Digitalna vezja
UL, FRI

P9 Avtomati

DV Trebar

Avtomat - Končni stroj stanj

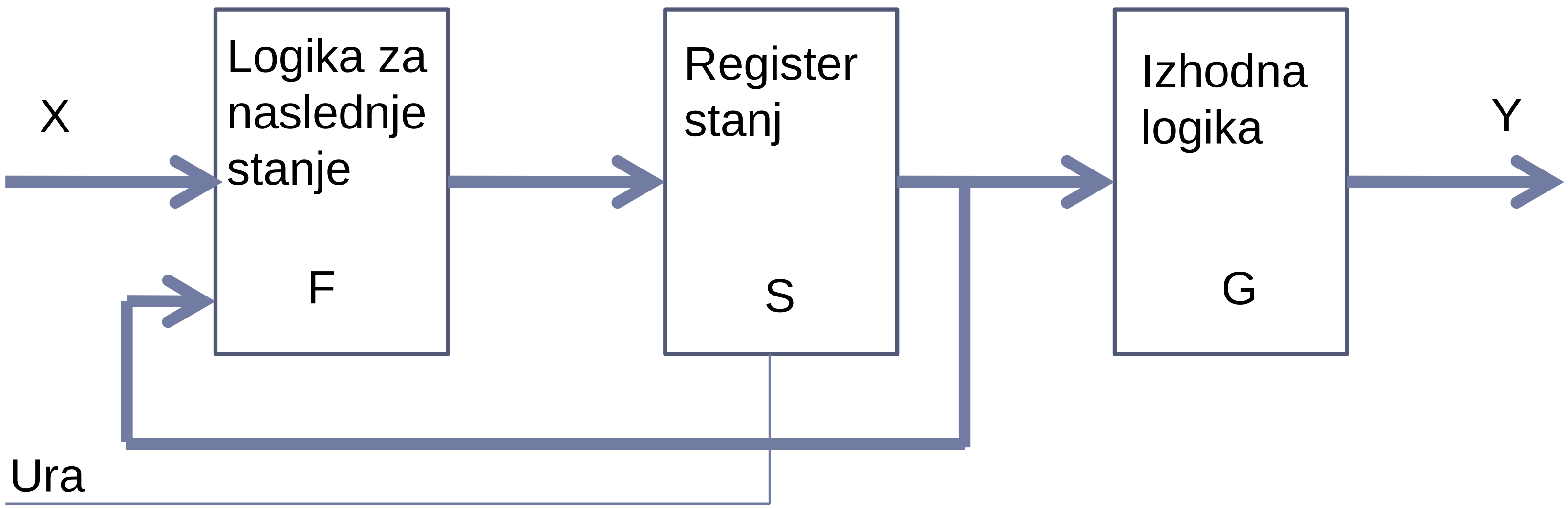
- Je sekvenčno vezje (stroj) s končnim številom stanj, podano z vnaprej določenim zaporedjem prehajanja med stanji.
- Definicija:
  - X - množica vhodov (n)
  - Y - množica izhodov (m)
  - S – množica stanj (p)
    [S(t)-trenutna stanja (TS), S(t+1)- naslednja stanja (NS)]
  - logični funkciji F (določa naslednje stanje) in G (določa izhode)
- Delovanje:
  - Logična funkcija F določa naslednje stanje (NS) in je odvisna od trenutnega stanja in vhodov.
  - Izhodna logična funkcija G je odvisna samo od trenutnega stanja (TS), ali pa od trenutnega stanja (TS) in vhodov.
- Predstavitev:
  - Diagram prehajanja stanj
  - Tabela prehajanja stanj
- Tip:
  - Moore
  - Mealy

Moore-ov avtomat

- Logika za naslednje stanje s funkcijo F v odvisnosti od vhodov (X) in trenutnega stanja (S(t)) določa naslednje stanje (S(t+1)), ki se shrani v register stanj.

