Announcements

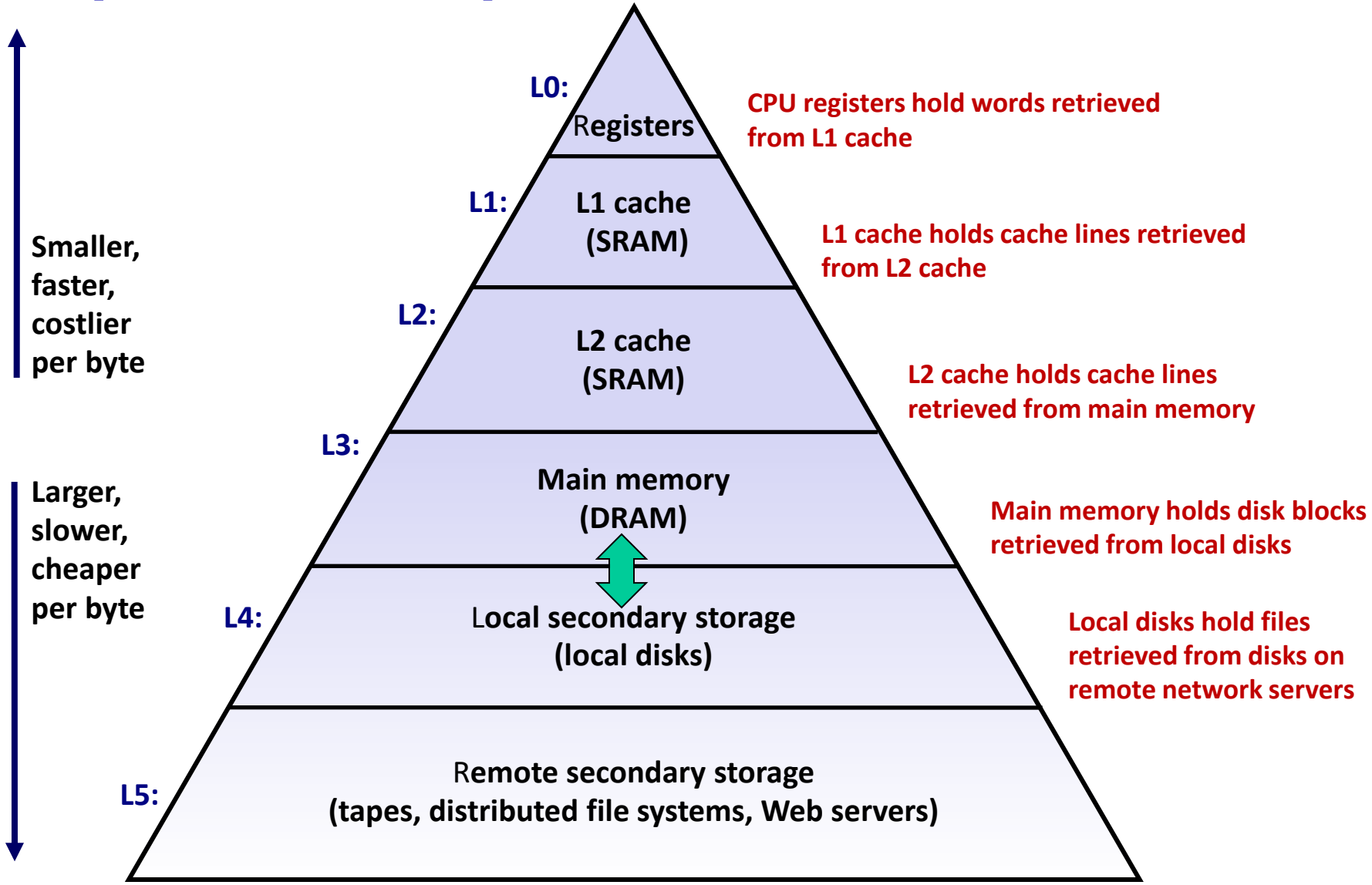
- Assignment 8
  - Posted Thursday, November 20; due December 2
- Quiz 11
  - Monday, December 1
  - Covers Lectures 29-31
- Graded Lab 3
  - December 3-4
  - Covers Assignments 4-7

Lecture 31

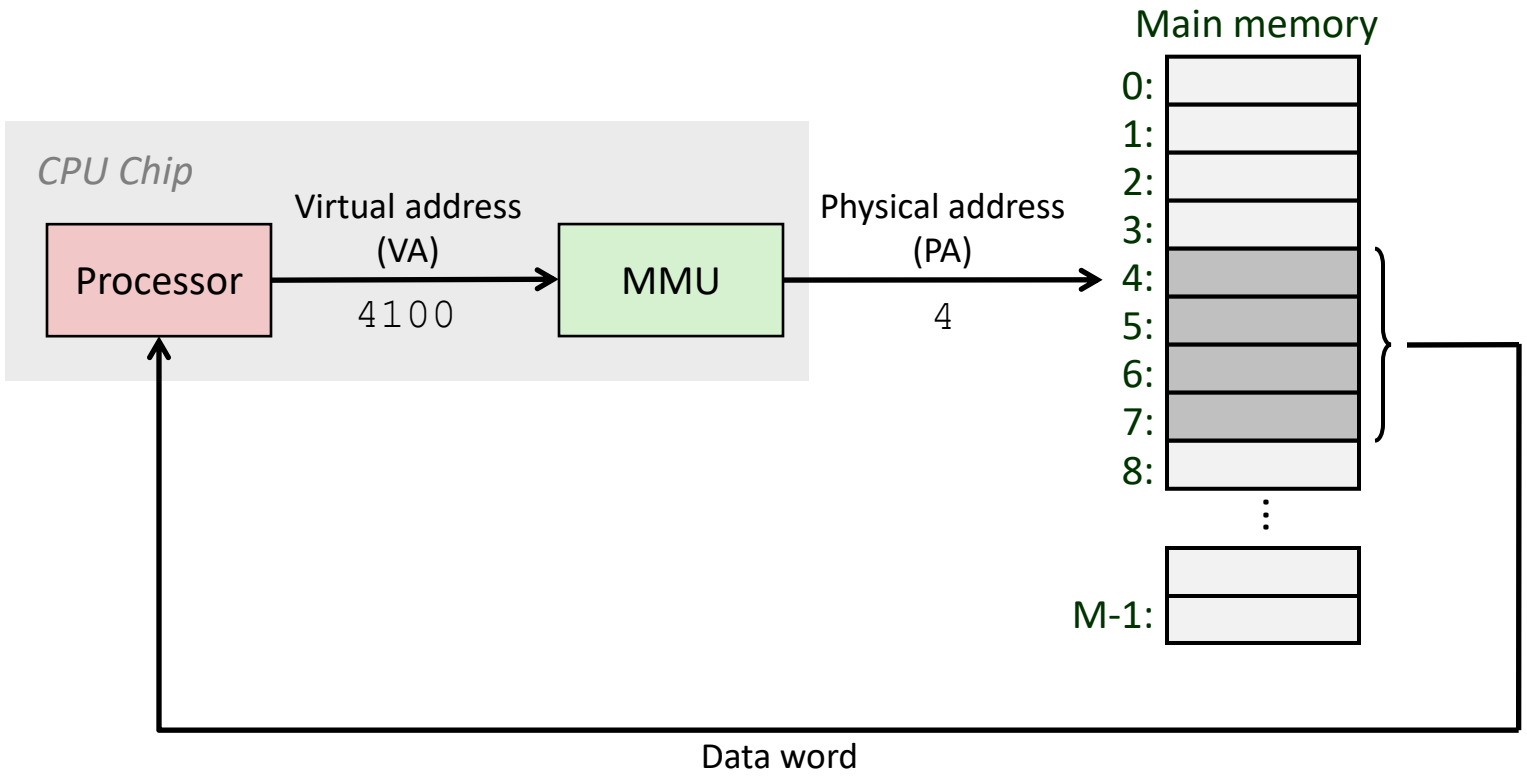
Address Translation

CPSC 275
Introduction to Computer Systems

Memory Hierarchy



A System Using Virtual Addressing



- Used in virtually all modern computer systems.

Address Spaces

- Virtual address space: set of $N = 2^n$ virtual addresses

  $\{0, 1, 2, 3, \ldots, N\text{-}1\}$

  - Compiler generates *relocatable* virtual addresses

- Physical address space: set of $M = 2^m$ physical addresses

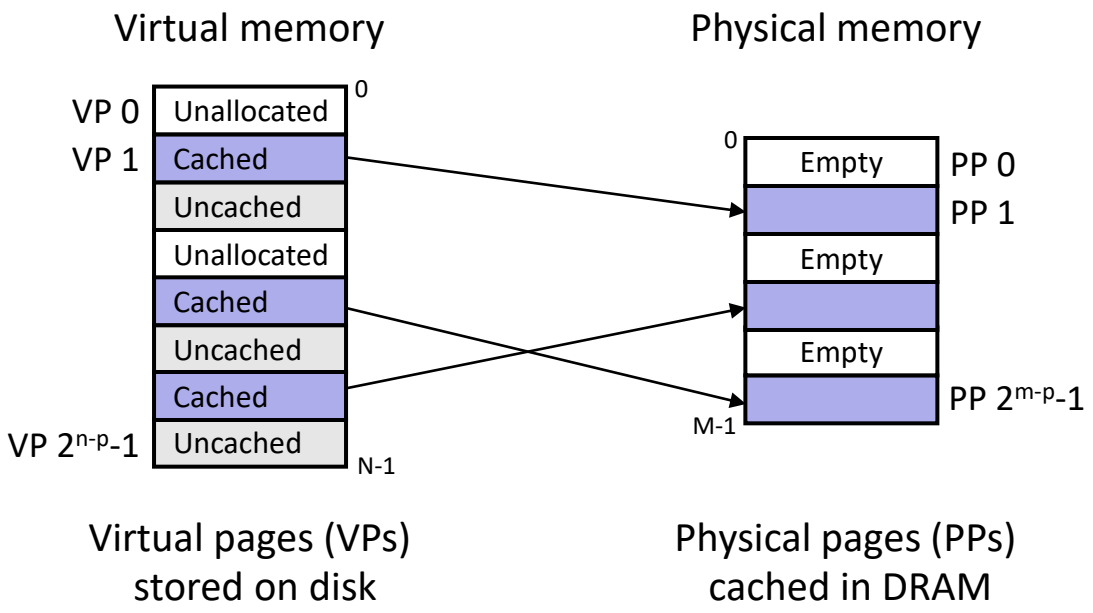
  $\{0, 1, 2, 3, \ldots, M\text{-}1\}$

  - These are actual addresses in DRAM.

