

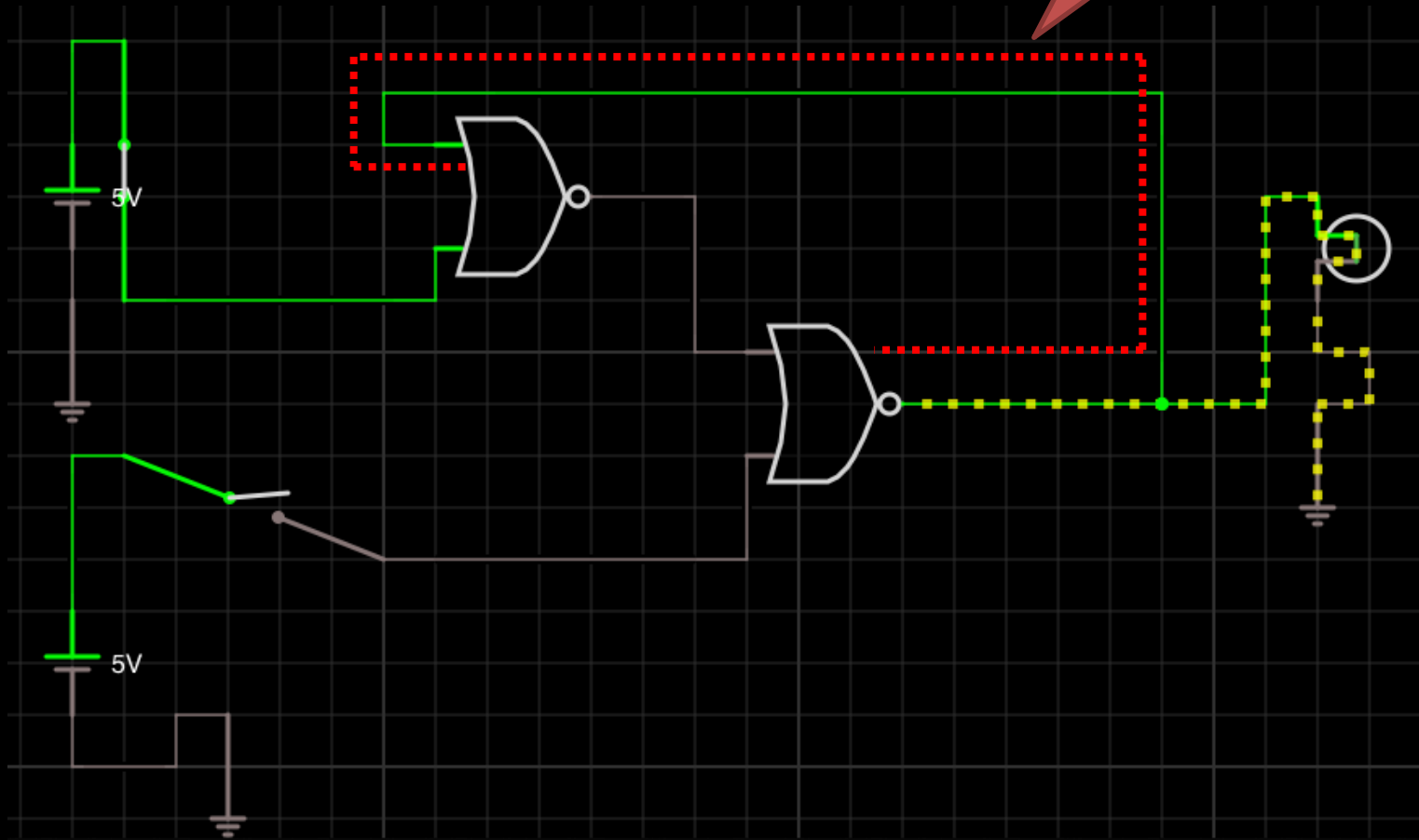
Code : Ch14, 피드백과 플립플롭

---

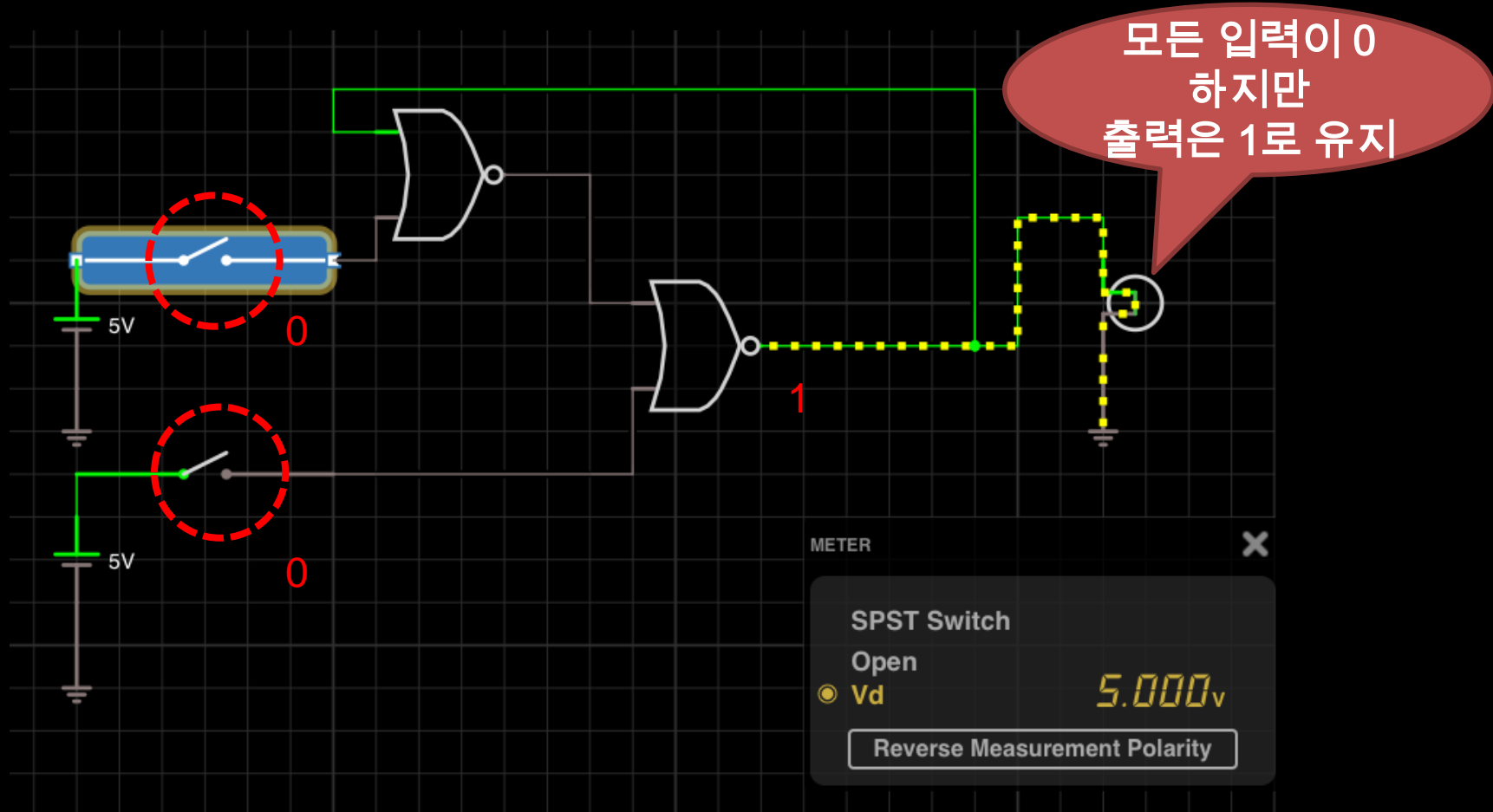
# 용어정리

- 귀환회로 (Feedback)
  - 어떤 게이트의 출력이 다른 게이트의 입력으로 들어가는 경우
- 안정된 상태 ( stable state )
  - 입력이 없는 경우에도 전(前)출력이 유지되는 상태
- 레벨 트리거 ( level trigger )
  - Clock이 완전히 상승된 경우에
- 엣지 트리거 ( edge trigger )
  - Clock이 상승되는 순간
- 래치 ( Latch )
  - 레벨 트리거 속성을 이용하는 논리회로
- 플립플롭 ( Flip flop )
  - 엣지 트리거 속성을 이용하는 논리회로

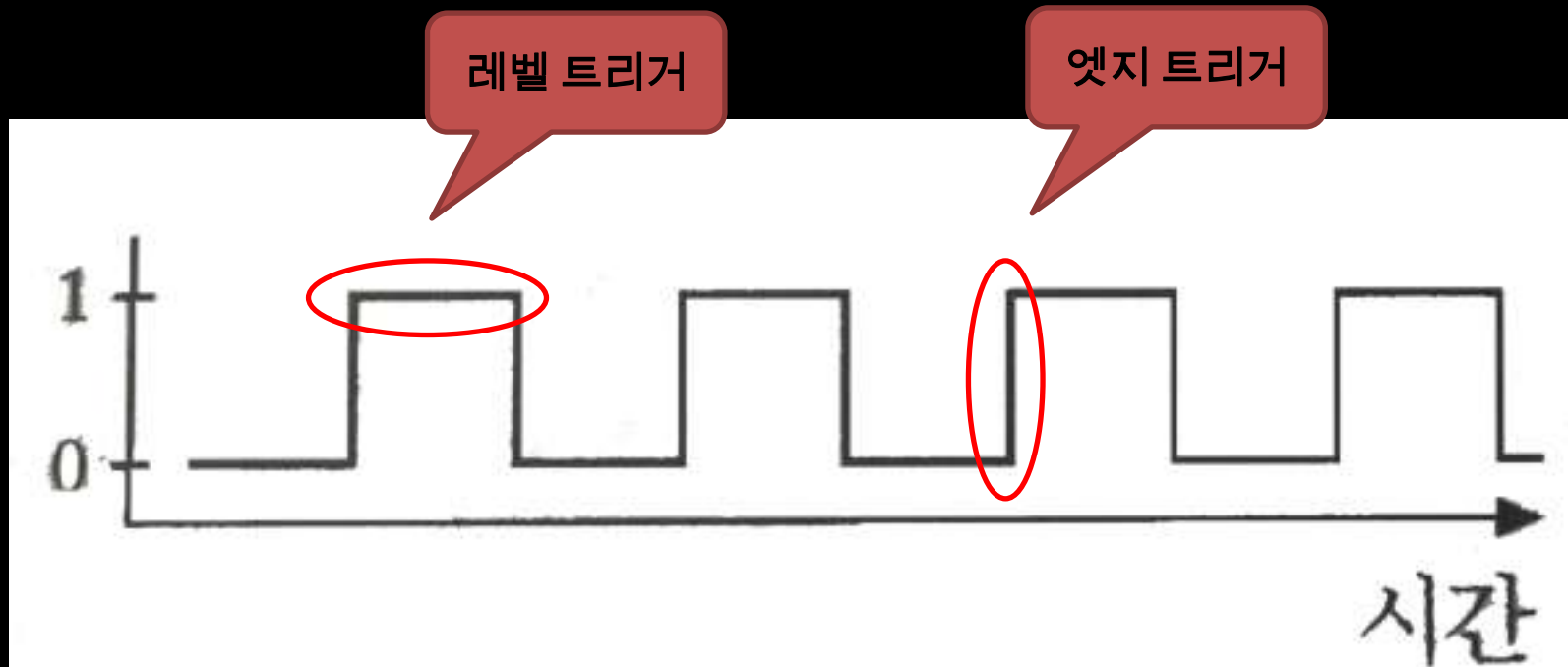
# 귀환회로 (Feedback)



