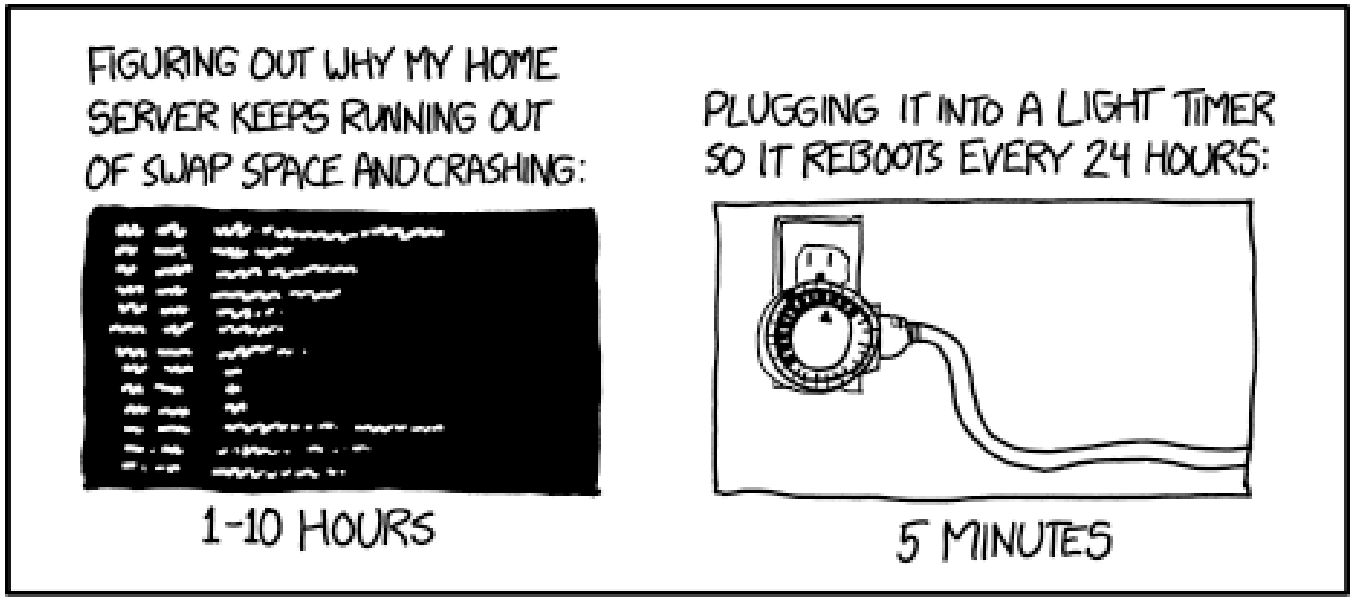
Virtual Memory II

CSE 351 Autumn 2023

Instructor:
Justin Hsia

Teaching Assistants:

Afifah Kashif	Malak Zaki
Bhavik Soni	Naama Amiel
Cassandra Lam	Nayha Auradkar
Connie Chen	Nikolas McNamee
David Dai	Pedro Amarante
Dawit Hailu	Renee Ruan
Ellis Haker	Simran Bagaria
Eyoel Gebre	Will Robertson
Joshua Tan	

