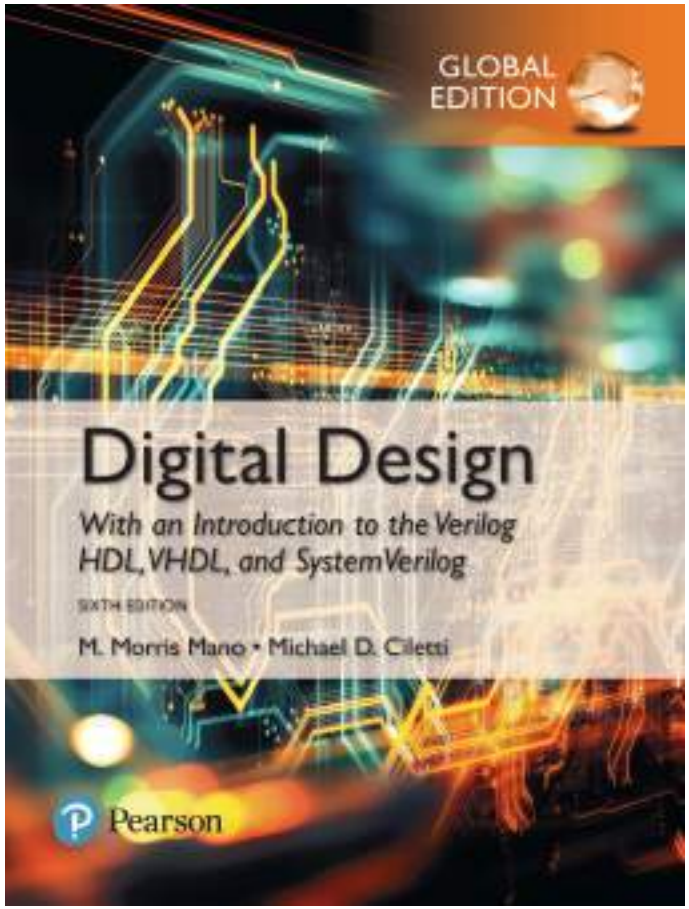
Digital Design

With an Introduction to the Verilog HDL, VHDL, and SystemVerilog

6th Edition, Global Edition



Chapter 05

Synchronous Sequential Logic

동기식 순차 논리

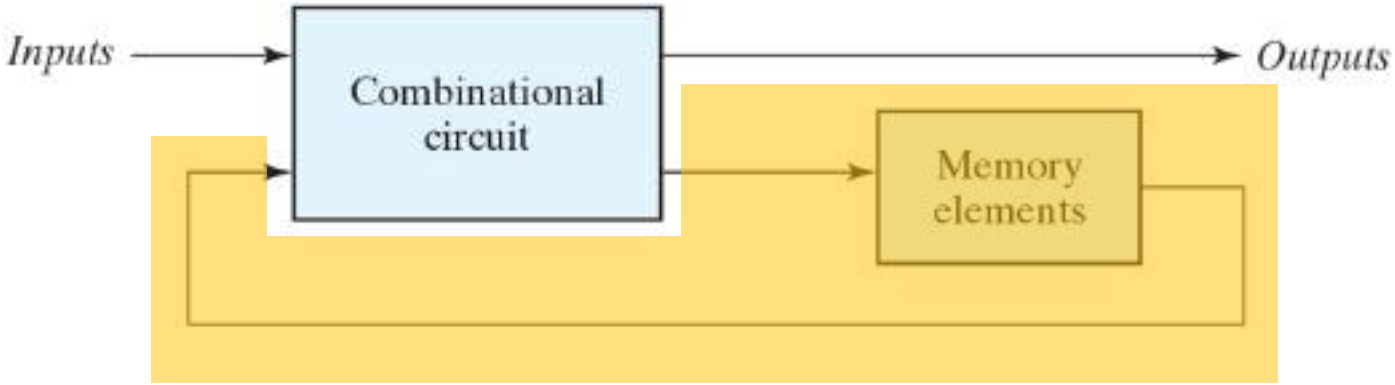
전자공학과 김동한 교수

Pearson

5.1 개요 & 5.2 순차 논리회로

- 4장의 디지털 회로는 출력이 입력에 의해서만 즉각적으로 결정되며 기억요소가 없는, 즉 과거의 입력에 영향을 받지 않는 조합회로였음
- 순차 논리회로: 피드백 경로(feedback path)를 형성하는 기억요소가 조합 논리회로에 연결. 기억요소란 2진 정보를 저장할 수 있는 장치로 순차회로의 상태(state)를 저장

Figure 5.1
Block diagram of sequential circuit.



추가됨

- 기억요소의 현재 상태와 외부 입력으로부터 출력값을 결정하고, 외부 입력은 기억요소들의 상태 변화에 대한 조건을 결정함
  - 순차 논리회로(5장): 입력, 출력, 내부 상태의 시간에 따른 시퀀스
  - 조합 논리회로(4장): 출력은 현재의 입력에 의해서만 결정
- 신호의 타이밍에 따라 동기식(synchronous)와 비동기식(asynchronous) 순차회로로 나눔
- 동기식 순차 논리회로: 이산시점에서만 기억 요소에 영향을 주는 신호를 사용함. 동기화는 clock generator에서 주기적으로 clock(clk)으로 구현함 (clock sequential circuit or synchronous circuit)

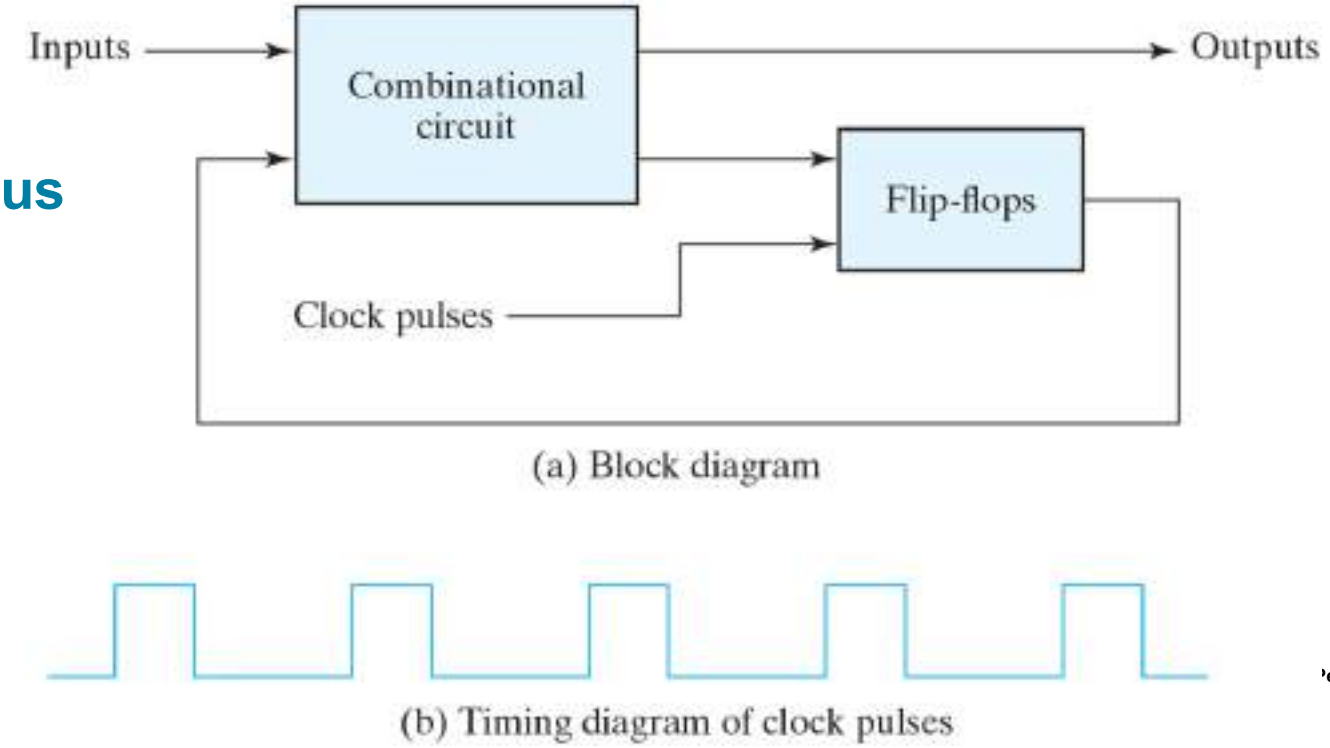


Figure 5.2 Synchronous clocked sequential circuit.

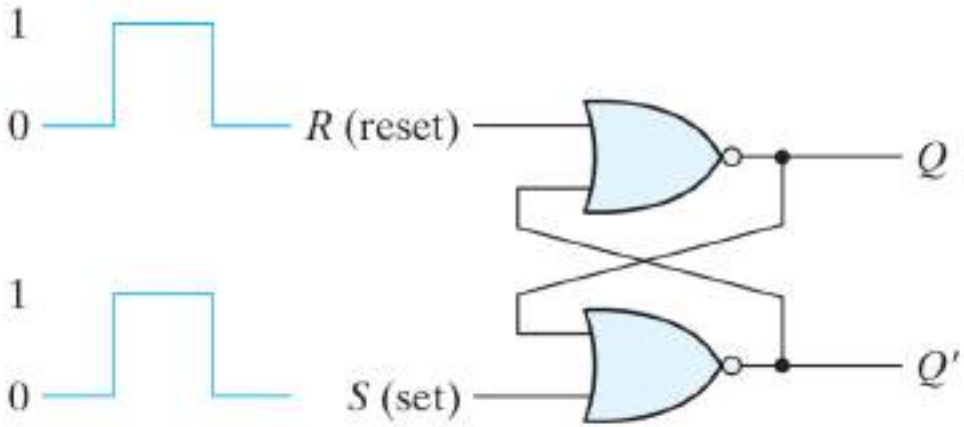
- 클럭형 순차 논리회로에 사용되는 기억요소(메모리)가 플립플롭(flip-flop)이며, 한 bit의 정보를 저장하는 2진 저장 장치임

5.3 저장 요소: 래치

- 디지털 회로의 저장 요소: 입력에 의해 상태를 바꾸도록 지시될 때까지 (회로에 전력이 공급되는 한) 2진 상태를 유지
  1. 래치: 신호 레벨에 의해 동작하는 저장 요소. 레벨 반응형
  2. 플립플롭: 클럭 천이(변화)에 의해 제어. 에지 반응형
- 래치는 비동기 순차회로를 설계하고 2진 정보를 저장하는데 유용하지만, 실질적으로 동기 순차회로에는 사용하지 않음. 그러나 래치가 플립플롭의 구성 블록이기 때문에 여기서 설명함

- SR 래치
  - 2개의 cross-coupled NOR 게이트 또는 NAND 게이트로 구성
  - 리셋용 R, 세트용 S로 표시되는 2개의 입력 단자

Figure 5.3
***SR* latch with NOR gates.**



(a) Logic diagram

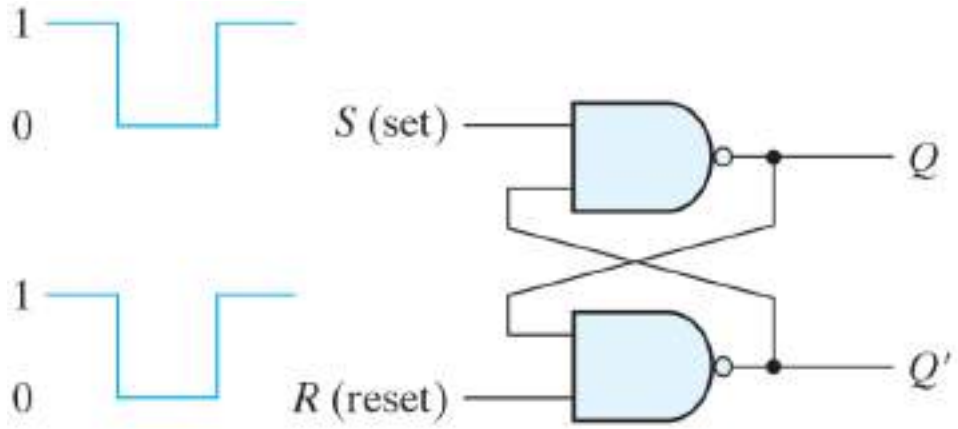
S	R	Q	Q'	
1	0	1	0	
0	0	1	0	(after $S = 1, R = 0$)
0	1	0	1	
0	0	0	1	(after $S = 0, R = 1$)
1	1	0	0	(forbidden)

