

# Digital System Assignment 2

- FSM (Sequence Detector) -

# Sequence Detector

---

## ■ 목표

- FSM(finite state machine)을 이용해 Sequence Detector를 설계한다.
- 최소 개의 state를 사용한다.
- Mealy/Moore Machine으로 각각 설계한다.

## ■ 입력

- CLK
- RESET
- X

## ■ 출력

- 1-bit Z

# Sequence Detector

## ■ Sequence Detector

- "1010"의 연속된 입력이 들어오면 '1'을 출력하고, 그렇지 않으면 '0'을 출력한다.
- RESET이 들어오면 아무 입력도 받지 않은 초기 상태로 돌아간다.
- 'Z=1'이 출력되어도, 현재까지 들어온 입력은 모두 유효하다.

## ■ 기본 사항

- RESET은 CLK에 관계없이 (asynchronous) 동작한다.
- 상태 변화는 CLK의 Rising Edge에 맞춰(synchronous) 동작한다.
- Reset : 0이 입력되면 활성화(Active low)

X : 0 0 1 1 0 0 0 1 0 1 0 1 0 0 0

Z : 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0

# State graph

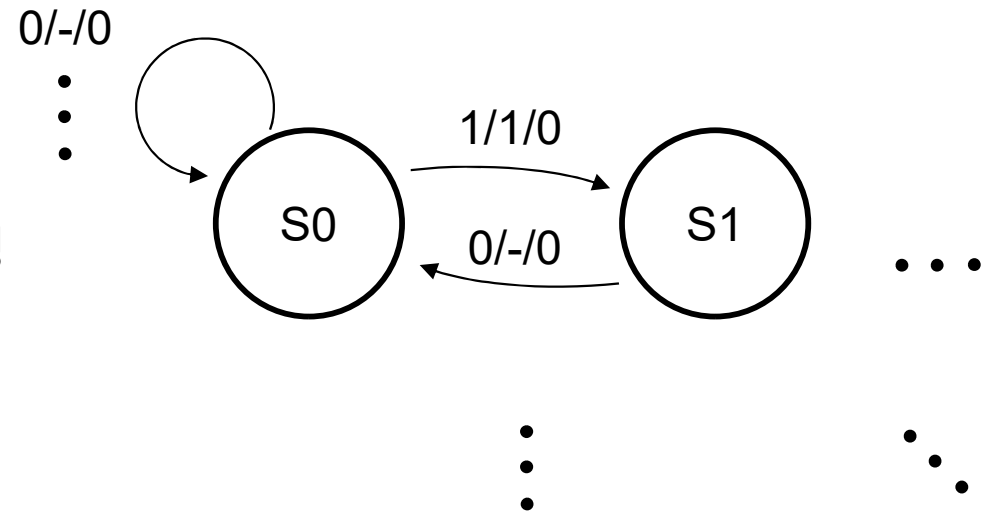
## ■ 상태기계

- 아래 예시를 참고해 Sequence Detector의 밀리/무어 상태기계를 완성하고 상태도(state graph)를 각각 보고서에 추가하시오.

- ❖ 상태의 개수가 최소가 되도록 설계
- ❖ 각 상태의 이름을 정하고, 각 상태가 어떤 상태인지 반드시 설명

- Mealy 기계 상태도 예시

- ❖ 표시는 RESET/X/Z 순서
- ❖ RESET과 X는 입력, 출력은 Z
- ❖ Moore도 Moore의 특성에 맞게 작성



## Assignment 2

---

### ■ Entity

- 아래의 Entity를 그대로 사용하시오

```
entity Seq_Detector_Mealy is
  port (
    CLK : in std_logic;
    RESET : in std_logic;
    X : in std_logic;
    Z : out std_logic
  );
end Seq_Detector_Mealy;
```

```
entity Seq_Detector_Moore is
  port (
    CLK : in std_logic;
    RESET : in std_logic;
    X : in std_logic;
    Z : out std_logic
  );
end Seq_Detector_Moore;
```

## Assignment 2

### ■ Mealy Machine Source Code 예시

```
architecture Behavioral of Seq_Detector_Mealy is
```

```
  type states is (s0
```

```
    signal state, next_state : states;
```

```
begin
```

```
  -- state transition
```

```
  process
```

```
    begin
```

```
      if (RESET = '0') then
```

```
        elsif (CLK = '1' and CLK'event) then
```

```
          end if;
```

```
        end process;
```

```
  -- Next_state, Outputs
```

```
  process
```

```
    begin
```

```
      case state is
```

```
        when s0 =>
```

## Assignment 2

### ■ Moore Machine Source Code 예시

```

architecture Behavioral of Seq_Detector_Moore is
  type states is (s0, );
  signal state, next_state : states;
begin
  -- state transition
  process 
  begin
    if (RESET = '0') then
      
    elsif (CLK = '1' and CLK'event) then
      
    end if;
  end process;

```

-- outputs

process

begin

case state is

end process;

-- next state

process

begin

case state is

# TestBench Code - Mealy

## ■ Testbench (180ns까지 결과 첨부)

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

) entity Seq_Detector_Mealy_TB is
) end Seq_Detector_Mealy_TB;

) architecture Behavioral of Seq_Detector_Mealy_TB is
)   component Seq_Detector_Mealy is
)     port (
)       CLK : in std_logic;
)       RESET : in std_logic;
)       X : in std_logic;
)       Z : out std_logic);
)   end component;

