

Combinational Logic Design p.128

일반적으로 컴퓨터 설계에서 매우 유용하게 사용되는 조합회로 블록들에 대해 알아본다.

Decoder, Encoder, Multiplexer

1

1

Rudimentary Logic Functions p.114 기본논리함수

- 값고정(value fixing)
- 전달(transferring)
- 보수화(inverting)
- 유효화(enabling).. 나중에

■ 단일 변수 X에 대한 함수들

- 값 고정(0, 1)
- 전달(X)
- 반전(X') : not gate

X	F = 0	F = X	F = \bar{X}	F = 1
0	0	0	1	1
1	0	1	0	1

2

2

1

p.115
(양의논리에서)

X	F = 0	F = X	F = \bar{X}	F = 1
0	0	0	1	1
1	0	1	0	1

1(+)에 해당하는 전압

V_{CC} or V_{DD}

1 ——— F = 1



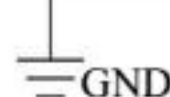
F = 1

X ——— F = X

(c)

0 ——— F = 0

F = 0



X ——— F = \bar{X}

(d)

(a)

0(-)에 해당하는 전압

(b)

3

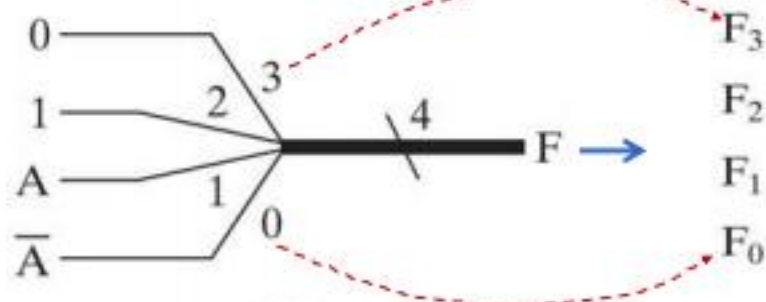
3

다중 비트 함수(도식화 방법) p.116

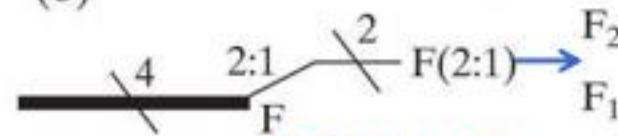
0 ——— F_3
1 ——— F_2
A ——— F_1
 \bar{A} ——— F_0

(a) 함수정의

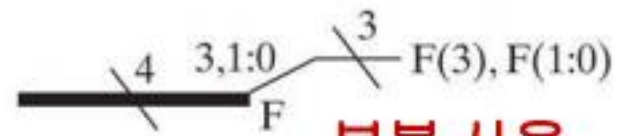
정의 후 (b)와 같이
도식화할 수 있다.



(b)



(c) 부분 사용



(d) 부분 사용

5

5

2

유효화(Enabling)

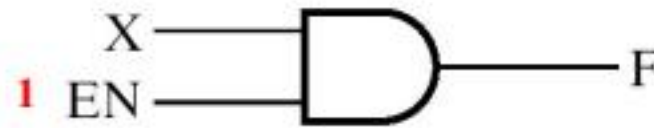
p.118

- Enabling : 입력이 출력으로 전달된다 는 의미

– an input signal → an output

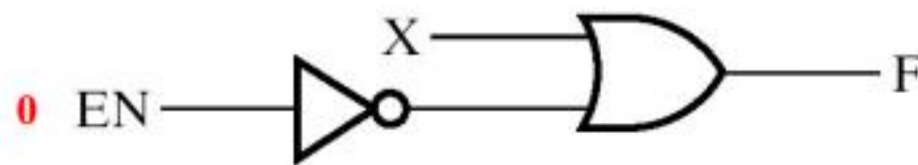
- 그림 a에서 if EN =?, 유효한가?
- 그림 b에서 if EN =?, 유효한가?

유효화 예)



(a)

유효하지 않다 = x의 전달을 가로막는다.



(b)

6

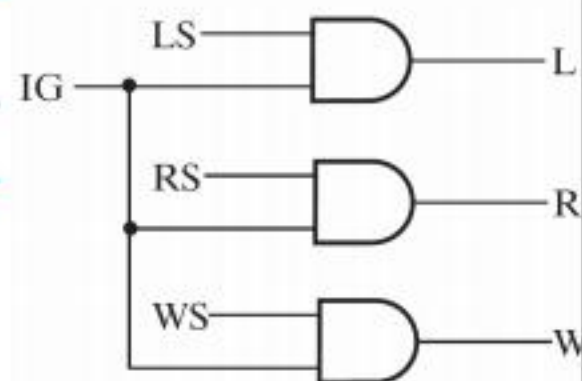
6

유효화(Enabling)

p.118

- 스위치 유효화

Input Switches				Accessory Control		
IG	LS	RS	WS	L	R	W
0	X	X	X	0	0	0
1	0	0	0	0	0	0
1	0	0	1	0	0	1
1	0	1	0	0	1	0
1	0	1	1	0	1	1
1	1	0	0	1	0	0
1	1	0	1	1	0	1
1	1	1	0	1	1	0
1	1	1	1	1	1	1



7

7

예)값고정을 사용한 조명제어 p.116

- M1 : P만이 조명을 켜고 끈다.
 - M2 : R만이 조명을 켜고 끈다.
 - M0 : P 또는 R이 조명을 켜고 끈다.
- $H(P,R) = ?$

P	R	H(M0)	H(M1)	H(M2)
0	0			
0	1			
1	0			
1	1			

P



R

8

8

조명제어 함수의 기능 정의

Mode:		M_0	M_1	M_2
P	R	$H = \bar{P}R + P\bar{R}$	$H = P$	$H = R$
0	0	0	0	0
0	1	1	0	1
1	0	1	1	0
1	1	0	1	1

9

9

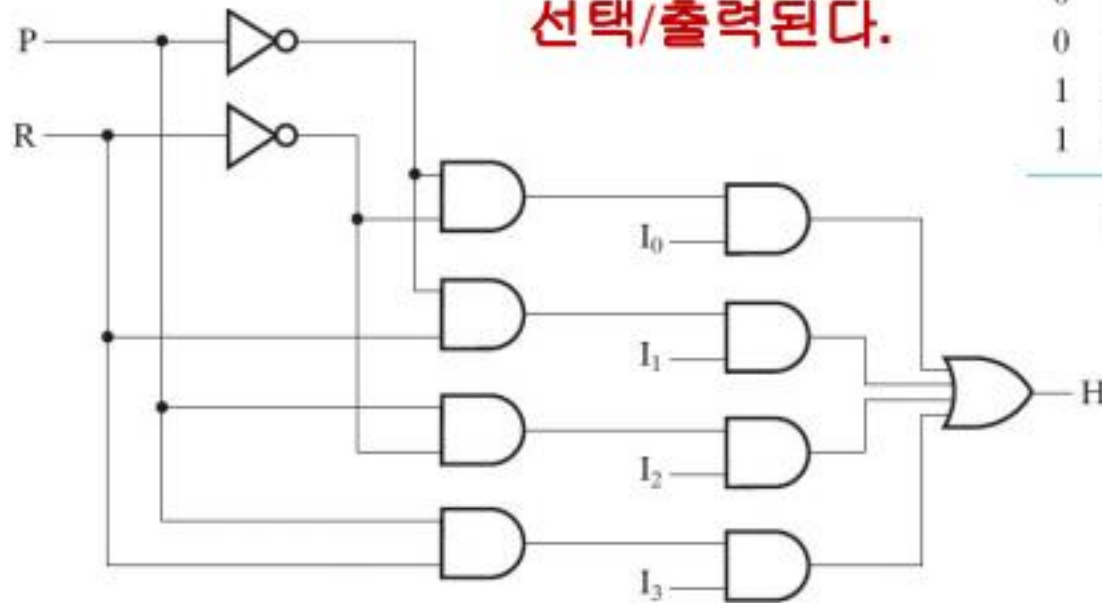
값 고정을 사용한 회로 (value fixing) p.117

■ $H(P,R,I_0,I_1,I_2,I_3)=P'R'I_0+P'RI_1+PR'I_2+PRI_3$

**P,R에 따라 고정값 I가
선택/출력된다.**

P	R	H
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

(b)



(a)

10

10

값고정을 사용한 조명제어 p.116

- M1 : P만이 조명을 켜고 끈다.
- M2 : R만이 조명을 켜고 끈다.
- M0 : P 또는 R이 조명을 켜고 끈다.

