Digital systems and basics of electronics

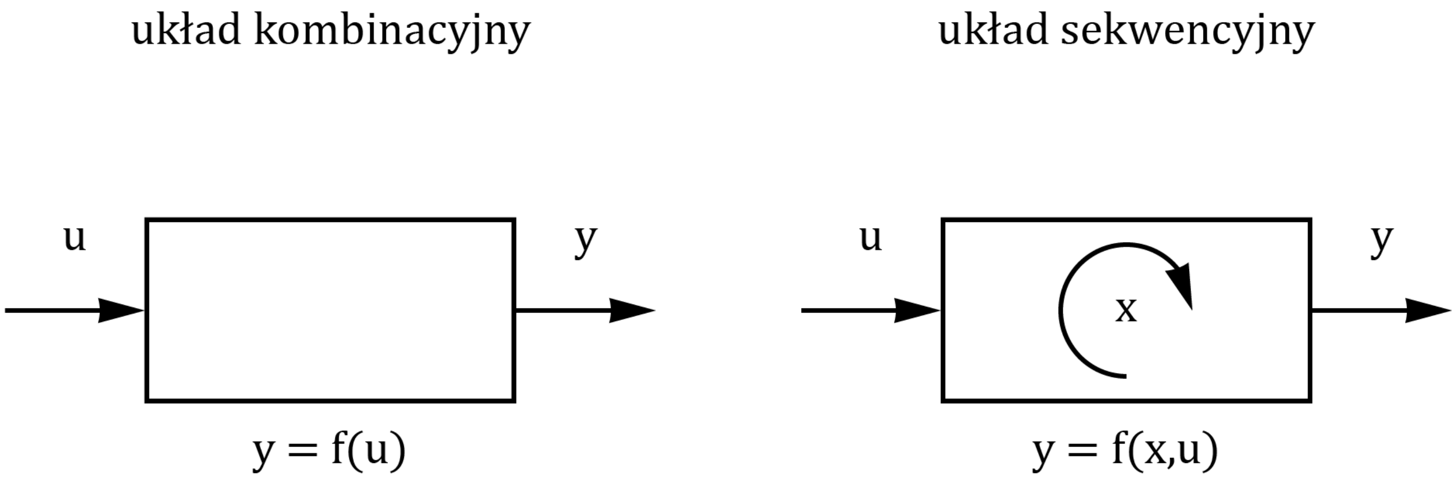
Adam Szmigielski

aszmigie@pjwstk.edu.pl

materials: $ftp(public)://aszmigie/SYC/ENG$

Designing of Sequential Circuits - lecture 10

Combinational and sequential circuits

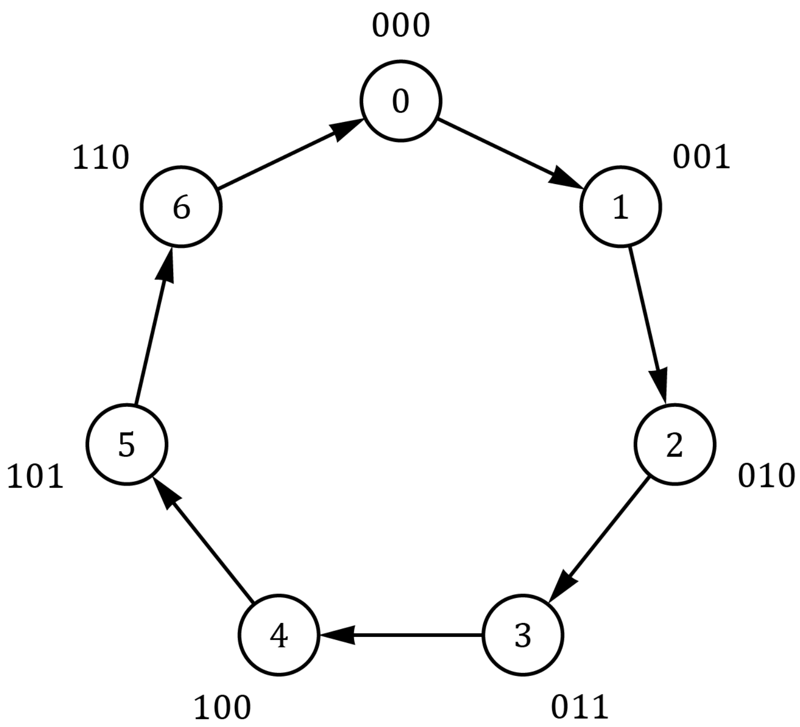


Internal state

W *sequential circuit* there are *internal states*.

- *Internal states* or (shorter *states*) "keep within limits"possibility of system change (evolution), because the future value of state depends on current state and value of inputs,
- System behavior is modelled by *directed graph* - with each *node* is associated *state* and *value of outputs*, each *arrow* determinates *next state* under the influence of current *input*,
- With each *internal state* are associated value of outputs,
- It is possible that *state* can not be measured externally (IN/OUT model of system),
- In *sequential circuits* state is modelled by *flip-flops* and combinational elements.

State diagram - transfer graph



Directed graph models behavior of sequential circuit ($mod7$ counter).

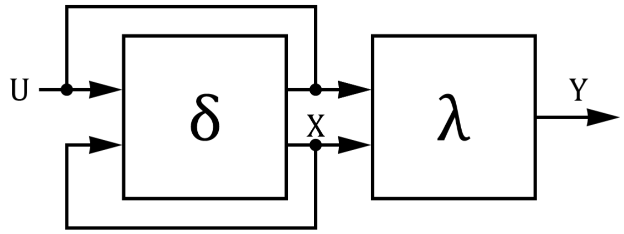
State codding

S_t	S_{t+1}	Y_t
S_0	S_1	0
S_1	S_2	1
S_2	S_3	2
S_3	S_4	3
S_4	S_5	4
S_5	S_6	5
S_6	S_0	6

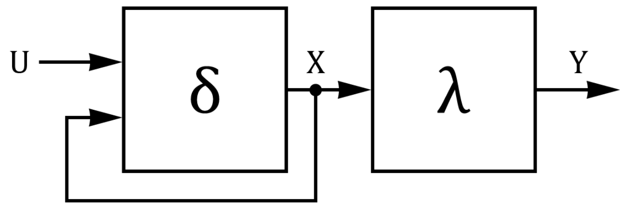
codding	S_t	S_{t+1}	Y_t
$S_0 \to 000$	000	001	000
$S_1 \to 001$	001	010	001
$S_2 \to 010$	010	011	010
$S_3 \to 011$	011	100	011
$S_4 \to 100$	100	101	100
$S_5 \to 101$	101	110	101
$S_6 \to 110$	110	000	110

Mealy's and Moore's state machines

automat Mealy'ego



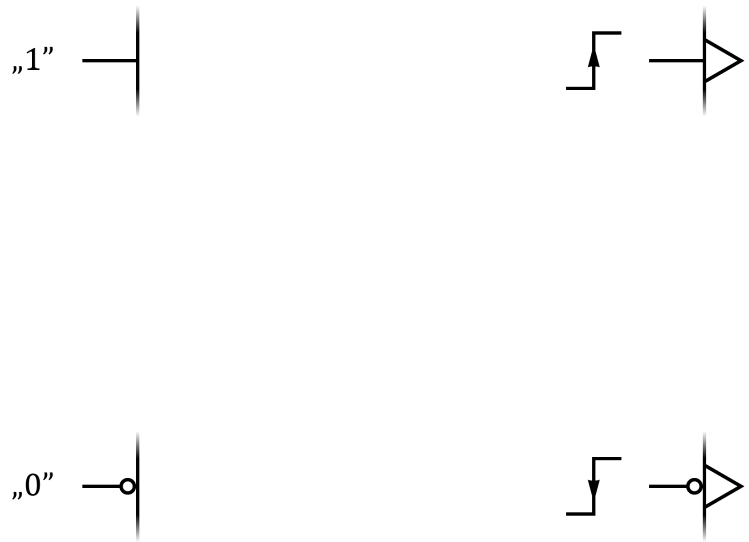
automat Moore'a



Circuit (λ) realizing *output function* is a combinational circuit and block (δ) realizes memory (*excitation function*).

$$\begin{cases} x(k+1) = \delta(x(k), u(k)) & \text{state equation: } \delta \text{ - } \textit{excitation function} \\ y(k) = \lambda(x(k), u(k)) & \text{output equation: } \lambda \text{ - } \textit{output function} \end{cases}$$

Clock



- The sequence of states is important in sequential circuits,
- Additional signal called *clock* is used to synchronize changes,
- Clock activated with *positive slope* are mainly used.

Designing of Sequential Circuits

Design stages

We focus on *Moore's state machine* design.
Main steps of system synthesis:

1. **Step 1:** Diagram design - directed graph models system behavior. *Node* represents *state* and *value of outputs*, each *arrow* determinates *next state* under the influence of current *input*. *Deterministic state machine* for each state must take into account all combination of inputs.
2. **Step 2:** State codding - use the smallest number of variables (1 bit 1 flip-flop).
3. **Step 3:** Excitation function - for each flip-flop we need to find signal given on flip-flop inputs.
4. **Step 4:** Output function - for each output draw the Karnough map and find output function - mapping *internal stages* into outputs.

Example

Task:

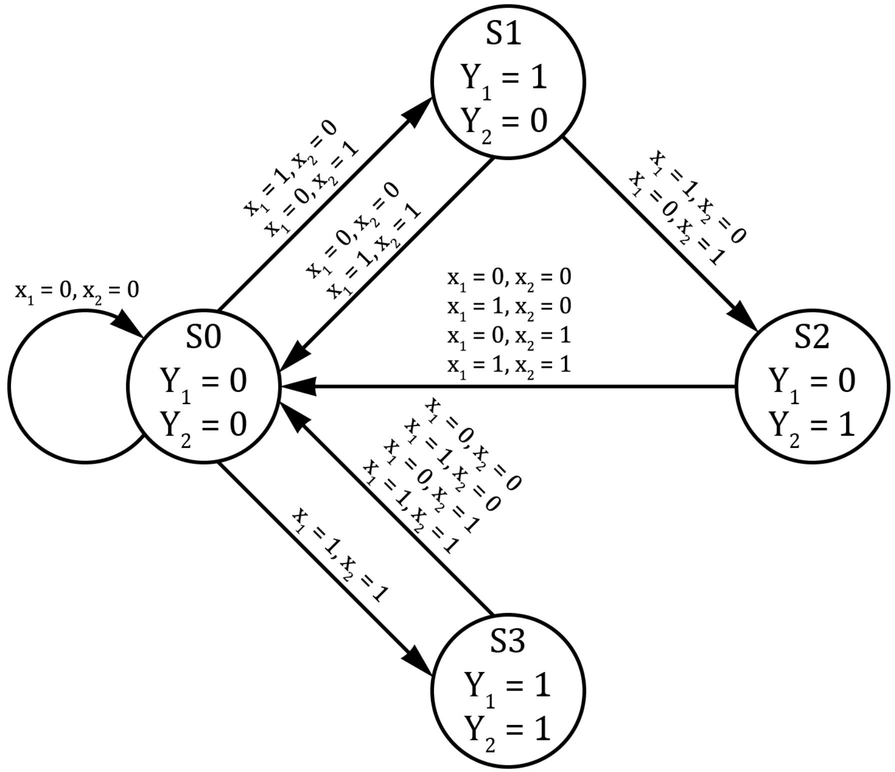
Design two inputs x_1, x_2 and two outputs y_1, y_2 sequential circuit working as follow:

1. If both inputs are ones $(x_1 = 1, x_2 = 1)$ then both outputs y_1, y_2 periodically shout be switched on and off (sequence $00 \to 11 \to 00$),
2. Two zeros on inputs $(x_1 = 0, x_2 = 0)$ should switch off outputs $(y_1 = 0, y_2 = 0)$,
3. The rest input combinations cause that outputs follow $mod3$ counter.

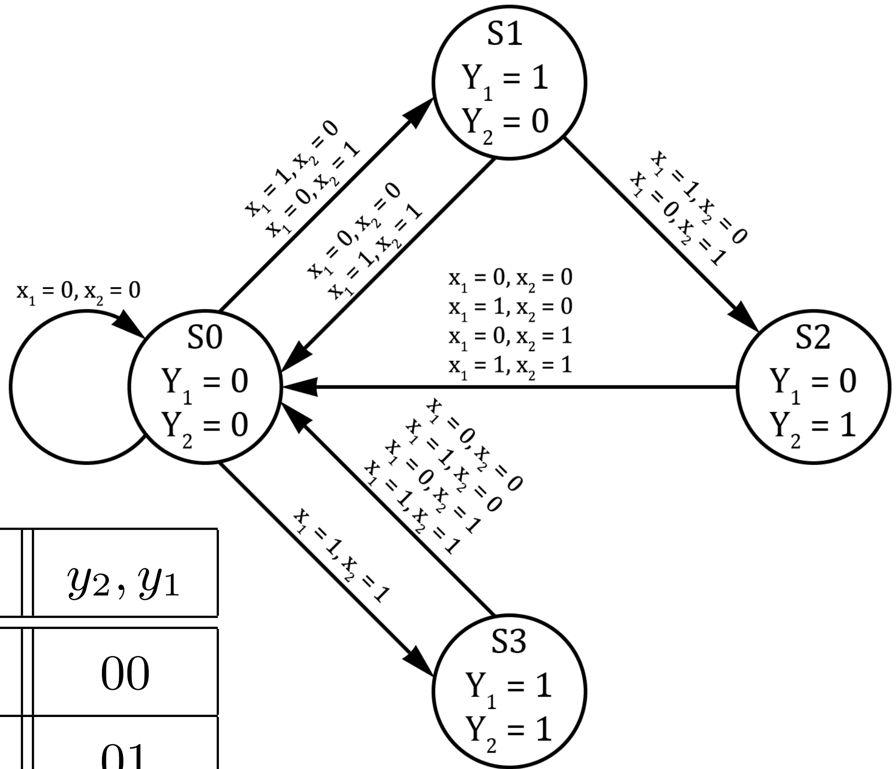
Determination of states - state diagram

Because outputs y_1, y_2 can reach all combinations (00, 01, 10 and 11) we need at least 4 states.

For state S_0: $y_1 = 0$ i $y_2 = 0$
For state S_1: $y_1 = 0$ i $y_2 = 1$
For state S_2: $y_1 = 1$ i $y_2 = 0$
For state S_3: $y_1 = 1$ i $y_2 = 1$



State table and state diagram



S^{t+1} :

$S^t \backslash x_2, x_1$	00	01	11	10	y_2, y_1
S_0	S_0	S_1	S_3	S_1	00
S_1	S_0	S_2	S_0	S_2	01
S_2	S_0	S_0	S_0	S_0	10
S_3	S_0	S_0	S_0	S_0	11

State codding table

S^{t+1} :

$S^t \backslash x_2, x_1$	00	01	11	10	y_2, y_1
S_0	S_0	S_1	S_3	S_1	00
S_1	S_0	S_2	S_0	S_2	01
S_2	S_0	S_0	S_0	S_0	10
S_3	S_0	S_0	S_0	S_0	11

Q_1^{t+1}, Q_0^{t+1} :

$\frac{x_2, x_1}{Q_1^t, Q_0^t}$	00	01	11	10	y_2, y_1
$S_0 \mapsto 00$	00	01	10	01	00
$S_1 \mapsto 01$	00	11	00	11	01
$S_2 \mapsto 11$	00	00	00	00	10
$S_3 \mapsto 10$	00	00	00	00	11

Excitation function **for D-type flip-flops**

Q_1^{t+1}, Q_0^{t+1} :

$\frac{x_2,x_1}{Q_1^t,Q_0^t}$	00	01	11	10	y_2, y_1
$S_0 \mapsto 00$	00	01	10	01	00
$S_1 \mapsto 01$	00	11	00	11	01
$S_2 \mapsto 11$	00	00	00	00	10
$S_3 \mapsto 10$	00	00	00	00	11

D_1 :

$\frac{x_2,x_1}{Q_1^t,Q_0^t}$	00	01	11	10
00	0	0	1	0
01	0	1	0	1
11	0	0	0	0
10	0	0	0	0

D_0 :

$\frac{x_2,x_1}{Q_1^t,Q_0^t}$	00	01	11	10
00	0	1	0	1
01	0	1	0	1
11	0	0	0	0
10	0	0	0	0

$D_1 = \overline{Q_1} Q_0 \overline{x_2} x_1 + \overline{Q_1 Q_0} x_2 x_1 + \overline{Q_1} Q_0 x_2 \overline{x_1}$ $\quad D_0 = \overline{Q_1}\, \overline{x_2} x_1 + \overline{Q_1} x_2 \overline{x_1}$

Output functions

y_2, y_1 :

$\frac{x_2, x_1}{Q_1^t, Q_0^t}$	00	01	11	10	y_2, y_1
$S_0 \mapsto 00$	00	01	10	01	00
$S_1 \mapsto 01$	00	11	00	11	01
$S_2 \mapsto 11$	00	00	00	00	10
$S_3 \mapsto 10$	00	00	00	00	11

y_2 :

$Q_1 \backslash Q_0$	0	1
0	0	0
1	1	1

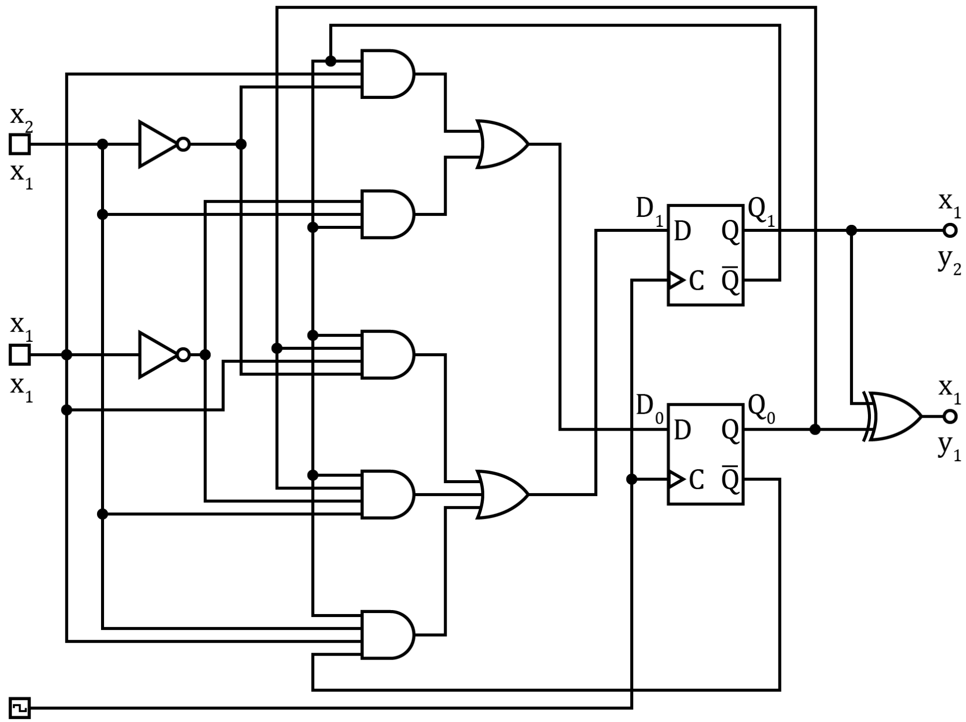
$y_2 = Q_1$

y_1 :

$Q_1 \backslash Q_0$	0	1
0	0	1
1	1	0

$y_1 = \overline{Q_1} Q_0 + Q_1 \overline{Q_0} = Q_1 \otimes Q_0$

Realization of sequential circuit



- Excitation function: $D_0 = \overline{Q_1}\,\overline{x_2}x_1 + \overline{Q_1}x_2\overline{x_1}$
  $D_1 = \overline{Q_1}Q_0\overline{x_2}x_1 + \overline{Q_1Q_0}x_2x_1 + \overline{Q_1}Q_0x_2\overline{x_1}$
- Output function: $y_2 = Q_1$
  $y_1 = \overline{Q_1}Q_0 + Q_1\overline{Q_0} = Q_1 \otimes Q_0$

State codding

The proper choice of codding is important to obtain simple *excitation function.* Unfortunately there is no general method to minimize *excitation function.* We can try to do that follow the rules (according to priority):

- **Rule 1:** - Codes of states, which have the same successor (next state), should differ only one bit,
- **Rule 2:** - Codes of states, which have the same predecessor (previously state), should differ only one bit,
- **Rule 3:** - Codes of states, which have the same outputs under the influence of the same inputs, should differ only one bit.

State minimization - equivalence of states

Two states are *equivalent* (and can be replace with one state) if the follow conditions are satisfied:

1. Outputs, associated with two states, are the same,
2. All successors of these two states are the same or *equivalent*.

State minimization - example

S^{t+1} :

$S^t \backslash x_2, x_1$	00	01	11	10	Y - wyjście
1	5	3	2	1	0
2	5	3	1	4	0
3	3	4	4	5	1
4	5	3	2	2	0
5	6	7	1	1	0
6	3	3	1	7	0
7	7	1	1	5	1

- There are not identical states,
- Candidates for *equivalent sates*: $\{1,2\}$, $\{1,4\}$, $\{1,5\}$, $\{1,6\}$, $\{2,4\}$, $\{2,5\}$, $\{2,6\}$, $\{3,7\}$, $\{4,5\}$, $\{4,6\}$, $\{5,6\}$,
- *Equivalent states*: 3 with 7 and states: 1, 2, 4, 5 and 6.

State minimization - choice of equivalent states

$S^t \backslash x_2, x_1$	00	01	11	10	Y - output
1	5	3	2	1	0
2	5	3	1	4	0
3	3	4	4	5	1
4	5	3	2	2	0
5	6	7	1	1	0
6	3	3	1	7	0
7	7	1	1	5	1

candidates	second condition of equ.	Non-equivalence	1 iteration	2 iteration
$\{1, 2\}$	$\{1, 2\}$, $\{1, 4\}$			
$\{1, 4\}$	$\{1, 2\}$			
$\{1, 5\}$	$\{5, 6\}$, $\{3, 7\}$, $\{1, 2\}$	Non-equivalent		see $\{5, 6\}$
$\{1, 6\}$	$\{3, 5\}$, $\{1, 2\}$, $\{1, 7\}$	Non-equivalent	see $\{3, 5\}$	
$\{2, 4\}$	$\{1, 2\}$, $\{2, 4\}$			
$\{2, 5\}$	$\{5, 6\}$, $\{3, 7\}$, $\{1, 4\}$	Non-equivalent		see $\{5, 6\}$
$\{2, 6\}$	$\{3, 5\}$, $\{4, 7\}$	Non-equivalent	see $\{3, 5\}$	
$\{3, 7\}$	$\{3, 7\}$, $\{1, 4\}$			
$\{4, 5\}$	$\{5, 6\}$, $\{3, 7\}$, $\{1, 2\}$	Non-equivalent		see $\{5, 6\}$
$\{4, 6\}$	$\{3, 5\}$, $\{1, 2\}$, $\{2, 7\}$	Non-equivalent	see $\{3, 5\}$, $\{2, 7\}$	
$\{5, 6\}$	$\{3, 6\}$, $\{3, 7\}$, $\{1, 7\}$	Non-equivalent	see $\{3, 6\}$, $\{1, 7\}$	

- A - states $\{1, 2, 4\}$ B - states $\{3, 7\}$ C - state $\{5\}$ D - state $\{6\}$

State minimization - reduced table

- A - stats $\{1, 2, 4\}$ B - stats $\{3, 7\}$ C - state $\{5\}$ D - state $\{6\}$

$S^t \backslash x_2, x_1$	00	01	11	10	Y - output
1	5	3	2	1	0
2	5	3	1	4	0
3	3	4	4	5	1
4	5	3	2	2	0
5	6	7	1	1	0
6	3	3	1	7	0
7	7	1	1	5	1

$S^t \backslash x_2, x_1$	00	01	11	10	Y - output
A	C	B	A	A	0
B	B	A	A	C	1
C	D	B	A	A	0
D	B	B	A	B	0

Systems fully not determined

In the case of system which are not fully determined

- The one or more of input combination is not determined - e.g. slide 11, point 3 when system would count $mod3$ only due to input combination $x_2x_1 = 10$ (for input combination $x_2x_1 = 01$ system would not be determined),
- During codding states we obtain redundant states - e.g. counter $mod3$ (needs 3 states, which can be realized with 2 flip-flops - total states 4, 1 redundant)

If state is not determined we can put instead that state any value we want.

Mealy's state machine - remarks

- In *Mealy's machine state* state is not unequivocally related to output,
- For that reason it is possible reduce the potential number of states at the cost of more complicated output function,
- System diagram should take into consideration input-output dependencies.

Task for laboratory

1. Design and realize traffic lights for vehicles. System should work in two modes:
   - *normal mode* - Input is equal 1. On output there is a sequence of lights
     red $\rightarrow$ *red + yellow* $\rightarrow$ *green* $\rightarrow$ *yellow*
   - *emergency mode* - 0 given on input causes **continuous yellow light on output**.
2. Design and realize traffic lights for vehicles. System should work in two modes:
   - *normal mode* - Input is equal 1. On output blinks as follow:
     red $\rightarrow$ *red + yellow* $\rightarrow$ *green* $\rightarrow$ *yellow*
   - *emergency mode* - 0 given on input causes **blinking yellow light on output**.

   Clock in emergency mode should be 8 times bigger then in normal.