



Sekvenčna vezja



Sekvenčna vezja

Kombinacijsko in sekvenčno izvajanje nalog

- s kombinacijskimi vezji izvajamo naloge, katerih rezultat (izhod) je v vsakem trenutku enolično določen s trenutno vrednostjo končnega števila dejavnikov (vhodov)
- za izvedbo naloge te dejavnike in rezultat izrazimo z Booleovimi spremenljivkami, zapišemo pravilnostno tabelo, odtod preklopno funkcijo in iz nje zgradimo vezje
- za izvajanje nalog, katerih rezultat ni odvisen le od trenutne, temveč tudi od preteklih vrednosti vhodov, pa potrebujemo še način shranjevanja podatkov
- če je zgodovina vhodov, ki vpliva na rezultat, časovno omejena, lahko tudi takšno nalogo načeloma opišemo s pravilnostno tabelo, v kateri navedemo izhodne vrednosti pri vseh možnih zaporedjih vrednosti vhodov

x_1	x_2	y
0	0	0
0	1	0
1	0	1
1	1	0

t_{-n}		...	t_{-1}		t_0		t_0
x_1	x_2	...	x_1	x_2	x_1	x_2	y
0	0	...	0	0	0	0	0
0	0	...	0	0	0	1	1
...	...	...	...	...	...	...	...



Sekvenčna vezja

Kombinacijsko in sekvenčno izvajanje nalog

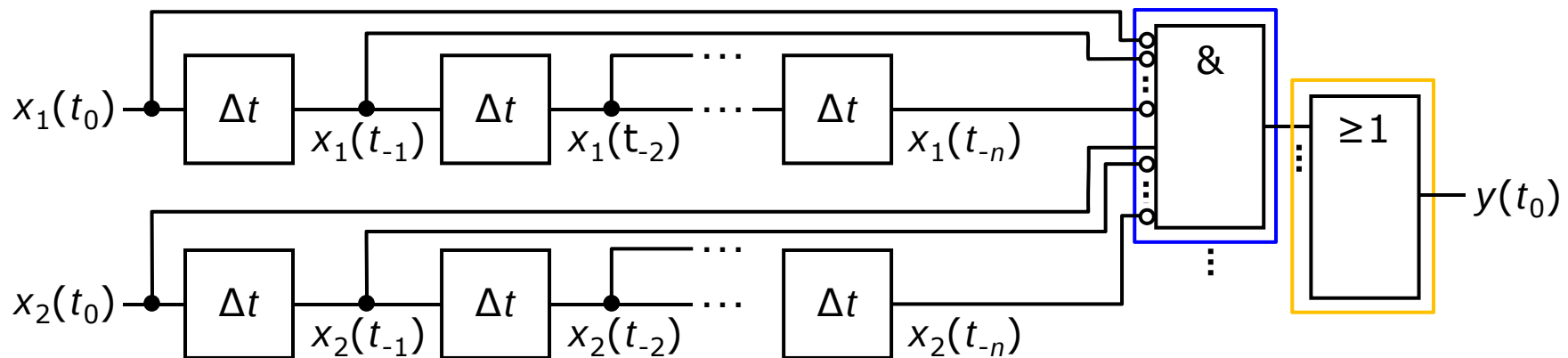
t_{-n}		...	t_{-1}		t_0		t_0
x_1	x_2	...	x_1	x_2	x_1	x_2	y
0	0	...	0	0	0	0	0
0	0	...	0	0	0	1	1
...	...	...	...	...	...	...	...

- enice funkcije y so mintermi ...

- celotna funkcija y pa je njihova vsota:

$$y(t_0) = \bar{x}_1(t_{-n})\bar{x}_2(t_{-n}) \dots \bar{x}_1(t_{-1})\bar{x}_2(t_{-1})\bar{x}_1(t_0)x_2(t_0) + \dots$$

- kombinacijsko vezje, ki izvaja takšno funkcijo, lahko zgradimo z n zakasnilnimi elementi:



- za splošno nalogo takšne vrste, kjer na rezultat vpliva dolga zgodovina vhodov, je tak pristop zelo okoren; če zgodovina ni omejena, pa je sploh neizvedljiv



