

01

컴퓨터 시스템의 구성

01. 컴퓨터 시스템의 구성

I. 하드웨어의 구성

- 하드웨어는 중앙처리장치(CPU), 기억장치, 입출력장치로 구성되어 있음.
- 각 장치는 시스템 버스로 연결됨.

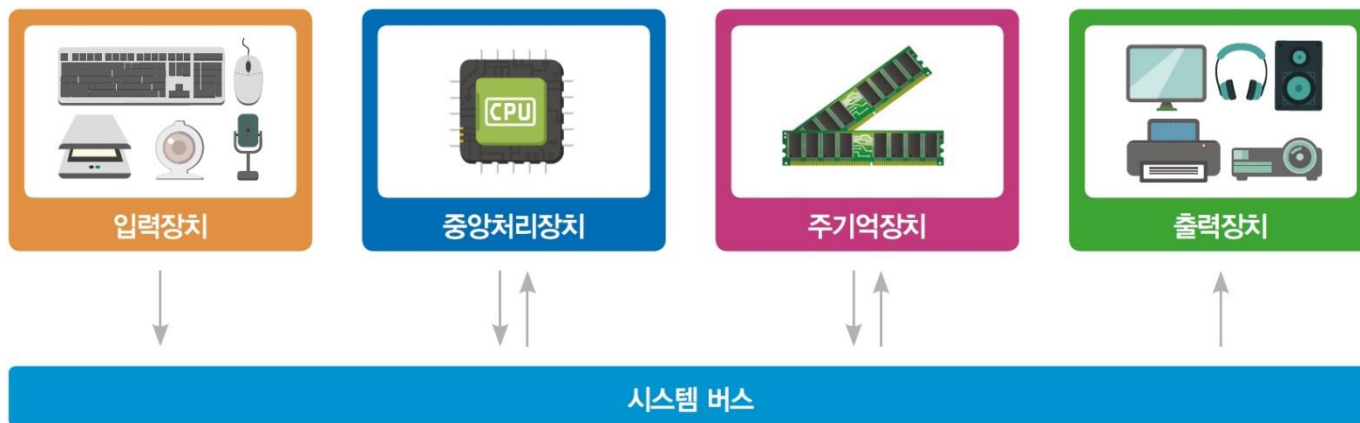


그림 3-1 하드웨어의 구성

01. 컴퓨터 시스템의 구성

I. 하드웨어의 구성

■ 중앙처리장치

- **중앙처리장치** : 컴퓨터의 머리에 해당하는데, 프로그램의 명령어와 데이터를 읽어 와서 명령어를 실행하고 입력된 데이터를 처리함.
- **중앙처리장치의 구성**
 - 산술논리연산장치 : 데이터의 계산 및 비교를 담당
 - 제어장치 : 명령어를 해석하고 시스템 실행을 담당
 - 레지스터 : 명령어를 실행하는 동안 필요한 정보를 저장

01. 컴퓨터 시스템의 구성

I. 하드웨어의 구성

■ 기억장치

- 기억장치 : 컴퓨터에 필요한 정보를 저장하는 장치
- 기억장치의 구분
 - 주기억장치 : 메인메모리(main memory)라고도 하며, 실행 중인 프로그램과 프로그램 실행에 필요한 데이터를 일시적으로 저장하는 장치임.
 - 보조기억장치 : 주기억장치보다 속도는 느리지만 많은 자료를 영구적으로 보관할 수 있음.

01. 컴퓨터 시스템의 구성

I. 하드웨어의 구성

■ 입출력장치

- **입력장치** : 컴퓨터에 데이터를 전달하는 장치로, 중앙처리장치나 주기억장치에 데이터를 입력함.
- **출력장치** : 작업 결과를 인간이 인지할 수 있도록 나타내는 장치임

01. 컴퓨터 시스템의 구성

I. 하드웨어의 구성

■ 시스템 버스

- 시스템 버스 : 컴퓨터에 있는 중앙처리장치, 주기억장치, 입출력장치 간 데이터를 주고받기 위해 사용하는 경로(간단히 '버스'라고도 부름).
- 시스템 버스의 종류 : 데이터 버스, 주소 버스, 제어 버스로 구분됨.

01. 컴퓨터 시스템의 구성

- 데이터 버스

- 중앙처리장치와 다른 장치 사이 또는 중앙처리장치 내부의 모듈들 사이에서 데이터가 이동하는 경로.

- 주소 버스

- 중앙처리장치가 사용하고자 하는 데이터의 주소가 이동하는 경로.
- 주소 버스의 비트 수, 즉 버스의 대역폭이 주기억장치의 용량을 결정함.
- 만약 어떤 데이터를 주기억 장치에서 중앙처리장치로 읽어 오려면 데이터가 저장된 주소 값을 메모리에 전달해야 함.
- 주소 값은 주소 버스를 통해 전달되고, 주기억장치는 해당 주소에 저장된 데이터를 데이터 버스에 싣는다.
- 주소 버스가 32비트이면 2³²개를 지정할 수 있으면 주기억 장치의 용량은 4GB이다.

