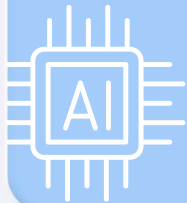




디지털논리회로 [Digital Logic Circuits]

14강.

기억장치와 PLD(1)



컴퓨터과학과 강지훈 교수





제8장 | 기억장치와 PLD

contents

학습 목차

▶ 14 강

01 기억장치 개요

02 RAM

- RAM의 구성
 - RAM의 확장 방법
-

03 ROM

- ROM의 구성
- RAM을 이용한 조합논리회로의 구현



14강. 기억장치와 PLD(1)

➡ 제8장. 기억장치와 PLD

8.1 개요



8.1 기억장치의 개요



• 기억장치

- 처리할 프로그램과 데이터를 기억하거나, 처리한 후의 결과를 기억하는 장치
- 주기억장치와 보조기억장치로 구분
 - 주 기억장치 – RAM, ROM, PLD
 - 보조기억장치 – 자기테이프, 플로피 디스크, 하드 디스크, 레이저 디스크 등



8.1.1 기억장치의 특성



- 기억장치는 2진 기억소자의 모임

- 기억장치에 기억되는 정보는 2진 비트들의 조합
- 기억장치 내부에 이러한 비트들의 조합이 배열되어 데이터를 기억함
- 2진 데이터가 기억장치에 저장되는 단위는 word 단위



8.1.1 기억장치의 특성



- Word(단어)

- Word는 m개의 비트로 구성되며, m은 8, 16, 32, 64 등의 값
- 각 word의 위치는 주소로 식별함
- 기억장치의 성능
 - 기억장치의 정보를 읽거나 쓰는 속도
 - 한 word를 읽어내는데 걸리는 시간 – 접근 시간(access time)



8.1.2 기억장치의 종류



- 기억장치의 기억소자는 반도체 메모리를 이용

- 반도체 메모리

- 기억 소자를 실리콘 칩에 고밀도로 집적화한 것 – 플립플롭, 커패시터, 고정배선 등
 - RAM(Random Access Memory)
 - ROM(Read Only Memory)

- PLD(Programmable Logic Device)

- 전자적 퓨즈로 접속된 내부 논리 게이트를 갖고 있는 집적회로
 - 기억장치로 활용 가능
 - PLA(Programmable Logic Array), PAL(Programmable Array Logic)



14강. 기억장치와 PLD(1)

▶ 제8장. 기억장치와 PLD

8.2 RAM



• RAM(Random Access Memory)

- 임의의 기억장소에 사용자가 작성한 프로그램이나 데이터를 저장하고 필요할 때마다 그 내용을 읽을 수 있는 기억장치
 - 우리가 흔히 메인 메모리라고 부름
- 저장된 값을 자유롭게 변경시킬 수 있음
- 전원이 들어오지 않으면 정보가 손실되어 휘발성 메모리라고 함
- RWM(Read Write Memory)라고도 함
 - RAM이라는 용어는 Random Access라는 주소 접근 방식에 초점을 맞춘 이름
 - ROM(Read Only Memory)과 구별하기 위해 RWM이라는 이름으로 부르기도 함



8.2.1 RAM의 종류



- SRAM(Static RAM)

- 플립플롭으로 구성되어 사용하기 용이
 - 플립플롭 구조라 데이터를 안정적으로, 빠르게 저장하고 읽어올 수 있음
- 접근 시간이 빠름
- 주로 캐시 메모리(cache memory) 에 사용



8.2.1 RAM의 종류



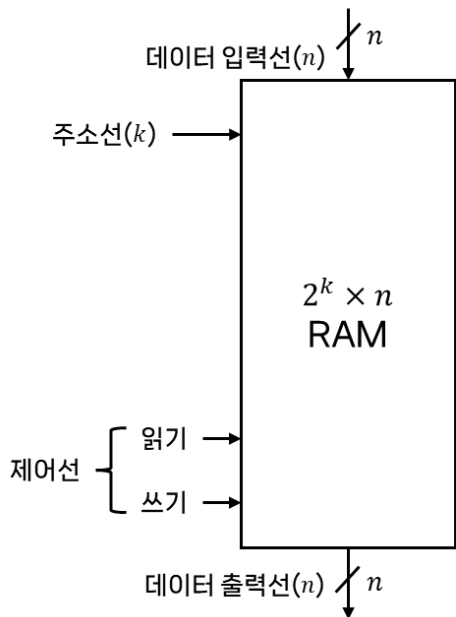
- **DRAM(Dynamic RAM)**

- 커패시터(capacitor)로 충전되는 전하의 형태로 정보 저장
- 주기적으로 커패시터를 충전시키는 동작 필요- refreshing 동작
 - 비트 1 값을 저장할 때 커패시터에 전하를 저장하는데, 시간이 지나면 저장된 전하가 누설됨
 - 그래서 1이 제대로 유지될 수 있도록 전원이 들어온 동안은 일정 주기마다 refresh 해야 함
- 주로 컴퓨터의 주기억장치로 사용

