

Chapter 3

디지털 논리회로

본 자료는 저자의 파워포인트 파일을 기반으로 작성하였습니다

학습목표

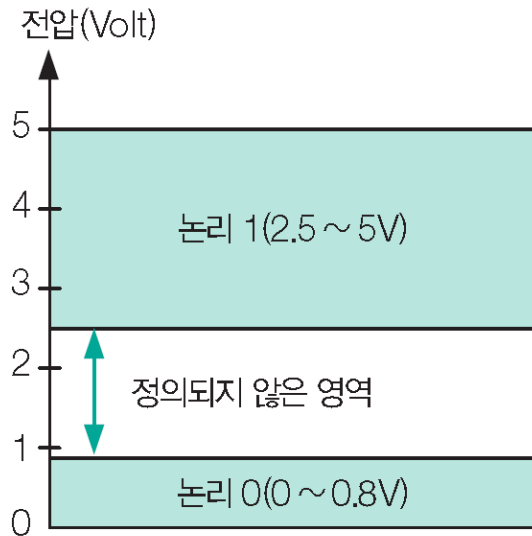
- 논리 게이트의 동작 원리와 진리표, 게이트 기호를 이해하고 이를 활용할 수 있다.
- 곱의 합과 최소항, 합의 곱과 최대항의 개념을 이해하고 이를 활용할 수 있다.
- 여러 가지 조합 논리 회로의 동작 원리를 이해하고 응용 회로를 설계할 수 있다.
- 여러 가지 순서 논리 회로의 동작 원리를 이해하고 응용 회로를 설계할 수 있다.

내용

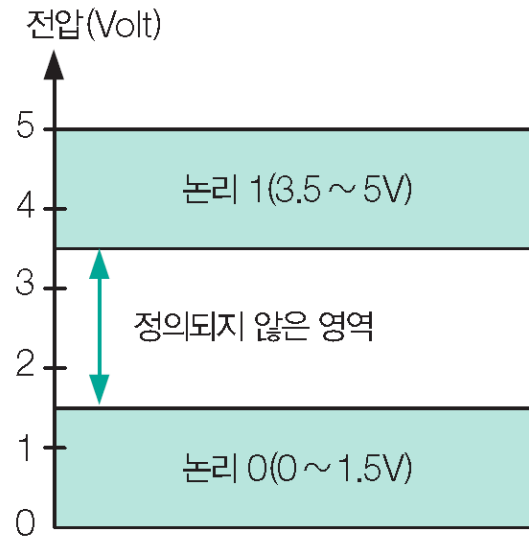
- 01 논리 게이트
- 02 불 대수
- 03 조합 논리 회로
- 04 순서 논리 회로
- 05 집적 회로

1 논리 게이트의 개념과 종류

□ TTL과 CMOS 논리 레벨 정의영역



(a) TTL



(b) CMOS

그림 3-1 TTL과 CMOS 논리 레벨 정의 영역

□ NOT 게이트

- 한 개의 입력과 한 개의 출력을 갖는 게이트로 논리 부정을 나타낸다.
- NOT 게이트를 **인버터**(inverter)라고도 한다.

입력 A	출력 F
0	1
1	0

$$F = \overline{A}$$

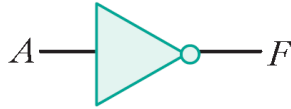


그림 3-2 NOT 게이트의 진리표와 논리 기호

01 논리 게이트

□ 버퍼 게이트

- 버퍼 buffer는 입력 신호를 그대로 출력하여 단순 전송을 표현하는 게이트다.
입력이 0이면 0을 출력하고, 입력이 1이면 1을 출력한다.

입력 A	출력 F
0	0
1	1

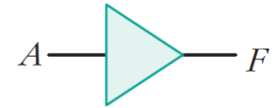
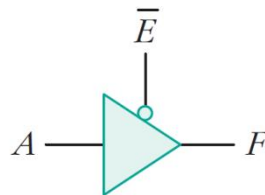
$$F = A$$


그림 3-3 버퍼의 진리표와 논리 기호

❖ 3상태 버퍼(tri-state buffer)

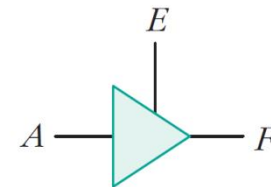
- 출력이 3개 레벨(High, Low, 하이 임피던스) 중의 하나를 갖는 논리소자

