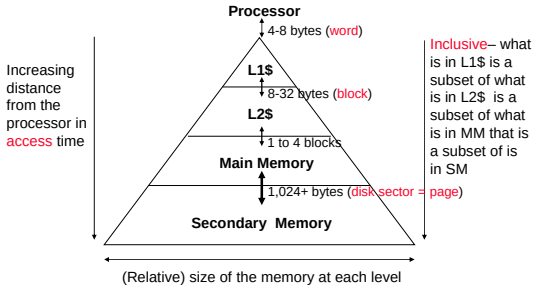
EEC 170
Computer Architecture
Fall 2005

Virtual Memory Hardware Support

Review: The Memory Hierarchy

- Take advantage of the principle of locality to present the user with as much memory as is available in the cheapest technology at the speed offered by the fastest technology



Virtual Memory

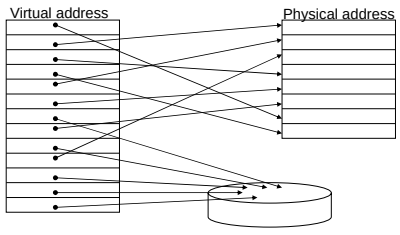
- Use main memory as a "cache" for secondary storage to
  - Allow efficient and safe sharing of memory among multiple programs
  - Provide the ability to easily run programs larger than the size of physical memory
  - Simplify loading a program for execution by providing for code relocation (i.e., the code can be loaded anywhere in main memory)
- What makes it work? – again the Principle of Locality
  - A program is likely to access a relatively small portion of its address space during any period of time
- Each program is compiled into its own address space – a virtual address space
  - During run-time each virtual address must be translated to a physical address (an address in main memory)

Two Programs in Physical Memory

- Board and Chalk Example!
  - Picture of two programs in virtual space mapping to one physical memory
  - page base address and offset within a page are required to get to an actual instruction
  - The two programs share the physical space

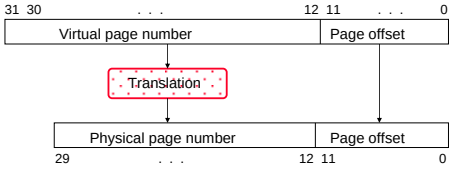
Virtual Memory Space

- A virtual address is translated to a physical address by a combination of hardware and software
  - A virtual memory block is called a page and a virtual memory miss is called a page fault



Address Translation

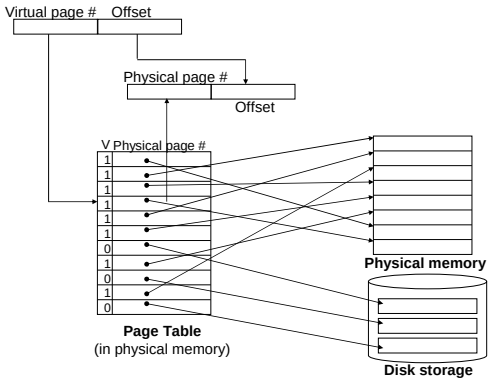
Virtual Address (VA)



Physical Address (PA)

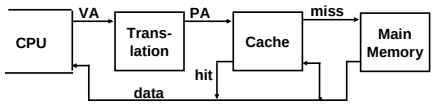
- So each memory request first requires an address translation from the virtual space to the physical space

Address Translation Mechanisms



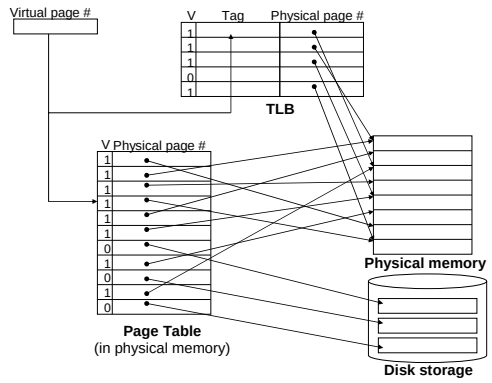
Virtual Addressing with a Cache

- It takes an extra memory access to translate a VA to a PA



- This makes memory (cache) accesses very expensive; this is the hit time that defines the clock cycle time
- The fix is to use a Translation Lookaside Buffer (TLB) – a small cache that keeps track of recently used address mappings to avoid a page table access

Making Address Translation Fast



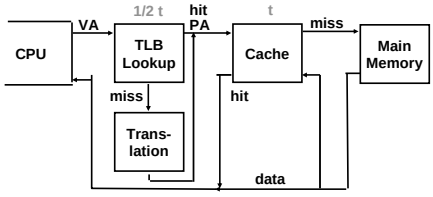
Translation Lookaside Buffers (TLBs)

