피드백과 플립플롭

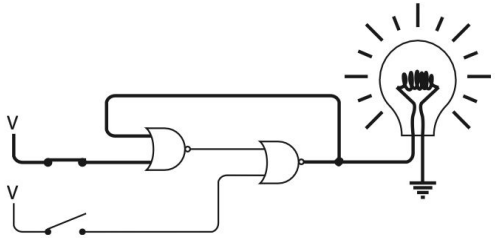
윤경담
스마트팜학과

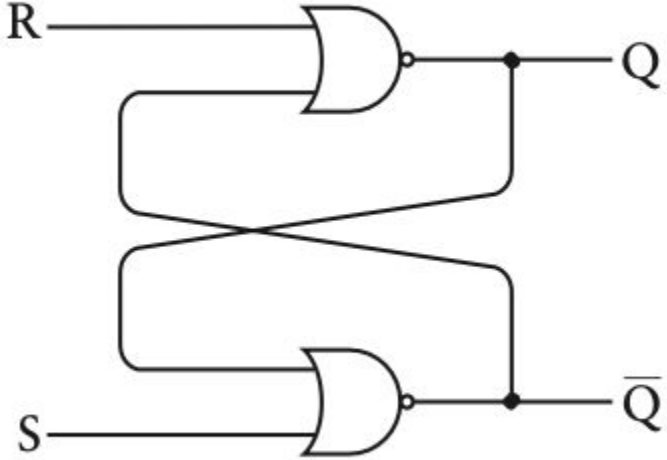
17장: 피드백과 플립플롭

NOR 게이트

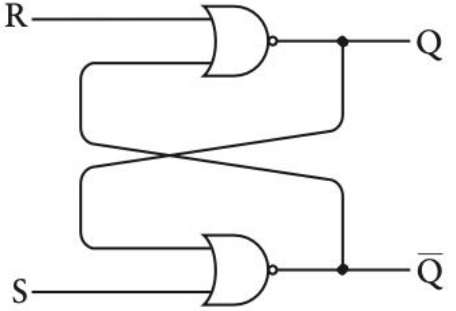
NOR	0	1
0	1	0
1	0	0

R-S 플립플롭 (Reset-Set Flip-Flop)



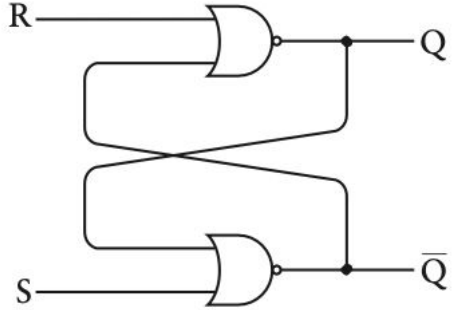


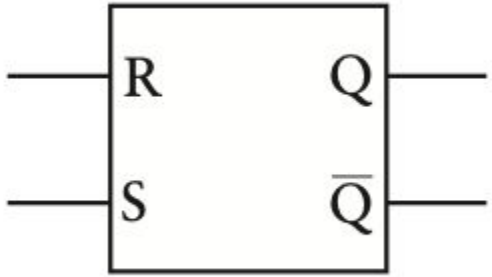
R-S 플립플롭 (Reset-Set Flip-Flop)



입력		출력	
S	R	Q	$\overline{Q}$
1	0	1	0
0	1	0	1
0	0	Q	$\overline{Q}$
1	1	허용되지 않음	

R-S 플립플롭 (Reset-Set Flip-Flop)





Reset/Set vs. Hold (보존)

입력		출력	
S	R	Q	$\overline{Q}$
1	0	1	0
0	1	0	1
0	0	Q	$\overline{Q}$
1	1	허용되지 않음	

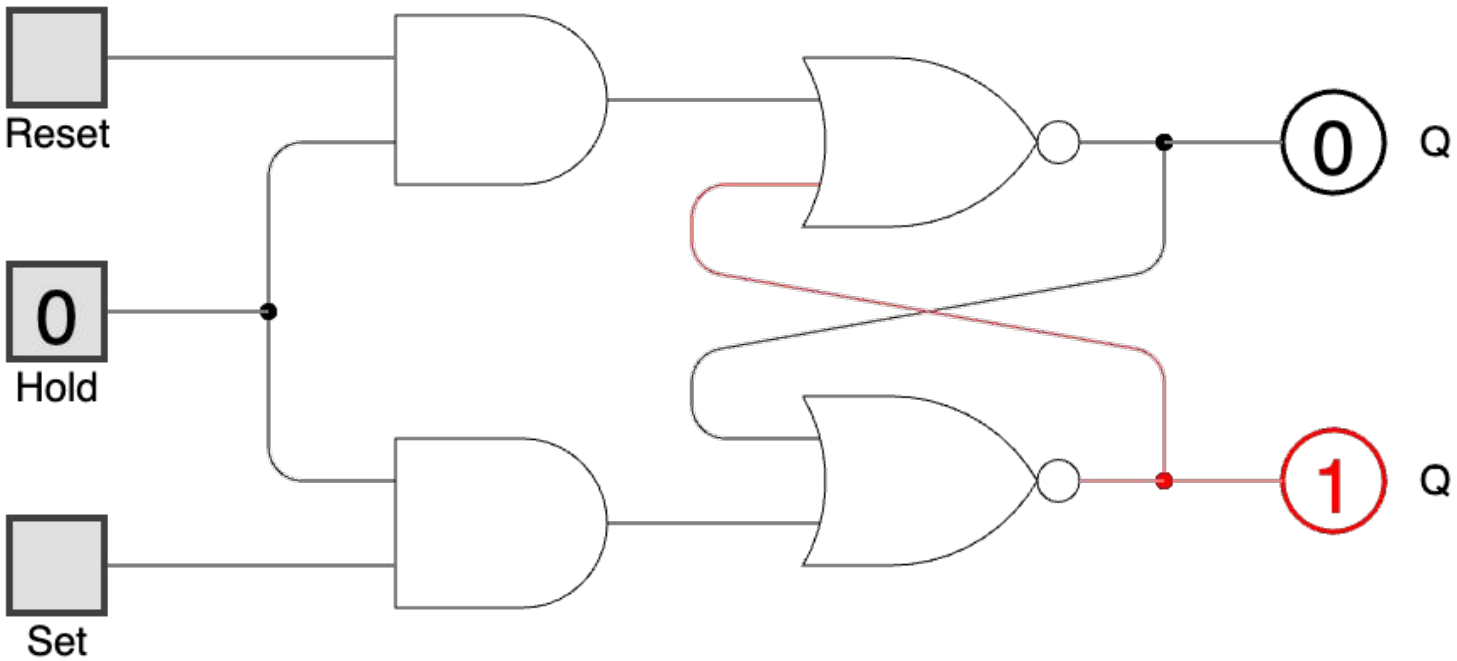
입력		출력
데이터	값 보존	Q
0	1	0
1	1	1
0	0	Q
1	0	Q

Reset/Set vs. Hold (보존)

입력		출력	
S	R	Q	$\overline{Q}$
1	0	1	0
0	1	0	1
0	0	Q	$\overline{Q}$
1	1	허용되지 않음	

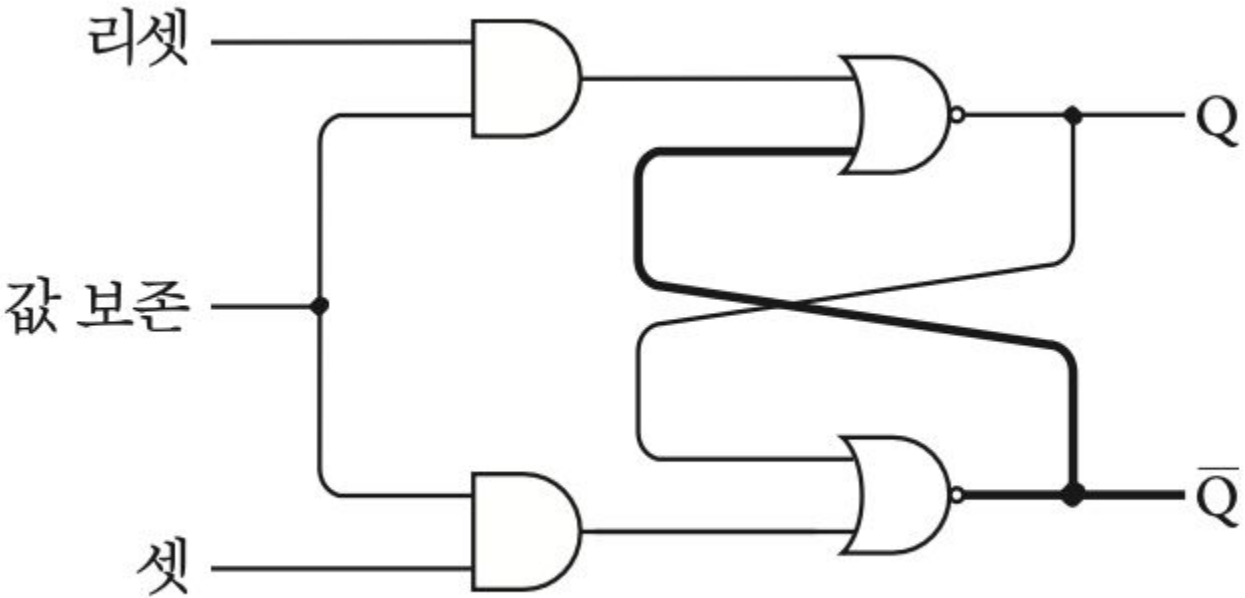
입력		출력
데이터	값 보존	Q
0	1	0
1	1	1
X	0	Q

Hold: 예제

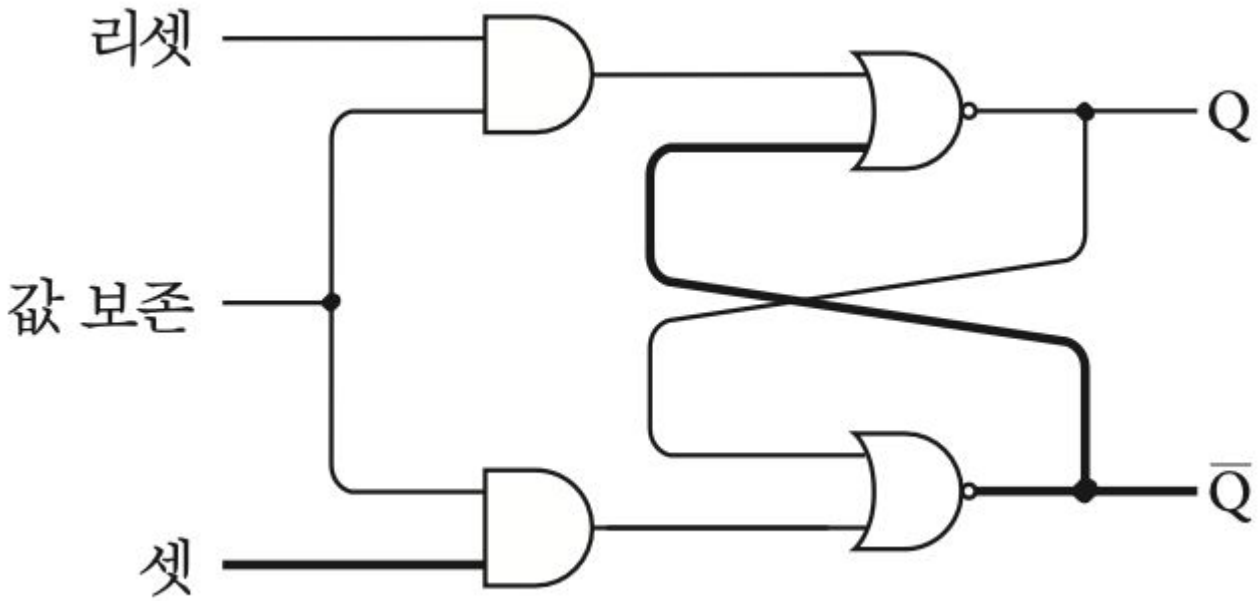


https://www.codehiddenlanguage.com/Chapter17/

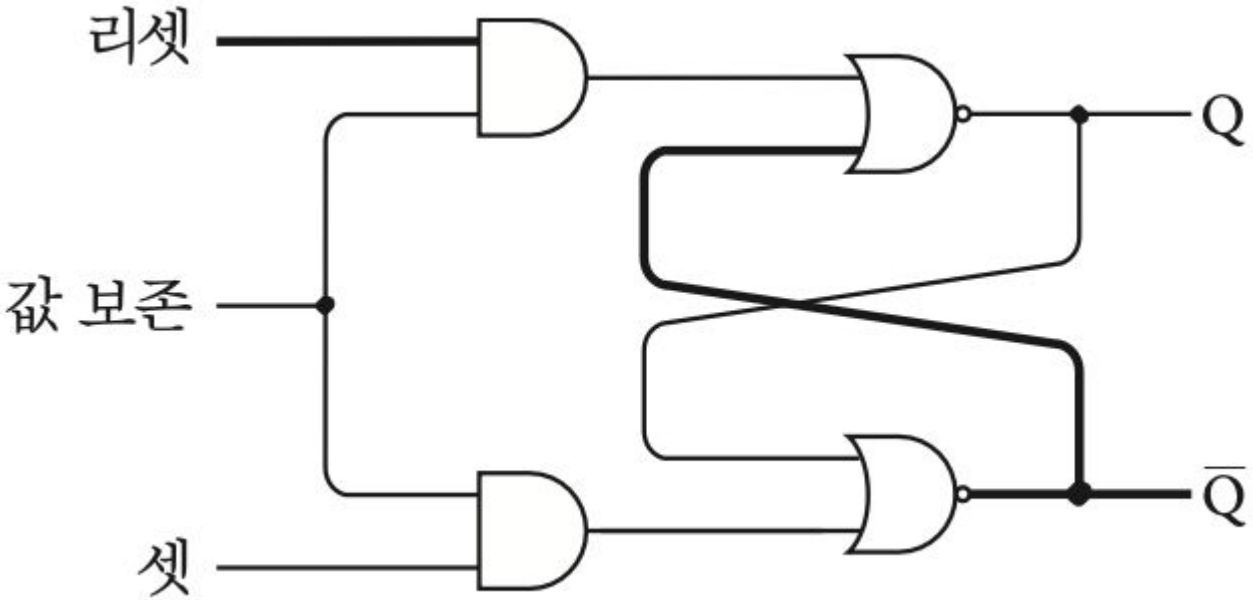
Hold (보존)



