11주차 예비보고서

전공: 기계공학과 학년: 3학년 학번: 20191820 이름: 김형준

**1.**

플립플롭(Flip-Flop)은 1bit의 정보를 보관 및 유지할 수 있는 회로이며 순치회로의 기본 요소이다. 입력값을 항상 결과값에 반영하는 Latch와 다르게 플립플롭은 이전 상태를 유지하다가 클럭 신호(Clock Pulse)의 값이 바뀌는 시점에서 값이 변경 또는 유지된다.

클럭 신호가 0에서 1로 변할 때 반응하는 것을 leading edge triggered, 1에서 0으로 변할 때 반응하는 것을 trailing edge triggered라 한다.

클럭 신호 변화시 플립플롭에 저장되는 값은 플립플롭의 입력값과 현재 플립플롭에 저장된 값에 따라 결정된다.

RS 플립플롭은 가장 기본적인 형태의 플립플롭으로, 두 개의 입력 단자 S(Set), R(Reset)을 갖고 두 개의 출력 단자 Q(t)와 Q’(t)를 갖는다.

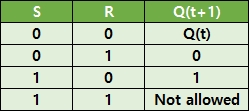
S와 R의 값이 모두 0인 경우, 플립플롭에 저장된 정보를 그대로 유지한다.

S가 1이고 R이 0인 경우, 플립플롭에 저장된 정보를 1로 Set 한다.

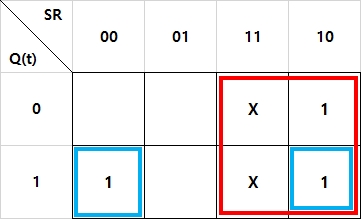
반대로, S가 0이고 R이 1인 경우, 플립플롭에 저장된 정보를 0으로 Reset 한다.

S와 R의 값이 모두 1인 경우의 입력은 유효하지 않은 input으로 간주하여 허용되지 않는다. (실제 회로에서 S=R=1인 신호가 들어오게 되면 오류가 발생하거나 하드웨어가 망가질 수 있음)

이를 표로 나타내면 다음과 같다.



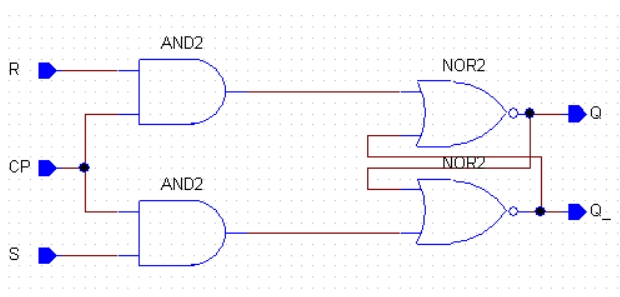
[RS 플립플롭의 characteristic table]



위와 같이 카르노맵을 사용해 RS 플립플롭의 논리식을 구하면, 를 구할 수 있다.

RS 플립플롭은 아래와 같이 RS Latch에 2개의 AND게이트를 추가해 만든 Gated Latch

에 Enable 신호로 Clock Pulse를 받게 해 구현할 수 있다.



**2.**

JK 플립플롭은 R=S=1일 때, 오류가 발생하는 RS 플립플롭의 단점을 보완한 플립플롭으로, R 대신 K, S 대신 J로 입력신호를 나타내며, 기본적으로 RS 플립플롭과 동일한 역할을 하지만 R=S=1일 때에도 유효한 input으로 처리한다. 이 경우, 저장된 정보에 보수를 취하게 된다.

따라서 JK 플립플롭은 두 개의 입력 단자 J, K을 갖고 두 개의 출력 단자 Q(t)와 Q’(t)를 갖고, J와 K의 값이 모두 0인 경우, 플립플롭에 저장된 정보를 그대로 유지한다.

J가 1이고 K가 0인 경우, 플립플롭에 저장된 정보를 1로 Set 한다.

