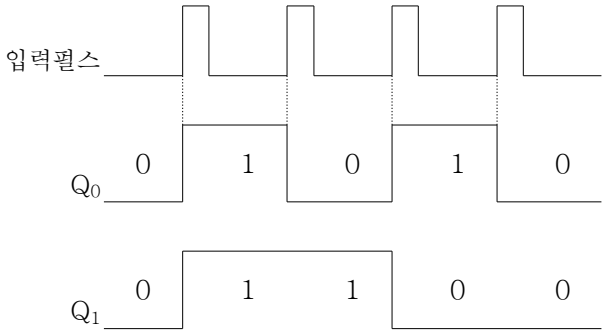
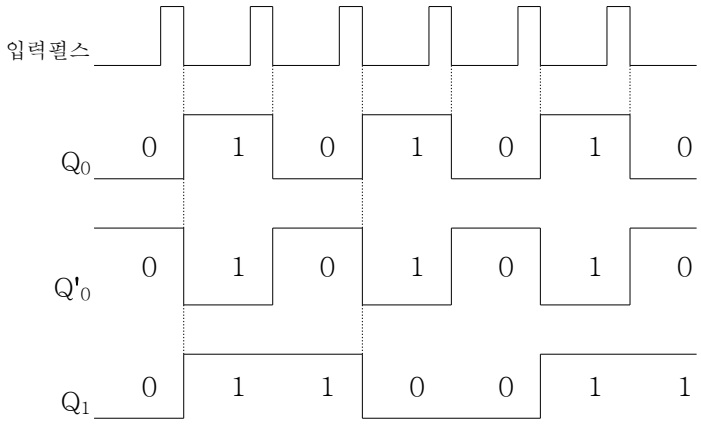
8.1



==> 입력 펄스의 상승 에지에서 트리거 되며, 업 카운팅이 아닌 다운 카운팅을 하게 된다.

8.2



CLK	1	2	3	4	5	6
Q_0	0	1	0	1	0	1
Q_1	0	0	1	1	0	0
값	0	1	2	3	0	1

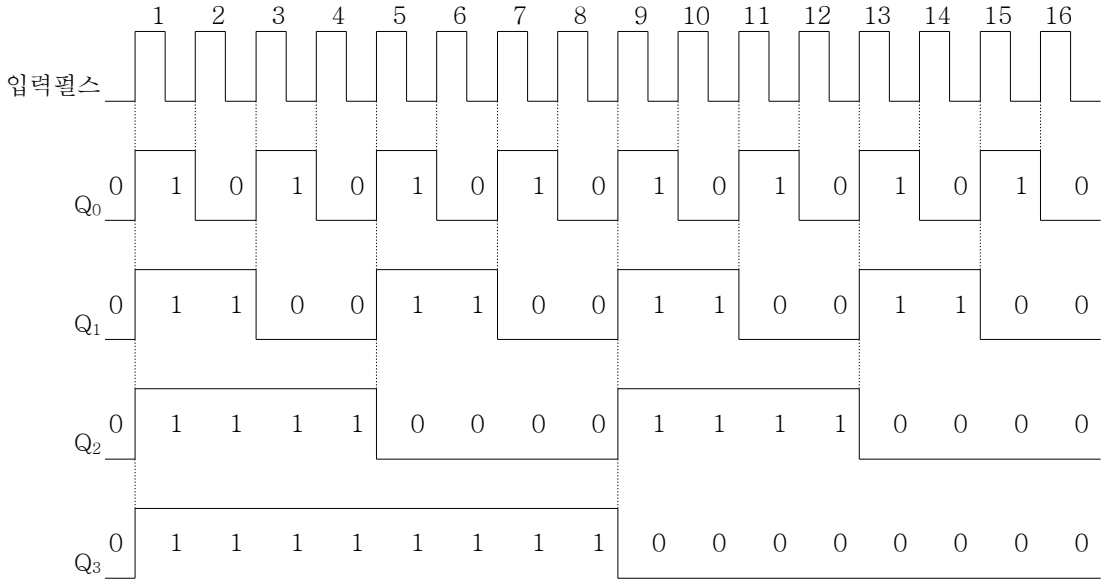
원래 회로

CLK	1	2	3	4	5	6
Q_0	0	1	0	1	0	1
Q_1	0	1	1	0	0	1
값	0	3	2	1	0	3

변경된 회로

==> 회로는 다운 카운터로 변경된다.