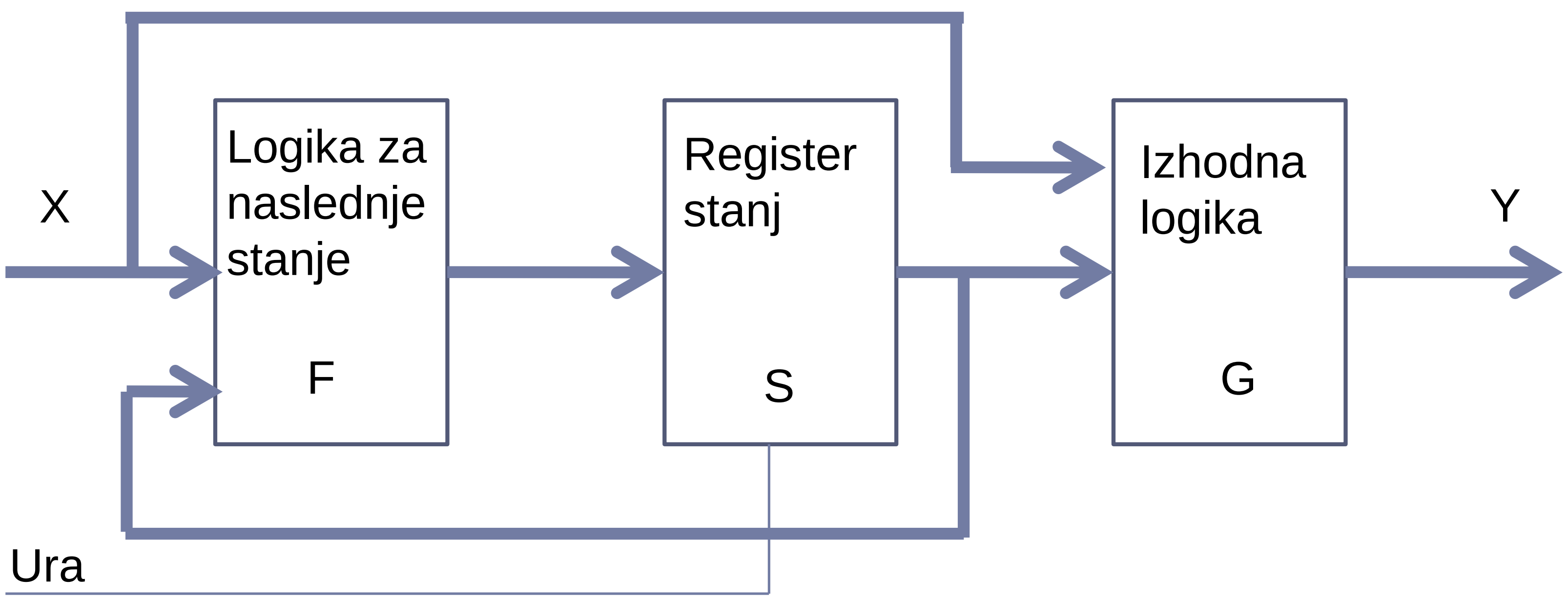
  S(t+1)=F[X, S(t)]
- Izhodna logika s funkcijo G v odvisnosti od trenutnega stanja (S(t)) določa izhode (Y).

  Y=G[S(t)]



Mealy-jev avtomat

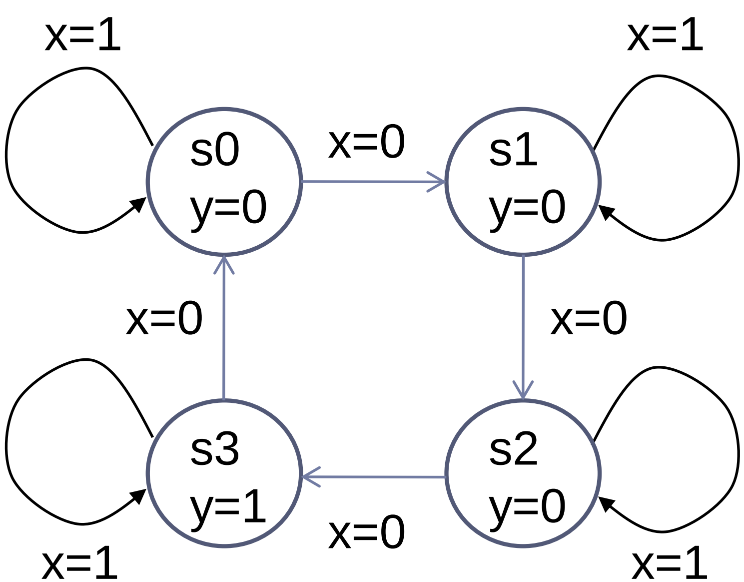
- Logika za naslednje stanje s funkcijo F v odvisnosti od vhodov (X) in trenutnega stanja (S(t)) določa naslednje stanje (S(t+1)), ki se shrani v register stanj.
  S(t+1)=F[X, S(t)]
- Izhodna logika s funkcijo G v odvisnosti od vhodov (X) in trenutnega stanja (S(t)) določa izhode (Y).
  Y=G[X, S(t)]



