

# 제2장 서버를 열어보자

## 2.1 물리 서버

### 2.1.1 서버 외관과 설치 장소 / 2.1.2 서버 내부 구성

## 2.2 CPU

## 2.3 메모리

## 2.4 I/O장치

### 2.4.1 하드 디스크 드라이브 / 2.4.2 네트워크 인터페이스 / 2.4.3 I/O 제어

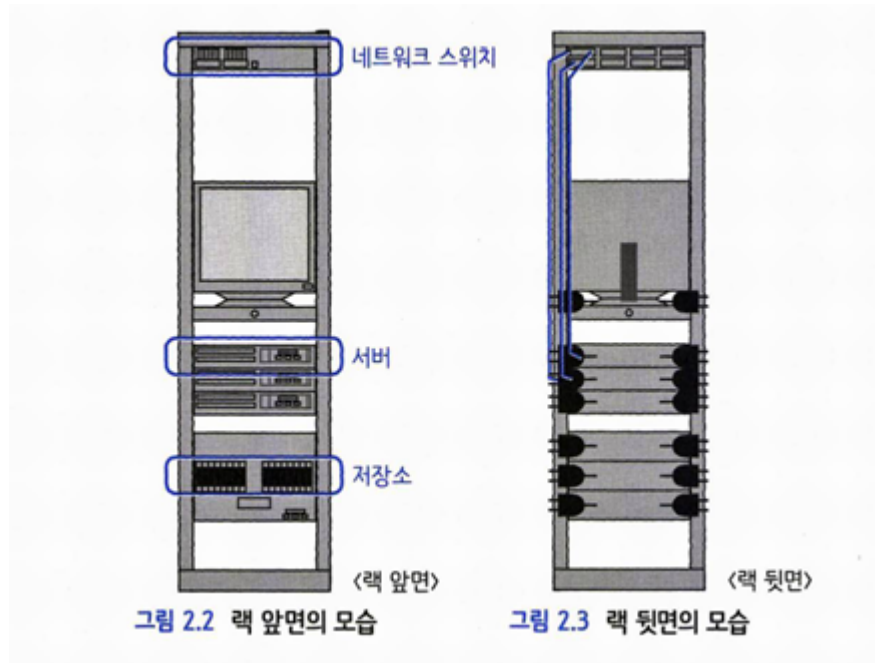
## 2.5 버스

## 2.1 물리 서버

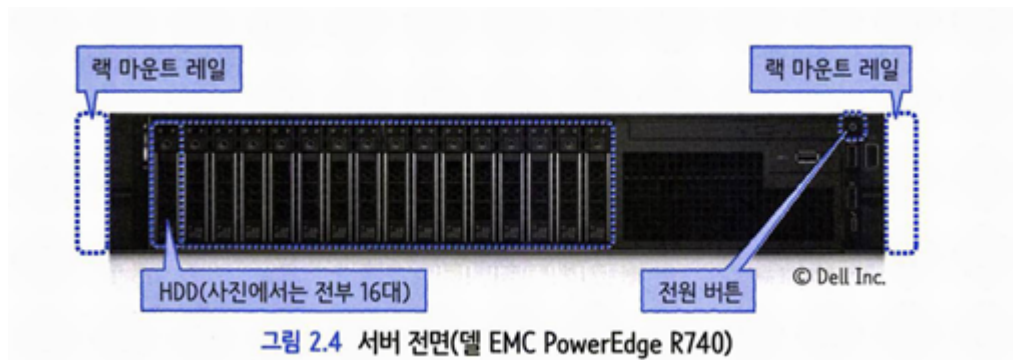
### 2.1.1 서버 외관과 설치 장소

- 서버실은 서버가 대량으로 설치돼 있고, 서버에서 나오는 열기를 식히기 위해 실내 온도를 항상 낮게 설정할 뿐더러 빛도 들어오지 않는 환경이다.

서버는 랙(Rack)이라는 것에 장착된다. 랙에는 서버 외에도 HDD가 가득 장착돼 있는 저장 소나 인터넷 및 LAN을 연결하기 위한 네트워크 스위치 등도 탑재돼 있다. 전원이나 네트워크 케이블 배선 등은 모두 랙 뒷면에서 연결된다.



대표적인 서버 아키텍처 중 하나인 인텔의 CPU를 사용한 IA 서버에 대한 설명이다.