Sekvenčna vezja

Spremenljivke stanja, stanja in sekvenčna vezja

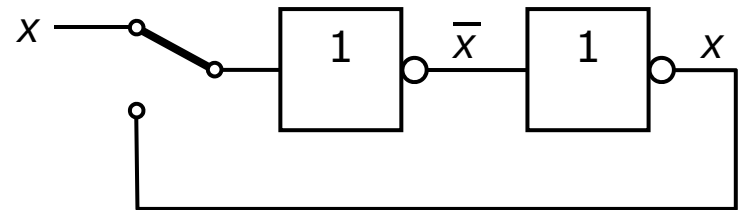
- v praksi se običajno omejimo na naloge, katerih rezultat je moč zapisati kot funkcijo končnega števila trenutnih vrednosti vhodov in končnega števila vrednosti notranjih spremenljivk – **spremenljivk stanja**; celotnemu naboru spremenljivk stanja pravimo **stanje**
- **sekvenčno vezje** (*angl. sequential logic circuit*): vezje, katerega naslednje stanje in vrednosti na izhodih so enolično določeni s trenutnim stanjem in trenutnimi vrednostmi na vseh vhodih
- ker je število spremenljivk stanja končno, te spremenljivke pa so Booleove, je končno tudi število možnih stanj; če je spremenljivk stanja n , je torej število možnih stanj 2^n
- v **sinhronskem sekvenčnem vezju** (*angl. synchronous sequential circuit*) vse strukture spremenijo stanje sočasno; to dosežemo s **proženjem**, ki je lahko aperiodično ali periodično (npr. s pulzi **ure**)



Sekvenčna vezja

Spominske celice: primitivna spominska celica

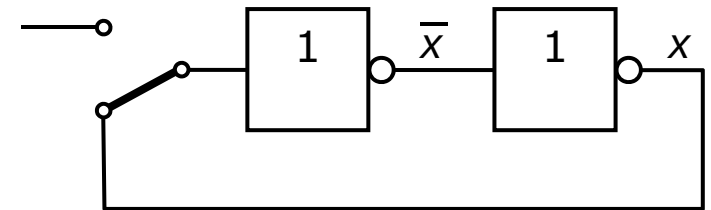
- sekvenčna vezja temeljijo na gradnikih, ki jim omogočajo pomnjenje (hranjenje) podatkov o prejšnjem stanju – **spominskih celicah**
- najpreprostejšo, t.i. **primitivno spominsko celico**, zgradimo iz dveh zaporedno vezanih negatorjev:
- če na vhod pripeljemo vrednost x , se po eni zakasnitvi na izhodu prvega negatorja pojavi $\bar{x}$, po še eni pa na izhodu drugega negatorja x



Sekvenčna vezja

Spominske celice: primitivna spominska celica

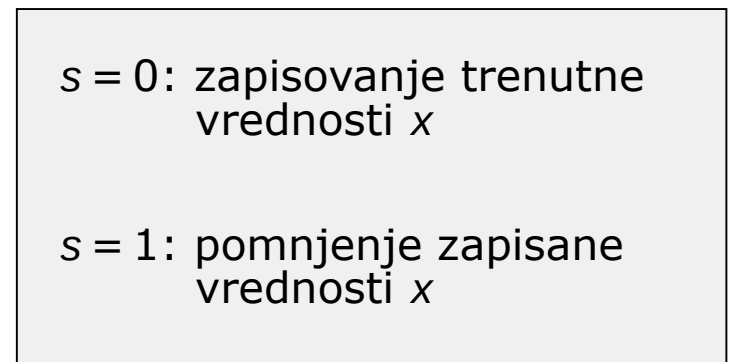
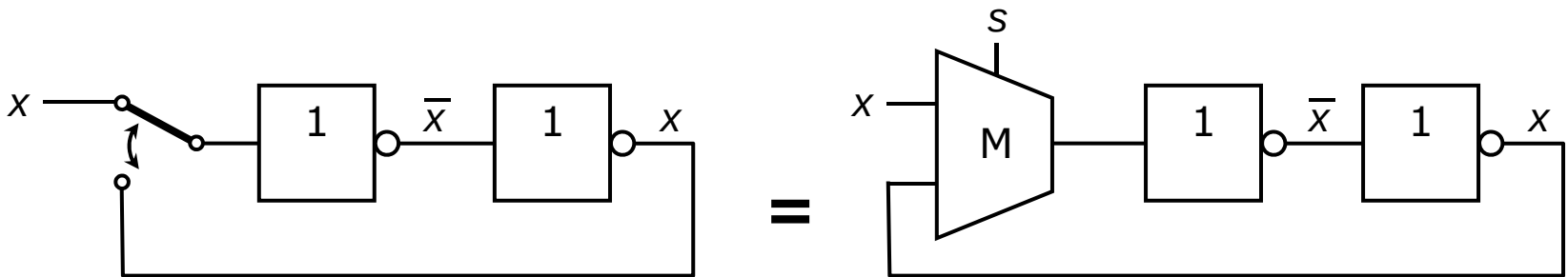
- sekvenčna vezja temeljijo na gradnikih, ki jim omogočajo pomnjenje (hranjenje) podatkov o prejšnjem stanju – **spominskih celicah**
- najpreprostejšo, t.i. **primitivno spominsko celico**, zgradimo iz dveh zaporedno vezanih negatorjev:
- če na vhod pripeljemo vrednost x , se po eni zakasnitvi na izhodu prvega negatorja pojavi $\bar{x}$, po še eni pa na izhodu drugega negatorja x
- če stikalo preklopimo, s tem ustvarimo povratno zvezo, ki bo odtlej v spominski celici vzdrževala zapisani vrednosti $\bar{x}$ in x
- če želimo v spominsko celico shraniti novo vrednost spremenljivke x , ponovno preklopimo stikalo na vhod in nato nazaj v povratno zvezo



