

11

CHAPTER

부울 대수

Boolean Algebra



단원의 주요 목표

부울식의 개념을 이해하고 부울 대수와 관련된 다양한 논제들을 고찰한다.

- 부울식의 기본 연산과 부울식의 법칙에 대해 알아본다.
- 부울식의 표현과 곱의 합 표현 방법을 파악한다.
- 부울식의 기본 법칙과 카노우맵으로 부울 함수를 간소화한다.
- 부울 함수를 논리합 등의 게이트를 사용하여 논리 회로를 설계한다.
- 전자레인지와 전자투표기 등 논리 회로의 응용을 탐구한다.
- 부울 대수의 응용과 4차 산업혁명과의 관계를 살펴본다.

11

CHAPTER

부울 대수

Boolean Algebra



CONTENTS

11.1 부울식

11.2 부울식의 표현

11.3 부울 함수의 간소화

11.4 논리 회로 설계

11.5 논리 회로의 응용

11.6 부울 대수의 응용과 4차 산업혁명과의 관계

■ 요약 및 생활 속의 응용

■ 연습문제

부울 대수(Boolean algebra)

- 1854년 영국의 수학자 **조지 부울**(George Boole, 1815~1864)이 쓴《사고의 법칙 연구(An Investigation of the Laws of Thought)》에서 수학적 논리 형태로 처음 소개됨
- 1938년 새넨(Shannon)이 부울 대수의 기본 개념을 이용하여 회로 함수에 대한 설계로 발전시킴
- 0과 1의 조합으로 연산되는 것을 **부울 대수**라고 함
- 전기 장치나 컴퓨터 회로는 켜짐과 꺼짐의 두 가지 상태로 나타냄
- 스위치나 회로는 닫힘과 열림의 두 가지 상태 중 하나인 참 또는 거짓, 1 또는 0으로 표현될 수 있음

11.1 부울식



- **부울식**은 두 원소를 가지는 집합 $A = \{0, 1\}$ 와 이항 연산자 (binary operator) $+$ (OR)과 \cdot (AND), 그리고 단항 연산자 (unary operator) $'$ (complement)로 표현되는 식을 말함
- 부울식에서 연산자의 **연산 우선순위**는 $' > \cdot > +$ 순임

$$1) + : 0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 1$$

$$2) \cdot : 0 \cdot 0 = 0$$

$$0 \cdot 1 = 0$$

$$1 \cdot 0 = 0$$

$$1 \cdot 1 = 1$$

11.1 부울식



$$3) \quad 1 : 0' = 1$$

$$1' = 0$$

예를 들어, $0 \cdot 1 + 1 \cdot 1 + 0' \cdot 1'$ 이라는 부울식이 주어졌다면, 이 식을 연산 우선순위와 기본 연산에 따라 계산하면

$$\begin{aligned} 0 \cdot 1 + 1 \cdot 1 + 0' \cdot 1' &= 0 \cdot 1 + 1 \cdot 1 + 1 \cdot 0 \\ &= 0 + 1 + 0 \\ &= 1 \end{aligned}$$



여기서 잠깐!!

부울 연산은 논리에서의 AND, OR, NOT에 대응한다. 즉,

$\cdot \rightarrow$ AND

$+$ \rightarrow OR

$' \rightarrow$ NOT

이다.

11.1 부울식



예제 11-1

다음 부울식의 결과값을 구해보자.

$$(1) 0 + 1'$$

$$(2) 0 \cdot 0 + 1' \cdot 0$$

$$(3) (1 + 0') \cdot 1$$

$$(4) (1 \cdot 0 + 0 \cdot 1)' + 0' \cdot 0'$$

풀이 (1) $0 + 1' = 0 + 0 = 0$

$$\begin{aligned} (2) 0 \cdot 0 + 1' \cdot 0 &= 0 \cdot 0 + 0 \cdot 0 \\ &= 0 + 0 \\ &= 0 \end{aligned}$$

$$\begin{aligned} (3) (1 + 0') \cdot 1 &= (1 + 1) \cdot 1 \\ &= 1 \cdot 1 \\ &= 1 \end{aligned}$$

$$\begin{aligned} (4) (1 \cdot 0 + 0 \cdot 1)' + 0' \cdot 0' &= (0 + 0)' + 1 \cdot 1 \\ &= (0)' + 1 \\ &= 1 + 1 \\ &= 1 \end{aligned}$$

11.1 부울식



정의 11-1 부울식은 다음과 같이 정의된다.

- 1) 상수 0과 1 그리고 부울 변수는 부울식이다.
- 2) f_1 과 f_2 가 부울식일 때 f_1' , $f_1 + f_2$, $f_1 \cdot f_2$, (f_1) 역시 부울식이다.

부울 변수가 x, y, z 라 할 때 정의에 따라 다음 식들은 부울식이 된다.

$$x' + y \cdot z + (x + y + z)'$$

$$0 + (x \cdot y)' + (x' + z \cdot y)$$

- 부울식에서 부울 변수 x, y, z 가 가질 수 있는 값인 0 또는 1을 정해줌으로써 각 부울식에 대한 값을 구할 수 있음
- 두 부울식이 같은 진리표(truth table)를 가질 경우 두 부울식을 **동치(equivalent)**라 함
- 두 함수 f_1 과 f_2 가 동치인 경우 $f_1 = f_2$ 로 표시함

11.1 부울식



〈표 11.1〉 부울식의 법칙

p, q, r 을 부울 변수라 한다.

