Chapter 2
Boolean Algebra

전자공학부
박 준 구

강의 개요

- 부울 대수
  - 부울 대수란?
  - 기본 연산
  - 진리표
  - 교환, 결합,분배 법칙
  - 간략화란?
  - SOP(Sum of Product), POS(Product of Sum)
  - 드모르간 법칙

2.1 Introduction

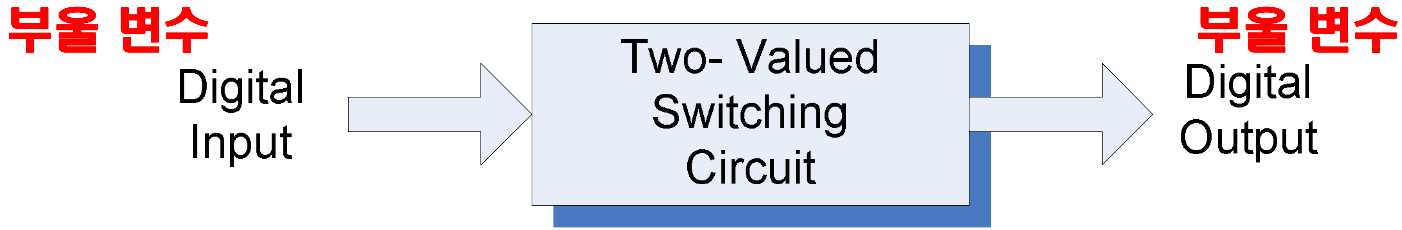
✓ 디지털 시스템의 논리설계를 위해 필요한 수학: **부울대수(Boolean Algebra)**

✓ 부울대수: 집합론의 부울대수, 수학적 논리의 부울대수, 스위칭 부울대수 등

✓ 우리가 취급하는 스위칭 소자는 2 상태 소자.

(높은 출력 전압과 낮은 출력전압을 갖는 2 상태 소자)

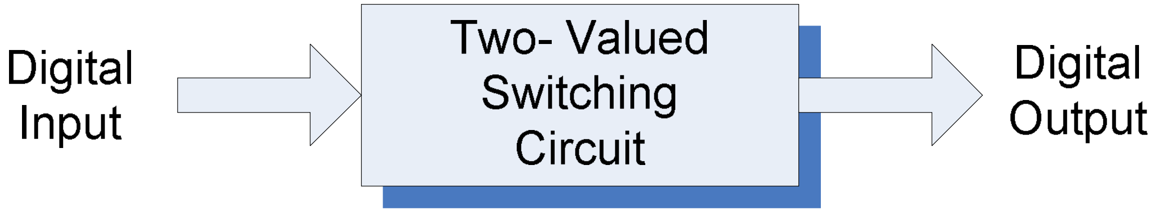
✓ 논리회로 설계에서의 부울대수는 2진 부울대수 또는 2진 스위칭 대수라고 함.



부울 변수는 단지 두 개의 서로 다른 값만을 갖는다.

✓ 즉 2진 부울 변수가 가질수 있는 값은 “0” 과 “1” 또는 참과 거짓, High와 Low “F” 와 “T” 등으로 표현된다.

2.1 Introduction



✓ 부울 대수에서 사용하는 "0" 과 "1" 은 수치적인 값이 아니다.
✓ 논리회로에서 다른 두 가지 상태와 스위칭 변수의 두값을 표한 할 뿐.

✓ 논리 회로에서 "0" 은 낮은 전압대를 표현, "1" 은 높은 전압대를 표현한다.
✓ 스위칭 회로세서 "0" 은 열린 회로를 , "1" 은 닫힌 회로를 표현한다.

2.2 Basic Operations

부울대수의 기본 연산: AND, OR, NOT(Complementation)

✓ 부울변수가 가질 수 있는 2개의 값으로 0과 1을 사용한다고 가정하자.

◆ 1. NOT(Complementation) 연산

✓ 부울변수 x의 값이 0인 경우, 이 부울변수에 NOT 연산을 하면 부울변수 x는 1이 된다. 마찬가지로 부울변수 x가 1의 값을 가지고 있는 경우, NOT 연산에 의해 부울 변수 x는 0이 된다.

