Memory Architecture IV

CS2053 Computer Architecture

Computer Science & Engineering

University of Moratuwa

Sulochana Sooriyaarachchi

Chathuranga Hettiarachchi

Outline

- Memory types
- Memory access
- Memory hierarchy
  - Main memory
  - Cache
  - Permanent storage
- Example architecture RV32I
  - Superscalar architecture
  - Pipelining
- Virtual memory

Virtual Memory

Virtual Memory

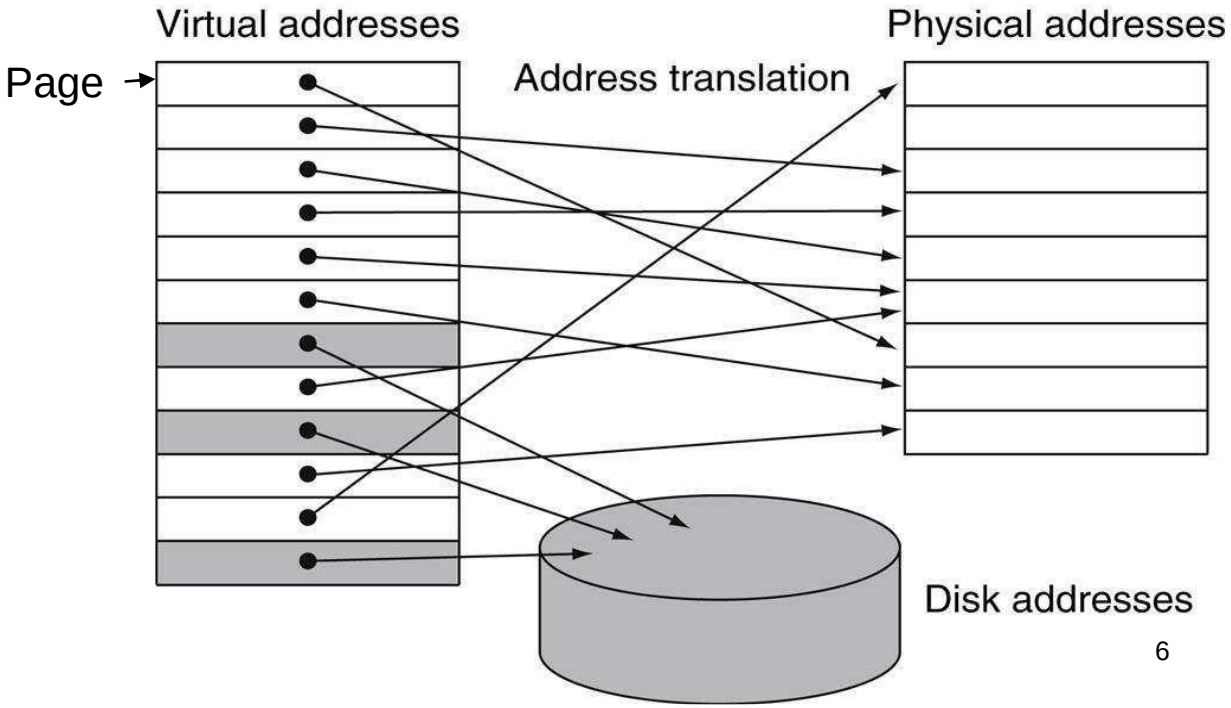
- Main memory act as “cache” for secondary storage
- Motivation
  - To share memory among several programs
  - To relieve the programmer from limited memory constraint
- Protection needed to avoid read/ write to memory portions of other programs/ virtual machines
  - Programs have own address space

