

Theme: Towards Reconfigurable High-Performance Computing
Lecture 3

Platforms I: Advanced Architectural Features

Andrzej Nowak

CERN openlab (Geneva, Switzerland)

Inverted CERN School of Computing, 3-5 March 2008

Introduction

- **Recap**
 - Multi-core hardware is becoming prevalent, and is tightly coupled with the software which drives it

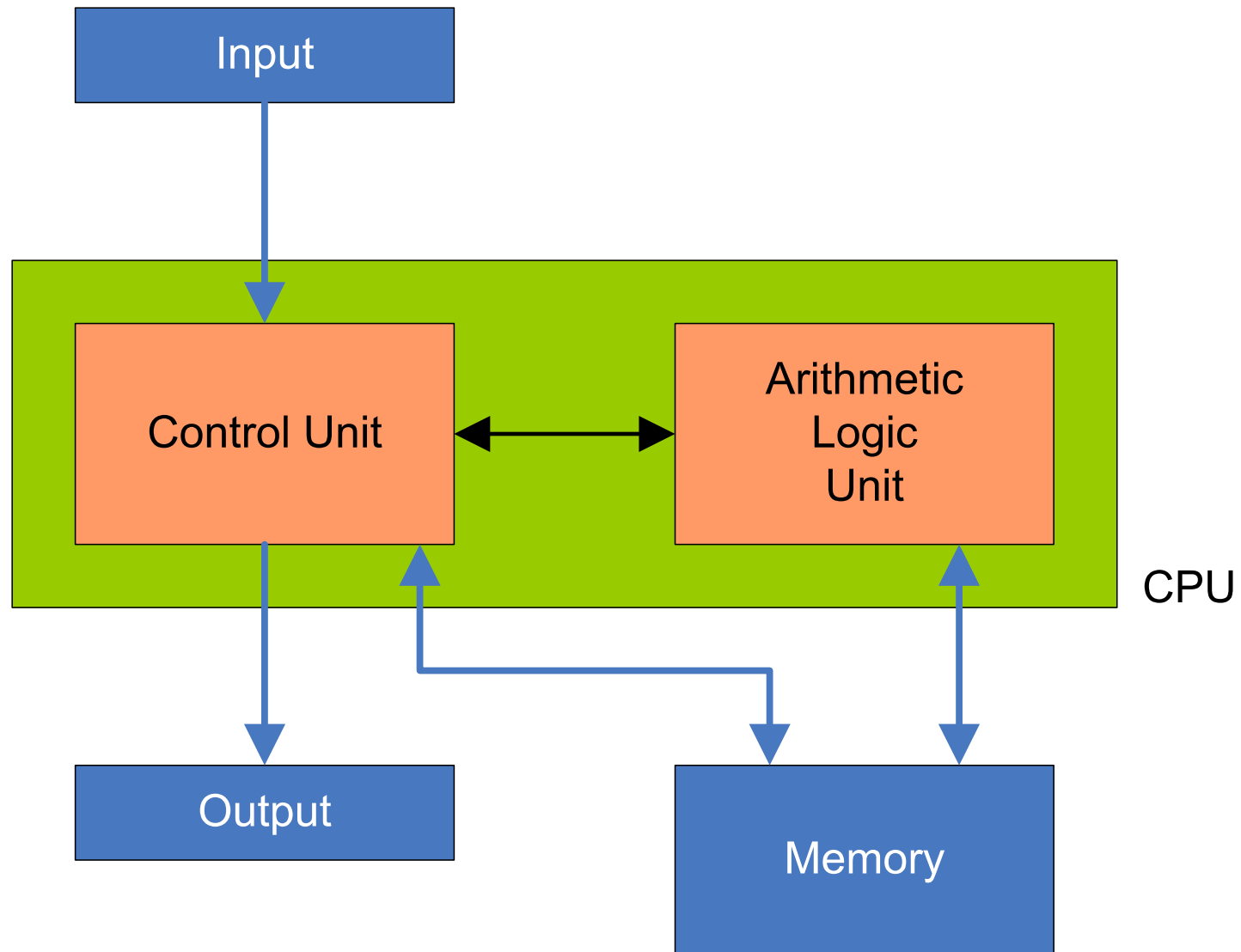
- **Objectives:**
 - Explain key architectural concepts
 - Discuss x86 architectural extensions
 - Discover interesting multi-core designs and interconnects

- **Contents:**
 - Systems architecture basics
 - Instruction set extensions
 - Compilers and parallelism
 - Advanced multi-core architecture discussion

COMPUTER ARCHITECTURES

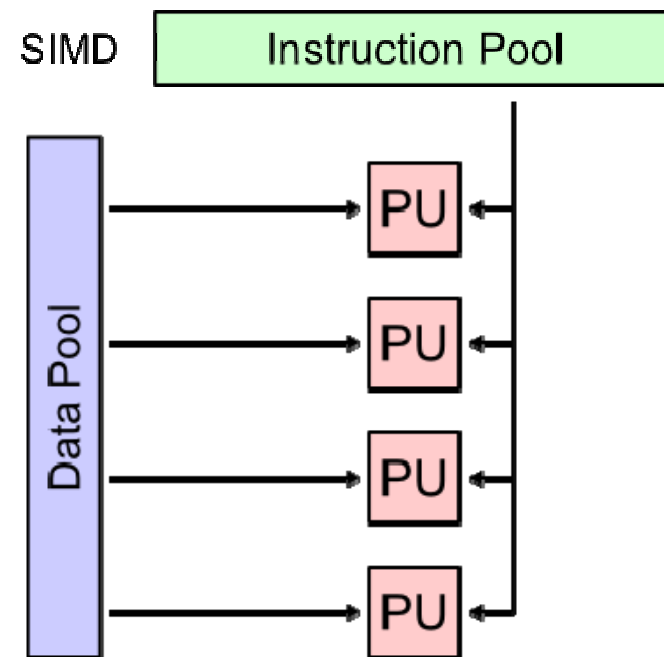
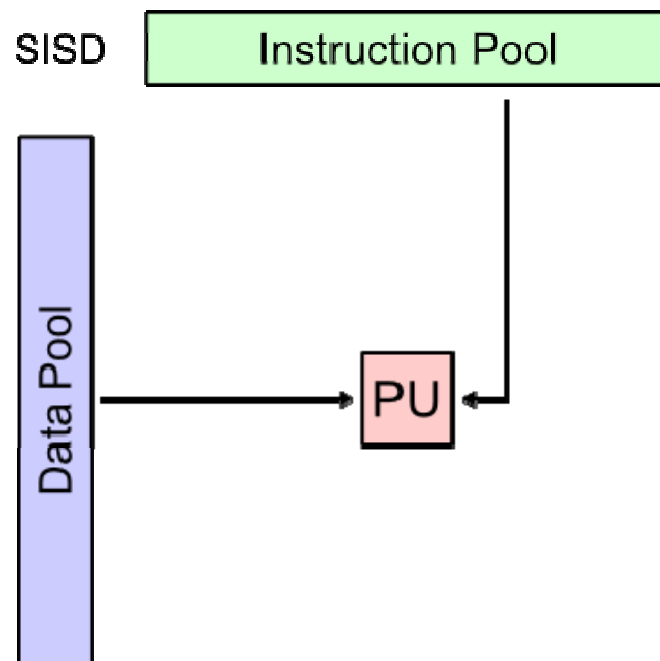
And their extensions

Von Neumann architecture



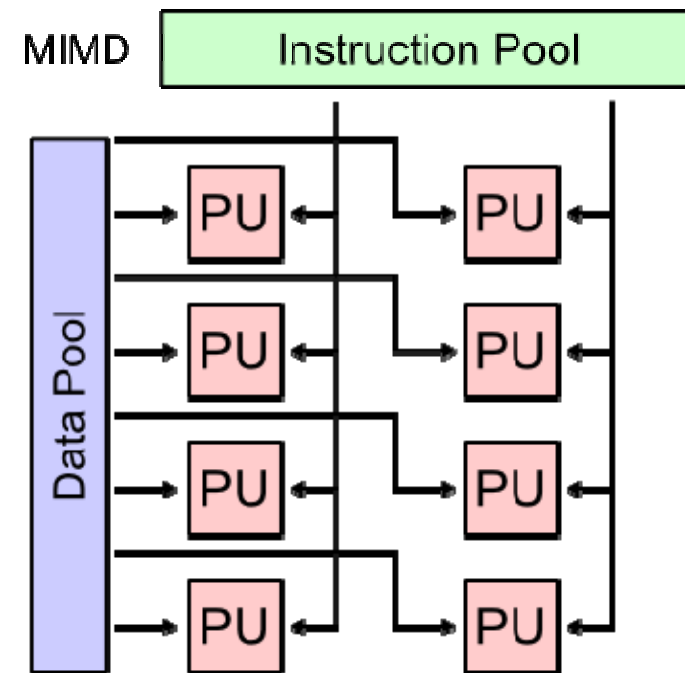
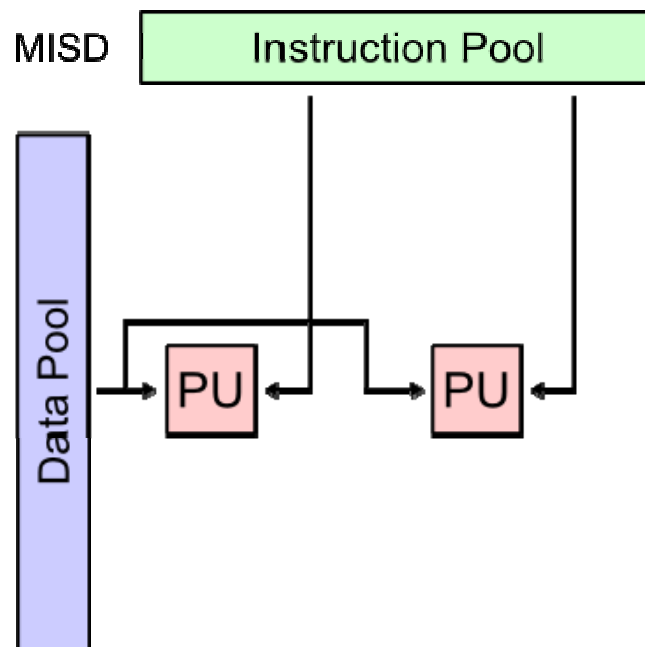
Flynn's taxonomy (1)

- **SISD – Single Instruction, Single Data**
 - Classical Von Neumann's model
- **SIMD – Single Instruction, Multiple Data**
 - A GPU



Flynn's taxonomy (2)

- **MISD – Multiple Instruction, Single Data**
 - Redundant systems, pipeline systems (disputable)
- **MIMD – Multiple Instruction, Multiple Data**
 - Distributed systems



X86 architectural extensions

- **Extensions in Intel CPUs:**
 - FPU, MMX, SSE, SSE2, SSE3, SSSE3, SSE4 (SSE4.1 + SSE4.2), SSE5, EM64T
- **Extensions in AMD CPUs:**
 - AMD (pre K6), MMX, SSE, SSE2, SSSE3, SSE4, 3DNow!, 3DNow!+, 3DNow! Professional (SSE + 3DNow!)
- **Understanding SIMD extension history is helpful in understanding modern vector instructions**

