Part 3.5: The Page Table

segments: large, only a few
pages: small, many
|page| ~ 4KB

Page Table Organization

$\text{VPN} \mapsto \text{PTE}$

↳ page table entry

Linear Page Table: array

- OS indexes the array by VPN, returns PTE
- PTE contains: - PFN → phys frame number
  - control bits

⇒ stored in memory

Common Control Bits

- Valid / Invalid: if translation is correct

  $\rightarrow$ =1 when mem has been allocated

- Protection:

  $\rightarrow$ R/W/X

  ↳ execute

swapping

- Present Bit:

  =1 if the page is in RAM

- Dirty Bit:

  =1 if the page has been written to

- Reference Bit:

  =1 if the page has been acsessed/used

Example Page Table Entry (PTE)

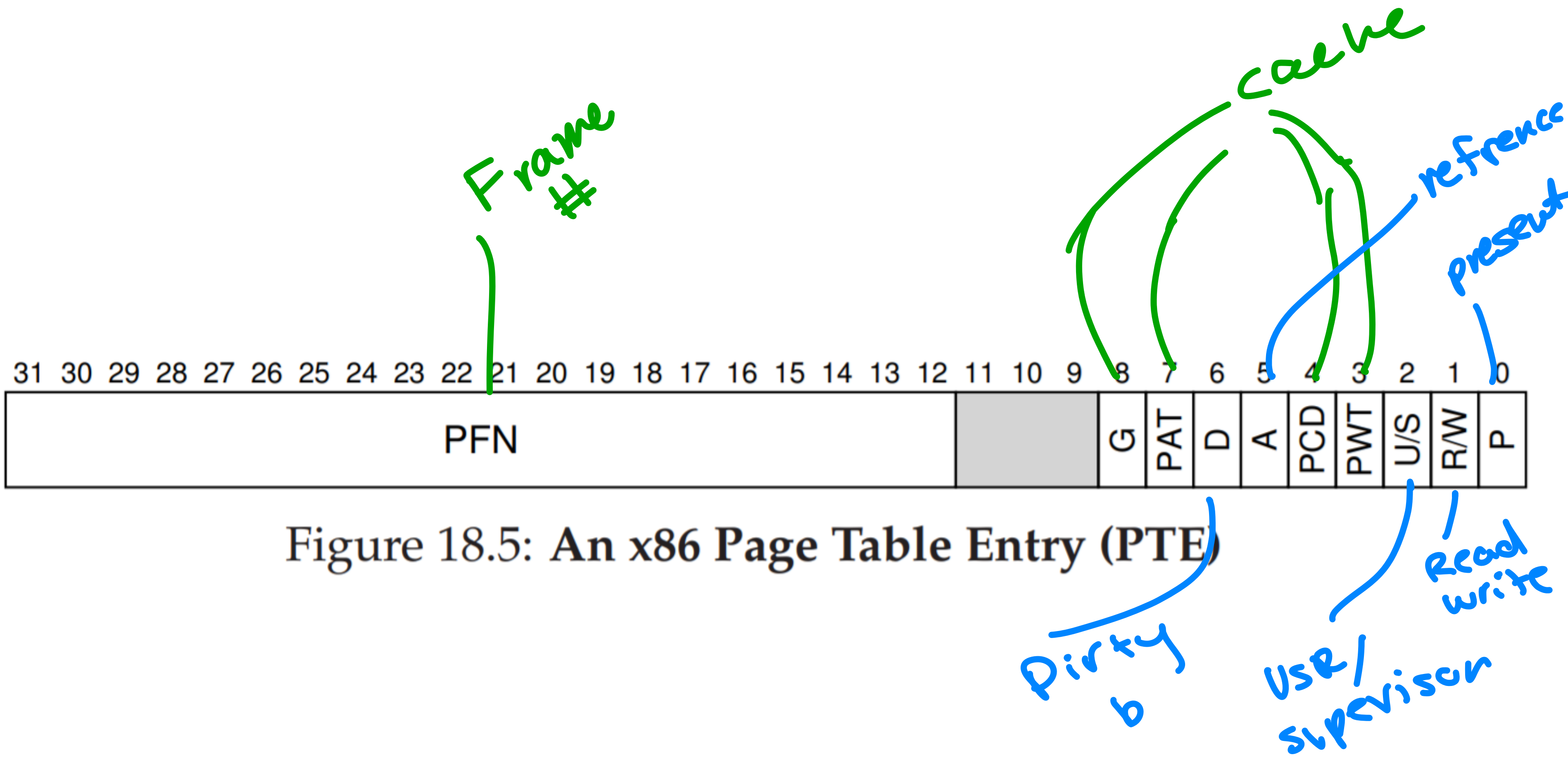


Figure 18.5: **An x86 Page Table Entry (PTE)**

Page Table Issue (1) – Speed

A *single* memory access (e.g., `lw`) now requires:

1. Extract the VPN from the address
2. form address of PTE PgTbl[VPN]
3. Fetch PTE
4. verify the valid and protection bits
5. Form the physical address
6. Fetch memory at physical address

*require accessing RAM

Solving Speed - 1

Our problem:

1. CPU is fast
2. Memory is slow

} cache!

