

과제는 2~3주로 들어들 수 있음!  
중간고사는 1주차화요일  
기말고사는 16주차로 넘어갈 수 있음

# 컴퓨터구조론

Computer Architecture

## 01 컴퓨터 시스템 개요

박숙영

blue@sookmyung.ac.kr

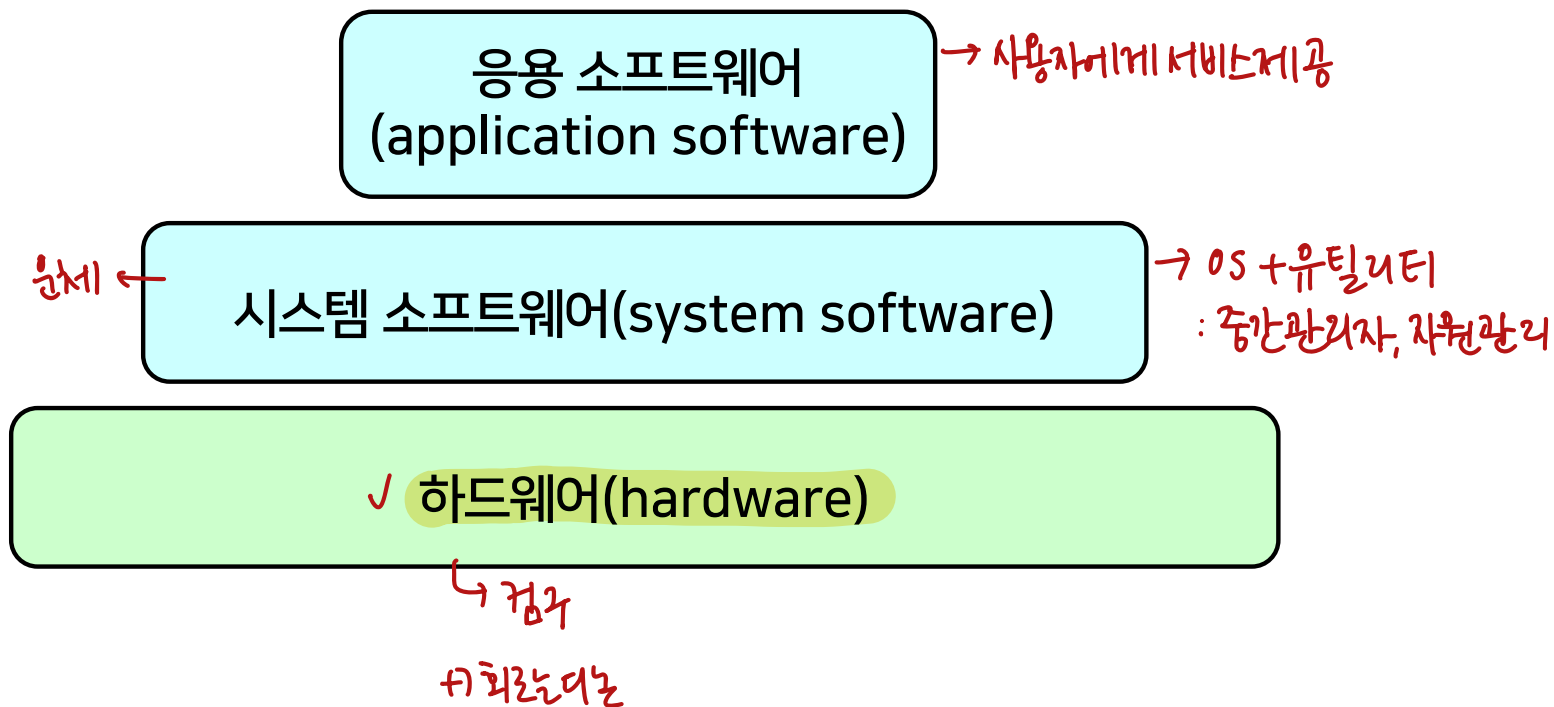
# 컴퓨터시스템 개요 *overview*

---

- 1.1 컴퓨터의 기본 구조
- 1.2 정보의 표현과 저장
- 1.3 시스템의 구성
- 1.4 컴퓨터구조의 발전과정 *검라이*

# 컴퓨터의 기본 구조

## ■ 컴퓨터시스템의 구성



# 하드웨어와 소프트웨어

## ■ 하드웨어(hardware)

- 컴퓨터 정보들의 전송 통로를 제공해 주고, 그 정보에 대한 처리가 실제 일어나게 해주는 물리적인 실체들

시스템버스

↓ 실체들만 보지 마세요

## ■ 소프트웨어(software)

작업지시

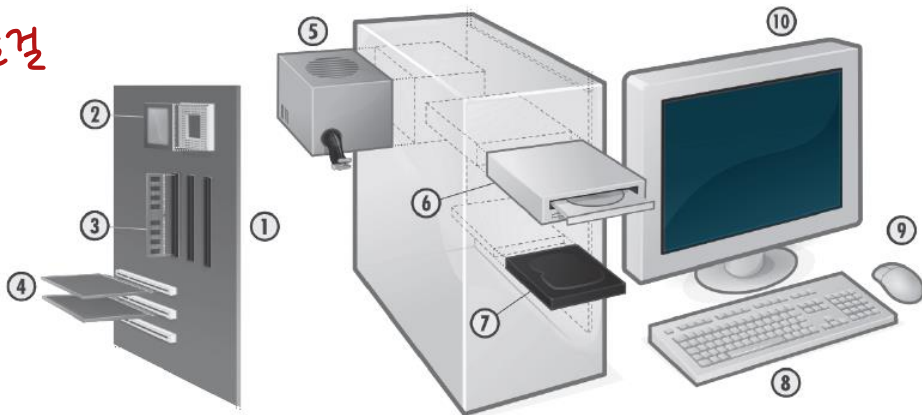
- 정보들이 이동하는 방향과 정보 처리의 종류를 지정해주고, 그러한 동작들이 일어나는 시간을 지정해주는 명령(command)들의 집합 = 프로그램

시스템 소프트웨어(system software): 운영체제(Windows 11, Unix, Linux, MacOS 등), 각종 유틸리티 프로그램들(프린터 드라이버, 등)

응용 소프트웨어(application software): 워드프로세서, 웹브라우저, MS-Excel, 등

# 컴퓨터 하드웨어의 주요 요소들

선으로 연결해야 하는 것  
칩형식으로  
⇒ 거리↓, 속도↑

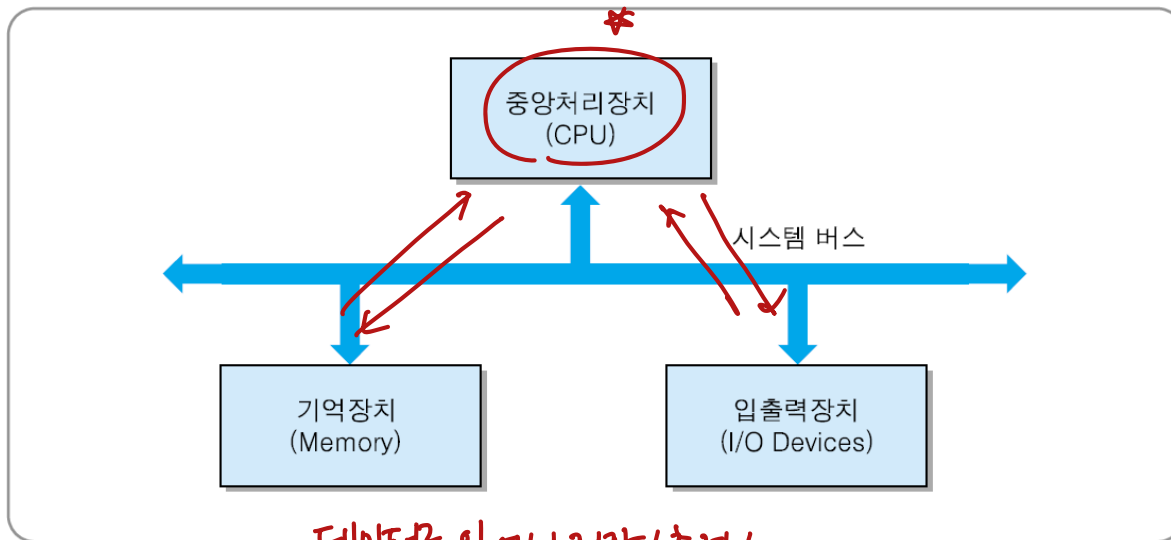


- ① 메인 보드(main board)=마더보드
- ② CPU 및 GPU 칩
- ③ 주기억장치 모듈
- ④ 확장 보드: 사운드 카드 등 그래픽
- ⑤ 전원공급장치(power supply)
- ⑥ 광 저장장치: CD-ROM, DVD
- ⑦ 하드 디스크, SSD 외부보조저장장치(영구)
- ⑧ 키보드
- ⑨ 마우스
- ⑩ 디스플레이 모니터

# 컴퓨터의 기본 구조

초기의 컴퓨터 목적: 계산

- 컴퓨터의 기능: 프로그램 코드를 정해진 순서대로 수행
  - 데이터를 읽어서(read), 처리(processing)하고, 저장(store)한다
- 주요 구성요소들: 중앙처리장치, 기억장치, 입출력장치



데이터를 읽어서 저장/출력/...

