Advanced Architectures

Master Informatics Eng.

2017/18

A.J.Proença

Data Parallelism 2 (*SIMD++, Intel MIC, NVidia GPU ...*)

(most slides are borrowed)

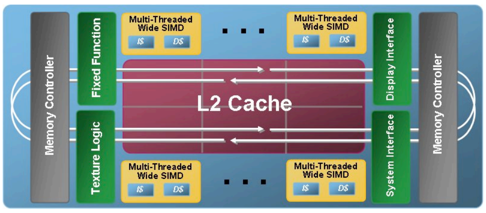
Beyond Vector/SIMD architectures

- Vector/SIMD-extended architectures are hybrid approaches
  - mix (super)**scalar + vector** op capabilities on a single device
  - **highly pipelined** approach to reduce memory access penalty
  - **tightly-closed access to shared memory**: lower latency
- Evolution of Vector/SIMD-extended architectures
  - **CPU cores with wider vector units**
    - x86 many-core: **Intel** MIC (Xeon KNL/KNM)
    - ...
  - **coprocessors (require a host scalar processor): accelerator devices**
    - on disjoint physical memories (e.g., **Xeon KNC** with PCI-Expr**, PEZY-SC**)
    - ...
  - **heterogeneous processors (multicore with GPU-cores,SoC)**
    - ...

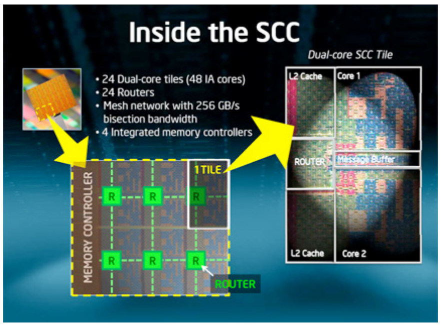
Intel MIC: Many Integrated Core

Intel evolution, from:

- Larrabee (80-core GPU) & SCC

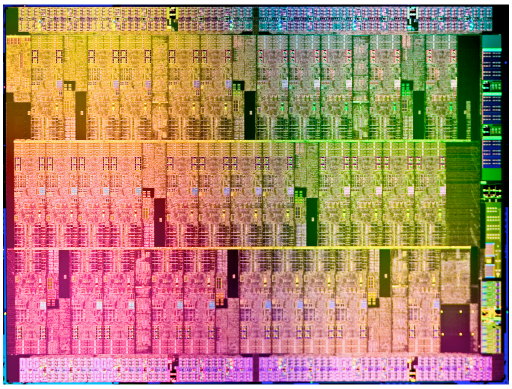


Single-chip **C**loud **C**omputer, 24x dual-core tiles

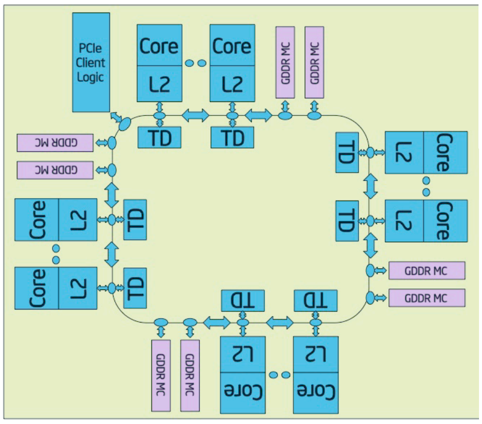


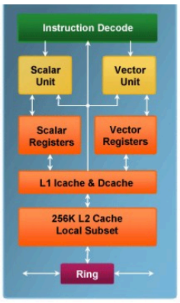
to MIC:

- Knights Ferry (pre-production, Stampede)
- Knights Corner →
  Xeon Phi co-processor up to 61 Pentium cores
- Knights Landing & Knights Mill
  Xeon Phi full processor up to 36x dual-core Atom tiles

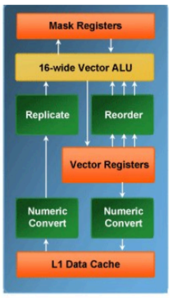


Intel Knights Corner architecture

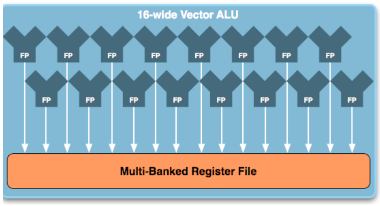


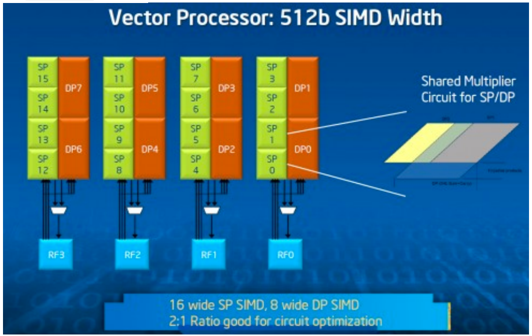


Processor Core

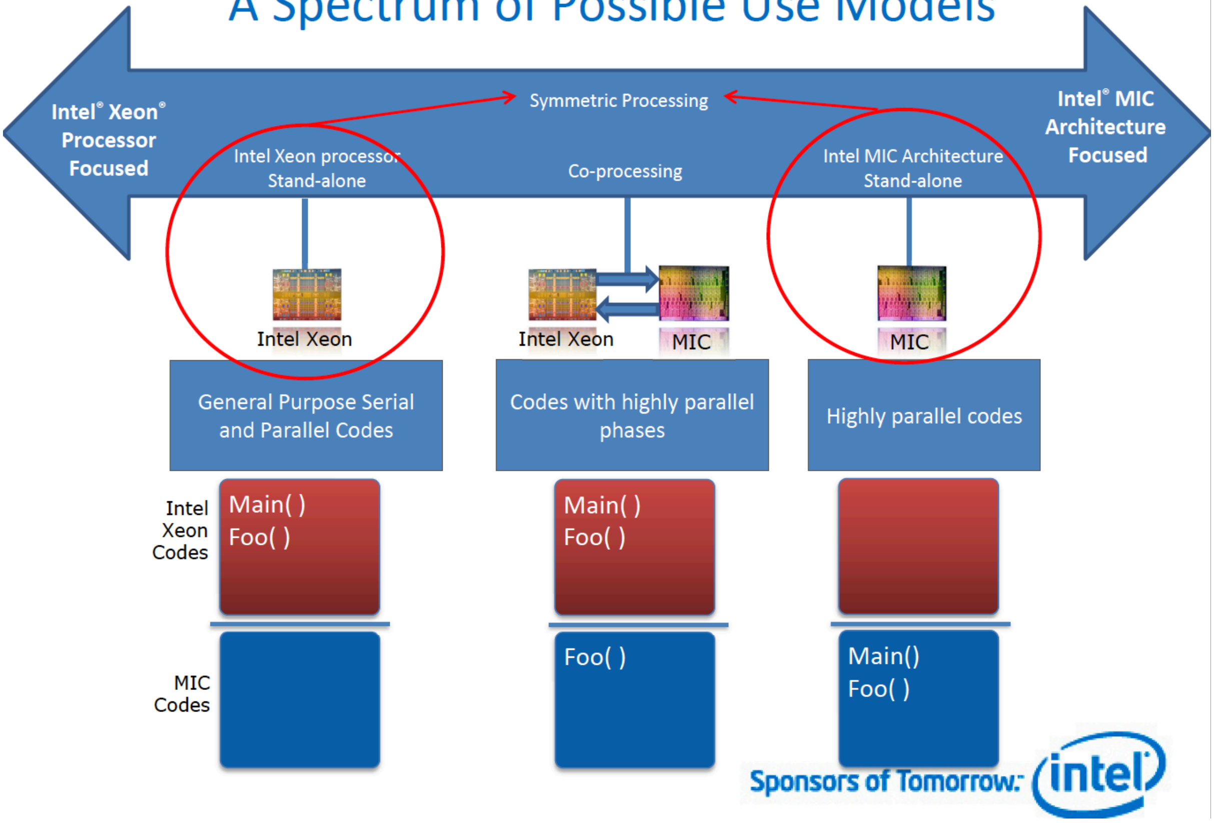


Vector Unit




Vector Processor: 512b SIMD Width
SP
DP7
DP6
DP5
DP4
DP3
DP2
DP1
DP0
RF3
RF2
RF1
RF0
Shared Multiplier Circuit for SP/DP
16 wide SP SIMD, 8 wide DP SIMD
2:1 Ratio good for circuit optimization

A Spectrum of Possible Use Models



Intel® Xeon® Processor Focused

Symmetric Processing

Intel® MIC Architecture Focused

Intel Xeon processor Stand-alone

Co-processing

Intel MIC Architecture Stand-alone

General Purpose Serial and Parallel Codes

Codes with highly parallel phases

Highly parallel codes

Intel Xeon Codes

MIC Codes

Main( )
Foo( )

