컴퓨터공학실험2 14주차 결과 보고서

전공: 컴퓨터공학 학년: 2학년 학번: 20201635 이름: 전찬

**0. 목차**

1. Sequence Detector 1101 Moore machine 구현

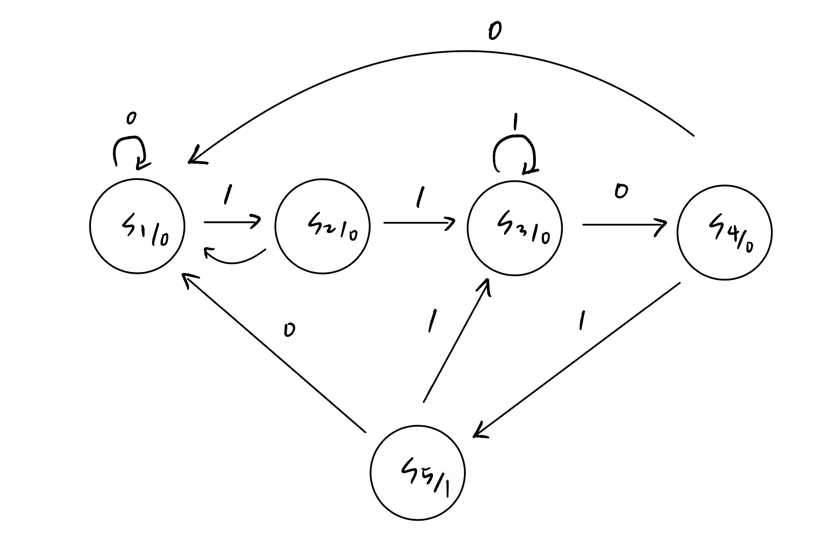
2. Sequence Detector 10101 machine 구현

2.1. Mealy machine 구현

2.2 Moore machine 구현

**1. Sequence Detector 1101 Moore machine**

Moore machine과 Mealy machine의 가장 큰 차이점은 graph 형태(state diagram)로 구현했을 때, edge에 output이 존재하는지, vertex에 output이 존재하는지 이다. 따라서 1101 sequence detector에서 Mealy machine은 총 4가지 상태를 기본으로 구현하였는데, 이와 다르게 Moore machine은 5개의 상태로 구현해낼 수 있다. 이에 대한 state diagram은 아래와 같다.(S1 = 0, S2 = 1, S3 = 11, S4 = 110, S5 = 1101의 상태를 나타낸다.)



<sequence detector 1101 Moore machine의 state diagram>

또한 이를 코드로 구현한 결과는 아래와 같다. 이때 code에서 각 out은 diagram에서 각 S를 의미하며, detect = output을 의미한다.

`timescale 1ns / 1ps

module moore\_1101(

input seq, clock, reset,

output reg out1, reg out2, reg out3, reg out4, reg out5, reg detect

);

always @(posedge clock) begin

if(reset==0) begin

out1 = 1;

out2 = 0;

out3 = 0;

out4 = 0;

out5 = 0;

detect = 0;

end

else if(out1==1) begin

out2 = seq;

out1 = ~seq;

end

else if(out2==1) begin

out3 = seq;

out1 = ~seq;

out2 = 0;

end

else if(out3==1) begin

out4 = ~seq;

out3 = seq;

end

else if(out4==1) begin

out5 = seq;

out1 = ~seq;

out4 = 0;

detect = seq;

end

else if(out5==1) begin

out1 = ~seq;

out3 = seq;

out5 = 0;

detect = 0;

