Introduction to Sequential Circuits

Models of Digital Circuits

Combinational Logic :

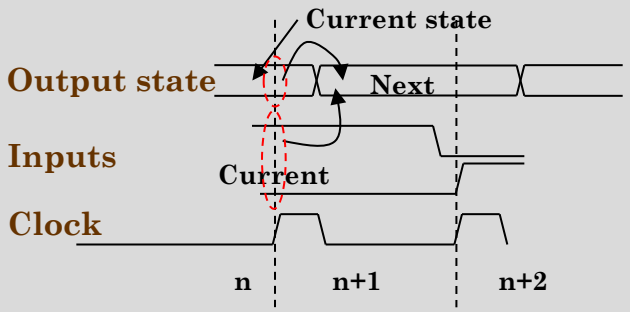
Output states of Combinational Logic depends only on the current states of input variables.

Sequential Logic :

Next Output state (n+1) of Sequential Logic depends on the current state of input variables and current output state (n).

We need a Memory element for old (previous) state.

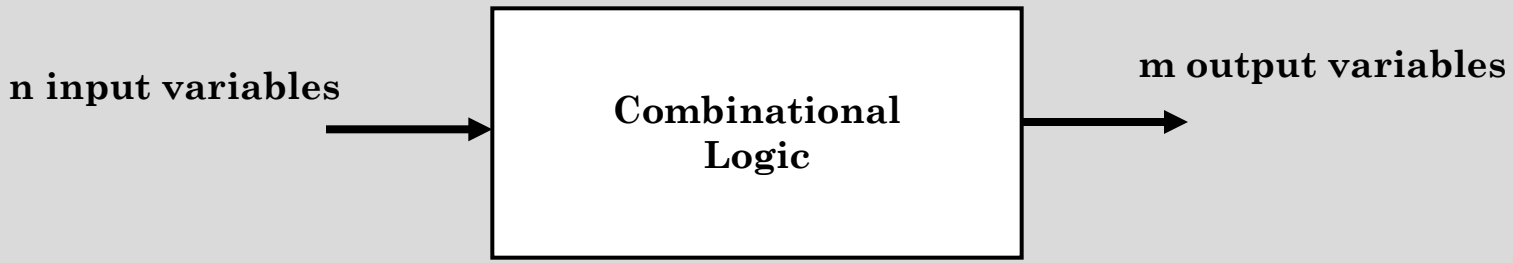
Latch
Flip-Flop
Delay



If memory element is clocked, then circuit is synchrous, if not, circuit is asynchronous.

