컴퓨터 구조

1장 컴퓨터 시스템 개요1

안형태

anten@kumoh.ac.kr

디지털관 139호

컴퓨터 시스템 개요

학습 목표

- 컴퓨터의 기본 구조와 동작 원리 학습
- CPU와 기억 장치 및 I/O 장치 등 전체 컴퓨터시스템의 구성 학습
- 컴퓨터 분류 방법 학습
- 컴퓨터 발전 동향 학습

학습 순서

- 컴퓨터의 기본 구조
- 시스템의 구성
- 정보의 표현과 저장
- 컴퓨터 구조의 발전 과정

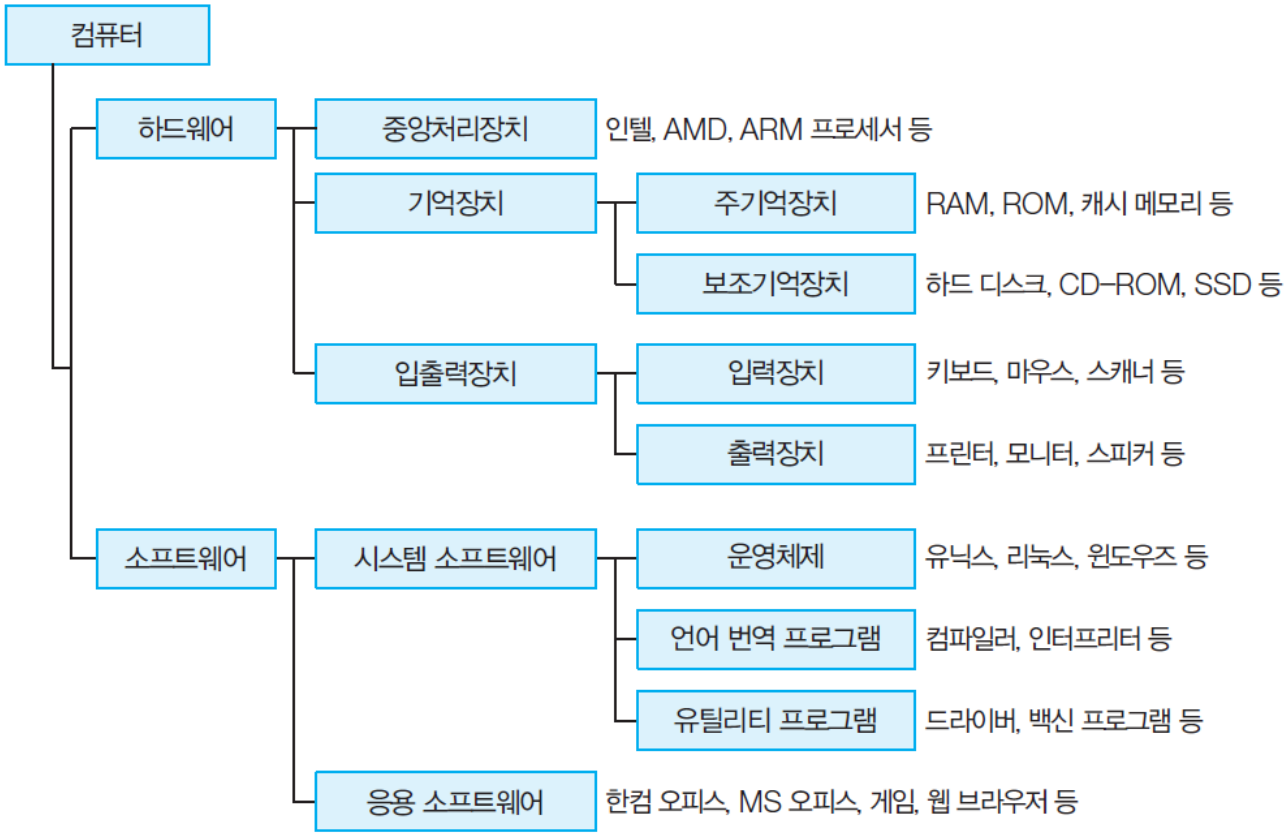
컴퓨터 시스템 개요

1. 컴퓨터의 기본구조

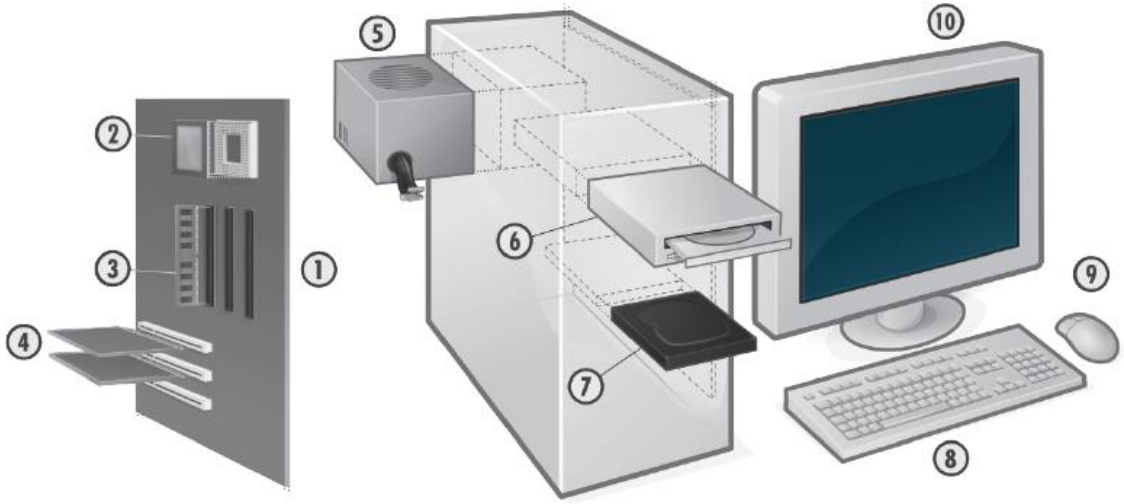
컴퓨터 시스템의 구성

컴퓨터의 기본 구성

- **하드웨어:** 컴퓨터에서 각종 정보의 전송 통로를 제공해 주고, **정보에 대한 처리가 실제 일어나게 해주는 물리적인 실체들**
- **소프트웨어:** 정보들이 이동하는 방향과 정보 처리의 종류를 지정해주고, 그러한 동작들이 일어나는 시간을 지정해주는 **명령(command)들의 집합**



