UNIT 3
D FLIP-FLOPS, REGISTER, COUNTER

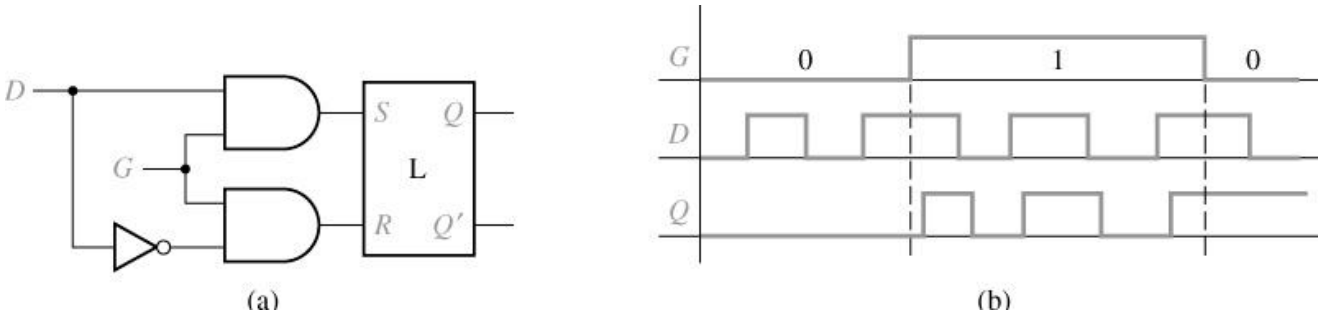
2012학년 2학기

마이크로 프로세서 실습

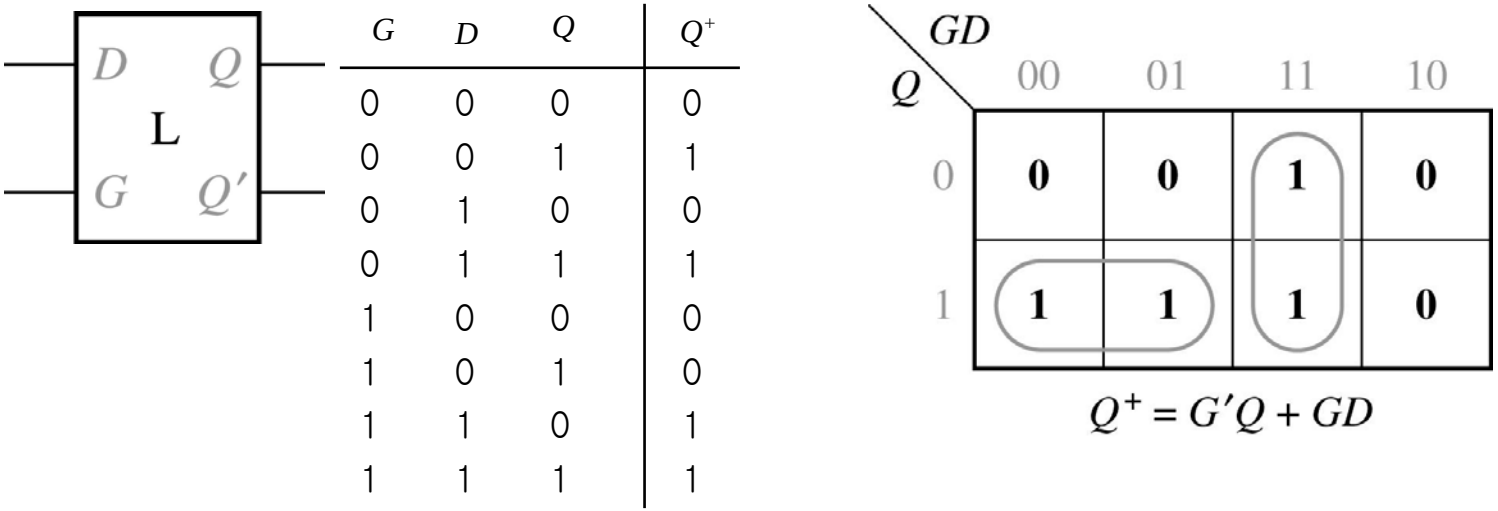
CONTENTS

1 D FLIP-FLOP

GATED D LATCH



Gated D Latch



G	D	Q	Q^+
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Q \ GD	00	01	11	10
0	**0**	**0**	**1**	**0**
1	**1**	**1**	**1**	**0**

$$Q^+ = G'Q + GD$$

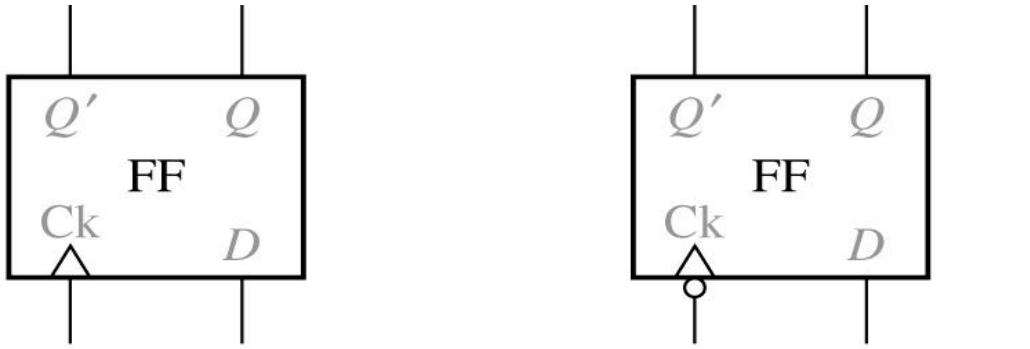
Symbol and Truth Table for Gated Latch

EDGE-TRIGGERED D FLIP-FLOP [1 OF 3]

- D Latch와 달리 D Flip-Flop은 data input (D)와 Clock(Ck)로 구성
- D Flip-Flop은 input의 변화가 아닌 Clock에 반응하여 변화
- 출력이 clock 입력의 0에서 1로 전환 시에 변화한다면 Flip-Flop은 clock의 rising-edge에서 trigger
- 출력이 clock 입력의 0에서 1로 전환 시에 변화한다면 Flip-Flop은 clock의 falling-edge에서 trigger

EDGE-TRIGGERED D FLIP-FLOP [2 OF 3]

- Clock 입력에 반전 표시 방울이 있는 Flip-Flop은 falling edge trigger Flip-Flop
- 방울이 없는 경우 rising edge Filp-Flop
- Active edge(활성 에지)라는 용어는 Flip-Flop의 상태 변화를 촉발하는 clock edge(falling or rising)을 지칭한다.



(a) Rising-edge trigger (b) Falling-edge trigger

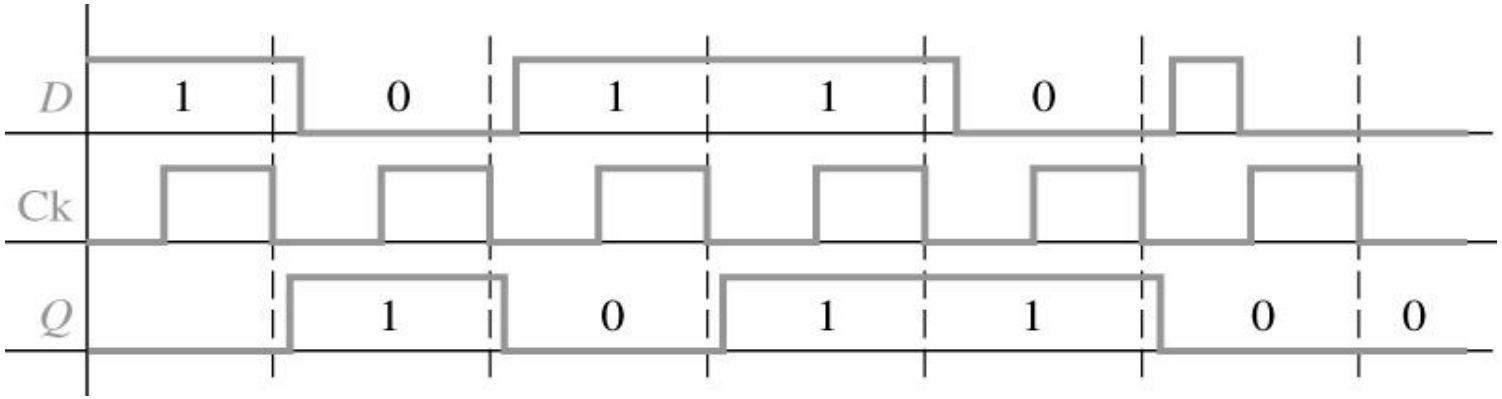
D Flip-Flops

EDGE-TRIGGERED D FLIP-FLOP [3 OF 3]

- Active Clock edge 후의 D Flip-Flop의 상태(Q+)는 Active edge에서의 input D 와 같다

D	Q	Q^+
0	0	0
0	1	0
1	0	1
1	1	1

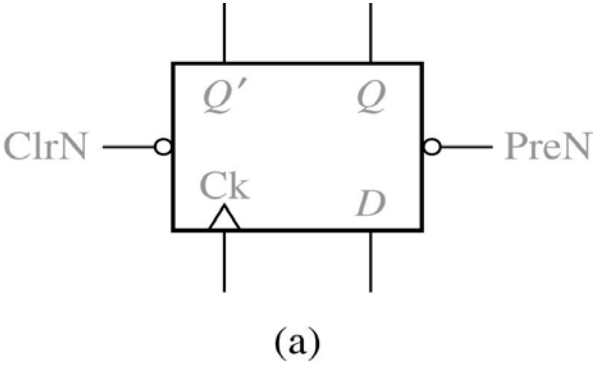
$$Q^+ = D$$



Timing for D Flip-Flop(falling edge trigger)

Flip-Flops with Additional Inputs [1 of 4]

- Flip-Flop은 Clock과는 독립적으로 Flip-Flop을 initial state로 만들기 위한 부가적인 입력을 가질 수 있다.
- Initial state란 0 or 1의 상태를 나타낸다.
- Initial state 0으로 만드는 input : Clear (Clr)
- Initial state 1로 만드는 Input : Preset(pre)



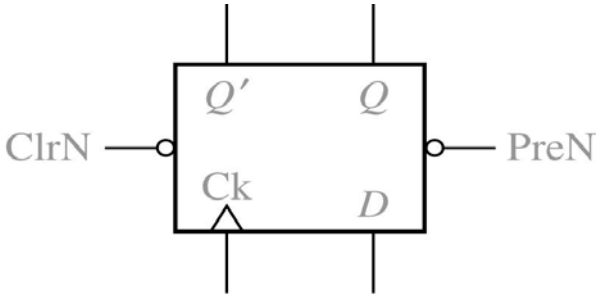
(a)

Ck	D	PreN	ClrN	Q^+
x	x	0	0	(not allowed)
x	x	0	1	1
x	x	1	0	0
↑	0	1	1	0
↑	1	1	1	1
0,1,↓	x	1	1	Q(no change)

D Flip-Flop with Clear and Preset

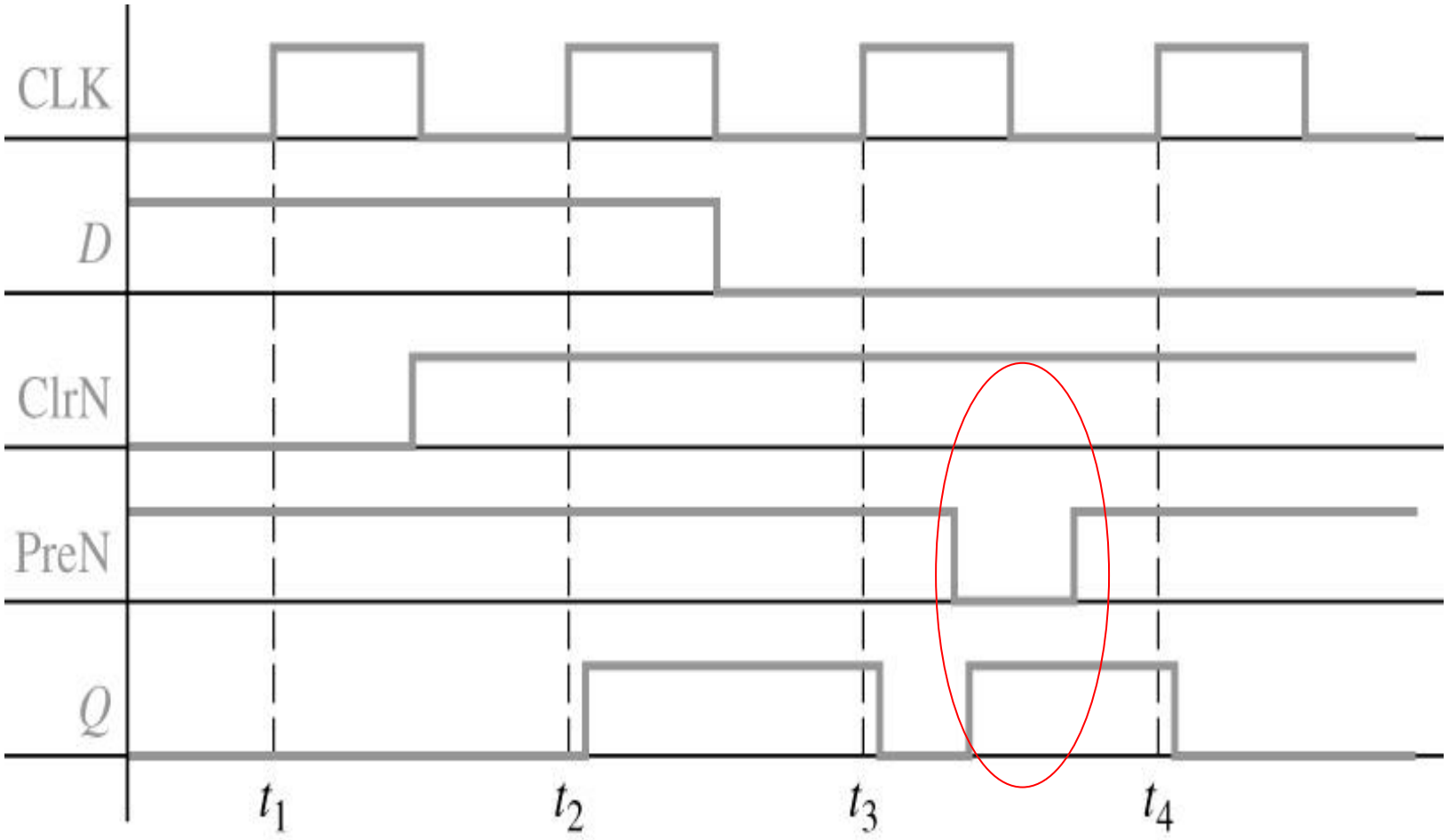
Flip-Flops with Additional Inputs [2 of 4]