# 컴퓨터의 기본적인 구성요소들

## ★ ■ 중앙처리장치(Central Processing Unit: CPU)

- 프로세서(processor)
- '프로그램 실행'과 '데이터 처리'라는 중추적인 기능의 수행을 담당하는 요소
- 컴퓨터의 특성과 성능에 가장 큰 영향을 미치는 구성 요소

→ CPU (그래픽처리) → 슈퍼컴퓨터, AI, ...의 성능

## ■ 기억장치

- (1) 주기억장치(main memory)
  - CPU 가까이 위치하며, 반도체 기억장치 칩들로 구성
  - 고속 액세스 → 빠른 읽기/쓰기
  - 가격이 높고 면적을 많이 차지 → 저장 용량의 한계
  - 영구 저장 능력이 없기 때문에, 일시적 저장장치로만 사용  
: 휘발성 메모리

# 컴퓨터의 기본적인 구성요소들 (계속)

- (2) 보조저장장치(auxiliary storage device)
  - 2차 기억장치(secondary memory)
  - 저장 밀도가 높고, 비트 당 가격이 낮음
  - 읽기/쓰기 속도가 느림 → 메인보드와 케이블로 연결돼있음
  - 영구 저장 능력을 가진 저장장치 : 하드 디스크(hard disk), 플래시 메모리(flash memory), SSD(solid-state drive), CD-ROM, 등

↙ : 비휘발성 메모리

상대적으로 비싼  
보조기억장치



# 컴퓨터의 기본적인 구성요소들 (계속)

---

## ■ 입출력장치(I/O device) = 주변장치 (CPU, 메모리 외의 것)

- 입력장치(input device)와 출력장치(output device)의 통칭
- 사용자-컴퓨터 간의 상호작용(interaction)을 위한 장치들
- CPU는 해당 장치 제어기(device controller)를 통하여 동작 제어
- 주변장치(peripheral device)라고도 부름

↳ CPU는 강제 제어  
CPU가 너무 빠르니까

## 1.2 정보의 표현과 저장

전기가 들어온다/안들어온다

- 컴퓨터 정보: 2진수 비트들로 표현된 프로그램 코드와 데이터
- 프로그램 코드

low  
level

- 기계어(machine language)
  - 기계 코드(machine code)라고도 함
  - 컴퓨터 하드웨어 부품들이 이해할 수 있는 언어로서, 2진 비트들로 구성
- 어셈블리 언어(assembly language)
  - 기계어 명령어에 대해 사람이 알아보기 쉬운 니모닉 기호를 정해서 사람이 좀 더 쉽게 컴퓨터의 행동을 제어할 수 있도록 한 것  
ex) ADD, STOR(store), ...
  - 어셈블러(assembly)로 번역되며, 기계어와 일대일 대응

high  
level

(자연어에  
가깝음)

- 고급 언어(high-level language)
  - 영문자와 숫자로 구성되어 사람이 이해하기 쉬운 언어
  - C, C++, PASCAL, Python 등
  - 컴파일러(compiler)를 이용하여 기계어로 번역
- 의미가 명확해야 함 → 수학식 논리, 규칙

# 프로그램 언어의 번역 과정

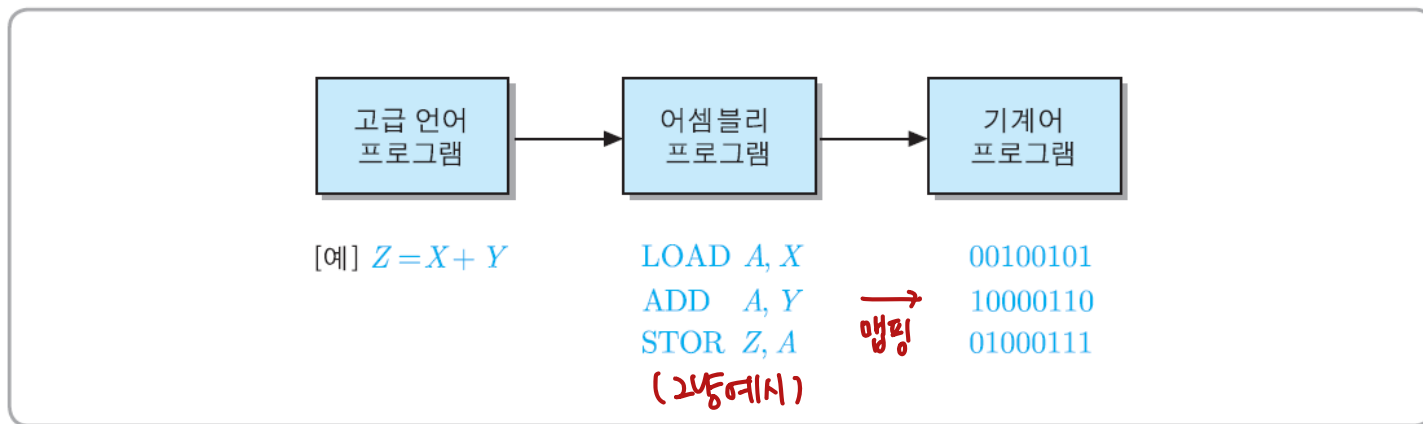
## ■ [예] $Z = X + Y$

- LOAD A, X : 기억장치 X번지의 내용을 읽어서, 레지스터 A에 적재(load)
- ADD A, Y : 기억장치 Y번지 내용을 읽어서, 레지스터 A에 적재된 값과 더하고, 결과를 다시 A에 적재
- STOR Z, A : 그 값을 기억장치 Z 번지에 저장(store)

어셈블리어  
ex) STA

임시저장소

AC (어큐뮬레이터) : 경우에 따라 명시하지 않아도 됨



# 프로그램 언어 번역 소프트웨어

---

## ■ 컴파일러(compiler)

- 고급언어 프로그램을 기계어 프로그램으로 번역하는 소프트웨어

언어를 개발한다 == 컴파일러/인터프리터를 개발한다

## ■ 어셈블러(assembler)

- 어셈블리 프로그램을 기계어 프로그램으로 번역하는 소프트웨어
- 니모닉스(mnemonics)
  - 어셈블리 명령어가 지정하는 연산을 가리키는 알파벳 기호
  - 'LOAD', 'ADD', 'STOR' 등

# 기계어(machine language)의 형식

## ■ [예]

연산코드	오퍼랜드
0 0 1	0 0 1 0 1

## ■ 연산 코드(op code)

- CPU가 수행할 연산을 지정해 주는 비트들
- 비트 수 = '3'이라면, 지정될 수 있는 연산의 최대 수:  $2^3 = 8$ 개

## ■ 오퍼랜드(operand)

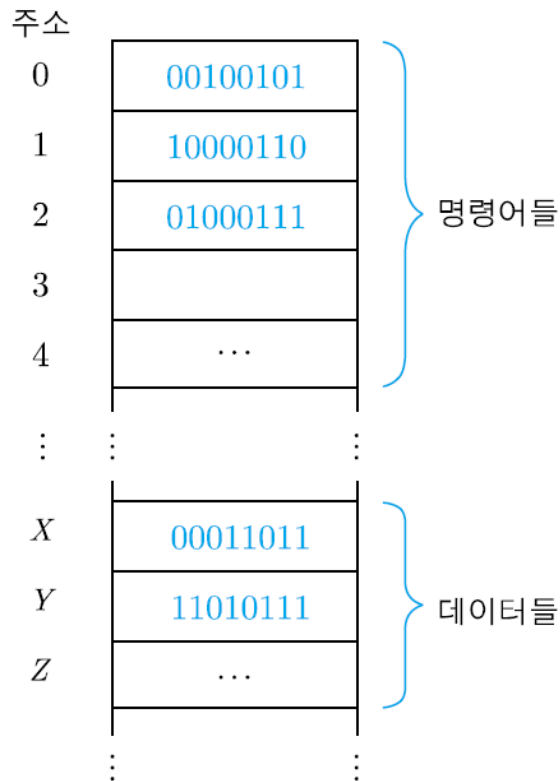
- 연산에 사용될 데이터 혹은 그것이 저장되어 있는 기억장치 주소(memory address)
- 비트 수 = '5' 라면, 주소지정(addressing) 할 수 있는 기억 장소의 최대 수:  $2^5 = 32$  개

