

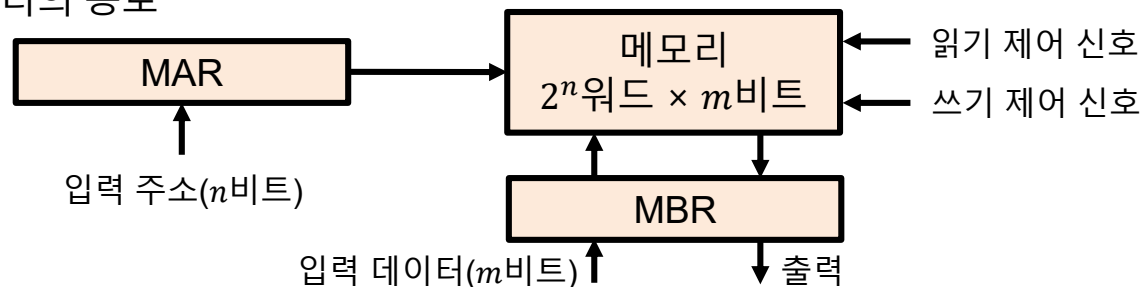
Lecture 11

메모리와 프로그래머블 논리 장치

메모리 구조

정의

- 메모리는 2진 데이터를 **워드(word)**라는 비트의 집합으로 저장함
- 워드는 메모리에서 입출력 시 한 번에 전송되는 비트들을 의미함
- 메모리와 주변 장치 사이의 데이터 전송은 제어 신호 2개와 레지스터 2개를 통해 이루어짐
 - 읽기 제어 신호(read control signal) : 메모리로부터 데이터를 읽으라는 신호
 - 쓰기 제어 신호(write control signal) : 데이터를 메모리에 저장하라는 신호
 - 메모리 주소 레지스터(MAR, memory address register) : 메모리에 있는 특정한 워드와 데이터 전송을 수행하는 경우, 해당 워드의 주소가 MAR에 전송됨
 - 메모리 버퍼 레지스터(MBR, memory buffer register) : 레지스터와 외부 장치 사이에서 전송되는 데이터의 통로



메모리 구조

■ 데이터 쓰기

- 메모리에 저장하려는 데이터, 주소, 쓰기 제어 신호 3가지 정보를 입력으로 제공해야 함
- 쓰기 제어 신호가 메모리에 입력되면 데이터는 MBR에, 주소는 MAR에 저장됨
- 메모리의 내부 회로가 MAR에 주소를 받아 이 주소가 지정하는 위치에 MBR의 내용을 저장함

■ 데이터 읽기

- 메모리에 읽어내려는 주소, 읽기 제어 신호 2가지 정보를 입력으로 제공해야 함
- 읽기 제어 신호가 메모리에 입력되면 주소는 MAR에 저장됨
- 메모리의 내부 회로가 MAR에 주소를 받아 이 주소가 지정하는 위치에 워드를 MBR에 일시적으로 저장하고 MBR로 외부에 내보냄

메모리 구조

- 예, K4M51323LC – 삼성의 mobile SDRAM

K4M51323LC - S(D)N/G/L/F

Mobile-SDRAM

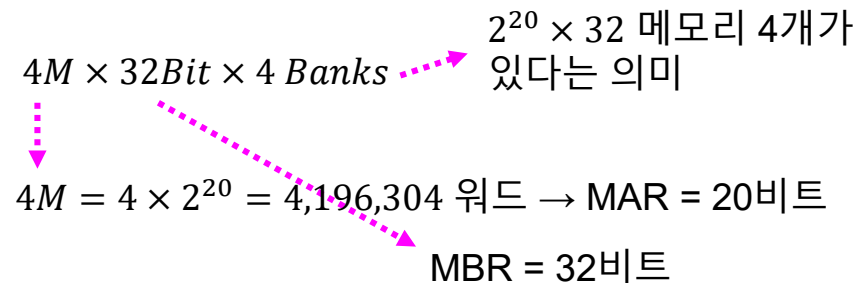
4M x 32Bit x 4 Banks Mobile SDRAM in 90FBGA

FEATURES

- VDD/VDDQ = 2.5V/2.5V
- LVCMOS compatible with multiplexed address.
- Four banks operation.
- MRS cycle with address key programs.
 - CAS latency (1, 2 & 3).
 - Burst length (1, 2, 4, 8 & Full page).
 - Burst type (Sequential & Interleave).
- EMRS cycle with address key programs.
- All inputs are sampled at the positive going edge of the system clock.
- Burst read single-bit write operation.
- Special Function Support.
 - PASR (Partial Array Self Refresh).
 - Internal TCSR (Temperature Compensated Self Refresh)
- DQM for masking.
- Auto refresh.
- 64ms refresh period (8K cycle).
- Commercial Temperature Operation (-25°C ~ 70°C).
- Extended Temperature Operation (-25°C ~ 85°C).
- 90Balls FBGA (-SXXX -Pb, -DXXX -Pb Free).

GENERAL DESCRIPTION

The K4M51323LC is 536,870,912 bits synchronous high data rate Dynamic RAM organized as 4 x 4,196,304 words by 32 bits, fabricated with SAMSUNG's high performance CMOS technology. Synchronous design allows precise cycle control with the use of system clock and I/O transactions are possible on every clock cycle. Range of operating frequencies, programmable burst lengths and programmable latencies allow the same device to be useful for a variety of high bandwidth and high performance memory system applications.

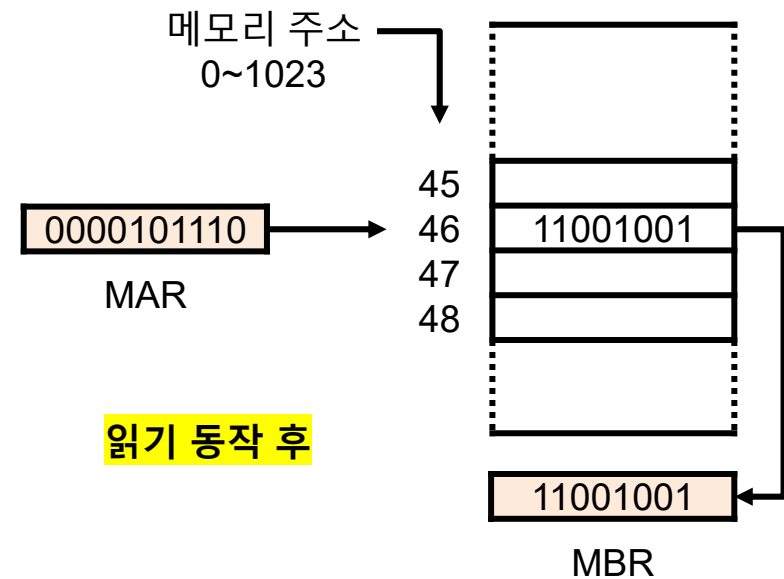
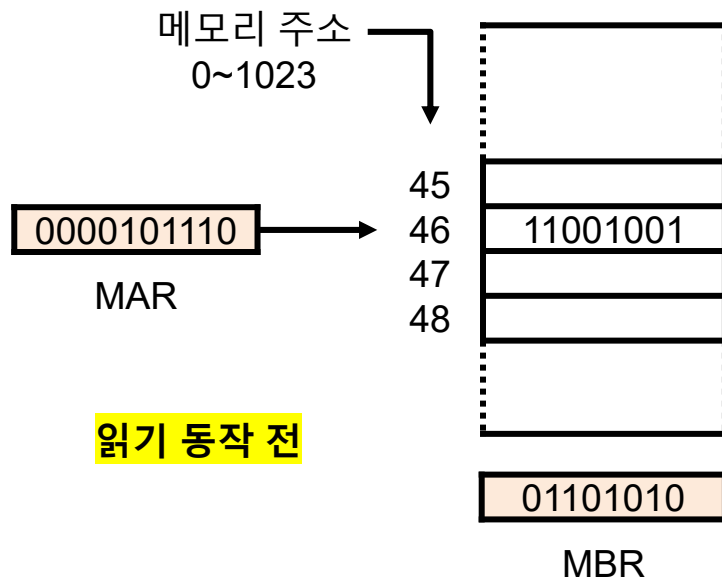


Source: <https://www.alldatasheet.com/datasheet-pdf/view/145190/SAMSUNG/K4M51323LC.html> (accessed on 2024.09.21)

메모리 동작

■ 메모리 읽기 동작

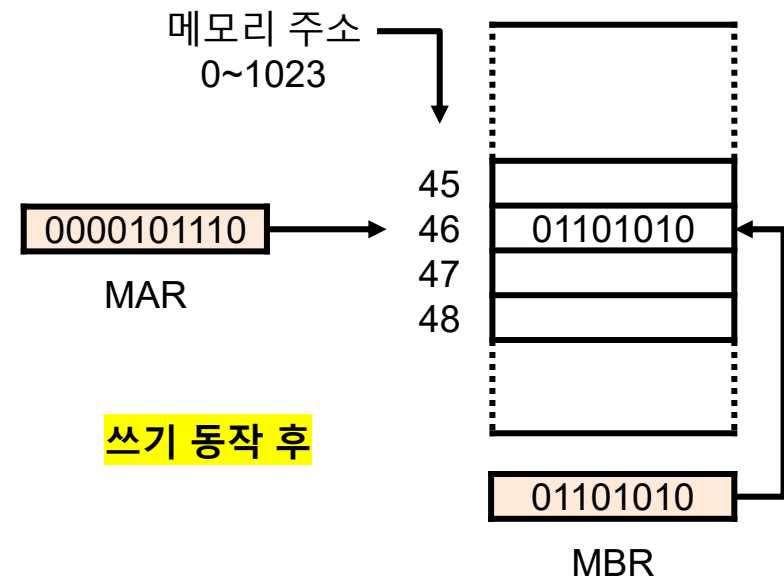
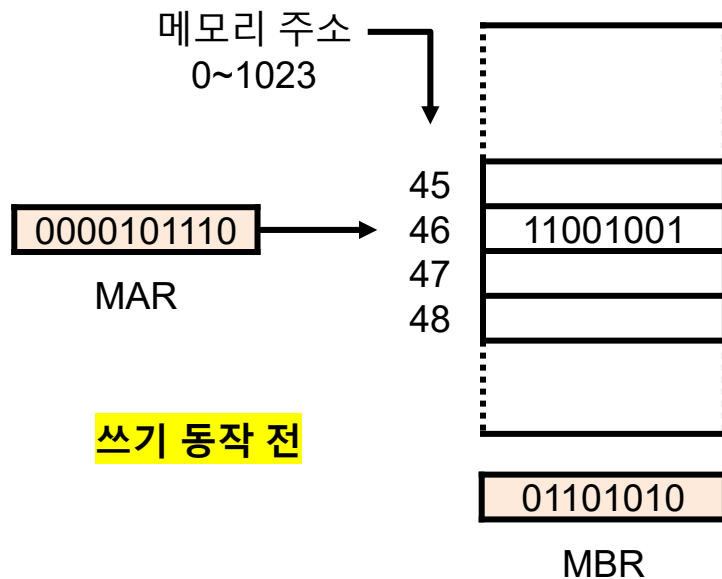
- MAR이 지정하는 주소의 메모리 내용을 읽은 동작 순서
 - 선택된 워드의 주소를 MAR로 전송함
 - 읽기 제어 신호를 동작시킴



메모리 동작

■ 메모리 쓰기 동작

- MAR이 지정하는 주소의 메모리에 새 데이터를 저장하는 동작 순서
 - 지정된 메모리의 주소를 MAR로 전송함
 - 저장하려는 데이터 비트를 MBR로 전송함
 - 쓰기 제어 신호를 동작시킴



메모리 분류

■ 접근 방법에 의한 분류

- **액세스(access)** : 메모리의 주소에 해당하는 위치에 데이터를 읽거나 씀
- 액세스하는 방법에 따라 2가지로 구분할 수 있음
 - 랜덤 액세스 메모리(RAM, random access memory)
액세스 시간은 어느 위치나 동일하게 걸리는 메모리 형태
 - 순차 액세스 메모리(SAM, sequential access memory)
원하는 위치에 도달하는 데 일정한 시간이 경과되는 형태이므로 액세스 시간은 위치에 따라 달라짐

랜덤 액세스 메모리



순차 액세스 메모리



메모리 분류

■ 기록 기능에 의한 분류

- 대부분의 메모리는 정보를 기록하는 기능과 기록된 정보를 읽는 판독 기능이 있음
- 사용자 기록이 가능할 지에 따라 2가지로 구분할 수 있음
 - RWM(read and write memory)
사용자가 기록과 판독을 모두 수행할 수 있는 메모리
 - ROM(read only memory)
판독만 가능한 메모리
 - ROM에 기록된 정보는 전원이 꺼져도 지워지지 않은 특징이 있음
 - ROM 종류
 - MROM(mask ROM) : 제조 시 정보가 기록되어 있음
 - PROM(programmable ROM) : 제조된 후 사용자가 기록할 수 있음
 - Fuse-link PROM : 한 번만 기록이 가능함
 - EPROM(erasable PROM) : 자외선을 쬐여서 내용을 지운 후에 다시 기록이 가능함
 - EEPROM(electrically EPROM) : 전기적으로 내용을 지운 후에 다시 기록이 가능함

