

## Combinational Logic Design p.105

### 2장 : 논리 연산들에 대한 논리 게이트들

경제적인 설계를 위한 부울 함수 단순화 방법에 대해 이해하였다.

### 3장 : 2장의 이해를 바탕으로 조합 회로의 설계 과정을 배운다.

기본 함수 블록들(디코더, 인코더..)의 기능을 배운다.

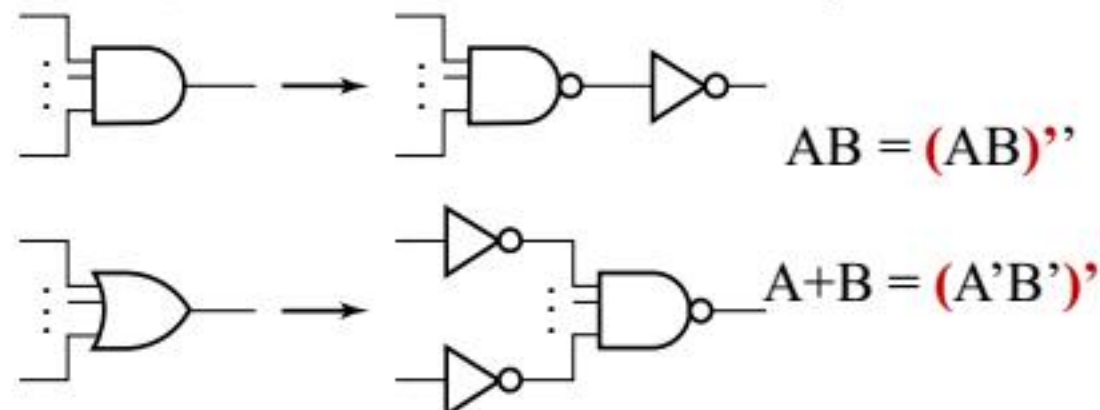
1

1

## Technology Mapping

110

- NAND, NOR는 현재 집적회로 기술에서 가장 간단하고 가장 빠른 게이트이다.
- 그리고 universal Gate 이다. (p.40,42 참고)
- AND, OR, NOT 로 구성된 회로 ➔ NAND gate로 매핑



