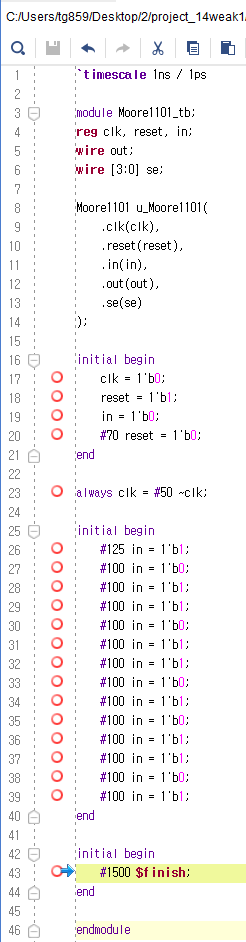
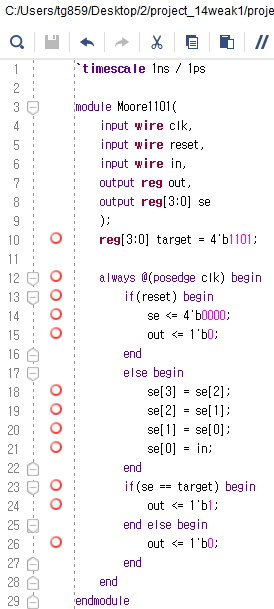
14주차 결과보고서

전공: 컴퓨터공학과 학년: 4학년 학번: 20191583 이름: 김태곤

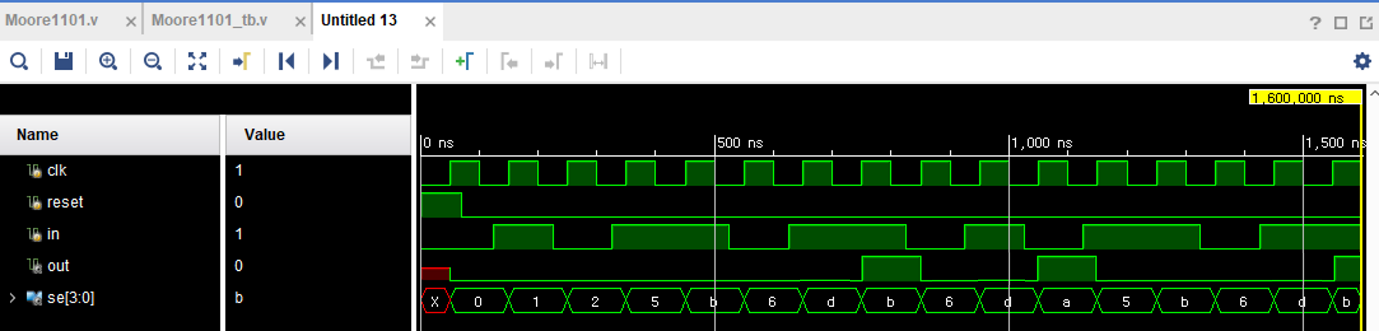
**1. Sequence Detector 1101 Moore machine 구현**

