09 강

컴퓨터과학 개론

컴퓨터 구조 (1)

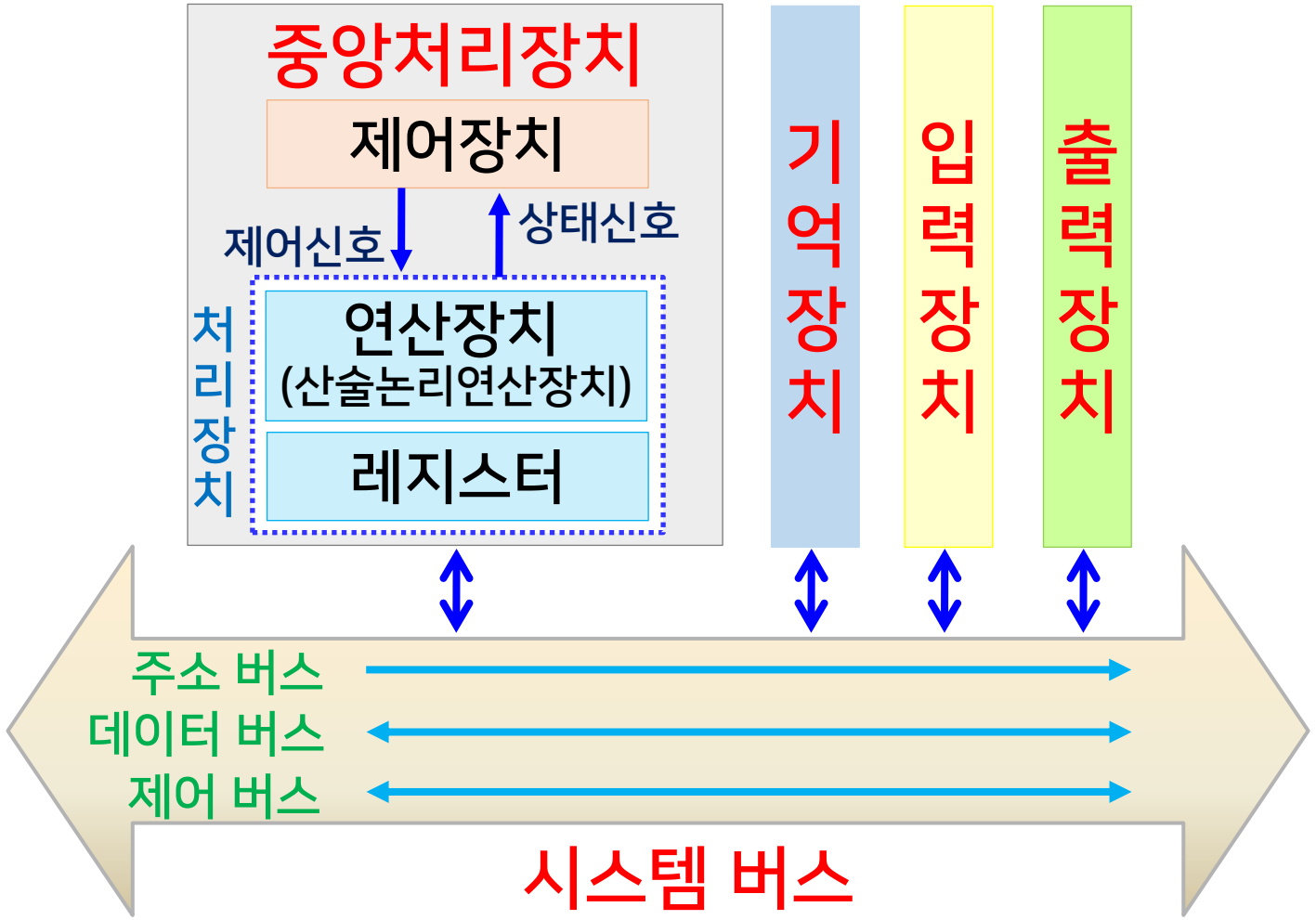
컴퓨터과학과 이관용 교수

학습목차

01

불 대수와 논리 게이트

컴퓨터 하드웨어의 기본 구성



시스템 버스

중앙처리장치, 기억장치, 입출력장치 간의 물리적 연결 및 데이터 교환의 통로 역할을 하는 버스

- 전송되는 데이터 내용에 따른 구분 → 주소 버스, 데이터 버스, 제어 버스

주소 버스

- CPU가 기억장치나 입출력장치의 주소 정보를 전송하는 신호선의 집합
- 버스의 폭이 시스템의 메모리 용량을 결정
  - → n개의 신호선이면 2^n개의 주소 지정 가능
- 단방향 버스

시스템 버스

데이터 버스

- CPU와 기억장치/입출력장치 사이에 데이터 전송을 위한 신호선의 집합
- 버스의 폭 → 한 번에 전송할 수 있는 비트의 수 → "워드"
- 양방향 버스

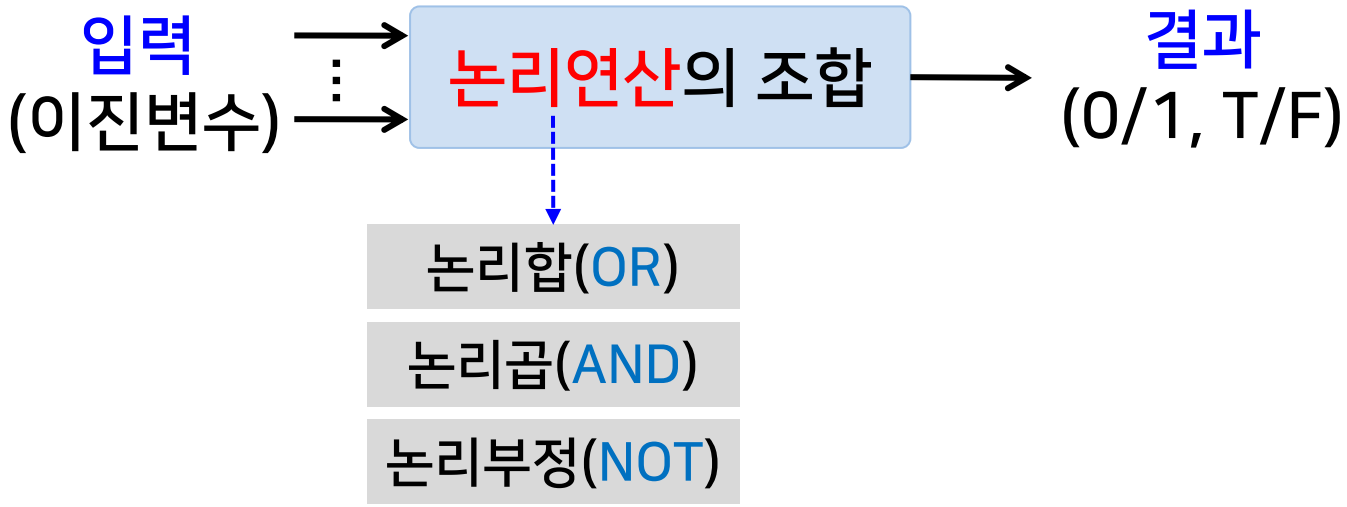
제어 버스

- CPU와 시스템 내의 각종 장치의 동작을 제어하기 위한 다양한 신호들의 통로
- 버스의 폭(제어신호의 수)은 CPU나 시스템의 구성에 따라 달라짐
- 양방향 버스

불 대수 Boolean Algebra

이진 변수의 논리연산을 다루는 대수

- 영국의 수학자 조지 불 George Boole
- 논리적 문제를 해결하기 위한 수학적 방법 → "논리 대수", "스위치 대수"