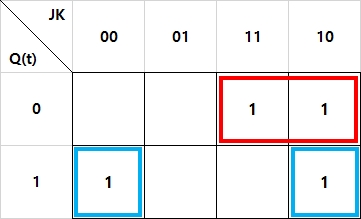
반대로, J가 0이고 K가 1인 경우, 플립플롭에 저장된 정보를 0으로 Reset 한다.

J와 K의 값이 모두 1인 경우, 플립플롭에 저장된 정보에 보수를 취한다.

이를 표로 나타내면 다음과 같다.



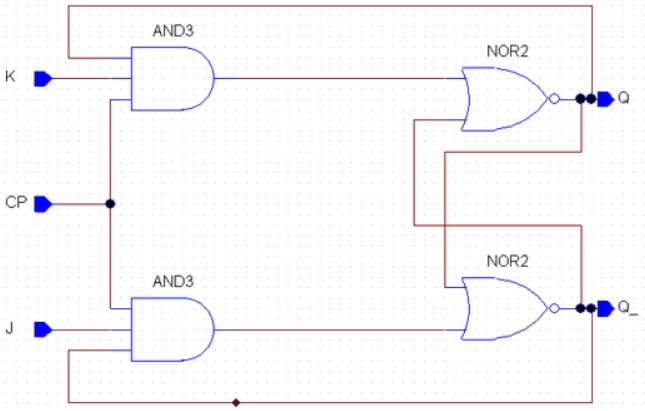
[JK 플립플롭의 characteristic table]



위와 같이 카르노맵을 사용해 JK 플립플롭의 논리식을 구하면, 를 구할 수 있다.

JK 플립플롭은 아래와 같이 Gated Latch로 구현한 RS 플립플롭에서 Q(t)와 Q’(t)를

각각의 AND 게이트의 입력에 추가함으로써 구현할 수 있다.



**3.**

D 플립플롭은 RS 플립플롭의 변형으로, 입력 D(Delay)를 그대로 출력하는 플립플롭이다.

D 플립플롭은 입력된 내용과 동일한 상태가 되도록 데이터를 일시적으로 보관할 때 사용되거나 디지털 신호의 전송되는 시간을 늦춰주는 지연 목적으로 사용된다.

D 플립플롭은 한 개의 입력 단자 D를 갖고 두 개의 출력 단자 Q(t)와 Q’(t)를 갖는다.

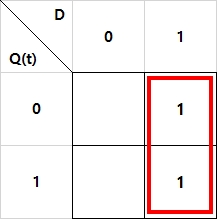
따라서 D가 1인 경우, 플립플롭에 저장된 정보를 1로 Set 한다.

반대로, D가 0인 경우, 플립플롭에 저장된 정보를 0으로 Reset 한다.

이를 표로 나타내면 다음과 같다.

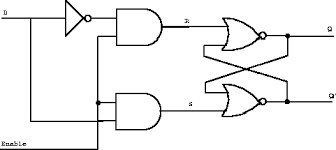
****

[D 플립플롭의 characteristic table]

****

위와 같이 카르노맵을 사용해 D 플립플롭의 논리식을 구하면, 를 구할 수 있다.

D 플립플롭은 아래와 같이 Gated Latch로 구현한 RS 플립플롭에서 입력 신호 R, S를 입력 신호 D를 사용해 각각 D’, D로 입력함으로써 만들 수 있다. 따라서 D가 1일 때 R=0, S=1이, D가 0일 때 R=1, S=0이 RS 플립플롭에 입력됨으로써 D 플립플롭의 기능을 수행하게 된다.

****

**4.**

T 플립플롭은 JK 플립플롭의 J와 K단자를 하나로 연결한 것으로, 입력 T(Toggle)이 1인 경우 저장된 값을 반전시키는 플립플롭이다.

T 플립플롭은 주로 계수기(Counter) 회로에 사용된다.

T 플립플롭은 한 개의 입력 단자 T를 갖고 두 개의 출력 단자 Q(t)와 Q’(t)를 갖는다.

