High Performance Computing

Matthew Jacob

Indian Institute of Science

Least Recently Used (LRU) Policy

- Keep track of when each page was last used
  - With a timestamp
  - LRU page: the one with the smallest timestamp
  - Requires a large number of comparisons
- Or, keep track of the stack of recently used pages
  - LRU page: at the bottom of the stack
  - Stack must be updated on every memory access
- So, LRU might be too expensive in practise

Alternative Page Replacement Policies

1. First in First Out (FIFO)
   - Keep track of the order in which the pages came into memory
   - Advantage: Does not have to be updated when a page is re-accessed
   - Problem: Unlike LRU, FIFO does not guarantee that the number of page faults will decrease if you increase the size of main memory

Alternative Page Replacement Policies

1. First in First Out (FIFO)
2. Approximate LRU

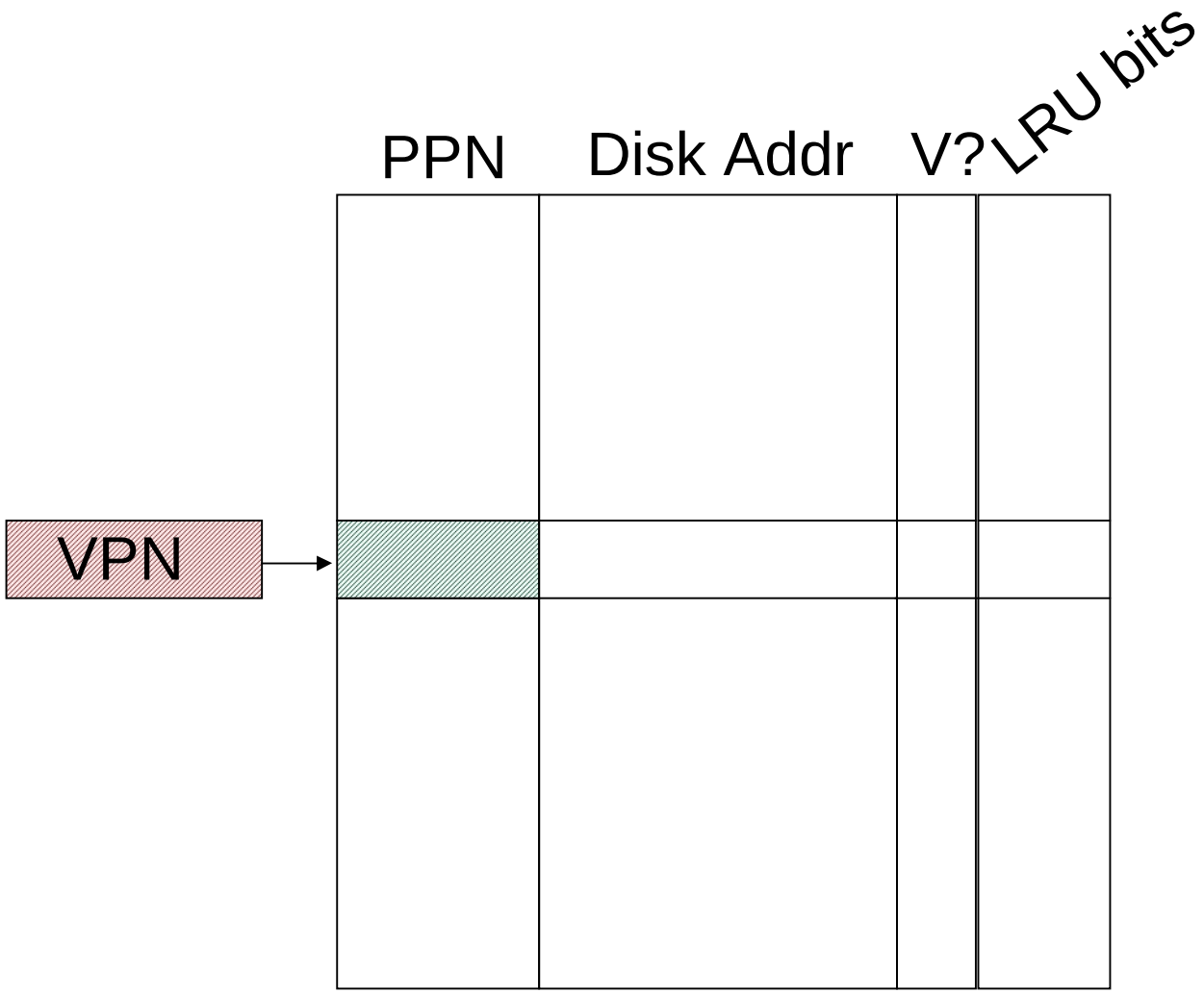
Example: Maintain 1 bit of status information with each physical page