- 밑의 D Flip-Flop은 부가 입력신호 ClrN과 PreN은 작은 방울(반전 표시)을 가지고 있기 때문에 Flip-Flop을 1로 set하거나 0으로 clear시키기 위해서 1이 아닌 0이 입력 되야 한다는 것을 나타낸다
- 이를 active low(저레벨 활성)이라고 부른다
- Clear + active low = ClrN
- Preset + active low = PreN
- 따라서 ClrN에 0이 입력되면 output Q는 0이 되고, PreN에 0이 입력되면 output Q는 1이 된다
- 이 입력들은 D input이나 Clock input에 상관 없이 비동기적으로 동작한다.



(a)

Flip-Flops with Additional Inputs [3 of 4]



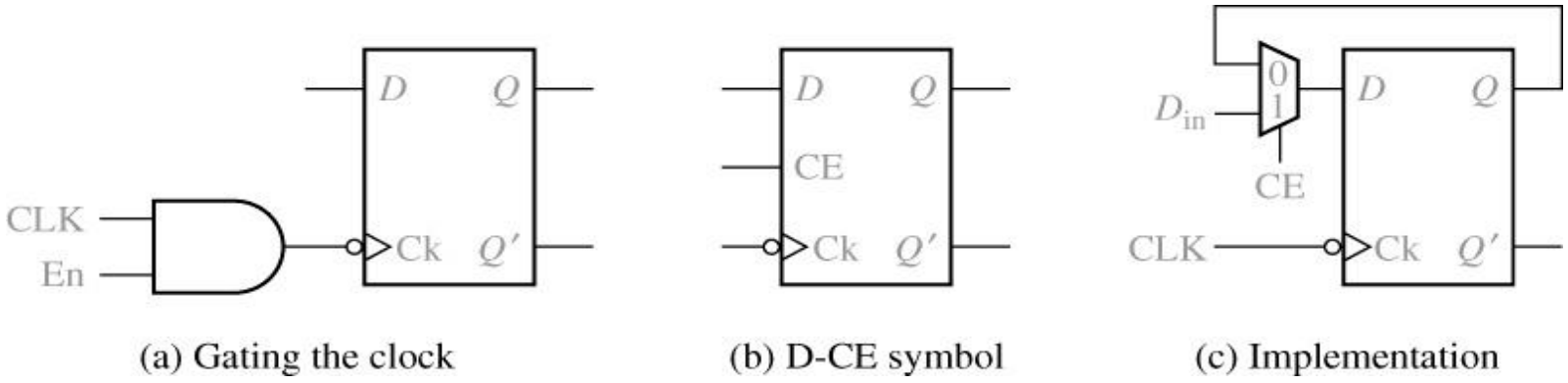
Timing Diagram for D Flip-Flop with Asynchronous Clear and Preset

Flip-Flops with Additional Inputs [4 of 4]

- Flip-Flop의 enable(En), clock(Ck) 이 변하더라도 이전 데이터 값 유지 필요시
  - Clock 자체의 동작을 gating 하는 방식 : clock 지연 발생의 문제점 (a)
  - Clock Enable (CE) 을 사용하는 방식 : CE = 1에서만 Flip-Flop 회로 동작 (b)

The characteristic equation : $Q^+ = Q \cdot CE' + D \cdot CE$

The MUX output : $Q^+ = D = Q \cdot CE' + D_{in} \cdot CE$



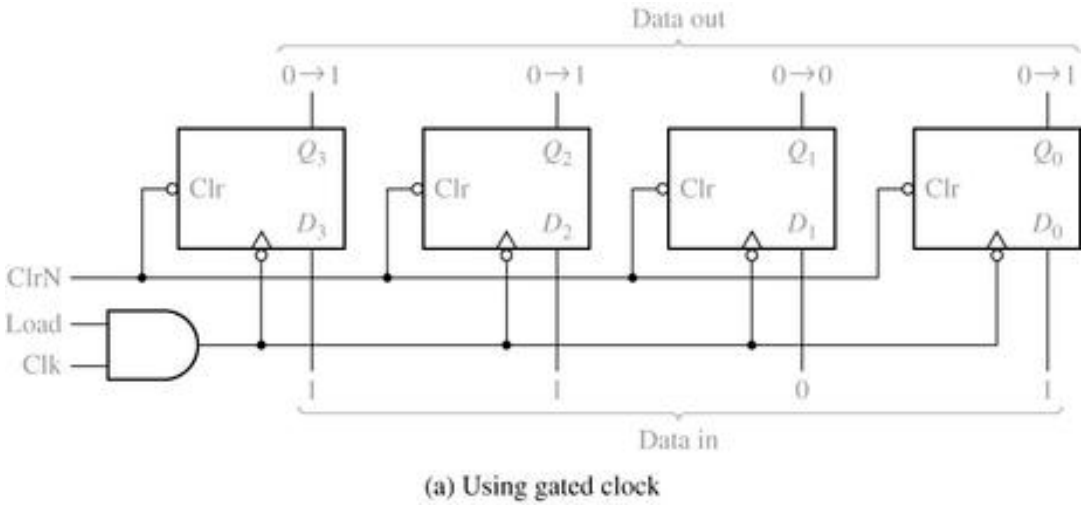
(a) Gating the clock (b) D-CE symbol (c) Implementation

D Flip-Flop with Clock Enable

2 REGISTER

REGISTERS AND REGISTER TRANSFERS [1 OF 8]

- 여러 개의 D Flip-Flop을 하나의 common clock 으로 묶어 하나의 register를 형성 (binary data 저장 및 shift 역할)
- Load, Clk, ClrN 신호의 의미와 역할을 이해
- Clk를 Load 신호와 AND gate로 묶었을 때의 문제점

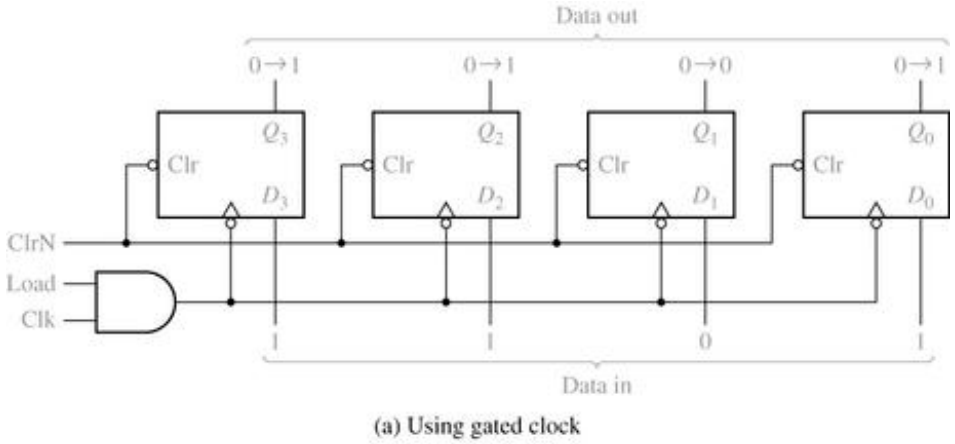


(a) Using gated clock

4-bit D Flip-Flop register

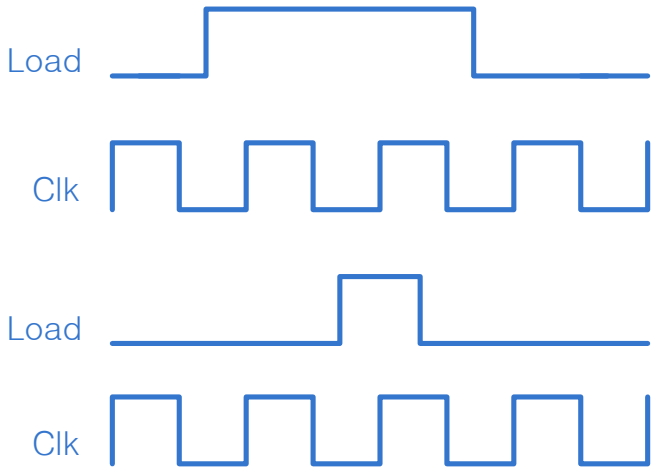
REGISTERS AND REGISTER TRANSFERS [2 OF 8]

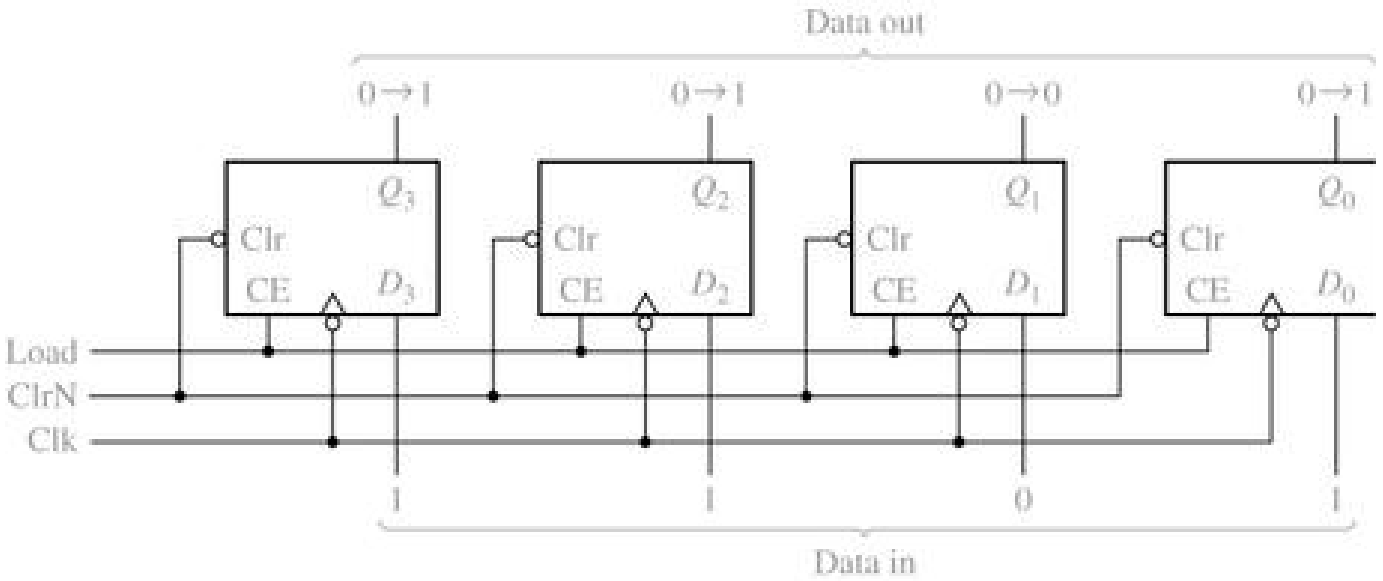
- (Load = 1, Clk ↓) D -> Q
- (Load = 0) Data out 은 유지
- Gated Clk의 경우 clock delay, timing 문제 발생 -> Clock Enable 방식으로 해결



