



[J02122] 컴퓨터구조

2022년 1학기

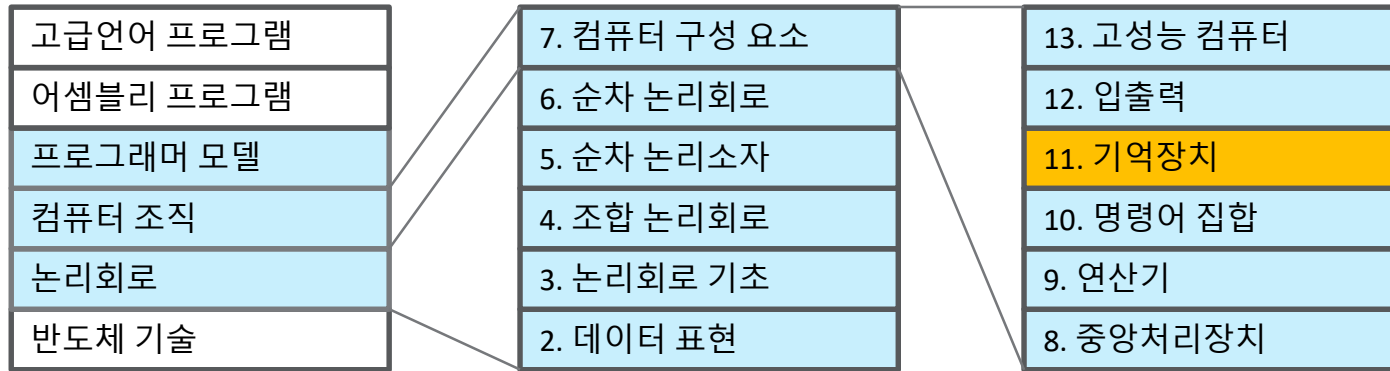
상명대학교 소프트웨어학과 박희민

- 11.1 기억장치 특성
- 11.2 반도체 기억장치
- 11.3 캐시 기억장치
- 11.4 가상 기억장치
- 11.5 요약

2022-05-25

CHAP11 기억장치

제11장 기억장치



- 학습 목표
 - 기억장치의 특성을 나타내는 용어를 설명할 수 있다.
 - 캐시 기억장치 구조를 제시하고 동작원리를 설명할 수 있다.
 - 가상 기억장치 구조를 제시하고 동작 원리를 설명할 수 있다.
- 내용
 - 11.1 기억장치 특성
 - 11.2 반도체 기억장치
 - 11.3 캐시 기억장치
 - 11.4 가상 기억장치
 - 11.5 요약

11.1 기억장치 특성

- 학습 목표
 - 기억장치 종류별 특징을 설명할 수 있다.
 - 기억장치와 관련된 용어의 의미를 이해한다.
- 내용
 - 11.1.1 기억장치 종류
 - 11.1.2 물리적 특성
 - 11.1.3 용량과 전송 단위
 - 11.1.4 액세스 방법
 - 11.1.5 성능 척도
 - 11.1.6 기억장치 계층

11.1.1 기억장치 종류

- 주기억장치
 - 중앙처리장치와 on-line 연결
 - 중앙처리장치가 필요할 때 즉시 사용할 수 있다.
 - ROM (Read Only Memory): 비휘발성, 부트로더
 - RAM (Random Access Memory): 휘발성
- 보조기억장치
 - 중앙처리장치와 off-line 연결
 - 중앙처리장치가 데이터를 사용하려면 별도로 연결 과정을 거쳐야 한다.
 - 데이터를 반영구적으로 보관하는 일종의 입출력장치
 - 플로피 디스크, 하드 디스크, CD-ROM, USB 기억장치, SSD

11.1.2 물리적 특성

- 휘발성(volatile)
 - 기억장치에서 전원을 제거하면, 저장된 내용이 없어진다.
 - RAM
- 비휘발성(nonvolatile)
 - 기억장치에서 전원을 제거하더라도, 내용을 그대로 유지한다.
 - ROM, 보조기억장치
- 삭제 가능(erasable)
 - 한 번 기록된 내용을 (on-line 상태로) 지우고 다른 값으로 갱신할 수 있다.
 - RAM, 하드 디스크
- 삭제 불가능(non-erasable)
 - 한 번 기록된 내용을 (on-line 상태로) 지우거나 갱신할 수 없다.
 - ROM, CD-ROM

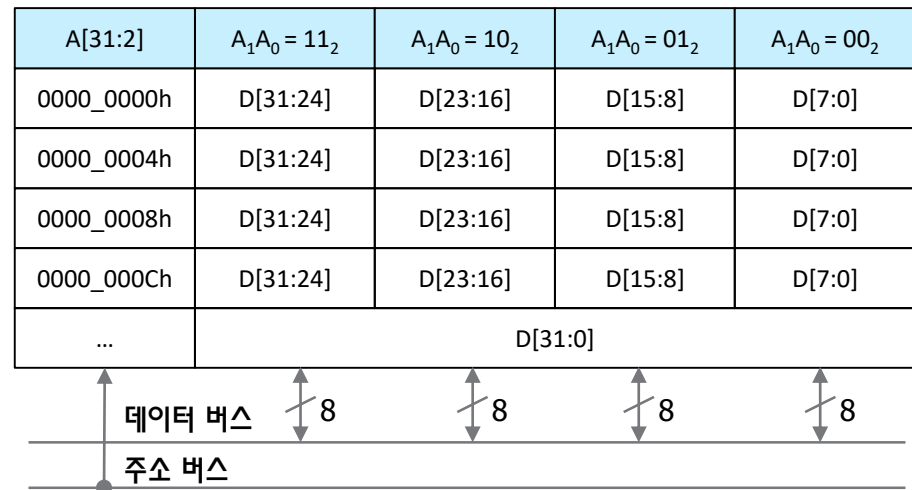
11.3 용량과 전송 단위

- 기억장치 용량

- 기억장치가 저장할 수 있는 데이터의 전체 크기
- 일반적으로 바이트 단위

- 주기억장치

- 바이트 단위 구성
- 정렬된 단어 단위 액세스
- 8, 16, 32, 64 비트 단어



주기억장치 조직 예: 바이트 단위 주소 지정/32비트 단어

- 보조기억장치

- 바이트 단위 구성
- 블록 단위 액세스
- 블록 크기: 512 bytes, 1Kbytes, 2Kbytes, 또는 4Kbytes

11.1.4 액세스 방법

구분	액세스 방법	예	액세스 시간
순차 액세스 (sequential access)	처음부터 차례대로	자기 테이프	파일 위치로 이동하는 시간 추가
직접 액세스 (direct access)	헤드를 임의의 위치로 이동하여 액세스	자기 디스크 CD-ROM	디스크 헤드 이동 시간 추가
임의 액세스 (random access)	임의의 위치 액세스 + 액세스 시간 동일	반도체 주기억장치 ROM, RAM	주소 제공 후 액세스하므로 항상 같음
연관 액세스 (associative access)	테이블 구조에 대한 내용(키)에 의한 액세스	연관 기억장치 CAM	데이터 위치와 무관하게 항상 같음

