Translation, Protection, and Virtual Memory

2/25/16

CS 152 Section 6

Colin Schmidt

Agenda

- Protection
- Translation
- Virtual Memory
- Lab 1 Feedback
- Questions/Open-ended discussion
- Hand back Lab 1

Protection

- Why?

			Supervisor			User		
Type	Meaning	Global	R	W	X	R	W	X
0	Pointer to next level of page table.		—					
1	Pointer to next level of page table—global mapping.	•						
2	Supervisor read-only, user read-execute page.		•			•		•
3	Supervisor read-write, user read-write-execute page.		•	•		•	•	•
4	Supervisor and user read-only page.		•			•		
5	Supervisor and user read-write page.		•	•		•	•	
6	Supervisor and user read-execute page.		•		•	•		•
7	Supervisor and user read-write-execute page.		•	•	•	•	•	•
8	Supervisor read-only page.		•					
9	Supervisor read-write page.		•	•				
10	Supervisor read-execute page.		•		•			
11	Supervisor read-write-execute page.		•	•	•			
12	Supervisor read-only page—global mapping.	•	•					
13	Supervisor read-write page—global mapping.	•	•	•				
14	Supervisor read-execute page—global mapping.	•	•		•			
15	Supervisor read-write-execute page—global mapping.	•	•	•	•			

Table 4.2: Encoding of PTE Type field.

Translation

- Why?
  - Multiple users, programs
- Types
  - Base and Bound
  - Paging
  - Segmentation
  - Paged Segements
    - Typical Computer Arch hybrid

Segmentation vs Paging

- words per address
  - 2 vs 1
- programmer visible
  - maybe vs no
- Allocating/replacing block
  - swapping in a new block is hard because size is varied and must be contiguous
- Memory use inefficiency
  - External vs internal fragmentation
- Efficient Disk traffic
  - Not always vs Yes

Page Tables

- Linear vs Hierarchical
  - Linear is simple and inefficient
- How big should a page be?
  - Bigger -> more fragmentation
  - Smaller -> less TLB reach
  - 4KB in RISC-V, why?
    - Legacy
    - Transparent Superpage Support
    - Multi-level TLB hierarchy

What is in a PTE?

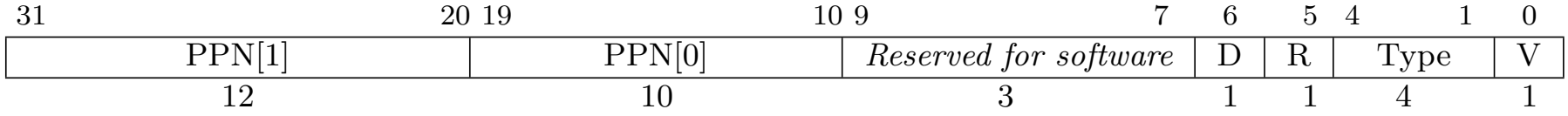


Figure 4.14: Sv32 page table entry.

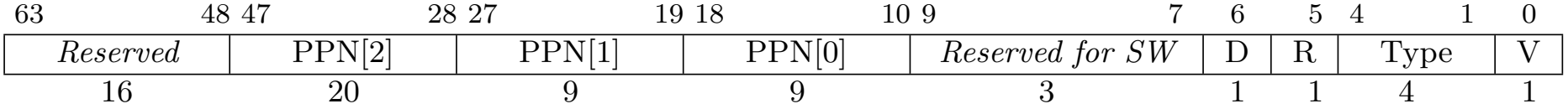


Figure 4.17: Sv39 page table entry.

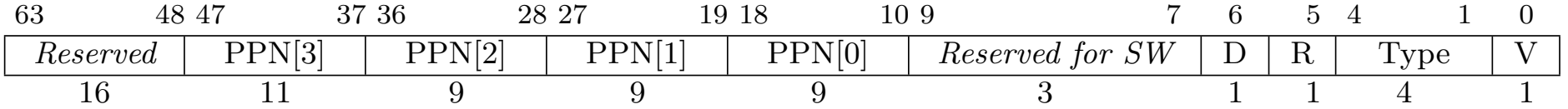


Figure 4.20: Sv48 page table entry.