# 프로그램 코드와 데이터의 기억장치 저장 → 메모리

## ■ 프로그램 코드(명령어)와 데이터는 지정된 기억 장소에 저장

### ■ 단어(word) 단위로 저장

- 단어: 각 기억 장소에 저장되는 정보의 기본 단위로서, CPU에 의해 한 번에 처리될 수 있는 비트들의 그룹
- 단어 길이의 예: ~~8비트~~, ~~16비트~~, 32비트, 64비트
- 주소지정 단위: 단어 단위 혹은 바이트(byte) 단위



# 1.3 시스템의 구성

---

## ■ 1.3.1 CPU와 기억장치의 접속

### ■ 시스템 버스(system bus)

- CPU와 시스템 내의 다른 요소들 사이에 정보를 교환하는 통로

- 기본 구성

- 주소 버스(address bus)
- 데이터 버스(data bus)
- 제어 버스(control bus)

케이블 / 보드판의 선들 / 칩의 접합부 / ...



# 시스템 버스

## ■ 주소 버스(address bus)

- CPU가 외부로 발생하는 주소 정보를 전송하는 신호 선들의 집합
- 주소 선의 수는 CPU와 접속될 수 있는 최대 기억장치 용량을 결정 *ex) 16비트면 16line (동시에 16바이트동)*
- [예] 주소 버스의 비트 수 = 16 비트라면,  
최대  $2^{16} = 64K$  개의 기억 장소들의 주소지정 가능

## ■ 데이터 버스(data bus)

- CPU가 기억장치 혹은 I/O 장치와의 사이에 데이터를 전송하기 위한 신호 선들의 집합
- 데이터 선의 수는 CPU가 한 번에 전송할 수 있는 비트 수를 결정
- [예] 데이터 버스 폭 = 32 비트라면, CPU와 기억장치 간의 데이터 전송은 한 번에 32 비트씩 가능



# 시스템 버스 (계속)

---

- 제어 버스(control bus)

- CPU가 시스템 내의 각종 요소들의 동작을 제어하기 위한 신호 선들의 집합
- [예]
  - 기억장치 읽기/쓰기(Memory Read/Write) 신호
  - I/O 읽기/쓰기(I/O Read/Write) 신호
  - 인터럽트(Interrupt) 신호
  - 버스 제어(Bus Control) 신호

# CPU와 시스템 버스 간의 접속

메모리에 있는 정보가 주소값이라면? (ex. 포인터)

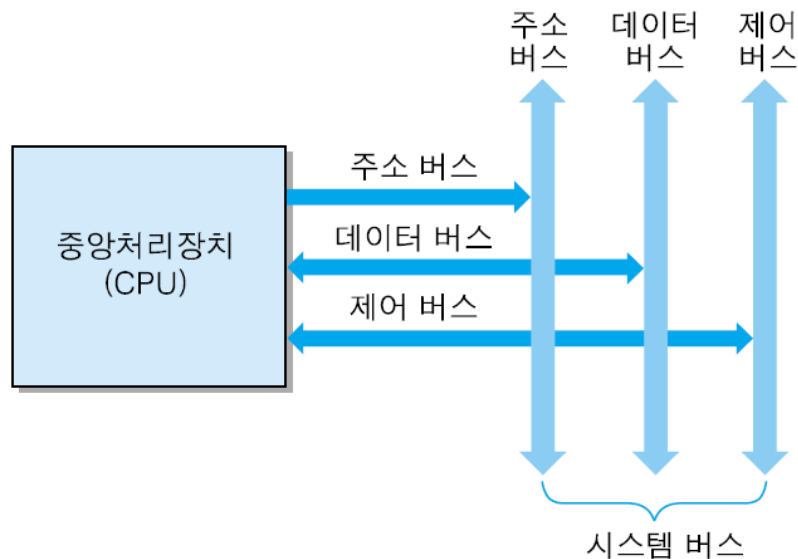
→ CPU가 판단하므로 주소버스는 무조건 단방향

## ■ 주소 버스 : 단방향성(unidirectional)

- 주소는 CPU로부터 기억장치 혹은 I/O 장치들로 보내지는 정보

## ■ 데이터 버스, 제어 버스 : 양방향성(bidirectional)

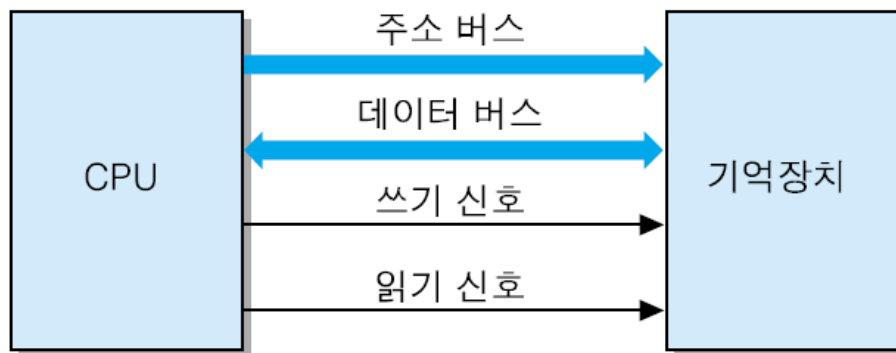
- 읽기와 쓰기 동작을 모두 지원



# CPU와 기억장치 간의 접속

## ■ 필요한 버스 및 제어신호

- 주소 버스 : 목적지
- 데이터 버스 : 어떤데이터를?
- 제어 신호: 기억장치 읽기(memory read) 신호, 기억장치 쓰기(memory write) 신호  
(CPU의 control unit이)



# CPU와 기억장치 간의 접속 (계속)

- 기억장치 쓰기 동작 : CPU가 데이터를 저장할 기억 장소의 주소와 저장할 데이터를 각각 주소 버스와 데이터 버스를 통하여 보내는 동시에, 쓰기 신호를 활성화



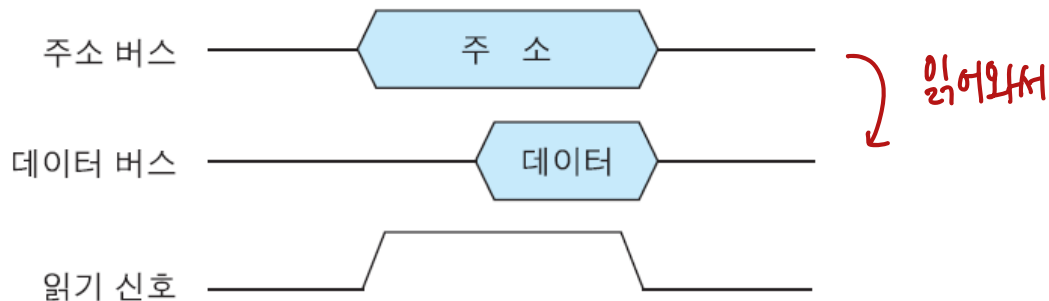
(a) 기억장치 쓰기 동작의 시간 흐름도

- 기억장치 쓰기 시간(memory write time)
  - CPU가 주소와 데이터를 보낸 순간부터 저장이 완료될 때까지의 시간

# CPU와 기억장치의 접속 (계속)

## ■ 기억장치 읽기 동작

- CPU가 기억장치 주소를 주소 버스를 통하여 보내는 동시에, 읽기 신호를 활성화
- 일정 지연 시간이 경과한 후에 기억장치로부터 읽혀진 데이터가 데이터 버스 상에 실리고, CPU는 그 데이터를 버스 인터페이스 회로를 통하여 읽음



(b) 기억장치 읽기 동작의 시간 흐름도

## ■ 기억장치 읽기 시간(memory read time)

- 주소를 발생한 시간부터 기억장치의 데이터가 CPU에 도착할 때까지의 시간  
(올리고읽게)

