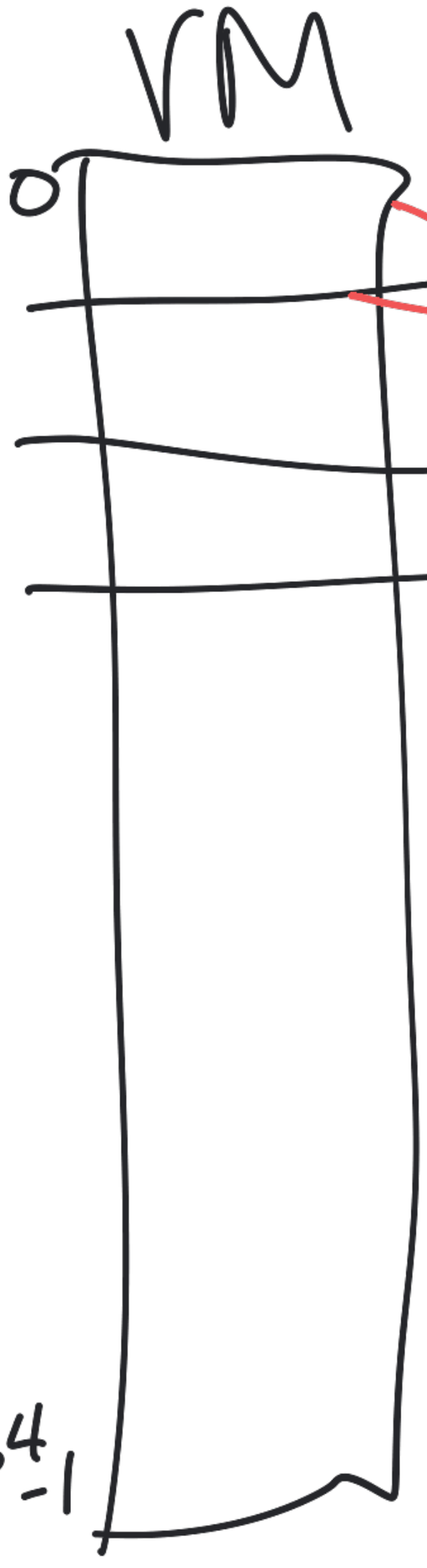
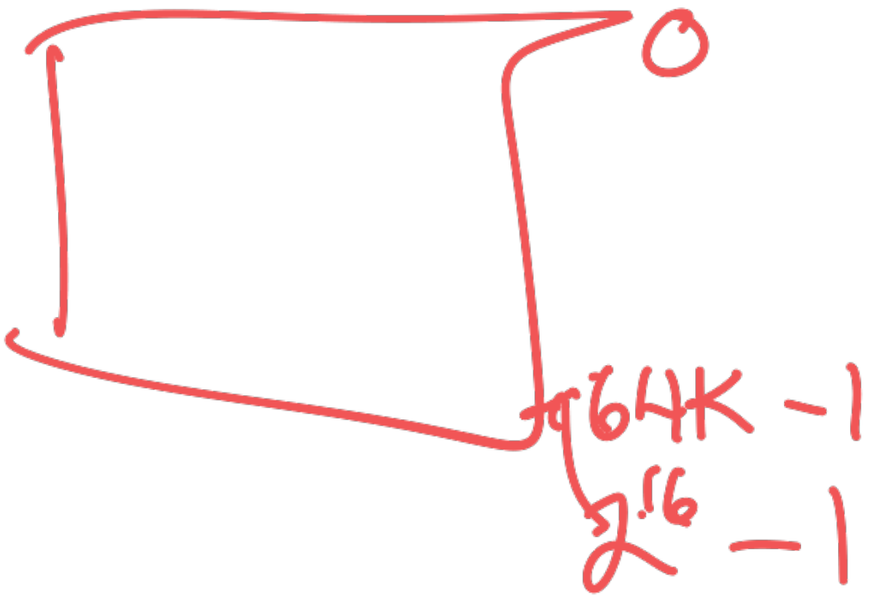
# CS208

