14주차 결과보고서

전공: 심리학과 학년: 3학년 학번: 20190345 이름: 김동현

1.

1101 Sequence detector를 moore machine으로 구현하였다. 이를 state diagram으로 표현하면 아래의 그림과 같다. A는 아무런 입력도 없는 상태, B는 1이 들어온 상태, C는 11이 들어온 상태, D는 110이 들어온 상태, E는 1101이 들어온 상태이다. 이때 상태에 따라 출력값이 결정되는 것은 Moore machine의 특징으로,상태와 출력값이 같이 표시되고 있다. 입력값이 0인지 1인지에 따라 다음상태 혹은 자기자신으로 상태가 변화된다. 이때 1101이 나타난 이후에도 101이 들어온다면 다시 1101 sequence가 입력된 것과 같이 overlapping이 발생하는 것을 알 수 있다.

텍스트, 손목시계이(가) 표시된 사진

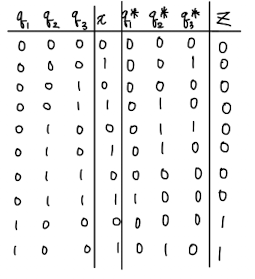
자동 생성된 설명

State diagram을 통해 state table을 표현하면 아래의 그림과 같다. 현재 상태에 따라 출력값이 결정되기 때문에, 1101을 나타내는 E상태의 경우 1이 출력된다.

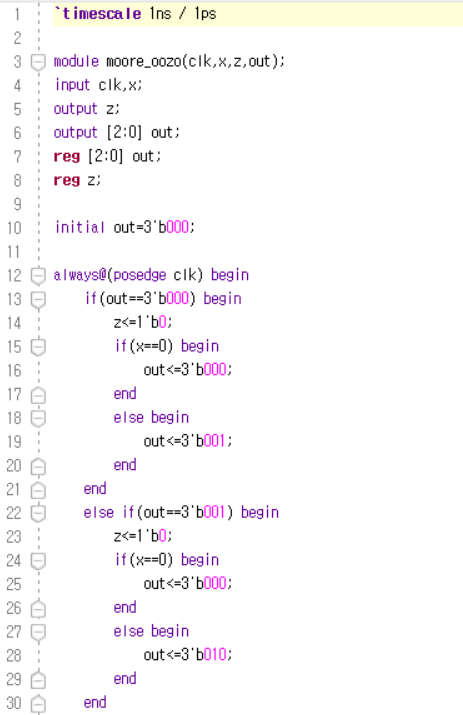
테이블이(가) 표시된 사진

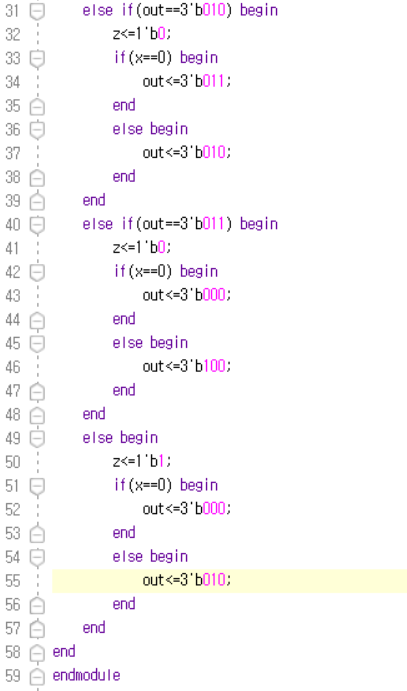
자동 생성된 설명

상태를 logic diagram으로 나타내기 위해 2진수로 표현하고자 한다. 이를 위해 5개의 상태를 3개의 비트로 표현하고자 한다. A는 000, B는 001, C는 010, D는 011, E는 100으로 표현하며, 각 비트는 q1,q2,q3로 표현한다. q1\*, q2\*,q3\*는 각 비트의 다음 상태를, x는 입력, z는 출력값을 의미한다.

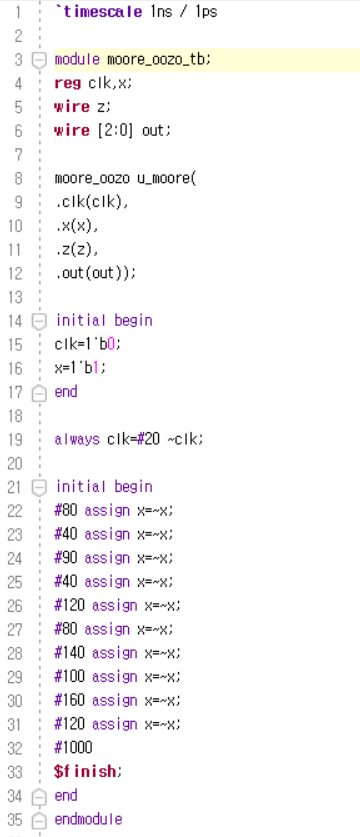
,

이러한 state table을 이용하여 다음과 같이 verilgo로 구현하였다. 아래는 1101 sequence detector moore machine의 design source code이다.

