

CS184b: Computer Architecture (Abstractions and Optimizations)

Day 25: May 27, 2005
Transactional Computing



Caltech CS184 Spring2005 -- DeHon

Today

- Transactional Model
- Hardware Support
- Course Wrapup

2

Caltech CS184 Spring2005 -- DeHon

Transactional Compute Model

- Computation is a collection of atomic transactions
- Transactions are performed in some ordering
 - Specified sequential ordering
 - Some consistent sequential interleaving
- Looks like sequential (multithreaded) computation

3

Caltech CS184 Spring2005 -- DeHon

Atomic Blocks

- Indivisible
- Cannot get interleaving of operations among blocks
- Each block
 - reads state at some point in time;
 - provides all its updates (writes) at that point

4

Caltech CS184 Spring2005 -- DeHon

How Get?

- How do we get atomic blocks in our current models?
- Get parallelism with this model?

5

Caltech CS184 Spring2005 -- DeHon

Simple/Sequential Version

- Acquire a mutex
- Perform transaction
- Release mutex

6

Caltech CS184 Spring2005 -- DeHon

Parallel/Locking Version

- Read inputs, locking each as encounter
- Lock all outputs
- Perform operations
- Write outputs
- Unlock inputs/outputs

- OneChip Vector case is a single source/sink version of this

Caltech CS184 Spring2005 -- DeHon

7

Parallel Locking Dangers

What do we have to worry about in this version?

- Lock acquisition order → deadlock
- Parallelism?

Caltech CS184 Spring2005 -- DeHon

8

Optimistic (Speculative) Transactions

- **Pattern:** speculate/rollback
 - Common case: non-interacting
- Read inputs
- Perform computations
- Check no one changed inputs
 - Got correct inputs? → write outputs atomically
 - Inputs changed → rollback and try again

Caltech CS184 Spring2005 -- DeHon

9

Optimistic Transactions

- Compare
 - advanced load
 - ll/sc
 - branch prediction
- Doesn't need to be sequentialized
- Just need to look sequential
- It is adequate to
 - detect sequentialization violation

Caltech CS184 Spring2005 -- DeHon

10

Hardware Microarchitecture

Caltech CS184 Spring2005 -- DeHon

11

How support in hardware?

- What need?
 - Keep/mark inputs
 - Hold on to outputs
 - Check inputs
 - Commit outputs

Caltech CS184 Spring2005 -- DeHon

12

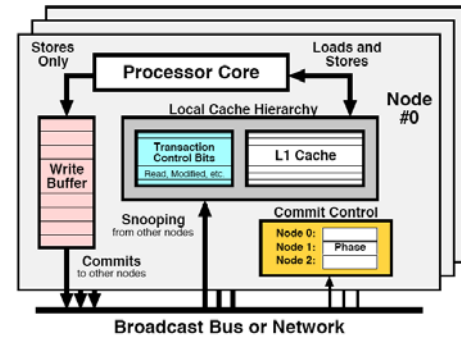
Microarchitecture

- Mark inputs in processor-local cache
- Writes into processor-local cache
- Write buffer for outputs
 - Hold until end of transaction
- Commit bus
- Snoop on bus for invalidations
- Dump write buffer to commit bus when turn to commit

Caltech CS184 Spring2005 -- DeHon

13

Stanford TCC uArch



Caltech CS184 Spring2005 -- DeHon

[Hammond et al./ISCA2004]

14

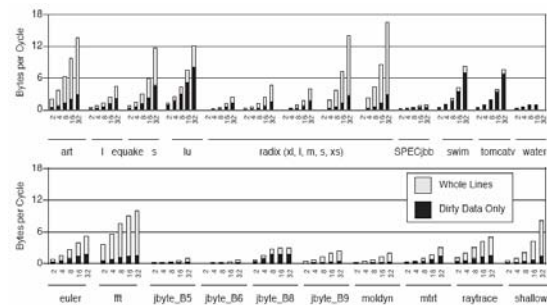
uArch Concerns?

- Impact of commit sequentialization?
- Time spent rolling back?
- Size of cache/write-buffer needed?

Caltech CS184 Spring2005 -- DeHon

15

Commit Traffic

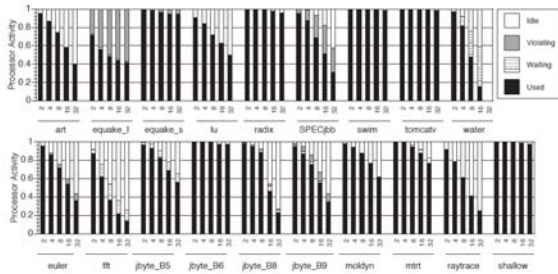


Caltech CS184 Spring2005 -- DeHon

[Hammond et al./ISCA2004]

16

Where time go?

