A short introduction into *Hardware*

... in general

... and what do we have at CERN

What's *Hardware* anyway ??

“If it hurts when it falls on your feet ...

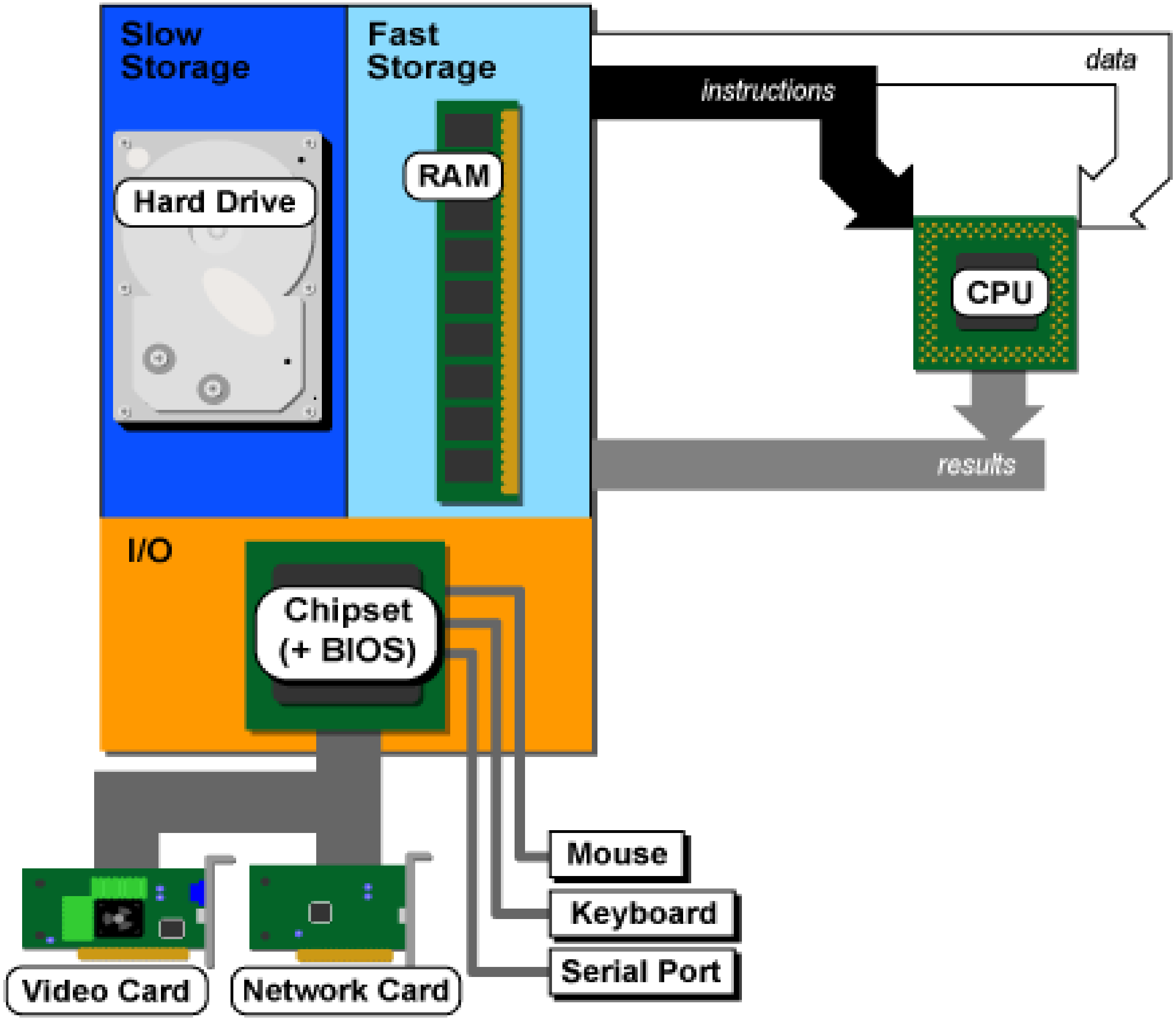
... then it's hardware !”

... well mostly ;-)

What you're about to hear

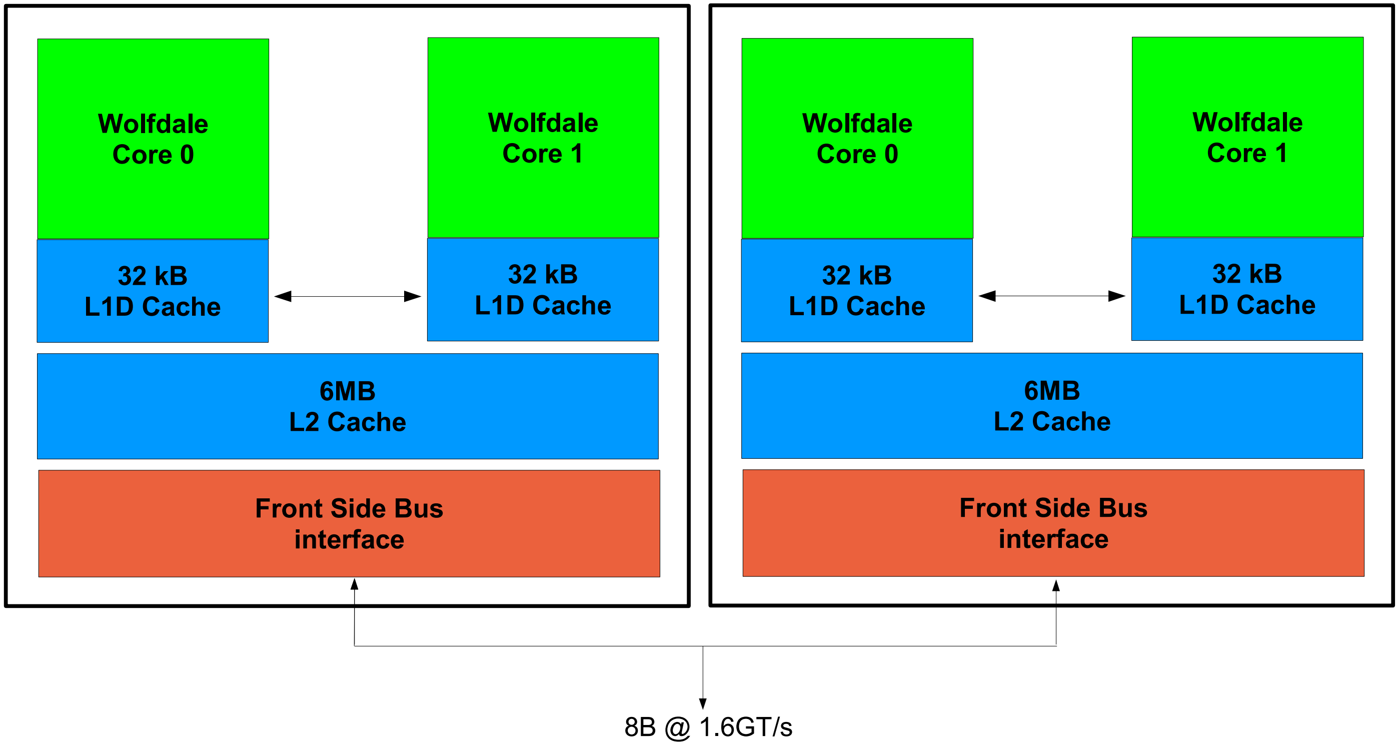
- **In general**
  - CPUs – en detail
  - Storage and I/O (i.e. network)
- **CERN Computer Center**

What's a computer



Intel “*Core2* ” ...

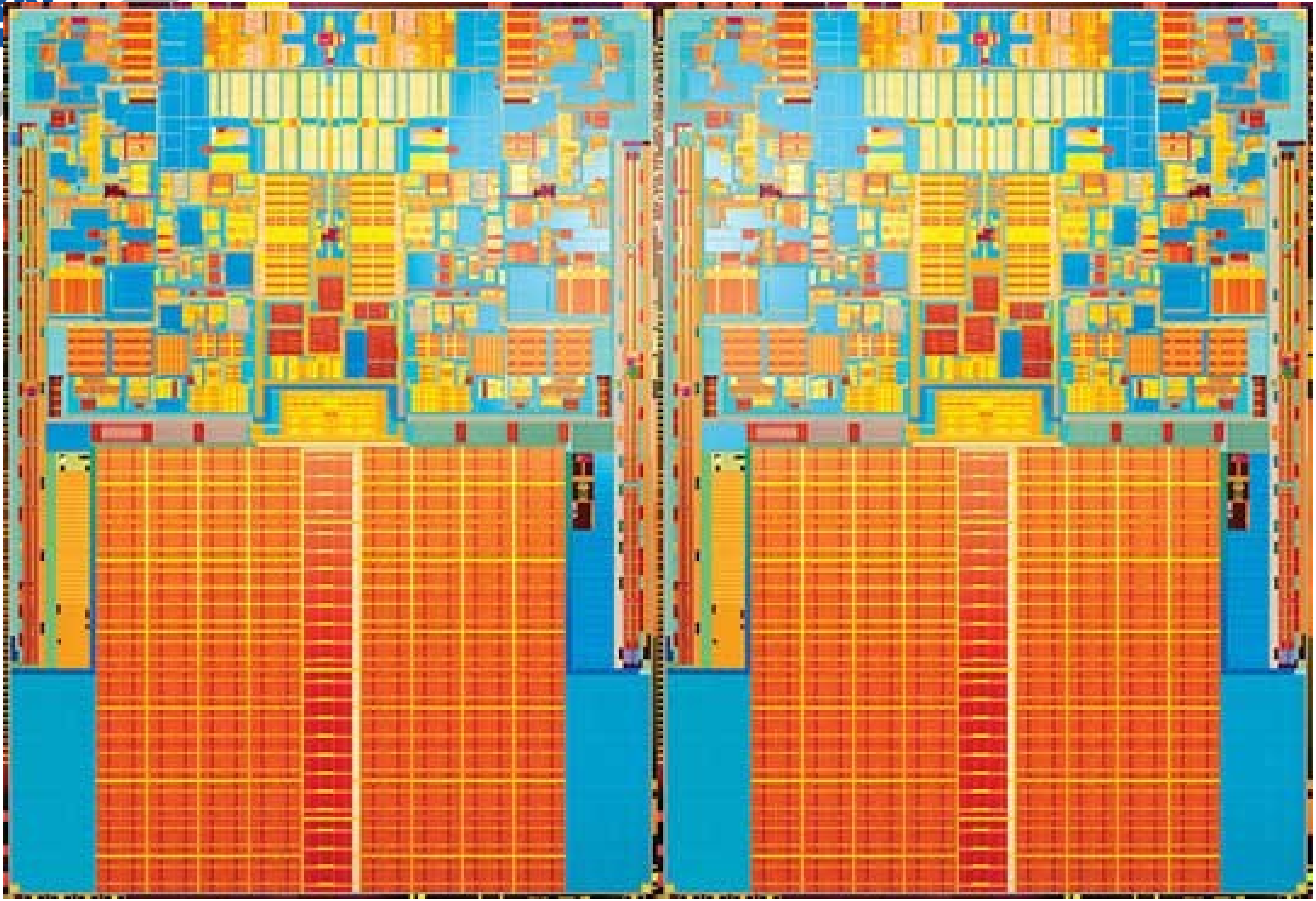
Harpertown



The general layout of a Harpertown “processor”

