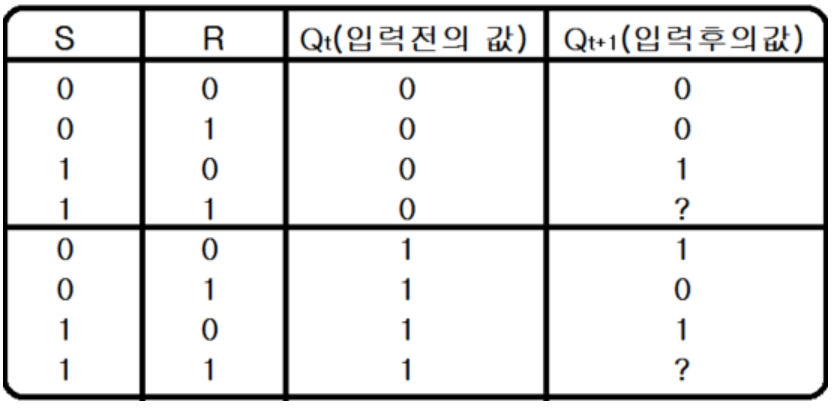
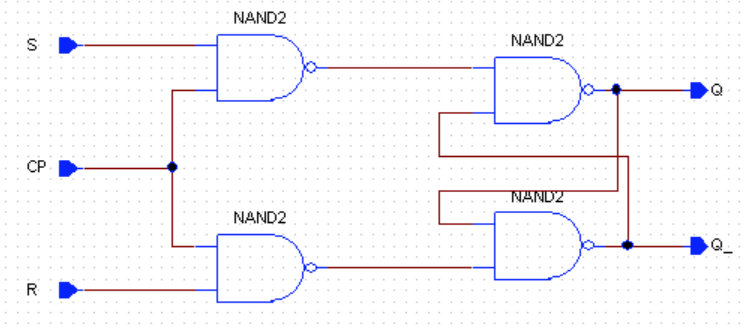
11주차 예비보고서