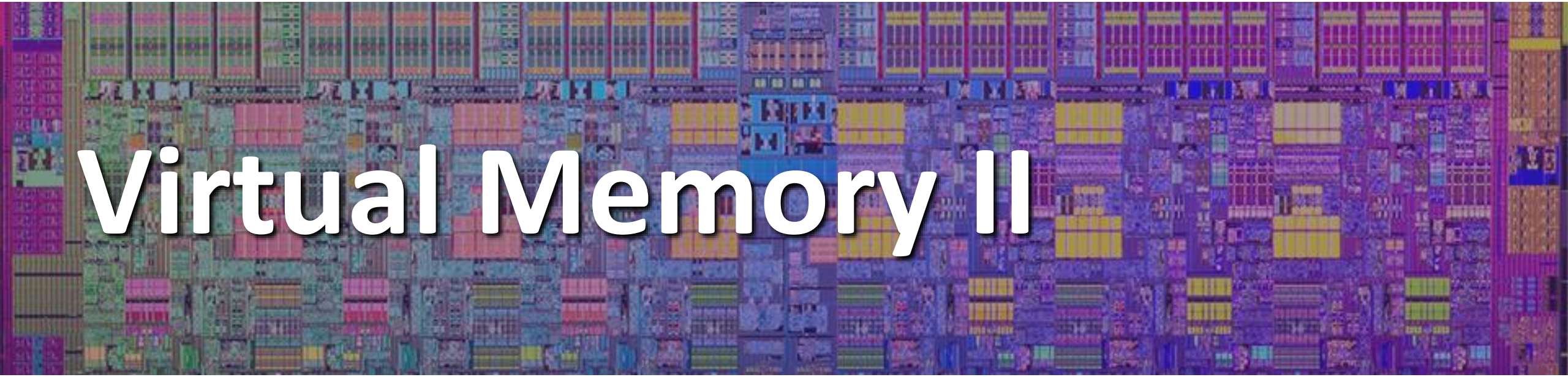


https://xkcd.com/1495/

Relevant Course Information

- HW24 due Friday, HW25 due next Wednesday (12/6)
- Today is the last day to submit Lab 4
- Lab 5 due next Thursday (12/7)
  - The most significant amount of C programming you will do in this class – combines lots of topics from this class: pointers, bit manipulation, structs, examining memory
  - Understanding the concepts *first* and efficient *debugging* will save you lots of time
  - Light style grading

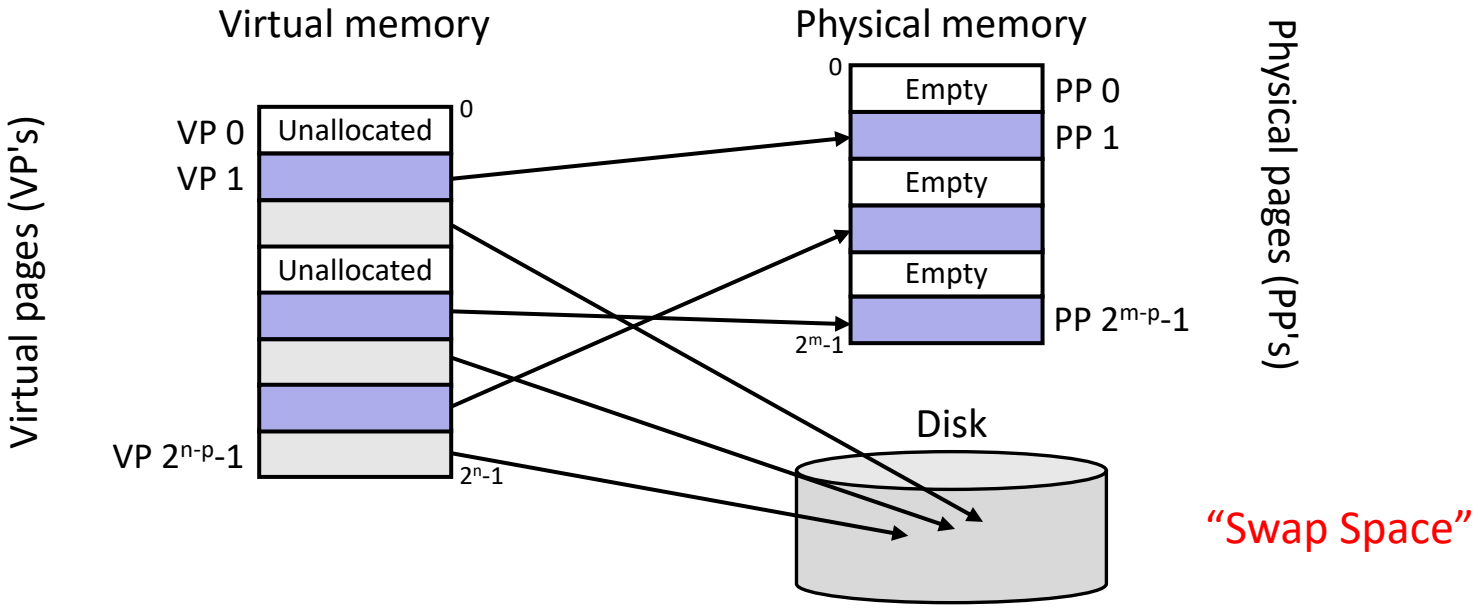
- No lessons in Week 11 – "normal" lectures
- Final exam: 12/11-13
  - Final review section on 12/7, final review session on 12/8



