Computer System Organization Recitation [Fall 2017]

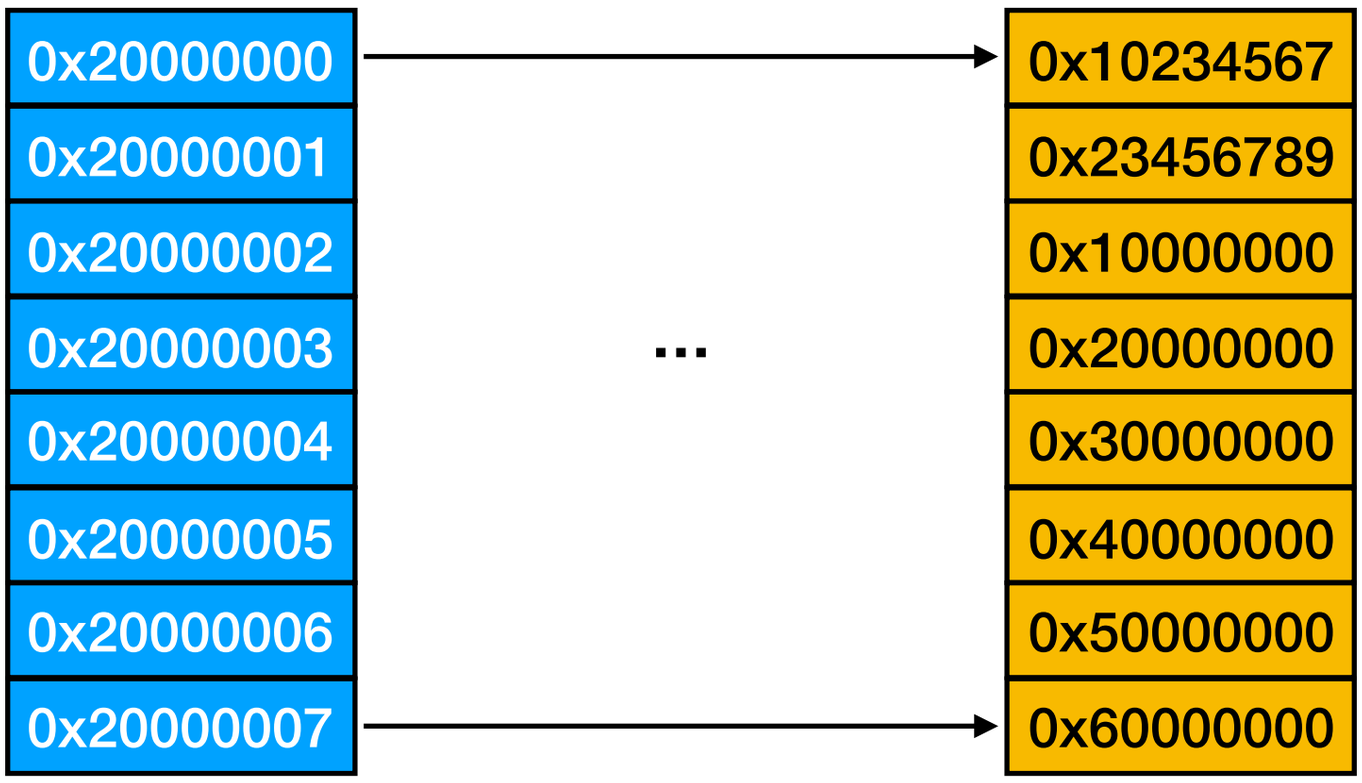
CSCI-UA 201-006

R7: Pagetable

How to translate from the virtual space to physical space

- Can we use a static mapping?
  - *paddr = f(vaddr)*
  - *e.g., f(x) = x + 0x210000, f(x) = (x & 0x10000000) + 100*
- What's the problem with this method?
  - Static is bad because a physical address is wasted if the corresponding virtual address is not used.
  - How do we separate the virtual address 0x10000 in process 1 and the virtual address 0x10000 in process 2?
    - Can encode the process id to *f(x).*
    - *Process id will cause another problem…*