✓ NOT이라는 부울연산으로 프라임 (’)부호를 사용한다.

$$0' = 1 \quad \text{과} \quad 1' = 0$$

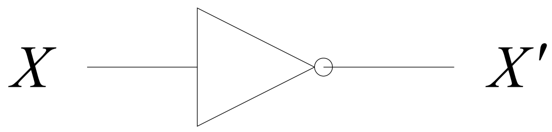
✓ 만일 x가 스위칭 변수인 경우

$$X = 0\text{이면} \quad X' = 1\text{이고} \quad X = 1\text{이면} \quad X' = 0\text{이다.}$$

2.2 Basic Operations

◆ 1. NOT 연산

✓ NOT의 또 다른이름을 반전(Inverse)이라고 한다. 부울변수 x의 반전을 나타내는 전자회로는 Inverter이다. Inverter의 기호는



✓ 논리 0이 낮은 전압이라하고 논리 1이 높은 전압이라고 하면, 인버터 입력에서의 낮은 전압은 그 출력에서는 높은 전압으로 나타나고, 그 반대도 성립한다.

2.2 Basic Operations

◆ 2. AND 연산

✓ AND 연산의 정의는 다음과 같다.

A	B	$C = A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

✓ 여기서 "•" 는 AND 연산을 나타낸다(비록 2진수 곱셈처럼 보이지만 여기서의 0과 1은 2진수가 아니라 부울 상수이므로 곱셈이 아니고 AND 연산이다.

부울 변수: A, B , C

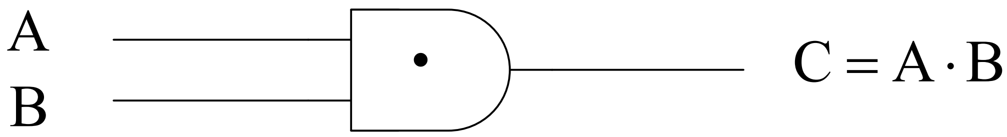
부울 상수: 0, 1

부울 식: C= A • B

2.2 Basic Operations

◆ 2. AND 연산

✓ A와 B가 1일 때만 C=1가 된다. AND 연산을 수행하는 논리 Gate는 다음과 같이 표시한다.



✓ “ • ” 기호는 부울식에서는 일반적으로 생략하여 A • B 대신 AB로 표기한다.

2.2 Basic Operations

◆ 3. OR 연산

✓ OR 연산의 정의는 다음과 같다.

A	B	$C = A + B$
0	0	0
0	1	1
1	0	1
1	1	1

✓ 여기서 " + " 는 OR 연산을 나타낸다(비록 2진수 덧셈처럼 보이지만 여기서의 0과 1은 2진수가 아니라 부울 상수이므로 덧셈이 아니고 OR 연산이다.

부울 변수: A, B , C

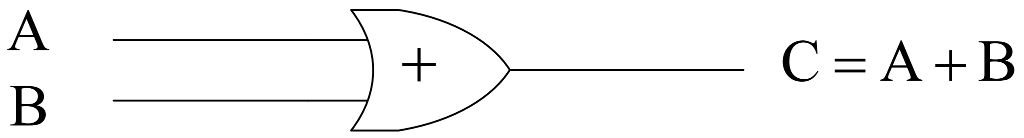
부울 상수: 0, 1

부울 식: $C = A + B$

2.2 Basic Operations

◆ 3. OR 연산

✓ A 또는 B가 1일 때 C=1가 된다. OR 연산을 수행하는 논리 Gate는 다음과 같이 표시한다.

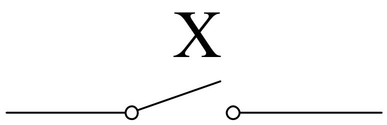


✓ OR 연산을 내포적 OR(Inclusive-OR)라고도 한다.

2.2 Basic Operations

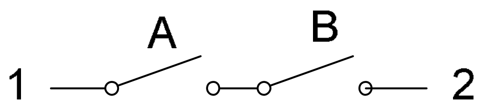
◆ switching 대수의 적용

✓ Switch X가 0이면 스위치는 열려 있는 것으로 간주하고, X = 1이면 스위치는 닫혀 있는 것으로 간주하고, 부울 대수를 Switching 대수로 표현해 보자.



