

Final Review L12

VE370

Contents

- VM
- TLB

VM

- Why VM
 - VM allows programs to exceed the main memory size limit
 - CPU or OS translate **virtual addresses** to **physical addresses**
 - Protected from other programs
 - VM allows sharing of main memory among multiple programs
 - Programs are divided into self-contained, mutually exclusive small pieces – **overlay**
 - Each overlay contains both code and data
 - Each overlay is smaller than main memory
 - Overlays are loaded and swapped to fit in smaller main memory

VM

- Page
- Page fault

Page table

- Translator for program

Page fault

Page Fault

- Page Fault
 - Requested page in virtual address space not mapped to page in main memory
- On page fault, the page must be fetched from disk
 - Takes millions of clock cycles
 - Handled by OS
- Try to minimize page fault rate

Handling page fault

Page Writes

- Disk writes take millions of cycles
 - Write through is impractical, even with write buffer
 - Millions of processor clock cycles
 - Use write-back
 - Dirty bit in page table is set when page is written
 - Write-back first if dirty bit is on
 - Writing entire page is more time efficient than writing a word
 - CPU switches to another process/program while waiting – context switch

Page fault - replacement rule

- LRU

In class exercise

- Given
 - 4KB page size, 16KB physical memory, LRU replacement
 - Virtual address: byte addressable, 20 bits (how many bytes?)
 - Page table for program A stored in page #0 of physical memory, starting at address 0x0100, assume only 2 valid entries in page table:
 - Virtual page number 0 => physical page number 1
 - Virtual page number 1 => physical page number 2
- Show physical memory including page table
- Complete following table

Virtual Address	Virtual page number	Page fault?	Physical Address
0x00F0C			
0x01F0C			
0x20F0C			
0x00100			
0x00200			
0x30000			
0x01FFF			
0x00200			

In class exercise

Virtual Address	Virtual page number	Page fault?	Physical Address
0x00F0C	0	no	0x1F0C
0x01F0C	1	no	0x2F0C
0x20F0C	0x20	yes	0x3F0C
0x00100	0	no	0x1100
0x00200	0	no	0x1200
0x30000	0x30	yes	0x2000
0x01FFF	1	yes	0x3FFF
0x00200	0	no	0x1200

TLB

- Part of page table + Valid, dirty, reference
- Full Associativity
- One translation per TLB entry