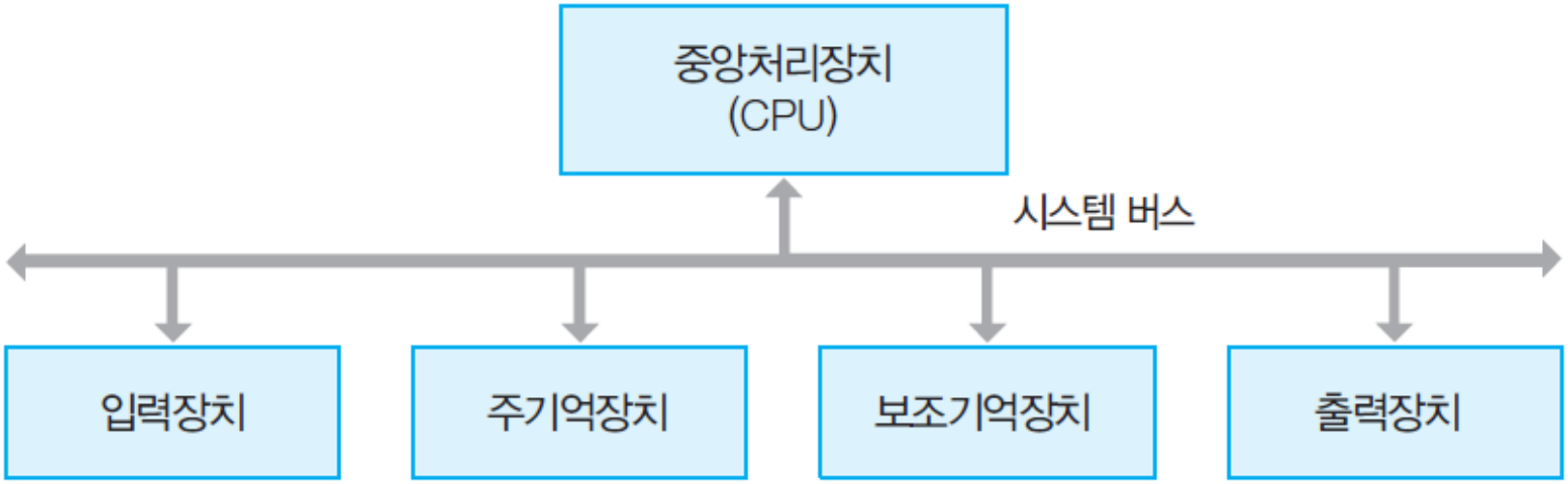
컴퓨터 하드웨어의 주요 요소들



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

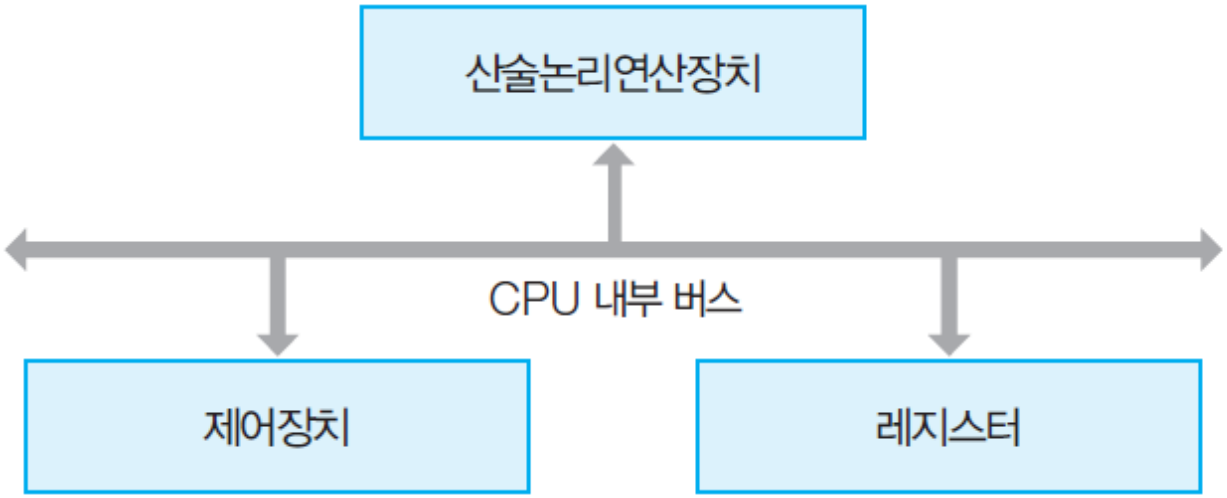
컴퓨터 시스템의 하드웨어

- 주요 하드웨어 구성 요소들
  - **중앙 처리 장치(Central Processing Unit, CPU)**
  - **기억 장치(Memory):** 주기억장치, 보조기억장치
  - **입출력 장치(Input/Output Device, I/O Device):** 입력 장치, 출력 장치
  - **시스템 버스**
    - 컴퓨터의 기능을 수행하기 위해 각 구성 요소들은 시스템 버스를 통해 상호 연결



중앙 처리 장치(CPU)

- 컴퓨터의 특성을 결정하며, 컴퓨터의 핵심 기능인 **프로그램 실행**과 **데이터 처리**를 담당
  - 프로세서(processor) 또는 마이크로프로세서(microprocessor)라고도 부름
  - **산술 논리 연산 장치(Arithmetic and Logic Unit, ALU):** 산술 연산, 논리 연산, 보수 연산, 시프트 연산을 수행
  - **제어 장치 (Control Unit, CU):** 프로그램의 명령어를 해독하여 명령어 실행에 필요한 제어 신호를 발생시키고 컴퓨터의 모든 장치를 제어
  - **레지스터(register):** 중앙 처리 장치 내부에 있는 데이터를 일시적으로 보관하는 임시 기억 장치로, 프로그램 실행 중에 사용되며 고속으로 액세스함



기억 장치

주기억장치(Main Memory, MM)

- 반도체 칩으로 구성되어 **고속 접근(access)**이 가능하지만, **가격이 비싸고 면적을 많이 차지 → 저장 용량에 한계**가 존재
- 프로그램 실행 중에 일시적으로만 사용되는 **휘발성(volatility)** 메모리
  - 전원이 꺼지면 데이터가 지워짐
- CPU ↔ 시스템 버스 ↔ 주기억장치
  - CPU 내부의 레지스터들과 시스템 버스가 직접 연결됨

보조기억장치: 2차 기억 장치(secondary memory)

- 하드 디스크, SSD(solid state drive), 플래시 메모리(flash memory)와 같은 비휘발성 메모리이며, **저장 밀도가 높고 저가이지만 속도가 느림**
- CPU에서 당장 필요하지 않은 많은 양의 데이터나 프로그램을 저장
- CPU가 직접 접근(읽기/쓰기)하지 못하고 **별도의 제어기를 통해 접근**
  - CPU ↔ 시스템 버스 ↔ 보조기억장치의 제어기 ↔ 보조기억장치

입출력 장치

- 사용자와 컴퓨터 간의 **상호작용(interaction)**을 위한 장치
  - CPU가 직접 데이터를 교환하지 못하고 **별도의 제어기를 통해 접근**

- 입력 장치(input device): 데이터를 전자적인 2진 형태로 변환하여 컴퓨터 내부로 전달
  - 키보드, 마우스, 마이크 등

- 출력 장치(output device): 중앙 처리 장치가 처리한 전자적인 형태의 데이터를 사람이 이해할 수 있는 데이터로 변환하여 출력
  - 모니터, 스피커, 프린터 등

소프트웨어

❑ 소프트웨어

- 컴퓨터를 구성하고 있는 하드웨어를 잘 동작시킬 수 있도록 제어하고, 지시하는 모든 종류의 프로그램
  - 프로그램: 컴퓨터를 사용해 어떤 일을 처리하기 위해 순차적으로 구성된 명령들의 집합
- 소프트웨어는 **시스템 소프트웨어**와 **응용 소프트웨어**로 구분

❑ 시스템 소프트웨어

- 하드웨어를 관리하고 응용 소프트웨어를 실행하는 데 필요한 프로그램
- **[예]** 운영체제(OS), 언어 번역 프로그램, 유틸리티 프로그램 등
  - 유틸리티 프로그램: 각종 주변 장치(보조기억장치, 입출력 장치)들을 구동하는 데 필요한 드라이버 프로그램, 백신 프로그램, 압축 프로그램, 디스크 조각 모음 등

[TMI] 시스템 소프트웨어 종류

운영체제 (Operating System)	• 컴퓨터 하드웨어 자원인 중앙 처리 장치, 기억 장치, 입출력 장치, 네트워크 장치 등을 제어하고 관리 • 종류: 유닉스(UNIX), 리눅스(LINUX), 윈도우(Windows), 맥 OS(MAC OS), iOS, 안드로이드 등
언어 번역 프로그램	• 고급 언어 프로그램을 컴퓨터가 이해할 수 있는 기계어로 변환하는 프로그램 • 인터프리터(interpreter): 소스 프로그램을 한 줄씩 해석하고 실행하기 때문에 실행 속도가 컴파일러보다 느릴 수 있음 ([예] JavaScript, HTML, SQL, Python 등) • 컴파일러(compiler): 전체 소스 프로그램을 한 번에 기계어로 번역하여 실행하기 때문에 실행 속도가 빠름 (C, C++, JAVA 등)
장치 드라이버 (Device Driver)	• 컴퓨터에 연결된 주변 장치를 제어하는 운영체제 모듈
링커 (Linker)	• 여러 개로 분할해 작성된 프로그램에 의해 생성된 목적 프로그램 또는 라이브러리 루틴을 결합하여 실행 가능한 하나의 프로그램으로 연결하는 프로그램
로더 (Loader)	• 하드 디스크 같은 저장 장치에 보관된 프로그램을 읽어 주기억장치에 적재한 후 실행 가능한 상태로 만드는 프로그램 • 로더는 할당, 연결, 재배치, 적재 기능을 수행

