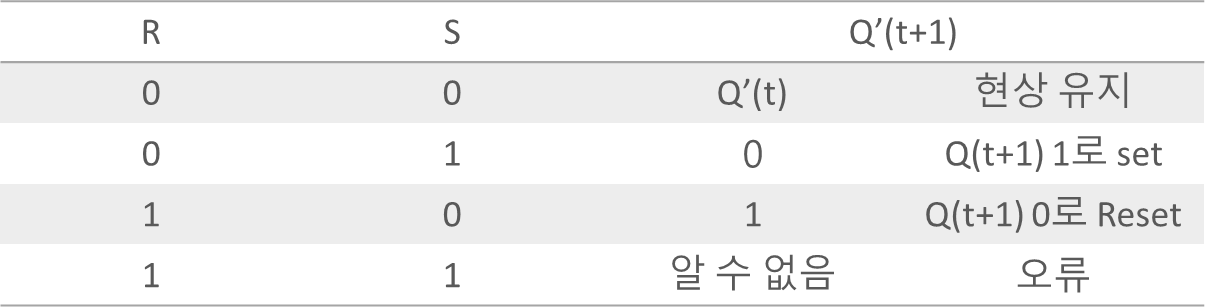
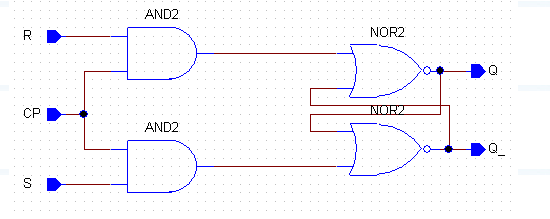
11주차 예비보고서

전공: 물리학과 학년: 3학년 학번: 20191286 이름: 김나현

1. RS 플립플롭에 대해서 조사하시오.

RS 플립플롭은 가장 기본적인 형태의 플립플롭으로, 두 개의 입력 단자 S와 R을 갖고 두 개의 출력 단자 Q(t)와 Q’(t+1)을 갖습니다. 이때, 두 개의 입력 단자 중 S는 Set을 의미하고, R은 Reset을 의미합니다. 따라서, RS 플립플롭은 S와 R의 입력에 따라 Set 또는 Reset을 하는 플립플롭이라고 생각할 수 있습니다. RS 플립플롭의 진리표를 그려보면 다음과 같습니다.

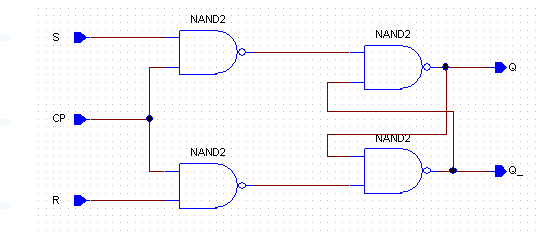
<Table 1> RS 플립플롭의 진리표

R와 S이 모두 0으로, 비활성화 상태인 경우, Q’(t+1)의 출력 단자의 출력은 ~Q(t), 즉 현재 출력의 보수와 동일하고 S만 활성화 상태인 경우, Q’(t+1)은 0이 되므로 Q(t+1)은 1로 Set되고, R만 활성화 상태인 경우, Q(t+1)은 0으로 Reset되므로 Q’(t+1)은 1이 되고, S와 R이 동시에 활성화 상태인 경우, 오류가 발생하므로 Q(t+1)과 Q’(t+1)의 출력을 알 수 없게 됩니다. 이처럼 RS 플립플롭은 가장 기본적이고 간단한 형태의 플립플롭이라는 장점이 있지만 두 입력이 모두 활성화될 때에는 오류가 발생하고 심한 경우에는 하드웨어의 손상까지 야기할 수 있다는 단점이 있습니다. 다음 <Figure 1>은 RS 플립플롭의 schematic입니다.

