

Administrative Matters

- Course: Computer Organization
- Time/Location: 1B4EF-EC114
- Instructor: 李毅郎
- E-mail: ylli@cs.nctu.edu.tw
- URL: <http://www.cis.nctu.edu.tw/~ylli>
- Office: 工三441
- Office Hours: 1CD or make an appointment by @
- Teaching Assistants:
 - 孫若美: zoe40523@gmail.com
 - 楊承諺: jsamare01@gmail.com
 - 楊芷瑀: ka0802fance.cs02@nctu.edu.tw
 - 鳳明傑: a23321261@gmail.com
 - 林世庭: cxzasd3661512@gmail.com
- Prerequisites: Digital Circuit Design.



Administrative Matters

- Required Text:
 - David A. Patterson and John L. Hennessy, Computer Organization & Design-- The Hardware/Software Interface, 4th edition, 2009, MORGAN KAUFMAN.
- References:
 - Randal E. Bryant & David O'Hallaron, "Computer Systems: A Programmer's Perspective", Prentice Hall, ISBN 0-13-034074-X

Course Contents

- Course Goals
 - Learn the components of a computer and their relations,
 - Learn the interface between software and hardware,
 - Design a simple CPU.
- Course Contents
 - Chap 1. Computer Abstractions and Technology – 3 hrs
 - Chap 2. Instructions: Language of the Computer – 8 hrs
 - Chap 3. Arithmetic for Computers – 6 hrs
 - Chap 4. The Processor – 12 hrs
 - Chap 5. Large and Fast: Exploiting Memory Hierarchy – 9 hrs
 - Chap 6. Storage and Other I/O Topics – 6 hrs
 - Chap 7. Multicores, Multiprocessors, and Clusters – 2 hrs

Grading Policy

- Grading
 - Examinations: 2 exams, 50%
 - Term Project: 25% (1 or 2 members/team)
 - One-member teams are encouraged by bonus
 - Quizzes: 25%
 - Class participation: bonus
- Course Web Site: eCampus
- Academic Honesty: *Avoiding cheating at all cost.*



Chapter 1

Computer Abstractions and Technology

The Computer Revolution

- Progress in computer technology
 - Underpinned by Moore's Law
- Makes novel applications feasible
 - Computers in automobiles
 - Cell phones
 - Human genome project
 - World Wide Web
 - Search Engines
- Computers are pervasive (everywhere)

Classes of Computers

- Desktop computers
 - General purpose, variety of software
 - Subject to cost/performance tradeoff
- Server computers
 - Network based
 - High capacity, performance, reliability
 - Range from small servers to building sized
- Embedded computers
 - Hidden as components of systems
 - Stringent power/performance/cost constraints

Historical Perspective

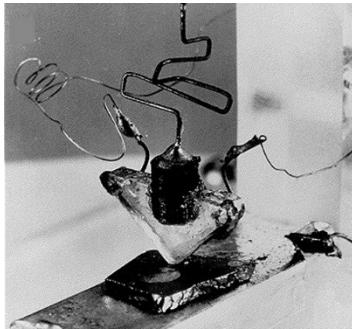
ENIAC (Electronic Numerical Integrator and Calculator) built in World War II was the first general purpose computer around 1946

- in Moore School of Electrical Engineering at the University of Pennsylvania, by John Mauchly and J. Presper Eckert
- Used for computing artillery firing tables
- 80 feet long by 8.5 feet high and several feet wide
- Each of the twenty 10 digit registers was 2 feet long
- Used 18,000 vacuum tubes
- Performed 1900 additions per second



Historical Perspective – Cont.

- UNIVAC I (Universal Automatic Computer) – the first commercial computer in USA
 - It correctly predicted the outcome of the 1952 presidential election

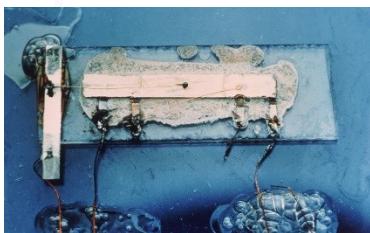


Transistor by W. Shockley, J. Bardeen, W. Brattain of Bell Lab. in 1947



Historical Perspective – Cont.

- IBM System/360, Model 40, 50, 65, and 75 (1964)



A integrated transistor with resistors and capacitors on a single semiconductor chip, which is a monolithic IC by Jack Kilby of TI in 1958



1.6MHz, 32KB ~ 256KB



2.0MHz, 128KB ~ 256KB



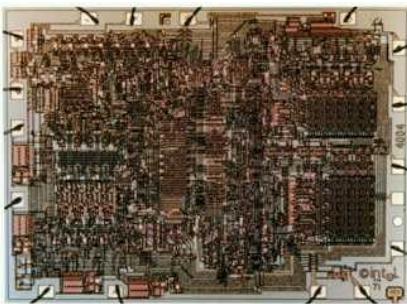
5.0MHz, 256KB ~ 1MB



5.1MHz, 256KB ~ 1MB

Historical Perspective – Cont.

- Cray-1 – the first commercial vector supercomputer, announced in 1976
 - The fastest computer for scientific applications
 - The best price/performance for scientific applications



The first microprocessor Intel 4004 in 1971

1. 108 KHz, 0.06 MIPS
2. 2300 transistors (10 microns)
3. Bus width: 4 bits
4. Memory addr.: 640 bytes
5. For Busicom calculator (original commission was 12 chips)



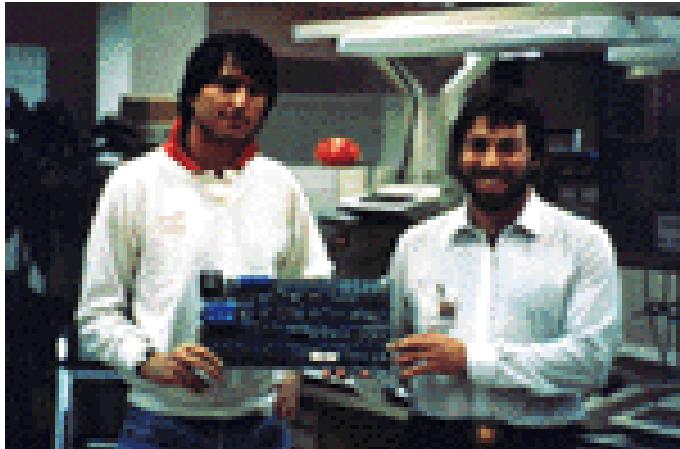
Historical Perspective – Cont.

- Xerox Alto (by Xerox Palo Alto) – the primary inspiration for the modern desktop computer in 1972
 - A bit-mapped graphic display
 - A mouse
 - A local-area network
 - A window-based user interface WYSIWYG (What You See Is What You Get)



Historical Perspective – Cont.

- Apple I by Steve Wozniak in 1976 at Palo Alto
- Apple II by Steve Jobs and Steve Wozniak using a Motorola 6502 8-bit CPU



Historical Perspective – Cont.

- IBM PC / Compatible PC
- PC DOS (Disk Operating System)
 - CP/M, IBM DOS, MS DOS



Microsoft Corporation, 1978

They Made America: Two Centuries of Innovators from the Steam Engine to the Search Engine (2004)
ISBN 0-316-27766-5 by Harold Matthew Evans



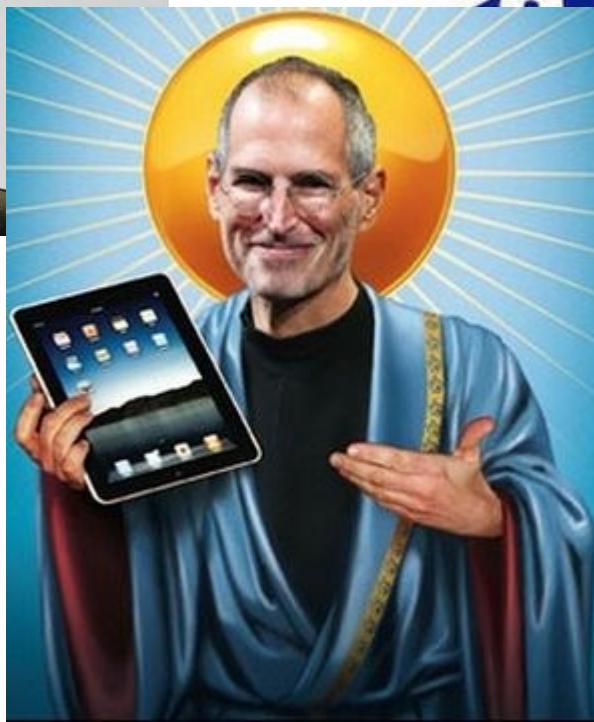
Historical Perspective – Cont.

■ Embedded computers



Historical Perspective – Cont.

■ AS Dynasties - Apple vs. Samsung



Copyright by http://www.theregister.co.uk/Wrap/ipad2_story_wrapup/



Historical Perspective – Cont.

- Burn down to fire for IP

-reference screen shots



iphone 4.0 screen shot
(badges,background)



iphone 4.0 screen shot
(folder view)



Android 2.1 screen shot



Samsung 2010 icons

Historical Perspective – Cont.

陪審團判定三星侵害的蘋果專利

軟體／設計專利	是否侵權
1. 當使用者將畫面滾動至邊界時畫面會回彈的功能	侵權
2. 滾動；雙指縮放	侵權
3. 點擊放大並集中	侵權
4. iPhone的正面外觀、螢幕與喇叭槽	侵權
5. iPhone的正面外觀、圓角和邊框	侵權
6. 主頁面的圖示排列	侵權
7. 平板電腦設計，包括iPad的長方形外觀和圓角	未侵權

資料來源：紐約時報

余曉惠／製表



Historical Perspective – Cont.

I. Overall Comments

Confidential

- A total of 126 issues were found, and 27 new issues were found in S1 (21.4%) and there were 99 issues that overlap with Lismore
- Basic Functions take up the largest percentage of the issues with 21.4%, followed by Visual Interaction Effect (17.5%), Browsing and Messaging (16.7% respectively)

Items	i-Phone	S1
Basic Function (27 items)	Effective and efficient use of space Ex) Shows keypad/font and Calendar schedule in a large view	Has poor use of space for a large LCD Ex) Keypad and font are small, and schedule list field is narrow
Browsing (21 items)	Edit and delete functions are appropriately placed Ex) YouTube search history deletion and addition of countries in weather application Cut & Paste of contents function is supported	There are no edit/delete functions, and there are unnecessary functions Ex) YouTube search history deletion, addition of countries in weather application, and Cut & Paste of contents functions are not supported
Connectivity (19 items)	Easily synchronized with other devices Ex) Can switch to BT while playing music Wi-Fi set up can be configured in one screen	Synchronizing with other devices is complicated and difficult Ex) Can't switch to BT while playing music; Wi-Fi ON/OFF and Setting screens are separately implemented
Messaging (21 items)	Received messages are easily recognized and accessed Ex) The number of received messages is indicated on the email icon Easy to move to previous/next e-mail	Receiving events is difficult to recognize and access Ex) Difficult to recognize because the icon for received E-Mail is black No move button to move to the next e-mail
Multimedia (16 items)	Various convenience functions are offered during playback and editing Ex) Fine tuning during music play and picture Cut & Paste functions are supported Edit function is supported for large-size video files	Lack detailed convenience functions Ex) Music fine tuning and picture copying are not supported Can't attach the desired parts from a large-capacity video file
Visual Interaction Effect (22 items)	Fun factor is increased by adding Effects to even little parts Ex) Effect for saving mails, Screen transition effect for maps	Effects are inserted only for major menus Ex) No effects for moving into folders and for map screen transitions



Highest-Clock-Frequency CPU (2011)

- 8 cores AMD FX CPU
 - 8.428 GHz
 - Without any limit to overshooting
- Currently the clock rates of 3GHz to 4GHz are most common

Historical Perspective – Cont.

- zEnterprise System – Main Frame

- 96 5.2GHz microprocessors.
- 50 billion instructions per second.
- Ever fastest (5.2GHz) microprocessor in the world z196.



Historical Perspective – Cont.

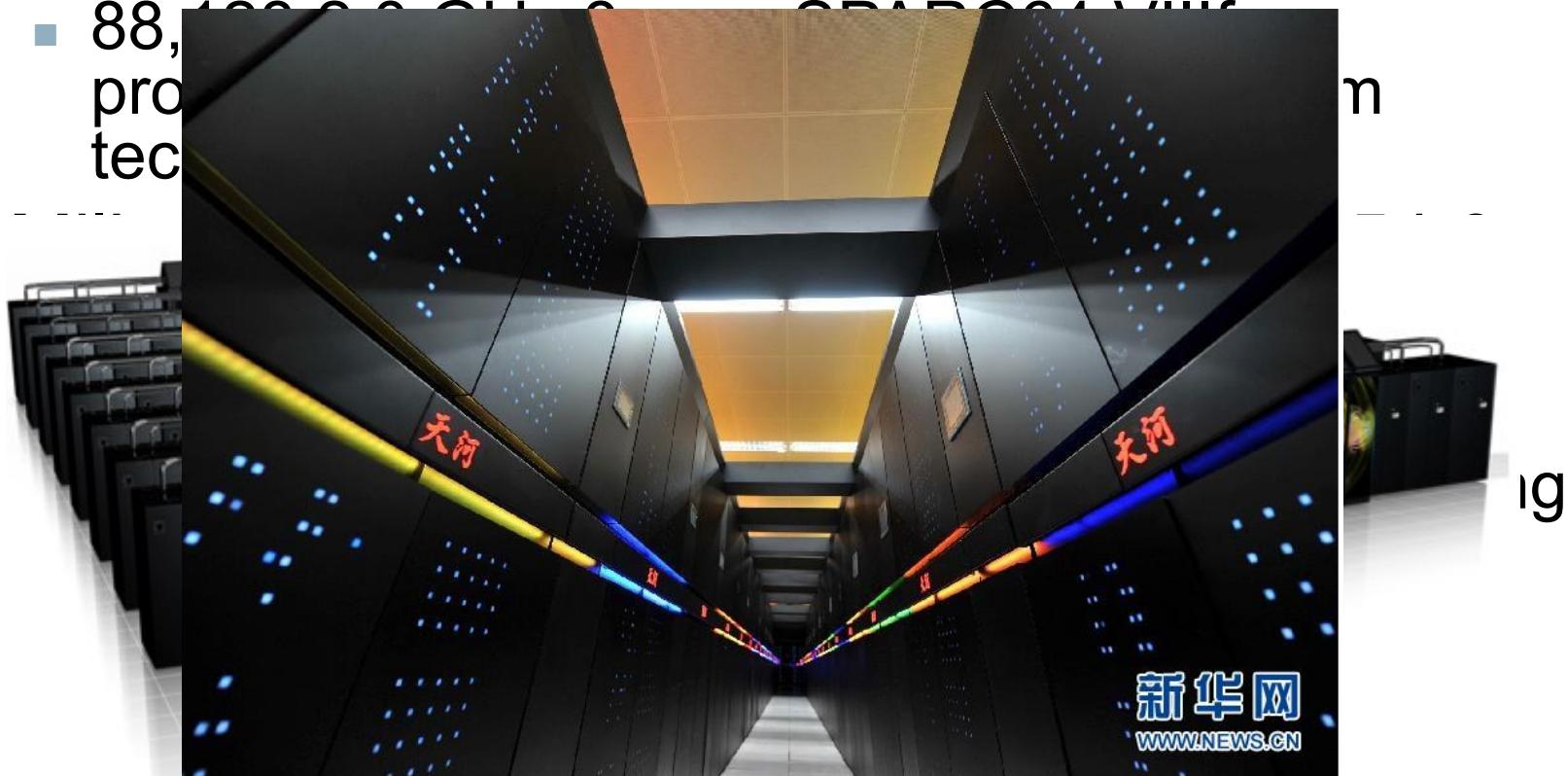
Roadrunner

- 10^{15}
- 6 billion flops for 4 laptops
-



Historical Perspective – Cont.

