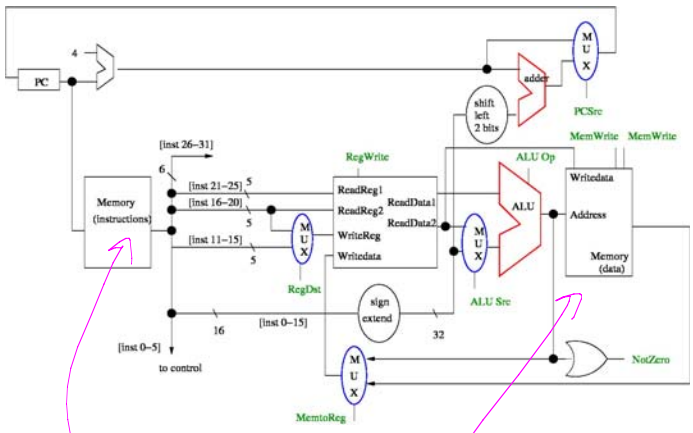
lecture 17

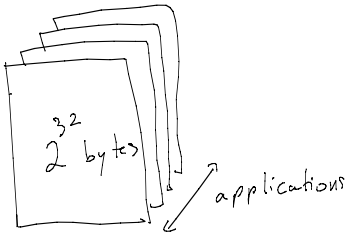
virtual memory vs. physical memory



Today

virtual memory

physical memory



"Process"
(running program)

- RAM
- disk
- flash
- etc...

Sizes of Memory

$2^{10} \approx 1$ KB (kilobyte)

$2^{20} \approx 1$ MB (megabyte)

$2^{30} \approx 1$ GB (gigabyte)

$2^{40} \approx 1$ TB (terabyte)

$2^{50} \approx 1$ PB (petabyte)

$2^{60} \approx 1$ EB (exabyte)

⋮

Floppy disk (1.4 MB)

(obsolete)

magnetic

CD (1 GB), DVD (10 GB)

optical (laser)

Hard Disk Drive (HDD)

200 GB - 1 TB

magnetic

HDD access time ?

- 100 rotations per second (fast?)

$\Rightarrow$ average access $\frac{1}{2 \times 100}$ sec

$= 5$ ms

$\Rightarrow \frac{10^9 \text{ clock cycles}}{\text{sec}} \times \frac{1}{200} \text{ sec} = \frac{\frac{1}{2} 10^7 \text{ clock cycles}}{\text{access}}$

SLOW!

Flash (SSD - solid state drive)

USB

card (digital camera, cell phone)

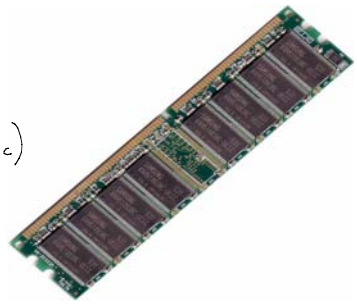
tablet PC (flash can replace HDD)

access time $\sim 10^{-4}$ sec

Semiconductor (silicon)

RAM

SRAM
- fast and expensive
- access in one clock cycle (10^{-9} sec)



DRAM
- slower but less expensive
- access in ~10 clock cycles

Semiconductor (silicon)