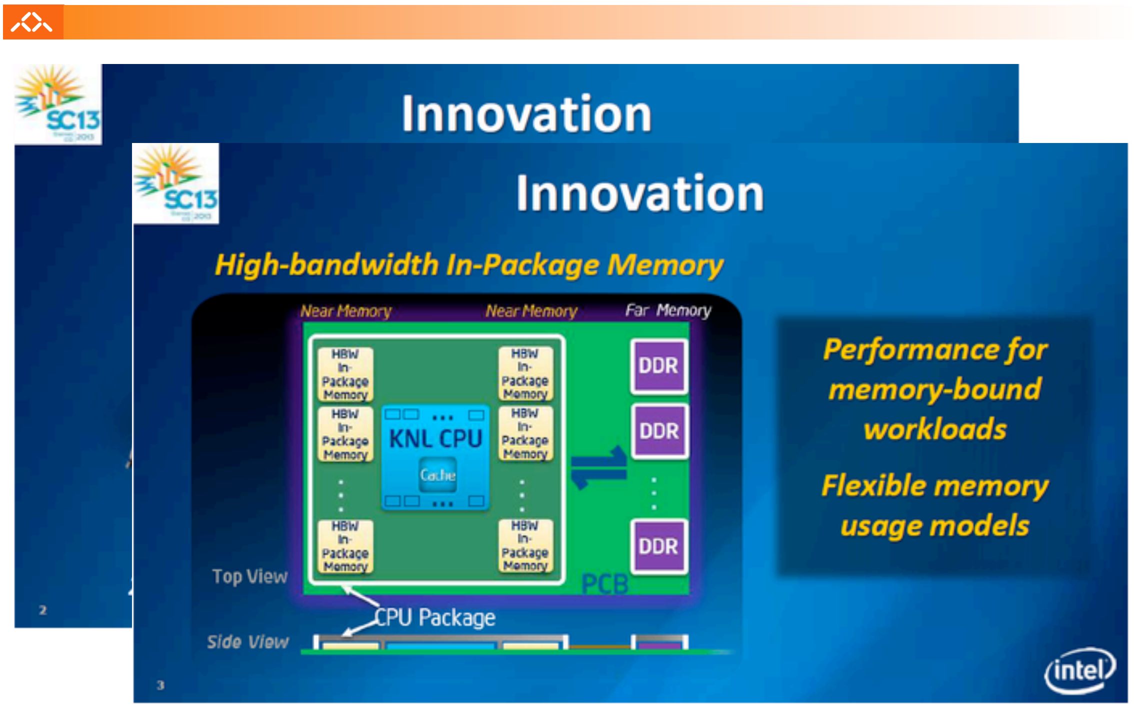
Main( )
Foo( )

Foo( )

Main()
Foo( )

Sponsors of Tomorrow.

The new Knights Landing architecture



Innovation

Innovation

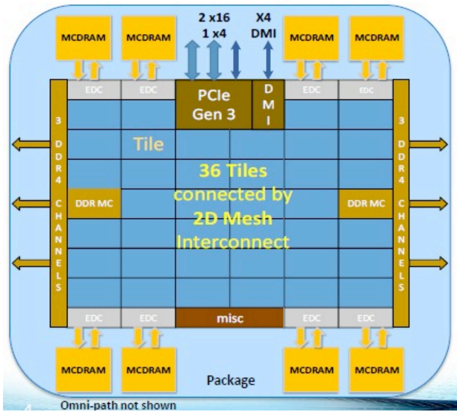
High-bandwidth In-Package Memory

Performance for memory-bound workloads

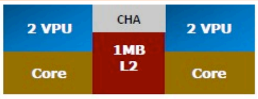
Flexible memory usage models

Intel Knights Landing in 2016: Xeon Phi com 72 cores

Knights Landing Overview



TILE

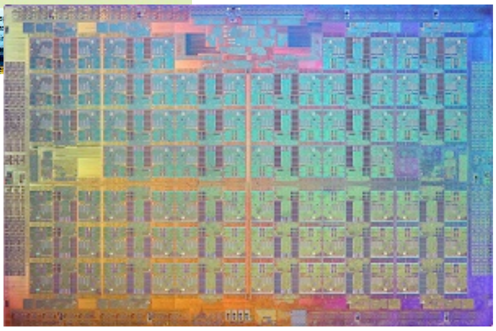


Chip: 36 **Tiles** interconnected by **2D Mesh**
Tile: 2 Cores + 2 VPU/core + 1 MB L2

Memory: MCDRAM: 16 GB on-package; High BW
DDR4: 6 channels @ 2400 up to 384GB
IO: 36 lanes PCIe Gen3. 4 lanes of DMI for chipset
Node: 1-Socket only
Fabric: Omni-Path on-package (not shown)

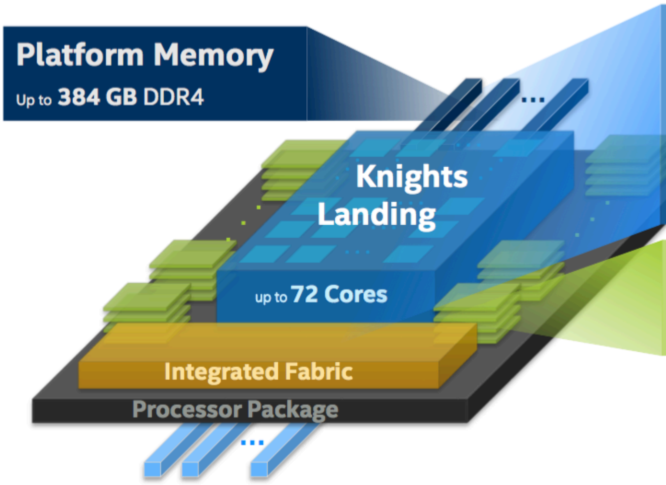
Vector Peak Perf: 3+TF DP and 6+TF SP Flops
Scalar Perf: ~3x over Knights Corner
Streams Triad (GB/s): MCDRAM : 400+; DDR: 90+

Source Intel: All products, computer systems, dates and figures s
are subject to change without notice. KNL data are preliminary bas
without notice. 1Binary Compatible with Intel Xeon processors usi
numbers are based on STREAM-like memory access pattern whe
estimated based on internal Intel analysis and [illegible]



More details in a later set of slides...

INTEL® XEON PHI™ X200 PROCESSOR OVERVIEW

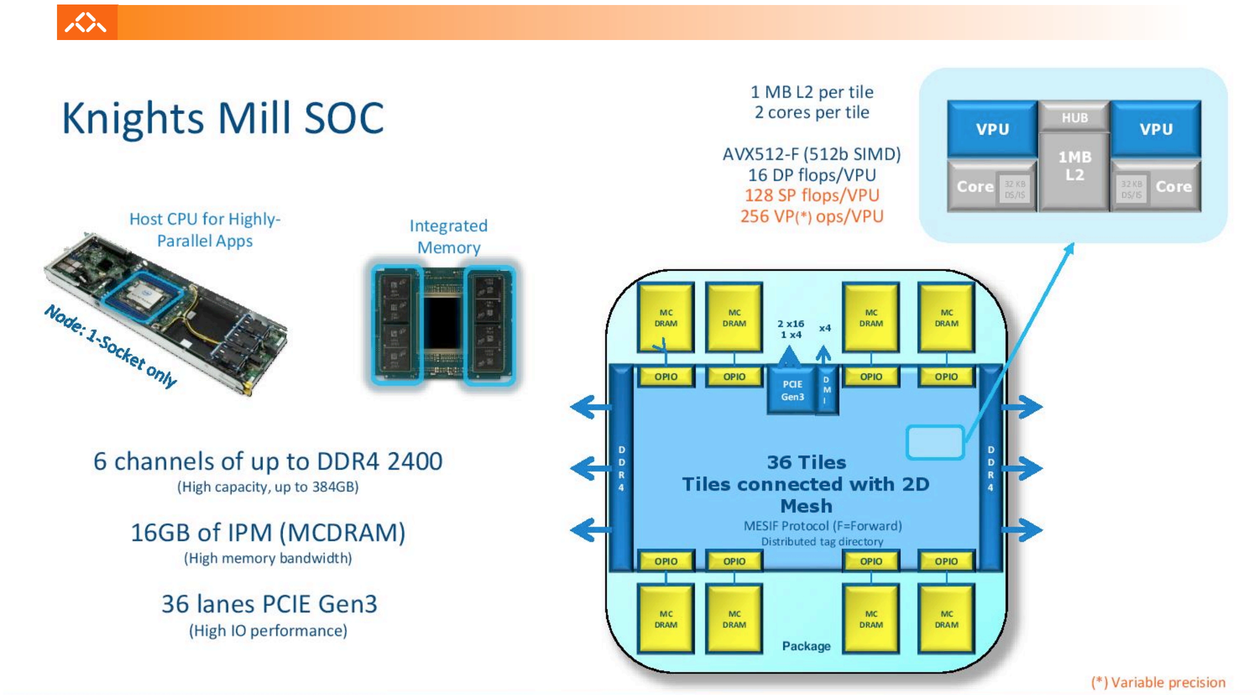


Platform Memory
Up to **384 GB DDR4**

Compute
- Intel® Xeon® Processor Binary-Compatible
- **3+ TFLOPS, 3X ST** (single-thread) perf. vs KNC
- **2D Mesh** Architecture
- **Out-of-Order** Cores

On-Package Memory (MCDRAM)
- Up to **16 GB** *at launch*
- Over **5x STREAM** *vs. DDR4 at launch*

Intel Knights Mill expected in 2018: similar to KNL, but...



