

06

캐시 기억장치

# 학습 내용

- 01 컴퓨터 기억장치의 계층적 구조
- 02 캐시 기억장치 원리
- 03 캐시 기억장치의 설계
- 04 교체 알고리즘

## 학습 목표

- 기억장치들간의 상호 연관성에 대해 설명할 수 있다.
- 캐시 기억장치의 원리에 대해서 설명할 수 있다.
- 캐시 기억장치와 주기억장치 사이의 정보교환 방법에 대해 설명할 수 있다.
- 캐시 기억장치의 교체 알고리즘에 대해 설명할 수 있다.

지난시간  
돌아보기

- 5주차.  
컴퓨터 시스템의 구성과 기능 -

지/난/시/간/의/ 학/습/내/용

컴퓨터의 구성요소

컴퓨터의 기능

버스와 상호 연결

## 지난시간 돌아보기

### 컴퓨터의 구성요소

#### ✓ 컴퓨터 구조

- 명령어 세트, 데이터 표현에 사용되는 비트의 수, 입출력 메커니즘, 주소 지정 방식 등과 특성
- 컴퓨터 조직 : 컴퓨터 구조의 특성들을 구현하는 방법

#### ✓ 구성

- 하드웨어 : 컴퓨터 정보들의 전송 통로 제공, 정보의 실제 처리가 일어나게 하는 물리적 실체 (중앙처리장치, 기억장치, 입출력장치 등)
- 소프트웨어 :  
시스템 소프트웨어, 응용 소프트웨어로 분류

## 지난시간 돌아보기

### 컴퓨터의 기능

#### ✓ 명령어의 수행 과정

- 명령어 인출, 명령어 해독, 명령어 실행, 프로그램 카운터 증가 순으로 이루어짐

#### ✓ CPU 내의 레지스터

- 프로그램 카운터, 명령어 레지스터, 기억장치 주소 레지스터, 기억장치 버퍼 레지스터, 입출력 주소 레지스터, 입출력 버퍼 레지스터 등이 있음

## 지난시간 돌아보기

### 버스와 상호 연결

#### ✓ 버스

- 컴퓨터에서 두 개 혹은 그 이상의 장치들을 연결하는 공유 전송 매체

#### ✓ 버스의 분류

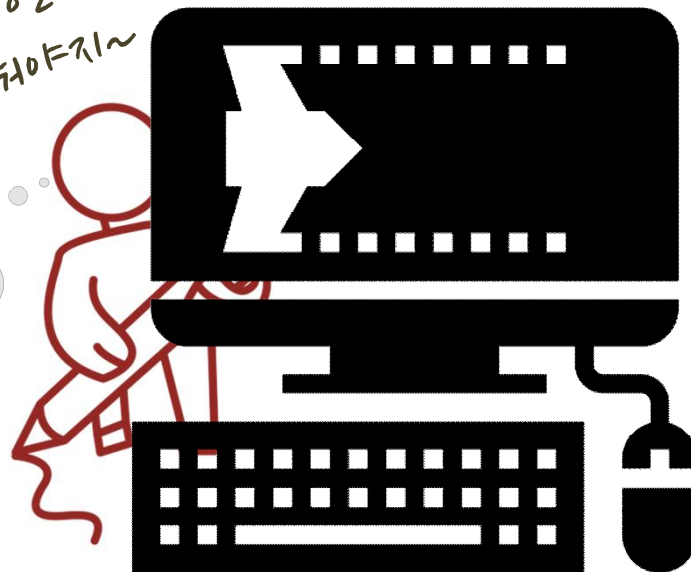
- 시스템 버스 : 프로세서, 기억장치 및 I/O 장치간의 통신을 위한 상호 연결
- 데이터 버스 : 모듈들 사이의 데이터 전송 통로
- 제어 버스 : 데이터 버스와 주소 버스의 사용을 제어하는 신호들을 전송하는 통로

## 생각 해보기

자주 사용되는 명령들을  
미리 기억해 두면 어떤 장점이 있을까요?