<Figure 1> RS 플립플롭의 schematic

<Figure 1>을 보며 RS 플립플롭의 동작에 대해 예상해보면 R과 S가 모두 0일 때는 위, 아래 AND 게이트가 모두 0이 되어 Q(t)가 0일 때는 Q’(t+1)에 1이 출력되고, Q(t)가 1일 때는 (t+1)에 0이 출력되므로 R과 S가 모두 0일 때는 Q’(t+1)에서 Q’(t)를 출력된다는 사실을 알 수 있습니다. 또한, S만 1일 때는 Q(t)가 0이든, 1이든 상관없이 Q’(t+1)에서 0을 출력하고, R만 1일 때는 Q(t)가 0이든, 1이든 상관없이 Q(t+1)에서 1을 출력한다는 것을 알 수 있습니다.

위 <Figure 1>과 같이, AND 게이트와 NOR 게이트로 구현된 RS 플립플롭은 NAND 게이트만을 이용하여 다음 <Figure 2>와 같이 구현할 수 있습니다.

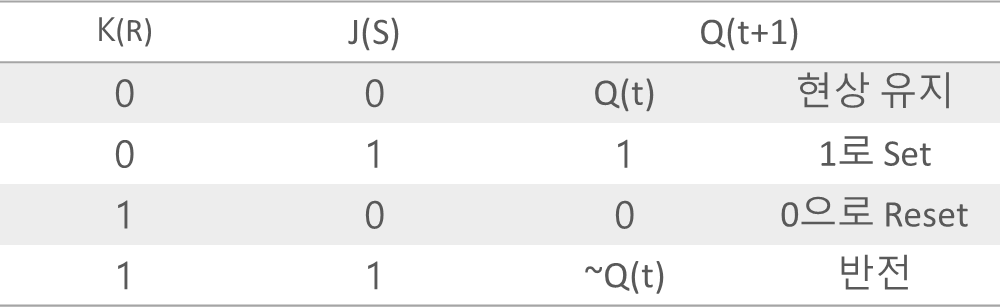


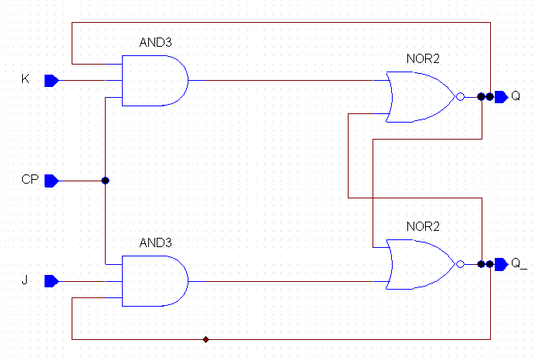
<Figure 2> NAND 게이트만을 이용하여 구현한 RS 플립플롭의 schematic

1. JK 플립플롭에 대해서 조사하시오.

JK 플립플롭은 R과 S가 동시에 활성화되었을 때 오류가 발생하는 RS 플립플롭의 단점을 보완한 플립플롭으로, R 대신 K를, S 대신 J를 사용하여 표현하고 RS 플립플롭과 마찬가지로 Q(t)과 Q(t+1), 두 개의 출력 단자를 가집니다. 이때, R와 S가 모두 1일 때 오류가 발생하였던 RS 플립플롭과 달리, JK 플립플롭은 J와 K가 모두 1일 때 Q(t)을 반전시킨, ~Q(t)을 출력합니다. JK 플립플롭의 진리표를 그려보면 다음과 같습니다.