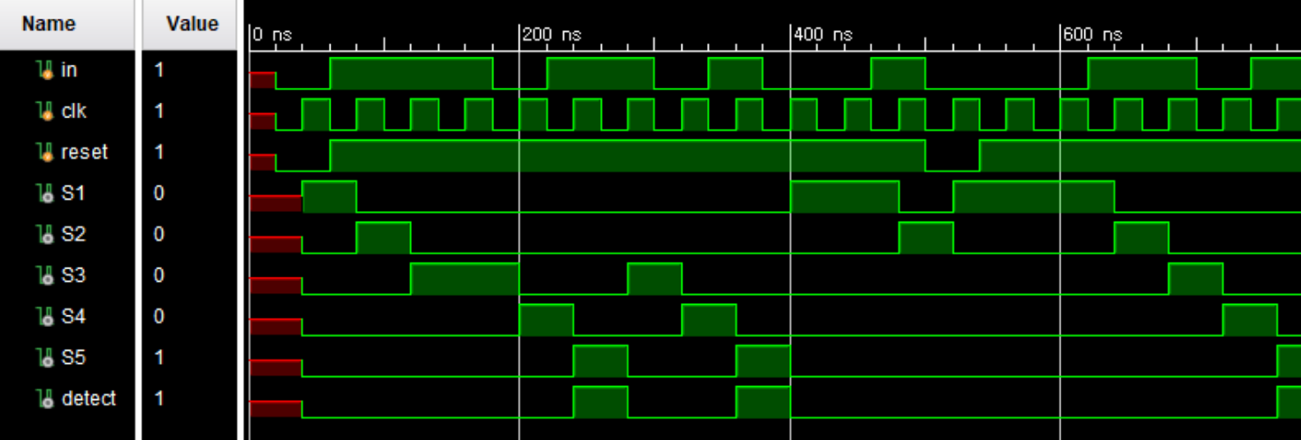
end

end

endmodule

<sequence detector 1101 Moore machine의 Verilog code>

또한 이에 대한 simulation 결과는 아래와 같다.



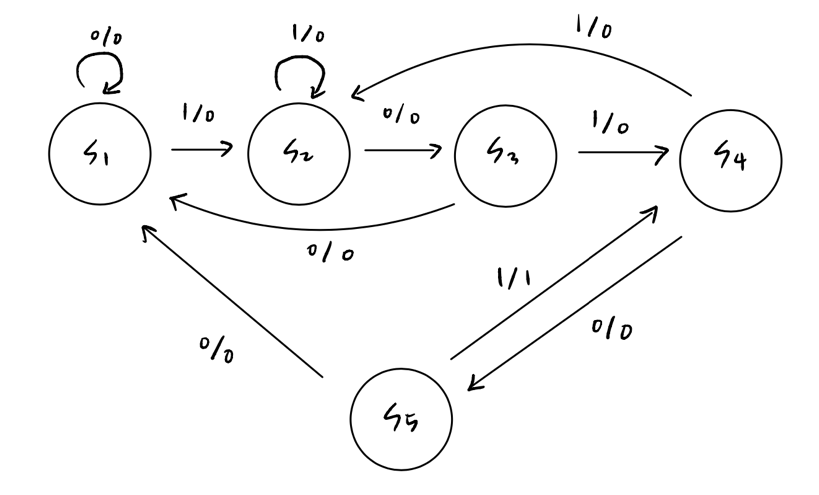
<sequence detector 1101 Moore machine의 Verilog code>

결과를 보면, 우선 기본적으로 reset을 통해서 S1 = 1, S2,,,5 = 0을 만드는 형태이다. 이때 이 code를 FPGA에서 실행시킬 때 reset button이 active LOW 형태이기 때문에, reset = 0인 경우에 reset을 수행할 수 있도록 code를 작성했다. 이후 simulation에서 각 clock마다 나오는 output은 S1 = 0, S2 = 1, S3 = 11, S4 = 110, S5 = 1101(+ S5 = detect의 형태)의 상태를 잘 나타내고 있음을 파악할 수 있다. 또한 overlapping 형태로 구현했다는 점을 code, simulation 을 통해서 파악할 수 있는데, 1101의 sequence가 검출된 이후, 1이 in로 들어가면 11의 상태인 S3 = 1이 되는 형태로, overlapping을 바탕으로 구현을 수행했다.

**2. Sequence detector 10101 machine**

**2.2. Sequence detector 10101 Mealy machine**

실습과 동일하게 10101 sequence detector 또한 Mealy/Moore 형태로 구현할 수 있다. 이때 Mealy machine은 5개의 state (S1 = 0, S2 = 1, S3 = 10, S4 = 101, S5 = 1010)을 가짐을 파악할 수 있으며, 이에 대한 state diagram과 state table은 아래와 같다.



<sequence detector 10101 Mealy machine의 state diagram>

|  |  |  |
| --- | --- | --- |
| 10101 Mealy machine의 state table | | |
| present state | next state | |
| input = 0 | input = 1 |
| S1 | S1 (output = 0) | S2 (output = 0) |
| S2 | S3 (output = 0) | S2 (output = 0) |
| S3 | S1 (output = 0) | S4 (output = 0) |
| S4 | S5 (output = 0) | S2 (output = 0) |
| S5 | S0 (output = 0) | S4 (output = 1) |

<sequence detector 10101 Mealy machine의 state table>

이를 바탕으로 구현한 code는 아래와 같다.

