

## 레지스터와 레지스터 전송 p.307

이제까지의 학습에 대한  
분석, 이해 능력을 확인한다.

조합회로  
순차회로  
레지스터 + 조합회로 블록  
카운터 설계

1

1

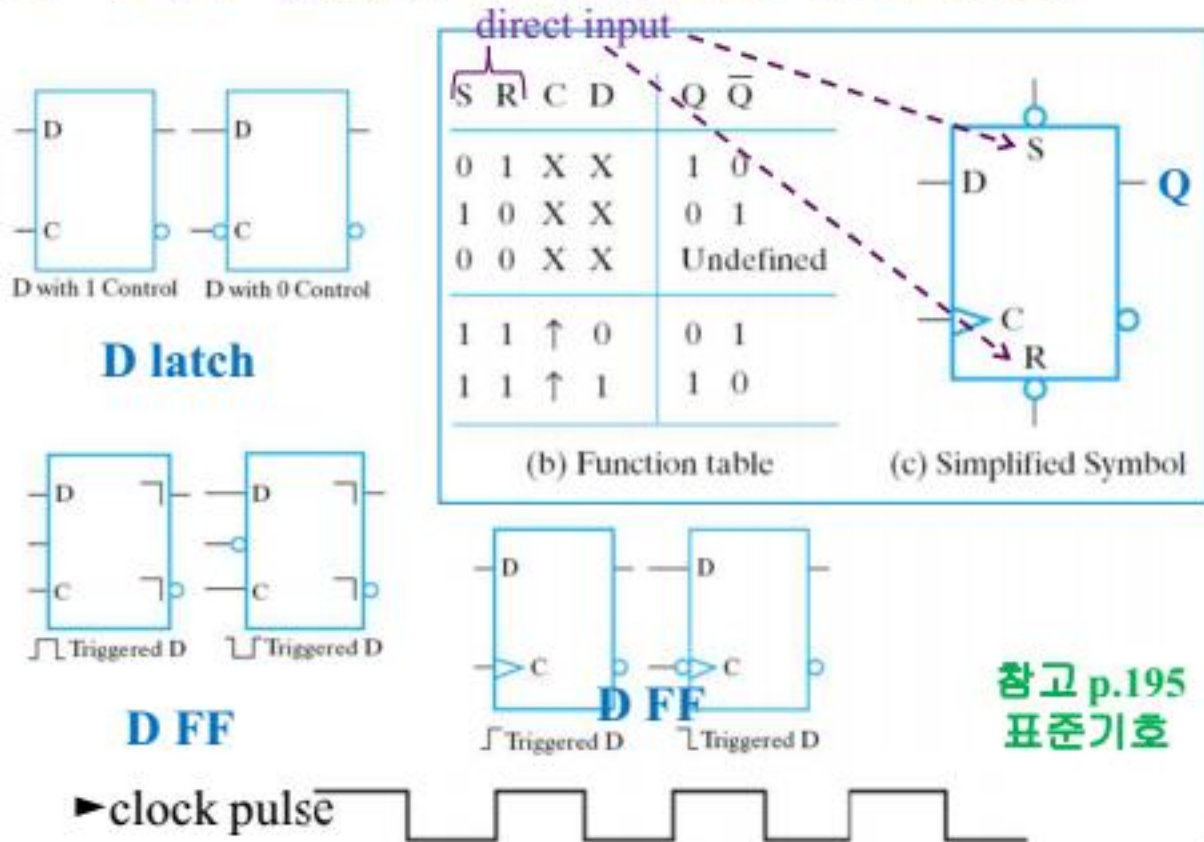
## 6-1 Registers and load enable p.308

- Register : Flip-Flop의 집합
  - Flip-Flop : 한 비트의 정보를 저장하는 기억요소.
  - n bit Register : n개의 FF과 이들의 저장상태(비트) 전이를 수행하는 게이트들로 구성
  - 연산 등을 위해 n비트 정보를 임시 저장
  - CPU / FPU등에 내장, 32bit 레지스터 ~
- Counter : 미리 정해진 순서에 따라 정보를 바꾸는 레지스터
  - 특수형태의 레지스터
    - m개의 FF과 이들의 상태를 정해진 순서대로 바뀌게 하는 게이트들로 구성
    - ex) 4비트 2진 카운터, BCD 카운터, 임의순서 카운터

2

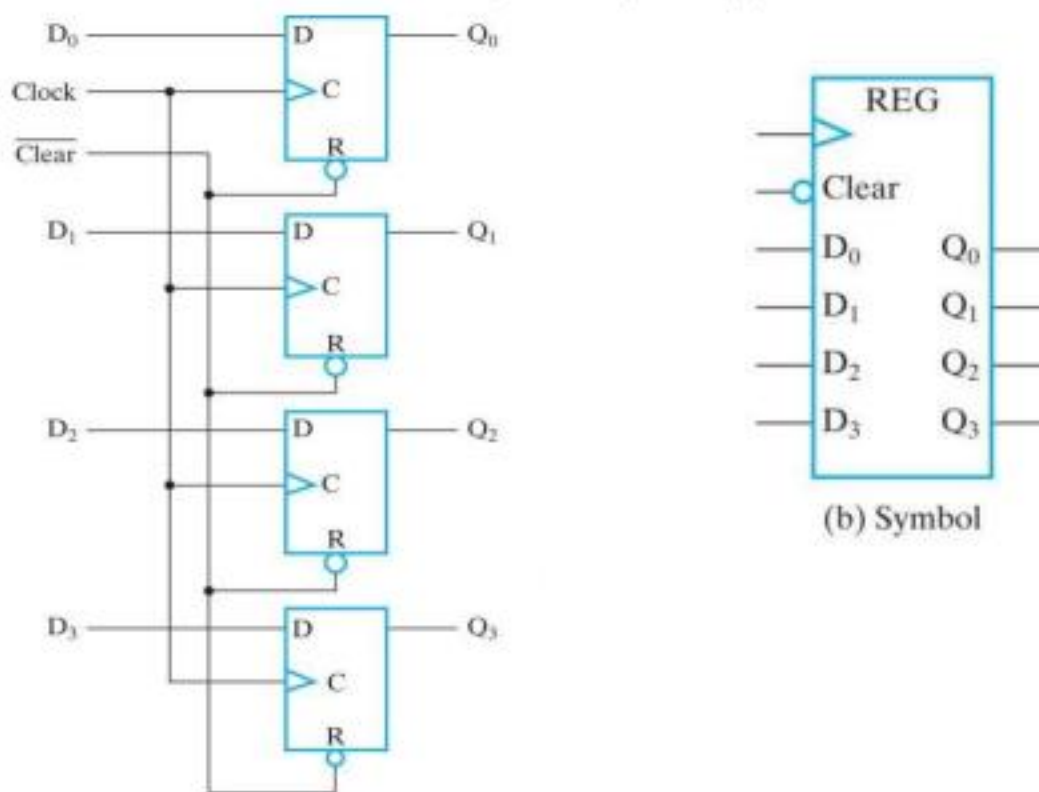
2

## 기억요소 : Latch, Flip-Flop 그래픽 기호 정리



3

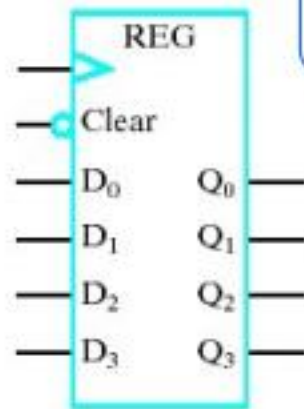
## 4비트 레지스터(D-ff) p.309



4

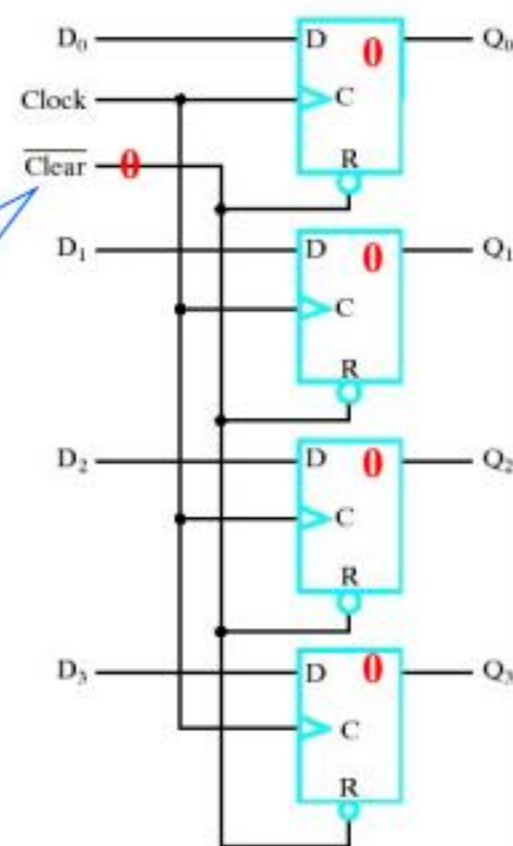
## Register p.309

- 4bit D형 FF로 구성된 레지스터



(b) Symbol

Reset이 필요할때  
0을 입력한다.  
(비동기)