Introduction to Sequential Circuits

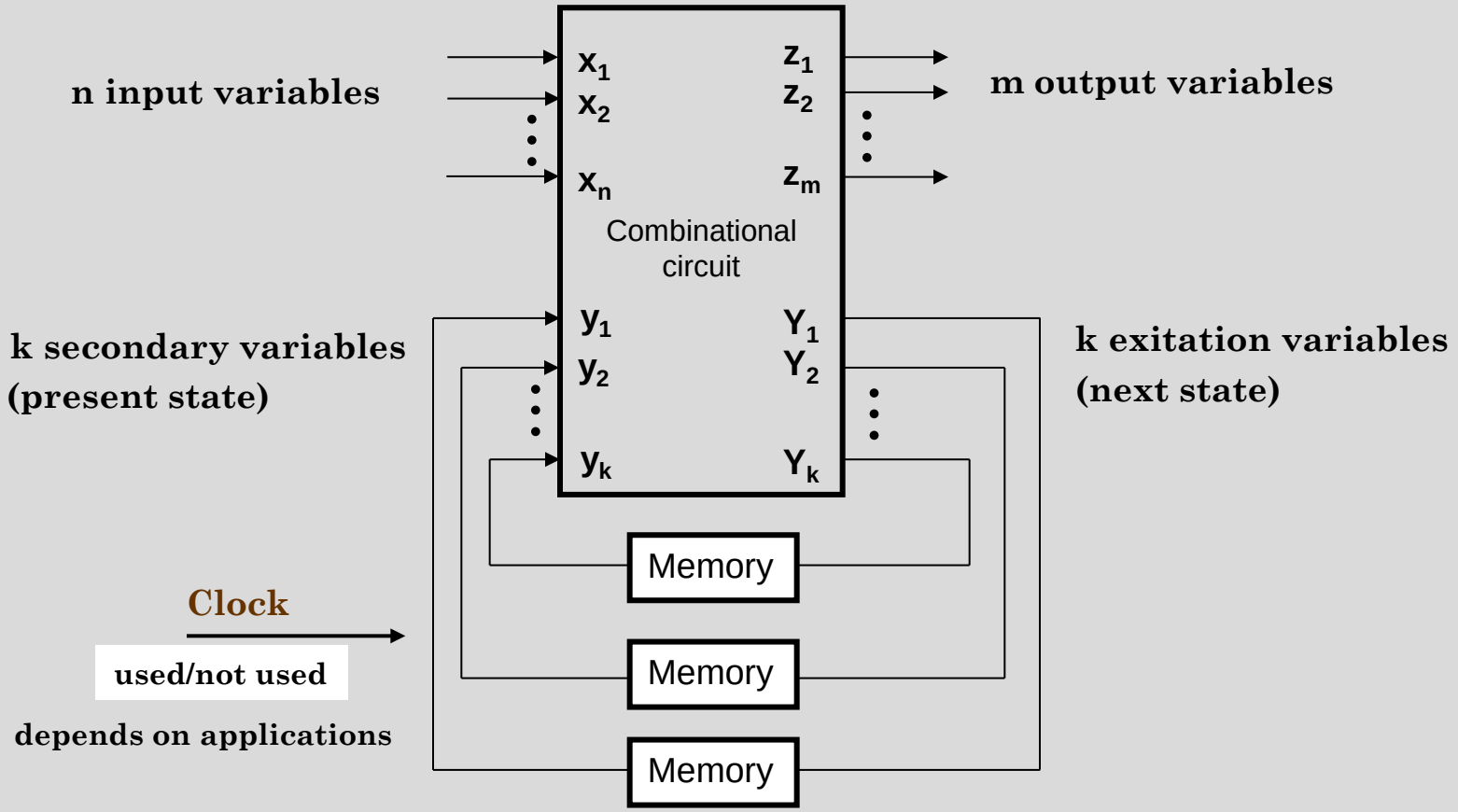
Models of Digital Circuits



Introduction to Sequential Circuits

Models of Digital Circuits

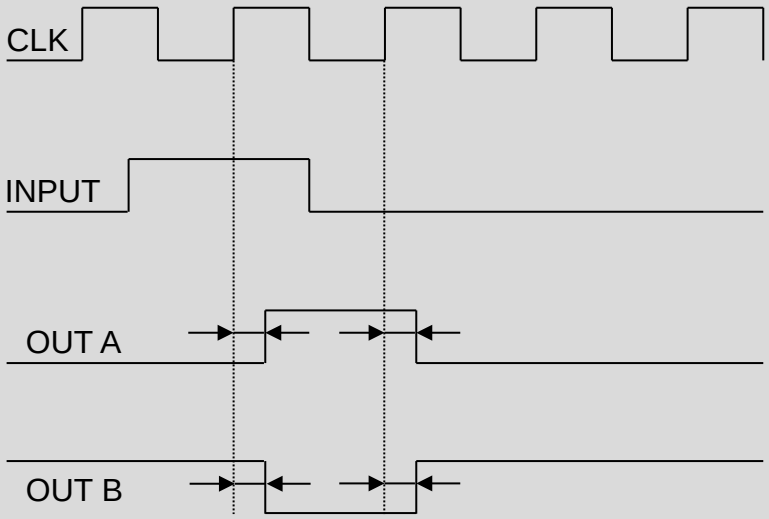
Model for sequential logic



Introduction to Sequential Circuits

Models of Digital Circuits

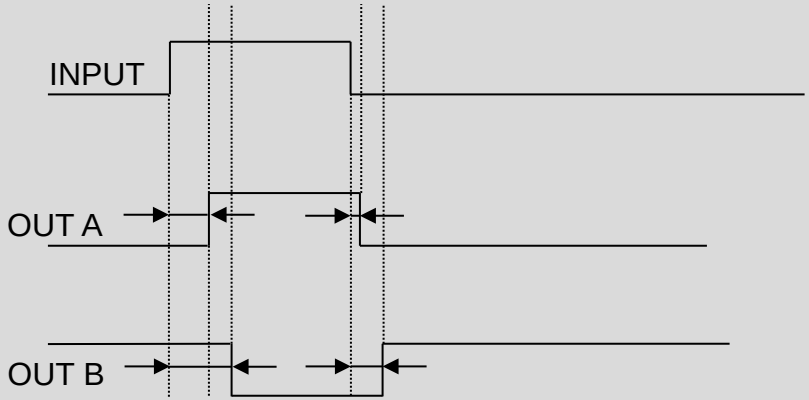
Synchronous sequential logic



The change of internal state occurs in response to the clock pulses.

Memory element : clocked, (gated)

Asynchronous sequential logic



The change of internal state occurs when there is change in the input variables.

Memory element : unclocked or time-delay elements.

Introduction to Sequential Circuits

Synchronous or asynchronous ?

Synchronous sequential logic

Properly designed system

↓

"No timing problems"

Asynchronous sequential logic

No clocked flip-flops Feedback bath

↓

Timing problems

↓

The design of asynchronous sequential circuits is difficult

Used when speed of operation is important

Fast response to the change of input variable

No clock distribution

↓

Redused logic and power dissipation

Introduction to Sequential Circuits

Models of Digital Circuits

Wery Important for Asynchronous Sequential Logic.

To ensure proper operation, circuits must be allowed to attain a stable state before the input is changed to a new value.

