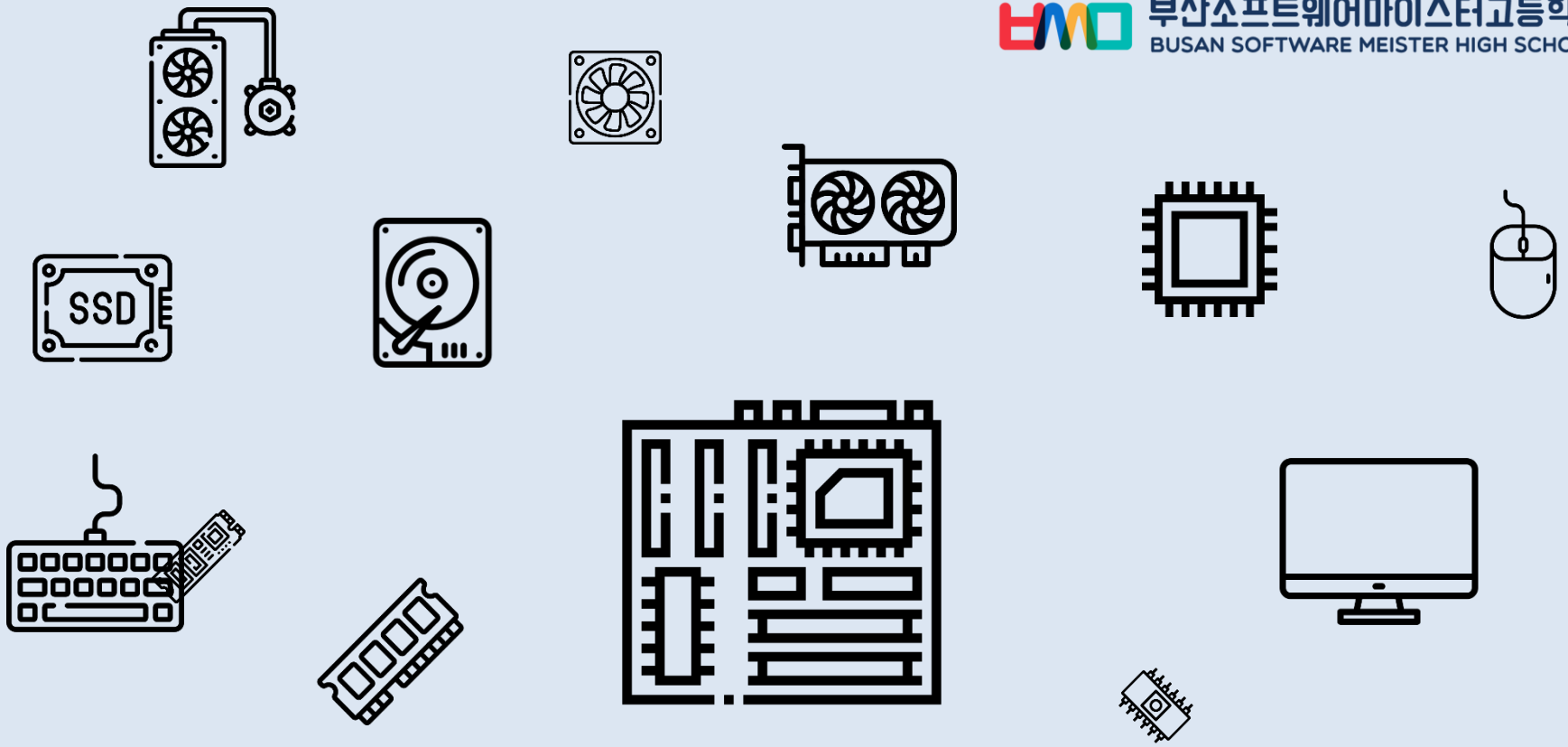


컴퓨터 구조

Chapter 05. 기억 장치

2 계층적 기억 장치 시스템

❖ 기억 장치 시스템에서 액세스 속도, 가격, 용량의 관계

- 액세스 속도가 빨라질수록 비트당 가격은 높아진다.
- 용량이 증가할수록 비트당 가격은 감소하고, 액세스 속도도 낮아진다.

