

## 13주차 예비보고서

전공: 신문방송학과

학년: 4학년

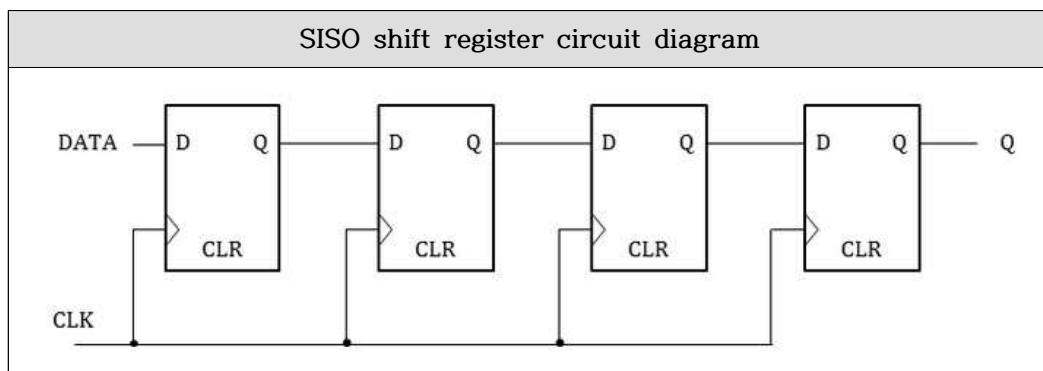
학번: 20191150

이름: 전현길

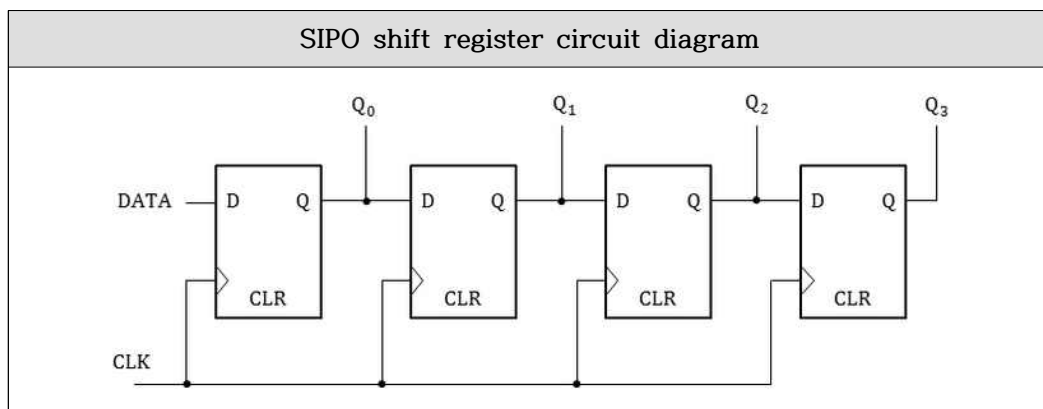
### 1. Shift Register에 대해서 조사하시오. (예시 포함)

flip-flop이 1개의 bit를 저장하는 소자라면, register는 여러 개의 bit를 저장하는 소자를 가리킨다. register의 종류로 register, shift register, counter 등이 있는데, flip-flop을 연결하는 방식에 따라 서로 다른 기능을 갖는다.

shift register는 데이터를 저장하고, 이동시킬 때 사용되는 회로이다. 일반적으로 D형 data latch로 구성되며, 구현 방식에 따라 SISO, SIPO, PISO, PIPO로 나눌 수 있다. 이 때 S는 Serial, I는 In, P는 Parallel, O는 Out을 의미한다.