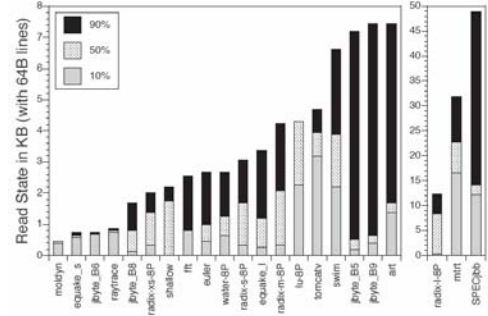


[Hammond et al./ISCA2004]

17

Caltech CS184 Spring2005 -- DeHon

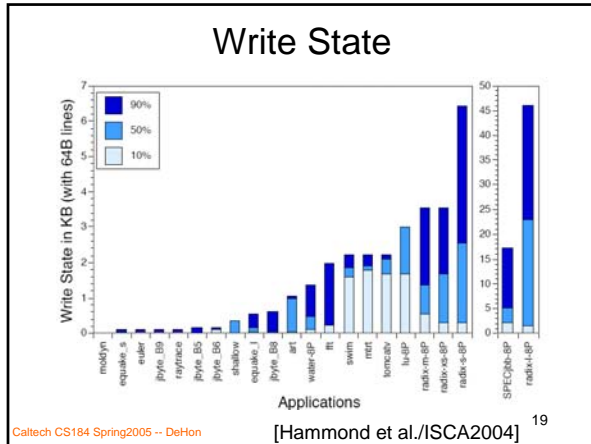
Read State



[Hammond et al./ISCA2004]

18

Caltech CS184 Spring2005 -- DeHon



- ### Other Patterns?
- Double Buffer
 - Parallel Prefix for Commits
- 21
- Caltech CS184 Spring2005 -- DeHon

- ### Programming
- Explicit
 - Parallel/transactional looping constructs
 - forks
 - Add like futures
 - Force ordering to match sequential language spec and is “safe”
 - Automatically add
 - Support “Mostly Functional” programming
 - “Make the fast case common”
- 22
- Caltech CS184 Spring2005 -- DeHon

- ### Performance Issues
- Too many violations?
 - Force some transactions to wait on others
 - Explicit/pragma/program construct
 - Feedback-based
 - Empirical dataflow discovery?
- 23
- Caltech CS184 Spring2005 -- DeHon

Stanford TCC Prototyping

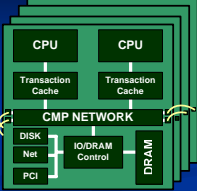
(from [Workshop on Architecture Research using FPGA Platforms](#) at HPCA 2004)

24

Caltech CS184 Spring2005 -- DeHon

ATLAS Overview


- A multi-board emulator for transactional parallel systems
- Goals
 - 16 to 64 CPUs (8 to 32 boards)
 - 50 to 100MHz
 - Stand-alone full-feature system
 - OS, IDE disks, 100Mb Ethernet, ...
- ATLAS architecture space
 - Small, medium, and large-scale CMPs and SMPs
 - UMA and NUMA
 - Flexible transactional memory hierarchy & protocol
 - Flexible network model
 - Flexible clocking, latency, and bandwidth settings



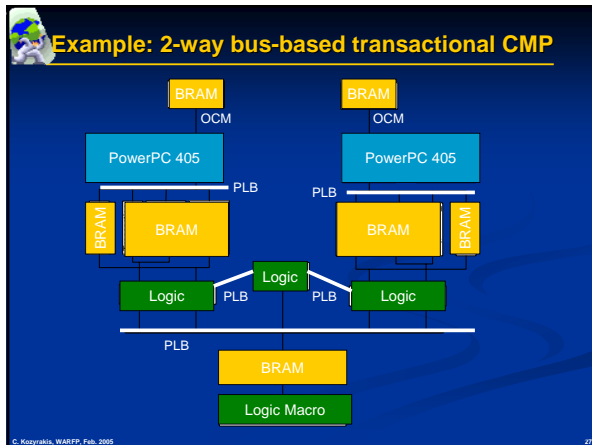
C. Kozyrakis, WARPP, Feb. 2005

Building Block: Xilinx ML310 Board

- XC2VP30 FPGA features
 - 2 PowerPC 405 cores
 - 2.4Mb dual-ported SRAM
 - 30K logic cells
 - 8 RocketIO 3.125Gbps transceivers
- System features
 - 256MB DDR, 512MB CompactFlash
 - Ethernet, PCI, USB, IDE, ...
- Design and development tools
 - Foundation ISE for design entry, synthesis, ...
 - For the transactional memory hierarchy and network
 - Chipscope Pro logic analyzer for debugging
 - EDK for system simulation, system SW development, configuration, ...
 - Montavista Linux 3.1 Pro



C. Kozyrakis, WARPP, Feb. 2005



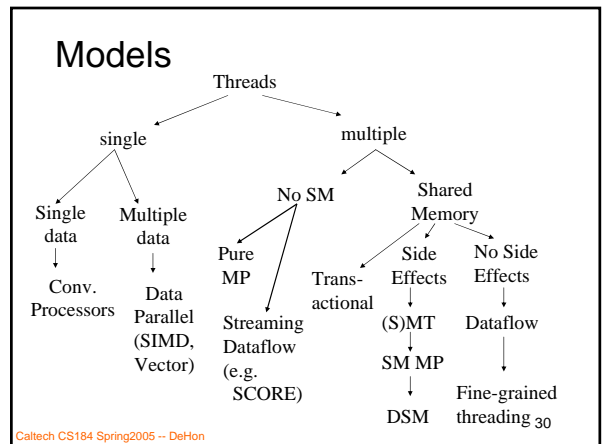
Big Ideas [Transactional]

- Model/Implementation
 - Only needs to appear sequentialized
- Speculation/Rollback
- Common Case

Caltech CS184 Spring2005 -- DeHon 28

Model Review

Caltech CS184 Spring2005 -- DeHon 29



Big Ideas

- Model
- Implementation support model semantics
 - But may be very different, optimized
- Scaling and change
- Common case fast; fast case common
 - Measure to understand