- 연관 기억장치(associative memory)
 - CAM(Content Addressable Memory)
 - 고속 탐색을 위한 특수 기억장치

11.1.5 성능 척도

성능 척도	정의
액세스 시간 (access time)	기억장치에 주소와 제어신호를 제공한 후 데이터가 읽혀지거나 쓰여지기 시작할 때까지 시간
사이클 시간 (cycle time)	기억장치를 연속적으로 액세스할 때 소요되는 시간 액세스 시간 + 여유 시간
전송률 (transfer rate)	1초에 액세스할 수 있는 데이터의 양. $1/(\text{사이클 시간})$ 단위는 bps (bits per second)
대역 폭 (bandwidth)	데이터 통신에서 채널이 수용할 수 있는 주파수 대역의 범위 기억장치에서는 전송률과 같은 의미로 사용함

[예제 11-1]

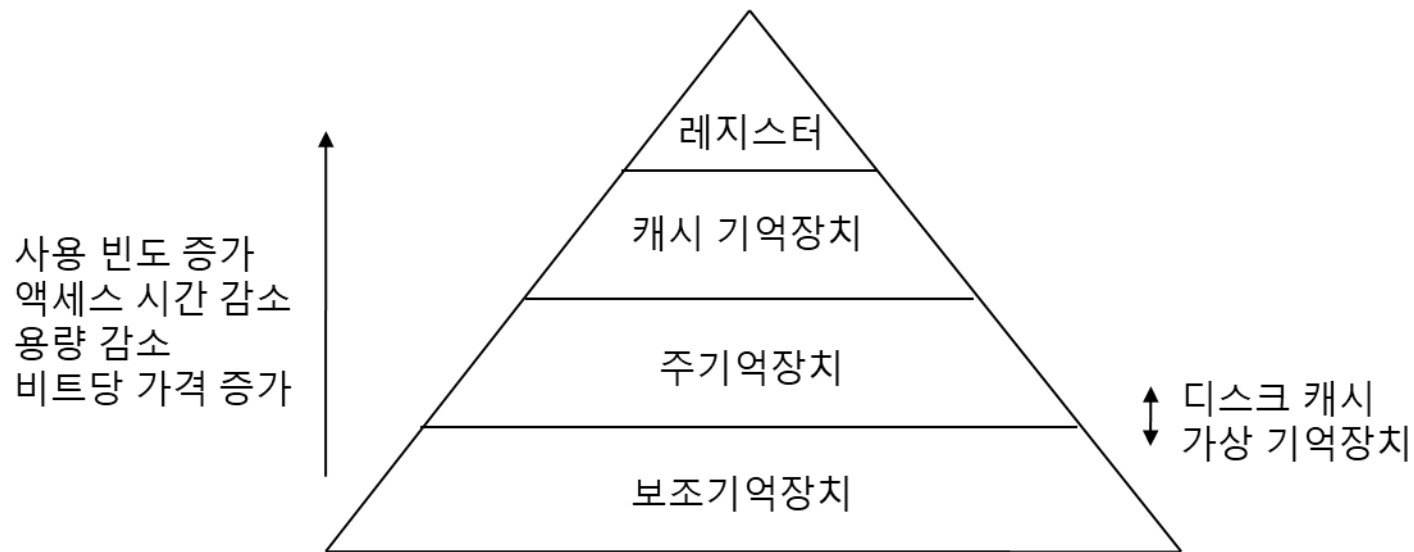
- 기억장치 사이클 시간이 100 nsec/byte일 때, 비트 단위 데이터 전송률은?

[풀이]

- 전송률 = $1/(\text{사이클 시간})$
 $= 1/100 \text{ (nsec/byte)} = 10 \text{ Mbytes/sec} = 80 \text{ Mbits/sec}$

11.1.6 기억장치 계층

- 기억장치 계층
 - 액세스 속도는 빠르고, 용량은 많게.
 - 당장 처리할 데이터를 속도가 빠른 장치로 옮겨놓고 사용



11.1 기억장치 특성 요약

- 주기억장치
 - ROM: 부트로더 저장. 비휘발성, 삭제 불가능
 - RAM: 프로그램 실행용. 휘발성, 삭제 가능
 - 바이트 단위 구성, 단어 단위 액세스
- 액세스 방법
 - 순차 액세스: 처음부터 차례대로 액세스
 - 직접 액세스: 헤드를 위치로 이동하고 나서 액세스
 - 임의 액세스: 주소를 제공하고 액세스
 - 연관 액세스: 키필드와 일치하는 데이터 액세스
- 기억장치 계층
 - 평균 액세스 속도 개선 목적
 - 속도가 느리고 용량이 큰 기억장치의 데이터를
 - 속도가 빠르고 용량이 작은 기억장치로 옮겨놓고 사용.

11.2 반도체 기억장치

- 학습 목표
 - 반도체 기억장치의 종류와 특성에 대한 이해
- 내용
 - 반도체 기억장치 종류
 - 용도별 분류

반도체 기억장치 분류

분류	종류	사용 방법	휘발성	삭제(단위)	프로그램	기타
RAM	SRAM	읽기/쓰기	휘발성	전기(바이트)	온라인, 전기	
	DRAM	읽기/쓰기	휘발성	전기(바이트)	온라인, 전기	재충전
ROM	Mask ROM	읽기	비휘발성	불가	오프라인, 생산 공정	
	PROM	읽기	비휘발성	불가	오프라인, 전용 장치	
	EPROM	읽기	비휘발성	자외선(전체)	오프라인, 전용 장치	
	EEPROM	읽기/쓰기	비휘발성	전기(바이트)	온라인, 전기	
	Flash Memory	읽기/쓰기	비휘발성	전기(블록)	전기	

용도별 분류

RAM 특성 비교

종류	셀	재충전	제어기	셀 크기	용량	속도	용도
SRAM	플립플롭	불필요	불필요	크다.	적다.	빠름	캐시 기억장치, 소형 시스템
DRAM	캐패시터	필요	필요	작다.	많다.	느림	대형 시스템

반도체 기억장치의 용도별 분류

구분	용도	반도체 기억장치 소자	인터페이스
주기억장치	읽기 전용	ROM, PROM, EPROM, NOR Flash	주소, 데이터, 읽기, (프로그램)
	읽기/쓰기	SRAM, DRAM	주소, 데이터, 읽기, 쓰기
보조기억장치	읽기/쓰기	EEPROM, NAND Flash	입출력

