6. 논리식의 간소화

논리회로

부경대 컴퓨터·인공지능공학부 최필주

목차

- 카르노맵
- 퀸-맥클러스키(Quine-McCluskey) 간소화 알고리즘
- NAND와 NOR 게이트
- Summary

카르노 맵 (Karnaugh Map)

- 개요
  - 1953년 Maurice Karnaugh가 소개
  - 논리식의 간소화 방법 중의 한 가지
  - 함수에서 사용할 최소항들을 각 칸 안에 넣어서 표로 만들어 놓은 것

- 카르노 맵의 표현
  - n개의 변수를 2^n개의 칸으로 표현
  - 변수가 2~4개일 때 2차원으로 쉽게 표현 가능

카르노 맵 (Karnaugh Map)

- 2변수 카르노 맵 표현 방법

$A \backslash B$	0	1
0	$\overline{A}\,\overline{B}$	$\overline{A}B$
1	$A\overline{B}$	AB

$A \backslash B$	$\overline{B}$	B
$\overline{A}$	$\overline{A}\,\overline{B}$	$\overline{A}B$
A	$A\overline{B}$	AB

$A \backslash B$	$\overline{B}$	B
$\overline{A}$	m_0	m_1
A	m_2	m_3

$B \backslash A$	$\overline{A}$	A
$\overline{B}$	m_0	m_2
B	m_1	m_3

카르노 맵 (Karnaugh Map)

2변수 카르노 맵 표현 방법 – 무관항 표현

A \ B	0	1
0	1	
1		1

$$F(A,B)=\sum m(0,3)$$

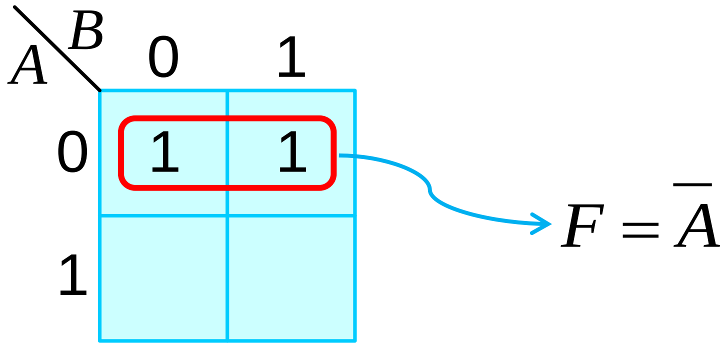
A \ B	0	1
0	1	x
1		1

$$F(A,B)=\sum m(0,3)+\sum d(1)$$

- 출력 = 1인 항과 무관항만 표시
  - 무관항 (don’t care) : 결과에 영향을 미치지 않는 항, d나 x로 표시
- 출력 0은 표시해도 되나 일반적으로 생략

카르노 맵 (Karnaugh Map)

● 카르노 맵을 이용한 간소화 방법



<카르노 맵 사용하여 간소화>

$$F = \overline{A}\,\overline{B} + \overline{A}B$$
$$= \overline{A}(\overline{B} + B) = \overline{A}\cdot 1 = \overline{A}$$

<대수 법칙 사용하여 간소화>

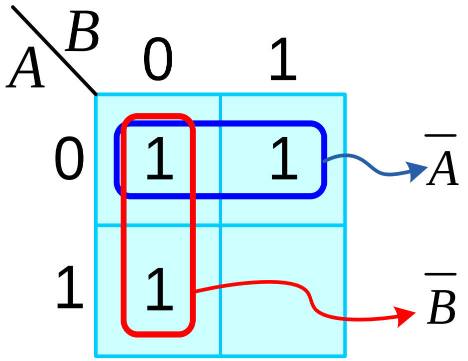
- 출력 = 1인 항(implicant)을 직사각형(정사각형 포함) 형태로 묶음
  - 2^n개(1, 2, 4, 8, 16)씩
  - 바로 이웃한 항들끼리
  - 최대한 크게 (중복 가능, 무관항은 포함 가능) ➔ PI
    - PI(prime implicant, 필수항): 더 이상 합쳐질 수 없는 항

카르노 맵 (Karnaugh Map)

- 카르노 맵을 이용한 간소화 방법 – 예

A	B	F
0	0	1
0	1	1
1	0	1
1	1	0

<진리표>



$$F = \overline{A} + \overline{B}$$

<카르노 맵 사용하여 간소화>

$$\begin{aligned} F &= \sum m(0,1,2) = \overline{A}\,\overline{B} + \overline{A}B + A\overline{B} \\ &= \overline{A}(\overline{B} + B) + \overline{B}(\overline{A} + A) \\ &= \overline{A}\cdot 1 + \overline{B}\cdot 1 \\ &= \overline{A} + \overline{B} \end{aligned}$$

<대수 법칙 사용하여 간소화>

  - 대수 법칙을 사용한 간소화 시에도 $\bar{A}\bar{B}$를 중복하여 사용함

카르노 맵 (Karnaugh Map)

- 3변수 카르노 맵 표현 방법

$A \backslash BC$	$\overline{B}\,\overline{C}$	$\overline{B}C$	BC	$B\overline{C}$
$\overline{A}$	$\overline{A}\,\overline{B}\,\overline{C}$	$\overline{A}\,\overline{B}C$	$\overline{A}BC$	$\overline{A}B\overline{C}$
A	$A\overline{B}\,\overline{C}$	$A\overline{B}C$	ABC	$AB\overline{C}$

$C \backslash AB$	$\overline{A}\,\overline{B}$	$\overline{A}B$	AB	$A\overline{B}$
$\overline{C}$	$\overline{A}\,\overline{B}\,\overline{C}$	$\overline{A}B\overline{C}$	$AB\overline{C}$	$A\overline{B}\,\overline{C}$
C	$\overline{A}\,\overline{B}C$	$\overline{A}BC$	ABC	$A\overline{B}C$

$AB \backslash C$	$\overline{C}$	C
$\overline{A}\,\overline{B}$	$\overline{A}\,\overline{B}\,\overline{C}$	$\overline{A}\,\overline{B}C$
$\overline{A}B$	$\overline{A}B\overline{C}$	$\overline{A}BC$
AB	$AB\overline{C}$	ABC
$A\overline{B}$	$A\overline{B}\,\overline{C}$	$A\overline{B}C$

$A \backslash BC$	00	01	11	10
0	0	1	3	2
1	4	5	7	6

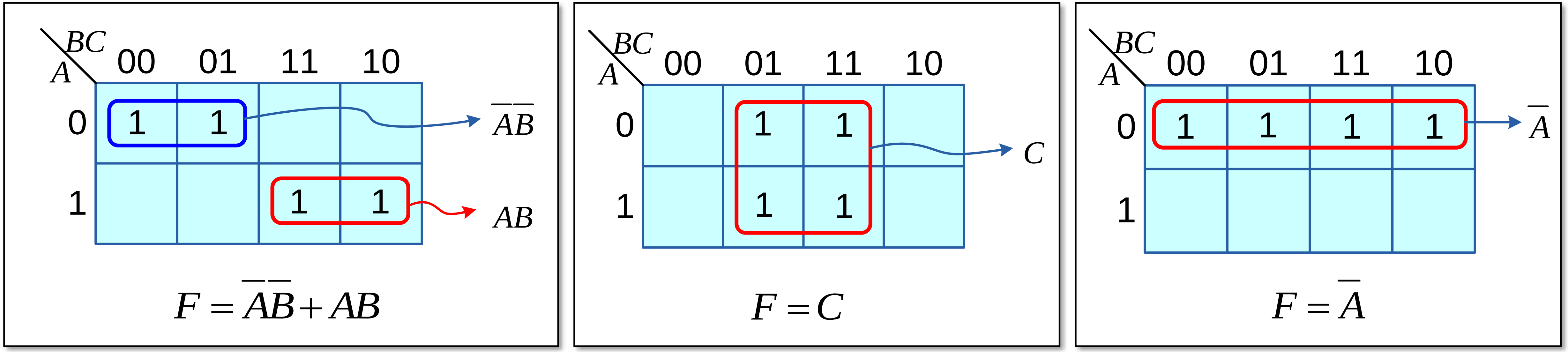
$C \backslash AB$	00	01	11	10
0	0	2	6	4
1	1	3	7	5