- Initialized all the bits to 0
- Set a bit to 1 when that page is referenced
- Replace a page that has its bit equal to 0
- Reset all the bits to 0 every once in a while

Alternative Page Replacement Policies

1. First in First Out (FIFO)
2. Approximate LRU
3. Random
   - Randomly pick an i which is between 1 and n
   - Replace page P_i

Keeping Track: Page Table Entry

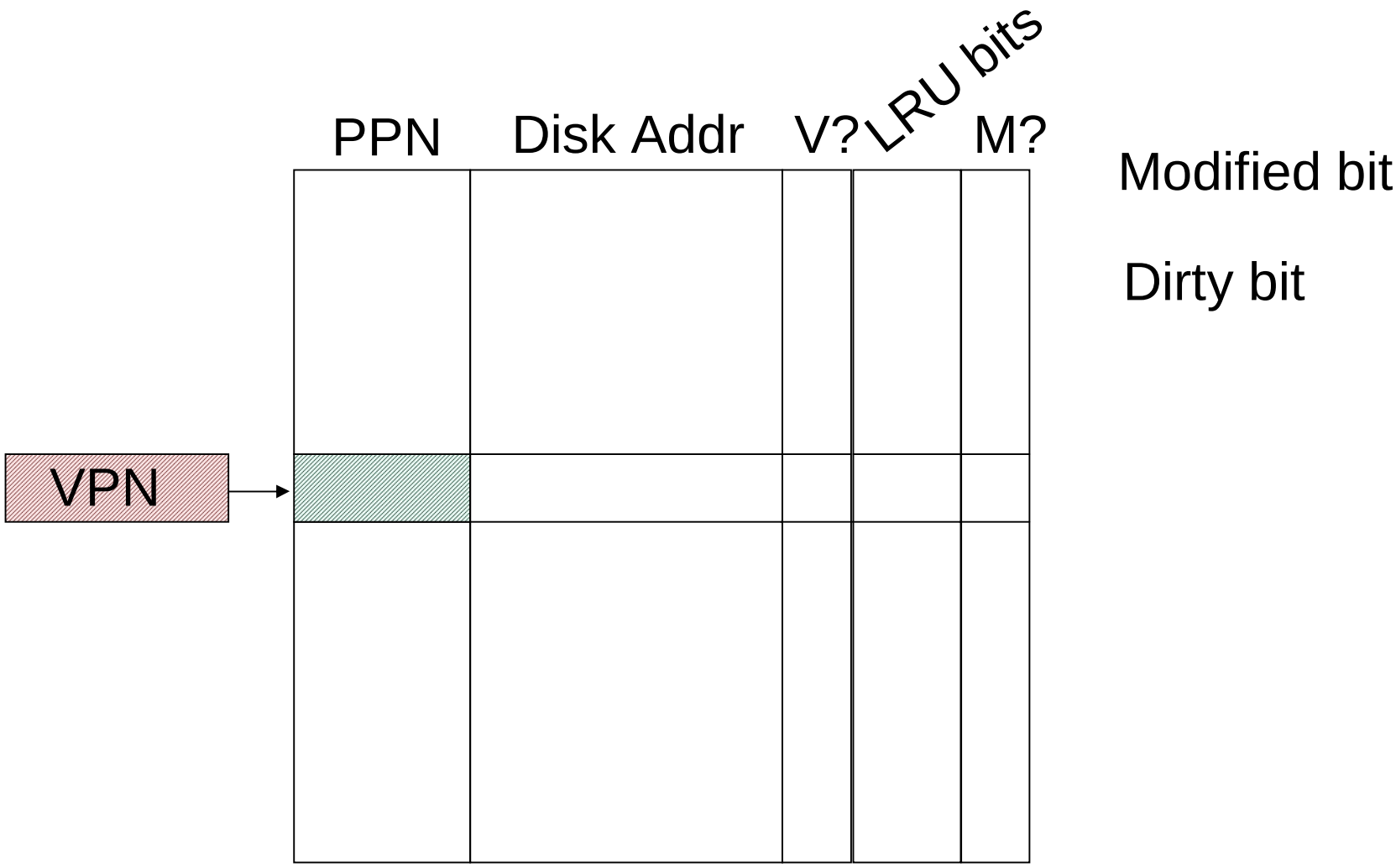


Page Fault Handler

1. Identify slot in main memory to be used
2. Get page contents from disk
3. Update page table entry

- Problem: The victim page identified by the page replacement policy might have been modified while it was in main memory
- It cannot just be overwritten by the incoming page
  - but must first be copied back to disk
- Optimization: Keep track of whether it has been modified while in memory

