

# 컴퓨터 구조

---

1

## 기억장치 (1) (제 9주 차)

서울사이버대학교

오 창 환

# 학습 목표

---

2

- 기억장치 시스템의 특성들, 기억장치 계층 등을 설명할 수 있다.
- 반도체 기억장치를 설명할 수 있다.
- 기억장치 모듈의 설계를 설명할 수 있다.

# 학습 내용

---

3

- 기억장치 시스템의 특성들, 기억장치 계층
- 반도체 기억장치
- 기억장치 모듈의 설계

# 기억장치 시스템의 특성들 (1)

4

- 컴퓨터시스템은 프로그램과 데이터를 저장하기 위하여 기억장치를 가지고 있는데, CPU가 어떤 정보를 기억장치에 쓰거나 기억장치로부터 읽는 동작을 액세스(access)한다고 말함.
- 기억장치의 액세스 유형은 일반적으로 다음과 같이 분류됨.
  - \* 순차적 액세스(sequential access) : 자기 테이프 저장장치가 이 방식을 이용하는데, 저장되는 모든 정보는 테이프의 처음 위치에서 시작하여 연속적으로 위치하게 됨. 테이프 내 임의의 위치에 저장된 정보를 읽기 위해서는 그 위치에 도달할 때까지 앞부분의 테이프를 회전시켜야 하기 때문에 정보가 저장된 위치에 따라 액세스 시간이 크게 달라짐.
  - \* 직접 액세스(direct access) : 읽기/쓰기 메커니즘이 각 레코드(혹은 블록)의 근처로 직접 이동한 후에 순차적 검색을 통해 최종 위치에 도달함.  
이 방식에서도 액세스 시간은 정보의 저장 위치에 따라 가변적이며, 자기 디스크와 CD-ROM이 직접 액세스 방식을 이용하는 저장장치임.

## 기억장치 시스템의 특성들 (2)

5

\* 임의 액세스(random access) : 기억장치 내의 모든 저장 위치들은 고유의 주소를 가지고 있으며, 별도의 읽기/쓰기 메커니즘을 가지고 있으므로 어떤 위치든 임의로 선택될 수 있고, 직접 주소 지정되고 액세스될 수 있음. 결과적으로 기억장치 내의 어떤 위치를 액세스하든 그에 걸리는 시간이 항상 일정함.

반도체 기억장치들이 임의 액세스 방식을 이용함.

\* 연관 액세스(associative access) : 임의 액세스의 변형으로서, 각 기억장소에는 키(key)값에 해당하는 비트들과 데이터가 함께 저장되어 있음. 액세스 요구에는 주소 대신에 원하는 비트 패턴이 포함되어 있는데, 그 비트들과 각 기억장소의 키 비트들을 비교하여 일치하는 기억장소의 데이터가 읽혀져 출력됨.

키 값의 비교에 걸리는 시간은 기억장소의 위치에 관계없이 일정하지만, 입력 비트 패턴과 키 값들을 한 개씩 순차적으로 비교하면 시간이 많이 걸리기 때문에, 모든 기억장소의 키 값들과 동시에 비교할 수 있는 하드웨어를 포함하고 있음.

이 유형의 기억장치는 회로가 복잡하고 가격이 매우 높기 때문에 특수한 용도로만 사용됨.

# 기억장치 시스템의 특성들 (3)

6

- 기억장치 시스템을 설계하는 데 있어서 고려해야 할 주요 특성은 용량(capacity)과 액세스 속도임.

- 기억장치에서 용량을 나타내는 단위는 바이트(byte: 1바이트 = 8비트) 혹은 단어(word)임.

일반적으로 단어 길이는 8, 16, 32 혹은 64비트인데, 단어의 길이는 CPU가 실행할 명령어의 길이 혹은 내부 연산에서 한 번에 처리할 수 있는 데이터 비트의 수와 같음.

- 기억장치 시스템의 또 다른 주요 특성인 액세스 속도와 관련된 파라미터들을 보면 다음과 같음.

- \* 액세스 시간(access time) : 주소와 읽기/쓰기 신호가 기억장치에 도착하는 순간부터 데이터가 저장되거나 읽혀지는 동작이 완료되는 순간까지의 시간을 말함.

- \* 데이터 전송률(data transfer rate) : 기억장치로부터 초당 읽혀지거나 쓰여질 수 있는 비트 수를 말함. 따라서 이것은  $(1/\text{액세스 시간}) \times (\text{한 번에 읽혀지는 데이터 비트 수})$ 인데, 예를 들어서 어떤 RAM의 액세스 시간이 100ns이고, 한 번에 32비트 씩 읽혀진다면 데이터 전송률은  $(1/100 \times 10^{-9}) \times 32 = 320[\text{Mbits/sec}]$ 가 됨.

# 기억장치 계층 (1)

7

- 컴퓨터 기억장치 설계에 있어서 가장 중요한 요소들은 다음과 같이 세 가지로 요약할 수 있음.
  - \* 용량
  - \* 속도
  - \* 가격
- CPU가 프로그램 코드나 데이터를 인출하고자 할 때, 만약 그것들이 주기억장치에 존재하지 않는다면 보조저장장치로부터 주기억장치로 이동시킨 후에 인출하게 됨. 그런데 주기억장치의 용량이 적다면, 그러한 이동이 빈번히 일어나게 되는데 디스크의 속도는 주기억장치의 속도에 비해 대략 10만 배 정도 느리기 때문에 CPU는 그때마다 오랜 시간을 기다려야 함.  
속도를 높이기 위해서는 주기억장치 용량을 늘려야겠지만 이럴 경우 가격이 올라가는 문제점이 있으므로 기억장치의 설계에서 가격과 용량 및 속도 사이에는 적절한 조정(tradeoff)이 필요함.

