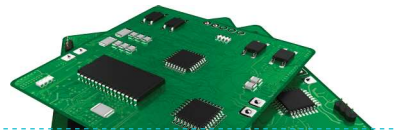
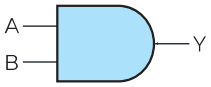
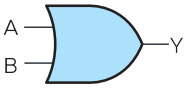
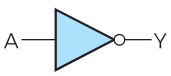
3 논리 게이트



성 | 취 | 기 | 준

1. 논리 회로에서의 AND, OR, NOT 게이트의 동작을 설명할 수 있다.
2. 논리 회로에서의 NAND, NOR, XOR, XNOR 동작을 설명할 수 있다.
3. 버퍼 기능에 대해 설명할 수 있다.

생각 열기

논리 게이트	IC 외형과 논리 기호
AND (IC SN7408)	 
OR (IC SN7432)	 
NOT (IC SN7404)	 

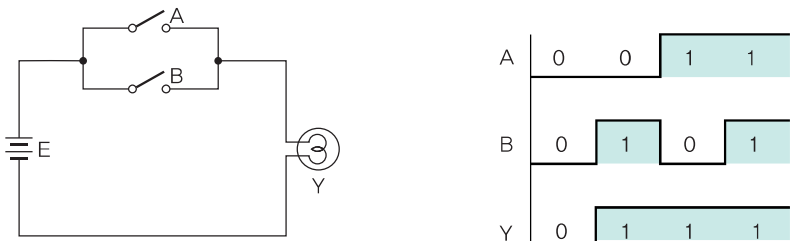
1 기본 논리 게이트

1) AND 연산

AND 연산은 2개 이상의 논리 변수를 논리적으로 곱하는 연산으로 입력 논리 변수의 값이 동시에 모두 1이면 그 출력 결과는 1이고, 이 외의 출력 결과는 0임을 의미한다. 연산 기호는 '·' 또는 '×'를 사용하며 생략할 수 있다. 논리 변수 A, B를 AND 연산하여 나타낸 논리식 $Y = A \times B$ 또는 $Y = A \cdot B$로 표현한다.

$$Y = AB$$

AND 연산을 전기적인 스위치 회로로 구성하면 <그림 II-6>과 같다.



〈그림 Ⅱ-6〉 AND 스위치 회로(왼쪽)와 입출력 파형(오른쪽)

AND 회로는 스위치가 직렬로 연결되어 있기 때문에 스위치 A, B가 모두 닫힐 때만 전류가 흐른다. 스위치 A, B 중에서 어느 하나가 열리거나 A와 B 모두 열릴 때 전류가 흐르지 않는 AND 연산을 하게 된다. 실제 널리 사용하는 AND 게이트의 IC는 TTL 계열의 7408(2 입력), 7411(3 입력)과 CMOS 계열의 4081(2 입력), 4073(3 입력) 등이 있다. AND 게이트의 논리 기호와 논리식은 <표 II-1>과 같다.

〈표 Ⅱ-1〉 AND 게이트의 논리 기호와 논리식

논리 기호	논리식
A, B → Y	$Y = AB$ $= A \cdot B$ $= A \times B$

또한, 진리표는 논리식에 사용된 변수가 가질 수 있는 모든 경우의 값에 대한 결과를 보여 주는 것으로서, 불 대수식을 증명하거나 논리 회로의 입출력 관계를 나타내는 데 주로 사용된다. AND 연산 결과와 진리표는 <표 II-2>와 같다.

〈표 Ⅱ-2〉 AND 연산 결과와 진리표

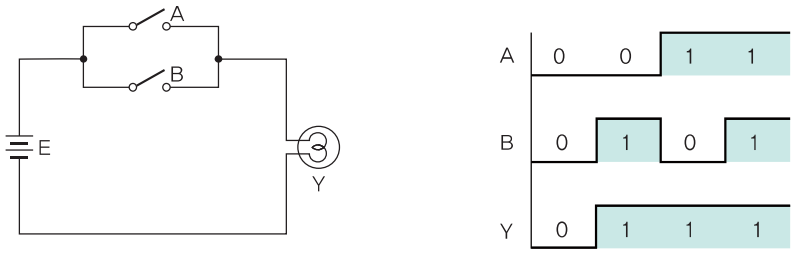
입력 변수		진리표	연산 결과
A	B	$Y = AB$	$A \cdot B = Y$
0	0	0	$0 \cdot 0 = 0$
0	1	0	$0 \cdot 1 = 0$
1	0	0	$1 \cdot 0 = 0$
1	1	1	$1 \cdot 1 = 1$