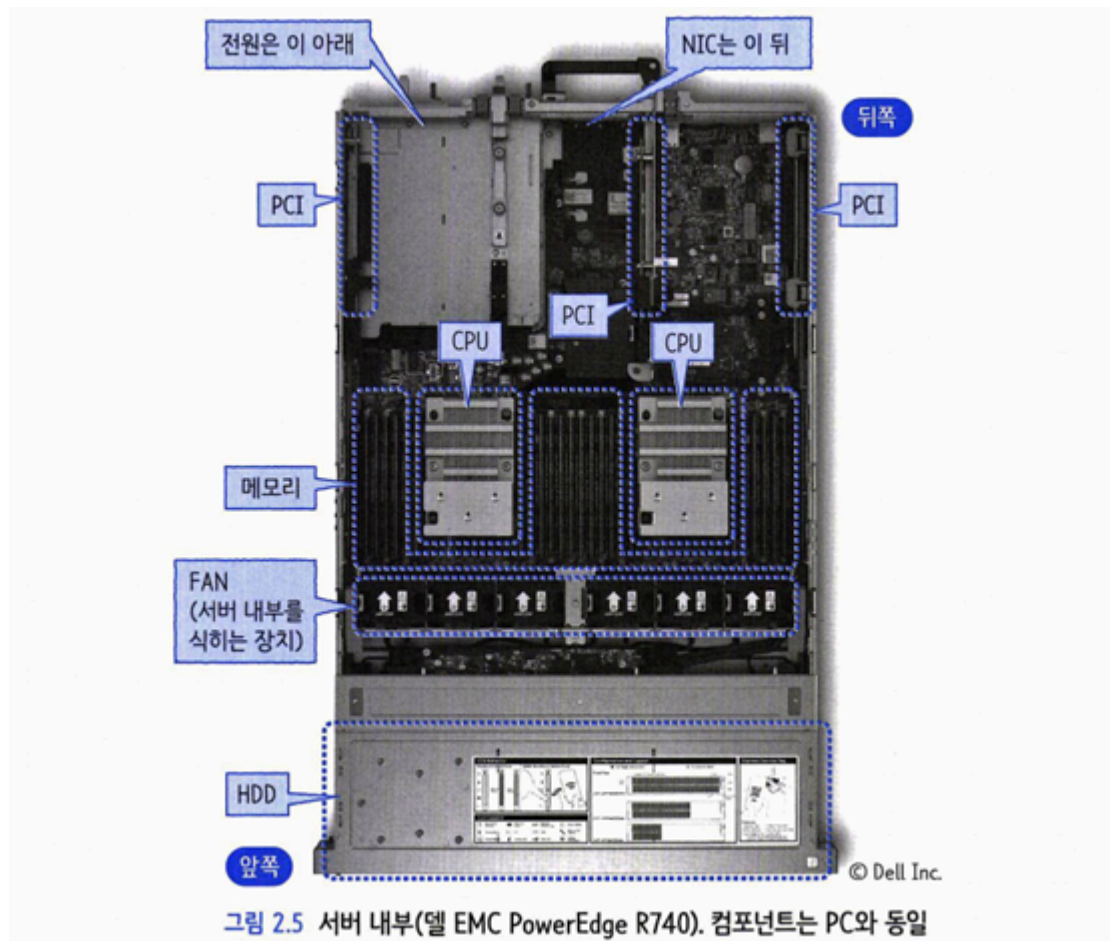


델 테크놀러지 가 판매하는 델 EMC PowerEdge R740이라는 모델의 서버 전면 사진이다. 일반적인 서버는 이렇게 옆으로 긴 형태다. 또한, 옆에 랙 마운트 레일 이라는 것이 있어서 장롱 서랍처럼 설치할수 있다.

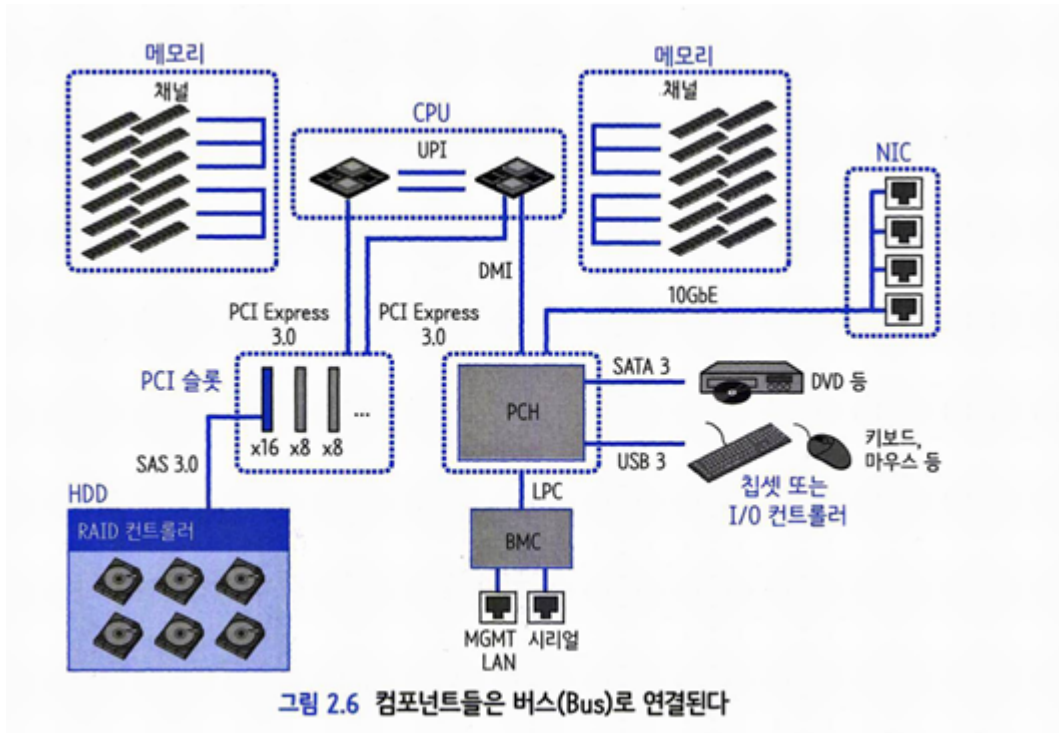
전면에는 HDD나 전원 버튼 등이 있다. HDD는 교체하기 쉽게 되어 있어서 손으로 당겨 꺼낼 수 있다.

## 2.1.2 서버 내부 구성

- 이런 서버는 위쪽 뚜껑을 열 수 있다. 그림 2.5는 뚜껑을 열어 놓은 사진이다.



이것은 PowerEdge R740의 CPU인 Intel Xeon 프로세서를 사용한 버스 접속의 일반적인 예다



컴포넌트를 연결하는 선을 '버스(Bus)'라고 한다. CPU가 두 개 연결돼 있고, 그 옆에 메모리가 배치돼 있다. CPU와 메모리는 물리적으로 직접 연결되는 것을 알 수 있다. PCI Express 슬롯이라는 것이 있다. 이것은 외부 장치를 연결하는 곳이다. Xeon 확장 프로세서 아키텍처에선 CPU가 PCI를 직접 제어한다. 이 서버에서는 칩셋이 네트워크 인터페이스를 4개까지 직접 제어할 수 있다. 아래에는 BMC는 컴포넌트가 있다. 이것은 서버의 H/W 상태를 감시하며, 독립적으로 움직인다.

## 2.2 CPU

-CPU는 Central Processing Unit의 약자다. 서버 중심에 위치해서 연산 처리를 실시한다.

