Snoop-Based Multiprocessor Design III: Case Studies

Todd C. Mowry
CS 418
March 2, 2011

Case Studies of Bus-based Machines

SGI Challenge, with Powerpath bus

SUN Enterprise, with Gigaplane bus

- Take very different positions on the design issues discussed above

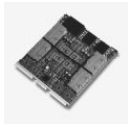
Overview

For each system:

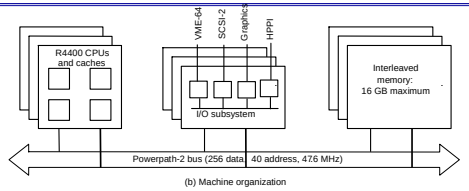
- Bus design
- Processor and Memory System
- Input/Output system
- Microbenchmark memory access results

Application performance and scaling (SGI Challenge)

SGI Challenge Overview



(a) A four-processor board



(b) Machine organization

36 MIPS R4400 (peak 2.7 GFLOPS, 4 per board) or 18 MIPS R8000 (peak 5.4 GFLOPS, 2 per board)

8-way interleaved memory (up to 16 GB)

4 I/O busses of 320 MB/s each

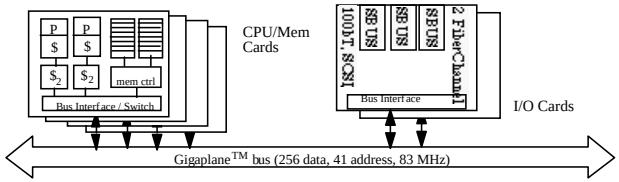
1.2 GB/s Powerpath-2 bus @ 47.6 MHz, 16 slots, 329 signals

128 Bytes lines (1 + 4 cycles)

Split-transaction with up to 8 outstanding reads

- all transactions take five cycles

SUN Enterprise Overview



Up to 30 UltraSPARC processors (peak 9 GFLOPs)

Gigaplane™ bus has peak bw 2.67 GB/s; upto 30GB memory

16 bus slots, for processing or I/O boards

- 2 CPUs and 1GB memory per board
  - memory distributed, unlike Challenge, but protocol treats as centralized
- Each I/O board has 2 64-bit 25Mhz SBUSes

Bus Design Issues

Multiplexed versus non-multiplexed (separate addr and data lines)

Wide versus narrow data busses

Bus clock rate
- Affected by signaling technology, length, number of slots...

Split transaction versus atomic

Flow control strategy

SGI Powerpath-2 Bus

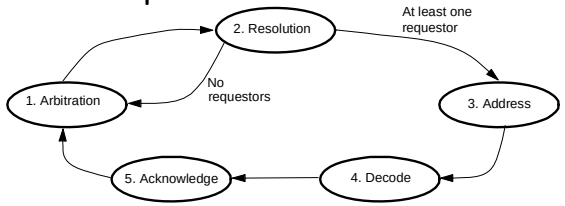
Non-multiplexed, 256-data/40-address, 47.6 MHz, 8 o/s requests

Wide => more interface chips so higher latency, but more bw at slower clock

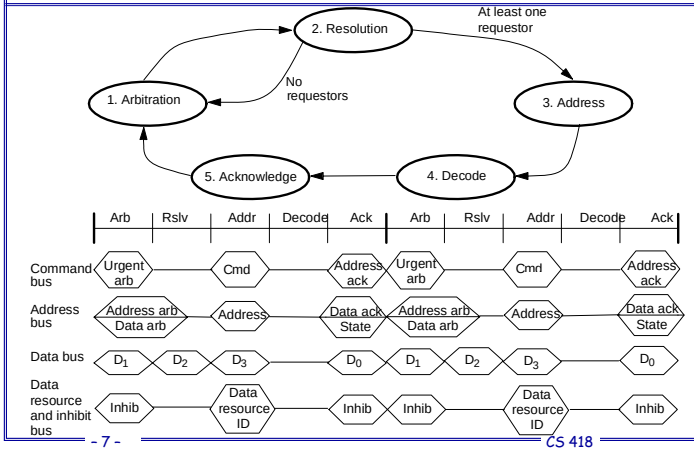
Large block size also calls for wider bus

Uses Illinois MESI protocol (cache-to-cache sharing)

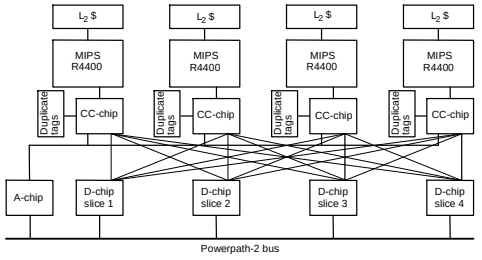
More detail in chapter



Bus Timing



Processor and Memory Systems



4 MIPS R4400 processors per board share A and D chips

A chip has address bus interface, request table, control logic

CC chip per processor has duplicate set of tags

Processor requests go from CC chip to A chip to bus

4 bit-sliced D chips interface CC chip to bus

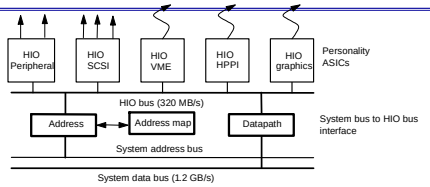
Memory Access Latency

250ns access time from address on bus to data on bus

But overall latency seen by processor is 1000ns!