1. $p \bullet p = p$

$$p + p = p$$

멱등 법칙(idempotent law)

2. $p + 0 = p$

$$p \bullet 1 = p$$

항등 법칙(identity law)

3. $p + q = q + p$

$$p \bullet q = q \bullet p$$

교환 법칙(commutative law)

4. $p + (q + r) = (p + q) + r$

$$p \cdot (q \cdot r) = (p \cdot q) \cdot r$$

결합 법칙(associative law)

11.1 부울식



$$\begin{aligned} 5. \quad & p + (q \cdot r) = (p + q) \cdot (p + r) \\ & p \cdot (q + r) = (p \cdot q) + (p \cdot r) \end{aligned}$$

분배 법칙(distributive law)

$$\begin{aligned} 6. \quad & p + (p \cdot q) = p \\ & p \cdot (p + q) = p \end{aligned}$$

흡수 법칙(absorption law)

$$\begin{aligned} 7. \quad & p + p' = 1 \\ & p \cdot p' = 0 \end{aligned}$$

역 법칙(inverse law)

$$8. \quad (p')' = p$$

보 법칙(complement law)

$$\begin{aligned} 9. \quad & p + 1 = 1 \\ & p \cdot 0 = 0 \end{aligned}$$

우등 법칙(dominance law)

$$\begin{aligned} 10. \quad & (p + q)' = p' \cdot q' \\ & (p \cdot q)' = p' + q' \end{aligned}$$

드 모르간의 법칙(De Morgan's law)

11.1 부울식



예제 11-2

p, q, r 이 부울 변수일 때 부울식 $(p + q')'r' = p'qr'$ 이 성립함을 살펴보자.

$$\begin{aligned} \text{풀이 } (p + q')'r' &= p'(q')'r' \quad \text{드 모르간의 법칙} \\ &= p'qr' \quad \text{보 법칙} \end{aligned}$$



예제 11-3

다음 부울식을 <표 11.1>의 법칙들을 이용하여 간단히 해보자.

(1) $x + xy$

(2) $wy + xy + wz + xz$

$$\begin{aligned} \text{풀이 } (1) \quad x + xy &= x1 + xy \quad \text{항등 법칙} \\ &= x(1 + y) \quad \text{분배 법칙} \\ &= x1 \quad \text{우등 법칙} \\ &= x \quad \text{항등 법칙} \end{aligned}$$

$$\begin{aligned} (2) \quad wy + xy + wz + xz &= (w + x)y + (w + x)z \quad \text{분배 법칙} \\ &= (w + x)(y + z) \quad \text{분배 법칙} \end{aligned}$$

11.1 부울식



예제 11-4

다음 부울식이 성립함을 진리표를 이용하여 살펴보자.

$$x + yz = (x + y)(x + z)$$

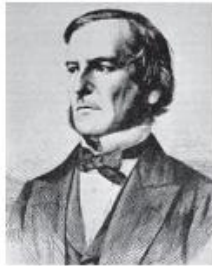
풀이

x	y	z	yz	$x + y$	$x + z$	$x + yz$	$(x + y)(x + z)$
0	0	0	0	0	0	0	0
0	0	1	0	0	1	0	0
0	1	0	0	1	0	0	0
0	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1
1	0	1	0	1	1	1	1
1	1	0	0	1	1	1	1
1	1	1	1	1	1	1	1

11.1 부울식



부울대수와 부울식의 개척자- 조지 부울



부울(George Boole, 1815~1864)

부울은 영국의 수학자이자 논리학자로서 부울 대수를 창안하여 기호논리학 분야에 큰 업적을 남겼다. 불변식론을 연구하여 중요한 성과를 얻었으며 영국학파의 중요한 인물이 되었다. 그의 뛰어난 업적은 기호논리학의 창시와 부울 대수(Boolean algebra)에 관한 연구이다. 1854년에 논리와 추론을 수학적으로 다룬 《논리와 확률의 수학적 기초를 이루는 사고(思考)의 법칙 연구》로 기호논리학의 바탕을 이루었다.

11.2 부울식의 표현



부울 변수들에 대한 함수를 **부울 함수(Boolean function)**라 하며,
 n 개의 부울 변수 x_1, x_2, \dots, x_n 에 대한 부울 함수는 $f(x_1, x_2, \dots, x_n)$ 으
 로 표시함

부울 함수로 표시하면

$$f(x, y) = x + xy$$

$$f(x, y, w, z) = wy + xy + wz + xz$$

11.2 부울식의 표현



부울 함수(Boolean function)

- 부울 변수와 부울 연산자로 구성된 부울식으로 표현할 수 있음
- n 개의 부울 변수가 있을 때 그 변수들로부터 얻을 수 있는 조합은 $00\dots0, 00\dots1, \dots, 11\dots1$ 로 2^n 개임
- 0은 그에 해당되는 원소가 없는 경우임
- 1은 있는 경우로 생각함



정의 11-2 n 개의 부울 변수로 만들어지는 진리표에서 변수의 각 항을 **최소항(Minterm)**이라고 한다.



여기서 잠깐!!

n 개의 변수가 있으면 2^n 개의 최소항이 있게 된다. 각 최소항들은 n 개의 부울 변수들의 곱으로 나타낸다.

11.2 부울식의 표현



예제 11-5

$x=1, y=0, z=1$ 일 때 부울 함수 $f(x, y, z)$ 의 값은 1이 되고, 다른 경우에는 0의 값을 가진다고 하자. 이때 그 부울 함수에 대한 진리표를 구하고, $f(x, y, z)=1$ 인 최소항을 부울 변수의 곱으로 나타내어보자.

풀이 부울 변수가 x, y, z 이므로, $2^3=8$ 개의 최소항이 만들어진다.

x	y	z	$f(x, y, z)$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

위의 진리표에서 $x=1, y=0, z=1$ 일 때 $f(x, y, z)=1$ 이므로

$$f(x, y, z) = xy'z \text{이다.}$$

11.2 부울식의 표현



부울함수의 표현

부울 함수는 부울 변수에 대한 최소항들 중에서 1의 값을 가지는 최소항들의 부울 합을 식으로 표현하는 함수를 말한다.

각각의 최소항은 부울 변수의 곱으로 나타내어지는데, 해당되는 변수가 x 일 때 그 값이 1이면 x , 0이면 x' 로 표시한다. 세 변수 x, y, z 값이 각각 0, 1, 1이라면 그에 해당하는 최소항을 변수들의 곱으로 나타내면 $x'yz$ 가 된다.

이와 같이 표시한다면 예제 11-5의 진리표에 나타난 최소항들은 $x'y'z'$, $x'y'z$, ..., xyz 순으로 나타낼 수 있다. 그중에서 $f(x, y, z)$ 의 값이 1이 되는 최소항들을 합으로 나타내면 부울 함수가 된다.