PC사용자가 이 명령들을 자주 쓰니까,  
기억해 두어야지~

명령어







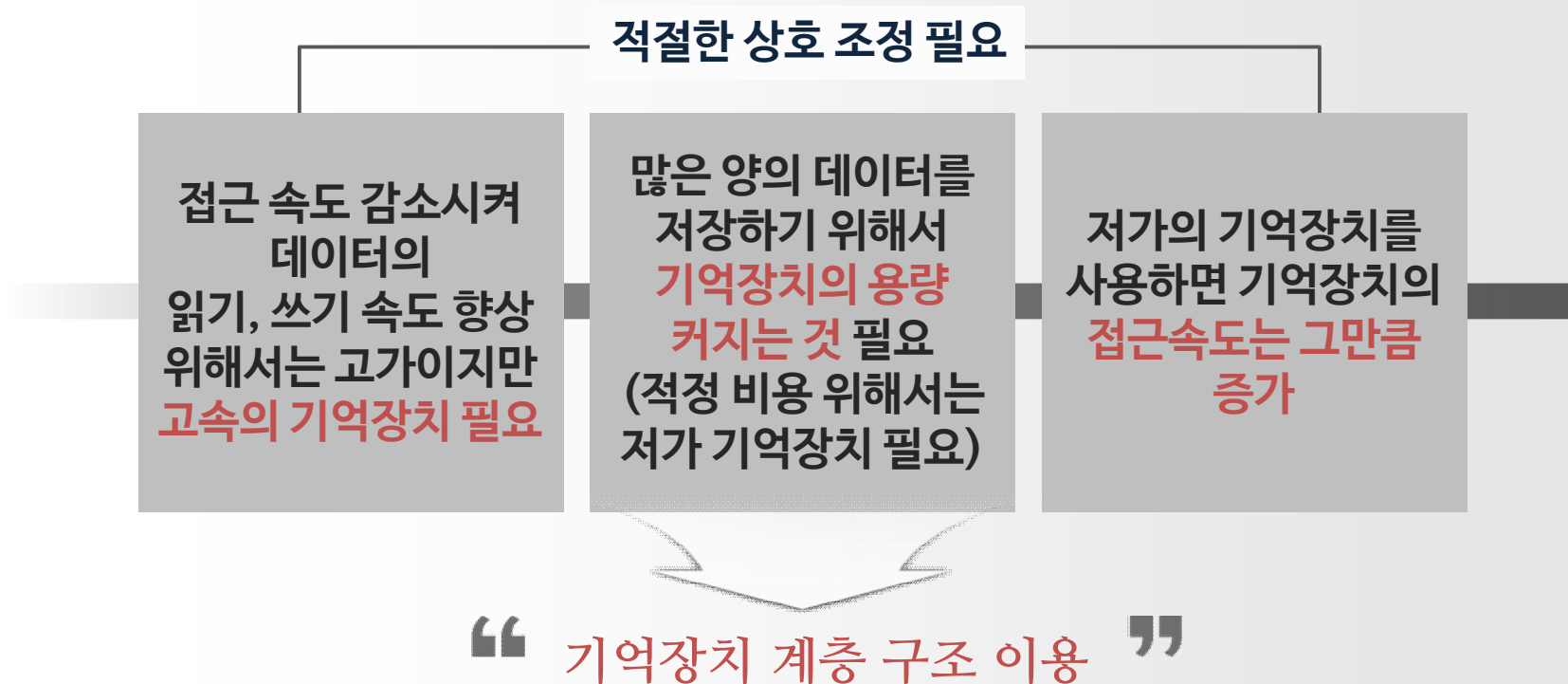
01

# 컴퓨터 기억장치의 계층적 구조

- 1) 기억장치 계층구조
- 2) 동작 방법

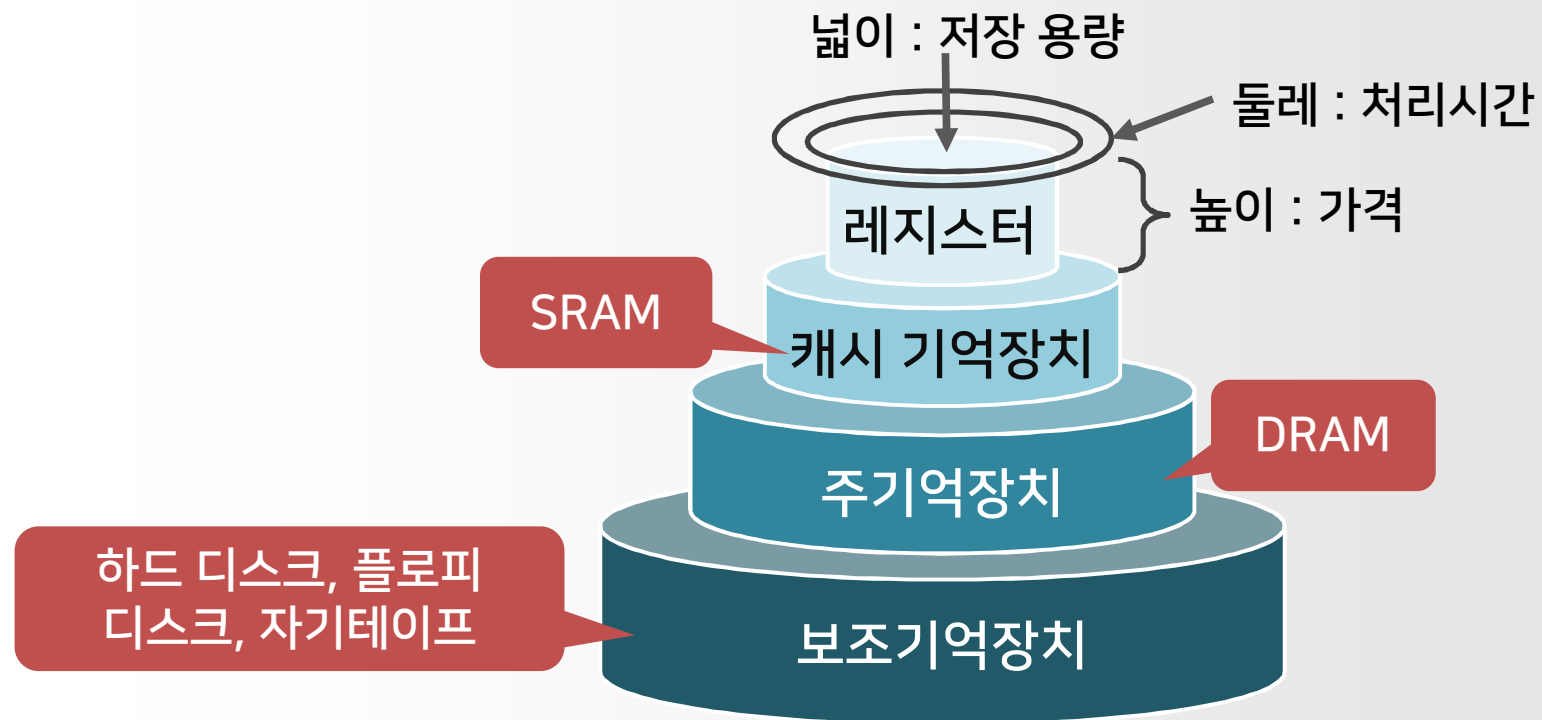
## 1) 기억장치 계층구조

- 컴퓨터의 기억장치의 용량, 접근 속도, 가격의 상관관계