- Every byte in main memory:
  - one physical address
  - one (or more) virtual addresses

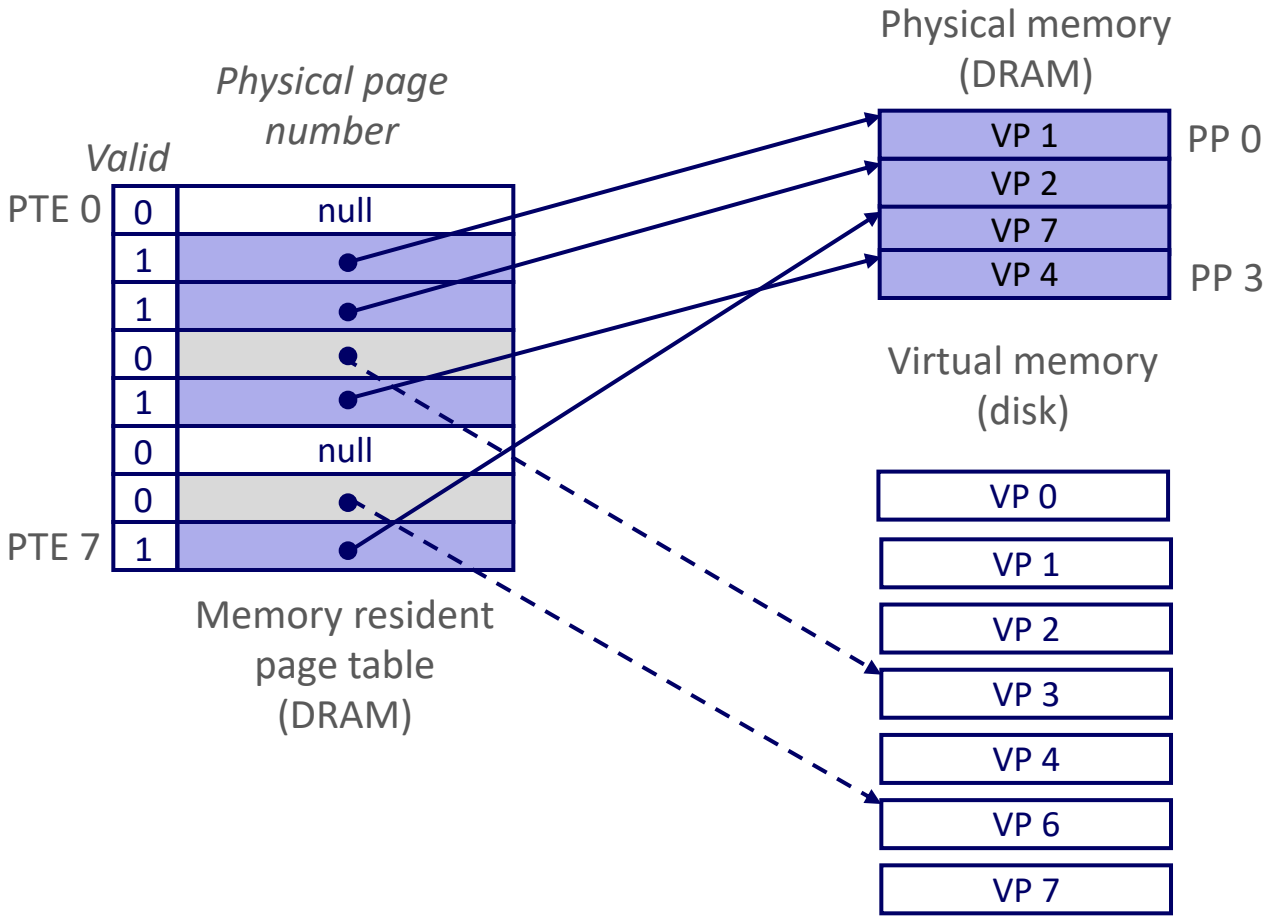
VM as a Tool for Caching

- *Virtual memory* is an array of *N* contiguous bytes stored on disk.
- Some contents on disk are cached in *physical memory* (*DRAM cache*)
  - These cache blocks are called *pages* (size is $P = 2^p$ bytes)



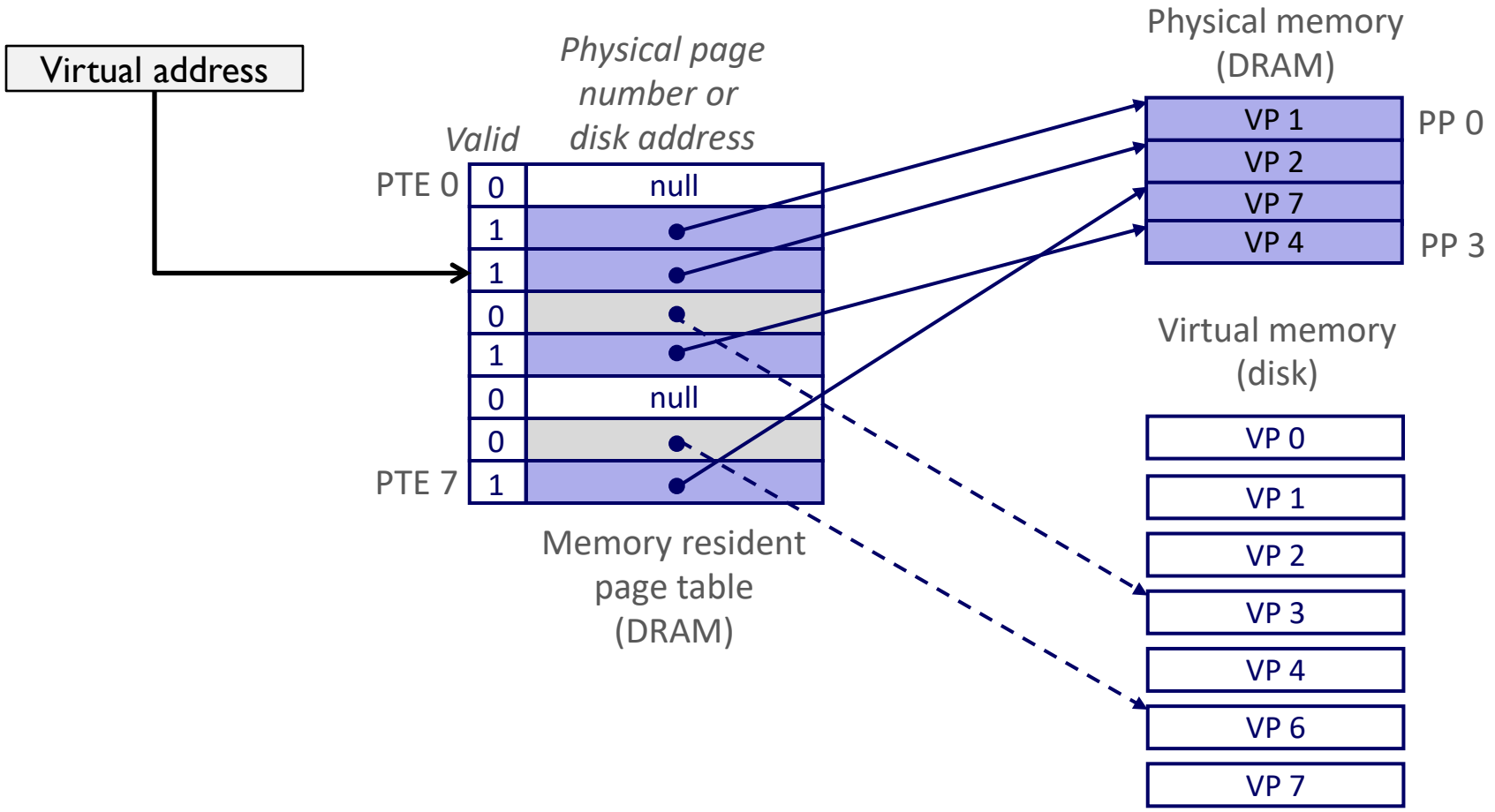
Page Tables

- A *page table* is an array of page table entries (PTEs) that maps virtual pages to physical pages – per-process data structure in DRAM



