전공: 컴퓨터공학 학년: 2학년 학번: 20161603 이름: 신민준

1. Shift register에 대해 조사하시오.

Flip-flop들은 1bit 짜리의 데이터를 보관하는데 사용된다. 따라서 여러 비트의 데이터를 저장하기 위해선 그만큼의 갯수의 flip-flop들이 필요하다. 예를 들어, n-bit의 데이터를 저장하는 데에는 n 개의 flip-flop들이 필요한 식이다. Register는 이러한 정보를 저장하는 데에 사용되는 장치이며, 여러 개의 flip-flop들이 순차적으로 연결되어 있는 구조를 가진다.

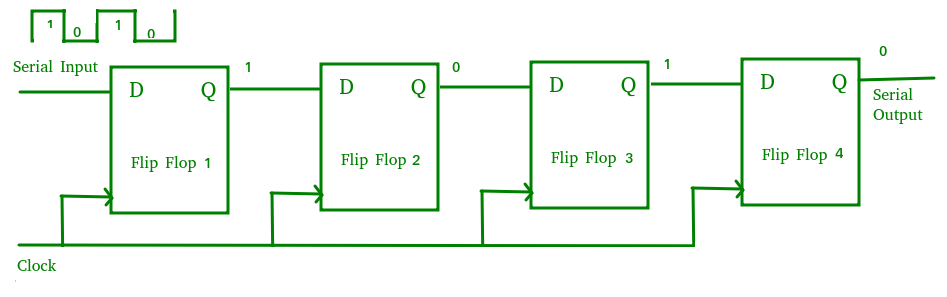
이러한 register들에 저장되어있는 정보를 이동하기 위해서는 shift register의 도움이 필요한데, 이 shift register 또한 다른 register들과 마찬가지로 여러 비트의 데이터를 저장하는데 사용하는 flip-flop들의 그룹이다. Shift register에 저장되어 있는 비트들은 register 안에서 움직이거나, 또는 clock 신호를 추가해 비트들을 register 안/밖으로 움직일 수도 있다. n-bit의 shift register는 마찬가지로 n 개의 flip-flop들을 연결시켜 만들 수 있다.

왼쪽으로 비트를 shift하는 register를 shift left register라 하고, 반대로 오른쪽으로 비트를 움직이는 register를 shift right register라 칭한다.

기본적으로 shift register들은 다음의 네 카테고리로 분류된다.

* Serial In Serial Out shift register(SISO)

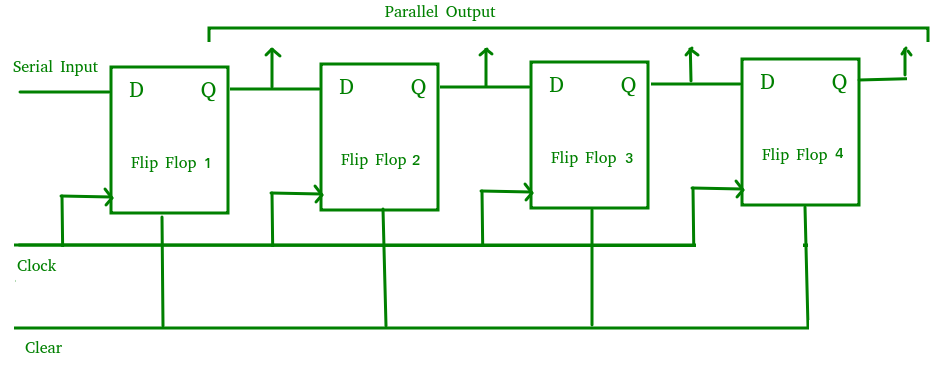
이 shift register는 하나의 입력 line에서 순차적 입력을 받고 하나의 출력 line으로 순차적 출력을 해주는 register이다. 출력이 단 하나만 존재하기 때문에, 데이터는 이 register에서 1비트 단위로 이동하며, 순차적으로 이동한다. 아래의 논리 회로는 이러한 SISO register를 보이고 있다.