Virtual Memory II

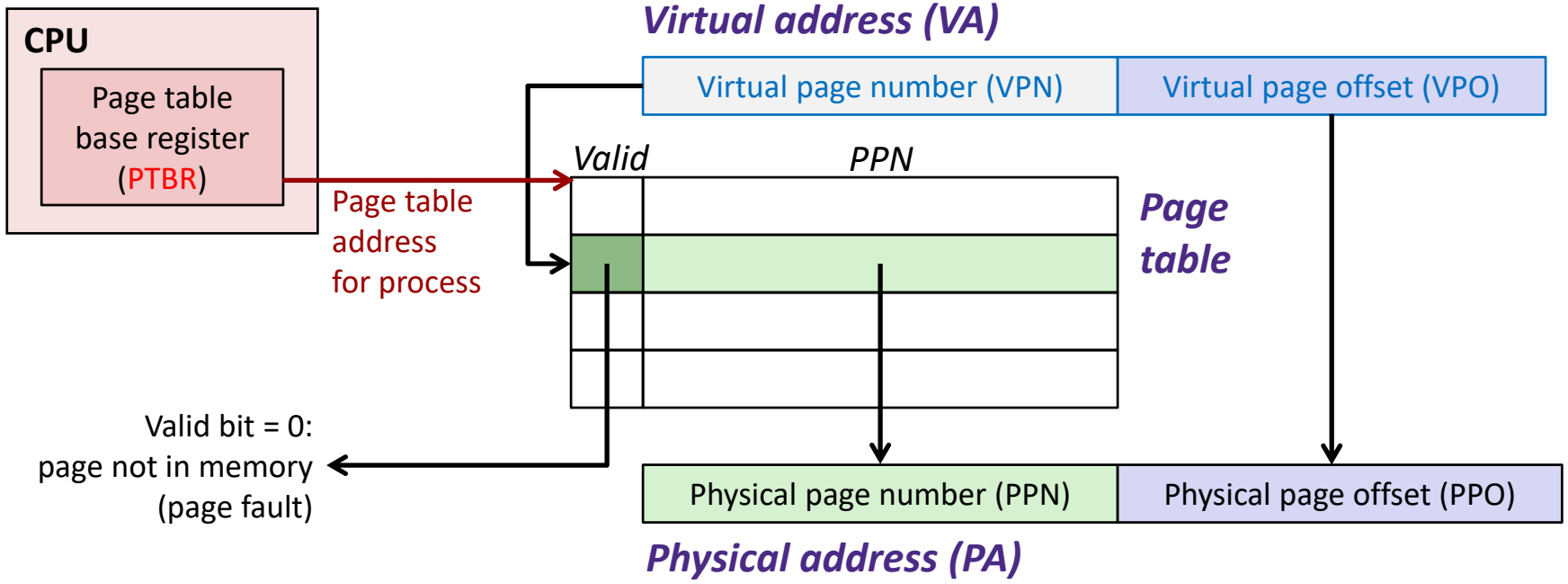
Lesson Summary (1/3)

- Can think of physical memory as a cache of virtual memory
  - Data is transferred between physical memory and swap space (disk) in ***pages***
  - Physical memory has caching parameters and properties
    - Large page size, fully associative, write-back, replacement policy
  - Caveats: virtual pages may not exist, data doesn't have to exist in both physical memory and disk



