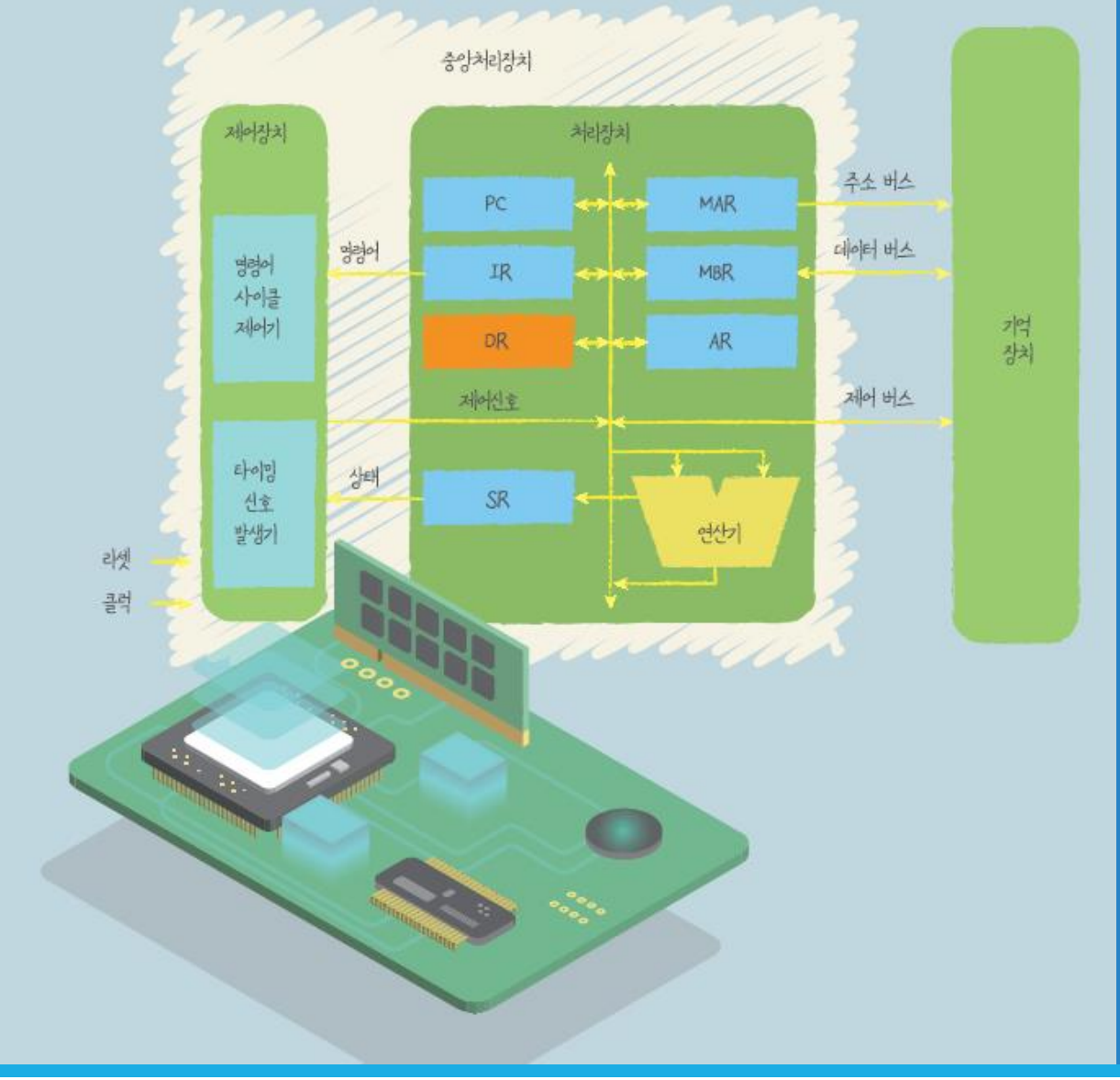
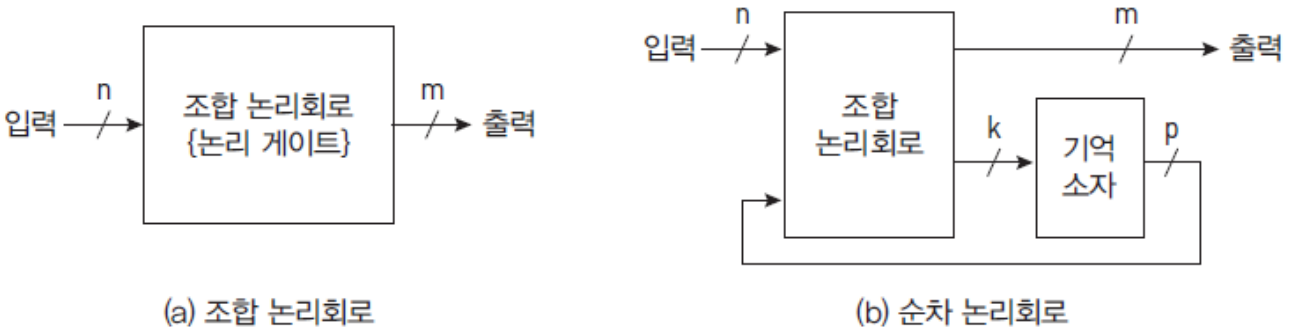