회로에서는 4개의 D Flip-flop들이 synchronous하게 순차적으로 연결되어 있는 것을 확인할 수 있다. 따라서, 왼쪽에서 순차적으로 데이터를 입력받고, 우측으로 입력받은 데이터를 출력하는 것을 확인할 수 있다. 이러한 SISO shift register는 주로 신호에 지연시간을 주기 위해 사용된다.

* Serial In Parallel Out shift register(SIPO)

SIPO shift register는 하나의 입력 line으로부터 순차적 입력을 받고, 병렬 line을 통해 출력하는 register이다. 아래의 회로도로 SIPO shift register를 나타낼 수 있다.



4개의 서로 연결된 D Flip-flop으로 구성되며, 첫 flip-flop의 출력이 다음 flip-flop의 입력으로 연결되어있는 구조로 연결되어있다. 추가적으로, CLR(Clear) 신호가 Flip-Flop들에 연결되어 리셋을 할 수 있도록 되어있다. 이 네 Flip-flop들은 같은 clock signal에 따라 활성화되므로 서로 synchronous하다. SIPO register는 순차적인 데이터를 병렬적인 데이터로 변환하는 데 주로 의의가 있기 때문에, 하나의 데이터 line의 신호를 여러 개의 line으로 demux 작업을 해야 하는 통신 분야에서 주로 사용된다.

* Parallel In Serial Out shift register(PISO)

PISO shift register는 동시에 입력이 각 flip-flop에 전달되는 병렬적인 입력을 받아, 하나의 line으로 순차적으로 출력하는 종류이다. 아래의 다이어그램을 통해 PISO shift register의 구조를 확인할 수 있다.

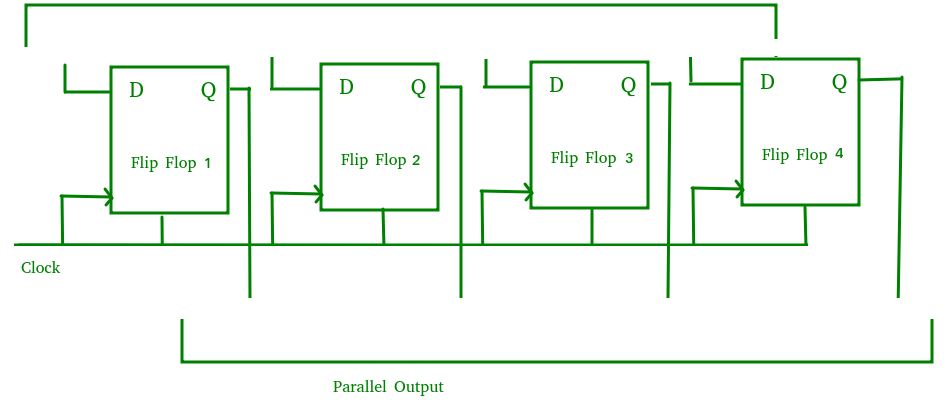
텍스트, 화이트보드이(가) 표시된 사진

자동 생성된 설명

연결된 4개의 D flip-flop으로 구성된 상태이다. Clock 신호는 각각의 flip-flop에 직접 연결되어 있지만, 입력되는 데이터는 상단의 multiplexer를 거쳐 각각의 flip-flop에 개별적으로 연결되어있는 것을 볼 수 있다. 이전의 flip-flop에서의 데이터와 방금 입력받은 데이터는 mux의 입력으로 들어오고, 해당 mux의 출력은 다음 flip-flop의 입력으로 연결된다. 같은 clock signal에 의해 묶여있기 때문에, 위의 네 flip-flop들은 서로 동기화 되어있는 상태다. PISO shift register는 여러 line의 병렬적인 데이터를 하나의 line으로 보내는 순차적 출력으로 바꿔주는 데에 사용된다.

* Parallel In Parallel Out shift register(PIPO)

PIPO shift register는 데이터를 병렬적으로 받아, 병렬적인 출력을 해주는 shift register이다. 아래의 회로도는 이 PIPO register를 보이고 있다.

