



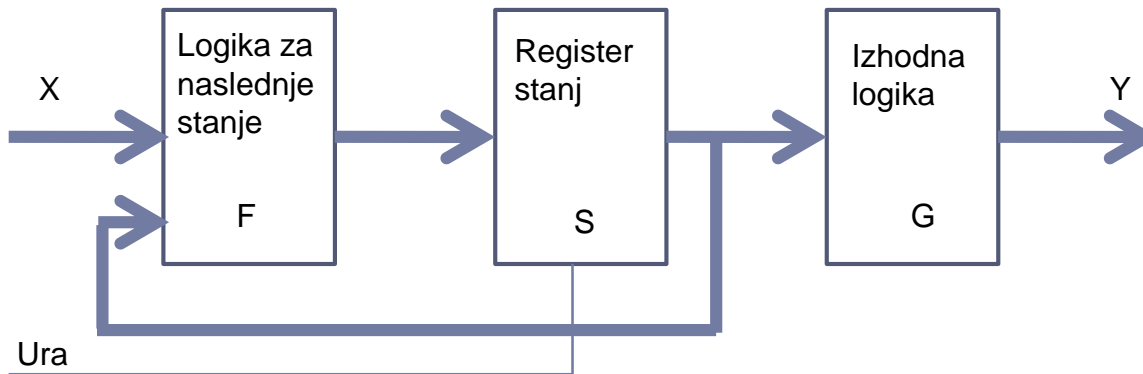
Digitalna vezja UL, FRI



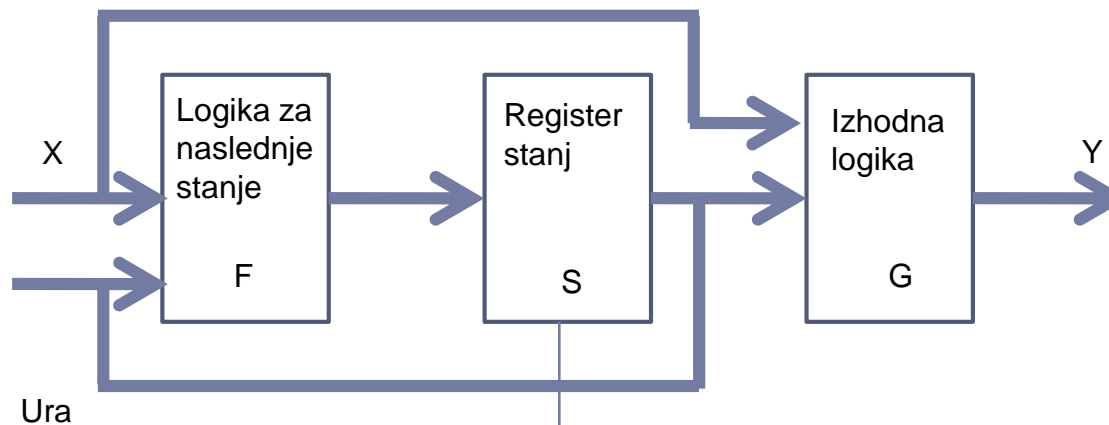
Vaja 10, Avtomati

Avtomat

Moore: $S(t+1) = F[X, S(t)], Y = G[S(t)]$



Mealy: $S(t+1) = F[X, S(t)], Y = G[X, S(t)]$



Vaja 1: Avtomat za razpoznavanje niza

Realizirati želimo avtomat za razpoznavanje desetiškega števila 9, ki se pojavlja v zaporednem nizu dvojiških vrednosti 0 in 1, če je spredaj vrednost 0. Ob pojavu števila 9 naj se izhod nastavi na 1, sicer pa je enak 0. Zahteve:

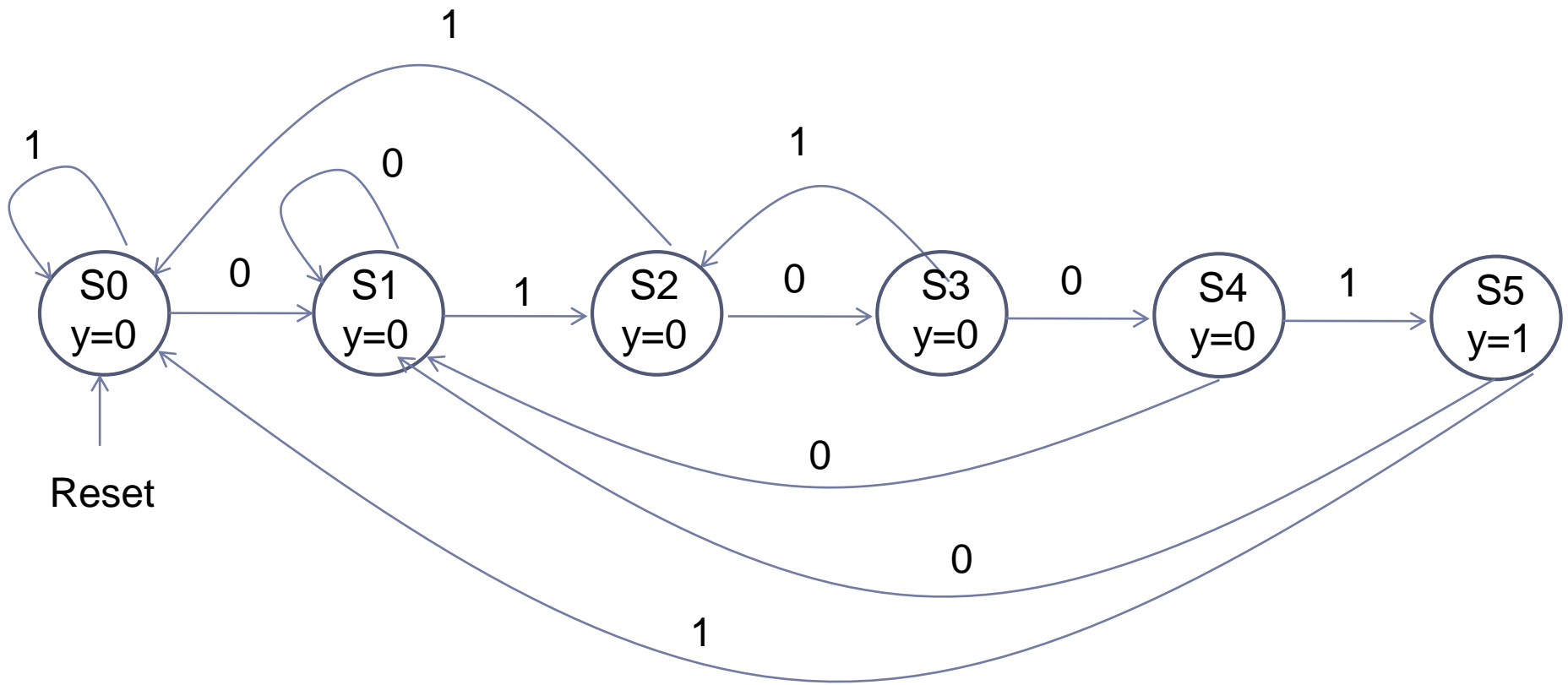
1. Določite vhodno množico (X), izhodno množico (Y) in množico stanj (S).
2. Narišite diagram prehajanja stanj
3. Zapišite tabelo prehajanja stanj
4. Realizirajte avtomat z D pomnilno celico in 4/1 MUXi
5. Preverite delovanje avtomata v logisimu z uporabo naslednjih vrednosti na vhodu:

$X=0100100111010010$

$Y=00000100000000010$

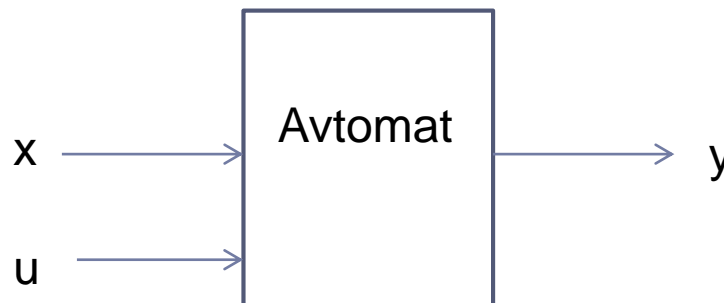
Rešitev:

- ▶ Množice: $X=(0,1)$, $Y=(0,1)$, $S=(S_0, S_1, S_2, S_3, S_4, S_5)$



Vaja 1:

- ▶ Realizirajte avtomat z enim vhodom – x in enim izhodom – y . Na vhodu je prisoten digitalni signal (0,1) z določenimi kombinacijami ničel in enic. Na izhod naj bo priključena lučka, ki zasveti ($y=1$), če se je pojavilo zaporedje 011.
- ▶ Za realizacijo uporabite dve JK pomnilni celici in logična vrata (NAND) za določanje vhodov za eno pomnilno celico ter 4/1 MUX za drugo pomnilno celico, izhod y pa realizirajte z logičnimi vrati (AND, OR, NOT).

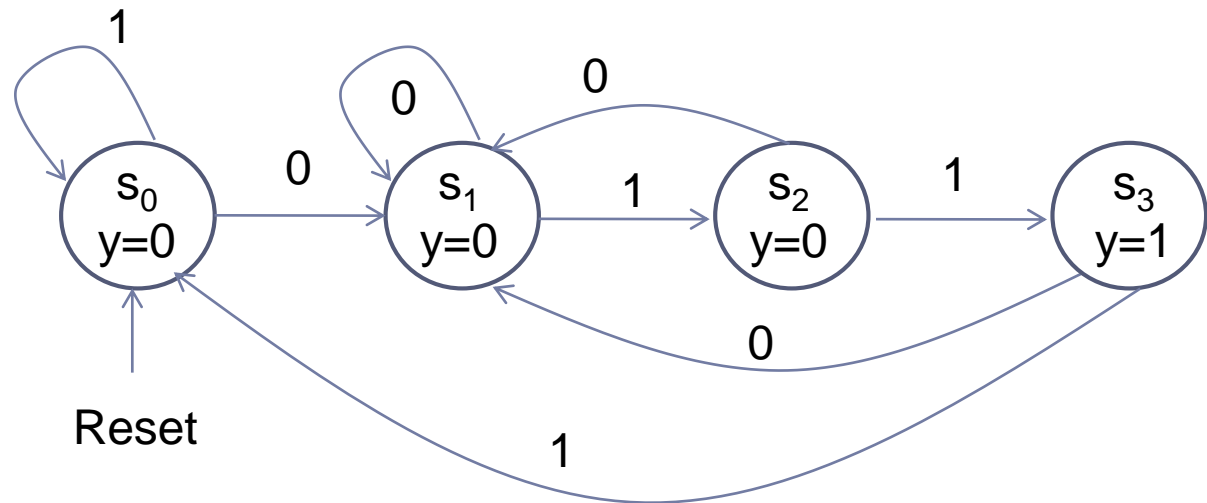


x : 1001101100101100

y : 0000100100000100

Vaja 1: Diagram prehajanja stanj

- ▶ Moore:
- ▶ Diagram prehajanja stanj za razpoznavanje niza 011 v zaporedju ničel in enic na vhodu. Izhod y naj se postavi na 1 vsakič, ko se pojavi iskani niz.