GB당 가격	저장 용량	액세스 시간
100만 달러	수십~수백 Byte	1ns 이하
10만 달러 1만 달러	L1 : 수십 KB L2 : 수 MB	1~10ns
1000달러	수백 MB	10~50ns
10달러	수백 GB	자기 : 10~50ms 광 : 100~500ms
1달러	수 TB	0.5s 이상

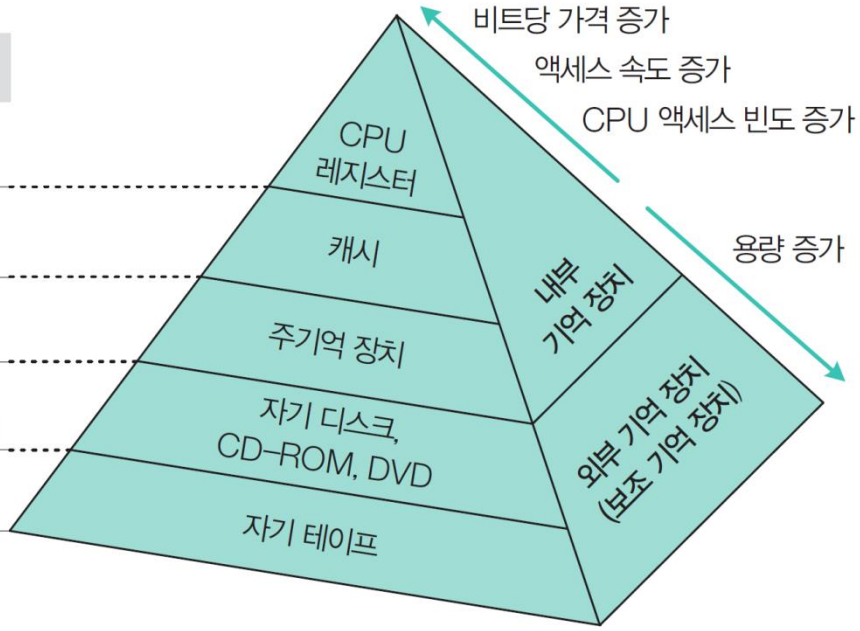


그림 6-1 기억 장치의 계층 구조

평균 액세스 속도를 빠르게 하면서
가격 대비 성능도 적절히 유지하는 방법이 계층적 구조다.

01 기억 장치 시스템의 개요

❖ 기억 장치의 계층 구조

❶레지스터	• 액세스 속도가 가장 빠르지만 비트당 가격도 가장 높아 CPU에 소량 존재한다. • RISC 계열의 CPU는 레지스터가 200개 이상이지만 보통은 수십 개 정도다.
❷캐시	• 캐시는 주기억 장치와 CPU의 속도 차를 줄이기 위해 레지스터와 주기억 장치 사이에서 CPU가 자주 사용하는 명령어나 데이터를 일시 저장하는 버퍼 역할을 한다. • 캐시는 SRAM으로 구성되는데, CPU 내부에 있으면 L1 캐시, 외부에 있으면 L2 캐시라고 하는데 요즘은 L2 캐시도 CPU 내부에 집적한다.
❸ 주기억 장치	• DRAM으로 구성했다가 요즘은 SDR SDRAM이나 DDR SDRAM으로 구성한다. • 프로그램을 수행하려면 먼저 해당 프로그램의 명령어와 데이터를 주기억 장치에 적재해야 한다.
❹외부 기억 장치	• 대규모 데이터를 영구 저장하기 위해 사용된다. CPU가 직접 액세스할 수 없고 제어기를 통해 액세스할 수 있다. • 보조 기억 장치라고도 한다. • 고속 액세스가 가능한 것은 자기 디스크이며, 요즘에는 플래시 메모리도 보편적으로 사용된다

2 반도체 기억 장치

- 반도체 메모리는 다양한 관점으로 분류할 수 있으나 대표적으로 쓰기 기능, 휘발성/비휘발성, 재사용 여부, 기억 방식 등에 따라 분류한다.
- 읽기와 쓰기를 모두 수행할 수 있는 메모리를 RWM(Read and Write Memory), 읽기만 가능한 메모리를 ROM이라고 한다. 일반적으로 RAM은 RWM 메모리를 가리킨다.