전공: 컴퓨터공학과 학년: 3학년 학번: 20191612 이름: 윤기웅

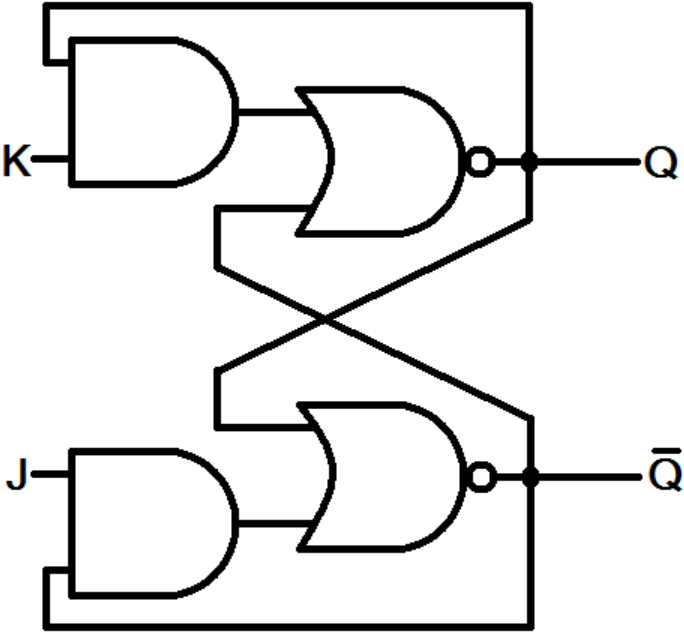
**1.**

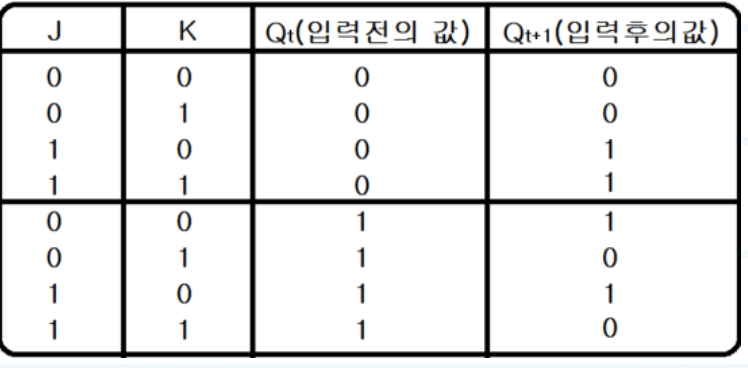
순서 논리회로 중 하나인 rs 플립플롭은 기억장치라고 생각할 수 있다. 플립플롭은 1비트의 정보를 저장하고 기억하는 회로이다. 펄스로 주어지는 입력에 따라 2진 정보를 바꾸기가 가능하다. RS 플립플롭의 R와 S는 각각 Reset, Set의 줄임말이고 이 두 입력을 받아서 현재 상태인 Q와 다음의 상태인 Q'가 출력이 된다. 그래서 만약 입력 두 개가 모두 0인 경우는 현 상태가 그대로 Q'가 되고 R만 1인 경우에는 0을 출력하고 S만 1인 경우는 1을 출력한다. 그런데 문제는 입력이 모두 1인 경우에 결과를 알 수 없다는 것이다.





**2.**

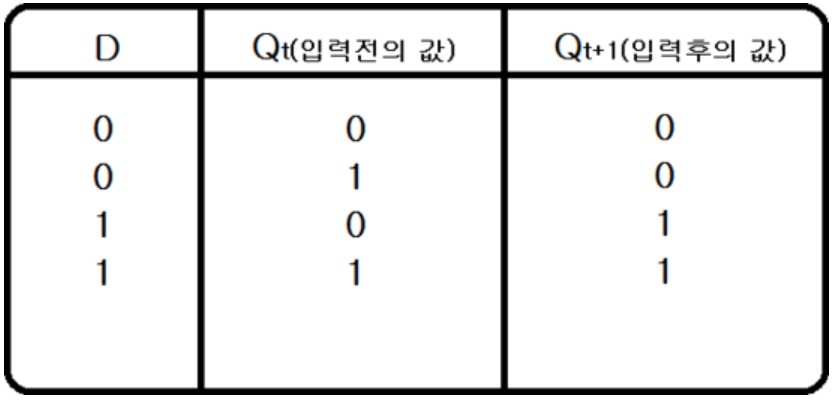
****

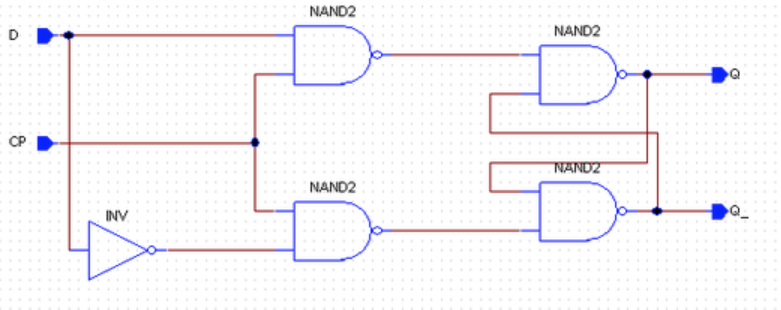
****

JK 플립-플롭은 D와 T 플립플롭의 기능을 모두 갖고 있어서 상태의 유지도 가능하고 상태의 반전도 가능한 회로이다. 입력으로 J,K 두 입력을 받고 Q인 현재상태와 Q’ 다음상태가 존재한다. 이전의 RS플립플롭의 문제점을 보완하기 위해 나온 플립-플롭이라 볼 수 있고 입력이 둘 다 0인 경우 Q에 변화가 없고 둘 다 1인 경우는 Q의 보수를 결과로 갖는다. 만약 K만 1인 경우는 0을 출력하고 J만 1인 경우는 1을 출력한다. 이전의 RS 플립플롭하고 거의 비슷하지만 입력이 모두 1인 경우에 어떤 결과가 나오는 지 모르는 오류를 수정한 것이다.