메모리 분류

- 기억 방식에 의한 분류

- RAM은 기억 방식에 따라 2가지로 구분할 수 있음

- SRAM(static RAM)

주로 2진 정보를 저장하는 내부 플립플롭으로 구성됨, 저장된 정보는 전원이 공급되는 동안 보존됨

- DRAM(dynamic RAM)

2진 정보를 커패시터(capacitor)에 공급되는 전하 형태로 보관됨, 커패시터에 사용되는 전하는 시간이 경과하면 방전되므로 DRAM에서는 일정한 시간 안에 **재충전(refresh)**해야 함

- DRAM은 전력 소비가 적고 단일 메모리 칩에 더 많은 정보를 저장할 수 있으며, SRAM은 사용하기 쉽고 읽기와 쓰기 사이클(cycle)이 더 짧다는 특징이 있음

메모리 분류

■ 휘발성/비휘발성 메모리

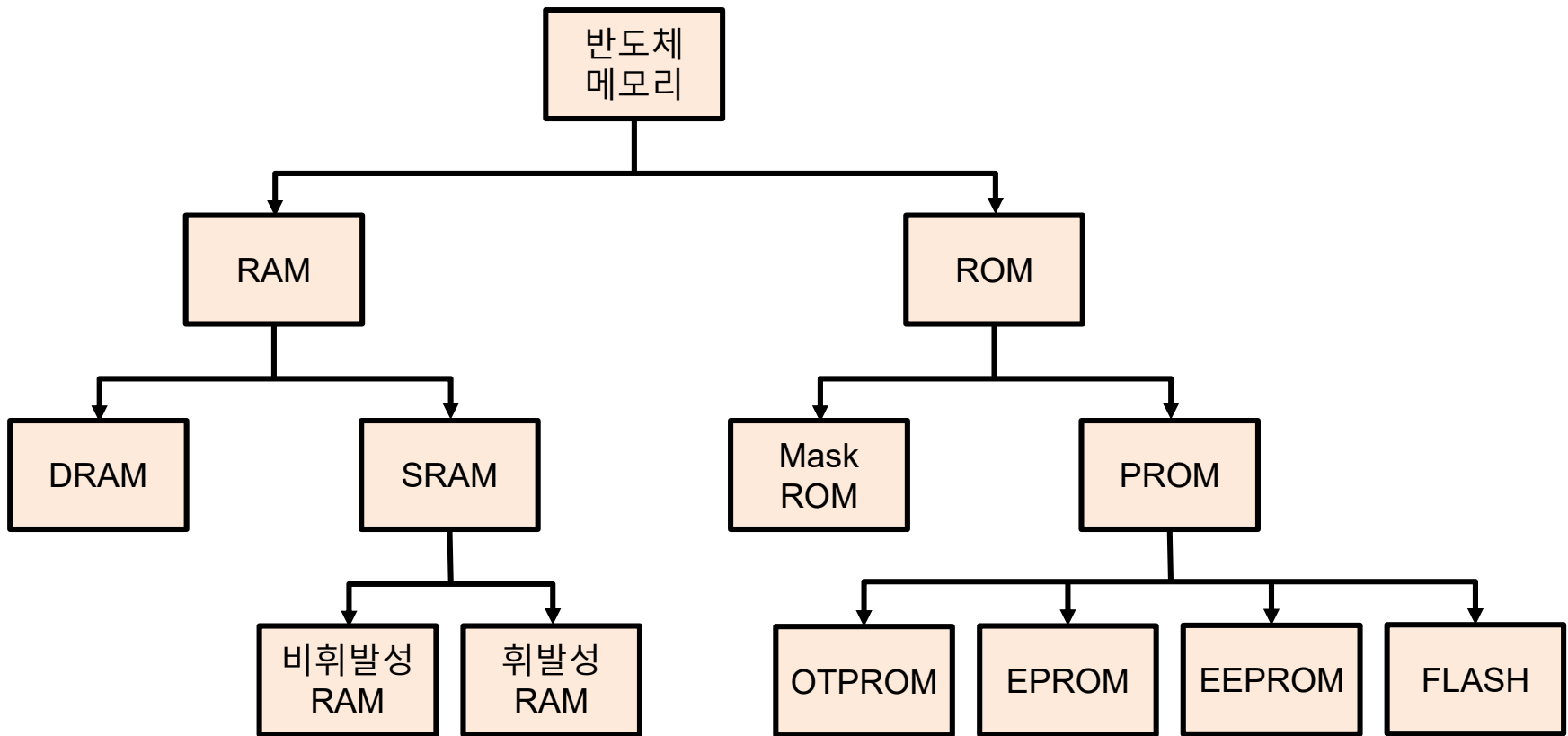
- 휘발성(volatile) : 일정한 시간이 지나거나 전원이 꺼지면 저장된 내용이 지워짐
 - RAM은 외부에서 공급되는 전원을 통해 정보를 저장하기 때문에 휘발성 메모리에 해당함
- 비휘발성(non-volatile) : 전원이 차단되어도 기록된 정보가 계속 유지됨
 - 자기 소자에 저장된 정보는 자화 방향으로 나타내는데, 이 자화 방향은 전원이 차단된 후에도 계속 유지함
 - ROM은 비휘발성 메모리에 해당함
 - 비휘발성 특성 때문에 디지털 컴퓨터 동작에 필요한 프로그램을 저장하는 데 사용할 수 있음

메모리 분류

- 기억 소자에 의한 분류
 - 기억 소자에 따라 크게 2가지로 구분할 수 있음
 - 바이폴라(bipolar) 메모리
 - 메모리 셀 및 주변 회로에 BJT(bipolar junction transistor)를 사용함
 - 액세스 시간이 빠르지만 전력 소비가 많음
 - 집적도가 높은 경우에는 사용하지 않음
 - MOS(metal oxide semiconductor) 메모리
 - PMOS, NMOS 또는 CMOS를 사용함
 - 바이폴라 메모리에 비해서 속도가 느리지만 전력 소비가 적음
 - VLSI와 같은 집적도가 높은 경우에 적합함
 - 위 2가지의 메모리 종류 외에는 CCD(charge coupled device)와 MBM(magnetic bubble memory) 등이 있지만 현재의 메모리 제작에는 거의 사용되지 않음

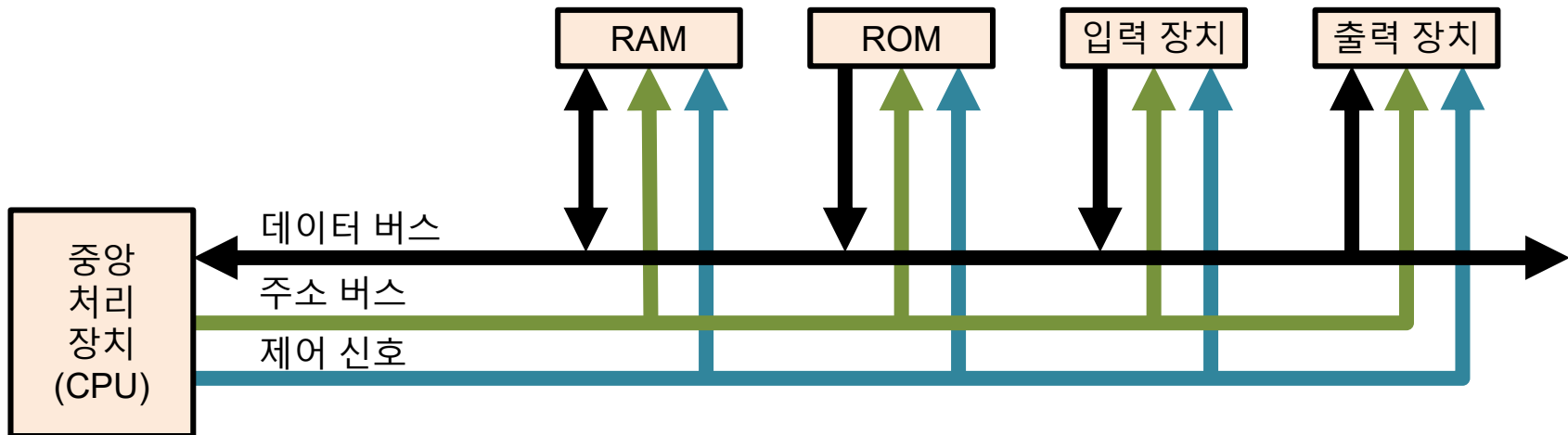
메모리 분류

■ 요약



컴퓨터에서의 메모리

- 주소 버스
 - 1개의 메모리 위치를 선택하기 위해 CPU에서 메모리 IC까지 2진 주소를 전달하는 단방향 버스
- 데이터 버스
 - CPU와 메모리 IC 사이에서 데이터 옮기는 양방향 버스
- 제어 버스
 - CPU에서 메모리 IC까지 제어 신호를 전달하는 단방향 버스



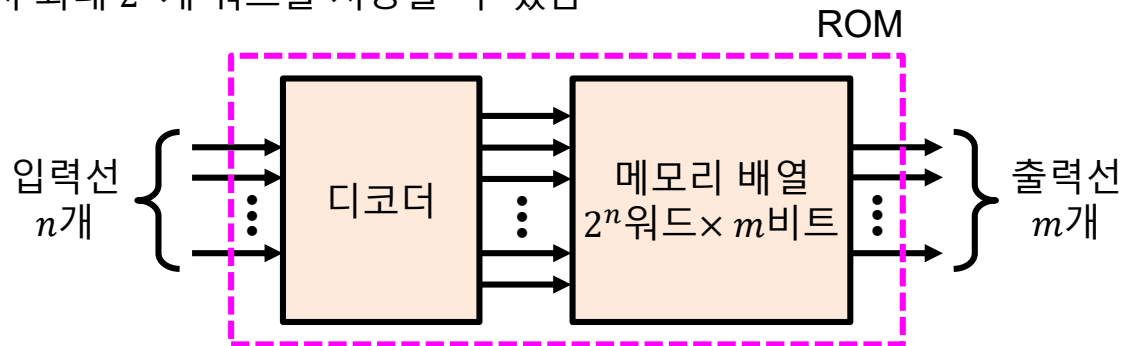
ROM

ROM

- 저장된 데이터를 읽을 수는 있으나 특별한 장치가 없이는 데이터를 기록하거나 변경할 수 없음
- 코드 변환, 수학적 환산표, 컴퓨터 명령어 등 변하지 않고 자주 사용하는 데이터를 저장하는 데 많이 사용됨

ROM 구조

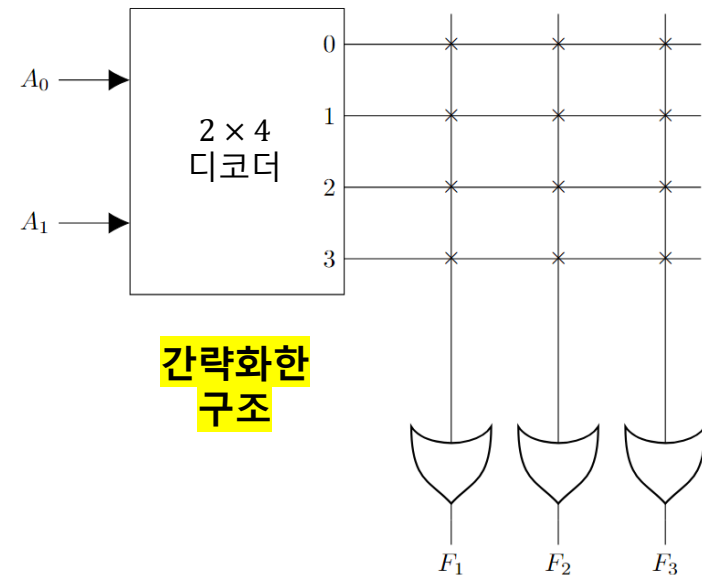
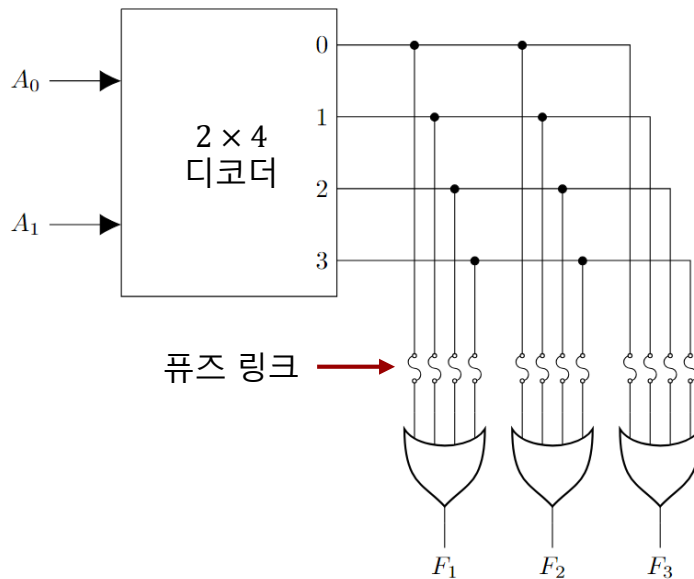
- 입력선 n 개 \leftrightarrow 입력 n 비트 \leftrightarrow 주소
- 출력선 m 개 \leftrightarrow 출력 m 비트 \leftrightarrow 워드
- ROM에서 최대 2^n 개 워드를 저장할 수 있음



