Termites, Chainsaws, and Bulldozers: Managing Software Complexity in the Multicore Transition

Doug Burger

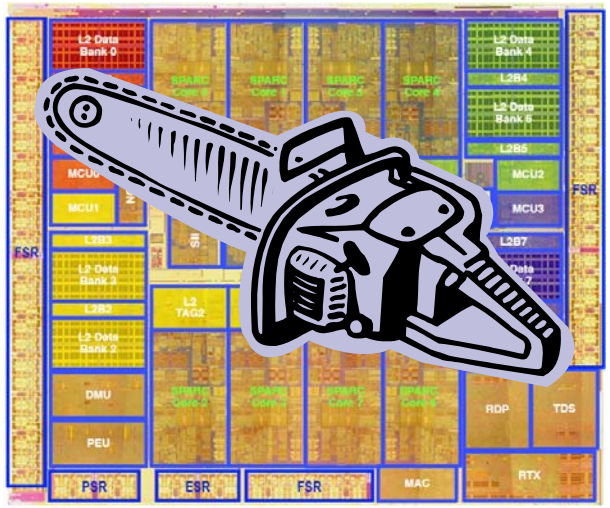
September 25, 2007

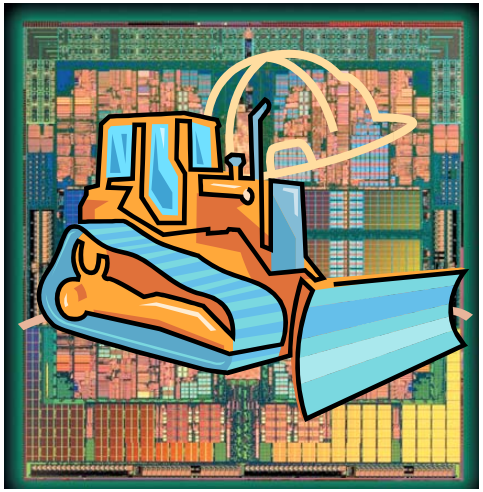
Department of Computer Sciences
The University of Texas at Austin

Critical Multicore Design Issues

- Programmability is hard enough, but there are exacerbating issues
  - Power efficiency
  - General purpose capabilities (for many markets)
  - Resiliency
  - Delivering memory/interconnect bandwidth to cores
- The granularity (termites, chainsaws, bulldozers) is a key question
  - Poor alignment with an application results in wasted power and performance
- Spatial planning will be key
  - Many technologically imposed boundaries (chip, MCM, board interfaces)
  - Will need to perform multiple decompositions for different levels of the hierarchy
  - What software hooks will make that possible?
  - Proposed metric: km/s of operand routing per chip

