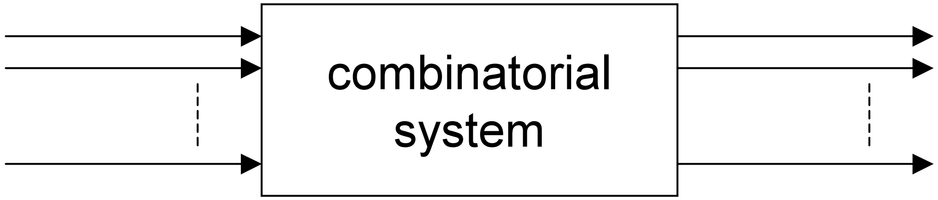
Combinatorial systems

$X = x_1, x_2 \dots x_n$



Logic gates

$Y = y_1, y_2 \dots y_m$

$y_1 = B_1(x_1, x_2 \dots x_n)$
$y_2 = B_2(x_1, x_2 \dots x_n)$
$\dots$
$y_m = B_m(x_1, x_2 \dots x_n)$

$Y = B(X)$

B_x – boolean functions

Any change of input signal X modifies the output signal Y
with maximum speed limited by the signal propagation time.

Output vector Y is a function of current value of X in any moment.

Combinatorial systems - description

np.

$$y(a, b, c) = a*(b+\bar{c}) + (\bar{a}+b)*c$$

Truth table

a	b	c	y(a, b, c)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Sum of maxterms

$$y(a, b, c) = \bar{a}\bar{b}c + \bar{a}bc + a\bar{b}\bar{c} + ab\bar{c} + abc$$

Product of minterms

$$y(a, b, c) = (a+b+c)\,(a+\bar{b}+c)\,(\bar{a}+b+\bar{c})$$

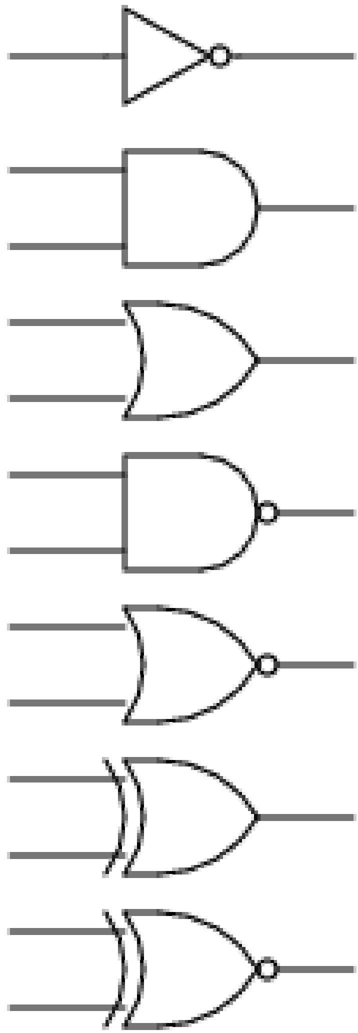
Optimalization algorithms of boolean functions !

Boolean algebra

Identity element	$a+0=a$	$a*1=a$
Commutativity	$a+b=b+a$	$a*b=b*a$
Associativity	$a+(b+c)=(a+b)+c$	$a*(b*c)=(a*b)*c$
Distributivity	$a+(b*c)=(a+b)*(a+c)$	$a*(b+c)=(a*b)+(a*c)$
Complement	$a+\bar{a}=1$	$a*\bar{a}=0$

