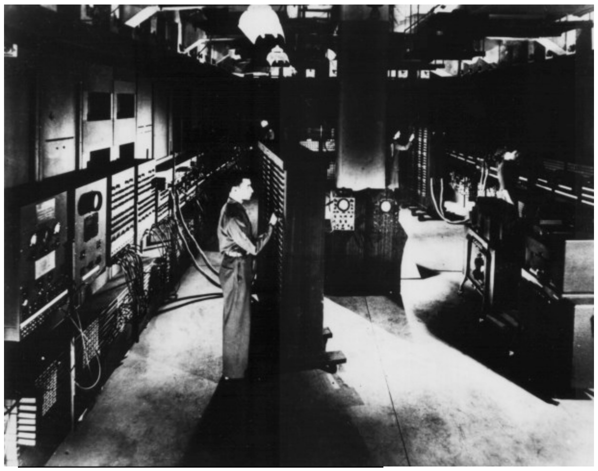
Chapter 1

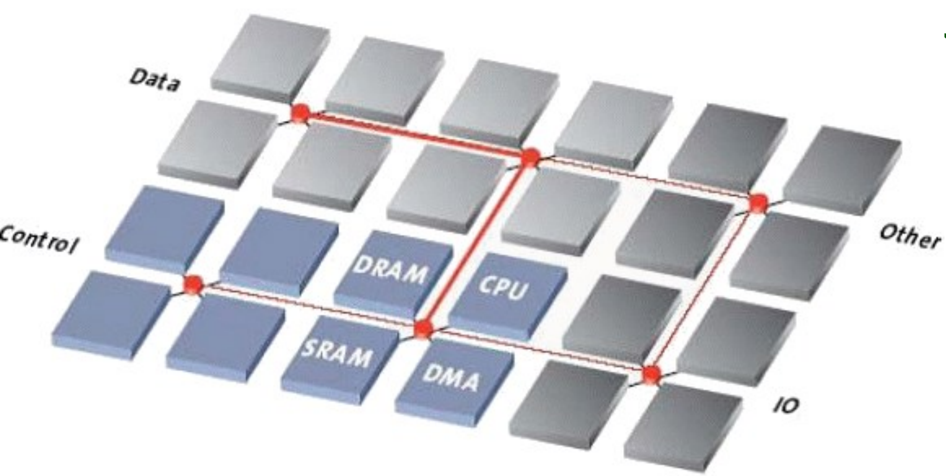
Computer Abstractions and Technology



Kuei-Chung Chang

Email: changkc@fcu.edu.tw

Ext. 3753, Office 211



ation Engineering and Computer Science

Feng Chia Univ.

Spring 2019

The Computer Revolution

- Makes novel applications feasible
  - Computers in automobiles
  - Cell phones
  - Human genome project
  - World Wide Web
  - Search Engines

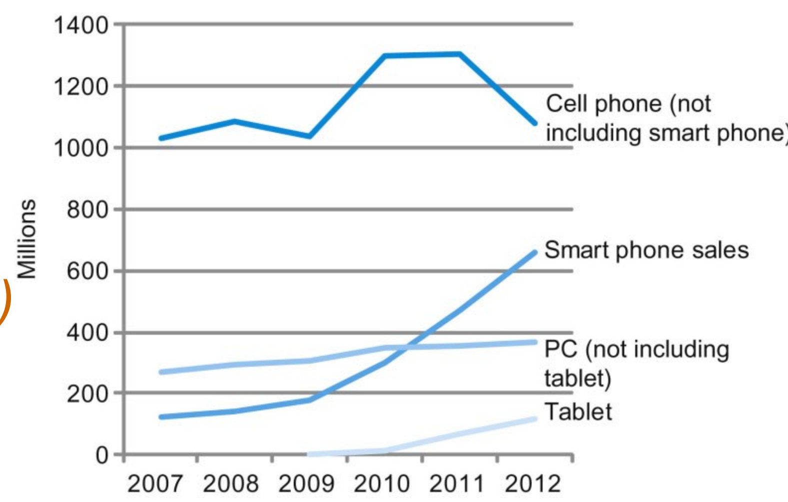
- Computers are pervasive

Classes of Computers

- This course is all about how computers work
- But, what do we mean by a computer?
  - Different types:
    - desktop, servers, embedded devices
  - Different uses:
    - automobiles, graphics, finance, genomics…
  - Different manufacturers:
    - Intel, Apple, IBM, Microsoft, Sun…
  - Different underlying technologies and different costs!

