CS533: Chip Multiprocessors

Josep Torrellas

University of Illinois in Urbana-Champaign

March 3, 2015

Introduction

- Increasing chip density
  - Over 1 billion transistors/chip already
- Conventional dynamic-issue superscalar approach
  - ILP limited in single thread of control
  - Large issue window $\rightarrow$ more complexity, less returns
- Optimal usage of silicon space needed
  - Simultaneous multithreading (SMT)
  - Chip multiprocessor (CMP)

SMT Motivation

- Wide superscalar is the way to go
  - Must continue to improve single-program performance
- Multiple programs share fetch & issue bandwidth
  - Thread-level parallelism (TLP) improves throughput of wide superscalar
  - Can still exploit high ILP in a single program

CMP Motivation

K. Olukotun et al. "The case for a single-chip multiprocessor" *ASPLOS*, 1996

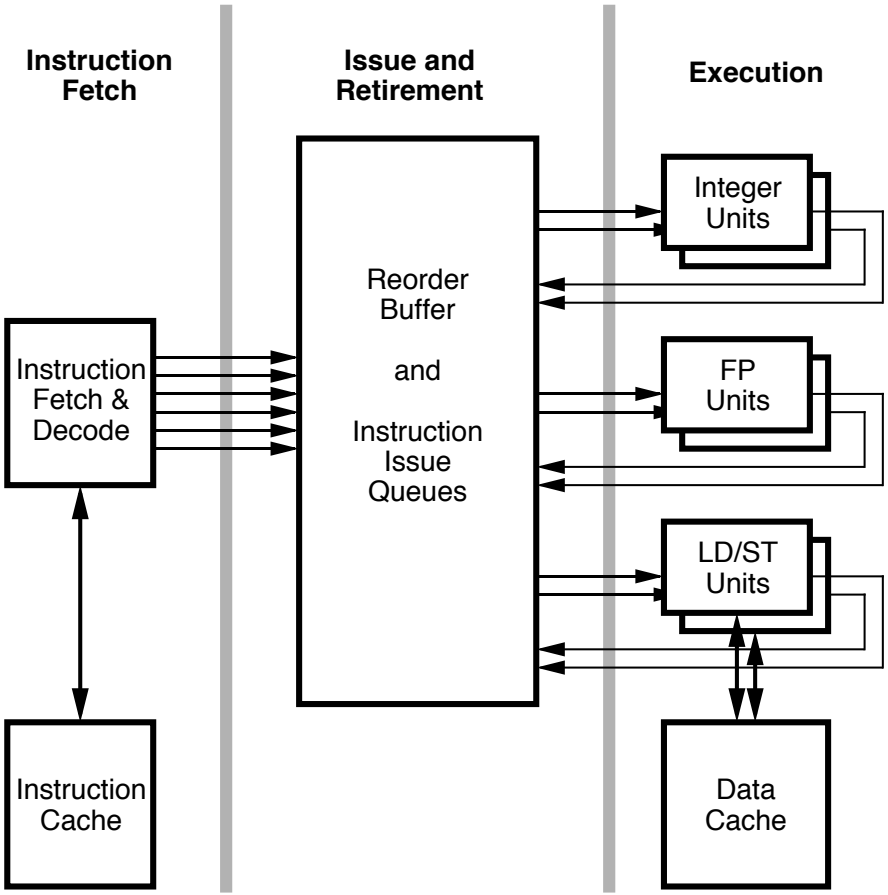
- Wide superscalar is not the way to go
  - Trades fast clock for minor IPC gain
  - Design and verification complexity
- Multipe simple processors (e.g., 2-way or 4-way superscalar)
  - Exploit TLP
  - Moderate ILP plus fast clock

Aggressive Superscalars

- Multiple instruction issue, dynamic scheduling, speculative execution, non-blocking caches, ...
- Trend
  - Wider instruction issue
  - Larger amounts of speculative execution
- However,
  - Limited amounts of ILP
  - Fundamental circuit limitations

$\Longrightarrow$ Better resource utilization: Multiple processors on a chip

Dynamic Superscalar CPU



- Instruction queue often implemented as multiple instruction queues for different type of instructions
- Stages:
  - Fetch
  - Issue
  - Execute

Fetch Phase

- Goal: Present a large window of decoded instructions
- Constraints:
  1. Mispredicted branches
  2. Instruction misalignment
  3. Cache misses

Fetch Phase (continued)