Keeping Track: Page Table Entry

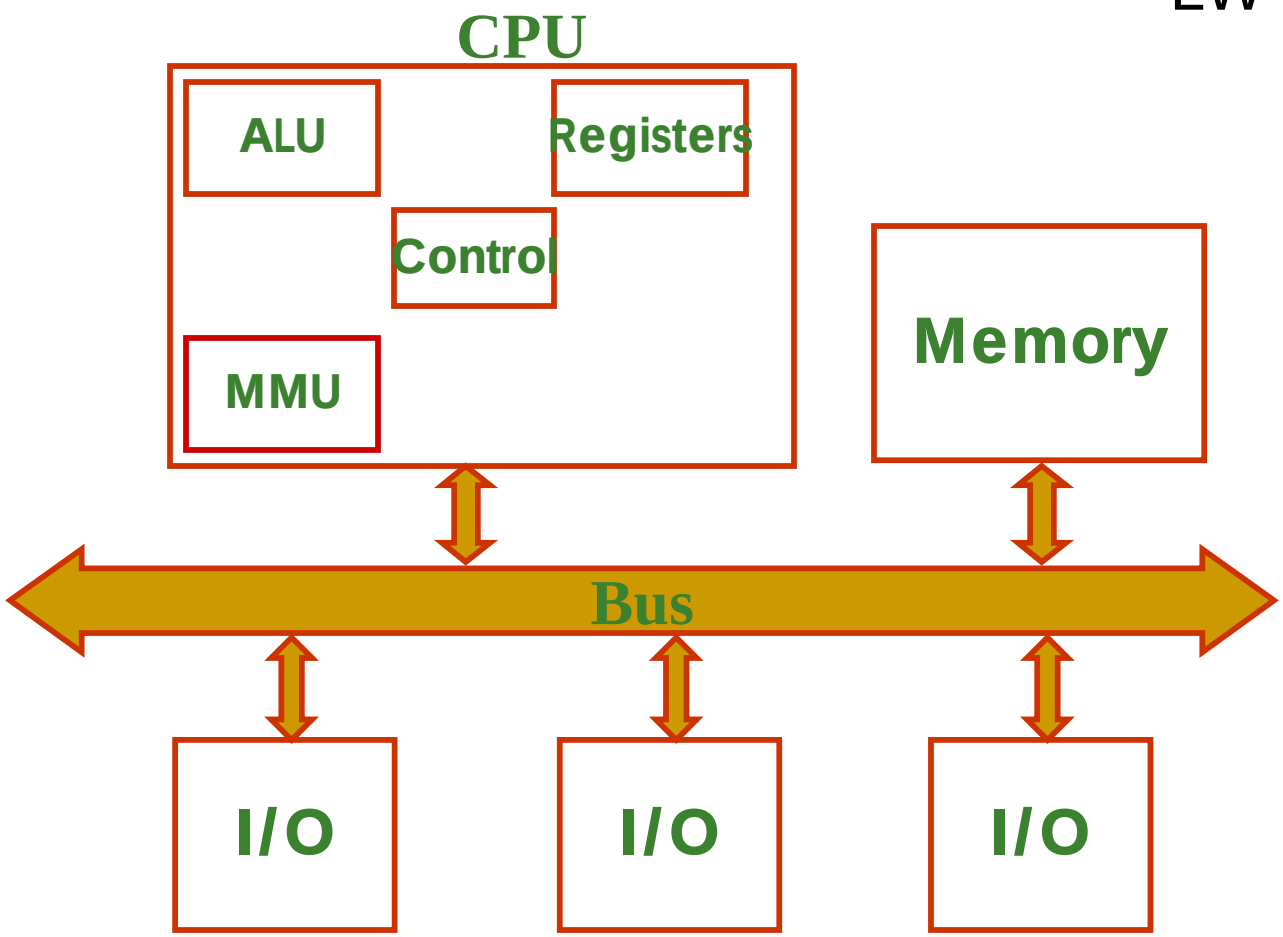


Implementation of Address Translation

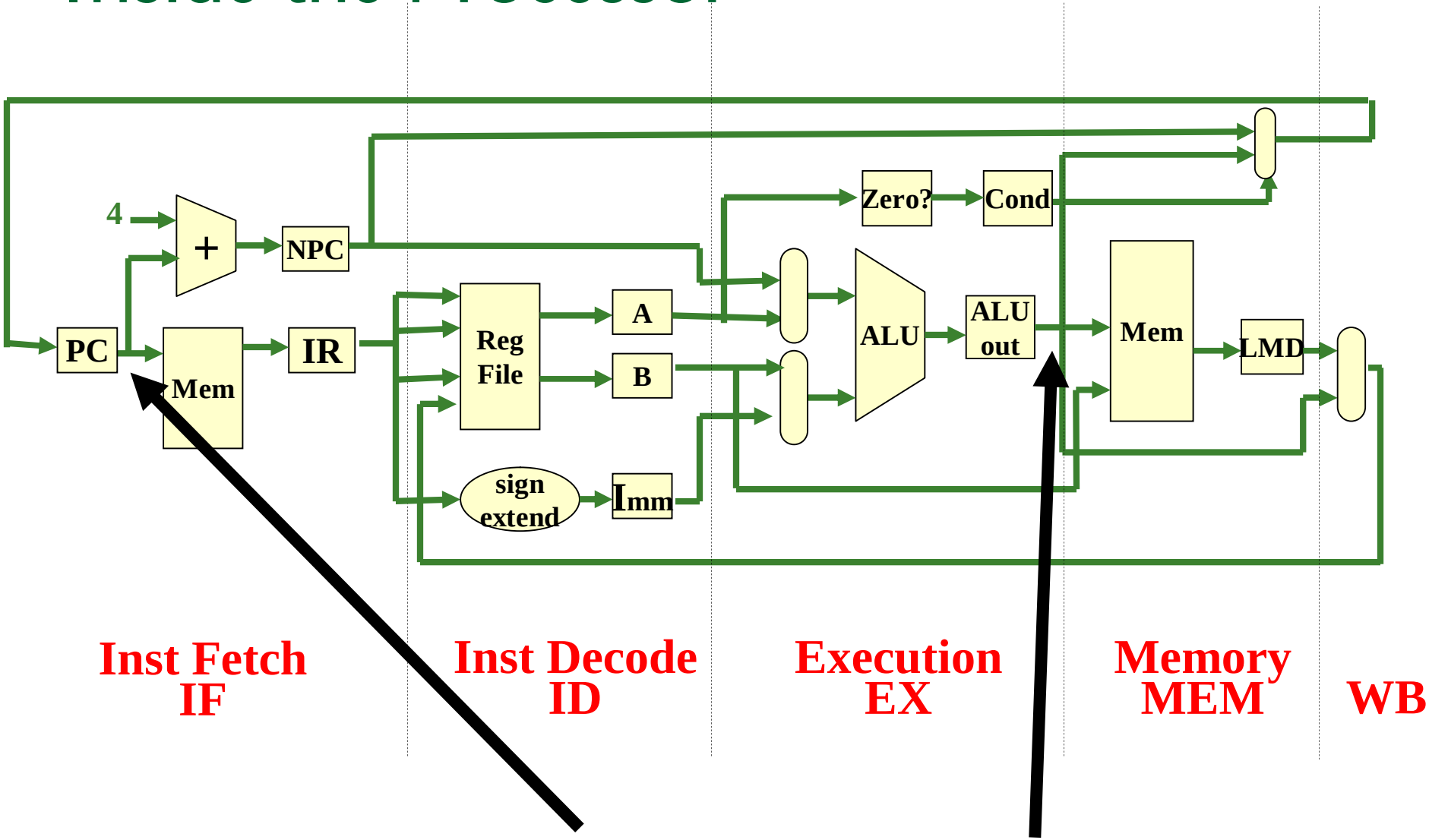
- Memory Management Unit (MMU): part of CPU; hardware that does address translation
- Recall: The Big Problem of page table size
  - e.g., 32b addresses, 256B page size
  - Page table size: 64MB (per process)
  - Could be stored in memory
  - Probably in a virtual address space
  - To translate a virtual memory address, the MMU has to read the relevant page table entry out of memory

Computer Organization: Hardware

LW R1, -8(R29)



Inside the Processor



Implementation of Address Translation

- Memory Management Unit (MMU): part of CPU; hardware that does address translation
- Recall: The Big Problem of page table size
  - e.g., 32b addresses, 256B page size
  - Page table size: 64MB (per process)
  - Could be stored in memory
  - Probably in a virtual address space
  - To translate a virtual memory address, the MMU has to read the relevant page table entry out of memory
  - Hardware must provide page table entries faster than that (most of the time)

Implementation of Address Translation

- Translation Lookaside Buffer (TLB): memory in MMU that contains some page table entries that are likely to be needed soon
- Recall: Cache memory contains data/instructions that the CPU is likely to need soon
- If the required page table entry is present in the TLB, then the MMU can do the translation fast
- Otherwise: TLB miss
- Must be handled, possibly like the OS handles a page fault