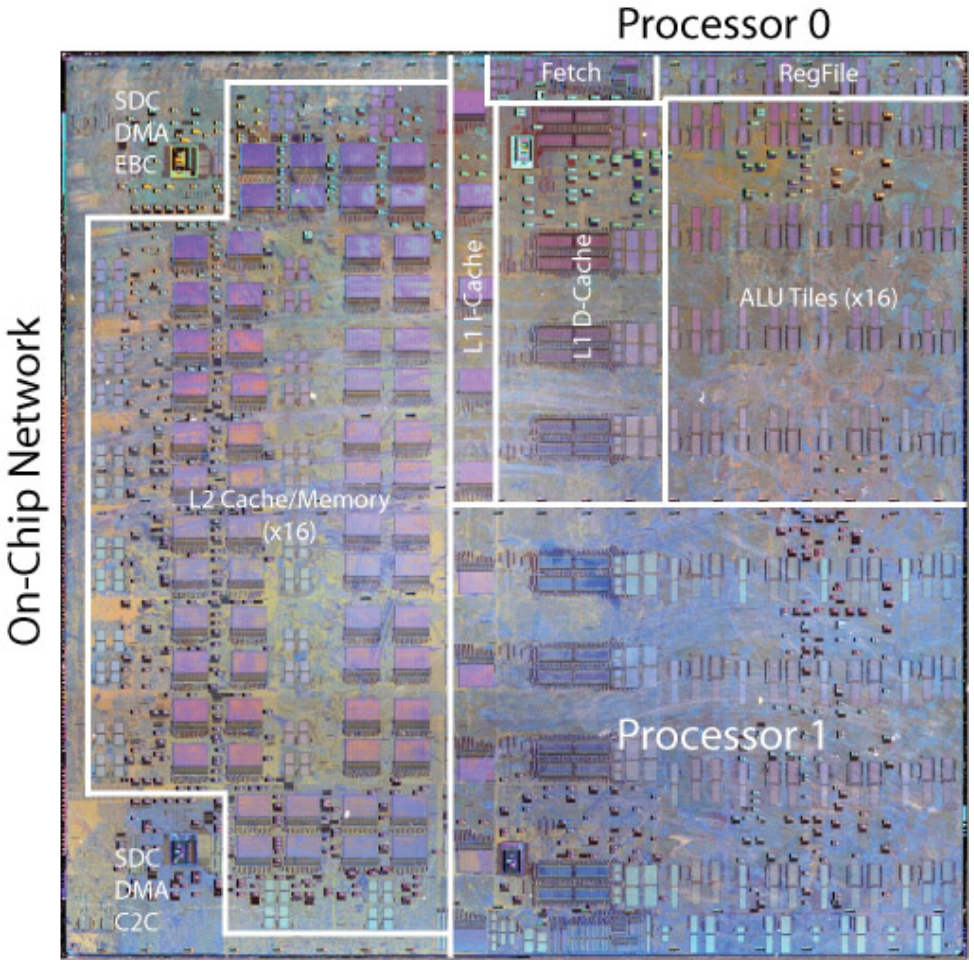
Termites, Chainsaws, and Bulldozers





- Termites and chainsaws
  - Good performance when tasks are abundant
  - But: burden of communication, synchronization, load balancing
- Multiprocessor performance starts with a powerful uniprocessor
  - Bulldozers serve a broader range of applications
  - Better to build ten 10x processors than two hundred 1x processors