11.2 부울식의 표현



예제 11-6

부울 변수에 대한 진리표가 다음과 같을 때 부울 함수 $f(x, y, z)$ 를 구해보자.

x	y	z	$f(x, y, z)$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

풀이 부울 함수는 $f(x, y, z) = 1$ 인 최소항들의 합이므로

$$f(x, y, z) = x'y'z + xy'z' + xyz'$$
 이다.

11.2 부울식의 표현



부울식의 곱의 합 표현

- 곱의 합(sum of products) 또는 논리합 표준형(Disjunctive Normal Form : DNF)은 부울 함수를 나타내는 데 있어서 최소항들의 합으로 표현하는 것임
- 최소항들은 부울 변수의 곱으로 표현됨
- 부울 함수는 부울 변수의 곱의 합으로 표현됨



예제 11-7

다음의 부울 함수를 부울 변수의 곱의 합으로 나타내어보자.

$$(1) f(x, y) = x' + y$$

$$(2) f(x, y, z) = (x + y)z'$$

풀이 부울 함수를 부울 변수에 대한 진리표로 나타내고, 그 진리표에서 부울 변수에 대한 곱의 합의 표현으로 부울 함수를 나타내면 된다.

11.2 부울식의 표현



(1) $f(x, y) = x' + y$ 에 대한 진리표를 나타내면 다음과 같다.

x	y	x'	$(x' + y)$
0	0	1	1
0	1	1	1
1	0	0	0
1	1	0	1

위의 진리표에서 $x' + y$ 의 값이 1인 최소항들을 곱의 합으로 나타내면

$$f(x, y) = x'y' + x'y + xy$$

가 된다.

11.2 부울식의 표현



(2) $f(x, y, z) = (x + y)z'$ 의 진리표는 다음과 같다.

x	y	z	$x + y$	z'	$(x + y)z'$
0	0	0	0	1	0
0	0	1	0	0	0
0	1	0	1	1	1
0	1	1	1	0	0
1	0	0	1	1	1
1	0	1	1	0	0
1	1	0	1	1	1
1	1	1	1	0	0

위의 진리표에서 $f(x, y, z)$ 의 곱의 합을 구하면

$$f(x, y, z) = x'yz' + xy'z' + xyz'$$

가 된다.

11.3 부울 함수의 간소화



- 부울 변수의 최소항들로부터 얻어진 부울 함수는 좀 더 간단한 형태의 식으로 만들 수 있음
- 간소한 형태의 식이란 같은 기능이나 결과를 도출하면서도 더 적은 변수와 연산자를 사용한 식을 말함

부울 함수식을 간소하게 표현하는 방법

- (1) 부울식의 기본 법칙 사용
- (2) 카노우맵을 사용

- 부울식의 법칙을 이용하여 부울 함수를 간소하게 하는 방법은 부울식의 기본 법칙을 이용 하는 것으로, 그 과정이 복잡하고 간소화에 대한 확인이 쉽지 않음

11.3 부울 함수의 간소화



예제 11-8

다음 부울 함수를 부울식의 기본 법칙을 이용하여 간단히 해보자.

$$f(x, y, z) = x'y'z' + x'yz' + xy'z' + xyz'$$

$$\begin{aligned} \text{풀이 } f(x, y, z) &= x'y'z' + x'yz' + xy'z' + xyz' \\ &= x'z'(y' + y) + xz'(y' + y) && \text{분배 법칙} \\ &= x'z' \cdot 1 + xz' \cdot 1 && \text{역 법칙} \\ &= x'z' + xz' && \text{항등 법칙} \\ &= (x' + x)z' && \text{분배 법칙} \\ &= 1 \cdot z' && \text{역 법칙} \\ &= z' && \text{항등 법칙} \end{aligned}$$



정의 11-3

카노우맵(Karnaugh map)은 부울 변수들에 대한 최소항들을 도표로 그린 후 인접한 최소항들을 서로 묶어서 표현함으로써 부울 함수를 간소화하는 방법이다.

11.3 부울 함수의 간소화



여기서 잠깐!!

카노우맵은 1953년 카노우(Maurice Karnaugh)에 의해 제안되었는데, 사각형 모양의 표를 이용하여 부울 함수를 비교적 쉽게 간소화할 수 있는 방법이다. 그러나 통상 6개 이하의 변수를 다루는 데에만 사용할 수 있다.

- **카노우맵**을 그릴 때는 부울 변수들로부터 나타내어지는 모든 경우의 최소항들을 사각형으로 연결시켜서, 최소항들 중 1의 값을 가지는 사각형 안에 '1'을 표시함
- '1'로 표시된 사각형들에서 **부울식의 공통점을 찾아내어 부울 함수를 간소화**하는데, 사각형을 연결시킬 때는 인접한 사각형끼리는 한 변수의 변화만 있게 만들어야 함

예를 들어, x 와 y 가 부울 변수일 때 xy 와 $x'y'$ 는 두 변수 모두 다른 값을 가지기 때문에 서로 인접하지 않음

11.3 부울 함수의 간소화



여기서 잠깐!!

부울 함수가 논리 회로에 대응되므로 부울 함수의 간소화는 논리 회로 설계 시 필요한 게이트의 수를 가능한 줄일 수 있다. 요즘에는 반도체 칩 기술의 발전으로 회로 표현을 간단히 하는 것의 중요성이 다소 줄어들었으나 가능한 간략하게 표현하는 것이 효율적이다.