$H(P,R) = ?$

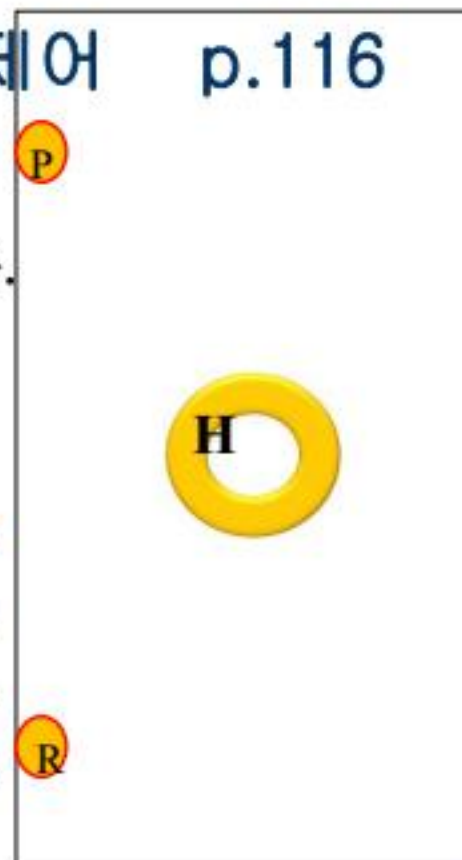
P	R	H(M0)	H(M1)	H(M2)
0	0	0	0	0
0	1	1	0	1
1	0	1	1	0
1	1	0	1	1

I_0

I_1

I_2

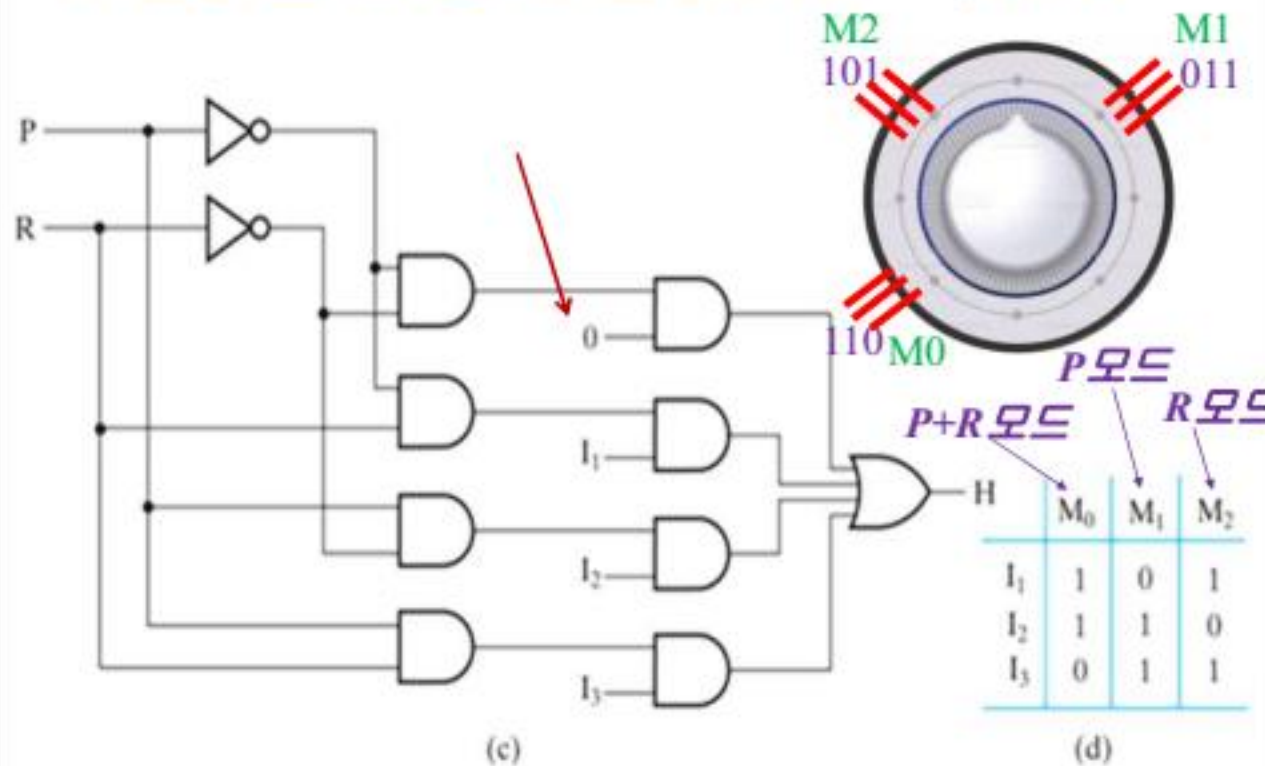
I_3



11

11

값고정을 사용한 조명제어 p.117



12

Decoding p.120

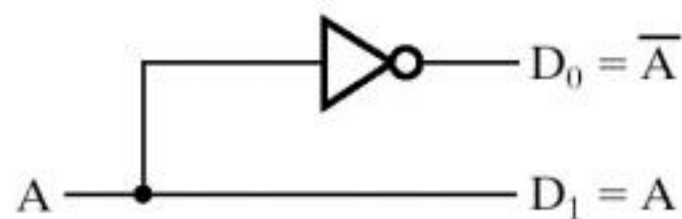
■ decoder

- n비트로 코딩된 2진 정보를 최대 2^n 개의 출력으로 변환하는 조합회로
- n개의 입력 $\Rightarrow 2^n$ 개의 출력
- n-to-m라인 디코더($m < 2^n$)

■ 1-to-2 decoder

A	D ₀	D ₁
0	1	0
1	0	1

(a)



(b)