8.3



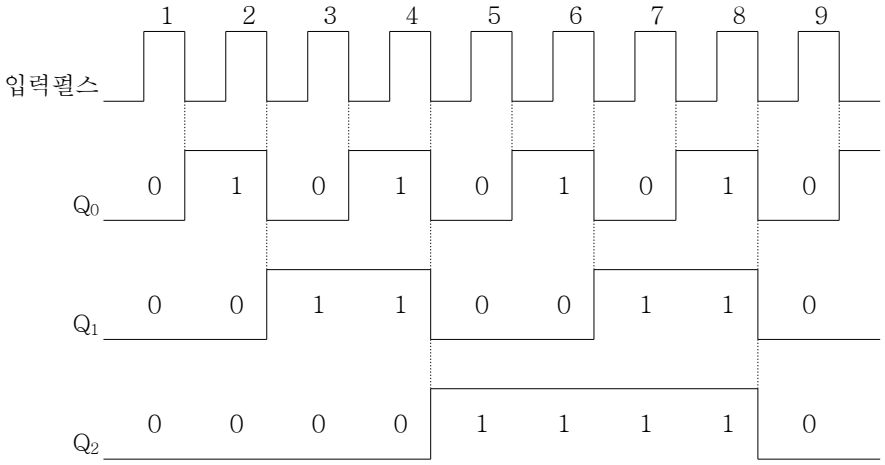
8.4 5개의 플립-플롭들에 의한 전체 전파지연시간은 5×10ns = 50ns가 된다. 그리고 다음 입력 펄스에 의한 상태 변화는 10ns 후에 발생하므로, 출력값을 확인할 여유 시간으로 20ns를 허용하려면 20ns-10ns = 10ns가 더 필요하게 된다. 결과적으로, 입력 신호에 허용되는 최단 주기는 50ns+ 10ns = 60ns가 된다.

8.5 (1) 8-비트 리플 카운터: 8×15ns=120ns
(2) 8-비트 동기식 카운터: 15ns

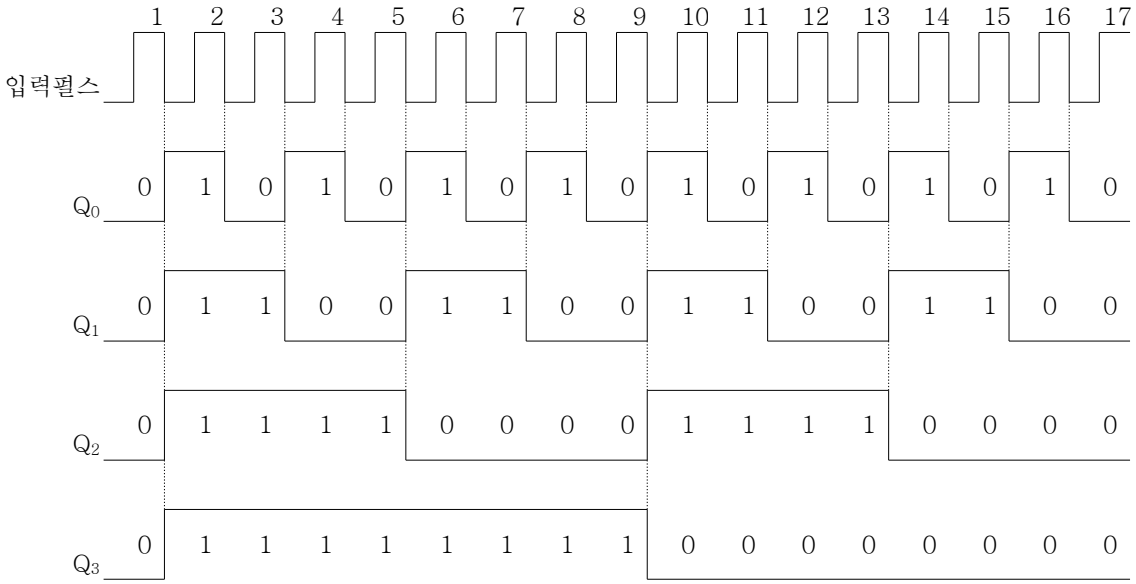
8.6
- 1010 경우 : 1010 → 1011 → 0100 → 0101 이므로, 2번만에 정상적인 상태 시퀀스로 진입
- 1011 경우 : 1011 → 0100 → 0101 이므로, 1번만에 정상적인 상태 시퀀스로 진입
- 1100 경우 : 1100 → 1101 → 0110 → 0111 이므로, 2번만에 정상적인 상태 시퀀스로 진입
- 1101 경우 : 1101 → 0110 → 0111 이므로, 1번만에 정상적인 상태 시퀀스로 진입
- 1110 경우 : 1110 → 1111 → 0000 → 0001 이므로, 2번만에 정상적인 상태 시퀀스로 진입
- 1111 경우 : 1111 → 0000 → 0001 이므로, 1번만에 정상적인 상태 시퀀스로 진입

