

조합회로 정리

- 입력 변수들이 출력값을 결정한다.
- F_i (입력변수들) = 출력값



1

1

Sequential Circuits p.185

순차회로 = 조합회로 + 2진 정보 기억요소

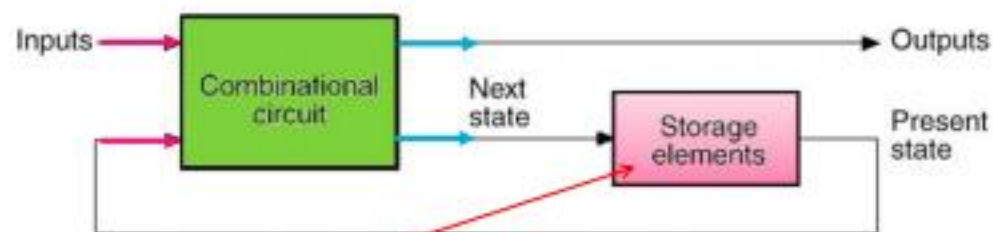
기억요소 : 정보를 저장할 수 있는 회로

2

2

Sequential Circuit Definitions p.186

■ 순차회로의 블록 다이어그램



■ 상태(state)

- 주어진 시간에 기억장치 요소에 **저장된 2진 정보**

■ 순서회로 함수의 표현

- $F(\text{입력, 현 상태}) = \text{출력, 다음 상태}$

3

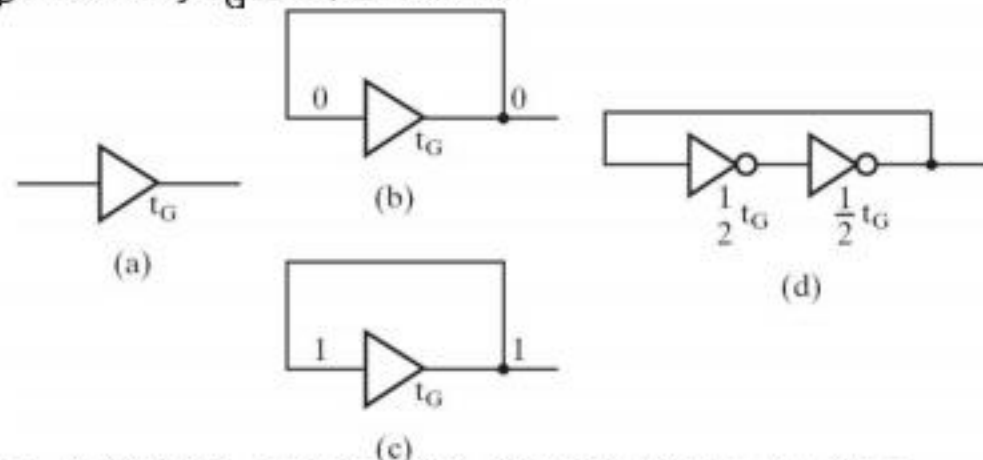
3

Logic Structures for Storing Information

■ 무한 저장 구조

p.187

■ (a) gate delay t_G 를 가진 buffer



- 가정 : (b)(c)(d) 무한히 임의의 정보를 저장할 수 있다.

■ (d)의 회로를 바꾼다면 비동기식 기억장치가 가능하지 않을까?

- 입력값에 따라 원하는 값(1,0)을 저장할 수 있게 하자.

4

4

Latch

p.189

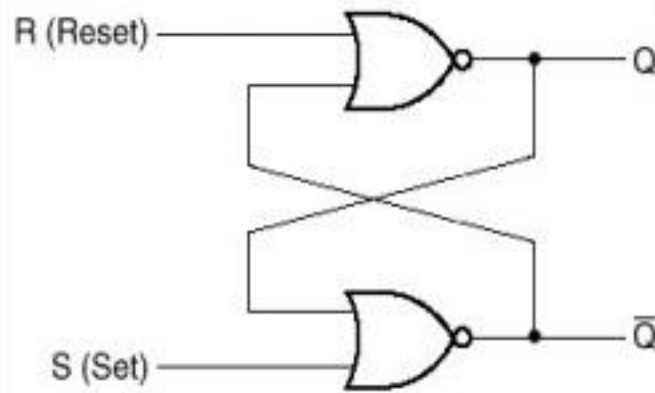
■ 래치 : 가장 기본적인 기억장치 요소

- 래치로 FF(동기식기억장치)를 만든다.

