Master in High Performance Computing

# P1.2 course:

# DAY 2: Introduction to modern CPUs

Stefano Cozzini

CNR/IOM and eXact-lab srl

ma per seguir virtute e canoscenza
SISSA
Scuola Internazionale Superiore di Studi Avanzati

ICTP The Abdus Salam International Centre for Theoretical Physics

MHPC
Master in High Performance Computing

# Outline

- Moore law
- Components of a modern CPU
- Von Neumann architecture
- Modern features of CPU and cores
  - SIMD
  - Pipelines
  - superscalar processing and ILP
- Survey of modern CPUs for HPC
- Final consideration

MHPC
Master in High Performance Computing

## Moore's law

- The number of transistors in a dense integrated circuit doubles approximately every two years.

  [ Gordon Moore, the co-founder of Intel and Fairchild Semiconductor, 1965 ]

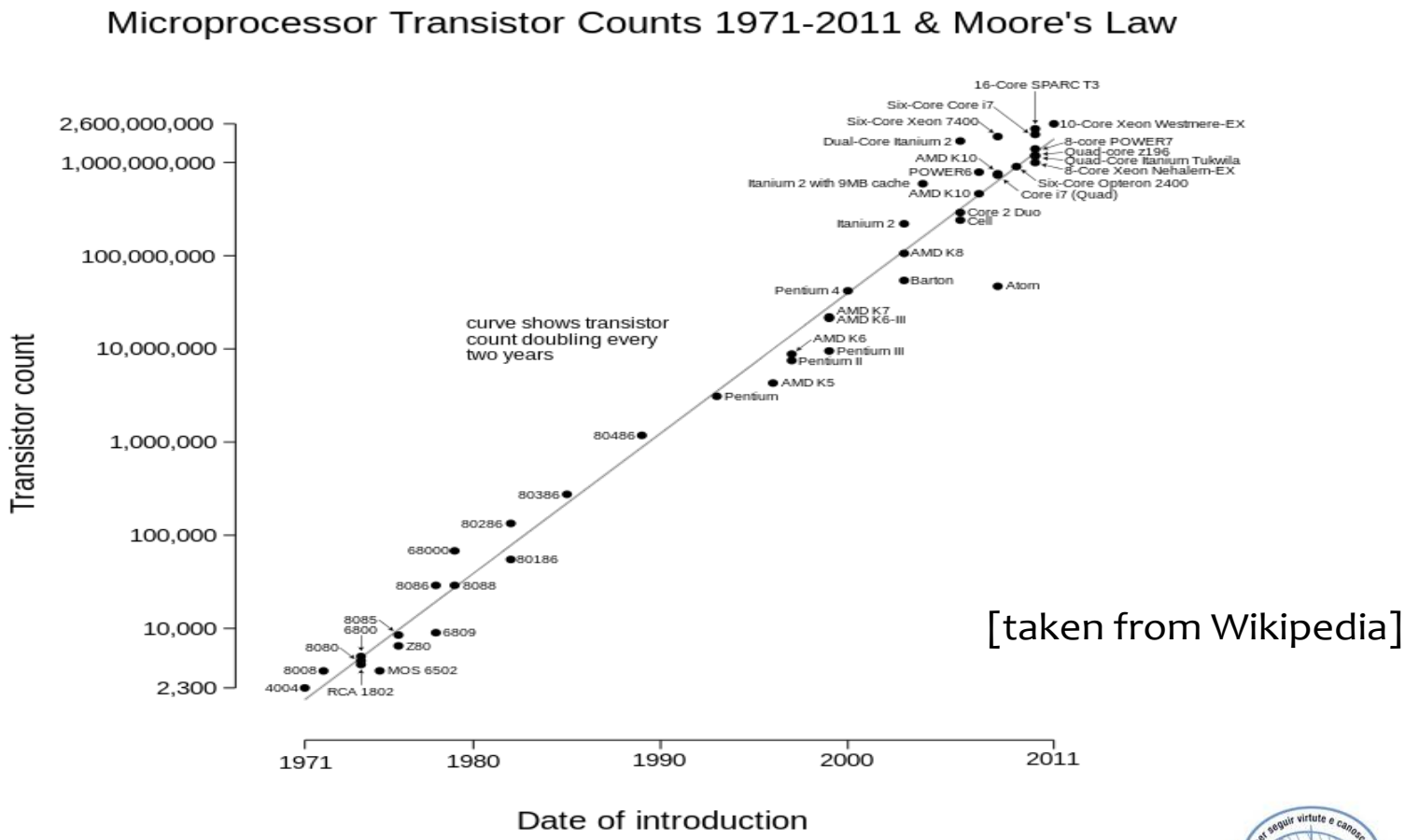
- The period is often quoted as 18 months because of Intel executive David House, who predicted that chip performance would double every 18 months (being a combination of the effect of more transistors and the transistors being faster)

[adapted from Wikipedia]

SISSA ICTP

MHPC
Master in High Performance Computing

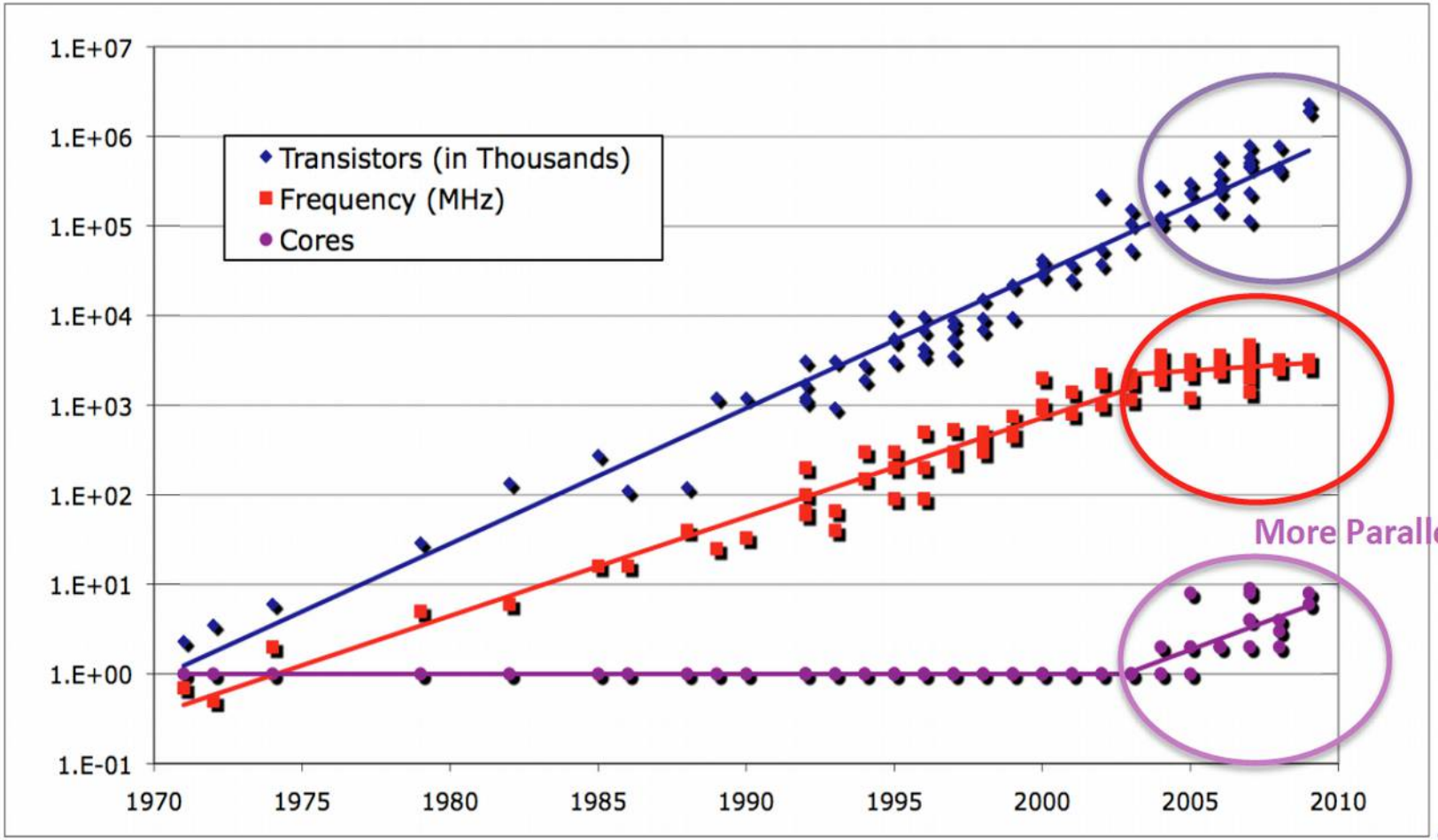
# Moore's law:




[taken from Wikipedia]