<Table 2> JK 플립플롭의 진리표

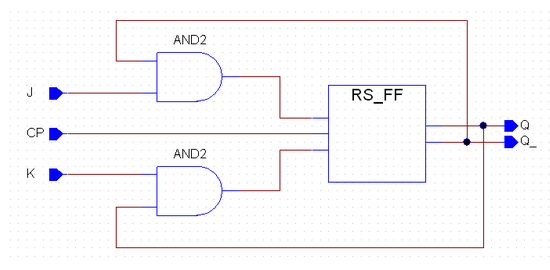


JK 플립플롭에서 K는 RS 플립플롭의 R과 J는 RS 플립플롭의 S와 동일한 역할을 하므로 JK 플립플롭에서 J와 K가 모두 비활성화 상태인 경우, RS 플립플롭에서 S와 R이 모두 비활성화 상태인 것과 동일하게 Q(t+1)에서 Q(t)을 그대로 출력하게 되고 J만 활성화 상태인 경우, RS 플립플롭에서 S만 활성화 상태인 것과 동일하게 Q(t+1)이 1로 Set되며 S만 활성화 상태인 경우, RS 플립프롭에서 R만 활성화 상태인 것과 동일하게 Q(t+1)이 0으로 Reset됩니다. 다만, RS 플립플롭과 달리 JK 플립플롭은 J와 K가 모두 활성화 상태인 경우, 오류가 발생하지 않고 Q(t+1)에서 Q(t)의 보수인 ~Q(t)를 출력하게 됩니다. 다음 <Figure 3>은 JK 플립플롭의 schematic입니다.

<Figure 3> JK 플립플롭의 schematic

RS 플립플롭과 달리, JK 플립플롭은 두 개의 입력 J, K에 대해 각각 K와 Q, clock 펄스를 AND 연산, J와 Q’, clock 펄스를 AND 연산함으로써 J와 K가 모두 1일 때도 오류가 발생하지 않고 Q(t)이 1이었으면 R만 1인 RS 플립플롭처럼 Q(t+1)에서 0을 출력하고 Q(t)가 0이었으면 S만 1인 RS 플립플롭처럼 Q(t+1)에서 1을 출력하는, 즉 Q(t)의 보수를 출력합니다.

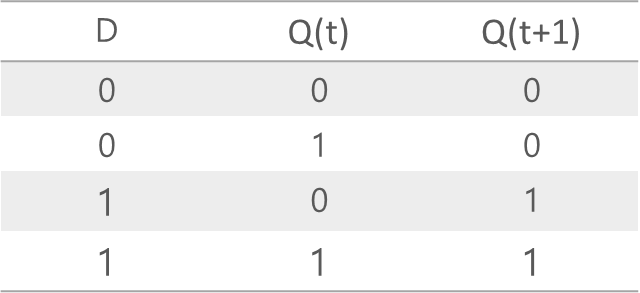
위 <Figure 3>과 같이, AND 게이트와 NOR 게이트로 구현된 JK 플립플롭은 RS 플립플롭에 두 개의 AND 게이트를 추가하여 다음 <Figure 4>와 같이 구현할 수 있습니다.



<Figure 4> RS 플립플롭에 두 개의 AND 게이트를 추가하여 구현한 JK 플립플롭의 schematic

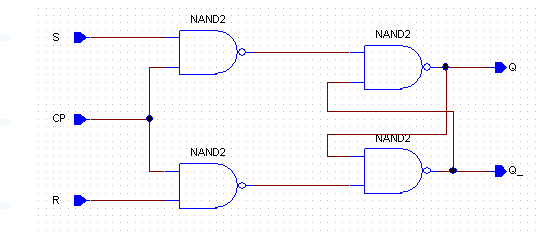
1. D 플립플롭에 대해서 조사하시오.

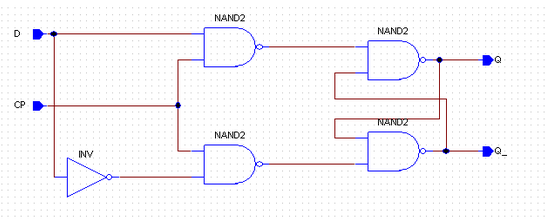
D 플립플롭은 앞선 두 개의 플립플롭과 달리, 하나의 입력 단자를 갖고, 현재 상태와 무관하게 입력을 출력 단자 Q(t+1)에서 그대로 출력한다는 특징이 있습니다. 입력을 그대로 출력한다는 D 플립플롭의 특성에 의해 delay를 의미하는 D를 붙여 D 플립플립이라고 명명합니다. D 플립플롭에서 D는 RS 플립플롭의 S의 역할을, ~D는 RS 플립플롭의 R의 역할을 합니다. D 플립플롭의 진리표를 그려보면 다음과 같습니다.

