University of Glasgow

School of Computing Science

60 YEARS OF COMPUTING AT GLASGOW

Hardware Trends

Advanced Operating Systems

Lecture 3

Lecture Outline

- Hardware trends and systems performance
  - Moore's law
  - Dennard Scaling
  - Power constraints
- System heterogeneity
  - CPU cores, memory, storage, and networking

The PDP-11/40 and Unix

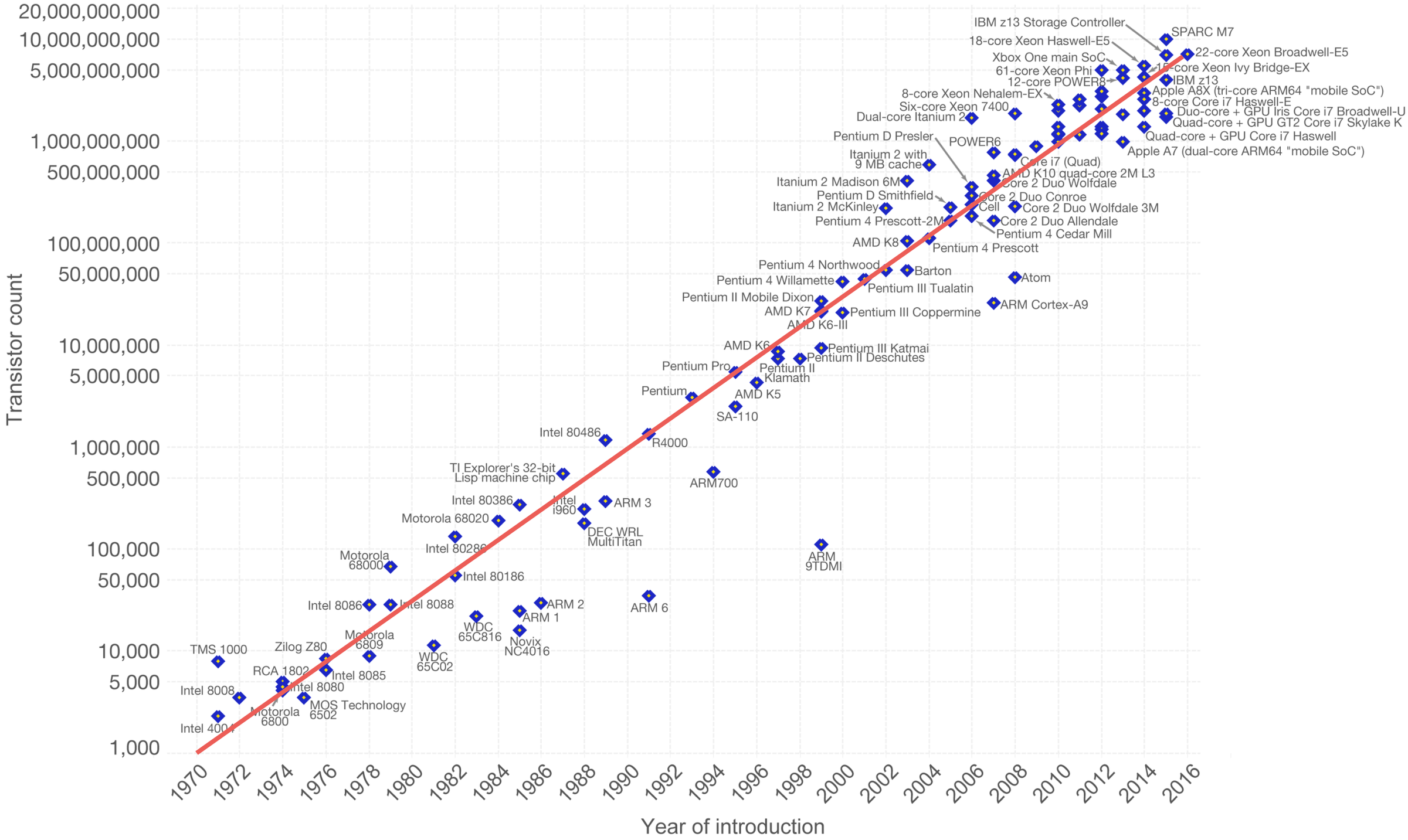
- The overwhelming majority of modern systems run some variant of Unix
  - macOS, iOS, Android, Linux
- Unix was designed in the early 1970s to run on PDP-11/40 minicomputers:
  - “The PDP-11/40 was designed to fit a broad range of applications, from small stand alone situations where the computer consists of only 8K of memory and a processor, to large multi-user, multi-task applications requiring up to 124K of addressable memory space. Among its major features are a fast central processor with a choice of floating point and sophisticated memory management, both of which are hardware options.”
    https://pdos.csail.mit.edu/6.828/2005/readings/pdp11-40.pdf
- Unix has proven surprisingly resilient and portable – is it still the right model?

