

Module 1: "Multi-core: The Ultimate Dose of Moore's Law"

Lecture 2: "Moore's Law and Multi-cores"

The Lecture Contains:

- ☰ Moore's Law
- ☰ Scaling Issues
- ☰ Multi-core
- ☰ Thread-level Parallelism
- ☰ Communication in Multi-core
- ☰ Tiled CMP (Hypothetical Floor-plan)
- ☰ Shared Cache CMP
- ☰ Niagara Floor-plan
- ☰ Implications on Software
- ☰ Research Directions
- ☰ References

◀ Previous Next ▶

Module 1: "Multi-core: The Ultimate Dose of Moore's Law"

Lecture 2: "Moore's Law and Multi-cores"

Moore's Law:

- Number of transistors on-chip doubles every 18 months
 - So much of innovation was possible only because we had transistors
 - Phenomenal 58% performance growth every year
- Moore's Law is facing a danger today
 - Power consumption is too high when clocked at multi-GHz frequency and it is proportional to the number of switching transistors
- Wire delay doesn't decrease with transistor size

Scaling Issues:

- Hardware for extracting ILP has reached the point of diminishing return
 - Need a large number of in-flight instructions
 - Supporting such a large population inside the chip requires power-hungry delay-sensitive logic and storage
 - Verification complexity is getting out of control
- How to exploit so many transistors?
 - Must be a de-centralized design which avoids long wires

Multi-core:

- Put a few reasonably complex processors or many simple processors on the chip
 - Each processor has its own primary cache and pipeline
 - Often a processor is called a core
 - Often called a chip-multiprocessor (CMP)
- Did we use the transistors properly?
 - Depends on if you can keep the cores busy
 - Introduces the concept of thread-level parallelism (TLP)



Thread-level Parallelism:

- Look for concurrency at a granularity coarser than instructions
 - Put a chunk of consecutive instructions together and call it a thread (largely wrong!)
 - Each thread can be seen as a “dynamic” subgraph of the sequential control-flow graph: take a loop and unroll its graph
 - The edges spanning the subgraphs represent data dependence across threads (the spanning control edges are usually converted to data edges through suitable transformations)
 - The goal of parallelization is to minimize such edges
 - Threads should mostly compute independently on different cores; but need to talk once in a while to get things done!
- Parallelizing sequential programs is fun, but often tedious for non-experts
 - So look for parallelism at even coarser grain
 - Run multiple independent programs simultaneously
 - Known as multi-programming
 - The biggest reason why quotidian Windows fans would buy small-scale multiprocessors and multi-core today
 - Can play games while running heavy-weight simulations and downloading movies
 - Have you seen the state of the poor machine when running anti-virus?

Communication in Multi-core:

- Ideal for shared address space
 - Fast on-chip hardwired communication through cache (no OS intervention)
 - Two types of architectures
 - Tiled CMP: each core has its private cache hierarchy (no cache sharing); Intel Pentium D, Dual Core Opteron, Intel Montecito, Sun UltraSPARC IV, IBM Cell (more specialized)
 - Shared cache CMP: Outermost level of cache hierarchy is shared among cores; Intel Woodcrest (server-grade Core duo), Intel Conroe (Core2 duo for desktop), Sun Niagara, IBM Power4, IBM Power5

Module 1: "Multi-core: The Ultimate Dose of Moore's Law"

Lecture 2: "Moore's Law and Multi-cores"

Tiled CMP (Hypothetical Floor-plan):

 Previous Next 

Module 1: "Multi-core: The Ultimate Dose of Moore's Law"

Lecture 2: "Moore's Law and Multi-cores"

