**컴퓨터공학 설계 및 실험Ⅱ**

5주차 예비보고서

서강대학교 공학부 컴퓨터공학 전공

20171646 박태윤

**1. De Morgan의 정리에 대해 조사하시오.**

영국의 수학자 Augustus De Morgan의 이름에서 따온 법칙으로 교집합(논리곱)이 합집합(논리합)으로 바뀌어 쓰여지거나 그 역을 나타내는 법칙이다. 집합론 혹은 수학 논리학 같은 분야에서 또는 과 같이 쓰이며 이는 논리회로에서도 논리곱과 논리합의 관계에서 혹은 와 같이 쓰인다. 드모르간의 법칙은 A + B의 complement(여집합)가 A’ \* B’이라면 A + B의 부정인 (A + B)’와 A’ \* B’가 같게 되므로 성립한다.즉, A + B의 complement가 A’ \* B’임을 증명하면 드모르간의 법칙을 증명할 수 있는데, 두 식의 논리합이 1이고 논리곱이 0이라면 complement한 관계를 증명할 수 있다.

- A + B + A’ \* B’ = 1

A + B + A’ \* B’

= (A + B + A’) \* (A + B + B’)

= (1 + B) \* (A + 1)

= 1 \* 1

= 1

- (A + B) \* A’ \* B’ = 0

(A + B) \* A’ \* B’

= (A \* A’ \* B’ + B \* A’ \* B’)

= (0 \* B’ + 0 \* A’)

= (0 + 0)

= 0

**2. 논리회로의 간소화에 대해 조사하시오(예시 포함).**

해당 논리회로를 나타내는 boolean식을 다양한 법칙을 통해 간소화를 시킬 수 있는데, 논리회로를 간소화 시키면 소비 전력을 감소시키거나 동작 속도를 향상시킬 수 있는 등의 장점을 지닌다.

|  |  |
| --- | --- |
| 교환 법칙 | A+B = B+A |
| A\*B = B\*A |
| 결합 법칙 | (A+B)+C = A+(B+C) |
| (A\*B)\*C = A\*(B\*C) |
| 분배 법칙 | A\*(B+C) = A\*B+A\*C |
| A+(B\*C) = (A+B)(A+C) |
| 동치 법칙 | A+A = A |
| A\*A = A |
| 흡수 법칙 | A+A\*B = A |
| A\*(A+B) = A |
| 부정 법칙 | (A’)’ = A |
| 보수 법칙 | A+A’ = 1 |
| A\*A’ = 0 |
| 드 모르간의 정리 | (A+B)’ = A’\*B’ |
| (AB)’ = A’+B’ |

다음과 같은 Boolean 식이 있다고 하자.

F = xy’z + x’y’z + xyz

위의 식을 간소화 시키면

F = xy’z + x’y’z + xyz

= xz(y + y’) + x’y’z (분배법칙)

= xz + x’y’z (보수법칙)

= z(x + x’y’) (분배법칙)

= z(x + x’)(x + y’) (분배법칙)

= z(x + y’) (보수법칙)

= xz + y’z (분배법칙)

가 된다. 실제로 두 식에 관하여 진리표를 작성하면 다음과 같이 똑같은 진리표를 만들 수 있다.

|  |  |  |  |
| --- | --- | --- | --- |
| x | y | z | F |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

최종적으로 다음과 같이 논리 회로를 간단하게 만들 수 있다.

|  |  |
| --- | --- |
| (before) | (after)  텍스트이(가) 표시된 사진  자동 생성된 설명 |

**3. 카르노 맵에 대해 조사하시오(예시 포함).**

카르노맵은 복잡한 논리식을 좀 더 직관적이고 간단하게 간소화 시킬 수 있는 방법이다. 카르노맵을 만들기 위해서는 우선 변수의 개수를 파악한 후 2^n(n = 변수 개수)만큼의 테이블을 만들어 준다. 다음에는 변수의 조합이 0인지 1인지에 따라 값을 채워주고 묶을 수 있는 규칙에 따라 묶어준 뒤 묶어진 값을 간소하게 표현을 한다. 예시를 보면 다음과 같다.