Source: Wikipedia

Moore’s Law – The number of transistors on integrated circuit chips (1971-2016)

Our World in Data

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important as other aspects of technological progress – such as processing speed or the price of electronic products – are strongly linked to Moore's law.

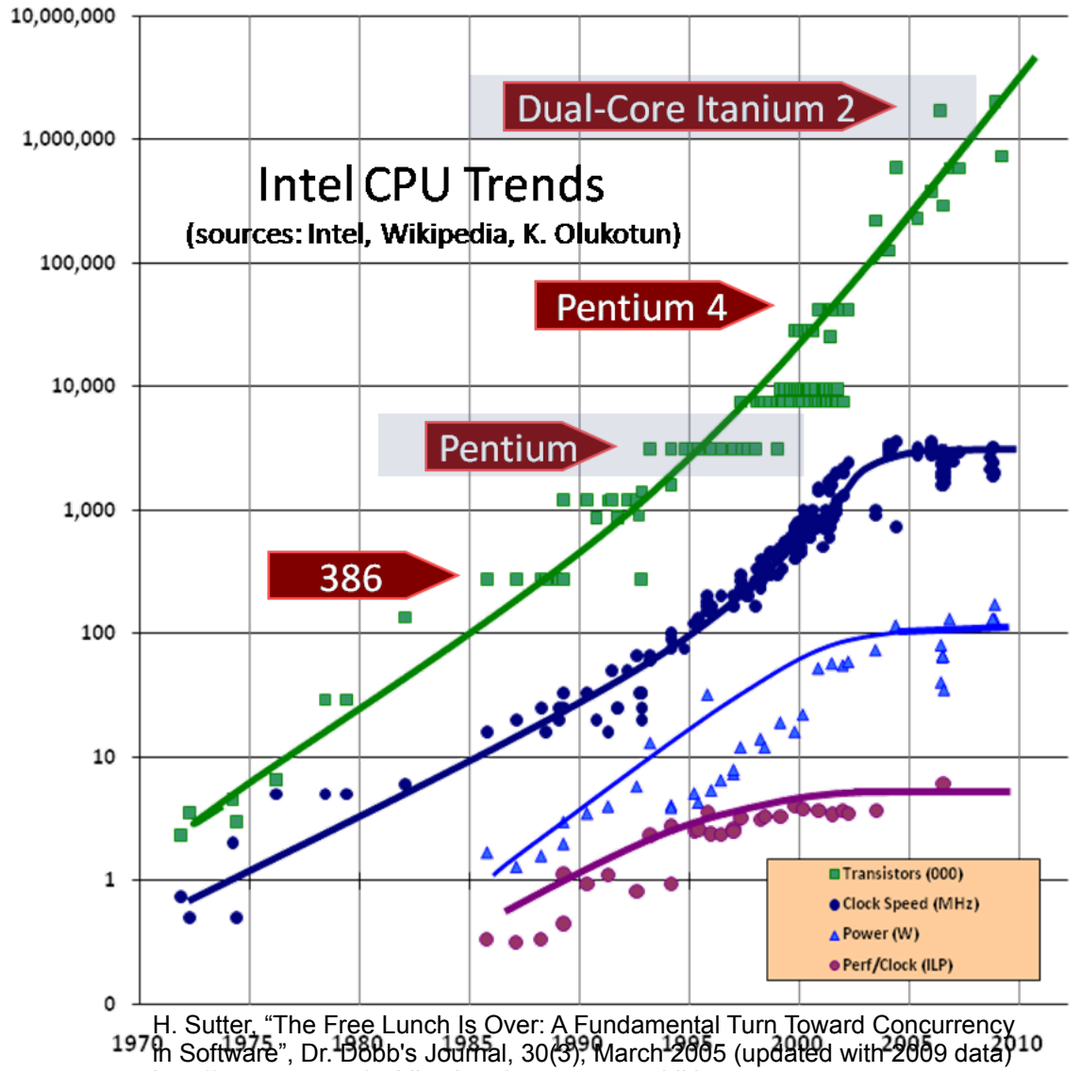


Data source: Wikipedia (https://en.wikipedia.org/wiki/Transistor_count)
The data visualization is available at OurWorldinData.org. There you find more visualizations and research on this topic.