01. 컴퓨터 시스템의 구성

I. 하드웨어의 구성

■ 시스템 버스

• 제어 버스

- 중앙처리장치가 기억장치나 입출력장치에 제어 신호를 전달하는 통로.
- 중앙처리장치에서 발생하는 제어 명령이나 상태 신호가 이 버스를 통해 이동함.



그림 3-2 시스템 버스의 구성

01. 컴퓨터 시스템의 구성

I. 하드웨어의 구성

■ 시스템 버스

하나 더 알기 메인보드와 포트

■ 메인보드(Main Board)

- 중앙처리장치와 주기억장치 등 다양한 컴퓨터 부품을 연결시켜 주는 판.
- 메인보드에 복잡하게 연결되어 있는 선들이 '버스'임.



(a) 메인보드 구성 형태



(b) 메인보드 단면 확대(버스)

그림 3-3 메인보드

02

중앙처리장치

02. 중앙처리장치

I. 중앙처리장치의 구성

- **중앙처리장치(CPU)** : 컴퓨터의 두뇌에 해당하는 곳으로, 기계어로 된 명령어를 해독한 후 해당 명령어를 수행하기 위한 세부 작업을 실행하는 장치임.
- 중앙처리장치는 3개의 세부 모듈인 연산장치, 제어장치, 레지스터로 구성되어 있으며, 이들은 각각 내부 버스로 연결되어 있음.

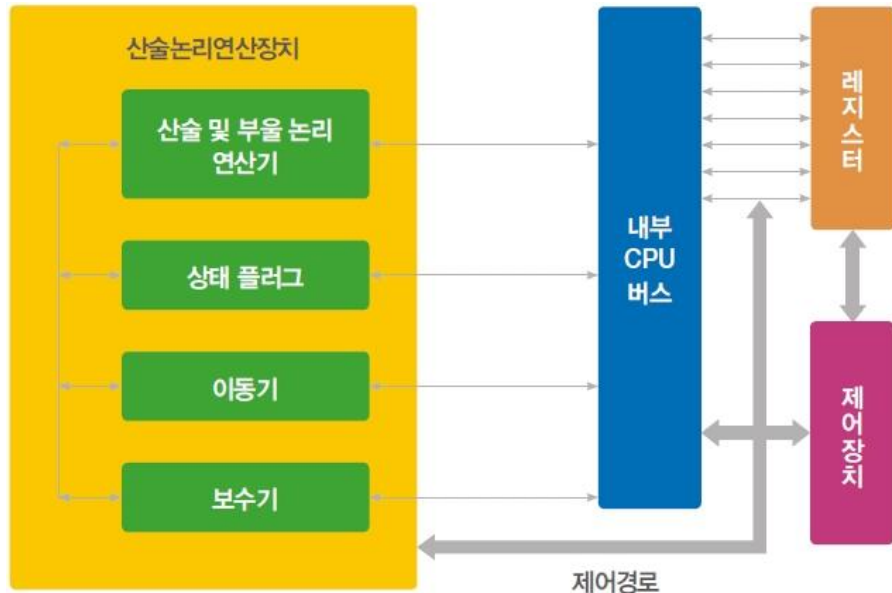


그림 3-8 중앙처리장치의 구성

02. 중앙처리장치

I. 중앙처리장치의 구성

■ 연산장치

- 연산장치 : 중앙처리장치 내부에서 실제 연산을 담당하며, 산술연산과 논리연산을 수행함.
 - 산술연산 : 덧셈, 뺄셈, 곱셈, 나눗셈 등 일상생활에서 자주 사용하는 사칙연산.
 - 논리연산 : 논리곱(AND), 논리합(OR), 부정(NOT) 등을 수행하는 연산.

하나 더 알기

산술연산과 논리연산의 연산 속도

- 컴퓨터에서는 산술연산(덧셈연산)과 논리연산 모두 동일한 결과가 나옴.
- 자리올림이 발생하지 않는 특별한 경우에는 논리연산이 산술연산보다 빠르게 결과를 냄.

02. 중앙처리장치

I. 중앙처리장치의 구성

■ 제어장치

- **제어장치** : 명령어가 순서대로 실행되도록 제어하는 장치로, 수행할 명령어를 해독한 후 명령어 수행에 필요한 제어 신호를 생성함.
- 중앙처리장치 내부에 있는 데이터를 주기억장치에 저장하려면 메모리에 Write 제어 명령을 보내야 하는데, 이 일을 수행하는 것이 제어장치임.

■ 레지스터

- **레지스터** : 중앙처리장치와 속도가 비슷한 고속의 기억장치.
- 프로그램 실행에 사용되는 모든 데이터는 주기억장치에 저장되지만 실행 자체는 중앙처리장치 내부에서 이루어지므로 주기억장치에 있는 데이터 또한 중앙처리장치 내부에 다시 저장해야 함. 그 장소를 레지스터라고 부름.

02. 중앙처리장치

II. 중앙처리장치의 동작

- 중앙처리장치의 명령어 처리 방식

- ① 인출 단계(Fetch) : 주기억장치에 저장된 명령어 하나를 읽어 옴.
- ② 해독 단계(Decode) : 읽어 온 명령어를 제어 정보로 해독.
- ③ 실행 단계(Execute) : 해독된 명령어를 실행.

