NYU
New York University

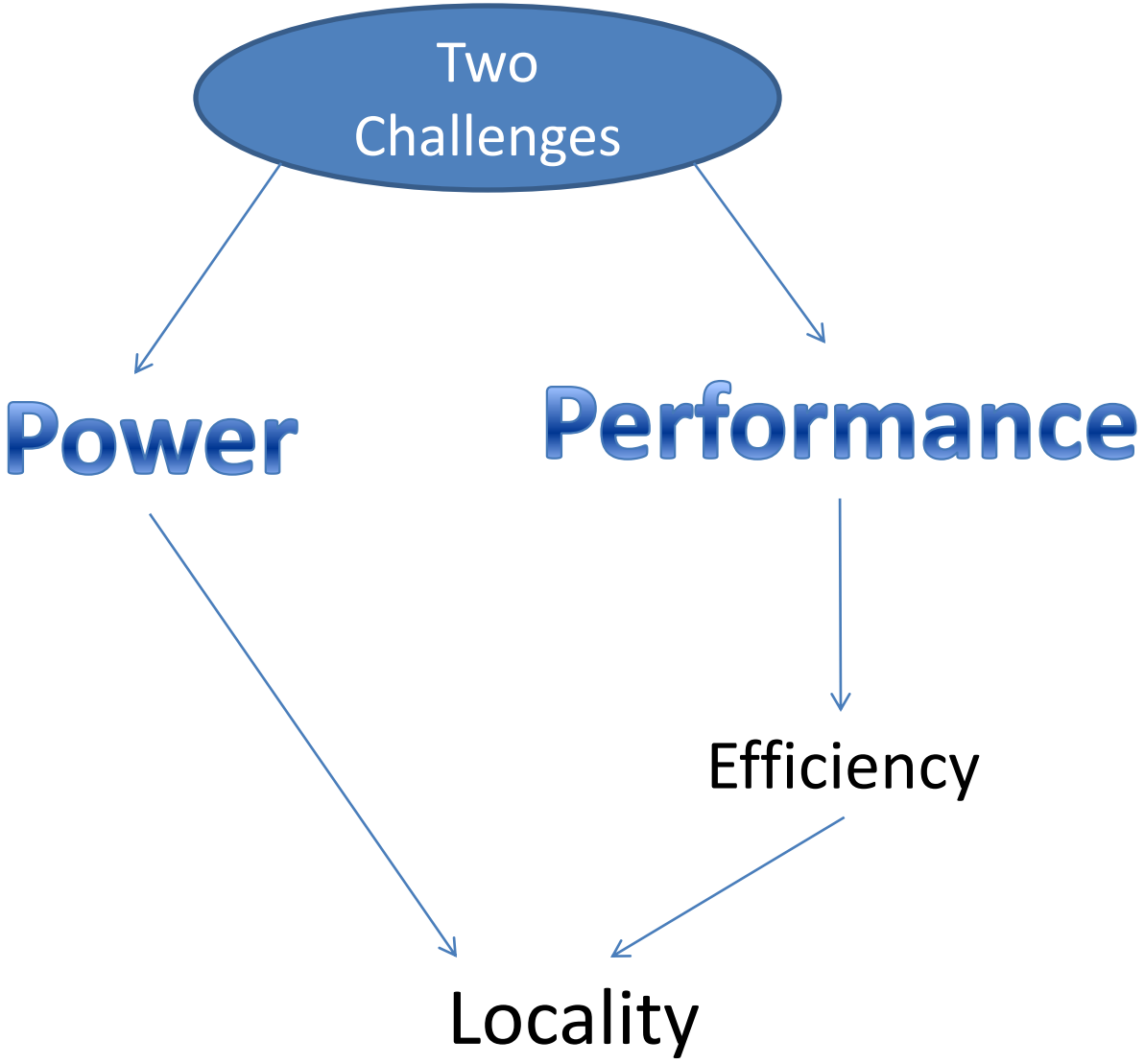
CSCI-GA.3033-012
Multicore Processors: Architecture & Programming

Lecture 4: Know Your Hardware

Mohamed Zahran (aka Z)
mzahran@cs.nyu.edu
http://www.mzahran.com

The Purpose of This Lecture

- Get to know the design of some state-of-the art processors
- Think about ways to exploit this hardware in your programs
- Compare how your program will look like if you did not know about the hardware

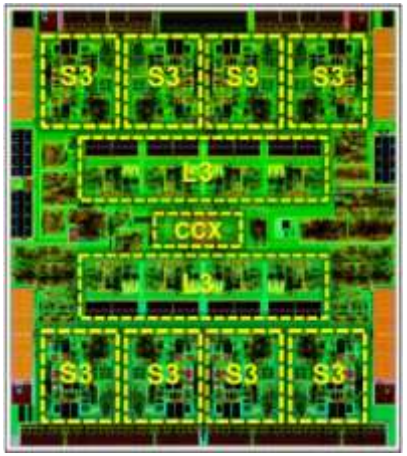


Data movement costs more than computation.