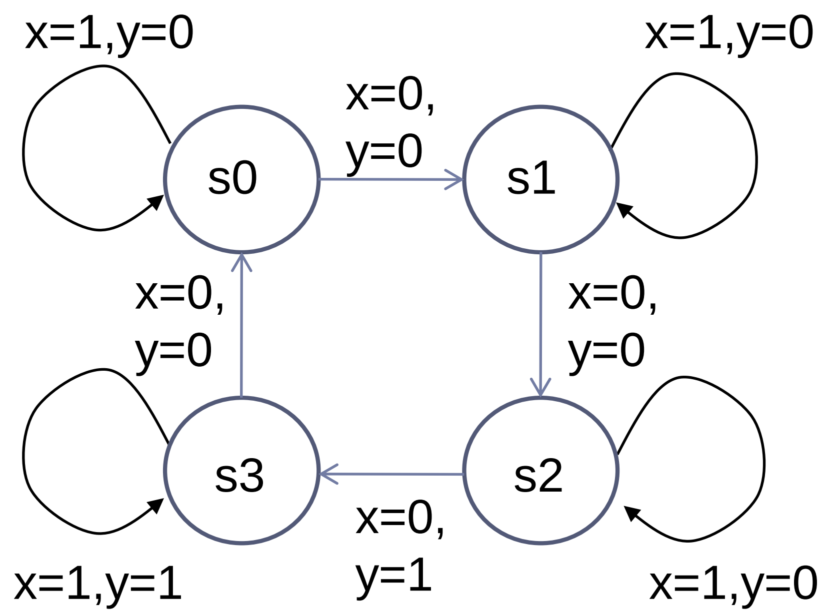
Predstavitev avtomatov

- Diagram prehajanja stanj opisuje delovanje avtomata – opisuje delovanje vezja.
- Primer: Avtomat ima na izhodu vrednost 1, če se na vhodu pojavi zaporedje treh ničel (000), ki mu mora slediti še ena ničla, da se vrne v začetno stanje (s0)

Moore-ov avtomat



Mealy-jev avtomat



Predstavitev avtomatov

- Tabela prehajanja stanj- osnova za določanje ali analizo delujočega digitalnega vezja,

Moore-ov avtomat

	x=0	x=1	Y
s0	s1	s0	0
s1	s2	s1	0
s2	s3	s2	0
s3	s0	s3	1

Trenutna stanja S(t)

Naslednja stanja S(t+1)

Izhod Y

Mealy-jev avtomat

	x=0	x=1
s0	s1,0	s0,0
s1	s2,0	s1,0
s2	s3,1	s2,0
s3	s0,0	s3,1

Trenutna stanja S(t)

Naslednja stanja, Izhod S(t+1),Y

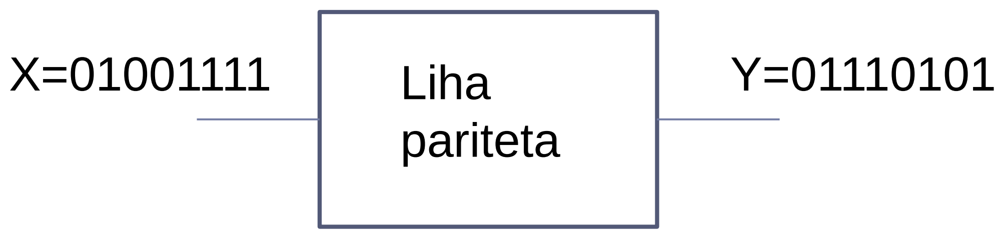
Paritetna kontrola

Štetje števila enic v vhodnem nizu.