# paged virtual memory

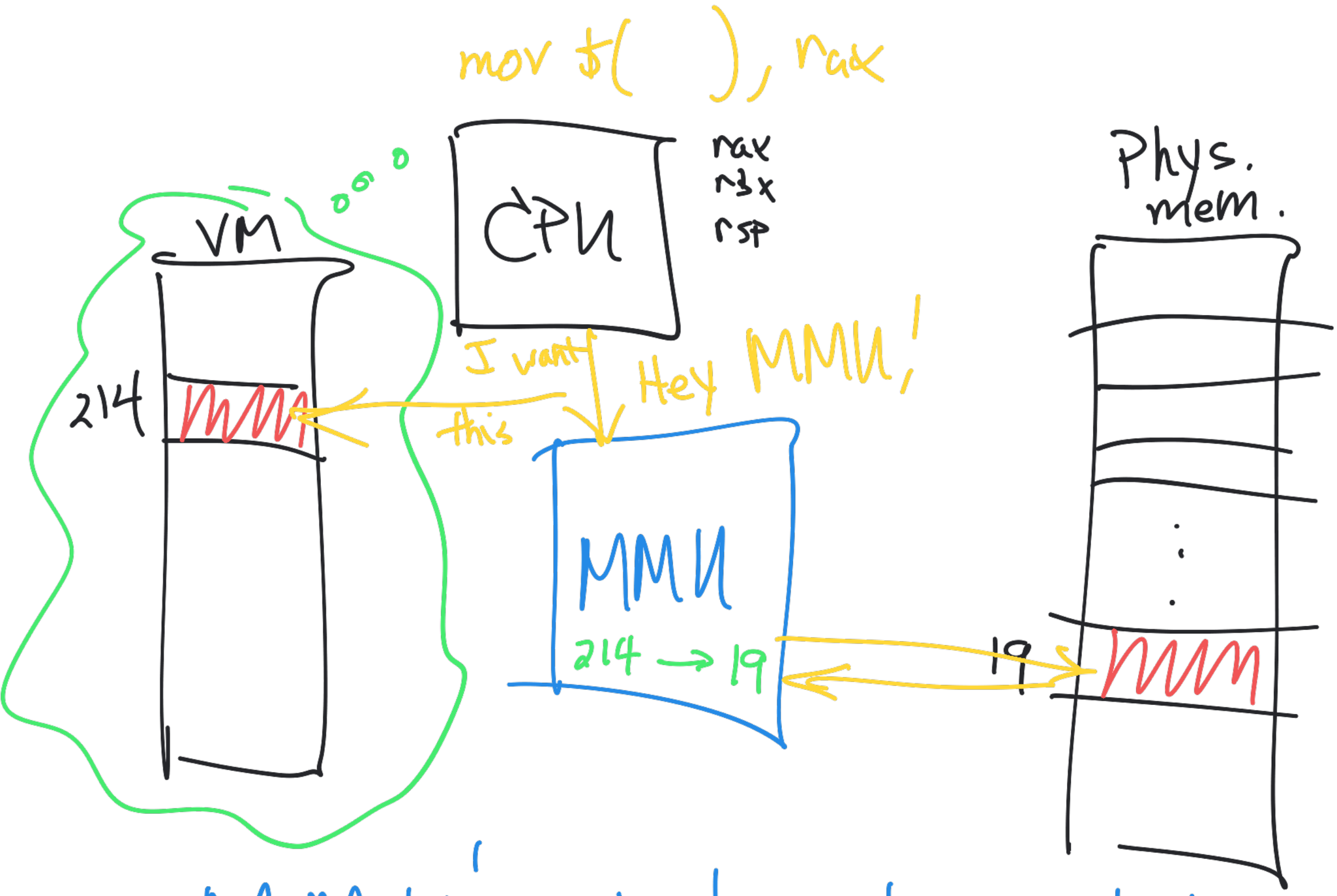
- Pages
- What's a page table for?
- how do page tables work?
  (splitting up addresses)
- multi-level page tables
- how much virtual space does a program really need?




