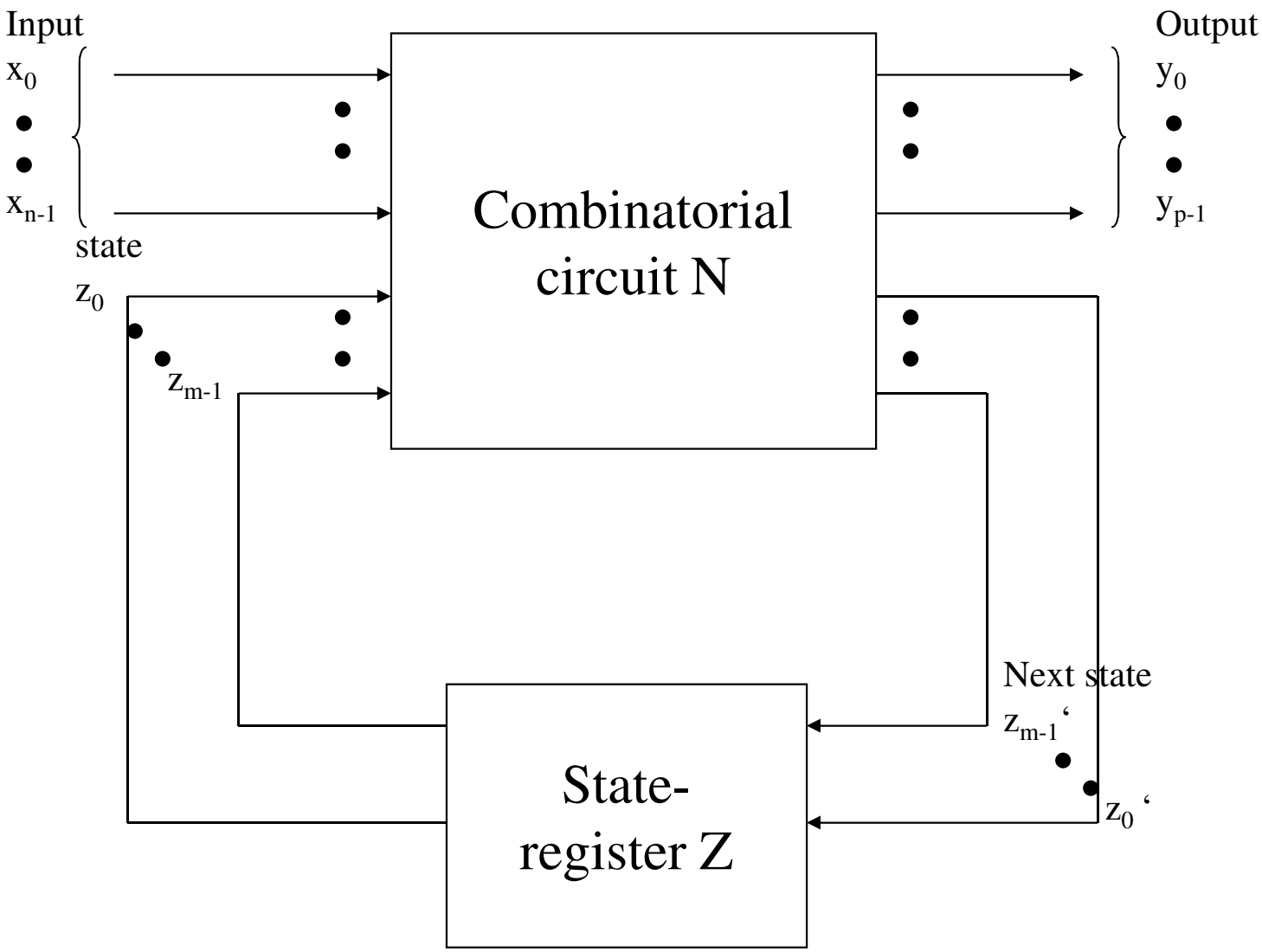
Definition:

A sequential circuit is a circuit whose output depends on the actual input and the actual inner state.


Input
x_0
x_{n-1}
state
z_0
z_{m-1}
Combinatorial circuit N
Output
y_0
y_{p-1}
Next state
z_{m-1}'
z_0'
State-register Z

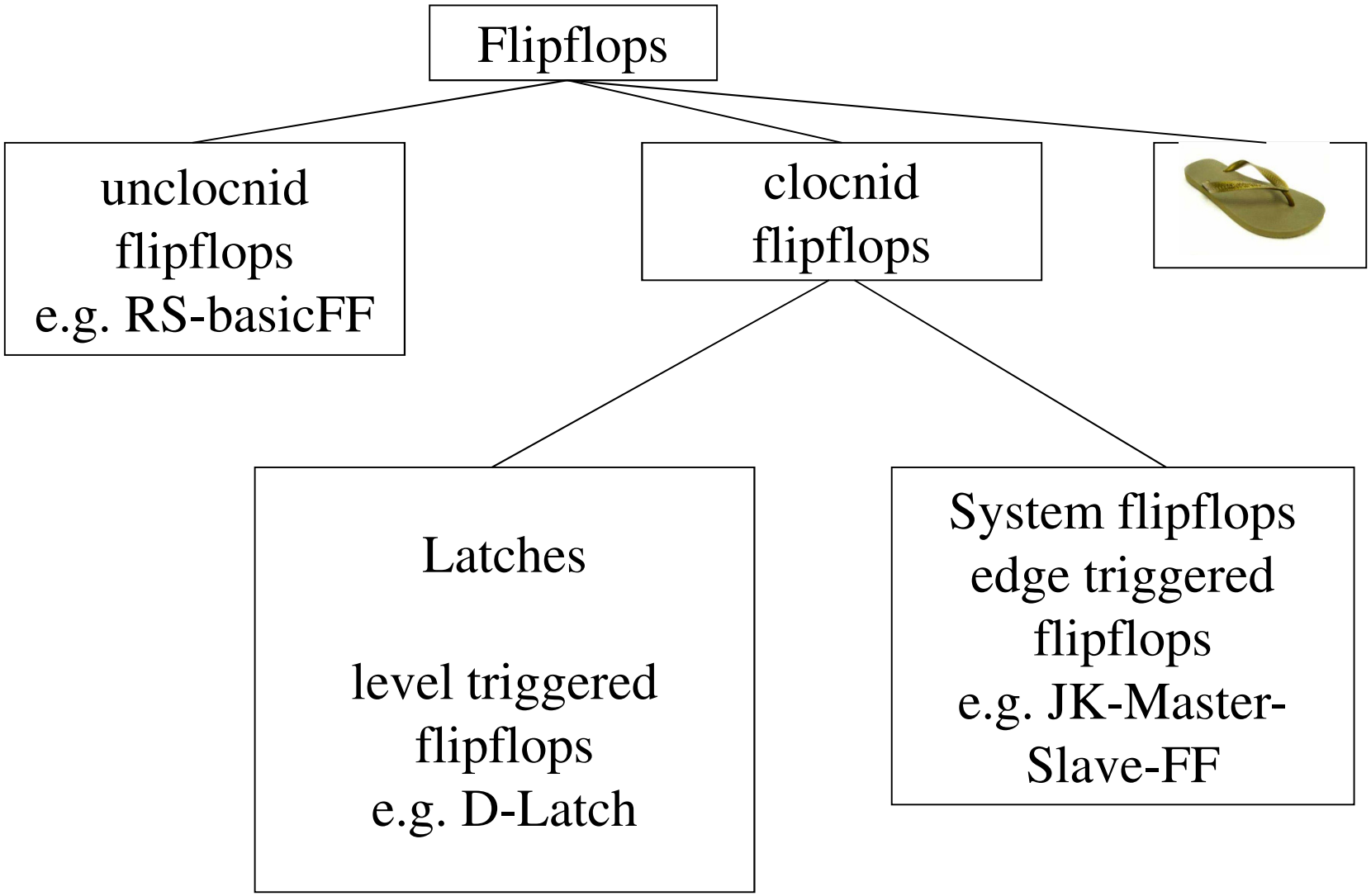
Definition:

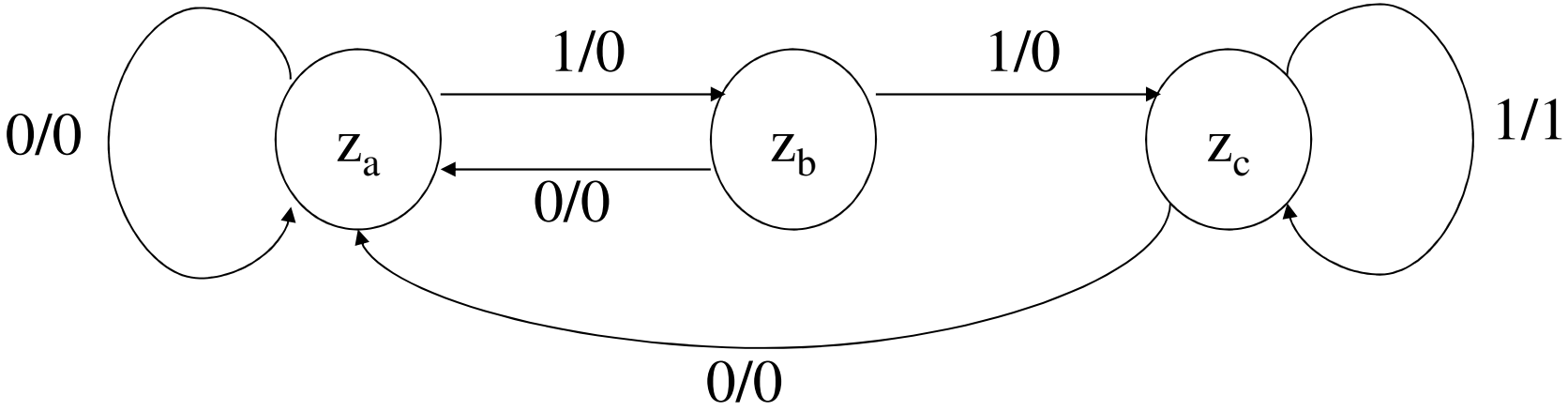
An element to store the values 0 and 1 is called a flipflop.

A flipflop can change between its states according to an appropriate input combination.

Definition:

The state Z of a sequential circuit is an m-tupel $(z_{m-1}, z_{m-2},\ldots,z_1,z_0)$ with components $z_i \in B=\{0,1\}$.
Each component z_i stands for a flipflop.


Flipflops
unclocnid flipflops e.g. RS-basicFF
clocnid flipflops
Latches
level triggered flipflops e.g. D-Latch
System flipflops edge triggered flipflops e.g. JK-Master-Slave-FF


0/0
z_a
1/0
z_b
0/0
1/0
z_c
1/1
0/0

x	z_1	z_0	z_1‘	z_0‘	y
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	X	X	X
1	0	0	0	1	0
1	0	1	1	0	0
1	1	0	1	0	1
1	1	1	X	X	X

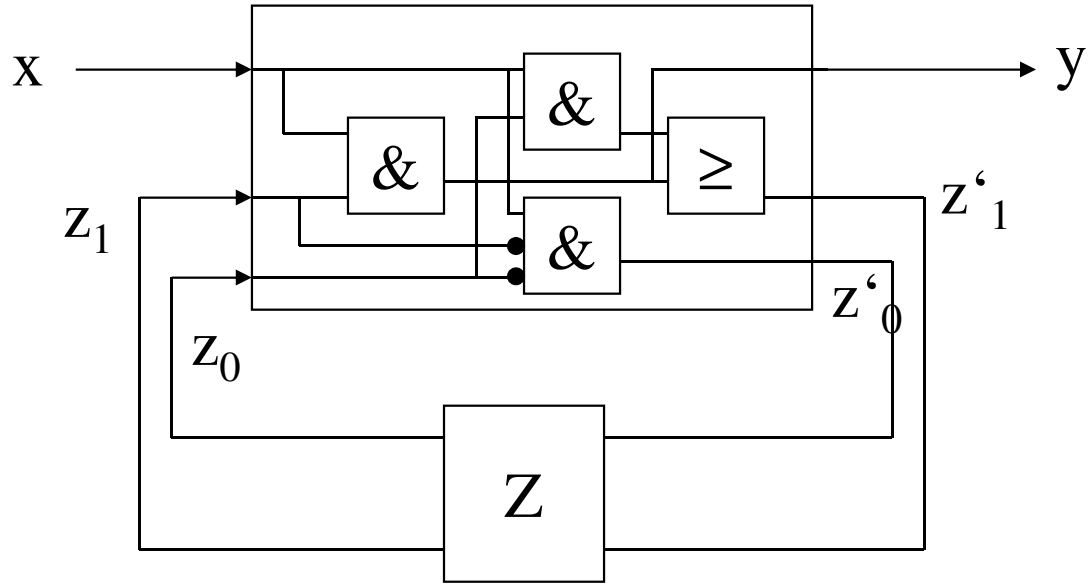
KV-diagrams:

$z_0' = x\ \underline{z_0}\ \underline{z_1}$

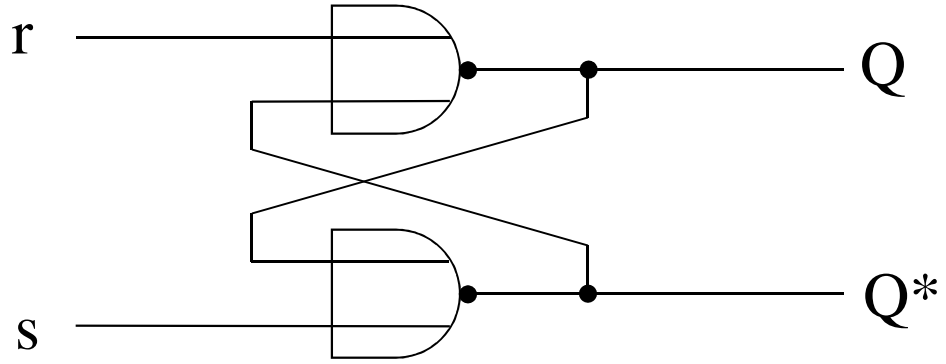
$z_1' = x\ z_0 + x\ z_1$

$y = x\ z_1$

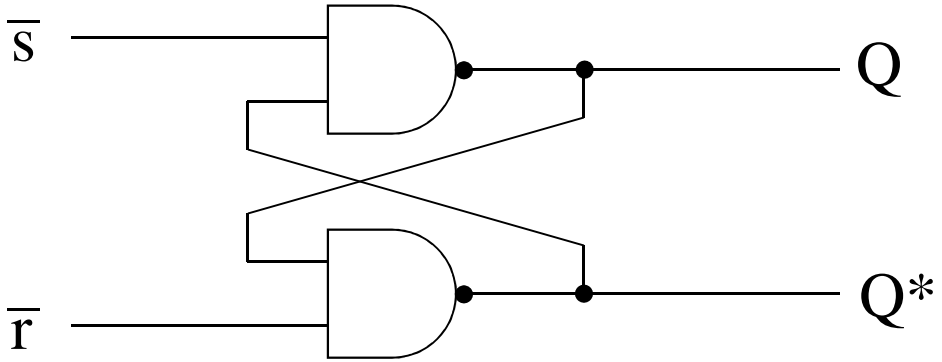
The complete sequential circuit:



Das r-s-Flipflop



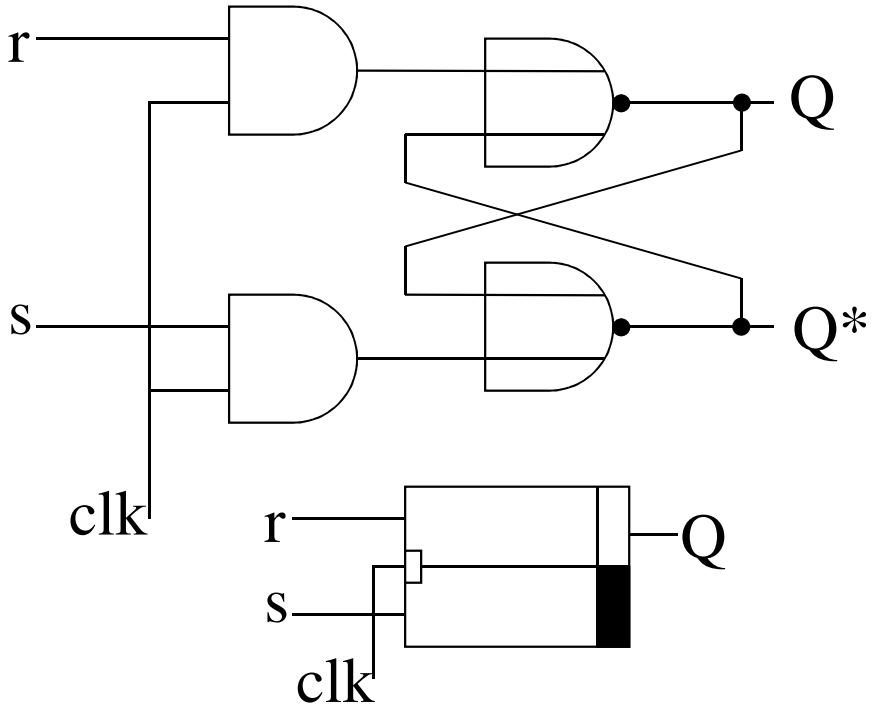
r	s	Q	Q*
0	0	Q_{alt}	Q^*_{alt}
0	1	1	0
1	0	0	1
1	1	0	0