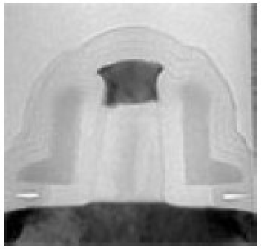
- The L2-Cache is smart enough to hold information which is used by both cores only once!
- Intels current quad-core CPUs (Harpertown/Yorkfield) are basically two dual-cores in one package

… with a picture …



Process technology... (Intel)

90nm
2003

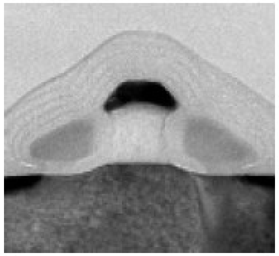


Initially Single Core

A Server 'way' meant 1 a core per socket

Beta on dual cores; Monectito and Dempsey

65nm
2005



Dual Core

A way now = 2cores

Each nm = 30% space

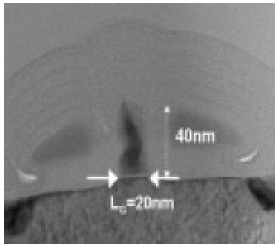
Beta Quad cores

A way = 4cores

Beta Direct Connect

We Are Here

45nm
2007



Quad Cores

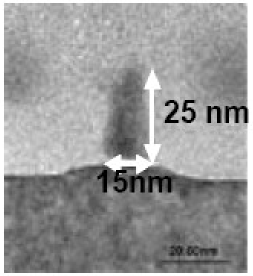
A way = 4cores

Gain 30% area never save power

Cores? Memory? Interconnect? Feature?

Beta Oct Cores?

32nm
2009



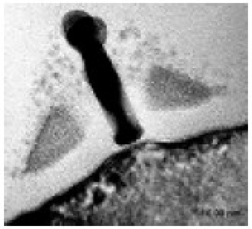
Oct Cores

Beta Multi-Cores

Direct Connect

???

22nm
2011



Shouldn't we be saving Power?

Each Technology shift provides:

- 30% die space
- 10% drop in volts gives 30% power saving cancelled by adding cores

2013 –2017

16nm
2013

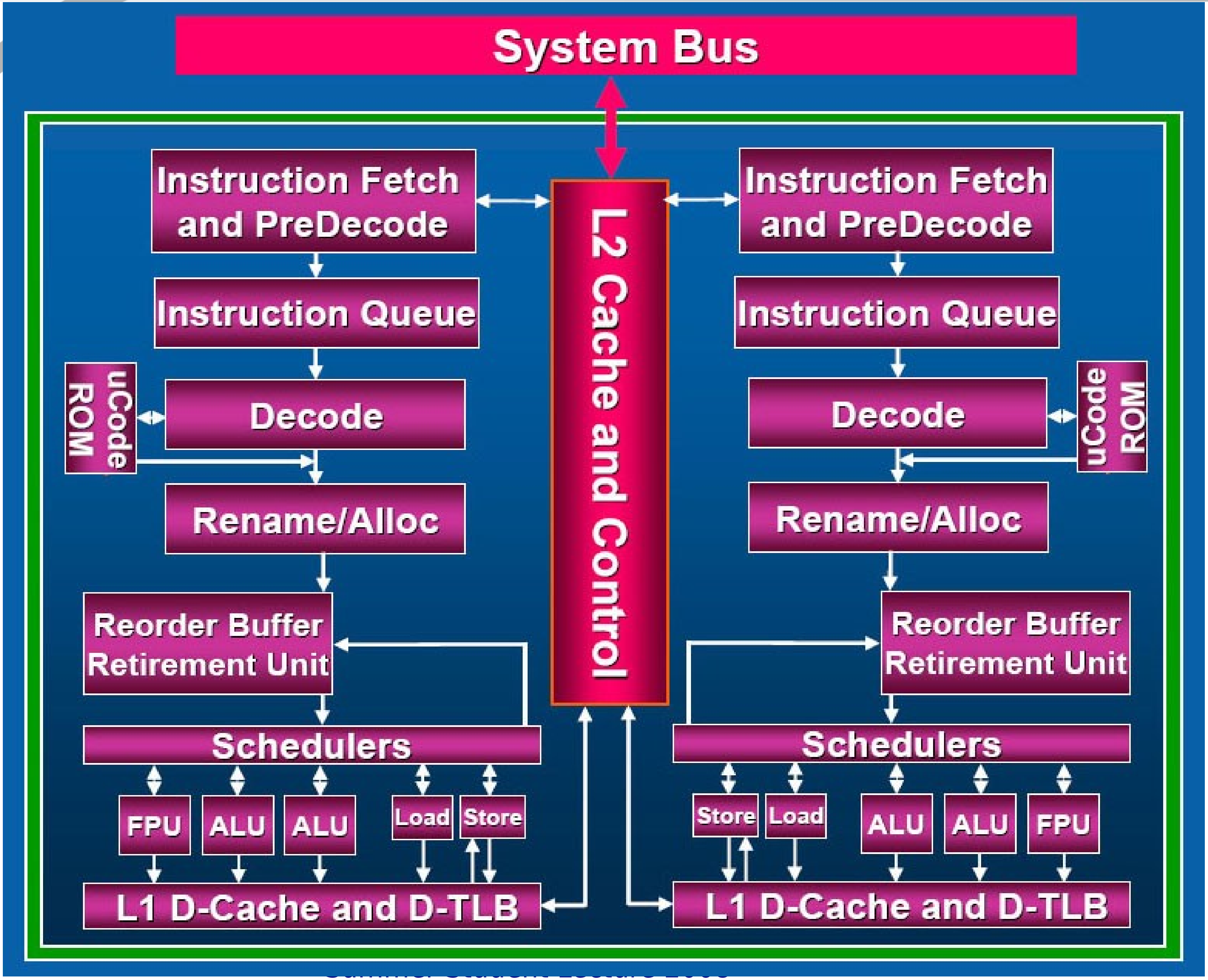
11nm
2015

8nm
2017

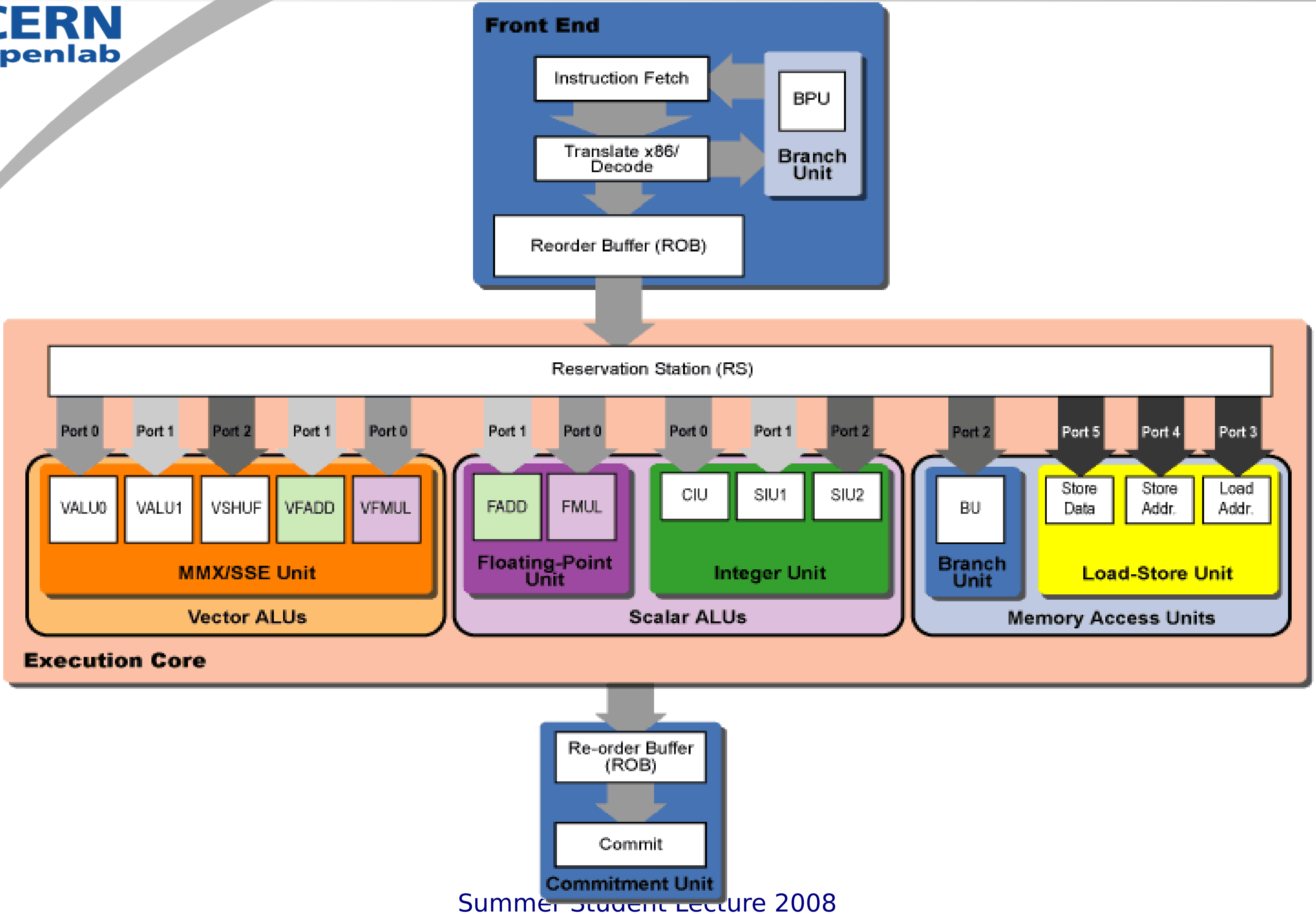
Roadmap

Research

… a schematic view …



… and a deep look inside



Internal “Parallelism” I

- All current processors have several execution ports/pipelines
  - They are hidden from the user
  - Six ports on the Core 2 mircoarchitecture
    - Three ports for load and store unit
    - Three ports for execution units