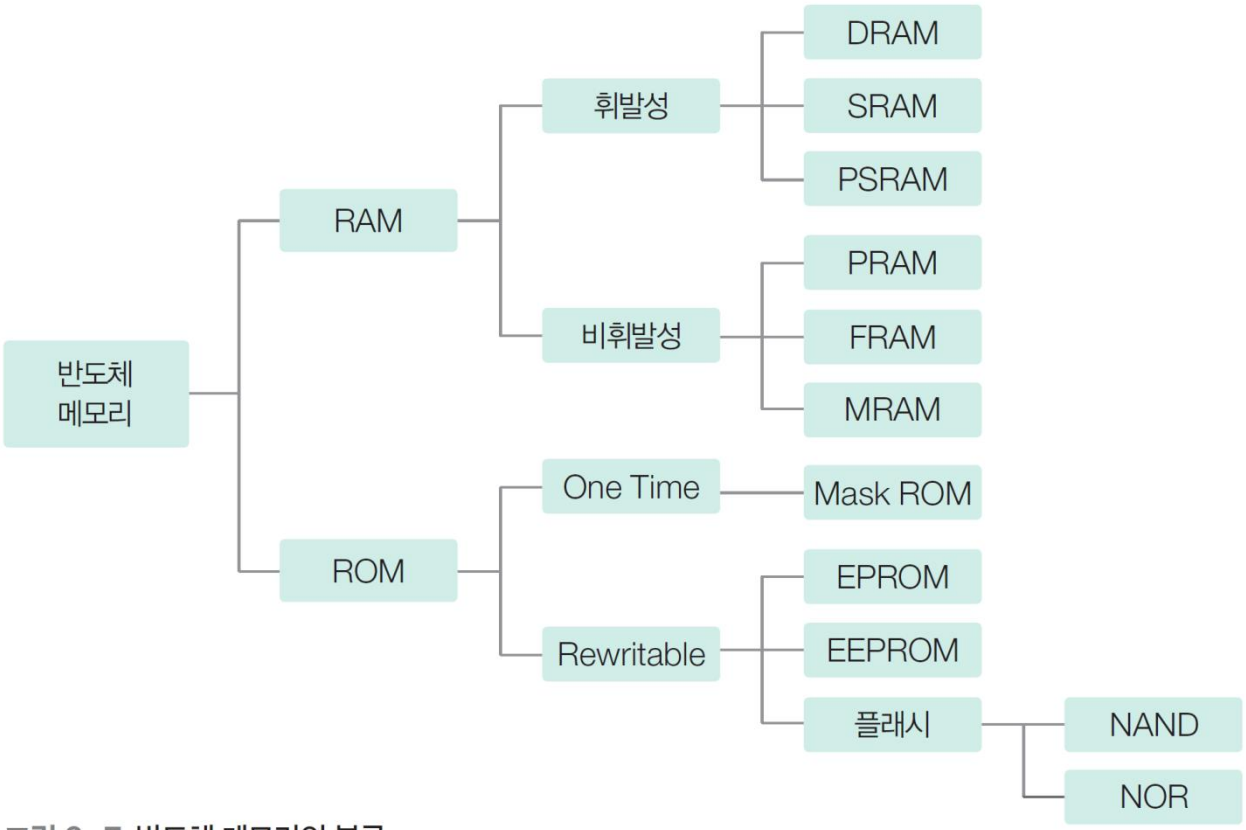


그림 6-7 반도체 메모리의 분류

❑ ROM(Read Only Memory)

- 저장된 데이터를 읽을 수는 있으나, 별도의 장치 없이는 변경할 수 없다.
- 변경할 필요가 없는 프로그램이나 데이터를 저장하는 데 사용된다.
- 컴퓨터 시스템에서는 RAM과 함께 주기억 장치의 일부분으로 ROM을 사용하고 있다.

  - 시스템 초기화 프로그램 및 진단 프로그램
  - 빈번히 사용되는 함수 및 서브루틴
  - 제어 장치의 마이크로 프로그램

02 주기억 장치

❖ ROM의 종류

마스크 ROM (mask ROM)	• 제조 과정에서 데이터를 영구적으로 저장하며, 저장된 것은 절대 변경할 수 없다. • 동일한 형태가 대량으로 필요할 때는Mask ROM이 경제적이다.
PROM (programmable ROM)	• 사용자가 ROM 라이터를 이용하여 프로그램을 할 수 있다. • 일단 프로그램을 하면 퓨즈의 연결 형태가 그대로 유지되며, 변경할 수 없다.
EPROM (erasable PROM)	• 퓨즈가 절단되어도 모든 퓨즈들이 절단되지 않은 초기 상태로 복원할 수 있는 ROM이다. • 복원하는 과정은 일정 시간 자외선을 쪼이면 된다.
EEPROM (electrically EPROM)	• EPROM과 같으나, 복원 과정에서 자외선 대신에 전기 신호를 사용하여 지우는 PROM이다.

