cs5965 Advanced OS Implementation

Lecture 03 – Boot-Time Page Table

Anton Burtsev

Booting into 64bit Rust code

- Well, on x86 you can’t just longjump into 64bit mode
  - Paging is required
- The *absolute minimum* to get your 64-bit code running is:
  - Enable PAE in CR4
  - Load CR3 with a PML4
    - Identity mapping is a good idea
  - Set Long-Mode Extension (LME) in IA32_EFER
  - Enable paging (CR0.PG=1)
  - Far jump into a 64-bit code segment

x86 paging

- X86 64bit mode supports two page table organizations
  - 4-level
  - 5-level

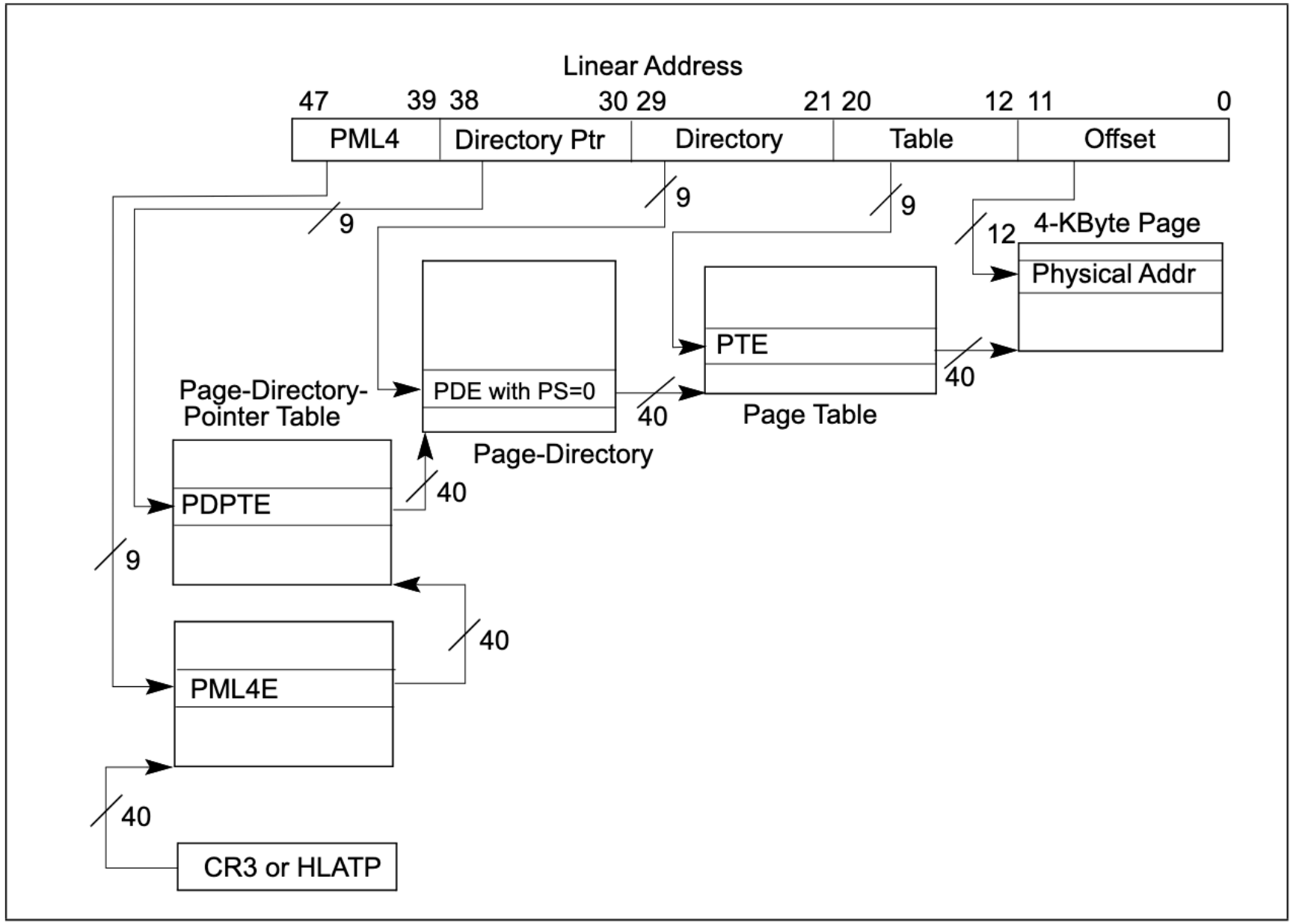

Figure 5-8. Linear-Address Translation to a 4-KByte Page Using 4-Level Paging

- 4-levels
- 4KB pages
- We use bits 0 through 47 in the virtual address, for a total of 256 TiB