2) OR 연산

OR 연산은 2개 이상의 논리 변수를 논리적으로 합하는 연산으로서, 입력 논리 변수의 값 중에서 어느 것이라도 1이면 그 출력 결과는 1이 되는 것을 의미한다. 연산 기호는 + 를 사용하고, 논리 변수 A, B를 OR 연산하여 그 결과를 Y로 나타내는 논리식은 다음과 같다. 이때, 논리식에서 '+' 기호는 대수적인 덧셈이 아니라는 점을 주의해야 한다.

$$Y = A + B$$

OR 연산을 전기적인 스위치 회로로 구성하면 <그림 II-7>과 같다.



〈그림 Ⅱ-7〉 OR 스위치 회로(왼쪽)와 입출력 파형(오른쪽)

〈표 Ⅱ-3〉 OR 게이트의 논리 기호와 논리식

논리 기호	논리식
	$Y = A + B$

〈표 Ⅱ-4〉 OR 연산 결과와 진리표

입력 변수		진리표	연산 결과
A	B	$Y = A + B$	$A + B = Y$
0	0	0	$0 + 0 = 0$
0	1	1	$0 + 1 = 1$
1	0	1	$1 + 0 = 1$
1	1	1	$1 + 1 = 1$

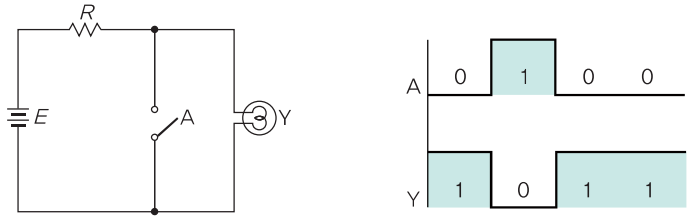
OR 회로는 스위치가 병렬로 연결되어 있기 때문에 스위치 A, B 모두 열려 있을 때에는 전류가 흐르지 않는다. 스위치 A, B 중에서 어느 하나가 닫히거나 A와 B 모두 닫힐 때에만 전류가 흐르게 되어 OR 연산을 하게 된다. 실제 널리 사용하는 OR 게이트의 IC는 TTL 계열의 7432(2 입력)와 CMOS 계열의 4071(2 입력), 4075(3 입력) 등이 있다. OR 게이트의 논리 기호와 논리식은 <표 II-3>과 같다.

OR 연산 결과와 진리표는 <표 II-4>와 같다.

3) NOT 연산

NOT 연산은 하나의 논리 변수에 대하여 부정하는 연산으로 입력 논리 변수의 값이 1이면 그 출력 결과는 0이고, 입력 논리 변수의 값이 0이면 그 출력 결과는 1이다. 논리식은 $Y = \overline{A}$와 같이 쓰며, 'Y는 A의 부정이다'는 의미이다. 논리 변수 A를 부정한 결과를 Y로 나타낸 논리식은 <그림 II-8>과 같다.

$$Y = \overline{A}$$

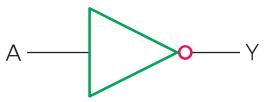


〈그림 Ⅱ-8〉 NOT 스위치 회로(왼쪽)와 입출력 파형(오른쪽)

NOT 연산은 입력 변수 A가 0이면 출력 Y는 1이고, 입력 변수 A가 1이면 출력 Y는 0이 된다. NOT 게이트는 일반적으로 인버터(inverter)라고도 부르며, 입력에 대한 보수(complement)를 얻을 수 있다. 실제 널리 사용하는 NOT 게이트의 IC는 TTL 계열의 7404, 7405, 7406과 CMOS 계열의 4049, 4069 등이 있다. NOT 게이트의 논리 기호와 논리식은 <표 II-5>와 같다.