Knights Mill SOC

Host CPU for Highly-Parallel Apps

Node: 1-Socket only

Integrated Memory

1 MB L2 per tile
2 cores per tile

AVX512-F (512b SIMD)
16 DP flops/VPU
128 SP flops/VPU
256 VP(*) ops/VPU

6 channels of up to DDR4 2400
(High capacity, up to 384GB)

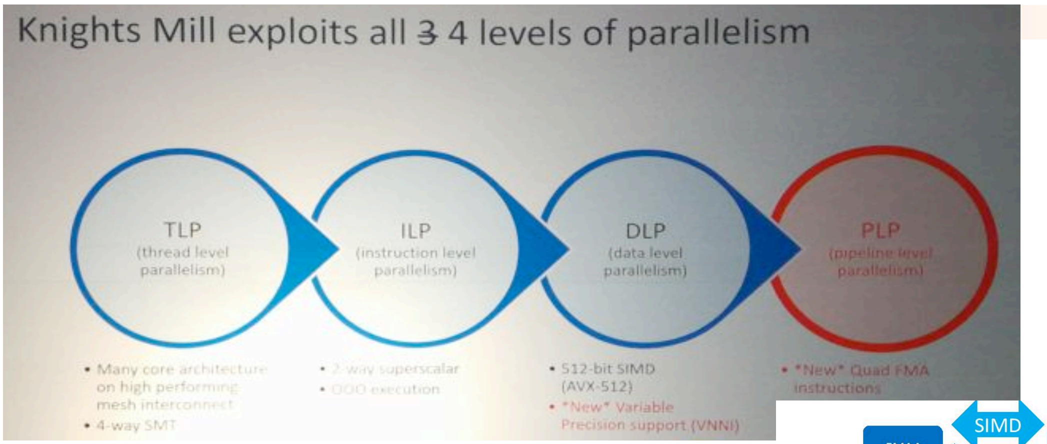
16GB of IPM (MCDRAM)
(High memory bandwidth)

36 lanes PCIE Gen3
(High IO performance)

36 Tiles
Tiles connected with 2D Mesh
MESIF Protocol (F=Forward)
Distributed tag directory

(*) Variable precision

Intel Knights Mill expected in 2018: similar to KNL, but...



Knights Mill exploits all ~~3~~ 4 levels of parallelism

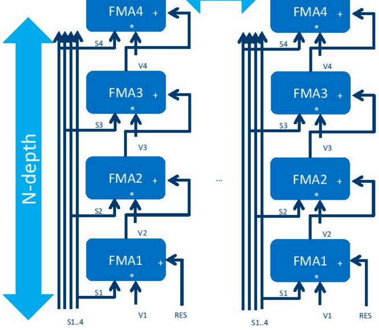
TLP (thread level parallelism)
- Many core architecture on high performing mesh interconnect
- 4-way SMT

ILP (instruction level parallelism)
- 2-way superscalar
- OOO execution

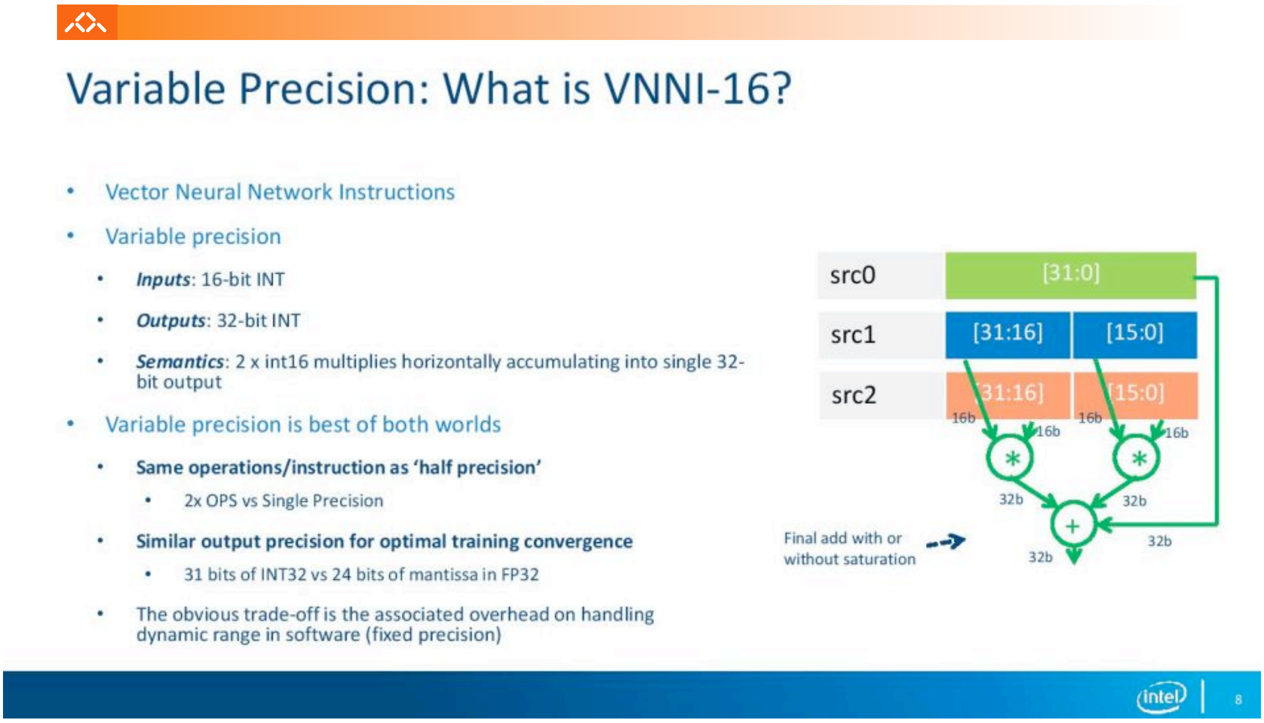
DLP (data level parallelism)
- 512-bit SIMD (AVX-512)
- *New* Variable Precision support (VNNI)

PLP (pipeline level parallelism)
- *New* Quad FMA instructions

Quad FMA FP32 Instructions



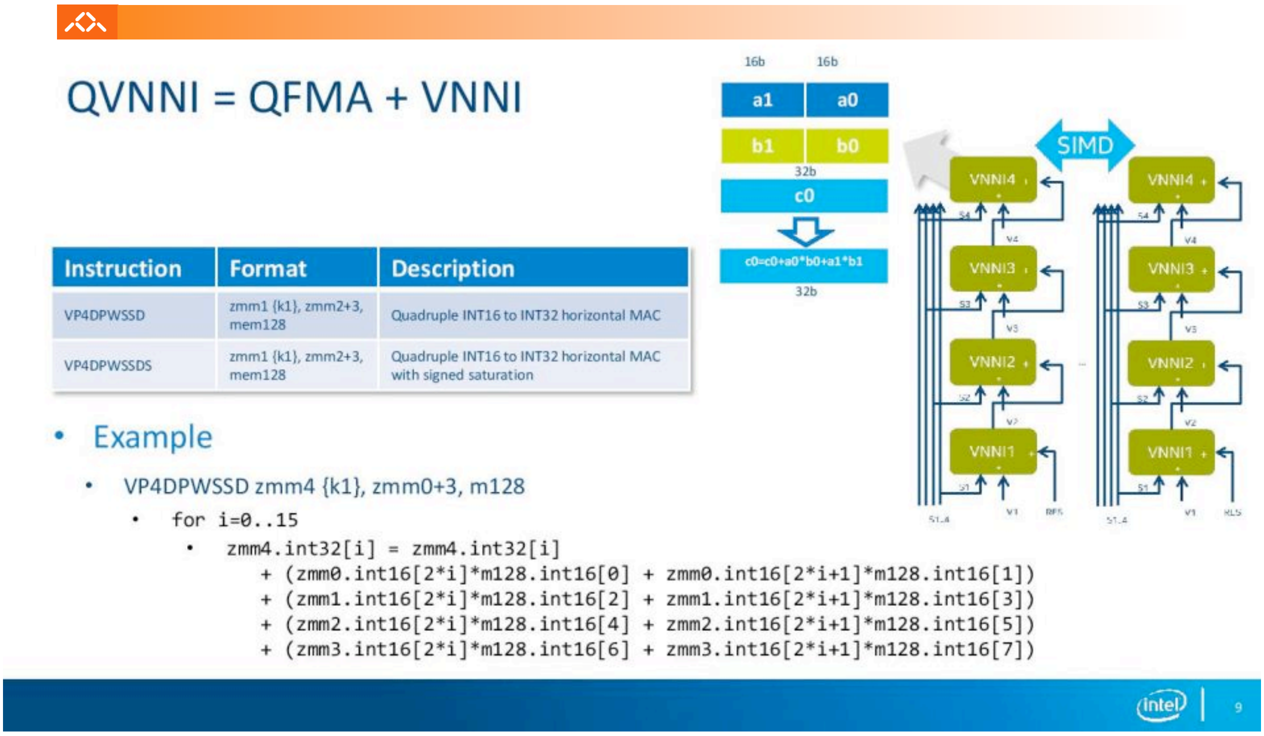
Intel Knights Mill expected in 2018: similar to KNL, but...