- ※ EP(Erase-Program) cycle
 - 삭제 가능한 ROM 종류의 재기록 가능 횟수
 - 일반적으로 10만번

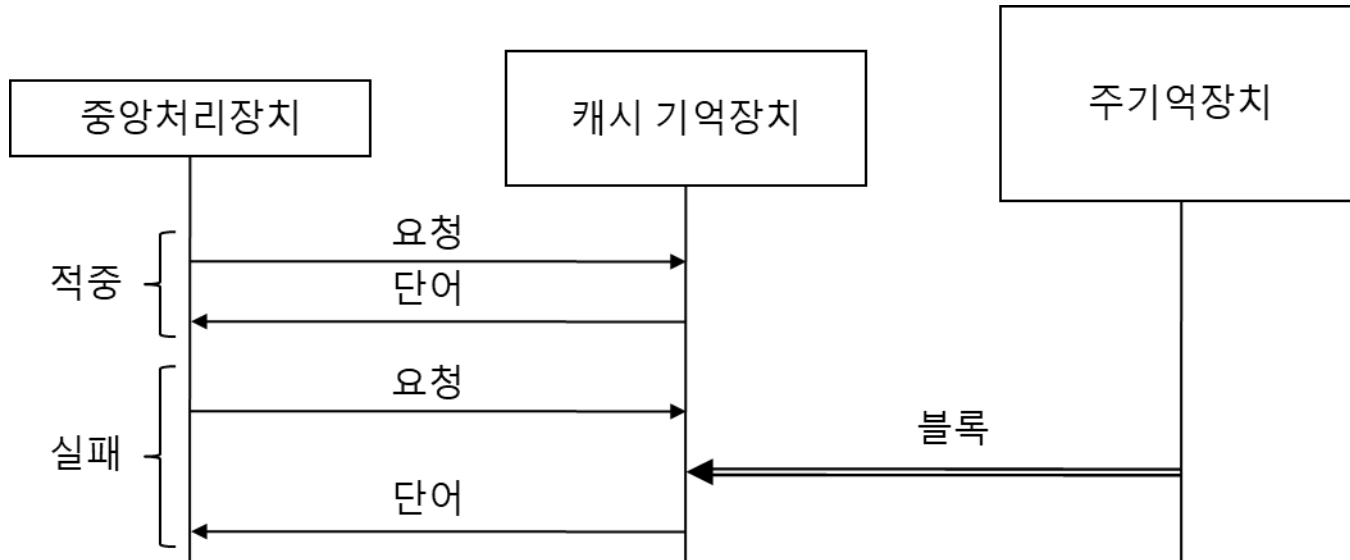
11.3 캐시 기억장치

- 학습 목표
 - 캐시 기억장치를 도입함으로써 주기억장치의 평균 액세스 속도가 개선되는 원리를 설명할 수 있다.
 - 캐시 기억장치의 구조를 제시하고 운영 방법을 설명할 수 있다.
- 내용
 - 11.3.1 참조의 지역성
 - 11.3.2 캐시 기억장치 구조
 - 11.3.3 캐시 설계시 고려 사항
 - 11.3.4 매핑 함수

11.3.1 참조의 지역성

- 참조의 지역성 (locality of reference)
 - 프로세서가 프로그램을 수행할 때 기억장치의 특정 부분을 집중적으로 참조하는 경향이 있다.
- 공간적 참조의 지역성
 - 기억장치에 가깝게 배치되어 있는 명령어 또는 데이터를 다음 번에도 계속 참조할 가능성이 크다.
 - 명령어> 거리가 짧은 루프 반복 실행.
 - 데이터> 데이터는 기억장치에 같은 영역에 가깝게 배치되어 있음.
- 시간적 참조의 지역성
 - 한 번 사용한 명령어와 데이터를 짧은 시간 안에 다시 참조할 가능성이 크다.
 - 명령어> 함수 호출 후 복귀
 - 데이터> 함수 안 프로그램은 같은 지역 변수를 여러 번 사용함.

데이터 전송 단위



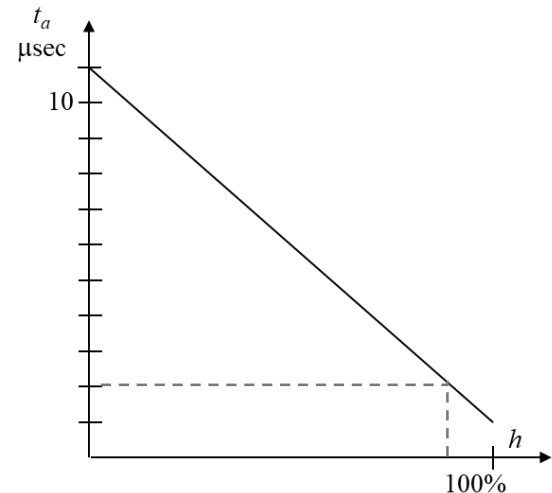
- **적중(hit)**
 - 중앙처리장치가 원하는 단어가 캐시 기억장치에 들어 있음.
 - 캐시에서 단어 액세스
- **실패(miss)**
 - 중앙처리장치가 원하는 단어가 캐시 기억장치에 없음.
 - 주기억장치에서 단어를 포함하고 있는 블록을 캐시로 전송 후, 캐시에서 단어 액세스

적중률

$$\text{hit ratio} = \frac{\text{number of hits}}{\text{number of accesses}}$$

$$t_a = h \times t_c + (1 - h) \times (t_m + t_c) = t_c + (1 - h) \times t_m$$

- [예제 11-2]
 - 캐시 액세스 시간 = $1\mu\text{sec}$
 - 주기억장치 액세스 시간 = $10\mu\text{sec}$
- 1) 적중률에 대한 평균 액세스 속도 그래프는?
- 2) 적중률이 90%일 때, 평균 액세스 시간은?
- [풀이]
 - 1) 평균 액세스 시간 $t_a = t_c + (1 - h)t_m = 11 - 10h$
 - 2) $t_a(0.9) = 11 - 0.9 \times 10 = 2\mu\text{sec}$



