11주차 예비보고서

전공: 컴퓨터공학과 학년: 3학년 학번: 20191599 이름: 송경호

**1.**

플립-플롭은 이진 상태를 저장하고 전환하는 역할을 하는 회로이다. 또한 RS 플립-플롭은 여러 플립-플롭의 종류 중 가장 기본적인 형태이다.

RS 플립-플롭은 두 개의 입력을 받아 두 개의 출력을 발생시키는데 이때 입력 비트가 R과 S이며 이들은 각각 Reset과 Set을 뜻한다. 또한 출력은 각 각 현재 상태와 다음 상태를 나타낸다. 입력 비트 Reset은 출력을 0으로 설정하는 것을 의미하며 Set은 출력을 1로 설정하는 것을 의미한다. 따라서 (R,S)가 (0,1)인 경우 다음 상태는 1, (1,0)인 경우 다음 상태는 0, (0, 0)인 경우 상태가 유지된다. 마지막으로 (1, 1)의 입력은 허용하지 않는다. 이에 따른 진리표는 아래와 같다.

|  |  |  |  |
| --- | --- | --- | --- |
| **S** | **R** | **Q (Prev)** | **Q\* (Next)** |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | - |
| 1 | 1 | 1 | - |

추가적으로 RS는 R과 S의 입력만이 아닌 CP(Clock Pulse)도 입력으로 받는데 CP의 값이 0일 경우 R과 S의 값이 무관하게 Q의 값은 변하지 않으며 CP의 값이 1일 경우 R과 S의 값에 따라 플립-플롭의 입력이 바뀐다. 즉 CP가 일종의 스위치처럼 작용하게 된다.

도표, 라인, 스케치, 폰트이(가) 표시된 사진

자동 생성된 설명

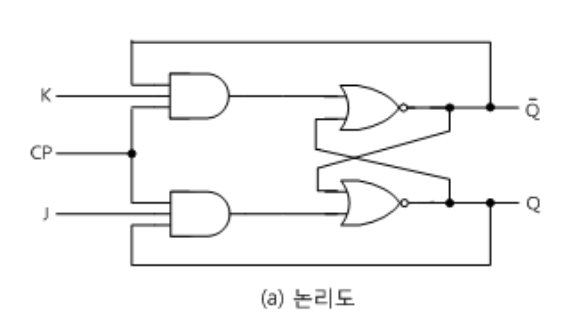
위는 RS 플립-플롭의 회로이며 두개의 AND Gate와 두 개의 NOR Gate로 이루어졌다. 이때 CP의 값이 0일 때 R과 S가 (0,0)인 경우와 같은 입력이 발생하여 값이 유지되는 것을 알 수 있다. 또한 Q와 Q\*의 출력이 서로의 입력으로 작용하는 특이한 형태이다. 또한 해당 회로를 보면 CP와 R 그리고 S가 모두 1일 때 어떠한 Q와 Q\*의 조합으로도 모순이 발생하는 것을 알 수 있다. 따라서 해당 입력은 허용되지 않는다.

**2.**

JK 플립-플롭은 RS 플립-플롭에서 허용하지 않은 R과 S가 (1,1)인 경우에 대한 출력도 가능하도록 보완한 회로이다. JK 플립-플롭도 RS 플립-플롭과 마찬가지로 두개의 입력, J와 K를 받으며 현재 상태와 다음 상태를 나타내는 두 개의 비트를 출력한다. 이때 J는 Jump를 K는 Kill을 의미하며 각 각 RS 플립-플롭의 Set과 Reset와 동일한 역할을 수행한다. 따라서 Jump가 high인 경우 다음 상태는 1, Kill이 high인 경우 다음 상태는 0이 된다. 이때 RS 플립-플롭은 (1,1)의 입력을 허용하지 않은 반면 JK 플립-플롭에서 (1,1)의 입력은 현재 상태를 토글하는 동작을 수행한다. 따라서 (1,1)의 입력에 대해 현재 상태가 0 이었다면 다음 상태는 1로, 현재 상태가 1 이었다면 다음 상태는 0으로 변한다. 아래는 JK 플립-플롭의 진리표이다.

|  |  |  |  |
| --- | --- | --- | --- |
| **J** | **K** | **Q (Prev)** | **Q\* (Next)** |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

추가적으로 JK 플립-플롭에서도 CP 입력은 존재하며 RS 플립-플롭과 마찬가지로 플립-플롭의 스위치로써 작용한다. 다음은 JK 플립-플롭의 회로도이다.



이전의 RS 플립-플롭의 경우 (1,1)이 입력되는 순간 모순이 발생하는 회로의 구성을 띄고 있었다. JK 플립-플롭은 현재와 다음 상태를 나타내는 Q와 Q\*의 출력을 다시 입력과의 AND 연산을 수행하는 방식을 통해 이러한 문제를 해결하고 있다.

**3.**

D 플립-플롭 역시 논리 회로에서 가장 기본적으로 사용되는 플립-플롭 중 하나이다. D 플립-플롭은 하나의 입력, 그리고 현재와 다음 상태를 나타내는 두개의 출력을 갖는 회로로 특이한 점은 다음 상태의 출력으로 입력 D가 그대로 전송된다는 점이다. 따라서 D 플립 플롭은 어떠한 연산을 통해 입력을 변환시켜 출력하는 것이 아닌 신호의 전송 시간을 지연시켜주는 버퍼에 많이 사용되는 플립플롭이다.

|  |  |  |
| --- | --- | --- |
| **D** | **Q (Prev)** | **Q\* (Next)** |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

D 플립-플롭의 입력 D는 Delay 혹은 Data를 의미한다. 또한 현재 상태인 Q에 상관 없이 D가 그대로 Q\*로 전달된 것을 볼 수 있다. 아래는 D 플립-플롭의 회로이다.

도표, 라인, 평면도, 디자인이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 도표, 폰트이(가) 표시된 사진

자동 생성된 설명

D 플립 플롭 역시 CLK, 위 두 플립 플롭의 CP와 동일한 역할을 하는 입력이 존재하며 해당 비트가 입력의 유효성 여부를 결정한다. 각 각 CLK는 clock의 준말이고 cp는 clock pulse의 준말로 다른 용어를 사용한 것 뿐이다. 추가적으로 D 플립 플롭에는 두 가지 종류가 존재하는데 바로 좌측의 회로와 같은 구성을 하는 Trailing-edge Triggered 형태와 CLK에 부정을 취한 Leading-edge Triggered 형태가 있다. 앞서 말했듯, CLK는 입력의 유효성 여부를 결정하므로 Trailing-edge Triggered는 CLK가 1에서 0으로 전환 될 때 입력을 출력으로 전달하게 되고 Leading-edge는 반대로 CLK가 0에서 1로 전환될 때 입력을 출력으로 전달한다.

**4.**

T 플립-플롭은 D 플립-플롭의 변형 형태로 보거나, 앞서 설명한 JK 플립-플롭의 두 입력을 하나로 연결한 것을 볼 수 있다. D 플립 플롭과 마찬가지로 하나의 입력과 현재 상태와 다음 상태를 나타내는 두개의 출력을 갖는다. T 플립 플롭은 입력 T가 0일 경우 현재 상태인 Q를 다음 상태인 Q\*으로 그대로 전달하고, T가 1일 경우 ~Q가 Q\*으로 전달된다.

|  |  |  |
| --- | --- | --- |
| **T** | **Q (Prev)** | **Q\* (Next)** |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

결과론적으로 봤을 때, T 플립-플롭은 T와 Q를 XOR 연산을 한 결과가 Q\*임을 알 수 있으며 T 플립-플롭 역시 CP 펄스를 통해 입력의 유효성을 조작할 수 있으므로 아래와 같은 회로의 형태를 띄게 된다.

도표, 라인, 스케치, 기술 도면이(가) 표시된 사진

자동 생성된 설명

T 플립 플롭은 이러한 특성을 살려 주기적인 상태 전환을 수행하는데 활용된다. CP를 주기적으로 Set과 Unset 하며 간단한 방식으로 토글 동작을 구현할 수 있다.

**5.**

Latch는 정보를 보관하고 유지하는 데 사용되는 1비트 회로이다. Latch의 가장 큰 특징은 입력이 변화할 경우 그 즉시 해당 변화가 결과에 반영된다는 점이다. 이전에 설명한 4개의 플립-플롭은 Clock 신호가 변경되는 엣지에서만 신호가 반영된다. 그러나 Latch는 Clock을 필요로 하지 않고 결과가 즉시 반영되기 때문에 회로가 비교적 간단하다. 결과론적으로 Latch는 데이터의 변화 직전까지 일정한 출력을 유지하고, 입력이 변화할 경우 출력도 변화한다.

Latch 역시 플립-플롭과 마찬가지로 RS, JK, D, T의 형태가 존재하며, 각 Latch들은 모두 플립-플롭에서 Clk 입력이 없어진 형태이다. 즉, 플립-플롭에서 Clk 입력이 항상 1인 경우라고 생각할 수 있다.

텍스트, 스크린샷, 도표, 라인이(가) 표시된 사진

자동 생성된 설명

위는 RS Latch의 예시이다. RS 플립 플롭과 동일하지만 Clk가 없다는 것을 알 수 있다. 따라서 S와 R의 입력에 따라 Q와 Q\*이 영향을 받는다. 위에서 첨부한 진리표와 같은 역할을 수행하지만 대신 Clk 비트가 없기 때문에 해당 내용들에 제한이 발생하지 않는다는 점이 플립-플롭과의 차이점이다.

**6.**

Clock은 일정 주기마다 High와 Low 신호를 왕복하는 신호이다.

**라인, 도표, 폰트, 종이접기이(가) 표시된 사진

자동 생성된 설명**

Clock 신호는 Clock Period 주기를 갖고 High와 Low, 다시말해 1과 0사이를 반복하는데 이때 Clock 신호가 0에서 1로 오르는 지점을 Rising edge, 1에서 0으로 내리는 지점을 Falling edge라고 부른다. 또한 Clock period는 nth rising edge에서 n+1th rising edge 혹은 falling edge 간의 간격임을 알 수 있다.

