

Lecture 11.

Synthesis and minimization of state machines. Elementary state machines.

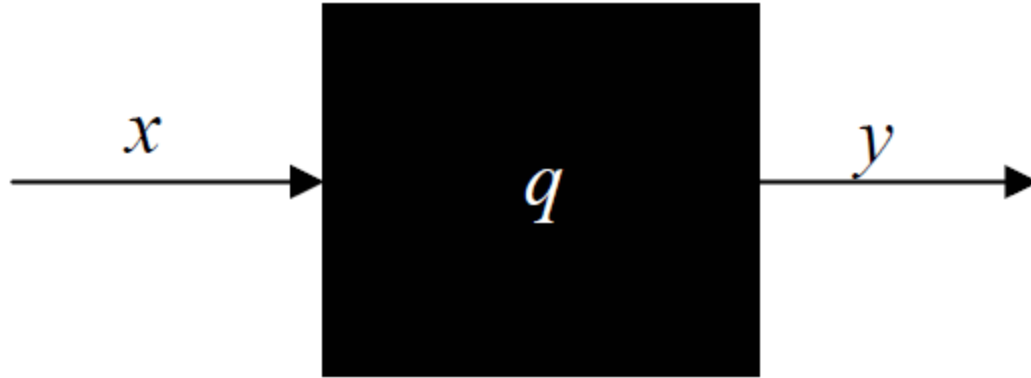
Dmitry Zaitsev

<http://daze.ho.ua>

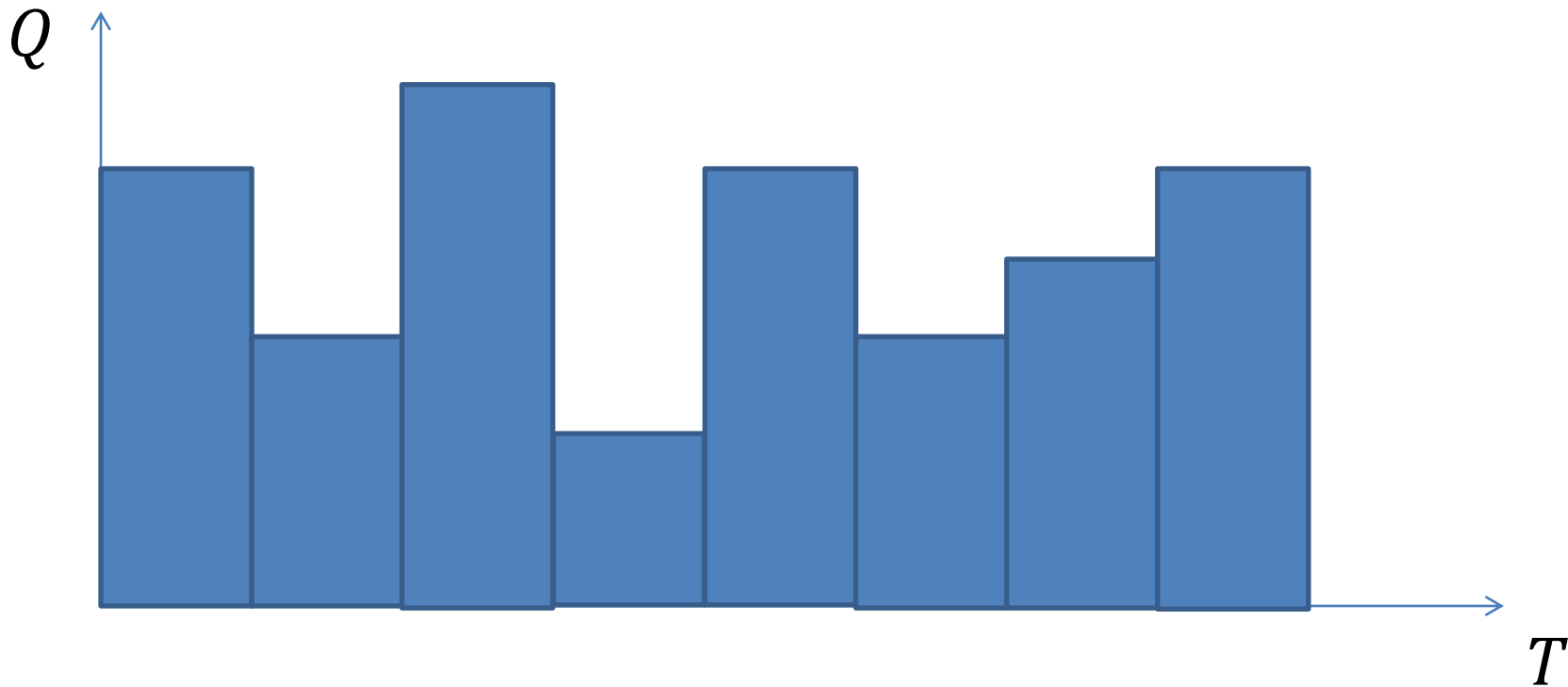
State machines

- Input $x \in X$
- Output $y \in Y$
- Internal state $q \in Q$
- Finite machines: X, Y, Q – finite