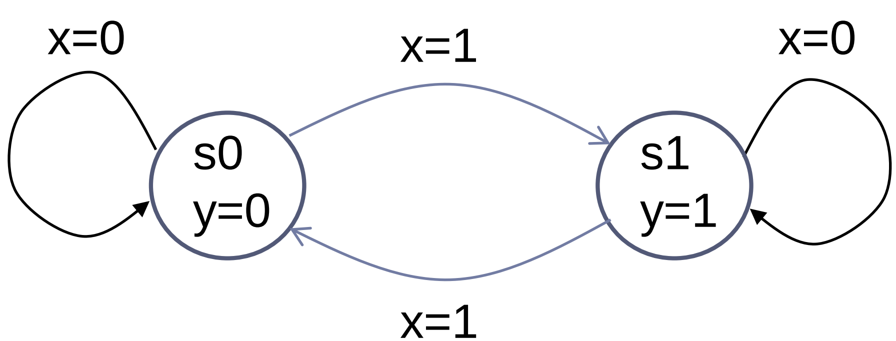
<u>Primer:</u> Liha pariteta vhodnega signala X=(0,1)

Izhod y= 1, če je bilo do danega trenutka na vhodu prisotnih liho število enic in y=0, če je bilo prisotno sodo število enic.

Potrebujemo dve stanji: s0 in s1, ki določata vrednost izhoda



1. Določili bomo diagram prehajanja stanj za Moore-ov avtomat.
2. Stanje s0 = SOD, stanje s1= LIH



	x=0	x=1	Y
s0	s0	s1	0
s1	s1	s0	1

a) Načrtovanje vezja (Pomnilna celica D)

2. Tabelo prehajanja stanj pretvorimo v aplikacijsko tabelo za načrtovanje sekvenčnega vezja: Stanji s0 in s1 je potrebno kodirati: s0=0, s1=1

x	s (t)	s(t+1)	y
0	s0	s0	0
0	s1	s1	1
1	s0	s1	0
1	s1	s0	1

x	s (t)	s(t+1)	D	y
0	0	0	0	0
0	1	1	1	1
1	0	1	1	0
1	1	0	0	1

Funkcija prehajanja stanj (F) za D

Funkcija izhoda (G)

3. Izračun funkcij naslednjega stanja (F) in izhoda (G).

   Če izberemo <u>pomnilno celico D</u>, je potrebno izračunati krmiljenje vhoda D za določanje naslednjega stanja. Funkcija F se določi iz s(t+1) in je odvisna od trenutnega stanja s(t) in vhoda x. Zapišemo jo v obliki vsote produktov in dobimo (glej tabelo):

   F: $D = x'.s(t)+x.s(t) = x\nabla s(t)$

   G: Izhod y je enak trenutnemu stanju s(t), to pomeni, da y=s(t).

b) Načrtovanje vezja (Pomnilna celica JK)

x	s (t)	s(t+1)	J=K	y
0	0	0	0	0
0	1	1	0	1
1	0	1	1	0
1	1	0	1	1

Funkcija prehajanja stanj (F) za JK

Funkcija izhoda (G)

3. Izračun funkcij naslednjega stanja (F) in izhoda (G).

Če izberemo pomnilno celico JK, je potrebno izračunati krmiljenje vhoda J=K za določanje naslednjega stanja. Zapišemo jo v obliki vsote produktov in dobimo (glej tabelo):

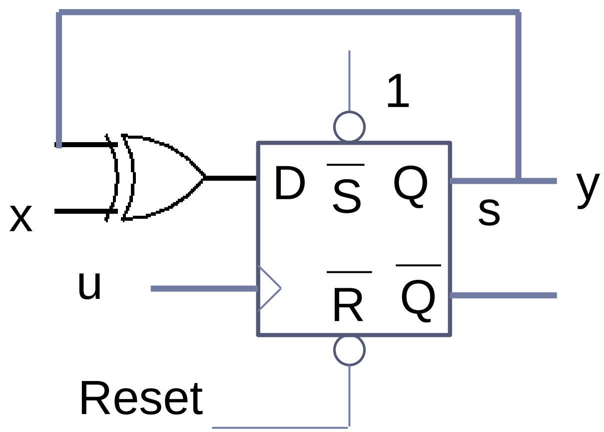
F: J=K =x

G: Izhod y je enak trenutnemu stanju s(t), to pomeni, da y=s(t).