SISSA
ICTP

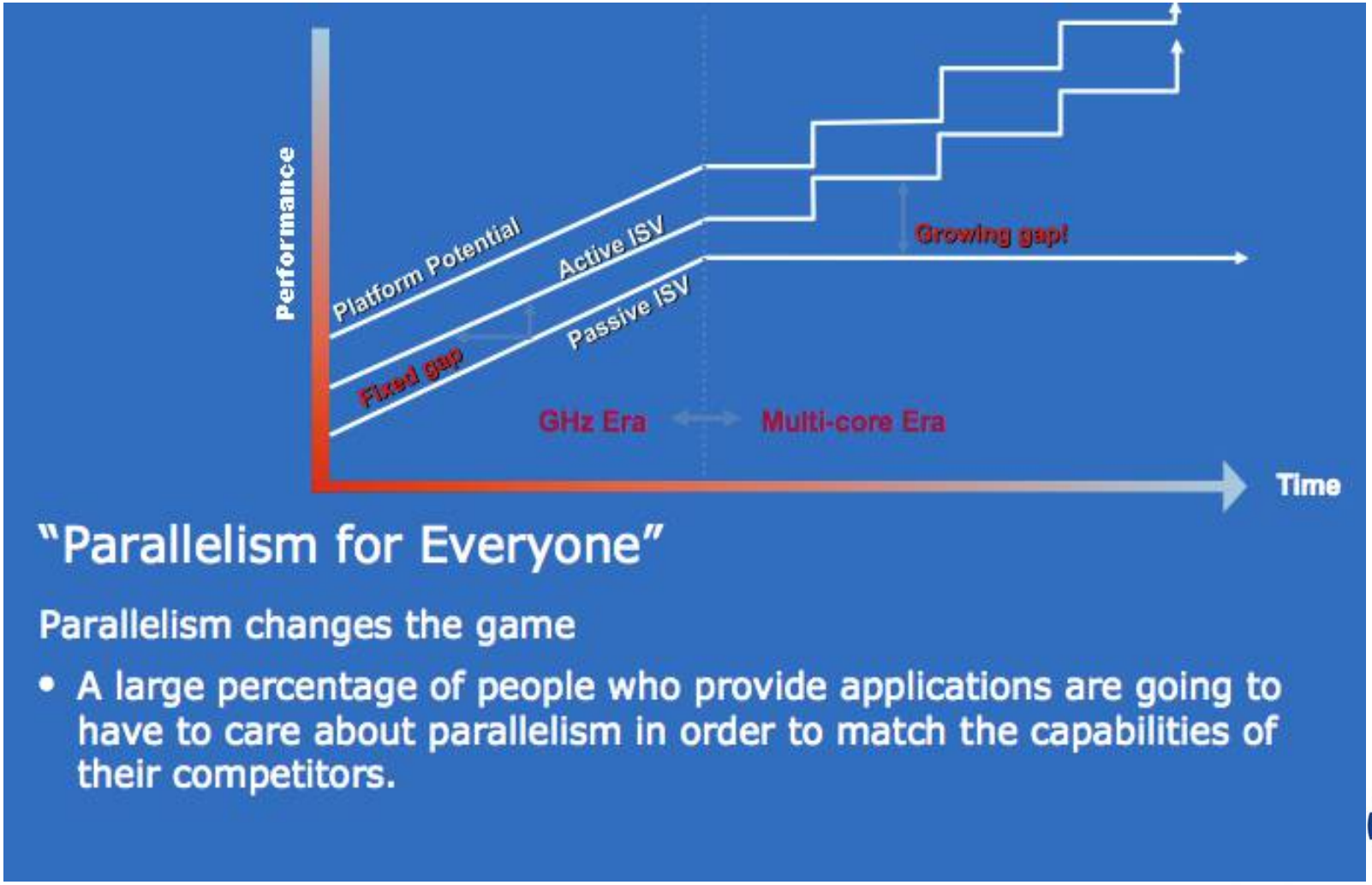
# CPUs and Moore law



## Observations

- Silicon technology is slowing down Moore's law
- End of the law in 5 years ?
- Exploitation of Moore law means parallelism at all level..
- We explore it at core/cpu level today..

## a picture from Intel..



MHPC
Master in High Performance Computing

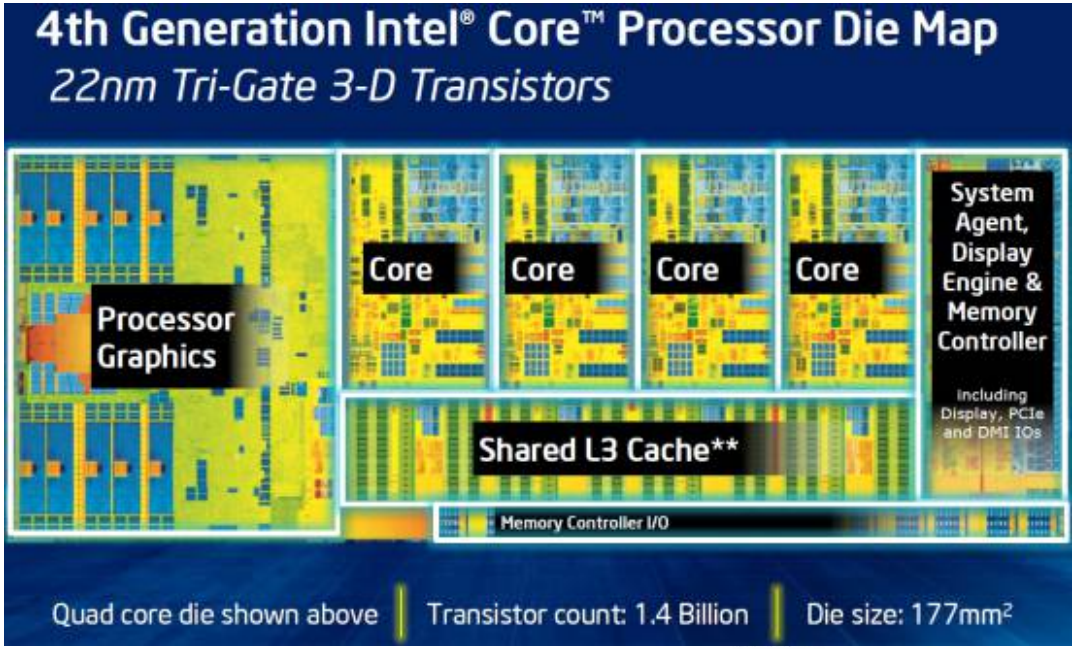
## Something to read on Moore's law

- https://www.technologyreview.com/s/601441/moores-law-is-dead-now-what/
- https://www.kth.se/social/upload/507d1d3af276540519000002/Moore's%20law.pdf
  (available on the github repository)

SISSA
ICTP

MHPC
Master in High Performance Computing

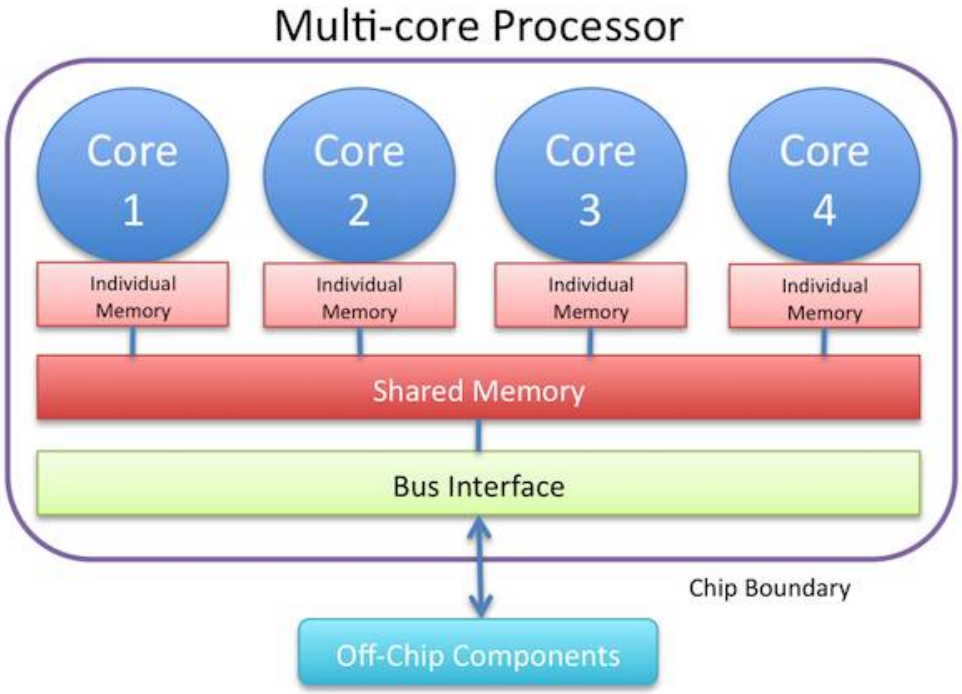
# What does a CPU look like?

MHPC
Master in High Performance Computing

## Modern CPU are multicore

- Because of power, heat dissipation, etc increasing tendency to actually lower clock frequency but pack more computing cores onto a chip.
- These cores will share some resources, e.g. memory, network, disk, etc but are still capable of independent calculations



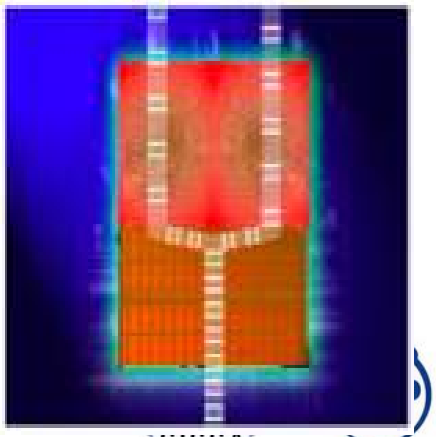

Picture from: http://www.cse.wustl.edu/~jain/cse567-11/ftp/multcore/

SISSA ICTP

MHPC
Master in High Performance Computing

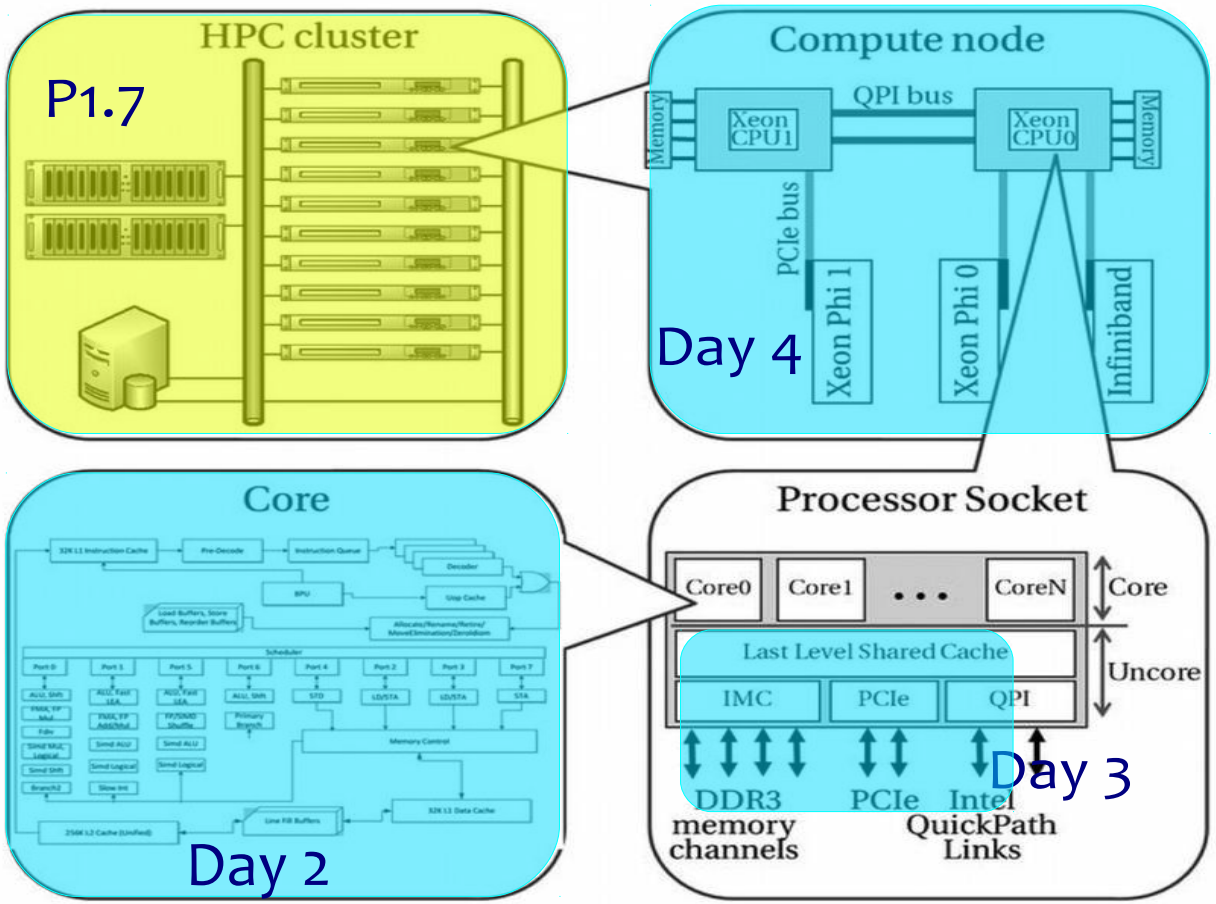
## What are cpu multi-core processors?