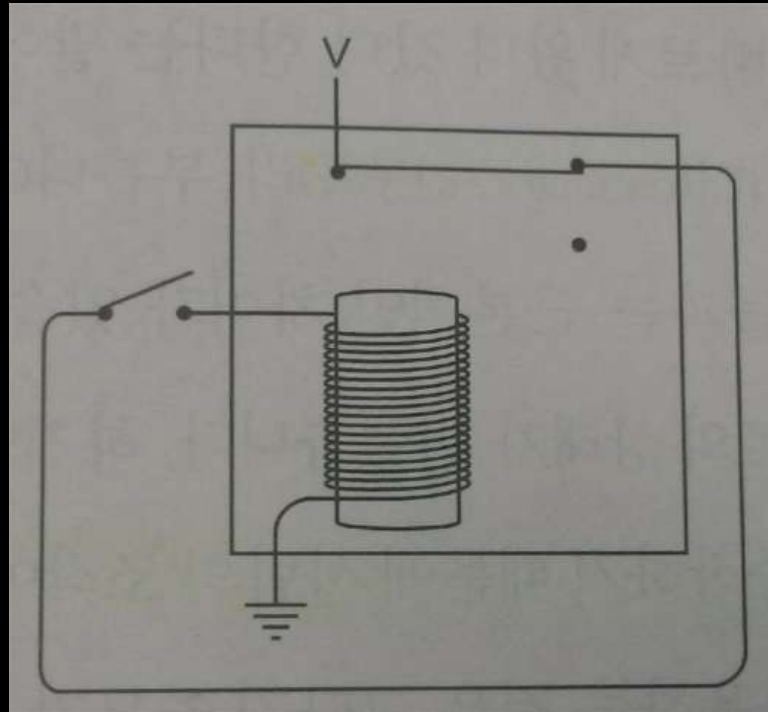
# 안정된 상태 (Stable state)



# 레벨 트리거, 엣지 트리거



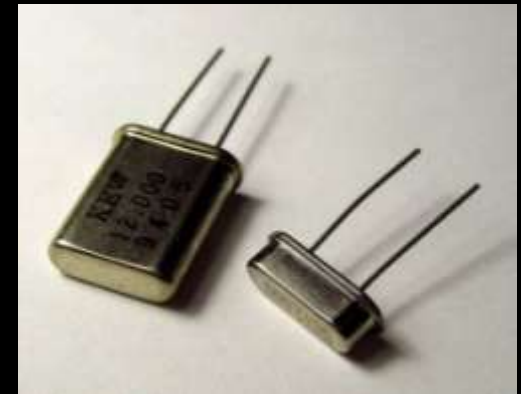
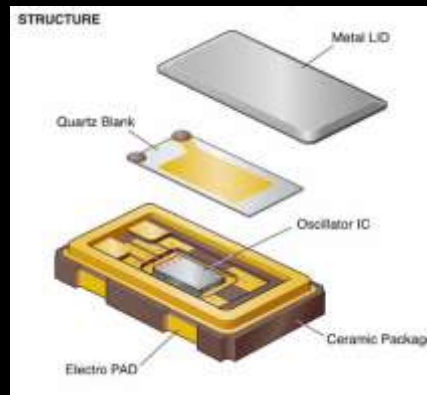
# Buzzer



- Buzzer는 Clock(clk)를 만들 수 있는 인버터의 개념이 됩니다.

# 오실레이터(ocillator)

- 인버터(not gate)를 이용
- 출력이 0과 1사이로 빠르게 바뀜
- 외부개입(스위치) 없이 출력(상태)를 변화
- 동기적으로 동작하는 회로 제작을 가능



# 오실레이터 (oscillator)

SCOPE



t = 262.742

0 I = 0.000

0 Vd = 0.000

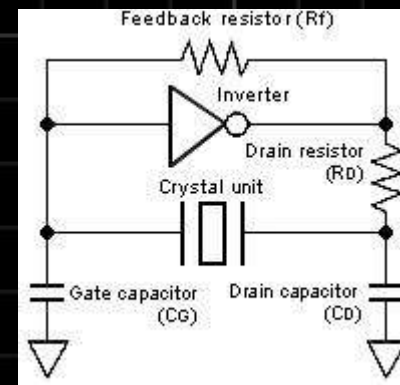
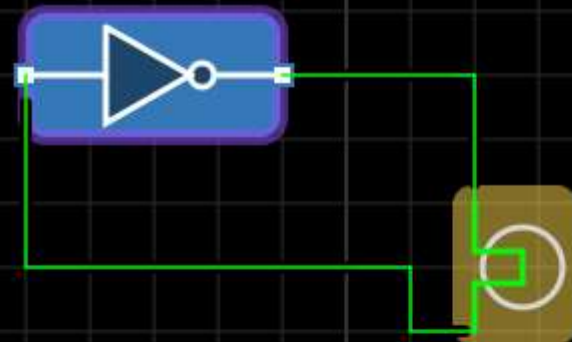
300 T = 300.000

3 Vo = 5.000

RECORD

78.1ms

	RMS	P-P	Freq	Val
I	0.00	0.00	0.00Hz	0.00
Vd	0.00	0.00	0.00Hz	0.00
T	300.00	0.00	2.19Hz	300.00
Vo	3.54	5.00	128.00Hz	0.00