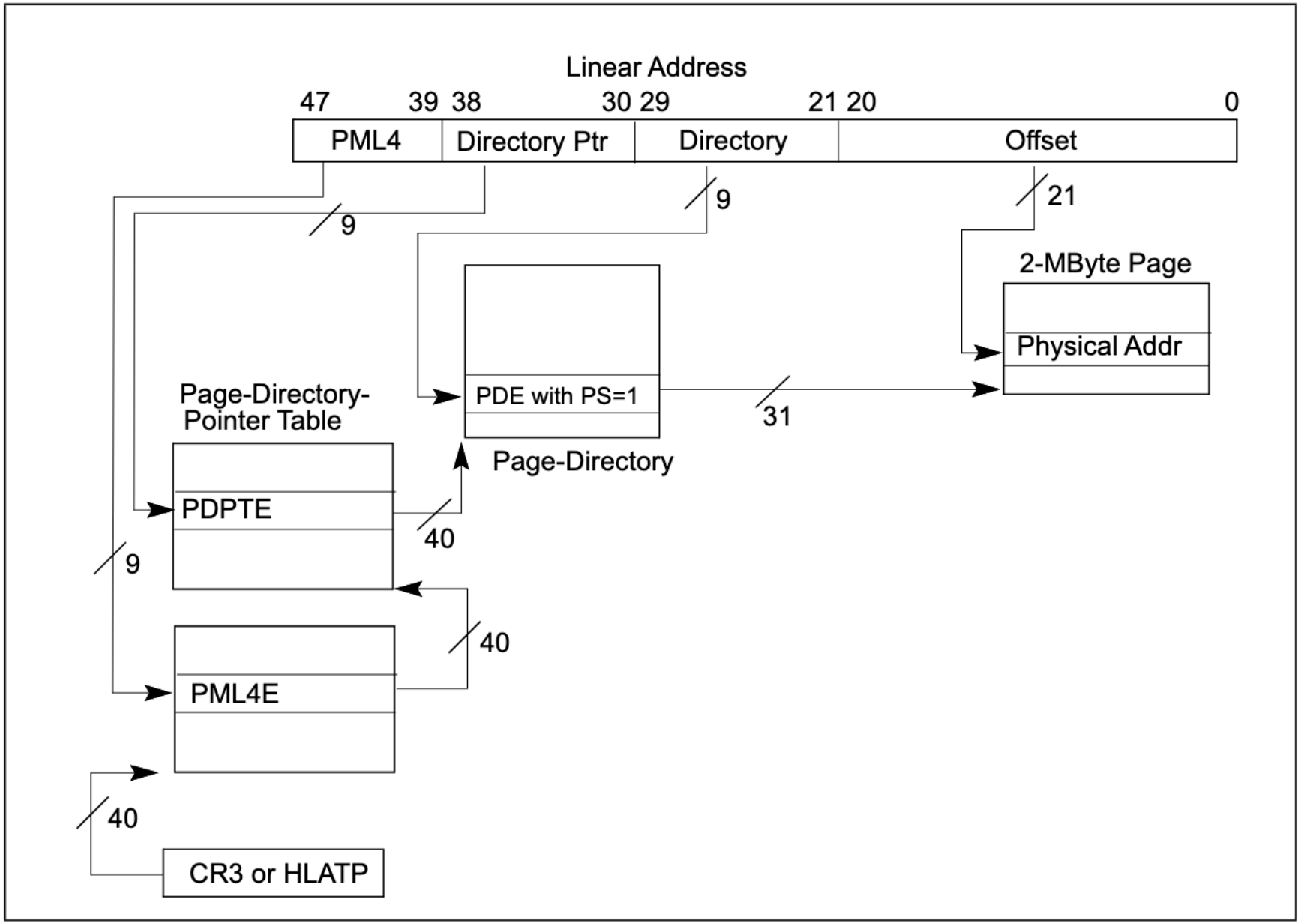

Figure 5-9. Linear-Address Translation to a 2-MByte Page using 4-Level Paging

- 4-levels
- 2MB pages

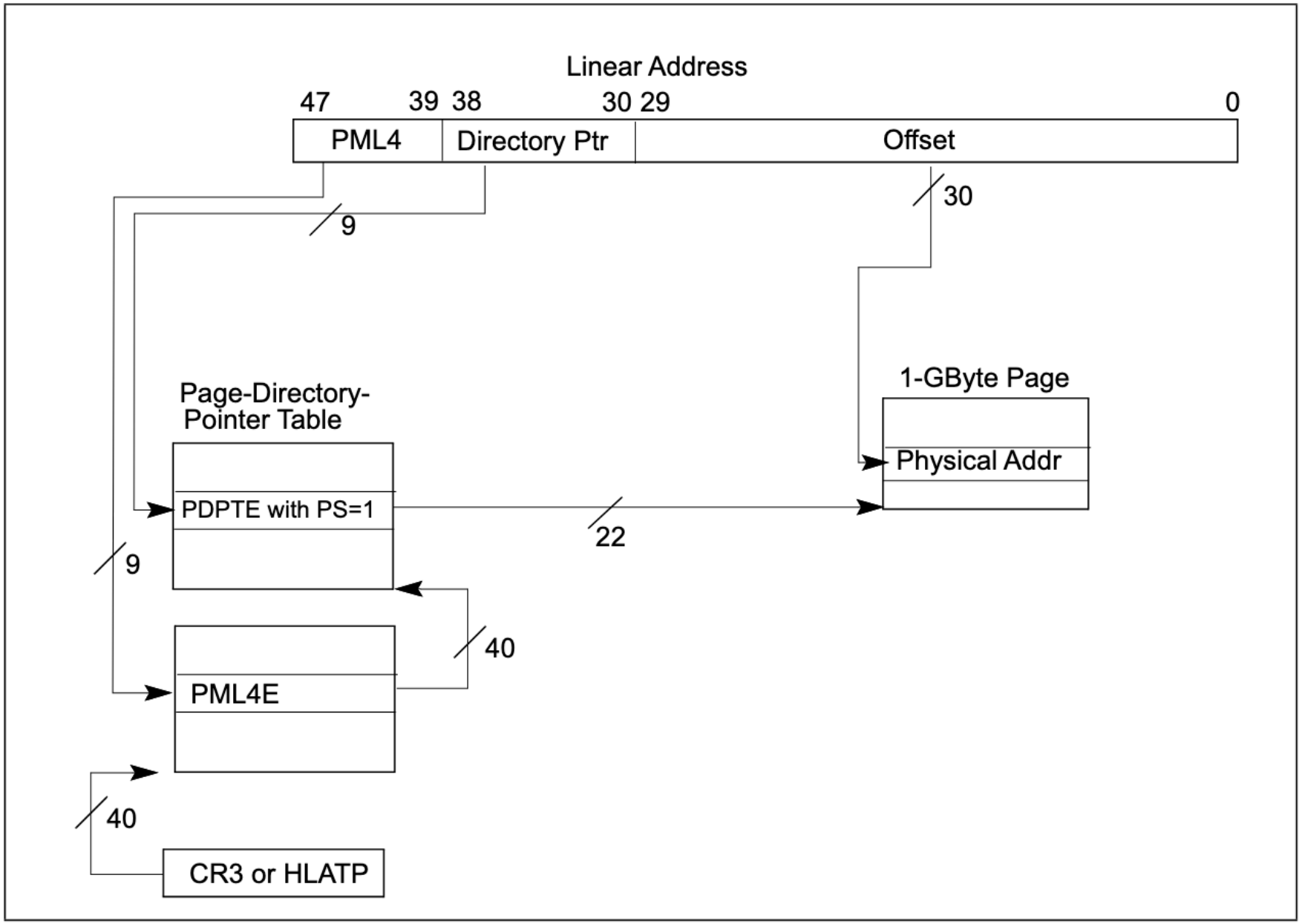

Figure 5-10. Linear-Address Translation to a 1-GByte Page using 4-Level Paging

- 4-levels
- 1GB pages

5-level page table

- 5-level paging adds another 9 bit page table descriptor, making it possible to use bits 0 through 56
- This multiplies the address space by 512 and increases the limit to 128 PiB.

