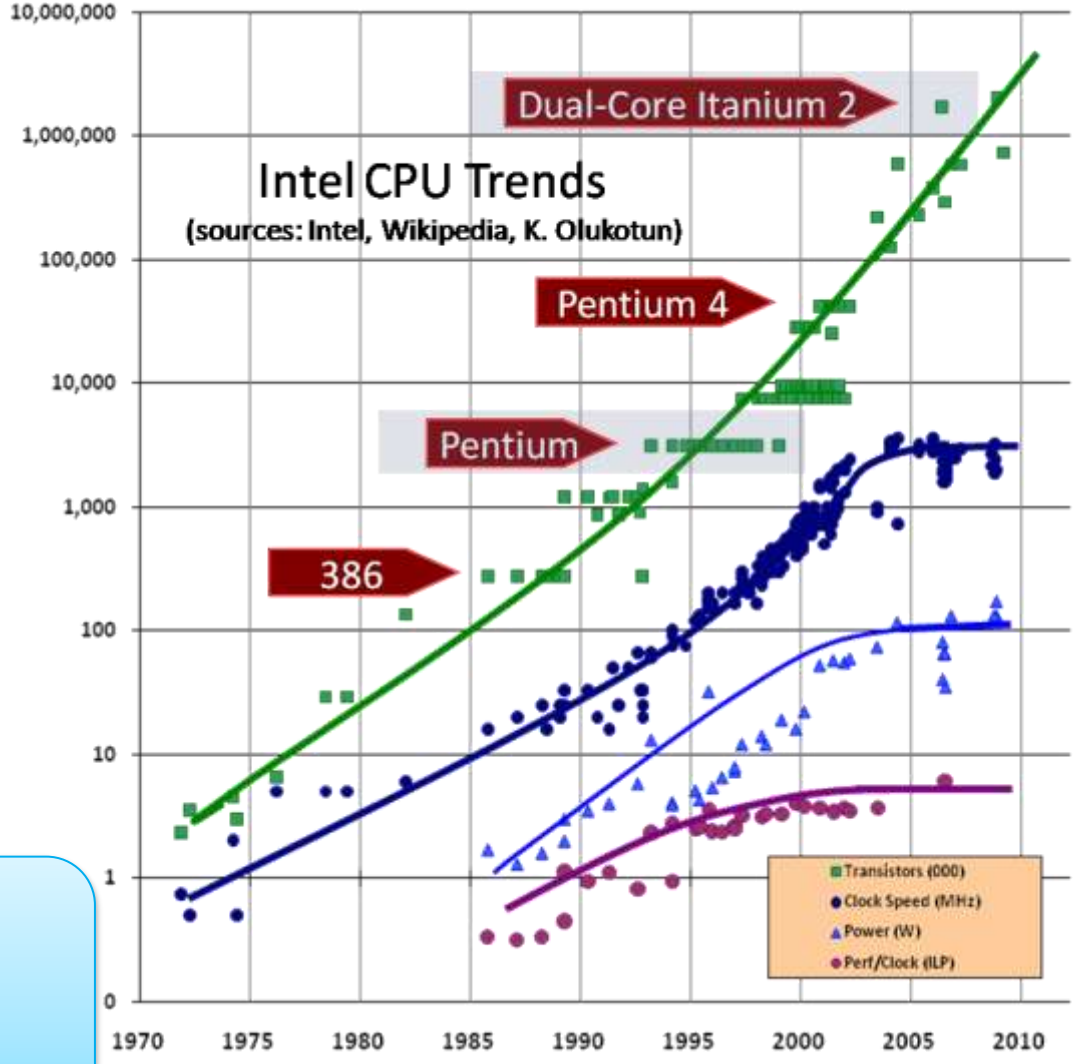
- Intel® Many Integrated Core Architecture
- Intel® Xeon Phi™ Coprocessor Overview
- Core and Vector Processing Unit
- Intel® Initial Many Core Instructions
- Interconnect and Cache Hierarchy
- Performance

Intel® Many Integrated Core Architecture

CPU Performance Trends

After ~2004 only the number of transistors continues to increase
We have hit limits in

- Power
- Instruction level parallelism
- Clock speed



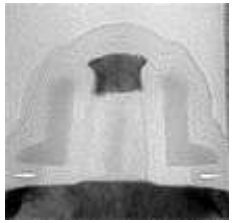
Herb Sutter, http://www.gotw.ca/publications/concurrency-ddj.htm

Single core scalar performance is now only growing slowly

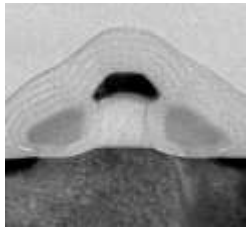
Moore's law is alive and well at Intel

New Intel technology generation every 2 years
Intel R&D technologies drive this pace well into the decade

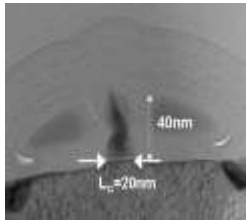
90nm
2003



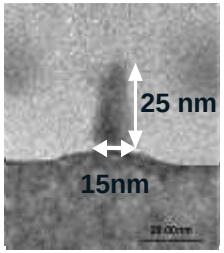
65nm
2005



45nm
2007

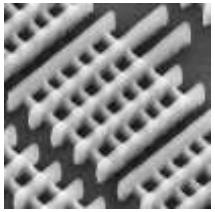


32nm
2009



Hi-K metal-gate

22nm
2011



3-D Tri-gate

14nm

We will have lots of transistors!

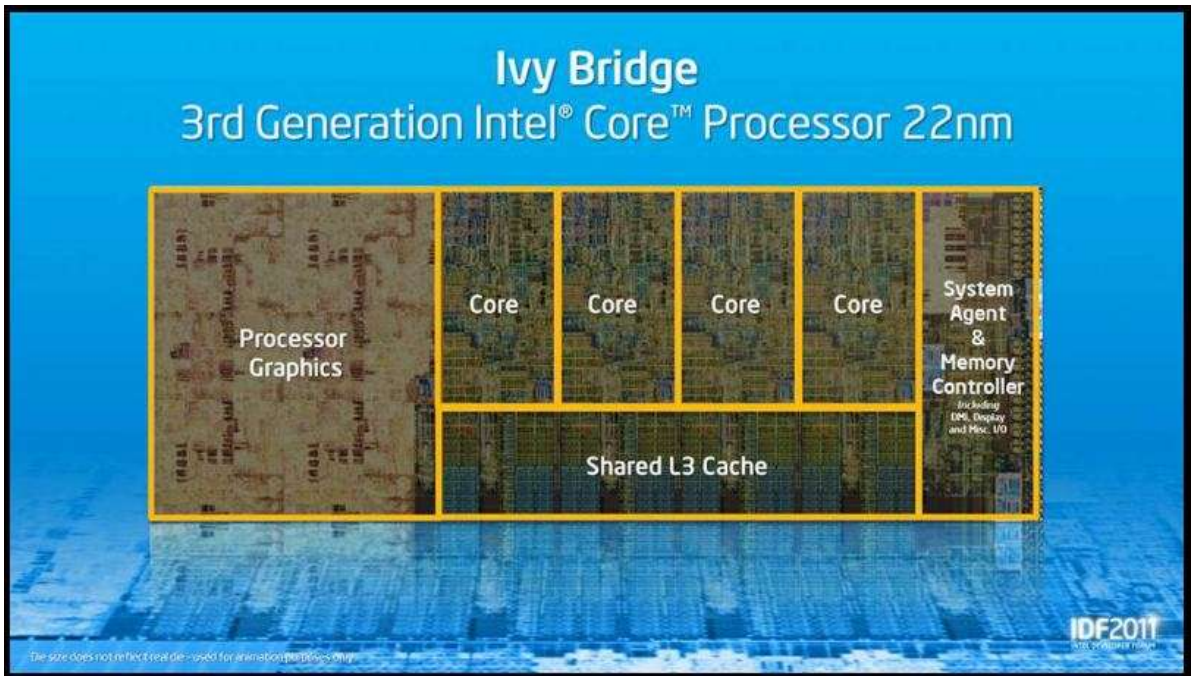
How do we use all the transistors?