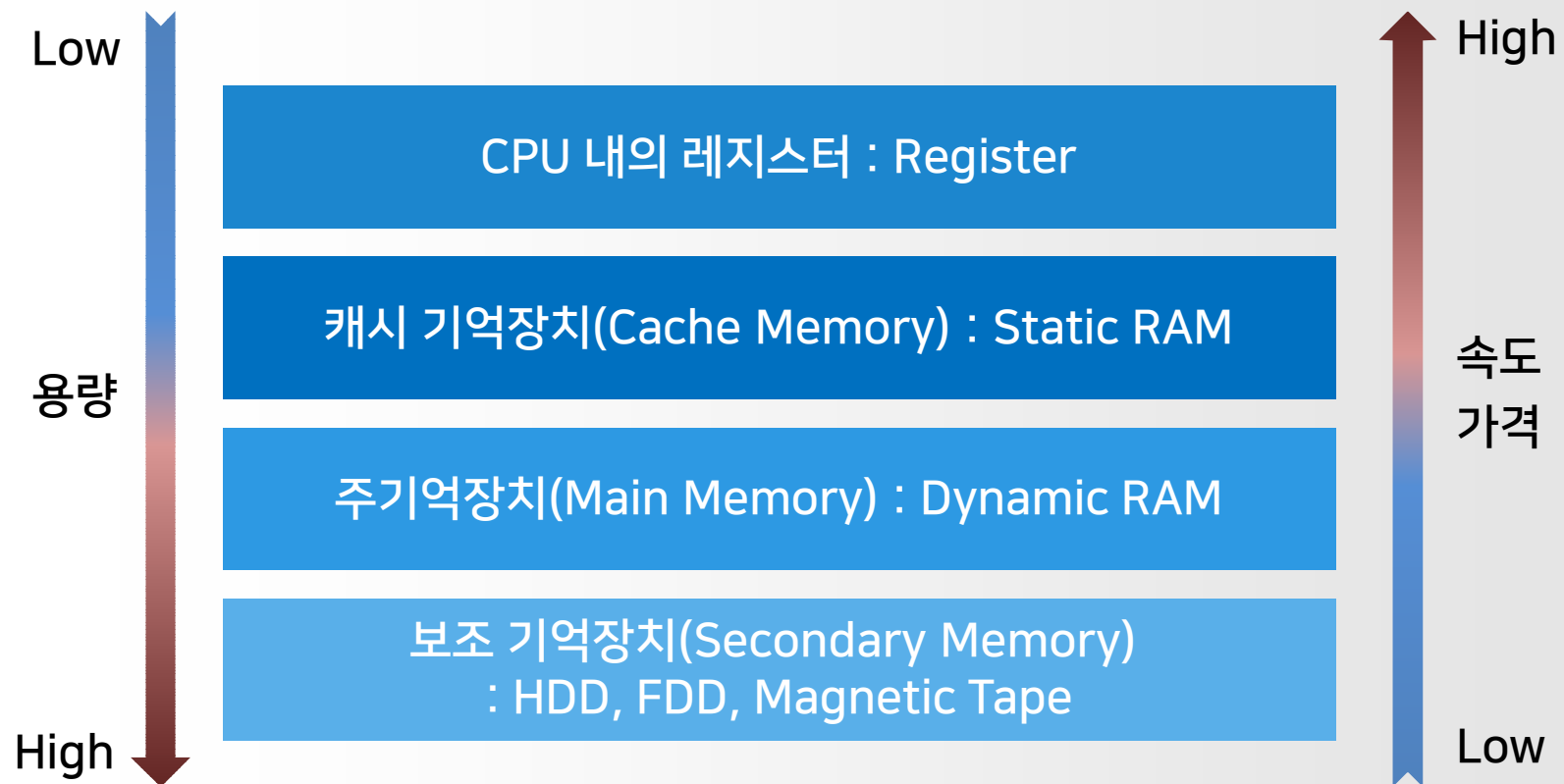
# 1) 기억장치 계층구조

- 기억장치 계층구조



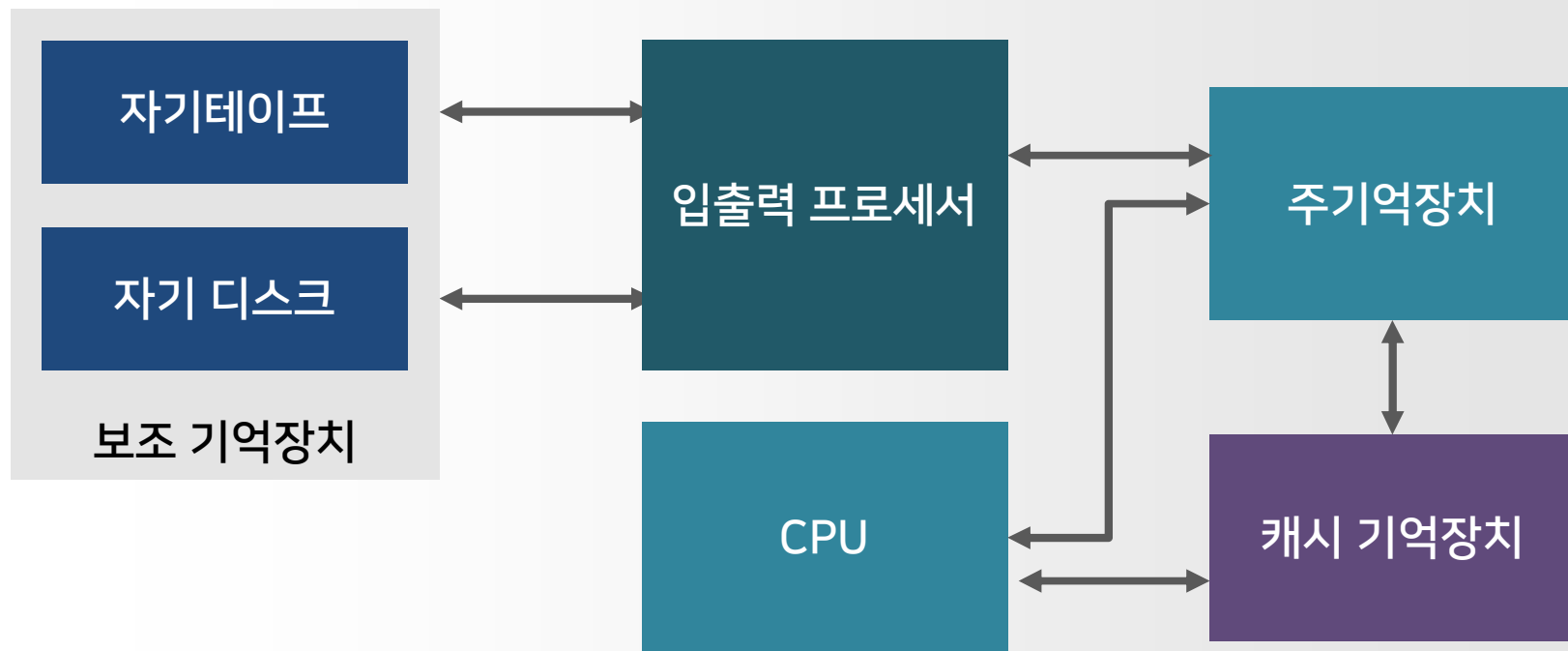
# 1) 기억장치 계층구조

## ■ 기억장치 계층구조



## 2) 동작 방법

- 동작 방법





02

## 캐시 기억장치의 원리

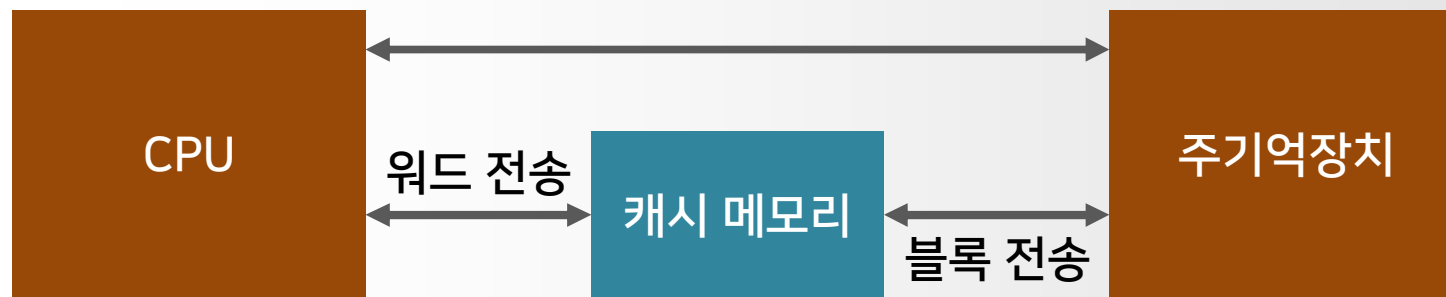