**3.**

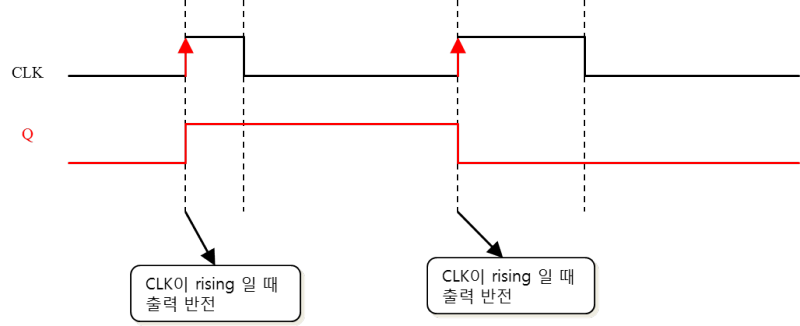
D가 delay의 약자이고 그 의미처럼 입력으로 들어온 값을 그대로 출력한다. 다른 플립 플롭처럼 특별한 기능이 없고 입력만 그대로 내보내는데 이는 입력 값을 그냥 잠시 보관하고 다른 곳에 전송하기 전에 시간을 늦추는 역할을 한다. 아래의 그림을 보면 알 수 있듯이 D 플립플롭은 RS 플립플롭을 응용한 플립플롭이다. 하나의 입력에 대해서 한 쪽은 그대로 다른 한 쪽은 인버터를 거친 값을 써서 RS 플립플롭을 구성하면 된다.

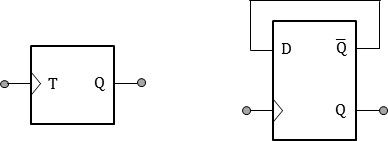


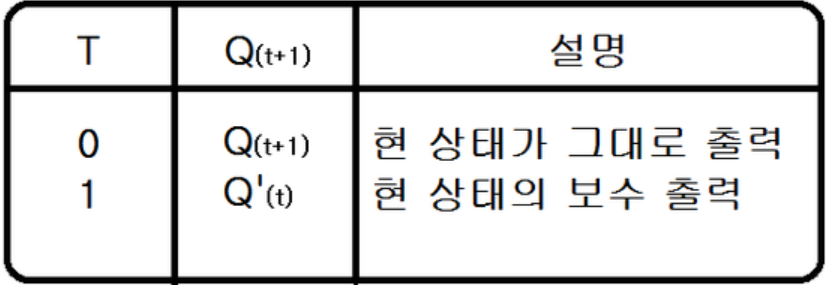


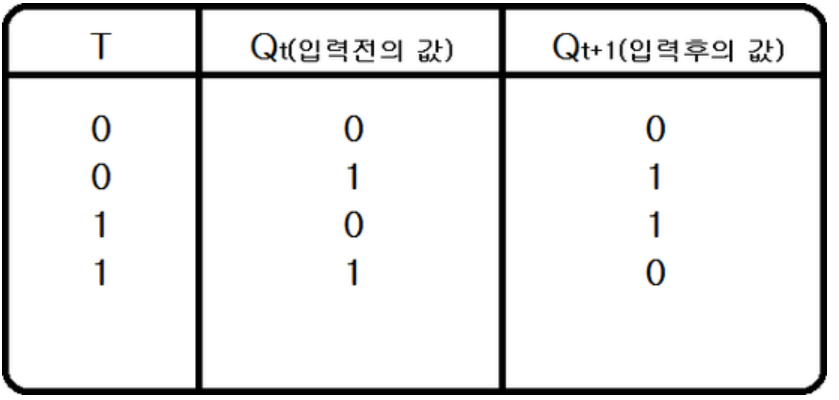
**4.**

xor의 연산과 유사하게 실행되며 하나의 입력값인 T에 의해서 Q의 값에 영향을 준다. T는 toggle의 앞 글자를 따서 명명했다. 만약 T의 값이 0인 경우 Q에 영향을 주지 못하고 반대로 T가 1인 경우는 Q의 값을 보수화 시킨다. 아래의 시간차트가 T 플립-플롭의 시간 차트인데 클락이 상승하는 경우에만 Q상태가 변하는 것을 볼 수 있다.