Recall: Page

Unit of memory management

- Translation: There is one translation table entry per page
- Data movement: A page of data is copied together between main memory and disk
- We used the example of 256B in each page

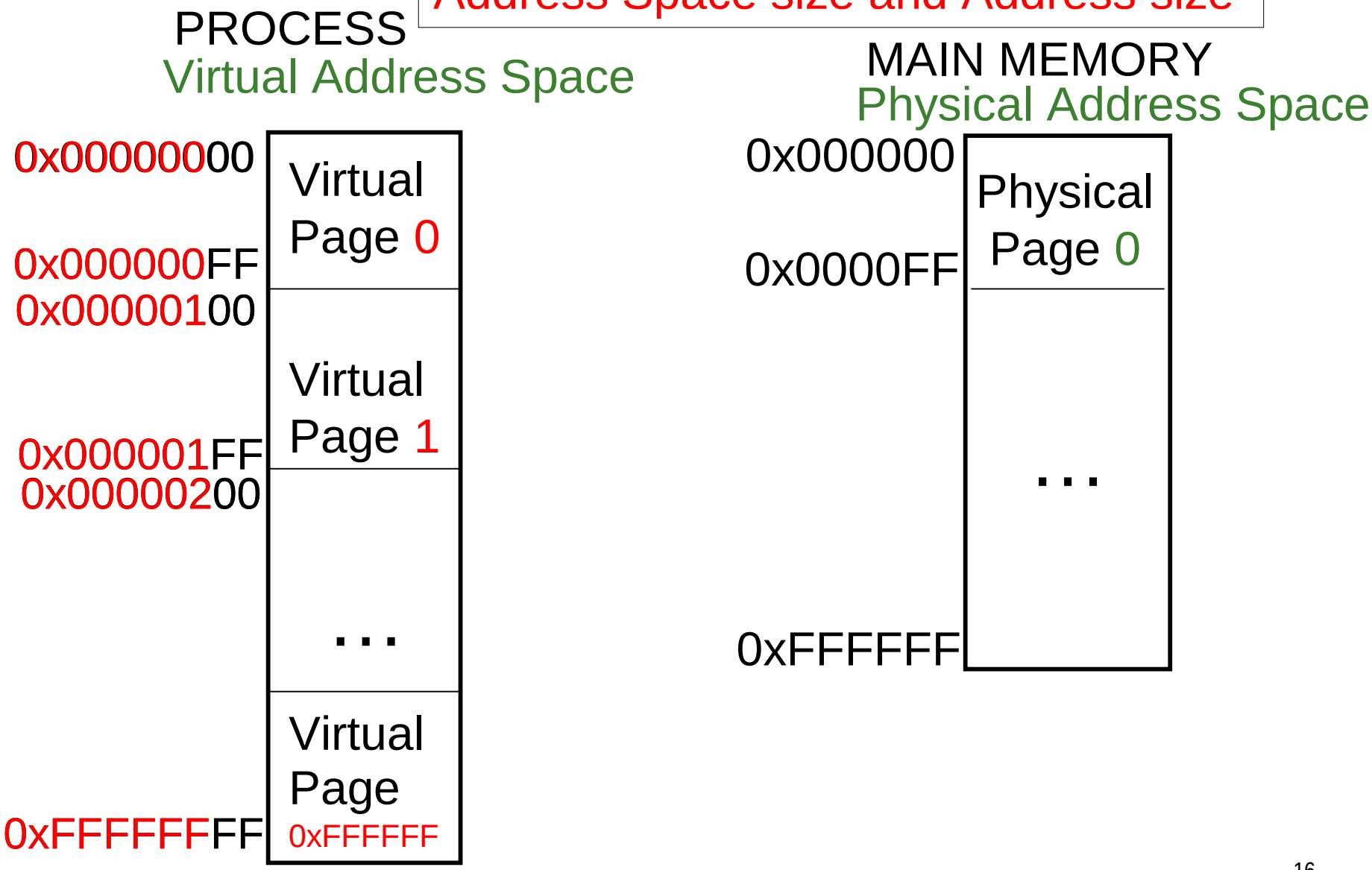
- Question: How big is a page in practice?

Page Size

- A tradeoff is involved here: the larger the page size
  - the smaller the page table, but
  - more potentially unused memory space within a page (internal fragmentation)
- The unit of transfer to hard disks is typically 512B (disk sector)
- A typical page size: 4KB

Recall: Address Space

Address Space size and Address size



Address Space Size, Address Size