입력 A \overline{E}		출력 F
0	0	0
1	0	1
0	1	하이 임피던스
1	1	하이 임피던스



(a) 제어 단자가 Low일 때 동작

입력 A E		출력 F
0	0	하이 임피던스
1	0	하이 임피던스
0	1	0
1	1	1



(b) 제어 단자가 High일 때 동작

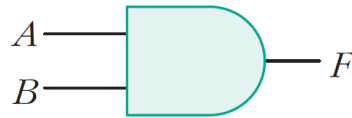
그림 3-4 3상태 버퍼의 진리표와 논리 기호

01 논리 게이트

AND 게이트

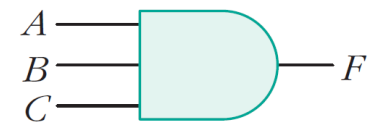
- 입력이 모두 1인 경우에만 출력이 1이고, 입력 중 0이 하나라도 있으면 출력은 0이 되는 **논리곱**이다.

입력		출력
<i>A</i>	<i>B</i>	<i>F</i>
0	0	0
0	1	0
1	0	0
1	1	1



(a) 2입력인 경우: $F = AB$

입력			출력
<i>A</i>	<i>B</i>	<i>C</i>	<i>F</i>
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1



(b) 3입력인 경우: $F = ABC$

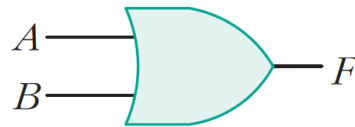
그림 3-5 AND 게이트의 진리표와 논리 기호

01 논리 게이트

□ OR 게이트

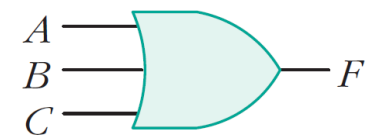
- 입력이 모두 0인 경우에만 출력이 0이고, 입력 중 1이 하나라도 있으면 출력은 1이 되는 **논리합**이다.

입력		출력
A	B	F
0	0	0
0	1	1
1	0	1
1	1	1



(a) 2입력인 경우: $F = A + B$

입력			출력
A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1



(b) 3입력인 경우: $F = A + B + C$

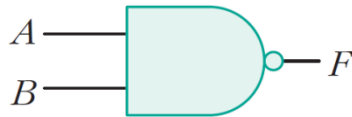
그림 3-6 OR 게이트의 진리표와 논리 기호

01 논리 게이트

□ NAND 게이트

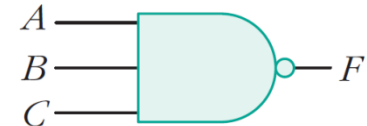
- 입력이 모두 1인 경우에만 출력이 0이고, 입력에 0이 하나라도 있는 경우는 모두 출력이 1이다.
- AND 게이트와 반대로 동작하여 NOT-AND의 의미로 NAND 게이트라고 한다.

입력		출력
A	B	F
0	0	1
0	1	1
1	0	1
1	1	0



(a) 2입력인 경우: $F = \overline{AB}$

입력			출력
A	B	C	F
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0



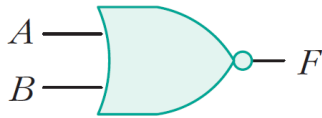
(b) 3입력인 경우: $F = \overline{ABC}$

그림 3-7 NAND 게이트의 진리표와 논리 기호

□ NOR 게이트

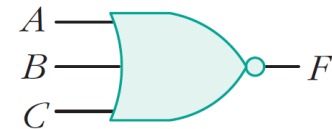
- 입력이 모두 0인 경우에만 출력이 1이고, 입력에 1이 하나라도 있는 경우는 모두 출력이 0이다.
- OR 게이트와 반대로 동작하여 NOT-OR의 의미로 NOR 게이트라고 한다.

입력		출력
A	B	F
0	0	1
0	1	0
1	0	0
1	1	0



(a) 2입력인 경우: $F = \overline{A+B}$

입력			출력
A	B	C	F
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0



(b) 3입력인 경우: $F = \overline{A+B+C}$

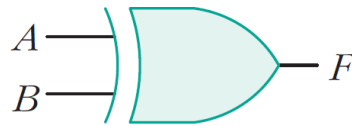
그림 3-8 NOR 게이트의 진리표와 논리 기호

01 논리 게이트

□ XOR 게이트 (eXclusive OR gate)

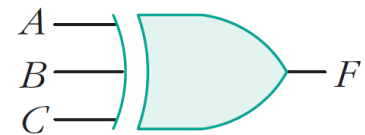
- 입력에 1이 홀수 개이면 출력이 1이고, 짝수 개이면 출력이 0이다.
- 2입력 XOR 게이트는 두 입력 중 하나가 1이면 출력이 1이고, 두 입력 모두 0이거나 1인 경우에만 출력이 0이 되는 것이다.