응용 소프트웨어

- 응용 소프트웨어: 애플리케이션, 앱, 어플 등으로 불림
  - 컴퓨터 시스템을 일반 사용자들이 특정한 용도에 활용하기 위해 만든 프로그램

용도	예
사무용	• 한글, MS-office 군
그래픽용	• 포토샵, 일러스트레이터
멀티미디어용	• 팟플레이어, 꿀뷰
게임용	• LOL, 발로란트, 원신, 로스트아크, 메이플
통신 및 네트워크	• 크롬, 엣지, 카카오톡, 페이스북, 인스타그램

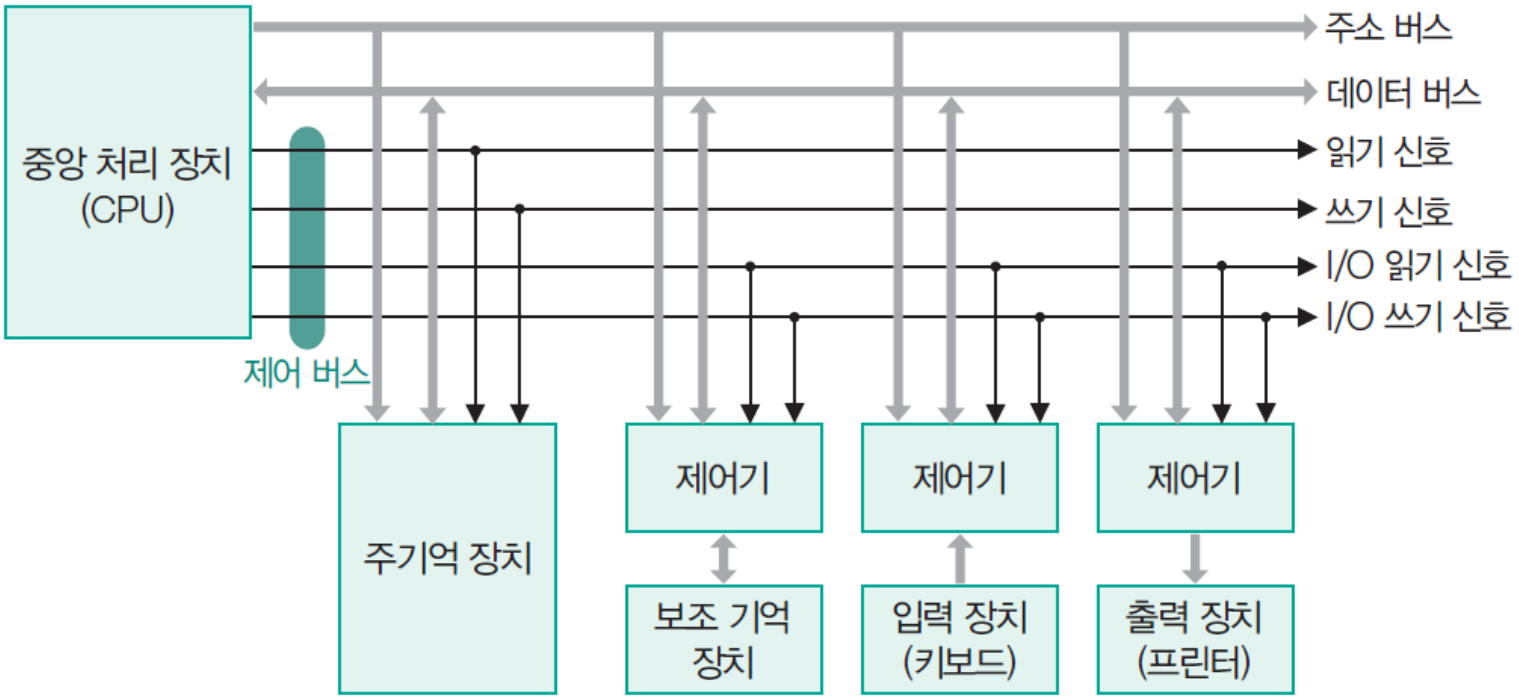
컴퓨터 시스템 개요

2. 시스템의 구성

CPU와 시스템 버스 간의 접속

시스템 버스(system bus)

- CPU와 기억 장치 및 입출력 장치 사이에 정보를 교환하는 통로
- 기본 구성
  - **주소 버스(address bus)**
  - **데이터 버스(data bus)**
  - **제어 버스(control bus)**



주소 버스(Address bus)

- CPU가 기억 장치나 입출력 장치를 지정하는 주소 정보를 전송하는 신호 선들의 집합

- **단방향(uni-directional) 전송:** 주소는 CPU로부터 기억 장치 혹은 I/O 장치로 전송되는 정보

- 주소 선의 수는 CPU와 접속될 수 있는 최대 기억 장치 용량을 결정
  - [예] 주소 버스의 비트 수 = 12 비트라면, 최대 2^{12} = 4K 개의 기억 장소들의 주소를 지정 가능

데이터 버스(Data bus)

- CPU가 기억 장치나 입출력 장치 사이에 데이터를 전송하기 위한 신호선들의 집합

- **양방향(bi-directional) 전송:** 읽기와 쓰기 동작을 모두 지원

- 데이터선의 수는 CPU가 한 번에 전송할 수 있는 데이터 비트의 수를 결정
  - [예] 데이터 버스 폭 = 32 비트라면, CPU와 기억 장치 간의 데이터 전송은 한 번에 32 비트씩 가능

제어 버스(Control bus)

- CPU가 시스템 내의 기억 장치 및 I/O 등 각종 요소의 동작을 제어하는 데 필요한 신호선들의 집합
  - 기억 장치 읽기/쓰기(Memory Read/Write) 신호
  - I/O 읽기/쓰기(I/O Read/Write) 신호
  - 인터럽트(Interrupt) 신호
  - 버스 제어(Bus Control) 신호

- 설계는 양방향이지만, 주로 단방향 사용

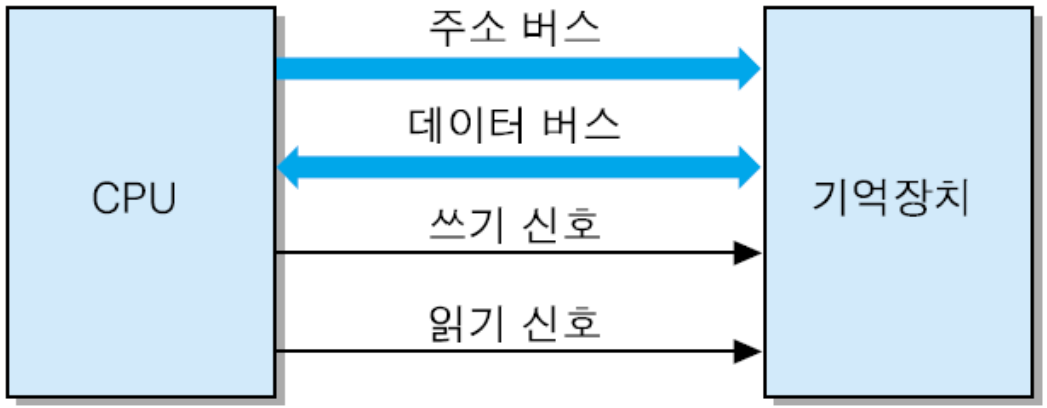
CPU와 기억 장치의 접속

액세스(access)