<Table 3> D 플립플롭의 진리표

위의 <Table 3>을 보면, D가 0일 때는 RS 플립플롭의 S가 0이고, R이 1일 때와 같이 Q(t+1)이 0으로 Reset되고 D가 1일 때는 RS 플립플롭의 S가 1이고, R이 0일 때와 같이 Q(t+1)이 1로 Set됩니다. 이때, Q(t+1)은 Q(t)과 무관하게 D에만 의존하여 결정된다는 것을 알 수 있습니다.

D 플립플롭은 SR 플립플롭의 입력 중 S에 D를 입력하고, R에 D의 보수를 입력함으로써 구현할 수 있으므로 앞에서 설명한 NAND 게이트만을 이용하여 구현한 RS 플립플롭, <Figure 2> 회로를 일부 수정하여 다음 <Figure 5>와 같이 구현할 수 있습니다.



<Figure 2> NAND 게이트만을 이용하여 구현한 RS 플립플롭의 schematic

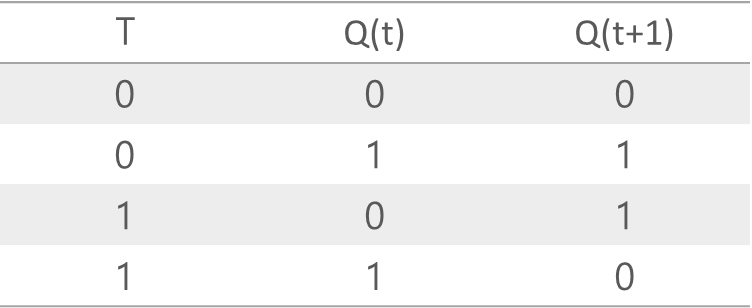
<Figure 5> NAND 게이트를 이용하여 구현한 D 플립플롭의 schematic

위의 <Figure 5>와 같이, D 플립플롭에서는 D가 RS 플립플롭의 S, D’가 RS 플립플롭의 R과 같은 역할을 하기 때문에 D가 1일 때는 Q(t+1)을 항상 1로 Set하고, D가 0일 때는 Q(t+1)을 항상 0으로 Reset한다는 것을 알 수 있습니다.

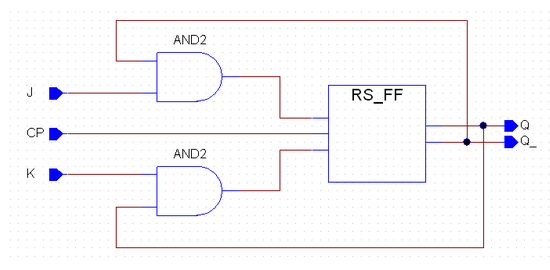
1. T 플립플롭에 대해서 조사하시오.

T 플립플롭은 JK 플립플롭의 J와 K, 두 개의 입력 단자를 연결하여 하나의 입력 단자, T를 갖는 플립플롭으로 구현한 것으로, 이때 T는 Toggle이나 Trigger의 의미를 가집니다. T 플립플롭은 T가 0일 때는 Q(t+1)에서 현재 출력인 Q(t)을 그대로 출력하고, T가 1일 때는 Q(t+1)에서 현재 출력인 Q(t)를 반전한 ~Q(t)를 출력하게 됩니다. T 플립플롭의 진리표를 그려보면 다음과 같습니다.

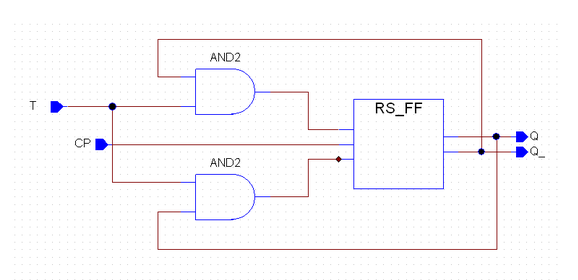
<Table 4> T 플립플롭의 진리표



위의 <Table 4>를 보면 T와 Q가 서로 같을 때는 Q(t+1)이 0이 되고, T와 Q가 서로 다를 때는 Q(t+1)이 1이 되므로 T 플립플롭의 동작이 마치 XOR 게이트와 같다는 것을 알 수 있습니다.

T 플립플롭은 JK 플립플롭의 입력 중 J에 T와 Q’의 AND 연산한 값을 입력하고, K에 T와 Q의 AND 연산한 값을를 입력함으로써 구현할 수 있으므로 앞에서 설명한 RS 플립플롭에 AND 게이트 두 개를 추가하여 구현한 JK 플립플롭, <Figure 4> 회로를 일부 수정하여 다음 <Figure 6>과 같이 구현할 수 있습니다.

<Figure 4> RS 플립플롭에 두 개의 AND 게이트를 추가하여 구현한 JK 플립플롭의 schematic



<Figure 6> JK 플립플롭을 수정하여 구현한 T 플립플롭의 schematic

위의 <Figure 6>과 같이, T 플립플롭에서는 JK 플립플롭의 J 입력 대신 T와 Q를 AND 연산한 값이 들어가므로 T가 0이면 JK 플립플롭에서 J와 K가 모두 비활성화된 것과 같으므로 출력이 변하지 않아 Q(t+1)이 Q(t)가 되고, T가 1이고 Q(t)가 1이면 JK 플립플롭에서 K가 활성화되는 것과 같으므로 Q(t+1)가 0이 되고, T가 1이고 Q(t)가 0이면 JK 플립플롭에서 R이 활성화되는 것과 같으므로 Q(t+1)가 1이 됩니다.

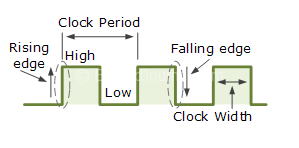
1. Latch의 기능에 대해서 조사하시오.

플립플롭과 래치의 가장 큰 차이점은 플립플롭은 edge trigger 타입의 메모리 회로인 반면, 래치는 level trigger 타입의 메모리 회로라는 것입니다. 따라서, 래치의 출력은 여러 개의 입력 중 단 한 개의 입력이라도 변화가 있으면 출력이 즉각적으로 변하게 됩니다.

래치는 플립플롭과 마찬가지로 S, R 두 개의 입력 단자를 갖는 SR 래치나 SR 플립플롭에서 S와 R을 연결하여 구현한 D 플립플롭과 같이, SR 래치에서 S과 R을 연결하여 구현한 D 래치 등이 있습니다.

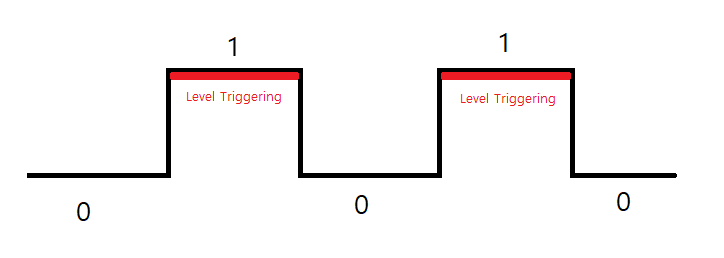
1. Clock의 기능에 대해서 조사하시오.

Clock은 synchronous system에서 동기화를 위해 사용되는 신호로, 논리 상태 High와 논리 상태 Low로 구성된 주기적인 구형파 펄스를 의미합니다. 클럭 펄스에서 한 사이클의 길이를 주기, T라고 하며 주기의 역수는 클럭의 주파수를 의미합니다. Clock의 주기에서 논리상태가 High인 구간을 clock width라고 하는데 clock의 주기에서 clock width가 차지하는 비율을 duty cycle이라고 하며 일반적으로 synchronous 디지털 논리 회로에서는 duty cycle이 50 % 이하인 구형파 펄스를 이용하여 동기화합니다. Clock에서 논리 상태 Low에서 논리 상태 High로 전환되는 지점을 상승 에지 또는 positive edge라고 하며 논리 상태 High에서 논리 상태 Low로 전환되는 지점을 하강 에지 또는 negative edge라고 합니다.

