

# Chapter 7. 다단 게이트 회로, NAND, NOR 게이트

⊗ AND와 OR 게이트로 이루어진 2단 이상의 회로를 설계하는 방법

→ 원하는 형태의 게이트회로 (NAND, NOR 게이트)로 바꾸는 기술

## 7.1 다단 게이트 회로 (AND 와 OR 게이트)

게이트의 단 수 : 회로 입력과 출력 사이에 직렬로 연결된 게이트의 최대 수 (NOT 게이트는 제외)

⊕ 모든 변수와 그 변수의 보수가 회로입력으로 활용가능

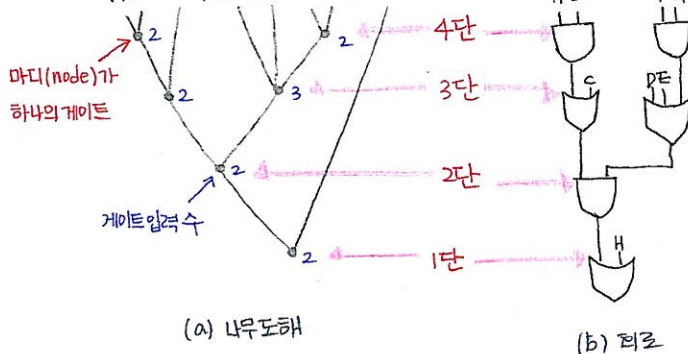
- 2단 이상의 회로 종류
- ① AND-OR 회로 : 출력에 OR 게이트  
 그 앞에 AND 게이트의 단으로 된 2단 회로
  - ② OR-AND 회로 : 출력에 AND 게이트  
 그 앞에 OR 게이트의 단으로 된 2단 회로
  - ③ OR-AND-OR 회로 : 출력에 OR 게이트  
 그 앞에 AND 게이트 단이 있고 또 그 앞에  
 OR 게이트 단으로 된 3단 회로

⊗ AND-OR 회로에서 논리곱의 합식을 인수분해하면 단수를 증가시킬 수 있음  
 곱의 합식

OR-AND 회로에서 논리합의 곱 식에서 몇몇 항을 곱하면 단수를 증가시킬 수 있음  
 합의 곱식

⊗ 회로에서 '게이트 수', '게이트 입력 수' 그리고 '단 수'는 해당 식을 조사하여 결정

$$\text{ex) } Z = (AB + C)(D + E + FG) + H$$



⇒ 4단  
 6개 게이트  
 13개 게이트 입력

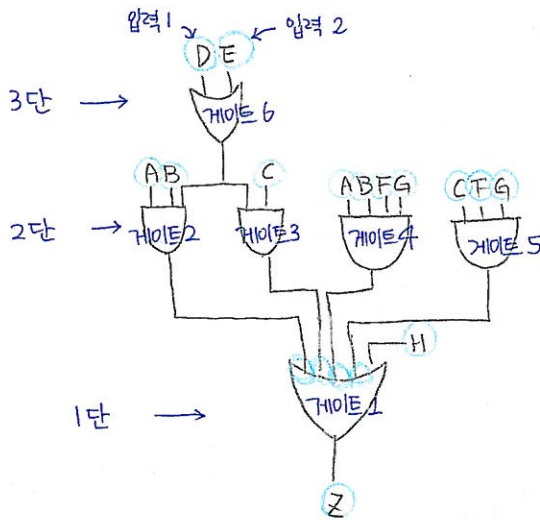
Z를 3단으로 바꾸기

tb\_elec\_engineer@naver.com

$$Z = (AB + C) [(D + E) + FG] + H$$

(gm)

$$= AB(D + E) + C(D + E) + ABFG + CFG + H$$



⇒ 3단

6개 게이트

19개 게이트 입력

ex) 다음 함수를 실현하기 위한 AND와 OR 회로를 구하라

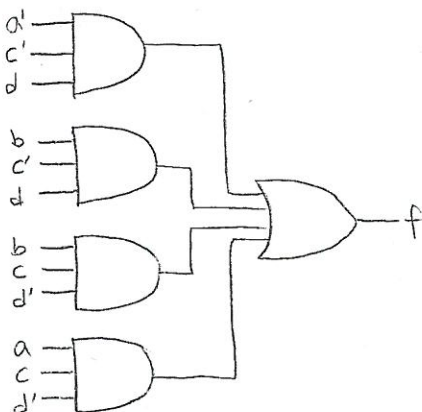
$$f(a, b, c, d) = \sum m(1, 5, 6, 10, 13, 14)$$