State machine as a black box



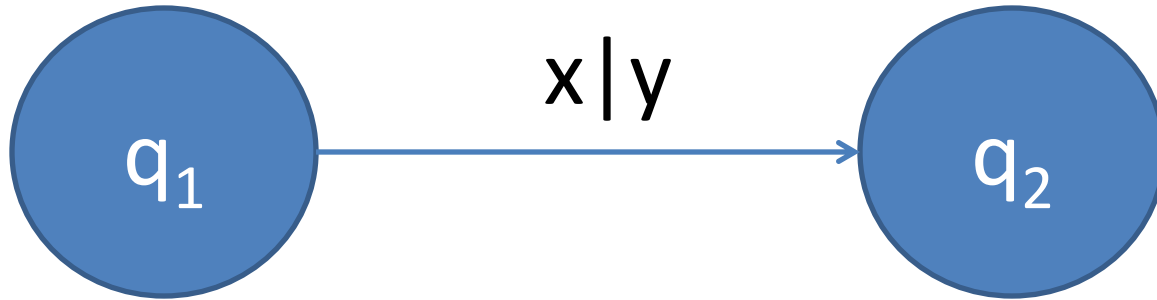
Discrete time



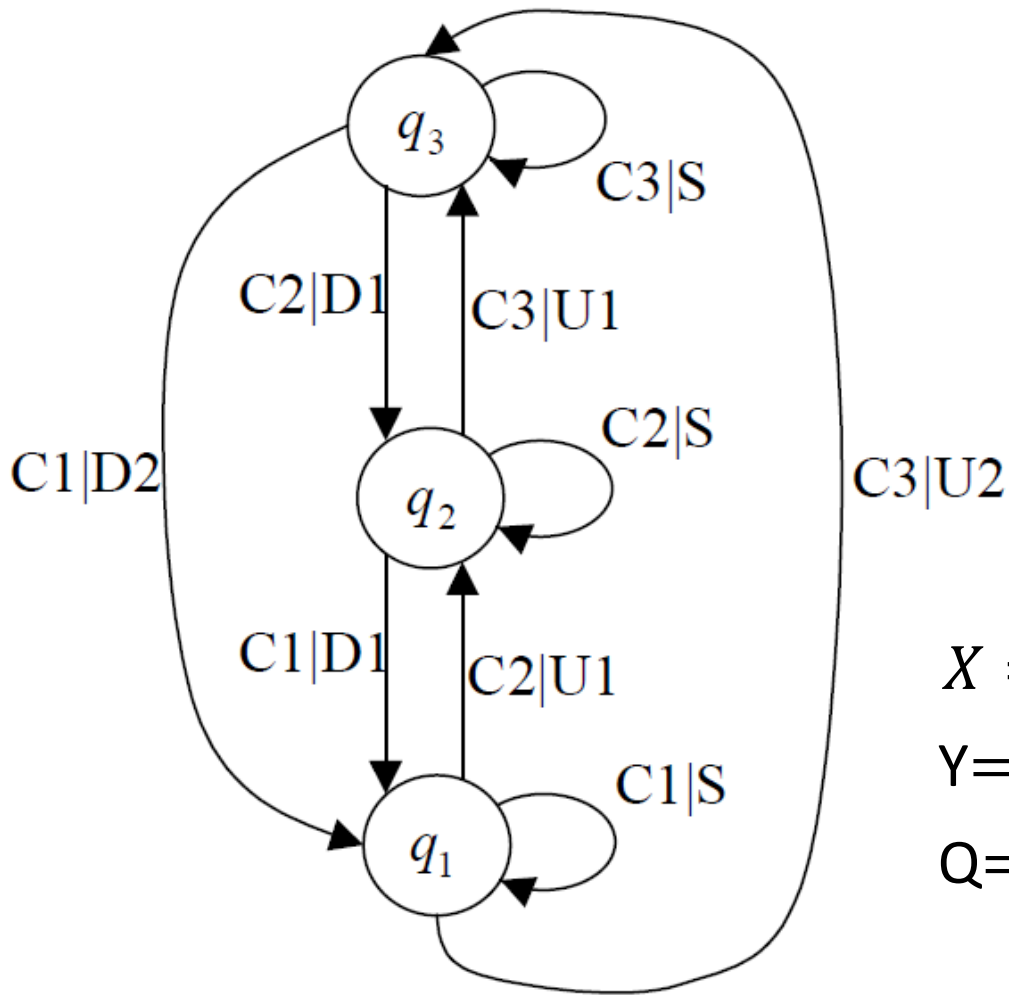
Mealy and Moore machines

- Mealy: next state and output depend on combination – the current state and input
- Moore: next state depends on combination – the current state and input; output depends only on the current state

State diagram



Elevator in 3 storey building



$$X = \{C1, C2, C3\}$$

$$Y = \{S, U1, U2, D1, D2\}$$

$$Q = \{q_1, q_2, q_3\}$$

Tabular form

Q/X	C1	C2	C3
q_1	$q_1 S$	$q_2 U1$	$q_3 U2$
q_2	$q_1 D1$	$q_2 S$	$q_3 U1$
q_3	$q_1 D2$	$q_2 D1$	$q_3 S$

Mealy machine

- $A = (X, Q, Y, q_0, F)$
- $X = \{x\}$ input alphabet
- $Q = \{q\}$ state alphabet
- $Y = \{y\}$ output alphabet
- q_0 initial state
- $F: X \times Q \rightarrow Y \times Q$ transition function

