

Microprocessor (W13)

- Memory 3 -

Dong Min Kim
Department of IoT
Soonchunhyang University
dmk@sch.ac.kr

04 가상 기억 장치

05 연관 기억 장치

04 가상 기억 장치

❖ 가상 기억 장치(virtual memory)

- 하드 디스크처럼 용량이 큰 보조 기억 장치를 주기억 장치처럼 사용하는 개념
- CPU가 참조하는 가상 주소를 주기억 장치의 실제 주소로 변환하는 주소 매핑이 필요하다.

❖ 주소 매핑(address mapping)

- 프로그래머가 프로그램에 표시한 주소를 가상 주소 또는 논리 주소라 하고, 이들의 주소 집합을 **주소 공간**이라 한다.
- 실제 프로그램이 적재되는 주기억 장치의 주소를 물리 주소라 하고, 이들의 집합을 **메모리 공간**이라 한다.

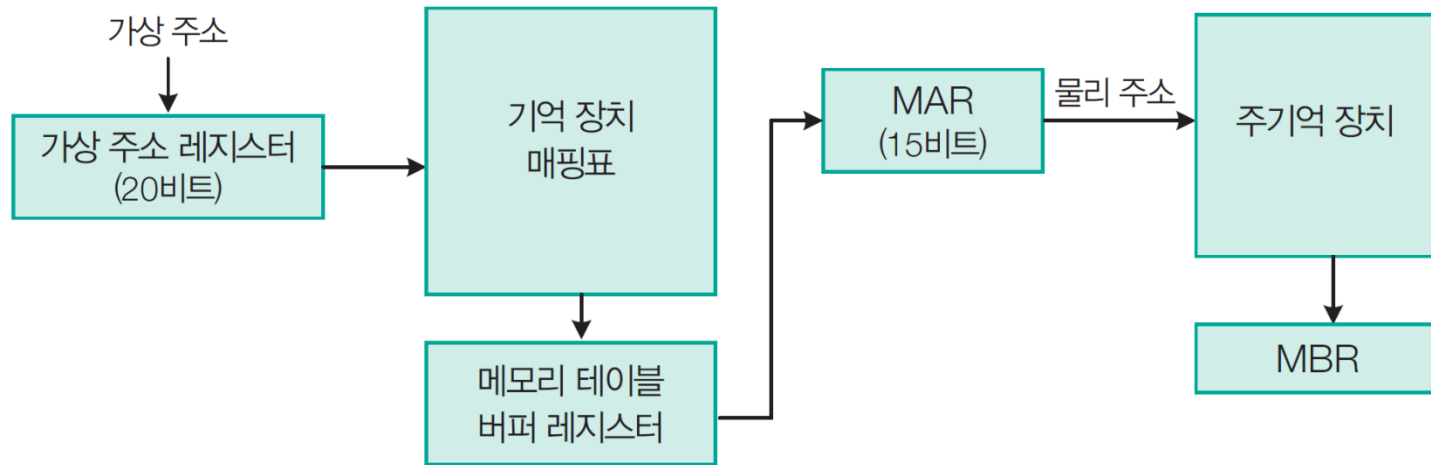


그림 6-36 가상 주소를 물리 주소로 변환하여 정보를 얻는 과정

1 가상 기억 장치의 매핑

□ 페이지에 의한 매핑

- **페이지**(page) : 주소 공간을 고정 크기로 나눈 것
- **블록**(block) : 메모리 공간을 고정 크기로 나눈 것
- 페이지에 대한 기억 장치 매핑표를 가지고 페이지를 블록으로 변환한다.
- 가상 주소 페이지가 주기억 장치에 존재하지 않는 경우에는 **페이지 오류**(page fault)가 발생되었다고 하며, 이러한 페이지 오류가 자주 발생하는 경우를 **스래싱**(thrashing)이라고 한다.
- 주소 공간이 8K, 메모리 공간이 4K인 컴퓨터 시스템의 예(최대 페이지 4개가 메모리 공간의 블록 4개 중 하나에 들어간다.)