11.3.2 캐시 기억장치 구조

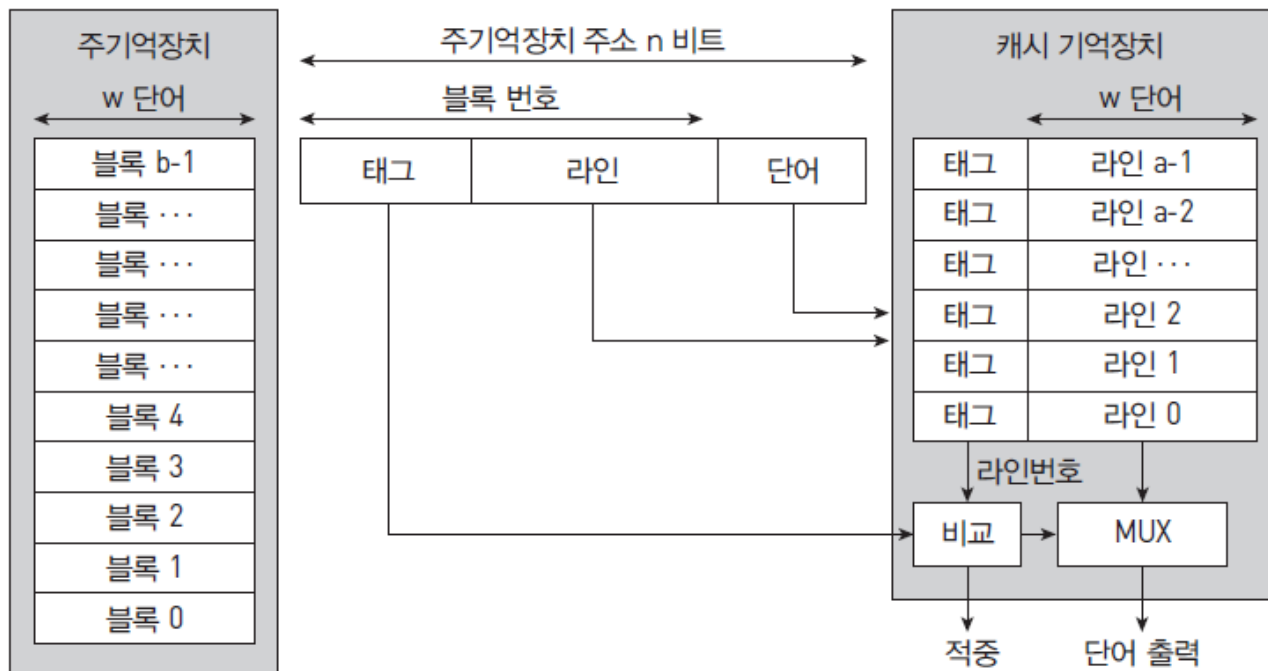
- 전송 단위
 - 주기억장치 전송 단위: 블록 캐시 기억장치 전송 단위: 라인 (슬롯)
 - 일반적으로 블록 크기 = 라인 크기

<표 11-3> 주기억장치와 캐시 기억장치의 제원

항목	주기억장치	캐시 기억장치
주소선의 수	n비트	k비트
전체 용량	2^n 단어	2^k 단어
데이터 전송 단위 및 크기	블록: w단어	라인: w단어
블록(라인) 수	블록 수: $b = 2^n/w$	라인 수: $a = 2^k/w$

- [예제 11-3]
 - 주기억장치 용량 = 16M바이트, 캐시 기억장치 용량 = 32K바이트
 - 블록 크기 = 라인 크기 = 64바이트
 - 주기억장치 블록 수와 캐시 라인 수는?
- [풀이]
 - 주기억장치 블록 수 $b = 16\text{M바이트} / 64\text{바이트} = 256\text{K블록}$
 - 캐시 기억장치 라인 수 $a = 32\text{K바이트} / 64\text{바이트} = 512\text{라인}$

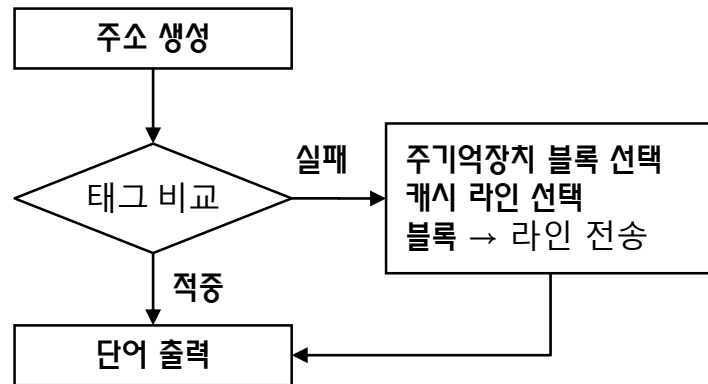
주기억장치와 캐시 기억장치 구조



- 주기억장치 주소 = {태그 필드, 라인 필드, 단어 필드}
 - 단어 필드:** 캐시 라인 안에서 단어 지정. 길이 = $\log_2 W$ (W : 라인 내 단어 수)
 - 라인 필드:** 캐시의 라인 번호 지정. (길이는 매핑 방법에 따라 다르다.)
 - 태그 필드:** 캐시 라인에 어느 주기억장치 블록이 들어 있는지 나타낸다. 캐시의 태그 영역에 저장된다.

데이터 액세스

데이터 읽기



데이터 쓰기

- 읽기보다 복잡
 - 캐시와 주기억장치를 모두 수정해야 한다.
 - 별도의 정책 필요

- 캐시 동작

- 캐시 도입 목적이 액세스 속도 개선이므로 캐시 동작은 모두 하드웨어로 처리된다. 프로그래머는 캐시가 있음을 모른다.
- 빠르게 검색할 수 있도록 캐시는 연관기억장치로 만들어져 있다.

11.3.3 캐시 설계시 고려 사항

- 계층적 구성
 - On-chip 레벨 1 캐시: 명령어 캐시, 데이터 캐시
 - Off-chip 레벨 2 캐시



- 매핑 함수 (11.3.4절)
 - 주기억장치 블록을 캐시 라인으로 복사하는 규칙
 - 직접 매핑, 완전 연관 매핑, 집합 연관 매핑

캐시 설계 요소

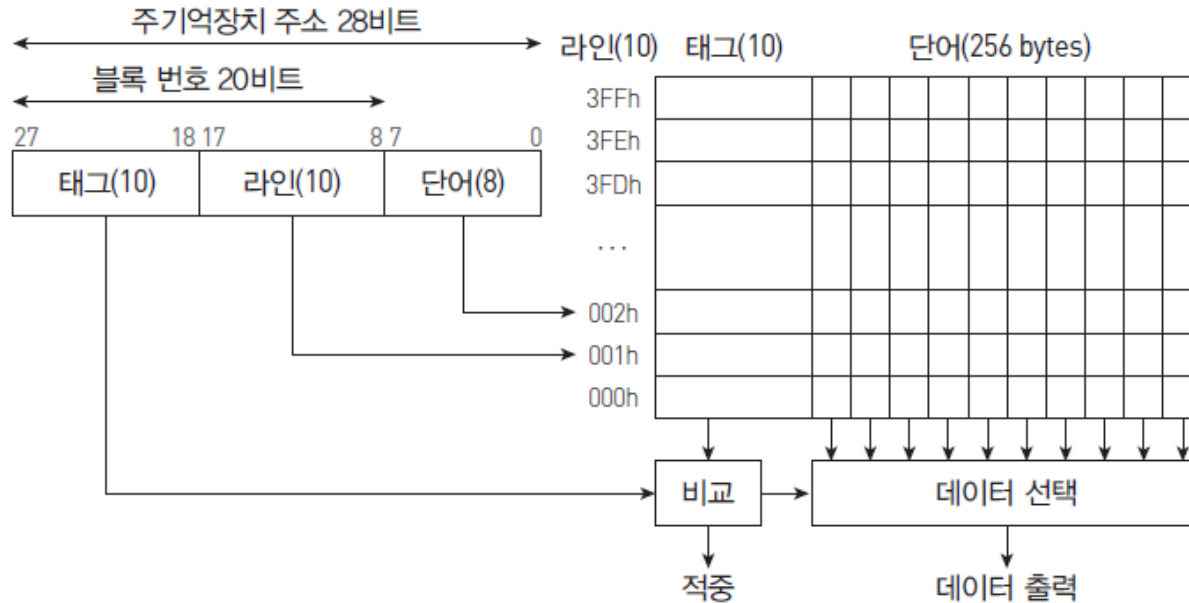
- 교체 정책
 - 캐시가 모두 채워졌을 때, 교체 라인 선택 방법
 - LRU (Least Recently Used): 최근 사용 빈도가 가장 작은 라인 교체
- 쓰기 정책
 - 데이터 쓰기 처리 방법
 - Write-through: 캐시와 주기억장치 모두 수정
 - Write-back: 캐시만 수정. 캐시 라인이 교체될 때 주기억장치로 복사.
- 일관성 유지
 - 주기억장치와 캐시의 데이터를 동일하게 유지하는 방법
 - 멀티 프로세서 환경에서 중요