Moore machine

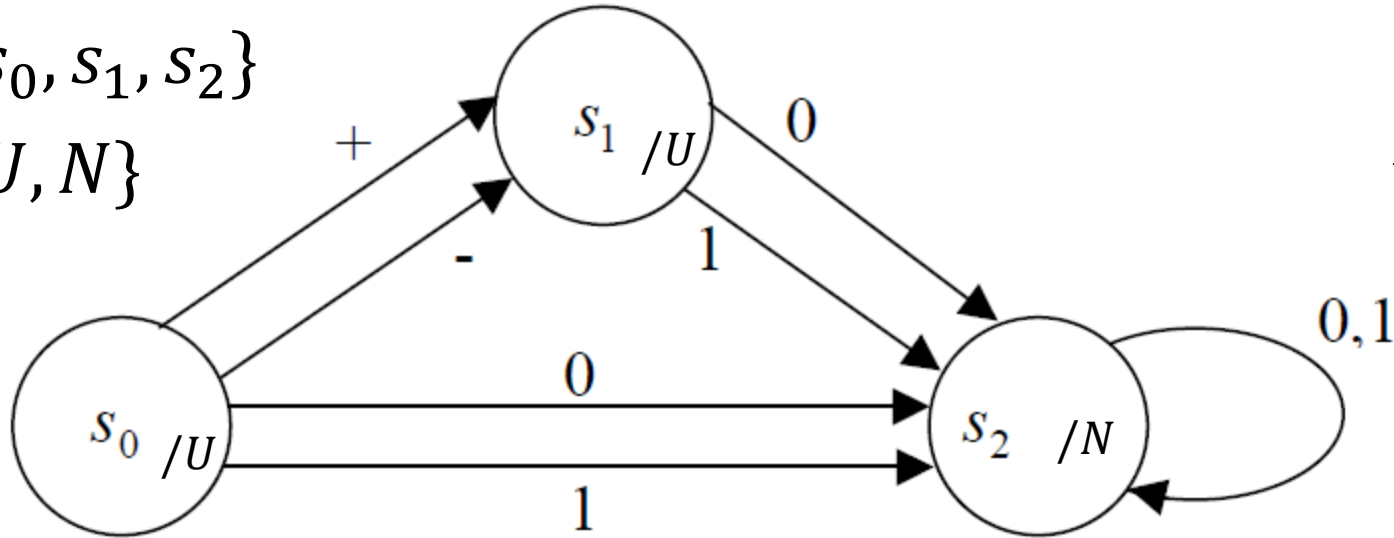
- $B = (X, S, Y, s_0, P, R)$
- $X = \{x\}$ input alphabet
- $S = \{s\}$ state alphabet
- $Y = \{y\}$ output alphabet
- s_0 initial state
- $P: X \times S \rightarrow S$ transition function
- $R: S \rightarrow Y$ output function