X = 0 ➔ 스위치 열림
X = 1 ➔ 스위치 담힘

◆ Switch 직렬연결



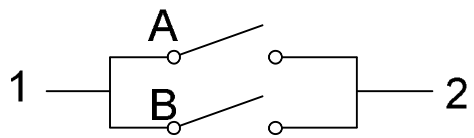
T = 0 ➔ 단자 1과 2는 열린 회로
T = 1 ➔ 단자 1과 2는 닫힌 회로

✓ 스위치 A가 닫히고, 스위치 B 가 닫혔을 때만 단자 1과 2는 닫힌 회로가 된다. 이를 부울 대수로 나타내면 AND 연산과 같다.

$$T = A \cdot B$$

2.2 Basic Operations

◆ Switch 병렬연결



T = 0 ➔ 단자 1과 2는 열린 회로
T = 1 ➔ 단자 1과 2는 닫힌 회로

✓ 스위치 A가 닫혀 있거나 스위치 B 가 닫혀있으면 단자 1과 2는 닫힌 회로가 된다. 이를 부울 대수로 나타내면 OR 연산과 같다.

$$T = A + B$$

그러므로 직렬로 연결된 스위치는 AND 연산을,
병렬로 연결된 스위치는 OR 연산을 수행한다.

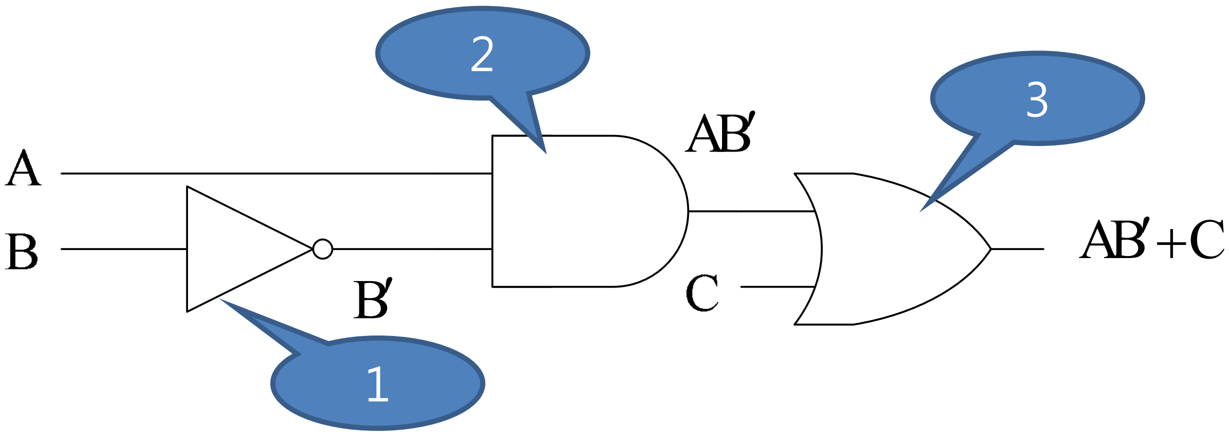
2.3 Boolean Expressions and Truth

✓ 부울식은 하나 또는 이상의 부울 변수(A, B, X, Y 등)와 상수(0또는 1) 그리고 기본 연산(AND, OR, NOT)으로 구성된 식이다.

✓ 가장 간단한 부울식은 단 하나의 부울 변수나 상수로만 구성된 식이다.