- Integrated circuit (IC) chips (CPU) containing more than one identical physical processor (core) . OS perceives each core as a discrete processor.
- Each core has its own complete set of resources, and may share the on-die cache layers (generally)
- Cores may have on-die communication path to front-side bus (FSB)
- What is a multi processor?
  - a collection of multicore cpus !



MHPC
Master in High Performance Computing

# First steps of the journey: core




SISSA

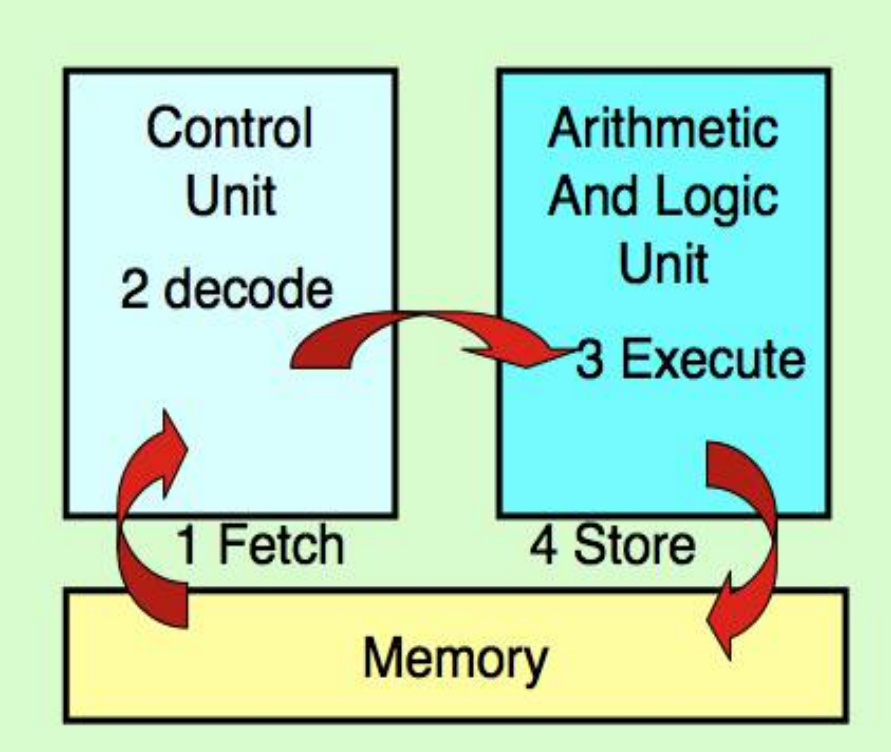
ICTP

## What is a core?

Level 0 approximation:
von Nuemann architecture

**Von Nuemann architecture:**

- Control Unit: processes instructions  ALU: math and logic operations
- At each cycle the CPU fetches both data and a description of what operations need to be performed  and stores them in registers.



MHPC
Master in High Performance Computing

## Instruction Set Architecture (ISA)

- It is the boundary between SW and HW
- The deeper level accessible to the programmers
- The interface between the programmer and the microarchitecture
- Different microarchitectures can have the same ISA (binary Compatible)
- Different generation of microarchitectures can be backward compatible

SISSA
ICTP

MHPC
Master in High Performance Computing

# Analysis of a simple program on Von Neumann architecture:

```
void store(double *a, double *b,
double *c) {
 *c = *a + *b;
}
```

```
[exact@master ~]$ gcc -O2 -S -o - frammento.c
        .file "frammento.c"
        .text
        .p2align 4,,15
.globl store
        .type   store, @function
store:
.LFB0:
        .cfi_startproc
        movsd   (%rdi), %xmm0   #load *a to mmx0
        addsd   (%rsi), %xmm0   # load b and add to *a
        movsd   %xmm0, (%rdx)  # store to C
        ret
        .cfi_endproc
.LFE0:
        .size   store, .-store
        .ident  "GCC: (GNU) 4.4.7 20120313 (Red Hat 4.4.7-4)"
        .section .note.GNU-stack,"",@progbits
```

SISSA ICTP

**Workflow:**

- Instruction fetch
- Instruction decode: determine operation and operands
- Memory fetch: Get operands from memory
- Perform operation
- Write results back
- Continue with next instruction

## How to go faster ?

- since many years CPU designers have been trying to improve and optimize the "Classical Model":
  - increasing clock frequency (Moore's law)
  - reducing the number of cycles needed to perform a single instruction (a = c + d)
    - flow/branch prediction
  - improve the number of outputs x cycle
    - Instructions Level Parallelism (ILP)
    - Execution => in-order Vs. out-of-order

MHPC
Master in High Performance Computing

# Contemporary cores are a little bit more complicated..

SISSA
ICTP

# What is in a core really ?

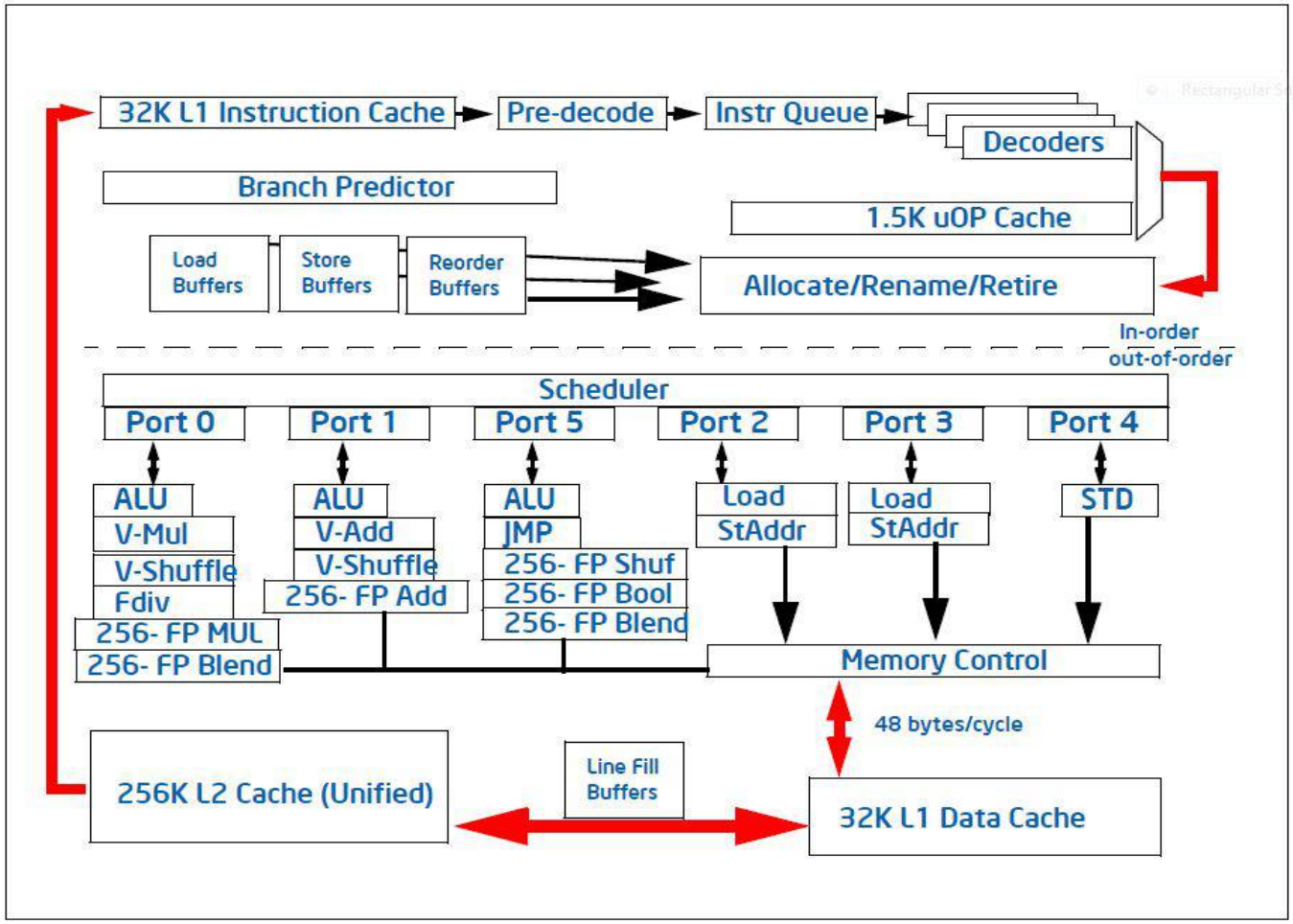



Figure 2-1. Intel microarchitecture code name Sandy Bridge Pipeline Functionality

MHPC
Master in High Performance Computing

## Instruction handling: out of order execution.

- It is unrealistic on modern CPU to assume all the instruction are executed sequentially
- “out of order” instruction handling has been present for almost 20 years
- Processor/core is allowed to change the order of the instructions in your code.
  - Be careful on this on Floating Point operations
- Several units on modern core are dedicated to this