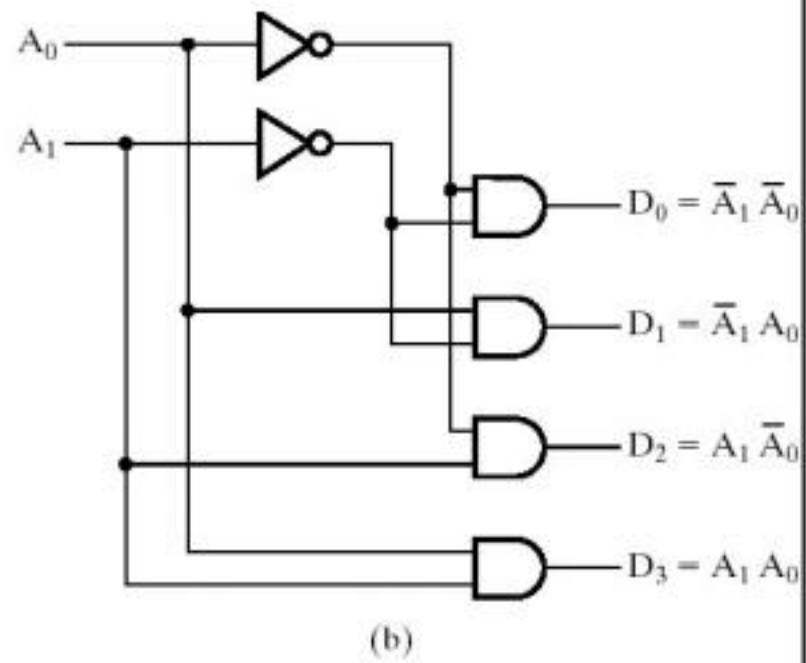
13

Decoding p.120

■ 2-to-4 decoder

A_1	A_0	D_0	D_1	D_2	D_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

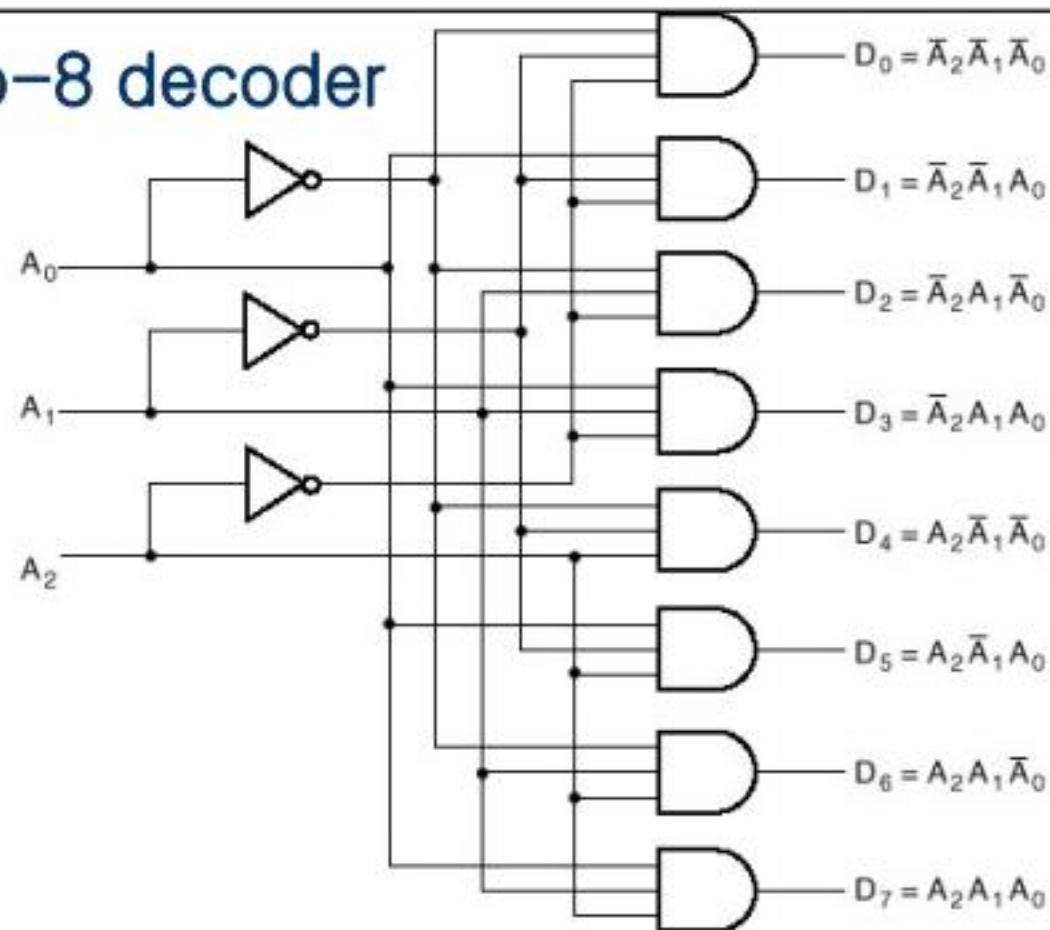
(a)



