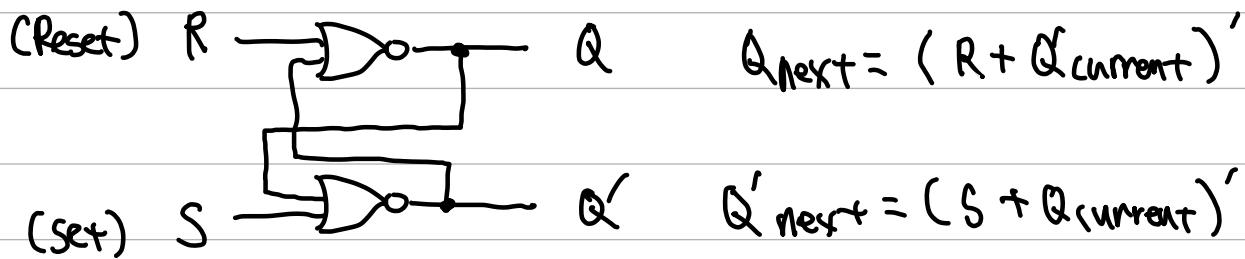


5. Sequential Logic Circuits

SR Latch



$$S=0, R=0 \text{ 일 때}$$

$$Q_{\text{next}} = (0 + Q'_{\text{current}})' = Q_{\text{current}}$$

$$Q'_{\text{next}} = (0 + Q_{\text{current}})' = Q'_{\text{current}}$$

→ 현재값 유지 (store)

$$S=1, R=0 \text{ 일 때}$$

$$Q'_{\text{next}} = (1 + Q_{\text{current}})' = 0$$

$\underbrace{\qquad\qquad\qquad}_{= 1 + 1} = 0$

$$Q_{\text{next}} = (0 + 0)' = 1$$

Q의 값이 1로 Set 된다.

$S=0, R=1$ 일 때

$$Q_{\text{next}} = (1 + Q'_{\text{current}})' = 0$$

:= 1 + 0 = 1

$$Q'_{\text{next}} = (0 + 0)' = 1$$

Q 의 값이 0으로 reset, clear 된다.

$S=1, R=1$ 일 때?

$$Q_{\text{next}} = (1)' = 0$$

된다 0이 된다.

$$Q'_{\text{next}} = (1)' = 0$$

↓ 그 후에 $S=0, R=0$ 상태가 되면

$$Q_{\text{next}} = (0+0)' = 1$$

$$Q'_{\text{next}} = (0+0)' = 1 \quad \Leftrightarrow \quad Q_{\text{next}} = (0+1)' = 0$$

$$Q'_{\text{next}} = (0+1)' = 0$$

값이 계속 바뀌게 된다.

$S \quad R \quad Q$

0 0 No Change

0 1 0 (reset)

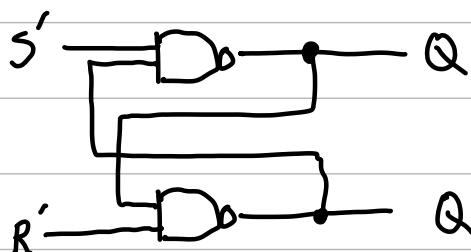
1 0 1 (set)

1 1 Avoid!

Inputs		Current		Next		
S	R	Q	Q'	Q	Q'	
0	0	0	1	0	1) <i>Set</i>
0	0	1	0	1	0	
0	1	0	1	0	1) <i>reset</i>
0	1	1	0	0	1	
1	0	0	1	1	0) <i>Set</i>
1	0	1	0	1	0	

Input이 같아도 상태에 따라서 결과가 다를 수 있음!

SR' Latch

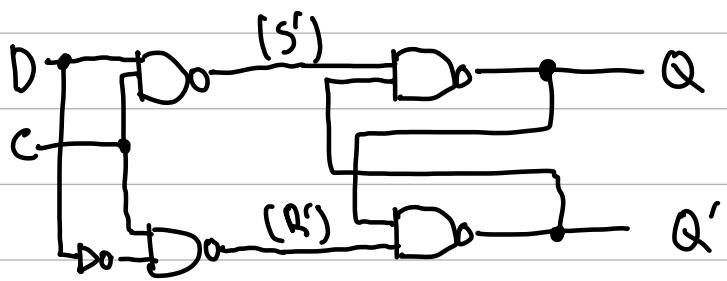


S'	R'	Q
1	1	No Change
1	0	0 (reset)
0	1	1 (set)
0	0	Avoid

C	S	R	S'	R'	Q
0	X	X	1	1	No Change
1	0	0	1	1	No Change
1	0	1	1	0	0 (reset)
1	1	0	0	1	1 (set)
1	1	1	0	0	Avoid