■ SR 래치 : NOR 게이트 이용

래치 설계의 기본 요건

1. 0 또는 1 저장
2. 저장 값을 바꿀 수 있게
3. Q와 Q'는 보수관계가 되도록



(a) Logic diagram

S	R	Q	Q̄
1	0	1	0
0	0	1	0
0	1	0	1
0	0	0	1
1	1	0	0

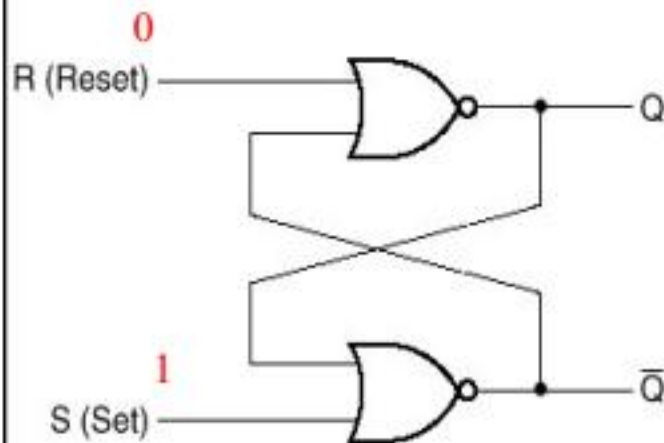
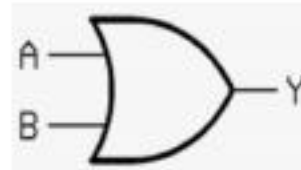
(b) Function table