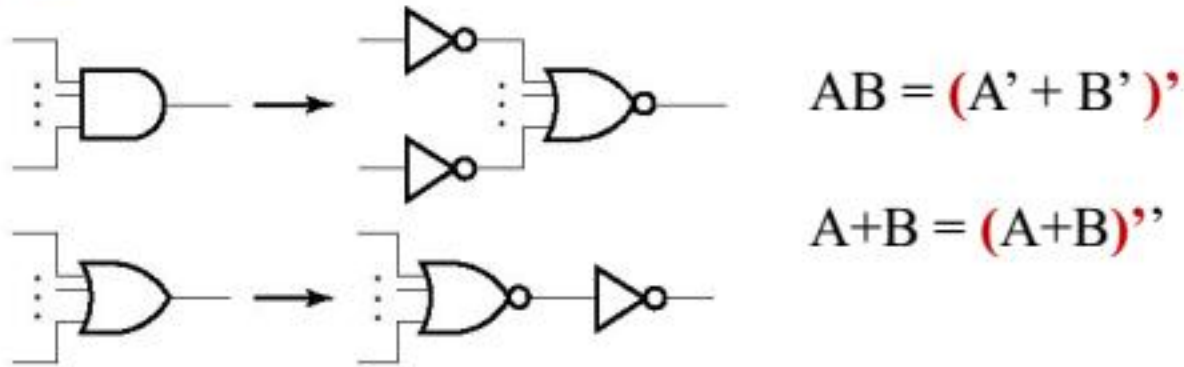
3

3

## Technology Mapping(cont'd)

p. 111

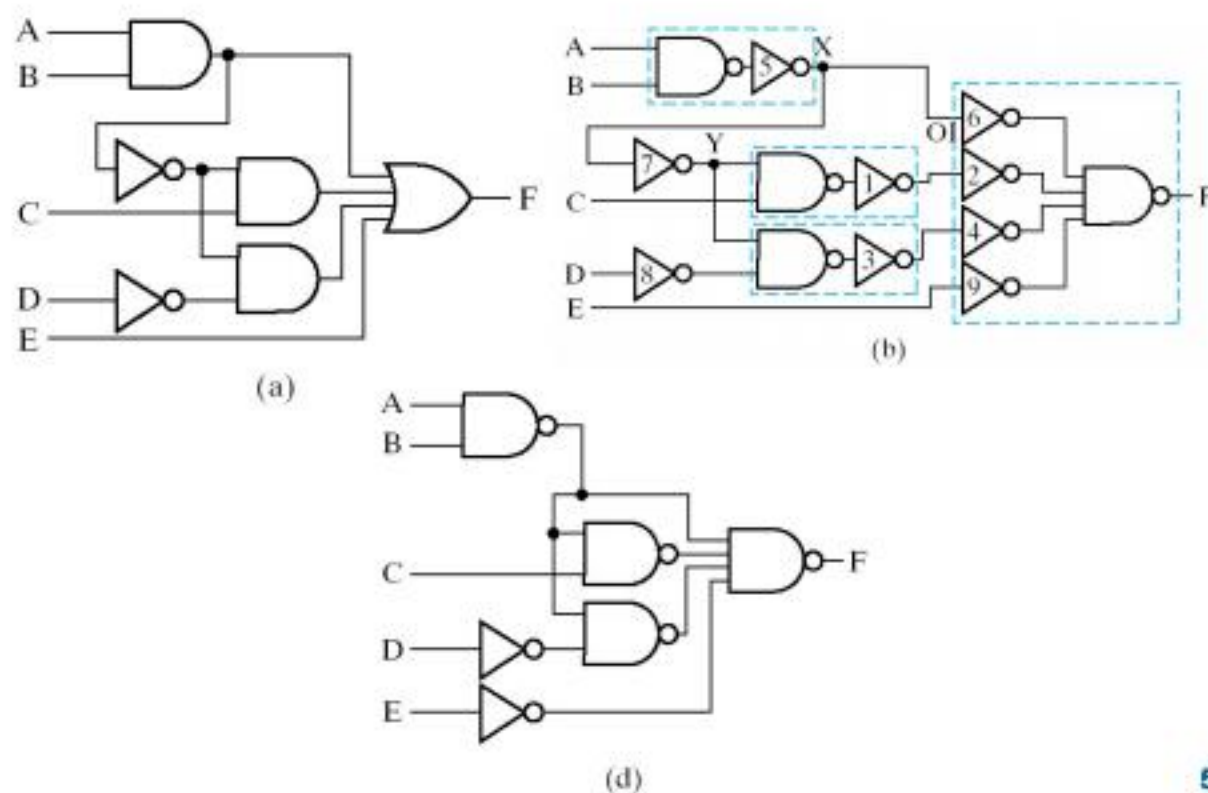
■ AND, OR, NOT 로 구성된 회로 → NOR gate로 매핑



4

4

$F = AB + (AB)'C + (AB)'D' + E$  (NAND구현) p. 112

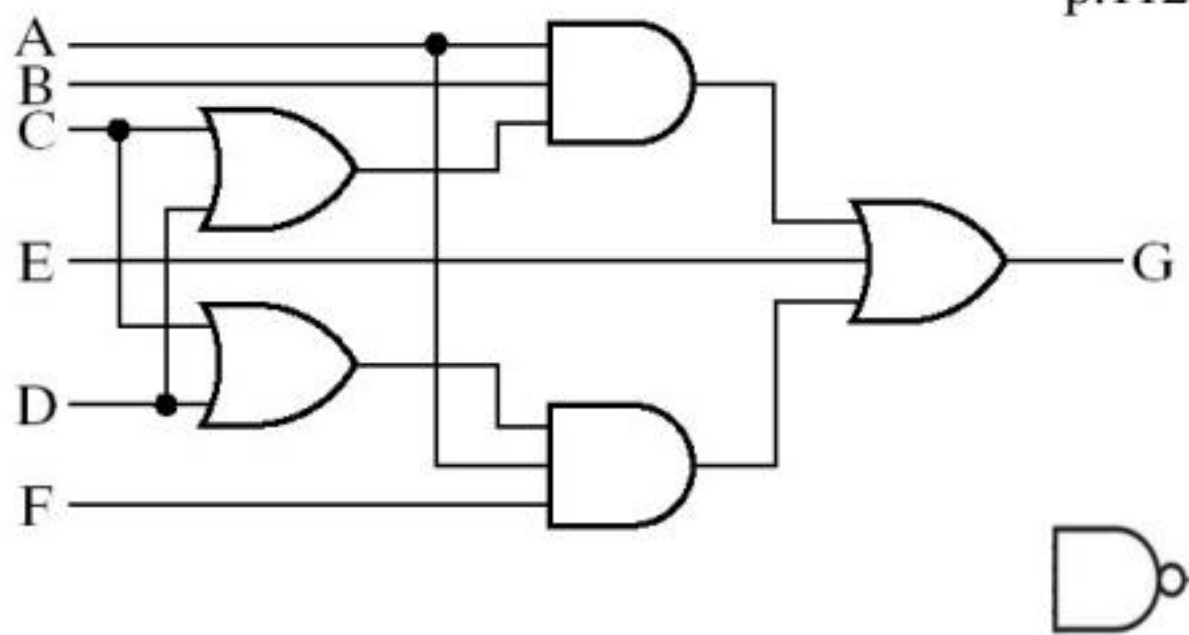


5

5

NAND mapping : 아래 논리회로도를 모두 nand로 바꾸시오.

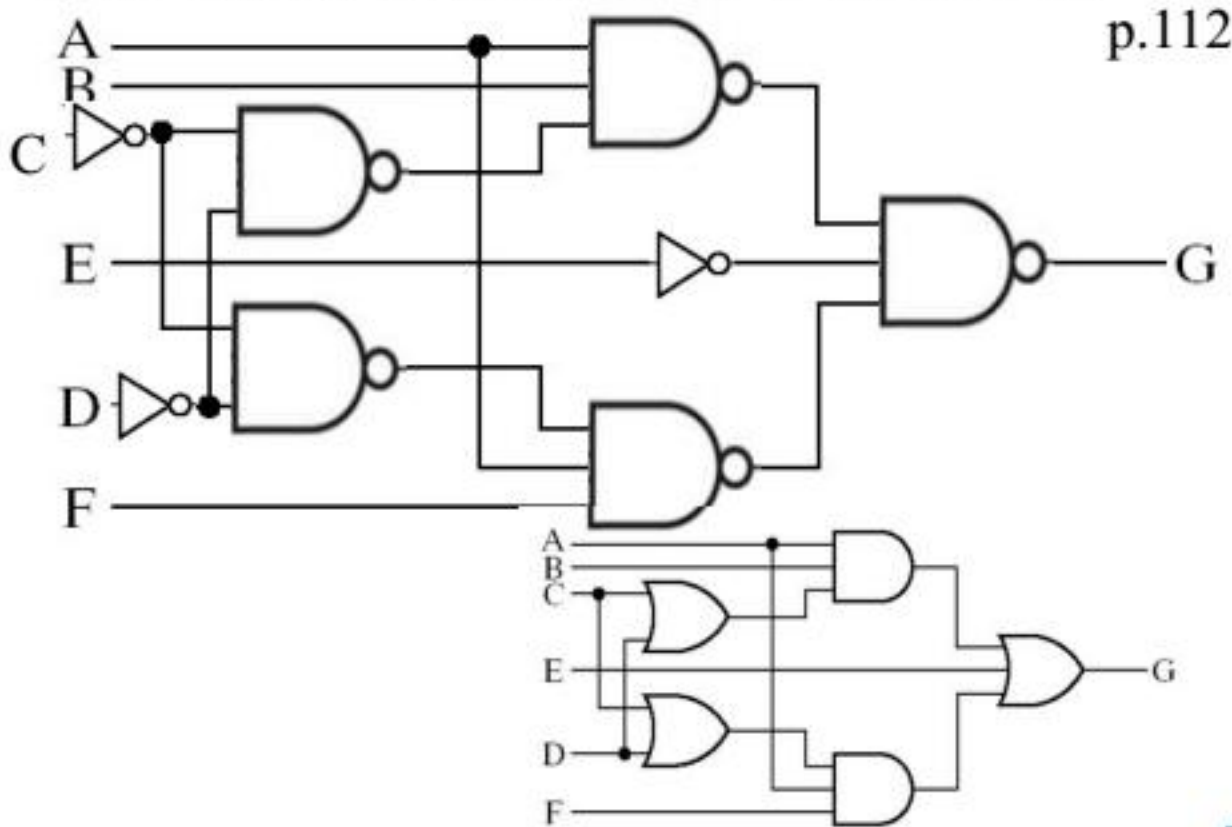
p.112



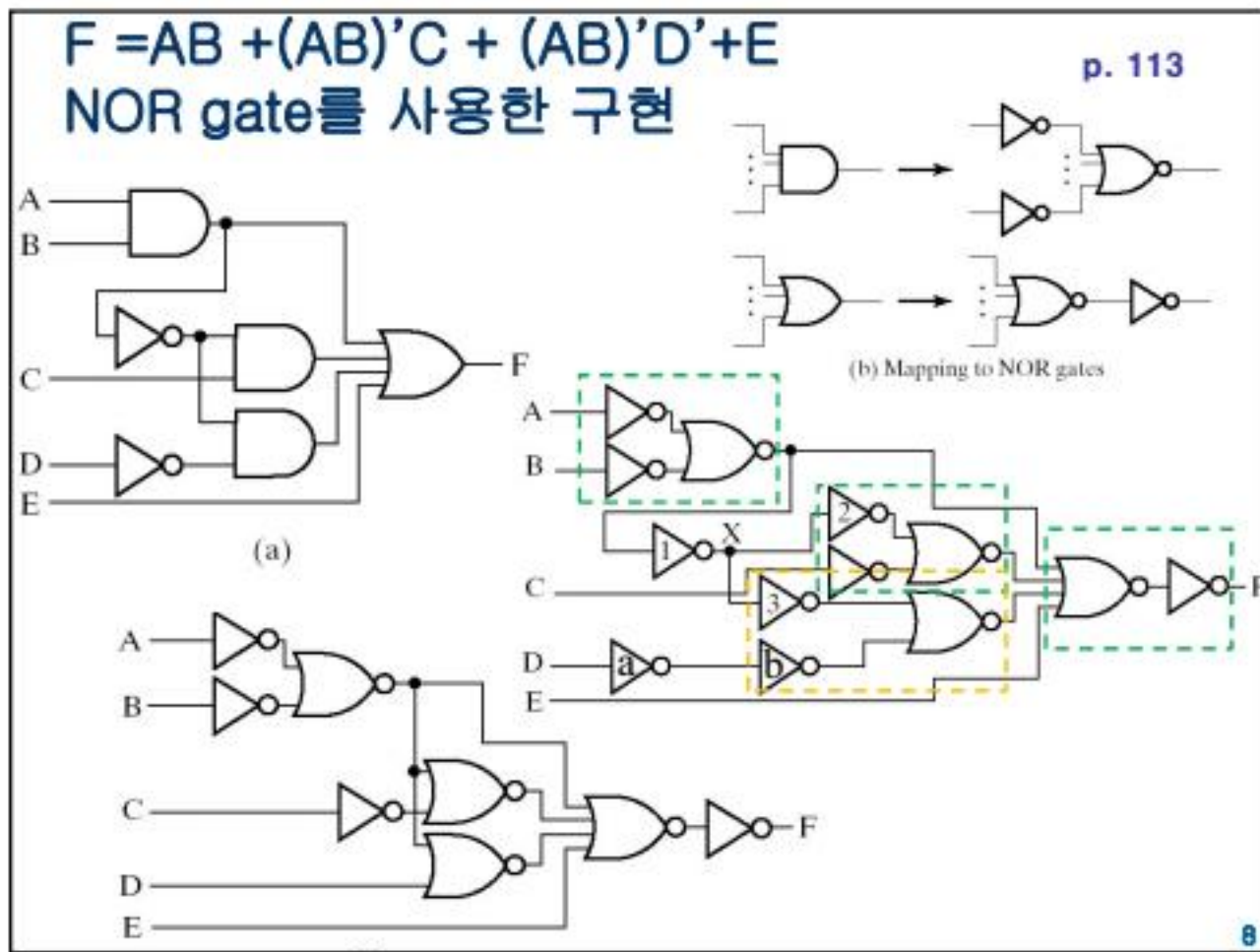
6

NAND mapping : 아래 논리회로도를 모두 nand로 바꾼다.

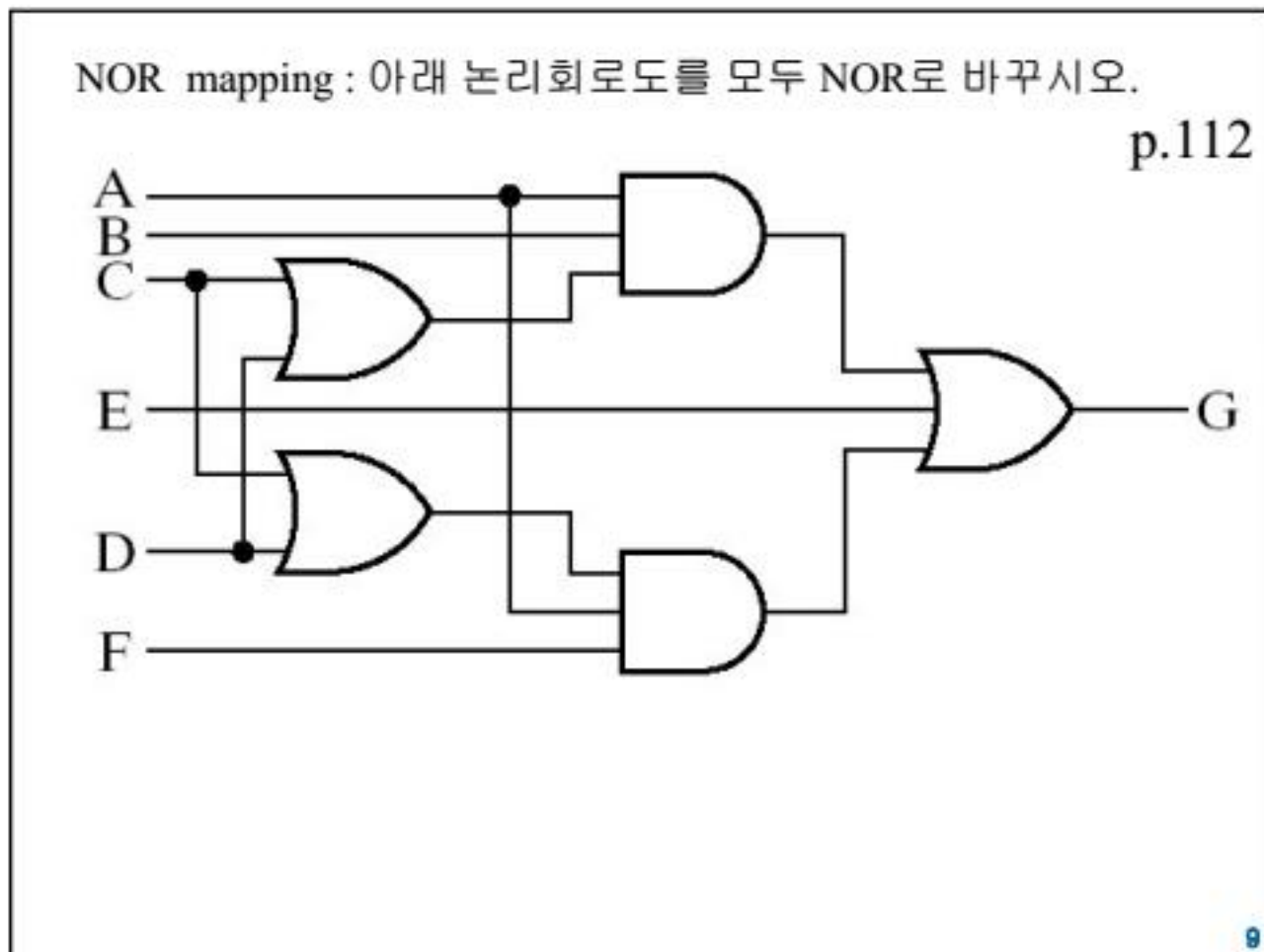
p.112



7



8



9

## NAND mapping / NOR mapping p.113

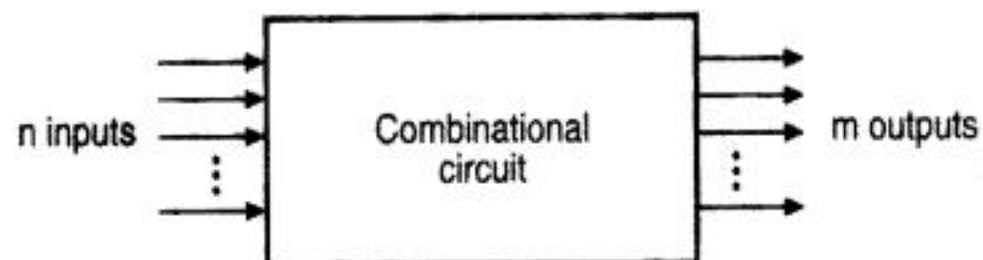
- 게이트 지연시간의 관점
  - 곱의 합  $\rightarrow$  NAND mapping이 유리
  - 합의 곱  $\rightarrow$  NOR mapping이 유리
- 이 외에 다양한 관점에서 기술적 고려를 해야 한다.