Variable Precision: What is VNNI-16?

- Vector Neural Network Instructions
- Variable precision
  - ***Inputs***: 16-bit INT
  - ***Outputs***: 32-bit INT
  - ***Semantics***: 2 x int16 multiplies horizontally accumulating into single 32-bit output
- Variable precision is best of both worlds
  - **Same operations/instruction as 'half precision'**
    - 2x OPS vs Single Precision
  - **Similar output precision for optimal training convergence**
    - 31 bits of INT32 vs 24 bits of mantissa in FP32
  - The obvious trade-off is the associated overhead on handling dynamic range in software (fixed precision)

Intel Knights Mill expected in 2018: similar to KNL, but...



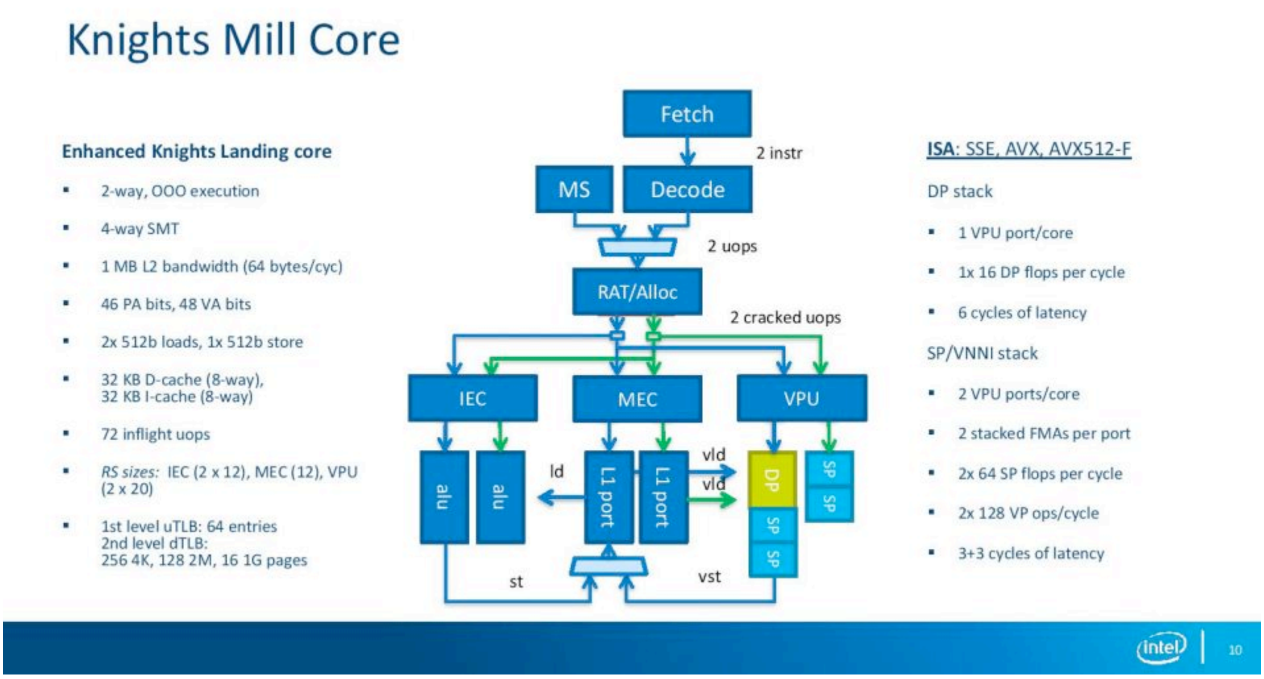
QVNNI = QFMA + VNNI

Instruction	Format	Description
VP4DPWSSD	zmm1 {k1}, zmm2+3, mem128	Quadruple INT16 to INT32 horizontal MAC
VP4DPWSSDS	zmm1 {k1}, zmm2+3, mem128	Quadruple INT16 to INT32 horizontal MAC with signed saturation

- Example
  - VP4DPWSSD zmm4 {k1}, zmm0+3, m128

```
• for i=0..15
    • zmm4.int32[i] = zmm4.int32[i]
        + (zmm0.int16[2*i]*m128.int16[0] + zmm0.int16[2*i+1]*m128.int16[1])
        + (zmm1.int16[2*i]*m128.int16[2] + zmm1.int16[2*i+1]*m128.int16[3])
        + (zmm2.int16[2*i]*m128.int16[4] + zmm2.int16[2*i+1]*m128.int16[5])
        + (zmm3.int16[2*i]*m128.int16[6] + zmm3.int16[2*i+1]*m128.int16[7])
```

Intel Knights Mill expected in 2018: similar to KNL, but...



Knights Mill Core

Enhanced Knights Landing core

- 2-way, OOO execution
- 4-way SMT
- 1 MB L2 bandwidth (64 bytes/cyc)
- 46 PA bits, 48 VA bits
- 2x 512b loads, 1x 512b store
- 32 KB D-cache (8-way), 32 KB I-cache (8-way)
- 72 inflight uops
- *RS sizes:* IEC (2 x 12), MEC (12), VPU (2 x 20)
- 1st level uTLB: 64 entries 2nd level dTLB: 256 4K, 128 2M, 16 1G pages

ISA: SSE, AVX, AVX512-F

DP stack

- 1 VPU port/core
- 1x 16 DP flops per cycle
- 6 cycles of latency

SP/VNNI stack

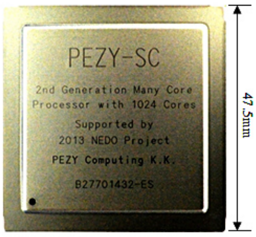
- 2 VPU ports/core
- 2 stacked FMAs per port
- 2x 64 SP flops per cycle
- 2x 128 VP ops/cycle
- 3+3 cycles of latency

PEZY-SC: Peta_Exa_Zetta_Yotta-SuperComputer: a 1024-core many-core processor chip

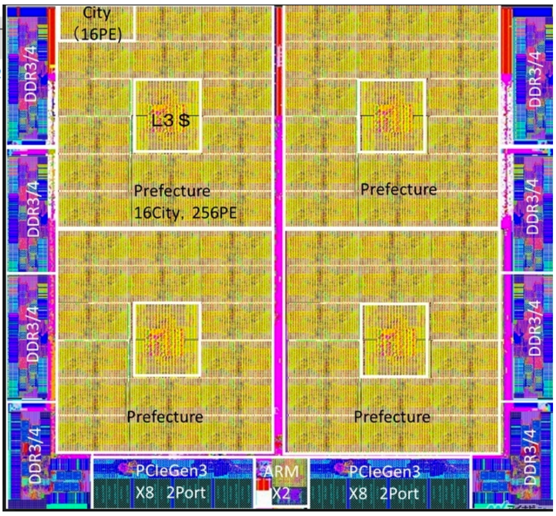
Green500 Rank	MFLOPS/W	Site*	Computer*	Total Power (kW)
1 94	6,673.84	Advanced Center for Computing and Communication, RIKEN	Shoubu - ZettaScaler-1.6, Xeon E5-2618Lv3 8C 2.3GHz, Infiniband FDR, PEZY-SCnp	149.99
2 486	6,195.22	Computational Astrophysics Laboratory, RIKEN	Satsuki - ZettaScaler-1.6, Xeon E5-2618Lv3 8C 2.3GHz, Infiniband FDR, PEZY-SCnp	46.89
3 1	6,051.30	National Supercomputing Center in Wuxi	Sunway TaihuLight - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway	15,371.00
4 440	5,272.09	GSI Helmholtz Center	ASUS ESC4000 FDR/G2S, Intel Xeon E5-2690v2 10C 3GHz, Infiniband FDR, AMD FirePro S9150	57.15
5 446	4,778.46	Institute of Modern Physics (IMP), Chinese Academy of Sciences	Sugon Cluster W780I, Xeon E5-2640v3 8C 2.6GH QDR, NVIDIA Tesla K80	
6 122	4,112.11	Stanford Research Computing Center	XStream - Cray CS-Storm, Intel Xeon E5-2680v2 Infiniband FDR, Nvidia K80	