****

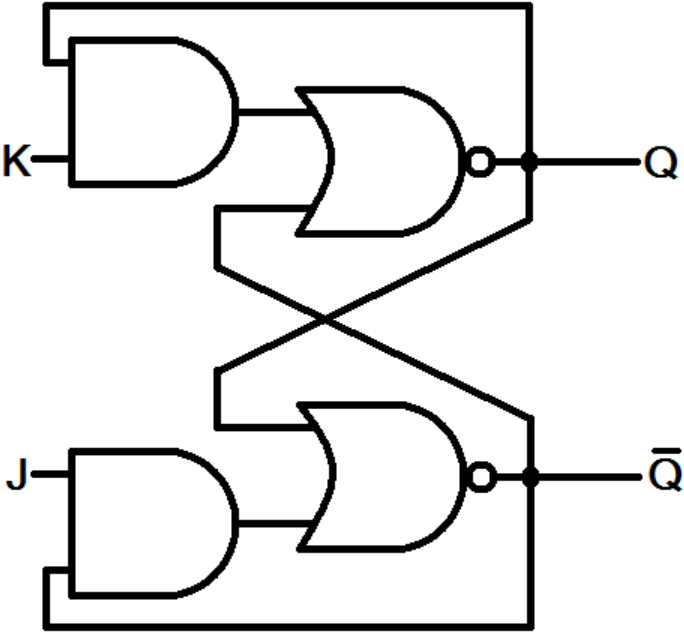
****

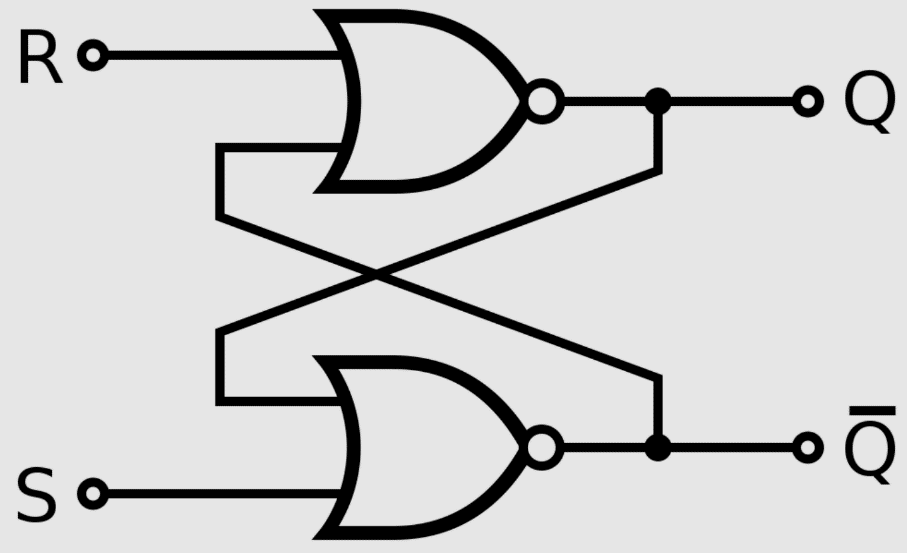
****



**5.**

시스템이 전원이 공급된 상태에서 상태 변환에 필요한 신호가 발생할 때까지 현재의 상태를 유지하는 기억장치이다. 또한 전원이 꺼지면 휘발되는 특성을 갖고 있으면서 SRAM을 구성한다. latch와 flip flop 모두 이와 같은 기능을 갖고 있지만 차이는 클락을 갖고 있는 지의 여부이다. Latch는 클락이 없어서 비동기식 동작을 한다. 즉 플립 플롭을 만들려면 래치의 입력과 클럭 신호를 곱해서 둘 다 신호가 들어오는 경우만 동작하게 만들면 된다.