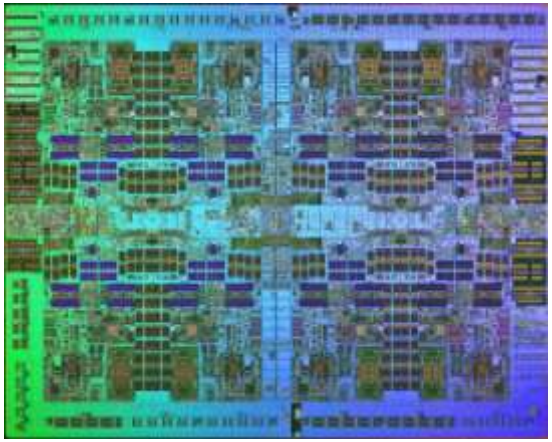
Your Parallel Program

- Threads
  - Granularity
  - How many?
- Thread types
  - Processing bound
  - Memory bound
- What to run? When? Where?
- Communication
- Degree of interaction

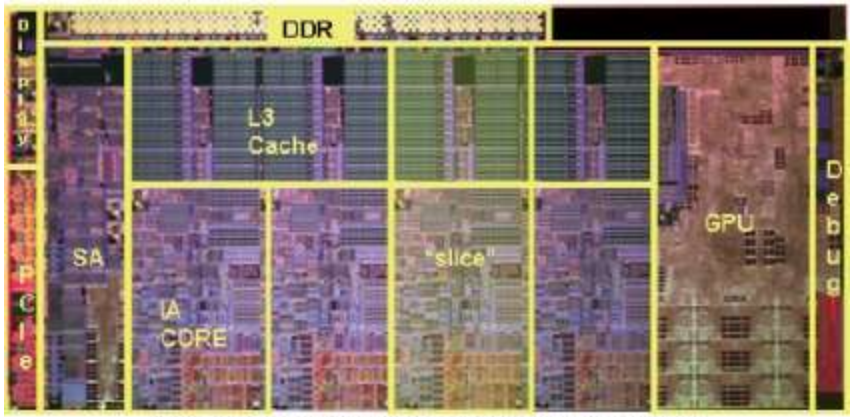
Processor We Will Look at



SPARC T4

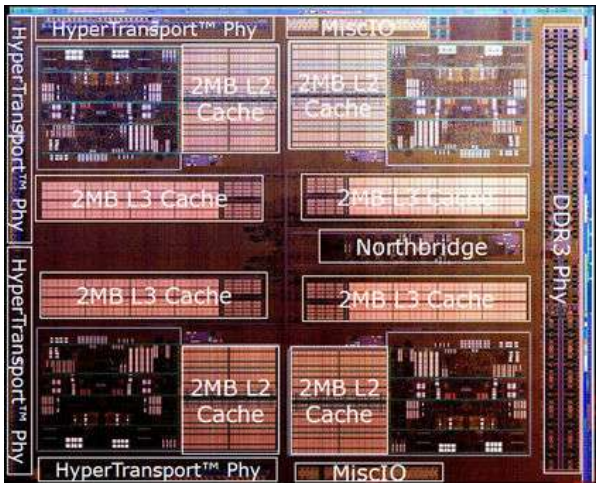


IBM Power 7



Sandy Bridge die photo (courtesy of ISSCC).

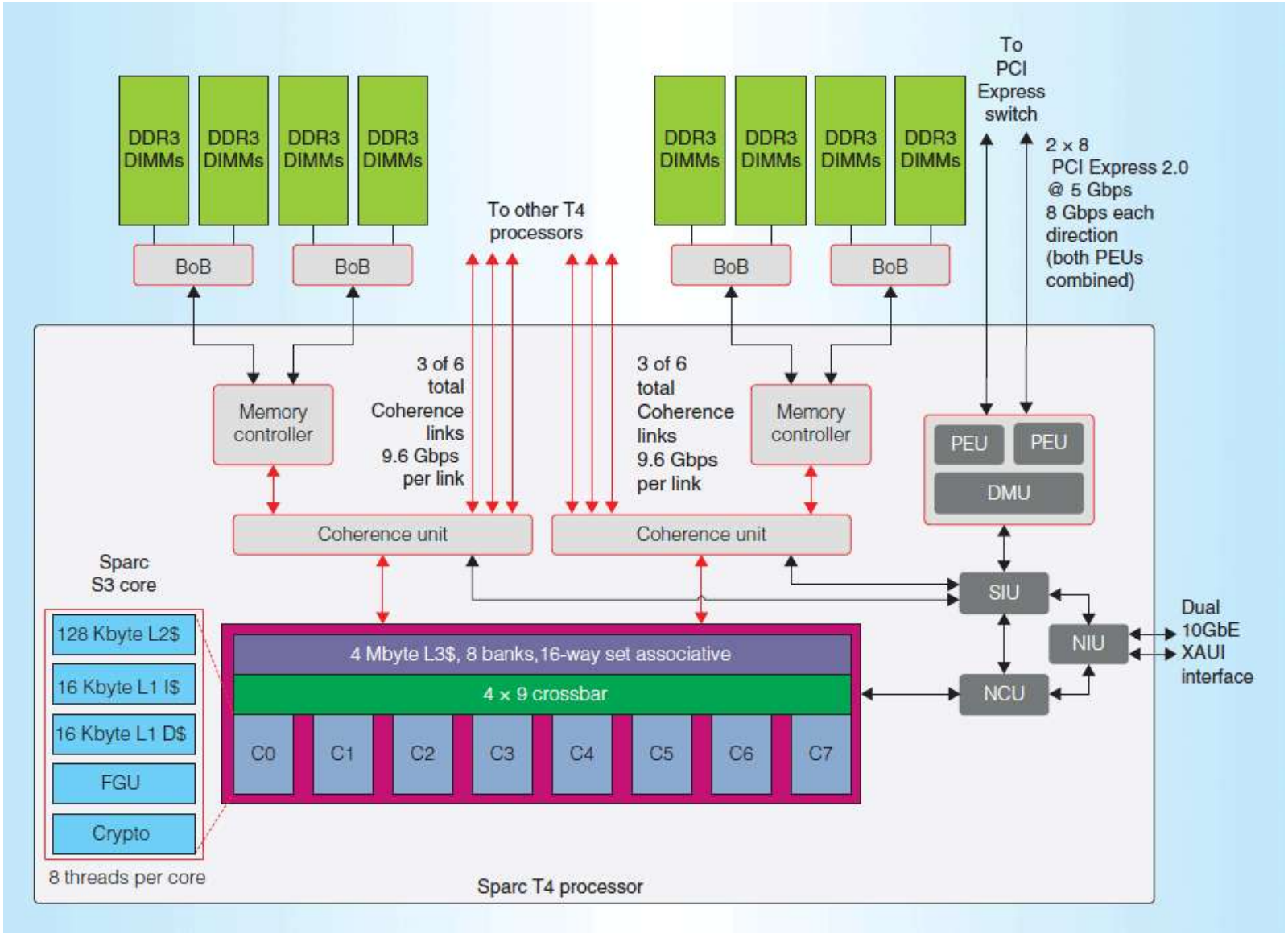
Intel Sandy Bridge



AMD 15h (Bulldozer)

