



# Ch.1-1 Computer Abstractions & Technology

## ▼ 1.1 서론

### ▼ The Computer Revolution

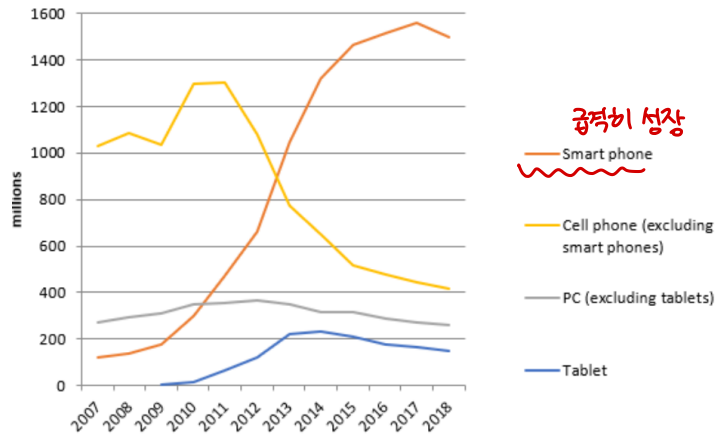
거의 모든 분야에 영향을 미치고 있음

- computers in automobiles
- cell phones
- human genome project
- world wide web
- search engines

### ▼ Classes of Computers

- ① personal computers(desktop) → 성능↓ cost↓
  - general purpose, variety of software → 사양, 목적에 따라 개인화
- ② server computers
  - 네트워크를 통해서만 접근
  - 고성능
  - 가격과 성능의 폭이 큼
- ③ supercomputers
  - 서버 중 하나
  - high-end scientific, engineering calculations
  - 서버의 작은 부분 차지 + 컴퓨터 시장 전체에서도 작은 부분 차지
- ④ embedded computers
  - system 내부에 숨겨져 있음
  - stringent power/performance/cost constraints

### ▼ The PostPC Era



- PMD : personal mobile device
  - battery operated
  - connects to the internet → 무선으로 연결
  - hundreds of dollars
  - smart phones, tablets, electronic glasses
- Cloud computing : 거대 규모의 데이터 센터를 이용
  - WSC : 창고 규모의 컴퓨팅
  - SaaS : 서비스로서의 소프트웨어 → 소프트웨어 산업에서 혁명
  - 요즘에는 응용의 일부만 PMD 수행 + 나머지는 클라우드에서 실행
  - Amazon, Google

### ▼ What you will learn

1. C, java와 같이 상위 언어로 작성된 프로그램을 어떻게 h/w언어로 번역 + 실행?
2. 소프트웨어와 하드웨어 사이의 인터페이스는 무엇이며 어떻게 지시하는가?
3. 프로그램의 성능을 결정하는 요소는 무엇이며 어떻게 향상하는가?
4. 하드웨어 설계자는 성능 개선을 위해 어떤 기술을 사용하는가?
5. 병렬 프로세싱?

### ▼ Understanding Performance

h/w or s/w	impact
algorithm	operation 작업 수를 결정
<u>programming language, compiler, architecture</u>	각 operation에 해당하는 기계어 명령어 수 결정
<u>processor, memroy system</u>	instruction 실행 속도 결정
I/O system	입출력 작업 속도 결정

## ▼ 1.2 Seven great Ideas in Computer Architecture

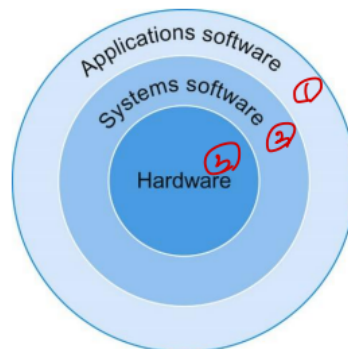
1. Use Abstraction to Simplify Design *복잡함 이용.*
  - a. 설계 시간이 길어지는 것을 방지 → 하위 수준의 상세한 사항이 안 보이도록 상위 단순
2. Make the Common Case Fast *library fast.*
  - a. system이 너무 복잡해지는 것을 방지
3. Performance via Parallelism *병행성*
4. Performance via Pipelining
5. Performance via Prediction *예측*
6. Hierarchy of Memories
7. Dependability via Redundancy (신용도 개선)

## ▼ 1.3 Below your Program

응용 소프트웨어가 하드웨어에게 지시하기 위해서는 단순한 컴퓨터 명령어로 번역하는 sw

→ Abstraction

⇒ 이러한 s는 계층적으로 구성됨



1. application sw(응용 sw) : high level로 쓰임
2. system sw(시스템 sw)
  - a. compiler : 상위 수준의 언어를 기계어로 번역
  - b. operating system : 사용자와 hw 간의 인터페이스 역할 *한는 SUPERVISING PROGRAM*
    - i. 기본적인 입출력 작업 처리
    - ii. 보조기억장치 및 메모리 할당 *storage*