5

5

Fuction Table 을 완성하라

p.189



(a) Logic diagram

S	R	Q	Q̄

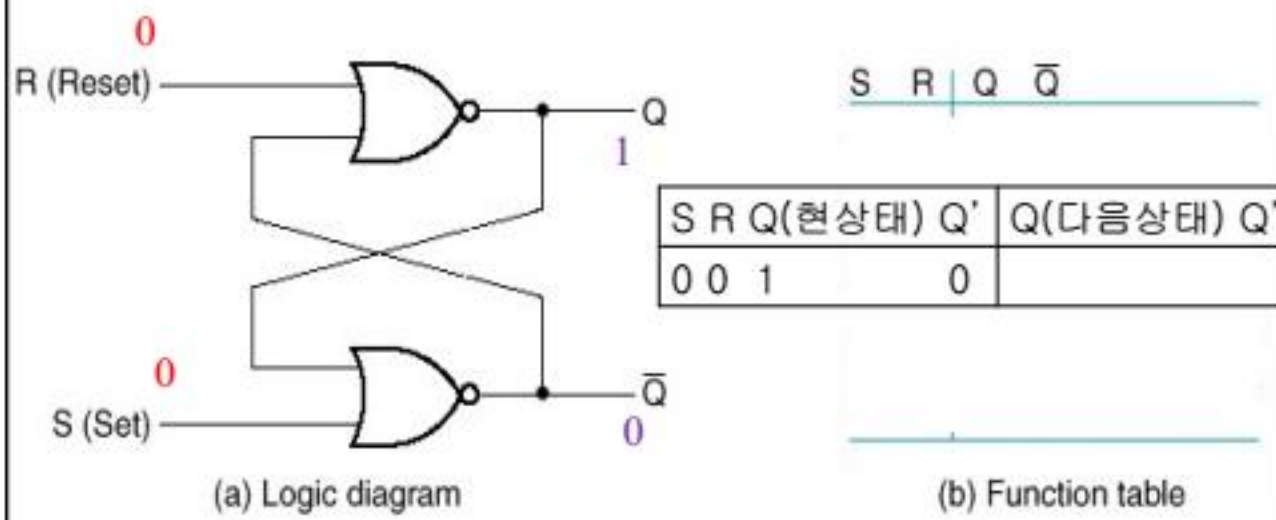
(b) Function table

6

6

Fuction Table 을 완성하라

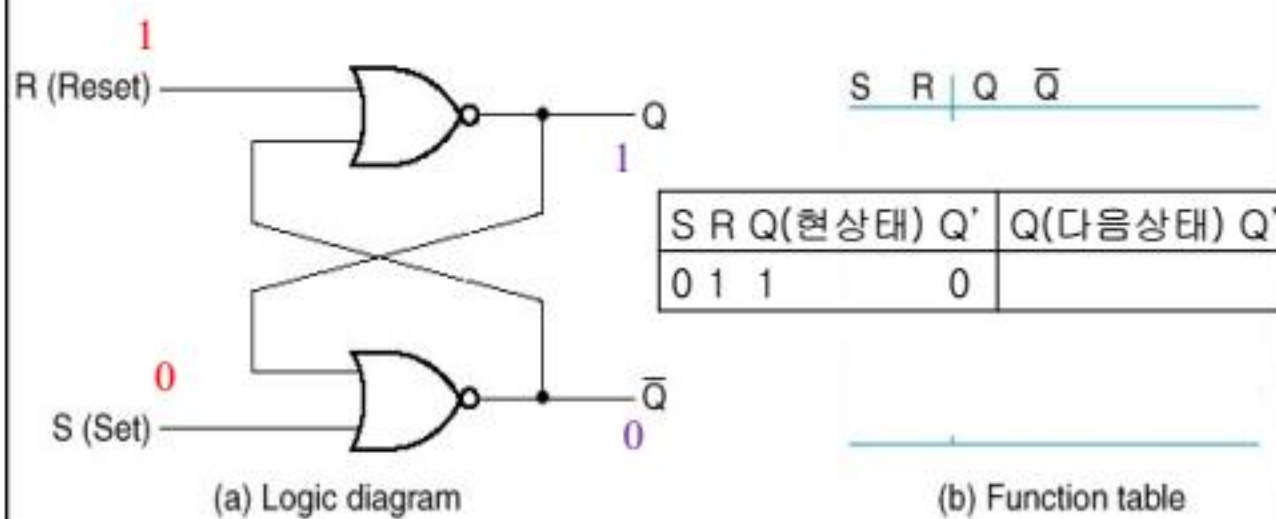
p.189



7

Fuction Table 을 완성하라

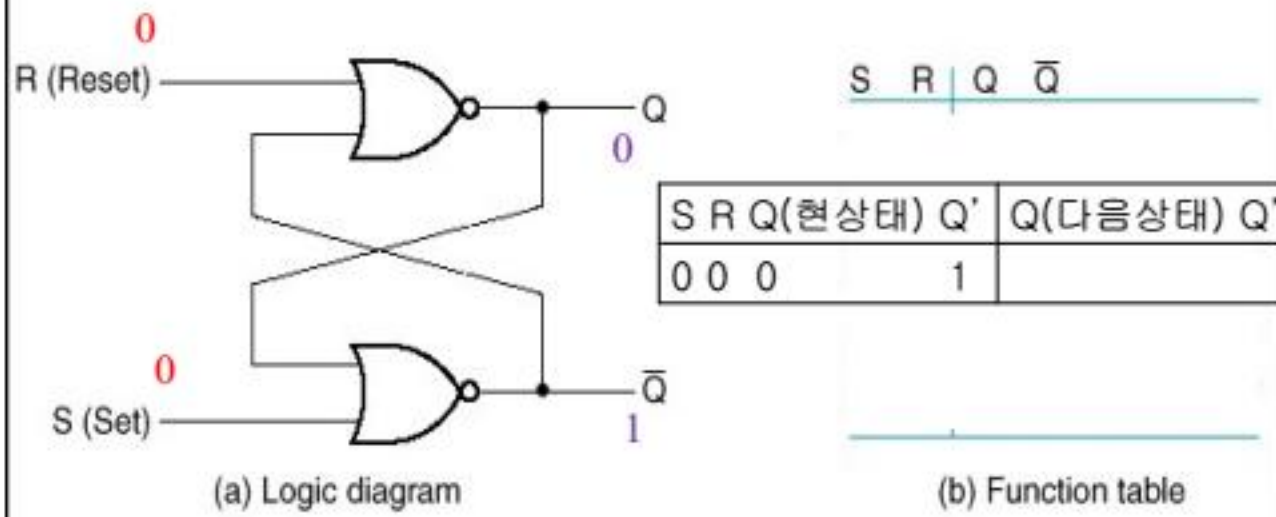
p.189



8

Fuction Table 을 완성하라

p.189

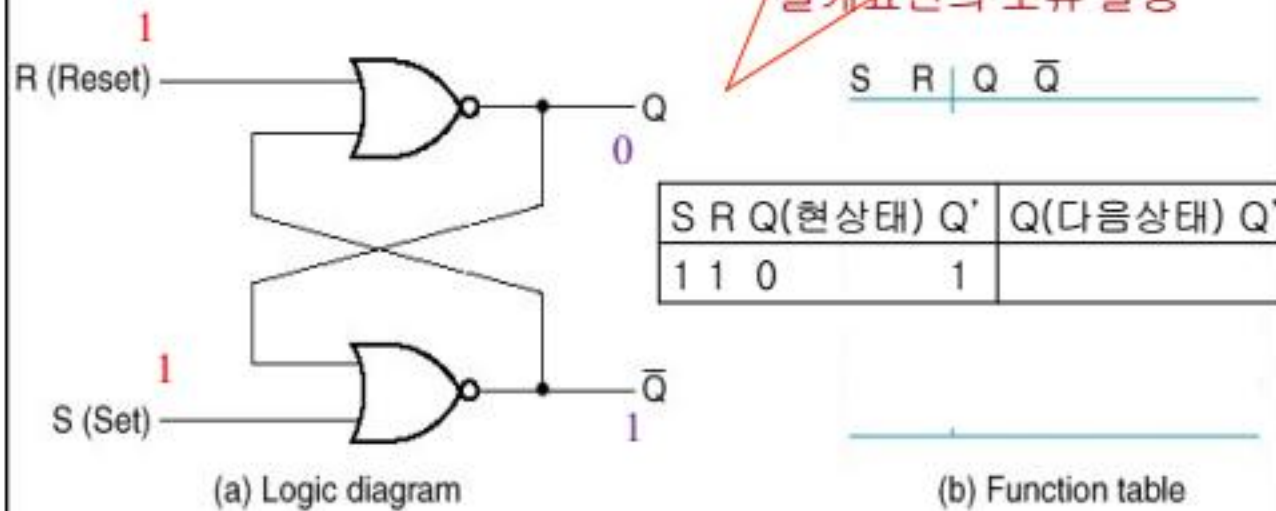


9

Fuction Table 을 완성하라

래치 설계의 기본 요건

3. Q와 Q'는 보수관계가 되도록
설계요건의 오류 발생



10

10

Latch

p.189

■ 래치 : 가장 기본적인 기억장치 요소