14

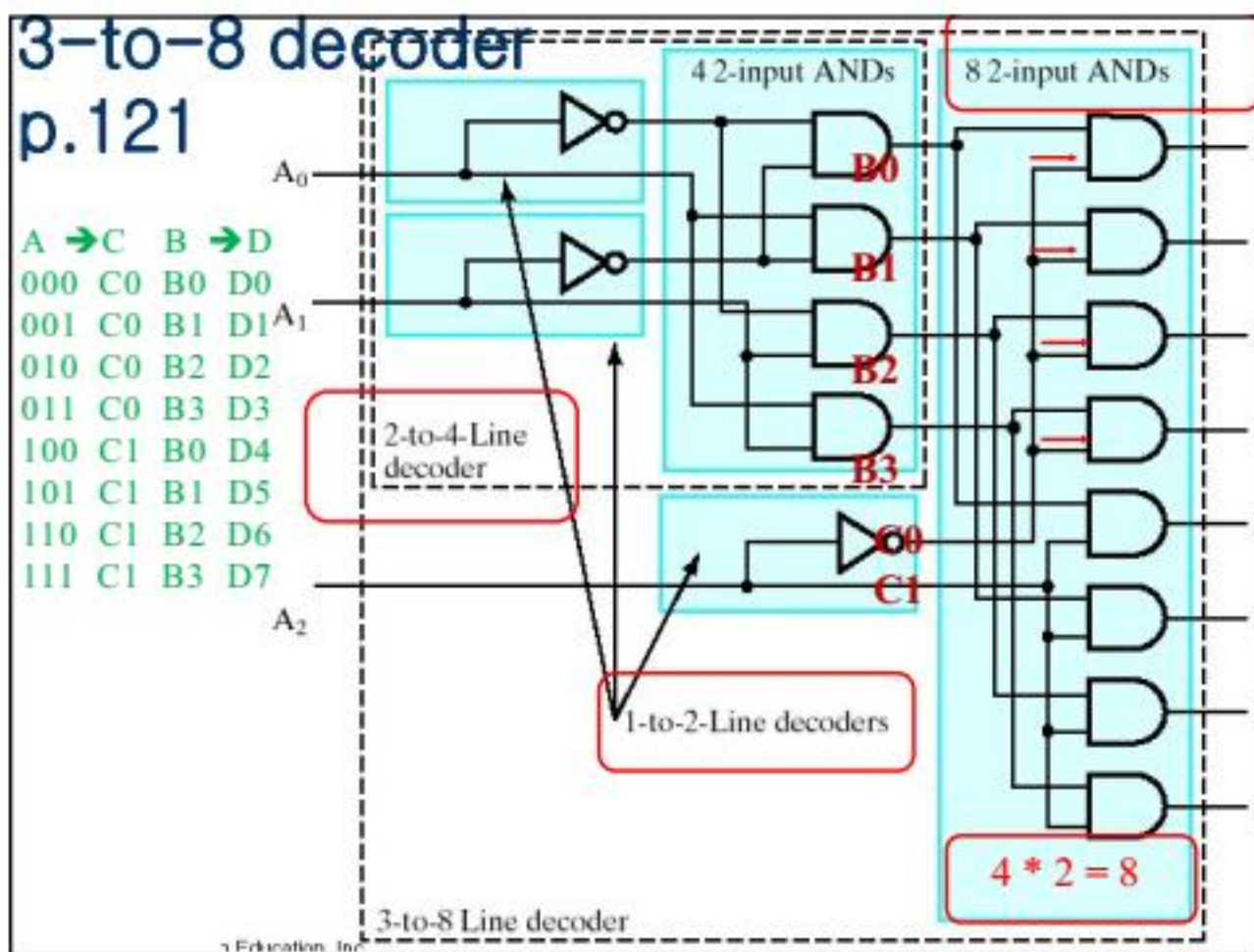
14

3-to-8 decoder

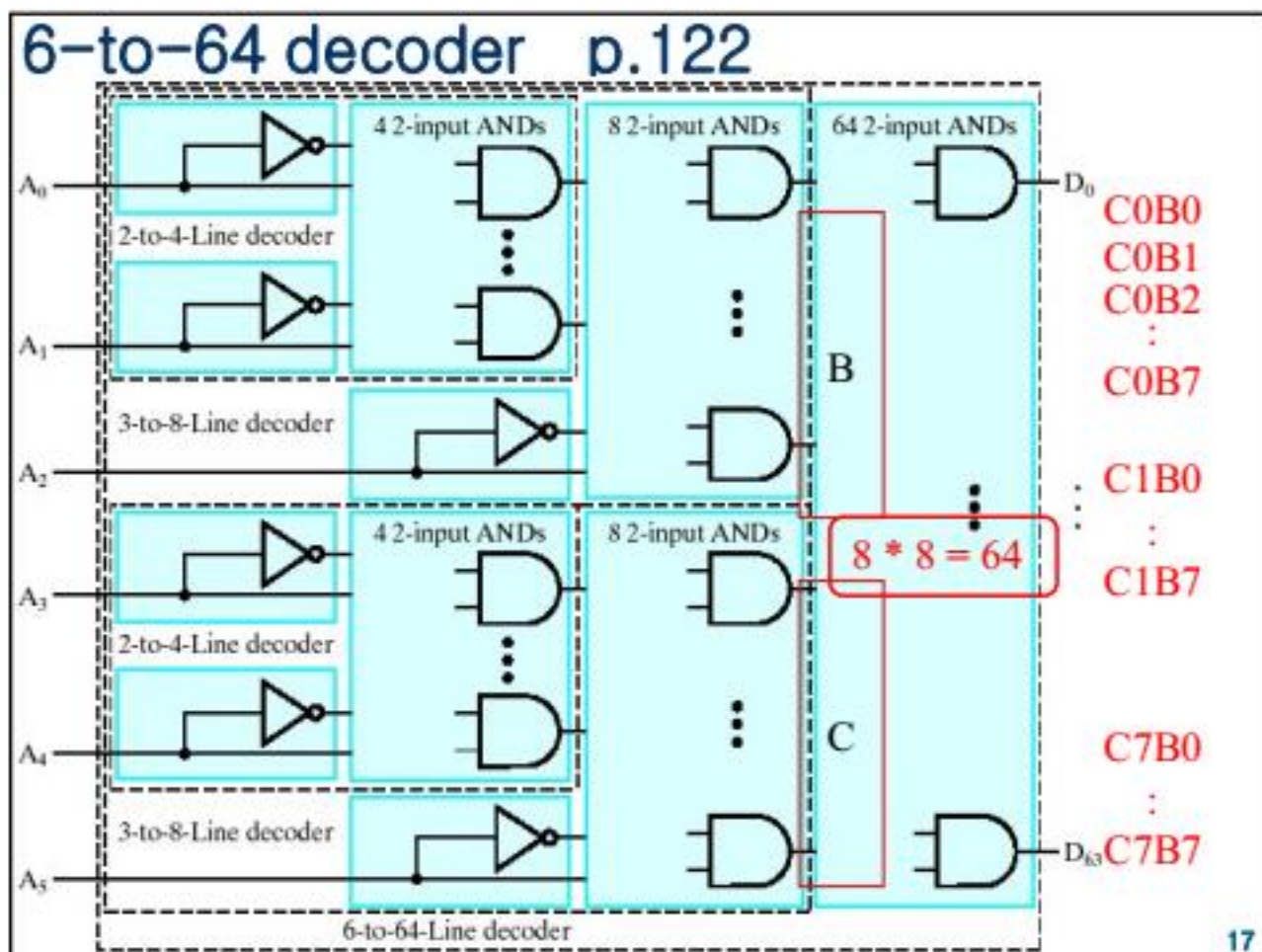


15

15



16



17

17

6-to-64 decoder (비용 절감 측면)

P.123

- $D0 = C0 B0$ (000 000)
- $D1 = C0 B1$ (000 001)
- $D2 = C0 B2$ (000 010)
- :
- $D7 = C0 B7$ (000 111)
- $D8 = C1 B0$ (001 000)
- $D9 = C1 B1$ (001 001)
- :
- $D15 = C1 B7$ (001 111)
- $D16 = C2 B0$ (010 000)
- :
- $D63 = C7 B7$ (111 111)

Gate input cost 비교

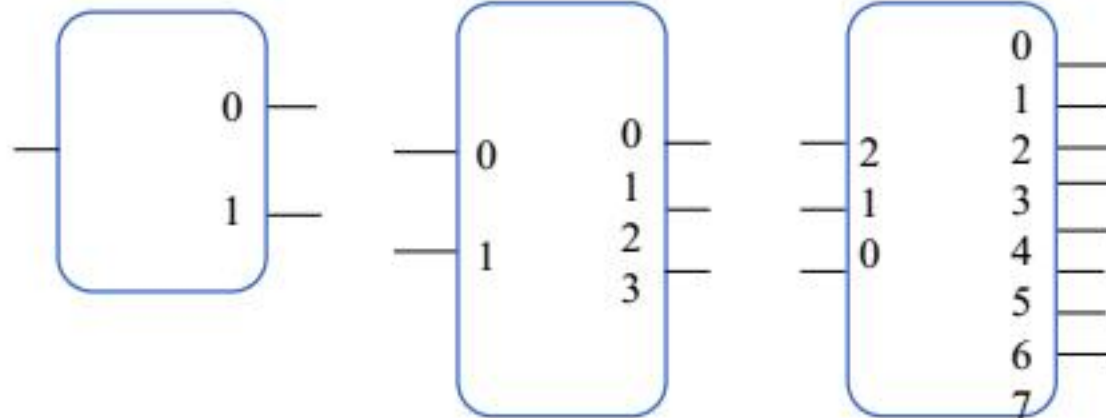
6입력 and gate 사용의 경우
 $6 + (6 \times 64) = 390$

그림 3-20의 경우
 $6 + 2(2 \times 4) + 2(2 \times 8) + 2 \times 64$
 $= 182$

18

18

Decoder 1X2 2X4 3X8



19

19

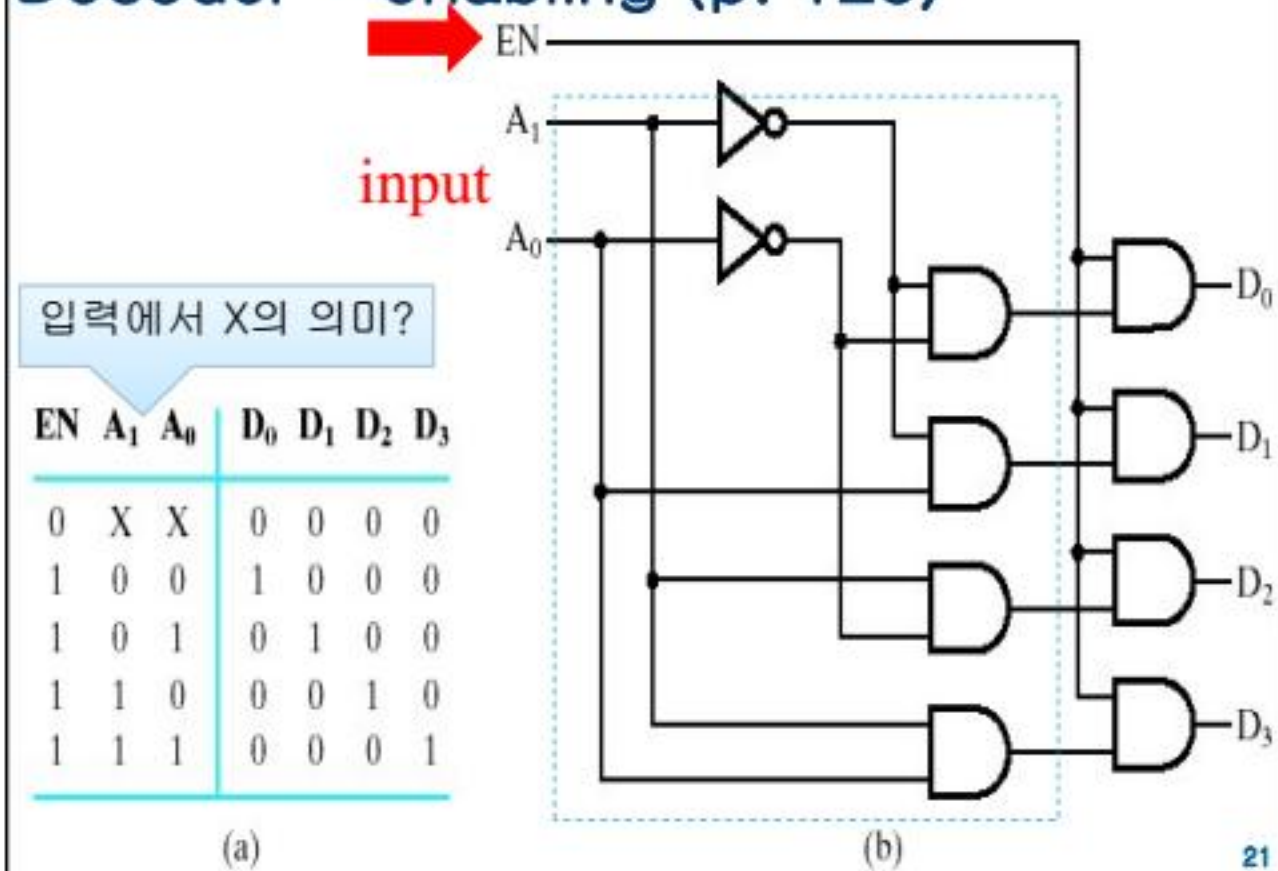
Function Blocks의 또 다른 적용



20

20

Decoder + enabling (p. 123)