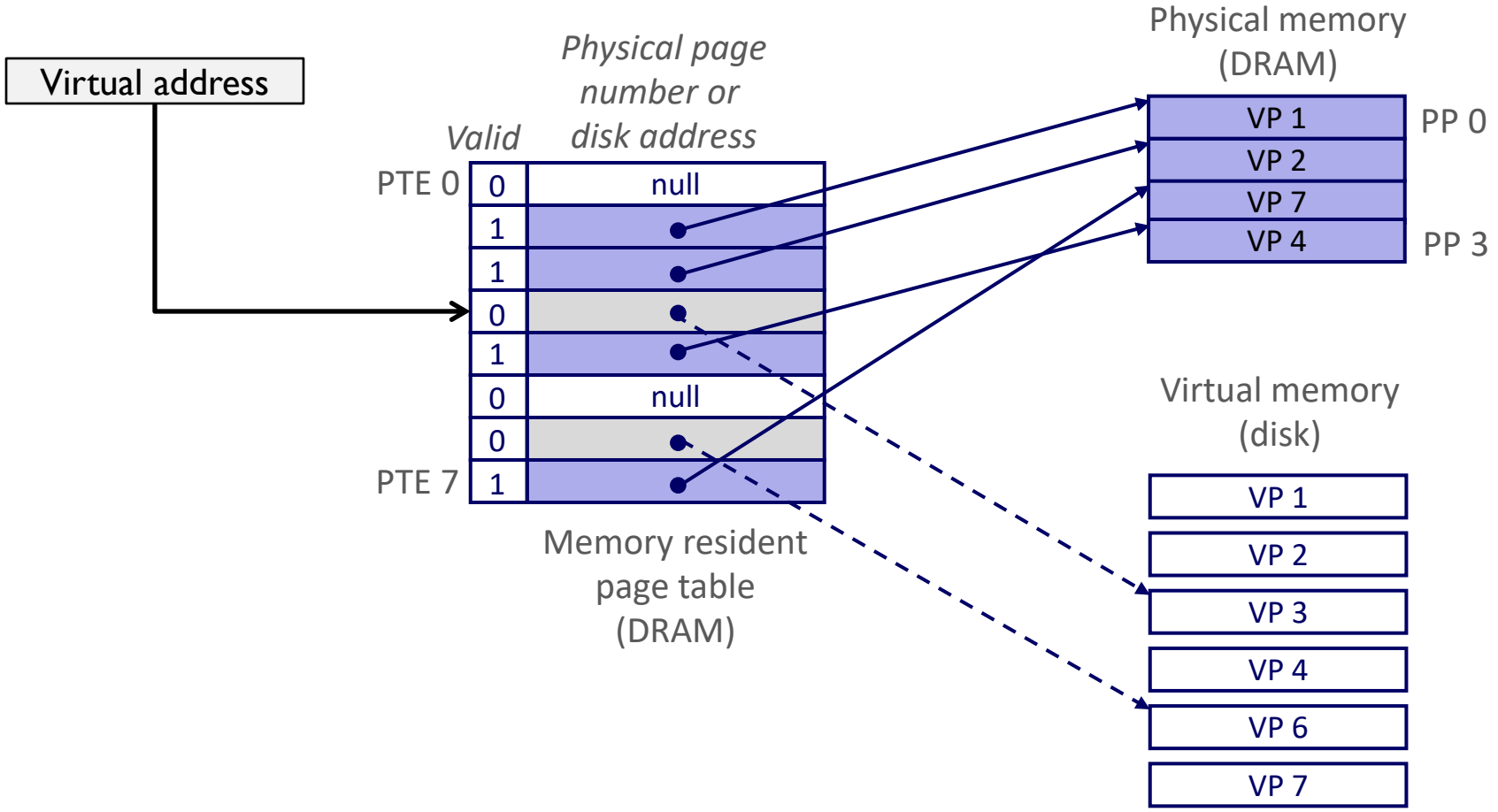
Page Hit

- *Page hit:* reference to VM word that is in physical memory



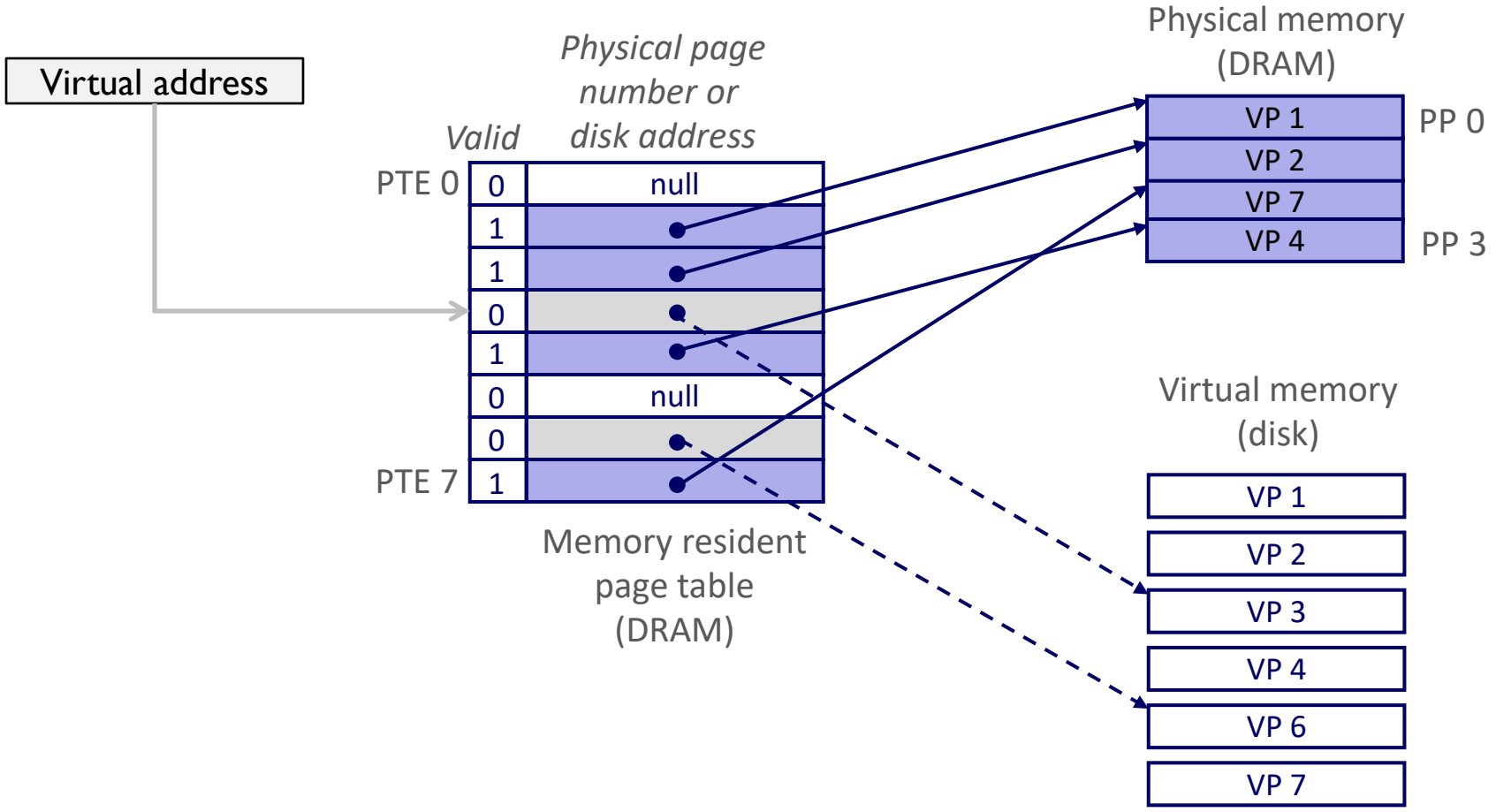
Page Fault

- *Page fault:* reference to VM word that is not in physical memory



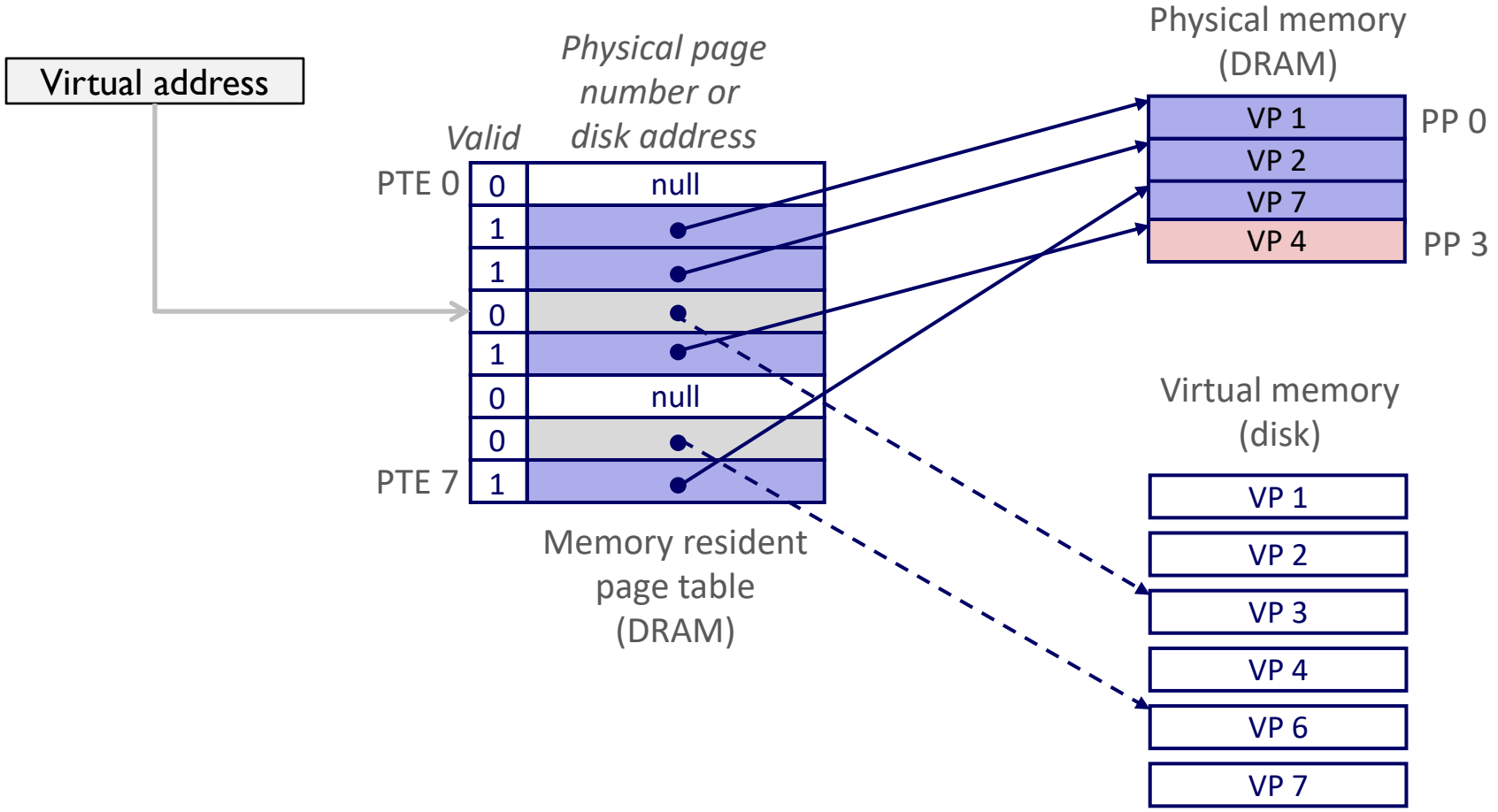
Handling Page Fault

- Page miss causes page fault



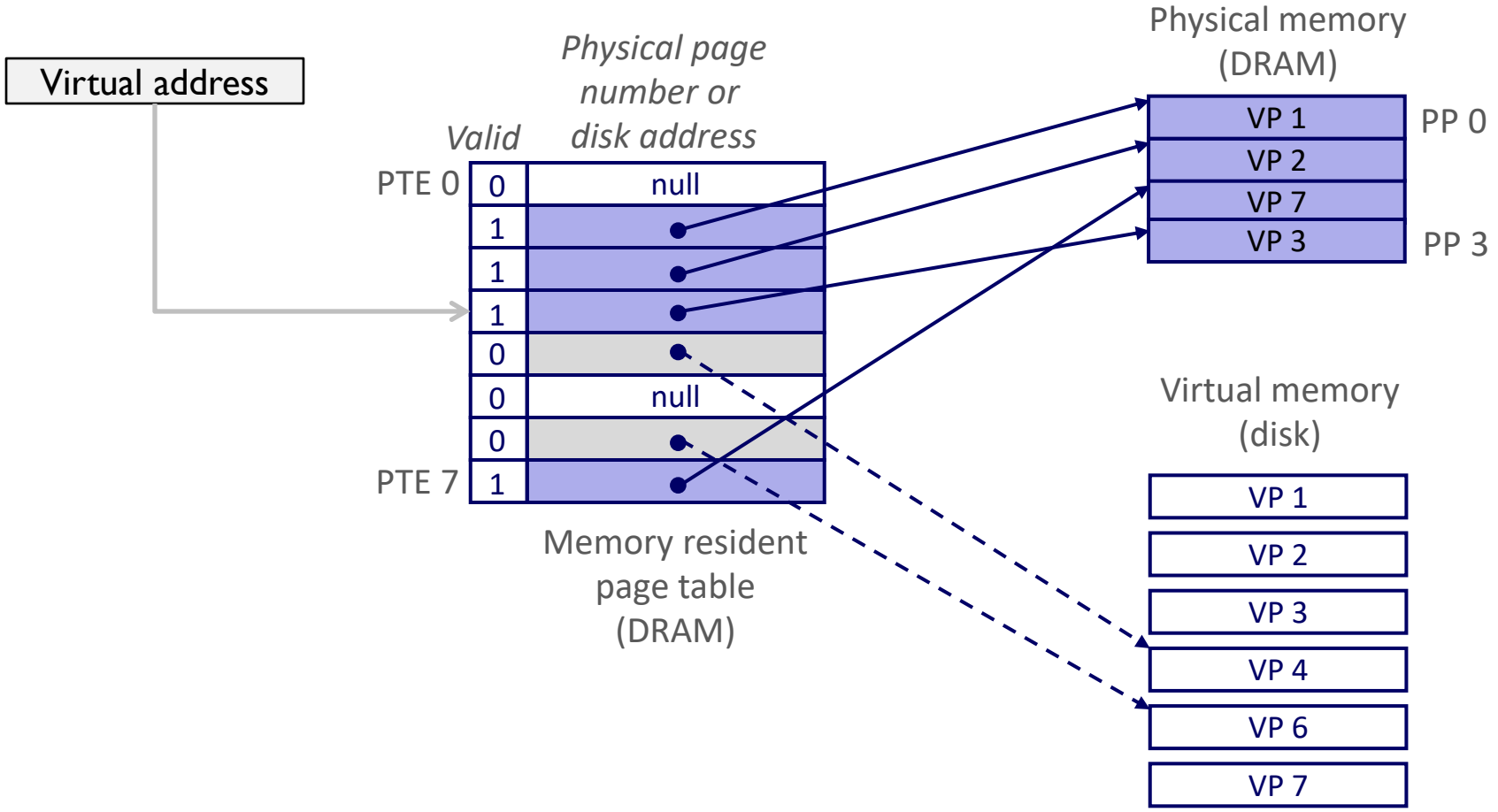
Handling Page Fault

- Page miss causes page fault