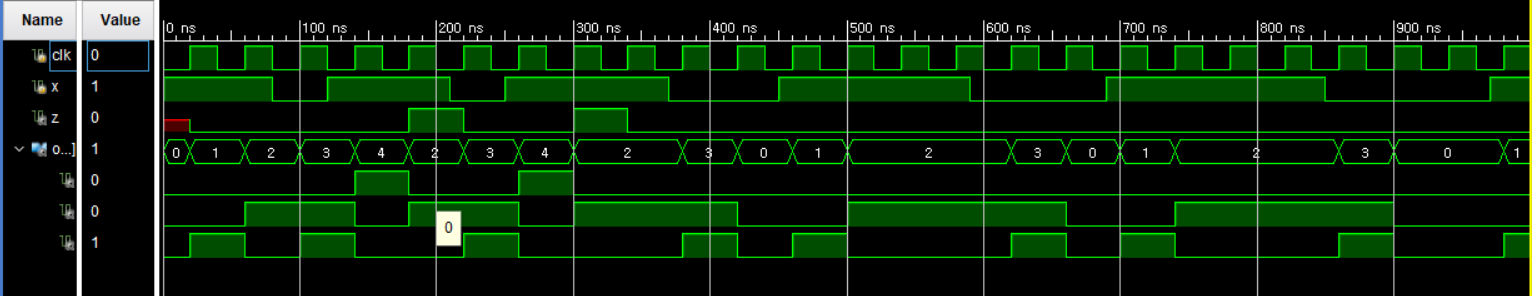




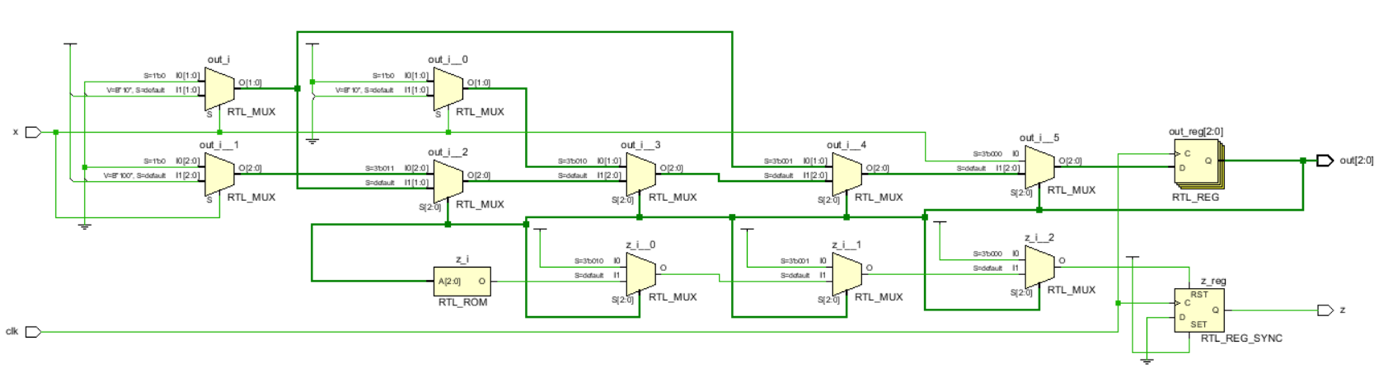
Clk는 클럭펄스를, x는 입력값, z는 출력값, out배열은 상태를 의미한다. 초기 상태는 000으로 시작한다. 이때 클럭펄스가 rising edge일 경우 현재 상태와 입력값을 if-else if-else문을 활용하여 다음 상태와 출력값을 결정한다. 1101 sequence를 나타내는 100일 경우 출력값을 1로 설정하고, 그 이외의 경우에는 출력값을 0으로 설정한다. 이를 시뮬레이션으로 나타내기 위해 아래와 같이 simulation source code를 구현하였다.



Clk은 0, x는 1로 초기값을 설정하고, 20클럭마다 clk를 변화하도록 하였다. 이때 1101이라는 sequence를 구현하기 위해 x값의 변화를 하드 코딩으로 나타내었다. 아래는 시뮬레이션 결과이다.



