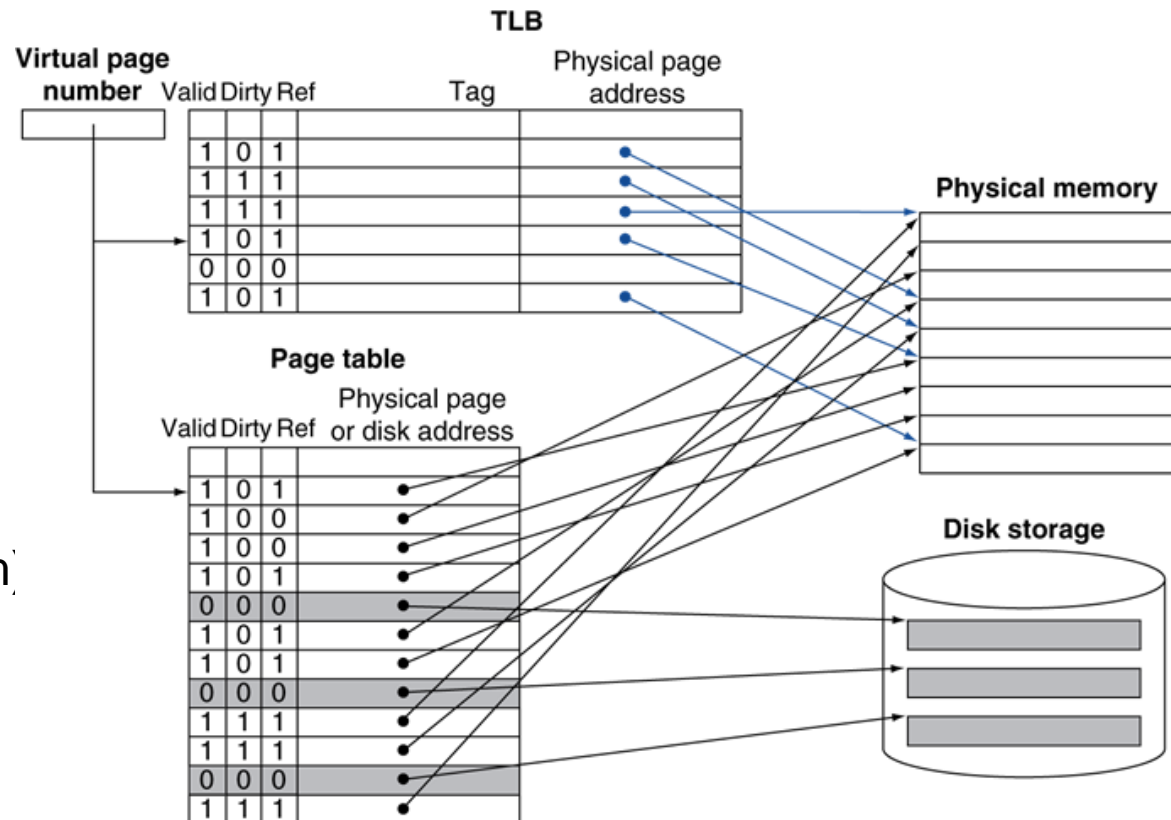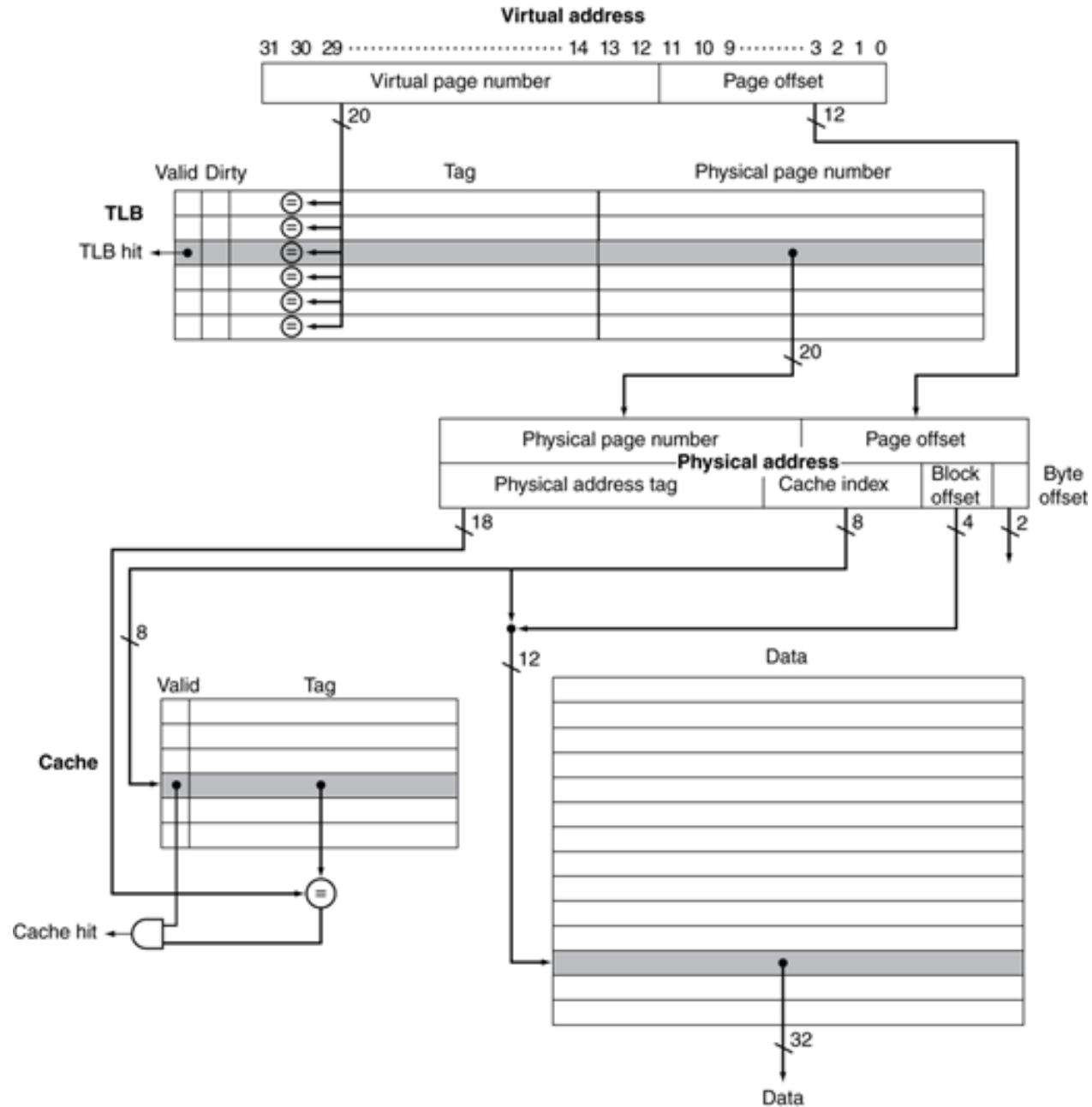# Optimizing address translation (TLB)