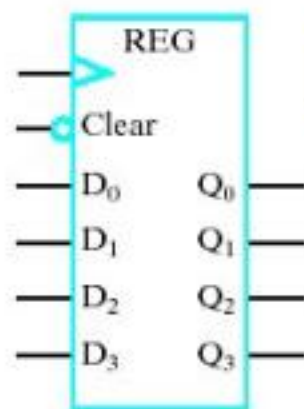
(a) Logic diagram

5

5

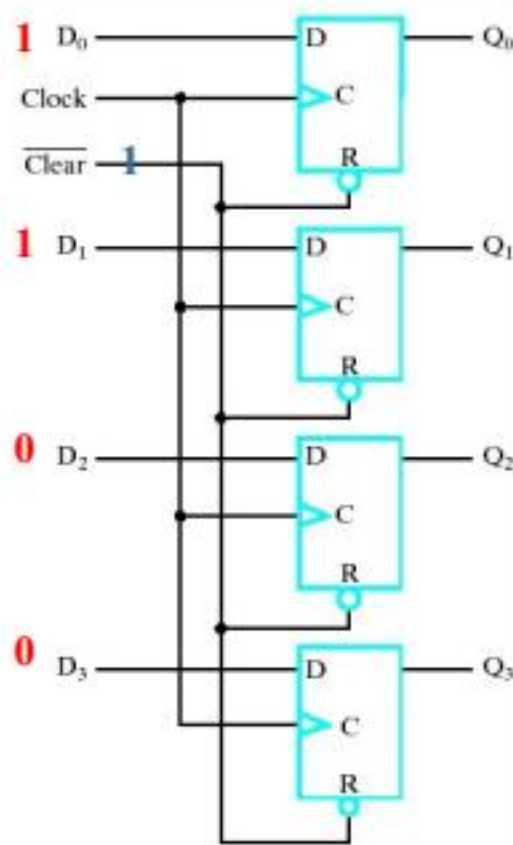
## Register p.309

- 4bit D형 FF로 구성된 레지스터



(b) Symbol

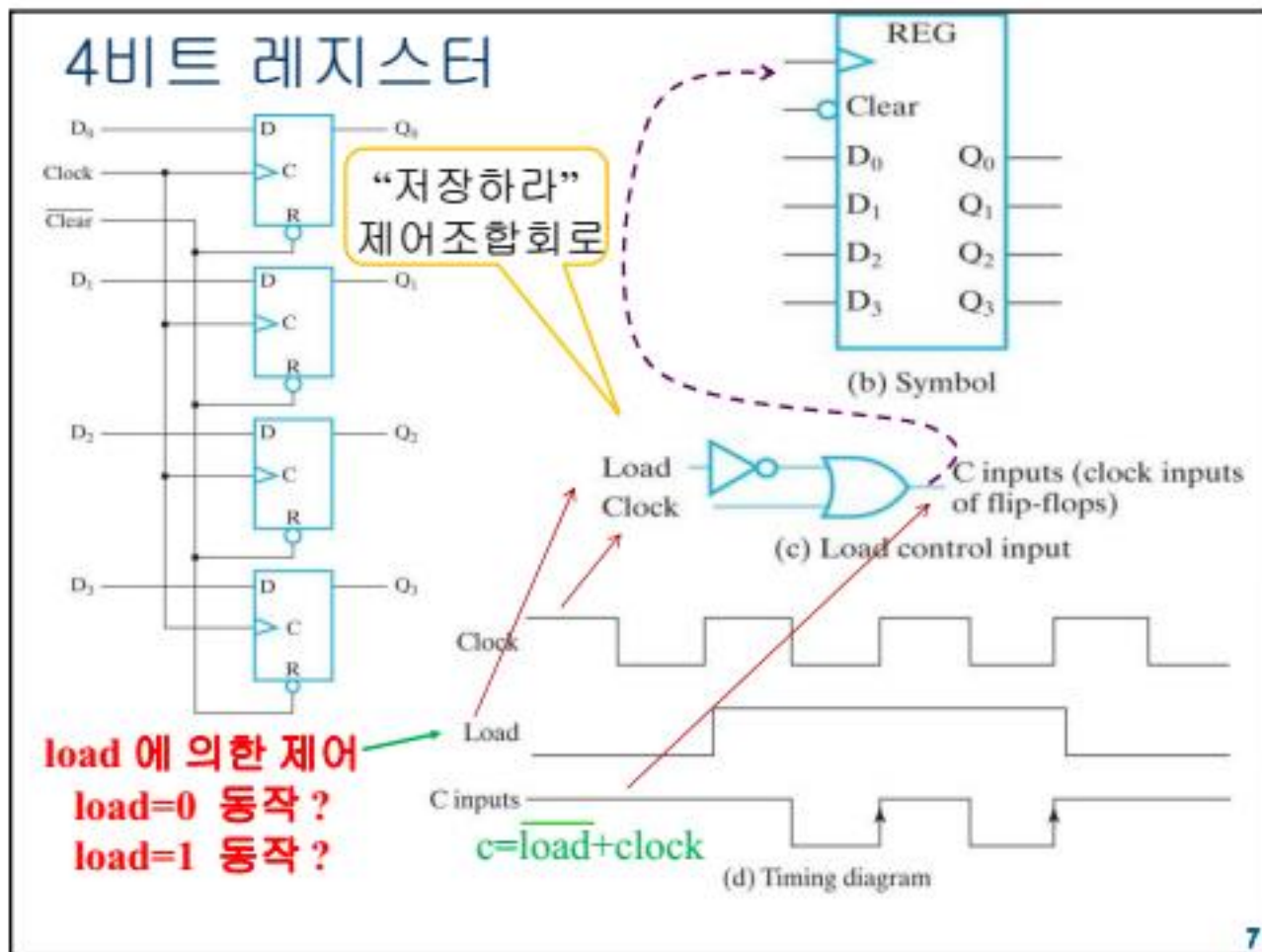
병렬 loading



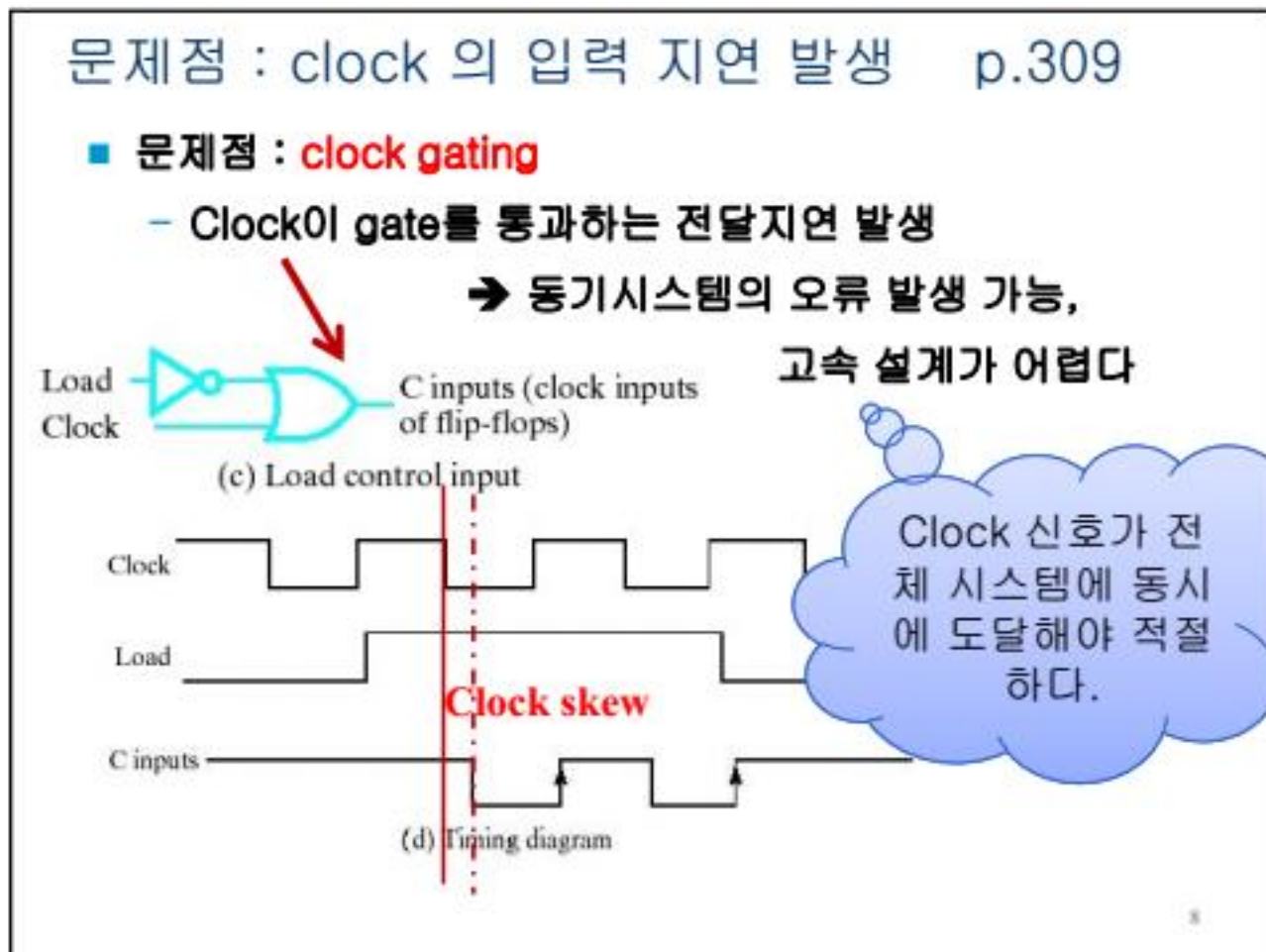
(a) Logic diagram

6

6



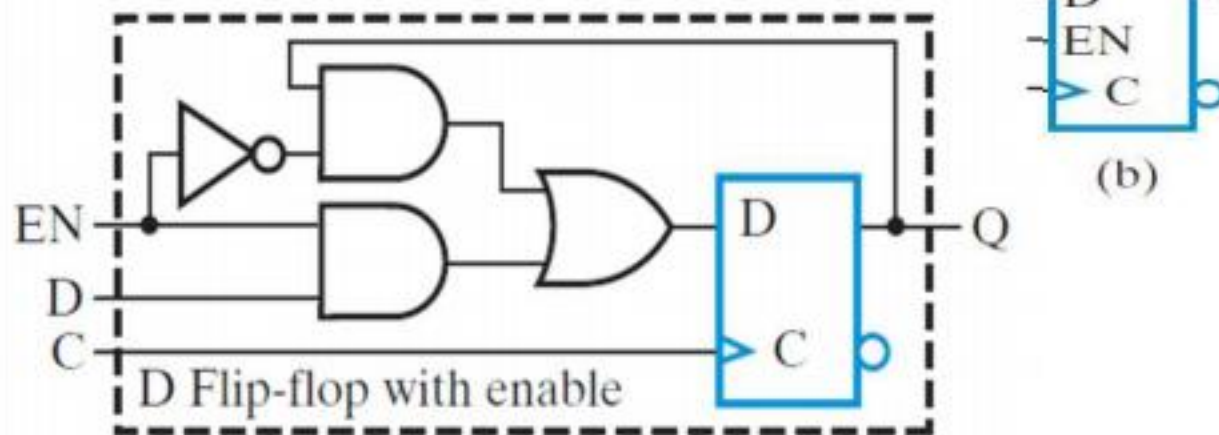
7



8



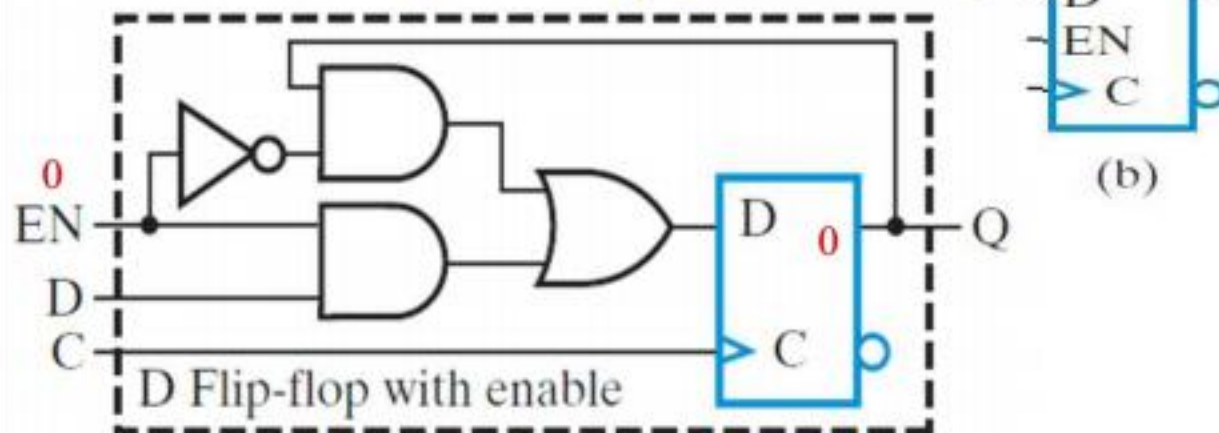
다른 해결 p.311 순차회로 분석?



EN (enable)	현재상태 Q	다음상태 Q
0	0	
0	1	
1	0	
1	1	

9

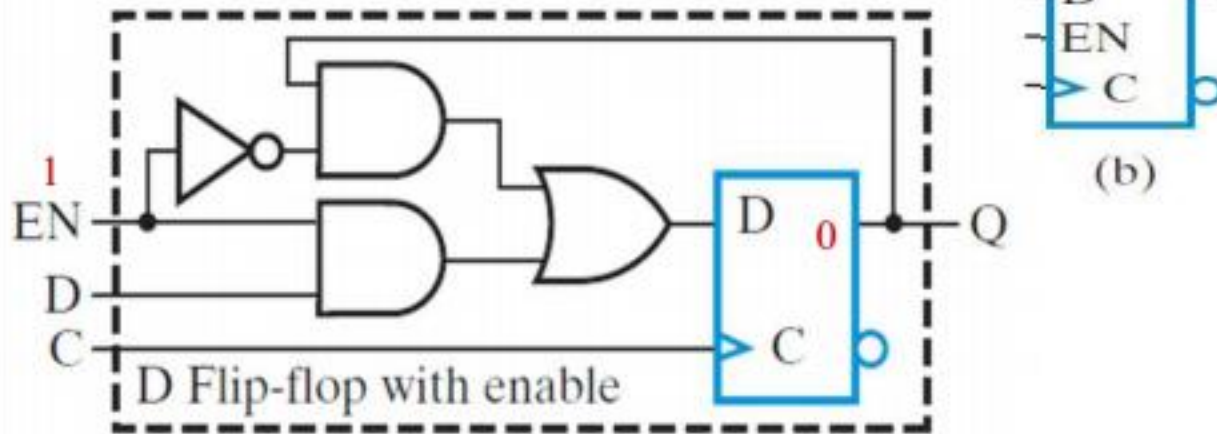
다른 해결 p.311 순차회로 분석?



