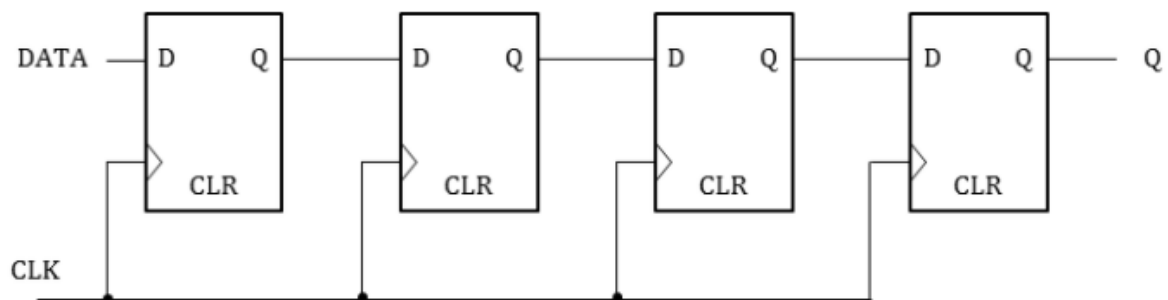


## 1. Shift register에 대해 조사하시오.

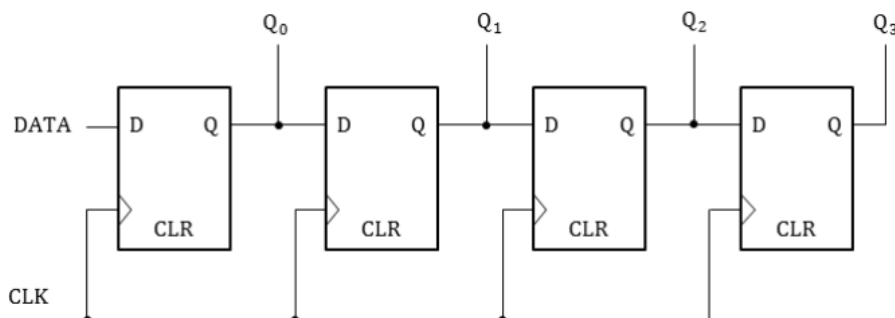
1비트씩 저장할 수 있는 플립플롭 여러 개를 일렬로 배열해서 적당히 연결함으로써 여러 비트로 구성된 2진수를 저장할 수 있게 한 것을 레지스터라고 한다. 시프트 레지스터는 디지털 회로에서 선형 방식으로 설치된 프로세서 레지스터의 집합으로 데이터를 저장하거나 데이터를 옆으로 이동할 때 사용된다. 레지스터의 종류에는 SIPO, SISO, PISO, PIPO가 있다.

SISO(직렬입력-직렬출력) 시프트 레지스터는 입력 데이터를 그대로 저장하고 있다가 CLK에 따라서 플립플롭을 거치면서 시간 delay를 거쳐서 출력되는 회로를 말한다. 이렇듯 클럭이 입력될 때마다 클럭의 하강엣지에서 입력 데이터가 한 비트씩 오른쪽으로 시프트 하면서 저장되게 된다. 이는 새로운 클럭펄스의 하강엣시마다 반복되기 때문에 네번째 클럭펄스의 하강엣지에서 Qd에 처음에 입력된 데이터 비트가 나타나게 된다.