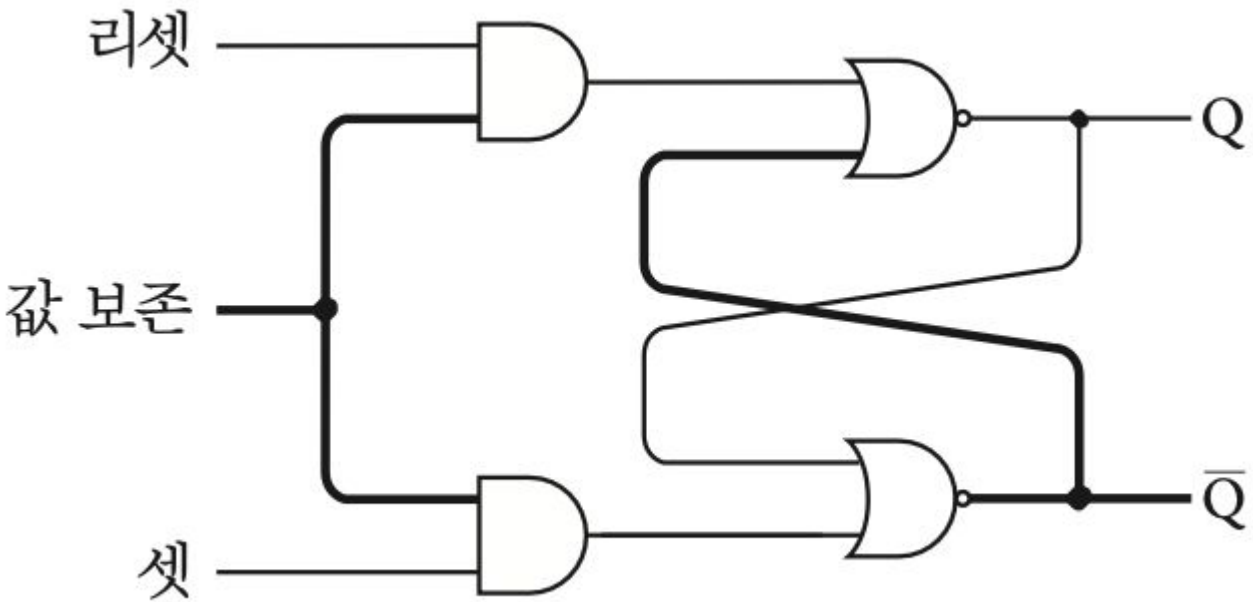
Hold (보존)



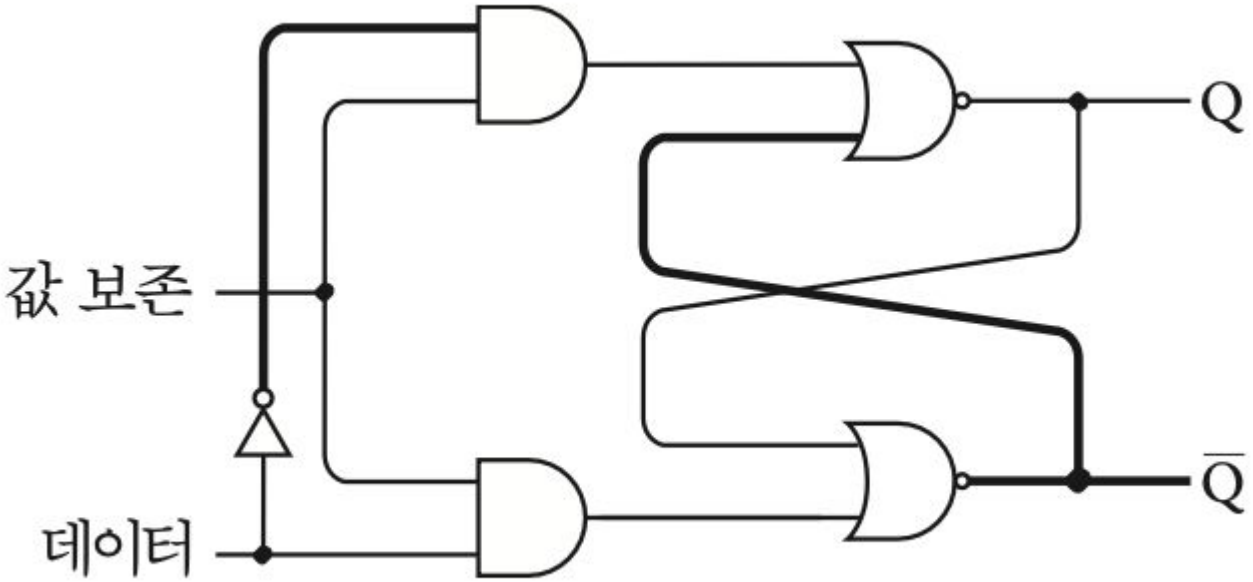
Hold (보존)



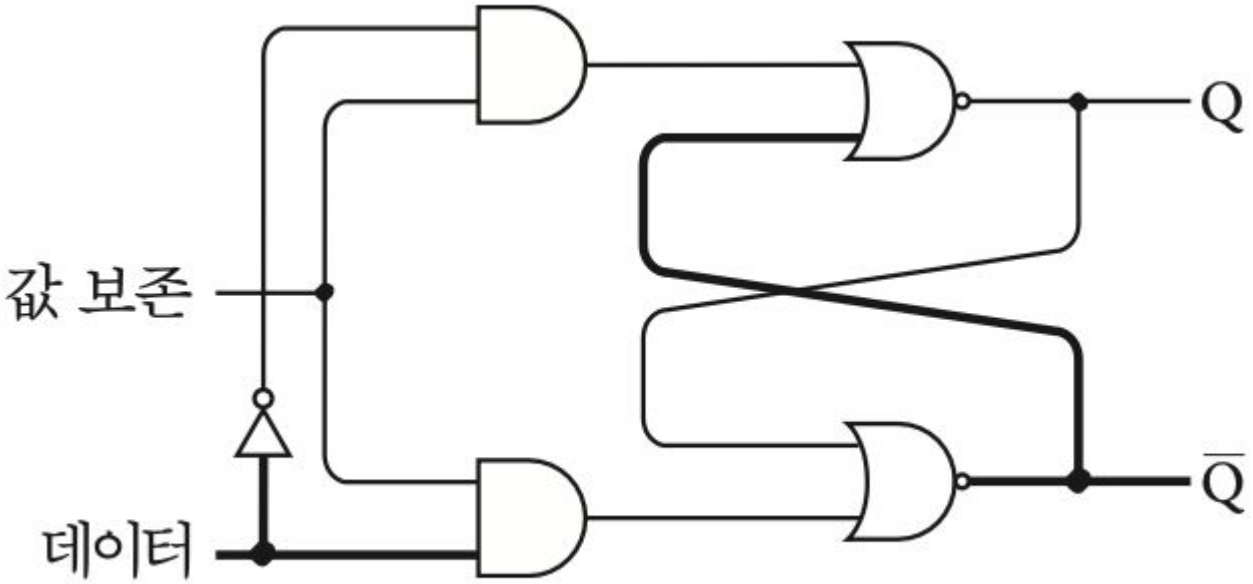
Hold (보존)



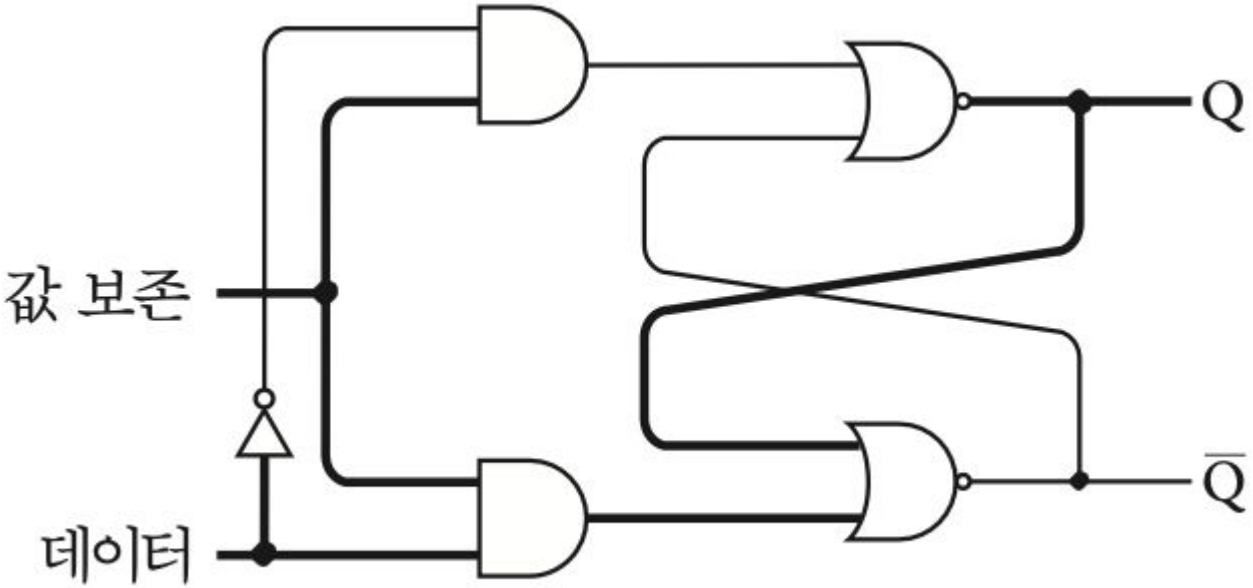
Reset/Set → Data (데이터)



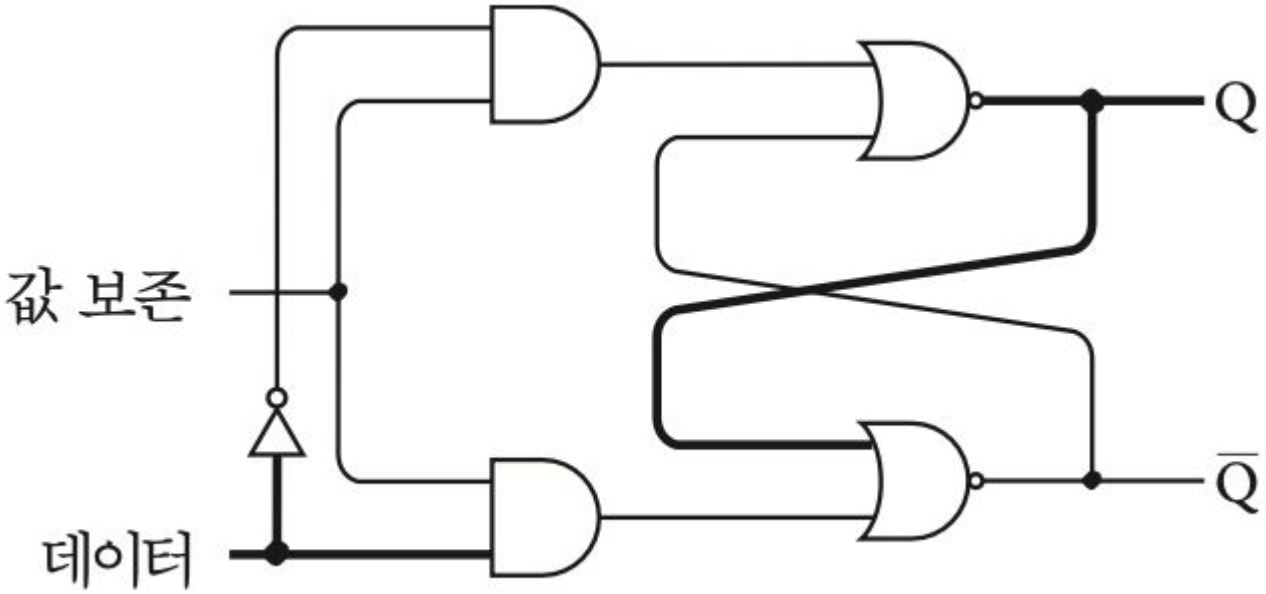
Reset/Set → Data (데이터)