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

V	Virtual Page #	Physical Page #	Dirty	Ref	Access

- TLB access time is comparable to cache access time (and much less than main memory access time)
- TLBs are usually small, typically not more than 128 to 256 entries even on high end machines

A TLB in the Memory Hierarchy



- TLB miss – is it a page fault or merely a TLB miss?
  - If the page exists in memory, then the TLB miss can be handled (in hardware or software) by loading the translation information from the page table into the TLB
  - If the page is not in memory, the it's a true page fault
  - TLB misses are much more frequent than true page faults

Some Virtual Memory Design Parameters

	Paged VM	TLBs
Total size (blocks)	16,000 to 250,000	16 to 512
Total size (KB)	250,000 to 1,000,000,000	0.25 to 16
Block size (B)	4000 to 64,000	4 to 32
Miss penalty (clocks)	10,000,000 to 100,000,000	10 to 1000
Miss rates	0.00001% to 0.0001%	0.01% to 2%

Two Machines' Cache Parameters

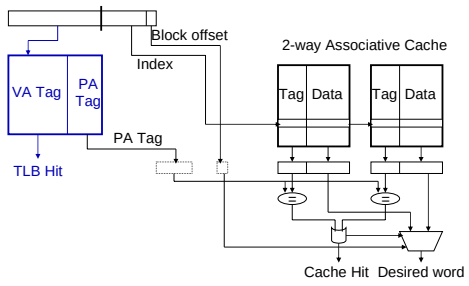
	Intel P4	AMD Opteron
TLB organization	1 TLB for instructions and 1TLB for data Both 4-way set associative Both use ~LRU replacement Both have 128 entries TLB misses handled in hardware	2 TLBs for instructions and 2 TLBs for data Both L1 TLBs fully associative with ~LRU replacement Both L2 TLBs are 4-way set associative with round-robin LRU Both L1 TLBs have 40 entries Both L2 TLBs have 512 entries TBL misses handled in hardware

TLB Event Combinations

TLB	Page Table	Cache	Possible? Under what circumstances?
Hit	Hit	Hit	Yes – what we want!
Hit	Hit	Miss	Yes – although the page table is not checked if the TLB hits
Miss	Hit	Hit	Yes – TLB miss, PA in page table
Miss	Hit	Miss	Yes – TLB miss, PA in page table, but data not in cache
Miss	Miss	Miss	Yes – page fault
Hit	Miss	Miss	Impossible – TLB translation not possible if page not present in memory
Hit	Miss	Hit	Impossible – TLB translation not possible if page not present in memory
Miss	Miss	Hit	Impossible – data not allowed in cache if page is not in memory

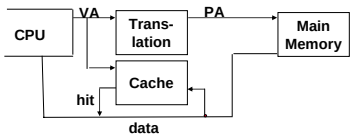
Reducing Translation Time

- Can overlap the cache access with the TLB access
  - Works because high order bits of the VA are used to access the TLB while the low order bits are used as index into cache



Why Not a Virtually Addressed Cache?

- A virtually addressed cache would only require address translation on cache misses



but

- Two different virtual addresses can map to the same physical address (processes sharing data), i.e., two different cache entries hold data for the same physical address – synonyms
  - Must update all cache entries with the same physical address or the memory becomes inconsistent

The Hardware/Software Boundary

- What parts of the virtual to physical address translation is done by or assisted by the hardware?
  - Translation Lookaside Buffer (TLB) that caches the recent translations
    - TLB access time is part of the cache hit time
    - Can allot an extra stage in the pipeline for TLB access
  - Page table storage, fault detection and updating
    - Page faults result in interrupts (precise) that are then handled by the OS
    - Hardware must support (i.e., update appropriately) Dirty and Reference bits (e.g., ~LRU) in the Page Tables
  - Disk placement
    - Bootstrap (e.g., out of disk sector 0) so the system can service a limited number of page faults before the OS is even loaded

Summary

- The Principle of Locality:
  - Program likely to access a relatively small portion of the address space at any instant of time.
    - Temporal Locality: Locality in Time
    - Spatial Locality: Locality in Space
- Caches, TLBs, Virtual Memory all understood by examining how they deal with the four questions: 1) Where can block be placed? 2) How is block found? 3) What block is replaced on miss? 4) How are writes handled?
- Page tables map virtual address to physical address
- TLBs are important for fast translation

Reminders

- Next lecture
  - Quiz on chapter 7
  - Final review
- HW #6
  - On cache and memory (chapter 7)
  - Due on Friday 12/9 at 5pm
- Final exam schedule
  - Friday December 16th at 1:30pm
  - Office hours on Thursday 12/15 3-5pm