EN (enable)	현재상태 Q	다음상태 Q
0	0	
0	1	
1	0	
1	1	

10

## 다른 해결 p.311 순차회로 분석?

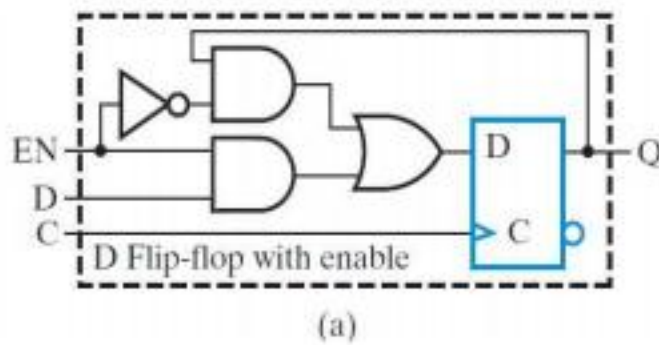


EN (enable)	현재상태 Q	다음상태 Q
0	0	
0	1	
1	0	
1	1	

11

11

## 순차회로 분석



$$D_Q = QE' + ED$$

Q(t)	E	D	Q(t+1)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

EN (enable)	현재상태 Q	다음상태 Q
0	0	0
0	1	1
1	0	D
1	1	D

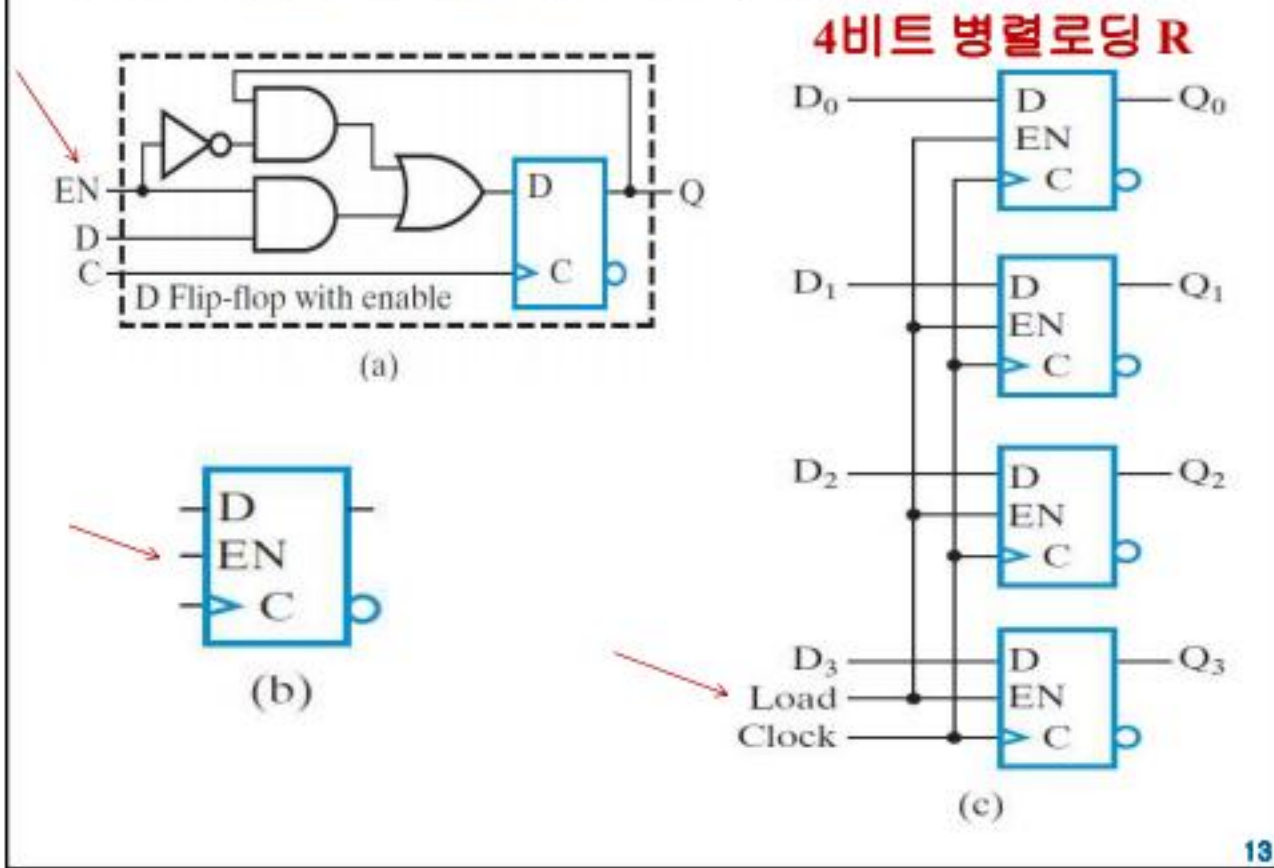
no change

load

12

12

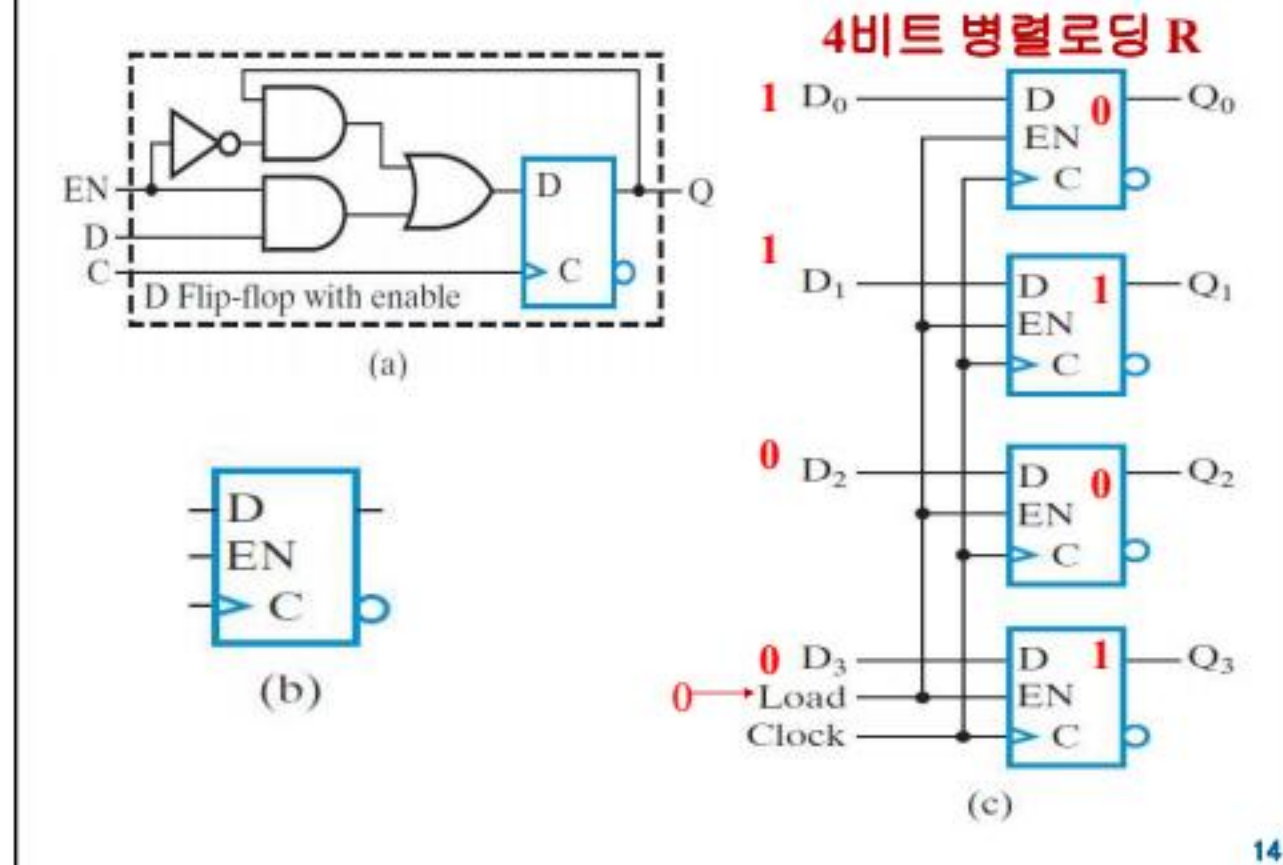
clock 지연문제 해결 (enable제어신호) p.311



