



# CS 152 Computer Architecture and Engineering

## Lecture 11 - Virtual Memory and Caches

Krste Asanovic

Electrical Engineering and Computer Sciences  
University of California at Berkeley

<http://www.eecs.berkeley.edu/~krste>

<http://inst.eecs.berkeley.edu/~cs152>



## Today is a review of last two lectures

- Translation/Protection/Virtual Memory
- This is complex material - often takes several passes before the concepts sink in
- Try to take a different path through concepts today

## VM features track historical uses:

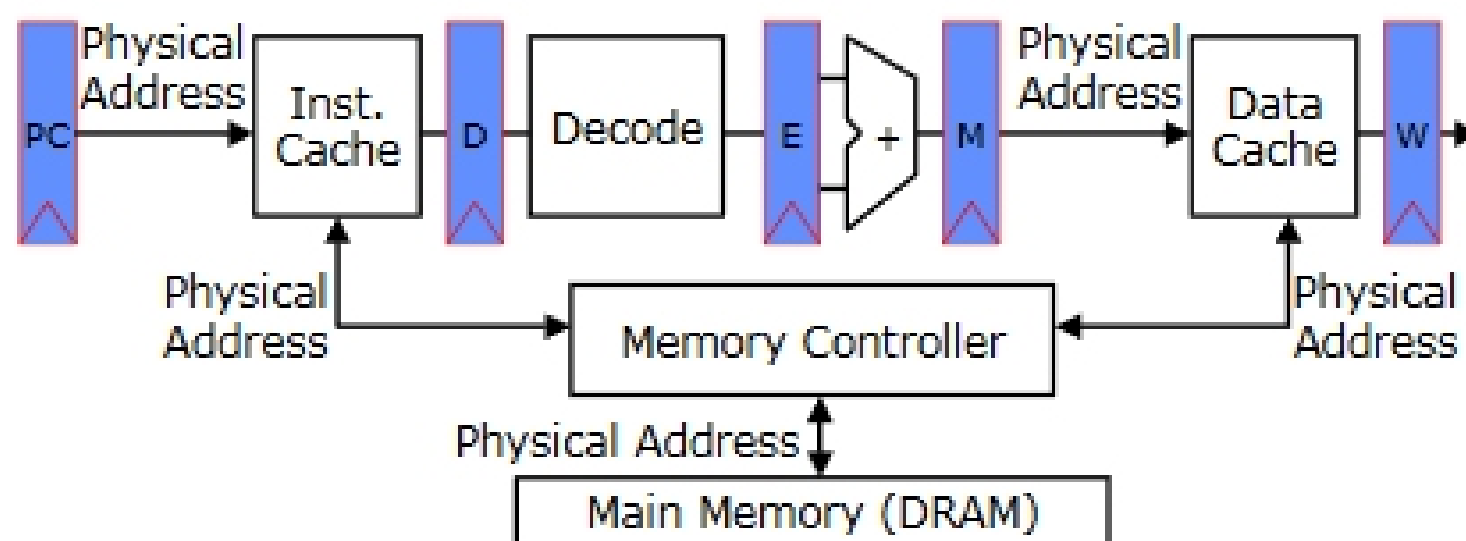
- Bare machine, only physical addresses
  - One program owned entire machine
- Batch-style multiprogramming
  - Several programs sharing CPU while waiting for I/O
  - Base & bound: translation and protection between programs (not virtual memory)
  - Problem with external fragmentation (holes in memory), needed occasional memory defragmentation as new jobs arrived
- Time sharing
  - More interactive programs, waiting for user. Also, more jobs/second.
  - Motivated move to fixed-size page translation and protection, no external fragmentation (but now internal fragmentation, wasted bytes in page)
  - Motivated adoption of virtual memory to allow more jobs to share limited physical memory resources while holding working set in memory
- Virtual Machine Monitors
  - Run multiple operating systems on one machine
  - Idea from 1970s IBM mainframes, now common on laptops
    - » e.g., run Windows XP on top of Mac OS X
  - Hardware support for two levels of translation/protection
    - » Guest OS virtual -> Guest OS physical -> Host machine physical