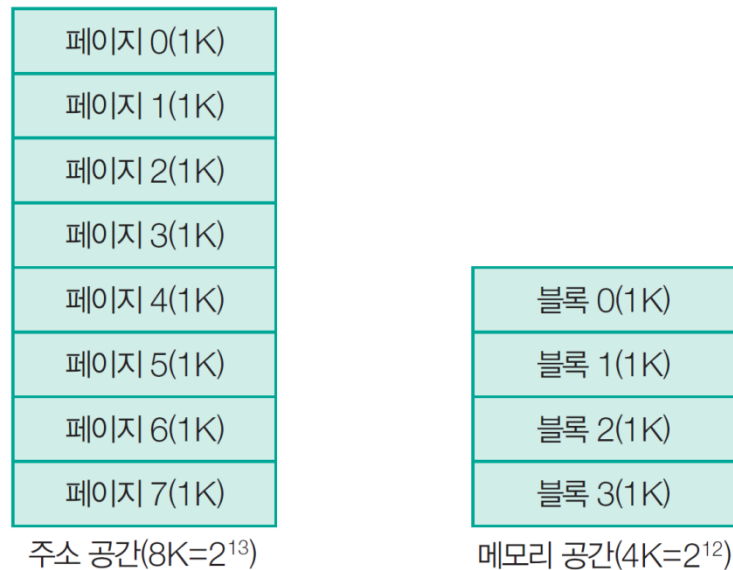


그림 6-37 주소 공간과 메모리 공간의 분리(1K)

04 가상 기억 장치

❖ 기억 장치 매핑표

- 페이지 번호에서 블록 번호로의 변환만 하면 된다.
- 그림에서 페이지 1, 2, 5, 6은 주기억 장치 블록 2, 0, 1, 3에 각각 저장되어 있다.
- 현존 비트가 1이면 해당 페이지가 주기억 장치에 존재함을 의미