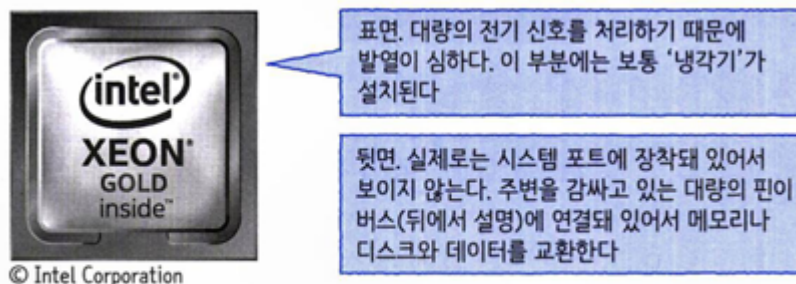


그림 2.7 CPU(Xeon 확장형 프로세서, Intel Xeon Gold 5115)

CPU는 명령을 받아서 연산을 실행하고 결과를 반환한다. 명령과 데이터는 기억 장치나 입출력 장치를 통해 전달된다. 현재는 이 CPU를 '코어(core)'라고 하며, 하나의 CPU에 여러 개의 '코어'가 존재하는 멀티 코어화가 진행되고 있다. '코어'는 각자가 독립된 처리를 할 수 있다.

명령이나 데이터는 기억 장치에 있지만 운영체제(Operating System, OS)라는 소프트웨어에서 명령을 내리고, OS에서 동작하는 웹 서버나 데이터베이스의 실체인 '프로세스와 사용자 키보드, 마우스 등을 통한 입력을 통해 OS에 명령을 내린다. CPU가 자발적으로 처리하는 것은 아니다.

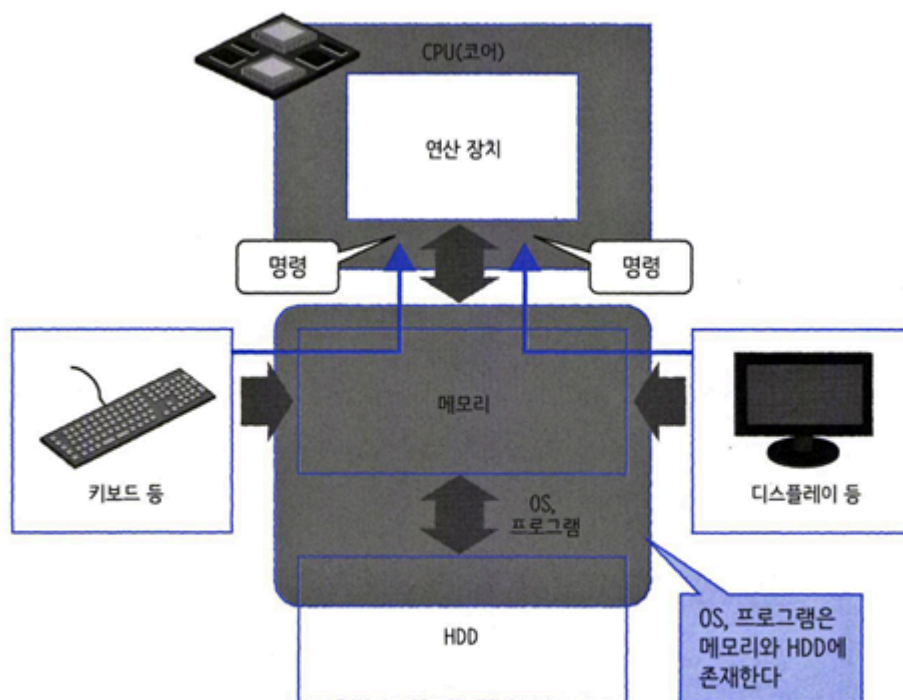
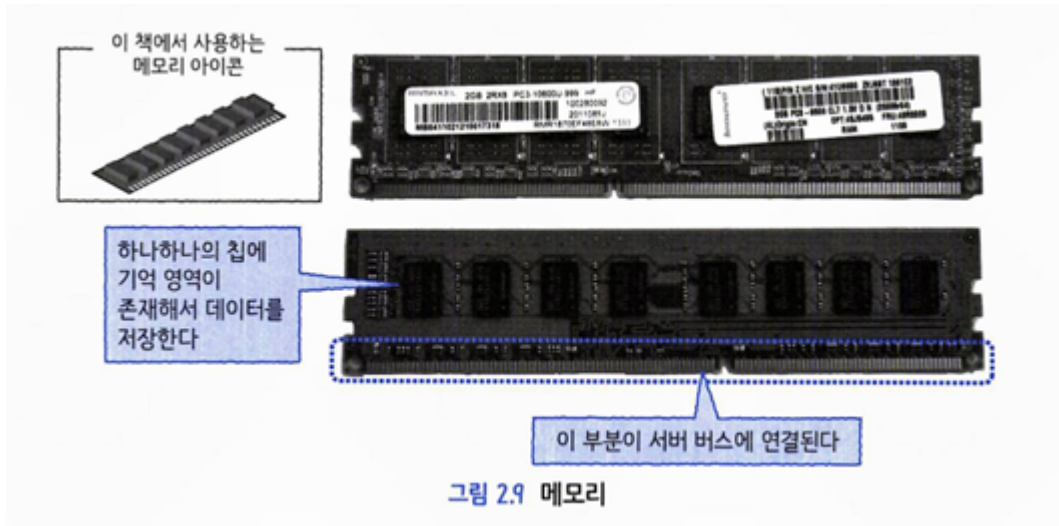


그림 2.8 CPU는 명령을 기다리고 있다

## 2.3 메모리

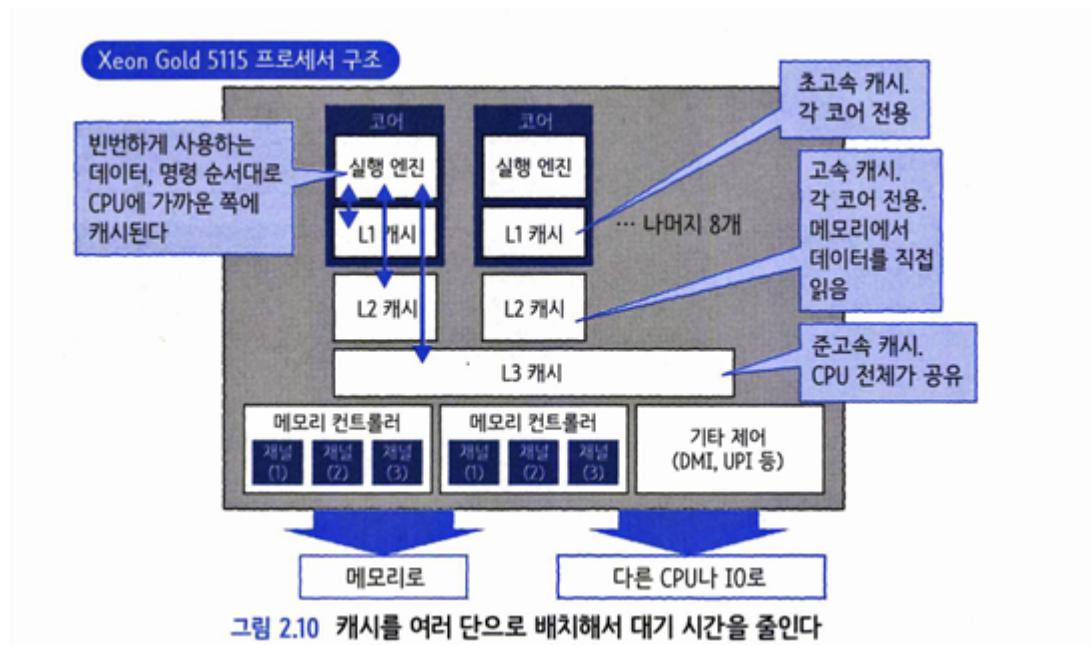
-메모리는 그 명칭 그대로 기억 영역을 말한다(그림 2.9). CPU 옆에 위치하며, CPU에 전달하는 내용이나 데이터를 저장하거나 처리 결과를 받는다





메모리에 저장되는 정보는 영구성이 없다. '영구성이 없다'라는 것은 서버를 재시작하면 없어지는 정보임을 의미한다. 이런 결점에도 메모리를 사용하는 이유는 메모리 액세스가 매우 빠르게 이루어지기 때문이다. 그리고 데이터 저장 시에 물리적인 모터 등을 구동하는 것이 아니라 전기적인 처리만으로도 데이터를 저장하기 때문이다.

그런데 CPU 자체도 메모리를 가지고 있다. 이것은 레지스터나 1차(L1)/2차(L2)캐시라고 불리며, CPU 내부에 존재한다. 메모리보다 더 빠르긴 하지만, 용량이 메모리에 비해 매우 작다.



영역이 여러 단계로 나누어져 있는 이유는 액세스 속도 때문이다. 일반적으로 캐시 메모리가 커질 수록 액세스 속도가 느려진다. 하지만 가능한 CPU 가까운 곳에 많은 캐시를 두고 싶은 것도 사실이다. 이 때문에 캐시를 여러 단계로 배치해서 초고속으로 액세스하고 싶은 데이터는 L1 캐시에, 준고속으로 액세스하고 싶은 데이터는 L2 캐시에 두는 형태로 만든 것이다.

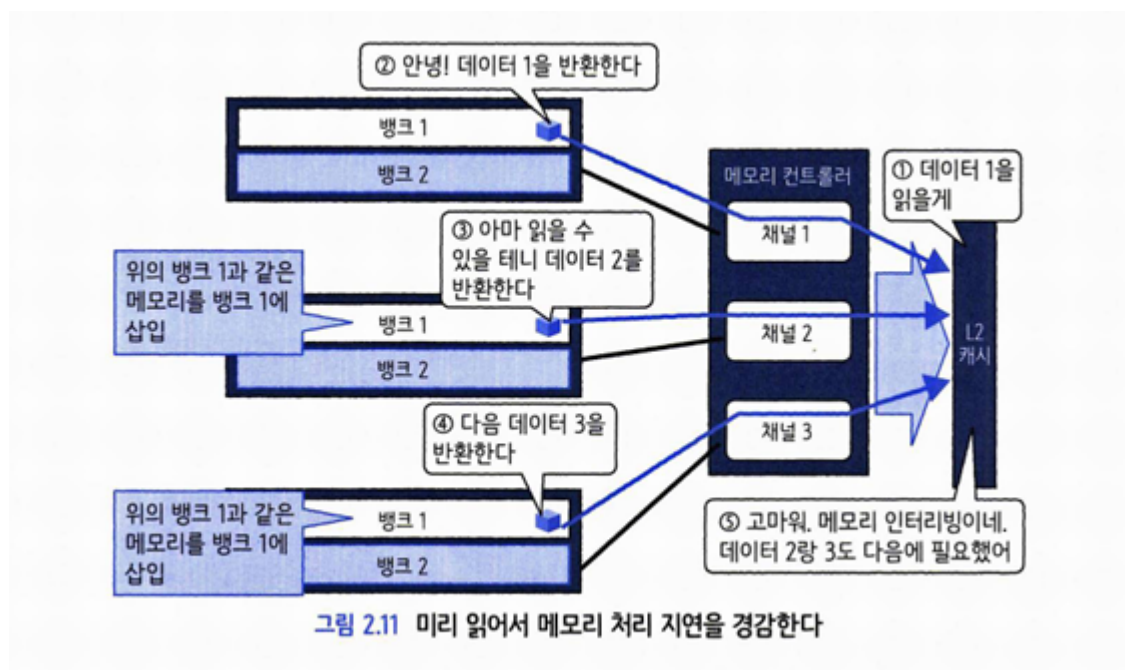
또한, 메모리에는 미리 데이터를 CPU에 전달해서 처리 지연을 줄이는 '메모리 인터리빙' 이라는 기능이 있다.

그림 2.10의 왼쪽 아래에 메모리 컨트롤러와 채널이 있다. 채널은 메모리와 CPU 간 데이터 경로를 말한다.

최대 세 개의 채널을 사용해서 데이터 1을 요구하면 데이터 2와 3도 함께 보내 버린다. 이것은 대부분의 데이터가 연속해서 액세스 된다는 규칙을 기반으로 만들어진 것이다. 먼저 읽어서 처리 지연을 줄여 주는 것이다.

이 기능을 활용하기 위해서는 모든 채널의 동일 뱅크에 메모리를 배치해야 한다. 채널 영역도 많이 사용할 수 있다. 다른 제조사의 CPU나 메모리도 원칙은 같기 때문에 사양을 확인해 보도록 하자.

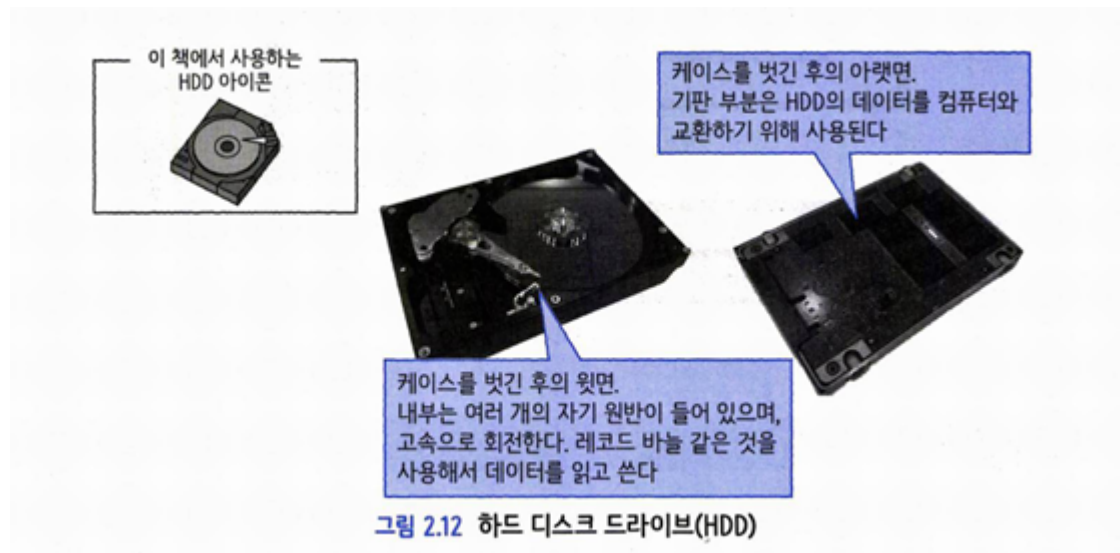
이와 같이 메모리는 다단계 구조를 가지고 각각의 액세스 속도에 맞게 사용되기 때문에 CPU의 데이터 처리 속도를 줄일 수 있다



## 2.4 I/O장치

### 2.4.1 하드디스크 드라이브(HDD)

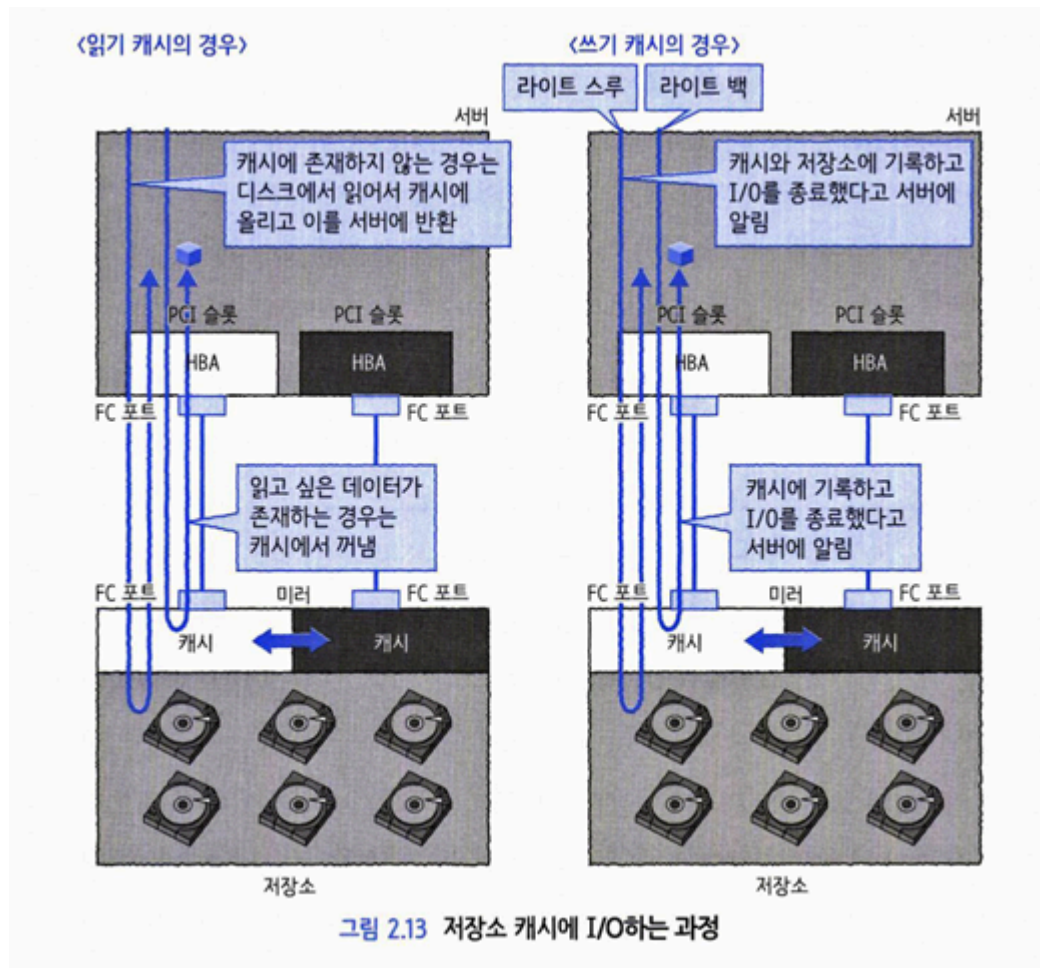
-서버에서는 메모리에 비해 CPU에서 떨어진 곳에 HDD가 배치된다. 주로 장기 저장 목적의 데이터 저장 장소로 사용한다. 메모리도 디스크도 기억 영역이지만 액세스 속도가 다르며, 전기가 흐르는지 여부에 따라 데이터가 손실되거나 손실되지 않거나 하는 점이 다르다. 메모리는 전기가 흐르지 않으면 데이터가 사라지며, 디스크는 전기가 없어도 데이터가 사라지지 않는다.