Moore machine example

$X = \{+, -, 0, 1\}$

$S = \{s_0, s_1, s_2\}$

$Y = \{U, N\}$



010001

NNNNNN

-110110

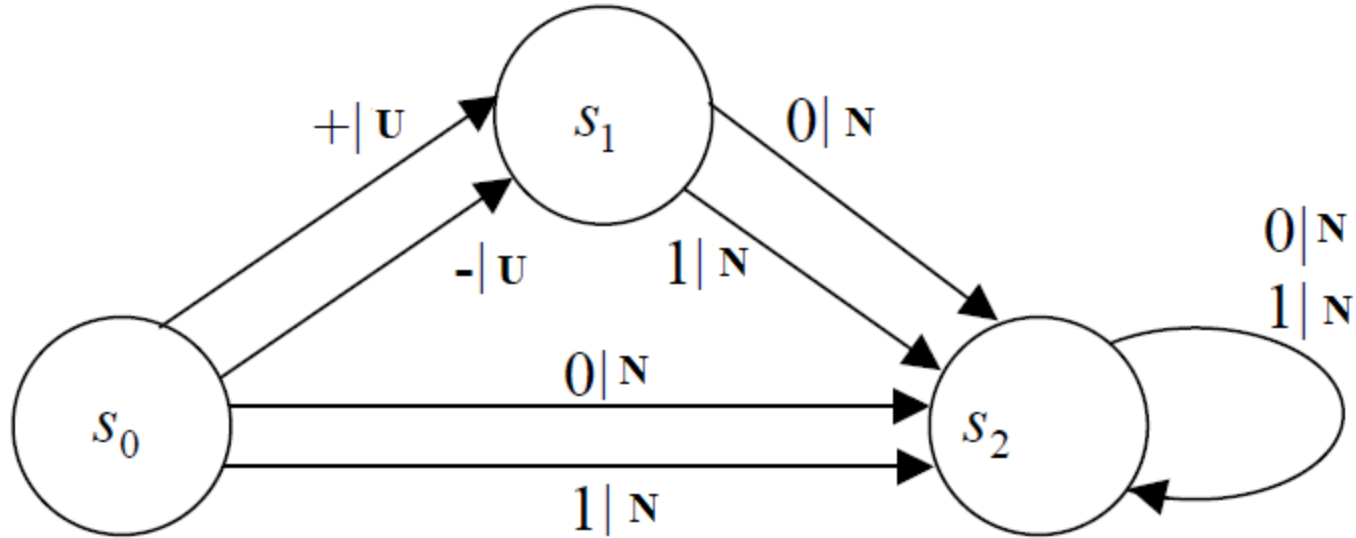
UNNNNNN

recognize integer binary constants

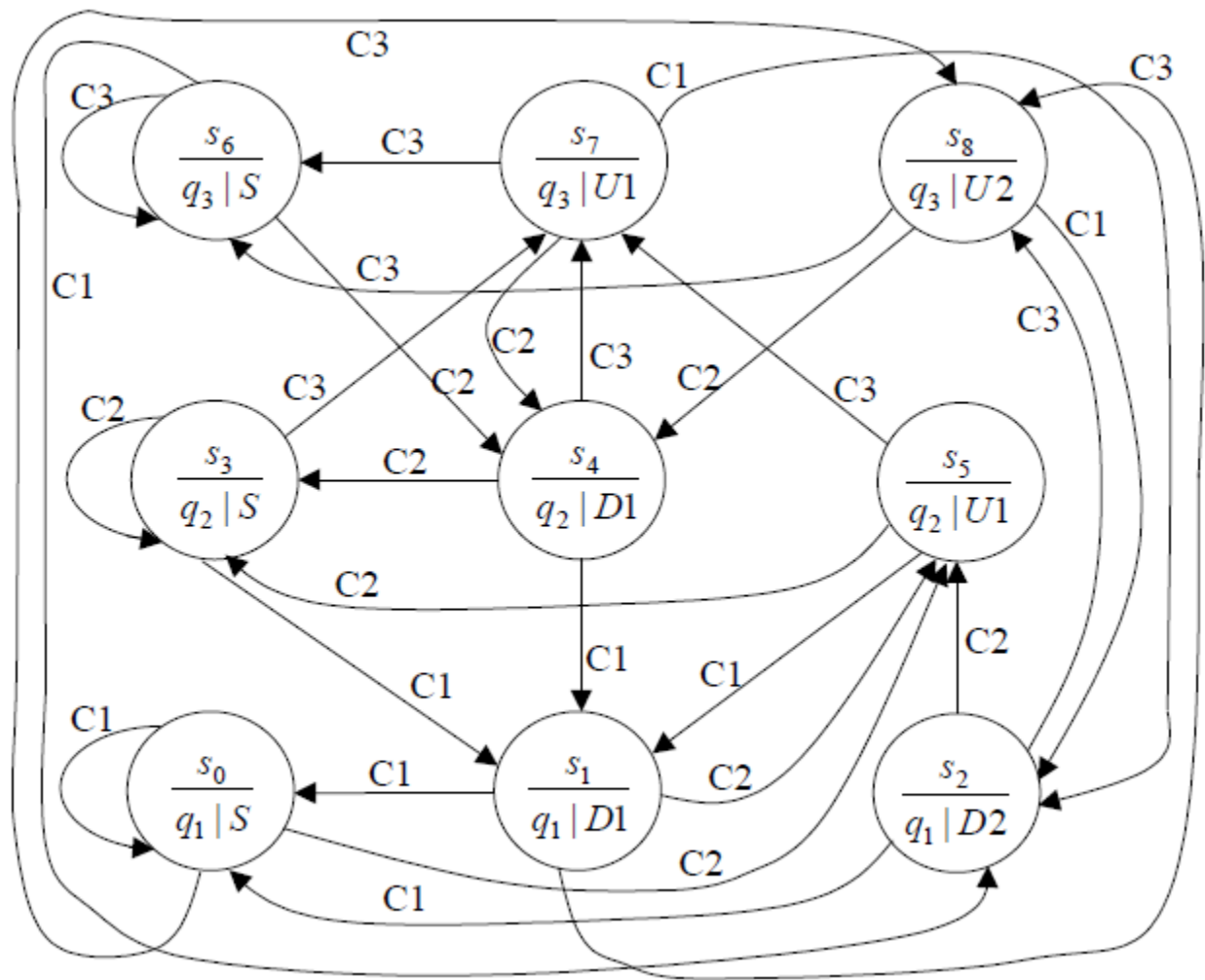
Mealy and Moore machine are equivalent

- mutual transformations:
- from Moore to Mealy – “move” output symbols from state into its incoming arcs
- from Mealy to Moore:
 - for each combination of state and output in Mealy machine create a new state of Moore machine
 - choose the transition state in Moore machine according to the combination of state and output in Mealy machine
 - choose as initial state in Moore machine any state corresponding to the initial state in Mealy machine