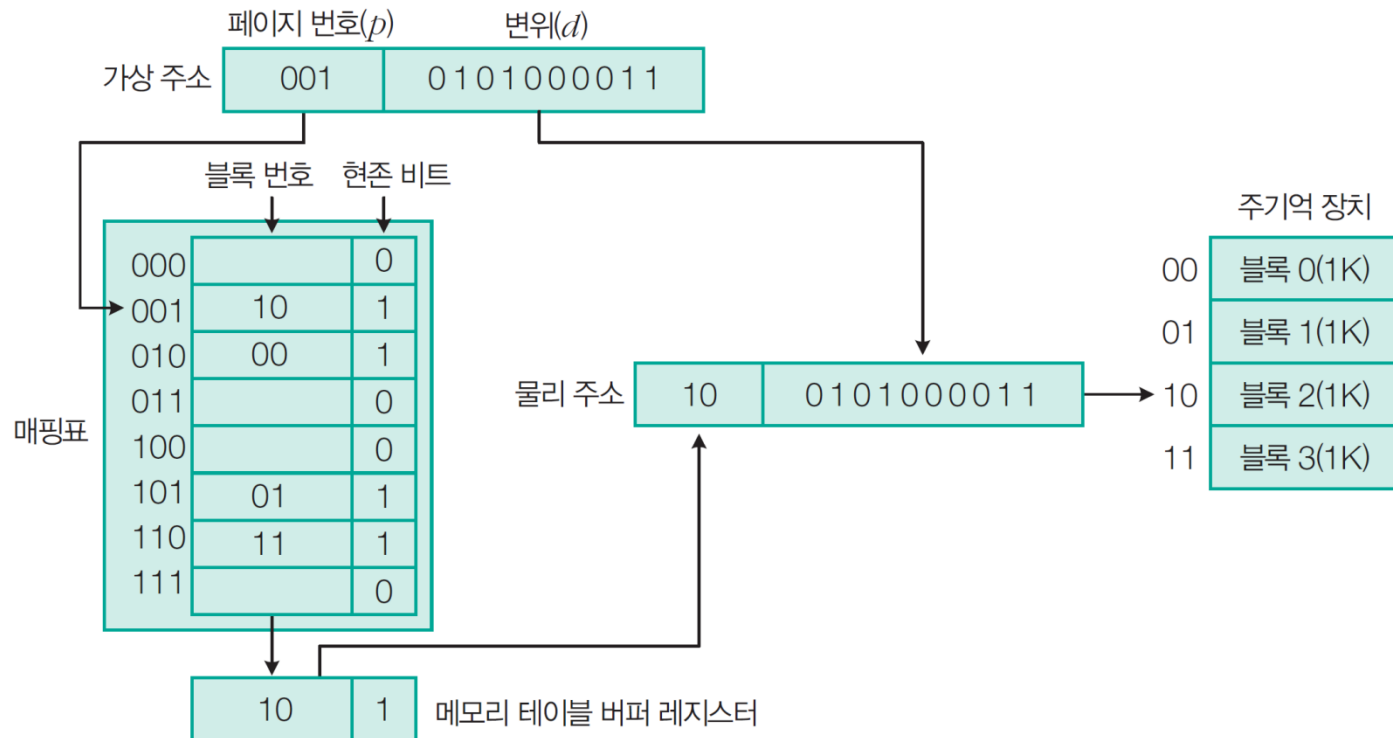


그림 6-38 기억 장치 매핑표 구조

04 가상 기억 장치

예제 6-13

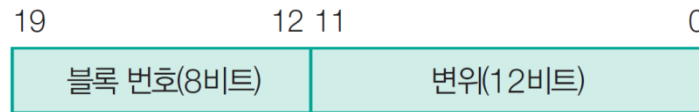
용량이 1MB인 주기억 장치를 가진 컴퓨터 시스템에서 32비트의 가상 주소를 사용한다고 하자. 페이지 크기가 1K 워드고, 1워드는 4바이트라고 할 때 가상 주소와 물리 주소의 구성은 어떻게 될까?

풀이

한 페이지는 $1K(=2^{10})$ 워드고, 한 워드가 $4(=2^2)$ 바이트이므로 한 페이지의 크기는 2^{12} 이므로 변위는 12비트이어야 한다. 주기억 장치의 용량이 $1M(=2^{20})$ 바이트고, 한 페이지의 크기는 2^{12} 바이트이므로 주기억 장치의 페이지 수는 $256(=2^8)$ 개다.

$$(\text{주기억 장치의 용량})/(\text{한 페이지의 크기})=2^{20}/2^{12}=2^8$$

따라서 물리 주소는 20비트로 구성되므로 페이지 주소에 8비트, 변위에 12비트가 할당된다.



가상 주소는 32비트이므로 페이지 번호에 20비트가 할당된다.



End of Example

04 가상 기억 장치

□ 연관 기억 장치를 이용한 기억 장치 매핑

- 앞 방법에서 기억 장치 매핑표는 기억 장치의 이용상 비효율적이다.
- 이를 위해 연관 기억 장치를 사용해 기억 장치 매핑표의 워드 수와 주기억 장치의 블록 수를 같게 한다.
- 가상 주소는 인자 레지스터로 전송되며, 키 레지스터의 마스크 값에 따라 이 레지스터의 페이지 번호 비트는 연관 기억 장치의 모든 페이지 번호 부분과 비교된다.
- 매치가 일어나지 않으면 운영체제에 의해 요구되는 페이지가 보조 기억 장치에서 주기억 장치로 옮겨진다.

