Advanced Operating Systems
1.Introduction.1

Chun-Han Lin (林均翰)

CSIE, NTNU



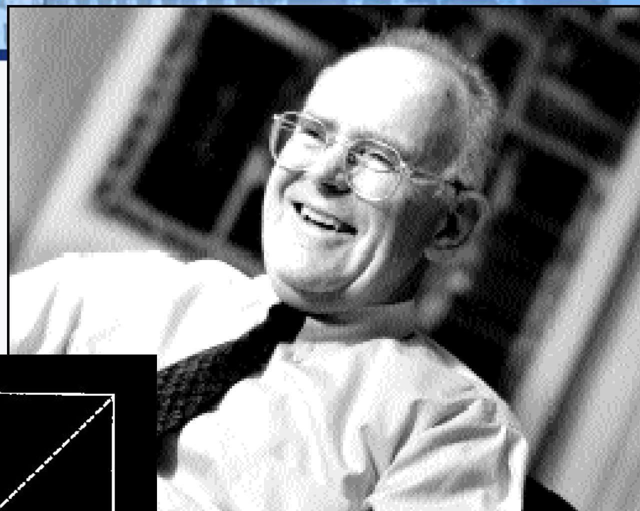
Key Points 1.1

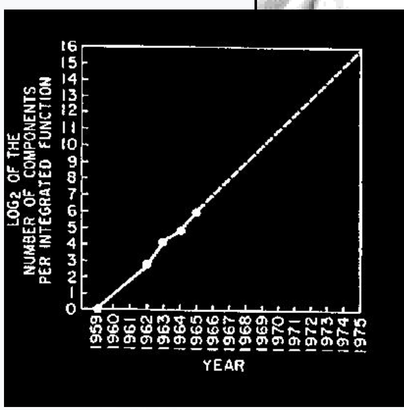
- **Chapter goals**
- **Moore’s Law**
- **Social impacts**

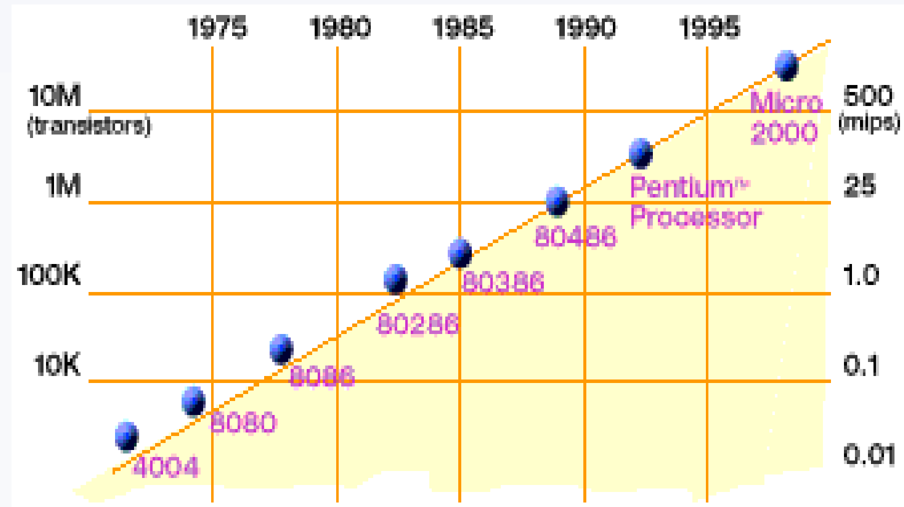
Goals

- Why is writing an operating system (OS) interesting and difficult?
- Why study OS?
- OS Basics

Technology Trends: Moore's Law







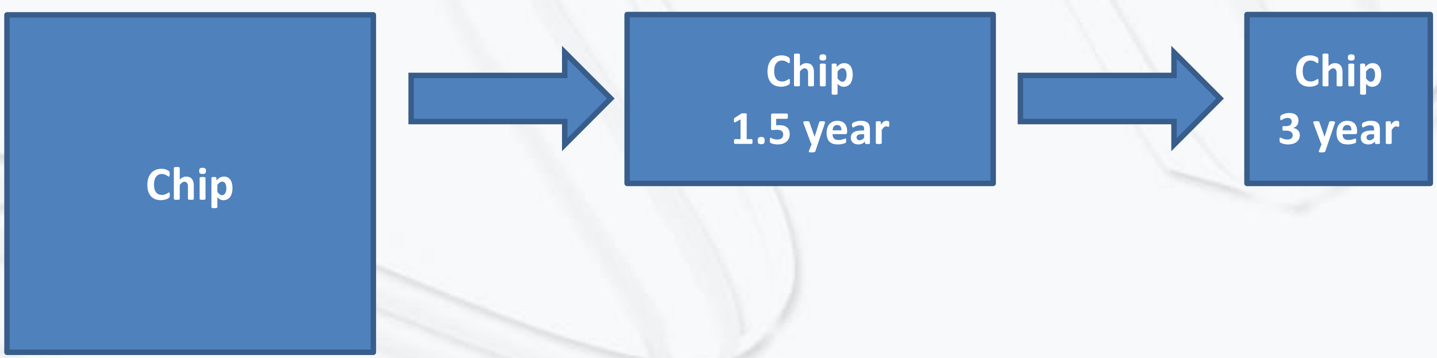
Gordon Moore (co-founder of Intel) predicted in 1965 that the transistor density of semiconductor chips would double roughly every 18 months.

2x transistors/chip every 1.5 year is called <u>Moore's Law</u>.

Microprocessors have become smaller, denser, and more powerful.