(1) 두 변수에 대한 카노우맵

두 변수를 x 와 y 라 할 때, 그에 필요한 사각형은 변수들이 가질 수 있는 모든 경우인 $x'y'$, $x'y$, xy' , xy 에 대한 것임

		y		0	1
x	0	$x'y'$	$x'y$		
	1	xy'	xy		

〈그림 11.1〉 두 변수에 대한 카노우맵

11.3 부울 함수의 간소화



- x 가 0이고 y 가 0일 때는 그에 해당되는 부울식은 $x'y'$ 가 됨
- 다른 사각형들도 그 위치에 따라 부울식의 값이 정해짐

예를 들어, $f(x, y) = x'y' + x'y + xy'$ 인 경우 이 함수를 카노우맵으로 나타내면 다음과 같이 표시됨

	y	0	1
x			
0		1	1
1		1	

〈그림 11.2〉 $f(x, y) = x'y' + x'y + xy'$ 의 카노우맵

11.3 부울 함수의 간소화



- 카노우맵의 사각형에 해당되는 논리값이 1이면 '1'로 표시, 0이면 그냥 공백으로 남겨둠
- '1'로 표시된 사각형이 인접할 경우, 그들을 함께 묶어서 표현하는 방법은 다음과 같음

$x \backslash y$	0	1
0	1	
1		1

(1)

$x \backslash y$	0	1
0	1	1
1		

(2)

$x \backslash y$	0	1
0	1	1
1	1	

(3)

$x \backslash y$	0	1
0	1	1
1	1	1

(4)

〈그림 11.3〉 두 변수에 대한 카노우맵의 예

11.3 부울 함수의 간소화



두 변수에 대한 카노우맵의 간소화

- (1)의 경우에는 서로 인접한 1이 없어 묶을 수가 없으므로 각각 따로 표현하면 $x'y' + xy$ 가 됨
- (2) y 의 값이 0이든 1이든 관계없이 x 의 값이 0 일 때, 다시 말해 x' 일 때 1의 값을 가지므로 x' 가 됨
- (3) 윗부분의 묶음은 (2)의 경우와 같고, 아래쪽으로의 묶음은 x 의 값에 관계없이 y' 이므로, 부울식은 $x' + y'$ 가 됨
- (4) x 와 y 의 값에 관계없이 항상 1이므로 부울식은 1이 됨



여기서 잠깐!!

카노우맵을 사용하여 인접한 항들끼리 묶을 때에는 항상 2의 배수로 묶는다. 3개가 인접해 있을 경우에는 가운데 있는 항이 양쪽으로 사용되어 2개씩 따로 묶어서 나타낸다. 카노우맵에서는 한꺼번에 많은 개수의 항을 묶어서 식을 표현할수록 훨씬 간단한 부울식을 구할 수 있다.

11.3 부울 함수의 간소화



예제 11-9

다음 부울 함수에 대한 카노우맵을 그린 후 간소화해보자.

$$(1) f(x, y) = x'y + xy$$

$$(2) f(x, y) = xy' + xy + x'y$$

풀이 (1) 카노우맵에서 x 와는 관계가 없으므로 $f(x, y) = y$ 가 된다

y	0	1
x		
0		1
1		1

(2) $f(x, y) = x + y$ 가 된다.

y	0	1
x		
0		1
1	1	1

11.3 부울 함수의 간소화



(2) 세 변수에 대한 카노우맵

- 변수가 세 개인 경우에는 $2^3 = 8$ 개의 사각형이 필요함
- 세 변수에 대한 카노우맵 그림은 아래와 같음

yz	00	01	11	10
x				
0	$x'y'z'$	$x'y'z$	$x'yz$	$x'yz'$
1	$xy'z'$	$xy'z$	xyz	xyz'

〈그림 11.4〉 세 변수에 대한 카노우맵

11.3 부울 함수의 간소화



세 변수에 대한 카노우맵을 그릴 때 유의해야 할 사항

- yz 에 대한 사각형에서 오른쪽 방향으로 00, 01 다음에 10이 아니라 11이 먼저임
- 두 변수가 한꺼번에 변하는 것은 인접할 수가 없기 때문에 01 옆에는 10이 아닌 11이 옴
- 왼쪽 끝의 사각형들은 오른쪽 끝의 사각형들과 인접되어 있다고 생각해야 함

예) 카노우맵이 아래 그림과 같은 경우에는 서로 연결되어 있다고 생각하므로 4개의 사각형을 1개로 묶어서 표시하면 \bar{x} 가 됨

$x \backslash yz$	00	01	11	10
0	1			1
1	1			1

〈그림 11.5〉 세 변수의 카노우맵

11.3 부울 함수의 간소화



예제 11-10

다음의 진리표를 보고 카노우맵을 사용하여 부울 함수를 간소화해보자.

x	y	z	$f(x, y, z)$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

풀이 진리표로부터 카노우맵을 그린 후 인접한 것들끼리 묶으면 다음과 같다.

$x \backslash yz$	00	01	11	10
0		1	1	1
1		1	1	

4개를 묶어놓은 것은 x 와 y 값에 관계없이 z 값이 1이면 되므로 z 이고, 옆으로 2개를 묶은 것은 z 값에 관계없이 $x'y$ 이다. 그러므로 $f(x, y, z) = z + x'y$ 이다.

11.3 부울 함수의 간소화



예제 11-11

다음 부울식을 카노우맵을 사용하여 간소화해보자.

$$x'y'z' + x'yz' + x'y'z + x'yz$$

풀이

yz	00	01	11	10
x				
0	1	1	1	1
1				

네 개를 하나로 묶을 수 있으므로 부울식의 결과는 x' 이다.

11.3 부울 함수의 간소화



(3) 네 변수에 대한 카노우맵

(옵션: 네 변수 카노우맵과 (예제 11-12, 13)은 난이도가 높아 다른 논리회로 강좌에서 다룰 수도 있음)

- 네 변수에 대한 카노우맵도 세 변수에 대한 카노우맵과 같이 가로와 세로 축에 01 다음에 11이 옴
- 왼쪽 끝 사각형들과 오른쪽 끝 사각형들이 인접하고 있다고 보며, 위쪽 사각형들과 아래쪽 사각형들 역시 인접하고 있다고 봄

zw	00	01	11	10
xy 00	$x'y'z'w'$	$x'y'z'w$	$x'y'zw$	$x'y'zw'$
01	$x'yz'w'$	$x'yz'w$	$x'yzw$	$x'yzw'$
11	$xyz'w'$	$xyz'w$	$xyzw$	$xyzw'$
10	$xy'z'w'$	$xy'z'w$	$xy'zw$	$xy'zw'$

〈그림 11.6〉 네 변수에 대한 카노우맵

11.3 부울 함수의 간소화



예제 11-12

카노우맵을 사용하여 다음 부울 함수를 간소화해보자.

$$f(x, y, z, w) = x'y'z'w' + x'y'zw' + x'yz'w + x'yzw + xyz'w + xyzw + xy'z'w' + xy'zw'$$

풀이

zw \ xy	00	01	11	10
00	1			1
01		1	1	
11		1	1	
10	1			1

가운데 있는 4개는 하나로 묶어서 yw 가 되고, 양쪽 구석의 것들도 서로 인접하고 있다고 생각하므로 하나로 묶을 수 있다. 그러므로 4개를 하나로 묶으면 $y'w'$ 이 된다. 따라서 $f(x, y, z, w) = yw + y'w'$ 이다.