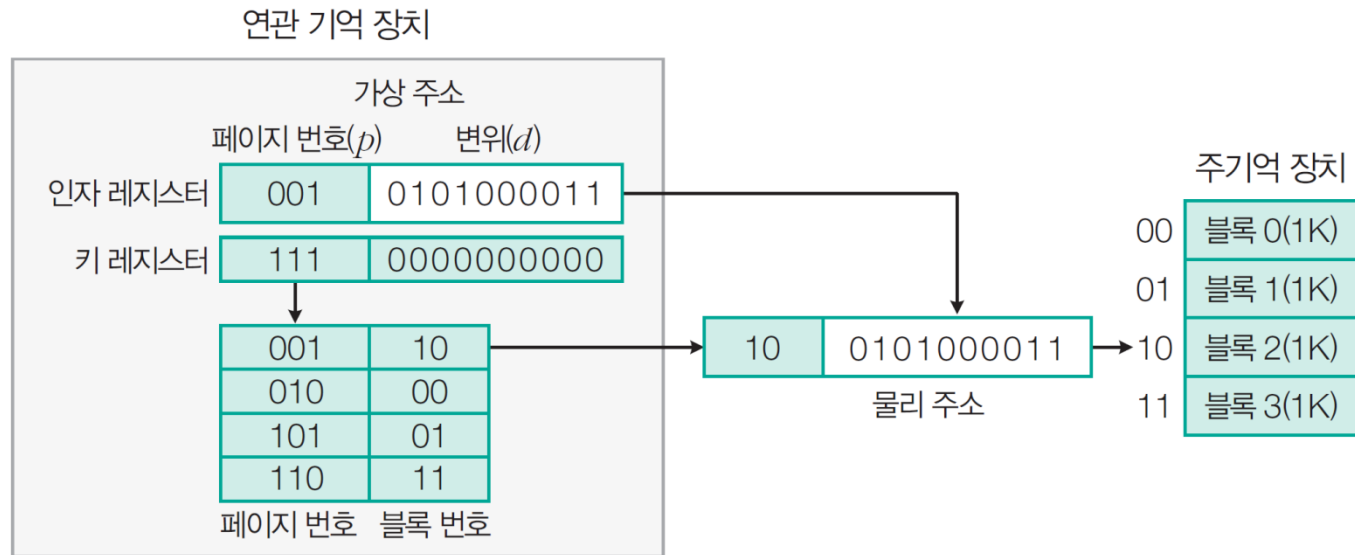


그림 6-39 연관 기억 장치를 이용한 매핑표 구조

04 가상 기억 장치

□ 세그먼트에 의한 매핑

- 고정 길이의 페이지가 아니라 가변 길이의 **세그먼트**로 매핑한다.
- 세그먼트로 된 프로그램에 의해 지정되는 주소를 논리 주소라고 한다.
- 프로그래머가 프로그램을 세그먼트화하고, 다시 시스템이 각 세그먼트를 페이지화한다.
- 세그먼트의 논리 주소에서 물리 주소로 매핑은 기억 장치 매핑표와 비슷한 세그먼트 테이블을 참조해서 이루어진다.
- 속도 향상을 위해 세그먼트 표나 페이지 표를 접근 시간이 빠른 연관 기억 장치에 저장한다.