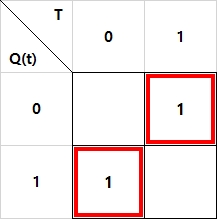
따라서 T가 1인 경우, 플립플롭에 저장된 정보에 보수를 취한다.

반대로, T가 0인 경우, 플립플롭에 저장된 정보를 그대로 유지한다.

이를 표로 나타내면 다음과 같다.

****

[T 플립플롭의 characteristic table]

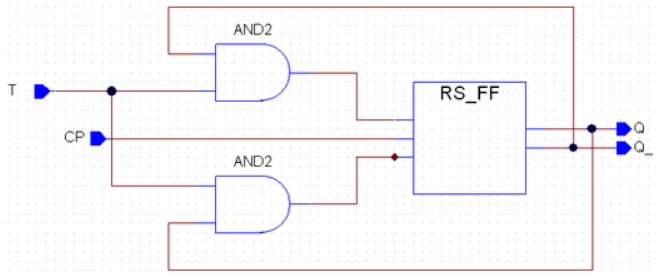
****

위와 같이 카르노맵을 사용해 T 플립플롭의 논리식을 구하면,

를 구할 수 있다.

T 플립플롭은 아래와 같이 JK 플립플롭에서 입력 신호 J, K를 입력 신호를 하나로 묶어서 입력 신호 T로 만들어 각각의 AND 게이트에 입력시킨다.

따라서 T가 1일 때 J=1, K=1이, T가 0일 때 J=0, K=0이 JK 플립플롭에 입력됨으로써 T 플립플롭의 기능을 수행하게 된다.

****

**5.**

Latch는 플립플롭과 유사하게 1비트의 정보를 저장하는 회로로써, 2개 또는 그 이상의 논리 게이트가 피드백으로 연결되어 구성된다.

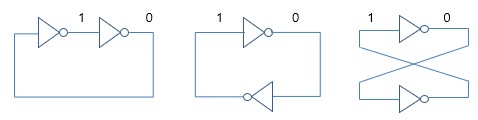
Latch는 순차 논리회로의 기본요소이며, 각종 Counter, 레지스터 RAM 등을 구성하는 기본요소이다.

클럭 신호(Clock Pulse)의 값이 바뀌는 시점인 클럭 엣지(CLK)에서 동작하는 플립플롭과 달리, Latch는 enable 신호가 아예 없거나, enable 신호가 있는 경우 enable 신호의 level에 따라 동작한다 (level sensitive).

따라서 플립플롭과 다르게, Latch는 비동기식 순차 논리회로 소자이다.

Latch에 클럭 입력과 함께 결합되면 플립플롭으로 만들 수 있다.

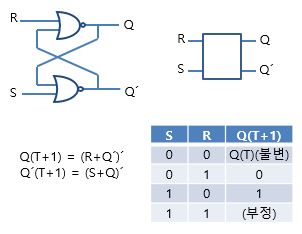
Latch에는 인버터형 Latch, SR Latch, D Latch 등의 여러가지 종류가 존재한다.



위와 같은 인버터형 Latch의 경우, 입력 없이 출력만 존재하며 순환적이다.

이론적으로만 존재하며, 실용성이 없어 실제로는 사용되지 않는다.

SR Latch의 경우, SR 플립플롭 처럼 S(Set), R(Reset) 입력이 있는 Latch이다.

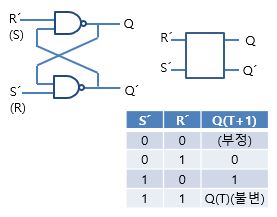


위와 같이 SR Latch는 S=R=0인 경우 저장된 값을 그대로 유지하고,

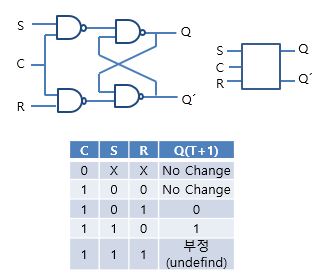
S=0, R=1인 경우 저장된 값을 0으로 Reset, S=1, R=0인 경우 저장된 값을 1로 Set한다.

