

## CS 61C: Great Ideas in Computer Architecture (Machine Structures)

### Traps, Exceptions

Instructors:  
Randy H. Katz  
David A. Patterson

<http://inst.eecs.Berkeley.edu/~cs61c/fa10>

11/19/10 Fall 2010 -- Lecture #35 1

## Review

- Can separate Memory Management into orthogonal functions:
  - *Translation* (mapping of virtual address to physical address)
  - *Protection* (permission to access word in memory)
  - But most modern systems provide support for all functions with single page-based system
- All desktops/servers have full demand-paged virtual memory
  - Portability between machines with different memory sizes
  - Protection between multiple users or multiple tasks
  - Share small physical memory among active tasks
  - Simplifies implementation of some OS features
- Hardware support: User/Supervisor Mode, invoke Supervisor Mode, TLB, Page Table Register

2

## A Problem in the Early Sixties

- There were many applications whose data could not fit in the main memory, e.g., payroll
  - *Paged memory system reduced fragmentation but still required the whole program to be resident in the main memory*
- Programmers moved the data back and forth from the diskstore by *overlaying* it repeatedly on the primary store

*tricky programming!*

11/19/10 Fall 2010 -- Lecture #34 3

## Demand Paging in Atlas (1962)

“A page from secondary storage is brought into the primary storage whenever it is (implicitly) demanded by the processor.”  
*Tom Kilburn*

Primary  
32 Pages  
512 words/page

Secondary  
(~disk)  
32x6 pages

Primary memory as a *cache* for secondary memory

User sees 32 x 6 x 512 words of storage

11/19/10 Fall 2010 -- Lecture #34 4

## Demand Paging Scheme

- On a *page fault*:
  - Input transfer into a free page is initiated
  - If no free page is left, a *page is selected to be replaced* (based on usage)
  - The replaced page is written on the disk
    - to minimize disk latency effect, the first empty page on the disk was selected
  - The *page table is updated* to point to the new location of the page on the disk

11/19/10

Fall 2010 -- Lecture #34

5

## Notes on Page Table

- Solves Fragmentation problem: all chunks same size, so all holes can be used
- OS must reserve “*Swap Space*” on disk *for each process*
- To grow a process, ask Operating System
  - If unused pages, OS uses them first
  - If not, OS swaps some old pages to disk
  - (Least Recently Used to pick pages to swap)
- How/Why grow a process?

11/19/10

Fall 2010 -- Lecture #34

6

## Impact on TLB

- Need to keep track of whether page needs to be written back to disk if its been modified
- There is a “*Page Dirty Bit*” in TLB that is set when any data in page is written
- When TLB entry replaced, corresponding Page Dirty Bit is set in Page Table Entry

11/19/10

Fall 2010 -- Lecture #35

7

## Hardware/Software Support for Memory Protection

- Different tasks can share parts of their virtual address spaces
  - But need to protect against errant access
  - Requires OS assistance
- Hardware support for OS protection
  - Privileged supervisor mode (aka kernel mode)
  - Privileged instructions
  - Page tables and other state information only accessible in supervisor mode
  - System call exception (e.g., syscall in MIPS)

Fall 2010 -- Lecture #34

8

## Modern Virtual Memory Systems

*Illusion of a large, private, uniform store*

**Protection & Privacy**  
 several users, each with their private address space and one or more shared address spaces  
 page table = name space

**Demand Paging**  
 Provides the ability to run programs larger than the primary memory  
 Hides differences in machine configurations

*The price is address translation on each memory reference  
 But disk sooooo slow that don't run if going to disk all the time*

9

## Another View of Virtual Memory

- Use main memory as a “cache” for secondary (disk) storage
  - Managed jointly by CPU hardware and the operating system (OS)
- Programs share main memory
  - Each gets a private virtual address space holding its frequently used code and data
  - Protected from other programs
- CPU and OS translate virtual addresses to physical addresses
  - VM “block” is called a page
  - VM translation “miss” is called a page fault