13

13

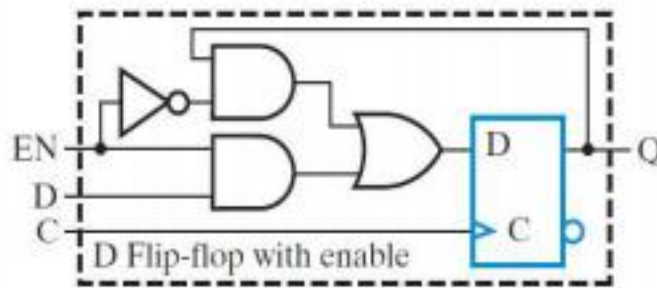
동작 확인 p.311



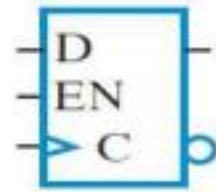
14

14

동작 확인 p.311

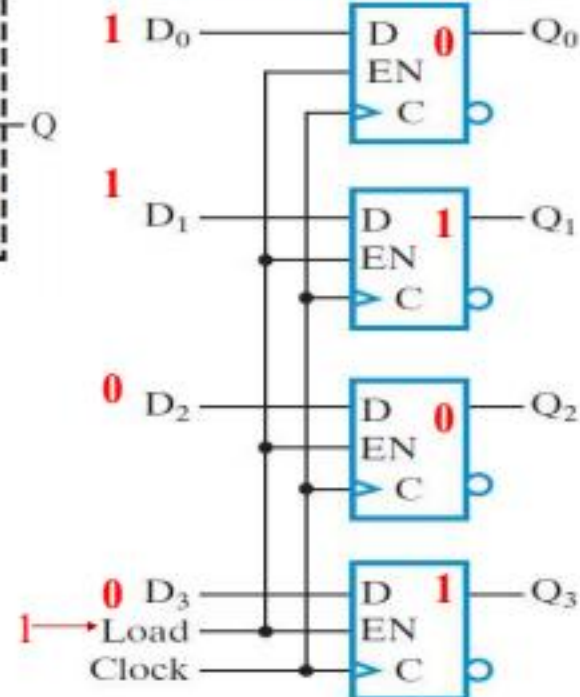


(a)



(b)

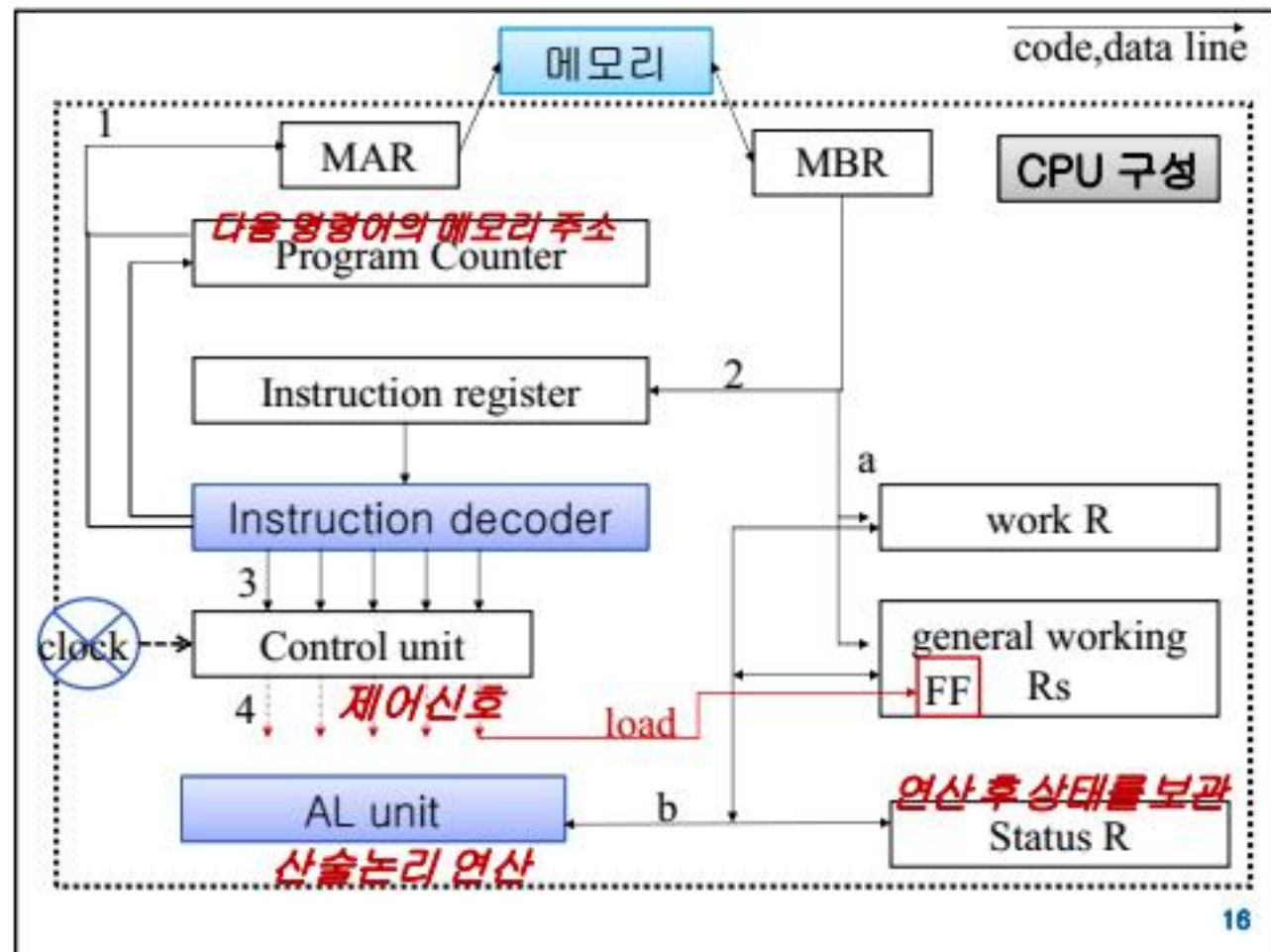
### 4비트 병렬로딩 R



(c)

15

15



16

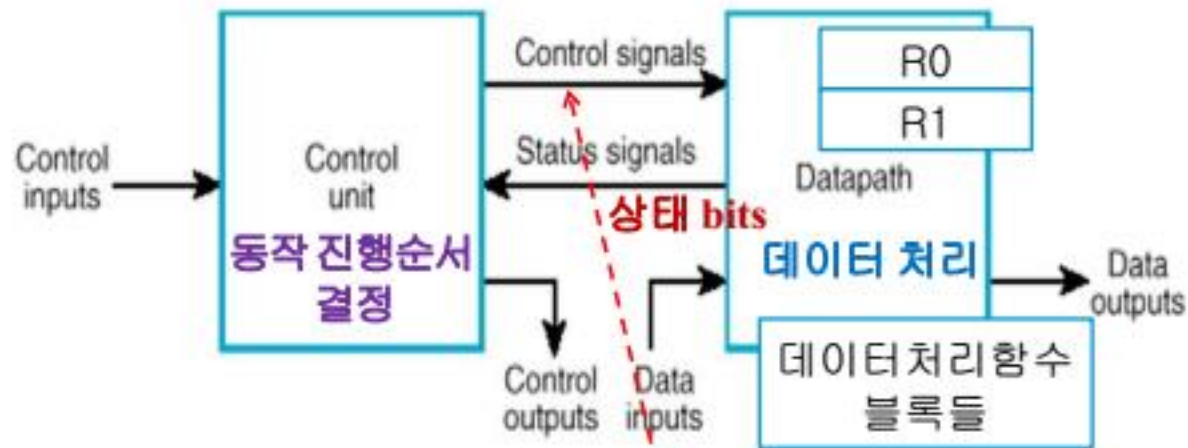
16



## 6.2 레지스터 전송 Register Transfer

- 디지털 시스템은 크게 두 모듈로 나뉜다

- 데이터 처리장치 + 제어장치



- 레지스터의 동작 : shift, count, clear, load..
- micro operation : 레지스터의 data에 대한 기본 동작