Top500 Rank

Green500 list
June'2016



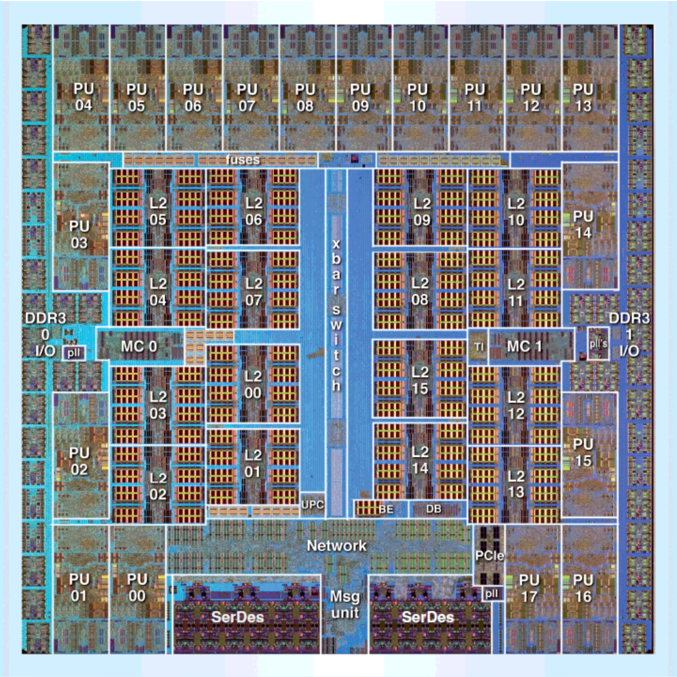
PEZY-SC
2nd Generation Many Core Processor with 1024 Cores
Supported by 2013 NEDO Project
PEZY Computing K.K.
B27701432-ES
47.5mm



Beyond Vector/SIMD architectures

- Vector/SIMD-extended architectures are hybrid approaches
  - mix (super)**scalar + vector** op capabilities on a single device
  - **highly pipelined** approach to reduce memory access penalty
  - **tightly-closed access to shared memory**: lower latency
- Evolution of Vector/SIMD-extended architectures
  - **CPU cores with wider vector units**
    - x86 many-core: **Intel** MIC / Xeon KNL
    - IBM Power cores with SIMD extensions: BlueGene/Q Compute
    - other many-core: **ShenWay** 260
  - **coprocessors (require a host scalar processor): accelerator devices**
    - on disjoint physical memories (e.g., **Xeon KNC** with PCI-Expr**, PEZY-SC**)
    - ...
  - **heterogeneous processors (multicore with GPU-cores,SoC)**
    - ...

IBM Power BlueGene/Q Compute *(chip)*

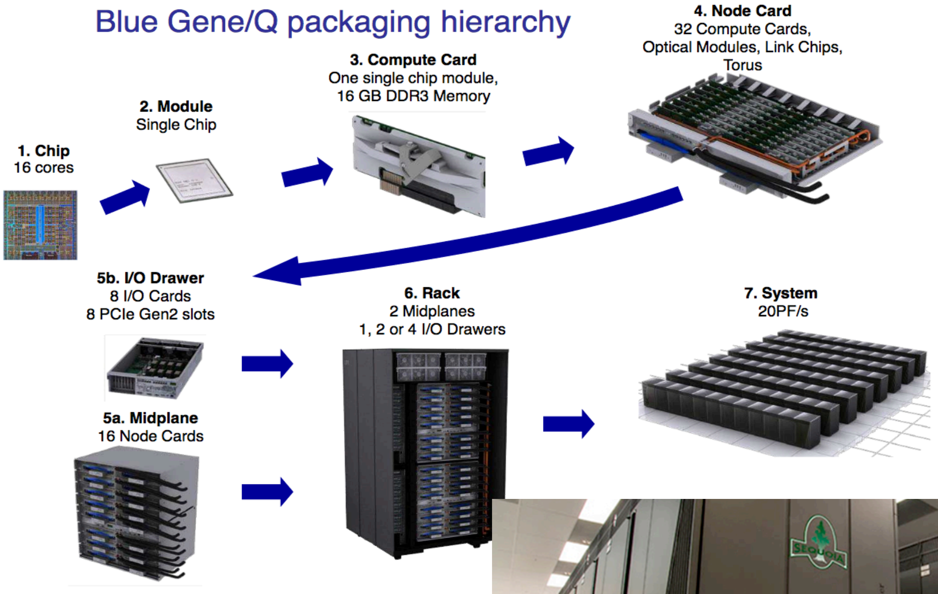


Features:

- launched in 2010/11
(TOP500: #1 in Jun12, #4 in Jun16)
- 18-cores
  - 16 compute,
1 OS support, 1 redundant
  - 64 bits PowerISA
  - 1.6 GHz
  - L1 I/D cache => 16 kB / 16 kB
  - **each core: quad-FPU**
(4-wide double precision SIMD)
  - each core: 4-way multi-threaded
- shared L2 cache: 32 MB
- dual memory controller
- IBM ended development of BlueGene project in 2015...

IBM Power BlueGene/Q Compute (Sequoia system)

Blue Gene/Q packaging hierarchy



1. Chip
16 cores

2. Module
Single Chip

3. Compute Card
One single chip module,
16 GB DDR3 Memory

4. Node Card
32 Compute Cards,
Optical Modules, Link Chips,
Torus

5b. I/O Drawer
8 I/O Cards
8 PCIe Gen2 slots

5a. Midplane
16 Node Cards

6. Rack
2 Midplanes
1, 2 or 4 I/O Drawers

7. System
20PF/s

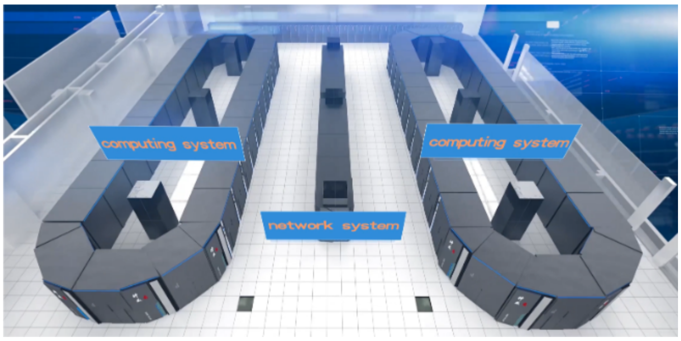
Ref: SC2010

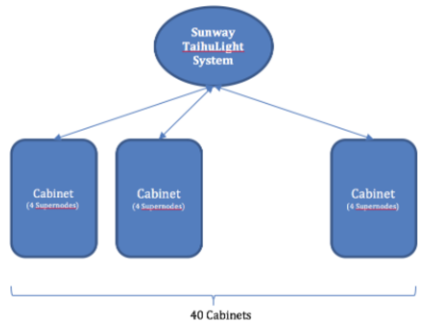
TOP 500 The List.

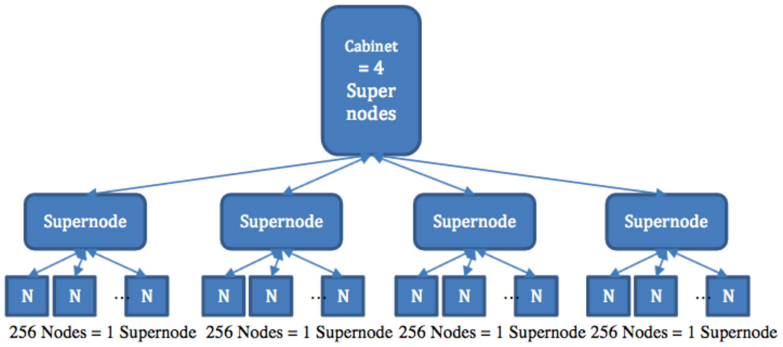
Jun'12: #1
Nov'12: #2
Jun'13: #3
Nov'13: #3
Jun'14: #3
Nov'14: #3
Jun'15: #3
Nov'15: #3
Jun'16: #4
Nov'16: #4
Jun'17: #5

#1 in June'16 TOP500: Sunway TaihuLight

Overview of the Sunway TaihuLight System

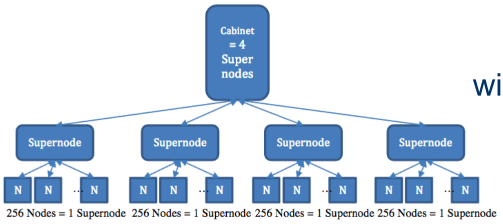




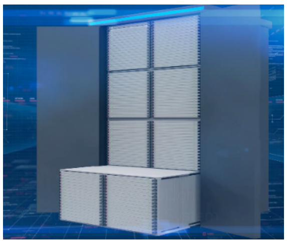


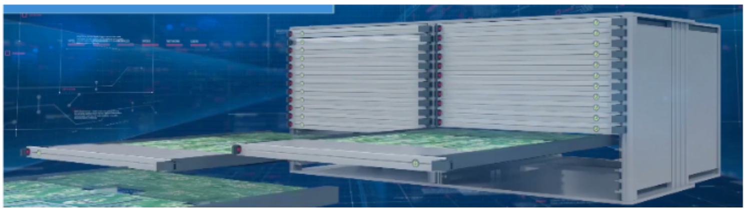
http://www.netlib.org/utk/people/JackDongarra/PAPERS/sunway-report-2016.pdf