Sekvenčna vezja

Spominske celice: primitivna spominska celica

- za zapisovanje v primitivno spominsko celico potrebujemo stikalo
- v digitalni izvedbi je to dvovhodni multipleksor:

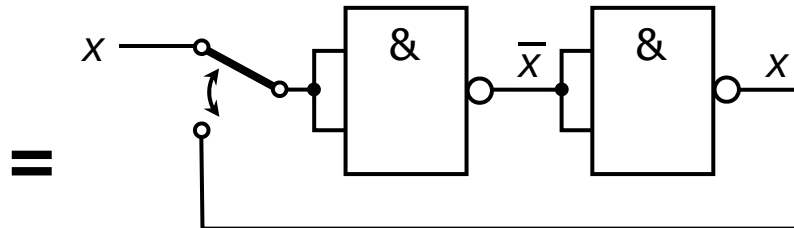
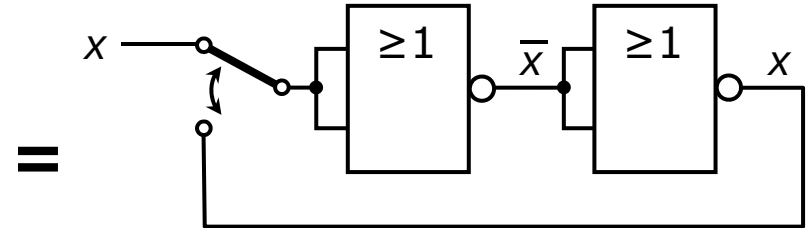
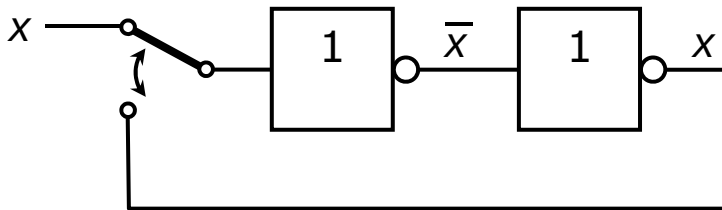




Sekvenčna vezja

Spominske celice: primitivna spominska celica

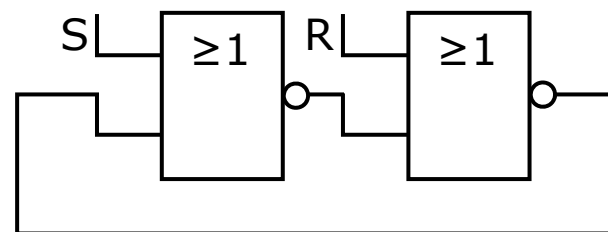
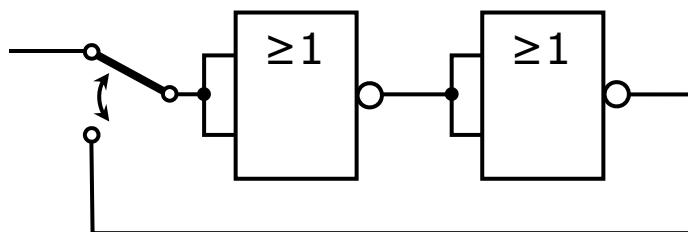
- ker velja $\bar{x} = x \downarrow x = x \uparrow x$, lahko namesto iz para zaporednih negatorjev primitivno spominsko celico z enakim delovanjem zgradimo tudi iz para zaporednih vrat NOR ali para zaporednih vrat NAND:



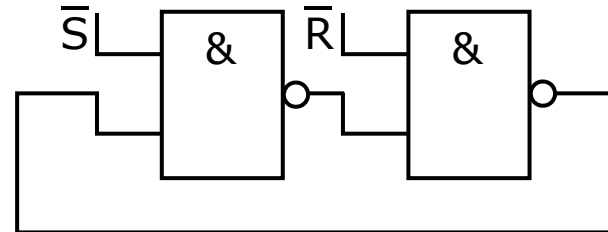
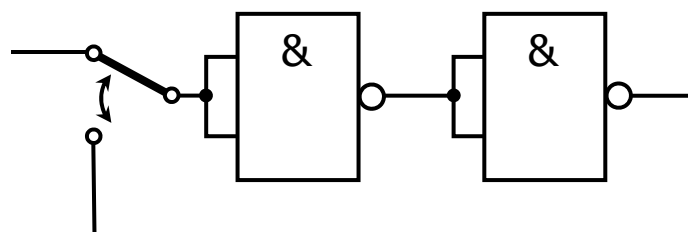
Sekvenčna vezja

Spominske celice: nesinhronizirana zapaha SR in $\bar{S}\bar{R}$

- izvedbi s parom dvovhodnih vrat omogočata predelavo v obliko, v kateri stikala (multipleksorja) ne potrebujemo več, saj za zapisovanje in pomnjenje uporabimo po en vhod vsakih vrat:



nesinhronizirani zapah SR

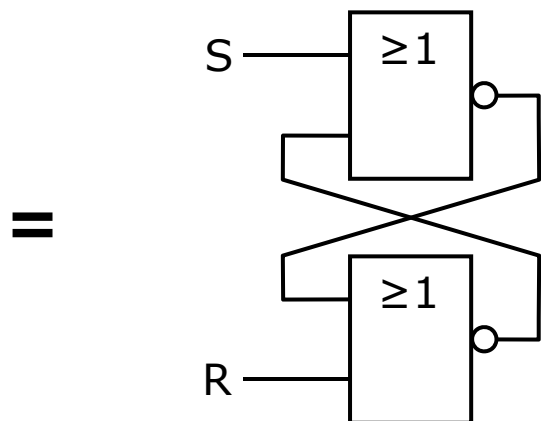
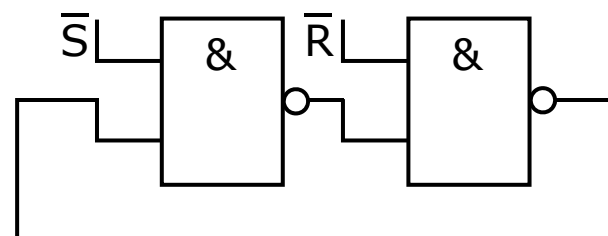
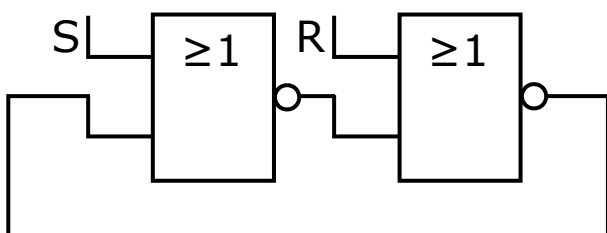