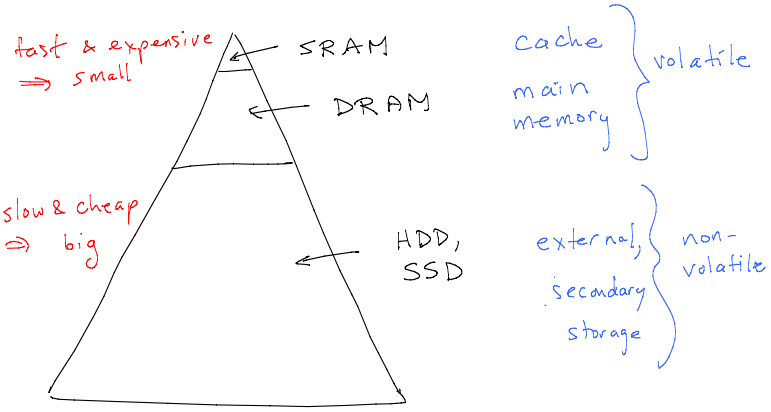
volatile vs. non-volatile

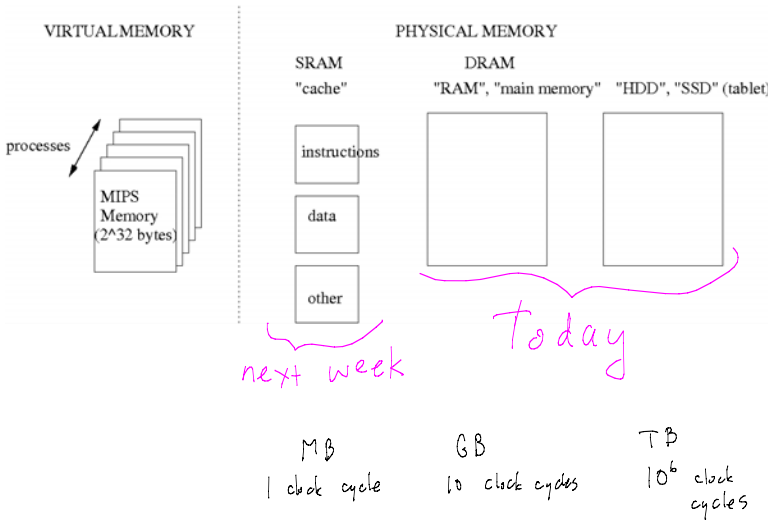
- RAM (SRAM, DRAM)
- flash (SSD)
- disk (HDD, CD, DVD)

(Does storage vanish when power is off?)

Memory "Hierarchy"

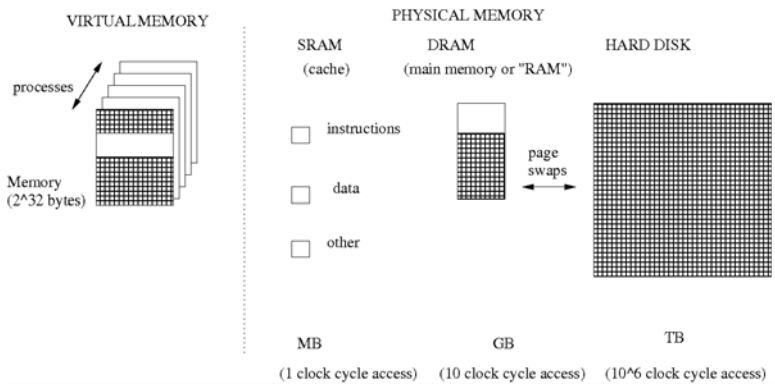
(access from top)





How translate a virtual address to a physical address ?

Key Concept : a page



Example: 1 page = 2^{12} bytes

How many pages ?

Virtual Memory (2^{32} bytes)

RAM (e.g. 1 GB)

i.e. DRAM

HDD (e.g. 1 TB)

Example: 1 page = 2^{12} bytes

Virtual Memory

2^{20} pages $\left(2^{32} = 2^{20} \cdot 2^{12}\right)$

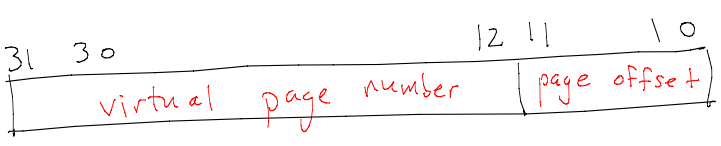
Physical Memory

RAM 2^{18} pages $\left(2^{30} = 2^{18} \cdot 2^{12}\right)$

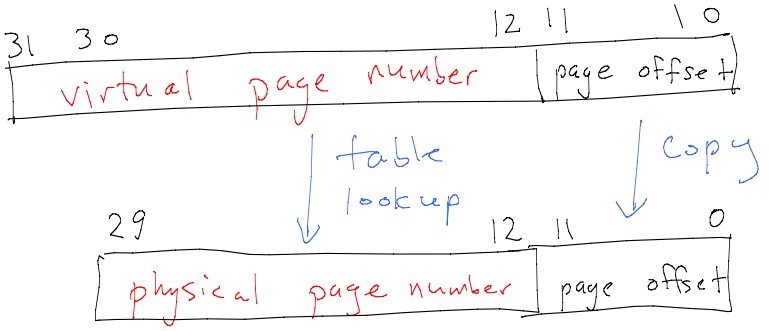
HDD 2^{28} pages $\left(2^{40} = 2^{28} \cdot 2^{12}\right)$

How translate a virtual address to a physical address ?

virtual address.

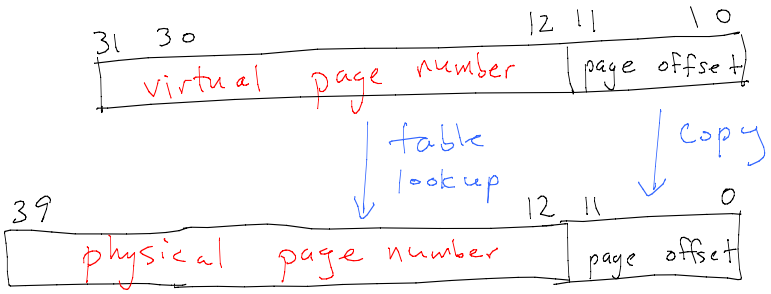


virtual address.



