

컴퓨터공학 설계 및 실험 2

14주차 발표

Sequence Detector

20181294 임승섭
20191583 김태곤
20211607 한스

목차

- 1. State Machine
- 2. Mealy Machine
- 3. Moore Machine
- 4. Sequence Detector

1. State Machine

개념

- 컴퓨터 프로그램 및 순차 논리 회로 설계에 사용되는 계산 모델
- 상태 목록, 초기 상태, 각 전이를 유발하는 **입력**으로 정의
- 일부 입력에 따라 한 상태에서 다른 상태로 변경될 수 있다

1. State Machine

종류

- Mealy (결정론적 FSM) : 출력은 상태 및 입력에 따라 달라짐
- Moore (비결정론적 FSM) : 출력은 상태에만 의존
- Harel, UML state-charts 등 복잡한 타입도 존재

1. State Machine

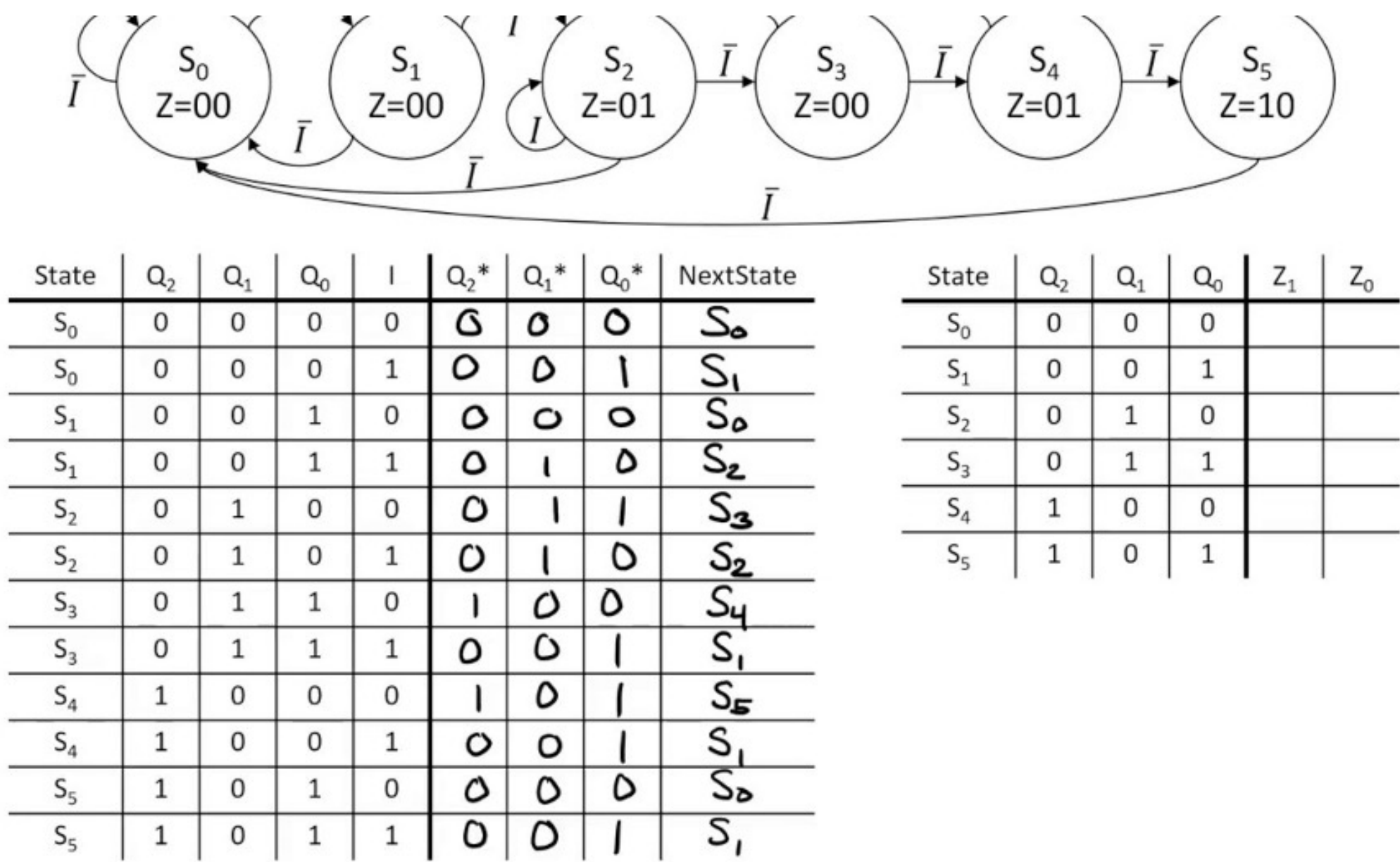
용어

- State (상태) : 전이를 실행하기 위해 대기 중인 시스템의 상태 설명
- Transition (전이) : 조건이 충족되거나 이벤트가 수신될 때 실행될 일련의 동작
 - an entry action (진입 동작) : 상태에 진입할 때 수행
 - an exit action (종료 동작) : 상태를 벗어날 때 수행