8.7
0001 1001 1001 → 0010 0000 0000 이 되므로, 모두 6개의 플립-플롭들의 상태들이 변화된다.

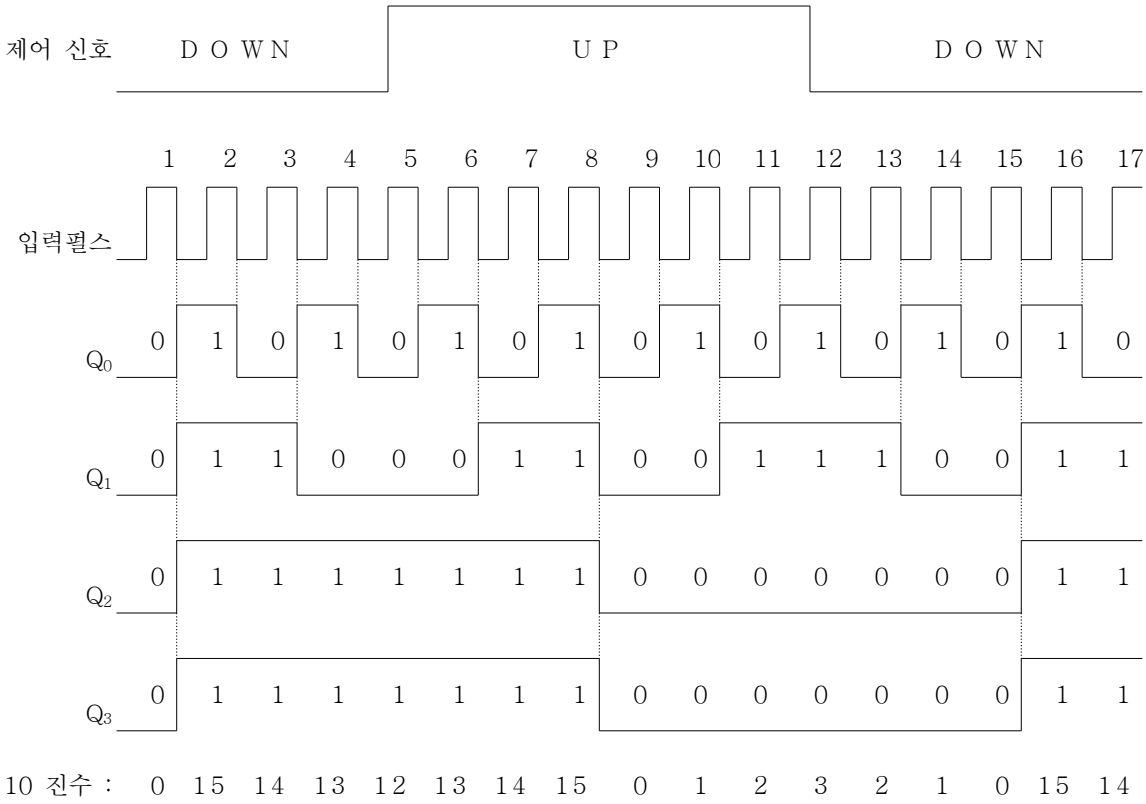
8.8


1 2 3 4 5 6 7 8 9
입력펄스
Q0 0 1 0 1 0 1 0 1 0
Q1 0 0 1 1 0 0 1 1 0
Q2 0 0 0 0 1 1 1 1 0

