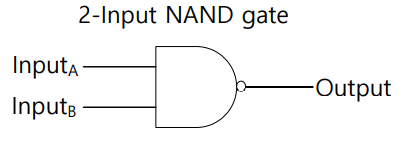
5주차 예비보고서

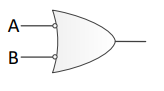
전공: 수학과 학년: 4학년 학번: 20171273 이름: 심현우

1. **De Morgan의 정리에 대해 조사하시오.**

드 모르간의 정리란 NOT을 연산할 때, 간단한 방법으로 계산할 수 있게 해주는 정리이다. 여기서 NOT의 연산이란 두개 이상의 입력 변수가 AND또는 OR의 논리 구조를 가지고 있고 게이트를 통과한 후 NOT의 게이트를 지나는 상황을 의미한다. 이는 Schematic으로 보면 쉽게 알 수 있다. 예를 들어, NAND의 상황을 보자.



4주차에서 구현했던 NAND 게이트이다. 이는 boolean식으로 표현하면 C=~(A&B)이다. schematic에서도 확인할 수 있듯이 게이트 이후에 NOT을 가지고 있는 것을 확인할 수 있다. 이제 드모르간의 정리에 의하여 이를 계산해보자.



두 schematic은 동등한 논리를 가지고 있다. 실제로 계산해보면 같은 truth table을 갖는 것을 확인할 수 있다. 밑의 schematic을 보면 input의 값을 먼저 NOT해준 뒤 OR게이트를 통과하는 것을 알 수 있다. 이는 boolean식으로 보면 다음과 같다. C=~A | ~B. 이렇게 위의 schematic에서 밑으로 바꾸는 과정이 드모르간의 정리를 적용한 것이다. 전체를 포함하는 NOT을 안쪽에 각각 적용하여 A는 ~A로 &는 | 로 B는 ~B로 계산했다. 이는 OR게이트를 부정할 때도 마찬가지로 OR게이트는 NOT연산에 의하여 AND게이트가 된다.

요약하여 정리하면 NOT연산을 적용하면 input에 대해 A -> ~A, ~A -> A, 연산에 대해 & -> |, | ->&로 변환할 수 있다는 것이 드모르간의 정리이다.

1. **논리회로의 간소화에 대해 조사하시오.(예시포함)**

논리회로의 간소화는 입력과 출력이 동일한 경우에 대해서 논리회로를 가장 간단하도록 구성하는 것을 의미한다. 이는 논리 회로의 최적화라고도 불린다. 우리가 논리 회로의 간소화에 집중해야 하는 이유는 논리 회로를 간소화하는 것에 따라 이점이 많기 때문이다. 우선 논리 회로가 간단할수록 소비하는 전력이 감소한다. 또한 동작 속도도 향상된다. 따라서 집적도를 높여 간단한 회로를 구성하는 것은 항상 고려해야할 문제이다.

논리회로 간소화를 구현하는 방법은 두가지가 있다. 하나는 논리 회로 자체를 간소화하는 방법과, 또 하나는 구현하고자 하는 논리회로를 boolean으로 표현한 후 boolean식을 간단하게 만들어 회로로 구현하는 방법이다. 이때, 두번째 방법 사용 시, 위의 조사했던 드모르간의 정리와 다른 간소화를 위한 연산을 이용하여 boolean식을 정리해 간단한 합과 곱의 식을 유도한다. 또는 카르노 맵을 사용하여 간단한 회로를 계산한다.

예를 들어, 다음과 같은 Boolean식을 간소화 해보자. 회로는 Output Y와 input A, B, C, D 4가지로 구성되어 있다. boolean식은 다음과 같다.

Y=AB+~ACD+~ABD+~AC~D+ABCD

이를 간소화하는 과정은 다음과 같다.