## 1.3.2 CPU와 I/O 장치의 접속

---

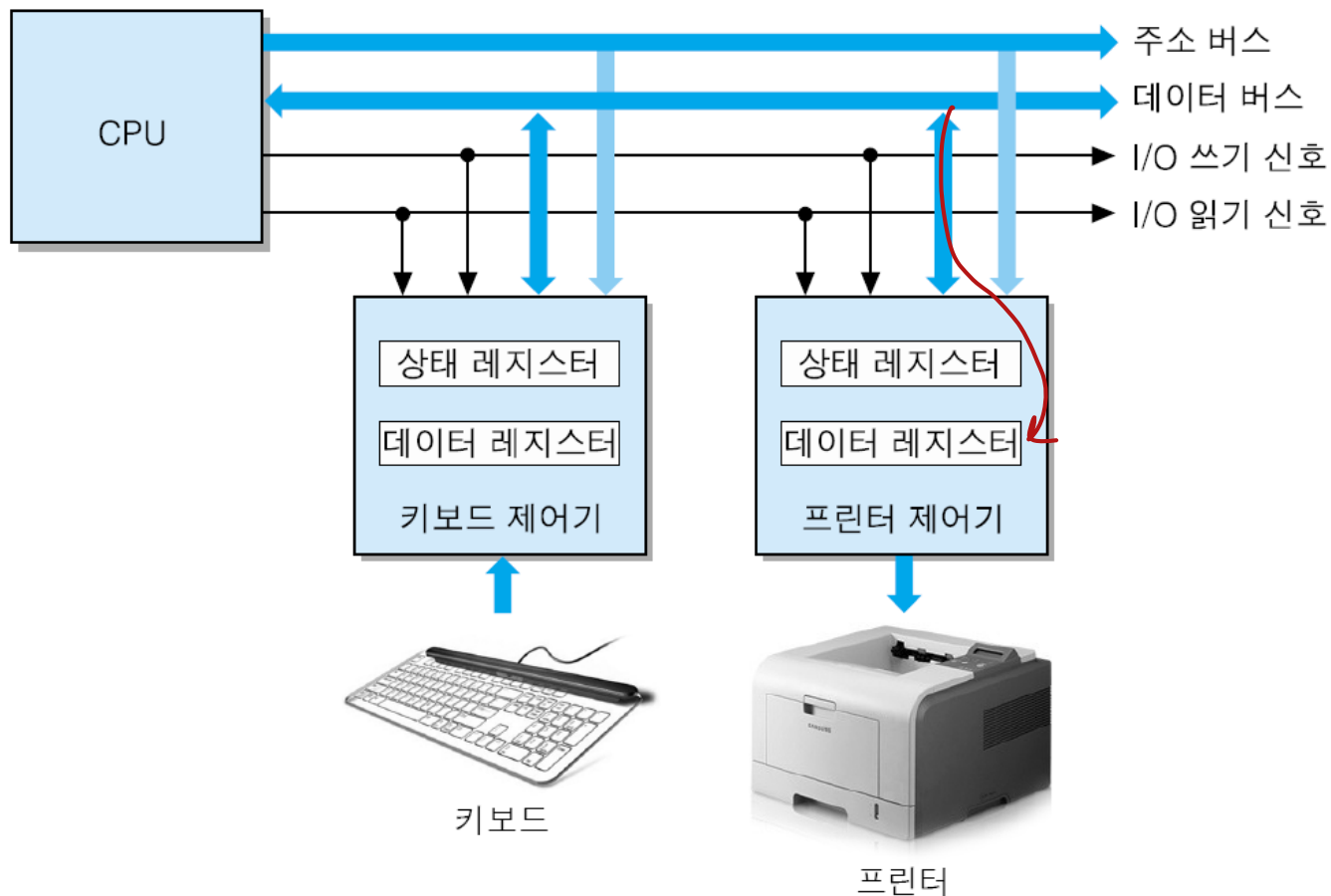
### ■ 필요한 버스 및 제어 신호

- 주소 버스
- 데이터 버스
- 제어 신호 : I/O 읽기 신호, I/O 쓰기 신호

### ■ 접속 경로: CPU $\leftrightarrow$ 시스템 버스 $\leftrightarrow$ I/O 장치 제어기 $\leftrightarrow$ I/O 장치

CPU가 너무빠르니까

# I/O 장치 접속 사례 : CPU - 키보드 & 프린터



# I/O 장치 제어기(I/O device controller)

---

- 기능: CPU로부터 I/O 명령을 받아서, 해당 I/O 장치를 제어하고, 데이터를 이동함으로써 명령을 수행하는 전자회로 장치 (예: 키보드 제어기, 프린터 제어기, 등)
- 상태 레지스터
  - I/O 장치의 현재 상태를 나타내는 비트들을 저장한 레지스터
  - 준비 상태(RDY) 비트, 데이터 전송확인(ACK) 비트, 등
- 데이터 레지스터
  - CPU와 I/O 장치 간에 이동되는 데이터를 일시적으로 저장하는 레지스터



# 키보드의 데이터 입력 과정

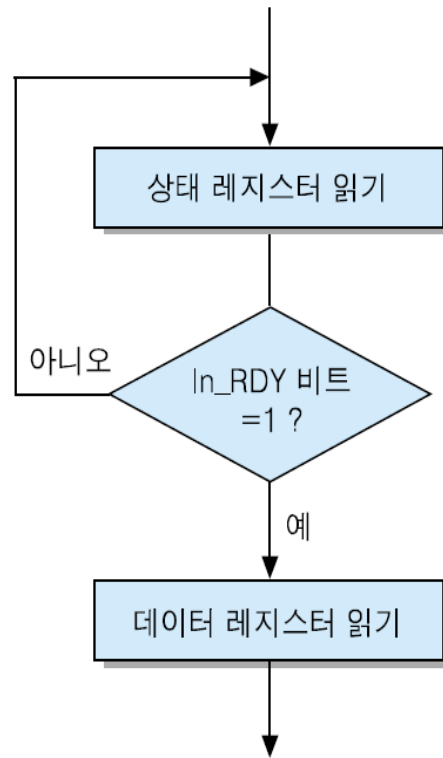
## ■ 키보드 제어기

- 키보드의 어떤 한 키(key)를 누르면,  
그 키에 대응되는 ASCII 코드가 키보드 제어기의 데이터 레지스터에  
저장되고, 동시에 상태 레지스터의 In\_RDY 비트가 1로 세트

(ready)

## ■ CPU

- 키보드 제어기로부터 상태 레지스터의 내용을 읽어서 In\_RDY 비트가  
세트 되었는지 검사(In\_RDY 비트는 데이터 레지스터에 외부로부터  
데이터가 적재되었는지를 표시)
- 만약 세트 되지 않았으면, 1번을 반복하며 대기. 만약 세트 되었다면,  
데이터 레지스터의 내용을 읽음



# 프린터의 데이터 출력 과정

---

## ■ CPU

- 프린터 제어기의 상태 레지스터의 내용을 읽어서 Out\_RDY 비트 검사(Out\_RDY 비트: 프린터가 출력할 준비가 되었는지를 표시)
- 만약 세트 되지 않았으면, ① 번을 반복하며 대기  
만약 세트 되었다면, 프린트할 데이터를 프린터 제어기의 데이터 레지스터에 저장(데이터 블록 전체를 제어기 내의 데이터 기억장치에 저장하는 것도 가능: spooling)