Sequential Logic System

From Prof. Joongnam Jeon's lecture slides

Sequential Logic System



[그림 6-1] 조합 논리회로와 순차 논리회로

Combinatorial Logic

- output = f(input)

Sequential logic

- output, state = f(input, state)

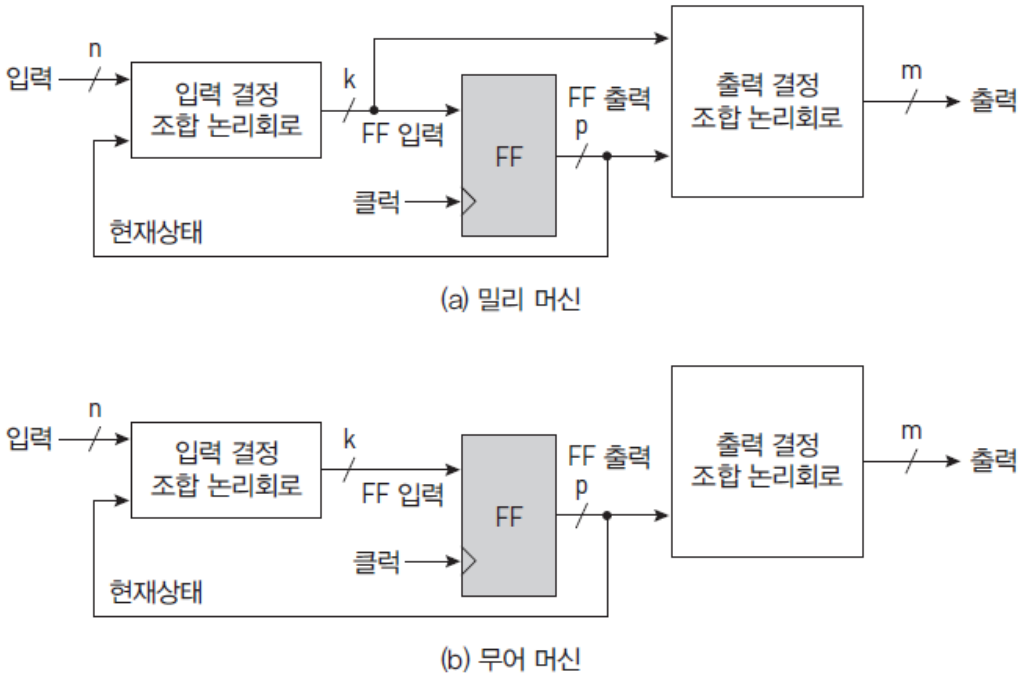
Synchronous Sequential Logic

Synchronous and Asynchronous Digital System

- Synchronous: all flip-flops share the same clock pulse
- Asynchronous: flip-flops do not share the same clock pulse