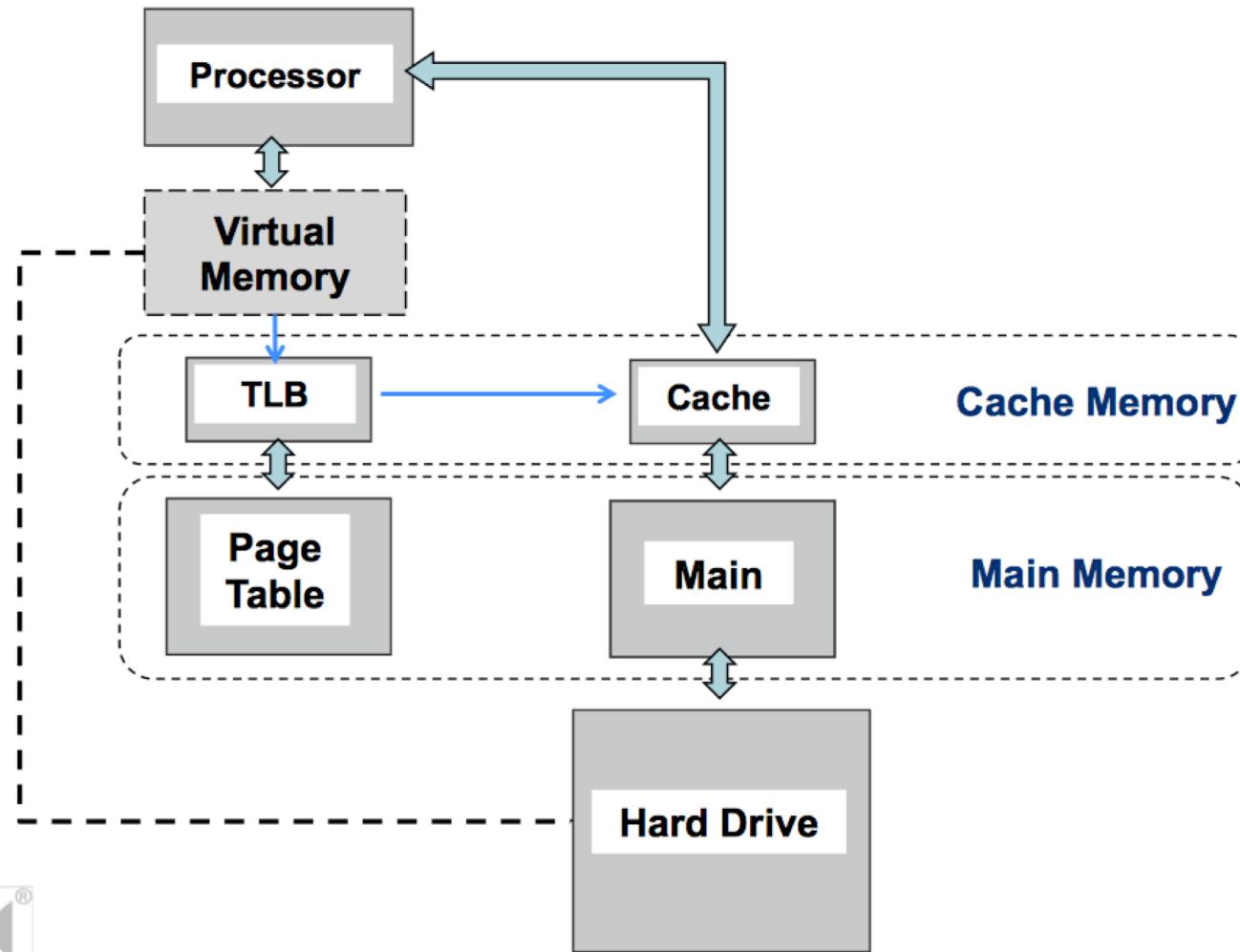
TLB hit or miss

- Virtual address found in TLB – hit
 - Provide physical address
 - Reference bit on
 - Dirty bit on if physical address used for write (might not be needed if dirty bit maintained in physical memory)
- Virtual address not in TLB – TLB miss
 - If page is in memory (page table valid bit = 1)
 - Load the page table entry from memory to TLB and retry
 - Why not just use page table since it's accessed anyway?
 - Could be handled in hardware
 - Can get complex for more complicated page table structures
 - Or in software
 - Raise a special exception, switch to OS with optimized handler
 - If page is not in memory (page fault)
 - Page fault exception – OS handles fetching the page and updating the page table and TLB
 - Then restart the faulting instruction

