

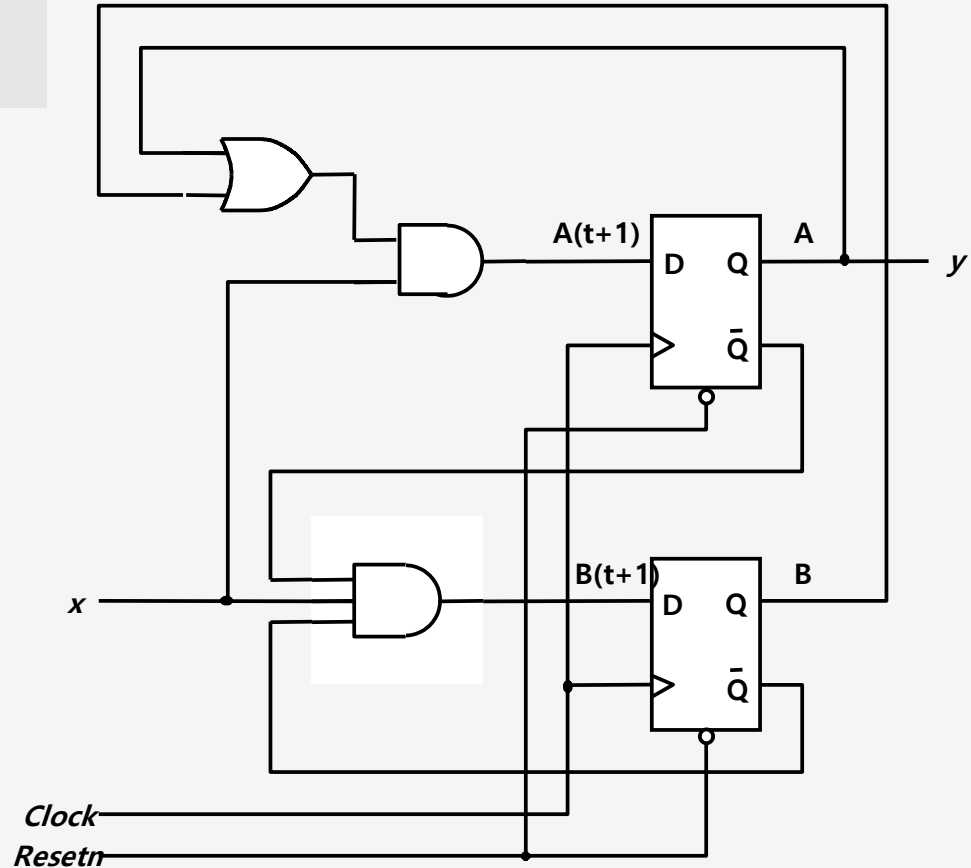
# SSL analysis & Design

컴퓨터 로직설계

휴먼지능정보공학과  
201810776 소재휘

# Lab 04-1: SSL Analysis

1. Analyze the following circuit **manually in a paper**.
  - Obtain the state diagram.
  - Build a timing sequence with **your own arbitrary input sequence** & initial state.
  - Draw the timing diagram that matches the timing sequence.
2. Design the circuit with BDF in Quartus II.
  - Obtain a timing diagram with input sequence & initial state used in Step 1.
3. Verify the analysis with comparing the timing diagrams obtained in Step 1 & Step 2.





## #4-1 State-table/State-equation

### SSL analysis & Design

우선 주어진 회로를 분석하여 현재 State와 Next-state를 한 눈에 볼 수 있도록 State-table을 만들었습니다.

그리고 이 State-table을 이용하여 K맵을 통한 State-equation을 도출할 수도 있지만, 회로가 주어져 있으므로 Next-state들과 Output Y에 대한 SOP expression을 도출해 낼 수 있었습니다.

#1. State-table

State		Next-state				Output Y
		x=0		x=1		
		A	B	A	B	
0	0	0	0	0	1	0
0	1	0	0	1	0	0
1	0	0	0	1	0	1
1	1	0	0	1	0	1

#1 state-equation

$$A(t+1) = X(A+B)$$

$$B(t+1) = A'B'X$$

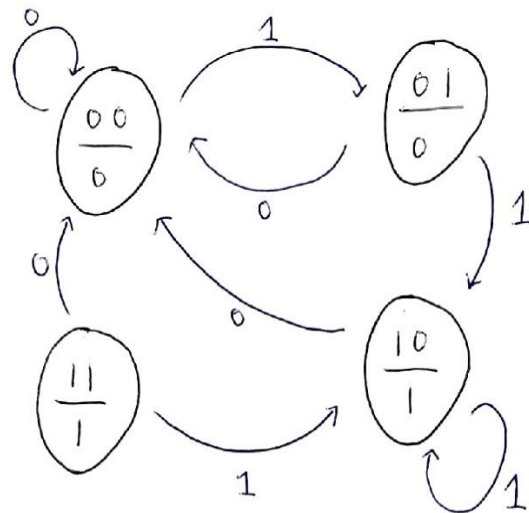
$$Y = A$$



## #4-1 State-diagram

### SSL analysis & Design

#1 State-diagram.



앞에서 만든 State-table을 바탕으로 State-diagram을 도출해 냈습니다.  
00,01,10,11인 상태를 각각 s1, s2, s3, s4라고  
한다면 각각의 output y는 0,0,1,1 이었고,  
s1에서 input이 0이면 s1, 1이면 s2로 상태가  
바뀌는 것과 같은 상태 전이를  
diagram으로 나타냈습니다.



## #4-1 Timing-sequence/diagram

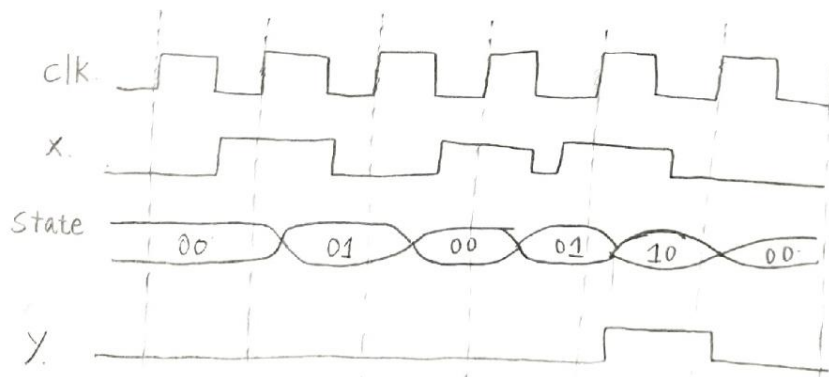
### SSL analysis & Design

다음은 직접 Quartus를 통하여 구현을 하기 전  
예상되는 timing-sequence와 timing-  
diagram을 구현하였습니다.