① f 간략화

ab \ cd	00	01	11	10
00				
01	1	1	1	
11				
10		1	1	1

$$f = a'c'd + bc'd + bcd' + acd'$$

② AND-OR 게이트 회로  
(곱의 합식)



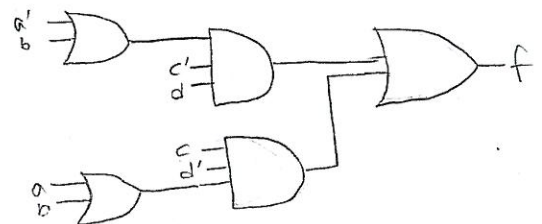
2단

5개 게이트

16개 게이트 입력

③ 최소화하여 OR-AND-OR 게이트 회로

$$f = c'd(a' + b) + cd'(a + b)$$



3단

5개 게이트

12개 게이트 입력

단수는 증가하지만  
⇒ 회로키가 줄어들수  
있음

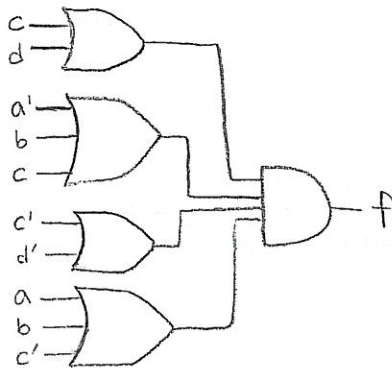
④ OR-AND 게이트 회로  
(함의 공식)

→ 카노맵에서 0값 이용

ab \ cd	00	01	11	10
00	0	0	0	0
01				0
11	0	0	0	0
10	0			

$$f' = c'd' + ab'c' + cd + a'b'c$$

$$\Rightarrow f = (c+d)(a'+b+c)(c'+d')(a+b+c')$$



2단

5개 게이트

14개 게이트 입력

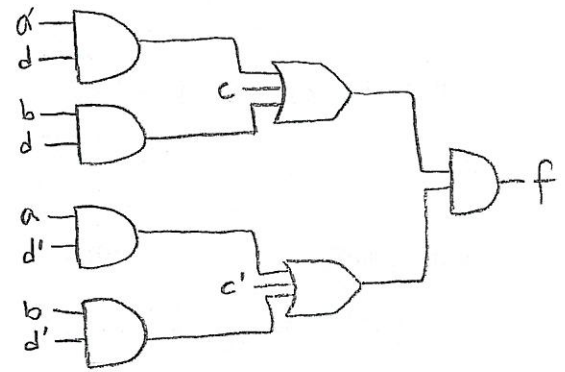
## ⑤ 몇몇 항을 곱하여 AND-OR-AND 게이트 회로

$$(x+y)(x+z) = x + yz \text{ 이항}$$

$$f = (c+d)(a'+b+c)(c'+d')(a+b+c')$$

$$\Rightarrow f = [c+d(a'+b)][c'+d'(a+b)]$$

$$= (c + a'd + bd)(c' + ad' + bd')$$



3단

7개 게이트

16개 게이트 입력

⑥ f'에 대한 식이 n단을 가지면 이 식의 보수인 f에 대한 n단 식이 된다.

OR 연산을 줄격으로 갖는 f'에 대한 n단 식을 먼저 찾고 f에 대한 식을 보수화 하면

AND 게이트 줄격을 갖는 n단 회로인 f를 실현할 수 있다.

④에서 OR 연산을 줄격으로 갖는  $f' = c'd' + ab'c' + cd + a'b'c$ 

$$= c'(d' + ab') + c(d + a'b') = c'(d' + a)(d' + b') + c(d + a')(d + b')$$

에 대한 3단 식을 찾음

f'에 대한 식을 보수화하면  $f = (c + a'd + bd)(c' + ad' + bd')$  인 ⑤에서

AND 게이트 줄격을 갖는 3단 회로인 f를 실현했다.