S=R=1인 경우 유효하지 않은 입력으로 취급한다.

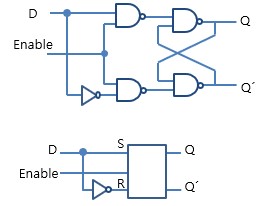
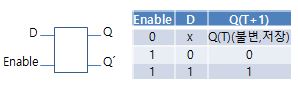
위 SR Latch는 NOR 게이트를 사용해 만들었는데, NAND 게이트를 사용해서 SR Latch를 만들 수도 있다. 이는 아래와 같은 S’R’ Latch로, 각각의 입력이 S, R 대신 S’, R’이 들어가게 된다.



위의 SR Latch들은 제어 신호가 없어 입력 신호(S, R)이 바뀌면 즉시 저장된 값에 입력값의 변화가 반영되는데, 각각의 입력에 제어 신호(C)를 추가해 제어 신호가 1일 때만 결과값에 변화가 일어나게 하게 만든 SR Latch를 아래와 같은 Gated SR Latch라 한다.



위와 같은 Gated SR Latch에 클럭 신호(CLK)와 결합시키면 SR 플립플롭이 된다.



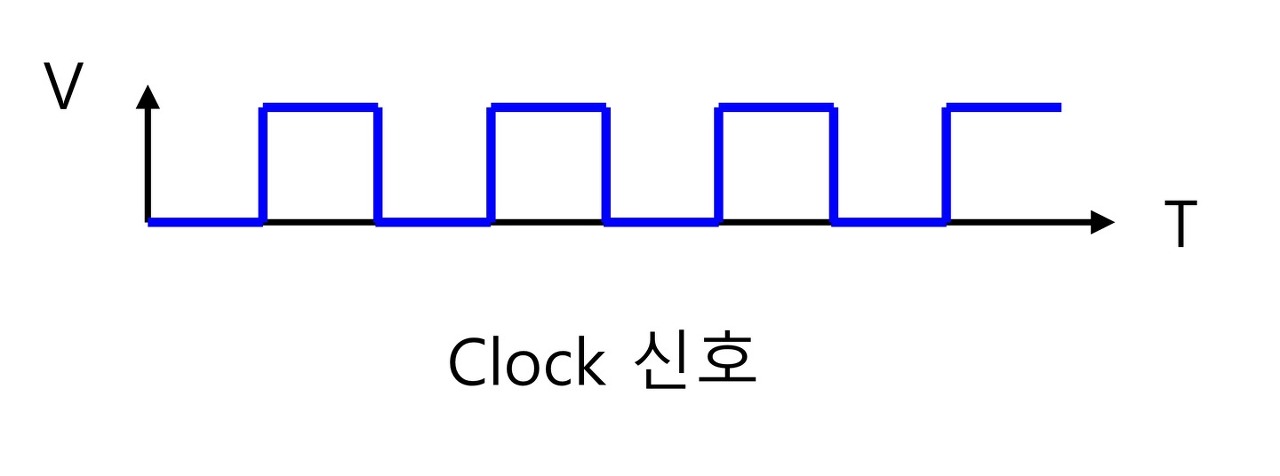
D Latch는 위와 같이 Gated SR Latch의 두 입력 신호 S, R을 각각 D, D’로 입력시킨 회로이다. D = 0이면 저장된 값을 0으로 Reset하고, D = 1이면 저장된 값을 1로 Set한다.

D Latch도 Gated SR Latch와 마찬가지로 클럭 신호(CLK)와 결합시키면 D 플립플롭이 된다.

**6.**

Clock은 순차 논리 회로에서 gate들이 일정 시간 주기의 delay를 갖도록 만드는 신호를 의미한다. Clock은 0과 1을 번갈아가며 출력하는 주기적인 Square wave이다.

컴퓨터 내부의 시계와도 같은 역할을 하며, 컴퓨터 내부의 모듈들의 동기화를 이루는데 사용된다. 아래는 Clock 신호를 시간에 따라 나타낸 그림이다.