- The “*SIMD*” - *Single Instruction Multiple Data* units
  - Execute the same instruction on more than one variable
  - *Streaming SIMD Extensions* - “*SSE*” execution units
  - ... 3DNow! ... AltiVec ...
  - ... or “*vector*” units

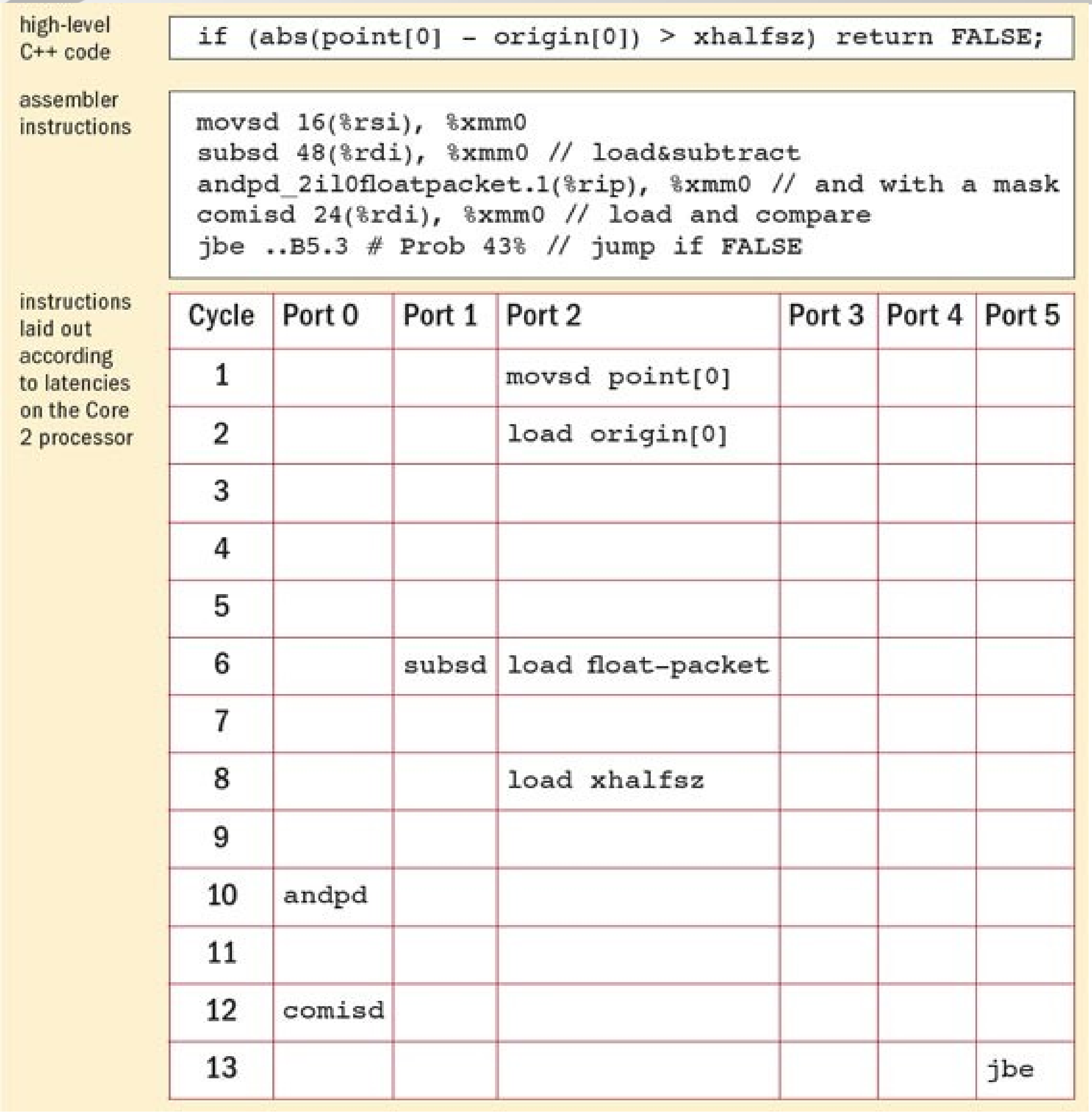
Internal “Parallelism” II

- Utilisation of the pipelines/ports is controlled by the out-of-order engine and the compiler
  - The out-of-order engine can correct a lot of bad coding ... but only so much (and it consumes a lot of power!)
  - The weak point of in-order machines is it's reliance on the compiler (and the programmer !!!!) ...
  - The programmer has to make it possible for the compiler to optimise.

Internal "Parallelism" III

- Utilization of the *SSE* units is controlled by the compiler
  - If the compiler knows about them, he'll use them automatically
  - "gcc -O2" on 32-bit OS does **NOT** use them
    - The compiler assumes that there are no additional instructions available compared to a Pentium...
  - "gcc -O2" on 64-bit OS **DOES** use them
    - The compiler knows that the *SIMD* units are present on the 64-bit capable processors (AMD and Intel)
  - At the moment gcc uses only one variable. Using the max. possible number of variables at the same time will come in later versions.
  - Other compilers, e.g. Intel icc, make use of as many variables as possible.

How does HEP code look like …



high-level C++ code

```
if (abs(point[0] - origin[0]) > xhalfsz) return FALSE;
```

assembler instructions

```
movsd 16(%rsi), %xmm0
subsd 48(%rdi), %xmm0 // load&subtract
andpd_2il0floatpacket.1(%rip), %xmm0 // and with a mask
comisd 24(%rdi), %xmm0 // load and compare
jbe ..B5.3 # Prob 43% // jump if FALSE
```

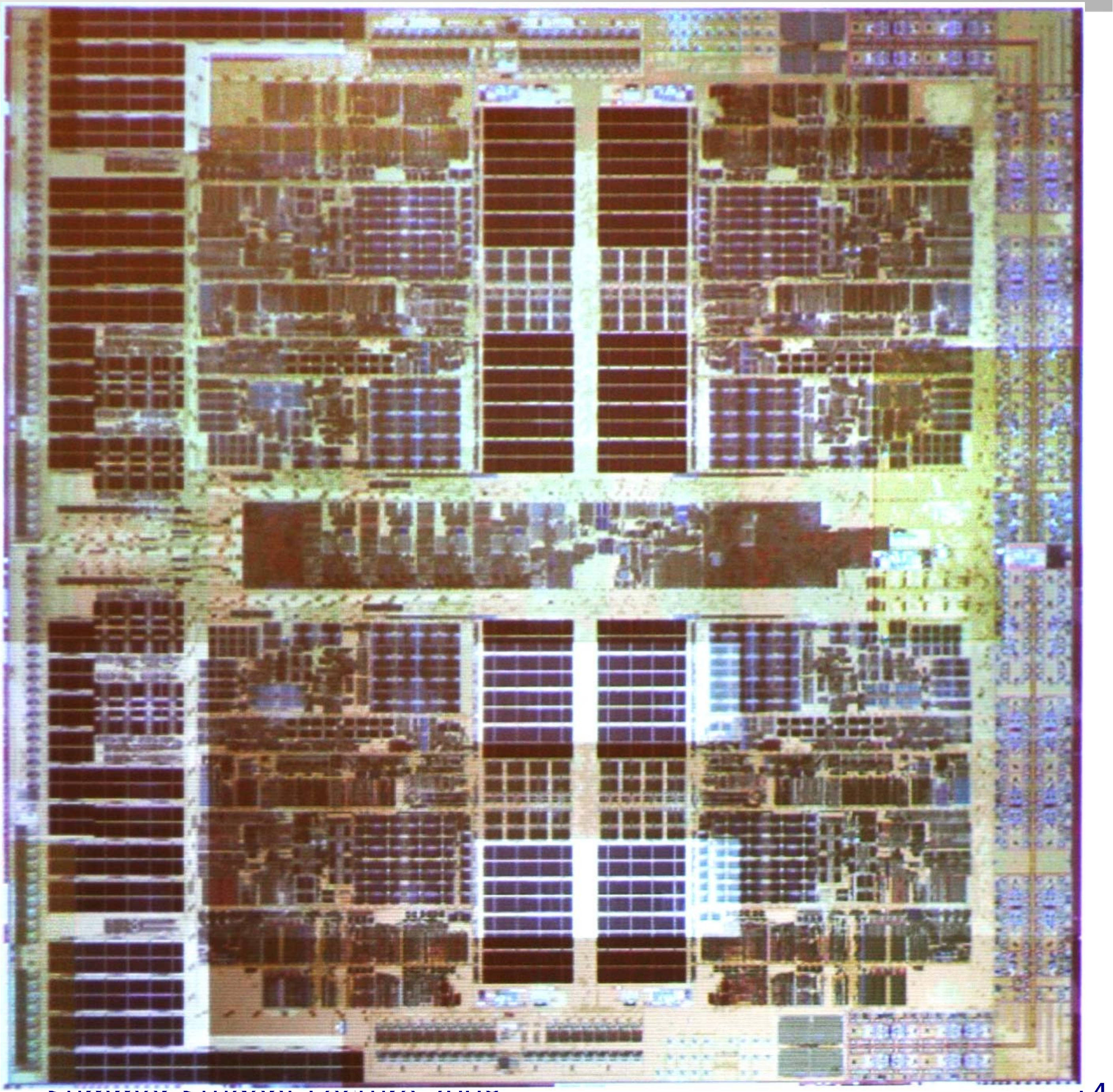
instructions laid out according to latencies on the Core 2 processor

Cycle	Port 0	Port 1	Port 2	Port 3	Port 4	Port 5
1			movsd point[0]			
2			load origin[0]			
3						
4						
5						
6		subsd	load float-packet			
7						
8			load xhalfsz			
9						
10	andpd					
11						
12	comisd					
13						jbe

Taken from a CERN Courier article (April 2007 issue) by Sverre Jarp

Other Multicore CPUs – AMD Barcelona

- AMDs next Gen CPU
  - Four general purpose cores on single die
  - Somewhat delayed