- Avoid two memory accesses for each read or write
- Rely on locality of reference to the page table
- Translation Lookaside Buffer (TLB) - cache holding page table mappings
- TLB tag: higher bites of the virtual address
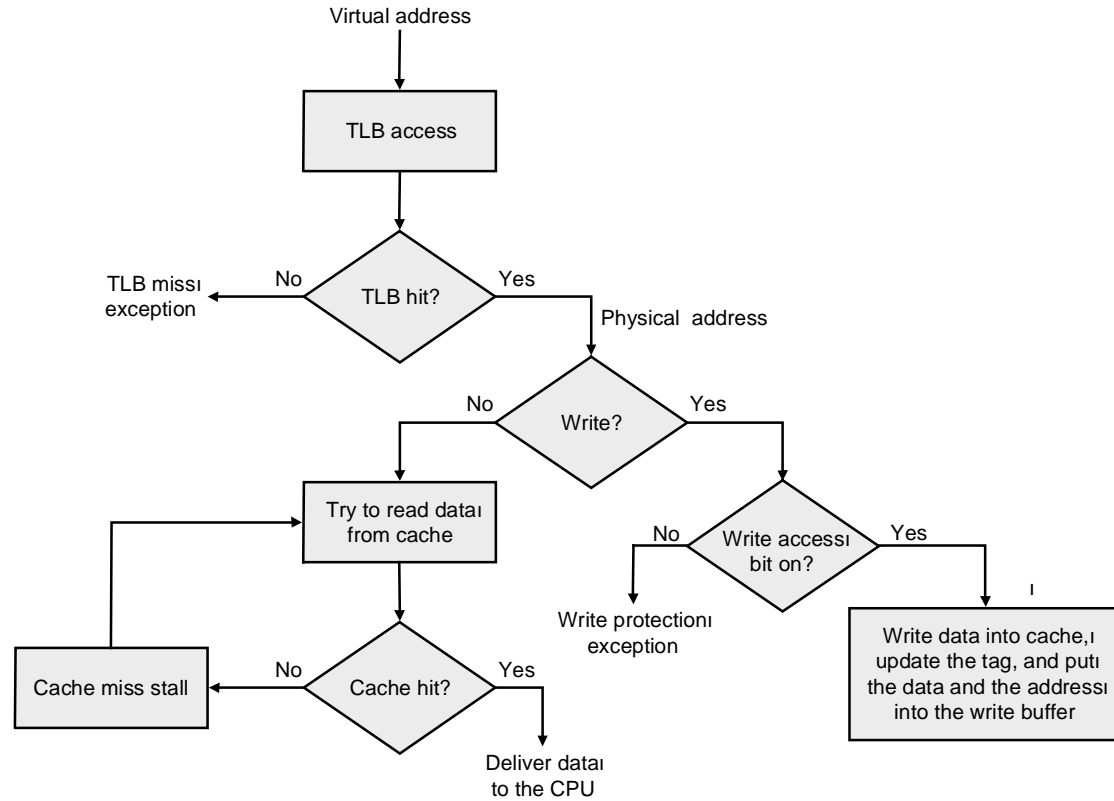- Valid bit, reference bit, dirty bit

