CS/COE1541: Introduction to Computer Architecture

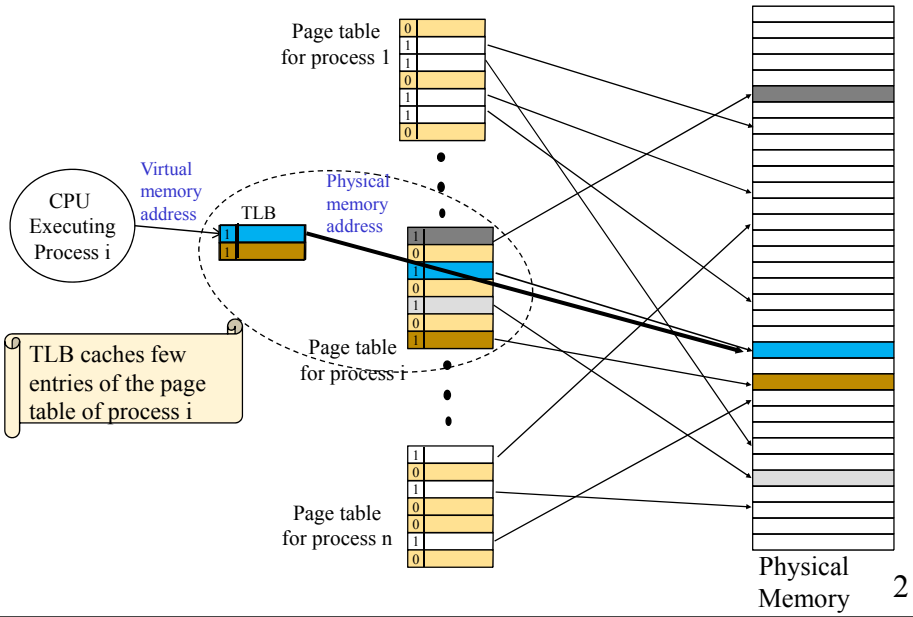
Dept. of Computer Science
University of Pittsburgh

http://www.cs.pitt.edu/~melhem/courses/1541p/index.html

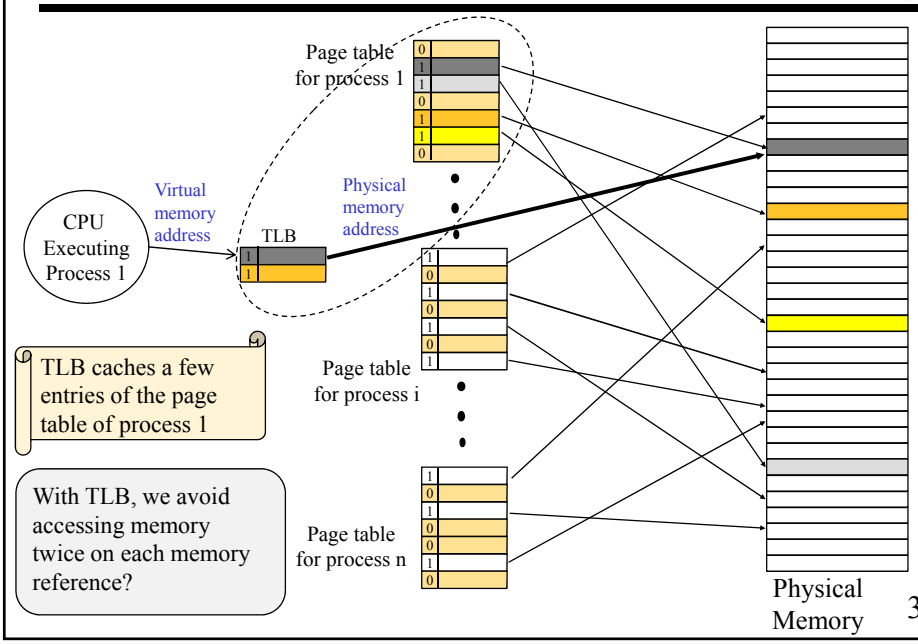
Chapter 5: Exploiting the Memory Hierarchy
Lecture 7: More on Virtual Memory