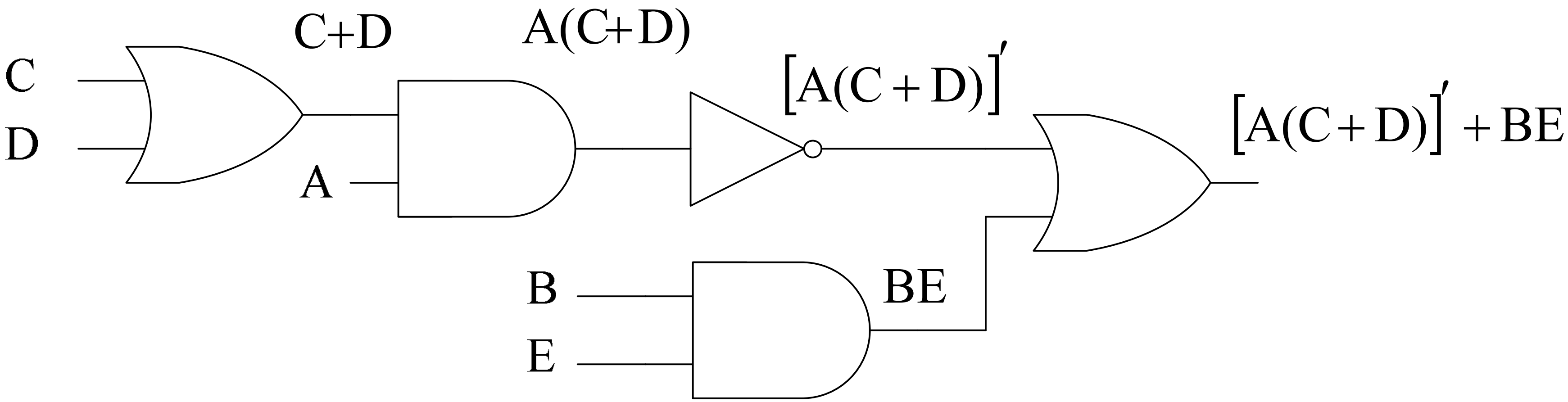
$$AB' + C \qquad (2\text{-}1)$$

✓ 식(2-1)에서 부울 변수 B의 보수화가 먼저 실행되고, 부울변수 A와 AND 연산을 한다. 그리고 마지막으로 부울변수 C와 OR연산의 순서로 진행된다.



2.3 Boolean Expressions and Truth

$$[A(C+D)]' + BE \qquad (2\text{-}2)$$



✓ 식(2-2)에서 A=B=C=1이고, D=E=0이면 부울 식의 최종 값은 다음과 같다.

$$[A(C+D)]' + BD = [1(1+0)]' + 1\cdot 0 = [1\cdot 1]' + 0 = 0 + 0 = 0$$

2.3 Boolean Expressions and Truth

◆ 문자(Literal)

✓ 부울식에서 변수, 변수의 보수, 그리고 변수와 변수의 AND 연산은 각각 문자라고 한다.

$$ab'c+a'b+a'bc'+b'c'$$

✓ 위의 부울식에서는 모두 10개의 문자로 구성되었다.

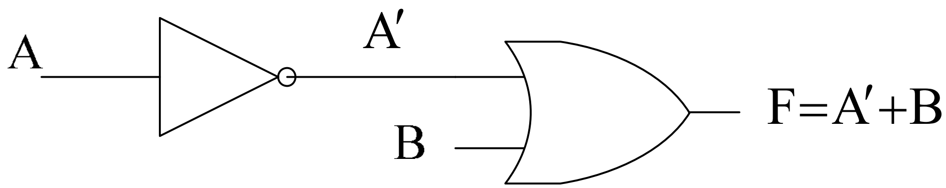
◆ 진리표(Truth Table) 또는 조합표

✓ 진리표는 부울식에 있는 부울 변수들 값의 가능한 모든 조합에 대한 부울식의 값을 나타낸다.

✓ n-변수 식의 진리표는 2^n 개의 행으로 나타낸다.

2.3 Boolean Expressions and Truth

◆ 진리표(Truth Table) 또는 조합표



A	B	A′	F=A′+B
0	0	1	1
0	1	1	1
1	0	0	0
1	1	0	1

2.3 Boolean Expressions and Truth

$$
\begin{aligned}
AB' + C &= (A+C)(B'+C) \\
&= AB' + AC + CB' + CC \qquad \textbf{[2-3]} \\
&= AB' + C(A+B'+1)
\end{aligned}
$$

A	B	C	B'	AB'	$AB'+C$	$A+C$	$B'+C$	$(A+C)(B'+C)$
0	0	0	1	0	0	0	1	0
0	0	1	1	0	1	1	1	1
0	1	0	0	0	0	0	0	0
0	1	1	0	0	1	1	1	1
1	0	0	1	1	1	1	1	1
1	0	1	1	1	1	1	1	1
1	1	0	0	0	0	1	0	0
1	1	1	0	0	1	1	1	1

2.4 Basic Theorems Tables

◆ 하나의 부울변수를 갖는 부울대수의 기본법칙 및 정리

✓ 부울변수 X가 0이나 1일때의 연산

$$X+0=X \quad (2-4) \qquad X\cdot 1=X \quad (2-4D)$$

$$X+1=1 \quad (2-5) \qquad X\cdot 0=0 \quad (2-5D)$$

✓ 멱등법칙(Idempotent law)

$$X+X=X \quad (2-6) \qquad X\cdot X=X \quad (2-6D)$$

✓누승법칙(Involution law)

$$(X')'=X \quad (2-7)$$

✓상보의 법칙(law of complementarity)

$$X+X'=1 \quad (2-8) \qquad X\cdot X'=0 \quad (2-8D)$$

2.4 Basic Theorems Tables

◆ 하나의 부울변수를 갖는 부울대수의 기본법칙 및 정리에 대한 증명

✓ 부울변수 X가 가질 수 있는 두 가지 값인 0과 1을 모두 보여 줌으로서 증명할 수 있다. 예를 들어 정리 (2-8)은

X = 0 이면, 0+0' = 0+1 = 1 이고,
X = 1 이면, 1+1' = 1+0 = 1 이다.

