**Moore Machine**

무어 머신(Moore Machine)은 유한 상태 기계(Finite State Machine, FSM)의 한 유형으로, 출력이 현재 상태에 의해서만 결정되는 방식이다. 이 구조는 출력이 입력에 직접적으로 의존하지 않고, 오직 현재 시스템의 상태(State)와 연결되어 있으며, 상태가 변경될 때만 출력이 변화한다.

도표, 라인, 평면도, 기술 도면이(가) 표시된 사진

자동 생성된 설명

텍스트, 원, 폰트, 도표이(가) 표시된 사진

자동 생성된 설명

**무어 머신의 구성 요소**

**1. 입력 (Input):**

시스템에 제공되는 외부 신호 또는 데이터.

입력은 상태 전이에 영향을 주지만, 출력에는 직접적인 영향을 미치지 않는다.

입력 신호는 조합 논리를 통해 처리된다.

**2. 조합 논리 (Combinational Logic):**

현재 상태와 입력을 받아 다음 상태를 결정하는 논리 회로.

입력 신호와 현재 상태를 기반으로 새로운 상태 정보를 계산한다.

**3. 플립플롭 (Flip-Flop):**

클럭 신호에 동기화되어 상태를 저장하는 기억 장치.

플립플롭은 조합 논리에서 제공된 다음 상태 정보를 저장하며, 클럭 신호에 의해 상태가 갱신된다. 이 저장된 상태 정보가 다음 출력과 상태 전이를 결정하는 데 사용된다.

**4. 출력 조합 논리 (Output Combinational Logic):**

현재 상태를 기반으로 출력을 생성하는 논리 회로.

플립플롭에 저장된 현재 상태 정보를 입력으로 받아, 해당 상태에 대한 출력을 결정한다.

**5. 클럭 신호 (CLK):**

플립플롭의 상태 전이를 동기화하는 주기적인 신호.

클럭 신호는 플립플롭이 새로운 상태 정보를 받아들이고, 이를 출력으로 연결하도록 동기화한다.

**6. 출력 (Output):**

시스템의 외부로 나가는 신호 또는 데이터.

출력은 현재 상태에 의해서만 결정된다.

현재 상태가 변화할 때만 출력이 변경된다.

**무어 머신의 동작 원리**

**1. 초기 상태 설정** : 시스템이 초기 상태 에서 시작한다.

**2. 입력 처리** : 입력 가 시스템에 제공된다.

**3. 상태 전이** : 상태전이 함수 는 현재 상태 s와 입력 i에 따라 새로운 상태 s’으로 전이된다.

**4. 출력 생성** : 새로운 상태 s’에서 출력함수 를 통해출력 o가 생성된다.

**데이터의 흐름**

**1. Input -> Combination Logic** : 입력 신호가 조합 논리에 들어가고, 현재 상태와 결합하여 다음 상태를 결정한다.

**2. Combination Logic -> flip flop** : Combination Logic에서 결정된 다음 상태가 flip flop으로 전송된다.

**3. flip flop (현재 상태) -> Output Combination Logic** : flip flop에 저장된 현재 상태는 Output Combination Logic에 입력으로 제공된다.

**4. Output Combination Logic -> Output** : Output Combination Logic는 현재 상태에 기반하여 출력을 생성한다.

**무어 머신의 장점**

**1. 예측 가능성** : 출력이 상태에 의해서만 결정되기 때문에 시스템의 출력이 상태 변화에 따라 일정하게 유지된다.

**2. 디버깅 용이성** : 상태에 따른 출력이 명확히 구분되므로 시스템의 상태를 쉽게 추적하고 디버깅할 수 있다.

**3. 직관적인 설계** : 상태 다이어그램을 통해 시스템의 상태와 출력을 쉽게 설계하고 이해할 수 있다.

**무어 머신의 응용**

**제어 시스템** : 산업 자동화에서 특정 조건에 따라 기계를 제어하는 시스템.

**통신 시스템** : 데이터 전송 중 특정 패턴을 감지하고 처리하는 시스템 .

**디지털 회로 설계** : 복잡한 논리 회로에서 상태 기반 제어를 구현하는 시스템.

**JK Flip Flop을 사용한 Moore Machine circuit**

도표, 평면도, 라인, 기술 도면이(가) 표시된 사진

자동 생성된 설명

**Sequence Detector**

시퀀스 디텍터는 디지털 시스템에서 특정 비트 패턴을 입력 데이터 스트림에서 인식하고, 해당 패턴이 발견되면 신호를 출력하는 순차 상태 머신이다.

시퀀스 디텍터는 입력 비트 스트림(예: 시리얼 데이터)에서 미리 정의된 패턴을 찾는 기능을 수행한다. 입력 데이터에서 특정 패턴(예: 0001 또는 1100 등)을 감지하여 패턴이 검출되면 출력 신호(1)를 생성한다.

시퀀스 디텍터는 밀리머신과 무어머신을 활용하여 설계할 수 있다.