Program's address space to physical address mapping

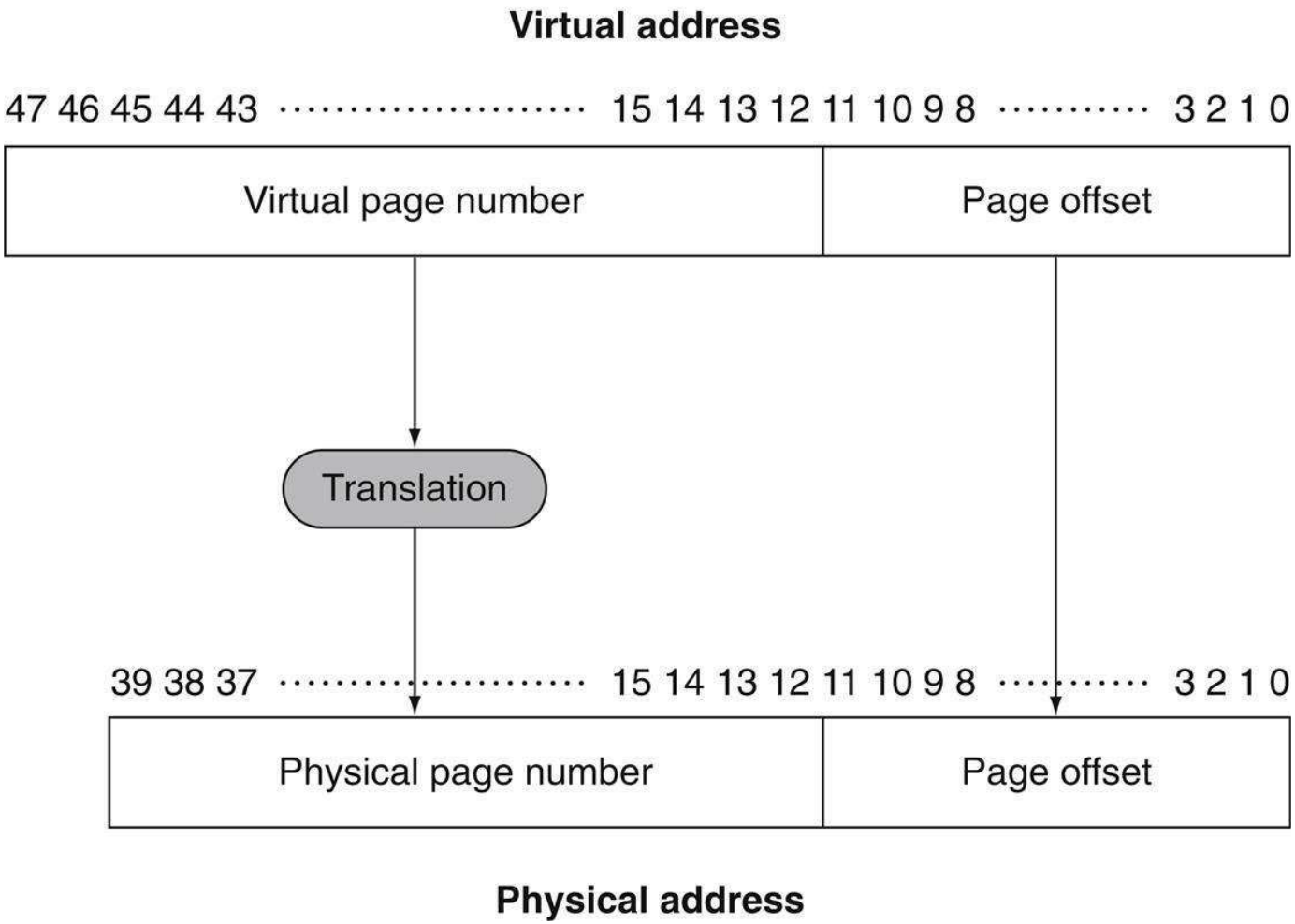
- Physical address: an address is main memory
- Protection: a set of *mechanisms* to ensure that
  - multiple processes *sharing*
  - the processor, memory, or I/O devices
  - cannot interfere intentionally/ unintentionally with one another
  - by reading/ writing to each other's data
  - Protection also isolates OS from user processes
- Operation of cache and virtual memory have similarities, but terminology is different

Virtual memory terminology

- Main memory = physical memory
- Secondary storage: e.g. magnetic disks
- Virtual memory blocks = *pages*
- A page not in main memory = *page fault*