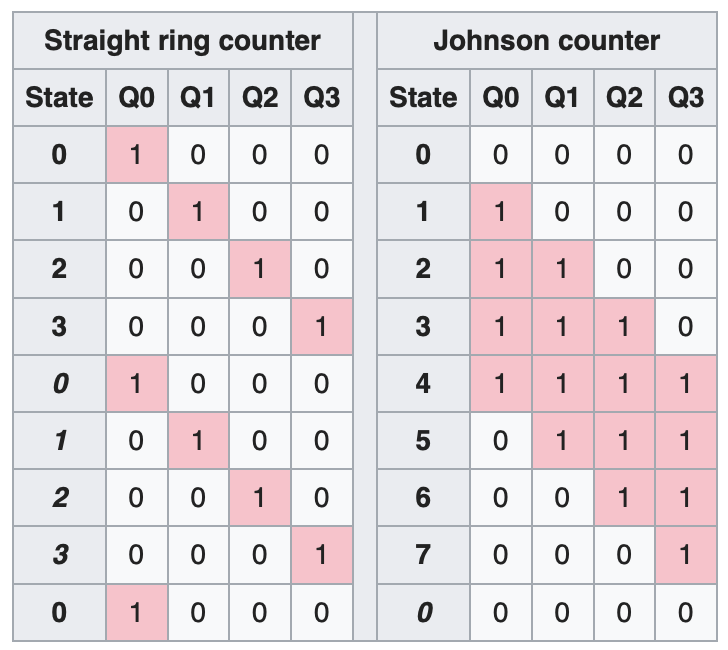


각각의 4 flip-flop들은 CLR 신호, clock 신호와 직접 연결되어 있는 모습을 보인다. 이러한 register에서는 데이터의 shift 작업이 필요하지 않기 때문에, 각각의 flip-flop 간의 연결이 존재하지 않는다. 각각의 flip-flop들에게 데이터는 따로 전해지고, 출력도 각각의 flip-flop이 연결된 개별적인 line들로 전해진다. 이러한 PIPO shift register는 임시 저장 장치로서 자주 사용되며, SISO register처럼 실행 딜레이를 주기 위해서도 사용된다.

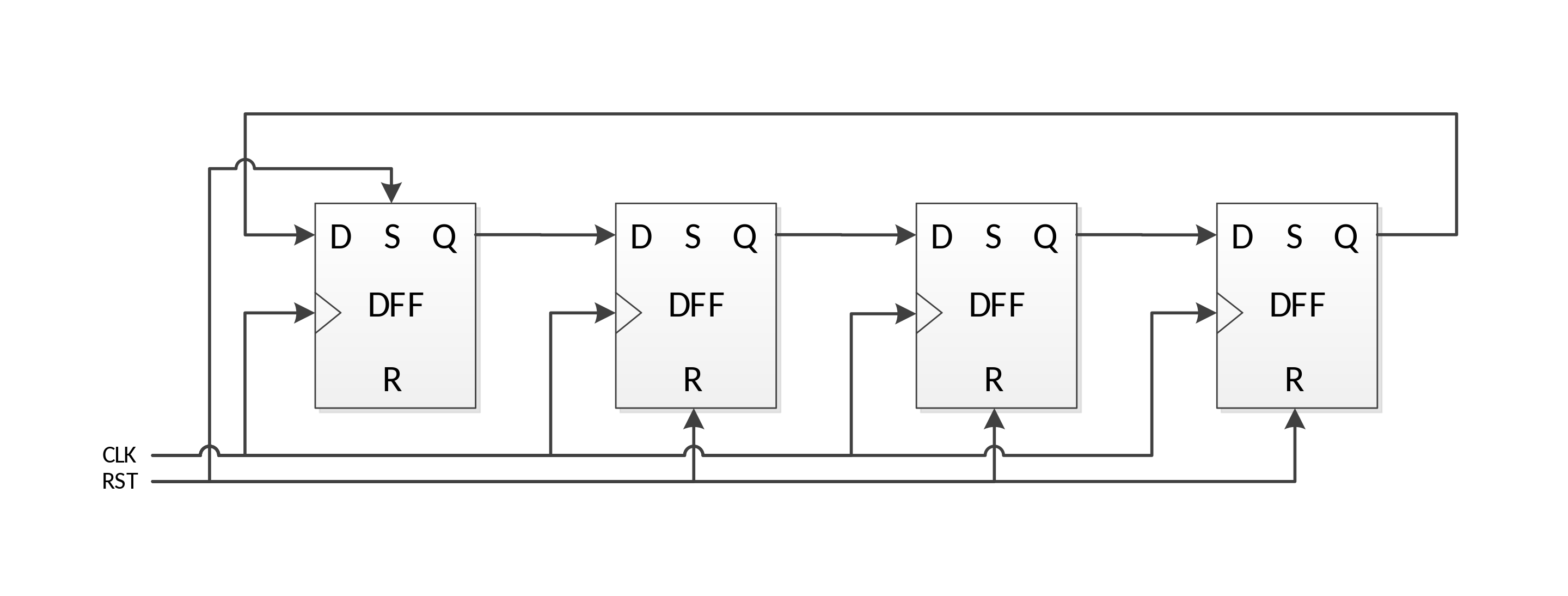
1. Ring counter에 대해서 조사하시오.