**(verilog source, simulation 결과, 상태도(State Table) 및 상태표(State Diagram)작성)**

1. **Verilog Source**

****

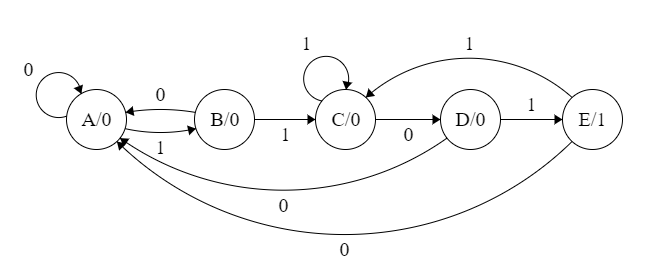
1. **Simulation**



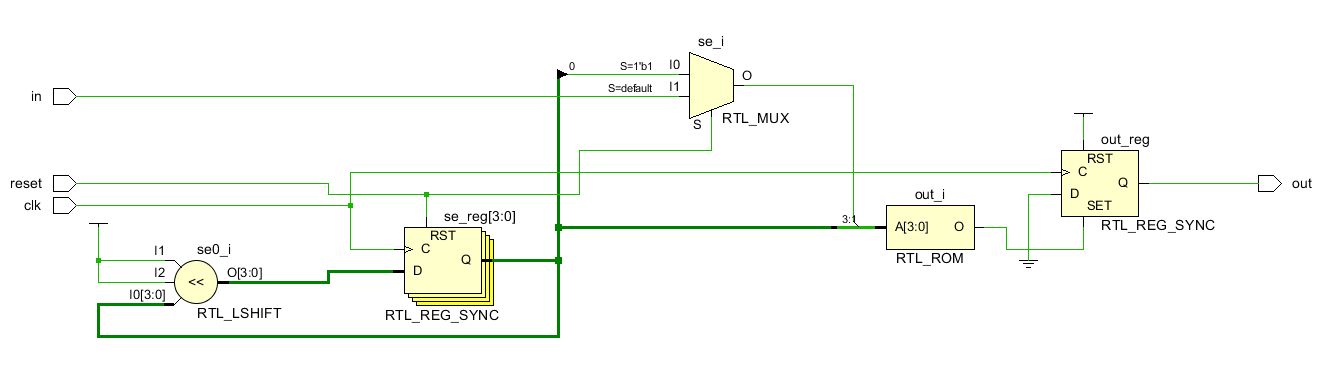
1. **State Table**

|  |  |  |  |
| --- | --- | --- | --- |
| Present State | Next State | | Output |
| X=0 | X=1 |
| A | A | B | 0 |
| B | A | C | 0 |
| C | D | C | 0 |
| D | A | E | 0 |
| E | A | C | 1 |

1. **State Diagram**

****

1. **Schematic**

****

**6) 결과 및 과정**

**이번 실험에서는 Moore Machine을 이용해 겹침(overlapping)을 허용하는 1101 시퀀스를 검출하는 회로를 설계하고, 이에 대한 검증을 수행하였다. 실험 과정은 다음과 같이 진행되었다.**

**Moore1101 모듈은 clk, reset, in을 입력으로 받고, out과 se를 출력한다. clk 신호의 상승 모서리에서 reset 신호가 입력된 경우, 상태 se를 0000으로 초기화하고, out 또한 0으로 초기화한다. 이후 상태는 다음과 같이 전이된다.**

**se의 각 비트는 오른쪽으로 시프트하며, in의 값은 se[0]에 할당된다.**

**현재 상태 se가 목표 상태인 1101과 일치하는 경우 out이 1로 설정된다. 그렇지 않으면 0으로 설정된다.**

**테스트 벤치에서는 회로를 초기화하기 위해 reset 신호를 70ns 동안 1로 설정하였다. clk 신호는 50ns마다 변화하도록 설정하여, 상승 모서리에서 상태 전이가 발생하도록 하였다. 초기 입력 신호 in의 값은 0으로 설정하였다.**

**회로 초기화가 완료된 후 in 신호를 순차적으로 변화시켜 1101 시퀀스를 형성하였다. in 신호는 다음과 같은 순서로 변경되었다: 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1. 각 변화는 clk 신호의 상승 모서리에서 발생하며, Moore Machine의 상태 전이와 out 출력 값을 확인하였다.**

**실험 결과, 특정 시점마다 out 값이 1101 시퀀스의 끝에 도달할 때마다 1로 출력되는 것을 확인할 수 있었다. 예를 들어, 750ns, 1050ns, 1550ns 시점에 in 값이 1101 시퀀스를 형성하여 out이 1로 설정되었으며, 그 외의 시간에는 0이 유지되었다. 이때 Moore Machine은 State에 in의 값이 아닌 State에 따라 값이 변하므로, 실제 in의 값이 들어온 clk보다 한 clk 늦게 변하는 것을 알 수 있다. 이로써 out 값이 1101 시퀀스 탐지 시점에 성공적으로 1로 설정됨을 확인하였다.  
 특히, 겹침을 허용하여 연속된 1101 시퀀스 내에서도 성공적으로 시퀀스를 탐지하였다. 예를 들어, 750ns에서 시퀀스의 마지막 1을 인식하고 상태가 전이되었지만, 1050ns에서 새로운 1101 시퀀스를 다시 탐지하여 출력이 1로 설정되었다. 이번 실험을 통해 Moore Machine을 사용한 겹침 1101 시퀀스 탐지 회로가 정상적으로 작동함을 확인할 수 있었다.**