기본 논리연산의 진리표

논리합
Union, OR

입력		출력
X	Y	$F = X + Y$
0	0	0
0	1	1
1	0	1
1	1	1

$X \cup Y, X \vee Y$

논리곱
Intersection, AND

입력		출력
X	Y	$F = X \cdot Y$
0	0	0
0	1	0
1	0	0
1	1	1

$XY, X \cap Y, X \wedge Y$

논리부정
Negation, NOT

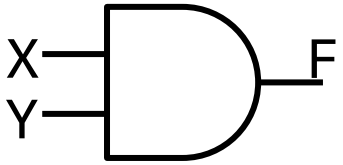
입력	출력
X	$F = X'$
0	1
1	0

$\sim X, \overline{X}$

논리 게이트

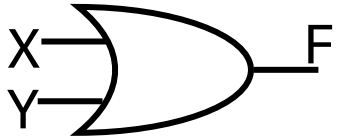
논리연산을 하드웨어로 구현한 것

- 논리회로를 구성하는 기본 소자



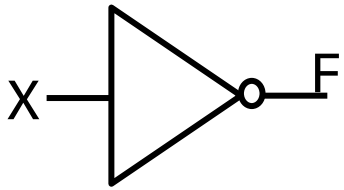
AND

$F = X \cdot Y$



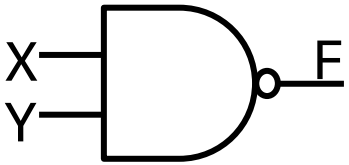
OR

$F = X + Y$



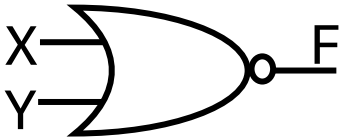
NOT

$F = X'$



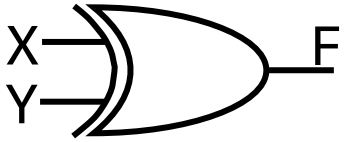
NAND

$F = (X \cdot Y)'$



NOR

$F = (X + Y)'$



XOR

$F = X' \cdot Y + X \cdot Y'$

복합 논리연산의 진리표

NAND

입력		출력
X	Y	F
0	0	1
0	1	1
1	0	1
1	1	0

NOR

입력		출력
X	Y	F
0	0	1
0	1	0
1	0	0
1	1	0

XOR

입력		출력
X	Y	F
0	0	0
0	1	1
1	0	1
1	1	0

논리 게이트의 완전 집합

원하는 임의의 회로를 구성할 수 있는 게이트들의 부분집합

- { NOT, AND }, { NOT, OR },
  { AND, XOR }, { OR, XOR },
  { NAND }, { NOR } 등

- NAND 게이트만 사용 → NOT, OR, AND의 기능 구현이 가능

$$X' = (X \cdot X)'$$

$$X + Y = ((X \cdot X)' \cdot (Y \cdot Y)')'$$

$$X \cdot Y = ((X \cdot Y)' \cdot (X \cdot Y)')'$$

불 대수의 기본 법칙

항등법칙	$X + 0 = X$ $X + 1 = 1$	$X \cdot 0 = 0$ $X \cdot 1 = X$
멱등법칙	$X + X = X$	$X \cdot X = X$
지배법칙	$X + X' = 1$	$X \cdot X' = 0$
교환법칙	$X + Y = Y + X$	$X \cdot Y = Y \cdot X$
결합법칙	$X + (Y + Z) = (X + Y) + Z$	$X \cdot (Y \cdot Z) = (X \cdot Y) \cdot Z$
분배법칙	$X + (Y \cdot Z) = (X + Y)(X + Z)$	$X \cdot (Y + Z) = (X \cdot Y) + (X \cdot Z)$
드모르간 법칙	$(X + Y)' = X' \cdot Y'$	$(X \cdot Y)' = X' + Y'$
흡수법칙	$X + X \cdot Y = X$	$X \cdot (X + Y) = X$
이중 부정	$(X')' = X$	

불 대수와 논리회로

- ‘불 함수’ ⟷ ‘논리회로’ 형태로 표현 가능

- 불 함수의 간소화

F = AB'+B
← 교환법칙
= B+AB'
← 분배법칙
= (B+A)(B+B')
← 지배법칙
= (B+A)·1
← 항등법칙
= A+B

F = AB+AC+AB'C'
← 분배법칙
= A(B+C+B'C')
← 드모르간 법칙
= A((B+C)+(B+C)')
← 지배법칙
= A·1
← 항등법칙
= A

02

논리회로

논리회로의 종류

논리회로

- 논리 게이트들로 구성된 회로
- 종류 → 조합회로 combinational circuit, 순서회로 순차회로, sequential circuit

조합회로

- 출력값이 단순히 현재 입력값의 조합에 의해서만 결정되는 회로

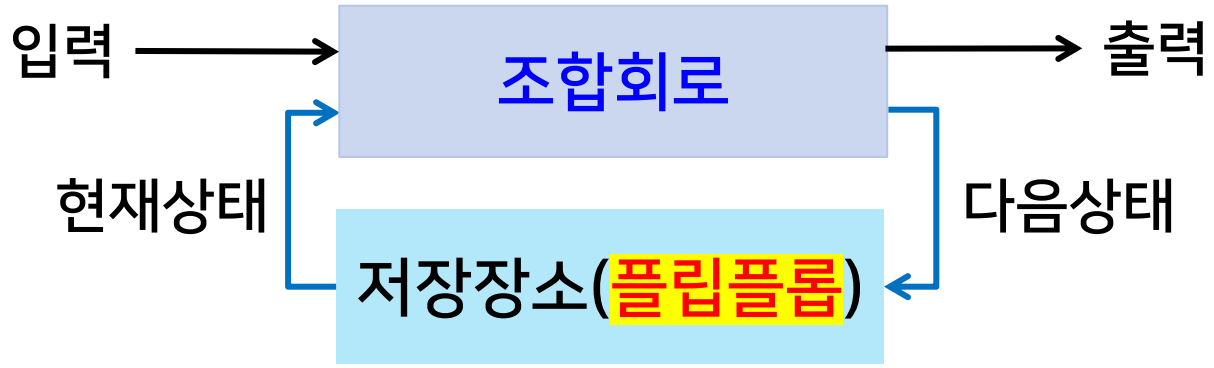
- 전가산기, 디코더, 인코더, 멀티플렉서, 디멀티플렉서 등
  - 주로 연산을 위해서 사용되는 회로가 해당

논리회로의 종류

순서회로

- 연산의 각 단계마다 회로의 특정 상태가 저장되고 참조되는 회로

→ 출력값이 입력값과 기억소자에 저장된 현재 상태에 따라 결정됨



- 카운터, 레지스터 등

플립플롭 flip-flop

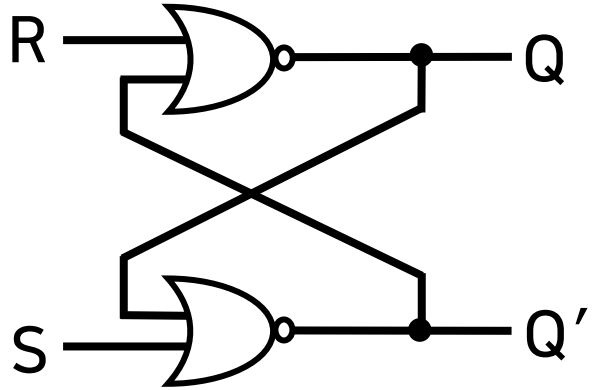
1비트의 이진 정보를 저장할 수 있는 장치

- 입력신호에 의해서 상태를 바꾸도록 지시할 때까지는 현재의 이진 상태를 유지하는 논리소자
- 종류(입력의 개수와 상태를 변화시키는 방법에 따라)
  - RS, T, D, JK 등