- CPU가 데이터를 기억 장치의 특정 장소에 저장하거나, 이미 저장된 내용을 읽는 동작

기억 장치 액세스에 필요한 버스 및 제어신호

- 주소 버스
- 데이터 버스
- 제어 신호: 기억 장치 읽기(memory read) 신호, 기억 장치 쓰기(memory write) 신호



CPU와 기억 장치 간의 접속

기억 장치 쓰기 동작

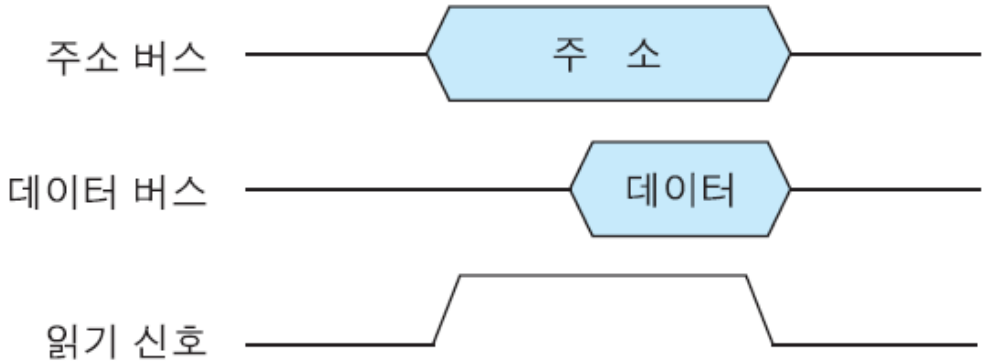
- CPU가 데이터를 저장할 기억 장소의 주소와 저장할 데이터를 각각 주소 버스와 데이터 버스를 통하여 보내는 동시에, 쓰기 신호를 활성화(activate)
- 기억 장치 쓰기 시간(memory write time): CPU가 주소와 데이터를 보낸 순간부터 데이터가 기억장치에 저장이 완료될 때까지의 시간

(a) 기억장치 쓰기 동작의 시간 흐름도

CPU와 기억 장치 간의 접속

기억 장치 읽기 동작

- CPU가 기억 장치 주소를 주소 버스를 통하여 보내는 동시에, 읽기 신호를 활성화
- 일정 지연 시간이 경과한 후에 기억 장치로부터 읽혀진 데이터가 데이터 버스 상에 실리고, CPU는 그 데이터를 버스 인터페이스 회로를 통하여 읽음
- 기억 장치 읽기 시간(memory read time): 주소를 발생한 시간부터 기억 장치의 데이터가 CPU에 도착할 때까지의 시간



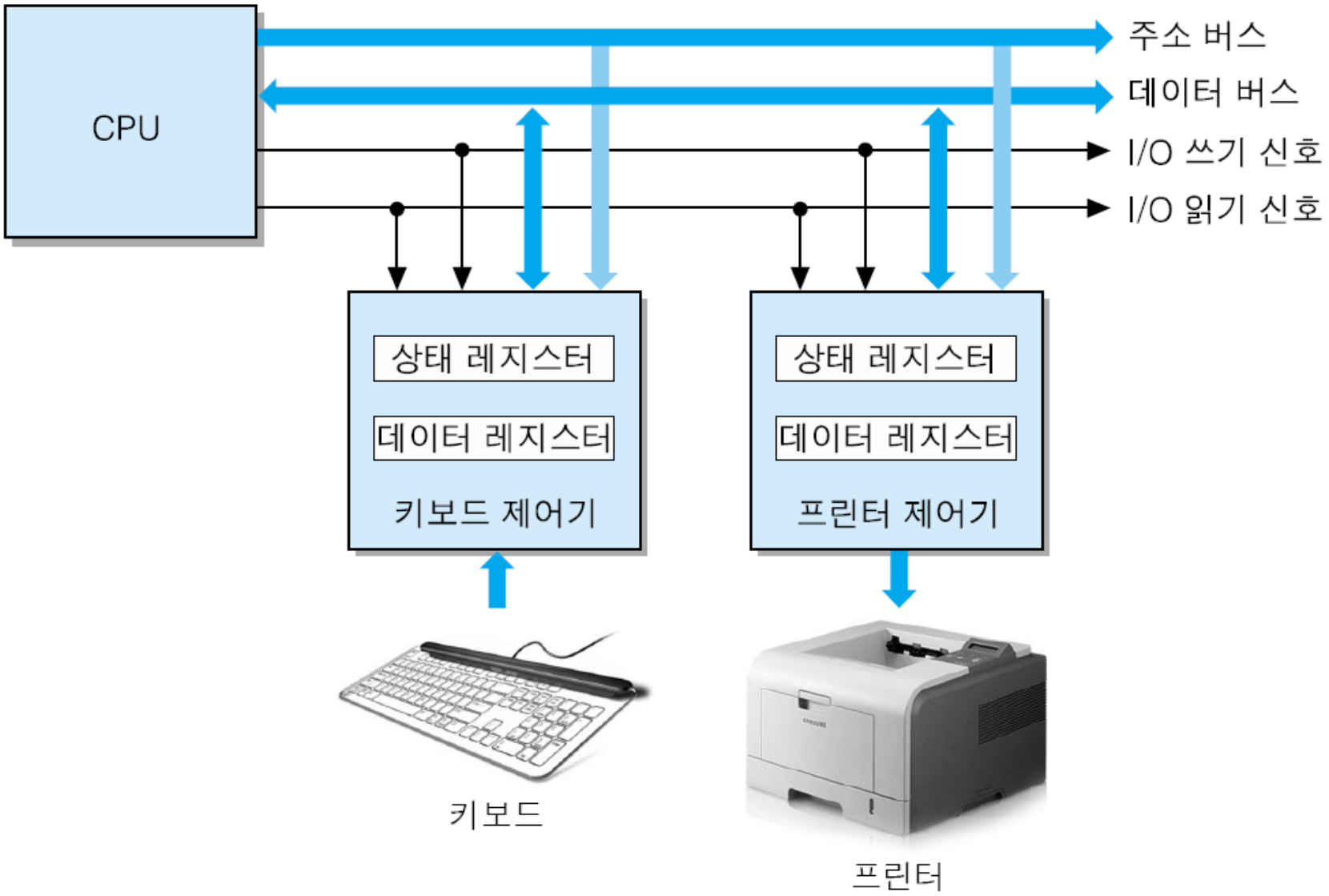
(b) 기억장치 읽기 동작의 시간 흐름도

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 ↔ 시스템 버스 ↔ I/O 장치 제어기 ↔ I/O 장치

I/O 장치 접속 사례: CPU와 키보드 & 프린터



I/O 장치 제어기(I/O device controller)

- I/O 장치 제어기
  - 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(IN_RDY) 비트, 데이터 전송 확인(acknowledgment, ACK) 비트 등