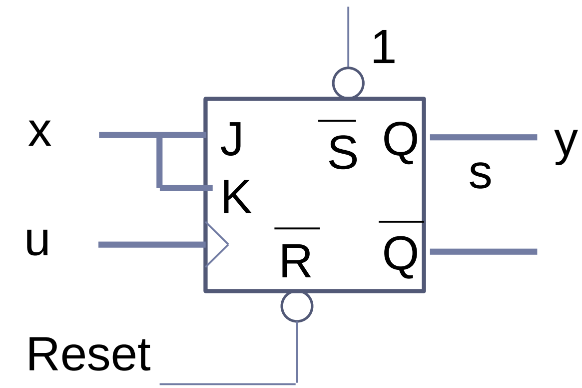
Delovanje vezja

3. Vezje za določanje lihe paritete. Vezju dodamo še asinhronski vhod za postavitev začetnega stanja s0 (Reset)

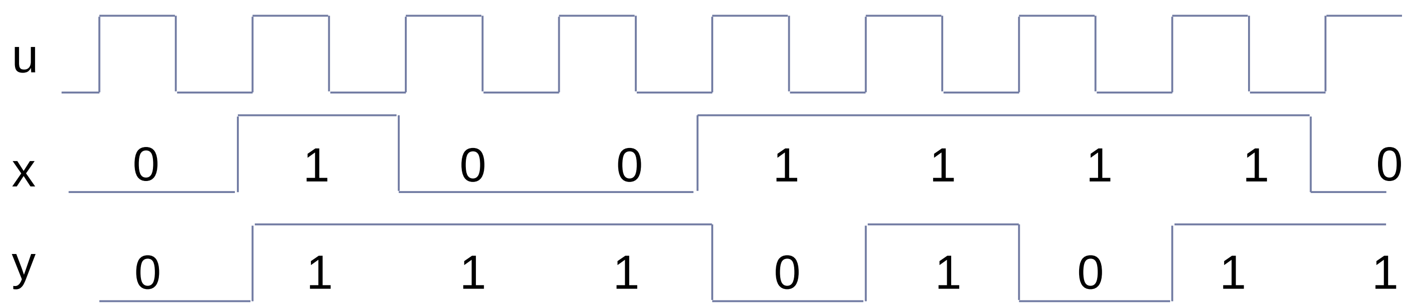
a)



b)