- Page fault handler selects a victim to be evicted (here VP 4)



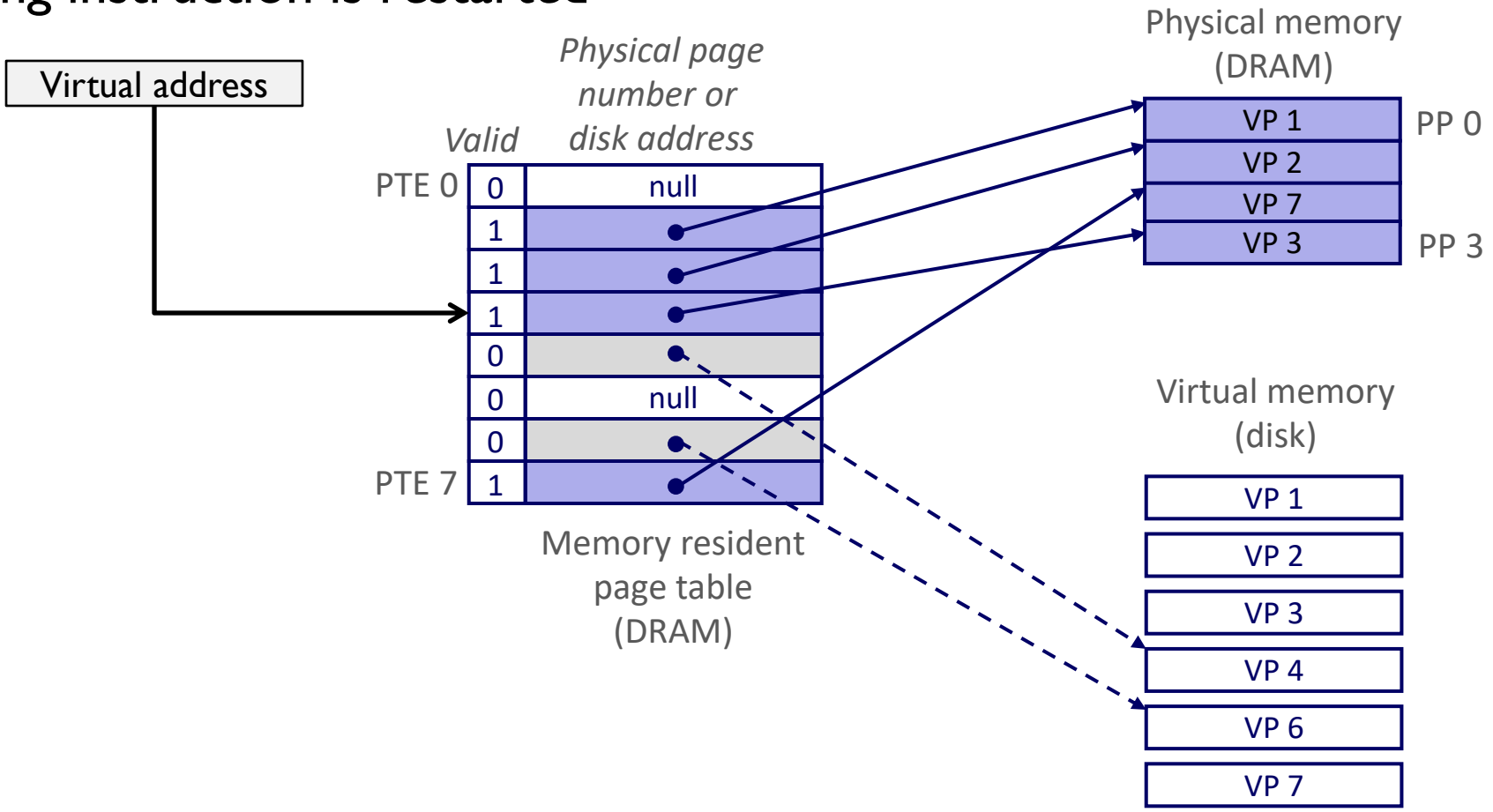
Handling Page Fault

- Page miss causes page fault
- Page fault handler selects a victim to be evicted (here VP 4)



Handling Page Fault

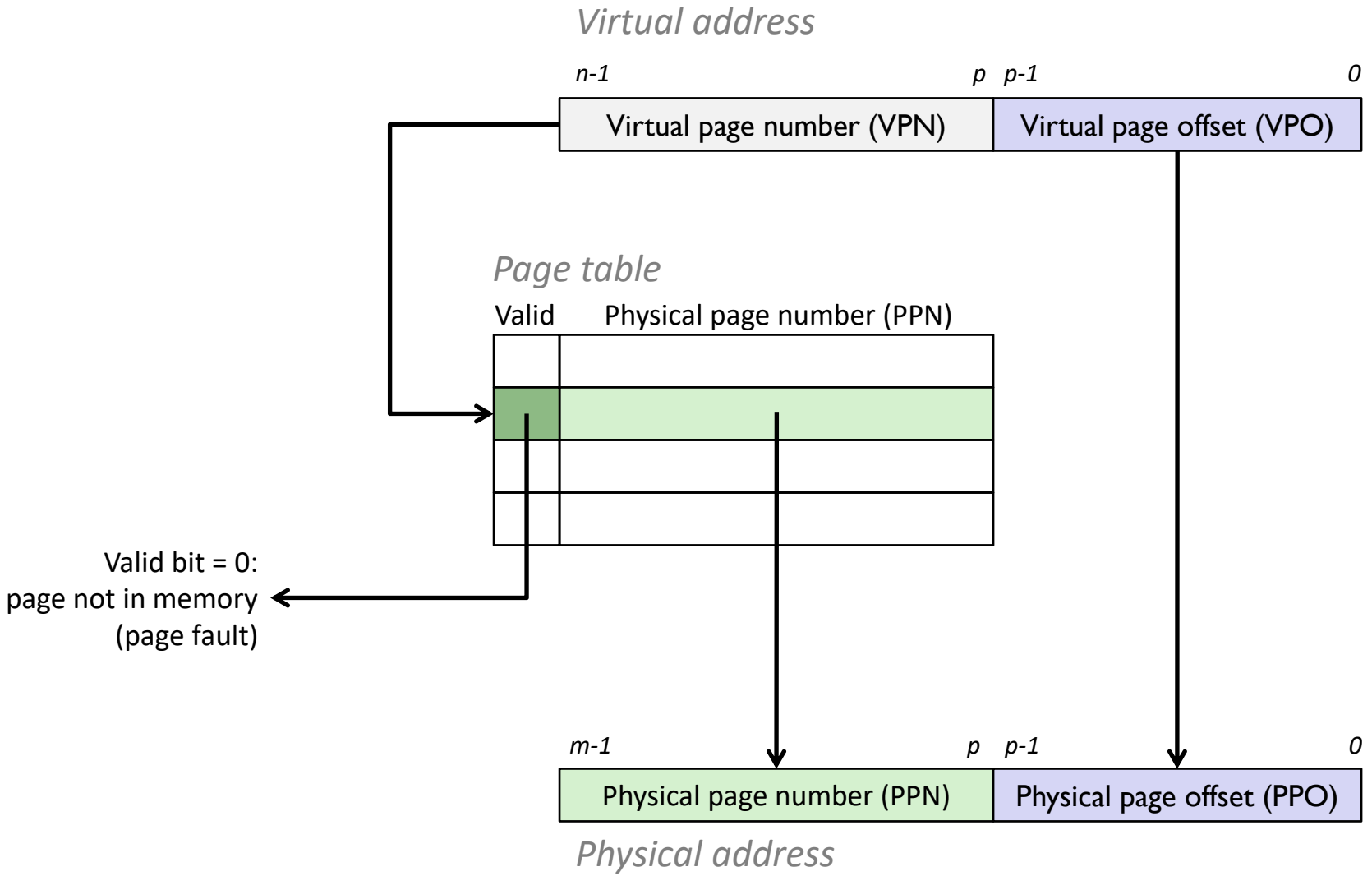
- Page miss causes page fault
- Page fault handler selects a victim to be evicted (here VP 4)
- Offending instruction is restarted