- 데이터 레지스터(데이터 버퍼)
  - CPU와 I/O 장치 간에 이동되는 데이터를 일시적으로 저장하는 레지스터

- 상태 레지스터와 데이터 레지스터는 주소를 가짐

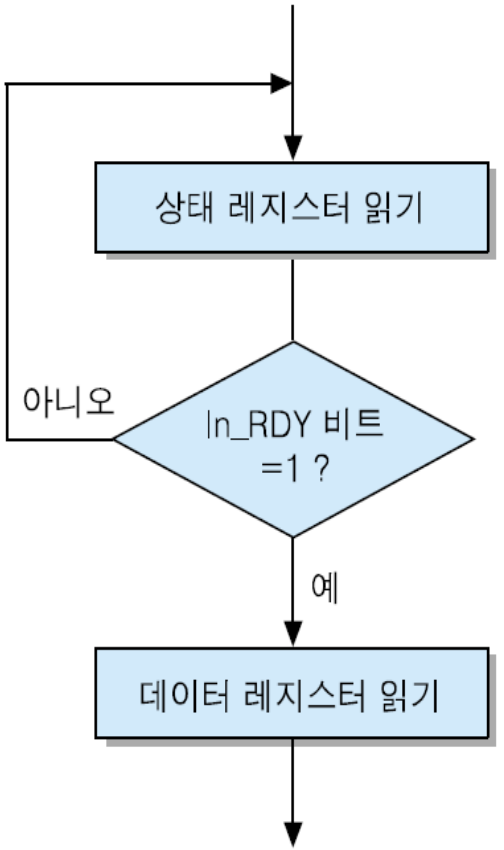
키보드의 데이터 입력 과정

키보드 제어기

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

CPU 동작

- ① 키보드 제어기로부터 상태 레지스터의 내용을 읽어서 In_RDY 비트가 1로 세트 되었는지 검사
  - In_RDY 비트는 데이터 레지스터에 외부로부터 데이터가 적재됐는지를 표시
- 만약 세트 되었다면, 데이터 레지스터의 내용을 읽음
- 만약 세트 되지 않았으면, ① 반복하며 대기

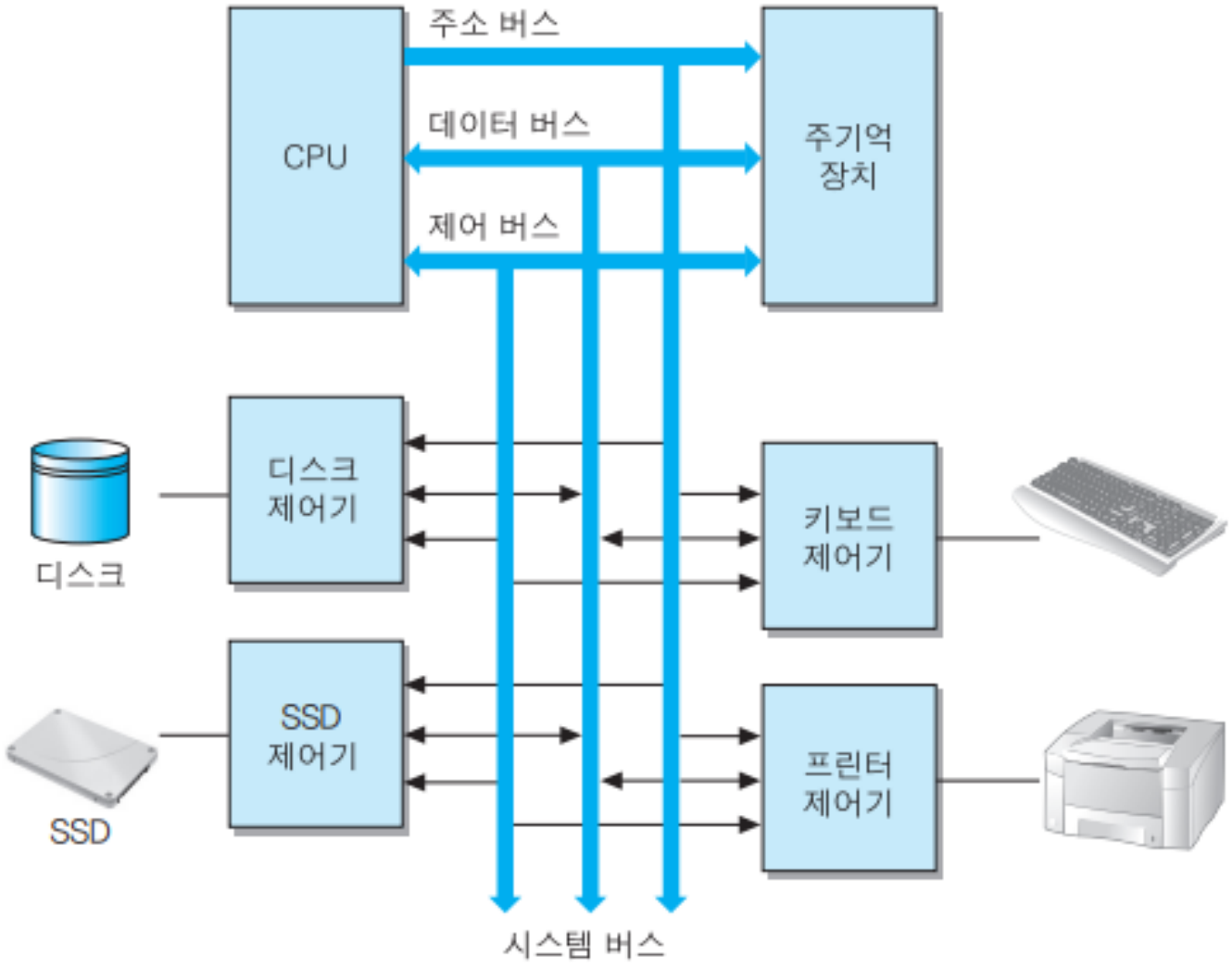


CPU와 보조기억장치의 접속

- 보조기억장치들(하드 디스크, SSD, 플래시 메모리 등)도 각 장치를 위한 제어기를 통하여 키보드나 프린터와 유사한 방법으로 접속
  - CPU는 보조기억장치의 제어기를 통해서 접속

- **차이점: 데이터 전송 단위**
  - 키보드: Byte 단위 전송
  - 보조기억장치: 블록(512Bytes) 혹은 페이지(2K, 4KBytes) 단위로 전송
    - 제어기 내에 한 블록 이상을 임시 저장할 수 있는 데이터 기억 장치(버퍼) 필요

컴퓨터 시스템의 전체 구성



컴퓨터의 기능

- 프로그램 코드를 정해진 순서대로 수행
  - 데이터를 읽어서(read), 처리(processing)하고, 저장(store)함

- 기본적 수행 기능들
  - **프로그램 실행:** CPU가 주기억장치로부터 프로그램 코드를 읽어서 실행
  - **데이터 저장:** 프로그램 실행 결과를 주기억장치에 저장
  - **데이터 이동:** 하드 디스크나 SSD에 저장되어 있는 명령어와 데이터 블록을 주기억장치로 이동
  - **데이터 입력 및 출력:** 사용자가 키보드나 마우스를 통해 입력하는 명령어나 데이터를 읽거나, CPU가 처리한 결과를 모니터나 프린터로 출력
  - **제어:** 프로그램에서 정해진 순서에 따라 실행되도록 각종 제어 신호를 발생
    - 필요에 따라서 실행 순서를 변경하도록 조정

컴퓨터 시스템 개요

3. 정보의 표현과 저장

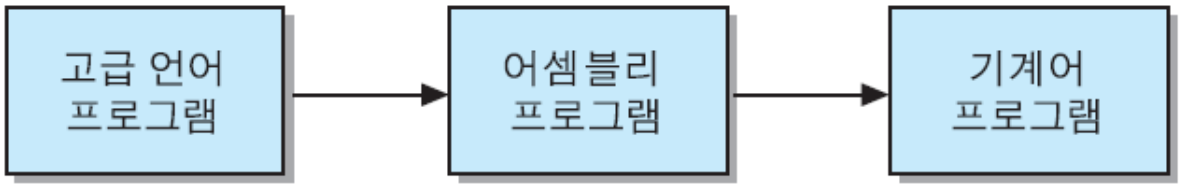
정보의 표현과 저장

- 컴퓨터 정보: 2진수 비트들로 표현된 **프로그램 코드**와 **데이터**

- **프로그램 코드**
  - **기계어(machine language)**
    - 컴퓨터 하드웨어 부품들이 이해할 수 있는 언어로 2진수 비트들로 구성
  - **어셈블리 언어(assembly language)**
    - 고급 언어와 기계어 사이의 중간 언어
    - 어셈블러(assembler)에 의해 기계어로 번역되며, 기계어와 일대일 대응
  - **고급 언어(high-level language)**
    - 영문자와 숫자로 구성되어 사람이 이해하고 작성하기 쉬운 언어
    - 컴파일러(compiler)나 인터프리터(interpreter)를 통해 기계어로 변환
    - [예] C, C++, JAVA, Python 등