다음 timing-sequence를 보면 clock의 rising  
edge 부분의 x의 값에 따라서 State의 변화를  
예상하였습니다. 예를 들어 00인 상태에서 다음  
clock의 rising edge에서 x의 값이 1이 되면  
State가 01로 변할 것입니다.

또한 timing-sequence와 timing-diagram에서  
확인할 수 있는 사실이 있습니다. T1에서 rising  
edge에서 x의 값이 1이므로 현재 상태가 바로  
01로 변하는 것이 아닌 00임을 예상하였습니다.  
왜냐하면 전파 지연을 감안하였기 때문입니다.

# 1 timing - sequence & timing - diagram.



	t <sub>0</sub>	t <sub>1</sub>	t <sub>2</sub>	t <sub>3</sub>	t <sub>4</sub>	t <sub>5</sub>	t <sub>6</sub>
x	0	1	0	1	1	0	0
State	00	00	01	00	01	10	00
y	0	0	0	0	0	1	0



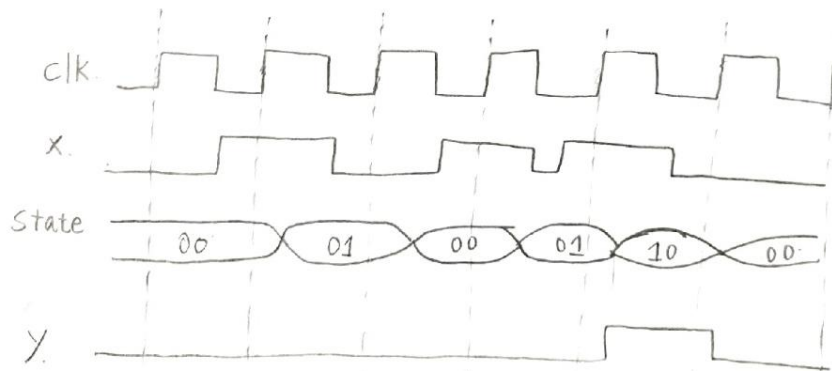
# #4-1 Timing-sequence/diagram

## SSL analysis & Design

따라서 제가 직접 회로를 설계하고 시뮬레이팅을 할 때 확인해야 할 사항은 크게 두 가지입니다.

1. Clock의 rising edge에서의 State-diagram대로의 Next-State의 변화 검증
2. 전파 지연으로 인한 State와 output y의 즉각적인 변화가 일어나지 않음.
3. Reset의 값에 따른 State의 값의 변화

#1 timing-sequence & timing-diagram.

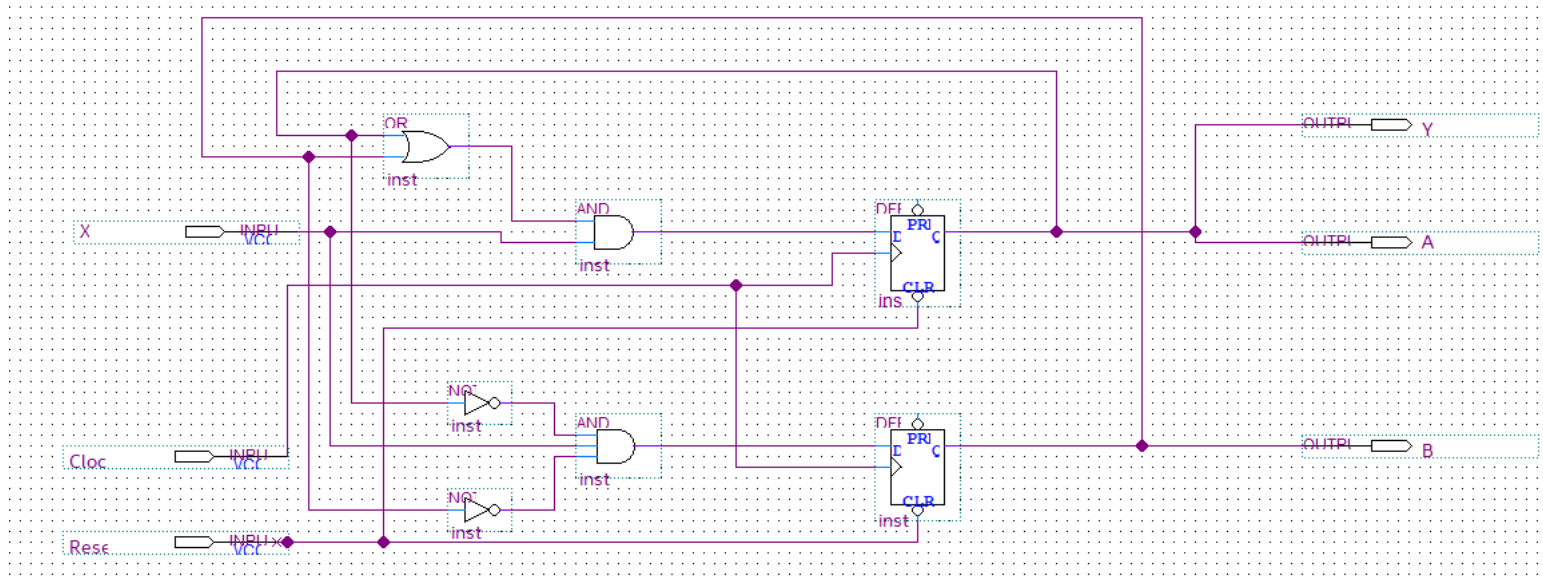


	$t_0$	$t_1$	$t_2$	$t_3$	$t_4$	$t_5$	$t_6$
x	0	1	0	1	1	0	0
State	00	00	01	00	01	10	00
y	0	0	0	0	0	1	0