ROM

4 × 3 ROM의 내부 논리 구조

- 입력 변수 2개는 디코더를 통해 4개가 출력됨
- 각 디코더의 출력은 4개 주소 중에서 1개만 선택함
- 디코더의 4개 출력은 각 OR 게이트의 퓨즈(fuse)를 통해 연결됨
- 내부 퓨즈가 $4 \times 3 = 12$ 개 있으며, 프로그램으로 퓨즈의 연결을 절단할 수 있음

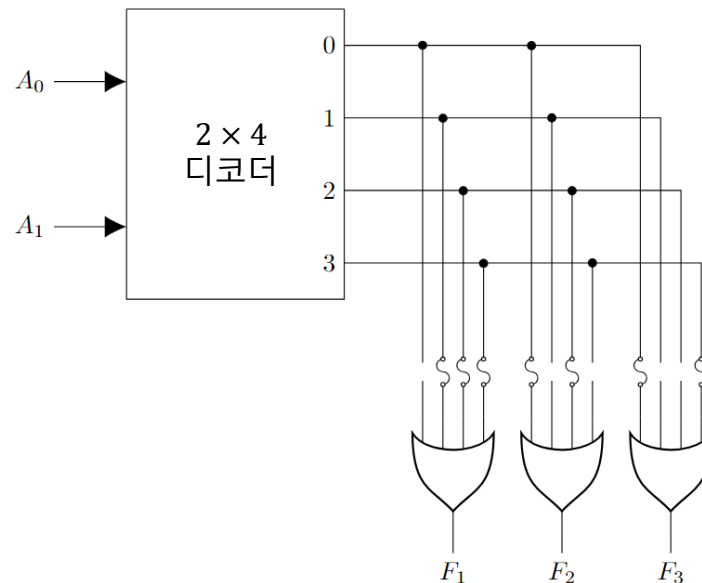


ROM



ROM을 사용한 조합논리회로의 구현

- ROM의 각 출력은 입력 변수 n 개의 최소항을 모두 합한 것이며, 임의의 불 함수는 곱의 합(SOP)으로 표현할 수 있음
- 입력 n 개와 출력 m 개를 갖는 조합논리회로를 구성하는 경우 $2^n \times m$ ROM이 필요하며, 퓨즈를 절단하는 것을 **ROM 프로그래밍**한다고 함
- 예, $F_1(A_0, A_1) = \sum m(1,2,3)$, $F_2(A_0, A_1) = \sum m(0,3)$, $F_3(A_0, A_1) = \sum m(0,4)$



RAM

■ RAM

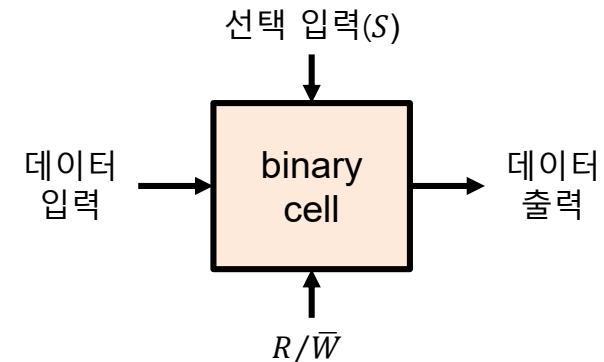
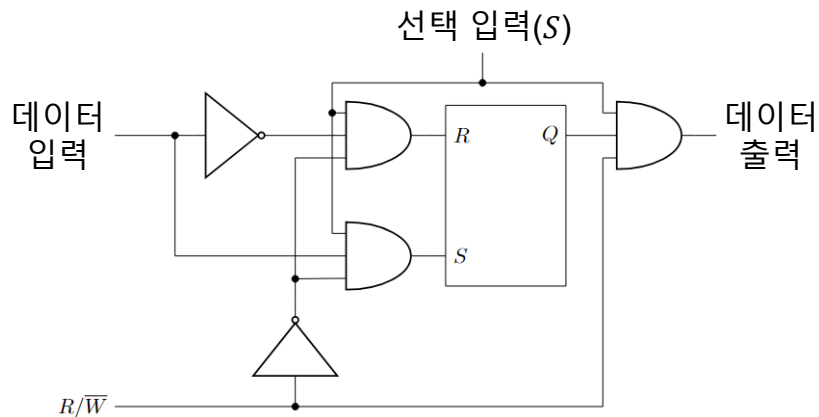
- 액세스 시간은 어느 위치나 동일하게 걸리는 메모리
- RWM(read and write memory) : 메모리에 저장된 데이터를 읽을 수 있으며, 또 필요한 데이터를 저장할 수 있음
- SRAM과 DRAM이 있음
 - SRAM
 - 바이폴라(bipolar) 또는 MOS 트랜지스터를 사용해 제조됨
 - 저장 소자는 **래치**(latch)이므로 전원이 켜 있는 동안에는 언제나 데이터를 저장하거나 읽을 수 있음
 - DRAM
 - MOS 트랜지스터를 사용해 제조됨
 - 저장 소자는 **커패시터**(capacitor)이므로 일정 시간마다 주기적으로 재충전(refreshing)해야 함

RAM



■ SRAM

- 휘발성 메모리
- 데이터 비트를 **래치**에 저장함
- 각 워드가 n 비트이고, 워드 m 개로 구성된 RAM의 내부 구조는 **메모리 셀** $m \times n$ 개와 각 워드를 선택하는 데 필요한 주소를 출력하는 논리회로로 구성됨
- 1비트 데이터를 저장하는 SRAM의 메모리 셀의 논리회로가 다음과 같음

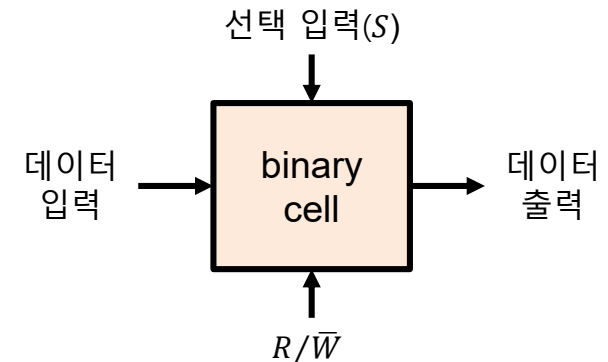
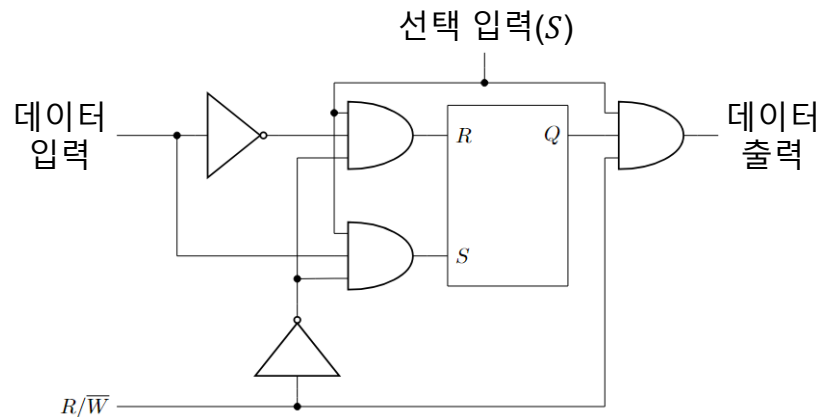


RAM



■ SRAM의 메모리 셀

- 선택 입력(S)은 여러 메모리 셀 중에서 주소를 통해 선택되었을 때 논리 1이 입력됨
- $S = 1$ 일 때
 - R/\bar{W} 입력이 1이면 래치에 저장된 데이터 비트가 데이터 출력 단자를 통해 출력됨
 - R/\bar{W} 입력이 0이면 데이터 입력 단자에 있던 데이터 비트가 래치로 전송되어 저장됨

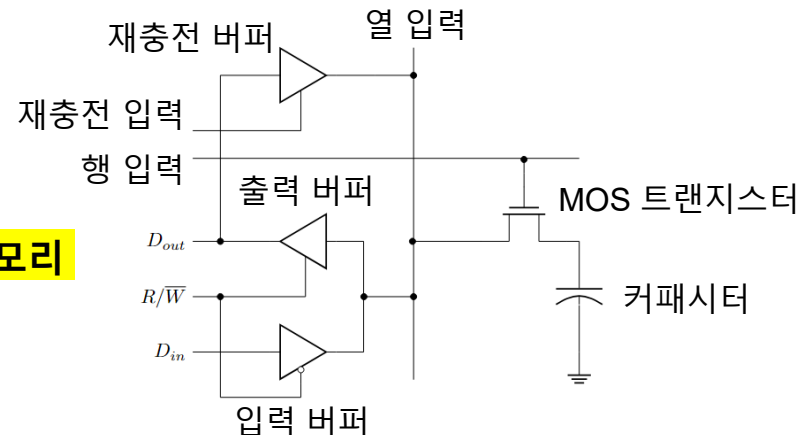


RAM

■ DRAM

- 휘발성 메모리
- 데이터 비트를 **커패시터**에 저장함
- DRAM의 메모리 셀은 구조가 매우 단순하기 때문에 SRAM보다 비트당 가격이 저렴하고, 고밀도 칩을 구성할 수 있음
- 메모리 셀을 구성하는 커패시터는 일정한 시간이 지나면 저장된 데이터가 소멸되기 때문에 주기적으로 **재충전**해 줘야 함

DRAM의 메모리 셀 구조



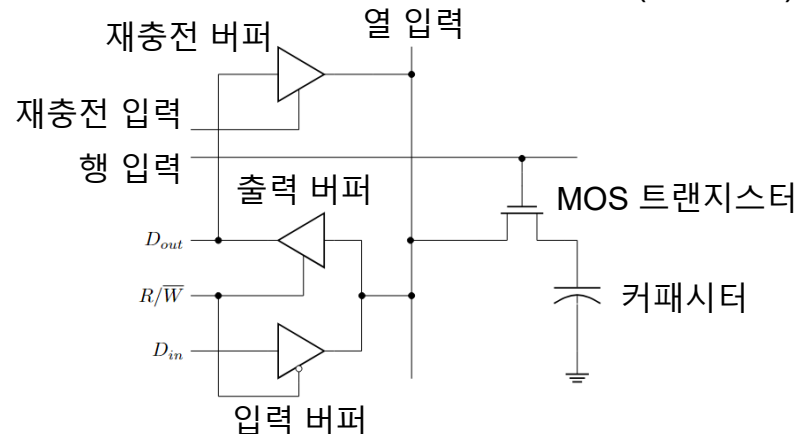
RAM



■ DRAM의 메모리 셀 동작

■ 쓰기 모드

- $R/\bar{W} = 0$ 일 때 쓰기 모드에서 3상태 버퍼인 입력 버퍼가 인에이블(enable)되고, 출력 버퍼는 디스에이블(disable)됨
- 메모리 셀에 논리 1을 저장하려면 데이터 입력 D_{in} 을 논리 1로 하고, 행 입력이 논리 1이면 MOS 트랜지스터는 on 상태가 되어서 커패시터에는 양(+)의 전압이 충전됨. 논리 0을 저장하기 위해서는 데이터 입력(D_{in})을 논리 0으로 하면 커패시터는 충전되지 않음
- 커패시터에 논리 1이 저장되어 있는 경우는 커패시터는 방전하게 됨. 그러나 열 입력이 논리 0이므로 트랜지스터는 off가 되어, 커패시터의 전하(1 또는 0)는 트래핑(trapping)됨