17

17

## Micro Operations p.316

- Registers의 동작(제어신호)

- Load, Count, Shift, Add, Subtract, Clear, Set

- Micro operation

- 레지스터의 data에 대한 기본 동작을 *micro operations* 이라 부른다.
  - 레지스터에 내용을 load한다
  - 1증가한다
  - 두 레지스터를 더한다. (Add 연산)
  - 다른 레지스터로 내용을 옮긴다. (Move 연산)
- 한 클럭 동안의 동작
- 제어장치의 제어신호 → one micro operation 지시

18

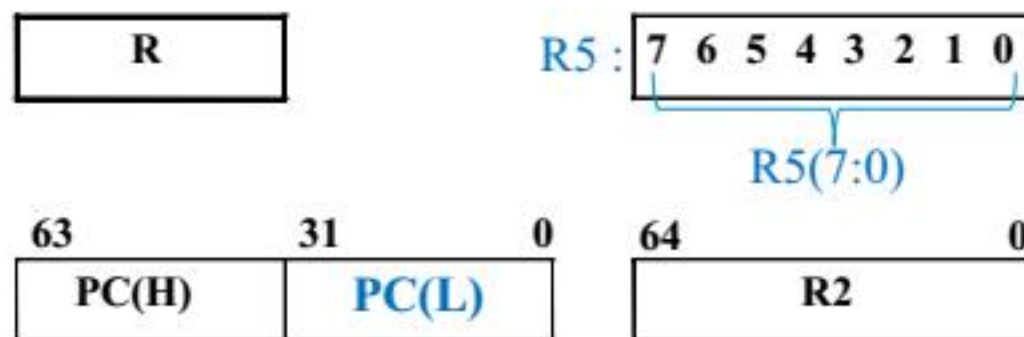
18

## 6-3 Register Transfer Operations

### ■ Register 이동식에 쓰이는 기호명칭

- R1, R2...
- AR : address register(MAR)
- IR : instruction register
- PC : program counter

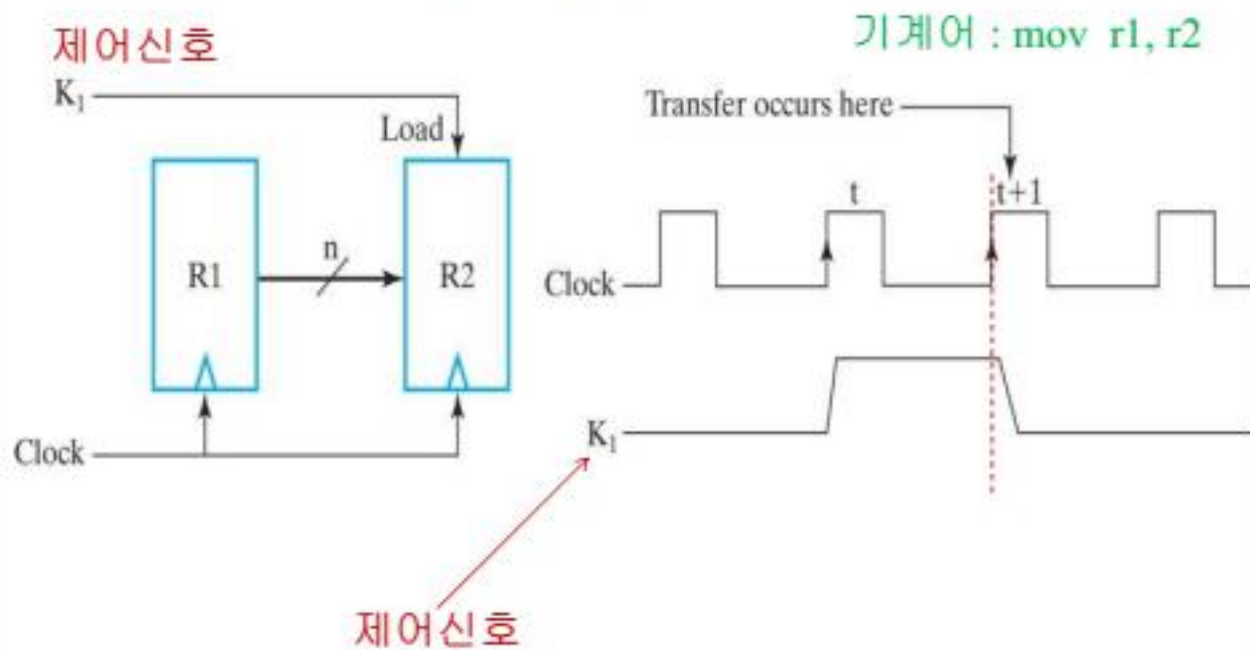
### ■ register block diagram



19

19

## 레지스터 전송 $K_1: (R2 \leftarrow R1)$

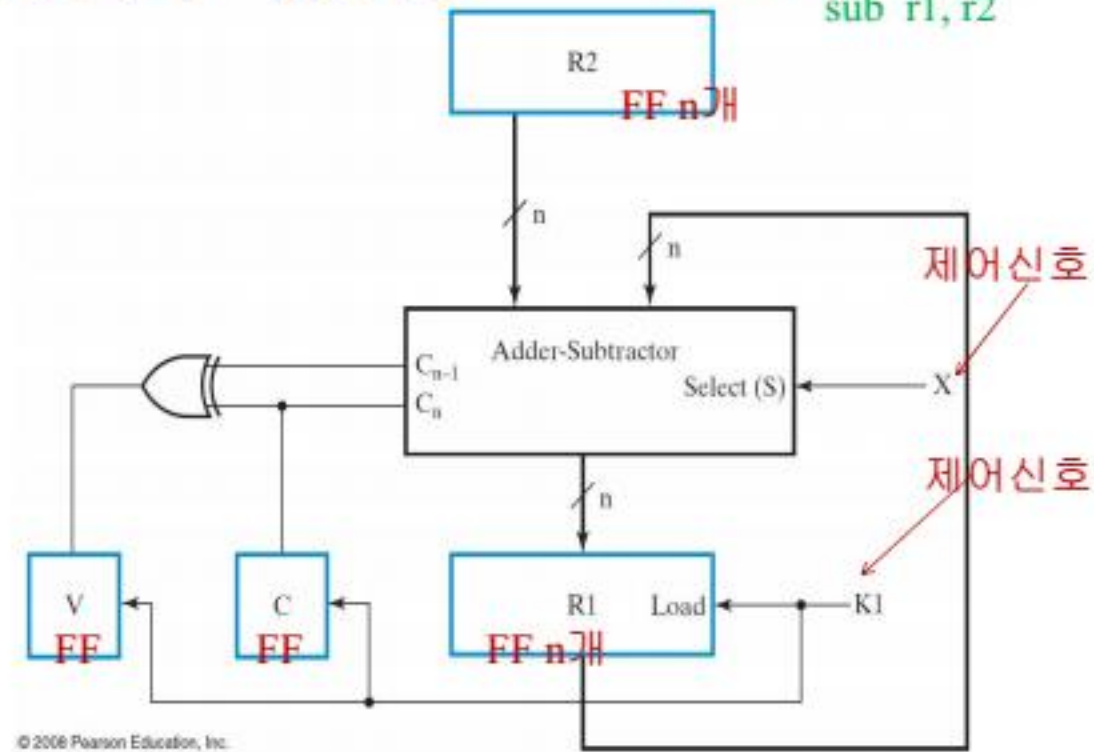


20

20

# 가감산기 p.318

기계어 : add r1, r2  
sub r1, r2

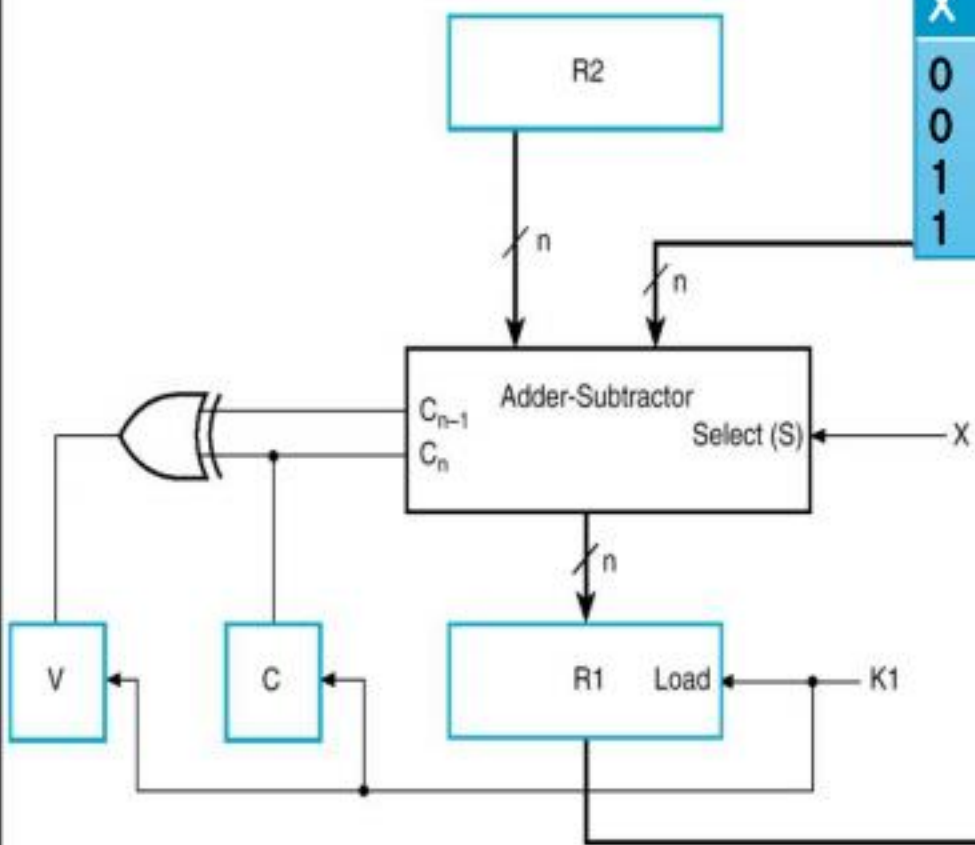


© 2008 Pearson Education, Inc.