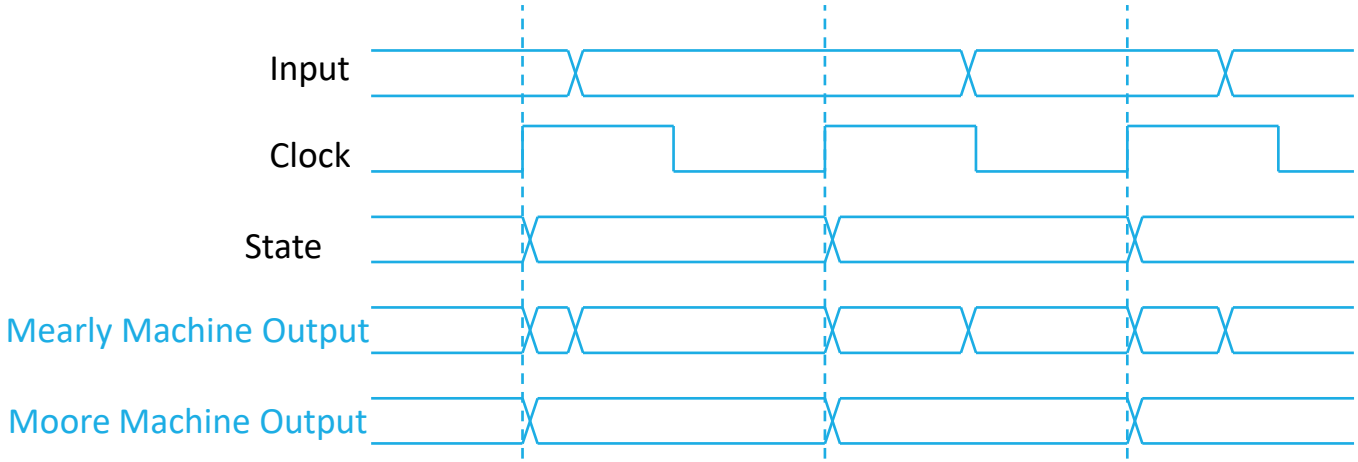
Synchronous Digital System

- **Mealy machine**
  - output = f(input, state)
  - state = f(input, state)
- **Moore machine**
  - output = f(state)
  - state = f(input, state)

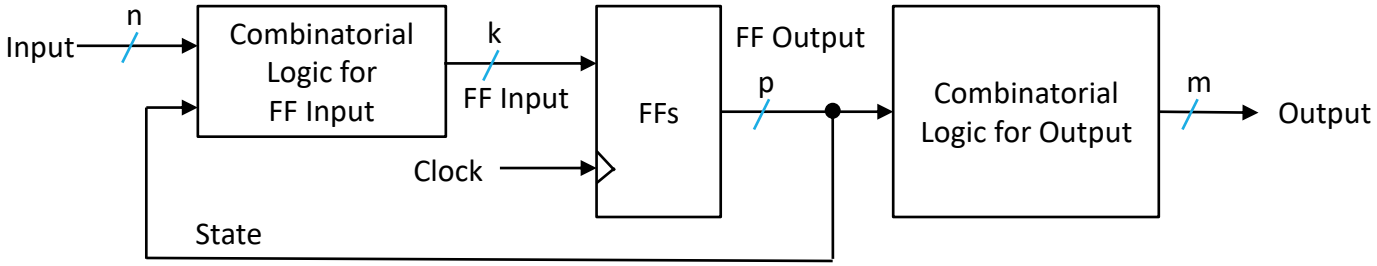


Temporal Behaviors

- Mealy machine
  - output = f(input, state)
  - state = f(input, state)
- Moore machine
  - output = f(state)
  - state = f(input, state)
  - most digital systems are designed as Moore machines