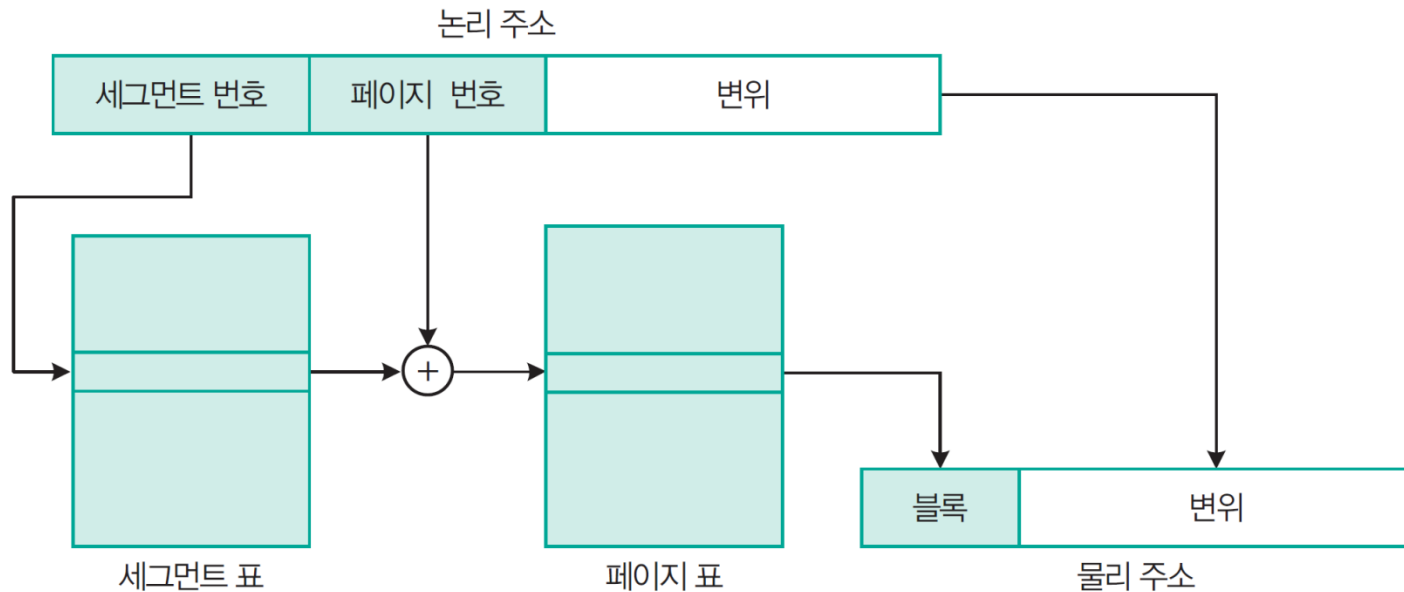
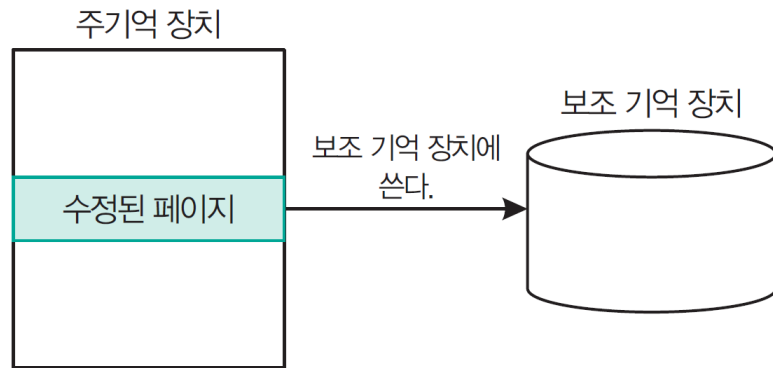


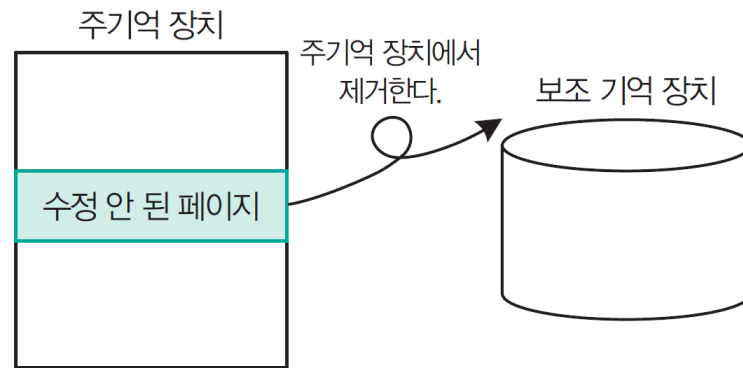
그림 6-40 세그먼트에 의한 매핑에서 논리 주소를 물리 주소로 매핑

2 페이지 교체 알고리즘

- 참조하고자 하는 페이지가 주기억 장치에 없을 경우 **페이지 오류**(page fault)라 하고, 이 조건이 발생하면 요구된 페이지가 주기억 장치로 옮겨질 때까지 프로그램 수행이 중단된다.
- 보조 기억 장치에서 주기억 장치로 페이지를 전송하는 것은 입출력 동작이므로 운영체제는 이 일을 I/O 프로세서에 맡긴다.
- 페이지 교체 알고리즘** : 새로운 페이지가 주기억 장치로 전송될 때, 주기억 장치가 꽉 차 있으면 제거할 페이지를 선택(FIFO, LRU, LFU 알고리즘 등)



(a) 제거할 페이지가 수정된 경우



(b) 제거할 페이지가 수정되지 않은 경우

그림 6-41 페이지 교체할 때 동작

04 가상 기억 장치

- 다음 예를 통해 각 알고리즘을 살펴보자.