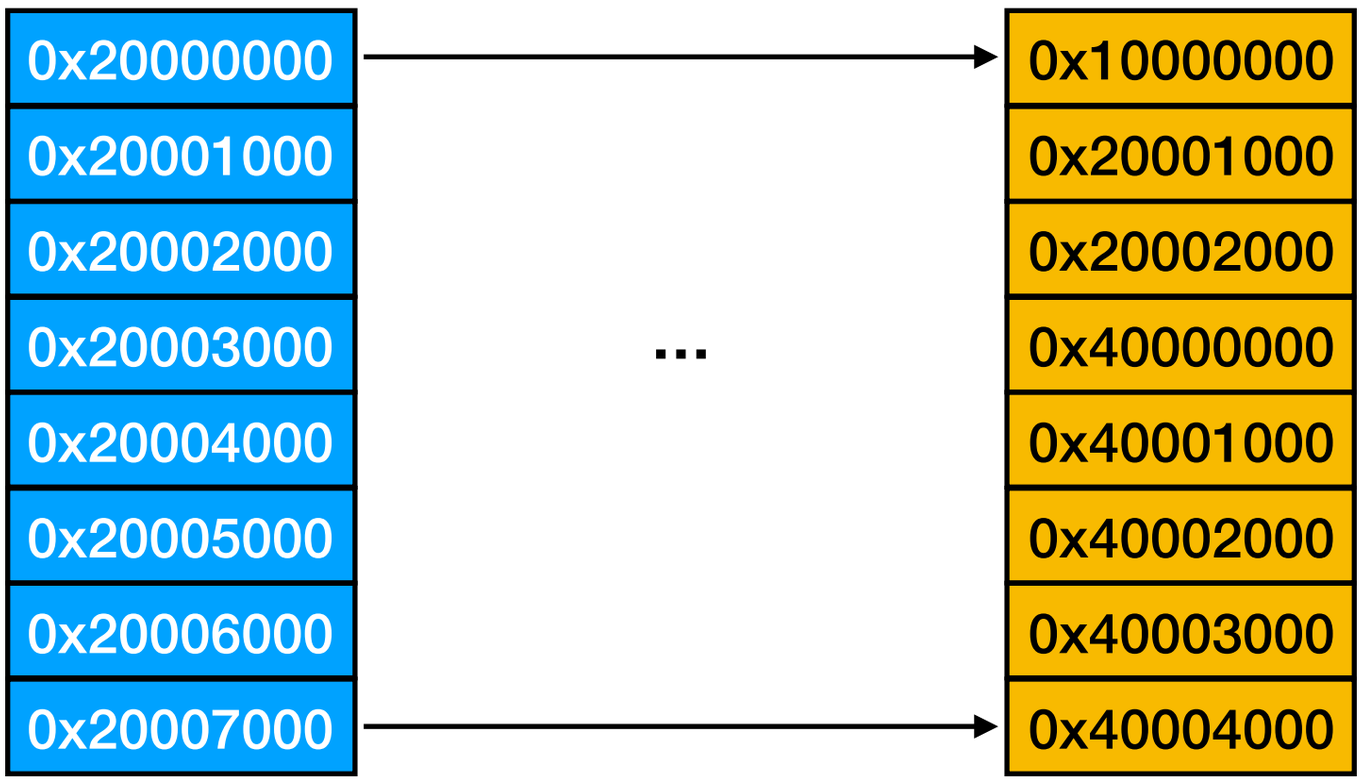
How about a table?



How about a table?

- What's the problem with this method?
  - How many bytes does a table need?
    - To map N bytes, 4N (on a 32-bit machine) or 8N (on a 64-bit machine) are needed.
    - Memory utilization N / (N + 8N) = 11.11%

How about a page table?