Inverter

○ Vi

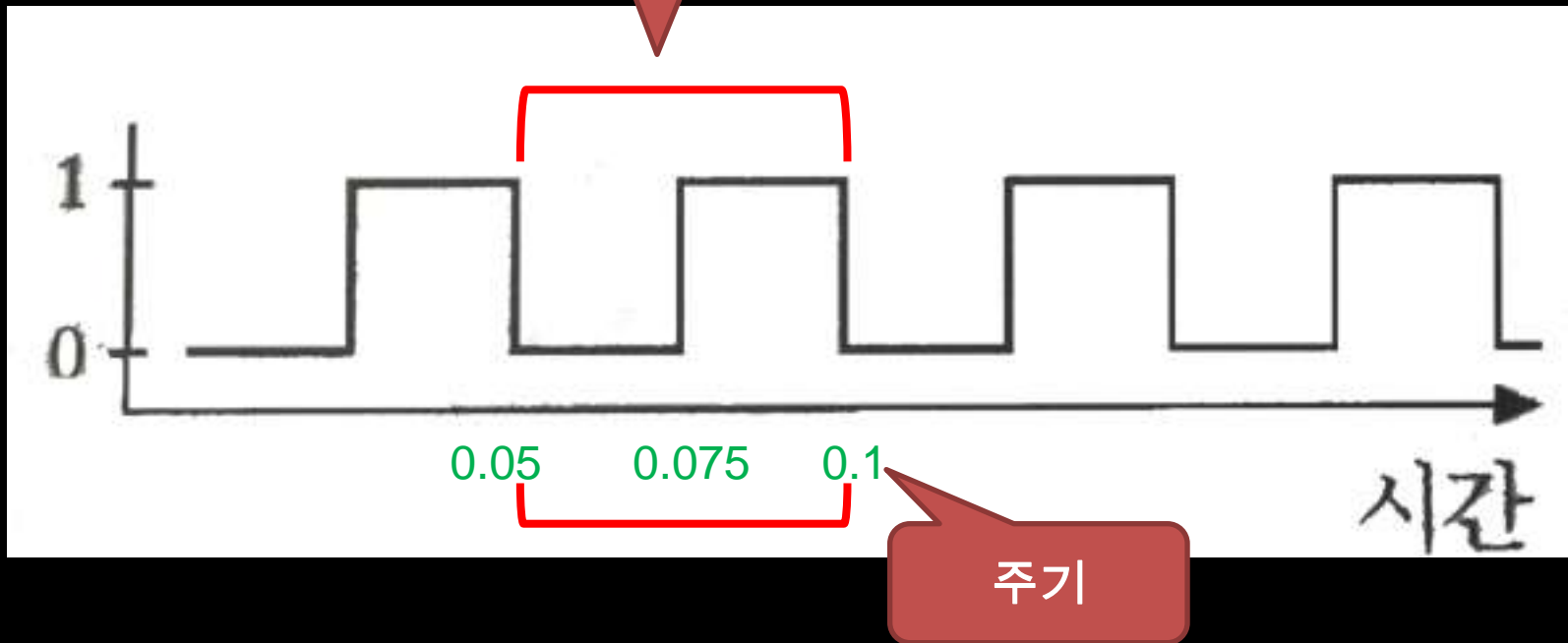
● Vo

$V_v$   
 $V_v$

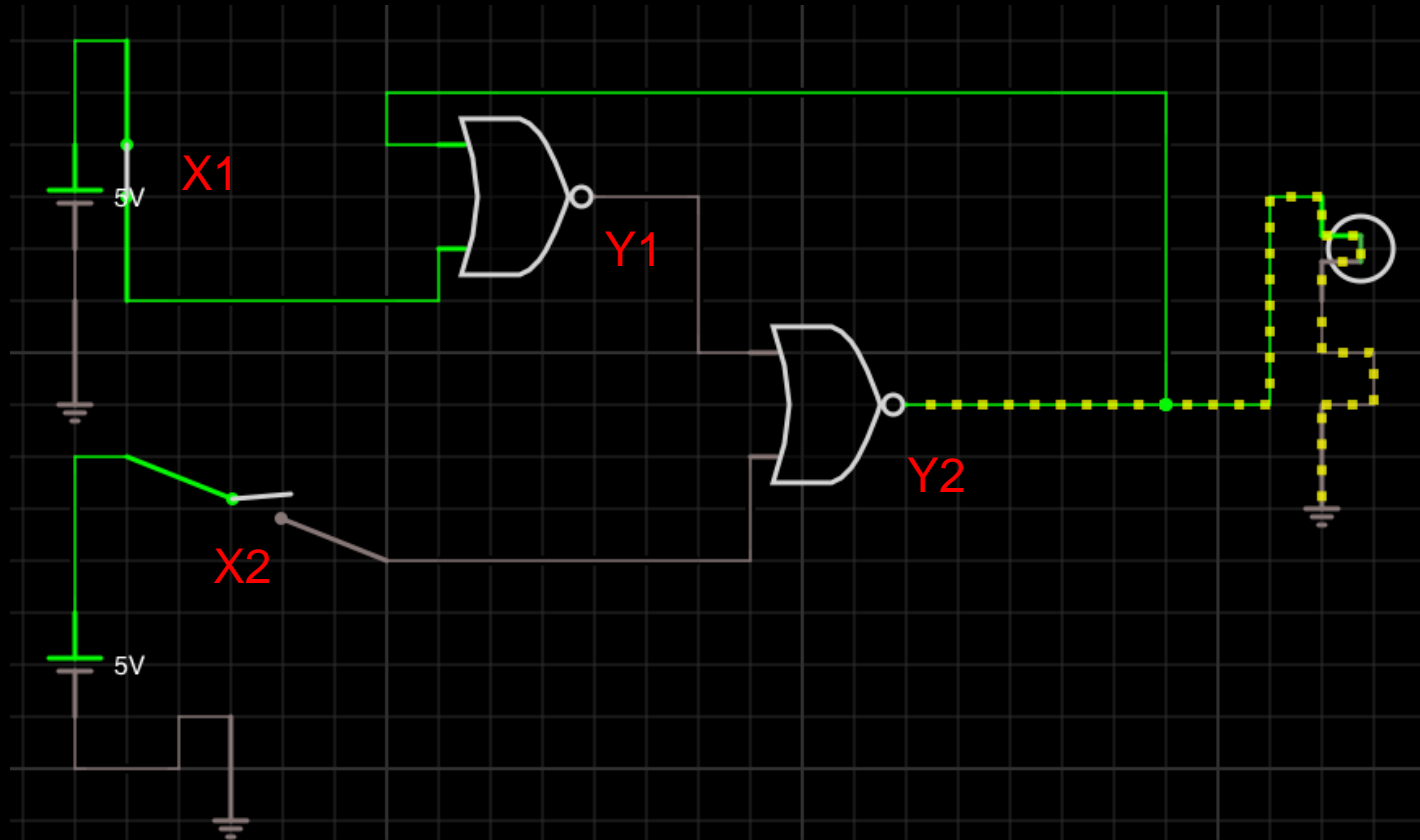


# 클럭 (clock)

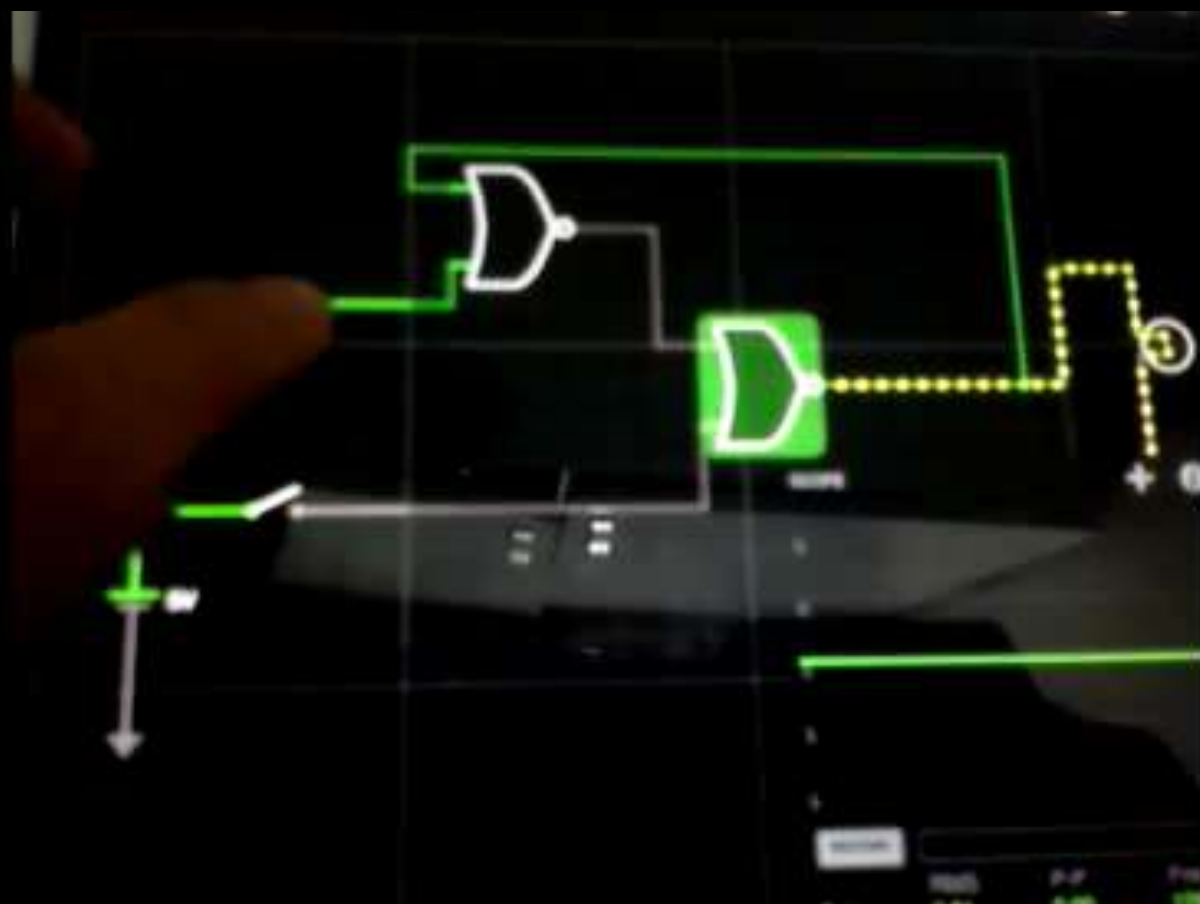
사이클



주기가 0.05라며 주파수는  $1/0.05$ 로 초당 20사이클  
이는 20Hz 라고 표현

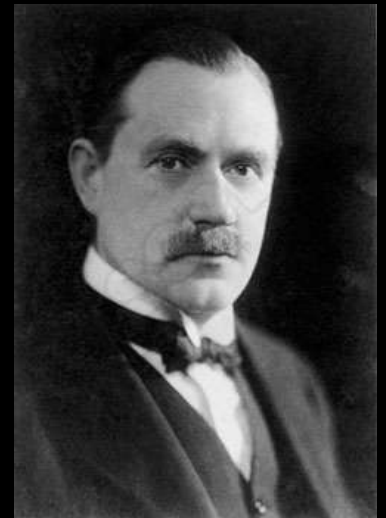


X1을 닫으면 전구가 켜지며, 그 이후에는 스위치가 열리더라도 Y2가 유지  
 X2을 닫으면 전구가 꺼지며, 그 이후에는 스위치가 열리더라도 Y2가 유지



# Flip flop ?

- 정보를 유지할 수 있음
- 이전 출력을 기억
- 1개의 f/f는 1bit의 정보를 저장
- Register의 구성요소
- RS, JK, D, T f/f가 존재
- 이 챕터에서는 D f/f만을 다룸



William Henry Eccles

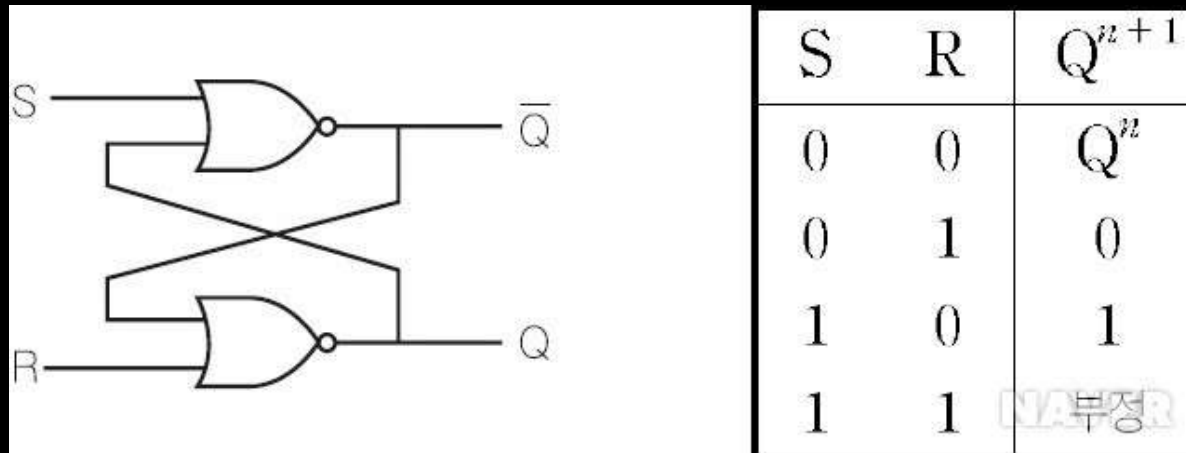
# Filp flop은 시소와 유사

- 2가지의 안정된 상태만 존재 ( $Q$ ,  $Q'$ )
  - 불안정한 상태로 남아 있지 않음



# SR f/f

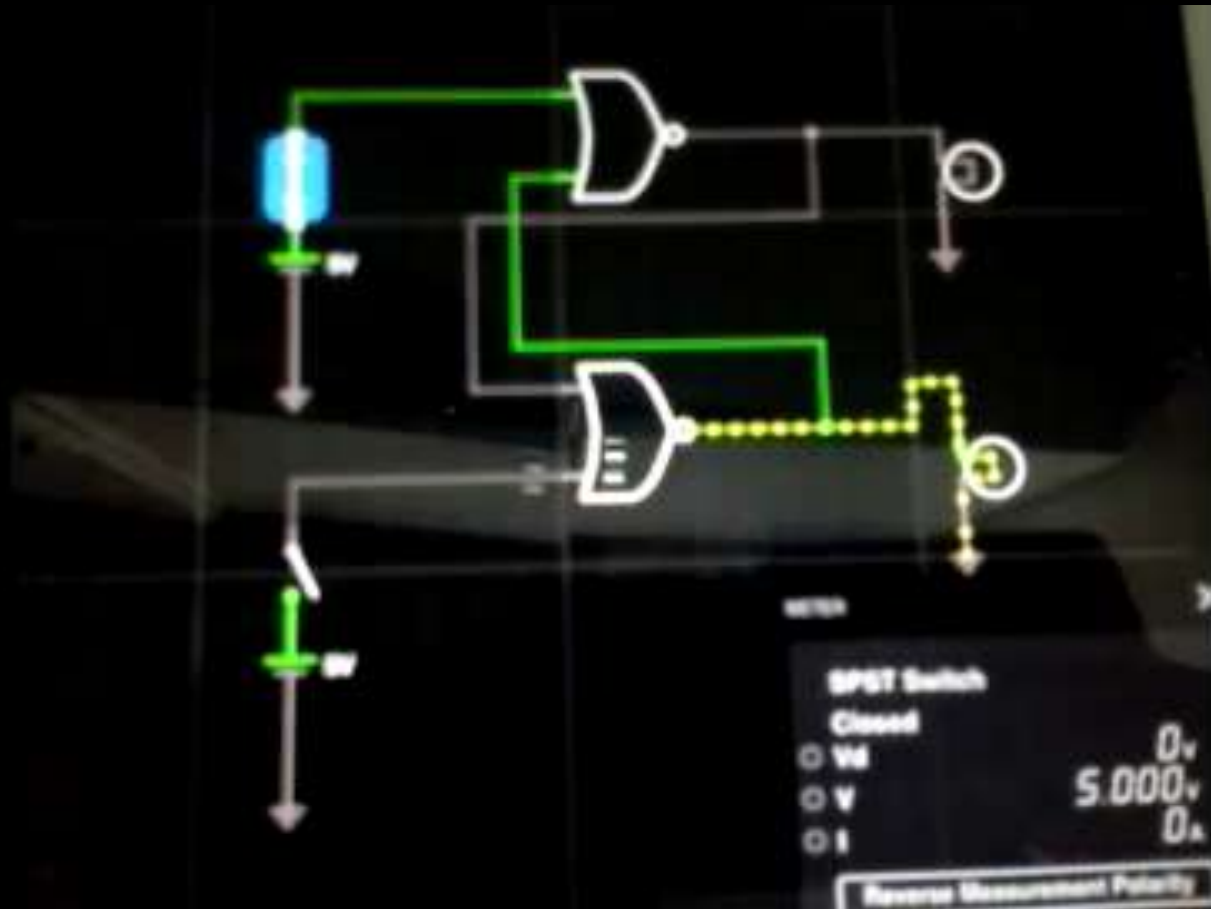
- 2개의 NOR 게이트, 2개의 귀환회로로 구성



Q와 Q'는 서로 반대되는 출력  
 S(set)은 Q를 1로 설정 (Q'는 0)  
 R(reset)은 Q를 0으로 초기화 (Q'는 1)