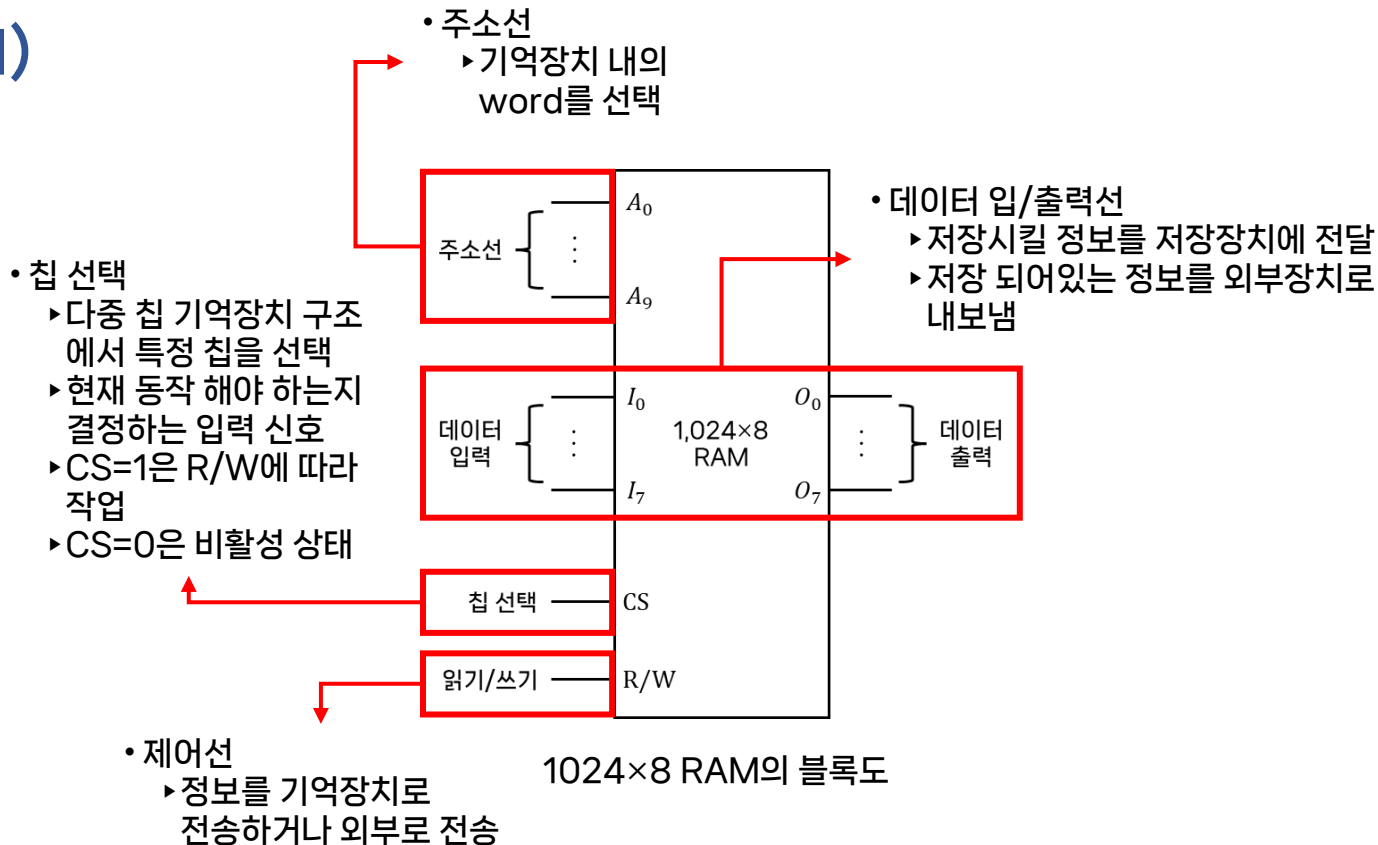


8.2.2 RAM의 구성

• RAM의 블록도(1)

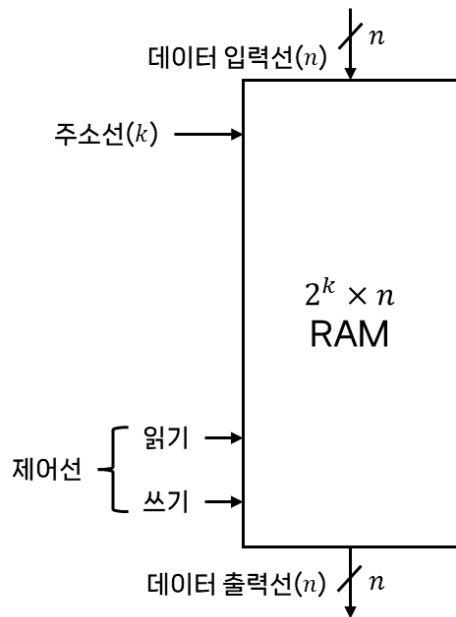


RAM 블록도





• RAM의 블록도(2)