Lecturer: Rami Melhem

Translation Lookaside Buffer (TLB) to cache the page table



Caching the page table in a TLB

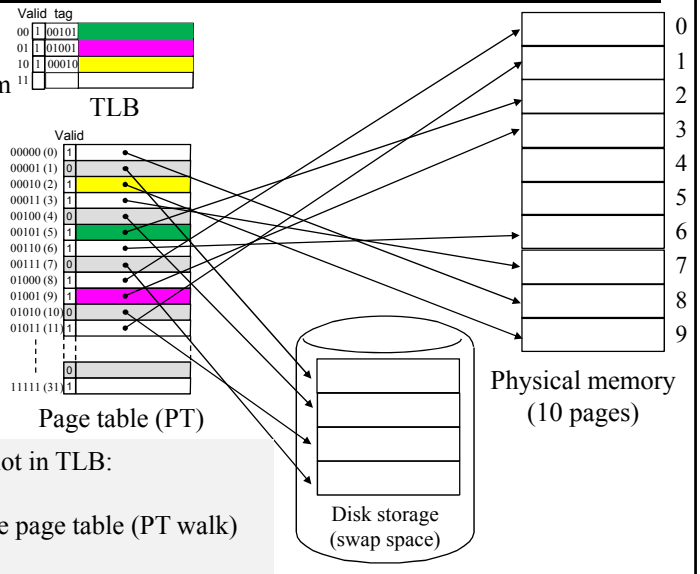


TLB caches a few entries of the page table of process 1

With TLB, we avoid accessing memory twice on each memory reference?