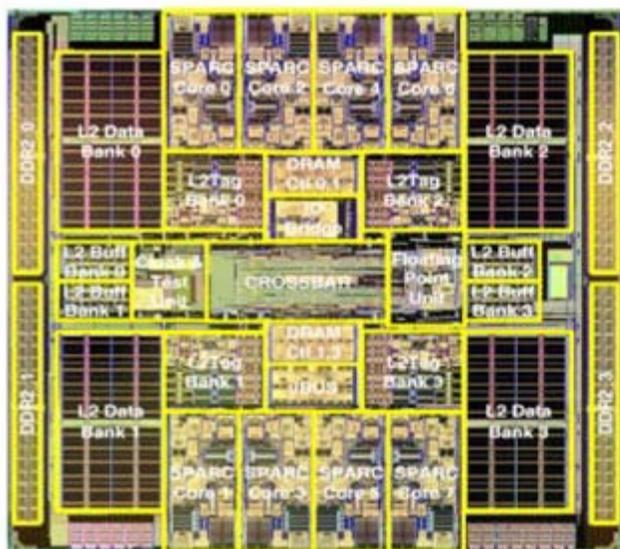
Shared Cache CMP:

 **Previous** **Next** 

Module 1: "Multi-core: The Ultimate Dose of Moore's Law"

Lecture 2: "Moore's Law and Multi-cores"

Niagara Floor-plan:

**Features**

- Eight 64 bit multithreaded SPARC Cores
- Shared 3 MB L2 Cache
- 16 KB Icache per Core
- 8 KB Dcache per Core
- Four 144b DDR2 DRAM interface (400 Mts)
- 3.2 GB/s JBUS I/O
- Crypto public key (RSA)
- Extensive RAS

Technology

- 90 nm CMOS Process
- 9 LM Copper interconnect
- Power: 63 watts@1.2 Ghz
- Die Size: 378 mm²
- 279M Transistors
- Package: Flip-chip ceramic LGA (1933 pins)

Implications on Software:

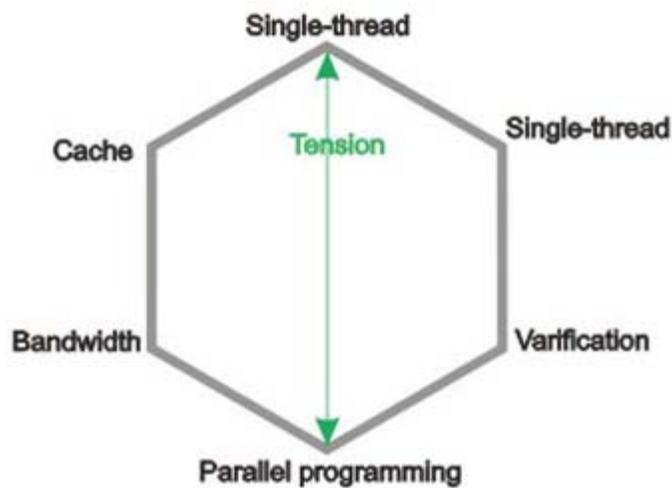
- A tall memory hierarchy
 - Each core could run multiple threads
 - Each core in Niagara runs four threads
 - Within core, threads communicate through private cache (fastest)
 - Across cores communication happens through shared L2 or coherence controller (if tiled)
 - Multiple such chips can be connected over a scalable network
 - Adds one more level of memory hierarchy
- A very non-uniform access stack

Module 1: "Multi-core: The Ultimate Dose of Moore's Law"

Lecture 2: "Moore's Law and Multi-cores"

Research Directions:

- Hexagon of puzzles
 - Running single-threaded programs efficiently on this sea of cores
 - Managing energy envelope efficiently
 - Allocating shared cache efficiently
 - Allocating shared off-chip bandwidth and memory banks efficiently
 - Making parallel programming easy
 - Transactional
 - Speculative parallelization
 - Verification of hardware and parallel software and tolerate faults



References:

- A good reading is Parallel Computer Architecture by Culler, Singh with Gupta
 - Caveat: does not talk about multi-core, but introduces the general area of shared memory multiprocessors
- Papers
 - Check out the most recent issue of Intel Technology Journal
 - <http://www.intel.com/technology/itj/>
 - <http://www.intel.com/technology/itj/archive.htm>
 - Conferences: ASPLOS, ISCA, HPCA, MICRO, PACT
 - Journals: IEEE Micro, IEEE TPDS, ACM TACO