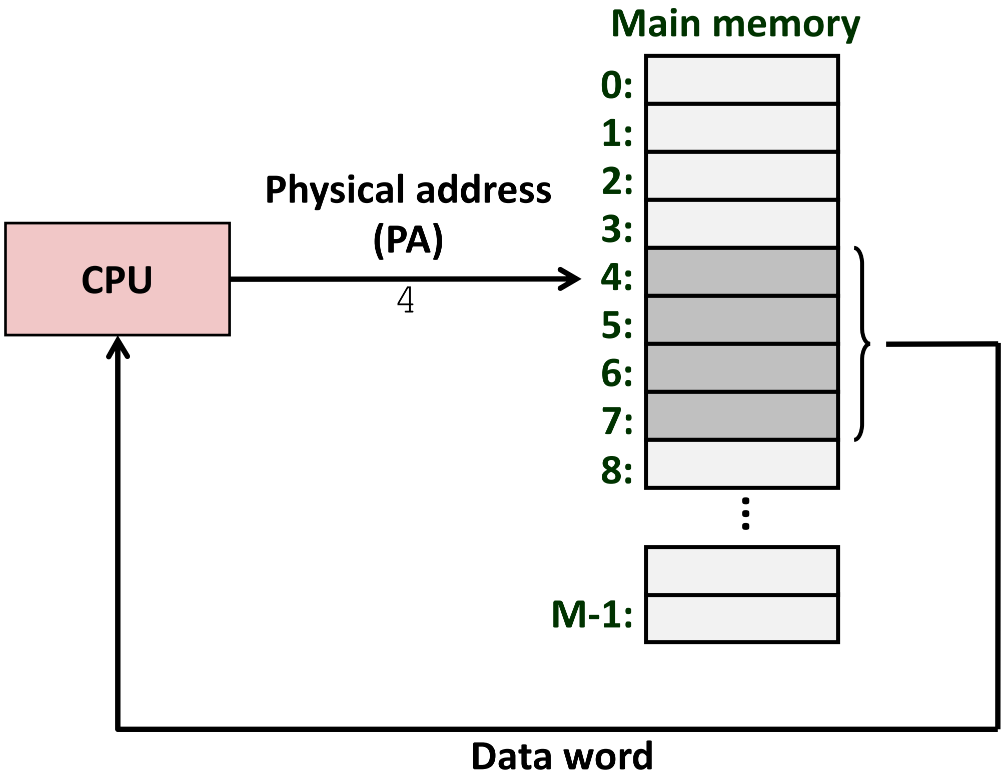
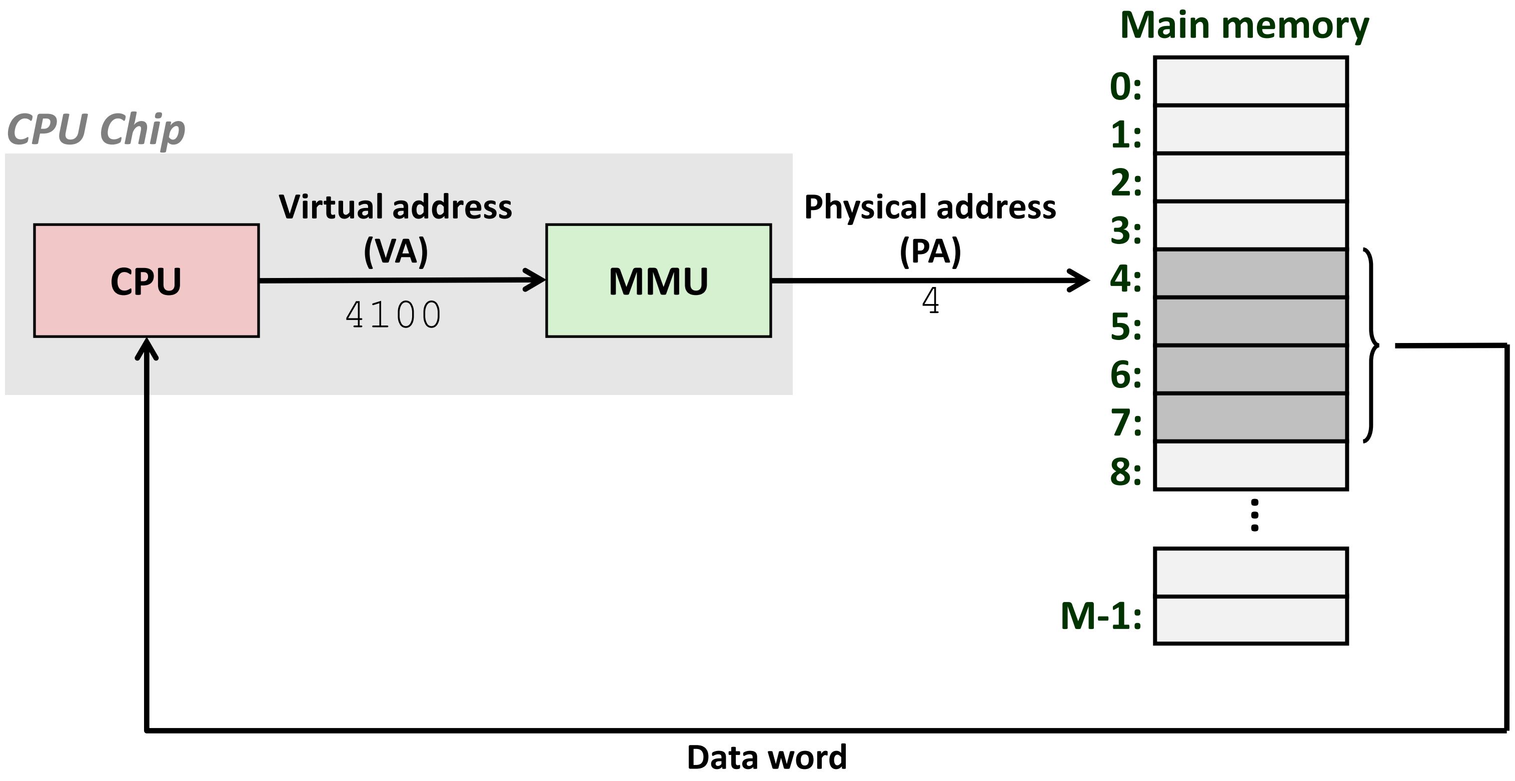
Virtual Memory: *Address spaces*

A System Using Physical Addressing



- Used in “simple” systems like embedded microcontrollers in devices like (older) cars, elevators, and digital picture frames
- Usually mono-process

A System Using Virtual Addressing



- Used in all modern servers, laptops, and smart phones
- One of the great ideas in computer science
- CPU is “oblivious” just like with HDD abstraction

Address Spaces

- **Linear address space:** Ordered set of contiguous non-negative integer addresses:

  {0, 1, 2, 3 … }

- **Virtual address space:** Set of N = 2^n virtual addresses (very large!)

  {0, 1, 2, 3, …, N-1}

- **Physical address space:** Set of M = 2^m physical addresses

  {0, 1, 2, 3, …, M-1}

Why Virtual Memory (VM)?

- **Uses main memory efficiently**
  - Use DRAM as a cache for parts of a virtual address space
  - DRAM much smaller than total virtual address space

- **Simplifies memory management**
  - Each process gets the same uniform *linear* address space
  - A process does not have to know what part of memory is theirs

- **Isolates address spaces**
  - One process can’t interfere with another’s memory
  - User program cannot access privileged kernel information and code