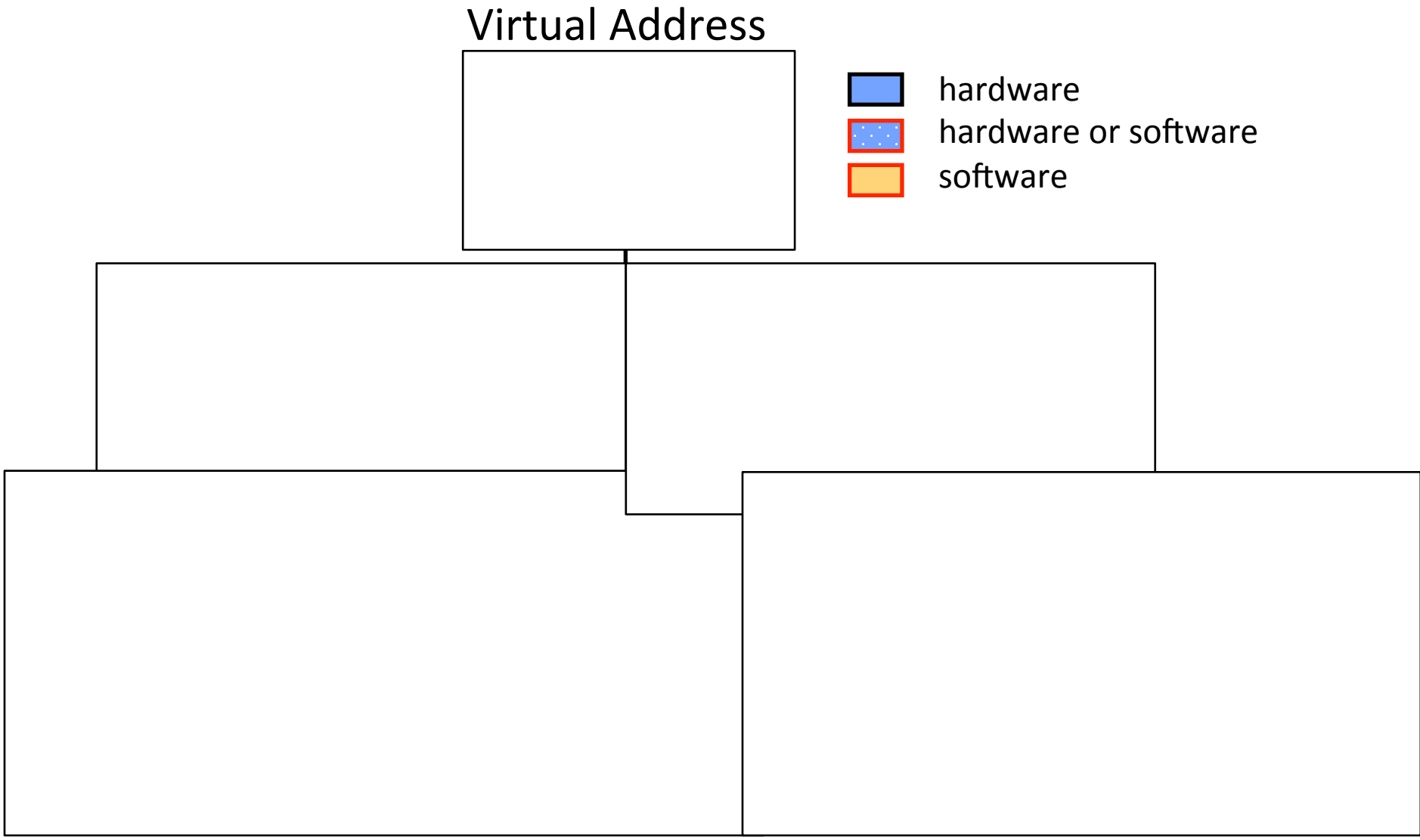
Page Table Walk

- Hardware or Software?
  - Hierarchical page table is pretty well agreed upon so makes sense to put in hardware
  - Can always add disable to PTW in hardware that allows fall back to software

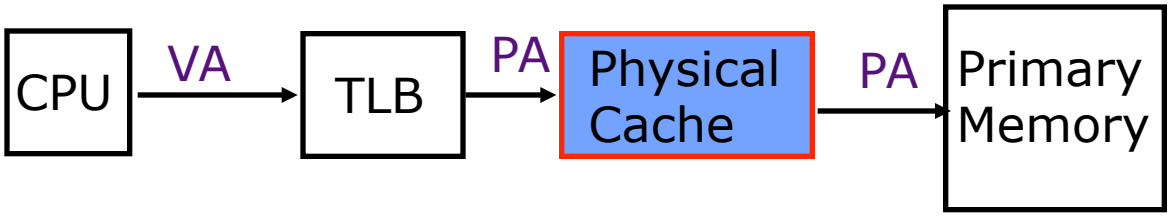
TLB

- Why?
- Separate D vs I TLB?
  - Hazards
- How big should your TLB be?
  - Factor TLB miss, and refill into AMAT
  - Think about Iso-Area AMAT curves
  - Increase assoc vs add entry

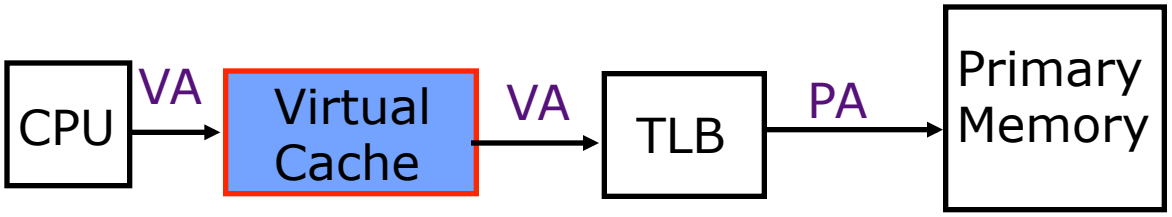
Address Translation:
putting it all together