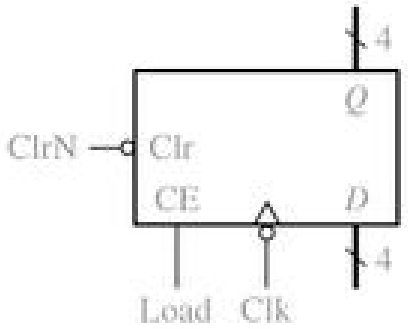
(a) Using gated clock

4-bit D Flip-Flop register



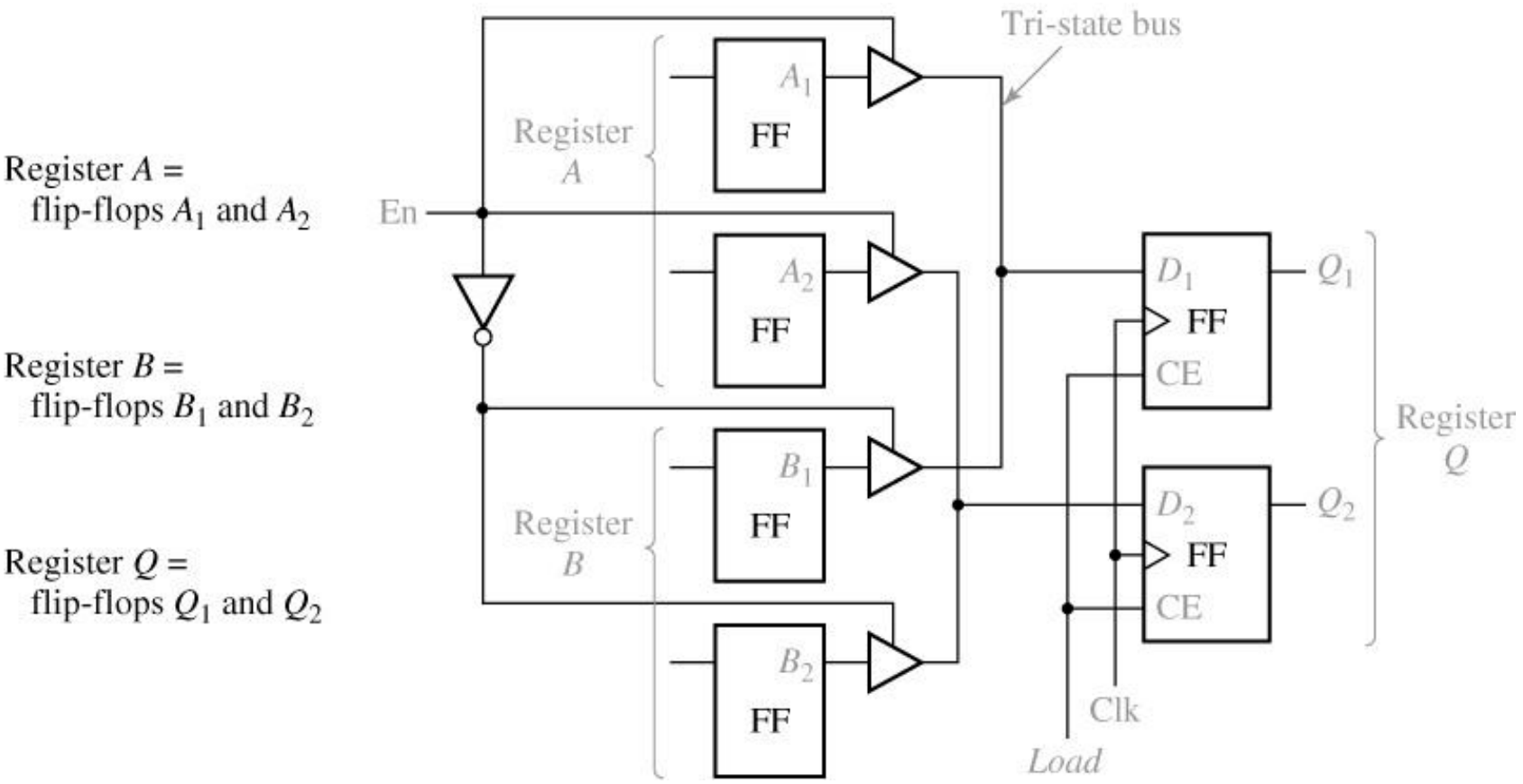


(b) With clock enable

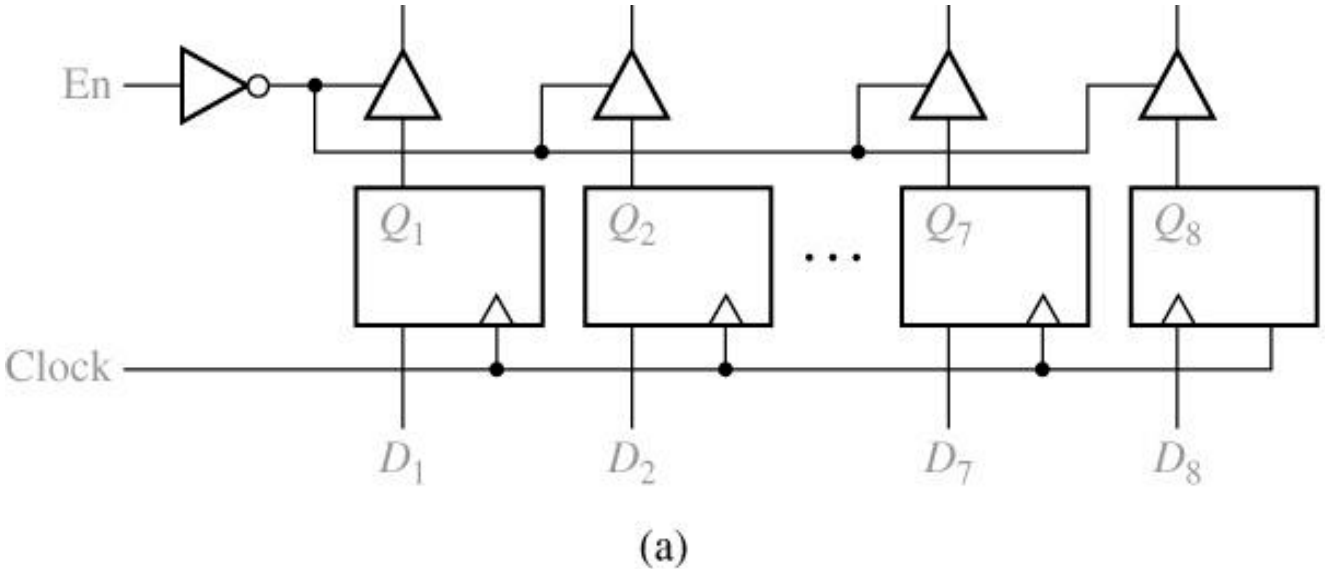


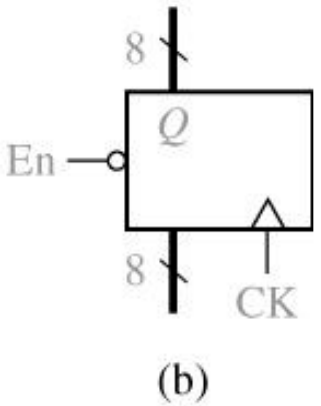
(c) Symbol

- Data Transfer Between Registers
  - 3-state bus 이용, A/B 두 개의 register중 출력 선탱
  - (Load =1, En = 1, Clk↑) A-> Q, (Load = 1, En = 0, Clk ↑) B -> Q



REGISTERS AND REGISTER TRANSFERS [5 OF 8]

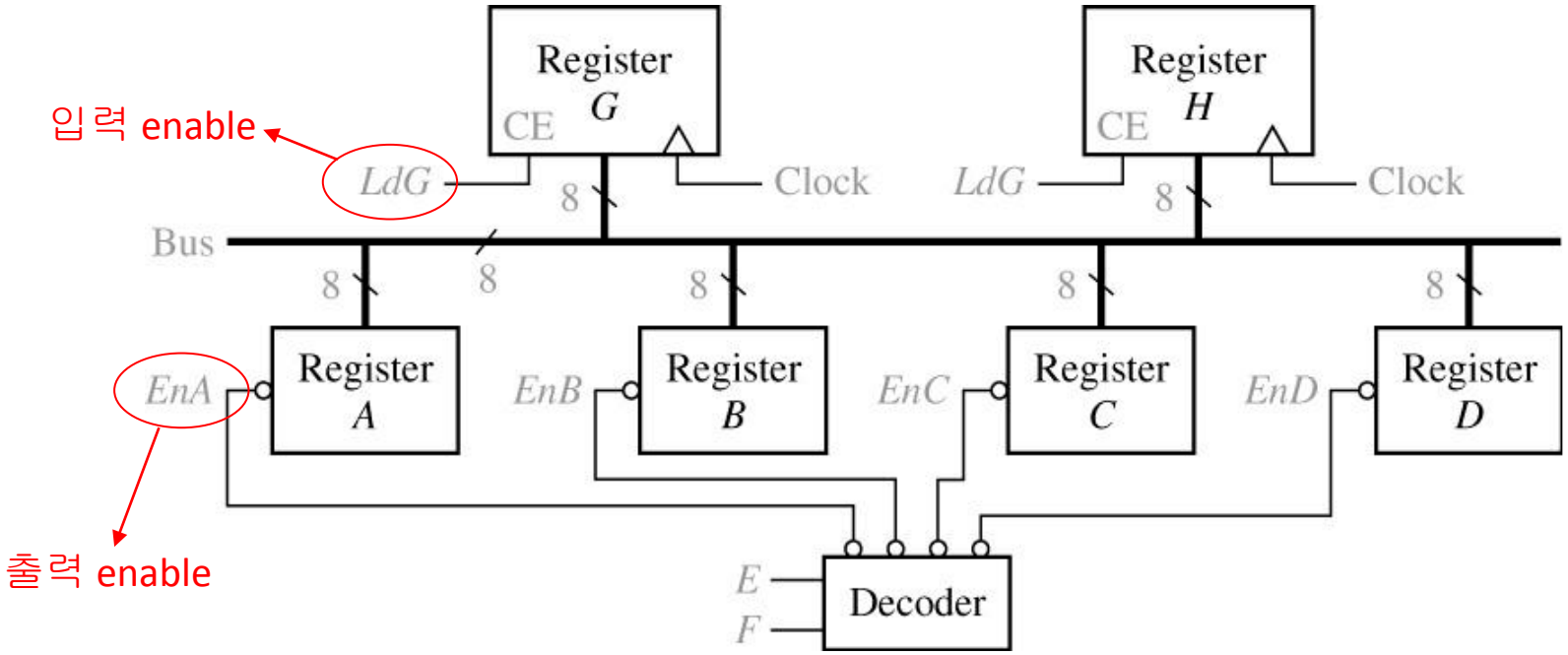




Logic Diagram for 8 bit register with Tri-state output

When En=0, D → Out

REGISTERS AND REGISTER TRANSFERS [6 OF 8]

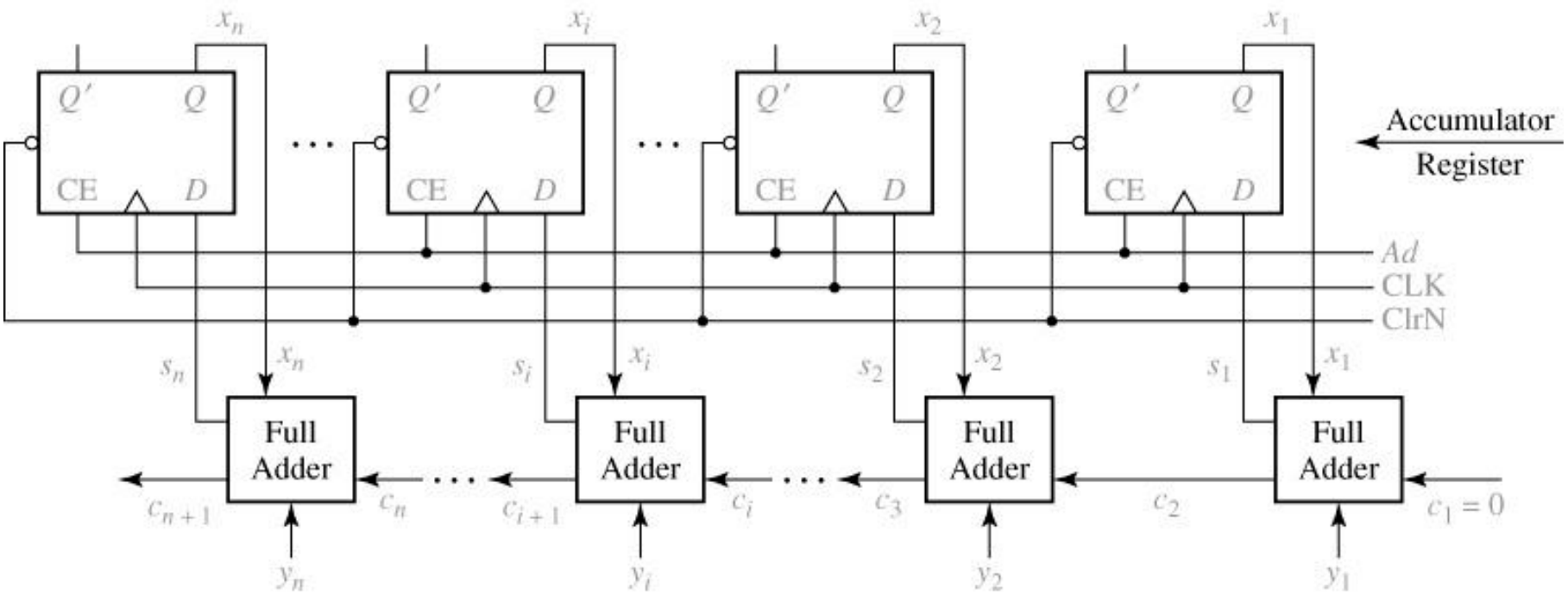


Data Transfer using a Tri-State Bus

Decoder 입력인 E,F 에 의해 레지스터 A, B, C, D 값을 G 혹은 H 에 저장

$$If\ EF = 00, A \rightarrow G(LdG = 1)\ H(LdH = 1)$$
$$If\ EF = 01, B \rightarrow G(LdG = 1)\ H(LdH = 1)$$
$$If\ EF = 10, C \rightarrow G(LdG = 1)\ H(LdH = 1)$$
$$If\ EF = 11, D \rightarrow G(LdG = 1)\ H(LdH = 1)$$