# Instruction handling sections on Sandybridge core:

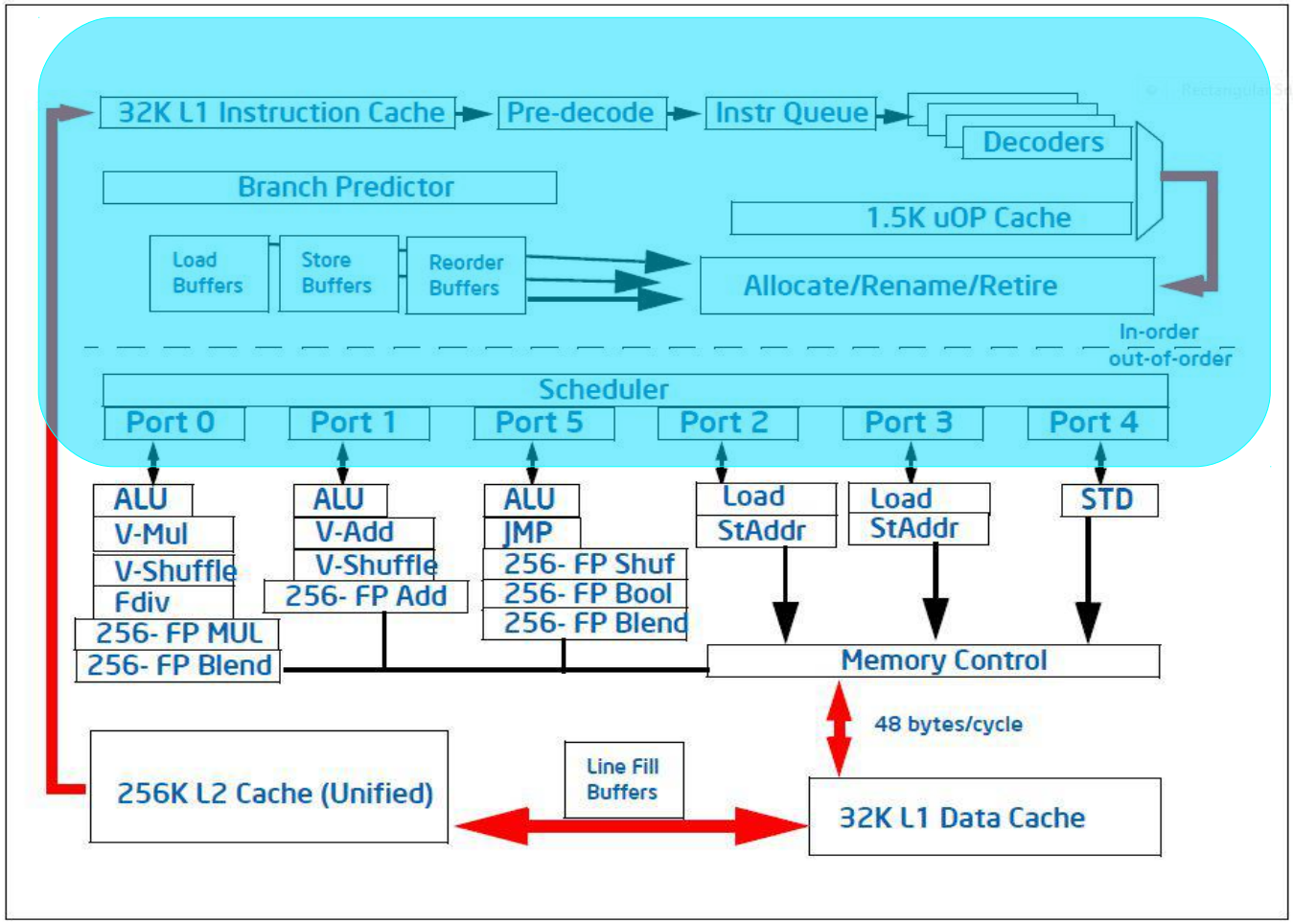



Figure 2-1. Intel microarchitecture code name Sandy Bridge Pipeline Functionality

MHPC
Master in High Performance Computing

## Many functional unit

- Circuitry on the chip which performs a given type of operation on operands in registers is known as a functional unit.
- For HPC we are mainly interested in Floating Point Units
- Some FPUs can perform more than one operation per cycle..
  - Multiple floating point units
  - 1 Mul + 1 Add, or 1 Fused Multiply-Add (FMA)

$x \leftarrow c*x+y$

SISSA ICTP

## What else in the cores ?

- On modern CPU/Cores there are many other stuff:
  - Superscalar execution
  - Pipelined functional units
  - Floating point instruction set extensions

MHPC
Master in High Performance Computing

## Superscalar CPU architecture

- The ability to issue/execute more than one instruction at the same time
- aka Instruction level parallelism (ILP)
- On sandybridge:
  - Six issue ports/ execution units that can execute Instruction simultaneously
  - e.g. Floating Point-Add (Port 1) and Floating-Point Mult (Port 0)

SISSA
ICTP

# Superscalarity on Sandybridge

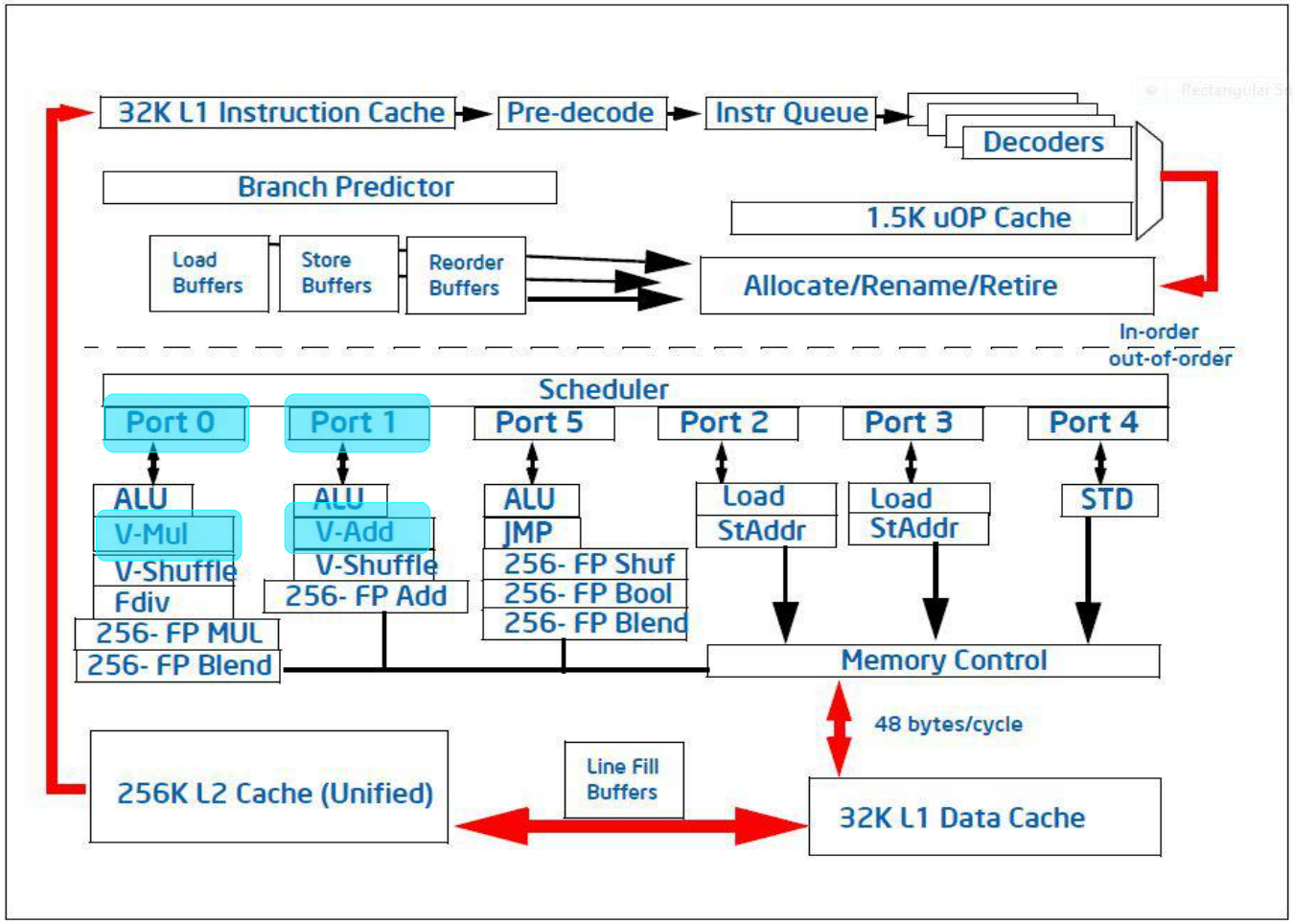



Figure 2-1. Intel microarchitecture code name Sandy Bridge Pipeline Functionality

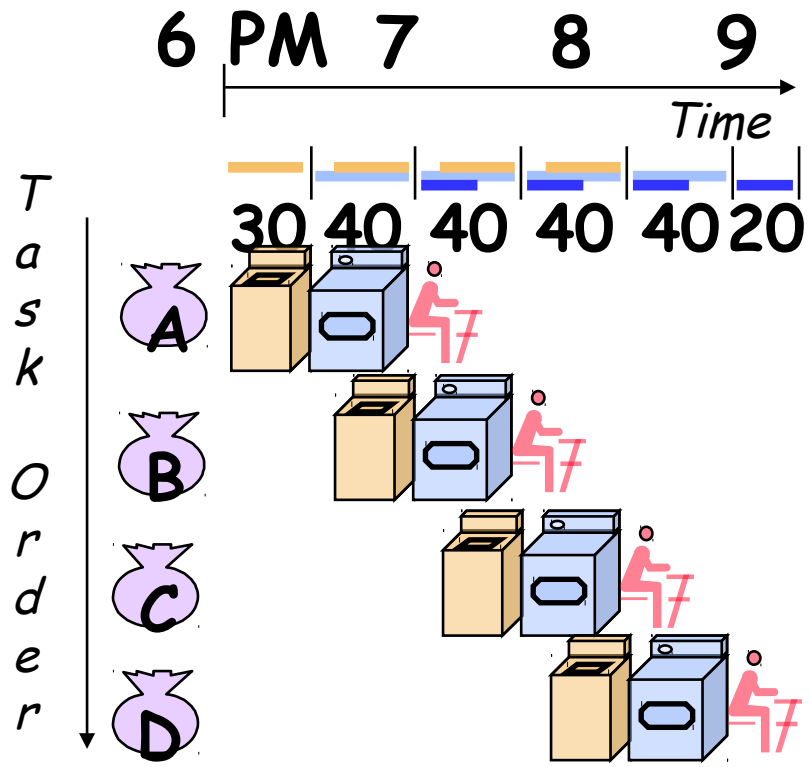
## Pipelined Functional Units

- Most integer and floating point functional units are pipelined, meaning that they can have multiple independent executions of the same instruction placed in a queue.
- The idea is that after an initial startup latency, the functional unit should be able to generate one result every clock period (CP).
- Each stage of a pipelined operation can be working simultaneously on different sets of operands.
- For each stage of the pipeline there is a dedicated HW

