



inst.eecs.berkeley.edu/~cs61c
UC Berkeley CS61C : Machine Structures

Lecture 39 – Intra-machine Parallelism

2010-04-30




Head TA Scott Beamer
www.cs.berkeley.edu/~sbeamer



CS61C L39 Intra-machine Parallelism (1) Beamer, Spring 2010 © UCB

Review


- **Parallelism is necessary for performance**
 - It looks like it's the future of computing
 - It is unlikely that serial computing will ever catch up with parallel computing
- **Software Parallelism**
 - Grids and clusters, networked computers
 - MPI & MapReduce – two common ways to program
- **Parallelism is often difficult**
 - Speedup is limited by serial portion of the code and communication overhead



CS61C L39 Intra-machine Parallelism (2) Beamer, Spring 2010 © UCB

Today


- Superscalars
- Power, Energy & Parallelism
- Multicores
- Vector Processors
- Challenges for Parallelism



CS61C L39 Intra-machine Parallelism (3) Beamer, Spring 2010 © UCB

Disclaimers


- Please don't let today's material confuse what you have already learned about CPU's and pipelining
- When *programmer* is mentioned today, it means whoever is generating the assembly code (so it is probably a compiler)
- Many of the concepts described today are *difficult* to implement, so if it sounds easy, think of possible hazards



CS61C L39 Intra-machine Parallelism (4) Beamer, Spring 2010 © UCB

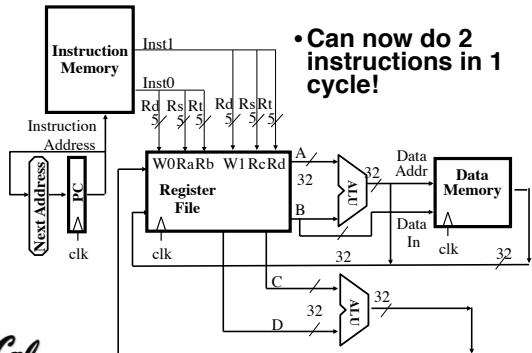
Superscalar

- Add more functional units or pipelines to the CPU
- Directly reduces CPI by doing more per cycle
- Consider what if we:
 - Added another ALU
 - Added 2 more read ports to the RegFile
 - Added 1 more write port to the RegFile




CS61C L39 Intra-machine Parallelism (5) Beamer, Spring 2010 © UCB

Simple Superscalar MIPS CPU



• Can now do 2 instructions in 1 cycle!



CS61C L39 Intra-machine Parallelism (6) Beamer, Spring 2010 © UCB

Simple Superscalar MIPS CPU (cont.)

- **Considerations**
 - ISA now has to be changed
 - Forwarding for pipelining now *harder*
- **Limitations of our example**
 - Programmer must explicitly generate instruction parallel code
 - Improvement only if other instructions can fill slots
 - Doesn't scale well



CS61C L39 Intra-machine Parallelism (7)

Beamer, Spring 2010 © UCB

Superscalars in Practice

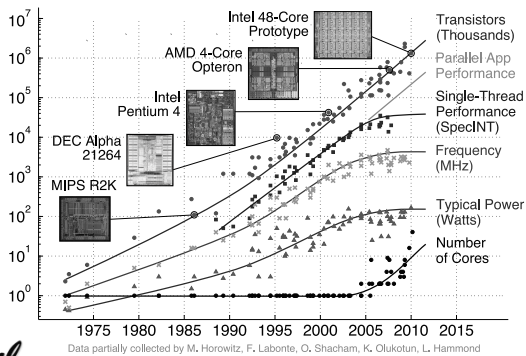
- **Modern superscalar processors often hide behind a scalar ISA**
 - Gives the illusion of a scalar processor
 - Dynamically schedules instructions
 - Tries to fill all slots with useful instructions
 - Detects hazards and avoids them
- **Instruction Level Parallelism (ILP)**
 - Multiple instructions from same instruction stream in flight at the same time
- **Examples: pipelining and superscalars**



CS61C L39 Intra-machine Parallelism (8)

Beamer, Spring 2010 © UCB

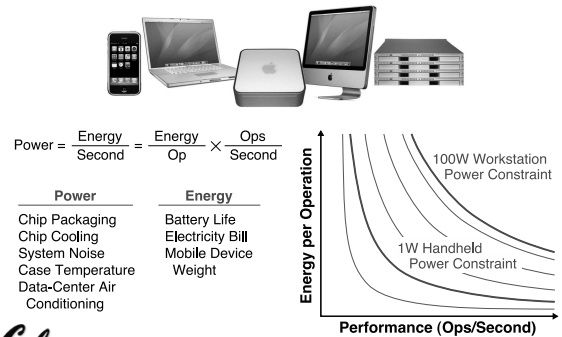
Scaling



CS61C L39 Intra-machine Parallelism (10)

Beamer, Spring 2010 © UCB

Energy & Power Considerations



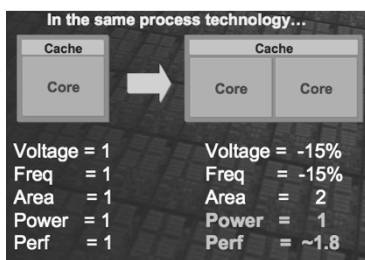
CS61C L39 Intra-machine Parallelism (12)

Courtesy: Chris Batten

Beamer, Spring 2010 © UCB

Parallel Helps Energy Efficiency

- **Power $\sim CV^2f$**
- **Circuit delay is roughly linear with V**



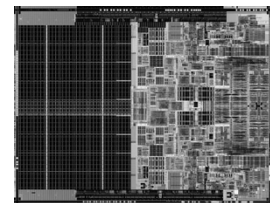
CS61C L39 Intra-machine Parallelism (13)