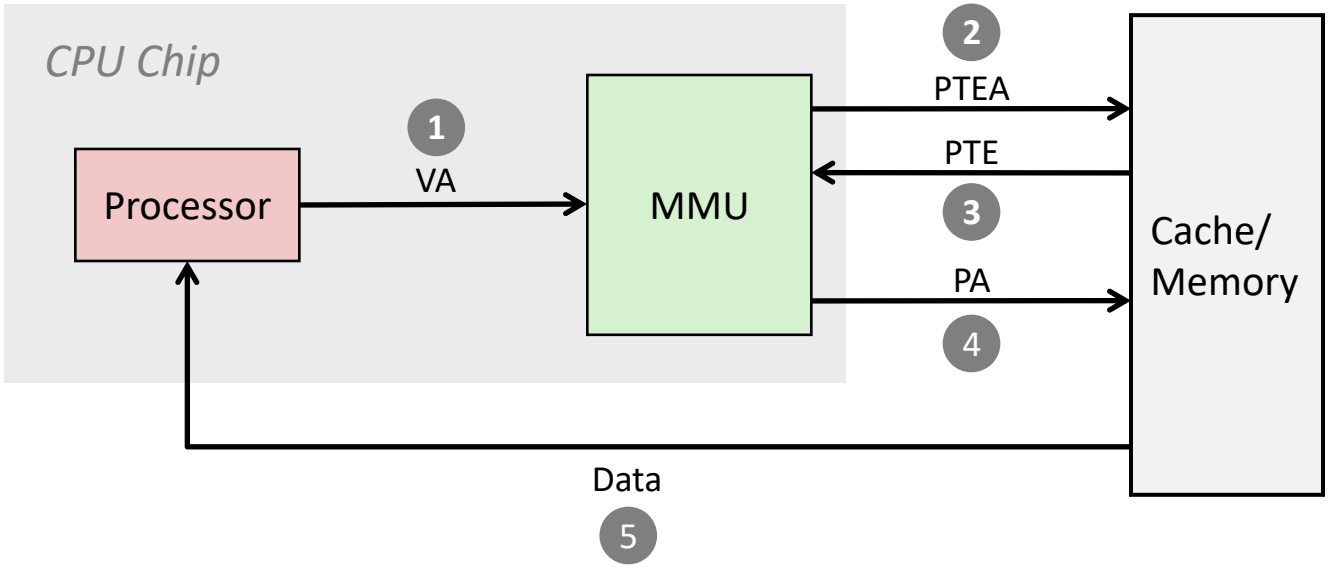
Locality to the Rescue Again!

- Virtual memory works because of locality
- At any point in time, programs tend to access a set of active virtual pages called the *working set*
  - Programs with better temporal locality will have smaller working sets
- If (working set size < main memory size)
  - Good performance for one process after compulsory misses
- If (SUM(working set sizes) > main memory size)
  - *thrashing:* performance meltdown where pages are swapped in and out continuously

Address translation with page table



Address Translation: Page Hit



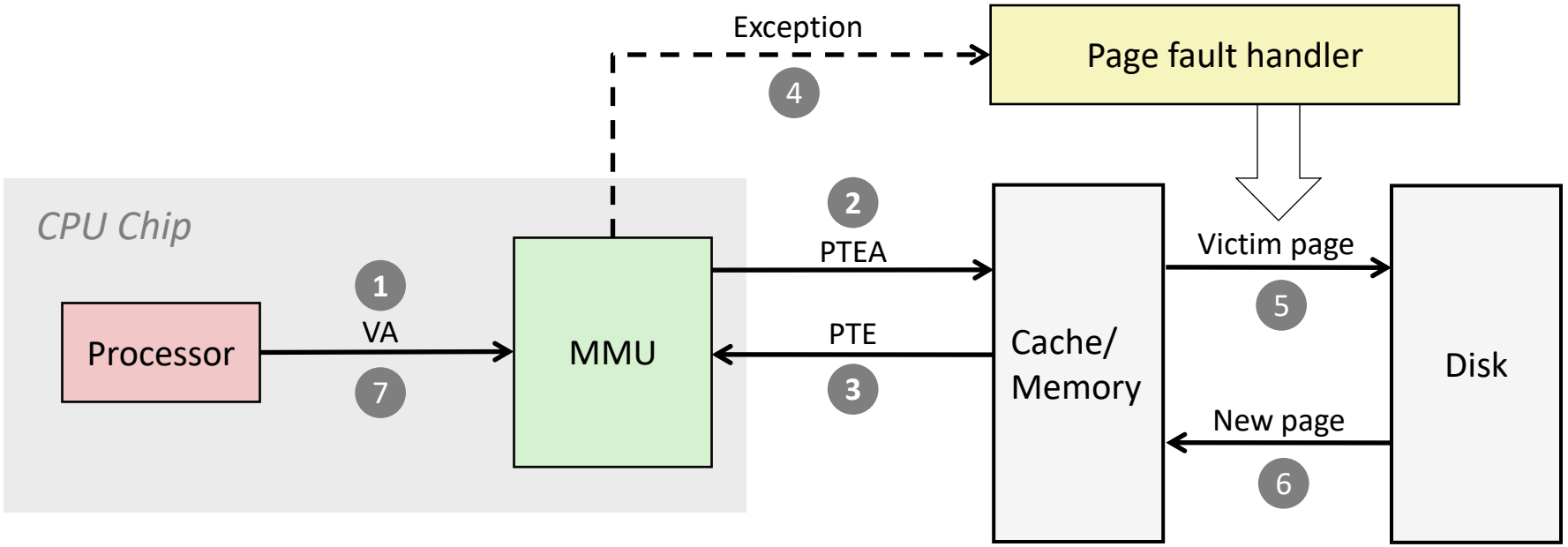
1) Processor sends virtual address to MMU

2-3) MMU fetches PTE from page table in memory

4) MMU sends physical address to cache/memory

5) Cache/memory sends data word to processor

Address Translation: Page Fault



1) Processor sends virtual address to MMU

2-3) MMU fetches PTE from page table in memory

4) Valid bit is zero, so MMU triggers page fault exception

5) Handler identifies victim (and, if *dirty*, pages it out to disk)

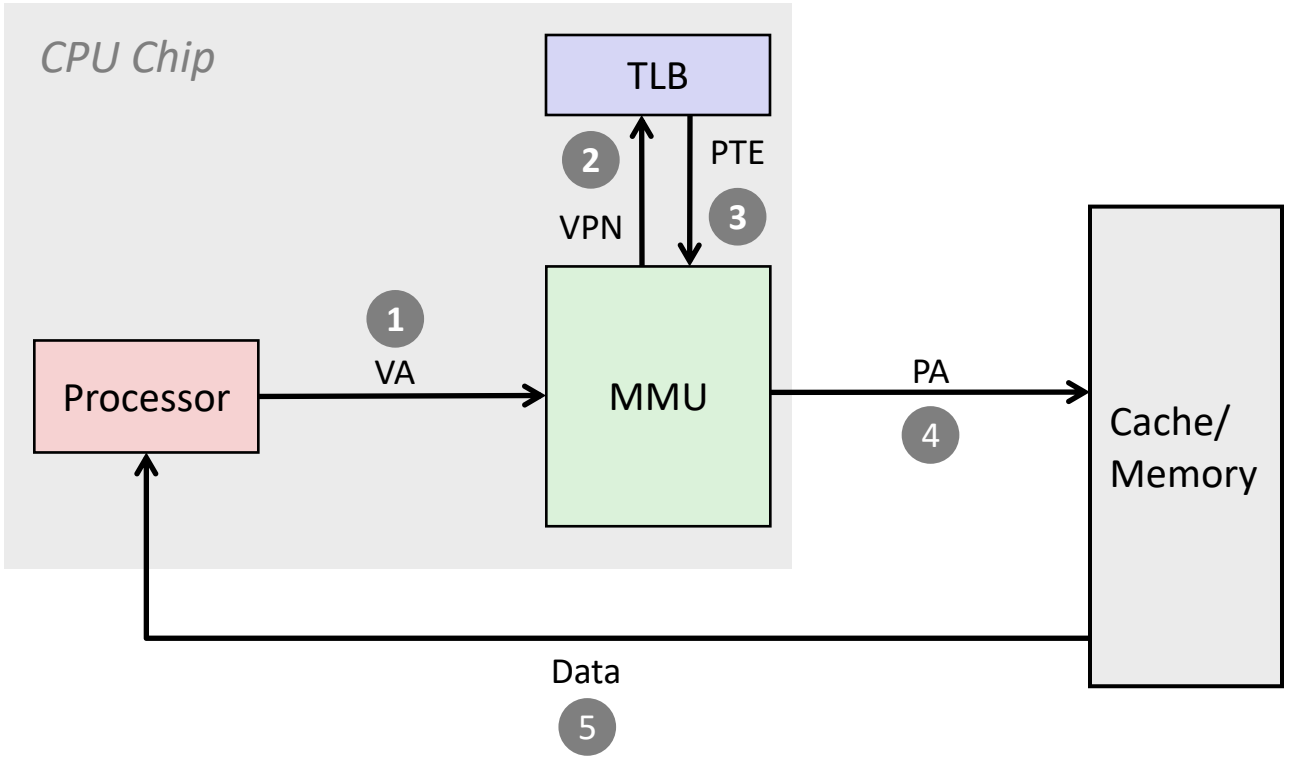
6) Handler pages in new page and updates PTE in memory