- TLB size: 32-4096 entries
- Block size: 1-2 page table entries
- Hit time: 0.5-1 clock cycle
- Miss penalsy: 10-30 clock cycles
- Miss rate: 0.01% - 1%
- Associativity: various (e.g. random)

TLB and cache organization

Virtual address

31 30 29 ·································· 14 13 12 11 10 9 ········· 3 2 1 0

Virtual page number	Page offset

20

12

TLB

TLB hit ←

Valid	Dirty		Tag	Physical page number

20

Physical page number	Page offset

Physical address

Physical address tag	Cache index	Block offset	Byte offset

18

8

8

12

Cache

Valid	Tag

Data

=

Cache hit ←

=

32

Data

DECStation 3100: processing read or write

Virtual address

TLB access

TLB hit?

No → TLB missı exception

Yes → Physical address

Write?

No → Try to read dataı from cache

Yes → Write accessı bit on?

No → Write protectionı exception

Yes → Write data into cache,ı update the tag, and putı the data and the addressı into the write buffer

Try to read dataı from cache → Cache hit?

No → Cache miss stall

Yes → Deliver dataı to the CPU

Overall operation of a memory hierarchy

Cache	TLB	Page table	Possible? If so under what circumstance
miss	hit	hit	Possible, but never detected (page table is not checked after a TLB hit).
hit	miss	hit	TLB misses, but entry found in page table. After retry, found in cache.
miss	miss	hit	TLB misses, but entry found in page table. After retry, miss in cahce.
miss	miss	miss	TLB misses and is followed by page fault. After retry , miss in cache.
miss	hit	miss	Impossible: cannot have translation in TLB if page is not in memory.
hit	hit	miss	Impossible: cannot have translation in TLB if page is not in memory.
hit	miss	miss	Impossible: data cannot be in cache if page is not present in memory.

Handling TLB misses and page faults

TLB miss (handled in software or hardware):
• The page is in memory, but its physical address is missing. A new TLB entry must be created
• The page is not in memory and the control is transferred to the operating system to deal with a page fault

Page fault (handled by the operating system):
• Causing exception (interrupt): using EPC and Cause register
• Instruction page fault:
  • Store the state of the process
  • Look up the page table to find the disk address of the referenced page
  • Choose a physical page to replace
  • Start a read from disk for the referenced page
  • Execute another process until the read completes
  • Restart the instruction which caused the fault
• Data access page fault:
  • Occurs in the middle of an instruction
  • MIPS instructions are restartable: prevent the instruction from completing and restart it from the beginning
  • More complex machines: interrupting instructions (saving the state of CPU)