EPROM(UVEPROM)

❑ RAM(random access memory)

- RAM은 휘발성이어서 사용하려면 전원을 계속 공급해야 하므로 일시적인 저장 장치로만 활용된다.
- RAM은 데이터의 읽기와 쓰기가 모두 가능하다.
- 임의 액세스 방식을 사용해 CPU가 지정하는 주소에 있는 정보를 직접 액세스할 수 있어 메모리의 위치에 관계없이 액세스 시간이 동일하다.
- **SRAM**은 플립플롭을 사용해 정보를 저장하지만, **DRAM**은 커패시터에 전하를 충전하는 방식으로 정보를 저장한다.

❖ SRAM과 DRAM의 특징 비교

표 6-4 SRAM과 DRAM의 특징 비교

구분	SRAM	DRAM
구성 소자	플립플롭	커패시터
집적도	낮음	높음
전력 소모	많음	적음
동작 속도	빠름	느림
가격	고가	저가
재충전 여부	필요 없음	필요함
용도	캐시	주기억 장치

03 캐시 기억 장치

❖ 캐시의 사용 목적

- CPU와 주기억 장치의 속도 차이로 인한 CPU 대기 시간을 최소화 시키기 위하여 CPU와 주기억 장치 사이에 설치하는 고속 반도체 기억장치

❖ 캐시의 특징

- 주기억 장치보다 액세스 속도가 높은 칩 사용
- 가격 및 제한된 공간 때문에 용량이 적다.

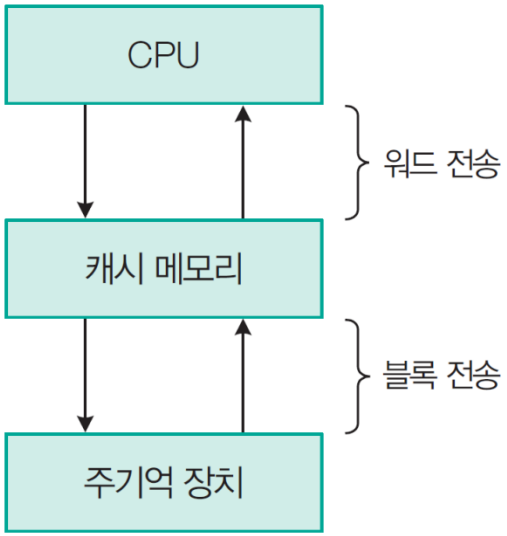


그림 6-20 캐시 위치

❖ 캐시의 기본적인 동작 흐름도

- CPU가 기억 장치에서 어떤 정보(명령어 또는 데이터)를 읽으려는 경우 먼저 해당 정보가 캐시에 있는지 검사한다.
- 있다면 해당 정보를 즉시 읽어 들이고, 없다면 해당 정보를 주기억 장치에서 캐시로 적재한 후 읽어 들인다.