#1 since June'16 TOP500: Sunway TaihuLight

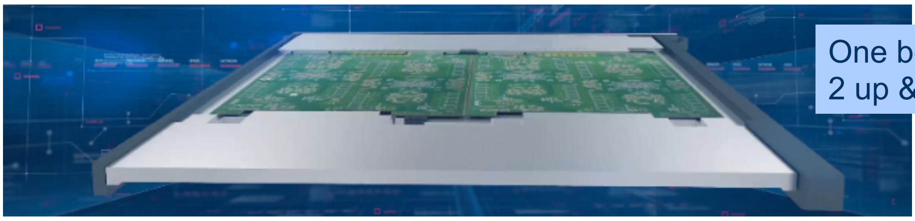


One cabinet
with 4 Supernodes





One Supernode
with 32 boards

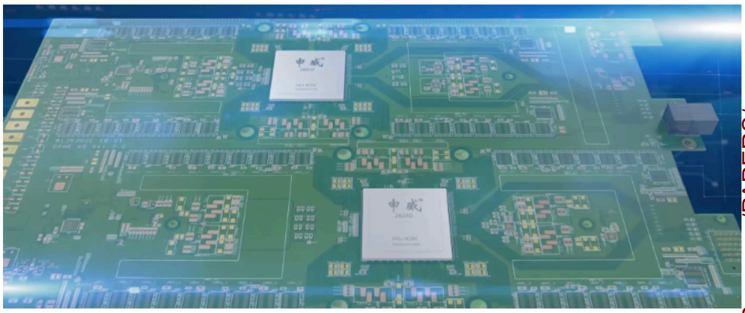


One board with 4 cards,
2 up & 2 down

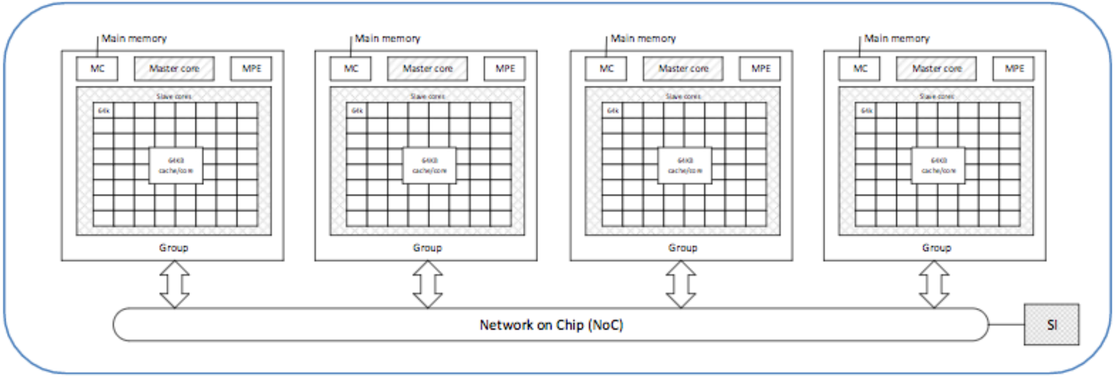
http://www.netlib.org/utk/people/JackDongarra/PAPERS/sunway-report-2016.pdf

#1 in June'16 TOP500: Sunway TaihuLight

One card with two nodes
(two SW26010 chips)



SW26010: the 4x64-core 64-bit RISC processor (with 256-bit vector instructions & only cache L1)



http://www.netlib.org/utk/people/JackDongarra/PAPERS/sunway-report-2016.pdf

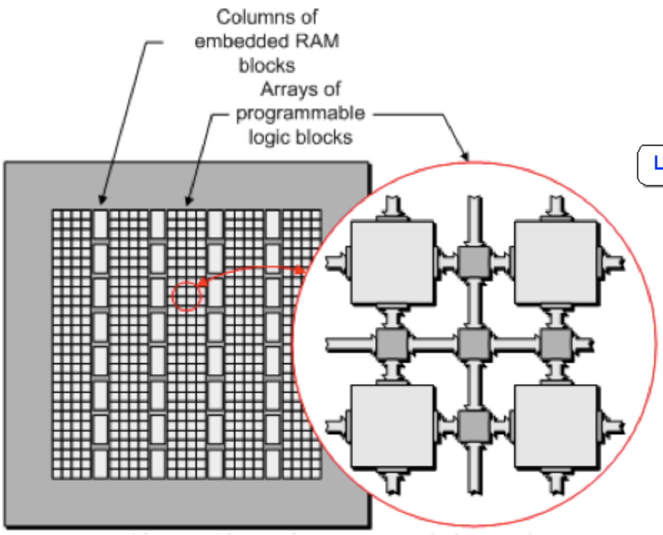
Beyond Vector/SIMD architectures

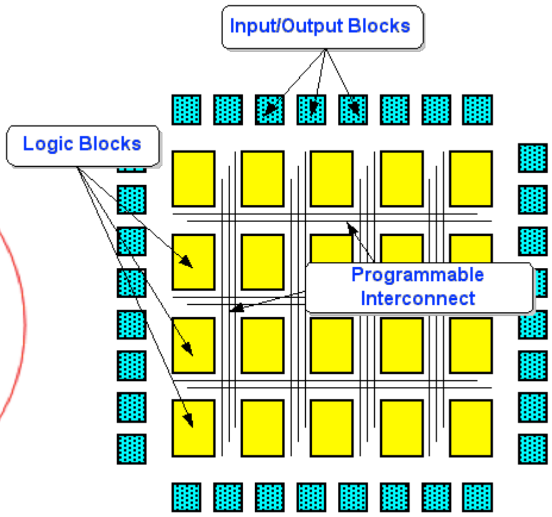
- Vector/SIMD-extended architectures are hybrid approaches
  - mix (super)**scalar + vector** op capabilities on a single device
  - **highly pipelined** approach to reduce memory access penalty
  - **tightly-closed access to shared memory**: lower latency
- Evolution of Vector/SIMD-extended architectures
  - **CPU cores with wider vector units**
    - x86 many-core: **Intel** MIC / Xeon KNL
    - IBM Power cores with SIMD extensions: BlueGene/Q Compute
    - other many-core: **ShenWay** 260
  - **coprocessors (require a host scalar processor): accelerator devices**
    - on disjoint physical memories (e.g., **Xeon KNC** with PCI-Expr, **PEZY-SC**)
    - ISA-free architectures, code compiled to silica: **FPGA**
    - ...
  - **heterogeneous processors (multicore with GPU-cores, SoC)**
    - ...

What is an FPGA

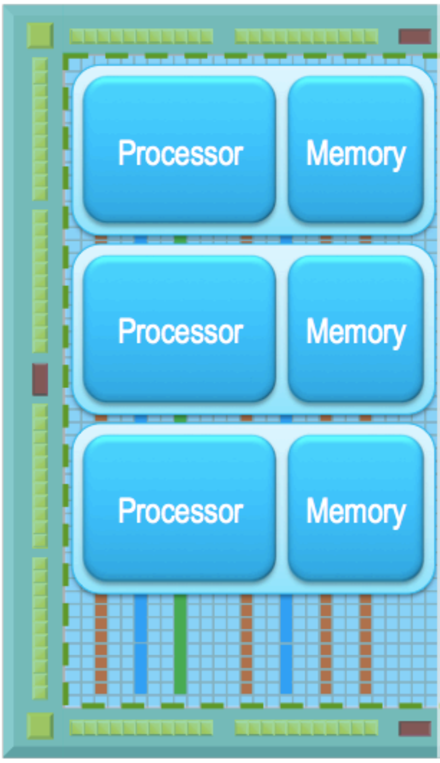
Field-Programmable Gate Arrays (FPGA)

A fabric with 1000s of simple configurable logic cells with LUTs, on-chip SRAM, configurable routing and I/O cells


Columns of embedded RAM blocks
Arrays of programmable logic blocks

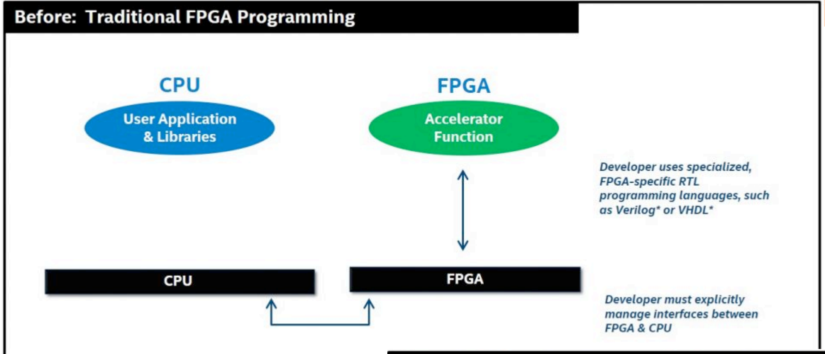


FPGA as a multiple configurable ISA



- Many coarse-grained processors
  - Different Implementation Options
    - Small soft scalar processor
    - or Larger vector processor
    - or Customized hardware pipeline
  - Each with local memory
- Each processor can exploit the fine grained parallelism of the FPGA to more efficiently implement it's "program"
- Possibly heterogeneous
  - Optimized for different tasks
- Customizable to suit the needs of a particular application