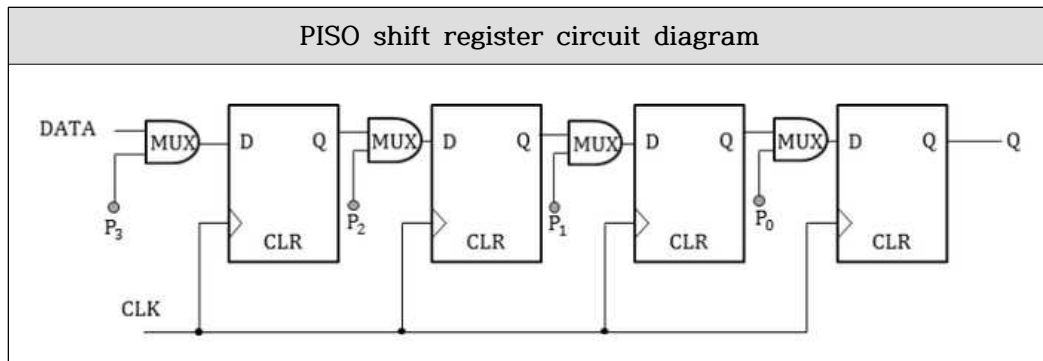


SISO shift register는 클럭 신호에 따라 왼쪽 또는 오른쪽 방향으로 한 번에 한 비트씩 데이터가 In, Out 신호에 따라 직렬로 이동한다. 사용할 수 있는 데이터는 마지막으로 출력되는 1bit이다.

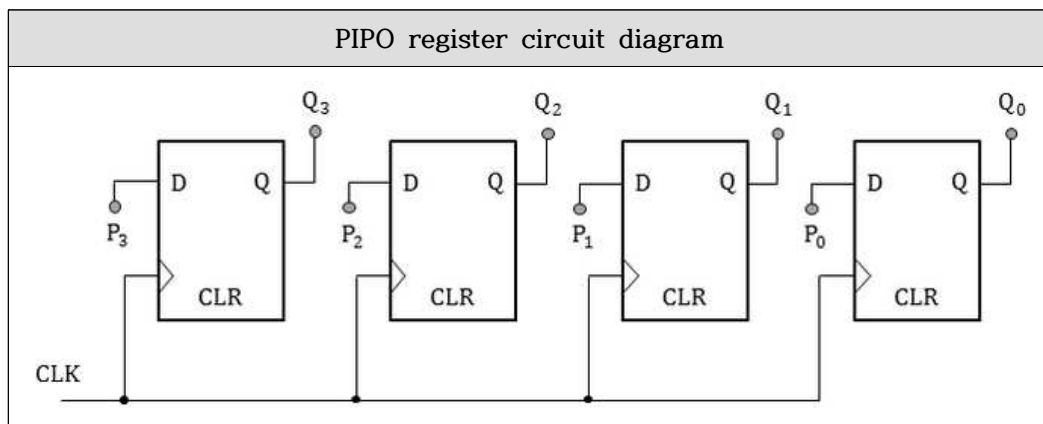


SIPO shift register에서는 한 번에 한 비트씩 직렬 데이터가 로드되며,

저장된 데이터를 즉시 출력으로 사용해 병렬로 데이터를 사용할 수 있다.



PISO shift register에서는 레지스터에 값이 동시에 로드되고, 클럭 시그널에 따라 한 번에 1bit씩 데이터가 이동한다. 사용할 수 있는 데이터는 마지막으로 출력되는 1bit이다.



PIPO shift register에서는 레지스터에 값이 동시에 로드되고, 클럭 시그널에 따라 동일한 클럭 펄스에 의해 출력이 한꺼번에 전송된다. 사용할 수 있는 데이터는 시프트 레지스터의 bit sequence 길이만큼이다. 회로가 다음 D flip-flop의 D 입력에 결선되어 있지 않은 것으로 볼 수 있듯이 PIPO register는 shift register로 분류되지는 않지만, 한 클럭 펄스 내에 출력을 제공하므로 매우 빠른 속도를 갖는다.