add a cache for address translation

Translation Lookaside Buffer

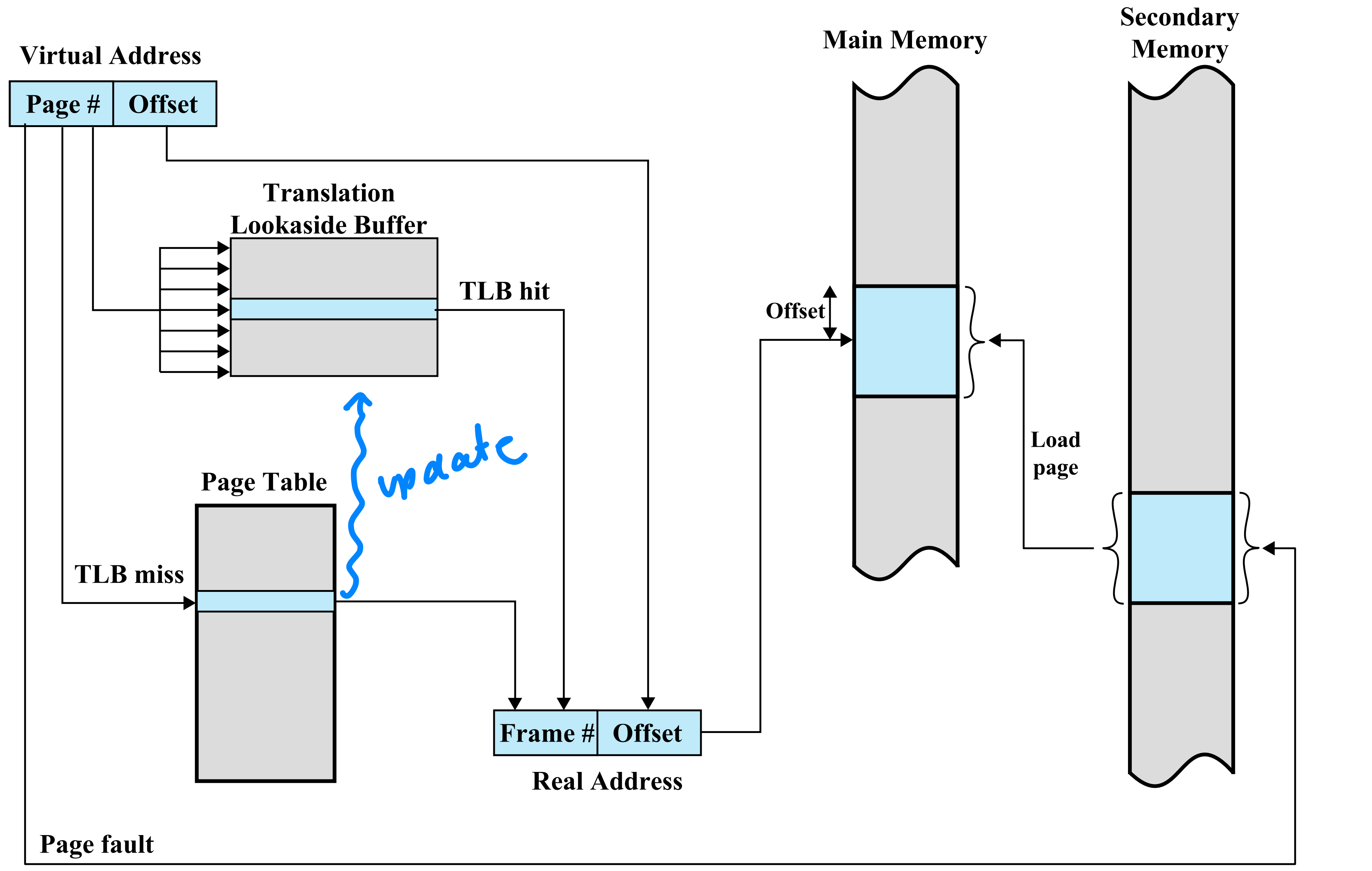
TLB

Solving Speed – 2

Memory access steps with a TLB:

1. Extract the VPN
2. Check if the TLB holds the PTE
   a) TLB - hit: extract PTE from TLB
   b) TLB - Miss:
      i. form PTE address
      ii. fetch PTE
      iii. check valid and protection bits
      iv. update TLB
3. Fetch memory at physical address