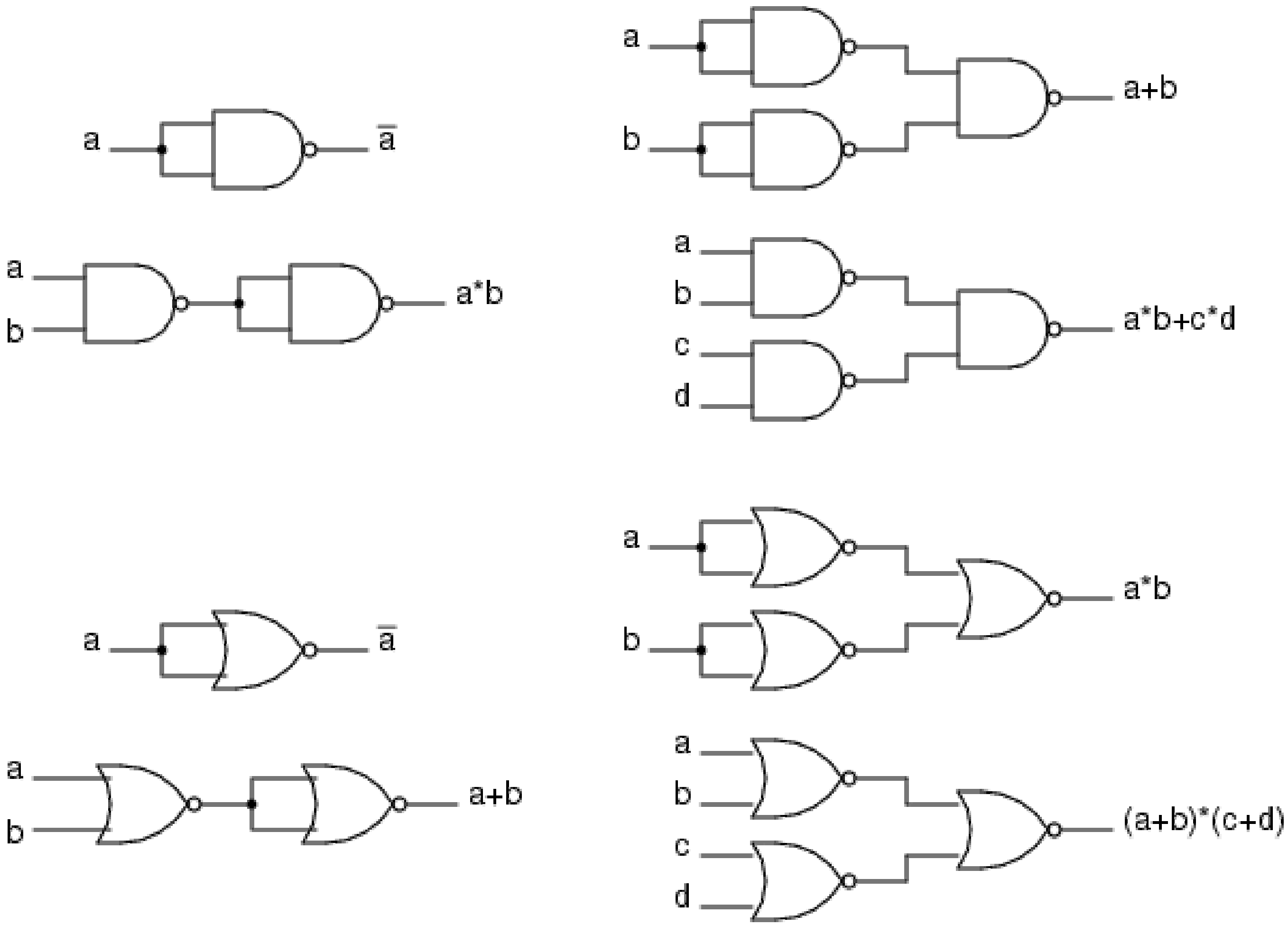
Idempotency	$a+a=a$	$a*a=a$
Complement	$a+1=1$	$a*0=0$
Absorption	$a+a*b=a$	$a*(a+b)=a$
Element Elimination	$a+\bar{a}*b=a+b$	$a*(\bar{a}+b)=a*b$
De Morgan’s Laws	$\overline{a+b}=\bar{a}*\bar{b}$	$\overline{a*b}=\bar{a}+\bar{b}$

Logic gates

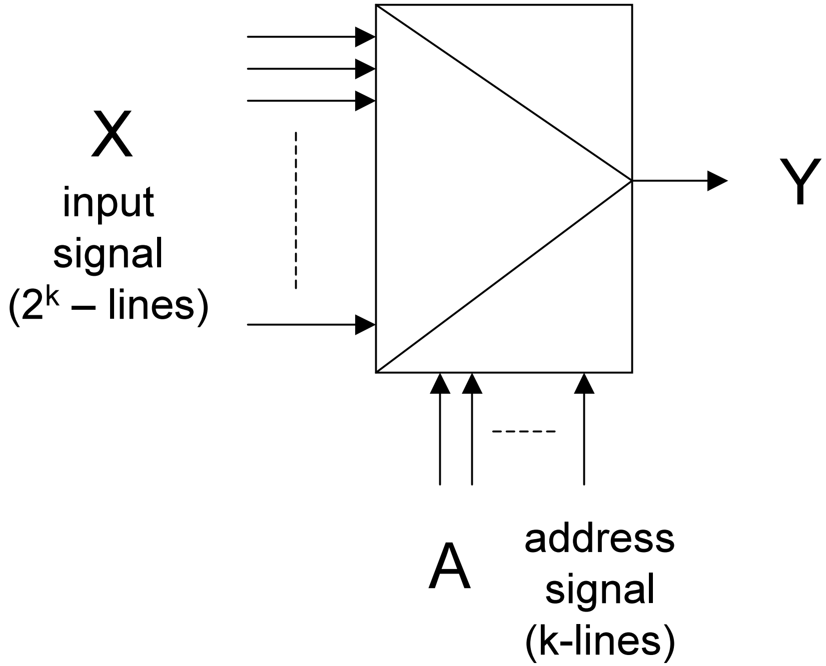


	NOT	$F = \overline{a}$
	AND	$F = a \bullet b$
	OR	$F = a + b$
	NAND	$F = \overline{a \bullet b}$
	NOR	$F = \overline{a + b}$
	XOR	$F = a \oplus b$
	XNOR	$F = \overline{a \oplus b}$