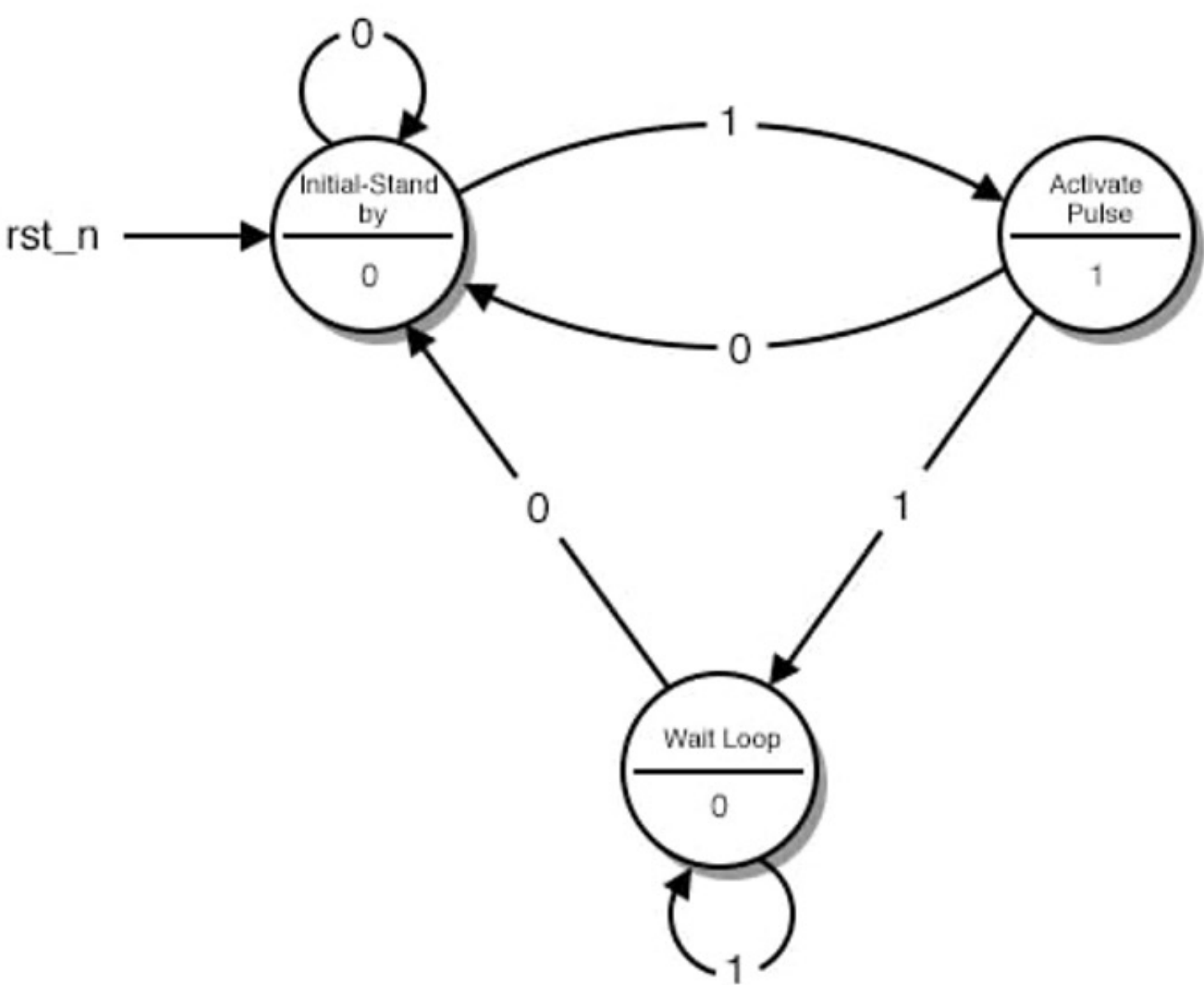
1. State Machine

표현 방법

- State/event table



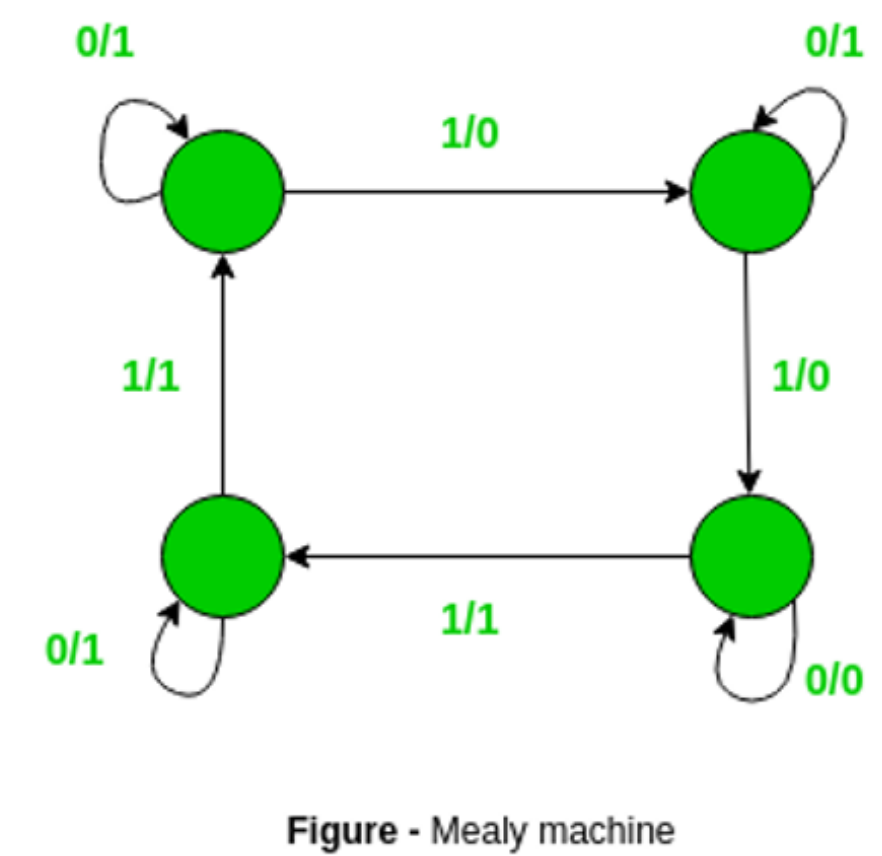
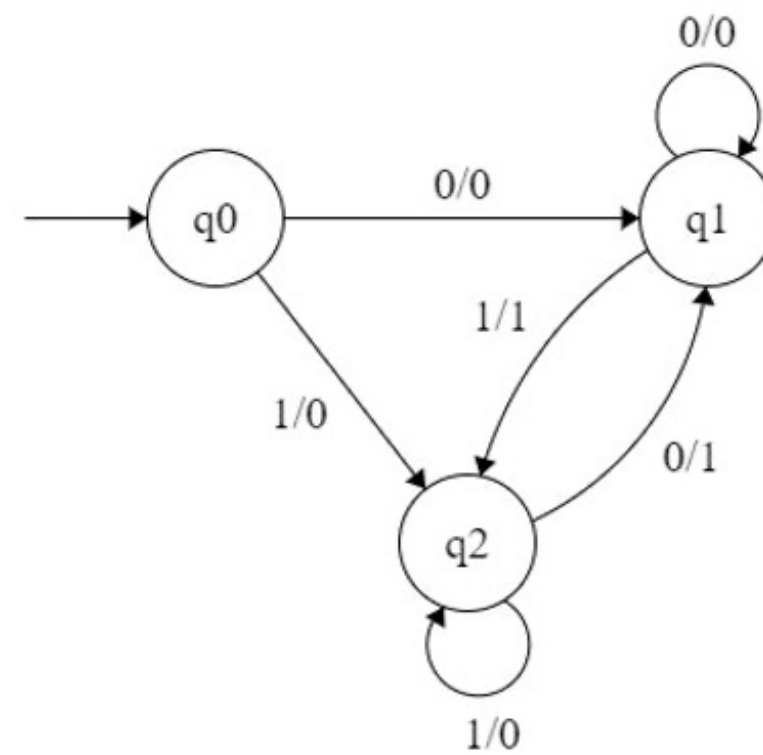
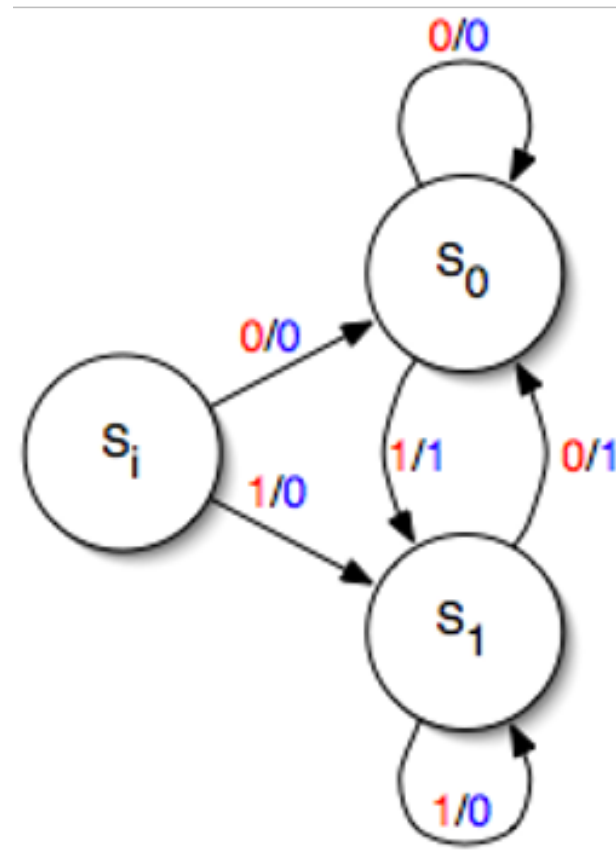
- State Diagram



2. Mealy Machine

개념

- 현재 상태와 입력의 조합에 의해 출력이 결정되는 방식의 기계 (결정론적 FSM)
- State diagram에서 transition edge에 출력 값 명시
- (Simple type) 각 transition edge에 input 값 & output 값 표시



2. Mealy Machine

구현 (101 Mealy Sequence Detector)

- 1. State Diagram
- 2. Code Assignment
- 3. Make Present State / Next State table
- 4. Draw K-maps for Dx, Dy and output (Z)
- 5. Implement the circuit