****

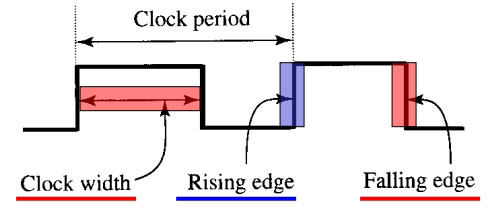
Clock 신호의 한 사이클의 길이를 주기라 하며, 주기의 역수는 Clock의 주파수를 의미한다.

Clock 신호의 주기가 짧을수록 초당 실행 시킬 수 있는 명령어의 수가 늘어나므로 컴퓨터의 속도는 빨라지게 된다.

Clock 신호의 요소로는 신호의 크기, 주기를 결정하는 주파수, 한 주기동안 High와 Low상태의 시간 비인 듀티비가 있다.

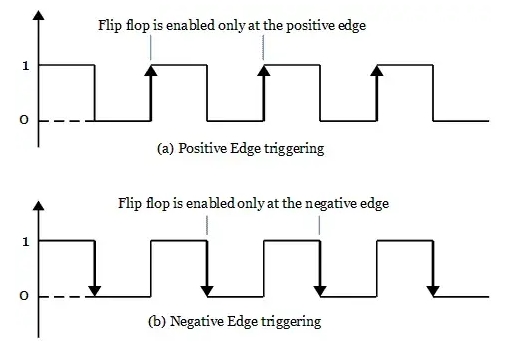
Clock 신호의 크기는 전압으로 나타내며, 디지털 회로에서 전압은 보통 논리게이트의 전압과 같게 설계한다. 일반적으로 0V(ground) 상태를 Low로, Vcc(5V, 3.3V 등)상태를 High 상태로 발생시킨다.

주파수는 수정 발진기(crystal oscillator)를 사용해서 만들고, 일반적으로 50%의 듀티비를 갖는 Square wave를 사용한다.

****

Clock 신호가 Low에서 High로 변할 때를 Rising edge라 하고, High에서 Low로 변할 때는 Falling edge라 한다.

**7.**

****

Trigger는 입력값이 결과에 반영되는 순간을 말하며, Trigger는 Clock신호의 어떤 부분에서이벤트를 발생시키느냐에 따라 Level Trigger와 Edge Trigger로 나뉜다.

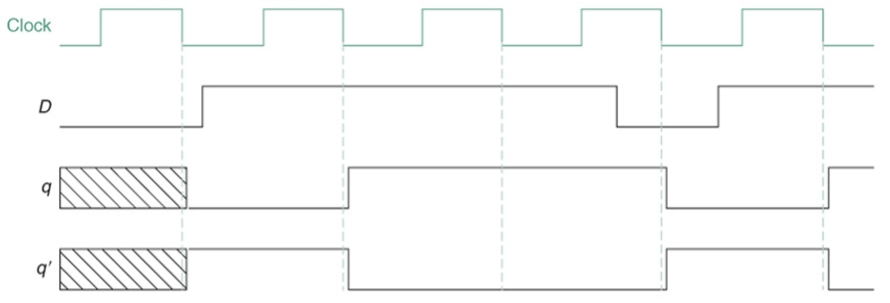
Edge Trigger는 클럭의 값이 변하는 그 순간에만 동작하는 방식이다. Edge Trigger는 위 그림과 같이 Positive edge Trigger와 Negative edge Trigger로 나눠진다.

Positive edge Trigger는 클럭이 0에서 1로 상승할 때 플립플롭이 동작하는 방식이며,

클럭 신호가 0에서 1로 변하는 순간(Rising edge)를 검출한다.

반면에, Negative edge Trigger는 클럭이 1에서 0으로 하강할 때 플립플롭이 동작하는 방식이며, 클럭 신호가 1에서 0으로 변한 순간(Falling edge)를 검출한다.