(b) Function table

  - S = 1, R = 0 → 출력 Q = 1, Q' = 0, 세트 상태
  - S = 0, R = 1 → 출력 Q = 0, Q' = 1, 리셋 상태
  - S = 0, R = 0 → 상태유지
  - S = 1, R = 1 → 출력은 모두 0, 금지됨…. metastable (예측 불가능)

- SR 래치
  - NAND 게이트로 구성하면 NOR 래치의 입력의 보수임 (S'R' 래치)

Figure 5.4
***SR* latch with NAND gates.**



(a) Logic diagram

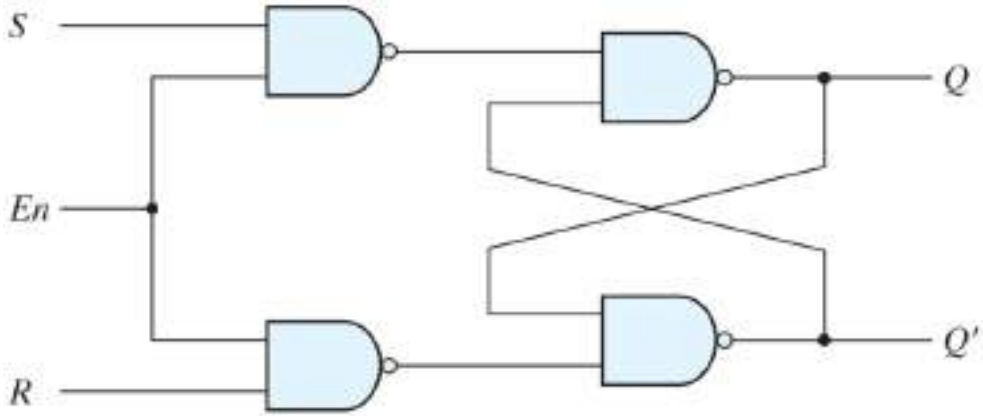
S	R	Q	Q'	
1	0	0	1	
1	1	0	1	(after $S = 1, R = 0$)
0	1	1	0	
1	1	1	0	(after $S = 0, R = 1$)
0	0	1	1	(forbidden)

(b) Function table

  - S = 1, R = 0 → 출력 Q = 0, Q' = 1, 세트 상태
  - S = 0, R = 1 → 출력 Q = 1, Q' = 0, 리셋 상태
  - S = 1, R = 1 → 상태유지
  - S = 0, R = 0 → 출력은 모두 0, 금지됨…. metastable (예측 불가능)

- 제어 입력을 가진 SR 래치
  - SR 래치에 En제어 입력을 추가함

Figure 5.5
SR latch with control input.



(a) Logic diagram

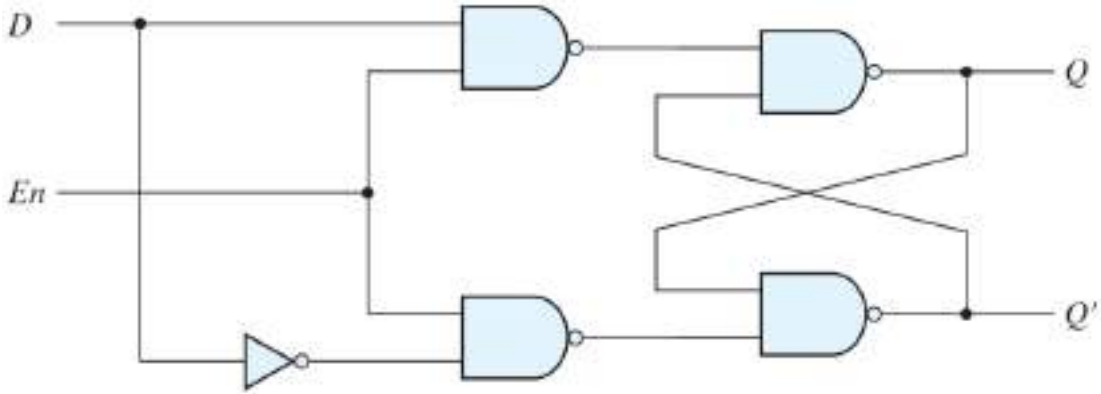
En	*S*	*R*	Next state of *Q*
0	X	X	No change
1	0	0	No change
1	0	1	$Q = 0$; reset state
1	1	0	$Q = 1$; set state
1	1	1	Indeterminate

(b) Function table

  - En = 0 → S, R에 상관없이 상태유지
  - En = 1, S = 1, R = 0 → 출력 Q = 1, Q' = 0, 세트 상태
  - En = 1, S = 0, R = 1 → 출력 Q = 0, Q' = 1, 리셋 상태
  - En = 1, S = 0, R = 0 → 상태유지
  - En = 1, S = 1, R = 1 → 정의 불가

- D 래치 (투과형 래치)
  - SR 래치에서 정의되지 않는 상태를 피하려면 S와 R 입력이 동시에 1이 되지 않도록 해야 함
  - D 래치는 두 개의 입력 D(date), En만을 가짐

Figure 5.6
***D* latch.**



(a) Logic diagram

En	*D*	Next state of *Q*
0	X	No change
1	0	$Q = 0$; reset state
1	1	$Q = 1$; set state

(b) Function table

  - En = 0 → 상태유지
  - En = 1, D = 0 → 출력 Q = 0, 리셋 상태
  - En = 1, D = 1 → 출력 Q = 1, 세트 상태
  - En이 1이면 입력 D가 그대로 출력 Q가 됨 (투과됨). En이 0이면 출력 Q가 유지됨 (저장됨)

Figure 5.7
Graphic symbols for latches.

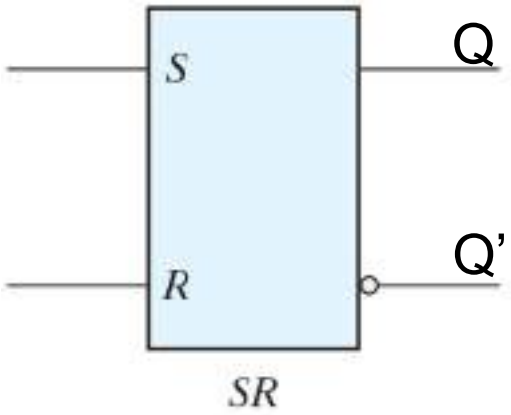


Figure 5.3
SR latch with NOR gates

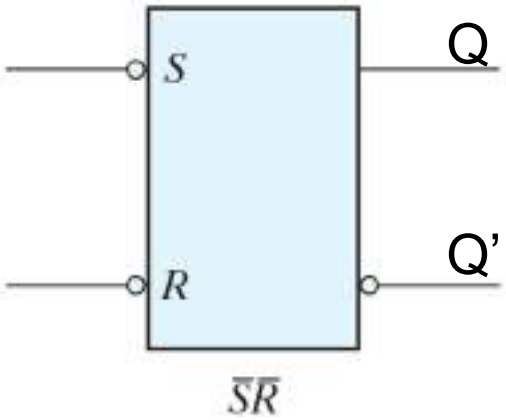


Figure 5.4
SR latch with NAND gates.

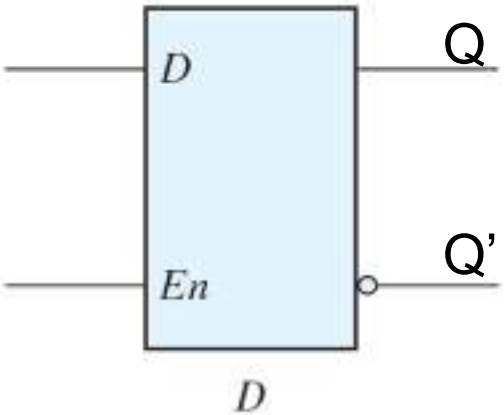


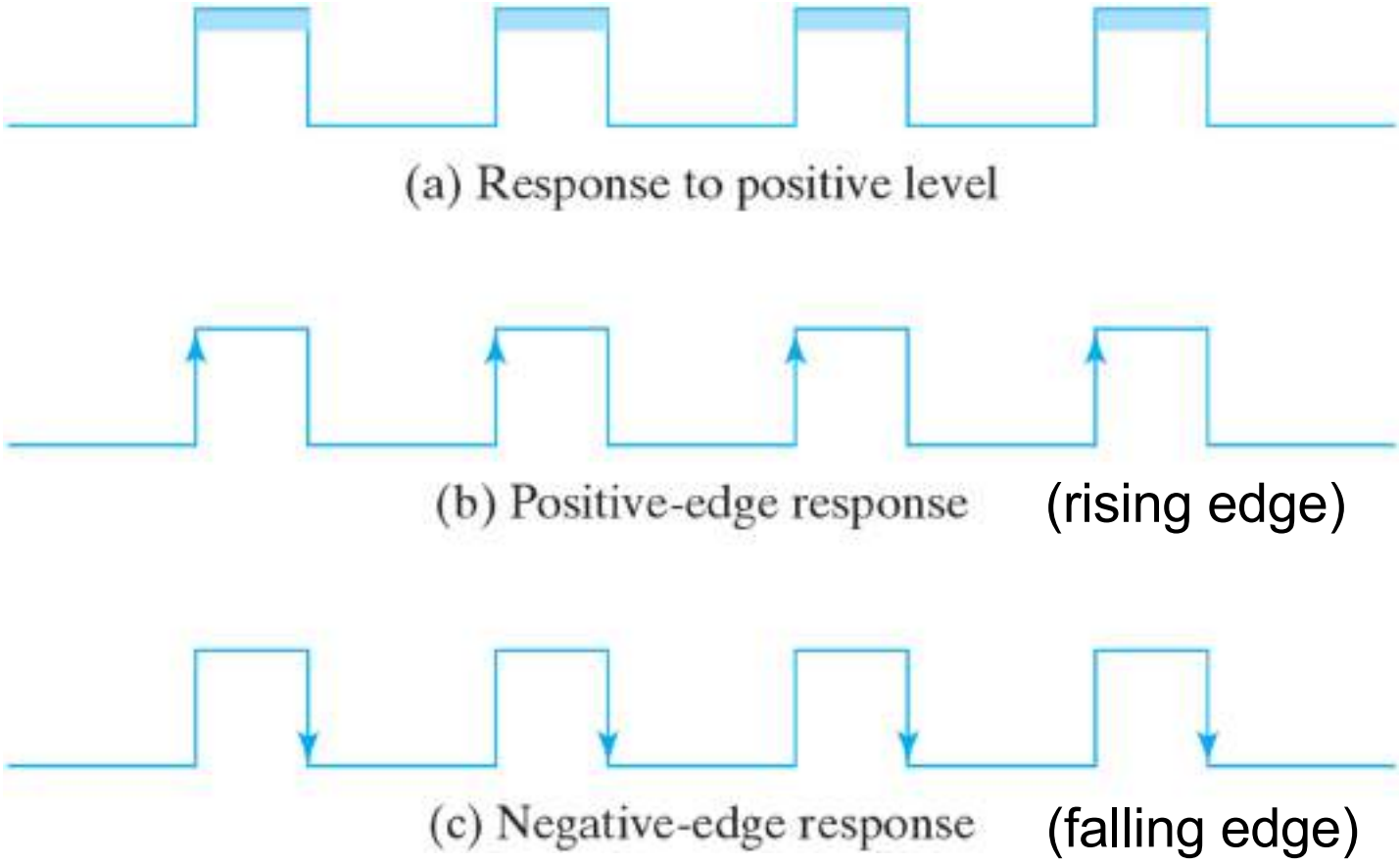
Figure 5.6
D latch.