F = A’B’C’ + A’BC’ + AB’C’ + AB’C + ABC’ + ABC

이 식을 카르노맵으로 표현하면 다음과 같다.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | B’C’ | B’C | BC | BC’ |
|  |  | 0 0 | 0 1 | 1 1 | 1 0 |
| A’ | 0 | 1 | 0 | 0 | 1 |
| A | 1 | 1 | 1 | 1 | 1 |

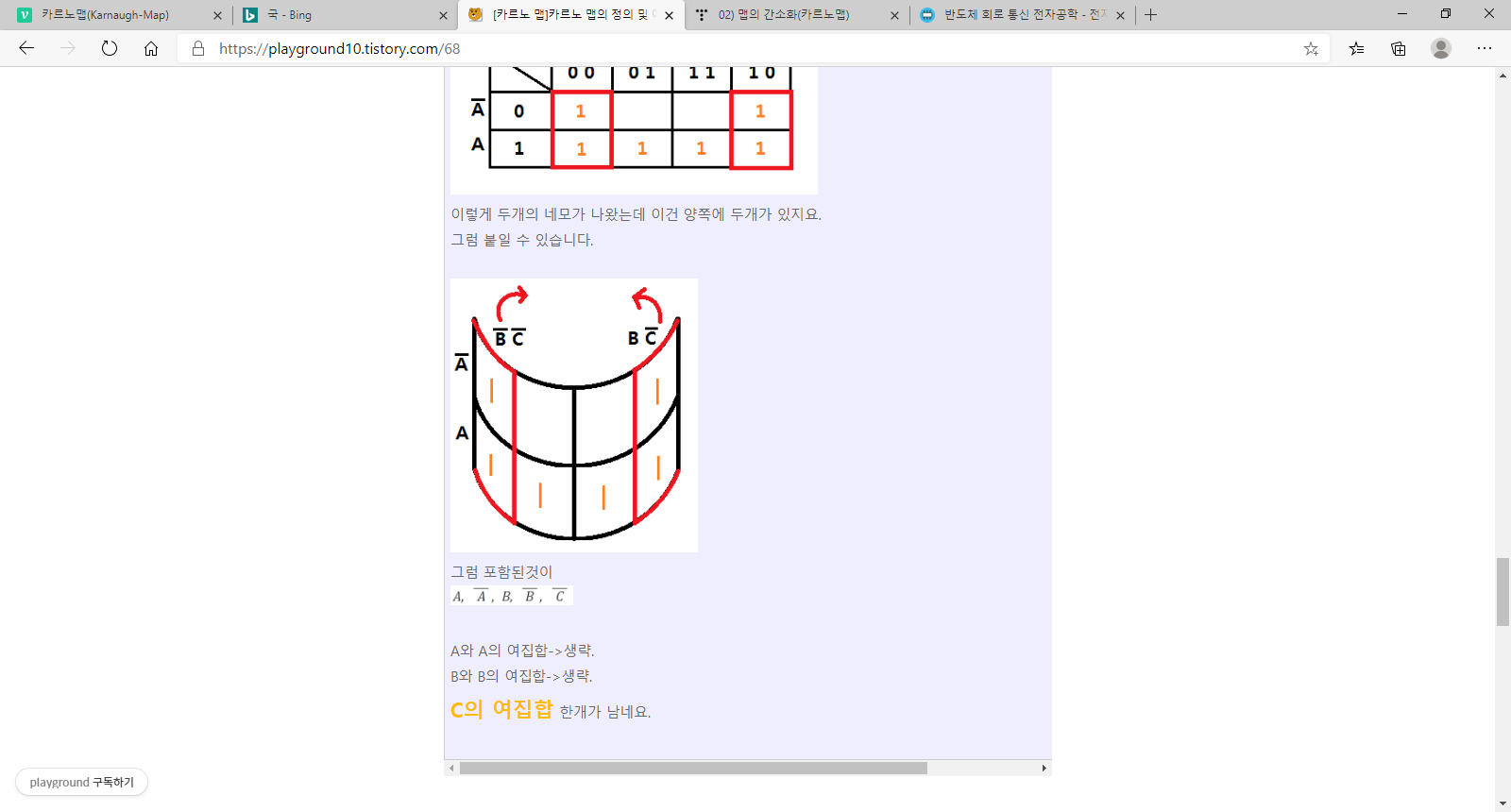
여기서 1은 그 항을 가지고 있는지 아닌지를 보여주는데, 예를 들어 F에서 A’B’C’항이 있으므로 노란색으로 표시된 곳에는 1을, A’B’C항은 가지고 있지 않으므로 초록색으로 표시된 곳에는 0을 써주는 것이다. 이때, 이렇게 3변수 이상의 카르노 맵에서 위의 BC와 같이 두 변수를 묶어서 표현해 주는 부분은 (00, 01, 10, 11)이 아닌 (00, 01, 11, 10)으로 표현을 해 주어야 한다.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | B’C’ | B’C | BC | BC’ |
|  |  | 0 0 | 0 1 | 1 1 | 1 0 |
| A’ | 0 | 1 | 0 | 0 | 1 |
| A | 1 | 1 | 1 | 1 | 1 |

표에서 인접하게 1이 연결된 부분만을 골라서 표시하였다. 우선 빨간색 동그라미로 표시된 부분을 보면 A라면 B’C’, B’C, BC, BC’모두 존재하는 것을 확인할 수 있다. 이는 A로 간소화 시킬 수 있다는 것을 의미하는데 실제로 boolean식으로 확인을 해보면