상태가 0 또는 1에서 변하지 않는 경우에는 Edge Trigger가 이벤트를 발생시키지 않는다.



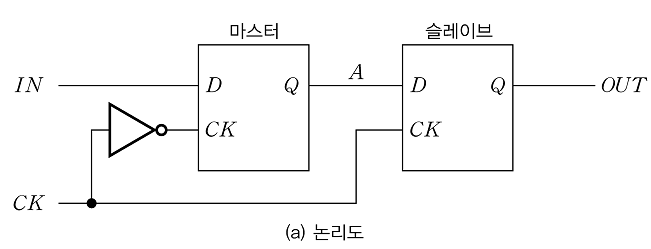
위는 D 플립플롭의 타이밍도이다. 사진을 보면 입력 D에 의한 결과값의 변화가 바로 적용되는 것이 아니라, Clock 신호의 Falling Edge 부분에서 Q와 Q’의 값이 변하는 것을 확인할 수 있다. (Negative edge Trigger)

**8.**

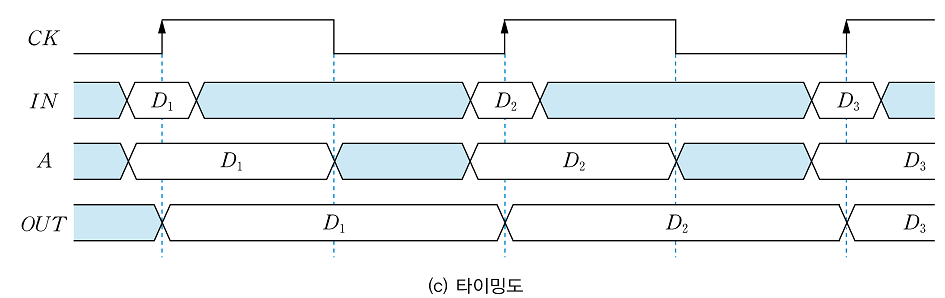
Master-Slave는 어떤 하나의 장치가 다른 장치를 제어하도록 여러 개를 연결하는 것을 말한다. Master는 하나의 일을 수행할 때 동작의 주체를 담당하고, 다른 Slave들을 제어한다.

Slave는 Master에 종속되어 있으며, Master의 지시에 따라 행동한다.

순차 논리회로에서는 주로 두 개의 Latch 또는 플립플롭을 연결하여 Master-Slave 구조를 사용한다.



위는 D Latch 2개를 Master-Slave 구조로 연결하여 만든 D 플립플롭 회로이다.



CK가 0이면 마스터가 입력(D)를 출력(A)로 내보내고, 슬레이브는 동작하지 않는다.

반대로, CK가 1이면 마스터가 동작하지 않고, 슬레이브는 들어오고 있던 입력(A)가 출력(OUT)으로 값을 내보낸다.

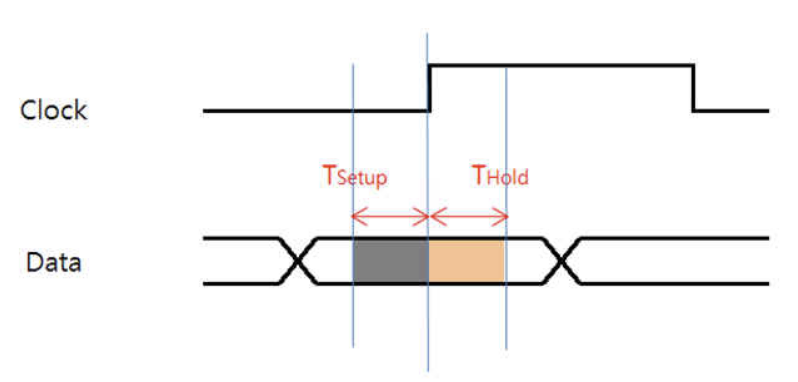
위 타이밍도에서 CK가 rising edge 직전, 마스터에 입력 D가 1이 되어 출력 A가 1이 되고, rising edge 직후, CK가 1이 되어 마스터가 동작하지 않고, 슬레이브가 입력값 A(=1)을 받아 그대로 출력 OUT으로 1을 내보낸다. 또한, falling edge 직후, CK가 0이 되어 마스터가 입력값 0을 받아 A로 출력하지만, 슬레이브가 비활성화 되어 OUT의 값은 1로 유지된다.