7) Handler returns to original process, restarting faulting instruction

Speeding up translation with a TLB

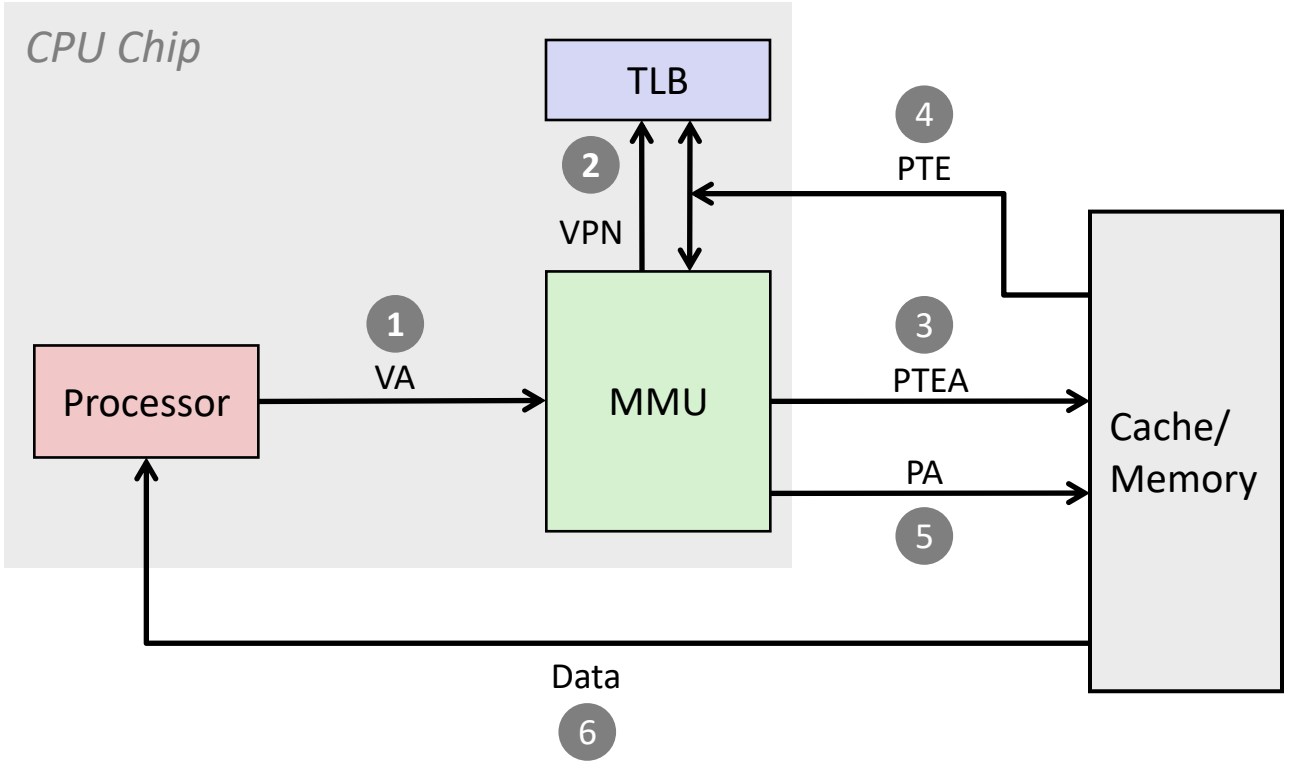
- *Translation Lookaside Buffer* (TLB)
  - Small hardware cache in MMU
  - Maps virtual page numbers to physical page numbers
  - Contains complete page table entries for small number of pages, i.e., it caches page table.

TLB Hit



A TLB hit eliminates a memory access

TLB Miss



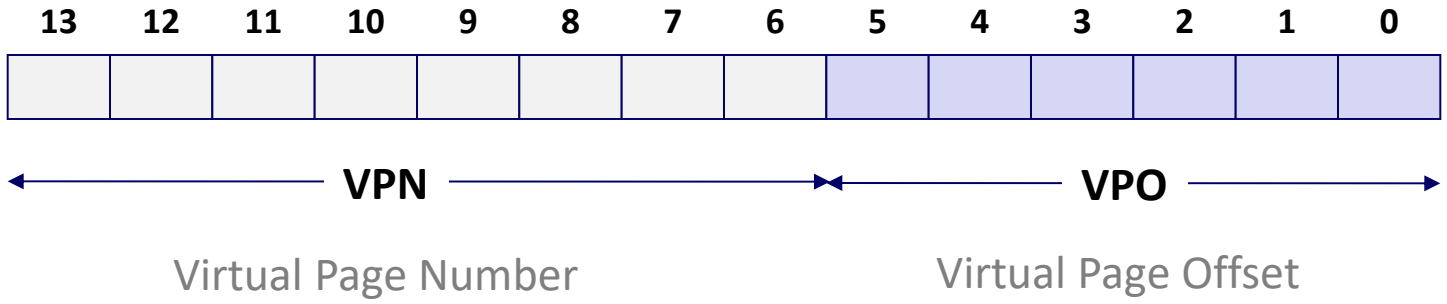
A TLB miss incurs an additional memory access

Simple Memory System Example

- Addressing
  - 14-bit virtual addresses
  - 12-bit physical address
  - Page size = 64 bytes = 2^6

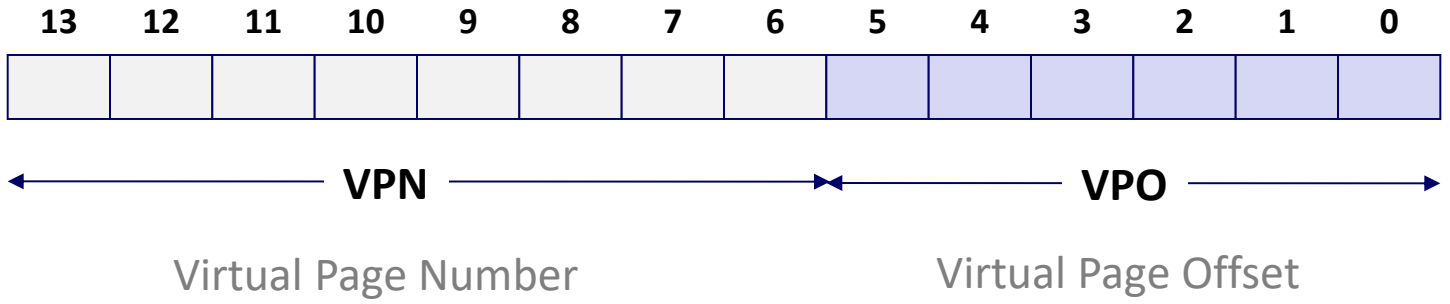
Simple Memory System Example

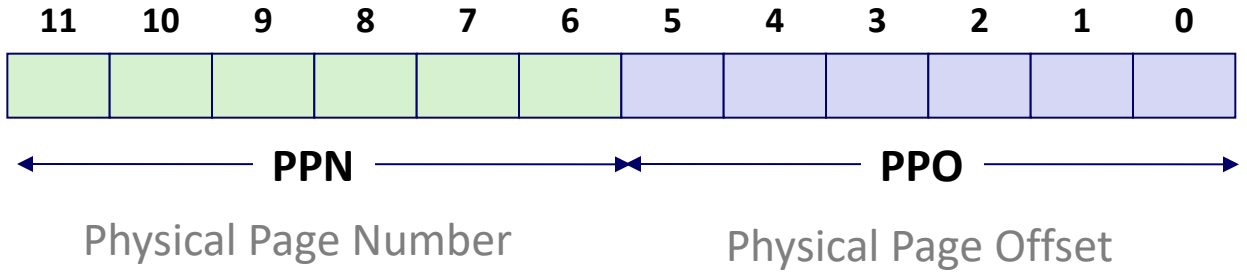
- Addressing
  - 14-bit virtual addresses
  - 12-bit physical address
  - Page size = 64 bytes = 2^6



Simple Memory System Example

- Addressing
  - 14-bit virtual addresses
  - 12-bit physical address
  - Page size = 64 bytes = 2^6





Simple Memory System Page Table

Q: How many entries are in the page table?

A: $2^{14}/2^{6} = 2^{8} = 256$

We only show first 16 entries (out of 256)

VPN	*PPN*	*Valid*
00	28	1
01	–	0
02	33	1
03	02	1
04	–	0
05	16	1
06	–	0
07	–	0

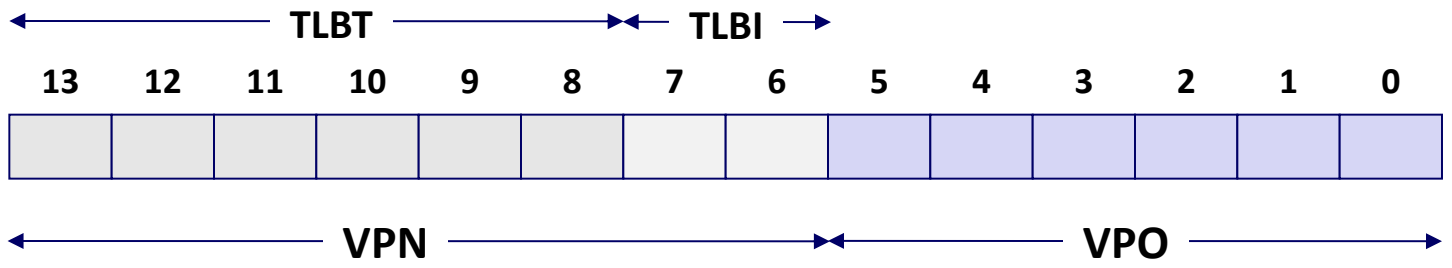
VPN	*PPN*	*Valid*
08	13	1
09	17	1
0A	09	1
0B	–	0
0C	–	0
0D	2D	1
0E	11	1
0F	0D	1

Simple Memory System TLB

- 16 entries
- 4-way associative

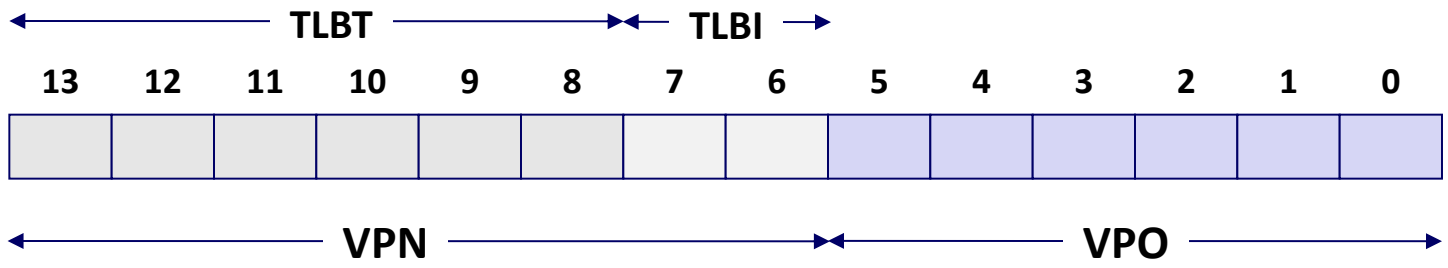
Simple Memory System TLB

- 16 entries
- 4-way associative



Simple Memory System TLB

- 16 entries
- 4-way associative



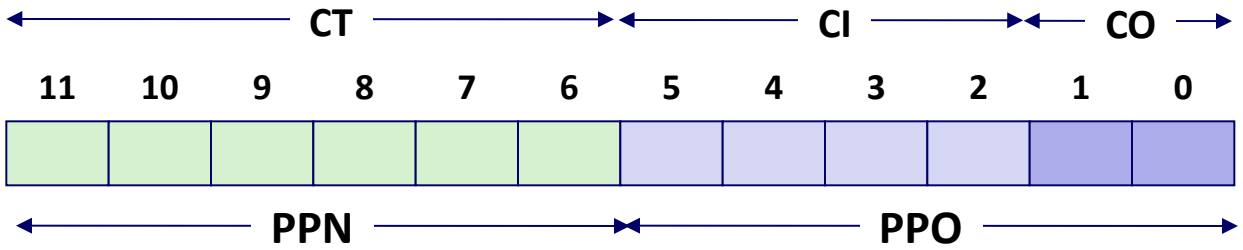
Set	*Tag*	*PPN*	*Valid*	*Tag*	*PPN*	*Valid*	*Tag*	*PPN*	*Valid*	*Tag*	*PPN*	*Valid*
0	03	–	0	09	0D	1	00	–	0	07	02	1
1	03	2D	1	02	–	0	04	–	0	0A	–	0
2	02	–	0	08	–	0	06	–	0	03	–	0
3	07	–	0	03	0D	1	0A	34	1	02	–	0

Simple Memory System Cache

- 16 lines, 4-byte block size
- Physically addressed
- Direct mapped

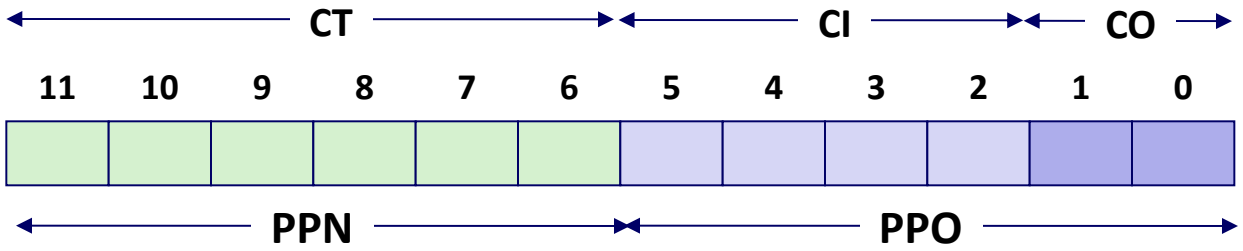
Simple Memory System Cache

- 16 lines, 4-byte block size
- Physically addressed
- Direct mapped



Simple Memory System Cache

- 16 lines, 4-byte block size
- Physically addressed
- Direct mapped

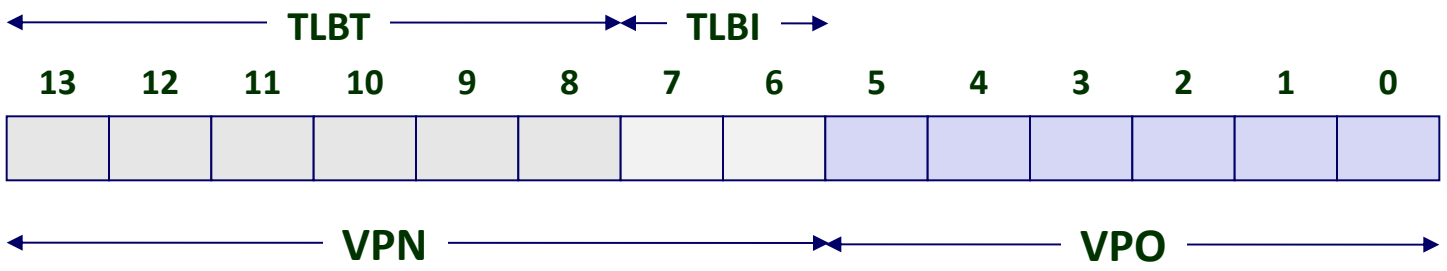


Idx	*Tag*	*Valid*	*B0*	*B1*	*B2*	*B3*
0	19	1	99	11	23	11
1	15	0	–	–	–	–
2	1B	1	00	02	04	08
3	36	0	–	–	–	–
4	32	1	43	6D	8F	09
5	0D	1	36	72	F0	1D
6	31	0	–	–	–	–
7	16	1	11	C2	DF	03

Idx	*Tag*	*Valid*	*B0*	*B1*	*B2*	*B3*
8	24	1	3A	00	51	89
9	2D	0	–	–	–	–
A	2D	1	93	15	DA	3B
B	0B	0	–	–	–	–
C	12	0	–	–	–	–
D	16	1	04	96	34	15
E	13	1	83	77	1B	D3
F	14	0	–	–	–	–