21

21

Combinational Logic Design p.128

일반적으로 컴퓨터 설계에서 매우 유용하게 사용되는 조합회로 블록들에 대해 알아본다.

Decoder, Encoder, Multiplexer

1

1

Rudimentary Logic Functions p.114 기본논리함수

- 값고정(value fixing)
- 전달(transferring)
- 보수화(inverting)
- 유효화(enabling).. 나중에

■ 단일 변수 X에 대한 함수들

- 값 고정(0, 1)
- 전달(X)
- 반전(X') : not gate

X	F = 0	F = X	F = \bar{X}	F = 1
0	0	0	1	1
1	0	1	0	1

2

2

1

p.115
(양의논리에서)

X	F = 0	F = X	F = \bar{X}	F = 1
0	0	0	1	1
1	0	1	0	1

1(+)에 해당하는 전압

V_{CC} or V_{DD}

1 ——— F = 1



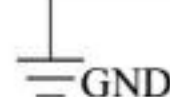
F = 1

X ——— F = X

(c)

0 ——— F = 0

F = 0



X ——— F = \bar{X}

(d)

(a)

0(-)에 해당하는 전압

(b)

3

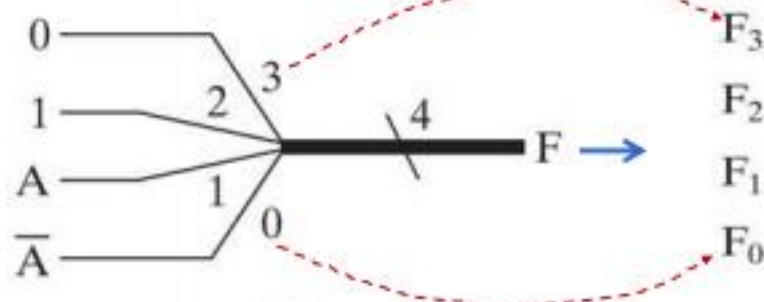
3

다중 비트 함수(도식화 방법) p.116

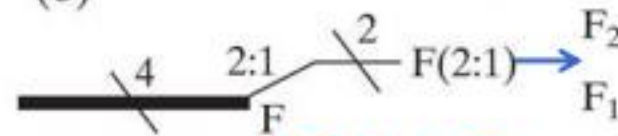
0 ——— F_3
1 ——— F_2
A ——— F_1
 \bar{A} ——— F_0

(a) 함수정의

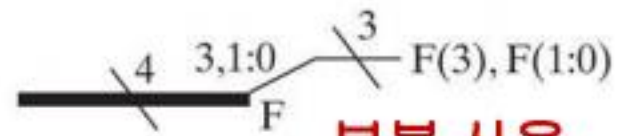
정의 후 (b)와 같이
도식화할 수 있다.



(b)



(c) 부분 사용



(d) 부분 사용

5

5

2

유효화(Enabling)

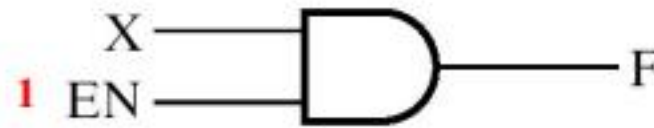
p.118

- Enabling : 입력이 출력으로 전달된다 는 의미

– an input signal → an output

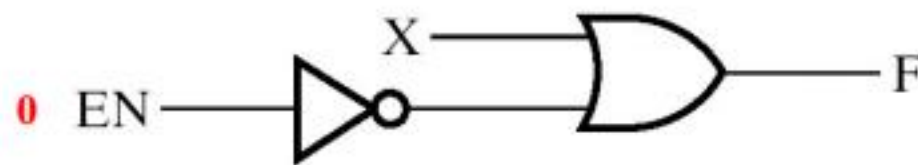
- 그림 a에서 if EN =?, 유효한가?
- 그림 b에서 if EN =?, 유효한가?

유효화 예)



(a)

유효하지 않다 = x의 전달을 가로막는다.



(b)

6

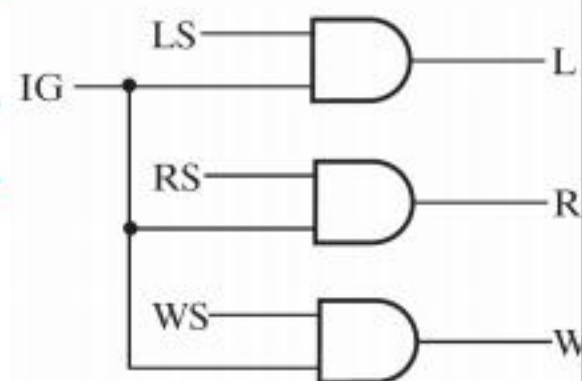
6

유효화(Enabling)

p.118

- 스위치 유효화

Input Switches				Accessory Control		
IG	LS	RS	WS	L	R	W
0	X	X	X	0	0	0
1	0	0	0	0	0	0
1	0	0	1	0	0	1
1	0	1	0	0	1	0
1	0	1	1	0	1	1
1	1	0	0	1	0	0
1	1	0	1	1	0	1
1	1	1	0	1	1	0
1	1	1	1	1	1	1



7

7

예)값고정을 사용한 조명제어 p.116

- M1 : P만이 조명을 켜고 끈다.
 - M2 : R만이 조명을 켜고 끈다.
 - M0 : P 또는 R이 조명을 켜고 끈다.
- $H(P,R) = ?$

P	R	H(M0)	H(M1)	H(M2)
0	0			
0	1			
1	0			
1	1			

P



R

8

8

조명제어 함수의 기능 정의

Mode:		M_0	M_1	M_2
P	R	$H = \bar{P}R + P\bar{R}$	$H = P$	$H = R$
0	0	0	0	0
0	1	1	0	1
1	0	1	1	0
1	1	0	1	1

9

9

값 고정을 사용한 회로 (value fixing) p.117

■ $H(P,R,I_0,I_1,I_2,I_3)=P'R'I_0+P'RI_1+PR'I_2+PRI_3$

**P,R에 따라 고정값 I가
선택/출력된다.**



P	R	H
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

(b)

(a)

10

10

값고정을 사용한 조명제어 p.116

- M1 : P만이 조명을 켜고 끈다.
- M2 : R만이 조명을 켜고 끈다.
- M0 : P 또는 R이 조명을 켜고 끈다.