## ■ 프린터 제어기

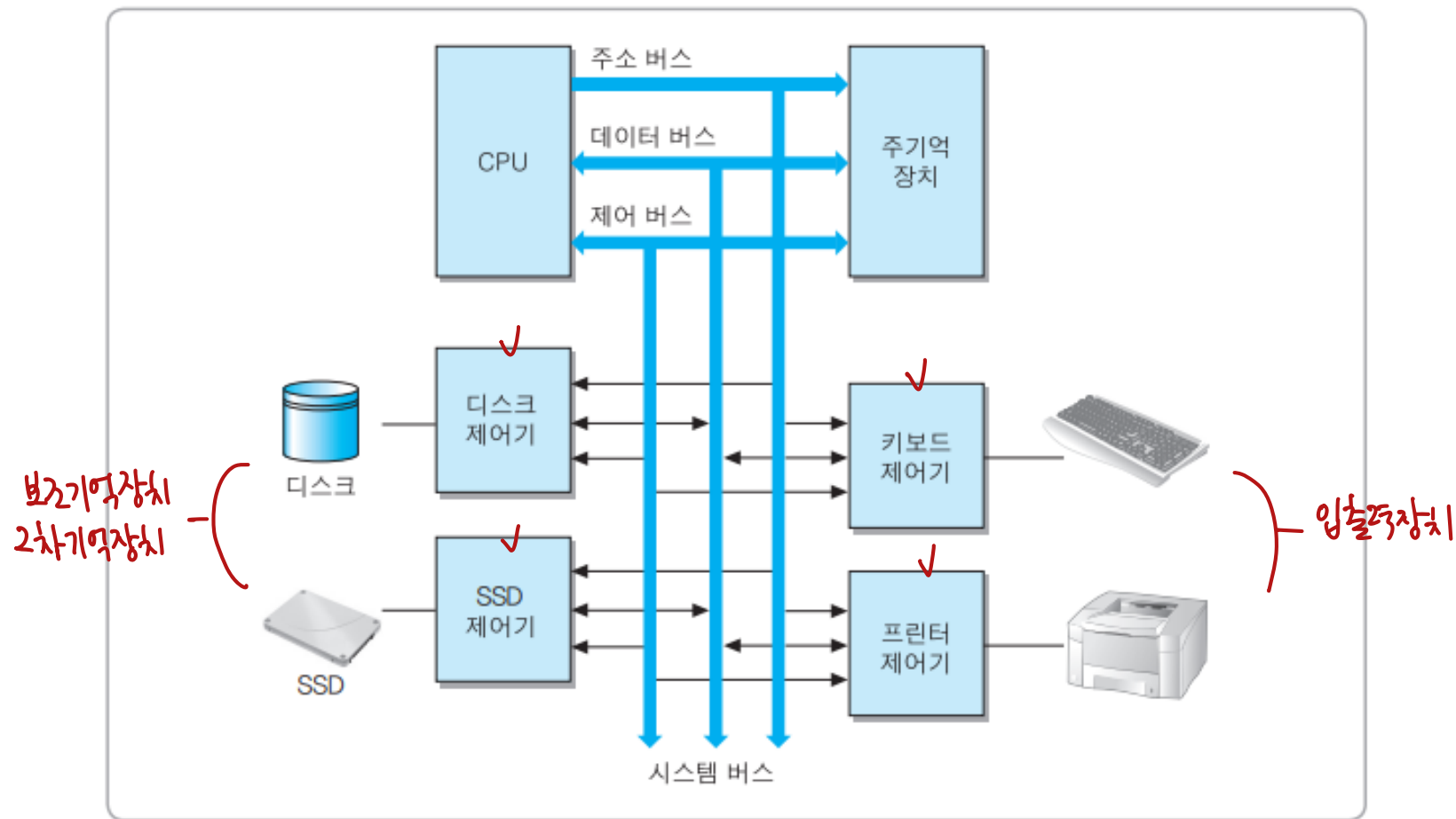
- 데이터 레지스터(혹은 데이터 기억장치)의 내용을 프린터로 보내고,
- 프린터의 하드웨어를 제어하면서 인쇄 동작 수행

# CPU와 보조저장장치의 접속

---

- 보조저장장치들(디스크, 플래시 메모리, SSD, CD-ROM 등)도 각 장치를 위한 제어기를 통하여 키보드나 프린터와 유사한 방법으로 접속
- 차이점 : 데이터 전송 단위
  - 키보드 : 바이트 단위 전송
  - 보조저장장치 : 블록(512바이트) 혹은 페이지(2K, 4K 바이트) 단위로 전송
    - → 제어기 내에 한 블록 이상을 임시 저장할 수 있는 데이터 기억장치(버퍼) 필요

### 1.3.3 컴퓨터시스템의 전체 구성



# 컴퓨터의 기본적인 기능들

→ 명령문의 집합

- 프로그램 실행 : CPU가 주기억장치로부터 프로그램 코드를 읽어서 실행
- 데이터 저장 : 프로그램 실행 결과로서 얻어진 데이터를 주기억장치에 저장
- 데이터 이동 : 디스크 혹은 SSD에 저장되어 있는 프로그램과 데이터 블록을 기억장치로 이동
- 데이터 입력/출력 : 사용자가 키보드를 통하여 보내는 명령이나 데이터를 읽어 들인다. 또한 CPU가 처리한 결과값이나 기억장치의 내용을 프린터(혹은 모니터)로 출력
- 제어 : 프로그램이 순서대로 실행되도록 또는 필요에 따라 실행 순서를 변경하도록 조정하며, 각종 제어 신호들을 발생

이 코드에 따라

## 1.4 컴퓨터 구조의 발전 과정 (컴라이) 빠르게 읽기

---

### ■ 주요 부품들의 발전 과정

- 릴레이(relay) → 진공관 → 트랜지스터 → 반도체 집적회로(IC)
- 발전 과정에서 개선된 특성들:
  - 처리속도 향상
  - 저장용량 증가
  - 크기 감소
  - 가격 하락
  - 신뢰도 향상

### ■ 초기 컴퓨터들의 근본적인 설계 개념과 동작 원리가 현대의 컴퓨터들과 거의 같음

# 최초의 컴퓨터

---

- 1642년, Blaise Pascal(프랑스)
- 덧셈과 뺄셈을 수행하는 기계적 카운터
- 다이얼의 위치에 의하여 십진수를 표시하는 6개의 원형판 세트들로 구성
- 각 원형판은 일시적으로 숫자를 기억하는 레지스터로 사용



파스칼 계산기 (1652년 제품)

이미지 출처:

[https://ko.wikipedia.org/wiki/%ED%8C%8C%EC%8A%A4%EC%B9%BC\\_%EA%B3%84%EC%82%B0%EA%B8%B0](https://ko.wikipedia.org/wiki/%ED%8C%8C%EC%8A%A4%EC%B9%BC_%EA%B3%84%EC%82%B0%EA%B8%B0)

# Leibniz의 기계

---

- 1671년, Gottfried Leibniz(독일)
- 덧셈과 뺄셈 및 곱셈과 나눗셈도 할 수 있는 계산기
- Pascal의 계산기에 두 개의 원형판들을 추가하여 반복적 방법으로 곱셈과 나눗셈을 수행
- 이후 많은 계산 기계들의 조상이 됨



라이프니츠 계산기

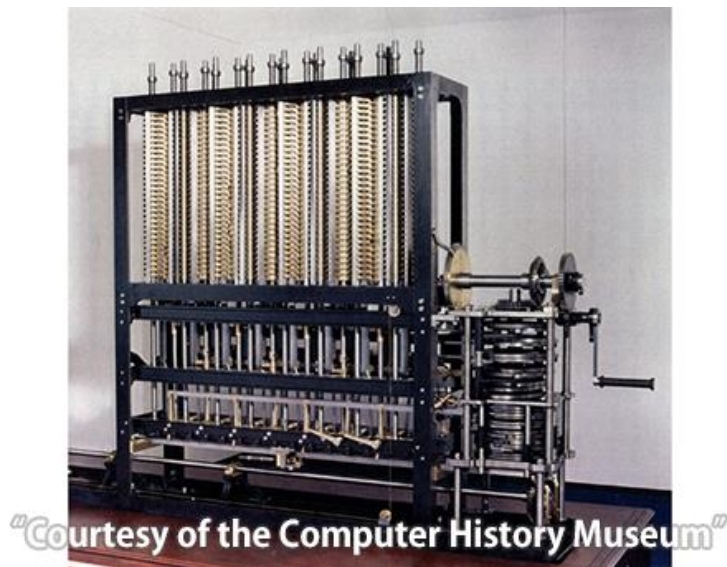
이미지 출처:

[http://wiki.hash.kr/index.php/%EB%9D%BC%EC%9D%B4%ED%94%84%EB%8B%88%EC%B8%A0\\_%EA%B3%84%EC%82%B0%EA%B8%B0](http://wiki.hash.kr/index.php/%EB%9D%BC%EC%9D%B4%ED%94%84%EB%8B%88%EC%B8%A0_%EA%B3%84%EC%82%B0%EA%B8%B0)



# Difference Engine (차분기관)

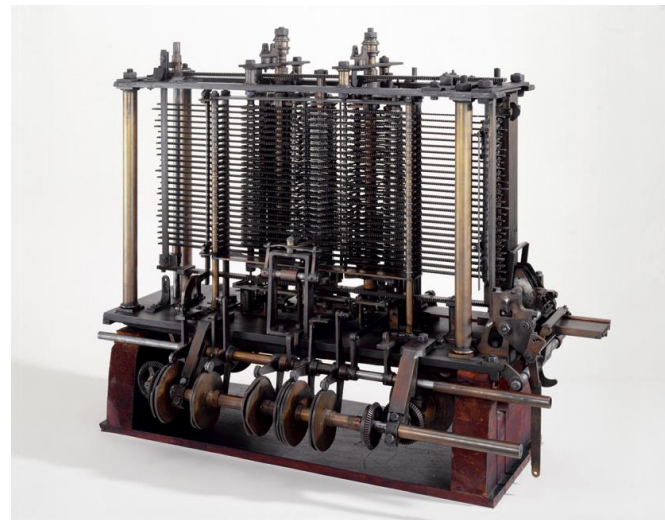
- 19세기 초, Charles Babbage(영국, 현대 컴퓨터의 할아버지)
- 표에 있는 수들을 자동적으로 계산하고, 그 결과를 금속천공기를 거쳐서 프린트
- 덧셈과 뺄셈만 수행 가능