## 7.2 NAND 와 NOR 게이트

tb\_elec\_engineer@naver.com

Gm

- ⊕ AND 나 OR 게이트보다 일반적으로 더 빠르고 더 적은 부품을 사용  
어떤 논리 함수도 NAND 게이트로만 또는 NOR 게이트만으로 실현 가능하다.

### • NAND 게이트

(AND - NOT 게이트 ⇒ NAND 게이트)



$$F = (X_1 X_2 \cdots X_n)' = X_1' + X_2' + \cdots + X_n'$$

### • NOR 게이트

(OR - NOT 게이트 ⇒ NOR 게이트)



$$F = (X_1 + X_2 + \cdots + X_n)' = X_1' X_2' \cdots X_n'$$

⊗ ⇒ OR과 NOT을 포함하여 AND 실현하기

드모르 법칙 사용

$$XY = (X' + Y')'$$

⇒ AND와 NOT을 포함하여 OR 실현하기

$$X + Y = (X' Y')'$$

## 7.3 NAND와 NOR 게이트를 사용하여 2단 회로 설계하기

gm

AND와 OR 게이트로 구성된 2단 회로  $\rightarrow$  NAND 게이트 또는 NOR 게이트로 구성된 회로 $F = (F')'$  또는 드모건 법칙을 적용

$$\hookrightarrow (X_1 + X_2 + \dots + X_n)' = X_1' X_2' X_3' \dots X_n'$$

$$(X_1 X_2 \dots X_n)' = X_1' + X_2' + \dots + X_n'$$

(ex) 최소논리곱의 합 형태를 다른 여러가지 2단 형태로 변환

$$F = A + BC' + B'CD \quad \rightarrow \text{AND-OR}$$

$$= [(A + BC' + B'CD)']'$$

$$= [A' \cdot (BC')' \cdot (B'CD)']' \quad \rightarrow \text{NAND-NAND}$$

$$= [A' \cdot (B' + C) \cdot (B + C' + D)']' \quad \rightarrow \text{OR-NAND}$$

$$= A + (B' + C)' + (B + C' + D)' \quad \rightarrow \text{NOR-OR}$$

최소 논리합의 곱 형태를 다른 여러가지 2단 형태로 변환

$$F = (A + B + C)(A + B' + C')(A + C' + D) \quad \rightarrow \text{OR-AND}$$

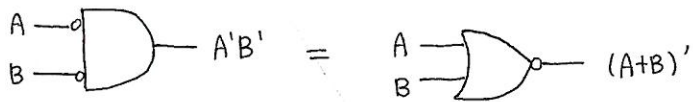
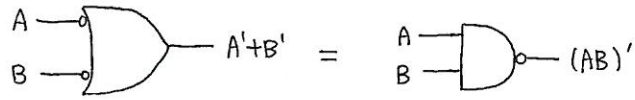
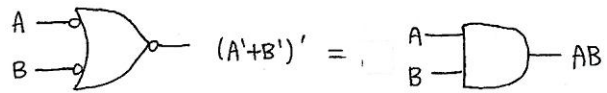
$$= \{[(A + B + C)(A + B' + C')(A + C' + D)]'\}'$$

$$= [(A + B + C)' + (A + B' + C')' + (A + C' + D)']' \quad \rightarrow \text{NOR-NOR}$$