MMX

- Intel's first attempt at adding SIMD capabilities to their CPUs; introduced in 1997
- **Packet data type concept**
 - 64 bits = 2x 32bits = 4x 16bits = 8x 8bits
- **8 “new” 64bit integer registers – MM0 ... MM7 (mapped onto x87 the stack)**
- **Major flaws:**
 - floating point and SIMD could not be used at the same time
 - integer operations only
- **Embedded XScale CPUs (ARM family) use iwMMXt – Intel Wireless MMX Technology**
 - 64 bit packed data type
 - 16 data regs, 8 control regs

SSE

- **Introduced in 1999 with the Pentium III, 70 new instructions**
- **Fixed the 2 main MMX deficiencies**
- **8 truly new 128-bit registers – XMM0 ... XMM7 – 4x 32-bit float**
- **Later on, another 8 registers added**
- **FP Instructions:**
 - Data movement (M->R, R->M, R->R)
 - Arithmetic, bitwise, comparison
 - Data shuffling, data unpacking, simple data type conversion
- **INT instructions: simple arithmetic and movement**
- **Flaws:**
 - Register states had to be saved “manually” by the OS
 - Execution resources shared with the FPU
- **AMD introduced SSE in AthlonXPs (Palomino – 2001)**

3DNow!, AltiVec

- **3DNow! stands for AMD extensions to MMX, introduced in 1998**
 - 32-bit FP support
 - Some instructions from this family were added to the Pentium III as SSE
 - Later upgraded to 3DNow!+
- **AltiVec – Apple and IBMs vector extensions for PowerPC**
 - Developed between 1996 and 1998
 - Also known as “Velocity Engine” (Apple) and “VMX” (IBM)
 - Widely used by Apple in their flagship applications, as well as 3rd party developers such as Adobe
 - Technical details
 - 32 128-bit vector registers (can be split up into 8, 16 or 32 bit pieces)
 - Three register operands
 - Support for a special RGB data type, which does not map onto 64-bit floats easily
 - The IBM CELL supports AltiVec, as well as the IBM Power6

SSE2

- **Introduced with the Pentium 4 in 2001, 144 new instructions**
- **Technical details:**
 - 8 registers
 - 64-bit floating point
 - Minimized cache pollution
 - More sophisticated format conversions
 - Extended MMX instructions allow operation on XMM registers
- **Flaws:**
 - Accessing misaligned data introduces a penalty
 - Unimpressive throughput compared to MMX
- **AMD introduced SSE2 in 2003 in the Athlon64 and Opteron families**
 - 8 additional registers

SSE3, SSSE3

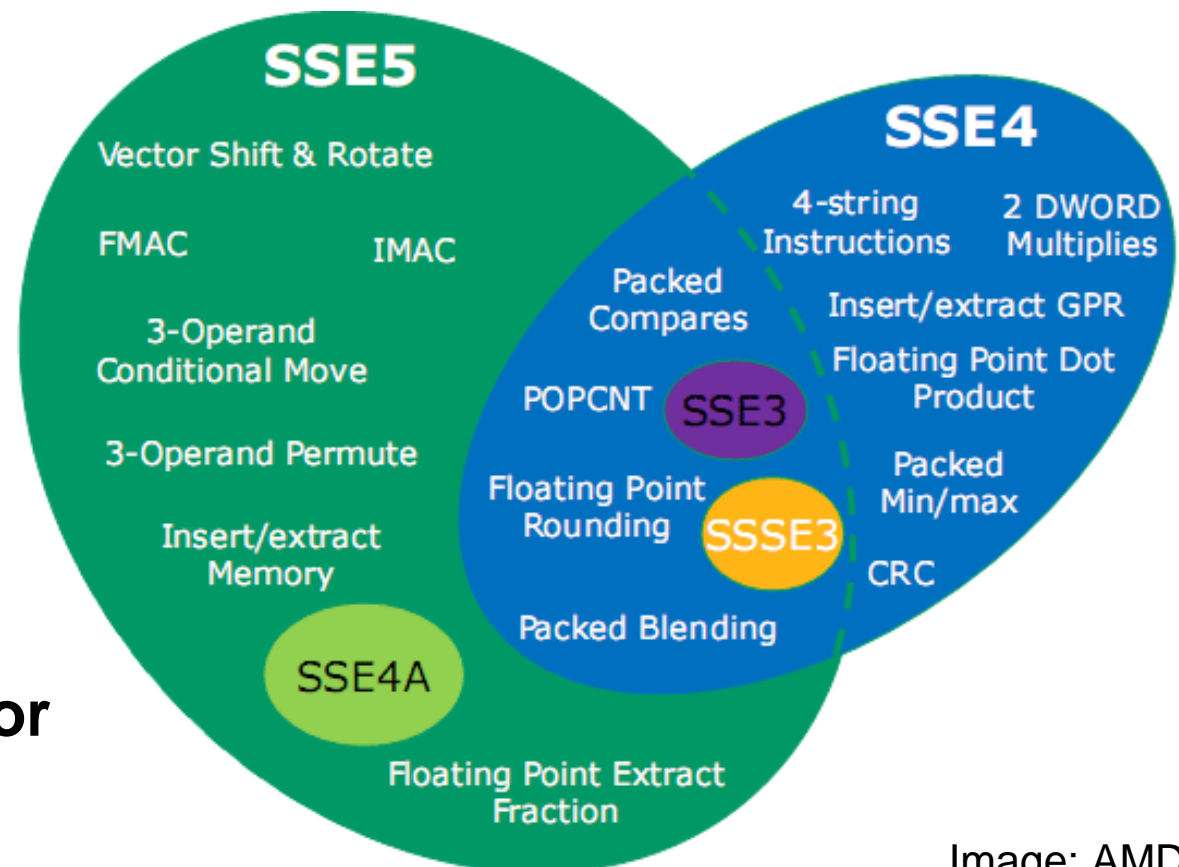
- **SSE3 introduced in 2004 in the Pentium 4 (“Prescott” – hence the a.k.a. name “PNI”)**
- **SSE3 Technical details:**
 - Horizontal operations portfolio expanded, i.e. add/subtract elements in a single vector
 - Improved misaligned data loading
 - FP -> Int conversion simplified
- **SSSE3 is really a new iteration, introduced in Intel Core chips**
 - 16 new instructions – some packed and horizontal operations
 - No new registers
 - Operates on MMX or XMM registers
 - Unsupported in AMD chips

SSE4

- **54 new instructions introduced publicly in 2007**
 - SSE4.1: 47, SSE4.2: 7 (only in Nehalem)
- **Technical details:**
 - No new data types, no new registers
 - Compiler vectorization improved
 - Significant packed dword computation improvement
 - Some instructions are not multimedia related
 - Some instructions take an implicit third operand
- **You can use SSE4 with Intel compilers from version 10.0 onwards**

SSE5

- **AMD specific – a 128-bit extension of a 64-bit extension to the 32-bit original x86 instruction set; targeted for 2009**
- **170 new instructions, targeting:**
 - HPC
 - Multimedia
 - Security applications
- **Features:**
 - 3 operand ops
 - Fused instructions
 - MADD instructions
- **SSE5 software simulator**



AMD x86-64 (a.k.a. EM64T or Intel64)

- **Roles reversed – Intel had to follow AMD’s lead**
- **64-bit operations fully supported**
 - Arithmetic
 - Registers
 - Virtual addresses
- **Expanded virtual and physical address space**
- **SSE, SSE2 and SSE3 (Intel) instructions included**
- **Cleanups**
- **There are some differences between AMD’s and Intel’s implementations**

AMD Lightweight Profiling

- **Only 2 new instructions**
 - Enable/disable profiling
 - Retrieve results
- **No interrupts needed (current situation is the opposite)**
- **Profiling on the fly supported**
- **Drawbacks:**
 - New silicon needed
 - Profiling on the fly might not be that easy due to OS designs
- **Introduced no sooner than late 2008-2009**
- **Intel does not comment, but we already know that upcoming Performance Monitoring Units will not differ greatly from what we have today**

AMD Extensions for Software Parallelism

- **No details yet, apart from the fact that this extension will upgrade the existing x86 instruction set**
- **The instruction set and surrounding optimizations will be “broad”, AMD says**
- **Analysts say that this feature might have a profound impact on the processor industry**
- **Intel does not comment**

X86 extensions summary

- **During the last 10 years:**
 - We've moved from simple 32-bit integer operations to complex 64-bit packed and floating point instructions
 - We've received some dedicated hardware for the extensions in question
 - We've moved from 32-bit to 64-bit – more throughput, but more memory used as well
- **The future:**
 - Non x86 architectures
 - The LRB instruction set
 - X86 derived
 - Mostly multimedia / HPC processing
- **As always, manuals from Intel or AMD will come in handy when programming using extensions**

PARALLEL PROGRAMMING

And the missing golden bullet for the gun of multi-core

The Core 2 issue ports

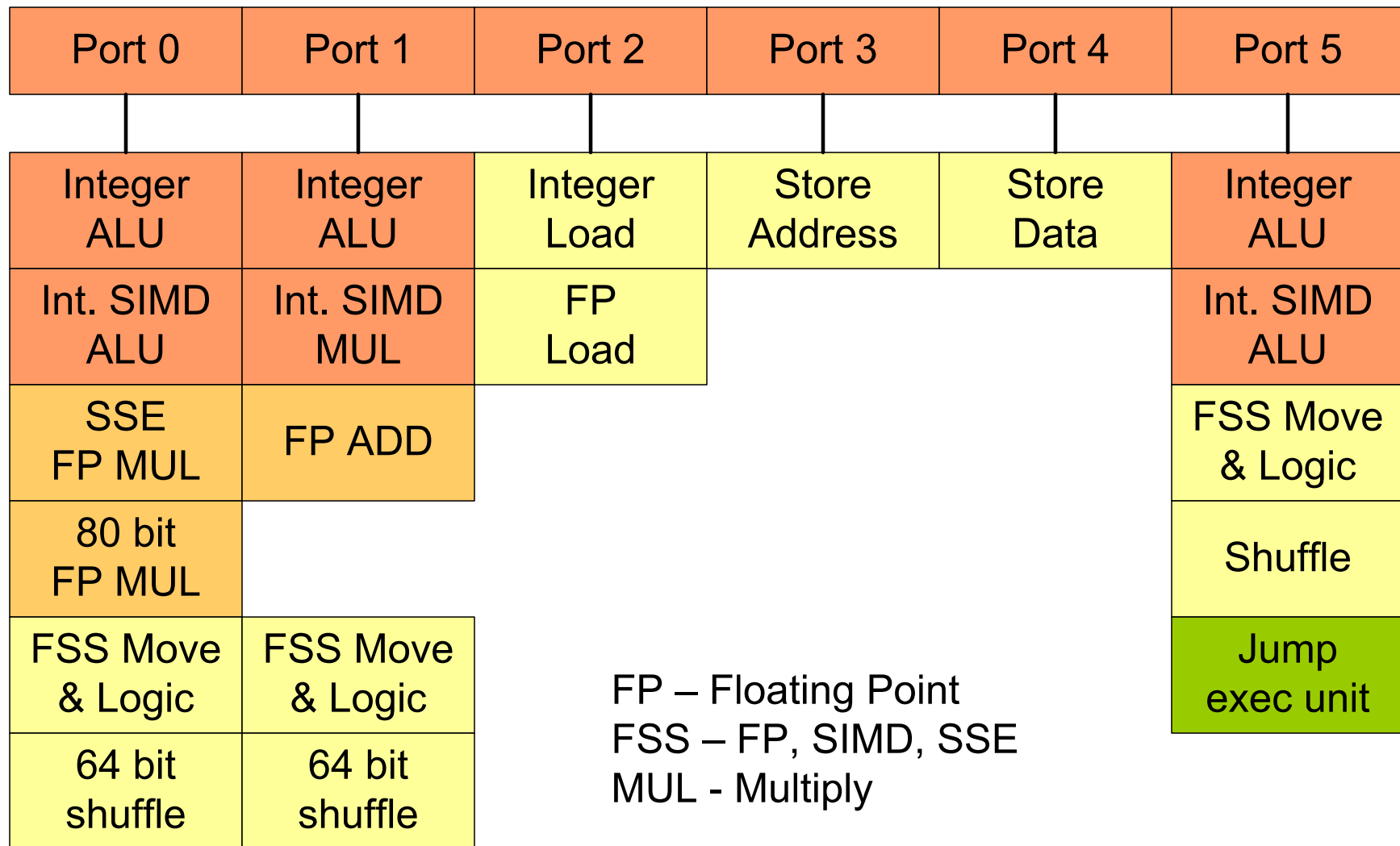


Image: based on Sverre Jarp's work

Instruction layout (Core 2)