- 1) 동작 원리
- 2) 동작 순서
- 3) 참조의 지역성

- 4) 동작 과정
- 5) 적중률

## 1) 동작 원리

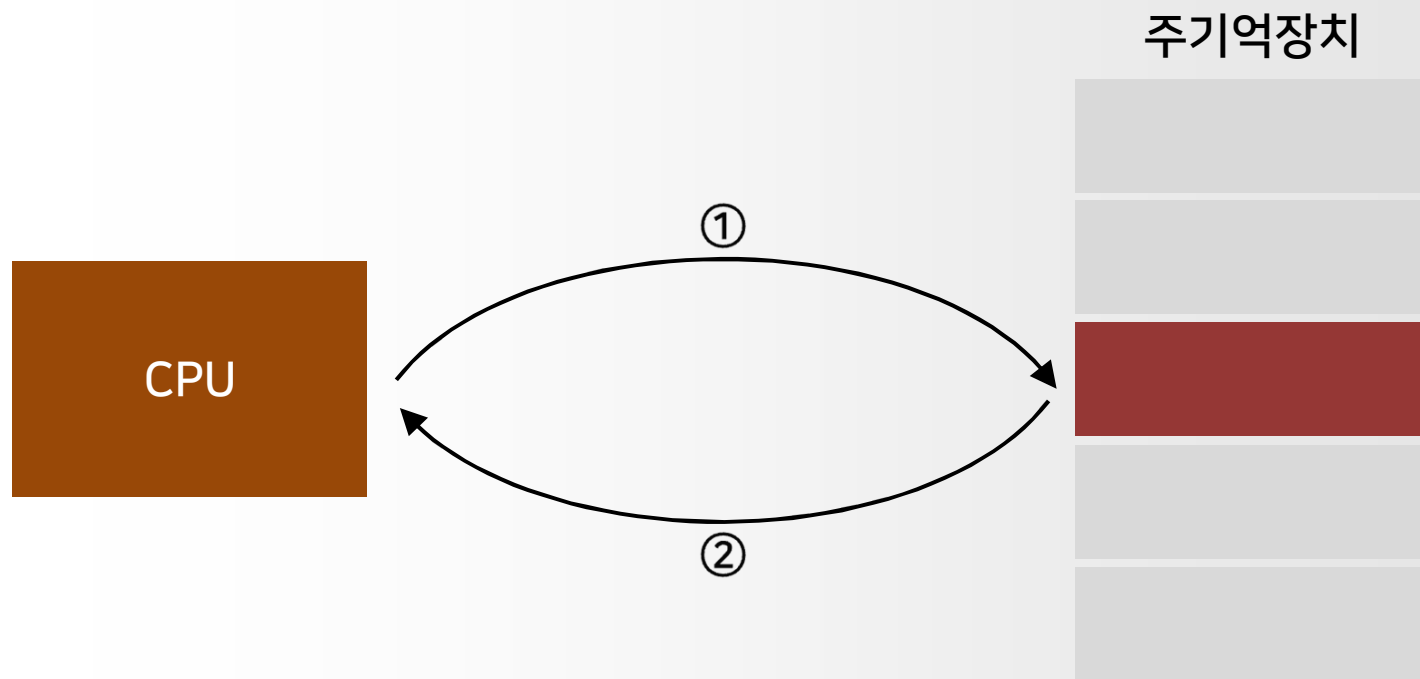
### ■ 캐시 기억장치

- ▶ 빠른 접근 시간을 제공하는 기억장치
- ▶ 수행할 명령어나 오퍼랜드를 주기억장치로부터 가져와 저장하고 있다가 빠른 속도로 중앙처리장치에 제공