프로그램 언어의 번역 과정

- 프로그램은 고급 언어 → 어셈블리어 → 기계어 순으로 변환



[예] $Z = X + Y$

LOAD A, X	00100101
ADD A, Y	10000110
STOR Z, A	01000111

- **[예]** Z = X + Y
  - LOAD A, X: 기억 장치 X번지의 내용을 읽어서, 레지스터 A에 적재(load)하라는 의미
  - ADD A, Y: 기억 장치 Y번지 내용을 읽어서, 레지스터 A에 적재된 값과 더하고, 결과를 다시 A에 적재하라는 의미
  - STOR Z, A: 그 값을 기억 장치 Z 번지에 저장(store)

프로그램 언어 번역 소프트웨어

❑ 컴파일러(compiler)

- 고급 언어로 작성된 프로그램 코드를 기계어 프로그램으로 번역하는 소프트웨어
- 전체 프로그램을 한 번에 번역하므로, 실행 전에 오류 탐지 가능
  - [예] C, C++, JAVA

❑ [TMI] 인터프리터(interpreter)

- 고급 언어로 작성된 프로그램 코드를 한 줄 씩 해석하여 바로 실행하는 소프트웨어
  - [예] Python, JavaScript

❑ 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

❑ 연산코드(operation code, OP code)

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

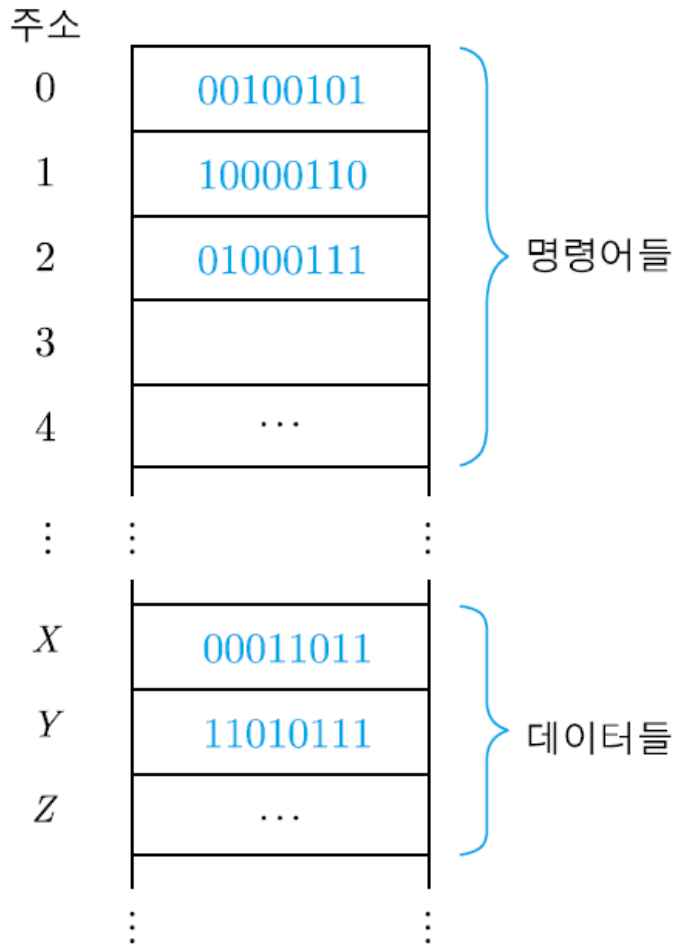
❑ 오퍼랜드(operand, 피연산자)

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

프로그램 코드와 데이터의 기억 장치 저장

- 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
  - 주소 지정 단위(addressable unit): 주소가 지정된 각 기억 장소 당 저장되는 데이터 길이
    - 단어 단위 혹은 바이트(Byte) 단위



$Z = X + Y$

LOAD *A*, *X*	00100101
ADD *A*, *Y*	10000110
STOR *Z*, *A*	01000111

컴퓨터 시스템 개요

4. 컴퓨터 구조의 발전 과정

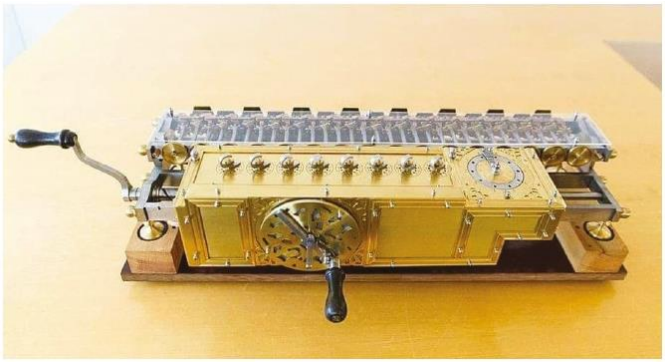
컴퓨터의 역사

출처: https://www.youtube.com/watch?v=RFADkuA0hHc

컴퓨터의 역사

초기의 계산 도구

- 계산을 하는 도구로서 가장 간단한 것은 주판
  - 기원전 약 3000년 전 고대 메소포타미아 인들이 가장 먼저 사용했다고 추정

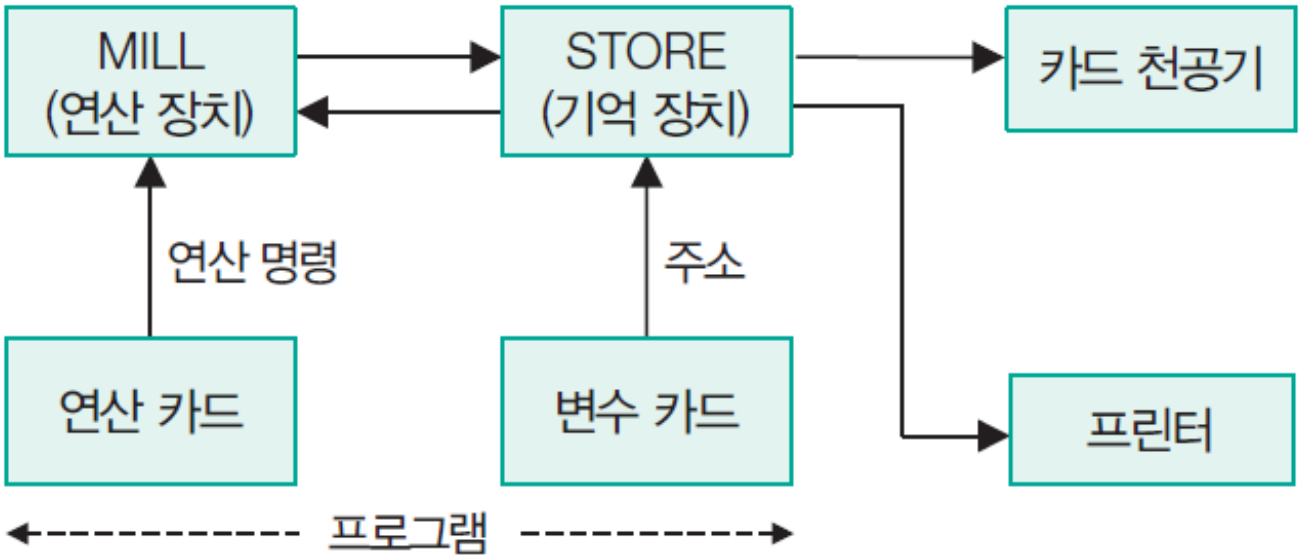


기계식 계산기

- **톱니바퀴를 이용한 기계식 계산기:** 톱니바퀴로 연결된 바퀴판들로 덧셈과 뺄셈 수행 (1642년, 파스칼)
  - 라이프니츠는 이를 개량하여 곱셈과 나눗셈도 가능한 계산기를 발명(1671년)했으며, 이후 기계 장치에 더 적합한 진법을 연구해서 17세기 후반에 2진법을 창안
- **차분 기관(difference engine):** 표에 있는 수들을 자동적으로 산술연산(덧셈, 뺄셈)하고, 그 결과를 금속 천공기를 거쳐서 프린트하는 최초의 계산 기계를 설계 (1823년, 찰스 배비지)