S	R	Q	Q'
1	0	1	0
0	1	0	1
0	0	Q	Q'
1	1	허용되지 않음	

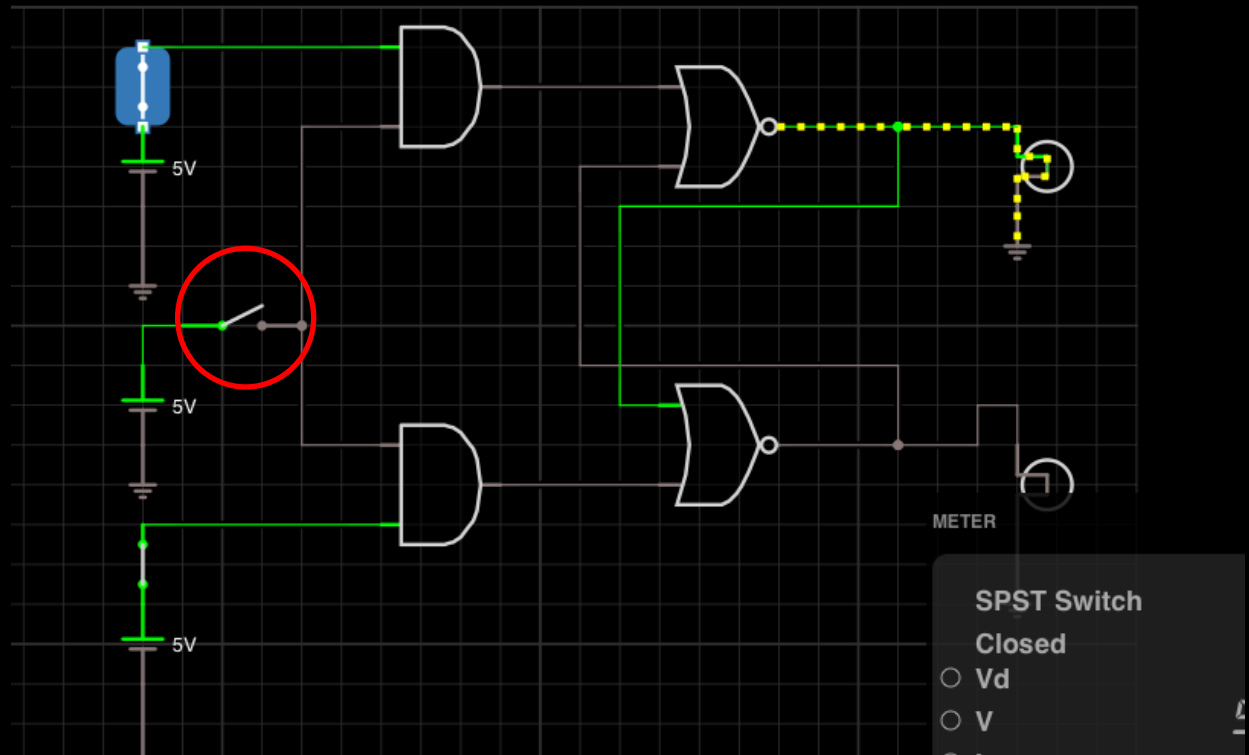
# SR f/f의 동작



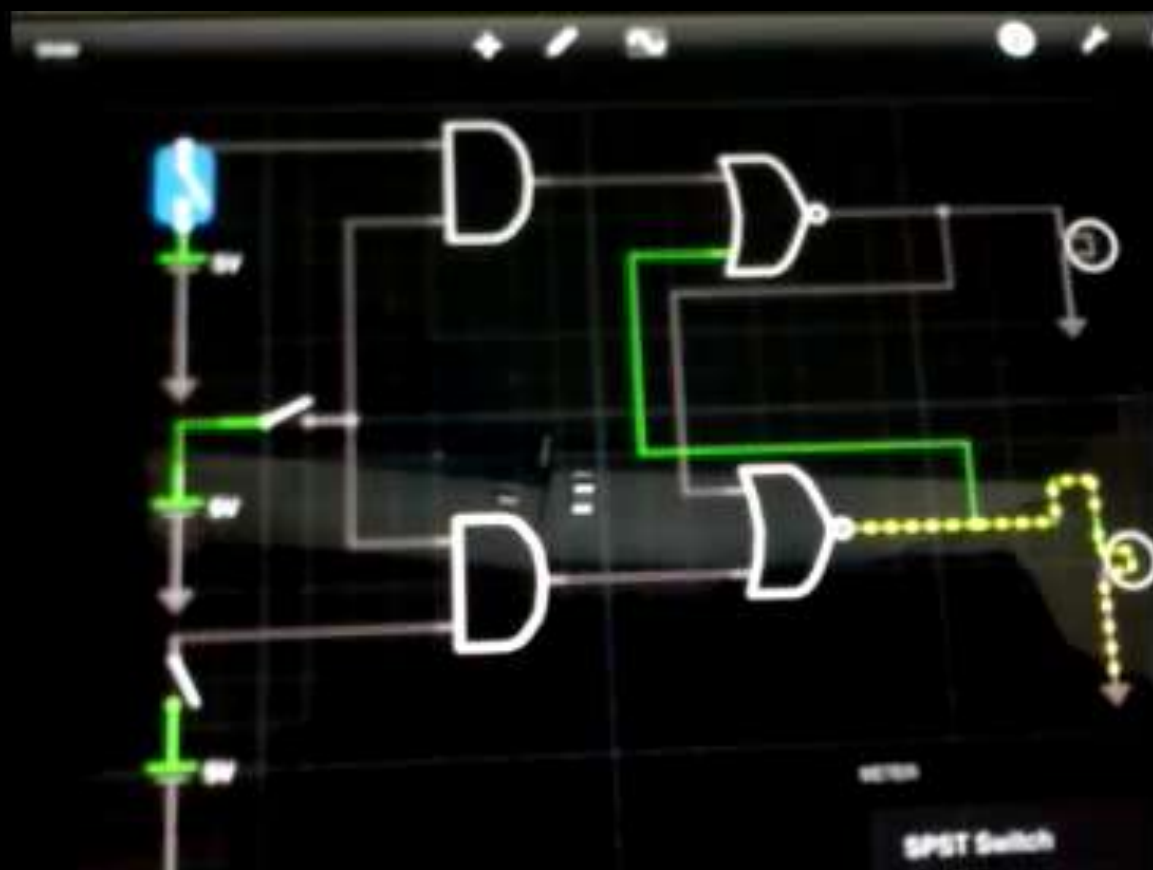
# 값 보존 신호를 가진 RS f/f

- 값 보존 신호가 0일 경우 데이터가 어떤 값이든 출력에 영향 없음
- 즉 값보존신호가 1일 경우 Reset과 set이 작동

입력	출력
0 1	0
1 1	1
X 0	Q



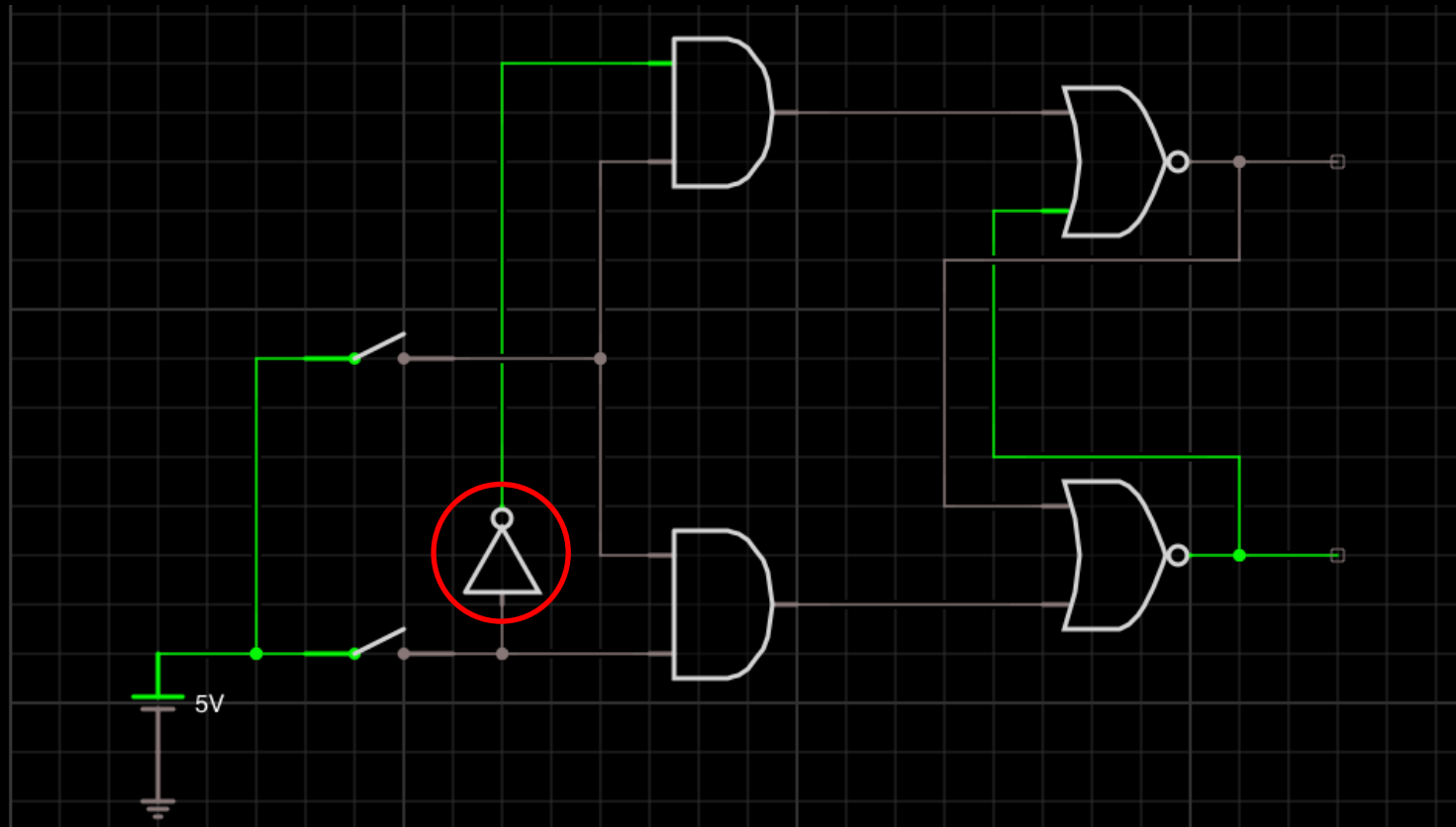


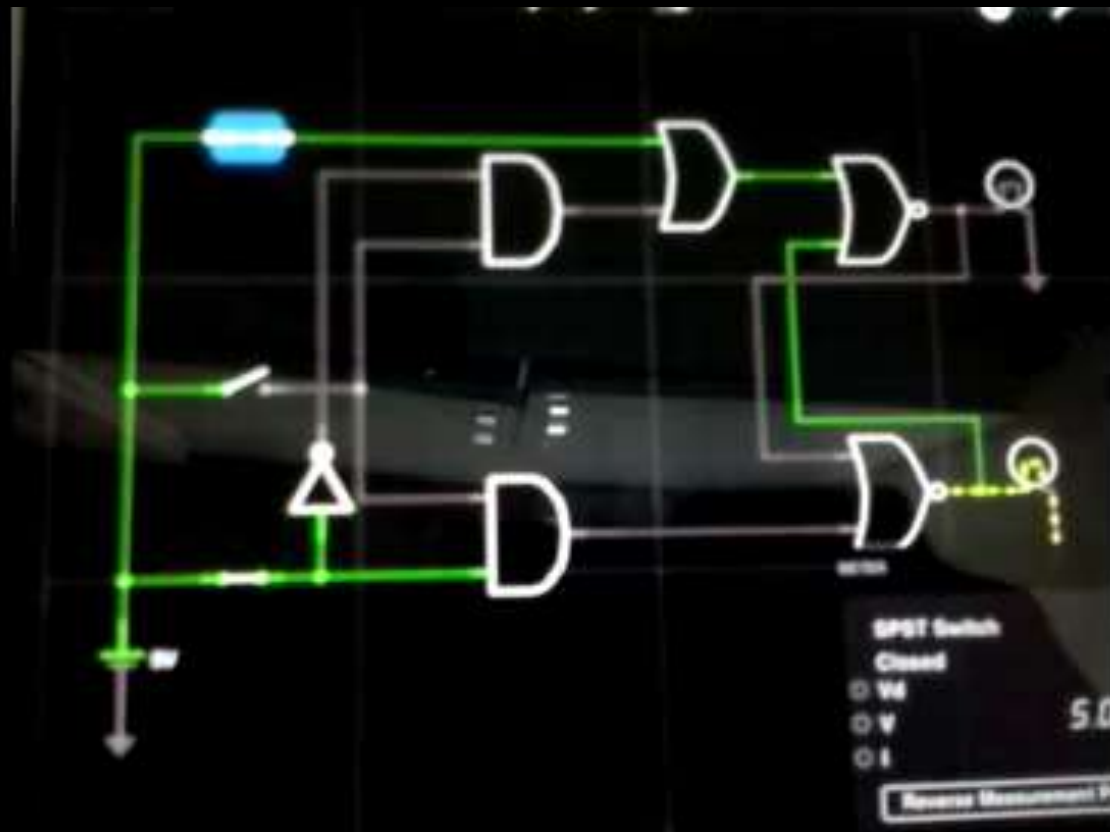


# 입력을 3개에서 2개로

- RS f/f에서는 입력이 1, 1은 불가