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

ASM code: `movsd 16(%rsi), %xmm0`
`subsd 48(%rdi), %xmm0 // load & subtract`
`andpd _2i10floatpacket.1(%rip), %xmm0 // and with a mask`
`comisd 24(%rdi), %xmm0 // load and compare`
`jbe ..B5.3 # Prob 43% // jump if FALSE`

| Cycle | Port 0 | Port 1 | Port 2 | Port 3 | Port 4 | Port 5 |
|-------|--------|--------|-------------------|--------|--------|--------|
| 1 | | | load 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 |

Common parallel programming libraries (1)

- **Pthreads, Windows threads**
 - Fine grained control
 - Lightweight
 - Shared memory only
 - OS dependent
 - Often painful to debug

- **OpenMP**
 - A simple set of #pragma extensions
 - Several languages supported: C, C++, Fortran
 - Several implementations exist – compiler dependent
 - Gcc 4.2 and ICC support OpenMP
 - Several data scopes and scheduling models available
 - Can be used in a hybrid model with MPI
 - Shared memory only

Common parallel programming libraries (2)

- **Intel TBB – Threading Building Blocks**
 - An extension to C++
 - A set of algorithms and data types to facilitate parallel programming
 - Parallel sort, while, for, reduce
 - Container types: queue, vector, hash map
 - Scalable memory allocators
 - Mutexes, atomic operations
 - Automatic scaling to utilize all available processing units
 - Licensed on the GPLv2
 - Future features:
 - I/O tasks
 - Thread pinning (affinity)
 - New container classes
 - Improved interoperability with Intel Threading Tools

Common parallel programming libraries (3)

- **MPI – Message Passing Interface**
 - A language independent communications protocol
 - Point to point message passing and global operations
 - Numerous implementations exist
 - No shared memory concept in MPI-1 (v 1.2)
 - MPI-2 (v. 2.1) introduces numerous enhancements
 - Limited shared memory concept
 - Parallel I/O
 - Dynamic management
 - Remote memory support

- **PVM – Parallel Virtual Machine**
 - A network of machines is used as a single entity
 - Diminishing popularity

New and experimental compilers

- **Intel STM (transactional memory)**
 - A prototype version of the ICC C/C++ compiler
 - Added transactional programming constructs
 - Also works with OpenMP
 - Basic construct: `__tm_atomic { statements; }`
 - Very interesting development, worth following
- **Intel Ct (parallel programming language)**
 - An experimental data parallel programming environment
 - Designed to facilitate multi-core programming and increase portability
 - Best with vectors, sparse matrices, trees, linked lists
 - Mostly graphics-oriented so far

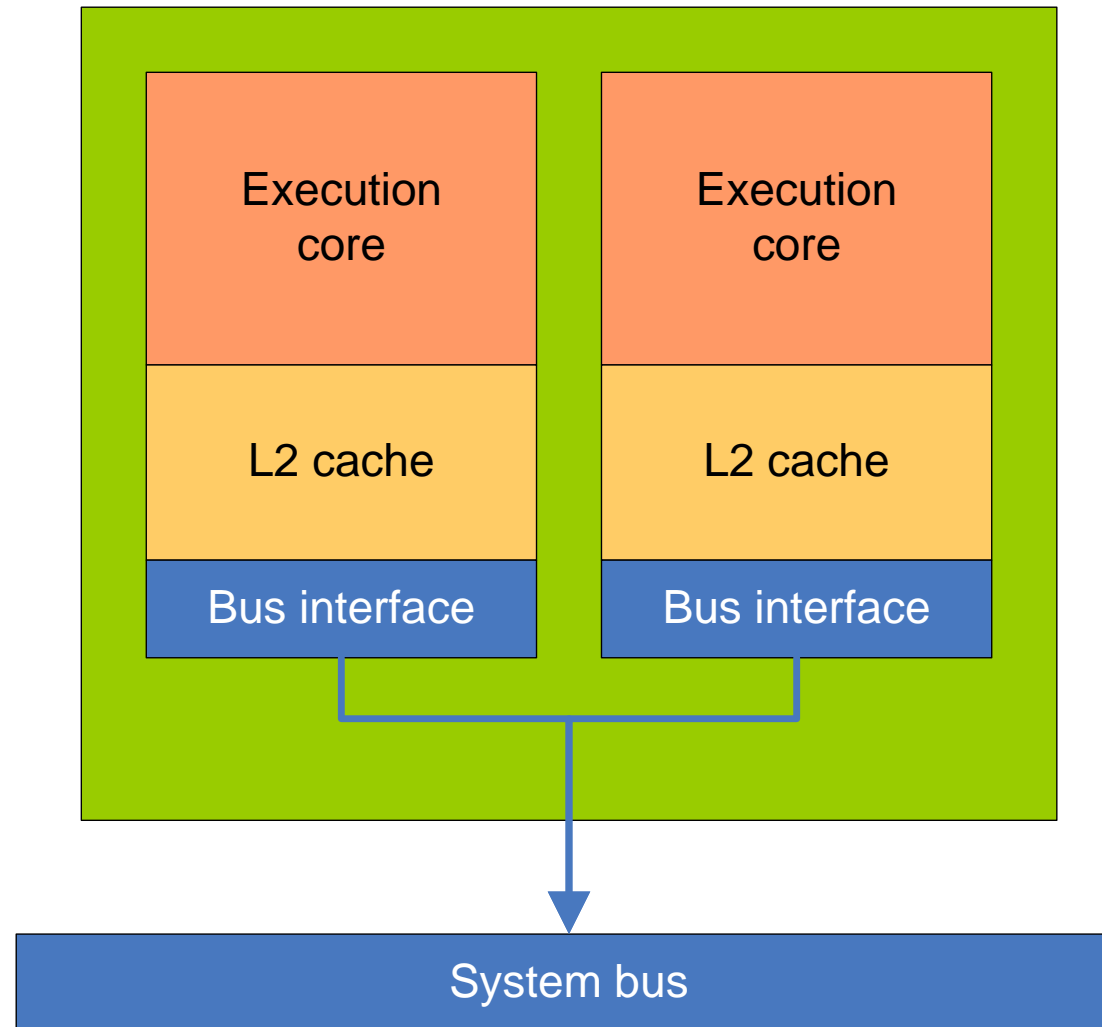
ADVANCED ARCHITECTURES

Multi-core architectures – high level overview

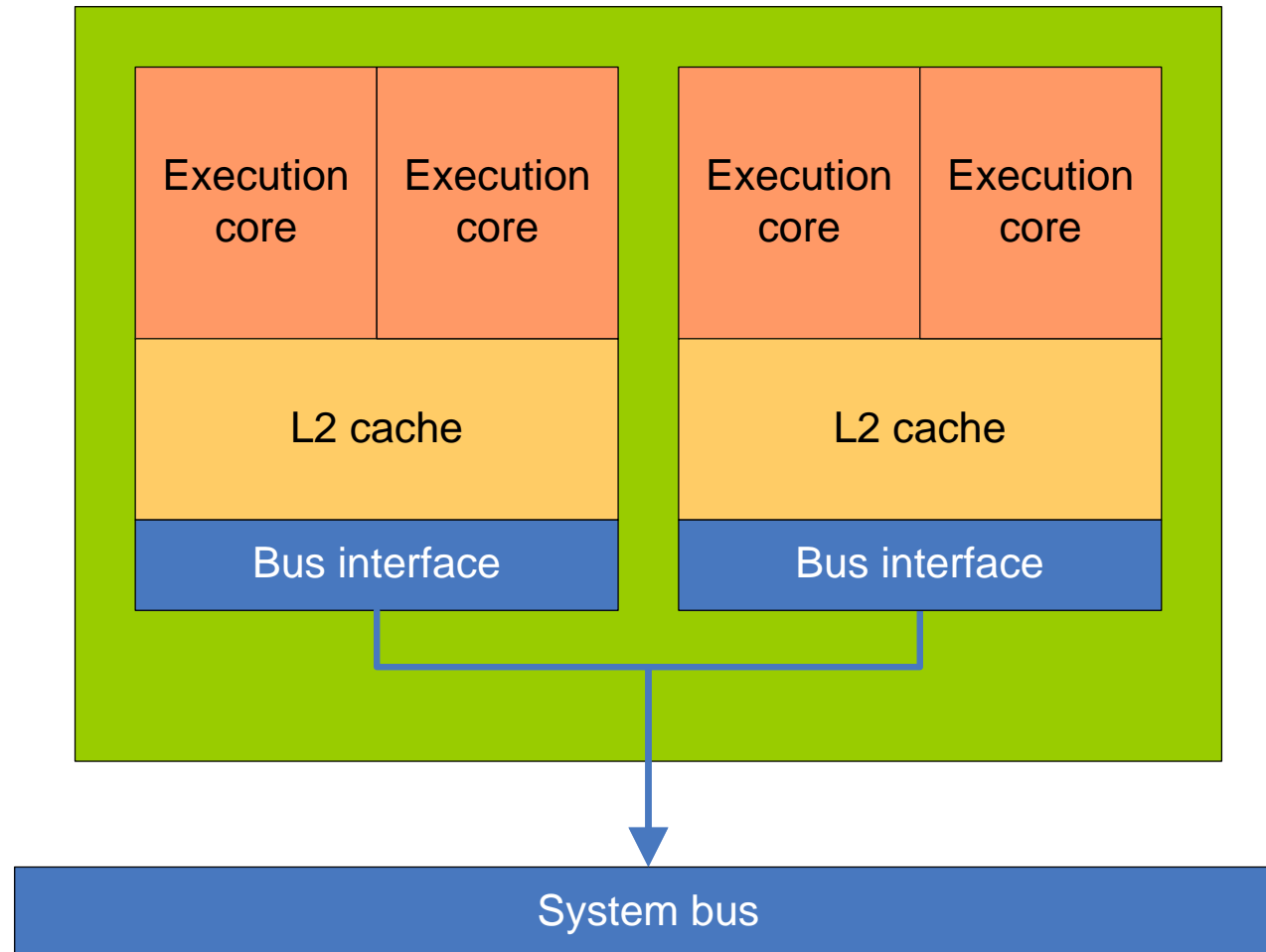
- **Modern consumer and mainstream architectures following the general trend**
 - Intel Pentium D, Intel Core, Intel Core2, Intel Itanium 2
 - AMD Athlon X2, AMD Phenom
- **Upcoming consumer and mainstream architectures**
 - Intel “Nehalem” (Core 3), Intel “Tukwila” (Itanium 3)
 - AMD “Fusion”
- **Less well known designs**
 - Sun “Niagara”, “Niagara 2” (UltraSPARC T1 and T2)
 - IBM CELL, Power6
 - Intel “Larrabee”
 - NVIDIA G80
 - Intel “Polaris”
 - SiCorTex