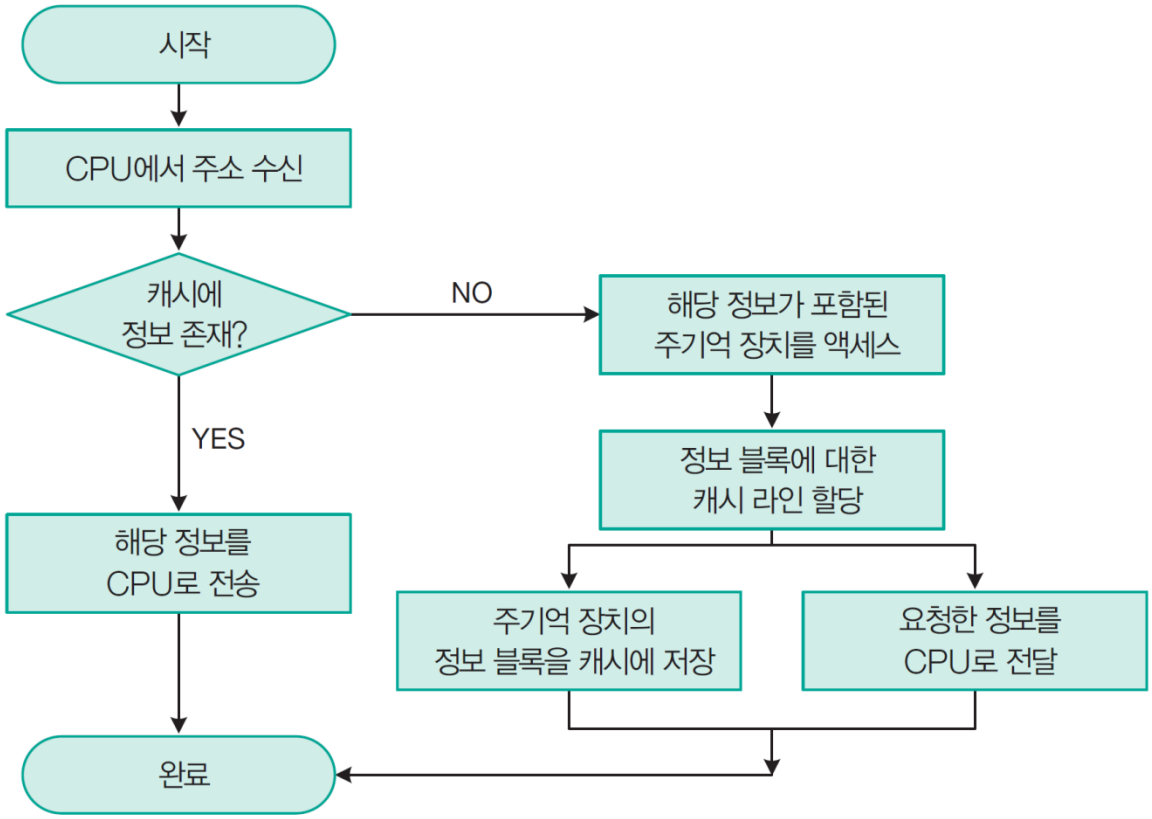


그림 6-21 캐시 읽기 동작 순서

03 캐시 기억 장치

❖ 캐시의 히트율 및 평균 액세스 시간

- 캐시 히트(cache hit) : CPU가 원하는 데이터가 캐시에 있는 상태
- 캐시 미스(cache miss) : CPU가 원하는 데이터가 캐시에 없는 상태. 이 경우에는 주기억 장치로부터 데이터를 읽어온다.
- 히트률(hit ratio) : 캐시에 히트되는 정도(H)

$$H = \frac{\text{캐시에 히트되는 횟수}}{\text{캐시 메모리 액세스 횟수}}$$

- 캐시의 미스율(miss ratio) = $(1 - H)$

❖ 참조 지역성(locality of reference)

- **공간적 지역성**(spatial locality) : 기억장치 내에 인접하여 저장되어 있는 데이터들이 연속적으로 액세스될 가능성이 높다. 예를 들어 표나 배열의 데이터들이 저장되어 있는 기억 장치 영역은 관련 연산이 수행되는 동안 자주 액세스된다.
- **시간적 지역성**(temporal locality) : 최근에 액세스된 프로그램이나 데이터가 가까운 미래에 다시 액세스될 가능성이 높다. 예를 들어 서브루틴이나 루프 프로그램들은 반복적으로 호출되며, 공통 변수들도 자주 액세스된다.