MHPC
Master in High Performance Computing

# What is Pipelining?

Dave Patterson's Laundry example: 4 people doing laundry
wash (30 min) + dry (40 min) + fold (20 min)




In this example:

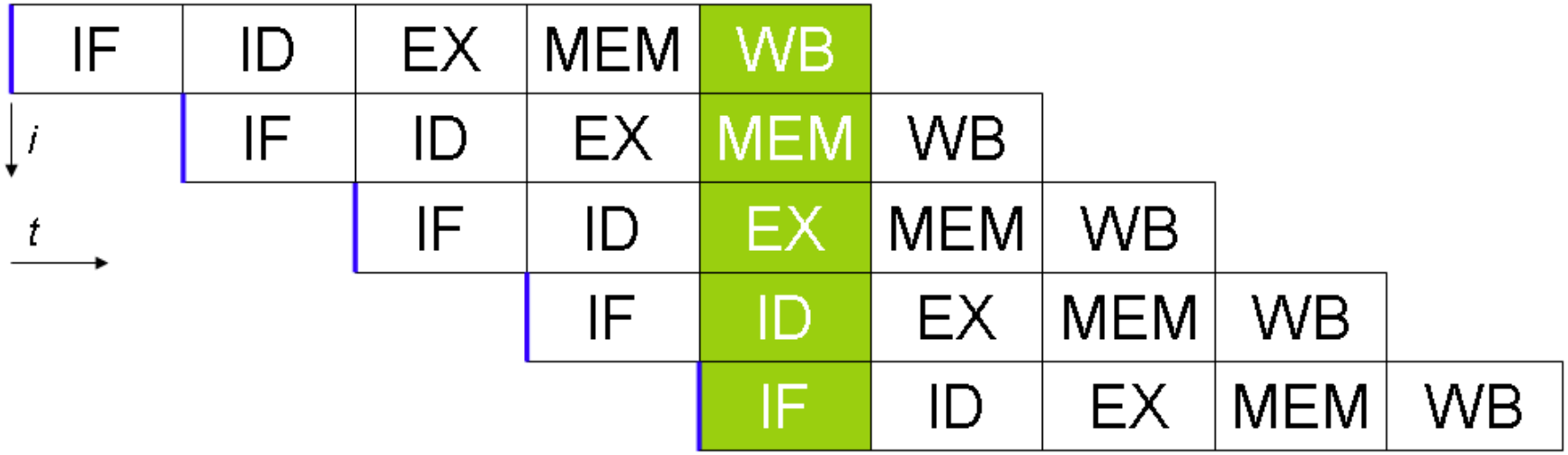
Sequential execution takes 4 * 90min = 6 hours

Pipelined execution takes 30+4*40+20 = 3.3 hours

SISSA
ICTP

MHPC
Master in High Performance Computing

# The standard RISC 5 stage pipeline




IF = Instruction Fetch
ID = Instruction Decode
EX = Execute,
MEM = Memory access
WB = Register write back

SISSA
ICTP

MHPC
Master in High Performance Computing

## Pipeline concepts/jargon

- Wind-up phase: time it takes to load the pipeline (a.k.a. latency)
- Wind-down phase: time it takes to drain the pipeline
- Segments=stages=steps of the pipeline
- Speedup: *Tseq*/*Tpipe*
- Throughput = *N*/*Tpipe*
- Pipelining helps throughput, but not latency
- Pipeline rate limited by slowest pipeline stage

# Pipeline analysis: $N_{1/2}$

Given a pipeline with *s* segments and *N* operations:

- $T_{seq}$ (the time without pipelining) = $sN$
- $T_{pipe} = s+N-1$

  → Speed-up= $s*N/s+N-1$

  Throughput= $N/T_{pipe} = 1/(1 + (s-1)/N)$
- for N large enough
  - *Speedup ~ s*
  - *Throughput ~ 1*

    i.e. 1 result per clock cycle
- With *N* operations, actual rate is $N/(s+N)$
- This is half of the asymptotic rate if $s=N$

# Asymptotic behavior of pipeline throughput

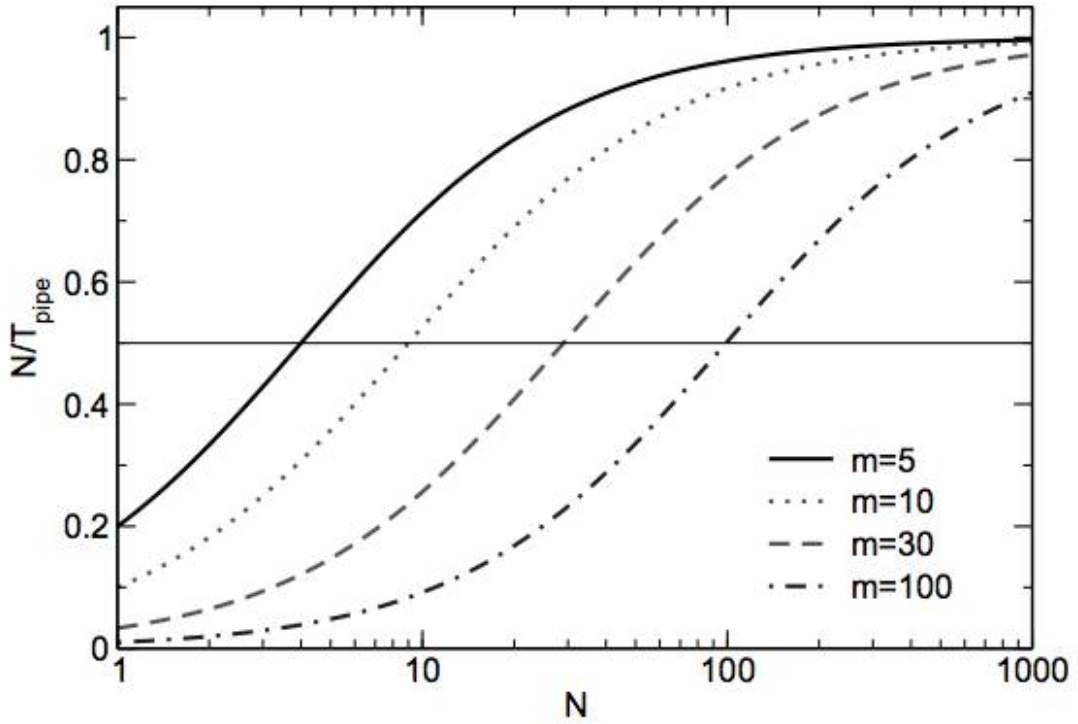


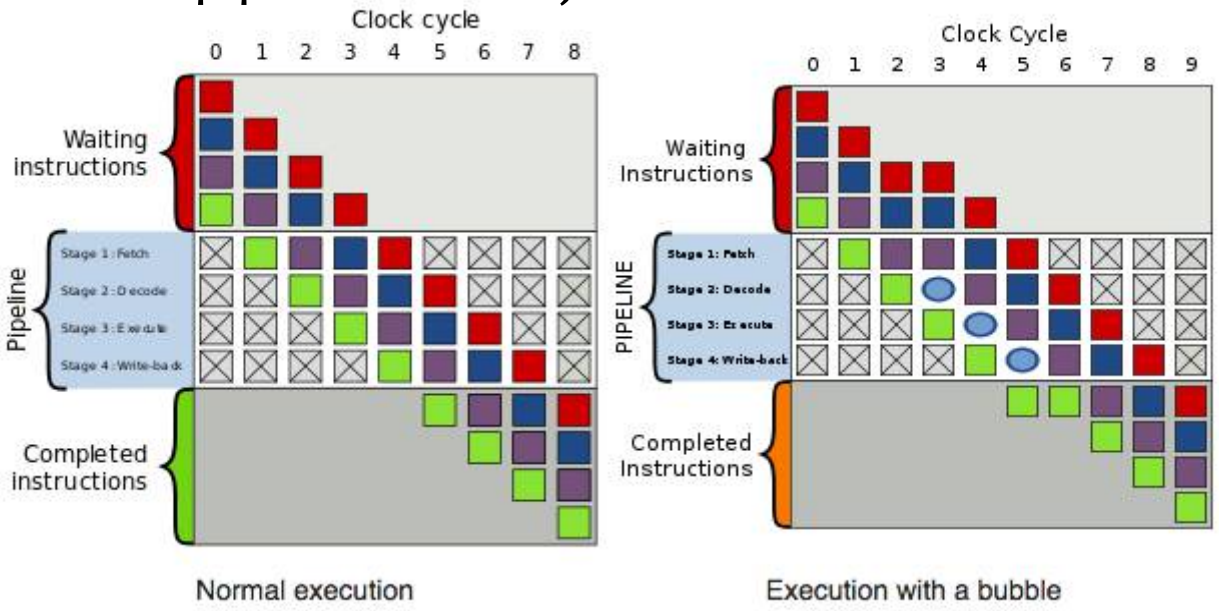

**Figure 1.6:** Pipeline throughput as a function of the number of independent operations. *m* is the pipeline depth.

# Pipeline drawbacks

Standard lengths of modern processor: from 10 to 35

- Not efficient at all if loops are short and tight

- Not efficient if there are jumps (branches) or stalling in instructions (the so called pipeline bubble)



MHPC
Master in High Performance Computing

## Branch Prediction

- The "instruction pipeline" is all of the processing steps (also called segments) that an instruction must pass through to be "executed".
- Higher frequency machines have a larger number of segments.
- Branches are points in the instruction stream where the execution may jump to another location, instead of executing the next instruction.
- For repeated branch points (within loops), instead of waiting for the loop to branch route outcome, it is predicted.