Multi-core architectures – Intel Pentium

D



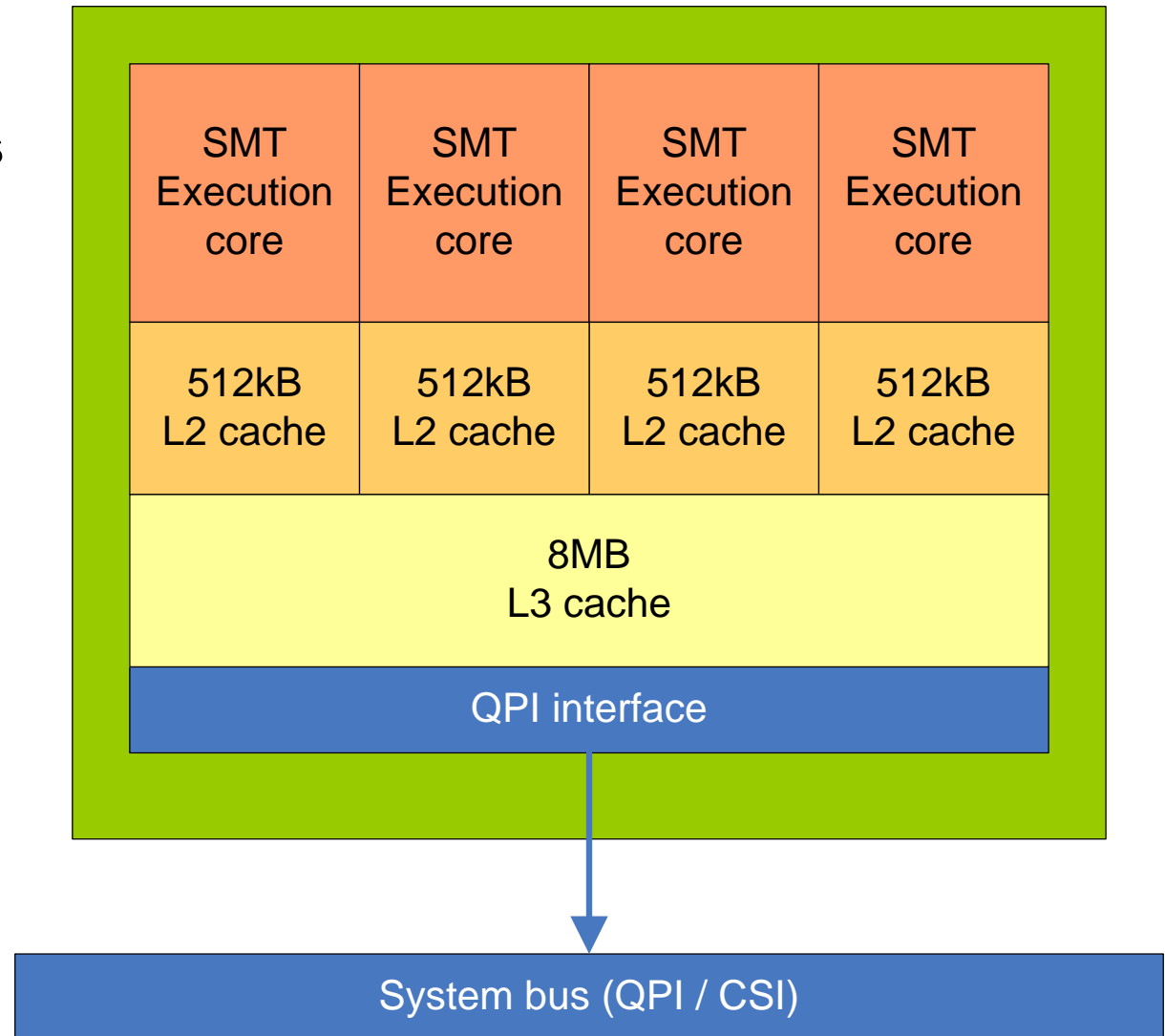
Multi-core architectures – Intel Core 2



Multi-core architectures – Intel

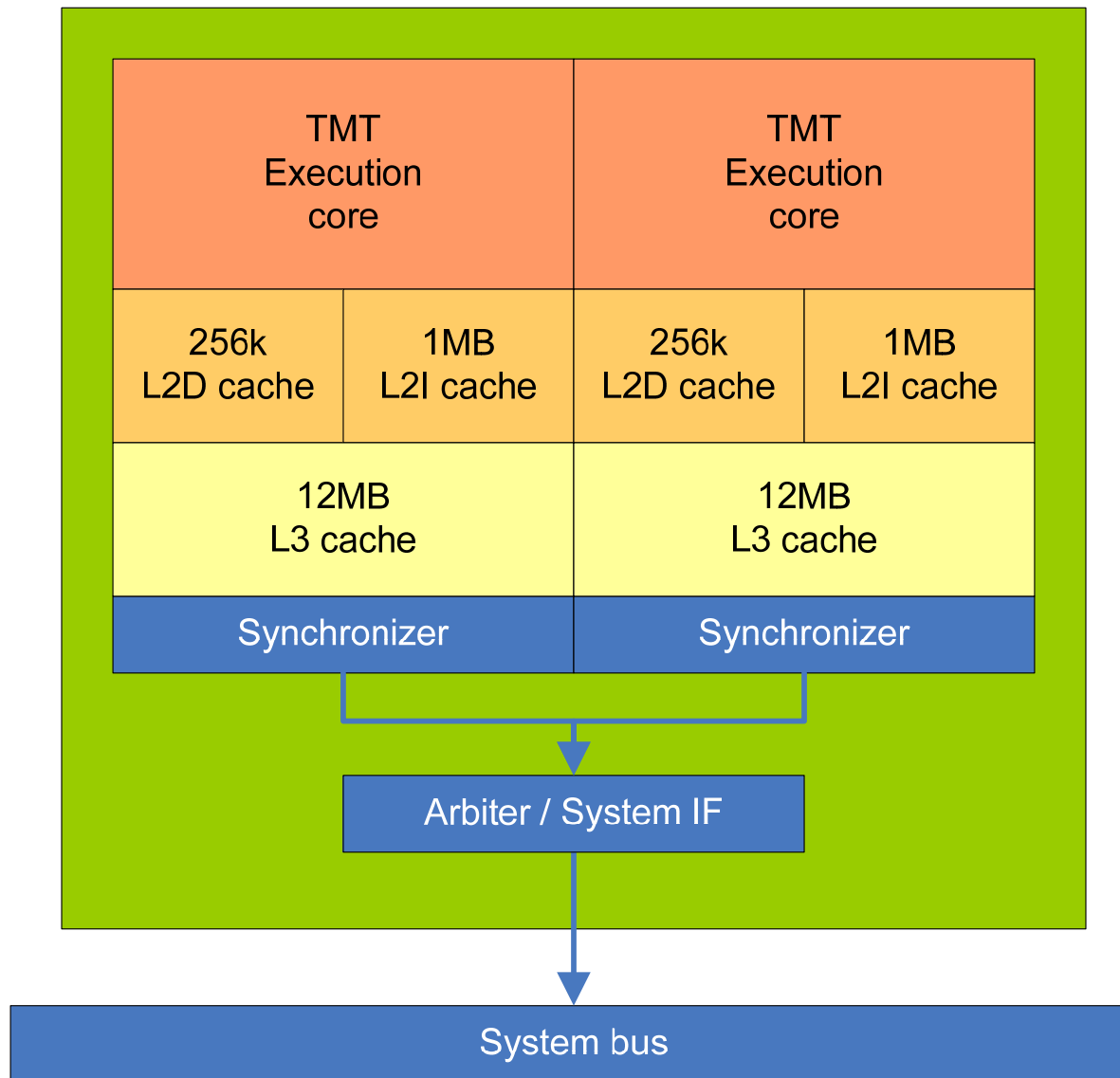
“Nehalem”

- **Release: YE 2008**
- **4-8 cores, 2 SMT threads per core**
- **Next generation interconnect (QPI)**
- **Advanced cache management**
- **Exclusive L2 and shared L3 caches**



Based on undisclosed data, might vary from actual product

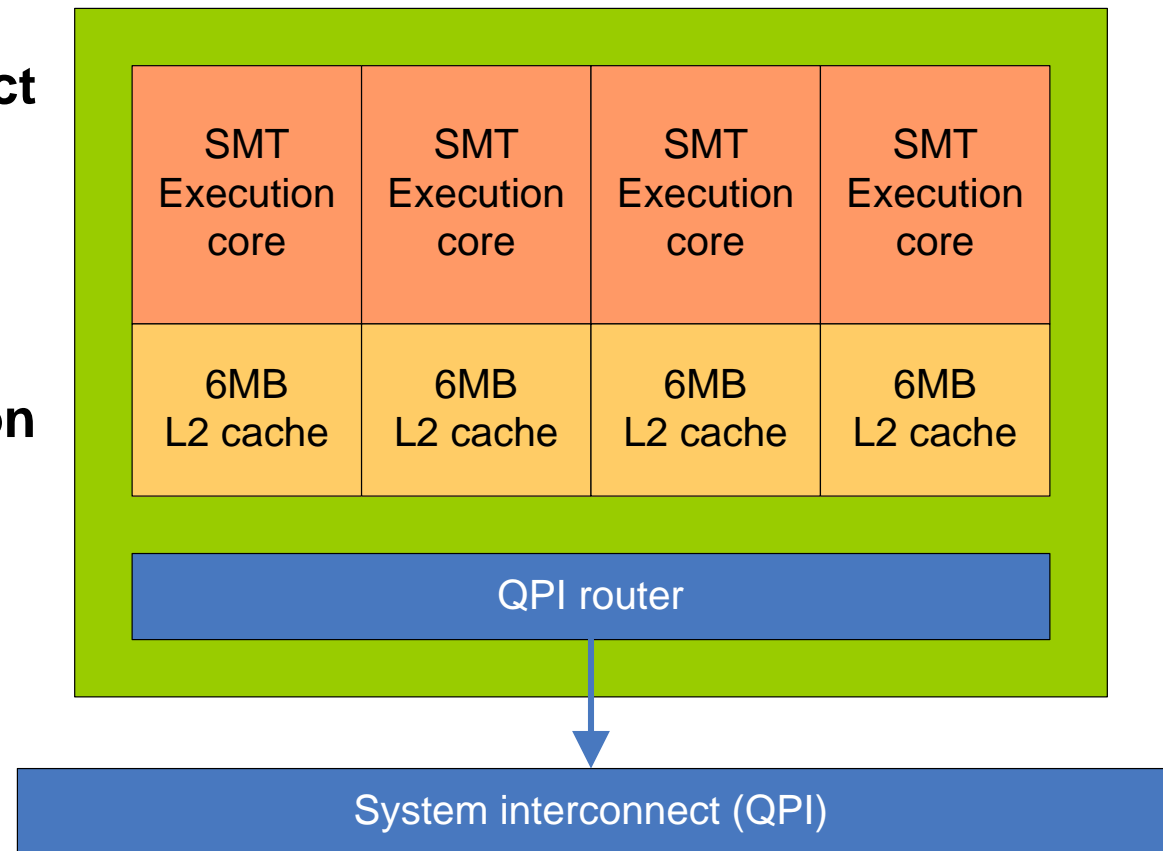
Multi-core architectures – Intel Itanium 2 (“Montecito”)