11.3.4 매핑 함수

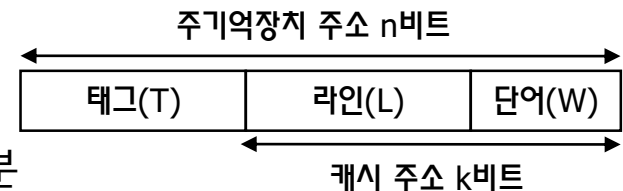
- 매핑 함수 (mapping function)
 - 주기억장치 한 개의 블록을 캐시 라인에 할당하는 규칙
- 종류
 - 직접 매핑 (direct mapping)
 - 완전 연관 매핑 (full associative mapping)
 - 세트 연관 매핑 (set-associative mapping)
- 예제

종류	용량	주소	블록 내 단어 수	블록(라인) 수
주기억장치	256M바이트	$n = 28$ 비트	$w = 256$ 바이트	$b = 1$ M블록
캐시 기억장치	256K바이트	$k = 18$ 비트	$w = 256$ 바이트	$a = 1$ K라인

직접 매핑



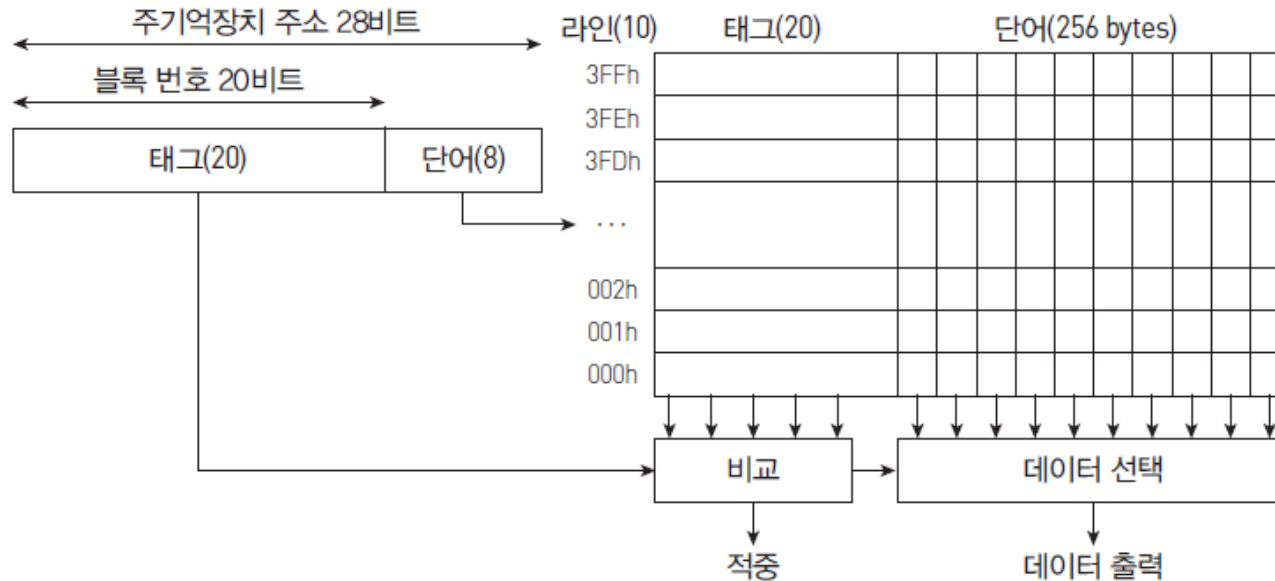
- 직접 매핑: 주기억장치 블록을 라인 필드에 해당하는 캐시 라인에 저장
- 주기억장치 n 비트 주소
 - 단어 필드 $W = \log_2(\text{라인 크기})$
 - 라인 필드 $L = \log_2(\text{캐시 용량} / \text{라인 크기})$
 - 태그 필드 $T = n - (W + L)$ // 주소의 나머지 부분



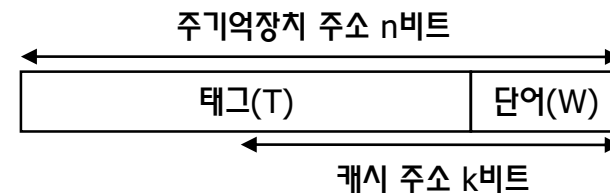
직접 매핑 특성

- [예제 11-4] 주기억장치 주소 50C_0108h
 - 1) 단어를 저장할 캐시 라인 번호는?
 - 2) 태그 영역에 저장할 값은?
 - 3) 적중 여부를 판정할 비교기의 비트 크기와 수는?
- [풀이] 50C_0108 = 0101_0000_1100_0000_0001_0000_1000
= (태그)01_0100_0011_(라인)00_0000_0001_(단어)0000_1000
 - 1) 라인 번호 = 001h, 라인 내 08번째 단어
 - 2) 태그 = 143h
 - 3) 10비트 비교기 한 개
- 특성
 - 주기억장치 블록이 저장될 캐시 라인이 하나로 고정 → 교체정책 불필요.
 - 히트 검사: 태그 비트 크기의 비교기 한 개 필요.
 - 같은 캐시 라인으로 매핑되는 두 개의 블록을 번갈아 사용하면 효율이 떨어짐.