## #4-1 Analysis circuit

SSL analysis & Design

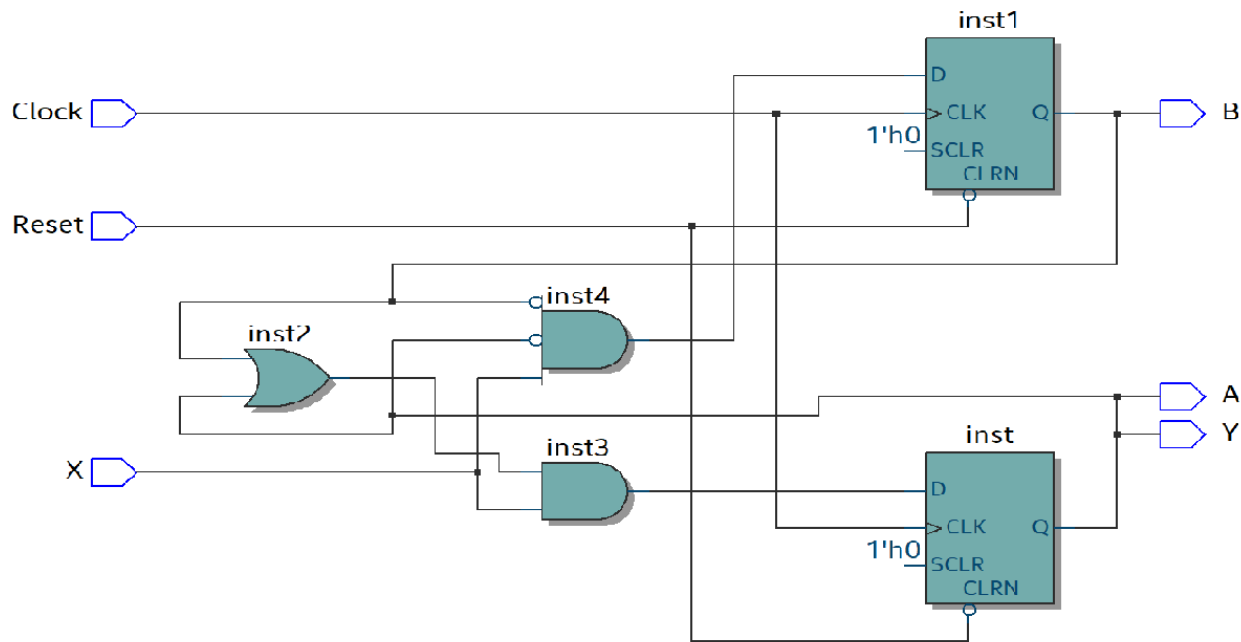


직접 Quartus를 이용하여 해당 회로를 구현하였습니다.



## #4-1 RTL viewer

SSL analysis & Design



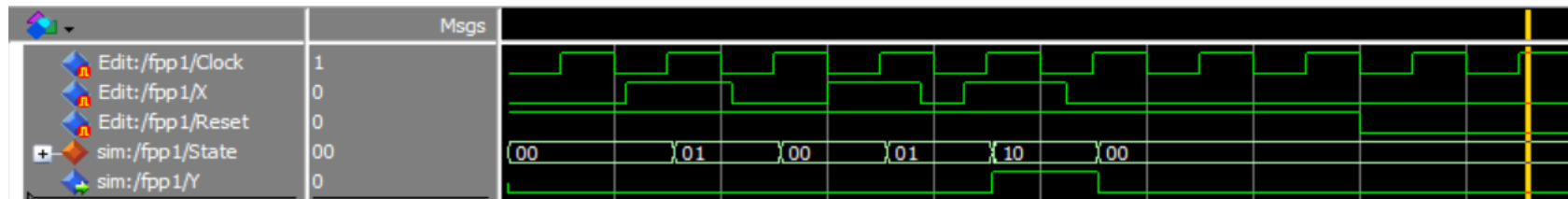
RTL viewer를 통하여 제가 구현하려는 회로와 디자인한 회로가 일치함을 확인하였습니다.





## #4-1 Simulation

### SSL analysis & Design



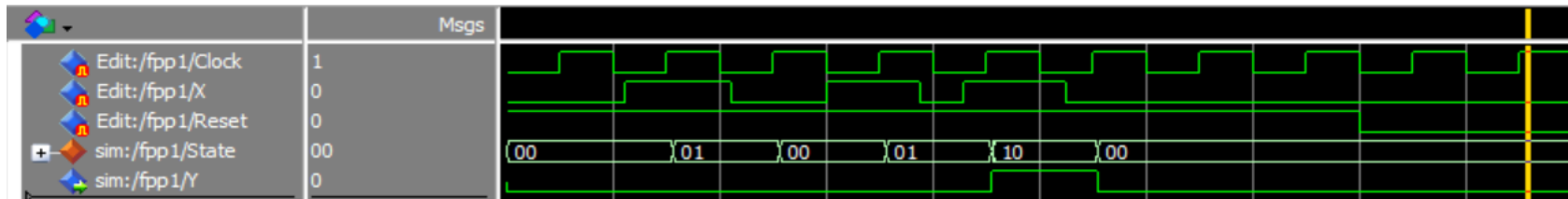
Clock의 rising edge에서의 State의 변화를 검증해 보겠습니다. 00,01,10,11인 상태를 각각 s0,s1,s2,s3라고 하겠습니다. 시뮬레이션을 Timing-sequence로 나타내 보겠습니다.

	t0	t1	t2	t3	t4	t5	t6
x	0	1	0	1	1	0	0
state	s0	s0	s1	s0	s1	s2	s0
y	0	0	0	0	0	1	0

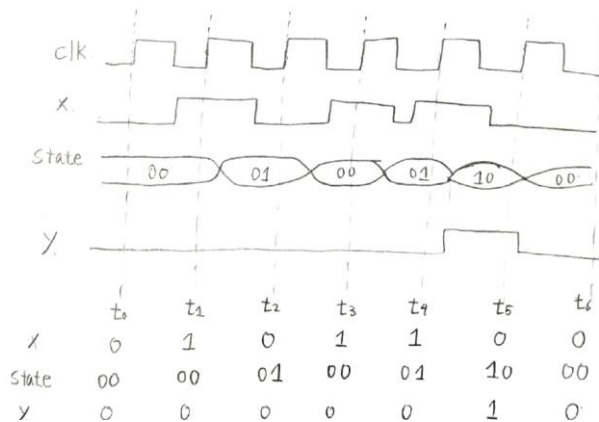


# #4-1 Simulation

## SSL analysis & Design



#1 Timing-sequence & timing-diagram.



