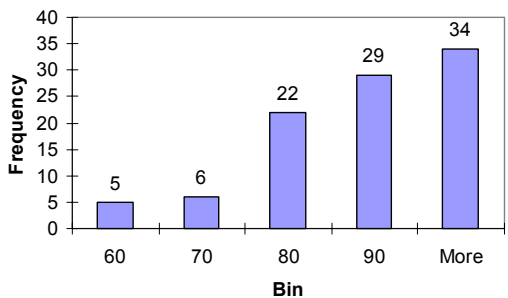


## Virtual Memory (cont'd)

## Topics

- Reading Assignment
  - Read sections 7.6-7.7 in the textbook
- Exam results
- Grading criteria for class
- Multilevel caches
- Virtual Memory review
  - Speeding up address translation

Exam 2



## Exam Regrades

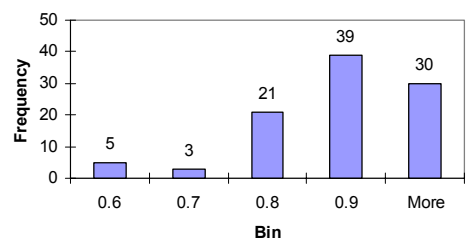
- One week (until Friday, Nov. 17)
  - Submit written request/explanation
  - We'll handle the request within 48 hours
- Contacting us for regrades:
  - Email TAs ([cs333@cs.virginia.edu](mailto:cs333@cs.virginia.edu)) for an appointment.
  - See TAs in office hours
    - Office hours next week are the scheduled lab hours and in lab locations

## Grading Criteria

- |                           |                                |
|---------------------------|--------------------------------|
| • Current                 | • Proposed                     |
| Midterms (2) – 50%        | Midterms (2) – <b>40%</b>      |
| Homework and Labs – 20%   | (15% + 25%)                    |
| Final Exam – 20%          | Homework and Labs – <b>30%</b> |
| Class Participation – 10% | Final Exam – 20%               |
|                           | Class Participation – 10%      |

## With Proposed Grading Criteria

Overall Grade



## Opportunities to Boost Your Grade

- Extra credit assignments, preliminary list
  1. Pipelining a datapath (in SMOK)
    - Take lab 4 solution and add the hardware and support to change the datapath to a 2-stage pipelined datapath (+ submit a report)
  2. Add more hardware support for more SRC instructions
    - Any instructions not supported by current Lab 4 datapath (+ submit report)
  3. Investigate architecture XYZ, brief report to the class on some interesting information about it (+ submit report).
    - Itanium, Opteron, Sony CELL, Pentium 4 anything..
    - Talk about interesting ISA features, datapath design, caches, virtual memory, other interesting features

## Multilevel Caches

## Multilevel Caches

- Memory Hierarchy
  - Level 1 (L1) cache
    - Improve hit speed
  - Level 2 (L2) cache
    - Reduce L1 miss penalty
    - Only if a miss occurs in L2, will the long penalty for memory access occur

## Multilevel Cache Performance

- Processor with base performance of **1.0 CPI**
  - Assumes all references hit in the L1 cache and clock rate of **5 GHz**
- Main memory access time **100 ns** (includes miss handling)
- Question: If the miss rate per instruction at the **L1 cache is 2%**, how much faster will the processor be if an L2 cache is added (**5 ns access time (hits and misses), 0.5% miss rate to main memory**) is added

## Class Participation

## Class Participation 1

- 7.19
  - 2-way set associative cache (4ns access time, 60 ns miss time)
  - Without cache, main memory access time was 50 ns
  - Benchmark runs suggest 90% speedup

  1. What is the approximate hit ratio?

## Class Participation 2

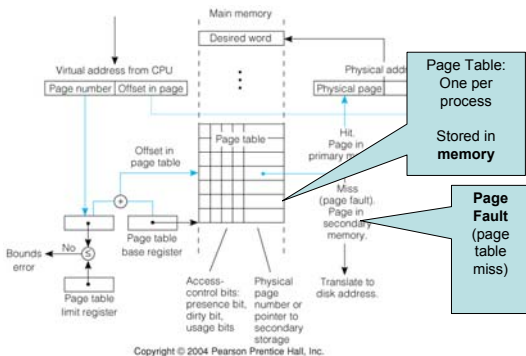
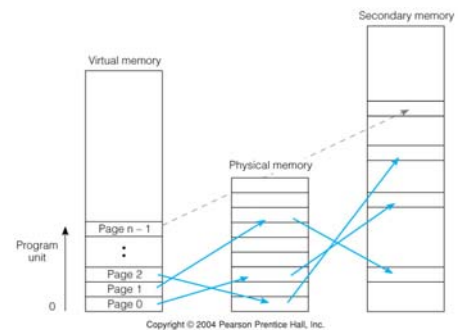
- 7.20
  - 128 MB main memory
  - 64 KB direct-mapped cache
  - 16 bytes per line
  - byte addressable
- 1. How many lines are in the cache?
- 2. How is the main memory address partitioned? (How many bits for: tag, index, offset)

## Class Participation 3

- 7.21
    - 128 MB main memory
    - 2 MB cache
    - Blocks are 32 bytes in size
    - byte addressable
- Show the number of bits for tag, index, offset for:
1. fully associative
  2. direct-mapped
  3. 8-way set associative

## Virtual Memory

## Virtual Memory



## Choosing Page Sizes

- Smaller page size
  - May increase number of page faults
    - More memory traffic
  - Increases size of page table (takes up more memory space)
- One solution: multilevel page tables
  - Highest level page table entry contains pointers to other page tables (tree)

## Question

- Memory is much cheaper now than in 1960s, do we still need virtual memory?

## Speeding up Address Translation

- Translation for each memory reference
- Each memory reference requires:
  - Page table access
    - to get page table entry
  - Access the memory location given the physical address calculated

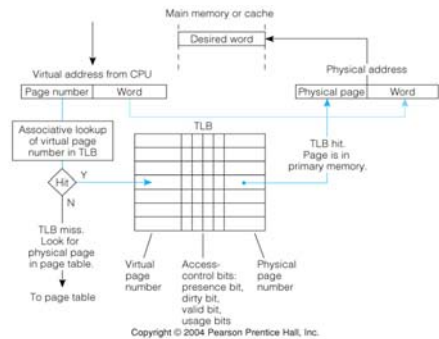
2 memory accesses == very slow  
Can this be sped up?

## Translation Lookaside Buffer (TLB)

- Small cache (hardware)
  - stores most recent page table references
  - generally fully-associative
- TLB entry
  - contains virtual to physical page number translation

Virtual Page number	A, P, D, U, V bits	Physical Page number
---------------------	--------------------	----------------------

## TLB



## TLB, Caches, Main Memory, Disk