Intel64/AMD64 CPU comparison

Harpertown

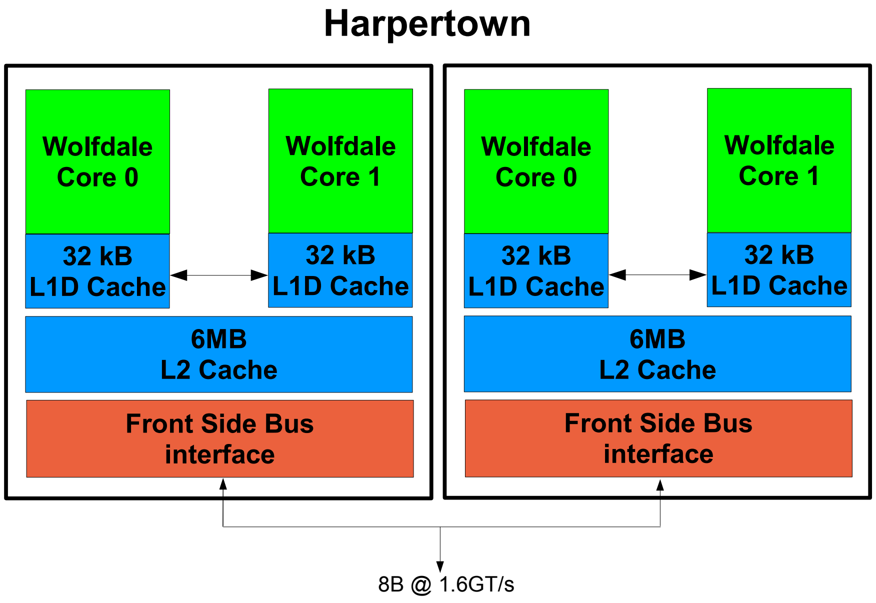
- Intels current server CPU
- 4-cores in a 2x2 package
- Connection to memory and other CPUs via Front Side Bus

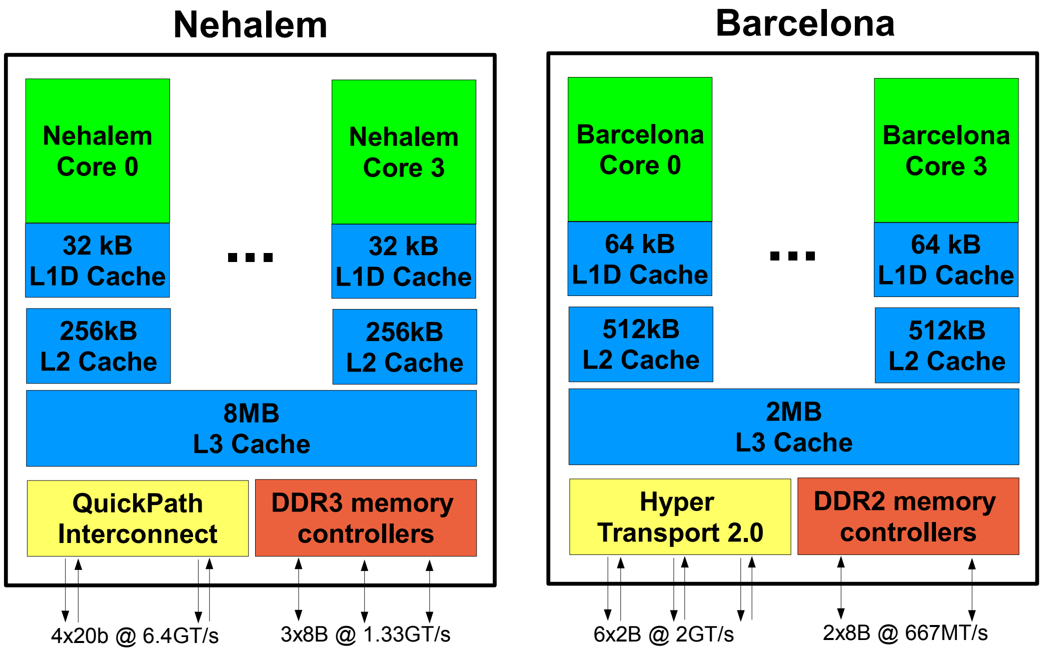
Barcelona

- AMDs “current” server CPU (serious delays)
- 4-cores on single die
- Direct connection to DDR2 memory via built-in controllers
- Connection to other CPUs and other I/O via HyperTransport 2.0

Nehalem

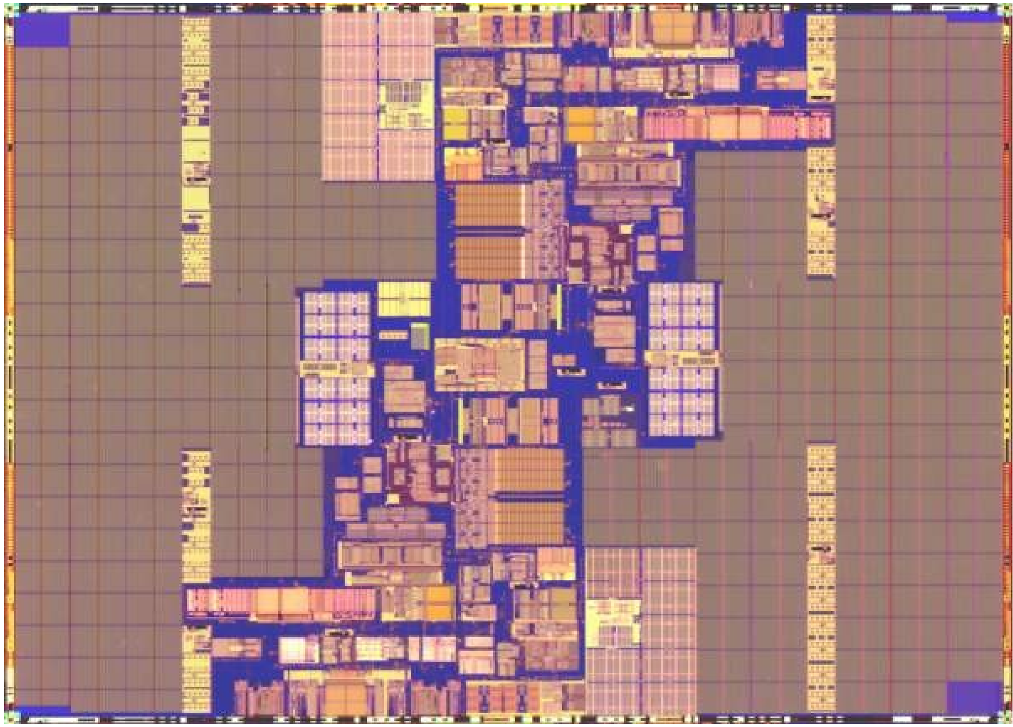
- Intels next-gen server CPU
- 4-cores on single die
- Direct connection to DDR3 memory via built-in controllers
- Connection to other CPUs and other I/O via QuickPath Interconnect