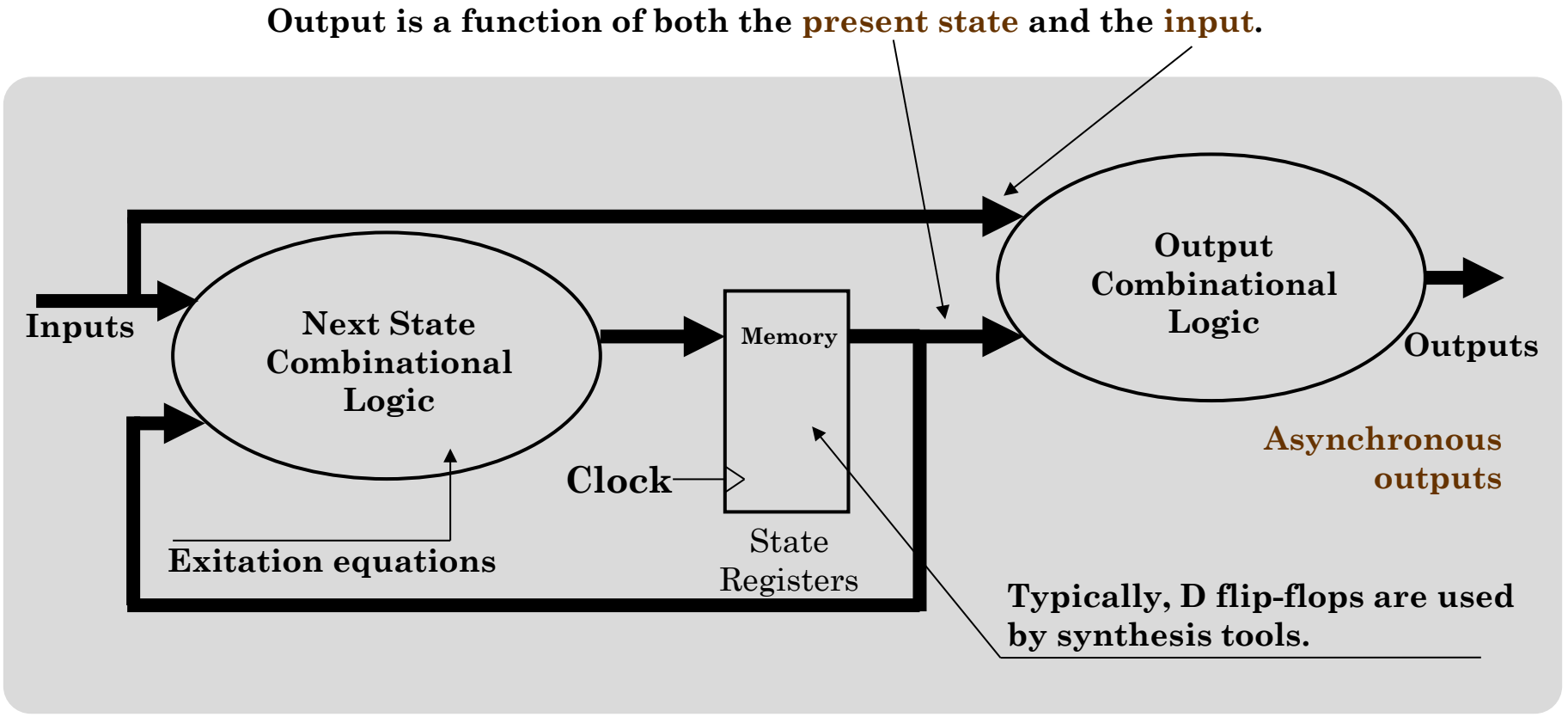
Because of delays in the wires and the gate circuits it is "impossible" to say which variable changes its state first when two or more input variables change at "exactly same instant time".

Fundamental mode

Input signals change one at a time and only when the circuit is in the stable state.

In this course material we design and analyze only **synchronous** sequential logic.

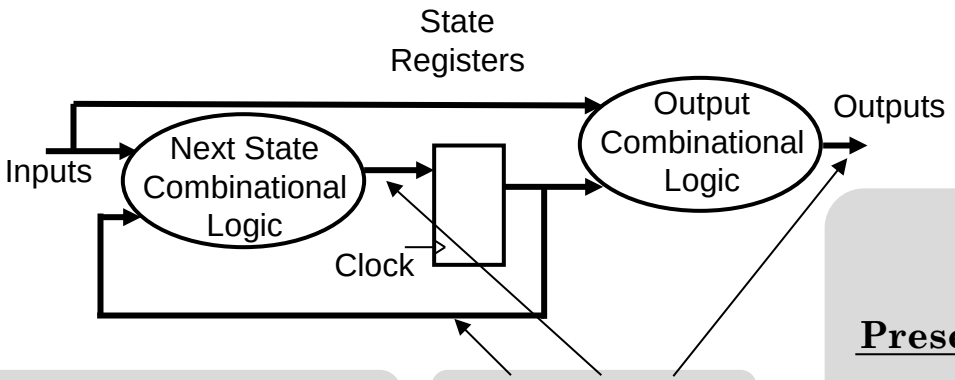
Mealy Type state Machine



Block diagram of Mealy type state machine

Introduction to Sequential Circuits

Mealy Type state Machine



Present states
A(t) B(t)

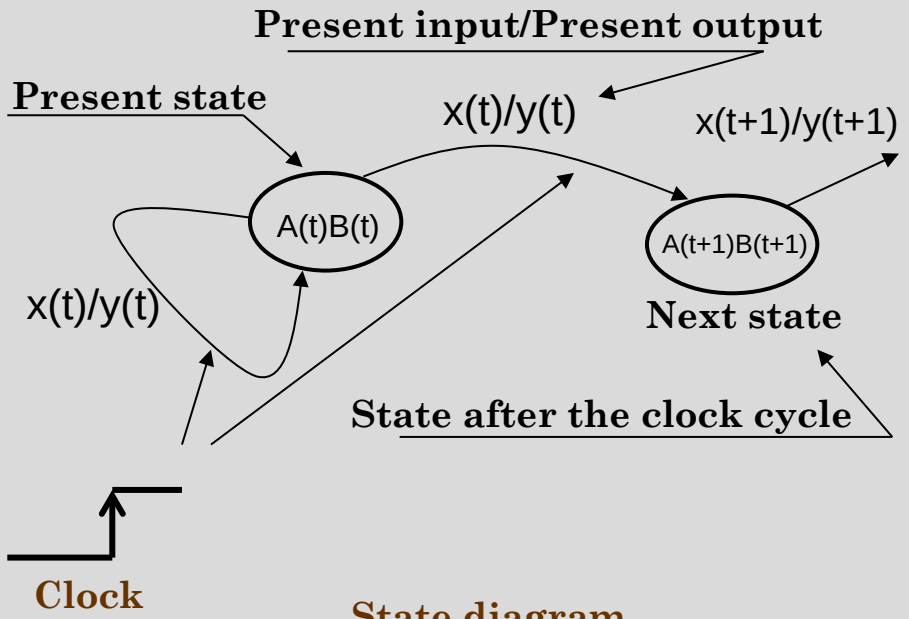
Present input
x(t)