아직 이런 HDD가 많은 시스템에서 이용되고 있지만, 최근에는 기술이 발달해서 SSD (Solid State Disk, 반도체 디스크) 라는, 물리적인 회전 요소를 사용하지 않는 디스크가 사용되고 있다. SSD는 메모리와 같이 반도체로 만들어졌지만, 전기가 없어도 데이터가 사라지지 않는다. HDD가 많이 탑재돼 있는 하드웨어를 '스토리지 (Storage, 저장소)' 라고 한다. 저장소는 I/O의 서브 시스템이라고도 불리는 장치로서, 내부에는 CPU와 캐시가 존재하고 수 많은 HDD 외에도 여러 기능을 탑재하고 있다.

서버와 I/O 시에는 HDD가 직접 데이터 교환을 하는 것이 아니라 캐시를 통해서 한다.



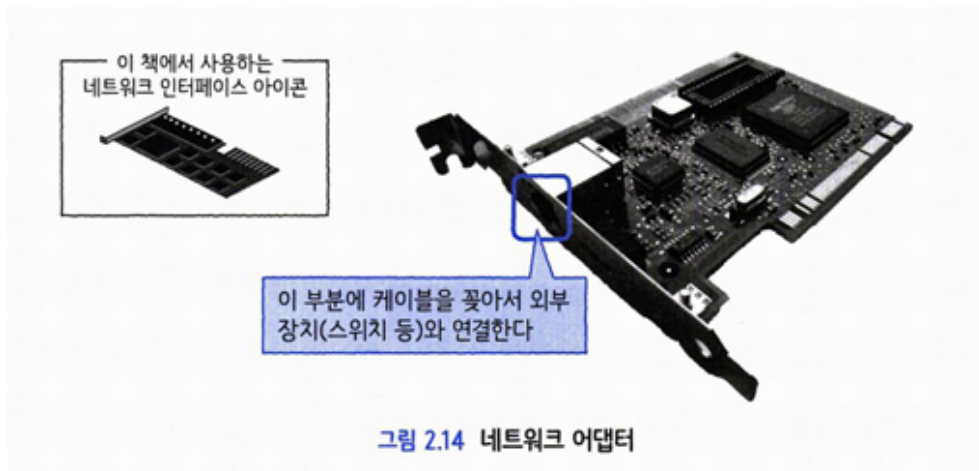


대형 저장소와 연결할 때는 일반적으로 '파이버 채널' 이라는 케이블을 사용해서 SAN(Storage Area Network)이라는 네트워크를 경유한다. SAN에 접속하기 위한 파이버 채널 인터페이스를 FC 포트라고 한다. 보통은 서버 시스템 포트에 FC 포트가 없기 때문에 PCI 슬롯에 HBA 라는 카드를 삽입한다.

하나는 읽기/쓰기 시에 캐시라는 메모리 영역에 액세스하는 방법이다. 읽기 캐시의 경우는 캐시상에 데이터 복사본만 있으면 되지만, 쓰기 시에는 캐시에만 데이터를 기록하고 완료했다고 간주하는 경우 데이터를 잃을 가능성이 있음을 의미한다. 장점은 캐시에 저장해서 쓰기 처리가 종료되기 때문에 고속 I/O를 실현할 수 있다는 점이다. 이런 쓰기 I/O를 '라이트 백 (Write Back)'이라고 한다. 대부분의 저장소 제품에서는 이 캐시를 별도의 캐시와 미러링해서 안정성을 높이고 있다. 다른 하나의 I/O는 캐시와 HDD에 모두 액세스하는 I/O다. 읽기 시에 캐시에 데이터가 없으면 읽기 처리를 위해 액세스한다. 쓰기 시에는 캐시와 디스크를 모두 읽어서 라이트 백과 비교하고, 더 확실한 쪽에 쓰기 처리를 실시하기 위해 액세스한다. 이 경우 쓰기 캐시의 장점은 없다. 이런 쓰기 I/O를 '라이트 스루(Write Through)'라고 한다. 기본적으로는 캐시의 장점을 살리기 위해 '라이트 백'으로 설정한다.