Integrate other system components

- Graphics
- Memory i/f
- PCI i/f

Add cache

Replicate cores



This is a desktop part, but it has four cores each with two HW threads and 256 bit (8 single or 4 double) SIMD FP units.

Data and thread parallelism are mandatory to extract all available performance.

History of SIMD ISA extensions

Intel® Pentium® processor (1993)

MMX™ (1997)

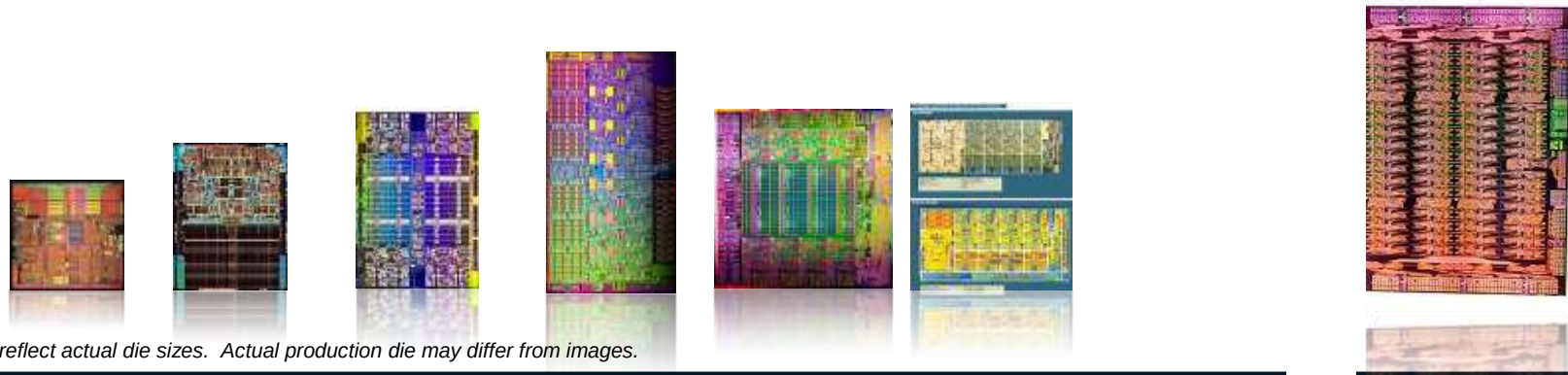
Intel® Streaming SIMD Extensions (Intel® SSE in 1999 to Intel® SSE4.2 in 2008)

Intel® Advanced Vector Extensions (Intel® AVX in 2011 and Intel® AVX2 in 2013)

Intel Many Integrated Core Architecture (Intel® MIC Architecture in 2013)

* Illustrated with the number of 32-bit data elements that are processed by one "packed" instruction.

Intel Architecture Multicore and Manycore
More cores. Wider vectors.

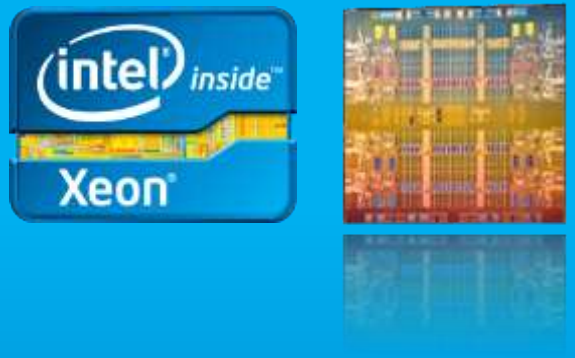


Images do not reflect actual die sizes. Actual production die may differ from images.

	Intel® Xeon® processor 64-bit	Intel Xeon processor 5100 series	Intel Xeon processor 5500 series	Intel Xeon processor 5600 series	Intel Xeon processor E5 Product Family	Intel Xeon processor code name Ivy Bridge	Intel Xeon processor code name Haswell	Intel® Xeon Phi™ Coprocessor
Core(s)	1	2	4	6	8			61
Threads	2	2	8	12	16			244

Intel® Xeon Phi™ Coprocessor extends established CPU architecture and programming concepts to highly parallel applications