REGISTERS AND REGISTER TRANSFERS [7 OF 8]

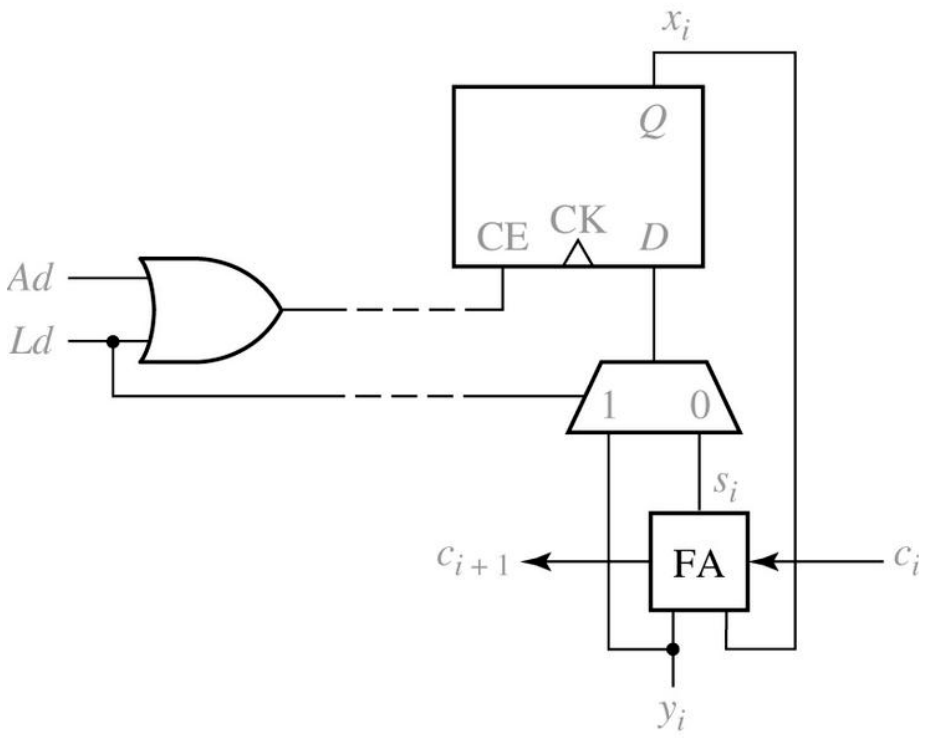


N-bit Parallel Adder with Accumulator

레지스터에 숫자(x_i)를 저장하고, y_n 더해 그 결과를 다시 accumulator (x_i) 에 저장

$(x_i + y_i)$ ➔ x_i

REGISTERS AND REGISTER TRANSFERS [8 OF 8]



Adder Cell with Multiplexer

$x_i + y_i = s_i$ ➔ x_i

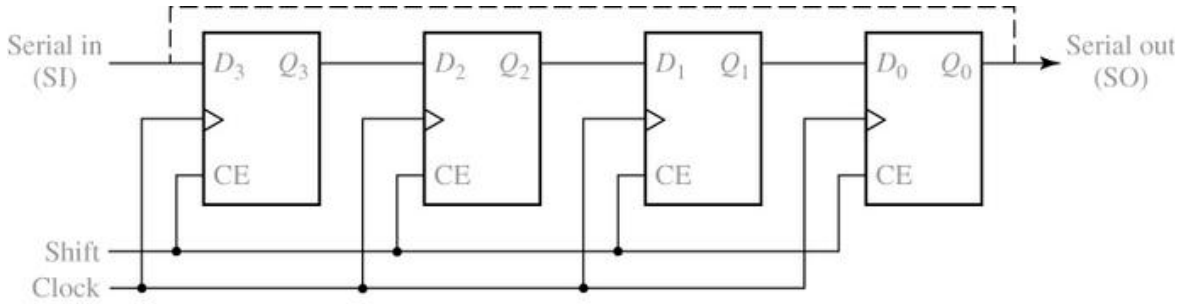
(Data Load)
@ Ld =1 , Ad = don't care, Clk ↑
y_i ➔ Multiplexer Out ➔ x_i
input data is saved at DFF

(addition)
@ Ld =0 , Ad=1, Clk ↑
y_i renewed
$x_i + y_i = s_i$

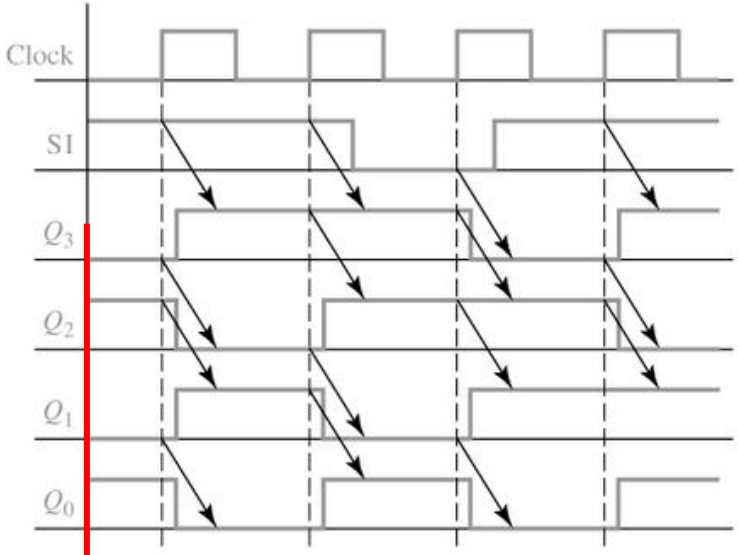
(Data Accumulation)
@ Ld = 0, Ad = 1, Clk ↑
s_i ➔ x_i
$x_i + y_i = s_i$ for each Clk ↑

SHIFT REGISTERS [1 OF 8]

Binary data 저장 후, 매 clock 마다 data를 왼쪽 혹은 오른쪽으로 이동



(a) Flip-flop connections



(b) Timing diagram

Initially 0101

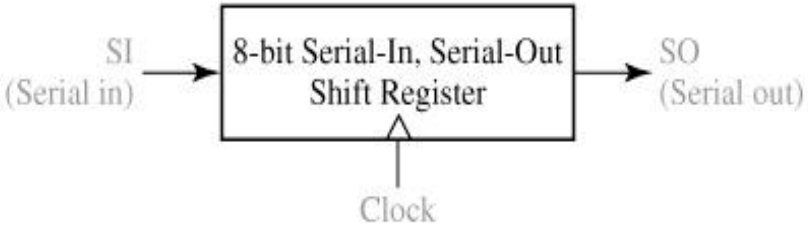
Right-Shift Register

@shift=1
shift right for each clock rising
@shift=0
output unchanged
초기상태: 0101
SI(serial in): 1

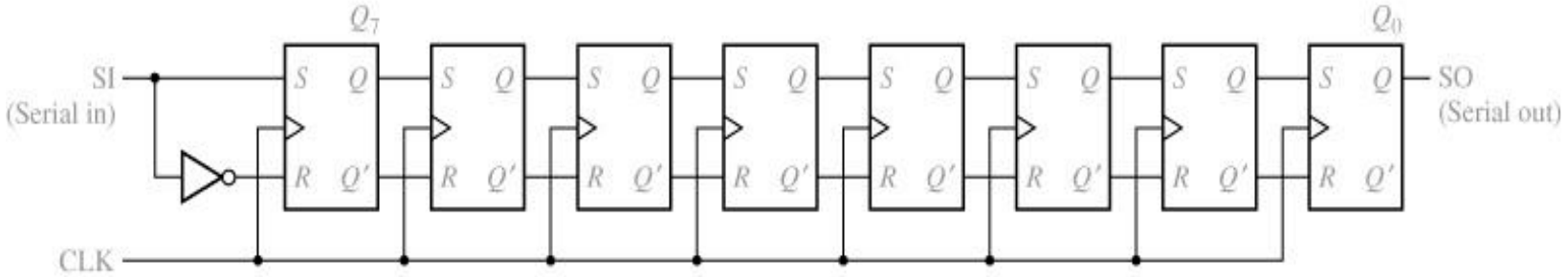
0101→1010→1101→0110→1011

SHIFT REGISTERS [2 OF 8]

- S-R Latch 를 이용한 Shift register 구성
- Serial-in : Data가 한번에 한 bit씩 첫 Flip-Flop으로 shift (cf. parallel in/out)
- Serial-out : Data를 마지막 Flip-Flop으로 부터 읽을 수 있음



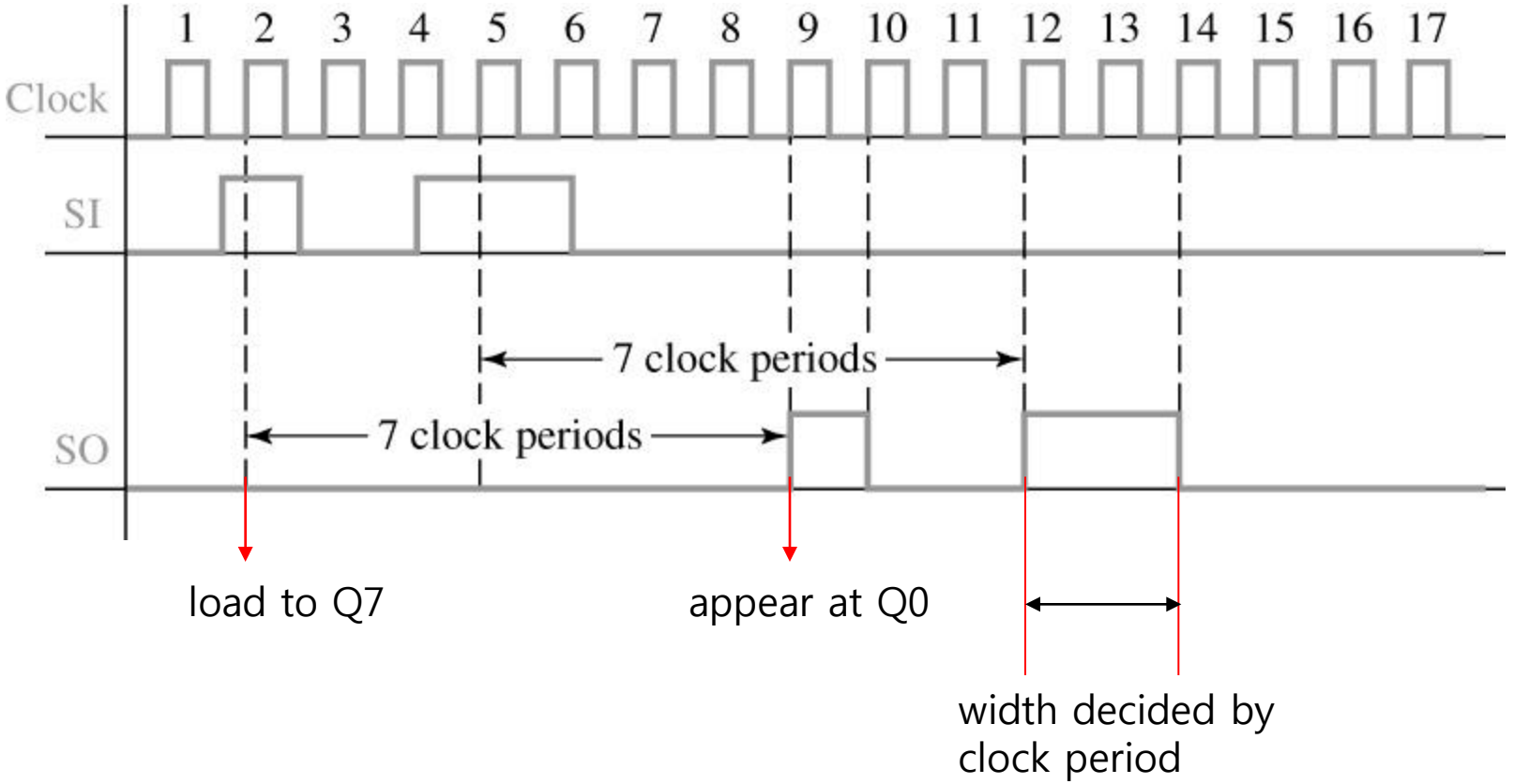
(a) Block diagram



(b) Logic diagram

8-bit serial-in, serial-out shift register

SHIFT REGISTERS [3 OF 8]