- K Computer – the first one to reach 10^{16}
 - 10.51 PFLOPS (peta = 10^{15})
 - 88,000 CPU cores, 160 Gbps interconnect



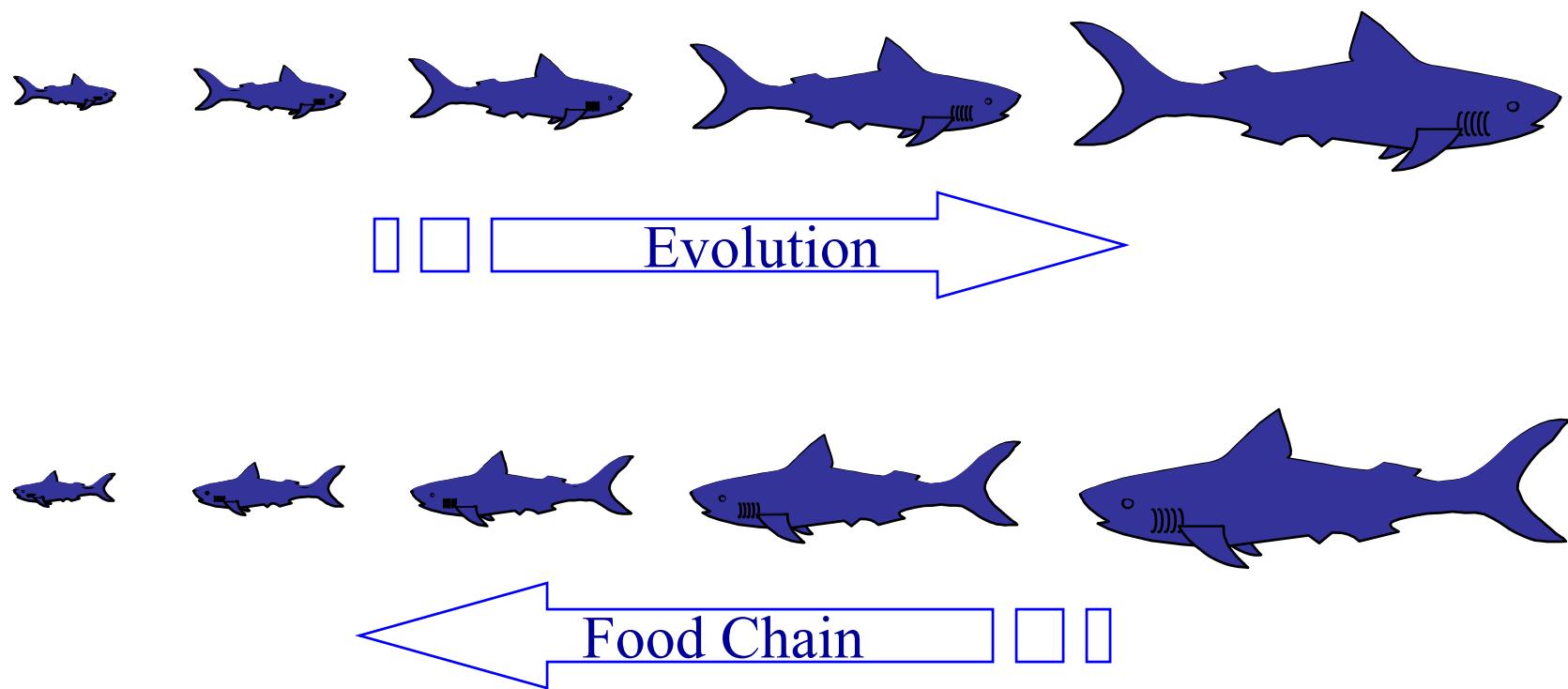
TOP500 in 2017

Top 10 positions of the 50th TOP500 in November 2017^[15]

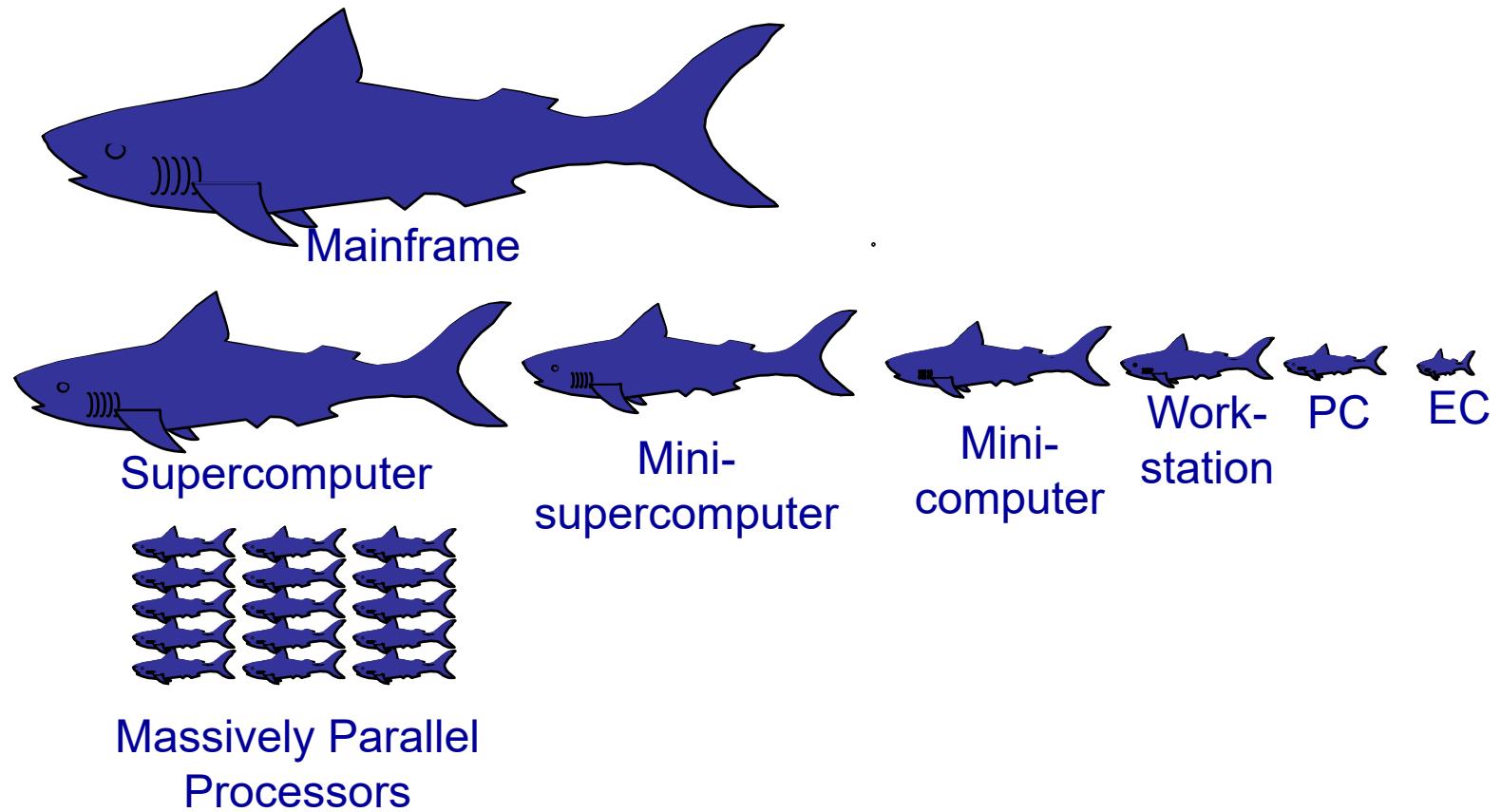
Rank ↴	Rmax Rpeak ↴ (PFLOPS)	Name ↴	Model ↴	Processor ↴	Interconnect ↴	Vendor ↴	Site country, year	Operating system ↴
1	93.015 125.436	<i>Sunway TaihuLight</i>	Sunway MPP	SW26010	Sunway ^[16]	NRCPC	National Supercomputing Center in Wuxi China, 2016 ^[16]	Linux (Raise)
2	33.863 54.902	<i>Tianhe-2</i>	TH-IVB-FEP	Xeon E5-2692, Xeon Phi 31S1P	TH Express-2	NUDT	National Supercomputing Center in Guangzhou China, 2013	Linux (Kylin)
3	19.590 25.326	<i>Piz Daint</i>	Cray XC50	Xeon E5-2690v3, Tesla P100	Aries	Cray	Swiss National Supercomputing Centre Switzerland, 2016	Linux (CLE)
4	19.136 28.192	<i>Gyoukou</i>	ZettaScaler-2.2 HPC system	Xeon D-1571, PEZY-SC2	Infiniband EDR	ExaScaler	Japan Agency for Marine-Earth Science and Technology Japan, 2017	Linux (CentOS)
5	17.590 27.113	<i>Titan</i>	Cray XK7	Opteron 6274, Tesla K20X	Gemini	Cray	Oak Ridge National Laboratory United States, 2012	Linux (CLE, SLES based)
6	17.173 20.133	<i>Sequoia</i>	Blue Gene/Q	A2	Custom	IBM	Lawrence Livermore National Laboratory United States, 2013	Linux (RHEL and CNK)
7	14.137 43.902	<i>Trinity</i>	Cray XC40	Xeon E5-2698v3, Xeon Phi	Aries	Cray	Los Alamos National Laboratory United States, 2015	Linux (CLE)
8	14.015 27.881	<i>Cori</i>	Cray XC40	Xeon Phi 7250	Aries	Cray	National Energy Research Scientific Computing Center United States, 2016	Linux (CLE)
9	13.555 24.914	<i>Oakforest-PACS</i>	Fujitsu	Xeon Phi 7250	Intel Omni-Path	Fujitsu	Kashiwa, Joint Center for Advanced High Performance Computing Japan, 2016	Linux
10	10.510 11.280	<i>K computer</i>	Fujitsu	SPARC64 VIIIfx	Tofu	Fujitsu	Riken, Advanced Institute for Computational Science (AICS)	Linux



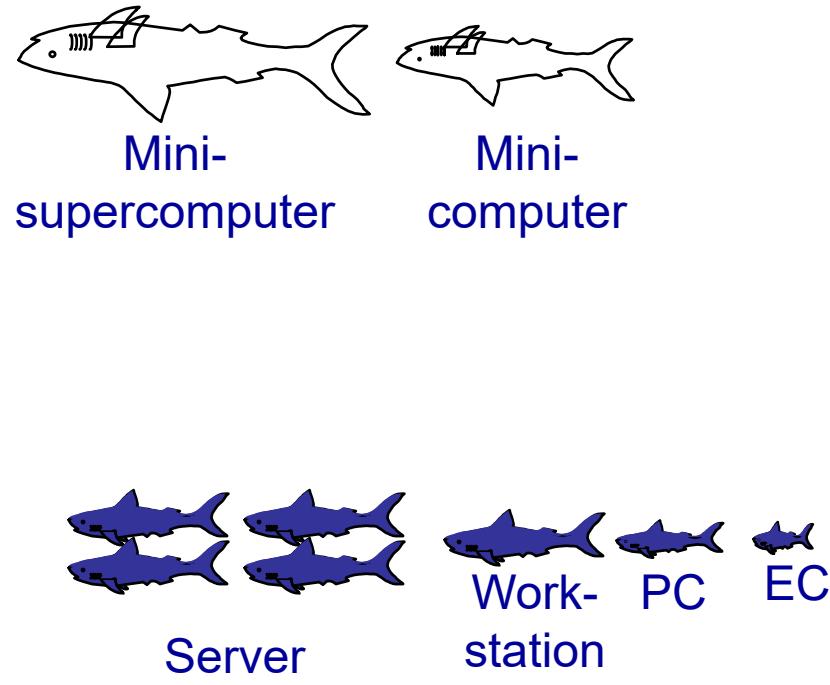
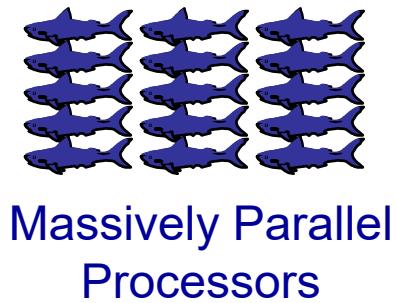
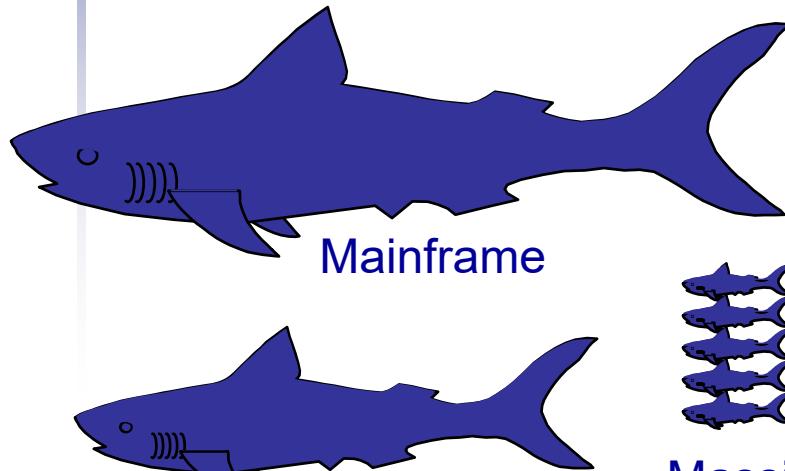
Evolution & Food Chain



1988 Computer Food Chain

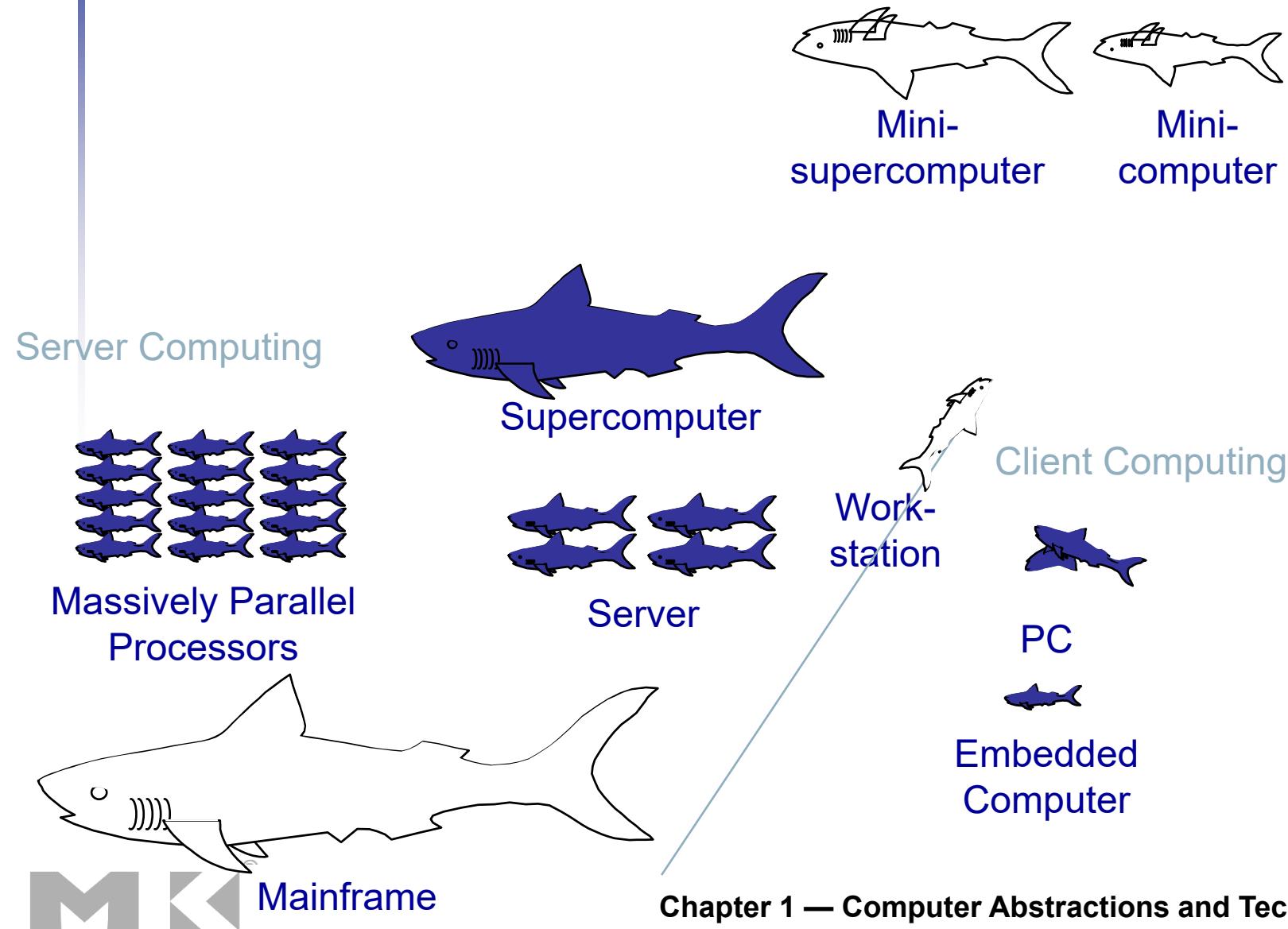


1998 Computer Food Chain

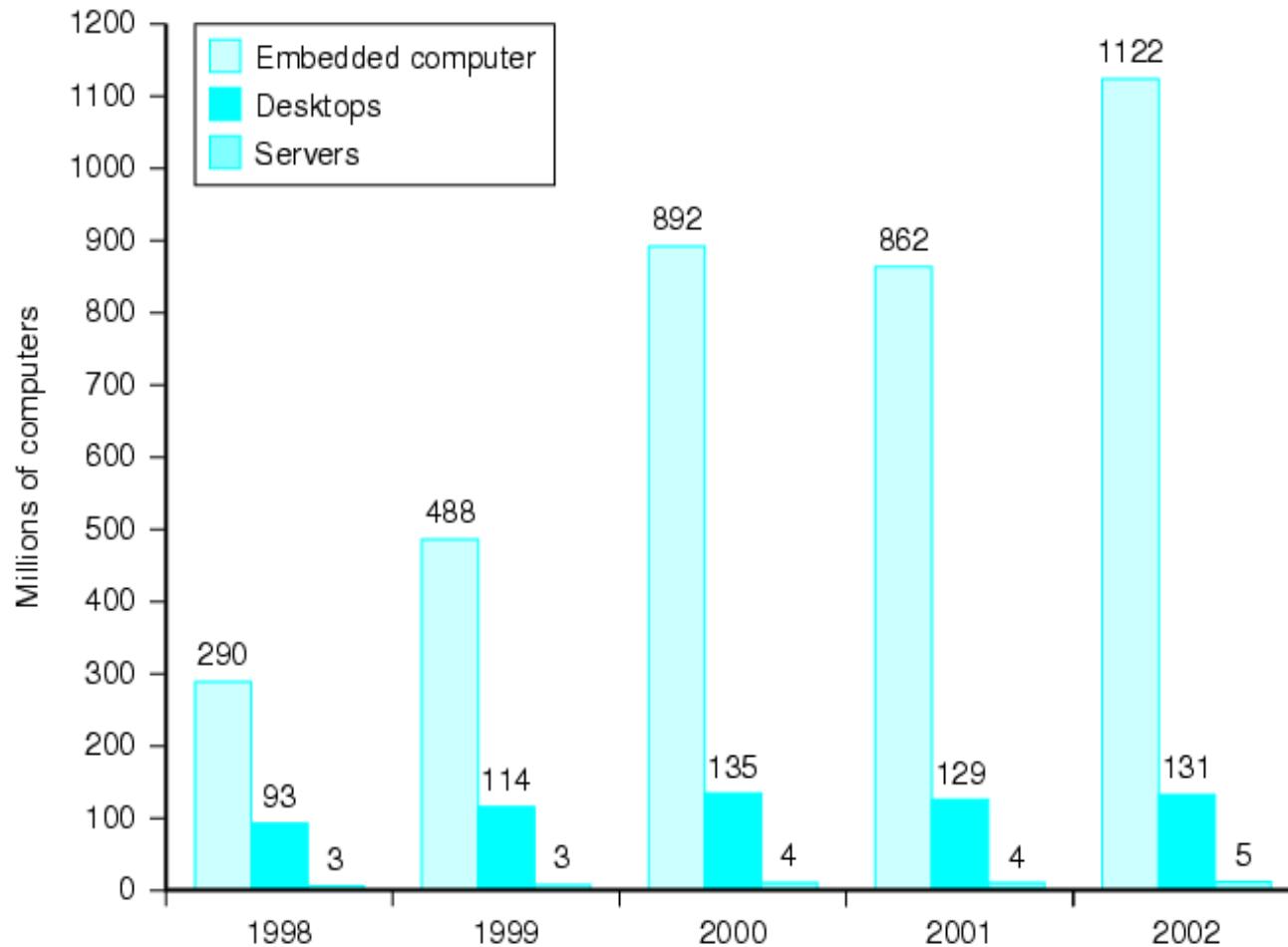


Are mini-computers eaten by mainframe or supercomputer?