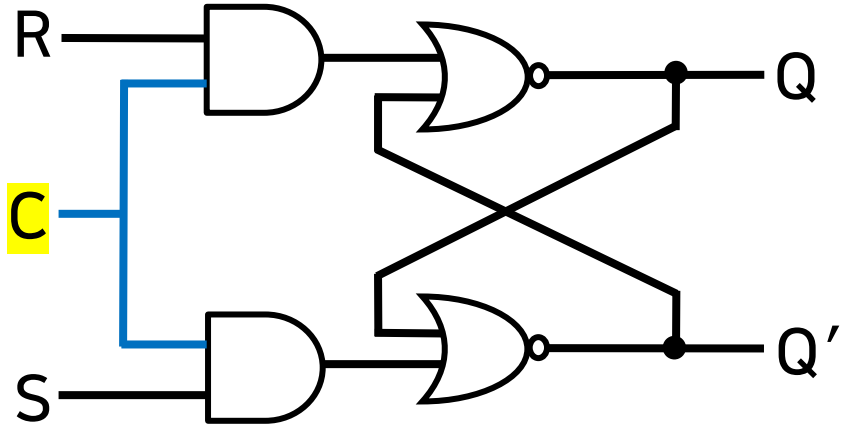
플립플롭

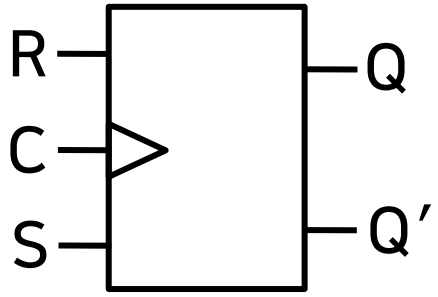
RS Reset-Set 플립플롭

RS 래치



RS 플립플롭

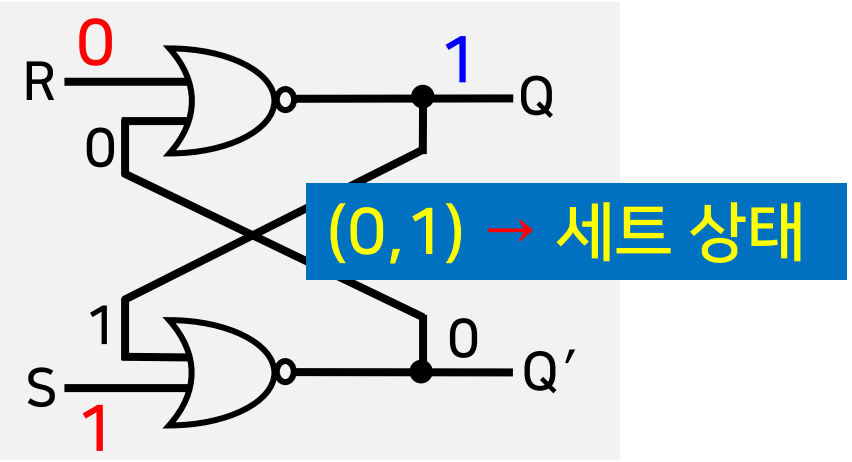


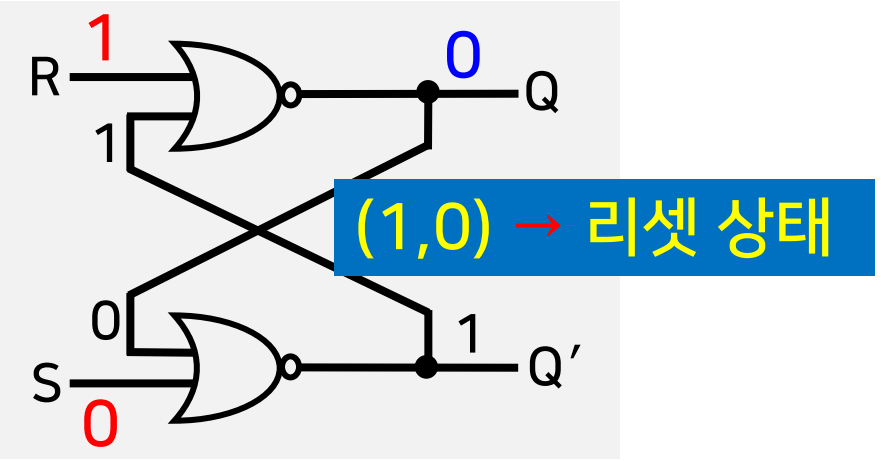


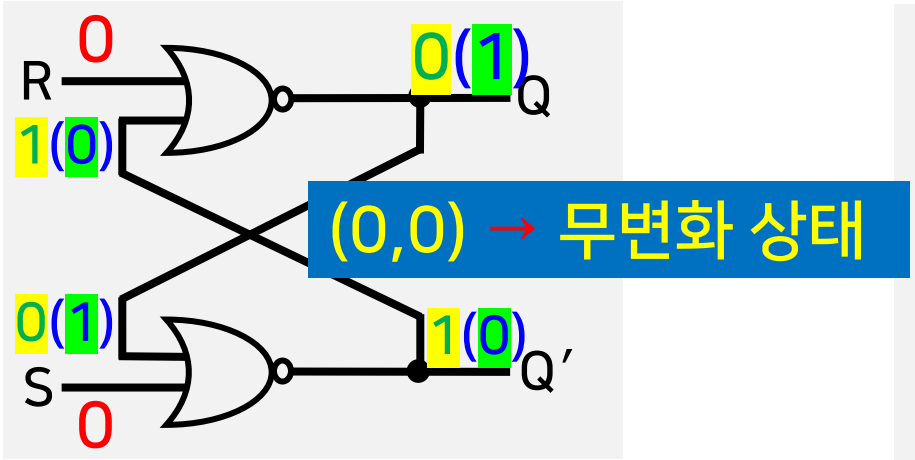
블록도

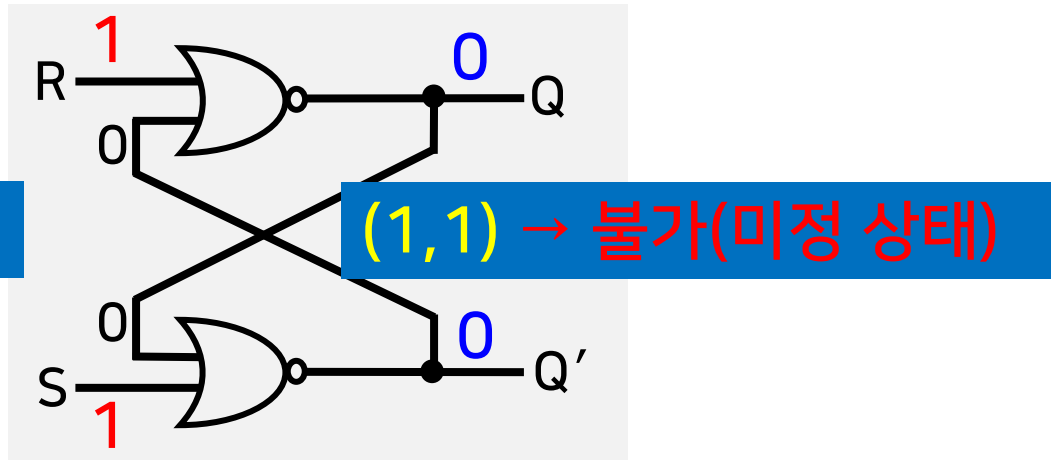
플립플롭

RS 래치의 동작



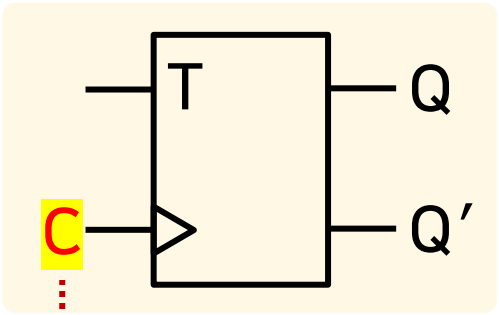






플립플롭

T Toggle 플립플롭



C=1 → 동작

- T=0 → Q(t+1) = Q(t)
- T=1 → Q(t+1) = Q(t)′

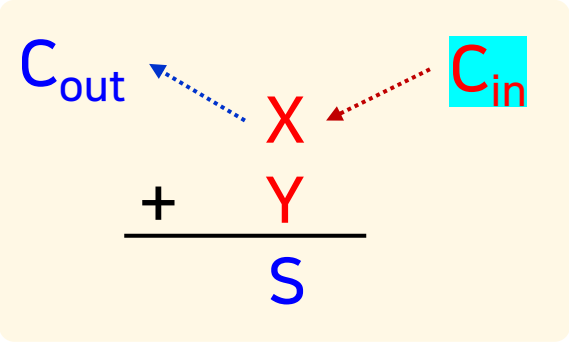
clock pulse

→ 주기적으로 생성되는 신호로
컴퓨터 내의 시스템 구성요소들의 작동을 동기화해 주는 역할

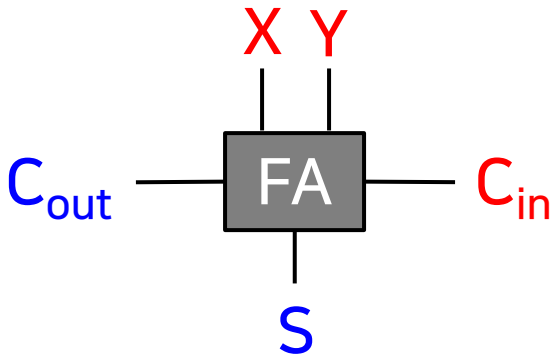
전가산기 Full Adder

