

# Microprocessor (W1)

## - Basics of Microporcessor -

Dong Min Kim  
Department of IoT  
Soonchunhyang University  
dmk@sch.ac.kr

## 학습목표

- 컴퓨터 하드웨어의 구성 요소와 소프트웨어의 개념을 이해할 수 있다.
- 컴퓨터에서 사용하는 고급 언어와 기계어의 관계를 살펴본다.
- 새로운 하드웨어 부품의 출현에 따른 컴퓨터의 세대별 발전 과정을 알아본다.
- 여러 측면에서 컴퓨터를 분류하고 설명할 수 있다.
- 폰 노이만 구조와 하버드 구조를 비교하여 설명할 수 있다.

## 내용

- 01 컴퓨터 시스템의 구성
- 02 컴퓨터의 역사
- 03 컴퓨터의 분류
- 04 폰 노이만, 비 폰 노이만, 하버드 구조



〈1980년의 16B 하드디스크〉



# 과거와 현재



아폴로 11호에서 달 착륙을 유도하는 데 쓰인 컴퓨터의 클럭 스피드는 1.024MHz였다. 램은 정보의 저장 단위인 한 비트에 한 문자를 넣는다고 가정했을 때 4KB였고, 저장 공간은 72KB였다. 아폴로 11호에는 유도 컴퓨터 2대와 발사체 발사용 컴퓨터, 탈출 유도 시스템 등 4대의 컴퓨터가 쓰인 것으로 알려져 있다.



미국 반도체기업 사이프러스의 'CYPD4225' 마이크로칩을 쓰고 있는 걸로 나타났다. 이 칩의 계산 속도를 나타내는 클럭 스피드는 48MHz다. 램은 8KB를, 저장 공간은 128KB로 나타났다. 이 충전기의 가격은 54.99달러(약 6만 4800원)이다.

<https://www.epnc.co.kr/news/articleView.html?idxno=90911>  
<https://www.dongascience.com/news.php?idx=34277>

# 01 컴퓨터 시스템의 구성

## □ 컴퓨터의 기본 구성

- **하드웨어** : 컴퓨터에서 각종 정보를 입력하여 처리하고 저장하는 동작이 실제 일어나게 해 주는 물리적인 실체
- **소프트웨어** : 정보 처리의 종류를 지정하고 정보의 이동 방향을 결정하는 동작이 일어나는 시간을 지정해 주는 명령(command)들의 집합(프로그램)

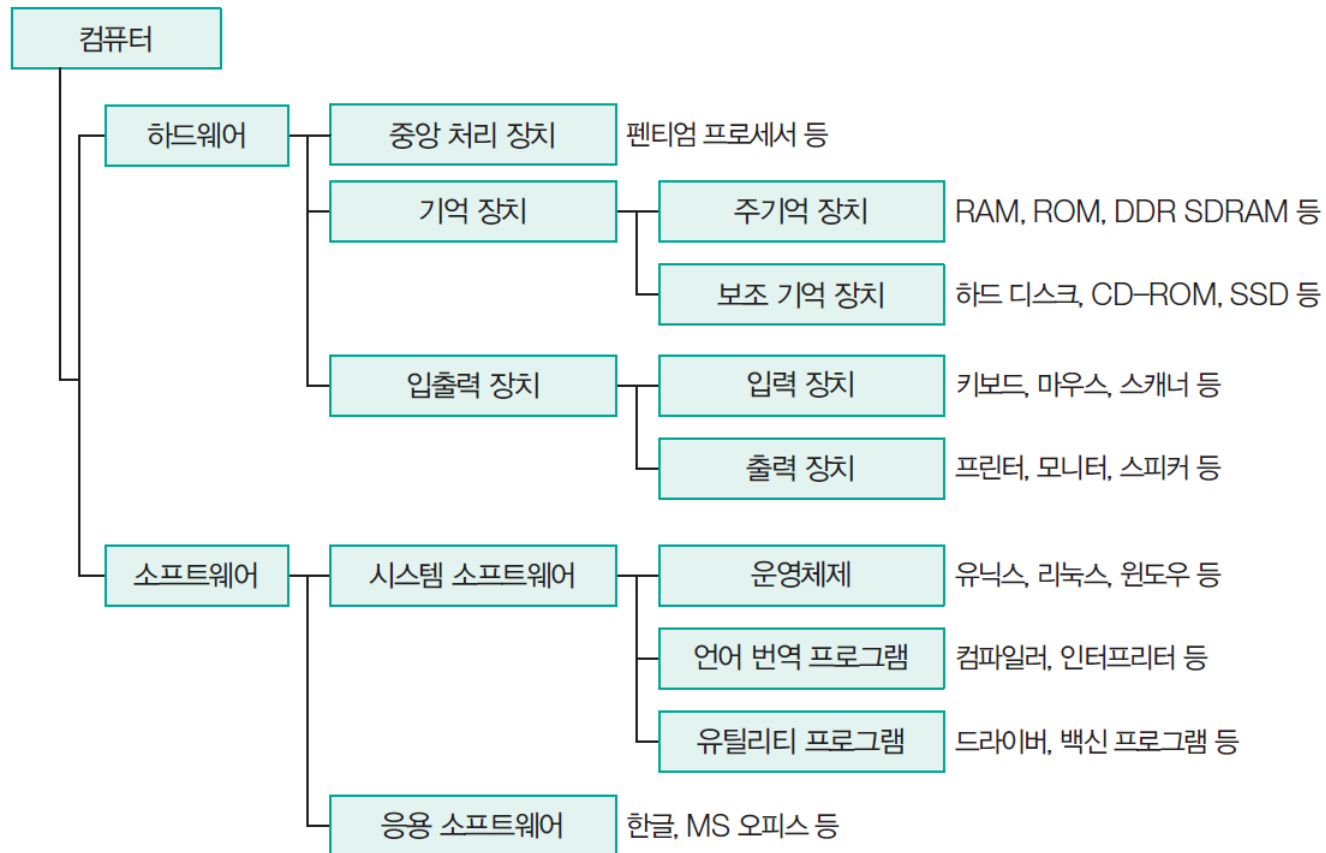


그림 1-1 컴퓨터의 기본 구성

# 01 컴퓨터 시스템의 구성

## 1 하드웨어

- 컴퓨터 하드웨어는 중앙 처리 장치, 주기억 장치, 보조 기억 장치, 입력 장치, 출력 장치, 시스템 버스로 구성된다.
- 컴퓨터의 기능을 수행하기 위해 각 구성 요소들은 시스템 버스를 통해 상호 연결되어 있다.