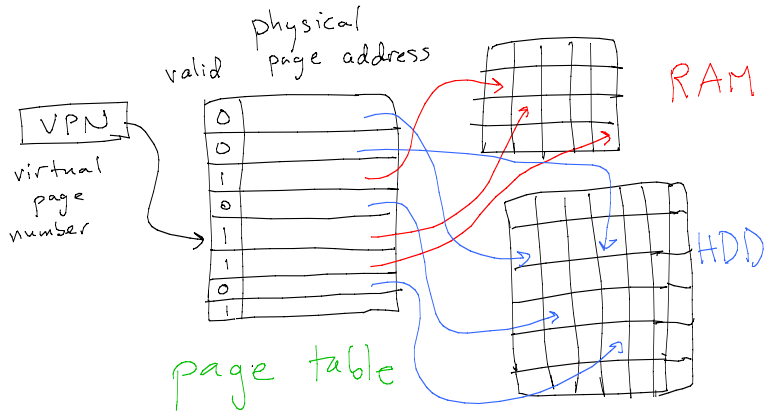
physical address (main memory - RAM)

virtual address.

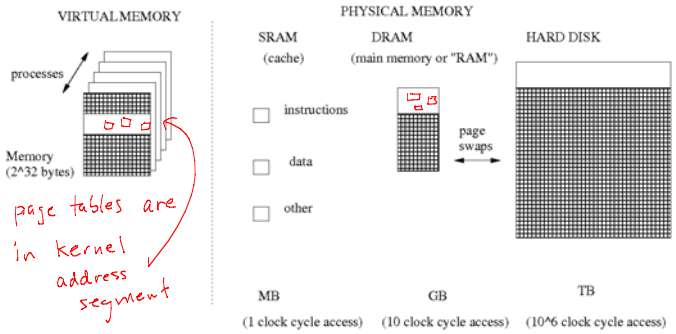


physical address (HDD)

Translation: virtual page → physical page

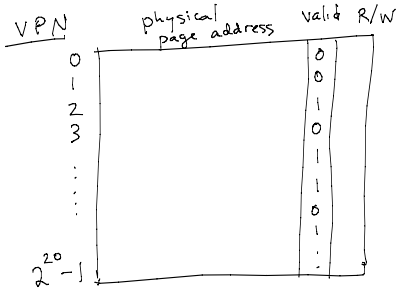


Where are the page tables?



The part of Memory holding page tables is not mapped using page tables. Rather, the V→P mapping is fixed.

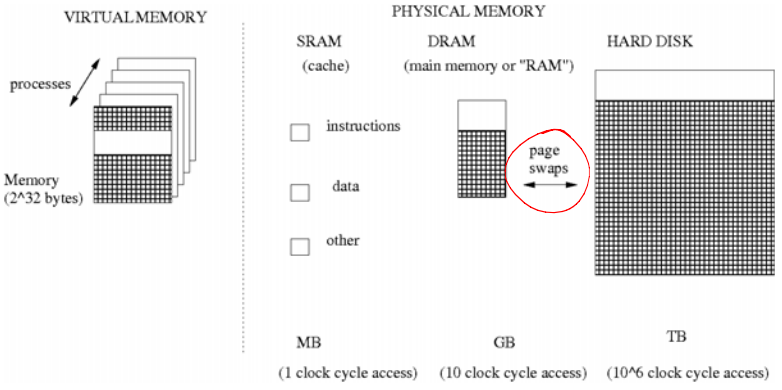
Note (ASIDE)



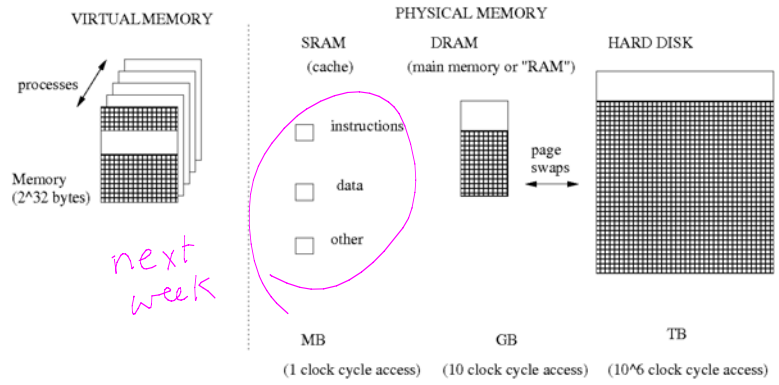
Don't need data structure to hold all possible (VPN, phys. page). e.g. use hash table instead.