11.3 부울 함수의 간소화



예제 11-13

다음의 진리표에 나타난 부울 함수를 간소화해보자.

x	y	z	w	$f(x, y, z, w)$
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

11.3 부울 함수의 간소화



풀이 먼저 부울 함수에 대한 카노우맵을 그리면

zw \ xy	00	01	11	10
00				
01				
11	1	1	1	
10		1	1	

이 되고, 카노우맵에서 부울 함수를 구하면 $f(x, y, z, w) = xw + xyz'$

11.4 논리 회로 설계



논리 회로

- 부울 함수로 표현된 식은 컴퓨터에서 사용되는 기본적인 논리 회로를 설계하는 데 활용함
- 컴퓨터 회로를 설계하는 데 있어서 입력과 출력은 논리 회로의 **게이트(gate)**들을 상호 연결함으로써 구성할 수 있음
- 입력은 부울 변수로, 출력은 부울 함수로, 부울 연산자는 게이트로 표현함
- 논리값이 1일 때는 회로의 스위치가 연결된(on) 상태이고, 0일 때는 연결되지 않은(off) 상태를 나타냄



여기서 잠깐!!

논리 회로(logic circuit)란 부울 대수의 기본 연산인 논리합, 논리곱, 논리부정 등의 연산을 실행하기 위한 회로로서 **논리 게이트(logic gate)**라고도 한다. 논리 회로는 2진 정보를 취급하며 통상 2개 이상의 입력 단자와 하나의 출력 단자로 구성된다.

11.4 논리 회로 설계



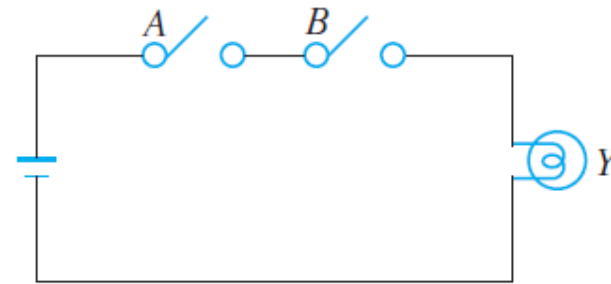
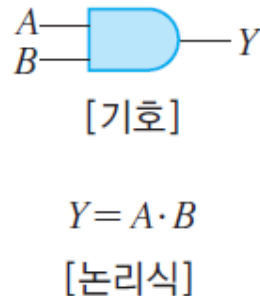
(1) 논리곱 회로(AND gate)

- **논리곱 회로**는 논리곱(AND) 조건을 만족시키는 회로로서 2개의 입력 A와 B가 모두 1인 경우에만 출력 Y가 1이 됨
- 논리곱 연산자는 '.'으로 표현함
- AND 게이트는 스위치가 직렬로 연결되어 있으므로 두 스위치가 모두 1(ON)이 되어야 회로가 연결됨

논리곱 회로

입력		출력
A	B	$A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

[진리표]



〈그림 11.7〉 논리곱 회로의 진리표, 기호 및 회로

11.4 논리 회로 설계



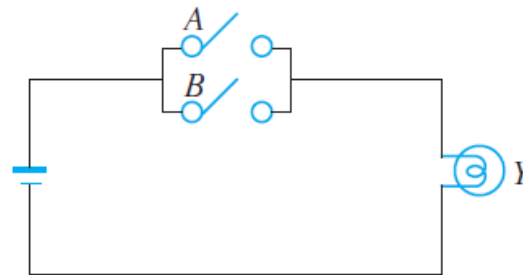
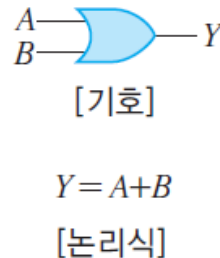
(2) 논리합 회로(OR gate)

- **논리합 회로**는 논리합(OR) 조건을 만족시키는 회로로서 **입력 A와 B 중 적어도 하나가 1인 경우에 출력 Y가 1이 되는 논리 회로임**
- 논리합 연산자는 ' + '로 표현함
- OR 게이트는 스위치가 병렬로 연결되어 있으므로 두 스위치 중 하나만 1이 되면 회로가 연결됨

논리합 회로

입력		출력
A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

[진리표]



〈그림 11.8〉 논리합 회로의 진리표, 기호 및 회로

11.4 논리 회로 설계



(3) 논리부정 회로(NOT gate)

- 논리부정 회로는 논리부정(NOT) 조건을 만족시키는 회로로서 입력 A가 1일 경우에는 출력 Y가 0이 됨
- 입력 A가 0일 경우에는 출력 Y가 1이 됨
- 논리부정 연산자는 ' ' 또는 ' - '로 표현됨

논리부정 회로

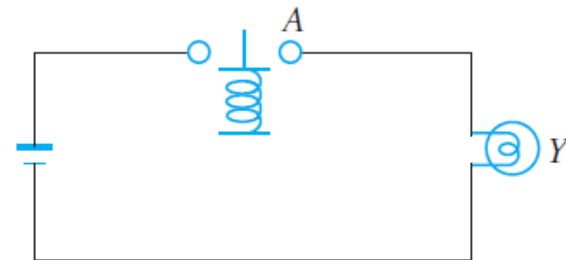
입력	출력
A	A'
0	1
1	0

[진리표]



$$Y = \bar{A} \text{ 또는 } Y = A'$$

[논리식]



〈그림 11.9〉 논리부정 회로의 진리표, 기호 및 회로

11.4 논리 회로 설계



여기서 잠깐!!