입력		출력
A	B	F
0	0	0
0	1	1
1	0	1
1	1	0



(a) 2입력인 경우: $F = \bar{A}B + A\bar{B} = A \oplus B$

입력			출력
A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1



(b) 3입력인 경우: $F = A \oplus B \oplus C$

그림 3-9 XOR 게이트의 진리표와 논리 기호

01 논리 게이트

□ XNOR 게이트 (eXclusive NOR gate)

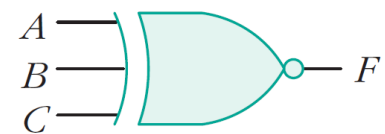
- 입력에 1이 짝수 개이면 출력이 1이고, 홀수 개이면 출력이 0이다.
- XOR 게이트에 NOT 게이트를 연결한 것과 출력이 같으므로 XOR 게이트와 반대다.
- 2입력 XNOR 게이트는 두 입력이 다르면 출력이 0이고, 두 입력이 같으면 출력이 1이라고 생각하면 쉽다.

입력		출력
A	B	F
0	0	1
0	1	0
1	0	0
1	1	1



(a) 2입력인 경우: $F = \overline{A \oplus B} = A \odot B$

입력			출력
A	B	C	F
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0



(b) 3입력인 경우: $F = \overline{A \oplus B \oplus C} = A \odot B \odot C$

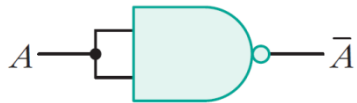
그림 3-10 XNOR 게이트의 진리표와 논리 기호

2 유니버설 게이트

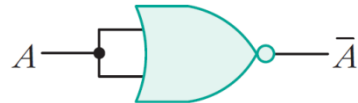
- NAND와 NOR 게이트만으로도 모든 회로를 만들 수 있으므로 이 둘을 특별히 **유니버설 게이트** (universal gate) 또는 **범용 게이트**라고 한다.

□ NOT 게이트 구성

- $A = 0$ 이면 입력 2개에 모두 0이 입력되므로 출력 $F = 1$ 이 된다.
- $A = 1$ 이면 입력 2개에 모두 1이 입력되므로 출력 $F = 0$ 이 된다.



(a) NAND 게이트 이용



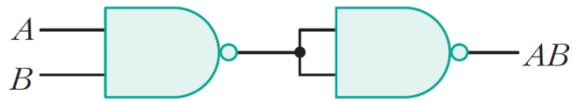
(b) NOR 게이트 이용

그림 3-11 NOT 게이트 구성

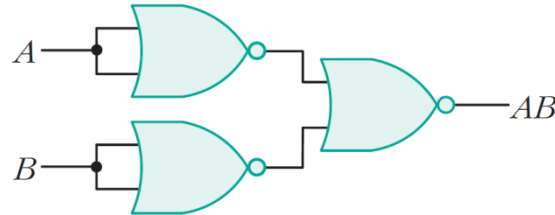
01 논리 게이트

□ AND 게이트 구성

- AND 게이트의 논리식 $F=AB$ 를 이중 부정하고 드모르간의 정리를 적용해 $F=AB=\overline{\overline{A}\overline{B}}=\overline{\overline{A}+\overline{B}}$ 로 변형하여 논리 회로를 구성한다.



(a) NAND 게이트 이용

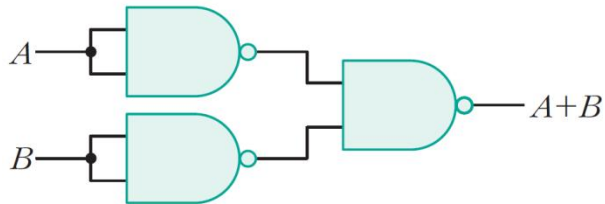


(b) NOR 게이트 이용

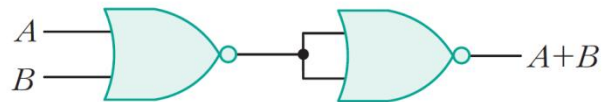
그림 3-12 AND 게이트 구성

□ OR 게이트 구성

- OR 게이트의 논리식 $F=A+B$ 를 이중 부정하고 드모르간의 정리를 적용해 $F=A+B=\overline{\overline{A}+\overline{B}}=\overline{\overline{A}\overline{B}}$ 로 변형하여 논리 회로를 구성한다.



(a) NAND 게이트 이용



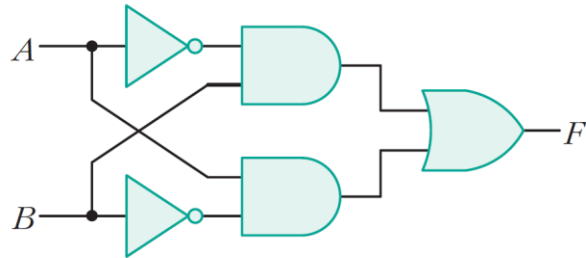
(b) NOR 게이트 이용

그림 3-13 OR 게이트 구성

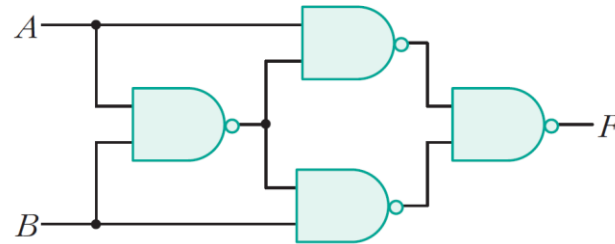
□ XOR 게이트 구성

- 2입력 XOR 게이트의 논리식은 $F = \overline{A}B + A\overline{B}$ 로, 두 입력이 모두 0이거나 1이면 출력이 0이 된다. 이 논리식을 불 대수 법칙으로 다음과 같이 정리하여 논리 회로를 구성한다.

$$\begin{aligned} F &= \overline{\overline{A}B + \overline{A}\overline{B}} = (A + B)\overline{AB} = A \cdot \overline{AB} + B \cdot \overline{AB} \\ &= \overline{\overline{A \cdot \overline{AB} + B \cdot \overline{AB}}} = \overline{\overline{A \cdot \overline{AB}} \cdot \overline{B \cdot \overline{AB}}} \end{aligned}$$



(a) AND-OR-NOT 게이트 이용: $\overline{A}B + A\overline{B}$

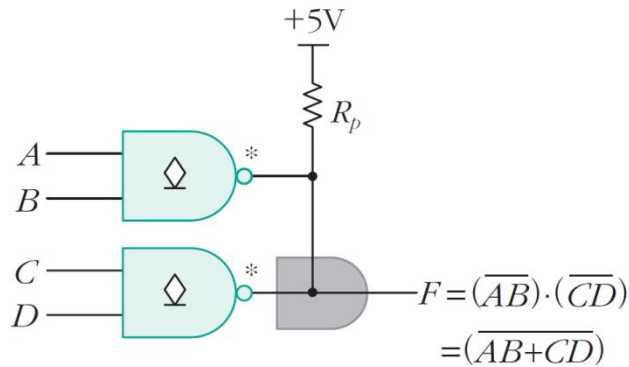


(b) NAND 게이트 이용

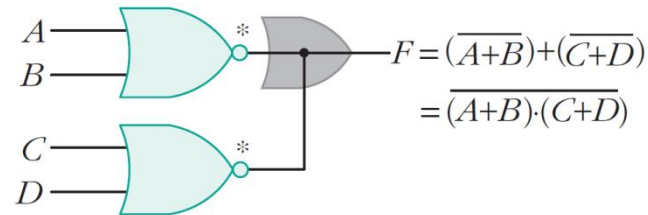
그림 3-14 XOR 게이트 구성

3 와이어드 로직

- 집적 회로(IC)에서는 대부분 NAND나 NOR 게이트 형태로 구성한다.
- 둘 이상의 NAND 또는 NOR 게이트의 출력을 연결해 특정 논리 함수를 수행하는 형태를 **와이어드 로직(wired logic)**이라고 한다.



(a) TTL의 wired-AND 구성



(b) ECL의 wired-OR 구성

그림 3-15 와이어드 로직 예

수고하셨습니다!