)   signal CLK : std_logic;
)   signal RESET : std_logic;
)   signal X, Z : std_logic;
)   begin
)     SD : Seq_Detector_Mealy port map (CLK=>CLK, RESET=>RESET, X=>X, Z=>Z);

)   rst_operation : process
)     begin
)       RESET <= '0';
)       wait for 15ns;
)       RESET <= '1';
)       wait for 93ns;
)       RESET <= '0';
)       wait for 10ns;
)       RESET <= '1';
)       wait;
)     end process;

```

```

clk_operation : process
begin
  CLK <= '1';
  wait for 5ns;
  CLK <= '0';
  wait for 5ns;
end process;

```

```

input_operation : process
begin
  X <= '1';
  wait for 25ns;
  X <= '0';
  wait for 20ns;
  X <= '1';
  wait for 10ns;
  X <= '0';
  wait for 20ns;
  X <= '1';
  wait for 10ns;
  X <= '0';
  wait for 10ns;
  X <= '1';
  wait for 10ns;
  X <= '0';
  wait for 10ns;
  X <= '1';
  wait for 10ns;
  X <= '0';
  wait for 10ns;
  X <= '1';
  wait for 10ns;
  X <= '0';
  wait for 10ns;
  X <= '1';
  wait for 20ns;
  X <= '0';
  wait;
end process;

```

end Behavioral;



# TestBench Code - Moore

## ■ Testbench (150ns까지 결과 첨부)

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity Seq_Detector_Moore_TB is
end Seq_Detector_Moore_TB;

architecture Behavioral of Seq_Detector_Moore_TB is
    component Seq_Detector_Moore is
        port (
            CLK : in std_logic;
            RESET : in std_logic;
            X : in std_logic;
            Z : out std_logic);
        end component;

    signal CLK : std_logic;
    signal RESET : std_logic;
    signal X, Z : std_logic;
begin
    SD : Seq_Detector_Moore port map (CLK=>CLK, RESET=>RESET, X=>X, Z=>Z);

    rst_operation : process
    begin
        RESET <= '0';
        wait for 15ns;
        RESET <= '1';
        wait for 80ns;
        RESET <= '0';
        wait for 10ns;
        RESET <= '1';
        wait;
    end process;

```

```

clk_operation : process
begin
    CLK <= '1';
    wait for 5ns;
    CLK <= '0';
    wait for 5ns;
end process;

```

```

input_operation : process
begin
    X <= '1';
    wait for 25ns;
    X <= '0';
    wait for 10ns;
    X <= '1';
    wait for 10ns;
    X <= '0';
    wait for 10ns;
    X <= '1';
    wait for 10ns;
    X <= '0';
    wait for 10ns;
    X <= '1';
    wait for 10ns;
    X <= '0';
    wait for 10ns;
    X <= '1';
    wait for 10ns;
    X <= '0';
    wait for 10ns;
    X <= '1';
    wait for 10ns;
    X <= '0';
    wait;
end process;
end Behavioral;

```

# Submission

---

## ■ 제출

- 제출 기한 : 5/14 (화) 11:00 PM
- i-campus에 파일 업로드 (보고서1개 + 소스코드 4개(테스트벤치 포함)를 하나로 압축해 제출)
- 압축된 파일명 : **학번\_이름\_HW2.xxx**
- Late Penalty
  - ❖ 1s ~ 1 day : -10%
  - ❖ 1day ~ 2day : -30%
  - ❖ 2day ~ 3day : -50%
  - ❖ 3day ~ : Cannot submit

■ 과제관련 문의 : [myun243@skku.edu](mailto:myun243@skku.edu)

# Submission

## ■ 제출 유의사항 (미흡시 불이익)

### ● 설계한 VHDL 코드와 waveform을 첨부한 보고서

#### ❖ 설계한 VHDL 코드와 설명

- 설계의 핵심이 되는 내용은 반드시 보고서에 설명
- 설계한 방법과 이를 바탕으로 동작 원리 상세히 설명
- 주어진 TestBench 결과에 짜맞추어 진행한 경우, 대량 감점

#### ❖ 주어진 TB를 이용한 시뮬레이션 결과 waveform과 설명 및 분석

- Mealy / Moore 결과에 차이가 있었다면 (혹은 없었다면), 어떠한 원인에 의해 결과에 변화가 (혹은 동일하게) 나올 수 있었는 지 논리적으로 분석할 것

#### ❖ 자체적으로 추가 TB를 작성하여 시뮬레이션 진행한 후, 결과 분석

- 소스 코드 별도 제출 x
- 보고서에 별도 작성 x
- 채점은 위의 예시와 다른 테스트벤치를 이용해 진행함

# Submission

---

## ■ 제출 유의사항 (미흡시 불이익)

- 직접 작성한 VHDL 소스코드 파일 제출
  - ❖ Source code 파일명 : 학번\_이름\_Moore.vhd / 학번\_이름\_Mealy.vhd
  - ❖ Testbench code 파일명 : 학번\_이름\_Moore\_tb.vhd / 학번\_이름\_Mealy\_tb.vhd
  - ❖ 보고서 파일명 : 학번\_이름.xxx
  - ❖ 소스코드의 핵심 내용은 반드시 주석으로 설명할 것
- 보고서와 소스코드 파일을 하나로 압축해 제출