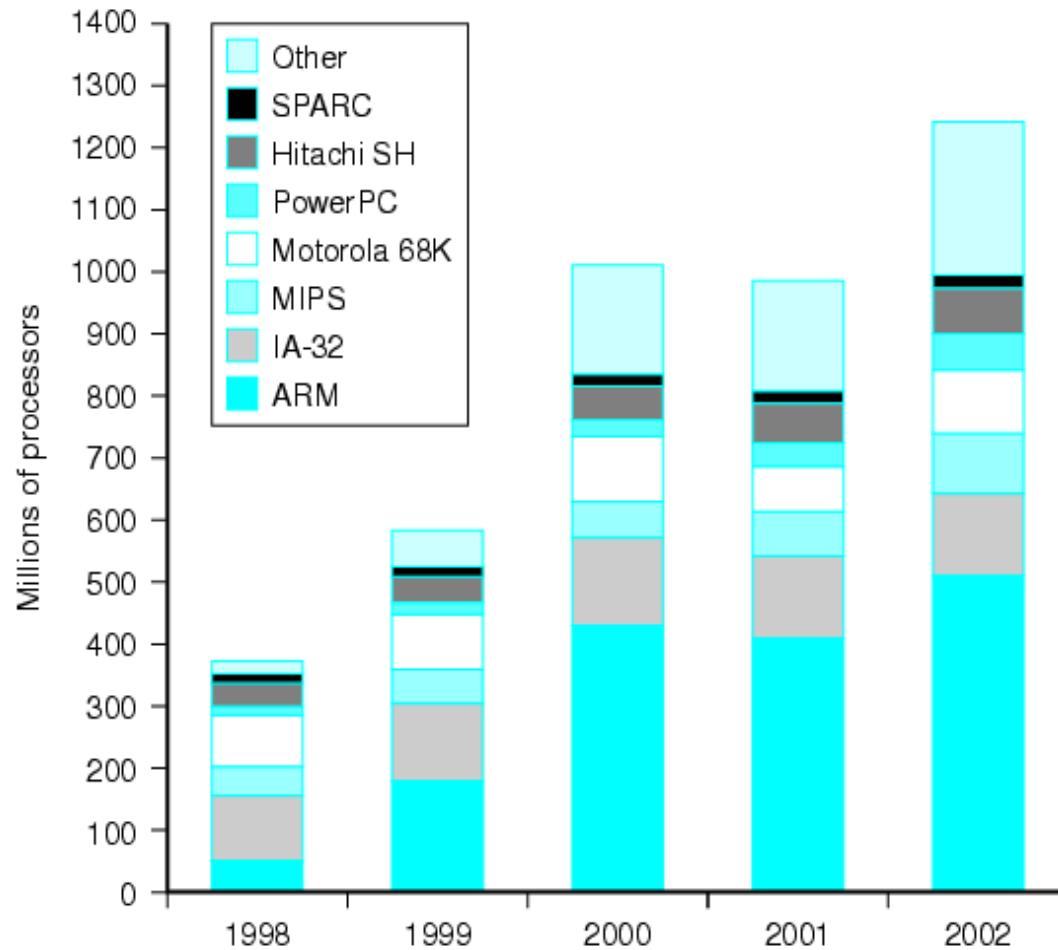
2007 Computer Food Chain



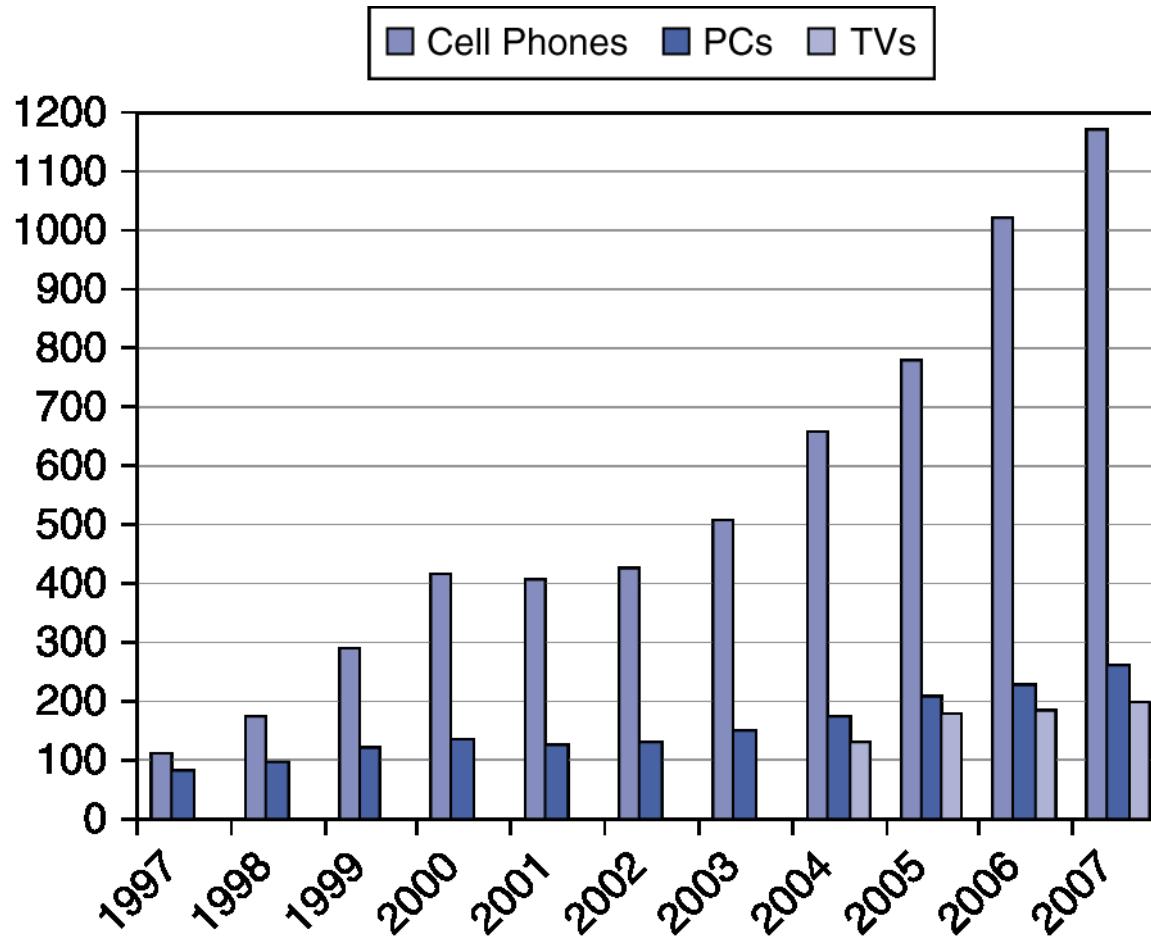
Sales Statistics for Computers



Sales Statistics for Microprocessors



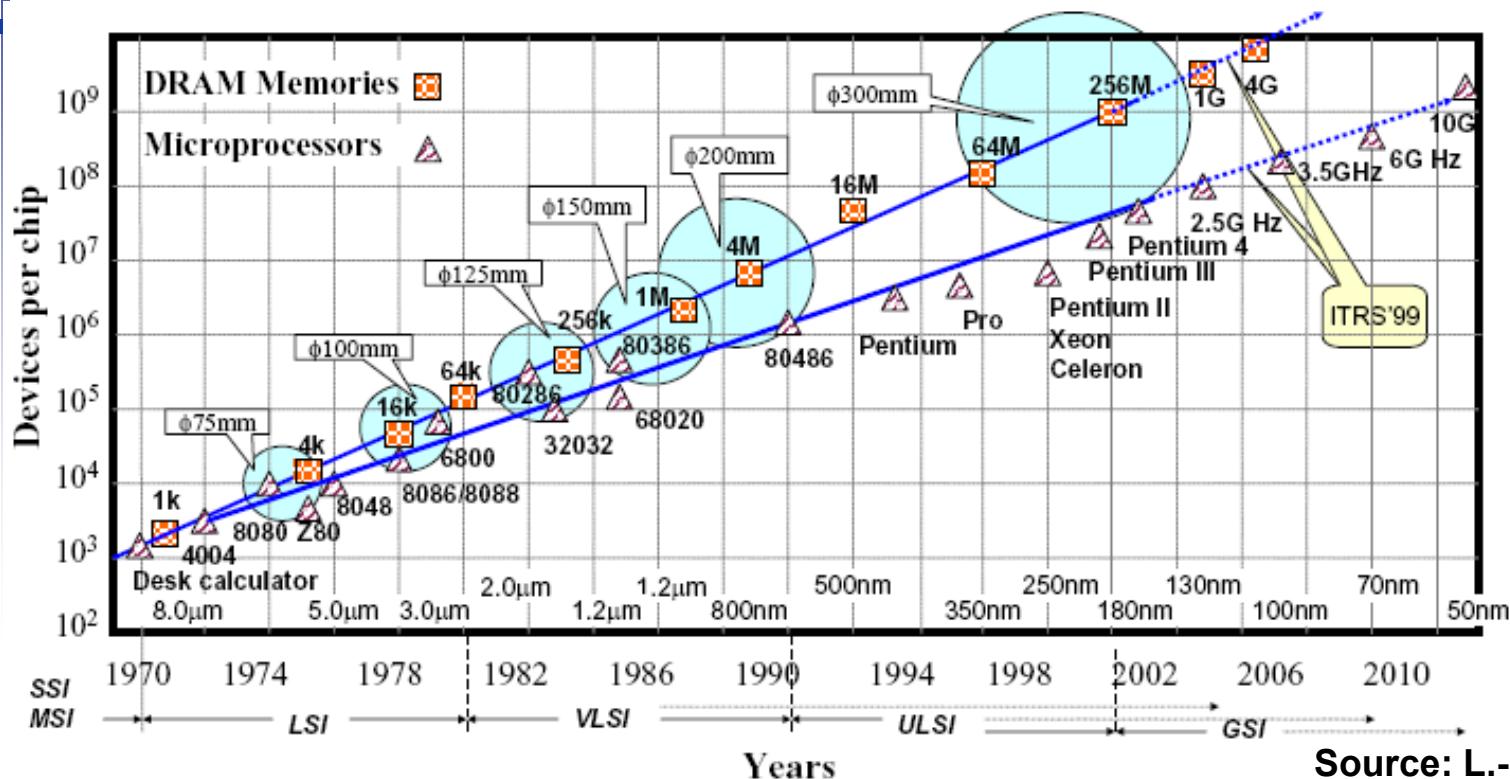
The Processor Market



Why These Changes?

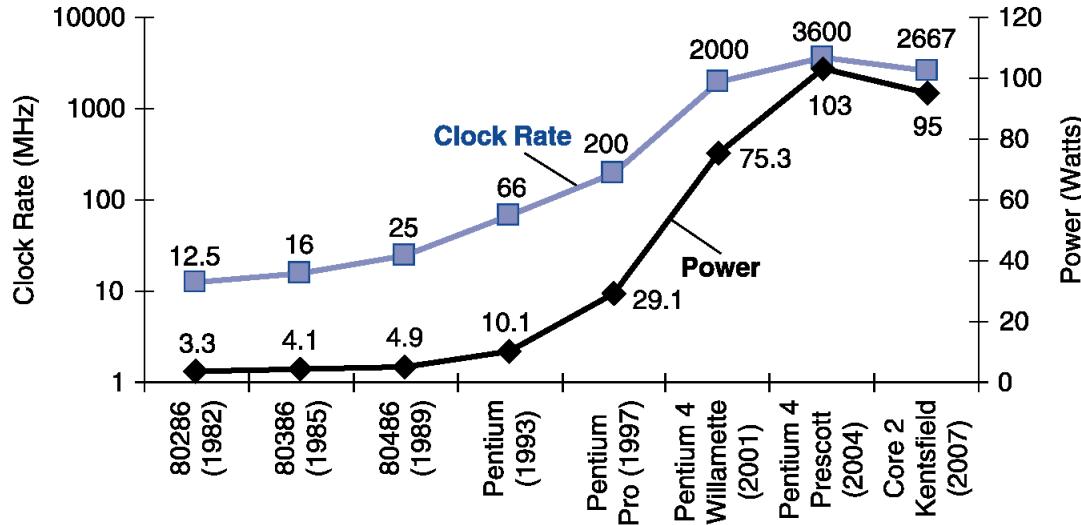
- Continuous advances in IC manufacturing technology, design methodology, and computer-aided design tools allow embedded computers to have more computation power
 - ECL → CMOS
 - Shrinkage of feature size → increasing transistor numbers in a chip
 - System on Chip (SOC) design methodology
 - Manual design → electrical design automation by CAD tools
- Increasing progress in communication technology
 - LAN → WAN
 - Wire → 3G wireless (64Kbps ~ 384Kbps)
 - Diversify the applications of embedded computers

Moore's Law and Performance Comparison



Year	Technology used in computers	Relative performance/unit cost
1951	Vacuum tube	1
1965	Transistor	35
1975	Integrated circuit	900
1995	Very large scale integrated circuit	2400000
2005	Ultra large scale integrated circuit	6200000000