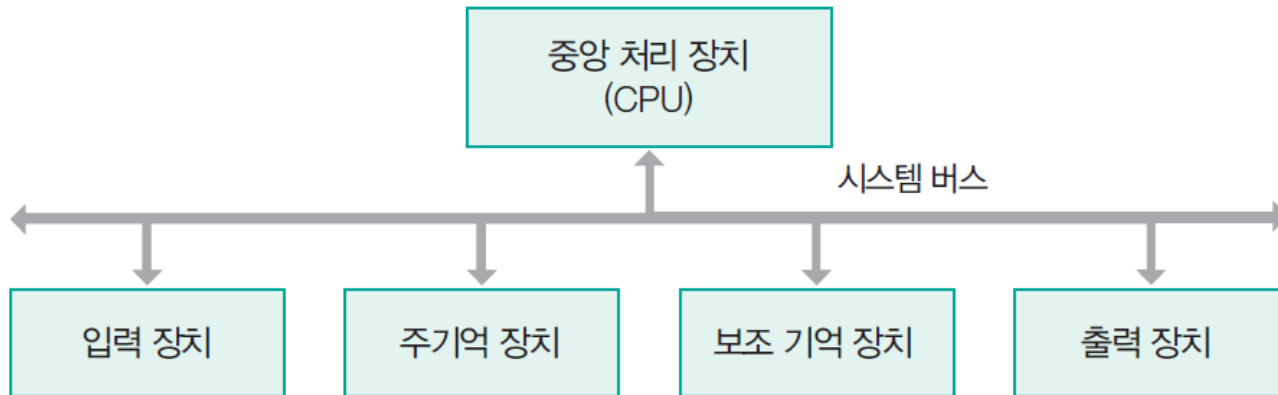


그림 1-2 컴퓨터의 하드웨어 구성

# 01 컴퓨터 시스템의 구성

## □ 중앙 처리 장치

- **중앙 처리 장치**(Central Processing Unit, CPU) : 사실상 컴퓨터의 특성을 결정하며, 컴퓨터의 핵심 기능인 프로그램 실행과 데이터 처리를 담당한다.
- 중앙 처리 장치를 **프로세서**(processor) 또는 **마이크로프로세서**(microprocessor)라고도 부른다.
- **산술 논리 연산 장치**(Arithmetic and Logic Unit, ALU) : 산술 연산 논리 연산, 보수 연산, 시프트 shift 연산을 수행한다.
- **제어 장치**(Control Unit, CU) : 프로그램의 명령어를 해독하여 명령어 실행에 필요한 제어 신호를 발생시키고 컴퓨터의 모든 장치를 제어한다.
- **레지스터**(register) : 중앙 처리 장치 내부에 있는 데이터를 일시적으로 보관하는 임시기억 장치로, 프로그램 실행 중에 사용되며 고속으로 액세스할 수 있다.

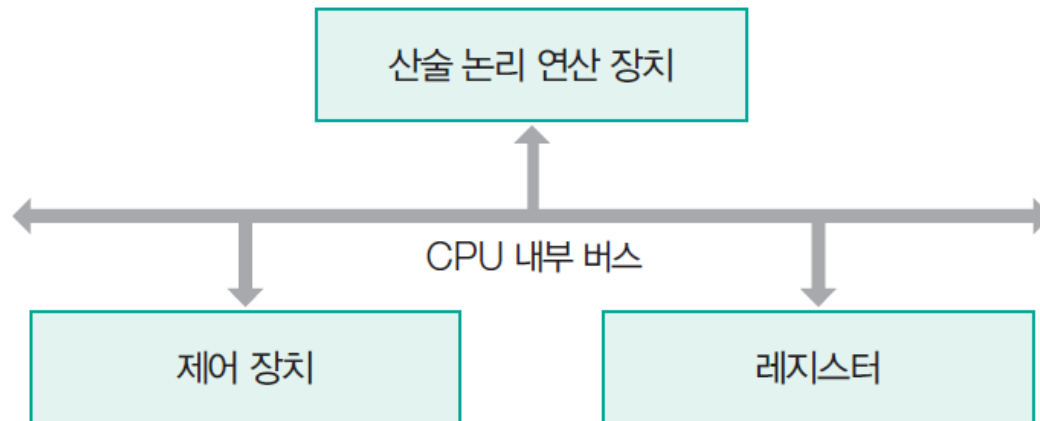


그림 1-3 중앙 처리 장치 내부 구조

# 01 컴퓨터 시스템의 구성

## □ 기억 장치

- **주 기억 장치**(main memory) : 반도체 칩으로 구성되어 고속으로 액세스가 가능하지만 고가다. 또한 프로그램 실행 중에 일시적으로만 사용되는 휘발성 메모리다.
- **보조 기억 장치**(auxiliary storage unit) : 하드 디스크나 SSD, CD-ROM 같은 비휘발성 메모리이며, 저장 밀도가 높고 저가이지만 속도가 느리다. 중앙 처리 장치에 당장 필요하지 않은 많은 양의 데이터나 프로그램을 저장하는 장치다.

## □ 입출력 장치

- **입력 장치**(input device) : 데이터를 전자적인 2진 형태로 변환하여 컴퓨터 내부로 전달한다. 키보드, 마우스가 있다.
- **출력 장치**(output device) : 중앙 처리 장치가 처리한 전자적인 형태의 데이터를 사람이 이해할 수 있는 데이터로 변환하여 출력한다. LCD 모니터, 프린터, 스피커가 있다.
- 입력 장치와 출력 장치를 합하여 **입출력 장치**(Input/Output device, I/O device)라고도 한다.



# 01 컴퓨터 시스템의 구성

## □ 시스템 버스

- **시스템 버스**(system bus) : 중앙 처리 장치와 기억 장치 및 입출력 장치 사이에 정보를 교환하는 통로이며 주소 버스, 데이터 버스, 제어 버스가 있다.

주소 버스 (address bus)	<ul style="list-style-type: none"><li>• 기억 장치나 입출력 장치를 지정하는 주소 정보를 전송하는 신호 선들의 집합이다.</li><li>• 단방향이다.</li></ul>
데이터 버스 (data bus)	<ul style="list-style-type: none"><li>• 기억 장치나 입출력 장치 사이에 데이터를 전송하기 위한 신호선들의 집합이다.</li><li>• 데이터선의 수는 중앙 처리 장치가 한 번에 전송할 수 있는 데이터 비트의 수를 결정한다.</li><li>• 양방향 전송이 가능해야 한다.</li></ul>
제어 버스 (control bus)	<ul style="list-style-type: none"><li>• 중앙 처리 장치가 시스템 내의 각종 요소의 동작을 제어하는 데 필요한 신호선들의 집합이다.</li><li>• 기억 장치 읽기와 쓰기 신호, 입출력 장치 읽기와 쓰기 신호 등이 있다.</li><li>• 단방향이다.</li></ul>

# 01 컴퓨터 시스템의 구성

## □ 컴퓨터 시스템의 구성

- 주기억 장치는 중앙 처리 장치와 시스템 버스를 통해 연결된다.
- 보조 기억 장치나 입출력 장치는 속도가 느리기 때문에 인터페이스 회로나 제어기를 통해 중앙 처리 장치와 연결되어야 한다.