5.4 저장 요소: 플립플롭

- 래치 또는 플립플롭의 제어 입력이 변경되면 상태가 전환되며, 이것을 트리거(trigger)라고 함.
- D 래치는 En가 논리 1 레벨(보통 5V)로 갈 때마다 트리거되는 플립플롭이며 En가 논리 1레벨을 유지하는 한, 데이터 입력(D)가 변경되면 출력(Q)의 상태가 변경됨 → 저장 요소로 사용될 때 문제됨
- 그림 5.8의 정레벨(positive level) 응답을 에지 응답으로 바꿔서 순차 회로에 사용하도록 래치를 수정함

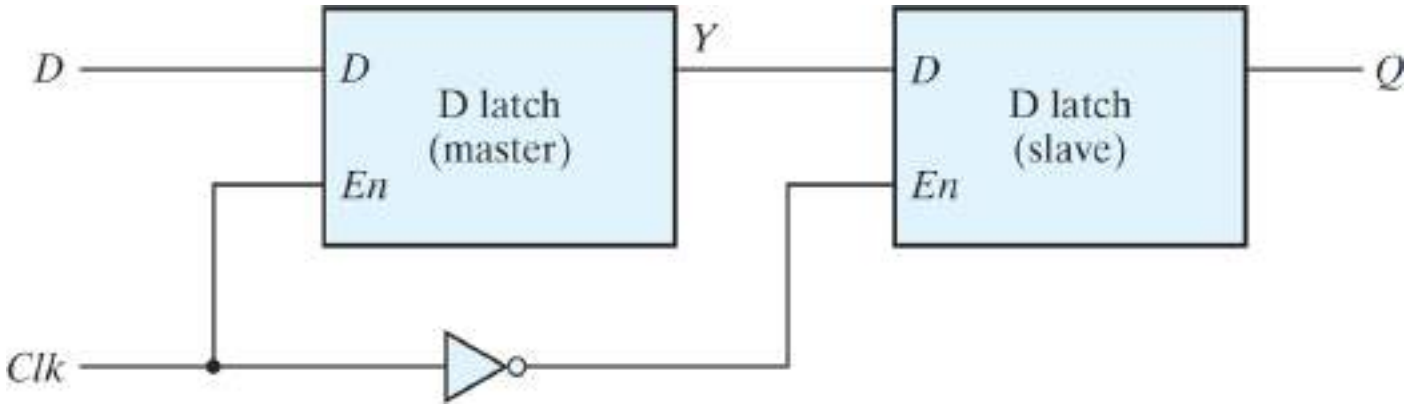
1. 플립플롭의 출력을 격리하고 입력이 변하는 동안 출력이 영향을 받지 않도록 2개의 래치를 사용하는 방법
2. 동기화 신호(클럭)의 신호 천이(0→1 or 1→0) 동안에만 트리거되도록

Figure 5.8
Clock response in latch and flip-flop.



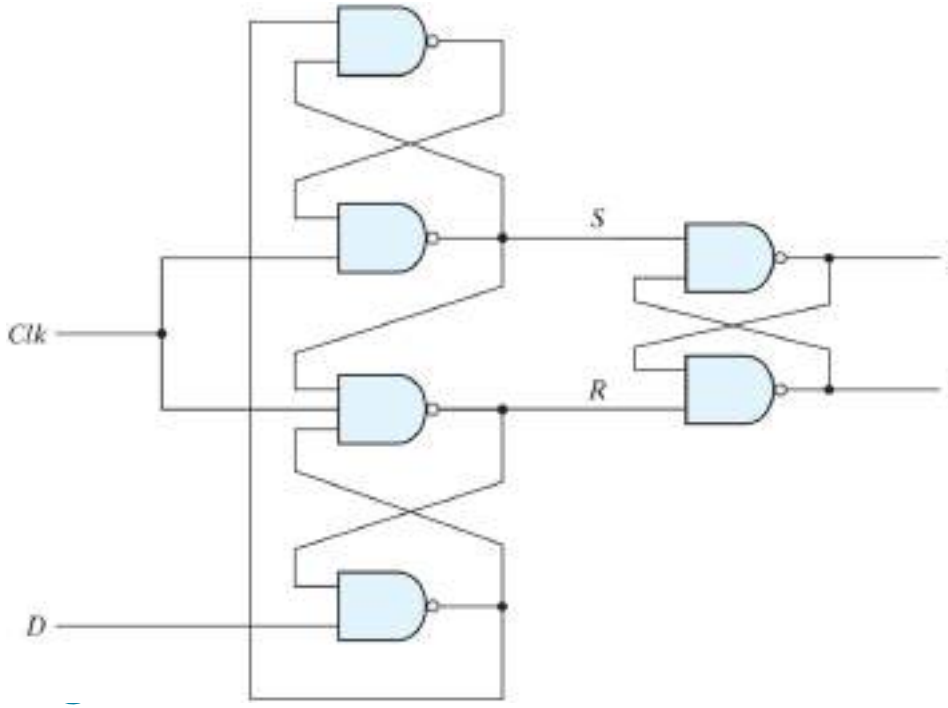
- 에지 트리거 D 플립플롭
  - 방법 1: 마스터-슬레이브 D 플립플롭
    - 2개의 D 래치와 1개의 인버터로 구성
    - clk의 하향 에지에서 Y의 값이 Q로 전달
      - CLK = 0이면 master는 disable, slave 는 enable (Q=Y)
      - CLK = 1이면 master는 enable, slave 는 disable (D=Y)
      - 따라서 출력은 클럭이 1에서 0으로 천이될 때만 변함
    - 상향 에지에서 동작시키려면 clk 앞에 인버터 추가하면 됨

Figure 5.9
Master–slave *D* flip-flop.



- 에지 트리거 D 플립플롭
  - 방법 2: D형 상향 에지 트리거 플립플롭
    - 3개의 SR 래치로 구성
      - CLK = 0이면, S와 R은 레벨 1을 유지하고, 출력 유지
      - CLK가 1로 변할 때, D = 0이면, R = 0, Q = 0
      - CLK = 1이면, 입력 D가 변해도 R = 0 유지

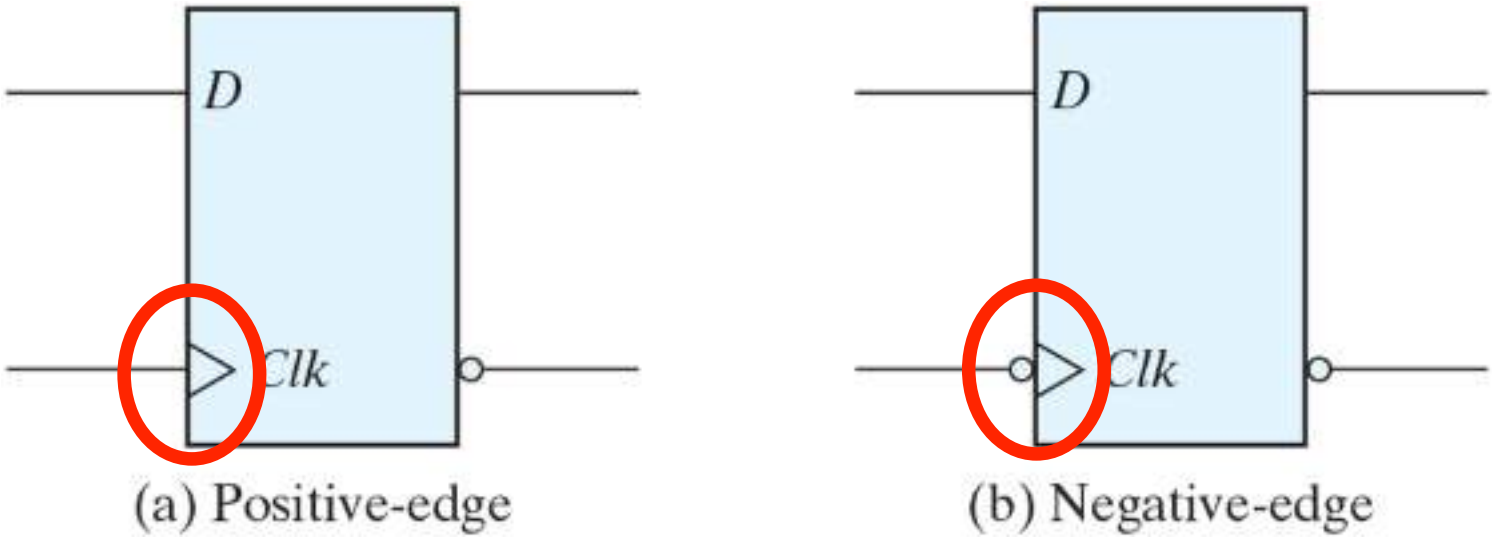
Figure 5.10
***D*-type positive-edge-triggered flip-flop.**



Clk	D	S	R	Q(t+1)	
0	X	1	1	Q	
↓ 1	0	1	0	0	(reset)
1	1	1	0	0	
0	X	1	1	Q	
↓ 1	1	0	1	1	(set)
1	0	0	1	1	

- setup time: 클럭 천이가 발생하기 전에 입력 D가 일정한 값을 유지해야 하는 최소 시간 (데이터 북 참조)
- hold time: D 입력에 클럭의 상향 천이가 일어난 후 변하지 않아야 하는 최소 시간 (데이터 북 참조)

Figure 5.11
Graphic symbol for edge-triggered *D* flip-flop.

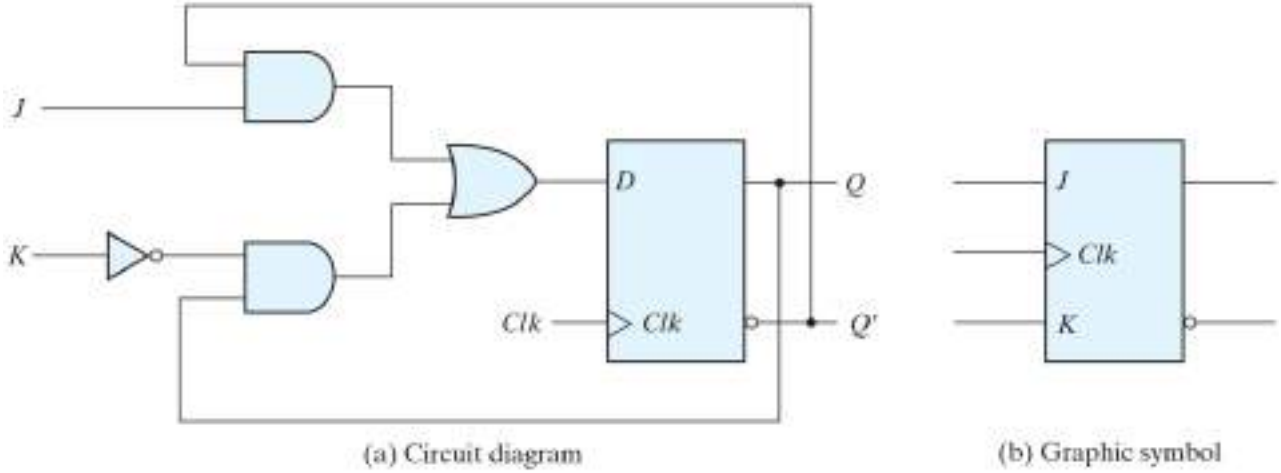


- 동적 지시자(dynamic indicator, >)

- 기타 플립플롭들
  - 가장 적은 수의 게이트로 만든 가장 경제적이고 효율적인 플립플롭은 에지 트리거 D 플립플롭
  - 따라서 다른 유형의 플립플롭들은 D 플립플롭과 외부 논리 게이트를 사용하여 만듬
  - 디지털 시스템 설계에서 폭넓게 사용되는 플립플롭은 JK 플립플롭과 T 플립플롭임

- JK 플립플롭
  - D 플립플롭과 게이트로 구성
  - 입력 J는 상태를 1로 세팅, 입력 K는 상태를 0으로 리셋
  - 입력 J, K 모두 1로 될 때는 출력을 보수로 만듬

Figure 5.12
JK flip-flop.

