2018년 2학기 컴퓨터공학실험Ⅱ
CSE3016-05반 5주차 예비 보고서

학번: 20171665
이름: 이 선 호

2 0 1 8 . 1 0 . 1 2

목　　차

Ⅰ De Morgan 정리

1. De Morgan 정리

드 모르간의 정리는 논리곱과 논리합, 그리고 부정 연산 간의 관계를 기술하여 정리한 것이며, 공학적으로 특히 논리 회로에서 AND 연산과 OR 연산을 이용하는 논리식 정리에 응용된다. 전자 회로에서 주로 응용하는 드 모르간의 정리는 아래와 같다.

$$A + B = (A' \bullet B')'$$
$$A \bullet B = (A' + B')'$$

이를 증명하는 방법은 $A+B$와 $A' \bullet B'$가 서로 보수(complement) 관계에 있는지 확인해 보면 된다. 어떤 임의의 두 변수가 보수 관계에 있으면 두 변수를 논리곱 연산을 하면 0이 되고 논리합 연산을 하면 1이 나온다. 이를 증명하는 방법은 아래와 같다.

$$\begin{aligned}&(A+B) + A'B' \\ &= (A+B+A')(A+B+B') \\ &= (1+B)(1+A) = 1\end{aligned} \qquad \begin{aligned}&(A+B)A'B' \\ &= AA'B' + BA'B' \\ &= 0B' + 0A' = 0\end{aligned}$$

정리하자면, 둘 이상의 변수가 논리곱 연산으로 연결되어 있으면 이를 논리곱 연산으로 된 식으로 바꾸기 위해 각 변수에 부정을 하여 보수로 바꾼 후 이를 논리합으로 연결하면 된다는 것이다. 논리합 연산을 논리곱 연산으로 바꾸는 과정도 마찬가지이다.

Ⅱ 논리회로의 간소화

1. 논리회로의 간소화

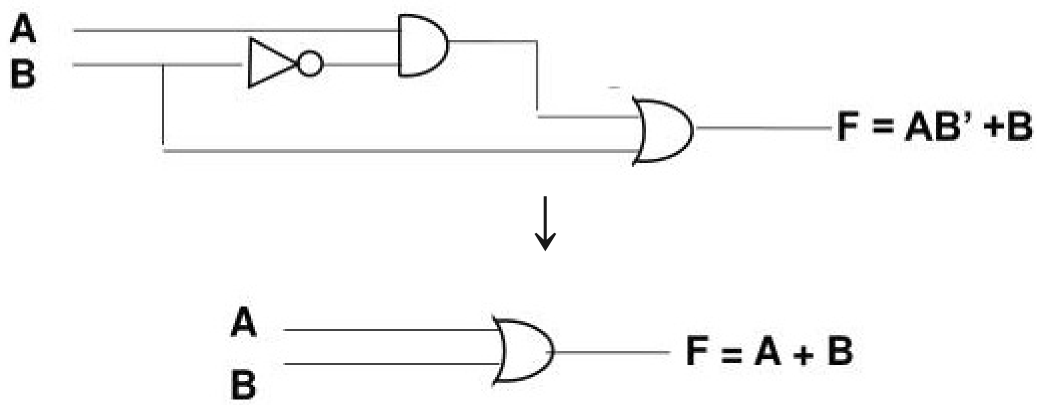
논리 회로에 사용되는 input과 게이트의 개수가 많아질수록 실행 시간이 길어지고 비용이 늘어날 수 있다. 그래서 논리식을 논리회로로 구현할 때 최소한의 input과 게이트로 작업을 해야 할 필요성이 있다. 그래서 논리회로의 논리식을 Switching Algebra를 통해서 간소화할 수 있어야 한다.

논리회로의 논리식을 간소화하는 원칙에는 크게 세 가지가 있다. 첫 번째로는 각 항에서 공통되는 변수를 묶는 것이다. 두 번째로는 분배 법칙을 이용하여 항의 합의 형태를 곱의 형태로 식을 바꾸는 것이다. 마지막으로는 생략되어 있는 항 또는 변수를 추가하여 다른 항들과 공통되는 변수로 묶이도록 식을 바꾸는 것이다.

예를 들어 A와 B를 input으로 받는 회로가 있고, 이를 $AB' + B$와 같은 논리식을 수행하여 결과를 출력하는 논리 회로가 있다고 가정한다. B는 inverter를 사용하여 B'로 바꾸고 이를 A와 AND 게이트로 연산한 결과를 또 다시 B와 OR 게이트로 연산을 하여 최종 값이 나오게 된다. 게이트는 총 3개가 쓰였다. 반면에 위의 논리식을 아래와 같이 간단히 바꿔서 논리회로를 재구성하면 회로의 효율을 높일 수 있다.

$$AB' + B = (B + A)(B + B')$$
$$= (B + A) \bullet 1 = A + B$$

논리식을 간소화했을 때 얻을 수 있는 논리회로의 게이트에서는 OR 게이트의 연산한 번 밖에 일어나지 않고, 사용되는 게이트도 1개로 줄여서 효율이 높아졌음을 알 수 있다. 이와 같은 논리식의 간소화를 통해 논리 회로를 간소화할 수 있다.



