Dead Beef?

Explain the significance of "dead beef" 8/16

Job: **Software Engineer**

0xdeadbeef, a value whose hex representation is easily recognized and can be used for, say, initializing memory

http://www.businessinsider.com/15-google-interview-questions-that-will-make-you-feel-stupid-2009-11 1

Victim Cache

A victim cache is a tiny cache on the "back side" of L1 cache that saves the most recent evictions

1. Hit in L1; Nothing else needed
2. Miss in L1 for block at location *b*, hit in victim cache at location *v:* swap contents of *b* and *v* (takes an extra cycle)
3. Miss in L1, miss in victim cache : load missing item from next level and put in L1; put the entry replaced in L1 in victim cache; if victim cache is full, evict one of its entries.

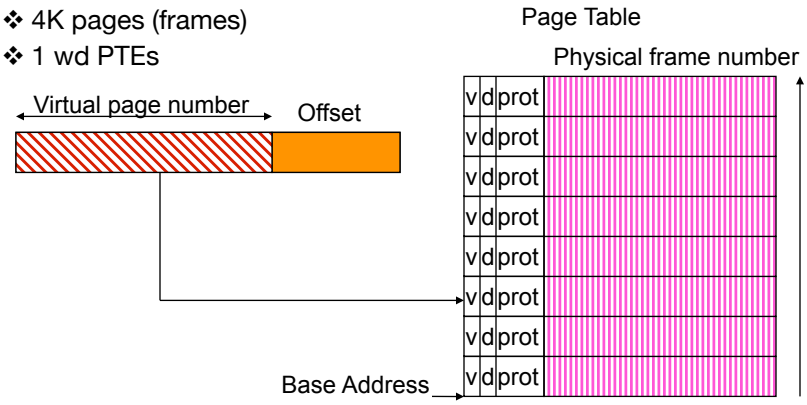
Victim buffer of 4 to 8 entries for a 32KB direct-mapped cache works well.