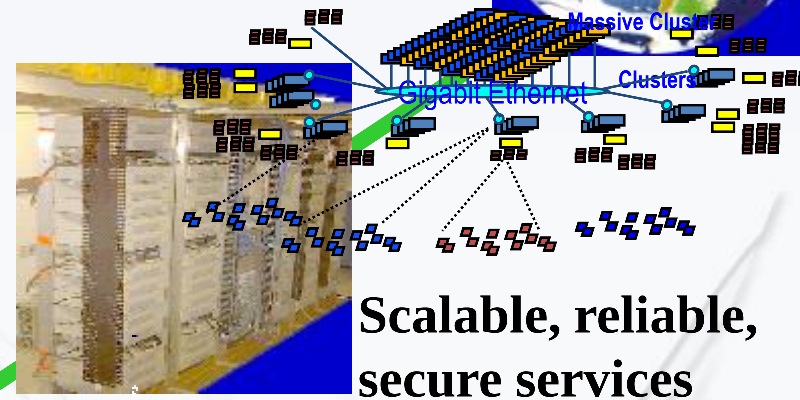
Impacts

- If the number of transistors on a chip implies the number of functions, …
  - Smaller => more devices
  - More powerful => more useful



Societal Scale Information Systems

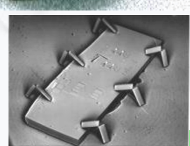
- The world is a large parallel system.
  - Microprocessors in everything
  - Vast infrastructure behind them

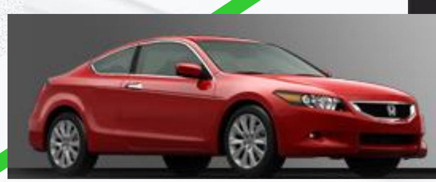


Scalable, reliable, secure services

Internet connectivity

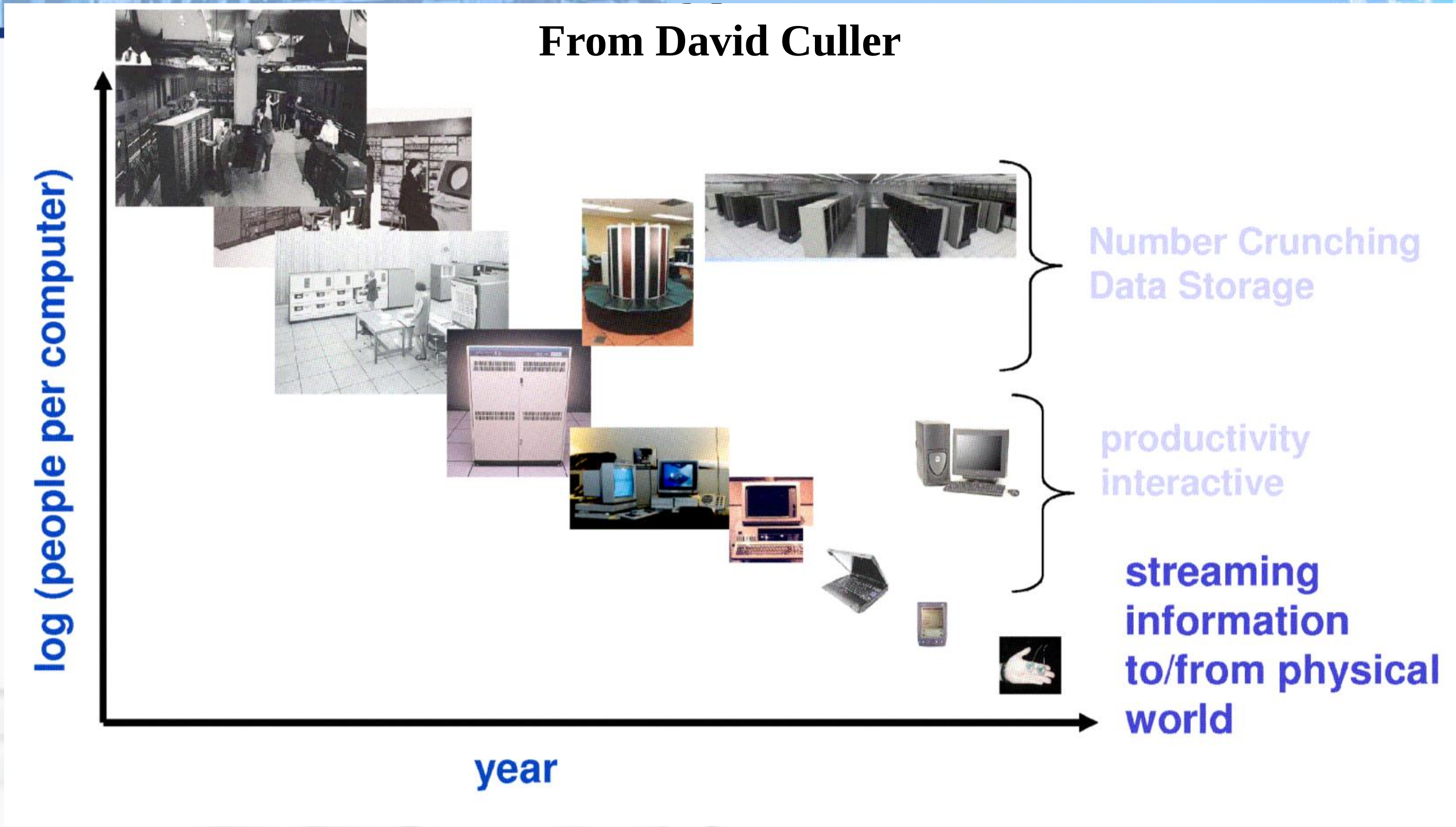
Databases, information collection, remote storage, online games, commerce, …





MEMS for sensor nets

People-to-Computer Ratio over Time



- **Today: multiple CPU per person!**
  - **Approaching 100s?**

Review 1.1

- **Chapter goals**
- **Moore’s Law**
- **Social impacts**

Advanced Operating Systems
1.Introduction.2

Chun-Han Lin (林均翰)

