Lecture 17

Adapted from instructor’s supplementary material from *Computer Organization and Design, 4th Edition*, Patterson & Hennessy, © 2008, MK]
SRAM / DRAM / Flash / RRAM / HDD
SRAM / DRAM / Flash / RRAM / HDD

• Purpose
  – Long term, nonvolatile storage
  – Lowest level in the memory hierarchy
    • slow, large, inexpensive

• General structure
  – A rotating platter coated with a magnetic surface
  – A moveable read/write head to access the information on the disk

• Typical numbers
  – 1 to 4 platters (each with 2 recordable surfaces) per disk of 1” to 3.5” in diameter
  – Rotational speeds of 5,400 to 15,000 RPM
  – 10,000 to 50,000 tracks per surface
    • cylinder - all the tracks under the head at a given point on all surfaces
  – 100 to 500 sectors per track
    • the smallest unit that can be read/written (typically 512B)
Magnetic Disk Characteristic

- Disk read/write components
  1. **Seek time**: position the head over the proper track (3 to 13 ms avg)
     - due to locality of disk references the actual average seek time may be only 25% to 33% of the advertised number
  2. **Rotational latency**: wait for the desired sector to rotate under the head (½ of 1/RPM converted to ms)
     - $0.5/5400\text{RPM} = 5.6\text{ms}$ to $0.5/15000\text{RPM} = 2.0\text{ms}$
  3. **Transfer time**: transfer a block of bits (one or more sectors) under the head to the disk controller’s cache (70 to 125 MB/s are typical disk transfer rates in 2008)
     - the disk controller’s “cache” takes advantage of spatial locality in disk accesses
       - cache transfer rates are much faster (e.g., 375 MB/s)
  4. **Controller time**: the overhead the disk controller imposes in performing a disk I/O access (typically < .2 ms)
How is the Hierarchy Managed?

• registers ↔ memory
  – by compiler (programmer?)

• cache ↔ main memory
  – by the cache controller hardware

• main memory ↔ disks
  – by the operating system (virtual memory)
  – virtual to physical address mapping assisted by the hardware (TLB)
  – by the programmer (files)
Memory Systems that Support Caches

- The off-chip interconnect and memory architecture can affect overall system performance in dramatic ways

One word wide organization (one word wide bus and one word wide memory)

- **Assume**
  1. 1 memory bus clock cycle to send the addr
  2. 15 memory bus clock cycles to get the 1st word in the block from DRAM (row cycle time), 5 memory bus clock cycles for 2nd, 3rd, 4th words (column access time)
  3. 1 memory bus clock cycle to return a word of data

- **Memory-Bus to Cache bandwidth**
  - number of bytes accessed from memory and transferred to cache/CPU per memory bus clock cycle
One Word Wide Bus, One Word Blocks

- If the block size is one word, then for a memory access due to a cache miss, the pipeline will have to stall for the number of cycles required to return one data word from memory:
  - cycle to send address
  - cycles to read DRAM
  - cycle to return data
  - total clock cycles miss penalty

- Number of bytes transferred per clock cycle (bandwidth) for a single miss is bytes per memory bus clock cycle
One Word Wide Bus, One Word Blocks

- If the block size is one word, then for a memory access due to a cache miss, the pipeline will have to stall for the number of cycles required to return one data word from memory:
  - 1 memory bus clock cycle to send address
  - 15 memory bus clock cycles to read DRAM
  - 1 memory bus clock cycle to return data
  - Total clock cycles miss penalty: 17

- Number of bytes transferred per clock cycle (bandwidth) for a single miss is:
  - bytes per memory bus clock cycle:

\[
\frac{4}{17} = 0.235
\]
One Word Wide Bus, Four Word Blocks

- What if the block size is four words and each word is in a different DRAM row?
  
  cycle to send 1\textsuperscript{st} address
  cycles to read DRAM
  cycles to return last data word
  
  total clock cycles miss penalty

- Number of bytes transferred per clock cycle (bandwidth) for a single miss is bytes per clock
One Word Wide Bus, Four Word Blocks

- What if the block size is four words and each word is in a different DRAM row?
  
  1 cycle to send 1<sup>st</sup> address
  
  4 x 15 = 60 cycles to read DRAM
  
  1 cycles to return last data word
  
  $\frac{62}{62}$ total clock cycles miss penalty
  
- Number of bytes transferred per clock cycle (bandwidth) for a single miss is bytes per clock
  
  $(4 \times 4)/62 = 0.258$
One Word Wide Bus, Four Word Blocks

- What if the block size is four words and all words are in the same DRAM row?
  - cycle to send 1\textsuperscript{st} address
  - cycles to read DRAM
  - cycles to return last data word
  - total clock cycles miss penalty

- Number of bytes transferred per clock cycle (bandwidth) for a single miss is
  - bytes per clock
One Word Wide Bus, Four Word Blocks

• What if the block size is four words and all words are in the same DRAM row?

\[
\begin{align*}
1 & \quad \text{cycle to send 1}\text{st address} \\
15 + 3 \times 5 &= 30 \quad \text{cycles to read DRAM} \\
\frac{1}{32} &\quad \text{cycles to return last data word} \\
\end{align*}
\]

total clock cycles miss penalty

• Number of bytes transferred per clock cycle (bandwidth) for a single miss is

\[
\frac{4 \times 4}{32} = 0.5
\]
(DDR) SDRAM Operation

- After a row is read into the SRAM register
  - Input CAS as the starting “burst” address along with a burst length
  - Transfers a burst of data (ideally a cache block) from a series of sequential addr’s within that row
    - The memory bus clock controls transfer of successive words in the burst
Interleaved Memory, One Word Wide Bus

- For a block size of four words
  - cycle to send 1st address
  - cycles to read DRAM banks
  - cycles to return last data word
  - total clock cycles miss penalty

- Number of bytes transferred per clock cycle (bandwidth) for a single miss is
  - bytes per clock
Interleaved Memory, One Word Wide Bus

- For a block size of four words:
  - 1 cycle to send 1st address
  - 15 cycles to read DRAM banks
  - $4 \times 1 = 4$ cycles to return last data word
  - 20 total clock cycles miss penalty

- Number of bytes transferred per clock cycle (bandwidth) for a single miss is
  \[(4 \times 4)/20 = 0.8\] bytes per clock
DRAM Memory System Summary

• Its important to match the cache characteristics
  – caches access one block at a time (usually more than one word)

• with the DRAM characteristics
  – use DRAMs that support fast multiple word accesses, preferably ones that match the block size of the cache

• with the memory-bus characteristics
  – make sure the memory-bus can support the DRAM access rates and patterns
  – with the goal of increasing the Memory-Bus to Cache bandwidth
Memory hierarchy summary
4 Questions for the Memory Hierarchy

• Q1: Where can a entry be placed in the upper level? *(Entry placement)*

• Q2: How is a entry found if it is in the upper level? *(Entry identification)*

• Q3: Which entry should be replaced on a miss? *(Entry replacement)*

• Q4: What happens on a write? *(Write strategy)*
Q1 & Q2: Where can an entry be placed/found?

<table>
<thead>
<tr>
<th></th>
<th># of sets</th>
<th>Entries per set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct mapped</td>
<td># of entries</td>
<td>1</td>
</tr>
<tr>
<td>Set associative</td>
<td>(# of entries) / associativity</td>
<td>Associativity (typically 2 to 16)</td>
</tr>
<tr>
<td>Fully associative</td>
<td>1</td>
<td># of entries</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Location method</th>
<th># of comparisons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct mapped</td>
<td>Index</td>
<td>1</td>
</tr>
<tr>
<td>Set associative</td>
<td>Index the set; compare set’s tags</td>
<td>Degree of associativity</td>
</tr>
<tr>
<td>Fully associative</td>
<td>Compare all entries’ tags</td>
<td># of entries</td>
</tr>
<tr>
<td></td>
<td>Separate lookup (page) table</td>
<td>0</td>
</tr>
</tbody>
</table>
Q3: Which entry should be replaced on a miss?

- Easy for direct mapped – only one choice
- Set associative or fully associative
  - Random
  - LRU (Least Recently Used)
- For a 2-way set associative, random replacement has a miss rate about 1.1 times higher than LRU
- LRU is too costly to implement for high levels of associativity (> 4-way) since tracking the usage information is costly
Q4: What happens on a write?

- **Write-through** – The information is written to the entry in the current memory level *and* to the entry in the next level of the memory hierarchy
  - Always combined with a write buffer so write waits to next level memory can be eliminated (as long as the write buffer doesn’t fill)
- **Write-back** – The information is written only to the entry in the current memory level. The modified entry is written to next level of memory only when it is replaced.
  - Need a dirty bit to keep track of whether the entry is clean or dirty
  - Virtual memory systems always use write-back of dirty pages to disk
- Pros and cons of each?
  - Write-through: read misses don’t result in writes (so are simpler and cheaper), easier to implement
  - Write-back: writes run at the speed of the cache; repeated writes require only one write to lower level
Summary: Improving Cache Performance

0. Reduce the time to hit in the cache
   - smaller cache
   - direct mapped cache
   - smaller blocks
   - for writes
     - no write allocate – no “hit” on cache, just write to write buffer
     - write allocate – to avoid two cycles (first check for hit, then write)
       pipeline writes via a delayed write buffer to cache

1. Reduce the miss rate
   - bigger cache
   - more flexible placement (increase associativity)
   - larger blocks (16 to 64 bytes typical)
   - victim cache – small buffer holding most recently discarded blocks
Summary: Improving Cache Performance

2. Reduce the miss penalty
   – smaller blocks
   – use a write buffer to hold dirty blocks being replaced so don’t have to wait for the write to complete before reading
   – check write buffer (and/or victim cache) on read miss – may get lucky
   – for large blocks fetch critical word first
   – use multiple cache levels – L2 cache not tied to CPU clock rate
   – faster backing store/improved memory bandwidth
     • wider buses
     • memory interleaving, DDR SDRAMs
Summary: The Cache Design Space

• Several interacting dimensions
  – cache size
  – block size
  – associativity
  – replacement policy
  – write-through vs write-back
  – write allocation
• The optimal choice is a compromise
  – depends on access characteristics
    • workload
    • use (I-cache, D-cache, TLB)
  – depends on technology / cost
• Simplicity often wins