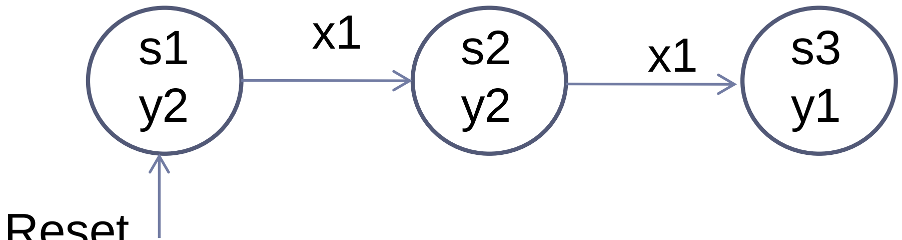


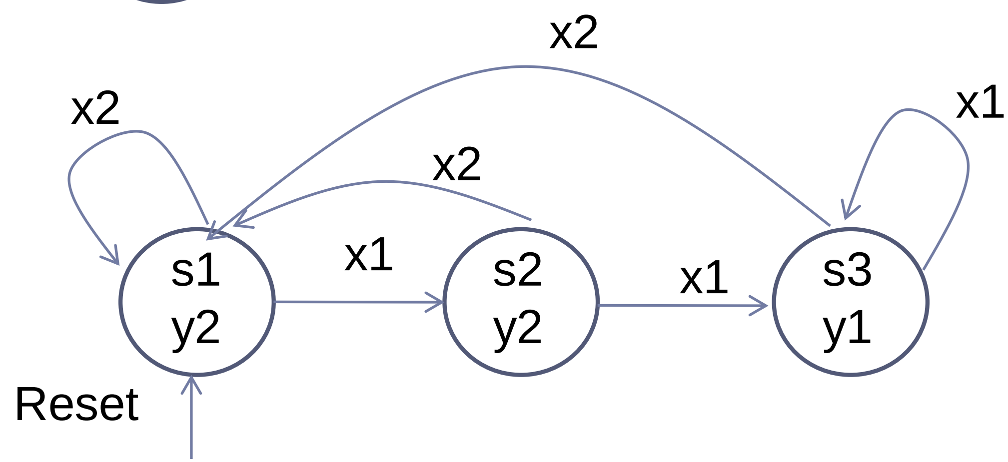
Časovni diagram



Primer 1: Razpoznavanje niza

- Diagram prehajanja stanj za razpoznavanje niza črk, če je X=(x1,x2) in Y=(y1,y2). Na izhodu naj se pojavi y1, če se na vhodu pojavita vhoda x1x1.



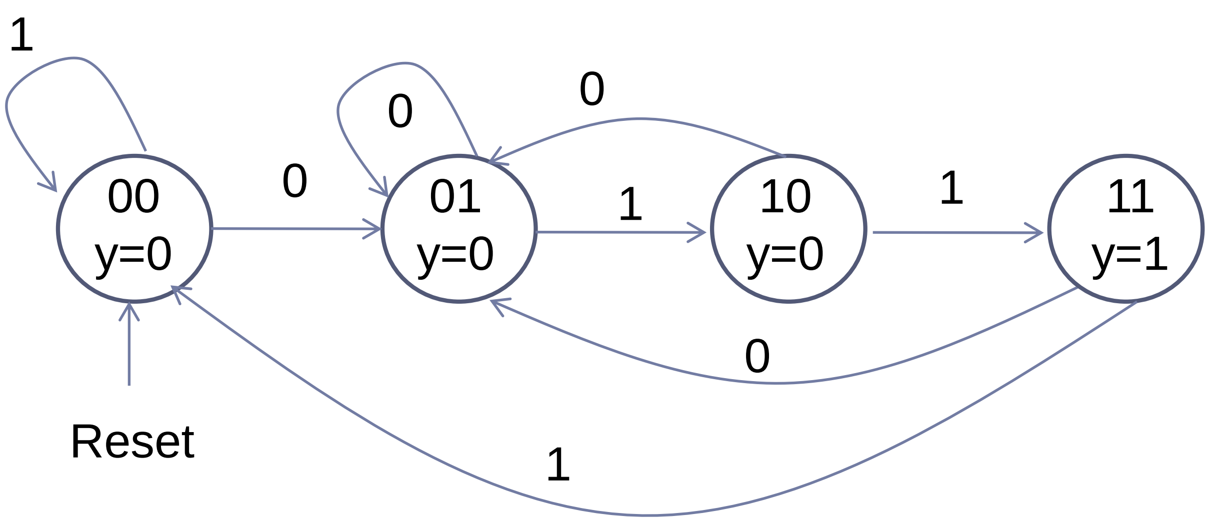


<u>Vhod</u>: x1 x2 x2 x1 x1 x1 x2 x1 x2 x1 x1 x1 x1 x1 …

<u>Izhod</u>: y2 y2 y2 y2 y1 y1 y2 y2 y2 y2 y1 y1 y1 y1 …

Primer 2: Razpoznavanje niza

- 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.



<u>Vhod</u>: 10011011001001100 Na vhodu se 3x pojavi niz 011

<u>Izhod</u>: 00001001000000100

Na izhodu = 1 ob vsakem pojavu niza 011

Primer 2 Rešitev

- Tabela stanj:

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

- Realizacija: JK pomnilna celica, AND, OR, NOT

Primer 2 Rešitev a

- Minimizacija

x

Q_1		1	1	1
		1		

Q_0

$$J_1 = K_1 = \bar{x}.Q_1 \vee x.Q_0$$

x

Q_1	1	1		1
		1		1

Q_0

$$J_0 = K_0 = \bar{x}.\overline{Q_0} \vee x.Q_0 \vee x.Q_1$$

x

Q_1		1	1	

Q_0

$$y = Q_1.Q_0$$

- Vhod: 0110011011010110
- Izhod:0010000100100010