Intel® Multicore Architecture

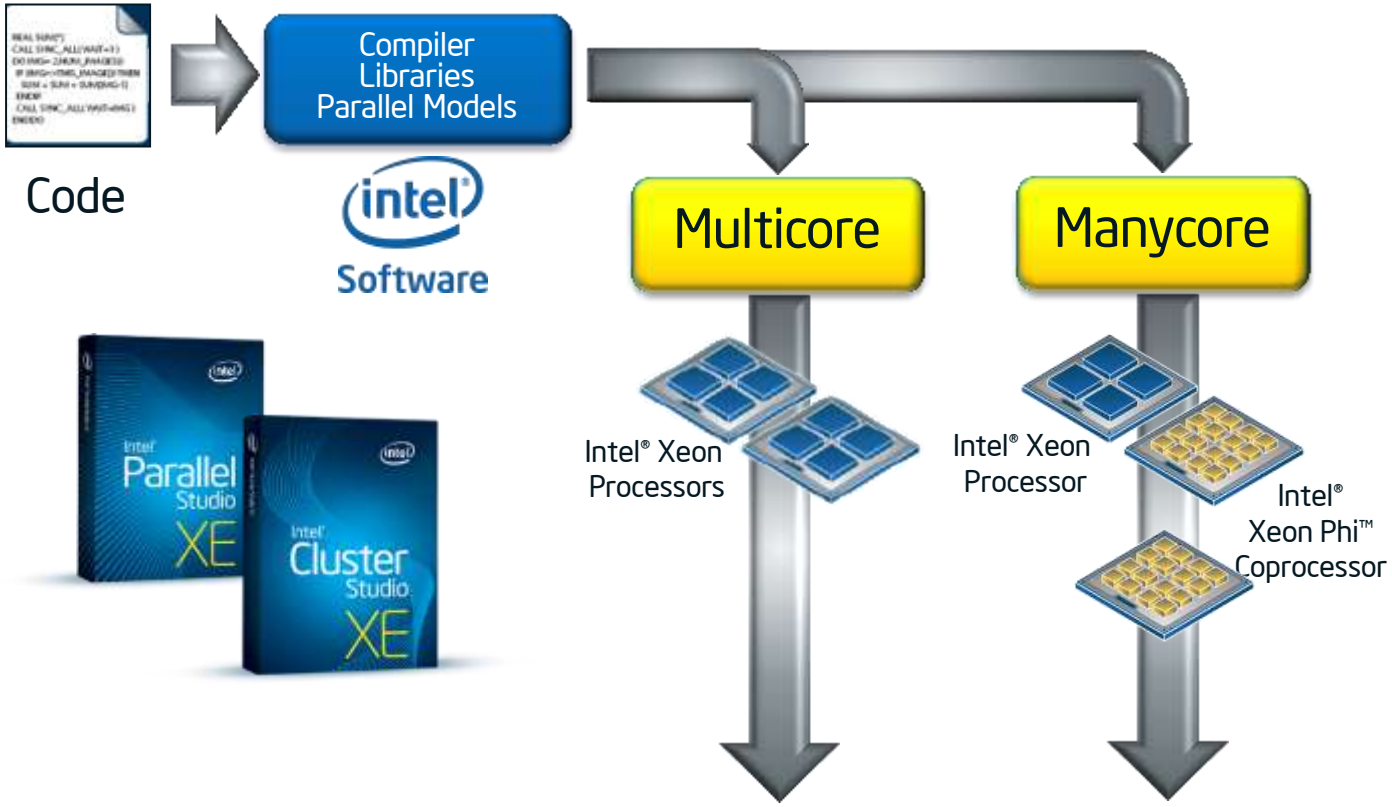


- Foundation of HPC Performance
- Suited for full scope of workloads
- Industry leading performance and performance/watt for serial & parallel workloads
- Focus on fast single core/thread performance with "moderate" number of cores

Intel® Many Integrated Core Architecture

- Performance and performance/watt optimized for highly parallelized compute workloads
- Common software tools with Xeon enabling efficient application readiness and performance tuning
- IA extension to Manycore
- Many cores/threads with wide SIMD

Consistent Tools & Programming Models



Tool Support for Latest Intel Processors and Coprocessors

	Intel® Ivy Bridge microarchitecture	Intel® Haswell microarchitecture	Intel® Xeon Phi™ coprocessor
Intel® C++ and Fortran Compiler	✔ AVX	✔ AVX2, FMA3	✔ IMCI
Intel® TBB library	✔	✔	✔
Intel® MKL library	✔ AVX	✔ AVX2, FMA3	✔
Intel® MPI library	✔	✔	✔
Intel® VTune™ Amplifier XE†	✔ Hardware Events	✔ Hardware Events	✔ Hardware Events
Intel® Inspector XE	✔ Memory & Thread Checks	✔ Memory & Thread	✔ Memory & Thread††

Intel® Xeon Phi™ Coprocessor Overview

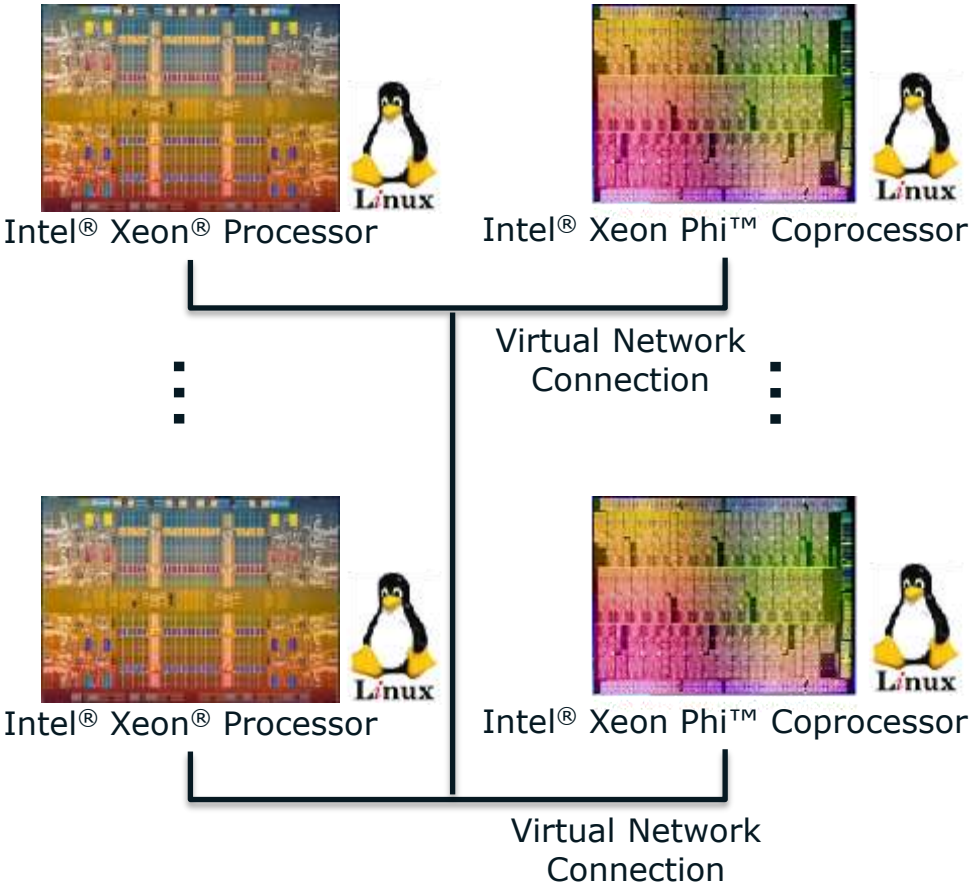
Intel® Xeon Phi™ Coprocessor

www.intel.com/xeonphi

Up to 61 Cores, 244 Threads
512-bit SIMD instructions
>1TFLOPS DP-F.P. peak
Up to 16GB GDDR5 Memory, 352 GB/s
PCIe* x16
Up to 300W TDP (card)