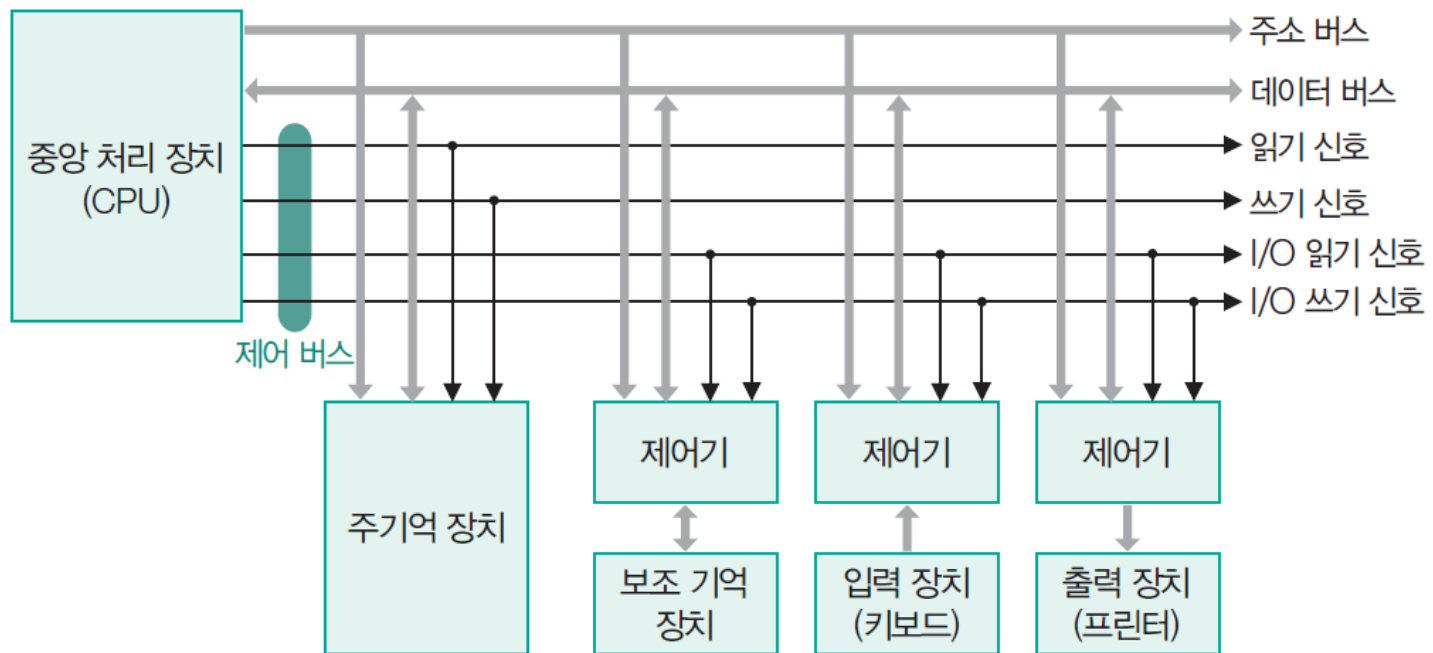


그림 1-4 중앙 처리 장치와 기억 장치 및 입출력 장치의 연결

# 01 컴퓨터 시스템의 구성

## ❖ 컴퓨터 시스템의 수행 기능

- **프로그램 실행** : 중앙 처리 장치가 주기억 장치에서 프로그램 코드를 읽어 실행
- **데이터 저장** : 프로그램 실행 결과를 주기억 장치에 저장
- **데이터 이동** : 하드 디스크 같은 보조 기억 장치에 저장되어 있는 명령어와 데이터 블록을 주기억 장치로 이동
- **데이터 입출력** : 사용자가 키보드나 마우스를 통해 입력하는 명령어나 데이터를 입력하거나 중앙 처리 장치가 처리한 결과를 모니터나 프린터로 출력
- **제어** : 프로그램에서 정해진 순서에 따라 실행되도록 각종 제어 신호를 발생

# 01 컴퓨터 시스템의 구성

## 2 소프트웨어

- 소프트웨어 : 컴퓨터를 구성하고 있는 하드웨어를 잘 동작시킬 수 있도록 제어하고, 지시하는 모든 종류의 프로그램을 의미한다.
- 프로그램 : 컴퓨터를 사용해 어떤 일을 처리하기 위해 순차적으로 구성된 명령들의 집합이다.
- 소프트웨어는 시스템 소프트웨어와 응용 소프트웨어로 나눌 수 있다.

### □ 시스템 소프트웨어

- 시스템 소프트웨어(system software) : 하드웨어를 관리하고 응용 소프트웨어를 실행하는 데 필요한 프로그램이다.
- 운영체제(OS), 언어 번역 프로그램, 유틸리티 프로그램 등이 이에 속한다.
- 유틸리티 프로그램은 각종 주변 장치들을 구동하는 데 필요한 드라이버 프로그램, 백신 프로그램, TCP/IP 같이 컴퓨터를 네트워크로 연결하는 데 필요한 각종 프로그램 등을 말한다.

# 01 컴퓨터 시스템의 구성

## □ 시스템 소프트웨어(계속)

운영체제 (Operating System)	<ul style="list-style-type: none"><li>• 컴퓨터 하드웨어 자원인 중앙 처리 장치, 기억 장치, 입출력 장치, 네트워크 장치 등을 제어하고 관리하는 역할을 한다.</li><li>• 종류 : 유닉스 UNIX, 리눅스 LINUX, 윈도우 Windows, 맥 OS MAC OS, OS, 안드로이드 등</li></ul>
언어 번역 프로그램	<ul style="list-style-type: none"><li>• 고급 언어 프로그램을 컴퓨터가 이해할 수 있는 기계어로 변환하는 프로그램</li><li>• <b>인터프리터</b>(interpreter)는 소스 프로그램을 한 줄씩 기계어로 번역하여 실행하기 때문에 실행 속도가 컴파일러보다 느리다. Basic이나 자바스크립트, HTML, SQL, Python 등</li><li>• <b>컴파일러</b>(compiler)는 전체 소스 프로그램을 한 번에 기계어로 직접 번역하여 실행하기 때문에 실행 속도가 빠르며, C, C++, C#, 자바 등</li></ul>
장치 드라이버 (Device Driver)	<ul style="list-style-type: none"><li>• 컴퓨터에 온라인으로 연결된 주변 장치를 제어하는 운영체제 모듈을 말한다.</li></ul>
링커 (Linker)	<ul style="list-style-type: none"><li>• 여러 개로 분할해 작성된 프로그램에 의해 생성된 목적 프로그램 또는 라이브러리 루틴을 결합하여 실행 가능한 하나의 프로그램으로 연결하는 프로그램</li><li>• 연결 편집기(Linkage Editor)라고도 한다.</li></ul>
로더 (Loader)	<ul style="list-style-type: none"><li>• 하드 디스크 같은 저장 장치에 보관된 프로그램을 읽어 주기억 장치에 적재한 후 실행 가능한 상태로 만드는 프로그램이다.</li><li>• 로더는 할당, 연결, 재배치, 적재 기능을 수행한다.</li></ul>

# 01 컴퓨터 시스템의 구성

## □ 응용 소프트웨어

- 컴퓨터 시스템을 일반 사용자들이 특정한 용도에 활용하기 위해 만든 프로그램으로, 애플리케이션 application, 앱, 어플이라고도 한다.