## 1) 동작 원리

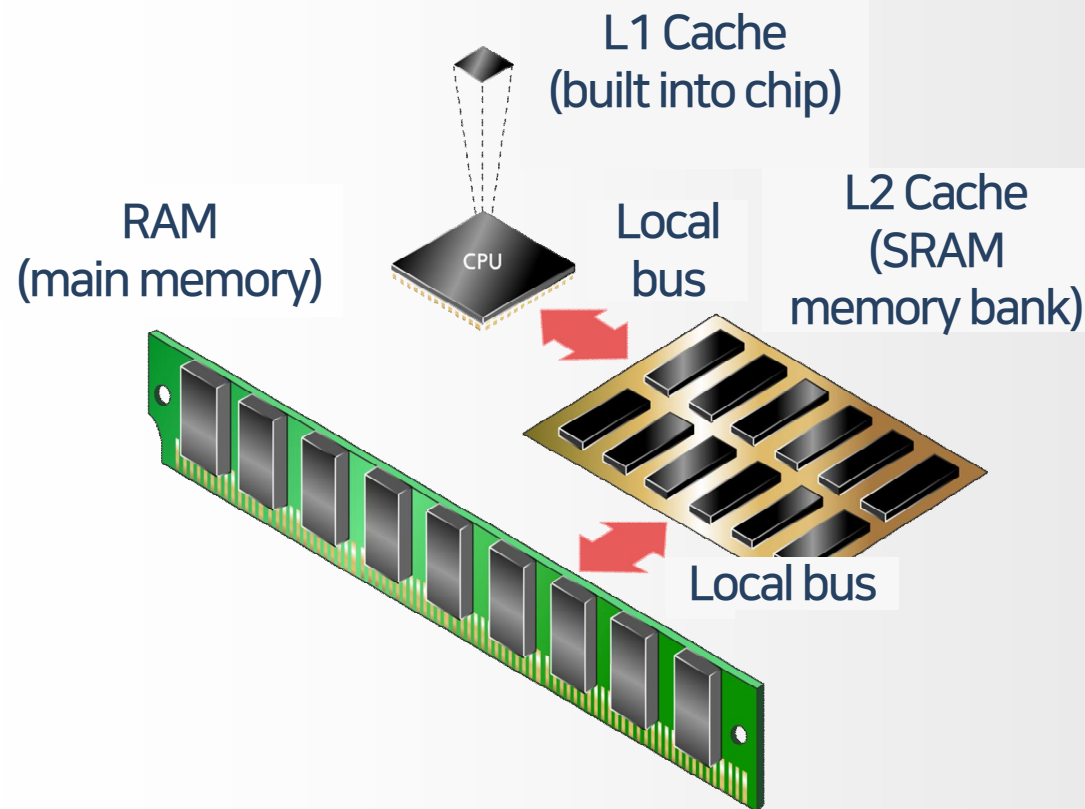
- 캐시 기억장치의 동작 원리





# 1) 동작 원리

- 캐시 메모리의 계층화



## 2) 동작 순서

### ■ 캐시 기억장치가 있는 시스템

- ▶ 중앙처리장치가 기억장치를 참조할 필요가 있을 경우  
캐시 기억장치를 먼저 조사

#### 적중 (Hit)

- 캐시 기억장치에 접근하여  
그 내용을 찾았을 때

#### 실패 (Miss)

- 캐시 기억장치에 접근하여  
그 내용을 찾지 못하였을 때

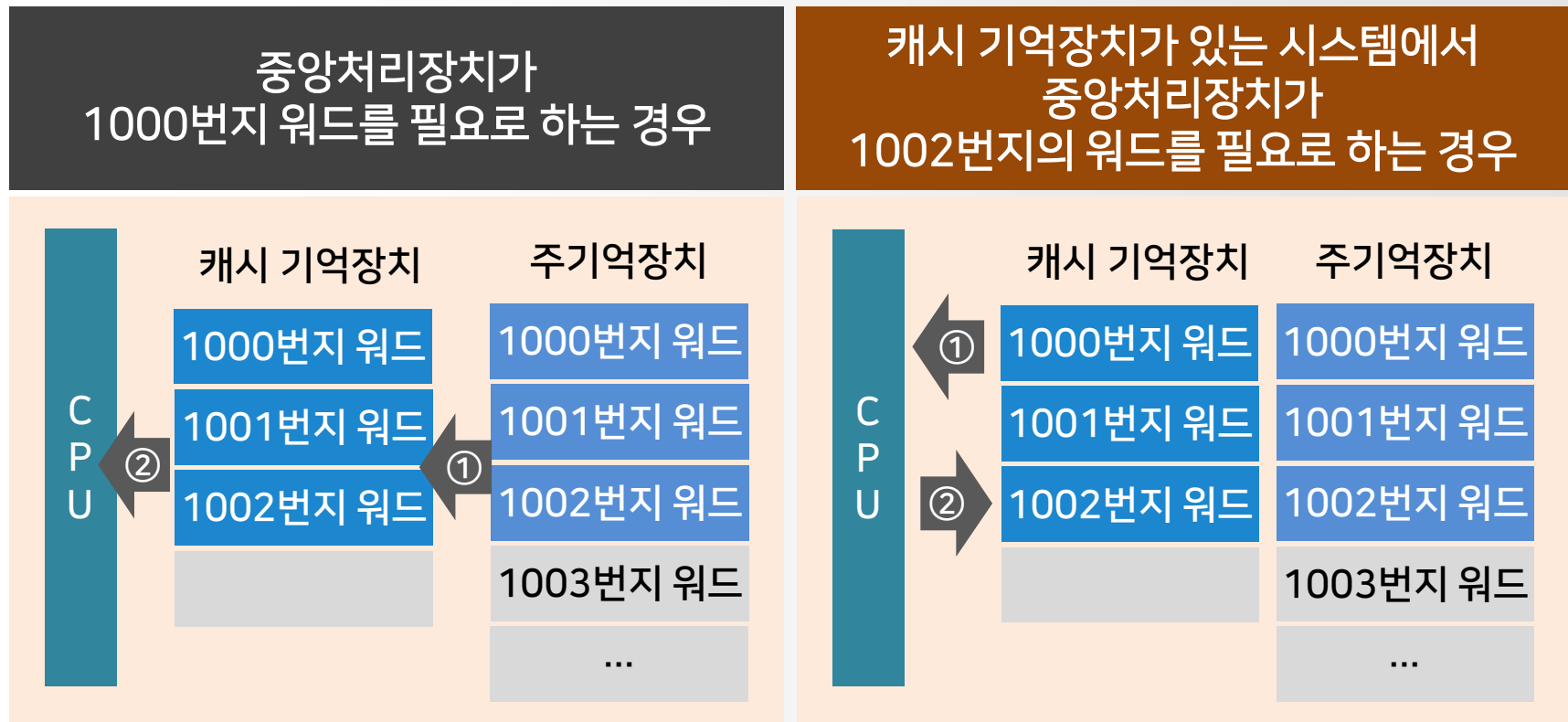
## 2) 동작 순서

- 캐시 기억장치가 있는 시스템



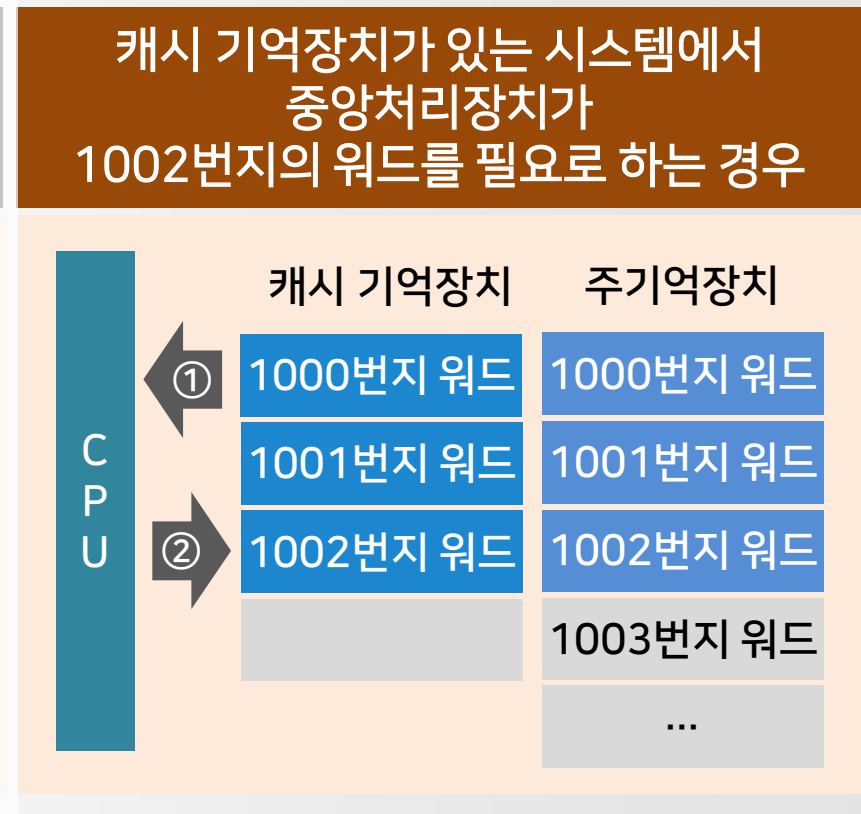
## 2) 동작 순서

- 캐시 기억장치가 있는 시스템



## 2) 동작 순서

## ■ 캐시 기억장치가 있는 시스템



### 3) 참조의 지역성

- 참조의 지역성(locality of reference)