Classes of computing applications & characteristics

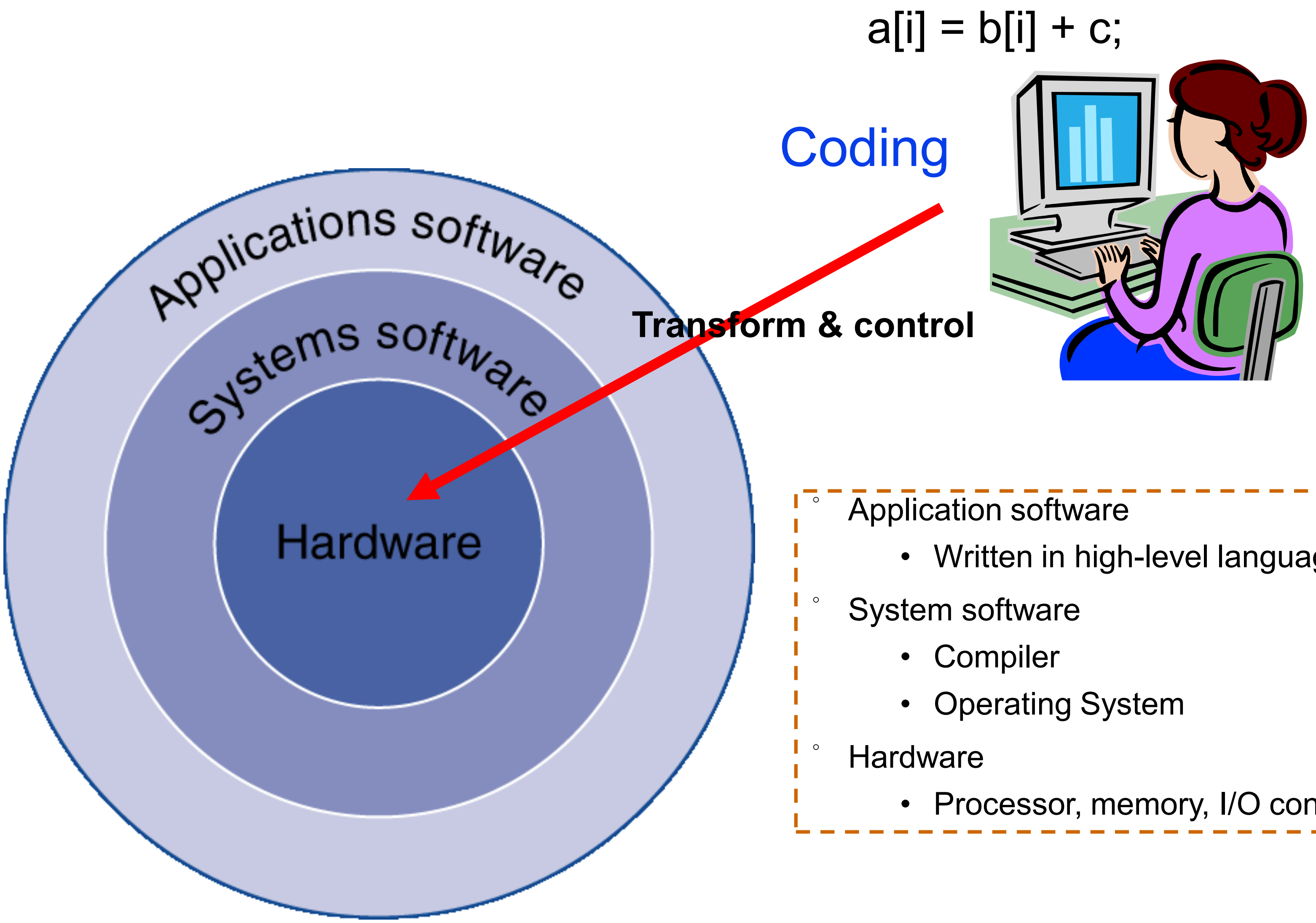
- Desktop computers
  - Emphasize delivering good performance to a single user at low cost and are used to execute thirty-party software
- Servers
  - Oriented to carrying large workloads
  - Scientific or engineering applications
- *Embedded computers*
  - *Designed to run one application or one set of related applications*
  - *During the last years, the growth in cell phones (embedded devices) has been much faster than the growth rate of PCs.*
  - *Eg. iPhone, iPad*



The PostPC Era

- Personal Mobile Device (PMD)
  - Battery operated
  - Connects to the Internet
  - Hundreds of dollars
  - Smart phones, tablets, electronic glasses
- Cloud computing
  - Large collections of servers that provide services over the Internet.
  - Software as a Service (SaaS)
  - Portion of software run on a PMD and a portion run in the Cloud

Hierarchical layers – Below Your Program



- Application software
  - Written in high-level language
- System software
  - Compiler
  - Operating System
- Hardware
  - Processor, memory, I/O controllers

Abstraction - The HW/SW Interface

Application software

Systems software
(OS, compiler)

Hardware

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;
}
```

Compiler

Assembly language program (for MIPS)

```
swap:
      muli $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
```

Assembler

Binary machine language program (for MIPS)