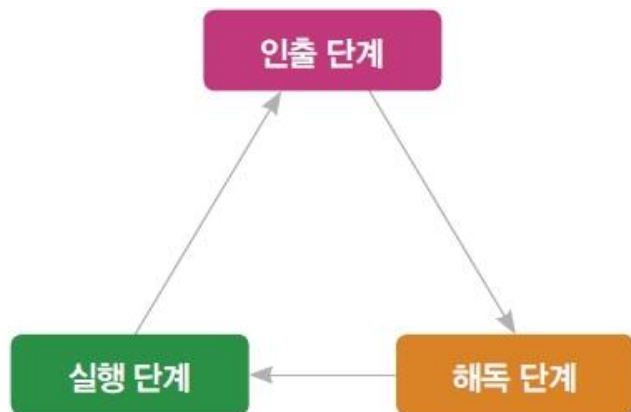


그림 3-9 중앙처리장치의 명령어 처리 방식

02. 중앙처리장치

III. 중앙처리장치 관련 용어

■ 클록 펄스

- **클록(Clock)** : 컴퓨터에서 일정한 박자를 만들어 내는 장치이며, 클록이 일정한 간격으로 만드는 틱(Tick)을 펄스(Pulse)라고 함.
- **클록 펄스(Clock Pulse)** : 컴퓨터 내에서 동작하는 각 구성 요소의 모든 동작을 동기화하기 위해 사용하는 전자적 펄스.
- 하강 지점에서 Q_a , Q_b , Q_c 같은 클록 펄스의 주기가 일정하게 실행됨.

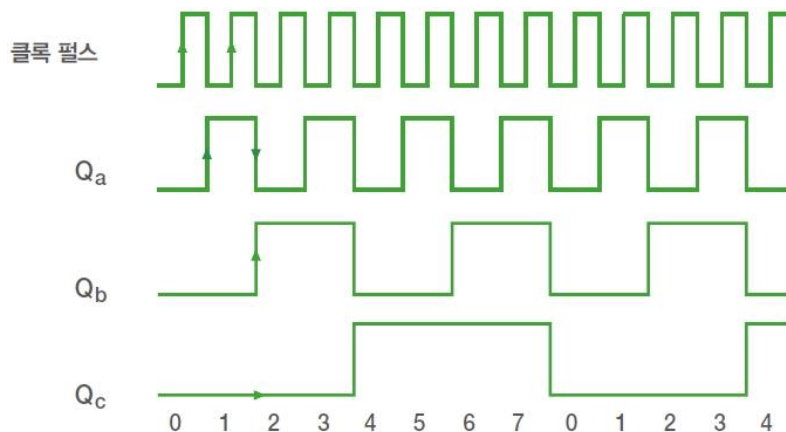


그림 3-10 클록 펄스의 주기

02. 중앙처리장치

III. 중앙처리장치 관련 용어

▪ 인터럽트

- **인터럽트(Interrupt)** : 중앙처리장치가 특정한 기능을 수행하던 중 급하게 다른 일을 처리하고자 할 때 사용하는 기능.

✓ **TIP.** 컴퓨터는 동시에 여러 가지 일을 처리하는 것처럼 보이지만, 시간을 잘게 쪼개서 매우 빠른 속도로 일을 나눠 수행하는 시분할(Time-sharing) 방식을 이용함.

• 인터럽트의 처리 과정

- 작업을 수행하던 중 인터럽트가 발생하면 컴퓨터는 수행하던 일을 중지하고 현재 상태(CPU 내부의 레지스터 값 등)를 따로 보관함.
- 인터럽트 서비스 루틴(ISR)을 수행해 인터럽트가 처리되면 저장했던 이전 작업의 상태를 복구시켜 수행을 재개.

02. 중앙처리장치

III. 중앙처리장치 관련 용어

■ 인터럽트

• 인터럽트의 종류

- 입출력 인터럽트 (I/O Interrupt) : 입출력 종료 등의 이유로 중앙처리장치 수행을 요청하는 인터럽트.
- 소프트웨어 인터럽트(Software Interrupt) : 중앙처리장치가 명령을 수행하는 중간에 일어나는 인터럽트.

02. 중앙처리장치

III. 중앙처리장치 관련 용어

- 인터럽트

하나 더 알기

그래픽처리장치(GPU)

- 그래픽처리장치 : 3차원 그래픽 처리를 담당하며, 중앙처리장치(CPU)와 비교했을 때 벡터 연산 등 3차원 그래픽에 필요한 연산 능력이 우수함.

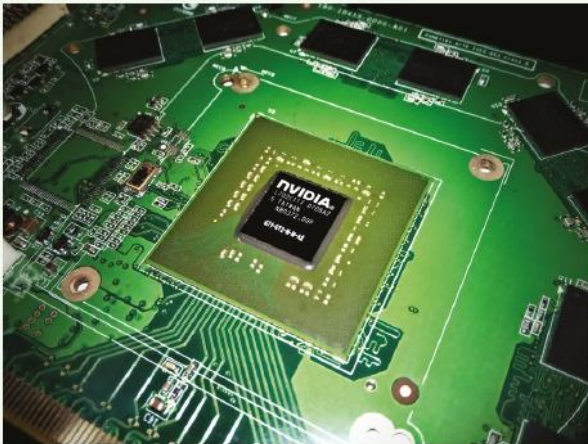


그림 3-11 GPU

03

기억장치

03. 기억장치

- 기억장치의 분류 : 레지스터, 캐시메모리, 주기억장치, 보조기억장치
- 기억장치를 계층으로 나눈 이유는 전체 비용을 고려했을 때 용량과 접근 속도를 최적화하기 위해서임.