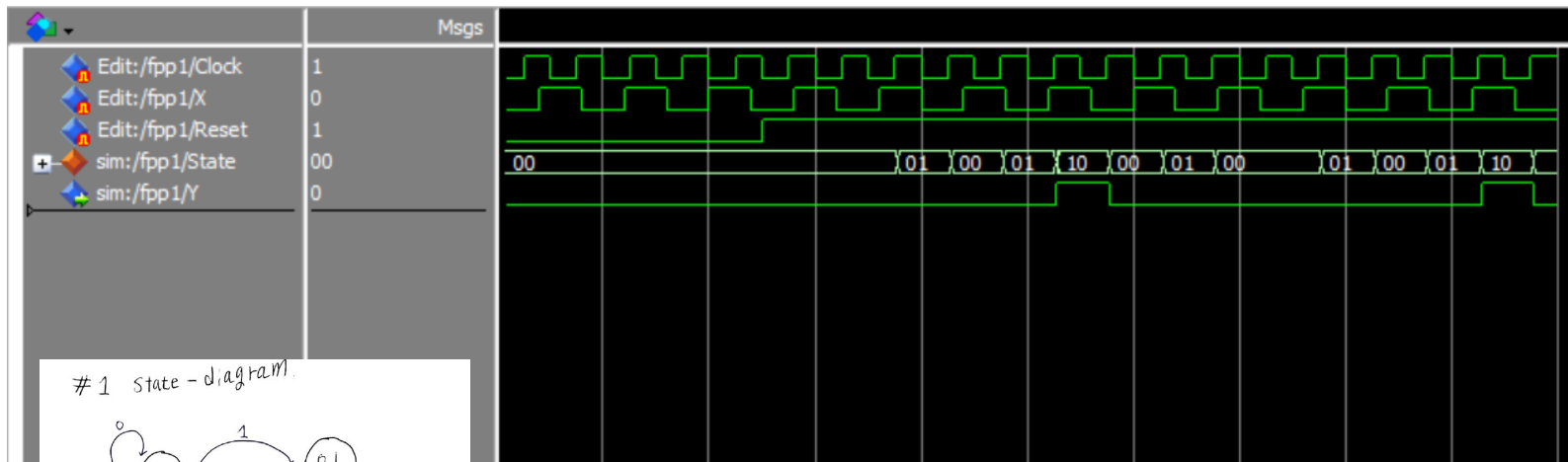
이 시뮬레이팅을 통하여 도출한 Timing Diagram이 제가 예상하였던 Timing sequence와 diagram과 동일함을 알 수 있었습니다.

예를 들어  $t_2$ 에서의 rising edge에서의 x값이 0이므로  $t_2$ 의 Next-state인  $t_3$ 의 state가  $s_1$ 에서  $s_0$ 로 바뀌었습니다. 이와 같이 앞의 state-diagram대로 rising edge에서의 Input의 값에 따라서 Next-state가 변한다는 것을 검증할 수 있었습니다.

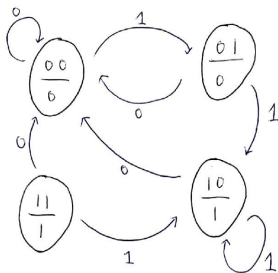


# #4-1 Simulation

## SSL analysis & Design



# 1 State-diagram

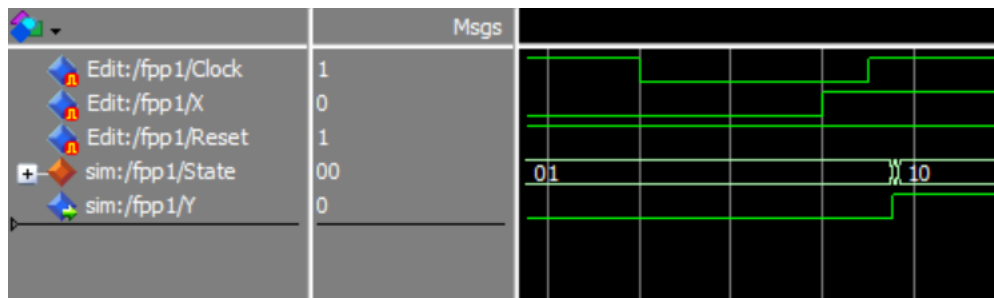


저의 Timing diagram대로 한 시뮬레이팅 외에 무작위로 X값과 clock값을 넣어 한번 더 시뮬레이팅을 해보았습니다. state-diagram의 결과를 따라 상태가 바뀌는 것을 알 수 있습니다. 따라서 State의 변화가 정상적으로 이루어짐을 검증하였습니다. 또한 reset이 0일 경우에는 00인 상태로 유지된다는 사실을 동시에 검증하였습니다.



## #4-1 Simulation

### SSL analysis & Design



마지막으로 검증할 사실은 전파 지연입니다. 다음과 같이 clock의 rising edge이 되는 시점에서 바로 현재 상태와 output의 값이 즉각적으로 바뀌지 않는 것을 알 수 있습니다. 이를 통하여 전파 지연이 일어난다는 사실을 검증하였습니다.

## Lab 04-2: SSL Design

- The circuit has one input and one output.
- All changes in the circuit occur on the positive edge of a clock signal. (including the input)
- The output is equal to 1 if during three preceding clock cycles the input was equal to 010. Otherwise, the output is 0.

- Build a timing sequence of your own with arbitrary initial state & input sequence.
- Manually draw the timing diagram which matches to the timing sequence.
- Obtain the circuit with D-FFs.
- Verify the circuit by simulation with the timing diagram you obtained in the previous step.

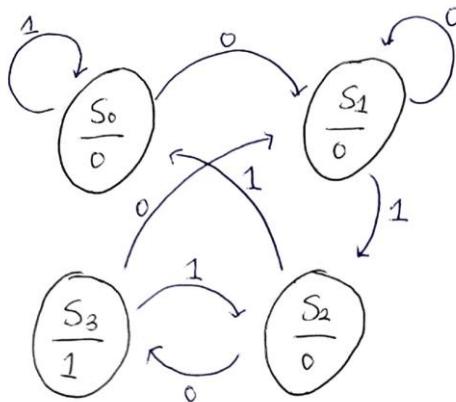


## #4-2 State-diagram

### SSL analysis & Design

'010'이 연속으로 나올 때 output이 1이 나오게 하기 위해서 우선적으로 대략적인 상태 변화를 파악하기 위해서 State-diagram을 도출해 냈습니다.  
s0, s1, s2, s3은 각각 초기 상태, '0'을 기억하는 상태, '01'을 기억하는 상태, '010'을 기억하는 상태를 뜻합니다.

#2 sequence - diagram



$S_0$  = 초기 상태.  
 $S_1$  = '0'을 기억하는 상태.  
 $S_2$  = '01'을 기억하는 상태.  
 $S_3$  = '010'을 기억하는 상태.



## #4-2 State-table/State-equation

### SSL analysis & Design

State-diagram을 통하여 나올 수 있는 State와 Next-state의 경우의 수를 한 눈에 볼 수 있도록 State-table을 만들었습니다.

State-table을 바탕으로 K-map을 통하여 Next-state인  $A(t+1)$ 과  $B(t+1)$ 의 State-equation을 도출할 수 있었습니다.  
그리고  $Y$ 의 State-equation은  $A$  and  $B(=AB)$ 임을 알 수 있습니다.

#1 State-table.

State		Next-State				Output Y
		x=0		x=1		
		A	B	A	B	
0	0	0	1	0	0	0
0	1	0	1	1	0	0
1	0	1	1	0	0	0
1	1	0	1	1	0	1

#2. State-equation.

$$A(t+1) = XB + AB'X'$$

$$B(t+1) = X'$$

$$Y = AB$$

X \ AB	00	01	11	10
0	0	0	0	1
1	0	1	0	0

X \ AB	00	01	11	10
0	0	0	0	0
1	1	1	1	1

A	B	Y
0	0	0
0	1	0
1	1	1
1	0	0

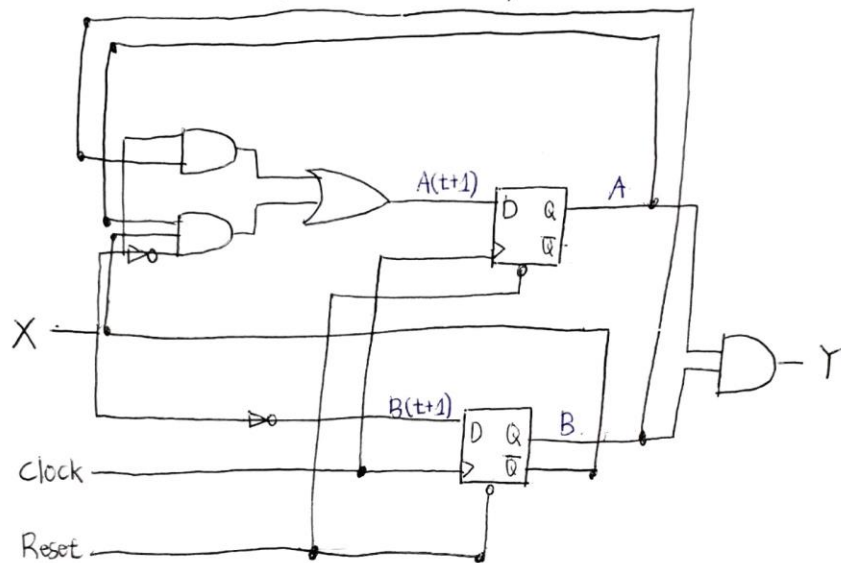


## #4-2 Design circuit

### SSL analysis & Design

State-equation을 통해서 두개의 DFF와  
Combinational logic들을 이용한  
회로도를 그려볼 수 있었습니다.

# Design of synchronous sequential circuits.







## #4-2 Timing-sequence/diagram

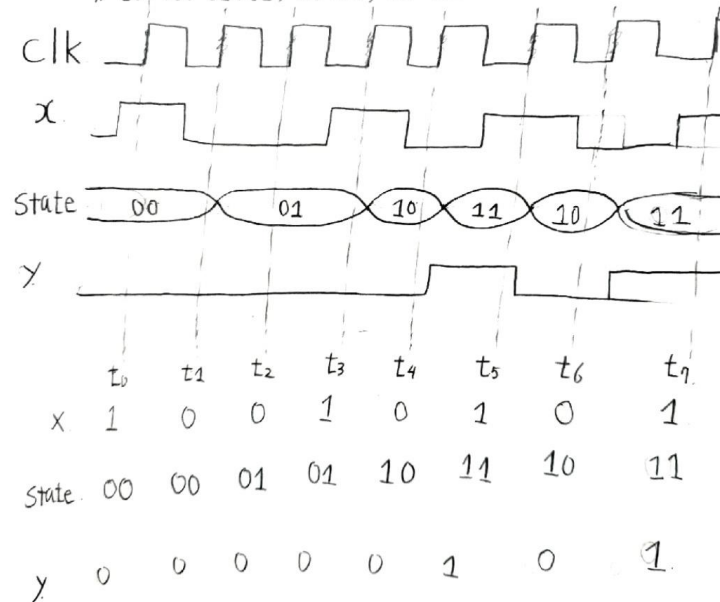
SSL analysis & Design

앞과 마찬가지로 Quartus를 통하여 구현을 하기 전 예상되는 timing-sequence와 timing-diagram을 구현하였습니다.