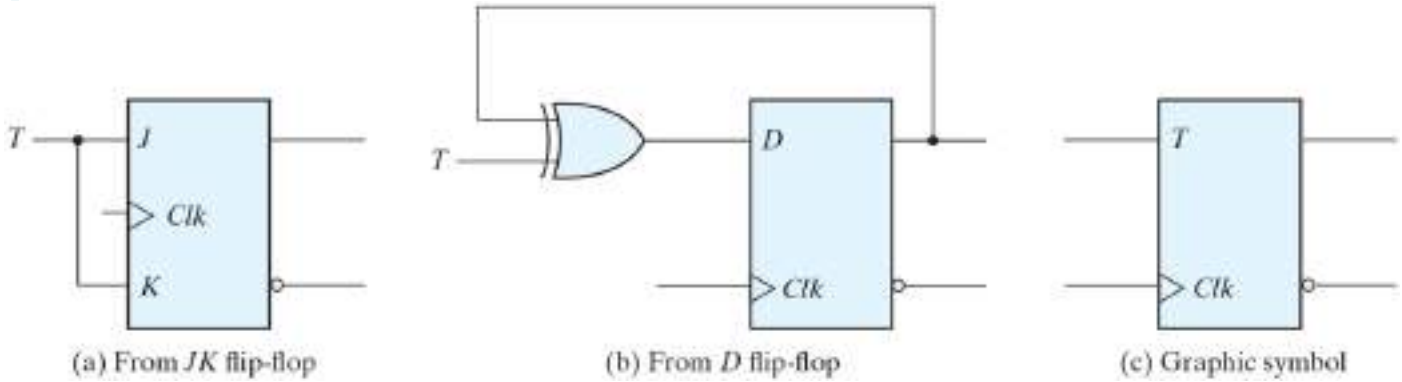


(a) Circuit diagram (b) Graphic symbol

$$D = JQ' + K'Q$$

- J = 1이고 K = 0일 때, D = Q' + Q = 1임. 그 다음 클럭 에지에서 출력이 1로 변함
- J = 0이고 K = 1일 때, D = 0임. 그 다음 클럭 에지에서 출력이 0으로 변함
- J = K = 1일 때는 D = Q'임. 그 다음 클럭 에지에서 출력이 보수로 변함
- J = K = 0일 때는 D = Q임. 그 다음 클럭 에지에서 출력이 변하지 않음

Figure 5.13
T flip-flop.



(a) From *JK* flip-flop (b) From *D* flip-flop (c) Graphic symbol

- T 플립플롭
  - 보수로 만드는 플립플롭. JK 플립플롭의 입력을 묶음
  - T = 0 (즉 J = K = 0) 일 때는 출력이 변하지 않음
  - T = 1 (즉 J = K = 1) 일 때는 출력이 보수화됨
  - 그림 (b)처럼 D 플립플롭과 XOR 게이트로 설계 가능
    - D = T⊕Q = TQ' + T'Q

- 특성표 (characteristic table)
  - Q(t)는 클럭 펄스가 아직 가해지지 않은 상태, Q(t+1)은 한 클럭이 경과한 후의 상태 혹은 다음 상태.

Table 5.1
Flip-Flop Characteristic Tables.

***JK* Flip-Flop**

J	*K*	*Q*(*t* + 1)	
0	0	$Q(t)$	No change
0	1	0	Reset
1	0	1	Set
1	1	$Q'(t)$	Complement

***D* Flip-Flop**

D	*Q*(*t* + 1)	
0	0	Reset
1	1	Set

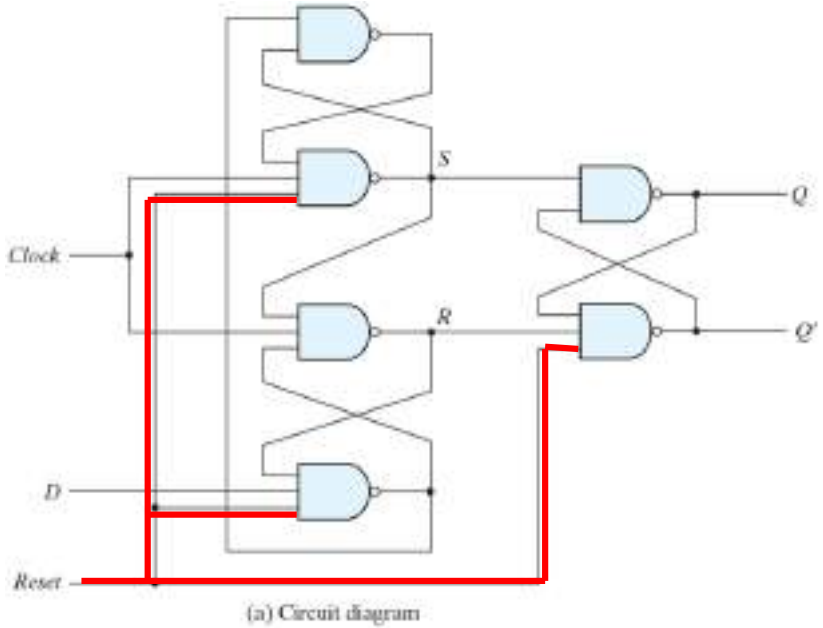
***T* Flip-Flop**

T	*Q*(*t* + 1)	
0	$Q(t)$	No change
1	$Q'(t)$	Complement

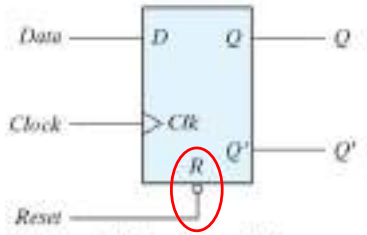
- 특성식 (characteristic equation)
  - 플립플롭의 논리적 특성을 대수적으로 표현한 것
  - D 플립플롭
    - $Q(t+1) = D$
  - JK 플립플롭
    - $Q(t+1) = JQ' + K'Q$
  - T 플립플롭
    - $Q(t+1) = T \oplus Q = TQ' + T'Q$

- 직접 입력
  - 클럭과 무관하게 특정한 상태로 변화시키는 비동기 입력
  - 클럭이 작동하기 전에 플립플롭을 시작 상태로 만들 때 유용함
  - 그림 5.14는 비동기 리셋을 가진 상향 에지 트리거 D 플립플롭임
    - 세 개의 NAND 게이트에 리셋이 추가됨
  - 1로 세팅하는 프리셋(preset) 또는 직접 세트(direct set)
  - 0으로 리셋하는 클리어(clear) 또는 직접 리셋(direct reset)

Figure 5.14
D flip-flop with asynchronous reset.



(a) Circuit diagram



(b) Graphic symbol

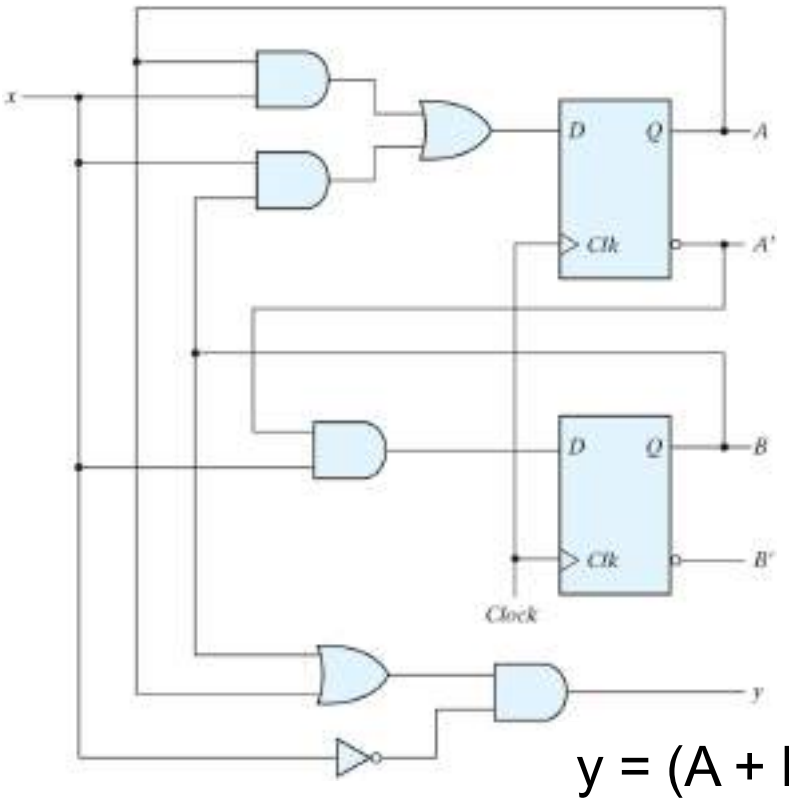
R	Clk	D	Q	Q'
0	X	X	0	1
1	↑	0	0	1
1	↑	1	1	0

(c) Function table

5.5 클럭형 순차 회로 분석

- 출력과 다음 상태는 입력과 현재 상태에 의한 함수임
- 입력, 출력, 내부 상태의 시간 순서를 보여 주는 표나 다이어그램을 구함
- 상태식 (state equation 혹은 transition equation): 현재 상태와 입력들로 다음 상태를 명시함

Figure 5.15
Example of sequential circuit.



- A(t+1) = A(t)x(t) + B(t)x(t)
- B(t+1) = A'(t)x(t)
  - (t)는 생략 가능
- 좌변은 한 클럭 에지 후의 플립플롭의 다음 상태
- 우변은 다음 상태를 1로 만드는 현재 상태와 입력조건의 부울 표현

y = (A + B)x'

- 상태표(state table or transition table)
  - 현재 상태, 입력, 다음 상태, 출력 표기

Table 5.2 / 5.3
State Tables for the Circuit of Fig. 5.15.

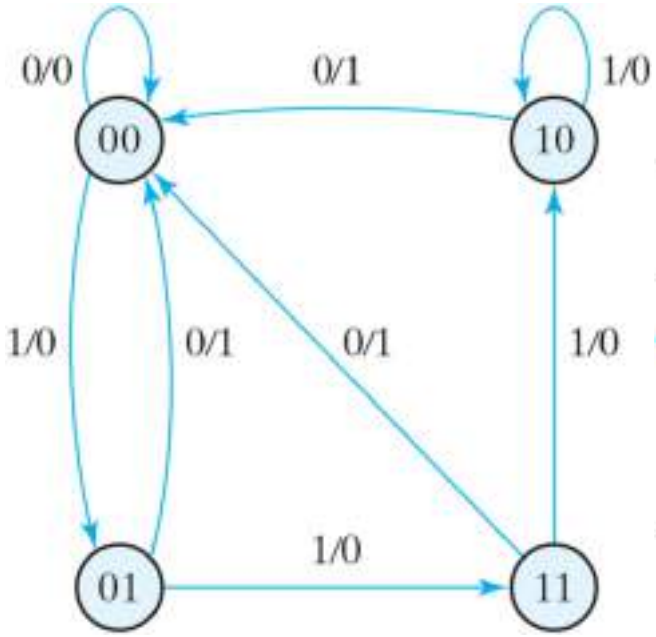
Present State		Input	Next State		Output
A	B	x	A	B	y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	1	0	0
1	1	0	0	0	1
1	1	1	1	0	0

Present State		Next State				Output	
		x = 0		x = 1		x = 0	x = 1
A	B	A	B	A	B	y	y
0	0	0	0	0	1	0	0
0	1	0	0	1	1	1	0
1	0	0	0	1	0	1	0
1	1	0	0	1	0	1	0

  - 현재 상태와 입력의 모든 조합을 나열한 후, 논리도와 상태식으로 다음 상태 결정
  - 그림 5.15과 위의 상태표에서 A(t+1) = Ax + Bx, B(t+1) = A'x, y = Ax' + Bx'
  - m개의 플립플롭과 n개의 입력을 갖는 순차회로는 상태표에서 2m+n개의 열을 갖음. 다음 상태는 플립플롭 갯수인 m개의 열을 갖음

- 상태도(state diagram)
  - 상태표를 시각화
  - 원 안의 2진수를 플립플롭의 상태, 화살표 근처의 사선 앞 숫자는 현재 상태의 입력, 사선 뒤 숫자는 입력에 의한 현재 상태에서의 출력값

Figure 5.16
State diagram of the circuit of Fig. 5.15.