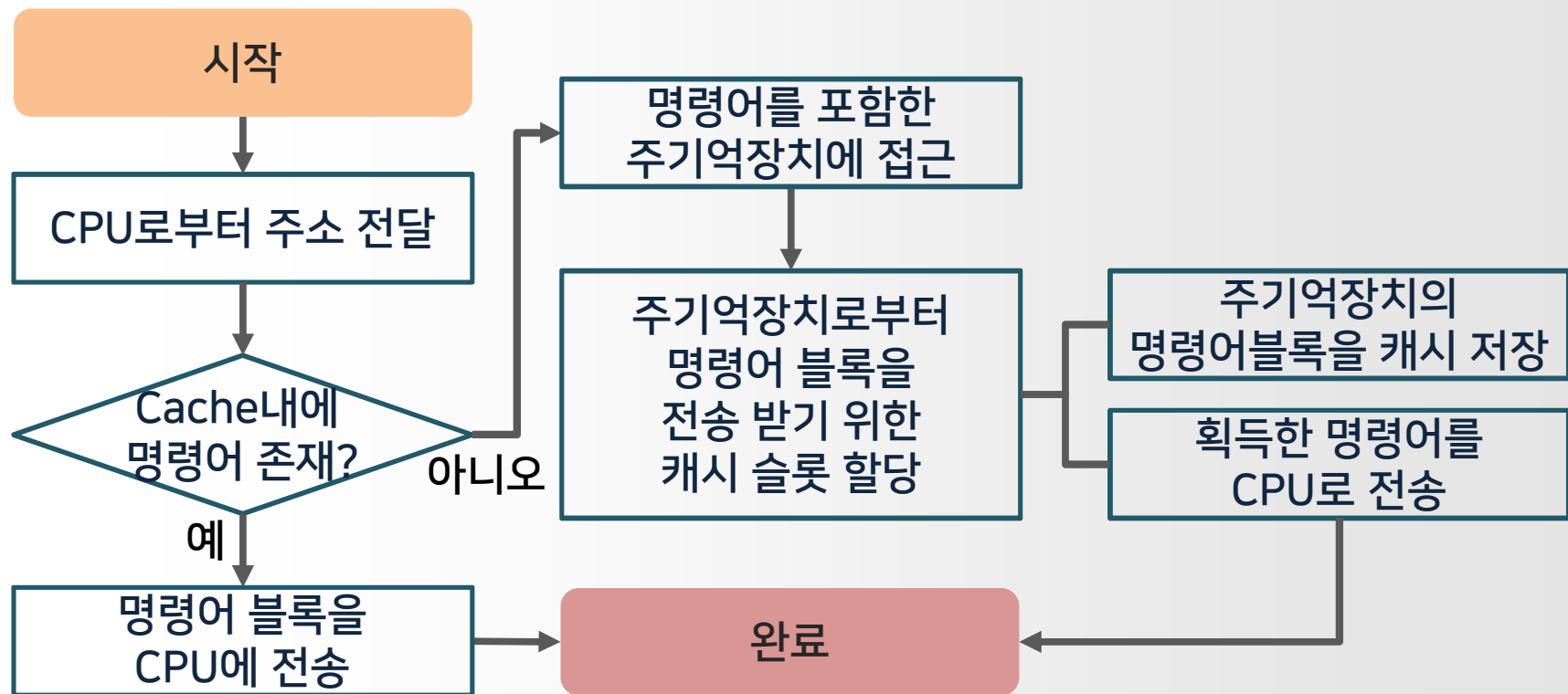
#### 캐시

- 중앙처리장치가 수행할 명령어와 필요한 정보를 저장하고 있다가 즉시 제공함으로써 처리가 신속하게 이루어지도록 함
- 동작 : 프로그램 내장형 컴퓨터의 특성인  
기억장치 참조의 지역성(Locality of reference)으로 인해 가능

- ▶ 주어진 시간 동안 중앙처리장치의 기억장치 참조는 제한된 영역에서만 이루어지는 현상
- ▶ 짧은 시간 동안 중앙처리장치가 접근하는 범위는 지역적으로 제한되는 것을 의미

## 4) 동작 과정

### ■ 캐시 기억장치의 동작 과정



## 5) 적중률

### ■ 적중률 (Hit ratio)

#### ▶ 캐시 기억장치 성능 측정

$$\text{적중률} = \frac{\text{적중수}}{\text{전체 메모리 참조 횟수}}$$

#### • 주기억장치와 캐시 기억장치 사이에서의 평균 기억장치 접근 시간

$$T_{\text{average}} = \overset{\text{적중률}}{H_{\text{hit\_ratio}}} \times \overset{\text{캐시 기억장치 접근시간}}{T_{\text{cache}}} + (1 - H_{\text{hit\_ratio}}) \times \overset{\text{주기억장치 접근시간}}{T_{\text{main}}}$$

평균 기억장치 접근시간

→ 캐시의 적중률이 높아질수록 평균 기억장치 접근 시간은 캐시 액세스시간에 접근



## 예제

$T_{\text{cache}} = 50\text{ns}$ ,  $T_{\text{main}} = 400\text{ns}$ 일 때,  
적중률을 증가시키면서 기억장치 접근시간( $T_{\text{average}}$ )을 계산해 봅시다.

- 적중률 70%의 경우  $T_{\text{average}} = 0.7 \times 50\text{ns} + 0.3 \times 400\text{ns} = 155\text{ns}$
- 적중률 80%의 경우  $T_{\text{average}} = 0.8 \times 50\text{ns} + 0.2 \times 400\text{ns} = 120\text{ns}$
- 적중률 90%의 경우  $T_{\text{average}} = 0.9 \times 50\text{ns} + 0.1 \times 400\text{ns} = 85\text{ns}$
- 적중률 95%의 경우  $T_{\text{average}} = 0.95 \times 50\text{ns} + 0.05 \times 400\text{ns} = 67.5\text{ns}$
- 적중률 99%의 경우  $T_{\text{average}} = 0.99 \times 50\text{ns} + 0.01 \times 400\text{ns} = 53.5\text{ns}$



캐시 기억장치의 적중률이 높아질수록  
평균 기억장치 접근시간은 캐시 기억장치 접근시간에 근접하게 되어  
컴퓨터의 처리 속도의 성능 향상을 가져옴!



03

## 캐시 기억장치의 설계

- 1) 설계 목표 및 고려 사항
- 2) 캐시 기억장치의 크기와 인출 방식
- 3) 사상함수

## 1) 설계 목표 및 고려 사항

- 캐시 기억장치를 설계함에 있어 공통적인 목표

- ▶ 캐시 적중 시  
캐시 기억장치로부터 데이터를 읽어오는 시간을 짧게 함
- ▶ 캐시 실패 시 주기억장치로부터 캐시 기억장치로  
데이터를 읽어 오는 시간을 최소화함
- ▶ 주기억장치와 캐시 기억장치 사이에  
데이터의 일관성을 유지함

## 1) 설계 목표 및 고려 사항

### ■ 캐시 기억장치를 설계할 때 고려 사항

고려  
사항

캐시 기억장치의 크기(Size)

인출방식(Fetch algorithm)

사상함수(Mapping function)

교체 알고리즘(Replacement algorithm)

쓰기 정책(Write policy)

블록 크기(Block size)

캐시 기억장치의 수(Number of caches)

## 2) 캐시 기억장치의 인출 방식

### ■ 인출 방식

#### 요구 인출(demand fetch) 방식

- 현재 필요한 정보만 주기억장치로부터 블록 단위로 인출해 오는 방식
- 매번 주기억장치에서 인출을 수행하므로 경우에 따라 실패율이 높아져서 캐시 기억장치의 효과를 얻지 못할 때도 있음

#### 선인출(prefetch) 방식

- 현재 필요한 정보 외에도 앞으로 필요할 것으로 예측되는 정보도 미리 인출하는 방식
- 지역성(locality)이 높은 경우에 효과가 높지만 그렇지 못한 경우에는 인출 시간이 길어지기 때문에 효율이 떨어짐

### 3) 사상함수

- 사상함수란?