Next states
A(t+1) B(t+1)

Present output
y(t)

t	t+1	t
ABx	AB	y
000	00	0
001	01	0
010	00	1
011	11	0
100	00	1
101	10	0
110	00	1
111	10	0

State table of Mealy FSM



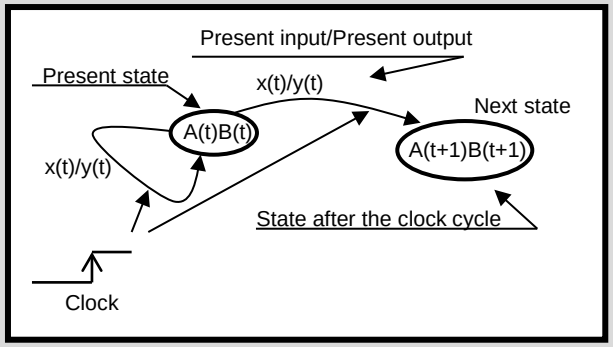
State diagram

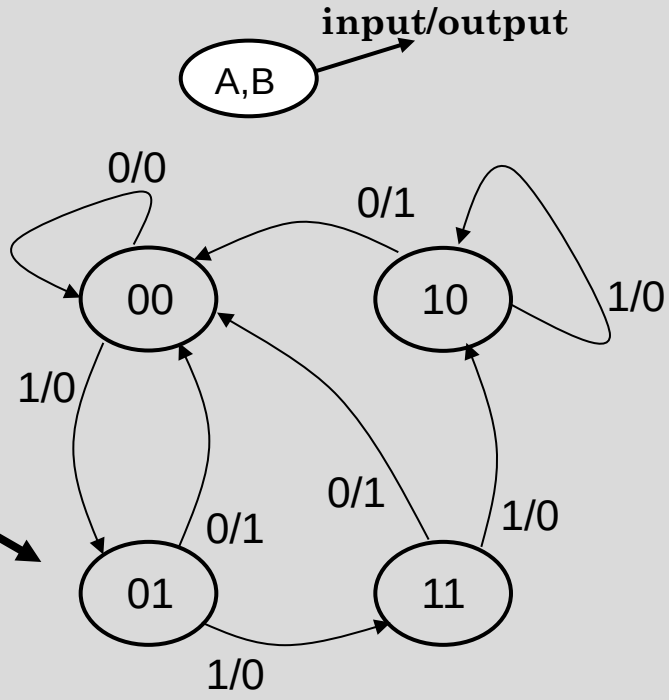
Mealy Type state Machine

Example :

t	t+1	t
ABx	AB	y
000	00	0
001	01	0
010	00	1
011	11	0
100	00	1
101	10	0
110	00	1
111	10	0