2

Virtual to Physical Mapping

To illustrate we assume: Memory addresses increase up

- 32 bit virtual addresses
- 4K pages (frames)
- 1 wd PTEs

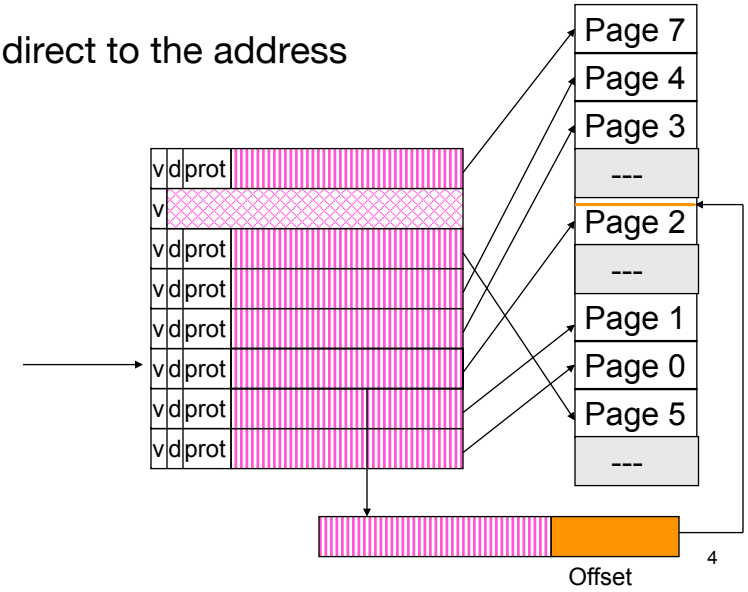


Translation: (VPN<<2)+PTbase = PT Entry Address

3

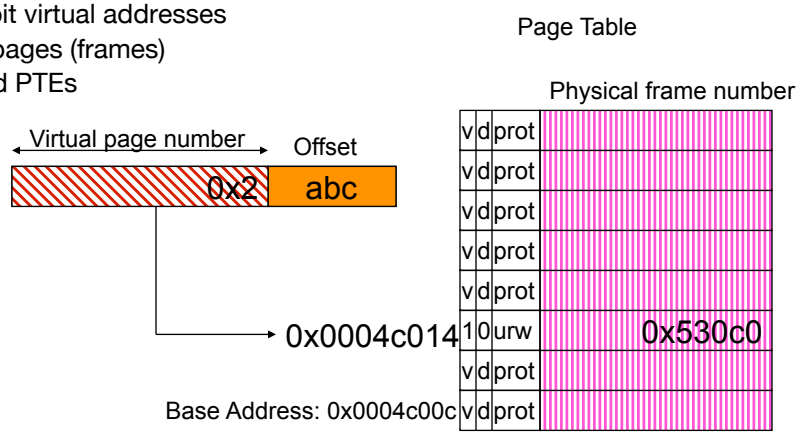
Memory Reference From Page Table

Go indirect to the address



Virtual to Physical Mapping

To illustrate we assume:
32 bit virtual addresses
4K pages (frames)
1 wd PTEs



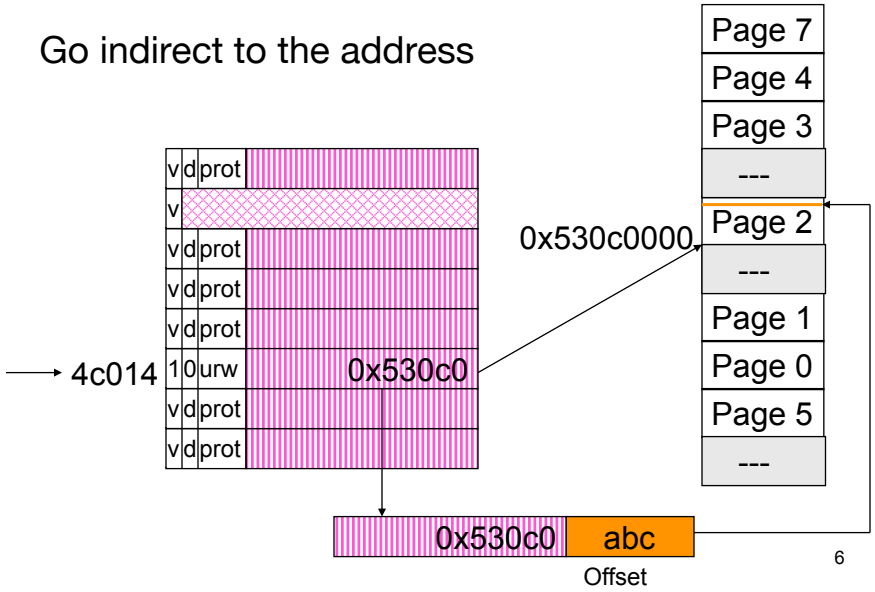
Translation: (VPN<<2)+PTbase = PT Entry Address