8.9

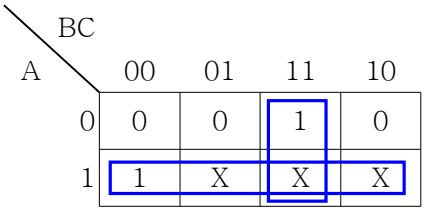

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
입력펄스
Q0 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0
Q1 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0
Q2 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0
Q3 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0

8.10

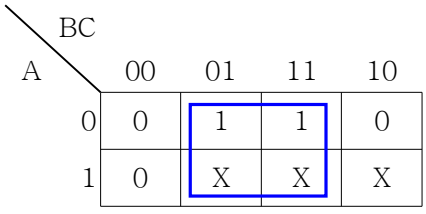


8.11

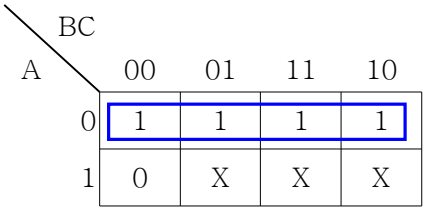
현재 상태			다음 상태			플립-플롭 입력들		
A	B	C	A	B	C	T_A	T_B	T_C
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	0	0	0	1	0	0



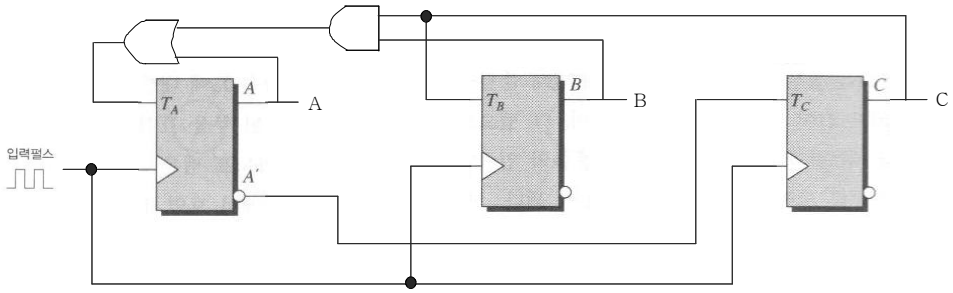
T_A=A+ BC



T_B=C



T_C=A´



8.12

현재 상태			다음 상태			플립-플롭 입력들					
A	B	C	A	B	C	J_A	K_A	J_B	K_B	J_C	K_C
0	0	0	0	0	1	0	X	0	X	1	X
0	0	1	0	1	0	0	X	1	X	X	1
0	1	0	0	1	1	0	X	X	0	1	X
0	1	1	1	0	0	1	X	X	1	X	1
1	0	0	0	0	0	X	1	0	X	0	X