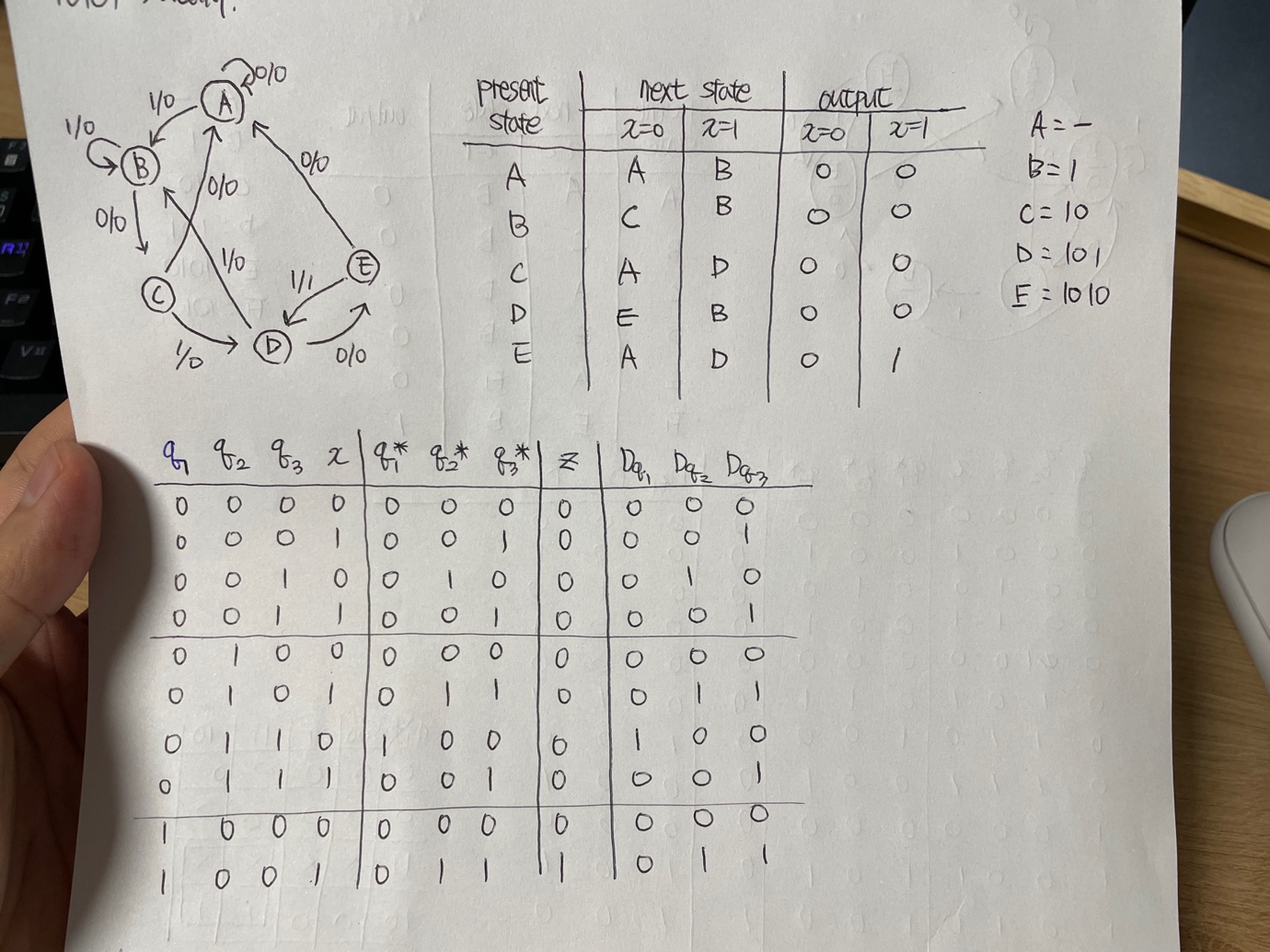
Clk가 rising하는 순간 상태가 변화하는 것을 확인할 수 있으며, 000->001->010->011->100으로 out값이 진행되어야 1101을 나타낸다. 위의 시뮬레이션 결과를 보면 0ns~200ns까지 1101이 입력되어 결과값이 1로 출력되는 것을 확인할 수 있다. Overlapping을 확인하기 위해 1101이 입력된 이후 101을 입력하였다. 이는 200ns ~ 300ns까지 101이 입력되어 결과값이 1로 출력되는 것을 확인할 수 있다. 이외의 경우에는 1101 sequence가 확인되지 않아 결과값이 0으로 출력되는 것을 확인할 수 있다. 뿐만 아니라 입력이 입력되고 다음 클럭에서 출력값이 나타나는 moore machine의 특징을 확인할 수 있다.