완전 연관 매핑



- 완전 연관 매핑: 주기억장치 블록을 임의의 캐시 라인에 저장
- 주기억장치 주소 n 비트
 - 단어 필드 $W = \log_2(\text{라인 크기})$
 - 태그 필드 $T = n - W$

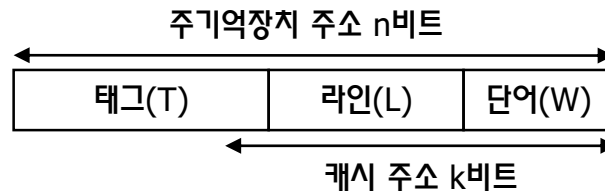


완전 연관 매핑 특성

- [예제 11-5] 주기억장치 주소 50C_0108h
 - 1) 단어를 저장할 캐시 라인 번호는?
 - 2) 태그 영역에 저장할 값은?
 - 3) 적중 여부를 판정할 비교기의 비트 크기와 수는?
- [풀이] 50C_0108 = 0101_0000_1100_0000_0001_0000_1000
= (태그)0101_0000_1100_0000_0001_(단어)0000_1000
 - 1) 라인 번호 = 1024개 중 임의의 라인, 라인 내 08번째 단어
 - 2) 태그 = 50C01h
 - 3) 20비트 비교기 1024 개
- 특성
 - 주기억장치 블록이 임의의 캐시 라인으로 매핑 → 교체 정책 필요
 - 히트 검사를 위해 모든 라인 검사 필요.
 - 비교기 수 = 태그 비트 크기 × 캐시 라인 수

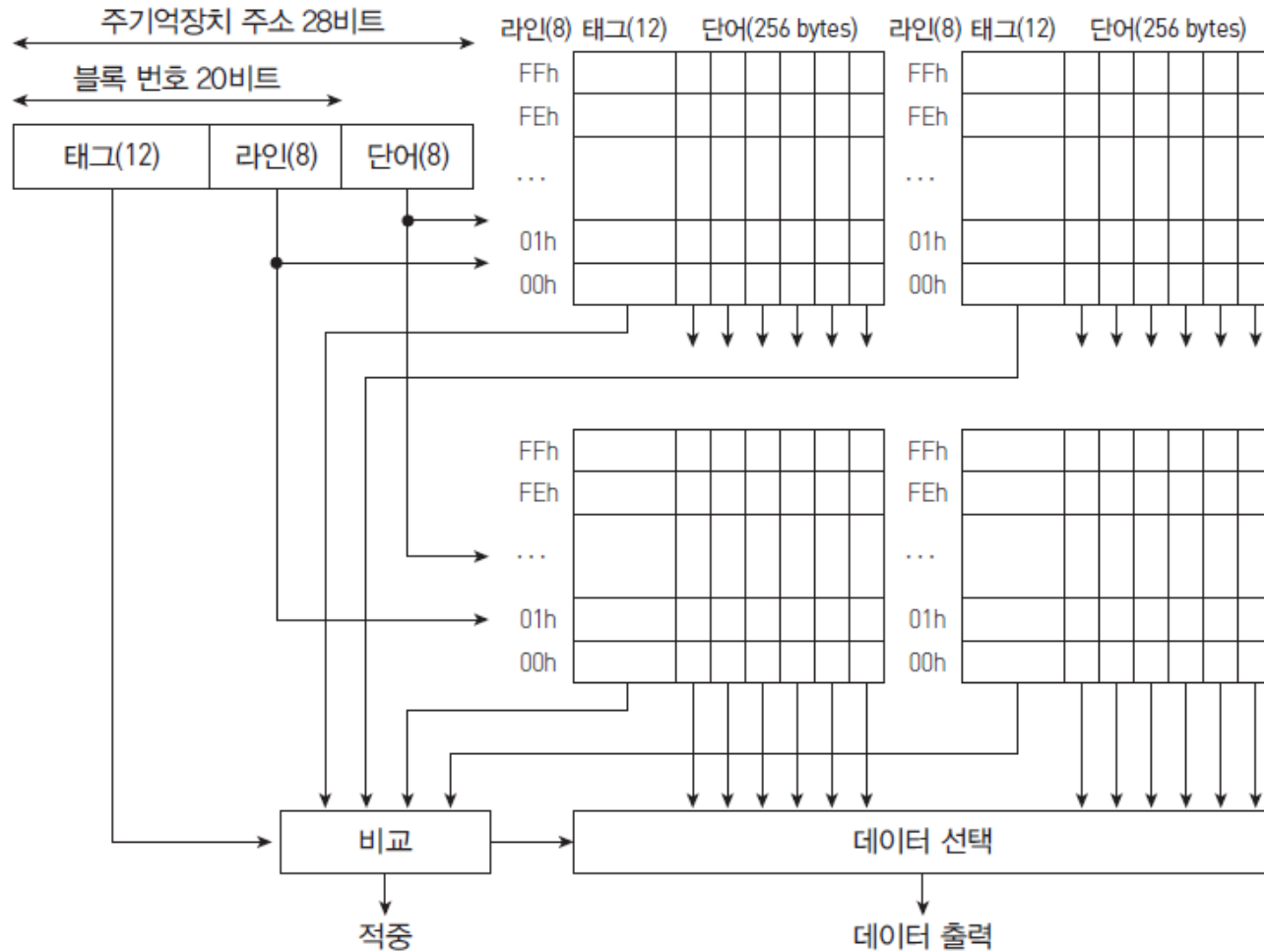
세트 연관 매핑

- s-way 세트 연관 매핑
 - 캐시를 s개의 집합으로 나누고, (일반적으로 s는 2, 4, 8 중 하나)
 - 주기억장치 블록을 각 캐시 집합 중 하나의 라인에 배정
 - 캐시 세트의 용량 = 캐시 용량 / s
- 주기억장치 주소 n비트
 - 단어 필드 $W = \log_2(\text{라인 크기})$
 - 라인 필드 $L = \log_2(\text{캐시 세트 용량/라인 크기}) = \log_2(\text{캐시 세트 라인 수})$
 - 태그 필드 $T = n - (W+L)$ // 주소의 나머지 부분



캐시 용량 = 2^k
캐시 세트 수 = s
캐시 세트 용량 = $2^k/s$

4-way 세트 연관 매핑



세트 연관 매핑 특성

- [예제 11-6] 주기억장치 주소 50C_0108h
 - 1) 단어를 저장할 캐시 라인 번호는?
 - 2) 태그 영역에 저장할 값은?
 - 3) 적중 여부를 판정할 비교기의 비트 크기와 수는?
- [풀이] 50C_0108 = 0101_0000_1100_0000_0001_0000_1000
= (태그)0101_0000_1100_(라인)0000_0001_(단어)0000_1000
 - 1) 각 세트의 01라인 중 하나에 저장된다. 라인 내 08번째 단어
 - 2) 태그 = 50Ch
 - 3) 12비트 비교기 4 개
- 특성
 - 주기억장치 블록을 캐시 세트 중 하나의 라인에 저장 → 교체정책 필요.
 - 히트 검사를 위해 각 세트마다 하나의 라인 검사 필요.
 - 비교기 수 = 태그 비트 크기 × 세트 수