Realisation of boolean functions

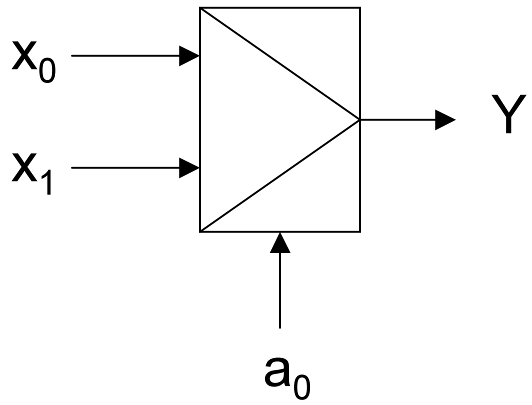


Multiplexers

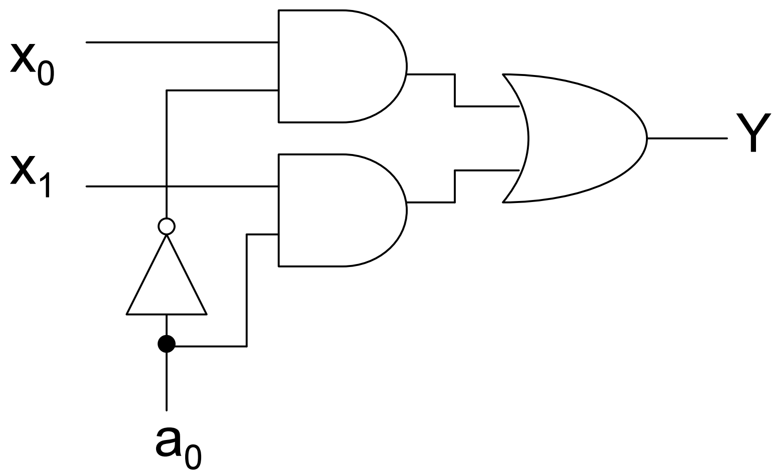


$$Y = x_0 \cdot \overline{a_{k-1}} \cdot \ldots \cdot \overline{a_1} \cdot \overline{a_0} + x_1 \cdot \overline{a_{k-1}} \cdot \ldots \cdot \overline{a_1} \cdot a_0 + \ldots + x_{2^k-1} \cdot a_{k-1} \cdot \ldots \cdot a_1 \cdot a_0$$

