14주차 결과보고서

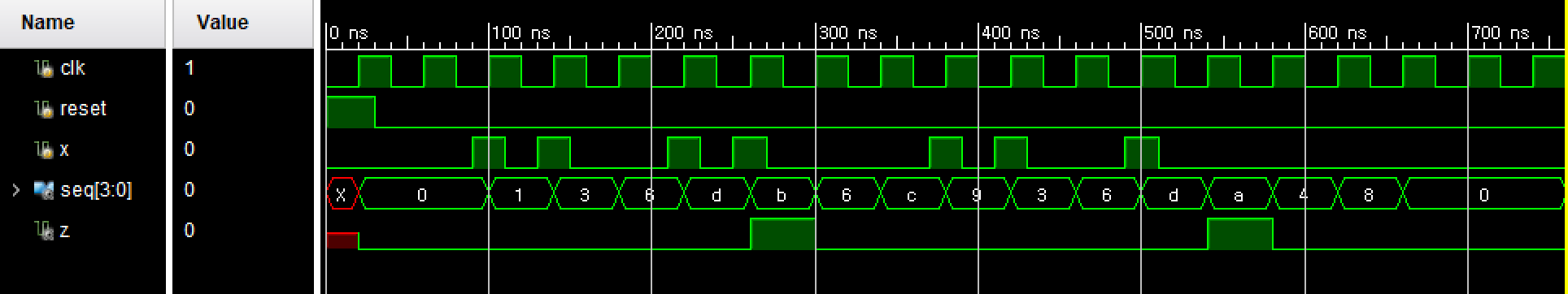
전공 : 국제한국학과 학년 : 4학년 학번 : 20181202 이름 : 김수미

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

1) Verilog Source

|  |  |
| --- | --- |
| Moore Machine | |
| Design Source | Test Bench |
| 텍스트이(가) 표시된 사진  자동 생성된 설명 | 텍스트, 영수증, 스크린샷이(가) 표시된 사진  자동 생성된 설명 |

2) Simulation



3) 상태도(State Diagram)

|  |
| --- |
| Moore Machine |
|  |

4) 상태표(State Table)

|  |
| --- |
| Moore Machine |
| |  |  |  |  | | --- | --- | --- | --- | | 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 | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | Input(X) | 0 | 1 | 1 | 0 | 1 | 1 | | State | ? | A | B | C | D | E | | Output(Z) | ? | 0 | 0 | 0 | 0 | 1 | |

5) 결과 및 과정

실습시간에는 1101 sequence를 detect하는 mealy machine을 구현했는데, 결과보고서 작성에서는 1101 sequence를 detect하는 moore machine을 구현해 보았다. Moore machine은 현재 input과 과거의 input인 state를 모두 고려하는 mealy machine과는 달리 state만을 고려하고, 현재의 input은 next state에서 다시 과거의 input에 대한 기록으로 참조된다. 따라서 1101의 sequence를 detect 하는 코드를 작성할 때 1101의 sequence가 들어오는 순간 해당 sequence를 detect 했음을 감지하는 것이 아닌, 1101의 마지막 input 1이 들어온 다음 state에서 이를 감지했음을 알려야 한다.

작성한 코드를 바탕으로 simulation을 수행한 결과 1101의 sequence가 들어올 때 마다 output 값이 1로 set 되며, 1101이 들어오자마자 set되는 것이 아닌 1101이 들어온 다음 state에서 set 되는 것을 확인할 수 있다.

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

(verilog source, simulation 결과, 상태도, 상태표 작성)

1) Verilog Source

|  |  |
| --- | --- |
| Mealy Machine | |
| Design Source | Test Bench |
| 텍스트이(가) 표시된 사진  자동 생성된 설명 | 테이블이(가) 표시된 사진  자동 생성된 설명 |
| Moore Machine | |
| Design Source | Test Bench |
| 텍스트이(가) 표시된 사진  자동 생성된 설명 | 테이블이(가) 표시된 사진  자동 생성된 설명 |

2) Simulation

|  |
| --- |
| Mealy Machine |
|  |
| Moore Machine |
|  |

3) 상태도(State Diagram)

|  |  |
| --- | --- |
| Mealy Machine | Moore Machine |
|  |  |

4) 상태표(State Table)

|  |
| --- |
| Mealy Machine |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | 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 | |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | Input(X) | 0 | 1 | 0 | 1 | 0 | 1 | 1 | | State | ? | A | B | C | D | E | D | | Output(Z) | ? | 0 | 0 | 0 | 0 | 1 | 0 | |
| Moore Machine |
| |  |  |  |  | | --- | --- | --- | --- | | 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 | |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | Input(X) | 0 | 1 | 0 | 1 | 0 | 1 | 0 | | State | ? | A | B | C | D | E | F | | Output(Z) | ? | 0 | 0 | 0 | 0 | 0 | 1 | |

5) 결과 및 과정

이번에는 10101의 5bit sequence를 detect하는 mealy machine과 moore machine을 Verilog 코드로 작성해 보았다. 이전에 작성한 4bit sequence detector에서 1bit만 더 detect 할 수 있게 수정해 주면 되기 때문에 작성하는 것은 크게 어렵지 않다. Shift register을 구현했던 것처럼 오른쪽으로 한 칸씩 bit의 값을 이동시킴과 동시에 MSB에서 input 데이터 값을 받아 저장하고, 현재 5bit register에 저장된 값이 detect 하고자 하는 sequence와 동일한지를 검사해주면 된다. 단 mealy machine은 10101의 sequence에서 마지막 input 값 1이 들어오자마자 sequence를 detect 했음을 output 값을 통해 표시해야 하며, moore machine은 마지막 input 값 1이 들어와서 다음 state에 메모리에 그 값이 저장되면 그 때서야 sequence가 detect 되었음을 ouput을 통해 표시해야 한다.

작성한 코드를 바탕으로 simulation을 수행한 결과 mealy machine은 10101의 sequence가 들어오자마자 output 값이 1로 set 되고, moore machine은 10101 sequence 가 들어오고 나서 바로 다음 stage에 output 값이 1로 set되는 것을 확인할 수 있다.