- 300ns for request to get from processor to bus
  - down through cache hierarchy, CC chip and A chip
- 400ns later, data gets to D chips
  - 3 bus cycles to address phase of request transaction, 12 to access main memory, 5 to deliver data across bus to D chips
- 300ns more for data to get to processor chip
  - up through D chips, CC chip, and 64-bit wide interface to processor chip, load data into primary cache, restart pipeline

Challenge I/O Subsystem



Multiple I/O cards on system bus, each has 320MB/s HIO bus

- Personality ASICs connect these to devices (standard and graphics)

Proprietary HIO bus

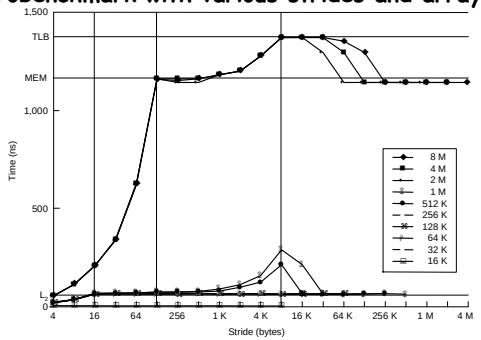
- 64-bit multiplexed address/data, same clock as system bus
- Split read transactions, up to 4 per device
- Pipelined, but centralized arbitration, with several transaction lengths
- Address translation via mapping RAM in system bus interface

Why the decouplings? (Why not connect directly to system bus?)

I/O board acts like a processor to memory system

Challenge Memory System Performance

Read microbenchmark with various strides and array sizes



Ping-pong flag-spinning microbenchmark: round-trip time 6.2 ms.

Sun Gigaplane Bus

Non-multiplexed, split-transaction, 256-data/41-address, 83.5 MHz

- Plus 32 ECC lines, 7 tag, 18 arbitration, etc. Total 388.

Cards plug in on both sides: 8 per side

112 outstanding transactions, up to 7 from each board

- Designed for multiple outstanding transactions per processor

Emphasis on reducing latency, unlike Challenge

- Speculative arbitration if address bus not scheduled from prev. cycle
- Else regular 1-cycle arbitration, and 7-bit tag assigned in next cycle

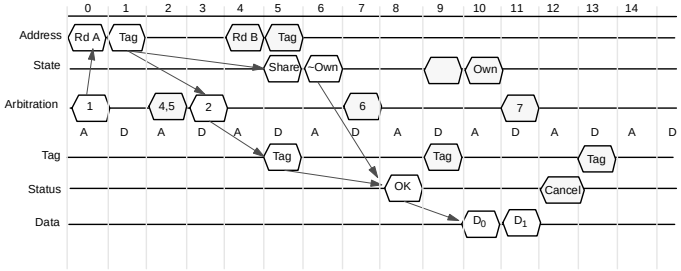
Snoop result associated with request phase (5 cycles later)

Main memory can stake claim to data bus 3 cycles into this, and start memory access speculatively

- Two cycles later, asserts tag bus to inform others of coming transfer

MOESI protocol (owned state for cache-to-cache sharing)

Gigaplane Bus Timing



Enterprise Processor and Memory System

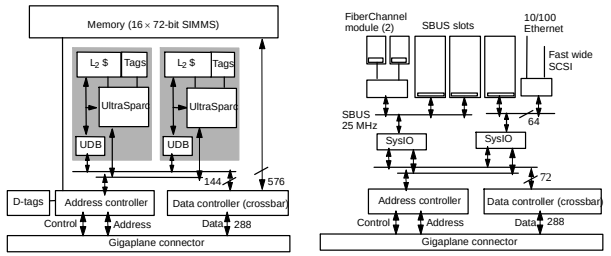
2 procs per board, external L2 caches, 2 mem banks with x-bar

Data lines buffered through UDB to drive internal 1.3 GB/s UPA bus

Wide path to memory so full 64-byte line in 1 mem cycle (2 bus cyc)

Addr controller adapts proc and bus protocols, does cache coherence

- its tags keep a subset of states needed by bus (e.g. no M/E distinction)



Enterprise I/O System

I/O board has same bus interface ASICs as processor boards

But internal bus half as wide, and no memory path

Only cache block sized transactions, like processing boards

- Uniformity simplifies design
- ASICs implement single-block cache, follows coherence protocol

Two independent 64-bit, 25 MHz Sbuses

- One for two dedicated FiberChannel modules connected to disk
- One for Ethernet and fast wide SCSI
- Can also support three SBUS interface cards for arbitrary peripherals

Performance and cost of I/O scale with no. of I/O boards

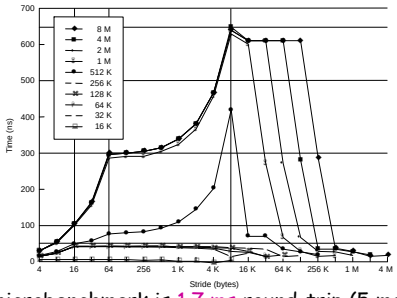
Memory Access Latency

300ns read miss latency

11 cycle min bus protocol at 83.5 Mhz is 130ns of this time

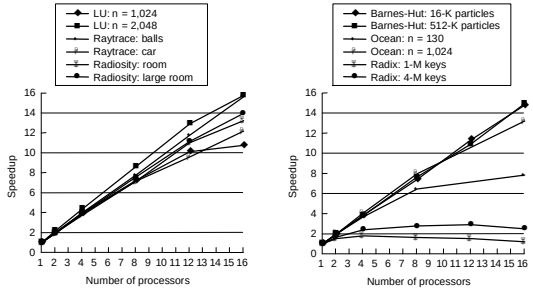
Rest is path through caches and the DRAM access

TLB misses add 340 ns



Ping-pong microbenchmark is 1.7 ms round-trip (5 mem accesses)

Application Speedups (Challenge)



- **Problem in Ocean with small problem: communication and barrier cost**
- **Problem in Radix: contention on bus due to very high traffic**
  - also leads to high imbalances and barrier wait time

Application Scaling under Other Models