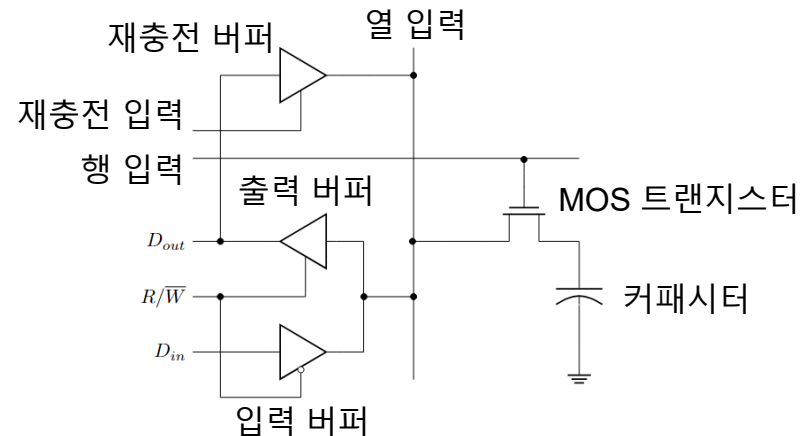
RAM



■ DRAM의 메모리 셀 동작

■ 읽기 모드

- $R/\bar{W} = 1$ 일 때 읽기 모드에서 3상태 버퍼인 출력 버퍼가 인에이블(enable)되고, 입력 버퍼는 디스에이블(disable)됨
- 행 입력이 논리 1이면 MOS 트랜지스터는 on 상태가 되어 커패시터는 비트 선(bit line)을 통해 출력 버퍼에 연결됨
- 저장된 데이터는 출력(D_{out})을 통해 외부로 출력됨



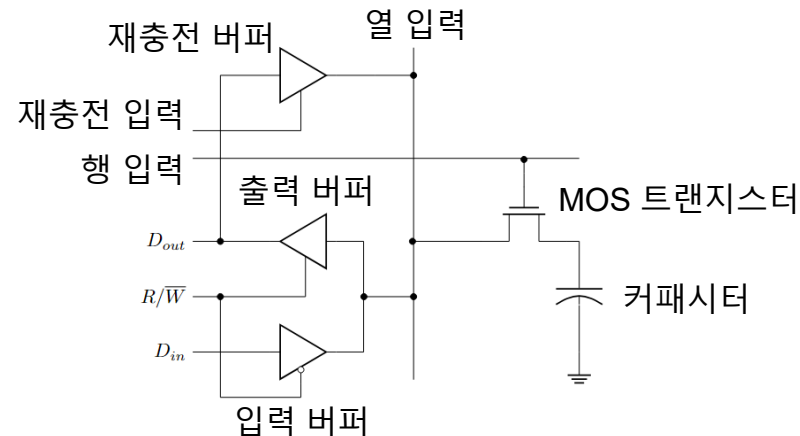
RAM



■ DRAM의 메모리 셀 동작

■ 재충전

- R/\bar{W} 와 행 입력과 재충전 입력을 논리 1로 하면 DRAM 메모리 셀을 재충전하게 됨
 - 행 입력이 1이므로 MOS 트랜지스터가 on이 되어 커패시터는 비트 선에 연결됨
 - $R/\bar{W} = 1$ 이므로 출력 버퍼는 인에이블되고, 저장된 데이터 비트는 재충전 입력이 1이 되어 인에이블되므로 재충전 버퍼에 다시 입력됨
- 저장된 비트의 논리값을 커패시터에 다시 충전하게 됨



RAM



■ SRAM과 DRAM 비교

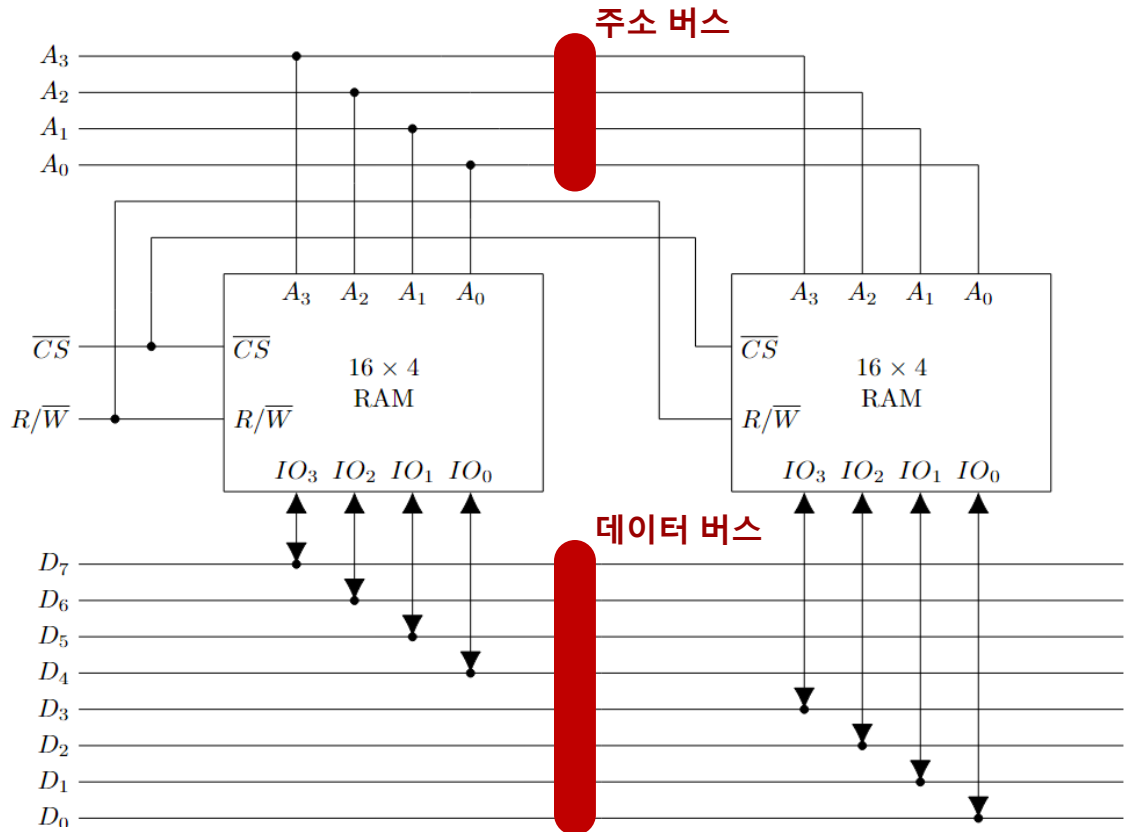
구분	SRAM	DRAM
구송 소자	플립플롭	커패시터
집적도	낮음	높음
전력 소모	많음	적음
동작 속도	빠름	느림
가격	고가	저가
재충전 여부	필요하지 않음	필요함
용도	캐시 메모리	주기억 장치

RAM



■ 메모리 확장

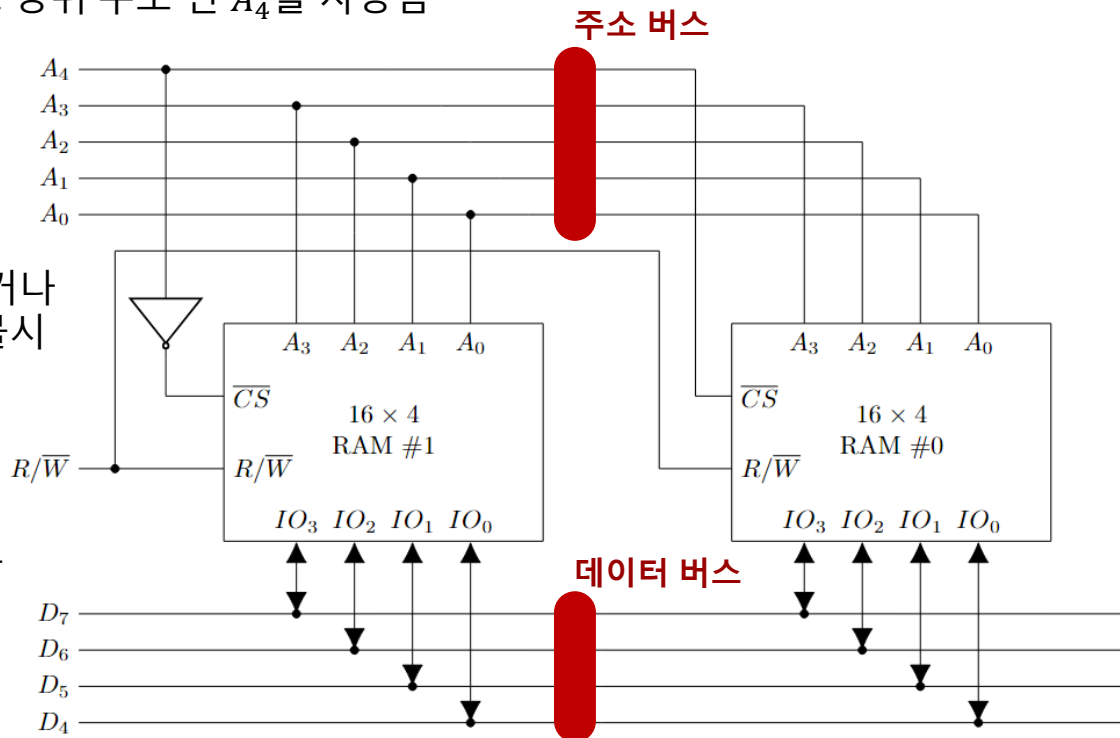
- 16×4 RAM 2개를 이용해 워드 길이가 8비트인 16×8 RAM을 구성함
- \overline{CS} (chip select)는 RAM을 선택하는 입력이며, R/\overline{W} 는 선택된 RAM 칩의 읽기와 쓰기 동작을 제어함
- $\overline{CS} = 1$ 이면 RAM 칩은 선택되지 않고 출력은 하이 임피던스 상태가 됨
- $\overline{CS} = 0$ 이고 $R/\overline{W} = 1$ 이면 주소에 의해 선택된 8비트의 데이터가 출력 선을 통해 출력됨
- 연결된 메모리의 주소 수는 각각의 메모리 주소 수와 같음



RAM

워드 용량 확장

- 16 × 4 RAM 2개를 사용해 32 × 4 RAM을 구성함
- 기존의 4비트 주소 버스에 추가로 상위 주소 선 A_4 를 사용함
- A_4 는 2개의 RAM 중 하나를 선택할 수 있도록 동작함. 다른 주소 선 $A_3 \sim A_0$ 는 선택된 RAM 모듈에 대해서 16개의 메모리 중 하나를 선택할 때 사용됨
- $A_4 = 0$ 일 때, RAM #0의 \overline{CS} 는 읽거나 쓰기가 가능하도록 칩을 인에이블시킴. RAM #0의 주소 공간 범위는 $A_4A_3A_2A_1A_0 = 00000_2 \sim 01111_2$
- $A_4 = 1$ 일 때, RAM #1의 \overline{CS} 는 읽거나 쓰기가 가능하도록 칩을 인에이블됨. RAM #1의 주소 공간 범위는 $A_4A_3A_2A_1A_0 = 10000_2 \sim 11111_2$



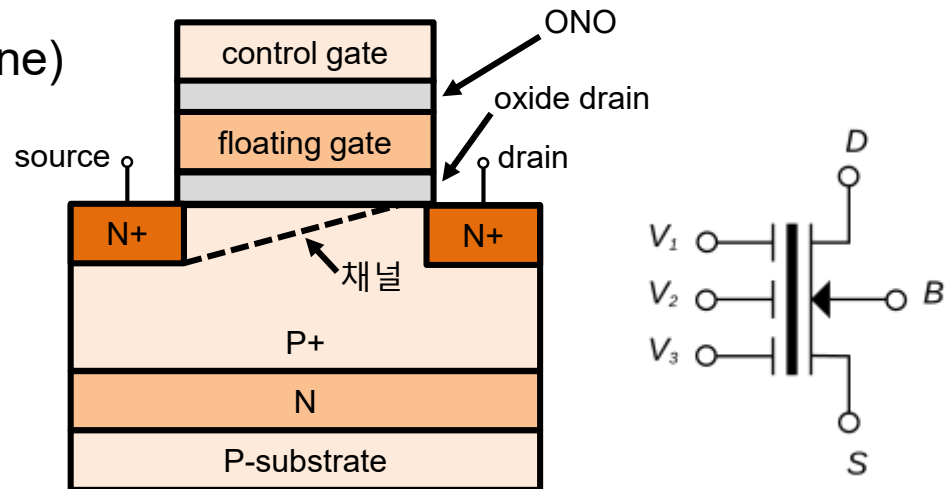
플래시 메모리

- 플래시 메모리(**flash memory**)
 - ROM의 비휘발성 장점과 정보의 입출력이 자유로운 RAM의 장점을 동시에 지닌 반도체 메모리
 - 2001년부터 USB 드라이브, thumb 드라이브라는 이름으로 소개되면서 주목을 받기 시작했으며, 휴대용 디지털 기기에 사용되면서 그 사용량이 급격히 증가하기 시작했음
 - 메모리의 내부 전자 회로 형태에 따라 **NAND 플래시**와 **NOR 플래시**로 나눔
 - NAND 플래시: 저장 단위인 메모리 셀을 수직으로 배열하고 대용량화에 유리하고 쓰기 및 지우기 속도가 빠름
 - NOR 플래시 : 저장 단위인 메모리 셀을 수평으로 배열하고 읽기 속도가 빠름