Virtual-Address Caches



Alternative: place the cache before the TLB



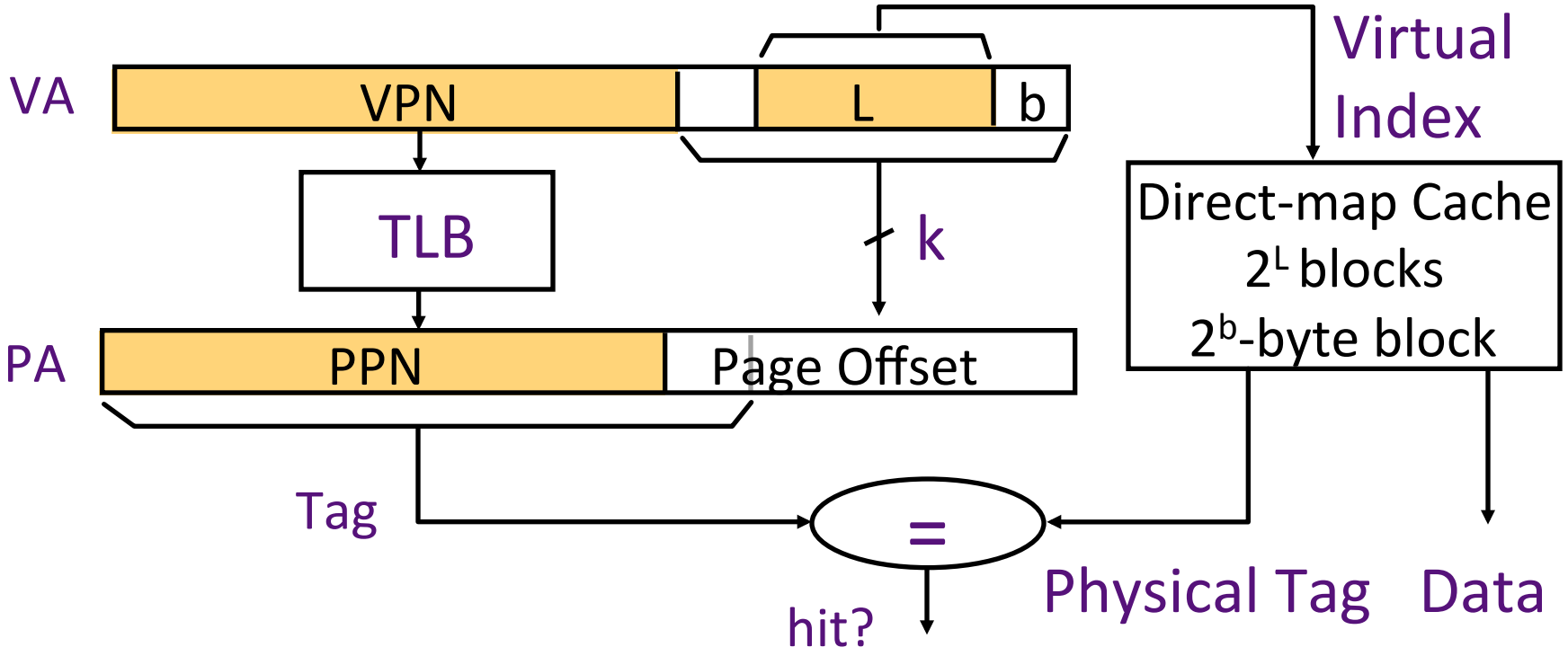
(StrongARM)

- one-step process in case of a hit (+)
- cache needs to be flushed on a context switch unless address space identifiers (ASIDs) included in tags (-)
- *aliasing problems* due to the sharing of pages (-)
- maintaining cache coherence (-) (*see later in course*)

Aliasing

- Physical Cache?
  - How?
- Virtual Cache?
  - How?
- Virtual Tag, Physical Index?
  - How?
- Physical Tag, Virtual Index?
  - How?

Concurrent Access to TLB & Cache (Virtual Index/Physical Tag)



Index L is available without consulting the TLB

=>*cache and TLB accesses can begin simultaneously!*

Tag comparison is made after both accesses are completed

Cases: $L + b = k$, $L + b < k$, $L + b > k$

Virtual Index, Physical Tag

- Lets design an 4-way set associative version

Questions

- Any topics want to discuss as a group?

Lab 1 Feedback

- New Requirements
  - NO DUMPS
    - Any copy paste should almost always be in appendix
  - Must be typed
  - Must be digitally submitted
    - Name the file with your first and last name
  - May impose a page limit, but not yet

Lab 1 Feedback

- Make it interesting
- Non-interesting things
  - Big tables of numbers
  - Big list of calculations (1 example is good)
- Interesting Things
  - Comparative graphs
  - Insightful comments
  - Synthesis of ideas
  - Cool open-ended things

Questions and Hand Back