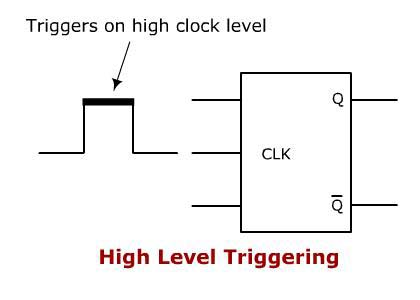


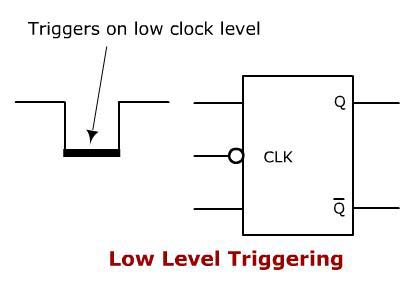
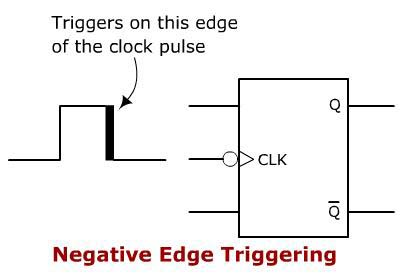
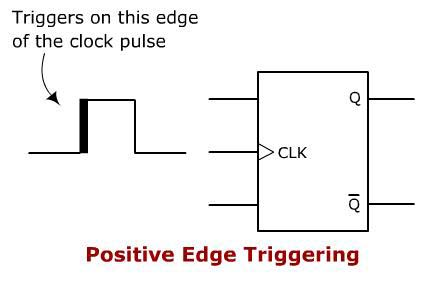
<Figure 7> clock pulse

1. Edge-Trigger의 특성에 대해서 조사하시오.

광장이(가) 표시된 사진

자동 생성된 설명트리거란 논리 회로의 출력이나 상태 변화를 촉발하는 기동 신호를 의미하며 트리거의 종류에는 level trigger, edge trigger가 있습니다. Level trigger는 앞서 설명하였던 래치 중 clocked latch에 사용되는 트리거 방법으로, 논리 상태가 High이거나 Low일 때만 입력 데이터를 받아들이는 것이고 Edge trigger는 대부분의 플립플롭에서 사용되는 트리거 방법으로, 클럭 천이가 발생하는 지점에서만 입력 데이터를 받아들이는 것입니다. 논리 상태가 High일 때를 트리거로 인식하는 것은 positive level trigger이고, 논리 상태가 Low일 때를 트리거로 인식하는 것은 negative level trigger입니다. 클럭 천이 중 상승 에지를 트리거로 인식하는 것은 상승 에지 트리거이고, 하강 에지를 트리거로 인식하는 것은 하강 에지 트리거입니다.

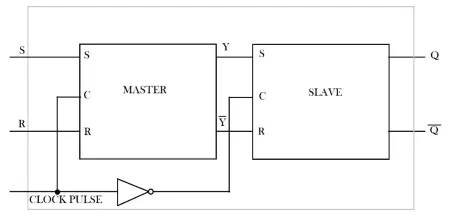
<Figure 8> Trigger



<Figure 9> Trigger의 기호

1. Master-Slave의 개념에 대해서 조사하시오.

Master-slave 플립플롭은 직렬로 연결된 두 개의 플립플롭을 의미하며 두 개의 플립플롭 중 하나는 Master 플립플롭으로 동작하고, 다른 하나는 Slave 플립플롭으로 동작하게 됩니다. 각 플립플롭은 서로 상보적인 clock 펄스에 연결되게 되는데, 하나의 플립플롭이 하강 에지 트리거에 의해 동기화된다고 하면 다른 하나의 플립플롭은 상승 에지 트리거에 의해 동기화됩니다. 다음과 같은 Master-Slave 플립플롭 논리 회로에서 clock 펄스가 상승 에지일 때, Master 플립플롭은 활성화 상태가 되고 Slave 플립플롭은 비활성화 상태가 됩니다. 반면, clock 펄스가 하강 에지일 때는 Master 플립플롭은 비활성화 상태가 되고 Slave 플립플롭은 활성화 상태가 됩니다



<Figure 10> Master-Slave 플립플롭의 예시