플래시 메모리

■ 플래시 메모리의 셀 구조

- FGMOS(floating gate MOSFET)라는 특별한 구조의 MOSFET에 전하 (electrical charge)를 축적하여 데이터를 기억함
- FGMOS의 구조는 제어 게이트(control gate), 플로팅 게이트(floating gate), 드레인(drain), 소스(source)로 구성됨
- 플로팅 게이트는 모든 주위가 전기적으로 절연되어 공중에 떠 있는 게이트이며, 제어 게이트는 메모리 셀 어레이의 워드라인(WL, word line)을 겸하는 게이트임
- 플로팅 게이트에 축적된 전하의 유무에 따라 0과 1의 데이터를 저장하며, 축적된 전하는 전원 공급이 없어도 2~10년 동안 전하를 저장할 수 있음

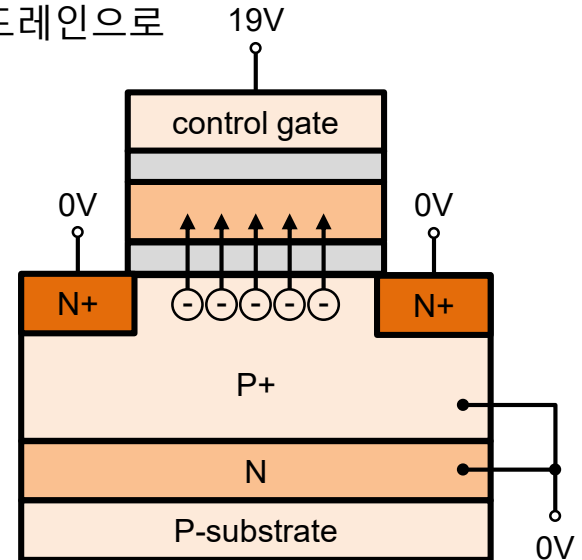


플래시 메모리

■ 플래시 메모리의 기본 동작

■ 쓰기 동작

- 플래시 메모리를 구입했을 때나 지우기 동작 후에는 각 메모리 셀의 전자는 모두 제거되었으므로 모든 메모리 셀은 초기에 논리 1 상태에 있음
- 쓰기 동작은 논리 0인 셀의 플로팅 게이트에는 전자를 채우고 논리 1인 셀에는 전자를 채우지 않은 것으로 이해할 수 있음
- 제어 게이트에 약 12~19V의 전압을 인가하면 소스에서 드레인으로 흘러가던 전자가 얇은 산화물 층을 뚫고 플로팅 게이트로 끌려 들어가게 됨
 - NAND 플래시인 경우에는 터널 주입 (tunnel injection)이라고 함
 - NOR 플래시인 경우에는 hot-electron injection이라고 함
- 산화물로 둘러싸인 플로팅 게이트는 외부와 차단되어서 플로팅 게이트에 들어온 전자는 큰 전기장의 영향을 받지 않는 이상 외부로 빠져나갈 수 없음 → 전원이 끊긴 상태에서도 데이터가 지워지지 않음

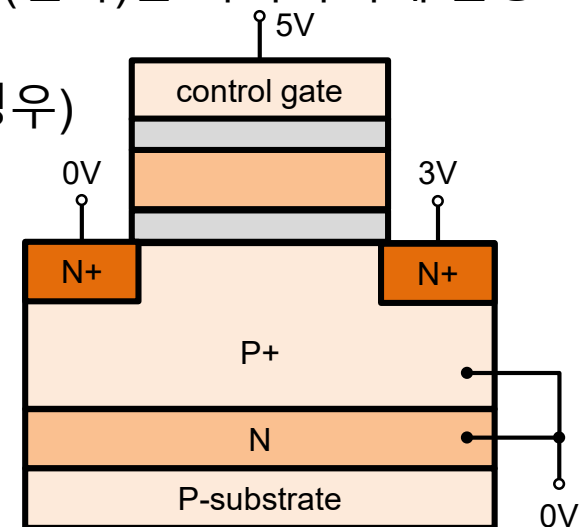


플래시 메모리

■ 플래시 메모리의 기본 동작

■ 읽기 동작

- 제어 게이트에 +5V 전압을 인가하며, 플로팅 게이트에 있는 전자의 유무에 따라 제어 게이트 전압에 의해서 FGMOS의 on/off가 결정됨
- 플로팅 게이트에 전자가 채워져 있으면(논리 0인 경우) 제어 게이트 전압이 플로팅 게이트에 저장되어 있는 음전하(전자)를 극복하기에 불충분하므로 FGMOS는 off됨
- 플로팅 게이트에 전자가 없으면 (논리 1인 경우) 제어 게이트 전압은 FGMOS를 on시키기에 충분함
- FGMOS가 on되면 드레인에서 소스로 전류가 흐르는데 이 전류가 감지되면 논리 1이고, off되어 전류가 감지되지 않으면 논리 0이 됨

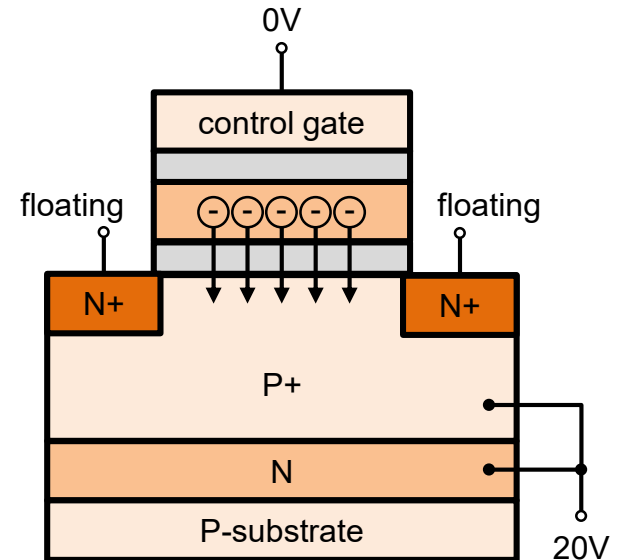


플래시 메모리

■ 플래시 메모리의 기본 동작

■ 지우기 동작

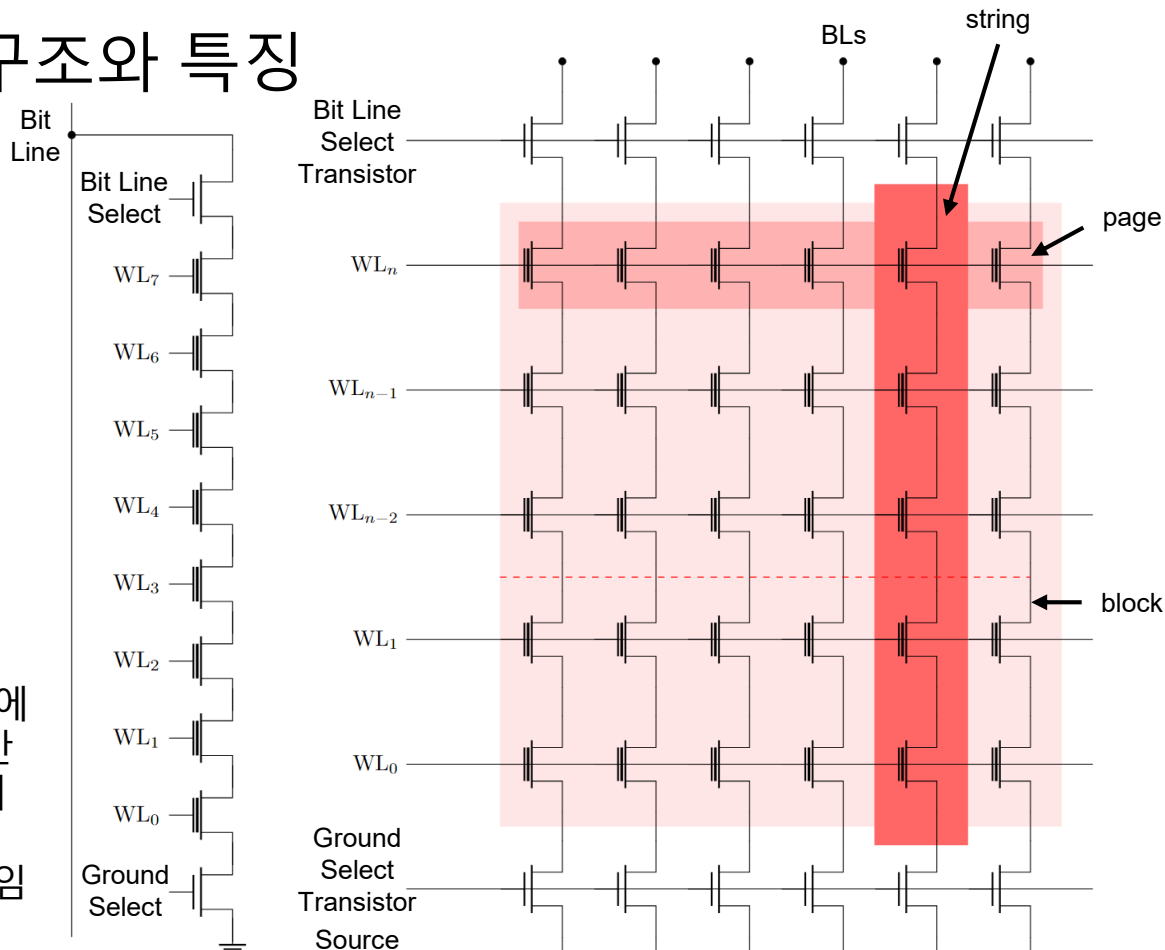
- P+와 N층에 약 20V 전압을 인가하면 쓰기 동작과는 반대로 플로팅 게이트에 있던 전자들이 강한 전기장의 힘에 이끌려 산화물 층을 통과하여 밖으로 나오게 되고 플로팅 게이트에는 전자가 사라져서 논리 1의 상태로 바뀌게 됨
 - NAND 플래시인 경우에는 **터널 릴리즈** (tunnel release)라고 함
 - NOR 플래시인 경우에는 **F-N 터널링** (Fowler-Nordheim tunneling)이라고 함
- 플래시 메모리는 쓰기 동작 전에 항상 지우기 동작을 수행함



플래시 메모리

■ NAND 플래시의 구조와 특징

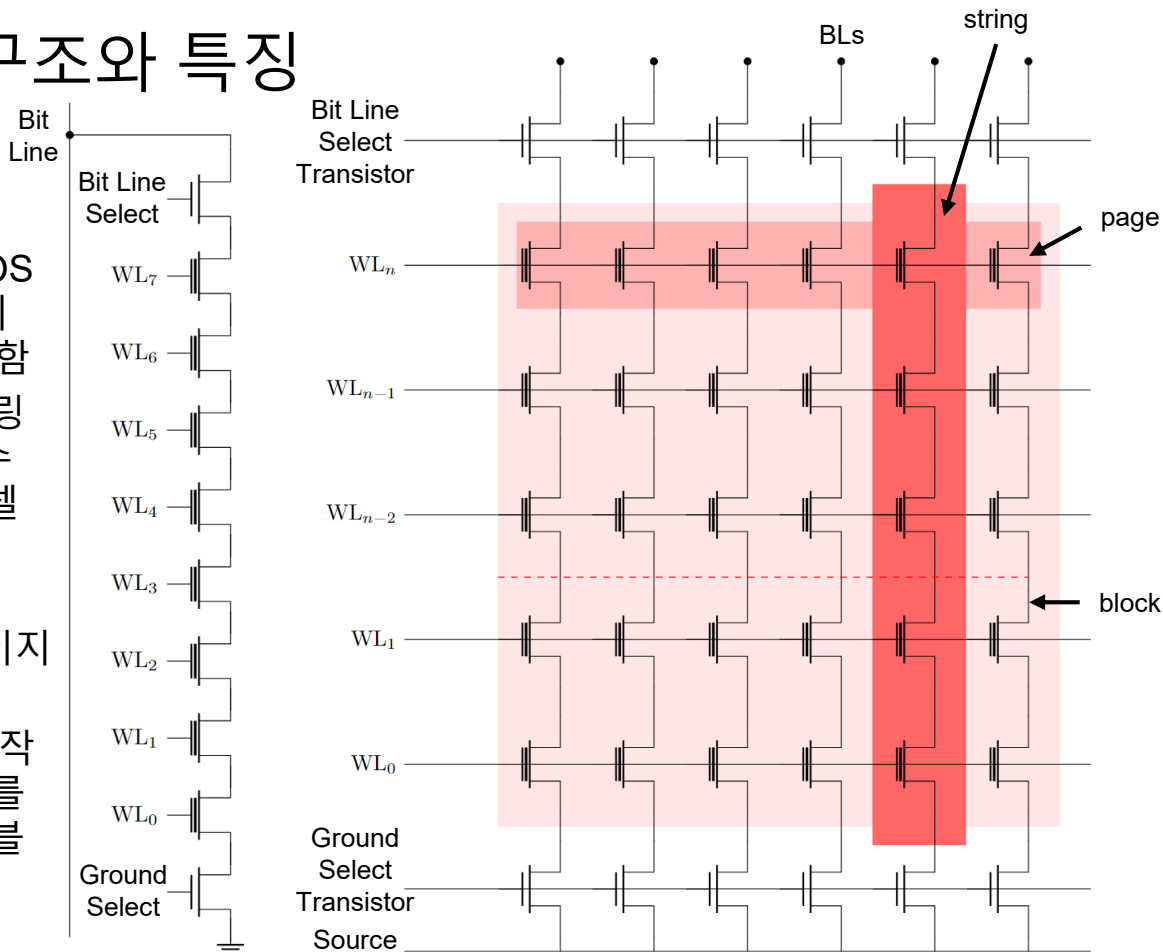
- 메모리 셀을 2차원 행렬로 배열하고 있으며, 이것을 **메모리 어레이**(array)라고 함
- 1개의 FGMOS가 1개의 메모리 셀에 해당됨
- 메모리 셀 어레이에서는 **워드라인**(WL, word line)이 서로 인접하고 있어 같은 비트에 연결된 FGMOS들이 소스와 드레인을 공유하는 형태로 접속되어 있음
- 같은 **비트 라인**(BL, bit line)에 연결된 FGMOS의 수는 일반적으로 32개나 64개인데, 이를 **스트링**(string)이라고 하며, 스트링은 연결의 최소 단위임



