

Chip Multiprocessors

Josep Torrellas

CS533

Introduction

- Increasing chip density
 - 1 billion transistors/chip soon
- Conventional dynamic-issue superscalar approach
 - ILP limited in single thread of control
 - Large issue window → 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

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

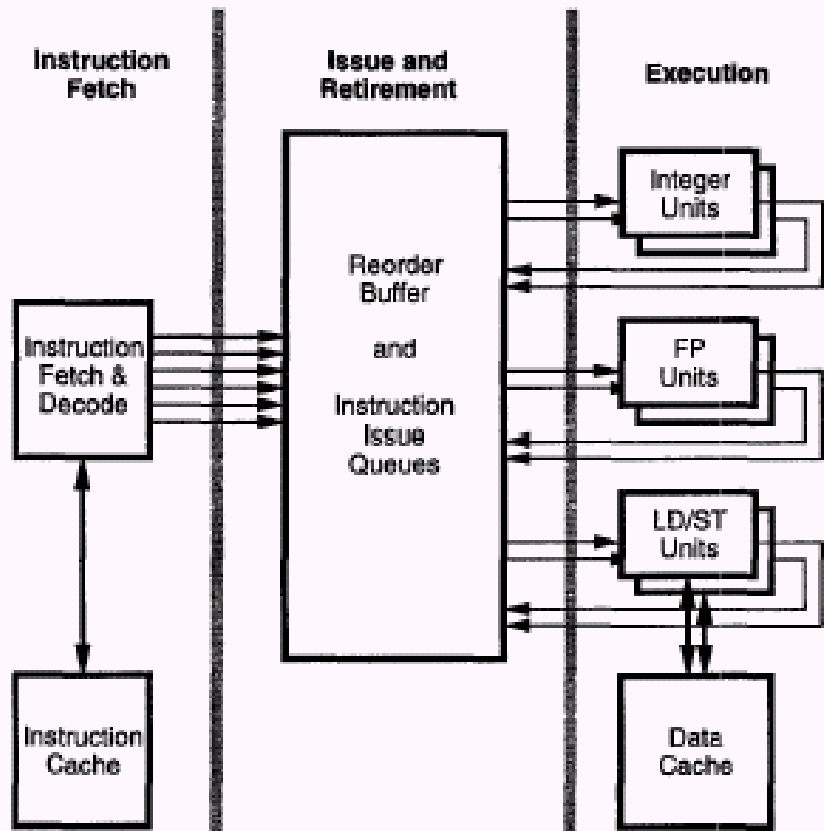
“The Case for a Single-Chip Multiprocessor”

Kunle Olukotun, *et al.* [ASPLOS '96]

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
- ⇒ Better resource utilization: Multiple processors on a chip

Dynamic Superscalar CPU



- Instruction queue often implemented as multiple instruction queues for different types 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 (cont'd)

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 cache into banks and fetch from multiple banks

Fetch Phase (cont'd)

3. Cache misses

- High miss rate → limit ability to maintain large window
- Can hide some cache miss latency by executing other instructions already in the window (dynamically scheduled CPUs)

⇒ 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 (cont'd)

- 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 → **Wires are slow**
- ⇒ Overall: Instruction issue queue will limit cycle time

Execution Phase

- Operand values: fetched from register file or bypassed from earlier instructions
- Wide superscalar has problems with
 - Register file
 - Larger to accommodate more renamed registers
 - Many ports
 - Complexity $\sim \#ports^2 \sim \text{issue width}^2$
 - Bypass logic
 - Complexity $\sim \#execution\ units^2$
 - **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 grains
 - < 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 (@ 500MHz)

