5주차 예비보고서

전공: 컴퓨터공학과 학년: 3학년 학번: 20191599 이름: 송경호

**1.**

De Morgan의 정리는 논리합은 논리곱의 부정으로, 그리고 논리곱은 논리합의 부정으로 표현할 수 있다는 법칙이다.

**De Morgan 제 1법칙 :**  혹은 **De Morgan 제 2법칙 :**  혹은

제1법칙의 경우 A, B의 곱을 부정한 결과는 A, B를 각 각 부정한 것을 더한 결과와 서로 같다는 법칙이다. 제2법칙의 경우 A, B의 합을 부정한 결과는 A, B를 각 각 부정한 것을 곱한 결과와 서로 같다는 의미이다.

아래 진리표를 통해 De Morgan 법칙을 확인할 수 있다.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| A | B | A’ | B’ | AB | (AB)’ | A’+B’ |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| A | B | A’ | B’ | A+B | (A+B)’ | A’B’ |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 |

**2.**

입력과 출력이 동일하다는 가정하에 논리회로는 최대한 간소화하는 것이 좋다. 논리회로는 간소화를 통해 회로의 속도를 향상시키고 전력 소비를 감소시킬 수 있다. 논리 회로는 간소화를 위해 Boolean algebra를 근거한 다양한 법칙을 적용시킨다. 다음은 논리회로 간소화에 사용될 수 있는 법칙이다

**2-1) 단일 변수 법칙**

|  |  |  |
| --- | --- | --- |
| Complement law | a+a’=1 | a\*a’=0 |
| Identity law | a+0=a | a\*1=a |
| Null law | a+1=1 | a\*0=a |
| Idempotency law | a+a=a | a\*a=a |
| Involution law | (a’)’=a | |

**2-2) 다변수 법칙**

|  |  |  |
| --- | --- | --- |
| Commutative law | a+b=b+a | ab=ba |
| Associative law | a+(b+c)=(a+b)+c | a(bc)=(ab)c |
| Distributive law | a(b+c)=ab+ac | a+bc=(a+b)(a+c) |
| Adjacency law | ab+ab’=a | (a+b)(a+b’)=a |
| a’b’+a’b+ab+ab’=1 | (a’+b’)(a’+b)(a+b)(a+b’)=0 |
| Simplification law | a+a’b=a+b | a(a’+b)=ab |
| DeMorgan law | (a+b)‘=a’b’ | (ab)‘=a’+b’ |
| (a+b+c…)‘=a’b’c’… | (abc…)‘=a’+b’+c’+… |
| Absorption law | a+ab=a | a(a+b)=a |
| Consensus law | ab+a’c+bc=ab+a’c | (a+b)(a’+c)(b+c)=(a+b)(a’+c) |
| ab+a’c=(a+c)(a’+b) | |

**2-3) 예시**

논리식 간소화의 예시이다.

A’B + ABC’ + ABC

**= Distributive Law =>** A’B+AB(C + C’)

**= Complement Law =>** A’B + AB

**= Distributive Law =>** (A + A’)B

**= Complement Law =>** B

**3.**

**3-1) 카르노맵이란?**

카르노맵(Karnaugh Map)이란 디지털 논리 회로 설계와 최적화에 사용되는 도표이다. 논리 함수를 시각적으로 표현할 수 있어 최소화를 보다 쉽고 편리하게 할 수 있다는 장점이 있다. 진리표에서 하나의 항이 카르노 맵에서는 한 칸으로 나타나며 각 칸은 서로 다른 논리 상태를 나타낸다. 다음은 카르노 맵의 예시이다.

텍스트, 스크린샷, 소프트웨어, 디스플레이이(가) 표시된 사진

자동 생성된 설명

여기서 주목해야 할 점은 yz의 항을 00, 01, 10, 11이 아닌 00, 01, 11, 10으로 썼다는 점인데, 이를 통해 각 칸의 인접한 칸들은 서로 한 변수의 논리 상태만 다르게 된다. 이를 이용하여 출력값 1인 칸들을 2의 거듭제곱 단위의 implicant로 묶어 논리식을 간소화한다.