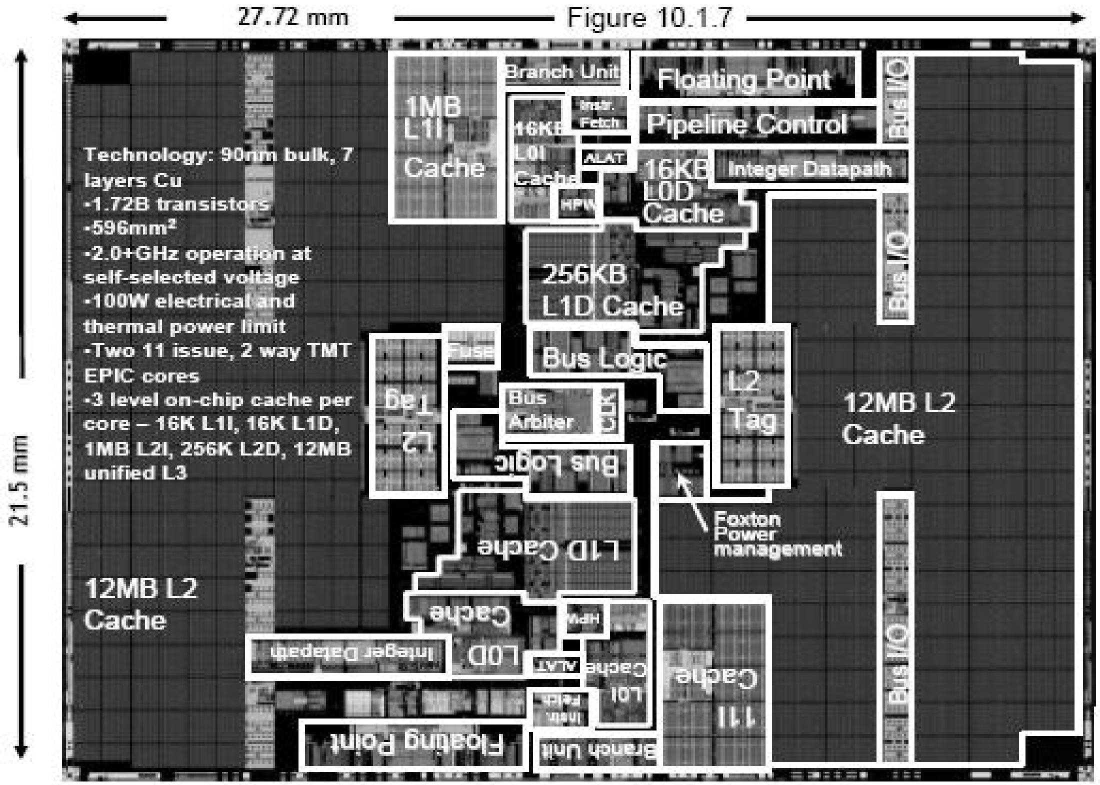




Other Multicore CPUs - Montecito

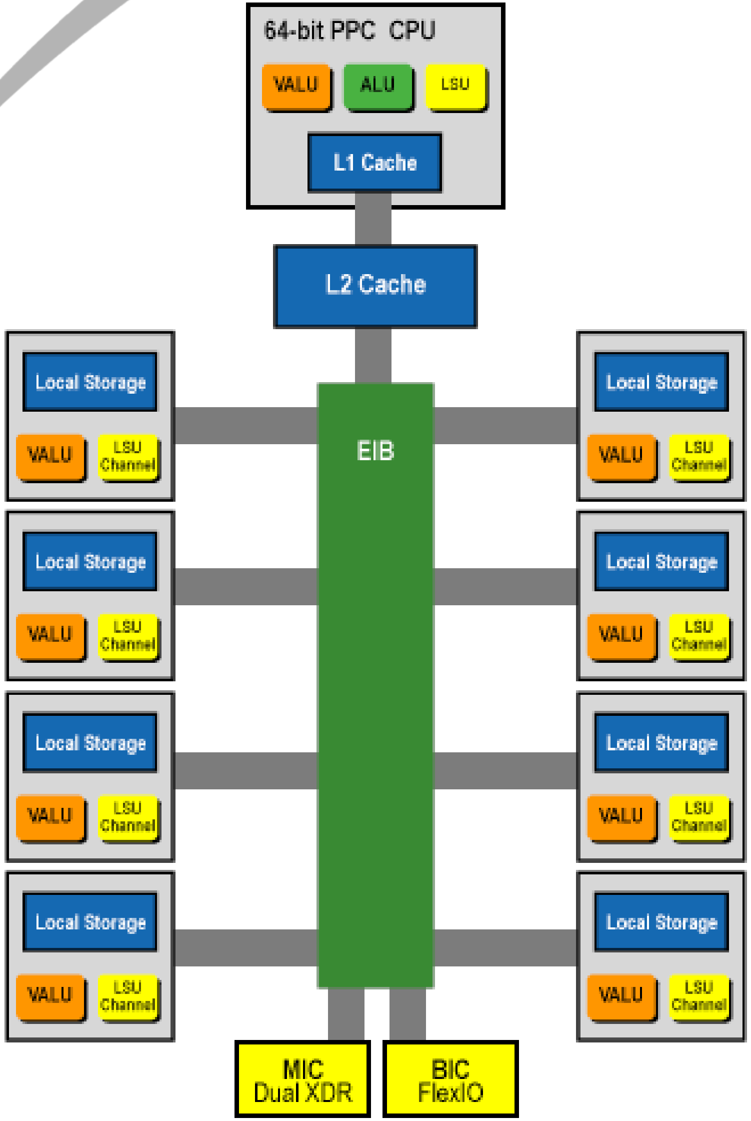
- Itanium processor family
- Dual-core design
  - 1.72 billion transistors
    - ~57M for core logic
    - ~107M L1/L2 Caches
    - ~1550M L3 Cache
    - ~7M bus and I/O logic



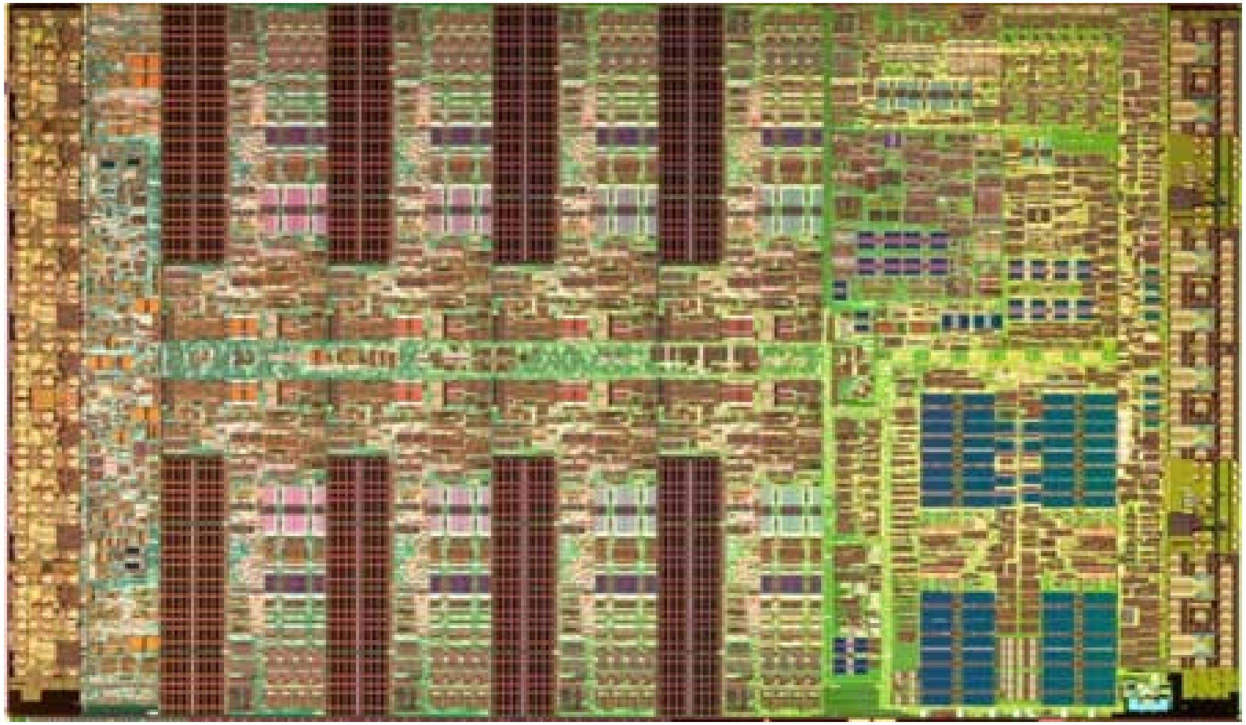


Technology: 90nm bulk, 7 layers Cu
•1.72B transistors
•596mm²
•2.0+GHz operation at self-selected voltage
•100W electrical and thermal power limit
•Two 11 issue, 2 way TMT EPIC cores
•3 level on-chip cache per core – 16K L1I, 16K L1D, 1MB L2I, 256K L2D, 12MB unified L3

Other Multicore CPUs – IBM Cell



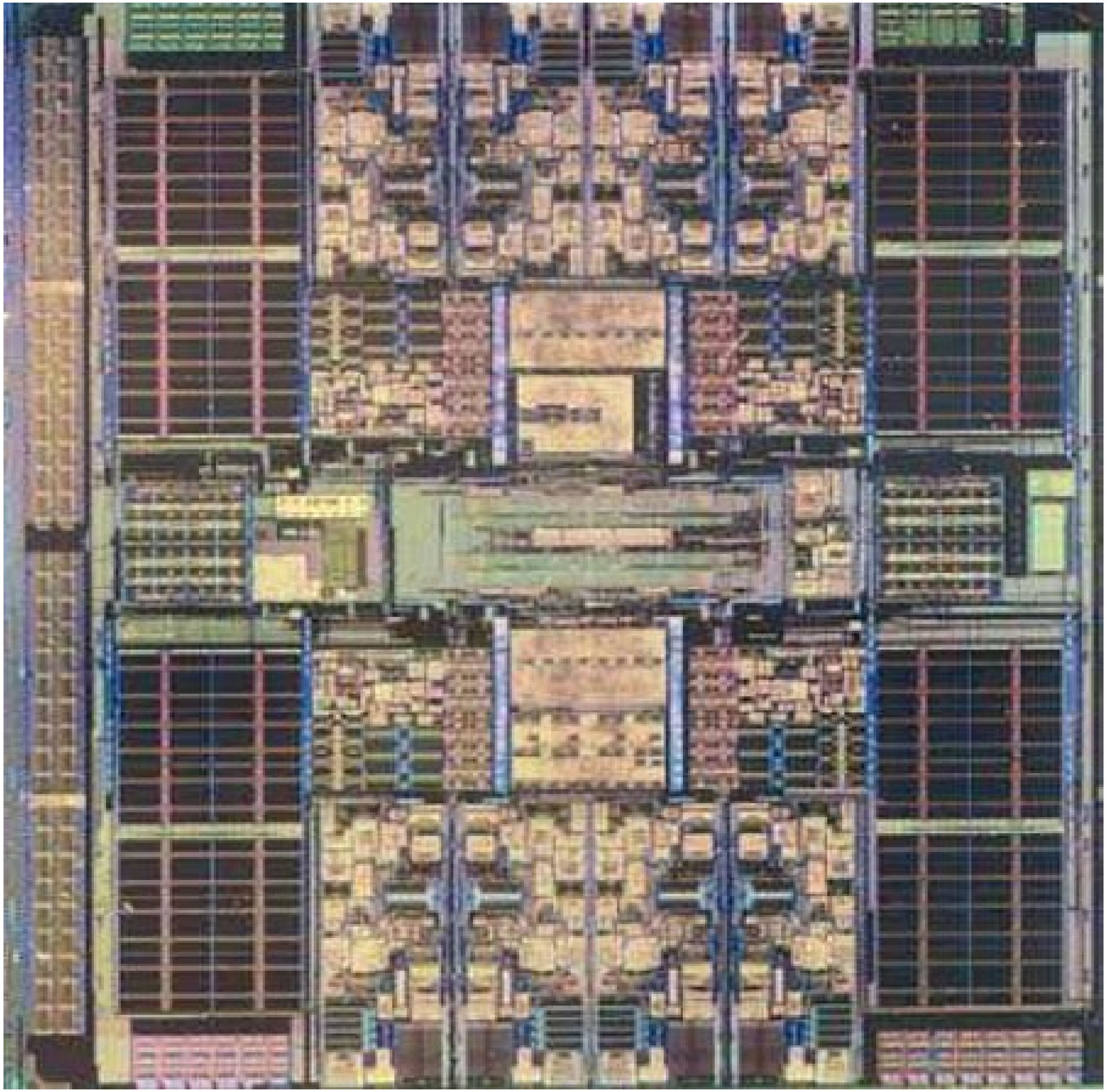
The CELL Architecture



- One general purpose POWERPC processing unit
- Eight special purpose DSP-cores: “synergistic processing units” (SPUs)
- Very high theoretical performance
- Real life efficiency still not good...
- Used in specialized environments
  - Game consoles
  - Special purpose computers