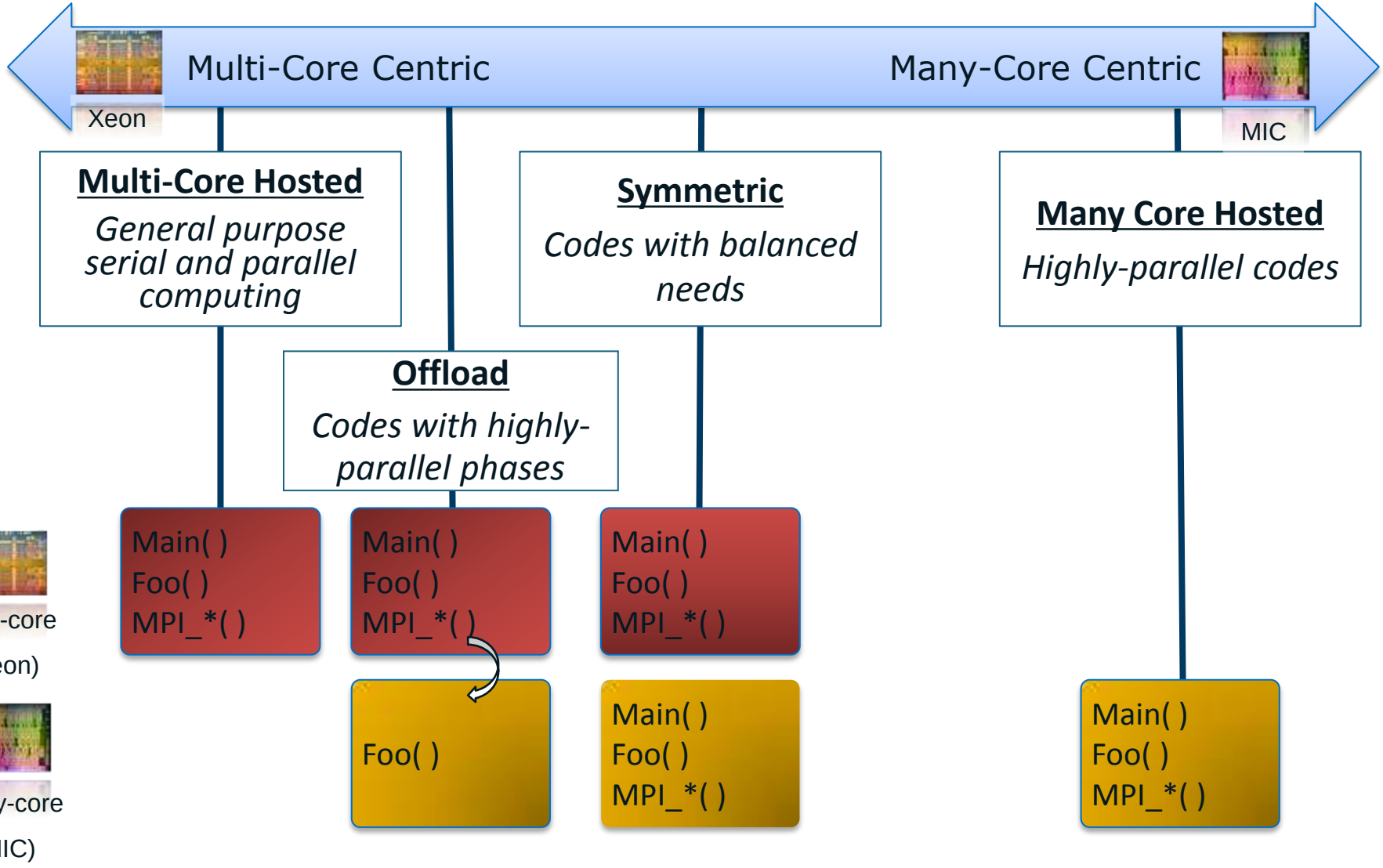
22nm with the world's first
3-D Tri-Gate transistors
Linux* operating system
IP addressable
Common x86/IA
Programming Models and SW-Tools

