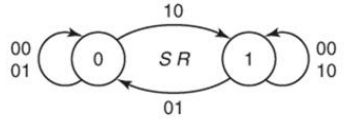
**11주차 예비보고서**

**전공: 생명과학과 학년: 4학년 학번: 20182186 이름: 김승원**

**1. SR flip flop 조사**

SR flip flop은 S(set)과 R(Reset) 두 개의 입력을 가지는 flip flop이다. 보통 flip flop의 많은 경우 clock이 연결되어 있고 상태 값 q를 결정한다. Set은 다음 차례의 clock edge(trailing or leading edge)에서 flip flop에 1을 저장한다. Reset 입력은 다음 차례의 유효 clock edge에서 flip flop에 0을 저장한다.

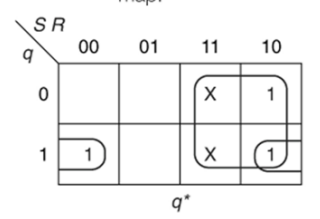
테이블이(가) 표시된 사진

자동 생성된 설명텍스트, 시계이(가) 표시된 사진

자동 생성된 설명

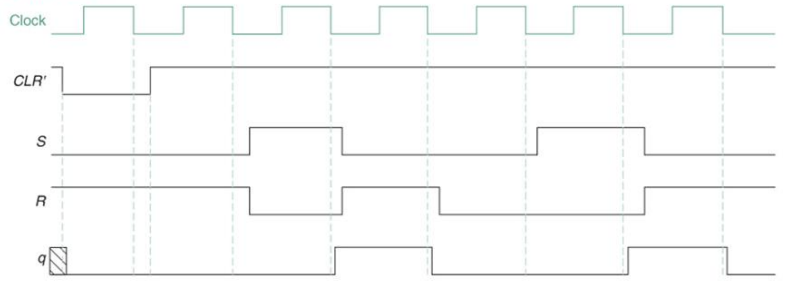
**<Figure 1. SR flip flop state diagram 및 truth table>**

진리표에서 확인할 수 있듯이 S와 R이 모두 1인 경우는 허락되지 않는다. Don’t care에 해당하는 경우라고 보면 되고, 이를 통해 카르노 맵으로 나타내면 아래와 같다.



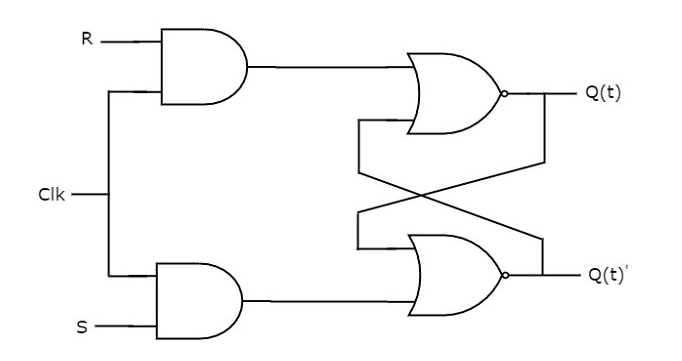
**<Figure 2. SR flip flop K-map 표현>**

따라서 Combinational logic에서 했던 것처럼 표현이 가능하다. 즉, minimal한 Boolean expression으로 q\* = S + R’q로 표현이 가능하다.



**<Figure 3. SR flip flop의 timing diagram>**

S와 R 입력에 따라 q값이 변화한다. 변화되는 시점은 trailing edge에서 변화 point가 생기는데, trailing edge 시점의 S와 R의 값에 따라 q에 truth table과 같이 알맞은 값이 저장된다.