21

21

## 가산 및 감산



제어신호		기능
X	K	
0	0	
0	1	
1	0	
1	1	

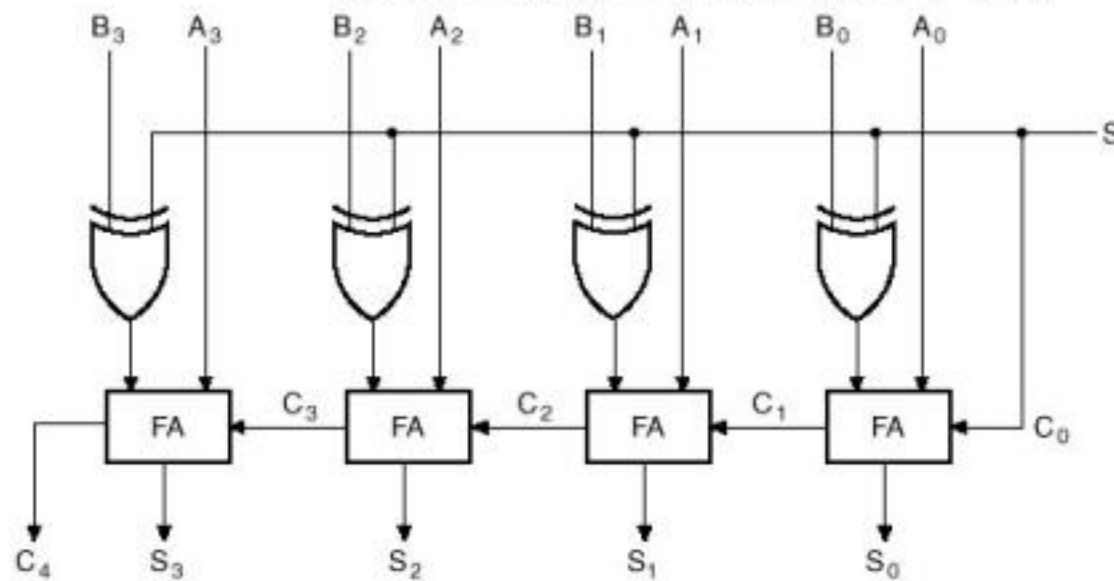
22

22

## 2진 가산기-감산기 p.155

$A + B$   
 $A - B$

- Signed integer 가산 + 감산
- 두 연산 대상 A와 B의 부호를 전혀 신경쓰지 않는다.  
부호비트까지 포함하여 계산한다.



23

23

## Full Adder ( FA ) p.148

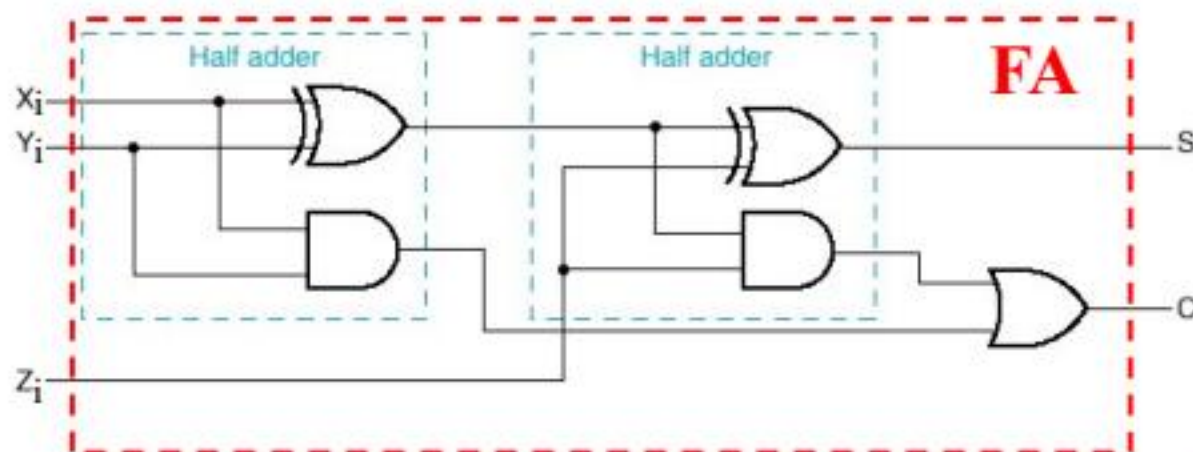
■ int x, y, r;

■ x = 13, y = 7;

■ r = x + y;

x : 0000 1101  
y : 0000 0111

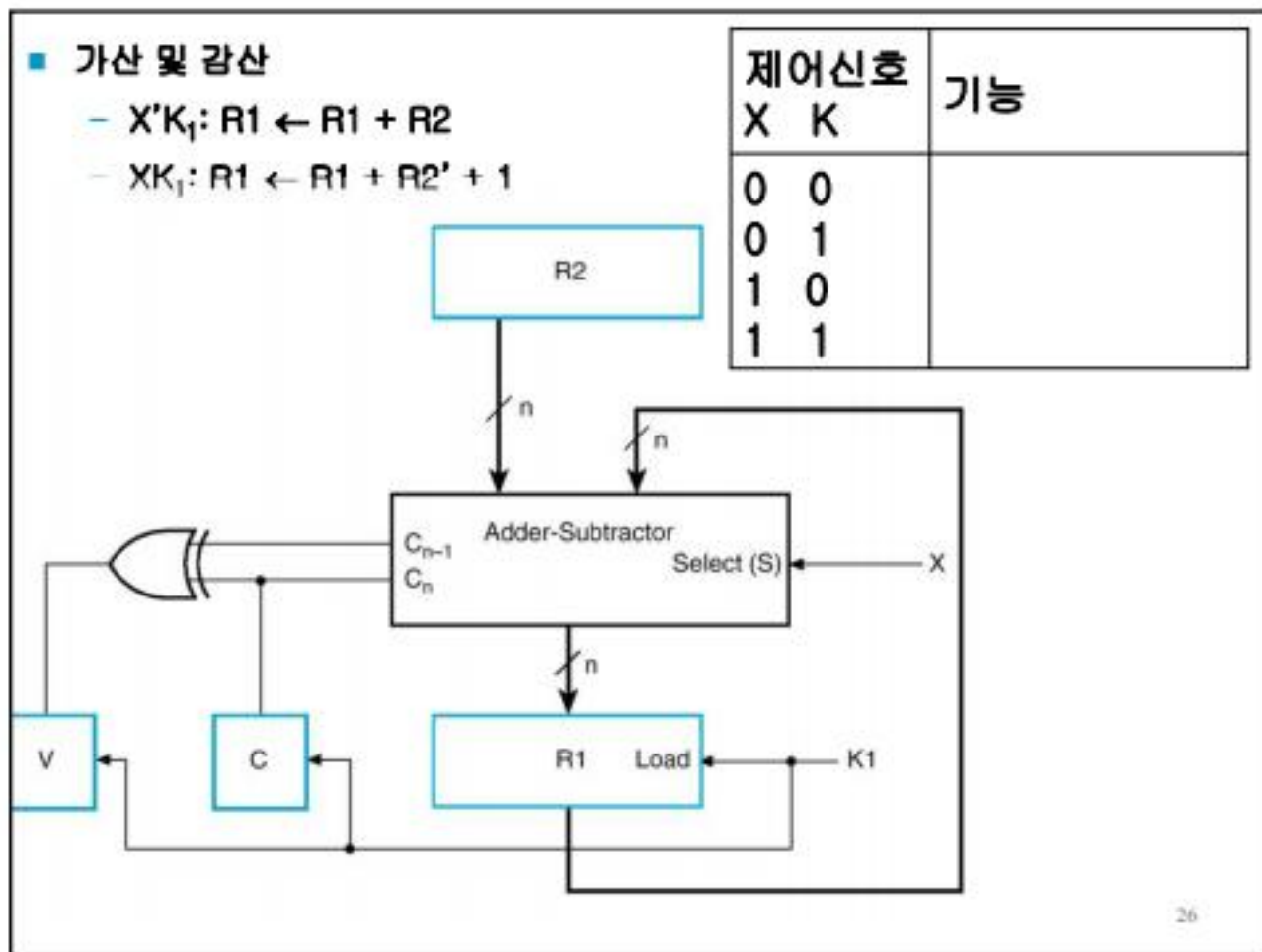
z : 0001 1110  
x : 0000 1101  
y : 0000 0111  
-----  
0001 0100