1. Mispredicted branches
   - Branch predictor buffers (e.g., 64 kbits)
     - Selective branch predictor
     - Reduce misprediction $< 5\%$
2. Instruction misalignment
   - Necessary to align a packet of instructions for decoder
   - If issue width > 4, then with high probability, need fetch across a branch for a single packet of instructions
     - Need fetch from 2 cache lines and merge
     - Scheme: Divide the instruction ncache into banks and fetch from multiple banks

Fetch Phase (continued)

3. Cache misses
   - High miss rate $\rightarrow$ limits ability to maintain large window
   - Can hide some cache miss latency by executing other instructions already in the window (dynamically scheduled CPUs)

$\Longrightarrow$ Overall: Fetch not limit

Issue Phase

- Packet of renamed instructions is inserted into instruction issue queue
- An instruction is issued when all operands are ready
- Renaming implemented with RAT, instruction queue and reorder buffer

Issue Phase (continued)

- Once instruction in instruction queue
  - Instructions that issue must update their dependences
  - Many comparators (e.g., HP-PA8000: 20% die area)
- Also: large window to find independent instructions
  - Size of instruction issue queue is large
  - Need broadcast of tags $\rightarrow$ **Wires are slow**

$\Longrightarrow$ Overall: Instruction issue queue wil limit cycle time

Execution Phase

- Operand values: fetched from register file or bypassed from earlier instructions
- Wide superscalar has probelms with
  - Register File
    - Larger to accommodate more rename registers
    - Many ports
    - Complexity $\approx$ #ports^2 $\approx$ issue width^2
  - Bypass logic
    - Complexity $\approx$ #execution units^2
    - **Wire delay**
  - Functional Units
    - More ports needed to data cache

Single-Chip Multiprocessor

- Technology push
  - Benefits of wide issue are limited
  - Decentralized microarchitecture: easier to build several simple fast processors than one complex processor
- Application pull
  - Applications exhibit parallelism at different granularities
  - $<$ 10 instructions/cycle (INT applications)
  - $>$ 40 instructions/cycle (loops in FP applications)

CMP

- INT applications
  - Use moderate issue processor (e.g., 2-way or 4-way) with very high clock rate
- FP applications
  - need compiler to parallelize
  - If cannot parallelize or serial section, CMP runs slow

Microarchitectures Compared

- SS: 6-way superscalar (@ 500MHz)
- CMP: Four 2-way superscalars (@ 500 MHz)

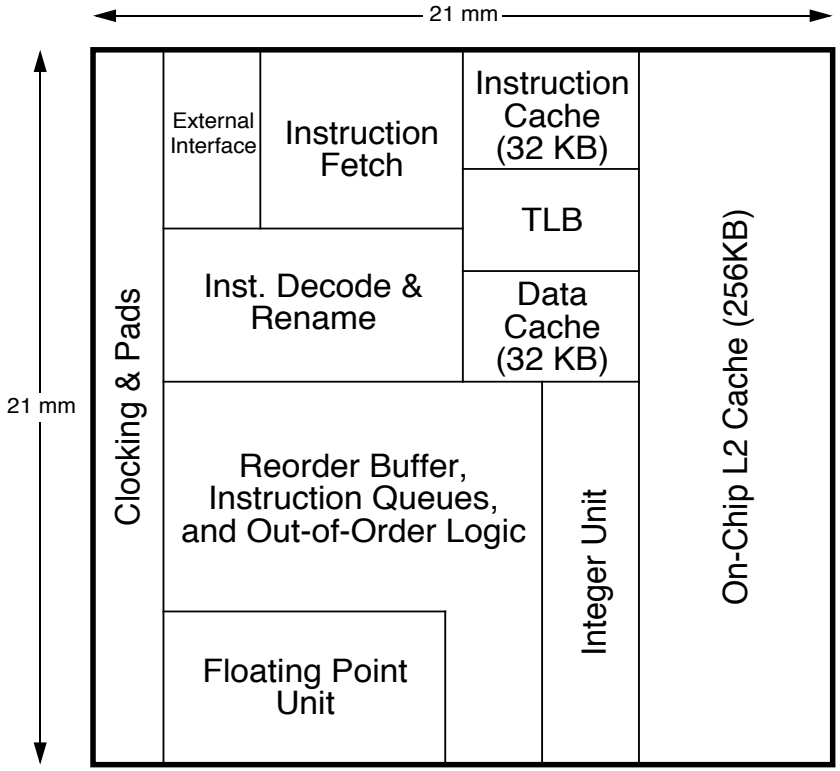


Figure 2. Floorplan for the six-issue dynamic superscalar microprocessor.