Moore’s Law and Hardware Trends

- Moore’s law has held up well
- Other features have not:
  - Clock speed
  - Performance per clock cycle
  - Power consumption

- Starting to reach physical limits



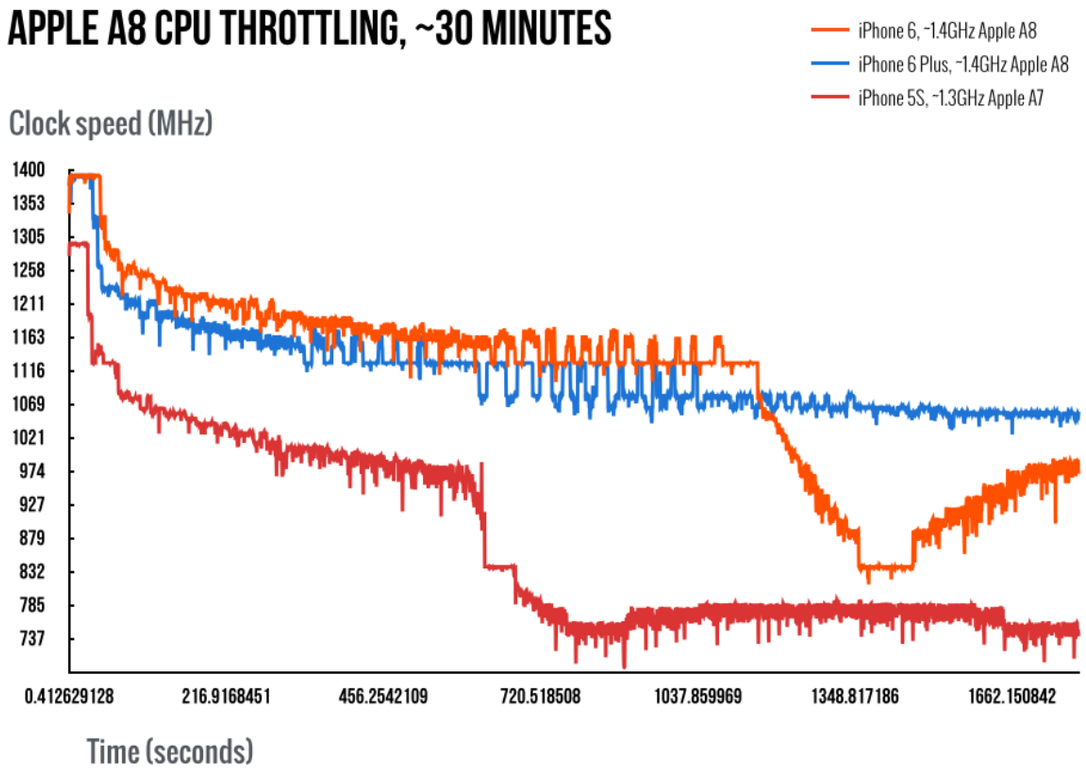
H. Sutter, “The Free Lunch Is Over: A Fundamental Turn Toward Concurrency in Software”, Dr. Dobb's Journal, 30(3), March 2005 (updated with 2009 data) http://www.gotw.ca/publications/concurrency-ddj.htm

More recent data in A. Danowitz et al, “CPU DB: Recording Microprocessor History”, Communications of the ACM, 55(4), April 2012, Pages 55-63. DOI: 10.1145/2133806.2133822 – shows the trends continue into 2012

Dennard Scaling

- Moore's law: transistor count, not performance
- Power consumption driven by Dennard scaling:
  - Power $\propto C \cdot F \cdot V^2$ where:
    C is capacitance
    F is clock frequency
    V is voltage
  - Size of transistor directly affects V and C
  - Implies smaller transistors (Moore's law) allows higher frequency for same power consumption
- Approximation that ignores *leakage current*
  - Limitations of semiconductor physics that dominate as sizes decreases – eventually became dominant source of power consumption
  - Dennard scaling equation no longer applies

Thermal Throttling



Source: arstechnica.com

- CPU thermal throttling – cannot sustain performance
  - Graph is for Apple A8 SoC
  - Many other processors behave similarly
- Dark silicon – one consequence of Moore's law and breakdown of Dennard scaling; can't power the entire chip at once

Implications of Hardware Trends

- Moore's law → how much longer?
- Breakdown in Dennard scaling:
  - Increased thermal throttling
  - Multicore systems
  - Power constraints – mobile *and* data centre

- Mobile devices have performance in bursts
- Data centres scale wider rather than faster