`timescale 1ns / 1ps

module mealy\_10101(

input seq, clock, reset,

output reg out1, reg out2, reg out3, reg out4, reg out5, reg detect

);

always @(posedge clock) begin

if(reset==0) begin

out1 = 1;

out2 = 0;

out3 = 0;

out4 = 0;

out5 = 0;

detect = 0;

end

else if(out1==1) begin

out2 = seq;

out1 = ~seq;

end

else if(out2==1) begin

out3 = ~seq;

out2 = seq;

end

else if(out3==1) begin

out4 = seq;

out1 = ~seq;

out3 = 0;

end

else if(out4==1) begin

detect = 0;

out5 = ~seq;

out2 = seq;

out4 = 0;

detect = 0;

end

else if(out5==1) begin

out4 = seq;

detect = seq;

out1 = ~seq;

out5 = 0;

end

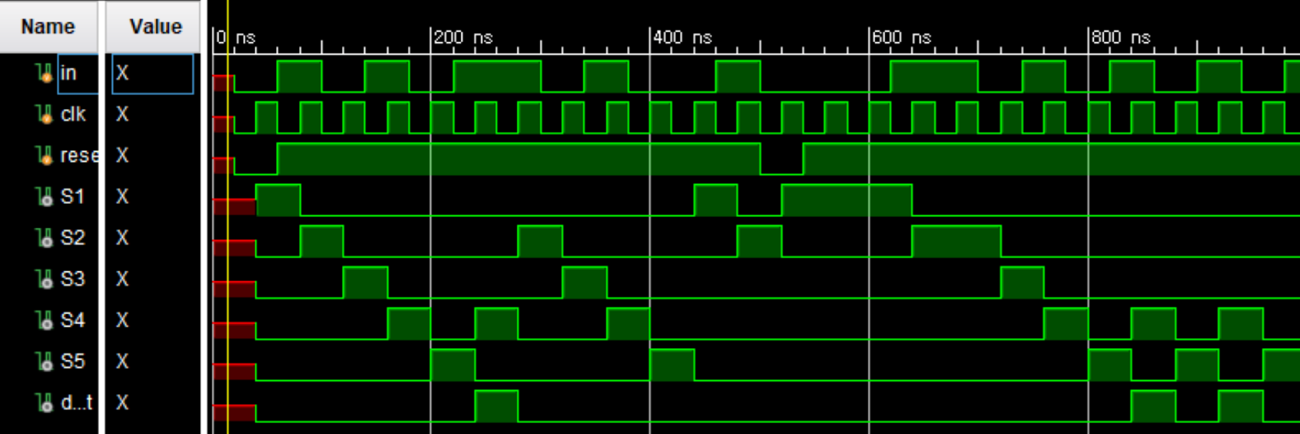
end

endmodule

<sequence detector 10101 Mealy machine의 Verilog code>

위 code에서 각 out은 diagram에서 각 S를 의미하며, detect = output을 의미한다. 또한 Mealy machine이기 때문에 5개의 state를 갖는 형태임을 파악할 수 있다.

이에 대한 simulation 결과는 아래와 같다.

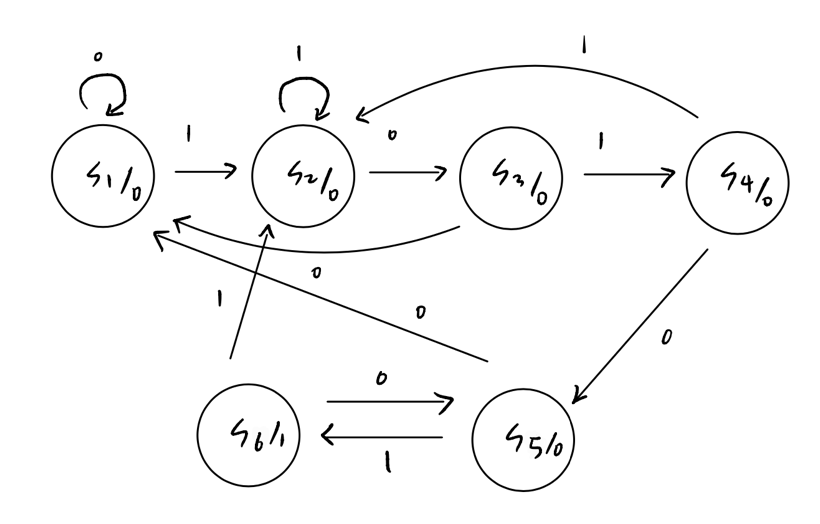


<sequence detector 10101 Mealy machine의 simulation 결과>

위에서도 동일하게 설명한 것인데, reset = 0일 때 reset 과정을 진행하며, 이후에는 overlapping을 기반으로 simulation이 진행된다. 따라서 input = 01010101(가장 오른쪽 부분)이 들어올 때 S1 = 1 -> S2 = 1 => S3 = 1 -> S4 = 1 -> S5 = 1 -> S4 = 1(output = 1)-> S5 = 1 -> S4 = 1(output = 1) 형태로 진행되는 것을 파악할 수 있다.