  - 0,0 / 0, 1 / 1, 0만이 입력으로 활용
- 출력값이 변하지 않게 하는 값보존 신호의 존재
  - 0, 1 / 1,0만이 입력으로 존재

Not gate 사용  
입력을 3->2로

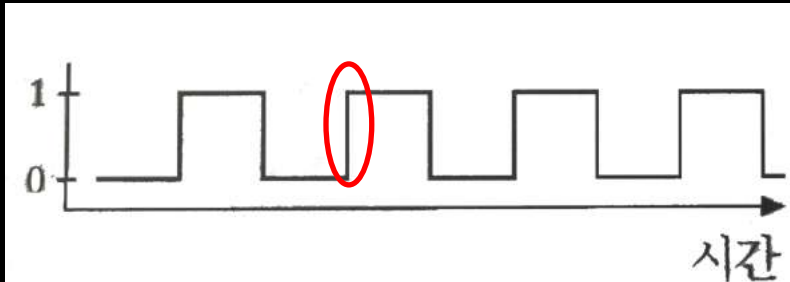




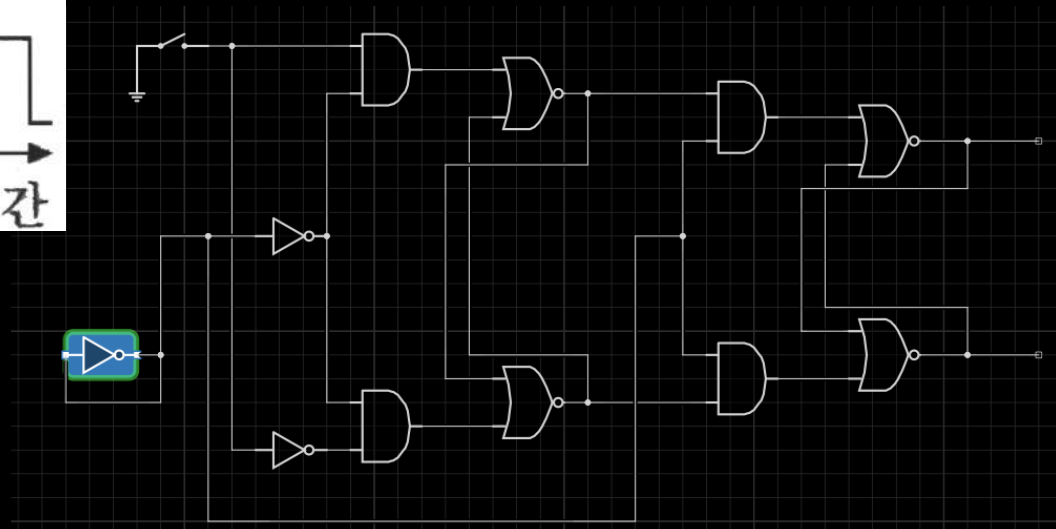


# Edge trigger D type f/f

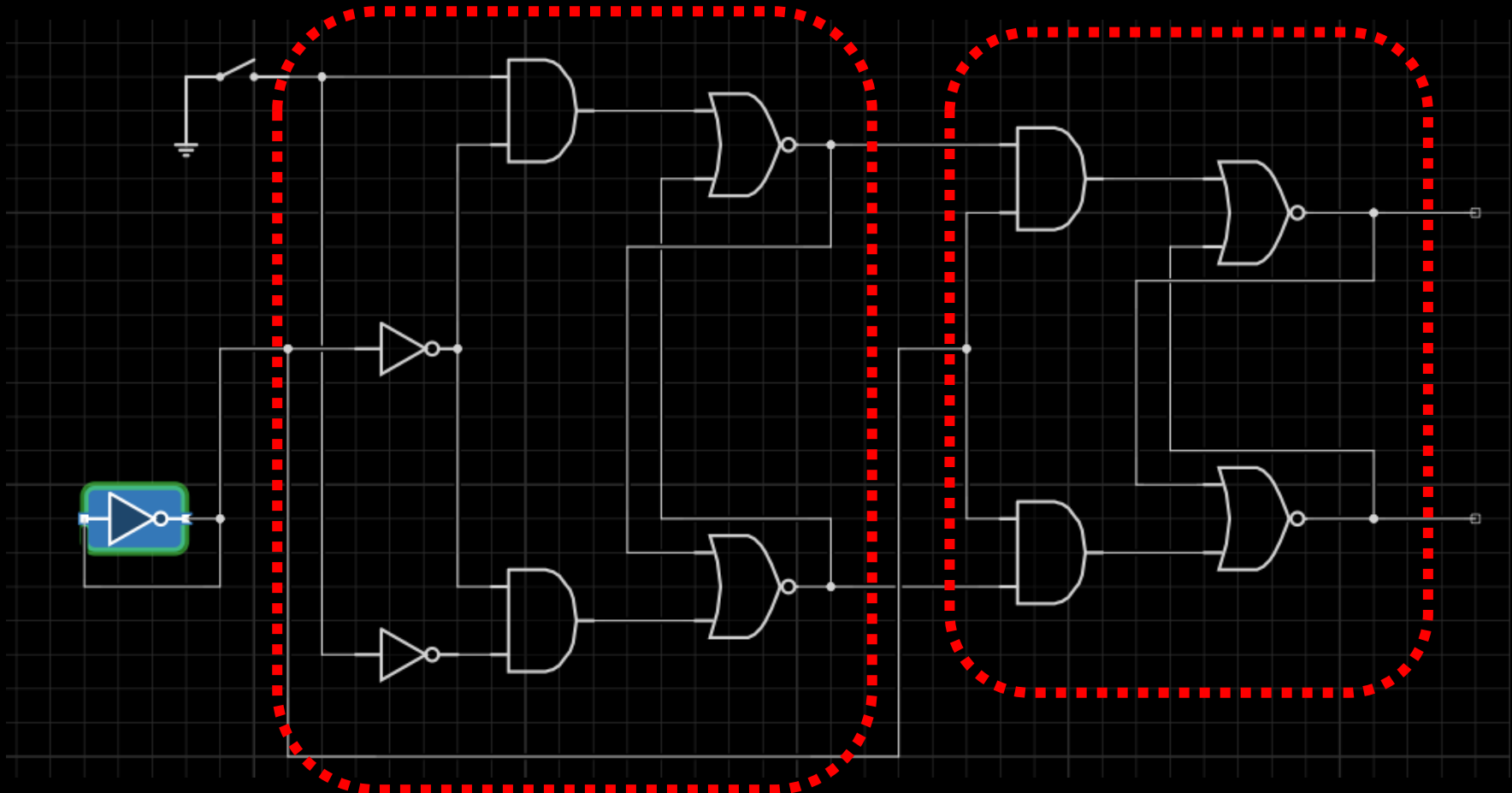
- 클럭(값보존)이 1이라도 데이터 입력이 출력에 영향을 주지 않음
- 클럭이 천이(0->1)될때만 입력이 출력에 영향



D	clk	Q	Q'
0	상	0	1
1	상	1	0
X	0	Q	Q'