William Holt, HOT Chips 2005

Beamer, Spring 2010 © UCB

Multicore

- **Put multiple CPU's on the same die**
- **Cost of multicore: complexity and slower single-thread execution**
- **Task/Thread Level Parallelism (TLP)**
 - Multiple instructions streams in flight at the same time



CS61C L39 Intra-machine Parallelism (14)

Beamer, Spring 2010 © UCB

Cache Coherence Problem

CPU0

Cache	
Addr	Value
16	5

CPU1

Cache	
Addr	Value
16	0

CPU0:
LW R2, 16(R0)

CPU1:
LW R2, 16(R0)

CPU1:
SW R0, 16(R0)

View of memory no longer "coherent".

Loads of location 16 from CPU0 and CPU1 see different values!

Shared Main Memory	
Addr	Value
16	0

Cal CS61C L39 Intra-machine Parallelism (15) From: John Lazzaro Beamer, Spring 2010 © UCB

Real World Example: IBM POWER7

Cal CS61C L39 Intra-machine Parallelism (16) Beamer, Spring 2010 © UCB

Administrivia

- Proj3 due tonight at 11:59p
- Performance Contest posted tonight
- Office Hours Next Week
 - Watch website for announcements
- Final Exam Review – 5/9, 3-6p, 10 Evans
- Final Exam – 5/14, 8-11a, 230 Hearst Gym
 - You get to bring: two 8.5"x11" sheets of handwritten notes + MIPS Green sheet

Cal CS61C L39 Intra-machine Parallelism (17) Beamer, Spring 2010 © UCB

Flynn's Taxonomy

• Classification of Parallel Architectures

	Single Instruction	Multiple Instruction
Single Data	SISD: Instruction Pool → PU	MISD: Instruction Pool → PU → PU
Multiple Data	SIMD: Instruction Pool → PU → PU → PU → PU	MIMD: Instruction Pool → PU → PU → PU → PU

Cal CS61C L39 Intra-machine Parallelism (18) www.wikipedia.org Beamer, Spring 2010 © UCB

Vector Processors

- Vector Processors implement SIMD
 - One operation applied to whole vector
 - Not all program can easily fit into this
 - Can get high performance and energy efficiency by amortizing instruction fetch
 - Spends more silicon and power on ALUs
- Data Level Parallelism (DLP)
 - Compute on multiple pieces of data with the same instruction stream

Cal CS61C L39 Intra-machine Parallelism (19) Beamer, Spring 2010 © UCB

Vectors (continued)

- Vector Extensions
 - Extra instructions added to a scalar ISA to provide short vectors
 - Examples: MMX, SSE, AltiVec
- Graphics Processing Units (GPU)
 - Also leverage SIMD
 - Initially very fixed function for graphics, but now becoming more flexible to support general purpose computation

Cal CS61C L39 Intra-machine Parallelism (20) Beamer, Spring 2010 © UCB

Parallelism at Different Levels

- **Intra-Processor (within the core)**
 - Pipelining & superscalar (ILP)
 - Vector extensions & GPU (DLP)
- **Intra-System (within the box)**
 - Multicore – multiple cores per socket (TLP)
 - Multisocket – multiple sockets per system
- **Intra-Cluster (within the room)**
 - Multiple systems per rack and multiple racks per cluster



CS61C L39 Intra-machine Parallelism (21)

Beamer, Spring 2010 © UCB

Conventional Wisdom (CW) in Computer Architecture

1. Old CW: Power is free, but transistors expensive
 - New CW: **Power wall** Power expensive, transistors “free”
 - Can put more transistors on a chip than have power to turn on
2. Old CW: Multiplies slow, but loads fast
 - New CW: **Memory wall** Loads slow, multiplies fast
 - 200 clocks to DRAM, but even FP multiplies only 4 clocks
3. Old CW: More ILP via compiler / architecture innovation
 - Branch prediction, speculation, Out-of-order execution, VLIW, ...
 - New CW: **ILP wall** Diminishing returns on more ILP
4. Old CW: 2X CPU Performance every 18 months
 - New CW is **Power Wall** + **Memory Wall** + **ILP Wall** = **Brick Wall**



CS61C L39 Intra-machine Parallelism (22)

Beamer, Spring 2010 © UCB

Implication: We must go parallel

Parallelism again? What’s different this time?

“This shift toward increasing parallelism is not a triumphant stride forward based on breakthroughs in novel software and architectures for parallelism; instead, this plunge into parallelism is actually a retreat from even greater challenges that thwart efficient silicon implementation of traditional uniprocessor architectures.”

– Berkeley View, December 2006

- HW/SW Industry bet its future that breakthroughs will appear before it’s too late

view.eecs.berkeley.edu



CS61C L39 Intra-machine Parallelism (23)

Beamer, Spring 2010 © UCB

Peer Instruction

- 1) All energy efficient systems are low power
- 2) My GPU at peak can do more FLOP/s than my CPU

12
a) FF
b) FT
c) TF
d) TT



CS61C L39 Intra-machine Parallelism (24)

Beamer, Spring 2010 © UCB

“And In conclusion...”

- Parallelism is all about energy efficiency
- Types of Parallelism: ILP, TLP, DLP
- Parallelism already at many levels
- Great opportunity for change



CS61C L39 Intra-machine Parallelism (26)

Beamer, Spring 2010 © UCB