그림 3-12 기억장치의 계층 구조

03. 기억장치

I. 주기억장치(메인메모리, Main Memory)

- 주기억장치 : 외부에서 들어온 데이터를 보관하는 공간으로, 작업에 필요한 프로그램과 데이터를 저장하는 장소(이하 메인메모리).

청소를 하자...



그림 3-13 메인보드에 장착된 메인메모리(RAM)

03. 기억장치

I. 주기억장치(메인메모리, Main Memory)

- 메인메모리는 0이나 1을 저장할 수 있는 비트 단위의 메모리가 모여 만들어진 바이트로 구성되어 있는데, 메모리의 주소는 00000000번지부터 시작함.

주소	내용
00000000	11010100
00000001	01010100
.	.
.	.
.	.
11111111	00101010

그림 3-14 메인메모리의 개념도

- 메인메모리의 구분 : 램(RAM)과 롬(ROM)

03. 기억장치

I. 주기억장치(메인메모리, Main Memory)

■ 램

- 램(RAM, Random Access Memory) : 전원이 끊기면 저장된 내용이 지워지는 휘발성 메모리.
- 램의 분류
 - 디램(DRAM) : 일정 시간이 지나면 저장된 데이터가 사라짐. 일반 메모리
 - 에스램(SRAM) : 전원이 연결되어 있는 동안에는 데이터를 계속 보관할 수 있음.

전원이
꺼지면
사라짐



그림 3-15 다양한 램 형태

03. 기억장치

I. 주기억장치(메인메모리, Main Memory)

■ **롬** 부팅 전 -> CMOS -> Bootstrap Loader

- **롬(ROM, Read Only Memory)** : 전원이 제거되어도 저장된 내용은 지워지지 않는 비휘발성 메모리(Non-volatile Memory)로, 랜덤 접근의 특징을 가지고 있음.

- **롬의 분류**

- 마스크롬(Mask ROM)
- 피롬(PROM)
- 이피롬(EPROM) 초기화 후 사용 -> 읽기 쓰기 가능
- 이이피롬(EEPROM) 초기화 안해도 된다 -> 읽기 쓰기 가능



그림 3-16 다양한 롬 형태

03. 기억장치

II. 캐시메모리

- **캐시메모리(Cache Memory)** : CPU와 이보다 상대적으로 느린 메인메모리 사이에 위치한 장치.
- **캐시메모리의 목적** : CPU가 메인메모리의 특정 주소에 접근할 때 그 주변까지 캐시메모리로 읽어 내, 향후 비슷한 곳으로 찾아가야 할 때 접근 속도를 높이는 것이 사용 목적임.



그림 3-17 캐시메모리의 위치

03. 기억장치

II. 캐시메모리

- 캐시메모리는 지역성의 원칙을 활용한 메모리
 - 시간적 지역성(Temporal Locality) : 한 번 사용한 정보는 곧 다시 쓰일 가능성이 높다는 의미.
 - 공간적 지역성(Spatial Locality) : 한 번 사용된 적 있는 정보 영역의 주변부가 다시 사용될 가능성이 높다는 의미.

03. 기억장치

II. 캐시메모리

- 캐시메모리의 동작 과정

- ① CPU가 메인메모리에 있는 데이터를 읽어야 할 때는 먼저 캐시메모리를 조사한다.
- ② 캐시메모리에 데이터가 있으면 바로 CPU로 전달된다.
- ③ 만일 데이터가 없다면 메인메모리에서 보통 블록 단위로 가져온다.
- ④ CPU에는 워드 단위로 전송한다.