**2. Sequence Detector 10101 을 구현(mealy, moore machine 모두)**

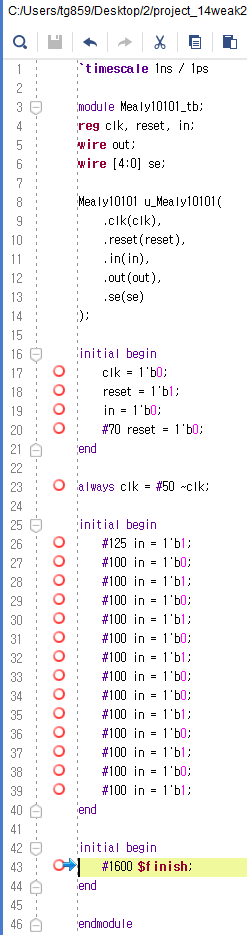
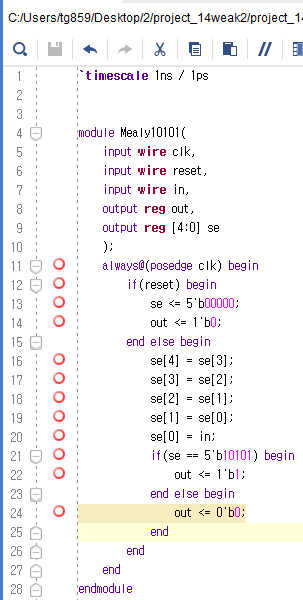
**상태도(State Table) 및 상태표(State Diagram) 작성.**

**Verliog Code 및 Simulation 결과 작성.**

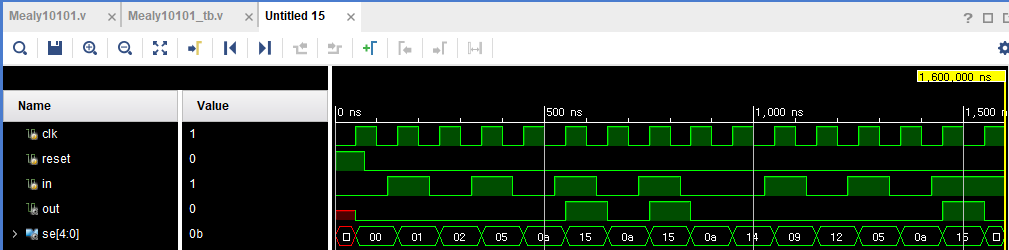
**(수업에서 배우지 않은 방식으로 코드를 작성하거나 copy하는 경우 감점됩니다.)**