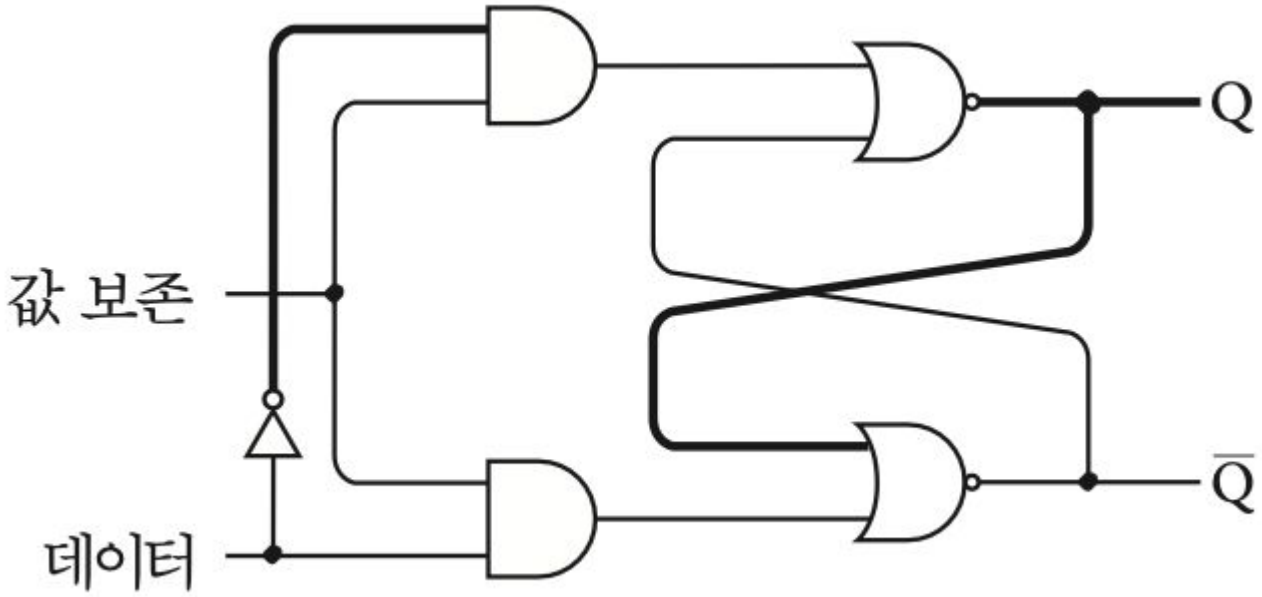
Reset/Set → Data (데이터)



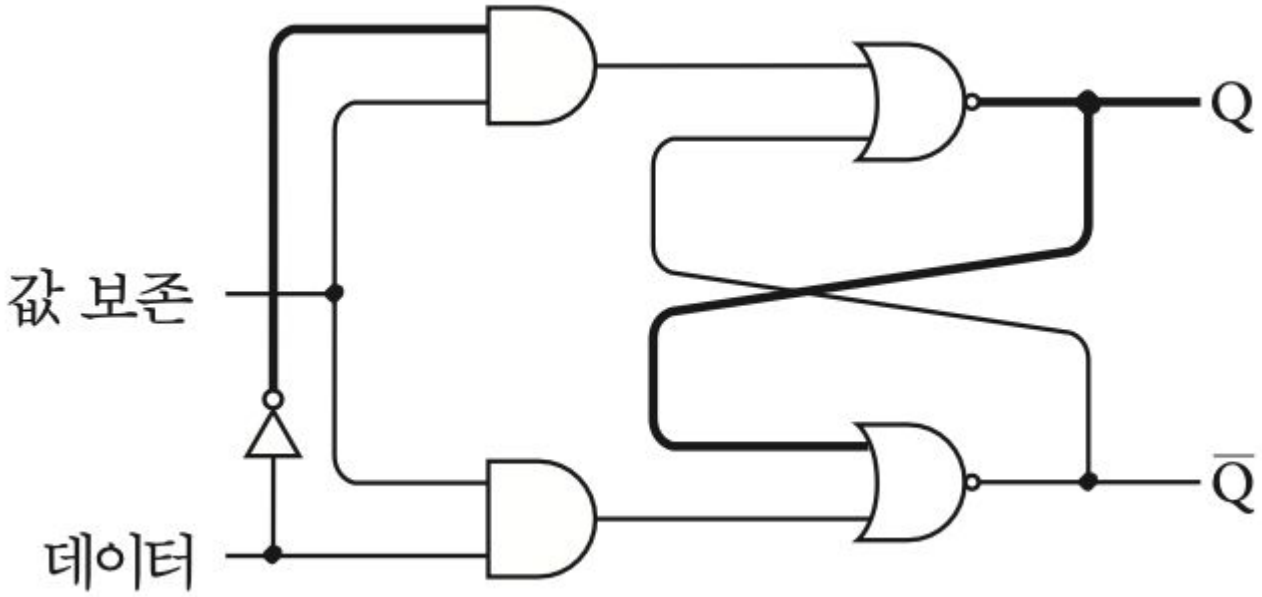
Reset/Set → Data (데이터)



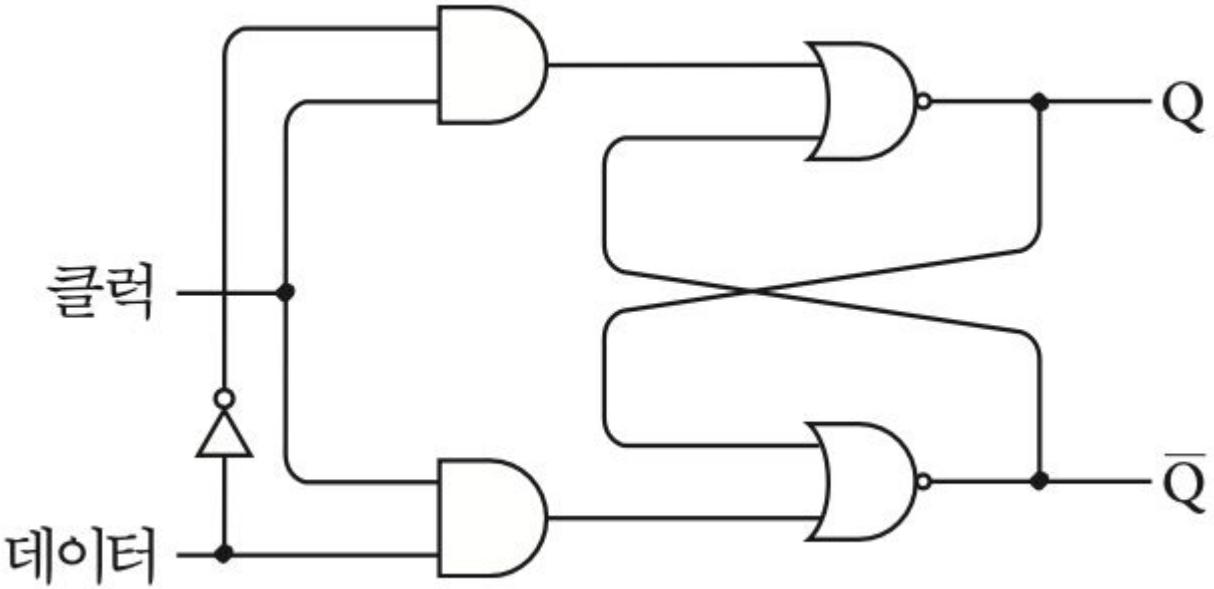
Reset/Set → Data (데이터)



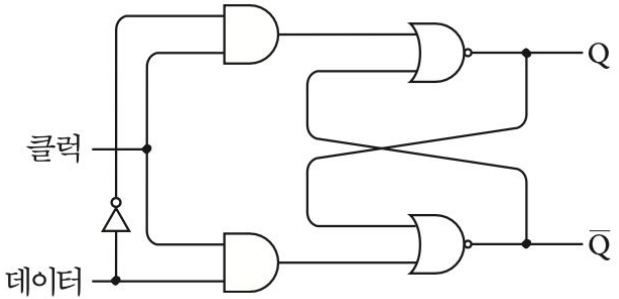
Level-triggered D-type Flip-Flop



Level-triggered D-type Latch



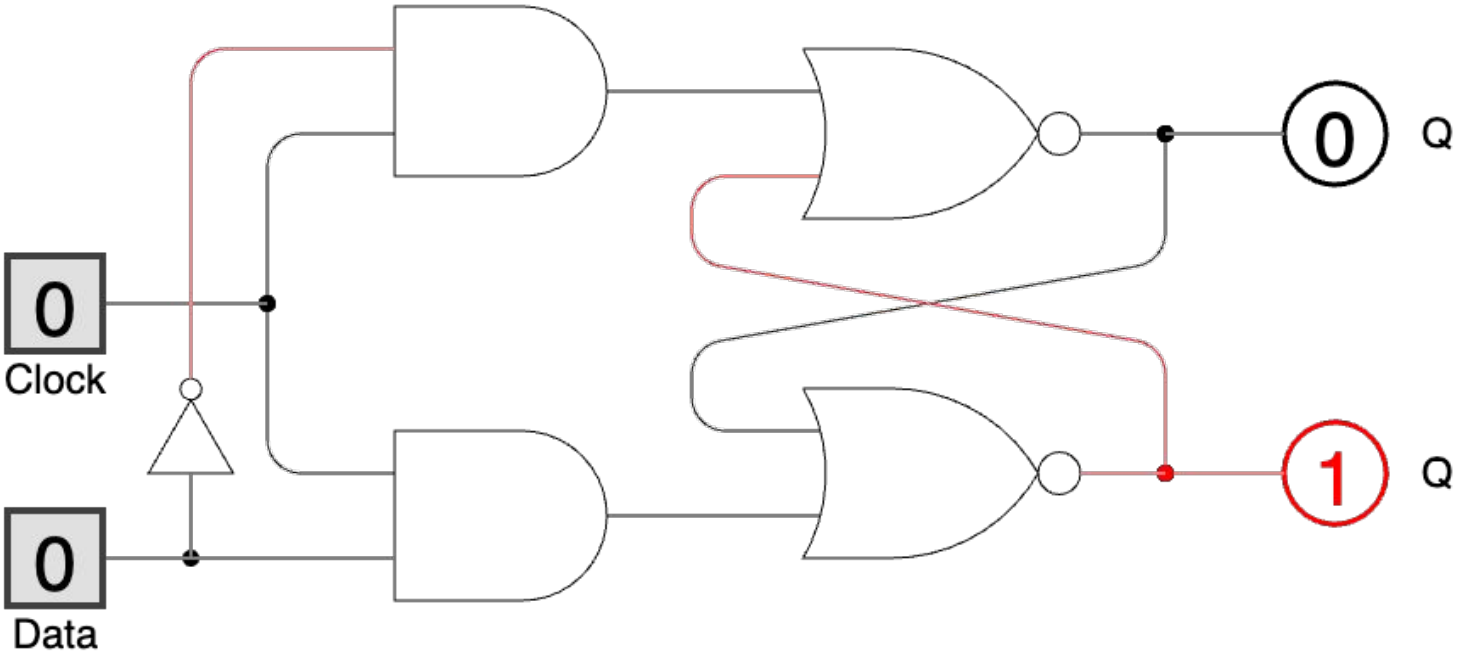
Level-triggered D-type Latch



입력		출력	
D	Clk	Q	Q-bar
0	1	0	1
1	1	1	0
X	0	Q	Q-bar

1-bit 메모리

Level-triggered D-type Latch: 예제

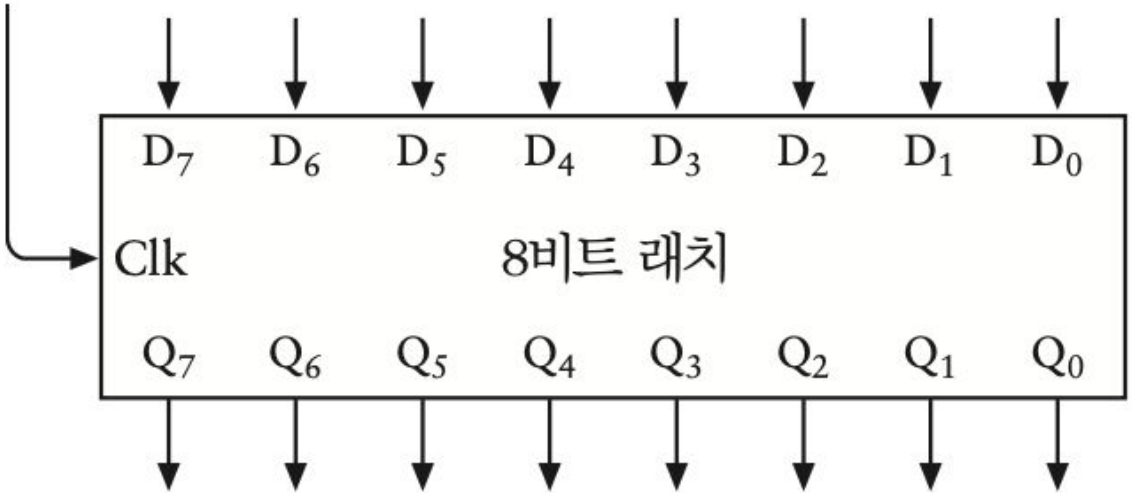


https://www.codehiddenlanguage.com/Chapter17/

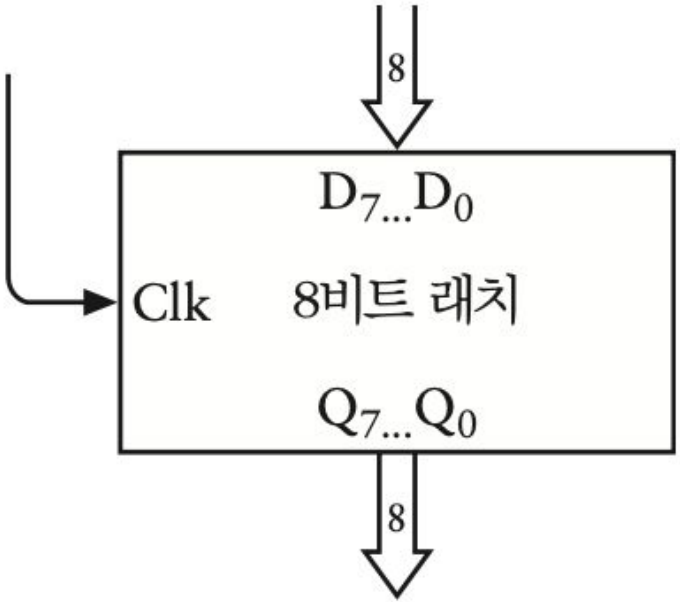
가산기: 누적 (accumulation)