$$= (A'B'C' + A'BC + A'CD) \quad \rightarrow \text{AND-NOR}$$

$$= (A'B'C')' \cdot (A'BC)' \cdot (A'CD)' \quad \rightarrow \text{NAND-AND}$$

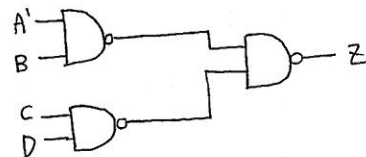
0 (버블) : 출력과 입력에 모두 사용 가능



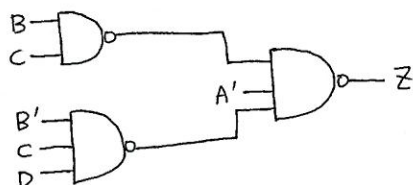
예제) NAND로만 이루어진 회로 바꾸기



$\Leftrightarrow$



$\Leftrightarrow$



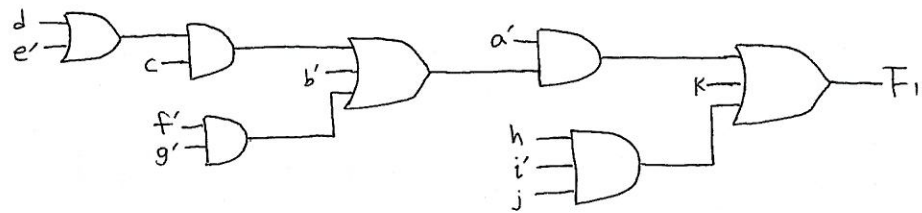


7-4 다단 NAND와 NOR 게이트회로 설계

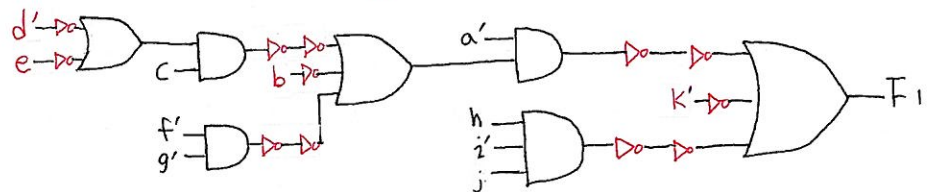
gm

예제)  $F_1 = a' [b' + c(d+e') + f'g'] + hi'j + k$

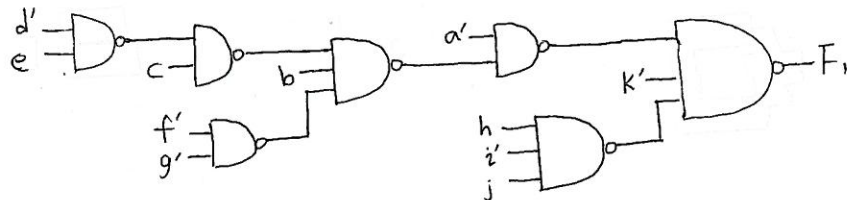
AND-OR 회로



AND와 OR  
사이에 -> 추가



NAND 회로



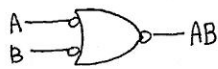
⇒ AND 와 OR 게이트가 서로 번갈아가며 나타나 깔끔하게 바꿀 수 있음

gm

드모르간 법칙을 이용하여 다른 형태로 게이트 표현하기

① AND

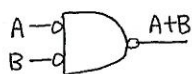
$$AB = (A' + B')'$$



$$= \text{NAND gate with inputs A and B, output AB}$$

② OR

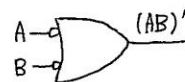
$$A + B = (A'B')'$$



$$= \text{NAND gate with inputs A and B, output A+B}$$

③ NAND

$$(AB)' = A' + B'$$



$$= \text{NAND gate with inputs A and B, output (AB)'}$$

④ NOR

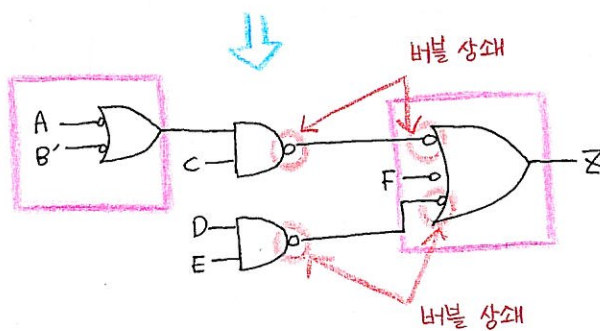
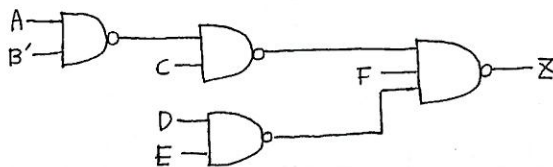
$$(A+B)' = A'B'$$



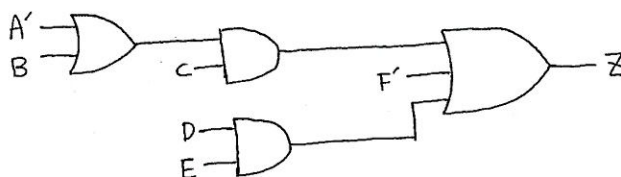
$$= \text{NAND gate with inputs A and B, output (A+B)'}$$