✓또한 이들 정리는 부울변수 X 대신 어떤 식을 대입해도 성립한다.

$(AB'+D)E+1=1$ 정리 (2-5) 사용

$(AB'+D)(AB'+D)'=0$ 정리 (2-8D) 사용

2.4 Basic Theorems Tables

◆ 스위칭 회로를 이용한 기본 정리의 설명

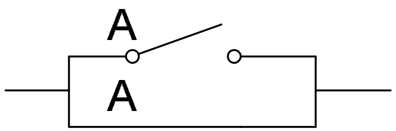
✓ $A \cdot A = A$ 의 설명

✓ $A + A = A$ 의 설명

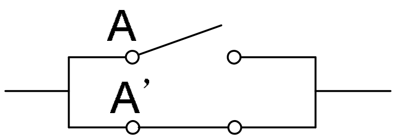
✓ $A + 0 = A$ 의 설명

2.4 Basic Theorems Tables

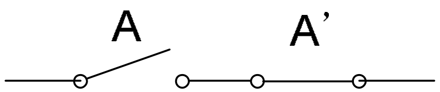
- ✓ $A+1=1$ 의 설명



- ✓ $A+A'=1$ 의 설명



- ✓ $A \cdot A'=0$ 의 설명



2.5 Commutative, Associative, and Distributive Laws

◆ 부울대수에서의 AND 연산과 OR 연산에 대한 교환법칙 (Commutative)

$$X+Y=Y+X \qquad (2-9) \qquad\qquad X\cdot Y=Y\cdot X \qquad (2-9D)$$

◆ 부울대수에서의 AND 연산과 OR 연산에 대한 결합법칙 (Associative)

$$(X+Y)+Z=X+(Y+Z)=X+Y+Z \qquad (2-10)$$

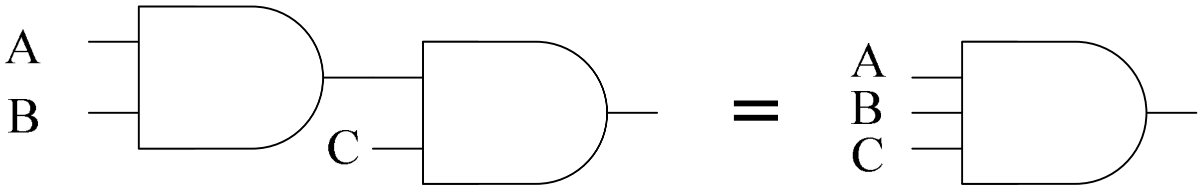
$$(XY)Z=X(YZ)=XYZ \qquad (2-10D)$$

✓AND에 대한 결합법칙을 진리표를 통해 증명하면 다음과 같다.

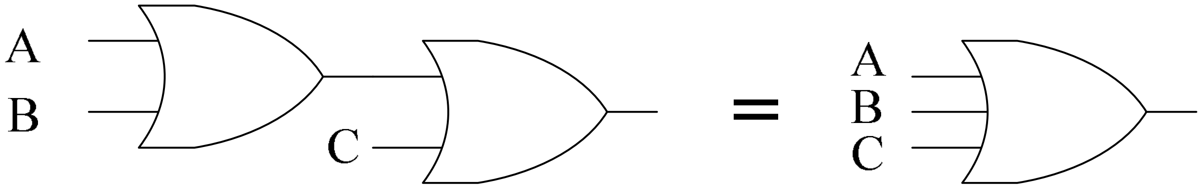
X	Y	Z	XY	YZ	$(XY)Z$	$X(YZ)$
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	0	1	0	0
1	0	0	0	0	0	0
1	0	1	0	0	0	0
1	1	0	1	0	0	0
1	1	1	1	1	1	1

2.5 Commutative, Associative, and Distributive Laws

✓AND연산과 OR 연산에 대한 결합법칙은 다음과 같이 논리 Gate로도 설명가능하다.