$AB \backslash C$	0	1
00	0	1
01	2	3
11	6	7
10	4	5

  - 설계자가 선호하는 대로 변수 배치 가능

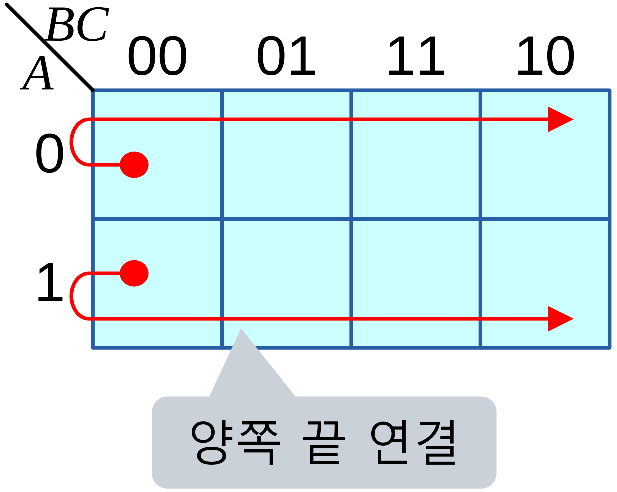
카르노 맵 (Karnaugh Map)

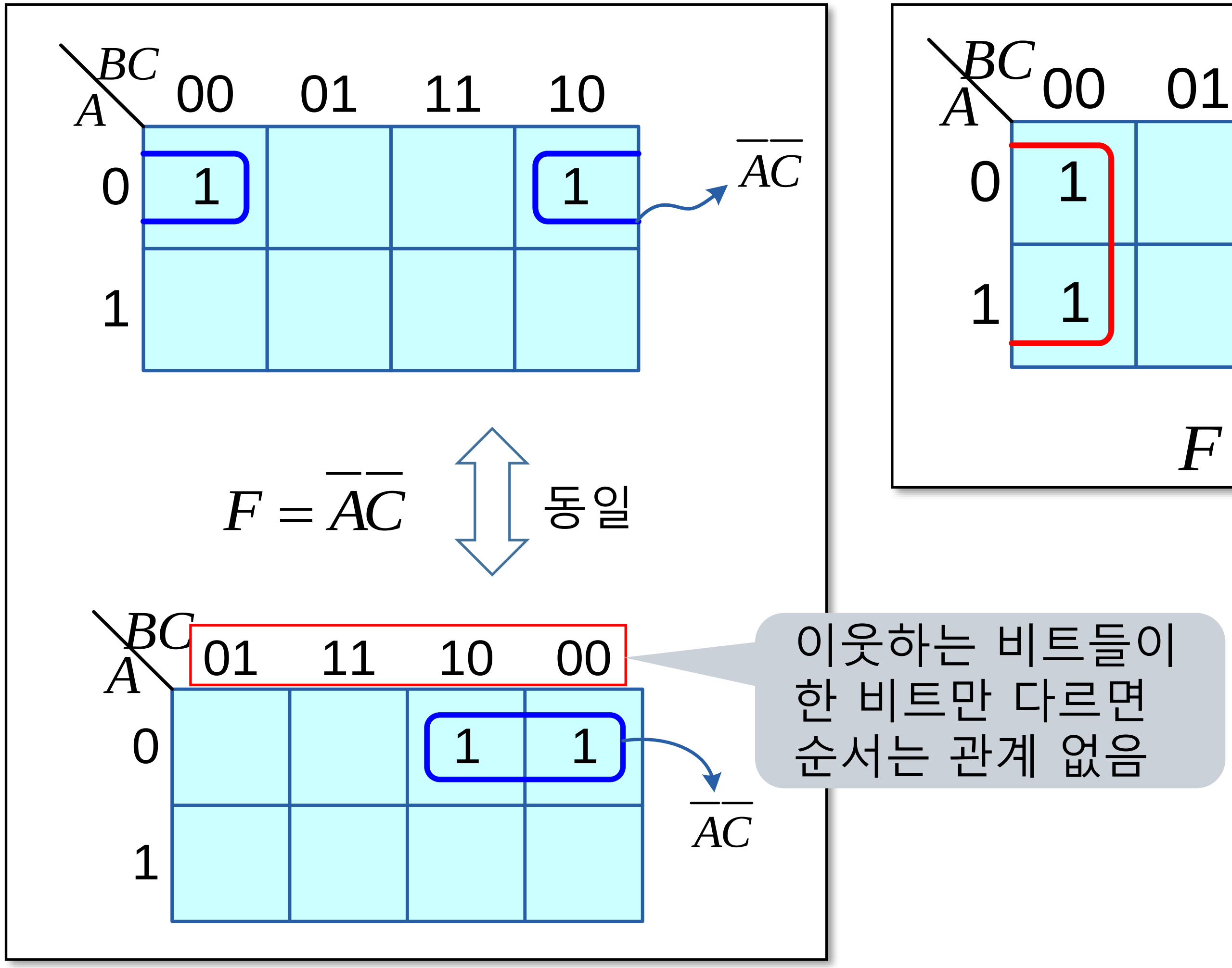
- 3변수 카르노 맵 표현 방법 – 간소화 예: 직사각형으로 묶기

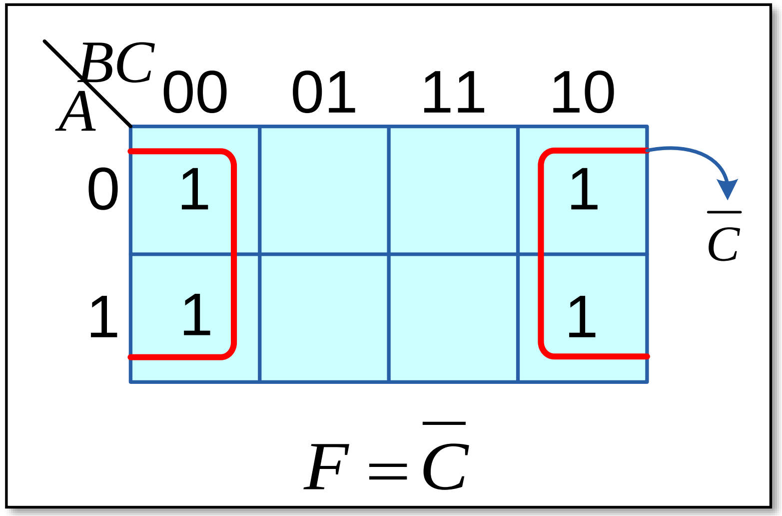


카르노 맵 (Karnaugh Map)