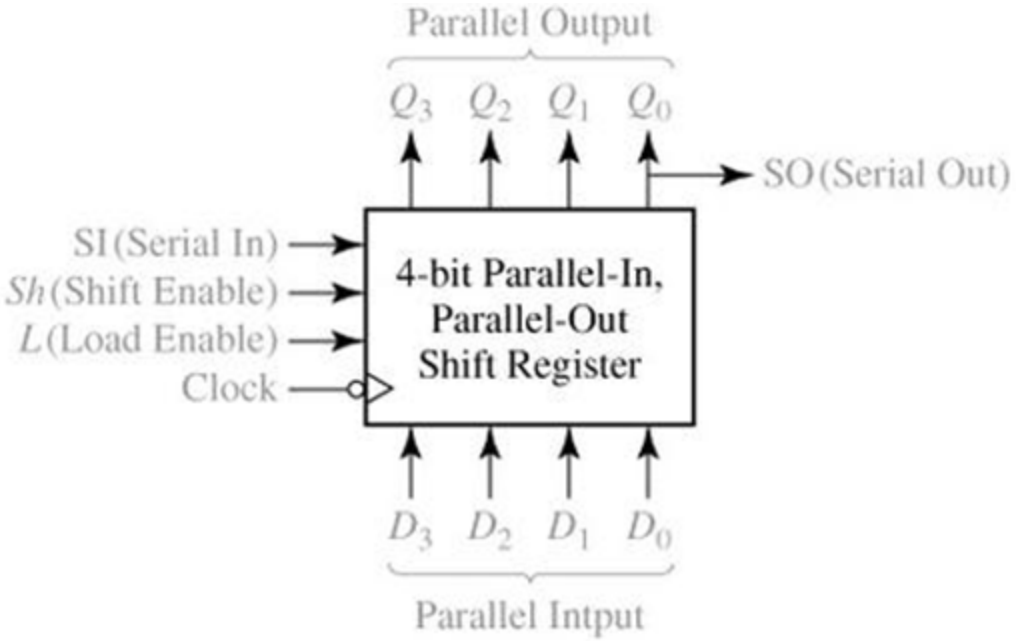


Typical timing diagram for shift register

Serial in 이후 8번째 클럭(즉, 7개 클럭 주기 후) 상승 에지에서 serial-out

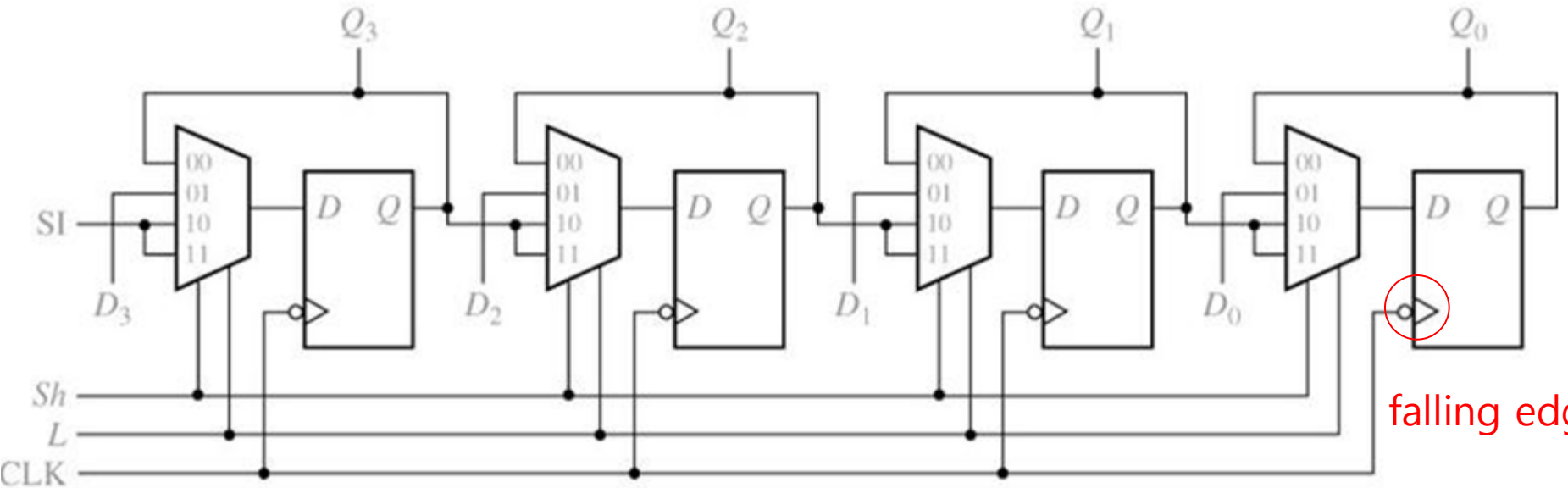
SHIFT REGISTERS [4 OF 8]

- 여러 개의 data 동시 load 및 out



Parallel-in, Parallel-out right shift register

SHIFT REGISTERS [5 OF 8]

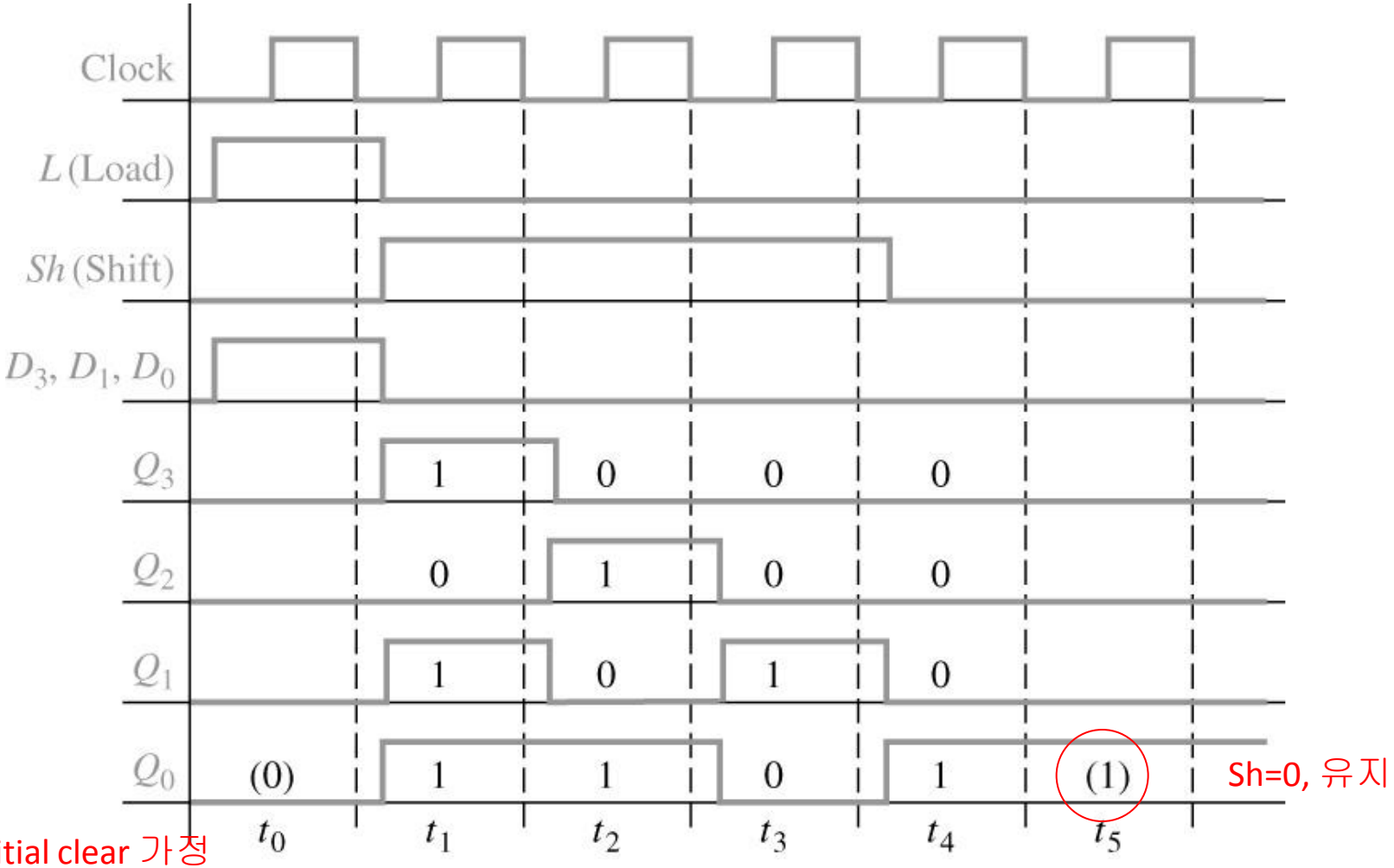


Parallel-in, Parallel-out right shift register

Input		Next State				Action
Sh(Shift)	L(Load)	Q_3^+	Q_2^+	Q_1^+	Q_0^+	
0	0	Q_3	Q_2	Q_1	Q_0	no change
0	1	D_3	D_2	D_1	D_0	load
1	×	SI	Q_3	Q_2	Q_1	right shift

Load = 1 Load i/p
Shift = 1 shift right
용도 : Parallel-data를 serial-data 로 변환
Q) Sh=L=0 에서 현 상태를 유지 하는 이유는?

SHIFT REGISTERS [6 OF 8]



Timing Diagram for shift register

SHIFT REGISTERS [7 OF 8]

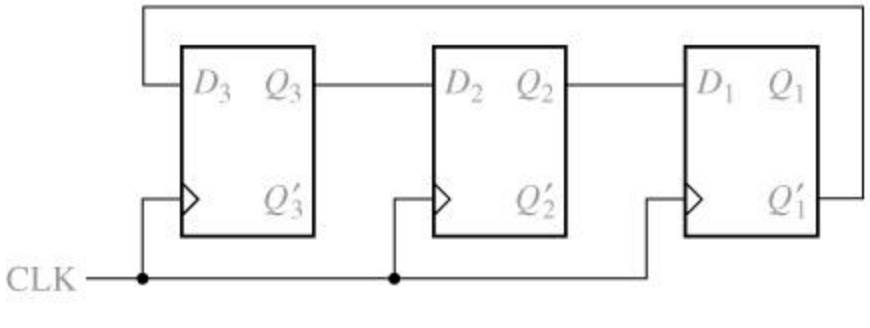
Input		Next State				Action
Sh(Shift)	L(Load)	Q_3^+	Q_2^+	Q_1^+	Q_0^+	
0	0	Q_3	Q_2	Q_1	Q_0	no change
0	1	D_3	D_2	D_1	D_0	load
1	×	SI	Q_3	Q_2	Q_1	right shift

$$Q_3^+ = Sh' \cdot L' \cdot Q_3 + Sh' \cdot L \cdot D_3 + Sh \cdot \mathrm{SI}$$
$$Q_2^+ = Sh' \cdot L' \cdot Q_2 + Sh' \cdot L \cdot D_2 + Sh \cdot Q_3$$
$$Q_1^+ = Sh' \cdot L' \cdot Q_1 + Sh' \cdot L \cdot D_1 + Sh \cdot Q_2$$
$$Q_0^+ = Sh' \cdot L' \cdot Q_0 + Sh' \cdot L \cdot D_0 + Sh \cdot Q_1$$

The next-state equations for the Flip-Flot

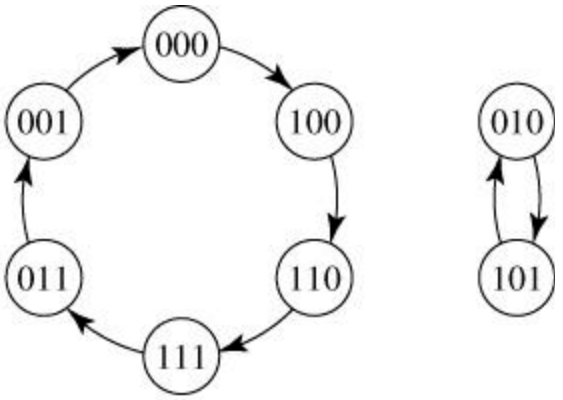
SHIFT REGISTERS [8 OF 8]

- Shift register with inverted feedback -> johnson counter



(a) Flip-flop connections

3-bit shift register



(b) State graph

Successive states

마지막 FF 의 반전된 출력을 첫 FF 입력으로 피드백
두 개의 루프 존재 가능
. 000 => 100 => 110 => 111 => 011 => 001 => 000 …
. 010 => 101 => 010 => 101