Multi-core architectures – Intel Itanium

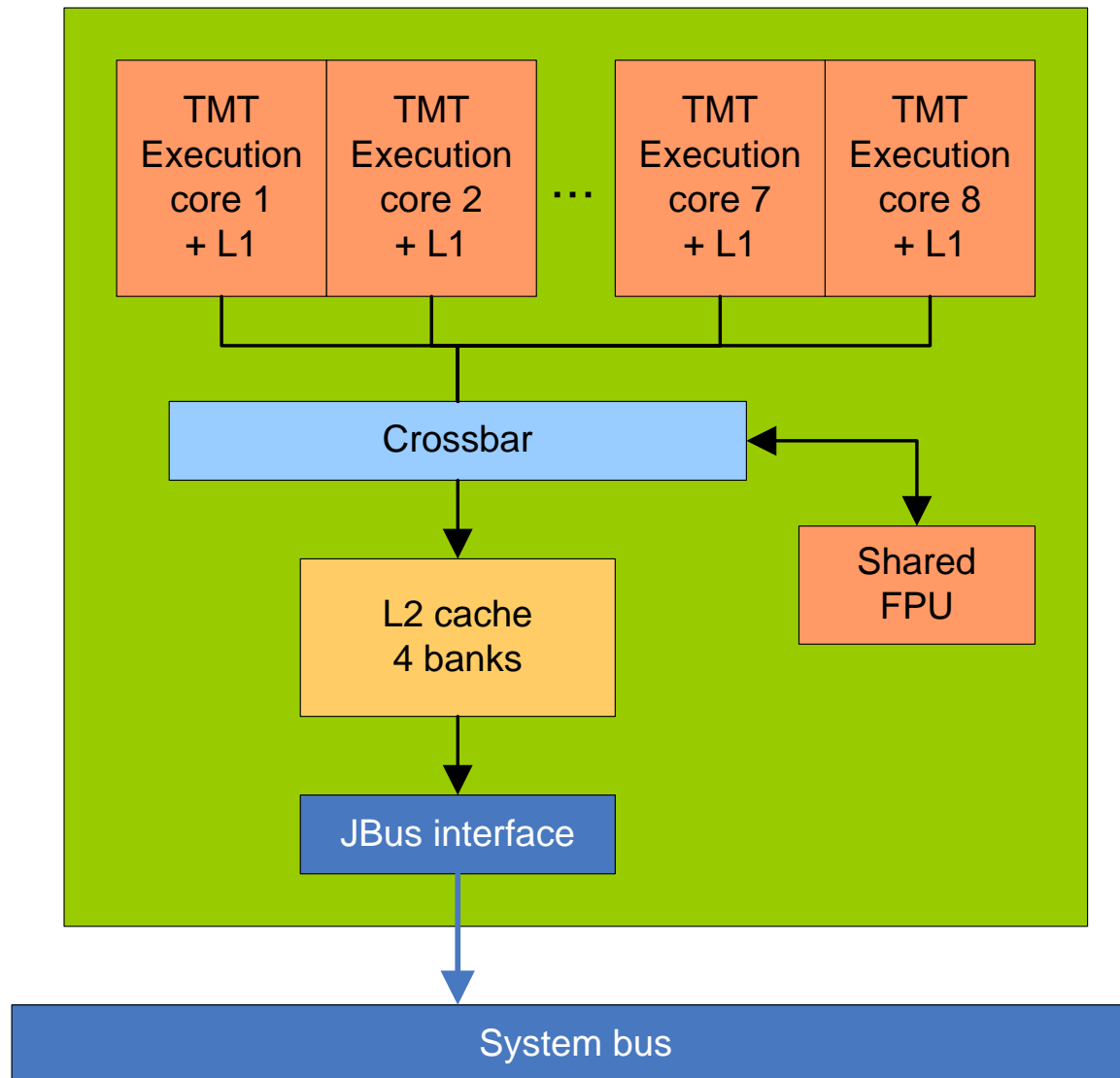
3 (“Tukwila”)

- Release: ~2008
- Estimated 40GFlops / socket
- 24MB L2 cache
- Next generation interconnect (QPI)
- 30% improvement over “Montecito” (Itanium 2)
- Socket compatible with Xeon