• 기억장치의 규모

- ▶ 저장할 수 있는 word의 수와 각 word를 구성하는 비트 수로 정의

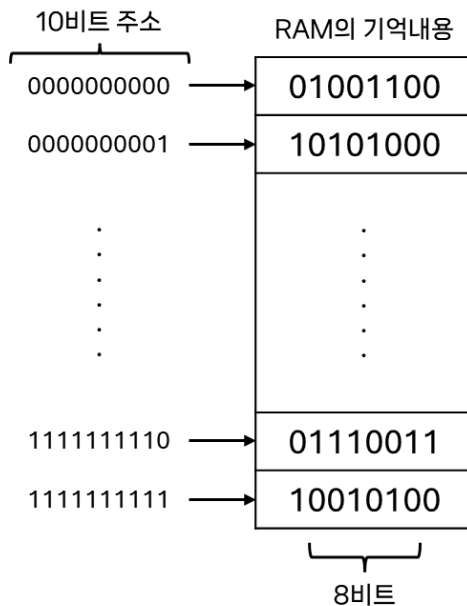
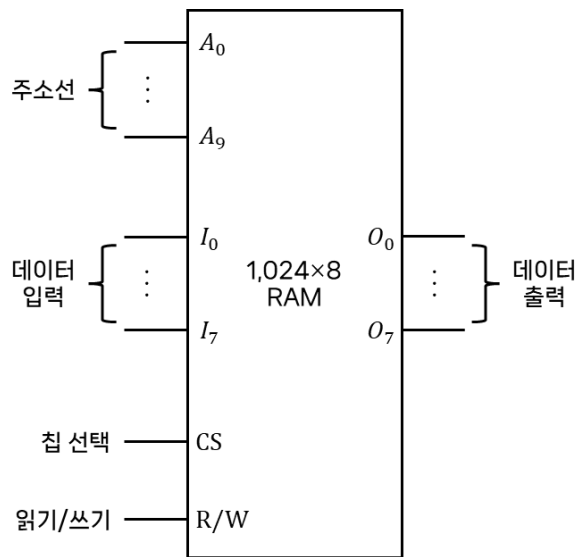
• $2^k \times n$ 의 예를 들면

- ▶ 저장할 수 있는 word 수는 2^k 개, 각 word는 n 비트
- ▶ 주소선으로 기억장치 내의 특정 word 선택 가능
- ▶ 각 word는 그 위치를 식별하기 위한 주소가 할당
 - ▶ 주소선의 수가 k 개라면 주소는 0부터 $2^k - 1$ 까지의 범위를 가짐
 - ▶ $2^k \times n$ 인 기억장치의 특정 word를 선택하려면 k 비트의 주소를 사용



8.2.2 RAM의 구성

• 1024×8 RAM의 블록도



• 1024×8 RAM의 경우

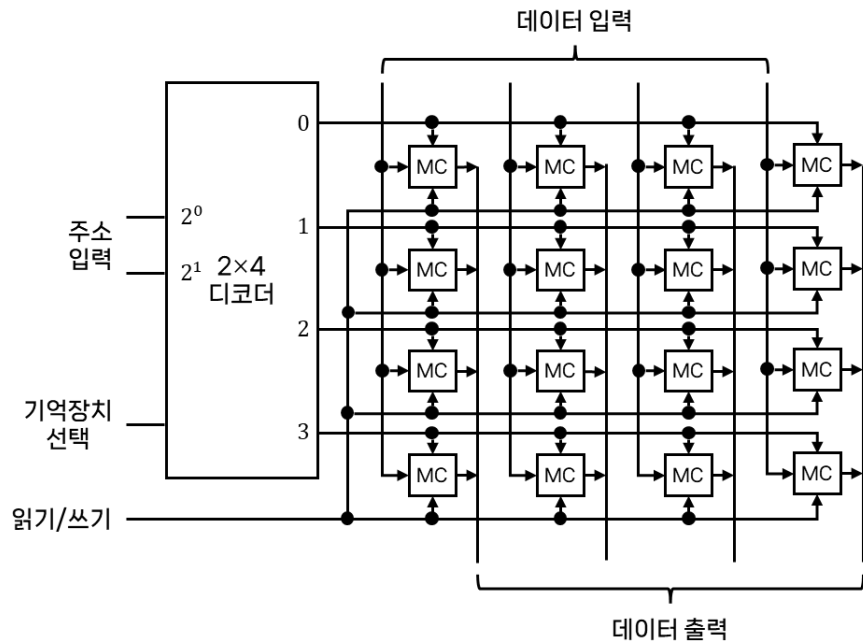
- ▶ 8비트로 구성된 1024개의 word
- ▶ 1024개의 word 선택을 위해서는 10 비트의 주소선이 필요함($2^{10}=1024$)
- ▶ 8비트 word 입출력을 위해서 데이터 입/출력선은 각각 8개가 필요함



8.2.2 RAM의 구성



• 내부 구성(1)



