11주차 예비보고서

전공: 아트&테크놀로지학과 학년: 4학년 학번: 20191048 이름: 김도솔

**1.**

**RS 플립-플롭**은 가장 기본적인 형태의 플립 플롭으로, 두 개의 입력(R과 S)과 두 개의 출력(Q와 ~Q 또는 Q')을 가지고 있다. 여기서 R은 "Reset"을 나타내고, S는 "Set"을 나타낸다. RS 플립 플롭은 주로 디지털 시스템에서 데이터를 저장하거나 전달하는 데 사용된다.

RS 플립-플롭은 다음과 같이 동작한다.

* Set (S) 입력이 활성화될 때: Set 입력이 1로 설정되면 플립 플롭의 출력(Q)은 1이 된다. 이는 플립 플롭을 "설정" 상태로 만든다.
* Reset (R) 입력이 활성화될 때: Reset 입력이 1로 설정되면 플립 플롭의 출력(Q)은 0이 된다. 이는 플립 플롭을 "리셋" 상태로 만든다.
* 두 입력이 모두 0인 경우: 어떠한 입력도 활성화되지 않으면, 플립 플롭의 상태는 이전 상태를 유지한다.
* 두 입력이 모두 1인 경우 (금지 상태): RS 플립 플롭은 두 입력이 동시에 1로 설정되는 것을 허용하지 않는다. 이 상태에서는 플립 플롭의 동작이 정의되지 않기 때문이다.

RS 플립 플롭은 순차 논리 회로에서 사용되며, 레지스터, 카운터, 메모리 등 다양한 디지털 시스템에서 중요한 역할을 한다. 다양한 변형이 존재하며, 뒤에서 알아볼 JK 플립 플롭, D 플립 플롭 등이 이에 해당된다.

|  |  |
| --- | --- |
| Circuit diagram | Truth table |
| https://blog.kakaocdn.net/dn/ZNzKc/btrbyIa2dpY/p9FtzOk1AgYNmhpQNkU1r0/img.png | https://velog.velcdn.com/images%2Ftonyhan18%2Fpost%2F499756e5-0fbe-4437-8fa6-e724e49bc336%2Fimage.png |

**2.**

**JK 플립-플롭**은 앞서 알아본 RS 플립 플롭의 확장형으로, RS 플립 플롭의 한계를 극복하기 위해 고안된 플립 플롭이다. JK 플립 플롭은 세 개의 입력(J, K, 그리고 클럭)과 두 개의 출력(Q와 ~Q 또는 Q')을 가지고 있다. 여기서 J는 "Set"을 나타내고, K는 "Reset"을 나타낸다. JK 플립 플롭은 RS 플립 플롭에서 발생하는 금지 상태(두 입력이 모두 1)를 피할 수 있다.

JK 플립 플롭은 다음과 같이 동작한다.

* J 입력이 활성화될 때: J 입력이 1로 설정되면, 클럭 신호가 활성화될 때 플립 플롭의 출력(Q)은 1이 된다.
* K 입력이 활성화될 때: K 입력이 1로 설정되면, 클럭 신호가 활성화될 때 플립 플롭의 출력(Q)은 0이 된다.
* J와 K가 모두 0인 경우: 어떠한 입력도 활성화되지 않으면, 플립 플롭의 상태는 이전 상태를 유지한다.
* J와 K가 모두 1인 경우: J와 K가 모두 1로 설정되면, 출력(Q)이 이전 상태의 반전된 값으로 토글된다. 즉, 1이면 0으로, 0이면 1로 바뀐다.

JK 플립 플롭은 시퀀셜 논리 회로에서 사용되며, 다양한 카운터 및 시프트 레지스터 디자인에 활용된다.

|  |  |
| --- | --- |
| Circuit diagram | Truth table |
| https://blog.kakaocdn.net/dn/cETY79/btrbq37V3PB/IMTzgFsetnb9dpRVzrdk71/img.png | https://velog.velcdn.com/images%2Ftonyhan18%2Fpost%2F59e53942-75a7-49ca-acd1-086023f851b9%2Fimage.png |

**3.**

**D 플립-플롭**은 디지털 논리 회로에서 사용되는 또 다른 종류의 플립 플롭으로, D는 "데이터(Data)"를 나타낸다. D 플립 플롭은 하나의 입력(D), 클럭(C), 그리고 두 개의 출력(Q와 ~Q 또는 Q')을 가지고 있다.

D 플립 플롭은 다음과 같이 동작한다.