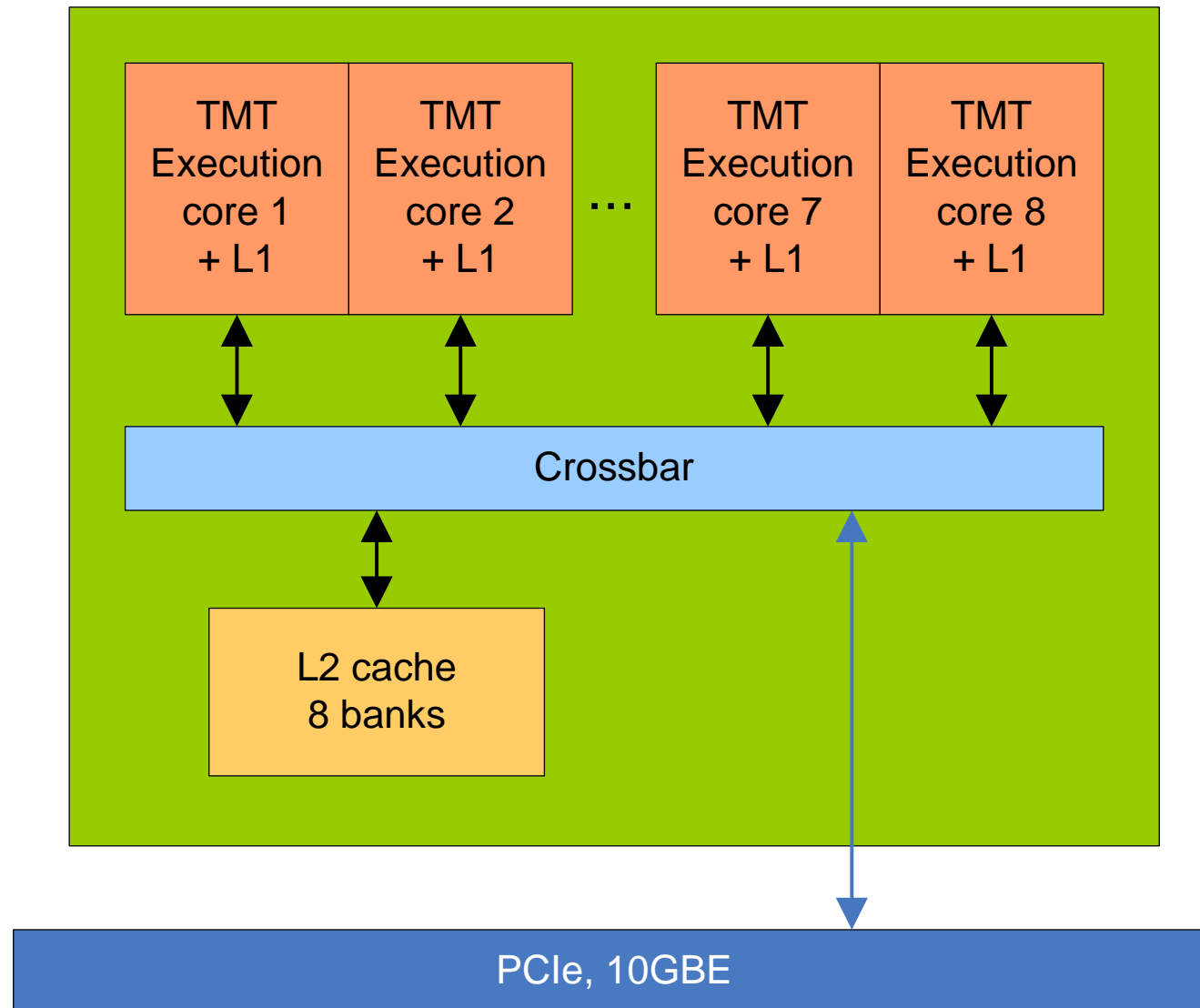
Multi-core architectures – UltraSPARC

T1



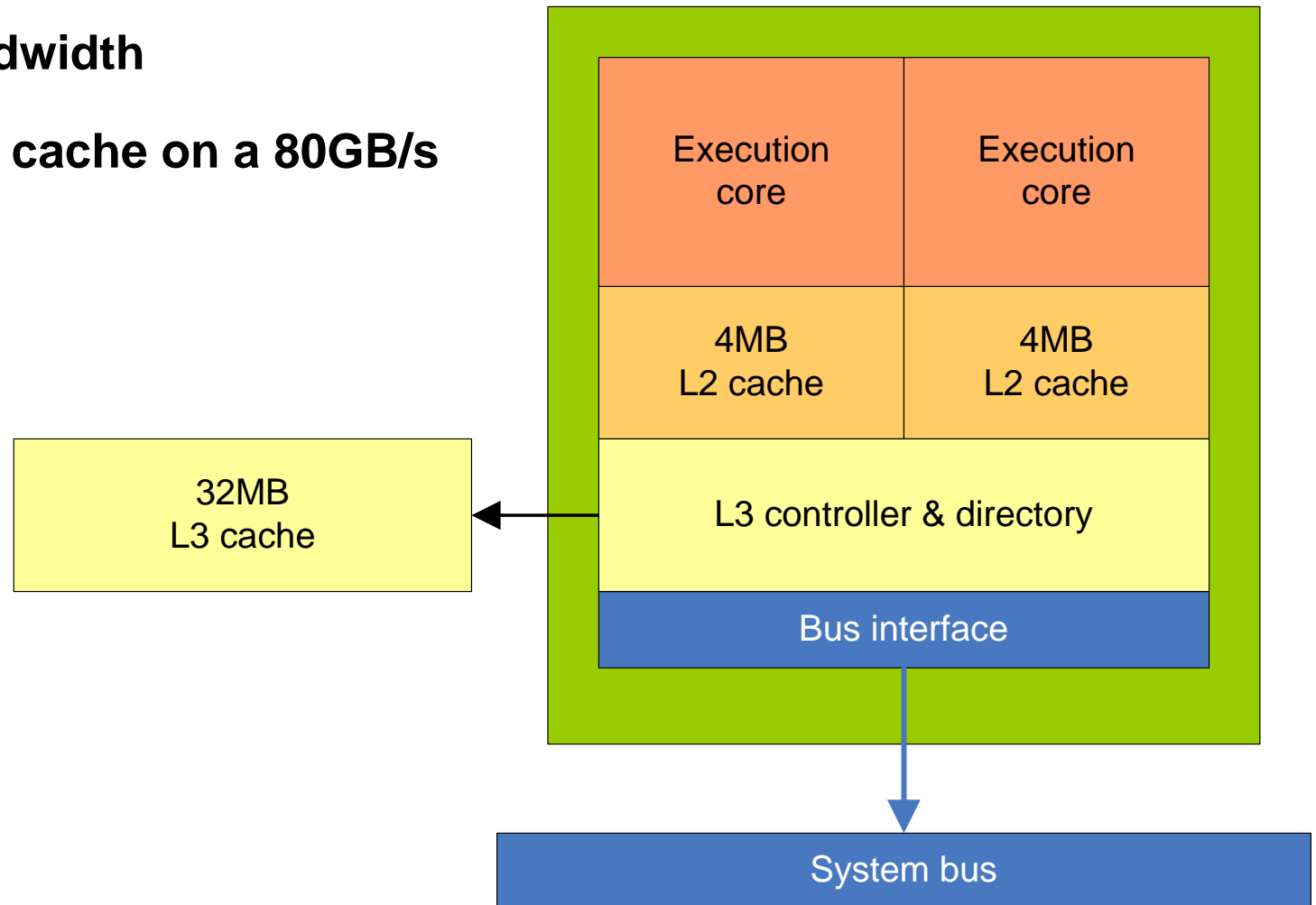
Multi-core architectures – UltraSPARC

T2



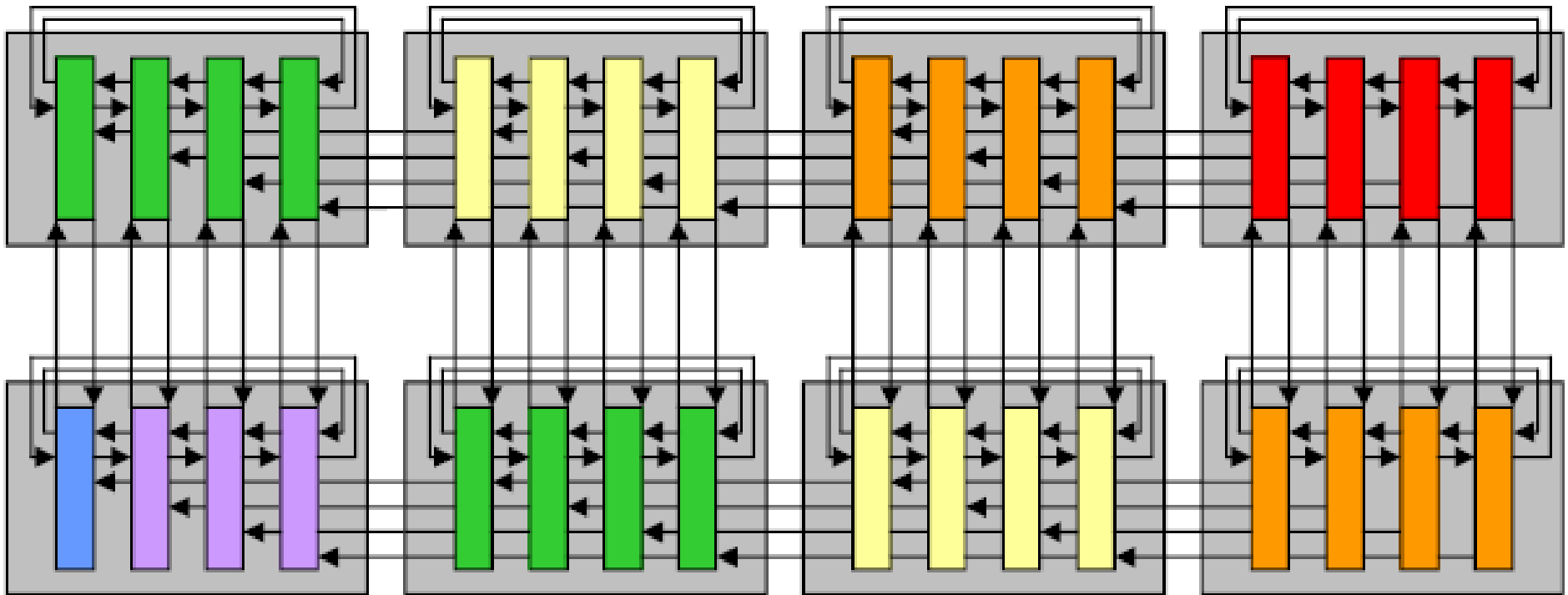
Multi-core architectures – IBM Power6

- 4.7GHz top frequency
- 500GB/s of bandwidth
- 32MB off-die L3 cache on a 80GB/s bus