표 1-1 응용 소프트웨어의 용도와 예

용도	예
사무용	한글, MS-office 제품군
그래픽용	포토샵, 페인트샵, 일러스트레이터
멀티미디어용	WinAMP, GOM Player, PowerDVD
게임용	스타크래프트, 워크래프트, 퀘이크
통신 및 네트워크용	인터넷 익스플로러, 크롬, 모질라, MSN 메신저, 카카오톡

## 3 프로그램 처리 과정

- 프로그램은 고급 언어 → 어셈블리어 → 기계어 순으로 변환



그림 1-5 고급 언어 프로그램의 변환 과정과 예

# 01 컴퓨터 시스템의 구성

## ❖ 기계어 구조 예

### ① LOAD A, X

- 메모리 10번지의 내용을 읽어 레지스터 A에 로드(LOAD)하라는 의미

### ② ADD A, Y

- 레지스터 A의 내용과 메모리 11번지의 내용을 더해 그 결과를 레지스터 A에 로드하라는 의미

### ③ STOR Z, A

- 레지스터 A의 내용을 메모리 12번지에 저장하라는 의미

	연산 코드	오퍼랜드
LOAD A, X	001	01010
ADD A, Y	011	01011
STOR Z, A	010	01100
	3비트	5비트

그림 1-6 기계어 구조 예

## ❖ 연산 코드(opcode)

- CPU가 수행할 연산을 지정하는 비트들
- 비트 수 = 3이면, 지정할 수 있는 연산의 최대 수 :  $2^3 = 8$ 개

## ❖ 오퍼랜드(operand)

- 연산에 사용될 데이터가 저장되어 있는 기억 장치 주소
- 비트 수 = 5 이면, 주소지정할 수 있는 기억 장소의 최대 수 :  $2^5 = 32$  개



# 01 컴퓨터 시스템의 구성

## ❖ 프로그램과 데이터가 주기억 장치에 저장되어 있는 형태

- 명령어와 데이터는 지정된 기억 장소에 저장
- 워드(word) 단위로 저장. 워드는 CPU에 의해 한 번에 처리될 수 있는 비트들의 그룹(8, 16, 32, 66비트)

주소	
0	00101010
1	01101011
2	01001100
3	...
4	...
5	...
6	...
7	...
8	...
9	...
10	10010110
11	00011110
12	...
13	...

명령어

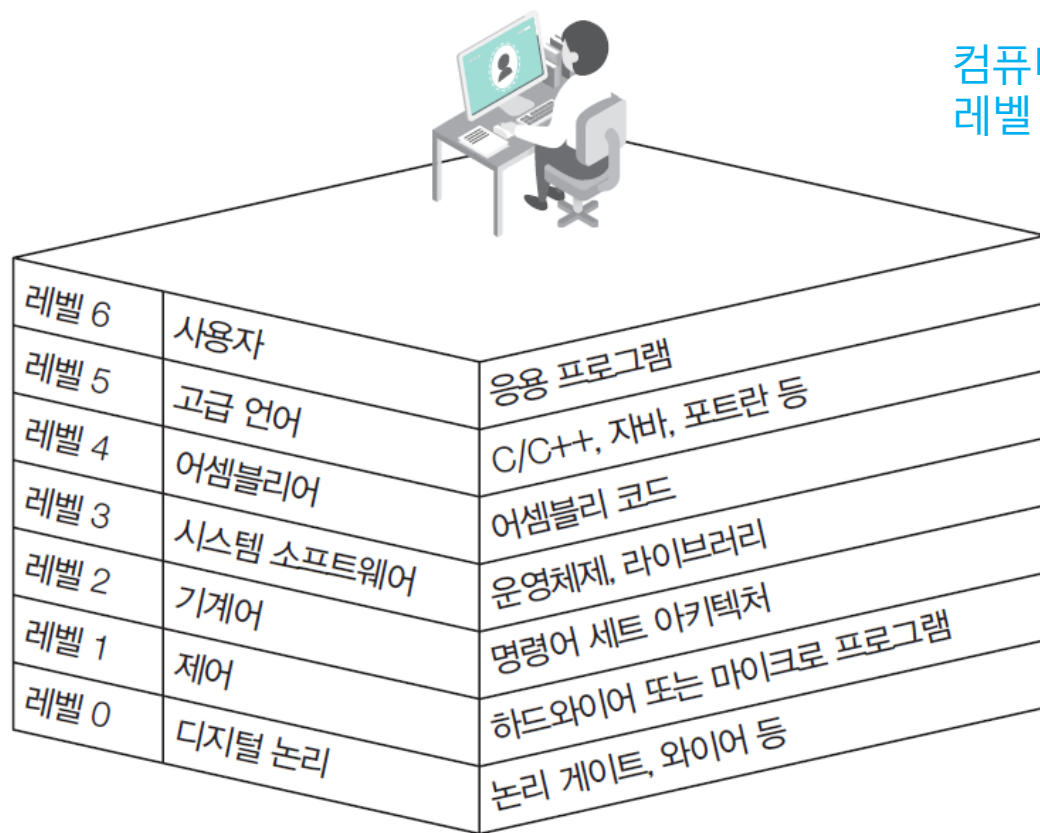
데이터

그림 1-7 프로그램과 데이터가 주기억 장치에 저장되어 있는 예

# 01 컴퓨터 시스템의 구성

## 4 컴퓨터 시스템의 계층 레벨

- **레벨 6**: 사용자 레벨이며 응용 프로그램들로 구성된다. 이 레벨에서 사용자들은 워드 프로세서, 그래픽 패키지, 게임 등과 같은 프로그램을 동작시킨다.
- **레벨 5**: C/C++, JAVA, FORTRAN 같은 고급 언어로 구성되어 있다. 고급 언어 → 어셈블리어 → 기계어 순으로 변환된다.



컴퓨터 구조에서는  
레벨 0부터 레벨 2까지 다룬다.

그림 1-8 컴퓨터 시스템의 추상화 레벨

# 01 컴퓨터 시스템의 구성

- **레벨 4**: 어셈블리어를 포함한다. 어셈블리어 명령어 1개가 정확히 기계어 명령어 1개로 변환된다.
- **레벨 3**: 시스템 소프트웨어 레벨로 운영체제와 라이브러리들로 구성된다. 이 레벨은 다중 프로그래밍, 메모리 할당, 프로세스 관리와 기타 중요 기능을 담당한다.
- **레벨 2**: 명령어 세트 아키텍처(Instruction Set Architecture, ISA) 레벨인 컴퓨터 시스템의 특정 구조에 의해 인식되는 기계어로 구성되어 있다.
- **레벨 1**은 제어 레벨로 제어 장치가 명령어들을 해독(decode)하고 실행하며, 데이터들을 이동할 시간과 장소들을 결정한다. 제어 장치는 레벨 2에서 기계어 명령어를 한번에 하나씩 읽어서 필요한 동작을 수행하기 위한 신호들을 발생시킨다. 제어 장치는 하드와이어 hardwired 방식과 마이크로 프로그램 microprogrammed 방식 중 하나로 구현할 수 있다(5장에서 설명).

## 02 컴퓨터의 역사

### 1 컴퓨터의 발전 과정

#### □ 초기의 계산 도구

- 계산을 하는 도구로서 가장 간단한 것은 주판이며, 기원전 약 3000년 전 고대 메소포타미아 인들이 가장 먼저 사용했다고 추정된다.
- 주판을 제외하면 17세기에 이르도록 계산을 위한 특별한 도구가 없었다.

#### □ 기계식 계산기

- **톱니바퀴를 이용한 수동 계산기** : 기어로 연결된 바퀴판들로 덧셈과 뺄셈 수행(1642년 파스칼)
- 1671년 독일의 라이프니츠는 이를 개량하여 곱셈과 나눗셈도 가능한 계산기를 발명하였다. 또 라이프니츠는 기계 장치에 더 적합한 진법을 연구해서 17세기 후반에 2진법을 창안했다.
- **차분 기관**(difference engine) : 1823년 영국의 배비지가 삼각함수를 유효숫자 5자리까지 계산하여 종이에 표로 인쇄하였다.

## 02 컴퓨터의 역사

### □ 기계식 계산기(계속)

- **해석 기관**(analytical engine) : 방정식을 순차적으로 풀 수 있도록 고안된 기계식 계산기(1833년, 베비지). 오늘날 사용하는 컴퓨터의 기본 요소를 모두 갖춘 것이었다.