$H(P,R) = ?$

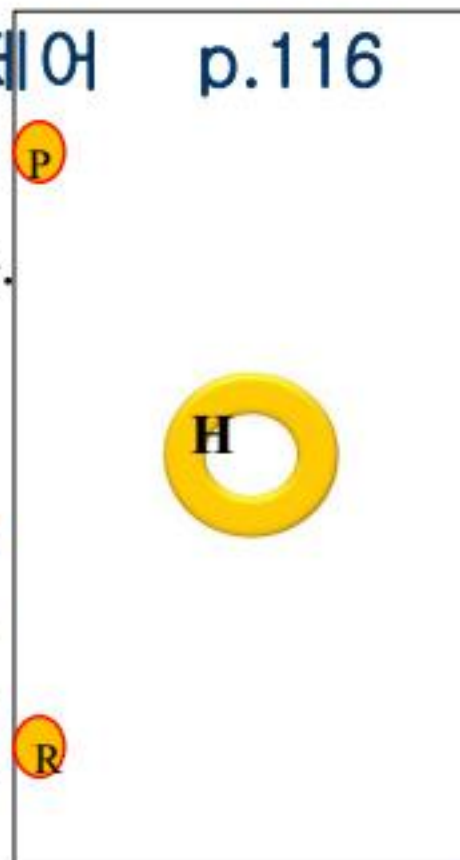
P	R	H(M0)	H(M1)	H(M2)
0	0	0	0	0
0	1	1	0	1
1	0	1	1	0
1	1	0	1	1

I_0

I_1

I_2

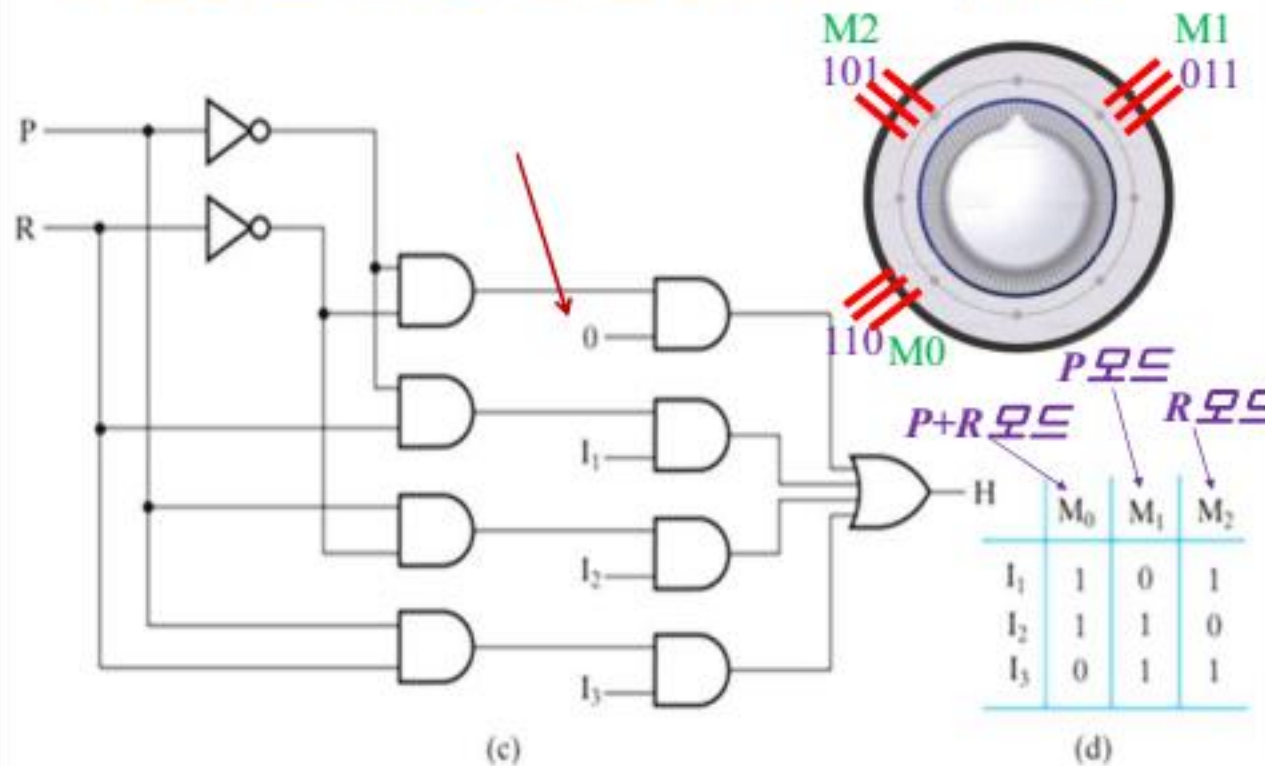
I_3



11

11

값고정을 사용한 조명제어 p.117



12

12

Decoding p.120

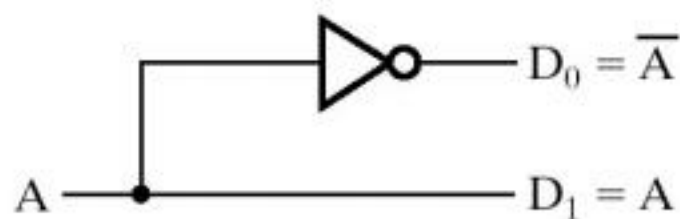
■ decoder

- n비트로 코딩된 2진 정보를 최대 2^n 개의 출력으로 변환하는 조합회로
- n개의 입력 $\Rightarrow 2^n$ 개의 출력
- n-to-m라인 디코더($m < 2^n$)

■ 1-to-2 decoder

A	D ₀	D ₁
0	1	0
1	0	1

(a)



(b)