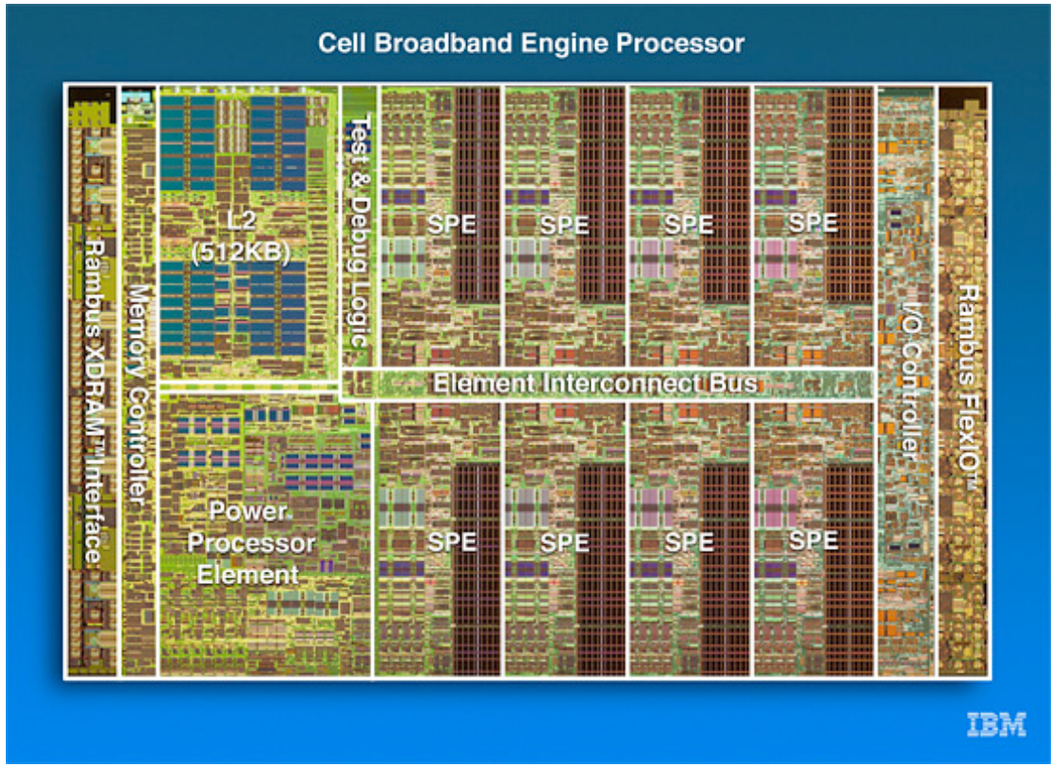
Heterogeneity

- Range of system designs increasing
  - Data centres
  - PCs and laptops
  - Mobile phones and tablets
  - Wearables
  - IoT