CSIE, NTNU

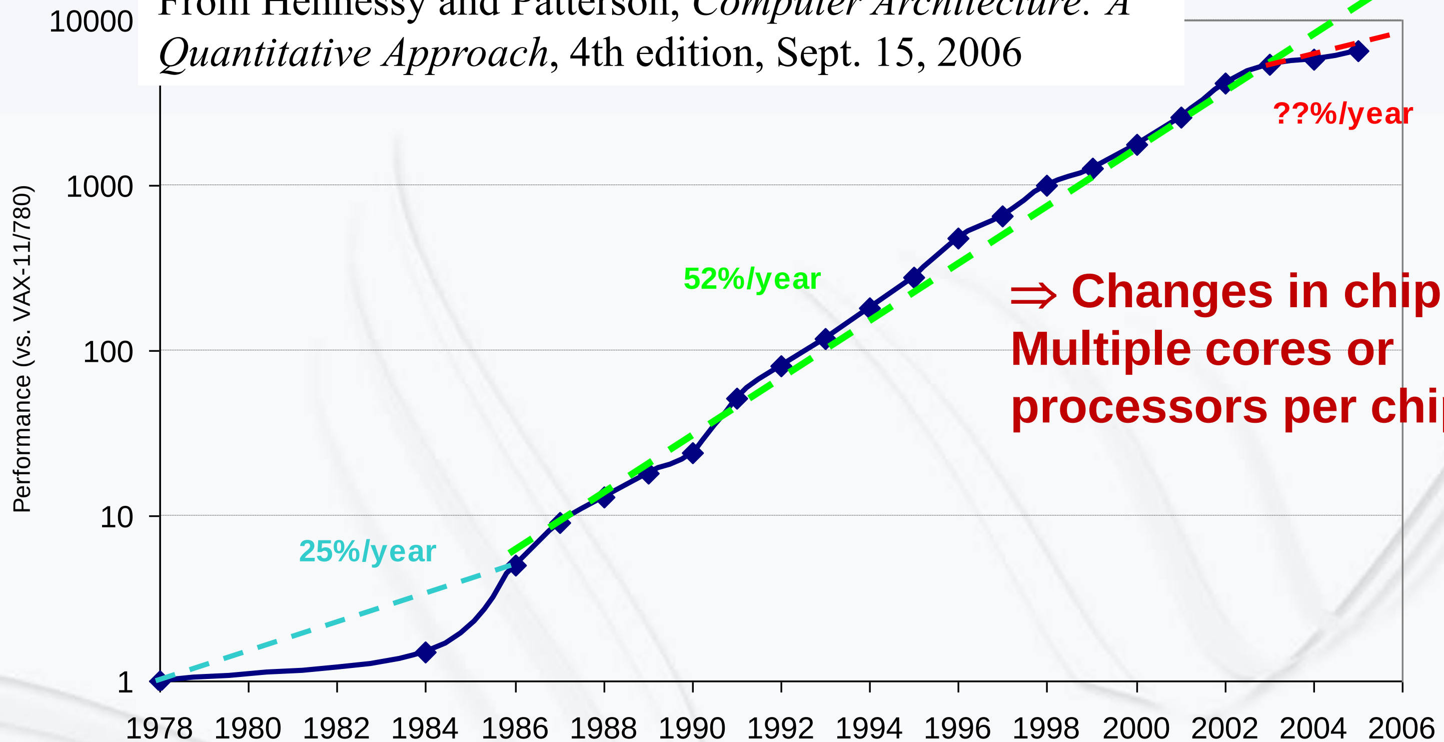


Key Points 1.2

- **Performance improvement challenge**
- **Power challenge**

New Challenge: Slowdown in Joy's Law of Performance

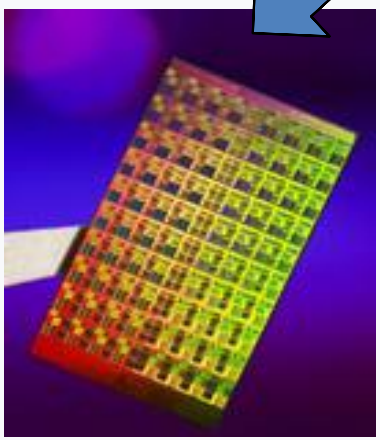
From Hennessy and Patterson, *Computer Architecture: A Quantitative Approach*, 4th edition, Sept. 15, 2006



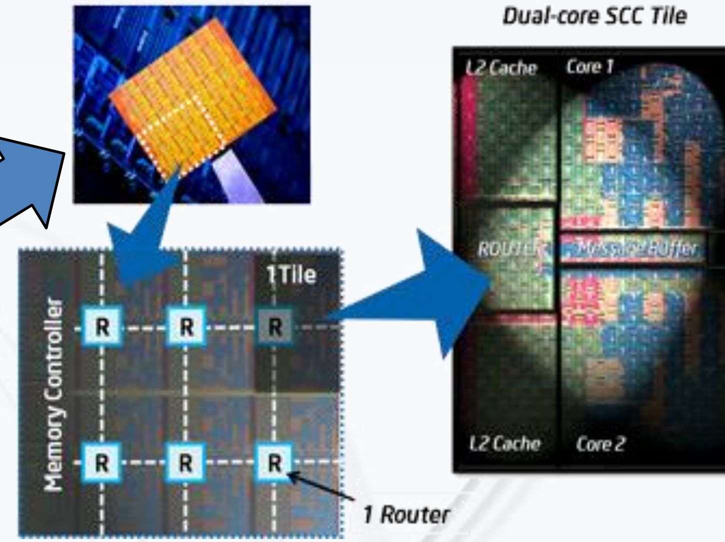
⇒ Changes in chip design: Multiple cores or processors per chip

- **VAX : 25% / year, 1978 to 1986**
- **RISC + x86: 52% / year, 1986 to 2002**
- **RISC + x86: ??% / year, 2002 to present**