Intel® Xeon Phi™ Coprocessor Becomes a Network Node



Intel® Xeon Phi™ Architecture + Linux enables IP addressability

Spectrum of Programming Models and Mindsets

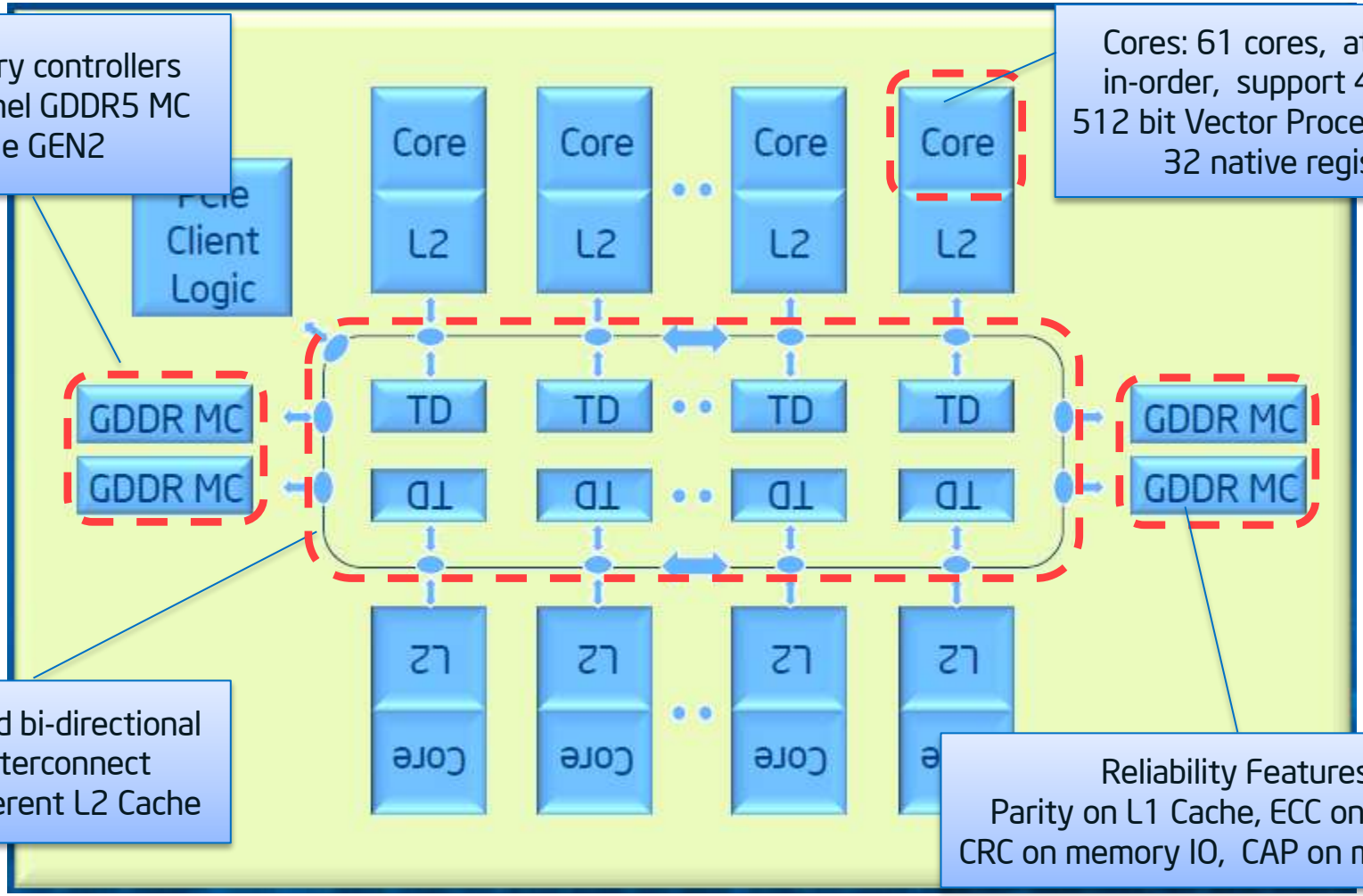


Range of models to meet application needs

Intel® MIC Programming Considerations

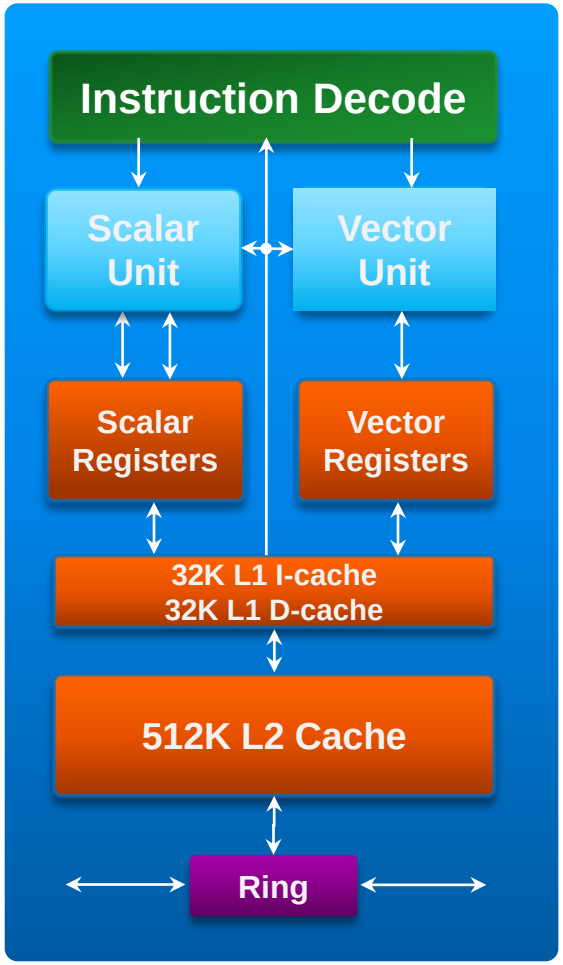
- Getting full performance from the Intel® MIC architecture requires both a high degree of **parallelism** and **vectorization**
  - Not all code can be written this way
  - Not all programs make sense on this architecture
- Intel® MIC is different from Xeon
  - It specializes in running highly parallel and vectorized code.
  - Not optimized for processing serial code
- Parallelism and vectorization optimizations are beneficial across both architectures

Intel® Xeon Phi™ Architecture Overview



Core and Vector Processing Unit

Core Architecture Overview



60+ in-order, low power IA cores in a ring interconnect

Two pipelines

- Scalar Unit based on Pentium® processors
- Dual issue with scalar instructions
- Pipelined one-per-clock scalar throughput

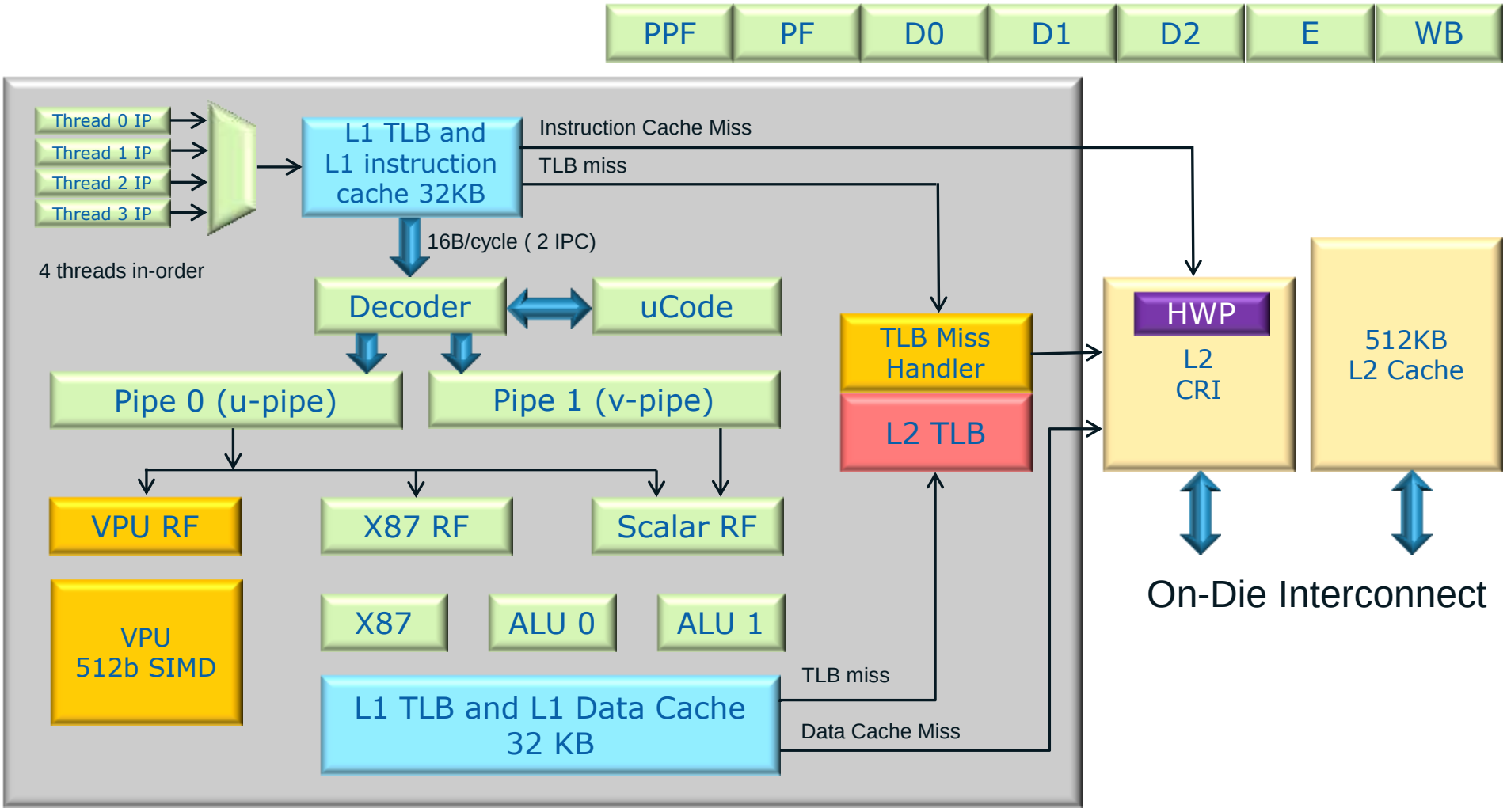
SIMD Vector Processing Engine

4 hardware threads per core

- 4 clock latency, hidden by round-robin scheduling of threads
- Cannot issue back to back inst in same thread: **Means minimum two threads per core to achieve full compute potential**