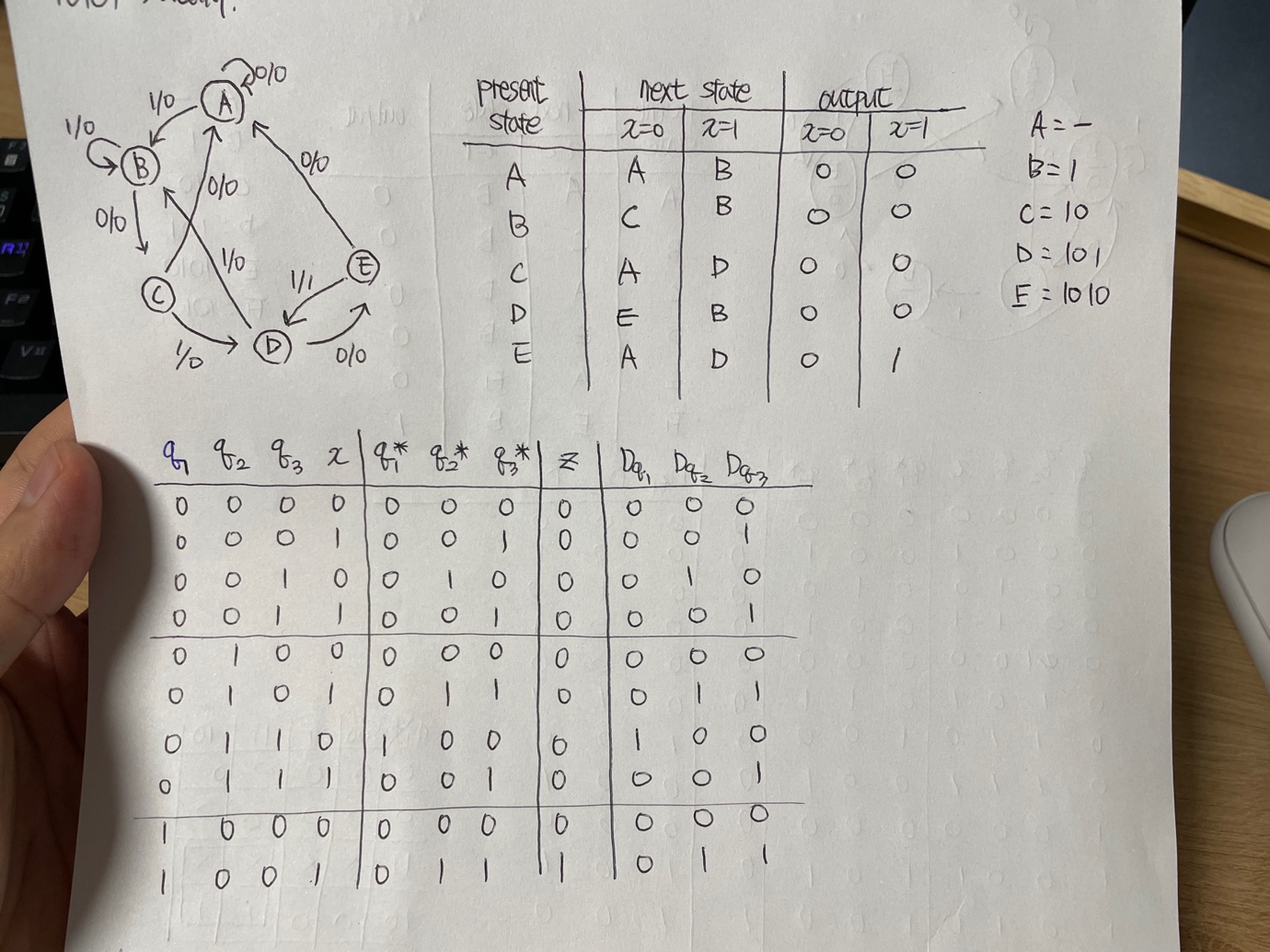
위는 1101 sequence detector moore machine의 schematic이다.

2.