A \ BC	00	01	11	10
0	0	0	1	0
1	X	X	X	X

$J_A=BC$

A \ BC	00	01	11	10
0	X	X	X	X
1	1	X	X	X

$K_A=1$

A \ BC	00	01	11	10
0	0	1	X	X
1	0	X	X	X

$J_B=C$

A \ BC	00	01	11	10
0	X	X	1	0
1	X	X	X	X

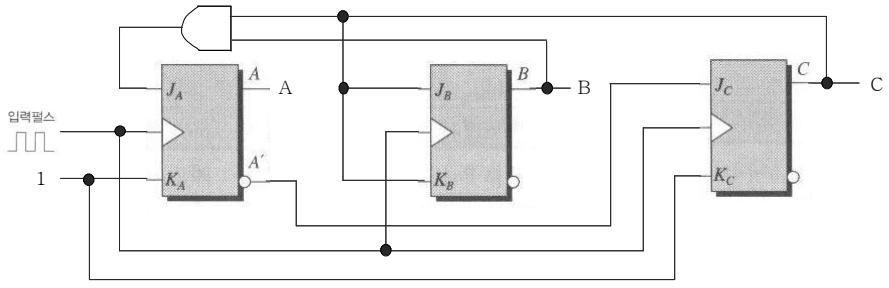
$K_B=C$

A \ BC	00	01	11	10
0	1	X	X	1
1	0	X	X	X

$J_C=A'$

A \ BC	00	01	11	10
0	X	1	1	X
1	X	X	X	X

$K_C=1$

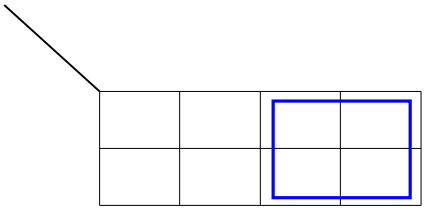


- JK 플립-플롭으로 구성하는 경우에, 회로 구성에 필요한 게이트 수가 하나 줄었다.

8.13

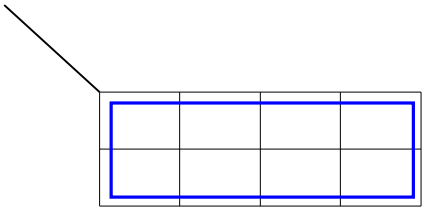
현재 상태			다음 상태			플립-플롭 입력들					
A	B	C	A	B	C	T_A	T_B	T_C			
0	0	0	0	1	0	0	1	0			
0	1	0	1	0	0	1	1	0			
1	0	0	1	0	1	0	0	1			
1	0	1	1	1	1	0	1	0			
1	1	1	0	0	0	1	1	1			

A \ BC	00	01	11	10
0	0	X	X	1
1	0	0	1	X



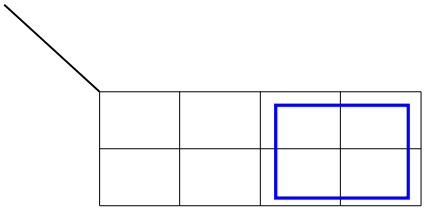
$T_A=B$

A \ BC	00	01	11	10
0	1	X	X	1
1	0	1	1	X



$T_B=A'+C$

A \ BC	00	01	11	10
0	0	X	X	0
1	1	0	1	X



$J_C=AC'+BC$

- JK 플립-플롭을 이용한 회로보다 더 복잡해진다.

8.14

현재 상태			다음 상태			플립-플롭 입력들		
A	B	C	A	B	C	T_A	T_B	T_C
0	0	0	0	0	1	0	0	1
0	0	1	0	1	1	0	1	0
0	1	1	0	1	0	0	0	1
0	1	0	1	1	0	1	0	0
1	1	0	1	1	1	0	0	1
1	1	1	1	0	1	0	1	0
1	0	1	1	0	0	0	0	1
1	0	0	0	0	0	1	0	0

A \ BC	00	01	11	10
0	0	0	0	1
1	1	0	0	0

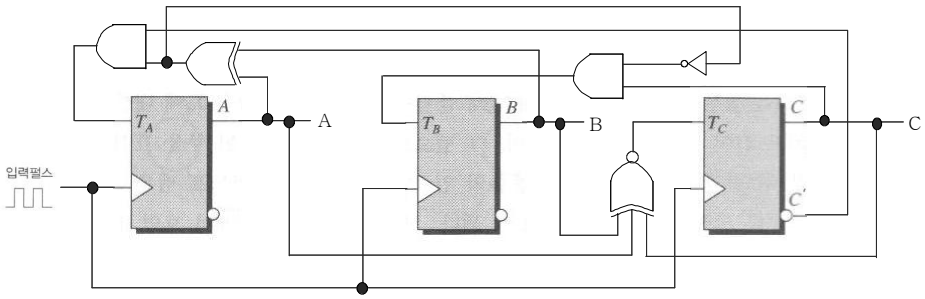
$T_A=AB'C'+A'BC'=C'(A\oplus B)$

A \ BC	00	01	11	10
0	0	1	0	0
1	0	0	1	0

$T_B=A'B'C+ABC=(A\oplus B)'C$

A \ BC	00	01	11	10
0	1	0	1	0
1	0	1	0	1

$T_C=(A\oplus B\oplus C)'$



- 회로의 복잡도가 더 낮아진다.