아랫자리에서 올라오는 올림수 carry를 고려한 가산기

- 가산기 → 두 이진수를 입력받아서 더하는 회로
- 반가산기 Half Adder → 아랫자리의 올림수를 고려하지 않는 가산기



개념도



블록도

전가산기

진리표 작성

X	0	0	0	0	1	1	1	1
Y	0	0	1	1	0	0	1	1
C_{in}	0	1	0	1	0	1	0	1
S	0	1	1	0	1	0	0	1
C_{out}	0	0	0	1	0	1	1	1

불 대수식 생성 및 간소화

$$S = X' \cdot Y' \cdot C_{in} + X' \cdot Y \cdot C_{in}' + X \cdot Y' \cdot C_{in}' + X \cdot Y \cdot C_{in}$$

$$= X \oplus Y \oplus C_{in}$$

⊕ : XOR

$$C_{out} = X' \cdot Y \cdot C_{in} + X \cdot Y' \cdot C_{in} + X \cdot Y \cdot C_{in}' + X \cdot Y \cdot C_{in}$$

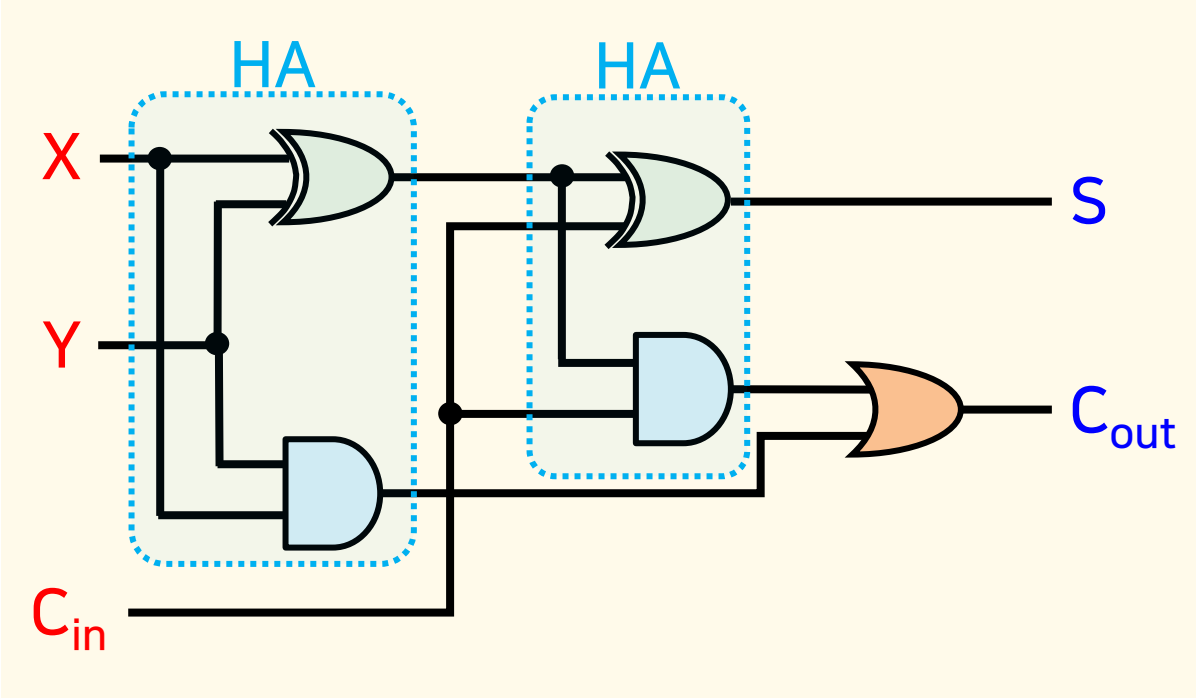
$$= (X \cdot Y) + (C_{in} \cdot (X \oplus Y))$$