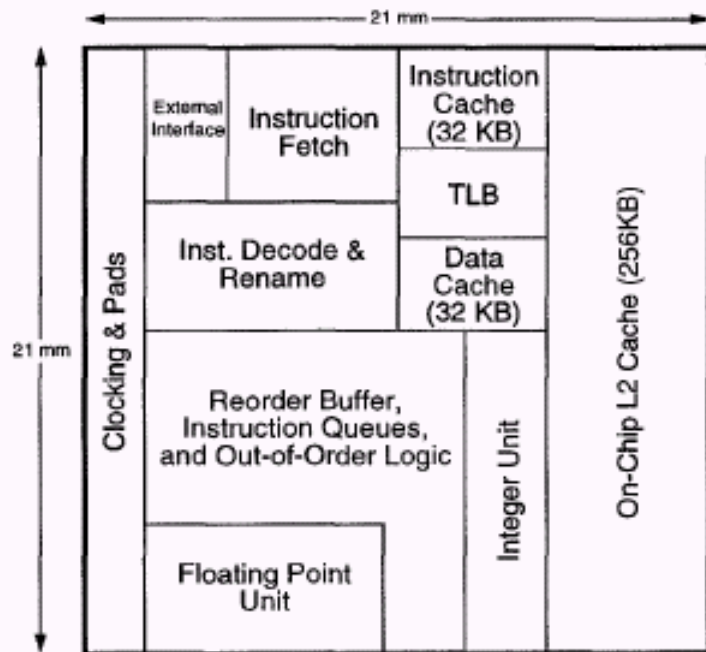


Figure 2. Floorplan for the six-issue dynamic superscalar microprocessor.

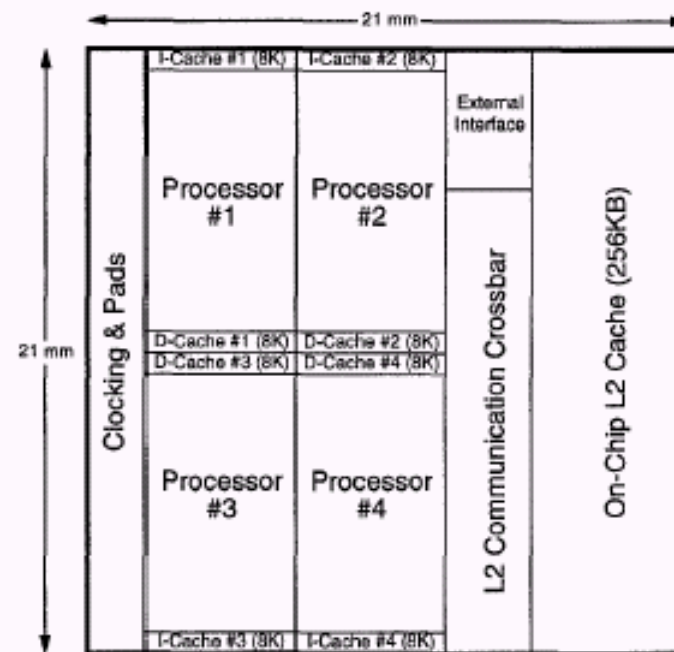
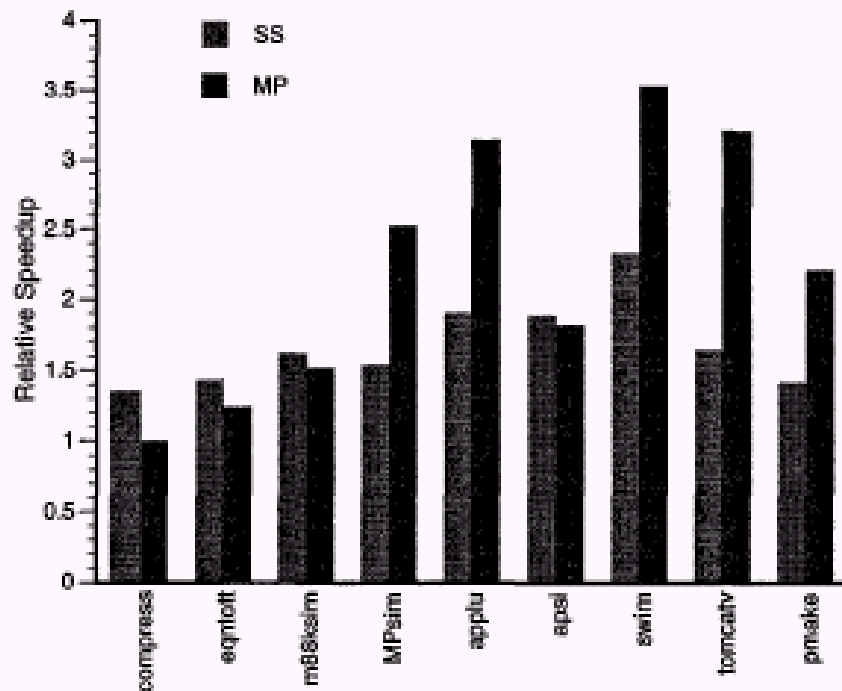


Figure 3. Floorplan for the four-way single-chip multiprocessor.

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× better