무어 머신을 사용한 시퀀스 디텍터는 출력이 상태에 의해 결정되므로 설계가 직관적이고 안정적이다. 출력이 일정하게 유지되는 특성 때문에 복잡한 출력 요구가 적은 시스템에 적합하다.

밀리 머신을 사용한 시퀀스 디텍터는 출력이 상태와 입력의 조합에 의해 결정되므로 반응성이 빠르다. 입력에 대한 즉각적인 출력 반응이 필요한 시스템에 적합하다.

**구성요소**

**입력 신호:**

정의: 디텍터가 분석할 데이터 스트림이다. 이 스트림은 0과 1의 연속적인 비트로 구성된다.

역할: 시퀀스 디텍터의 조합 논리에서 이 입력을 분석하여 패턴을 찾는다.

**상태 레지스터:**

정의: 현재 상태를 저장하는 플립플롭이나 레지스터.

역할: 현재 입력과 과거의 입력 비트들이 반영된 상태를 저장하여, 다음 상태를 결정하는 데 사용된다.

**조합 논리 (Combinational Logic):**

정의: 현재 상태와 입력 비트를 조합하여 다음 상태를 결정하는 논리 회로.

역할: 입력 신호와 현재 상태를 기반으로 다음 상태와 출력 신호를 결정한다.

**출력 신호:**

정의: 시퀀스가 검출되었음을 나타내는 신호.

역할: 시퀀스 디텍터가 미리 정의된 패턴을 감지하면 이 신호를 통해 이를 알린다.

**상태 전이 함수 (Transition Function):**

정의: 현재 상태와 입력 비트에 따라 다음 상태를 결정하는 함수.

역할: 다음 상태를 결정하여 상태 레지스터를 업데이트한다.

**클럭 신호 (Clock):**

정의: 상태 레지스터의 업데이트를 동기화하는 주기적인 신호.

역할: 입력 데이터 스트림의 각 비트를 처리할 때마다 상태를 업데이트한다.

**동작 원리**

**초기 상태 설정:** 시스템이 초기 상태에서 시작한다. 이 상태는 아무 비트도 검출되지 않은 상태이다.

**입력 비트 처리:** 입력 데이터 스트림의 비트들이 순차적으로 들어온다. 각 클럭 주기마다 입력 비트가 하나씩 처리된다.

**상태 전이:** 현재 상태와 입력 비트를 조합 논리에서 분석하여 다음 상태를 결정한다. 상태 전이 함수는 현재 상태와 입력 비트를 기반으로 다음 상태를 계산한다.

**출력 결정:** 조합 논리는 현재 상태와 입력 비트를 사용하여 패턴이 검출되었는지를 확인하고, 검출되었을 때 출력 신호를 활성화한다.

**패턴 검출:** 미리 정의된 패턴이 입력 스트림에 나타나면 시퀀스 디텍터는 출력 신호를 통해 이를 알린다.

**Overlapping과 Non Overlapping**

시퀀스 디텍터는 오버랩과 논오버랩 모드를 가질 수 있다.

**오버랩 (Overlapping)**

정의: 패턴이 검출된 후 새로운 패턴 검출이 직후에 시작된다. 즉, 패턴의 일부가 다음 패턴 검출의 일부로 중첩될 수 있다.

특징: 중첩된 패턴 검출이 가능하며, 입력 스트림에서 연속적인 패턴을 탐지할 때 유용하다.

**논오버랩 (Non-Overlapping)**

정의: 패턴이 검출된 후, 패턴의 끝 다음 비트부터 새로운 패턴 검출을 시작한다. 즉, 중첩이 없다.

특징: 하나의 패턴이 끝나야 다음 패턴 검출이 시작되므로 중복 검출이 방지된다.

\*오버랩(overlapping)과 논오버랩(non-overlapping)의 차이는 특정 패턴을 검출할 때 새로운 패턴 검출이 시작되는 시점에 있다.

오버랩은 패턴이 검출된 직후부터 새로운 패턴 검출을 시작할 수 있는 방식이고,

논오버랩은 패턴이 검출된 후 다음 패턴 검출을 시작하기 전에 이전 패턴이 완전히 끝나야 하는 방식이다.

예를들어 오버랩은 101을 검출한다고 할 때, 첫 번째 101이 검출된 후 바로 다음 101 검출을 시작한다. 논오버랩은 첫 번째 101이 검출된 후 해당 패턴의 끝 다음 비트부터 새로운 101 검출을 시작한다. 11010101의 입력이 있으면 오버랩은 2-4, 4-6에서 검출, 논오버랩은 2-4,5-7에서 검출한다.

**무어머신을 사용한 예시**

무어머신(Overlapping)을 사용하여 0001을 검출하는 State Diagram

스케치, 도표, 원, 그림이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 번호, 폰트이(가) 표시된 사진

자동 생성된 설명

**밀리 머신을 사용한 예시**

밀리머신(Overlapping)을 사용하여 1001을 검출하는 State Diagram

도표, 폰트, 원, 스케치이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 번호, 폰트이(가) 표시된 사진

자동 생성된 설명