Visual



Virtual Address
Page #
Offset
Translation Lookaside Buffer
TLB hit
update
Page Table
TLB miss
Frame #
Offset
Real Address
Main Memory
Offset
Load page
Secondary Memory
Page fault

Example

```
int sum = 0;
for (int i = 0; i < 10; i++) {
        sum += a[i];
}
```

≈ 70% hit ratio

a[0] → miss

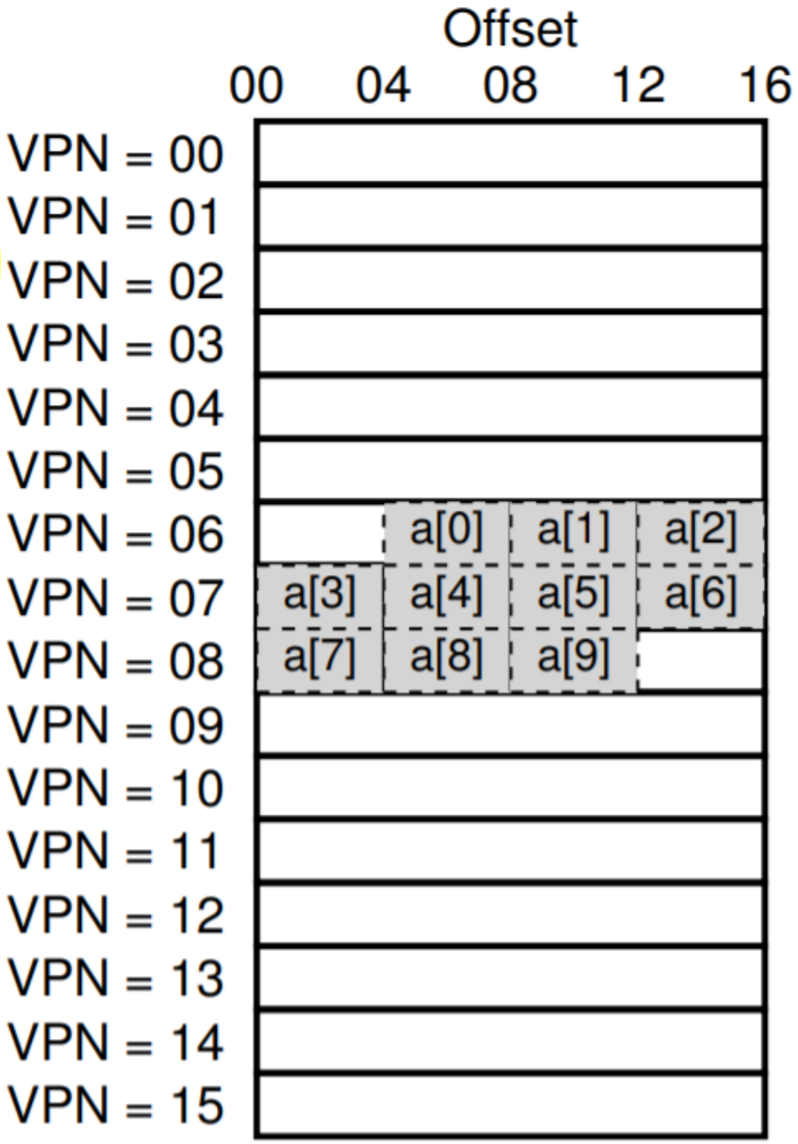
$\left.\begin{array}{l}\text{a[1]}\\ \text{a[2]}\end{array}\right\}$ → hits