Chapter 5 — Large and Fast: Exploiting Memory Hierarchy  
— 10

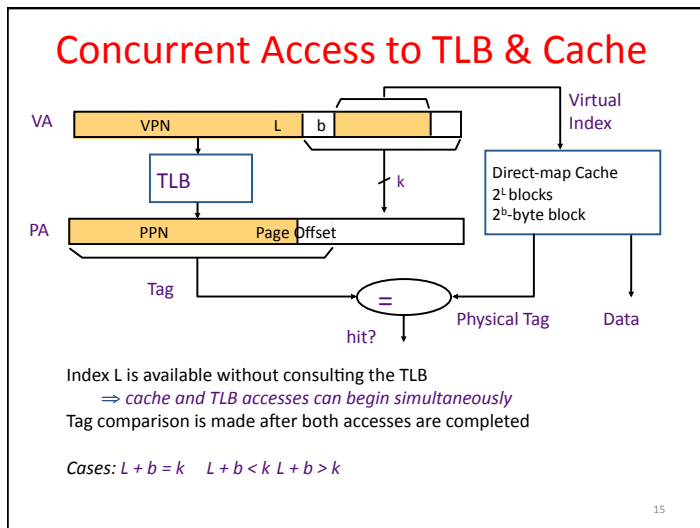
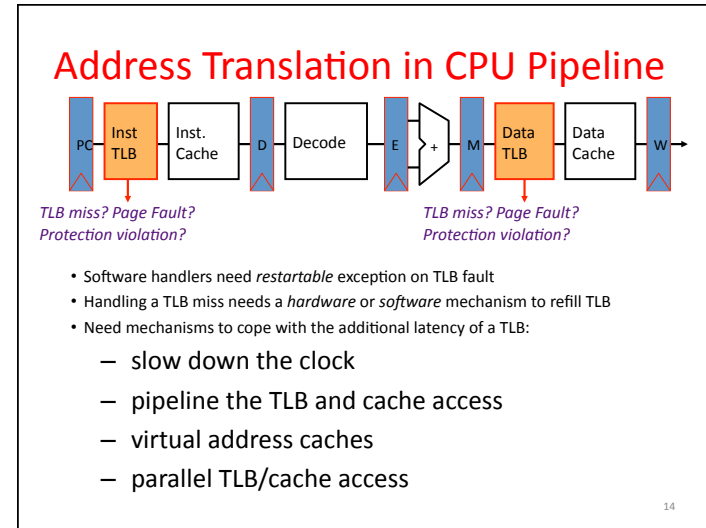
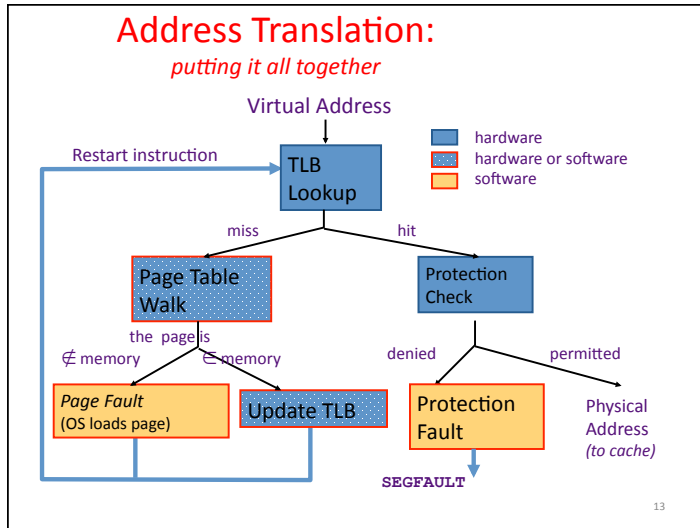
## Another View of Virtual Hierarchy