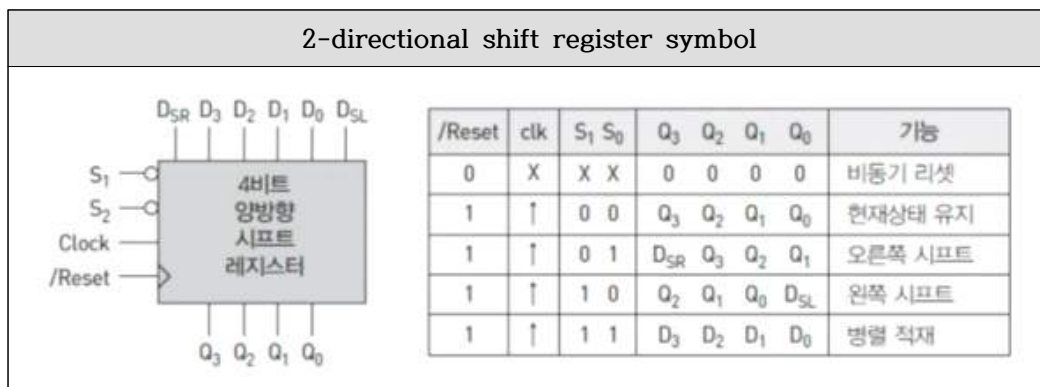
shift register의 실제 사용례는 다음과 같다.

1) 직렬 인터페이스와 병렬 인터페이스를 상호 전환한다.

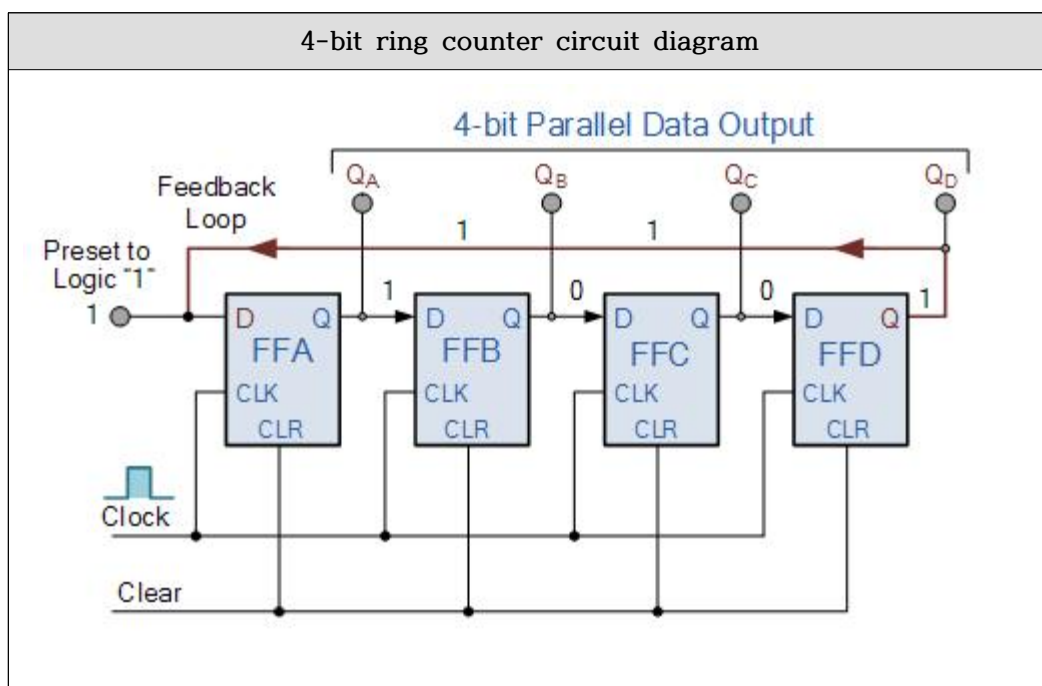
2) delay를 주기 위해 사용될 수 있다. 예를 들어, 임의의 gate에 들어와야 할 두 입력이 서로 다른 타이밍에 들어와 버그가 생길 수 있다. 이 때 더 빠르게 입력되는 데이터를 n-bit shift register에 입력해 n clock pulse만큼 delay를 줄 수 있다.