Vhod: 1001101100101100

Na vhodu se 3x pojavi niz 011

Izhod: 0000100100000100

Na izhodu = 1 ob vsakem pojavu niza 011

Vaja 1: Načrtovanje avtomata

► Kodiranje stanj:

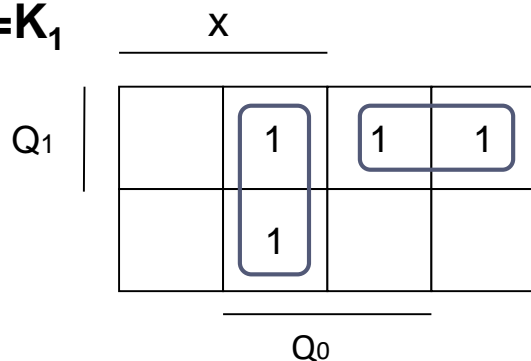
	Q_1	Q_0
s_0	0	0
s_1	0	1
s_2	1	0
s_3	1	1

► JK pomnilni celici:

	Q_1	Q_0	Q_1	Q_0	y	$J_1=K_1$	$J_0=K_0$
x	(t)	(t)	(t+1)	(t+1)			
0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	0
0	1	0	0	1	0	1	1
0	1	1	0	1	1	1	0
1	0	0	0	0	0	0	0
1	0	1	1	0	0	1	1
1	1	0	1	1	0	0	1
1	1	1	0	0	1	1	1

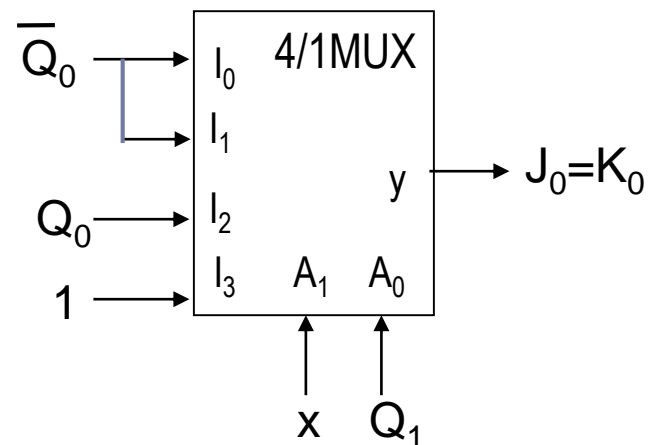
Vaja 1: Funkcije $J=K$ in izhod y

$J_1=K_1$

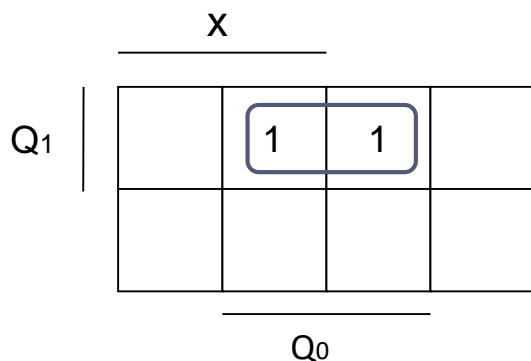


$$J_1 = K_1 = x \cdot Q_0 \vee \bar{x} \cdot Q_1 = (x \uparrow Q_0) \uparrow (\bar{x} \uparrow Q_1)$$

