lecture 18

cache 2

- TLB (hit and miss)
- instruction or data cache
- cache (hit and miss)

Wed. March 16, 2016
Last lecture I discussed the TLB and how virtual addresses are translated to physical addresses. I only discussed the cases of TLB hits. Here I will briefly discuss TLB misses.

NOTE: I have changed the order of the slides. In the lecture these slides were later in the lecture which was confusing.
If the address is not in the TLB, then an exception occurs. The TLB exception handler must replace the entry in the TLB.
TLB

hit

see above

miss

TLB miss handler checks page table (main memory)

Q: is desired word in main memory?

A: yes

TBL "refill"

no

page fault, (then try again)

copy translation from page table to TLB

move desired page from disk to main memory and update page table
I will have more to say about TLB misses and page faults later in the course. For the rest of this lecture, we assume that there is a TLB hit. The TLB only stores physical addresses that are in main memory (not on hard disk). Thus, we can proceed beyond the TLB in the pipeline and assume we are going to the data or instruction cache with a physical address that corresponds to an instruction or data in main memory, not on the hard disk.
physical address (RAM) e.g. 1 GB = $2^{30}$ bytes

<table>
<thead>
<tr>
<th>physical page number (18 bits)</th>
<th>page offset</th>
</tr>
</thead>
<tbody>
<tr>
<td>29 ... 12 11 ... 0</td>
<td></td>
</tr>
</tbody>
</table>

The cache is independent of paging. It works with physical address only.
Instruction or data cache

Suppose each cache holds $128\text{KB} = 2^{17}$ bytes.

Only a subset of $2^{30}$ bytes of RAM can be put in each of the instruction and data caches.

How to index and 'recognize' these bytes?
For any cache index, there are $2^{13}$ bytes in RAM whose addresses have that cache index.

Assume instruction or data cache has $2^{17}$ bytes.
Instruction or data cache

We will consider two cache designs ($2^{17}$ bytes).

$2^{15}$ words

$2^{13}$ blocks (4 words per block)
Cache design 1

Each cache entry has one word. (Assume word aligned.)

\[2^{17} / 2^2 = 2^{15}\] words in the cache

For any cache index, there are \(2^{13}\) words in RAM whose addresses have that cache index.
Cache design 2

Each cache entry has one "block" (e.g. four words).

Thus, $2^{13}$ "blocks" in the cache ($2^{17} / 2^2 / 2^2 = 2^{13}$)

![Diagram of cache design 2](image)

- **Tag**: 29 bits, indexes one of $2^{13}$ blocks.
- **"Cache index"**: 17 bits, indexes one word within the block.
- **b3 b2 00**: 3 bits, used for other purposes.

(indexes one of $2^{13}$ blocks)
(indexes one word within the block)
I will discuss the "dirty bit" next class.
lecture 18

cache 2

- instruction or data cache
- cache misses

Wed. March 16, 2016
Up to now, we have assumed that the desired address was represented in the cache (TLB, instruction, data).

What happens if there is a cache miss?
instruction cache (IF stage)

- hit
- miss

hit:
- copy block from main memory to cache
- set valid bit
- try again (now we hit)
The kernel program (OS) that handles cache misses is called the 'cache miss handler'. It performs a "cache refill".

Note that all of this happens in 'virtual' and 'physical' address space. Kernel program addresses also are translated.
The data cache works differently than the instruction cache, since programs write to the data region of Memory. Thus (physically) there are also writes from the cache to RAM.
If we write to data cache (sw), then we need a policy for maintaining consistency between cache and main memory.
Two policies for data cache (MEM stage)

- "write through"
  (always maintain consistency between cache and main memory - whenever we do 'sw', we also write the word back to main memory)

- "write back"
  (Only maintain consistency when necessary.)

  When there is cache miss and the cache line is "dirty", i.e. we have written into it (sw), first copy the block from the cache back to main memory)
data cache read lw ("write through")

hit
- copy word from cache to register

miss
- copy block from main memory to cache (write through ensures consistency)
  - set valid bit
  - try again (now we hit)
data cache write sw ("write through")

hit
- copy word from register to cache
- copy word from cache to main memory

miss
- copy block from main memory to cache
- set valid bit
- try again (now we hit)
data cache \textbf{read} lw ("write back")

- **hit**
  - copy word from cache to register

- **miss**
  - if block is dirty, then copy block from cache to main memory
  - copy new block from main memory to cache
  - set valid bit
  - try again (now we hit)
data cache write sw ("write back")

hit
- copy word from register to cache
- set dirty bit
- try again (now we hit)

miss
- if block is dirty, then copy block from cache to main memory
- copy new block from main memory to cache
- set valid bit
Memory lectures (16, 17, 18)
review of concepts
1. memory hierarchy

- Registers
- Cache (SRAM)
- Main memory (DRAM)
- Disk

- fast, small, expensive
- slow, big, cheap
2. paging

[address] = [ page number, page offset]
3. blocks

\[
\text{cache} = \text{number of blocks in cache} \times \text{block size}
\]

\[
\text{main memory} = \text{number of blocks in main memory} \times \text{block size}
\]

\[
[\text{physical address}] = [\text{block number}, \text{word number}, \text{byte number}]
\]

Pages contain blocks, but we never index blocks within a page. So, careful not to mix up the concepts.
4. maps

"one to one"

page table : virtual page number ----> physical page number
            (main memory or disk)

"many to one"

TLB      : virtual page number ----> physical page number

inst. cache : physical address ----> block of instructions

data cache : physical address ----> block of data words
The tags turn a "many-to-one" function into a "one-to-one" function, namely a sub-map.
Quiz 4 grades

Quiz 4 Class Statistics

Number of submitted grades: 133 / 190
Minimum: 0%
Maximum: 100%
Average: 54.42%
Mode: 62.5%
Median: 62.5%
Standard Deviation: 28.72%

Grade Distribution

Number of Users (%)

0%
25%
50%
75%
100%
Grade Received (%)

0 %
100 %
Mean Grades Snapshot

Quiz 1    3    / 4      (each worth 4%, take best 5 of 6)
Quiz 2    2.5 / 4
Quiz 3    3   /  4
Quiz 4    2   /  4
Quiz 5
Quiz 6

A1          89 / 100    (each worth 7.5%)
A2          80 / 100
A3
A4

Final Exam     / 50   (worth 50%, option for 70%)
Announcements

- Quiz 4: yellow sticky policy
- Quiz 4 solutions on the public web page.
- Quiz 5 is on Monday. It will cover the Memory lectures (16-18)
- Exercises 6
- A4 (hopefully next week)
- If you need help, please use mycourses discussion board (not facebook)

See me. I am available. I reply to email.