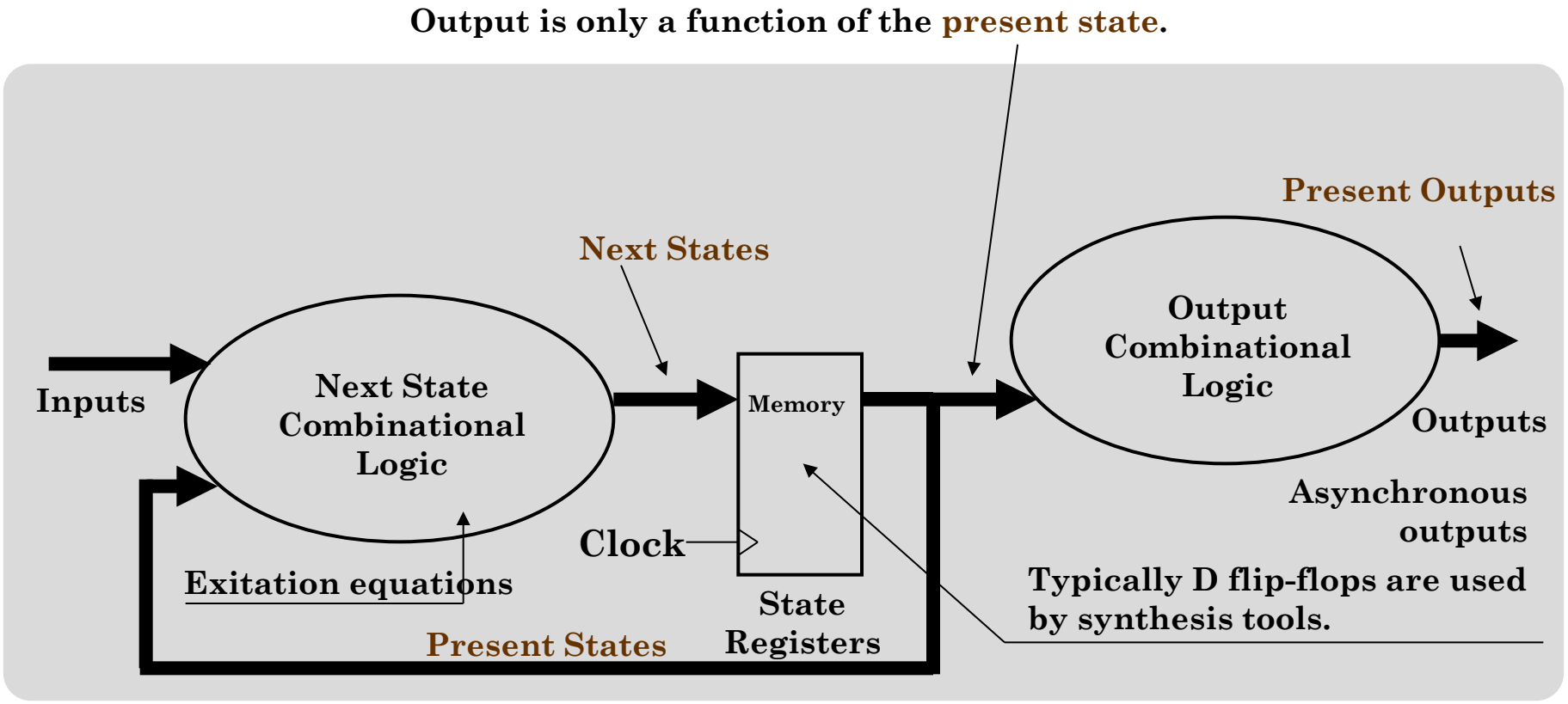
State table of Mealy FSM





State diagram of Mealy FSM

Moore Type state Machine



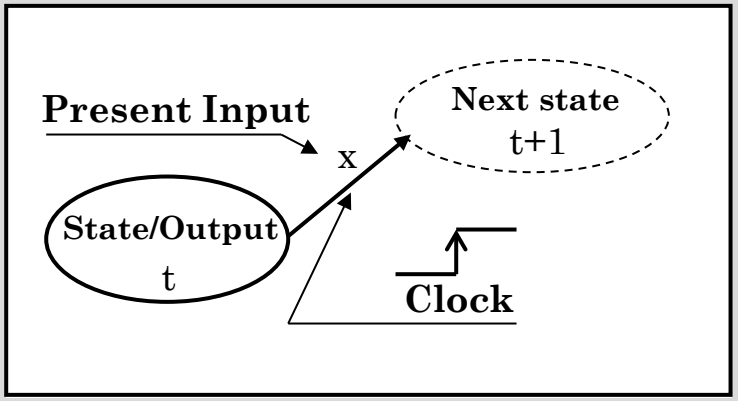
Block diagram of Moore type state machine

Moore Type state Machine

Example :

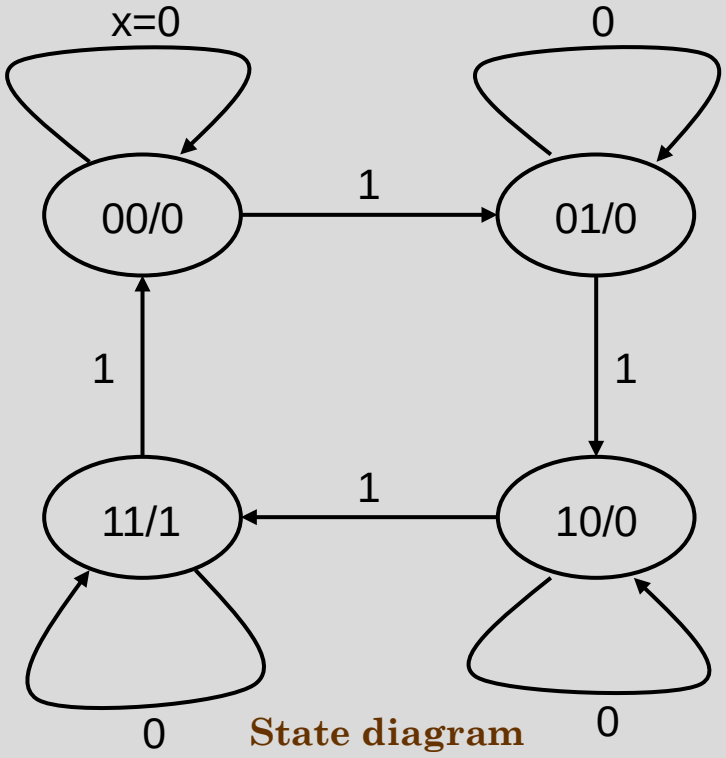
t	t+1	t
ABx	AB	y
000	00	0
001	01	0
010	01	0
011	10	0
100	10	0
101	11	0
110	11	1
111	00	1

State table of Moore FSM



State

AB/y

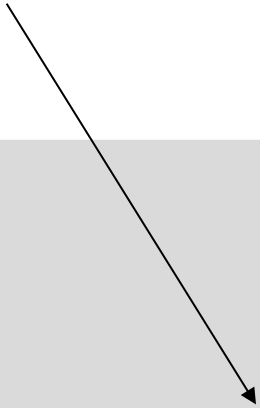


State diagram

Storage elements

Latches

Storage elements that operate with signal levels are referred to as latches.



SR latch with NAND gates

SR latch with NOR gates

SR latch with control input C

D latch with control input C (Transparent latch)

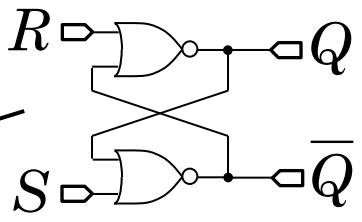
Introduction to Sequential Circuits

Storage elements

S 0 → 0 R 0 → 1		Allowed input transitions in fundamental mode
S 0 → 1 R 0 → 0		
S 0 → 1 R 0 → 1	Forbidden	$Q_{n+1} = \overline{Q}_{n+1} = 0$
S 0 → 0 R 1 → 0		
S 0 → 1 R 1 → 1	Forbidden	$Q_{n+1} = \overline{Q}_{n+1} = 0$
S 0 → 1 R 1 → 0	Forbidden	
S 1 → 0 R 0 → 0		
S 1 → 1 R 0 → 1	Forbidden	$Q_{n+1} = \overline{Q}_{n+1} = 0$
S 1 → 0 R 0 → 1	Forbidden	
S 1 → 0 R 1 → 0	Forbidden	
S 1 → 0 R 1 → 1	Forbidden	$Q_n = 0\ \overline{Q}_n = 0$
S 1 → 1 R 1 → 0		