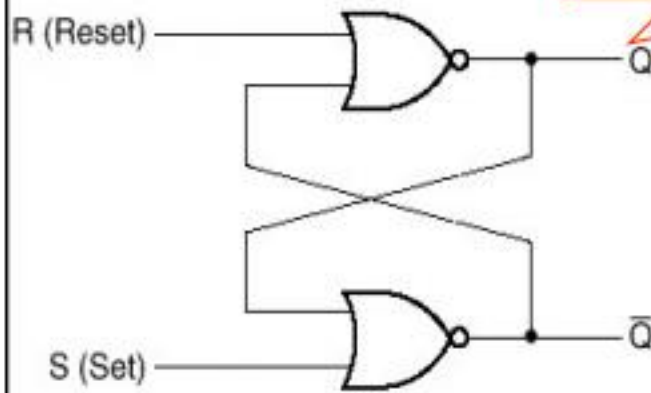
- 래치로 FF를 만든다.

■ SR 래치

- NOR 게이트 이용

래치 설계의 기본 요건

1. 0 또는 1 저장
2. 저장 값을 바꿀 수 있게
3. Q와 Q'는 보수관계가 되도록



(a) Logic diagram

S	R	Q	Q'
1	0	1	0
0	0	1	0
0	1	0	1
0	0	0	1
1	1	0	0

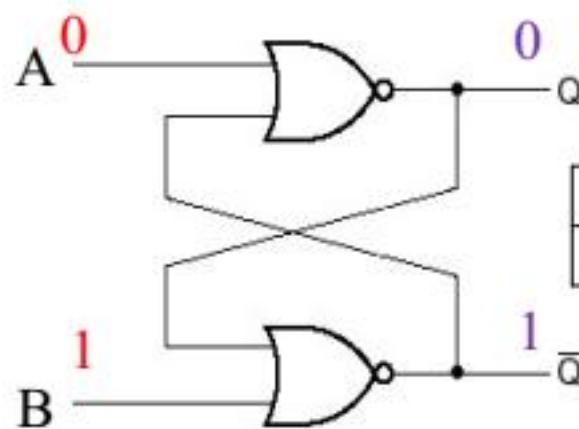
(b) Function table

11

11

p.189

Fuction Table 을 완성하라



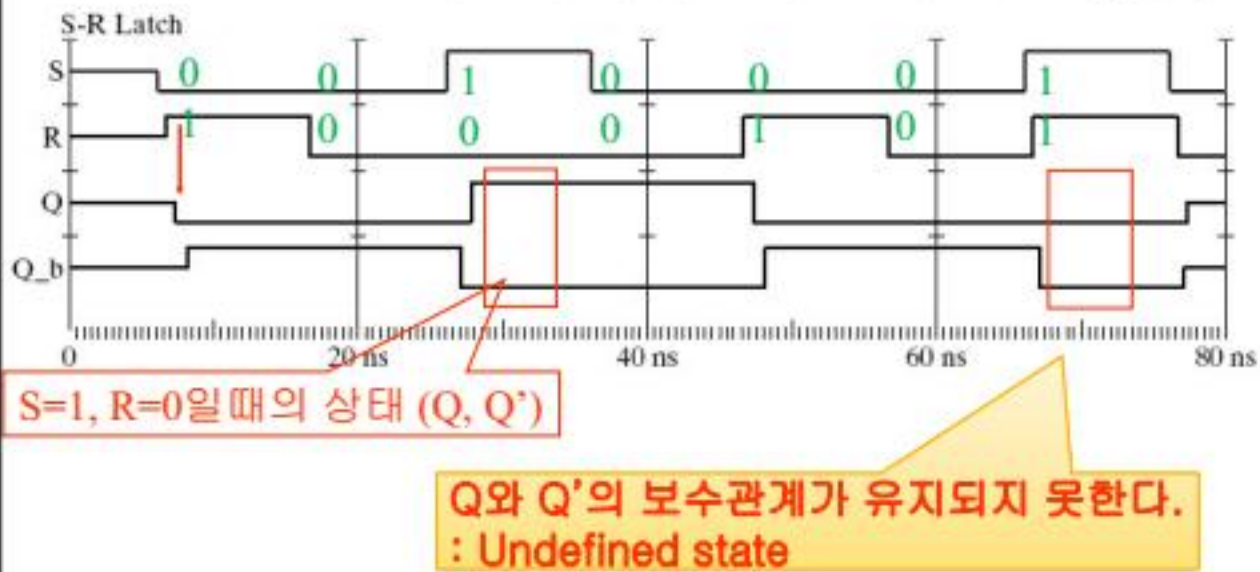
(a) Logic diagram

A	B	Q(현상태)	Q'	Q(다음상태)	Q'
0	1	0	1		

12

12

SR Latch의 타이밍 시뮬레이션 p.190



13

13

SR latch의 보수관계 래치가 필요하다.