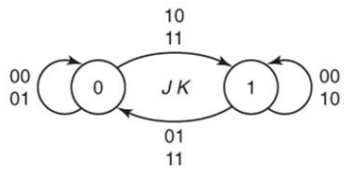


**<Figure 4. SR flip flop의 회로도>**

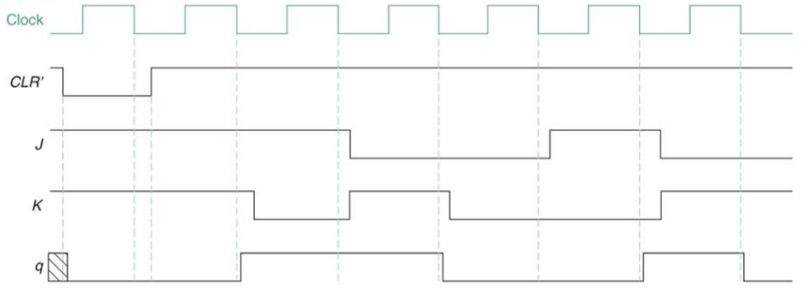
**2. JK flip flop**

JK flip flop은 J와 K로 input을 받는 flip flop이다. J와 K에는 별다른 뜻은 없다. 대부분의 기능은 SR flip flop의 두 입력의 기능과 같은데, SR flip flop과 다른 점은 J와 K가 모두 1일 때 not allowed가 아닌 어떠한 유효한 동작을 한다. J와 K가 모두 1일때는 Toggle동작을 만드는데, 이는 T flip flop의 역할인 상태를 반전시키는 동작을 하는 것이다. 그 이외에는 모두 SR flip flop과 동작을 동일하게 한다. SR flip flop과 같이 Boolean expression으로 나타내면 q\* = Jq’ + K’q이다.

텍스트, 시계이(가) 표시된 사진

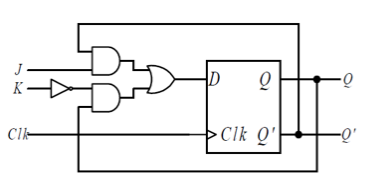
자동 생성된 설명

**<Figure 5. JK flip flop의 진리표와 state diagram>**

****

**<Figure 6. JK flip flop timing diagram>**

각각의 Trailing edge 시점에서 J, K의 입력이 1,1로 들어올 때 q상태값이 1로 transition이 일어나고 있고, J, K의 입력이 1,0으로 들어오면 1로 그대로 유지되는 것을 확인할 수 있다. 또한 각각 0, 1인 경우 q는 0으로 업데이트 된다. 마지막으로 0,0인 경우에는 q가 0으로 현재의 상태로 유지됨을 확인할 수 있다.

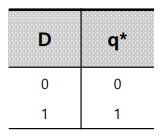
****

**<Figure 7. JK flip flop 회로도>**

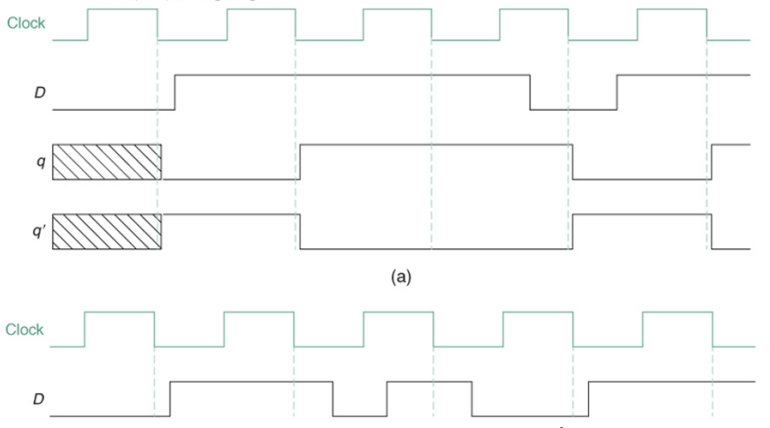
**3. D flip flop 조사**

D Flip flop은 가장 간단히 동작하는 flip flop이다. 프로그래밍이 가능한 논리회로 소자에 많이 사용된다. d는 delay로 알려져 있다. D flip flop은 input D값을 clock의 edge에서 capture해서 output Q에 반영한다. 이때 edge가 발생하지 않으면 Q는 변하지 않는 특징이 있다.