Coherent 512KB L2 Cache per core

Vector Processing Unit Extends the Scalar IA Core



Vector Instruction Performance

VPU contains 16 SP ALUs, 8 DP ALUs,

Most VPU instructions have a latency of 4 cycles and TPT 1 cycle

- Load/Store/Scatter have 7-cycle latency
- Convert/Shuffle have 6-cycle latency

VPU instructions are issued in u-pipe

Certain instructions can go to v-pipe also

- Vector Mask, Vector Store, Vector Packstore, Vector Prefetch, Scalar

Intel® Initial Many Core Instructions Overview

Intel® Initial Many Core Instructions Overview

Executed by the Xeon Phi's VPU

512-bit vectors

- 16x 4-bytes elements, or 8x 8-bytes elements
- 32 512-bit vector registers available
- Support for float32, float64 elements, and also int32, int64

Vector masks

Vector memory instructions

- Load/store
- Gather/scatter

Added hardware support for some operations

- Fused Multiply-Add
- Some transcendentals

Vector masks

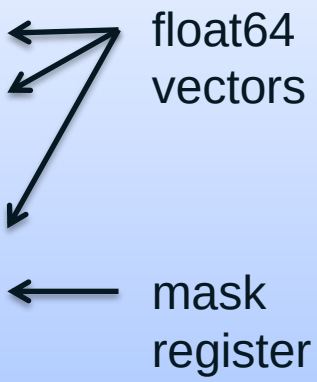
- Vector masks protect elements from updates during the execution of any operation (incl. arithmetic and memory ops)
- 8 special registers k{0, …, 7} to store vector masks
- Operations on masked elements are not executed

```
zmm0 = { 0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0 }
zmm1 = { 8.0, 7.0, 6.0, 5.0, 4.0, 3.0, 2.0, 1.0 }

zmm2 = { 9.0, 9.0, 9.0, 9.0, 9.0, 9.0, 9.0, 9.0 }
k3   = {  1 ,  1 ,  0 ,  1 ,  0 ,  1 ,  1 ,  1  }

vaddpd zmm2 {k3}, zmm0, zmm1

zmm2 : { 8.0, 8.0, 9.0, 8.0, 9.0, 8.0, 8.0, 8.0 }
```



Vectorization with vector masks

Vector masks greatly improve vectorization potential, e.g. allowing vectorization of:

- Loops with conditions

```
for (int i=0; i<n; i++) {
    if (residual[i] > epsilon)
        x[i] += correction[i];
}
```

- Short trip count loops and remainder loops:

```
for (int i=0; i<7; i++) {
    x[i] = expensive_simd_computation(y[i]);
}
```

Vector memory instructions

- IMCI provide usual vector load/store
  - Load/store 8 or 16 *contiguous* elements between memory and 512-bit registers

- Introduced support for gather/scatter
  - Allows vectorization of indirect accesses, by fetching sparse memory locations into a dense vector

```
for (int atom=0; atom<n; atom++) {
    accel[atom] = force[atom] /
                        mass[atom_type[atom]];
}
```

- New scalar prefetch instructions introduced

Hardware support for mathematical operations

Most new vector instructions are ternary, with 2 source and 1 destination vectors

Introduced fused multiply-add instructions

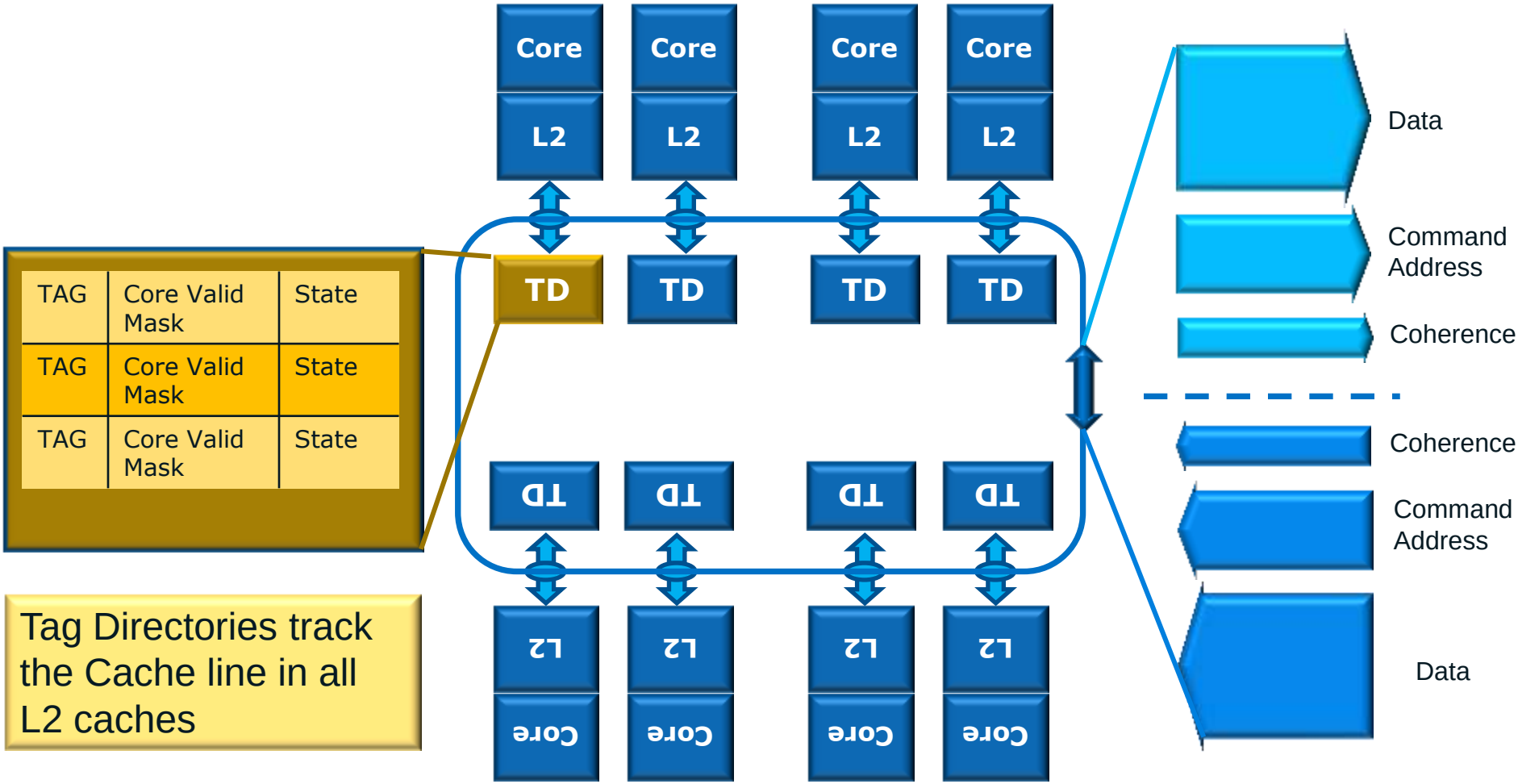
- 1-cycle throughput, up to 32 SP FP ops/cycle
- Standard IEEE 754-2008R (0.5 ulps, not 1 upls as two operations)