- 3변수 카르노 맵 표현 방법 – 간소화 예: 양쪽 끝 연결 이용

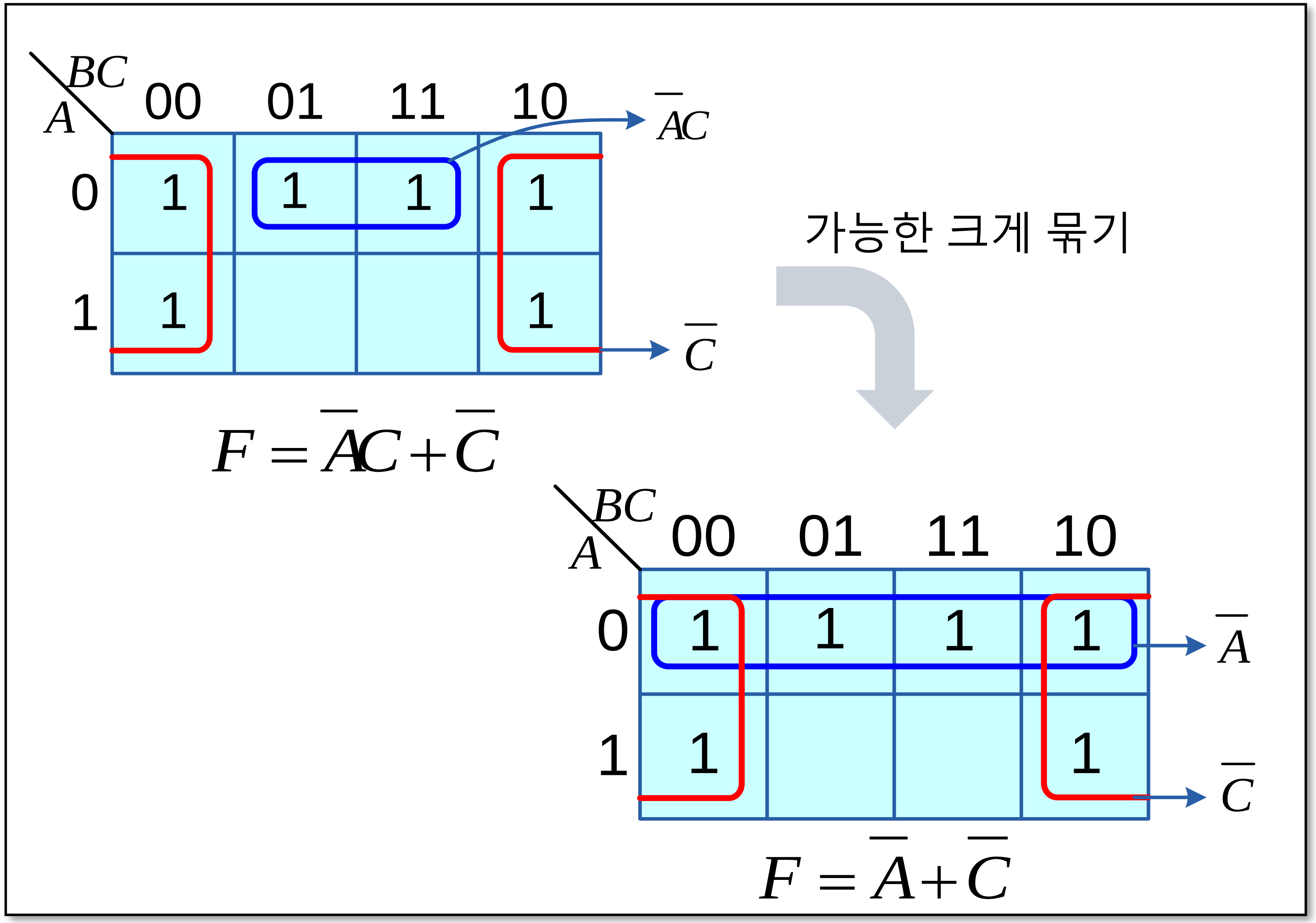


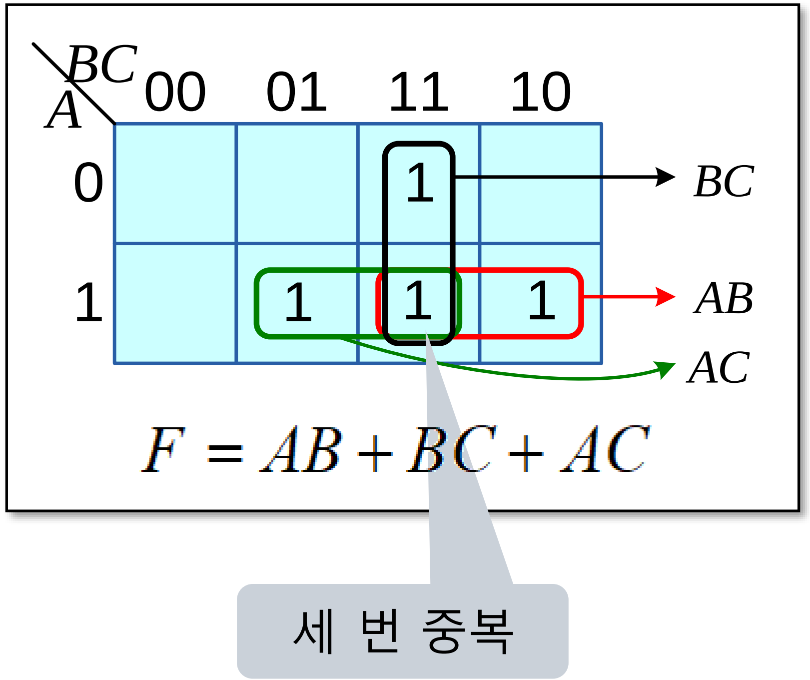




카르노 맵 (Karnaugh Map)

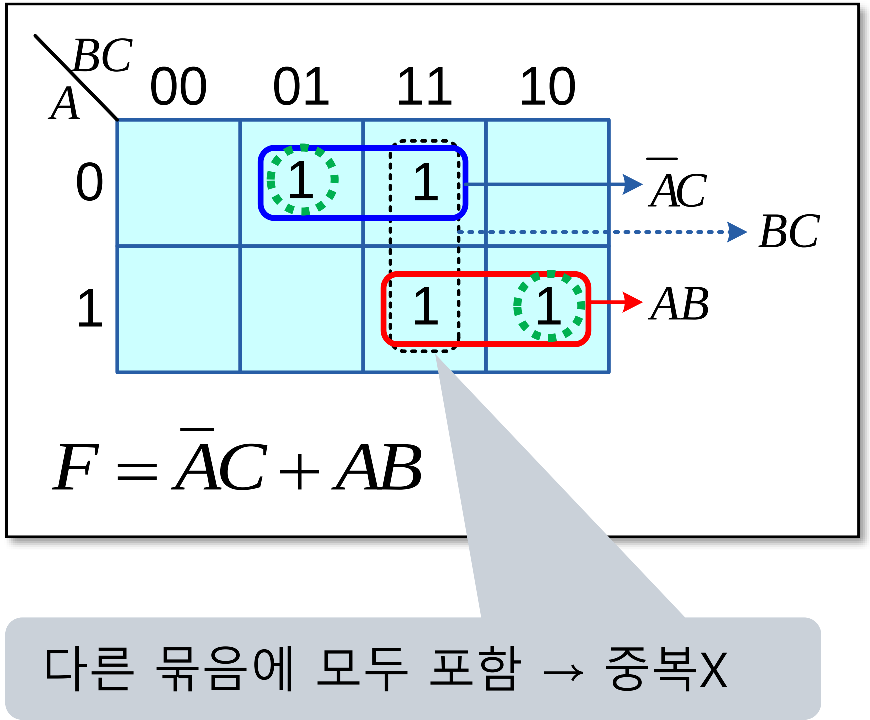
- 3변수 카르노 맵 표현 방법 – 간소화 예: 중복하여 묶기





카르노 맵 (Karnaugh Map)

- 3변수 카르노 맵 표현 방법 – 간소화 예: 불필요한 중복



  - Distinguished minterm (구분 최소항): 단 하나의 PI에만 포함된 minterm
  - EPI(Essential PI): Distinguished minterm을 포함한 PI

카르노 맵 (Karnaugh Map)

- 4변수 카르노 맵 표현 방법

AB \ CD	00	01	11	10
00	$\overline{A}\,\overline{B}\,\overline{C}\,\overline{D}$	$\overline{A}\,\overline{B}\,\overline{C}D$	$\overline{A}\,\overline{B}CD$	$\overline{A}\,\overline{B}C\overline{D}$
01	$\overline{A}B\overline{C}\,\overline{D}$	$\overline{A}B\overline{C}D$	$\overline{A}BCD$	$\overline{A}BC\overline{D}$
11	$AB\overline{C}\,\overline{D}$	$AB\overline{C}D$	$ABCD$	$ABC\overline{D}$
10	$A\overline{B}\,\overline{C}\,\overline{D}$	$A\overline{B}\,\overline{C}D$	$A\overline{B}CD$	$A\overline{B}C\overline{D}$

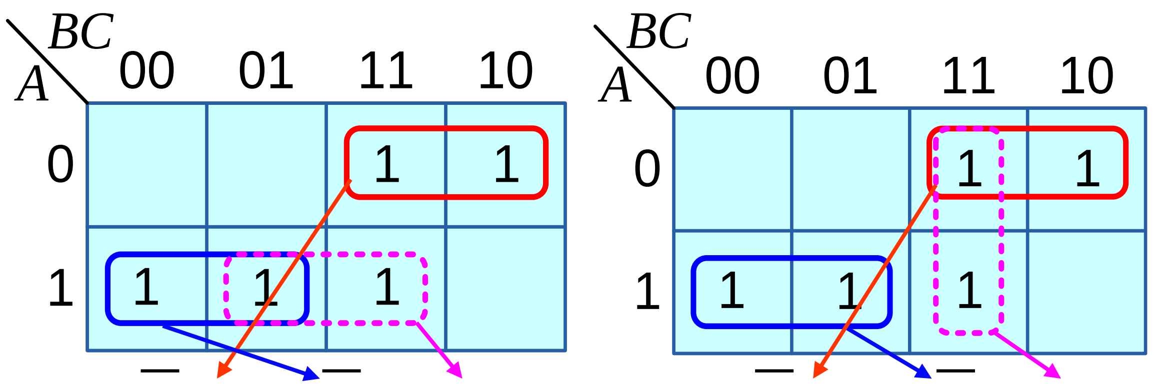
AB \ CD	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

AB \ CD	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

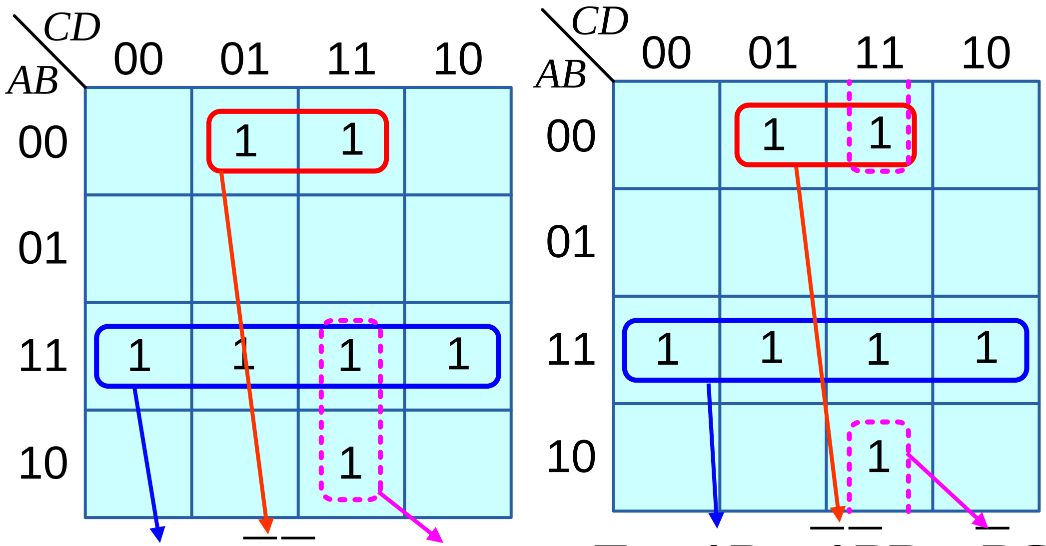
상하 좌우는
연결되어 있음

카르노 맵 (Karnaugh Map)

- 카르노 맵에서 선택적으로 묶을 수 있는 경우



$F = \bar{A}B + A\bar{B} + AC$ $\quad$ $F = \bar{A}B + A\bar{B} + BC$



$F = AB + \bar{A}\bar{B}D + ACD$ $\quad$ $F = AB + \bar{A}\bar{B}D + \bar{B}CD$

  - 2가지 답 가능

카르노 맵 (Karnaugh Map)

- 카르노 맵에서 선택적으로 묶을 수 있는 경우

AB \ CD	00	01	11	10
00	x	x	1	x
01	1	x		x
11	x	1	1	x
10	x			

$$F = \overline{A}\,\overline{B} + AB + B\overline{C}$$

AB \ CD	00	01	11	10
00	x	x	1	x
01	1	x		x
11	x	1	1	x
10	x			

$$F = \overline{A}\,\overline{B} + AB + \overline{A}\,\overline{C}$$

AB \ CD	00	01	11	10
00	x	x	1	x
01	1	x		x
11	x	1	1	x
10	x			

$$F = \overline{A}\,\overline{B} + AB + B\overline{D}$$

AB \ CD	00	01	11	10
00	x	x	1	x
01	1	x		x
11	x	1	1	x
10	x			

$$F = \overline{A}\,\overline{B} + AB + \overline{C}\,\overline{D}$$

AB \ CD	00	01	11	10
00	x	x	1	x
01	1	x		x
11	x	1	1	x
10	x			

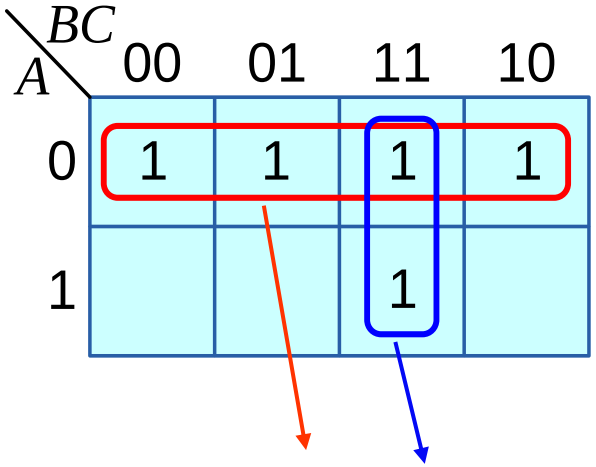
$$F = \overline{A}\,\overline{B} + AB + \overline{A}\,\overline{D}$$

  - 5가지 답 가능

카르노 맵 (Karnaugh Map)

- 논리식 → 카르노 맵 변경 후 간소화

$$F(A,B,C) = ABC + \bar{A}B + \bar{A}\bar{B}$$
$$= ABC + \bar{A}B(C+\bar{C}) + \bar{A}\bar{B}(C+\bar{C})$$
$$= ABC + \bar{A}BC + \bar{A}B\bar{C} + \bar{A}\bar{B}C + \bar{A}\bar{B}\bar{C}$$
$$= \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C} + \bar{A}BC + ABC$$
$$= \sum m(0,1,2,3,7)$$



$$F = \bar{A} + BC$$

- 방법1: 논리식의 생략된 부분 찾아 최소항으로 변경 후 표시
- 방법2: 카르노 맵에서 바로 표기
  - 예) AB: A=1, B=1인 부분에 C값 상관없이(0, 1) 모두 1표시

카르노 맵 (Karnaugh Map)

- 논리식 → 카르노 맵 변경 후 간소화

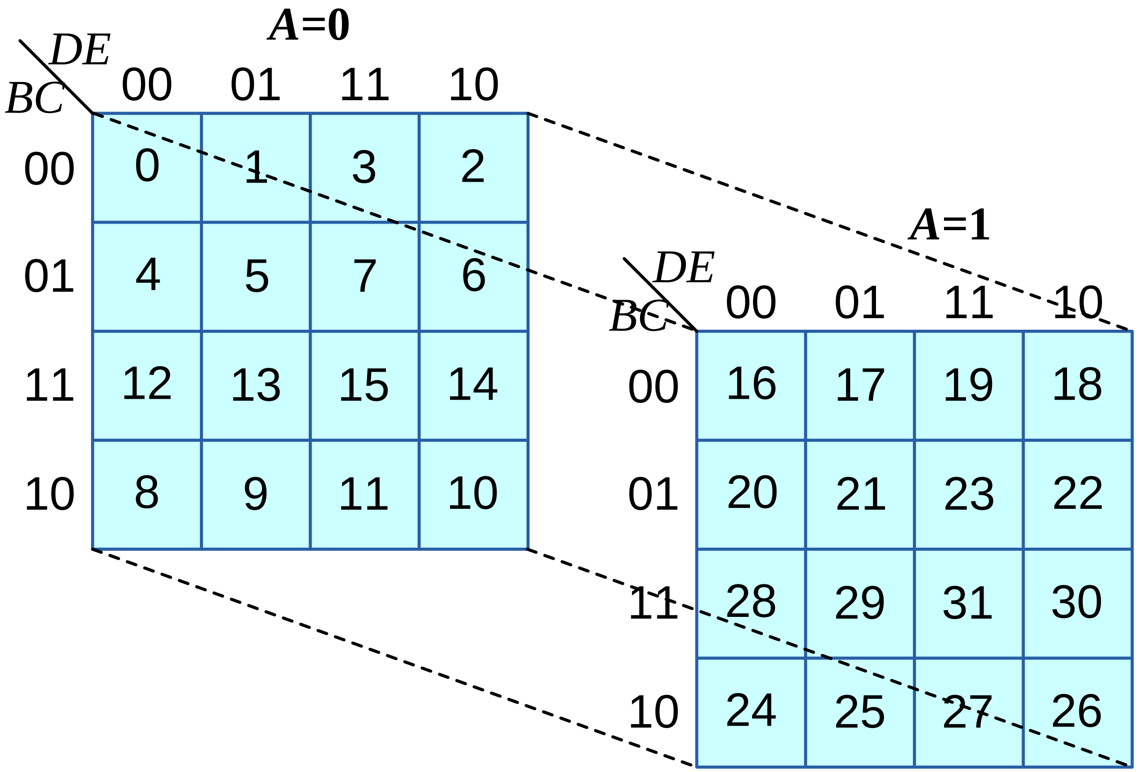
$$
\begin{aligned}
F(A,B,C,D) &= AB + ABC + \bar{A}CD + \bar{A}\bar{C}D + \bar{A}BC\bar{D} \\
&= AB(C+\bar{C})(D+\bar{D}) + ABC(D+\bar{D}) + \bar{A}(B+\bar{B})CD \\
&\quad + \bar{A}(B+\bar{B})\bar{C}D + \bar{A}BC\bar{D} \\
&= (ABC + AB\bar{C})(D+\bar{D}) + ABCD + ABC\bar{D} + \bar{A}BCD + \bar{A}\bar{B}CD \\
&\quad + \bar{A}B\bar{C}D + \bar{A}\bar{B}\bar{C}D + \bar{A}BC\bar{D} \\
&= ABCD + ABC\bar{D} + AB\bar{C}D + AB\bar{C}\bar{D} + ABCD + ABC\bar{D} + \bar{A}BCD \\
&\quad + \bar{A}\bar{B}CD + \bar{A}B\bar{C}D + \bar{A}\bar{B}\bar{C}D + \bar{A}BC\bar{D} \\
&= \sum m(15,14,13,12,7,3,5,1,6) = \sum m(1,3,5,6,7,12,13,14,15)
\end{aligned}
$$

AB \ CD	00	01	11	10
00		1	1	
01		1	1	1
11	1	1	1	1
10				

$$F = AB + \bar{A}D + BC$$

카르노 맵 (Karnaugh Map)

- 5변수인 경우



카르노 맵 (Karnaugh Map)

- 6변수인 경우

***AB*=00**

CD \ *EF*	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

***AB*=01**

CD \ *EF*	00	01	11	10
00	16	17	19	18
01	20	21	23	22
11	28	29	31	30
10	24	25	27	26

***AB*=11**

CD \ *EF*	00	01	11	10
00	48	49	51	50
01	52	53	55	54
11	60	61	63	62
10	56	57	59	58

***AB*=10**

CD \ *EF*	00	01	11	10
00	32	33	35	34
01	36	37	39	38
11	44	45	47	46
10	40	41	43	42

카르노 맵 (Karnaugh Map)

- 카르노 맵에서의 XOR와 XNOR
  - 1이 이웃하지 않아 카르노 맵에서 XOR나 XNOR를 표현하기 어려움

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

$F = A \oplus B \oplus \mathrm{C}$

AB \ CD	00	01	11	10
00		1		1
01	1		1	
11		1		1
10	1		1	

$F = A \oplus B \oplus C \oplus \mathrm{D}$

AB \ CD	00	01	11	10
00		1	1	
01	1			1
11		1	1	
10	1			1

$F = A \oplus B \oplus \mathrm{D}$

AB \ CD	00	01	11	10
00	1	1		
01			1	1
11	1	1		
10			1	1

$F = A \oplus B \oplus \mathrm{C}$

카르노 맵 (Karnaugh Map)

- 카르노맵을 XOR로 표현하기
  - 1인 부분이 **홀수** 번만 포함되도록 상자 배치

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

$F = A \oplus B \oplus C$

AB \ CD	00	01	11	10
00		1		1
01	1		1	
11		1		1
10	1		1	

$F = A \oplus B \oplus C \oplus D$

AB \ CD	00	01	11	10
00		1	1	
01	1			1
11		1	1	
10	1			1

$F = A \oplus B \oplus D$

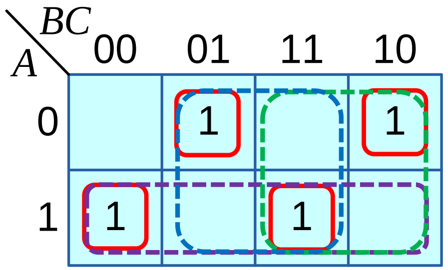
AB \ CD	00	01	11	10
00	1	1		
01			1	1
11	1	1		
10			1	1

$F = A \oplus B \oplus C$

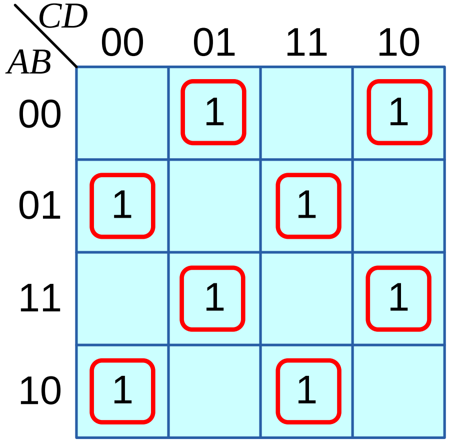
- ASIC 칩: XOR 게이트는 NAND, AND 게이트에 비해 큼
- FPGA 칩: 모든 게이트는 동일 크기

카르노 맵 (Karnaugh Map)

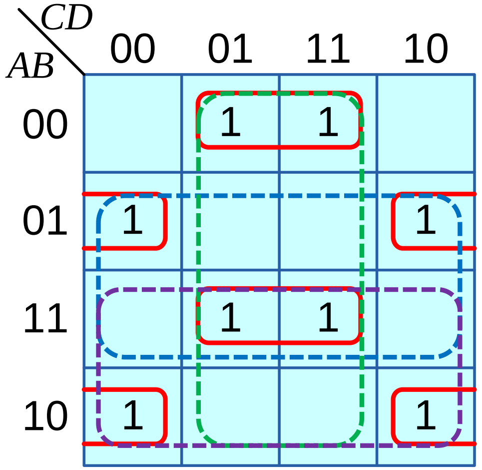
- 카르노맵을 XOR로 표현하기
  - 1인 부분이 **홀수** 번만 포함되도록 상자 배치



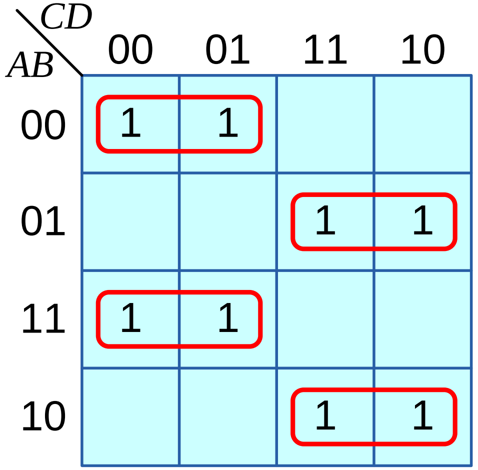
$F = A \oplus B \oplus C$



$F = A \oplus B \oplus C \oplus D$



$F = A \oplus B \oplus D$



$F = A \oplus B \oplus C$

퀸-맥클러스키(Quine-McCluskey) 간소화 알고리즘

- 개요
  - 퀸(Willard Van Orman Quine)과 맥클러스키(Edward J. McCluskey)가 1956년에 개발
  - 컴퓨터 알고리즘으로 개발하기에 적합
  - 입력 변수가 5개 이상일 경우 유용
    - 입력 변수가 4개 이하인 경우 카르노 맵 이용이 편리함

퀸-맥클러스키(Quine-McCluskey) 간소화 알고리즘

방법

인덱스 분류 → $AB + A\bar{B} = A$ 규칙을 이용하여 PI를 구함 → PI 차트를 이용하여 EPI 구함

$A\bar{B}C$: 101로 표현 (A=1, B=0, C=1)
$\bar{A}B\bar{C}$: 010로 표현 (A=0, B=1, C=0)
$A\bar{B}$: 10-로 표현 (A=1, B=0, C= ×)
AC : 1-1로 표현 (A=1, B=×, C=1)

용어정리

- Implicant: 간소화되거나 최소화될 항
- PI(Prime Implicant) : 최종적으로 남아있는 곱의 항
- EPI(Essential Prime Implicant) : PI중에서 유일한 PI

퀸-맥클러스키(Quine-McCluskey) 간소화 알고리즘

- 예시1 – 인덱스 분류

$$F(A,B,C) = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + A\bar{B}\bar{C} + A\bar{B}C$$
$$= \sum m(0,1,4,5)$$
$$= \sum m(000,\ 001,\ 100,\ 101)$$

Column 1			
index	decimal	$A\ B\ C$	
0	(0)	0 0 0	
1	(1) (4)	0 0 1 1 0 0	
2	(5)	1 0 1	

  - 값이 1인 입력 변수의 개수에 따라 index 분류

퀸-맥클러스키(Quine-McCluskey) 간소화 알고리즘

- 예시1 – PI구하기: index0-index1 통합

Column 1			
index	decimal	*A B C*	
0	(0)	0 0 0	✓
1	(1) (4)	0 0 1 1 0 0	✓
2	(5)	1 0 1	

Column 2			
index	decimal	*A B C*	
0	(0,1)	0 0 –	
1			

  - 이웃한 index 내 항들끼리 한 비트 차이나는 항을 통합

퀸-맥클러스키(Quine-McCluskey) 간소화 알고리즘

- 예시1 – PI구하기: index0-index1 통합

Column 1			
index	decimal	*A B C*	
0	(0)	0 0 0	✓
1	(1) (4)	0 0 1 1 0 0	✓ ✓
2	(5)	1 0 1	

Column 2			
index	decimal	*A B C*	
0	(0,1) (0,4)	0 0 - - 0 0	
1			

퀸-맥클러스키(Quine-McCluskey) 간소화 알고리즘

- 예시1 – PI구하기: index1-index2 통합

Column 1			
index	decimal	*A B C*	
0	(0)	0 0 0	✓
1	(1) (4)	0 0 1 1 0 0	✓ ✓
2	(5)	1 0 1	✓

→

Column 2			
index	decimal	*A B C*	
0	(0,1) (0,4)	0 0 - - 0 0	
1	(1,5)	- 0 1	

- 예시1 – PI구하기: index1-index2 통합

Column 1			
index	decimal	*A B C*	
0	(0)	0 0 0	✓
1	(1) (4)	0 0 1 1 0 0	✓ ✓
2	(5)	1 0 1	✓

Column 2			
index	decimal	*A B C*	
0	(0,1) (0,4)	0 0 - - 0 0	
1	(1,5) (4,5)	- 0 1 1 0 -	

퀸-맥클러스키(Quine-McCluskey) 간소화 알고리즘

- 예시1 – PI구하기: index0-index1 통합

Column 1			
index	decimal	$A\ B\ C$	
0	(0)	0 0 0	✓
1	(1) (4)	0 0 1 1 0 0	✓ ✓
2	(5)	1 0 1	✓

Column 2			
index	decimal	$A\ B\ C$	
0	(0,1) (0,4)	0 0 - - 0 0	✓ ✓
1	(1,5) (4,5)	- 0 1 1 0 -	✓ ✓

Column 3		
decimal	$A\ B\ C$	
(0, 1, 4, 5)	**- 0 -**	

퀸-맥클러스키(Quine-McCluskey) 간소화 알고리즘

- 예시2 – PI구하기

Column 1			
index	decimal	*A B C*	
0	(0)	0 0 0	✓
1	(1) (4)	0 0 1 1 0 0	✓ ✓
2	(5)	1 0 1	✓

Column 2			
index	decimal	*A B C*	
0	(0,1) (0,4)	0 0 - - 0 0	✓ ✓
1	(1,5) (4,5)	- 0 1 1 0 -	✓ ✓

Column 3		
decimal	*A B C*	
(0, 1, 4, 5)	**- 0 -**	●

- 더 이상 통합할 항이 없을 때 체크되지 않은 항이 PI(●)
- 위 예시에서 PI는 $\bar{B}$ 하나뿐

퀸-맥클러스키(Quine-McCluskey) 간소화 알고리즘

- 예시2 – 인덱스 분류
  - $F(A,B,C,D) = \Sigma m(0,1,2,3,5,7,8,10,12,13,15)$

Column 1			
index	decimal	$A\ B\ C\ D$	
0	(0)	0 0 0 0	
1	(1) (2) (8)	0 0 0 1 0 0 1 0 1 0 0 0	
2	(3) (5) (10) (12)	0 0 1 1 0 1 0 1 1 0 1 0 1 1 0 0	
3	(7) (13)	0 1 1 1 1 1 0 1	
4	(15)	1 1 1 1	

퀸-맥클러스키(Quine-McCluskey) 간소화 알고리즘

- 예시2 – PI구하기: index0-index1 통합

Column 1			
index	decimal	*A B C D*	
0	(0)	0 0 0 0	✓
1	(1) (2) (8)	0 0 0 1 0 0 1 0 1 0 0 0	✓
2	(3) (5) (10) (12)	0 0 1 1 0 1 0 1 1 0 1 0 1 1 0 0	
3	(7) (13)	0 1 1 1 1 1 0 1	
4	(15)	1 1 1 1	

Column 2			
index	decimal	*A B C D*	
0	(0, 1)	0 0 0 -	
1			
2			
3			

퀸-맥클러스키(Quine-McCluskey) 간소화 알고리즘

- 예시2 – PI구하기: index0-index1 통합

Column 1			
index	decimal	*A B C D*	
0	(0)	0 0 0 0	✓
1	(1) (2) (8)	0 0 0 1 0 0 1 0 1 0 0 0	✓ ✓
2	(3) (5) (10) (12)	0 0 1 1 0 1 0 1 1 0 1 0 1 1 0 0	
3	(7) (13)	0 1 1 1 1 1 0 1	
4	(15)	1 1 1 1	

Column 2			
index	decimal	*A B C D*	
0	(0, 1) (0, 2)	0 0 0 - 0 0 - 0	
1			
2			
3			

퀸-맥클러스키(Quine-McCluskey) 간소화 알고리즘

- 예시2 – PI구하기: index0-index1 통합

Column 1			
index	decimal	*A B C D*	
0	(0)	0 0 0 0	✓
1	(1) (2) (8)	0 0 0 1 0 0 1 0 1 0 0 0	✓ ✓ ✓
2	(3) (5) (10) (12)	0 0 1 1 0 1 0 1 1 0 1 0 1 1 0 0	
3	(7) (13)	0 1 1 1 1 1 0 1	
4	(15)	1 1 1 1	

Column 2			
index	decimal	*A B C D*	
0	(0, 1) (0, 2) (0, 8)	0 0 0 - 0 0 - 0 - 0 0 0	
1			
2			
3			

퀸-맥클러스키(Quine-McCluskey) 간소화 알고리즘

- 예시2 – PI구하기: index1-index2 통합

Column 1			
index	decimal	*A B C D*	
0	(0)	0 0 0 0	✓
1	(1) (2) (8)	0 0 0 1 0 0 1 0 1 0 0 0	✓ ✓ ✓
2	(3) (5) (10) (12)	0 0 1 1 0 1 0 1 1 0 1 0 1 1 0 0	✓ ✓ ✓ ✓
3	(7) (13)	0 1 1 1 1 1 0 1	
4	(15)	1 1 1 1	

Column 2			
index	decimal	*A B C D*	
0	(0, 1) (0, 2) (0, 8)	0 0 0 - 0 0 - 0 - 0 0 0	
1	(1, 3) (1, 5) (2, 3) (2, 10) (8, 10) (8, 12)	0 0 - 1 0 - 0 1 0 0 1 - - 0 1 0 1 0 - 0 1 - 0 0	
2			
3			

퀸-맥클러스키(Quine-McCluskey) 간소화 알고리즘

- 예시2 – PI구하기: index2-index3 통합

Column 1			
index	decimal	*A B C D*	
0	(0)	0 0 0 0	✓
1	(1) (2) (8)	0 0 0 1 0 0 1 0 1 0 0 0	✓ ✓ ✓
2	(3) (5) (10) (12)	0 0 1 1 0 1 0 1 1 0 1 0 1 1 0 0	✓ ✓ ✓ ✓
3	(7) (13)	0 1 1 1 1 1 0 1	✓ ✓
4	(15)	1 1 1 1	

Column 2			
index	decimal	*A B C D*	
0	(0, 1) (0, 2) (0, 8)	0 0 0 - 0 0 - 0 - 0 0 0	
1	(1, 3) (1, 5) (2, 3) (2, 10) (8, 10) (8, 12)	0 0 - 1 0 - 0 1 0 0 1 - - 0 1 0 1 0 - 0 1 - 0 0	
2	(3, 7) (5, 7) (5, 13) (12, 13)	0 - 1 1 0 1 - 1 - 1 0 1 1 1 0 -	
3			

퀸-맥클러스키(Quine-McCluskey) 간소화 알고리즘

- 예시2 – PI구하기: index3-index4 통합

Column 1			
index	decimal	*A B C D*	
0	(0)	0 0 0 0	✓
1	(1) (2) (8)	0 0 0 1 0 0 1 0 1 0 0 0	✓ ✓ ✓
2	(3) (5) (10) (12)	0 0 1 1 0 1 0 1 1 0 1 0 1 1 0 0	✓ ✓ ✓ ✓
3	(7) (13)	0 1 1 1 1 1 0 1	✓ ✓
4	(15)	1 1 1 1	✓

Column 2			
index	decimal	*A B C D*	
0	(0, 1) (0, 2) (0, 8)	0 0 0 - 0 0 - 0 - 0 0 0	
1	(1, 3) (1, 5) (2, 3) (2, 10) (8, 10) (8, 12)	0 0 - 1 0 - 0 1 0 0 1 - - 0 1 0 1 0 - 0 1 - 0 0	
2	(3, 7) (5, 7) (5, 13) (12, 13)	0 - 1 1 0 1 - 1 - 1 0 1 1 1 0 -	
3	(7, 15) (13, 15)	- 1 1 1 1 1 - 1	

퀸-맥클러스키(Quine-McCluskey) 간소화 알고리즘

- 예시2 – PI구하기: index0-index1 통합

Column 1			
index	decimal	*A B C D*	
0	(0)	0 0 0 0	✓
1	(1) (2) (8)	0 0 0 1 0 0 1 0 1 0 0 0	✓ ✓ ✓
2	(3) (5) (10) (12)	0 0 1 1 0 1 0 1 1 0 1 0 1 1 0 0	✓ ✓ ✓ ✓
3	(7) (13)	0 1 1 1 1 1 0 1	✓ ✓
4	(15)	1 1 1 1	✓

Column 2			
index	decimal	*A B C D*	
0	(0, 1) (0, 2) (0, 8)	0 0 0 - 0 0 - 0 - 0 0 0	✓ ✓ ✓
1	(1, 3) (1, 5) (2, 3) (2, 10) (8, 10) (8, 12)	0 0 - 1 0 - 0 1 0 0 1 - - 0 1 0 1 0 - 0 1 - 0 0	✓ ✓ ✓ ✓
2	(3, 7) (5, 7) (5, 13) (12, 13)	0 - 1 1 0 1 - 1 - 1 0 1 1 1 0 -	
3	(7, 15) (13, 15)	- 1 1 1 1 1 - 1	

Column 3			
index	decimal	*A B C D*	
0	(0,1,2,3) (0,2,8,10)	0 0 - - - 0 - 0	
1			
2			

퀸-맥클러스키(Quine-McCluskey) 간소화 알고리즘

- 예시2 – PI구하기: index1-index2 통합

Column 1			
index	decimal	*A B C D*	
0	(0)	0 0 0 0	✓
1	(1) (2) (8)	0 0 0 1 0 0 1 0 1 0 0 0	✓ ✓ ✓
2	(3) (5) (10) (12)	0 0 1 1 0 1 0 1 1 0 1 0 1 1 0 0	✓ ✓ ✓ ✓
3	(7) (13)	0 1 1 1 1 1 0 1	✓ ✓
4	(15)	1 1 1 1	✓

Column 2			
index	decimal	*A B C D*	
0	(0, 1) (0, 2) (0, 8)	0 0 0 - 0 0 - 0 - 0 0 0	✓ ✓ ✓
1	(1, 3) (1, 5) (2, 3) (2, 10) (8, 10) (8, 12)	0 0 - 1 0 - 0 1 0 0 1 - - 0 1 0 1 0 - 0 1 - 0 0	✓ ✓ ✓ ✓ ✓
2	(3, 7) (5, 7) (5, 13) (12, 13)	0 - 1 1 0 1 - 1 - 1 0 1 1 1 0 -	✓ ✓
3	(7, 15) (13, 15)	- 1 1 1 1 1 - 1	

Column 3			
index	decimal	*A B C D*	
0	(0,1,2,3) (0,2,8,10)	0 0 - - - 0 - 0	
1	(1,3,5,7)	0 - - 1	
2			

퀸-맥클러스키(Quine-McCluskey) 간소화 알고리즘

- 예시2 – PI구하기: index2-index3 통합

Column 1			
index	decimal	*A B C D*	
0	(0)	0 0 0 0	✓
1	(1) (2) (8)	0 0 0 1 0 0 1 0 1 0 0 0	✓ ✓ ✓
2	(3) (5) (10) (12)	0 0 1 1 0 1 0 1 1 0 1 0 1 1 0 0	✓ ✓ ✓ ✓
3	(7) (13)	0 1 1 1 1 1 0 1	✓ ✓
4	(15)	1 1 1 1	✓

Column 2			
index	decimal	*A B C D*	
0	(0, 1) (0, 2) (0, 8)	0 0 0 - 0 0 - 0 - 0 0 0	✓ ✓ ✓
1	(1, 3) (1, 5) (2, 3) (2, 10) (8, 10) (8, 12)	0 0 - 1 0 - 0 1 0 0 1 - - 0 1 0 1 0 - 0 1 - 0 0	✓ ✓ ✓ ✓ ✓
2	(3, 7) (5, 7) (5, 13) (12, 13)	0 - 1 1 0 1 - 1 - 1 0 1 1 1 0 -	✓ ✓ ✓
3	(7, 15) (13, 15)	- 1 1 1 1 1 - 1	✓ ✓

Column 3			
index	decimal	*A B C D*	
0	(0,1,2,3) (0,2,8,10)	0 0 - - - 0 - 0	
1	(1,3,5,7)	0 - - 1	
2	(5,7,13,15)	- 1 - 1	

퀸-맥클러스키(Quine-McCluskey) 간소화 알고리즘

- 예시2 – PI구하기: index2-index3 통합

Column 1			
index	decimal	*A B C D*	
0	(0)	0 0 0 0	✓
1	(1) (2) (8)	0 0 0 1 0 0 1 0 1 0 0 0	✓ ✓ ✓
2	(3) (5) (10) (12)	0 0 1 1 0 1 0 1 1 0 1 0 1 1 0 0	✓ ✓ ✓ ✓
3	(7) (13)	0 1 1 1 1 1 0 1	✓ ✓
4	(15)	1 1 1 1	✓

Column 2			
index	decimal	*A B C D*	
0	(0, 1) (0, 2) (0, 8)	0 0 0 - 0 0 - 0 - 0 0 0	✓ ✓ ✓
1	(1, 3) (1, 5) (2, 3) (2, 10) (8, 10) (8, 12)	0 0 - 1 0 - 0 1 0 0 1 - - 0 1 0 1 0 - 0 1 - 0 0	✓ ✓ ✓ ✓ ✓
2	(3, 7) (5, 7) (5, 13) (12, 13)	0 - 1 1 0 1 - 1 - 1 0 1 1 1 0 -	✓ ✓ ✓
3	(7, 15) (13, 15)	- 1 1 1 1 1 - 1	✓ ✓