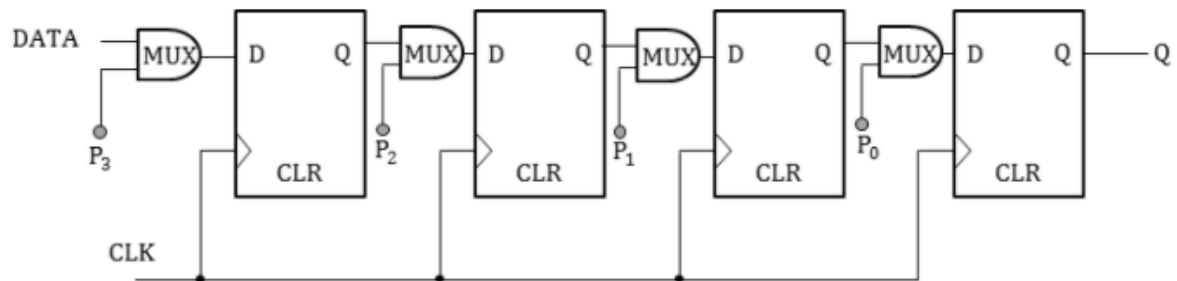


데이터가 1,1,0,1로 순차적으로 입력된다면 네 번째 클럭펄스 이후부터 출력 Qd에 입력 순서대로 비트가 순차적으로 나타나게 된다.

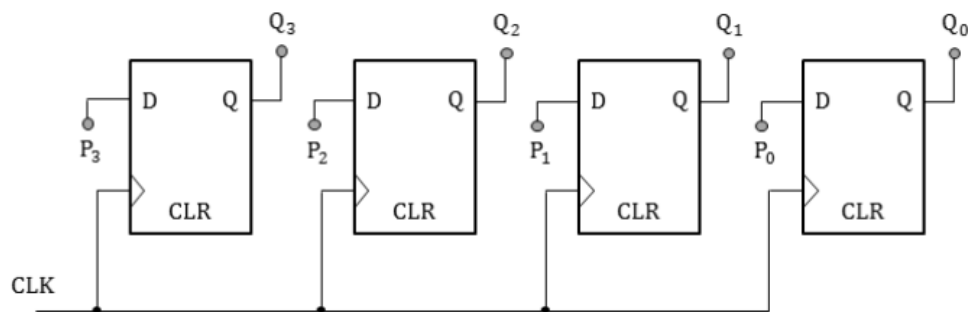
SIPO(직렬입력-병렬출력) 시프트 레지스터는 단일의 데이터가 시프트 레지스터를 거쳐서 다수의 출력으로 나타나는 구조를 말한다. 이는 SISO 레지스터와 기본적으로 동일한 구조를 가지고 있으며 각 출력에 3상태 버퍼가 연결된 점이 다르다. 처음에 모든 플립플롭의 출력Q를 0이라고 가정하면, 클럭펄스의 하강엣지에서 입력 데이터가 한 비트씩 오른쪽으로 시프트되면서 저장되게 된다. 4비트 레지스터는 저장된 클럭펄스 4개가 입력되면 4비트 직렬입력 데이터가 레지스터에 모두 저장되게 된다. 레지스터에 저장된 데이터를 출력하면 새로운 4비트 데이터가 레지스터에 차게 되는 4,8,12번째 클럭펄스 등에서 3상태 버퍼를 인에이블해서 동시게 읽어내게 된다.



PISO(병렬입력 직렬출력)은 SIPO 시프트 레지스터와 반대로 다수의 입력 데이터를 한 개의 출력으로 내보내는 회로를 말한다. 4개의 데이터를 병렬로 입력하고 시프트하기 위한 제어신호가 존재한다. 2X1 MUX 동작을 살펴보면  $S=0$ 이면 입력 A와 출력 F가 연결되게 되고  $S=1$ 이면 입력 B와 출력 F가 연결되게 된다.

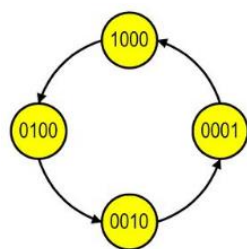


PIPO(병렬입력-병렬출력)은 SISO 시프트 레지스터와 마찬가지로 입력 데이터를 저장하고 있다가 CLK에 따라서 플립플롭을 거치면서 시간지연을 거쳐서 출력된다. SISO와는 다수 입력 대 다수 출력이라는 특징이 있다. WR 신호와 병렬출력을 제어하는  $\overline{RD}$ 에서  $\overline{RD}$ 가 0이면 각 플립플롭의 출력 데이터는 3상태 버퍼를 통해서 동시에  $O_a, O_b, O_c, O_d$ 에 출력되고, 1이면 출력되지 않는다. 즉,  $\overline{RD}$ 가 0일 때는 3상태 버퍼가 플립플롭의 출력Q를 레지스터의 출력  $O_a, O_b, O_c, O_d$ 에 연결해주고  $\overline{RD}$ 가 1일때는 양자 간 연결이 완전히 끊어지게 된다.