How many virtual pages?

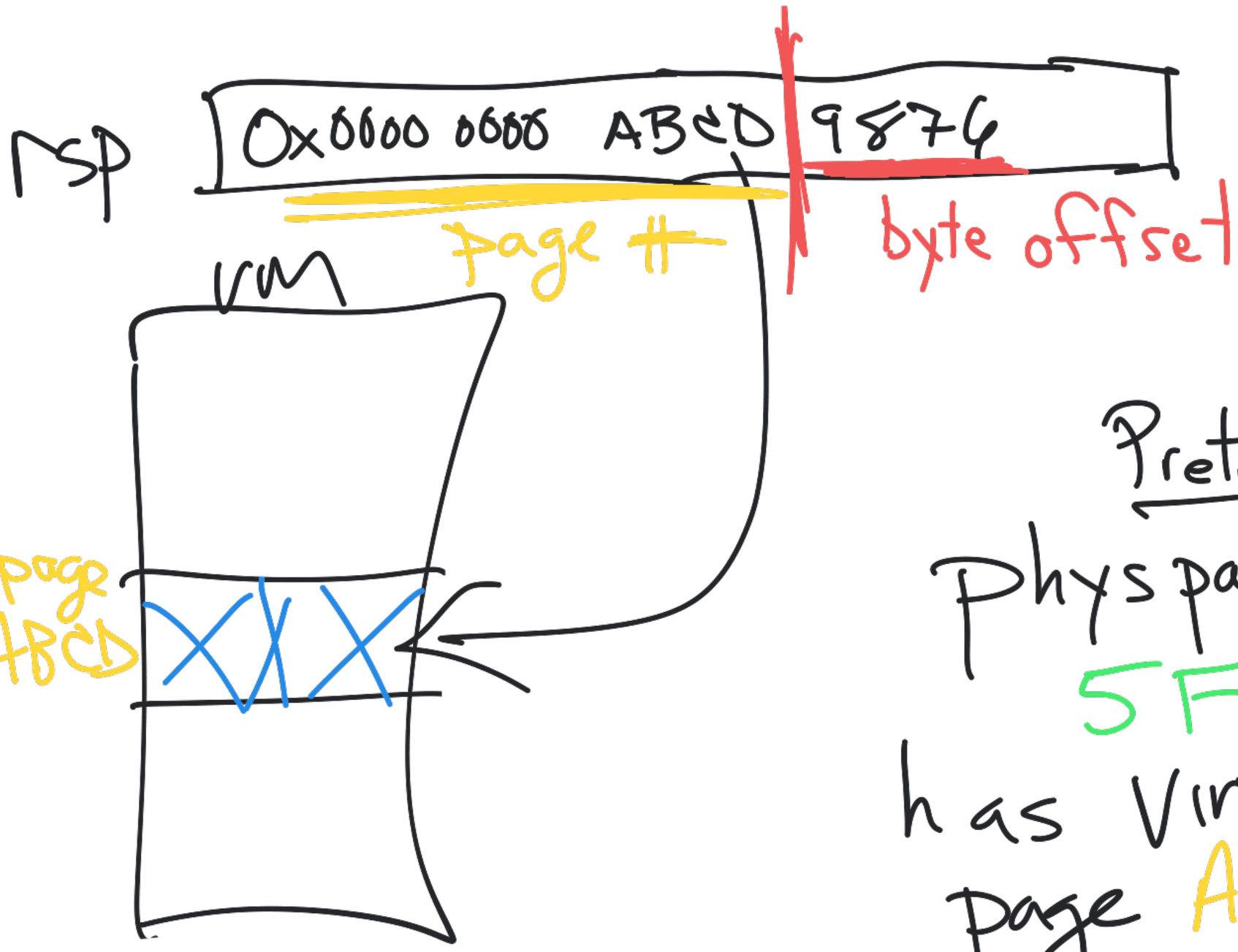
$$\frac{2^{64}\ \text{bytes}}{2^{16}\ \text{bytes/page}} = 2^{48}\ \text{pages}$$