FPGA as a computing accelerator



Before: Traditional FPGA Programming

CPU
User Application & Libraries

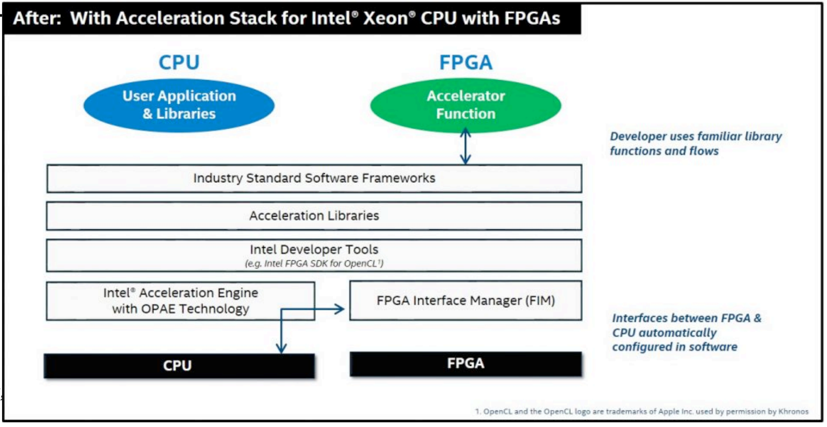
FPGA
Accelerator Function

Developer uses specialized, FPGA-specific RTL programming languages, such as Verilog or VHDL**

CPU

FPGA

Developer must explicitly manage interfaces between FPGA & CPU



After: With Acceleration Stack for Intel® Xeon® CPU with FPGAs

CPU
User Application & Libraries

FPGA
Accelerator Function

Developer uses familiar library functions and flows

Industry Standard Software Frameworks

Acceleration Libraries

Intel Developer Tools
(e.g. Intel FPGA SDK for OpenCL¹)

Intel® Acceleration Engine with OPAE Technology

FPGA Interface Manager (FIM)

CPU

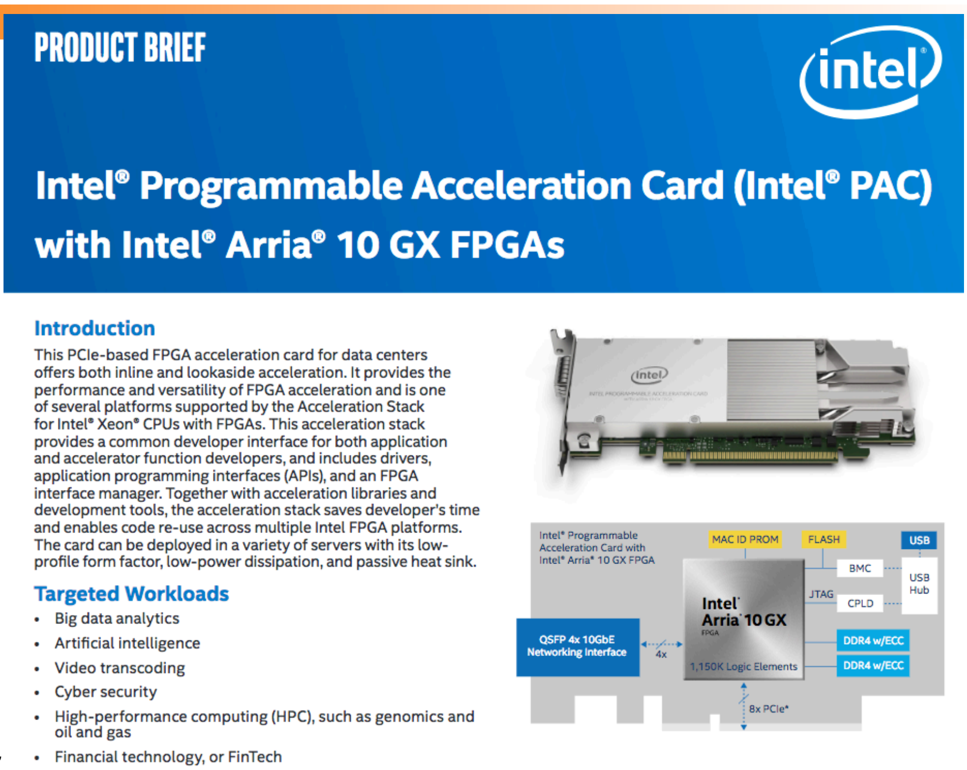
FPGA

Interfaces between FPGA & CPU automatically configured in software

1. OpenCL and the OpenCL logo are trademarks of Apple Inc. used by permission by Khronos

https://builders.intel.com/blog/fpga-in-the-data-center-programming-for-all/

The Intel Programmable Acceleration Card



PRODUCT BRIEF

Intel® Programmable Acceleration Card (Intel® PAC) with Intel® Arria® 10 GX FPGAs

Introduction

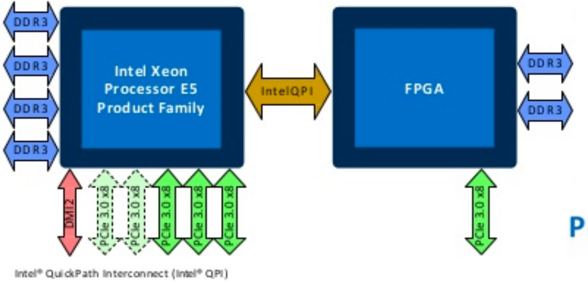
This PCIe-based FPGA acceleration card for data centers offers both inline and lookaside acceleration. It provides the performance and versatility of FPGA acceleration and is one of several platforms supported by the Acceleration Stack for Intel® Xeon® CPUs with FPGAs. This acceleration stack provides a common developer interface for both application and accelerator function developers, and includes drivers, application programming interfaces (APIs), and an FPGA interface manager. Together with acceleration libraries and development tools, the acceleration stack saves developer's time and enables code re-use across multiple Intel FPGA platforms. The card can be deployed in a variety of servers with its low-profile form factor, low-power dissipation, and passive heat sink.

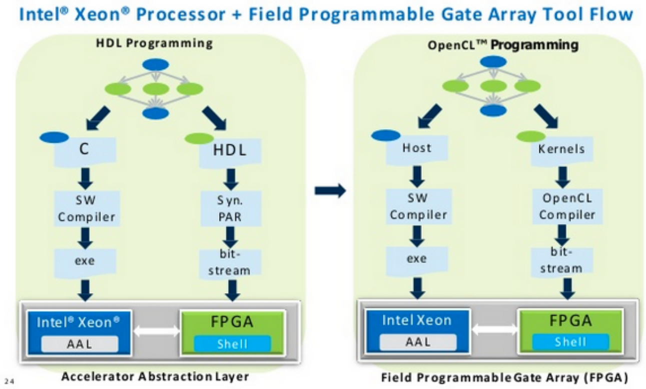
Targeted Workloads

- Big data analytics
- Artificial intelligence
- Video transcoding
- Cyber security
- High-performance computing (HPC), such as genomics and oil and gas
- Financial technology, or FinTech

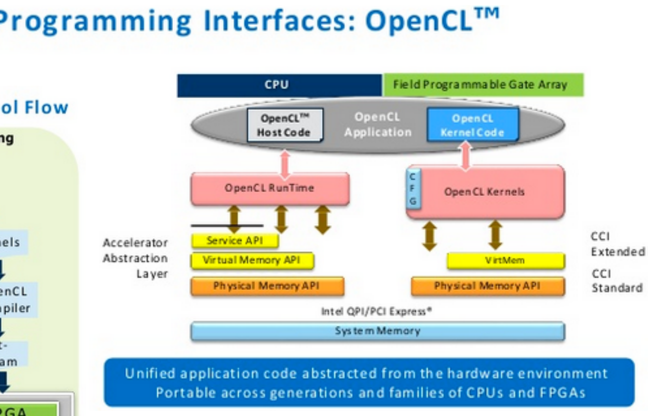
https://builders.intel.com/blog/fpga-in-the-data-center-programming-for-all/

Faster integration of programmable acceleration cards at Intel





Intel® Xeon® Processor + Field Programmable Gate Array Tool Flow



Programming Interfaces: OpenCL™

Unified application code abstracted from the hardware environment
Portable across generations and families of CPUs and FPGAs

https://www.slideshare.net/insideHPC/using-xeon-fpga-for-accelerating-hpc-workloads

Beyond Vector/SIMD architectures

- Vector/SIMD-extended architectures are hybrid approaches
  - mix (super)**scalar + vector** op capabilities on a single device
  - **highly pipelined** approach to reduce memory access penalty
  - **tightly-closed access to shared memory**: lower latency
- Evolution of Vector/SIMD-extended architectures
  - **CPU cores with wider vector units**
    - x86 many-core: **Intel** MIC / Xeon KNL
    - IBM Power cores with SIMD extensions: BlueGene/Q Compute
    - other many-core: **ShenWay** 260
  - **coprocessors (require a host scalar processor): accelerator devices**
    - on disjoint physical memories (e.g., **Xeon KNC** with PCI-Expr, **PEZY-SC**)
    - ISA-free architectures, code compiled to silica: **FPGA**
    - focus on SIMT/SIMD to hide memory latency: **GPU**-type approach
  - **heterogeneous processors (multicore with GPU-cores, SoC)**
    - ...

