Ch. 2 : Boolean Algebra

2.1 Introduction

- Boolean Algebra

  Basic mathematics for logic design

  Variable: X, Y, ... → only two state (0, 1)

- Gate

  A simple electronic circuit that realizes a logical operation

- Truth Table

  모든 가능한 입력 조합에 대해 출력이 어떻게 되는지 정리한 표.

  0 (F), 1 (T) 주로. → Positive logic

  ↳ 반대라면 → Negative logic

- Standard Gates & Symbols

  Buffer : 입력 값이 출력 그대로.

Not (Inverter or Complement) : 입출력을 뒤바꿈.

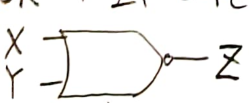
And : 모두 1이어야만 1 출력. 나머지는 0.

OR : 하나라도 1이 있으면 1. 아니면 0.

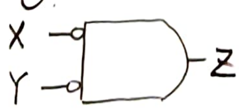
NAND : 모두 1이면 0. 나머지는 1

OR

NOR : 모두 0이면 1, 나머지는 0.



OR



XOR : 같으면 0, 다르면 1.

Equivalence : 같으면 1, 다르면 0

2.2 Basic Operations

- Not (Inverter)

  $0' = 1$, $1' = 0$.

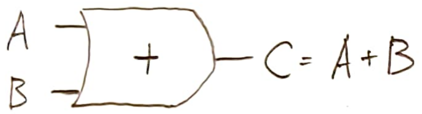
  $X' = 1$ if $X = 0$, $X' = 0$ if $X = 1$

- AND

  $0 \cdot 0 = 0$, $0 \cdot 1 = 0$, $1 \cdot 0 = 0$, $1 \cdot 1 = 1$.

- OR

  $0 + 0 = 0$, $0 + 1 = 1$, $1 + 0 = 1$, $1 + 1 = 1$

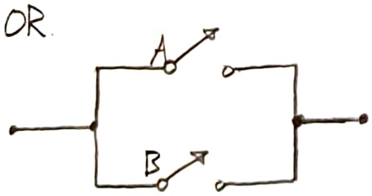


- Apply to Switch

$X = 0 \rightarrow$ open
$X = 1 \rightarrow$ Closed.

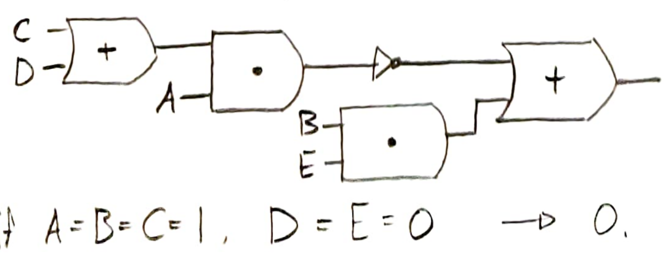
AND

OR.



2.3 Boolean Expressions and Truth Tables

- 예1: $[A(C + D)]' + BE$



if $A = B = C = 1$, $D = E = 0$ $\rightarrow$ 0.

- Literal : 논리식에서의 변수 혹은 변수의 Complement

$ab'c + a'b + a'bc' + b'c' \rightarrow$ 10 literals

- Truth table

n개의 변수 → 2^n 경우의 수

- 정리

Boolean function 세 가지 표현 방법

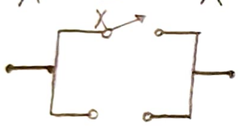
1. Logical expression
2. Truth table
3. Logic Circuit (network)

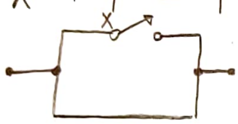
우선순위 : 괄호 우선, NOT AND OR 순.

2.4. Basic Theorems

- Operations with 0, 1

$X + 0 = X$, $X + 1 = 1$





$X \cdot 1 = X$, $X \cdot 0 = 0$

- Idempotent Laws

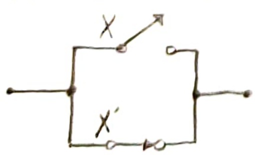
$X + X = X$, $X \cdot X = X$

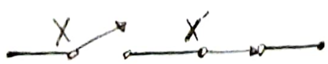
- Involution Laws

$(X')' = X$

- Complementary Laws

$X + X' = 1$, $X \cdot X' = 0$





2.5 Commutative, Associative, and Distributive Laws

- Commutative Laws

$XY = YX$, $X + Y = Y + X$

- Associative Laws

$(XY)Z = X(YZ) = XYZ$

$(X+Y)+Z = X+(Y+Z) = X+Y+Z$

- Distribute Laws

$X(Y+Z) = XY + XZ$

$X + YZ = (X+Y)(X+Z)$ → Valid only Boolean algebra
자주 활용됨

$XYZ = 1$ iff $X = Y = Z = 1$ (AND)
$X + Y + Z = 0$ iff $X = Y = Z = 0$ (OR)

2.6 Simplification Theorems

- Useful Theorems for Simplification

$XY + XY' = X$, $(X+Y)(X+Y') = X$ → 두 식은 Dual.

$X + XY = X$, $X(X+Y) = X$ → 두 식은 Dual.

$(X+Y')Y = XY$, $XY' + Y = X + Y$

2.7 Multiplying Out and Factoring

- Sum of product form

ex) $AB' + CD'E + AC'E$

to obtain a SOF form → Multiplying out using distributive laws.

- Product of Sum form

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

to obtain a POF form → all Sums are the Sum of Single variable

2.8 DeMorgan's Laws

- De Morgan's Laws

$(X+Y)' = X'Y'$, $(XY)' = X' + Y'$

- De Morgan's Laws for n variables

$(X_1 + X_2 + \cdots + X_n)' = X_1' X_2' \cdots X_n'$

$(X_1 X_2 X_3 \cdots X_n)' = X_1' + X_2' + \cdots + X_n'$

- example : Inverse of $F = A'B + AB'$

$F' = (A'B + AB')' = (A'B)'(AB')' = (A+B')(A'+B) = AA' + AB + B'A' + BB' = A'B' + AB$

- Dual

"dual" is formed by replacing AND with OR, OR with And, 0 with 1, 1 with 0

$(XYZ\ldots)^D = X+Y+Z+\cdots$, $(X+Y+Z+\ldots)^D = XYZ\ldots$

Dual 형태 얻기 위해선 전체를 Complementing 한 다음 (드모르간), 각각의 리터럴을 Complementing 한다.

~~***~~ 중요 : 어떤 공식이 있으면 그 공식의 Dual 형태 또한 항상 성립함.

어떤 리터럴의 Dual form은 자기 자신 그 자체이다. $A^D = A$.

→ literal : 변수나 변수의 Complement ex) A, A'