전가산기

논리회로도

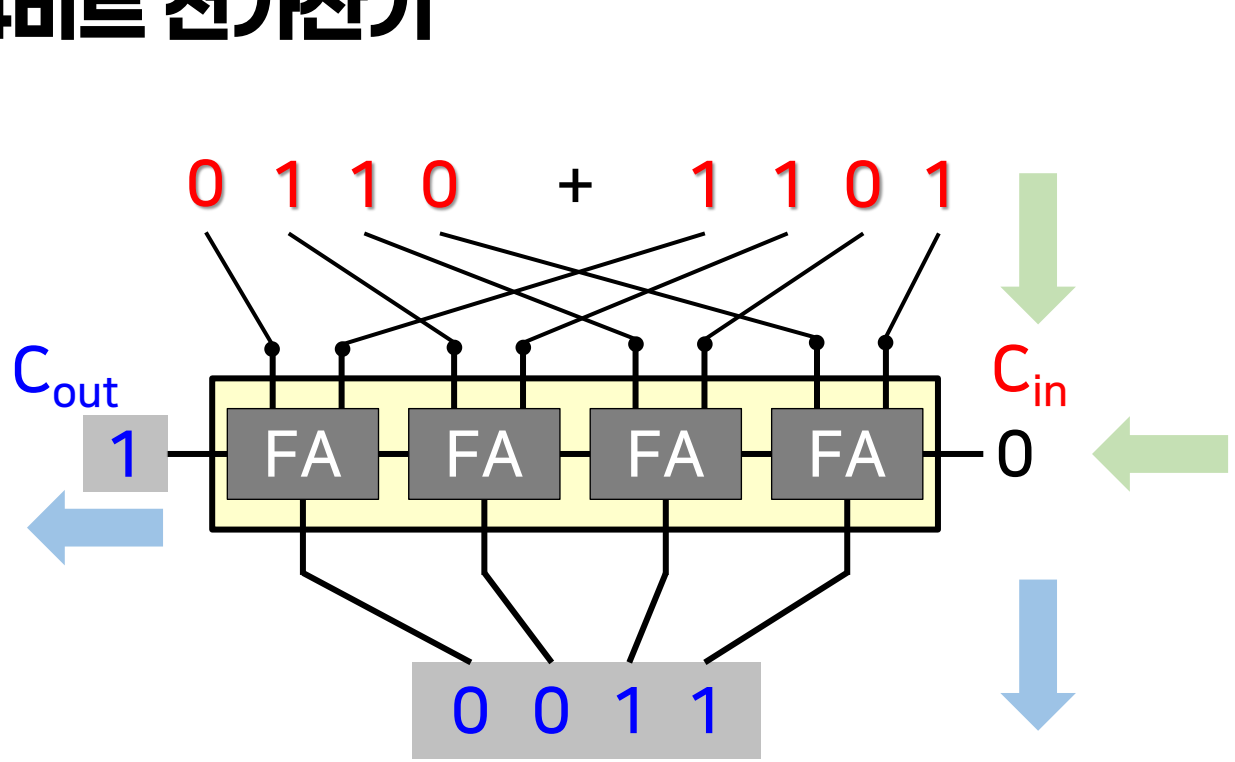
$S = X \oplus Y \oplus C_{in}$

$C_{out} = (X \cdot Y) + (C_{in} \cdot (X \oplus Y))$



전가산기

4비트 전가산기

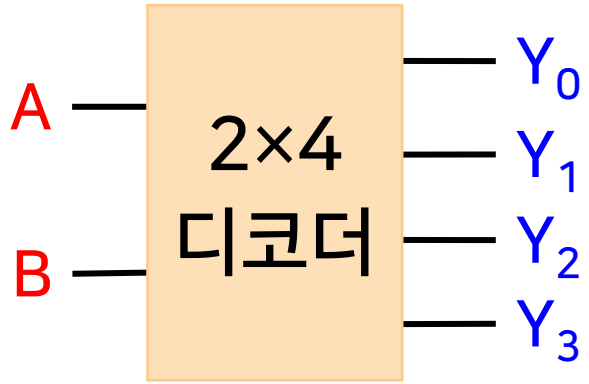


입력의 흐름

정보의 흐름

디코더

- n비트의 이진 코드를 최대 2^n개의 서로 다른 정보로 변환하는 장치
  - 출력 → 2^n개 중에서 오직 1개만 1, 나머지 모두는 0

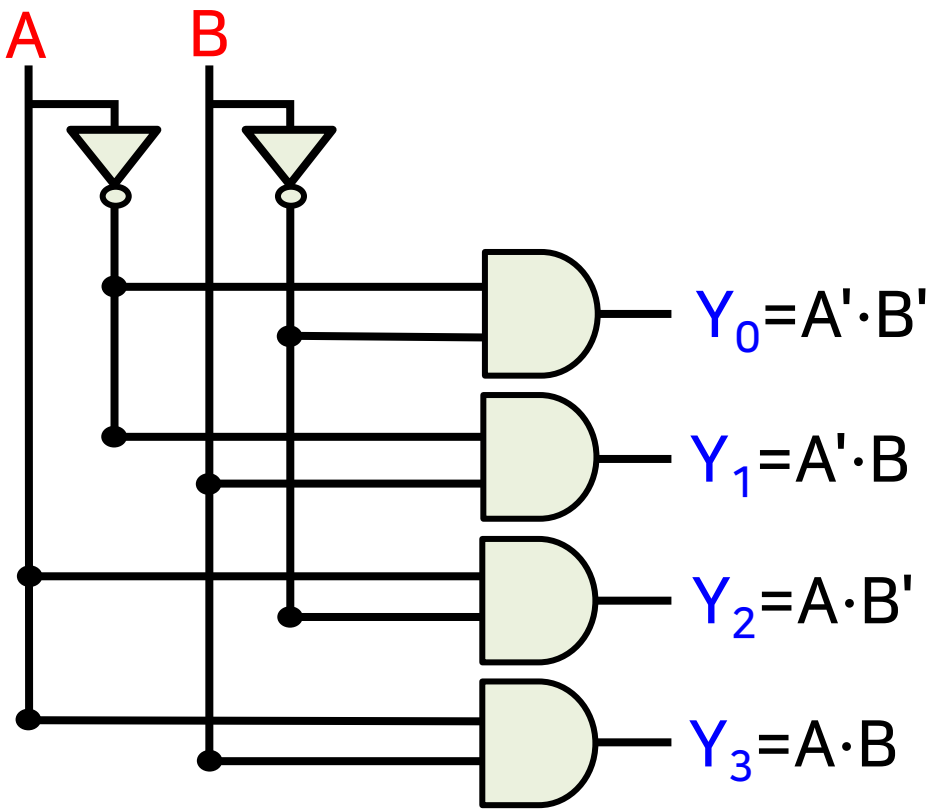


블록도

A	B	Y_0	Y_1	Y_2	Y_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

진리표

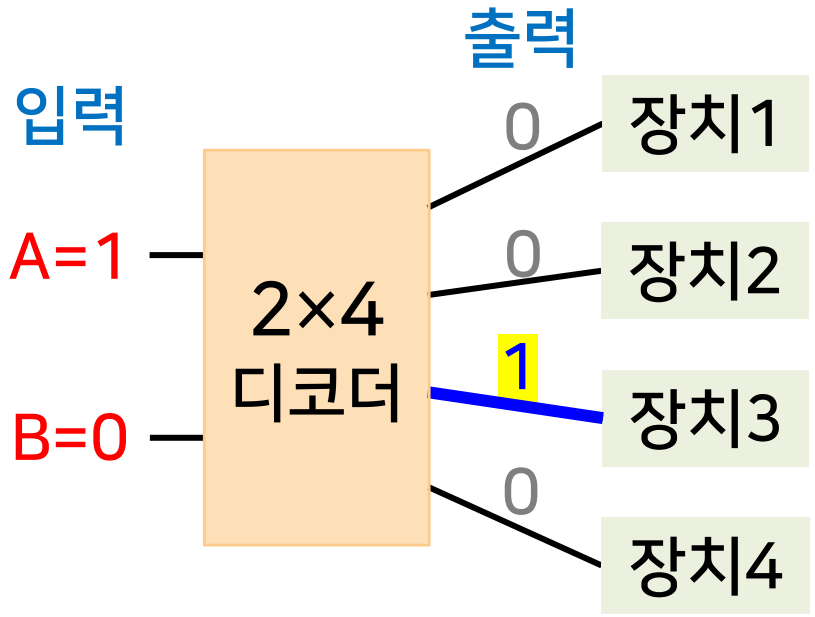
디코더



논리회로도

디코더

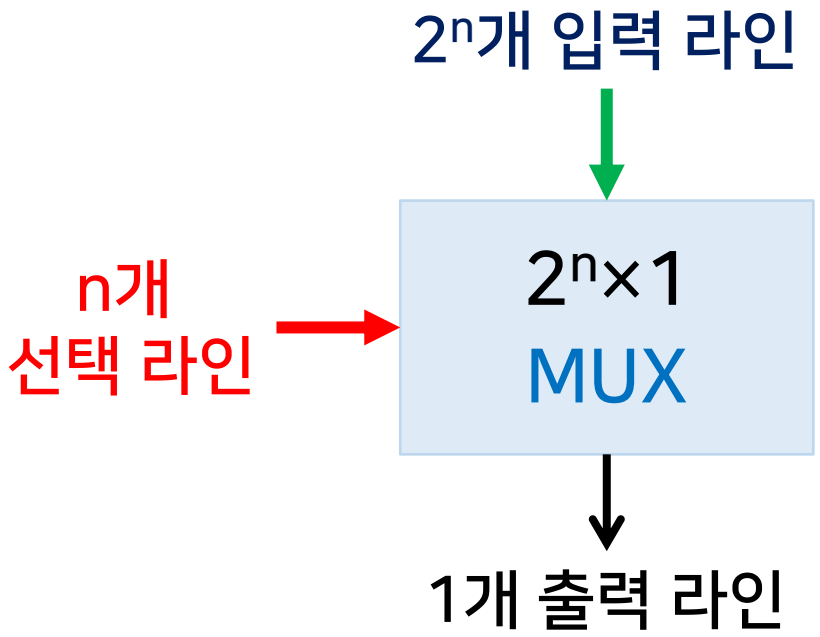
- **주소 방식으로 주어진 입력으로부터 각각의 하드웨어 구성 요소를 개별적으로 구동하기 위해 주로 사용**
  - 주기억장치에 접근할 때도 디코더의 기본 원리가 그대로 적용됨



멀티플렉서 multiplexer

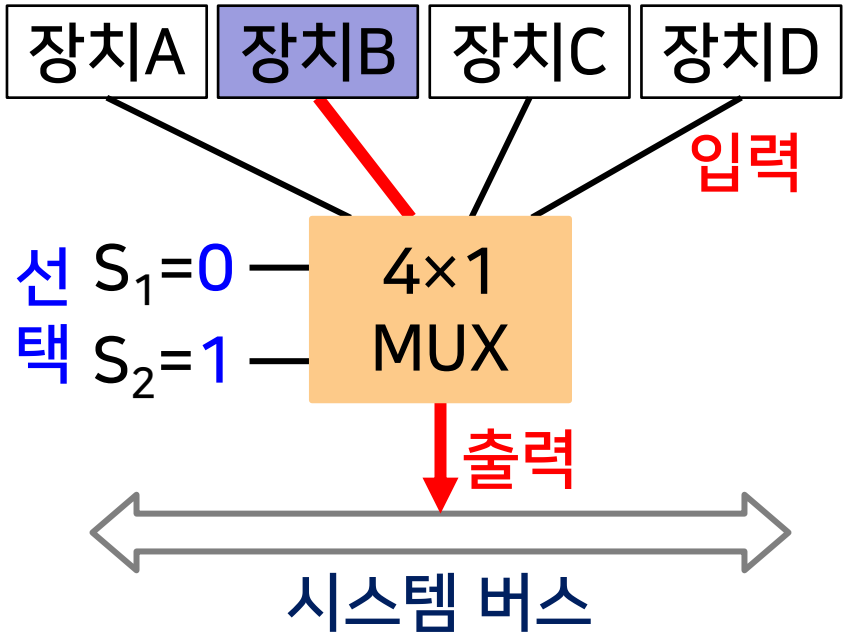
- **여러 개의 입력선 중에서 하나를 선택하여 단일의 출력으로 내보내는 회로**

→ "데이터 선택기"



멀티플렉서

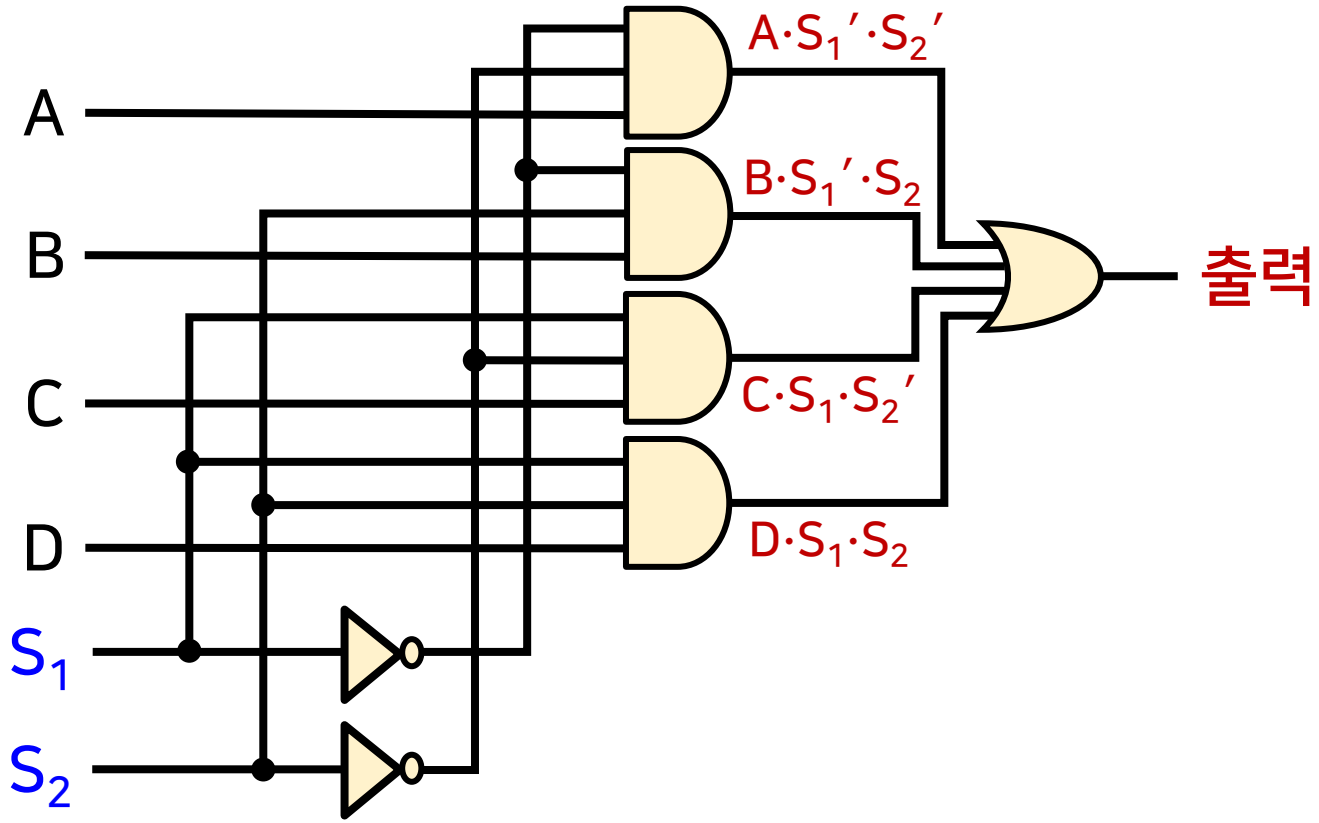
- 어떤 장치(회로)로부터 들어오는 데이터가 버스를 사용할 것인가를 정하는 경우에 사용



S_1	S_2	출력
0	0	A
0	1	B
1	0	C
1	1	D

멀티플렉서

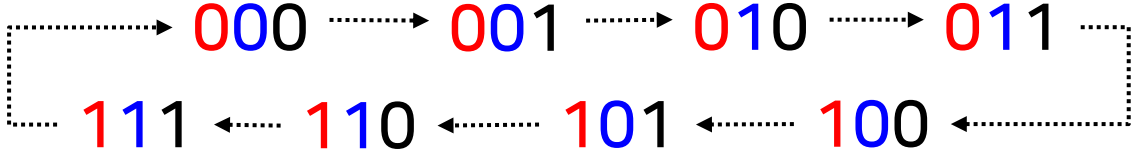
출력 = $(A \cdot S_1' \cdot S_2') + (B \cdot S_1' \cdot S_2) + (C \cdot S_1 \cdot S_2') + (D \cdot S_1 \cdot S_2)$

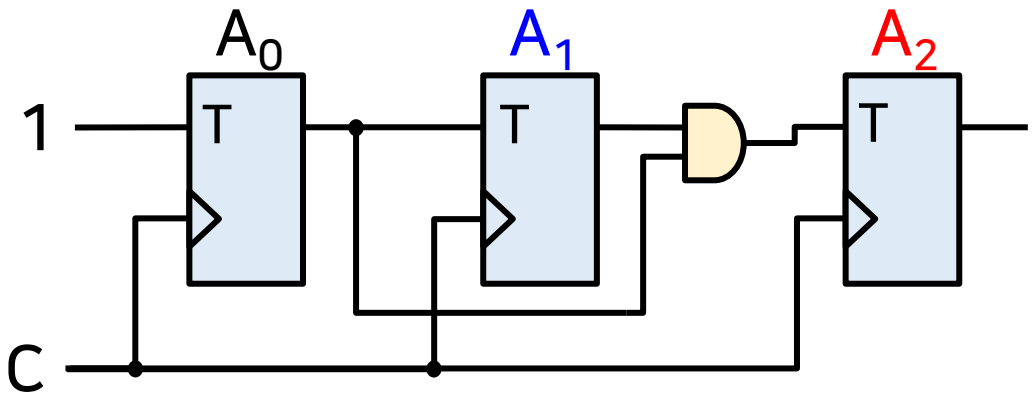


카운터

- **클록펄스가 입력될 때마다 미리 정해진 순서에 따라 상태가 변하는 장치**
  - 매번 구동 시마다 저장된 이진수가 1씩 증가하는 장치
  - 외부에서의 입력/출력이 없으며, 클록펄스를 통해서만 상태가 변함