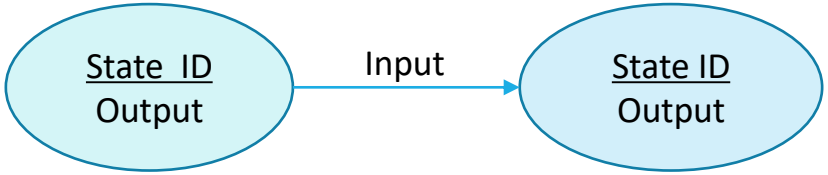


Representing Moore Machine



State diagram

상태 수 = 2^p

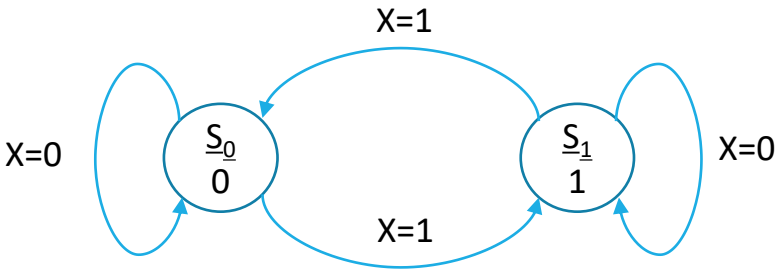


State table

$2^{(n+p)}$

State	Input	State'	Output
...	...	...	...

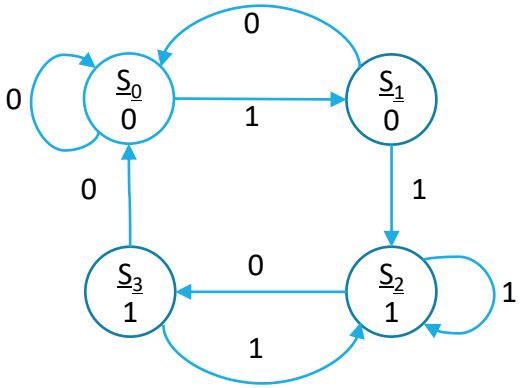
Example

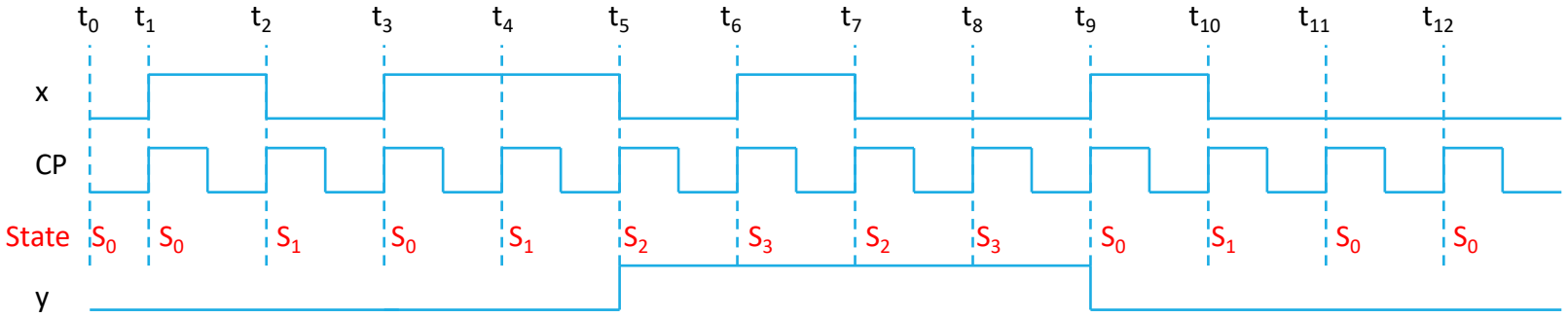


State	Input (X)	State*	Output (Y)
S_0	0	S_0	0
S_0	1	S_1	1
S_1	0	S_1	1
S_1	1	S_0	0