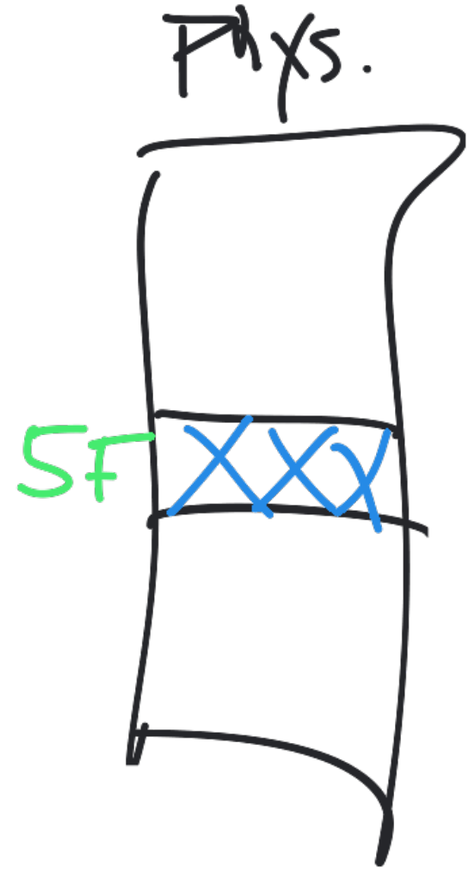



MMU's job: translate virtual pg# → physical pg#

Pretend page size = 64KB

= $2^{16}$ B

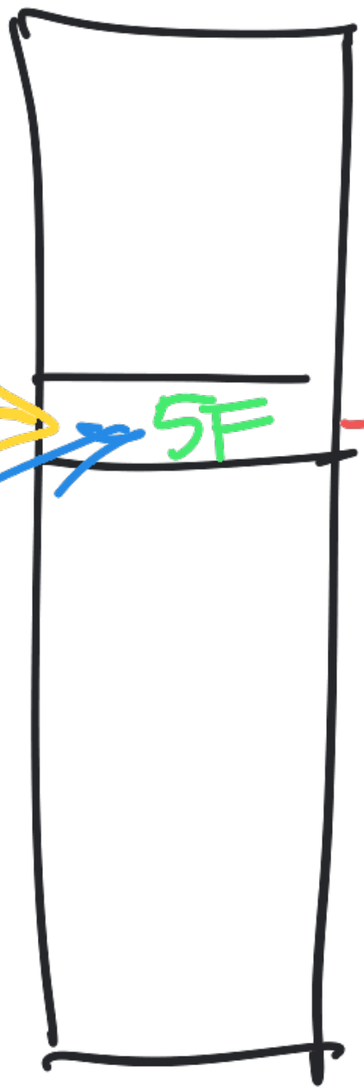



Pretend Phys page 5F has Virtual page ABCD