텍스트, 시계이(가) 표시된 사진

자동 생성된 설명

**<Figure 8. D flip flop diagrams>**

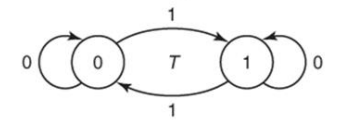
****

**<Figure 9. D flip flop timing diagram>**

첫번째 trailing edge triggered 후에 trailing edge 시점에 D값을 저장해서 상태 값 Q는 0으로 유지된다. 이후에 trailing edge 시점에 D값이 1이므로, q값이 1로 변하였고, 이후 trailing edge에서 D값이 그대로 1이므로, q는 1로 유지되는 모습을 확인할 수 있다. 이후 trailing edge 시점에 D값이 0으로 저장된 것을 보고 q는 1에서 0으로 transition이 일어난 것을 확인할 수 있다.

**4. T flip flop 조사**

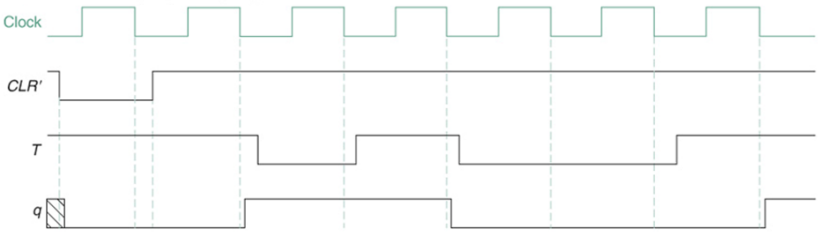
T flip flop은 T input 하나만 가진다. T는 toggle을 뜻한다. T flip flop의 동작의 경우 T가 1일 때 flip flop의 state를 toggle시키고, T가 0인 경우에는 state를 그대로 유지하는 특징이 있다.

텍스트, 시계이(가) 표시된 사진

자동 생성된 설명

**<Figure 10. T flip flop의 state diagram과 truth table>**

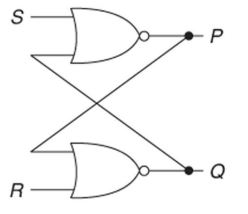
위에서 확인할 수 있듯이 toggle input이 0이면 그대로 output이 출력되고, T가 1인경우 q가 complement를 취한 상태로 q\*가 정해진다. 따라서 Boolean expression 식으로 q\* = T XOR q로 나타낼 수 있다.



**<Figure 11. T flip flop의 timing diagram>**

2번째 trailing edge triggered 시점을 확인해보면 T가 1이므로, q는 1로 상태가 변한 것을 확인할 수 있다. 그리고 3번째 trailing edge triggered 시점을 확인해보면 T가 0으로 q의 출력이 1로 유지되고, 4번째 trailing edge triggered 시점을 확인해보면 T가 1이므로, q값이 0으로 변한 것을 확인할 수 있다.

**5. Latch 조사**

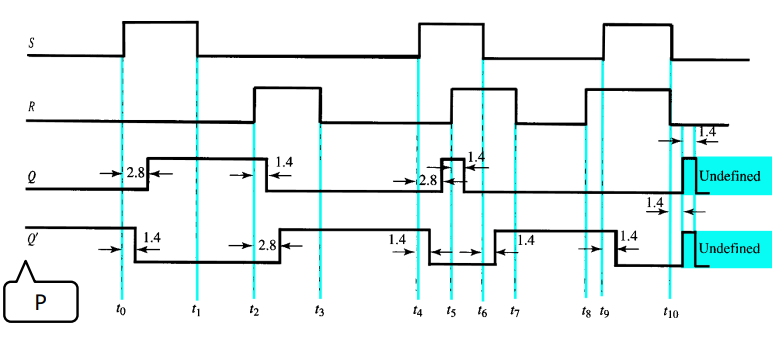


**<Figure 12. Latch 회로도>**

S와 R이 하나의 latch의 input이다. 회로도를 통해 확인할 수 있듯이 cross 연결을 통해서 출력 값을 p와 q 두 개인데, p는 간혹 q’으로 표기하기도 한다. 일반적으로는 q가 0일경우 p가 1이기 대문에 그렇게 표현한다. Latch는 sequential logic을 구성하기 위한 building block으로 사용한다.