iii. scheduling tasks & sharing resources

3. hw : processor, memory I/O controllers

## ▼ Levels of Program code

MIPS : assembly language program

	high-level language (C, C++, java)	<ul style="list-style-type: none"> <li>• Level of abstraction closer to problem domain</li> <li>• 생산성 유지, 이식성 유지</li> </ul>
→ compiler (one to many)	assembly language program (MIPS) ☆	기계 명령을 기호로 표현
→ assembler (one to one)	binary machine language	<ul style="list-style-type: none"> <li>• hw 표현법</li> <li>• binary digits(bits)</li> <li>• encoded instructions and data</li> </ul>

assembly language  
binary code  
⇒ 1:1 mapping

## ▼ advantages of higher-level languages

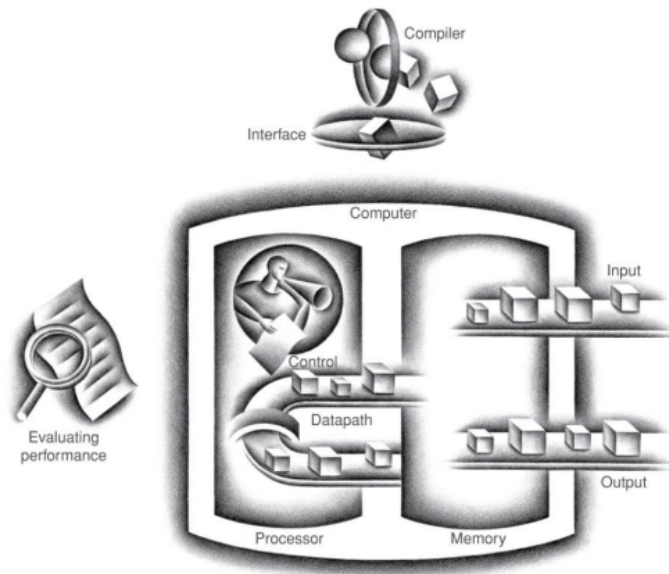
1. allow the programmer to think in a more natural language → 사용 목적에 적합하게!
  - a. improve programmer productivity + maintainability → 많이 만든 코드를 아껴
2. 개발한 기종과 상관없이 어느 컴퓨터에서든 실행이 가능
  - a. 컴파일러, 어셈블러가 상위 수준 언어 프로그램을 번역 가능하기 때문

## ▼ abstractions

- complexity → simplified
  - hide lower-level detail
- Instruction set architecture (ISA) → ex. MIPS, ARM, RISC-V ...
  - hw/low level sw interface
  - 중요한 추상화 개념
  - 장단점
    - advantage : 같은 architecture에 대한 다른 implementations
    - disadvantage : 때때로 새로운 innovation 거부
  - binary compatibility is extraordinary important ?
    -
- application binary interface
  - ISA plus system sw interface
- implementation

⇒ 추상화를 이해하기 위해 배울 게 많음

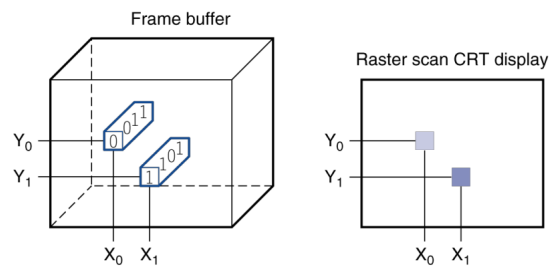
## ▼ 1.4 Components of Computer



### ▼ through the looking glass

LCD screen: picture elements (pixels)

- Mirrors content of frame buffer memory



### ▼ touchscreen

PostPC device

Supersedes keyboard and mouse

Resistive and Capacitive types

- Most tablets, smart phones use capacitive
- Capacitive allows multiple touches simultaneously



### ▼ inside the processor(CPU)

- datapath : performs operations on data
- control unit : 명령어에 따라 datapath 제어, 메모리, I/O 장치가 할 일을 제시
- cache memory : small fast SRAM memory → 데이터에 더 빠르게 접근

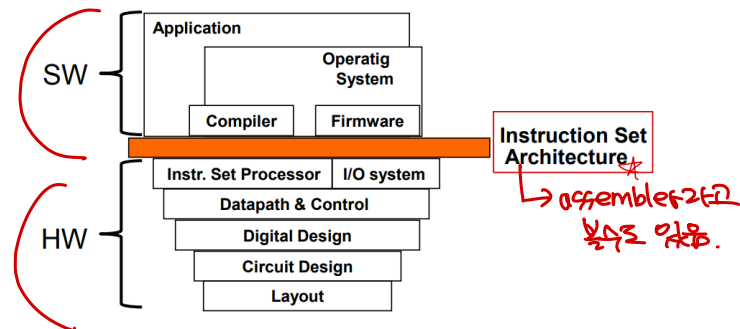
### ▼ a safe place for data

- Volatile main memory : 전원 꺼지면 data, instruction 모두 저장 x
- Non-Volatile secondary memory
  - magnetic disk
  - flash memory
  - optical disk (CDROM, DVD)

### ▼ Networks

- communication, resource sharing, nonlocal access
- LAN(local area)
- WAN(wide area)
- wireless : wifi, bluetooth

### ▼ Computer Architecture



많은 level의 abstraction 집합 → H/W와 S/W의 interface.

- ISA가 가장 중요 ★  
→ instruction set architecture.

\*ISA

: h/w and low level sw 간의 interface.

표현되는 instructions, machine language bit patterns.