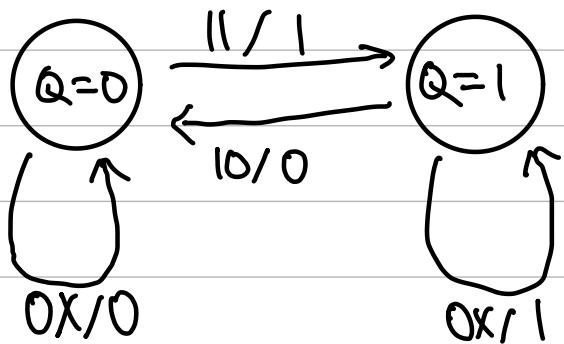
제어신호

D Latch D for data C for control

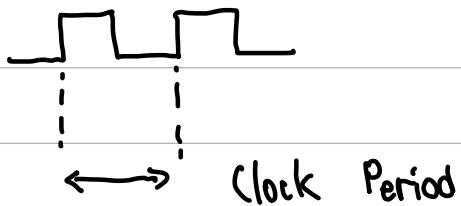


C	D	Q
0	X	No Change
1	0	0
1	1	1

State Diagram (input / output , n input \rightarrow 2^n arrows)



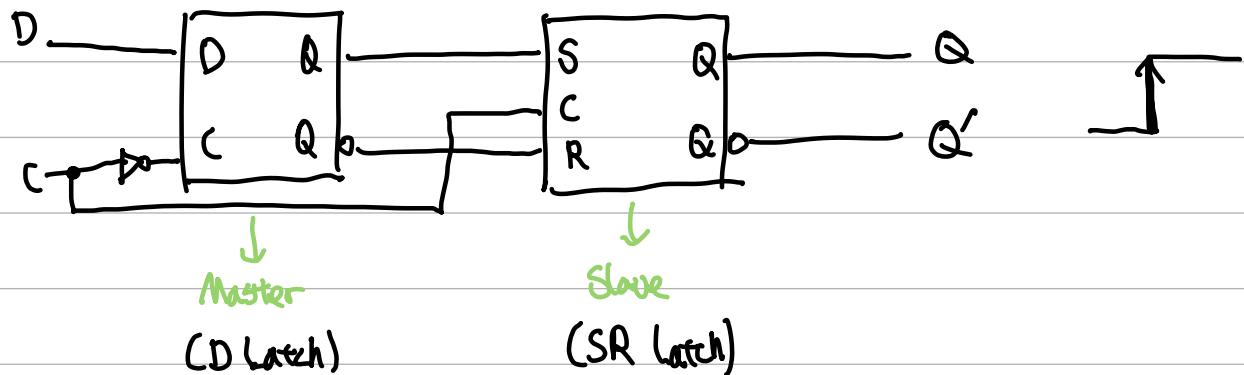
Clock



↓ inverse
clock frequency (Hertz)

D Flip-Flop

(Positive edge triggering)



$C=0$ 일 때 : Master 가 활성화, D input에 따라 Q 설정
 Slave는 현재값 유지

$C=1$ 일 때 : Master는 현재값 유지 (즉, (가 이전에 주입된 D 값)
 Slave가 활성화, Master의 Q 값이 Q로 업데이트.

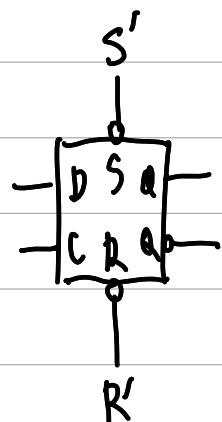
Latch: level sensitive

Flip-flop: edge sensitive

Positive

Negative

Direct Input



S' R' C D Q

0 0 X X Avoid

0 1 X X 1 (set)

1 0 X X 0 (reset)

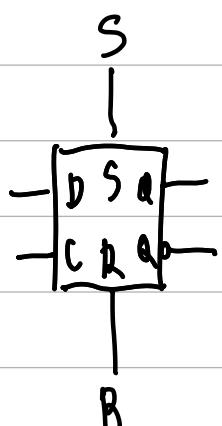
1 1 0 X No Change

1 1 1 0 0 (reset)

1 1 1 1 1 (set)