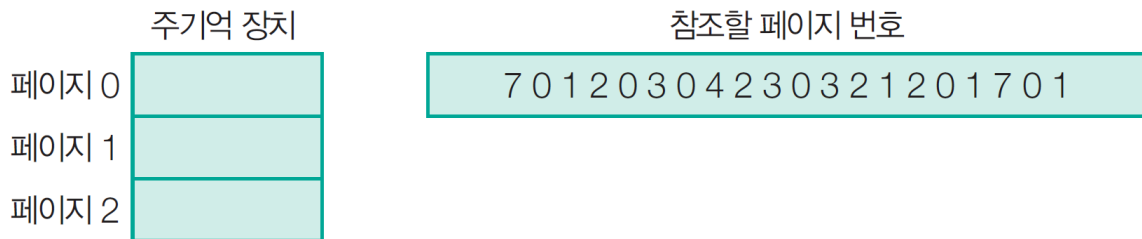


그림 6-42 페이지 교체 알고리즘 동작 예

❖ FIFO(First-In First Out) 알고리즘

- 주 기억 장치에 가장 오래 있었던 페이지를 교체
- 구현하기가 쉽다는 장점이 있으나, 어떤 상황에서는 페이지가 너무 자주 교체되는 단점이 있다.

7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
7	7	7	2	2	2	2	4	4	4	0	0	0	0	0	0	0	7	7	7
	0	0	0	0	3	3	3	2	2	2	2	2	1	1	1	1	1	0	0
		1	1	1	1	0	0	0	3	3	3	3	3	2	2	2	2	2	1

페이지 폴트 15회 발생

04 가상 기억 장치

❖ LRU(Least Recently Used) 알고리즘

- 최근까지 가장 오랫동안 사용되지 않았던 페이지를 선택하여 제거하는 방법

7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
7	7	7	2	2	2	2	4	4	4	0	0	0	1	1	1	1	1	1	1
	0	0	0	0	0	0	0	0	3	3	3	3	3	3	0	0	0	0	0
		1	1	1	3	3	3	2	2	2	2	2	2	2	2	2	7	7	7

페이지 폴트 12회 발생

❖ LFU(Least Frequently Used) 알고리즘

- 사용 빈도가 가장 낮은 페이지를 선택해서 제거하는 방법

7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
7	7	7	2	2	2	2	4	4	3	3	3	3	3	3	3	3	3	3	3
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1	1	1	3	3	3	2	2	2	2	2	1	2	2	1	7	7	1