Power Trends



- In CMOS IC technology

$$\text{Power} = \text{Capacitive load} \times \text{Voltage}^2 \times \text{Frequency}$$

x30

5V → 1V

x1000

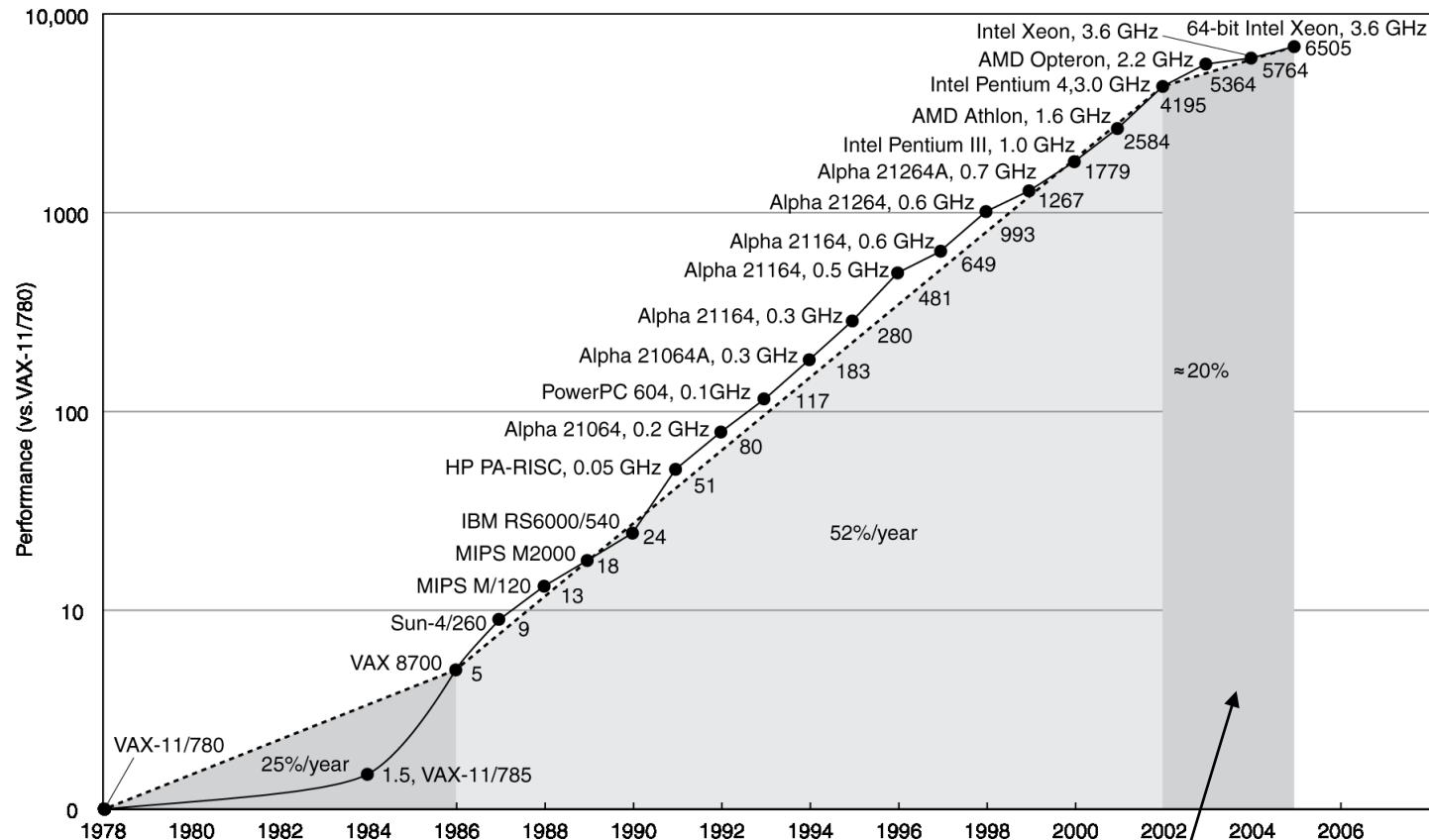
Reducing Power

- Suppose a new CPU has
 - 85% of capacitive load of old CPU
 - 15% voltage and 15% frequency reduction

$$\frac{P_{\text{new}}}{P_{\text{old}}} = \frac{C_{\text{old}} \times 0.85 \times (V_{\text{old}} \times 0.85)^2 \times F_{\text{old}} \times 0.85}{C_{\text{old}} \times V_{\text{old}}^2 \times F_{\text{old}}} = 0.85^4 = 0.52$$

- The power wall
 - We can't reduce voltage further
 - We can't remove more heat
- How else can we improve performance?

Uniprocessor Performance



Constrained by power, instruction-level parallelism,
memory latency

Multiprocessors

- Multicore microprocessors
 - More than one processor per chip
- Requires explicitly parallel programming
 - Compare with instruction level parallelism
 - Hardware executes multiple instructions at once
 - Hidden from the programmer
 - Hard to do
 - Programming for performance
 - Load balancing
 - Optimizing communication and synchronization

Power in AI chip

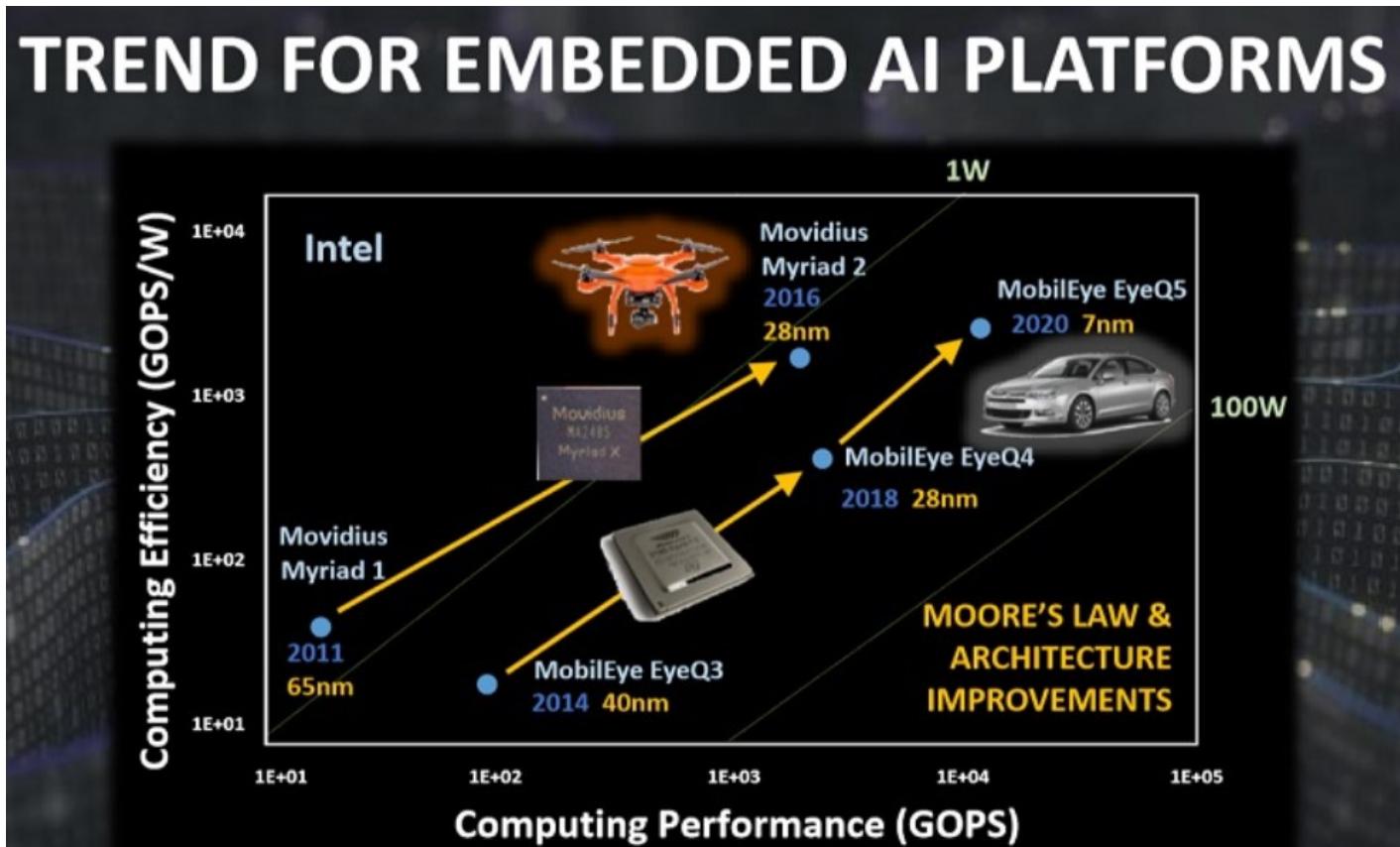


Figure source is from ISSCC 2018

Power in AI chip

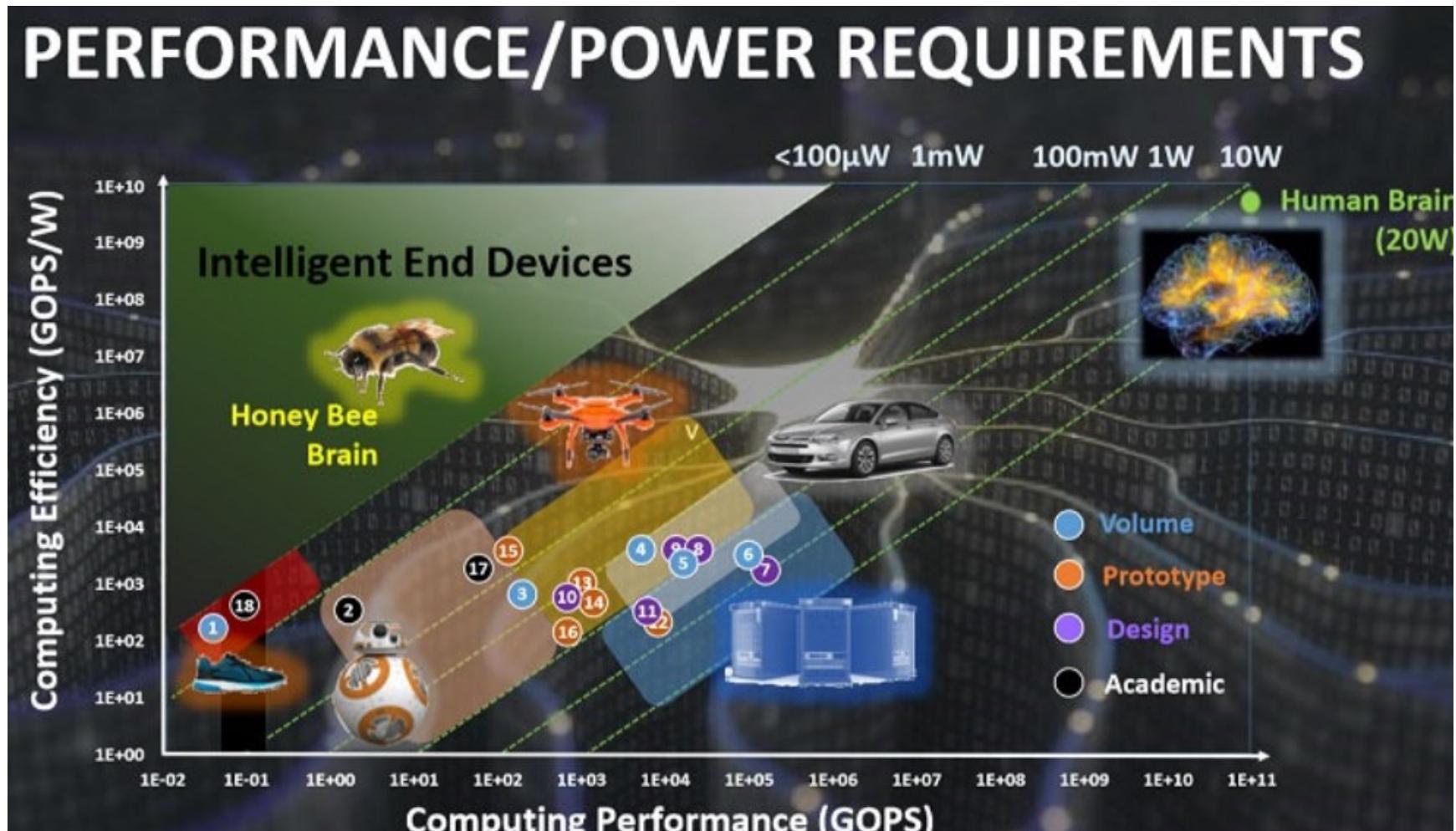
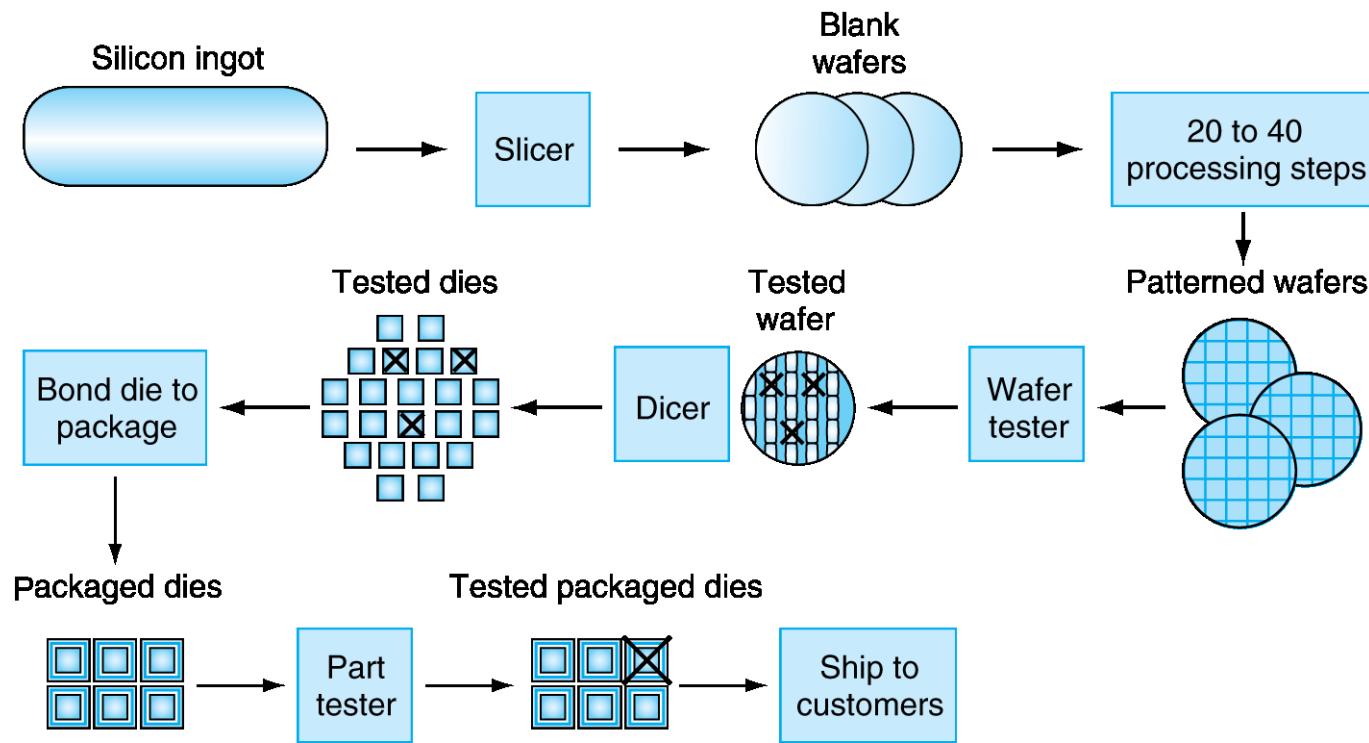


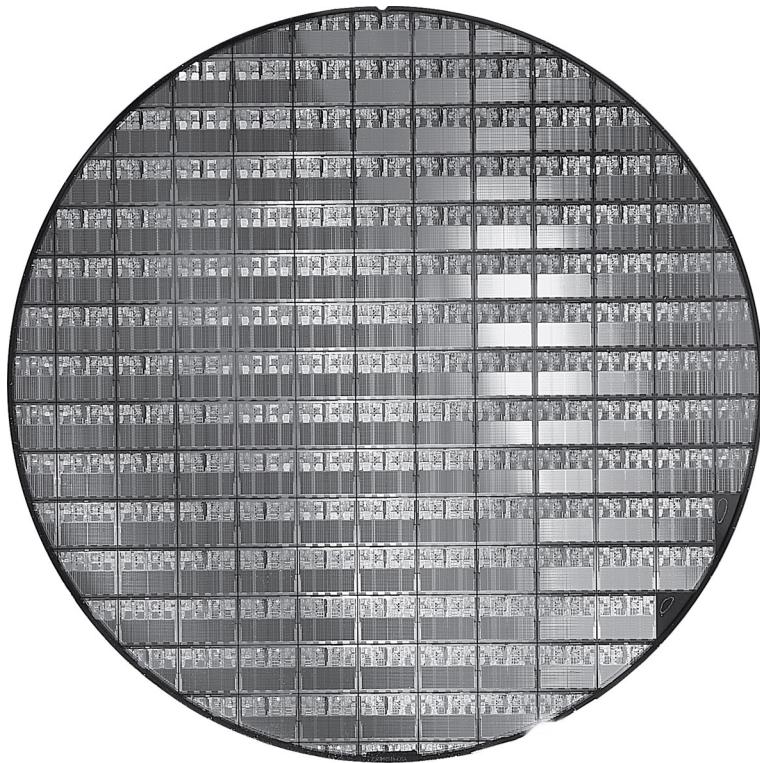
Figure source is from ISSCC 2018

Manufacturing ICs



- Yield: proportion of good dies per wafer

AMD Opteron X2 Wafer



- X2: 300mm wafer, 117 chips, 90nm technology
- X4: 45nm technology

Integrated Circuit Cost

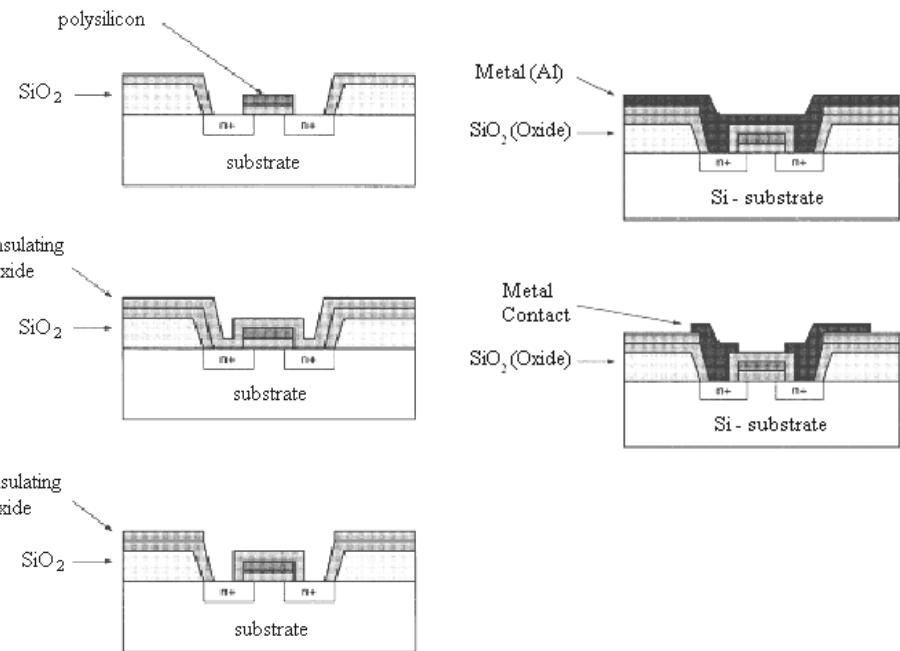
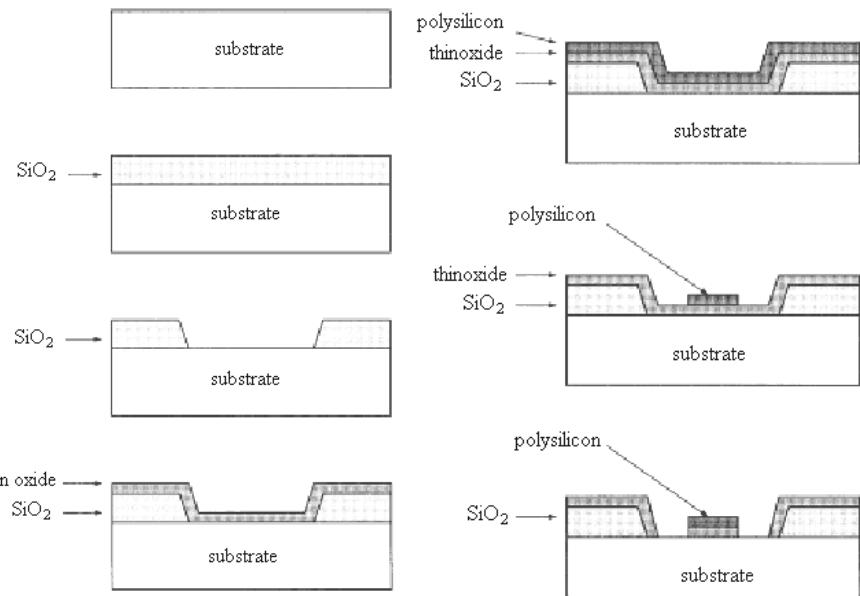
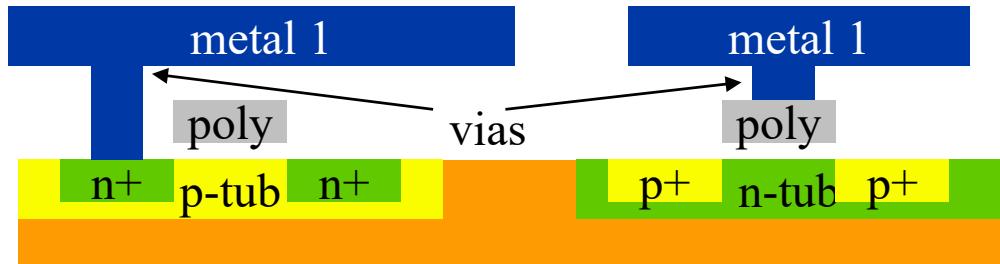
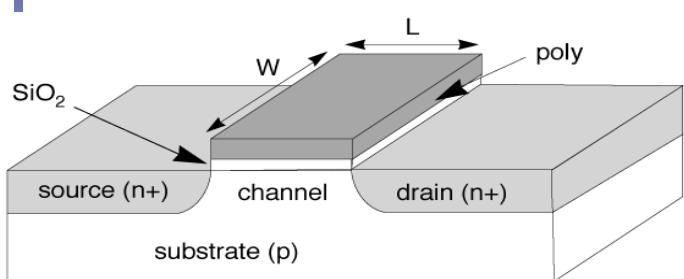
$$\text{Cost per die} = \frac{\text{Cost per wafer}}{\text{Dies per wafer} \times \text{Yield}}$$

$$\text{Dies per wafer} \approx \text{Wafer area}/\text{Die area}$$

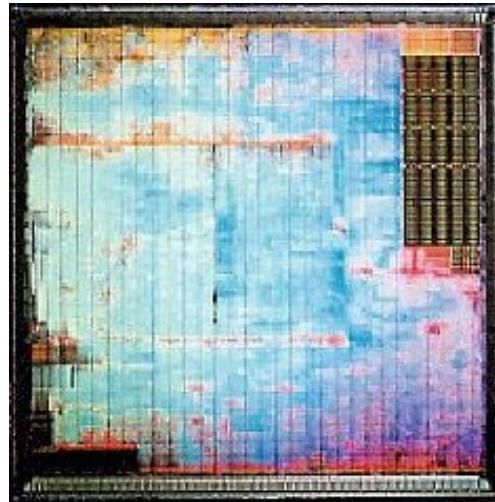
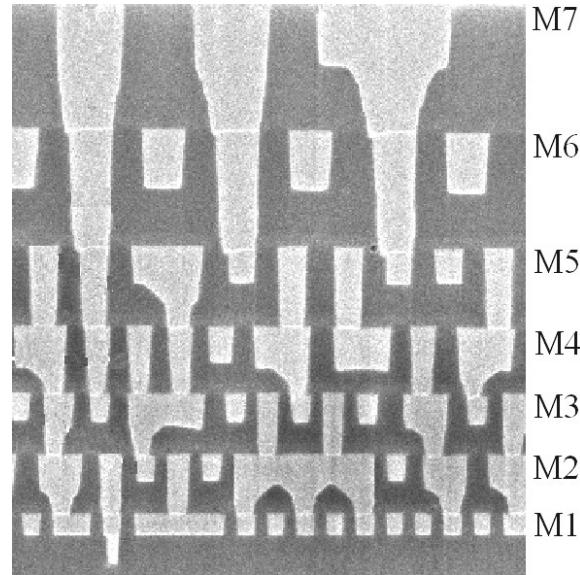
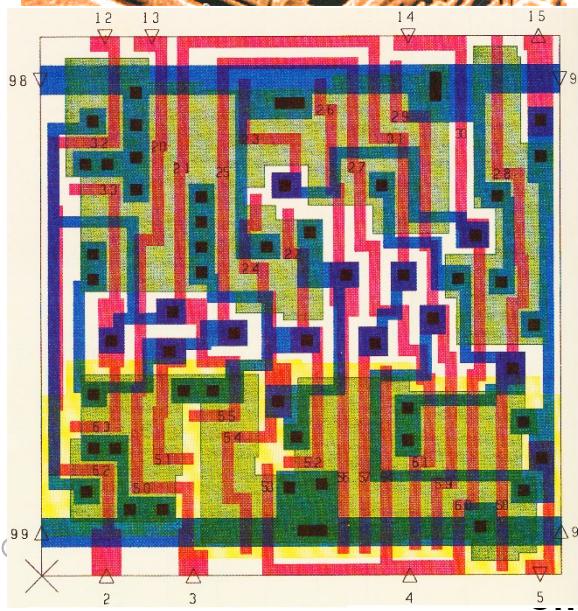
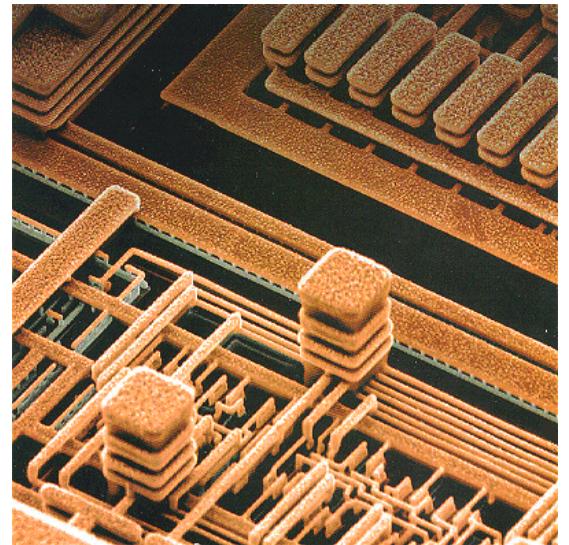
$$\text{Yield} = \frac{1}{(1 + (\text{Defects per area} \times \text{Die area}/2))^2}$$

- Nonlinear relation to area and defect rate
 - Wafer cost and area are fixed
 - Defect rate determined by manufacturing process
 - Die area determined by architecture and circuit design

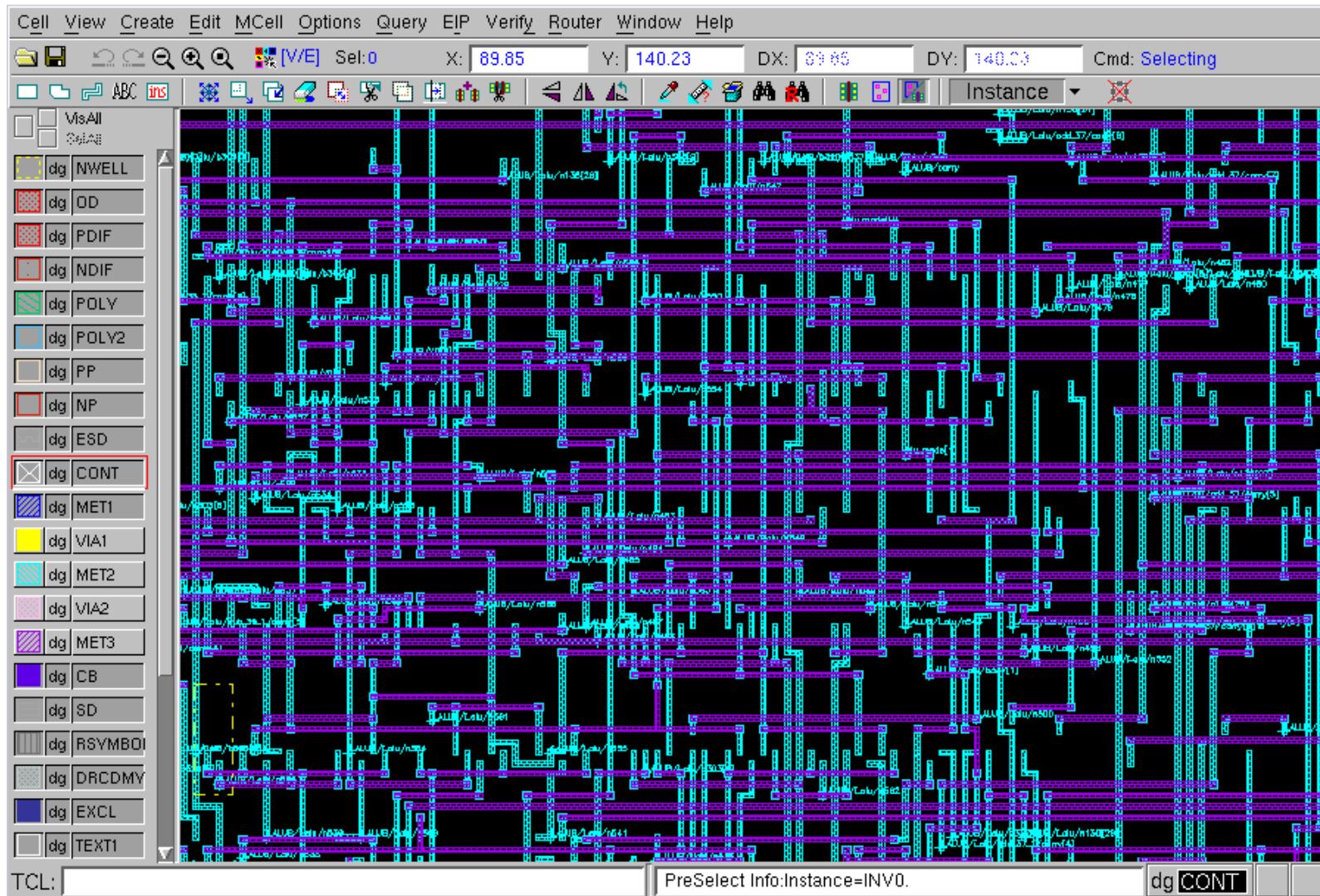
Device Layout



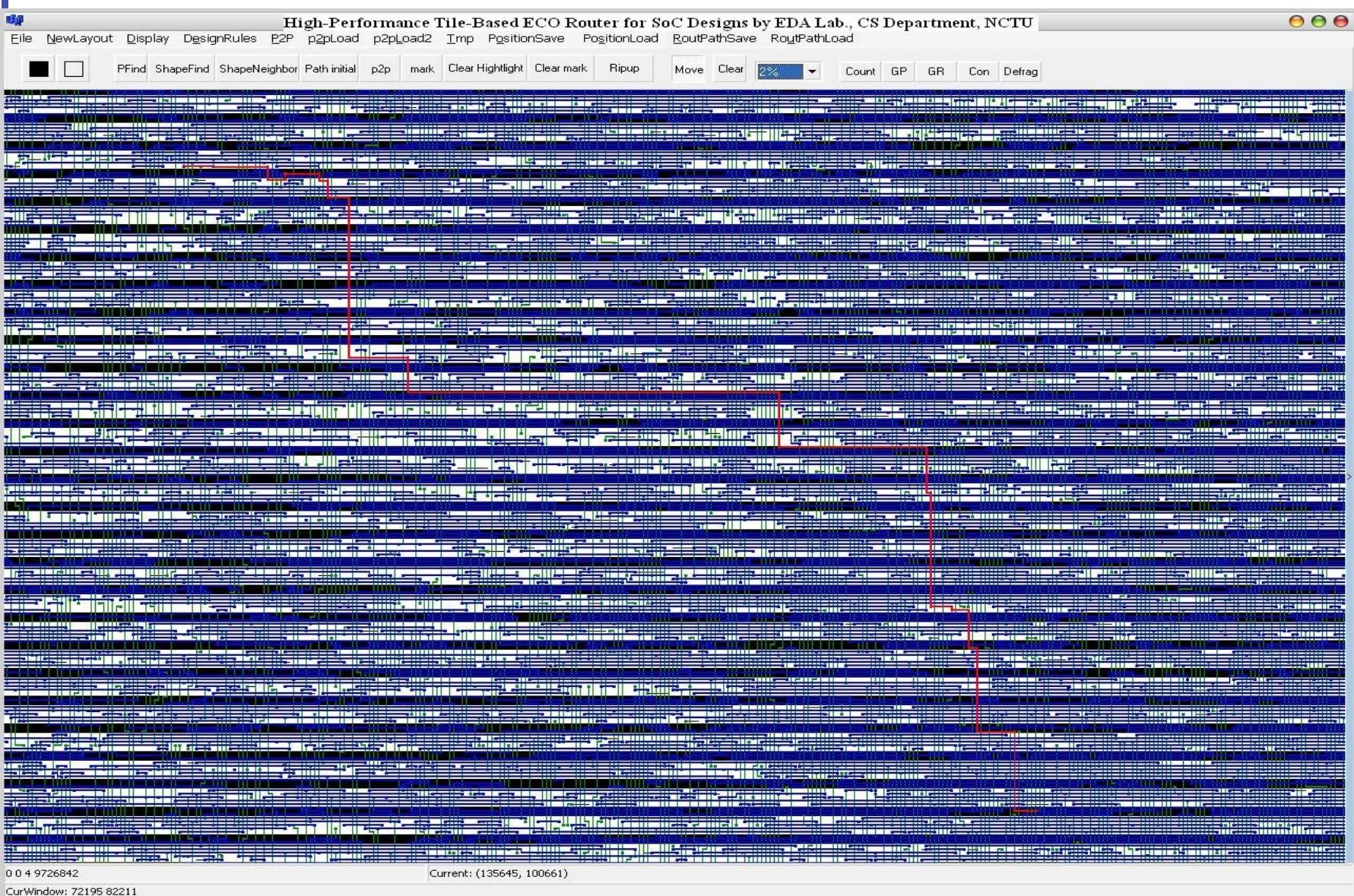
Interconnection Layout



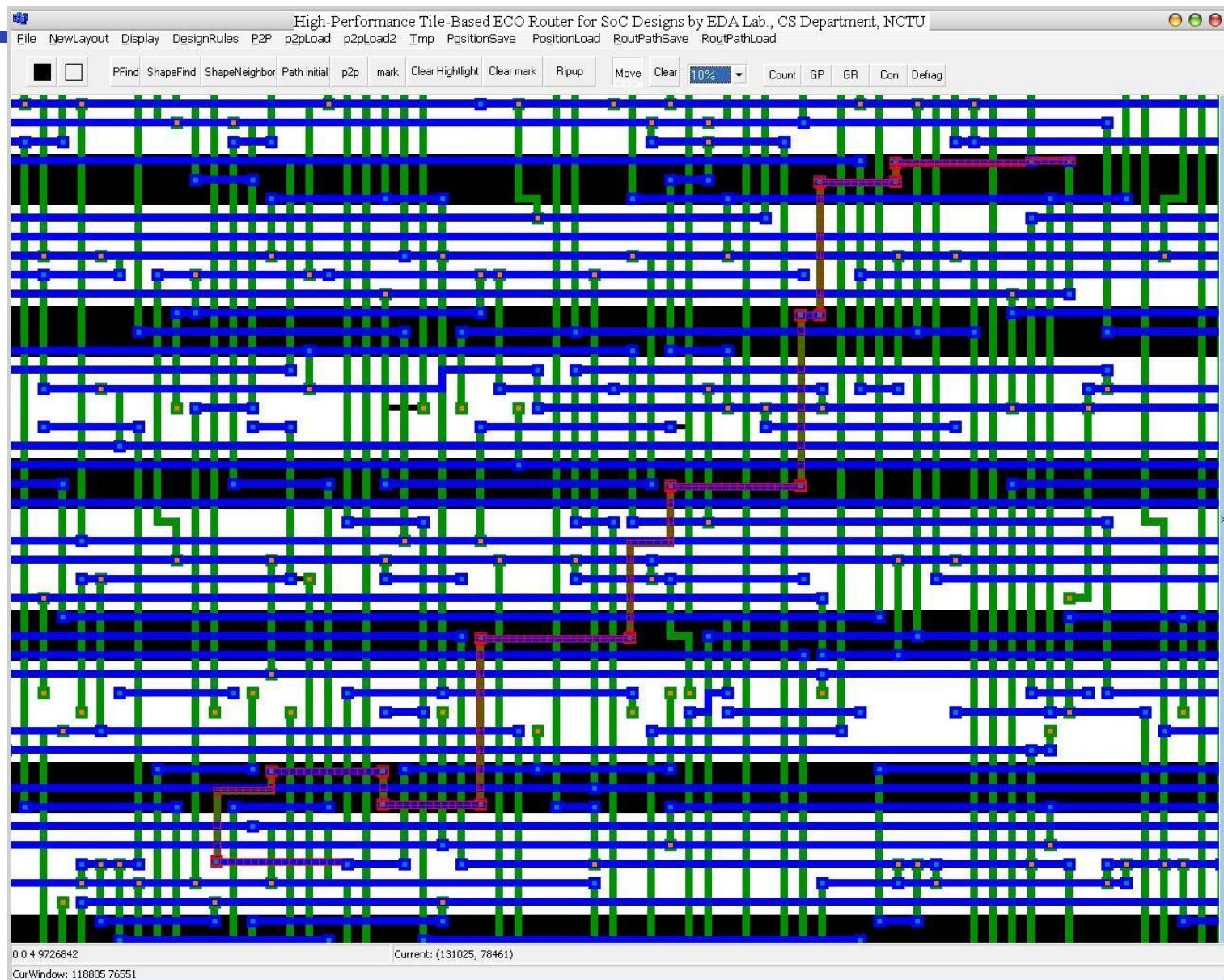
Industry Router



NCTU CS-EDA Lab Router



NCTU CS-EDA Lab Router



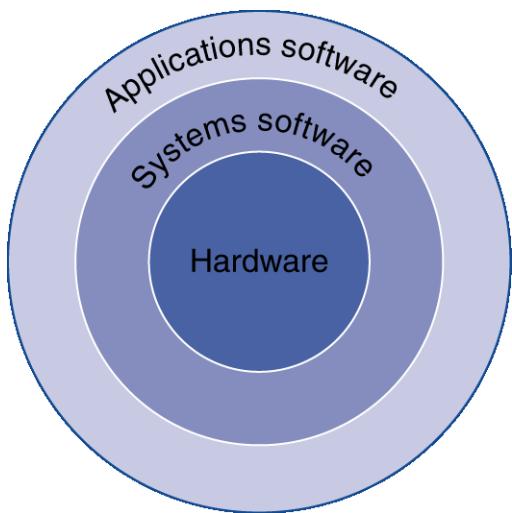
What You Will Learn

- How programs are translated into the machine language
 - And how the hardware executes them
- The hardware/software interface
- What determines program performance
 - And how it can be improved
- How hardware designers improve performance
- What is parallel processing

Understanding Performance

- Algorithm
 - Determines number of operations executed
- Programming language, compiler, architecture
 - Determine number of machine instructions executed per operation
- Processor and memory system
 - Determine how fast instructions are executed
- I/O system (including OS)
 - Determines how fast I/O operations are executed

Below Your Program



- Application software
 - Written in high-level language
- System software
 - Compiler: translates HLL code to machine code
 - Operating System: service code
 - Handling input/output
 - Managing memory and storage
 - Scheduling tasks & sharing resources
- Hardware
 - Processor, memory, I/O controllers

Levels of Program Code

■ High-level language

- Level of abstraction closer to problem domain
 - Provides for productivity and portability

■ Assembly language

- ## ■ Textual representation of instructions

■ Hardware representation

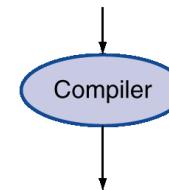
- Binary digits (bits)
 - Encoded instructions and data

High-level
language
program
(in C)

```

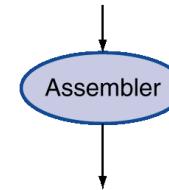
swap(int v[], int k)
{int temp;
    temp = v[k];
    v[k] = v[k+1];
    v[k+1] = temp;
}

```



Assembly language program (for MIPS)

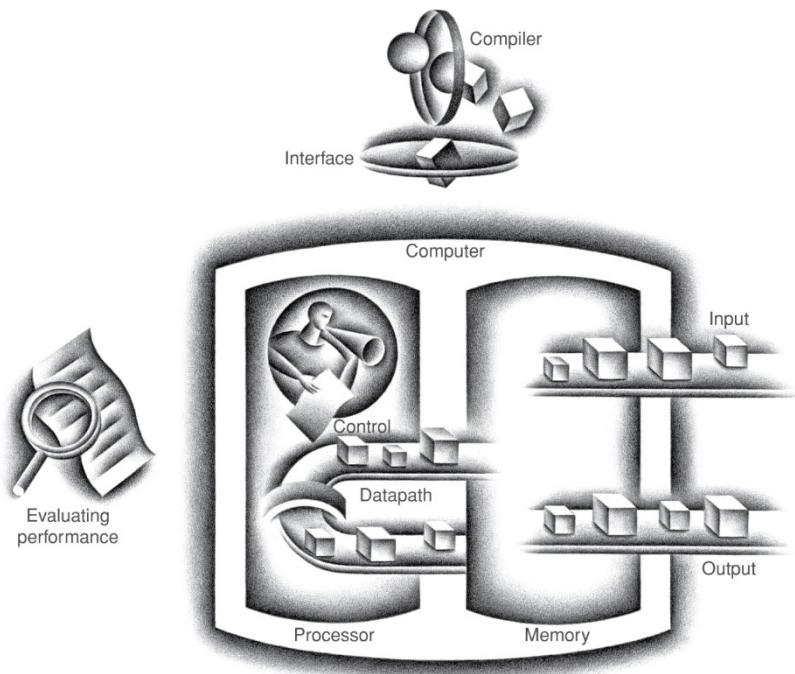
```
swap:  
    mul $2, $5,4  
    add $2, $4,$2  
    lw   $15, 0($2)  
    lw   $16, 4($2)  
    sw   $16, 0($2)  
    sw   $15, 4($2)  
    jr   $31
```



Binary machine
language
program
(for MIPS)

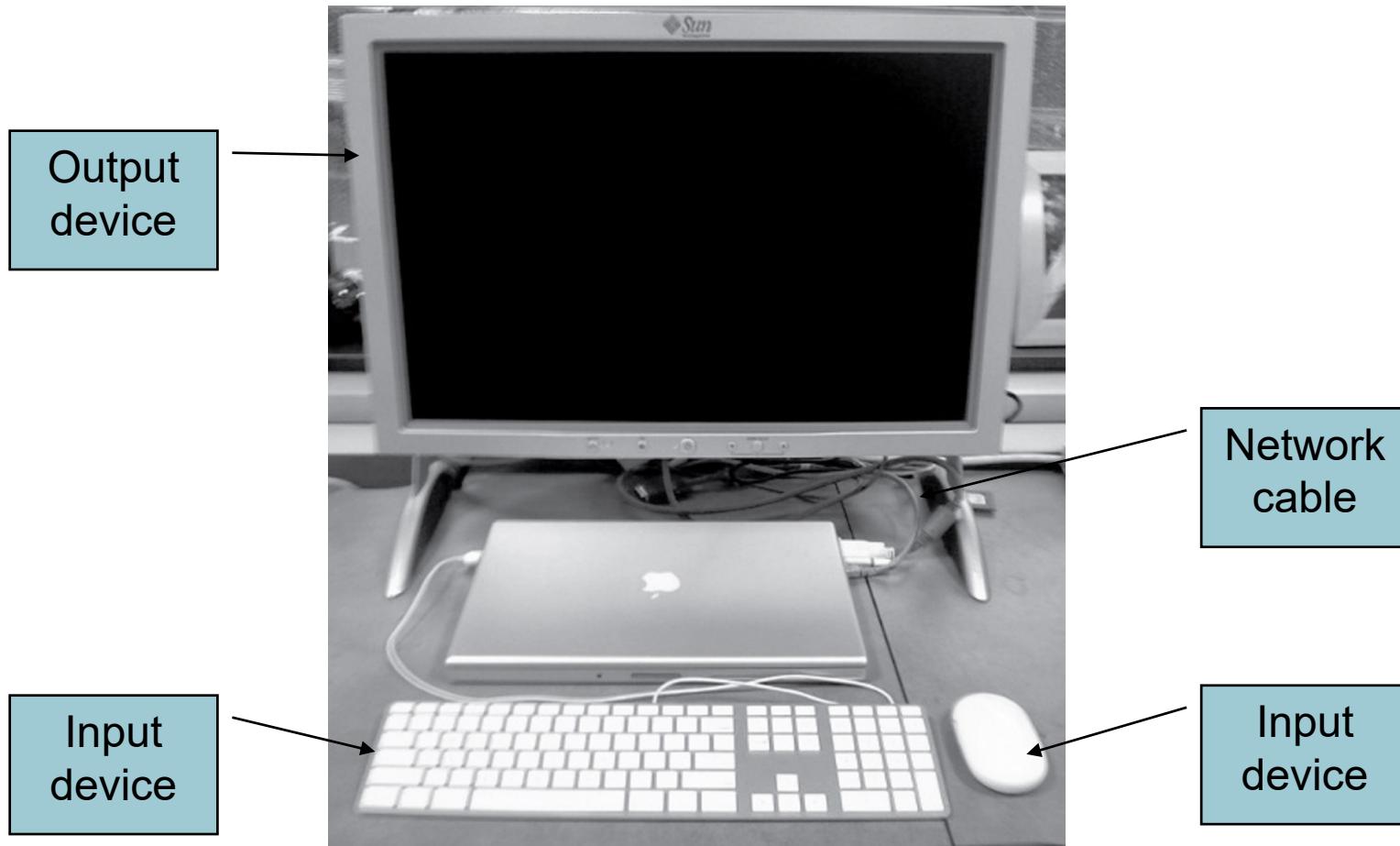
Components of a Computer

The BIG Picture



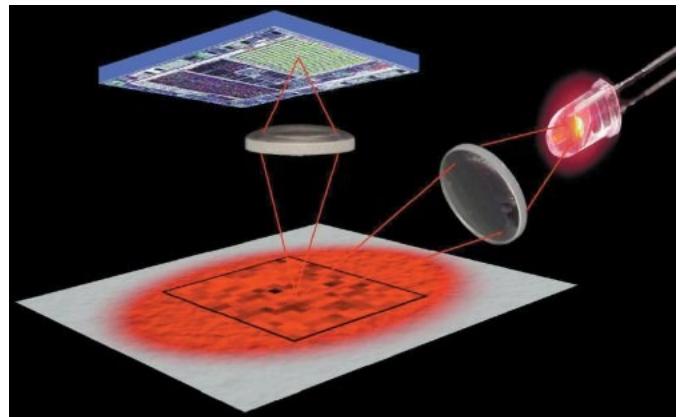
- Same components for all kinds of computer
 - Desktop, server, embedded
- Input/output includes
 - User-interface devices
 - Display, keyboard, mouse
 - Storage devices
 - Hard disk, CD/DVD, flash
 - Network adapters
 - For communicating with other computers

Anatomy of a Computer



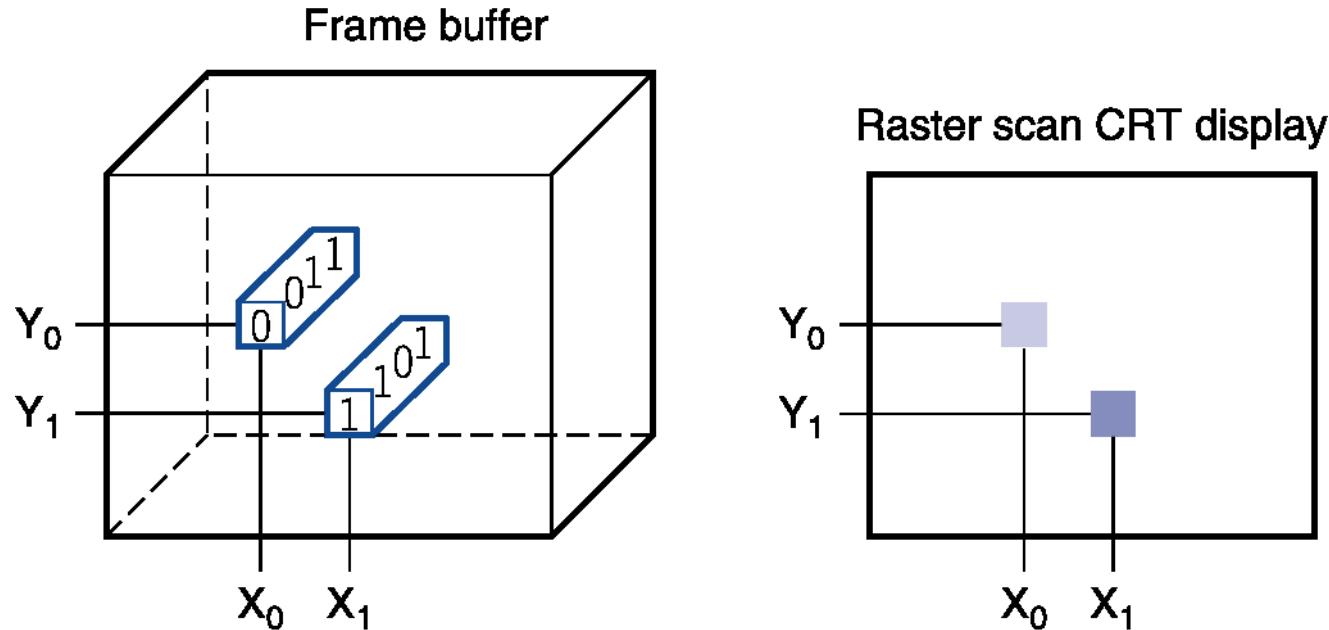
Anatomy of a Mouse

- Optical mouse
 - LED illuminates desktop
 - Small low-res camera
 - Basic image processor
 - Looks for x, y movement
 - Buttons & wheel
- Supersedes roller-ball mechanical mouse

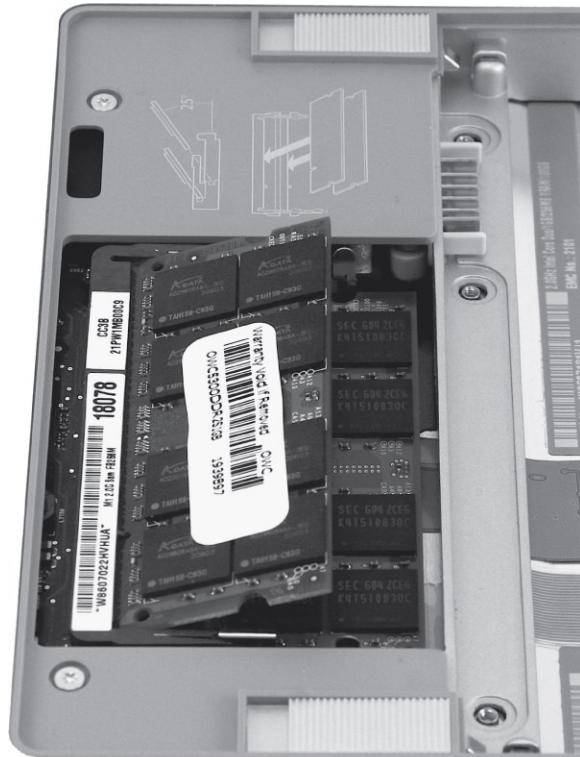
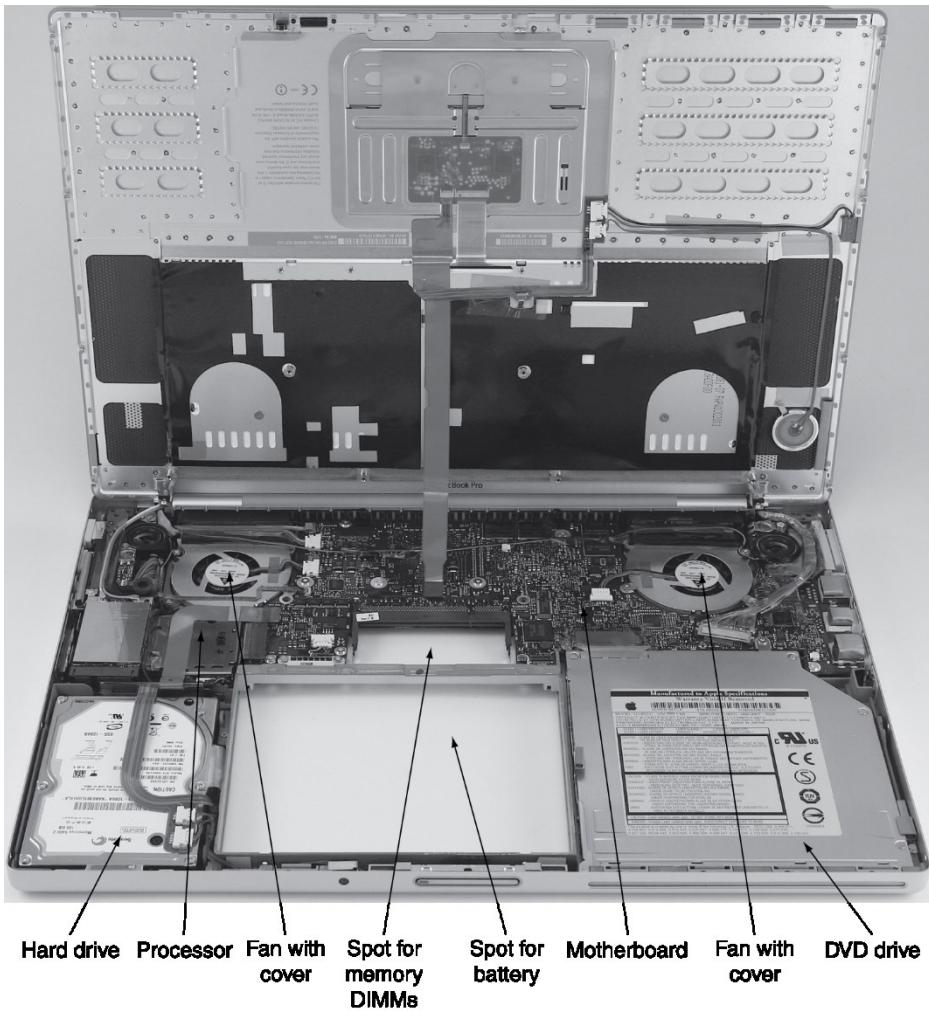