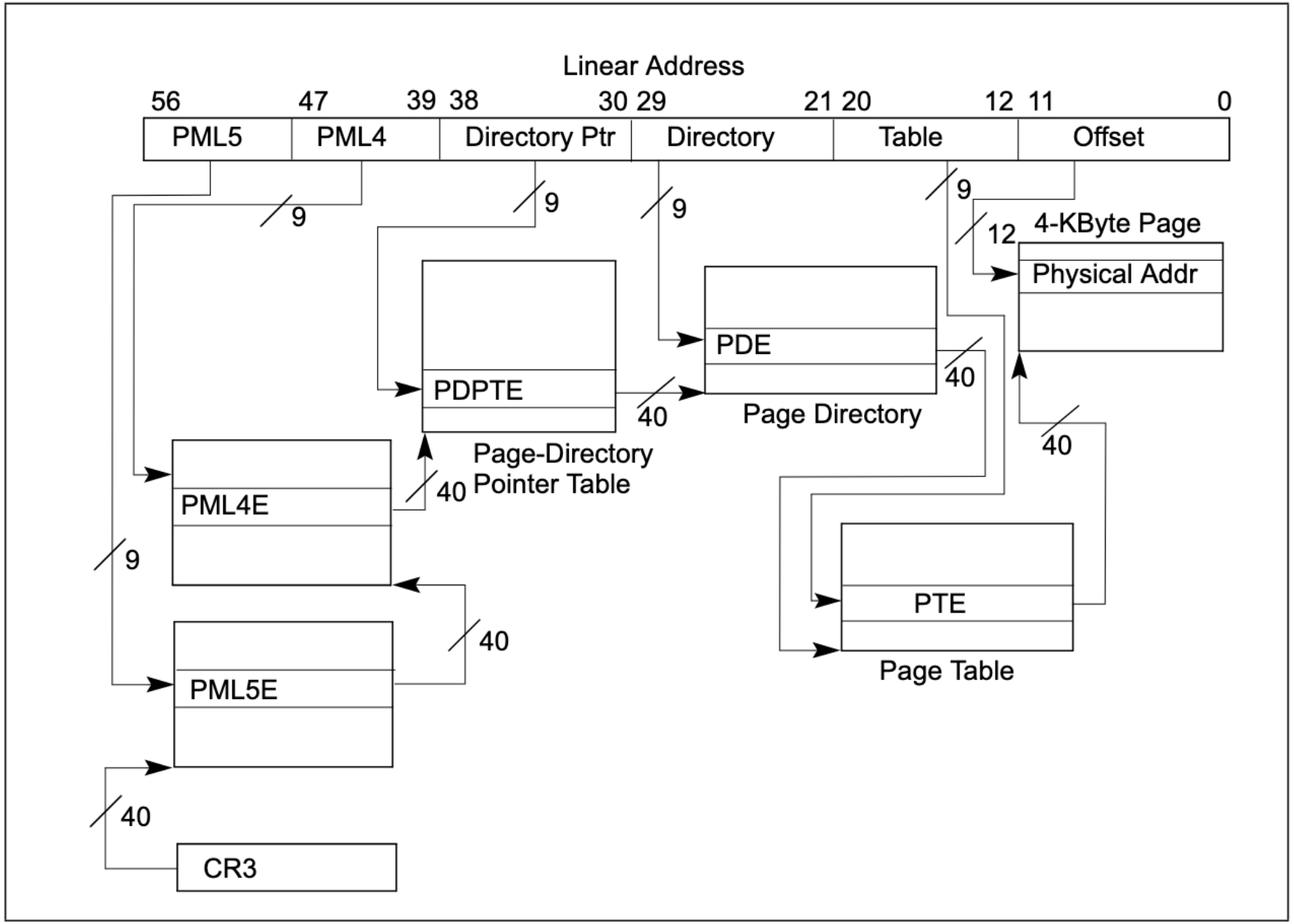

Figure 2-1. Linear-Address Translation Using 5-Level Paging

- 5-levels
- 4KB pages

Thank you!