두 가지 input signal S는 set signal이고, R은 reset signal이다. 사용자는 Q를 만들고자 하면 S를 1, R을 0으로 입력하면 된다. 또한 Q를 0으로 만들고 싶다면 S를 0으로, R을 1로 입력하면 된다. 즉, S와 R이 (1,0)이거나 (0,1)이면 위의 상태가 유지된다. 이때 만약 (0, 0)으로 input에 변화를 주게 되더라도 P와 Q는 직전 상태를 유지하는 특징이 있다. 하지만 (1,1)로 두 입력이 들어오게 되면 S와 R이 동시에 활성화되는데 이때 Latch는 동작하지 않는다. 두 출력이 (0, 0)이 됐을 때의 상황에서 Latch가 어떠한 상태가 될지 예측할 수 없는 undefined state가 되기 때문에 불안정한 상태가 된다. 따라서 S와 R이 모두 1이 되는 상황은 없도록 해야 한다.

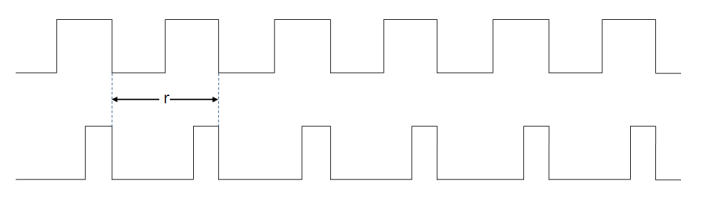
Latch 회로의 Boolean expression 식은 P = (S + Q), Q = (R + P)’으로 나타낼 수 있다.

****

**<Figure 13. SR latch timing>**

**6. Clock 조사**

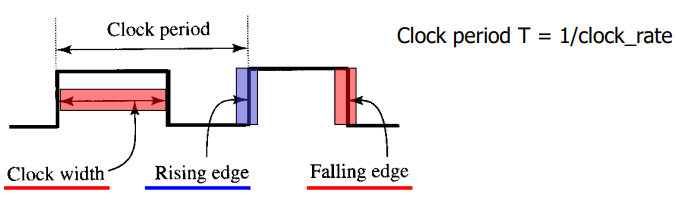
Clock은 0과 1사이를 일정한 주기로 바꾸는 역할을 한다. 전기신호를 통해 이러한 역할을 할 수 있도록 하며, clock에는 두 가지 종류의 clock이 있다.

****

**<Figure 14. Clock의 두 가지 pattern>**

위의 경우는 0과 1이 일정한 간격인 경우이고, 밑은 0이 더 길고 1이 더 짧은 clock이다. 둘 다 모두 가능한 패턴이다. 완벽하게 똑 같은 wave form이 반복되는 구간이 주기라고 할 수 있고, 이때 edge가 중요한 부분이다. 0에서 1로 가는 것을 rising edge라고 하며, 1에서 0으로 가는 것을 falling edge라고 한다. Rising edge는 positive edge 또는 leading edge라고도 하며, falling edge는 negative edge나 trailing edge라고도 표현한다.

**7. Edge trigger 조사**



**<Figure 15. Clock의 rising edge와 falling edge>**

Edge trigger clock이 변할 때만 data를 입력 받는 것이다. 각각의 flip flop timing diagram에서도 확인할 수 있었듯이 clock의 변화가 있을 때 만 상태의 변화가 일어난다. 반대로 말하면 만약 어떠한 trigger도 없었다면, 상태의 변화는 일어날 수 없다는 뜻이기도 하다. Edge의 종류는 0에서 1로 갈 때 leading edge trigger이고, 0에서 1로 가는 것은 trailing edge trigger이다.