SISSA
ICTP

# Floating Point Instruction Set Extensions

- additional floating point instructions beyond the usual floating point add and multiply instructions:
  - Square root instruction --usually not pipelined!
    - AMD Opteron / Intel Xeon
  - SIMD (a.k.a. vector) floating point instructions
    - AMD Opteron/ Intel Xeon
- Combined floating point multiply/add (MADD) instruction
  - AMD Opteron ("Barcelona" and after, using SIMD)
  - Intel Xeon ("Woodcrest" and after, using SIMD)

MHPC
Master in High Performance Computing

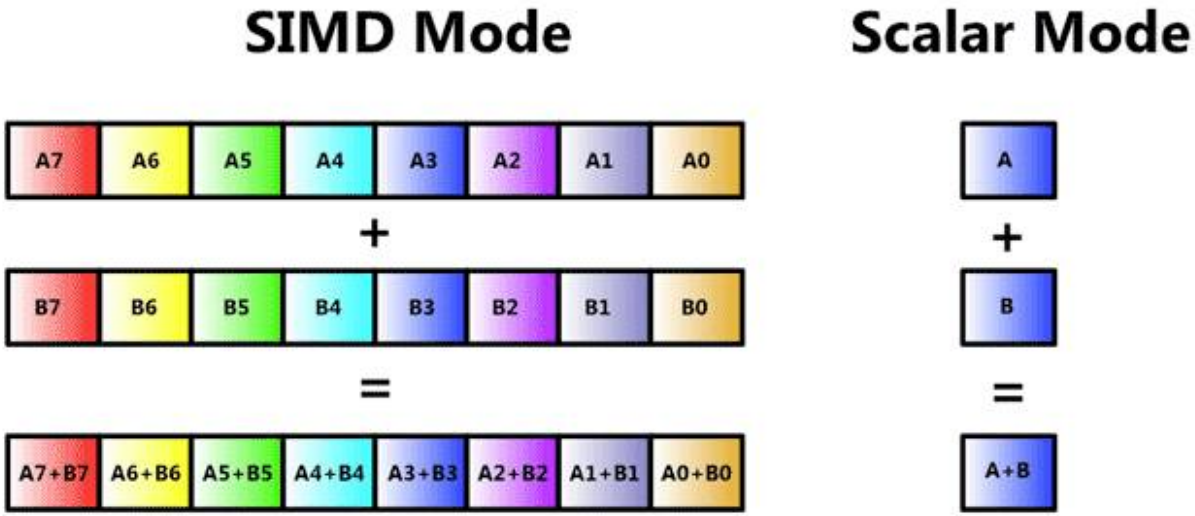
# Instruction Set Extensions

- Intel
  - MMX (Matrix Math eXtensions)
  - SSE (Streaming SIMD Extensions)
  - SSE2 / SSE4.2
    - AVX AVX-512 (Advanced vector eXtension)
      - From sandybridge processor on
- AMD
  - 3DNow!
  - AMD 3DNow!+ (or 3DNow! Professional, or 3DNow! Athlon) ...
- To check what you have on your machine:
  - cat /proc/cpuinfo
  - to enable them: use appropriate compiler flag..

SISSA ICTP

MHPC
Master in High Performance Computing

# SIMD approach:



## The AVX Instruction Set

Latest version of Intel and GNU compiler are in most cases able to perform auto-vectorization on simple loop constructs and with simplified data-structures

Machine-dependent vector intrinsic that use a function call syntax to emit machine code (to be used at large when compilers not work as expected)

Memory alignment is required

It is available as compiler option on both Intel and GNU compiler:

-mavx (GNU) | -xAVX (INTEL)

# Intel SIMD evolution

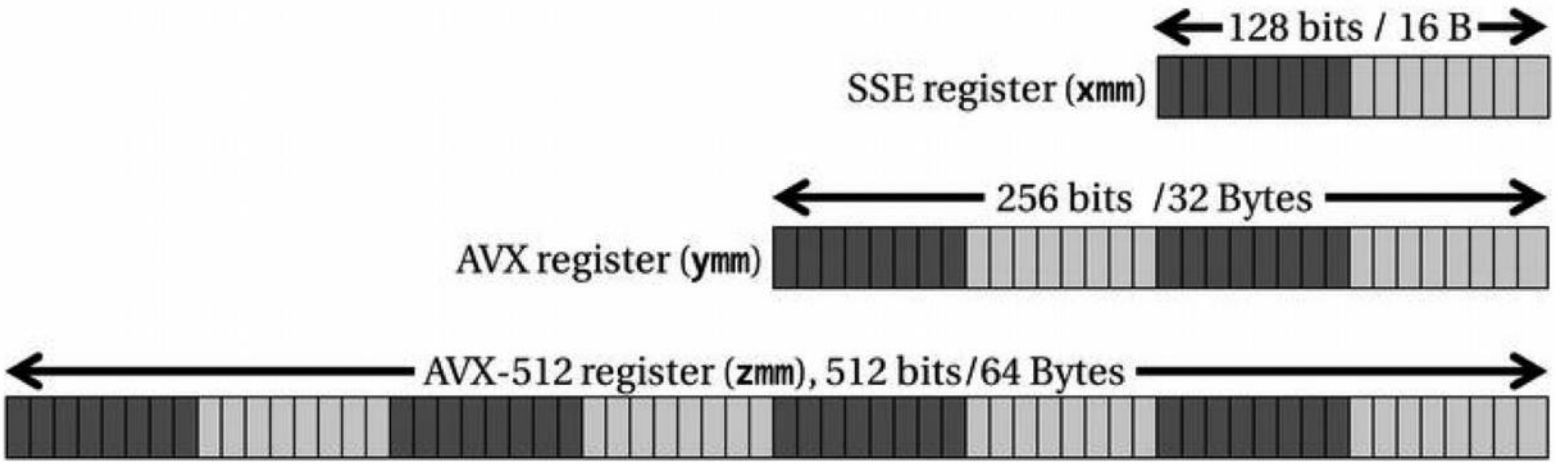



*Figure 2-9. SSE, AVX, and AVX-512 vector registers with packed 64-bit numbers*

MHPC
Master in High Performance Computing

## Number crunching on CPU: what do we count ?

- Rate of [million/billions of] floating point operations per second ([M|G]flops) FLOPs/S
- Theoretical peak performance:

determined by counting the number of floating-point additions and multiplications that can be completed during a period of time, usually the cycle time of the machine

FLOPS=Clock-rate*Number_of_FP_ operation*Number_of_cores

SISSA ICTP

MHPC
Master in High Performance Computing

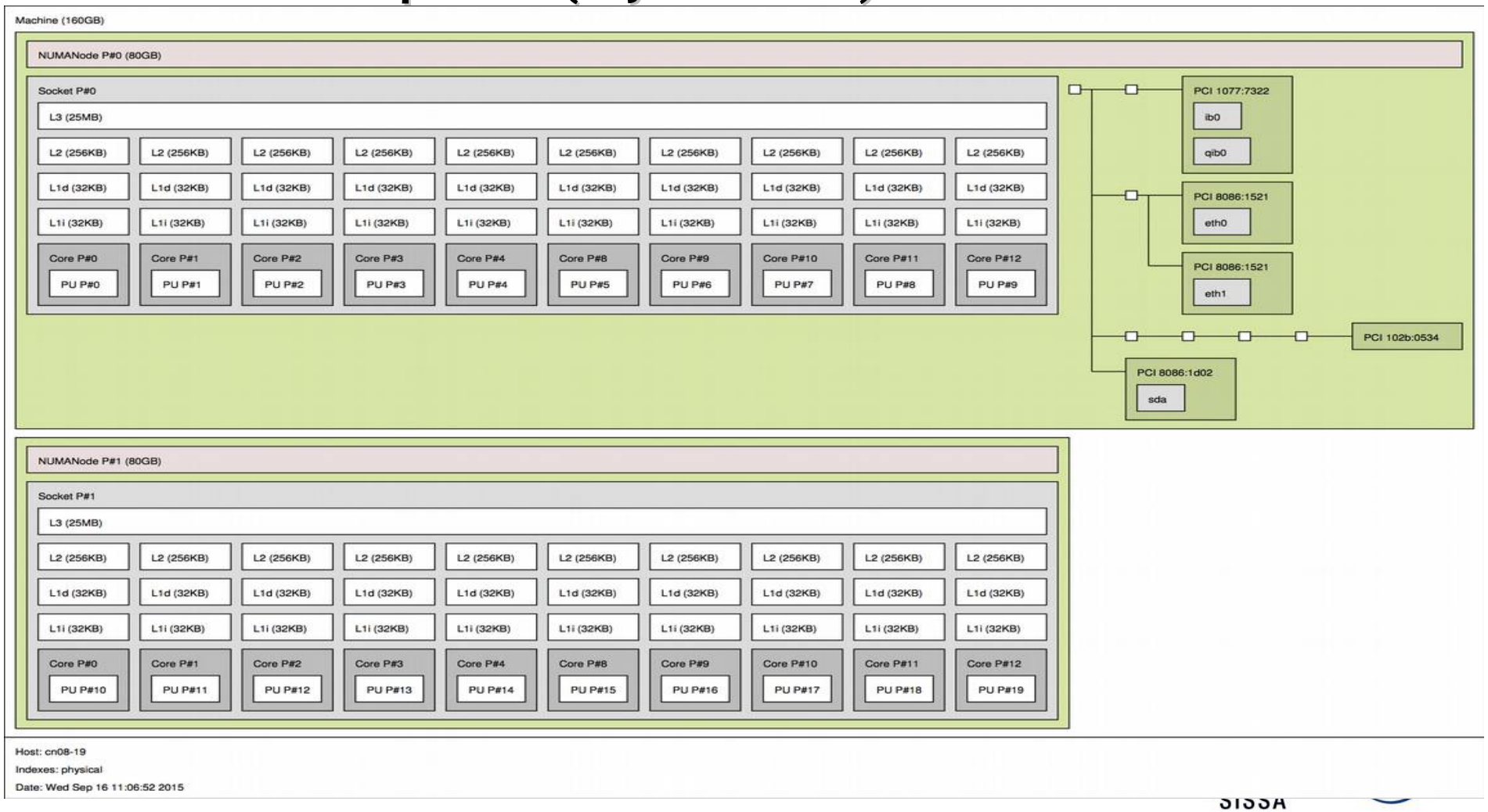
**Exercise: compute Theoretical Peak performance for your laptop/desktop:**

- Identify the CPU
- Identify the frequency
- Identify the number of floating point for cycle
- Identify how many cores
- Put all together in one single number

SISSA
ICTP

MHPC
Master in High Performance Computing

# A modern node picture (Ulysses-node)



MHPC
Master in High Performance Computing

**Peak performance on ulysses node:**

- Two Sockets
- 10 Cores per Socket @ 2.6 GHz
- Two-way superscalar with respect to floating-point
  - AVX Mult on Port 0, AVX Add on Port 1
- 256b AVX
    - Four double-precision operations with a single instruction