2-Input Sequence Detector

상태/출력	조건	비고
$S_0/0$	출력 1일 때 0이 연속해서 두번 입력된 상태	초기상태
$S_1/0$	출력 0일 때 1이 한번 입력된 상태	
$S_2/1$	출력 0일 때 1이 연속해서 두 번 입력된 상태	
$S_3/1$	출력 1일 때 0이 한번 입력된 상태	



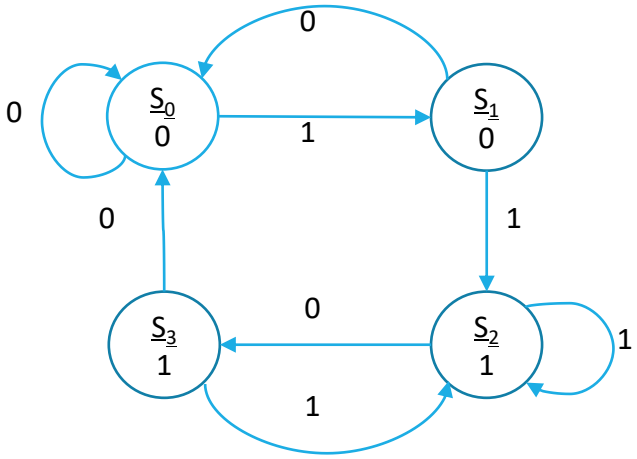


< 2-입력 검출기의 동작 예 (상태 표시) >

Sequence Detector

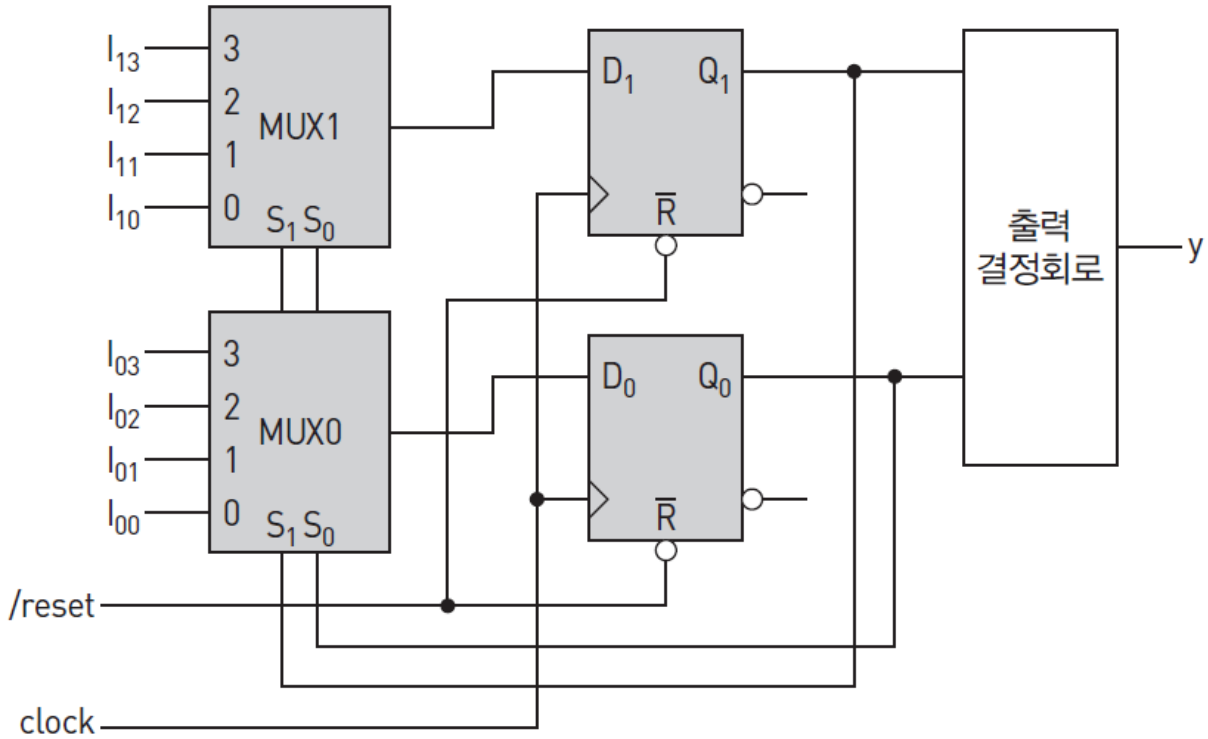
- A sequence detector determines whether a specific sequence of input is given
- Each state represents the [illegible]equence pattern
- Example. 2-Input Detecto[illegible]
  - detect whether the same [illegible]n a row