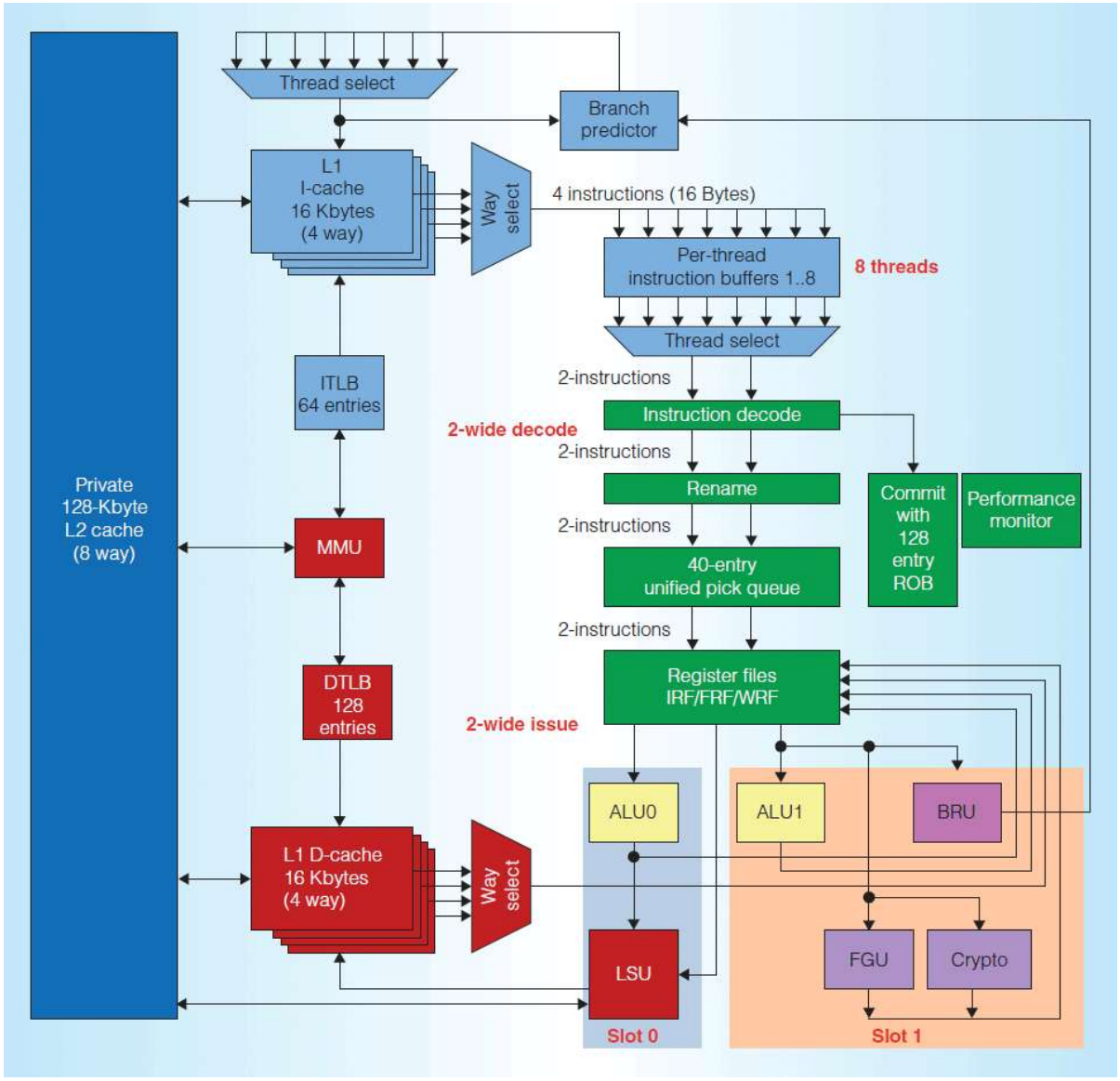
SPARC T4

- The next generation of Oracle multicore
- 855M transistors
- Supports up to 64 threads
  - 8 cores
  - 8 threads per core
  - Cannot be deactivated by software
- Private L1 and L2 and shared L3
- Shared L3
  - Shared among 8 cores
  - Banked
  - 4MB
  - 16-way set associative
  - Line size of 64 bytes



PEU: PCI-Express unit; DMU: Data management unit; NIU: Network interface unit; NCU: Non-cacheable unit; SIU: System interface unit

The Cores in SPARC T4: S3



The Cores in SPARC T4: S3

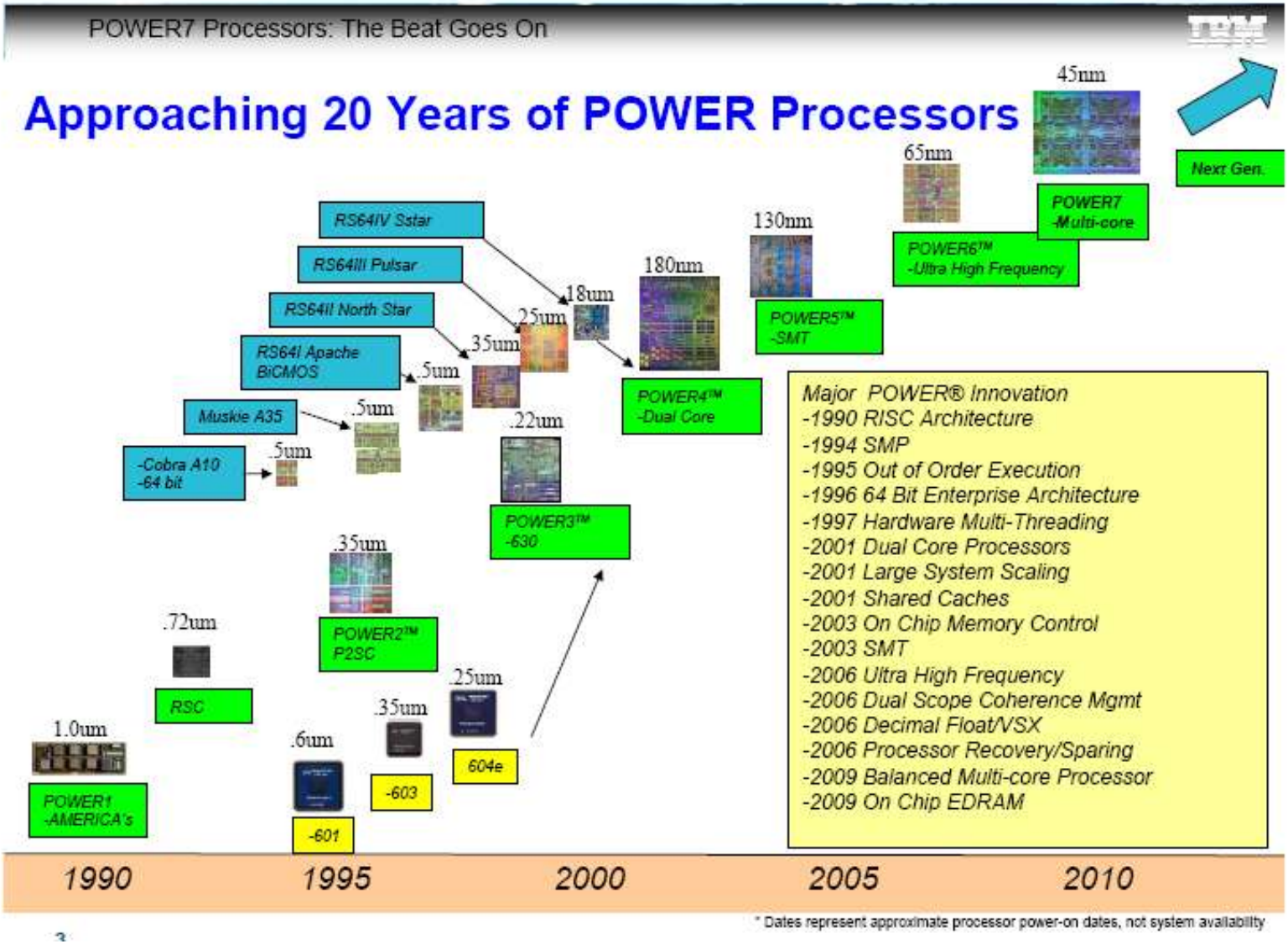
- Supports up to 8 threads
- DL1 and IL1:
  - 16KB
  - 4-way set associative
  - 32 bytes cache line
  - Shared by all 8 threads
- IL1 has 3 line prefetch on-miss
- DL1 has stride-based and next-line prefetchers

What to Do About Prefetching?

- Use arrays as much as possible. Lists, trees, and graphs have complex traversals which can confuse the prefetcher.
- Avoid long strides. Prefetchers detect strides only in a certain range because detecting longer strides requires a lot more hardware storage.
- If you must use a linked data structure, pre-allocate contiguous memory blocks for its elements and serve future insertions for this pool.
- Can you re-use nodes from your linked-list?

Questions

- Suppose that you have 8 threads that are computation intensive and another 8 memory bound… how will you assign them to cores on T4?
- What if all threads are computation bound?
- What if they are all memory bound?
- T4 gives the software the ability to *pause* a thread for few cycles. When will you use this feature?



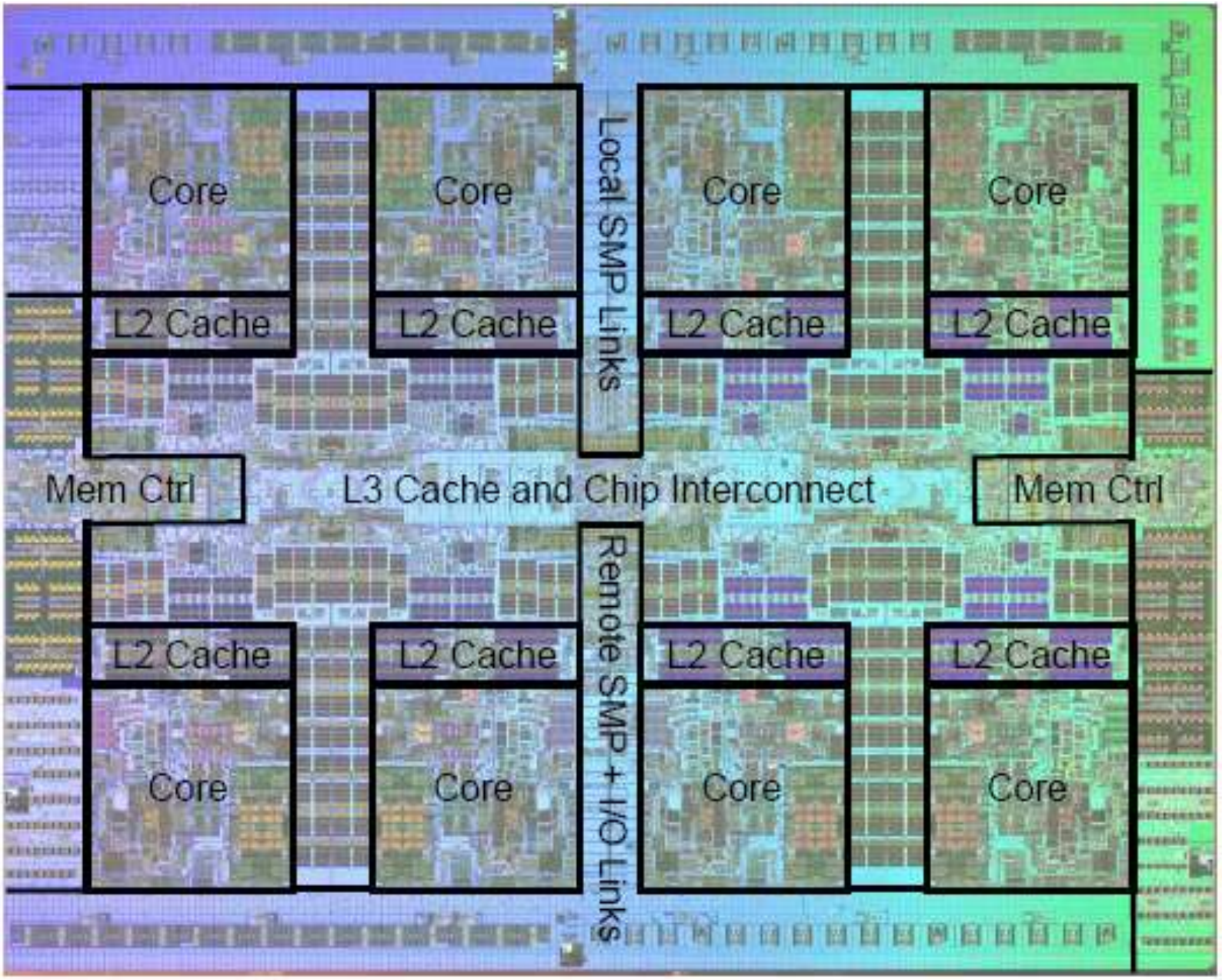
Source: Slides from Joel M. Tendler from IBM

IBM Power 7

- Supports global shared memory space for POWER7 clusters
  - So you can program a cluster as if it were a single system
- Design for power-efficiency, unlike Power 6
- ~1.2B Transistors
- Up to 8 cores and 4-way SMT
- TurboCore mode that can turn off half of the cores from an eight-core processor, but those 4 cores have access to all the memory controllers and L3 cache at increased clock speeds.
- 3.0 – 4.25 GHz clock speed

IBM Power 7: Cache Hierarchy

- 32KB DL1 and IL1 per core
- 256KB L2 per core
- eDRAM L3 4MB per core (total of 32MB)
  - Very flexible design for L3



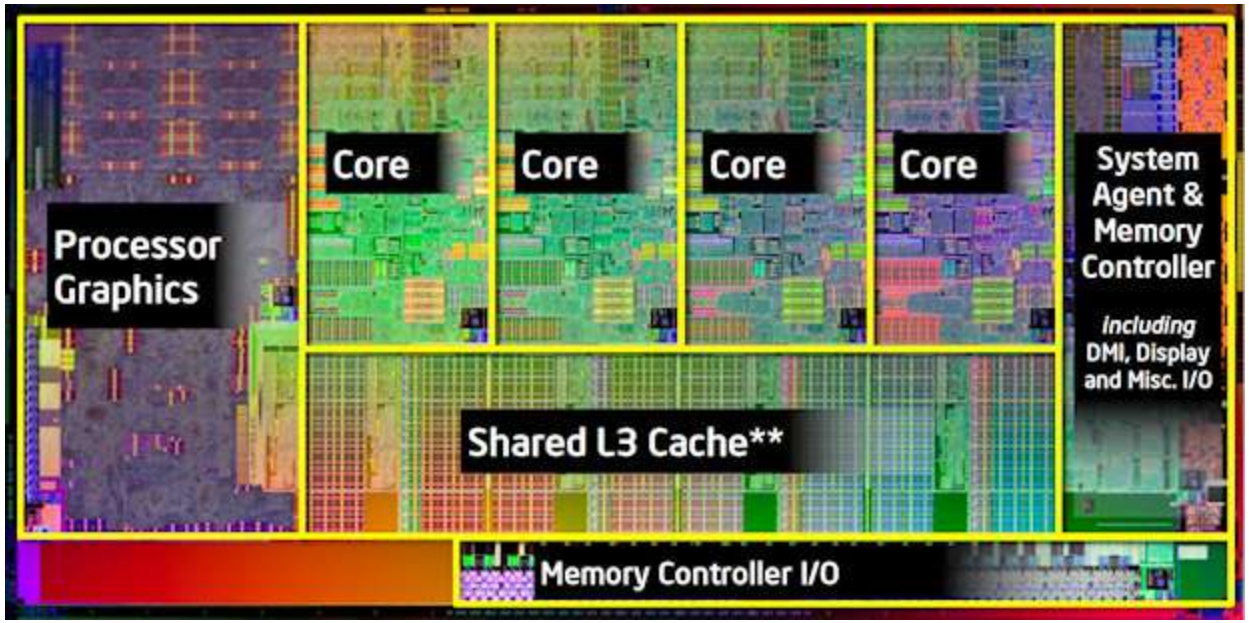
Source: Slides from Joel M. Tendler from IBM

Questions

- If you are writing the same programs for T4 and Power 7, will you change anything?
- As a programmer, how can you make use of the cache hierarchy?

Intel Sandy Bridge

- New microarchitecture
- 22nm

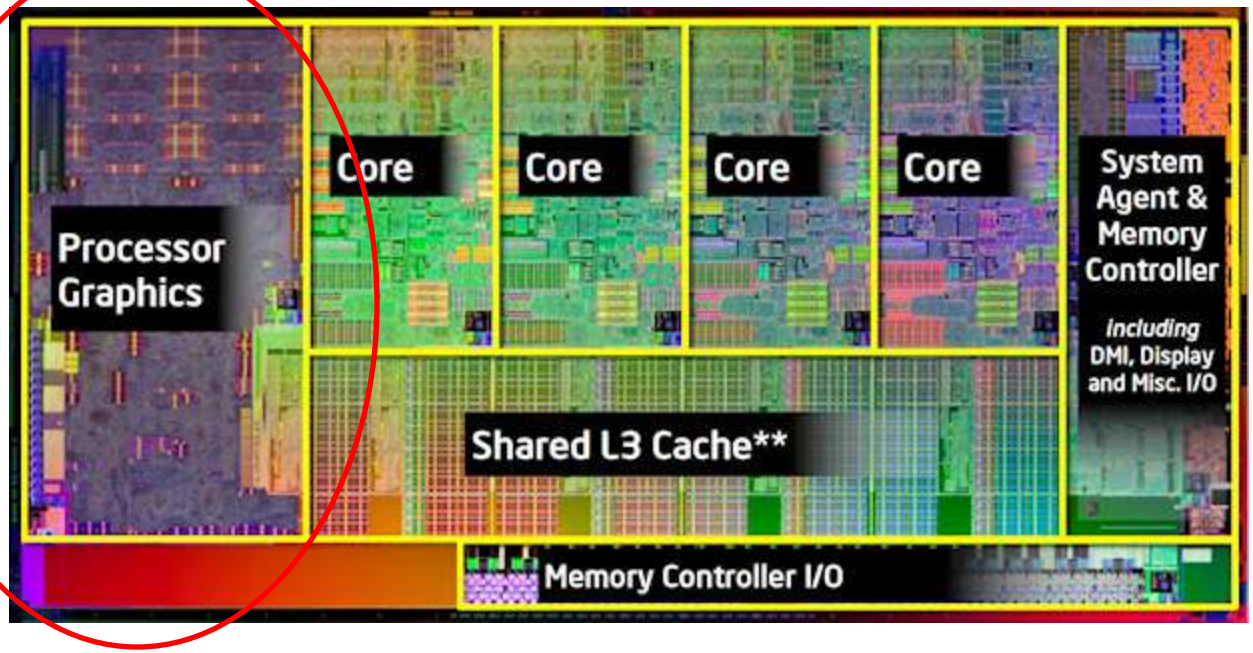


Sandy Bridge, source: Intel

Intel Sandy Bridge

- New microarchitecture
- 22nm

Heterogeneous Multicore

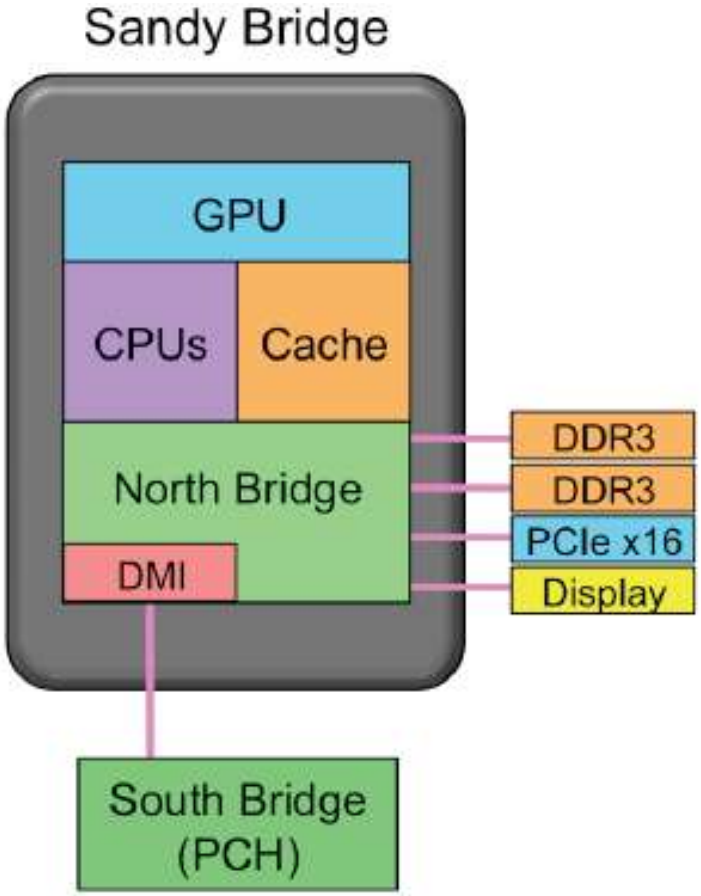


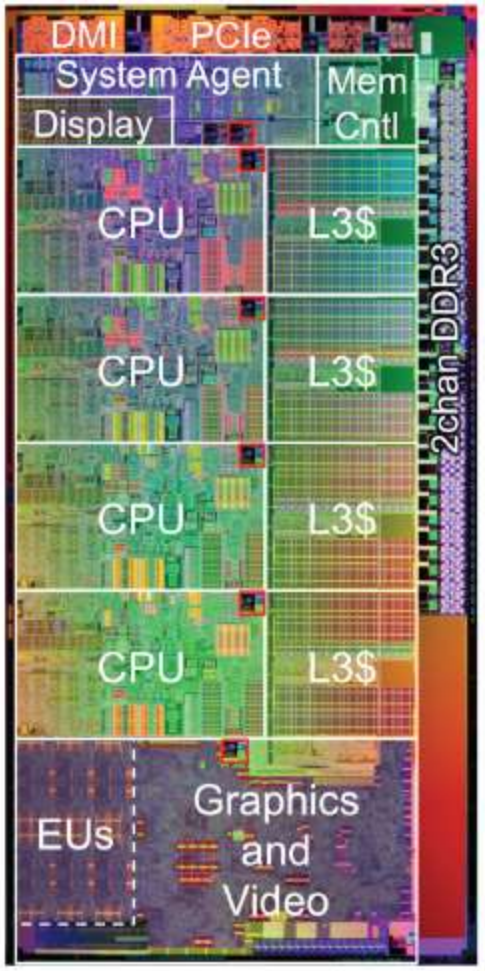
Sandy Bridge, source: Intel

Intel Sandy Bridge

- Improvement over its predecessor Nehalem
- Targeting multimedia applications
  - Introduced Advanced Vector Extensions (AVX)

- The GPU can access the large L3 cache
- Intel's team totally re-designed the GPU