MMU needs to translate


Page table




page table entry

Page size = $2^{16}$

address

48 | 16

indexing an array

page table

pg # | flags

phys

page #

phys mem

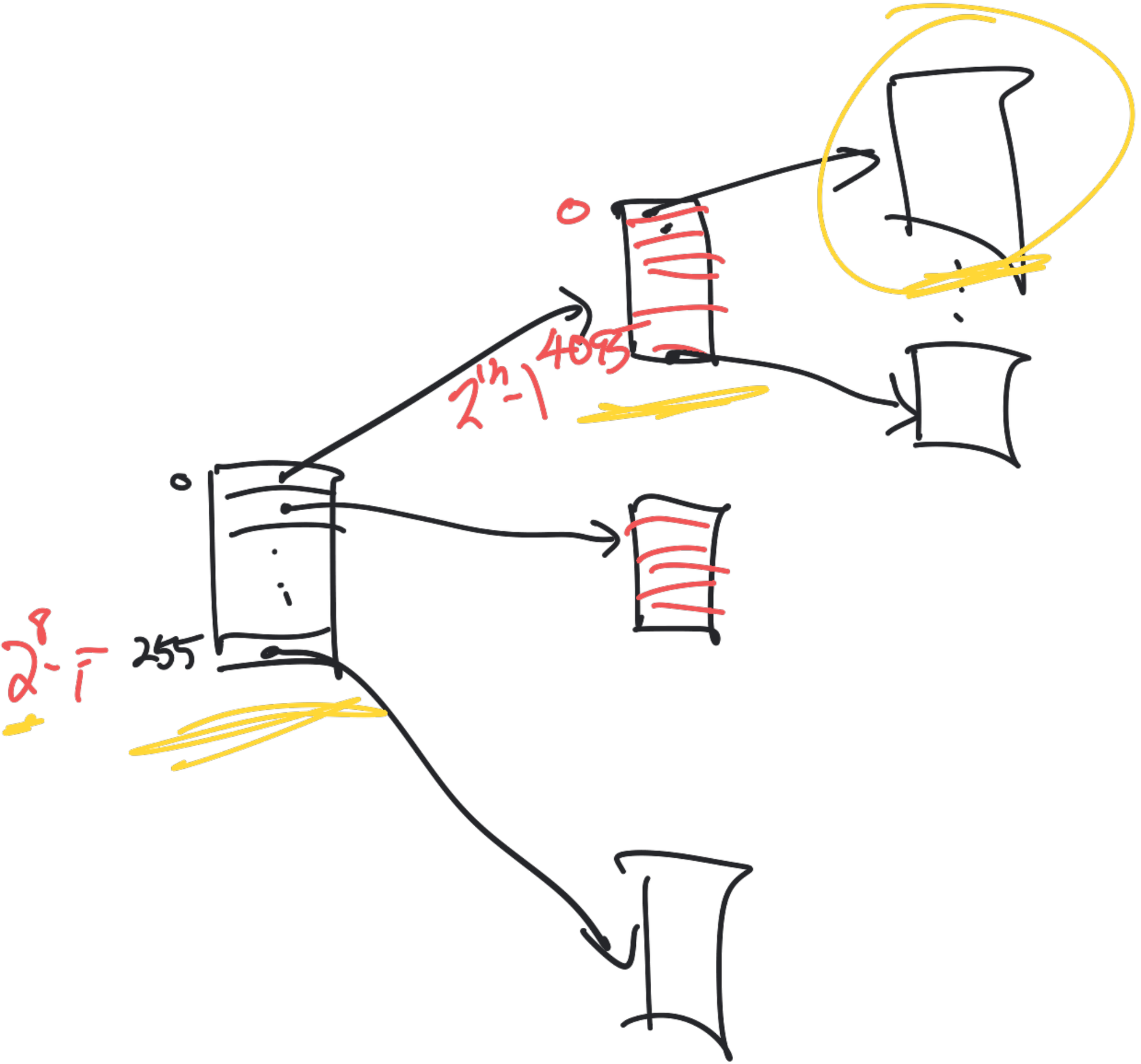
How big is the page table?