3 COUNTER

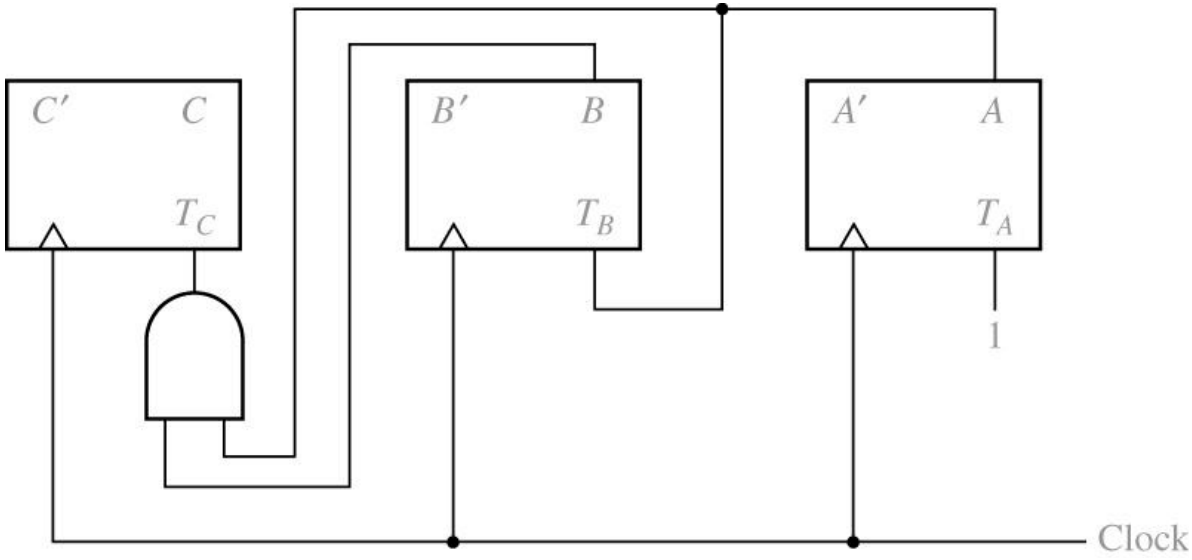
DESIGN OF BINARY COUNTERS [1 OF 14]

- Synchronous counter(동기식 카운터)
  - Flip-Flop입력이 공통입력 펄스에 의해 동기화 되는 counter

<C> changes
@ <A>=1, <B>=1
and CLK ↑

<B> changes
@ <A>=1, CLK ↑

T_A=1
<A> changes
for each CLK ↑

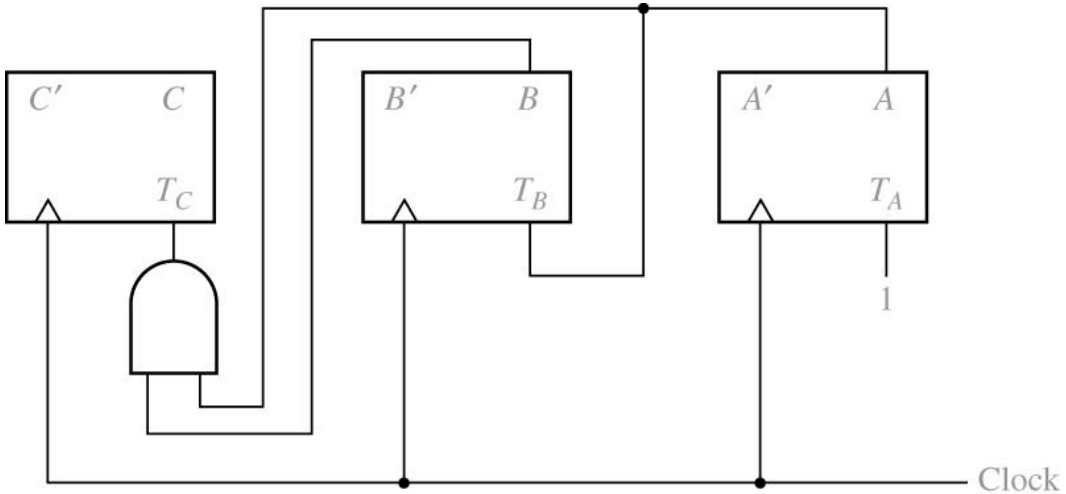


Synchronous Binary Counter

Counting sequence

CBA: 000→001→010→011→100→101→110→111→000

DESIGN OF BINARY COUNTERS [2 OF 14]



현재상태			차기상태		
C	B	A	C+	B+	A+
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

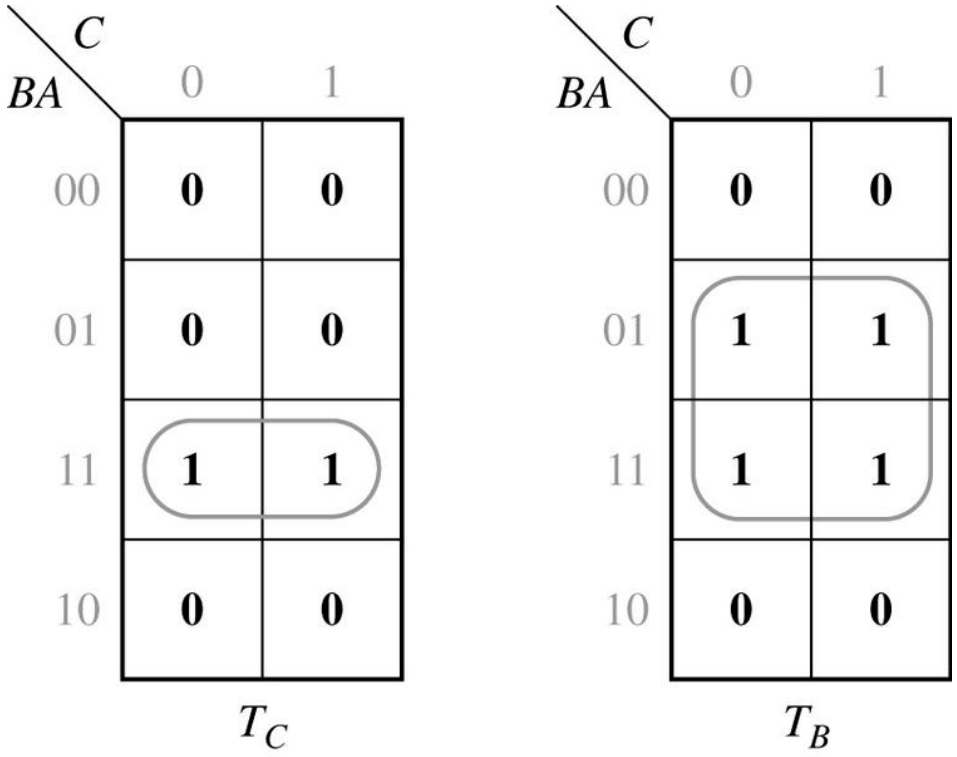
DESIGN OF BINARY COUNTERS [3 OF 14]

현재상태			차기상태			플립플롭 입력		
C	B	A	C+	B+	A+	T_C	T_B	T_A
0	0	0	0	0	1			
0	0	1	0	1	0			
0	1	0	0	1	1			
0	1	1	1	0	0			
1	0	0	1	0	1			
1	0	1	1	1	0			
1	1	0	1	1	1			
1	1	1	0	0	0			

State Table for Binary Counter

Design of Binary Counters [4 of 14]

플립플롭 입력		
T_C	T_B	T_A
0	0	1
0	1	1
0	0	1
1	1	1
0	0	1
0	1	1
0	0	1
1	1	1



Karnaugh Map for Binary Counter

T_C=AB, T_B=A, T_A=1

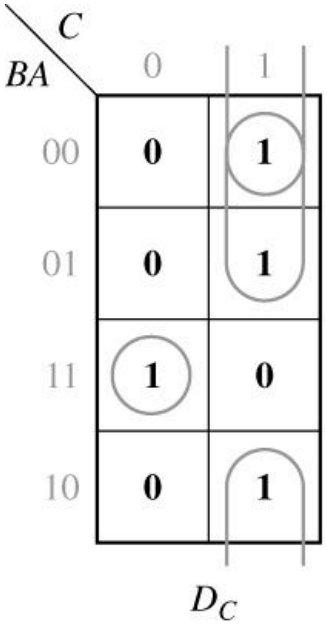
Design of Binary Counters [5 of 14]

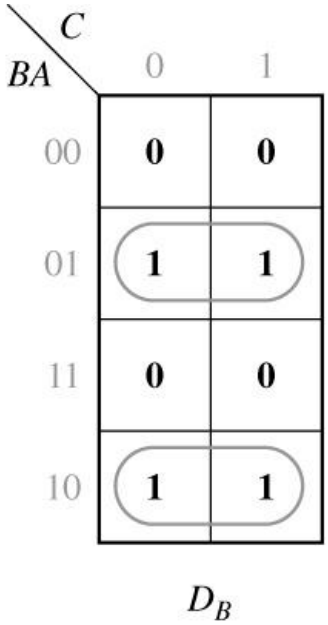
State Table for Binary Counter using D-FF

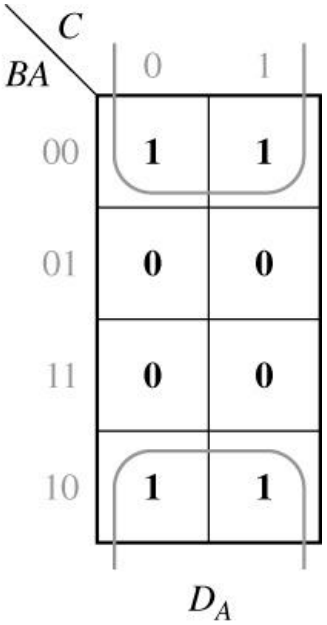
현재상태			차기상태			플립플롭 입력		
C	B	A	C+	B+	A+	D_C	D_B	D_A
0	0	0	0	0	1			
0	0	1	0	1	0			
0	1	0	0	1	1			
0	1	1	1	0	0			
1	0	0	1	0	1			
1	0	1	1	1	0			
1	1	0	1	1	1			
1	1	1	0	0	0			

DESIGN OF BINARY COUNTERS [6 OF 14]

플립플롭 입력		
D_C	D_B	D_A
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1
0	0	0







$D_A = A^+ = A'$

(A change state every clock cycle)

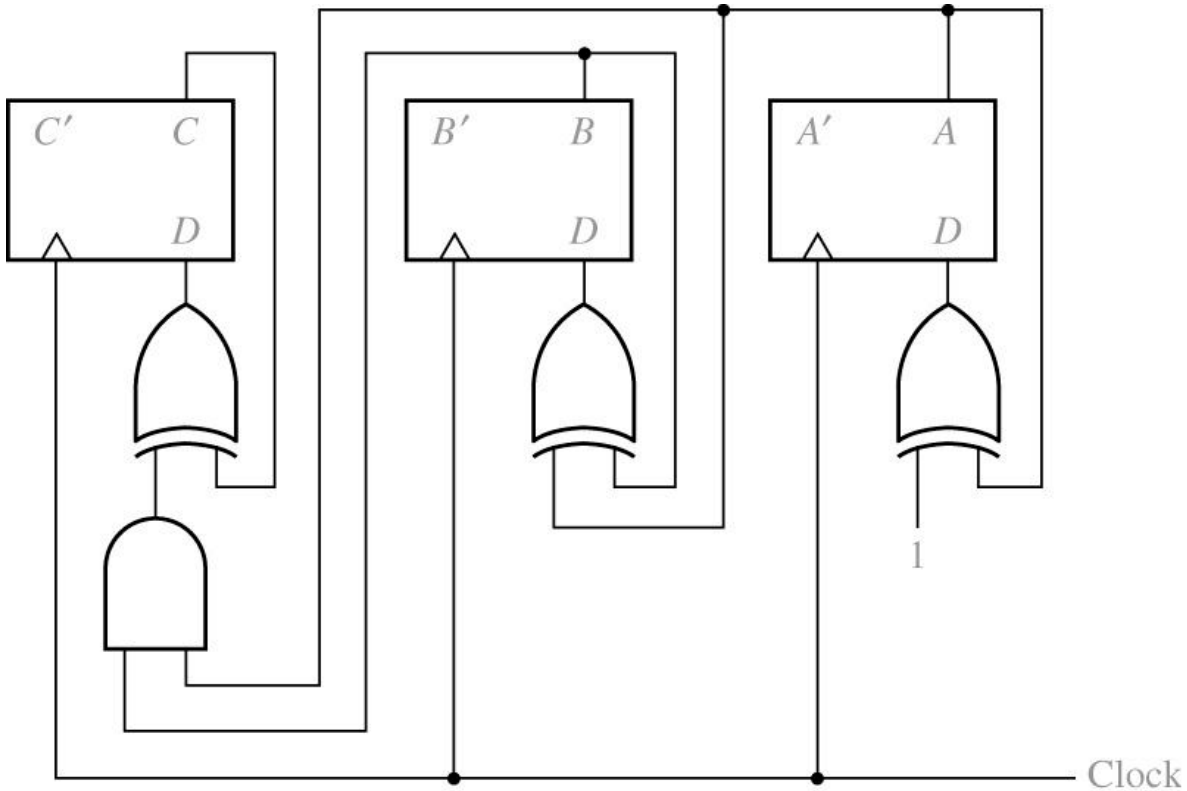
$D_B = B^+ = BA' + B'A = B \oplus A$

(B change state when $A = 1$)