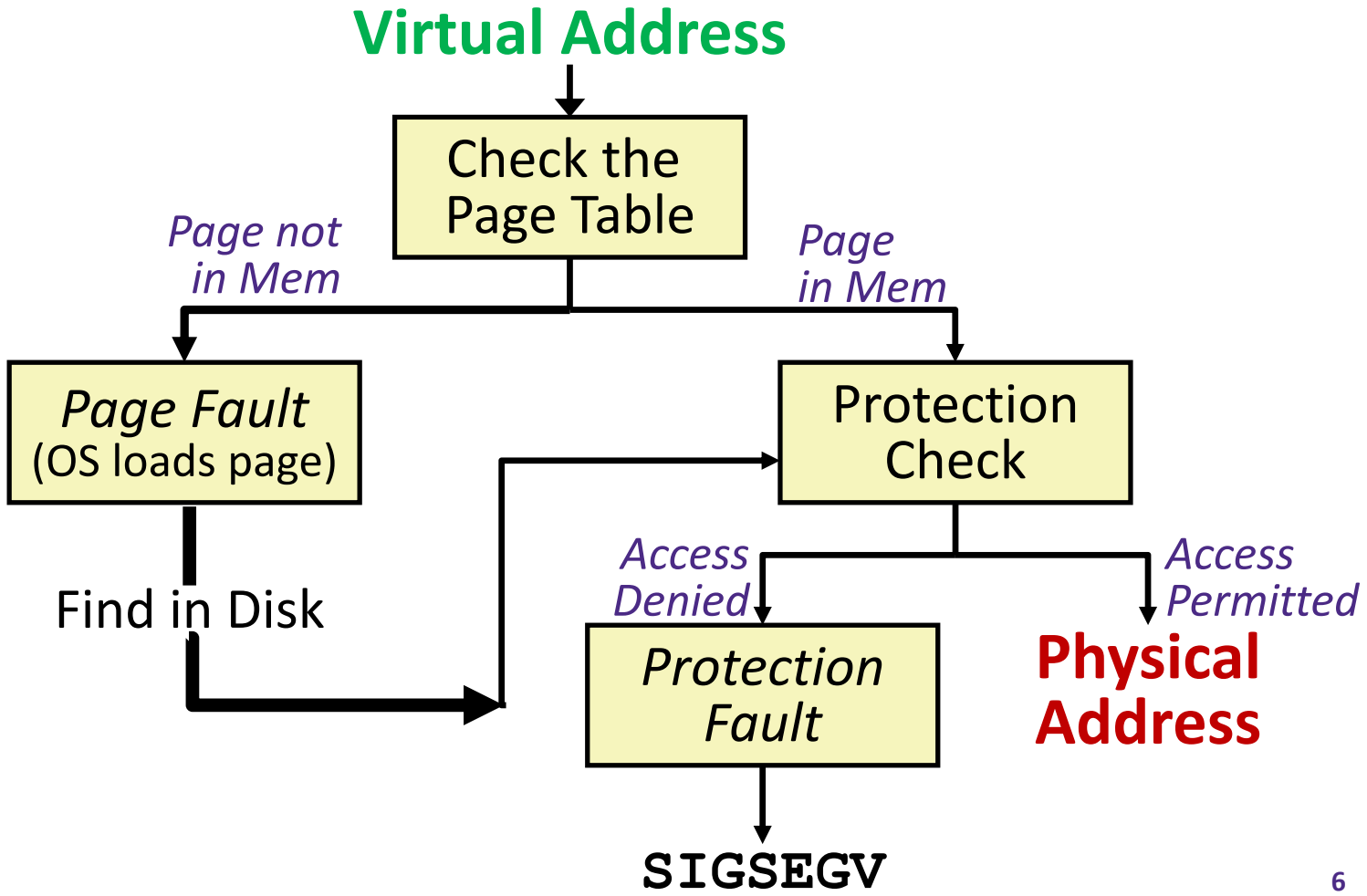
Lesson Summary (2/3)

- Address translation done via ***page tables***
  - Lookup tables (one per process) that map VPN → PPN
  - Uses management bits: valid bit, access rights (read, write, execute)
  - Stored in memory – page table for currently-running process is pointed to by ***page table base register*** (PTBR)



Lesson Summary (2/3)

- The address translation story (SO FAR) is check the page table in memory
  - Input: VPN, Output: PPN
  - *Page Fault:* Fetch page from disk to memory, update corresponding page table entry
  - *Page Table Hit:* Use existing page table entry