$$(AB)C = ABC$$



$$(A+B)+C = A+B+C$$

✓ 둘 또는 그 이상의 부울변수들이 모두 AND연산인 경우 변수의 모든 값이 1이면 결과 값은 1이다. 만약 변수들 중에서 어떤 것이 0을 가지고 있다면, AND 연산의 결과는 0이다.

$$X = Y = Z = 1 \text{일 때}, \ XYZ = 1$$

2.5 Commutative, Associative, and Distributive Laws

✓ 둘 또는 그 이상의 부울변수들이 모두 OR 연산인 경우 변수들 중에서 어느 한 변수가 1이면 결과 값은 1이다. 모든 변수들이 0이면 OR 연산의 결과는 0이다. $X=Y=Z=0$일 때, $X+Y+Z=0$

◆ 부울대수에서의 분배법칙(제1법칙)

$$X(Y+Z)=XY+XZ \qquad (2\text{–}11)$$

◆ 부울대수에서의 분배법칙(제2법칙)

$$X+(YZ)=(X+Y)(X+Z) \qquad (2\text{–}12)$$

2.5 Commutative, Associative, and Distributive Laws

◆ 부울대수에서의 분배법칙(제2법칙)의 증명

$$
\begin{aligned}
(X+Y)(X+Z) &= (X+Y)X+(X+Y)Z \\
&= XX+XY+XZ+YZ \\
&= X+XZ+XY+YZ \\
&= X\cdot 1+XZ+XY+YZ \\
&= X(1+Z+Y)+YZ \\
&= X+YZ
\end{aligned}
$$

2.6 Simplification Theorems

◆ 부울대수에서의 간략화 법칙

$$XY + XY' = X \qquad (2-12) \qquad (X+Y)(X+Y') = X \qquad (2-12D)$$

(2-12)의 증명) $XY + XY' = X(Y+Y') = X$

(2-12D)의 증명) $(X+Y)(X+Y') = X + (YY') = X + 0 = X$

$$X + XY = X \qquad (2-13) \qquad X(X+Y) = X \qquad (2-13D)$$

(2-13)의 증명) $X + XY = X \cdot 1 + XY = X(1+Y) = X$

(2-13D)의 증명) $X(X+Y) = XX + XY = X + XY = X$

$$(X+Y')Y = XY \qquad (2-14) \qquad XY' + Y = X + Y \qquad (2-14D)$$

(2-14)의 증명) $(X+Y')Y = XY + Y'Y = XY + 0 = XY$

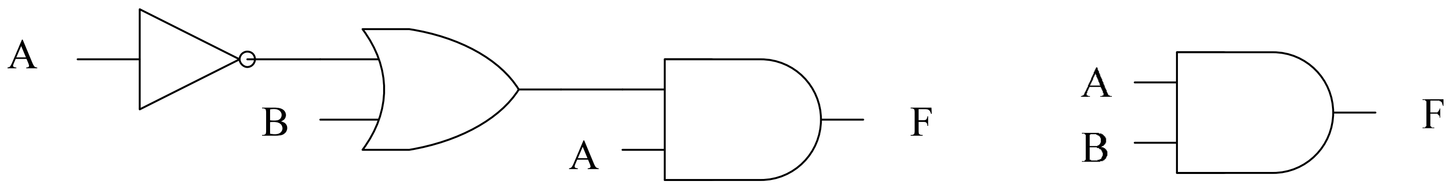
(2-14D)의 증명) $XY' + Y = (X+Y)(Y'+Y) = X + Y$

2.6 Simplification Theorems

✓ 간략화 정리는 어떤 부울식을 더 간단한 부울식으로 대체될 수 있다는 것을 의미한다. 디지털 시스템에서 시스템을 나타내는 부울식은 논리 게이트로 구현되기 때문에 부울식을 간략화 한다는 것은 부울식을 나타내는 논리 게이트 수의 감소시켜 비용적인 이득을 볼 수 있다는 장점이 있다.