Extended Math Unit (EMU)

- Added instructions for SP elementary functions: rcp, rsqrt, log2, exp2
- Benefits pow(), sqrt(), div(), ln()

Interconnect and Cache Hierarchy

Ring Interconnect / Distributed Tag Directories



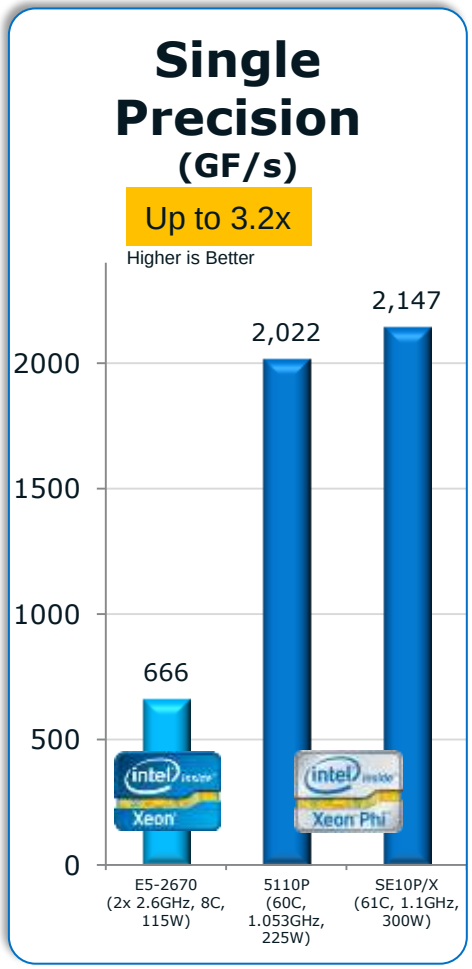
Cache Hierarchy

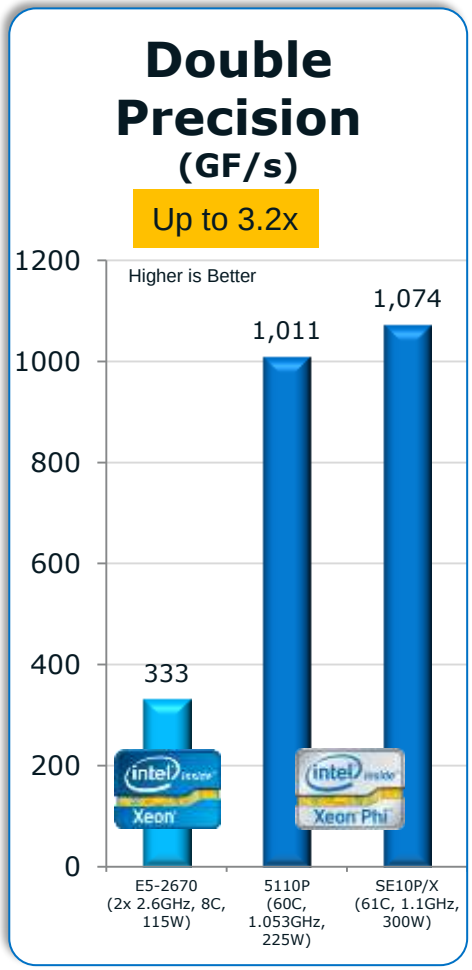
Parameter	L1	L2
Coherence	MESI	MESI
Size	32KB + 32 KB	512 KB
Associativity	8-way	8-way
Line Size	64 Bytes	64 Bytes
Banks	8	8
Access Time	2 cycle	23 cycle
Policy	Pseudo LRU	Pseudo LRU
Duty Cycle	1 per clock	1 per clock
Ports	Read or Write	Read or Write

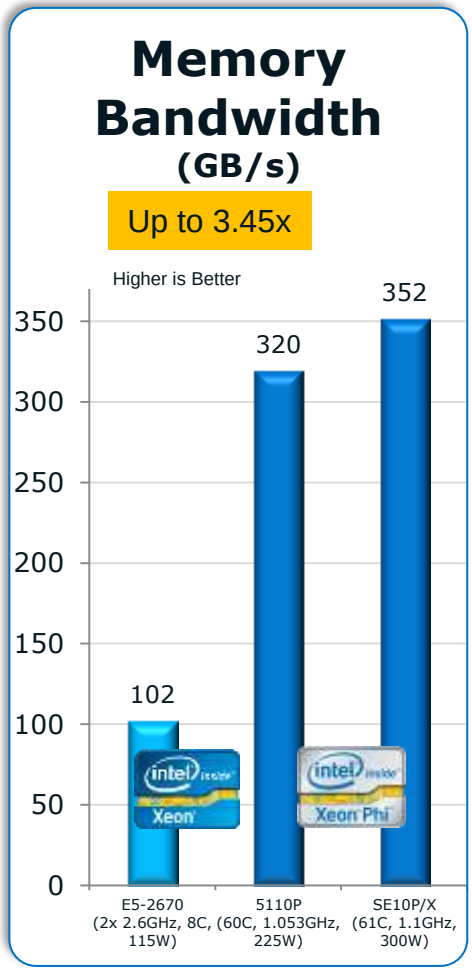
Benchmark Performance

Theoretical Maximum

(Intel® Xeon® processor E5-2670 vs. Intel® Xeon Phi™ coprocessor 5110P & SE10P/X)