* 입력(D)이 클럭 신호의 상승 에지(또는 하강 에지)에서 활성화될 때: 클럭 신호가 상승 에지에 도달할 때(또는 하강 에지에 도달할 때), 입력(D)의 값이 출력(Q)에 복사된다. 즉, D 플립 플롭은 클럭이 변할 때 입력 값을 저장하고 출력으로 내보낸다.
* 입력(D)이 변하지 않는 경우: 클럭 신호가 변할 때 입력(D)의 값이 변경되지 않으면, 출력(Q)은 이전의 값으로 유지된다.

D 플립 플롭은 단순하면서도 유용한 순차 논리 디바이스로, 데이터 레지스터, 메모리 셀, 그리고 다양한 디지털 논리 회로에서 사용된다.

|  |  |
| --- | --- |
| Circuit diagram | Truth table |
| https://blog.kakaocdn.net/dn/IBos0/btrbAJmO1vI/KNzxWTDK0lcOCiRJCXl0g1/img.png | https://blog.kakaocdn.net/dn/IBos0/btrbAJmO1vI/KNzxWTDK0lcOCiRJCXl0g1/img.png |

**4.** T 플립-플롭에 대해서 조사하시오.

**T 플립 플롭**은 토글(Toggle) 동작을 수행하는 플립 플롭이다. T 플립 플롭은 T 입력, 클럭(C), 그리고 두 개의 출력(Q와 ~Q 또는 Q')으로 구성되어 있다.

T 플립 플롭은 다음과 같이 동작한다.

* T 입력이 클럭 신호의 상승 에지(또는 하강 에지)에서 활성화될 때: 클럭 신호가 상승 에지에 도달할 때(또는 하강 에지에 도달할 때), T 입력의 값이 현재 상태의 출력(Q)에 XOR(배타적 논리합) 연산을 수행한다. 즉, T가 1이면 현재 상태의 출력이 토글된다.
* T 입력이 0인 경우: T 입력이 0이면 아무런 변화가 없다. 출력(Q)은 이전의 값으로 유지된다.

T 플립 플롭은 주로 카운터, 시프트 레지스터, 디바이더, 그리고 타이밍 회로 등에서 사용된다. 특히, T 플립 플롭은 시퀀셜 논리 회로에서 이전 상태의 반전과 같은 간단한 토글 동작이 필요한 경우에 유용하게 쓰인다.

|  |  |
| --- | --- |
| Circuit diagram | Truth table |
| https://blog.kakaocdn.net/dn/7rvoc/btrbuKfLFvW/hbGP4jwb3Jnx9412ahS9BK/img.png | https://blog.kakaocdn.net/dn/ecrrtj/btrbq31dTNd/gbkPncTWtxmHRGAJ7rs0rk/img.png |

**5.**

**Latch**는 1 비트의 정보(0 또는 1)를 보관하고 유지할 수 있는 회로로, 순차 회로의 기본적인 요소이다. 주요 기능은 디지털 논리 회로에서 정보를 저장하고 유지하는 것이다. Latch는 입력 신호의 상태를 기억하며, 특정한 시점에서 입력을 받아서 상태를 갱신할 수 있다. 좀 더 자세히 말하자면 하나 이상의 입력을 받아 회로의 상태를 변경할 수 있으며, 결과는 하나 또는 두 개의 출력으로 표현된다. 이때 출력 및 다음 상태는 입력뿐만 아니라 현재 회로의 상태에 따라 결정될 수 있다. 각 Latch의 유형에 따라 동작이 다르지만, 주로 메모리 셀로 사용되거나 레지스터, 카운터, 시프트 레지스터 등의 순차 논리 회로에서 중요한 역할을 한다. 플립 플롭과 비슷한 역할을 하지만 Latch는 비동기적으로 동작하며 입력 신호의 변화에 따라 즉시 출력을 갱신하는 반면, Flip-Flop은 동기적으로 클럭 신호에 의해 동작하고, 안정성이 높은 동작을 제공한다는 특징이 있다.

**6.**

**Clock**은 일정한 주기로 반복되는 신호로, synchronous system에서 동기화를 위해 사용된다. 이 신호는 high와 low 상태를 번갈아가며 가지며, 디지털 회로의 각 부분이 일정한 순서로 동작할 수 있도록 지시하는 역할을 한다. Clock의 주요 기능은 다음과 같다.

* 동기화(Synchronization): Clock은 전자 장치의 여러 부분 간에 동기화를 제공한다. 시스템의 모든 동작은 주로 Clock의 변화에 의해 조절되며, 이를 통해 각 부분이 일정한 순서로 동작할 수 있다.
* 타이밍 제어(Timing Control): Clock은 작업의 정확한 타이밍을 제어한다. Clock 신호의 주기는 일반적으로 나노초(1억 분의 1초) 또는 피코초(1조 분의 1초)의 단위로 매우 짧다. 따라서 Clock을 통해 정밀한 동작과 데이터 전송이 가능하다.
* 회로 동작 제어(Control of Circuit Operation): Clock이 전자 장치의 작동 주기를 나타내므로, Clock 신호의 상승 또는 하강 에지 등 특정 이벤트가 발생할 때에만 동작하도록 회로를 설계할 수 있다. 이것은 특정 작업이나 데이터 전송이 일어날 때 정확한 타이밍을 유지할 수 있도록 도와준다.
* 전원 소모 감소(Power Consumption Reduction): Clock 신호는 전자 기기의 동작을 제어하는 주기적인 신호이기 때문에, Clock 신호의 주파수를 낮추면 전원 소모를 줄일 수 있다. 이는 전력 효율성을 향상시키는 데 도움이 된다.