- How is 0x20000010 translated?
  - 0x20000010 - 0x20000000 + 0x10000000 = 0x10000010

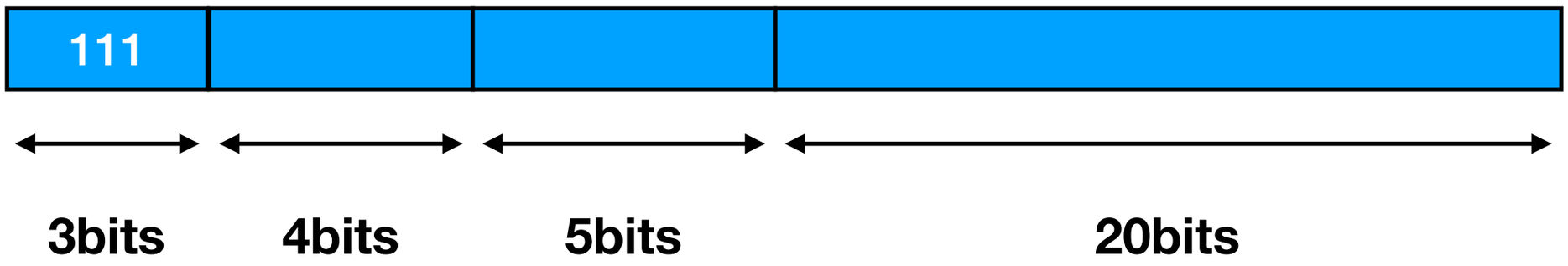
How about a page table?

- How many bytes does a table need?
  - Suppose the size of a page is P.
  - To map N bytes, N / P *4 (on a 32-bit machine) or N / P * 8 (on a 64-bit machine) are needed.
  - Memory utilization N / (N + 8N/P) = 1 / (1 + 8 / P)
  - If P = 4096, the memory utilization is 99.8% (around 8MB per 4GB).
  - It seems to be fine…,

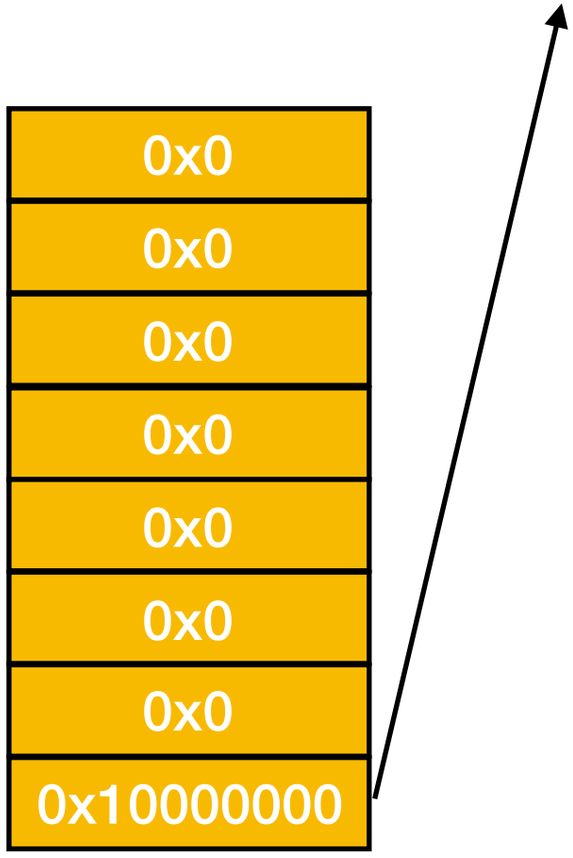
How many page tables do we need?

- Remember that each process has its own virtual memory space.
  - Each process needs a page table.
- NP * 8MB per 4GB
  - 200 * 8MB = 1.6GB