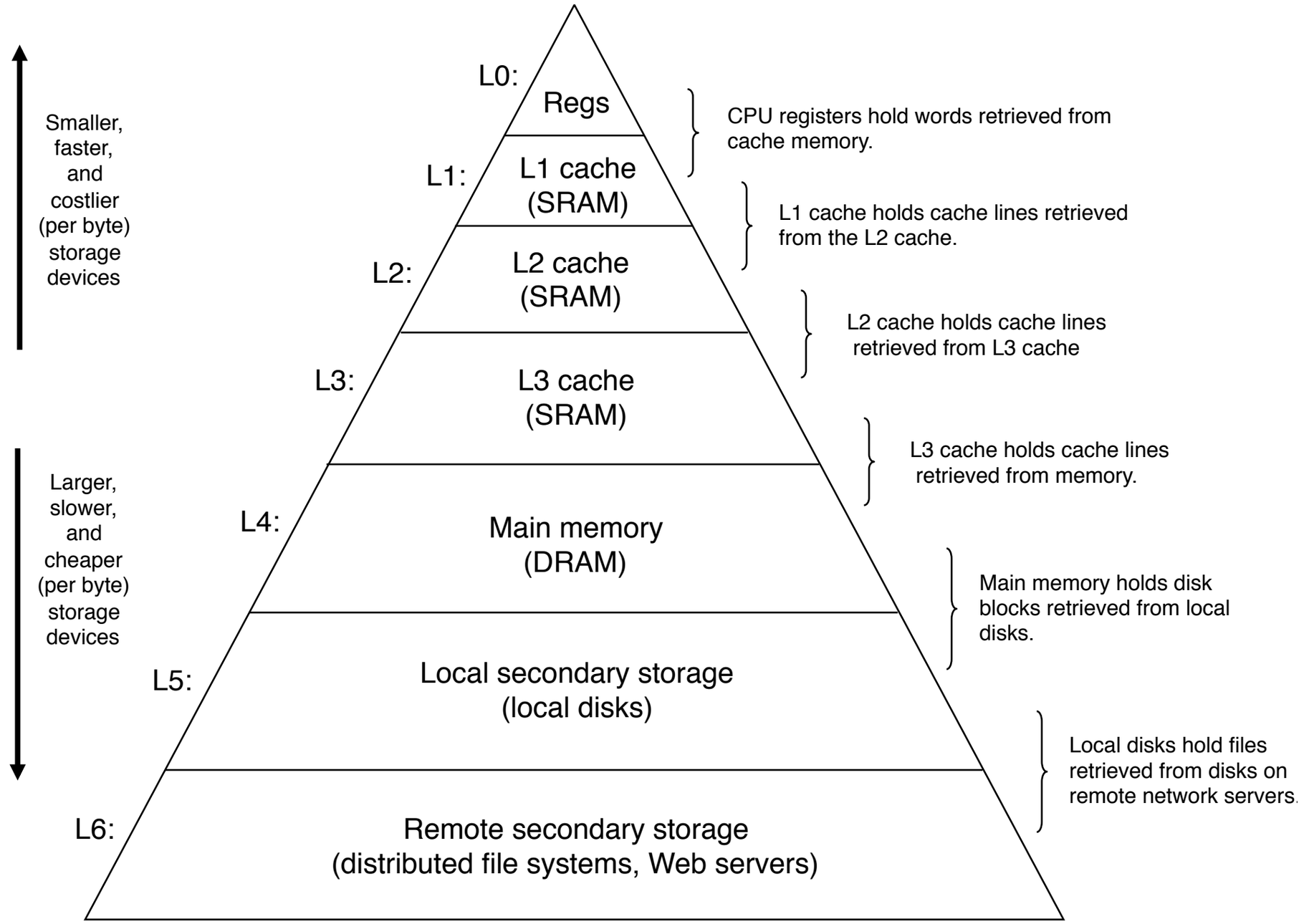
- Scalability and heterogeneity
  - Due to increasing range of different applications
  - To make effective use of Moore's law, while subject to power constraints → optimise hardware for particular use cases, power only when needed

Heterogeneity: Processor Architecture

- Heterogeneous multiprocessor: CPU with multiple special purpose cores
  - Canonical example → Cell Broadband Engine
- Asymmetric processing capabilities
  - High-performance and low-power cores on a single die (e.g., ARM big.LITTLE model, with both Cortex A7 and A15 cores on-die)
  - GPU-like cores for graphics operations, with single program multiple data model vs. a more traditional multiple program multiple data model
  - Offload for crypto algorithms, TCP stack, etc.
- Asymmetric memory access models
  - Non-cache coherent
  - Cores explicitly do not share memory
- Common for mobile phones, games consoles, and other non-PC hardware