4x4 RAM의 내부 구성도

• 내부 구조

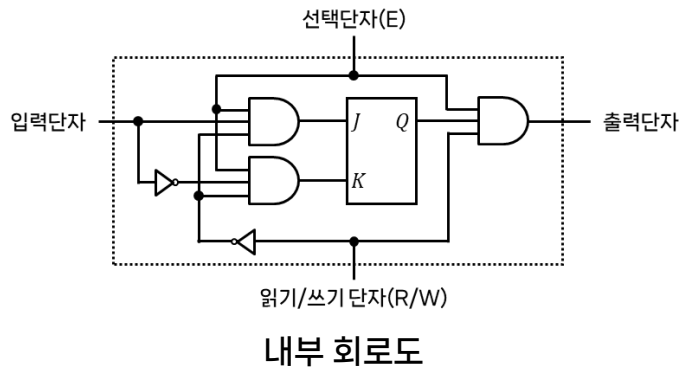
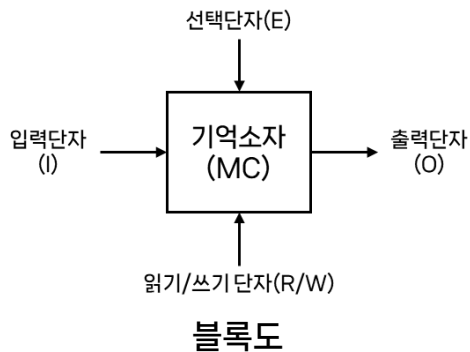
- ▶ 1비트 값을 저장하기 위한 기억소자인 메모리 셀과 word 선택을 위한 디코더로 구성됨
- ▶ 메모리 셀(MC: Memory Cell)
 - ▶ RAM 구성의 기본 단위
 - ▶ 하나의 MC는 1비트 정보를 저장하는 기억소자



8.2.2 RAM의 구성

• 내부 구성(2)

• 메모리 셀의 구조



- ▶ 선택단자(E): 기억소자 선택
- ▶ 읽기/쓰기 단자(R/W): 기억소자에 대한 읽기/쓰기 모드 결정
 - ▶ R/W=1: 읽기 모드 – 플립플롭과 출력단자 사이에 경로를 사용하여 저장된 데이터가 데이터 버스로 출력되게 함
 - ▶ R/W=0: 쓰기 모드 – 입력 단자로 부터 기억소자로의 경로를 사용하여 입력단자의 정보가 기억소자에 기억됨

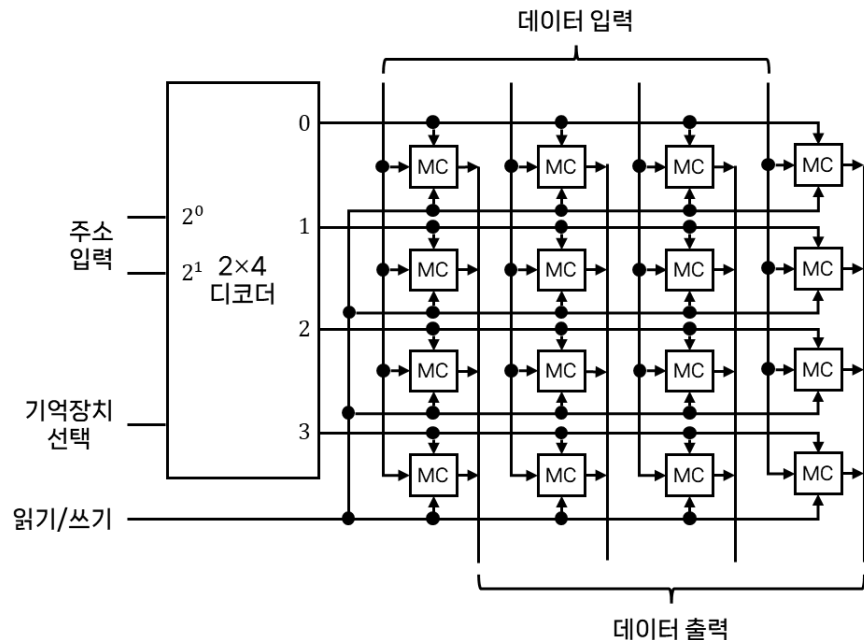




8.2.2 RAM의 구성



• 내부 구성(3)



- 16개의 메모리 셀로 구성

- ▶ 4비트로 구성된 4개의 word

- 기억부, 해독부, 제어부로 구성

- ▶ 기억부

- ▶ 여러 개의 메모리 셀로 이루어진 부분
- ▶ 2차원 행렬 형태로 구성됨

- ▶ 해독부

- ▶ 디코더로 구성
- ▶ 기억장치 선택 입력에 의해 구동

- ▶ 제어부

- ▶ 읽기/쓰기 단자의 입력신호에 따라 데이터 입/출력을 제어하는 부분



8.2.2 RAM의 구성

• RAM의 확장(1)

- RAM을 서로 연결하면 더 큰 용량의 기억장치 구성 가능
- 기억장치 용량: word의 수와 word를 나타내는 비트의 수에 의해 결정
- 기억장치의 용량을 증가시키는 방법
 - 기억장치 내의 word 수를 증가시키는 방법
 - ▶ Word의 수는 주소선의 수와 연관됨
 - ▶ Word를 선택하는 주소의 길이를 증가
 - ▶ 결국 주소선을 표현하는 비트의 수를 증가시키면 더 많은 word를 식별할 수 있음
 - Word의 비트 수를 증가시키는 방법
 - ▶ RAM을 병렬로 연결