다음 timing-sequence와 diagram을 보면 clock의 rising edge 부분의 x의 값에 따라서 Next-state의 값이 변할 것입니다.

#2 timing sequence & timing diag

# S<sub>0</sub>: 00, S<sub>1</sub>: 01, S<sub>2</sub>: 10, S<sub>3</sub>: 11로 정의한다.





## #4-2 Timing-sequence/diagram

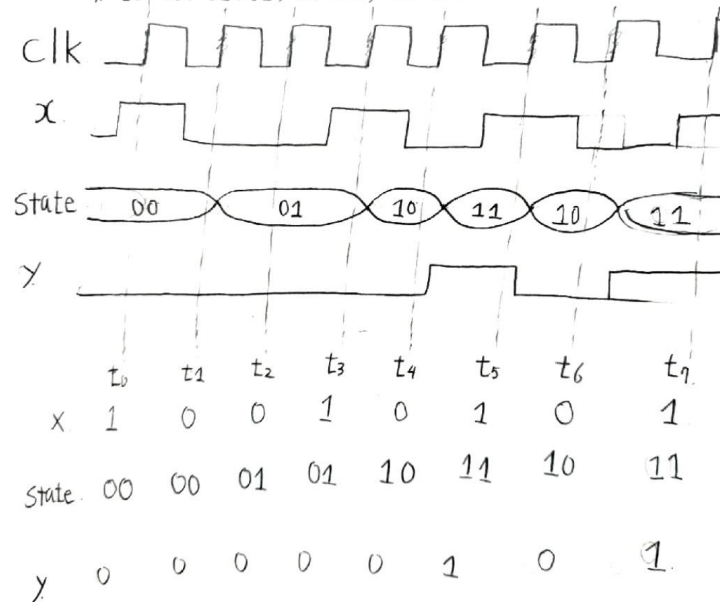
### SSL analysis & Design

이제 시뮬레이팅을 통해서 앞에서도 검증했던  
다음 사항들을 검증하겠습니다.

1. Clock의 rising edge에서의 State-diagram대로의 Next-State의 변화 검증
2. 전파 지연으로 인한 State와 output y의 즉각적인 변화가 일어나지 않음.
3. Reset의 값에 따른 State의 값의 변화

#2 timing sequence & timing diag

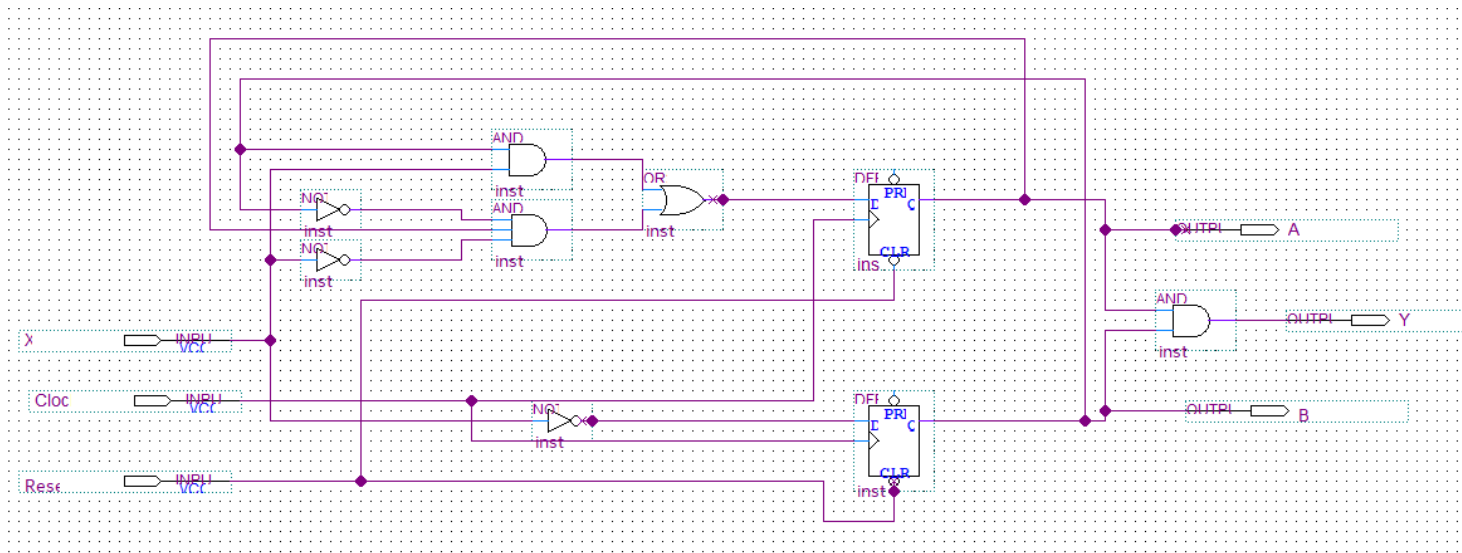
# S<sub>0</sub>: 00, S<sub>1</sub>: 01, S<sub>2</sub>: 10, S<sub>3</sub>: 11로 정의한다.