```
00000000101000010000000000011000
00000000000110000001100000100001
10001100011000100000000000000000
10001100111100100000000000000100
10101100111100100000000000000000
10101100011000100000000000000100
00000011111000000000000000001000
```

Levels of Transformation

Task or Application

Algorithm

Program

Machine Language (ISA)

Microarchitecture

Logic

Circuits

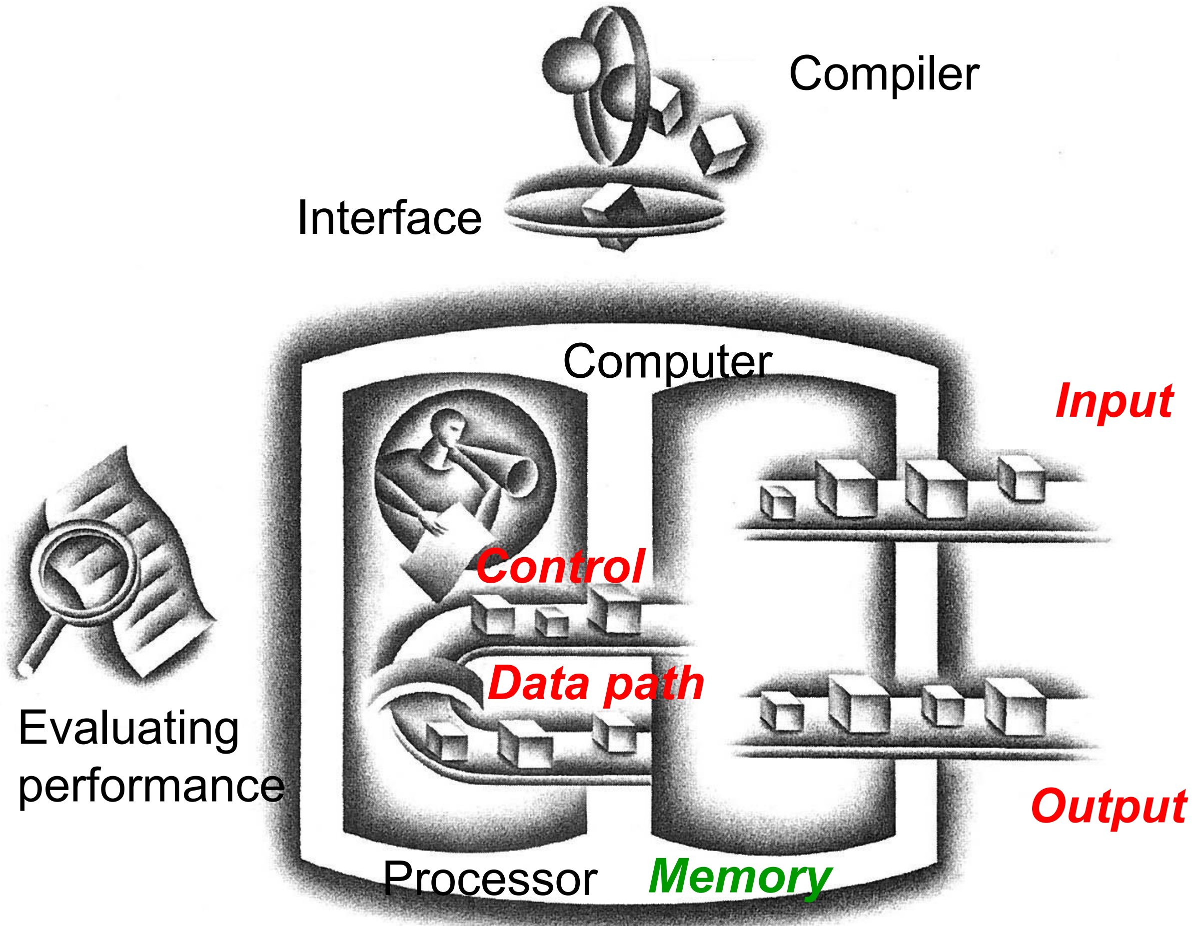
Devices

Under the Covers - What is a computer?

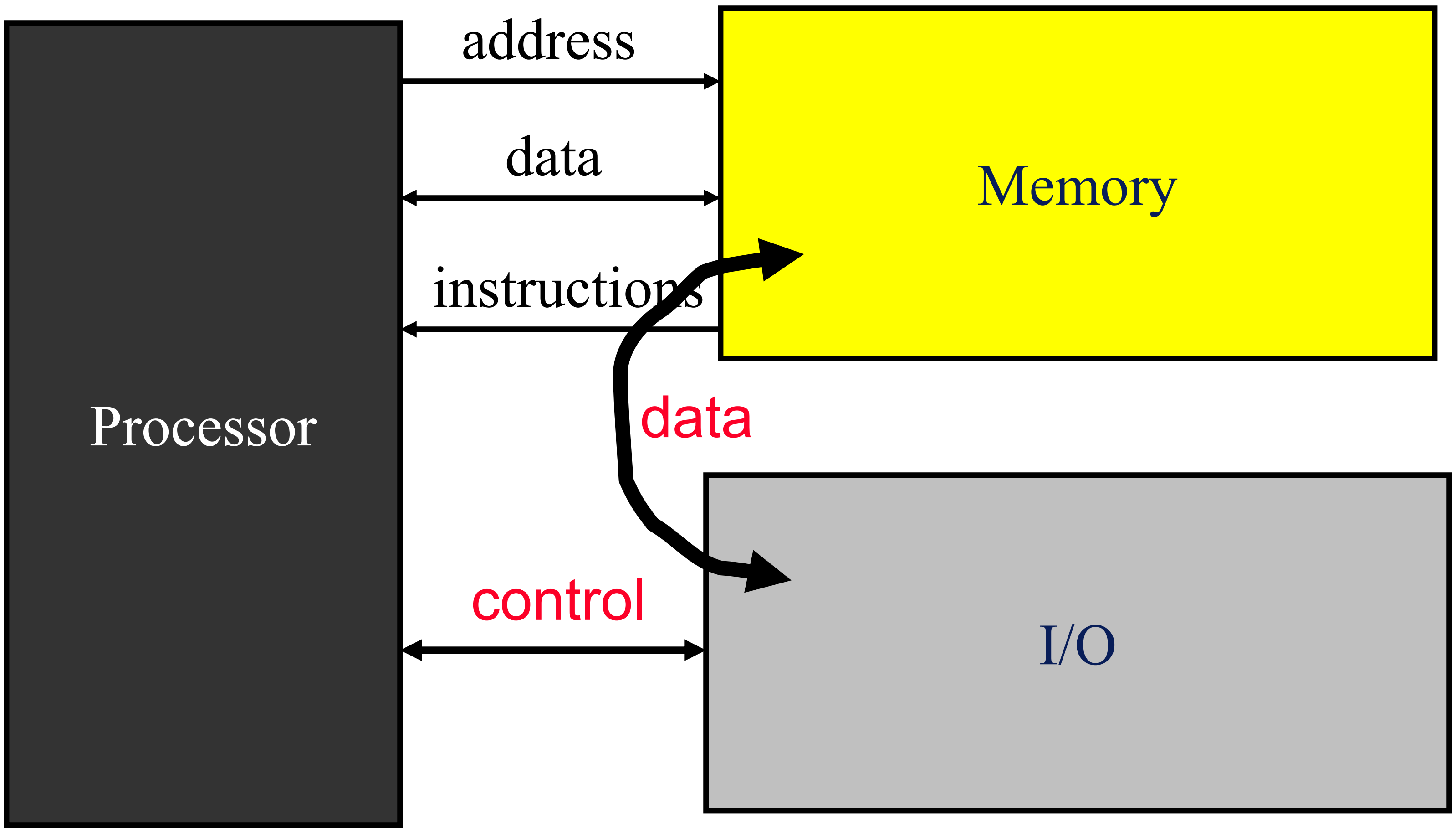
- ***Our primary focus***: the processor (datapath and control) in general-purpose computer
  - implemented using millions of transistors
  - Impossible to understand by looking at each transistor
    - Focus on macro blocks of these components
      - processor
      - input (mouse, keyboard)
      - output (display, printer)
      - memory (disk drives, DRAM, SRAM, CD)
    -