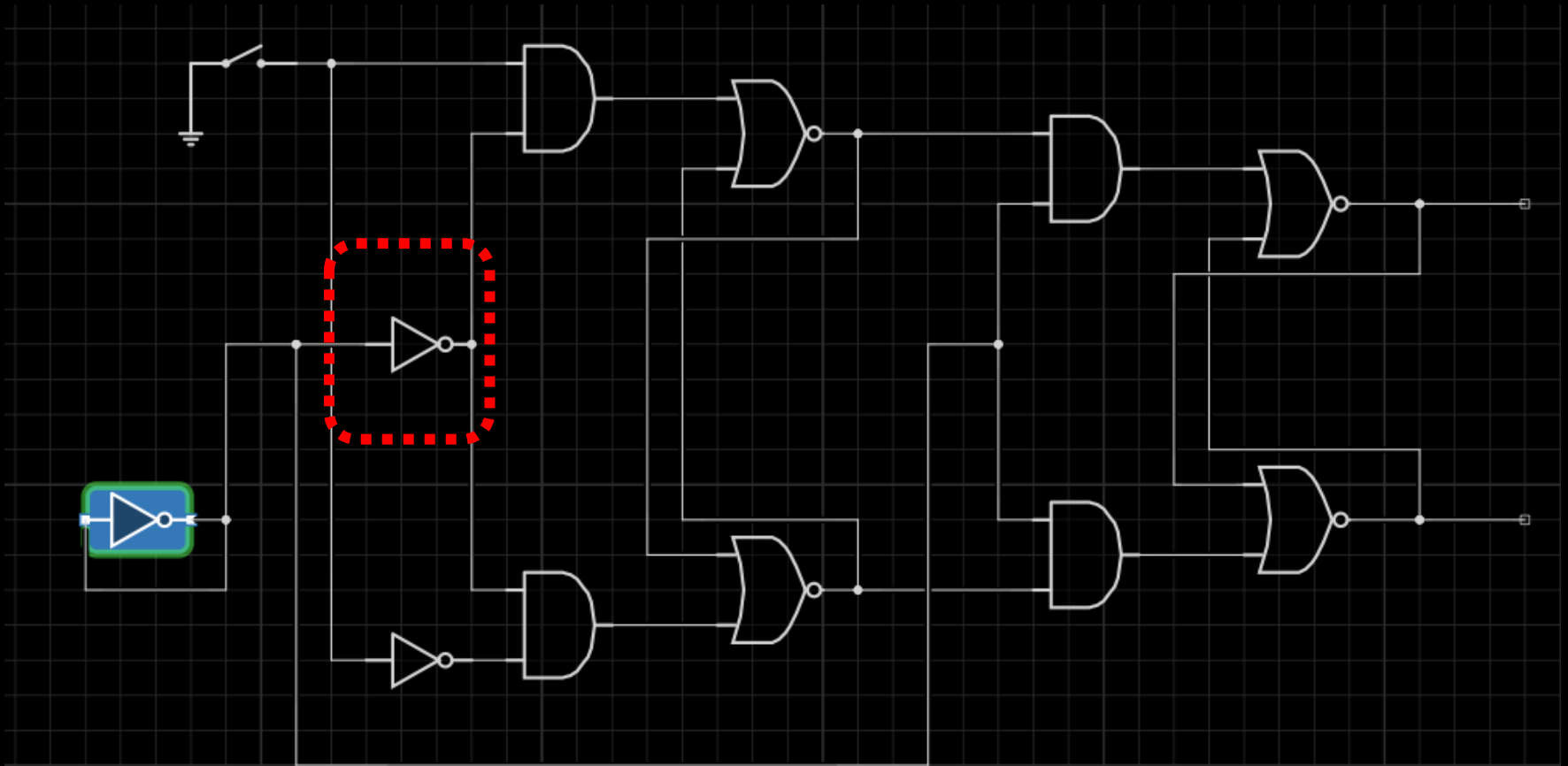
- 두 단계로 R-S f/f을 연결
  - 즉, 2개의 f/f으로 구성
- 데이터 입력이 바뀌어도 출력값에 영향 없음
  - 즉, 출력은 clk에 의해 영향

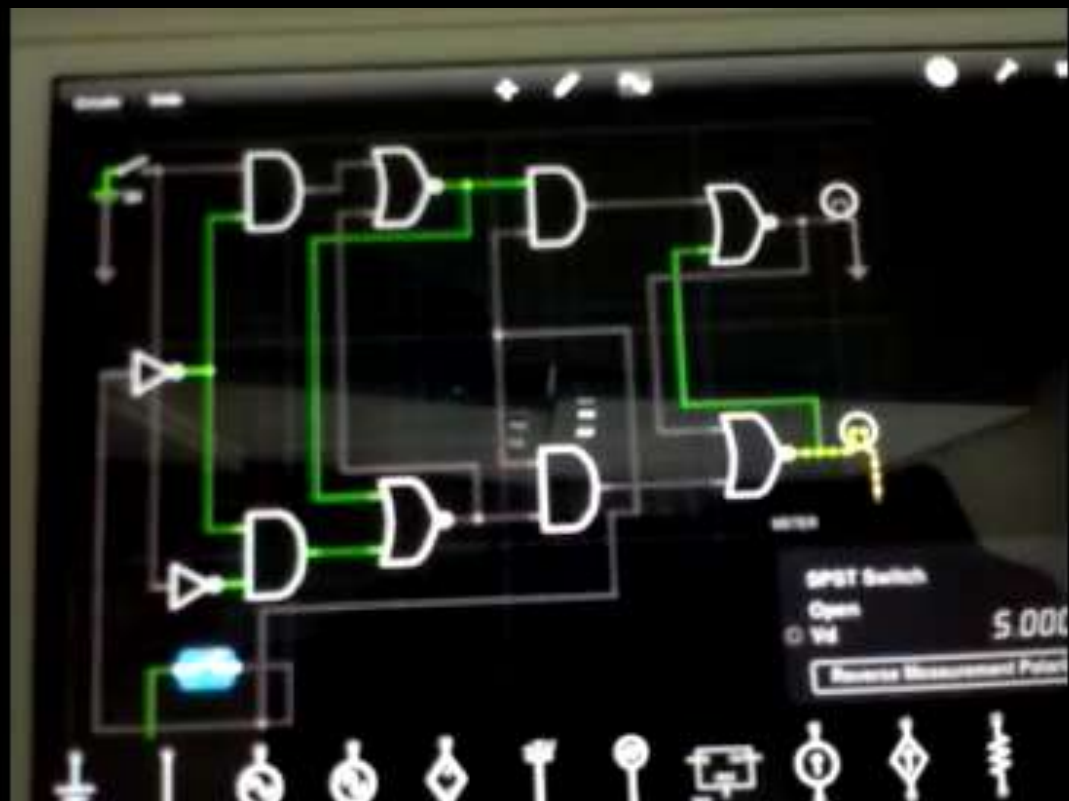


- 클럭입력이 두 단계 모두 제어

- 클럭입력이 0일때, 첫번째 f/f은 활성화 되어 값을 저장
- 클럭입력이 1일때, 두번째 f/f가 활성화 되어 값을 저장

**Not gate 사용으로 두 f/f을 제어**  
**결과적으로 0->1이 될때 두번째 f/f이 활성화**



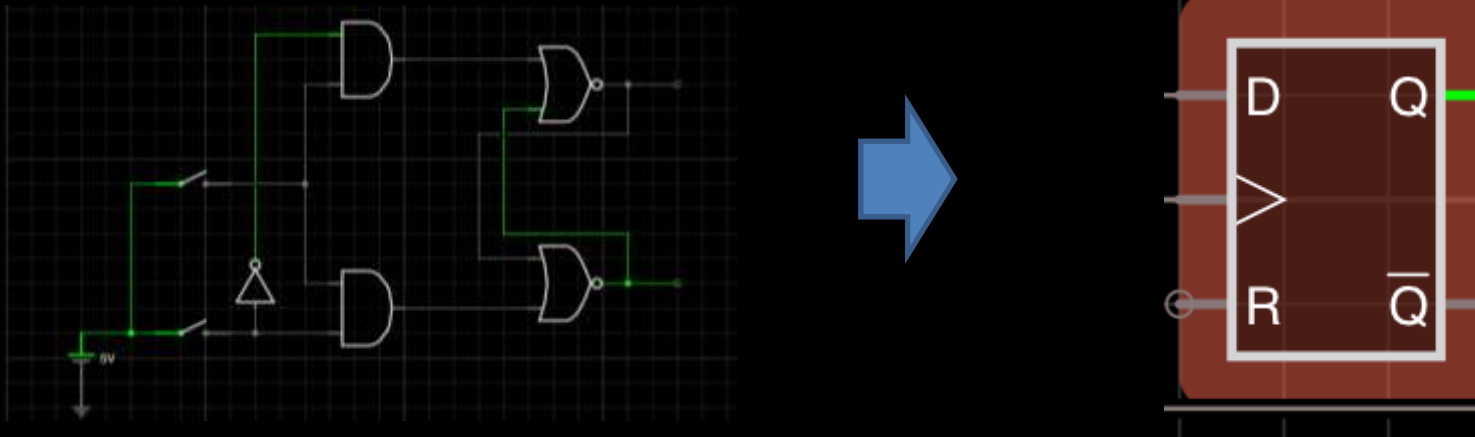




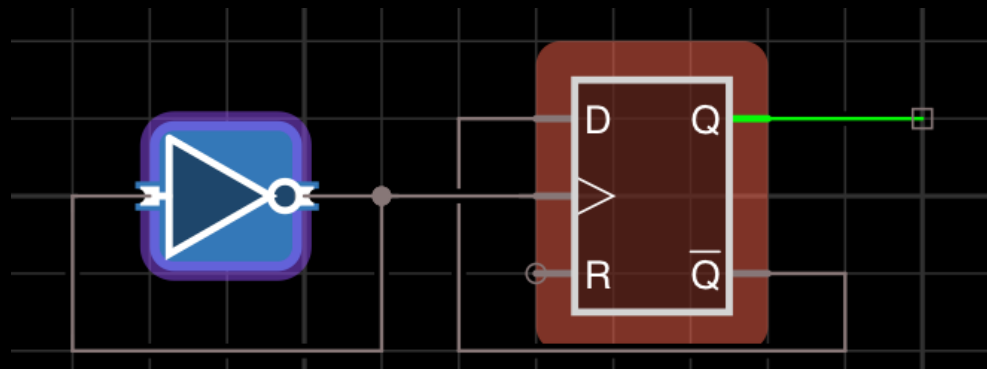
똑같은 회로(R-S f/f)을 두번 이용할 이유가 없습니다.

Edge trigger D type f/f을 다시 만들어 보겠습니다.

이제 D type latch를 아래와 같이 표현합니다.



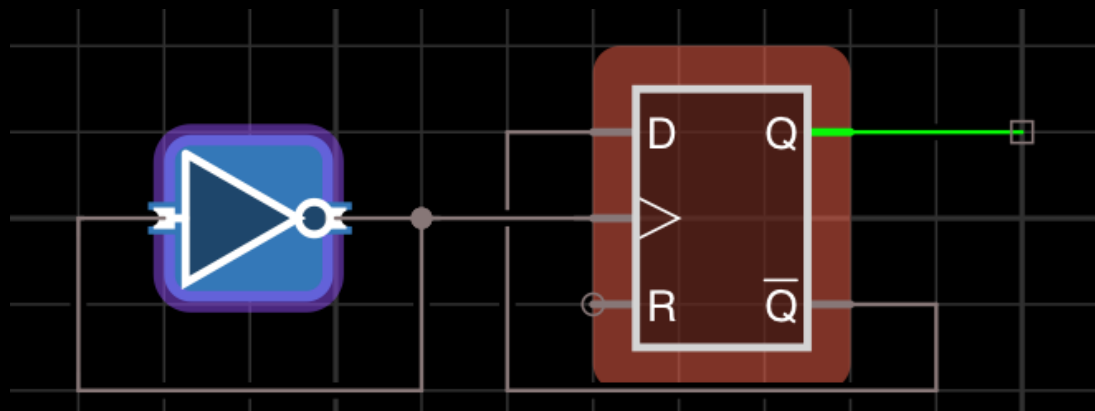
그리고 오실레이터 출력을 clk로 Q'를 입력으로 합니다.