Current State		Output
State	Q_1 Q_0	y
S_0	0 0	0
S_1	0 1	0
S_2	1 0	1
S_3	1 1	1



Current State		Input	Next State	
State	Q_1 Q_0	x	State	Q_1 Q_0
S_0	0 0	0	S_0	0 0
S_0	0 0	1	S_1	0 1
S_1	0 1	0	S_0	0 0
S_1	0 1	1	S_2	1 0
S_2	1 0	0	S_3	1 1
S_2	1 0	1	S_2	1 0
S_3	1 1	0	S_0	0 0
S_3	1 1	1	S_2	1 0

Implementation with MUX

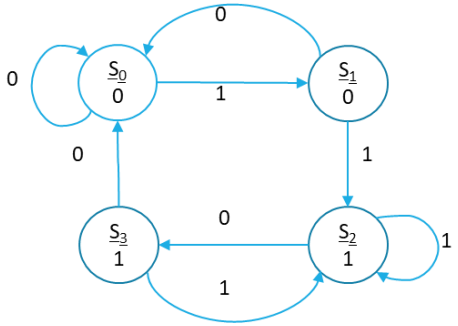


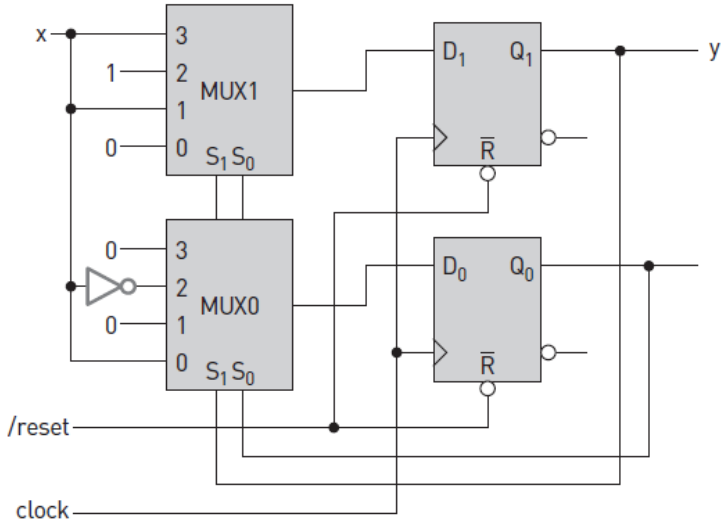
[그림 6-9] 멀티플렉서 구현의 일반 회로

Implementation with MUX

Q_1 Q_0	x	Q'_1	MUX1 In
0 0	0	0	0
0 0	1	0	
0 1	0	0	x
0 1	1	1	
1 0	0	1	1
1 0	1	1	
1 1	0	0	x
1 1	1	1	

Q_1 Q_0	x	Q^*_0	MUX0 In
0 0	0	0	x
0 0	1	1	
0 1	0	0	0
0 1	1	0	
1 0	0	1	x'
1 0	1	0	
1 1	0	0	0
1 1	1	0	