페이지 폴트 13회 발생

❖ NUR(Not Used Recently) 알고리즘

- LRU와 비슷한 알고리즘으로 참조 비트와 변형 비트를 사용하여 최근에 사용하지 않은 페이지를 교체하는 방법

표 6-10 참조 비트와 변형 비트에 따른 교체 우선순위

참조 비트	변형 비트	교체 순서
0	0	1
0	1	2
1	0	3
1	1	4

04 가상 기억 장치

❖ 캐시 메모리, 주기억 장치, 가상 기억 장치 간의 데이터 이동

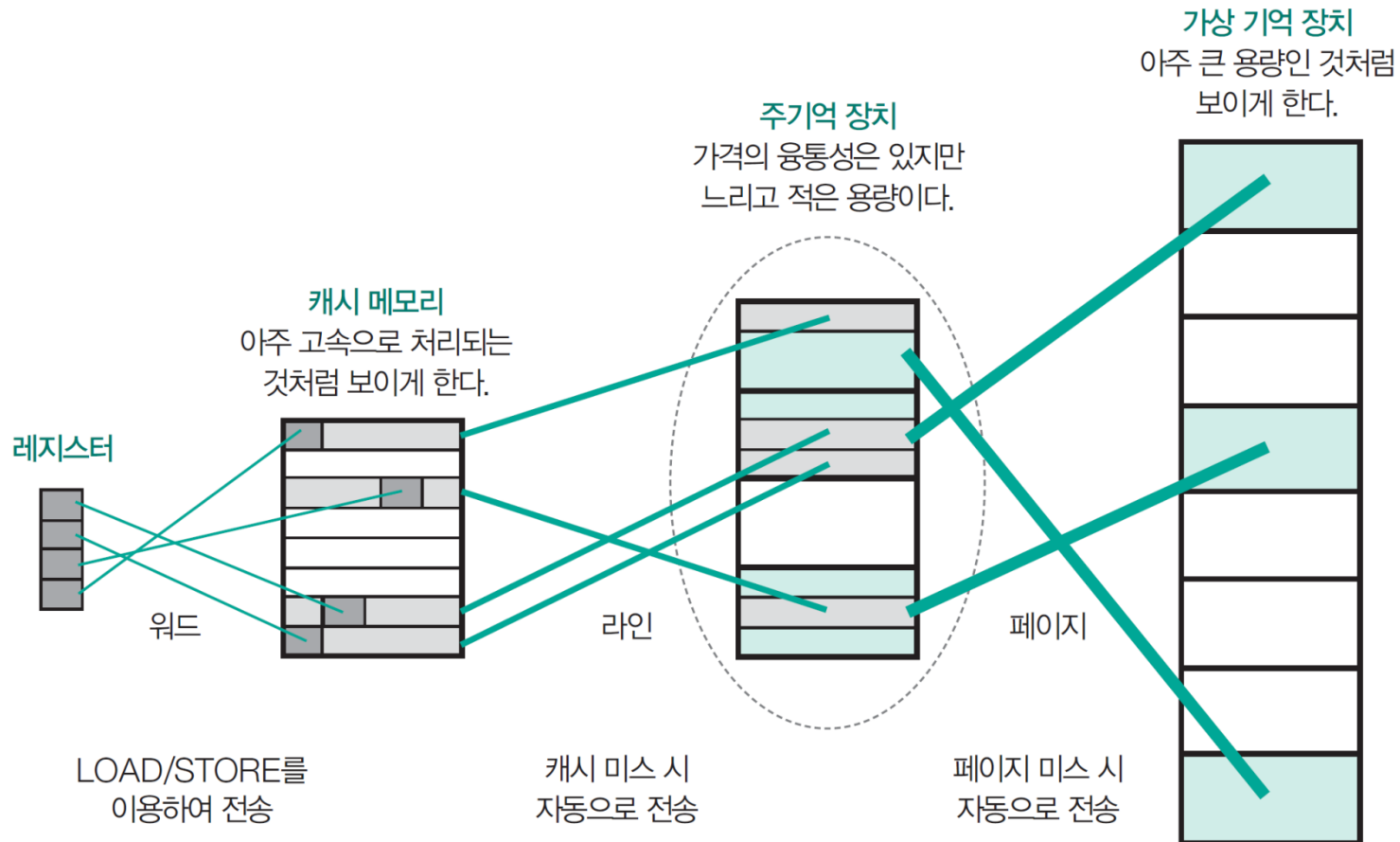


그림 6-43 메모리 계층 구조에서의 데이터 이동

05 연관 기억 장치

- **연관 기억 장치**(associative memory)는 메모리에 저장된 내용의 일부분을 이용해 원하는 정보가 저장된 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 비교 회로 추가로 비싸져 탐색 시간이 중요하고 빠른 처리에만 주로 사용된다.

❖ 연관 기억 장치의 구조

- 인자 레지스터는 n 비트, 키 레지스터는 n 비트, 매치 레지스터는 m 비트로 구성된다.
- 인자 레지스터에는 검색하려는 정보를 저장하고, 키 레지스터는 인자 레지스터의 특정 영역에 대한 마스크(mask)를 제공한다.