13

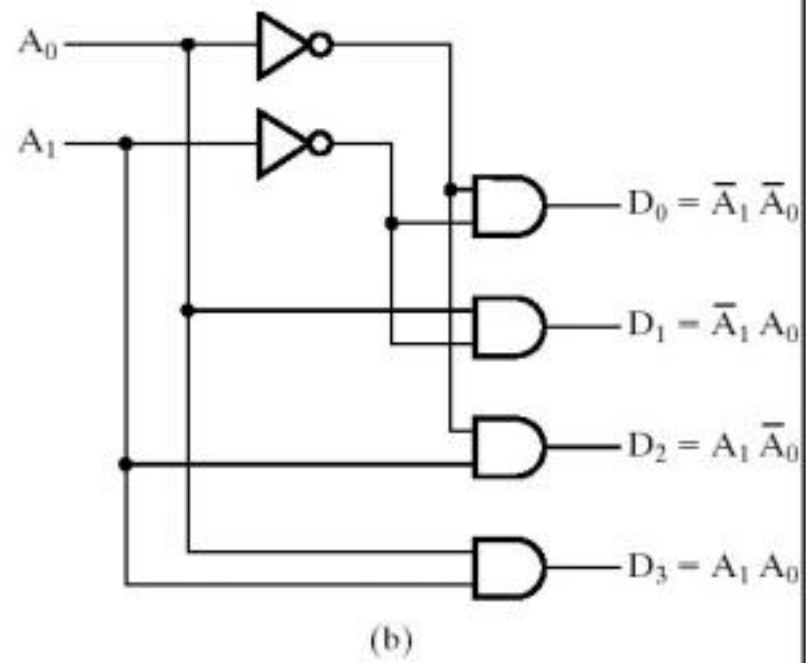
13

Decoding p.120

■ 2-to-4 decoder

A_1	A_0	D_0	D_1	D_2	D_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

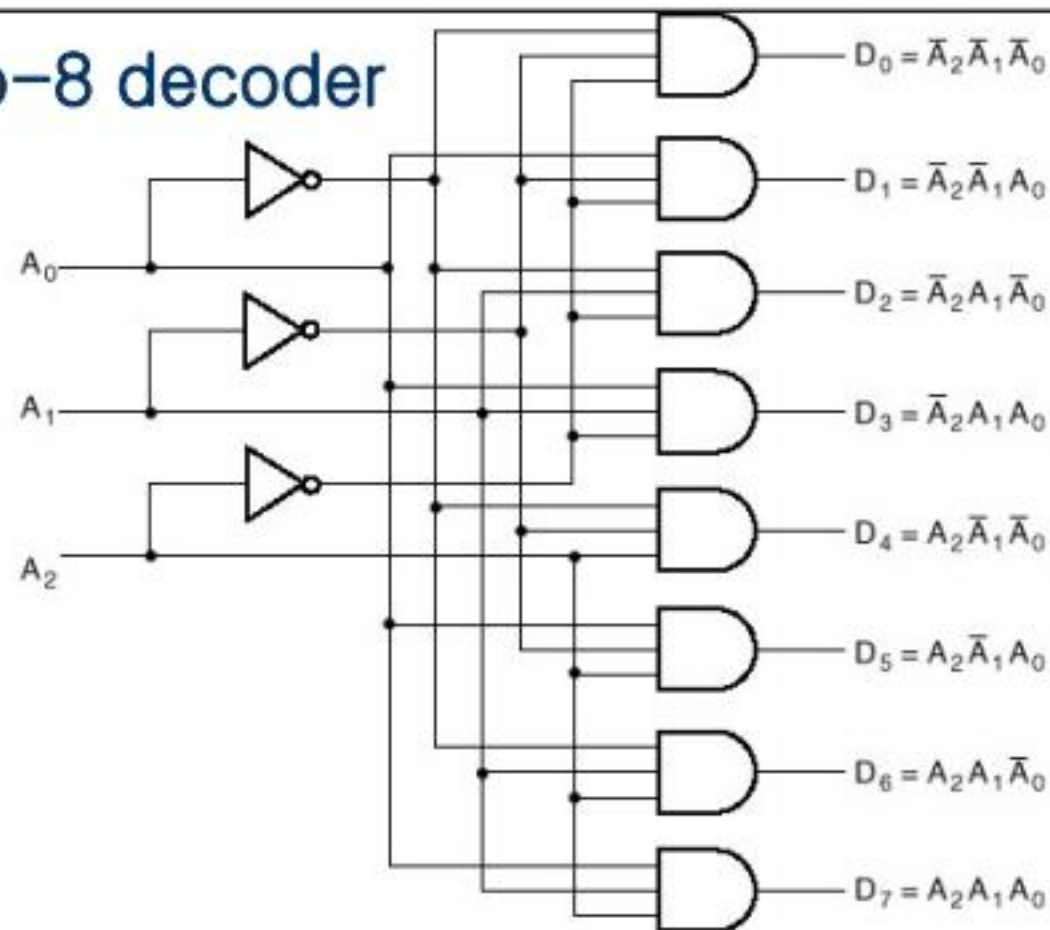
(a)