✓ 예를 들어

$F = A(A' + B)$ $\qquad$ $F = A(A' + B) = AA' + AB = AB$



2.6 Simplification Theorems

예제 1) $Z = A'BC + A'$ **를 간략화 하라.**

$$Z = A'BC + A' = A'BC + A' \cdot 1 = A'(BC+1)$$
$$= A'$$

예제 2) $Z = [A + B'C + D + EF][A + B'C + (D + EF)']$ **를 간략화 하라.**

$$Z = [\quad X \quad + \quad Y \quad][\quad X \quad + \quad Y' \quad]$$
$$= X$$
$$= A + B'C$$

예제 3) $Z = (AB + C)(B'D + C'E') + (AB + C)'$ **를 간략화 하라.**

$$Z = \quad Y' \qquad X \quad + \quad Y$$
$$= X + Y$$
$$= (B'D + C'E') + (AB + C)'$$

2.7 Multiplying Out and Factoring

◆ 논리곱의 합(Sum of Product) 형식

✓ 모든 곱들이 단 한 변수의 곱으로 이루어질 때 이 식을 논리곱의 합 형식이라고 한다.

$$AB' + CD'E + AC'E$$

$$ABC' + DEFG \; + H$$

$$A + B' + C + D'E$$

✓ 그러나 다음 식은 논리곱의 합 형식이 아니다.

$$(A+B)CD+EF$$

✓ 위와 같은 식을 논리곱의 합 형식으로 표현하기 위하여 곱셈전개 (Multiplying Out)를 하여 구할 수 있다.

$$(A+B)CD + EF = ACD + BCD + EF$$

2.7 Multiplying Out and Factoring

◆ 논리곱의 합(Sum of Product) 형식

- ✓ 논리곱의 합 형식을 만들기 위하여 곱셈전개를 하는 경우 제 2분배 법칙을 먼저 적용하는 것이 보다 간결한 형태의 논리곱의 합 형식을 얻을 수 있다.

$$(A+BC)(A+D+E) = A+BC(D+E) = A+BCD+BCE$$

$$(X+Y)(X+Z) = X+YZ$$ (제 2 분배 법칙)

- ✓ 그러나 제1의 분배 법칙을 사용하는 경우에도 동일한 논리곱의 합 형식을 얻을 수 있지만 복잡한 과정을 거쳐야 함을 알 수 있다.

$$\begin{aligned}(A+BC)(A+D+E) &= A+AD+AE+ABC+BCD+BCE \\ &= A(1+D+E+BC)+BCD+BCE \\ &= A+BCD+BCE\end{aligned}$$

- ✓ 위와 같은 식을 방법으로 문제를 해결하는 대신 제2의 분배법칙을 사용하면 많은 시간을 절약할 수 있을 것이다.

2.7 Multiplying Out and Factoring

◆ 논리합의 곱(Product of Sum) 형식

✓ 모든 곱들이 단일 변수들의 합으로 이루어질 때 이 식을 논리합의 곱 형식이라고 한다.

$$(A+B')(C+D'+E)(A+C'+E')$$

$$(A+B)(C+D+E)F$$

$$AB'C'(D'+E)$$

✓ 그러나 다음 식은 논리합의 곱 형식이 아니다.

$$(A+B)(C+D)+EF$$

2.7 Multiplying Out and Factoring

◆ 논리합의 곱(Product of Sum) 형식

예제 1) $A+B'CD$ **를 인수화 하라.**

$$A+B'CD=(A+B')(A+CD)=(A+B')(A+C)(A+D)$$

예제 2) $AB'+C'D$ **를 인수화 하라.**

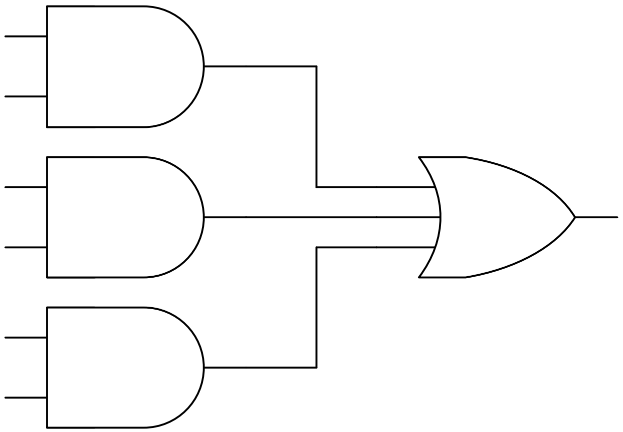
$$AB'+C'D=(AB'+C')(AB'+D)$$
$$=(A+C')(B'+C')(A+D)(B'+D)$$