Translation: (8)+4c00c = 4c014

5

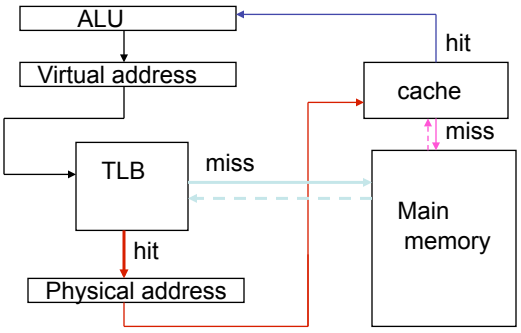
Memory Reference From Page Table

Go indirect to the address



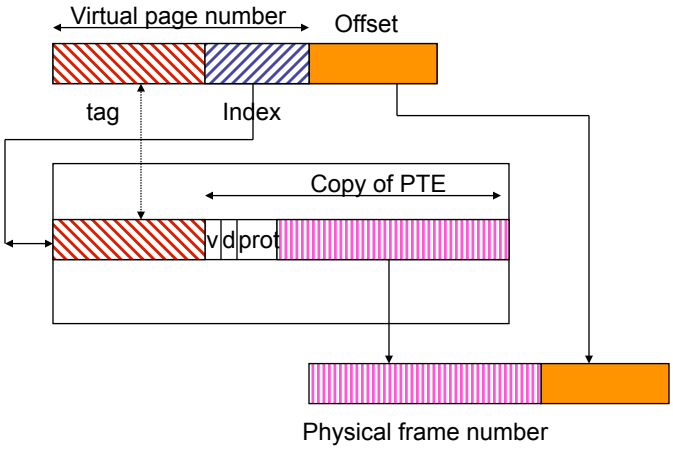
6

Virtual address to memory address



7

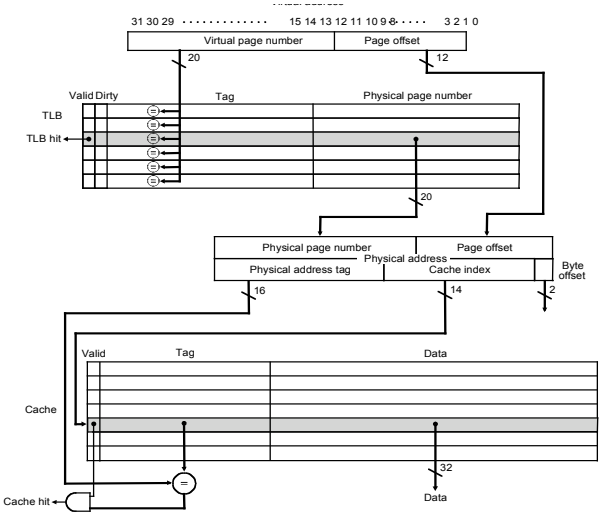
TLB organization



8

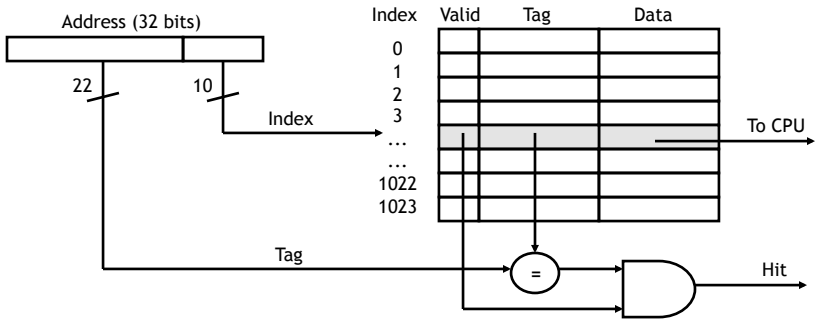
Caching Translations

Virtual to Physical translations are cached in a Translation Lookaside Buffer (TLB)



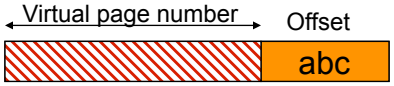
Virtually Mapped Cache

Our original description of the cache was virtually mapped … because address not translated



Which is better?

Virtual to Physical Mapping Practice



0x5
0x3
0xa
0x9

Base Address: 0x0004c000

To illustrate we assume:
32 bit virtual addresses
4K pages (frames)
1 wd PTEs

Page Table

0x0004c028	1	0	ur	0x53d1e
0x0004c024	0			
0x0004c020	1	0	ur	0x12020
0x0004c01c	1	1	urw	0x53d1d
0x0004c018	1	0	ur	0x53d1c
0x0004c014	1	0	urw	0x530c0
0x0004c010	1	0	ur	0x12022
0x0004c00c	1	1	urw	0x12021

…

11

Memory Reference From Page Table

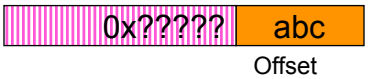
Go indirect to the address

0x0004c028	1	0	ur	0x53d1e
0x0004c024	0			
0x0004c020	1	0	ur	0x12020
0x0004c01c	1	1	urw	0x53d1d
0x0004c018	1	0	ur	0x53d1c
0x0004c014	1	0	urw	0x530c0
0x0004c010	1	0	ur	0x12022
0x0004c00c	1	1	urw	0x12021

0x53d1e000	Page a
0x53d1d000	Page 7
0x53d1c000	Page 6

0x530c0000	Page 5

0x12022000	Page 4
0x12021000	Page 3
0x12020000	Page 8



12

TLB After Exercise

The TLB needed to do what we just did

0x5	1	0	urw	0x530c0
0x3	1	1	urw	0x12021
0xa	1	0	ur	0x53d1e
0x4	1	0	ur	0x12022

13

Handling a TLB Miss

Hardware usually handles a TLB miss. How?

Using the V page number, multiply by 4 (e.g. shift 2)

Add result to page table base address stored in an OS-accessible register to get physical addr of PTE

In parallel, pick TLB entry to evict, writing back to page table, if necessary

- Eviction Policy
- How to decide if write back is needed?
- Finding the PTE to write back into

In parallel check that V page number is < the number of mapped pages, i.e. less than pg tab length

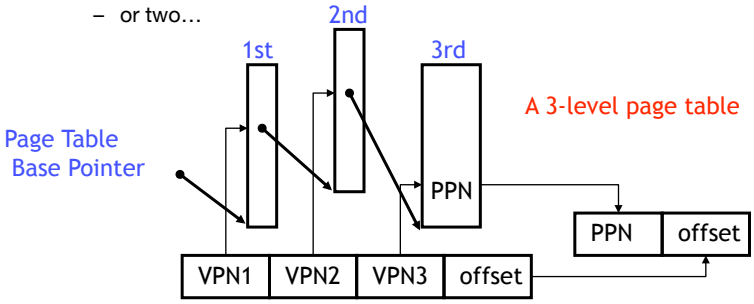
If valid PTE load into empty TLB slot, and restart

14

Dealing with large page tables

Multi-level page tables

- "Any problem in CS can be solved by adding a level of indirection"
  - or two…



Since most processes don't use the whole address space, don't allocate the tables that aren't needed

- Also, the 2nd and 3rd level page tables can be "paged" to disk

15