ManyCore Chips: The Future is Here

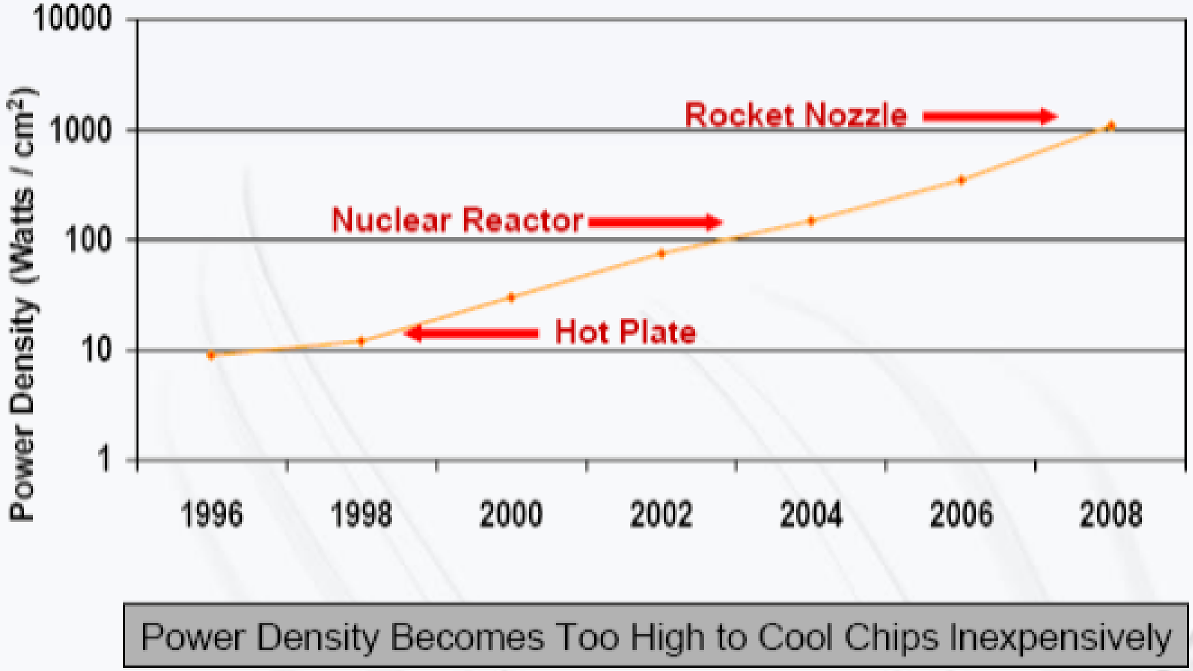


- Intel 80-core multicore chip (Feb 2007)
  - 80 simple cores
  - Two FP-engines/core
  - Mesh-like network
  - 100 million transistors
  - 65nm feature size
- Intel single-chip cloud computer (Aug 2010)
  - 24 “tiles” with two cores/tile
  - 24-router mesh network
  - 4 DDR3 memory controllers
  - Hardware support for message-passing



- **“ManyCore” refers to many processors/chip**
  - **64? 128? Hard to say exact boundary**
- **How to program these?**
  - **Use 2 CPUs for video/audio, 1 for word processor, 1 for browser, 76 for virus checking?**
- **Parallelism must be exploited at all levels.**

Another Challenge: Power Density



- Moore’s Law extrapolation
  - Potential power density reaches amazing levels!
- Flip side: battery life is very important.
  - Moore’s Law can yield more functionality at equivalent or less total energy consumption.

Review 1.2

- **Performance improvement challenge**

- **Power challenge**

Advanced Operating Systems
1.Introduction.3

Chun-Han Lin (林均翰)