3비트 카운터





03

기억장치

기억장치

ROM Read-Only Memory

- 읽기 전용 → 내용이 항상 고정되어 있어 조합회로 구성 가능

RAM Random Access Memory

- 읽기/쓰기 모두 가능

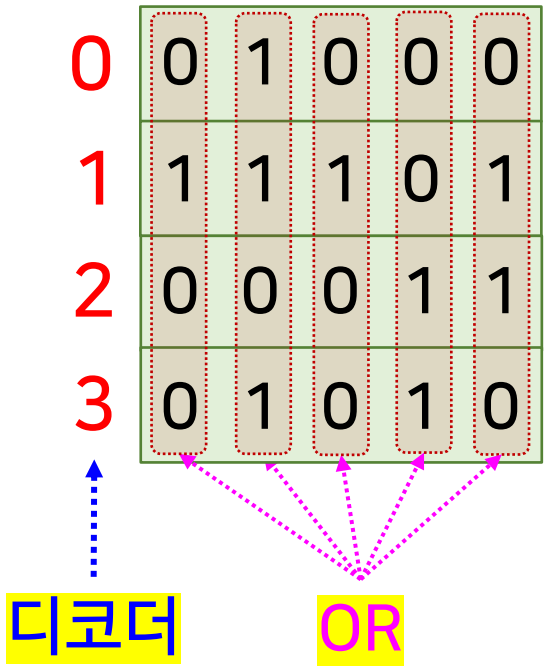
  → 상태를 저장/변경할 수 있는 순서회로로 구성 가능

- DRAM Dynamic RAM
  - 플립플롭이 아닌 축전지로 구현
  - 시간이 지남에 따라 전류가 방전 → 주기적인 리프레시 과정 필요

ROM

- 설계자가 저장되는 이진 정보를 결정하고,
  기억장치 내에서 필요한 내부 연결 패턴을 형성해서 구현

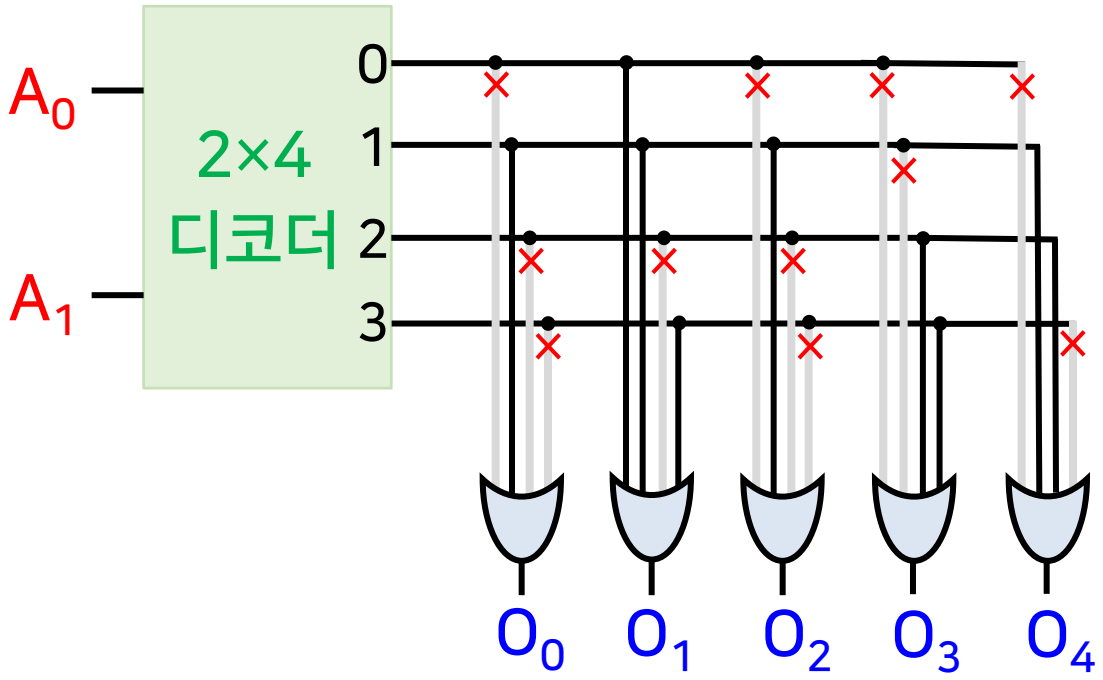
2비트 주소, 5비트 내용



ROM

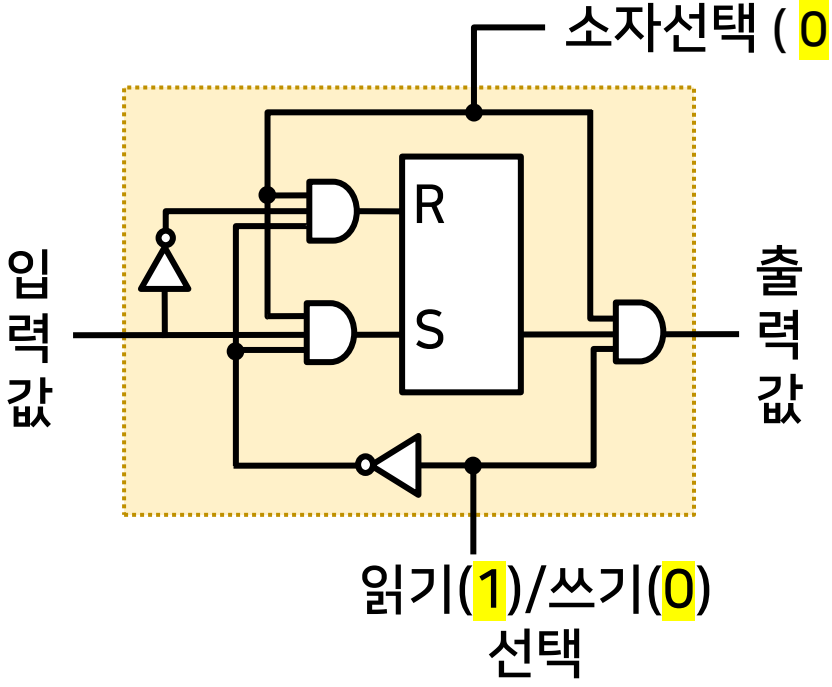
주소 → 2비트, 내용 → 5비트

주소		출력				
A_0	A_1	O_0	O_1	O_2	O_3	O_4
0	0	0	1	0	0	0
0	1	1	1	1	0	1
1	0	0	0	0	1	1
1	1	0	1	0	1	0