3/5/2009

CS152-Spring'09

3

## Bare Machine



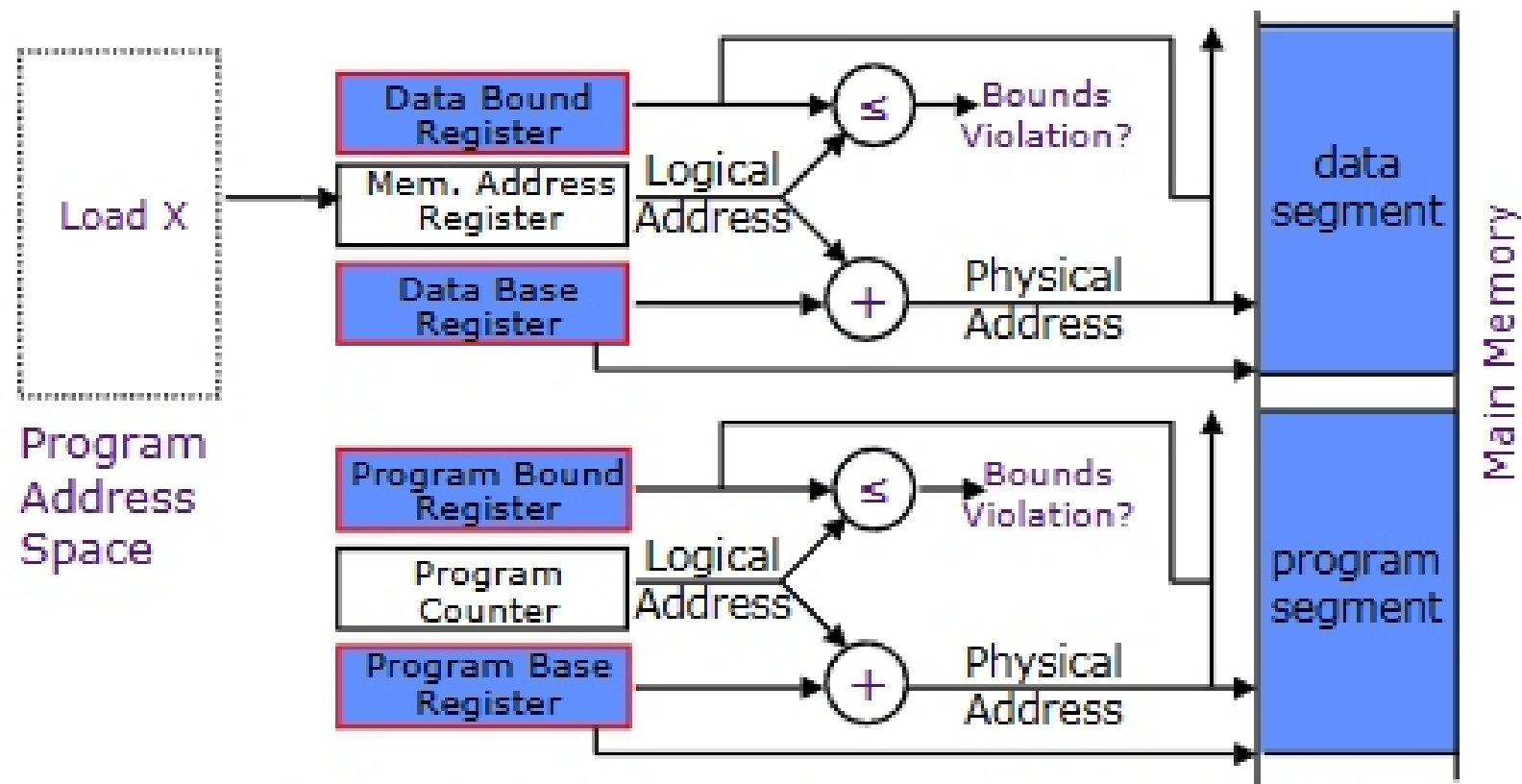
- In a bare machine, the only kind of address is a physical address