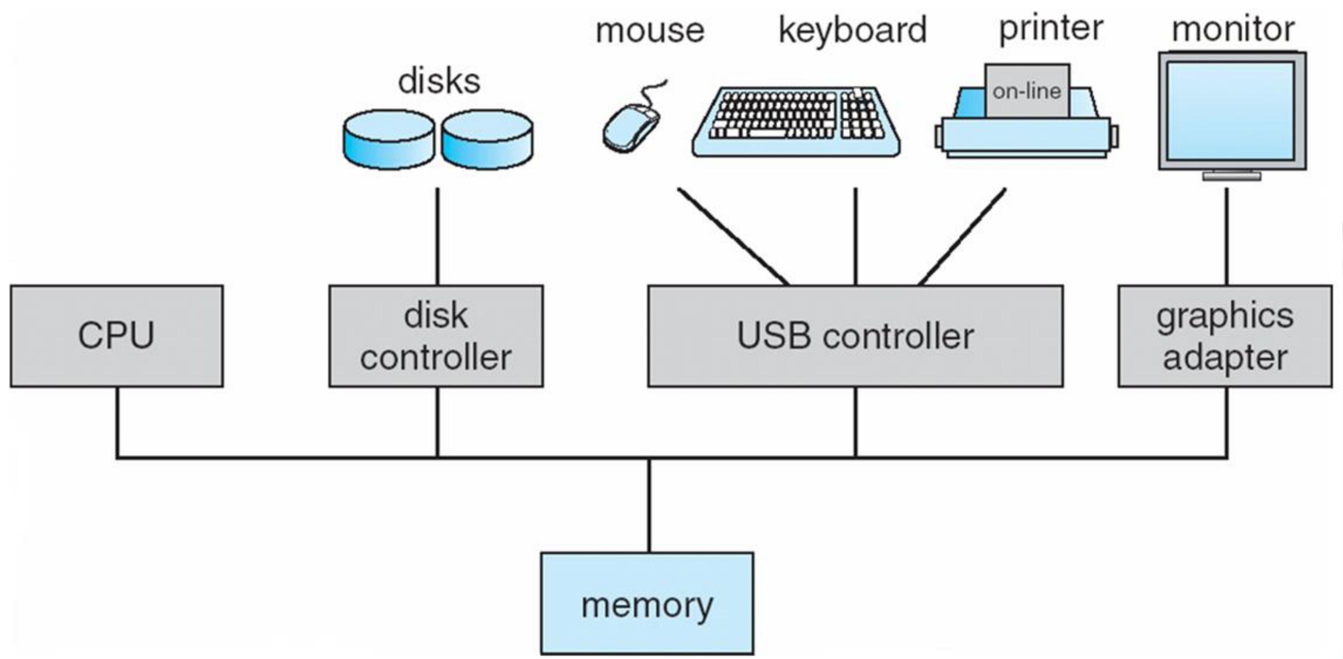
CSIE, NTNU

Key Points 1.3

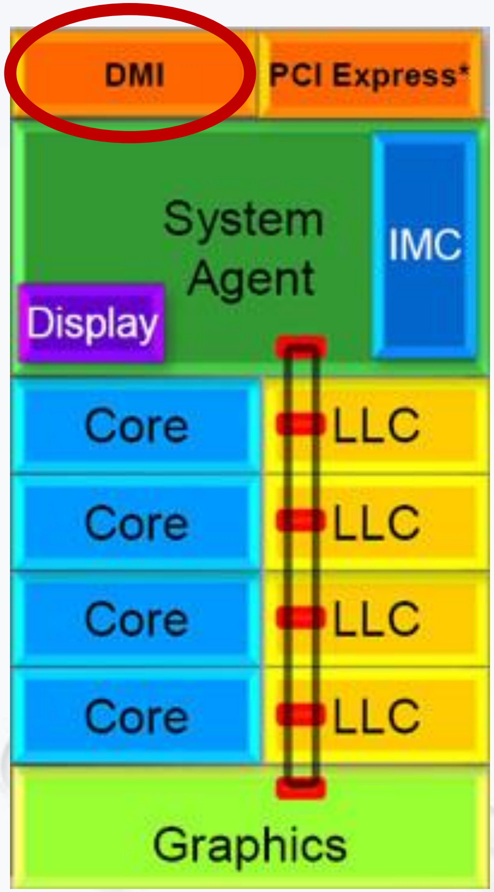
- **Organization of computer systems**
- **Sandy Bridge architecture**
- **Architecture topics**

Organization of Computer Systems

- One or more CPU, device controllers connect through common bus providing access to share memory
- Concurrent execution of CPU and devices competing for memory cycles

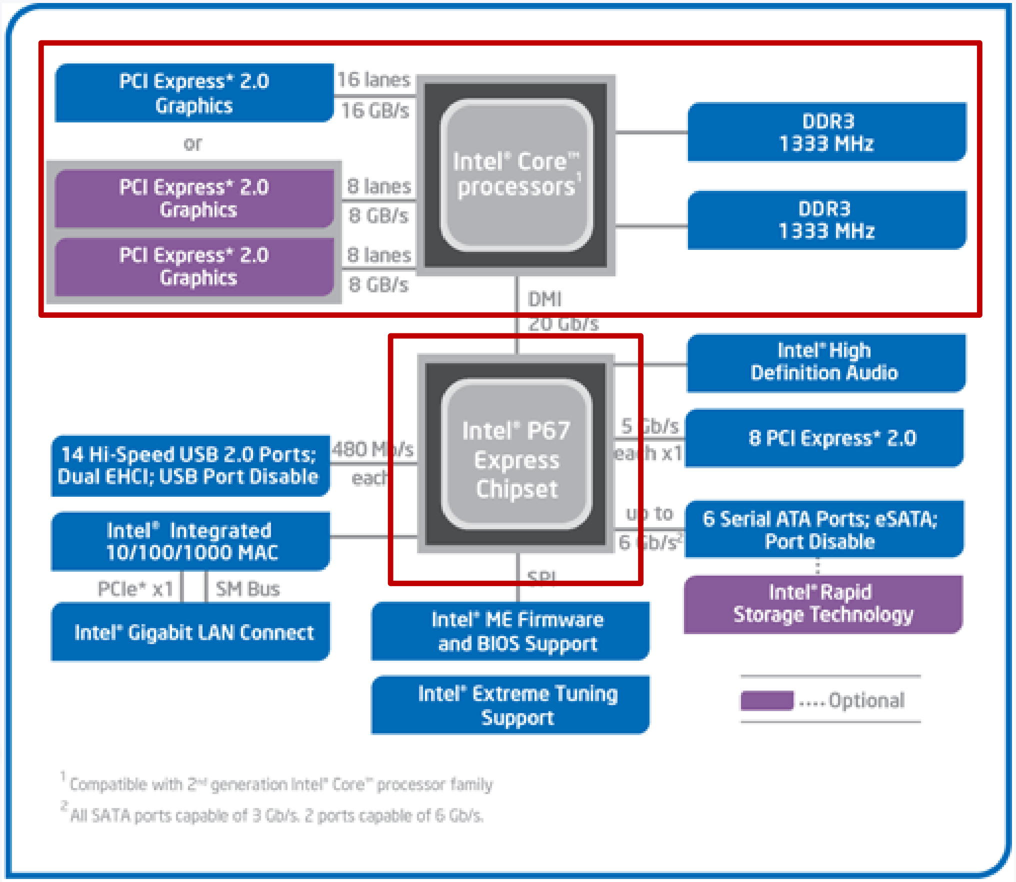


Chip-Scale Features of Sandy Bridge



- Four OOO cores, 4 μ-ops/cycle
  - New Advanced Vector eXtensions (256-bit FP)
  - AES instructions
  - Instructions to help with Galois-Field mult
- Integrated GPU
- System Agent (Memory and Fast I/O)
- Shared L3 cache divided in 4 banks
- On-chip ring bus network
  - Both coherent and non-coherent transactions
  - High-BW access to L3 Cache
- Integrated I/O
  - Integrated Memory Controller (IMC)
    - Two independent channels of DDR3 DRAM
  - High-speed PCI-Express for graphics cards
  - DMI connection to SouthBridge (PCH)

Sandy Bridge I/O: PCH



Sandy Bridge system configuration

- Platform Controller Hub (PCH)
  - Used to be "SouthBridge," but no "NorthBridge" now
  - Connected to processor with proprietary bus
    - Direct Media Interface (DMI)
  - Code name "Cougar Point" for Sandy Bridge processors
- Types of I/O on PCH
  - USB
  - Ethernet
  - Audio
  - BIOS support
  - More PCI Express (lower speed than on Processor)
  - Sata (for Disks)