예제 3) $C'D+C'E'+G'H$ **를 인수화 하라.**

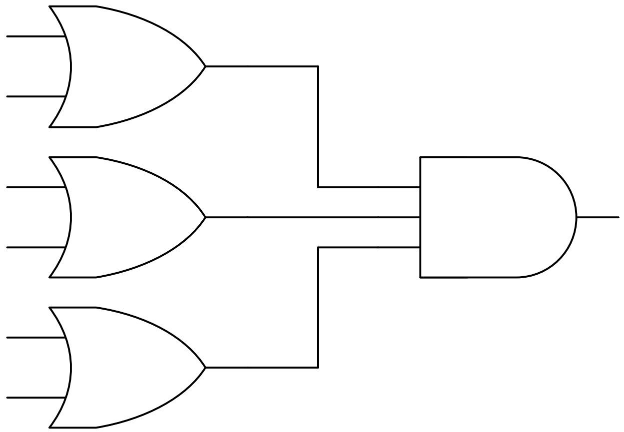
$$C'D+C'E'+G'H=C'(D+E')+G'H$$
$$=(C'+G'H)(D+E'+G'H)$$
$$=(C'+G')(C'+H)(D+E'+G')(D+E'+H)$$

2.7 Multiplying Out and Factoring

- ✓ 논리곱의 합식을 논리 게이트로 표시할 때 하나 이상의 AND 게이트가 입력단이 되고, 이 AND 게이트의 출력들이 OR 게이트의 입력이 되는 형태를 구성한다.
- ✓ 논리합의 곱식을 논리 게이트로 표시할 때 하나 이상의 OR 게이트가 입력단이 되고, 이 OR 게이트의 출력들이 AND 게이트의 입력이 되는 형태를 구성한다.



논리곱의 합식의 논리 게이트 표현



논리합의 곱식의 논리 게이트 표현

2.8 DeMorgan's Laws

◆ DeMorgan의 정리

$$(X+Y)' = X'Y' \qquad (2-21)$$

$$(XY)' = X' + Y' \qquad (2-22)$$

진리표에 의한 DeMorgan의 증명

X	Y	X'	Y'	$X+Y$	$(X+Y)'$	$X'Y'$	XY	$(XY)'$	$X'+Y'$
0	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	0	0	1	1
1	0	0	1	1	0	0	0	1	1
1	1	0	0	1	0	0	1	0	0

2.8 DeMorgan’s Laws

◆ *n* 변수에 대한 DeMorgan의 정리

$$(X_1 + X_2 + X_3 + \cdots + X_n)' = X_1' X_2' X_3' \cdots X_n' \qquad (2-23)$$

$$(X_1 X_2 X_3 \cdots X_n)' = X_1' + X_2' + X_3' + \cdots + X_n' \qquad (2-24)$$

✓ 부울변수 곱의 보수는 부불변수 각각의 보수의 합이다.

✓ 부울변수 합의 보수는 부울변수 각각의 보수의 곱이다.

예제 1) $(A' + B)C'$ **의 보수를 구하라.**

$$[(A' + B)C']' = (A' + B)' + C = AB' + C$$

2.8 DeMorgan's Laws

예제 2) $\left[(AB'+C)D'+E\right]'$ 를 구하라.

$$\left[(AB'+C)D'+E\right]' = \left[(AB'+C)D'\right]' E$$
$$= \left[(AB'+C)'+D\right]E = \left[(AB')'C'+D\right]E = \left[(A'+B)C'+D\right]E$$

◆ 쌍대(Dual)

✓주어진 부울식에서 AND 대신 OR, OR 대신 AND 그리고 1 대신 0을, 0 대신 1로 바꾸어 주어진 부울식의 쌍대(Dual)을 만들 수 있다.

✓이때 변수들과 변수들의 보수는 바꾸지 않고 그대로 둔다.

✓주어진 부울식의 쌍대를 만드는 방법은 주어진 부울식에 보수를 취하고 나서 개개의 변수에 보수를 취하여 구한다.

2.8 DeMorgan’s Laws

예제) $AB' + C$ 의 쌍대를 구하라.

$$(AB' + C)' = (AB')'C' = (A' + B)C'$$

요약

- 부울 대수의 기본
  - 기본 연산
  - 부울식과 진리표 작성 방법
  - 기본 정리들..
    - 멱등 (Idempotent) 법칙
    - 누승 (Involution)의 법칙
    - 상보(Complement)의 법칙
  - 교환, 결합, 분배 법칙
  - 간략화 방법
  - SOP, POS → and, or 로 정리.
  - 드모르간 법칙을 이용한 변환

Homework 2
숙제 #2

숙제는 "연습문제"

2.1 , 2.2 , 2.5 , 2.9 , 2.13 , 2.14 , 2.17 , 2.26 , 2.28

풀어서 제출할 것