Ring counter는 shift register에 연결되어 있는 flip-flop들로 구성되는 counter의 일종이다. 이 때 이 전의 flip-flop에서 나온 출력은 첫 flip-flop의 input으로 들어가므로, 전체적으로 ring과 같은 구조를 보인다.

이 ring counter는 두 개의 종류로 나뉘는데, straight ring counter는 마지막 shift register의 출력을 첫 shift register의 입력으로 주어 유일한 1(또는 0) 값을 이 ‘링’ 안에서 돌리는 ring counter이다. 또 다른 ring counter로는 twisted/Johnson counter가 있다. 이 ring counter는 마지막 shift register의 출력 값을 첫 register의 입력으로 넘겨주고, 1과 0의 연속적인 stream을 순환시킨다. 이 두 종류의 ring counter를 state table로 나타내면 다음과 같이 나타낼 수 있다.

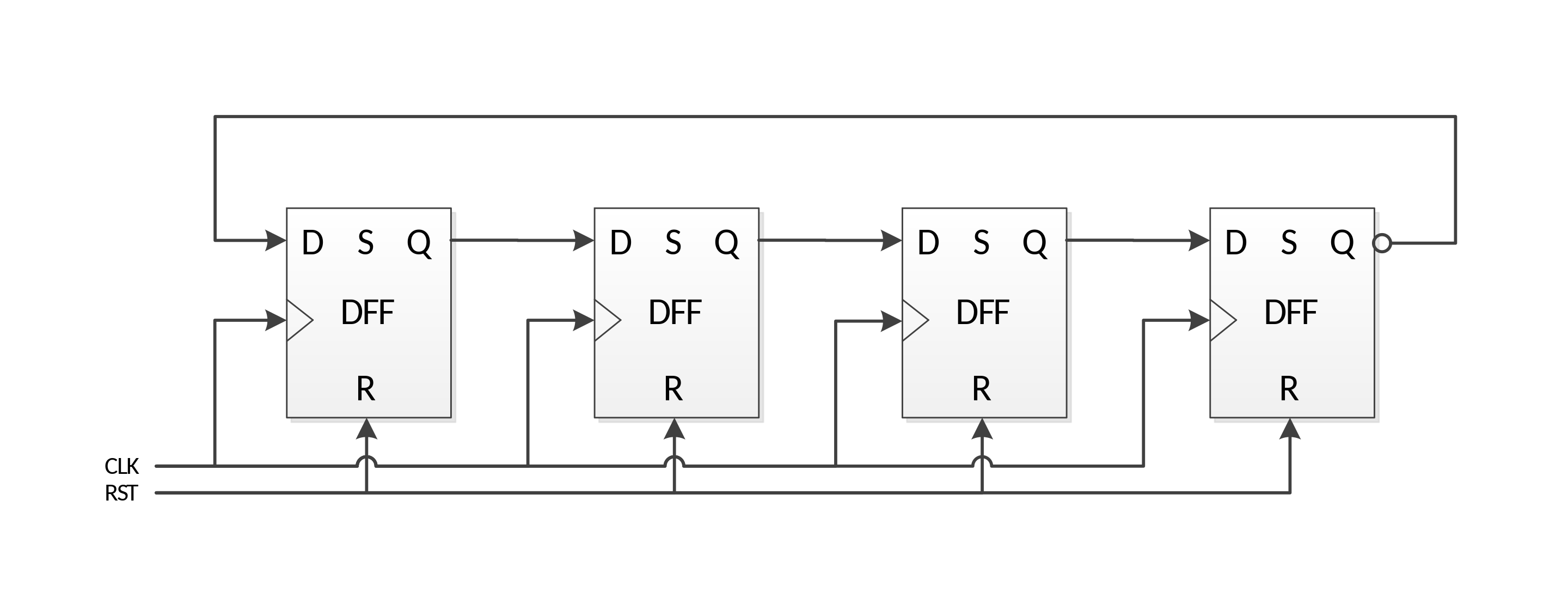


Straight ring counter의 회로도는 다음과 같이 나타난다.



이 때, rst 신호가 들어오면 초기값인 1000 state로 돌아가는 것을 확인할 수 있다.

Johnson counter는 이 회로에 invert 게이트 하나를 추가하여 다음과 같이 나타내면 된다.