Sample of Computer Architecture Topics

Input/Output and storage

Disks, Tape

RAID

DRAM

Emerging technologies: Interleaving, bus protocols

L2 Cache

Coherence, bandwidth, latency

L1 Cache

Addressing, protection, exception handling

Network communication

Other processors

Memory hierarchy

VLSI

Instruction Set Architecture

Pipelining, hazard resolution, superscalar, reordering, prediction, speculation, vector, dynamic compilation

Pipelining and instruction-level parallelism

Review 1.3

- Organization of computer systems
- Sandy Bridge architecture
- Architecture topics

Advanced Operating Systems
1.Introduction.4

Chun-Han Lin (林均翰)

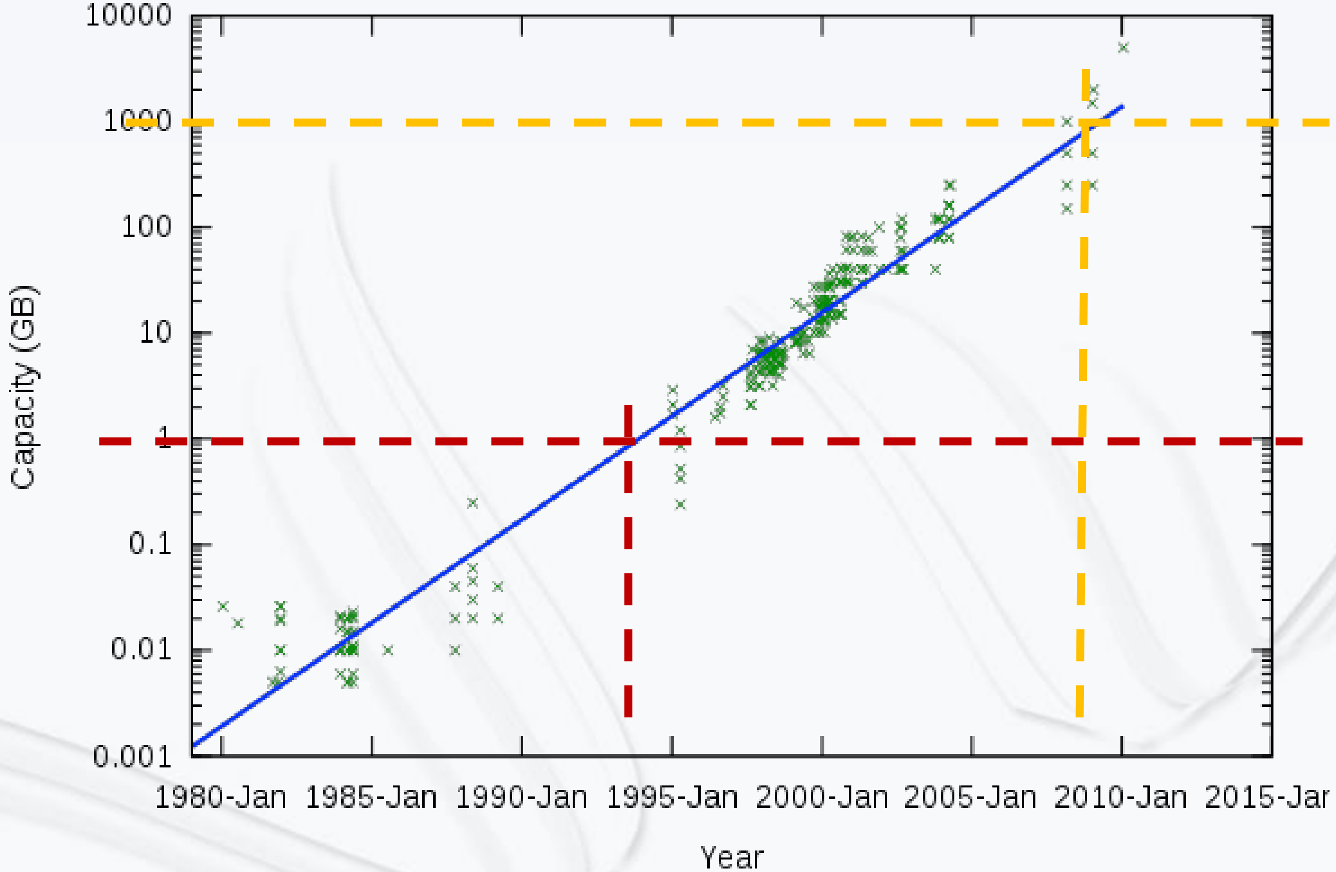
CSIE, NTNU



Key Points 1.4

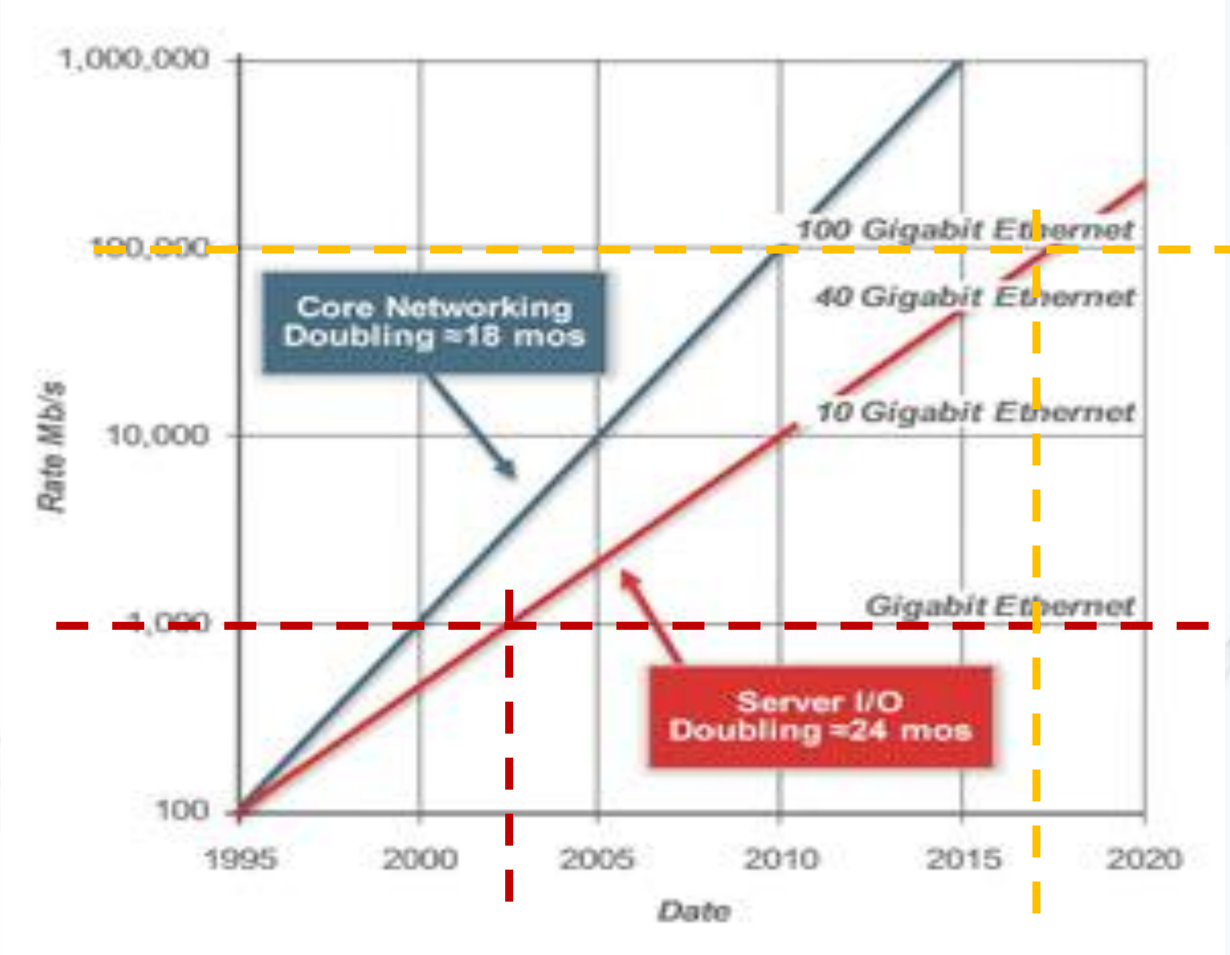
- **Storage capacity**
- **Network capacity**
- **Software complexity**

Storage Capacity



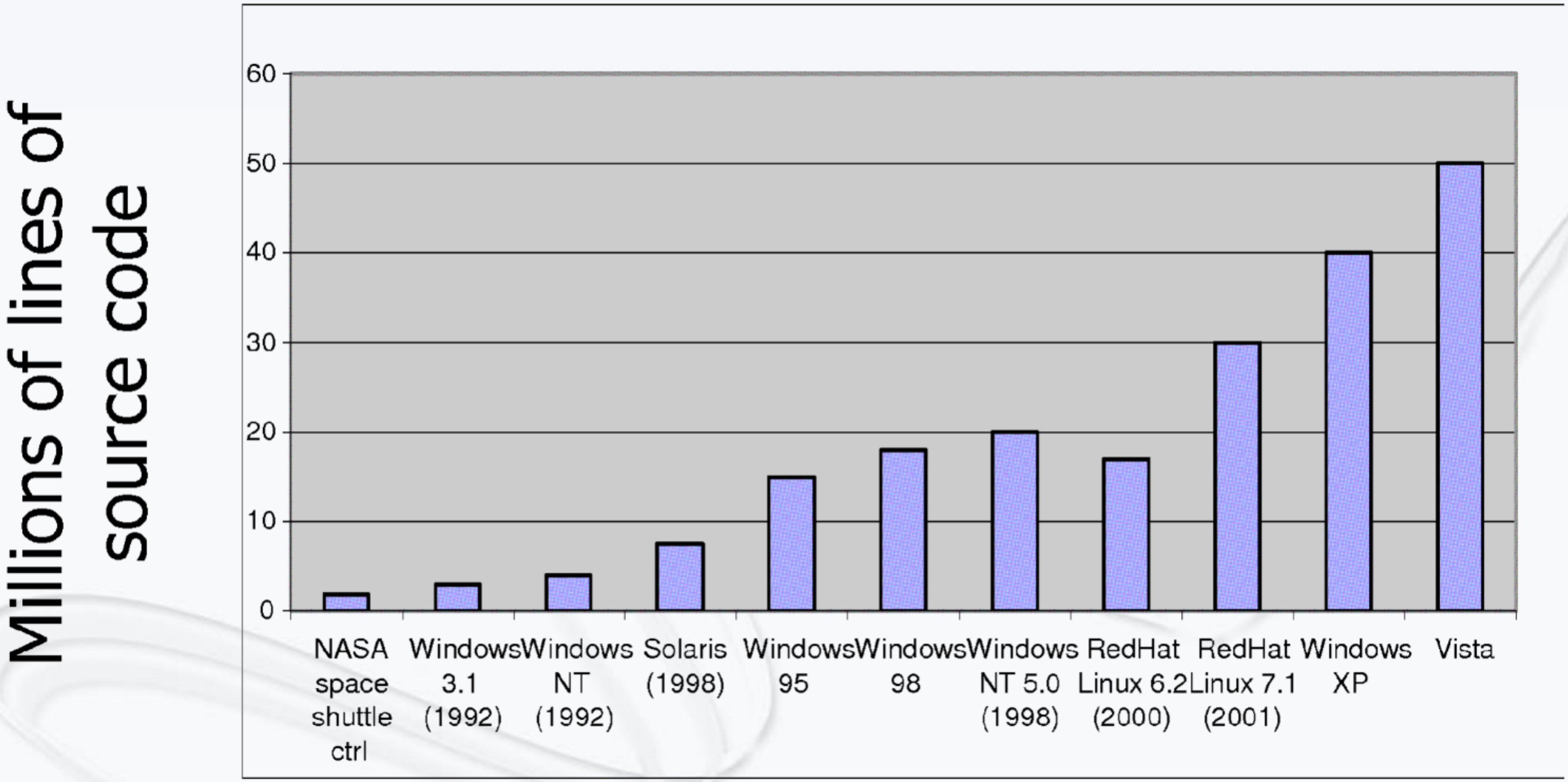
- **Hard disk capacity (in [GB])** (http://www.digitaltonto.com/2011/our-emergent-digital-future)