NOT 게이트에서의 작은 동그라미는 논리값을 반대로 바꾸는 역할을 한다. 따라서 이 작은 동그라미는 AND나 OR 등의 게이트에서 출력뿐만 아니라 입력으로도 사용될 수 있다.

- 그 외에 많이 쓰이는 게이트들은 XOR 게이트는 '**익스클루시브(Exclusive) OR**' 게이트라고 읽음
- 두 부울 변수가 서로 다른 값을 가질 경우에만 출력이 1이 됨
- XOR 게이트의 기호는 OR 게이트의 왼쪽에 활 모양의 곡선을 추가한 것임

NAND 게이트는 AND 게이트를 부정하는 것임




NOR 게이트는 OR 게이트를 부정하는 것임

Exclusive-NOR 게이트는 exclusive-OR 게이트를 부정하는 것임

11.4 논리 회로 설계







〈표 11.2〉 여러 가지 논리 게이트들의 기호와 진리표

게이트	기호	수식	진리표															
AND		$x = A \cdot B$	<table><tr><th>A</th><th>B</th><th>x</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	x	0	0	0	0	1	0	1	0	0	1	1	1
A	B	x																
0	0	0																
0	1	0																
1	0	0																
1	1	1																
OR		$x = A + B$	<table><tr><th>A</th><th>B</th><th>x</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	x	0	0	0	0	1	1	1	0	1	1	1	1
A	B	x																
0	0	0																
0	1	1																
1	0	1																
1	1	1																
NOT(inverter)		$x = A'$	<table><tr><th>A</th><th>x</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	A	x	0	1	1	0									
A	x																	
0	1																	
1	0																	

11.4 논리 회로 설계

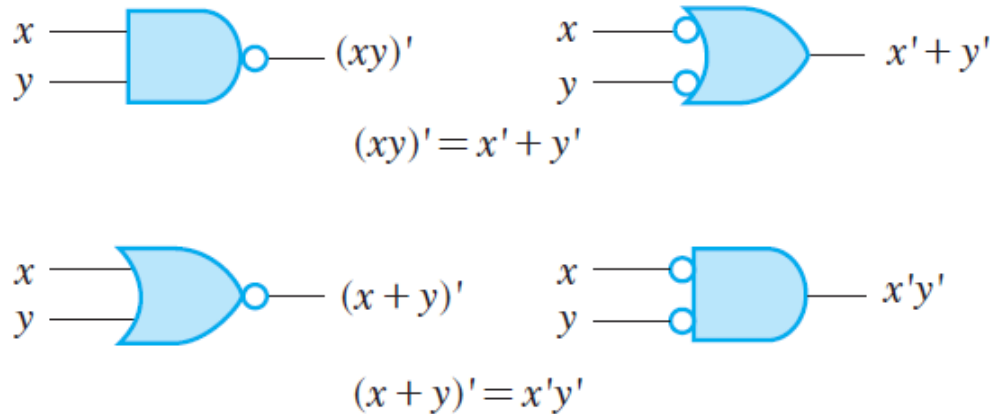


NAND		$x = (A \cdot B)'$	<table><tr><th>A</th><th>B</th><th>x</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	x	0	0	1	0	1	1	1	0	1	1	1	0
A	B	x																
0	0	1																
0	1	1																
1	0	1																
1	1	0																
NOR		$x = (A + B)'$	<table><tr><th>A</th><th>B</th><th>x</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	x	0	0	1	0	1	0	1	0	0	1	1	0
A	B	x																
0	0	1																
0	1	0																
1	0	0																
1	1	0																
exclusive-OR(XOR)		$x = (A \oplus B)$ 또는 $x = A'B + AB'$	<table><tr><th>A</th><th>B</th><th>x</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	x	0	0	0	0	1	1	1	0	1	1	1	0
A	B	x																
0	0	0																
0	1	1																
1	0	1																
1	1	0																
exclusive-NOR		$x = (A \oplus B)'$ 또는 $x = AB + A'B'$	<table><tr><th>A</th><th>B</th><th>x</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	x	0	0	1	0	1	0	1	0	0	1	1	1
A	B	x																
0	0	1																
0	1	0																
1	0	0																
1	1	1																

11.4 논리 회로 설계



- NAND 게이트와 NOR 게이트에 드 모르간의 법칙을 적용하면 다음과 같이 다른 표현이 가능함



〈그림 11.10〉 NAND와 NOR 게이트의 다른 표현



여기서 잠깐!!

모든 부울 함수는 최소항의 부울 합으로 나타낼 수 있으며, 각 최소항은 부울 곱으로 만들 수 있다. 따라서 부울 연산자 AND, OR, NOT만으로도 사실상 모든 논리 함수들을 나타낼 수 있다. 이것을 논리 함수의 **완전성(completeness)**이라고 한다.

11.4 논리 회로 설계



예제 11-14

다음 부울식을 논리 회로로 표현해보자.

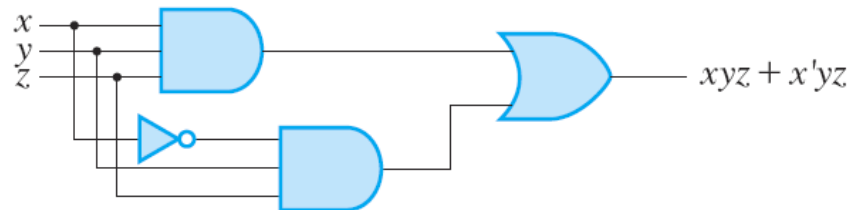
(1) yz

(2) $xyz + x'yz$

풀이 (1) 하나의 AND 회로를 사용하여 만든다.



(2) 두 개의 AND 회로와 하나의 NOT 회로, 그리고 하나의 OR 회로를 사용하여 만든다.