- 2진수
- 1의 보수
- 곱의 합
- 합의 곱
- 최소항
- 최대항

S=1일 때 set

S	R	Q	\bar{Q}
1	0	1	0
0	0	1	0
0	1	0	1
0	0	0	1
1	1	0	0

Set state

Reset state

Undefined

↔

S=0일 때 set

S	R	Q	\bar{Q}
0	1	1	0
1	1	1	0
1	0	0	1
1	1	0	1
0	0	1	1

Set state

Reset state

Undefined

14

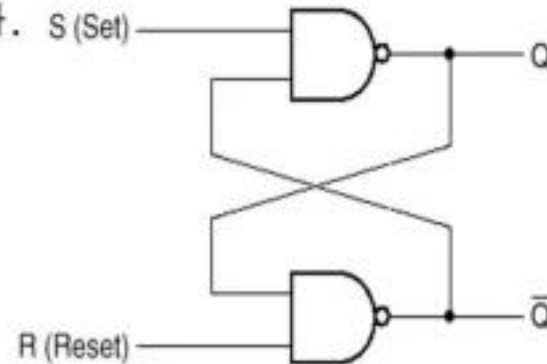
14

Latch with NAND gates

p.191

■ NAND 게이트 이용 S'R' 래치

- SR 래치의 입력신호와 보수의 입력(S' R')일때 같은 기능을 한다. S (Set)



(a) Logic diagram

S	R	Q	Q̄
0	1	1	0
1	1	1	0
1	0	0	1
1	1	0	1
0	0	1	1

(b) Function table

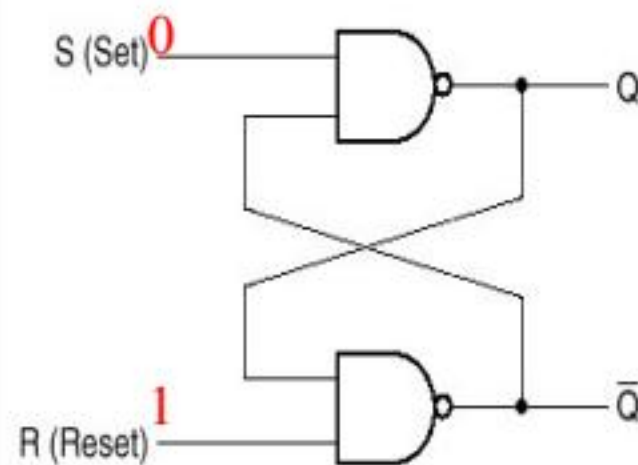
SR래치 의 보수 래치를 설계하자. → S,R 기호 붙이기
→ 명칭은 S'R'래치라 한다.