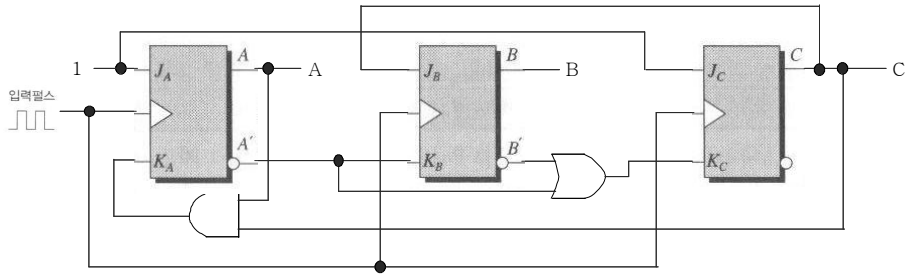
8.15

현재 상태			다음 상태			플립-플롭 입력들					
A	B	C	A	B	C	J_A	K_A	J_B	K_B	J_C	K_C
0	1	1	1	0	0	1	X	X	1	X	1
1	0	0	1	0	1	X	0	0	X	1	X
1	0	1	1	1	0	X	0	1	X	X	1
1	1	0	1	1	1	X	0	X	0	1	X
1	1	1	0	1	1	X	1	X	0	X	0

$J_A=1$ $K_A=BC$

$J_B=C$ $K_B=A'$

$J_C=1$ $K_C=A'+B'$



8.16

현재 상태				다음 상태				플립-플롭 입력들			
A	B	C	D	A	B	C	D	T_A	T_B	T_C	T_D
0	0	0	0	0	0	0	1	0	0	0	1
0	0	0	1	0	0	1	0	0	0	1	1
0	0	1	0	0	0	1	1	0	0	0	1
0	0	1	1	0	1	0	0	0	1	1	1
0	1	0	0	0	1	0	1	0	0	0	1
0	1	0	1	0	1	1	0	0	0	1	1
0	1	1	0	0	1	1	1	0	0	0	1
0	1	1	1	1	0	0	0	1	1	1	1
1	0	0	0	1	0	0	1	0	0	0	1
1	0	0	1	0	0	0	0	1	0	0	1

A \ BC	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	X	X	X	X
10	0	1	X	X

T_A=AD+ BCD

A \ BC	00	01	11	10
00	0	0	1	0
01	0	0	1	0
11	X	X	X	X
10	0	0	X	X

T_B=CD

A \ BC	00	01	11	10
00	0	1	1	0
01	0	1	1	0
11	X	X	X	X
10	0	0	X	X

T_C=A′D

A \ BC	00	01	11	10
00	1	1	1	1
01	1	1	1	1
11	X	X	X	X
10	1	1	X	X

T_D=1

- 8.3.4절의 입력 함수들과 일치한다.

8.17

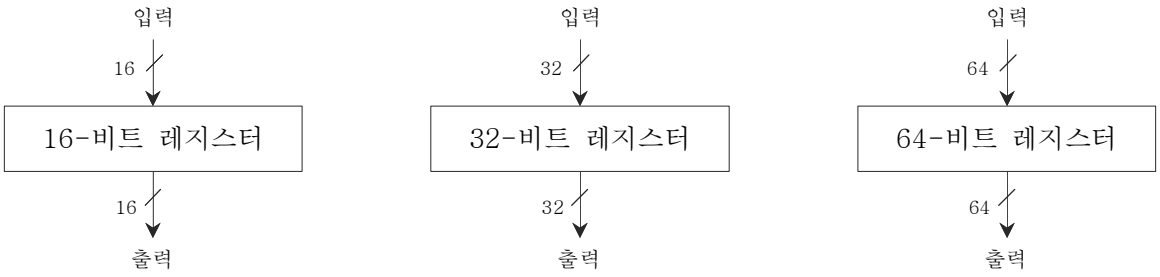
현재 상태				다음 상태				플립-플롭 입력들			
A	B	C	D	A	B	C	D	D_A	D_B	D_C	D_D
0	0	0	0	0	0	0	1	0	0	0	1
0	0	0	1	0	0	1	0	0	0	1	0
0	0	1	0	0	0	1	1	0	0	1	1
0	0	1	1	0	1	0	0	0	1	0	0
0	1	0	0	0	1	0	1	0	1	0	1
0	1	0	1	0	1	1	0	0	1	1	0
0	1	1	0	0	1	1	1	0	1	1	1
0	1	1	1	1	0	0	0	1	0	0	0
1	0	0	0	1	0	0	1	1	0	0	1
1	0	0	1	0	0	0	0	0	0	0	0

