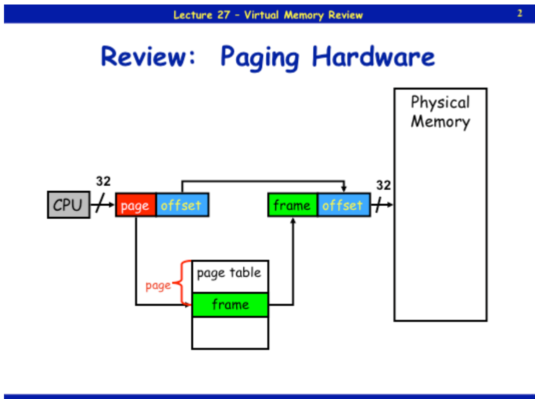
Board Notes on Memory Hierarchies, I/O, and Storage

Announcements:

- Please fill out CIF!
- Updated grades posted yesterday
- HW 7 + Lab 6 will be returned at the end of class.
- Lab feedback
  - Perhaps at the end of class today depending on time
  - If not, next lecture (where it fits in better).
- Next lecture is…
  - Introduction to parallel processing / multi-core
- Last lecture is…
  - Final exam review (like the midterm review)
- Today is…
  - A review of VM
  - A discussion of the rest of the memory / storage hierarchy

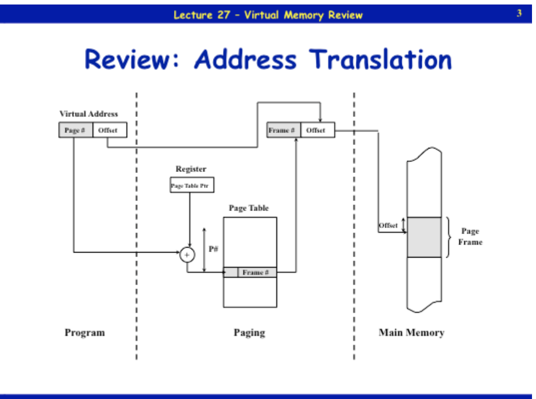
Began by explaining where this fits into course goals.

VM Review:


Lecture 27 - Virtual Memory Review 2

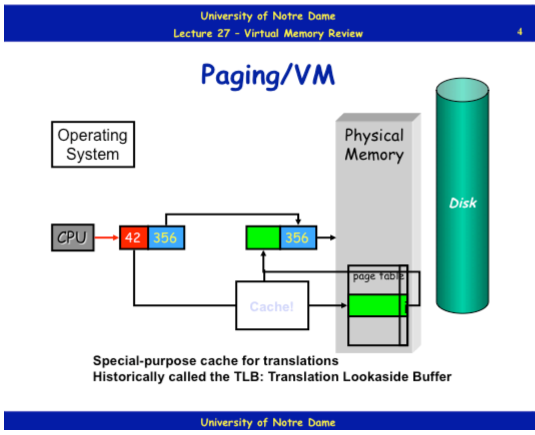
Review: Paging Hardware

University of Notre Dame


Lecture 27 - Virtual Memory Review 3

Review: Address Translation

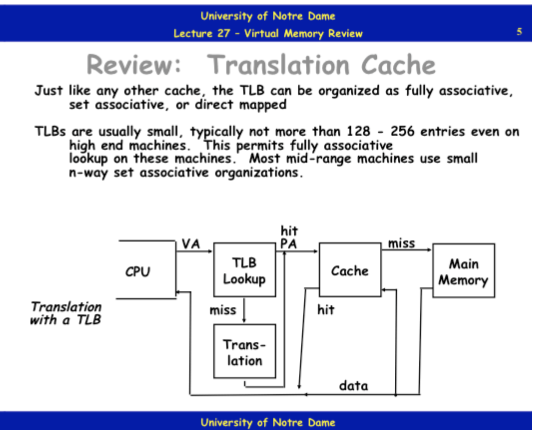
University of Notre Dame


Lecture 27 - Virtual Memory Review 4

Paging/VM

Special-purpose cache for translations
Historically called the TLB: Translation Lookaside Buffer

University of Notre Dame


Lecture 27 - Virtual Memory Review 5

Review: Translation Cache

Just like any other cache, the TLB can be organized as fully associative, set associative, or direct mapped

TLBs are usually small, typically not more than 128 - 256 entries even on high end machines. This permits fully associative lookup on these machines. Most mid-range machines use small n-way set associative organizations.

University of Notre Dame

VM Example(s):

- True or False – A miss in the $ implies a TLB miss.
  - False.
  - A cache block realistically holds much less than a page. Therefore, the block might have just been kicked out of the cache

- True or False – If we there is a Page Table miss, this means that the "page" of data we're looking for has not been loaded from disk.
  - True.
  - No entry exists for perform a physical address translation.

- Explain how the TLB helps to speed up a virtual address translation.
  - The TLB is the first place that we look when translating a virtual address to a physical address
  - It is a fast cache for the page table
  - It is indexed by the VPN from the VA
  - If the VPN from the VA matches the VPN associated with any one of the entries in the TLB (and the valid bit is set) we have a hit
  - The data supplied by the TLB is the physical frame number.
  - I.e.