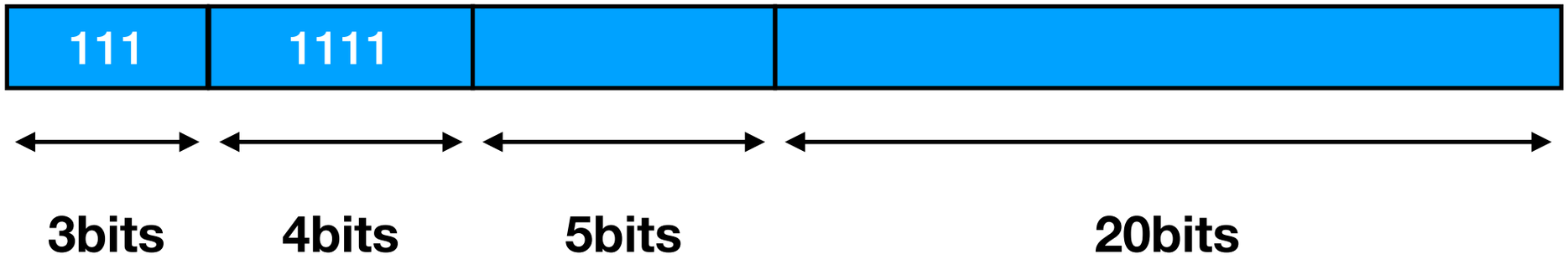
We need multi-level page tables



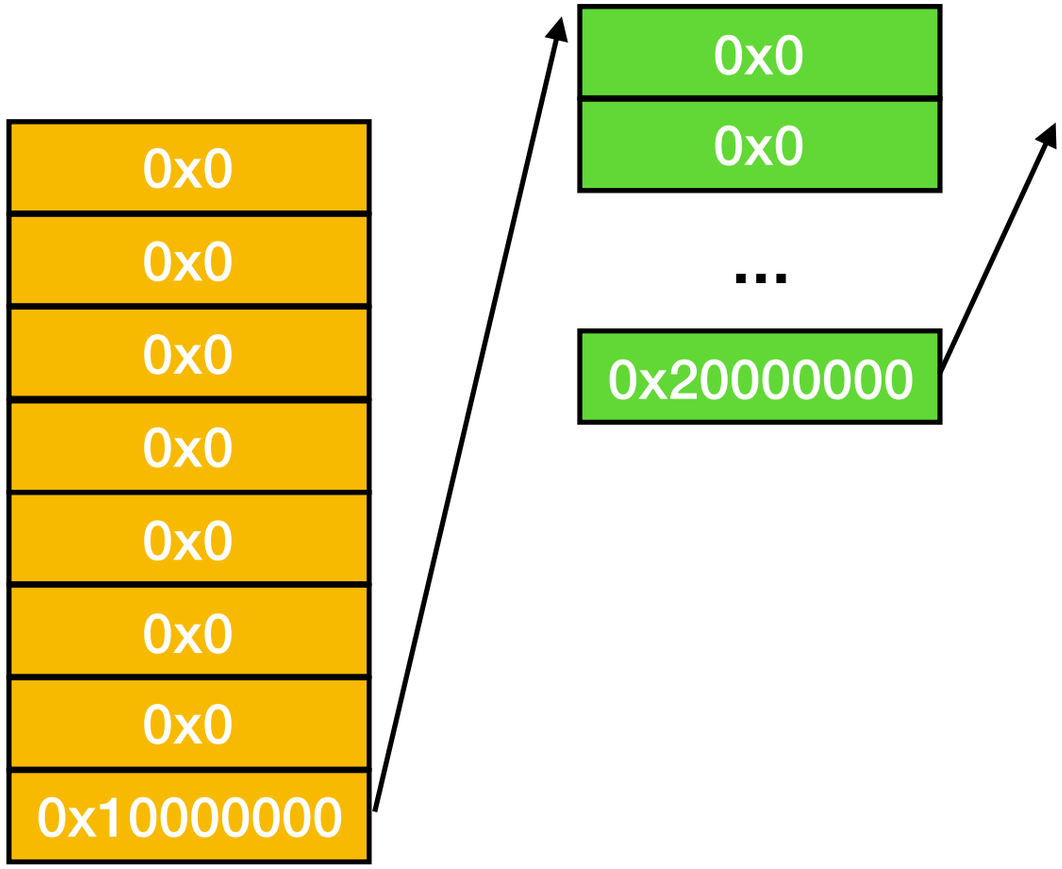
- In the recitation, we only implement a 32-bit MMU
  - 0xFFFF0001
  - b’111 = 7