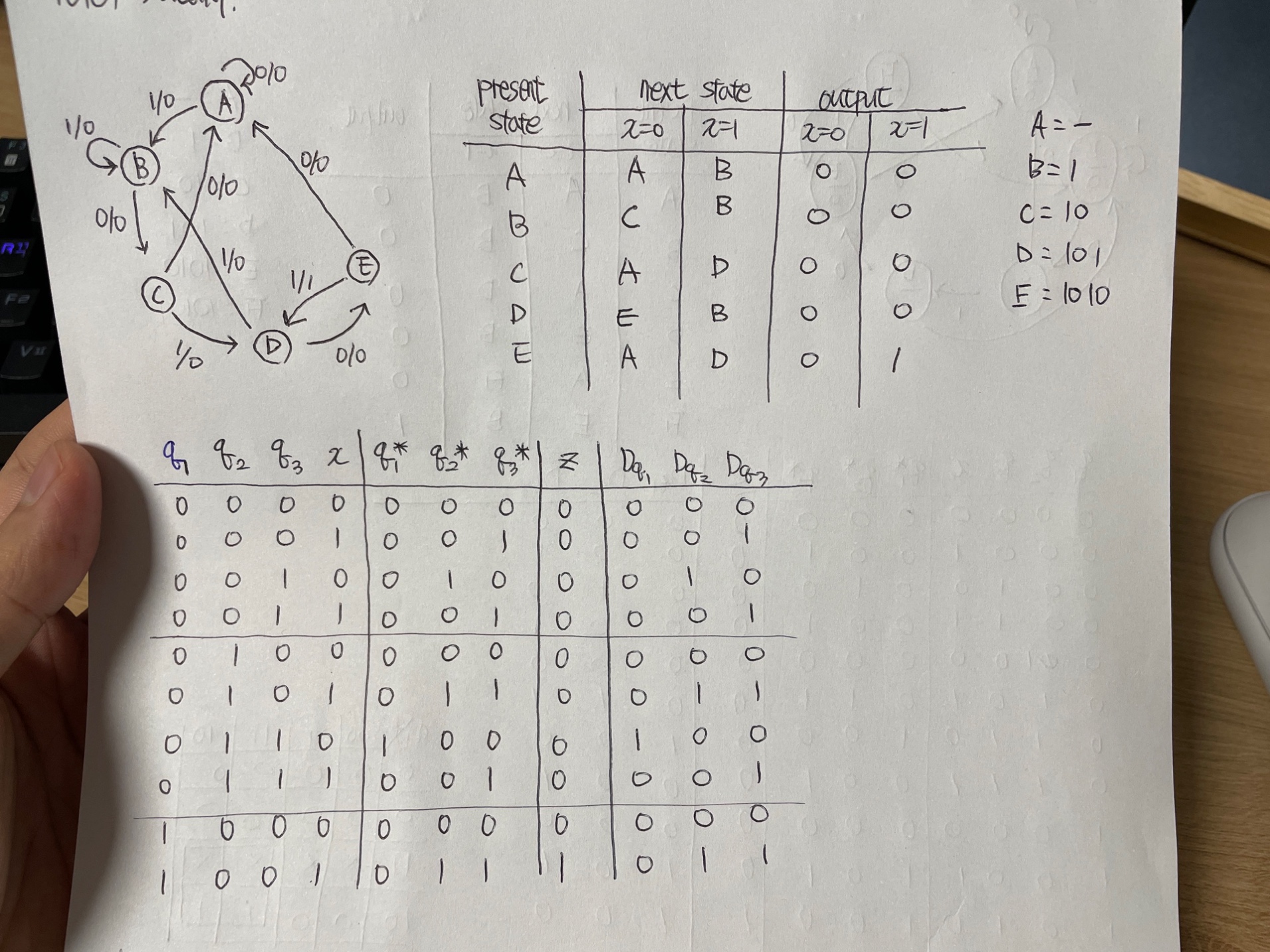
10101 Sequence detector를 mealy machine으로 구현하였다. 이를 state diagram으로 표현하면 아래의 그림과 같다. A는 아무런 입력도 없는 상태, B는 1이 들어온 상태, C는 10이 들어온 상태, D는 101이 들어온 상태, E는 1010이 들어온 상태이다. 이때 상태와 입력값에 따라 출력값이 결정되는 것은 Mealy machine의 특징으로, 입력값과 출력값이 같이 표시되고 있다. 입력값이 0인지 1인지에 따라 다음상태 혹은 자기자신으로 상태가 변화하며 출력값이 0인지 1인지가 결정된다. 이로 인해 10101을 나타내는 상태가 존재하지 않고, 1010 상태에서 1이 입력되었을 때, 1을 출력하고 overlapping을 커버하기 위해 101상태로 전이되는 것을 확인할 수 있다.



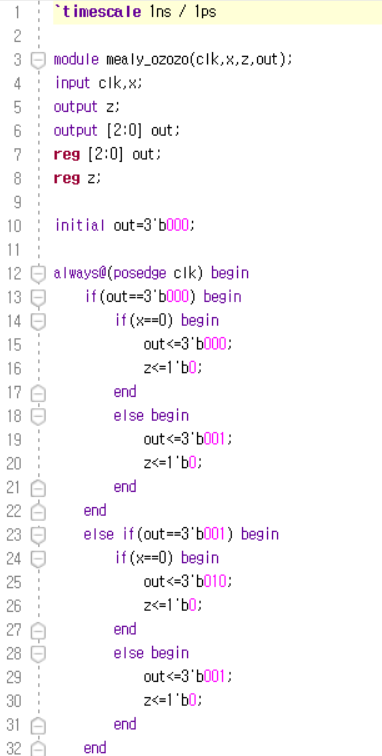
State diagram을 통해 state table을 표현하면 아래의 그림과 같다. 현재 상태와 입력값에 따라 출력값이 결정되기 때문에, 1010을 나타내는 E상태에서 1이 입력되는 경우에만 1이 출력된다.