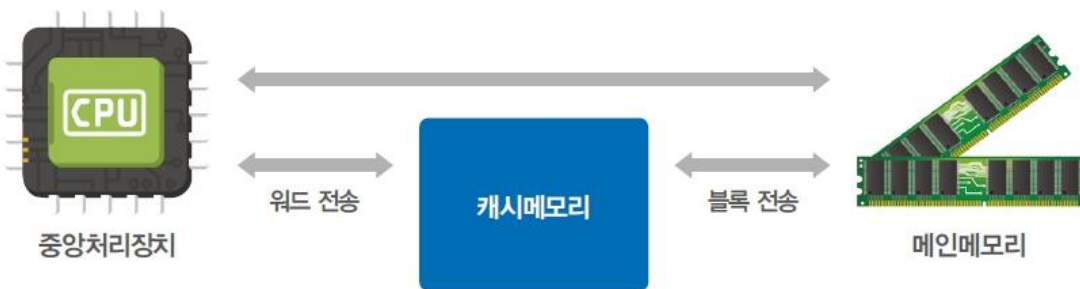


그림 3-18 캐시메모리의 개념도

03. 기억장치

II. 캐시메모리

- 캐시메모리의 동작 방식

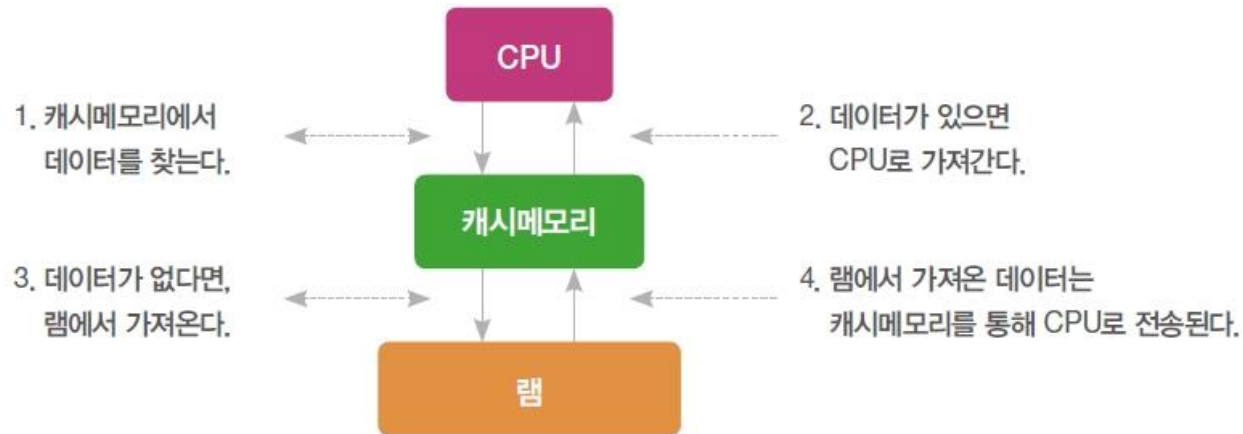


그림 3-19 캐시메모리의 동작 방식

03. 기억장치

II. 캐시메모리

- 캐시 적중(Cache Hit) : CPU에서 원하는 정보가 캐시메모리에 존재하는 상황.
- 캐시 미스(Cache Miss) : 원하는 정보가 존재하지 않는 상황.
- 캐시 적중률(Cache Hit Ratio) : 원하는 정보가 캐시메모리에 존재할 확률.

$$\bullet \text{ 캐시 적중률(H)} = \text{캐시 적중 횟수} \div \text{전체 기억장치 참조 횟수}$$

- 캐시메모리를 사용하는 이유 : 메인메모리에 있는 데이터를 좀 더 빠르게 사용하기 위해서임.

03. 기억장치

III. 보조기억장치

- **보조기억장치(Secondary Memory)** : CPU가 처리할 데이터나 프로그램을 저장하기 위한 기억장치로, 접근 속도가 느리지만 저렴하고 정보를 많이 저장할 수 있음.
- **보조기억장치의 종류**
 - 자기테이프
 - 자기디스크
 - 광디스크
 - 플래시메모리

03. 기억장치

III. 보조기억장치 순차적 접근

▪ 자기테이프

- 자기테이프(Magnetic Tape) : 많은 양의 정보를 기록하기 위해 개발된 테이프 형식의 외부 기억장치로, 플라스틱 테이프 표면에 자성체를 바른 형태.



그림 3-20 자기테이프

03. 기억장치

III. 보조기억장치

▪ 자기디스크

- 자기디스크(Magnetic Disk) : 자성체를 코팅한 원형의 플라스틱이나 금속판 여러 장을 하나의 축에 고정시켜 그 판의 양면에 정보를 저장하는 기억장치임.
- 자기디스크의 종류 : 하드디스크



03. 기억장치

III. 보조기억장치

■ 자기디스크

- 자기디스크는 비휘발성 자성체로 표면을 코팅한 원형의 회전판으로 구성되어 있음.
- 정보는 트랙이라 불리는 디스크 표면의 동심원에 나뉘져있는데, 트랙은 파이 조각 모양의 섹터로 구성되어 있음.
- 하드디스크는 여러 개의 디스크 판으로 구성되어 있으며, 디스크 헤드가 움직이면서 각 트랙을 이동함.