## #4-2 Design circuit

SSL analysis & Design

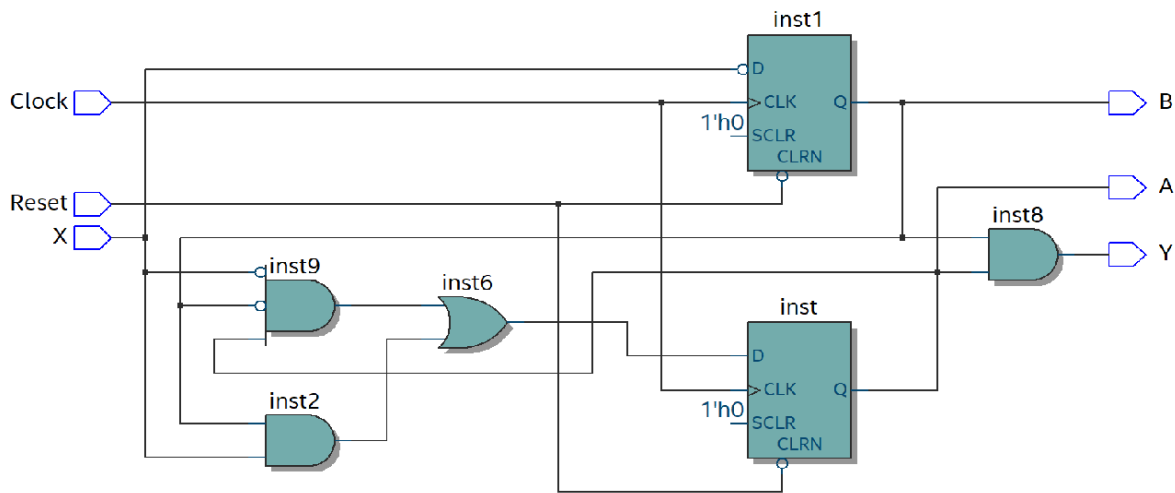


앞에서 그렸던 회로도를 Quartus 상에 구현하였습니다.



## #4-2 RTL viewer

SSL analysis & Design

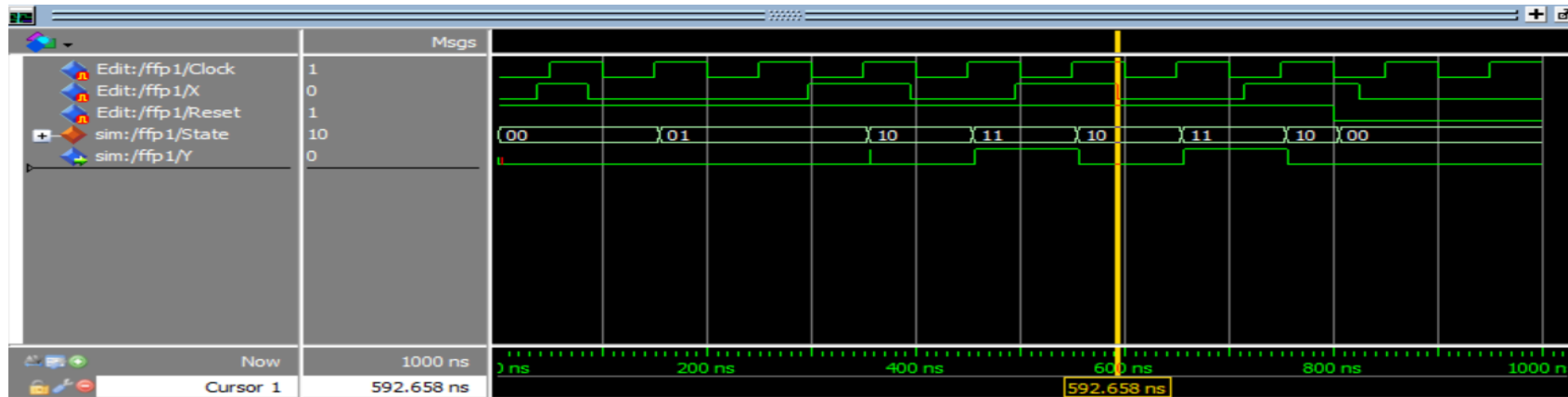


RTL viewer를 이용하여 구현하려는 회로와 디자인한 회로가 같다는 것을 확인하였습니다.



## #4-2 Simulation

### SSL analysis & Design



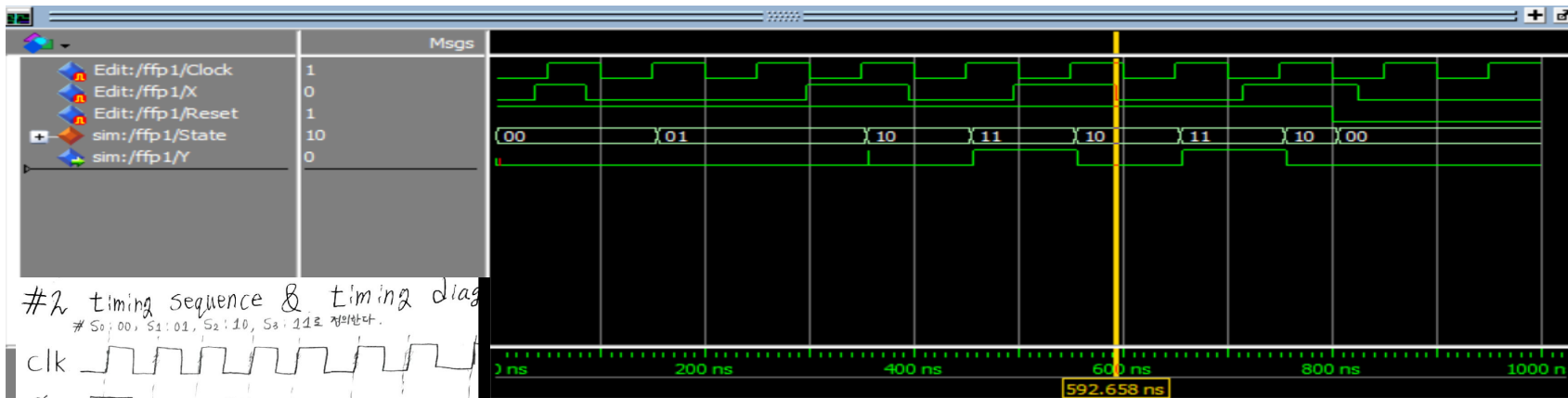
Clock의 rising edge에서의 State의 변화를 검증해 보겠습니다. 우선 시뮬레이션 결과를 Timing-sequence로 나타내 보겠습니다. s0,s1,s2,s3(=00,01,10,11)

	t0	t1	t2	t3	t4	t5	t6	t7
x	1	0	0	1	0	1	0	1
state	00	00	01	01	10	11	10	11
y	0	0	0	0	0	1	0	1

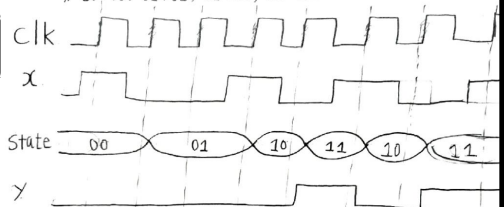


## #4-2 Simulation

### SSL analysis & Design



#2 timing sequence & timing diag  
# S<sub>0</sub>: 00, S<sub>1</sub>: 01, S<sub>2</sub>: 10, S<sub>3</sub>: 11로 정의한다.