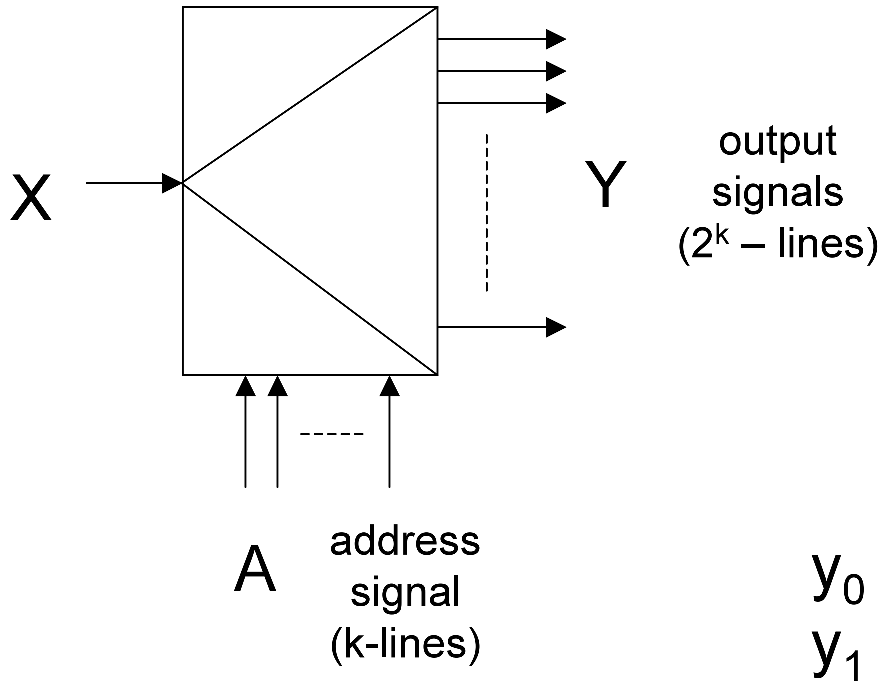
Multiplexer 2x1



$$Y = x_0 \cdot \overline{a_0} + x_1 \cdot a_0$$

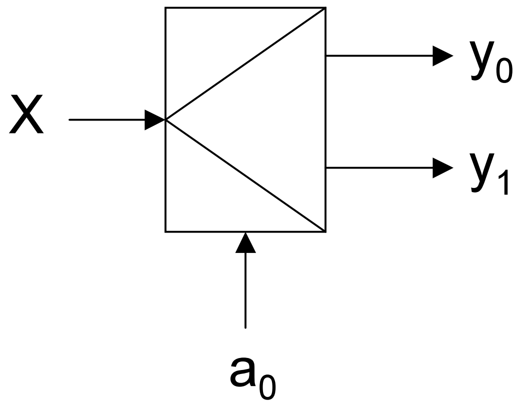


Demultiplexers

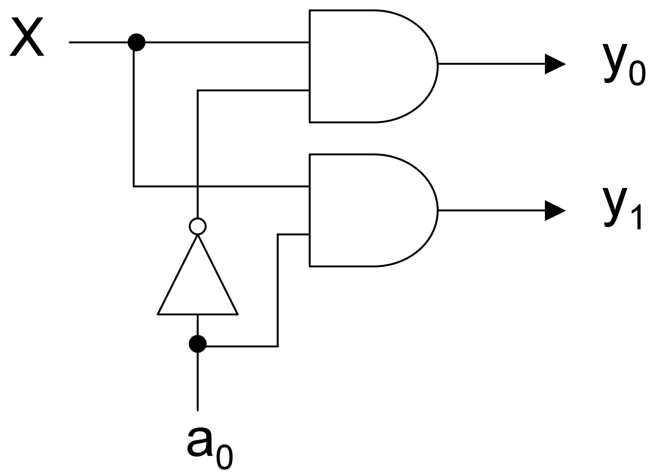


$$y_0 = X \cdot \overline{a_{k-1}} \cdot \ldots \cdot \overline{a_1} \cdot \overline{a_0}$$
$$y_1 = X \cdot \overline{a_{k-1}} \cdot \ldots \cdot \overline{a_1} \cdot a_0$$
$$\ldots$$
$$y_{2k-1} = X \cdot a_{k-1} \cdot \ldots \cdot a_1 \cdot a_0$$

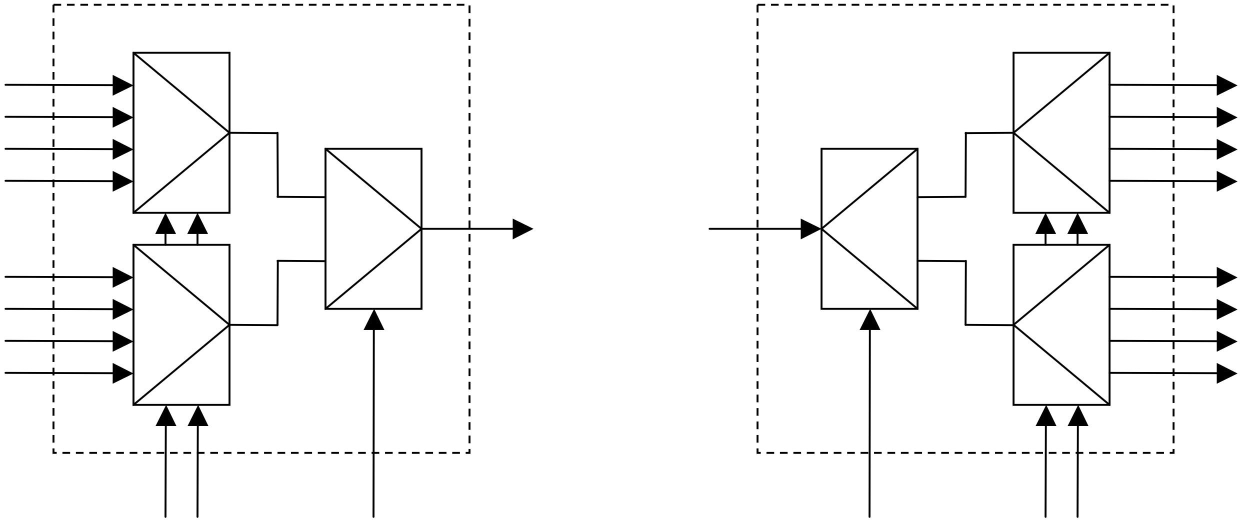
Demultiplexer 1x2



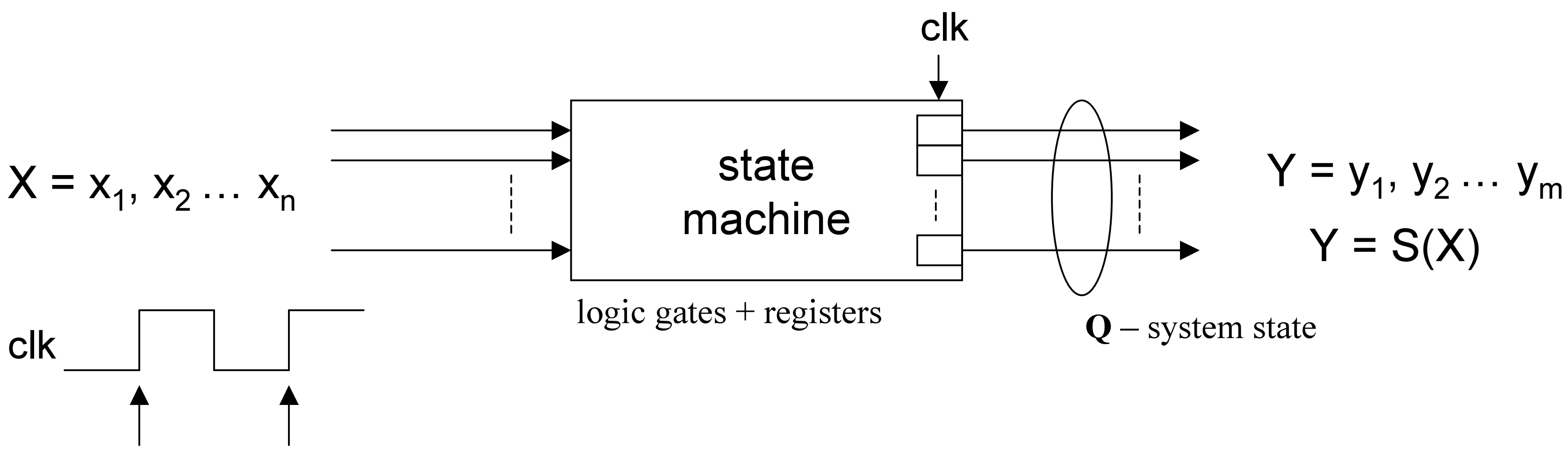
$$y_0 = X \cdot \overline{a_0}$$
$$y_1 = X \cdot a_0$$