Index	Data
…	…
VPN	PFN

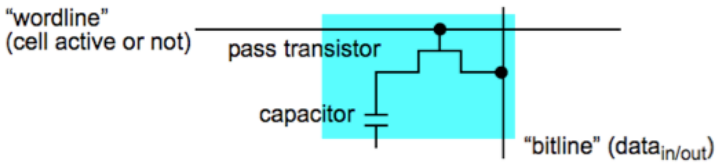
- If pages have 2^{13} addressable locations, how much coverage does a 128 entry TLB provide?
  - Each VPN is associated with 2^{13} addressable entries
  - Thus, if each TLB entry holds 1 VPN, then $2^7 \times 2^{13} = 2^{20}$ – which implies that ~1M addressable entries are covered by the entries in the TLB.

- What if…
  - Your CPU supplies the 32-bit virtual address: A C 3 0 1 0 9 7 $_{16}$
  - Pages have 214 addressable entries
  - The contents of the current Page Table Register are: 0 0 0 0 0 0 0 C $_{16}$
  - Where in physical memory do we look for the PFN?
    - Virtual address is: 1010 1100 0011 0000 00**01 0000 1001 0111**
      - (Portion in **bold** is the offset)
    - Our VPN is: 1010 1100 0011 0000 00
    - We need to add the VPN to the PTR
      - i.e. 1010 1100 0011 0000 00 + ….0C
    - Thus, the physical address of the *page table entry* that we want is:
      - … 1010 1100 0011 0011 00

- If a 32-bit virtual address is translated to a 28-bit physical address, 4 bits are used to keep track of LRU status, and each page table entry has a valid and a dirty bit, how many bits does each PT entry hold?
  - PFN = 28 bits – 14 bits = 14 bits
  - Thus, PTE = 14 bits + 4 bits + 1 bit + 1 bit = 20 bits

Next, complete discussion of memory hierarchy:

- Talk a little bit about the impact of technology on memory
  - This should help to explain why we see the register, L1 $, L2 $, main memory latencies that we have assumed thus far.
- A few words on how DRAM works:

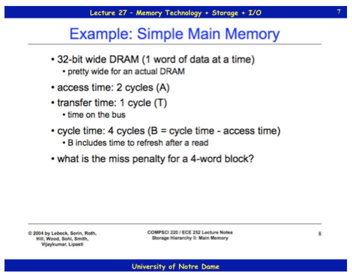
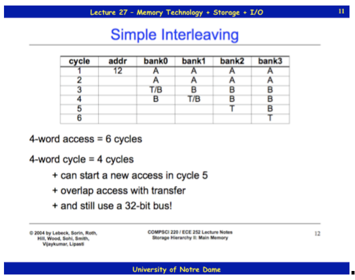


  - For <u>write</u>:
    - Information is placed on the Bit Line (BL), the word line (WL) is turned on. (This "closes" the transistor switch)
    - Depending on the data value capacitance (the data) is either charged or discharged
      - The presence of charge is equal to a 1; the absence of charge implies a logic 0.
  - For <u>read</u>:
    - Prior to reading, the bit line is charged up
    - Then, the WL is turned on, and a "charge redistribution" takes place between the BL and the storage capacitor
    - Voltage change determines the value of the stored data
  - A read is destructive
    - Must rewrite data post-read.
    - In fact, need to refresh every 2 ms or so b/c charge leaks off
      - Data is not available during that time
  - Challenges for DRAM
    - Fitting acceptably large capacitances (to have a sufficiently strong 1 and 0) into a smaller area – want strong 1/0 but also a physically small 1/0
    - Can make denser by making taller
      - Note: This is why it's hard to put DRAM and logic on the same chip. How the transistors are made is quite different

Memory can be organized in very different ways…

- Seen handout that looks at memory organization

Discuss memory organization more formally…

- See slides  to .

Lecture 27 - Memory Technology + Storage + I/O 7

Example: Simple Main Memory

- 32-bit wide DRAM (1 word of data at a time)
  - pretty wide for an actual DRAM
- access time: 2 cycles (A)
- transfer time: 1 cycle (T)
  - time on the bus
- cycle time: 4 cycles (B = cycle time - access time)
  - B includes time to refresh after a read
- what is the miss penalty for a 4-word block?

© 2004 by Lebeck, Sorin, Roth, Hill, Wood, Sohi, Smith, Vijaykumar, Lipasti
COMPSCI 220 / ECE 252 Lecture Notes Storage Hierarchy II: Main Memory 8

University of Notre Dame

Lecture 27 - Memory Technology + Storage + I/O 11

Simple Interleaving

cycle	addr	bank0	bank1	bank2	bank3
1	12	A	A	A	A
2		A	A	A	A
3		T/B	B	B	B
4		B	T/B	B	B
5				T	B
6					T

4-word access = 6 cycles

4-word cycle = 4 cycles

+ can start a new access in cycle 5
+ overlap access with transfer
+ and still use a 32-bit bus!

© 2004 by Lebeck, Sorin, Roth, Hill, Wood, Sohi, Smith, Vijaykumar, Lipasti
COMPSCI 220 / ECE 252 Lecture Notes Storage Hierarchy II: Main Memory 12

University of Notre Dame

Bandwidth between processor and memory can impact how your banks should be organized.

- Example:
  - Assume DRAM takes 120 ns to access, have 64-bit banks
  - Assume a 4 ns clock with no cache, and 1 64-bit reference / CC
  - If you need 64 bits each 4 ns CC, how many banks of memory do you need?
  - Answer:
    - Need: 64 bits / 4 ns = 16 bits / ns
    - Deliver: (64 bits / 120 ns) * # of banks
    - Therefore: (16 bits / ns) = (64 bits / 120 ns) x (# of banks)
      - # of banks = 30
      -

See slide on processor-memory integration.

Discuss Disk – I / 0.

- Finished slides 1st
- Then did written examples