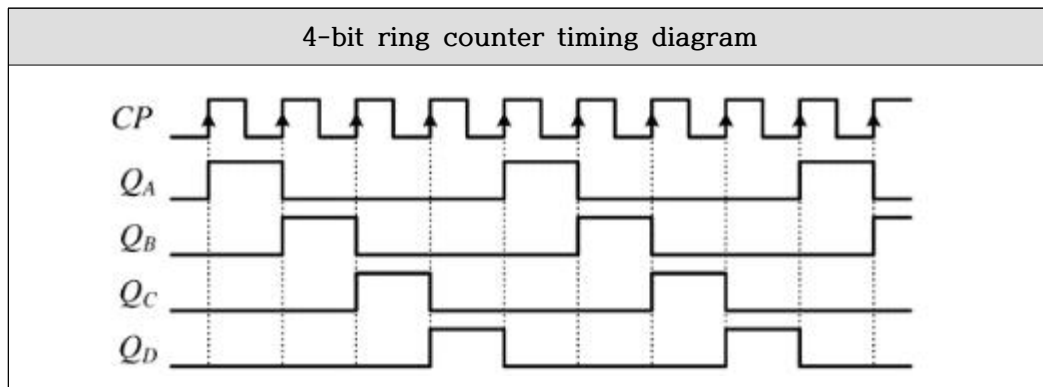
3) CPU에서 bit-sequence data에 2를 곱하거나 2로 나누는 **bit-shift** 연산을 위해 사용된다.

4) 양방향 shift register의 경우 **스택 자료구조를 구현**하는 데에 사용되기도 한다. 값이 push되었을 경우 새로운 입력을 추가하고 값들을 오른쪽으로 shift하고, pop되었을 경우 값을 왼쪽으로 shift한 뒤 top에 있는 데이터는 출력하는 것으로 스택을 구현할 수 있다.



## 2. Ring Counter에 대해서 조사하시오.





ring counter는 shift register를 응용해서 구현할 수 있는 카운터로, shift register의 마지막 출력을 다시 첫 번째 플립 플롭에 입력하는 것으로 간단히 구현된다. feedback loop를 구현함으로써 ring counter는 동일한 data bit를 끊임없이 순환시키는 동작을 갖게 된다. 값을 오른쪽으로 shift하는 임의의 4b it ring counter의 초기 bit가 1011이었다면,  $1011 \rightarrow 1101 \rightarrow 1110 \rightarrow 0111 \rightarrow 1011$ 의 순환을 끊임없이 반복하게 될 것이다.

위와 같은 4bit ring counter는 4개의 개별 상태가 존재하기 때문에, modulo-4 counter, mod-4 counter라고 부르기도 한다. 응용하여 8bit, 16bit ring counter는 mod-8, mod-16 ring counter라고 부르기도 한다. mod-n ring counter의 모든 플립플롭의 출력주파수는 클럭 주파수의  $1/n$ 의 값을 가진다.