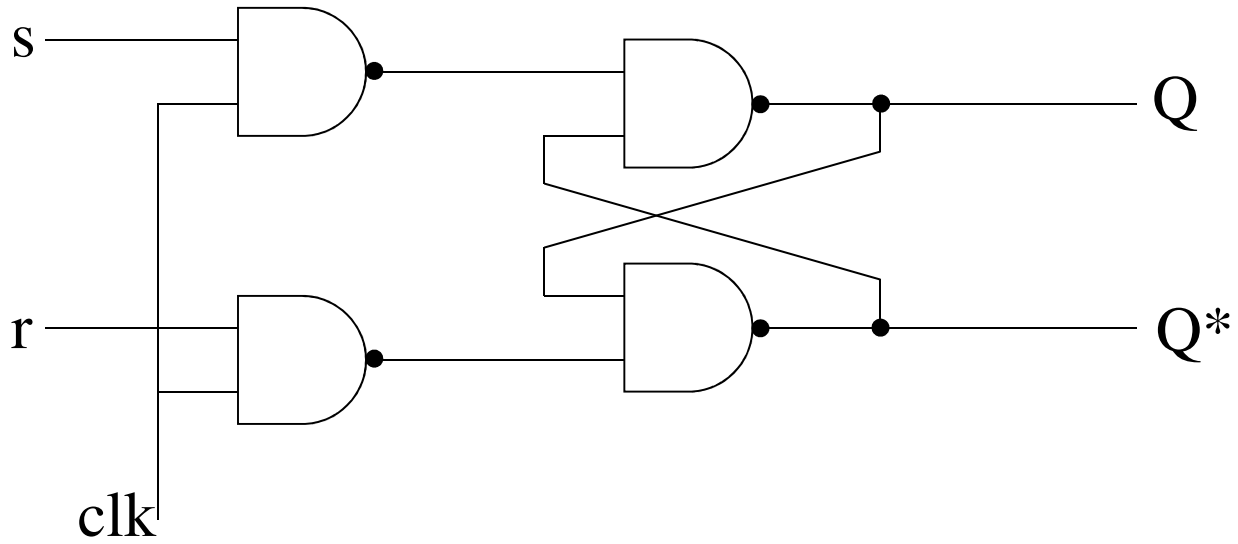
$\overline{r}$	$\overline{s}$	Q	Q*
1	1	Q_{alt}	Q^*_{alt}
1	0	1	0
0	1	0	1
0	0	1	1

Truth table:

clk	r	s	Q	Q*
0	0	0	Q_{alt}	Q^*_{alt}
0	0	1	Q_{alt}	Q^*_{alt}
0	1	0	Q_{alt}	Q^*_{alt}
0	1	1	Q_{alt}	Q^*_{alt}
1	0	0	Q_{alt}	Q^*_{alt}
1	0	1	1	0
1	1	0	0	1
1	1	1	forbidden	



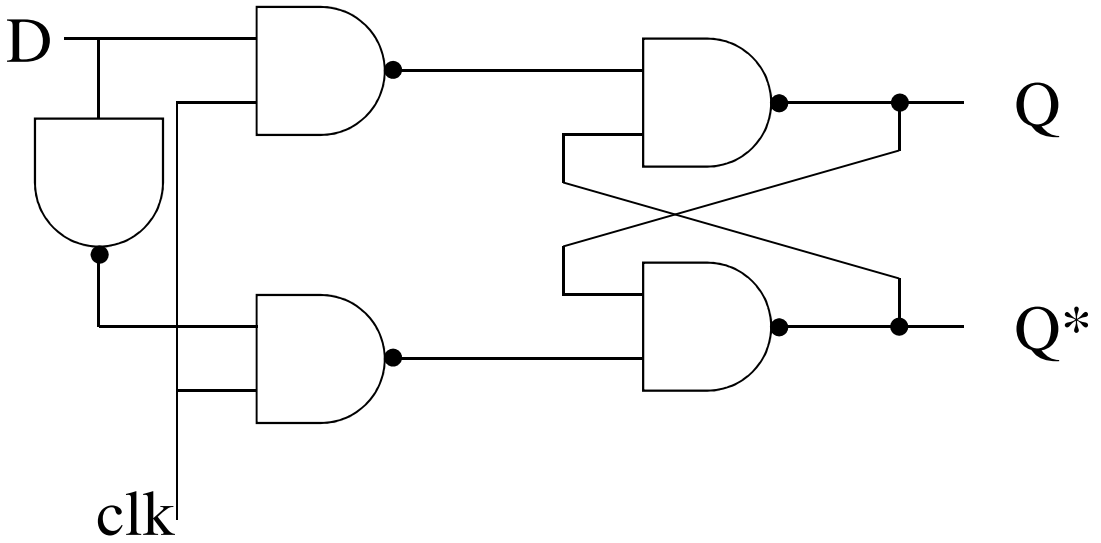
Nand Logic



D-Latch

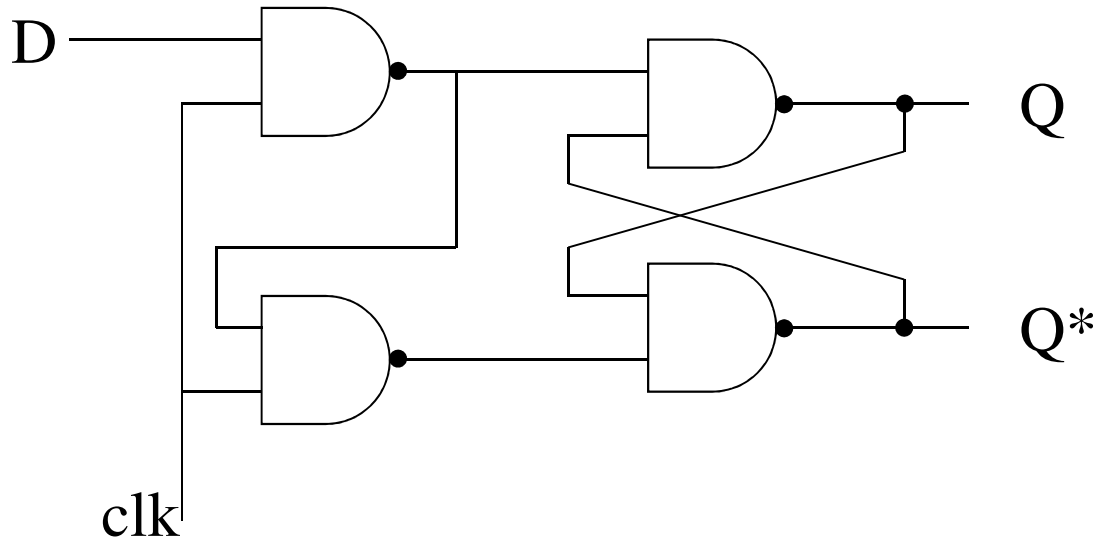
Truth table:

clk	D	Q	Q*
0	0	Q_{alt}	Q^*_{alt}
0	1	Q_{alt}	Q^*_{alt}
1	0	0	1
1	1	1	0



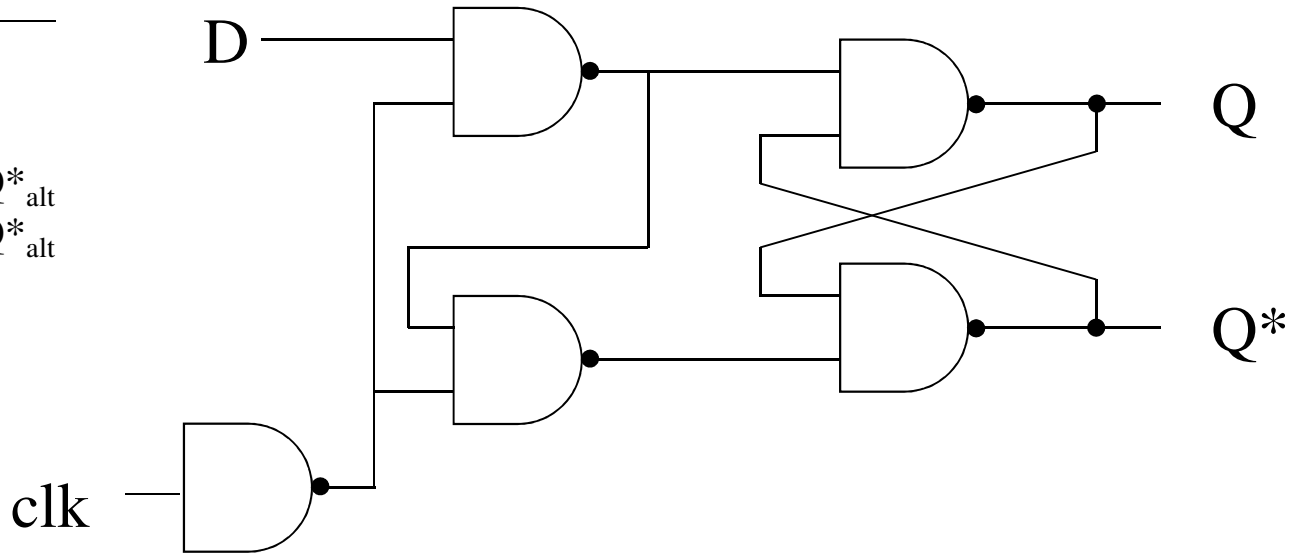
Saving one inverter

clk	D	Q	Q*
0	0	Q_{alt}	Q^*_{alt}
0	1	Q_{alt}	Q^*_{alt}
1	0	0	1
1	1	1	0

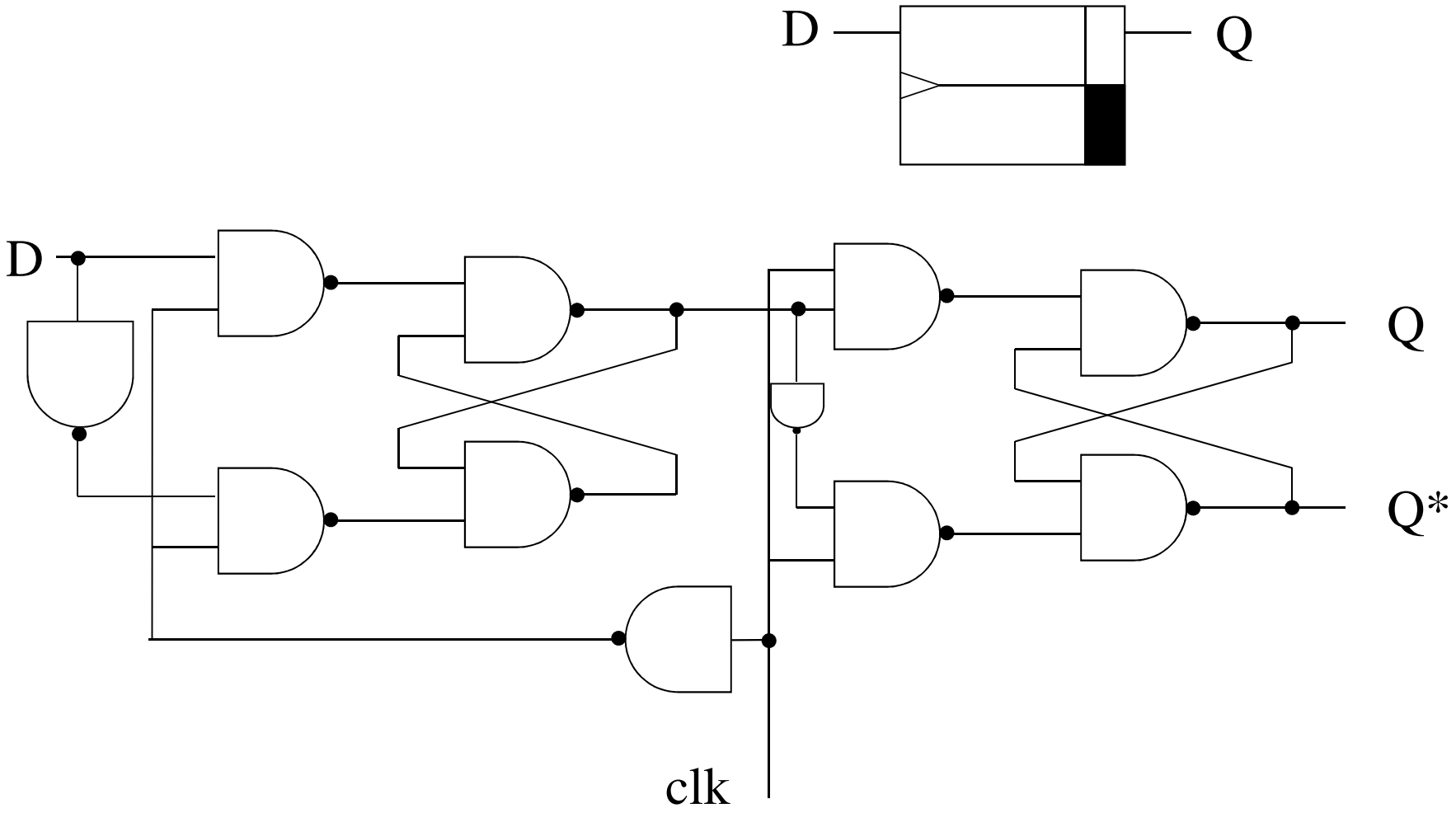


Negativ level triggered flipflop

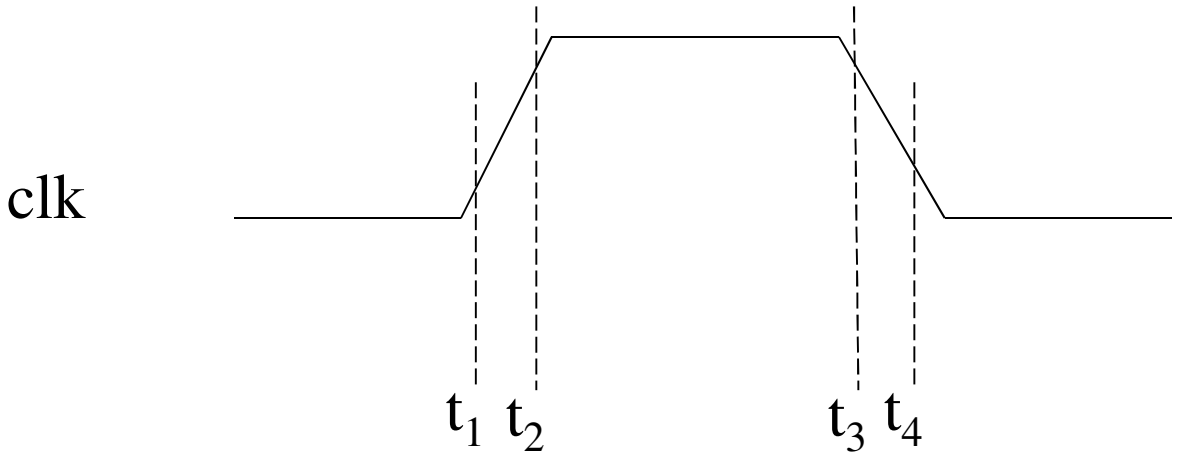
clk	D	Q	Q*
0	0	0	1
0	1	1	0
1	0	Q_{alt}	Q^*_{alt}
1	1	Q_{alt}	Q^*_{alt}