[https://smart.science.go.kr/scienceSubject/computer/view.action?menuCd=DOM\\_000000101001007000&subject\\_sid=256](https://smart.science.go.kr/scienceSubject/computer/view.action?menuCd=DOM_000000101001007000&subject_sid=256)

# Analytical Engine 분석기관

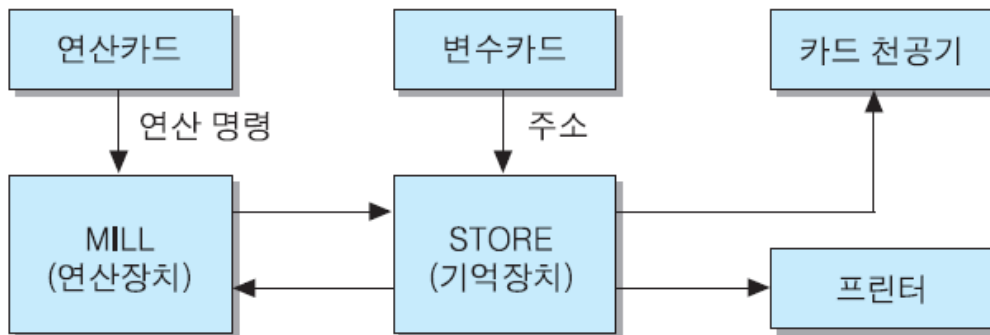
- 19세기 초, Charles Babbage(영국)
- 주요 특징들
  - 어떤 수학 연산도 자동적으로 수행할 수 있는 일반목적용 계산기계
  - 프로그래밍 가능 : 프로그램 언어 사용
  - 프로그램의 실행 순서 변경 가능
    - 수의 부호 검사를 이용한 조건 분기
    - 제어 카드를 이용한 실행 순서 변경
- 문제점
  - 주요 부품들이 기계적 장치라 속도가 느리고 신뢰도가 낮았음



<https://ko.wikipedia.org/wiki/%ED%95%B4%EC%84%9D%EA%B8%B0%EA%B4%80>

# Analytical Engine의 기본 구조

- 산술연산장치: MILL
- 기억장치: STORE
- 입력장치: 카드판독기
- 출력장치: 카드 천공기, 프린터



# ENIAC

---

- Electronic Numerical Integrator And Computer
- 1940년대 초, von Neumann(폰 노이만)이 개발
- 펜실바니아 대학에서 개발한 진공관을 사용한 **최초의 전자식 컴퓨터**
- 문제점: 프로그램의 저장 및 변경 불가능
  - ↓
- 폰 노이만의 설계 개념(**stored-program 원리**) 발표
  - 프로그램과 데이터를 내부에 저장
  - 2진수 체계(binary number system) 사용
  - 1945년 발표 후, EDVAC(Electronic Discrete Variable Computer) 개발에 실제 적용

# IAS 컴퓨터

---

- 1946년~1952년: 폰 노이만이 IAS에서 개발

- 폰 노이만: 프린스턴대학 및 IAS(Institute for Advanced Studies)의 수학 교수

- 프로그램 저장과 변경이 가능한 초기 디지털 컴퓨터

- 주요 구성요소

- 프로그램 제어 유닛(Program Control Unit) : 명령어 인출/해독
    - 산술논리연산장치(ALU)
    - 주기억장치 : 명령어와 데이터를 모두 저장
    - 입출력장치
- ) → 현재의 CPU

# IAS 컴퓨터의 구조

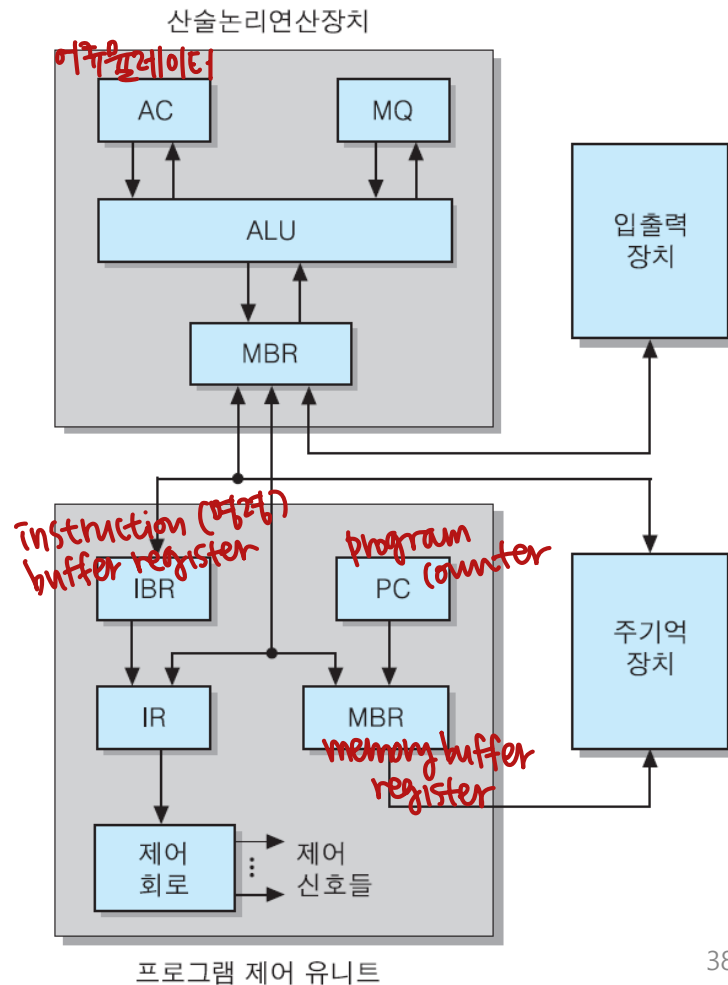
## ■ 폰 노이만 아키텍처

### (von Neumann Architecture):

- 프로그램 코드들을 기억장치에 저장된  
순서대로 **실행**하며, 그 주소는 CPU의 내부 레지스터인  
프로그램 카운터(program counter: **PC**)에  
의해 지정됨

해킹  
+  
저리

현재의 내부구조와 크게 다르지 않음



## 1.4.2 주요 컴퓨터 부품들의 발전 경위

---

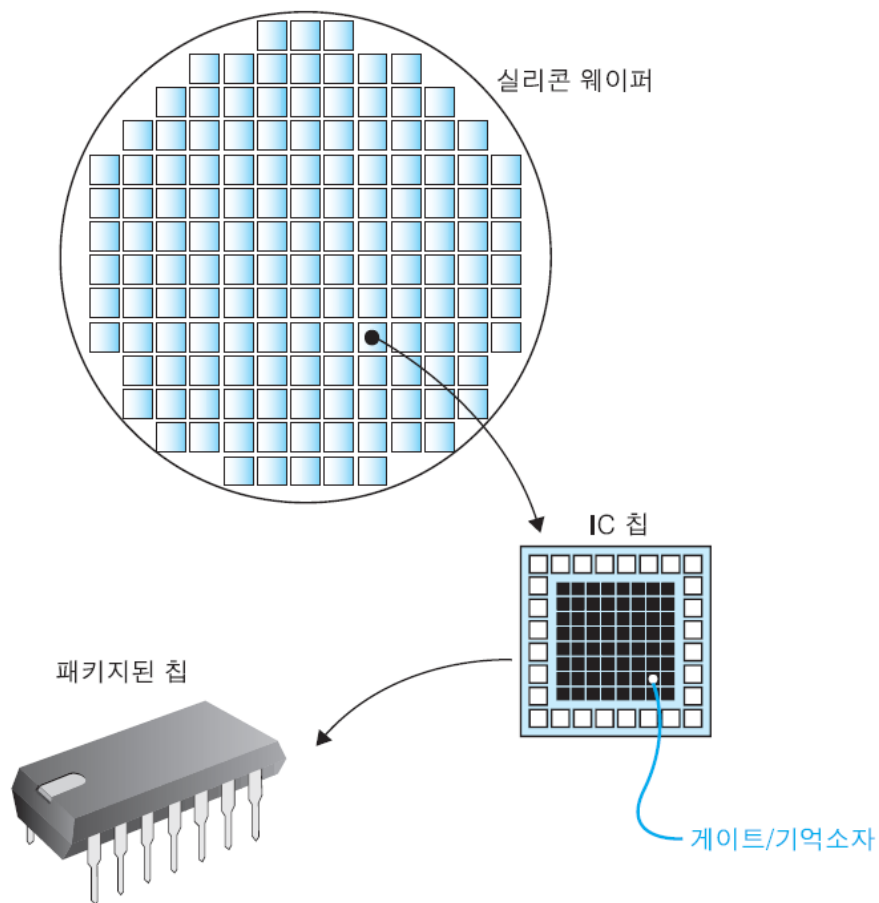
### ■ 트랜지스터(transistor)

