Lecture 17:
Parallel Architectures and Future Computer Architectures

Prof. Kunle Olukotun
EE 282h
Fall 98/99

Shared-Memory Multiprocessors

- Several processors share one address space
  - conceptually a shared memory
  - often implemented just like a multicomputer
    - address space distributed over private memories
- Communication is implicit
  - read and write accesses to shared memory locations
- Synchronization
  - via shared memory locations
    - spin waiting for non-zero barriers
  - barriers
Cache Coherence - A Quick Overview

- With caches, action is required to prevent access to stale data
  - Processor 1 may read old data from its cache instead of new data in memory or
  - Processor 3 may read old data from memory rather than new data in Processor 2’s cache

- Solutions
  - no caching of shared data
    - Cray T3D, T3E, IBM RP3, BBN Butterfly
  - cache coherence protocol
    - keep track of copies
    - notify (update or invalidate) on writes

Cache Coherence Protocols

- Cache lines have a state
  - I - invalid
  - S - possibly shared
  - E - exclusive
  - M - modified (and exclusive)
    - sometimes called a MESI protocol

- State table determines next state and action depending on current state and operation

<table>
<thead>
<tr>
<th>State</th>
<th>CPU-Read</th>
<th>CPU-Write</th>
<th>CPU-Evict</th>
<th>M-Read</th>
<th>M-Write</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Op</td>
<td>Next</td>
<td>Op</td>
<td>Next</td>
<td>Op</td>
</tr>
<tr>
<td>I</td>
<td>M-Read</td>
<td>S</td>
<td>M-Write</td>
<td>E</td>
<td>None</td>
</tr>
<tr>
<td>S</td>
<td>None</td>
<td>S</td>
<td>M-Write</td>
<td>E</td>
<td>None</td>
</tr>
<tr>
<td>E</td>
<td>None</td>
<td>E</td>
<td>None</td>
<td>M</td>
<td>None</td>
</tr>
<tr>
<td>M</td>
<td>None</td>
<td>M</td>
<td>None</td>
<td>M</td>
<td>M-Write</td>
</tr>
</tbody>
</table>

P1: Rd(A) Rd(A)
P2: Wr(A,5)
P3: Rd(A)
Cache Coherence Issues

- Finding the copies
  - A copy of a line must see all reads and writes to that line
  - At most one E-copy
  - Or any number of S-copies

- Broadcast
  - M-read and M-write operations are broadcast to all processors
  - ‘Snoopy’ bus

- Directories
  - A central directory keeps a record of all copies
  - Read and write operations forwarded to just the right nodes

```
Network
M
A:3

P1: Rd(A) Rd(A)
P2: Wr(A,5)
P3: Rd(A)
```

What to do with a billion transistors - and slow wires

- Technology changes the cost and performance of computer elements in a non-uniform manner
  - Logic and arithmetic is becoming plentiful and cheap
  - Wires are becoming slow and scarce

- This changes the tradeoffs between alternative architectures
  - Superscalar doesn’t scale well – global control and data

- So what will the architectures of the future be?

```
1 clk

1998
2001
2004
2007

64 x the area
4x the speed
slower wires

20 clks
```
**IA-64 aka VLIW**

- Compiler schedules instructions
- Encodes dependencies explicitly
  - saves having the hardware repeatedly rediscover them
- Support speculation
  - speculative load
  - branch prediction
- Really need to make communication explicit too
  - still has global registers and global instruction issue

**Single-Chip Multiprocessors**

- Build a multiprocessor on a single chip
  - linear increase in peak performance
  - advantage of fast interaction between processors
- Fine grain threads
  - make communication and synchronization very fast (1 cycle)
  - break the problem into smaller pieces
- Memory bandwidth
  - Makes more effective use of limited memory bandwidth
- Programming model
  - Need parallel programs
Base Hydra Design

- Single-chip multiprocessor
- Four processors
- Separate primary caches
- Write-through data caches to maintain coherence
- Shared 2nd-level cache
- Separate read and write busses
- Data Speculation Support

Processor with DRAM (PIM)
IRAM, VIRAM

- Put the processor and the main memory on a single chip
  - much lower memory latency
  - much higher memory bandwidth

- But
  - need to build systems with more than one chip

64Mb SDRAM Chip
Internal - 128 512K subarrays
4 bits per subarray each 10ns
51.2 Gb/s

External - 8 bits at 10ns, 800Mb/s
1 Integer processor ~ 100KBytes DRAM
1 FP processor ~ 500KBytes DRAM
1 Vector Unit ~ 1 MByte DRAM
Reconfigurable processors

- Adapt the processor to the application
  - special function units
  - special wiring between function units
- Builds on FPGA technology
  - FPGAs are inefficient
    - a multiplier built from an FPGA is about 100x larger and 10x slower than a custom multiplier.
  - Need to raise the granularity
    - configure ALUs, or whole processors
  - Memory and communication are usually the bottleneck
    - not addressed by configuring a lot of ALUs
- Programming model
  - Difficult to program
  - Verilog

Have a Good Christmas Break

- and good luck on the final.