Through the Looking Glass

- LCD screen: picture elements (pixels)
 - Mirrors content of frame buffer memory



Opening the Box

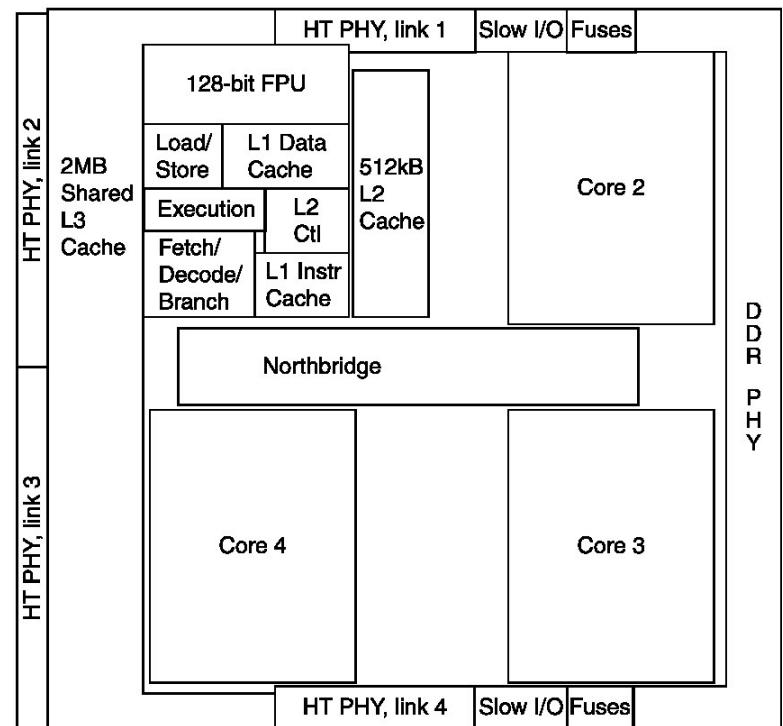
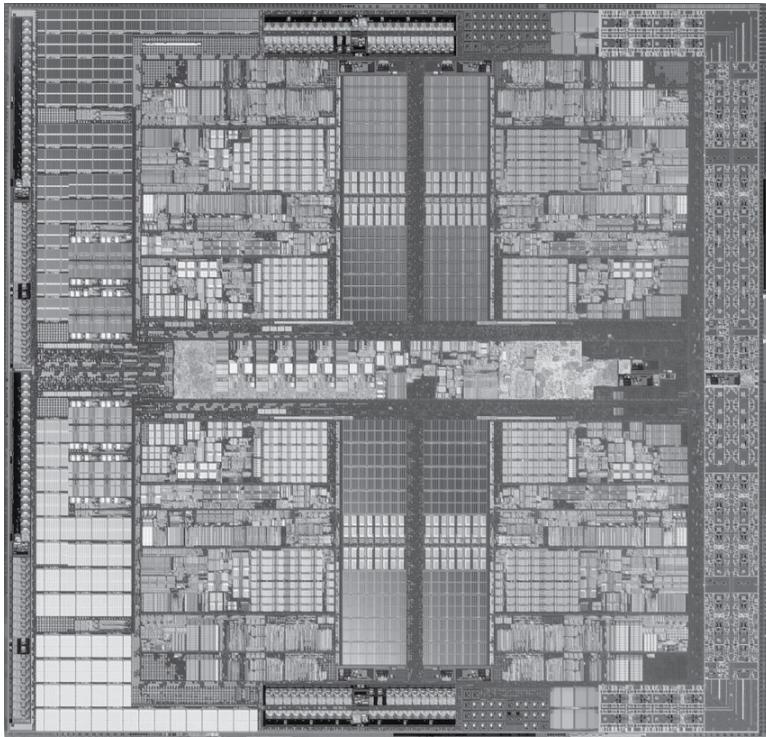


Inside the Processor (CPU)

- Datapath: performs operations on data
- Control: sequences datapath, memory, ...
- Cache memory
 - Small fast SRAM memory for immediate access to data

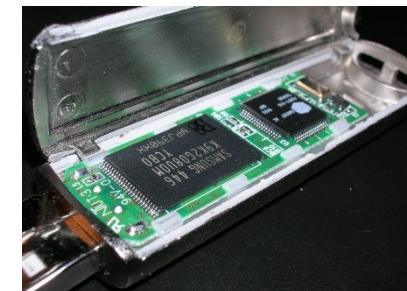
Inside the Processor

■ AMD Barcelona: 4 processor cores



A Safe Place for Data

- Volatile main memory
 - Loses instructions and data when power off
- Non-volatile secondary memory
 - Magnetic disk
 - Flash memory
 - Optical disk (CDROM, DVD)



Networks

- Communication and resource sharing
- Local area network (LAN): Ethernet
 - Within a building
- Wide area network (WAN: the Internet)
- Wireless network: WiFi, Bluetooth



Abstractions

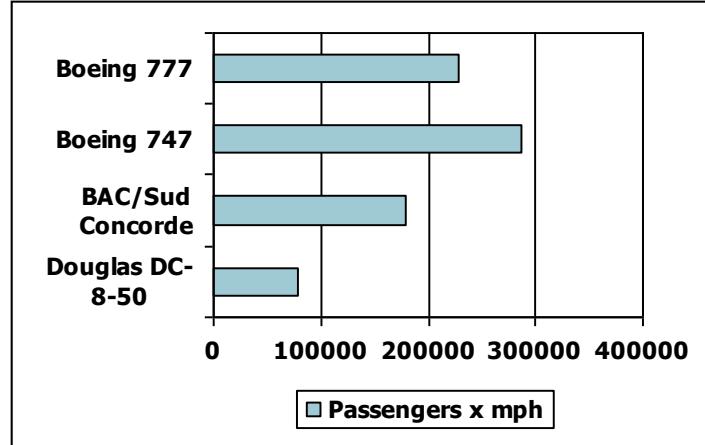
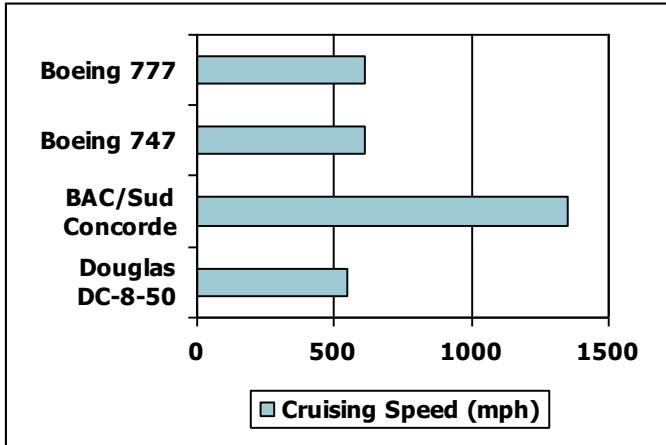
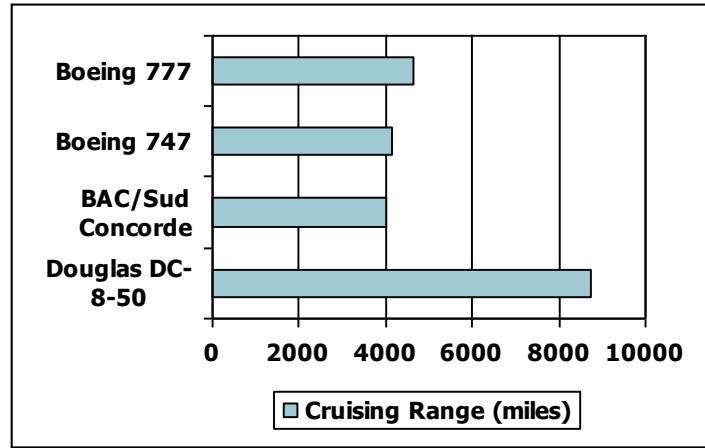
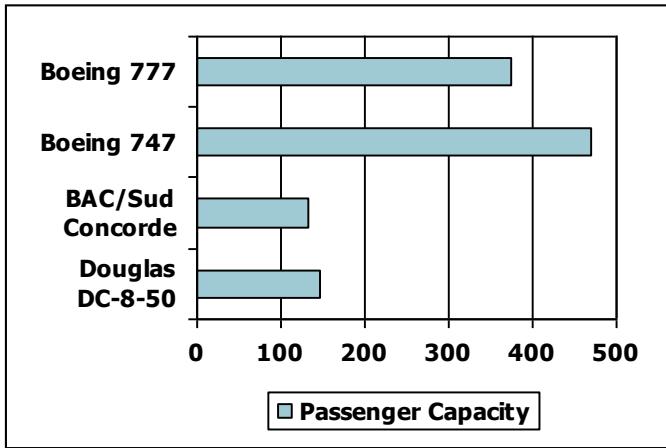
The BIG Picture

- Abstraction helps us deal with complexity
 - Hide lower-level detail
- Instruction set architecture (ISA)
 - The hardware/software interface
- Application binary interface
 - The ISA plus system software interface
- Implementation
 - The details underlying and interface



Defining Performance

- Which airplane has the best performance?



Response Time and Throughput

- Response time
 - The time it takes to do a task
- Throughput
 - Total work done per unit time
 - e.g., tasks/transactions/... per hour
- How are response time and throughput affected by
 - Replacing the processor with a faster version?
 - Adding more processors?
- We'll focus on response time for now...

Relative Performance

- Define Performance = 1/Execution Time
- “X is n time faster than Y”

$$\begin{aligned}\text{Performance}_x / \text{Performance}_y \\ = \text{Execution time}_y / \text{Execution time}_x = n\end{aligned}$$

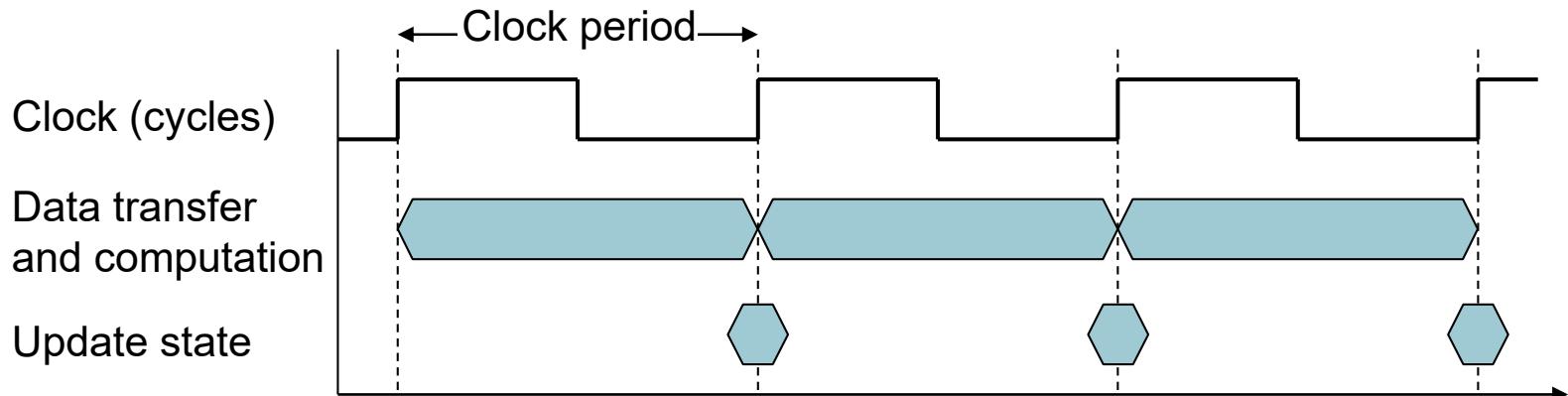
- Example: time taken to run a program
 - 10s on A, 15s on B
 - $\text{Execution Time}_B / \text{Execution Time}_A$
 $= 15s / 10s = 1.5$
 - So A is 1.5 times faster than B

Measuring Execution Time

- Elapsed time
 - Total response time, including all aspects
 - Processing, I/O, OS overhead, idle time
 - Determines system performance
- CPU time
 - Time spent processing a given job
 - Discounts I/O time, other jobs' shares
 - Comprises user CPU time and system CPU time
 - Different programs are affected in different ways by CPU and system performance

CPU Clocking

- Operation of digital hardware governed by a constant-rate clock



- Clock period: duration of a clock cycle
 - e.g., $250\text{ps} = 0.25\text{ns} = 250 \times 10^{-12}\text{s}$
- Clock frequency (rate): cycles per second
 - e.g., $4.0\text{GHz} = 4000\text{MHz} = 4.0 \times 10^9\text{Hz}$

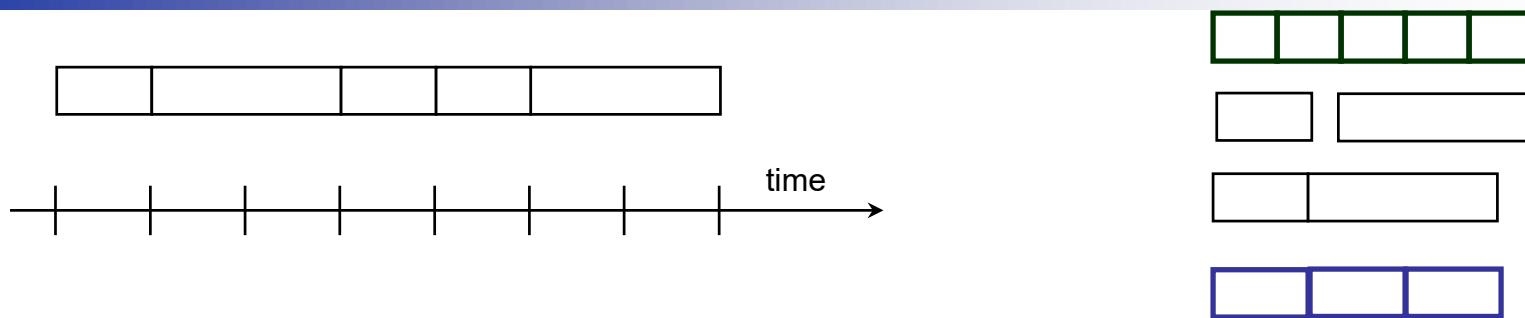
CPU Time

$\text{CPU Time} = \text{CPU Clock Cycles} \times \text{Clock Cycle Time}$

$$= \frac{\text{CPU Clock Cycles}}{\text{Clock Rate}}$$

- Performance improved by
 - Reducing number of clock cycles
 - Increasing clock rate
 - Hardware designer must often trade off clock rate against cycle count

Different numbers of cycles for different instructions



- Multiplication takes more time than addition
- Floating point operations take longer than integer ones
- Accessing memory takes more time than accessing registers
- *Important point: changing the cycle time often changes the number of cycles required for various instructions (more later)*

CPU Time Example

- Computer A: 2GHz clock, 10s CPU time
- Designing Computer B
 - Aim for 6s CPU time
 - Can do faster clock, but causes $1.2 \times$ clock cycles
- How fast must Computer B clock be?