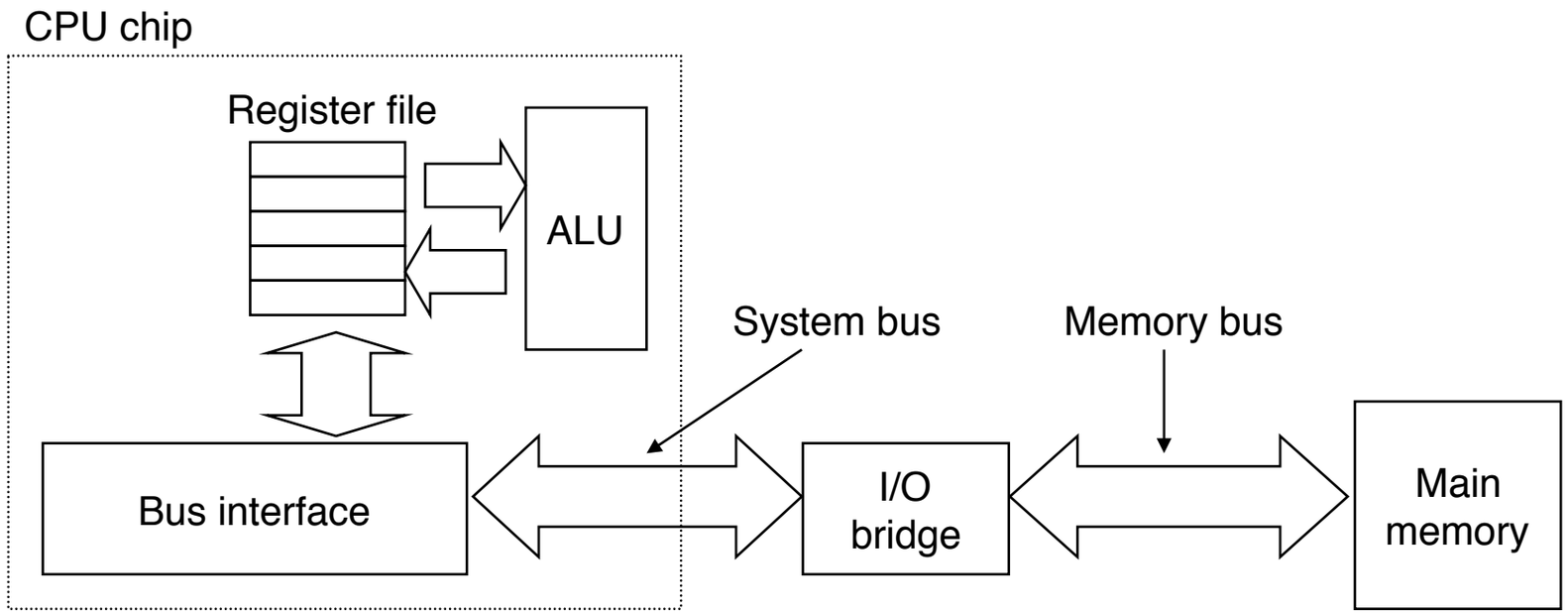
Heterogeneity: Memory Hierarchy



Source: Bryant & O'Hallaron, "Computer Systems: A Programmer's Perspective", 3rd Edition, Pearson, 2016
Fig. 6.21. http://csapp.cs.cmu.edu/3e/figures.html (website grants permission for lecture use with attribution)

Heterogeneity: Memory Hierarchy

- The traditional view of memory in a computer system:



Source: Bryant & O'Hallaron, "Computer Systems: A Programmer's Perspective", 3rd Edition, Pearson, 2016
Fig. 6.6. http://csapp.cs.cmu.edu/3e/figures.html (website grants permission for lecture use with attribution)

- Note:
  - One CPU
  - One memory bus
  - One block of memory
- Uniform memory access

Heterogeneity: Memory Hierarchy

- Organisation of a typical modern computer system:

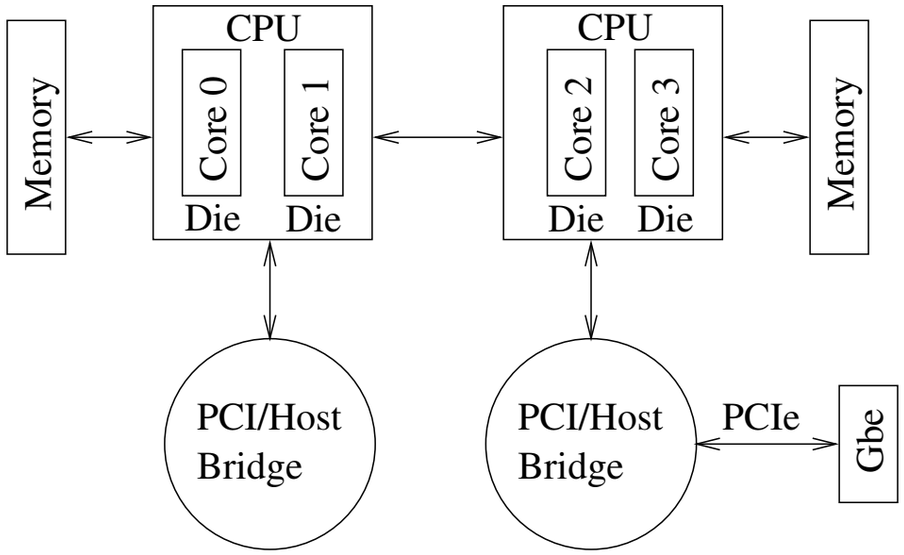


Figure 2. Structure of the AMD system

A. Schüpbach, *et al.*, Embracing diversity in the Barrelfish manycore operating system. Proc. Workshop on Managed Many-Core Systems, Boston, MA, USA, June 2008. ACM.