15

Fuction Table 을 완성하라



p.191



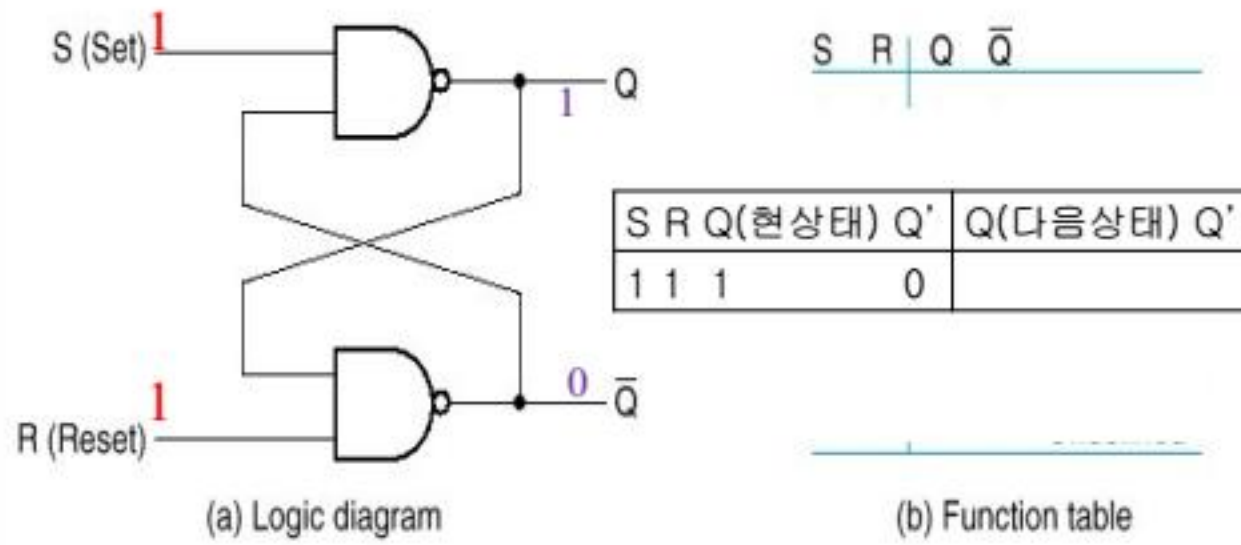
(a) Logic diagram

S	R	Q	Q̄

(b) Function table

16

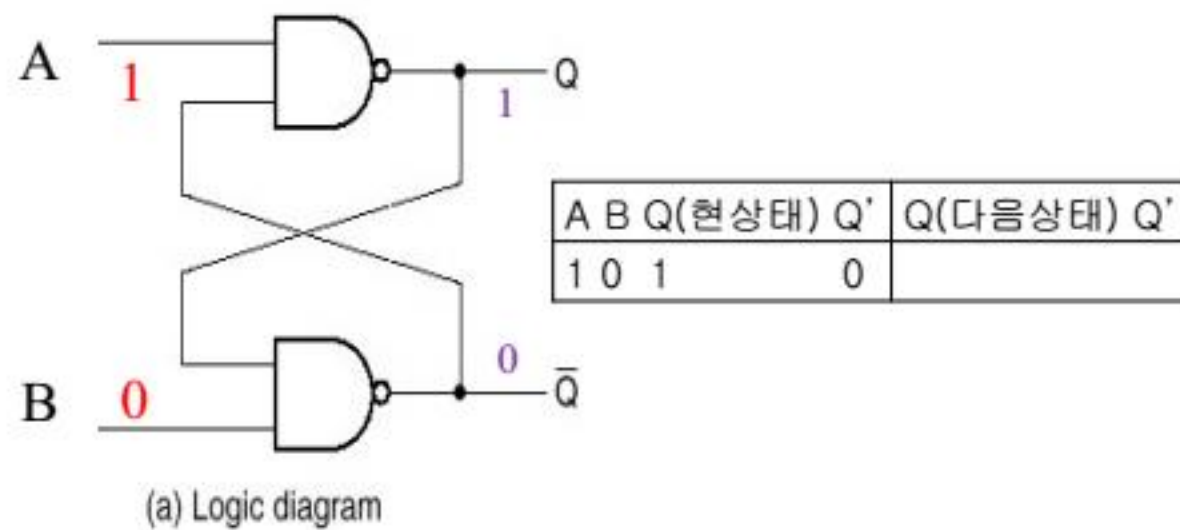
Fuction Table 을 완성하라



17

17

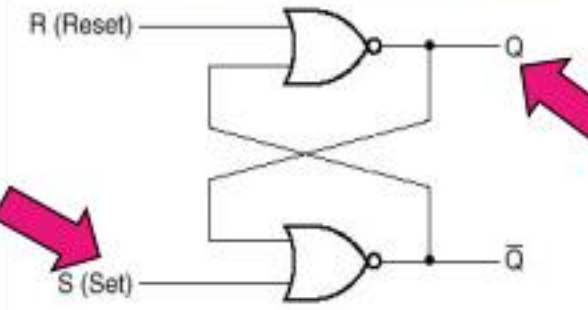
Fuction Table 을 완성하라



18

18

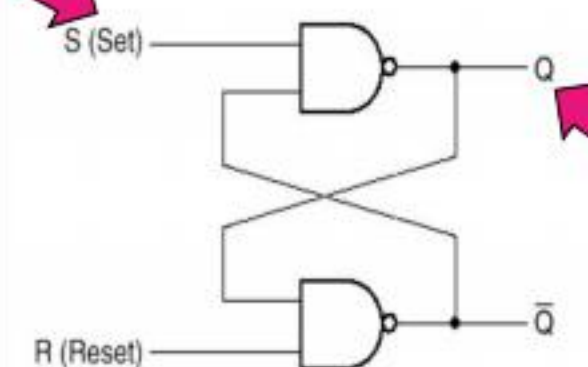
< SR latch, S'R' latch 정리 >



(a) Logic diagram

S	R	Q	\bar{Q}
1	0	1	0
0	0	1	0
0	1	0	1
0	0	0	1
1	1	0	0

(b) Function table



(a) Logic diagram

S	R	Q	\bar{Q}
0	1	1	0
1	1	1	0
1	0	0	1
1	1	0	1
0	0	1	1

(b) Function table

19

Latch with Control input

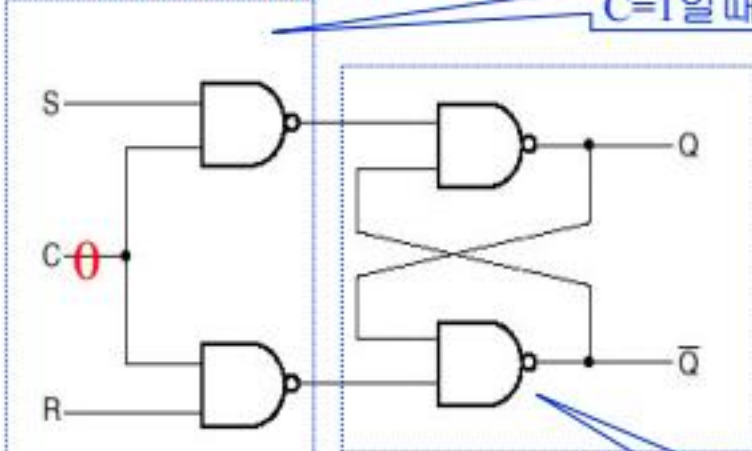
p.191

■ 제어입력을 갖는 SR 래치

— 제어입력 C : 래치의 상태가 변할 수 있는지 결정

— C 가 enable 신호로 동작

C=0일때 S, R에 상관없이 1이 출력