11.4 논리 회로 설계



$x \backslash yz$	00	01	11	10
0			1	
1			1	

$xyz + x'yz = yz$

〈그림 11.11〉 예제 11-14 (2)의 카노우맵

- 부울식을 간소화함으로써 예제 11-14 (2)의 복잡한 회로를 (1)의 간단한 회로로 표현할 수 있음
- 회로를 설계하기 전에 먼저 부울식을 간소화하는 과정이 필요함

11.4 논리 회로 설계



예제 11-15

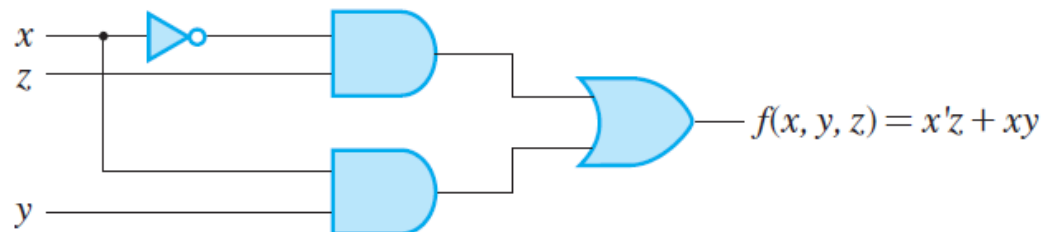
다음 부울 함수를 간소화하고 간소화된 함수의 논리 회로를 그려보자.

$$f(x, y, z) = x'y'z + x'yz + xyz + xyz'$$

풀이 먼저 카노우맵을 사용하여 식을 간소화하면 $f(x, y, z) = x'z + xy$ 가 된다.

yz \ x	00	01	11	10
0		1	1	
1			1	1

이것을 논리 회로로 나타내면 다음과 같다.

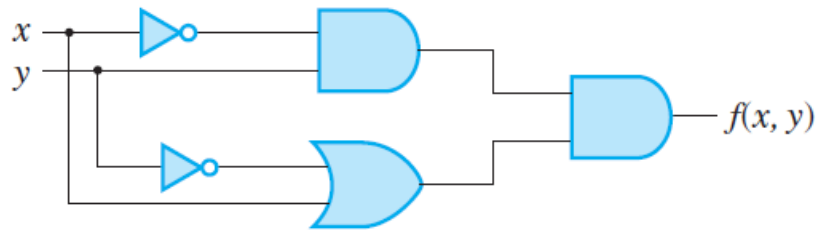


11.4 논리 회로 설계



예제 11-16

다음 논리 회로에 해당하는 부울 함수를 구해보자.



풀이 첫 번째 AND 게이트의 입력은 x' 과 y 이므로 출력은 $x'y$ 가 되고, OR 게이트의 입력은 x 와 y' 이므로 출력은 $x+y'$ 이다. 이 두 출력을 다시 AND 게이트의 입력으로 받으므로 출력되는 부울 함수는

$$\begin{aligned} f(x, y) &= x'y(x+y') \\ &= x'yx + x'yy' \\ &= (x'x)y + x'(yy') \\ &= 0 \cdot y + x' \cdot 0 \\ &= 0 \end{aligned}$$

11.5 논리 회로의 응용



- 논리 회로의 응용 범위는 매우 넓는데, 어느 제품이든지 회로가 들어가는 것은 모두 논리 회로를 응용한 것임
- 일상생활에서 사용하는 TV를 비롯한 가전 제품들이 대부분 논리 회로를 이용한 것임
- 논리 회로는 엘리베이터를 비롯한 수없이 많은 제품에 응용됨

(1) 전자레인지

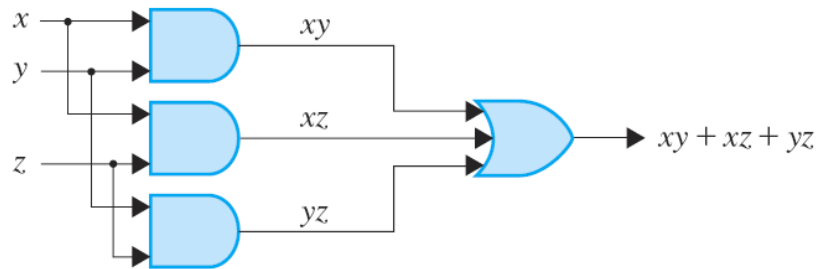
- 우리가 가정에서 사용하는 전자레인지에 이용되는 논리 회로를 생각해 봄
- 전자레인지는 우리가 사용하지 않을 때에는 문이 닫혀 있고(AND), 타이머가 준비되어 있으며(AND), 시작 버튼을 누르면(AND) 전자레인지가 작동됨
- 논리 회로는 우리 생활과 매우 밀접한 관계가 있음

11.5 논리 회로의 응용



(2) 전자투표기

- 3명으로 구성된 어떤 위원회에서 의사 결정을 할 때 '찬성' 또는 '반대' 중의 하나로 투표를 하는데 2명 이상이 '찬성'을 할 경우 안건이 통과된다고 가정함
- 안건이 통과되는지의 여부를 즉석에서 결정할 수 있는 소규모 전자투표기의 논리 회로는 아래 그림으로 표현됨



〈그림 11.12〉 전자 투표 회로

11.5 논리 회로의 응용

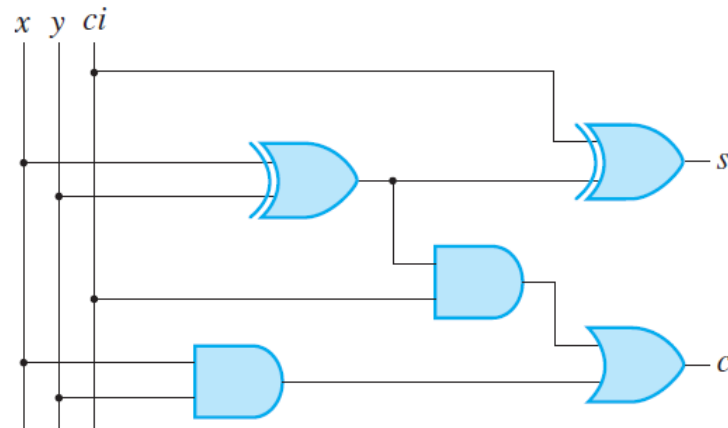


- 전자 투표 회로에서 x, y, z 중 2개씩 묶어서 AND 회로들을 구성됨
- 3개의 AND 게이트 중 어느 2개만 1이 되기만 하면 다음 단계인 OR 게이트에서는 1이 나옴
- 이 논리 회로는 우리가 원하는 것을 만족시킴