마지막 shift register로부터 첫 shift register로 값을 넘겨줄 때 값이 invert 되어 들어간다는 것을 확인할 수 있다.

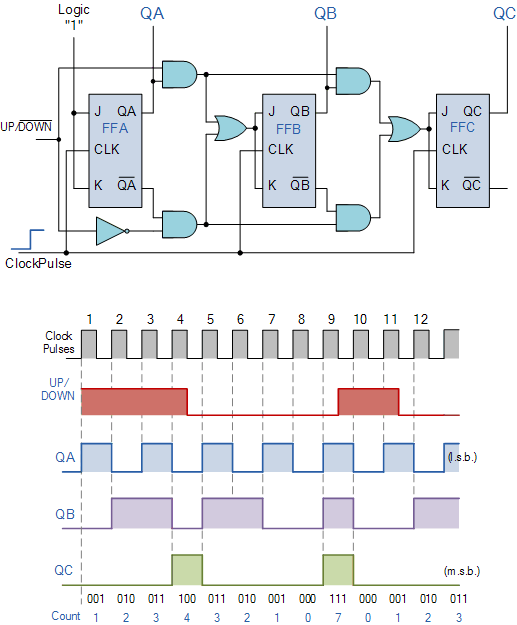
Ring counter들은 대부분 유한 상태 기계(FSM)을 만들기 위해 사용된다. Binary counter는 adder 회로가 필요하기 때문에 ring counter보다 더 복잡한 구조를 가지고, bit의 수가 커짐에 따라 propagation에 필요한 delay의 크기가 커질 수 밖에 없게 되는 반면, ring counter의 propagation delay는 코드에 사용되는 비트의 수와 무관하게 언제나 거의 상수 값으로 고정된다.

일반적으로 ring counter는 일반적인 숫자의 binary형태의 encoding보다 더 낮은 밀도를 가지는 코드를 만든다는 점에서 단점이 있다. 예를 들어, binary counter는 n-bit에서 개의 state를 가지는 반면, straight ring counter는 개의 state를, Johnson counter는 개의 state를 가질 수 있다. 따라서 만약 register의 값이 logic gate들의 값보다 더 비싼 경우에는 이 단점을 고려해 보아야 할 필요가 있다.

