

실험 Verilog-4: 예비보고서

전공: 컴퓨터공학

학년: 2

학번: 20171645

이름 박찬우

1. 목적

드모르간의 정리와 논리회로의 간소화 등의 이론에 대해 이해한다.

2. 요구 사항

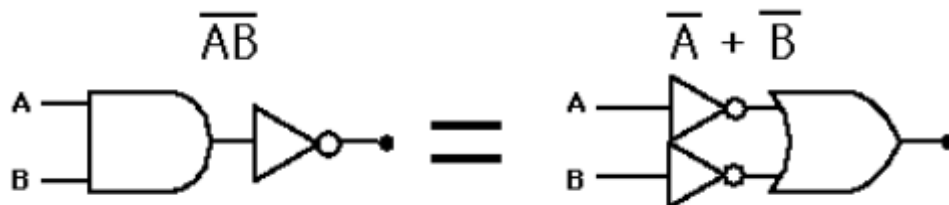
드모르간의 정리에 대해 조사한다.

드모르간의 정리는 두개의 정리로 구성되어있다.

드모르간의 제1정리 : 변수들의 곱의 보수는 각각의 변수에 보수를 취해 합친것과 같다.
이를 수식으로 나타내면 다음과 같다.

$$\text{not}(A \text{ and } B) = (\text{not } A) \text{ or } (\text{not } B)$$

그리고 이걸 전자회로로 나타내면 다음과 같다.



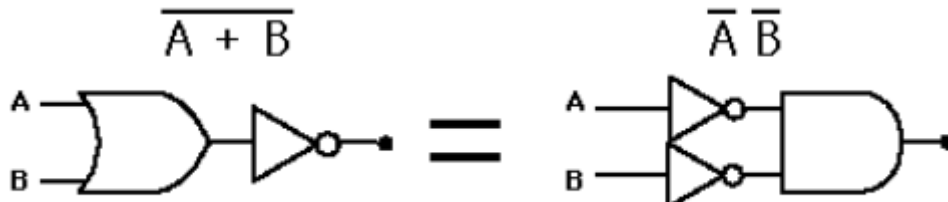
A NAND gate is equivalent to an inversion followed by an OR

즉, 드모르간의 제1법칙은 NAND gate = Negative OR gate를 의미한다.

드모르간의 제2정리 : 변수들의 합의 보수는 각각의 변수에 보수를 취해 곱한것과 같다.
이를 수식으로 나타내면 다음과 같다.

$$\text{not}(A \text{ or } B) = (\text{not } A) \text{ and } (\text{not } B)$$

그리고 이걸 전자회로로 나타내면 다음과 같다.



A NOR gate is equivalent to an inversion followed by an AND

즉, 드모르간의 제2법칙은 NOR gate = Negative AND gate를 의미한다.

논리회로의 간소화에 대해 조사한다.

논리회로를 간소화하기 위하여 불대수 논리식을 이용한다.
불대수의 기본공식은 다음과 같다.

1. $A + A = A$
2. $AA = A$
3. $A + A' = 1$
4. $AA' = 0$
5. $1 + A = 1$
6. $1A = A$
7. $0A = 0$
8. $(A')' = A$

위 내용을 일반 법칙이라 하고, 이 외에도 교환법칙, 분배법칙, 결합법칙, 드모르간의 법칙이 성립하는데 각각은 다음과 같다.

교환법칙 : $A+B = B+A$

결합법칙 : $A+A'B = (A+A')(A+B) = A+B$

드모르간의 법칙은 위의 설명 참고

불대수로 논리회로를 간소화하는 예시는 다음과 같다.

$$\begin{aligned} F &= xy'z + x'y'z + xyz \\ &= xz(y + y') + x'y'z \\ &= z(x + x'y') \\ &= z(x + y') \end{aligned}$$

카르노 맵에 대해 조사한다.

카르노 맵은 불 대수 위의 함수를 단순화 하는 방법으로, 사람의 패턴인식으로 연관된 상호관계를 이용해 확장된 논리표현을 줄이는 방법이라고 할 수 있다.

그 방법은 다음 과정을 따른다.

1. 입력과 출력의 관계를 진리표의 형태로 정리한다.
2. 1이라고 표시된 부분을 2^n 개씩 중복해서 묶는다.
3. 묶음은 곱으로, 묶음과 묶음은 합으로 표현한다.

변수가 2개인 카르노맵, 3개인 카르노맵의 예시는 각각 다음과 같다.

A \ B	0	1
0	1	
1	1	1

$$A' \cdot B' + A \cdot B' + AB \longrightarrow B' + A$$

A \ BC	00	01	11	10
0	1			1
1	1	1	1	1

$$A'B'C' + AB'C' + AB'C + ABC + ABC' + A'BC' \longrightarrow C' + A$$

Quine-McCluskey 최소화 알고리즘에 대해 조사한다.

앞서 언급한 카르노맵은 변수가 5개가 넘어가면 적용하기 매우 어려워지므로, 이때는 Quine-McCluskey 알고리즘을 이용해야 한다. 이는 다음 두 과정을 거친다.

1. $XY + XY' = X$ 형태의 식을 기본으로 하여, 가능한 많은 변수를 줄이도록 한다.
2. 나온 결과 표를 이용해서 최소한의 결과만으로 이루어진 minimum set을 구한다.

예를들어, 다음과 같은 경우를 생각한다.

$$f(a, b, c, d) = \sum m(0, 1, 2, 5, 6, 7, 8, 9, 10, 14)$$

minterm을 2진수로 표현했을 때의 1의 개수로 그룹을 나누면 다음과 같다.

group 0	0	0000
group 1	1	0001
	2	0010
	8	1000
group 2	5	0101
	6	0110
	9	1001
	10	1010
group 3	7	0111
	14	1110

인접한 그룹끼리 차례로 묶어 column 1을 완성하고, column 2부터는 다음 규칙을 따른다.

1. 차례대로 인접한 그룹 두개를 선택해 각 그룹의 원소 중 하나씩 고른다.
2. 중복 숫자 또는 - 가 3개면 겹치지 않는 숫자를 -으로 바꾸고, 겹치는 숫자는 그대로 표시해 새로운 column에 쓴다.
3. 가능한 모든 쌍을 찾아 반복한다. 이 쌍들을 하나의 group으로 한다. 이때, 마찬가지로 체크 표시를 한다.
4. 그룹 숫자가 가능한 적어질 때 까지 새로 생성된 group을 바탕으로 반복해준다.

이러한 과정을 통해, 아래와 같은 결과를 얻을 수 있다.

	Column I	Column II	Column III
group 0	0 0000 ✓	0, 1 000- ✓	0, 1, 8, 9 -00-
group 1	1 0001 ✓	0, 2 00-0 ✓	0, 2, 8, 10 -0-0
	2 0010 ✓	0, 8 -000 ✓	0, 8, 1, 9 -00-
	8 1000 ✓	1, 5 0-01	0, 8, 2, 10 -0-0
group 2	5 0101 ✓	1, 9 -001 ✓	2, 6, 10, 14 --10
	6 0110 ✓	2, 6 0-10 ✓	2, 10, 6, 14 --10
	9 1001 ✓	2, 10 -010 ✓	
	10 1010 ✓	8, 9 100- ✓	
group 3	7 0111 ✓	8, 10 10-0 ✓	
	14 1110 ✓	5, 7 01-1	
		6, 7 011-	
		6, 14 -110 ✓	
		10, 14 1-10 ✓	