SR Latch

SR latch with NOR gates

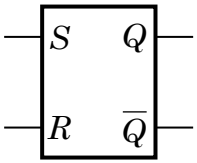


S	R	Q	$\overline{Q}$
0	0	?	?
0	1	0	1
1	0	1	0
1	1	0	0

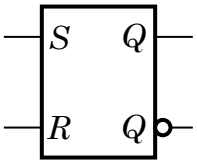
? = 0 or 1 Depends on previous state

"forced" states: reset and set

State table



Symbol



Symbol

S	R	Q_{n+1}	$\overline{Q}_{n+1}$	
0	0	Q_n	$\overline{Q}_n$	
0	1	0	1	reset state
1	0	1	0	set state
1	1	0	0	Forbidden

Function table of SR latch with NOR gates

Storage elements

$\overline{S}\overline{R}$ Latch

$\overline{S}\overline{R}$ latch with NAND gates

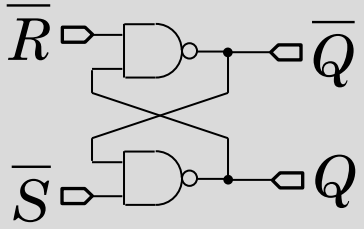
$\overline{S}$	$\overline{R}$	Q	$\overline{Q}$
0	0	1	1
0	1	1	0
1	0	0	1
1	1	?	?

? = 0 or 1

"forced" states: reset and set

Depends on previous state

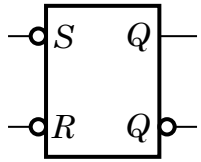
State table



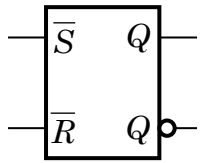
Forbidden

$\overline{S}$	$\overline{R}$	Q_{n+1}	$\overline{Q}_{n+1}$	
0	0	1	1	
0	1	1	0	set state
1	0	0	1	reset state
1	1	Q_n	$\overline{Q}_n$	

Function table of $\overline{S}\overline{R}$ latch with NAND gates



Symbol



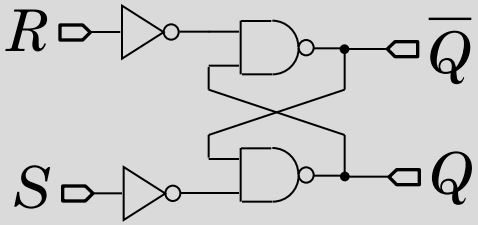
Symbol

Introduction to Sequential Circuits

Storage elements

SR Latch

SR latch with NAND gates



Forbidden →

S	R	Q_{n+1}	$\overline{Q}_{n+1}$
0	0	Q_n	$\overline{Q}_n$
0	1	0	1
1	0	1	0
1	1	1	1

Function table of SR latch with NAND gates

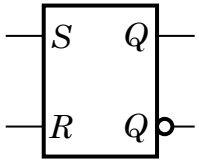
In general, the function table of SR latch

Forbidden →

S	R	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	?