$J_0=K_0$

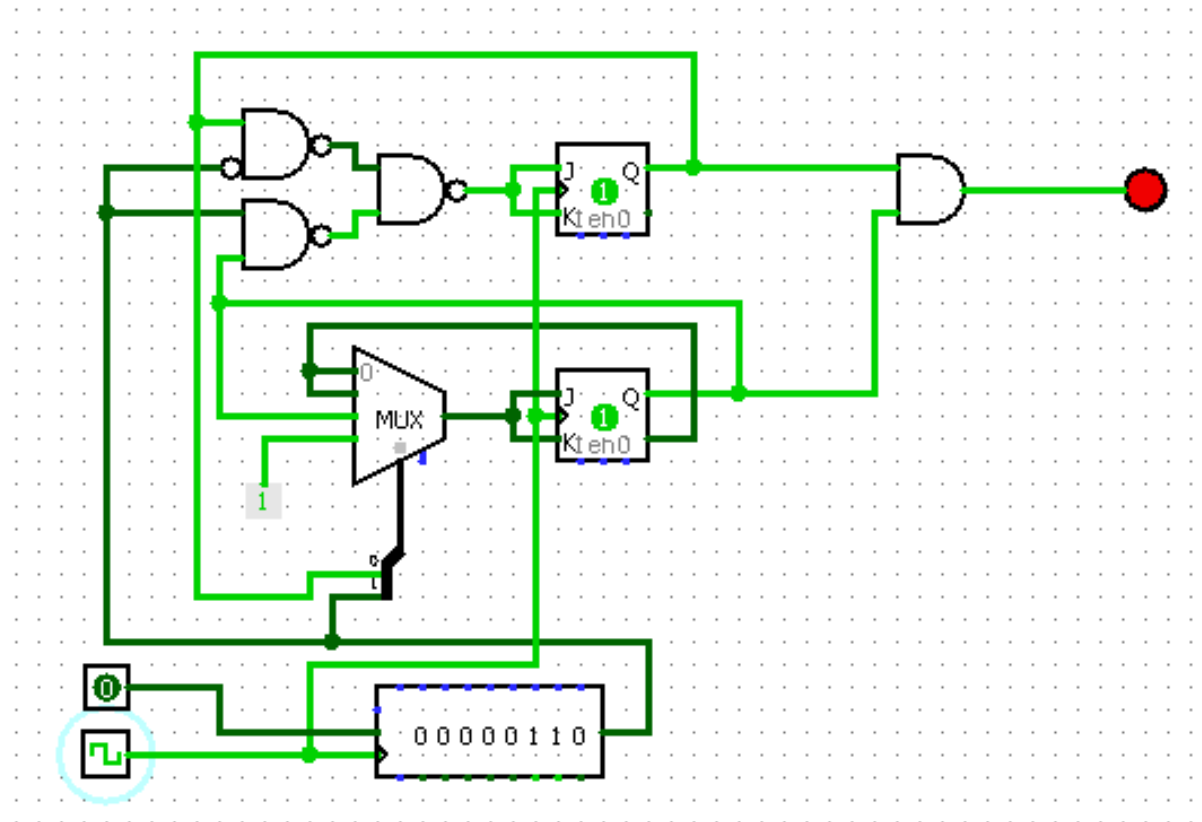


Izhod Y :



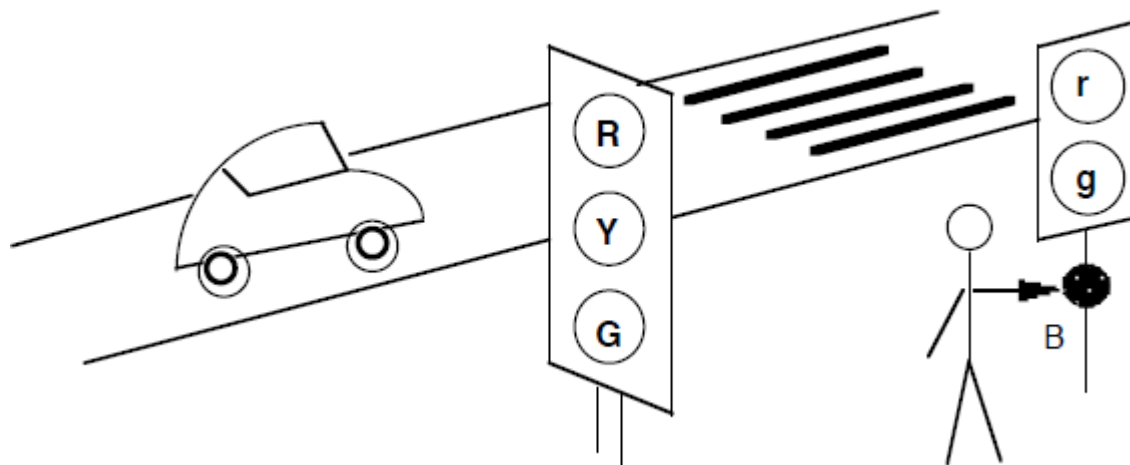
$$y = Q_1 \cdot Q_0$$

Vaja 1: Shema logičnega vezja



Vaja 2 Semafor

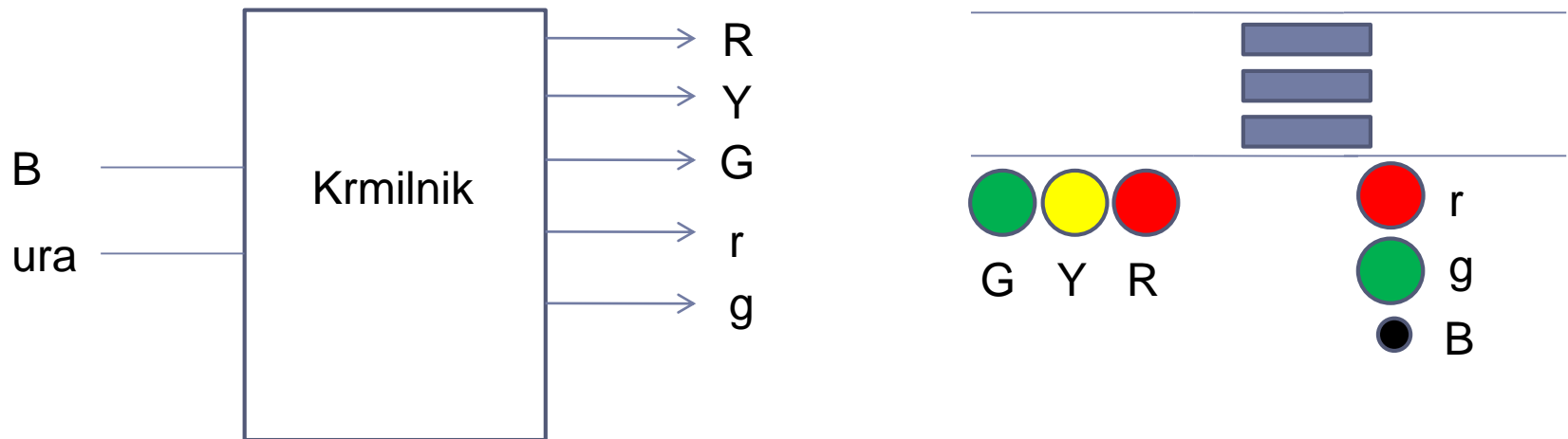
- Realizirajte krmilnik za semaforja avto-pešec, ki sta predstavljena na spodnji sliki:



- Krmilnik naj reagira na vhodni signal B, ki ga upravlja pešec.
- Pri realizaciji lahko
 - A) izпустite stanje, ko imata oba semaforja rdečo luč: Rg, Yr, Gr
 - B) stanja so: Rg, Rr, Yr, Gr
- Poenostavitev:
 - A) posamezne kombinacije luči lahko gorijo enak časovni interval.
 - B) na semaforju za pešca naj gori zelena luč 40 sec
- Pri določanju stanj pazite na varnost pešcev !

Semafor

- R – rdeča luč avto, Y – rumena luč avto, G – zelena luč avto
- r – rdeča luč pešec, g – zelena luč pešec
- B – gumb (1 – pritisnjen, 0 – ni pritisnjen)



- Narišite diagram prehajanja stanj in izpolnite aplikacijsko tabelo.
- Pri realizaciji uporabite dve JK pomnilni celici. Za določanje vhodov prve pomnilne celice uporabite 4/1 MUX, za določanje vhodov druge pomnilne celice logična vrata NAND.
- Funkcije za prižiganje luči realizirajte z logičnimi vrati (AND, OR, NOT).