정지3 Set, reset

기존 D flip-flop



S R C D Q

1 1 X X Avoid

1 0 X X 1 (set)

0 1 X X 0 (reset)

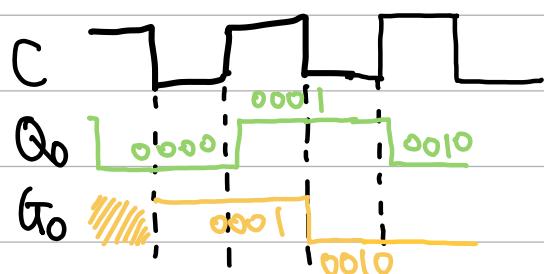
0 0 0 X No Change

0 0 1 0 0 (reset)

0 0 1 1 1 (set)

정지3 Set, reset

기존 D flip-flop

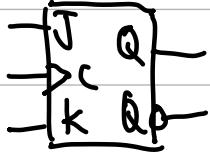


C가 처음 1초면 Q_0 은 Direct 초기화

C가 0이되면 Q 는 ALU로 출력, 6가 계산됨.

Flip-Flop Variation

JK F-F: SR Latch처럼 동작. JK=11 일땐 Q' current 3 바꿈.



J	K	Q_{next}
0	X	No Change
1	0	No Change
1	0	0 (reset)
1	1	1 (set)
1	1	Q' current (complement)

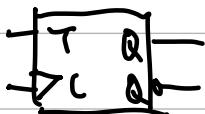


= D F-F



= T F-F

T F-F



T	C	Q_{next}
0	X	No Change
1	0	No Change
1	1	Q' current

Characteristic tables

D Q(t+1) Operation

0	0	reset	$Q(t+1) = 0$
1	1	set	

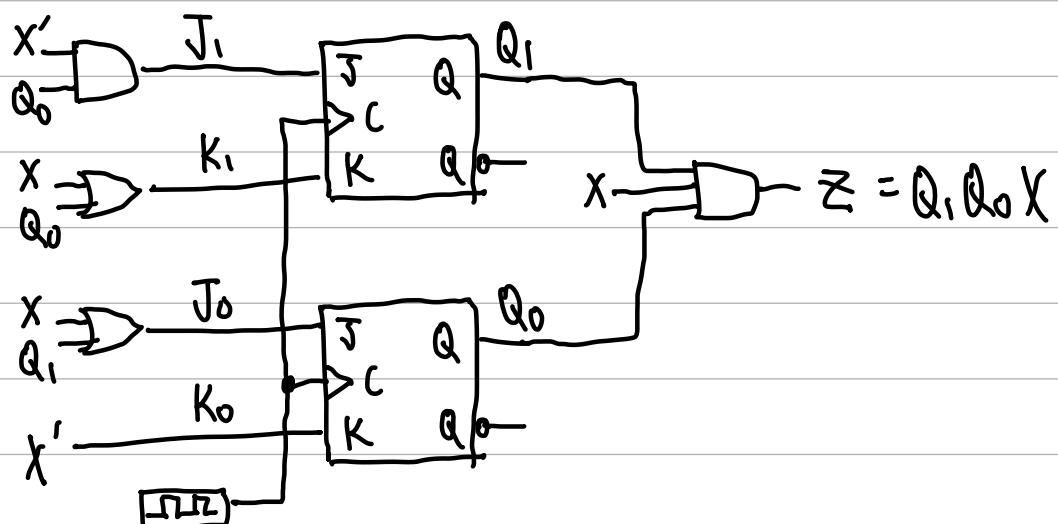
J K Q(t+1) Operation

0	0	$Q(t)$	No change	$Q(t+1) = K'Q(t) + JQ(t)$
0	1	0	reset	
1	0	1	set	
1	1	$Q'(t)$	complement	

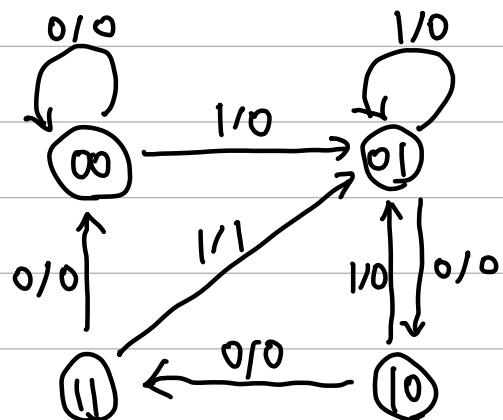
T Q(t+1) Operation

0	$Q(t)$	No change	$Q(t+1) = T'Q(t) + TQ'(t)$
1	$Q'(t)$	complement	$= T \oplus Q(t)$