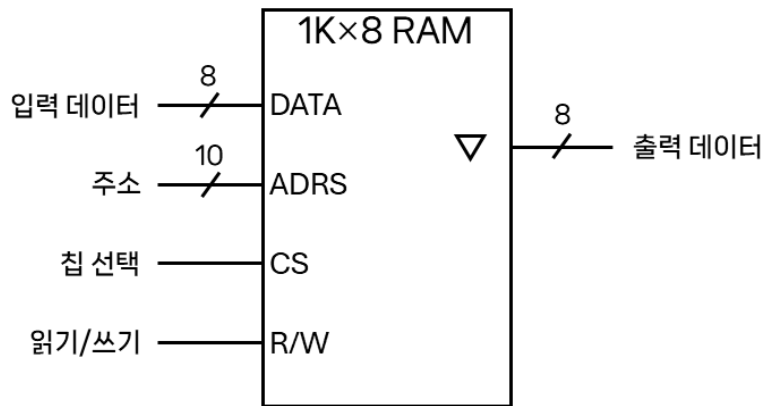


8.2.2 RAM의 구성



• RAM의 확장(2)

• 기억장치 내의 word 수를 증가시키는 방법



• 1K×8 RAM

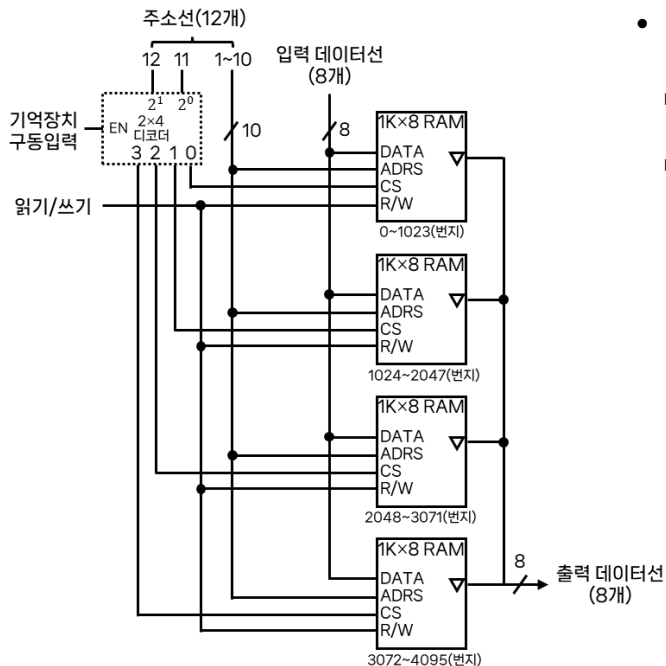
- ▶ 8비트로 표현된 1024개의 word를 가짐
- ▶ 10비트의 주소선과 8비트의 입출력선이 필요
- ▶ Word 수를 증가시키는 방법
 - ▶ 주소 비트를 1비트 증가시키면, 식별할 수 있는 word의 수는 2배 증가
 - ▶ 2개의 RAM을 사용하면, 전체 word의 수는 2배 증가하며, 주소비트를 1비트 증가시키면 됨



8.2.2 RAM의 구성

• Word의 수를 증가시키는 방법의 예

• 4K×8 RAM으로 확장



• 1K×8 RAM을 4K×8 RAM으로 확장

- ▶ 기억장치의 용량을 4배 증가, word를 구성하는 비트 수는 동일
- ▶ 4K의 word를 가진 기억장치를 구성하기 위해서는 12비트의 주소가 필요
 - ▶ 주소의 하위 10개 비트는 4개의 RAM 주소선에 공통으로 연결
 - ▶ 주소의 상위 2개 비트는 2×4 디코더에 연결
 - ▶ 디코더의 출력은 RAM의 칩 선택 단자에 연결
- ▶ 상위비트 2개는 RAM을 선택하고 하위비트 10개는 RAM 내부의 word를 선택함
 - ▶ 상위 비트 2개의 값이 00이면, 첫 번째 RAM을 선택하고 하위비트 10개를 통해 첫 번째 RAM 내부의 word를 선택하는 방법으로 주소를 사용