Present State		Next State				Output	
		$x = 0$		$x = 1$		$x = 0$	$x = 1$
A	B	A	B	A	B	y	y
0	0	0	0	0	1	0	0
0	1	0	0	1	1	1	0
1	0	0	0	1	0	1	0
1	1	0	0	1	0	1	0

  - 회로도→방정식→상태표→상태도

- 플립플롭의 입력식
  - 플립플롭의 입력을 만드는 회로를 부울 함수로 표현
  - 그림 5.15의 순차 회로는 2개의 D플립플롭 A와 B, 그리고 입력 x와 출력 y로 구성되고 플립플롭의 입력식과 출력식을 대수학적으로 표현하면

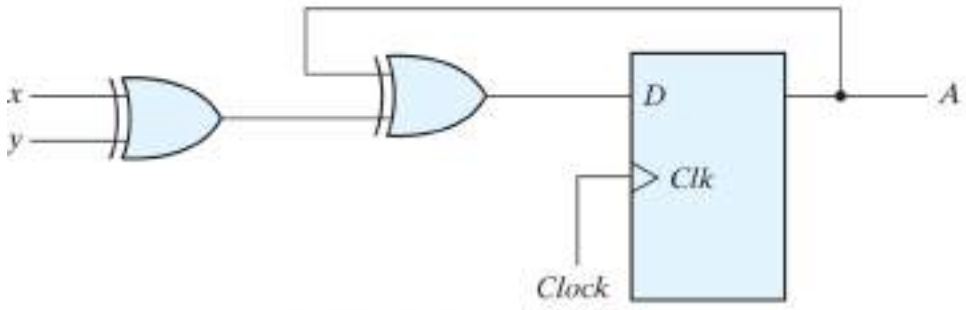
$$D_A = Ax + Bx$$

$$D_B = A'x$$

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

  - 이 예제에서는 입력식과 상태식이 일치하면 이는 D 플립플롭의 다음 상태가 입력과 같기 때문임
- D 플립플롭을 가진 회로의 분석
  - 예제에서 입력식은 $D_A = A \oplus B \oplus y$임. 출력식이 따로 주어지지 않았기 때문에 회로의 출력은 플립플롭의 출력임
  - 상태표는 플립플롭 A의 현재 상태를 나열한 열인 2개의 입력에 대한 2개의 열과 A의 다음 상태를 나열한 열로 이루어짐
  - 논리도는 그림 5.17(a)이고, 입력식에서 구해짐. 상태표를 적고 그림 5.17(c)처럼 상태도 작성

Figure 5.17
Sequential circuit with *D* flip-flop.

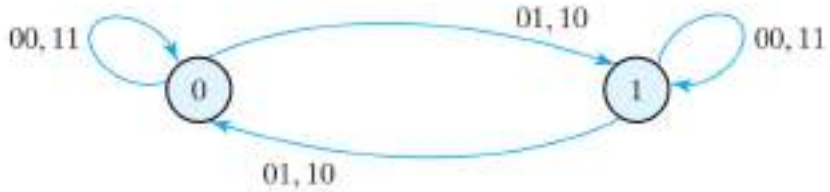


(a) Circuit diagram

Present state	Inputs		Next state
A	*x*	*y*	*A*
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

A(t+1)

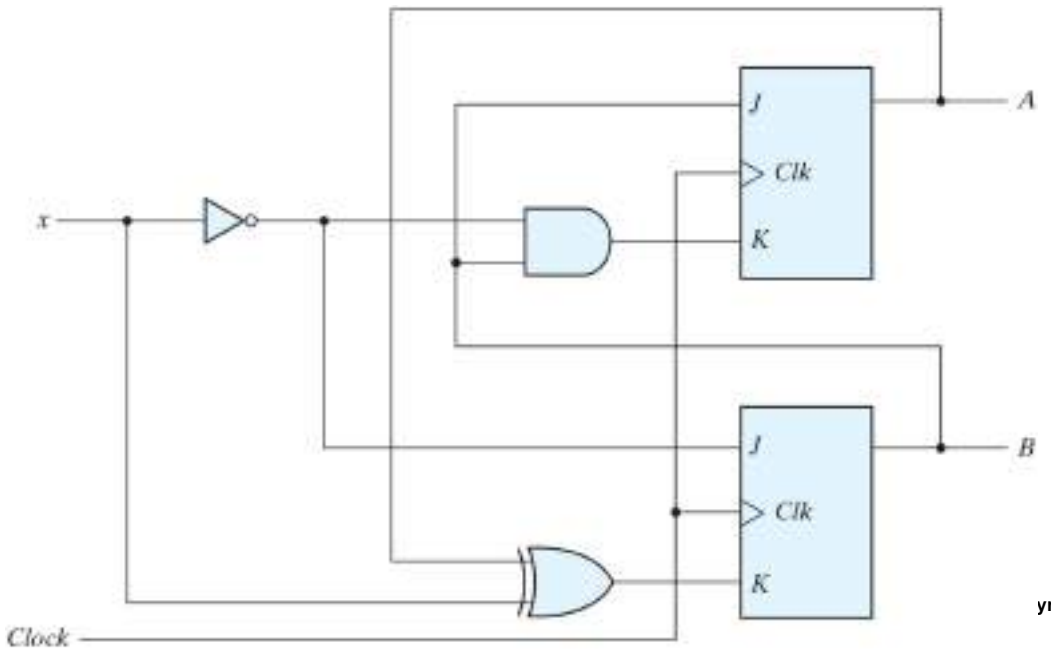
(b) State table



(c) State diagram

- JK 플립플롭을 가진 회로의 분석
  - D 플립플롭은 상태식과 다음 상태 값이 일치해서 쉽지만 JK나 T 플립플롭은 다음 상태를 얻기 위해서 플립플롭에 따른 특성표와 특성식을 참고해야 함
    1. 현재 상태와 입력들을 이용하여 플립플롭의 입력식을 결정
    2. 입력식에 대한 2진 값들을 나열
    3. 상태표의 다음 상태 값을 결정하기 위해 플립플롭의 특성표 이용
  - 그림 5.18을 예들 들면,

Figure 5.18
Sequential circuit with *JK* flip-flop.



- 플립플롭의 입력식은
  - $J_A = B, K_A = Bx'$
  - $J_B = x', K_B = A'x + Ax' = A \oplus x$
- 따라서 상태표는 작성 1처럼 현재상태, 입력의 조합 나열, 작성 2처럼 위의 입력식을 기반으로 작성, 이후 JK 플립플롭 특성표(표 5.1) 를 이용해서 작성3을 완성

Table 5.4
State Table for Sequential Circuit with JK Flip-Flops.

Present State		Input	Next State		Flip-Flop Inputs			
A	B	x	A	B	J_A	K_A	J_B	K_B
0	0	0	0	1	0	0	1	0
0	0	1	0	0	0	0	0	1
0	1	0	1	1	1	1	1	0
0	1	1	1	0	1	0	0	1
1	0	0	1	1	0	0	1	1
1	0	1	1	0	0	0	0	0
1	1	0	0	0	1	1	1	1
1	1	1	1	1	1	0	0	0

작성 1 작성 3 작성 2

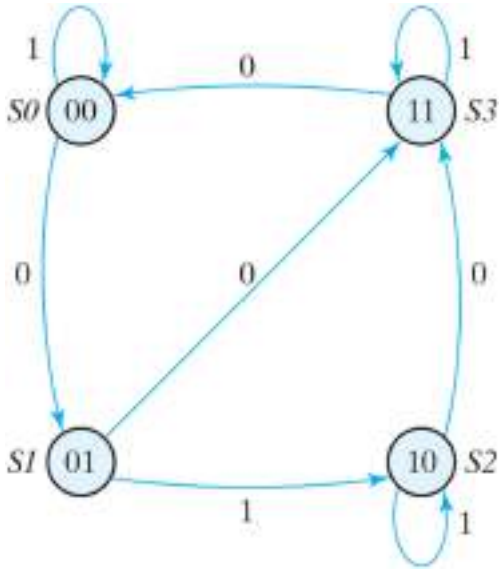
– 상태값은 특성식으로부터 상태식을 계산하여 얻어지고, 다음의 과정을 따름

1. 현재 상태와 입력 변수를 이용해서 입력식 결정
2. 입력식을 플립플롭 특성식으로 바꿈
3. 상태표에서 얻은 다음 상태를 결정하기 위해 상태식을 이용

– A(t+1) = JA' + K'A (일반적인 JK 플립플롭의 입력식)
B(t+1) = JB' + K'B (일반적인 JK 플립플롭의 입력식)

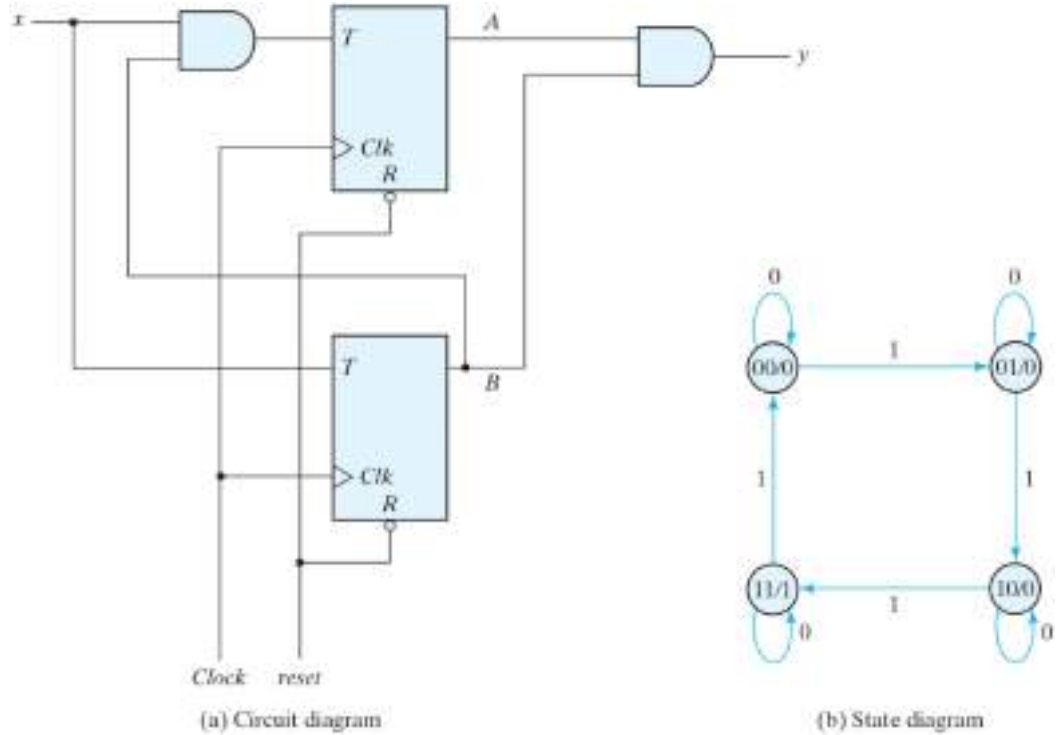
– A(t+1) = BA' + (Bx')'A = BA' + AB' + Ax
B(t+1) = x'B' + (A⊕x)'B = B'x' + ABx + A'Bx'

Figure 5.19
State diagram of the circuit of Fig. 5.18.