? = 0/1

Depend on latch implementation

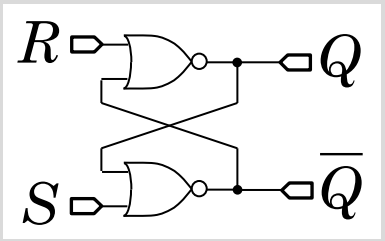


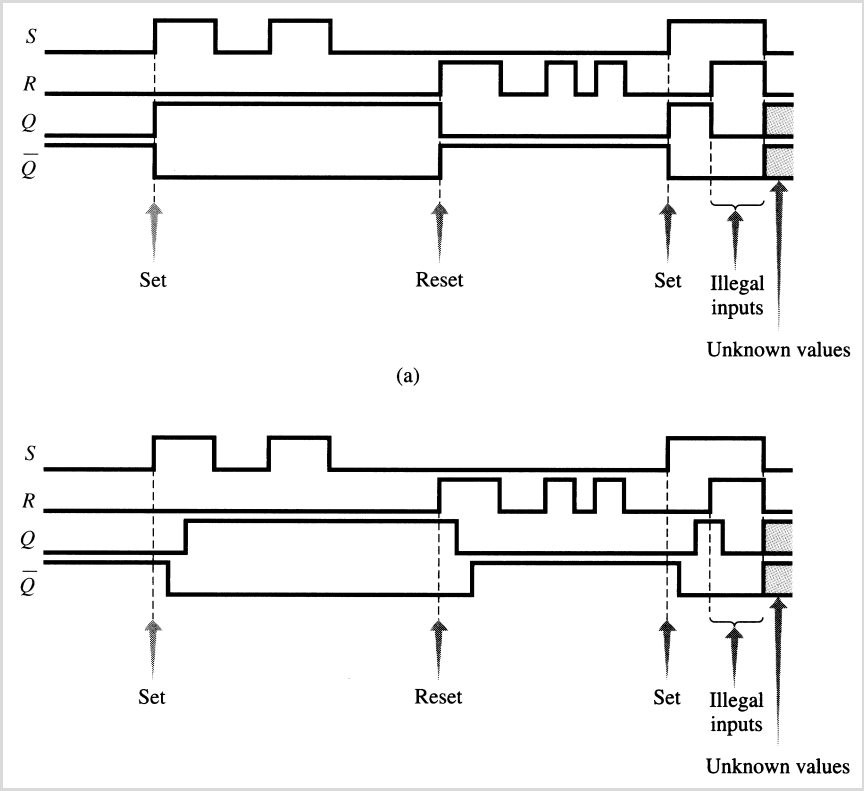
Symbol

Storage elements

SR Latch

Set-reset latch timing diagram





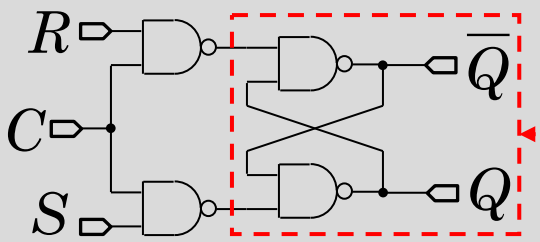
With zero-delay model

With non-zero-delay model

Storage elements

SR Latch

SR latch with Control input

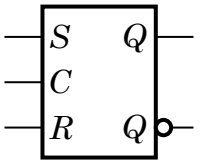


$\overline{S}$	$\overline{R}$	Q_{n+1}	$\overline{Q}_{n+1}$	
0	0	1	1	Forbidden
0	1	1	0	set state
1	0	0	1	reset state
1	1	Q_n	$\overline{Q}_n$	

Function table of $\overline{S}\overline{R}$ latch with NAND gates

C	S	R	Q_{n+1}	$\overline{Q}_{n+1}$	
1	0	0	Q_n	$\overline{Q}_n$	
1	0	1	0	1	reset state
1	1	0	1	0	set state
1	1	1	1	1	Forbidden
0	X	X	Q_n	$\overline{Q}_n$	C=0

Function table of gated SR latch with NAND gates



Symbol

Storage elements

D Latch

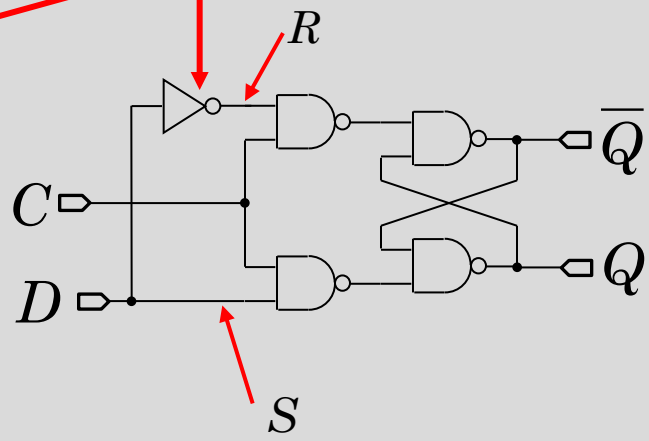
D latch with Control input

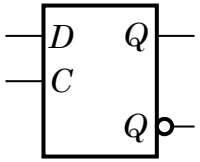
(Transparent latch)

C	S	R	Q_{n+1}	$\overline{Q}_{n+1}$	
1	0	0	Q_n	$\overline{Q}_n$	
1	0	1	0	1	← reset state
1	1	0	1	0	← set state
1	1	1	1	1	← Forbidden
0	X	X	Q_n	$\overline{Q}_n$	

Function table of gated SR latch with NAND gates

Assumption : $R = \overline{S}$





Symbol

Storage elements

D Latch

D latch with Control input

(Transparent latch)

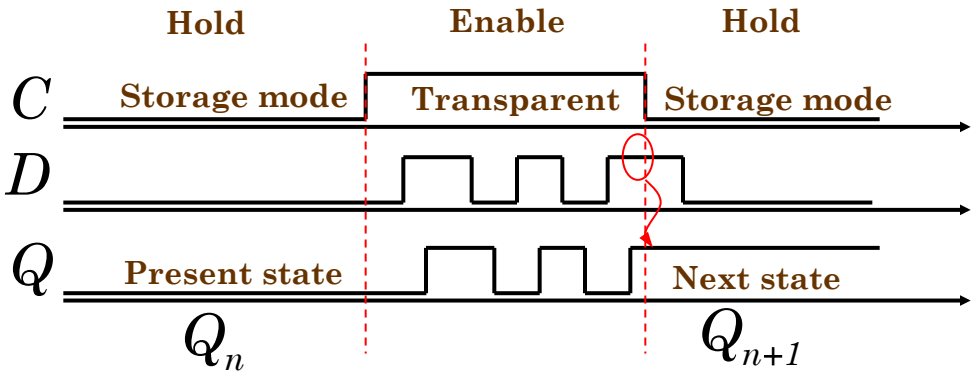
C	S	R	Q_{n+1}	$\overline{Q}_{n+1}$	
1	0	1	0	1	← reset state
1	1	0	1	0	← set state
0	X	X	Q_n	$\overline{Q}_n$	

(D and $\overline{D}$ are connected to S and R)