2 x 10 x 2.6 GHz x 2 instr/cycle x 4 ops/instr = ????? Gflop/s

SISSA ICTP

MHPC
Master in High Performance Computing

# What CPU for HPC ?

| Processor Generation | Count | System Share (%) | Rmax (GFlops) | Rpeak (GFlops) | Cores |
|---|---|---|---|---|---|
| Intel Xeon E5 (Haswell) | 221 | 44.2 | 172,605,096 | 271,881,324 | 8,636,254 |
| Intel Xeon E5 (IvyBridge) | 118 | 23.6 | 109,146,786 | 187,108,623 | 8,639,336 |
| Intel Xeon E5 (SandyBridge) | 66 | 13.2 | 48,003,244 | 67,528,012 | 3,222,875 |
| Intel Xeon E5 (Broadwell) | 21 | 4.2 | 28,625,858 | 33,317,165 | 940,140 |
| Power BQC | 19 | 3.8 | 47,047,481 | 55,155,100 | 4,308,992 |
| Xeon 5600-series (Westmere-EP) | 9 | 1.8 | 9,662,361 | 19,393,022 | 641,068 |
| Intel Xeon E7 (IvyBridge) | 8 | 1.6 | 3,291,391 | 6,954,496 | 398,880 |
| Intel Xeon E7 (Haswell-Ex) | 7 | 1.4 | 3,429,884 | 5,678,592 | 159,456 |
| Opteron 6300 Series "Abu Dhabi" | 5 | 1 | 1,894,061 | 6,710,298 | 656,128 |
| SPARC64 Xlfx | 5 | 1 | 10,422,200 | 11,530,310 | 354,384 |
| Opteron 6200 Series "Interlagos" | 5 | 1 | 20,216,100 | 30,720,688 | 862,264 |
| POWER7 | 4 | 0.8 | 2,788,135 | 3,337,539 | 112,448 |
| Opteron 6100-series "Magny-Cours" | 3 | 0.6 | 1,704,800 | 2,354,995 | 232,704 |
| Xeon 5500-series (Nehalem-EP) | 2 | 0.4 | 930,000 | 1,510,046 | 71,880 |
| Xeon E7 (Westmere-EX) | 1 | 0.2 | 354,300 | 655,258 | 38,400 |
| ShenWei | 1 | 0.2 | 795,900 | 1,070,160 | 137,200 |
| Intel Xeon Phi | 1 | 0.2 | 817,847 | 1,474,458 | 32,912 |
| SPARC64 IXfx | 1 | 0.2 | 1,043,000 | 1,135,411 | 76,800 |
| Xeon 5500-series (Nehalem-EX) | 1 | 0.2 | 1,050,000 | 1,254,550 | 138,368 |
| SPARC64 VIIIfx | 1 | 0.2 | 10,510,000 | 11,280,384 | 705,024 |
| Sunway | 1 | 0.2 | 93,014,594 | 125,435,904 | 10,649,600 |

## Some considerations

- Intel is dominating
- AMD is disappearing (or already disappeared)
- IBM is coming back with Power8 architecture
- ARM processor still to come

MHPC
Master in High Performance Computing

# Intel Xeon history...

| The Process Gap Between PCs And Servers At Intel | | | | | | |
|---|---|---|---|---|---|---|
| Chip Family | Process | PC Chip | Launch Date | Server Chip | Launch Date | Months |
| Nehalem | 45 nm | Core i7 | Nov-08 | Xeon 5500 | Mar-09 | 4 |
| Westmere | 32 nm | Core i7 | Jul-10 | Xeon 5600 | Feb-11 | 7 |
| Sandy Bridge | 32 nm | Core i7 | Jan-11 | Xeon E5 | Mar-12 | 14 |
| Ivy Bridge | 22 nm | Core i7 | Apr-12 | Xeon E5 v2 | Sep-13 | 17 |
| Haswell | 22 nm | Core i7 | Jun-13 | Xeon E5 v3 | Sep-14 | 15 |
| Broadwell | 14 nm | Core i7 | Jan-15 | Xeon E5 v4 | *Mar-16* | *14* |
| Skylake | 14 nm | Core i7 | *Aug-15* | Xeon E5 v5 | *Mar-17* | *19* |
| Kaby Lake | 14 nm | Core i7 | *Aug-16* | Xeon E5 v6 | *Mar-18* | *19* |
| Cannonlake | 10 nm | Core i7 | *Aug-17* | Xeon E5 v7 | *Mar-19* | *19* |
| ??? | 10 nm | Core i7 | *Aug-18* | Xeon E5 v8 | *Mar-20* | *19* |
| ??? | 10 nm | Core i7 | *Aug-19* | Xeon E5 v9 | *Mar-21* | *19* |
| ??? | 7 nm | Core i7 | *Aug-20* | Xeon E5 v10 | *Mar-22* | *19* |
| ??? | 7 nm | Core i7 | *Aug-21* | Xeon E5 v11 | *Mar-23* | *19* |
| ??? | 7 nm | Core i7 | *Aug-22* | Xeon E5 v12 | *Mar-24* | *19* |

*Source: The Platform*

SISSA ICTP

MHPC
Master in High Performance Computing

# Intel processor families:

**High Core Count Top Bin**

| Family | Model | Clock Speed | Cores / Threads | 1K Tray Unit Price | Rel Perf | $ / Rel Perf |
|---|---|---|---|---|---|---|
| Nehalem | X5570 | 2.93 GHz | 4 / 8 | $1,386 | 1.16 | $1,195 |
| Westmere | X5680 | 3.33 GHz | 6 / 12 | $1,663 | 1.98 | $840 |
| Sandy Bridge | E5-2690 | 2.9 GHz | 8 / 16 | $2,057 | 2.55 | $806 |
| Ivy Bridge | E5-2697 v2 | 2.7 GHz | 12 / 24 | $2,614 | 3.73 | $702 |
| Haswell | E5-2699 v3 | 2.3 GHz | 18 / 36 | $4,115 | 5.20 | $792 |
| Broadwell | E5-2699 v4 | 2.2 GHz | 22 / 44 | $4,115 | 6.34 | $649 |

**Standard Top Bin**

| Family | Model | Clock Speed | Cores / Threads | 1K Tray Unit Price | Rel Perf | $ / Rel Perf |
|---|---|---|---|---|---|---|
| Nehalem | E5540 | 2.53 GHz | 4 / 8 | $744 | 1.00 | $744 |
| Westmere | E5640 | 2.66 GHz | 4 / 8 | $744 | 1.04 | $744 |
| Sandy Bridge | E5-2640 | 2.5 GHz | 6 / 12 | $885 | 1.65 | $536 |
| Ivy Bridge | E5-2640 v2 | 2.0 GHz | 8 / 16 | $885 | 1.84 | $481 |
| Haswell | E5-2640 v3 | 2.6 GHz | 8 / 16 | $940 | 2.61 | $360 |
| Broadwell | E5-2640 v4 | 2.4 GHz | 10 / 20 | $939 | 3.14 | $299 |

**Highest Clock Speed**

| Family | Model | Clock Speed | Cores / Threads | 1K Tray Unit Price | Rel Perf | $ / Rel Perf |
|---|---|---|---|---|---|---|
| Nehalem | X5570 | 2.93 GHz | 4 / 8 | $1,386 | 1.16 | $1,195 |
| Westmere | X5677 | 3.46 GHz | 4 / 8 | $1,663 | 1.36 | $1,223 |
| Sandy Bridge | E5-2643 | 3.3 GHz | 4 / 8 | $885 | 1.45 | $610 |
| Ivy Bridge | E5-2637 v2 | 3.5 GHz | 4 / 8 | $996 | 1.61 | $619 |
| Haswell | E5-2637 v3 | 3.5 GHz | 4 / 8 | $995 | 1.76 | $566 |
| Broadwell | E5-2637 v4 | 3.5 GHz | 4 / 8 | $996 | 1.83 | $543 |

SISSA
ICTP

# Intel® Xeon® Processor E5 v4 Product Family Overview

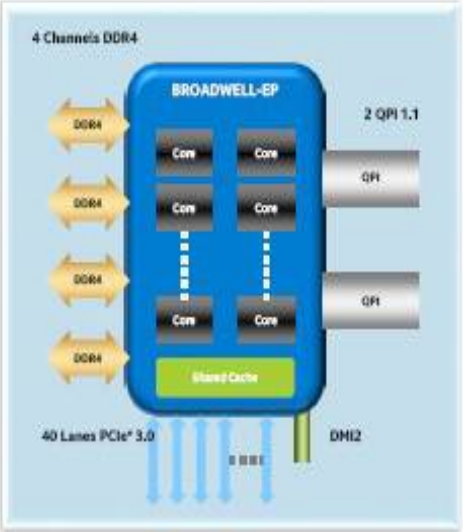
## New features:

- Broadwell microarchitecture
- Built on 14nm process technology
- Socket compatible◊ replacement for Intel® Xeon® processor E5-2600 v3 on Grantley

