5주차 예비보고서

학번 : 20211531

이름 : 나호영

1.

드모르간의 법칙은 AND와 OR 연산을 바꾸고 각 literals에 NOT을 취하는 것이다. 이를 식으로 표현하면 다음과 같다.

(A+B)’ = A’B’

(AB)’ = A’ + B’

드모르간 법칙의 증명 ( 출처 : 김주호 교수님 디지털회로개론 2장 강의 자료 )

Part 1: Show x + y + x’. y’ = 1.

x + y + x’. y’

= (x + y + x’) (x + y + y’) (Distributive Law)

= (x + x’ + y) (x + y + y’) (Commutative Law)

= (1 + y)(x + 1) X + X’ = 1

= 1 . 1 1 + X = 1

= 1 1 . X = 1

Part 2: Show (x + y) . x’. y’ = 0.

(x + y) . x’. y’

= (x . x’. y’ + y . x’. y’) (Distributive Law)

= (x . x’. y’ + y . y’ . x’) (Commutative Law)

= (0 . y’ + 0 . x’) X . X’ = 0

= (0 + 0) 0 . X = 0

= 0 X + 0 = X (With X = 0)

드모르간 법칙을 증명할 때, 중요한 것은 증명 과정 중에 드모르간 법칙을 사용하지 않고 증명하는 것이다. (A+B)’ = A’B’를 증명하기 위해서는 (A+B)와 A’B’가 보수 관계에 있어 위의 Part 1식의 OR 연산에서 1이 도출돼야 한다. 마찬가지로 보수 관계이기 때문에 Part 2 식의 AND 연산에서 0이 도출돼야 한다. 이의 증명 과정은 위와 같다. 드모르간 법칙을 이용하여 표현식에서 많은 literals을 좀 더 적은 literals로 표현할 수 있다.

2.

논리식의 간소화는 term(게이트의 수)를 줄이거나 literal(게이트의 input)을 줄이는 방식으로 진행된다. 적절한 간소화를 진행한다면 논리 회로의 실행 속도를 향상시킬 수 있고 소비 전력 또한 감소시킬 수 있다. 다양한 방식으로 간소화시킬 수 있는데 대표적으로 논리 회로를 boolean 함수로 표현한 뒤, 드모르간 법칙과 같은 boolean 기본 정리를 이용하여 연속적으로 함수를 줄여나가는 방식이다. 예를 들어서 아래와 같이 논리 회로를 간소화시킬 수 있다.

wx’(y’+xz)+wx’(x+y’z)

=wx’y’+wx’xz+wx’x+wx’y’z (분배 법칙)

=wx’y’+wx’y’z (보수 법칙)

=wx’y’(1+z) (경계 법칙)

=wx’y’

또한 카르노 맵 방식을 이용하여 논리 회로를 간소화할 수 있는데 이는 3번, 4번에서 자세히 다루도록 하겠다.

3.

카르노 맵 방식이란 논리식의 변수들의 진리표를 mapping시켜 행렬과 같은 맵으로 표현하는 방식이다. 변수들을 맵의 행과 열로 나타내고 각각에 0과 1을 배치해 0일 때는 해당 변수, 1일 때는 해당 변수의 not값이 할당된다. 카르노 맵에서는 변수들의 개수가 중요한데 예를 들어 가장 많이 사용하는 3변수 카르노 맵은 A와 BC 식으로 행, 열을 구성하고 4변수 카르노 맵은 AB와 CD 식으로 행, 열을 구성한다. 보통 5변수 이상의 카르노맵은 잘 사용되지 않는다. 그 후, 완성 된 맵에서 해당 논리식이 1인 것들을 인접하여 직사각형 형태로 2^n개씩 묶어낸다. 그리고 묶어낸 직사각형들을 불식으로 표현하여 더하면 간소화가 완료된다. 예를 들어서 아래와 같은 방식으로 카르노 맵을 사용한 간소화를 진행할 수 있다.

F = a’b + ab’ + ab식을,

카르노 맵으로 mapping

F의 값이 1에 해당하는 곳에 1을 넣는다.

|  |  |  |
| --- | --- | --- |
| a/b | 0 | 1 |
| 0 | 0 | 1 |
| 1 | 1 | 1 |

그 후 인접한1들을 묶어낸다.

|  |  |  |
| --- | --- | --- |
| a/b | 0 | 1 |
| 0 | 0 | 1 |
| 1 | 1 | 1 |

묶은 부분을 논리식으로 쓴다면 각각 ab’+ab = a, a’b+ab = b이므로 간소화한 논리식은 a+b이다.

4.

카르노 맵은 변수가 적을 때 간소화하기 좋은 방식이지만 변수가 많아진다면 퀸-맥클러스키 알고리즘을 이용한 프로그램으로 간소화하는 것이 좋다. 퀸-맥클러스키 법칙을 이용하기 위해선 주항을 알아야 하는데 주항이란 최대한 결합하여 더 이상 다른 항들과 결합할 수 없는 항을 말한다. 먼저 모든 주항들을 구해야 하는데 이를 위하여 2진수의 최소항들을 1의 개수에 따라 그룹으로 나눈다. 그 후 이웃한 그룹의 항들끼리 비교한다. 이 때 흡수 법칙을 이용하여 변수 한 개만 다른 항들을 서로 결합시키고 해당하는 다른 항은 – 처리, 즉 무관항 처리한다. 결합한 항들을 다시 1의 개수에 따라 그룹으로 나눈 뒤 마찬가지로 변수 한 개만 다른 항들을 서로 결합시킨다. 그 후 이 과정을 가능할 때까지 반복한다. 해당 과정 중에 결합되지 않았던 항들이 주항인데 모든 최소항은 최소한 주항들 중에 하나는 포함되기 때문에 논리식은 주항의 합과 같다. 따라서 주항들의 합으로 논리식을 간소화시킨 결과를 표현 할 수 있다.

5.

++SOP/POS, 간소화 복원

2번의 첫번 째 간소화 방식에 좀 더 자세한 이론을 덧붙이자면, SOP와 POS 개념이 필요하다. SOP란 sum of puoduct의 약자로 논리곱들의 논리합이고 POS는 porduct of sum의 약자로 논리합들의 논리곱이다. 이렇게 나타내는 것을 Canonical Form이라고 한다. 1번의 드모르간 법칙을 이용하여 해당 두 표현을 바꿀 수도 있다. Canonical Form을 이용해 Boolean 함수들을 이용하여 간소화시키는 방식은 사실 효율적이라고 볼 수는 없지만 어떠한 논리식이 원래는 어떻게 구성되어 있는지 minterm으로 보기 위해 복원할 때의 장점이 있다. 아래와 같은 예를 보자.

F = BC + AC + AB

=BC(A’+A) + AC(B’+B) + AB(C’+C)

=A’BC + ABC + AB’C + ABC + ABC’ + ABC

=A’BC + ABC + AB’C + ABC’ ( Σ(3,5,6,7) <-minterm 표현 )

결과적으로 구성하는 literals들을 SOP로 표현하여 논리식의 구조를 명확하게 파악할 수 있다.