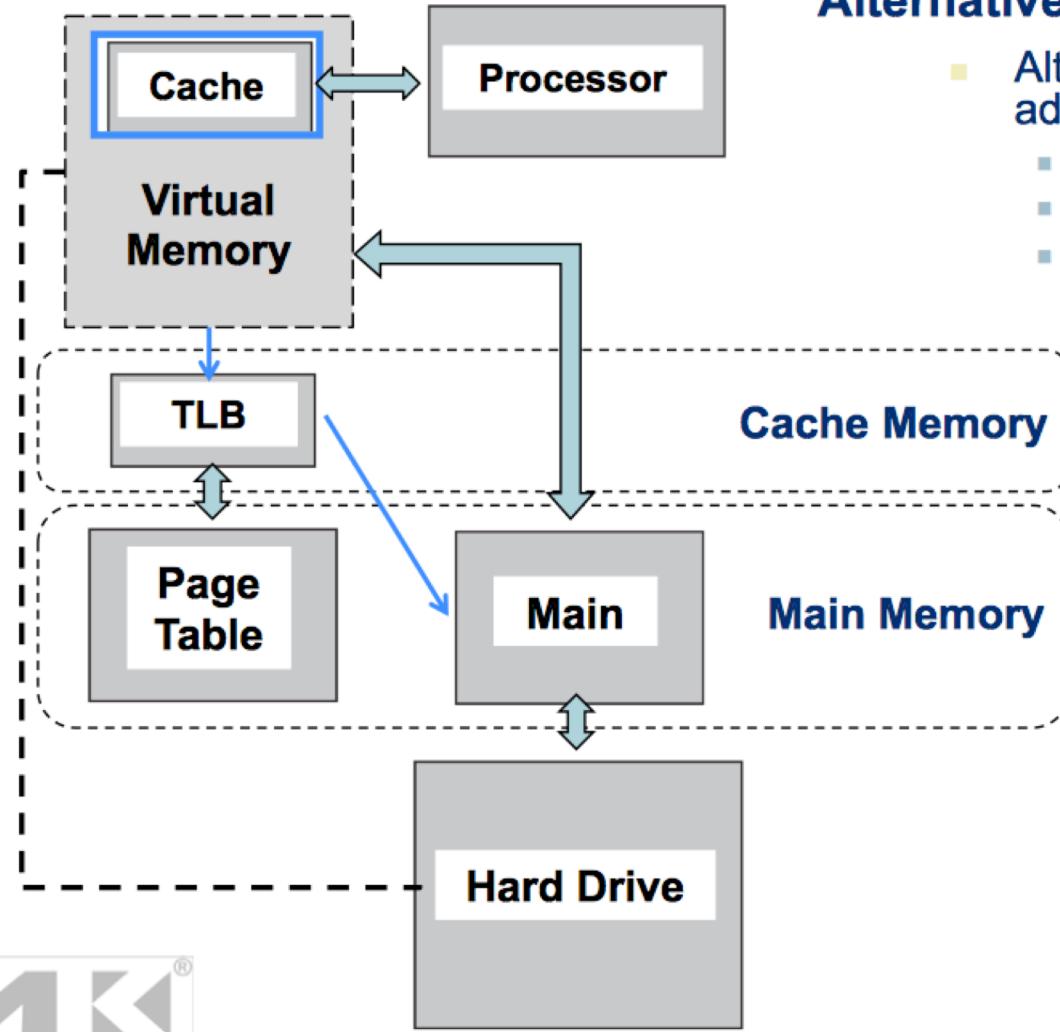
TLB miss vs. Page fault

- Recognize a TLB miss or page fault
 - TLB miss – by cache tag and cache valid bit
 - Page fault – by page table valid bit

TLB, Cache layout



Alternative layout



Alternative Memory Organization

- Alternative: use virtually addressed cache
 - No need for translation
 - Translate when cache miss
 - Complications due to aliasing
 - Different virtual spaces access the same physical location



TLB, Page table and cache relation

TLB	Page table	Cache	Possible? If so, under what circumstance?
Hit	Hit	Miss	Possible, although the page table is never really checked if TLB hits.
Miss	Hit	Hit	TLB misses, but entry found in page table; after retry, data is found in cache.
Miss	Hit	Miss	TLB misses, but entry found in page table; after retry, data misses in cache.
Miss	Miss	Miss	TLB misses and is followed by a page fault; after retry, data must miss in cache.
Hit	Miss	Miss	Impossible: cannot have a translation in TLB if page is not present in memory.
Hit	Miss	Hit	Impossible: cannot have a translation in TLB if page is not present in memory.
Miss	Miss	Hit	Impossible: data cannot be allowed in cache if the page is not in memory.

A hint for your review on memory

- Check L11 page 31-38, whether you can understand all the concepts