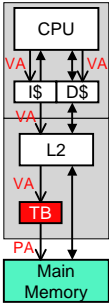
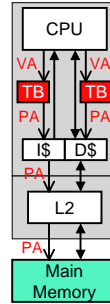


Virtual Caches



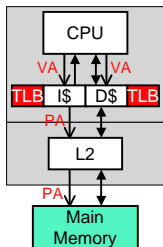
- Memory hierarchy so far: **virtual caches**
 - Indexed and tagged by VAs
 - Translate to PAs only to access memory
 - + Fast: avoids translation latency in common case
- What to do on process switches?
 - Flush caches? Slow
 - Add process IDs to cache tags
- Does inter-process communication work?
 - **Aliasing**: multiple VAs map to same PA
 - How are multiple cache copies kept in sync?
 - Also a problem for I/O (later in course)
 - Disallow caching of shared memory? Slow

Physical Caches



- Alternatively: **physical caches**
 - Indexed and tagged by PAs
 - Translate to PA at the outset
 - + No need to flush caches on process switches
 - Processes do not share PAs
 - + Cached inter-process communication works
 - Single copy indexed by PA
 - Slow: adds 1 cycle to t_{hit}

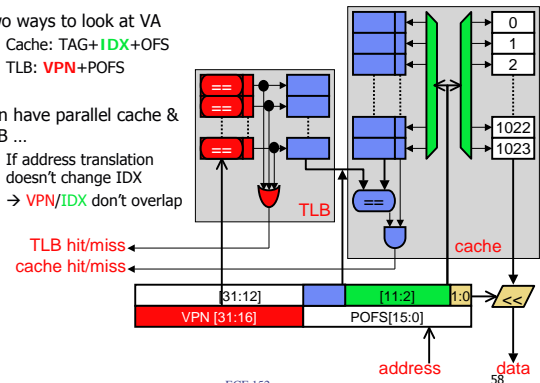
Virtual Physical Caches



- Compromise: **virtual-physical caches**
 - Indexed by VAs
 - Tagged by PAs
 - Cache access and address translation in parallel
 - + No context-switching/aliasing problems
 - + Fast: no additional t_{hit} cycles
- A TB that acts in parallel with a cache is a **TLB**
 - **Translation Lookaside Buffer**
- Common organization in processors today

Cache/TLB Access

- Two ways to look at VA
 - Cache: TAG+IDX+OFS
 - TLB: VPN+POFS
- Can have parallel cache & TLB ...
 - If address translation doesn't change IDX
 - → VPN/IDX don't overlap



Cache Size And Page Size



- Relationship between page size and L1 I\$(D\$) size
 - Forced by non-overlap between VPN and IDX portions of VA
 - Which is required for TLB access
 - I\$(D\$) size / **associativity** \leq page size
 - Big caches must be set associative
 - Big cache \rightarrow more index bits (fewer tag bits)
 - More set associative \rightarrow fewer index bits (more tag bits)
 - Systems are moving towards bigger (64KB) pages
 - To amortize disk latency
 - To accommodate bigger caches

TLB Organization

- **Like caches:** TLBs also have ABCs
 - What does it mean for a TLB to have a block size of two?
 - Two consecutive VPs share a single tag
- **Rule of thumb:** TLB should "cover" L2 contents
 - In other words: #PTEs * page size \geq L2 size
 - Why? Think about this ...

Flavors of Virtual Memory

- Virtual memory almost ubiquitous today
 - Certainly in general-purpose (in a computer) processors
 - But even some embedded (in non-computer) processors support it
- Several forms of virtual memory
 - **Paging** (aka flat memory): equal sized translation blocks
 - Most systems do this
 - **Segmentation:** variable sized (overlapping?) translation blocks
 - IA32 uses this
 - Makes life very difficult
 - **Paged segments:** don't ask

Summary

- DRAM
 - Two-level addressing
 - Refresh, access time, cycle time
- Building a memory system
 - DRAM/bus bandwidth matching
- Memory organization
- Virtual memory
 - Page tables and address translation
 - Page faults and handling
 - Virtual, physical, and virtual-physical caches and TLBs

Next part of course: I/O