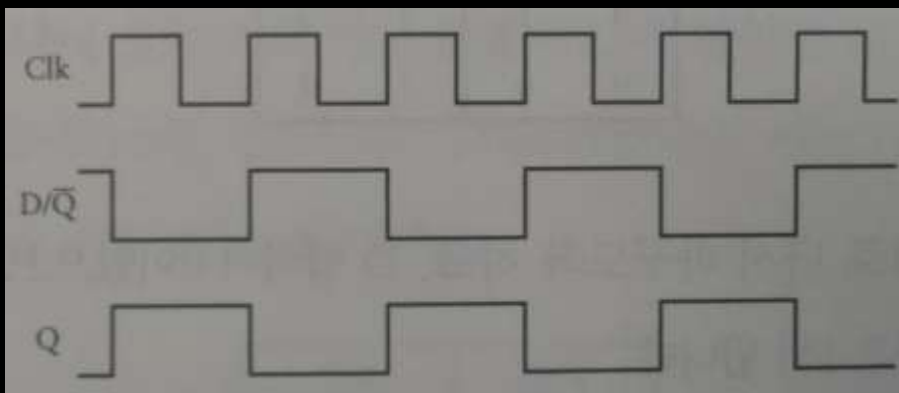
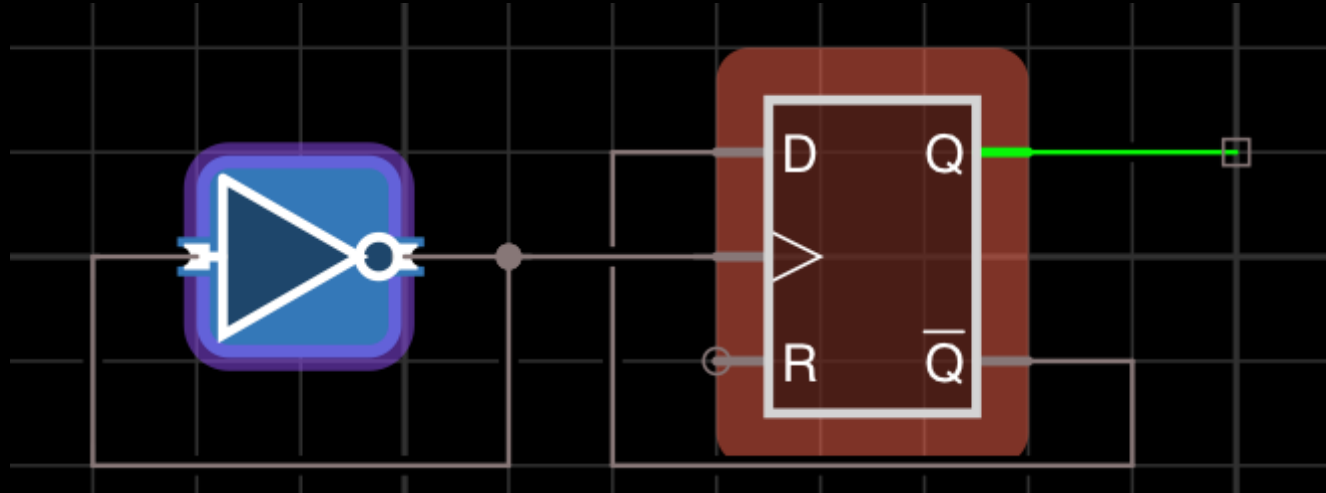
입력0, 출력(Q)가 0으로 시작. R-S f/f 2개를 사용하는 edge trigger D f/f과 같은 결과

D	clk	Q	Q'	D	clk	Q	Q'	D	clk	Q	Q'
1	0	0	1	1	0	0	1	1	0	0	1
	상			1	상	1	0	1	상	1	0
								0	1	1	0

D	clk	Q	Q'	D	clk	Q	Q'	D	clk	Q	Q'
1	0	0	1	1	0	0	1	1	0	0	1
1	상	1	0	1	상	1	0	1	상	1	0
0	1	1	0	0	1	1	0	0	1	1	0
0	0	1	0	0	0	1	0	0	0	1	0
				0	상	0	1	0	상	0	1
								1	1	0	1

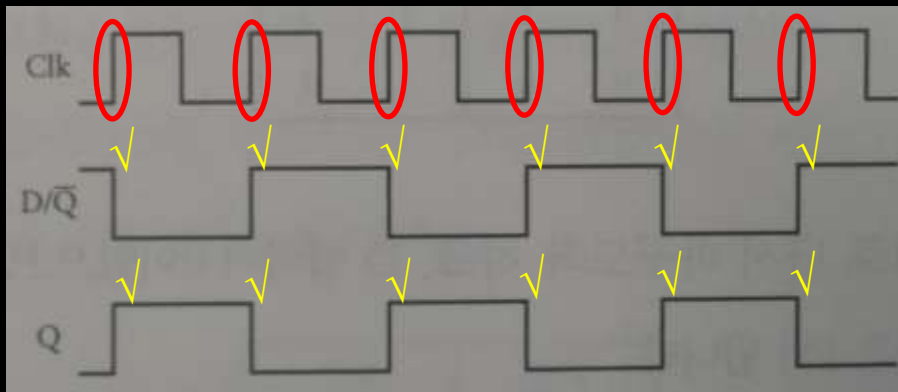
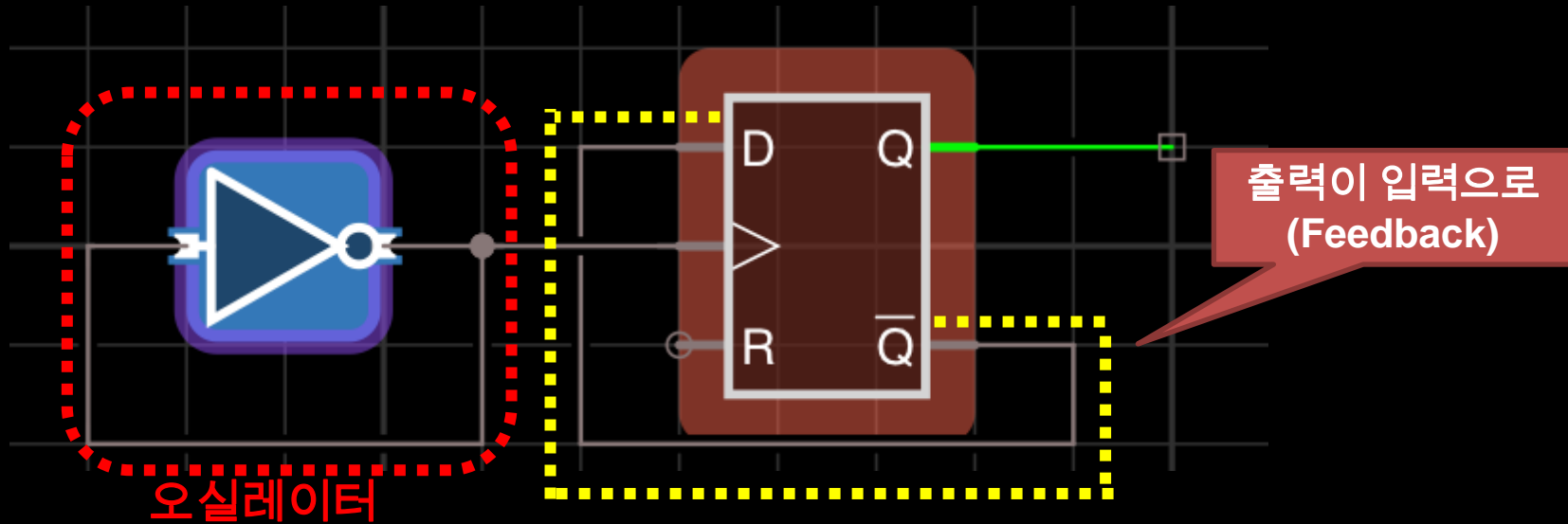


# Edge trigger type D f/f



D	clk	Q	Q'
1	0	0	1
1	상	1	0
0	1	1	0
0	0	1	0
0	상	0	1
1	1	0	1

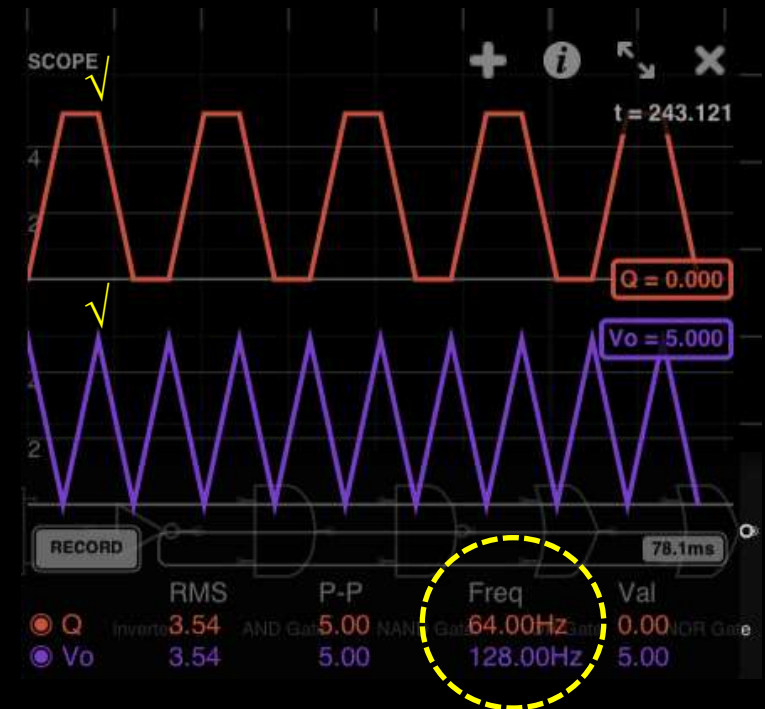
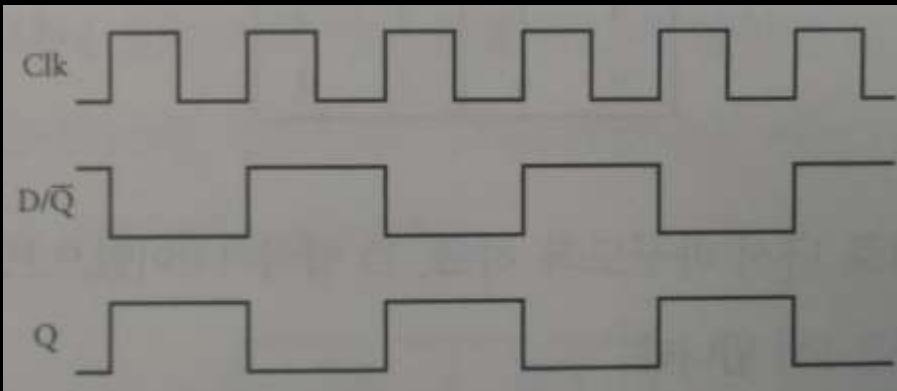
# Edge trigger type D f/f



D	clk	Q	Q'
1	0	0	1
1	상	1	0
0	1	1	0
0	0	1	0
0	상	0	1
1	1	0	1

Clk 상승할 때  
이전 Q, Q'의 값이 바뀐다.