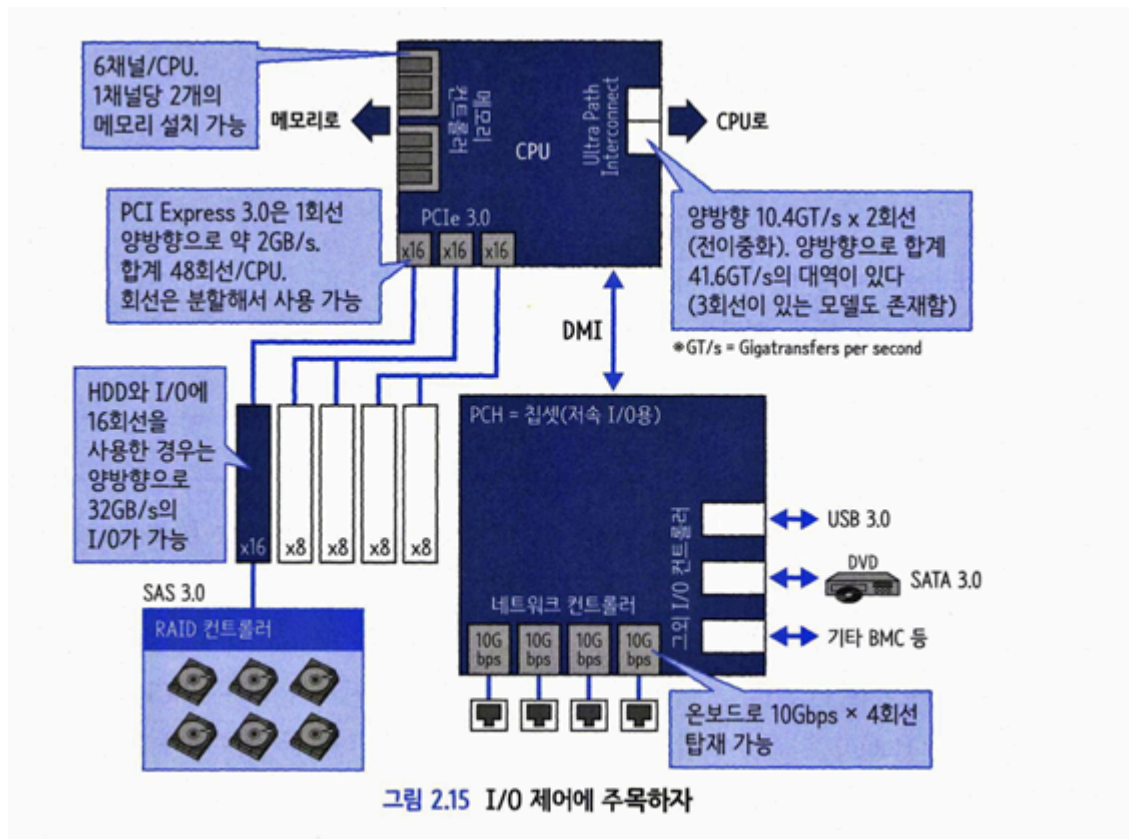
## 2.4.2 네트워크 인터페이스

-네트워크 인터페이스는 서버와 외부 장비를 연결하기 위한 것으로 외부 접속용 인터페이스다.

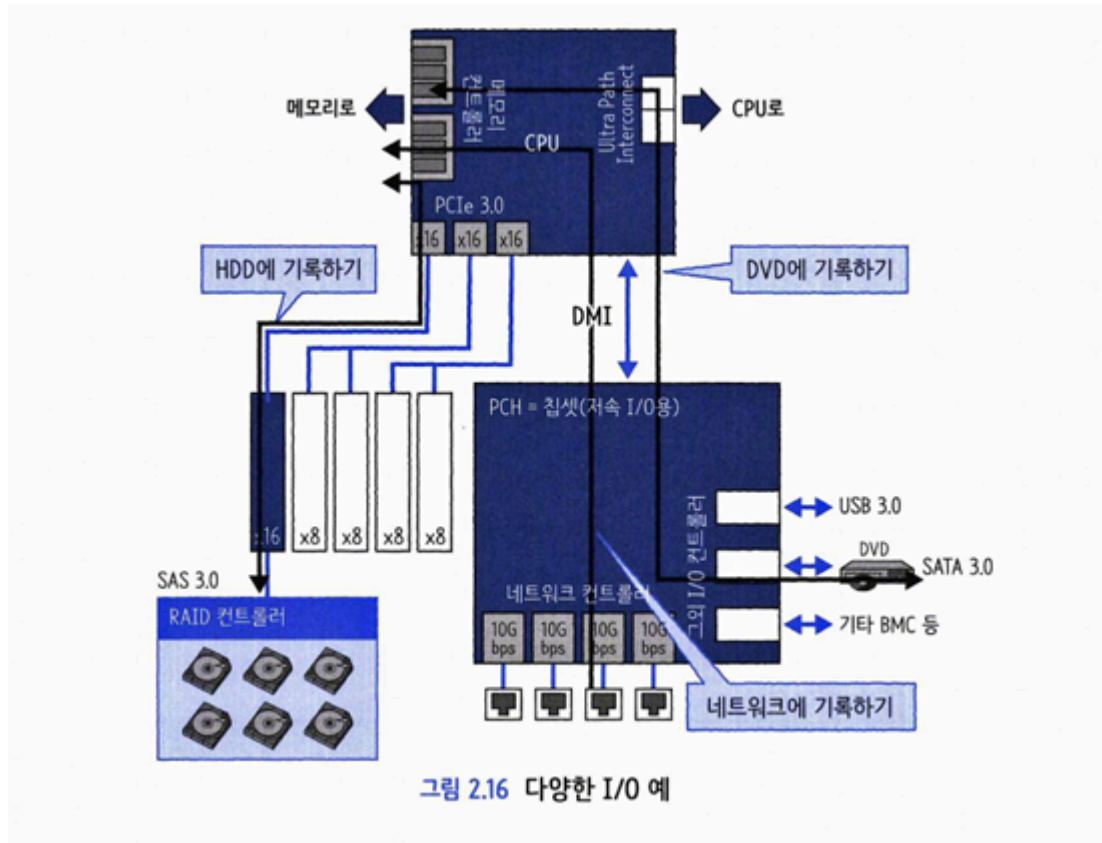


### 2.4.3 I/O제어

-I/O 제어는 발전 속도가 빨라서 일반적인 구성이라는 것이 없다. Xeon 확장형 프로세서의 경우는 PCH(PlatformController Hub)라는 칩셋이 탑재되어 있어서, CPU가 제어하는 메모리나 PCIe(PCI Express) 외의 처리 속도가 비교적 늦어도 용서 되는 I/O 제어를 담당하고 있다. 각 CPU/칩셋 구조마다 PCI 를 연결할 수 있는 회선이 정해져 있다.



I/O 시에 관련 처리를 가능한 I/O와 가까운 곳 (즉, CPU에서 멀리 있는 곳) 에서 처리 하는 것이 더 효율적이다. 즉, CPU와 칩셋의 관계는 역할 분담을 위한 것이다



## 2.5 버스

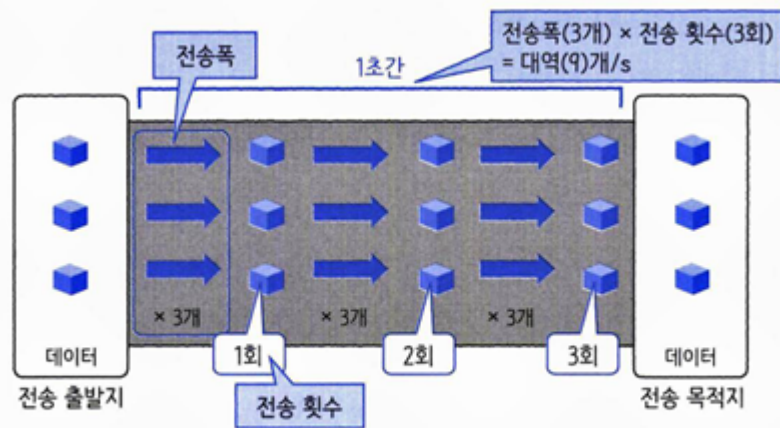
-버스(Bus)는 서버 내부에 있는 컴포넌트들을 서로 연결시키는 회선을 가리킨다.