What is Computer Organization?

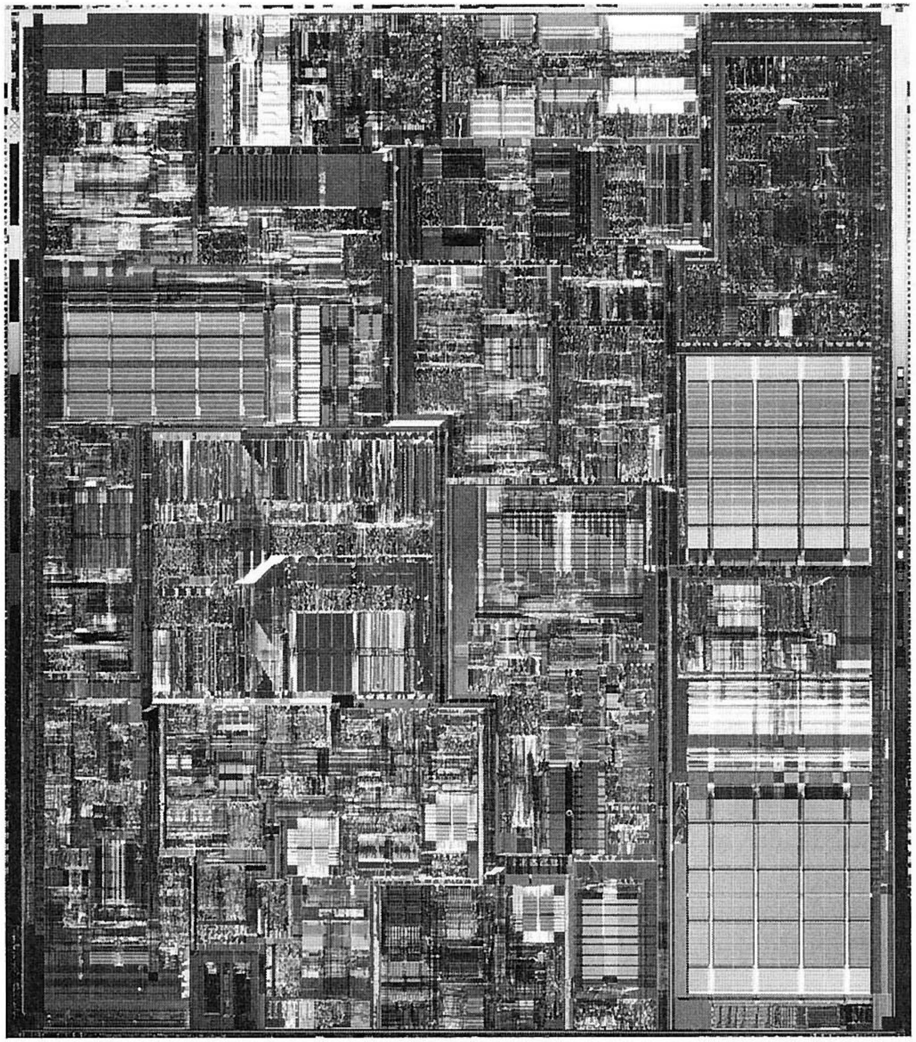
- "Computer Organization"
  - ***How computer systems are organized and designed***

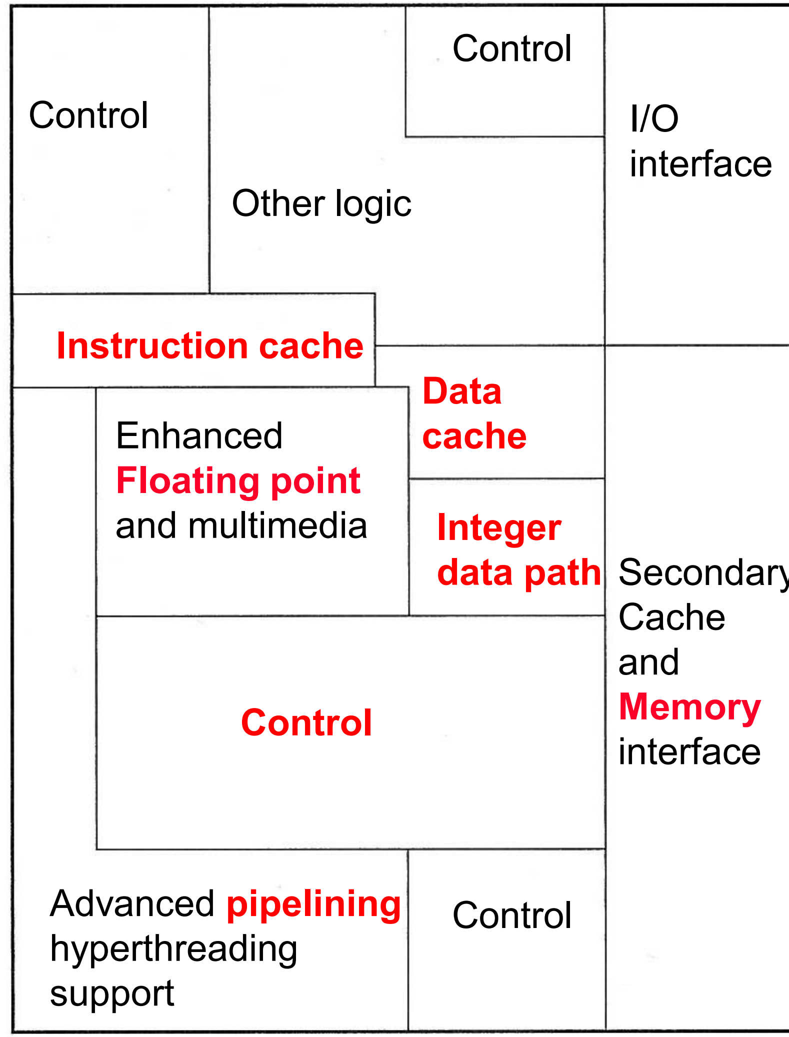


Computer Organization



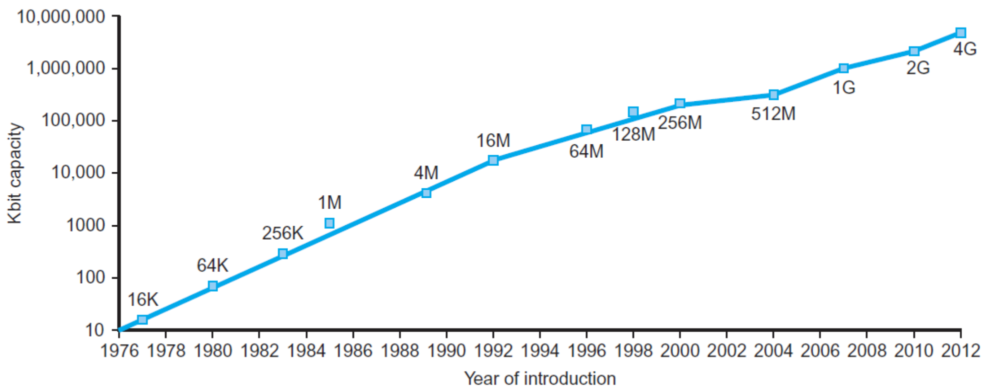
Inside the processor chip





Technology Trends

- Electronics technology continues to evolve
  - Increased capacity and performance
  - Reduced cost



DRAM capacity

Year	Technology	Relative performance/cost
1951	Vacuum tube	1
1965	Transistor	35
1975	Integrated circuit (IC)	900
1995	Very large scale IC (VLSI)	2,400,000
2013	Ultra large scale IC	250,000,000,000

Technology => dramatic change

- Processor
  - logic capacity: about 30% per year
  - ***clock rate***: about 20% per year
- Memory
  - DRAM capacity: about 60% per year
  - ***Memory speed***: about 10% per year
  - Cost per bit: improves about 25% per year
- Disk
  - capacity: about 60% per year
- Network Bandwidth
  - Bandwidth increasing more than 100% per year!

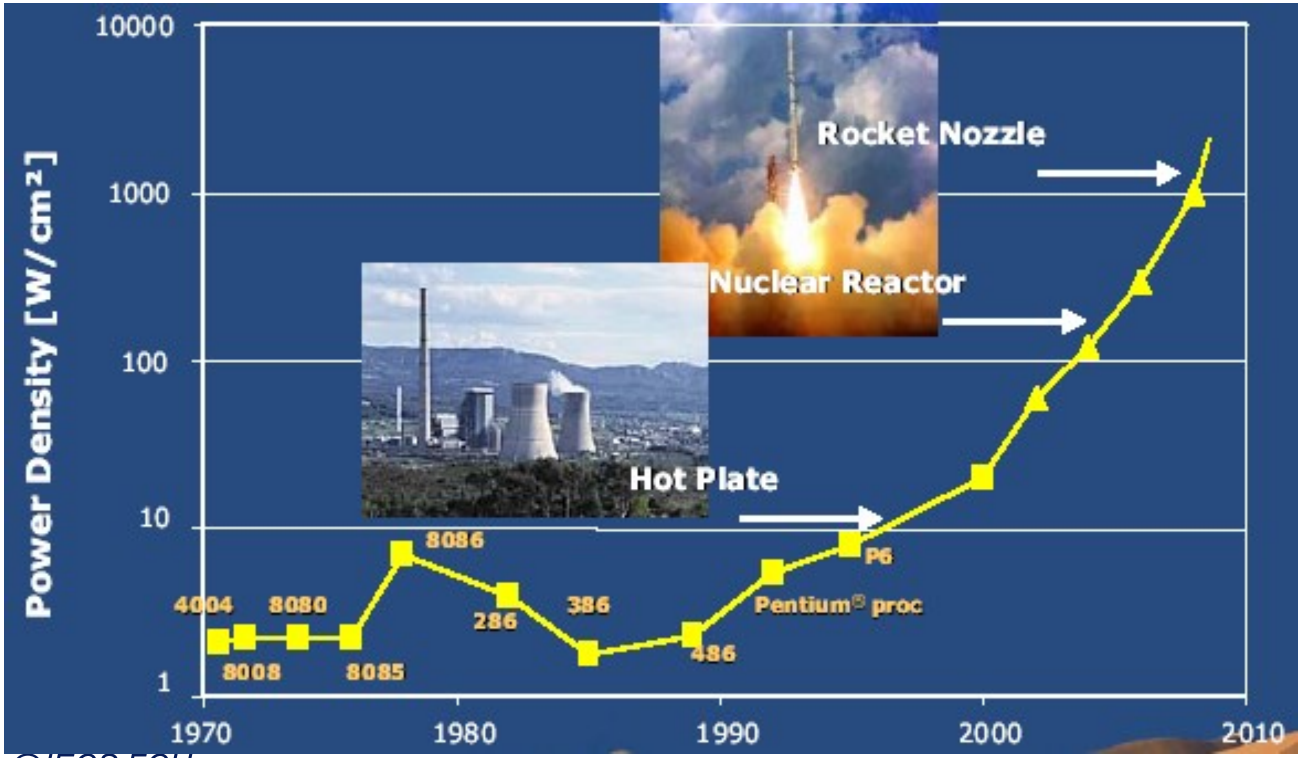
Performance

- Another slide

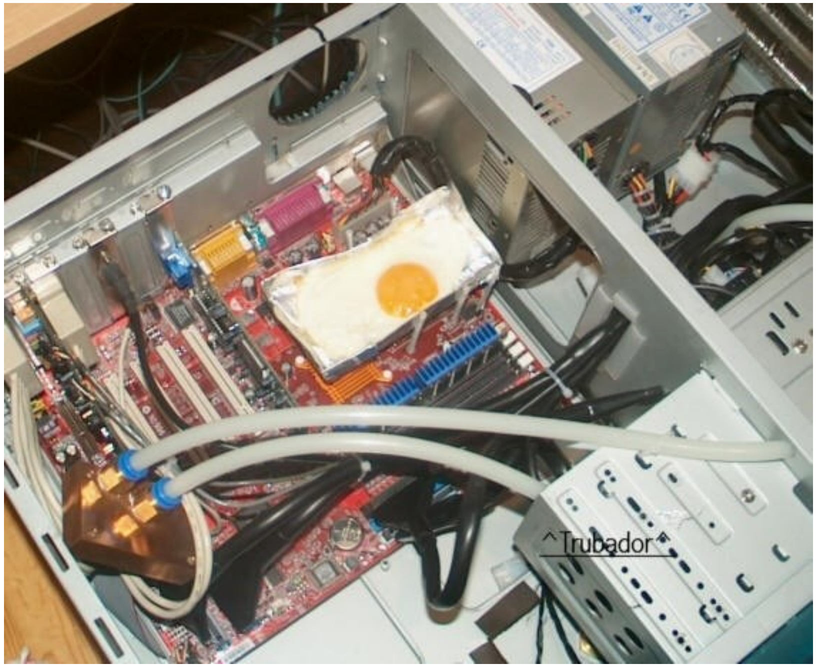
Challenges for the Hardware Designers

Major concerns in the future:

- The performance problem (especially scientific workloads)
- The *power dissipation problem* (especially embedded processors)
- The *temperature* problem
- The *reliability* problem



Interesting Videos for “Hot” CPU

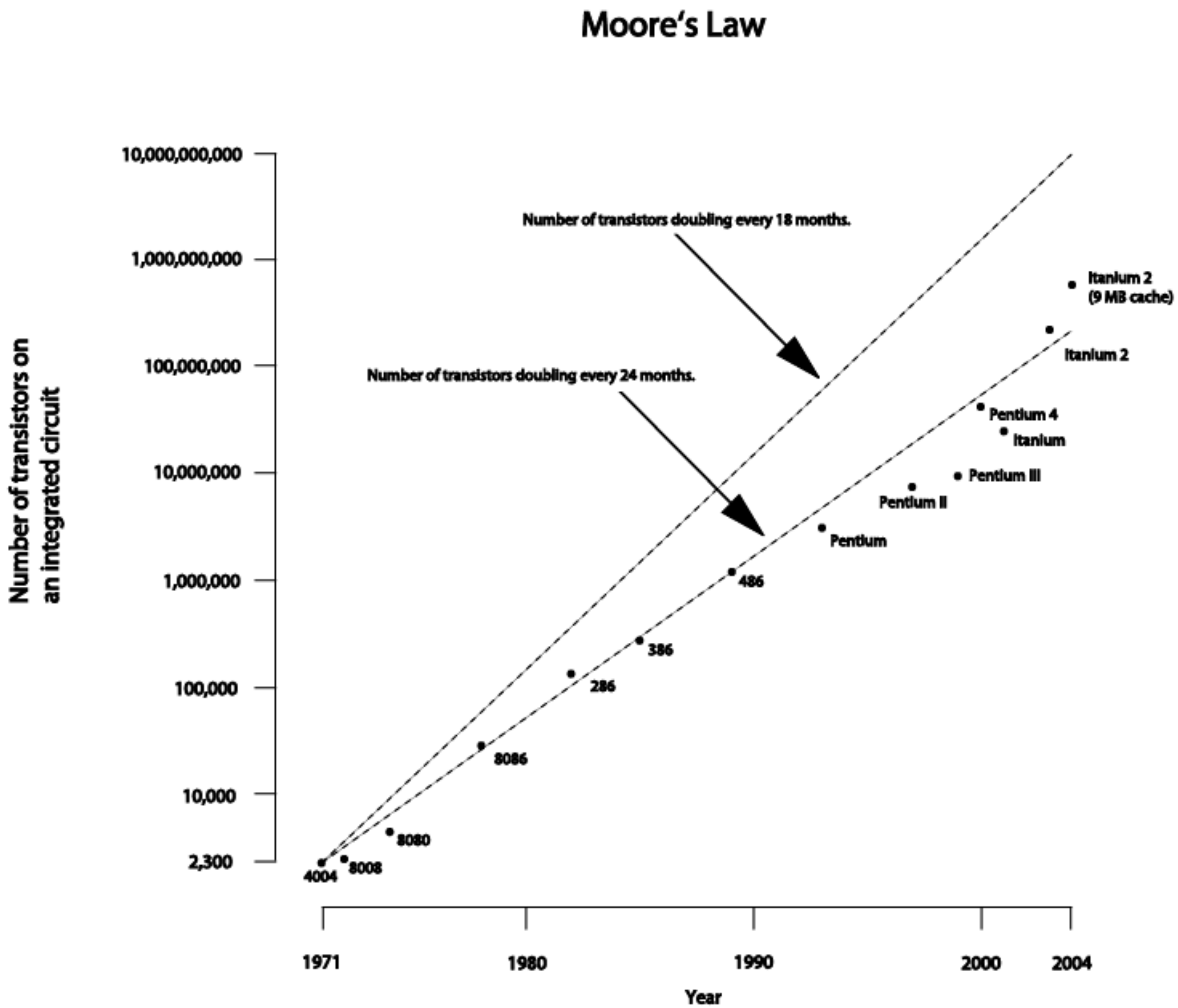


這是用CPU 煎蛋嗎? [PC AMDのCPU FX9590**]**
https://www.youtube.com/watch?v=IxGtV0CmsT0
https://www.youtube.com/watch?v=U9WuxWHGSto

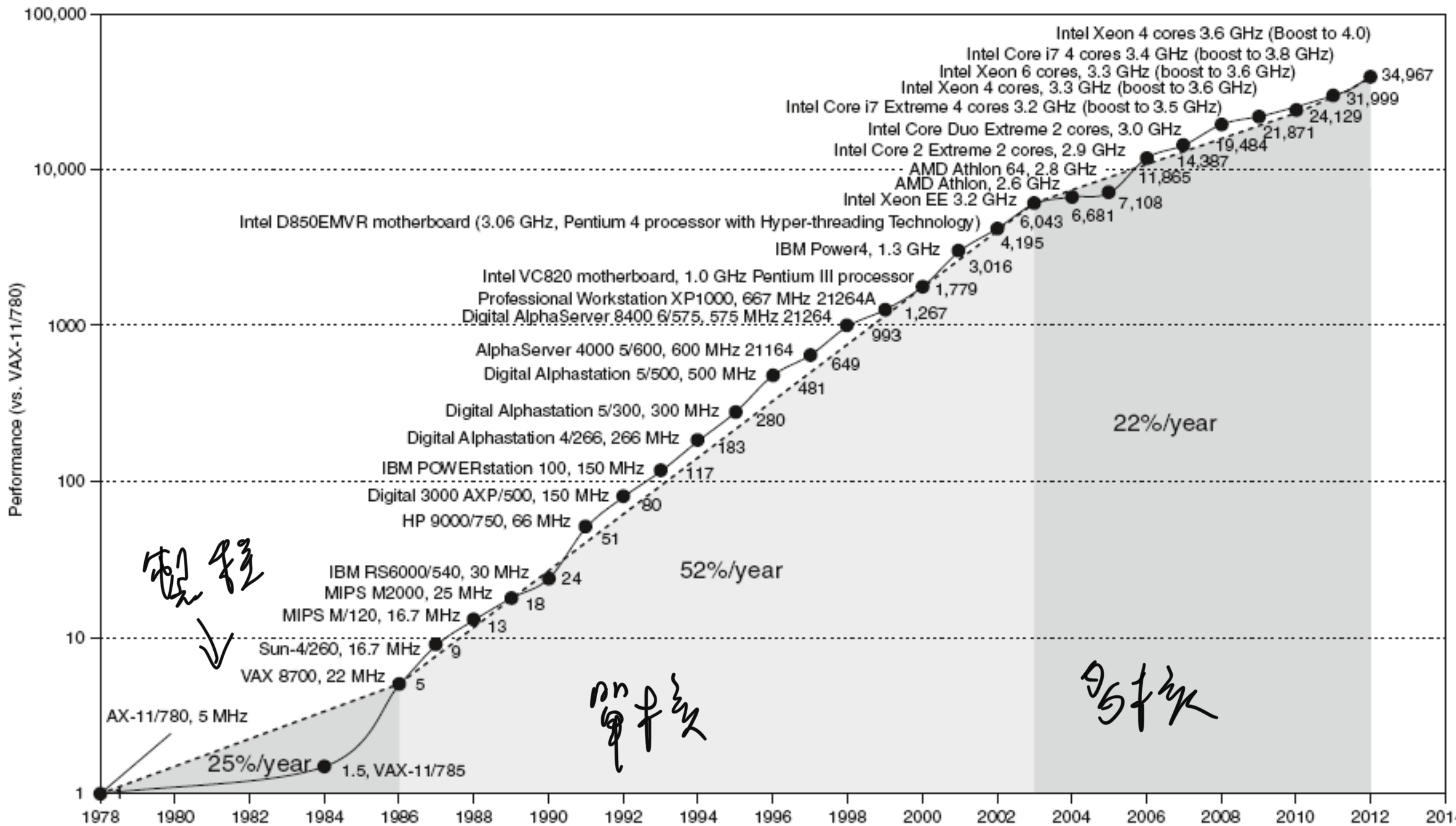
測試 CPU 的方式：
https://www.youtube.com/watch?v=NMmDl3jkhf4

Moore's Law

- Number of transistors doubling every 18 months



Crossroads: Uniprocessor Performance