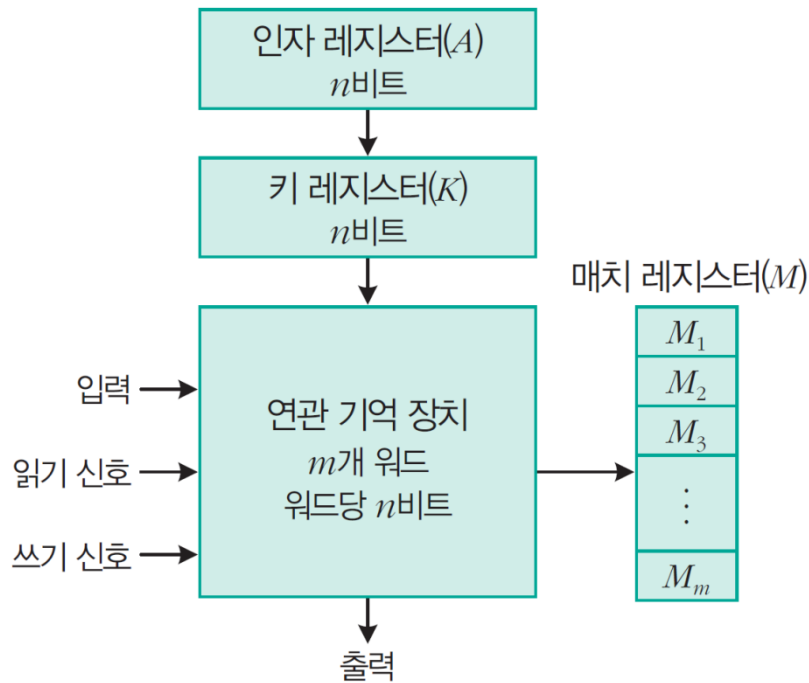


그림 6-44 연관 기억 장치의 구조

인자 레지스터(A)	101	11110	
키 레지스터(K)	111	00000	
워드 1	101	00100	일치 $M_1=1$
워드 2	111	11011	불일치 $M_2=0$
워드 3	010	01101	불일치 $M_3=0$
...
워드 m	100	11010	불일치 $M_m=0$

워드 1이 101이므로 매치 레지스터의 비트 $M_1=1$ 이다.

05 연관 기억 장치

❖ 메모리 배열과 외부 레지스터의 관계

- C_{ij} : i 워드 내의 j 번째 셀
- 인자 레지스터 A_j 비트는 $K_j=1$ 이라면 j 번째 칸의 모든 비트와 비교된다.
- 마스크되지 않은 모든 인자의 비트와 i 워드 비트들의 매치가 이루어지면 매치 레지스터의 대응하는 비트 $M_i=1$ 이 된다.

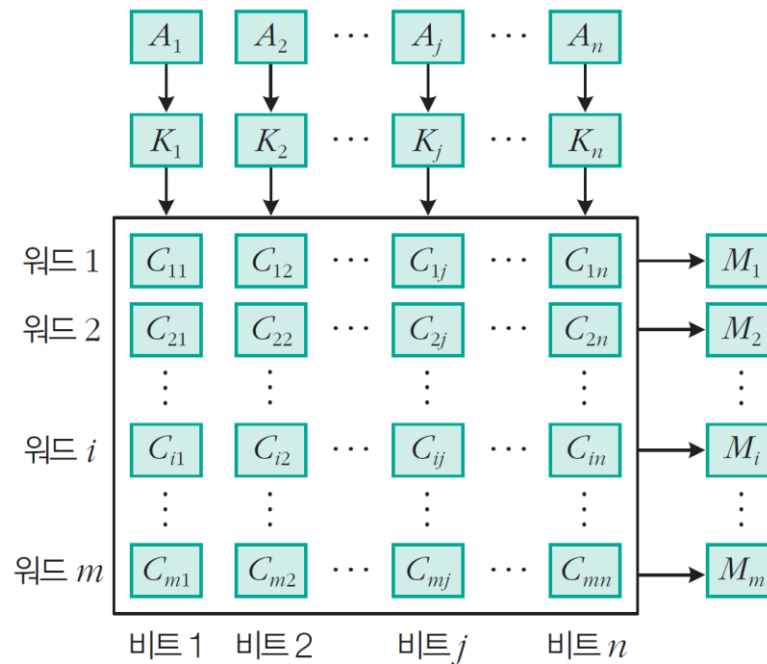


그림 6-45 워드당 n 개 셀을 가진 워드 m 개의 연관 기억 장치

Summary

- 가상 기억 장치의 필요성과 매핑 방법 이해
- 연관 기억 장치의 동작 원리 이해