Page fault

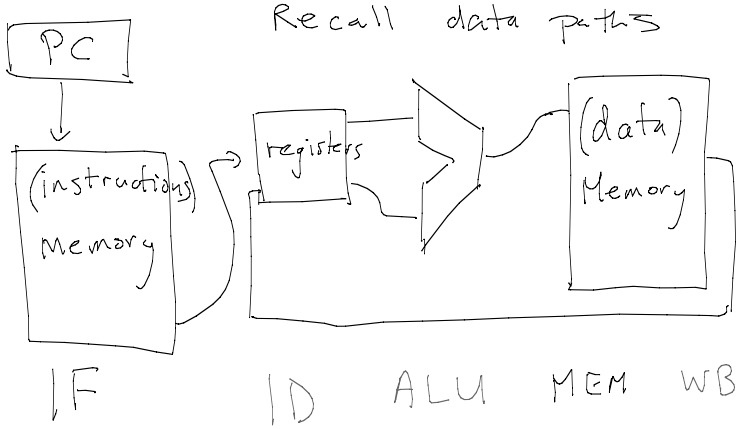
- When a program tries to access an address that belongs to a page on HDD, this page must be brought into main memory.
- A page swap occurs if a page must be moved out.
- The page table must be updated.
- All this is done by a kernel program called the page fault handler



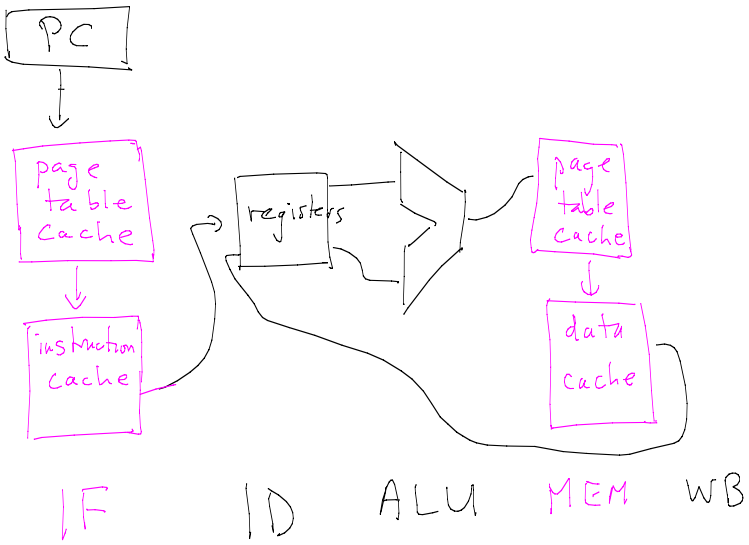
More on page swaps later in course (and in COMP 310)



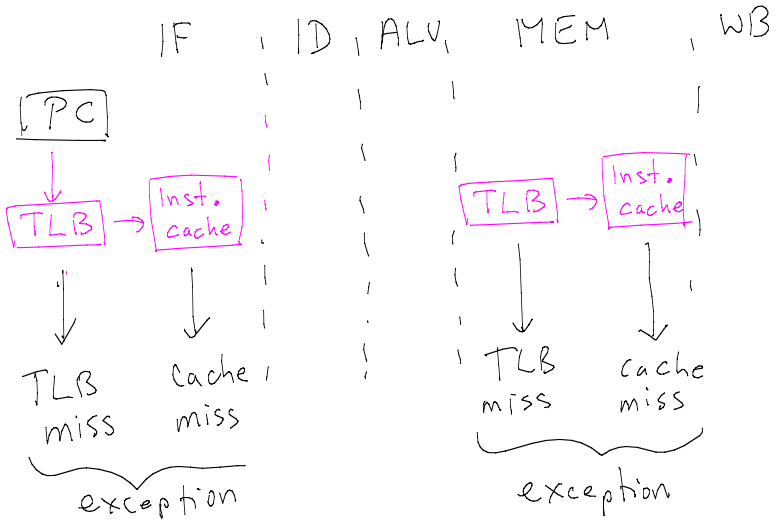
We need V → P translation for every fetch and every memory access. The cache allows us to do this in one clock cycle.



We need memory access in one clock cycle!



TLB - translation lookaside buffer



Next week

- Cache for page tables (TLB)
- Cache for data, instructions

Following 2 weeks

- I/O (including HDD, page faults)

Last week (after Easter break)

- TBD, not on final exam.