5주차 예비보고서

전공 : 아트앤테크놀로지 학년 : 3학년 학번 : 20191098 이름 : 백승주

**1.**

드모르간의 제 1 법칙 :

드모르간의 제 2 법칙 :

드 모르간이라는 수학자의 이름을 딴 드 모르간의 정리는 식의 성질이 만족한다는 것을 증명한 정리이다. 드 모르간의 법칙은 그대로 논리 회로의 Boolean 식에서도 적용할 수 있다. 드모르간의 제 1 법칙은 A+B식의 전체에 NOT을 취한 것은 A의 NOT과 B의 NOT를 곱한 것과 같음을 나타내며, 드모르간의 제 2 법칙은 AxB식의 전체에 NOT을 취한 것은 A의 NOT과 A의 NOT을 합한 것과 같음을 나타낸다.

**2.** 논리 회로를 간소화하기 위해 주요하게 사용되는 방법으로는 논리 회로를 Boolean 식으로 나타낸 다음 그 식을 여러 성질들을 이용해 간소화하는 것이다 . 그 성질들은 앞서 조사한 드모르간의 법칙을 비롯해 다음과 같은 성질들이 있다.

#A + B = B + A , AB = BA 교환법칙

#A + (B + C) = (A + B) + C , A(BC) = (AB)C 결합법칙

#A + A’ =1, A\*A = 0 보수 법칙

#A+1 = 1 ,A\*0 = 0 경ㄱㅖ 법칙

#A + 0 = A ,Aㆍ1 = A 항등원 법칙

#A + A = A , AA’ = 0 등역법칙

#A(B + C) = AB + AC , A + BC = (A + B)(A + C) 분배법칙

#A + AB = A , A(A + B) = A 흡수법칙

#(A + B)’ = A’B’ , (AB)’ = A’ + B’ 드 모르간의 법칙

#AB + AB’ = (A + B)(A + B’) = A 인접 법칙

위의 식들을 활용한 간단한 예를 보자. 예를 들어

X’YC + XY’C + XYC’ + XYC

와 같은 식이 있다고 하자. 이 식을 등역 법칙과 결합 법칙을 이용하면 다음과 같이 나타낼 수 있다.

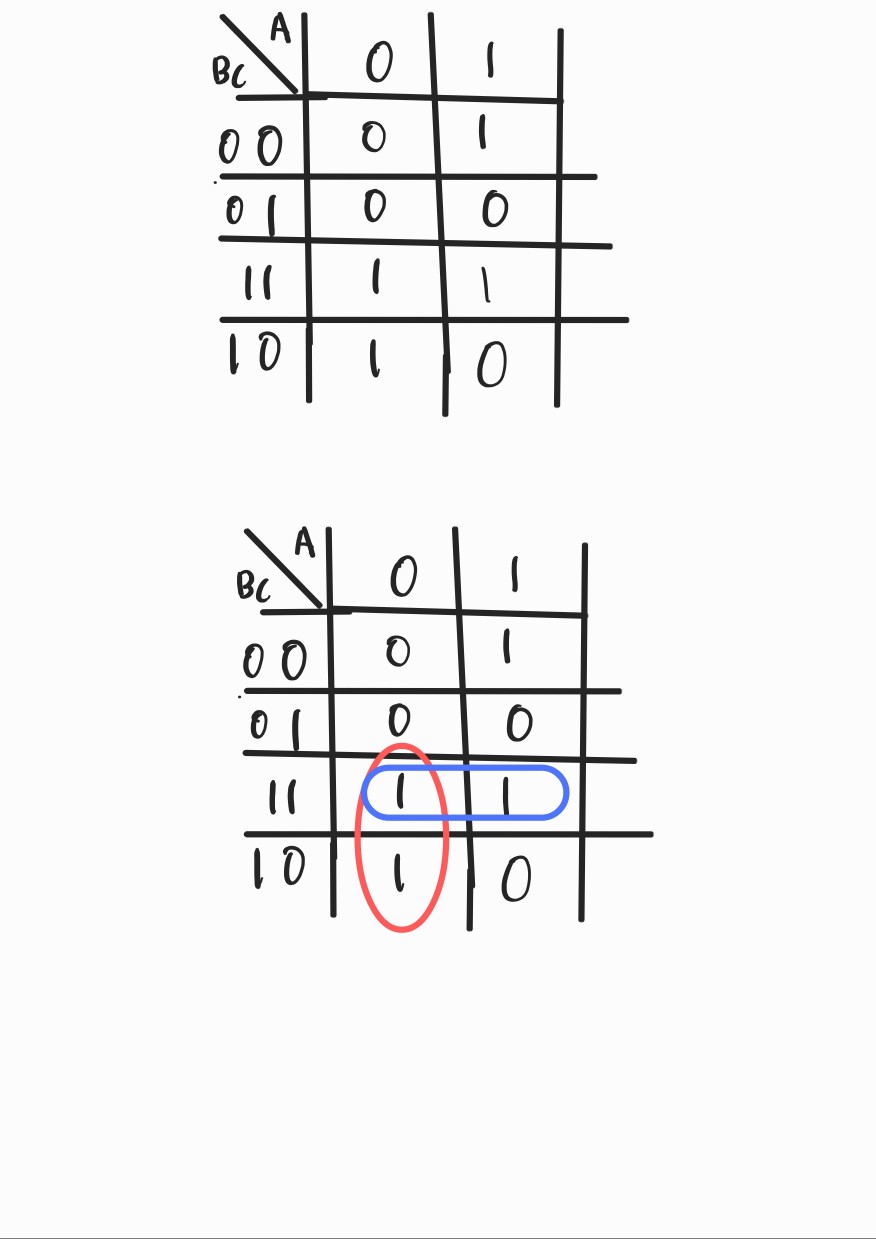
(X’YC + XYC) + (XY’C + XYC) + (XYC’ + XYC)