## 2. Ring Counter에 대해서 조사하시오.

링 카운터는 출력 중 한 비트만이 1이 되고 입력 펄스에 따라서 1의 값을 갖는 비트의 위치가 한 쪽 방향으로 순환되는 카운터이다.



상태표를 그려보면 아래와 같다.

현재상태				다음상태				플립플롭 입력			
Qa	Qb	Qc	Qd	Qa	Qb	Qc	Qd	Da	Db	Dc	Dd
1	0	0	0	0	1	0	0	0	1	0	0
0	1	0	0	0	0	1	0	0	0	1	0
0	0	1	0	0	0	0	1	0	0	0	1
0	0	0	1	1	0	0	0	1	0	0	0

이 상태표를 이용해서 카르노 맵을 통해서 간소화 시킬 수 있다.

		$Q_C Q_D$			
		00	01	11	10
$Q_A Q_B$	00	X	1	X	
	01		X	X	X
	11	X	X	X	X
	10		X	X	X

$D_A = Q_D$

		$Q_C Q_D$			
		00	01	11	10
$Q_A Q_B$	00	X		X	
	01		X	X	X
	11	X	X	X	X
	10	1	X	X	X

$D_B = Q_A$

		$Q_C Q_D$			
		00	01	11	10
$Q_A Q_B$	00	X		X	
	01	1	X	X	X
	11	X	X	X	X
	10		X	X	X

$D_C = Q_B$

		$Q_C Q_D$			
		00	01	11	10
$Q_A Q_B$	00	X		X	1
	01		X	X	X
	11	X	X	X	X
	10		X	X	X

$D_D = Q_C$

링 카운터는 앞단 플립플롭의 출력이 다음 단 플립플롭의 입력으로 연결되는 과정이 반복되고, 최종단 플립플롭의 출력은 맨 앞단 플립플롭의 입력으로 연결되는 구조를 가지고 있다.

처음에 Clear가 0이면 첫 번째 플립플롭의 출력 Qa는 1로 셋팅되고 나머지 플립플롭의 출력은 QbQcQd=000이 된다. Clear가 1일 경우는 링 카운터의 최초 출력은 QaQbQcQd = 1000이 된다. 이후부터 클럭펄스가 입력될 때마다 클럭펄스의 상승엣지에서 오른쪽으로 한 자리씩 이동하게 되고, 출력 Qd는 다시 Da로 입력되게 된다. 아래 회로도도 D 플립플롭을 이용해서 구현한 것이며 JK, SR 플립플롭을 이용해서 구현할 수도 있다. SR 플립플롭을 사용할 때는 맨 오른쪽 플립플롭의 출력 Q와  $\bar{Q}$ 를 맨 왼쪽 플립플롭의 S와 R에 연결하면 된다. 또한 JK 플립플롭을 사용할 때는 맨 오른쪽 플립플롭의 출력 Q와  $\bar{Q}$ 를 맨 왼쪽 플립플롭의 J와 K에 연결하면 된다.

링 카운터는 일련의 동작을 제어하는데 매우 유용하기 때문에 커피 자판기 등을 제어하는데 유용하게 사용된다.

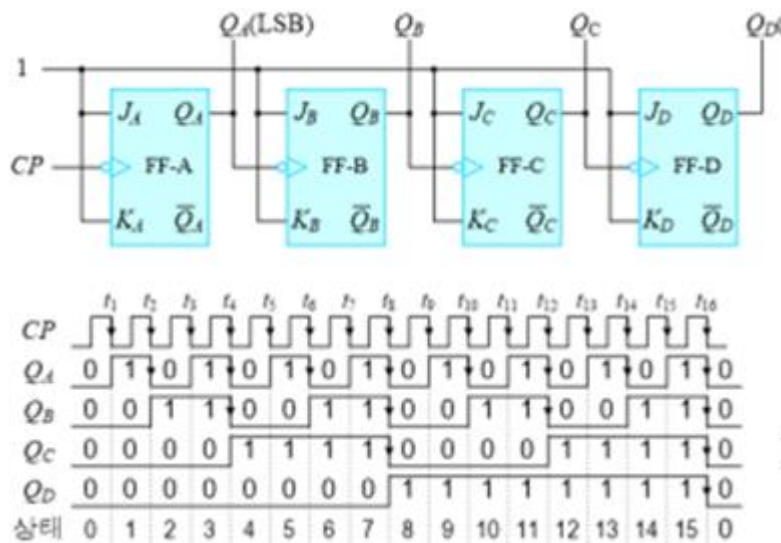
### 3. UP DOWN Counter에 대해서 조사하시오.

UP Counter는 상향 카운터로 수를 세어 올라가는 카운터를 말한다.

플립플롭 4개를 사용한 16진 카운터의 계수상태를 아래의 표를 통해서 살펴보면, 2진수의 4자리를 이용해서 0000부터 1111까지 올라가게 된다. Qd열은 8에 해당하는 자리로서 최상위 비트이며 Qa열은 1에 해당하는 자리로 최하위비트이다. 0000에서 1111까지 카운트하며 카운트하는 상태의 수가 16개 이므로 16진(mod-16) 카운터라고 한다.

클럭펄스	Qd	Qc	Qb	Qa	10진수
1	0	0	0	0	0
2	0	0	0	1	1
3	0	0	1	0	2
4	0	0	1	1	3
5	0	1	0	0	4
6	0	1	0	1	5
7	0	1	1	0	6
8	0	1	1	1	7
9	1	0	0	0	8
10	1	0	0	1	9
11	1	0	1	0	10
12	1	0	1	1	11
13	1	1	0	0	12
14	1	1	0	1	13
15	1	1	1	0	14
16	1	1	1	1	15

JK플립플롭을 사용한 4비트 비동기식 상향 카운터를 살펴보면, 모든 플립플롭의 입력은 J=K=1이므로 출력은 토글 상태로 동작하게 된다. 모든 플립플롭의 출력이 0000으로 되어 있다고 가정하면 각 플립플롭은 클럭펄스의 하강엣지에서 변화하게 된다. T1에서 플립플롭 FF-A는 출력을 반전해서 QaQcQbQa=0001 이 되고 t2에서 FF-A의 출력은 다시 0으로 되고 이것은 즉시 FF-B 클럭에 인가되어 FF-B의 출력은 1이 되어서 QdQcQbQa = 0010 이 된다. 계속해서 클럭펄스가 인가되고 각 플립플롭의 출력은 다음 플립플롭의 입력으로 들어가서 해당 플립플롭의 출력을 연쇄적으로 반전시키게 되며 각 플립플롭은 2진수로 0000에서 1111까지 카운트하게 된다.