a[3] → miss

a[4, 5, 6] → hits

a[8] → miss

a[8, a] → hits



Solving Speed - 3

TLBs (typically):

- 32 - 128 entries
- fully associate
- contains
  - entire PFE
  - mere control Bits

Solving Speed - 4

Reminder: Page-tables are *per-process*

VPN	PFN	valid	prot
10	100	1	rwx
—	—	0	—
10	170	1	rwx
—	—	0	—

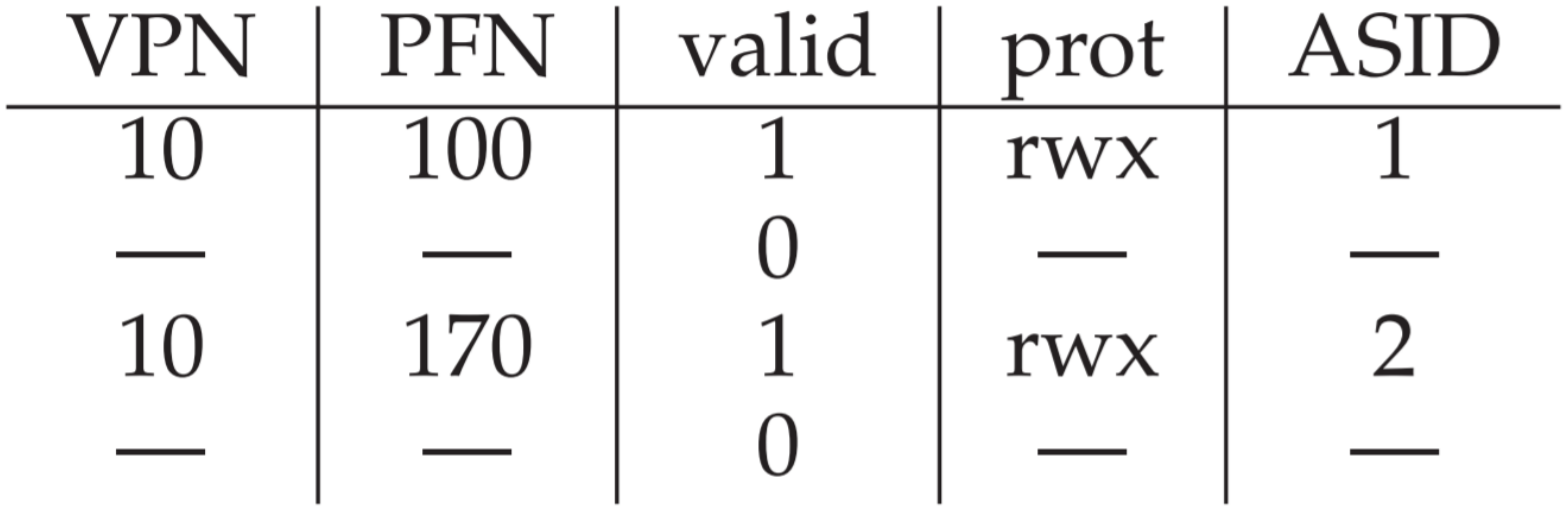
What happens to the TLB on a context switc

1) empty/flush the TLB on a context switch

2) add an address space identifier (ASID)

Solving Speed - 5

ASID: Address space identifier



VPN	PFN	valid	prot	ASID
10	100	1	rwx	1
—	—	0	—	—
10	170	1	rwx	2
—	—	0	—	—

Solving Speed: Summary

Add a TLB

Learning Group Time

Lots of time for learning groups today!