예제) 다른 형태의 NAND 게이트로 바꿔 AND-OR 회로로 변환하기

NAND 게이트 회로



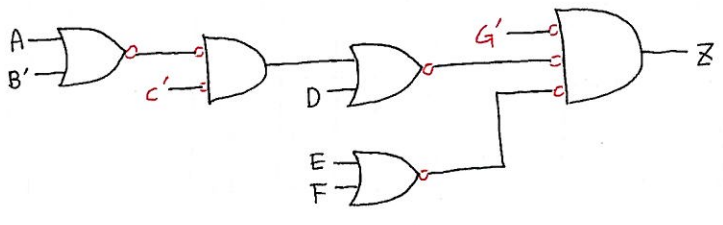
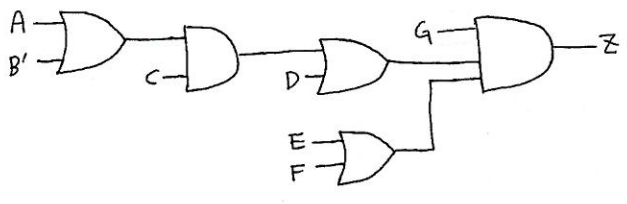
AND-OR 회로



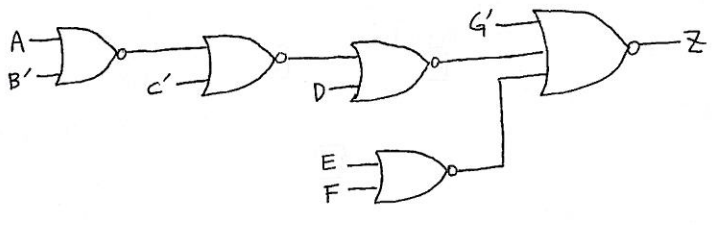


예제) OR와 AND 게이트 회로를 NOR 게이트로 변환하기

OR와 AND  
게이트 회로

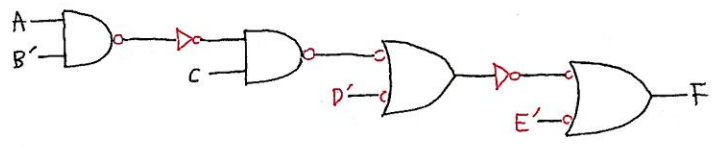
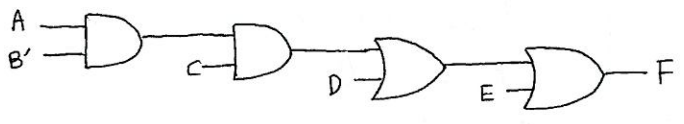


NOR 게이트 회로

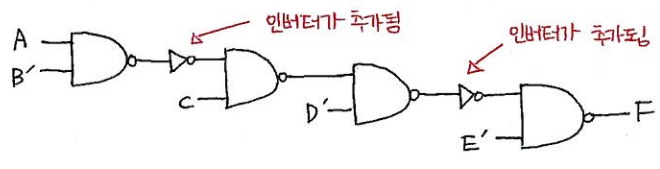


예제) AND-OR 회로를 NAND 게이트로 변환하기

AND-OR 회로



NAND 게이트 회로



gm

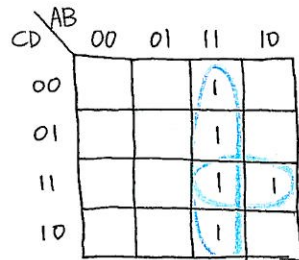
→ 입력은 서로 공유하지만 출력이 다른 회로에 대한 설계

예제)  $F_1(A, B, C, D) = \sum m(11, 12, 13, 14, 15)$

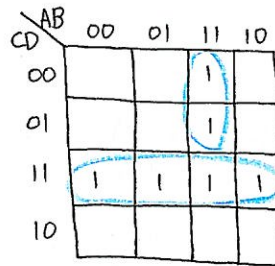
$F_2(A, B, C, D) = \sum m(3, 7, 11, 12, 13, 15)$