	t <sub>0</sub>	t <sub>1</sub>	t <sub>2</sub>	t <sub>3</sub>	t <sub>4</sub>	t <sub>5</sub>	t <sub>6</sub>	t <sub>7</sub>
x	1	0	0	1	0	1	0	1
State	00	00	01	01	10	11	10	11
y	0	0	0	0	0	1	0	1

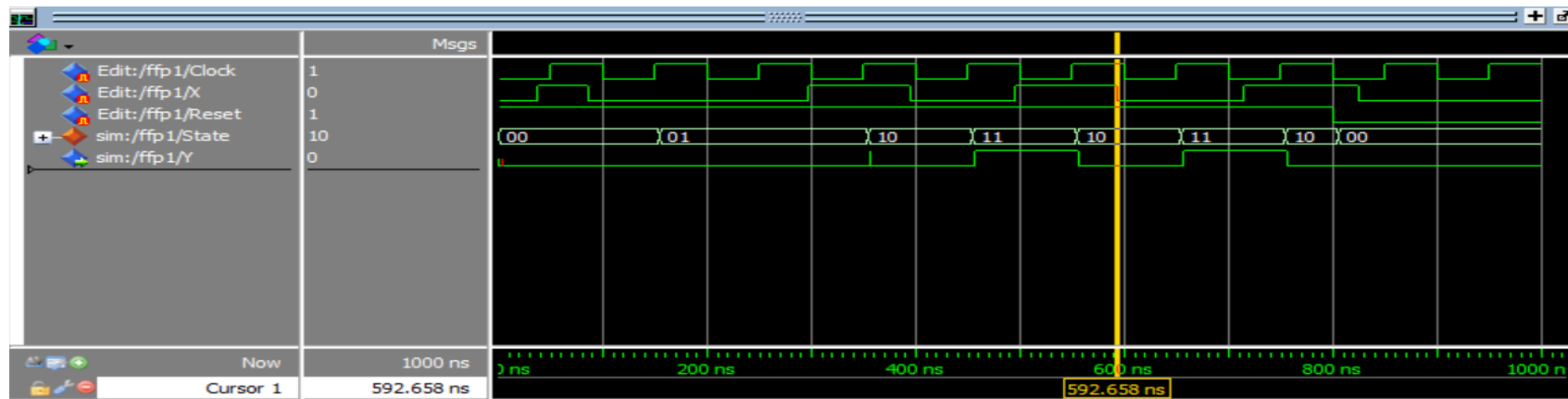
시뮬레이팅을 통하여 도출한 Timing Diagram 역시 예상하였던 Timing sequence, 그리고 diagram과 동일함을 알 수 있었습니다.

예를 들어 t<sub>3</sub>에서의 rising edge에서의 x값이 1이므로 t<sub>3</sub>의 Next-state인 t<sub>4</sub>의 state가 01에서 10로 바뀌었습니다. 따라서 앞의 state-diagram과 마찬가지로 rising edge에서의 Input의 값에 따라서 Next-state가 변한다는 것을 검증할 수 있었습니다.



## #4-2 Simulation

### SSL analysis & Design

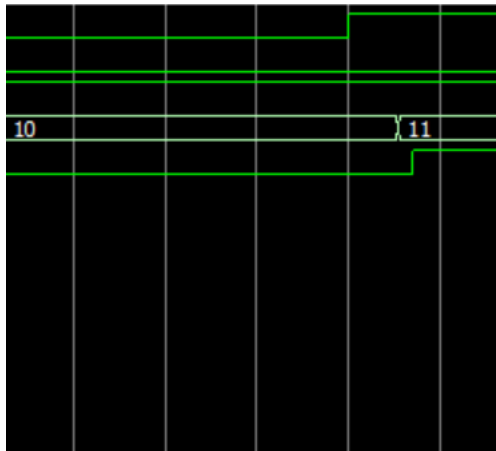


또한 알 수 있는 점은 Reset이 0일 때 rising edge에서의 값의 변화에도 불구하고 00으로 초기화 되는 사실을 알 수 있습니다. 따라서 Reset의 기능에 대해서 검증하였습니다.



## #4-2 Simulation

SSL analysis & Design



다음과 같이 시뮬레이션을 확대했을 때 clock의 rising edge에서 바로 현재 상태와 output의 값이 즉각적으로 바뀌지 않는 것을 알 수 있습니다.  
따라서 전파 지연이 일어난다는 사실을 검증하였습니다.





## #4 Discussion

### SSL analysis & Design

이번에는 Sequential logic을 구현하는 실습을 하였습니다. Moore Machine 회로를 이론으로 배웠던 내용들을 실제로 구현하는 과정을 통해서 회로를 구현할 때 쓰이는 도구를 사용하는 법을 알았습니다.

1. State-diagram을 통해서 Input값에 따른 상태의 변화도를 묘사할 수 있다.
2. State-table을 통해서 상태 변화의 경우의 수를 한 눈에 정리할 수 있다.
3. State-table을 통해 State-equation을 도출하고 이를 통해 회로도를 그려낼 수 있다.
4. Timing-sequence와 Timing-diagram을 통해서 상태의 변화를 유추해 낼 수 있다.

다음은 바탕으로 직접 시뮬레이팅을 하면서 다음과 같은 사실을 확인할 수 있었습니다.

1. Clock의 rising edge에서의 input값에 따라서 next-state의 상태 변한다.
2. 전파 지연이 일어나 state가 바로 변하는 것이 아닌 next-state에서의 상태가 바뀐다.
3. Reset은 초기 상태로 초기화 시킨다.

이번 실습을 통해서 sequential logic을 실제로 이용하고 구현하는 방법을 알 수 있었으며 메모리의 상태를 묘사할 때에 이 회로가 유용하게 쓰일 수 있음을 확인하였습니다. 그리고 sequential logic의 특징에 대해서 정확하게 이해할 수 있었습니다.