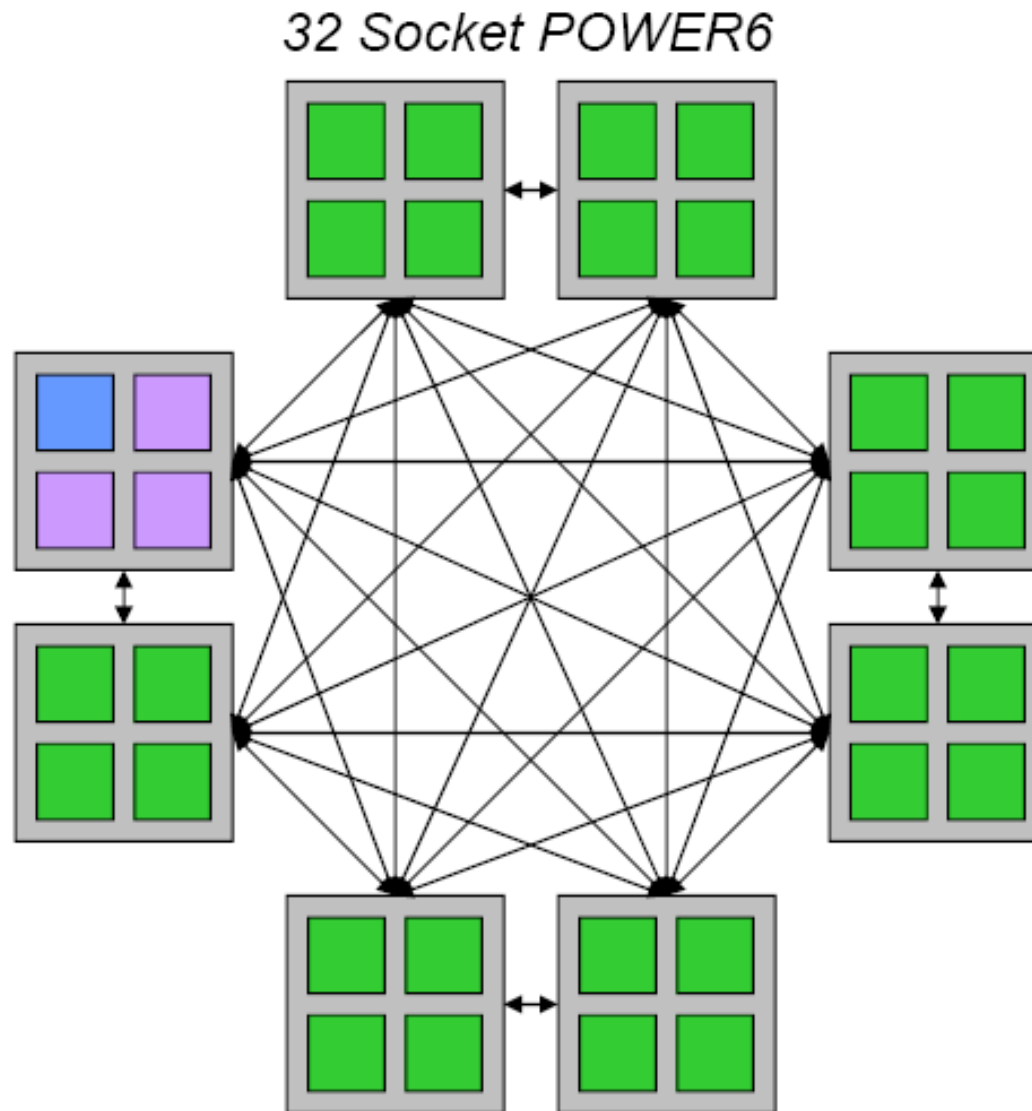


Multi-core architectures – Power5+ interconnects

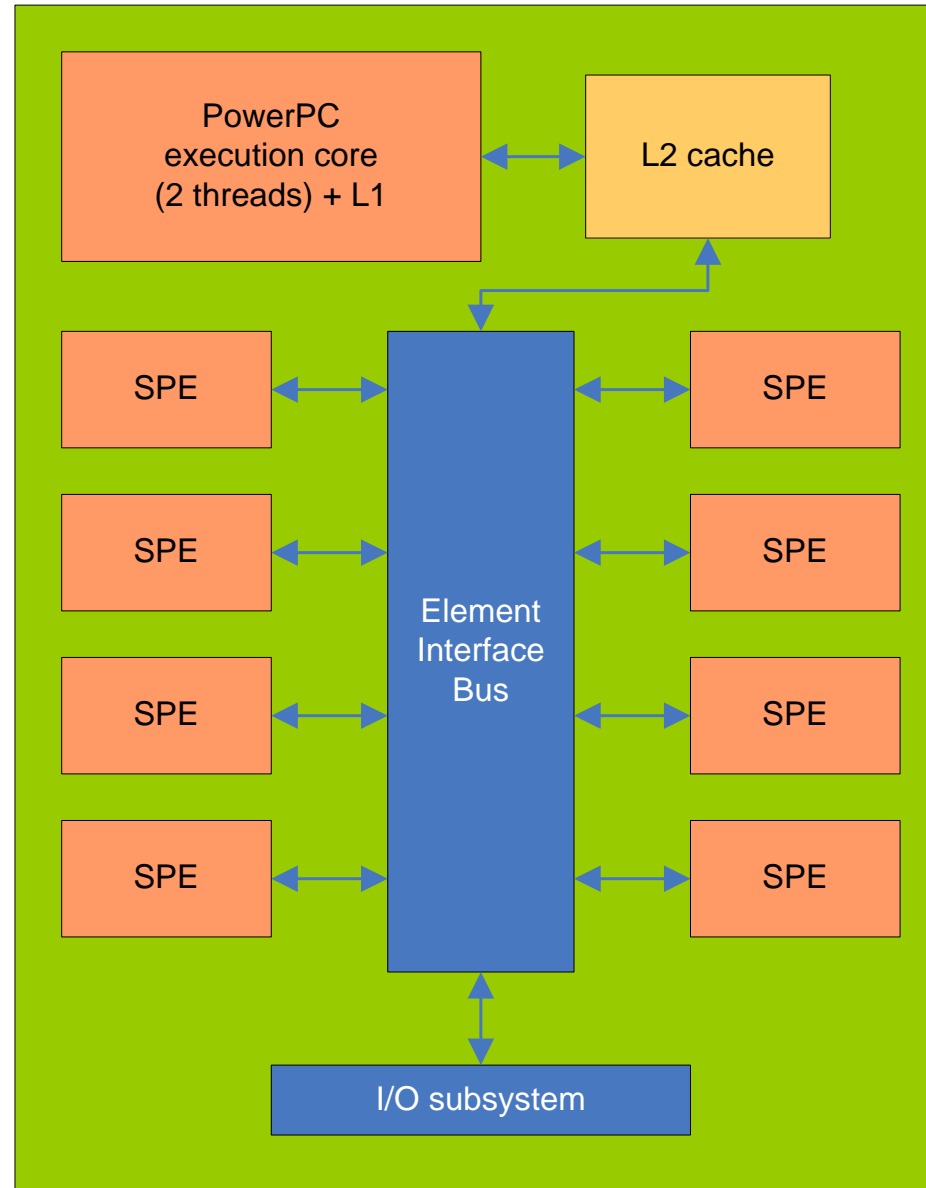
32 Socket POWER5+



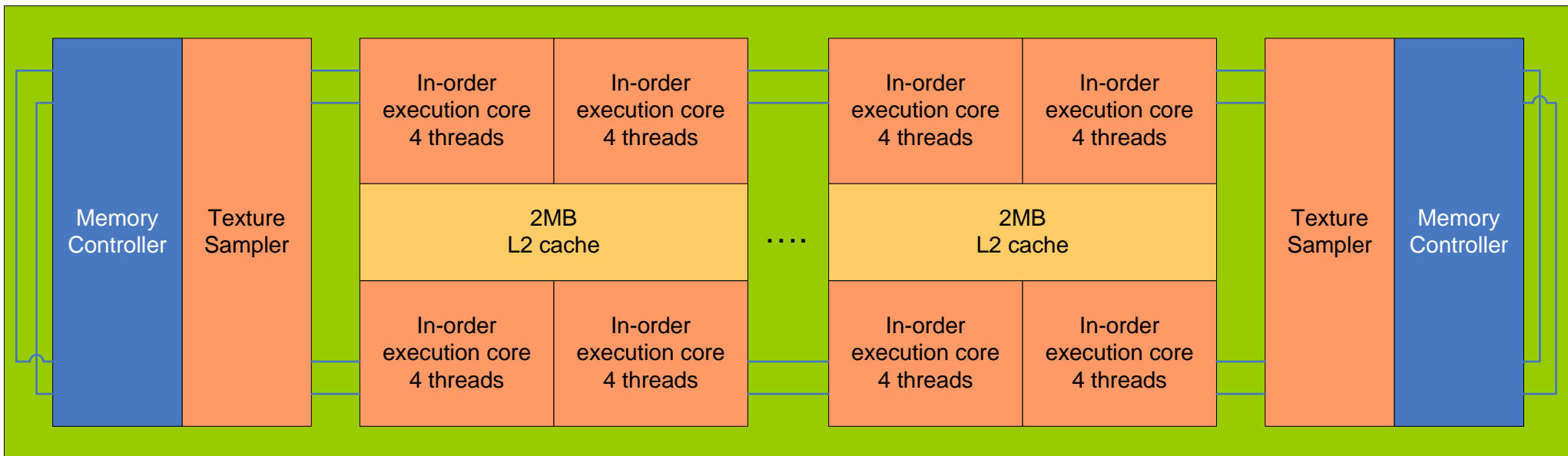
Multi-core architectures – Power6 interconnects



Interesting architectures – IBM CELL



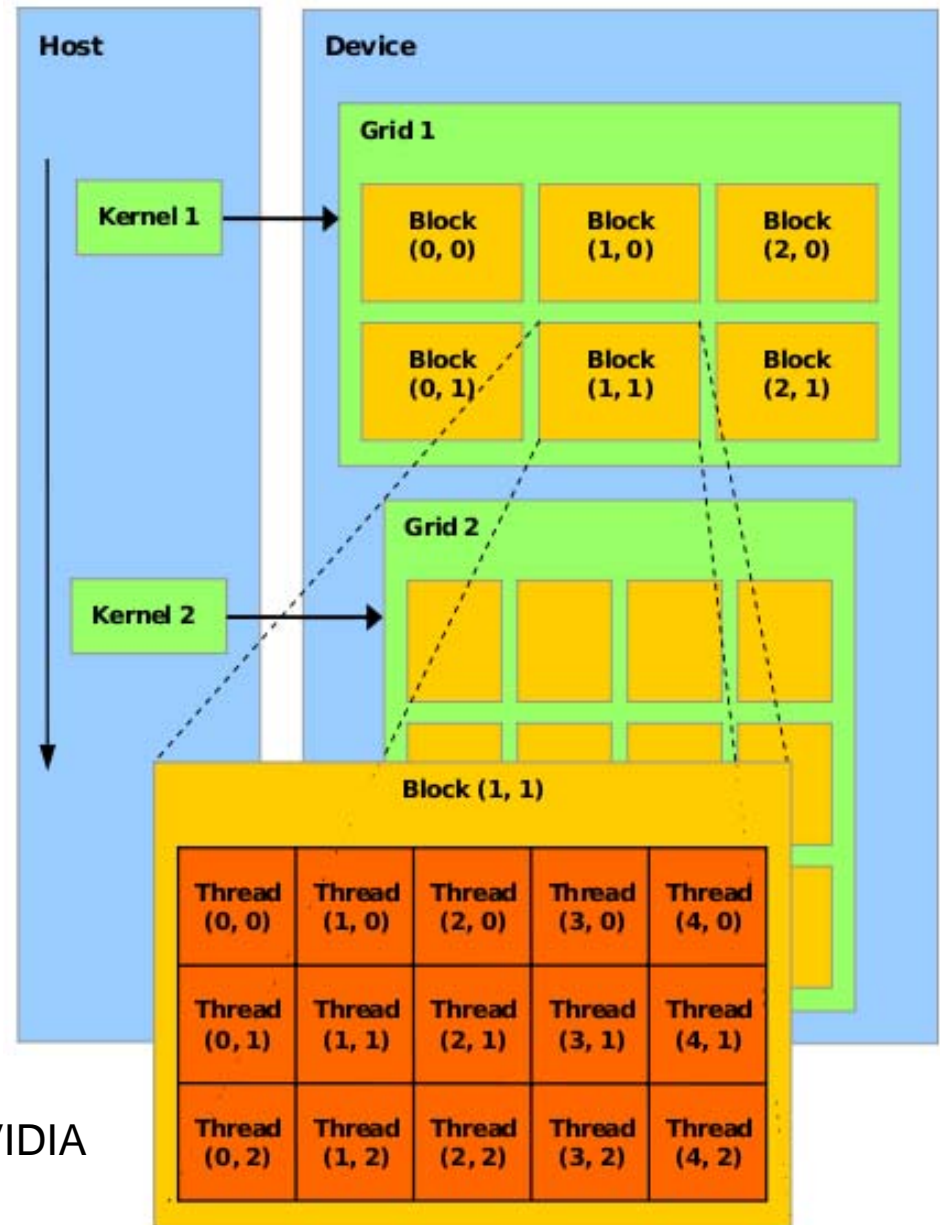
Multi-core architectures – Intel Larrabee



SPECULATIVE INFORMATION. Source: ArsTechnica

Multi-core architectures – NVIDIA G80

- Moved away from traditional GPU design
- 128 stream processors
- 330 GFLOPS peak
- Second generation: G92

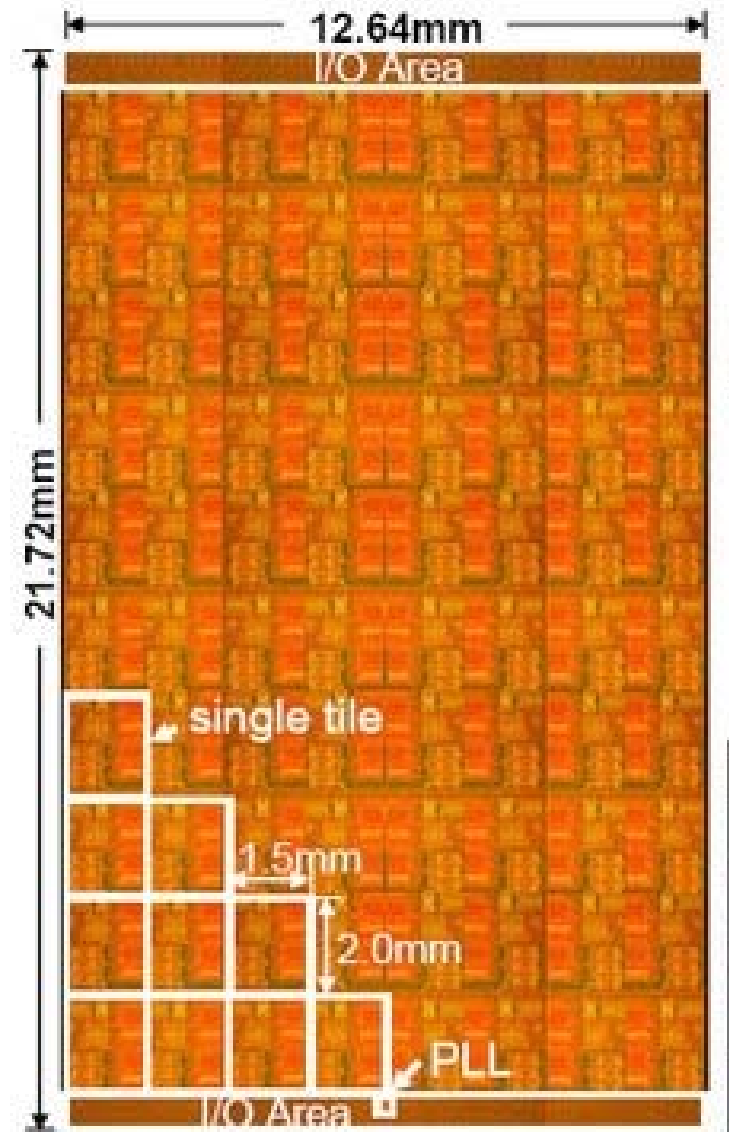


Graphics: NVIDIA

Multi-core architectures – Intel Polaris

(1)

- **80 cores**
- **Tiled (mesh) architecture**
- **Array area: 13mm x 28mm**
 - Single core: 2mm x 1.5mm
- **Modular, scalable design**
- **Fine grained power management**
- **Approximate performance:**
 - 1 TFLOP @ 50-60W
 - 1.5 TFLOPS @ ~100W
 - 2 TFLOPS @ ~200-250W