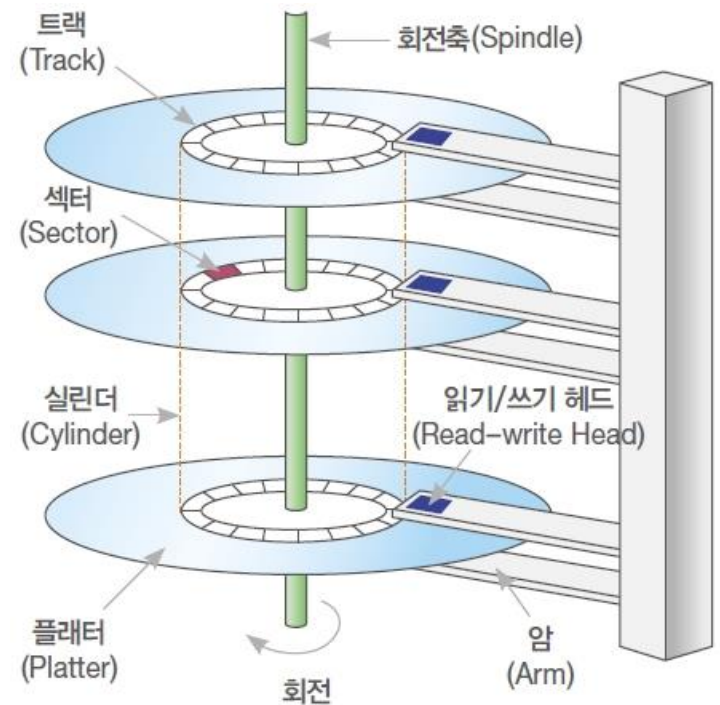


그림 3-22 자기디스크의 구조

02. 중앙처리장치

III. 보조기억장치

▪ 자기디스크

하나 더 알기

자기디스크를 사용하기 위한 접근 시간

- **직접 접근 방식** : 자기디스크 같은 기억장치는 디스크 헤드를 원하는 트랙으로 직접 이동시키는 방식.
- **직접 접근 방식의 기억장치가 데이터를 저장하거나 재생하는 단계**
 - ① 디스크 헤드를 적당한 트랙으로 옮긴다(Seek Time).
 - ② 디스크를 회전시켜 섹터를 찾는다(Rotation Latency).
 - ③ 디스크 헤드를 통해 데이터를 전송한다(Transfer Time).
- **디스크를 사용하는 데 필요한 접근 시간(수행 시간)**

• 디스크 접근 시간 = ①번 과정 + ②번 과정 + ③번 과정을 수행하는 데 드는 시간

03. 기억장치

III. 보조기억장치

▪ 플래시메모리

- 플래시메모리(Flash Memory) : 비휘발성 반도체 기반의 메모리로, 속도가 빠르고 크기가 작으며 전력소비가 적고 충격에 강함.
- 플래시메모리의 종류 : USB 메모리, 메모리 카드, 디지털카메라, SSD 등



(a) 메모리 카드



(b) USB



(c) SSD

그림 3-24 플래시메모리의 종류

04

입출력장치

04. 입출력장치

III. 입출력장치

▪ 동작인식 기기

- 동작인식 기기 : 신체 등을 움직여 컴퓨팅 기기에 입력하는 입력장치.
 - 키넥트 : MS의 동작인식 카메라를 사용해 별도의 제어기 없이도 몸동작으로 게임을 즐길 수 있는 장치.
 - 립모션 : 두 손과 손가락의 움직임으로 컴퓨터 제어를 가능하게 하는 장치.



(a) 키넥트를 이용한 제스처 인식

그림 3-27 동작인식 기기

(b) 립모션을 이용한 손동작 인식

04. 입출력장치

III. 입출력장치

▪ 3D 프린터

- **3D 프린터** : 입력된 도면을 바탕으로 3차원 입체 물품을 제작하는 출력장치로, 최근에 기계부품이나 건축물 제작 등으로 영역이 확대되었으며 인공 근육이나 인공 뼈까지 제작하는 수준에 다다름.

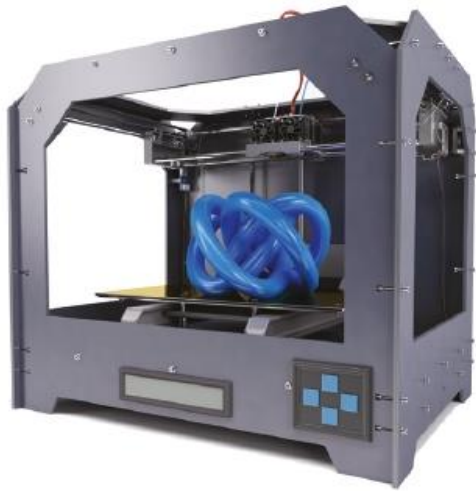


그림 3-28 3D 프린터



[‘혈관 갖춘’ 인공심장 3D 프린팅 첫 성공\(01:15\)](#)

05

병렬 컴퓨터

05. 병렬 컴퓨터

I. 병렬 컴퓨터의 개념

- **병렬 컴퓨터(Parallel Computer)** : 다수의 CPU를 병렬 처리해 초고속으로 작업을 수행하는 컴퓨터.
- 컴퓨터 부품의 초집적화가 진행될수록 발열 문제가 심각해지자 CPU의 소형화 작업도 진행이 어려워졌는데, 이러한 문제를 해결하기 위해 등장한 것이 바로 병렬 컴퓨터임.

05. 병렬 컴퓨터

II. 병렬 처리

- **병렬 처리(Parallel Processing)** : 동시에 동작하는 CPU를 여러 대 갖추고 있는 컴퓨터에서만 실행할 수 있는 처리 방식.
- **병렬 처리의 분류** : SIMD, MIMD, SISD

05. 병렬 컴퓨터

II. 병렬 처리

▪ SIMD

- **SIMD(Single Instruction Multiple Data)** : 모든 CPU가 같은 프로그램(명령어)을 수행하지만 병렬적으로는 다른 데이터를 처리하는 구조로, 가장 일반적인 병렬 연산 기법임.