스크린샷, 라인, 도표, 번호이(가) 표시된 사진

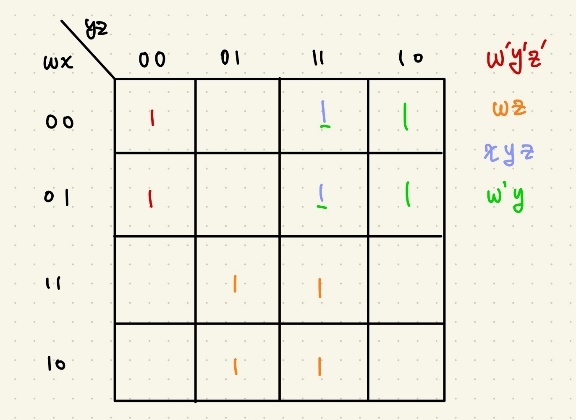
자동 생성된 설명

예를 들어, 위와 같은 3변수 카르노맵에 4x1과 1x2크기의 implicant로 출력값이 1인 칸들을 묶었다. 주의해야 할 점은 출력값이 1인 모든 칸을 묶어야 하며 함수의 간소화를 위해 최대한 큰 크기의 implicant로 칸들을 묶어야 한다는 점이다. 즉, 모든 출력값이 1인 칸은 Prime Implicant 혹은 Essential Prime Implicant에 속해야 한다. 이때, Prime Implicant란 더 큰 implicant에 포함되지 않는 묶음을 의미하며, Essential Prime Implicant란 Prime implicant를 형성하고 있는 칸들 중 적어도 하나가 다른 어떤 implicant에 속하지 않고 오직 해당 prime implicant에만 속하는 것을 말한다.

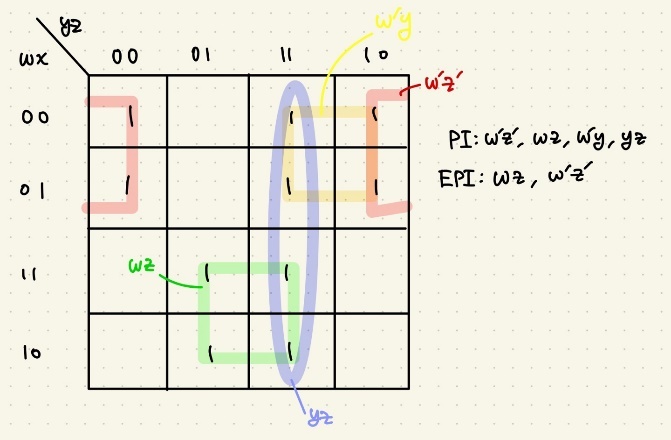
**3-2) 예시**

F = w’y’z’ + wz + xyz + w’y

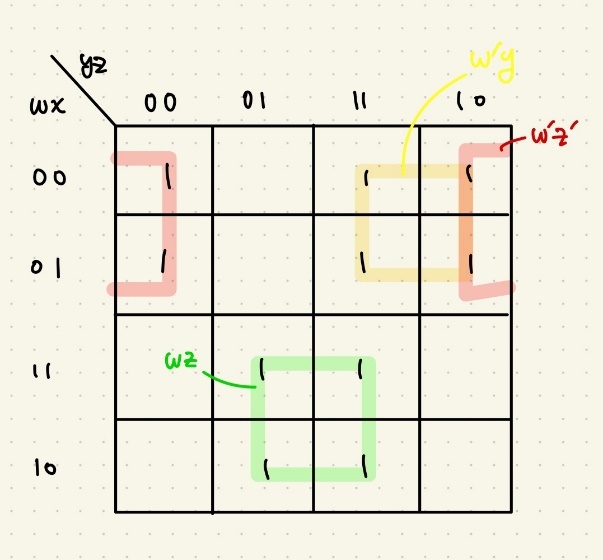
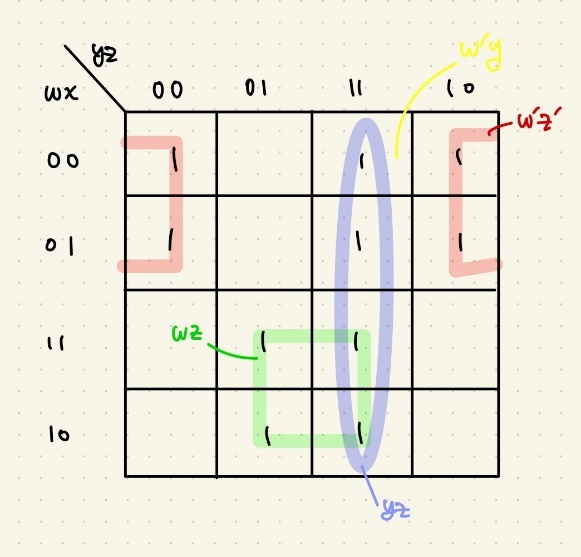
1. 논리 함수를 카르노맵으로 나타내고, 출력값을 배치한다.