$D_C = C^+ = C'BA + CB' + CA' = C'BA + C(BA)'$

$= C \oplus BA$ (C change state when $B = A = 1$)

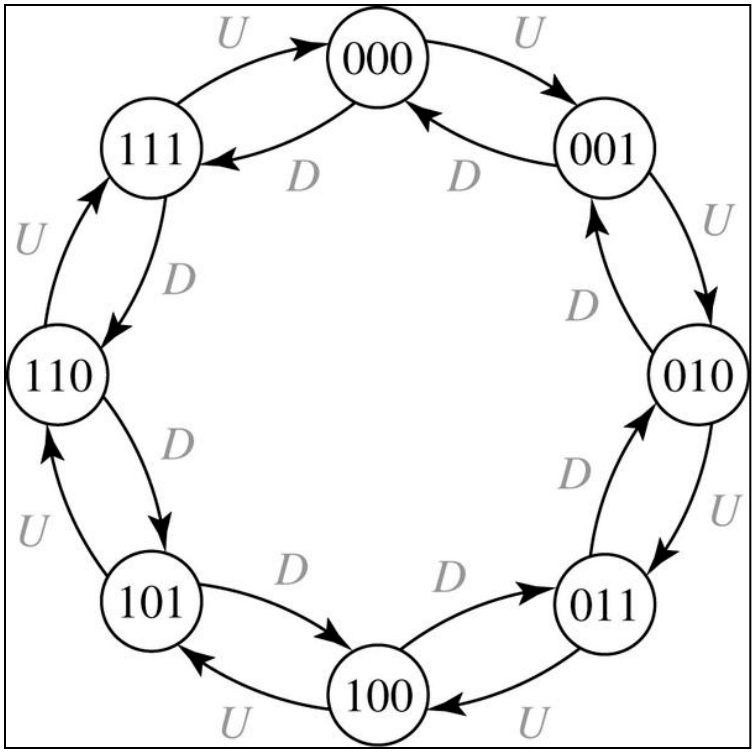
DESIGN OF BINARY COUNTERS [7 OF 14]



Binary Counter with D Flip-Flops

Design of Binary Counters [8 of 14]

- Up/Down Counter
  - 제어신호에 의해 계수 값 증가 혹은 감소



When U=1, Up counting

When D=1, Down counting

CBA	$C^+B^+A^+$	
	UP	$DOWN$
000	001	111
001	010	000
010	011	001
011	100	010
100	101	011
101	110	100
110	111	101
111	000	110

State Graph and Table for Up-Down counter

DESIGN OF BINARY COUNTERS [9 OF 14]

U=1, D=0 경우 이진 UP 카운터와 동일

현재상태			차기상태			플립플롭 입력		
C	B	A	C^+	B^+	A^+	D_C	D_B	D_A
0	0	0	0	0	1	0		
0	0	1	0	1	0	0		
0	1	0	0	1	1	0		
0	1	1	1	0	0	1		
1	0	0	1	0	1	1		
1	0	1	1	1	0	1		
1	1	0	1	1	1	1		
1	1	1	0	0	0	0		

$$D_A = A^+ = A'$$

$$D_B = B^+ = BA' + B'A = B \oplus A$$

$$D_C = C^+ = C'BA + CB' + CA' = C'BA + C(BA)' = C \oplus BA$$

DESIGN OF BINARY COUNTERS [10 OF 14]

U=0, D=1 경우 이진 down 카운터와 동일

현재상태			차기상태			플립플롭 입력		
C	B	A	C^+	B^+	A^+	D_C	D_B	D_A
0	0	0	1	1	1	1		
0	0	1	0	0	0	0		
0	1	0	0	0	1	0		
0	1	1	0	1	0	0		
1	0	0	0	1	1	0		
1	0	1	1	0	0	1		
1	1	0	1	0	1	1		
1	1	1	1	1	0	1		

$$D_A = A^+ = A \oplus 1 = A' \quad (A \text{ change state every clock cycle})$$

$$D_B = B^+ = B \oplus A' \quad (B \text{ change state when } A = 0)$$

$$D_C = C^+ = C \oplus B'A' \quad (C \text{ change state when } B = A = 0)$$

DESIGN OF BINARY COUNTERS [11 OF 14]

U=1, D=0 경우

$$D_A = A^+ = A'$$

$$D_B = B^+ = BA' + B'A = B \oplus A$$

$$D_C = C^+ = C'BA + CB' + CA' = C'BA + C(BA)' = C \oplus BA$$

U=0, D=1 경우

$$D_A = A^+ = A \oplus 1 = A' \quad (A \text{ change state every clock cycle})$$

$$D_B = B^+ = B \oplus A' \quad (B \text{ change state when } A = 0)$$

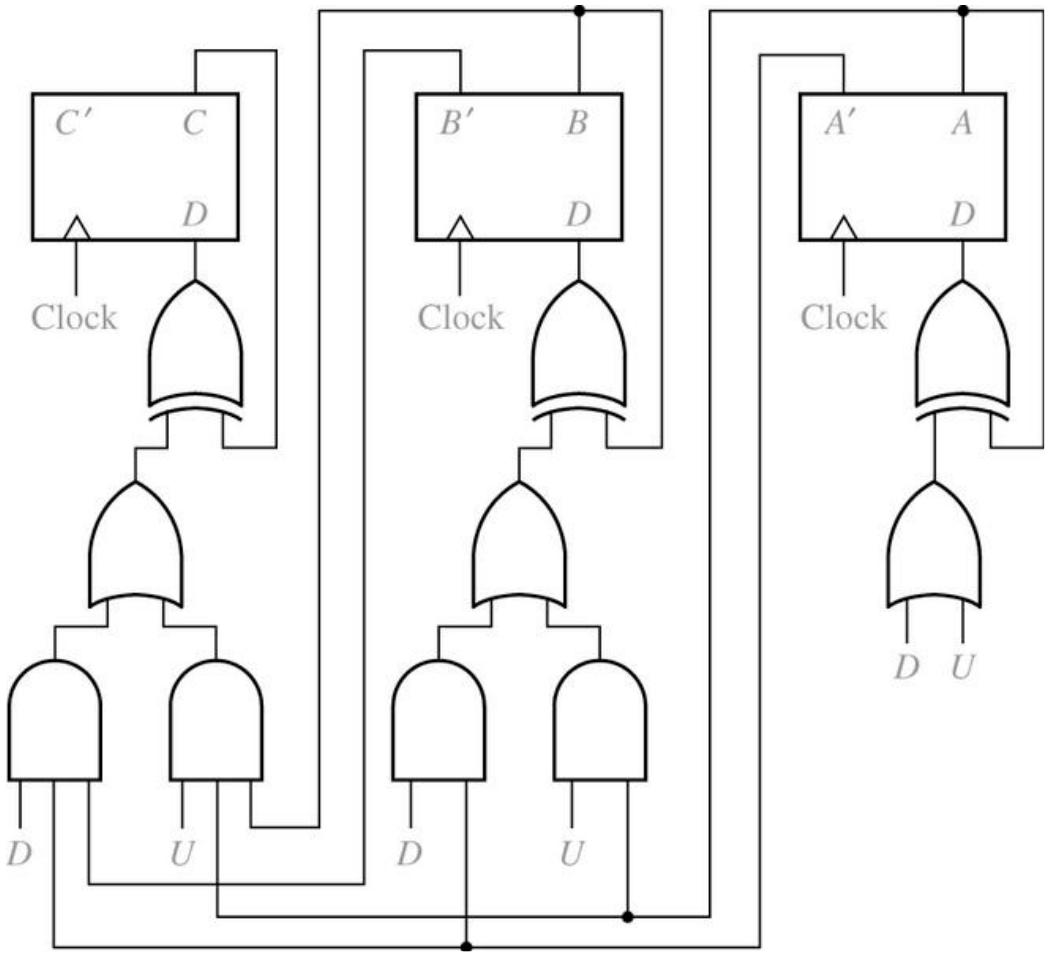
$$D_C = C^+ = C \oplus B'A' \quad (C \text{ change state when } B = A = 0)$$

$$D_A = A^+ = A \oplus (U + D)$$

$$D_B = B^+ = B \oplus (UA + DA')$$

$$D_C = C^+ = C \oplus (UBA + DB'A')$$

DESIGN OF BINARY COUNTERS [12 OF 14]

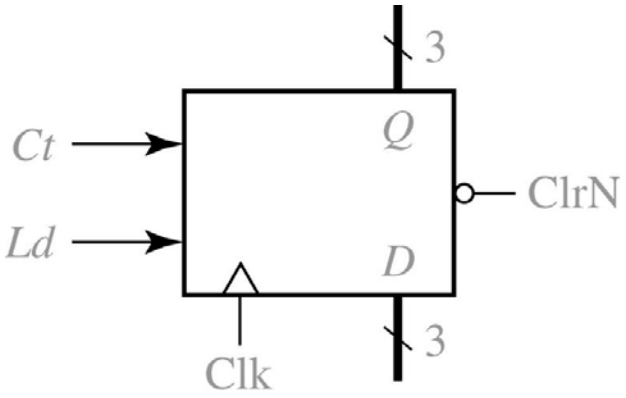


Loadable counter

Design of Binary Counters [13 of 14]

- Load 가능한 counter 설계 ~Ld 신호에 의해 초기값 load

Ld = Load, Ct = Count

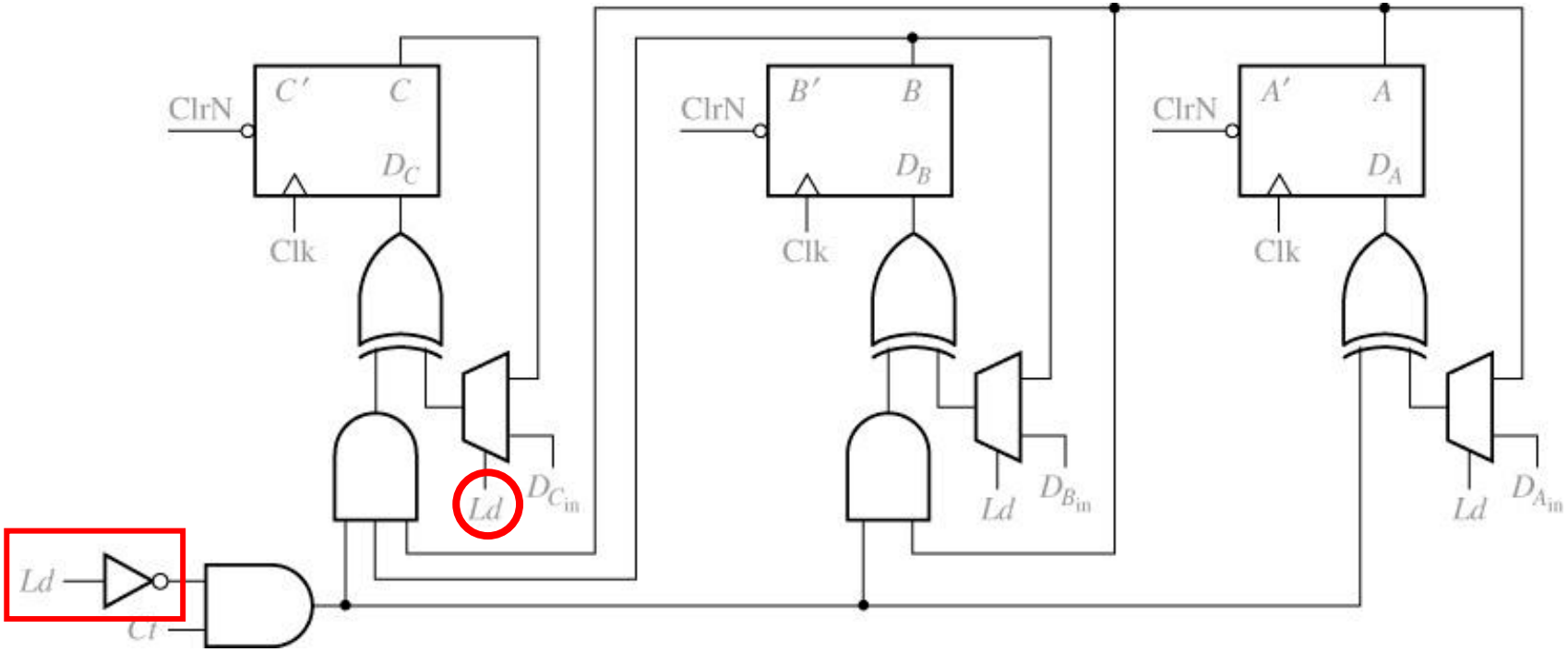


(a)

ClrN	Ld	Ct	C^+	B^+	A^+	
0	×	×	0	0	0	
1	1	×	D_C	D_B	D_A	(load)
1	0	0	C	B	A	(no change)
1	0	1	present state +1			