여러 개의 래치를 이용해서 다수의 비트를 저장할 수 있도록 하는 것은 여러모로 유용하게 사용할 수 있습니다. 12장에서 본 덧셈기를 이용해서 세 개의 8비트 숫자를 서로 더하려고 한다고 가정해 봅시다. 기존에 해왔던 것처럼 첫 번째 스위치들을 이용하여 첫 번째 숫자를 입력하고 두 번째 스위치들을 이용해서 두 번째 숫자를 입력하면 되지만, 그 결과는 어딘가 적어 두어야만 합니다. 그 후에 좀 전의 결과를 스위치로 입력하고, 세 번째 숫자를 또 다른 스위치를 이용하여 입력하면 됩니다. 하지만 중간 결과를 다시 입력할 필요가 없이 첫 번째 연산의 결과를 직접 이용하는 방법이 있을 것 같습니다.

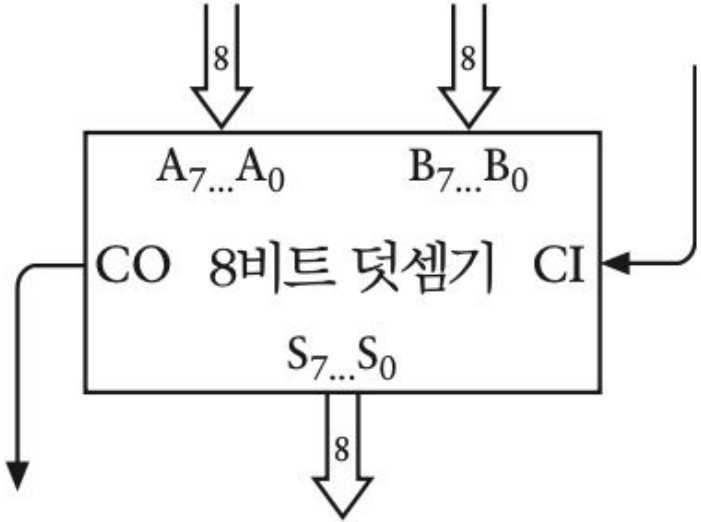
8비트 래치 (8-bit Latch)



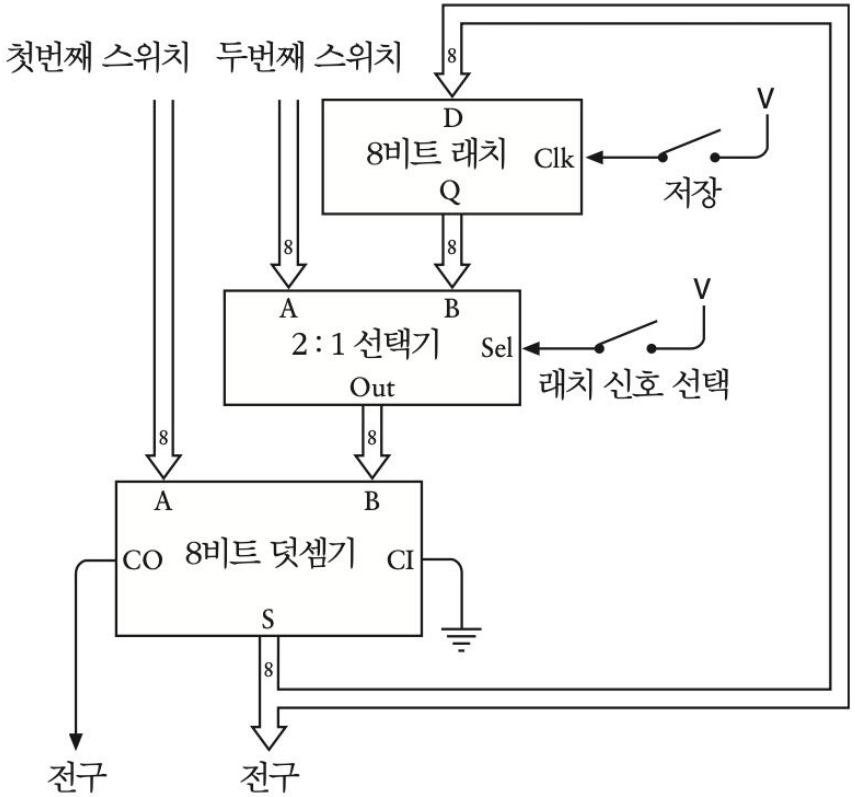
8비트 래치 (8-bit Latch)



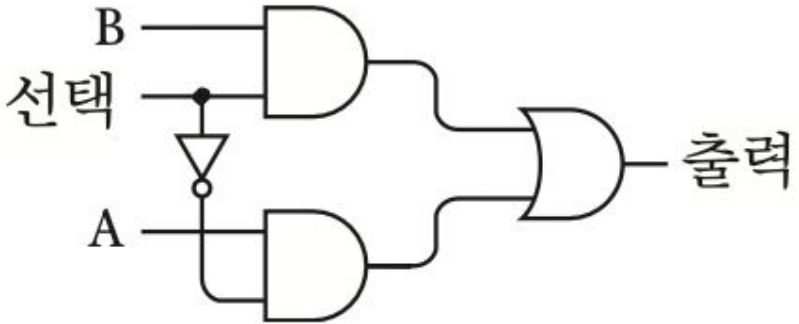
8비트 가산기 (8-bit Adder)



8비트 가산기 + 래치


첫번째 스위치
두번째 스위치
8
D
8비트 래치
Clk
Q
V
저장
8
8
A
B
2 : 1 선택기
Sel
Out
V
래치 신호 선택
8
8
A
B
CO
8비트 덧셈기
CI
S
8
전구
전구

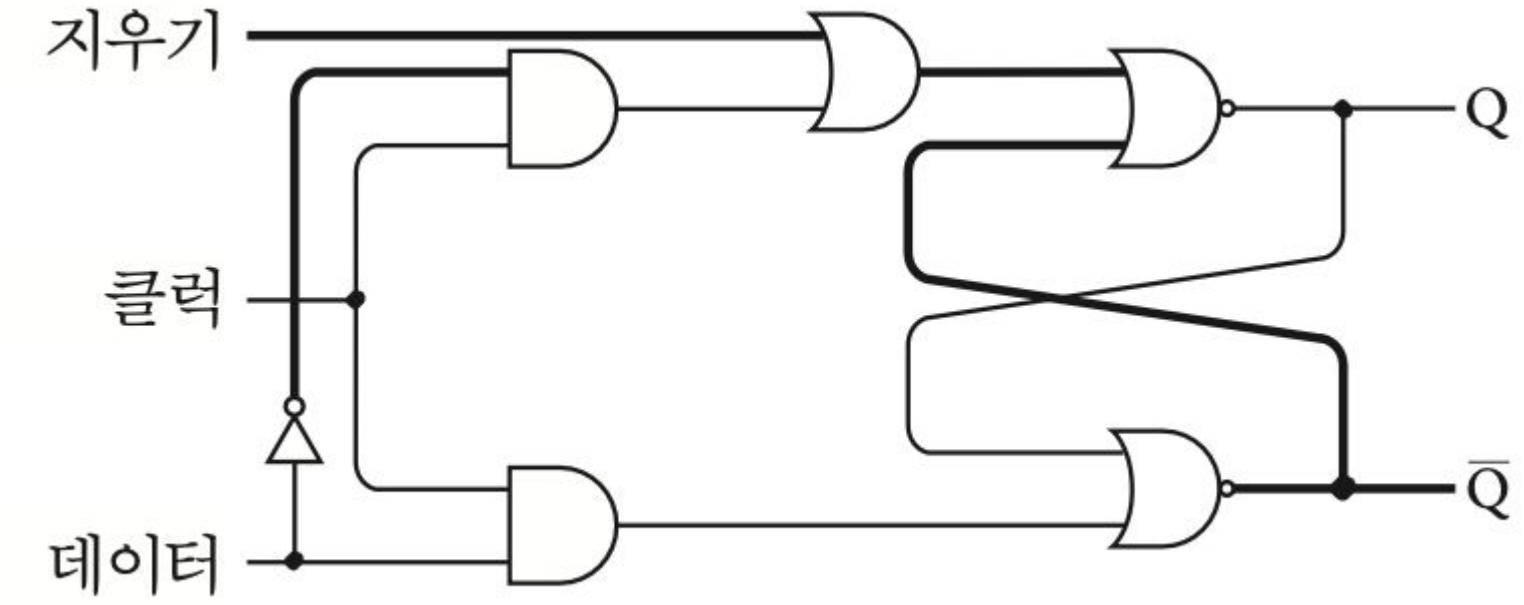
2:1 선택기 (2-Line-to-1-Line Selector)



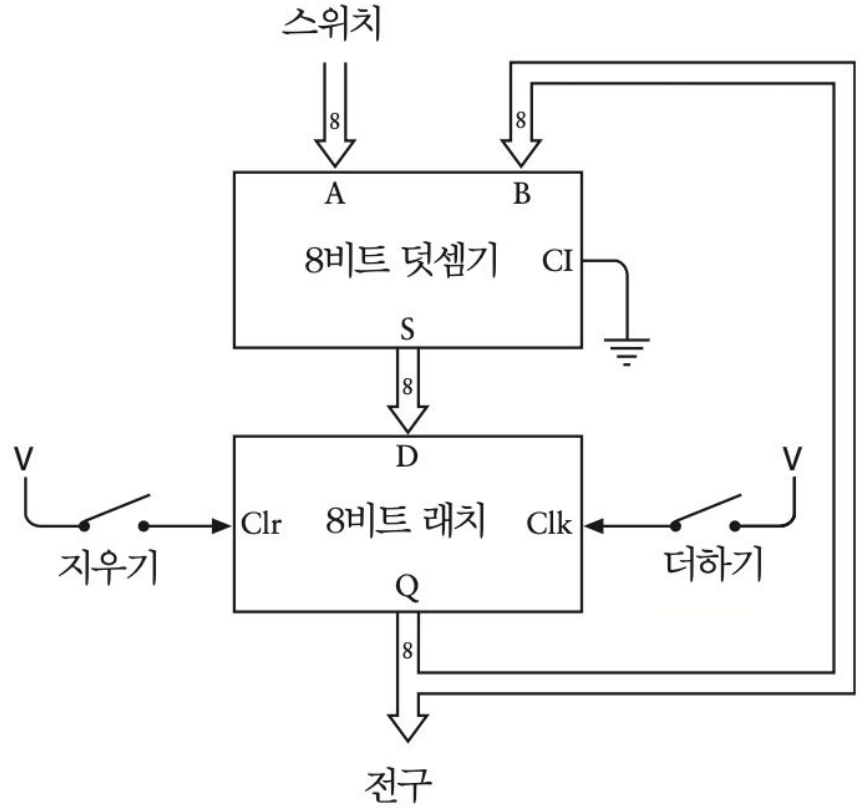
입력			출력
Select	A	B	Q
0	0	X	0
0	1	X	1
1	X	0	0
1	X	1	1

2:1 Multiplexer (MUX)

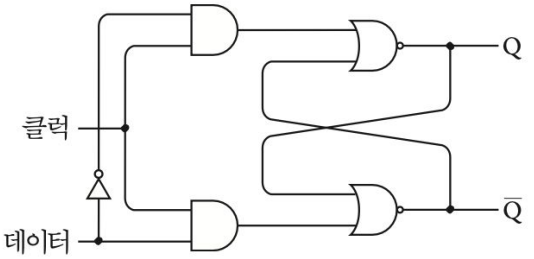
지우기 (clear)