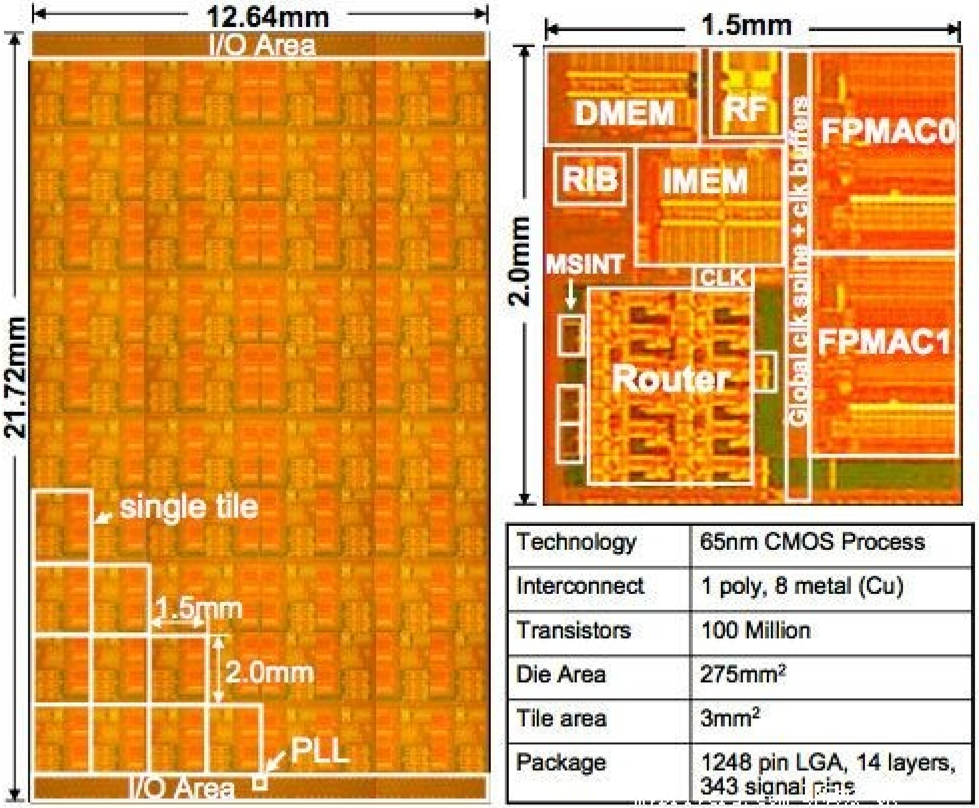
Other Multicores – SUN Niagara

- 8 cores (UltraSPARC)
  - Single port
  - In-order
  - 4 threads per core

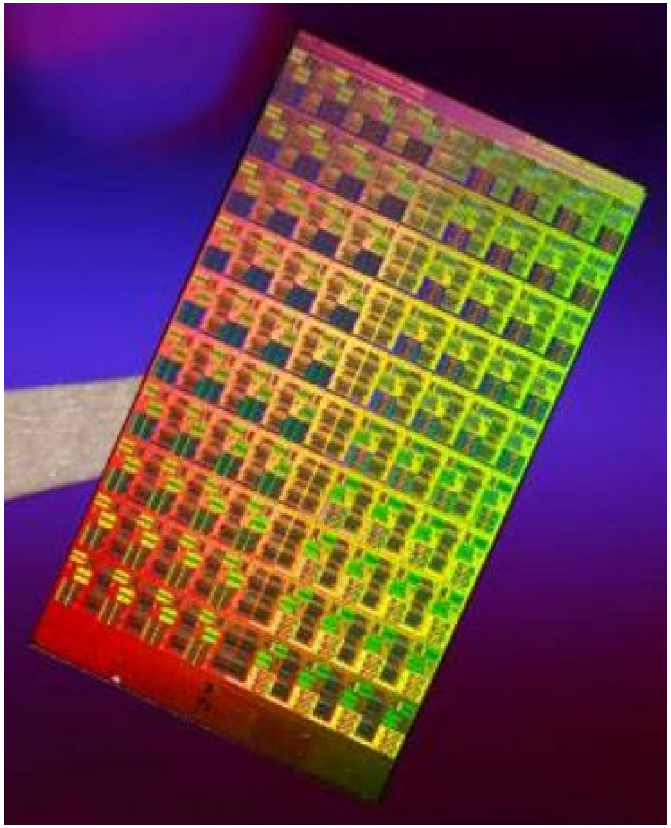


… a glimpse of the future …

Intel "*Polaris*" 80-core technology demonstrator



Technology	65nm CMOS Process
Interconnect	1 poly, 8 metal (Cu)
Transistors	100 Million
Die Area	275mm^2
Tile area	3mm^2
Package	1248 pin LGA, 14 layers, 343 signal pins



In the future we'll see large number of cores

- Moderate number of "fast" cores
  - Basically continuing the current scheme
- Large number of "slower" cores
  - 16 or even 32 simpler cores possible in the near future (2-3 years)
  - In the more distant future 128++ cores ??

Comment on GPUs

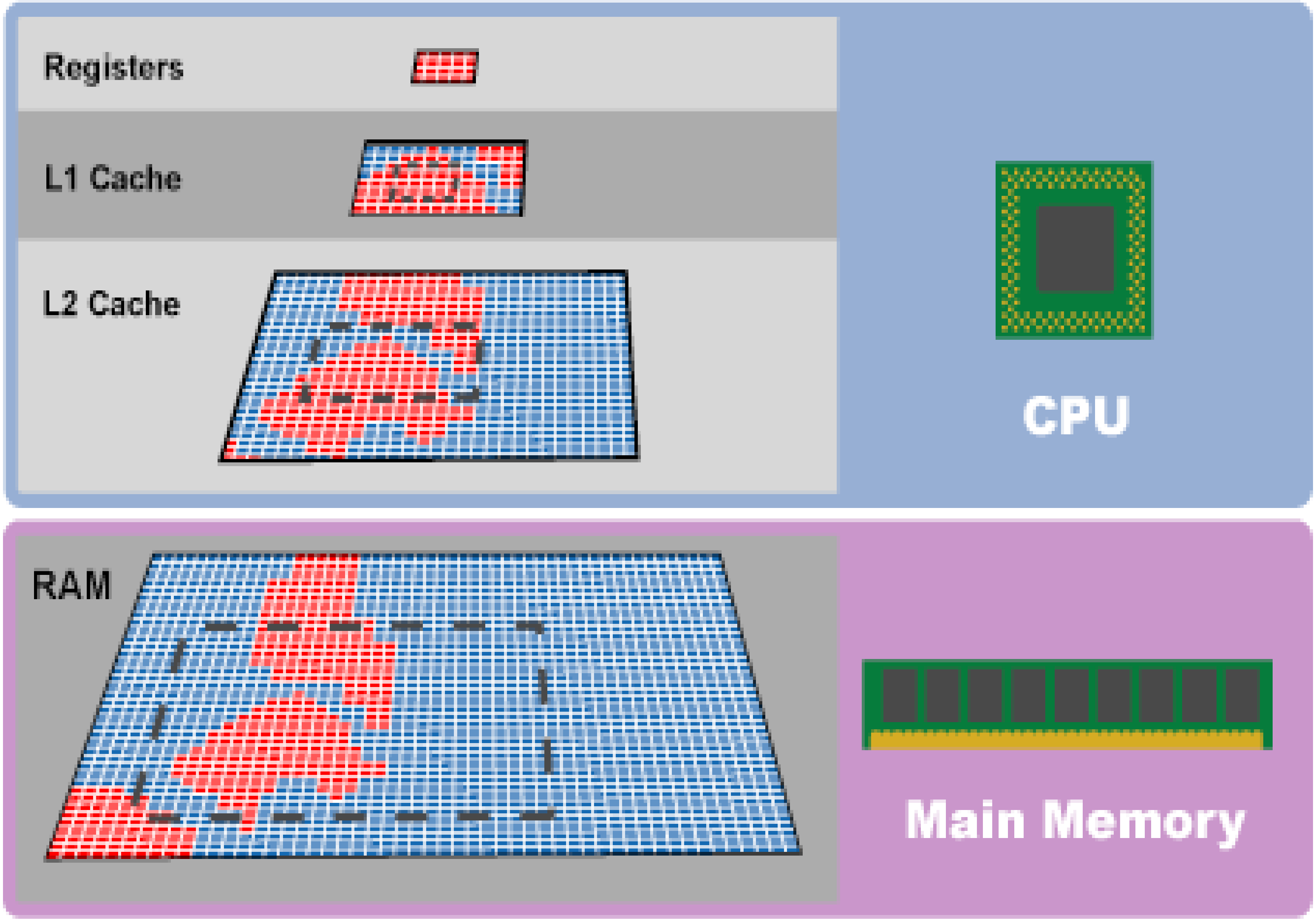
- Most recent Graphics Processing Units (GPUs) can be used for “normal” computing tasks
- GPUs are highly parallel processors and work best with streaming type of computation
  - Code should be highly parallel, i.e. threaded (HEP code isn't)
  - Code should have only a small number of branches, i.e. *if* or *switch* statements (HEP code is flooded with them)
  - Memory requirements per job have to be very small (HEP jobs take 2GB or more)
- Current GPUs implement only single precision FP
  - HEP (and others) need double precision
- Floating point format/implementation on current GPUs is not IEEE compliant (!!!) ... would you trust the results?

Remarks on multi-core CPUs

- The only way to have the per-socket performance keep increasing in the future (“Moore's Law”)
- The performance of a single core will not increase as much as it used to in the past ... it might even decrease ...

  - ➔ Performance gain mainly through multi-core/parallelism
  - ➔ Serious implications on software design
    - ➔ keyword: Multi-threading
  - ➔ Very fast connection to main memory is crucial
  - ➔ A lot of memory required! (2GB per core at the moment)
    - ➔ Multi-threaded programs would ease this requirement significantly

The memory hierarchy



Memory – some numbers

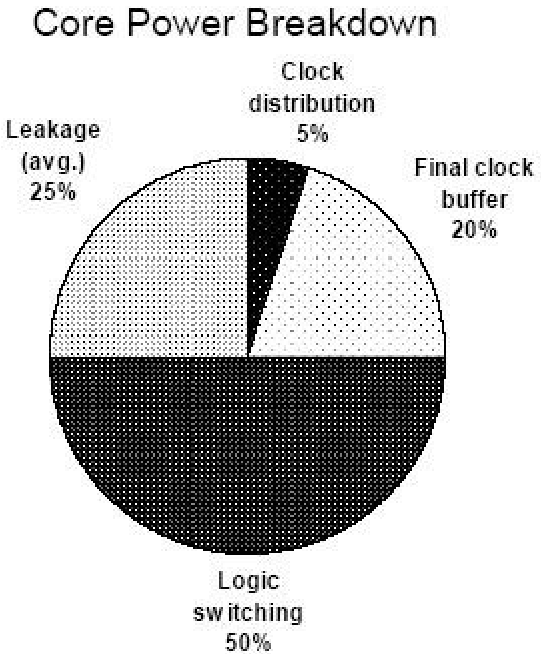
Memory Type	Access Time	Typical Size	Technology	Managed by
Registers	1 cycle	1 kB	Same as CPU	Compiler
Level 1 Cache	2 – 4 cycles	8 – 64 kB	SRAM	Hardware/compiler
Level 2 Cache	5 – 20 cycles	256 – 4096 kB	SRAM	Hardware/compiler
Level 3 Cache	15 – 50 cycles	0 – 24 MB	SRAM	Hardware/compiler
Main Memory	130 – 500 cycles	1 – 64 GB	DRAM	OS/user
Hard disk	~$2*10^7$ cycles	160 – 1000 GB	Magnetic	OS/user

- getting data from main memory takes very long
  - … and the CPU is sitting around just converting power to heat …
  - try to “prefetch” data into the cache (usually L2-Cache)
  - Methods like Simultaneous Multi-Threading try to make use of those wait cycles

If you start from a “worst case” scenario – always go to main memory (instead of L2 cache) – prefetching alone could speed up your application by a factor of 20 … but then, this scenario never occurs these days

Power Consumption

- The “Thermal Design Power” (TDP) of CPUs is between 65W and 130W
- The biggest showstopper for frequency scaling!
- ~ 25% of the consumption is caused by leakage currents!!
  - New technologies help to reduce those...
- Power consumption of memory becomes important
  - ~15W for a 2GB DIMM under load for FB-DIMMs (~10W for DDR2)
  - in a Harpertown system (two quad-core CPUs) with 16GB RAM the memory consumes almost as much power as the CPUs...