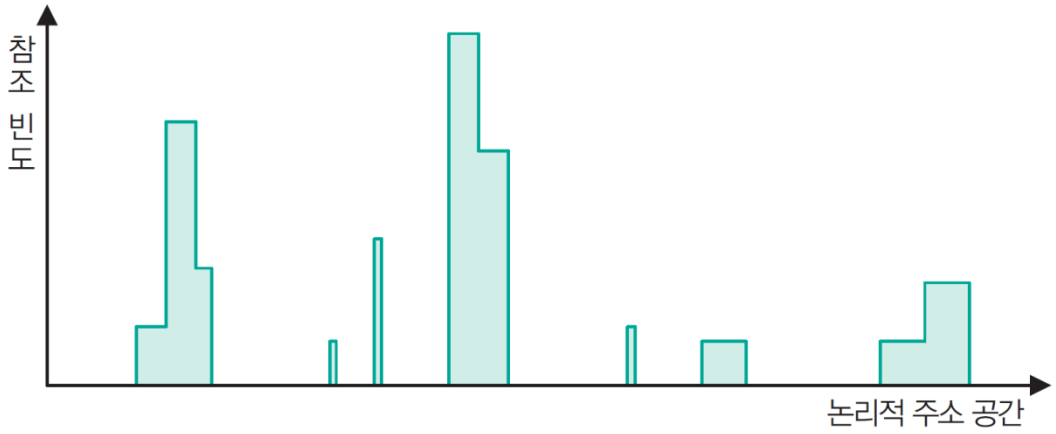


그림 6-22 주소 참조의 전형적인 비균등 분포

03 캐시 기억 장치

1 캐시 용량

- 용량이 커질수록 히트율은 높아지지만, 비용이 증가
- 용량이 커질수록 주소 해독 및 정보 인출을 위한 주변 회로가 복잡해지므로 액세스 시간이 다소 더 길어진다.
- CPU 칩 또는 메인보드의 공간에도 제한을 받는다.

2 쓰기 정책

- 캐시의 블록이 변경되었을 때 그 내용을 주기억 장치에 갱신하는 시기와 방법의 결정

❖ Write-through

- 모든 쓰기 동작들이 캐시로 뿐만 아니라 주기억 장치로도 동시에 수행되는 방식

❖ Write-back

- 캐시에서 데이터가 변경되어도 주기억 장치에는 갱신되지 않는 방식

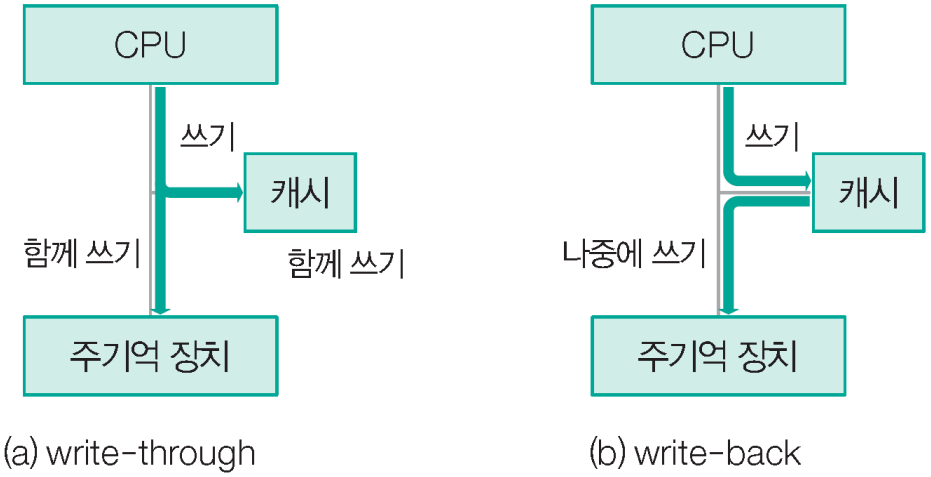


그림 6-34 쓰기 정책

❖ 각 방식의 장단점

Write-through

장점	• 캐시에 적재된 블록의 내용과 주기억 장치에 있는 그 블록의 내용이 항상 같다.
단점	• 모든 쓰기 동작이 주기억 장치 쓰기를 포함하므로, 쓰기 시간이 길어진다.

Write-back

장점	• 기억장치에 대한 쓰기 동작의 횟수가 최소화되고, 쓰기 시간이 짧아진다.
단점	• 캐시의 내용과 주기억 장치의 해당 내용이 서로 다르다.

⇨ 블록을 교체할 때는 캐시의 상태를 확인하여 주기억 장치에 갱신하는 동작이 선행되어야 하며, 그를 위하여 각 캐시 라인이 상태 비트(status bit)를 가지고 있어야 한다.