Column 3			
index	decimal	*A B C D*	
0	(0,1,2,3) (0,2,8,10)	0 0 - - - 0 - 0	
1	(1,3,5,7)	0 - - 1	
2	(5,7,13,15)	- 1 - 1	

퀸-맥클러스키(Quine-McCluskey) 간소화 알고리즘

- 예시2 – PI구하기: index2-index3 통합

Column 1			
index	decimal	$A\ B\ C\ D$	
0	(0)	0 0 0 0	✓
1	(1)	0 0 0 1	✓
	(2)	0 0 1 0	✓
	(8)	1 0 0 0	✓
2	(3)	0 0 1 1	✓
	(5)	0 1 0 1	✓
	(10)	1 0 1 0	✓
	(12)	1 1 0 0	✓
3	(7)	0 1 1 1	✓
	(13)	1 1 0 1	✓
4	(15)	1 1 1 1	✓

Column 2			
index	decimal	$A\ B\ C\ D$	
0	(0, 1)	0 0 0 -	✓
	(0, 2)	0 0 - 0	✓
	(0, 8)	- 0 0 0	✓
1	(1, 3)	0 0 - 1	✓
	(1, 5)	0 - 0 1	✓
	(2, 3)	0 0 1 -	✓
	(2, 10)	- 0 1 0	✓
	(8, 10)	1 0 - 0	✓
	(8, 12)	1 - 0 0	●
2	(3, 7)	0 - 1 1	✓
	(5, 7)	0 1 - 1	✓
	(5, 13)	- 1 0 1	✓
	(12, 13)	1 1 0 -	●
3	(7, 15)	- 1 1 1	✓
	(13, 15)	1 1 - 1	✓

Column 3			
index	decimal	$A\ B\ C\ D$	
0	(0,1,2,3)	0 0 - -	●
	(0,2,8,10)	- 0 - 0	●
1	(1,3,5,7)	0 - - 1	●
2	(5,7,13,15)	- 1 - 1	●

$$\bar{A}\bar{B} + \bar{B}\bar{D} + \bar{A}D + BD$$
$$+A\bar{C}\bar{D} + AB\bar{C}$$

퀸-맥클러스키(Quine-McCluskey) 간소화 알고리즘

예시2 – EPI구하기

- PI: $\bar{A}\bar{B}, \bar{B}\bar{D}, \bar{A}D, BD, A\bar{C}\bar{D}, AB\bar{C}$
- 최소항을 포함하는 유일한 PI (즉, EPI) 찾기

Implicants		0	1	2	3	5	7	8	10	12	13	15
$\bar{A}\bar{B}$	(0,1,2,3)	×	×	×	×							
$\bar{B}\bar{D}$	(0,2,8,10)	×		×				×	×			
$\bar{A}D$	(1,3,5,7)		×		×	×	×					
BD	(5,7,13,15)					×	×				×	×
$A\bar{C}\bar{D}$	(8, 12)							×		×		
$AB\bar{C}$	(12, 13)									×	×	

EPI 및 EPI가 포함하는 최소항 제거

Implicants		1	3	12
$\bar{A}\bar{B}$	(0,1,2,3)	×	×	
$\bar{A}D$	(1,3,5,7)	×	×	
$A\bar{C}\bar{D}$	(8, 12)			×
$AB\bar{C}$	(12, 13)			×

- $F = \bar{B}\bar{D} + BD + \bar{A}\bar{B} + A\bar{C}\bar{D}$ or $F = \bar{B}\bar{D} + BD + \bar{A}D + AB\bar{C}$ or
  $F = \bar{B}\bar{D} + BD + \bar{A}\bar{B} + AB\bar{C}$ or $F = \bar{B}\bar{D} + BD + \bar{A}D + A\bar{C}\bar{D}$

NAND와 NOR 게이트로의 변환

- 기본 게이트의 NAND, NOR 식

기본게이트		NAND로 표현		NOR로 표현	
NOT	$\bar{A}$	$\overline{A\cdot A}$		$\overline{A+A}$	
AND	$A\cdot B$	$\overline{\overline{A\cdot B}}$	NAND → NOT	$\overline{\bar{A}+\bar{B}}$	NOTx2 → NOR
OR	$A+B$	$\overline{\bar{A}\cdot\bar{B}}$	NOTx2 → NAND	$\overline{\overline{A+B}}$	NOR → NOT
NAND	$\overline{A\cdot B}$	$\overline{A\cdot B}$		$\overline{\overline{\bar{A}+\bar{B}}}$	NOTx2 → NOR → NOT
NOR	$\overline{A+B}$	$\overline{\overline{\bar{A}\cdot\bar{B}}}$	NOTx2 → NAND → NOT	$\overline{A+B}$	
XOR	$\bar{A}B+A\bar{B}$	$\overline{\overline{\bar{A}\cdot B}\cdot\overline{A\cdot\bar{B}}}$	NOT → NANDx2 → NAND	$\overline{\overline{\overline{A+\bar{B}}+\overline{\bar{A}+B}}}$	NOTx2 → NORx2 → NOR → NOT
XNOR	$\bar{A}\bar{B}+AB$	$\overline{\overline{\overline{\bar{A}\cdot B}\cdot\overline{A\cdot\bar{B}}}}$	NOT → NANDx2 → NAND → NOT	$\overline{\overline{A+\bar{B}}+\overline{\bar{A}+B}}$	NOTx2 → NOR → NOR

  - NAND, NOR를 만능 게이트라고 부름
  - 2-input NAND의 면적을 기준으로 gate equivalents (GEs) 표현

Summary

- 카르노 맵 (Karnaugh Map)
  - 출력 = 1인 항과 무관항만 표시
  - 출력 = 1인 항을 직사각형 형태로 묶음
    - 2^n개(1, 2, 4, 8, 16)씩
    - 바로 이웃한 항들끼리
    - 최대한 크게 (중복 가능, 무관항은 포함 가능)
  - 입력 변수가 3~4개일 때 유용

AB \ CD	00	01	11	10
00	x	x	1	x
01	1	x		x
11	x	1	1	x
10	x			

$$F = \overline{A}\,\overline{B} + AB + B\overline{C}$$

Summary

- 퀸-맥클러스키(Quine-McCluskey) 간소화 알고리즘
  - 변수가 5개 이상일 경우 적합, 컴퓨터 알고리즘으로 구현하기 적합
  - 방법

인덱스 분류	→	$AB + A\bar{B} = A$ 규칙을 이용하여 PI를 구함	→	PI 차트를 이용하여 EPI 구함

- 인덱스 분류: 값이 1인 입력 변수의 개수에 따라
- PI 구하기: 이웃한 index 내 항들끼리 한 비트 차이나는 항을 통합 → 반복 → 더 이상 통합되지 않는 항이 PI
- EPI 구하기: 차트 그리기(각 PI가 포함하는 최소항 표시) → 다른 PI에 중복 포함되지 않는 항을 갖는 PI (EPI) 찾기 → 차트에서 찾은 EPI 및 EPI가 포함하는 최소항 제거 → 남은 차트에서 EPI 선택

Implicants		0	1	2	3	5	7	8	10	12	13	15
$\bar{A}\bar{B}$	(0,1,2,3)	×	×	×	×							
$\bar{B}\bar{D}$	(0,2,8,10)	×		×				×	×			
$\bar{A}D$	(1,3,5,7)		×		×	×	×					
BD	(5,7,13,15)					×	×				×	×
$A\bar{C}\bar{D}$	(8, 12)							×		×		
$AB\bar{C}$	(12, 13)									×	×	

Summary

- NAND와 NOR 게이트
  - 만능 게이트: 다른 모든 게이트 구성 가능
  - NAND는 디지털 회로의 자원량을 나타내는 GE 계산 시에도 사용됨
    - GEs (Gate Equivalents): 전체 면적/NAND의 면적