Upper Level  
Faster  
Larger  
Lower Level

## Caching vs. Demand Paging

<p><i>Caching</i></p> <ul style="list-style-type: none"> <li>cache entry</li> <li>cache block (~32 bytes)</li> <li>cache miss rate (1% to 20%)</li> <li>cache hit (~1 cycle)</li> <li>cache miss (~100 cycles)</li> <li>a miss is handled in <i>hardware</i></li> </ul>	<p><i>Demand paging</i></p> <ul style="list-style-type: none"> <li>page frame</li> <li>page (~4K bytes)</li> <li>page miss rate (&lt;0.001%)</li> <li>page hit (~100 cycles)</li> <li>page miss (~5M cycles)</li> <li>a miss is handled mostly in <i>software</i></li> </ul>
---	--

12



- ### Administrivia
- As of today, made 1 pass over all Big Ideas in Computer Architecture
  - Following lectures go into more depth on topics you've already seen while you work on projects
    - 1 on Protection, Traps
    - 2 on Virtual Memory, TLB, Virtual Machines
    - 1 on Economics of Cloud Computing
    - 1.5 on Anatomy of a Modern Microprocessor (Sandy Bridge, latest microarchitecture from Intel)
- 16

### Administrivia

- Project 4: Single Cycle Processor in Logicsim
  - Due Part 2 due Saturday 11/27
  - Face-to-Face : Signup for 15m time slot 11/30, 12/2
- Extra Credit: Fastest Project 3 (due 11/29)
- Final Review: Mon Dec 6, 2-5PM (10 Evans)
- Final: Mon Dec 13 8AM-11AM (220 Hearst Gym)
  - Like midterm: T/F, M/C, short answers
  - Whole Course: readings, lectures, projects, labs, hmwks
  - Emphasize 2<sup>nd</sup> half of 61C + midterm mistakes

11/19/10

Fall 2010 -- Lecture #35

17

### Peer Instruction: Match the Phrase

Match the memory hierarchy element on the left with the closest phrase on the right:

- |                |                                   |
|----------------|-----------------------------------|
| 1. L1 cache    | a. A cache for page table entries |
| 2. L2 cache    | b. A cache for a main memory      |
| 3. Main memory | c. A cache for disks              |
| 4. TLB         | d. A cache for a cache            |

- |                       |                       |
|-----------------------|-----------------------|
| A) 1 a, 2 b, 3 c, 4 d | E) 1 b, 2 d, 3 c, 4 a |
| B) 1 a, 2 b, 3 d, 4 c | F) 1 d, 2 b, 3 a, 4 c |
| C) 1 b, 2 d, 3 a, 4 c | G) 1 d, 2 a, 3 b, 4 c |
| D) 1 d, 2 b, 3 c, 4 a | H) 1 d, 2 c, 3 b, 4 a |

11/19/10

Fall 2010 -- Lecture #30

18

### Peer Instruction: Match the Phrase

Match the memory hierarchy element on the left with the closest phrase on the right:

- |                |                                   |
|----------------|-----------------------------------|
| 1. L1 cache    | a. A cache for page table entries |
| 2. L2 cache    | b. A cache for a main memory      |
| 3. Main memory | c. A cache for disks              |
| 4. TLB         | d. A cache for a cache            |

- |                       |                       |
|-----------------------|-----------------------|
| A) 1 a, 2 b, 3 c, 4 d | E) 1 b, 2 d, 3 c, 4 a |
| B) 1 a, 2 b, 3 d, 4 c | F) 1 d, 2 b, 3 a, 4 c |
| C) 1 b, 2 d, 3 a, 4 c | G) 1 d, 2 a, 3 b, 4 c |
| D) 1 d, 2 b, 3 c, 4 a | H) 1 d, 2 c, 3 b, 4 a |

11/19/10

Fall 2010 -- Lecture #30

19

### Peer Instruction: True or False

A program tries to load a word X that causes a TLB miss but not a page fault. Which are True or False:

1. A TLB miss means that the page table does not contain a valid mapping for virtual page corresponding to the address X
2. There is no need to look up in the page table because there is no page fault
3. The word that the program is trying to load is present in physical memory.