1) Mealy Machine

1)) Verilog Code



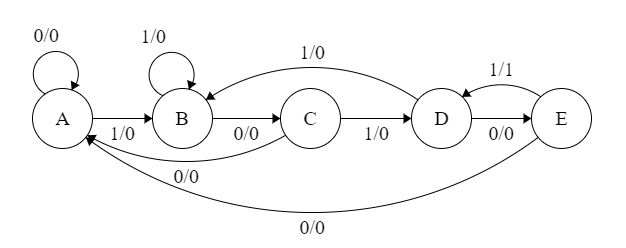
2)) Simulation



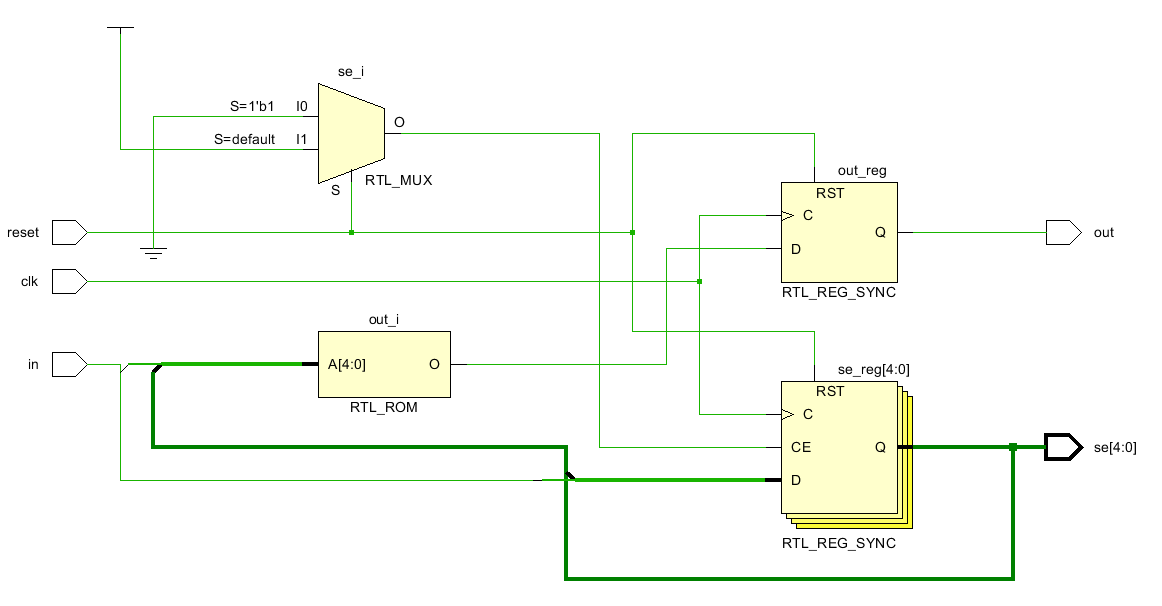
3)) State Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Present State | Next State | | Output | |
| X=0 | X=1 | X=0 | X=1 |
| A | A | B | 0 | 0 |
| B | C | B | 0 | 0 |
| C | A | D | 0 | 0 |
| D | E | B | 0 | 0 |
| E | A | D | 0 | 1 |

4)) State Diagram



5)) Schematic



6)) 결과 및 과정

**이번 실험에서는 Mealy Machine을 이용해 겹침(overlapping)을 허용하는 10101 시퀀스를 검출하는 회로를 설계하고, 이에 대한 검증을 수행하였다. 실험 과정은 다음과 같이 진행되었다.**

**Mealy10101 모듈은 clk, reset, in을 입력으로 받고, out과 se를 출력한다. clk 신호의 상승 모서리에서 reset 신호가 입력된 경우, 상태 se를 00000으로 초기화하고, out 또한 0으로 초기화한다. 이후 상태는 다음과 같이 전이된다.**

**se의 각 비트는 오른쪽으로 시프트하며, in의 값은 se[0]에 할당된다.**

**현재 상태 se가 목표 상태인 10101과 일치하는 경우 out이 1로 설정된다. 그렇지 않으면 0으로 설정된다.**

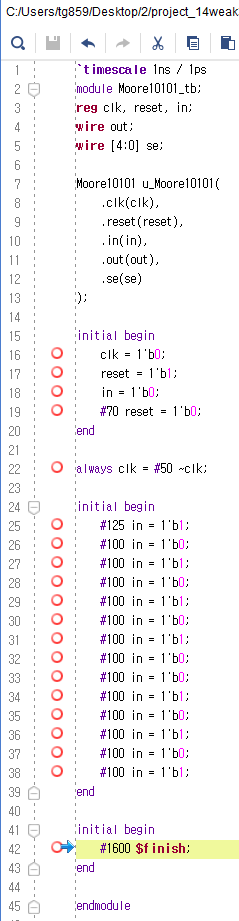
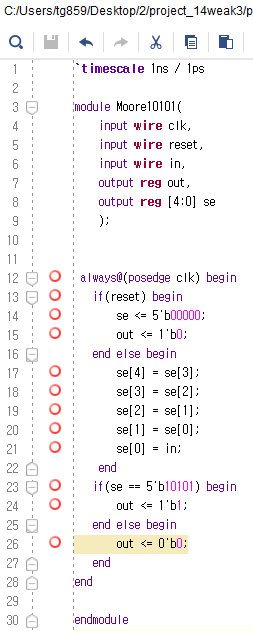
**테스트 벤치에서는 회로를 초기화하기 위해 reset 신호를 70ns 동안 1로 설정하였다. clk 신호는 50ns마다 변화하도록 설정하여, 상승 모서리에서 상태 전이가 발생하도록 하였다. 초기 입력 신호 in의 값은 0으로 설정하였다.**