14

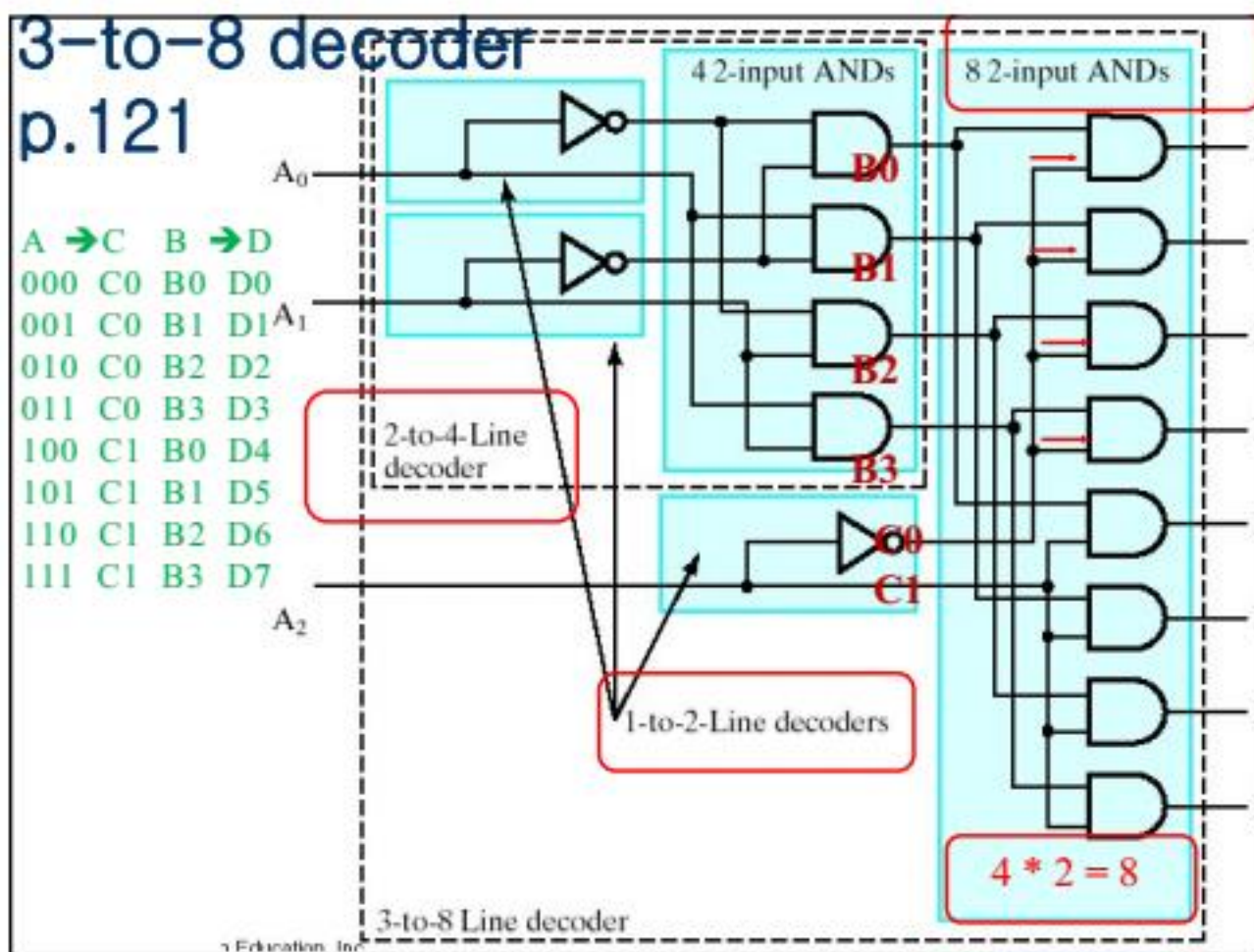
14

3-to-8 decoder

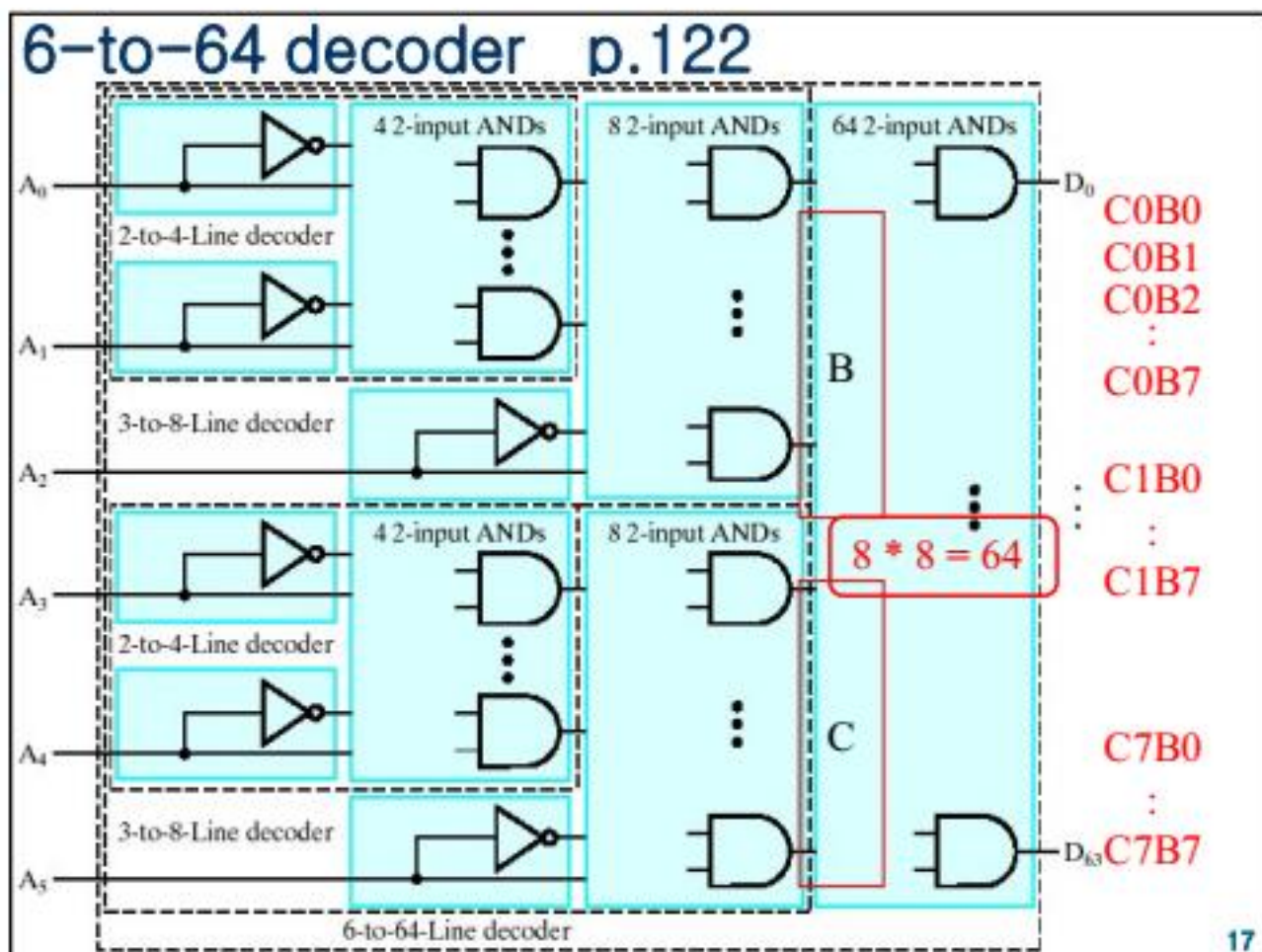


15

15



16



17

6-to-64 decoder (비용 절감 측면)

P.123

- $D0 = C0\ B0\ (000\ 000)$
- $D1 = C0\ B1\ (000\ 001)$
- $D2 = C0\ B2\ (000\ 010)$
- :
- $D7 = C0\ B7\ (000\ 111)$
- $D8 = C1\ B0\ (001\ 000)$
- $D9 = C1\ B1\ (001\ 001)$
- :
- $D15 = C1\ B7\ (001\ 111)$
- $D16 = C2\ B0\ (010\ 000)$
- :
- $D63 = C7\ B7\ (111\ 111)$

Gate input cost 비교

6입력 and gate 사용의 경우

$$6 + (6 \times 64) = 390$$

그림 3-20의 경우

$$6 + 2(2 \times 4) + 2(2 \times 8) + 2 \times 64 = 182$$

18

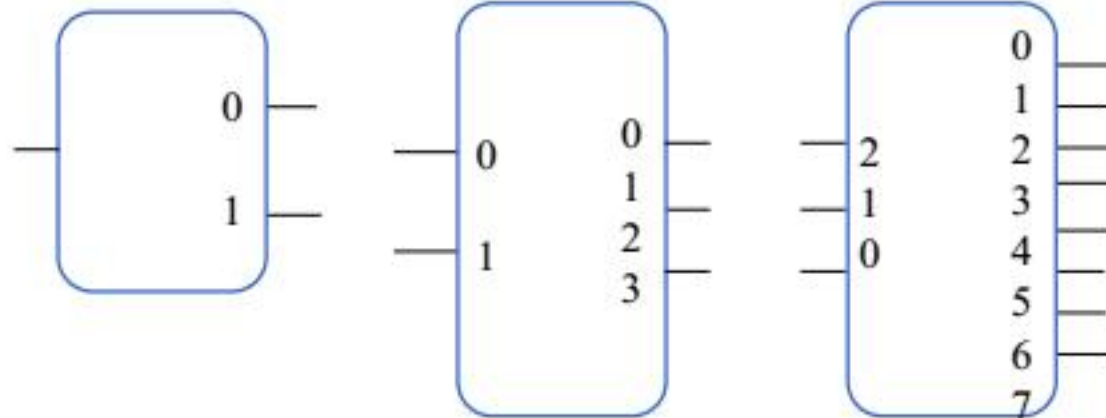
18

Decoder

1X2

2X4

3X8



19

19

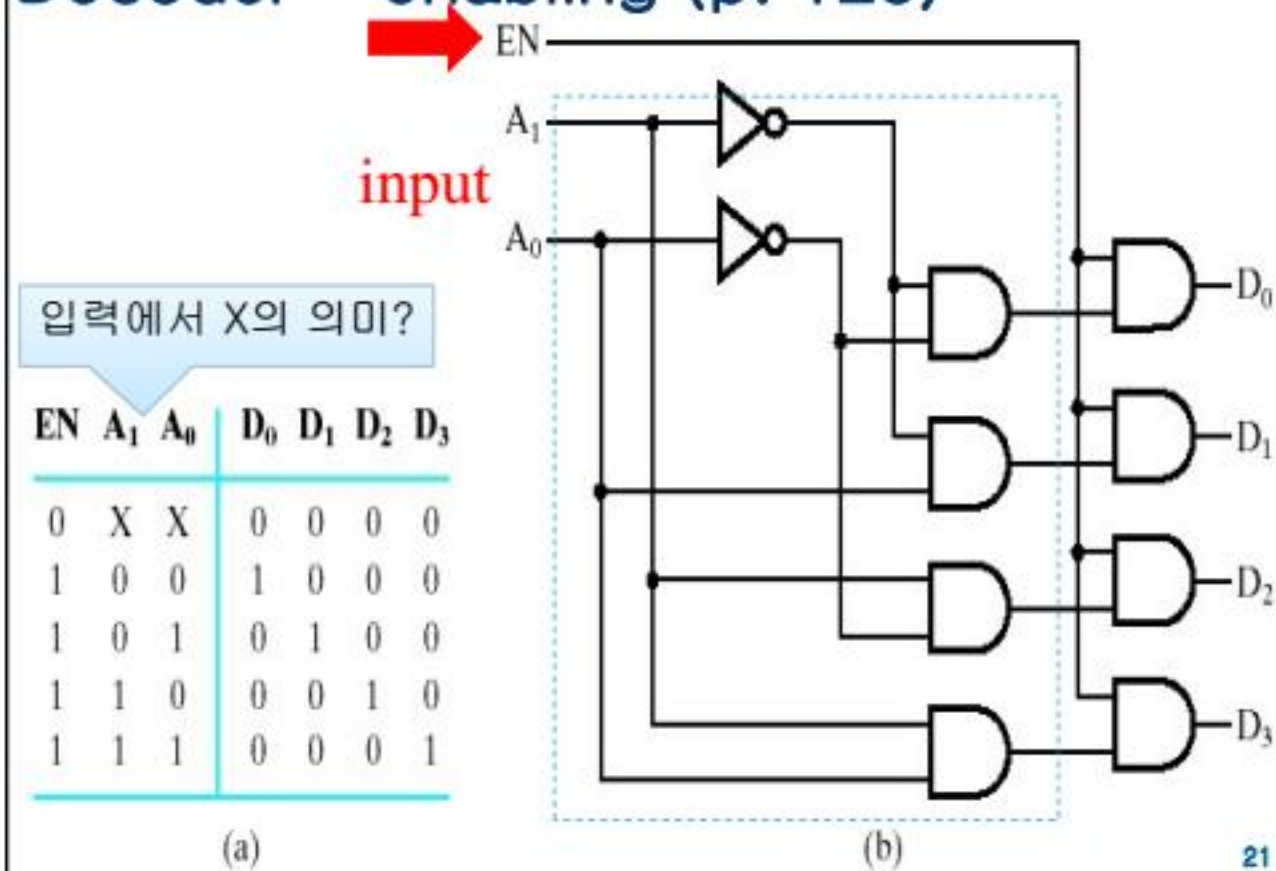
Function Blocks의 또 다른 적용



20

20

Decoder + enabling (p. 123)



21

21