- VAX : 25%/year 1978 to 1986 (**technology**)
- RISC + x86: 52%/year 1986 to 2002 (**advanced architecture**)
- RISC + x86: **22%**/year 2002 to present

Modern Trends

- Historical contributions to performance:
  - Better processes (faster devices) ~20%
  - Better circuits/pipelines ~15%
  - Better organization/architecture ~15%

	Pentium	P-Pro	P-II	P-III	P-4	Itanium	Montecito
Year	1993	1995	1997	1999	2000	2002	2005
Transistors	3.1M	5.5M	7.5M	9.5M	42M	300M	1720M
Clock Speed	60M	200M	300M	500M	1500M	800M	1800M

Moore's Law in action

What is Moore's Law?

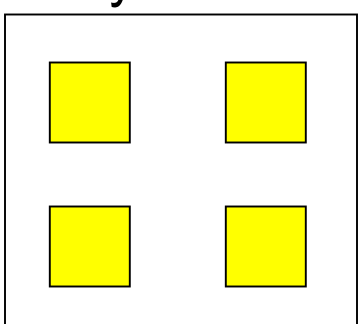
At this point, adding transistors to a core yields little benefit

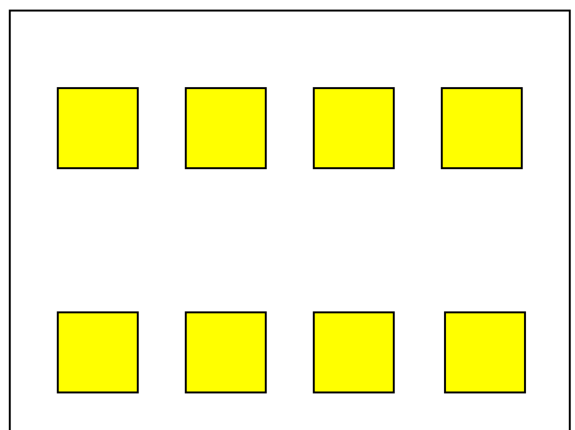
In the future, these techniques will not help much for a single core!
(That's why multi-core)

What does this mean to a programmer?

- ***In the past***, add a new chip directly meant 50% higher performance for a program