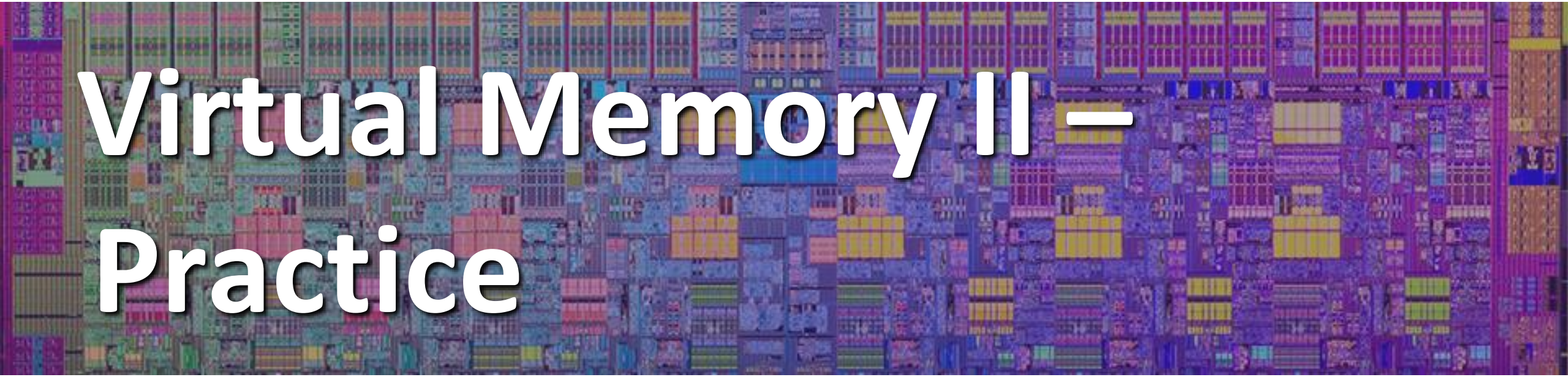


Lesson Q&A

- Terminology:
  - Paging: page size (P), page offset width (p) virtual page number (VPN), physical page numbers (PPN)
  - Page table (PT): page table entry (PTE), access rights (read, write, execute)

- Learning Objectives:
  - Determine virtual memory parameters related to addresses, page tables, [and TLBs].
  - Perform address translations (virtual address → physical address).
  - Describe the relationships between virtual memory parameters and policies.

- What lingering questions do you have from the lesson?

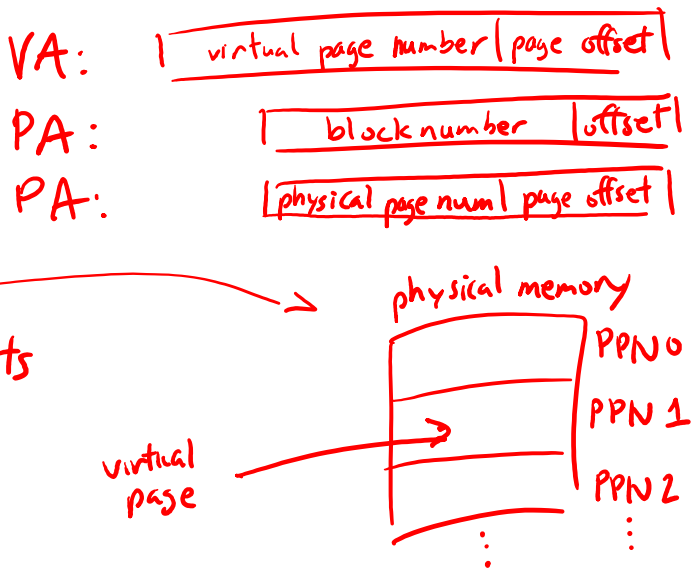


Virtual Memory II – Practice

Virtual Memory Concept Questions

- Which terms from caching are most similar/analogous to the new virtual memory terms?
  - block #, block size, cache line, cache set, index width, management bits, offset width, tag width