Storage and I/O

- Storage
  - Disk : up to several PB (PetaByte)
  - Tape : much more than disks (factor 10 - ...)
- I/O – concentrate on networking
  - Ethernet
    - LAN: 1Gb, 10Gb
    - WAN: 10Gb

Disk Storage

- Most common storage type (every PC has a disk...)
- basically two technologies
  - SATA (I/II) used in PCs and “cheap” servers
    - 1 – 48 disks
    - usually 4-8 ports on motherboards
    - up to 24 ports on special add-on cards
  - SCSI/SAS used in high end servers
    - 1 – “take-your-favourite-number” disks
    - SAS disks are different from SATA disks!!!
- Solid State Drives look promising, but still too small and too expensive for CERN

SATA: Serial Advanced Technology Attachment; SCSI: Small Computer System Interface; SAS: Serial Attached SCSI

Disk Storage – II

- Disks are inherently “unsafe”
  - Failure rates are relatively high
  - Data recovery after a crash difficult or impossible
- In order to increase overall reliability disks are organised in RAID systems
  - different RAID levels provides different levels of redundancy and performance.
  - RAID Level 0, 1, 5 or 6 are most commonly used
  - Combinations possibles, e.g. RAID 50
  - have a look at http://en.wikipedia.org/wiki/RAID for details
  - At CERN h/w RAID6 is used on all diskservers

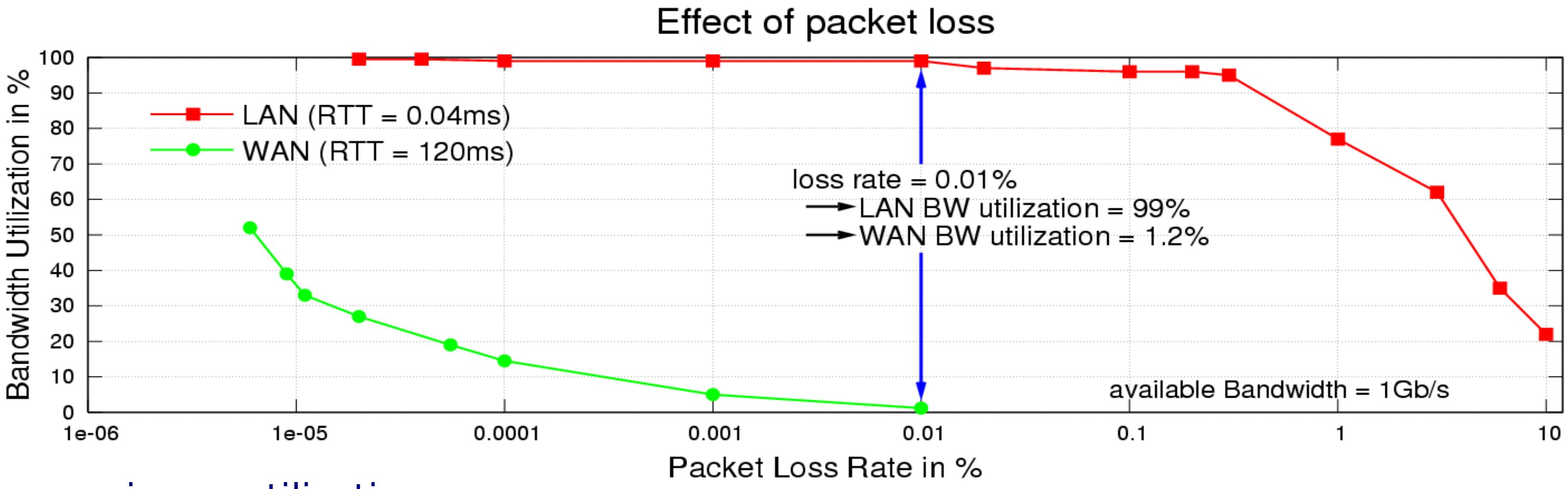
RAID: Redundant Array of Independent/Inexpensive Disks

Networking

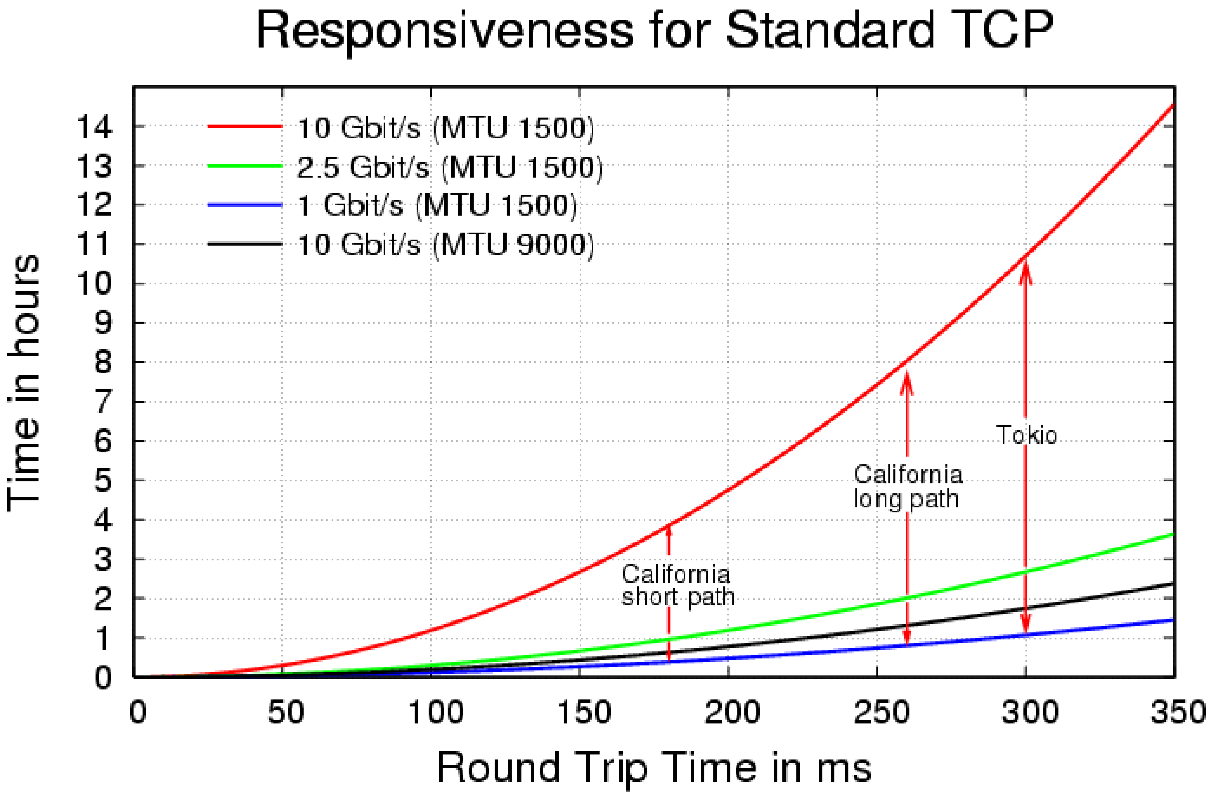
- Ethernet – based on TCP/IP
  - LAN – Local Area Network
    - 1Gb/s links to the hosts
    - 10Gb/s backbone infrastructure
    - 10Gb/s connection to disk- and tape-servers is a possibility in the near future
  - WAN – Wide Area Network (≅ Internet)
    - 10Gb/s going into production
    - 40Gb/s in (very) early tests

For a more detailed look: my summer student lecture 2004

LAN vs. WAN



In order to achieve maximum utilization on a WAN link (e.g. Chicago/Fermilab) with a single stream the packet loss rate must be less than 1 packet in 1 billion!



Responsiveness:
essentially the time to recover from a packet loss...

Further reading

- Quite a lot of in depth information can be found at arstechnica.com (in fact, I did “borrow” some of my graphics there)