- ***Today***, one can expect only a 20% improvement, unless… the program can be broken up into multiple threads
- Expect #threads to emerge as a major metric for software quality

4-way multi-core





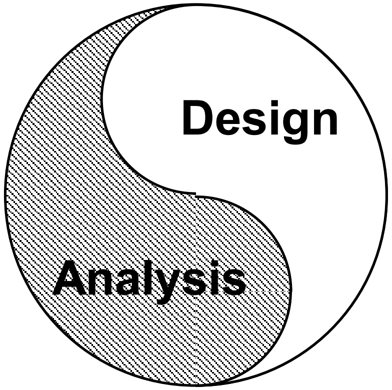
8-way multi-core

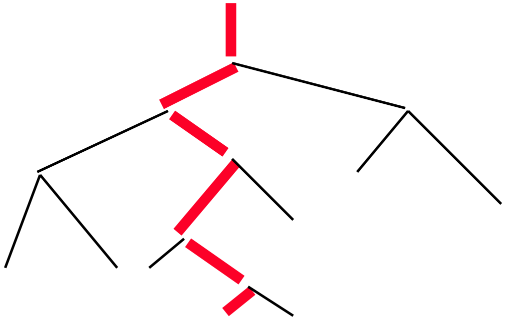
Measurement and Evaluation

Architecture is an iterative process

-- searching the space of possible designs

-- at all levels of computer systems



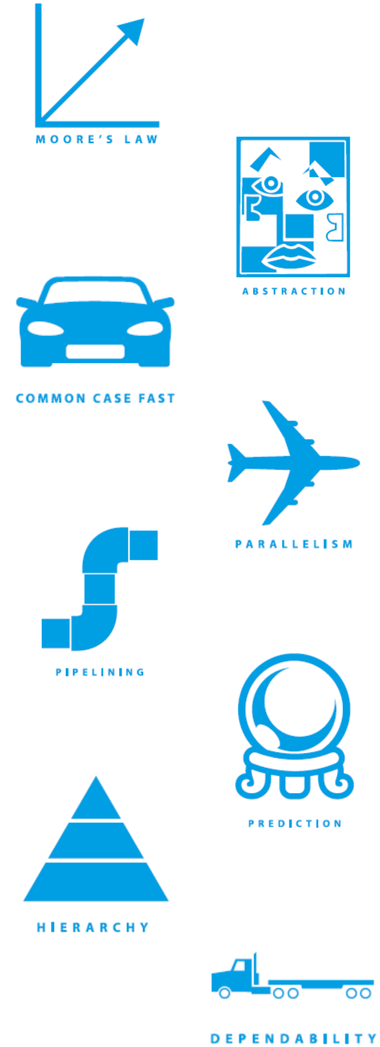


What will we learn in this stuff?

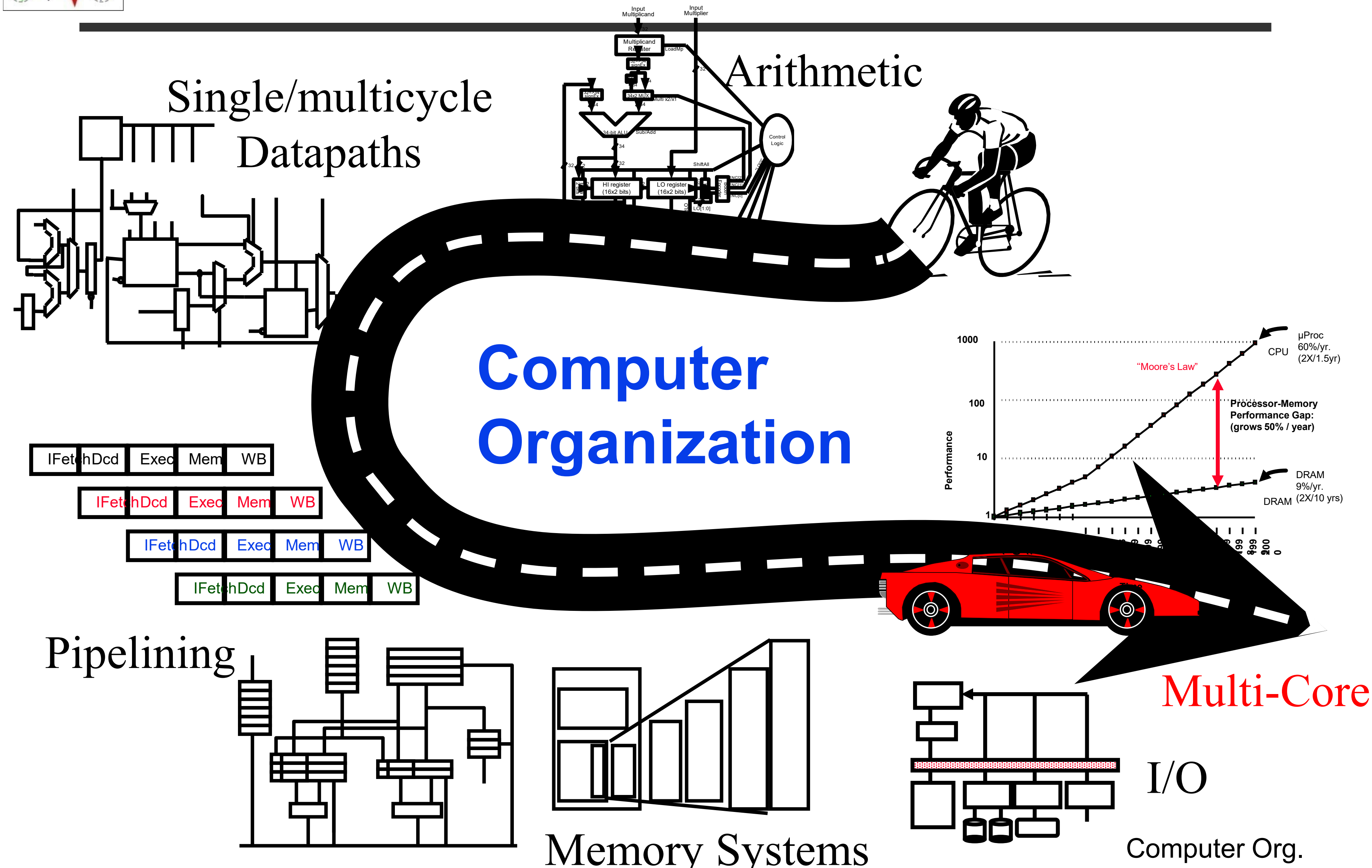
- **How** programs are translated into the machine language and how the hardware executes them
- The hardware/software **interface**
- What determines program **performance** and how it can be improved
- How hardware designers **improve performance**
- What is **parallel** processing

Eight Great Ideas

- Design for ***Moore’s Law***
- Use ***abstraction*** to simplify design
- Make the ***common case fast***
- Performance *via* ***parallelism***
- Performance *via* ***pipelining***
- Performance *via* ***prediction***
- ***Hierarchy*** of memories
- ***Dependability*** *via* redundancy



Where are we going??



Summary：How do computers work?

- Need to understand abstractions such as:
  - Applications software
  - *Systems software*
  - *Assembly Language*
  - *Machine Language*
  - *Architectural Issues: i.e., Caches, Virtual Memory, Pipelining*
  - Sequential logic, finite state machines
  - Combinational logic, arithmetic circuits
  - Boolean logic, 1s and 0s
  - Transistors used to build logic gates (CMOS)
  - Semiconductors/Silicon used to build transistors
  - Properties of atoms, electrons, and quantum dynamics
- *So much to learn ……………. !*

Q & A?