10

10

## Combinational circuit

- 조합회로
  - 입력 변수들만의 논리 연산에 의한 출력
  - $F(x, y, z)$
  - $G(I_1, I_2, I_3, I_4, I_5, \dots, I_n)$



$F(\text{입력변수들}) = \text{출력}$

11

11



## 디지털 설계 과정 p.106

1. Specification(구체화, 스펙화)
  - 회로의 기능 정의
  - 입력, 출력의 이름과 symbol 정하기
2. 수식화(Formulation)
  - the truth table, initial Boolean equations
3. Optimization : 2-level gates 최적화
4. Technology Mapping
  - **활용되는 함수적 블록들을 사용한 최적화**, 다단계 최적화 : 공유할 수 있는 항을 이용, 비용절감
  - 구현기술에 따른 최적화 설계 : 최대팬아웃, 허용지연시간 등
5. Verification (검증)
  - simulation

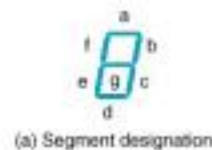
12

12

## BCD-to-7-segment converter(decoder) p.143

- BCD-to-7-세그먼트 디코더 : 전자계산기, 디지털시계
- Specification
  - 입력 4개(A,B,C,D)
  - 출력 7개(a~g)

### 진리표



BCD Input				Seven-Seg						
A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1
All other inputs				0	0	0	0	0	0	0

- 7개의 간략화된 부울 함수
- 각자 해보기

발생하지 않는다.  
선택 1. 무정의 조건  
선택 2. 모두 0으로 출력

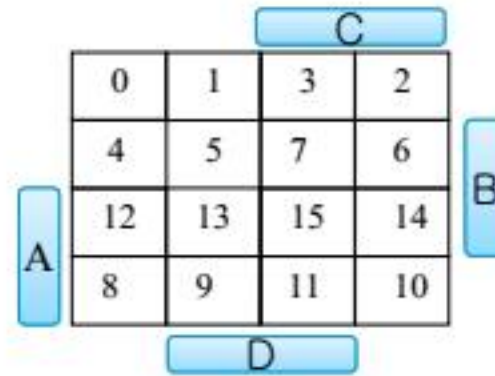
13

13

## BCD-to-7-segment converter(decoder) p.143

### 진리표

BCD Input				Seven-Segment Decoder						
A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1
All other inputs				0	0	0	0	0	0	0



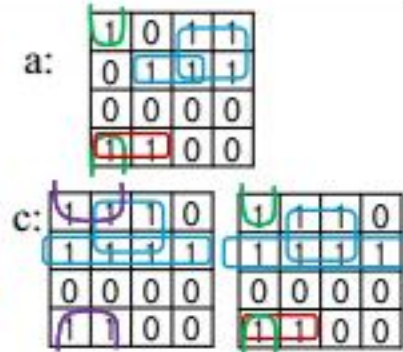
14

14

## BCD-to-7-segment decoder

### 27개의 And gate, 7개의 Or gate

- $a = A'C + A'BD + B'C'D' + AB'C'$
- $b = A'B' + A'C'D' + A'CD + AB'C'$
- $c = A'B + A'D + B'C'D' + AB'C'$
- $d = A'CD' + A'B'C + B'C'D' + A'BC' + A'BC'D$
- $e = A'CD' + B'C'D'$
- $f = A'BC' + A'C'D' + A'BD' + AB'C'$
- $g = A'CD' + A'B'C + A'BC' + AB'C'$



### 6 개의 곱항을 공유 : And gate 14개로 간략화

- 또다른 최적화 : e는 d의 부분항이다.  $B'C'D' + AB'C'$ 는 부분항이다....

15

15

## 연습

- GrayCode -to-BCD converter를 설계하시오.

단) 표에 지정된 10개의 Gray코드

이외에는 발생하지 않는다.

입력변수 ?개 : A,B... 로 열거

출력변수 ?개 : a,b,...로 열거

				C
	0	1	3	2
	4	5	7	6
A	12	13	15	14
	8	9	11	10
				D
				B

	Gray	BCD
0	0000	0000
1	0100	0001
2	0101	0010
3	0111	0011
4	0110	0100
5	0010	0101
6	0011	0110
7	0001	0111
8	1001	1000
9	1000	1001

16

16

## 연습(수정)

- GrayCode -to-BCD converter를 설계하시오.

입력변수 ?개 : A,B... 로 열거

출력변수 ?개 : a,b,...로 열거

	Gray	BCD
0	000	0000
1	100	0001
2	101	0010
3	111	0011
4	110	0100
5	010	0101
6	011	0110
7	001	0111

17

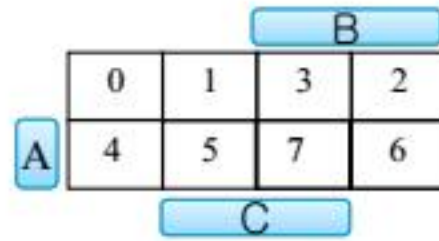
17



## 연습(수정)

- GrayCode -to-BCD converter를 설계하시오.

입력변수 ?개 : A,B... 로 열거  
출력변수 ?개 : a,b,...로 열거



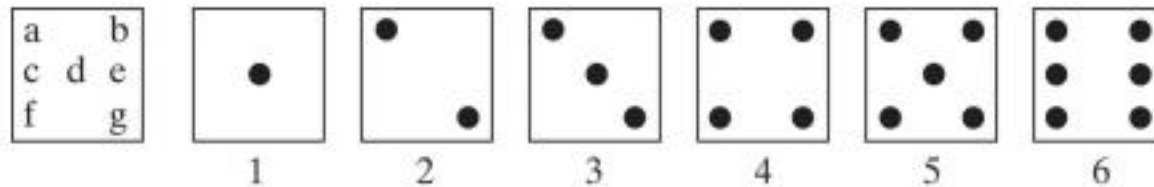
	Gray	BCD
0	000	0000
1	100	0001
2	101	0010
3	111	0011
4	110	0100
5	010	0101
6	011	0110
7	001	0111

18

18

## 연습 p.178그림 참고

- 1~6의 이진수가 입력되면 아래 모양과 같이 주사위의 각 세그먼트가 출력되도록 설계하시오.



- 예) (001)이 입력되면 d만 1로 출력되고 나머지 출력은 0이다.
- 입력변수 : 3개 X,Y,Z
- 출력변수 : 7개 a~g
- 단) 1~6 이외의 이진수가 입력되면 모두 0으로 출력된다.

19

19

## 정리

- 조합회로 설계
  - 스펙작성
  - 수식화
    - 진리표 → 부울함수
  - 최적화
    - 2-level → 논리 회로도
- 모든 부울함수는 NAND 게이트로 구현할 수 있다.
- 모든 부울함수는 NOR 게이트로 구현할 수 있다.
- NAND NOR mapping

20

## Combinational Logic Design p.105

### 2장 : 논리 연산들에 대한 논리 게이트들

경제적인 설계를 위한 부울 함수 단순화 방법에 대해 이해하였다.