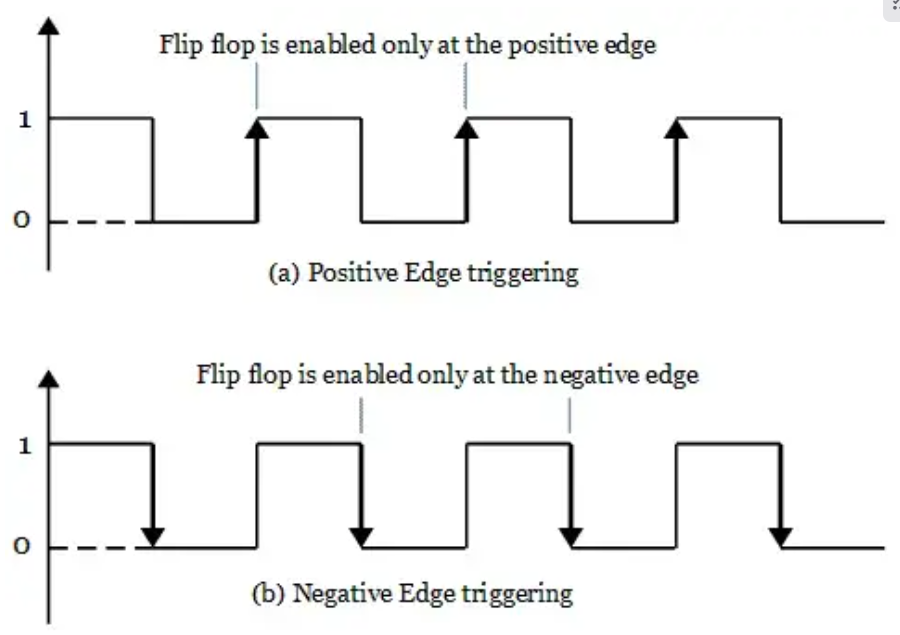
Clock은 일반적으로 Clock Generator에 의해 생성되며, 전자 시스템의 핵심 부분 중 하나이다. 디지털 시스템의 성능과 안정성은 클럭의 정확성과 동기화에 크게 의존하므로, 클럭 설계는 디지털 시스템 설계에서 핵심적인 부분 중 하나로 간주된다.

**7.**

Trigger란 입력 값이 결과에 반영되는 순간을 의미한다. 즉 입력 신호의 순간적인 변화를 말한다. 이는 Level trigger와 Edge trigger로 나뉘는데, **Edge trigger**는 Clock의 값이 변화하는 그 순간에만 동작하는 방식을 말한다. 이때 Edge는 클럭 신호의 변화를 의미하며, 이에 반응하여 동작하는 회로는 Edge-Triggered라고 한다.

Edge-Trigger는 다시 Rising Edge-Trigger와 Falling Edge-Trigger로 나뉜다. Rising Edge-Trigger는 Clock 신호가 0에서 1로 변할 때 동작하는 방식을 말한다. 즉, Clock의 상승 에지(positive edge)에서 동작하며, 이때 입력이나 상태를 샘플링하거나 변경한다. 반대로 Falling Edge-Trigger는 Clock 신호가 1에서 0으로 변할 때 동작한다. 즉, 클럭의 하강 에지(negative edge)에서 동작하며, 이때 입력이나 상태를 샘플링하거나 변경한다.

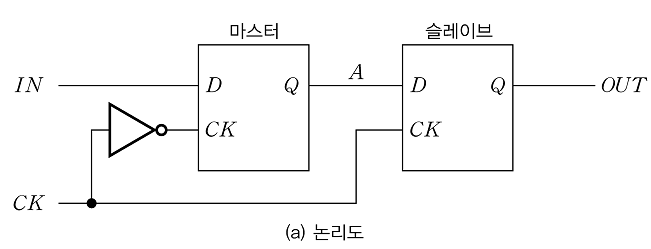
Edge-Trigger는 클럭 신호의 특정 변화에만 반응하므로, 정확한 타이밍 제어가 가능하다. 따라서 회로의 안정성과 신호의 정확성을 보장하는데 중요한 역할을 한다. 또한 클럭 신호가 변화하는 시점에만 데이터를 샘플링하거나 변경하므로, 데이터의 안정성을 보장할 수 있다. 추가적으로 클럭의 변화에만 반응하므로, 클럭이 정지되어 있는 동안에는 전력 소모를 줄일 수 있다.