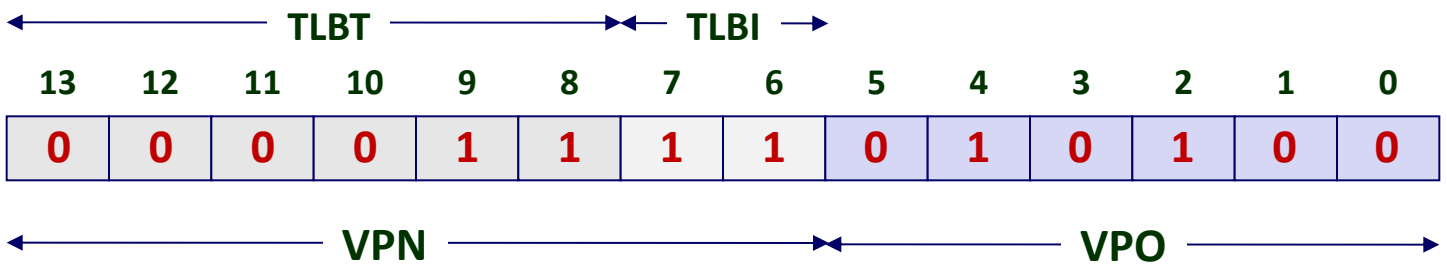
Address Translation Example

Virtual Address: `0x03D4`



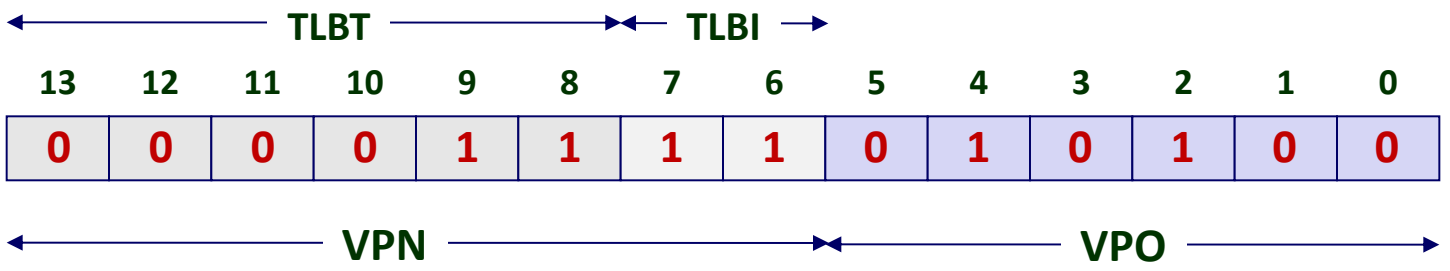
Address Translation Example

Virtual Address: `0x03D4`



Address Translation Example

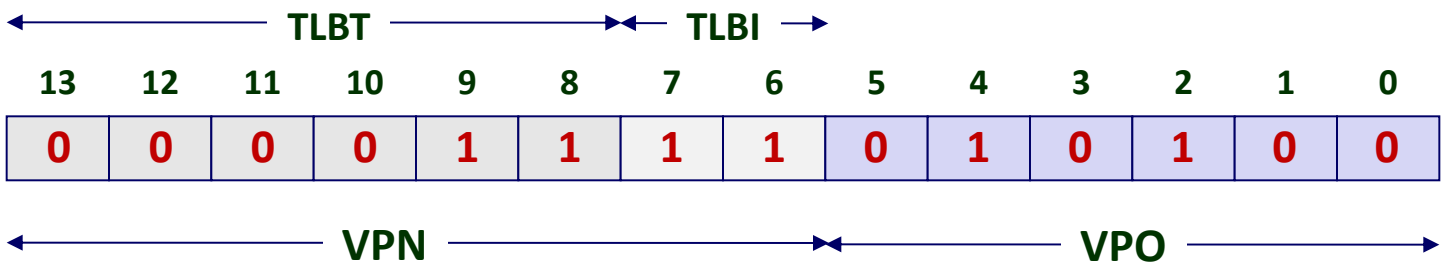
Virtual Address: `0x03D4`



VPN 0x0F TLBI 0x3 TLBT 0x03 TLB Hit? __ Page Fault? __ PPN: ____

Address Translation Example

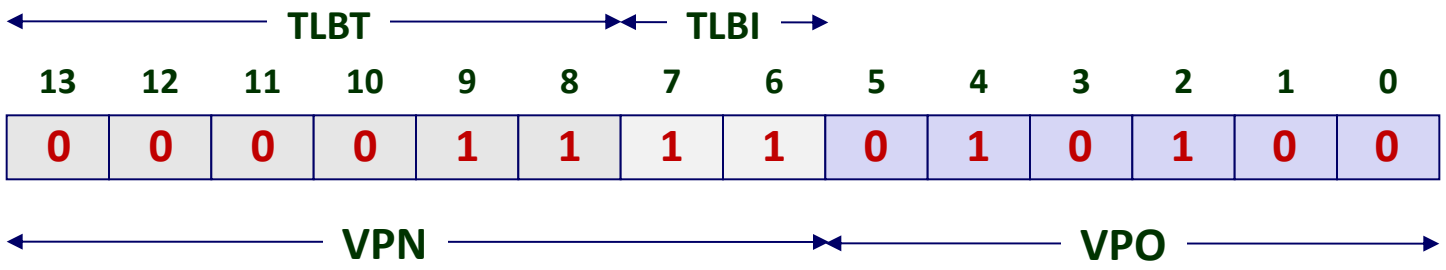
Virtual Address: `0x03D4`



VPN 0x0F TLBI 0x3 TLBT 0x03 TLB Hit? Y Page Fault? N PPN: 0x0D

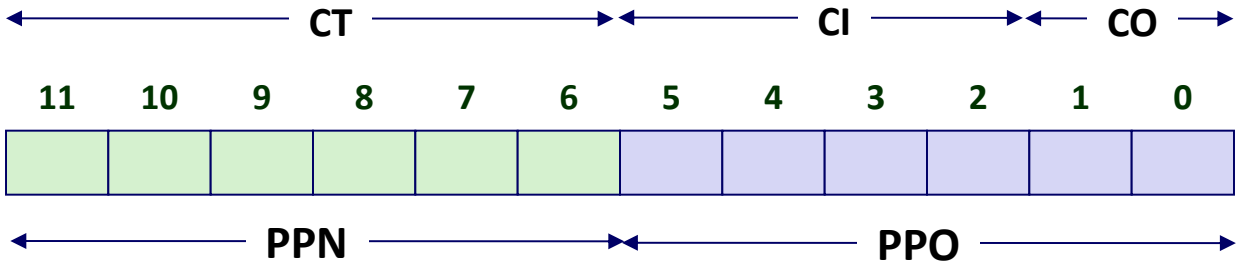
Address Translation Example

Virtual Address: 0x03D4



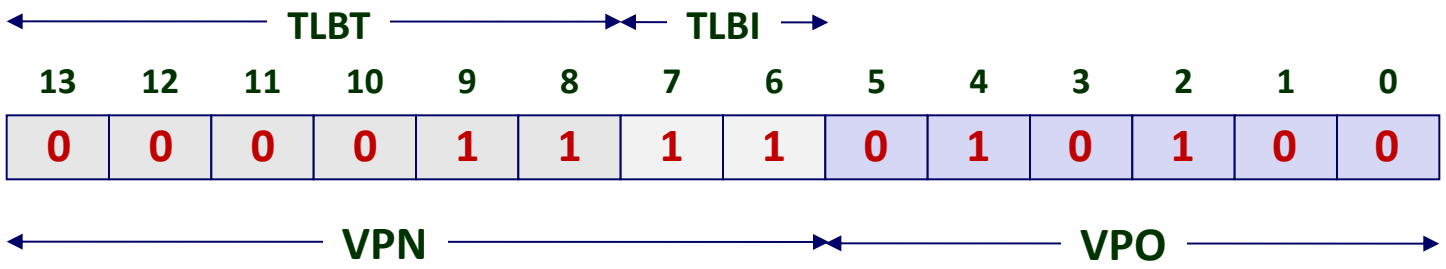
VPN 0x0F TLBI 0x3 TLBT 0x03 TLB Hit? Y Page Fault? N PPN: 0x0D

Physical Address



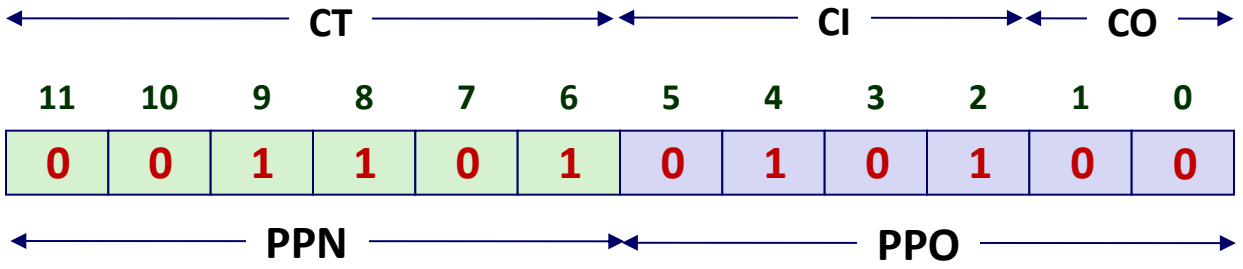
Address Translation Example

Virtual Address: 0x03D4



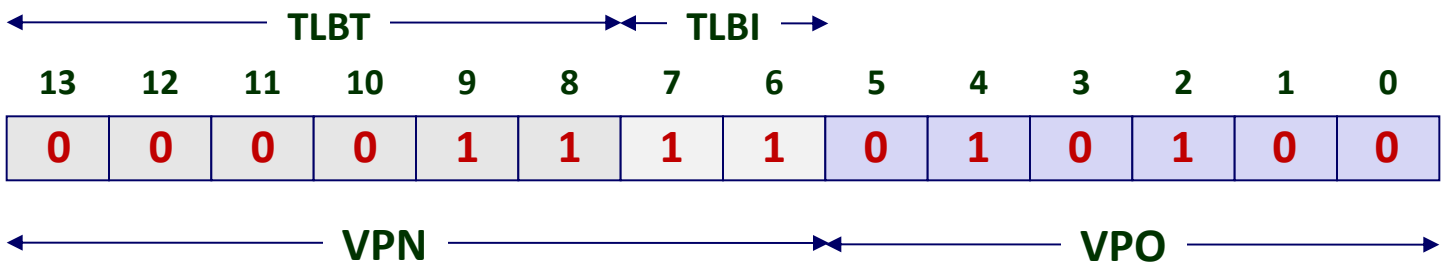
VPN 0x0F TLBI 0x3 TLBT 0x03 TLB Hit? Y Page Fault? N PPN: 0x0D

Physical Address



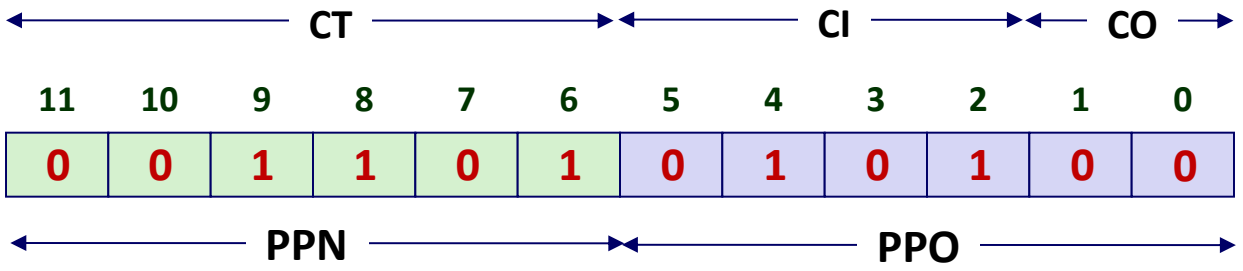
Address Translation Example

Virtual Address: 0x03D4



VPN 0x0F TLBI 0x3 TLBT 0x03 TLB Hit? Y Page Fault? N PPN: 0x0D

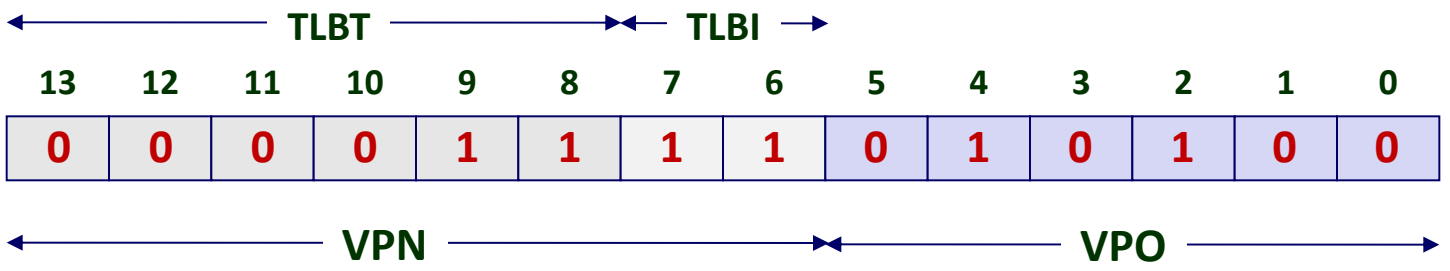
Physical Address



CO 0 CI 0x5 CT 0x0D Hit? __ Byte: ____

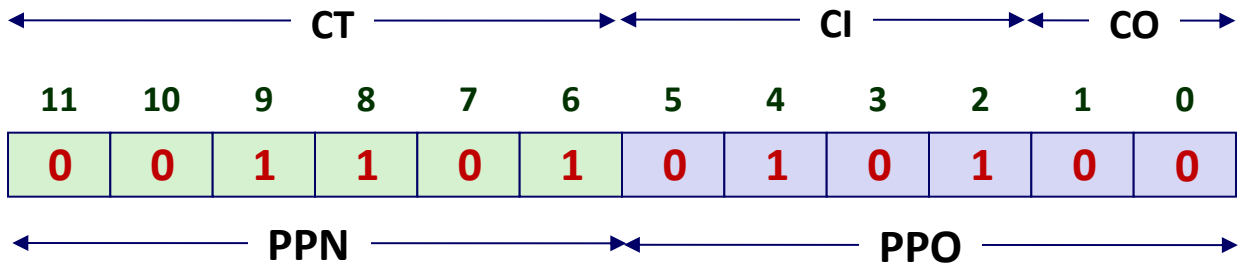
Address Translation Example

Virtual Address: 0x03D4



VPN 0x0F TLBI 0x3 TLBT 0x03 TLB Hit? Y Page Fault? N PPN: 0x0D

Physical Address



CO 0 CI 0x5 CT 0x0D Hit? Y Byte: 0x36