Analyzing sequential circuit



Present State	Input	FF Input	Next State	Output
$Q_1\ Q_0$	X	$J_1\ K_1\ J_0\ K_0$	$Q_1\ Q_0$	Σ
0 0	0	0 0 0 1	0 0	0
0 0	1	0 1 1 0	0 1	0
0 1	0	1 1 0 1	1 0	0
0 1	1	0 1 1 0	0 1	0
1 0	0	0 0 1 1	1 1	0
1 0	1	0 1 1 0	0 1	0
1 1	0	1 1 1 1	0 0	0
1 1	1	0 1 1 0	0 1	1



n개의 FF $\rightarrow 2^n$ 개의 노드

m개의 input \rightarrow 각 노드마다 2^m 개의 출발점

Mealy Model

Output이 present state와 input으로 결정됨.

Moore Model

Output이 present state만으로 결정됨.

State Reduction

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]

완전히 같은 일이 일어남.

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]

g 대신 e로 대체 후
또 같은 일이 일어나는 부분
찾아내서 삭제

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

이상 줄일 것 없음.

a b c d e f g State에 bit assignment

binary, gray code, one-hot 등...

M개의 state 가 있다면 $\lceil \log_2 M \rceil$ bit가 필요하다.

Sequence Recognizer

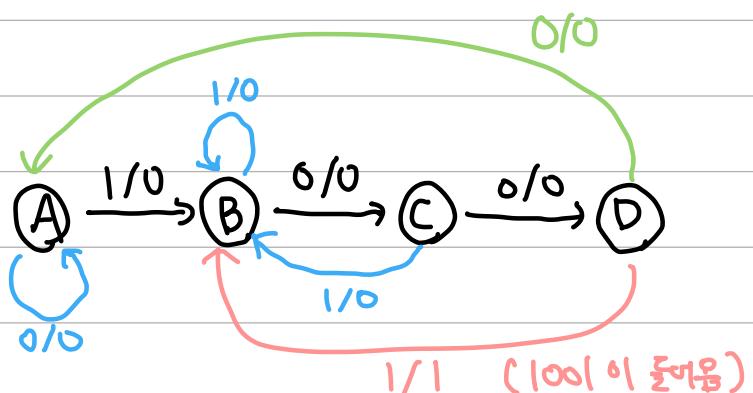
input에 "1001"이 순서대로 들어오면 output이 1이 되게 함.

A: 아무도 안맞음.

B: 1 까지 들어옴.

C: 10 까지 들어옴.

D: 100 까지 들어옴.



Mealy model임. → Moore로 만드는법?

Present State	input	Next State	Output
A	0	A	0
A	1	B	0
B	0	C	0
B	1	B	0
C	0	D	0
C	1	B	0
D	0	A	0
D	1	B	1

↓ State on code assign

Present State	input	Next State	Flip-Flop inputs	Output
Q_1, Q_0	X	Q_1, Q_0	J_1, K_1, J_0, K_0	Z
0 0	0	0 0	0 X 0 X	0
0 0	1	0 1	0 X 1 X	0
0 1	0	1 0	1 X X 1	0
0 1	1	0 1	0 X X 0	0
1 0	0	1 1	X 0 1 X	0
1 0	1	0 1	X 1 1 X	0
1 1	0	0 0	X 1 X 1	0
1 1	1	0 1	X 1 X 0	1

$$J_1 = X' Q_0, \quad K_1 = X + Q_0, \quad J_0 = X + Q_1, \quad K_0 = X', \quad Z = Q_1 Q_0 X$$

JK characteristic table

J	K	$Q(t+1)$	Operation
0	0	$Q(t)$	No Change
0	1	0	reset
1	0	1	Set
1	1	$Q(t)$	Complement

JK excitation table

$Q(t)$	$Q(t+1)$	J	K	Operation
0	0	0	x	No Change / reset
0	1	1	x	Set / complement
1	0	x	1	reset / complement
1	1	x	0	No Change / set

D excitation table

$Q(t)$	$Q(t+1)$	D	Operation
0	0	0	reset
0	1	1	set
1	0	0	reset
1	1	1	set

T excitation table

$Q(t)$	$Q(t+1)$	T	Operation
0	0	0	No Change
0	1	1	Complement
1	0	1	Complement
1	1	0	No Change