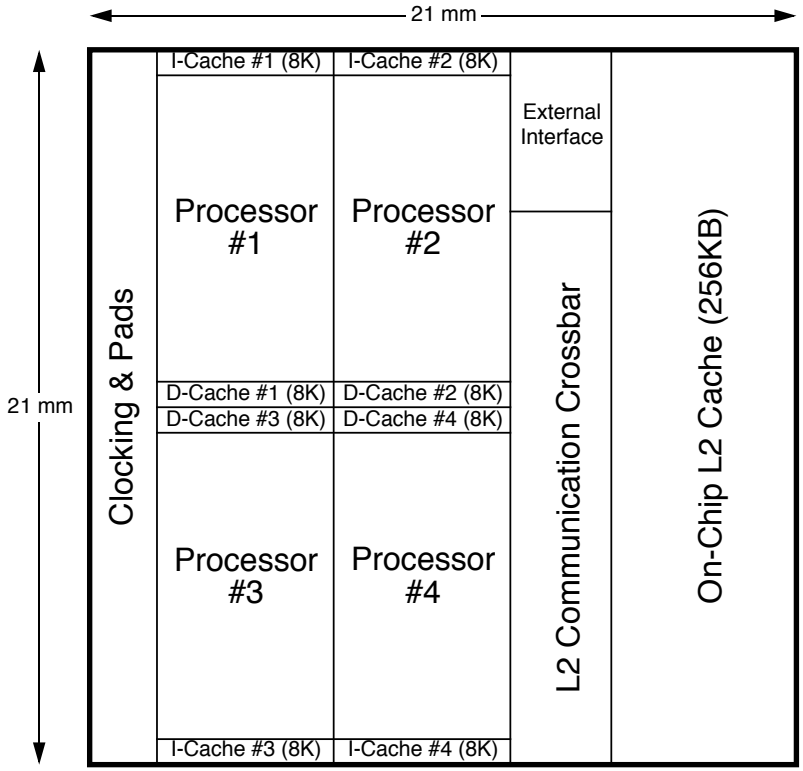
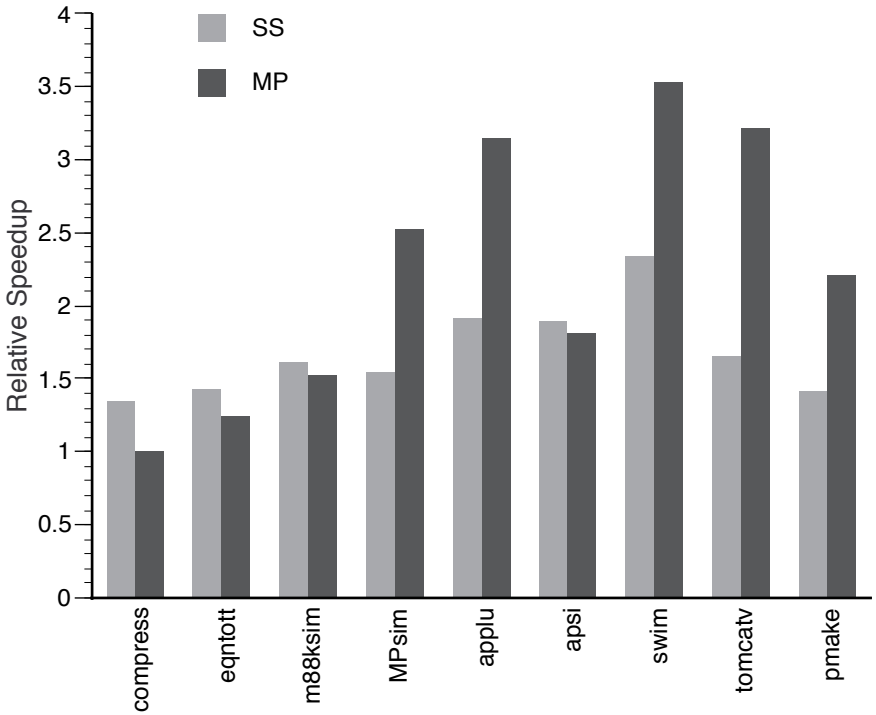


Figure 3. Floorplan for the four-way single-chip multiprocessor.

Performance Comparison



- ILP only
  - SS is 30% better than CMP
- ILP & fine grain threads
  - SS and CMP comparable
- ILP & coarse grain threads
  - CMP is 1.5-2$\times$ better