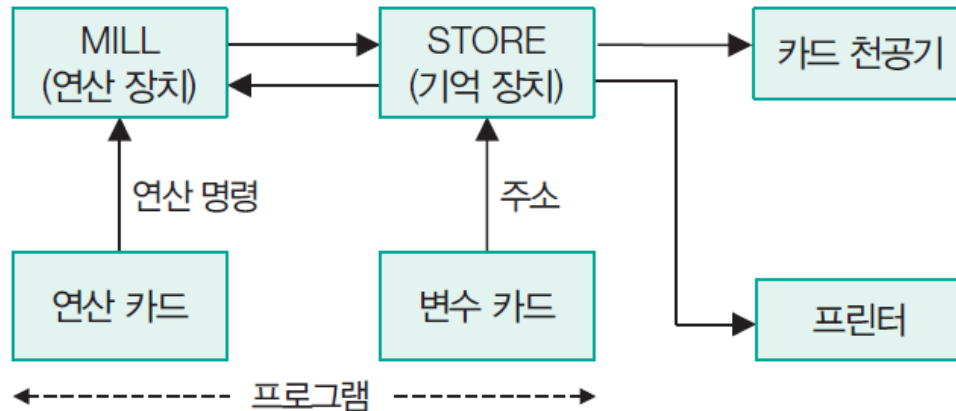


그림 1-9 해석 기관의 기본 구조

- **천공 카드 시스템**(Punch Card System, PCS) : 1889년 미국의 홀러리스는 인구조사에 활용하기 위한 시스템을 만들었다. 종이 카드에 구멍을 뚫어 자료를 처리하는 시스템으로 수 백 장의 카드를 읽고 기억하는 데 1분도 걸리지 않았다. 그 결과 1890년에 시행된 인구조사에서는 7년 이상이 걸리던 통계 처리 시간을 3년 이하로 단축시켰다.

## 02 컴퓨터의 역사

### □ 전기 기계식 계산기

- **MARK-I** : 1944년 하버드 대학의 에이킨이 개발. MARK-I은 배비지의 해석 기관을 실현시킨 것으로서 미해군의 탄도 계산에 이용되었으며, 수많은 수학이나 과학 문제를 해결하는 데 공헌했다.

### □ 전자식 계산기

- **에니악**(ENIAC, Electronic Numerical Integrator And Computer) : 진공관을 사용한 최초의 전자식 컴퓨터(1946년, 에커트와 모클리)
- **에드삭**(EDSAC, Electronic Delay Storage Automatic Calculator) : 10진수 체계와 프로그램 내장 방식의 계산기(1949년, 월키스)
- **에드박**(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, 모클리와 에커트)

컴퓨터의 발전 과정에 있어서 진공관 및 프로그램 내장 방식의 사용은 근대에서 현대로 넘어 오게 되는 분기점이 되었다.

## 2 컴퓨터의 세대별 분류

- 컴퓨터의 발전 과정을 세대별로 명확하게 설명하기는 어렵지만 새로운 하드웨어 부품의 출현을 기준으로 분류되고 있다.

표 1-2 컴퓨터 세대 구분

구분 \ 세대	1세대	2세대	3세대	4세대	5세대
주요 소자	진공관	트랜지스터	SSI, MSI	LSI, VLSI	VLSI, ULSI
주 기억 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( $10^{-3}$ )	$\mu$ s( $10^{-6}$ )	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)



릴레이



진공관



트랜지스터



IC

## 02 컴퓨터의 역사

### ❖ 집적 회로의 분류

- 집적 회로(Integrated Circuit, IC)가 개발된 3세대 이후에는 컴퓨터 세대에 대한 정의가 분명하지 않다.

SSI (Small Scale IC)	<ul style="list-style-type: none"><li>• 트랜지스터 수십 개가 집적된 소규모 IC로, 기본 게이트 기능과 플립 플롭이 이에 해당한다.</li></ul>
MSI (Medium Scale IC)	<ul style="list-style-type: none"><li>• 트랜지스터 수백 개가 집적된 중규모 IC로, 디코더, 인코더, 멀티플렉서, 디멀티플렉서, 카운터, 레지스터, 소형 기억 장치 등의 기능을 포함한다.</li></ul>
LSI (Large Scale IC)	<ul style="list-style-type: none"><li>• 트랜지스터 수천 개가 집적된 대규모 IC로, 8비트 마이크로프로세서나 소규모 반도체 기억 장치 칩이 이에 해당한다.</li></ul>
VLSI (Very Large Scale IC)	<ul style="list-style-type: none"><li>• 트랜지스터 수만에서 수십만 개 이상 집적된 초대규모 IC로, 대용량 반도체 메모리, 1만 게이트 이상의 논리 회로, 단일 칩 마이크로프로세서 등이 있다.</li></ul>
ULSI (Ultra Large Scale IC)	<ul style="list-style-type: none"><li>• 트랜지스터가 수백만 개 이상 집적된 극대규모 IC로, 인텔의 486이나 팬티엄, 수백 메가바이트 이상의 반도체 기억 장치 칩 등이 이에 해당한다.</li><li>• VLSI와 ULSI 사이의 정확한 구분은 사실 모호하다.</li></ul>



## 02 컴퓨터의 역사

### □ 1세대 컴퓨터

- 진공관을 사용함에 따라 컴퓨터 크기가 매우 크며, 열 발생량이 많고 전력 소모가 크다.
- 폰 노이만이 제안한 프로그램 내장의 개념을 도입했다.
- 수치 계산, 통계 등에 사용되었다.
- 컴퓨터 언어는 기계어와 어셈블리어를 사용했다.
- 대표적인 컴퓨터 : ENIAC, UNIVAC, EDSAC, EDVAC

### □ 2세대 컴퓨터

- 자기 드럼이나 자기 디스크 같은 대용량의 보조 기억 장치가 사용되었다.
- 운영체제의 개념을 도입했다.
- 다중 프로그래밍 기법을 사용했다.
- 온라인 실시간 처리 방식을 도입했다.
- 과학 계산, 일반 사무용으로 사용되었다.
- 트랜지스터를 사용함으로써 컴퓨터는 더 고속화되고, 기억 용량이 늘었으나 소형화되었다.
- 소프트웨어 개발에 주력한 시기로, 사용된 언어로는 FORTRAN, ALGOL, COBOL 등이 있다.

## 02 컴퓨터의 역사

### □ 3세대 컴퓨터

- 캐시(cache) 기억 장치가 등장했다.
- OMR, OCR, MICR이 도입되었다.
- 패밀리(family) 개념의 출현에 따라 프로그램의 호환성이 이루어졌다.
- 시분할 처리를 통해 멀티프로그래밍을 지원했다.
- 경영 정보 시스템(Management Information System, MIS)이 확립되었다.

### □ 4세대 컴퓨터

- 마이크로프로세서가 개발되었다.
- 가상 기억 장치의 개념이 도입되었다.
- 컴퓨터 네트워크가 발전되었다.
- 개인용 컴퓨터PC가 등장하여 대중화를 이루었다.
- 온라인 실시간 처리 시스템이 보편화되었고 기존 시스템에 비해 빠른 처리 속도를 갖게 되었다.