**회로 초기화가 완료된 후 in 신호를 순차적으로 변화시켜 10101 시퀀스를 형성하였다. in 신호는 다음과 같은 순서로 변경되었다: 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1. 각 변화는 clk 신호의 상승 모서리에서 발생하며, Mealy Machine의 상태 전이와 out 출력 값을 확인하였다.**

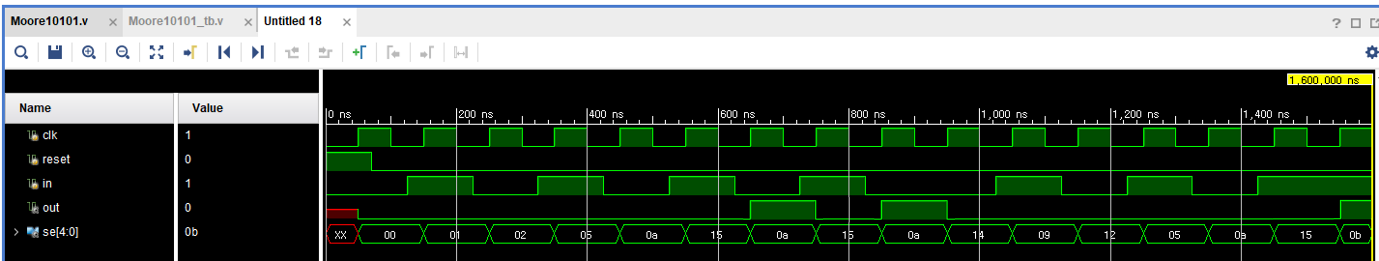
**실험 결과, 특정 시점마다 out 값이 10101 시퀀스의 끝에 도달할 때마다 1로 출력되는 것을 확인할 수 있었다. 예를 들어, 550ns, 750ns, 1450ns 시점에 in 값이 10101 시퀀스를 형성하여 out이 1로 설정되었으며, 그 외의 시간에는 0이 유지되었다. 이로써 out 값이 10101 시퀀스 탐지 시점에 성공적으로 1로 설정됨을 확인하였다.  
 특히, 겹침을 허용하여 연속된 10101 시퀀스 내에서도 성공적으로 시퀀스를 탐지하였다. 예를 들어, 550ns에서 시퀀스의 마지막 1을 인식하고 상태가 전이되었지만, 750ns에서 새로운 10101 시퀀스를 다시 탐지하여 출력이 1로 설정되었다. 이번 실험을 통해 Mealy Machine을 사용한 겹침 10101 시퀀스 탐지 회로가 정상적으로 작동함을 확인할 수 있었다.**