▶ 주기억장치와 캐시 기억장치 사이에서 정보를 옮기는 것

사상함수

직접사상 (direct mapping)

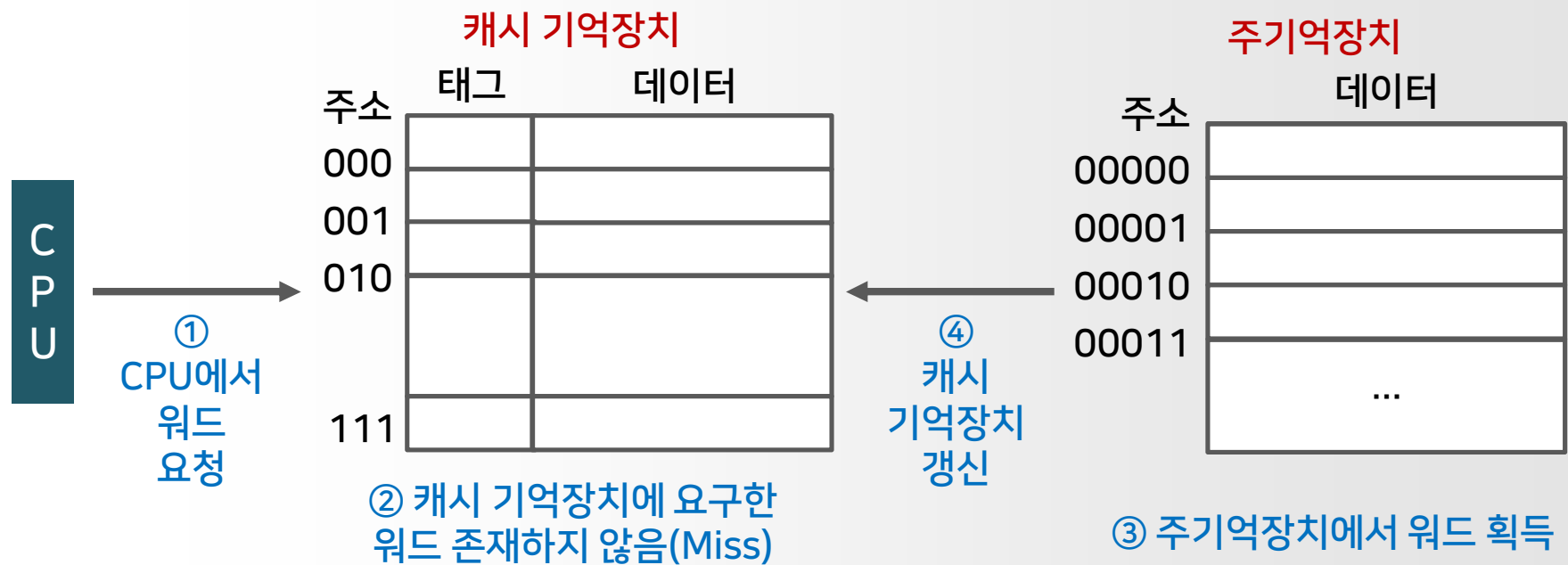
연관사상 (associative mapping)

집합연관사상 (set-associative mapping)

### 3) 사상함수

- 사상함수란?

▶ 주기억장치와 캐시 기억장치 사이에서 정보를 옮기는 것



### 3) 사상함수

#### ▪ 사상함수란?

직접사상

연관사상

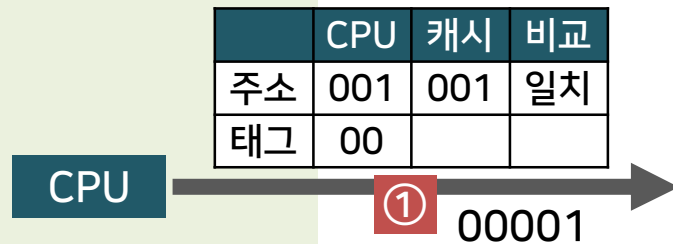
집합연관사상

- 주기억장치의 블록이 특정 라인에만 적재
- 캐시의 적중 여부는 그 블록이 적재 될 수 있는 라인만 검사
- 장점 : 사상 과정이 간단하고 작은 용량의 RAM을 캐시 기억장치로 사용하기 때문에 비용이 저렴
- 단점 : 주기억장치의 블록이 적재 될 수 있는 라인이 하나  
→ 프로그램이 동일한 라인에 적재되는 두 블록들을 반복적으로 액세스하는 경우 캐시 실패율이 매우 높아짐



## 직접사상

### 동작 1/ 중앙처리장치가 00001번지 워드를 필요하는 경우



① 처음의 2-bits(00)는 태그, 다음 3-bits(001)는 캐시 기억장치의 주소 표시

- ◆ 캐시 기억장치에서 010번지 주소를 참조
- ◆ 해당 번지가 비어 있으므로 실패 (miss) 주기억장치를 참조

#### 캐시 기억장치

주소	태그	데이터
000		
001		
010		
111		

<실행 전>

주소	태그	데이터
000		
001	00	1234
010		
111		

<실행 후>

#### 주기억장치

주소	데이터
00000	9999
00001	1234
00010	5678
00011	9012
10001	7890
10010	2750
	...
11111	2580

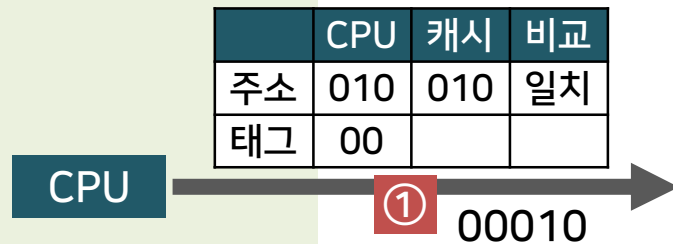
② 주기억장치로부터 워드 획득

	변경 전	변경 후
데이터		1234
태그		00

③ 캐시 기억장치의 해당 주소에 데이터, 태그 저장

## 직접사상

## 동작 2/ 중앙처리장치가 00010번지 워드를 필요로 하는 경우



① 처음의 2-bits(00)는 태그, 다음 3-bits(010)는 캐시 기억장치의 주소 표시

- ◆ 캐시 기억장치에서 001번지 주소를 참조
- ◆ 해당 번지가 비어 있으므로 실패 (miss) 주기억장치를 참조

### 캐시 기억장치

주소	태그	데이터
000		
001	00	1234
010		
111		

<실행 전>

주소	태그	데이터
000		
001	00	1234
010	00	5678
111		

<실행 후>

### 주기억장치

주소	데이터
00000	9999
00001	1234
00010	5678
00011	9012
10001	7890
10010	2750
	...
1111	2580

② 주기억장치로부터 워드 획득

	변경 전	변경 후
데이터		5678
태그		00

③ 캐시 기억장치의 해당 주소에 데이터, 태그 저장

### 3) 사상함수

#### ▪ 사상함수란?

직접사상

연관사상

집합연관사상

- 주기억장치의 블록이 캐시의 어느 라인에든 적재될 수 있어  
직접사상에서 발생하는 단점을 보완
- 적중 검사가 모든 라인에 대해서 이루어져야 하므로 검사 시간이 길어짐
- 캐시 슬롯의 태그를 병렬로 검사하기 위해서는  
매우 복잡하고 비용이 높은 회로가 필요