컴퓨터의 역사

- **해석 기관(analytical engine):** 네 가지 산술 연산 기능과 입력 및 출력장치를 모두 갖춘 최초의 일반목적용 계산기계 (1833년, 찰스 배비지)
  - 오늘날 사용하는 컴퓨터의 기본 요소를 모두 갖춤
  - 프로그래밍 가능
  - 프로그램 언어 사용
  - 제어 카드로 실행 순서 변경 가능
  - 수의 부호 검사를 이용한 조건 분기
  - **한계:** 주요 부품들이 기계적 장치라 속도가 느리고 신뢰도가 낮았음



컴퓨터의 역사

- 전기 기계식 계산기
  - **MARK-I:** 1944년 하버드 대학의 에이킨이 개발
    - 찰스 배비지의 해석 기관을 실현
    - 미해군의 탄도 계산 등 수많은 수학이나 과학 문제를 해결하는 데 공헌
- 전자식 계산기
  - **애니악(ENIAC, Electronic Numerical Integrator And Computer):** 진공관을 사용한 최초의 전자식 컴퓨터(1946년, 에커트와 모클리)
  - **애드삭(EDSAC, Electronic Delay Storage Automatic Calculator):** 10진수 체계와 프로그램 내장 방식의 계산기(1949년, 윌키스)
  - **애드박(EDVAC, Electronic Discrete Variable Automatic Computer):** 2진수 체계와 프로그램 내장 방식을 적용(1951년, 폰 노이만)
  - **유니박(UNIVAC, UNIVersal Automatic Computer):** 최초의 상용 컴퓨터, 인구조사통계국에 설치(1951년, 에커트와 모클리)
  - 컴퓨터의 발전 과정에 있어서 **진공관** 및 **프로그램 내장 방식**의 사용은 근대에서 현대로 넘어오게 되는 분기

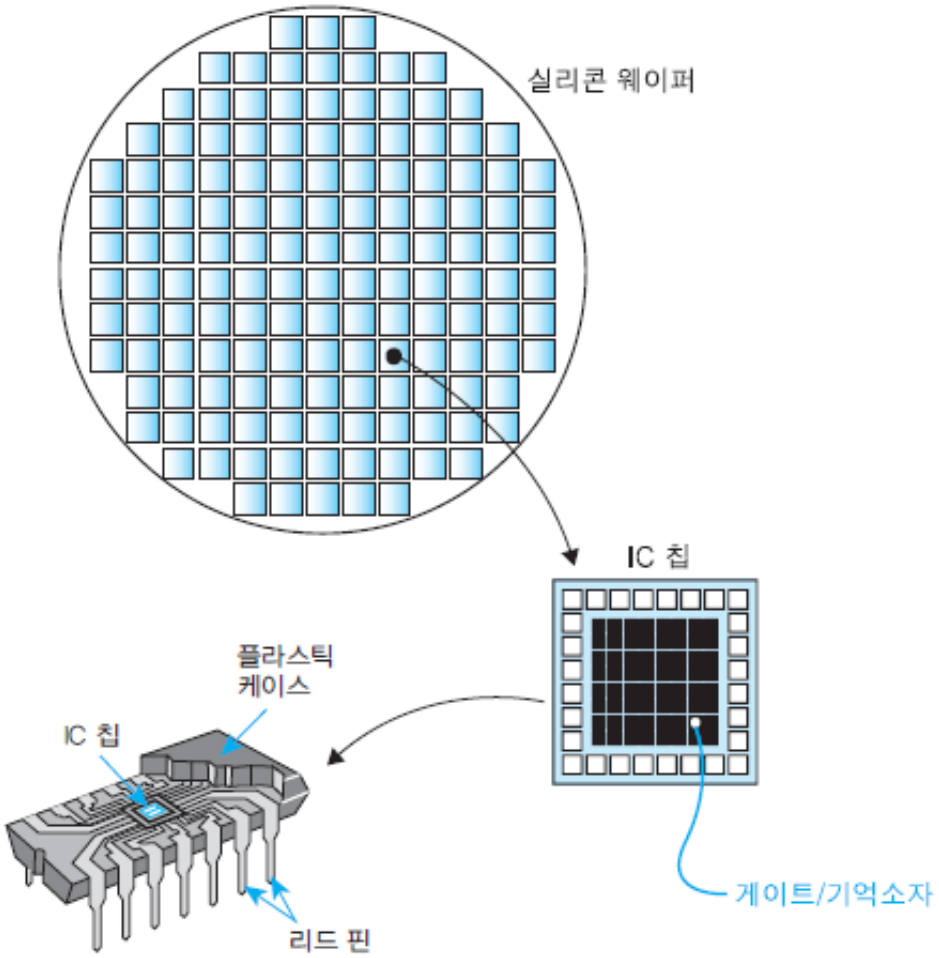
컴퓨터 구조의 발전 과정

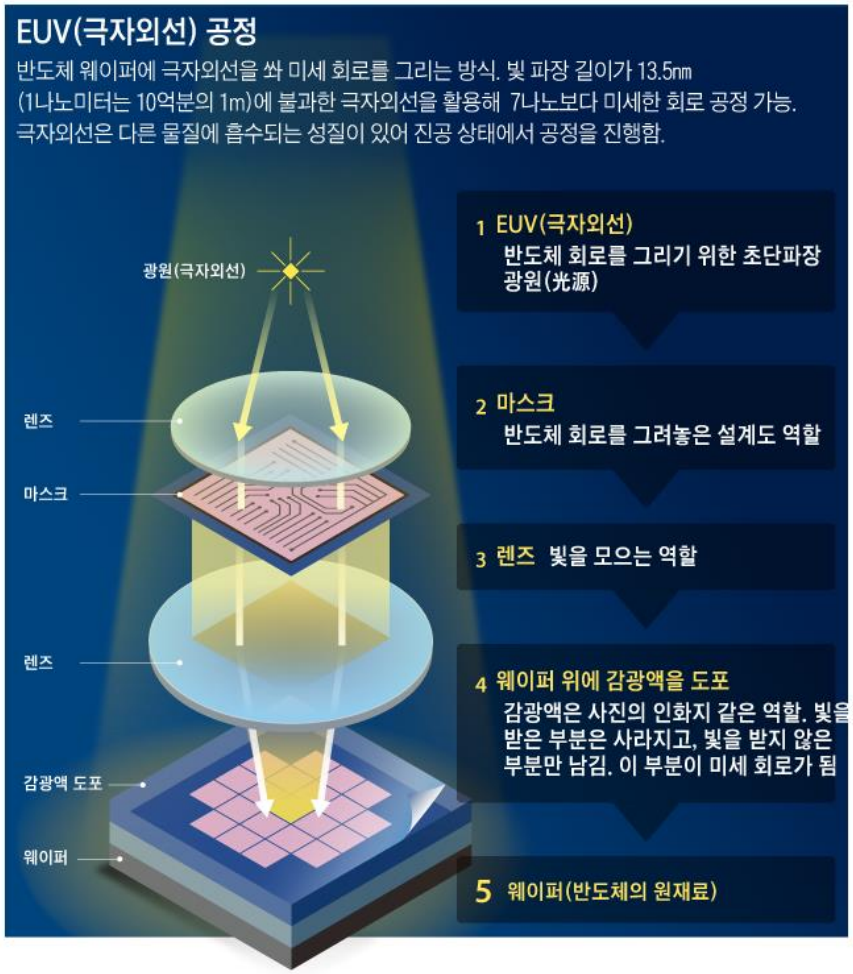
- 주요 부품들의 발전 과정
  - **릴레이(relay) → 진공관 → 트랜지스터 → 반도체 집적회로(IC)**