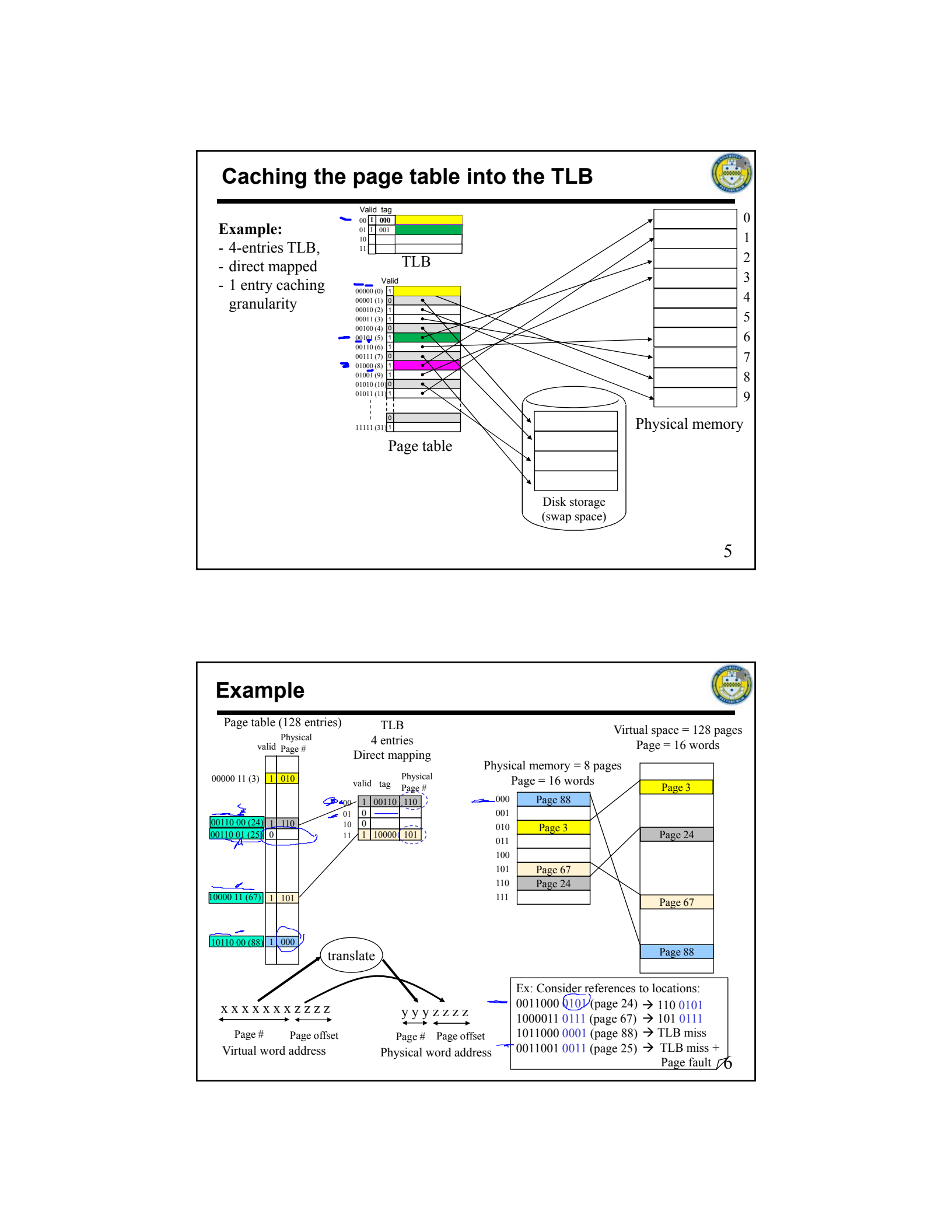
Caching the page table into the TLB

Example:
- 32-pages virtual mem
- 32-entry Page Table
- 4-entries TLB,
- Fully associative
- 1 entry caching granularity



If page table entry is not in TLB:
- TLB miss
- get the entry from the page table (PT walk)
- load it to the TLB.
- may have to replace a valid TLB entry

Caching the page table into the TLB



Example:
- 4-entries TLB,
- direct mapped
- 1 entry caching granularity

Example

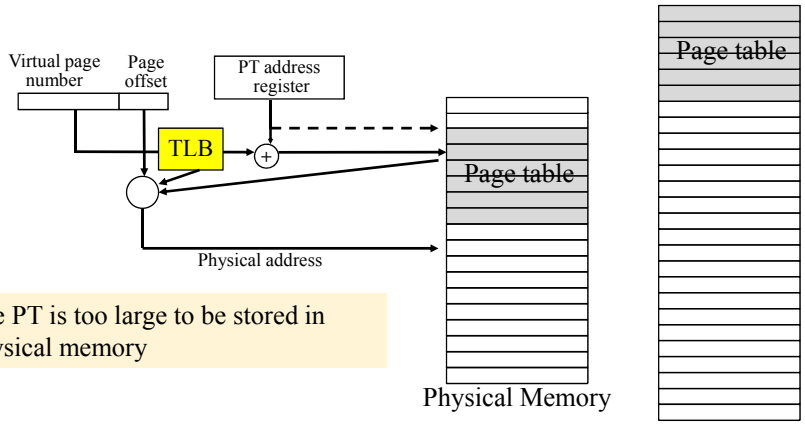
The Page Table (PT) is very large

Example: If VS = 32-bit address (4GBytes) and memory page size = 4KB

→ VS = 1 million pages, page table = 1 Million entry.

→ if each table entry = 4 bytes → page table occupies 4MB

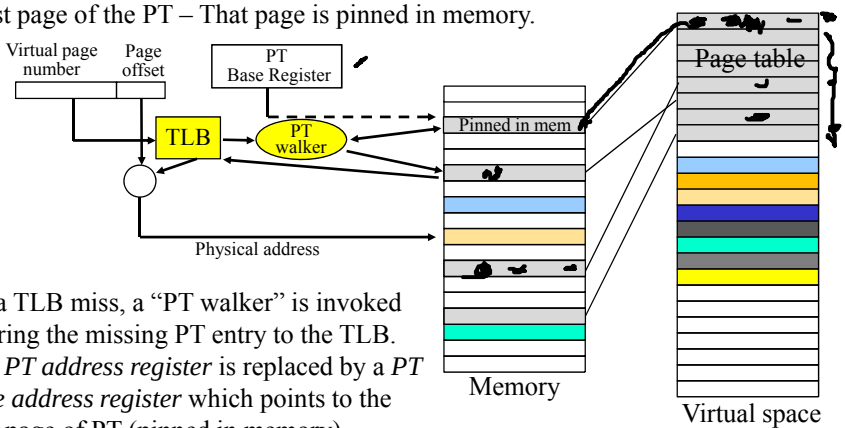
→ page table occupies 1024 memory pages (memory page size =4KB)