Equivalent Mealy machine



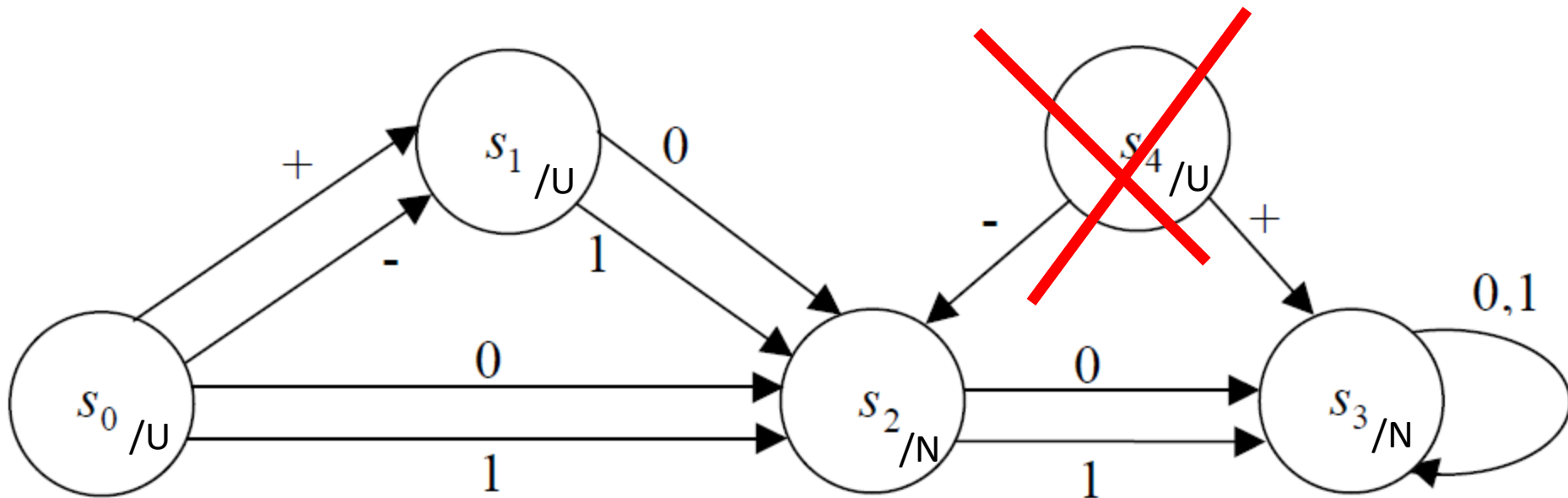
Equivalent Moore machine



Minimization of state machines

- remove states unreachable from the initial state
- represent all equivalent states as a single state of minimal machine
- basic minimization for Moore machine

Remove unreachable states



n-equivalence of states

- two states s_1 and s_2 are n -equivalent if for any input word σ of length n , output words of machines, starting in these states, coincide
- two states s_1 and s_2 are equivalent if they are n -equivalent for any nonnegative integer n
- 0-equivalent states have the same output (are recognized by empty word)

Basic minimization theorem

- Within a state machine with m states, two states are equivalent if and only if they are $(m - 2)$ -equivalent.
- Algorithm of minimization: create a sequence of state partitionings into 0, 1, 2 ... equivalent classes until no class splitting is observed

Minimization example

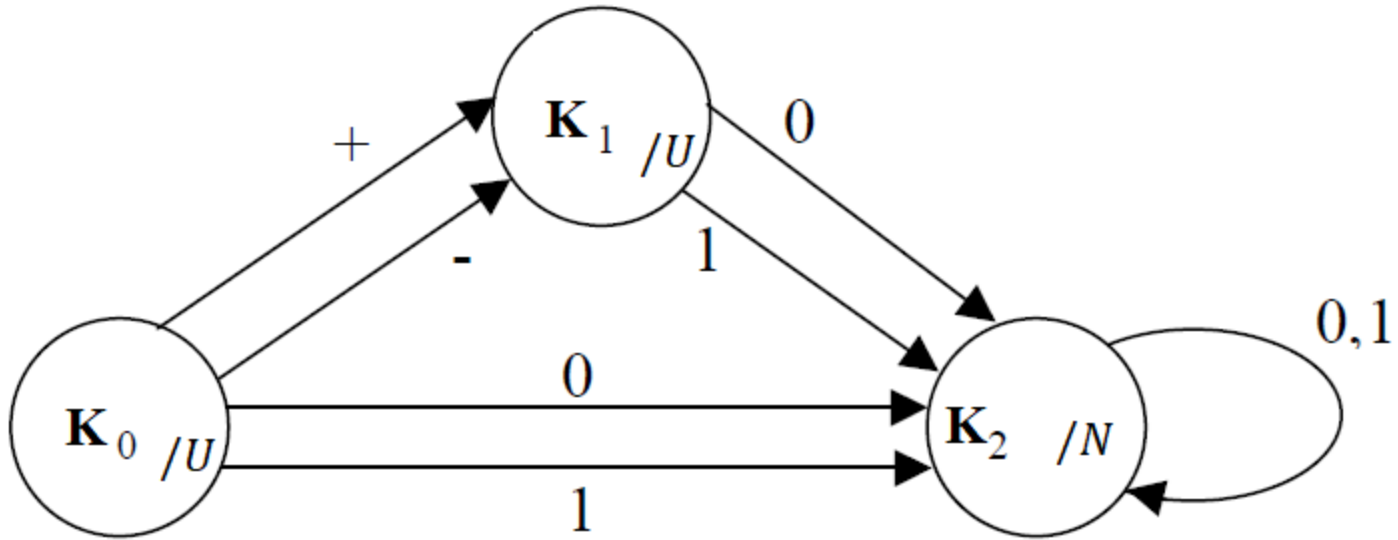
0-equivalent

Class	State	+	-	0	1
K0	s_0	K0	K0	K1	K1
	s_1			K1	K1
K1	s_2			K1	K1
	s_3			K1	K1

1-equivalent

Class	State	+	-	0	1
K0	s_0	K1	K1	K2	K2
K1	s_1			K2	K2
K2	s_2			K2	K2
	s_3			K2	K2