We need multi-level page tables

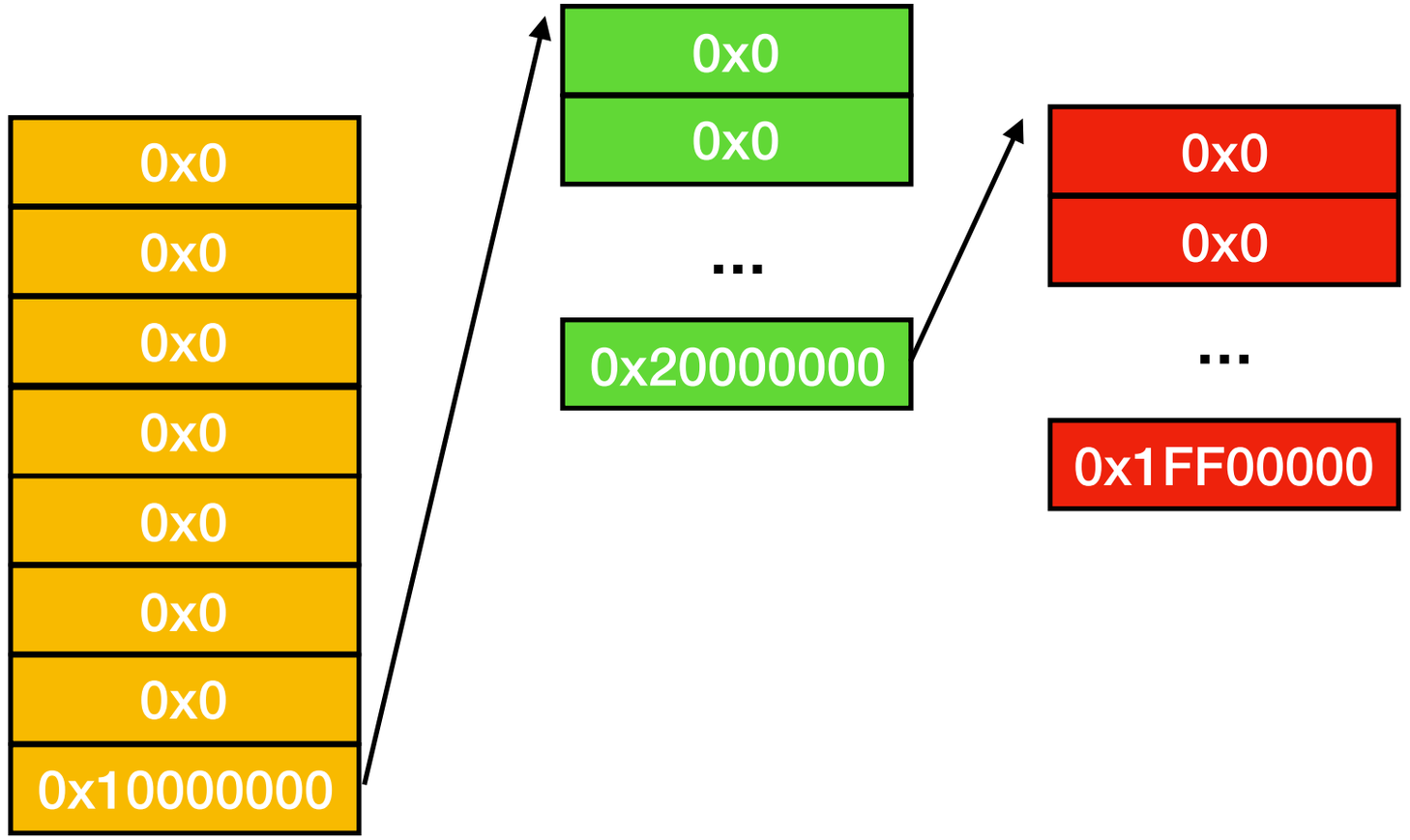


- In the recitation, we only implement a 32-bit MMU
  - 0xFFFF0001
  - b’111 = 7
  - b’1111 = 15