C	D	Q_{n+1}	
1	0	0	← reset state
1	1	1	← set state
0	X	Q_n	storage state

Function table of gated D latch

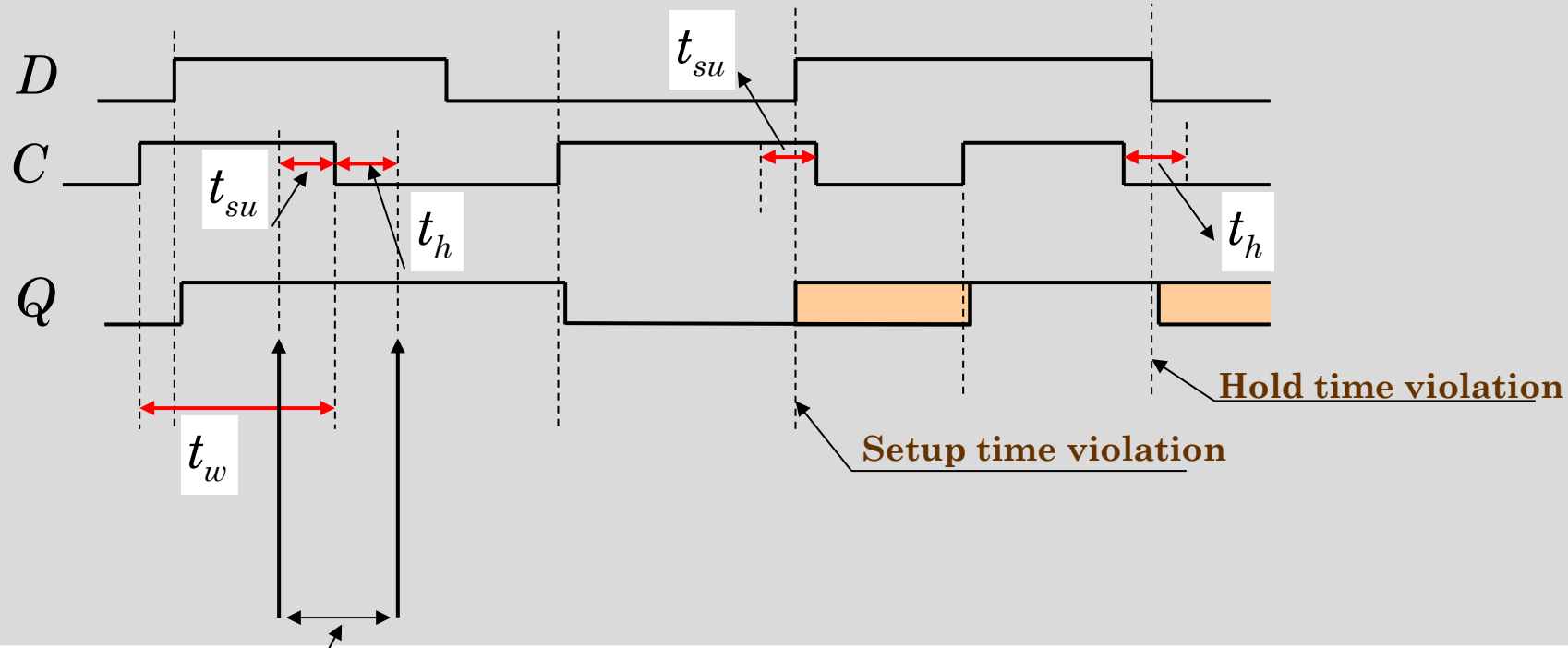
"Transparent mode" when C = 1



Storage elements

D Latch

D Latch Timing Constrains



t_w **Minimum pulse width of the gate signal (clock)**

t_{su} **Setup Time**

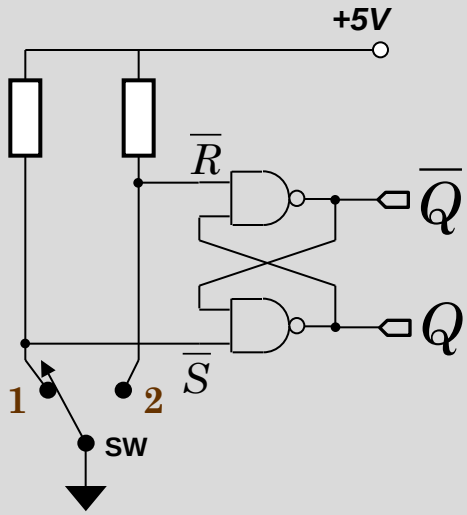
t_h **Hold Time**

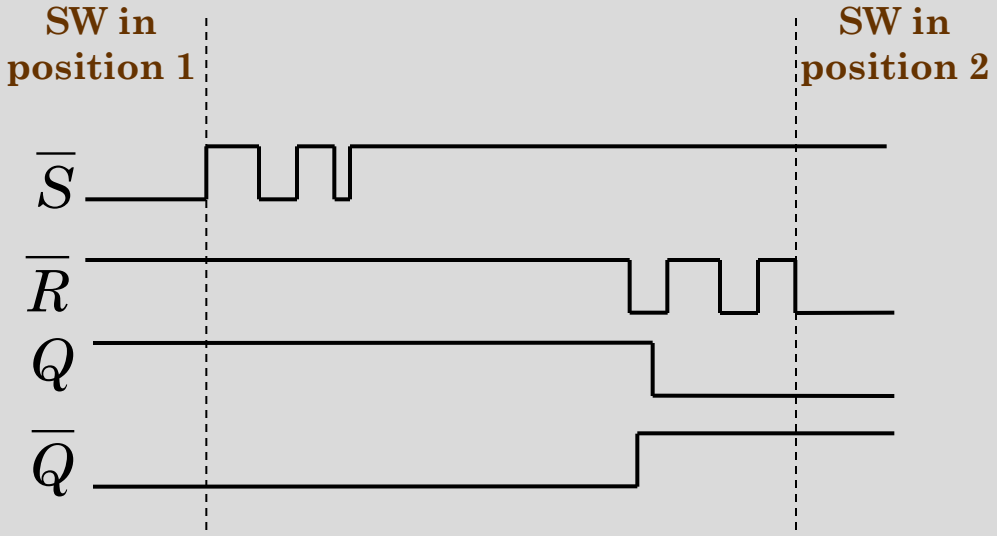
Storage elements

A simple application of $\overline{S}\overline{R}$-latch

$\overline{S}$	$\overline{R}$	Q_{n+1}	$\overline{Q}_{n+1}$	
0	0	1	1	Forbidden
0	1	1	0	set state
1	0	0	1	reset state
1	1	Q_n	$\overline{Q}_n$	

Function table of $\overline{S}\overline{R}$ latch with NAND gates





The End