3 라인 크기

❖ 블록 크기에 따른 특성

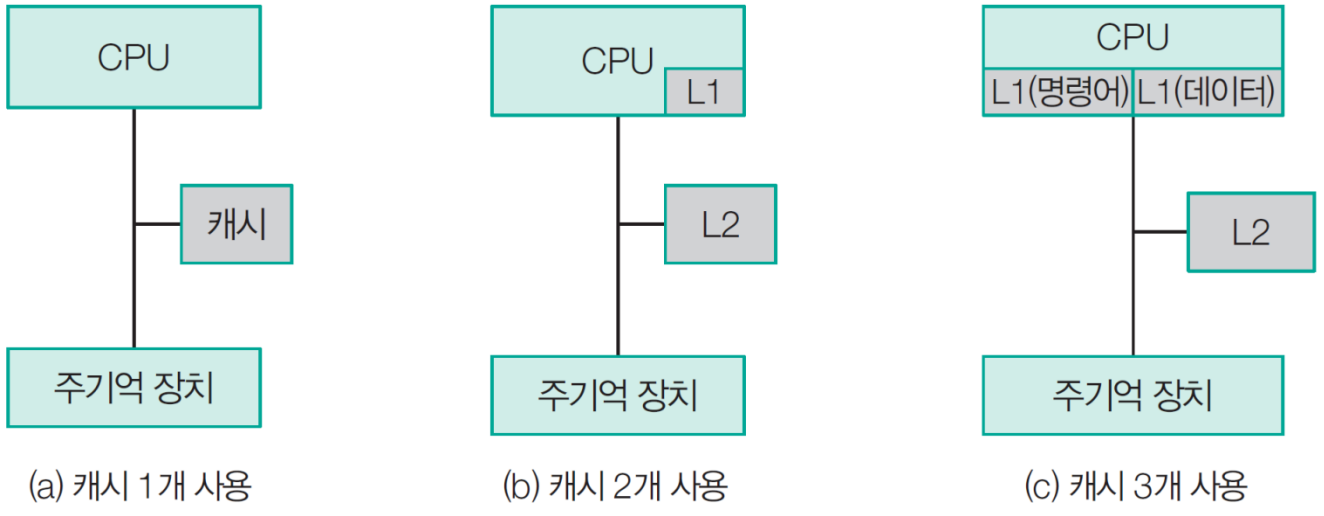
- 캐시 용량은 한정되어 있기 때문에 블록 크기가 커질수록 캐시에 들어올 수 있는 블록의 수는 줄어든다. 새로운 블록을 읽어 올 때마다 원래 캐시 내용은 교체되기 때문에 블록 수가 적으면 인출된 직후에 다른 블록과 다시 교체된다.
- 블록 크기가 커질수록 원하는 워드에서 멀리 떨어져 있는 워드들도 같이 읽혀 오며, 그들은 가까운 미래에 사용될 가능성은 낮다.

⇨ 대략적으로 블록 크기가 8~32바이트 정도가 최적에 가까운 것으로 알려져 있다.

4 캐시 수

❑ 계층적 캐시

- **온-칩 캐시**(on-chip cache) : 캐시 액세스 시간을 단축시키기 위하여 CPU 칩 내에 포함시킨 캐시 (그림의 L1)
- **계층적 캐시**(hierarchical cache) : 온-칩 캐시를 1차(L1) 캐시로 사용하고, CPU 외부에 더 큰 용량의 2차(L2) 캐시를 설치하는 방식
- **분리 캐시**(split cache) : 캐시를 명령어 캐시와 데이터 캐시로 분리



(a) 캐시 1개 사용 (b) 캐시 2개 사용 (c) 캐시 3개 사용

그림 6-35 다양한 계층적 캐시 구조

❑ 통합 캐시와 분리 캐시

❖ 통합 캐시(unified cache)

- 온-칩 캐시가 처음 출현했을 때, 대부분은 명령어와 데이터를 하나의 캐시에 저장

❖ 분리 캐시(split cache)

- 캐시를 명령어 캐시와 데이터 캐시로 분리
- 명령어 인출 유니트와 실행 유니트 간의 캐시 액세스 충돌 제거

04 가상 기억 장치

❖ **가상 기억 장치**(virtual memory)

- 보조기억장치의 일부 용량을 주기억장치처럼 가상하여 사용할 수 있도록 하는 기법이다.
- 가상기억장치의 가장 큰 목적은 주기억장치의 용량(주소 공간)의 확대이다.
- 주기억장치의 이용률과 다중 프로그래밍의 효율을 높일 수 있다.
- 사용자가 프로그램 크기에 제한 받지 않고 실행이 가능하다.

