CS230 Tut 11 Solutions

(1) a is rarest, b is next rare

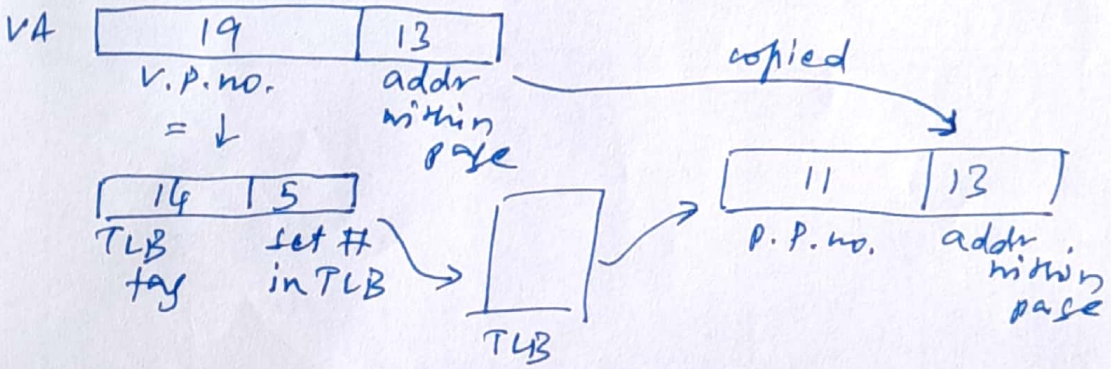
c vs d depends on program, but generally TLB miss rarer than cache miss

$c > d > b > a$

(2) VA = 32 bits PA = 24 bits Page = 8 KB = 13 bits

TLB = 256 entries = 2^8 entries = 2^5 sets of 8 entries each

a)



b) Fully assoc. TLB ⇒ all 19 bits of V.P.no used as tag

Direct mapped ⇒ 19 = 11 bit tag + 8 bit TLB entry number

(3) See video / slides

(4) It will produce correct behaviour at the cost of lower efficiency

(5) Use PID field in TLB - compared with PID register along with TLB tag comparison

(6)

TLB	L1	L2	Page	
hit	hit	hit	hit	- best case
hit	hit	hit	miss	- not possible
hit	hit	miss	hit	- "
hit	hit	miss	miss	- "
hit	miss	hit	hit	- only L1 miss
hit	miss	hit	miss	- not possible
hit	miss	miss	hit	- L1 & L2 miss
hit	miss	miss	miss	- possible if TLB entries can have invalid P.T. entries

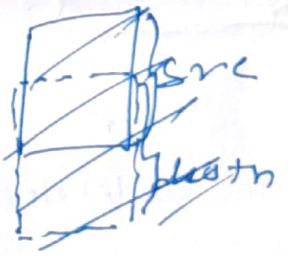
All "not possible" cases - if miss in higher level, can't have hit in lower level of cache hierarchy

(continued)

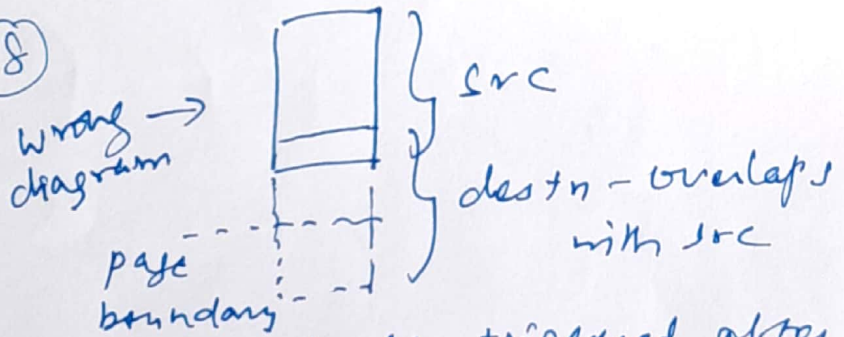
(continued)

TLB	L1	L2	Page	
miss	hit	hit	hit	- only TLB miss
miss	hit	hit	miss	- not possible
miss	hit	miss	hit	- not possible
miss	hit	miss	miss	- not possible
miss	miss	hit	hit	- TLB & L1 miss
miss	miss	hit	miss	- not possible
miss	miss	miss	hit	- TLB, L1, L2 miss
miss	miss	miss	miss	- worst case

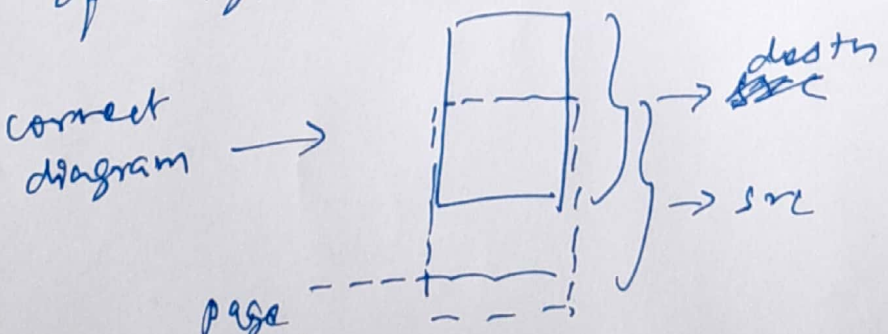
(7) Consider lw instruction
TLB entry for instruction & TLB entry for data keep kicking each other out - never ending cycle of TLB miss exceptions



(8)



- page fault triggered after first few bytes have been copied
- cannot restart instruction as some part of original src already overwritten



- say, last byte triggers page fault in src