2. Mealy Machine

사용

- 1. Number Classification (숫자 분류 시스템)
 - 상태 : a (짝수), b (홀수)
 - 입력 : 0 (짝수), 1 (홀수)
 - 출력 : 현재 상태에 따라 결정
- 2. Watch the Timer (타이머 기능)
 - 상태 : 타이머 비활성 상태, 설정 상태, 작동 상태
 - 입력 : 시간 설정 버튼, 타이머 시작 버튼
 - 출력 : 현재 시간, 타이머 알람

2. Mealy Machine

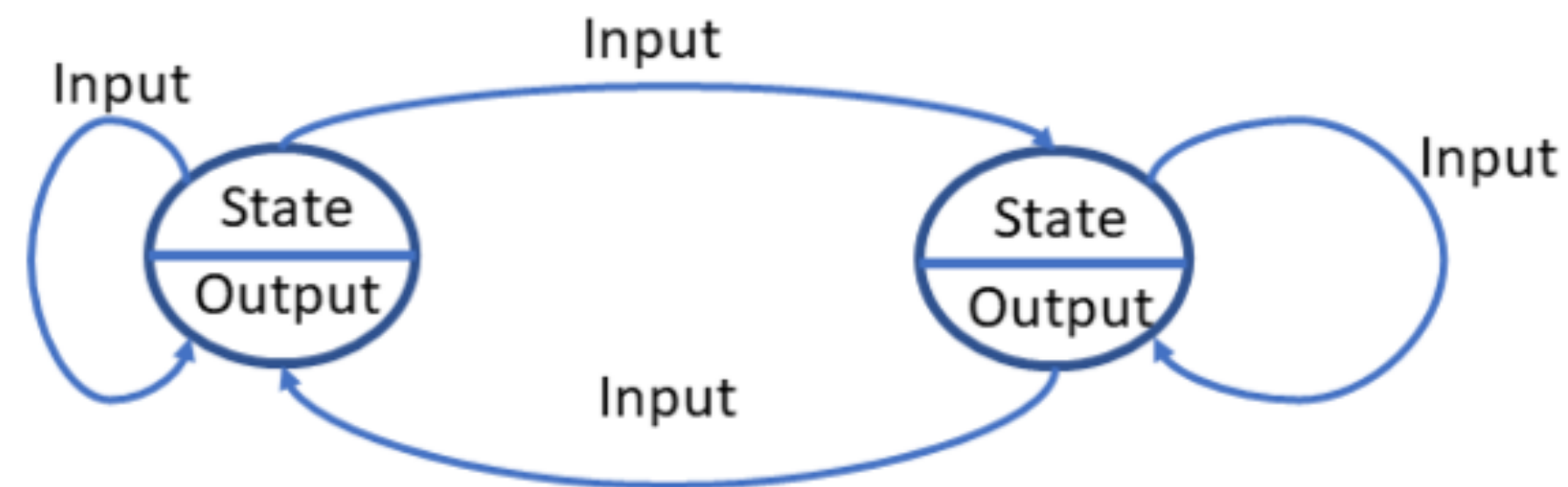
사용

- 3. Vending Machine (자판기)
 - 상태 : 대기 상태, 돈 투입 상태, 상품 선택 상태, 상품 제공 상태
 - 입력 : 돈 투입, 상품 선택, 상품 제공
 - 출력 : 잔돈 반환, 상품 제공
- 4. Traffic Light
 - 상태 : 빨간불, 초록불, 노란불
 - 입력 : 타이머
 - 출력 : 현재 신호 상태

3. Moore Machine

개념

- 유한 상태 기계의 한 유형
- 출력이 현재 상태에 의해서만 결정
- 출력이 입력에 직접적으로 의존하지 않고, 오직 현재 시스템의 상태에만 연결
- 상태가 변경될 때만 출력 변화



3. Moore Machine

구성 요소

- **1. 입력 (Input)**
 - 시스템에 제공되는 외부 신호 또는 데이터
 - 상태 전이에 영향을 주지만, 출력에 직접적인 영향을 미치지 않는다.
- **2. 조합 논리 (Combinational Logic)**
 - 현재 상태와 입력을 받아 다음 상태를 결정하는 논리 회로
- **3. 플립플롭 (Flip-Flop)**
 - 조합 논리에서 제공된 다음 상태 정보 저장
 - 클럭 신호에 의해 상태가 갱신