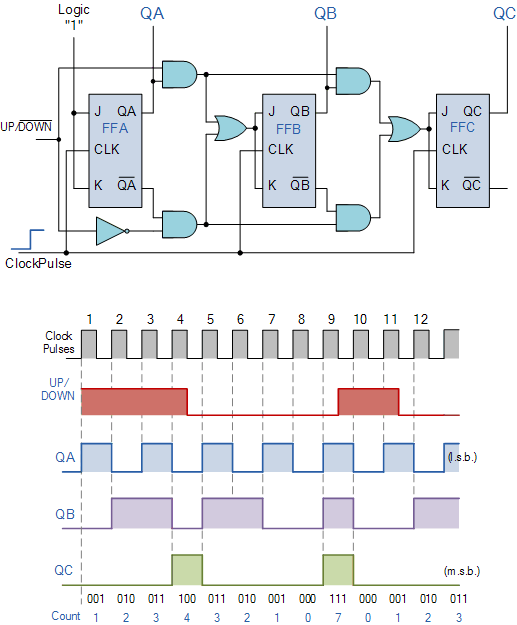
1. Up Down counter에 대해서 조사하시오.

Counter들은 0, 1, 2, …, 9 처럼 counting up하는 방식으로 제작하거나, 또는 반대로 9, 8, 7, 6, …, 0 순서로 counting down하는 방식으로 구현할 수 있다. 이 때 전자는 Up counter, 후자는 Down counter라고 한다. 이 때, 이러한 두 종류의 counter를 모두 사용할 수 있는 더 일반적인 형태의 counter를 Up/down counter, 또는 bidirectional counter라고 한다.

Bidirectional counter는 주어지는 어떠한 count 배열에 대해서도 위, 또는 아래로 counting하는 것이 모두 가능한 counter이며, 원하는 때 언제나 counting 방식을 바꿀 수도 있다. 이러한 bidirectional counter의 회로도는 아래에 보이는 것과 같다.



Clock 신호에 따른 세 flip-flop의 bit 값의 변화는 다음 그래프로 확인할 수 있다.