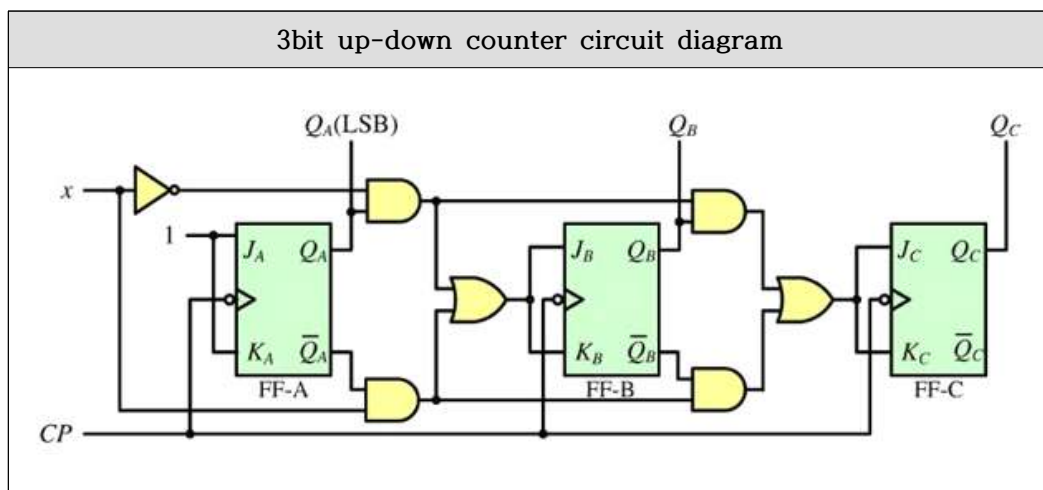
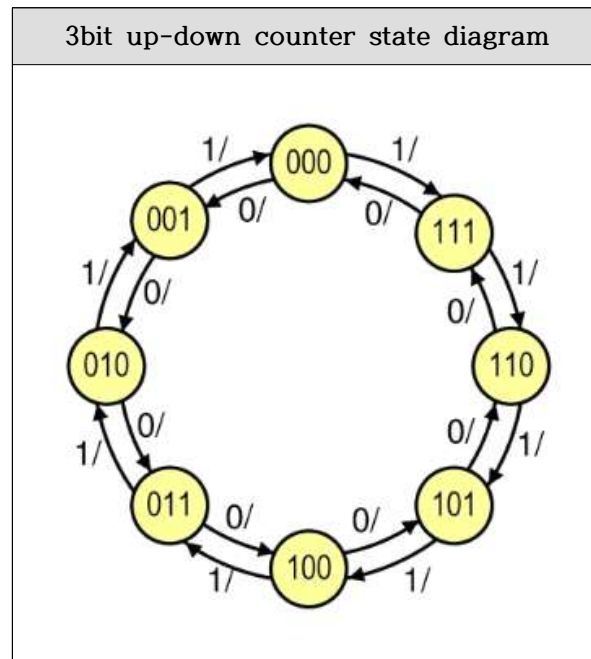
이러한 mod-n ring counter는 가능한  $2^n$ 가지의 출력 중에서 n가지의 출력만을 실제로 사용하기 때문에 출력 상태를 사용하는 측면에선 비효율적이다. 뿐만 아니라, timing diagram을 통해 볼 수 있듯이 mod-n ring counter의 출력 주파수는 clock pulse가 High인 경우나 Low인 경우가 불균형해진다. 하지만 입력을 1100 등으로 처음 입력해 균형 있게 동작하도록 하면 출력 주파수가 클럭 주파수의  $1/2$ 의 값만을 갖게 된다.

이런 mod-n ring counter의 문제를 해결한 것이 johnson ring counter로, Q 출력이 첫 번째 플립플롭으로 입력되는 것이 아니라  $\bar{Q}$  출력이 첫 번째 플립플롭으로 입력되도록 해 문제를 해결한다. 이를 바탕으로 아래와 같은 진리표를 갖는다.

클럭 펄스 번호	FFA	FFB	FFC	FFD
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1
5	0	1	1	1
6	0	0	1	1
7	0	0	0	1

이 결과로 johnson ring counter는  $2n$ 가지의 상태를 순회하게 되어 각 출력 bit들의 출력 주파수가 클럭 주파수의  $1/2n$ 이 되게 할 수 있고, 출력 펄스가 high일 때와 low일 때 역시 균등하게 할 수 있다.