3. Moore Machine

구성 요소

- **4. 출력 조합 논리 (Output Combinational Logic)**
 - 현재 상태를 기반으로 출력 생성
 - 플립플롭에 저장된 현재 상태 정보를 입력으로 받아, 해당 상태에 대한 출력 결정
- **5. 클럭 신호 (CLK)**
 - 플립플롭의 상태 전이를 동기화하는 주기적인 신호
- **6. 출력**
 - 시스템의 외부로 나가는 신호 또는 데이터
 - 현재 상태에 의해서만 결정

3. Moore Machine

데이터 흐름

- **1. Input -> Combinational Logic**
 - 입력 신호가 조합 논리에 들어가고, 현재 상태와 결합하여 다음 상태 결정
- **2. Combinational Logic -> Flip Flop**
 - 조합 논리에서 결정된 다음 상태가 플립플롭으로 전송
- **3. Flip Flop -> Output Combinational Logic**
 - 플립플롭에 저장된 현재 상태가 출력 조합 논리의 입력으로 제공
- **4. Output Combinational Logic -> Output**
 - 출력 조합 논리는 현재 상태에 기반, 출력 생성

4. Sequence Detector

개념

- 디지털 시스템에서 특정 비트 패턴을 입력 데이터 스트림에서 인식하고, 해당 패턴이 발견되면 신호를 출력하는 순차 상태 머신
- 입력 비트 스트림에 미리 정의된 패턴을 찾는 기능 수행.
패턴이 검출되면 출력 신호 생성
- Mealy Machine 사용 : 빠른 반응성. 즉각적인 output이 필요한 시스템에 적합
(출력이 상태와 입력의 조합에 의해 결정)
- Moore Machine 사용 : 직관적이고 안정적인 설계. 복잡한 출력 요구가 적은 시스템에 적합
(출력이 상태에 의해 결정. 출력이 일정하게 유지되는 특성)

4. Sequence Detector

구성 요소

- 1. 입력 신호

- 정의 : Detector가 분석할 데이터 스트림
- 역할 : 이 입력을 분석하여 패턴을 찾는다.

- 2. 상태 레지스터

- 정의 : 현재 상태를 저장하는 플립플롭이나 레지스터
- 역할 : 현재 입력과 과거 입력 비트가 반영된 상태 저장. 다음 상태 결정하는 데 사용

- 3. 조합 논리

- 정의 : 현재 상태와 입력 비트 조합하여 다음 상태 결정하는 논리 회로
- 역할 : 입력 신호와 현재 상태 기반으로 다음 상태와 출력 신호 결정

4. Sequence Detector

구성 요소

- 4. 출력 신호
 - 정의 : 시퀀스가 검출되었음을 나타내는 신호
 - 역할 : Sequence Detector가 패턴 감지하면 이 신호를 통해 알린다.
- 5. 상태 전이 함수
 - 정의 : 현재 상태와 입력 비트에 따라 다음 상태를 결정하는 함수
 - 역할 : 다음 상태 를 결정하여 상태 레지스터 업데이트
- 6. 클럭 신호
 - 정의 : 상태 레지스터의 업데이트를 동기화하는 주기적인 신호
 - 역할 : 입력 데이터 스트림의 각 비트를 처리할 때마다 상태 업데이트

4. Sequence Detector

동작 원리

- **1. 초기 상태 설정** : 아무 비트도 검출되지 않은 상태
- **2. 입력 비트 처리** : 입력 데이터 스트림의 비트들이 순차적으로 들어온다
- **3. 상태 전이** : 현재 상태와 입력 비트를 조합 논리에서 분석하여 다음 상태 결정
- **4. 출력 결정** : 조합 논리가 패턴이 검출되었는지 확인하고, 검출되었으면 출력 신호 활성화
- **5. 패턴 검출** : 미리 정의된 패턴이 입력 시스템에 나타나면 시퀀스 디텍터는 출력 신호를 통해 이를 알린다.

4. Sequence Detector

Overlapping과 Non Overlapping

- **1. Overlapping :**

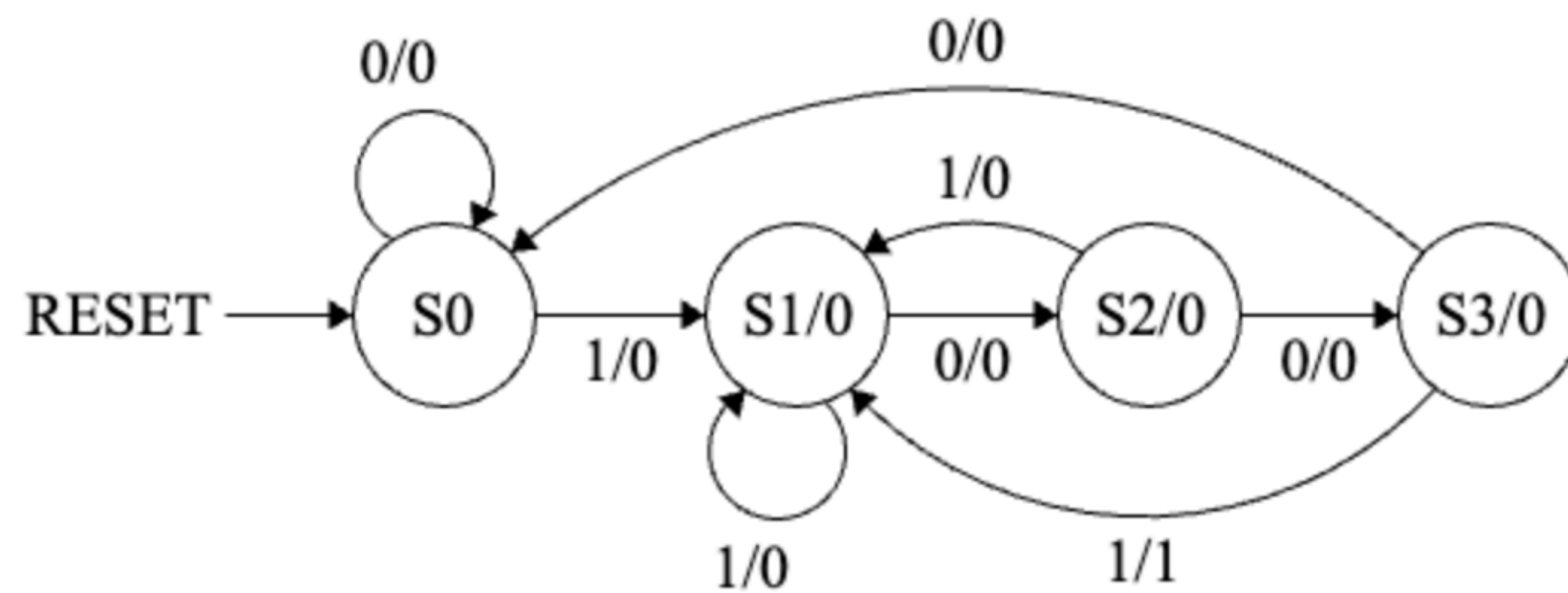
- 패턴이 검출된 후 새로운 패턴 검출이 직후에 시작된다. 즉, 패턴의 일부가 다음 패턴 검출의 일부로 중첩될 수 있다.
- 중첩된 패턴 검출이 가능하며, 입력 스트림에서 연속적인 패턴을 탐지할 때 유용하다

- **2. Non Overlapping:**

- 패턴이 검출된 후, 패턴의 끝 다음 비트부터 새로운 패턴 검출을 시작한다. 즉, 중첩이 없다.
- 하나의 패턴이 끝나야 다음 패턴 검출이 시작되므로 중복 검출이 방지된다.