C=1일때 S, R는 S', R'로 출력



(a) Logic diagram

C	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	Undefined

(b) Function table

20

20

10

(a) Logic diagram

C	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	Undefined

(a) Logic diagram

C	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	Undefined

(b) Function table

21

21

(a) Logic diagram

C	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	Undefined

(a) Logic diagram

C	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	Undefined

(b) Function table

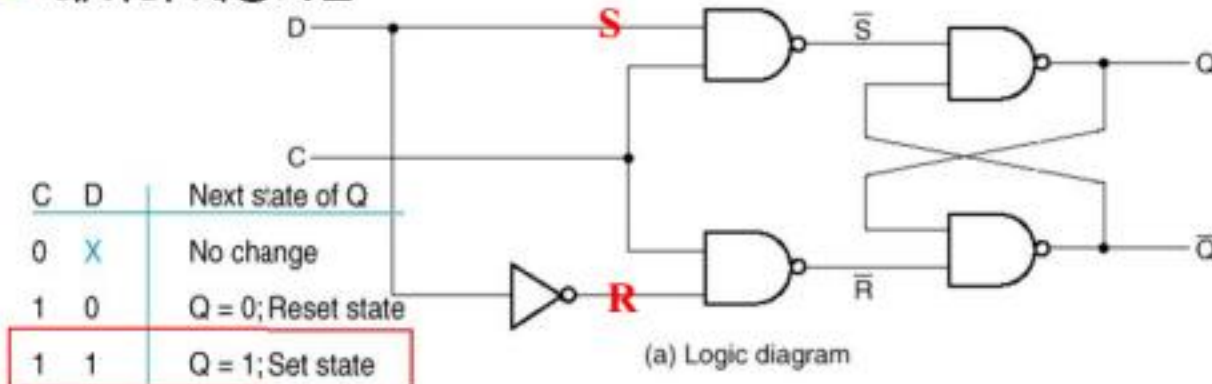
22

22

D Latch

p.192

- SR 래치의 문제점 : Undefined 상태
- 해결 : D 래치 → Undefined 상태 제거
- 두개의 입력 : D(data), C(control)
- 데이터 저장기능