NOT 연산 결과와 진리표는 <표 II-6>과 같다.

〈표 Ⅱ-5〉 NOT 게이트의 논리 기호와 논리식

논리 기호	논리식
	$Y = \overline{A}$

〈표 Ⅱ-6〉 NOT 연산 결과와 진리표

A	$Y = \overline{A}$
0	1
1	0

2 기타 논리 게이트

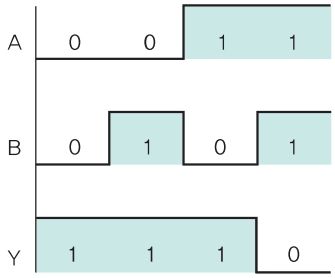
1) NAND 연산

NAND 연산은 AND 연산 결과에 NOT 연산을 결합한 것으로 AND 연산에 대한 부정 또는 보수를 나타낸다. 따라서 입력값 중 어느 것 하나라도 0이면 출력값이 1인 연산을 하고, 모든 입력값이 1일 때에만 0을 출력한다. NAND 게이트는 실제 가장 많이 사용되는 논리 게이트 중 하나이며, AND, OR, NOT 게이트 등의 여러 조합으로 구성할 수 있다. 논리 변수 A, B를 NAND 연산하여 나타낸 논리식 $Y = \overline{AB} = \overline{A \cdot B} = \overline{A \times B}$ 로 표현한다.

$$Y = \overline{AB}$$

NAND 연산 진리표와 입출력 파형은 <그림 II-9>와 같다.

입력		출력
A	B	$Y = \overline{AB}$
0	0	1
0	1	1
1	0	1
1	1	0



〈그림 Ⅱ-9〉 NAND 연산 진리표(왼쪽)와 입출력 파형(오른쪽)

실제 널리 사용하는 NAND 게이트의 IC는 TTL 계열의 7400(2 입력), 7410(3 입력), 7420(4 입력) 등이 있으며, CMOS 계열의 4011(2 입력), 4023(3 입력), 4012(4 입력) 등이 있다. NAND 게이트의 논리 기호와 논리식은 <표 II-7>과 같다.

〈표 Ⅱ-7〉 NAND 게이트의 논리 기호와 논리식

논리 게이트 구성	논리 기호	논리식
		$Y = \overline{AB}$ $= \overline{A \cdot B}$ $= \overline{A \times B}$

2) NOR 연산

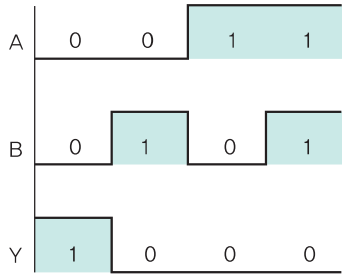
NOR 연산은 OR 연산 결과에 NOT 연산을 결합한 것이다. 이것은 입력값 중 어느 하나라도 1이면 출력값이 0인 연산을 하고, 모든 입력값이 0일 때에만 출력값이 1이다. 즉, OR 연산의 부정 또는 보수임을 알 수 있다. 논리 변수 A, B를 NOR 연산하여 나타낸 논리식 $Y = \overline{A + B}$ 로 표현한다.

$$Y = \overline{A + B}$$

NOR 연산 진리표와 입출력 파형은 <그림 II-10>과 같다.

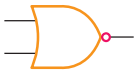
NOR 게이트는 NAND 게이트처럼 많이 이용되는 논리 소자이며 AND, OR, NOT 게이트로 조합이 가능하다. 실제로 널리 사용하는 NOR 게이트의 IC는 TTL에서는 7402(2 입력), 7427(3 입력)과 CMOS에서는 4001(2 입력), 4025(3 입력) 등이 있다. NOR 게이트의 논리 기호와 논리식은 <표 II-8>과 같다.