❖ 캐시 메모리, 주기억 장치, 가상 기억 장치 간의 데이터 이동

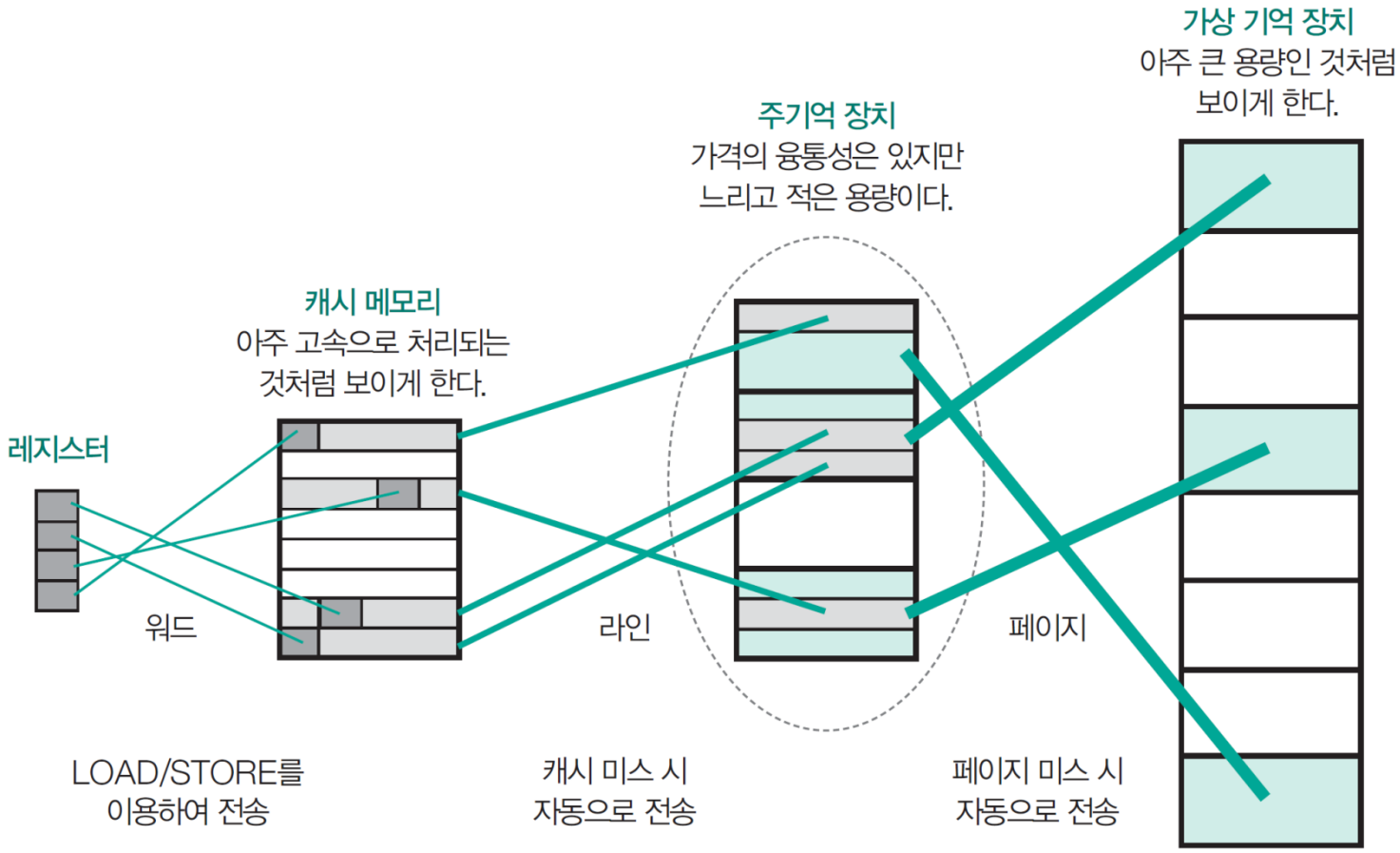


그림 6-43 메모리 계층 구조에서의 데이터 이동

05 연관 기억 장치

- **연관 기억 장치**(associative memory)는 메모리에 저장된 내용의 일부분을 이용해 원하는 정보가 저
- 장된 위치를 알아낸다.
- CAM(Content Addressable Memory) 또는 병렬 탐색 기억 장치라고도 한다.
- 기억 장치의 모든 단자를 동시에 읽어 주어진 특성과 비교하므로 주소로만 접근할 때보다 훨씬 빠르다.
- 하지만 병렬 판독 회로 추가로 비싸져 탐색 시간이 중요하고 빠른 처리에만 주로 사용된다.

Thank You