2) Moore Machine

1)) Verilog Code



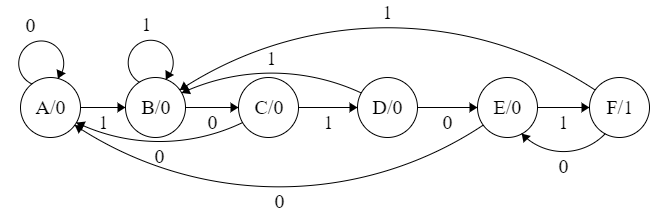
2)) Simulation



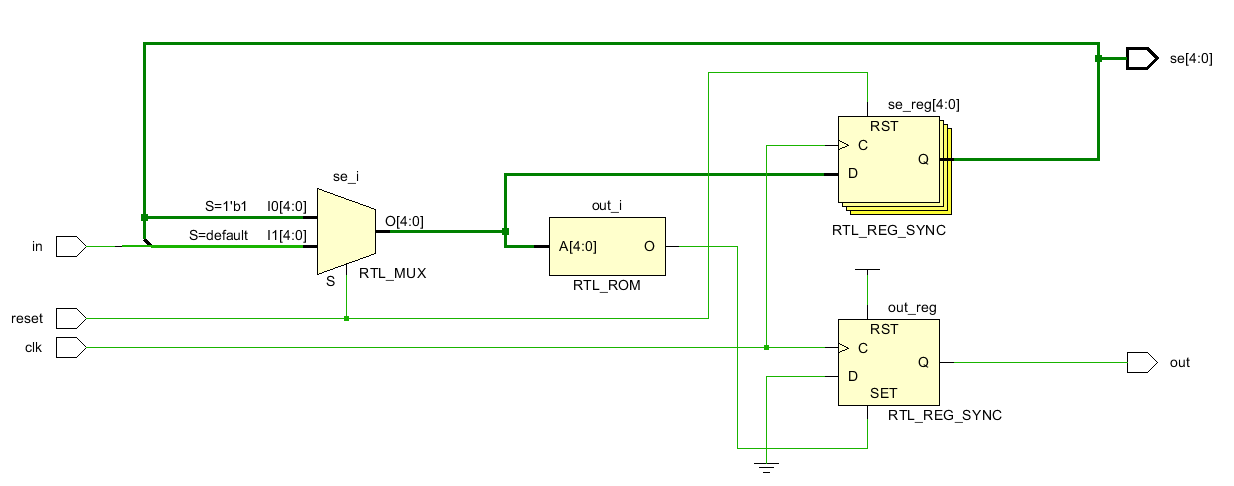
3)) State Table

|  |  |  |  |
| --- | --- | --- | --- |
| Present State | Next State | | Output |
| X=0 | X=1 |
| A | A | B | 0 |
| B | C | B | 0 |
| C | A | D | 0 |
| D | E | B | 0 |
| E | A | F | 0 |
| F | E | B | 1 |

4)) State Diagram



5)) Schematic



6)) 결과 및 과정

**이번 실험에서는 Moore Machine을 이용해 겹침(overlapping)을 허용하는 10101 시퀀스를 검출하는 회로를 설계하고, 이에 대한 검증을 수행하였다. 실험 과정은 다음과 같이 진행되었다.**

**Moore10101 모듈은 clk, reset, in을 입력으로 받고, out과 se를 출력한다. clk 신호의 상승 모서리에서 reset 신호가 입력된 경우, 상태 se를 00000으로 초기화하고, out 또한 0으로 초기화한다. 이후 상태는 다음과 같이 전이된다.**

**se의 각 비트는 오른쪽으로 시프트하며, in의 값은 se[0]에 할당된다.**

**현재 상태 se가 목표 상태인 10101과 일치하는 경우 out이 1로 설정된다. 그렇지 않으면 0으로 설정된다.**

**테스트 벤치에서는 회로를 초기화하기 위해 reset 신호를 70ns 동안 1로 설정하였다. clk 신호는 50ns마다 변화하도록 설정하여, 상승 모서리에서 상태 전이가 발생하도록 하였다. 초기 입력 신호 in의 값은 0으로 설정하였다.**

**회로 초기화가 완료된 후 in 신호를 순차적으로 변화시켜 10101 시퀀스를 형성하였다. in 신호는 다음과 같은 순서로 변경되었다: 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1. 각 변화는 clk 신호의 상승 모서리에서 발생하며, Moore Machine의 상태 전이와 out 출력 값을 확인하였다.**

**실험 결과, 특정 시점마다 out 값이 10101 시퀀스의 끝에 도달할 때마다 1로 출력되는 것을 확인할 수 있었다. 예를 들어, 650ns, 850ns, 1550ns 시점에 in 값이 10101 시퀀스를 형성하여 out이 1로 설정되었으며, 그 외의 시간에는 0이 유지되었다. 이때 Moore Machine은 State에 in의 값이 아닌 State에 따라 값이 변하므로, 실제 in의 값이 들어온 clk보다 한 clk 늦게 변하는 것을 알 수 있다. 이로써 out 값이 10101 시퀀스 탐지 시점에 성공적으로 1로 설정됨을 확인하였다.  
 특히, 겹침을 허용하여 연속된 10101 시퀀스 내에서도 성공적으로 시퀀스를 탐지하였다. 예를 들어, 650ns에서 시퀀스의 마지막 1을 인식하고 상태가 전이되었지만, 850ns에서 새로운 10101 시퀀스를 다시 탐지하여 출력이 1로 설정되었다. 이번 실험을 통해 Moore Machine을 사용한 겹침 10101 시퀀스 탐지 회로가 정상적으로 작동함을 확인할 수 있었다.**