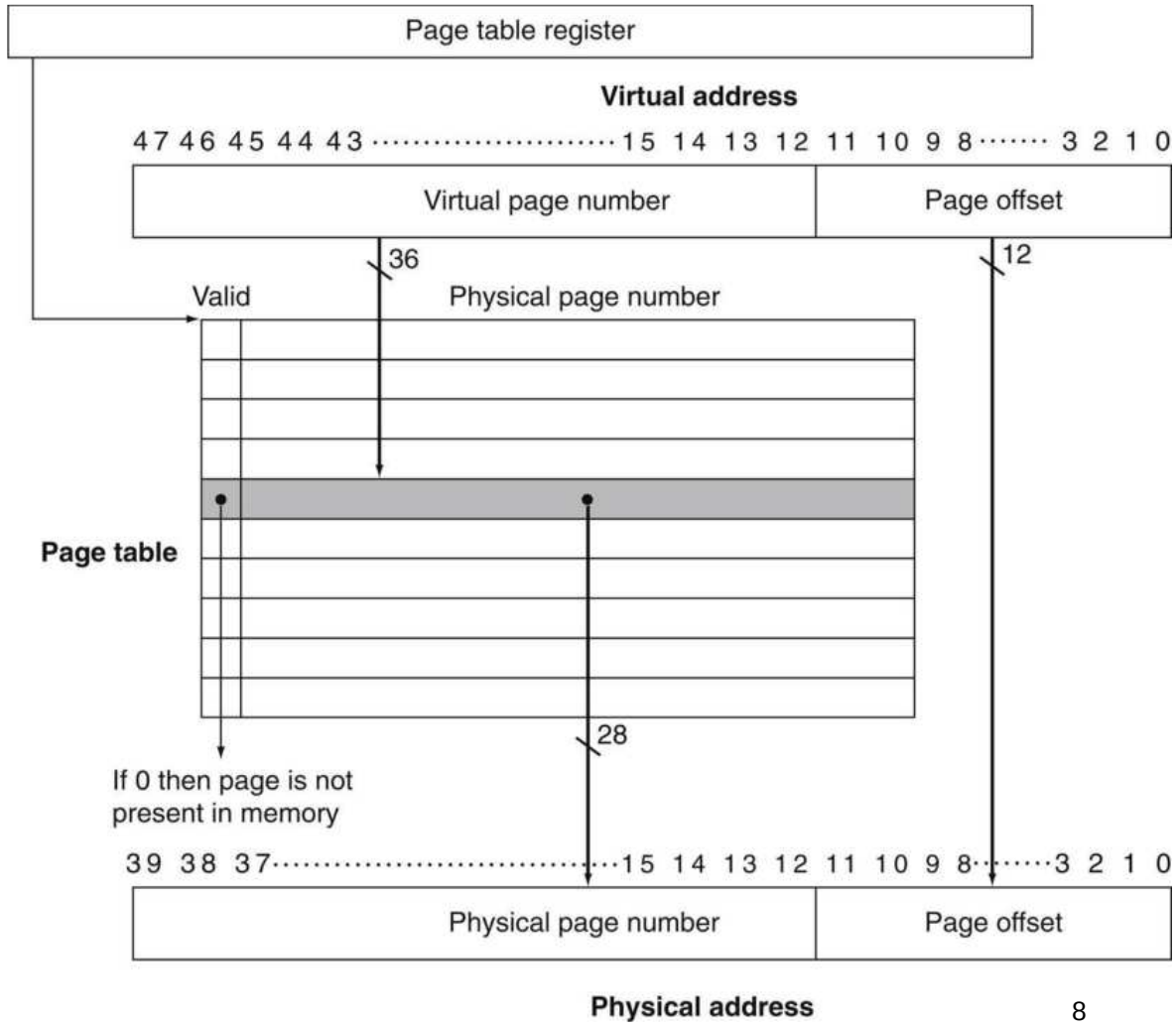


Address translation



Virtual memory configuration

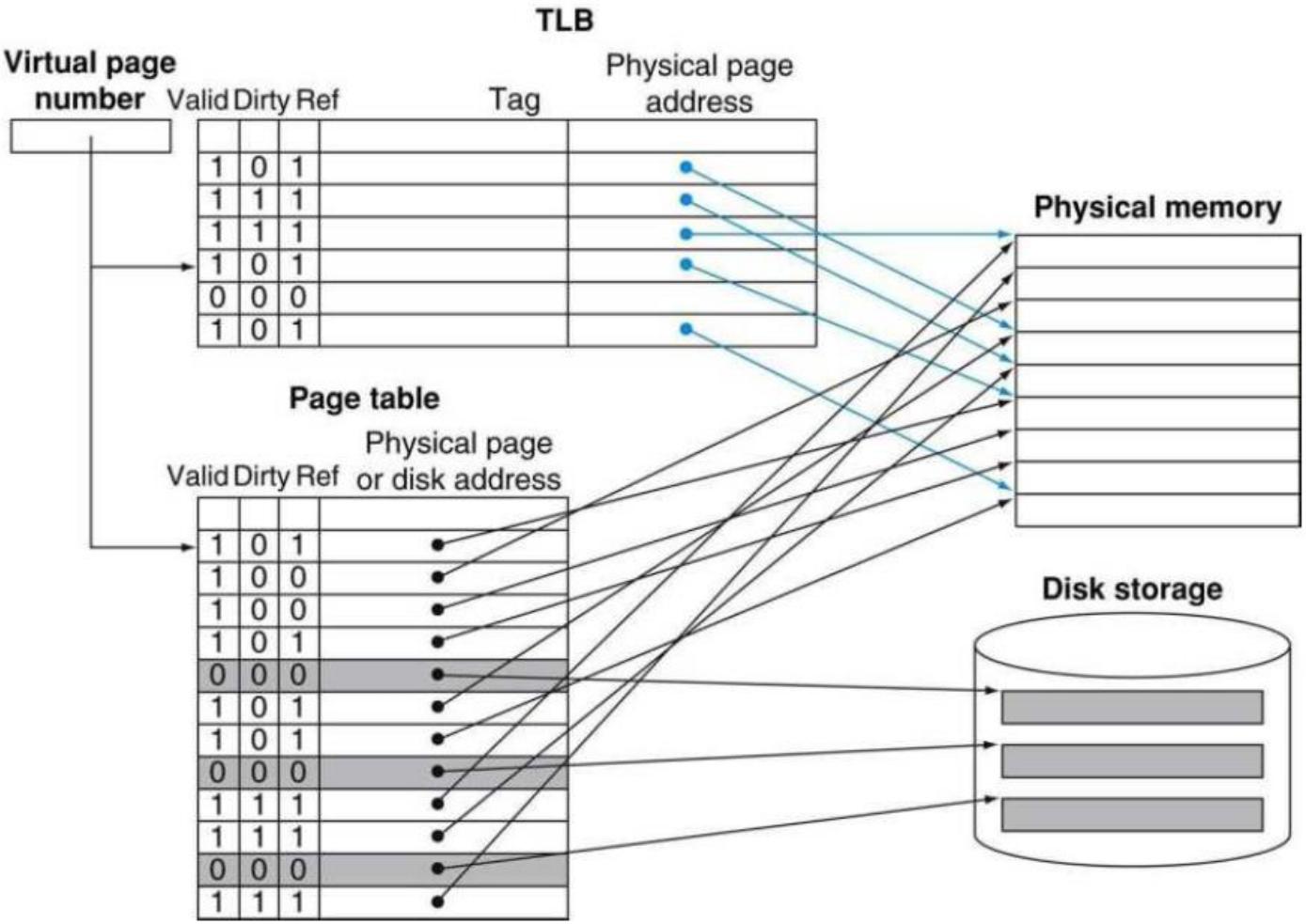
- Locate pages using a table that indexes the main memory = page table
- Page table resides in main memory
- Valid bit = 0→ page fault
  - OS should manage
  - Swap space



Page Table

- Maps virtual address to physical address
- Stored in main memory
- Virtual page number to index page table entry
  - Assume Byte-addressing
  - 4KiB pages → 2^{12} bytes in a page
  - 48-bit Virtual address → 12 bit page offset
  - 36 bits for indexing page table entries
  - 2^{36} page table entries ~ 64 billion entries!
- Techniques to reduce page table size
  - E.g. page the page table, multi-level table

Translation Look-aside Buffer



TLB

- A special cache to keep address translations
- Accesses within a page may have
  - Temporal locality
  - Spatial locality
- Reduce memory accesses needed to lookup the page table
- TLB includes additional status bits
  - dirty bit, reference bit

References

- Waterman, A., Lee, Y., Patterson, D. A., & Asanovi, K. (2014). *The risc-v instruction set manual. volume 1: User-level isa, version 2.0*. California Univ Berkeley Dept of Electrical Engineering and Computer Sciences.
- Harris, S. L., Chaver, D., Piñuel, L., Gomez-Perez, J. I., Liaqat, M. H., Kakakhel, Z. L., ... & Owen, R. (2021, August). RVfpga: Using a RISC-V Core Targeted to an FPGA in Computer Architecture Education. In *2021 31st International Conference on Field-Programmable Logic and Applications (FPL)* (pp. 145-150). IEEE.
- Hennessy, J. L., & Patterson, D. A. (2011). *Computer architecture: a quantitative approach*. Elsevier.
- Patterson, D. A., & Hennessy, J. L. (2016). *Computer organization and design ARM edition: the hardware software interface*. Morgan kaufmann.

THANK YOU