- 초기(제1세대) 전자식 컴퓨터의 핵심 부품인 진공관을 대체한 전자 부품
- 진공관보다 작고 싸며 더 적은 열을 발산  $\downarrow$   $\uparrow$
- 반도체 재료인 실리콘(Si)으로 만들어진 고체(solid-state) 장치
- 제2세대 컴퓨터들의 핵심 부품
- 초기 컴퓨터들은 약 1000개의 트랜지스터들로 구성

### ■ 집적 회로(Integrated Circuit: IC)

- 수만 개 이상의 트랜지스터들을 하나의 반도체 칩에 집적시킨 전자 부품
- 제3세대 컴퓨터들의 핵심 부품

# IC의 제조 과정





# 집적도에 따른 IC의 분류

---

## ■ SSI(Small Scale IC)

- 수십 개의 트랜지스터들이 집적되는 소규모 IC
- 최근에는 주로 기본적인 디지털 게이트(digital gate)들을 포함하는 칩으로만 사용됨

## ■ MSI(Medium Scale IC)

- 수백 개의 트랜지스터들이 집적되는 IC
- 카운터(counter), 해독기(decoder) 또는 시프트 레지스터(shift register)와 같은 조합 회로나 순차 회로를 포함하는 칩

## ■ LSI(Large Scale IC)

- 수천 개의 트랜지스터들이 집적되는 대규모 IC
- 8-비트 마이크로프로세서 칩이나 소규모 반도체 기억장치 칩
- 제4세대 컴퓨터 분류의 계기가 됨

# 집적도에 따른 IC의 분류 (계속)

---

## ■ VLSI(Very Large Scale IC) → PC 등장

- 수만 내지 수십만 개 이상의 트랜지스터들이 집적되는 초대규모 IC
- 마이크로프로세서 칩들과 대용량 반도체 기억장치 칩

## ■ ULSI(Ultra Large Scale IC)

- 수백만 개 이상의 트랜지스터들이 집적되는 32-비트급 이상 마이크로프로세서 칩들과 수백 메가비트 이상의 반도체 기억장치 칩들 및 앞으로 출현할 고밀도 반도체 칩들을 지칭하기 위한 용어로서, VVLSI(VeryVery Large Scale IC)라고도 불림

# IC 사용에 따른 이점

---

- 전기적 통로가 짧아짐 → 동작 속도가 크게 상승
- 컴퓨터 크기의 감소
- 칩 내부의 회로들 간의 상호연결 → 부품들의 신뢰도 향상
- 전력소모 감소 및 냉각장치의 소형화
- 컴퓨터 가격 하락
  - VLSI의 출현으로 개인용 컴퓨터(PC)가 개발됨

## ■ 특징

- 소형, 저가, 성능 향상

## ■ 주요 발전 동향

- 성능이 개선된 새로운 마이크로프로세서들의 등장에 따라 PC의 성능은 계속 향상
- 8-비트, 16-비트, 32-비트, 64-비트 CPU 사용으로 단어 길이 증가
- 프로세서가 다수의 명령어 실행 유닛들 혹은 CPU 코어들을 포함하는 슈퍼스칼라, 듀얼-코어 및 멀티-코어 구조로 발전
- 칩의 집적도가 높아지면서 주변 요소들이 CPU 칩 내부에 포함됨에 따라, 속도와 신뢰도가 크게 향상
- GPU(Graphic Processing Unit)를 계산보조장치로 사용함으로써 고속 그래픽 처리 뿐 아니라 복잡한 과학기술 계산들도 높은 속도로 처리할 수 있게 됨
- 주기억장치와 보조저장장치의 용량 증가, 종류 다양화

## ■ 종류(유형)

- 데스크탑(desktop) 컴퓨터, 노트북(notebook) 컴퓨터, 넷북(netbook), 태블릿(tablet) PC, ~~포켓~~  
~~(pocket) PC~~, 등  
스마트폰

- Embedded Computer(내장 컴퓨터라고도 부름)
- 기계 장치나 전자 장치들의 내부에 포함되어, 그 장치들의 동작을 제어(control)하는 컴퓨터들
  - [예] 가전제품, 컴퓨터 주변기기, 모바일폰, 비디오 게임기 등
- 8-비트 마이크로컨트롤러(micro-controller)를 이용한 초소형부터 32-비트 컴퓨터에 이르기까지 다양
- 최소의 비용으로, 필요한 만큼의 성능 제공
- 실시간 처리(real-time processing)
- IoT(Internet of Things) 및 지능형 로봇의 핵심 요소로 사용되어 더욱 다양해지고 보급도 확대

## ■ 워크스테이션(workstation)

- CPU : 64-비트 마이크로프로세서 사용
- 고속 그래픽 처리 하드웨어 포함
- 주요 응용 : 3차원 동영상처리, 시뮬레이션, 컴퓨터 이용 설계(CAD), 등
- OS : UNIX, LINUX

## ■ 슈퍼미니컴퓨터(Super-minicomputer)

- 시스템 구조 : 다중프로세서(multiprocessor) 구조
- CPU의 수 : 20 ~ 30 개
- 성능 : VAX-11 미니컴퓨터 성능의 수십 배 이상
- OS : UNIX, LINUX (multiprocessing, multi-user 지원)
- 서버(server)급 시스템의 다운사이징(downsizing)화 주도

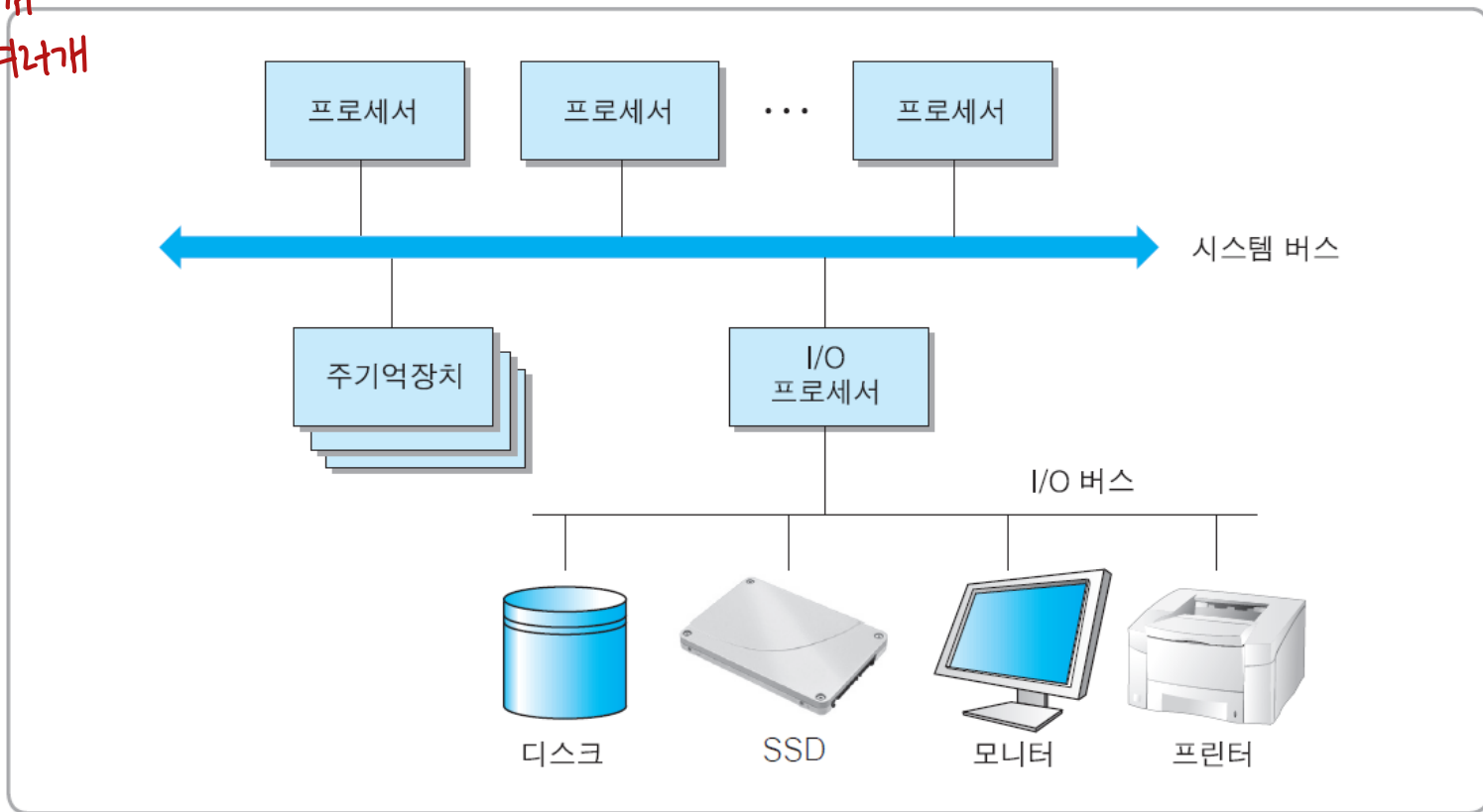
상대적으로 mini  
매우 큼...