nesinhronizirani zapah $\bar{S}\bar{R}$

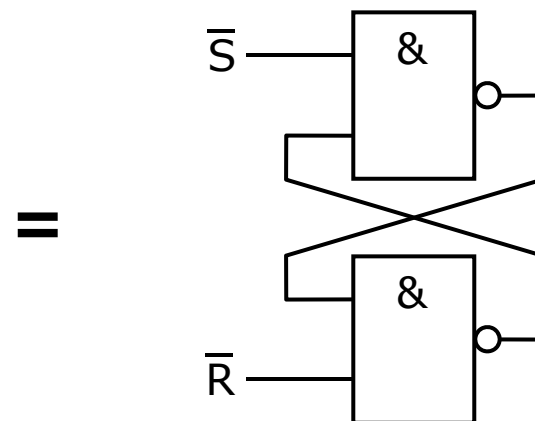
Sekvenčna vezja

Spominske celice: nesinhronizirana zapaha SR in $\bar{S}\bar{R}$

- ker sta S in R vhoda zapaha SR, $\bar{S}$ in $\bar{R}$ pa vhoda zapaha $\bar{S}\bar{R}$, vezji teh spominskih celic običajno prikažemo nekoliko drugače:



nesinhronizirani zapah SR

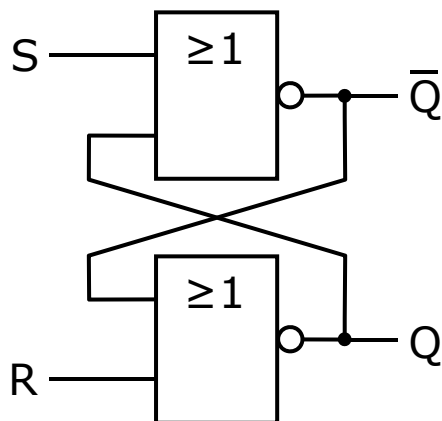


nesinhronizirani zapah $\bar{S}\bar{R}$

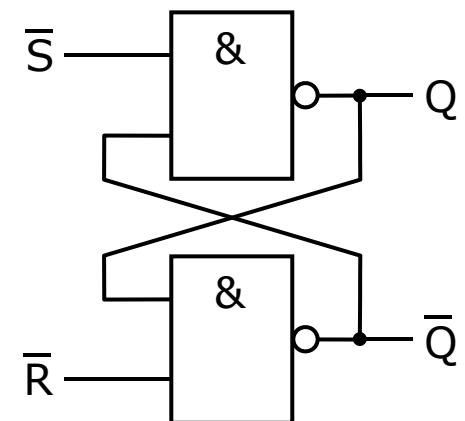
Sekvenčna vezja

Spominske celice: nesinhronizirana zapaha SR in $\bar{S}\bar{R}$

- da bi bili ti spominski celici uporabni, potrebujemo še dostop do vrednosti, ki jih hranita – vrednosti na izhodih njunih vrat
- kot bomo spoznali v nadaljevanju, sta ti vrednosti (tako kot v primitivni spominski celici) pri pravilnem delovanju oziroma uporabi spominske celice negaciji druga druge; običajno ju označujemo s simboloma Q in $\bar{Q}$ (to sta **spremenljivki stanja** spominske celice)



nesinhronizirani zapah SR

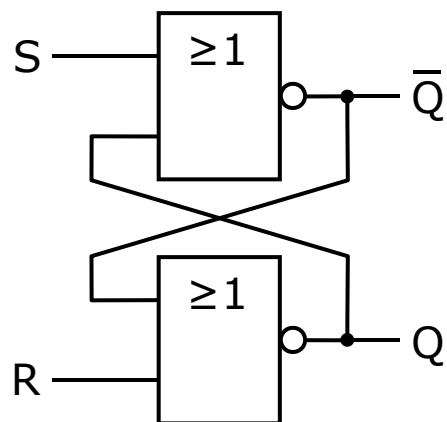


nesinhronizirani zapah $\bar{S}\bar{R}$



Sekvenčna vezja

Spominske celice: nesinhronizirani zapah SR



naslednji stanji Q in $\bar{Q}$

S	R	Q*	$\bar{Q}$ *
0	0		
0	1		
1	0	1	0
1	1		

1) pri $R=0$ postavimo $S=1$:

- ker velja $x \downarrow 1 = 0$, bo po eni zakasnitvi na izhodu zgornjih vrat vrednost $\bar{Q}=0$, ki se bo prenesla tudi na zgornji vhod spodnjih vrat;
- ker je $0 \downarrow 0 = 1$, bo po drugi zakasnitvi na izhodu spodnjih vrat vrednost $Q=1$, ki se bo prenesla tudi na spodnji vhod zgornjih vrat;
- ker je $1 \downarrow 1 = 0$, to ne bo spremenilo vrednosti na izhodu zgornjih vrat in celica bo ostala v stanju $\{Q=1, \bar{Q}=0\}$