### 3장 : 2장의 이해를 바탕으로 조합 회로의 설계 과정을 배운다.

기본 함수 블록들(디코더, 인코더..)의 기능을 배운다.

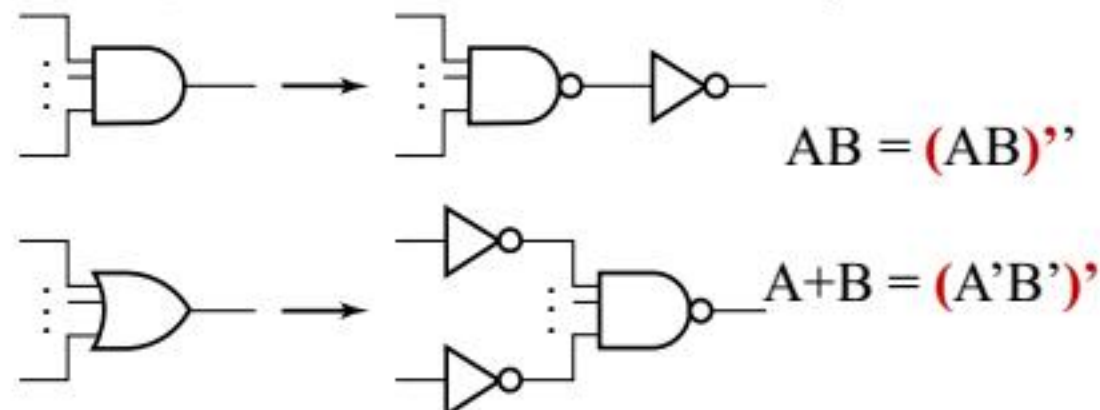
1

1

## Technology Mapping

110

- NAND, NOR는 현재 집적회로 기술에서 가장 간단하고 가장 빠른 게이트이다.
- 그리고 universal Gate 이다. (p.40,42 참고)
- AND, OR, NOT 로 구성된 회로 ➔ NAND gate로 매핑