# Edge trigger D f/f의 시뮬레이터 동작



Clk에 비하여  
D f/f의 출력 주파수가 1/2로 줄어듦

하나의 f/f을 거치면서

클럭 주파수가  $\frac{1}{2}$  감소하는

특징을 어떻게 활용할 수 있을까?

0000 : 0  
0001 : 1  
0010 : 2  
0011 : 3  
0100 : 4  
0101 : 5  
0110 : 6  
0111 : 7



0000 : 0  
0001 : 1  
0010 : 2  
0011 : 3  
0100 : 4  
0101 : 5  
0110 : 6  
0111 : 7



0 0 0 0 0 0 0 0  
1 1 1 1 0 0 0 0  
1 1 0 0 1 1 0 0  
1 0 1 0 1 0 1 0  
-----  
7 6 5 4 3 2 1 0

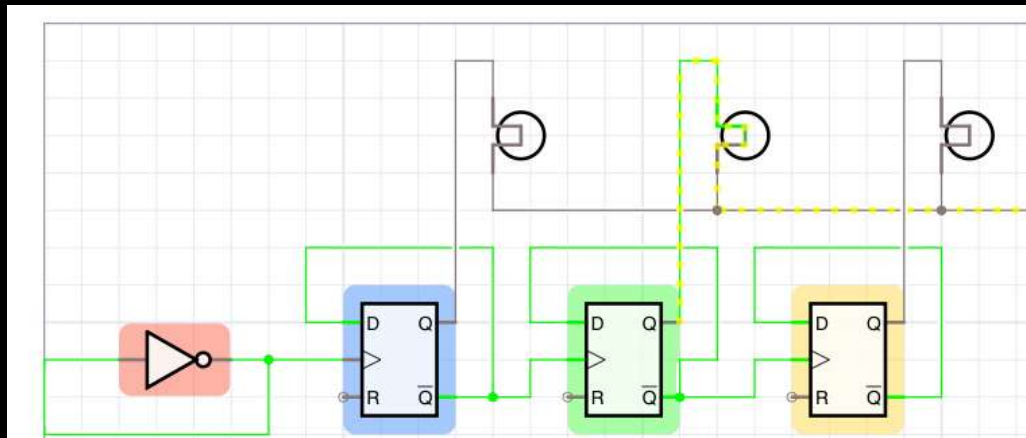


0 0 0 0 0 0 0 0  
0 0 0 0 1 1 1 1  
0 0 1 1 0 0 1 1  
0 1 0 1 0 1 0 1  
-----  
0 1 2 3 4 5 6 7

$$\frac{1}{2^n} \quad (n \geq 0)$$

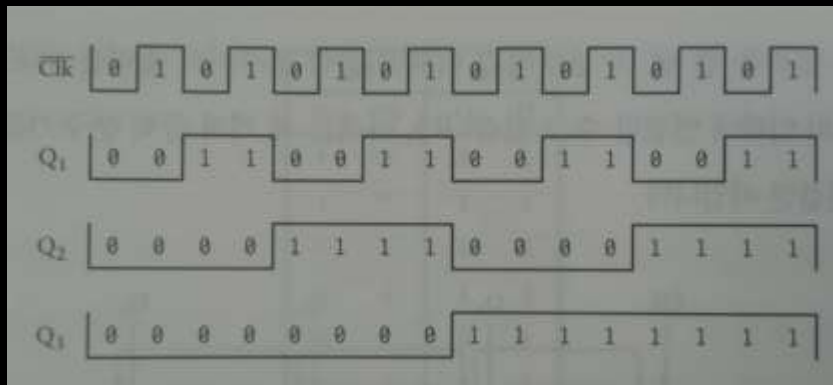
# 3bit ripple counter

- 각 f/f의 출력이 다음 f/f의 클럭 입력으로 들어가는 형태
  - 각 f/f을 거쳐 가므로 결과를 나타내기까지 시간이 걸림
    - 즉, 첫 f/f의 출력 시간과 마지막 f/f의 출력 시간이 틀림
    - 전파 지연 (propagation delay)
  - 비동기식 카운터



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

# 3bit ripple counter 시뮬레이터 동작



Q3

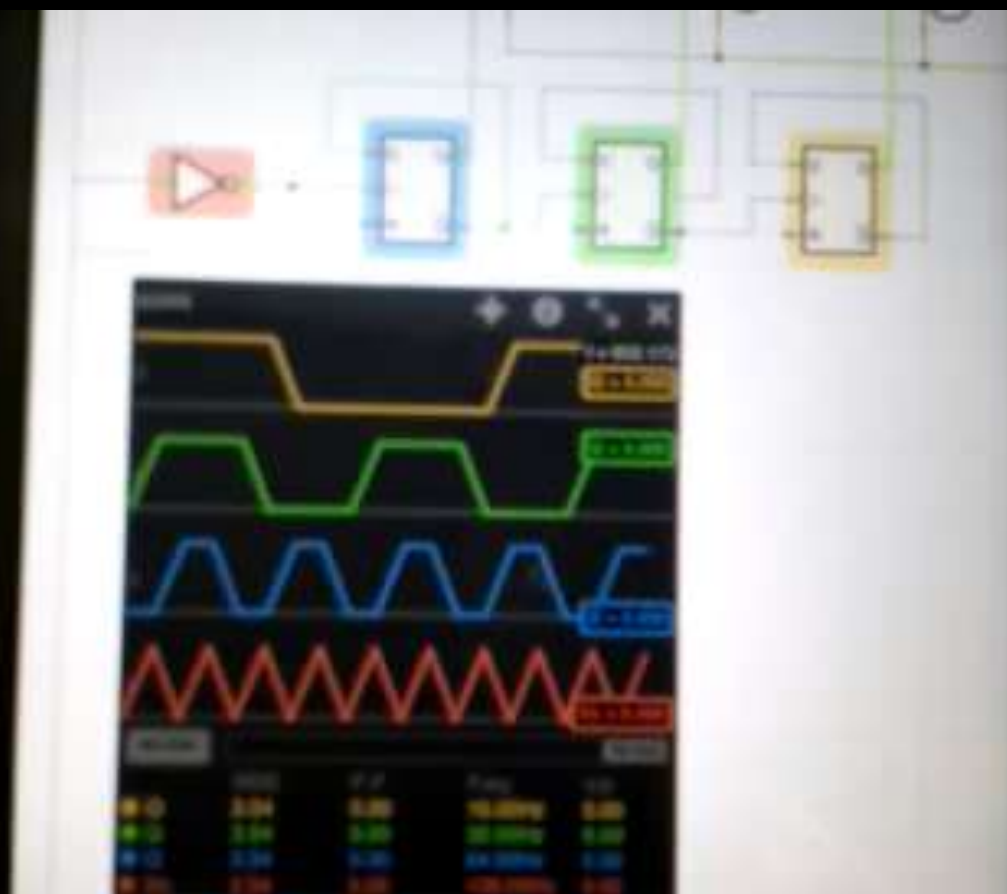
Q2

Q1

Clk

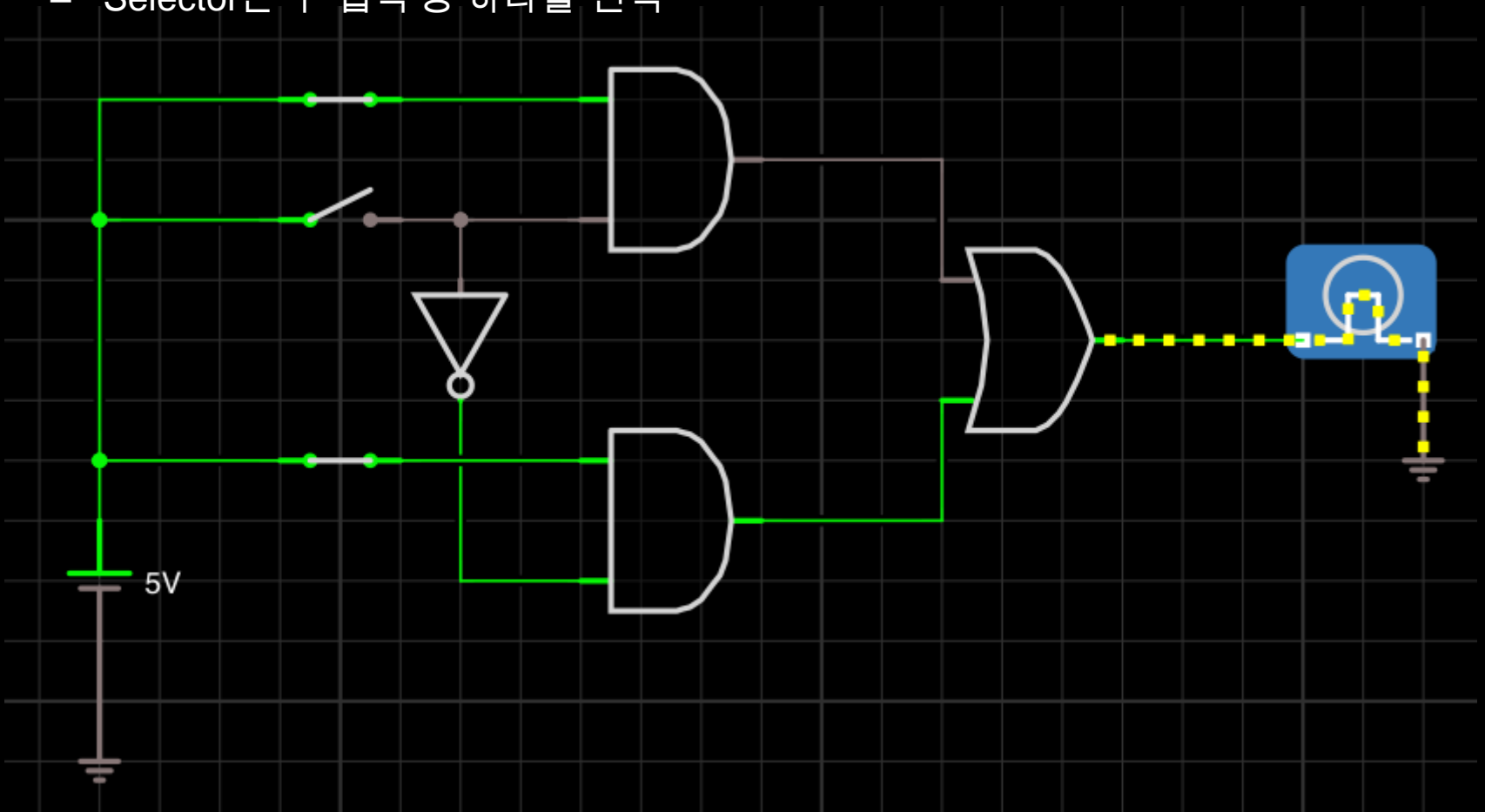






# Multiplex

- 다수의 입력 중 특정한 하나의 입력을 선택
  - 선택 스위치(selector)를 제외한 모든 입력은 ON (1)
  - Selector는 두 입력 중 하나를 선택



# Multiplex의 작동