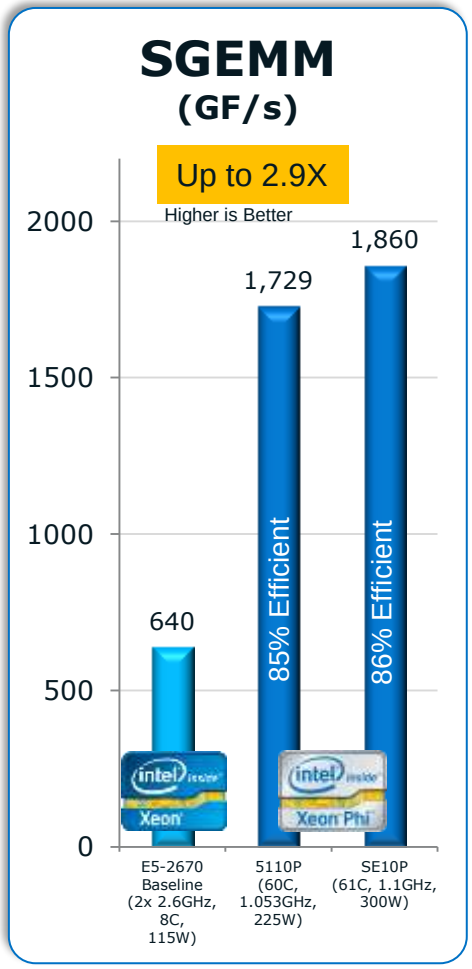


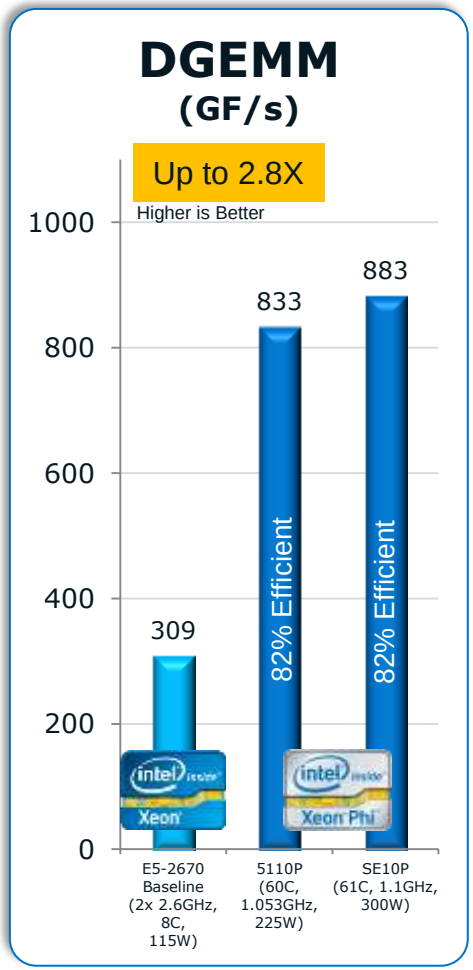


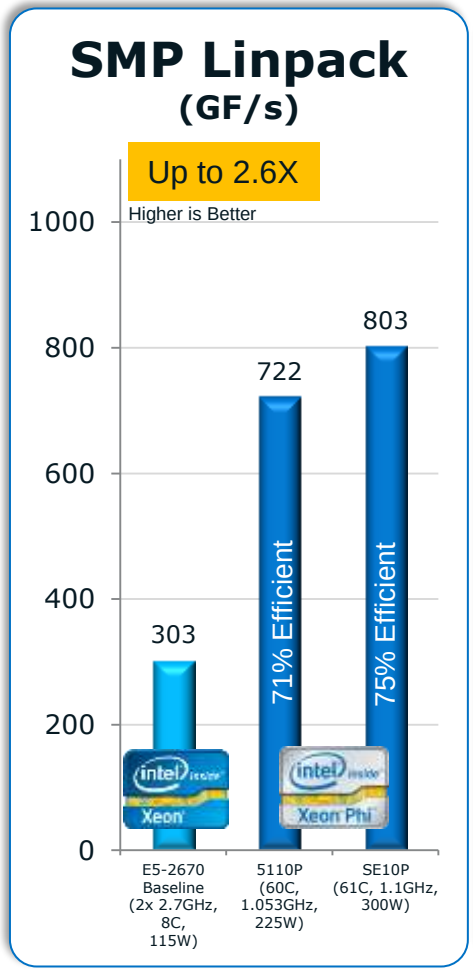
Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may 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.

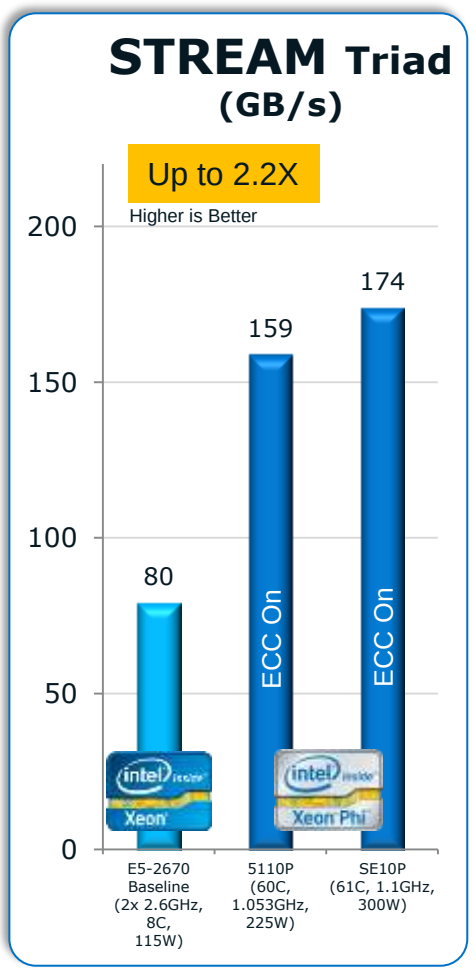
Source: Intel as of October 17, 2012 Configuration Details: Please reference slide speaker notes.
For more information go to http://www.intel.com/performance

Synthetic Benchmark Summary









Coprocessor results: Benchmark run 100% on coprocessor, no help from Intel® Xeon® processor host (aka native)

Next Intel® Xeon Phi™ Processor:
Knights Landing

Designed using Intel's cutting-edge
14nm process

Not bound by "offloading" bottlenecks
Standalone CPU
or PCIe Coprocessor

Leadership compute & memory bandwidth
Integrated
On-Package Memory

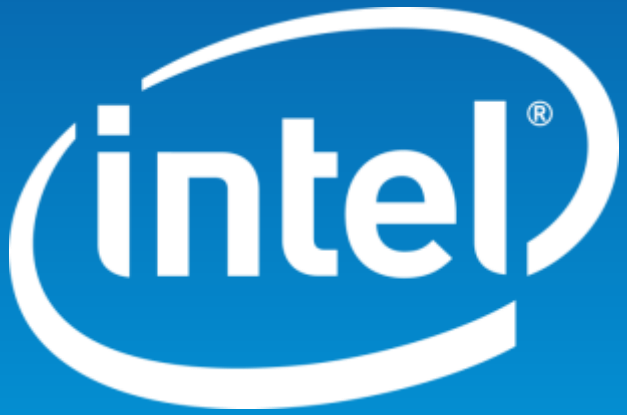
All products, computer systems, dates and figures specified are preliminary based on current expectations, and are subject to change without notice.

Useful Links

Check out http://software.intel.com/en-us/mic-developer including:

- Intel® Xeon Phi™ Coprocessor: Software Developers Guide http://www.intel.com/content/www/us/en/processors/xeon/xeon-phi-coprocessor-system-software-developers-guide.html

- Intel® Xeon Phi™ Coprocessor Instruction Set Architecture Reference Manual http://software.intel.com/sites/default/files/forum/278102/327364001en.pdf


intel