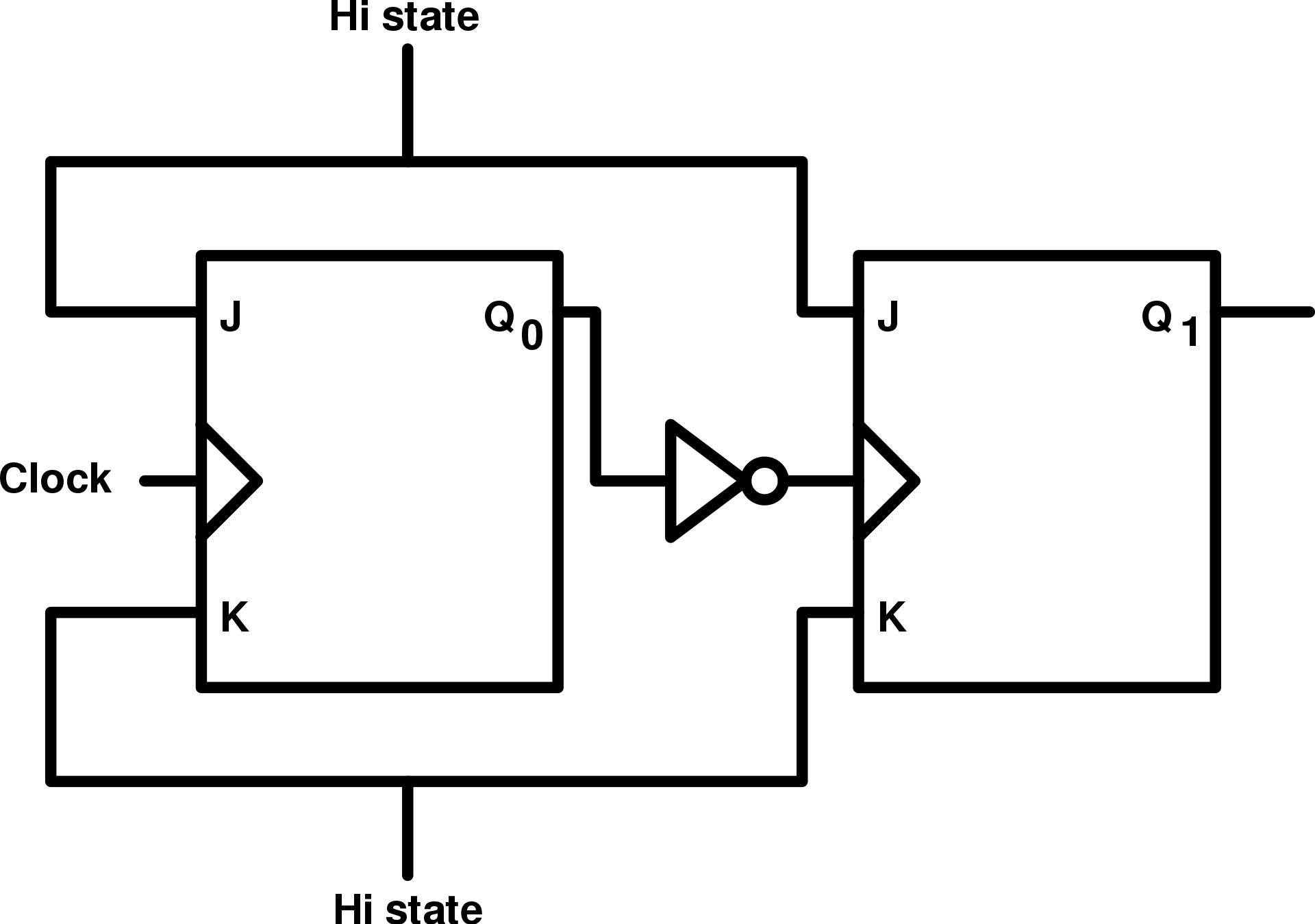


신호의 값에 따라 counting하는 방향이 바뀌는 것을 확인할 수 있다.

일반적으로, 대부분의 bidirectional counter 칩들은 counting을 하는 도중 어느 때라도 counting 방향을 바꾸도록 할 수 있는데, 이는 counting 방향을 정하는 추가적인 입력을 받음으로써 구현할 수 있다.

1. Ripple counter에 대해서 조사하시오.

위에서 살펴본 counter들과 다르게, ripple counter는 비동기식(asynchronous) counter이다. 이 때 외부의 clock과 연결된 부분은 오직 첫 번째 flip-flop 하나이다. 나머지 flip-flop들은 이 전의 flip-flop의 출력으로부터 clock 값을 받는다. 따라서 clock 신호가 flip-flop들 사이를 물결처럼 ripple하는 방식을 보인다. 아래는 두 개의 JK flip-flop을 사용한 ripple counter를 보이고 있다.



위 그림에서부터, 첫 번째 flip-flop은 각 clock cycle마다 비트의 값이 0에서 1로, 또는 1에서 0으로 바뀔 것임을 확인할 수 있다. 에서는 입력받은 clock cycle의 2배 길이의 clock 신호가 발생하게 된다. 만약 이 출력이 위 다이어그램처럼 두 번째 D Flip-flop의 clock signal로서 사용이 된다면, 이 추가된 flip-flop은 이 전의 flip-flop보다 2배 더 느리게 bit를 flip하게 된다.

이전 flip-flop의 출력을 다음 flip-flop의 clock 신호로 바꾸는 식으로 n개의 flip-flop을 구성하면, 까지 셀 수 있는 ripple counter를 구현할 수 있다.

1. 기타 이론.

* Ripple counter에서 flip-flop들의 출력을 clock 신호로 사용하는 방식은 data bit들 간의 타이밍 문제를 일으킬 수 있기 때문에, 일반적인 synchronous 회로 설계 디자인과는 호환이 되지 않는다.