## New processor technologies:

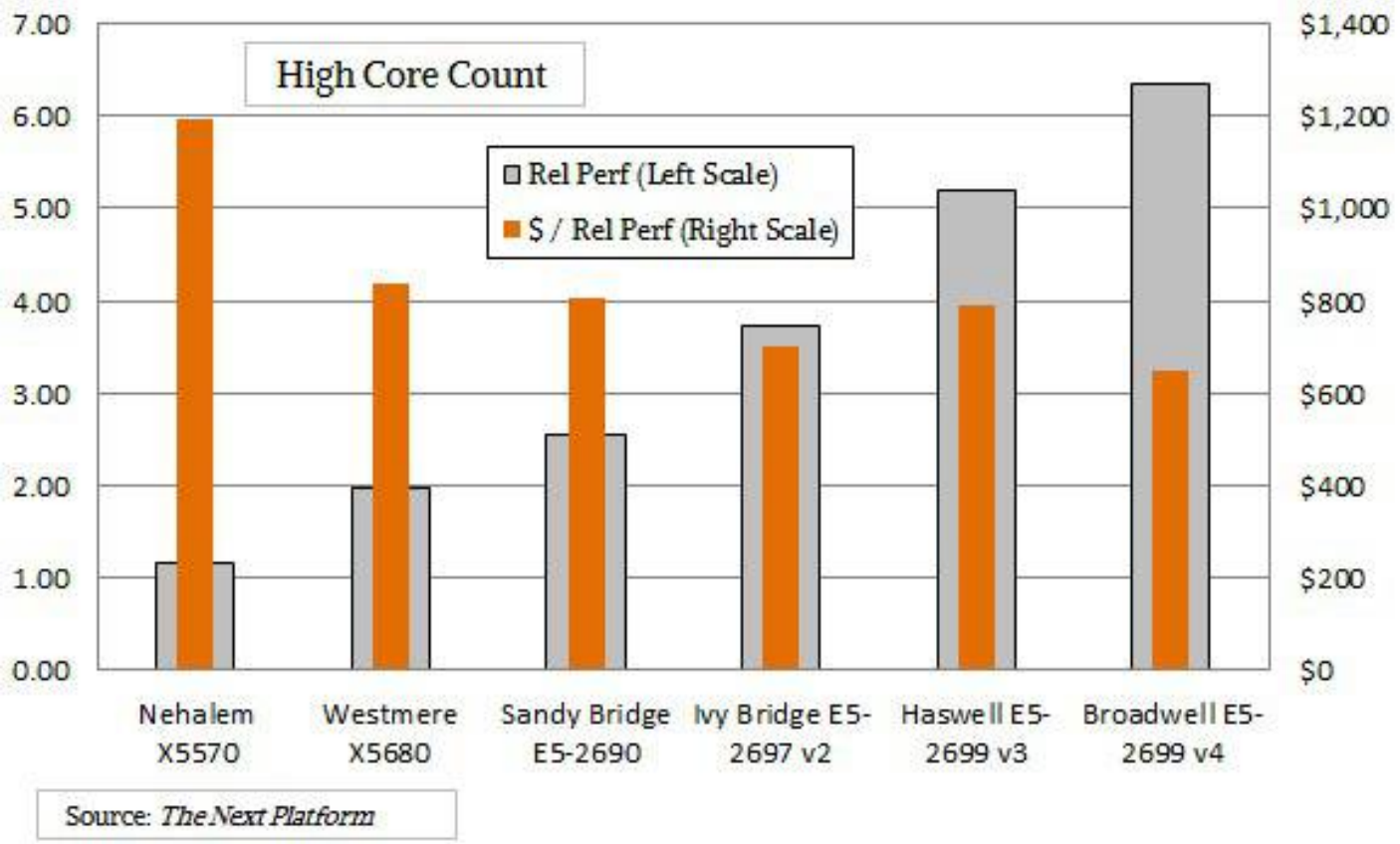
- Posted Interrupts
- Page Modification Logging
- Cache Allocation Technology
- Memory BW Monitoring
- Crypto Speedup
- Supervisor Mode Access Prevention
- New RDSEED instruction
- Intel® Processor Trace
- Hardware Controlled Power Management

| Features | Xeon E5-2600 v3 (Haswell-EP) | Xeon E5-2600 v4 (Broadwell-EP) |
|---|---|---|
| Cores Per Socket | Up to 18 | Up to 22 |
| Threads Per Socket | Up to 36 threads | Up to 44 threads |
| Last-level Cache (LLC) | Up to 45 MB | Up to 55 MB |
| QPI Speed (GT/s) | 2x QPI 1.1 channels 6.4, 8.0, 9.6 GT/s | |
| PCIe* Lanes / Speed(GT/s) | 40 / 10 / PCIe* 3.0 (2.5, 5, 8 GT/s) | |
| Memory Population | 4 channels of up to 3 RDIMMs or 3 LRDIMMs | + 3DS LRDIMM† |
| Memory RAS | ECC, Patrol Scrubbing, Demand Scrubbing, Sparing, Mirroring, Lockstep Mode, x4/x8 SDDC | + DDR4 Write CRC |
| Max Memory Speed | Up to 2133 | Up to 2400 |
| TDP (W) | 160 (Workstation only), 145, 135, 120, 105, 90, 85, 65, 55 | |



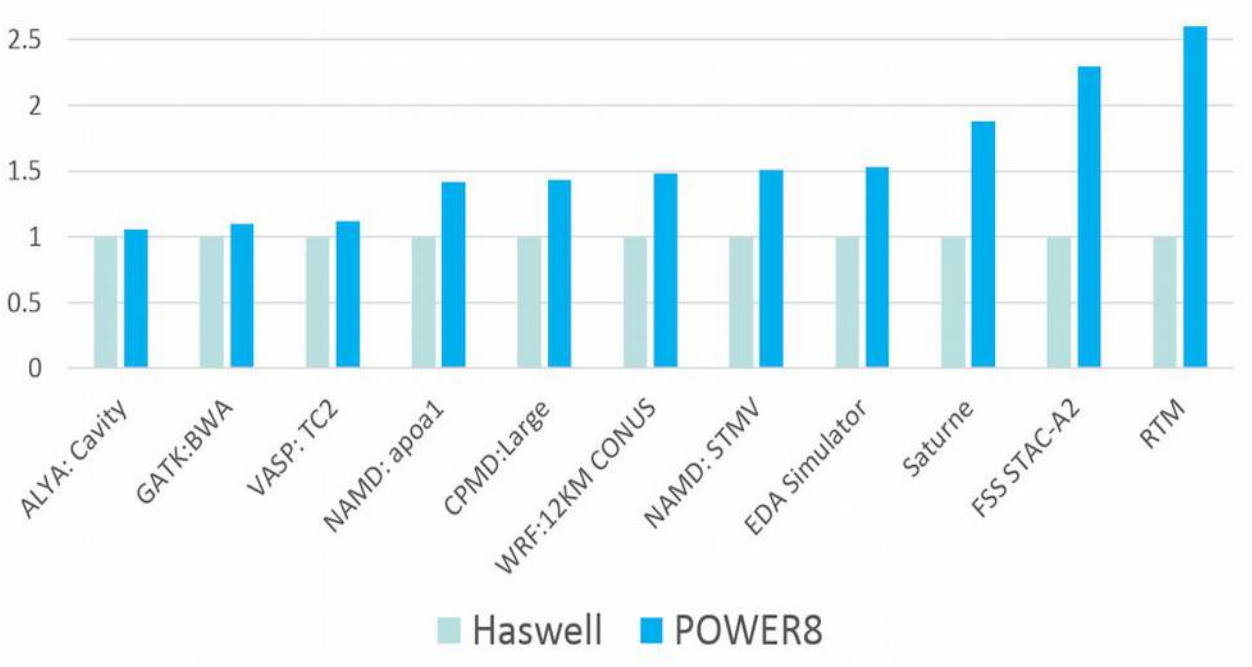

◊ Requires BIOS and firmware update
† Depends on market availability

# Performance vs $/performance



MHPC
Master in High Performance Computing

# Power8 vs Intel Xeon

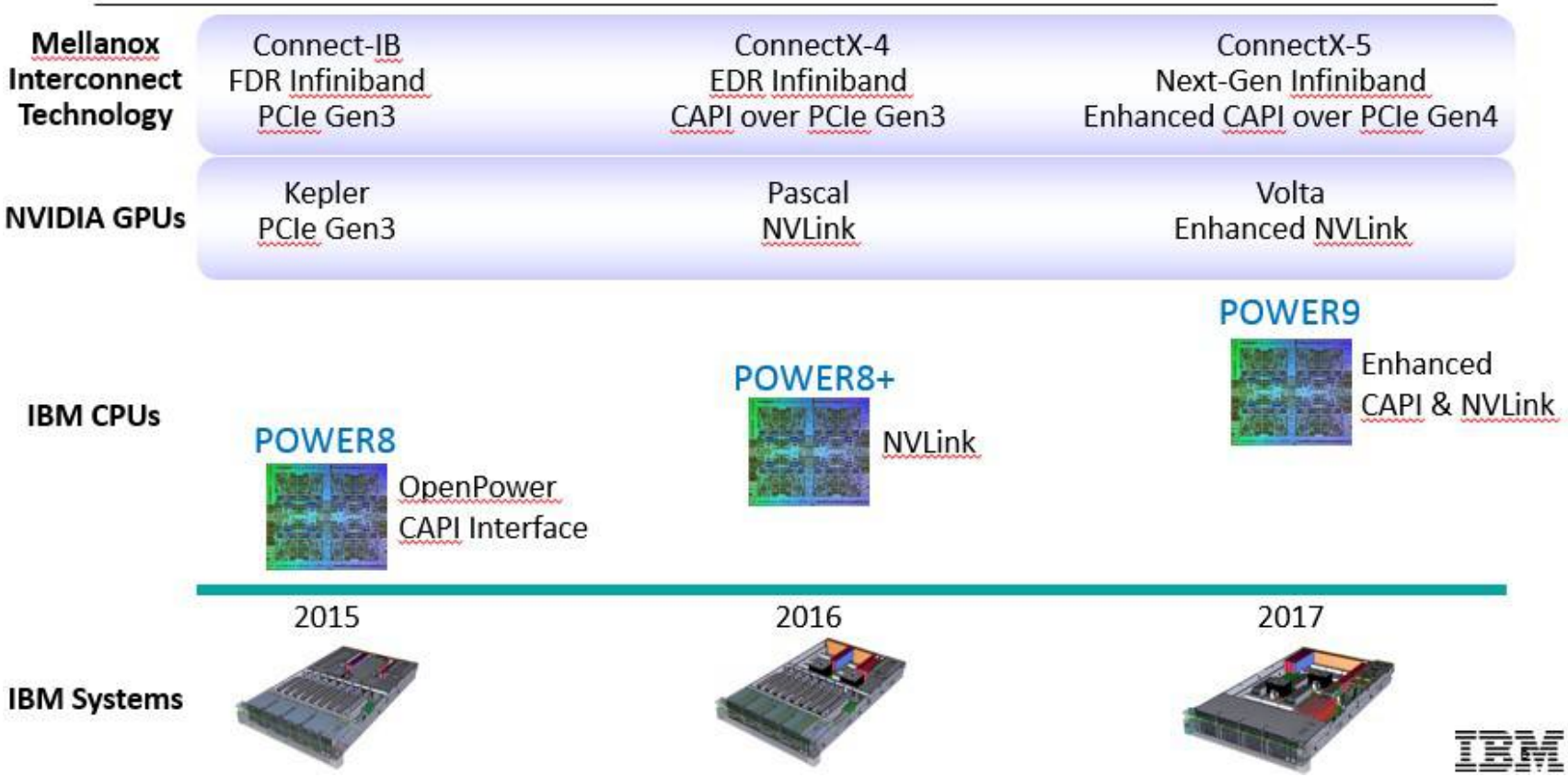



From: http://www.nextplatform.com/2015/07/15/ibm-readies-power8-for-openpower-push/

SISSA
ICTP

MHPC
Master in High Performance Computing

# Power8 roadmap




SISSA

ICTP

MHPC
Master in High Performance Computing

## Conclusions

- Modern CPU have a high degree of parallelism  some time hidden to the user
- In order to optimize on them you should be aware of this.
- Next presentation: some trick of the trade to understand bottleneck and increase performance

SISSA
ICTP