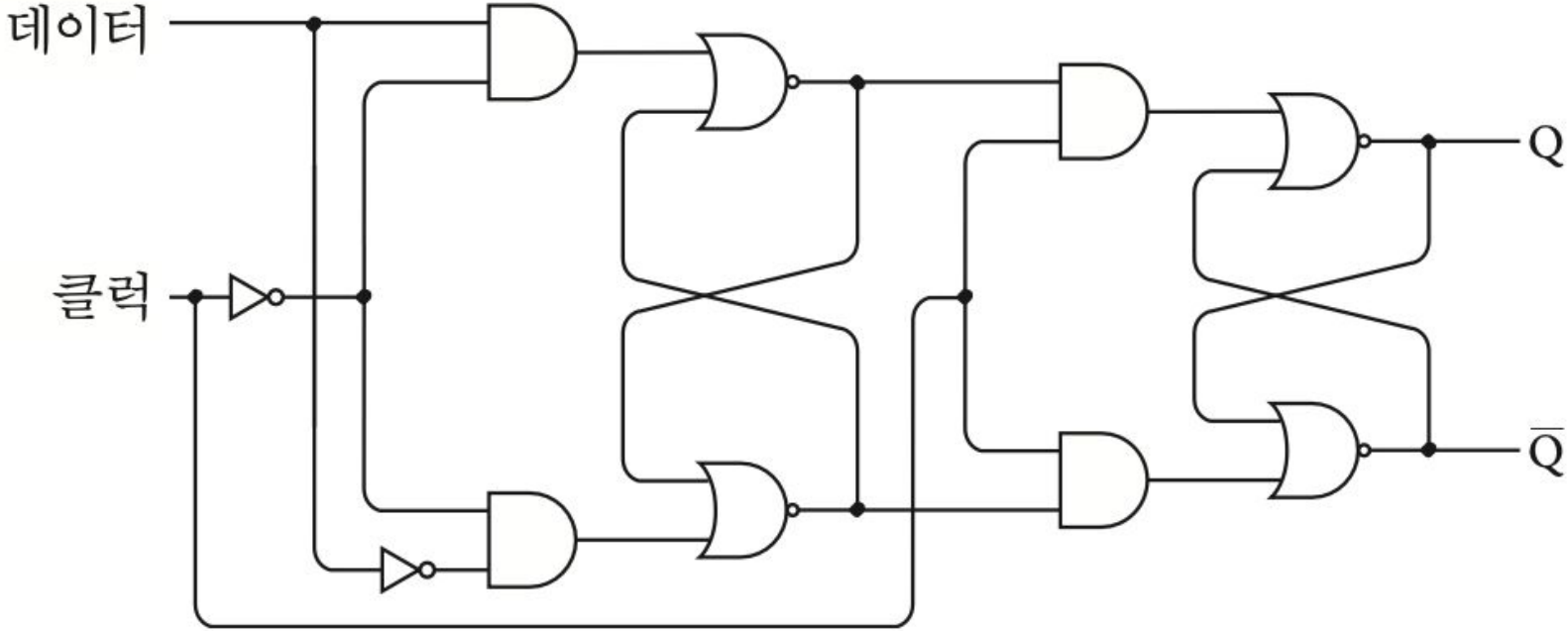


지우기 (clear)