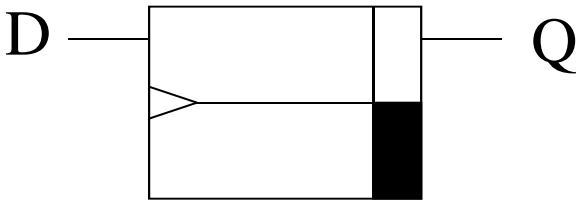
Master-Slave D-Flipflop

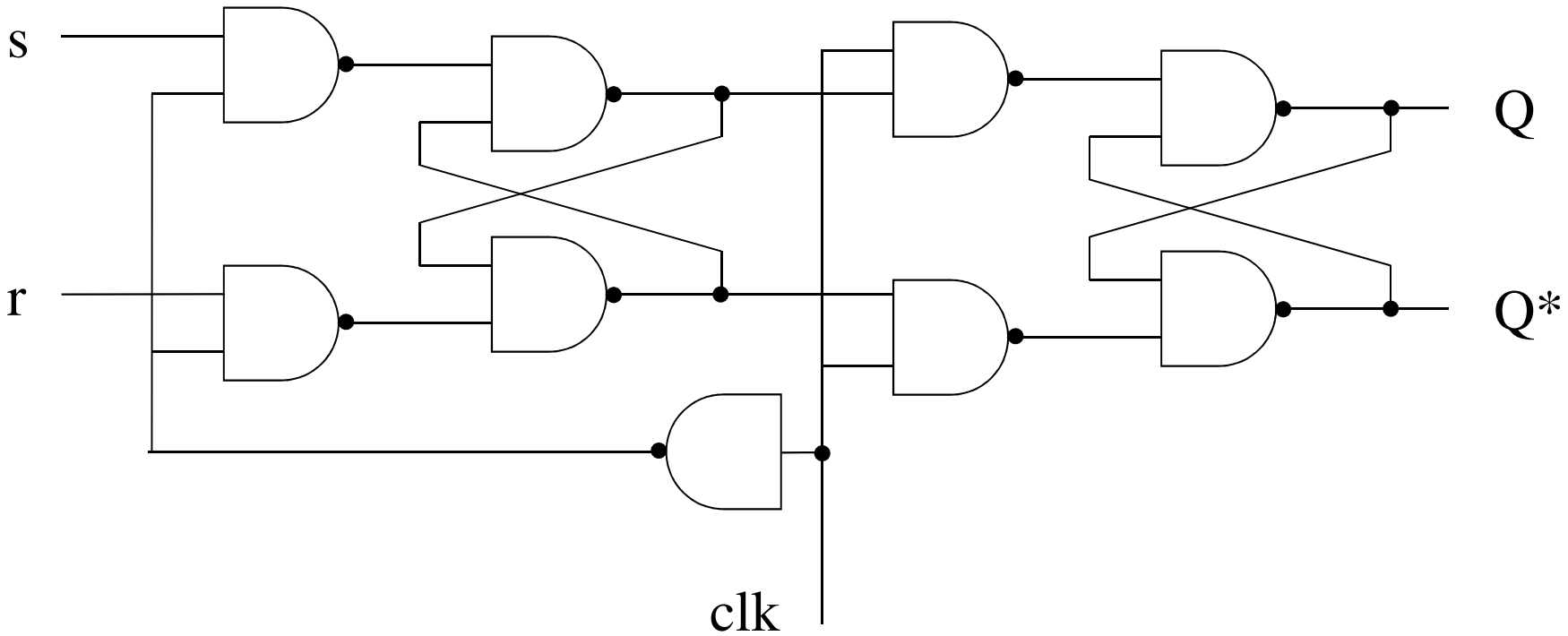


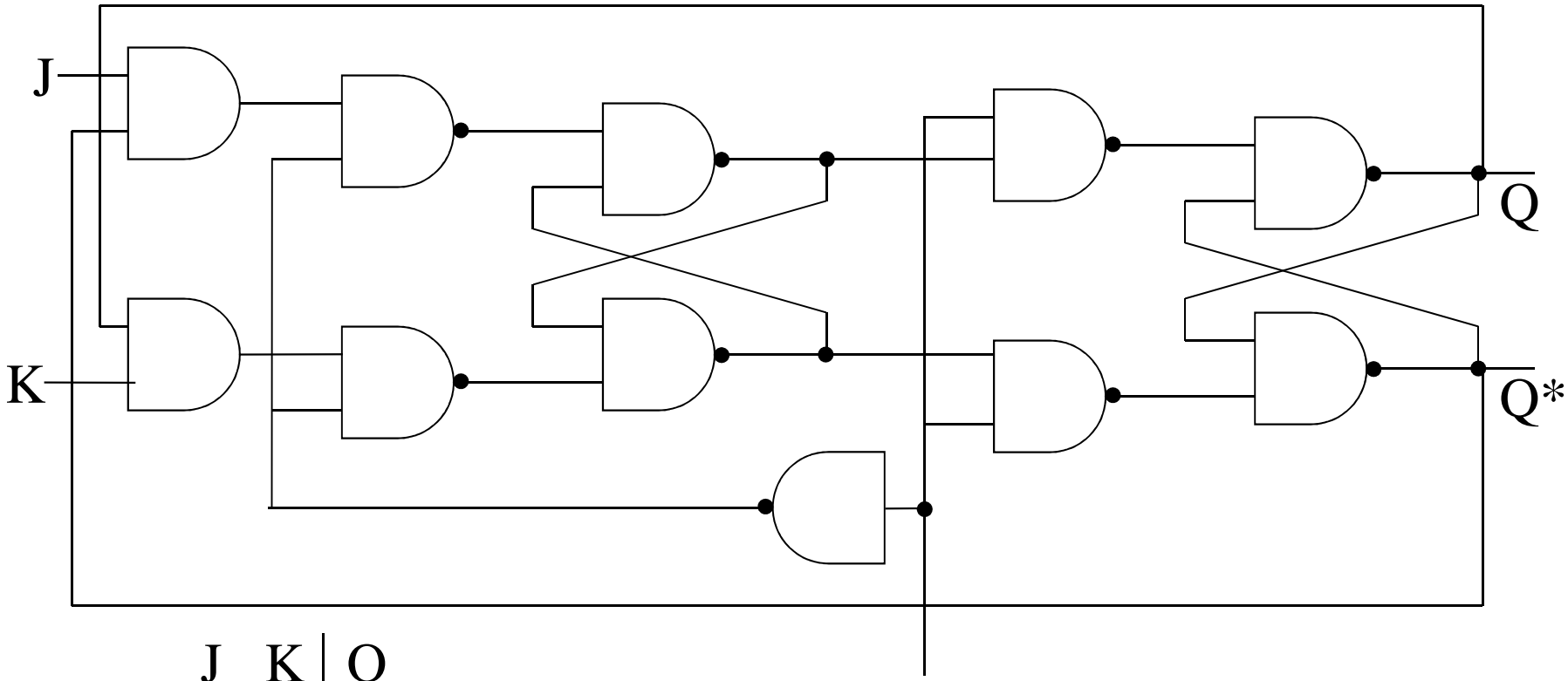
Master-Slave D-Flipflop, Timing



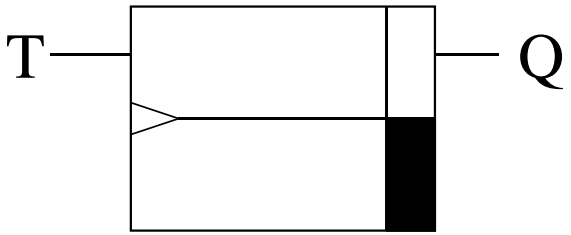
t_1: Master stores
t_2: Slave opens
t_3: Slave stores
t_4: Master opens

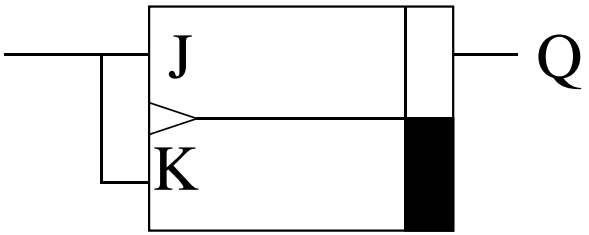



s
r
clk
Q
Q*



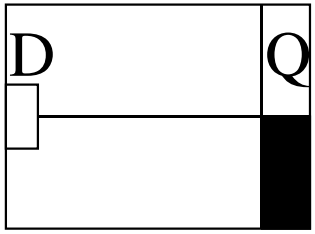
J	K	Q
0	0	Q_{alt}
0	1	0
1	0	1
1	1	$\overline{Q_{alt}}$