State table

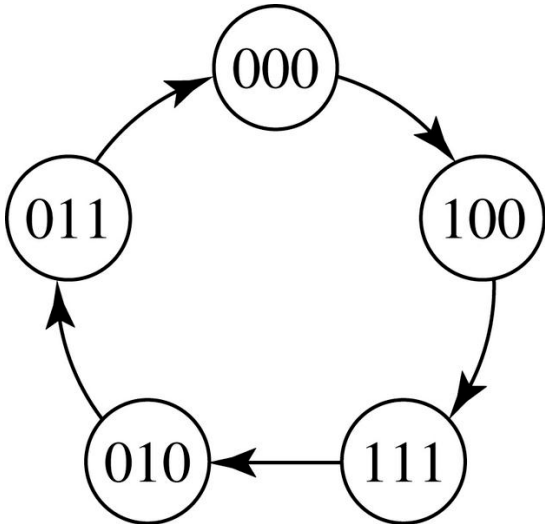
DESIGN OF BINARY COUNTERS [14 OF 14]



3bit binary counter circuit

COUNTERS FOR OTHER SEQUENCES [1 OF 6]

- The sequence of states of a counter is not in straight binary order

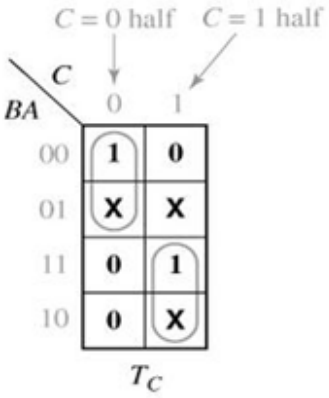


State Graph for Counter

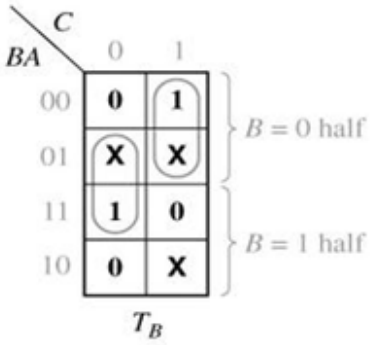
C	B	A	C^+	B^+	A^+
0	0	0	1	0	0
0	0	1	-	-	-
0	1	0	0	1	1
0	1	1	0	0	0
1	0	0	1	1	1
1	0	1	-	-	-
1	1	0	-	-	-
1	1	1	0	1	0

State Table

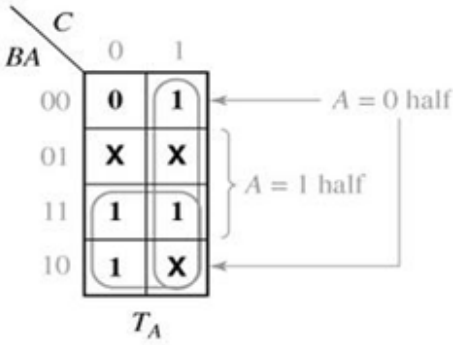
COUNTERS FOR OTHER SEQUENCES [2 OF 6]



$T_C = C'B' + CB$



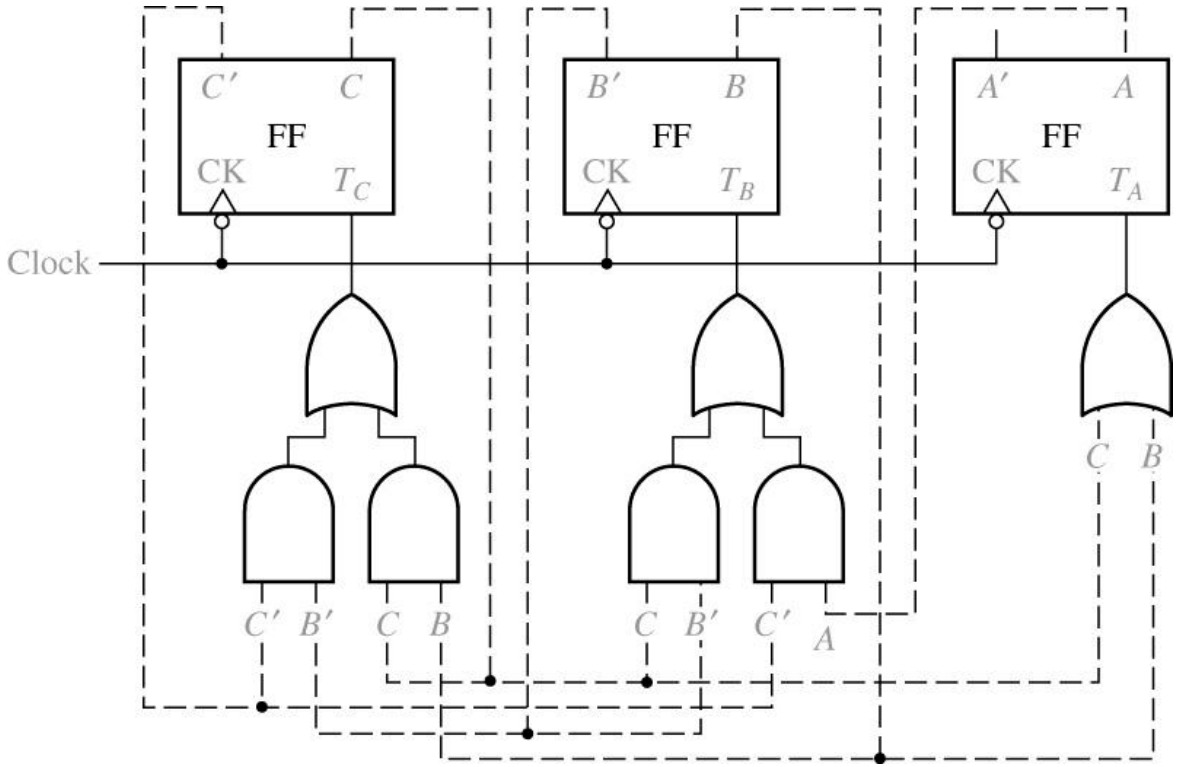
$T_B = C'A + CB'$



$T_A = C + B$

현재상태			차기상태			플립플롭 입력		
C	B	A	C+	B+	A+	T_C	T_B	T_A
0	0	0	1	0	0	1		
0	0	1	-	-	-	0		
0	1	0	0	1	1	0		
0	1	1	0	0	0	0		
1	0	0	1	1	1	0		
1	0	1	-	-	-	1		
1	1	0	-	-	-	1		
1	1	1	0	1	0	1		

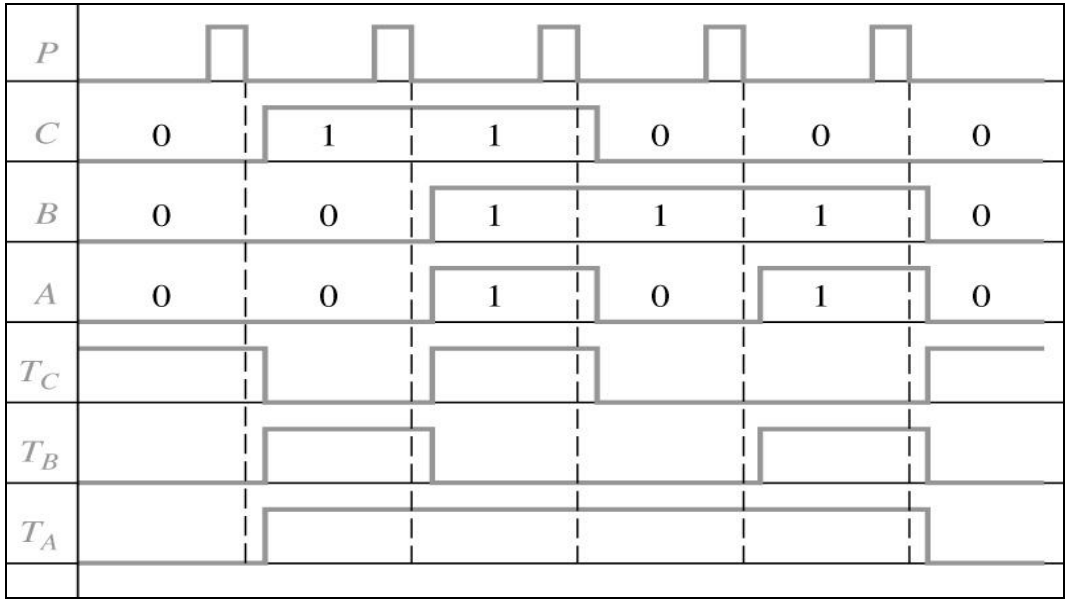
COUNTERS FOR OTHER SEQUENCES [3 OF 6]



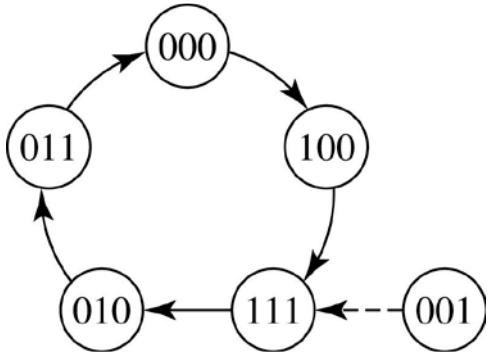
T_C=C'·B'+C·B T_B=C'·A+C·B' T_A=C+B

Counter Using T Flip-Flops

COUNTERS FOR OTHER SEQUENCES [4 OF 6]



Timing Diagram



State Graph for Counter

COUNTERS FOR OTHER SEQUENCES [5 OF 6]

- Counter Design using D Flip-Flop

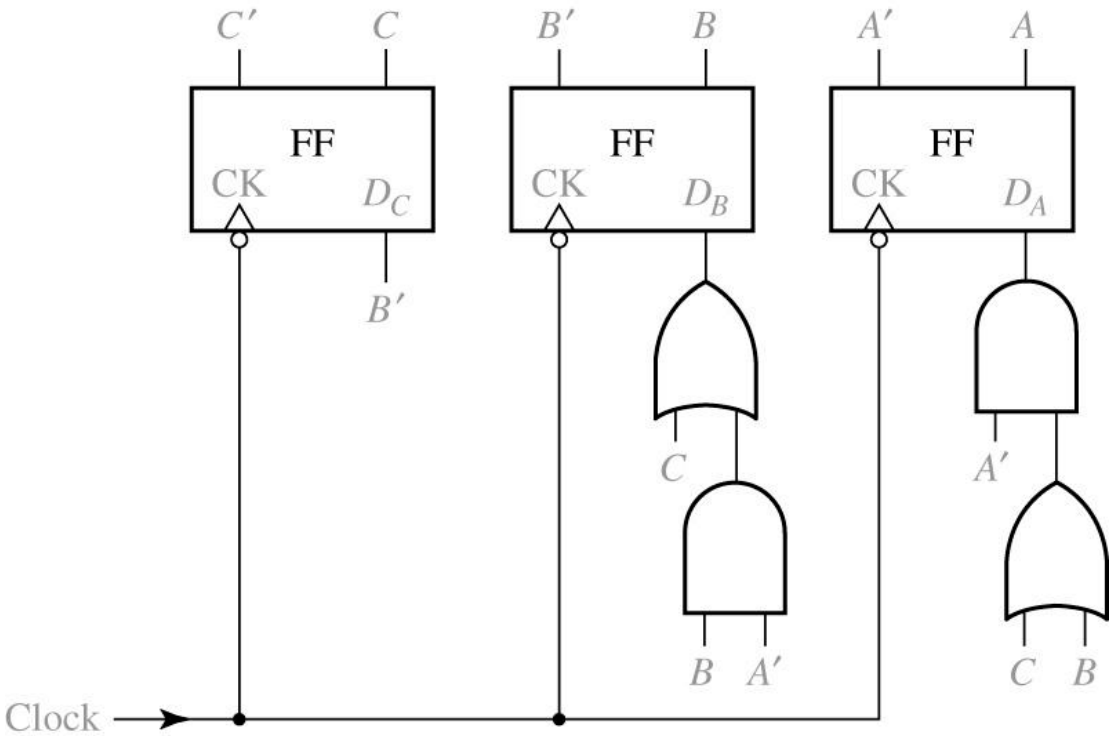
현재상태			차기상태			플립플롭 입력		
C	B	A	C+	B+	A+	D_C	D_B	D_A
0	0	0	1	0	0	1		
0	0	1	-	-	-	x		
0	1	0	0	1	1	0		
0	1	1	0	0	0	0		
1	0	0	1	1	1	1		
1	0	1	-	-	-	x		
1	1	0	-	-	-	x		
1	1	1	0	1	0	0		

D F/F 경우 Q+ = D 이므로, D 입력은 다음 상태 값과 동일함

COUNTERS FOR OTHER SEQUENCES [6 OF 6]

$$D_C = C^+ = B' \qquad D_B = B^+ = C + BA'$$

$$D_A = A^+ = CA' + BA' = A'(C + B)$$



Counter of Figure 12-21 Using D Flip-Flops

THANK YOU