Cascades of (de)multiplexers



State machines

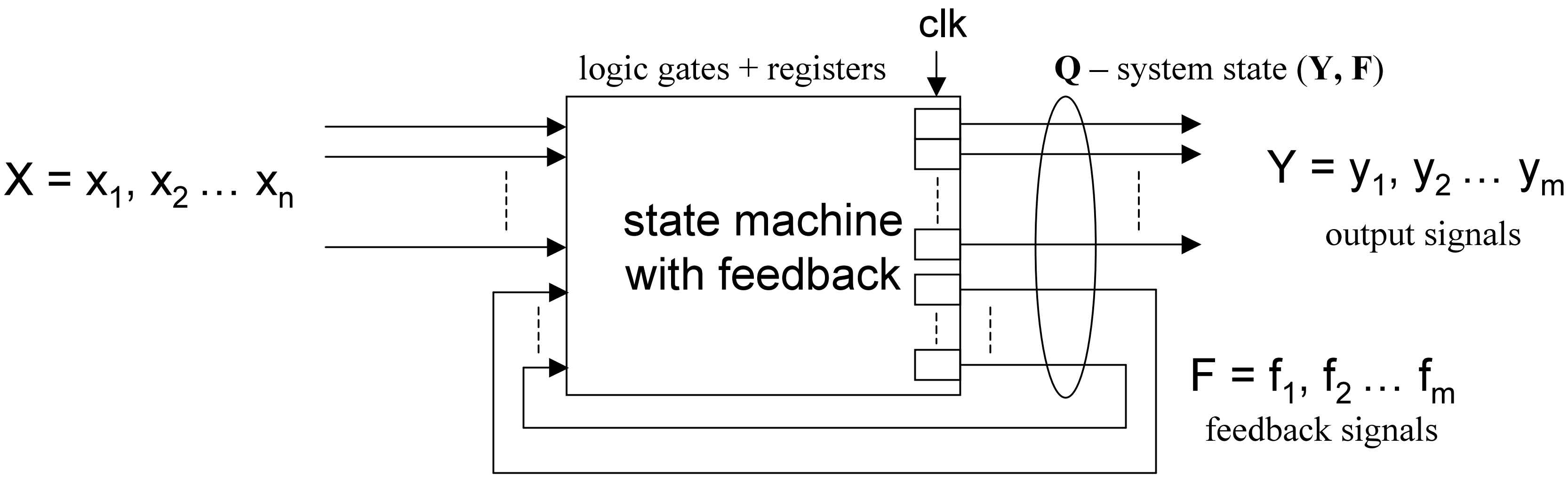


Any change of output signal Y is only possible at the change of the clock signal to memory elements (edge triggered write operation)

Output signals Y (when changed) are the function of input signal X only in the moment of the clock edge. In any other moment, the output Y is stable and independent of any change in input X.

Description of state machine consists in enumerating the output states Q and the sequence of their change (state diagram).

State machine with feedback



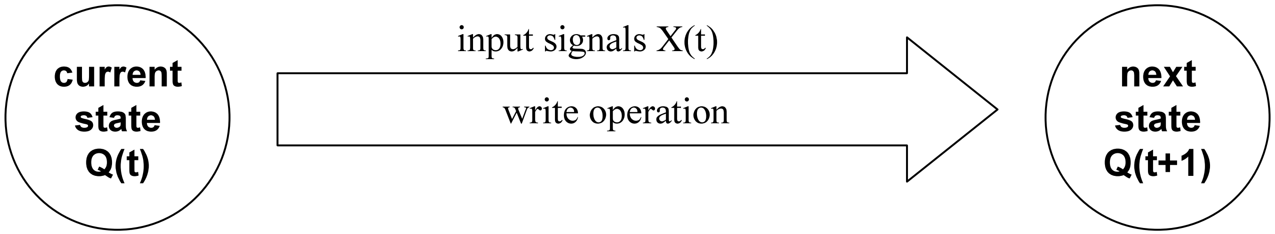
$Y = S(X,F)$

Change of Y (and Q) is synchronised with clock signal.

Next state of the system depends on current input signals X and on current system state F (feedback).