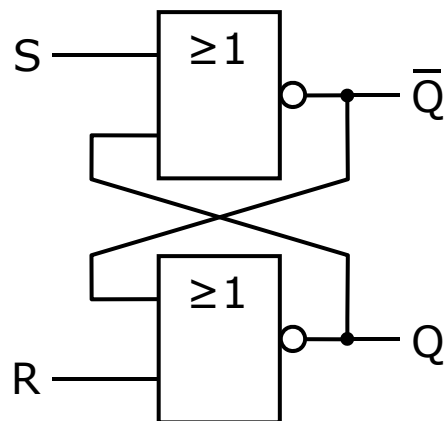
2) ohranimo $R=0$ in vrnimo S na vrednost 0:

- na izhodu zgornjih vrat imamo $0 \downarrow 1 = 0$, na izhodu spodnjih pa $0 \downarrow 0 = 1$; celica torej **ostaja v vzpostavljenem stanju** $\{Q=1, \bar{Q}=0\}$



Sekvenčna vezja

Spominske celice: nesinhronizirani zapah SR



S	R	Q^*	$\bar{Q}^*$
0	0	Q	$\bar{Q}$
0	1	0	1
1	0	1	0
1	1		

3) pri $S = 0$ postavimo $R = 1$:

- ker velja $x \downarrow 1 = 0$, bo po eni zakasnitvi na izhodu spodnjih vrat vrednost $Q = 0$, ki se bo prenesla tudi na spodnji vhod zgornjih vrat;
- ker je $0 \downarrow 0 = 1$, bo po drugi zakasnitvi na izhodu zgornjih vrat vrednost $\bar{Q} = 1$, ki se bo prenesla tudi na zgornji vhod spodnjih vrat;
- ker je $1 \downarrow 1 = 0$, to ne bo spremenilo vrednosti na izhodu spodnjih vrat in celica bo ostala v stanju $\{Q = 0, \bar{Q} = 1\}$

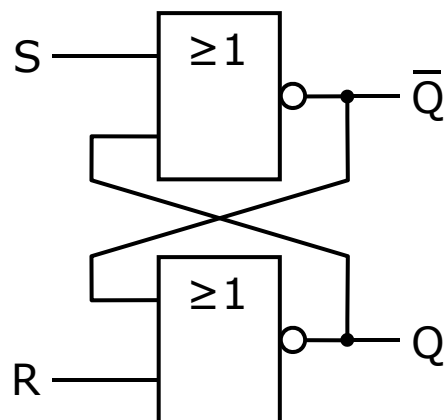
4) ohranimo $S = 0$ in vrnimo R na vrednost 0:

- na izhodu zgornjih vrat imamo $0 \downarrow 0 = 1$, na izhodu spodnjih pa $1 \downarrow 0 = 0$; celica torej **ostaja v vzpostavljenem stanju** $\{Q = 0, \bar{Q} = 1\}$



Sekvenčna vezja

Spominske celice: nesinhronizirani zapah SR



S	R	Q^*	$\bar{Q}^*$
0	0	Q	$\bar{Q}$
0	1	0	1
1	0	1	0
1	1	0	0

5) postavimo sedaj hkrati $S = 1$ in $R = 1$:

- ker velja $x \downarrow 1 = 0$, bo po eni zakasnitvi na izhodu zgornjih vrat vrednost $\bar{Q} = 0$ in hkrati na izhodu spodnjih vrat vrednost $Q = 0$, obe se bosta sočasno tudi prenesli na vhoda;
- ker je $1 \downarrow 0 = 0 \downarrow 1 = 0$, to ne bo spremenilo vrednosti na izhodih vrat in celica bo ostala v stanju $\{Q = 0, \bar{Q} = 0\}$

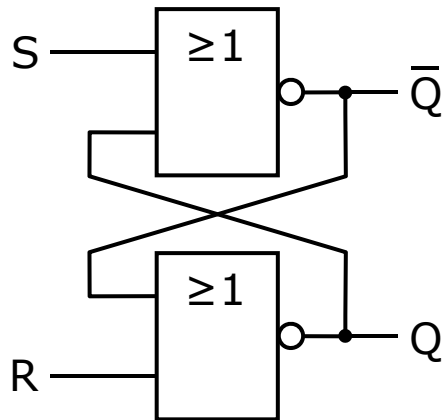
6) vrnimo S in R na vrednost 0:

- če prej vrnemo R, velja analiza iz primera (1) in dobimo stanje $\{Q = 1, \bar{Q} = 0\}$; če prej S, pa analiza iz primera (3) in stanje $\{Q = 0, \bar{Q} = 1\}$;
- če obe vrednosti vrnemo na 0 sočasno ali ne vemo, katera se je vrnila prej, pa stanja celice ne moremo predvideti



Sekvenčna vezja

Spominske celice: nesinhronizirani zapah SR



S	R	Q^*	$\bar{Q}^*$
0	0	Q	$\bar{Q}$
0	1	0	1
1	0	1	0
1	1	0	0

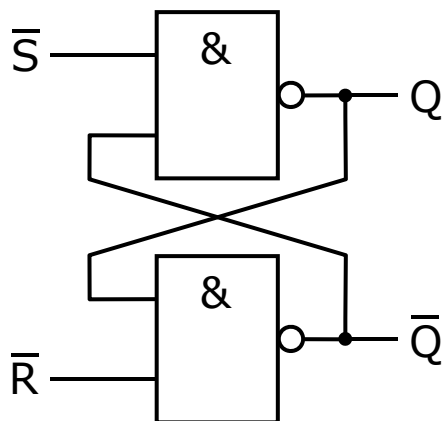
← $Q = \bar{Q}$!!!

- s kombinacijo $\{S = 1, R = 0\}$ torej v celico zapišemo vrednost $Q = 1$, s kombinacijo $\{S = 0, R = 1\}$ vrednost $Q = 0$, pri kombinaciji $\{S = 0, R = 0\}$ pa se zapisana vrednost ohranja
- odtod tudi ime celice: z vhodom $S = 1$ v celico zapišemo vrednost 1, z vhodom $R = 1$ pa jo z zapisom vrednosti 0 spet izbrišemo (*angl. set* = postaviti, *reset* = vrniti v izhodiščno stanje)
- pri naštetih treh kombinacijah je dosledno tudi poimenovanje notranjih spremenljivk Q in $\bar{Q}$, saj je $\bar{Q}$ dejansko vselej negacija Q
- pri kombinaciji $\{S = 1, R = 1\}$ pride do "protislovnega" stanja $\{Q = 0, \bar{Q} = 0\}$, po vrnitvi na $\{S = 0, R = 0\}$ pa stanje ni več predvidljivo



Sekvenčna vezja

Spominske celice: nesinhronizirani zapah $\bar{S}\bar{R}$ (*angl. $\bar{S}\bar{R}$ latch*)



$\bar{S}$	$\bar{R}$	Q^*	$\bar{Q}^*$
0	0	1	1
0	1	1	0
1	0	0	1
1	1	Q	$\bar{Q}$

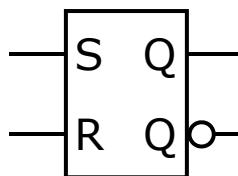
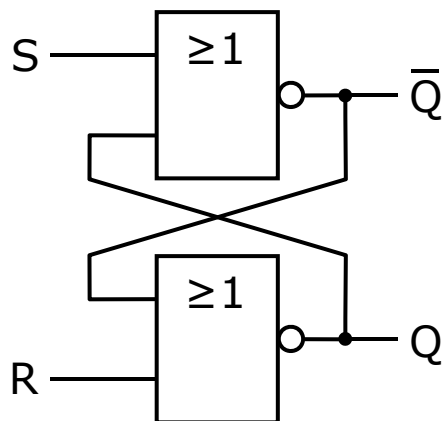
← $Q = \bar{Q}$!!!