Clock에서의 신호 공급 방식은 크게 두개이다. Autonomous(Master) Clock은 외부의 영향을 받지 않고 자체적으로 신호를 일정 주기마다 변화시키는 Clock이다. 반대로 Slave Clock은 외부의 데이터에 영향을 받아 신호를 변화시켜 공급한다.

**7.**

Edge-Trigger은 앞서 플립-플롭이나 clock에서 언급한 특정 신호 변화를 의미하는 Edge에서 특정 일을 수행하는 것을 의미한다. 주로 Clock에 많이 적용되며 Clock의 상승 엣지 혹은 하강 엣지에서 동작한다.

이에 따라 Edge Trigger는 Rising Edge Trigger와 Falling Edge Trigger 두 가지가 존재한다. 앞서 설명한 플립 플롭의 Trailing-edge Triggered는 Falling Edge-Trigger이며 Leading Edge Trigger는 Rising Edge Trigger이다.

Edge trigger는 회로의 동작을 정확하게 동기화하도록 도와주며 순차 논리 회로가 일관된 타이밍에 동작하도록 도와준다. 따라서 신호의 변화에 따른 동작에 높은 정확성과 안정성을 제공하여 결과론적으로 디지털 시스템의 신뢰성을 향상시킬 수 있다.

텍스트, 도표, 스크린샷, 라인이(가) 표시된 사진

자동 생성된 설명

해당 그림에서 점선으로 표시된 부분이 Clock에서 하강 엣지가 발생하는 부분들이다. q의 값들은 하강 엣지 부분에서 D와 동일한 값으로 업데이트되고 다음 하강 엣지 부분까지 해당 값을 유지한다. 따라서 위 그림은 D 플립 프롭의 Trailing-edge triggered timing diagram임을 알 수 있다.

**8.**

Master-Slave 개념은 주/종속 시스템을 의미한다. 하나의 Master에 대한 여러 Slave가 존재할 수 있으며 Master는 Slave들의 제어를, Slave들은 Master의 제어를 따라 요청된 작업을 수행한다.

Master-Slave 플립-플롭은 두 개의 플립-플롭과 하나의 Inverter로 구성된다. 이 때 두 플립-플롭은 직렬로 연결되어 있는데, 하나의 플립 플롭이 Master를 나머지 하나가 Slave로 작동된다.

도표, 라인, 폰트, 평면도이(가) 표시된 사진

자동 생성된 설명

위는 D 플립 플롭 두개를 연결한 Master-Slave 플립 플롭이다. 왼쪽 플립-플롭이 master이고 오른쪽이 slave이다. Clock이 1인 경우 D의 값이 Q로 전달되게 되며 slave 플립-플롭은 Clock으로 0의 값이 들어가기 때문에 동작하지 않는다. 반대로 Clock이 0이 되는 순간 Q의 값이 그대로 다음 D의 값으로 전달되며 해당 D의 값 역시 Q로 전달되게 된다. 따라서 최종적인 Q는 Clock이 0일 때, D의 입력에 전혀 영향을 받지 않게된다.

**9.**

**9-1) Gated RS Latch**

Gated RS Latch는 제어 입력이 활성화되는 경우에 AND Gate에서 S, R의 값이 그대로 나오게 되고, RS Latch와 동일하게 작동한다. 반면 제어 입력이 활성화되지 않은 경우, AND Gate에서 S와 R의 값이 무시되게 되며 RG와 SG 모두 0이 되어 Q\*의 값은 변화하지 않고 이전과 동일한 값을 유지하게 된다.

텍스트, 도표, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

**9-2) Level Trigger**

Edge Trigger외에도 Level Trigger라는 개념이 존재한다. 특정 시점에 동작하는 Edge Trigger와 달리 Level Trigger는 신호의 레벨에 따라 동작하는 것을 의미한다. 특정 신호의 전압이 특정 레벨에 도달하면 동작하며 해당 레벨을 유지하는 동안 계속해서 동작한다는 특성이 있다.

또한 레벨 트리거는 대부분 비동기적 논리 회로에서 사용된다. 또한 센서 입력, 외부 신호, 등의 특정 조건이 만족되야 할 때 특정 동작이 수행되어야 하는 경우 효과적으로 사용할 수 있다.

그러나 엣지 트리거와 달리 레벨 트리거는 추가적인 시간 제약이나 동기화에 대한 고려가 필요할 수 있다.

**9-3) Flip Flop의 사용 예시**

플립 플롭은 상태를 저장하거나 데이터 동기화에 효과적이다.

1. 레지스터 : 플립-플롭을 이요하여 레지스터를 구성하여 각 비트의 상태를 나태내는데 사용한다.

2. State Machine: 상태 머신에서 상태를 나타내고 전달하는데 플립-플롭을 사용한다. Moore 또는 Mealy machine 등에 사용된다.

3. 메모리셀: 플립 플롭은 메모리 셀의 핵심 구성 요소이다. 여러 플립 플롭이 모여 메모리 셀을 형성하고 이러한 메모리 셀이 모여 RAM과 같은 메모리 장치에서 데이터를 저장한다.