  http://arstechnica.com/articles/paedia/cpu/core.ars

  http://arstechnica.com/articles/paedia/cpu/caching.ars

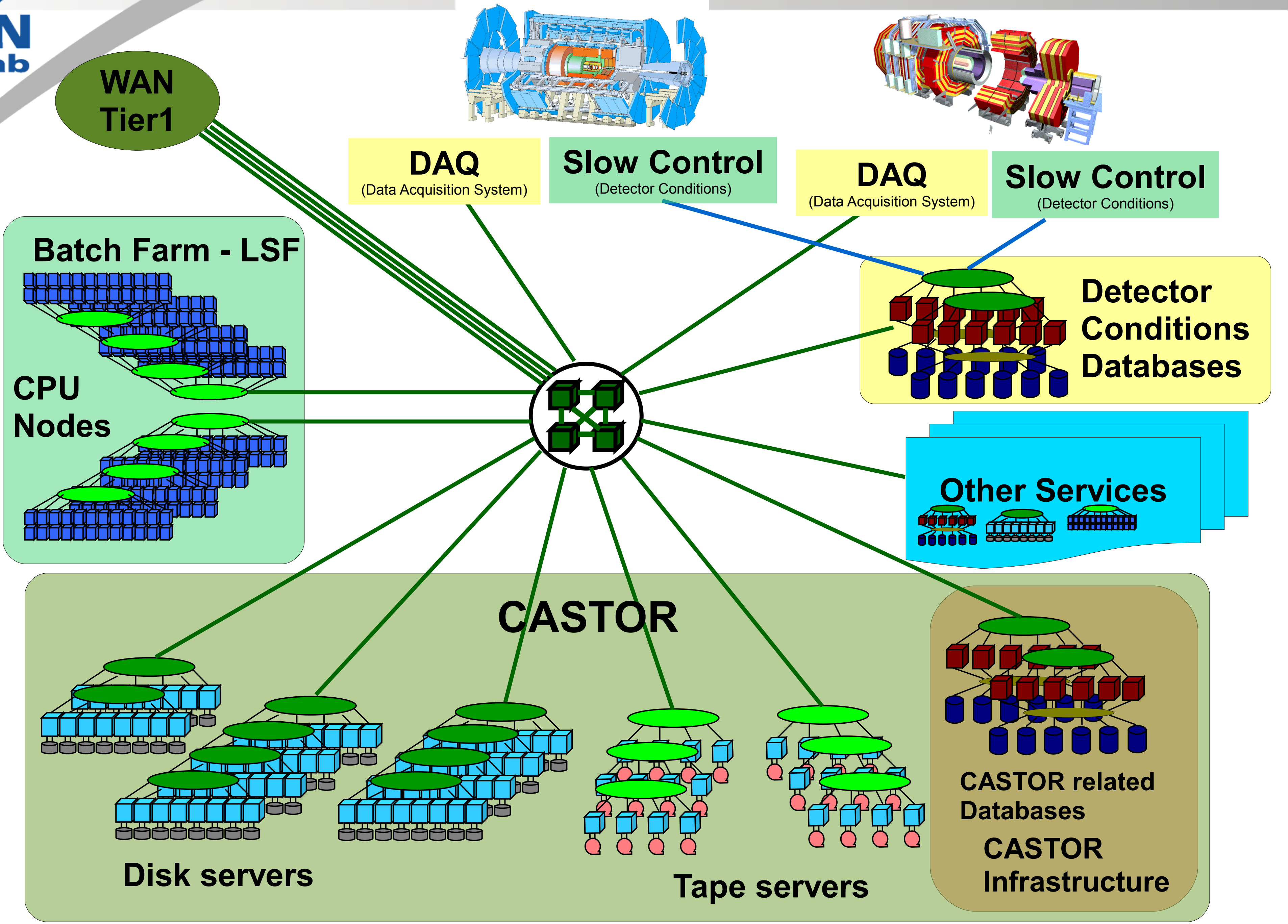
- A comparison of Intel and AMD processors

  http://www.anandtech.com/cpuchipsets/showdoc.aspx?i=2748&p=2

If you want to know “Core (2)” in almost every detail:

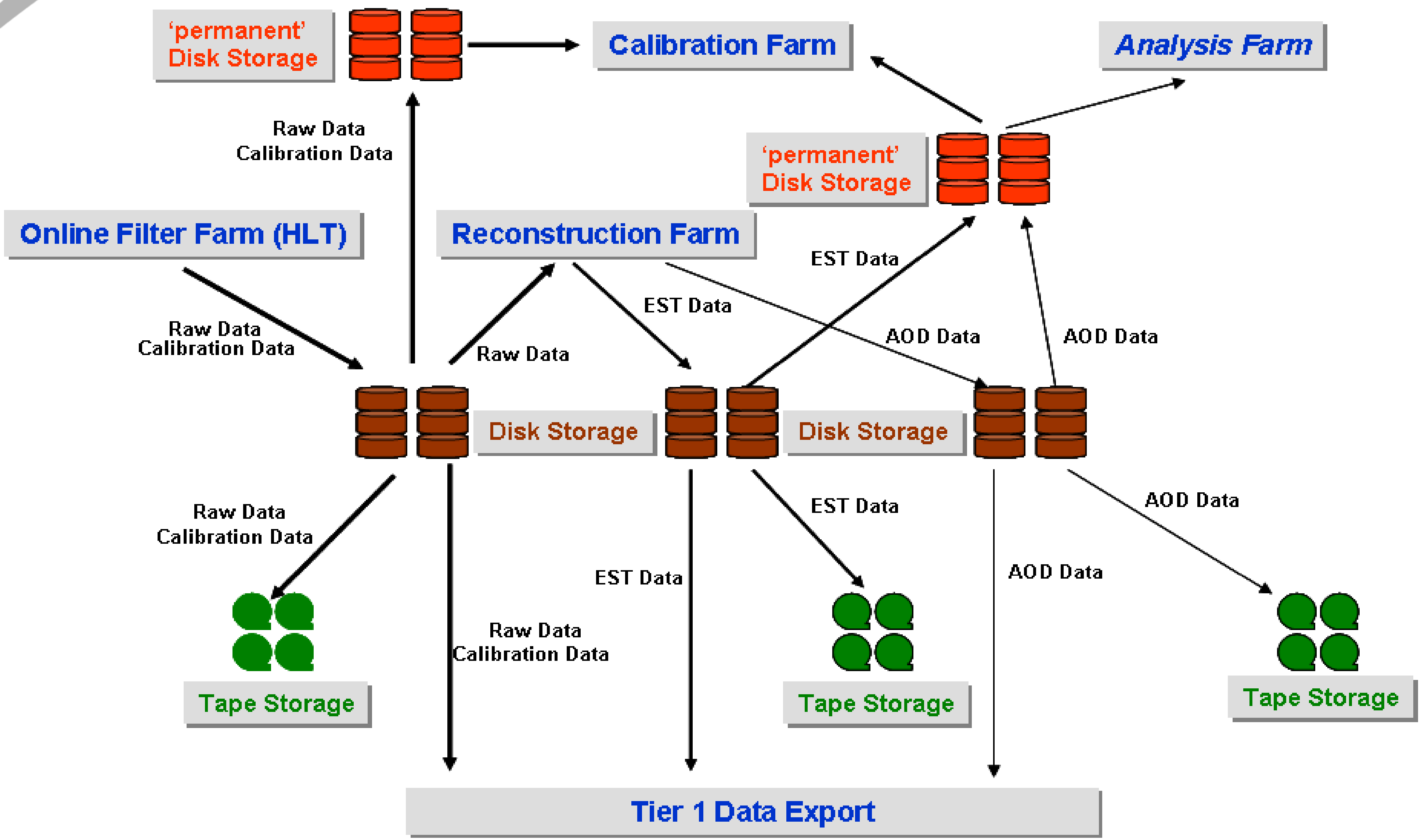
http://www.behardware.com/articles/623-1/intel-core-2-duo-test.html

LHC data flow schematics



LHC data processing schematics

Dataflow T0, CDR + Processing + Calibration



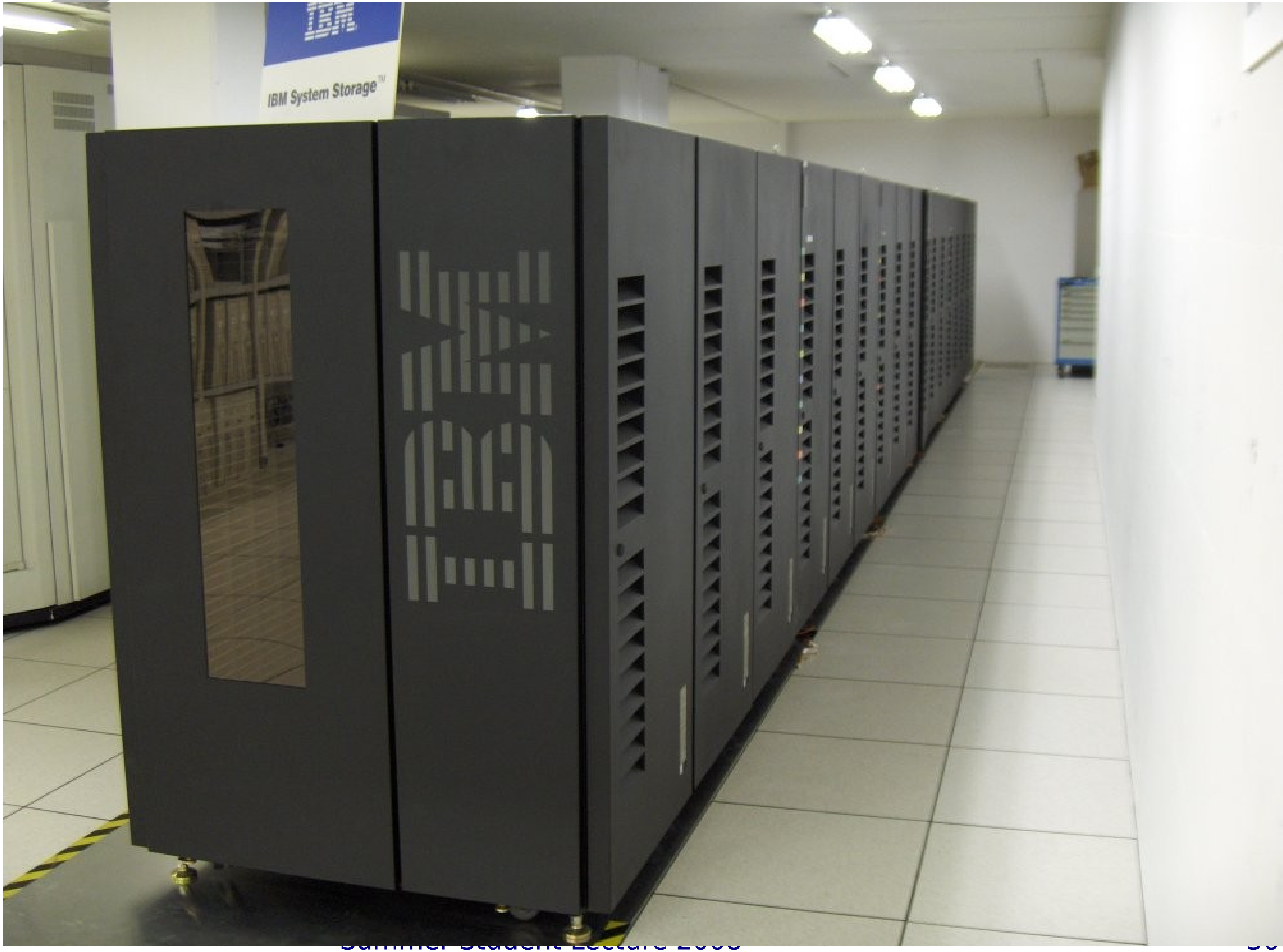
CERN CC in numbers

- Current Physics Computing
  - ~4400 dual-socket compute-nodes (~15000 cores) (almost entirely dual- and quad-core CPUs)
  - ~6.5PB usable diskspace (~9.3PB “raw” space)
  - Storagetek and IBM Tape robots
    - ~5PB robot at the moment
- ... in the (near) future
  - ~2000 dual-socket (?) compute-nodes (more cores per socket!!!)
    - 8-core CPUs soon
  - ~15PB usable diskspace & ~30 – 50PB tapespace
  - Some disk- and tapeservers might be connected via 10Gb
  - **CPU and disk limited by the 2.5MW available for the CC**

The CPU nodes

The Diskservers

The IBM tape silo



The new Storagetek tape silo(s)