**8.**

**Master-Slave**는 한 장치나 프로세스(Master)가 하나 이상의 다른 장치나 프로세스(Slave)를 통제하는 비대칭 제어 모델을 말한다. 디지털 회로 시스템에서는 순차 논리 회로에서 사용되는 디자인 패턴을 말하며, 플립 플롭이나 래치를 구성하는 방식 중 하나이다. 이 패턴은 여러 플립 플롭이나 래치를 조합하여 하나의 전체 레지스터를 만들거나, 더 복잡한 순차 논리 회로를 구현하는 데 사용된다.

이러한 개념의 적용 예시로 Master-Slave 플립 플롭에 대해 알아보자. 이는 두 개의 플립 플롭과 한 개의 인버터로 구성된다. 두 플립 플롭은 직렬로 연결되어 있고, 둘 중 하나의 enable 입력에는 인버터가 연결되어 있어 두 플립 플롭은 서로 반대의 Clock 신호를 받는다. Master-slave라는 이름이 붙은 이유는 위와 같은 구조 때문에 연결된 플립 플롭 중 두 번째 플립 플롭이 첫 번째 플립 플롭의 변화에 대해서만 상태가 변하기 때문이다. 이러한 두 번째 플립 플롭을 Slave 플립 플롭이라하고, 첫 번째 플립 플롭을 Master 플립 플롭이라 한다. 아래의 그림에서, Clock이 0일 때, Master 플립 플롭은 활성화 상태가 되어 Input이 Master 플립 플롭으로 들어간다. 이때 Slave 플립 플롭은 비활성화 상태가 되어 Master 플립 플롭으로부터 입력이 들어오나 동작하지 않는다. 반면, Clock이 1이 되면 Master 플립 플롭은 비활성화 상태가 되고, Slave 플립 플롭은 활성화 상태가 되어 Master 플립 플롭에서 들어오고 있던 입력이 Slave latch를 통해 출력으로 나간다. Master-Slave 플립 플롭은 안정성과 신호의 정확성을 보장하기 위해 설계되었다. 하나의 플립 플롭이 클럭 에지 동안 데이터를 갱신하고, 나머지 플립 플롭은 다음 클럭 에지에서 해당 데이터를 안정적으로 저장하므로, 전체적으로 안정성이 높다.



**9.**

플립 플롭은 디지털 논리 회로에서 정보를 저장하고 전달하는데 사용되는 중요한 요소이다. 이는 다양한 응용 분야에서 활용되며, 주로 순차 논리 회로에서 데이터 레지스터, 메모리 셀, 그리고 다양한 제어 회로에 쓰인다. 플립 플롭의 활용 예시는 다음과 같다.

1. 레지스터(Register): 플립 플롭은 레지스터를 구성하는 데 사용된다. 레지스터는 데이터를 저장하는데 주로 활용되며, 이러한 레지스터는 프로세서의 레지스터 파일이나 메모리 주소 레지스터(MAR) 등에 사용된다.
2. 메모리 셀(Memory Cell): 플립 플롭은 메모리 셀을 형성하는 데 사용된다. 메모리 셀은 RAM(Random Access Memory)에서 데이터를 저장하고 읽는 데 사용되며, 플립 플롭은 각각의 비트를 나타내는 데 활용된다.
3. 카운터(Counter): 플립 플롭은 카운터를 만드는 데 사용된다. 카운터는 특정 이벤트나 클럭 신호에 따라 값을 증가시키거나 감소시키는 데 사용되며, 이를 통해 시퀀스를 추적하거나 타이밍을 제어할 수 있다.
4. 시프트 레지스터(Shift Register): 플립 플롭은 시프트 레지스터를 만드는 데 사용된다. 시프트 레지스터는 입력 비트를 한 칸씩 이동시키는 데 사용되며, 데이터의 이동이나 저장에 활용된다.
5. 시퀀셜 논리 회로(Sequential Logic Circuits): 플립 플롭은 시퀀셜 논리 회로에서 상태를 저장하고 제어 신호에 따라 상태를 갱신하는 데 사용된다. 이는 제어 장치(Control Unit)와 같은 회로에서 중요한 역할을 한다.
6. 파이프라인(Pipeline): 플립 플롭은 파이프라인 구조에서 각 단계의 상태를 저장하는 데 활용된다. 파이프라인은 명령어의 처리를 여러 단계로 분할하여 동시에 여러 명령어를 처리하는 구조를 갖는다.