AB’C’ + AB’C + ABC + ABC’ = AB’(C’ + C) + AB(C’ + C) = AB’ + AB = A

인 것을 확인할 수 있다. 이는 빨간색 원 안에는 A와 B,B’ 그리고 C,C’가 포함되어 있기 때문에 B\*B’ = 0, C\*C’ = 0을 해서 A만 남는 것으로 생각할 수도 있다. 다음에 초록색 원으로 표시된 곳을 보자. 두 군데로 표시된 것처럼 보이지만 양쪽 끝에 두개가 있는 것으로



이렇게 붙여서 생각할 수 있다. 이 영역에 포함된 것은 A,A’ 그리고 B,B’ 그리고 C’이기 때문에 C’으로 간소화 할 수 있고 따라서 F는 최종적으로 A + C’으로 간소화 할 수 있다.

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

퀸 맥클러스키(Quine – McCluskey) 최소화 알고리즘 역시 카르노 맵처럼 논리 회로를 간소화 시킬 수 있는 방법 중 하나이다. 4개의 변수를 가지는 회로까지는 카르노 맵이 퀸 맥클러스키 알고리즘보다 간단함을 보이지만 그 이상에서는 퀸 맥클러스키 알고리즘을 통해 더 쉽게 간소화 시킬 수 있다. 퀸 맥클러스키 알고리즘은 크게 두 단계로 이루어지는데, 첫번째는 인덱스별로 구분하고 AB + AB’ = A를 적용하여 가능한 변수들을 제거한 후 결과 항들을 주항으로 만들어 주는 것이다. 두 번째는 주항 차트를 이용하여 주항의 집합을 구하는 것이다. 예를 들면

F = a’b’c’d’ + a’b’c’d + a’b’cd’ + a’bc’d + a’bcd’ + a’bcd + ab’c’d’ + ab’c’d + ab’cd’ + abcd’라는 식이 있다고 가정해보자.

이를 F(a,b,c,d) = (0,1,2,5,6,7,8,9,10,14)로 표현을 하는데 이는 a,b,c,d순서대로 이진수로 표현하여 F(a,b,c,d) = (0000, 0001, 0010, 0101, 0110, 0111, 1000, 1001, 1010, 1110)과 같다. 이를 다음과 같은 표로 표현할 수 있다.

(column 1)

|  |  |  |  |
| --- | --- | --- | --- |
| 그룹0 | 0 | 0000 | V |
| 그룹1 | 1 | 0001 | V |
| 2 | 0010 | V |
| 8 | 1000 | V |
| 그룹2 | 5 | 0101 | V |
| 6 | 0110 | V |
| 9 | 1001 | V |
| 10 | 1010 | V |
| 그룹3 | 7 | 0111 | V |
| 14 | 1110 | V |

1의 개수에 따라 그룹으로 묶었다. 그 전에 한 가지 살펴볼 규칙이 있다.

AB’CD’ + AB’CD = AB’C( )

1010 + 1011 = 101-

위의 식은 앞의 세 숫자 101(AB’C)이 같고 뒤의 숫자만 다르기 때문에 더해서 101-과같이 표시를 할 수 있다. 위의 방법을 앞서 작성한 표에 적용한다. 즉, 4개의 숫자 중 3개의 숫자가 같고 1개의 숫자만 다른 경우 위의 규칙을 활용하여 묶는다.

(column 2)

|  |  |  |  |
| --- | --- | --- | --- |
| 그룹0 | 0, 1 | 000- | V |
| 0, 2 | 00-0 | V |
| 0, 8 | -000 | V |
| 그룹1 | 1, 5 | 0-01 |  |
| 1, 9 | -001 | V |
| 2, 6 | 0-10 | V |
| 2, 10 | -010 | V |
| 8, 9 | 100- | V |
| 8, 10 | 10-0 | V |
| 그룹2 | 5, 7 | 01-1 |  |
| 6, 7 | 011- |  |
| 6, 14 | -110 | V |
| 10, 14 | 1-10 | V |

(column 3)

|  |  |  |  |
| --- | --- | --- | --- |
| 그룹0 | 0,1,8,9 | -00- |  |
| 0,2,8,10 | -0-0 |  |
| 0,8,1,9 | -00- |  |
| 0,8,2,10 | -0-0 |  |
| 그룹1 | 2,6,10,14 | --10 |  |
| 2,10,6,14 | --10 |  |