**텍스트, 시계이(가) 표시된 사진

자동 생성된 설명**

**<Figure 16. 좌측은 leading edge flip flop이고, 우측은 trailing edge flip flop>**

Leading edge flip flop과 trailing edge flip flop의 표현 방법은 위와 같다.

**8. Master-slave 조사**

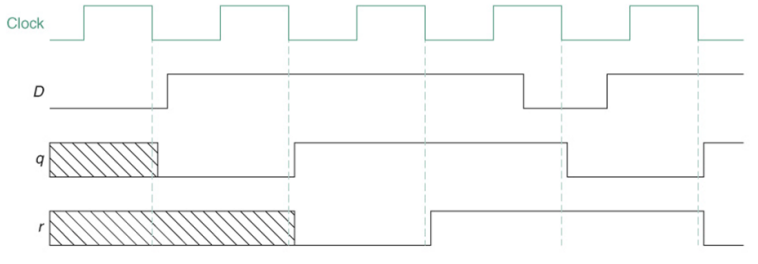
Master-slave는 device나 process(master)가 하나 이상의 다른 device 또는 process(slave)를 통제하는 제어 모델을 말한다. 일부의 시스템에서 master로 적절한 device로 선택하고, 다른 device들은 slave 역할을 수행하고 있다. Master-slave flip flop은 두 개의 연결된 flip flop이다. 두 개 중 하나는 Master flip flop으로 동작하고, 다른 하나의 flip flop의 경우 slave flip flop으로 동작한다. 두 개의 D flip flops로 구성할 수 있다.

**텍스트, 시계, 손목시계이(가) 표시된 사진

자동 생성된 설명**

**<Figure 17. 2개의 D flip flop으로 구성한 Master-slave flip flops>**

각각의 flip flop은 clock의 펄스에 연결되어 있게 된다. 하나의 flip flop이 trailing edge trigger에 의해 동기화 된다면, 다른 하나는 leading edge trigger에 의해 동기화된다. Master flip flop이 만약 trailing edge일 때 활성화 상태가 된다면, slave flip flop은 비활성화 상태가 된다. 반면에 Leading edge일 경우 Master flip flop은 비활성화 상태가 되지만, Slave flip flop은 활성화되는 것이다. 즉, D값에 따라 master flip flop이 영향을 받고 그 영향에 따라서 slave flip flop이 변하는 것을 확인할 수 있다.



**<Figure 18. Master-slave flip flops(D flip flops)의 timing diagram>**

첫 번째 trailing edge 시점에 D값을 캡처해서 q는 0이된다. 두 번째 trailing edge시점을 보면 D값이 1이니까 q는 1로 상태 값이 변한다. 이때 r은 q값인 0을 저장하여 0을 그대로 출력한다. 그 다음 세 번째 trailing edge 시점을 보면 D는 1로 유지되기에 q는 1로 그대로 유지되지만, r은 q값의 1을 캡처해서 1로 출력 값이 변하게 된다. 즉, r은 주기가 delay되어서 나타나게 되는 특징이 있다.

**9. 기타 이론**

Sequential logic system에는 Moore model과 Mealy model이 있다.

Moore machine은 output이 현재의 state에만 의존하는 machine model이다. 그 이유는 output이 clock edge에만 바뀌기 때문에 input이 입력되고 한 cycle이 지나서야 값이 변화한다. Mealy machine은 현재의 state와 함께 input이 output에 영향을 바로 미친다. 위의 한 cycle을 기다리고 나서야 값이 변화하는 Moore machine의 특징 때문에 Mealy machine이 Moore machine보다 더 빠르게 반응한다. 이러한 Moore machine과 mealy machine의 장점을 합친 것이 synchronous mealy machine이다. Flip flop을 사용해서 clock에 맞추어 state를 동기화한다. 즉, mealy machine처럼 output을 빠르게 계산하고 flip flop을 이용하여 next state와 동기화해 mealy machine에서 발생 가능성이 있는 glitchy 문제를 피할 수 있도록 하는 장점이 있다.