The PT is too large to be stored in physical memory

The Page Table (PT) is very large

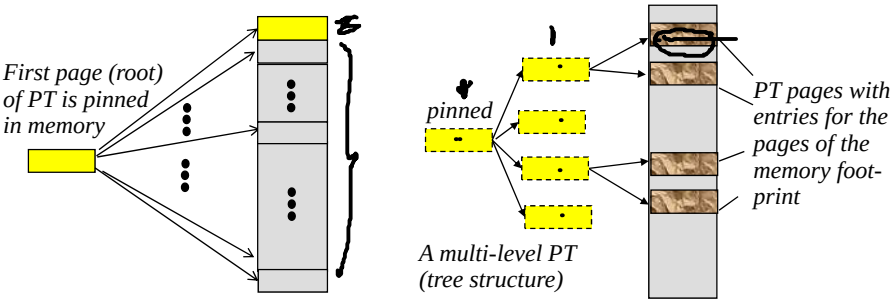
- The PT is stored in 1024 pages of the virtual memory space.
- PT's currently used pages are brought to memory, and like any other page in the virtual space, the location of a page in memory is recorded in PT
- Note that the 1024 PT entries corresponding to the pages of the PT can fit in the first page of the PT – That page is pinned in memory.



- On a TLB miss, a "PT walker" is invoked to bring the missing PT entry to the TLB.
- The *PT address register* is replaced by a *PT base address register* which points to the first page of PT (pinned in memory)

Multi level Page Tables (multi level PT)

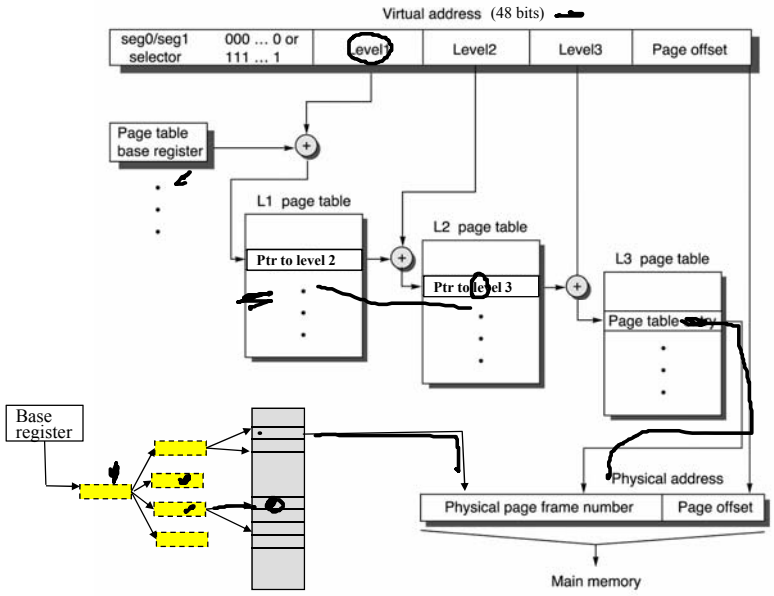
- In the example of 4GB VS and 4KB pages, the PT can be stored in 1024 pages
  - Pages of the PT are brought to memory on demand
  - The first page (root) of the PT keeps track of the locations of PT pages in memory.
- This is a "2-level" PT organization – may generalize to a multi-level PT organization
- Memory foot-print = the part of the VS which is actually used (accessed)
  - A large number of pages in the VS are not allocated or used (empty).
  - Hence a large number of entries of the PT are never accessed