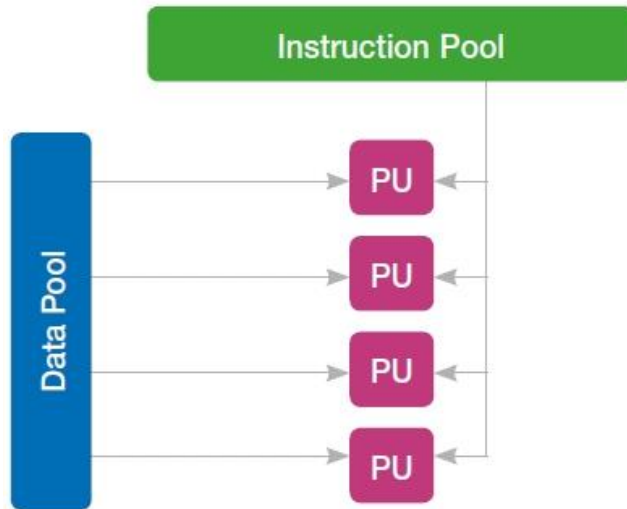


그림 3-29 SIMD 구조

05. 병렬 컴퓨터

II. 병렬 처리

▪ MIMD

- **MIMD(Multiple Instruction Multiple Data)** : 각각의 CPU가 서로 다른 프로그램을 수행하면서 서로 다른 데이터를 처리하는 구조.

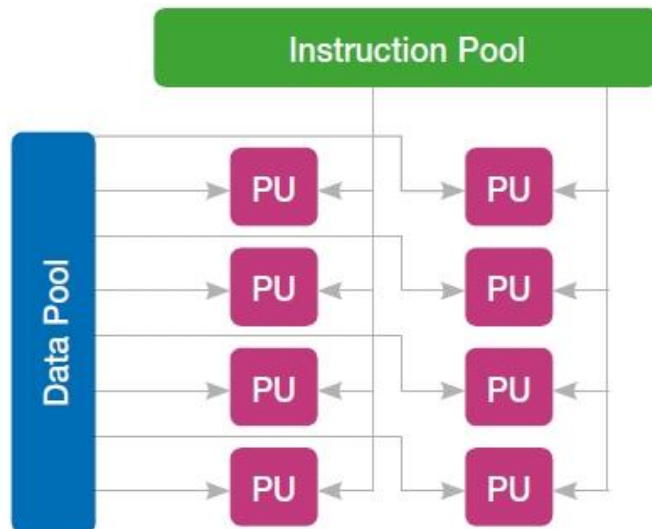


그림 3-30 MIMD 구조

05. 병렬 컴퓨터

II. 병렬 처리

▪ SISD

- **SISD(Single Instruction Single Data)** : 하나의 프로그램을 수행할 때 하나의 데이터를 이용해 처리하는 구조로, 일반 개인용 컴퓨터의 CPU 대부분이 SISD 구조임.

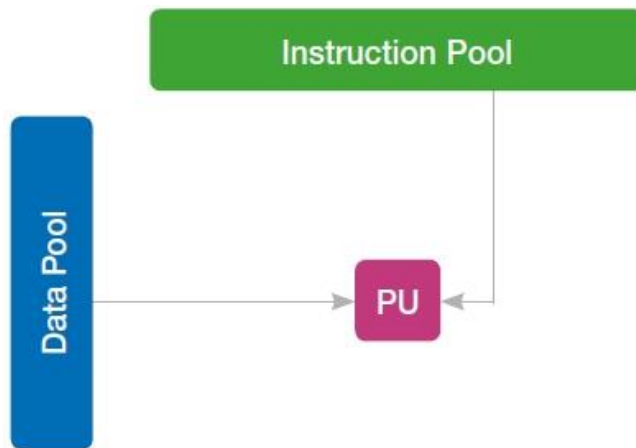


그림 3-31 SISD 구조

05. 병렬 컴퓨터

III. 병렬 컴퓨팅 환경

▪ 소규모 멀티코어 환경 : 멀티코어 CPU

- 멀티코어 CPU : 2개 이상의 코어(기억 소자)를 탑재한 CPU.
 - 멀티코어 CPU를 사용하게 된 데는 초고밀도집적회로와 연관이 있음. 무어의 법칙에 따라 1977년부터 18~24개월마다 반도체 트랜지스터의 집적도는 2배씩 증가했는데, 최근 들어 CPU의 냉각 문제가 발생하면서 회로의 집적도를 높여가기가 어려워짐. 그래서 여러 개의 코어를 집적한 멀티코어 CPU를 사용하게 됨.
- ✓ TIP. 코어(Core) : CPU에 내장된 처리 회로의 핵심 부분.

05. 병렬 컴퓨터

III. 병렬 컴퓨팅 환경

▪ 소규모 멀티코어 환경 : 멀티코어 CPU

- 멀티코어 프로세서 상황에서 성능을 최대한 이용하려면 병렬 프로그래밍 개념을 사용해야 함.
- 최근에 나온 개인용 컴퓨터는 멀티코어 CPU를 사용하고 있으며, 운영체제가 이를 지원.
- GPGPU, CUDA, OpenMP 등의 소프트웨어 개발 환경이 지원되면서 전보다 훨씬 쉽게 병렬 프로그래밍이 가능해짐.