C	D	Next state of Q
0	X	No change
1	0	Q = 0; Reset state
1	1	Q = 1; Set state

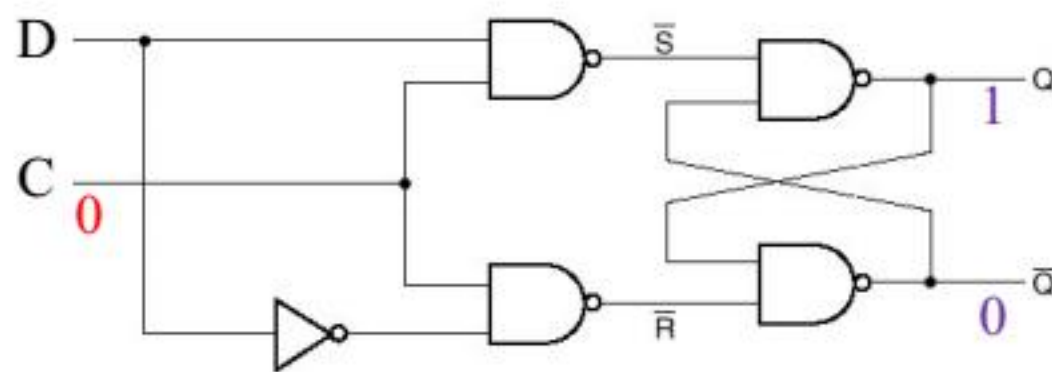
(b) Function table

23

23

Fuction Table 을 완성하라

p.192

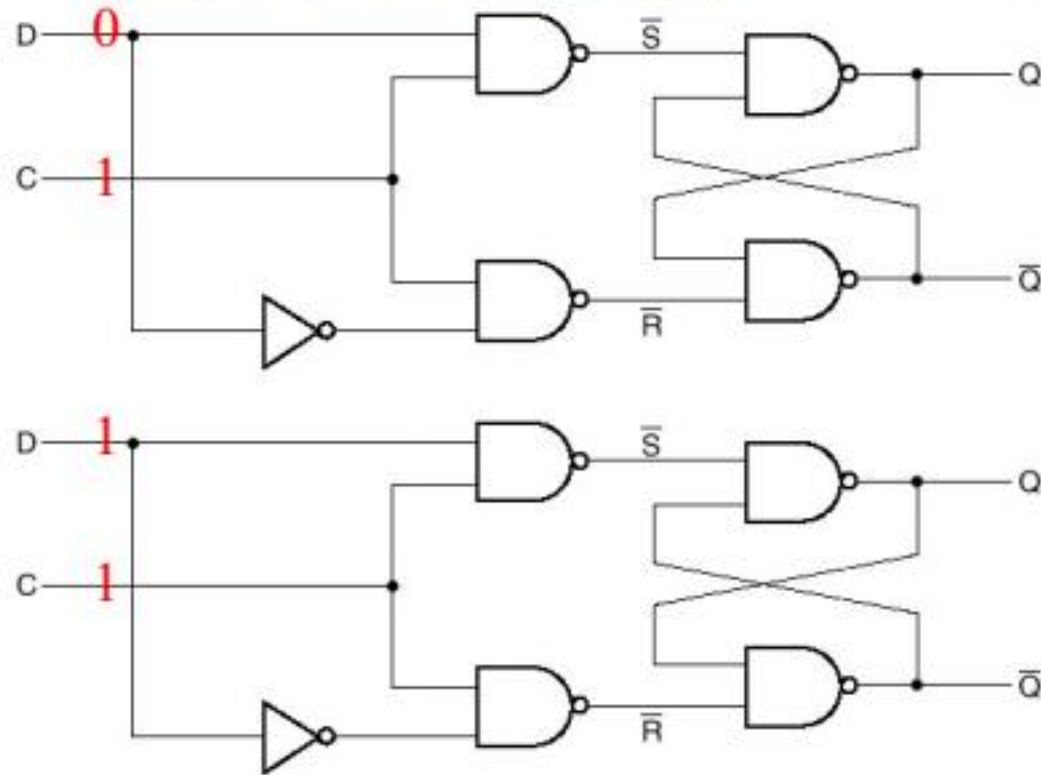


24

24

Function Table 을 완성하라

p.192

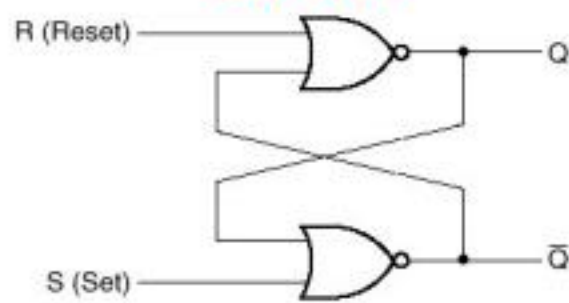


25

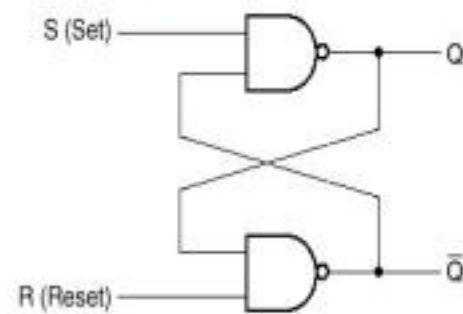
25

Latch 정리

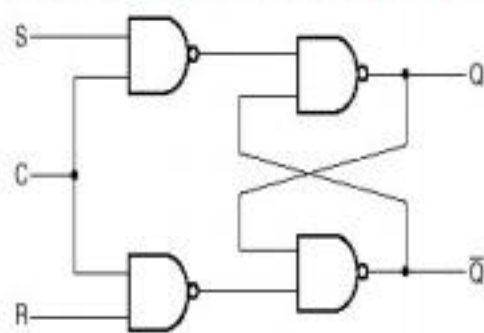
SR latch



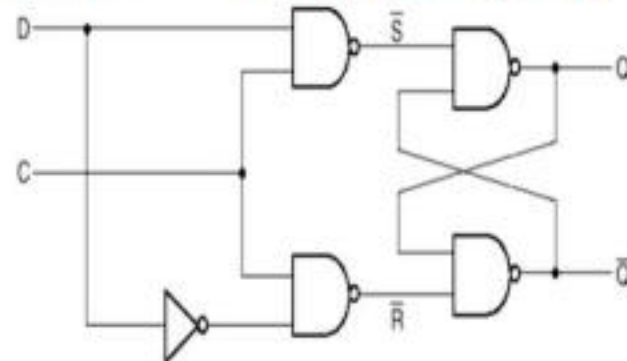
S'R' latch



SR latch with control(C=1, C=0)



D latch(C=1, C=0)



(a) Logic diagram

26

26