- T 플립플롭을 가진 회로의 분석
  - JK 플립플롭 회로 분석과 동일함. 단, 표 5.1의 플립플롭 특성표가 다름. $Q(t+1) = T \oplus Q = T'Q + TQ'$
  - 입력식 $T_A = Bx$, $T_B = x$, 출력식 $y = AB$

Figure 5.20
Sequential circuit with T flip-flops (Binary Counter).



(a) Circuit diagram (b) State diagram

– 상태표는 아래와 같음.

Table 5.5
State Table for Sequential Circuit with T Flip-Flops.

Present State		Input	Next State		Output
A	*B*	*x*	*A*	*B*	*y*
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	1	0
0	1	1	1	0	0
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	1

– 다음 상태 값은 특성식에서 T_A와 T_B를 대체하여 얻어진 상태식에서 유도됨

$$A(t+1) = (Bx)'A + (Bx)A' = AB' + Ax' + A'Bx$$

$$B(t+1) = x \oplus B$$

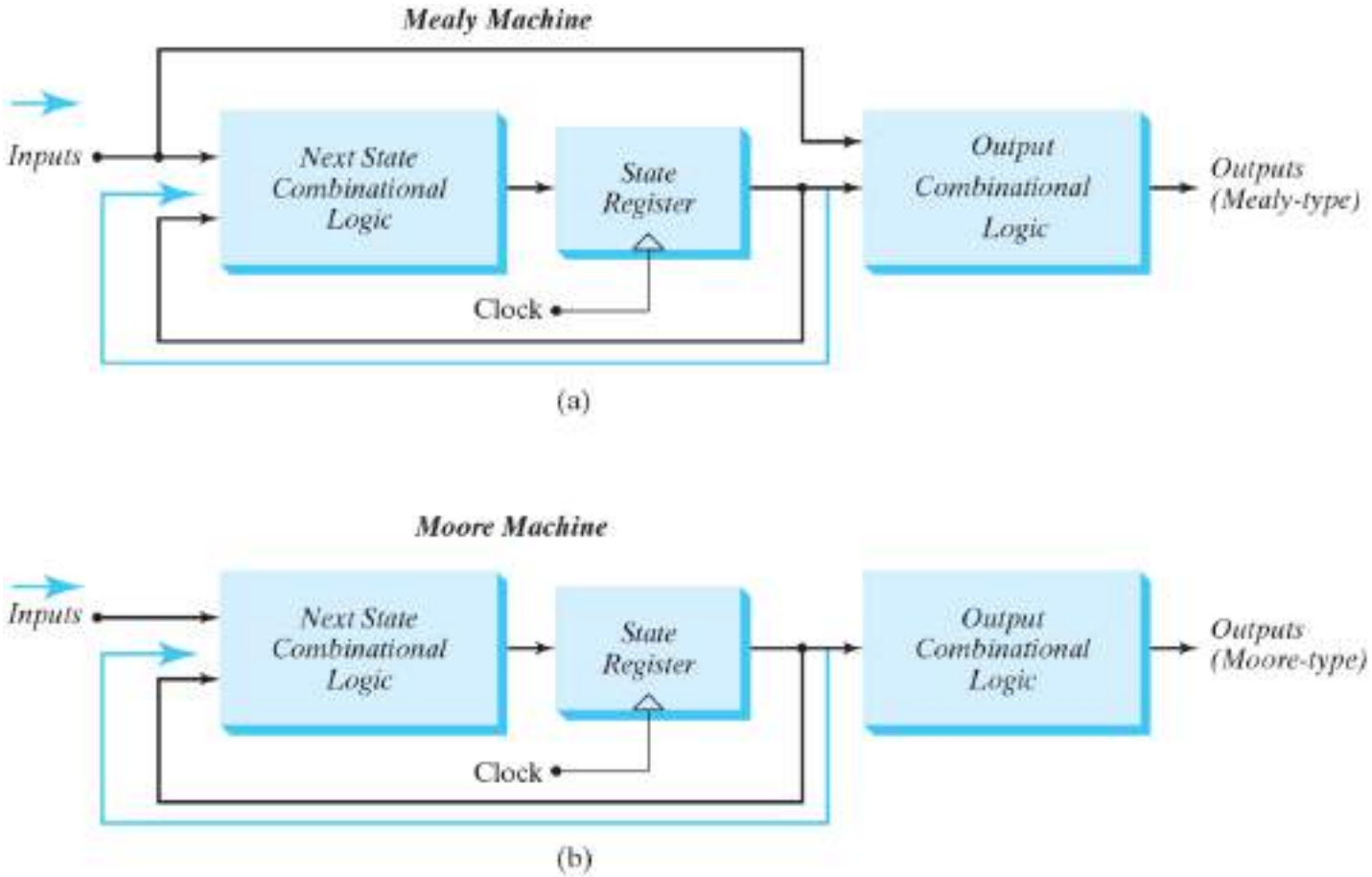
- 상태값은 특성식으로부터 상태식을 계산하여 얻어지고, 다음의 과정을 따름

1. 현재 상태와 입력 변수를 이용해서 입력식 결정
2. 입력식을 플립플롭 특성식으로 바꿈
3. 상태표에서 얻은 다음 상태를 결정하기 위해 상태식을 이용

- $A(t+1) = JA' + K'A$ (일반적인 JK 플립플롭의 입력식)
  $B(t+1) = JB' + K'B$ (일반적인 JK 플립플롭의 입력식)
- $A(t+1) = BA' + (Bx')'A = BA' + AB' + Ax$
  $B(t+1) = x'B' + (A \oplus x)'B = B'x' + ABx + A'Bx'$
- 그림 5.20(b)는 이 회로의 상태도임. 여기서 출력값은 입력값에는 무관하고 현재 상태에만 의존함.

- 유한 상태 머신의 밀리모델과 무어모델
  - 그림 5.21처럼 밀리 모델은 출력이 입력과 현재상태에 관한 함수이고, 무어 모델은 출력이 현재상태에 관한 함수임
  - 그림 5.15/16은 밀리 모델이고 그림 5.18/19는 무어 모델임
  - 무어 모델에서 순차 회로의 출력은 클럭으로 동기화됨. 왜냐하면 회로 출력이 클럭에 동기화되는 플립플롭의 출력에만 종속적이기 때문임. 그러나 밀리 모델에서는 클럭의 한 주기 동안에 입력이 바뀌면 출력이 바뀔 수 있음. 따라서 밀리형 회로를 동기화하기 위해서 순차 회로의 입력은 클럭과 동기화하고 출력은 클럭의 활성 에지 바로 전의 값으로 함

Figure 5.21
Block diagrams of Mealy and Moore state machines.



5.7 상태 축소와 할당

- 순차 회로 분석은 회로도에서 시작해서 상태표나 상태도에서 끝남
- 순차 회로 설계(5.8장)는 규격들의 집합에서 시작해서 논리도로 끝남
- 여기서는 설계과정에서 게이트와 플립플롭의 수를 줄여 설계를 간략화할 수 있는 유용한 성질들을 설명

- 상태축소: 순차 회로의 플립플롭의 수를 줄이는 것. 상태표에서 상태 수를 줄이면 플립플롭의 갯수를 줄일 수 있음. 다만 조합 게이트가 더 필요할 수도 있음
- 예를 들어,그림 5.25의 상태도를 보면, 원 안에 표시된 상태는 특별한 의미 없이 순서를 제공함. 여기에 초기 상태 a에서 시작하는 입력 순서열 01010110100을 생각해 보자.

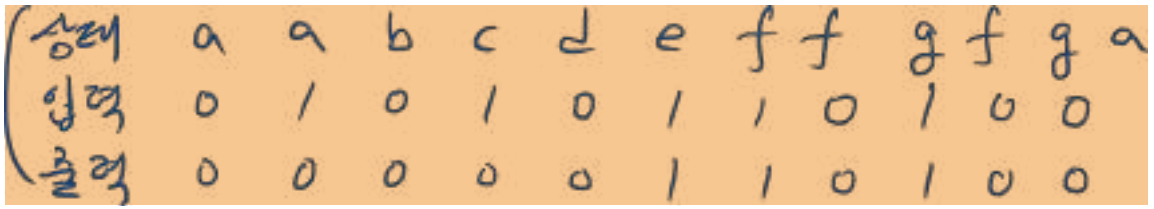


Figure 5.25
State diagram.

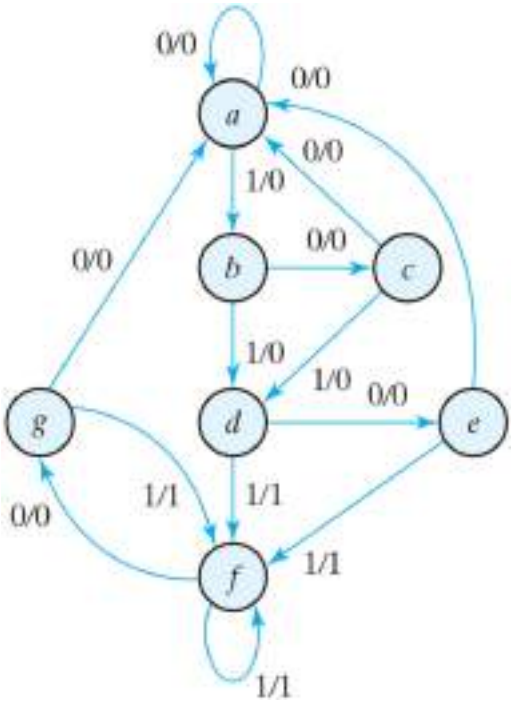


Table 5.6
State Table.

Present State	Next State		Output	
	$x = 0$	$x = 1$	$x = 0$	$x = 1$
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	f	0	1
e	a	f	0	1
f	g	f	0	1
g	a	f	0	1

- 상태표에서 다음 상태와 출력이 같은 상태를 찾아보면 e와 g가 같음. g행을 삭제하고 다음 상태 g를 모두 e로 변경 → 표 5.7이 됨

Table 5.7
Reducing the State Table.

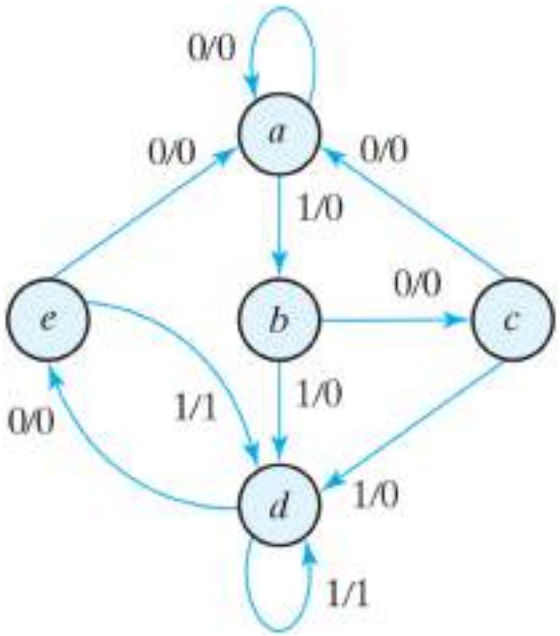
Present State	Next State		Output	
	$x = 0$	$x = 1$	$x = 0$	$x = 1$
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	f	0	1
e	a	f	0	1
f	e	f	0	1

- 위의 표에서 d와 f가 다음 상태와 출력이 동일한 등가이므로 f를 d로 대체함

Table 5.8
Reduced the State Table.