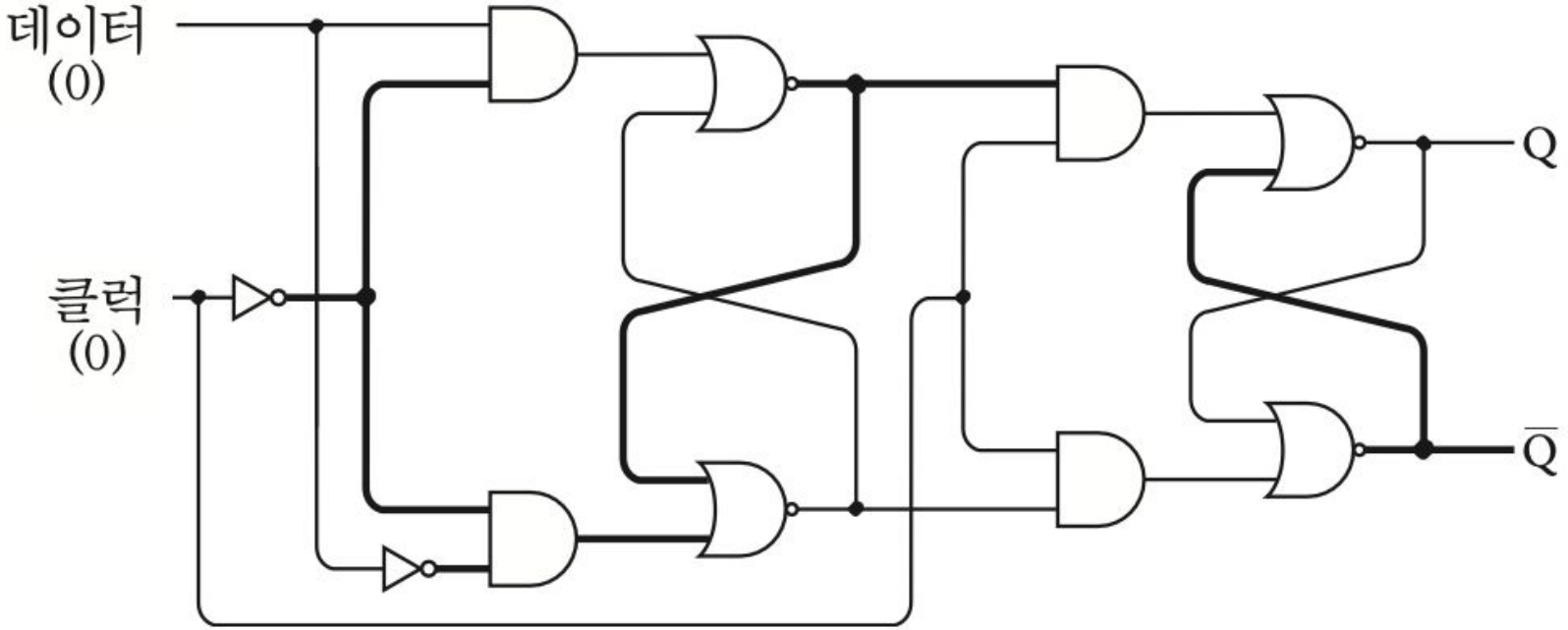


Edge-triggered D-type Flip-Flop

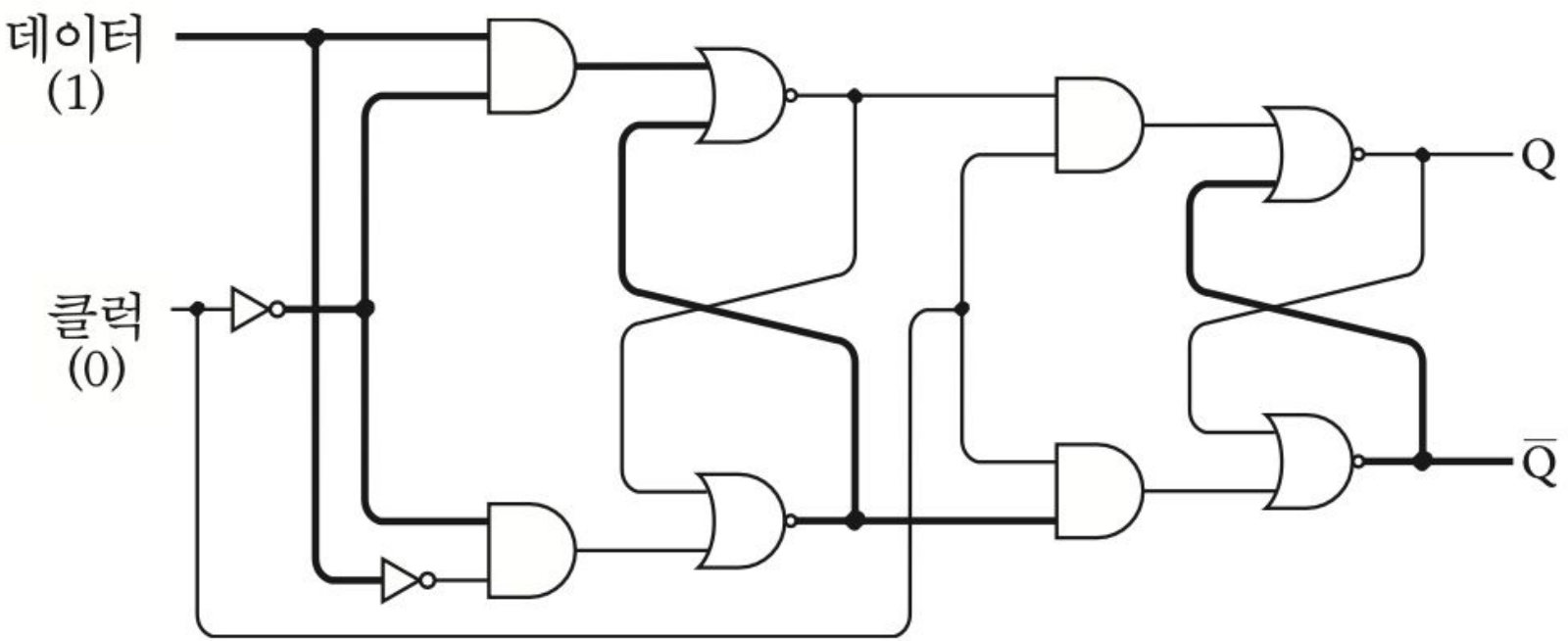




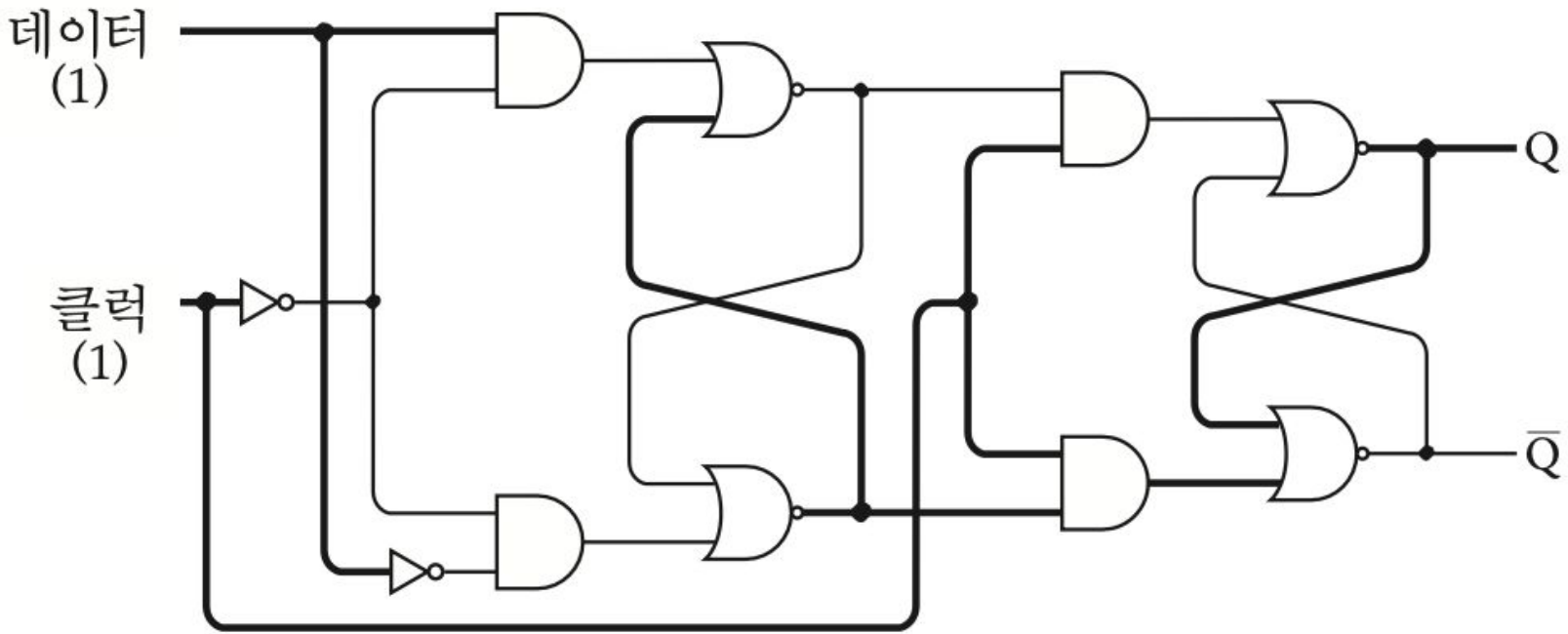
Edge-triggered D-type Flip-Flop



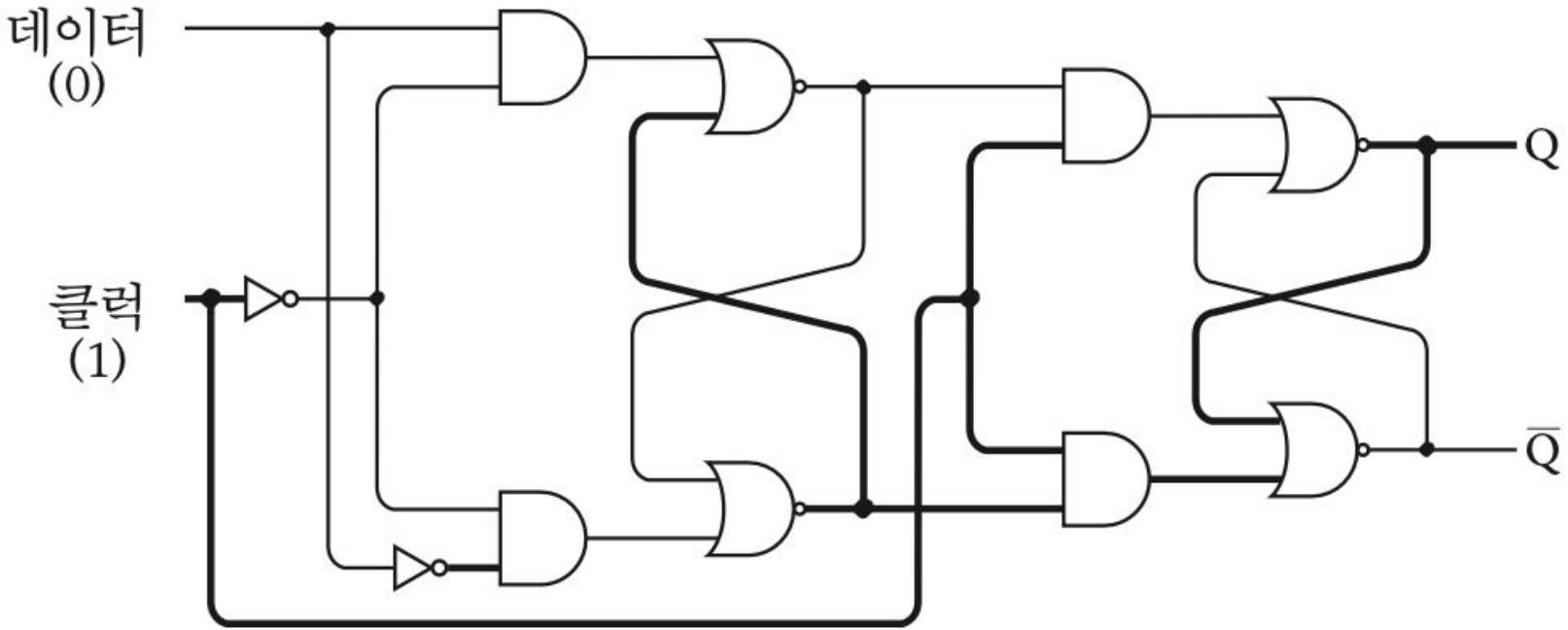
Edge-triggered D-type Flip-Flop



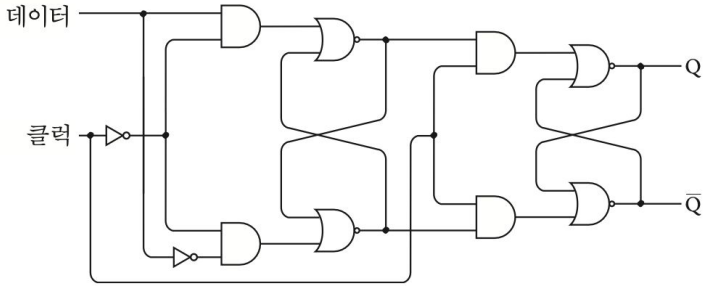
Edge-triggered D-type Flip-Flop



Edge-triggered D-type Flip-Flop

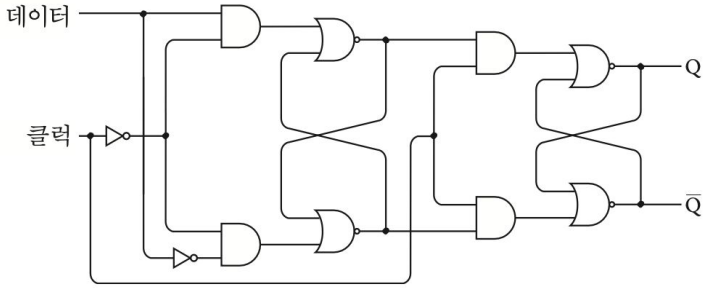


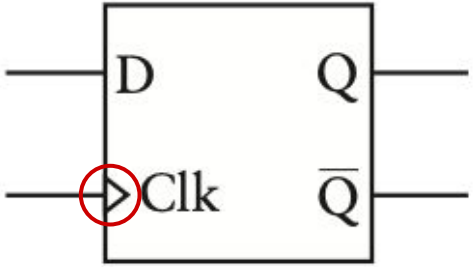
Edge-triggered D-type Flip-Flop



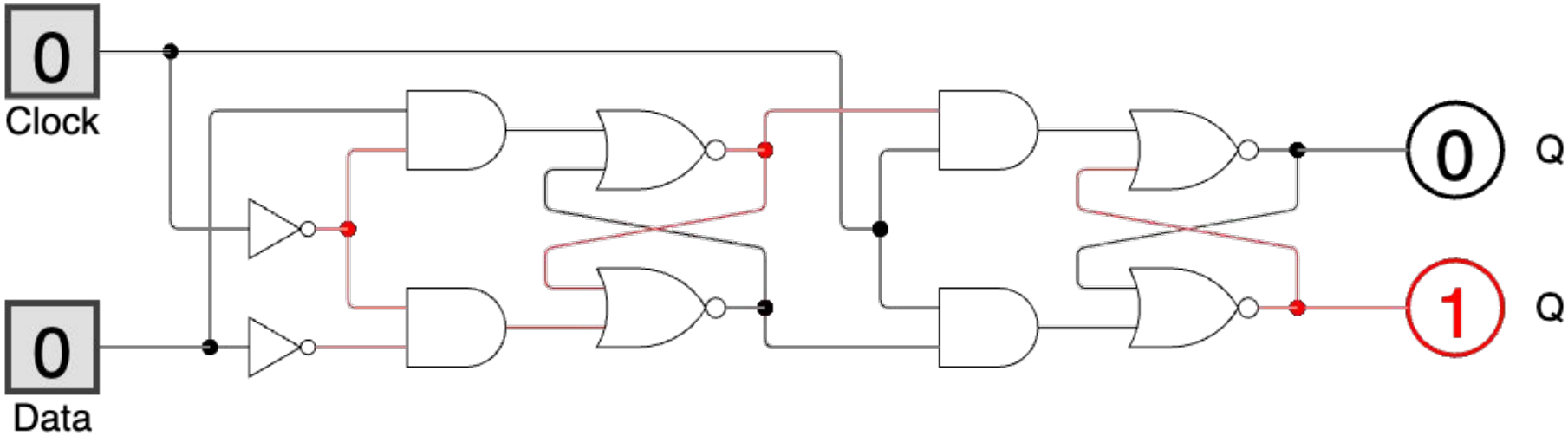
입력		출력	
D	Clk	Q	$\overline{Q}$
0	↑	0	1
1	↑	1	0
X	0	Q	$\overline{Q}$

Edge-triggered D-type Flip-Flop



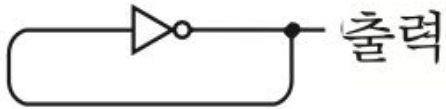


Edge-triggered D-type Flip-Flop: 예제

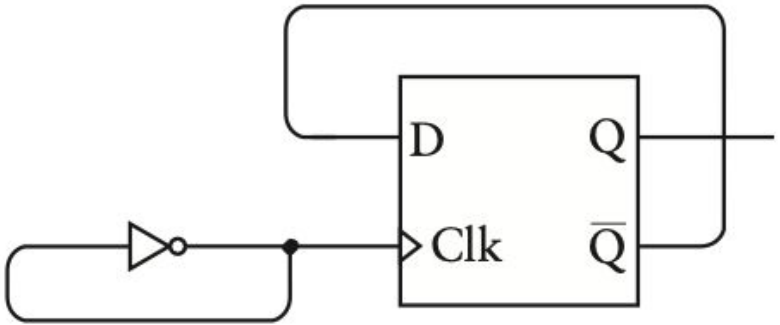


https://www.codehiddenlanguage.com/Chapter17/

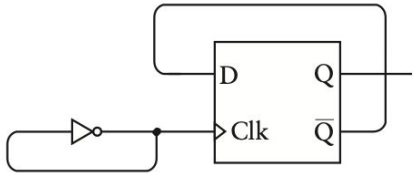
오실레이터



오실레이터 + 엣지 트리거 D 타입 플립플롭

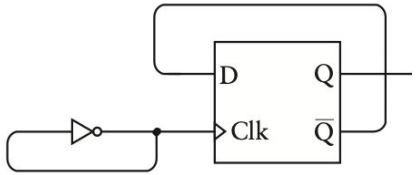


오실레이터 + 엣지 트리거 D 타입 플립플롭



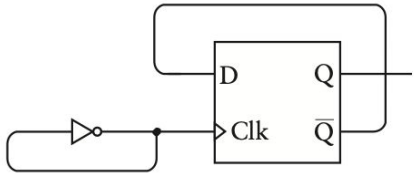
입력		출력	
D	Clk	Q	$\overline{Q}$
1	0	0	1

오실레이터 + 엣지 트리거 D 타입 플립플롭



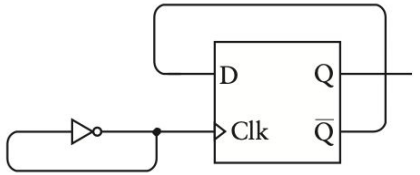
입력		출력	
D	Clk	Q	$\overline{Q}$
1	0	0	1
1	↑	1	0

오실레이터 + 엣지 트리거 D 타입 플립플롭



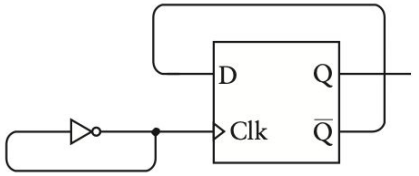
입력		출력	
D	Clk	Q	$\overline{Q}$
1	0	0	1
1	↑	1	0
0	1	1	0

오실레이터 + 엣지 트리거 D 타입 플립플롭



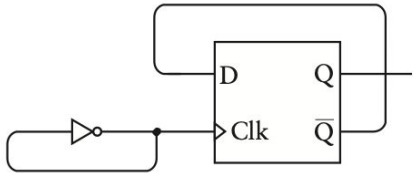
입력		출력	
D	Clk	Q	$\overline{Q}$
1	0	0	1
1	↑	1	0
0	1	1	0
0	0	1	0

오실레이터 + 엣지 트리거 D 타입 플립플롭



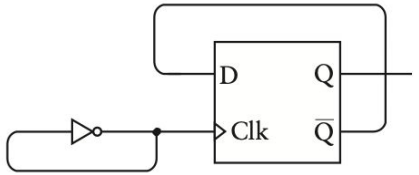
입력		출력	
D	Clk	Q	$\overline{Q}$
1	0	0	1
1	↑	1	0
0	1	1	0
0	0	1	0
0	↑	0	1