(3) 가산기(Adder)

컴퓨터의 중앙처리장치(CPU)에 있는 **산술논리장치 (Arithmetic Logic Unit : ALU)** 내에는 숫자들을 더하는 기능을 가진 가산기가 내장되어 있음



〈그림 11.13〉 가산기 회로

11.6 부울 대수의 응용과 4차 산업혁명과의 관계

(1) 부울 대수의 응용 분야

① 복잡하고 빠른 논리 회로를 이용하는 슈퍼컴퓨터

부울 대수를 이용하여 복잡한 논리 회로를 이용하는 예로는 슈퍼컴퓨터(Super computer)를 들 수 있다. 슈퍼컴퓨터란 그 당시의 가장 우수한 범용 컴퓨터보다 한두 단계 앞선 최첨단 컴퓨터를 말한다. 'Titan' 이란 슈퍼컴퓨터는 1초당 1경 7,590억 번의 연산이 가능한데, <그림 11.14>와 같은 슈퍼컴퓨터는 비행기나 자동차의 설계, 일기예보, 3차원 그래픽 시뮬레이션 등에 활용되고 있다.



<그림 11.14> 부울 대수를 활용하는 슈퍼컴퓨터

11.6 부울 대수의 응용과 4차 산업혁명과의 관계



② 디지털 가전제품에 논리 회로가 필수적임

부울 대수를 이용한 논리 회로의 응용은 우리의 일상생활에서 이루 셀 수 없을 만큼 많다. 전기와 회로를 이용하는 대부분의 제품은 부울 대수를 이용한 논리 회로를 사용한다. 예를 들어 컴퓨터, 프린터, 스마트폰, TV, 라디오, 냉장고, 에어컨, 가습기 등의 가전제품에 논리 회로가 널리 사용된다.

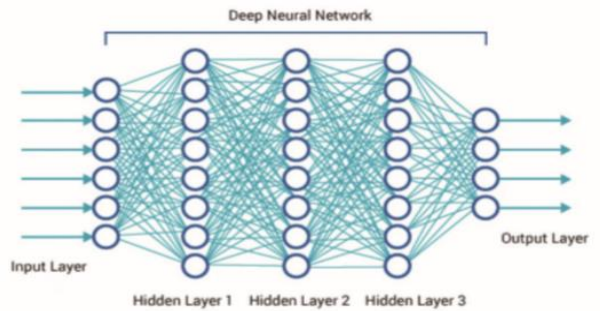
(2) 부울 대수와 4차 산업혁명과의 관계 : 신경망 기술

신경망(Neural Network)은 컴퓨터를 사용하여 인간의 신경세포인 뉴런(neuron)을 모델링하는 4차 산업혁명의 핵심 기술 분야 중 하나다. 즉 인간의 지능으로 할 수 있는 영역 중 학습이나 인식 등을 컴퓨터가 할 수 있도록 하는 방법을 연구하는 분야이다.

지능적인 로봇이 인간을 대신하여 문자나 숫자를 인식하거나 음성을 인식하는 것은 모두 신경망의 영역에 속한다. 신경망은 <그림 11.15>와 같은 구조를 가지는데 구글에서 만든 바둑 프로그램인 **알파고(AlphaGo)**는 프로기사의 능력을 뛰어넘는 바둑 실력을 가지고 있다.

4차 산업혁명에서 중요한 역할을 하는 신경망을 구현하기 위해서는 부울 대수를 활용하는 초고속의 논리 회로가 필요하다.

11.6 부울 대수의 응용과 4차 산업혁명과의 관계



〈그림 11.15〉 신경망 구조와 신경망 바둑 프로그램

요약 및 생활 속의 응용



- 부울 대수는 1854년 영국의 수학자 부울이 수학적 논리 형태로 처음 소개하였는데, 0과 1의 2진수 표현으로 명제의 참과 거짓, 전기 신호의 유와 무, 스위치의 ON과 OFF 등을 나타낼 수 있다.
- 부울식은 두 원소를 가지는 집합 $A = \{0, 1\}$ 와 이항 연산자 $+$ (OR)과 \cdot (AND), 단항 연산자인 $'$ (NOT)으로 표시되는 식을 말한다. 부울식에서 연산자의 연산 우선순위는 $' > \cdot > +$ 순이다.
- 부울 함수식을 간단히 표현하는 방법에는 부울식의 기본 법칙을 사용하는 방법과 카노우맵을 사용하는 방법이 있다.
- 카노우맵은 1953년 카노우에 의해 제안되었는데, 사각형 모양의 표를 이용하여 비교적 쉽게 부울 함수를 간소화할 수 있는 방법이다. 그러나 통상 6개 이하의 변수를 다루는 데만 활용할 수 있다.
- 카노우맵은 부울 변수들에 대한 최소항들을 도표로 그린 후, 인접한 최소항들을 함께 묶어서 표현함으로써 부울 함수를 간소화하는 방법이다.

요약 및 생활 속의 응용



- 카노우맵을 사용하여 인접한 항들끼리 묶을 때는 항상 2의 배수로 묶는다. 3개가 인접해 있을 경우에는 가운데 있는 항이 양쪽으로 사용되어 2개씩 따로 묶어서 나타낸다.
- 카노우맵에서는 한꺼번에 많은 개수의 항을 묶어서 식을 표현할수록 훨씬 간단한 부울식을 구할 수 있다.
- 부울 연산자 AND, OR, NOT만으로 사실상 모든 논리 함수들을 나타낼 수 있다. 이것을 논리 함수의 완전성이라고 한다.
- 우리가 일상생활에서 사용하는 TV를 비롯한 가전 제품들이 대부분 논리 회로를 이용한 것이며, 논리 회로는 엘리베이터를 비롯한 수없이 많은 제품에 응용되고 있다.
- 부울 대수의 응용 분야로는 복잡하고 빠른 논리 회로를 이용하는 슈퍼 컴퓨터, 디지털 가전제품 등에 논리 회로가 필수적이다. 부울 대수와 4차 산업혁명과의 관계로는 신경망 기술을 들 수 있다.

11

CHAPTER

부울 대수

Boolean Algebra

CHAPTER 11
FINISH