마지막으로, 다음 주기의 rising edge 직전 입력 D가 1이 되어 위 과정을 계속 반복하게 되어 OUT의 값은 rising edge에 0에서 1로 변하고, 1이 된 다음부터는 쭉 1로 유지하게 된다. 이 결과는 rising edge에 동작하는 D 플립플롭과 동일하게 동작했음을 보여준다.

따라서 Master-Slave 구조를 사용해 2개의 Latch를 결합하면 플립플롭을 만들 수 있다.

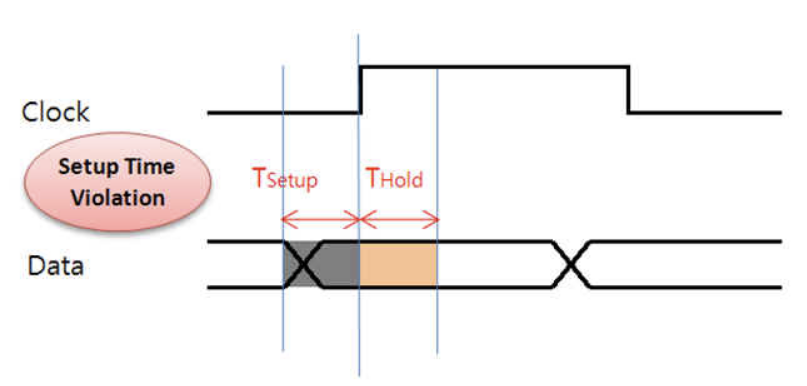
**9.**

- Setup Time과 Hold Time

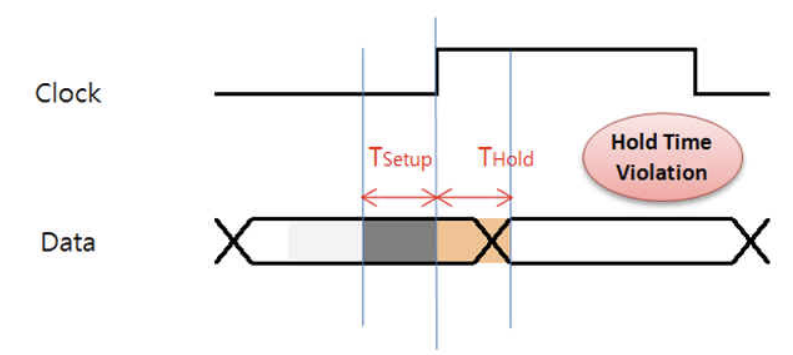


Setup Time()은 활성 에지 직전에 입력 D의 값을 불변으로 유지해야 하는 최소 시간의 길이를 의미한다. 즉, Switching이 일어나기 전까지 입력이 정확히 인식되는데 필요한 최소 유지 시간을 의미한다.

Hold Time()는 활성 에지 직후에 D의 값을 불변으로 유지해야 하는 최소 시간의 길이를 의미한다. 즉, Switching이 일어난 후 상태의 변화가 정확히 인식되도록 필요한 최소 시간을 의미한다.



위 그림은 Setup time 중에 Data의 값이 변경되어 Data의 값을 정확하게 인식하지 못해 오작동이 발생 할 수 있다. (Setup Time Violation)



위 그림은 Hold time 중에 Data의 값이 변경되어 Data의 값을 정확하게 인식하지 못해 오작동이 발생 할 수 있다. (Hold Time Violation)

따라서 순차 논리회로 설계시 Setup Time과 Hold Time Violation이 발생하지 않기 위해 Clock의 Delay와 Data의 Delay를 고려하여 설계해야 한다.