5주차 예비보고서

전공: 경영학과 학년: 4학년 학번: 20190808 이름: 방지혁

1.

드모르간의 정리란 크게 논리합의 측면과 집합론의 측면에서 접근할 수 있습니다.
드 모르간의 법칙에서 논리합을 부정하면 논리곱으로 바뀌고 각 변수는 부정되며, 논리곱을 부정하면 논리합으로 바뀌고 각 변수는 부정됩니다.

$\overline{A \cdot B} = \overline{A} + \overline{B}$ (드 모르간의 제 1법칙)

제 1법칙에서는 A와 B의 논리곱에 부정을 취하면 A의 부정과 B의 부정에 논리합을 취해준 것과 같습니다. A와 B가 모두 참이 아니라면 A가 거짓이거나 B가 거짓이게 됩니다.

$\overline{A + B} = \overline{A} \cdot \overline{B}$ (드 모르간의 제 2법칙)

제 2법칙에서는 A와 B의 논리합에 부정을 취한다면 이는 A의 부정과 B의 부정에 논리 곱을 취해준 것과 같습니다. A 혹은 B가 참이 아닌 경우 A도 거짓이고 B도 거짓인 경우와 같다는 것입니다.

2.

동일 법칙 (Identity Laws) : A + 0 = A, A · 1 = A
지배 법칙 (Domination Laws) : A · 0 = 0, A + 1 = 1
등멱 법칙 (Idempotent Laws) : A + A = A, A · A = A
부정 법칙 (Negation Laws) : A + A' = 1, A · A' = 0
교환 법칙 (Commutative Laws) : A + B = B + A, A · B = B · A
결합 법칙 (Associative Laws) :
A + (B + C) = (A + B) + C, A · (B · C) = (A · B) · C
분배 법칙 (Distributive Laws): A · (B + C) = A · B + B · C
드 모르간의 법칙 (De Morgan's Laws):
(A + B)' = A' · B', (A · B)' = A' + B'
이중 부정 법칙 (Double Negation Law) : (A')' = A
흡수 법칙 (Absorption Law) : A + A · B = A, A · (A + B) = A

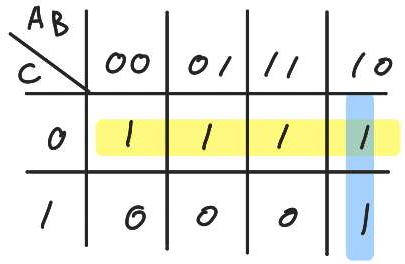
예시)
AB + A'C + BC
= AB + A'C + 1 · BC (동일 법칙)
= AB + A'C + (A + A') · BC (부정 법칙)
= AB + A'C + ABC + A'BC (분배 법칙)
= AB + ABC + A'C + A'BC (교환 법칙)

= AB · (1 + C) + A'C · (1 + B) (동일, 분배법칙)
= AB · 1 + A'C · 1 (지배 법칙)
= AB + A'C (동일 법칙)

AB + A · (CD + CD')
= AB + A · C · (D + D') (분배 법칙)
= AB + A · C · 1 (부정 법칙)
= AB + AC

3.
카르노 맵은 최소 논리곱의 합을 대수적으로 간소화를 하는 것보다 진리표를 이용하여 최소 논리식을 쉽게 구하는 접근법입니다. 변수의 개수에 따라 크기가 결정되며, 각 칸에 값을 삽입합니다. 간단히 설명하자면 가까이 붙어있는 1 또는 0(접근 방법에 따라 다름)를 서로 그룹화시켜 주는 것입니다.
이는 기존의 대수적 관점의 접근 방식보다 논리식을 더 간단하고 빠르게 얻어낼 수 있으며, 간단한 회로 즉, 작은 회로 설계 시에는 효과적입니다. 반면, 큰 회로 설계 시 카르노 맵이 지나치게 비대해져 소프트웨어 사용 같은 다른 방식을 취하게 됩니다.
예시)

A	B	C	F
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0



f = C' + AB'
이 때 AB의 순서가 00 01 10 11이 아닌 00 01 11 10인 이유는 양옆끼리는 하나의 비트만 차이가 나야 하기 때문입니다. 이런 식으로, 2^n의 크기를 가지는 직사각형들로 묶어주고 식으로 다시 나타내야 합니다. 또한, 카르노 맵을 이용해 간소화를 할 때에는 어떻게 항들을 묶느냐에 따라 여러 가지 답이 나올 수 있습니다. 4 변수 카르노맵

은 4X4표를 통해, 5 변수 카르노맵은 3차원으로 4X4표 두 개를 이용해 논리회로를 간소화할 수 있습니다.