이후 인접 법칙을 사용해 다음과 같이 간소화시킬 수 있다.

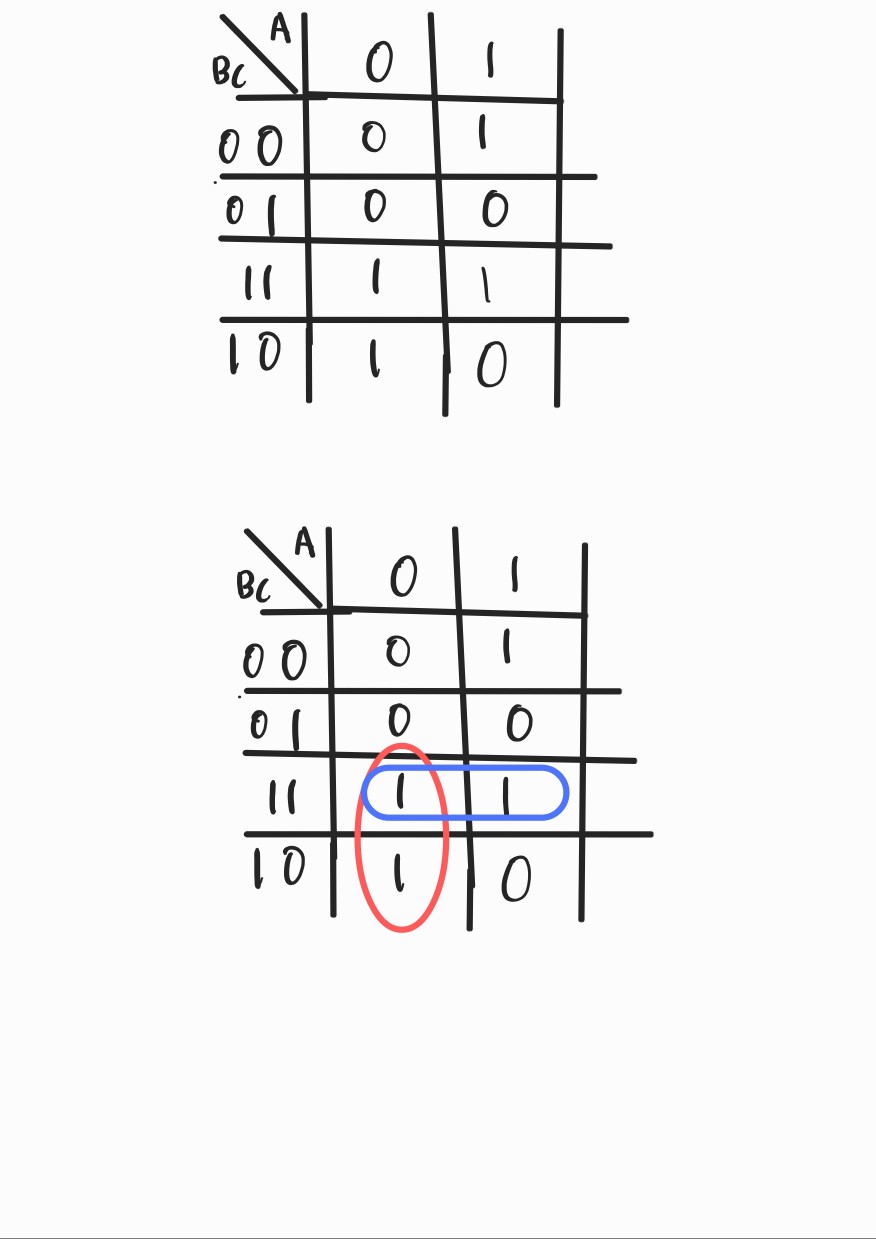
YC + XC + XY

**3.** 카르노맵(Karnaugh Map)이란 Boolean 식을 시각적으로 접근해 보다 쉽게 식을 간소화할 수 있는 하나의 방법이다. 다음의 예시를 통해 카르노 맵을 어떻게 그리고 어떻게 활용할 수 있는지를 알아보자.