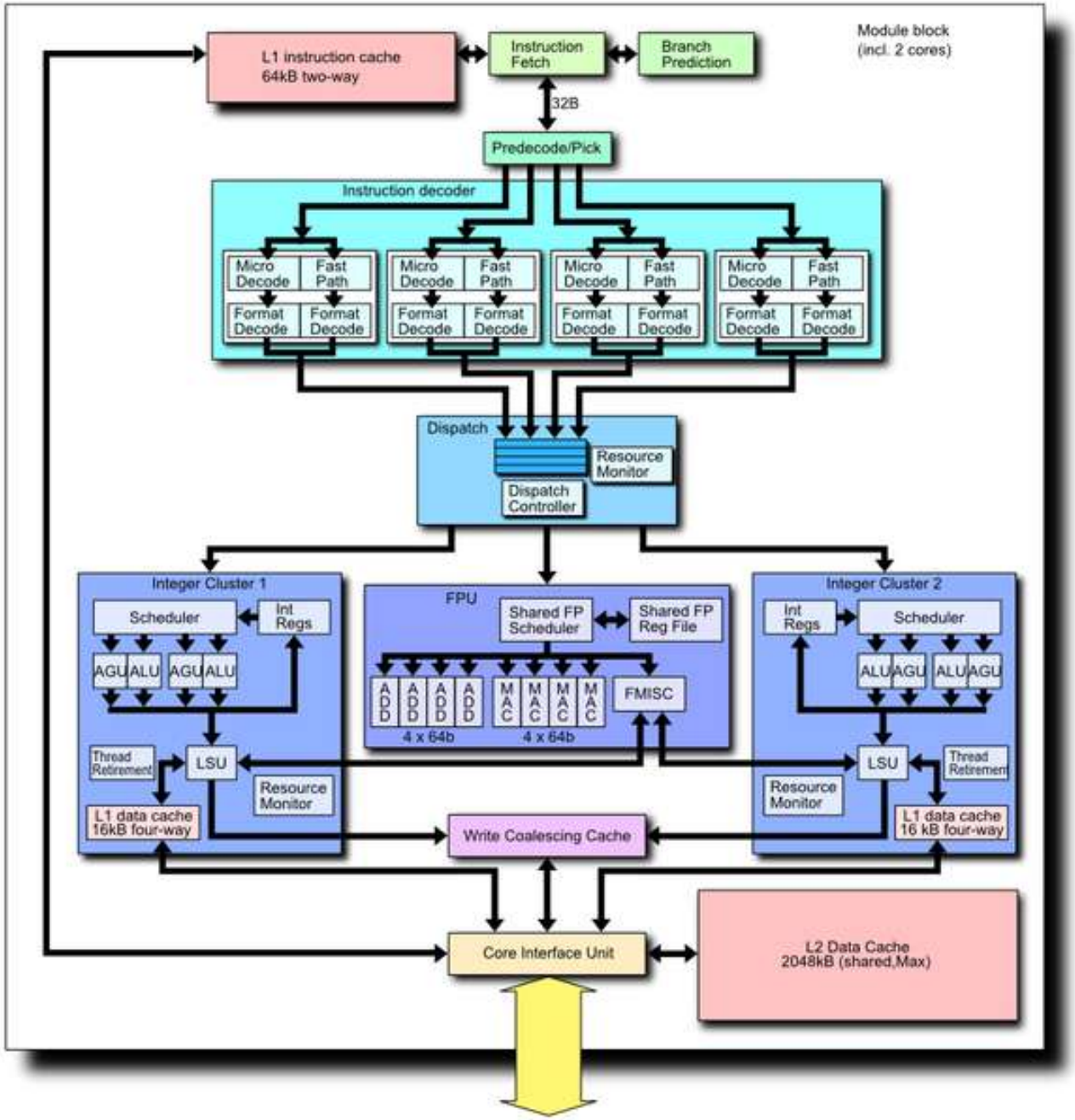
Features for You to Use

- Sandy bridge processors have 256bit wide vector units per core
- As a programmer you can:
  - Using AVX instructions
  - Use the compiler to vectorize your code
  - http://ispc.github.com/

Question: Can you design your program with different type of parallelism?

AMD 15h (Bulldozer)

- Designed from scratch
- 32nm technology
- combine the functions of what would normally be two discrete cores into a single package ("module" in AMD literature)
- The aim was to hit higher frequencies
- Performance worse than expected!


Module block
(incl. 2 cores)
L1 instruction cache
64kB two-way
Instruction
Fetch
Branch
Prediction
32B
Predecode/Pick
Instruction decoder
Micro
Decode
Fast
Path
Format
Decode
Format
Decode
Micro
Decode
Fast
Path
Format
Decode
Format
Decode
Micro
Decode
Fast
Path
Format
Decode
Format
Decode
Micro
Decode
Fast
Path
Format
Decode
Format
Decode
Dispatch
Resource
Monitor
Dispatch
Controller
Integer Cluster 1
Scheduler
Int
Regs
AGU
ALU
AGU
ALU
Thread
Retirement
LSU
Resource
Monitor
L1 data cache
16kB four-way
FPU
Shared FP
Scheduler
Shared FP
Reg File
ADD
ADD
ADD
ADD
MAC
MAC
MAC
MAC
FMISC
4 x 64b
4 x 64b
Integer Cluster 2
Int
Regs
Scheduler
ALU
AGU
ALU
AGU
LSU
Thread
Retirement
Resource
Monitor
L1 data cache
16 kB four-way
Write Coalescing Cache
Core Interface Unit
L2 Data Cache
2048kB (shared,Max)

AMD 15h (Bulldozer)

- According to AMD, Windows 7 doesn't understand Bulldozer's resource allocation very well.
  - Windows 7 sees 8 discrete cores
- Question: What can you do about this?

More Questions: Not Related to a Specific Processor

- Your code does not execute alone. Can you do something about it to avoid interference?
- As a programmer, what can you do about power?

Conclusions

- You need to know the big picture at least
  - number of cores and SMT capability
  - Interconnection
  - Memory hierarchy
  - What is available to software and what is not
- The art of delegation
  - What to do at user level and what to leave for the compiler, OS , and runtime