### □ 5세대 컴퓨터

- 5세대는 아직 명확하게 구분되지 않고 있으나 부품의 집적도보다는 획기적인 응용 소프트웨어의 출현에 의해 정의될 가능성이 있다.
- 현 시점에서는 VLSI, ULSI를 기본 소자로 하여 초미니, 초고속을 추구하며 기존 시스템의 수준을 벗어나 경영 정보, 지식 정보, 인공지능, 신경망, 퍼지, 멀티미디어, 가상 현실을 목표로 하고 있다.
- 컴퓨터와 인간의 인터페이스를 좀 더 인간에게 편리하도록 하기 위한 GUI 환경을 구현하고 있다.
- 컴퓨터의 성능을 향상시키기 위해 다중 프로세서를 사용한 병렬 처리 컴퓨터 시스템, 광 컴퓨터, 신경망 컴퓨터 등의 개발과 인공지능의 연구가 활발히 진행되고 있다.
- 비 폰 노이만 Non-Von Neumann 컴퓨터 구조가 제안되었다.
- 고도의 사람 대 기계 man-machine 인터페이스가 개발되었다.
- 객체 지향 프로그래밍 언어가 사용되고, 문자, 음성, 영상 정보가 통합되는 멀티미디어 시대가 도래했다.

### 3 무어의 법칙과 황의 법칙

#### □ 무어의 법칙

- 반도체 집적 회로의 트랜지스터 수가 12개월마다 2배로 증가한다는 법칙이다.
- 1970년대 중반부터 이 기간은 12개월에서 18개월로 늘어났다. 이럴 경우에 10년 동안 증가율은 1,000배가 아니라 100배 정도가 되었다. 1990년대부터는 18개월이 다시 24개월로 늘어났는데, 이 경우에 10년간 증가율은 32배 정도로 줄어들었다.
- 무어의 법칙은 한계에 다다르고 있다. 이것은 기술적인 한계뿐만 아니라 경제적인 한계도 있었다.

#### □ 황의 법칙

- 집적 회로를 뛰어넘는 메모리의 발전으로 인해서 앞으로는 1년에 2배씩 용량이 뛰어오를 것이라고 주장한 것
- 이는 계속해서 NAND flash 계열의 메모리가 지속적으로 발전하면서 실제로 증명되었다. 하지만 2010년, 불과 8년 만에 황의 법칙은 깨졌다.

## 03 컴퓨터의 분류

### □ 컴퓨터의 분류

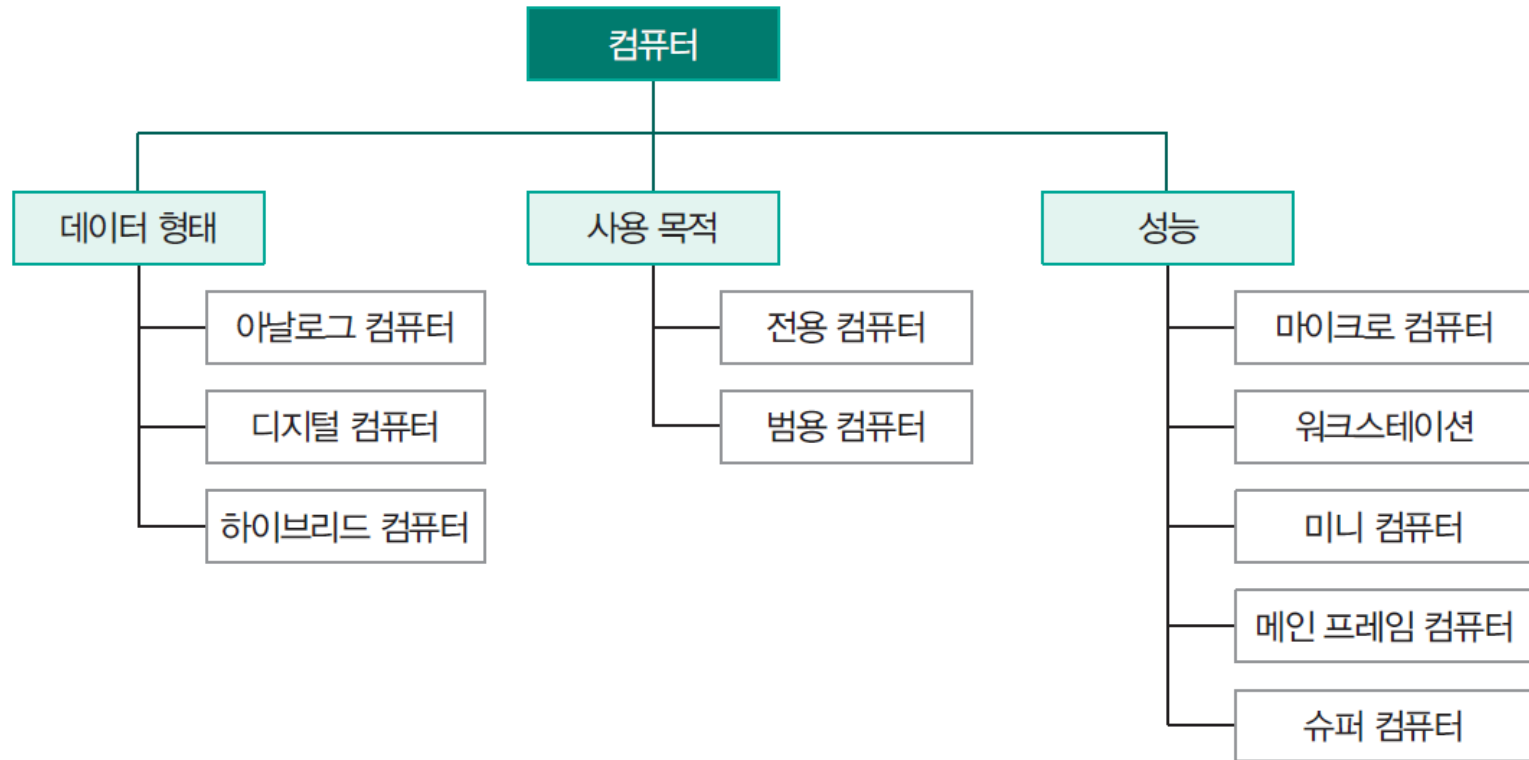


그림 1-10 컴퓨터의 분류

## 03 컴퓨터의 분류

### 1 데이터 형태에 따른 분류

#### ❑ 아날로그 컴퓨터(analog computer)

- 연속적인 변량을 사용하여 계산을 수행한다.
- 신속한 입력과 그 상태에 대한 즉각적인 반응을 얻을 수 있으므로 이 유형의 컴퓨터는 프로세스 제어 process control에 적합하다.
- 아날로그 컴퓨터에서는 전압, 전류, 온도, 압력 등의 데이터를 처리한다.

#### ❑ 디지털 컴퓨터(digital computer)

- 숫자나 문자를 코드화하여 필요한 정밀도까지의 결과를 얻을 수 있다.
- 디지털 컴퓨터는 숫자나 문자를 코드화하여 사용하며, 데이터를 분석하고 종합하여 처리한 결과를 숫자나 문자 등으로 정확히 구분할 수 있게 해준다.

#### ❑ 하이브리드 컴퓨터(hybrid computer)

- 아날로그나 디지털 신호를 모두 처리할 수 있으며 필요에 따라 아날로그 신호나 디지털 신호로 변환한다.
- 컴퓨터에서 처리한 결과는 필요에 따라 A/D 변환기나 D/A 변환기에 의해서 데이터를 아날로그 형태나 디지털 형태로 얻을 수 있다.