11.3 캐시 기억장치 요약

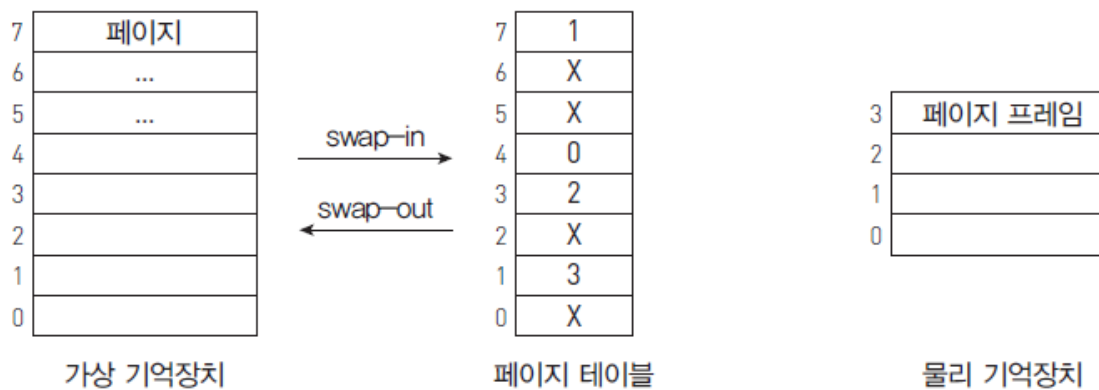
- 참조의 지역성
 - 프로세서는 프로그램을 실행할 때 기억장치의 특정 부분을 집중적으로 액세스 한다.
- 기억장치 평균 속도 개선
 - 속도가 빠른 캐시 기억장치에 데이터를 복사해 두고 여러 번 사용한다.
- 매핑 함수
 - 주기억장치 블록을 캐시 라인으로 매핑하는 규칙
 - 직접 매핑: 주기억장치 블록을 라인 필드에 해당하는 캐시 라인에 저장
 - 완전 연관 매핑: 주기억장치 블록을 임의의 캐시 라인에 저장
 - 집합 연관 매핑: 캐시를 s 개의 집합으로 나누고, 주기억장치 블록을 각 집합 중 하나의 라인에 배정

11.4 가상 기억장치

- 기억장치
 - 캐시 기억장치: 기억장치 액세스 속도 개선
 - 가상 기억장치: 기억장치 용량 확장
 - 운영체제가 운영 담당
 - 프로세서는 MMU(Memory Management Unit) 제공
- 학습 목표
 - 가상 기억장치를 도입함으로써 주기억장치의 용량을 확장하는 원리를 설명할 수 있다.
 - 가상 기억장치를 운영하는 방법을 설명할 수 있다.
- 내용
 - 11.4.1 페이지 테이블
 - 11.4.2 변환 우선참조 버퍼
 - 11.4.3 가상 기억장치 기능 확장

11.4.1 페이지 테이블

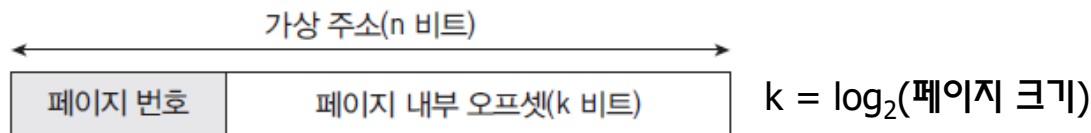
- 가상 기억장치
 - 가상 기억장치는 하드 디스크에 주기억장치 내용을 저장하려고 확보한 공간
 - 주기억장치 용량 < 프로세서 기억 공간
 - 가상 기억장치에서 데이터의 일부를 페이지 단위로 주기억장치로 옮겨 놓고 사용
- 가상 기억장치 운영
 - Page: 크기가 같게 기억장치를 나눈 영역, 일반적으로 4k바이트
 - Page frame: 물리 기억장치를 페이지 크기로 나눈 영역
 - 페이지 테이블: 페이지가 페이지 테이블에 배정된 상태를 기록하는 테이블
 - 페이지 부재 (page fault) → 요구 페이지징 (demand paging)



[그림 11-11] 가상 기억장치 운영 개념

기억장치 주소

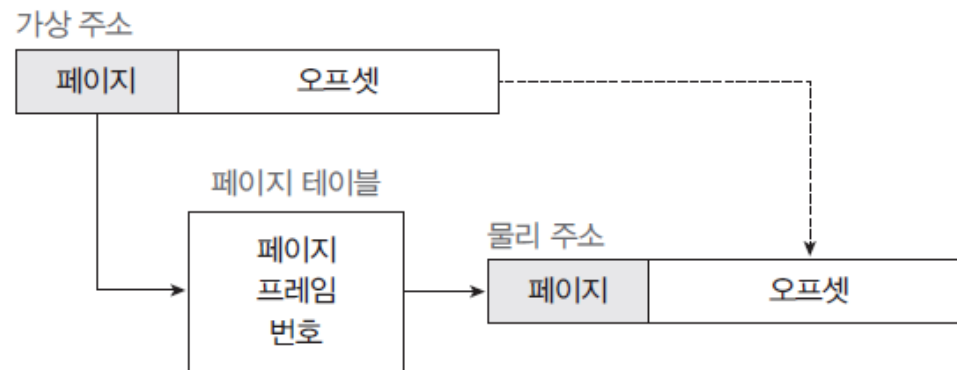
- 기억장치 주소
 - 논리 주소(logical address): 중앙처리장치가 데이터를 액세스하는 주소
 - 가상 주소(virtual address) 또는 선형 주소(linear address): 가상 기억장치에 대한 주소
 - 물리 주소(physical address): 주기억장치를 액세스하는 주소
 - 일반적으로, 가상 주소 = 선형 주소 = 논리 주소
- 가상주소 → 물리주소 변환
 - 가상 주소의 페이지 번호로 페이지 테이블을 참조하여 물리 주소의 페이지 프레임 번호를 구한다.



[그림 11-12] 가상 주소

페이지 테이블

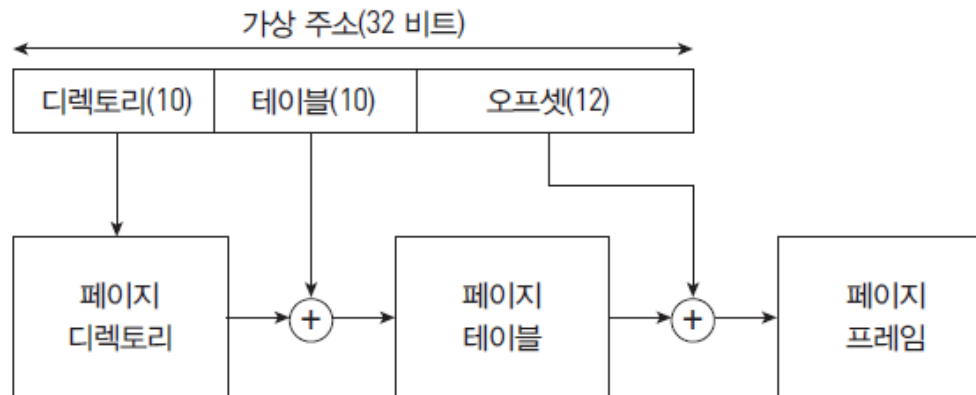
가상 주소의 페이지 번호 → (페이지 테이블) → 물리 주소의 페이지 프레임 번호