05. 병렬 컴퓨터

III. 병렬 컴퓨팅 환경

- 소규모 멀티코어 환경 : 멀티코어 CPU

✓ TIP. GPGPU(General Purpose Graphic Processing Unit) : 고성능 그래픽 카드를 이용한 병렬 컴퓨팅 환경.

✓ TIP. CUDA(Compute Unified Device Architecture) : NVIDIA사에서 제공하는 GPGPU를 이용한 병렬 컴퓨팅 아키텍처.



그림 3-32 멀티코어 CPU

05. 병렬 컴퓨터

III. 병렬 컴퓨팅 환경

▪ 대규모 병렬 컴퓨팅 환경 : 클러스터와 그리드

- 클러스터(Cluster) : 컴퓨터를 수십 대에서 수만 대까지 연결해 하나의 컴퓨터로 사용하는 환경으로, 근거리 네트워크(LAN)를 통해 연결된 컴퓨터들이 하나의 대형 멀티프로세서로 동작하는 시스템임.

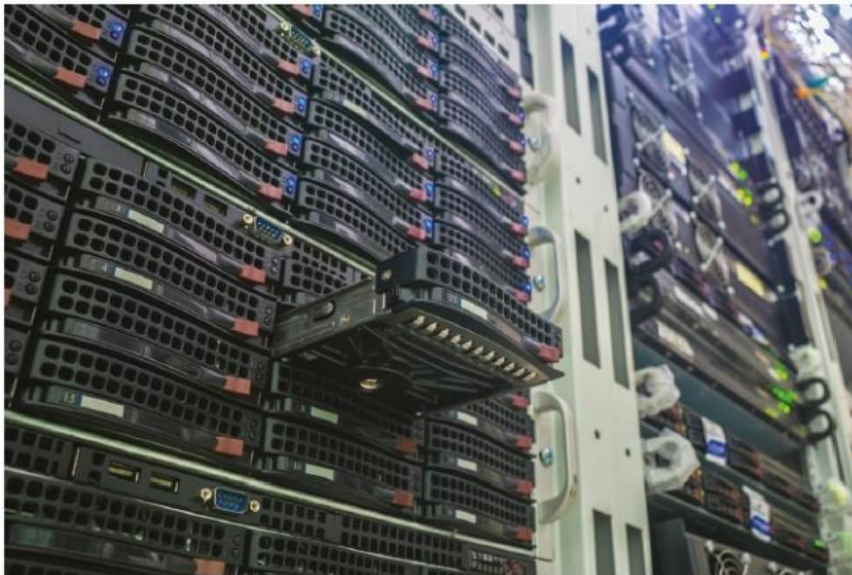


그림 3-33 클러스터

05. 병렬 컴퓨터

III. 병렬 컴퓨팅 환경

▪ 대규모 병렬 컴퓨팅 환경 : 클러스터와 그리드

- **그리드 컴퓨팅(Grid Computing)** : 이종의 리소스들로 구성되어 있으며, 광역 네트워크 기반의 동적인 구조임.
- 그리드 컴퓨팅은 네트워크에 연결된 여러 대의 컴퓨터에 데이터를 전송하여 연산하게 한 후, 서버에서 취합해 전체 연산을 수행.

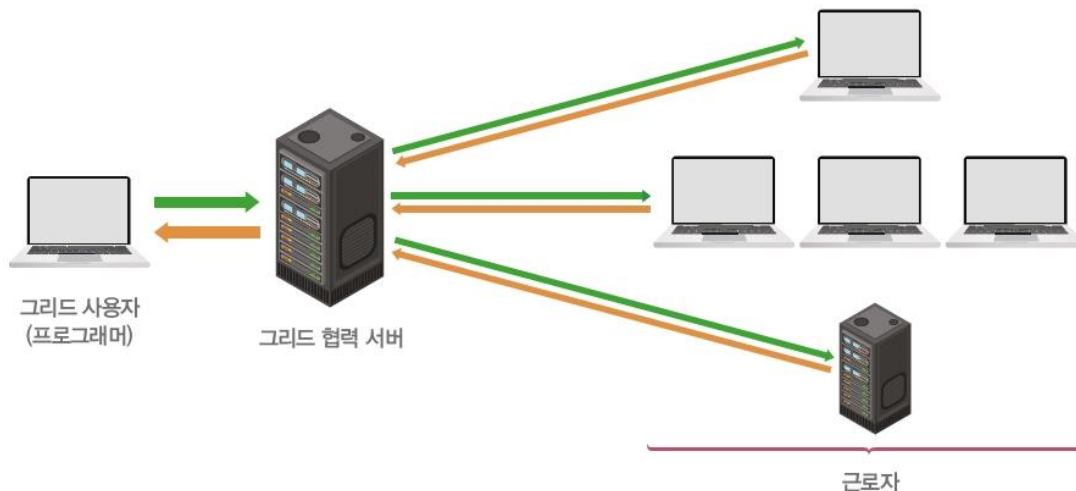


그림 3-34 그리드 컴퓨팅의 구조