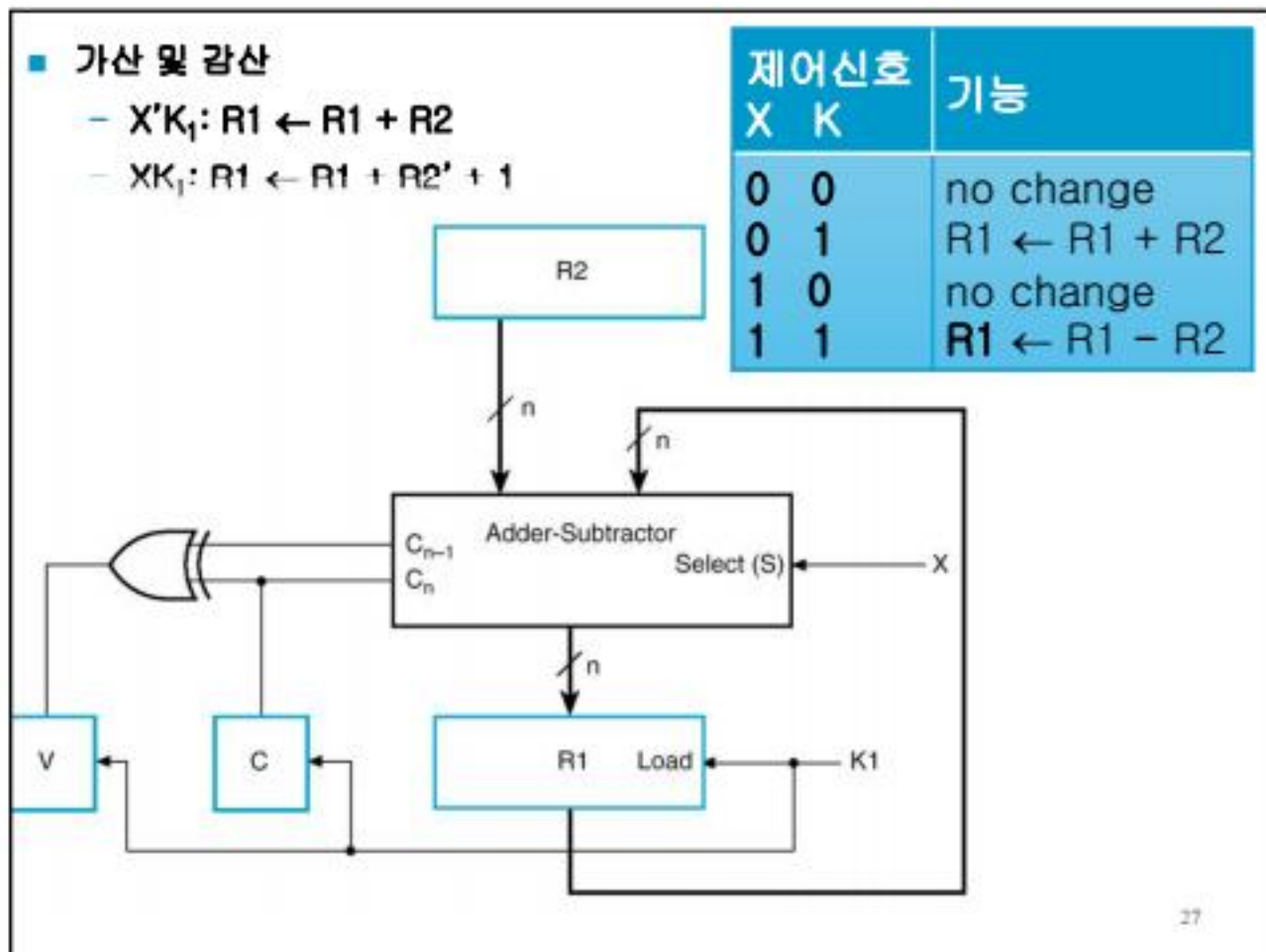
24

24



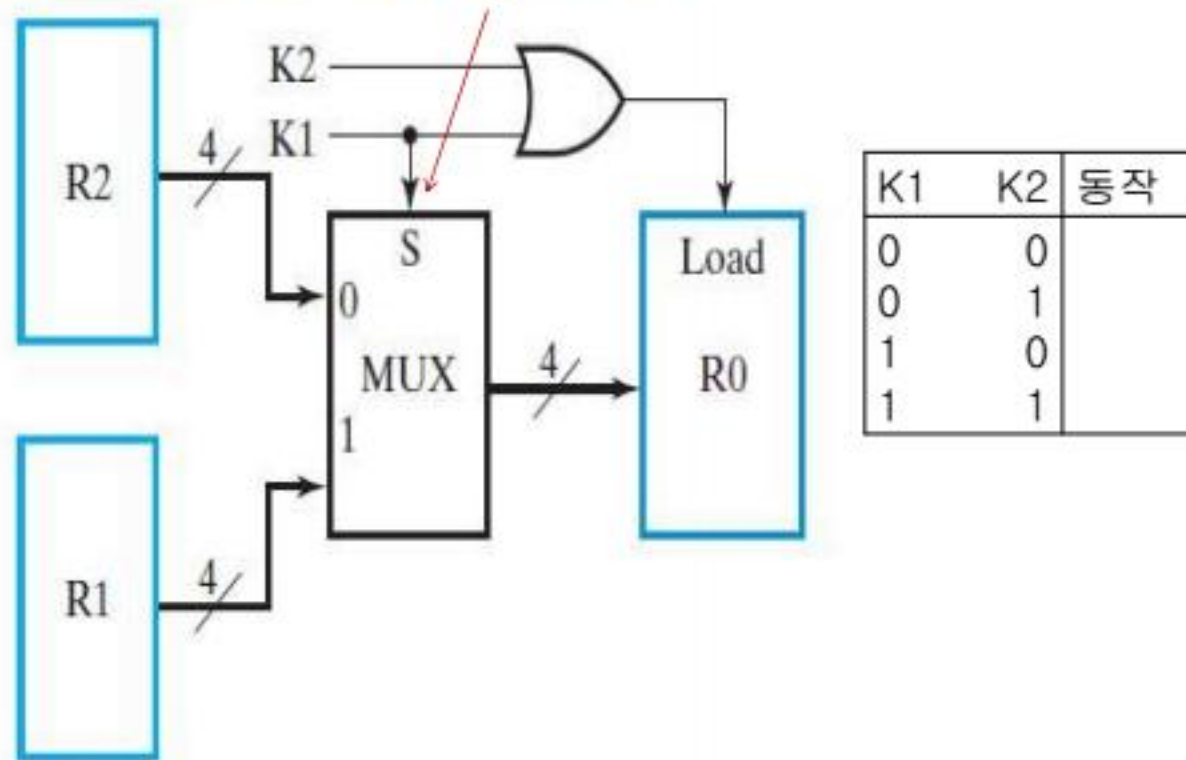


26



27

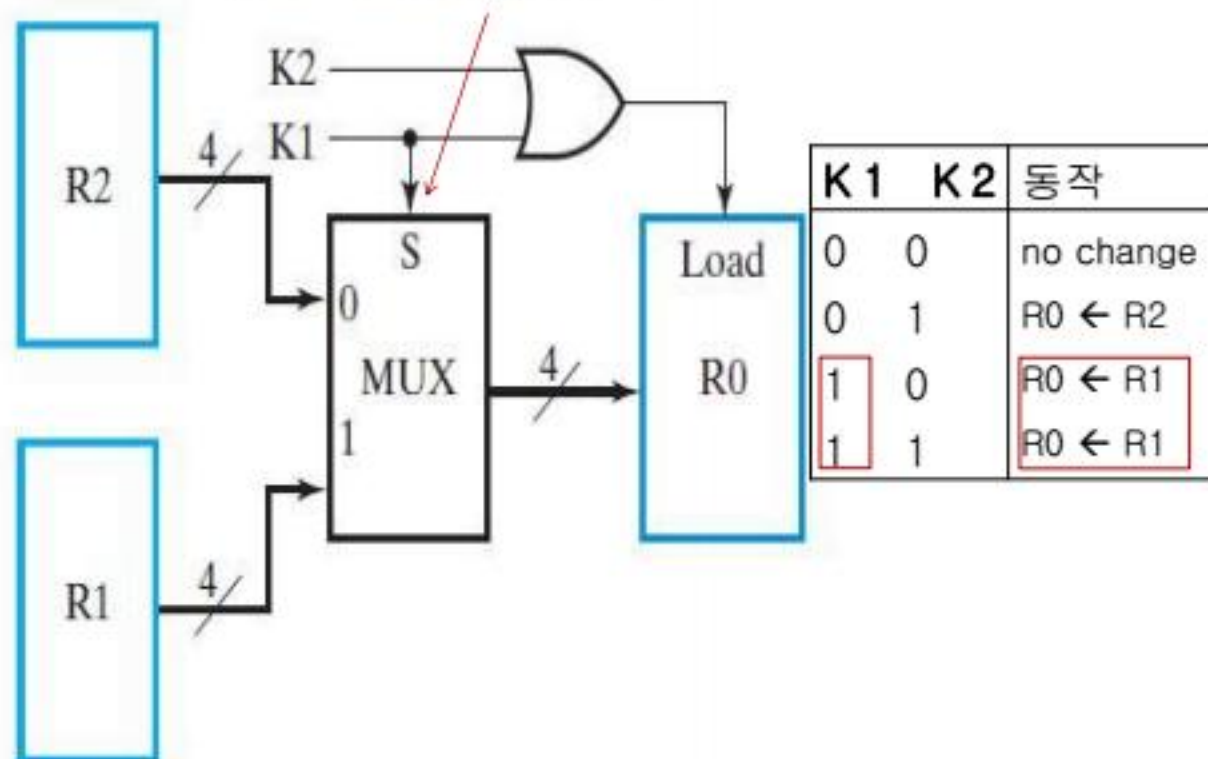
## MUX 기반 전송 p.323



28

28

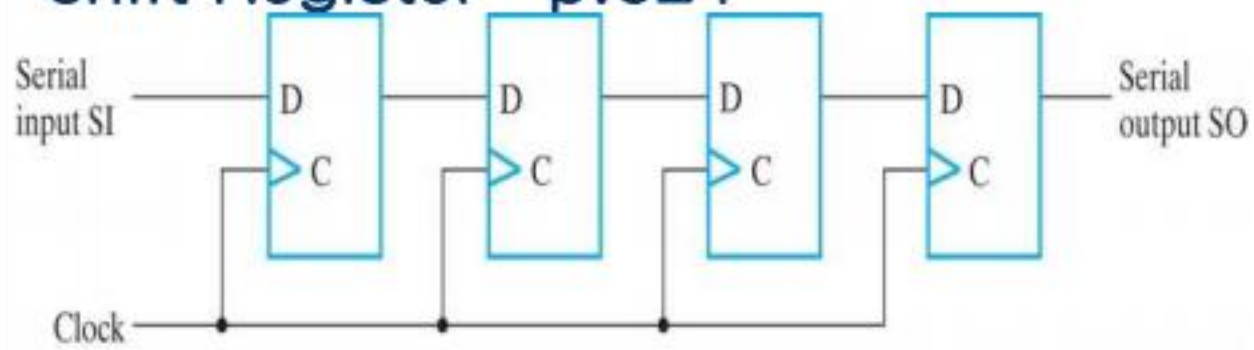
## MUX 기반 전송 p.323



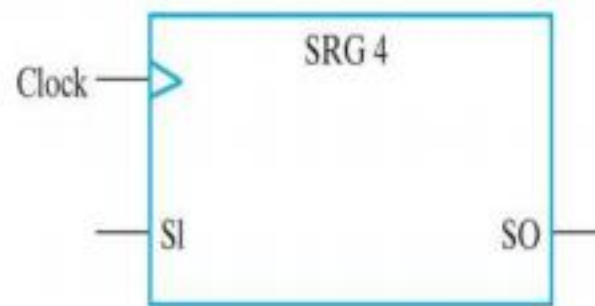
29

29

## shift Register p.324



(a) Logic diagram



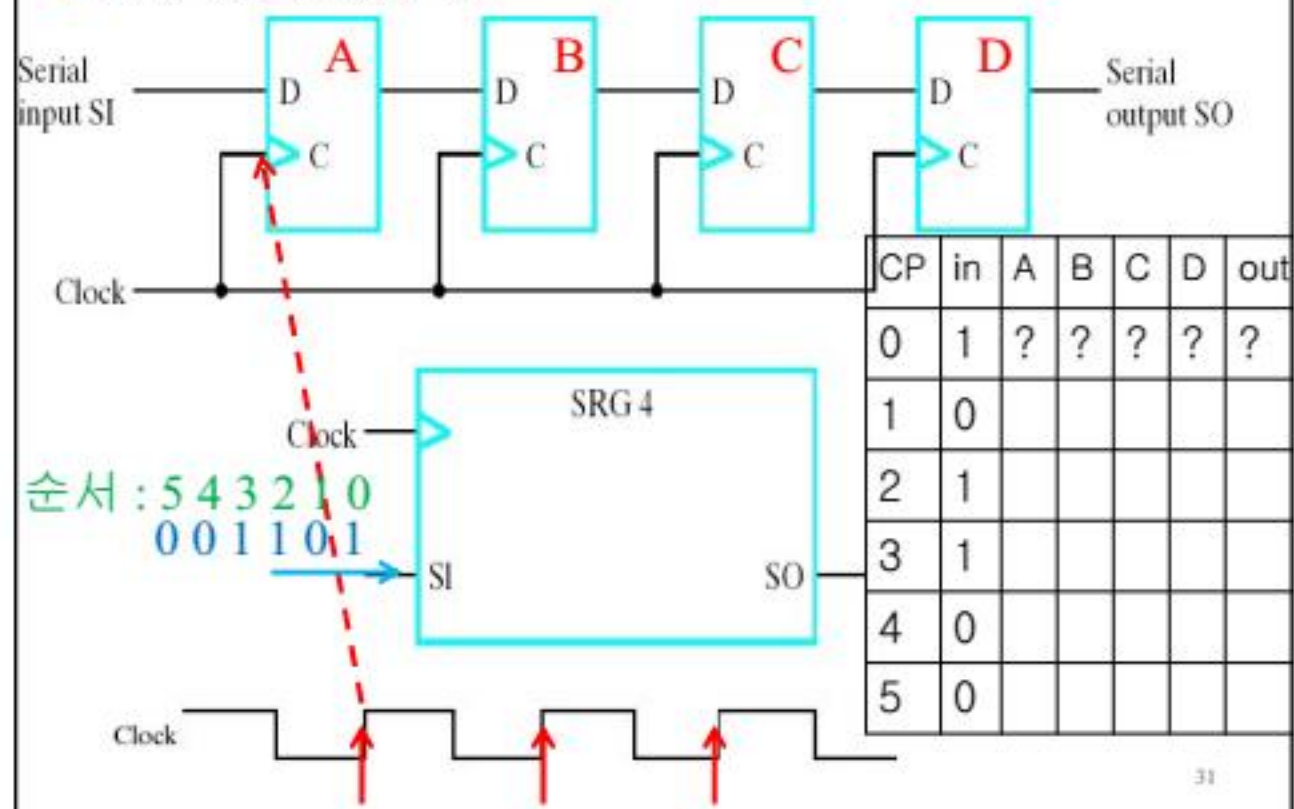
(b) Symbol

30

30

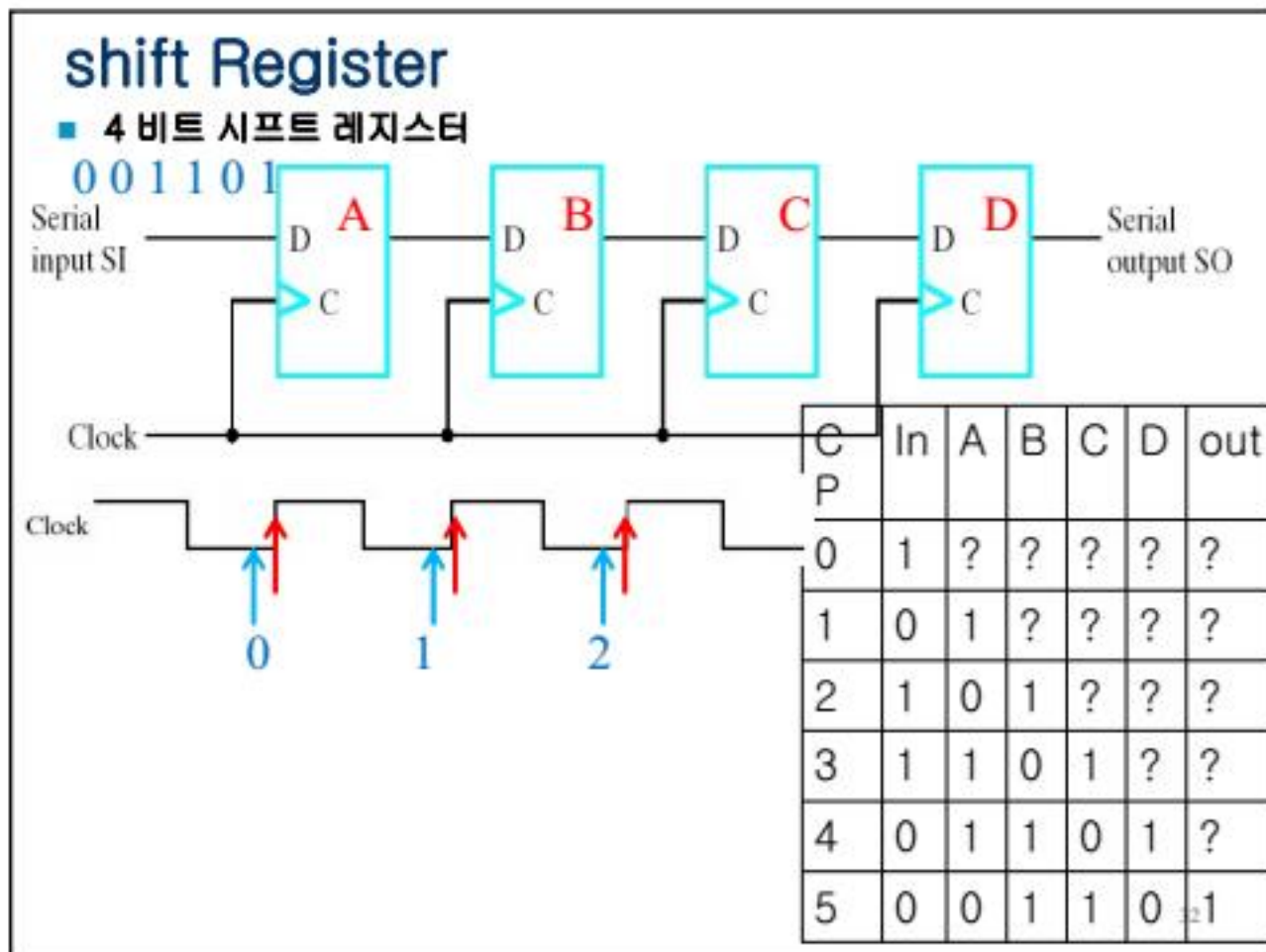
## shift Register

### ■ 4 비트 시프트 레지스터



31

31



32

## 카운터 설계

- 다음을 설계하자. <특정 순서의 카운터>
  - 000-001-010-100-101-110-000-... 이 순서로 반복해서 상태가 바뀐다.
  - 입력과 출력은 없다. clock의 positive edge마다 다음 상태로 바뀐다.
  - 초기상태는 000이다.
  - 가정 : 이 이외의 상태는 절대 발생하지 않는다.
  - positive edge triggered DFF 3개 사용 : 기호 A, B, C

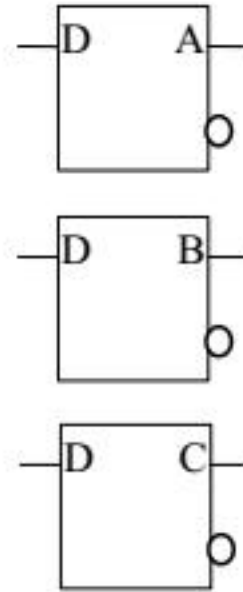
33

33



## 카운터 설계

- $DA = A'B + BC$
- $B = C$
- $C = B'C'$



34

34

## 정리

- 레지스터의 동작에 대한 이해
- 회로 분석
- 순차회로 설계

35

35