[그림 11-13] 페이지 테이블 참조

2 단계 페이징

- 페이지 테이블 크기 문제
 - 32-bit 프로세서: 4 Kbytes page, 20 bit 페이지 번호
 - Page table size 1 M entries
- 2단계 페이징



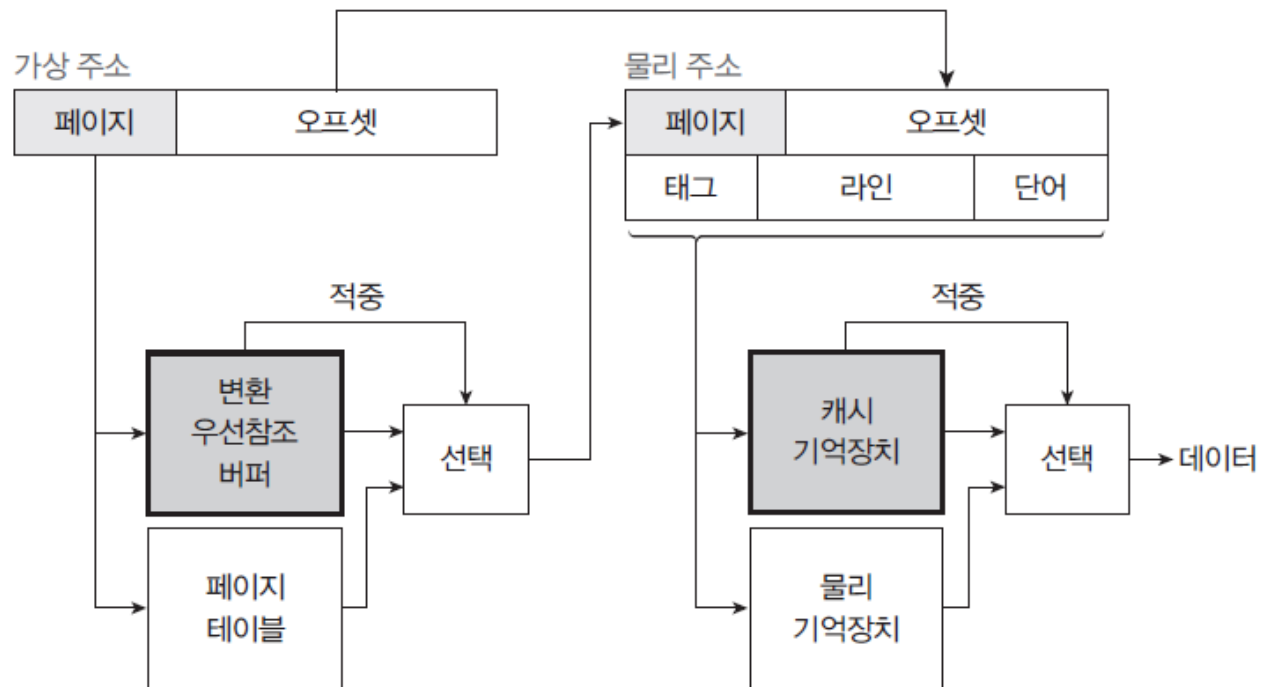
[그림 11-14] 2단계 페이징

11.4.2 변환 우선참조 버퍼

- 페이징 기법의 데이터 액세스
 - 기억장치 2번 액세스
 - 1) 가상 주소로 페이지 테이블을 액세스하여 물리 주소 생성
 - 2) 물리 주소로 데이터 액세스
- 변환 우선참조 버퍼(TLB, Translation Lookaside Buffer)
 - 프로세서 안에 내장된 특수한 연관 액세스 기억장치
 - 최근에 사용한 페이지 테이블의 내용을 복사하여 저장
 - TLB 히트이면, 페이지 테이블을 참조하지 않고 바로 물리주소 생성
 - TLB 항목이 모두 채워지면 교체 (LRU 알고리즘)

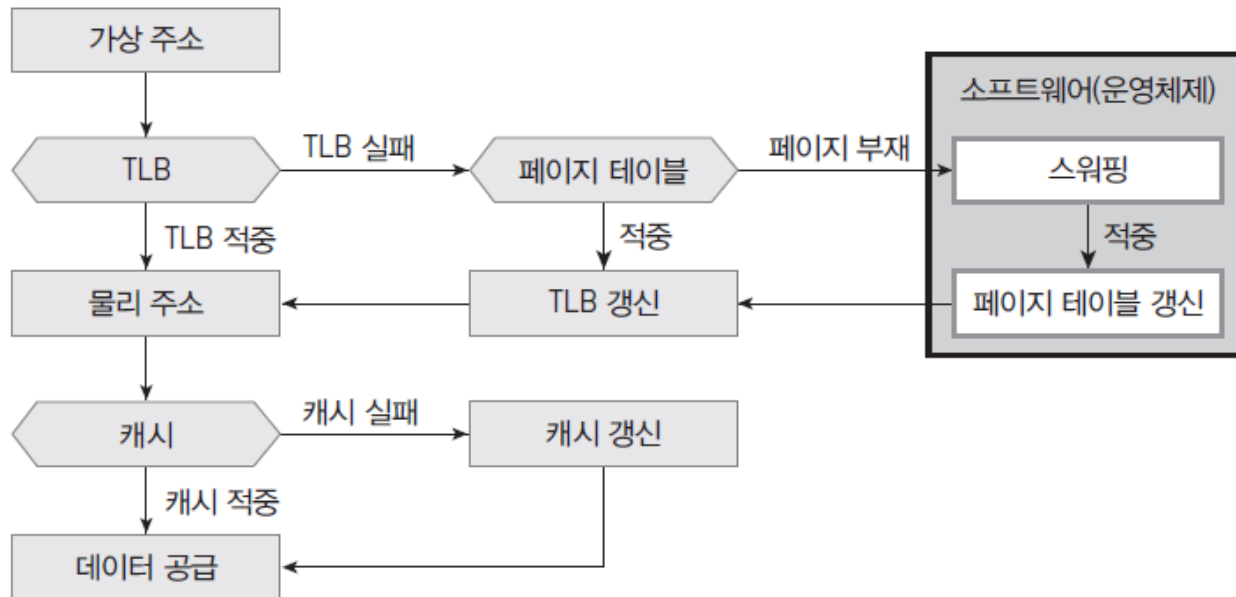
가상 주소 페이지 번호	페이지 프레임 번호	속성

기억장치 시스템



[그림 11-15] 기억장치 시스템 구성

기억장치 액세스 과정



[그림 11-16] 기억장치 액세스 과정