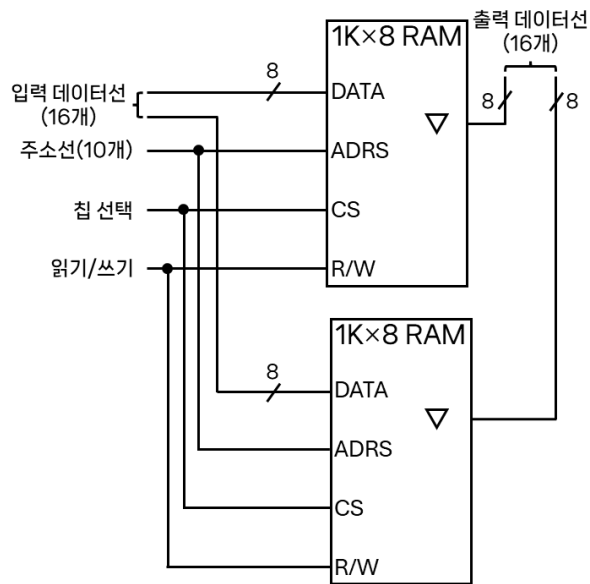




8.2.2 RAM의 구성

• Word를 구성하는 비트 수를 증가시키는 방법의 예

• 1K×16 RAM으로 확장



• 1K×8 RAM을 1K×16 RAM으로 확장

- ▶ 1K×8 RAM 2개를 병렬 연결
 - ▶ RAM에서 저장할 수 있는 word의 개수는 증가하지 않음
 - ▶ 각 word의 비트 수가 16비트, 2배로 증가
- ▶ 16개의 입/출력 선이 필요
 - ▶ 병렬로 연결된 2개의 RAM에 각각 입/출력
- ▶ 주소선과 칩 선택, 읽기/쓰기 입력선
 - ▶ 2개의 RAM에 공통으로 입력
 - ▶ 하나의 word를 2개의 RAM이 절반씩 나눠 갖기 때문에 동일하게 작동해야 함

14강. 기억장치와 PLD(1)

➡ 제8장. 기억장치와 PLD

8.3 ROM



- ROM(Read Only Memory)

- 기록된 데이터에 대해 읽기만 가능한 기억장치
- 일반적으로 내용을 변경할 수 없으며, 전원이 끊겨도 내용이 소멸되지 않는 비 휘발성 기억장치
- ROM의 종류
 - 마스크 ROM – 한번 저장된 데이터를 바꿀 수 없는 ROM
 - PROM(Programmable ROM) – 사용자가 데이터를 써 넣을 수 있는 프로그래밍 가능한 ROM, 기록된 내용은 변경 불가능
 - EPROM(Erasable PROM) – 한번 쓴 내용을 지울 수 있는 ROM



• ROM의 구성

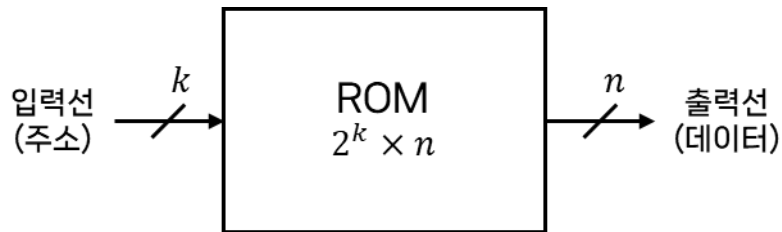
- ROM은 고정된 2진 정보를 저장하는 기억장치
 - 저장될 내용은 설계자에 의해 결정
 - 필요한 내부 연결 패턴을 기억장치 내에 형성하여 구현함
- ROM 칩은 디코더와 여러 개의 OR 게이트로 구성
 - ROM은 특별한 패턴 구조로 프로그래밍 할 수 있는 내부의 전자 퓨즈선으로 구성
 - 특정 형태로 프로그래밍 되면 전원의 유무에 상관없이 그 패턴을 유지



8.3.2 ROM의 구성

• ROM의 블록도

• $2^k \times n$ ROM의 블록도



• k 개의 입력선과 n 개의 출력선으로 구성

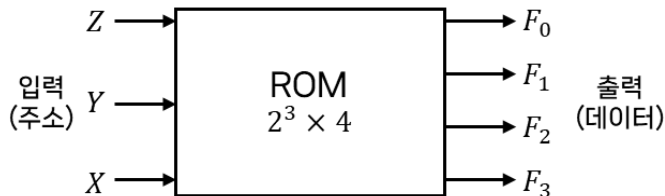
- ▶ 입력선: 기억장치의 주소
- ▶ 출력선: 주소에 의해 선택된 word의 데이터 출력



8.3.2 ROM의 구성



• 8×4 ROM의 블록도와 진리표



X	Y	Z	F ₃	F ₂	F ₁	F ₀
0	0	0	0	1	0	1
0	0	1	0	0	1	1
0	1	0	0	1	0	0
0	1	1	1	0	1	1
1	0	0	1	1	0	0
1	0	1	1	0	0	1
1	1	0	0	1	1	1
1	1	1	1	0	0	0

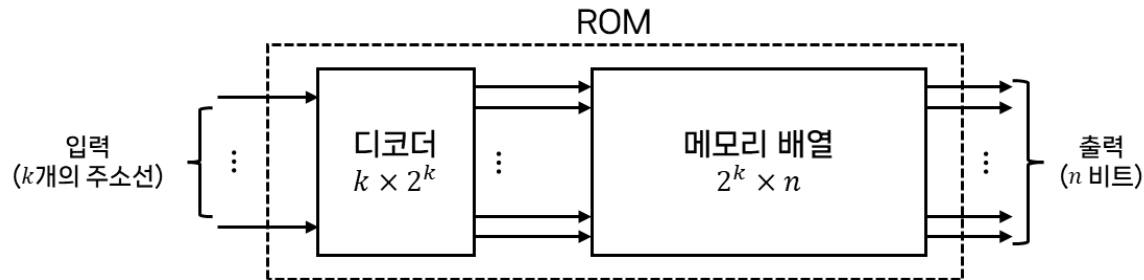
- 8×4 ROM의 블록도
 - ▶ 3비트를 사용한 8가지의 입력, 그에 따른 4개의 출력
- 8×4 ROM의 진리표
 - ▶ 3개의 입력선 조합에 대한 출력값을 표시
 - ▶ 입력선의 조합이 000이면, 출력선은 0101이 됨
 - ▶ 출력값은 ROM에 저장된 한 개의 word



8.3.2 ROM의 구성

• ROM의 내부 구조(1)

• 기본 구조



• 디코더와 OR 게이트로 구성

- ▶ k 개의 입력이 디코더에 입력되면 2^k 개의 디코더 출력중 하나가 1이 됨
- ▶ 디코더의 출력이 메모리 배열 중에서 1개의 word를 선택하고
선택된 word가 출력됨

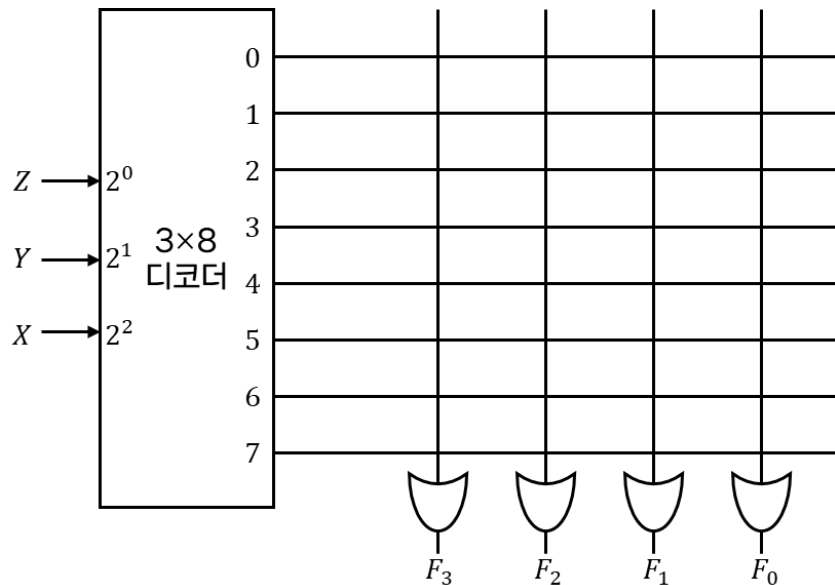


8.3.2 ROM의 구성



• ROM의 내부 구조(2)

• 8×4 ROM의 내부 구조



• 8×4 ROM의 블록도

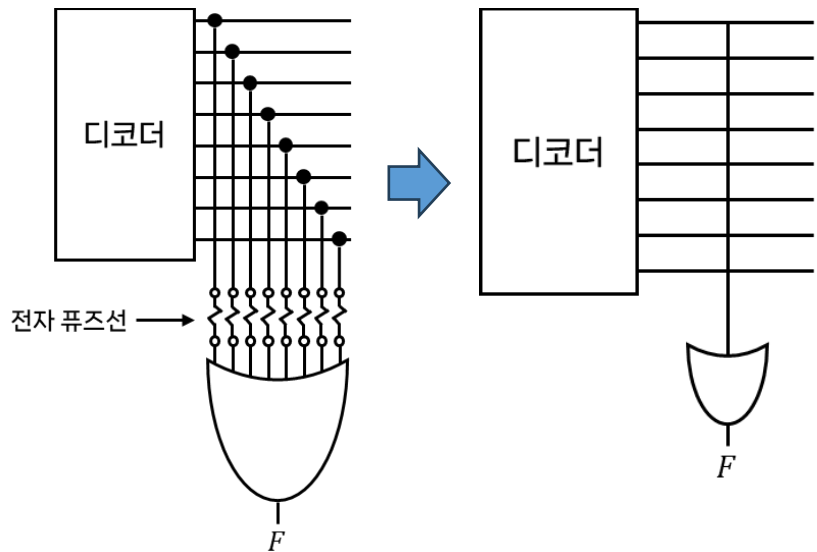
- ▶ 3×8 디코더는 3개의 입력과 8개의 출력
- ▶ 디코더의 출력은 메모리 배열의 주소를 가리킴
 - ▶ 디코더의 출력은 퓨즈선을 통해 4개의 OR 게이트와 연결
 - ▶ 이때 OR 게이트는 내부적으로 8개의 입력을 갖고 있음
 - ▶ 즉, 8개의 내부 입력선을 갖는 4개의 OR게이트를 통해 총 $8 \times 4 = 32$ 개의 퓨즈선으로 연결됨



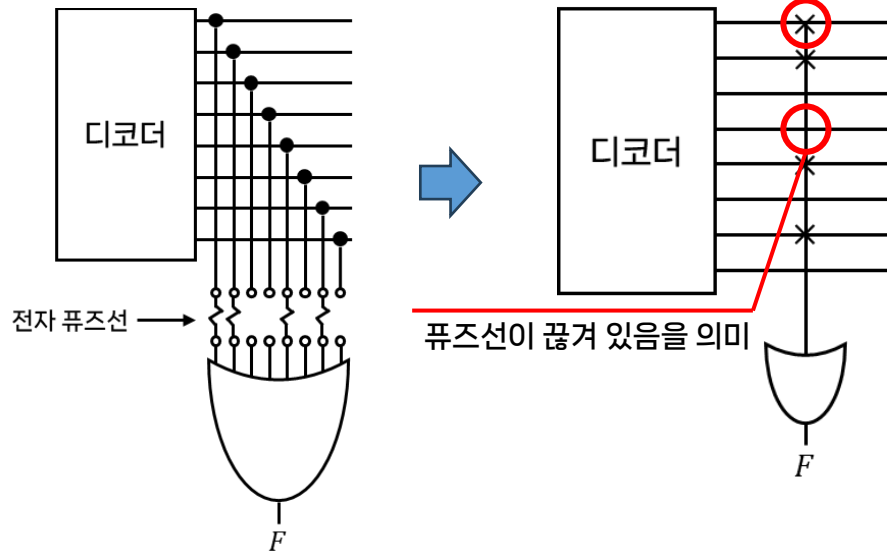
8.3.2 RDM의 구성

• ROM의 내부 구조(3)

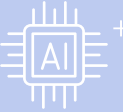
• OR 게이트의 연결 논리와 전자 퓨즈선의 표현



전자 퓨즈선으로 연결된 OR게이트의 세부적인 표현



퓨즈선의 단절 유무를 표현하는 방식



8.3.3 ROM을 이용한 조합논리회로 구현

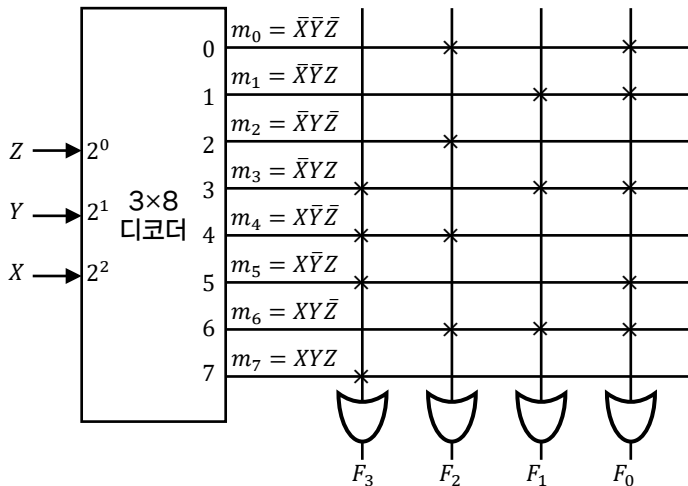


• ROM을 이용한 부울함수 구현

• ROM은 디코더와 OR 게이트로 이루어진 논리회로이므로 ROM을 이용해 부울함수의 구현이 가능

• 8×4 ROM을 이용해 부울함수 구현

X	Y	Z	F ₃	F ₂	F ₁	F ₀
0	0	0	0	1	0	1
0	0	1	0	0	1	1
0	1	0	0	1	0	0
0	1	1	1	0	1	1
1	0	0	1	1	0	0
1	0	1	1	0	0	1
1	1	0	0	1	1	1
1	1	1	1	0	0	0

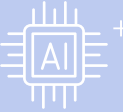


$$F_0(X, Y, Z) = \Sigma m(0, 1, 3, 5, 6)$$

$$F_1(X, Y, Z) = \Sigma m(1, 3, 6)$$

$$F_2(X, Y, Z) = \Sigma m(0, 2, 4, 6)$$

$$F_3(X, Y, Z) = \Sigma m(3, 4, 5, 7)$$



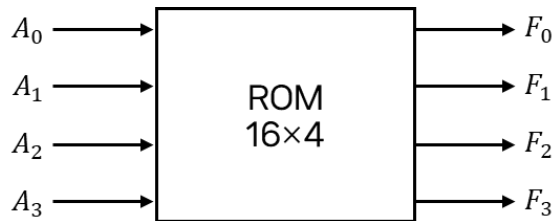
8.3.3 RDM을 이용한 조합논리회로 구현



• 진리표로 부터 직접 조합논리회로 구현 가능

• 16×4 ROM을 이용한 그레이 코드 변환기 구현

A_3	A_2	A_1	A_0	F_3	F_2	F_1	F_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0



- ▶ 진리표를 만족하기 위해서는 4개의 입/출력이 필요
- ▶ ROM의 4개 입력은 16개의 word를 식별할 수 있음
 - ▶ 즉, ROM의 크기는 16×4가 됨



내용 정리

Summary

Contents



14강 | 기억장치와 PLD(1)



디지털 +
논리회로



01 기억장치 개요

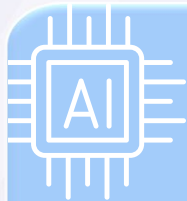
02 RAM

- RAM의 구성
- RAM의 확장

03 ROM

- ROM의 구성
- ROM을 이용한 조합논리회로





다음시간에는

15 강. 기억장치와 PLD(2)