플래시 메모리

■ NAND 플래시의 구조와 특징

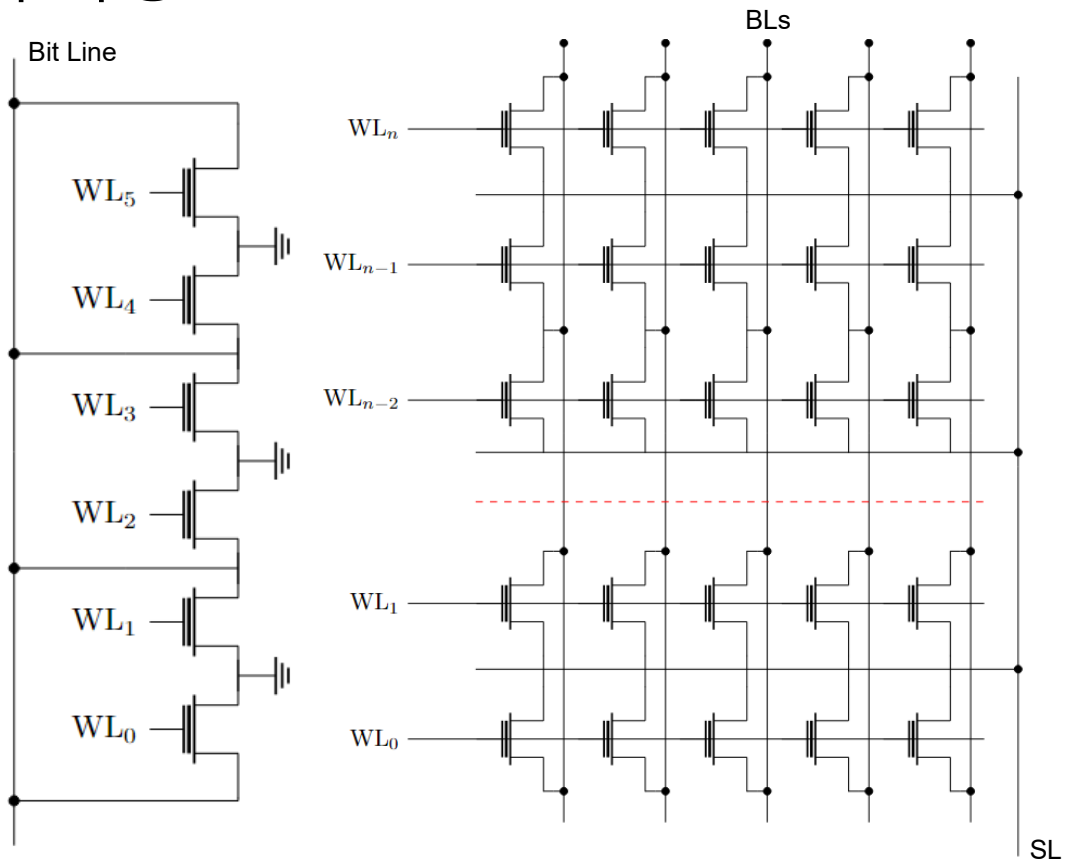
- 같은 워드라인에 연결된 FGMOS의 수는 읽기/쓰기의 최소 단위가 되며, 일반적으로 16K나 32K의 FGMOS가 1개의 워드라인에 연결되며, 이를 **페이지**(page)라고 함
- 워드라인의 개수는 한 스트링 내부에 있는 FGMOS의 개수와 같으므로 1개의 메모리 셀 어레이에는 32개나 64개의 워드라인이 있음
- **블록**(block)은 스트링과 페이지로 구성됨
- 페이지 단위로 읽기/쓰기 동작이 가능하지만 해당 페이지를 덮어쓰거나 지우려면 모든 블록을 재워야 함



플래시 메모리

■ NOR 플래시의 구조와 특징

- 1개의 FGMOS가 1개의 메모리 셀에 해당됨
- 각각의 FGMOS는 독립되어 있으며, 워드라인(WL)과 비트라인(BL) 이외에 **소스라인(SL)**이 있으며, 이는 메모리 셀 어레이에 공통이며, FGMOS의 소스 전압을 고정하기 위한 용도로 사용됨
- 랜덤 액세스 방식으로 바이트 또는 워드 단위로 읽기/쓰기 동작이 가능하지만 덮어쓰기와 지우기 동작은 임의로 접근할 수 없음



플래시 메모리

■ NAND 플래시와 NOR 플래시의 비교

구분	NAND 플래시	NOR 플래시
용도	데이터 저장용	프로그램 코드 저장용
읽기 속도	느리다	빠르다
쓰기 속도	빠르다	느리다
지우기 속도	빠르다	느리다
구조	셀이 직렬로 연결 데이터/주소 통합 구조	셀이 병렬로 연결 데이터/주소 분리 구조
액세스 단위	페이지 및 블록	워드 및 바이트
랜덤 액세스	Data Read 시 불가능	Data Read 시 가능
불량 섹터	있다	없다
단가	단가 낮음	단가 높음
저장 용량	대용량	소용량
사용 기기	USB 드라이브, 메모리 카드에 이용	휴대폰, 셋톱박스용 칩에 사용
주도 업체	삼성전자, 도시바	인텔, AMD

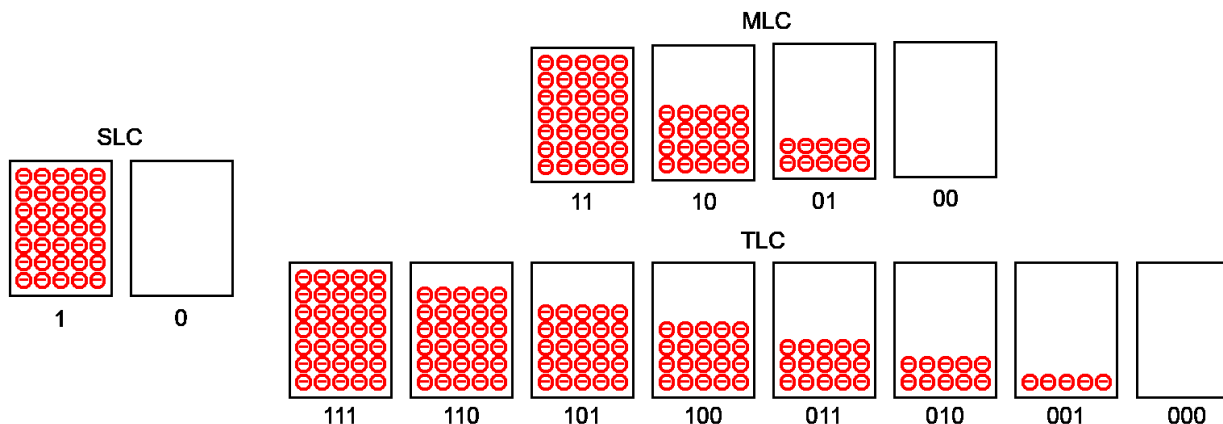
플래시 메모리

■ SLC, MLC, TLC의 차이와 특성

- 데이터를 저장하는 최소 단위인 메모리 셀에 데이터를 저장하는 방식에 따라 3가지 종류로 나눌 수 있음

- SLC(single level cell): 하나의 셀에 1비트의 정보를 저장 0, 1
- MLC(multi level cell) : 하나의 셀에 2비트의 정보를 저장 00, 01, 10, 11
- TLC(triple level cell) : 하나의 셀에 3비트의 정보를 저장 000, 001, 010, 011, 100, 101, 110, 111

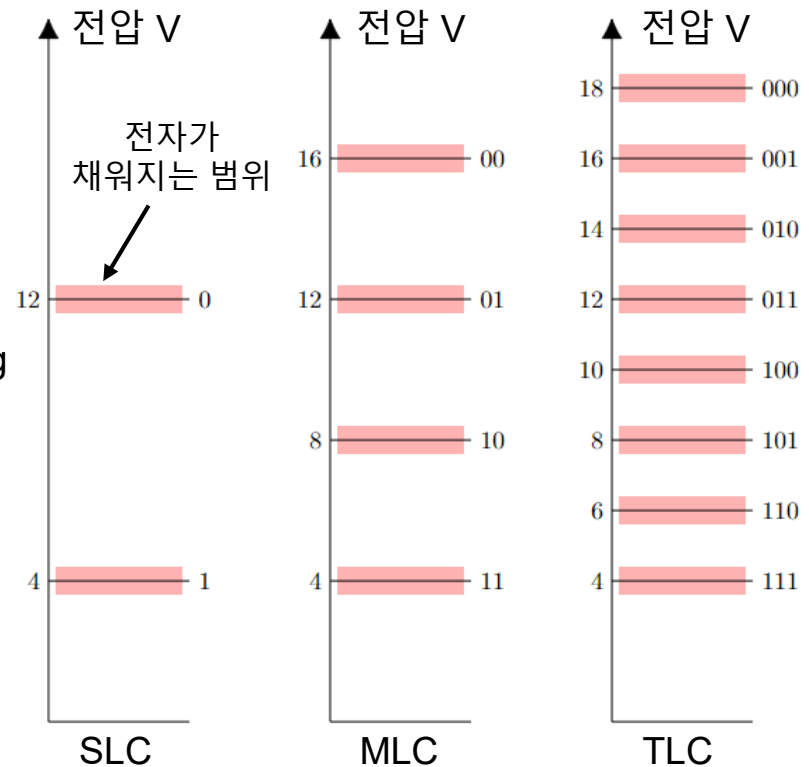
- 제어 게이트에 인가하는 전압에 따라 플로팅 게이트에 주입되는 전자의 양은 달라짐



플래시 메모리

■ SLC, MLC, TLC의 차이와 특성

- 데이터의 저장과 읽기에 오차 허용 범위를 두고 있음
- SLC는 각 데이터를 규정하는 범위 간에 간격이 충분히 넓기 때문에 데이터를 저장하고 읽는 데 오류가 적음
- TLC는 영역 간에 간섭이 일어나지 않을 정도로 촘촘하게 셀을 운용하기 때문에 데이터를 저장하고 읽는 데 오류 발생 확률이 높음
- 오류를 검출하고 정정할 수 있도록 데이터와 함께 오류 정정 코드인 **ECC** 코드(error correcting code)를 함께 저장함
- 오류 발생 확률에 따라 ECC 코드가 복잡해짐
- 전체적으로 SLC, MLC, TLC 순으로 성능이 현저하게 떨어짐
- 플래시 메모리에서 셀의 수명이 한정되어 있음
- SLC, MLC, TLC 순으로 수명이 급격하게 줄어 들음



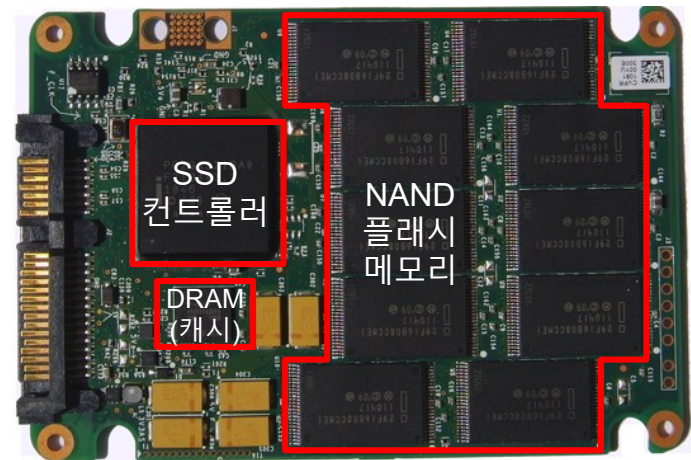
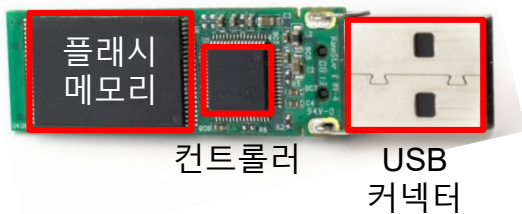
플래시 메모리