### 2 사용 목적에 따른 분류

#### ❑ 전용 컴퓨터(special purpose computer)

- 특수한 목적으로 사용하기 위한 컴퓨터로 주로 군사용, 기상 예보용, 천문학, 원자핵 물리 분야 등의 특정 업무에 사용된다.
- 전용 컴퓨터는 주로 고정 프로그램과 일정한 데이터만을 취급할 수 있도록 구성되어 있다.

#### ❑ 범용 컴퓨터(general purpose computer)

- 여러 업무에 광범위하게 사용할 수 있는 일반 목적용 컴퓨터다.
- 과학 계산, 통계 데이터 처리, 생산 관리, 사무 관리 등 광범위한 분야에 사용할 수 있다.
- 범용 컴퓨터는 여러 형태의 데이터를 취급할 수 있는 유연성, 기억 용량의 증대, 처리 속도의 신속화, 입출력 장치의 다양화 등 언제든지 필요한 시기에 확장 또는 조정할 수 있는 기능이 필요하다.

## 03 컴퓨터의 분류

### 3 성능과 규모에 따른 분류

- 정량보다는 정성적 측면에서 분류한 것이다.
- 근래에는 분류 기준을 기억 용량, 처리 능력, 가격 측면에 두고 있다.
- 이와 같은 분류는 컴퓨터 관련 기술의 급속한 발전으로 인해 구분이 모호해지고 있다.

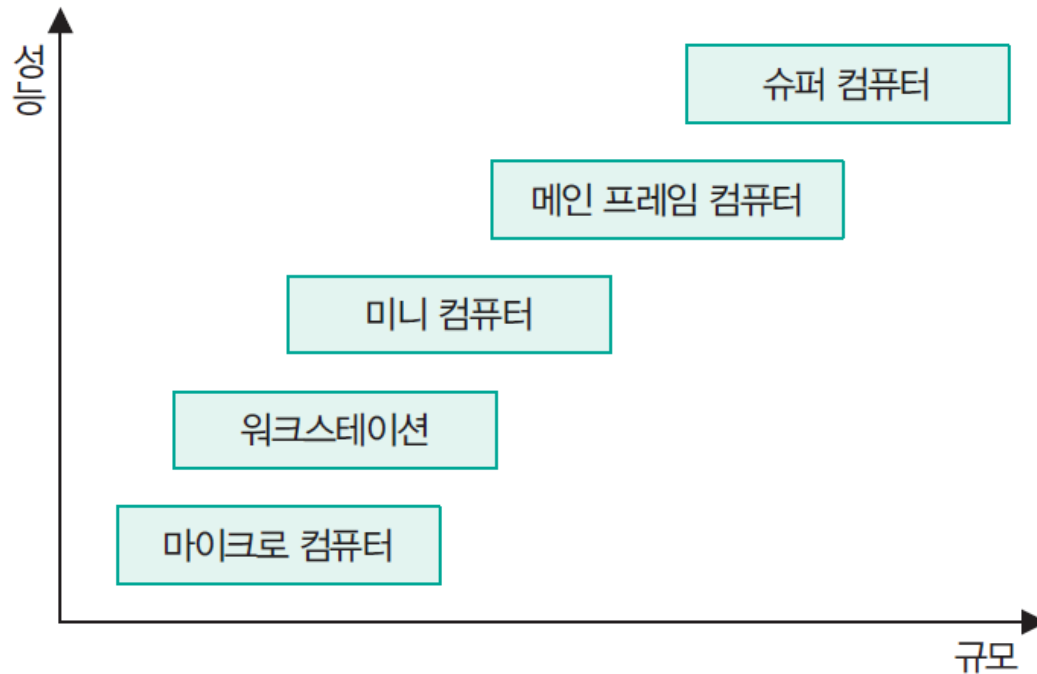


그림 1-11 처리 성능과 규모에 따른 컴퓨터 분류



## 03 컴퓨터의 분류

### □ 마이크로 컴퓨터(microcomputer)

- 최근 가장 널리 사용되고 있는 범용 컴퓨터다.
- 마이크로프로세서를 중앙 처리 장치로 사용하는 컴퓨터를 의미하며, 워크스테이션과 개인용 컴퓨터(PC)가 있다.
- 1970년대 후반에는 개인용 컴퓨터가 등장했다. 1980년대에 들어서면서 마이크로프로세서의 성능이 급속도로 향상되면서 개인용 컴퓨터의 응용 범위도 단순 응용에서부터 매우 복잡한 응용에까지 확산되고 있었다.
- 최근에는 개인용 컴퓨터의 발전으로 워크스테이션과의 구별이 모호해지고 있다.

### □ 워크스테이션(workstation)

- 과학자, 공학자, 엔지니어, 애니메이터와 같은 전문직 종사자들이 개인용 컴퓨터보다 더 우수한 성능을 요구하여 만들어진 컴퓨터다.
- 외형은 개인용 컴퓨터와 유사한데, 최근에는 워크스테이션의 가격이 떨어지고 개인용 컴퓨터의 성능이 강력해지기 때문에 워크스테이션과 개인용 컴퓨터의 구분이 사라지고 있다.

## 03 컴퓨터의 분류

### □ 미니 컴퓨터(minicomputer)

- 성능과 규모 측면에서 마이크로 컴퓨터와 메인 프레임 컴퓨터의 중간에 해당한다. 마이크로 컴퓨터보다 대용량 기억 장치와 고속의 주변 장치들로 구성되어 있어서 다수의 사용자가 동시에 한 미니 컴퓨터를 사용할 수 있었다.
- 최근에는 컴퓨팅 서버, 파일 서버, 네트워크 서버 등과 같은 각종 서버의 등장으로 인해 그 자취를 감추게 되었다.

### □ 메인 프레임 컴퓨터(main frame computer)

- 대형 컴퓨터라고도 부르는데, 1초에 수십억 개의 명령어를 처리할 수 있는 고속의 컴퓨터다.
- 대기업, 관공서, 대학 등에서 다수의 사용자가 공유하여 사용한다. 메인 프레임에는 단말기 terminal를 통해 접근할 수 있는데, 단말기는 키보드와 모니터가 통합된 장치를 말한다.
- 응용 분야는 은행, 보험, 병원 업무 등에 널리 사용되고 있다.

### □ 슈퍼 컴퓨터(super computer)

- 고속의 연산 처리를 위한 중앙 처리 장치, 대규모의 용량을 가진 주기억 장치, 강력한 병렬 처리를 지원하는 소프트웨어로 이루어진 컴퓨터다.
- 고성능 마이크로프로세서를 수십만 개까지 사용하여 작업을 병렬로 처리하기 때문에 초당 수조 개의 명령어들을 처리할 수 있도록 매우 빠르다.
- 슈퍼 컴퓨터는 기상예측, 석유탐사, 원자력 개발 등의 분야에서 사용되고 있다.

## 04 폰 노이만, 비 폰 노이만, 하버드 구조

### 1 폰 노이만 구조와 비 폰 노이만 구조

- 사실 폰 노이만 구조 개념을 처음으로 생각한 사람은 모클리와 에커트였다.
- 폰 노이만 구조의 프로그램 처리 과정
  - ① 프로그램 카운터를 이용해 메모리에서 실행할 명령어를 인출한다.
  - ② 제어 장치는 이 명령어를 해독한다.
  - ③ 명령을 실행하는 데 필요한 데이터를 메모리에서 인출하여 레지스터에 저장한다.
  - ④ 산술 논리 연산 장치는 명령을 실행하고, 레지스터나 메모리에 결과를 저장한다.