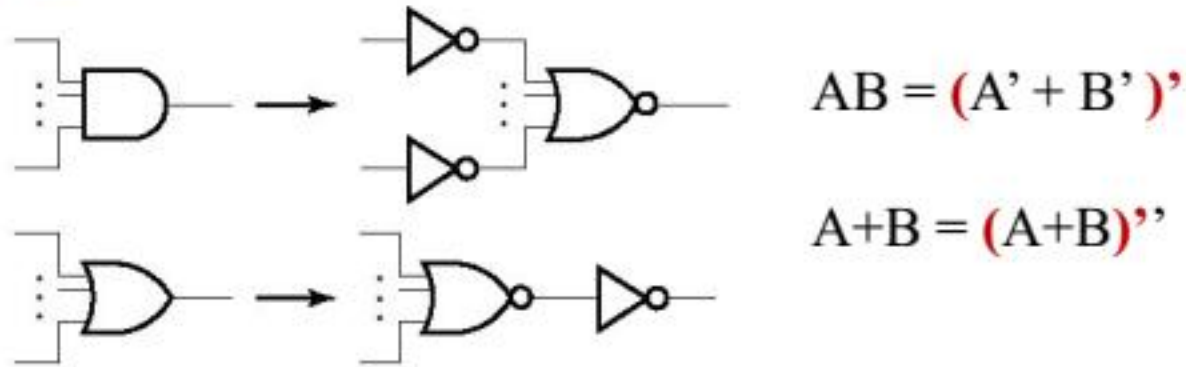
3

3

## Technology Mapping(cont'd)

p. 111

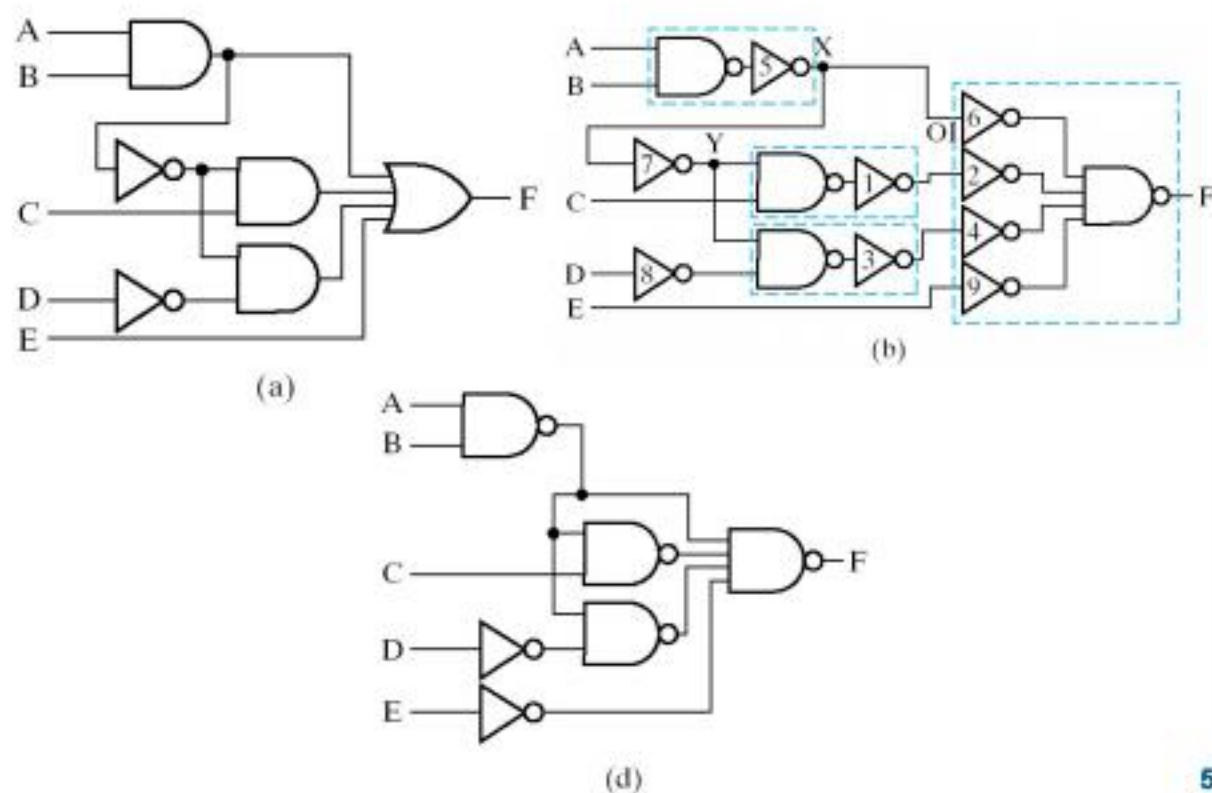
■ AND, OR, NOT 로 구성된 회로 → NOR gate로 매핑



4

4

$F = AB + (AB)'C + (AB)'D' + E$  (NAND구현) p. 112



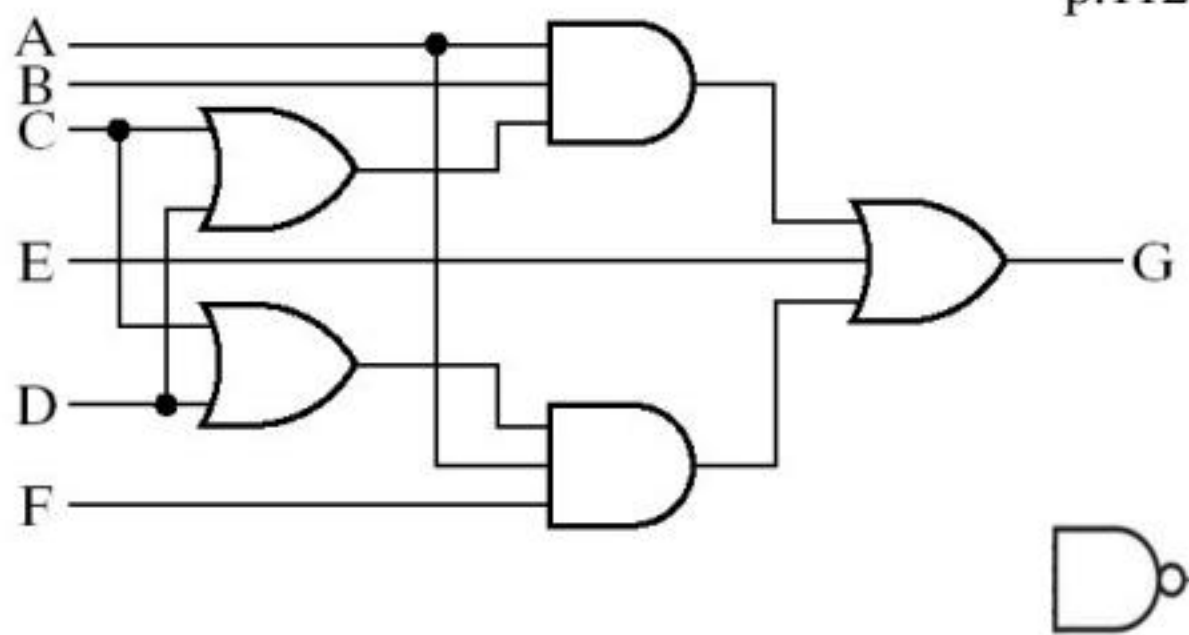
5

5



NAND mapping : 아래 논리회로도를 모두 nand로 바꾸시오.