$$\text{Clock Rate}_B = \frac{\text{Clock Cycles}_B}{\text{CPU Time}_B} = \frac{1.2 \times \text{Clock Cycles}_A}{6s}$$

$$\begin{aligned}\text{Clock Cycles}_A &= \text{CPU Time}_A \times \text{Clock Rate}_A \\ &= 10s \times 2\text{GHz} = 20 \times 10^9\end{aligned}$$

$$\text{Clock Rate}_B = \frac{1.2 \times 20 \times 10^9}{6s} = \frac{24 \times 10^9}{6s} = 4\text{GHz}$$

Instruction Count and CPI

Clock Cycles = Instruction Count \times Cycles per Instruction

CPU Time = Instruction Count \times CPI \times Clock Cycle Time

$$= \frac{\text{Instruction Count} \times \text{CPI}}{\text{Clock Rate}}$$

- Instruction Count for a program
 - Determined by program, ISA and compiler
- Average cycles per instruction
 - Determined by CPU hardware
 - If different instructions have different CPI
 - Average CPI affected by instruction mix



CPI Example

- Computer A: Cycle Time = 250ps, CPI = 2.0
- Computer B: Cycle Time = 500ps, CPI = 1.2
- Same ISA
- Which is faster, and by how much?

$$\text{CPU Time}_A = \text{Instruction Count} \times \text{CPI}_A \times \text{Cycle Time}_A$$

$$= I \times 2.0 \times 250\text{ps} = I \times 500\text{ps}$$

A is faster...

$$\text{CPU Time}_B = \text{Instruction Count} \times \text{CPI}_B \times \text{Cycle Time}_B$$

$$= I \times 1.2 \times 500\text{ps} = I \times 600\text{ps}$$

$$\frac{\text{CPU Time}_B}{\text{CPU Time}_A} = \frac{I \times 600\text{ps}}{I \times 500\text{ps}} = 1.2$$

...by this much



CPI in More Detail

- If different instruction classes take different numbers of cycles

$$\text{Clock Cycles} = \sum_{i=1}^n (\text{CPI}_i \times \text{Instruction Count}_i)$$

- Weighted average CPI

$$\text{CPI} = \frac{\text{Clock Cycles}}{\text{Instruction Count}} = \sum_{i=1}^n \left(\text{CPI}_i \times \frac{\text{Instruction Count}_i}{\text{Instruction Count}} \right)$$


Relative frequency

CPI Example

- Alternative compiled code sequences using instructions in classes A, B, C

Class	A	B	C
CPI for class	1	2	3
IC in sequence 1	2	1	2
IC in sequence 2	4	1	1

- Sequence 1: IC = 5
 - Clock Cycles
 $= 2 \times 1 + 1 \times 2 + 2 \times 3$
 $= 10$
 - Avg. CPI = $10/5 = 2.0$
- Sequence 2: IC = 6
 - Clock Cycles
 $= 4 \times 1 + 1 \times 2 + 1 \times 3$
 $= 9$
 - Avg. CPI = $9/6 = 1.5$



Performance Summary

The BIG Picture

$$\text{CPU Time} = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Clock cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Clock cycle}}$$

- Performance depends on
 - Algorithm: affects IC, possibly CPI
 - Programming language: affects IC, CPI
 - Compiler: affects IC, CPI
 - Instruction set architecture: affects IC, CPI, T_c

SPEC CPU Benchmark

- Programs used to measure performance
 - Supposedly typical of actual workload
- Standard Performance Evaluation Corp (SPEC)
 - Develops benchmarks for CPU, I/O, Web, ...
- SPEC CPU2006
 - Elapsed time to execute a selection of programs
 - Negligible I/O, so focuses on CPU performance
 - Normalize relative to reference machine
 - Summarize as geometric mean of performance ratios
 - CINT2006 (integer) and CFP2006 (floating-point)

$$\sqrt[n]{\prod_{i=1}^n \text{Execution time ratio}_i}$$

CINT2006 for Opteron X4 2356

Name	Description	ICx10 ⁹	CPI	Tc (ns)	Exec time	Ref time	SPECratio
perl	Interpreted string processing	2,118	0.75	0.40	637	9,777	15.3
bzip2	Block-sorting compression	2,389	0.85	0.40	817	9,650	11.8
gcc	GNU C Compiler	1,050	1.72	0.47	24	8,050	11.1
mcf	Combinatorial optimization	336	10.00	0.40	1,345	9,120	6.8
go	Go game (AI)	1,658	1.09	0.40	721	10,490	14.6
hmmer	Search gene sequence	2,783	0.80	0.40	890	9,330	10.5
sjeng	Chess game (AI)	2,176	0.96	0.48	37	12,100	14.5
libquantum	Quantum computer simulation	1,623	1.61	0.40	1,047	20,720	19.8
h264avc	Video compression	3,102	0.80	0.40	993	22,130	22.3
omnetpp	Discrete event simulation	587	2.94	0.40	690	6,250	9.1
astar	Games/path finding	1,082	1.79	0.40	773	7,020	9.1
xalancbmk	XML parsing	1,058	2.70	0.40	1,143	6,900	6.0
Geometric mean							11.7

High cache miss rates



SPEC Power Benchmark

- Power consumption of server at different workload levels
 - Performance: ssj_ops/sec
 - Power: Watts (Joules/sec)

$$\text{Overall ssj_ops per Watt} = \left(\sum_{i=0}^{10} \text{ssj_ops}_i \right) / \left(\sum_{i=0}^{10} \text{power}_i \right)$$

SPECpower_ssj2008 for X4

Target Load %	Performance (ssj_ops/sec)	Average Power (Watts)
100%	231,867	295
90%	211,282	286
80%	185,803	275
70%	163,427	265
60%	140,160	256
50%	118,324	246
40%	920,35	233
30%	70,500	222
20%	47,126	206
10%	23,066	180
0%	0	141
Overall sum	1,283,590	2,605
$\Sigma ssj_ops / \Sigma power$		493

Fallacies and Pitfalls

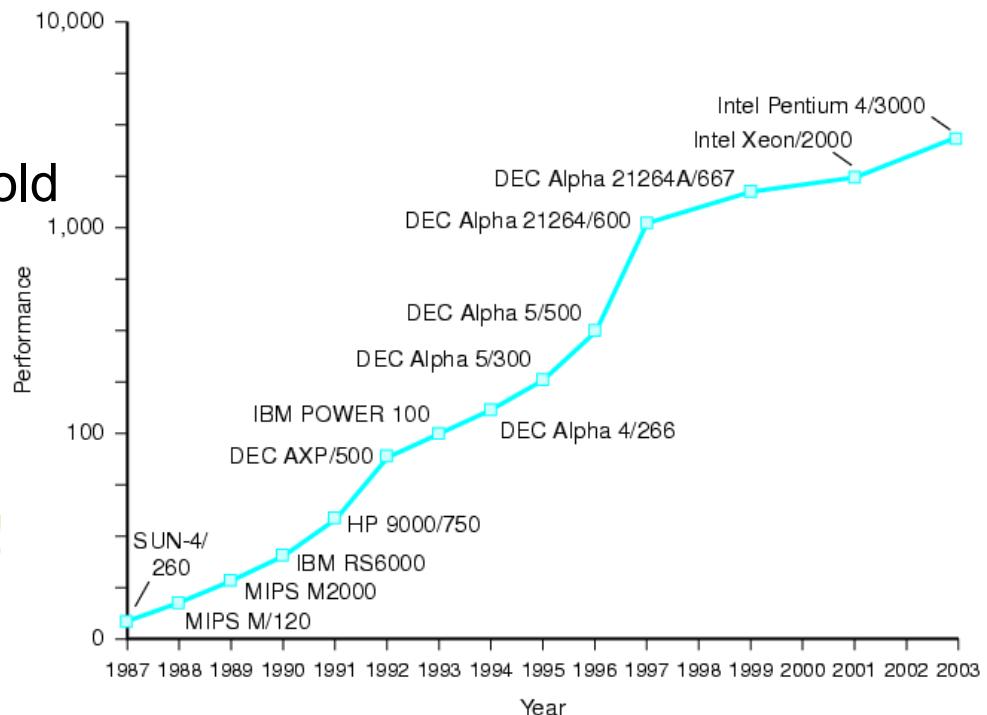
- Fallacy – computers have been built in the same, old-fashioned way for far too long, and this antiquated model of computation is running out of steam
- Pitfall – ignoring the inexorable progress of hardware when planning a new machine

- Is a three-year-later computer with a threefold speedup powerful?

improve 50%/year or
200%/18months

$$1.5^3 = 3.375$$

NO!



Pitfall: Amdahl's Law

- Improving an aspect of a computer and expecting a proportional improvement in overall performance

$$T_{\text{improved}} = \frac{T_{\text{affected}}}{\text{improvement factor}} + T_{\text{unaffected}}$$

- Example: multiply accounts for 80s/100s
 - How much improvement in multiply performance to get 5x overall?

$$20 = \frac{80}{n} + 20$$

- Can't be done!

- Corollary: make the common case fast



Fallacy: Low Power at Idle

- Look back at X4 power benchmark
 - At 100% load: 295W
 - At 50% load: 246W (83%)
 - At 10% load: 180W (61%)
- Google data center
 - Mostly operates at 10% – 50% load
 - At 100% load less than 1% of the time
- Consider designing processors to make power proportional to load

Pitfall: MIPS as a Performance Metric

- MIPS: Millions of Instructions Per Second
 - Doesn't account for
 - Differences in ISAs between computers
 - Differences in complexity between instructions

$$\begin{aligned}\text{MIPS} &= \frac{\text{Instruction count}}{\text{Execution time} \times 10^6} \\ &= \frac{\text{Instruction count}}{\frac{\text{Instruction count} \times \text{CPI}}{\text{Clock rate}} \times 10^6} = \frac{\text{Clock rate}}{\text{CPI} \times 10^6}\end{aligned}$$

- CPI varies between programs on a given CPU

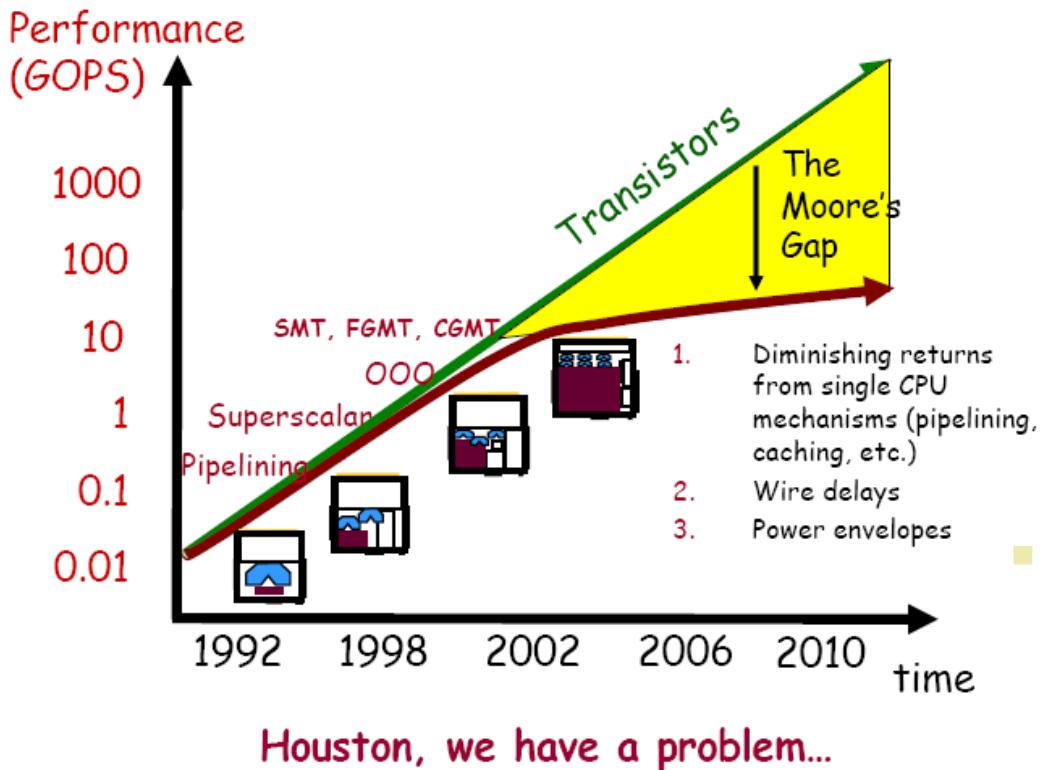


Concluding Remarks

- Cost/performance is improving
 - Due to underlying technology development
- Hierarchical layers of abstraction
 - In both hardware and software
- Instruction set architecture
 - The hardware/software interface
- Execution time: the best performance measure
- Power is a limiting factor
 - Use parallelism to improve performance

Future Trend of Computer Architecture

The "Moore's Gap"



The Moore's Gap - Example

Pentium 3	Pentium 4
1 GHz	1.4 GHz
Year 2000	Year 2000
0.18 micron	0.18 micron
28M transistors	42M transistors
343 (Specint 2000)	393 (Specint 2000)

Transistor count increased by 50%
Performance increased by only 15%

Energy

- Network transfer (1mm): 3pJ
- Off-chip memory read: 500pJ
- 32KB cache read: 50pJ
- ALU add: 2pJ

New wisdom

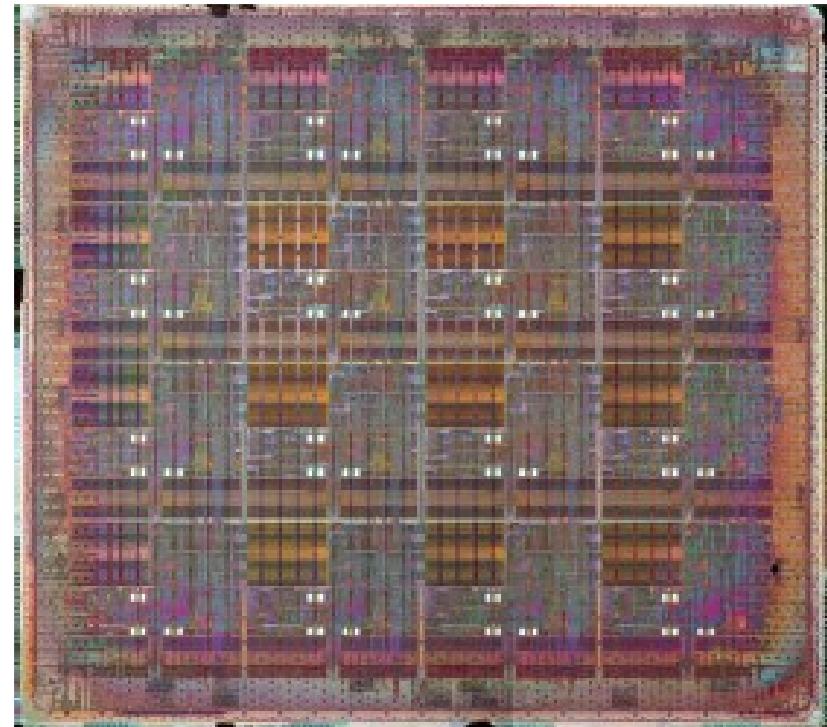
power is expensive, xor is free
multiplication is fast, memory access is slow

- Old wisdom
- Power is free xor is expensive
- Multiplication is slow memory access is fast

Source: Anant Agarwal, "The why, Where and How of Multicore"

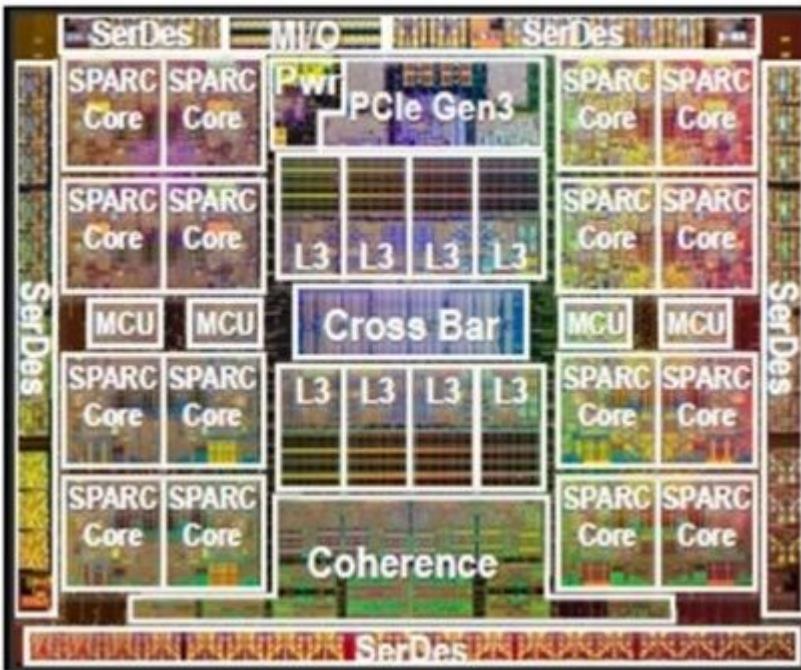
Multi-Core

- 16 cores in 2002 using IBM SA27E cell library by 0.18 micron technology.
- 425 MHz, 6.8 GOPS
- A chip with over several hundreds of cores is expectable in the near future. (by Intel)



SPARC Multi-Core

T5 Processor Overview



- 16 S3 cores @ 3.6GHz
- 8MB shared L3 Cache
- 8 DDR3 BL8 Schedulers providing 80 GB/s BW
- 8-way 1-hop glueless scalability
- Integrated 2x8 PCIe Gen 3
- Advanced Power Management with DVFS