## 연관사상



### 주기억장치

태그	단어	데이터
0000	00	'abcd'
0001	00	'take'
0010	00	
0011	00	'into'
0100	00	'fell'
0101	00	
0110	00	'down'
0111	00	
1000	00	
1001	00	
1010	00	'comp'
1011	00	
1100	00	'tape'
1101	00	
1110	00	
1111	00	'wake'

### 캐시 기억장치

태그	데이터	슬롯번호
0000	'abcd'	0
0011	'into'	1
0001	'take'	2
0110	'down'	3
0100	'fell'	4
1010	'comp'	5
1100	'tape'	6
1111	'wake'	7



### 3) 사상함수

#### ▪ 사상함수란?

직접사상

연관사상

집합연관사상

- 직접사상과 연관사상 방식을 조합한 방식
- 하나의 주소 영역이 서로 다른 태그를 갖는 여러 개의 집합으로 이루어지는 방식
- 두 개의 집합을 갖는 집합연관 캐시 기억장치의 구조

## 집합연관사상

주소	태그	데이터	태그	데이터
000	00	1 2 2 0	01	5 6 7 8
111	11	4 5 8 0	00	1 2 3 4
집합 1			집합 2	

- 1/ 두 개의 집합으로 구분(집합 1, 집합 2)
- 2/ 같은 주소 번지 000 번지에  
서로 다른 태그 00, 01로 구분되는 두 개의 데이터가 동시에 저장
- 3/ 동일한 주소를 가지고 다른 태그 번호를 갖는 번지에 접근하는 경우

직접사상의 경우 실패

집합연관사상방식의 경우 적중





04

## 교체 알고리즘

---

- 1) 개요
- 2) 페이지 교체 알고리즘

## 1) 개요

### ▪ 교체 알고리즘

▶ 캐시 기억장치가 가득 차 있는 상태에서  
캐시 기억장치의 일부를 제거하고,  
주기억장치로부터 새로운 데이터를 가져와야 하는 경우  
캐시의 내용을 제거하는 방식

- 직접사상방식에서는  
주기억장치의 데이터가 캐시 기억장치의 같은 주소에 저장되기 때문에  
교체 알고리즘을 사용할 필요 없음
- 연관사상 및 집합연관사상방식은 교체 알고리즘이 필요함



## 1) 개요

### ▪ 교체 알고리즘의 종류

최소 최근 사용 알고리즘  
(LRU, Least Recently Used)

- 현재까지 알려진 교체 알고리즘 중에서 가장 효과적
- 캐시 기억장치 내에서 사용되지 않은 채로 가장 오래 있었던 블록을 교체하는 방식

최소 사용 빈도 알고리즘  
(LFU, Least Frequently Used)

- 캐시 기억장치에 적재된 후 가장 적게 사용된 블록을 교체하는 방식

선입력 선출력 알고리즘  
(FIFO, First In First Out)

- 캐시 기억장치에 적재된 지 가장 오래된 블록을 교체하는 방식

랜덤 (Random)

- 캐시 기억장치에서 임의의 블록을 선택하여 교체하는 방식

## 2) 페이지 교체 알고리즘

### ▪ FIFO 기법에 의한 주기억장치 상태 변화 과정

시간	1	2	3	4	5	6	7	8	9	10	11	12	13	14
참조 스트링	1	2	6	1	4	5	1	2	1	4	5	6	4	5
주기억장치 상태	1	1	1	1	1	5	5	5	5	5	5	5	4	4
		2	2	2	2	2	1	1	1	1	1	1	1	5
			6	6	6	6	6	2	2	2	2	2	2	2
					4	4	4	4	4	4	4	6	6	6
페이지 부재 발생 여부	F	F	F		F	F	F	F				F	F	F

→ 페이지 부재의 발생 횟수 : 총 10번 발생

## 2) 페이지 교체 알고리즘

### ▪ LRU 기법에 의한 주기억장치 상태 변화 과정

시간	1	2	3	4	5	6	7	8	9	10	11	12	13	14
참조 스트링	1	2	6	1	4	5	1	2	1	4	5	6	4	5
주기억장치 상태	1	1	1	1	1	1	1	1	1	1	1	1	1	4
		2	2	2	2	5	5	5	5	5	5	5	5	5
			6	6	6	6	6	2	2	2	2	6	6	6
					4	4	4	4	4	4	4	4	4	4
페이지 부재 발생 여부	F	F	F		F	F		F				F		

→ 페이지 부재의 발생 횟수 : 총 7번 발생

## 2) 페이지 교체 알고리즘

### ▪ LFU 기법에 의한 주기억장치 상태 변화 과정

시간	1	2	3	4	5	6	7	8	9	10	11	12	13	14
참조 스트링	1	2	6	1	4	5	1	2	1	4	5	6	4	5
주기억장치 상태	1	1	1	1	1	1	1	1	1	1	1	1	1	4
		2	2	2	2	5	5	5	5	5	5	5	5	5
			6	6	6	6	6	2	2	2	2	6	6	6
					4	4	4	4	4	4	4	4	4	4
페이지 부재 발생 여부	F	F	F		F	F		F				F		

→ 페이지 부재의 발생 횟수 : 총 7번 발생

## 정리 하기

### 컴퓨터 기억장치의 계층적 구조

#### ✓ 개요

- 보조기억장치 : 낮은 속도지만 큰 용량
- 주기억장치 : 빠른 속도
- 버퍼기억장치 : 작은 용량이지만  
고속으로 중앙처리장치에 접근

## 정리 하기

### 캐시 기억장치의 원리

#### ✓ 개요

- 주기억장치에 비해 5~10배의 빠른 접근 속도를 제공하는 기억장치
- 주기억장치에서 자주 사용하는 명령들을 옮겨 저장
- 빠른 속도로 중앙처리장치에 제공하는 기억장치

#### ✓ 동작 순서

- 중앙처리장치가 명령어를 처리하기 위해 캐시 기억장치에 접근
- 적중(Hit) : 그 내용을 찾았을 때
- 실패(Miss) : 그 내용을 못 찾았을 때

## 정리 하기

### 캐시 기억장치의 설계

#### ✓ 사상함수

- 사상 : 주기억장치와 캐시 기억장치 사이에서 정보를 옮기는 것
- 직접사상, 연관사상, 집합연관사상 등이 있음

### 교체 알고리즘

#### ✓ 개요

- 캐시 기억장치가 가득 차 있는 상태에서 캐시 기억장치의 일부를 제거하고 주기억장치로부터 새로운 데이터를 가져와야 하는 경우 캐시의 내용을 제거하는 방식



- 다음 시간에 살펴 볼 내용 -

## 08주차 주기억장치

수고하셨습니다.