- **집적 회로(Integrated Circuit, IC):** 수만 개 이상의 트랜지스터들을 하나의 실리콘 반도체 칩에 집적시킨 전자 부품
  - **장점:** 처리속도 향상, 저장용량 증가, 크기 감소, 가격 하락, 신뢰도 향상
    - 회로들이 근접하여 전기적 통로가 짧아지므로 동작 속도가 크게 상승
    - 컴퓨터 크기의 감소
    - 칩 내부에서 회로들 간의 상호연결로 부품들의 신뢰도 향상
    - 전력소모 감소하여 냉각장치의 소형화
    - 컴퓨터 가격 하락

[TMI] IC 제조 과정


실리콘 웨이퍼
IC 칩
플라스틱
케이스
IC 칩
게이트/기억소자
리드 핀


EUV(극자외선) 공정
반도체 웨이퍼에 극자외선을 쏴 미세 회로를 그리는 방식. 빛 파장 길이가 13.5㎚
(1나노미터는 10억분의 1m)에 불과한 극자외선을 활용해 7나노보다 미세한 회로 공정 가능.
극자외선은 다른 물질에 흡수되는 성질이 있어 진공 상태에서 공정을 진행함.
광원(극자외선)
렌즈
마스크
렌즈
감광액 도포
웨이퍼
1 EUV(극자외선)
반도체 회로를 그리기 위한 초단파장 광원(光源)
2 마스크
반도체 회로를 그려놓은 설계도 역할
3 렌즈 빛을 모으는 역할
4 웨이퍼 위에 감광액을 도포
감광액은 사진의 인화지 같은 역할. 빛을 받은 부분은 사라지고, 빛을 받지 않은 부분만 남김. 이 부분이 미세 회로가 됨
5 웨이퍼(반도체의 원재료)

[TMI] 집적 회로의 분류

SSI (Small Scale IC)	• 트랜지스터 수십 개가 집적된 소규모 IC로, 기본 게이트 기능과 플립플롭 포함
MSI (Medium Scale IC)	• 트랜지스터 수백 개가 집적된 중규모 IC로, 디코더, 인코더, 멀티플렉서, 디멀티플렉서, 카운터, 레지스터, 소형 기억 장치 등의 기능을 포함
LSI (Large Scale IC)	• 트랜지스터 수천 개가 집적된 대규모 IC로, 8비트 마이크로프로세서나 소규모 반도체 기억 장치 칩이 이에 해당함
VLSI (Very Large Scale IC)	• 트랜지스터 수만에서 수십만 개 이상 집적된 초대규모 IC로, 대용량 반도체 메모리, 1만 게이트 이상의 논리 회로, 단일 칩 마이크로프로세서 등 • **VLSI의 출현으로 개인용 컴퓨터(PC)가 개발됨**
ULSI (Ultra Large Scale IC)	• 트랜지스터가 수백만 개 이상 집적된 극대규모 IC로, 수백 메가바이트 이상의 반도체 기억 장치 칩 등이 해당 • VLSI와 ULSI 사이의 정확한 구분은 모호

[TMI] 하드웨어 부품의 출현 기반 분류

- 컴퓨터의 발전 과정을 세대별로 명확하게 설명하기는 어렵지만, 새로운 하드웨어 부품의 출현을 기준으로 분류

구분 \ 세대	1세대	2세대	3세대	4세대	5세대
주요 소자	진공관	트랜지스터	SSI, MSI	LSI, VLSI	VLSI, ULSI
주기억 장치	자기 드럼, 수은 지연 회로	자기 코어	IC(RAM, ROM)	LSI, VLSI	VLSI
보조 기억 장치	천공 카드, 종이 테이프	자기 드럼, 자기 디스크	자기 디스크, 자기 테이프	자기 디스크, 자기 테이프	자기 디스크, 광 디스크
처리 속도	ms(10^{-3})	μs(10^{-6})	ns(10^{-9})	ps(10^{-12})	fs(10^{-15})
사용 언어	기계어, 어셈블리어	고급 언어(COBOL, FORTRAN, ALGOL)	고급 언어(LISP, PASCAL, BASIC, PL/I)	고급 언어 (ADA 등), 문제 지향적 언어	객체 지향 언어 (C++, 자바)

컴퓨터의 세대 분류

1 세대 컴퓨터

- **진공관**을 사용함에 따라 컴퓨터 크기가 매우 크며, 열 발생량과 전력 소모가 많음
- 폰 노이만(von neumann)이 제안한 **프로그램 내장** 개념을 도입
- 수치 계산, 통계 등에 사용
- 기계어와 어셈블리어를 사용
- 대표적인 컴퓨터: ENIAC, EDSAC, EDVAC, UNIVAC

2 세대 컴퓨터

- **트랜지스터**를 사용함으로써 컴퓨터는 더 고속화되고, 기억 용량이 증가했으나 크기는 소형화
- 자기 드럼이나 자기 디스크 같은 대용량의 보조기억장치가 사용
- 운영체제의 개념과 온라인 실시간 처리 방식을 도입
- 다중 프로그래밍 기법을 사용
- 과학 계산, 일반 사무용으로 사용됨
- 소프트웨어 개발에 주력한 시기로, FORTRAN, ALGOL, COBOL 등 사용

컴퓨터의 세대 분류

3세대 컴퓨터

- **집적회로**를 기본 회로 소자로 사용
- 캐시(cache) 기억 장치가 등장
- 패밀리(family) 개념의 출현에 따라 프로그램의 호환성 제공
- 시분할 처리를 통해 멀티프로그래밍을 지원

4세대 컴퓨터

- **마이크로프로세서** 등장
  - CPU 내부 회로 전체를 하나의 반도체 칩에 넣어 제조한 IC
- 가상 기억 장치의 개념이 도입
- 컴퓨터 네트워크가 발전
- 개인용 컴퓨터(PC)가 등장하여 대중화
- 온라인 실시간 처리 시스템이 보편화 및 기존 시스템에 비해 빠른 처리 속도를 갖춤

컴퓨터의 세대 분류

5세대 컴퓨터

- 5세대는 아직 명확하게 구분되지 않음
  - 부품의 집적도보다는 시스템 규모 또는 인공지능과 같은 획기적인 응용 소프트웨어의 출현에 의해 정의될 가능성이 있음
- 후보 기술들
  - 인공지능기반 응용 프로그램
  - 고도화된 다중 프로세서를 사용한 병렬 처리 컴퓨터
  - 양자 컴퓨터(quantum computer)
  - 광자 컴퓨터(optical computer)
  - 신경망 컴퓨터(neural computer)

[TMI] 무어의 법칙과 황의 법칙

❑ 무어의 법칙 (Moore's Law, 1965년)

- 반도체 집적 회로의 트랜지스터 수가 약 18~24개월마다 2배로 증가한다는 법칙
  - 반도체 성능 향상 및 가격 하락이 동시에 이루어지며, IT 산업의 급성장
- 무어의 법칙은 한계에 부딪침 (2010년)
  - 기술적인 한계(발열)뿐만 아니라 경제적인 한계(미세공정 비용)도 존재

❑ 황의 법칙 (Hwang's Law, 2002년)

- 집적 회로를 뛰어넘는 메모리의 발전으로 인해서 앞으로는 1년에 2배씩 메모리 반도체 용량이 증가할 것이라고 주장
- 이는 계속해서 NAND 플래시 계열의 메모리가 지속적으로 발전하면서 실제로 증명됨
  - 하지만 2010년, 불과 8년 만에 황의 법칙은 깨짐

Q & A

End!