여기서 V는, 다른 항과 결합할 수 있는 최소항을 표시한 것이다. Column 3까지 해서 간소화가 끝이 났는데, 마지막으로 주항의 집합을 구해주어야 한다. Column 3에서 뽑을 수 있는 주항은 (0,1,8,9)와 (0,8,1,9)가 같기 때문에 ‘-00-‘, (0,2,8,10)과 (0,8,2,10)이 같기 때문에 ‘-0-0’, (2,6,10,14)와 (2,10,6,14)가 같기 때문에 ‘—10’총 3개이고 column 2에서 다른 항과 결합하지 못한 항 (1, 5), (5, 7), (6, 7) 3개, 총 주항은 6개이고 이를 식으로 표현하면

F = (-00-) + (-0-0) + (--10) + (0-01) + (01-1) + (011-) = b’c’ + b’d’ + cd’ + a’c’d + a’bd + a’bc이다. 이 때, 여기서 boolean식을 통해 간소화를 시킬 수도 있지만 주항차트를 이용하여 추가적으로 간소화를 시킬 수도 있는데 이는 다음과 같다.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 5 | 6 | 7 | 8 | 9 | 10 | 14 |
| b’c’ | X | X |  |  |  |  | X | X |  |  |
| b’d’ | X |  | X |  |  |  | X |  | X |  |
| cd’ |  |  | X |  | X |  |  |  | X | X |
| a’c’d |  | X |  | X |  |  |  |  |  |  |
| a’bd |  |  |  | X |  | X |  |  |  |  |
| a’bc |  |  |  |  | X | X |  |  |  |  |

위와 같이 표를 만들 수 있는데 이 때 빨간색으로 표시된 X는 어떤 열에 X가 하나만 있을 때를 의미하고 그 X를 포함하는 항을 필수주항이라고 한다. 여기서 필수주항은 b’c’, cd’가 되는 것이다. 이후 필수주항이 포함하는 모든 행과 열을 삭제를 할 수 있다.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 5 | 6 | 7 | 8 | 9 | 10 | 14 |
| b’c’ | X | X |  |  |  |  | X | X |  |  |
| b’d’ | X |  | X |  |  |  | X |  | X |  |
| cd’ |  |  | X |  | X |  |  |  | X | X |
| a’c’d |  | X |  | X |  |  |  |  |  |  |
| a’bd |  |  |  | X |  | X |  |  |  |  |
| a’bc |  |  |  |  | X | X |  |  |  |  |

이렇게 하면 최종적으로 온전히 남는 주항은 a’bd밖에 없기 때문에

F = b’c’ + cd’ + a’bd가 된다.

**5. 기타이론.**

어떠한 boolean식도 각 변수들의 합이나 곱으로 나타낼 수 있고, 이렇게 나타낸 식을 Canonical Form(정규형)이라고 한다. 각 변수들의 곱으로 나타내는 minterm, 각 변수들의 합으로 나타내는 maxterm 두 가지가 존재한다. 3종류의 변수를 가지는 식을 예로 들면

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| x y z | minterm | notation |  | maxterm | notation |
| 0 0 0 | x’y’z’ | m0 |  | x + y + z | M0 |
| 0 0 1 | x’y’z | m1 |  | x + y + z’ | M1 |
| 0 1 0 | x’yz’ | m2 |  | x + y’ + z | M2 |
| 0 1 1 | x’yz | m3 |  | x + y’ + z’ | M3 |
| 1 0 0 | xy’z’ | m4 |  | x’ + y + z | M4 |
| 1 0 1 | xy’z | m5 |  | x’ + y + z’ | M5 |
| 1 1 0 | xyz’ | m6 |  | x’ + y’ + z | M6 |
| 1 1 1 | xyz | m7 |  | x’ + y’ + z’ | M7 |

위와 같은 표가 존재하고, F = x + yz’를 minterm으로 표현하면

F = x + yz’

= x(y + y’)(z + z’) + (x + x’)yz’

= xyz + xyz’ + xy’z + xy’z’ + x’yz’

따라서 F(x,y,z) = m(2,4,5,6,7)이다.

같은 식을 maxterm으로 표현하면

F = x + yz’

= (x + y)(x + z’)

= (x + y + z)(x + y + z’)(x + y + z’)(x + y’ + z’)

= (x + y + z)(x + y + z’)(x + y’ + z’)

따라서 F(x,y,z) = M(0,1,3)이다.

minterm과 maxterm은 보수관계이다. 이는 드모르간 법칙을 활용하여 확인할 수 있다.

F’ = (m(2,4,5,6,7))’ = (xyz + xyz’ + xy’z + xy’z’ + x’yz’)’

= (x’+y’+z’)(x’+y’+z)(x’+y+z’)(x’+y+z)(x+y’+z)

= M(2,4,5,6,7)

정리하면

F = m(2,4,5,6,7) = M(0,1,3)

F’ = M(2,4,5,6,7) = m(0,1,3)

이다.