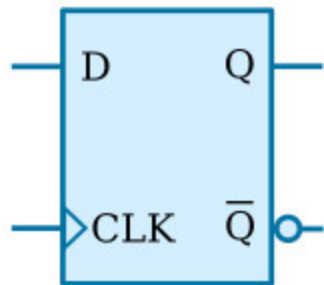
Minimal machine



Elementary automata – flip-flop

- store one bit: 0 or 1
- D delay
- T toggle
- RS set-reset
- JK Jack Kilby

D-flip-flop

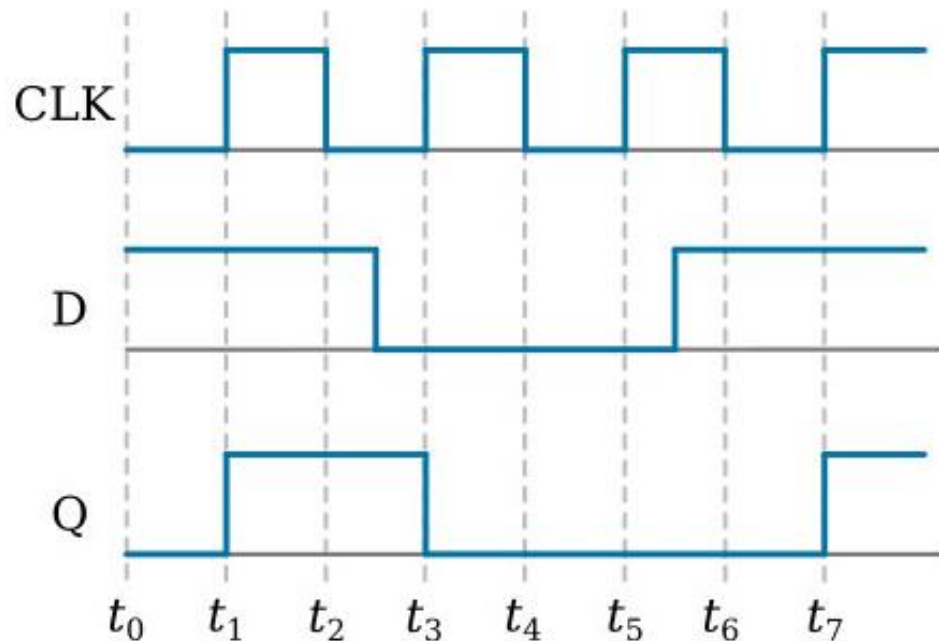


CLK	D	Q_{next}	Comment
Rising edge	0	0	Store 0
Rising edge	1	1	Store 1
Non-rising	X	Q	No change

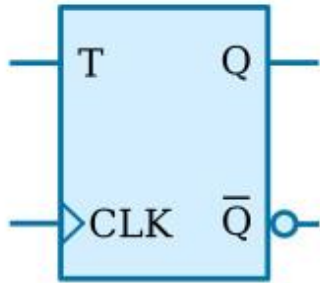
Q_{next} - "after the clock transition" output

Q - the current output

X - the signal is irrelevant

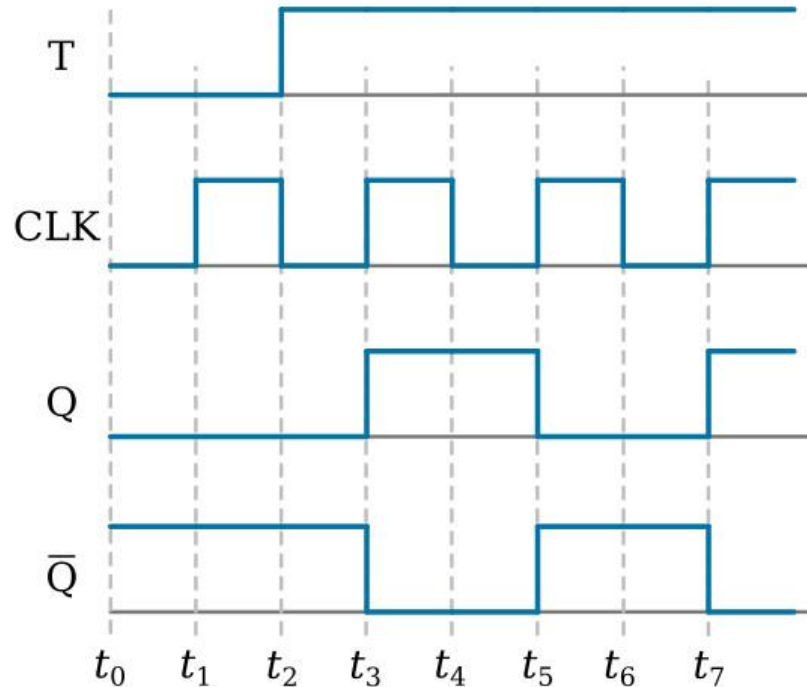


T-flip-flop

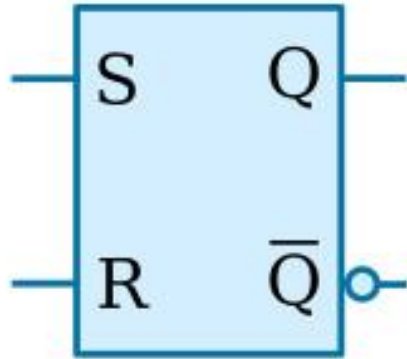


Truth table			
CLK	T	Q_{next}	Comment
Rising edge	0	Q	Hold state
Falling edge	0	Q	Hold state
Rising edge	1	\bar{Q}	Toggle
Falling edge	1	Q	No change

Q_{next} - "after the clock transition" output
 Q - the current output



RS-flip-flop

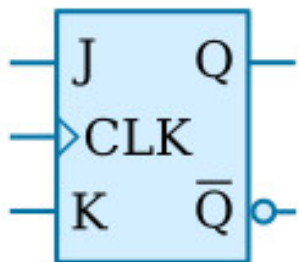


Characteristic table				
S	R	Q	\bar{Q}	Condition
0	0	*	*	latched
0	1	0	1	reset
1	0	1	0	set
1	1	X	X	jammed