3/5/2009

CS152-Spring'09

4

## Base and Bound Scheme



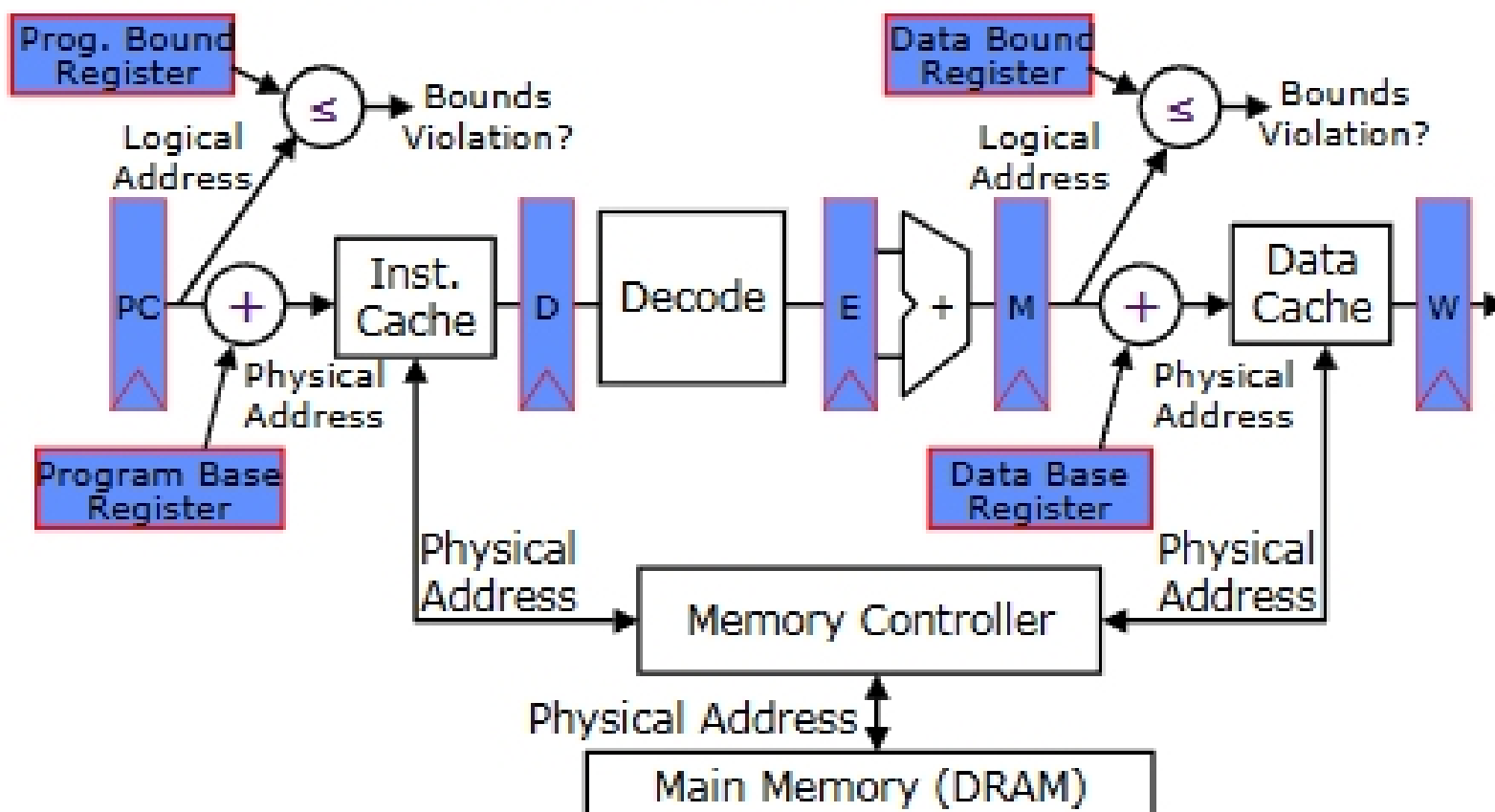
Logical address is what user software sees. Translated to physical address by adding base register.

3/5/2009

CS152-Spring'09

5

## Base and Bound Machine



[ Can fold addition of base register into (base+offset) calculation using a carry-save adder (sum three numbers with only a few gate delays more than adding two numbers) ]

3/5/2009

CS152-Spring'09

6