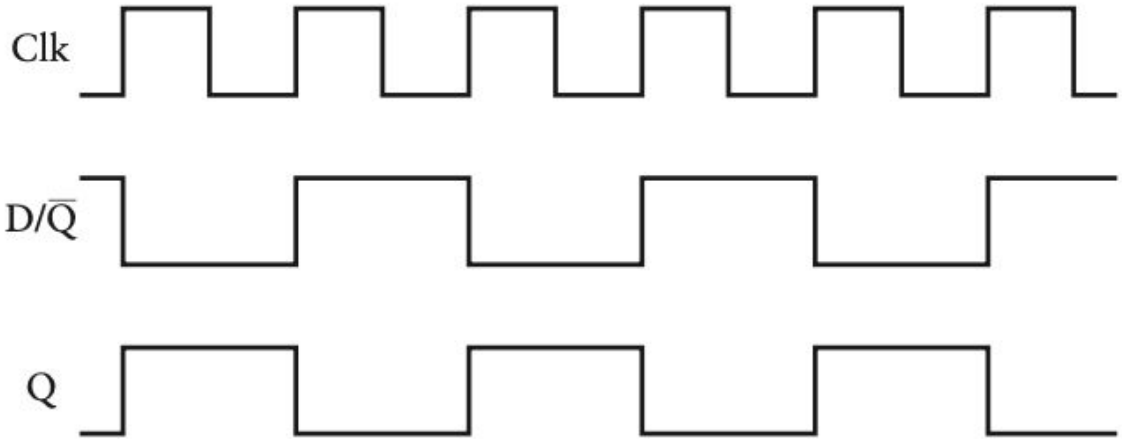
오실레이터 + 엣지 트리거 D 타입 플립플롭



입력		출력	
D	Clk	Q	$\overline{Q}$
1	0	0	1
1	↑	1	0
0	1	1	0
0	0	1	0
0	↑	0	1
1	1	0	1

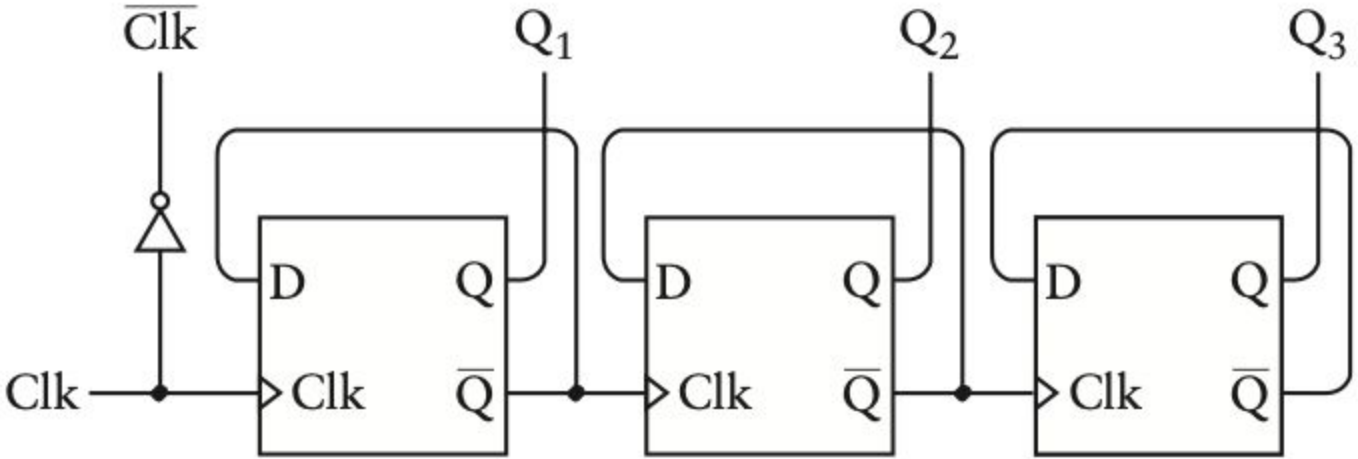
타이밍 다이어그램 (timing diagram)



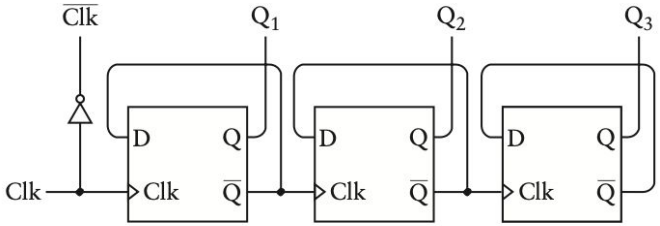


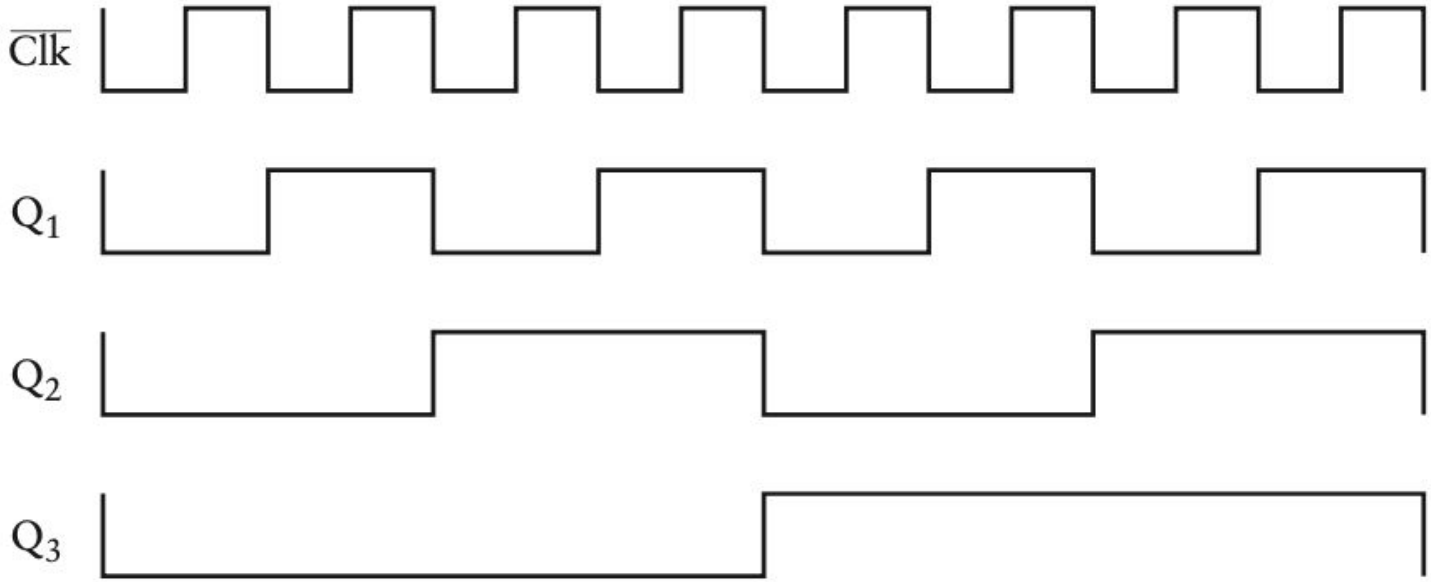
주파수 분주기 (frequency divider)

클럭 분주기 x3

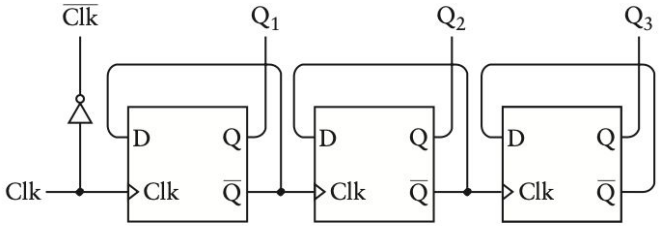

$\overline{Clk}$
Q_1
Q_2
Q_3
D
Q
Clk
$\overline{Q}$
D
Q
Clk
$\overline{Q}$
D
Q
Clk
$\overline{Q}$
Clk

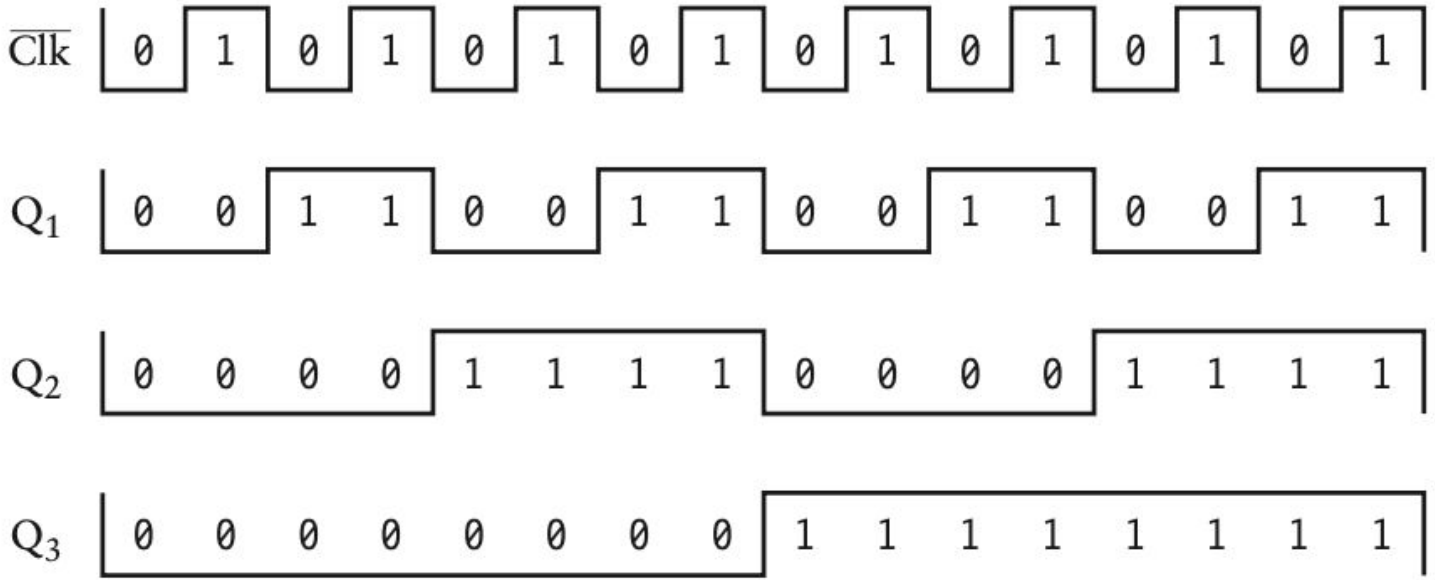
클럭 분주기 x3


$\overline{Clk}$
Q_1
Q_2
Q_3
Clk
D
Q
Clk
$\overline{Q}$
D
Q
Clk
$\overline{Q}$
D
Q
Clk
$\overline{Q}$

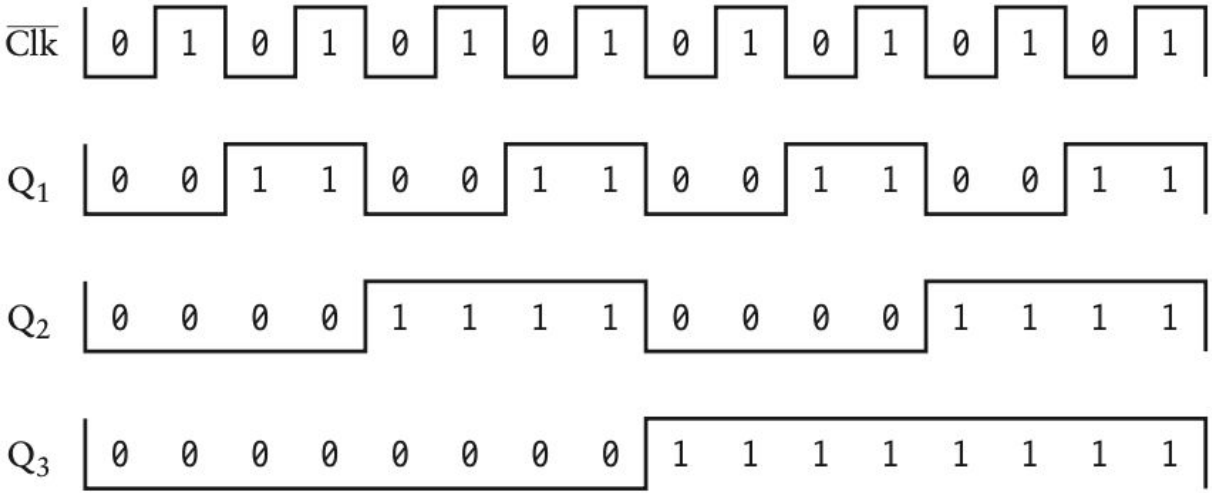

$\overline{Clk}$
Q_1
Q_2
Q_3

클럭 분주기 x3


$\overline{Clk}$
Q_1
Q_2
Q_3
D
Q
Clk
$\overline{Q}$

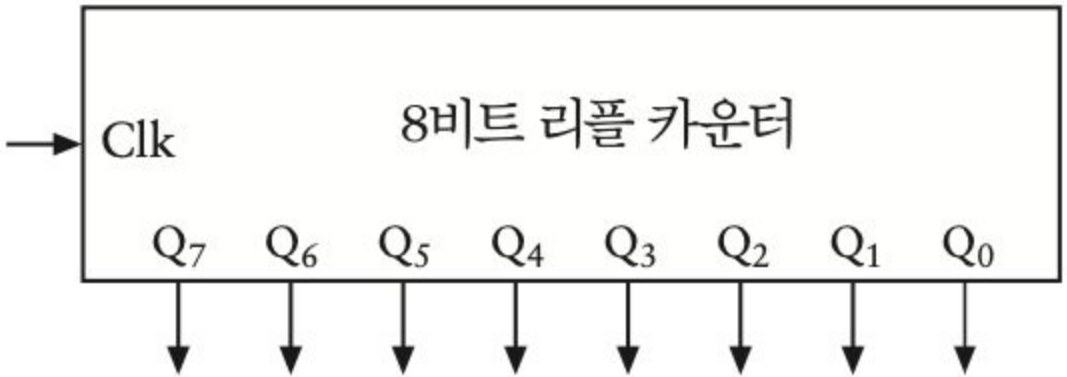

$\overline{Clk}$ 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
Q_1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1
Q_2 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1
Q_3 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1

계수기 (counter)