4. Sequence Detector

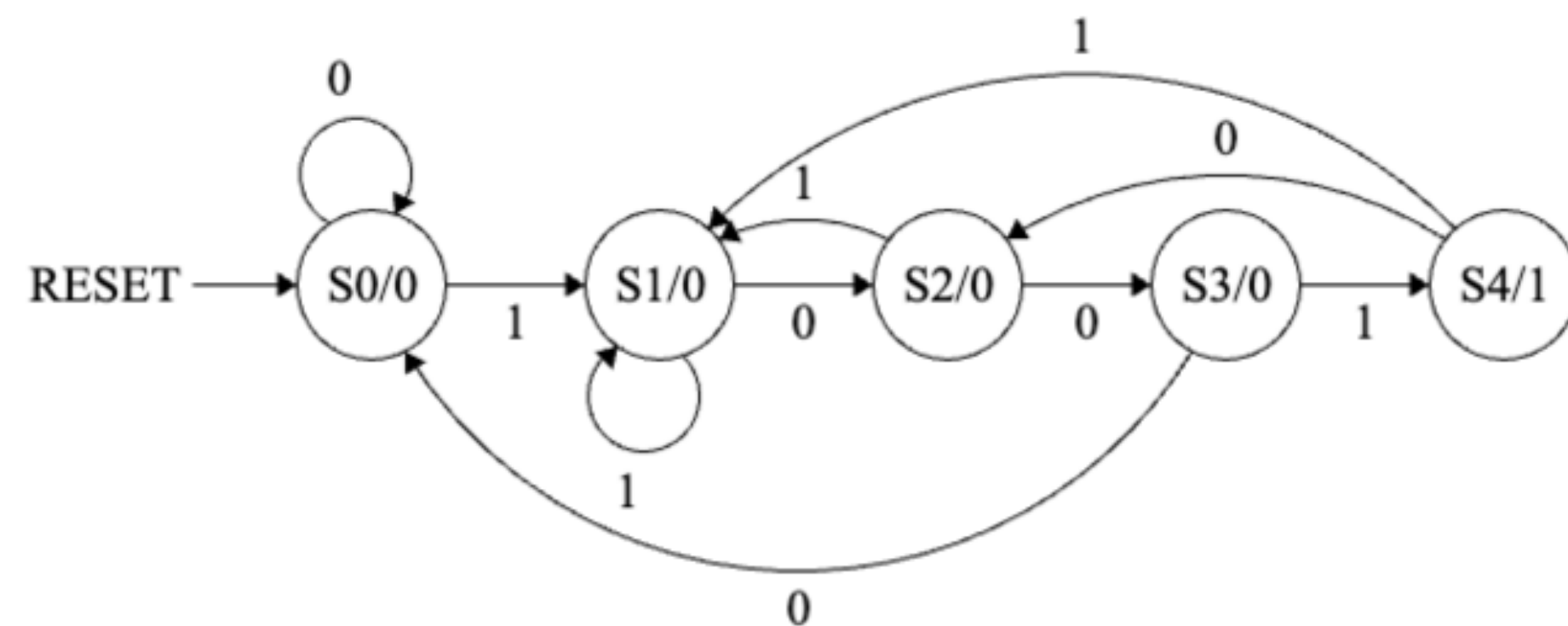
예시 - Mealy Machine



Present State	Input A= 0		Input A= 1	
	Next State	Output	Next State	Output
S0	S0	0	S1	0
S1	S2	0	S1	0
S2	S3	0	S1	0
S3	S0	0	S1	1

4. Sequence Detector

예시 - Moore Machine



Present State	Next State		Output
	Input A= 0	Input A= 1	
S0	S0	S1	0
S1	S2	S1	0
S2	S3	S1	0
S3	S0	S4	0
S4	S2	S1	1

감사합니다

임승섭 - 33.3%

김태곤 - 33.3%

한스 - 33.3%