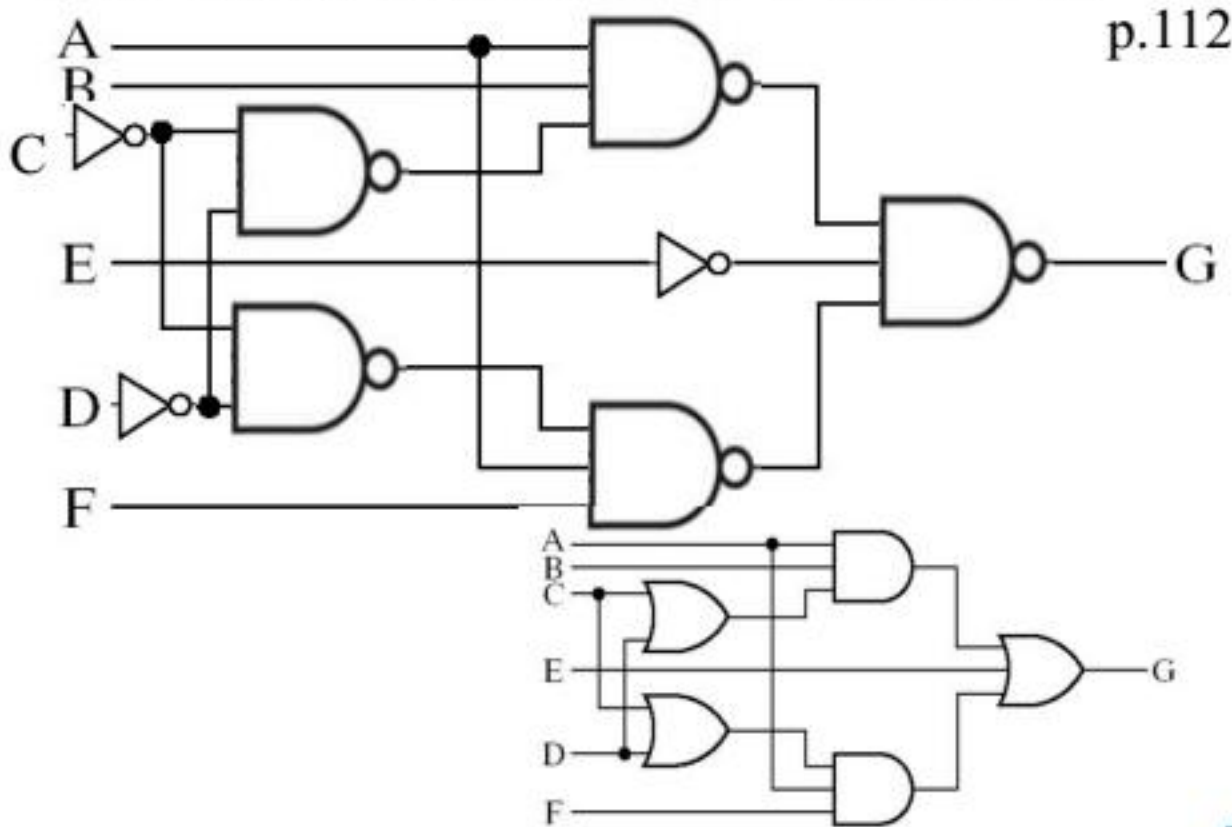
p.112



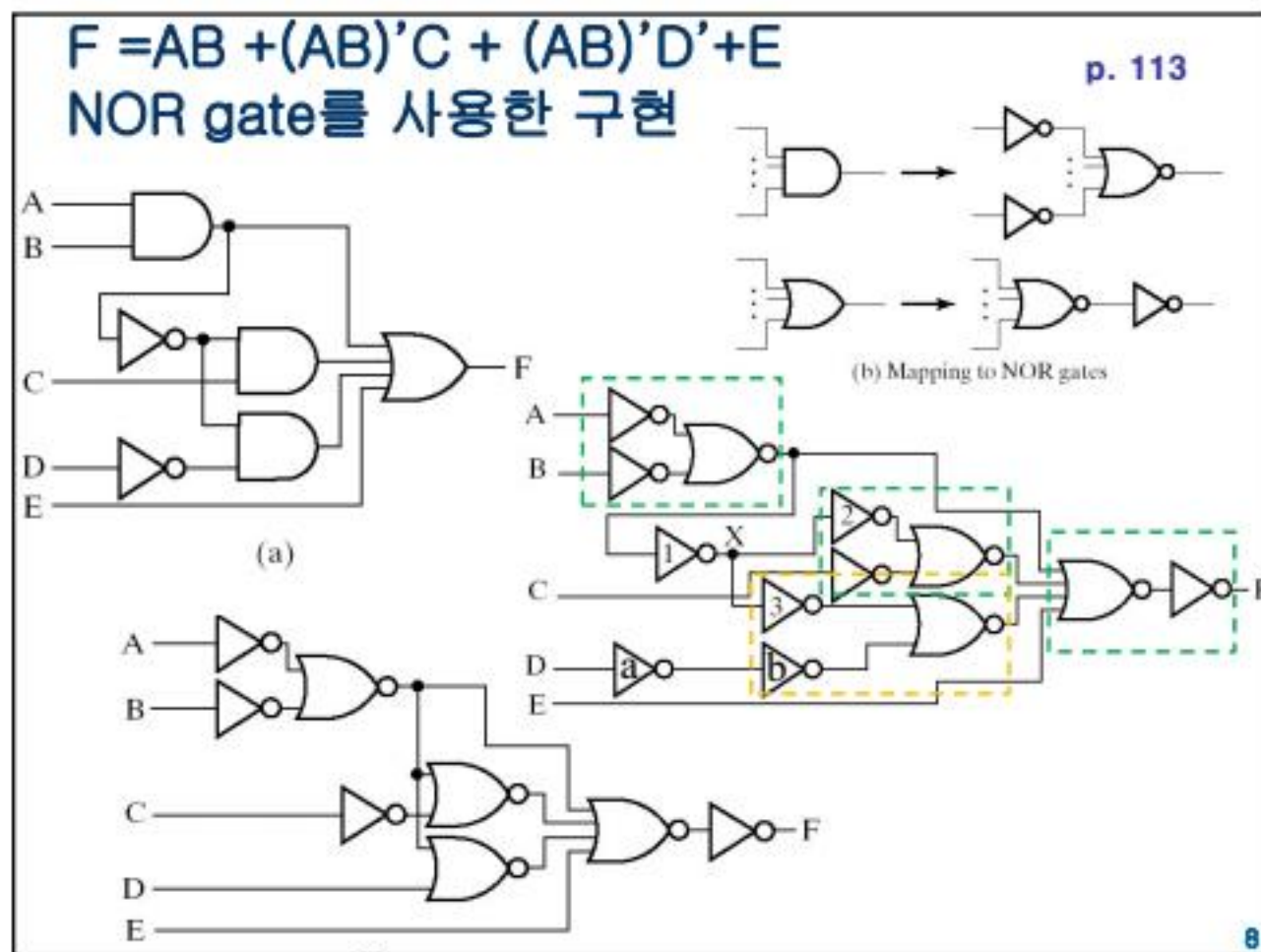
6

NAND mapping : 아래 논리회로도를 모두 nand로 바꾼다.

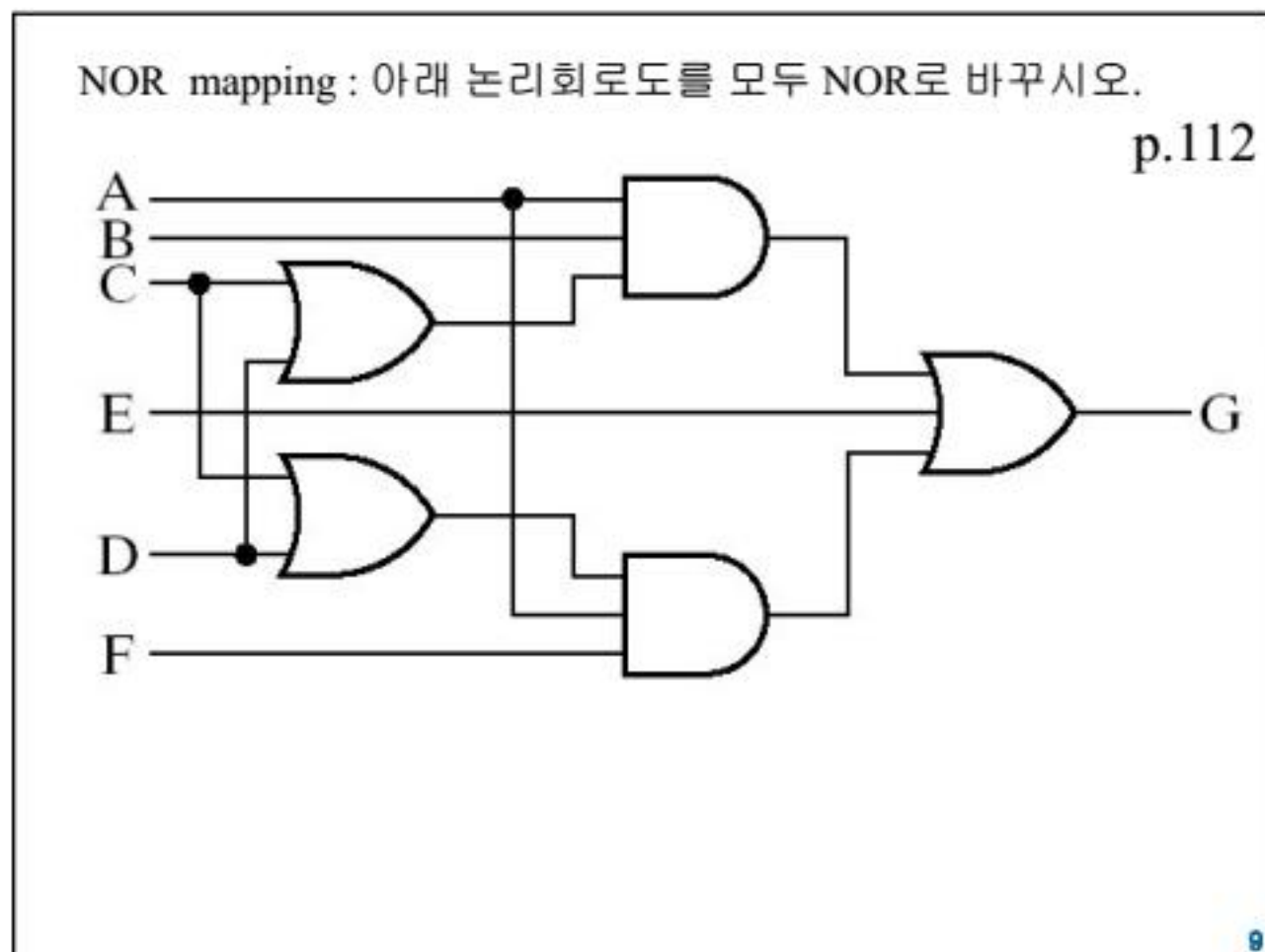
p.112



7



8



9

## NAND mapping / NOR mapping p.113

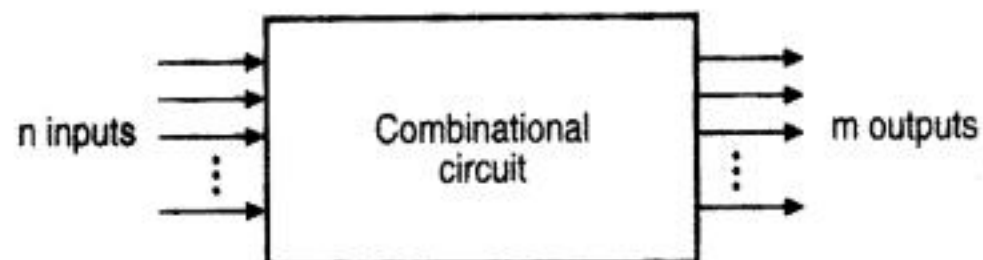
- 게이트 지연시간의 관점
  - 곱의 합  $\rightarrow$  NAND mapping이 유리
  - 합의 곱  $\rightarrow$  NOR mapping이 유리
- 이 외에 다양한 관점에서 기술적 고려를 해야 한다.