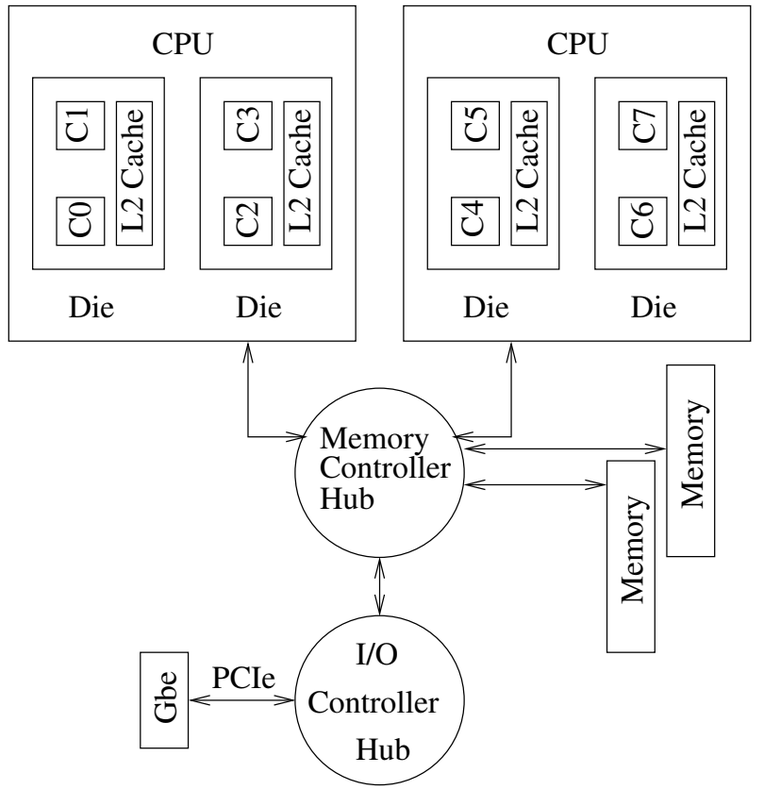
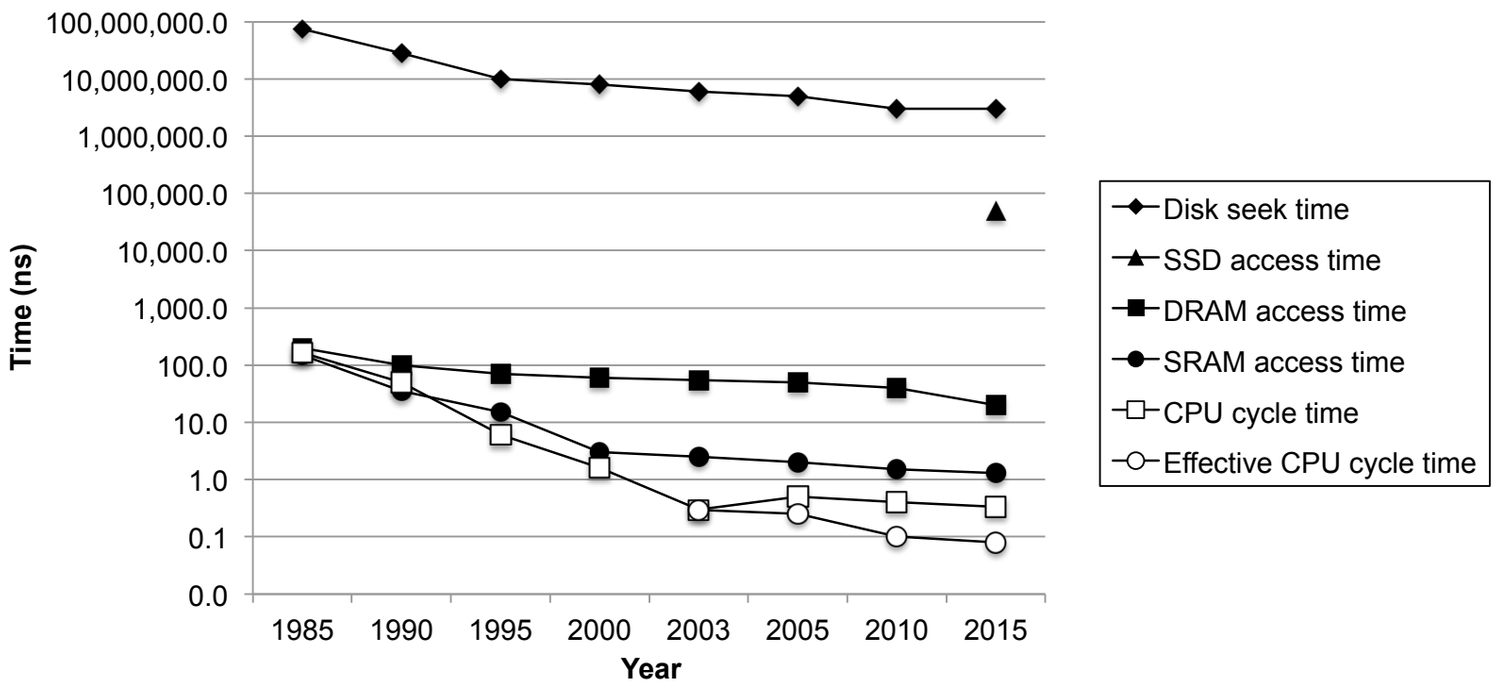