## 기억장치 계층 (2)

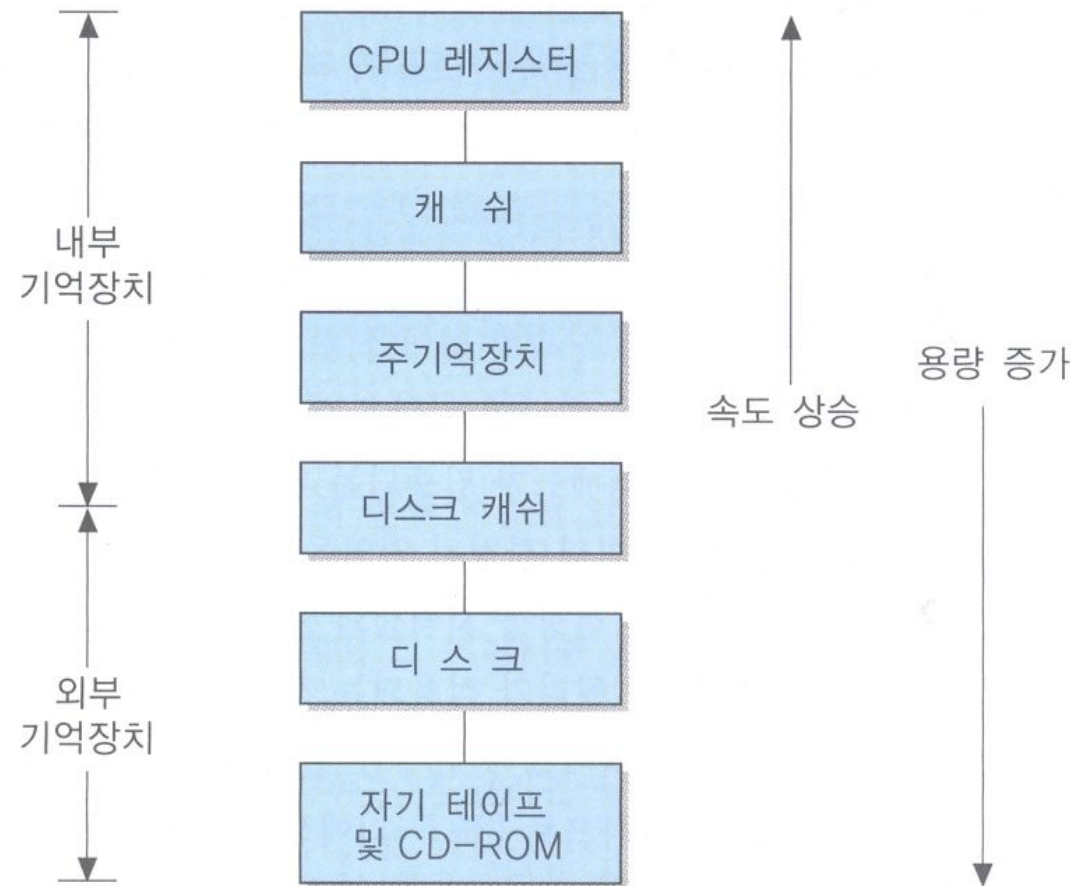
8

- 기억장치 특성들 간에는 다음과 같은 관계들이 성립되고 있음.
  - \* 액세스 속도가 높아질수록, 비트당 가격도 높아짐.
  - \* 용량이 커질수록, 비트당 가격은 낮아짐.
  - \* 용량이 커질수록, 액세스 속도는 낮아짐.
- 아래 그림의 계층 구조에서 최상위 계층의 기억장치, 즉 가장 빠르고, 가장 용량이 적으며, 비트당 가격이 가장 높은 기억장치는 프로세서 내부에 있는 레지스터들임. 어떤 CPU들은 수백 개의 레지스터들을 가지고 있지만, 일반적으로는 수십 개 정도의 레지스터들을 가지고 있음.
- 주기억장치는 액세스 시간이 대략 수 백 ns 정도로서 레지스터 액세스 시간보다 훨씬 더 길기 때문에, 프로그램 실행 중에 CPU가 필요로 하는 데이터가 레지스터에 없는 경우에 주기억장치로부터 읽어오기 위해서는 상당히 긴 시간을 기다려야 함. 이러한 문제를 해결하기 위해 대부분의 컴퓨터시스템에서는 CPU와 주기억장치 사이에 속도가 빠르지만 용량은 작은 캐쉬 기억장치(cache memory)를 설치함.



# 기억장치 계층 (3)

9



• 기억장치 계층

# 기억장치 계층 (4)

10

- 캐쉬는 프로그래머에게는 보이지 않으며, 사실상 CPU에게도 보이지 않음.
- 디스크는 CPU가 프로그램을 수행하는 동안에 빈번히 사용되기 때문에 시스템 성능에 직접적으로 영향을 주는데, 디스크의 액세스 시간은 수십 ms로서, 주기억장치보다 훨씬 더 느림.

따라서 그 속도 격차를 줄이기 위해 주기억장치와 디스크 사이에 디스크 캐쉬(disk cache)라고 하는 반도체 기억장치가 설치되기도 함.

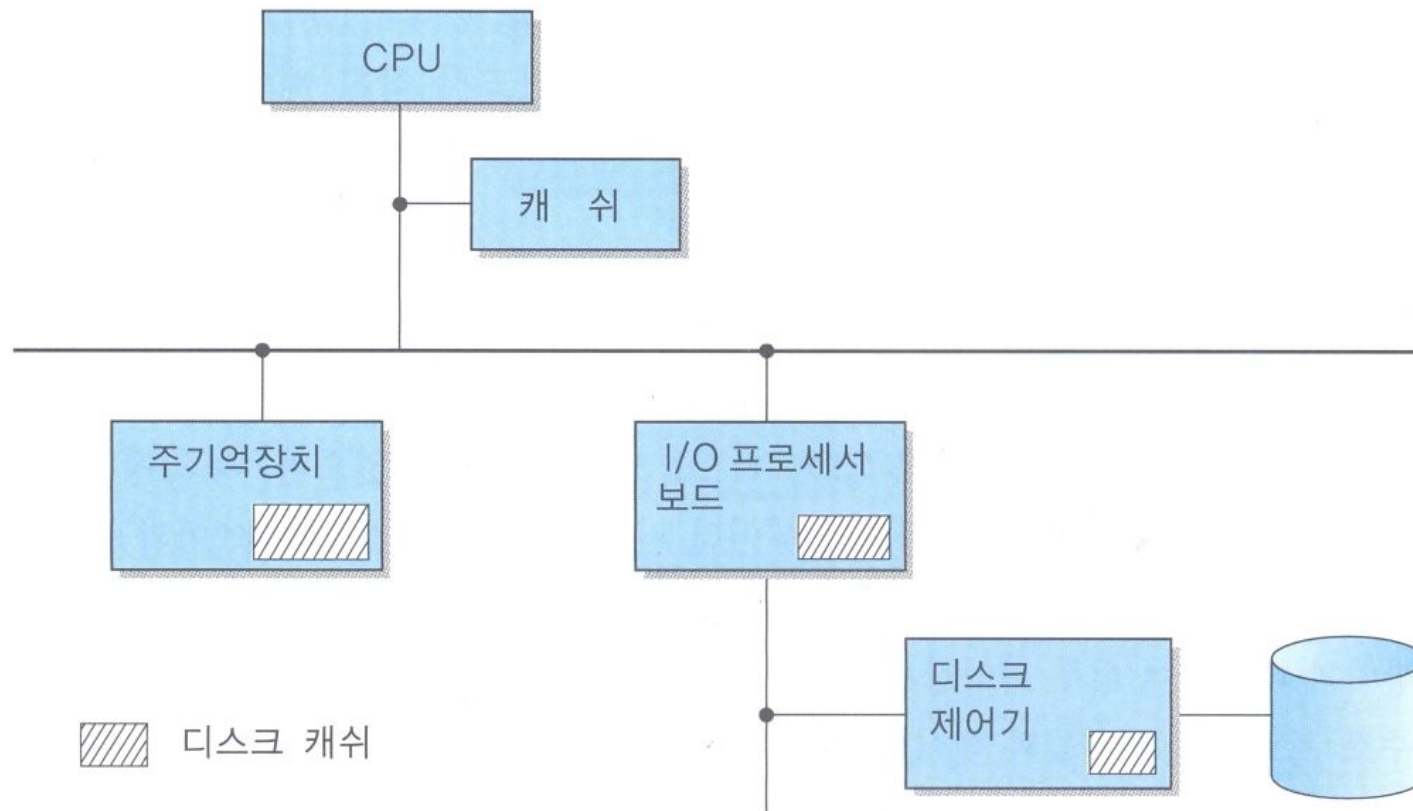
디스크 캐쉬로는 주기억장치의 일부분이 사용되거나, I/O 프로세서 보드 혹은 제어기 보드 상에 별도로 설치함.

디스크 캐쉬는 CPU와 제어기가 모두 직접 액세스할 수 있으므로 내부 및 외부 기억장치에 모두 속한다고 할 수 있음.

- 모든 계층의 기억장치들 사이에는 ‘지역성의 원리’가 적용됨. 즉 하위 계층으로 내려갈수록 용량이 더 커지고 비트당 가격은 떨어지는 반면에, CPU에 의한 액세스 빈도가 더 낮아지므로 전체적으로는 더 적은 비용으로, 더 빠르고 더 큰 용량의 기억장치시스템을 구성할 수 있게 되는 것임.

## 기억장치 계층 (5)

11



- 디스크 캐쉬의 위치

## 2 교시

# 반도체 기억장치 (1)

13

## (1) RAM(Random Access Memory)

- RAM은 임의 액세스 방식을 이용하는 반도체 집적회로 기억장치로서, 이 기억장치에 저장된 각 단어들은 CPU가 전송하는 주소에 의해 지정되어 직접 액세스 됨.

따라서 칩 내의 어느 위치에 있든, 액세스에 걸리는 시간이 같음.

RAM은 데이터를 읽는 것과 쓰는 것이 모두 가능하므로 정확히 표현한다면

RWM(Read Write Memory)라는 명칭이 더 적합하다고 말할 수 있음.

- 아래 그림은 1K x 8비트 용량의 RAM에 대한 블록 선도와 제어 신호들을 보여주고 있는데, 제어 입력 신호인 RD는 읽기(read) 신호이고, WR은 쓰기(write) 신호임.

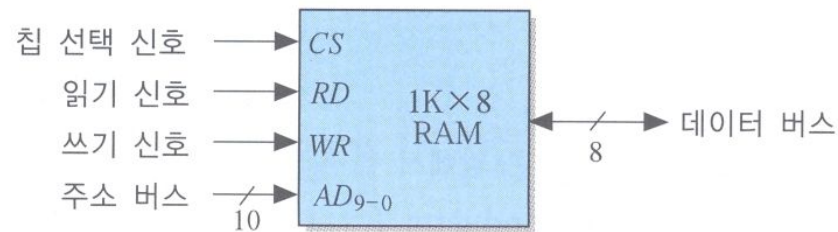
기억장치 시스템이 여러 개의 칩들로 구성되는 경우에는 칩 선택(chip select) 신호인 CS 신호에 의해 칩의 선택 여부가 결정됨.

CS 신호가 1로 활성화되면, 칩은 주소와 제어신호에 따라 적절한 동작을 수행하는데, 만약 RD=1이고 WR=0이면, 주소가 지정하는 기억장소로부터 8비트 데이터가 읽혀서 데이터 버스를 통해 출력됨.

## 반도체 기억장치 (2)

14

만약  $RD=0$ 이고  $WR=1$ 이면, 데이터 버스를 통해 들어오는 8비트 데이터가 주소가 지정하는 기억장소에 저장됨. 그러나, CS 신호가 0인 경우에는 이 칩이 선택되지 않은 것이므로, 제어 신호들이나 데이터 입출력 통로가 모두 전기적으로 단절된 상태, 하이 임피던스(high impedance) 상태가 됨.



(a) RAM의 블록 선도

CS	RD	WR	RAM의 동작
0	X	X	선택되지 않음
1	1	0	읽기 동작
1	0	1	쓰기 동작

(b) 제어 신호들에 따른 RAM의 동작

## 반도체 기억장치 (3)

15

- RAM은 제조 기술에 따라 DRAM(Dynamic RAM)과 SRAM(Static RAM)으로 분류됨.  
DRAM은 커패시터(capacitor)에 전하(charge)를 충전하는 방식으로 데이터를 저장하는 기억 소자들로 이루어져 있는데, 커패시터에 전하가 존재하는지의 여부에 따라 2진수 1과 0으로 구분됨.

커패시터는 근본적으로 방전(discharge)하는 성질이 있기 때문에 DRAM은 데이터의 저장 상태를 유지하기 위해 주기적으로 재충전(refresh)해 주어야 함.

반면에, SRAM에서는 기억소자로서 플립플롭을 이용하기 때문에 데이터가 안정된 상태에서 저장될 수 있으므로 전력이 공급되는 동안에는 재충전 없이도 데이터를 계속 유지할 수 있음.

- DRAM과 SRAM은 모두 휘발성이며, DRAM의 소자(cell)는 SRAM의 소자보다 더 간단하고 더 작으므로 DRAM의 밀도가 더 높으며, 같은 용량의 SRAM보다 가격이 더 싼.

## 반도체 기억장치 (4)

---

16

- 그러나 DRAM에는 재충전 회로가 추가되어야 하는데 기억장치의 용량이 커질수록 재충전 회로의 비용이 차지하는 비중이 상대적으로 줄어들기 때문에 DRAM은 대용량 기억장치에서 많이 채택되고 있음.
- SRAM이 DRAM보다 다소 더 빠르기 때문에 DRAM은 용량이 큰 주기억장치로 많이 사용되고, SRAM은 주로 높은 속도가 필요한 캐쉬 기억장치로 사용됨.



# 반도체 기억장치 (5)

17

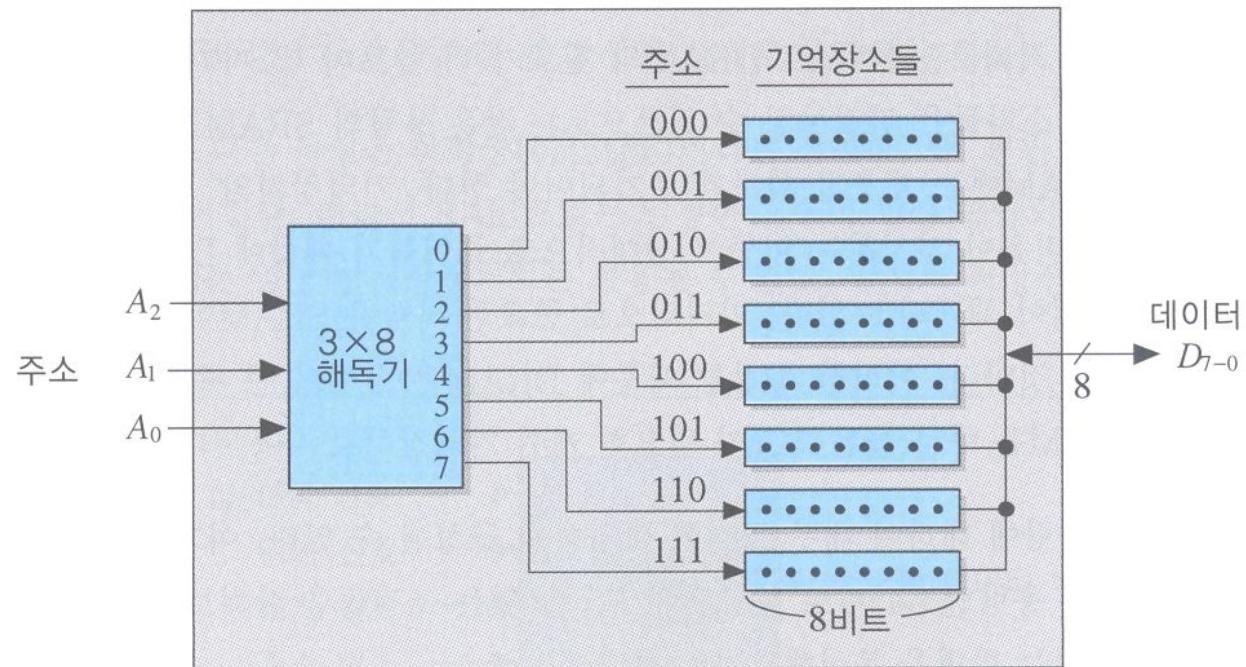
- RAM은 일정한 용량에 대하여 아래와 같이 내부 조직이 다양하게 구성될 수 있음.

## (1) 8 x 8 조직

- 8 x 8 비트 RAM은 아래 그림에서와 같이 8개의 기억장소들로 구성되며, 각 기억장소에는 8비트씩 저장할 수 있으므로  $8 = 2^3$ 의 3승, 즉 세 개의 주소 비트들 ( $A_2, A_1, A_0$ )이 있으면 기억장소들의 주소를 지정할 수 있고, 주소 범위는  $0(000_2)$ 번지부터  $7(111_2)$ 번지까지 됨.
- 3x8 해독기(decoder)는 주소 비트들을 해독하여 8개의 출력들 중의 하나를 활성화하면, 그 출력 선이 연결된 기억장소가 선택되고, 제어 신호에 따라 읽기 혹은 쓰기 동작이 수행됨.  
이와 같은 조직을 가진 RAM의 용량이 1K x 8비트라면, 10개의 주소 비트들이 필요할 것이며 또한 512K x 8비트 용량의 RAM인 경우에는 19비트, 16M x 8비트 용량의 RAM인 경우에는 24비트의 주소가 각각 입력되어야 함.

## 반도체 기억장치 (6)

18



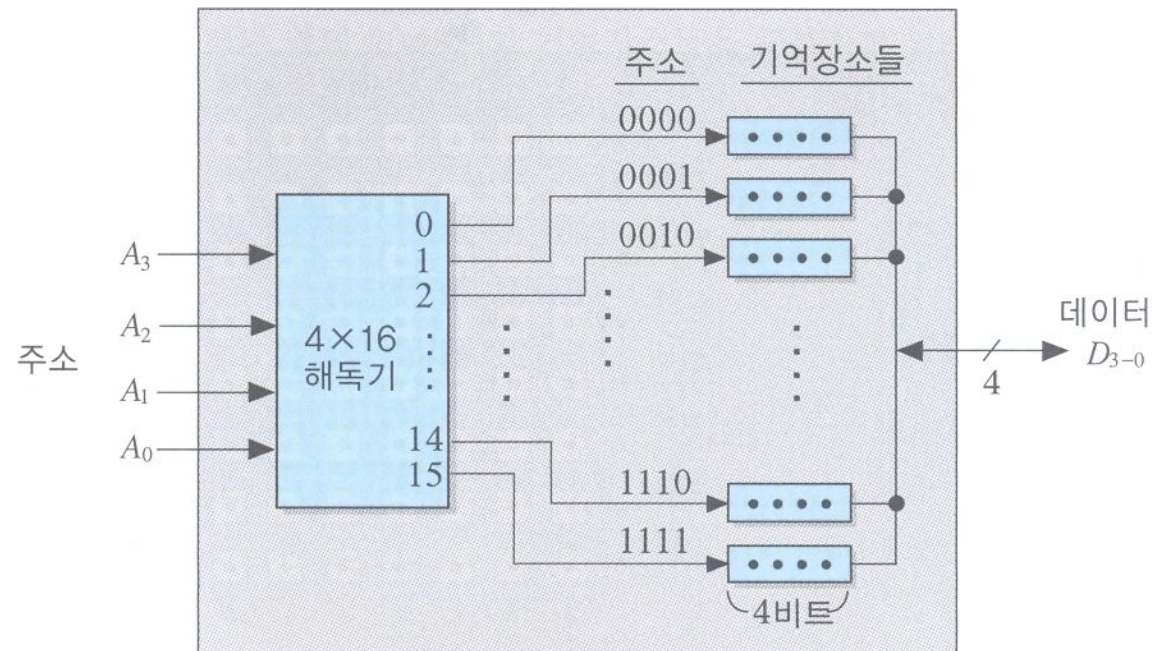
- 8 x 8 조직에서의 주소 지정

# 반도체 기억장치 (7)

19

## (2) 16 x 4 조직

- 16 x 4비트 RAM에서는 16개의 기억장소들이 서로 다른 주소를 할당 받으며, 각 기억장소에는 4비트 데이터가 저장됨. 이 칩에 대한 주소지정을 위해서는  $16 = 2^4$ 의 4승, 즉 4비트 주소( $A_3, A_2, A_1, A_0$ )가 필요하게 되고, 4 x 16 해독기가 16개의 출력신호들 중의 하나를 활성화시킴으로써 각 기억장소가 선택됨.



- 16 x 4 조직에서의 주소 지정

# 반도체 기억장치 (8)

20

## (3) 64 x 1 조직

- 64 x 1 비트 조직을 가진 RAM에서는 8 x 8의 장방형 구조에 위치한 64개의 기억 소자들에 대하여 읽기 및 쓰기가 한 비트씩 이루어지므로 이 칩에는 데이터 입출력 선이 한 개만 있음.

그리고 각 기억장소가 별도의 주소를 가지므로  $64 = 2^6$ 승, 즉 여섯 개의 주소 비트들( $A_5, A_4, A_3, A_2, A_1, A_0$ )이 필요함.

그런데 이러한 조직에서는 일반적으로 주소 비트들이 두 그룹으로 나누어져서 별도로 해독됨.

주소 비트들 중에서 상위 세 비트들은 8개의 행(row)들 중에서 한 개를 선택하고, 하위 세 비트들은 8개의 열(column)들 중에서 한 개를 선택함.

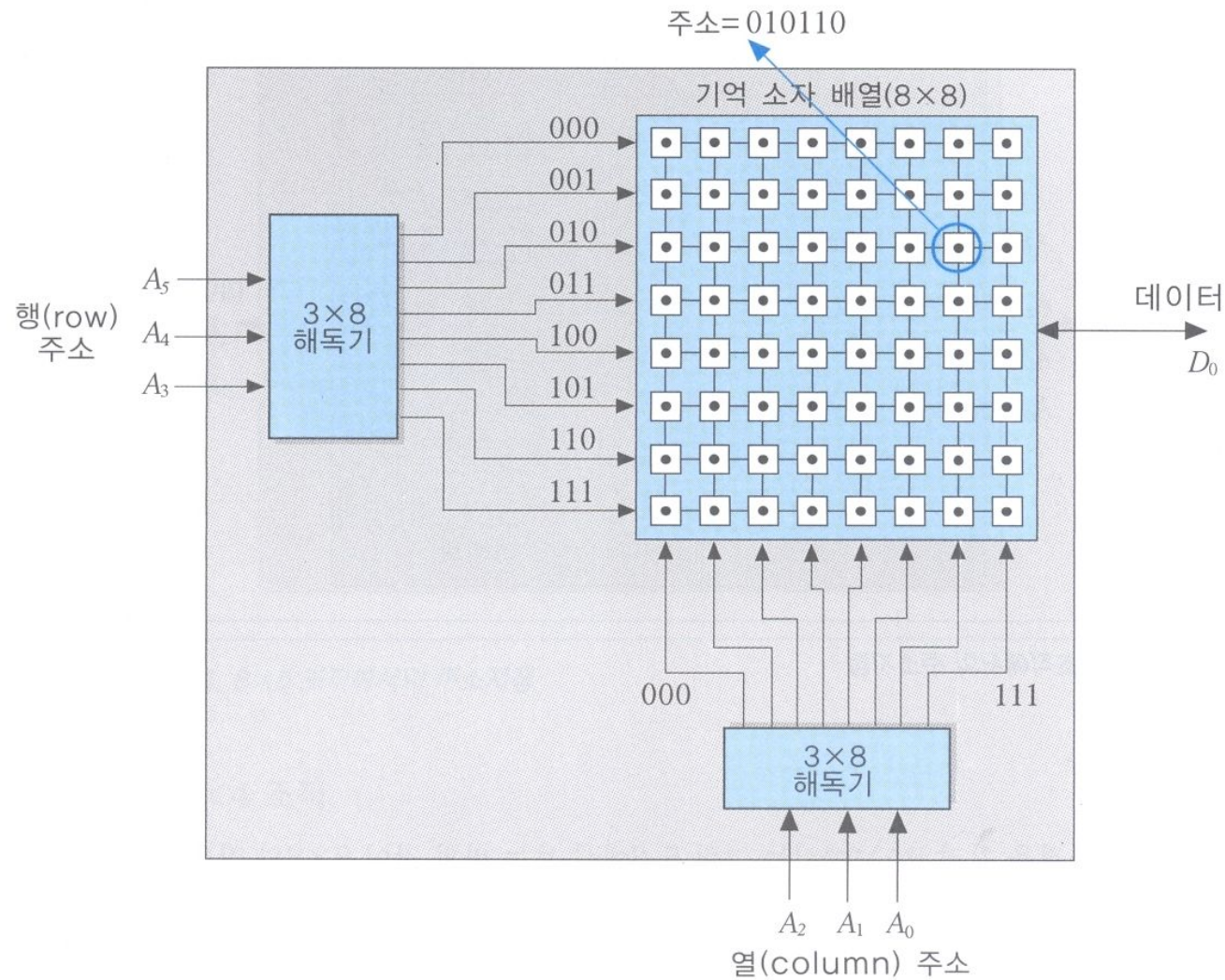
따라서 두 개의 3 x 8 해독기가 필요함.

이 방법으로 주소를 지정한 예를 들면 아래 그림에서 세 번째 행의 일곱 번째 기억 소자의 주소는 0101100이 됨.



# 반도체 기억장치 (9)

21



• 64 x 1 조직에서의 주소 지정

# 반도체 기억장치 (10)

22

## (4) 4M x 4 조직

- 지금까지는 주소지정의 기본 원리를 설명하기 위하여 실제 존재하지 않는 작은 용량의 RAM을 예로서 사용했는데, 이러한 원리에 근거하면 큰 용량의 RAM에 대한 내부 조직과 주소지정 방법도 쉽게 이해할 수 있음.
- 4M x 4 조직을 가진 16M DRAM의 내부 조직을 살펴보면 이 칩의 주소 입력 단에는 주소 버퍼들이 있고, 두 개의 해독기들이 각각 행과 열 주소를 해독함.  
이 조직의 핵심부인 기억 소자 배열은 2048 x 2048 x 4로 표시하는데, 그 의미는 기억장소들이 2048개의 행과 2048개의 열들로 이루어진 장방형 구조로 배열되어 있으며, 각 기억장소에는 4개의 데이터 비트들이 저장된다는 것을 나타냄.

# 반도체 기억장치 (11)

23

- 행과 열의 2048개의 기억장소들을 구분하기 위해서는 각각 11비트씩의 주소가 필요하고, 따라서 전체 22비트의 주소 선들이 있어야 하지만, 실제 주소 선은 11개만을 사용해도 되는데, 이것은 먼저 행의 주소가 주소 버스를 통하여 전송되어 오면 RAS(Row Address Strobe) 신호에 의해 그 주소가 행 주소 버퍼에 래치(latch) 됨.

다음 순간에 열의 주소가 전송되어 오면 CAS(column Address Strobe) 신호에 의해 열 주소 버퍼에 래치 됨.

그러면 두 버퍼에 래치된 주소들이 각각 행 주소 해독기와 열 주소 해독기로 보내져서 기억장소를 선택하게 됨.

- 선택된 기억장소에 대한 데이터 읽기와 쓰기는 4비트씩 이루어지므로 데이터 입출력 선은 4개가 필요함.

# 반도체 기억장치 (12)

---

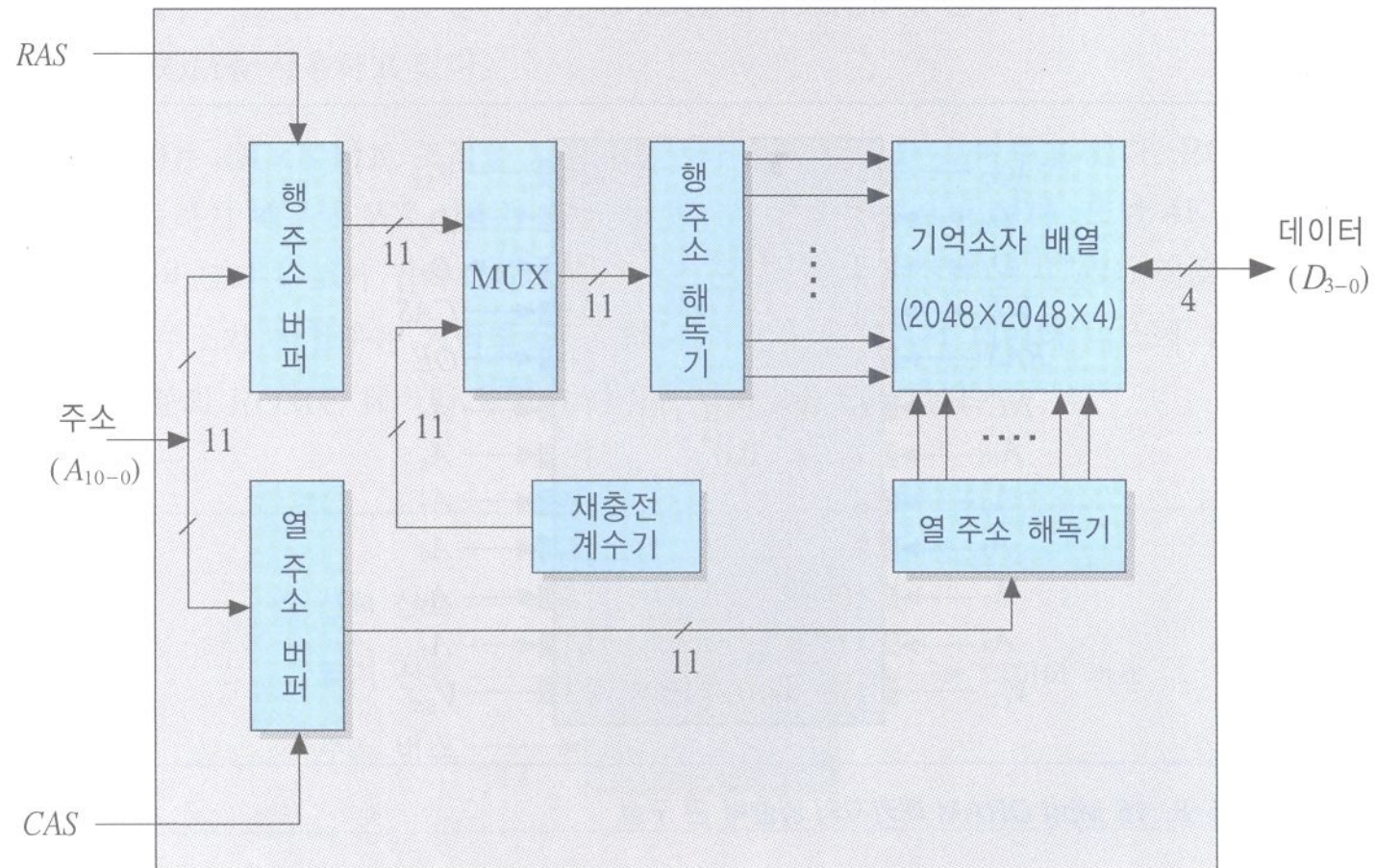
24

- 재충전 계수기(refresh counter)는 0부터 2047까지의 주소를 순서대로 발생하며, 그 주소는 기억장치 액세스가 일어나지 않는 사이클 동안에 멀티플렉서에 의해 선택되어 행 주소 해독기로 보내짐.  
그러면 그 주소가 지정하는 행에 위치한 모든 기억소자들이 동시에 선택되어 재충전 됨.  
재충전 동작은 각 행에 공통으로 접속된 재충전 신호 선으로 전기를 가하여 1이 저장되어 있는 기억 소자들에 전하를 충전함으로써 이루어짐.



# 반도체 기억장치 (13)

25

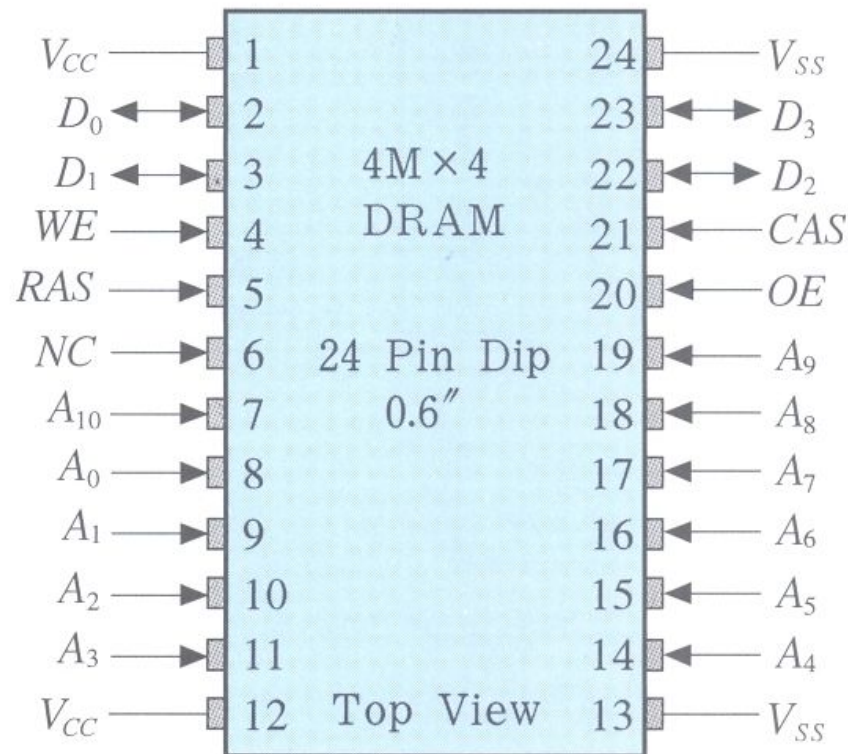


- 16M (4M x4) DRAM의 내부 조직

# 반도체 기억장치 (14)

26

- 아래 그림에서  $V_{CC}$ 는 전원 공급 핀이며,  $V_{SS}$ 는 접지 핀이고  $WE$ 핀은 쓰기 신호 선이며,  $OE$  핀은 출력 활성화 신호, 즉 읽기 신호 선임.  
 $NC$ 는 'Not Connected'의 약어로서 사용되지 않는 핀을 나타냄.



# 반도체 기억장치 (15)

27

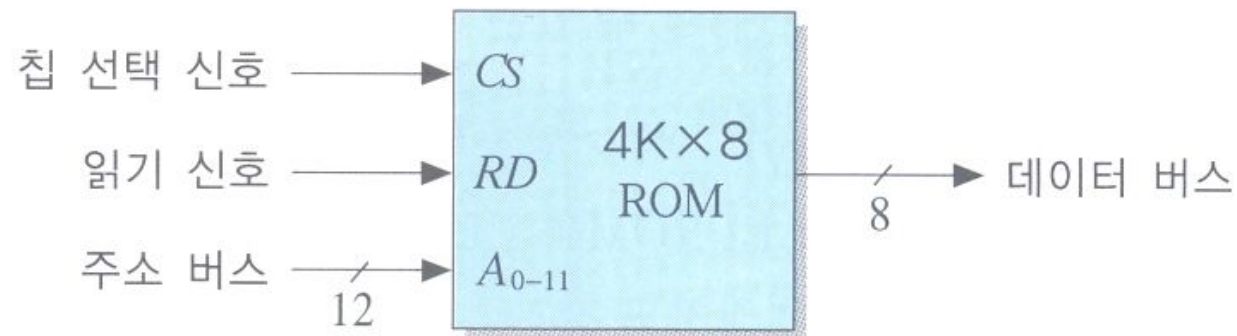
## (2) ROM(Read Only Memory)

- ROM은 명칭이 의미하듯이 내용을 읽는 것만 가능하고, 쓰는 것은 불가능하므로 프로그램이나 변경할 수 없는 데이터를 저장하는 데 사용됨.
- ROM에는 주로 다음과 같은 내용들이 저장됨.
  - \* 시스템 초기화 프로그램 및 진단 프로그램
  - \* 빈번히 사용되는 함수들을 위한 서브루틴들
  - \* 제어 유닛의 마이크로프로그램
- ROM은 전원 공급이 중단되어도 내용을 잃어버리지 않는 영구 저장이 가능한 반도체 기억장치이며, ROM을 사용함으로써 프로그램이나 혹은 데이터를 보조기억장치로부터 매번 이동시킬 필요가 없기 때문에 액세스 시간이 짧아진다는 장점이 있음.  
그러나 쓰기 동작이 불가능하므로 적은 양의 프로그램 코드나 데이터를 저장하는 데만 사용됨.

## 반도체 기억장치 (16)

28

- 아래 그림에서 ROM은 읽기만 가능하기 때문에 WR 신호는 필요 없고 RD 신호만 있으면 되며  $4K = 2^{\text{의 } 12\text{승}}$ 이므로 12비트의 주소가 입력되어야 함.  
칩 선택 신호인 CS와 읽기(RD) 신호가 활성화 되면, 주소가 지정하는 기억장소로부터 데이터가 읽혀져서 데이터 버스에 실리게 됨.



- ROM의 블록 선도

# 반도체 기억장치 (17)

29

- ROM은 제조 과정에서 데이터들을 칩 속에 내장시키고 그 이후에는 내용을 수정할 수 없기 때문에 아래와 같은 종류의 ROM들이 개발되었음.
  - \* PROM(Programmable ROM) : 한 번만 쓸 수 있다는 점에서는 ROM과 같지만 사용자가 전기적으로 데이터를 쓰는 것이 가능함.
  - \* EPROM(Erasable Programmable ROM) : PROM과는 달리 내용을 지울 수도 있기 때문에 사용자가 여러 번 쓸 수 있지만, 쓰기 전에 자외선을 이용하여 내용들을 지워서 초기 상태와 같아지도록 해야 함.
  - \* EEPROM(Electrically Erasable PROM) : 전기적으로 삭제 가능한 EPROM으로서 쓰기 전에 그 이전의 내용을 지울 필요가 없음.  
읽기 동작은 수백 ns 정도 걸리는데,  
쓰기 동작은 바이트당 수백 us 정도 소요되어 훨씬 더 오래 걸림.

# 반도체 기억장치 (18)

30

\* 플래쉬 기억장치(flash memory) : 1980년대 중반에 처음 소개되었으며, 가격과 기능면에서 EPROM과 EEPROM의 중간 정도임.

플래쉬 기억장치의 전체 내용을 지우는 데 걸리는 시간은 수 초 정도인데, 이 속도는 EEPROM보다 훨씬 더 빠른 것임.

칩 전체가 아닌 섹터 단위의 삭제가 가능하다는 특징을 가지고 있으나 바이트 단위의 삭제는 불가능하며, 갱신은 대략 10만 번 정도 허용됨.

---

## 3 교시



# 기억장치 모듈의 설계 (1)

32

- 컴퓨터의 단어 길이가 N이고 기억장치 칩의 데이터 입출력 비트 수가 B일 때, 한 번에 한 단어씩의 데이터 액세스가 가능하도록 하기 위해서는  $N/B$ 개의 칩들로부터 동시에 B 비트씩 액세스 될 수 있도록 해야 함.

예를 들어, 단어 길이가 32비트인 컴퓨터의 기억장치 모듈을 내부 조직이 1K x 8비트인 RAM 칩들을 이용하려면  $32/8 = 4$ 개의 칩들을 병렬로 배열하고, 각 칩에 8비트씩 저장하면 됨.

그리고 32개의 단어 비트들 중에서 그림의 맨 좌측 칩에 최상위 데이터 비트들인  $D_{31} \sim D_{24}$ 를 저장하고, 다음 칩들에 순서대로  $D_{23} \sim D_{16}$ ,  $D_{15} \sim D_8$ , 그리고 마지막 칩에  $D_7 \sim D_0$ 비트들을 저장함.

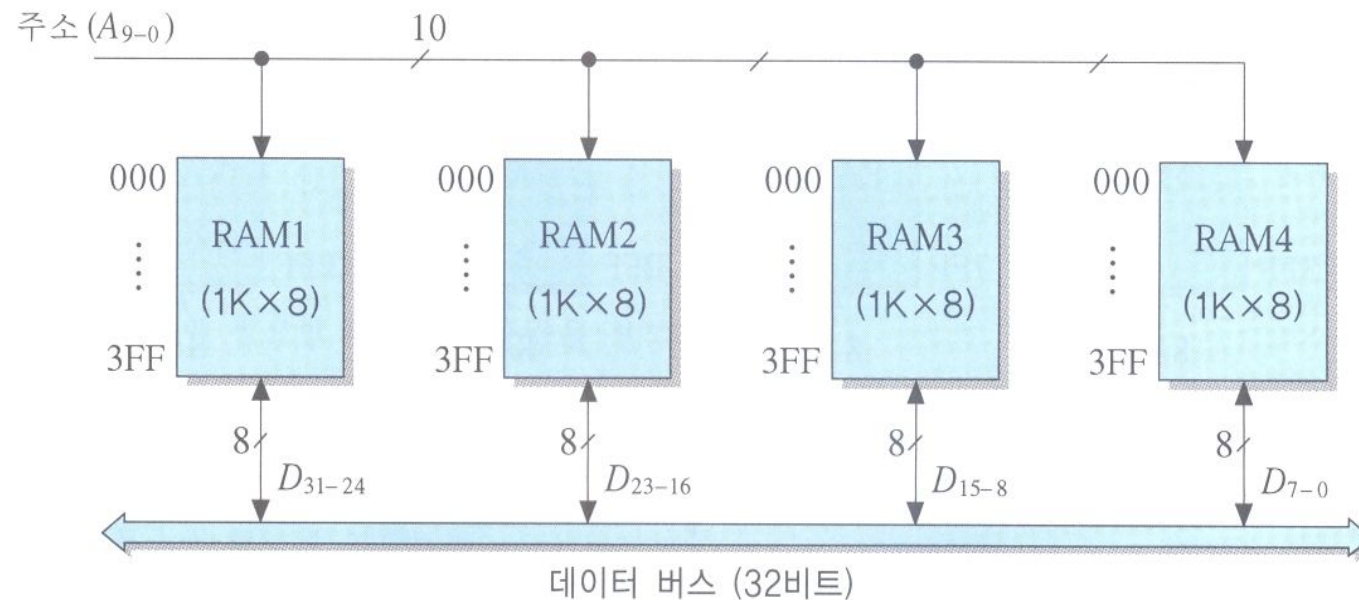
이 경우에 기억장치의 전체 용량은  $(1K \times 8) \times 4 = 32K$ 비트 = 1K 단어가 됨.



## 기억장치 모듈의 설계 (2)

33

- 10개의 주소 비트들( $A_9 \sim A_0$ )이 모든 칩들에 공통으로 연결되기 때문에 칩들의 동일한 위치에 있는 기억장소들은 동일한 주소를 가짐.  
따라서, 어떤 주소가 칩들에 가해지면, 그 주소에 해당하는 기억장소가 네 개의 RAM 칩들에 모두 존재하기 때문에 각 칩 당 8비트씩, 전체 32비트 단어에 대한 쓰기 혹은 읽기 동작이 한 번에 수행될 수 있음.

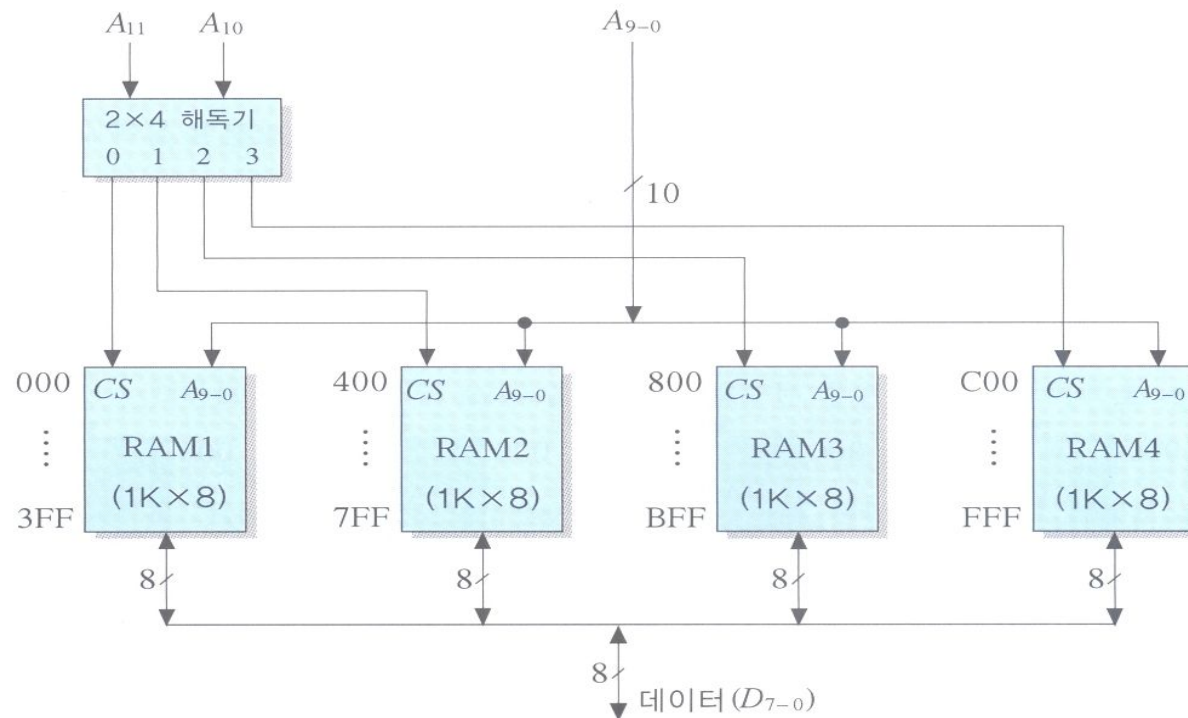


- 1K x 8비트 RAM들을 이용한 1K x 32비트 기억장치 모듈의 설계

## 기억장치 모듈의 설계 (3)

34

- 아래 그림은 1K x 8비트 RAM 칩들을 이용하여 4K x 8비트 기억장치 모듈을 구성하는 방법을 보여주는데, 각 기억장치 주소에 대하여 RAM 칩이 한 개씩만 선택되도록 하고, RAM 칩들은 모두 서로 다른 주소 영역을 가지도록 해야 함. 따라서 네 개의 RAM 칩들을 직렬로 연결하면 됨.



## 기억장치 모듈의 설계 (4)

35

- $4K = 2^{12}$  = 4096개의 기억장치 주소를 구분하기 위해서는 12비트의 주소가 필요하고, 2x4 해독기는 최상위의 두 주소 비트들( $A_{11}$ ,  $A_{10}$ )을 받아서 네 개의 칩 선택( $CS$ ) 신호들을 발생시킴.

RAM1에 해독기의 첫 번째 출력을 연결하고, RAM2에 두 번째 출력, RAM3에 세 번째 출력, 그리고 RAM4에 네 번째 출력을 연결한다면 주소 영역은 아래와 같음.

칩 번호	주소 영역											
	$A_{11}$	$A_{10}$	9	8	7	6	5	4	3	2	1	0
RAM1	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	1	1	1	1	1	1	1	1	1	1
RAM2	0	1	0	0	0	0	0	0	0	0	0	0
	0	1	1	1	1	1	1	1	1	1	1	1
RAM3	1	0	0	0	0	0	0	0	0	0	0	0
	1	0	1	1	1	1	1	1	1	1	1	1
RAM4	1	1	0	0	0	0	0	0	0	0	0	0
	1	1	1	1	1	1	1	1	1	1	1	1

- 각 RAM에 지정되는 주소 영역

# 기억장치 모듈의 설계 (5)

36

- ROM과 RAM들로 구성된 주기억장치를 가진 간단한 마이크로컴퓨터를 설계하는 단계는 아래와 같음.
  - ① 컴퓨터에 필요한 기억장치 용량을 결정함.
  - ② 사용 가능한 칩들을 조사하고, 그들에 대한 주소 표를 작성함.
  - ③ 세부적인 회로를 설계함.
- 여기서 설계할 마이크로컴퓨터는 단어 길이가 8비트이고, 1K 바이트의 RAM과 512바이트의 ROM으로 구성된 주기억장치를 가진다고 가정함.

RAM들의 주소는 0번지부터 시작하고, ROM의 주소는 800번지부터 시작하도록 함.  
그리고 사용할 수 있는 칩들로는 4개의 256 x 8비트 RAM들과 한 개의 1K x 8비트 ROM이라고 가정함.

## 기억장치 모듈의 설계 (6)

37

- 이러한 구성을 위한 기억장치 주소 표는 아래와 같이 작성할 수 있음.

각각의 RAM은 256개의 셀을 가지므로 주소 라인은 8개가 연결되어야 하므로 이들 주소들은 주소 표에서 don't care 값을 가지며 각 RAM의  $A_7 \sim A_0$ 는 공통적으로 연결되어야 하고, 4개의 칩을 선택하기 위한 주소로 2개가 필요하므로 이를 위해  $A_9$ 와  $A_8$ 을 활용함.

ROM은 크기가 1K이므로 주소 라인이 10개가 연결되어야 하며 따라서  $A_9 \sim A_0$ 를 ROM의 주소 라인에 연결시켜야 함에 따라 주소 표 상에서 이들 주소 값은 don't care 값을 가짐.

또한 ROM의 시작 주소가 hexa 800번지이므로 2진수로는 1000 0000 0000이 되며, 이는  $A_{11}$ 과  $A_{10}$ 이 각각 1과 0으로 고정되어야 함을 의미함.

- CPU로부터 발생하는 주소는 모두 16비트이지만, 이 용량의 기억장치 시스템에서는 그 주소들을 모두 사용할 필요는 없음.

# 기억장치 모듈의 설계 (7)

38

- 회로 설계에서 2 x 4 해독기의 출력을 사용하여 4개의 RAM들 중에서 하나를 선택할 수 있게 함.
- 그런데 주소 표를 살펴보면 모든 RAM은 A11과 A10이 각각 항상 0이어야 하고, ROM은 A11 = 1, A10 = 0이어야 하므로 이를 고려하여 두 개의 CS 입력 신호로 활용함. 그런데 RAM과 ROM의 주소 라인에 관한 차이점은 A11 뿐이므로 A11=0이면 RAM이 선택되게 하고 A11=1이면 ROM이 선택되도록 회로를 구성함.

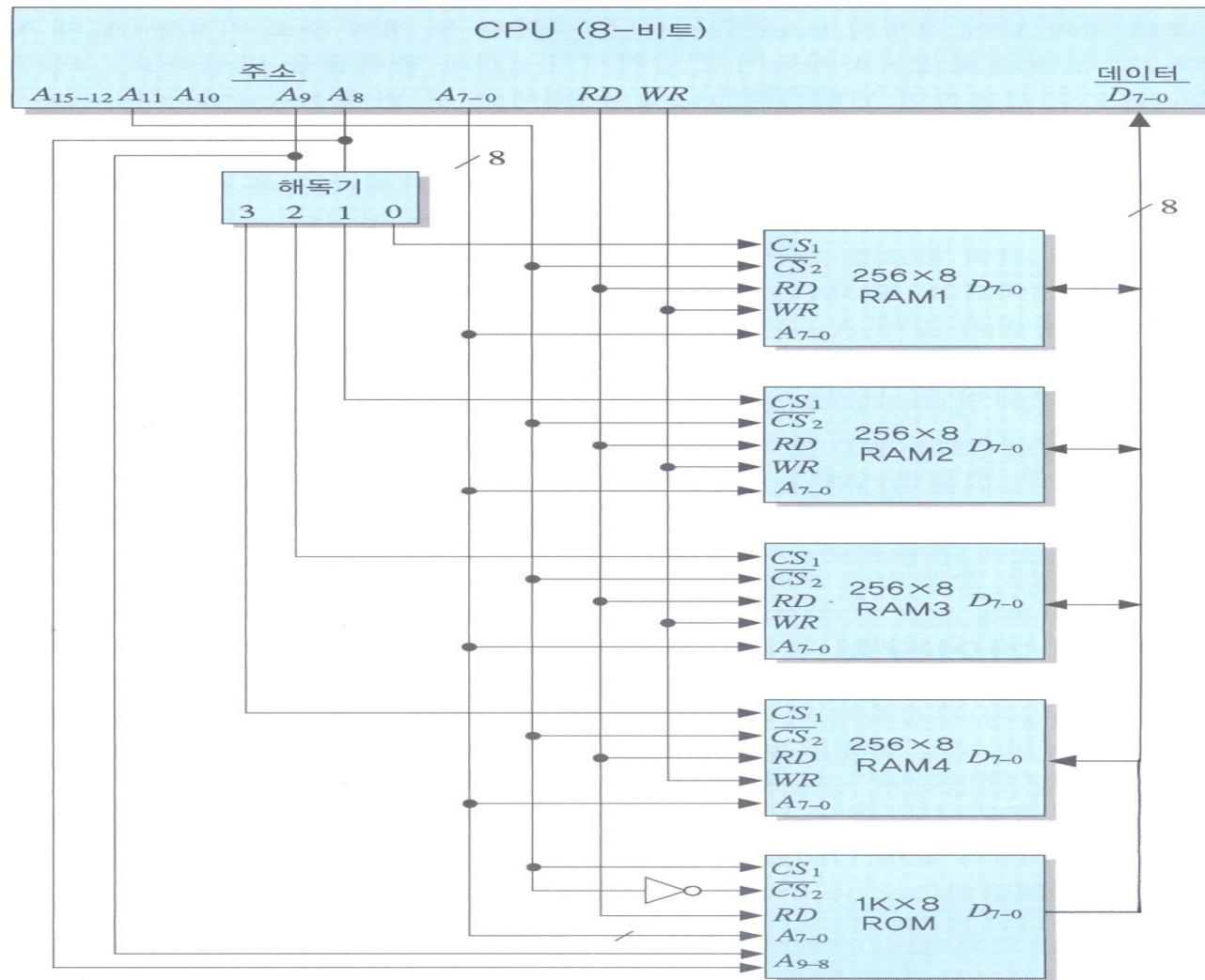
기억장치 칩	주소 영역 (16진수)	주소 비트들											
		A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
RAM1	000H~0FFH	0	0	0	0	x	x	x	x	x	x	x	x
RAM2	100H~1FFH	0	0	0	1	x	x	x	x	x	x	x	x
RAM3	200H~2FFH	0	0	1	0	x	x	x	x	x	x	x	x
RAM4	300H~3FFH	0	0	1	1	x	x	x	x	x	x	x	x
ROM	800H~BFFH	1	0	x	x	x	x	x	x	x	x	x	x

- 기억 장치 주소표



# 기억장치 모듈의 설계 (8)

39



# 셀프 테스트

40

- 기억장치 계층에서 CPU 레지스터와 주기억장치 사이에 놓여 있는 메모리는 무슨 메모리인가?

\* 캐쉬

해설) 캐쉬는 CPU와 주기억장치 사이에 놓여서 주기억장치의 속도 느낌을 보상하는 기능을 함.

- 64 x 4 조직의 RAM은 주소 라인과 데이터 라인이 각각 몇 개 필요한가?

\* 6 개, 4 개

해설) 64는 저장 장소의 개수를 뜻하고 4는 각 저장장소의 데이터 길이를 나타내므로

64는 주소 라인에 해당하여 2의 6승, 즉 6개가 요구되고 데이터 길이는 4비트이므로 4개 선이 요구됨.

- 4K x8 조직의 RAM 4개를 사용하여 16K x 8 비트 크기의 저장장치를 구성하고자 할 때 2 x 4 해독기의 입력으로는 주소 라인 몇 번과 몇 번이 연결되어야 하는가?

단, 최하위 비트의 주소 라인은 0번이다.

\* 13 번, 12 번

해설) 16K의 저장장치에서는 주소 라인이 14개 필요하고 각 RAM에는 주소가 12개씩 입력되어야 하므로 주소 라인이 0부터 시작할 때에 주소 13번과 12번이 해독기의 입력이 되어야 함.



- 컴퓨터 기억장치 설계에 있어서 가장 중요한 요소들은 세 가지, 즉 용량, 속도, 가격 등임.
- 기억장치 계층은 속도가 제일 빠른 순서부터 CPU 레지스터, 캐쉬, 주기억장치, 디스크 캐쉬, 디스크, 자기 테이프 및 CD-ROM 임.
- 모든 계층의 기억장치들 사이에는 지역성의 원리가 적용됨.
- SRAM이 DRAM보다 다소 더 빠르기 때문에 DRAM은 용량이 큰 주기억장치로 많이 사용되고, SRAM은 주로 높은 속도가 필요한 캐쉬 기억장치로 사용됨.
- 8 x 8 비트 RAM은 8개의 기억장소들로 구성되며, 각 기억장소에는 8비트씩 저장할 수 있고  $8 = 2^3$ 의 3승, 즉 세 개의 주소 비트들이 필요함.
- ROM은 명칭이 의미하듯이 내용을 읽는 것만 가능하고, 쓰는 것은 불가능하므로 프로그램이나 변경할 수 없는 데이터를 저장하는 데 사용됨.
- ROM에는 PROM, EPROM, EEPROM, 플래쉬 기억장치 등이 있음.
- ROM과 RAM들로 구성된 주기억장치를 가진 간단한 마이크로컴퓨터를 설계하는 단계는 ① 컴퓨터에 필요한 기억장치 용량을 결정함. ② 사용 가능한 칩들을 조사하고, 그들에 대한 주소 표를 작성함. ③ 세부적인 회로를 설계함.