- SLC, MLC, TLC의 차이와 특성

구분	SLC	MLC	TLC
셀당 비트 수(bit/cell)	1	2	3
재기록 가능 횟수	100,000	3,000~10,000	1,000
읽기 시간(read time)	25us	50us	~75us
쓰기 시간(write time)	200us~300us	600us~900us	~900us~1,350us
지우기 시간(erase time)	1.5ms~2ms	3ms	4.5ms

플래시 메모리

- 플래시 메모리의 응용
 - USB(universal serial bus) 메모리
 - CF(compact flash) 카드
 - SMC(smart media card)/XD 픽처 카드
 - MMC(multi media card)/MMC Micro 메모리/RS-MMC
 - SD(secure digital) 카드/미니 SD 카드
 - MS(memory stick)
 - SSD



프로그래머블 논리 장치

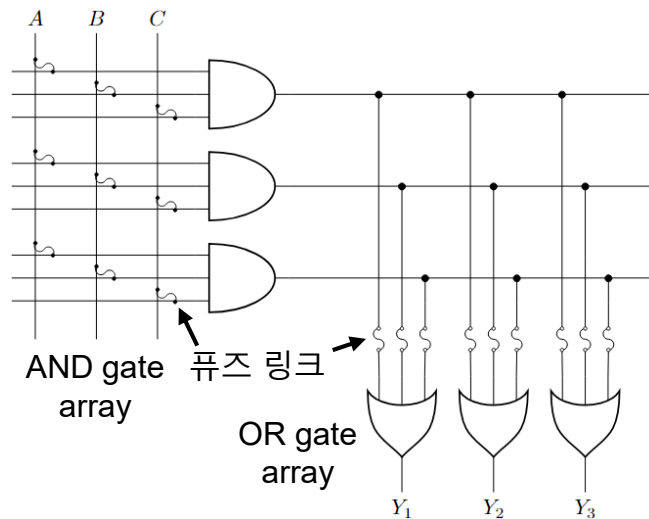
- PLD(programmable logic device)
 - 고정 기능 논리 소자에 비해 공간 절약, 사용의 편리성, 비용 절감, 집적도, 프로그램이 가능하다는 장점을 가짐
 - PLD의 종류
 - SPLD(simple programmable logic device)
 - CPLD(complex programmable logic device)
 - FPGA(field-programmable gate array)
- SPLD
 - AND 게이트와 OR 게이트의 대형 어레이(array) 구조를 갖는 IC로, 각 게이트 입력에 퓨즈 링크(fuse link)가 연결되어 있음
 - 사용자가 적당한 곳의 퓨즈 링크를 전기적으로 끊음으로써 AND-OR, 즉 곱의 합(SOP) 형식으로 된 조합논리회로를 실현할 수 있음
 - 어떤 PLD는 플립플롭을 거쳐 순서논리회로를 실현할 수 있음

프로그래머블 논리 장치

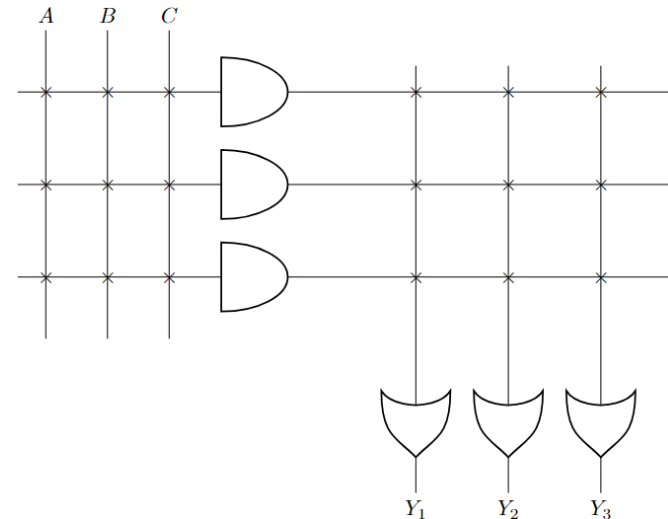
■ SPLD

- SPLD 구조는 AND 어레이, OR 어레이 및 퓨즈 링크로 구성됨
- 사용자가 원하는 퓨즈 링크를 끊는 것을 프로그래밍한다고 하며, 이것은 전용 장치를 사용해 쉽게 수행 할 수 있음

SPLD의 퓨즈 링크



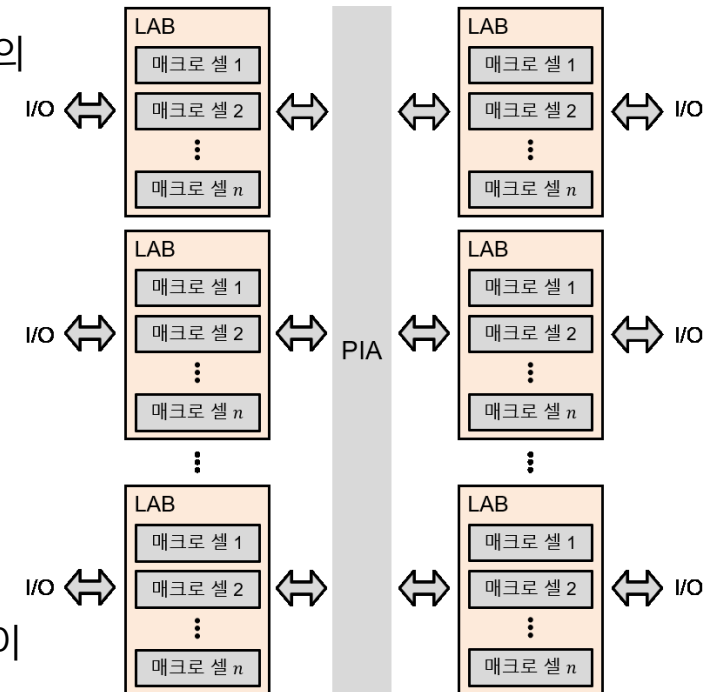
SPLD의 개략도



프로그래머블 논리 장치

■ CPLD

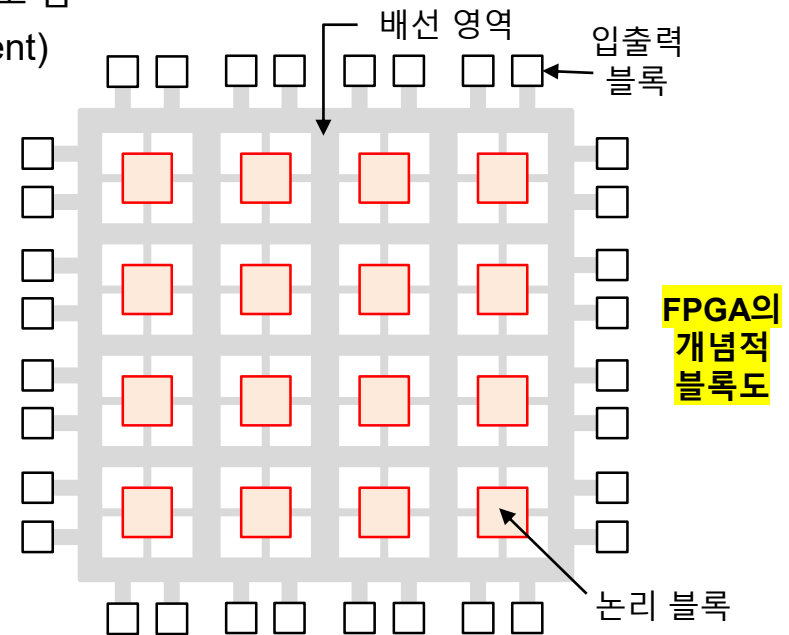
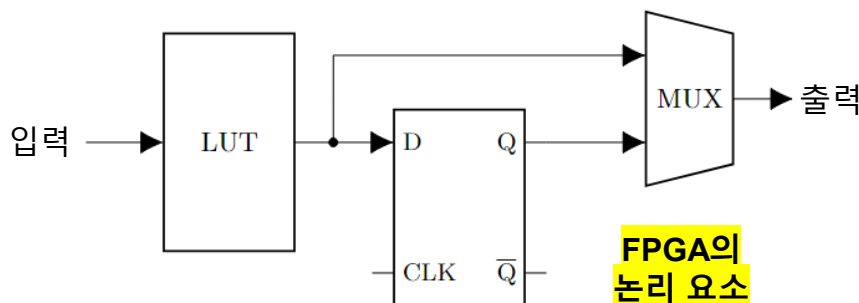
- SPLD보다 집적도가 훨씬 높기 때문에 복잡하고 더 많은 논리회로를 프로그래밍할 수 있으며, 보통 2~64개의 SPLD와 등가로 볼 수 있음
- CPLD 내부에 여러 개의 **LAB**(logic array block)와 LAB의 연결선인 **PIA**(programmable interconnection array)로 구성되어 있음
- 각 LAB는 **메트로 셀**(macro cell)이라는 여러 개의 PAL/GAL 어레이를 가지고 있으며 PIA를 이용해 다른 LAB 또는 I/O에 연결함으로써 복잡한 대형 논리회로를 구성할 수 있음
- 메트로 셀은 3가지 함수 블록으로 구성되는데, 프로그램 가능한 AND 어레이와 고정된 OR 어레이 및 프로그래머블 레지스터(programmable register)로 이루어짐
- CPLD 구조는 LAB의 연결 상태와 무관하게 신호 지연이 일정하기 때문에 타이밍 시뮬레이션이 필요 없음



프로그래머블 논리 장치

■ FPGA

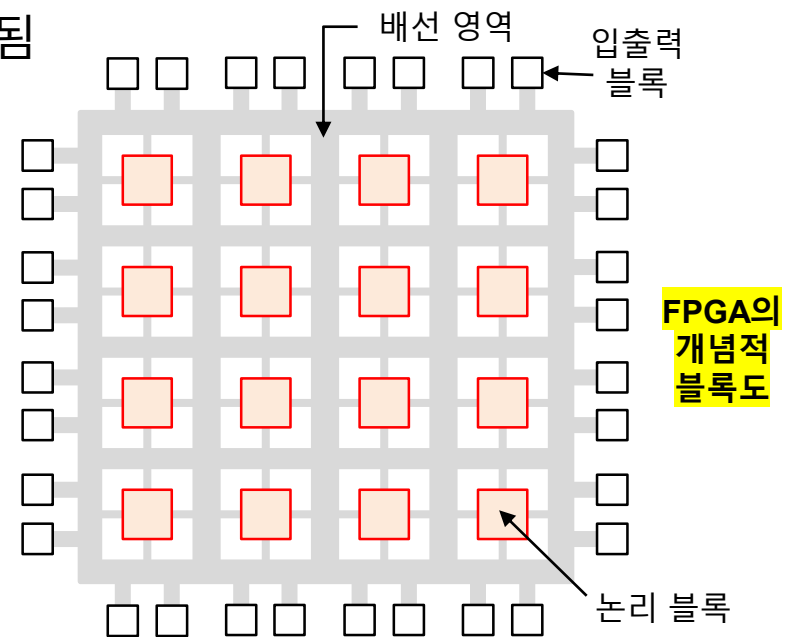
- SPLD나 CPLD와는 다른 내부 구조를 가지고 있으며, 최고의 논리회로 집적도를 갖음
- FPGA 구조는 제조사마다 다르지만 프로그램이 가능한 입출력 블록(I/O block)에 의해 둘러싸인 행과 열의 배선 영역(interconnection area)을 가진 **논리 블록**(logic block) 어레이로 구성됨
- 논리 블록을 **CLB**(configurable logic block)이라고도 함
- 각 논리 블록들은 여러 개의 **논리 요소**(logic element) 들을 갖고 있음
- FPGA는 CPLD의 SOP AND-OR 어레이와 달리 **LUT**(look-up table)라는 일종의 메모리를 기반으로 함



프로그래머블 논리 장치

■ FPGA

- 논리 블록 사이의 내부 결선을 위해 SRAM이나 안티퓨즈(antifuse) 방법을 사용함
 - SRAM을 사용하면 휘발성이 됨
 - 안티퓨즈를 사용하면 비휘발성이 됨
- SRAM을 사용하는 방법이 많이 사용되고 있음
- SRAM을 사용하는 경우 전원이 차단되면 새로 프로그래밍해야 되지만 실제 시스템에 장착된 FPGA의 경우 전원이 공급되면 자동으로 정보를 재저장하도록 설계되어 있음