Figure 1. Structure of the Intel system

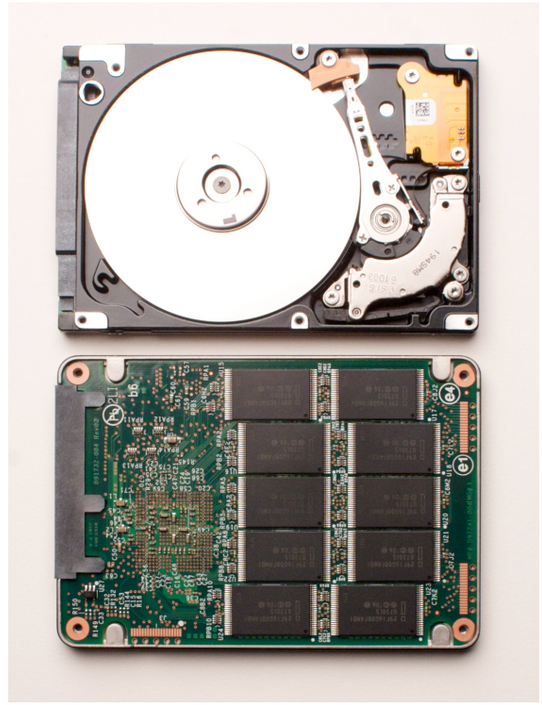
- Non-uniform memory access (NUMA)
  - Large on-chip cache memory; main memory off-chip, accessed via interconnect
  - Heterogeneity unavoidable, due to the physical layout of the hardware – some memory is physically closer to some CPU chips than to others
  - Cache coherency protocols maintain random access illusion
  - Memory access latency varies depending on which core is accessing which memory bank

Heterogeneity: Storage

- Hard disks → solid state disks
  - No moving parts
  - True random access – hard disk seek time varies with physical movement of disk heads
  - Lower latency, higher throughput
  - Lower power consumption
  - Wear levelling and block-level FEC – limited number of write cycles
  - Sophisticated controller hidden in the SSD – embedded computer, not device driver



Source: Bryant & O'Hallaron, "Computer Systems: A Programmer's Perspective", 3rd Edition, Pearson, 2016
Fig. 6.16. http://csapp.cs.cmu.edu/3e/figures.html (website grants permission for lecture use with attribution)



Source: Intel

Heterogeneity: Networking

- Network performance has caught up with CPU performance
  - DMA-based ring-buffer interface to NIC
  - Zero copy from packet reception to user space
  - Protocol header processing and kernel data structures are a significant cost – difficult to keep up with line rate processing of small packets on high speed links
  - Pushing towards different device driver and network APIs – sockets API a bottleneck

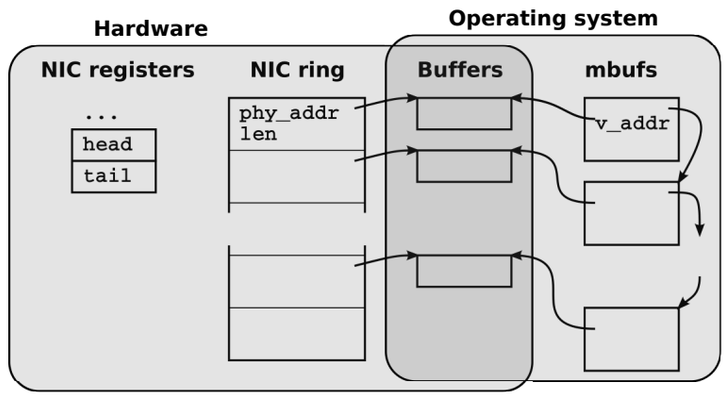


Figure 1: Typical NIC's data structures and their relation with the OS data structures.

From: L. Rizzo, "netmap: a novel framework for fast packet I/O", Proc. USENIX Annual Technical Conference, Boston, MA, USA, June 2012.

- Increasing range of interface types: wired and wireless
- Wireless and mobility expose the limitations of TCP congestion control and service model
- Multiple interfaces, wireless, and mobility push towards multipath protocols, protocols for path discovery and edge-to-network communication, and new APIs

Discussion

- Systems are increasingly heterogenous, and quite different to those used when the OS was designed

- How should change the programming model?
- Are we using appropriate programming languages, tools, and operating systems architectures?