4.

앞서 위의 항목들에서 서술했던 카르노 맵에 의한 간소화 방법은 변수의 개수가 4개를 초과한다면, 다소 어려워지는 경향이 있습니다. 이에 대해 보완 방법인 Quine-McCluskey 알고리즘이 존재합니다. 해당 방법은 도표를 이용해 논리식을 간소화할 수 있는 방법입니다. 이러한 방식은 크게 Prime Implicant 식별과 Prime Implicant 선택으로 나눌 수 있습니다.

우선 Prime Implicant 식별은 먼저 최소 항을 2진수로 변환한 표로 만듭니다. 2진수들 중 1개의 비트만 차이가 나는 최소 항들을 묶어주고, 그걸 다시 표로 구성합니다. 최소항들은 즉, 1의 개수에 따라 그룹으로 나누어지는 것입니다. 1의 개수가 0개인 항: 0000, 1의 개수가 1개인 항 0001, 0010, 0100, 1000 이런 식으로 나누어질 수 있는 것입니다. 각 그룹 내에서 비트 1개의 차이가 나는 항들을 찾습니다. 두 항이 1비트만 차이가 날 경우, 이를 결합해 그 차이가 나는 비트 위치를 -로 표현합니다. 이 때 결합이 되지 않은 항목은 PI(Prime Implicant)가 됩니다. 더 이상 결합될 수 없을 때까지 이 과정을 반복합니다.

이렇게 PI를 구한 후 PI 선택 과정으로 넘어갑니다. PI 선택에서는 PI 선택표를 그리고 카르노 맵과 같이 최소한의 항으로 최적화할 수 있도록 PI를 선택하면 됩니다. 해당 선택표의 행은 Prime Implicant들로 이루어집니다. 또한, 열은 최소항들로 구성됩니다. 그리고, 각 Prime Implicant가 포함하는 최소항을 표시하기 위해 해당 위치에 'X' 표시를 합니다. 이후 필수 prime implicant를 선택하고 이를 수식으로 변환하면 됩니다.

더 변수가 더 많아질 경우에는 Espresso 알고리즘 같은 다른 방식을 사용하는 것이 효율적일 수 있습니다.

5.

용어 정리

implicant : 카르노 맵에서 1의 묶음들을 말하는데, 이때 그 묶음은 크기가 2^n이어야 합니다.

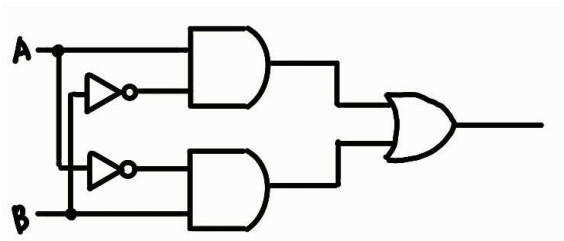
prime implicant : 더 큰 implicant에 포함되지 않는 implicant를 말합니다.

essential prime implicant : prime implicant에 포함되는 '1' 중 하나 이상이 다른 implicant에 속하지 않고 자신의 prime implicant에만 속하는 prime implicant를 말합니다.

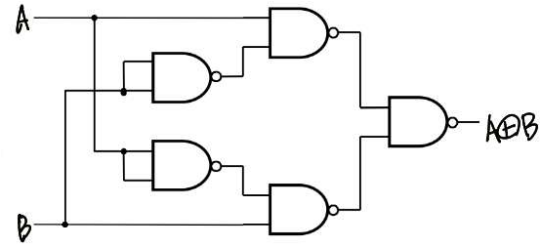
간략화된 함수는 essential prime implicant 모두와 non-essential prime implicant의 일부를 포함합니다.

NAND 최적화

이렇듯 알고리즘을 사용하여 여러 회로를 단순화할 수 있지만, 회로 설계상 nand가 제일 편리하기 때문에 xor gate를 nand로만 구성할 수 있습니다.



이게 기존의 xor gate(f = ab' + a'b)였다면



이런식으로 NAND gate만 사용하여 표현할 수도 있습니다.

Shannon 정리

논리식을 특정 변수 하나를 기준으로 분해하여 함수의 값을 두 가지 경우, 예를 들어 a가 1일 경우 a가 0일 경우로 나누는 방법입니다.

함수 f를 af(a = 1) + a'f(a = 0) 이러한 형태로 나눌 수 있습니다.

이렇듯 한 함수는 두 개의 작은 함수로 분리될 수 있습니다.