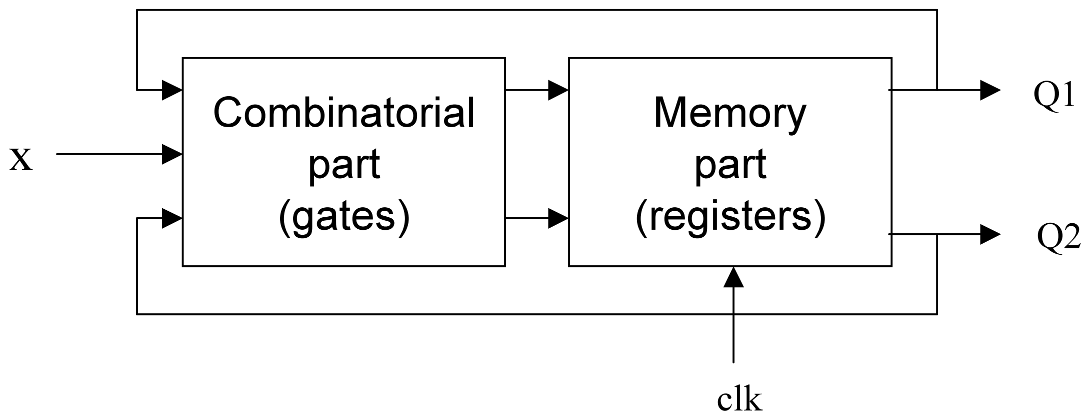
System state can be described by set of values Y, F and sequence of their change (state diagram).

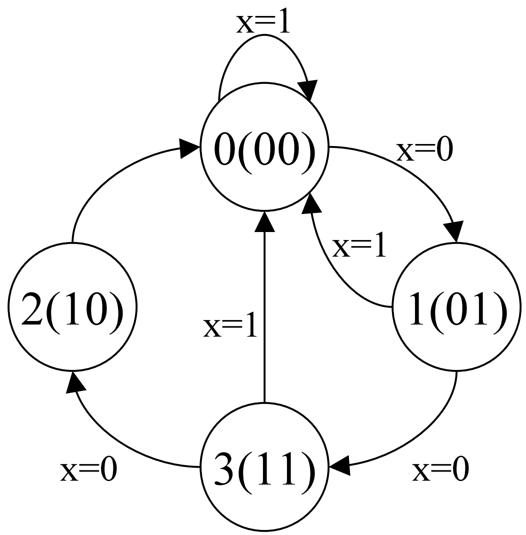
State diagrams



Next state depends on:
1) current state (signals Y, F)
2) input signals at the moment of write

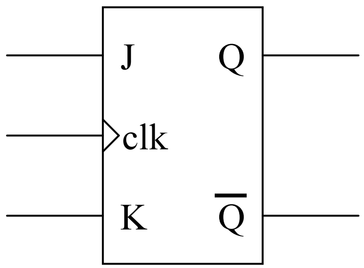
Example





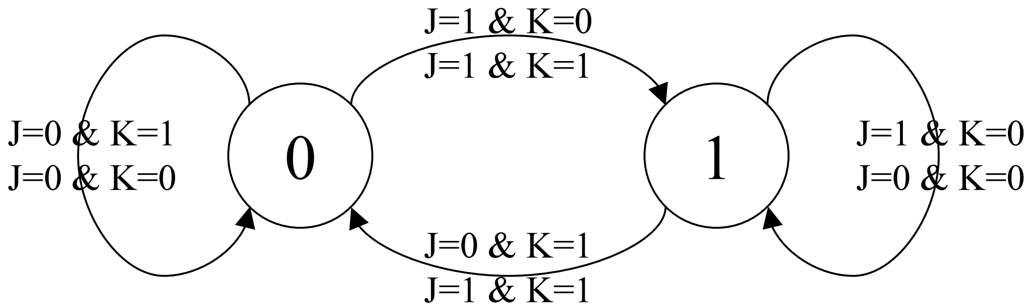
State diagrams - examples

JK Flip-flop

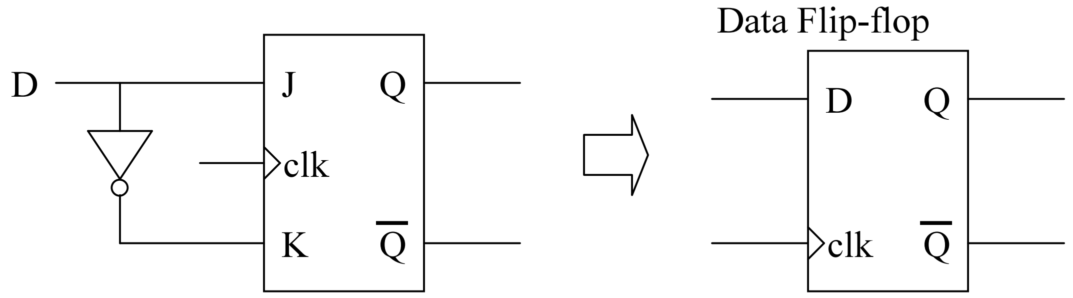


J(t)	K(t)	Q(t+1)
0	0	Q(t)
0	1	0
1	0	1
1	1	$\overline{Q}$(t)