TRIPS: A Scalable Uniprocessor

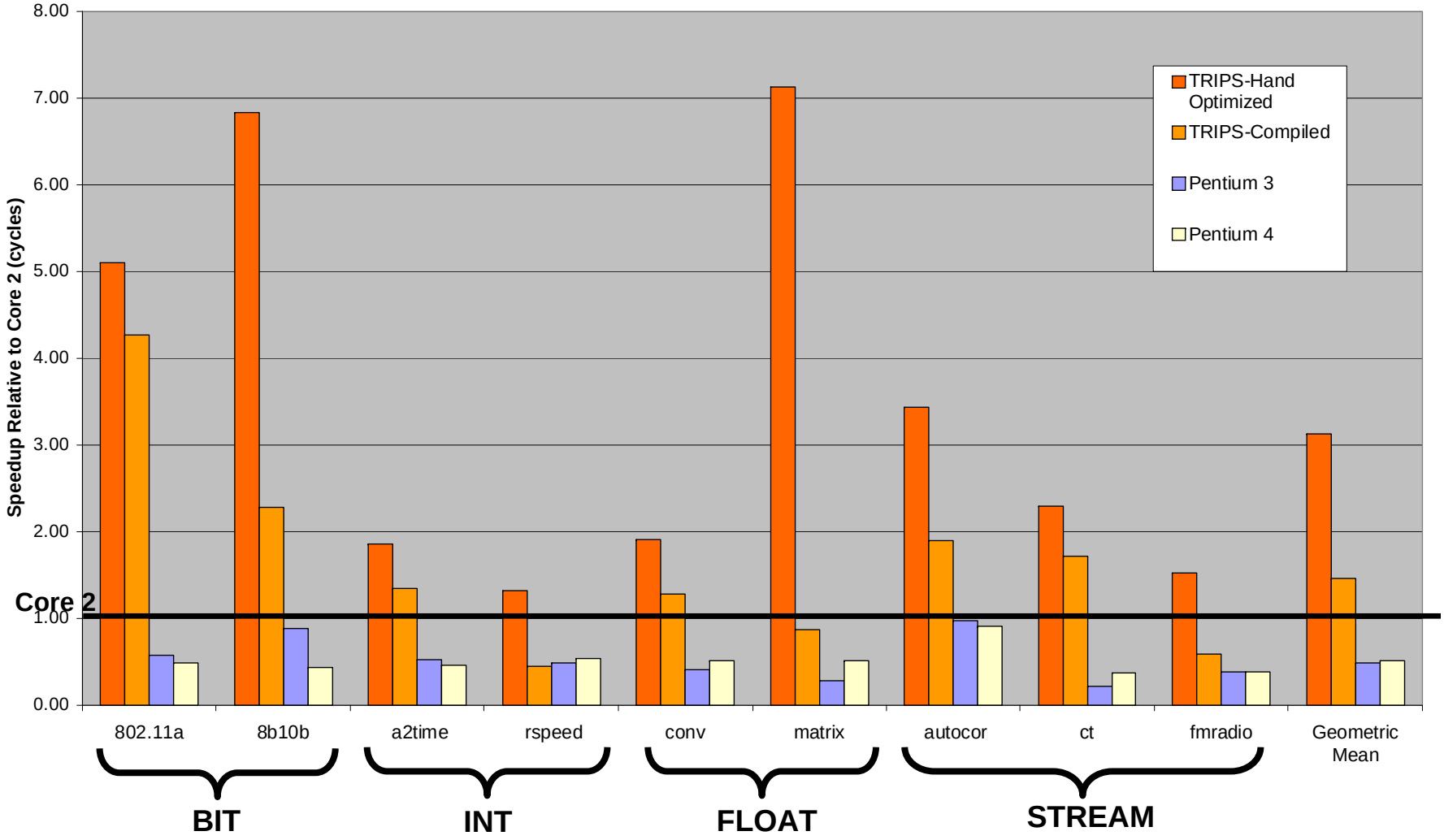


~75mm² per core at 130nm

~20mm² at 65nm

- Powerful uniprocessor
  - 16-issue per core
  - 4 LD/ST per cycle
  - 1024 instruction window
  - Data-driven ISA and execution
  - Performance scales beyond existing uniprocessor cores
- Tile-based design
  - Heterogeneous small tiles
  - Interconnected with on-chip networks
  - Distributed protocols for fetch/execute/commit
  - Fully functional first silicon

TRIPS Single-Processor Performance

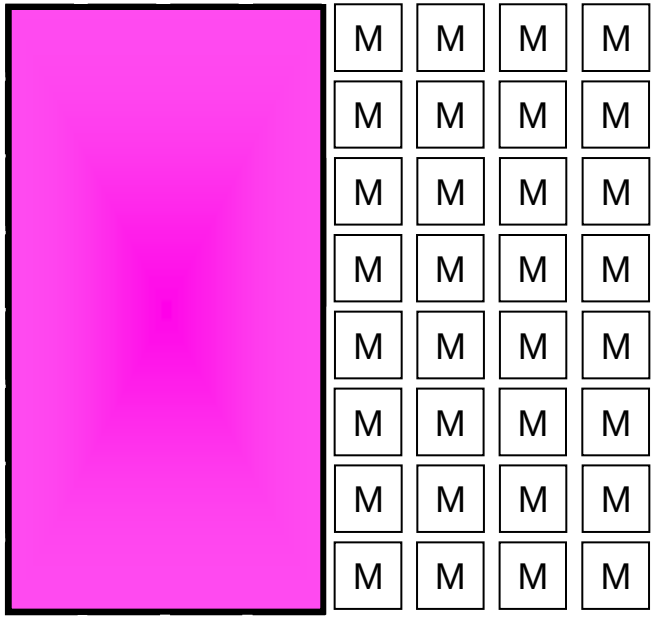


TRIPS Observations

- Strengths
  - Tiled approach has great design productivity
  - Existing applications run “out-of-the-box”
    - Recompilation required, but no parallelization
  - Performance good when applications have concurrency
- Inefficiencies on:
  - Applications with limited concurrency
    - 16-issue is overkill for many applications
  - Threaded parallelism
- What if we could have the best of both small and large cores, on demand?

Composable Lightweight Processors

- Concept
  - Building block is small core
  - Aggregate on demand into larger logical processors
  - Same binary can run on any configuration
- Example
  - 32 2-wide out-of-order homogeneous cores (TRIPS ISA)



- Configuration #1: Many threads
- Configuration #2: Power sensitive
- Configuration #3: Limited-concurrency threads
- Configuration #4: One critical thread

Summary

- Termites, Chainsaws, and Bulldozers each have their place
  - But, setting the core granularity at design time limits the range of applications that execute well
- An ideal general purpose multicore architecture
  - Provides performance and power efficiency to a wide range of programs
  - Matches granularity of application concurrency (ILP/TLP) with concurrency of cores
  - Can move across the spectrum of termites to bulldozers
- Current directions: uniprocessor/multiprocessor hybrids
  - Rather than force the programmer to parallelize all applications, provide only the analysis necessary for the architecture