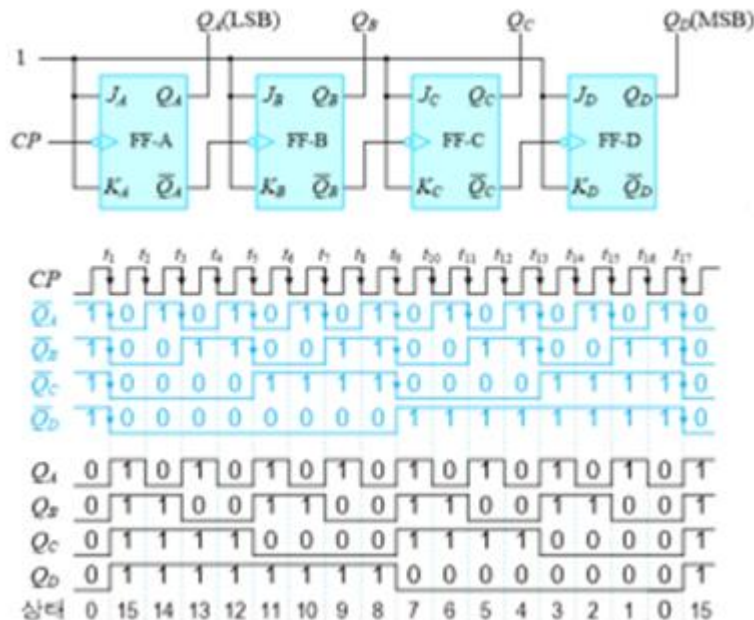


Down Counter는 클럭에 not 게이트가 없다는 점에서 다르다. 4비트 비동기식 상향 카운터에서 각 플립플롭의  $\bar{Q}$  출력을 다음 단의 클럭 입력으로 연결하면 하향 카운터가 된다. 카운터는 1111에서 시작해서 15번째 클럭펄스의 끝에서 0000으로 감소하게 된다.

클럭펄스	Qd	Qc	Qb	Qa	10진수
1	1	1	1	1	15
2	1	1	1	0	14
3	1	1	0	1	13
4	1	1	0	0	12
5	1	0	1	1	11
6	1	0	1	0	10
7	1	0	0	1	9
8	1	0	0	0	8
9	0	1	1	1	7
10	0	1	1	0	6
11	0	1	0	1	5
12	0	1	0	0	4
13	0	0	1	1	3
14	0	0	1	0	2
15	0	0	0	1	1
16	0	0	0	0	0

모든 플립플롭의 입력은  $J=K=1$  이기 때문에 출력은 토글 상태로 동작하게 되며 각 플립플롭은 2진수로 1111에서 0000까지 카운트하게 된다. 이의 설계방법은 4비트 비동기식 상향 카운터와 같다.

처음 모든 플립플롭의 출력이  $Q_d Q_c Q_b Q_a = 0000$ 이라고 가정하면  $\overline{Q_d} \overline{Q_c} \overline{Q_b} \overline{Q_a} = 1111$  이다. 출력  $Q_a$ 는 클록펄스의 하강엣지에서 상태를 반전하게 된다. 출력  $Q_b$ 는 맨 앞단 플립플롭의  $\overline{Q_a}$ 가 1에서 0으로 변할 때 마다 상태를 반전하게 된다. 출력  $Q_c$ 는 앞단 플립플롭의  $\overline{Q_b}$ 가 1에서 0으로 변할 때 마다 상태를 반전하며, 출력  $Q_d$ 는 앞단 플립플롭의  $\overline{Q_c}$ 가 1에서 0으로 변할 때 마다 상태를 반전한다.



#### 4. Ripple Counter에 대해서 조사하시오.

리플 카운터는 비동기식 카운터와 같은 말이다. 비동기식 카운터는 첫 번째 플립플롭의 CP입력에 만 클록펄스가 입력되고 다른 플립플롭의 각 플립플롭의 출력을 다음 플립플롭의 CP입력으로 사용한다. 즉 첫 단 플립플롭의 출력이 다음 단계의 플립플롭을 트리거하게 된다. 이 때 클록의 영향이 물결처럼 뒷단으로 파급된다는 뜻에서 리플 카운터로 불린다. 리플 카운터는 각 단의 플립플롭들의 상태가 동시에 변하지 않고 각 플립플롭을 통과할 때 마다 지연시간이 누적되기 때문에 고속 계수에는 적당하지 않다.

비동기식 카운터는 JK 플립플롭을 이용해서 구성하는데 모든 J와 K의 입력을 1로 해서 토글 상태가 되도록 한다. 비동기식 카운터는 출력을 거치면서 전달 지연시간이 발생하게 되고 이 지연시간은 플립플롭의 개수 \* 플립플롭의 지연시간으로 나타낼 수 있다.