Graphical Processing Units

- Question to GPU architects:
  - *Given the hardware invested to do graphics well, how can we supplement it to improve the performance of a wider range of applications?*
- Key ideas:
  - Heterogeneous execution model
    - CPU is the *host*, GPU is the *device*
  - Develop a C-like programming language for GPU
  - Unify all forms of GPU parallelism as *CUDA_threads*
  - Programming model follows SIMT: "*Single Instruction Multiple Thread* "

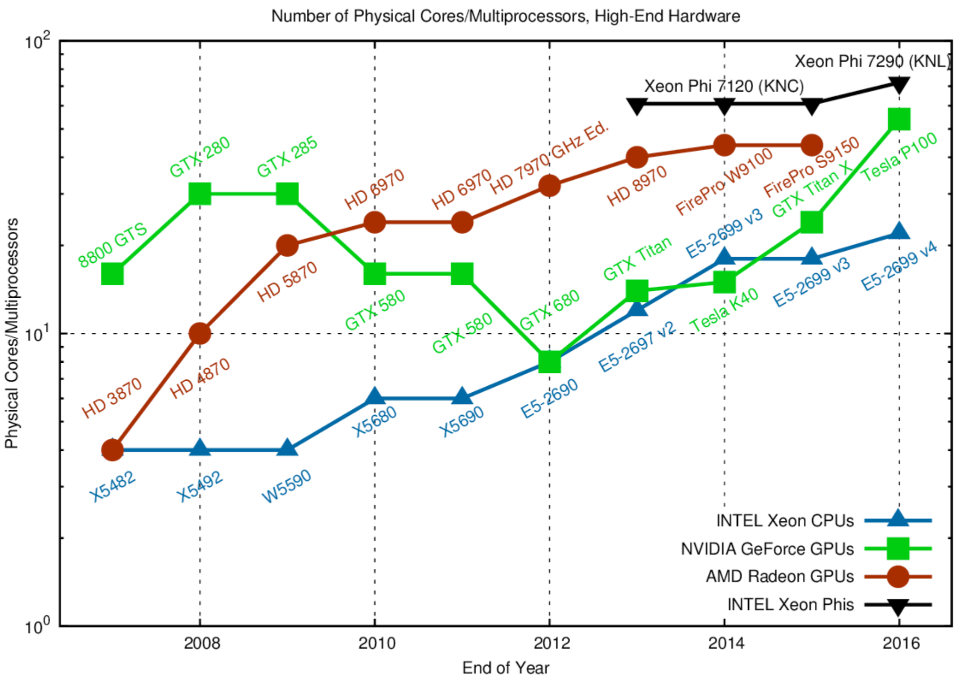
# cores/processing elements in several devices

Key question: what is a **core**?

a) IU+FPU? *GPU-type...*

b) A SIMD processor? *CPU-type..*

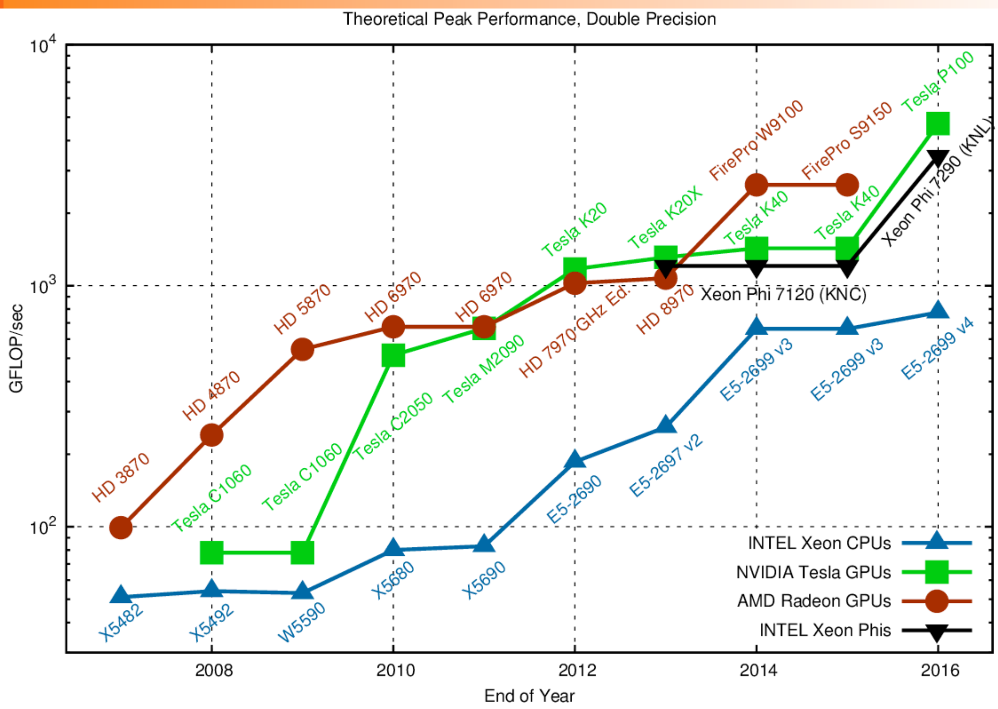
This updated slide and in this course: - **b)**



Note: the web link with these plots was updated in Aug'16

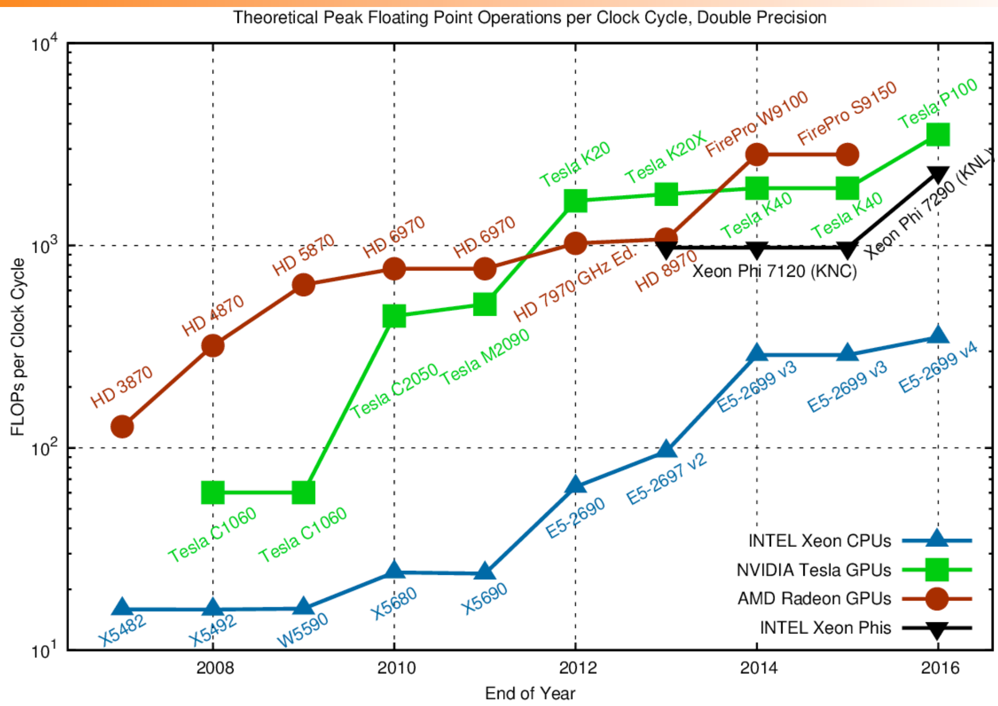
http://www.karlrupp.net/2013/06/cpu-gpu-and-mic-hardware-characteristics-over-time/

Theoretical peak performance in several computing devices (DP)



http://www.karlrupp.net/2013/06/cpu-gpu-and-mic-hardware-characteristics-over-time/

Theoretical peak FP Op's per clock cycle in several computing devices (DP)



http://www.karlrupp.net/2013/06/cpu-gpu-and-mic-hardware-characteristics-over-time/

NVIDIA GPU Architecture

Graphical Processing Units

- Similarities to vector machines:
  - Works well with data-level parallel problems
  - Scatter-gather transfers
  - Mask registers
  - Large register files
- Differences:
  - No scalar processor
  - Uses multithreading to hide memory latency
  - Has many functional units, as opposed to a few deeply pipelined units like a vector processor