- |                  |                  |
|------------------|------------------|
| A) 1 F, 2 F, 3 F | E) 1 T, 2 F, 3 F |
| B) 1 F, 2 F, 3 T | F) 1 T, 2 F, 3 T |
| C) 1 F, 2 T, 3 F | G) 1 T, 2 T, 3 F |
| D) 1 F, 2 T, 3 T | H) 1 T, 2 T, 3 T |

11

20

### Peer Instruction: True or False

A program tries to load a word X that causes a TLB miss but not a page fault or protection violations. Which are True or False:

1. A TLB miss means that the page table does not contain a valid mapping for virtual page corresponding to the address X
2. There is no need to look up the page table because there is no page fault
3. The word that the program is trying to load is present in physical memory.

- |                                |                  |
|--------------------------------|------------------|
| A) 1 F, 2 F, 3 F               | E) 1 T, 2 F, 3 F |
| <b>B) 1 F, 2 F, 3 T</b>        | F) 1 T, 2 F, 3 T |
| C) 1 F, 2 T, 3 F               | G) 1 T, 2 T, 3 F |
| <sup>11</sup> D) 1 F, 2 T, 3 T | H) 1 T, 2 T, 3 T |

21

### Peer Instruction: True or False

TLBs entries have valid bits and dirty bits. Data caches have them also.

- A. The valid bit means the same in both: if valid = 0, it must miss in both TLBs and Caches.
- B. The valid bit has different meanings. For caches, it means this entry is valid if the address requested matches the tag. For TLBs, it determines whether there is a page fault (valid=0) or not (valid=1).
- C. The dirty bit means the same in both: the data in this block in the TLB or Cache has been changed.
- D. The dirty bit has different meanings. For caches, it means the data block has been changed. For TLBs, it means that the page corresponding to this TLB entry has been changed.

- |                              |
|------------------------------|
| A) 1 F, 2 T, 3 F, 4 T        |
| B) 1 F, 2 T, 3 T, 4 F        |
| <b>C) 1 T, 2 F, 3 F, 4 T</b> |
| D) 1 T, 2 F, 3 T, 4 F        |

11/19/10

Fall 2010 -- Lecture #30

22

### Peer Instruction: True or False

TLBs entries have valid bits and dirty bits. Data caches have them also.

- A. The valid bit means the same in both: if valid = 0, it must miss in both TLBs and Caches.
- B. The valid bit has different meanings. For caches, it means this entry is valid if the address requested matches the tag. For TLBs, it determines whether there is a page fault (valid=0) or not (valid=1).
- C. The dirty bit means the same in both: the data in this block in the TLB or Cache has been changed.
- D. The dirty bit has different meanings. For caches, it means the data block has been changed. For TLBs, it means that the page corresponding to this TLB entry has been changed.

- |                              |
|------------------------------|
| A) 1 F, 2 T, 3 F, 4 T        |
| B) 1 F, 2 T, 3 T, 4 F        |
| <b>C) 1 T, 2 F, 3 F, 4 T</b> |
| D) 1 T, 2 E, 3 T, 4 F        |

11/19/10

Fall 2010 -- Lecture #30

23

### “And In Conclusion”

- Virtual Memory, Paging really used for Protection, Translation, Some OS optimizations
- Not really routinely paging to disk
- Can think of as another level of memory hierarchy, but not really used like caches today

11/19/10

Fall 2010 -- Lecture #35

24

## Acknowledgements

- These slides contain material developed and copyright by:
  - Arvind (MIT)
  - Krste Asanovic (MIT/UCB)
  - Joel Emer (Intel/MIT)
  - James Hoe (CMU)
  - John Kubiatowicz (UCB)
  - David Patterson (UCB)
- MIT material derived from course 6.823
- UCB material derived from course CS252

25