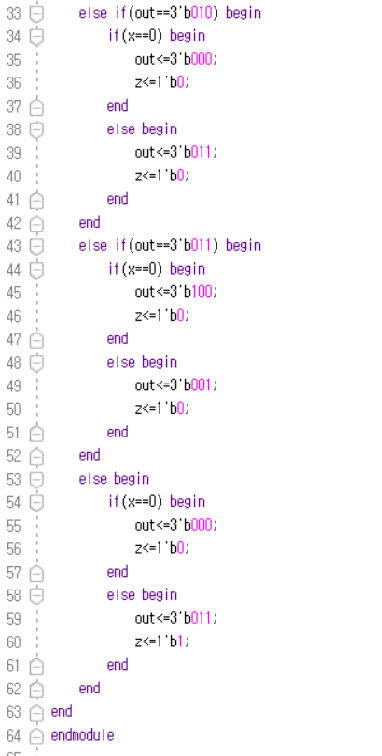


상태를 logic diagram으로 나타내기 위해 2진수로 표현하고자 한다. 이를 위해 5개의 상태를 3개의 비트로 표현하고자 한다. A는 000, B는 001, C는 010, D는 011, E는 100으로 표현하며, 각 비트는 q1,q2,q3로 표현한다. q1\*, q2\*,q3\*는 각 비트의 다음 상태를, x는 입력, z는 출력값을 의미한다.



이러한 state table을 이용하여 다음과 같이 verilog로 구현하였다. 아래는 10101 sequence detector mealy machine의 design source code이다.





Clk는 클럭펄스를, x는 입력값, z는 출력값, out배열은 상태를 의미한다. 초기 상태는 000으로 시작한다. 이때 클럭펄스가 rising edge일 경우 현재 상태와 입력값을 if-else if-else문을 활용하여 다음 상태와 출력값을 결정한다. 1010 sequence를 나타내는 100일 경우 입력값이 1일 경우 출력값을 1로 설정하고, 그 이외의 경우에는 출력값을 0으로 설정한다. 이를 시뮬레이션으로 나타내기 위해 아래와 같이 simulation source code를 구현하였다.

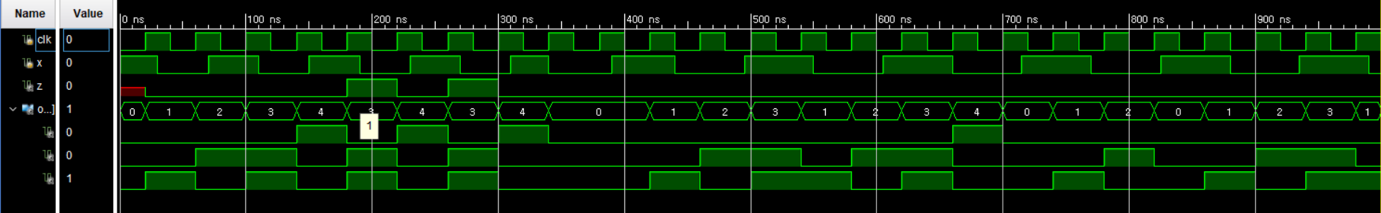
테이블이(가) 표시된 사진

자동 생성된 설명

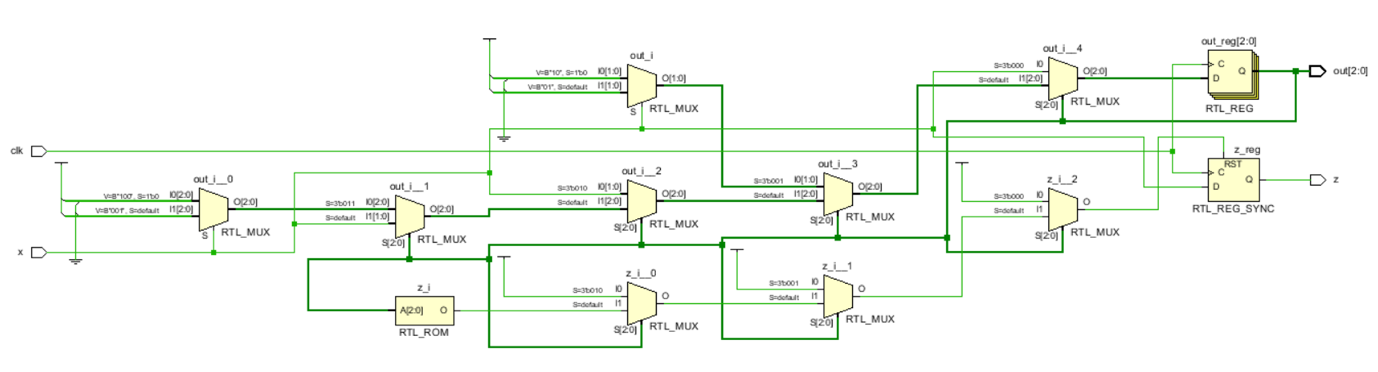
테이블이(가) 표시된 사진

자동 생성된 설명

Clk은 0, x는 1로 초기값을 설정하고, 20클럭마다 clk를 변화하도록 하였다. 이때 10101이라는 sequence를 구현하기 위해 x값의 변화를 하드 코딩으로 나타내었다. 아래는 시뮬레이션 결과이다.