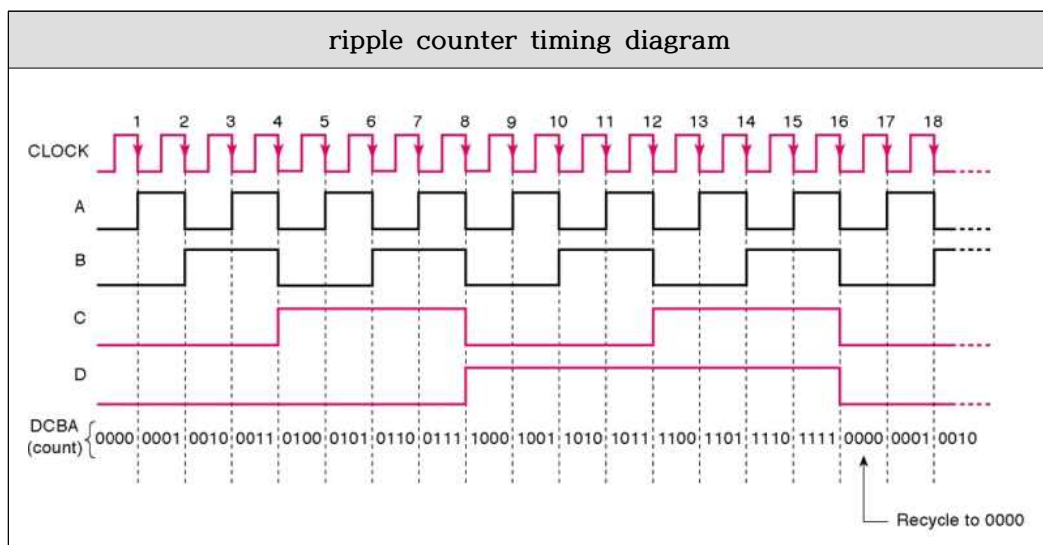
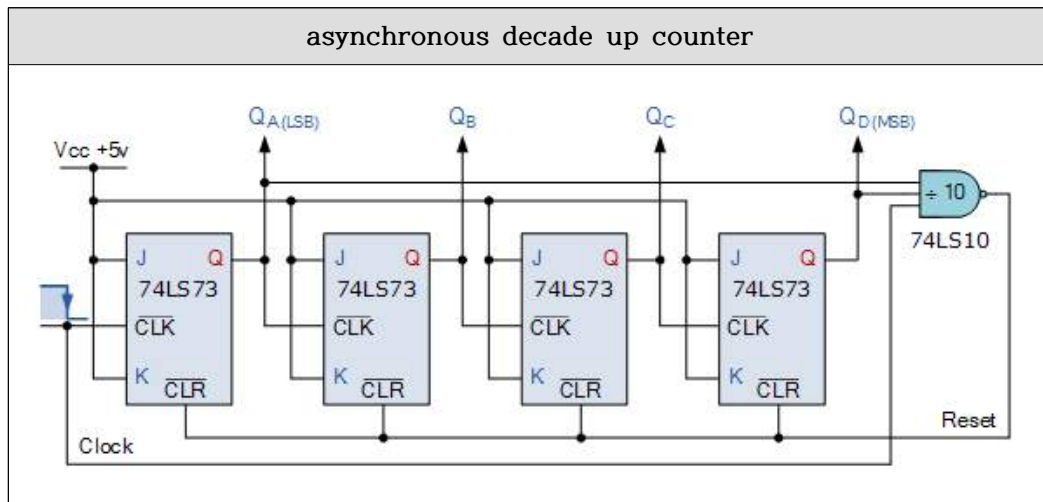
### 3. Up Down Counter에 대해서 조사하시오.



up-down counter는 값의 증가, 감소 모두가 가능한 카운터이다. 입력  $x$ 가 0일 때는 증가하고, 1일 때는 감소한다고 가정한다면 위와 같은 state diagram을 그릴 수 있다. 이를 바탕으로 현재 상태와 입력, 다음 상태를 표로 그리고, 이에 맞게 각 JK 플립플롭에 필요한 입력을 그린 뒤 카르노 맵을 작성하면 아래의 circuit diagram을 구성할 수 있다.

값을 유지할 경우도 필요하다면, 위 state diagram에 입력  $y$ 를 추가하여  $y$ 가 0일 경우 값이 유지되고,  $y$ 가 1일 경우 증가/감소하는 형식 등으로 구현할 수 있다.