Network Capacity



http://www.ospmag.com/issue/article/Time-Is-Not-Always-On-Our-Side

Increasing Software Complexity



MIT's 6.033 course

Review 1.4

- **Storage capacity**
- **Network capacity**
- **Software complexity**

Advanced Operating Systems
1.Introduction.5

Chun-Han Lin (林均翰)

CSIE, NTNU



Key Points 1.5

- **How do we tame complexity?**

- **OS tool: Virtual machine abstraction**

How do we tame complexity? (1/2)

- Every piece of computer hardware is different.
  - Different CPU
    - Pentium, PowerPC, ColdFire, ARM, MIPS, …
  - Different amounts of memory, disk, …
  - Different types of devices
    - Mice, keyboards, sensors, cameras, fingerprint readers, …
  - Different networking environment
    - Cable, DSL, wireless, firewalls, …

How do we tame complexity? (2/2)

- Questions
  - Does a programmer need to write a single program that performs many independent activities?
  - Does every program have to be altered for every piece of hardware?
  - Does a faulty program crash everything?
  - Does every program have access to all hardware?

OS Tool: Virtual Machine Abstraction

Applications

---- Virtual machine interface

Operating Systems

---- Physical machine interface

Hardware

- Software engineering problems
  - Turn hardware/software quirks ⇒ what do programmers want/need?
  - Optimize for convenience, utilization, security, reliability, …
- For any OS area, i.e., file systems, virtual memory, networking, scheduling, …
  - What's the hardware interface? (physical reality)
  - What's the application interface? (nicer abstraction)



Review 1.5

- How do we tame complexity?

- OS tool: Virtual machine abstraction

Advanced Operating Systems 1.Introduction.6

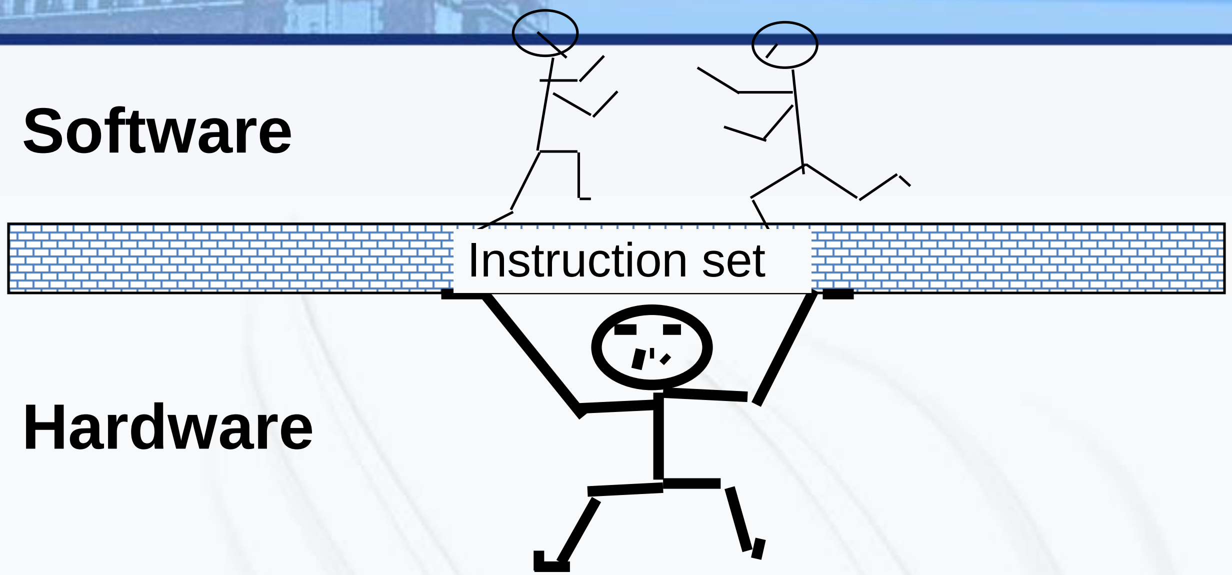
Chun-Han Lin (林均翰)
CSIE, NTNU

Key Points 1.6

- **Interfaces provide important boundaries**

- **Virtual machines**

Interfaces Provide Important Boundaries



- Why do interfaces look the way that they do?
  - History, functionality, stupidity, bugs, management, …
  - Machine interface
  - Human interface
  - Software engineering/management
- Should responsibilities be pushed across boundaries?
  - RISC architecture, graphical pipeline architecture, …

Virtual Machines (1/2)

- **Software emulation of an abstract machine**
  - **Make it look like hardware has features that you want**
  - **Programs from one hardware & OS on another one**
- **Programming simplicity**
  - **Each process thinks it has all memory/CPU time**
  - **Each process thinks it owns all devices**
  - **Different devices appear to have the same interface**
  - **Device interfaces can be more powerful than raw hardware**
    - **Bitmapped display $\Rightarrow$ windowing system**
    - **Ethernet card $\Rightarrow$ reliable and ordered networking (TCP/IP)**

Virtual Machines (2/2)

- **Fault isolation**
  - **Processes are unable to directly impact other processes**
  - **Bugs cannot crash whole machine**

- **Protection and Portability**
  - **Limits to what users programs are allowed to do**
  - **Stability of POSIX interface among systems**

Review 1.6

- **Interfaces provide important boundaries**

- **Virtual machines**