Clk가 rising하는 순간 상태가 변화하는 것을 확인할 수 있으며, 000->001->010->011->100으로 out값이 진행되고 이후 입력값이 1이 들어와야 10101을 나타낸다. 위의 시뮬레이션 결과를 보면 0ns~200ns까지 10101이 입력되어 결과값이 1로 출력되는 것을 확인할 수 있다. Overlapping을 확인하기 위해 10101이 입력된 이후 01을 입력하였다. 이는 200ns ~ 300ns까지 01이 입력되어 결과값이 1로 출력되는 것을 확인할 수 있다. 이외의 경우에는 10101 sequence가 확인되지 않아 결과값이 0으로 출력되는 것을 확인할 수 있다. 뿐만 아니라 입력값이 들어오자마자 출력값이 나타나는 mealy machine의 특징을 확인할 수 있다.



위는 10101 sequence detector mealy machine의 schematic이다.

10101 Sequence detector를 moore machine으로 구현하였다. 이를 state diagram으로 표현하면 아래의 그림과 같다. A는 아무런 입력도 없는 상태, B는 1이 들어온 상태, C는 10이 들어온 상태, D는 101이 들어온 상태, E는 1010이 들어온 상태, F는 10101이 들어온 상태이다. 이때 상태에 따라 출력값이 결정되는 것은 Moore machine의 특징으로,상태와 출력값이 같이 표시되고 있다. 입력값이 0인지 1인지에 따라 다음상태 혹은 자기자신으로 상태가 변화된다. 이때 10101이 나타난 이후에도 01이 들어온다면 다시 10101 sequence가 입력된 것과 같이 overlapping이 발생하는 것을 알 수 있다.

텍스트이(가) 표시된 사진

자동 생성된 설명

State diagram을 통해 state table을 표현하면 아래의 그림과 같다. 현재 상태에 따라 출력값이 결정되기 때문에, 10101을 나타내는 F상태의 경우 1이 출력된다.

텍스트, 화이트보드이(가) 표시된 사진

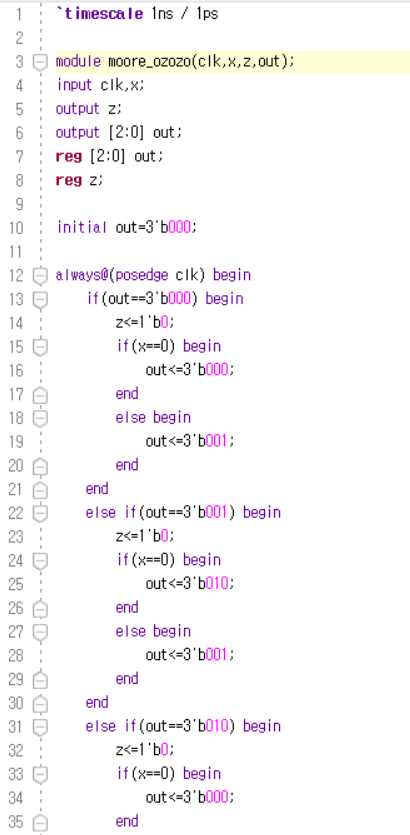
자동 생성된 설명

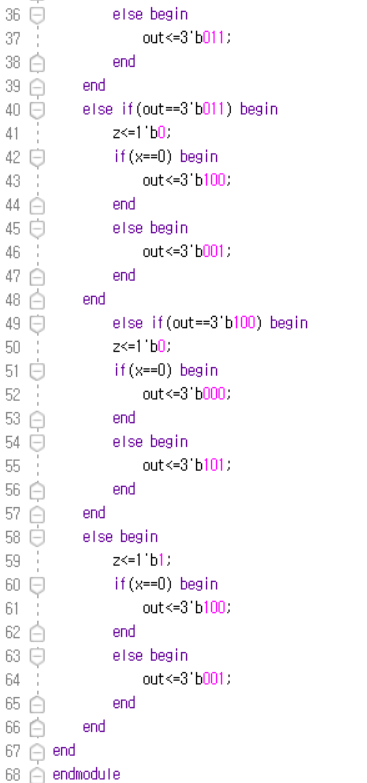
상태를 logic diagram으로 나타내기 위해 2진수로 표현하고자 한다. 이를 위해 6개의 상태를 3개의 비트로 표현하고자 한다. A는 000, B는 001, C는 010, D는 011, E는 100, F는 101으로 표현하며, 각 비트는 q1,q2,q3로 표현한다. q1\*, q2\*,q3\*는 각 비트의 다음 상태를, x는 입력, z는 출력값을 의미한다.