Present State	Next State		Output	
	$x = 0$	$x = 1$	$x = 0$	$x = 1$
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	d	0	1
e	a	d	0	1

Figure 5.26
Reduced State diagram.



- 그림 5.26의 상태도에 초기 상태 a에서 시작하는 입력 순서열 01010110100을 다시 넣어보면 상태는 다르지만 출력은 동일함

상태	a	a	b	c	d	e	d	d	e	d	e	a
입력	0	1	0	1	0	1	1	0	1	0	0	
출력	0	0	0	0	0	1	1	0	1	0	0	

- 상태할당: m개의 상태를 가진 회로는 코드가 식 $2^n >= m$을 만족하도록 n개의 비트를 사용해야 함. 사용하지 않는 상태는 don't care 조건이므로 잘 활용하면 게이트를 적게 사용할 수 있음
- 앞 슬라이드의 예제 표 5.6의 상태표가 사용 되었다면 2진 값을 7개의 상태에 할당해야 함. 나머지 한 상태는 사용하지 않음. 표 5.8의 상태표가 사용 되었다면 5개의 상태에 대한 할당이 필요하고 3개는 don't care 조건이 됨.
- 5개의 상태를 코드화하는 방법은 2진수 오름차순, 그레이코드, 원핫 할당

Table 5.9
Three Possible Binary State Assignments.

State	Assignment 1, Binary	Assignment 2, Gray Code	Assignment 3, One-Hot
a	000	000	00001
b	001	001	00010
c	010	011	00100
d	011	010	01000
e	100	110	10000

Table 5.10
Reduced State Table with Binary Assignment 1.

Present State	Next State		Output	
	$x = 0$	$x = 1$	$x = 0$	$x = 1$
000	000	001	0	0
001	010	011	0	0
010	000	011	0	0
011	100	011	0	1
100	000	011	0	1

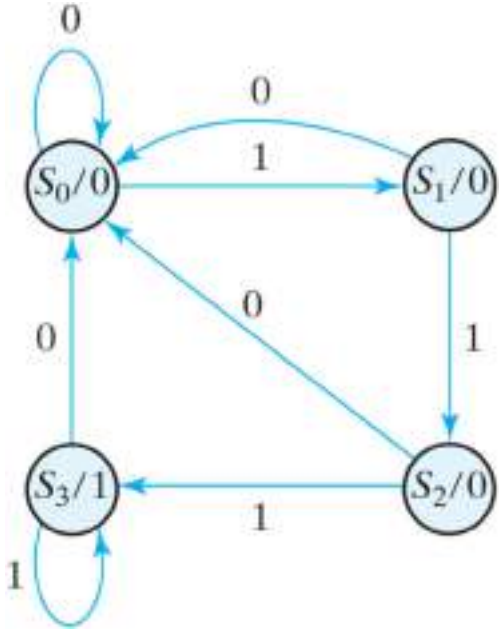
5.8 설계 과정

- 순차 회로 설계 과정: 플립플롭과 조합게이트로 구성. 플립플롭의 갯수는 상태수와 선택된 상태할당 코드에 의해 결정되고, 조합회로는 플립플롭의 입출력식을 계산해서 얻을 수 있는 상태표에서 유도됨
- 순차 뢰로 설계과정은
  - 기술된 내용과 요구되는 동작의 규격으로부터 회로의 상태도를 유도
  - 상태의 갯수를 줄임
  - 각 상태에 2진 값을 할당
  - 2진수를 코드화된 상태표를 구함
  - 플립플롭 종류 선택
  - 간략화된 플립플롭 입출력식을 유도
  - 논리도를 그림.

} 합성툴(SW)로 가능

- 예를 들어, 입력 라인을 통해 들어오는 비트의 나열에서 1 값이 3개 또는 그 이상으로 연속되는 곳을 검출하는 회로를 설계하면, (무어 모델?)

Figure 5.27
State diagram for sequence detector.



- D 플립플롭을 이용한 합성
  - 수작업으로 회로를 설계하려면 상태표를 만들고 상태를 2진 값으로 할당. 4개의 상태이므로 2개의 D 플립플롭 선택 (A, B).
  - D 플립플롭의 특성식은 Q(t+1) = DQ이고 표 5.11의 상태표 작성.

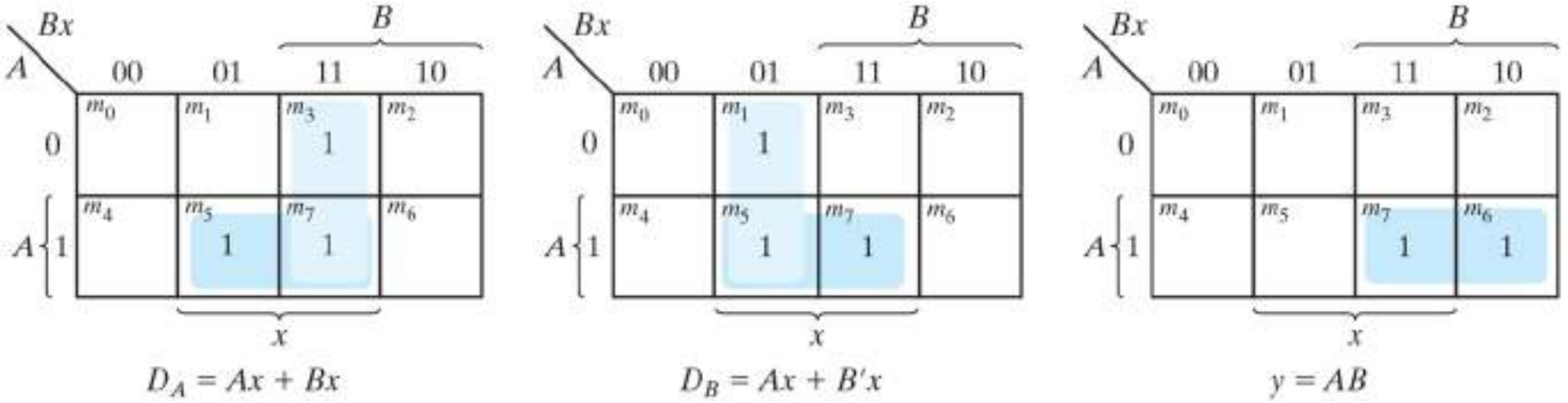
Table 5.11
State Table for Sequence Detector.

Present State		Input	Next State		Output
A	*B*	*x*	*A*	*B*	*y*
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	1	1	0
1	1	0	0	0	1
1	1	1	1	1	1

– 플립플롭의 입력식은 다음 상태 값들로 이루어진 A와 B열에서 직접 얻음. 그리고 다음과 같이 최소항의 합으로 표현됨

$$A(t+1) = D_A(A, B, x) = \Sigma(3, 5, 7)$$
$$B(t+1) = D_B(A, B, x) = \Sigma(1, 5, 7)$$
$$y(A, B, x) = \Sigma(6, 7)$$

Figure 5.28
K-Maps for sequence detector.



- 위의 그림처럼 카노맵을 통해서 간략화하면,

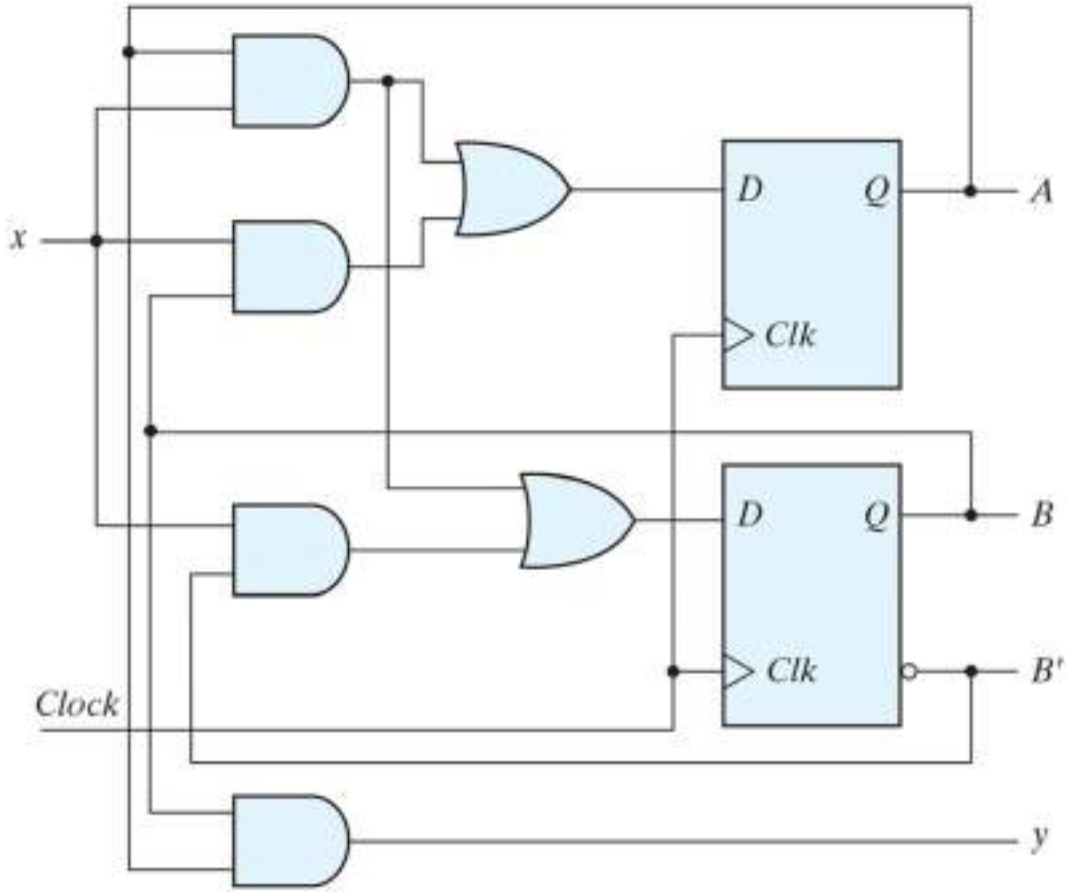
  $D_A = Ax + Bx$

  $D_B = Ax + B'x$

  $y = AB$

- 따라서 그림 5.29처럼 논리도를 얻을 수 있음. D 플립플롭을 사용한 설계의 장점은 플립플롭의 입력을 나타내는 입력을 나타내는 부울 식을 상태표에서 직접 얻을 수 있다는 점

Figure 5.29
Logic diagram of a Moore-type sequence detector.



- 여기표
  - D 플립플롭이 아닌 다른 플립플롭(JK, T)을 이용해서 순차 회로를 설계하면 더 복잡해짐. (입력식이 상태표에서 직접 구해지지 않기 때문에)
  - 이 경우, 플립플롭의 입력식을 구하기 위해서는 상태표와 입력식 사이의 함수 관계를 구해야 함
  - 여기표(excitation table): 현재 상태에서 다음 상태로의 천이와 이를 위한 플립플롭의 입력 조건을 나열한 표

Table 5.12
Flip-Flop Excitation Tables.

Q(t)	Q(t = 1)	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

(a) *JK* Flip-Flop

Q(t)	Q(t = 1)	T
0	0	0
0	1	1
1	0	1
1	1	0

(b) *T* Flip-Flop

- JK 플립플롭을 이용한 합성
  - D 플립플롭을 가진 회로의 경우와 다른 점은 입력식을 여기표에서 알 수 있는 현재 상태에서 다음 상태로의 상태 천이로부터 구한다는 점

Table 5.13
State Table and JK Flip-Flop Inputs.