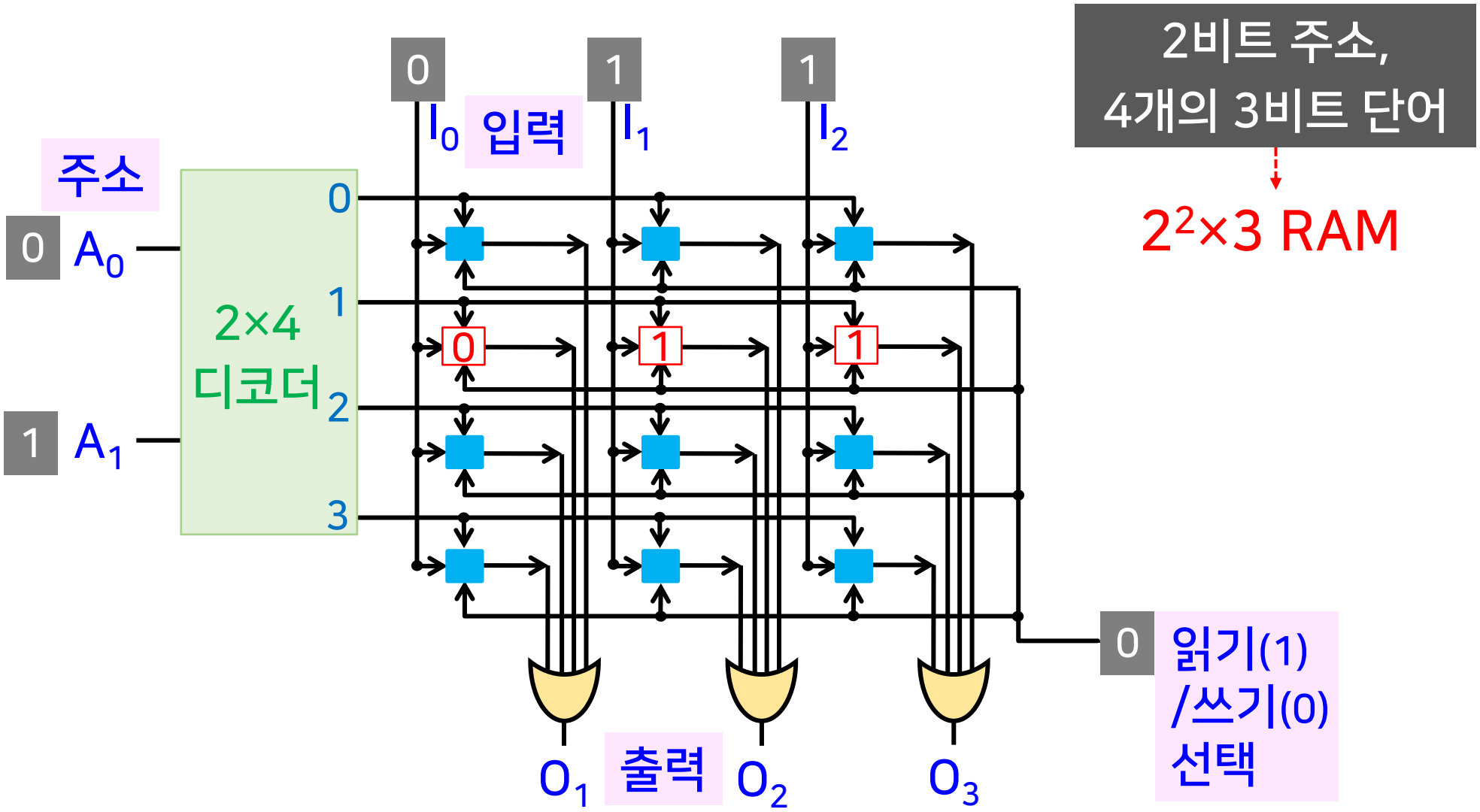
RAM

1비트 기억소자



입력값	소자선택	읽기/쓰기	비고
-	1	1	출력(읽기)
1	1	0	1 저장(쓰기)
0	1	0	0 저장(쓰기)
-	0	-	무변화

RAM


2비트 주소,
4개의 3비트 단어
$2^2 \times 3$ RAM
0
1
1
I_0 입력
I_1
I_2
주소
0 A_0
1 A_1
2×4
디코더
0
1
2
3
0
1
1
0
1
1
O_1
출력
O_2
O_3
0 읽기(1)
/쓰기(0)
선택

기억장치의 계층 구조 memory hierarchy

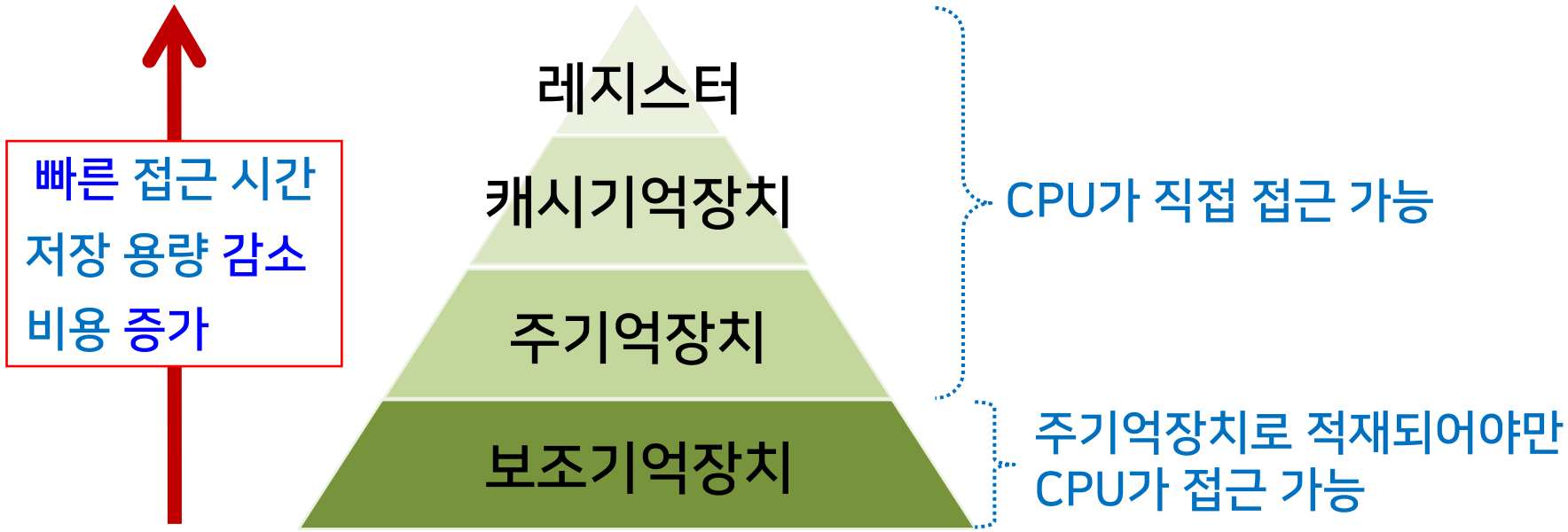
접근 속도와 저장 용량에 따라 계층적으로 분류 가능

- CPU가 데이터에 접근함에 있어서
  가장 적은 비용으로 가장 높은 성능을 얻기 위한 전략

참조의 지역성 locality of reference

- 공간적 지역성
  - 기억장소의 특정 위치에 있는 정보를 참조하고 있을 때
    다음 순간에 접근할 위치는 현재의 접근 위치와 근접해 있을 가능성이 큼
- 시간적 지역성
  - 최근에 접근한 위치들이 가까운 미래에 다시 접근할 가능성이 큼

기억장치의 계층 구조


빠른 접근 시간
저장 용량 감소
비용 증가
레지스터
캐시기억장치
주기억장치
보조기억장치
CPU가 직접 접근 가능
주기억장치로 적재되어야만
CPU가 접근 가능

기억장치의 계층 구조

- **레지스터**
  - CPU 내부에 존재하여 각종 연산에 직접적으로 사용
- **캐시 기억장치**
  - CPU와 주기억장치 사이에 위치한 접근 속도가 빠른 소규모 기억장치
    - CPU와 주기억장치 간의 속도 차이를 줄여 주는 역할

    → 수행 중인 명령어와 자주 사용되는 데이터를 저장하였다가 CPU 요청이 있을 즉시 제공
- **주기억장치**
  - 현재 수행 중인 프로그램 코드와 데이터 저장
- **보조기억장치**
  - 반영구적인 저장
  - 사용하려면 주기억장치에 적재되어야 함

정리하기

1. 불 대수와 논리 게이트

- CPU (제어장치, 산술논리연산장치, 레지스터), 기억장치, 입출력장치, 시스템 버스
- 불 대수, 논리연산, 논리 게이트

2. 논리회로

- 조합회로 vs 순서회로
- 플립플롭 → RS, T
- 조합회로 → 전가산기, 디코더, 인코더, 멀티플렉서, 디멀티플렉서 등
- 순서회로 → 카운터, 레지스터 등

3. 기억장치

- ROM → 조합회로로 구성 가능 (1개의 디코더와 여러 개의 OR 게이트)
- RAM → 순서회로로 구성 가능 (1비트 기억소자 (RS 플립플롭 사용), 디코더, OR 게이트)
- 계층 구조 → 레지스터, 캐시기억장치, 주기억장치, 보조기억장치

10 강

다음시간 안내

컴퓨터 구조 (2)