Y=AB+ABCD+~ACD+~AC~D+~ABD

=AB+AB(CD)+~AC(D+~D)+~ABD

=AB+~AC+~ABD

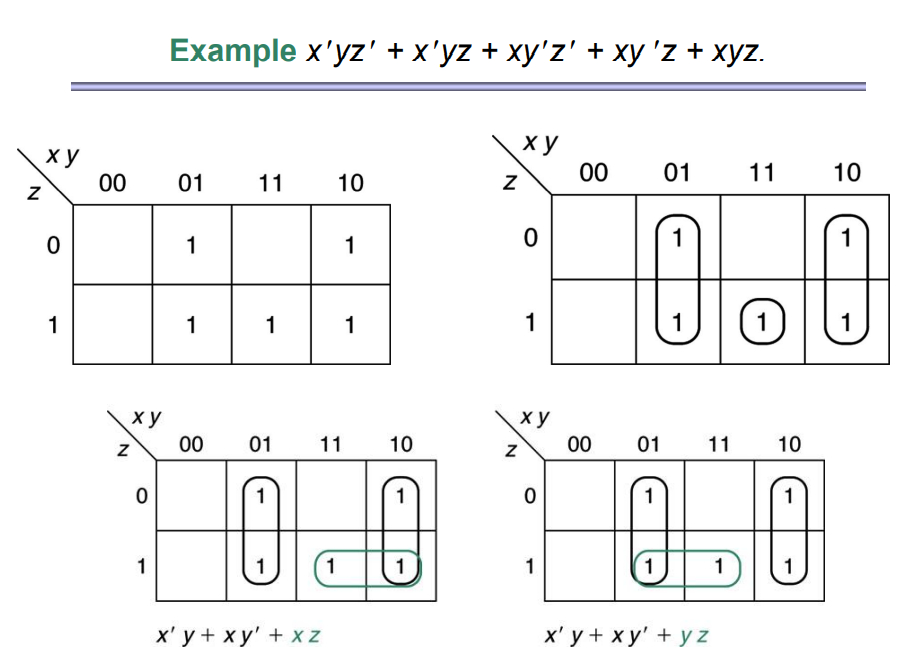
=B(A+~AD)+~AC

=B(A+D)+~AC

여기서 드모르간의 정리, simplification, distributive 등, 간소화를 위한 연산들을 사용하여 마지막, 5개의 input을 사용하는 boolean식을 이끌어냈다.

1. **카르노 맵에 대해 조사하시오.**

카르노 맵은 진리표를 확장된 표로 작성하여 논리회로의 간소화를 달성하기 위해 사용한다. 카르노 맵을 사용하는 과정은 다음과 같다. 우선 논리회로를 boolean식으로 나타낸다. 이 때, 위의 2번의 간소화 예의 과정처럼 boolean식을 간단하게 표현한다. 그 후, 이를 진리표로 나타낸다. 정리했던 boolean 식이 SOP(Sum Of Product)이면 표에서 인접한 항에 1이 최대가 되도록 구성한다. 여기서 SOP는 boolean식을 정리했을 때, product한 input들이 sum으로 표현되어 있는 식을 의미한다. 2번에서의 예시도 SOP라고 할 수 있다. 반대로 POS(Product Of Sum)은 진리표를 구성할 때 최대 인접항이 0이 되도록 구성한다. 인접한 항이 생기도록 구성했으면 이를 묶고 남아있는 변수들로 간소화된 boolean식을 구한다.



예를 들어, ~xy~z+~xyz+x~y~z+x~yz+xyz를 간소화 하는 과정은 다음과 같다. SOP으로 boolean식을 정리하기 위해서, 인접한 항이 1이 되도록 진리표를 구성한다. 이후 인접한 항을 묶고 남아있는 변수들로 boolean식을 구성했다.

1. **Quine-McCluskey 최소화 알고리즘에 대해 조사하시오.**

Quine-McCluskey 간소화 알고리즘 또한 논리 회로의 간소화를 하기 위한 방법이다. 이 알고리즘의 논리 과정 자체는 3번에서 조사했던 카르노 맵과 비슷하다. 하지만 직접 계산해야 하는 카르노 맵과 달리 컴퓨터에서 이 알고리즘 자체를 적용할 수 있어 직접 계산하지 않아도 된다는 이점이 있다. 위의 카르노 맵과 비슷하게 Quine-McCluskey를 적용하기 위해서 우선 minterm을 bit로 표현한 표를 만든다. 이 후, minterm을 다른 minterm과 결합하고 더 이상 결합하지 못할 때, prime table을 만든다. 마지막으로 prime table을 이용하여 간소화된 boolean식을 이끌어낸다.

1. **기타이론.**

Mimterm과 Maxterm에 대해서 알아보자. Minterm이란 SOP를 표현하기 위해 input이 product의 연산으로 구성되어 있는 것을 의미한다. 반대로 Maxterm은 POS을 표현하기 위해 Sum으로 구성되어 있는 term을 product연산으로 boolean식을 구성하는 것을 의미한다. 우선 minterm은 xyz 3개의 input에 따라 이를 각각 0과 1로 값을 준다. 이때 0은 NOT을 의미하도록 한다. 따라서 총 8가지의 경우가 생길 수 있고 이를 NOT과 Product를 사용하여 minterm을 표현할 수 있다.

테이블이(가) 표시된 사진

자동 생성된 설명

Maxterm도 마찬가지로 표현할 수 있다. Maxterm은 1을 NOT으로 하고 Sum연산을 통해 표현할 수 있다.

테이블이(가) 표시된 사진

자동 생성된 설명