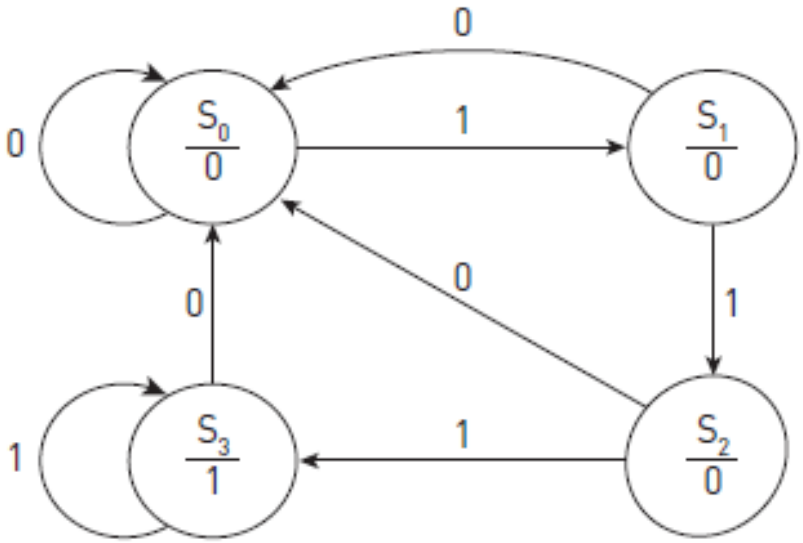


[그림 6-10] 2-입력 검출기의 논리회로도

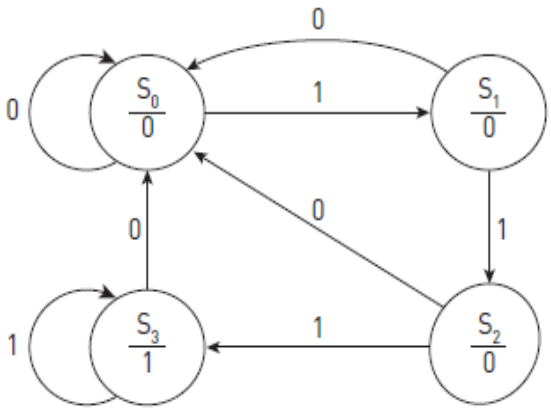
Example. 1-1-1 Detector

The output is one-bit indicating whether the last three input values were all one’s.

State Diagram



Truth Table



Current State		Input	Next State	
State	Q_1 Q_0	x	State	Q^*_1 Q^*_0
S_0	0 0	0		
S_0	0 0	1		
S_1	0 1	0		
S_1	0 1	1		
S_2	1 0	0		
S_2	1 0	1		
S_3	1 1	0		
S_3	1 1	1		

Next State		Output
State	Q^*_1 Q^*_0	y

Combinatorial Logic Implementation with Two D Flip-flops

Current State		Input	Next State	
State	Q_1 Q_0	x	State	Q^*_1 Q^*_0
S_0	0 0	0		
S_0	0 0	1		
S_1	0 1	0		
S_1	0 1	1		
S_2	1 0	0		
S_2	1 0	1		
S_3	1 1	0		
S_3	1 1	1		

Next State		Output
State	Q^*_1 Q^*_0	y

MUX Implementation

현재상태 (Q_1 Q_0)	입력 (x)	다음상태 (Q^*_1)	MUX1 출력
S_0	0	0	
S_0	1	0	
S_1	0	0	
S_1	1	1	
S_2	0	0	
S_2	1	1	
S_3	0	0	
S_3	1	1	

현재상태 (Q_1 Q_0)	입력 (x)	다음상태 (Q^*_0)	MUX0 출력
S_0	0	0	
S_0	1	1	
S_1	0	0	
S_1	1	0	
S_2	0	0	
S_2	1	1	
S_3	0	0	
S_3	1	1	

MUX Implementation