$T_A = AD' + BCD$ $T_B = BC' + BD' + B'CD$

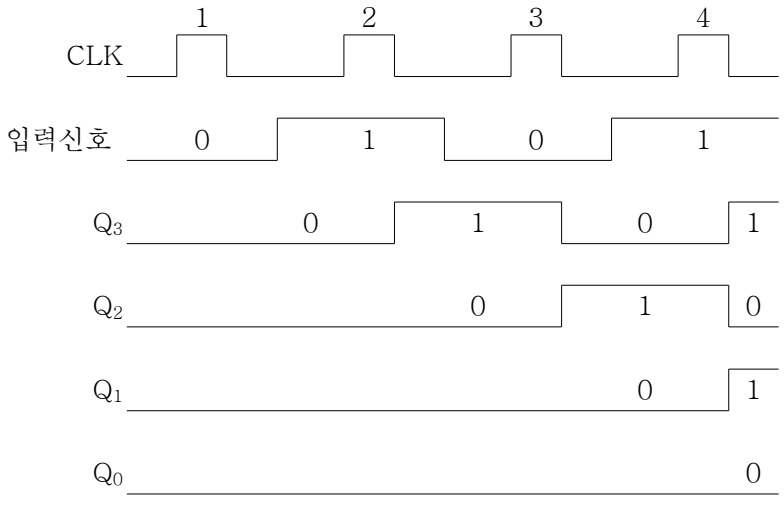
$T_C = CD' + A'C'D$ $T_D = D'$

- JK 플립-플롭을 사용하여 회로를 구성하는 것이 가장 효율적이다.

8.18



8.19



8.20

(1) 우측 시프트 - 입력 순서: 0, 1, 0, 1

CLK 주기 \ 데이터	d_3	d_2	d_1	d_0
초기 상태	0	0	0	0
t_1	0	0	0	0
t_2	1	0	0	0
t_3	0	1	0	0
t_4	1	0	1	0

(2) 좌측 시프트 - 입력 순서: 1, 0, 1, 0

CLK 주기 \ 데이터	d_3	d_2	d_1	d_0
초기 상태	0	0	0	0
t_1	0	0	0	1
t_2	0	0	1	0
t_3	0	1	0	1
t_4	1	0	1	0

8.21

(1) 우측 시프트: 10110101 → 01011010

(2) 좌측 시프트: 10110101 → 01101010

8.22

(1) 우측 시프트: 10110101 → 01011010 → 00101101

(2) 좌측 시프트: 10110101 → 01101010 → 11010100

8.23

	A_4	A_3	A_2	A_1	B_4	B_3	B_2	B_1
초기 상태	0	1	0	1	1	1	1	1
t_1	1	0	1	0	1	1	1	1
t_2	0	1	0	1	0	1	1	1
t_3	1	0	1	0	1	0	1	1
t_4	0	1	0	1	0	1	0	1

8.24 B 레지스터의 최상위 비트는 A 레지스터와 B 레지스터의 최하위 비트에 인가되어 두 레지스터가 모두 좌측 시프트 동작을 한다.

	A_4	A_3	A_2	A_1	B_4	B_3	B_2	B_1
초기 상태	0	0	0	0	1	0	1	0
t_1	0	0	0	1	0	1	0	1
t_2	0	0	1	0	1	0	1	0
t_3	0	1	0	1	0	1	0	1
t_4	1	0	1	0	1	0	1	0