$2^{48}$ entries

Each entry

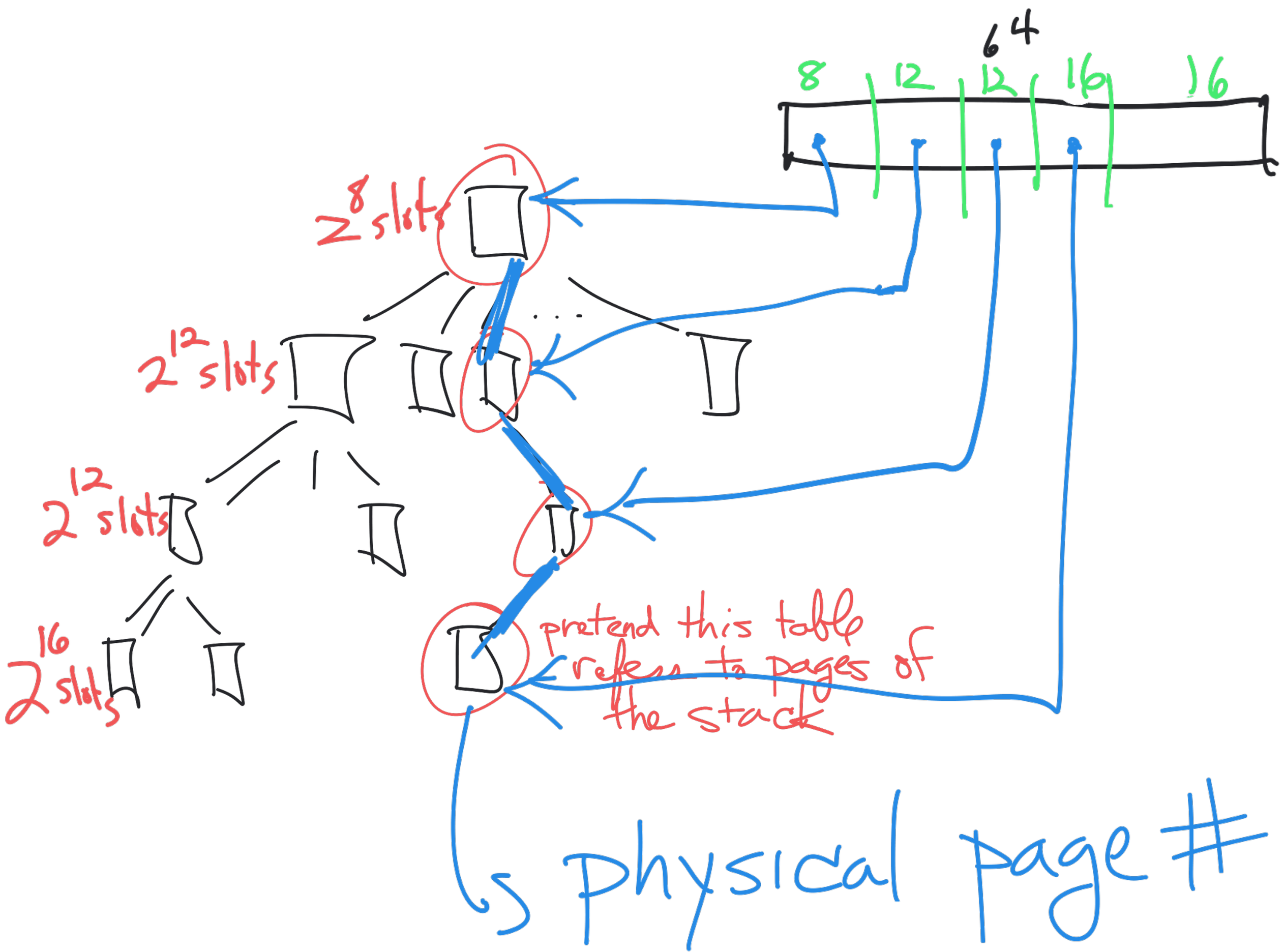
phys pg #

+ flags (Read-only, R/W, Shared, OS-only)




represents $\frac{1}{4096}$ of $\frac{1}{25\text{?}}$ of virtual memory

$\frac{1}{2^{20}}$


64
8
12
12
16
16
$2^8$ slots
$2^{12}$ slots
$2^{12}$ slots
$2^{16}$ slots
pretend this table refers to pages of the stack
physical page #

another approach

map virt pg # to phys pg #

} hash table

Exam point

We're not using much VM

64 KB page

addr

pg #

48

byte offset

16

0000 0000 0000 0000

⋮

1111 1111 1111 1111

4 hexadecimal digits

virtual memory is neato!