$F_3(A, B, C, D) = \sum m(3, 7, 12, 13, 14, 15)$

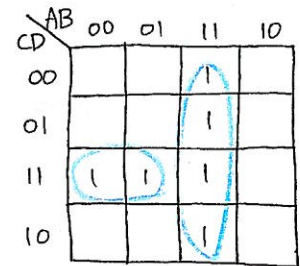
카노맵 이용 →



$F_1 = AB + ACD$

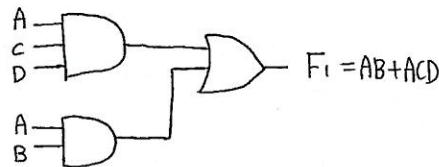


$F_2 = ABC' + CD$

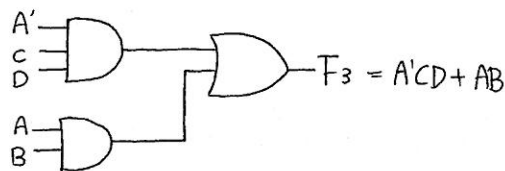
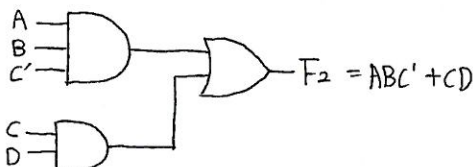


$F_3 = A'CD + AB$

결과회로 →



9개 게이트  
⇒ 21개 게이트 입력



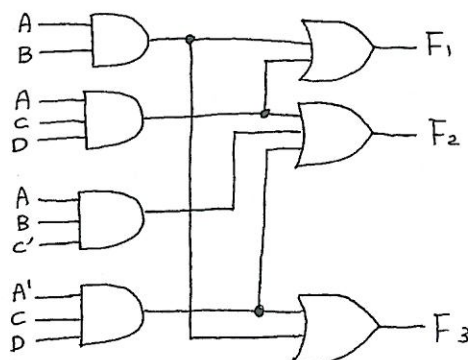
회로 간략화 하기 →

AB 항을  $F_1$ 과  $F_3$ 이 공유

$F_2 = ABC' + CD = ABC' + ACD + A'CD$  로 바꿀 수 있으며

$F_2$ 의  $ACD$  항은  $F_1$ 과  $A'CD$ 는  $F_3$ 과 공유

다중 출력 회로 →



⇒ 17개 게이트  
18개 게이트 입력  
축소된 회로를 만들 수 있음

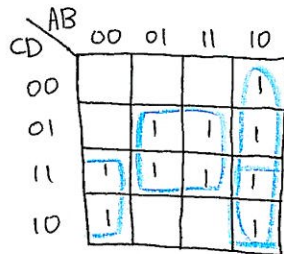
(gm)

예제)  $F_1 = \sum m(2, 3, 5, 7, 8, 9, 10, 11, 13, 15)$

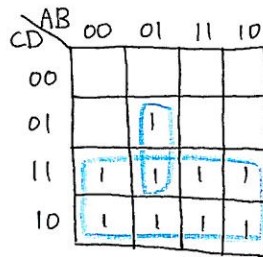
$F_2 = \sum m(2, 3, 5, 6, 7, 10, 11, 14, 15)$

$F_3 = \sum m(6, 7, 8, 9, 13, 14, 15)$

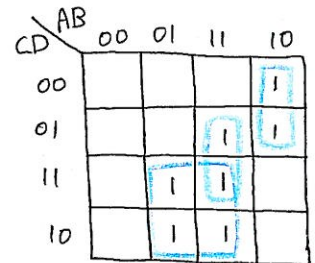
카노맵 이용 →



$F_1 = BD + B'C + AB'$



$F_2 = C + A'BD$



$F_3 = BC + AB'C' + ABD$

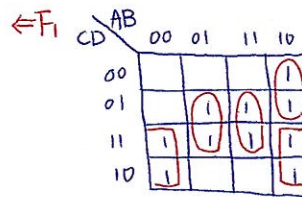
회로 간략화 하기 →

$F_1 = BD + B'C + AB'$

$= A'BD + ABD + B'C + AB'C'$

$F_2 = C + A'BD$

$F_3 = BC + AB'C' + ABD$



공통항을 만들기 위해 조합

다중 출력 회로 →