$$Q(t+1) = J(t)\,\overline{Q}(t) + \overline{K}(t)\,Q(t)$$

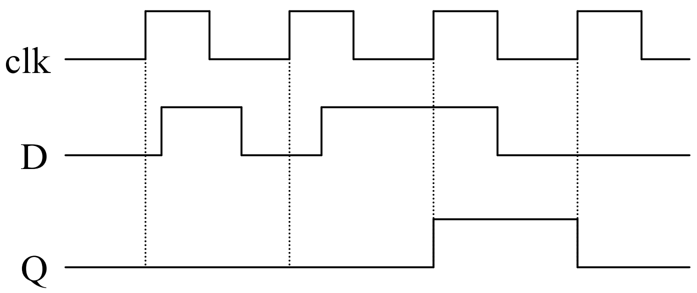


State diagrams - examples

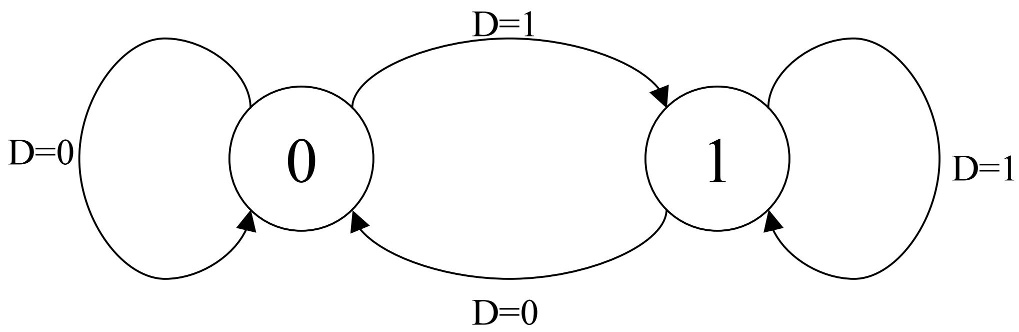


D(t)	Q(t+1)
0	0
1	1

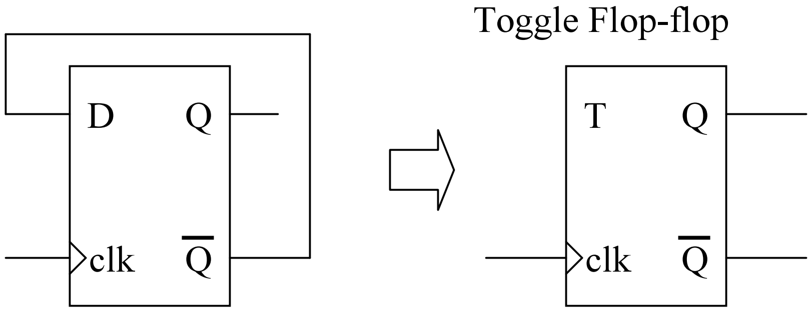
$$Q(t+1) = D(t)$$

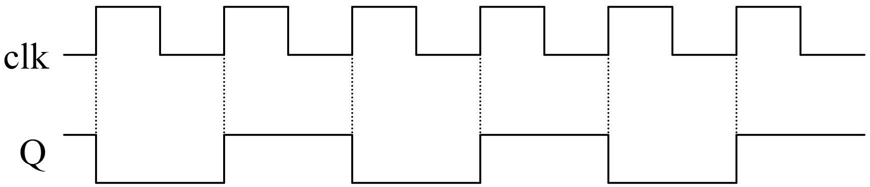


$$Q(t+1) = D(t)$$