- → 네트워크에 접속된 다수의 중형급 컴퓨터 시스템들을 응용(혹은 용도)별로 구분하여 사용하는 컴퓨팅 환경이 가능해지게 함

# 다중프로세서시스템의 구조

컴퓨터시스템의 분류와 발전 동향

CPU 여러개  
메모리 여러개



# 메인프레임 컴퓨터(mainframe computer)

컴퓨터시스템의 분류와 발전 동향

예전의 용어

- 중앙집중식 컴퓨팅을 위한 대형 컴퓨터
- IBM 360 및 370 계열, 3081, 3090 등으로 계속 발전하였으며, 최근 시스템 보안 및 통신 기능이 대폭 보강된 IBM zEnterprise 계열 출현
- 대용량 저장장치 보유
- 다중 I/O 채널을 이용한 고속 I/O 처리 능력 보유
- 대규모 데이터베이스 저장 및 관리용으로 사용
- 정부기관, 은행, 대형 인터넷포탈사이트 등에서 대규모 데이터베이스(빅데이터) 저장 및 관리용으로 사용

클라우드 사용이 힘든 경우



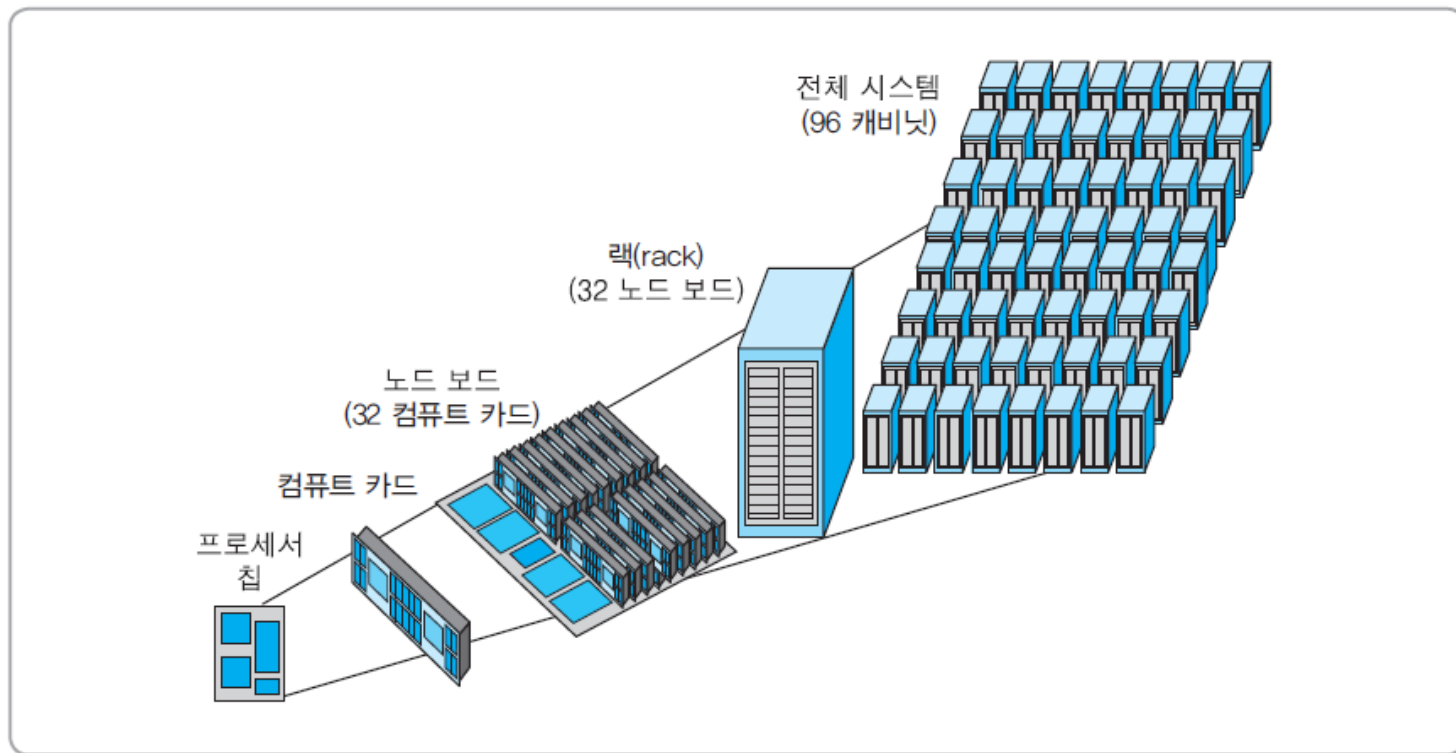
- 현존하는 컴퓨터들 중에서 처리 속도와 기억장치 용량이 다른 컴퓨터들에 비하여 상대적으로 월등한 컴퓨터 시스템들
- 분류 기준: 계속적으로 상승
  - 최초의 슈퍼컴퓨터인 CRAY-1의 속도는 100 MFLOPS
  - 최근의 슈퍼컴퓨터들의 속도는 PFLOPS급 (수백만 배 향상)
- 주요 응용 분야들 *계산이 엄청많이필요한것(클라우드 사용량↑)*
  - VLSI 회로 설계, 항공우주공학, 천문학(일기 예보), 구조 공학, 유전 탐사, 핵공학, 인공지능, 입체 영상처리 등과 같은 대규모 과학 계산 및 시뮬레이션

## ■ 파이프라인 슈퍼컴퓨터(pipeline supercomputer)

- 초기의 슈퍼컴퓨터 구조
- 복잡한 초고속 연산회로들이 포함된, 적은 수의 CPU들로 구성
- 연산장치: 슈퍼파이프라인 구조를 이용하여 고속 벡터 계산 수행
- 대표적인 시스템들 : CRAY Y-MP, CRAY-2, Fujitsu VP2000, VPP500 등

## ■ 대규모 병렬컴퓨터(massively parallel computer)

- 상호연결된 수백 혹은 수천 개 이상의 범용 프로세서들로 구성
- 프로세서들이 하나의 큰 작업을 분담하여 동시에 처리하는 병렬처리(parallel processing) 기술 이용
- 시스템 사례 : IBM BlueGene/Q 슈퍼컴퓨터
  - 2018년 11월 국제공인 슈퍼컴퓨터 TOP500 리스트([www.top500.org](http://www.top500.org)) 최상위 랭크
  - 1,572,864개의 64-비트 PowerPC 프로세서들 탑재
  - 속도: 20 PFLOPS, 기억장치: 1.57 PByte
  - 96 캐비닛 x 512 노드 x 32 프로세서



# 구조적 특징에 따른 슈퍼컴퓨터의 분류 (계속)

컴퓨터시스템의 분류와 발전 동향

- IBM Summit 슈퍼컴퓨터(2018년 11월 TOP500 List 세계 1위)
  - 컴퓨트 노드: 24-코어 POWER9 CPU 및 NVIDIA Volta GPU들로 구성 → 노드당 40 TFLOPS 성능
  - 전체 3400 노드 → 코어 수 = 240만 개
  - 최고 속도: 200 PFLOPS
  - 초고속 연결망인 NVLink를 이용하여 CPU-GPU 결합
  - 혼합형 계산(heterogeneous computing) 방식 이용

## ■ 클러스터 컴퓨터(Cluster Computer)

- 고속 LAN이나 네트워크스위치에 의해 서로 연결된 PC들 혹은 워크스테이션들의 집합체
- 클러스터 미들웨어(Cluster middleware)를 이용하여 노드(단위 컴퓨터)들에 포함된 모든 자원들을 단일 시스템 이미지(Single System Image: SSI)로 통합
- 저렴한 가격으로 고성능 고신뢰 병렬컴퓨팅 환경 구축 가능
- 대형 웹서버 및 슈퍼컴퓨터 설계 개념으로 널리 사용중

## 클러스터 컴퓨터의 구성도