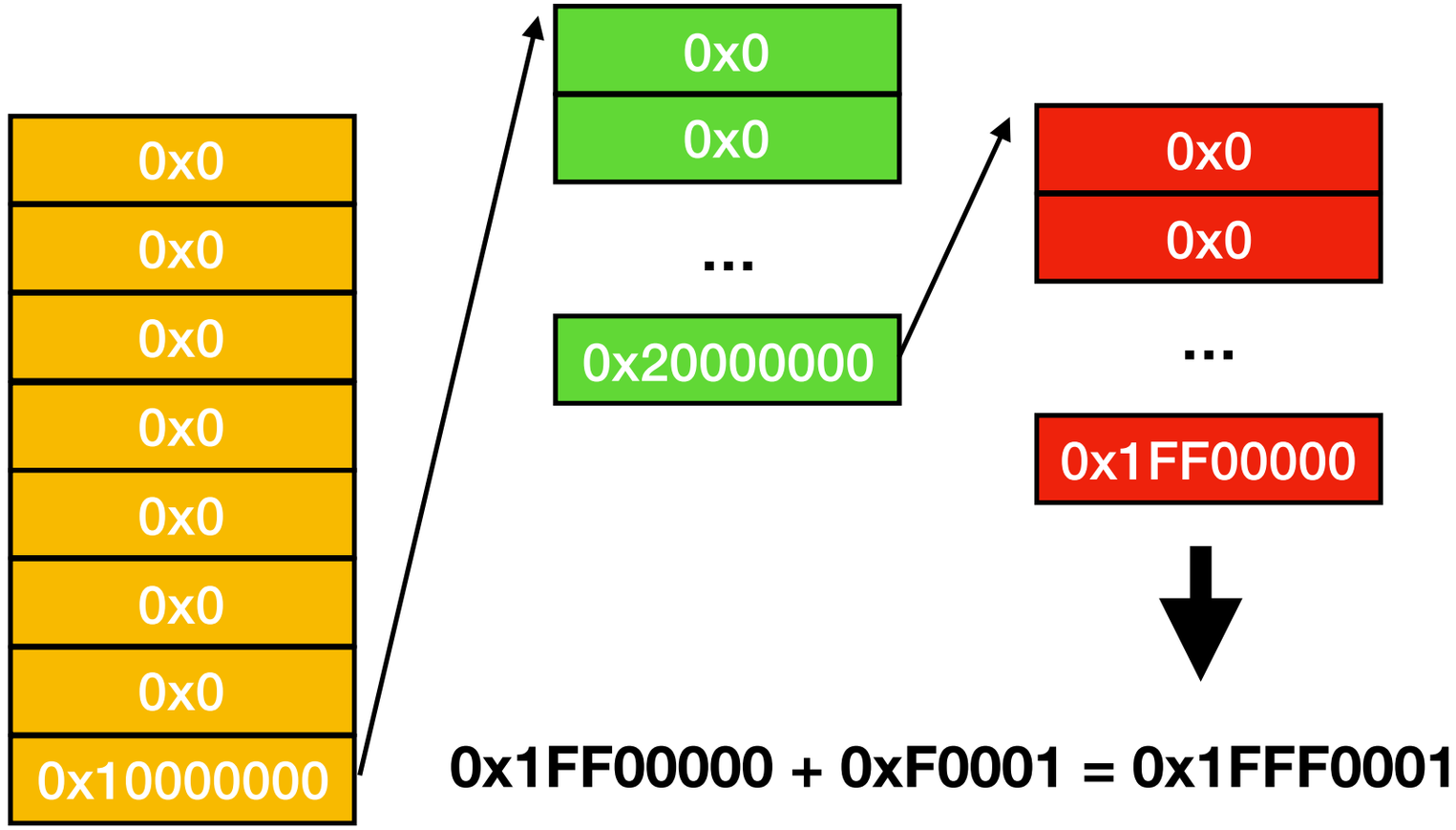
We need multi-level page tables

- In the recitation, we only implement a 32-bit MMU
  - 0xFFFF0001
  - b’111 = 7
  - b’1111 = 15
  - b’11111 = 31



We need multi-level page tables

- In the recitation, we only implement a 32-bit MMU
  - 0xFFFF0001
  - b’111 = 7
  - b’1111 = 15
  - b’11111 = 31
  - b’1110…01 = 0xF0001



More on…

- http://news.cs.nyu.edu/~zhaoguo/fa17-cso/notes/14-virtual_memory.pdf