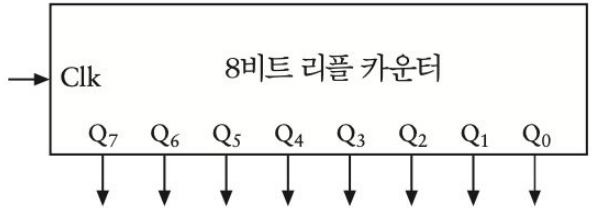


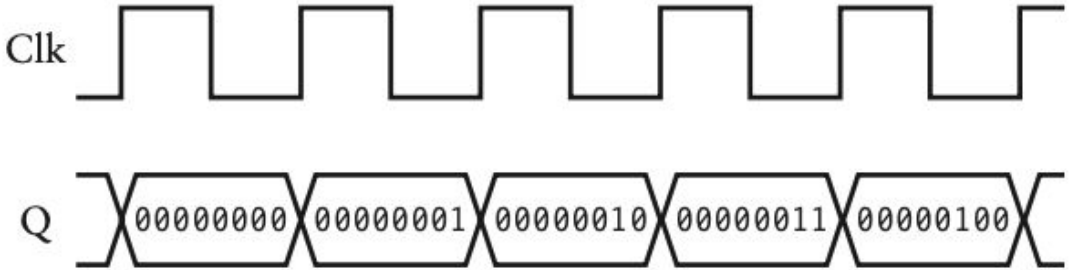
이진수	십진수
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14
1111	15

클럭 분주기 x8

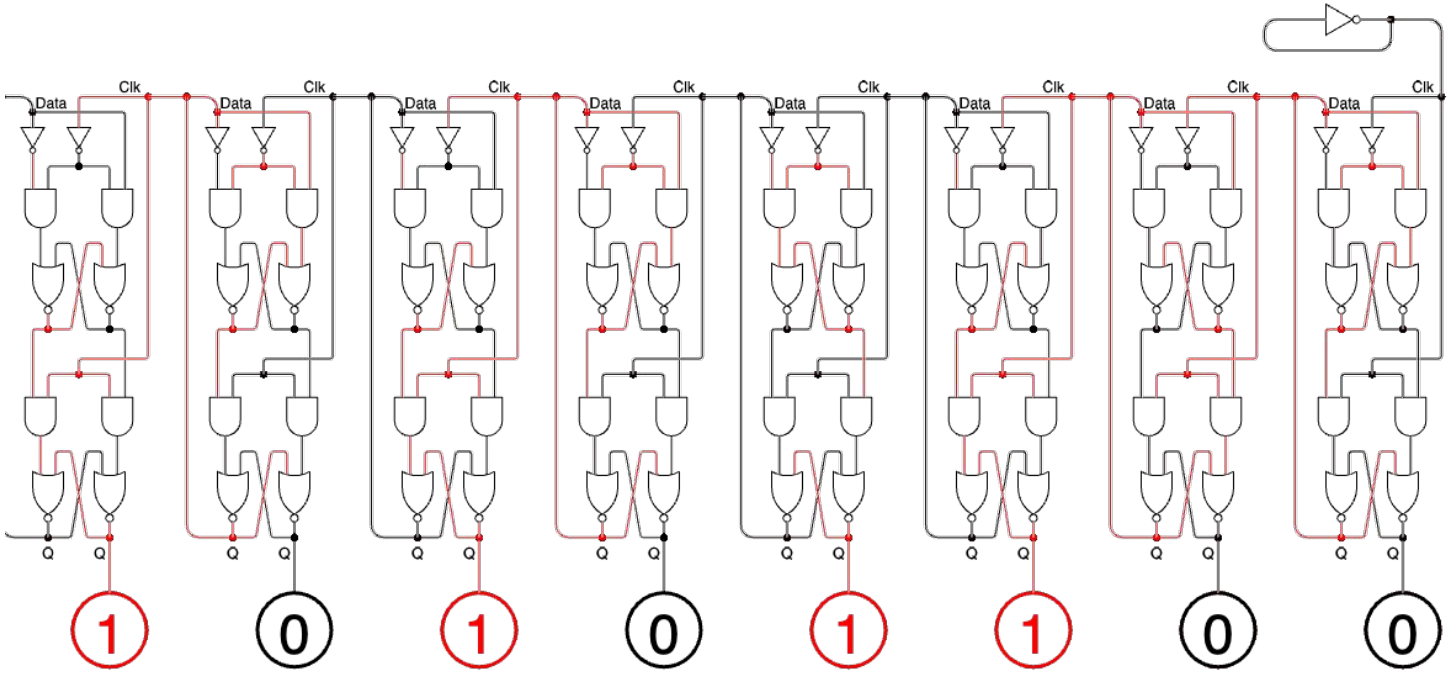


리플 카운터 (ripple counter)





Ripple Counter: 예제



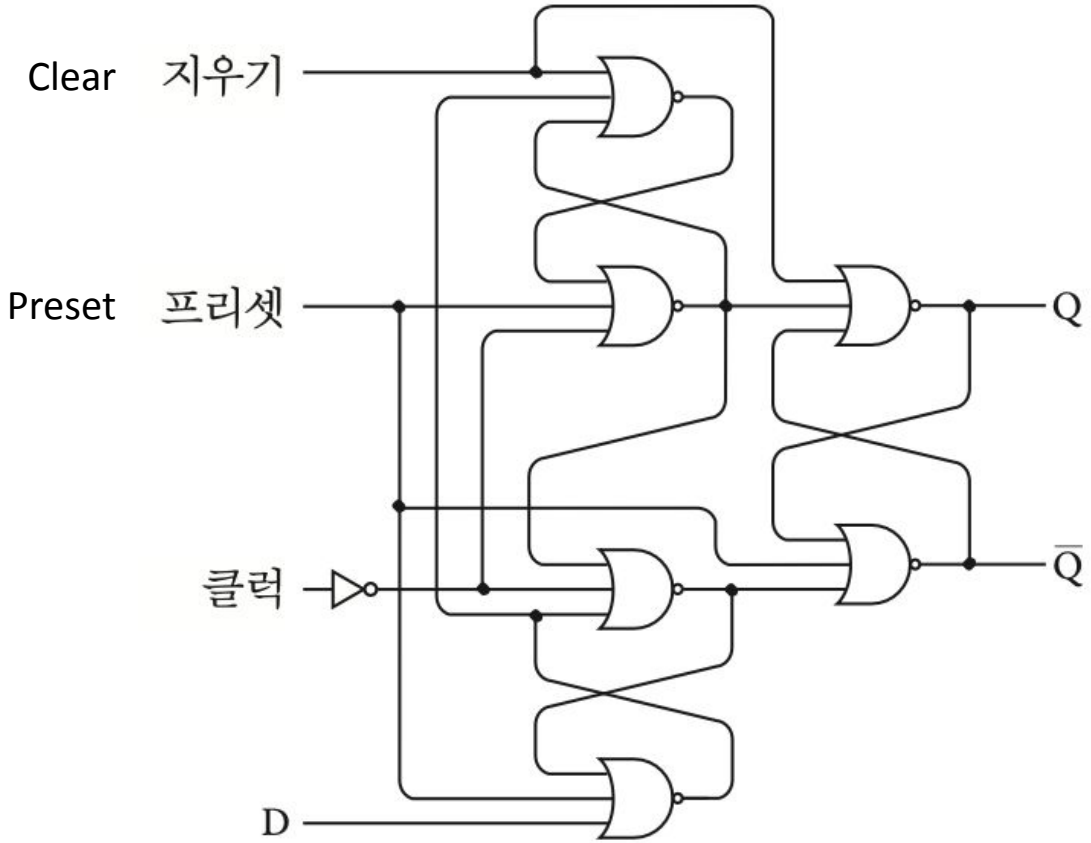
ACh = 172

https://www.codehiddenlanguage.com/Chapter17/

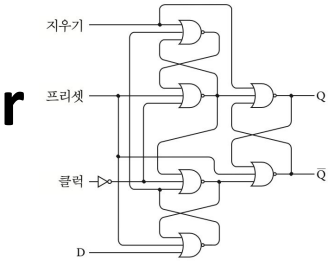
오실레이터의 주파수

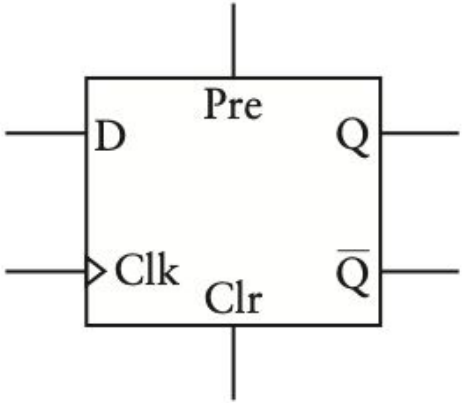
앞부분에서 오실레이터의 주파수를 알아낼 수 있는 방법을 알려드리겠다고 했습니다. 지금부터 알려드리도록 하지요. 만일 오실레이터가 8비트 카운터의 입력에 연결되어 있다면 카운터는 오실레이터에서 얼마나 많은 사이클이 발생했는지 알려줄 수 있습니다. 또한 카운터가 11111111(십진수로 255)에 도달하게 되면 그 값은 00000000으로 돌아가게 됩니다. 오실레이터의 주파수를 알아내기 위하여 이 카운터를 이용할 때 가장 쉬운 방법은 8비트 카운터의 출력을 전구와 연결하는 것입니다. 이제 출력이 0이 될 때(모든 전구가 꺼져 있는 경우입니다)까지 기다렸다가, 0이 되면 초시계를 켭니다. 이후에 모든 전구가 다시 꺼질 때 초시계를 끄면 됩니다. 이제 오실레이터에서 256사이클을 발생시키는 데 걸리는 시간을 찾아냈습니다. 예를 들어, 10초가 걸렸다고 하면, 오실레이터의 주파수는 256÷10이 되어 25.6Hz가 됩니다.

Edge-triggered D-type Flip-Flop with Preset and Clear

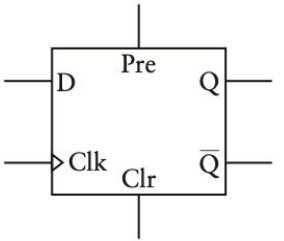


Edge-triggered D-type Flip-Flop with Preset and Clear



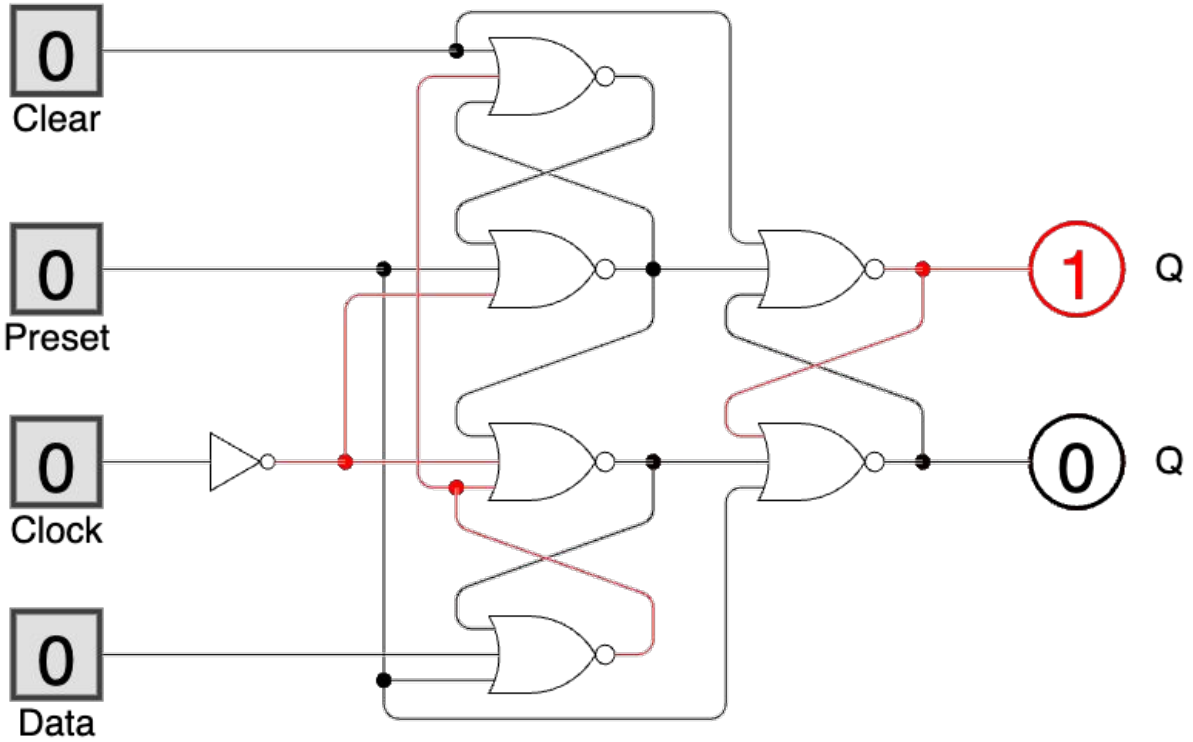


Edge-triggered D-type Flip-Flop with Preset and Clear



입력				출력	
Pre	Clr	D	Clk	Q	$\overline{Q}$
1	0	X	X	1	0
0	1	X	X	0	1
0	0	0	↑	0	1
0	0	1	↑	1	0
0	0	X	0	Q	$\overline{Q}$

Edge-triggered D-type Flip-Flop with Preset and Clear: 예제



https://www.codehiddenlanguage.com/Chapter17/