Present State		Input	Next State		Flip-Flop Inputs			
A	B	x	A	B	J_A	K_A	J_B	K_B
0	0	0	0	0	0	X	0	X
0	0	1	0	1	0	X	1	X
0	1	0	1	0	1	X	X	1
0	1	1	0	1	0	X	X	0
1	0	0	1	0	X	0	0	X
1	0	1	1	1	X	0	1	X
1	1	0	1	1	X	0	X	0
1	1	1	0	0	X	1	X	1

  - 위의 표는 표 5.11에 JK FF에 대한 여기표를 확장한 것임
  - 이를 다음 슬라이드의 J_A, K_A, J_B, K_B에 대한 카노맵으로 간략화함
  - 이후 그림 5.31처럼 논리도를 그림

Figure 5.30
Maps for *J* and *K* input equations.

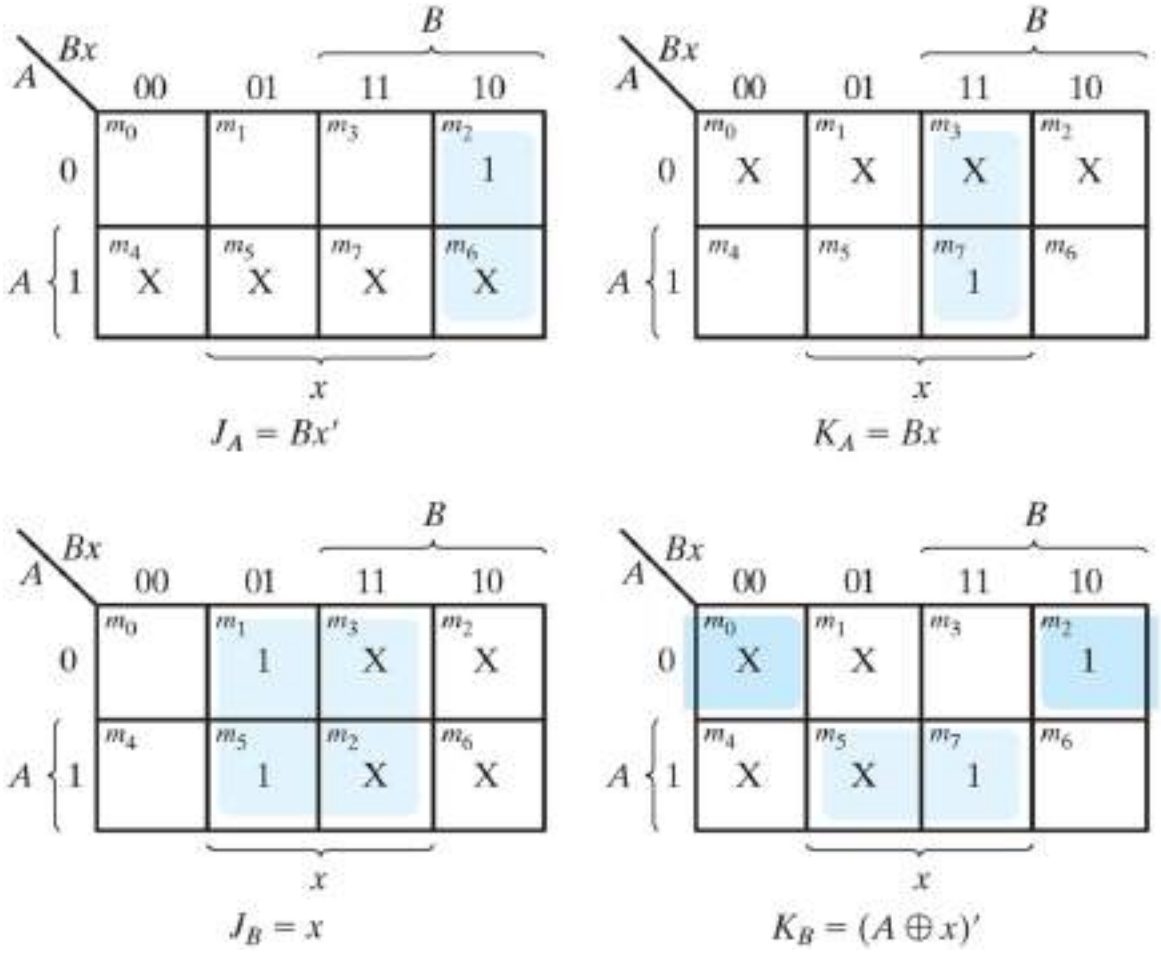
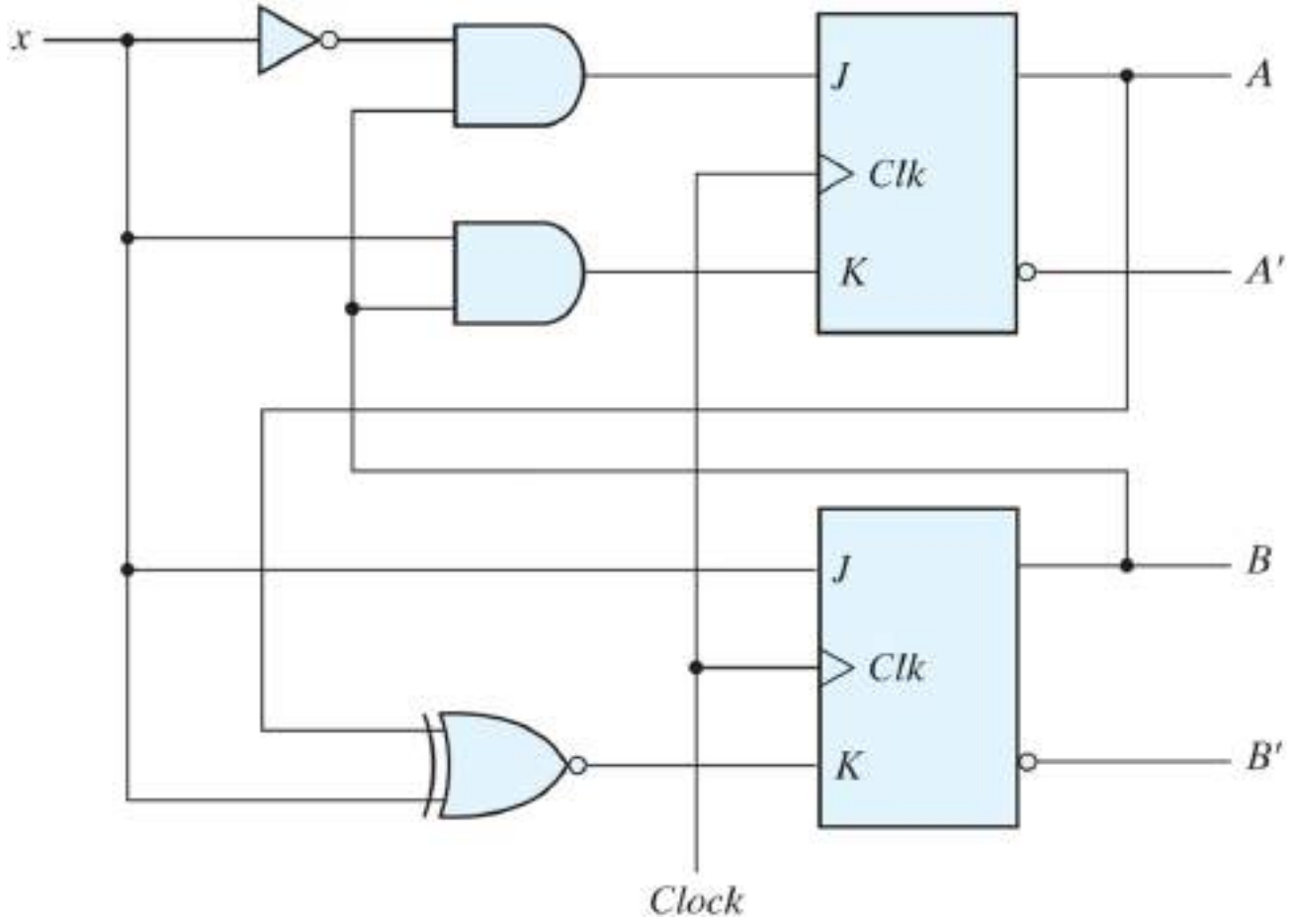
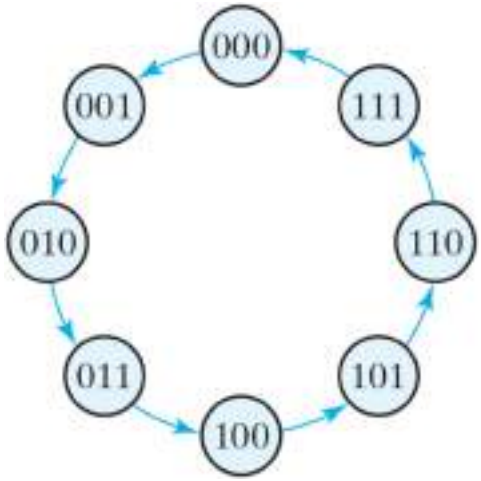


Figure 5.31
Logic diagram for sequential circuit with *JK* flip-flops.



- T 플립플롭을 이용한 합성
  - 2진 카운터 설계를 통해 설명할 것임. n비트 2진 카운터는 0에서 2^n-1까지 셀 수 있는 n개의 플립플롭으로 구성됨. 000 → … → 111 → 000 → … → 111 --> …

Figure 5.32
State diagram of three-bit binary counter.



  - 카운터의 상태도는 입력값과 출력값을 나타내지 않음. 회로의 입력은 클럭 뿐이고 출력값은 플립플롭의 현재 상태임

Table 5.14
State Table for Three-Bit Counter.

Present State			Next State			Flip-Flop Inputs		
A_2	A_1	A_0	A_2	A_1	A_0	T_{A2}	T_{A1}	T_{A0}
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	1	0	1	0	0	1
1	0	1	1	1	0	0	1	1
1	1	0	1	1	1	0	0	1
1	1	1	0	0	0	1	1	1

Figure 5.33
Maps for three-bit binary counter.

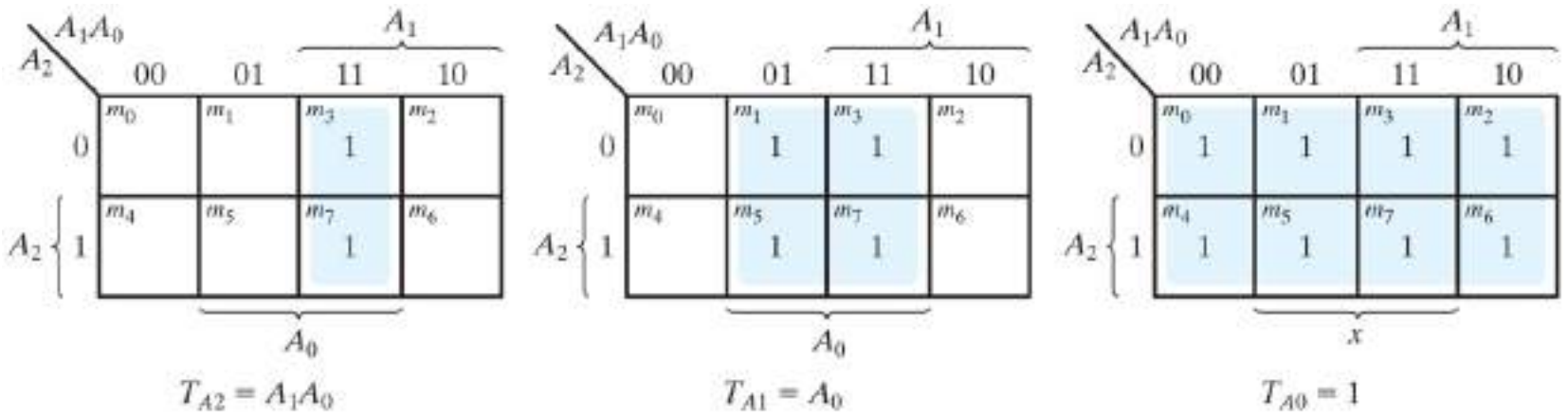


Figure 5.34
Logic diagram of three-bit binary counter.