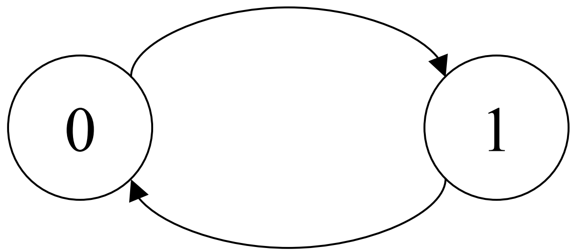


State diagrams - examples



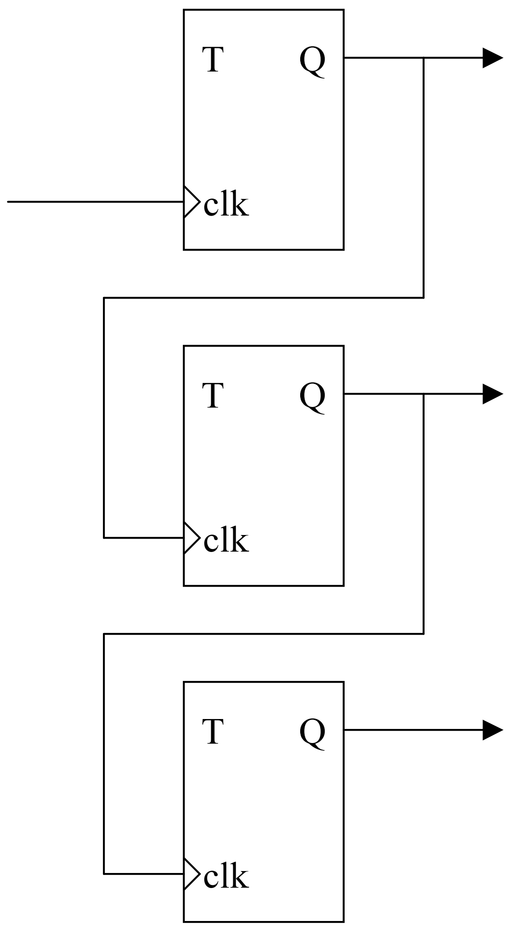


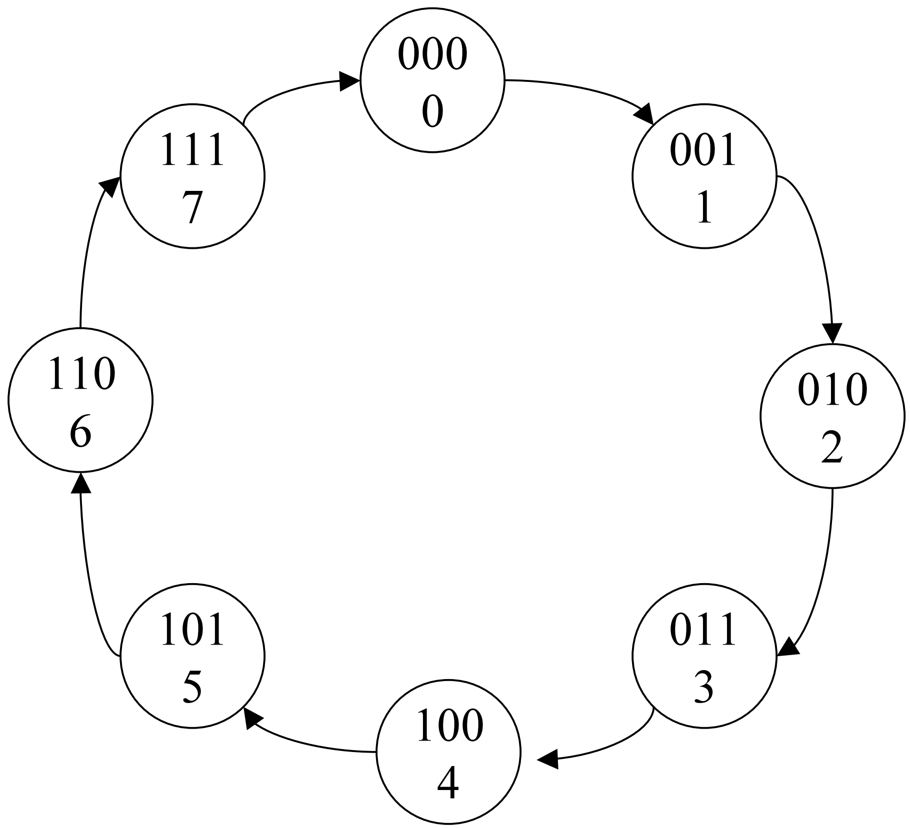
$$Q(t+1) = \overline{Q}(t)$$



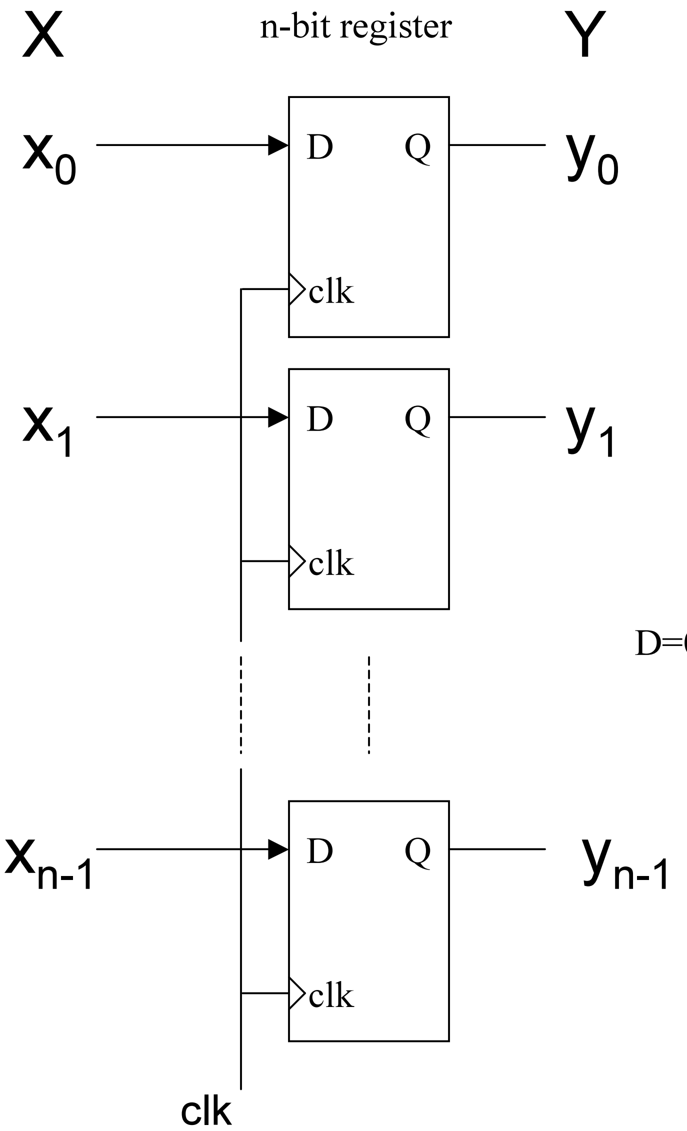
State diagrams - examples

Counter





State diagram - examples



D=0