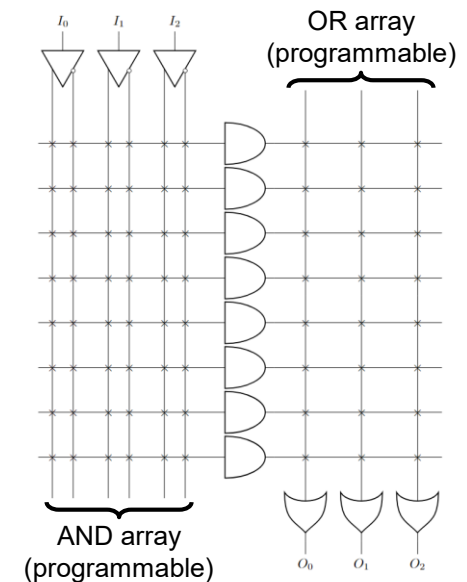
프로그래머블 논리 장치

PLA(programmable logic array)

- ROM 개념과 비슷하지만, 변수들을 완전히 디코딩하지 않으며 ROM처럼 최소항을 만들지도 않음
- AND 게이트와 OR 게이트로 구성되며, AND 게이트들의 집합으로 되므로 각각은 입력변수들의 곱의 항(product term)을 만들어냄
- AND 게이트와 OR 게이트는 처음에 연결 고리들로 모두 연결되어 있음. 특정 불 함수들은 필요 없는 연결 고리들은 끊고, 필요한 연결 고리들은 남겨둬으로써 곱의 합 형식으로 구현할 수 있음

예, 3입력-3출력 PLA의 구조를 고려함

- x 표시는 수평선과 수직선 간의 퓨즈 링크를 나타냄
- 입력은 반전 버퍼를 가지고 있고 각 AND 게이트는 퓨즈 링크가 붙은 입력 6개($I_0, \bar{I}_0, I_1, \bar{I}_1, I_2, \bar{I}_2$)를 가지고 있음
- 각 AND 게이트의 출력은 퓨즈 링크를 거쳐서 각 OR 게이트의 입력에 연결되어 있음
- PLA로 실현할 논리 함수는 SOP 형식이어야 함
- 프로그래밍은 AND 게이트 쪽부터 시작하며, AND 게이트 출력 중에서 필요한 것만 선택하여 OR 함으로써 필요한 출력을 얻기 위해서 OR 게이트 입력 쪽에서 프로그래밍을 실시함



프로그래머블 논리 장치

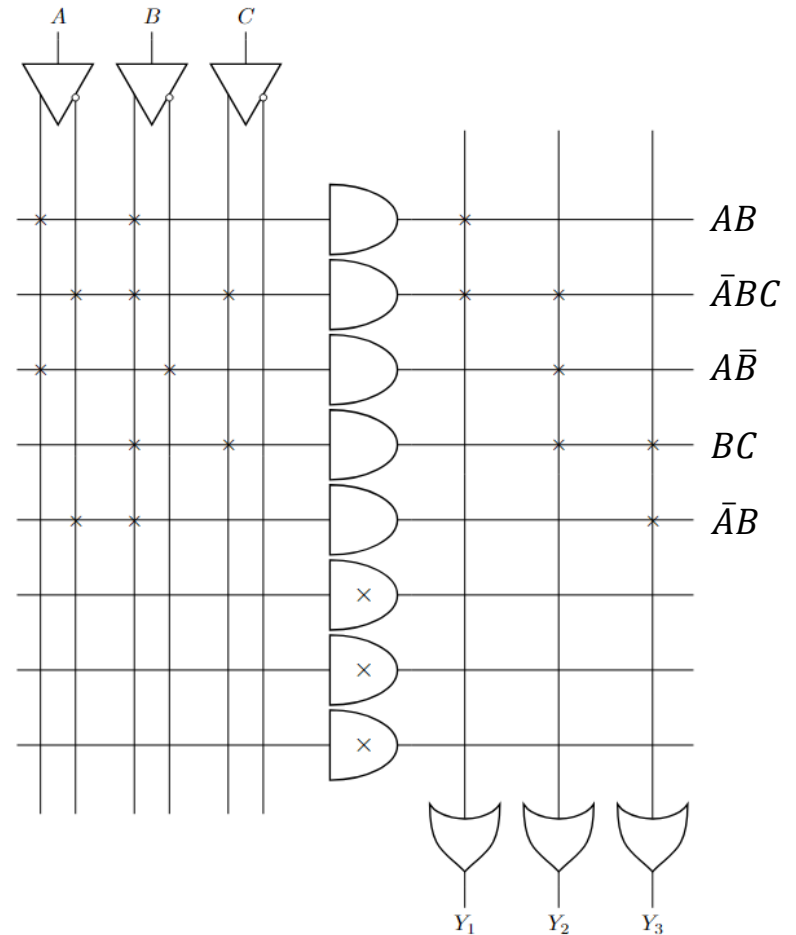
- PLA(programmable logic array)

- 예, 다음과 같은 논리식을 PLA로 구현함

$$Y_1 = AB + \bar{A}BC$$

$$Y_2 = A\bar{B} + BC + \bar{A}BC$$

$$Y_3 = \bar{A}B + BC$$



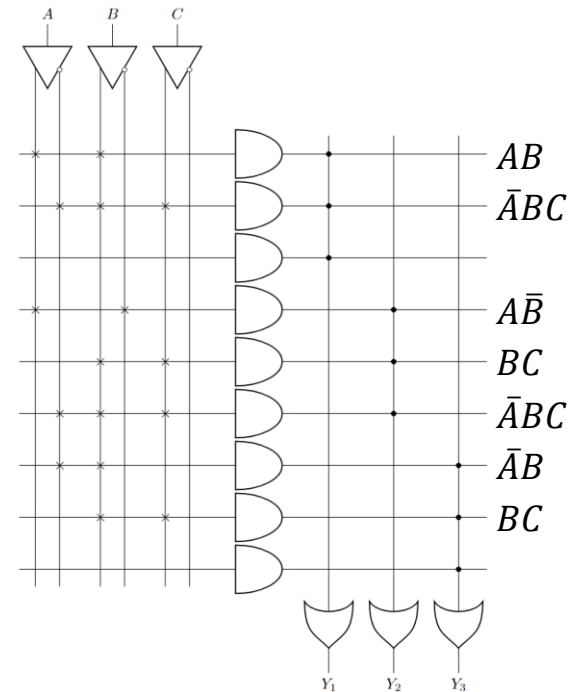
프로그래머블 논리 장치

- PAL(programmable array logic)
 - OR 입력이 제조 시부터 고정되어 있고 AND 입력만을 프로그래밍할 수 있음
 - PLA보다는 프로그래밍에 제한이 있지만 현재 가장 많이 쓰이는 PLD임
- 예, 다음과 같은 논리식을 PAL로 구현함

$$Y_1 = AB + \bar{A}BC$$

$$Y_2 = A\bar{B} + BC + \bar{A}BC$$

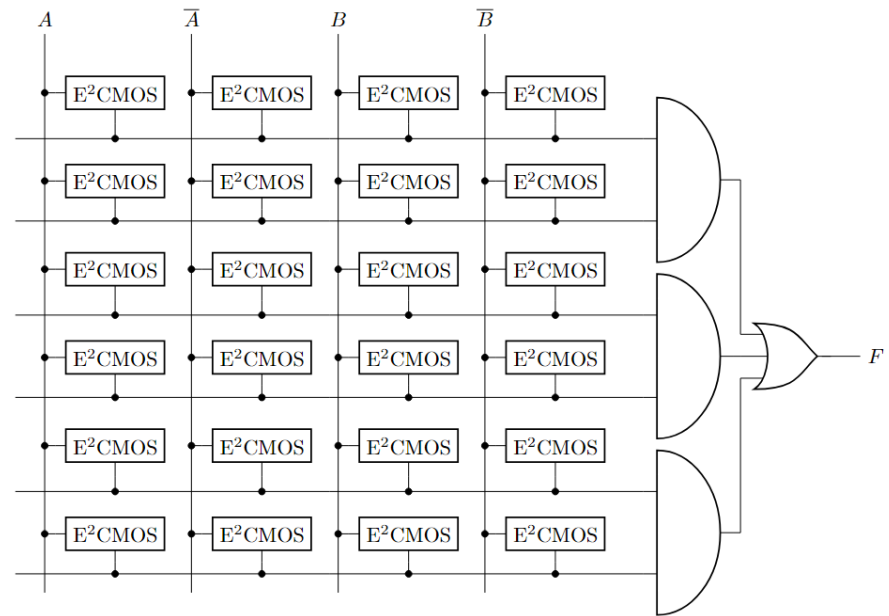
$$Y_3 = \bar{A}B + BC$$



프로그래머블 논리 장치

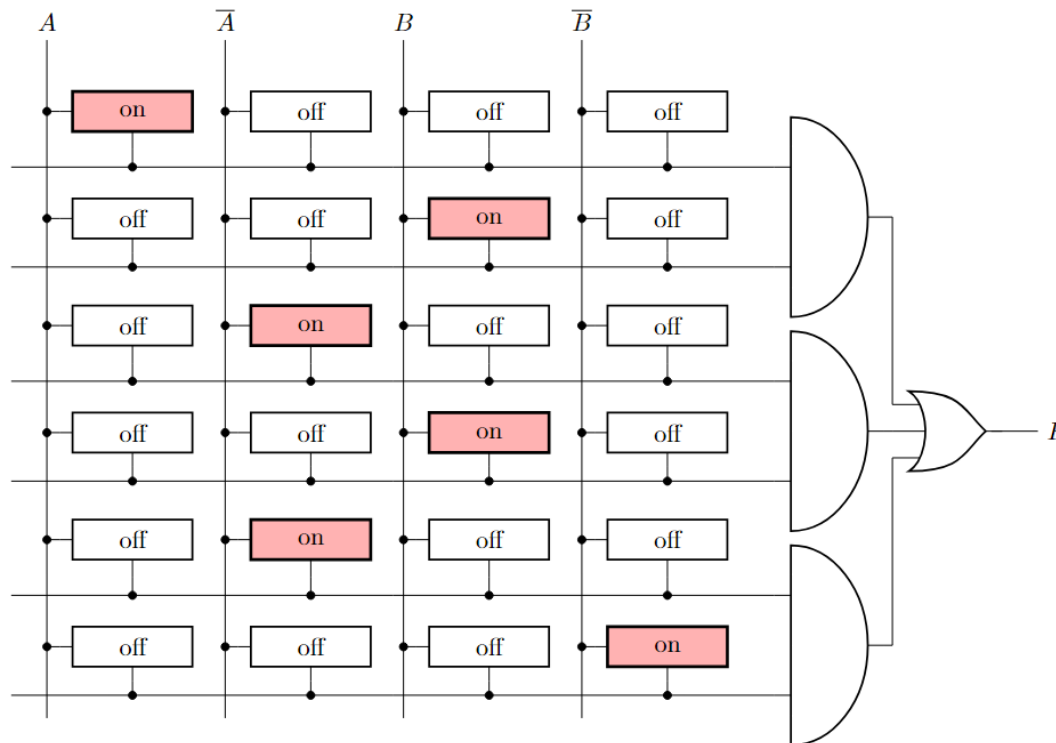
■ GAL(generic array logic)

- 반복적으로 프로그래밍 가능한 AND 어레이가 고정 OR 어레이에 연결된 구조이기 때문에 PAL과 마찬가지로 어떠한 SOP 형태의 논리식도 구현할 수 있음
- 반복적으로 프로그래밍 가능한 어레이는 행과 열로 된 도체의 격자로 각 교차점은 PAL의 퓨즈와는 달리 E²CMOS 셀로 구성되어 있음
- 각 행은 AND 게이트의 입력단에 연결되어 있으며 각 열은 입력 변수나 변수의 보수에 연결되어 있음
- E²CMOS 셀들이 on 또는 off 상태가 되도록 적절히 프로그래밍함으로써 입력 변수나 변수의 모든 조합이 AND 게이트에 입력되게 할 수 있으며, 따라서 원하는 어떠한 곱의 항 만들 수 있음



프로그래머블 논리 장치

- GAL(generic array logic)
 - 예, 논리 함수 $F = AB + \bar{A}B + \bar{A}\bar{B}$ 를 GAL로 구현함



프로그래머블 논리 장치

- PLD 프로그래밍
 - **결선도 입력**(schematic entry)과 **텍스트-기반 입력**(text-based entry) 방법이 있음
 - 경우에 따라 두 방법을 같이 사용하기도 함
 - **결선도 입력 방법**
 - 사용자가 소프트웨어를 이용해 논리 소자(게이트, 플립플롭 등)등을 입력시키고 컴퓨터 스크린 상에서 이들을 상호 연결하여 결선도를 만듦
 - SPLD 프로그래밍에 사용하는 소프트웨어는 WinCUPL, ABEL, PALASM, OrCAD-PLD, LOG/iC 등이 있음
 - **텍스트-기반 입력 방법**
 - 사용자가 **하드웨어 기술 언어**(HDL, hardware description language)를 이용해 텍스트 형태로 논리 설계를 입력시킴
 - 현재 광범위하게 사용되는 HDL에는 VHDL과 Verilog가 있음

프로그래머블 논리 장치

PLD 프로그래밍 과정의 흐름도