입력		출력
A	B	$Y = \overline{A + B}$
0	0	1
0	1	0
1	0	0
1	1	0



〈그림 Ⅱ-10〉 NOR 연산 진리표(왼쪽)와 입출력 파형(오른쪽)

〈표 Ⅱ-8〉 NOR 게이트의 논리 기호와 논리식

논리 게이트 구성	논리 기호	논리식
		$Y = \overline{A + B}$

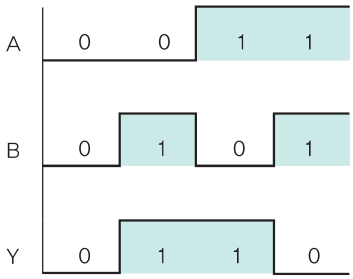
3 XOR 연산

XOR 연산은 배타적-OR, exclusive-OR, EXOR 등으로도 불리며, 두 입력 변수의 값이 같을 때에는 출력이 0이 되고, 입력 변수의 값이 서로 다를 때에는 출력값이 1이 되는 연산이다. 이를 '반일치 회로'라고 하며 주로 보수 회로에 사용된다. 논리 변수 A, B를 XOR 연산하여 나타낸 논리식 $Y = A \oplus B = A\overline{B} + \overline{A}B$ 로 표현한다.

$$Y = A \oplus B$$

XOR 연산 진리표와 입출력 파형은 <그림 II-11>과 같다.

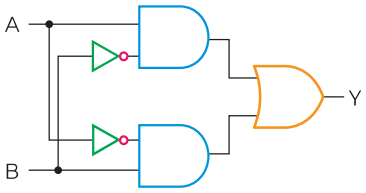
입력		출력
A	B	$Y = A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0



〈그림 Ⅱ-11〉 XOR 연산 진리표(왼쪽)와 입출력 파형(오른쪽)

실제 널리 사용하는 XOR 게이트의 IC는 TTL 계열의 7400(2 입력), 7410(3 입력), 7420(4 입력)과 CMOS 계열의 4030(2 입력) 등이 있다. XOR 게이트의 논리 기호와 논리식은 <표 II-9>와 같다.

〈표 Ⅱ-9〉 XOR 게이트의 논리 기호와 논리식

논리 게이트 구성	논리 기호	논리식
		$Y = \overline{A}B + A\overline{B}$ $= A \oplus B$

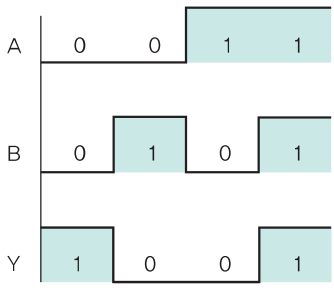
4 XNOR 연산

XNOR 연산은 배타적-NOR, exclusive-NOR, EXNOR 등으로도 불리며, XOR 연산을 부정한 것이다. 따라서 두 입력값이 같을 때는 출력값이 1이 되지만 다르면 0이 된다. 이를 일치 회로라고 하며 주로 비교 회로에 사용된다. 논리 변수 A, B를 XNOR 연산하여 나타낸 논리식 $Y = \overline{A \oplus B} = A \odot B = \overline{AB} + AB$로 표현한다.

$$Y = A \odot B = \overline{A \oplus B}$$

XNOR 연산 진리표와 입출력 파형은 <그림 II-12>와 같다.

입력		출력
A	B	$Y = \overline{A \oplus B}$
0	0	1
0	1	0
1	0	0
1	1	1

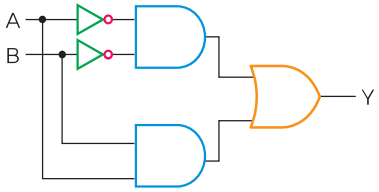


〈그림 Ⅱ-12〉 XNOR 연산 진리표(왼쪽)와 입출력 파형(오른쪽)

실제 널리 사용하는 XNOR 게이트의 IC는 TTL 계열의 74266(2 입력)이 있으며, XNOR 게이트

의 논리 기호와 논리식은 <표 II-10>과 같다.