* means previous state

X means 0 or 1

JK-flip-flop

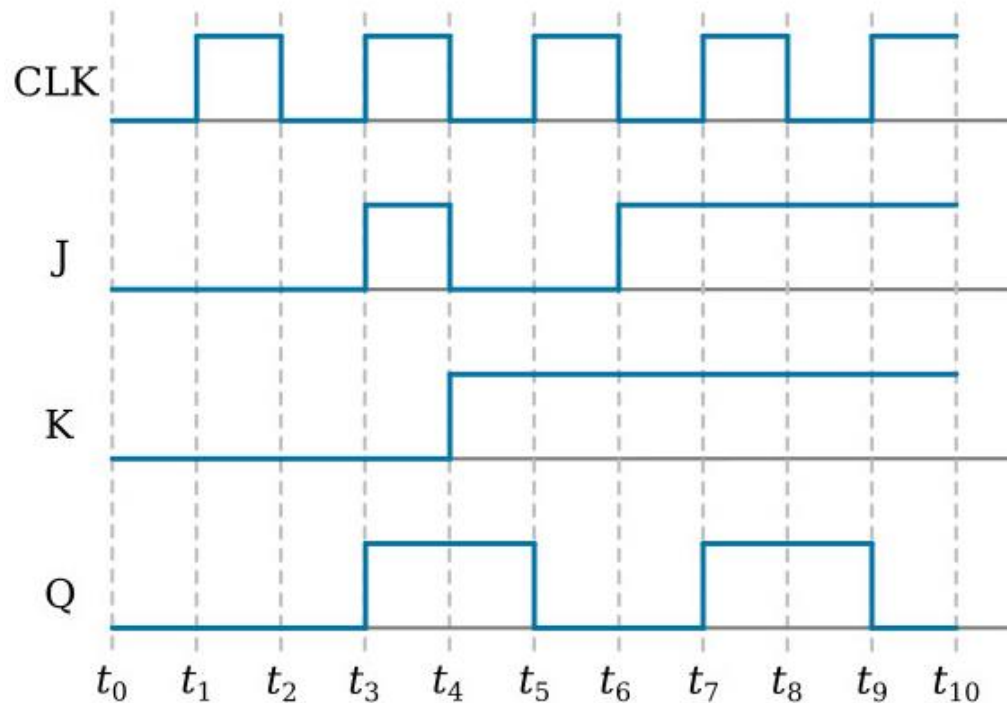


CLK	J	K	Q_{next}	Comment
Rising edge	0	0	Q	Hold state
Rising edge	0	1	0	Reset
Rising edge	1	0	1	Set
Rising edge	1	1	\bar{Q}	Toggle
Non-rising	X	X	Q	No change

Q_{next} - "after the clock transition" output

Q - the current output

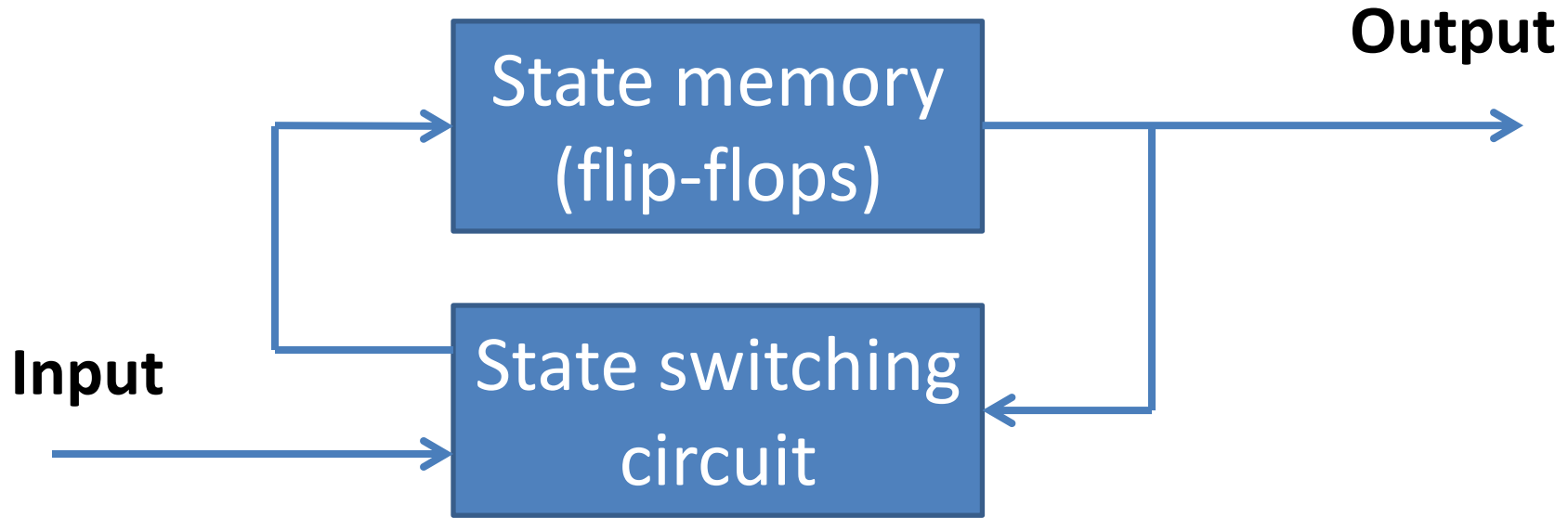
X - the signal is irrelevant



Synthesis of automata

- Verbal specification
- State diagram
- Enumerate inputs, outputs, and states
- Binary encoding of inputs, outputs, and states
- Choose flip-flops (for example D)
- Build truth table
- Draw layout with two combinatorial circuits – for switching states and producing output (same as state for Moore machine)

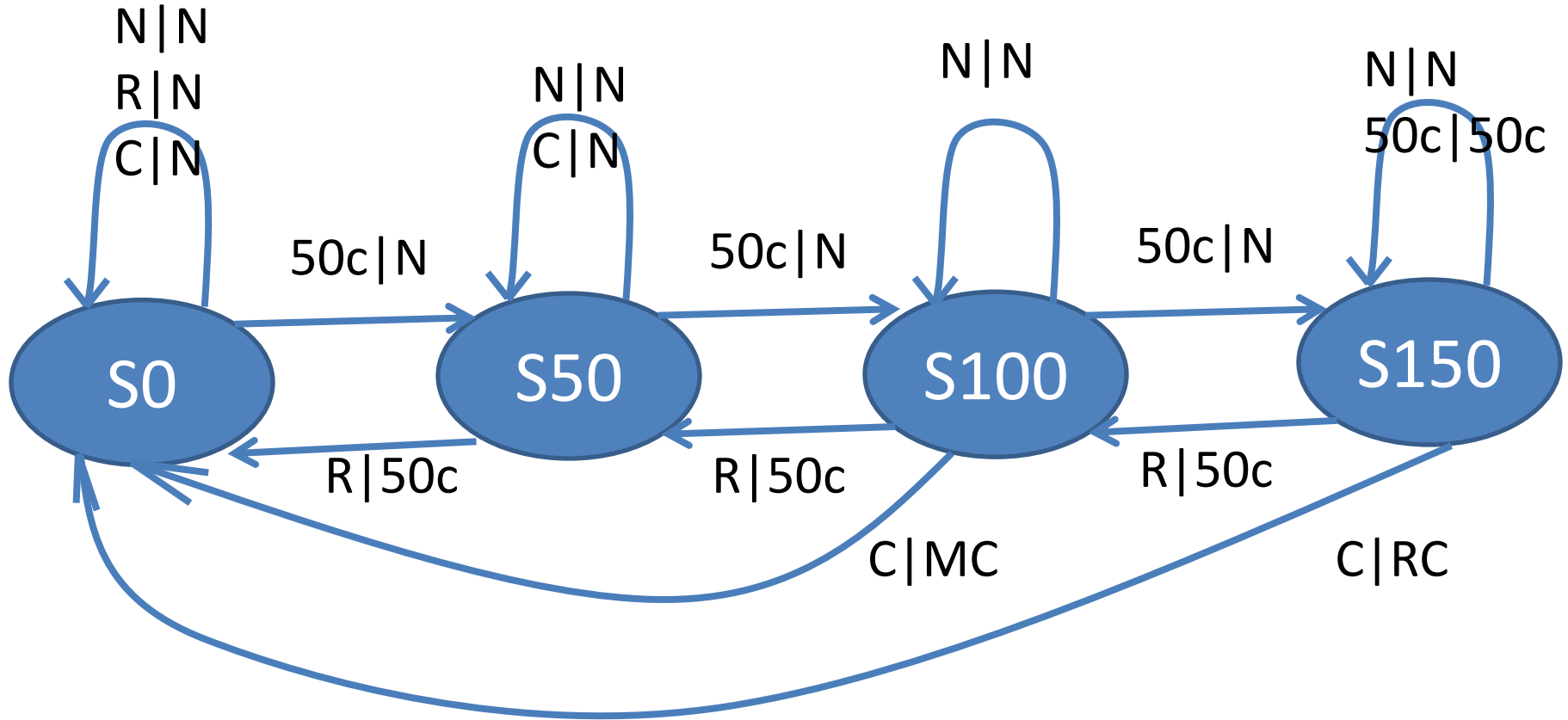
General layout of sequential logic



Chocolate vending machine

- Input: 50c, R, C, N
- Output: milk chocolate (1e), rum chocolate (1.5e), 50c, N
- State: s0, s50, s100, s150

State diagram



Binary encoding of inputs, outputs, and states

Input	Code	
	x1	x0
N	0	0
C	0	1
50c	1	0
R	1	1

Output	Code	
	y1	y0
N	0	0
MC	0	1
50c	1	0
RC	1	1

State	Code	
	q1	q0
q0	0	0
q50	0	1
q100	1	0
q150	1	1

Truth table

q1	q0	x1	x0	q1	q0	y1	y0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	1	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	1	0	0
0	1	0	1	0	1	0	0
0	1	1	0	1	0	0	0
0	1	1	1	0	0	1	0
1	0	0	0	1	0	0	0
1	0	0	1	0	0	0	1
1	0	1	0	1	1	0	0
1	0	1	1	0	1	1	0
1	1	0	0	1	1	0	0
1	1	0	1	0	0	1	1
1	1	1	0	1	1	1	0
1	1	1	1	1	0	1	0

Boolean functions

	q_1			
$q \backslash x$	00	01	11	10
00				
01				1
11	1		1	1
10	1			1

	q_0			
$q \backslash x$	00	01	11	10
00				1
01	1	1		
11	1			1
10			1	1

$$q_1 = q_1 \bar{x}_0 \vee q_1 q_0 x_1 \vee q_0 x_1 \bar{x}_0$$

$$q_0 = \bar{q}_0 x_1 \bar{x}_0 \vee \bar{q}_1 q_0 \bar{x}_1 \vee q_1 q_0 \bar{x}_0 \vee q_1 \bar{q}_0 x_1$$

	y_1			
$q \backslash x$	00	01	11	10
00				
01			1	
11		1	1	1
10			1	

	y_0			
$q \backslash x$	00	01	11	10
00				
01				
11		1		
10		1		

$$y_1 = q_0 x_1 x_0 \vee q_1 q_0 x_1 \vee q_1 x_1 x_0 \vee q_1 q_0 x_0$$

$$y_0 = q_1 \bar{x}_1 x_0$$