**논리식 : A’BC + A’BC’ + AB’C’ + ABC**

****

2-1. 첫째로 카르노 맵을 그려야 한다. 카르노 맵의 열은 각각 순서대로 00, 01, 11, 10을 나타내고, 행은 0, 1을 나타낸다. 카르노 맵의 열은 A의 값을, 각 행은 CD의 값을 나타낸다. 즉 카르노 맵의 크기는 변수의 개수에 따라 2의 n제곱 크기가 된다. 이후 해당 표에 맞게 논리식에 존재하는 항들의 위치에 1 표시를 해주면 된다. 논리식의 항들에 모두 1을 넣고 나머지 항들에 0을 넣으면 위와 같은 표가 완성된다.

****

2-2 . 이후 카르노 맵에서 1인 값들을 2의 N제곱수 만큼씩 묶을 수가 있다. 즉 위의 예시에서는 위와 같이 두 개의 묶음이 나온다. 그리고 이렇게 묶인 항들은 간소화된 논리식으로 나타낼 수 있다. 예를 들어 위 그림에서의 파란색의 묶음은 A’BC+ABC인데 이 식은 인접 법칙을 사용해 BC로 간소화가 가능하다. 마찬가지로 빨간색의 묶음 역시 A’B로 간소화가 가능하다. 즉 예시의 논리식은 카르노맵을 이용해 A’B+BC+AB’C로 간소화했다.

**4.** Quine-McCluskey 알고리즘이란 카르노맵처럼 논리식을 간소화시킬 수 있는 방법 중 하나로 카르노 맵과 거의 비슷하지만 카르노 맵과 비교하여 컴퓨터로 코드를 작성할 때 더 사용이 적합하고 식이 복잡해지면 효율적이다는 장점이 있다.

이 알고리즘을 사용하는 방식은 먼저 논리식의 항들을 나열한 후 이진수로 나타냈을 때 1비트의 차이를 가지는 항들을 묶고 차이가 있는 비트의 자리를 \_로 표시한다. 이 표시는 비트 자리의 변수가 소거되었다는 것을 의미하며 이러한 과정을 1비트 차이의 항들이 없어질 때까지 반복해 minterm 표를 만든다. 표의 minterm을 포함한 항들이 EPI가 되며 EPI가 포함된 항을 모두 포함하며 남은 항들을 모두 묶을 수 있는 최소한의 prime implicant를 찾는다. 이 알고리즘을 통해 변수가 많더라도 효율적인 논리식 간소화가 가능하다.

**5.**

5 -1 Minterm : minterm은 식 안에 들어있는 변수들이 not형태건 상관없이 모두 곱 연산으로 연결된 항을 의미한다. 예를 들어 x,y,z 세 개의 변수를 가질 경우를 생각해보자. 이 때 minterm으로는 xyz, xy’z, xy’z’, x’y’z’… 등이 있다.

5-2 . Maxterm: maxterm은 반대로 식 안의 모든 변수들이 not형태건 상관없이 모두 합 연산으로 연결된 항이다. x,y,z 변수를 이용하여 만든 maxterm으로 x+y+z, x+y’+z, x’y+z’ …있다.

5-3. 특정 논리식을 SOP(Sum of Product)로 표현할 수 있는데 그 중 모든 항이 minterm인 SOP를 canonical sum of product라고 부른다. 그리고 모든 논리식은 이러한 canonical SOP로 표현할 수 있다. 간단히 예를 들어보자.

a + bc’

= a(b + b’)(c + c’) + (a + a’)bc’

= abc + abc’ + ab’c + ab’c’ + abc’

이렇게 나타낸 canonical sop 전체에 NOT을 취하면 드모르간 법칙에 의해 maxterm들을 서로 곱한 canonical product of sum형태로 표현할 수도 있다. 다음은 위의 식에 드 모르간 법칙을 사용해 canonical product of sum 형태로 나타낸 예시다.

(abc + abc’ + ab’c + ab’c + a’b’c)’

= (a’+b’+c’)(a’+b’+c)(a’+b+c’)(a’+b+c)(a+b’+c)