〈표 Ⅱ-10〉 XNOR 게이트의 논리 기호와 논리식

논리 게이트 구성	논리 기호	논리식
		$Y = A \odot B$ $= \overline{A \oplus B}$ $= \overline{AB} + AB$

5 버퍼 연산

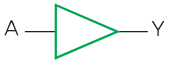
버퍼(buffer) 연산은 입력값이 출력값으로 그대로 나타나는 것으로서 논리적으로는 무의미해 보이지만 실제 회로에서는 다음과 같은 중요한 기능을 가진다. 첫째, 감쇄 신호의 회복 기능이다. 도선이나 여러 게이트를 통과하여 약해진 신호가 버퍼의 출력으로 나오면 감쇄된 신호가 회복된다. 즉, 5[V]의 논리값 1의 상태가 긴 도선의 고유 저항 등에 따른 전압 강하로 인해 논리값 0의 상태가 될 수도 있는데, 이때 도선의 중간에 버퍼 회로를 사용하면 원래의 논리값인 1에 해당하는 5[V]가 정격대로 출력된다. 따라서 버퍼 회로는 입력 신호를 그대로 통과시키는 것이 아니라 입력 신호나 데이터를 감지하여 정격 출력 신호나 데이터를 내보낸다. 둘째, 지연 시간(delay time) 기능이다. 입력된 신호가 버퍼 회로를 통과하여 출력하면 전달 지연 시간이 있게 된다. 마지막으로 팬 아웃(fan out)의 확대나 CMOS의 출력을 여러 개의 TTL에 접속할 때에도 버퍼를 사용할 수 있다. 논리 변수 A를 버퍼 연산하여 나타낸 논리식 $Y = A$로 표현한다.

〈표 Ⅱ-11〉 버퍼 연산 진리표

입력	출력
A	$Y=A$
0	0
1	1

$$Y = A$$

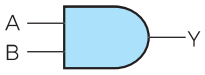
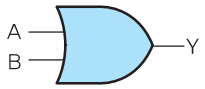
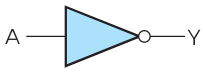
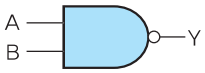
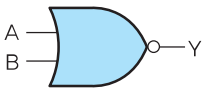
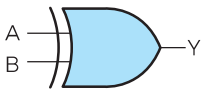
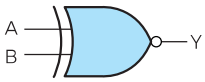
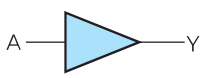
〈표 Ⅱ-12〉 버퍼 게이트의 논리 기호와 논리식

논리 기호	논리식
	$Y = A$

실제 널리 사용하는 버퍼 게이트의 IC는 TTL 계열의 7434와 CMOS 계열의 4050 등이 있다. 버퍼 게이트의 논리 기호와 논리식은 <표 II-12>와 같다.

도움말 코너

지금까지 학습한 논리 게이트를 정리하면 다음과 같다.

게이트	논리 기호	논리식	진리표	명칭
AND		$Y = AB$ $= A \cdot B$ $= A \times B$	A B Y 0 0 0 0 1 0 1 0 0 1 1 1	TTL : 7408 등 CMOS : 4081 등
OR		$Y = A + B$	A B Y 0 0 0 0 1 1 1 0 1 1 1 1	TTL : 7432 등 CMOS : 4071 등
NOT		$Y = \overline{A}$ $= A'$	A Y 0 1 1 0	TTL : 7404 등 CMOS : 4049 등
NAND		$Y = \overline{AB}$ $= \overline{A \cdot B}$	A B Y 0 0 1 0 1 1 1 0 1 1 1 0	TTL : 7400 등 CMOS : 4011 등
NOR		$Y = \overline{A + B}$	A B Y 0 0 1 0 1 0 1 0 0 1 1 0	TTL : 7402 등 CMOS : 4001 등
XOR		$Y = A \oplus B$ $= \overline{A}B + A\overline{B}$	A B Y 0 0 0 0 1 1 1 0 1 1 1 0	TTL : 7486 등 CMOS : 4030 등
XNOR		$Y = A \odot B$ $= \overline{AB} + AB$ $= \overline{A \oplus B}$	A B Y 0 0 1 0 1 0 1 0 0 1 1 1	TTL : 74266 등
버퍼 (buffer)		$Y = A$	A Y 0 0 1 1	TTL : 7434 등 CMOS : 4050 등