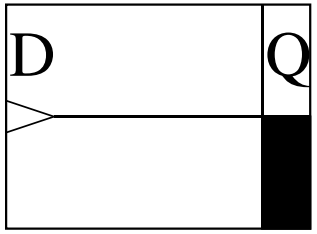


T	Q
0	Q_{alt}
1	$\overline{Q_{alt}}$

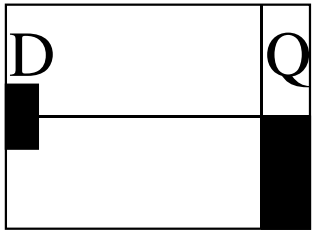
Positiv
Level triggered



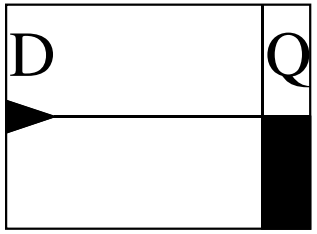
Positiv
Edge triggered

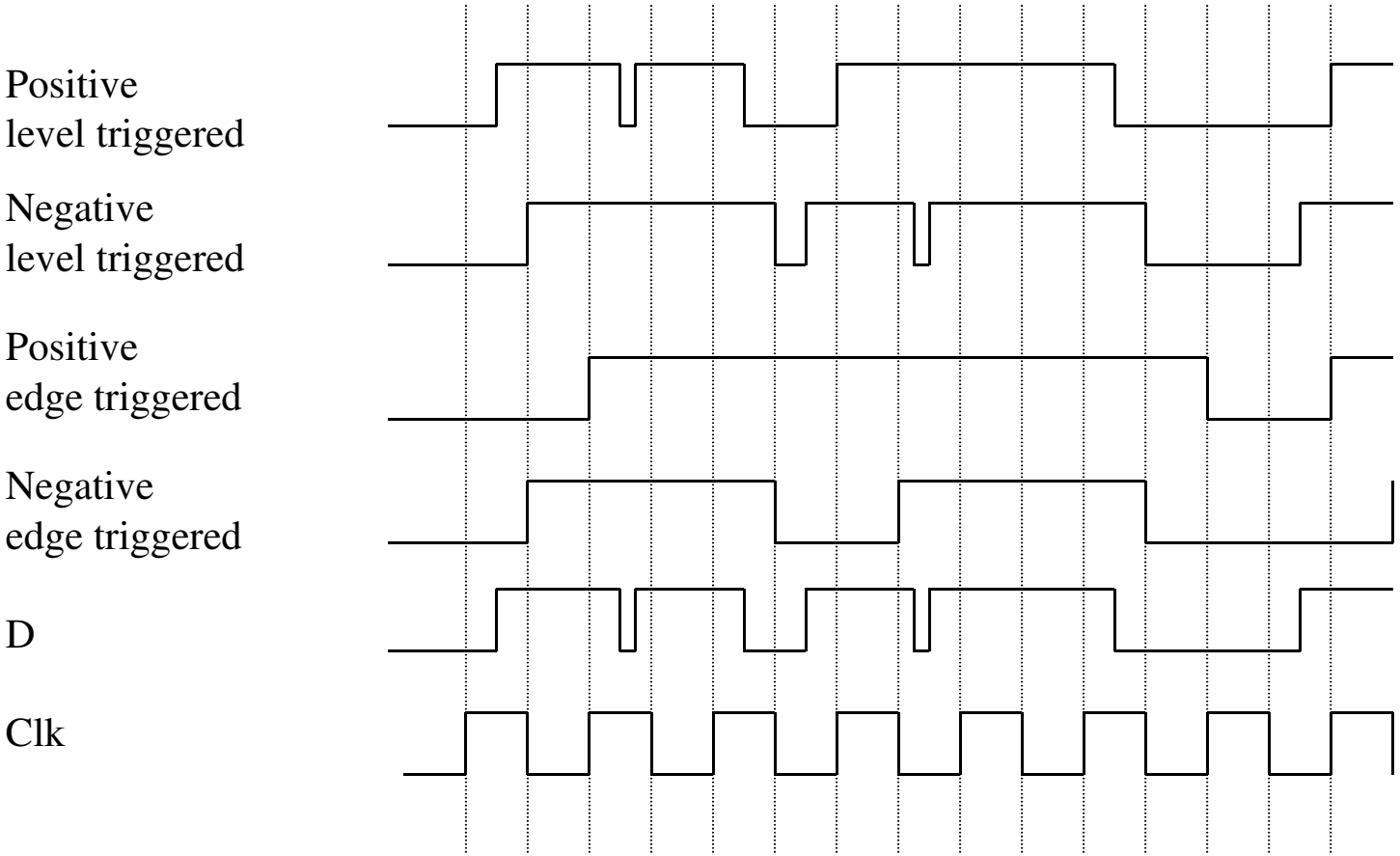


Negativ
Level triggered



Negativ
Edge triggered



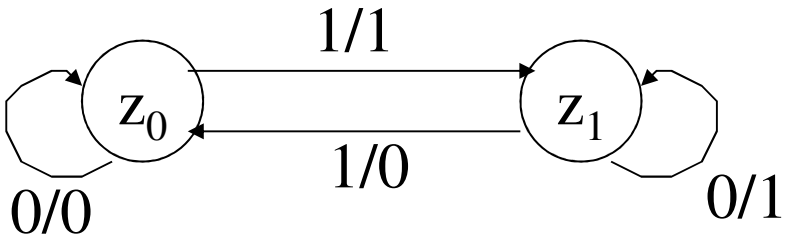

Positive
level triggered
Negative
level triggered
Positive
edge triggered
Negative
edge triggered
D
Clk

Example:
FSM for detecting even parity for bit serial binary value

$A = (X, Y, Z, \delta, \lambda)$

$X = \{ 0,1 \}$
$Y = \{ 0,1 \}$
$Z = \{ 0,1 \}$

δ	0	1
0	0	1
1	1	0



λ	0	1
0	0	1
1	1	0

x	z	z‘	y
0	0	0	0
0	1	1	1
1	0	1	1
1	1	0	0

Example:
Coke-Automaton

$A = (X, Y, Z, \delta, \lambda)$

$X = \{ 0.50, 1.-, ni, back \}$
$Y = \{ Cola, 0.50, 1.-, no \}$
$Z = \{ Empty, 0.50, 1.- \}$

δ	0.50	1.-	ni	back
empty	0.50	1.-	empty	empty
0.50	1.-	empty	0.50	empty
1.-	empty	0.50	1.-	empty

λ	0.50	1.-	ni	back
empty	no	no	no	no
0.50	no	Cola	no	0.50
1.-	Cola	Cola	no	1.-

Wertetabelle:

x_1	x_0	z_1	z_0	z_1‘	z_0‘	y_1	y_0
0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0
0	0	1	0	1	0	0	0
0	0	1	1	X	X	X	X
0	1	0	0	0	1	0	0
0	1	0	1	1	0	0	0
0	1	1	0	0	0	1	1
0	1	1	1	X	X	X	X
1	0	0	0	1	0	0	0
1	0	0	1	0	0	1	1
1	0	1	0	0	1	1	1
1	0	1	1	X	X	X	X
1	1	0	0	0	0	0	0
1	1	0	1	0	0	0	1
1	1	1	0	0	0	1	0
1	1	1	1	X	X	X	X

KV-Diagrams:

z_1 ‘

x_1 \ x_0				
				1
		d	d	
z_0	1	d	d	
			1	

z_1

y_1

x_1 \ x_0				
		1	1	
		d	d	1
z_0		d	d	
		1		

z_1

z_0 ‘

x_1 \ x_0				
			1	
		d	d	
z_0		d	d	1
	1			

z_1

y_0

x_1 \ x_0				
			1	
	1	d	d	1
z_0		d	d	
		1		

z_1

$$z_1' = \overline{x_0} x_1 \overline{z_0}\,\overline{z_1} + \overline{x_0}\,\overline{x_1} z_1 + x_0 \overline{x_1} z_0$$

$$z_0' = \overline{x_0} x_1 z_1 + \overline{x_0}\,\overline{x_1} z_0 + x_0 \overline{x_1}\,\overline{z_0}\,\overline{z_1}$$

$$y_1 = x_1 z_1 + x_0 z_1 + \overline{x_0} x_1 z_0$$

$$y_0 = x_1 z_0 + \overline{x_0} x_1 z_1 + x_0 \overline{x_1} z_1$$

Implementation as an FPLA: