

Page Table

0	8A	8		5A
1	12	9		71
2	E7	10	A	F2
3	0B	11	B	14
4	13	12	C	10
5	67	13	D	95
6	D3	14	E	66
7	42	15	F	4E

2a. $0x \underbrace{6}_{\leftarrow \text{Frame}} \underbrace{A833}_{\leftarrow \text{offset}}$

$0x5A833$

1a. Logical addr

$0x \underbrace{04375}_{\substack{\text{page} \\ \#}} \underbrace{\hspace{1cm}}_{\text{offset}}$



$0x \underbrace{8A, 4375}_{\text{physical addr.}}$

~~Page~~

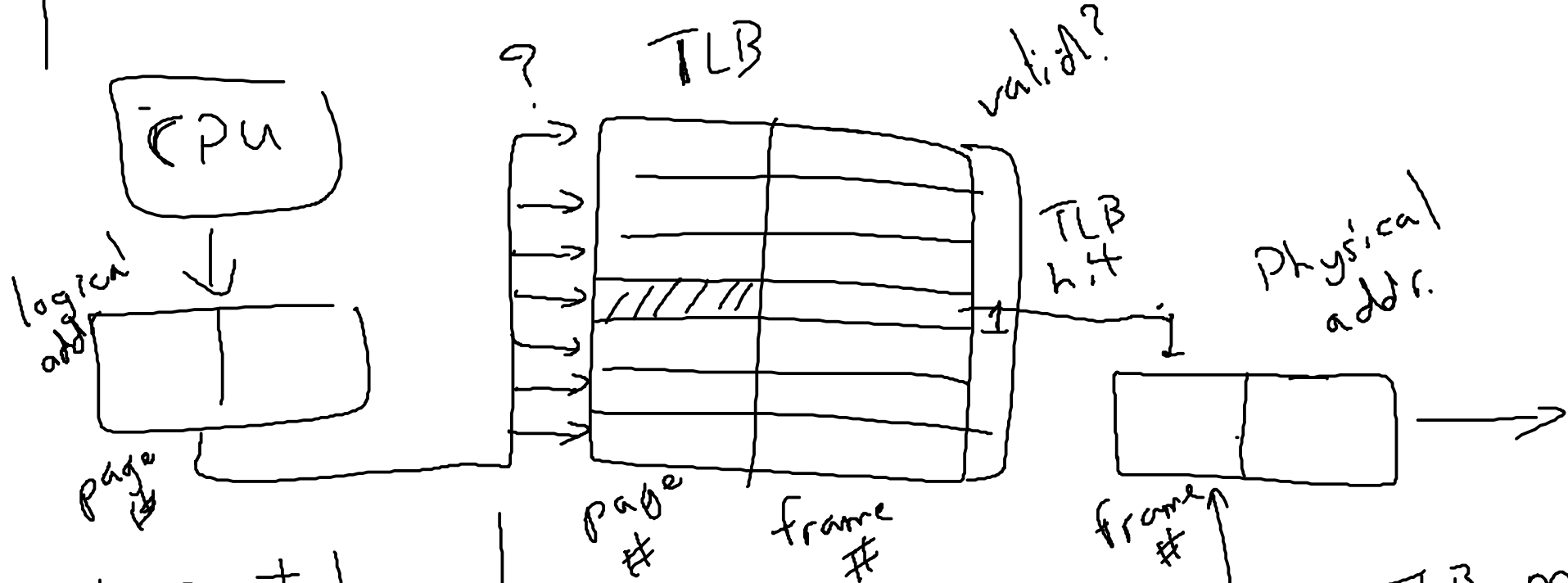
Page Table in Memory cache?

Cache issues

- mem. addr are hashed
- content addressable memory
(parallel requests)

TLB: Translation Lookaside Buffer

TLB:



Context Switch

① flush TLB

② include proc. id

TLB miss Page table



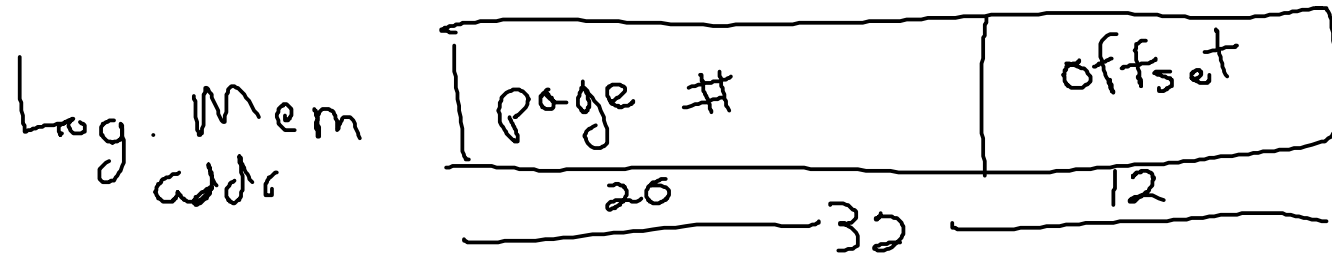
TLB miss:
TLB gets the
"missed"
entry

Page Sizes

Logical Mem: 4 GB 4×2^{30} - 32 bit addr.

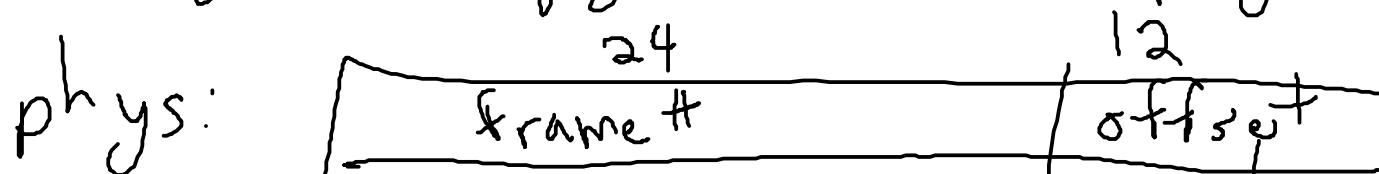
Page Frame Size: 4 KB 4×2^{10} - 12 bit offset

Phys. Mem: 64 GB $64 \times 2^{30} = 2^6 \times 2^{30} = 2^{36}$
36 bits



20 bit pg # - 2^{20} pages

2^{24} frames



Page Table

2^{20}

pages

2^{20} entries

each entry: 24 bit frame#

8 bit status < valid

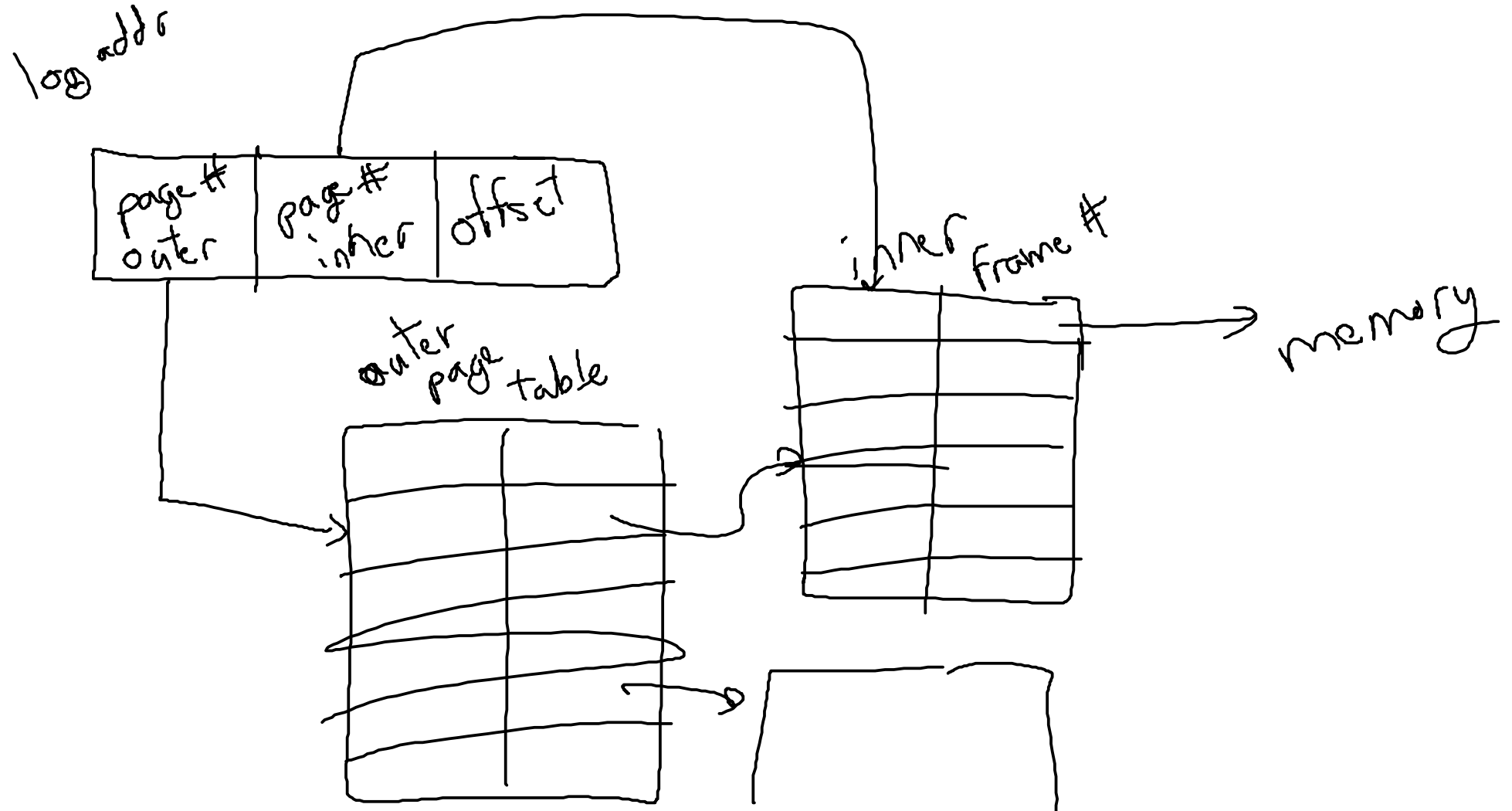
rwX

32 bits = 4 Bytes

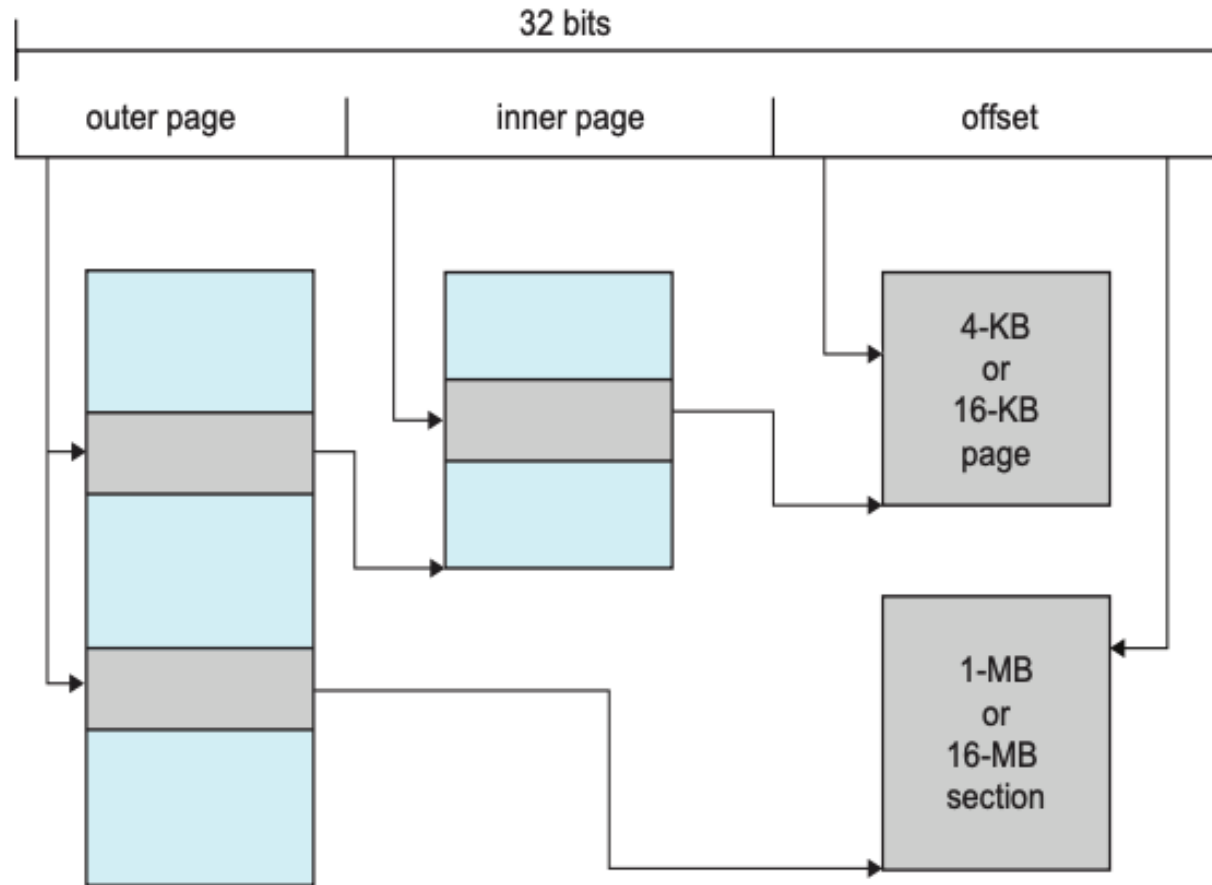
P.T. Size

$$4 \times 2^{20} = 4MB$$

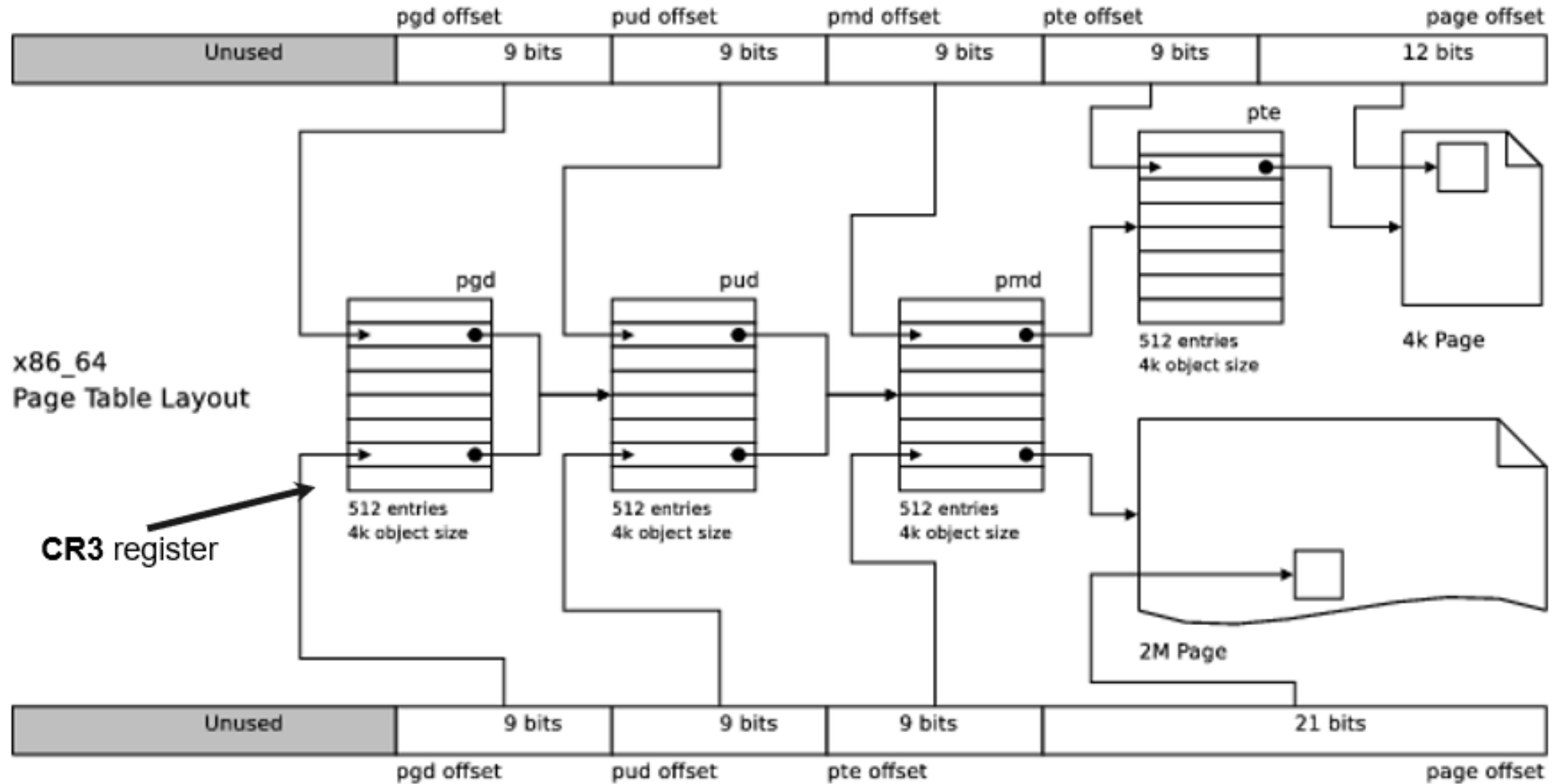
Two level paging



ARM Architecture

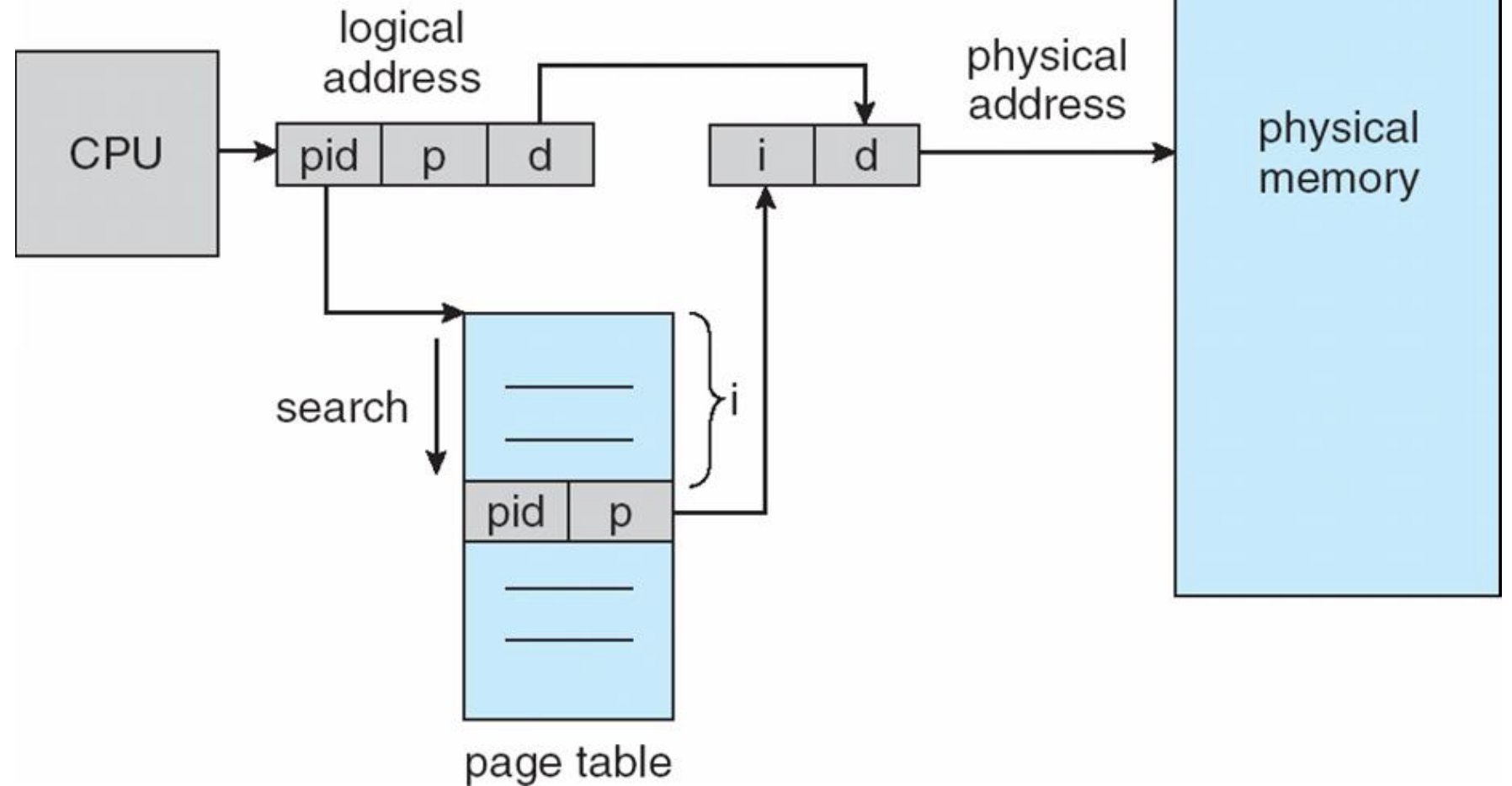


Four-Level Page Tables (x86-64)



PGD: Page Global Directory, **PUD:** Page Upper Directory **PMD:** Page Mid-level Directory, **PTE:** page table entry

Inverted Page Tables



Segmentation with Paging