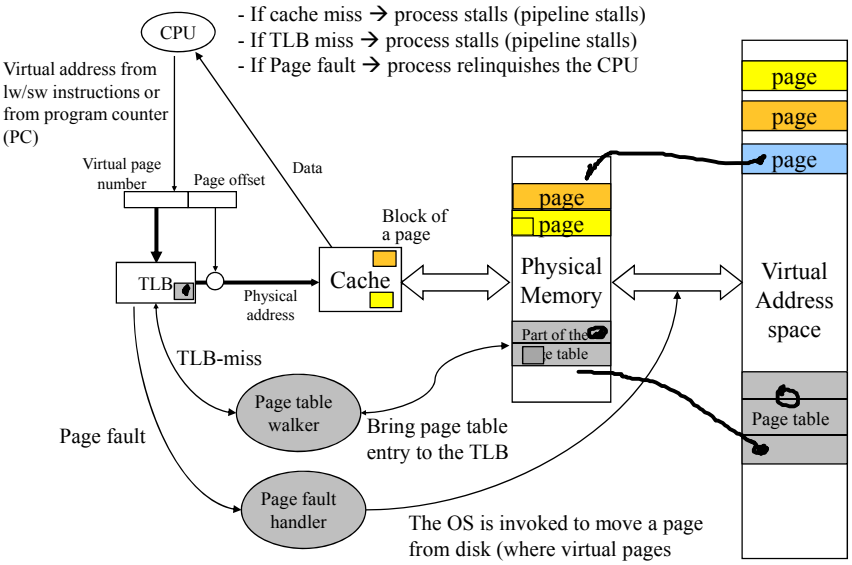
PT → stored in 1024 pages containing a million PT entries

Only a few of the 1024 PT pages are used

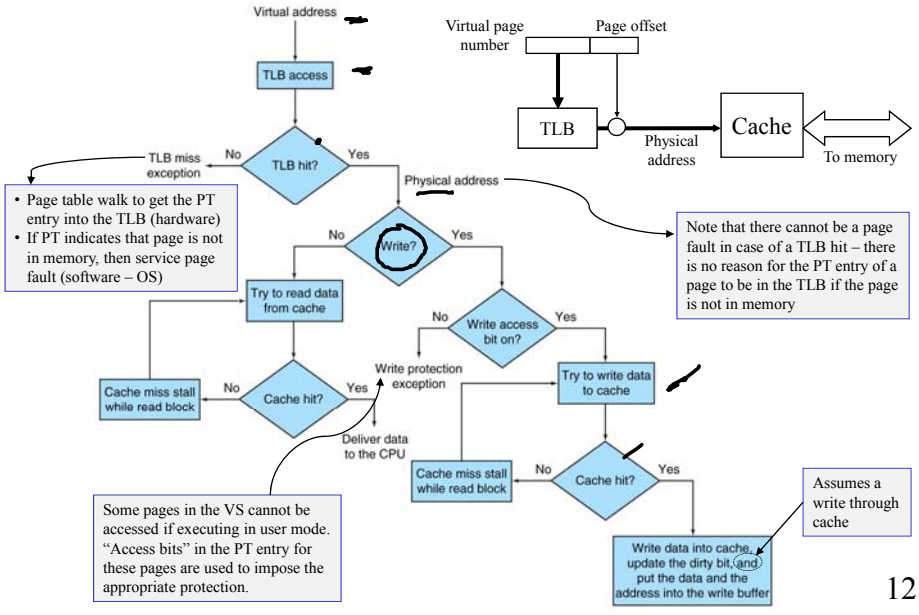
Alpha 21264 example (3-levels page tables)



The whole picture



TLBs and caches



2-Level TLB Organization for Cortex-A8 and Core-i7

Characteristic	ARM Cortex-A8	Intel Core I7
Virtual address	32 bits	48 bits
Physical address	32 bits	44 bits
Page size	Variable: 4, 16, 64 KiB, 1, 16 MiB	Variable: 4 KiB, 2/4 MiB
TLB organization	1 TLB for instructions and 1 TLB for data Both TLBs are fully associative, with 32 entries, round robin replacement TLB misses handled in hardware	1 TLB for instructions and 1 TLB for data per core Both L1 TLBs are four-way set associative, LRU replacement L1 I-TLB has 128 entries for small pages, 7 per thread for large pages L1 D-TLB has 64 entries for small pages, 32 for large pages The L2 TLB is four-way set associative, LRU replacement The L2 TLB has 512 entries TLB misses handled in hardware