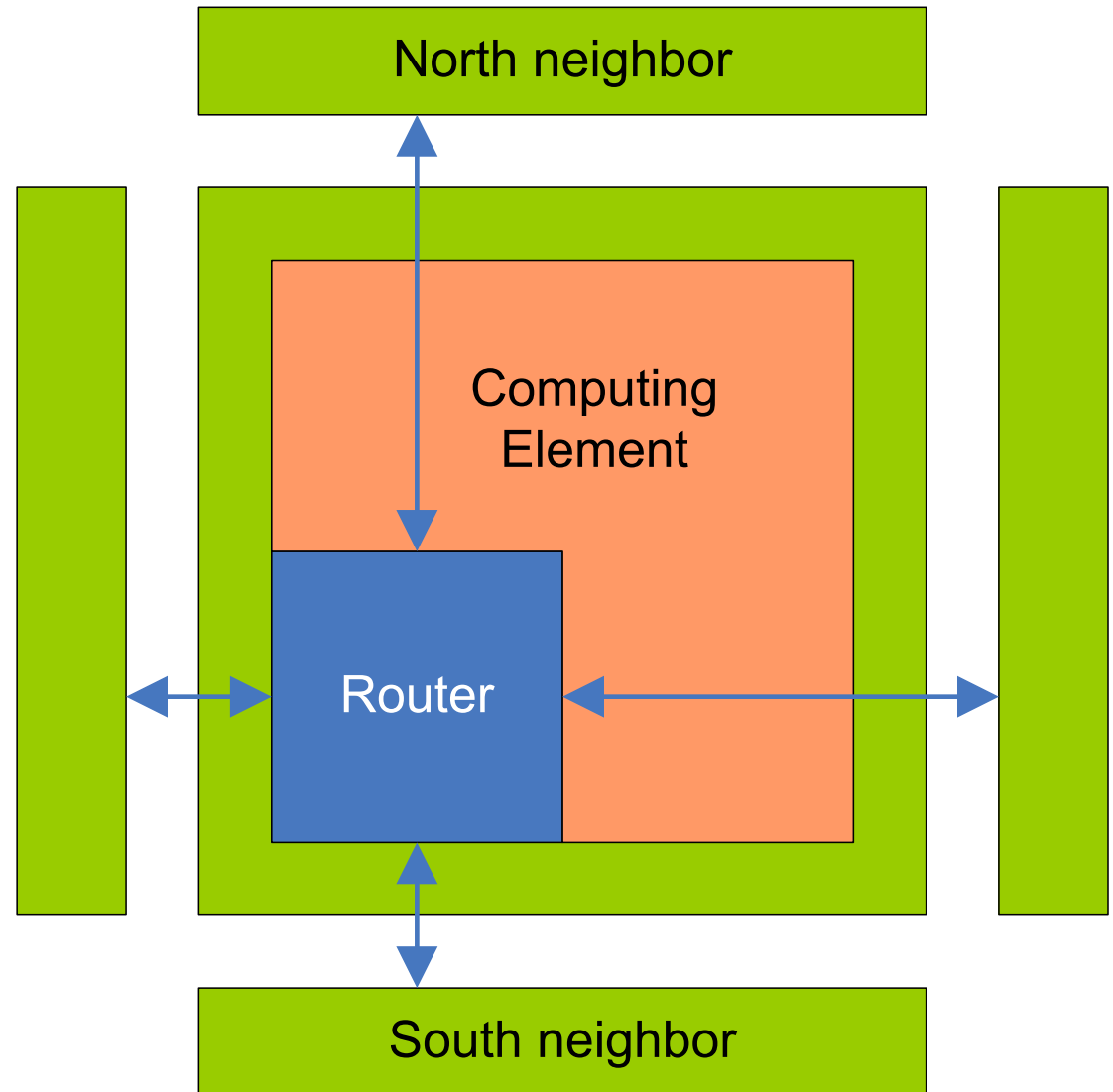


Data source: computerbase.de

Multi-core architectures – Intel Polaris

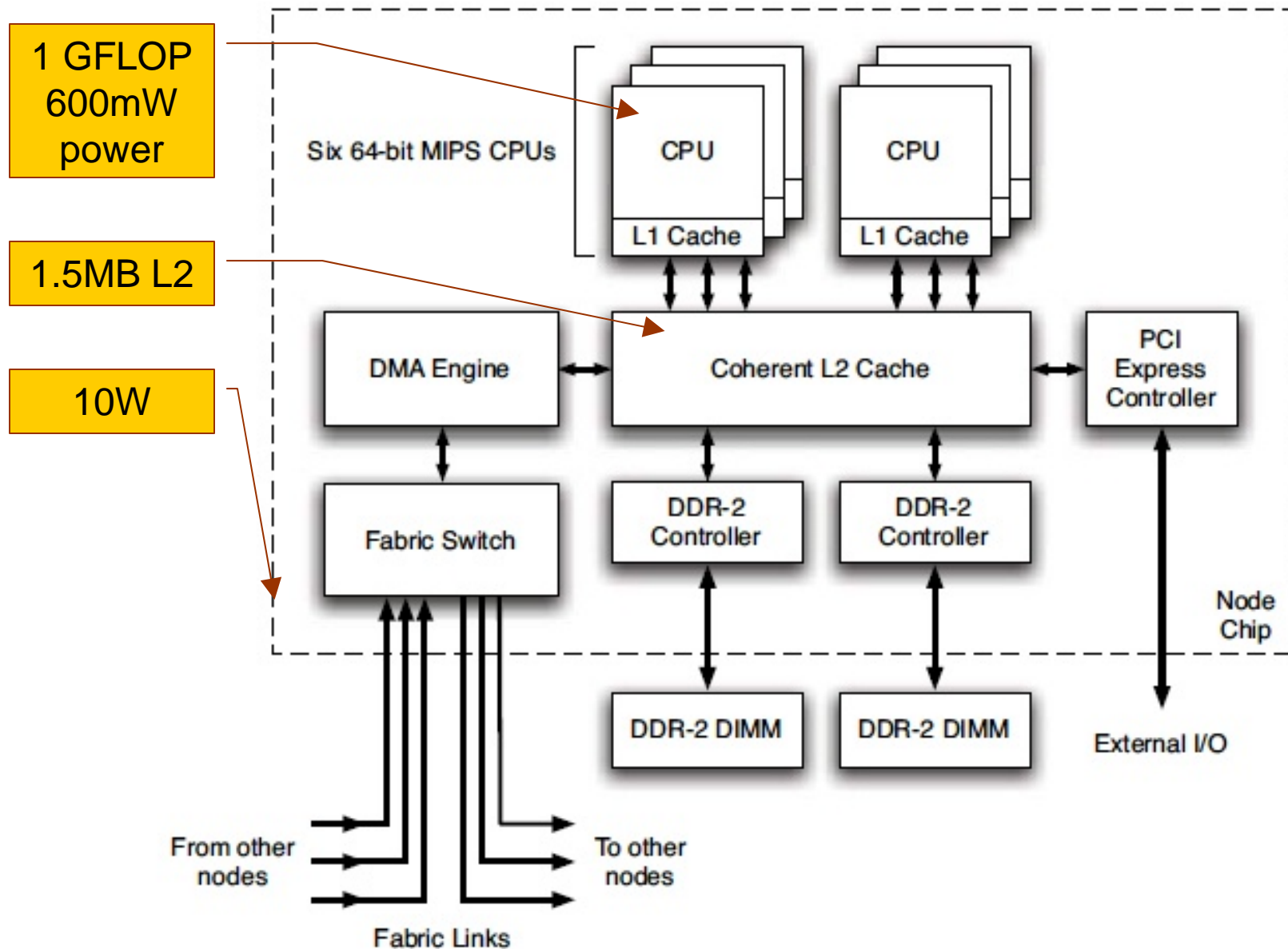
(2)

- **Core data:**
 - 2kB data memory
 - 3kB instruction memory
 - 32GBps interconnect
 - Tile area: 3mm²
 - Versatile, scalable design



Data source: computerbase.de

Interesting architectures – SiCorTex (1)



Interesting architectures – SiCorTex (2)

- 27 6-core nodes make up one blade
- **SC5832**
 - 36 blades
 - 5832 cores
 - 5.8 TFLOPS
 - 8 TB of DDR2 memory
 - The only computing system on the Top500 list with a single backplane
 - **18 kW**
 - ~\$2.5 M
- **SC648**
 - 0.648 TFLOP
 - 2kW
 - ~\$200 k

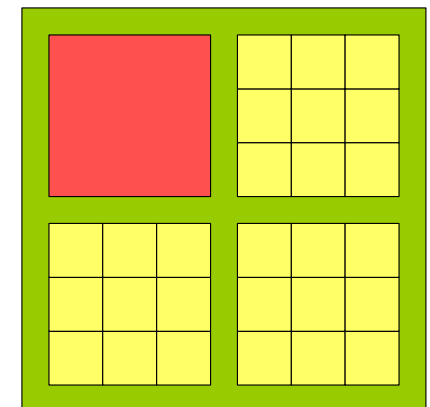
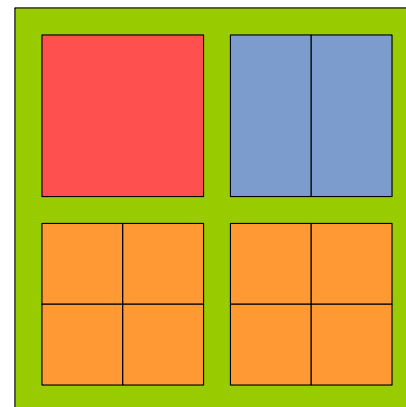
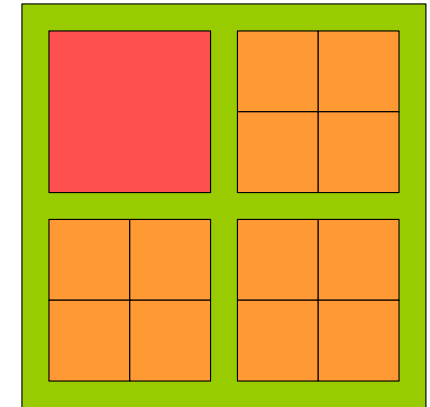
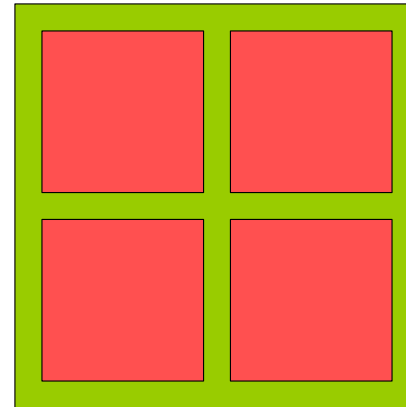


Interesting architectures – FPGAs

- **Programmable hardware**
 - Programmed using a low level hardware description language (commonly VHDL or Verilog)
 - Some higher level languages and methods are being developed
- **Heavily used in the industry, becoming popular in HPC**
- **Well suited for data streaming**
- **Common method: moving inner loops into very fast custom instructions**
- **Advantages**
 - Very fast
 - Can execute all implemented operations in parallel
- **More later**

Summary

- **Emerging trend – hybrid, heterogeneous solutions**
- **The future**
 - Large core designs?
 - Hybrid designs?
 - Small core designs?



Q&A