#### 4. Ripple Counter에 대해서 조사하시오.



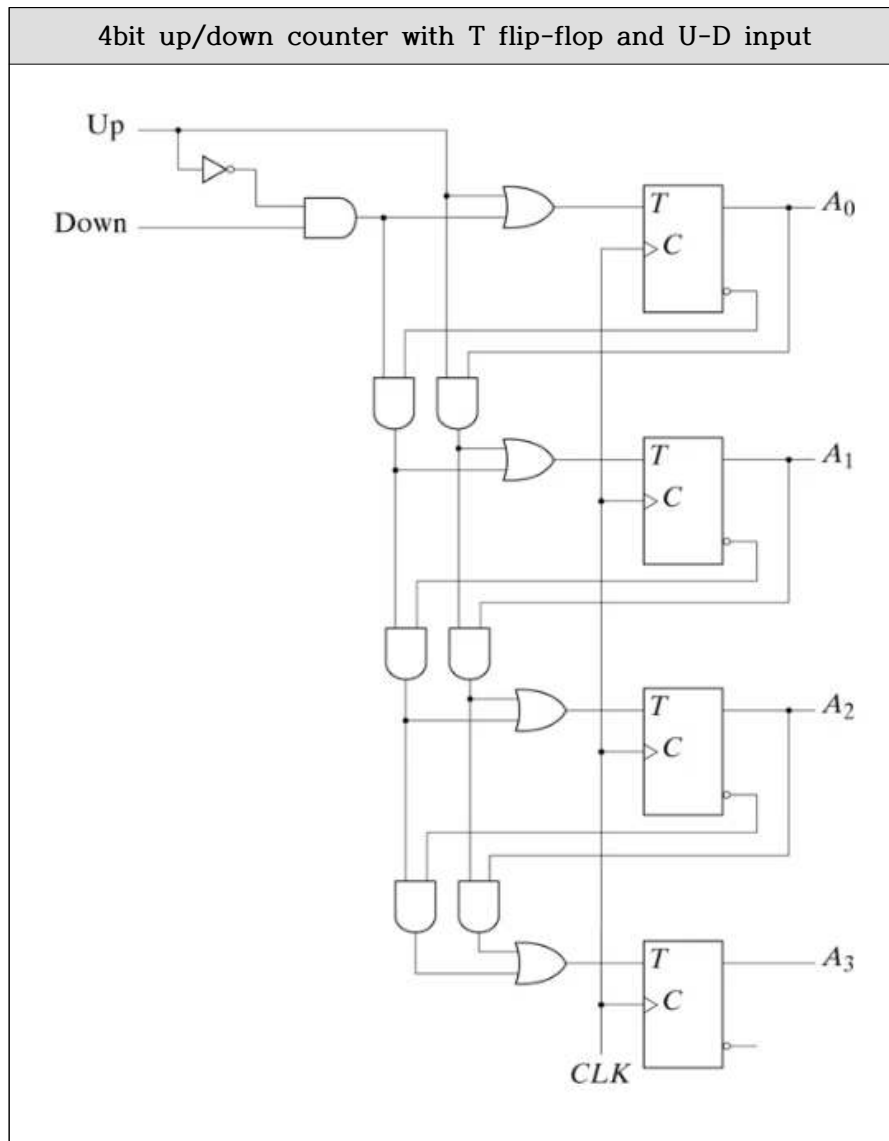
ripple counter는 asynchronous counter의 다른 호칭으로 첫 번째 카운터만 외부 클럭 신호에 의해 동작하고 나머지 플립플롭들은 전부 이전 플립플롭의 출력에 의해 구동되는 카운터를 의미한다. 어원은 이전 강의에서 많이 살펴보았던 ripple-carry adder 등과 동일하게, 물결이 퍼지듯 출력이 순차적으로 전송되는 것에서 유래했다.

위의 ripple decade up counter를 살펴보면, 첫 번째 플립플롭에만 clock signal이 입력되고 그 뒤의 플립플롭은 이전 카운터의 출력을 입력으로 받고 있는 것을 알 수 있다.

비동기식 카운터의 경우 이전 플립플롭의 출력이 결정되기 전에는 다음 플립플롭의 출력을 결정할 수 없으므로 플립플롭 1개당 전파지연이 축적된다. dec

ade counter의 경우 4bit만큼의 전파 지연만 발생하지만 bit 수가 클 경우 전파 지연은 점차 누적될 수 있다. 결과적으로 ripple counter의 마지막 출력 비트의 출력 주파수는 클럭 주파수의  $1/2^4$ 만큼이 된다.

## 5. 기타 이론



본문에서는 up/down counter을  $x = 0$ 일 때 증가하고  $x = 1$ 일 때 감소하도록 했고,  $y = 0$ 일 때 상태를 유지하고  $y = 1$ 일 때 상태를 변화시키도록 구현했지만 다른 방식으로 구현할 수도 있다.

위는 Up, Down 입력을 이용해 구현할 수 있는데,  $U = 1$ 일 때 값이 증가하고,  $D = 1$ 일 때 값이 감소하며,  $U = D = 0$ 일 때 값이 유지되는 카운터이다.



$U = D = 1$ 일 때의 출력을 정의하지 않지만, 입력값을 추상화시켜 이해할 수 있기 때문에 나름의 장단점이 있다.