Lecture 18: Reducing Cache Hit Time and Main Memory Design

Virtual Cache, pipelined cache, cache summary, main memory technology

### Improving Cache Performance

1. **Reducing miss rates**
   - Larger block size
   - Larger cache size
   - Higher associativity
   - Victim caches
   - Way prediction and Pseudassociativity
   - Compiler optimization

2. **Reducing miss penalty**
   - Multilevel caches
   - Critical word first
   - Read miss first
   - Merging write buffers

3. **Reducing miss penalty or miss rates via parallelism**
   - Non-blocking caches
   - Hardware prefetching
   - Compiler prefetching

4. **Reducing cache hit time**
   - Small and simple caches
   - Avoiding address translation
   - Pipelined cache access
   - Trace caches

### Fast Cache Hits by Avoiding Translation: Process ID impact

- Black is uniprocess
- Light Gray is multiprocess when flush cache
- Dark Gray is multiprocess when use Process ID tag
- Y axis: Miss Rates up to 20%
- X axis: Cache size from 2 KB to 1024 KB

### Fast Cache Hits by Avoiding Translation: Index with Physical Portion of Address

- If a direct mapped cache is no larger than a page, then the index is physical part of address
- Can start tag access in parallel with translation so that can compare to physical tag

- Limits cache to page size: what if want bigger caches and uses same trick?
  - Higher associativity moves barrier to right
  - Page coloring
- Compared with virtual cache used with page coloring?

### Pipelined Cache Access

For multi-issue, cache bandwidth affects effective cache hit time
- Queueing delay adds up if cache does not have enough read/write ports

Pipelined cache accesses: reduce cache cycle time and improve bandwidth

Cache organization for high bandwidth
- Duplicate cache
- Banked cache
- Double clocked cache

### Pipelined Cache Access

Alpha 21264 Data cache design
- The cache is 64KB, 2-way associative; cannot be accessed within one-cycle
- One-cycle used for address transfer and data transfer, pipelined with data array access
- Cache clock frequency doubles processor frequency; wave pipelined to achieve the speed
Trace Cache

- Trace: a dynamic sequence of instructions including taken branches
- Traces are dynamically constructed by processor hardware and frequently used traces are stored into trace cache
- Example: Intel P4 processor, storing about 12K mops

Summary of Reducing Cache Hit Time

- Small and simple caches: used for L1 inst/data cache
  - Most L1 caches today are small but set-associative and pipelined (emphasizing throughput?)
  - Used with large L2 cache or L2/L3 caches
- Avoiding address translation during indexing cache
  - Avoid additional delay for TLB access

Cache Optimization Summary

<table>
<thead>
<tr>
<th>Technique</th>
<th>MP</th>
<th>MR</th>
<th>HT</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nonblocking caches</td>
<td>+</td>
<td>2</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Software prefetching</td>
<td>-</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Small and simple cache</td>
<td>-</td>
<td>+</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Avoiding address translation</td>
<td>+</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pipeline cache access</td>
<td>+</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trace cache</td>
<td>+</td>
<td>3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Main Memory Background

- Performance of Main Memory:
  - Latency: Cache Miss Penalty
    - Access Time: time between request and word arrives
    - Cycle Time: time between requests
  - Bandwidth: I/O & Large Block Miss Penalty (L2)
- Main Memory is DRAM: Dynamic Random Access Memory
  - Dynamic since needs to be refreshed periodically (8 ms, 1% time)
  - Addresses divided into 2 halves (Memory as a 2D matrix):
    - RAS or Row Access Strobe
    - CAS or Column Access Strobe
- Cache uses SRAM: Static Random Access Memory
  - No refresh (6 transistors/bit vs 1 transistor)
  - Size: DRAM/SRAM - 4:8 even more today
  - Cost/Cycle time: SRAM/DRAM - 8:16
DRAM Internal Organization

- Square root of bits per RAS/CAS

Key DRAM Timing Parameters

- Row access time: the time to move data from DRAM core to the row buffer (may add time to transfer row command)
  - Quoted as the speed of a DRAM when buy
  - Typically 20 ns
- Column access time: the time to select a block of data in the row buffer and transfer it to the processor
  - Generally 20 ns
- Cycle time: between two row accesses to the same bank
- Data transfer time: the time to transfer a block (usually cache block); determined by bandwidth
  - PC100 bus: 8-byte wide, 100MHz, 800MB/s bandwidth, 80ns to transfer a 64-byte block
  - Direct Rambus, 2-channel: 2-byte wide, 400MHz DDR, 3.2GB/s bandwidth, 20ns to transfer a 64-byte block
- Additional time for memory controller and data path inside processor

Independent Memory Banks

- How many banks?
  - number banks ≤ number clocks to access word in bank
  - For sequential accesses, otherwise may return to original bank before it has next word ready
  - Increasing DRAM ⇒ fewer chips ⇒ harder to have banks
  - Exception: Direct Rambus, 32 banks per chip, 32 x N banks for N chips

DRAM History

- DRAMs: capacity +60%/yr, cost -30%/yr
  - 2.5X cells/area, 1.5X die size in 3 years
  - '98 DRAM fab line costs $2B
- DRAM only: density, leakage v. speed
  - Rely on increasing no. of computers & memory per computer (60% market)
  - SIMM or DIMM is replaceable unit
  - computers use any generation DRAM
- Commodity, second source industry
  - High volume, low profit, conservative
  - Little organization innovation in 20 years
- Order of importance: 1) Cost/bit 2) Capacity
  - First RAMBUS: 10X BW, +30% cost ⇒ little impact

Fast Memory Systems: DRAM specific

- Multiple CAS accesses: several names (page mode)
  - Extended Data Out (EDO): 30% faster in page mode
- New DRAMs to address gap; what will they cost, will they survive?
  - RAMBUS startup company, reinvent DRAM interface
  - Each Chip a module vs. slice of memory
  - Short bus between CPU and chips
  - Does own refresh
  - Variable amount of data returned
  - 1 byte / 2 ns (500 MB/s per channel)
  - 20% increase in DRAM area
  - Direct Rambus: 2 byte / 1.25 ns (800 MB/s per channel)
  - Synchronous DRAM: 2 banks on chip, a clock signal to DRAM, transfer synchronous to system clock (66 - 150 MHz)
  - DDR Memory: SDRAM + Double Data Rate, PC2100 means 133MHz times 8 bytes times 2
- Which will win, Direct Rambus or DDR?