**2.2. Sequence detector 10101 Moore machine**

Moore machine은 10101의 검출을 수행하기 위해서 6개의 state (S1 = 0, S2 = 1, S3 = 10, S4 = 101, S5 = 1010, S6 = 10101)을 가져야 한다는 것을 파악할 수 있다. 또한 S6일 때 output = 1을 가진다. 이에 대한 state diagram과 state table은 아래와 같다.



<sequence detector 10101 Moore machine의 state diagram>

|  |  |  |  |
| --- | --- | --- | --- |
| 10101 Moore machine의 state table | | | |
| present state | next state | | output |
| input = 0 | input = 1 |
| S1 | S1 | S2 | 0 |
| S2 | S3 | S2 | 0 |
| S3 | S1 | S4 | 0 |
| S4 | S5 | S2 | 0 |
| S5 | S0 | S6 | 0 |
| S6 | S5 | S2 | 1 |

<sequence detector 10101 Moore machine의 state table>

이를 바탕으로 Moore machine을 구현한 code는 아래와 같다.

`timescale 1ns / 1ps

module moore\_10101(

input seq, clock, reset,

output reg out1, reg out2, reg out3, reg out4, reg out5, reg out6, reg detect

);

always @(posedge clock) begin

if(reset==0) begin

out1 = 1;

out2 = 0;

out3 = 0;

out4 = 0;

out5 = 0;

out6 = 0;

detect = 0;

end

else if(out1==1) begin

out2 = seq;

out1 = ~seq;

end

else if(out2==1) begin

out3 = ~seq;

out2 = seq;

end

else if(out3==1) begin

out4 = seq;

out1 = ~seq;

out3 = 0;

end

else if(out4==1) begin

out5 = ~seq;

out2 = seq;

out4 = 0;

end

else if(out5==1) begin

out6 = seq;

detect = seq;

out1 = ~seq;

out5 = 0;

end

else if(out6==1) begin

out2 = seq;

out5 = ~seq;

out6 = 0;

detect = 0;

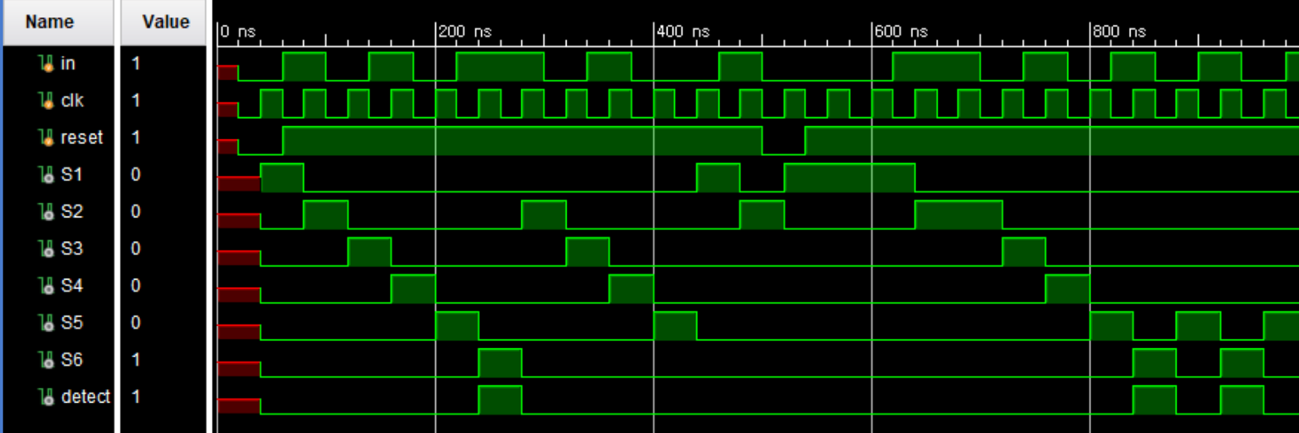
end

end

endmodule

<sequence detector 10101 Moore machine의 Verilog code>

이에 대한 simulation 결과는 아래와 같다.



<sequence detector 10101 Moore machine의 simulation 결과>