10

10

## Combinational circuit

- 조합회로
  - 입력 변수들만의 논리 연산에 의한 출력
  - $F(x, y, z)$
  - $G(I_1, I_2, I_3, I_4, I_5, \dots, I_n)$



$F(\text{입력변수들}) = \text{출력}$

11

11

## 디지털 설계 과정 p.106

1. Specification(구체화, 스펙화)
  - 회로의 기능 정의
  - 입력, 출력의 이름과 symbol 정하기
2. 수식화(Formulation)
  - the truth table, initial Boolean equations
3. Optimization : 2-level gates 최적화
4. Technology Mapping
  - **활용되는 함수적 블록들을 사용한 최적화**, 다단계 최적화 : 공유할 수 있는 항을 이용, 비용절감
  - 구현기술에 따른 최적화 설계 : 최대팬아웃, 허용지연시간 등
5. Verification (검증)
  - simulation

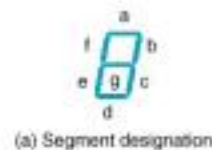
12

12

## BCD-to-7-segment converter(decoder) p.143

- BCD-to-7-세그먼트 디코더 : 전자계산기, 디지털시계
- Specification
  - 입력 4개(A,B,C,D)
  - 출력 7개(a~g)

### 진리표



BCD Input				Seven-Seg						
A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1
All other inputs				0	0	0	0	0	0	0

- 7개의 간략화된 부울 함수
- 각자 해보기

발생하지 않는다.  
선택 1. 무정의 조건  
선택 2. 모두 0으로 출력

13

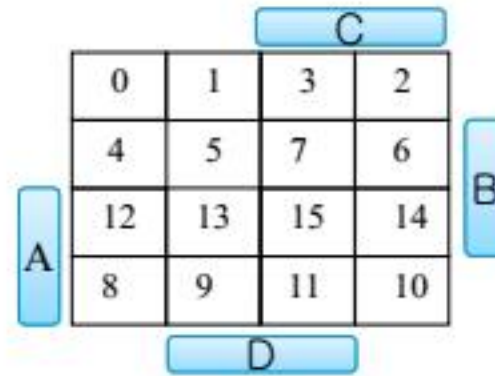
13



## BCD-to-7-segment converter(decoder) p.143

### 진리표

BCD Input				Seven-Segment Decoder						
A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1
All other inputs				0	0	0	0	0	0	0



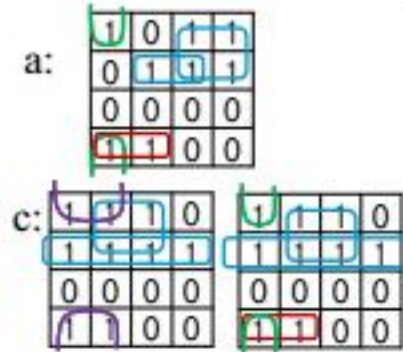
14

14

## BCD-to-7-segment decoder

### 27개의 And gate, 7개의 Or gate

- $a = A'C + A'BD + B'C'D' + AB'C'$
- $b = A'B' + A'C'D' + A'CD + AB'C'$
- $c = A'B + A'D + B'C'D' + AB'C'$
- $d = A'CD' + A'B'C + B'C'D' + A'BC' + A'BC'D$
- $e = A'CD' + B'C'D'$
- $f = A'BC' + A'C'D' + A'BD' + AB'C'$
- $g = A'CD' + A'B'C + A'BC' + AB'C'$



### 6 개의 곱항을 공유 : And gate 14개로 간략화

- 또다른 최적화 : e는 d의 부분항이다.  $B'C'D' + AB'C'$ 는 부분항이다....

15

15

## 연습

- GrayCode -to-BCD converter를 설계하시오.

단) 표에 지정된 10개의 Gray코드

이외에는 발생하지 않는다.

입력변수 ?개 : A,B... 로 열거

출력변수 ?개 : a,b,...로 열거

				C
	0	1	3	2
	4	5	7	6
A	12	13	15	14
	8	9	11	10
				D
				B

	Gray	BCD
0	0000	0000
1	0100	0001
2	0101	0010
3	0111	0011
4	0110	0100
5	0010	0101
6	0011	0110
7	0001	0111
8	1001	1000
9	1000	1001

16

16

## 연습(수정)

- GrayCode -to-BCD converter를 설계하시오.

입력변수 ?개 : A,B... 로 열거

출력변수 ?개 : a,b,...로 열거

	Gray	BCD
0	000	0000
1	100	0001
2	101	0010
3	111	0011
4	110	0100
5	010	0101
6	011	0110
7	001	0111

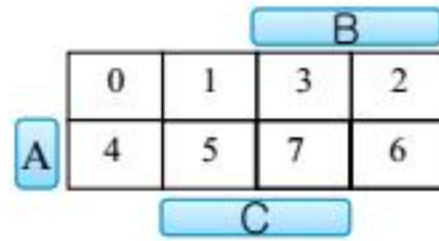
17

17

## 연습(수정)

- GrayCode -to-BCD converter를 설계하시오.

입력변수 ?개 : A,B... 로 열거  
출력변수 ?개 : a,b,...로 열거



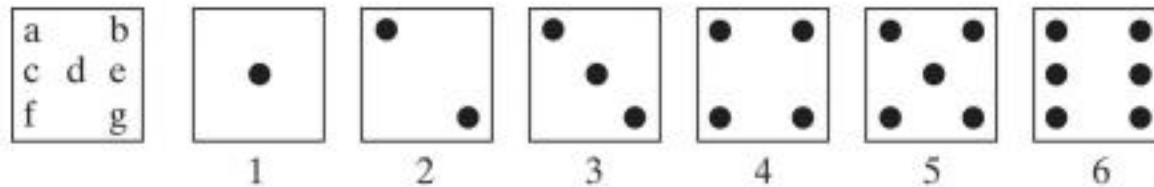
	Gray	BCD
0	000	0000
1	100	0001
2	101	0010
3	111	0011
4	110	0100
5	010	0101
6	011	0110
7	001	0111

18

18

## 연습 p.178그림 참고

- 1~6의 이진수가 입력되면 아래 모양과 같이 주사위의 각 세그먼트가 출력되도록 설계하시오.



- 예) (001)이 입력되면 d만 1로 출력되고 나머지 출력은 0이다.
- 입력변수 : 3개 X,Y,Z
- 출력변수 : 7개 a~g
- 단) 1~6 이외의 이진수가 입력되면 모두 0으로 출력된다.

19

19

## 정리

- 조합회로 설계
  - 스펙작성
  - 수식화
    - 진리표 → 부울함수
  - 최적화
    - 2-level → 논리 회로도
- 모든 부울함수는 NAND 게이트로 구현할 수 있다.
- 모든 부울함수는 NOR 게이트로 구현할 수 있다.
- NAND NOR mapping

20