Ⅲ 카르노 맵

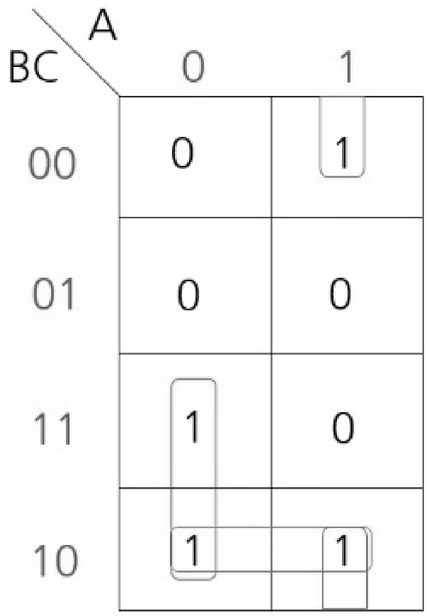
1. 카르노 맵

카르노 맵은 Boolean 값을 input과 output으로 사용하는 논리식의 함수를 단순화하는 방법이며, 논리식을 편리하게 단순화하는 방법을 제공하는 도구이다. 입력 변수와 출력

을 도식화하고, 같은 출력의 패턴을 찾아 묶음으로 단순화해준다.

카르노 맵은 Minimum SOP(Sum of Product)와 POS(Product of Sum) 모두 구하기에 용이하다. 어떤 논리식을 지니는 함수가 있을 때 카르노 맵은 각각의 최소항 또는 최대항에 대하여 한 개의 사각형으로 구성된다. 그래서 2변수 함수 맵에서는 4개의 사각형을 가지고, 3변수 맵은 8개의 사각형, 4변수 맵은 16개의 사각형을 가진다. 이처럼 n개의 input에 대하여 2의 n제곱만큼의 사각형을 가진다. 함수에 포함된 최소항에 대응하는 각 사각형에 1을 위치시키고, 포함되지 않는 사각형은 0을 쓰거나 공백으로 남겨둔다. 카르노 맵을 이용하여 Minimum SOP 또는 POS를 구하는 방법은 3변수 맵을 예시로 들면서 설명할 것이다.

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



3개의 input으로 위와 같은 truth table을 갖는 논리식이 존재한다고 가정한다. 이를 카르노 맵으로 표현하여 그리면 오른쪽의 카르노 맵과 같이 나타난다. 가로는 A의 input의 경우의 수를, 세로는 BC의 input의 경우의 수를 나열한 것이고, 값이 1로 출력되는 경우를 찾아 해당하는 영역을 가로와 세로를 참조하여 1로 채우는 것이다. 그 외에 나머지는 0으로 채운다.

최소화된 SOP를 찾으려면 묶음이 포함하는 사각형의 개수가 2의 거듭제곱 수만큼 되도록 대각선 방향을 제외하여 1로 채워진 사각형을 기준으로 동서남북 방향으로 인접한 1과 직사각형 형태로 묶는다. 이 때 최대한의 직사각형 크기와 최소한의 묶음 개수로 사각형들을 묶어야 나중에 식으로 도출했을 때 따로 최소화할 필요가 없어진다. 이후에 묶음들에 해당하는 input들을 minterm을 기준으로 하여 구하고 구한 항들을 더하여 간소화된 SOP를 얻는다. 최소화된 POS를 찾으려면 SOP와 방법은 마찬가지로 하되 대신 0을 묶어야 하며 묶음의 해당 input들을 maxterm 기준으로 판단하여 항을 구

하고 이를 곱하여 간소화된 POS를 얻어야 한다.
 위의 예제를 카르노 맵을 이용하여 간소화한 SOP를 구하면 $A'B+AC'$가 나온다. 무관항(Don't Care)이라는 것도 존재하는데, 이는 카르노 맵에서 묶는 과정 중 묶음에 포함시켜도 되지만 포함시키지 않아도 무관한 항을 의미한다.