이 버스에서 중요한 것은 무엇일까? 대답은 버스가 어느 정도의 데이터 전송 능력을 가지고 있는가, 즉 대역이 어느 정도인가가 중요하다.

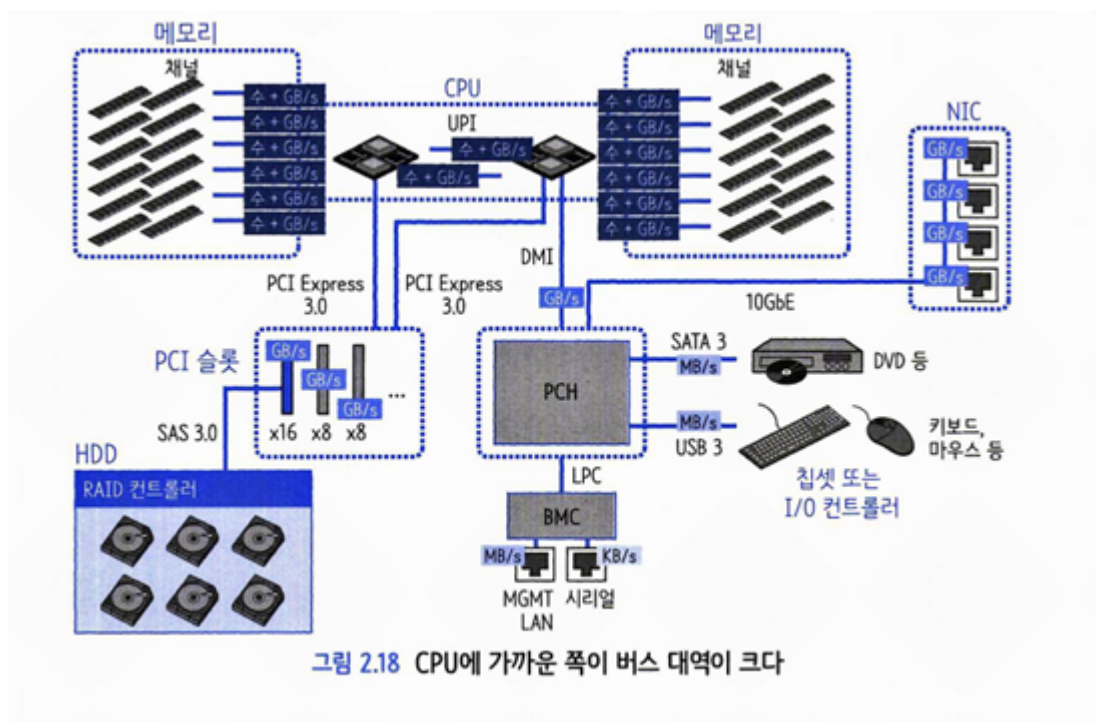
### 2.5.1 대역

-대역이란 무엇일까? 원래는 주파수 대역을 가리키지만, IT 인프라에서는 의미가 조금 다르다. 대역은 데이터 전송 능력을 의미한다. 대역은 '한번에 데이터를 보낼 수 있는 데이터의 폭(전송폭)' x '1초에 전송할 수 있는 횟수(전송 횟수)'로 결정된다.

전송 횟수는 '1초 ÷ 1 처리당 소요 시간(응답 시간)'으로 표현할 수 있다. 또한, 대역은 스루풋(Throughput, 처리량)'이라고도 부른다.



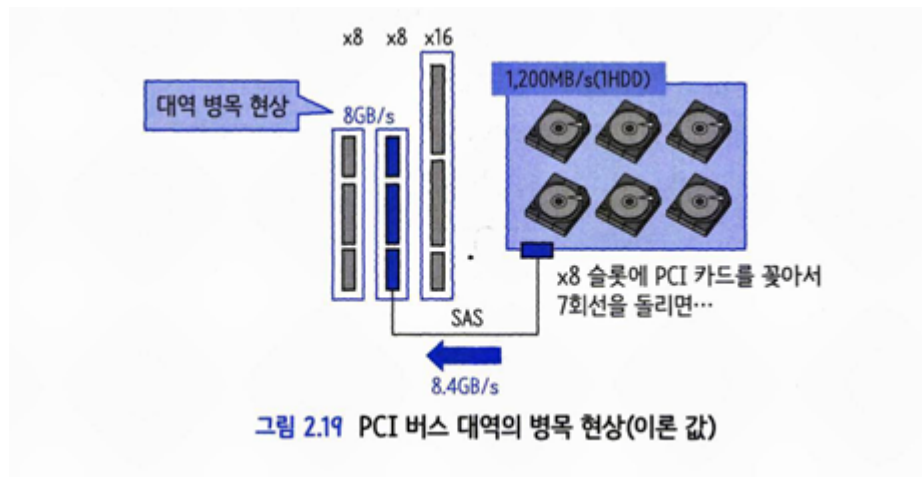
### 2.5.2 버스 대역



CPU에 가까운 쪽이 1초당 전송량이 크다는 것을 알 수 있다.

버스 흐름에서 중요한 것은 CPU와 장치 사이에 병목 현상이 없어야 한다는 것이다. 병목 현상 (Bottleneck) 은 데이터 전송이 어떤 이유로 막혀 있는 상태를 의미한다. 시스템 설계 시에 특히 놓치기 쉬운 것이 외부 장치 연결 시의 버스 대역에 관한 것이다.





## 2.6 정리

-CPU부터 HDD, 네트워크 등의 I/O 장치까지 일련의 데이터 흐름에 대해 살펴보았다. 이 처리 흐름을 그림으로 표현한 것이 그림 2.20이다. HDD 데이터는 다양한 전송 버스를 지나서 몇 번이고 캐시된 후에 CPU에 이른다. 또한, CPU에 가까울수록 고속이고 멀수록 대용량인 것을 알 수 있다.