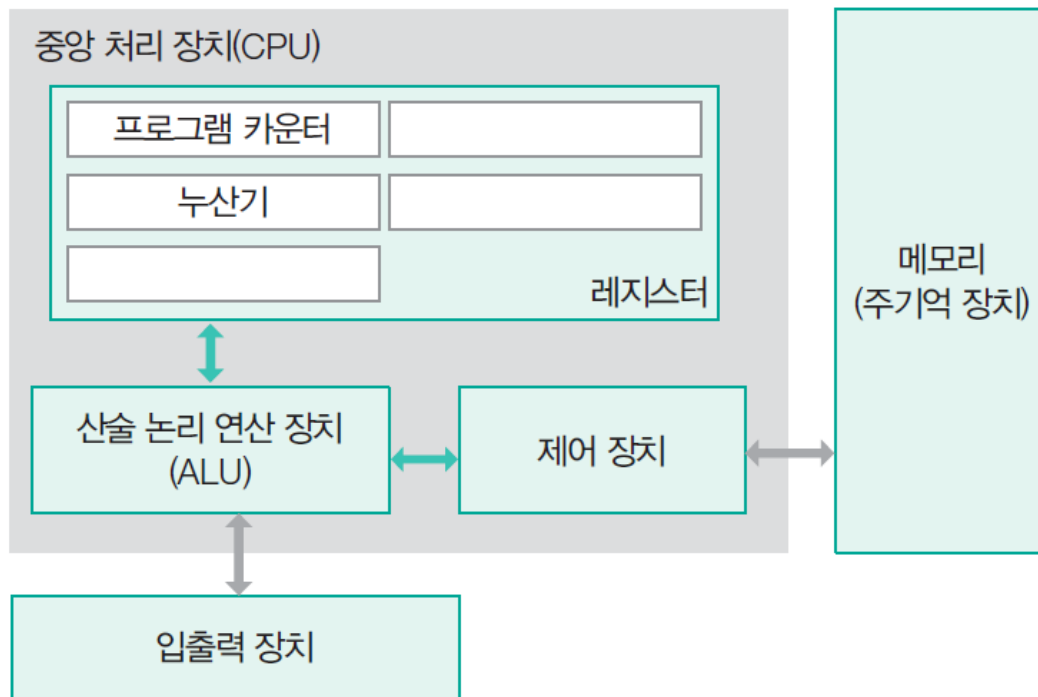


그림 1-12 폰 노이만 구조

## 04 폰 노이만, 비 폰 노이만, 하버드 구조

### ❖ 폰 노이만 병목 현상(Von-Neumann Bottleneck)

- 폰 노이만 구조의 컴퓨터는 중앙 처리 장치에서 명령어나 데이터를 메모리에서 가져와 처리한 후, 결과 데이터를 메모리에 다시 보내 저장한다. 또한 저장된 데이터가 필요할 땐 다시 메모리에서 중앙 처리 장치로 불러오는 방식으로 순차적으로 프로그램을 처리하므로 메모리나 시스템 버스에 병목 현상이 생겨 속도가 느려질 수밖에 없다.
- 이것을 데이터 경로의 병목 현상 또는 기억 장소의 지연 현상이라고도 한다.

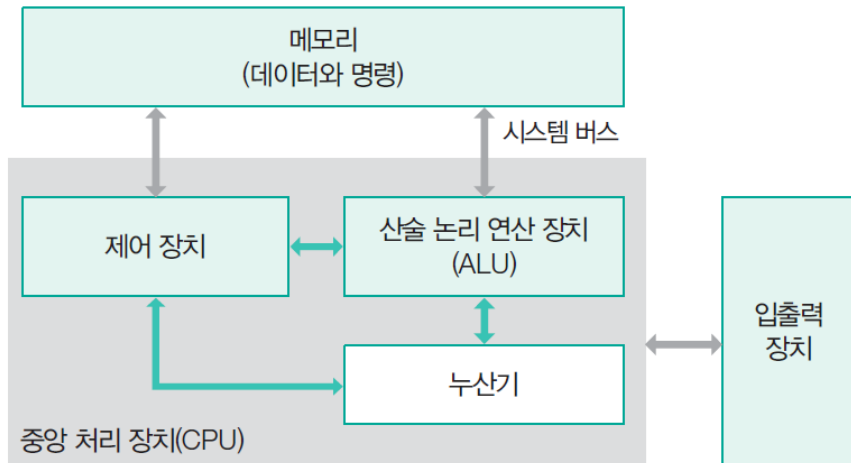
### ❖ 비 폰 노이만 구조

- 폰 노이만 구조가 아닌 컴퓨터를 통틀어 이르는 말로 데이터 처리의 고속화와 고도화를 위하여 프로그램 일부를 하드웨어화하거나, 병렬 처리 기능, 추론 기구를 채택한 컴퓨터를 가리킨다.
- 연구되고 있는 분야로는 신경망, 유전 알고리즘, 양자 컴퓨터, 병렬 컴퓨터를 비롯한 여러 분야들이 비 폰 노이만 범주에 속한다. 이 중에서 병렬 컴퓨팅이 현재 가장 많이 사용된다.

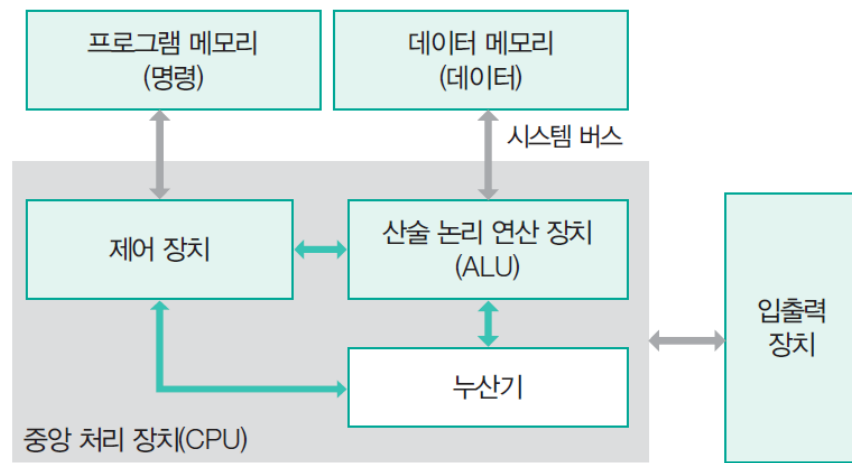
# 04 폰 노이만, 비 폰 노이만, 하버드 구조

## 2 폰 노이만 구조와 하버드 구조

- 폰 노이만 구조는 구조적으로 시스템 버스에 병목 현상이 발생한다.
- 하버드 구조는 폰 노이만 구조의 단점을 보완한 개념이다.
- 명령어 메모리 영역과 데이터 메모리 영역을 물리적으로 분리시키고, 각각을 다른 시스템 버스로 중앙 처리 장치에 연결함으로써 명령과 데이터를 메모리로부터 읽는 것을 동시에 처리할 수 있다.
- 그러나 하버드 구조는 비싸고, 공간도 많이 차지하며, 설계가 복잡하다.
- 최근 CPU 설계에서는 폰 노이만 구조와 하버드 구조 양자의 구조를 도입하고 있다.



(a) 폰 노이만 구조



(b) 하버드 구조

그림 1-13 폰 노이만 구조와 하버드 구조