Ⅳ Quine-McCluskey 최소화 알고리즘

1. Quine-McCluskey 최소화 알고리즘

 카르노 맵에서는 변수가 2개, 3개, 4개일 때 부울 연산의 논리식으로 된 함수를 간략화하는 효과적인 방법이지만, 변수가 5개 이상으로 늘어나면 사용하기가 어렵다. 이러한 문제를 쉽게 해결해주는 방법이 바로 퀸-맥클러스키(Quine-McCluskey) 최소화 알고리즘이며, 특히 디지털 컴퓨터에서 손쉽게 프로그램 될 수 있다.
 퀸-맥클러스키 최소화 알고리즘에서는 총 두 단계를 거쳐서 최소 논리곱의 합을 얻을 수 있다. 첫 번째는 분배 법칙을 이용하여 공통되는 변수를 묶는 방법으로, 여기서는 $AB+AB'=A(B+B')=A$의 Switching algebra를 이용하여 변수를 소거한다. 둡번째는 이렇게 소거하여 얻은 주항(Prime Implicant)을 서로 OR연산하여 묶는다. 이러한 단계를 거치면 간략회되는 함수를 얻을 수 있게 된다.
 과정을 더 구체적으로 설명하면 다음과 같다. 예를 들어 변수 4개가 입력되고 minterm의 합으로 된 식을 간략화하고자 하면 항에서 1의 개수가 0개인 항들을 모은 그룹, 1개인 그룹, 2개인 그룹, 3개인 그룹으로 묶을 수 있다. 이러면 그룹이 총 4개가 생긴다는 사실을 알 수 있다. 그 다음에 이웃한 그룹에서 두개의 항이 정확히 한 변수만 다를 때 (즉, 쉽게 말해서 input의 한 자리 값만 서로 차이가 날 때) 결합한다. 이웃하지 않은 그룹 또는 같은 그룹 내의 항들과의 비교는 하지 않고 오로지 이웃한 그룹들 내 각각의 주항들끼리 비교한다. 한 변수만 다른 경우에 해당하는 순서쌍(비교된 항, 비교하고자 한 항)을 기록하고, 그 때 어떤 자리의 변수에서 차이가 나는지를 적기 위해 '-' 표시를 사용한다. 이렇게 해서 얻은 결과는 그룹이 네 개에서 세 개로 줄어들었다는 것을 알 수 있고, 여기서 또 다시 비교 과정을 거쳐서 그룹이 한 개가 될 때까지 중복되는 항을 지워 나가면서 간략화된 함수를 얻을 수 있게 된다. 직접 그룹을 묶어서 나열하지 않고도 퀸-맥컬리스키 방식을 활용한 주항 차트를 이용하여 구할 수도 있다.

Ⅴ 기타 이론

1. 최소항과 최대항

Product term은 1 또는 문자가 한 번을 초과하여 사용되지 않으면서 AND로 연결된 항을 의미한다. 그리고 Sum term은 0 또는 문자가 한 번을 초과하여 사용되지 않으면서 OR로 연결된 항을 의미한다. Minterm은 모든 변수가 항상 한 번씩 사용된 product term을 의미하고, Maxterm은 모든 변수가 항상 한 번식 사용된 sum term을 가리킨다. 어떤 입력값이 주어지고 그 진리표에서 Minterm과 Maxterm을 찾는 작업을 할 때가 자주 있다. 이는 Minterm을 구하면 결과값이 1이 되도록 하는 input을 쉽게 찾을 수 있고, Maxterm을 구하면 결과값이 0이 되도록 하는 input을 쉽게 찾을 수 있기 때문이다. 앞에서 본 카르노 맵이나 퀸–맥컬리스키 알고리즘과 같이 논리 함수를 간략화하는 과정에서 Minterm과 Maxtern의 개념이 필요하다.

Minterm은 1로 출력되는 input을 참고하기 때문에 변수의 값이 1이면 변수 그대로 사용하고, 변수의 값이 0이면 그 변수의 보수 표현을 사용하여 이를 해당하는 입력에 해당하는 문자들과 AND연산으로 묶는다. 반면에 Maxterm은 0을 출력하는 input을 참고하기 때문에 변수의 값이 0이면 변수 그대로 사용하고, 변수의 값이 1이면 그 변수의 보수표현을 사용하여 이를 입력헤 해당하는 문자들과 OR 연산으로 묶는다. 단, 규칙에 맞게 모든 변수는 항상 한 번씩 사용되어야 한다.

Minterm으로 된 논리 함수를 Maxterm으로 된 논리 함수로 바꿀 수 있고, 그 반대도 물론 가능하다. 진리표를 기준으로 같은 번호에 해당하는 Minterm과 Maxterm은 보수 관계에 있기 때문에, 만일 Minterm으로 된 논리 함수를 보수 처리한 식을 구하고 싶으면, Minterm 번호에 대응하는 Maxterm을 찾아서 구한 항들을 AND 연산으로 묶으면 된다. 이 반대도 가능하다.