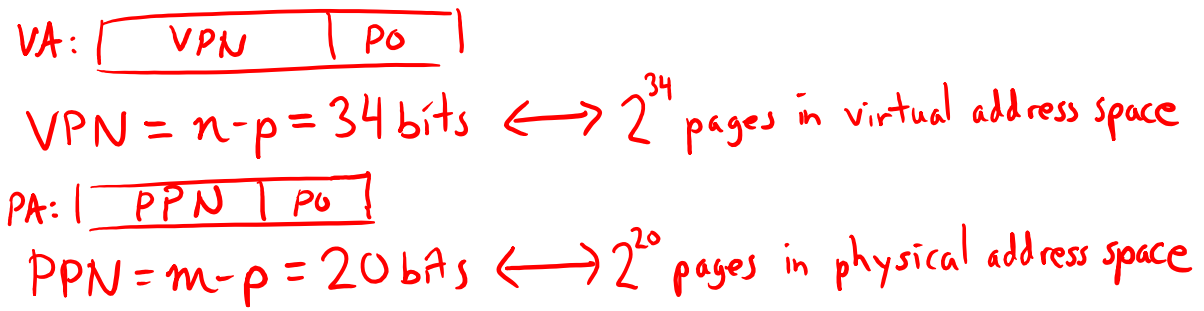
  - page size
    block size
  - page offset width
    (block) offset width
  - virtual page number
    block number
  - physical page number
    block number or cache set
  - page table entry
    cache line: data of interest + management bits
  - access rights
    management bits



VM Parameters Question

- How many bits wide are the following fields?
  - 16 KiB pages (2^{14} bytes, 2^{10} bytes) → $p = 14$ bits
  - 48-bit virtual addresses → $n = 48$ bits ⟷ 256 TiB virtual memory
  - 16 GiB physical memory (2^{34} bytes, 2^{30} bytes) → $m = 34$ bits

	VPN	PPN
(A)	34	24
(B)	32	18
(C)	30	20
(D)	34	20



Memory Review Question

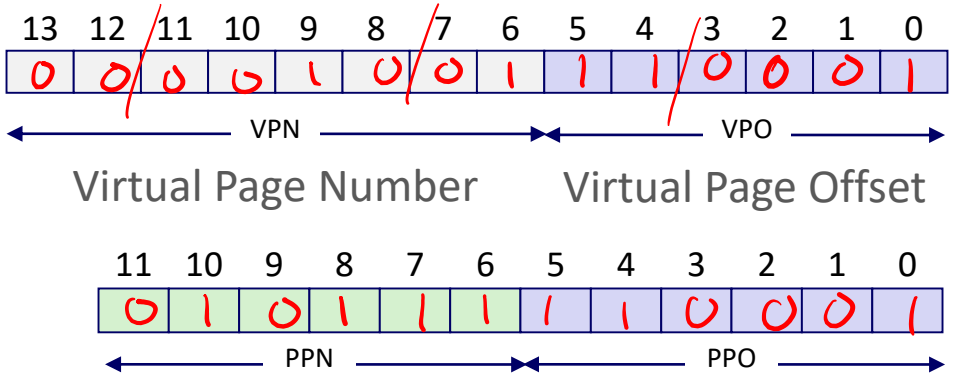
- What should the permission bits be for pages from the following sections of virtual memory?

Section	Read	Write	Execute
Stack	1	1	0
Heap	1	1	0
Static Data	1	1	0
Literals	1	0 (constants)	0
Instructions	1	0 (don't alter code)	1 (only instructions should be executable)

static in size

Homework Setup

- Simple Memory System Example
  - 14-bit virtual addresses n = 14 bits
  - 12-bit physical address m = 12 bits
  - Page size = 64 bytes = 2^6, p = 6 bits wide
  - Only showing first 16 entries of page table:
    - Other management bits not shown



VPN	PPN	Valid
0	28	1
1	–	0
2	33	1
3	02	1
4	–	0
5	16	1
6	–	0
7	–	0

VPN	PPN	Valid
8	13	1
9	17	1
A	09	1
B	–	0
C	–	0
D	2D	1
E	–	0
F	0D	1

- Questions:
  - Give a virtual address that causes a page fault.
    VPN of 1, 4, 6, 7, B, C, or E, any VPO
  - What virtual address corresponds to the physical address 0x5F1?
    PPN = 0x17 → VPN = 0x09, same offset so 0x0271

Group Work Time

- During this time, you are encouraged to work on the following:
  1) If desired, continue your discussion
  2) Work on the homework problems
  3) Work on the current lab

- Resources:
  - You can revisit the lesson material
  - Work together in groups and help each other out
  - Course staff will circle around to provide support