텍스트, 화이트보드이(가) 표시된 사진

자동 생성된 설명

이러한 state table을 이용하여 다음과 같이 verilog로 구현하였다. 아래는 10101 sequence detector moore machine의 design source code이다.





Clk는 클럭펄스를, x는 입력값, z는 출력값, out배열은 상태를 의미한다. 초기 상태는 000으로 시작한다. 이때 클럭펄스가 rising edge일 경우 현재 상태와 입력값을 if-else if-else문을 활용하여 다음 상태와 출력값을 결정한다. 10101 sequence를 나타내는 101일 경우 출력값을 1로 설정하고, 그 이외의 경우에는 출력값을 0으로 설정한다. 이를 시뮬레이션으로 나타내기 위해 아래와 같이 simulation source code를 구현하였다.

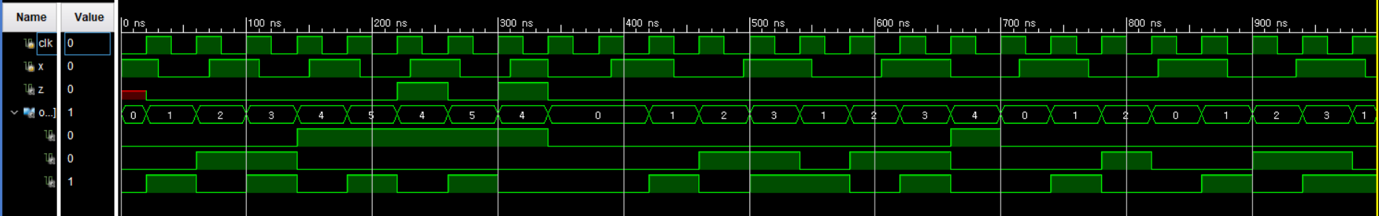
테이블이(가) 표시된 사진

자동 생성된 설명

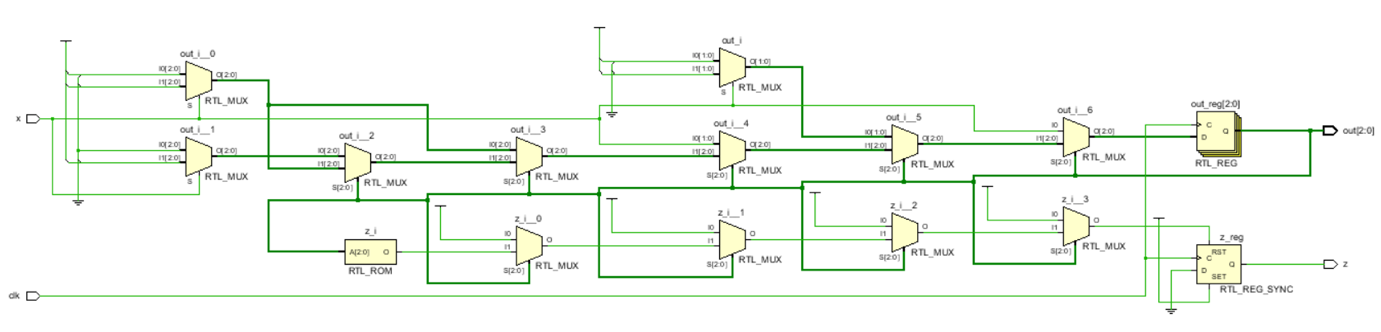
테이블이(가) 표시된 사진

자동 생성된 설명

Clk은 0, x는 1로 초기값을 설정하고, 20클럭마다 clk를 변화하도록 하였다. 이때 10101이라는 sequence를 구현하기 위해 x값의 변화를 하드 코딩으로 나타내었다. 아래는 시뮬레이션 결과이다.



Clk가 rising하는 순간 상태가 변화하는 것을 확인할 수 있으며, 000->001->010->011->100->101으로 out값이 진행되어야 10101을 나타낸다. 위의 시뮬레이션 결과를 보면 0ns~250ns까지 10101이 입력되어 결과값이 1로 출력되는 것을 확인할 수 있다. Overlapping을 확인하기 위해 10101이 입력된 이후 01을 입력하였다. 이는 250ns ~ 300ns까지 01이 입력되어 결과값이 1로 출력되는 것을 확인할 수 있다. 이외의 경우에는 10101 sequence가 확인되지 않아 결과값이 0으로 출력되는 것을 확인할 수 있다. 뿐만 아니라 입력이 입력되고 다음 클럭에서 출력값이 나타나는 moore machine의 특징을 확인할 수 있다.



위는 10101 sequence detector moore machine의 schematic이다.