2. PI와 EPI를 찾는다.



3. 출력이 1인 칸이 PI와 EPI에 포함되는 조합을 찾는다. 이때 EPI는 반드시 포함되게 된다.

4. 카르노맵을 이용하여 논리식을 간소화한다.

1. F = w’z’ + wz + w’y

2. F = w’z’ + wz + yz

**4.**

Quine-McCluskey 알고리즘도 카르노 맵과 동일하게 Boolean Algebra를 최적화하는 알고리즘이다. 그러나 카르노맵과 달리 표를 사용한다. 이러한 특성 때문에 프로그래밍 하기에 적합하고, 변수가 많아지면 복잡해지는 카르노맵과 다르게 효율적으로 실행할 수 있다. Quine-McCluskey 알고리즘은 예시와 함께 설명하겠다.

f(w, x, y, z) = ∑m(1,4,6,7,8,9,10,11,15)

1) 2진수 값 변환 테이블을 만든다. 이때 1의 개수에 따라 section을 나누어준다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | w | x | y | z |
| 1 | 0 | 0 | 0 | 1 |
| 4 | 0 | 1 | 0 | 0 |
| 8 | 1 | 0 | 0 | 0 |
| 6 | 0 | 1 | 1 | 0 |
| 9 | 1 | 0 | 0 | 1 |
| 10 | 1 | 0 | 1 | 0 |
| 7 | 0 | 1 | 1 | 1 |
| 11 | 1 | 0 | 1 | 1 |
| 15 | 1 | 1 | 1 | 1 |

2) 이웃한 section과 비교하여 1이 차이가 나는 경우 결합한다. 이 때 결합된 값은 don’t care bit(-)와 care bit(0 or 1)로 표시한다. 또한 결합이 된 항목에 체크를 해놓는다.

텍스트, 번호, 평행이(가) 표시된 사진

자동 생성된 설명 텍스트, 번호, 평행, 낱말맞추기 퍼즐이(가) 표시된 사진

자동 생성된 설명

3) 2)의 과정을 결합이 없을 때까지 반복한다. 이때, 두 항목을 결합하기 위해서는 care bit가 1이 차이가 나고 don’t care bit는 일치해야 한다.

텍스트, 번호, 평행이(가) 표시된 사진

자동 생성된 설명 라인, 스크린샷, 번호, 텍스트이(가) 표시된 사진

자동 생성된 설명

4) PI를 결정한다. 2)와 3)의 과정에서 결합이 한번도 이루어 지지 않은 체크 되지 않은 항목을 고른다.

텍스트, 번호, 낱말맞추기 퍼즐, 달력이(가) 표시된 사진

자동 생성된 설명

5) 카르노맵과 같이 어떤 PI를 사용할지 결정한다. 이번에도 역시 EPI를 먼저 구하고 다음으로 PI를 결정한다. 이를 위해 PI 선택표를 그려준다.

스크린샷, 텍스트, 라인, 도표이(가) 표시된 사진

자동 생성된 설명

다음과 같이 EPI를 선택한다.

텍스트, 스크린샷, 도표, 라인이(가) 표시된 사진

자동 생성된 설명

남은 PI에 대한 표를 그린다.

라인, 도표, 텍스트, 폰트이(가) 표시된 사진

자동 생성된 설명

6) 수식에 반영한다.

EPI : (1,9) -> x’y’z | (4,6) -> w’xz’ | (8, 9, 10, 11) -> wx’ / PI : (7,15) -> xyz

F = x’y’z + w’xz’ + wx + xyz

**5.**

McCluskey 최소화 알고리즘처럼 Don’t care bit은 출력이 완전히 정의되지 않은 함수에 사용된다. 어떠한 함수의 출력 값이 해당 함수의 그 어떤 입력값과도 무관한 경우 Don’t care bit으로 나타낸다. 이는 카르노맵에서 역시 사용 가능하며 카르노맵 칸에 X를 적어 Don’t care을 표시한다.

또한 어떤 논리식을 최소화 하는 형태는 반드시 하나일 필요는 없다. EPI를 모두 포함하고 남은 부분들을 PI로 포함하고 있는 형태이기만 하면 상관없다.