**6.**   컴퓨팅 시스템이나 임베디드 시스템에서 clock의 속도를 갖고 성능을 평가하는데 사용하기도 한다. 클락은 이러한 시스템에 들어가는 디지털회로의 심장박동이라 할 수 있다. 즉, 규칙적인 주기의 역할을 하면서 시스템의 작동의 기준을 제공한다. 모든 디지털 회로는 결국 클럭을 기준으로 발생하는 이벤트의 순서가 정해지고 동작한다고 봐야한다. 그렇다고 클럭이 빠를수록 무조건 좋은 것은 또 아니다. 각 장치나 시스템이 동작하는 시간 범위가 있기 때문이다. 디지털 회로 중에서 순차논리회로를 통해서 복잡한 일들을 처리 가능하다. 여러 플립 플롭에 의해 구성된 순차논리회로는 현재의 입력과 출력에 의해서 다음 번의 출력을 구성하기 때문에 이를 회로로 구현하려면 작업의 순차적인 진행이 중요하고 이 때 클럭이 반드시 필요하다. 그래서 플립 플롭 등은 클락이 있어서 동기화가 가능하다.

**7.**   어떤 시스템 상의 특정한 이벤트들을 감지하려면 트리거라는 것이 필요하다. 트리거는 원래는 전자에서 사용하는 내용이지만 시스템 프로그래밍에서 비슷하게 사용한다. 트리거는 레벨 트리거와 에지 트리거로 나누어진다. 레벨 트리거의 경우는 현재의 상태 변수를 중심으로 작동한다. 상태가 1이면 이벤트를 작동시키고 0일 경우는 아무 이벤트가 발생하지 아니한다. 그러나 에지 트리거는 조금 다르다. 상태가 1에서 0으로 변하는 경우나 혹은 0에서 1로 변하는 경우를 각각 하강 에지와 상승 에지로 부르고 에지가 발생하는 경우에 이벤트를 작동시킨다. 만약 소켓 통신의 상황을 가정하고 여기에 데이터가 10바이트 받게 되면, 상태가 0에서 1로 변경된다. 이 경우 레벨 트리거와 에지 트리거가 이벤트를 만들어낸다. 에지 트리거는 특정 일이 생기는 순간에 발생하고 레벨 트리거에 비해 짧은 순간에 개별적 일의 처리가 가능하다. 시스템에 전압이 들어오면 상승 에지가 발생하고 전압이 나가면 하강 에지가 발생한다.

8.

master slave 플립플롭은 플립플롭 2개와 1개의 인버터로 만들 수 있고 하나의 플립플롭은 주된 동작을 하고 나머지 하나는 종속된 동작을 수행한다. clock pulse가 0에서 1로 변하는 지 그리고 1에서 0으로 변하는 지에 따라서 주 플립플롭이 실행되는지 종 플립플롭이 실행되는 지 정해진다. 만약 clock pulse가 1이 되면 주 플립플롭, 0이 되면 종 플립플롭이 된다. 시간의 텀을 두고 주종 플립플롭이 실행되면 출력이 다시 입력이 되는 레이스 현상을 예방할 수 있다. 이전에 살펴본 jk 플립플롭이 이 문제를 갖고 있는데 이는 결과값이 다시 입력으로 들어가서 회로의 불안전한 상태를 유발한다는 특징을 갖고 있다.

9.

플립플롭은 전자 기기의 부품에 많이 사용된다. 레지스터가 특히 플립플롭을 많이 사용하는데 이 플립플롭은 각자 1비트의 데이터를 갖게 되며 플립플롭을 많이 사용할 수록 저장 가능한 데이터 양이 많아지게 된다. 레지스터가 입력과 출력을 직렬로 할지 병렬로 할지에 따라서 데이터의 이동방향을 정할 수 있다. 플립플롭은 카운터에도 사용이 가능한데 상호 연결된 많은 플립플롭을 조합해서 제어 장치에서 회로 동작 통제할 때 주된 역할을 한다.