- na enak način lahko analiziramo tudi delovanje zapaha $\bar{S}\bar{R}$ in ugotovimo, da s kombinacijo $\{S = 1, R = 0\}$ vanj zapišemo vrednost $Q = 1$, s $\{S = 0, R = 1\}$ vrednost $Q = 0$, pri $\{S = 0, R = 0\}$ se zapisana vrednost ohranja, pri $\{S = 1, R = 1\}$ pa nastopi stanje, v katerem velja $Q = 1$ in $\bar{Q} = 1$ (t.j. $\bar{Q}$ ni negacija Q) in stanje po vrnitvi na $\{S = 1, R = 1\}$ ni več predvidljivo
- zapah $\bar{S}\bar{R}$ torej deluje enako kot zapah SR , le
(i) z negativno vhodno logiko namesto pozitivne (odtod tudi ime) in
(ii) s "protislovnim" stanjem $\{Q = 1, \bar{Q} = 1\}$ namesto $\{Q = 0, \bar{Q} = 0\}$



Sekvenčna vezja

Spominske celice: nesinhronizirana zapaha SR in $\bar{S}\bar{R}$



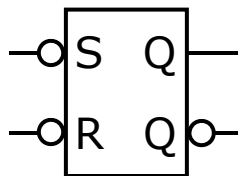
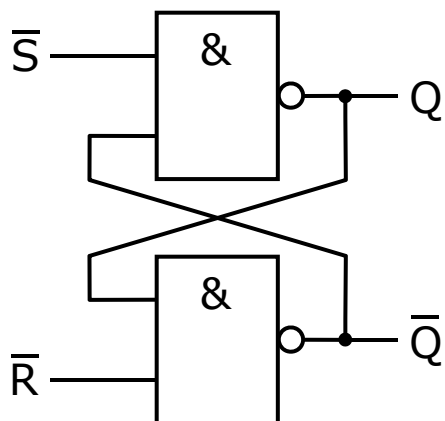
simbol

S	R	Q^*	$\bar{Q}^*$
0	0	Q	$\bar{Q}$
0	1	0	1
1	0	1	0
1	1	0	0

pravilnostna
tabela

Q	Q^*	S	R
0	0	0	0
0	1	1	0
1	0	0	1
1	1	0	0

vzbujalna
tabela



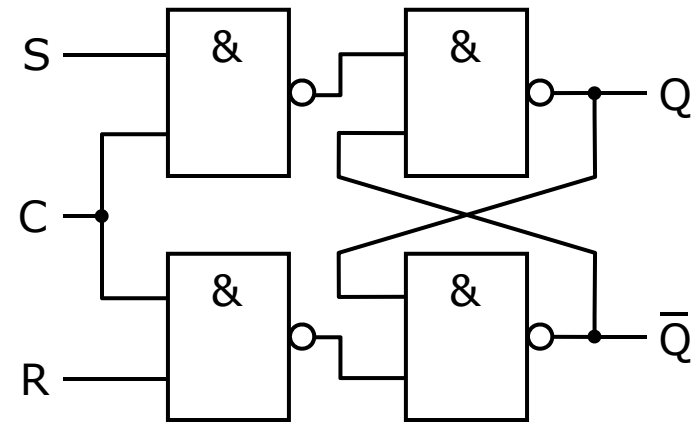
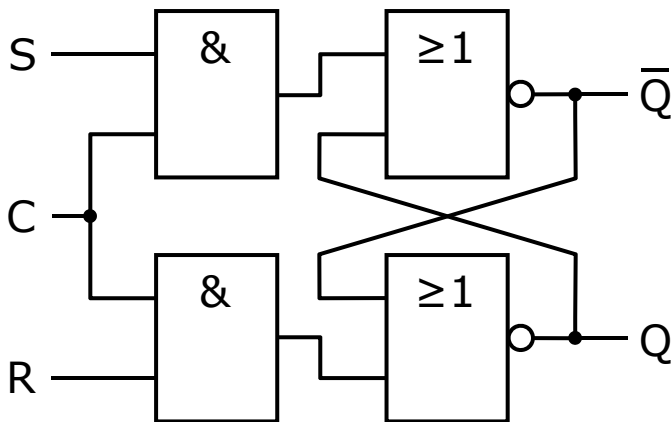
$\bar{S}$	$\bar{R}$	Q^*	$\bar{Q}^*$
0	0	1	1
0	1	1	0
1	0	0	1
1	1	Q	$\bar{Q}$

Q	Q^*	$\bar{S}$	$\bar{R}$
0	0	1	0
0	1	0	1
1	0	1	0
1	1	0	1

Sekvenčna vezja

Spominske celice: sinhronizirani zapah SR

- nesinhronizirana zapaha SR in $\overline{S}\overline{R}$ se vseskozi odzivata na vhoda
- to odzivnost lahko časovno omejimo tako, da na vhoda zapaha SR dogradimo par vrat AND, na vhoda zapaha $\overline{S}\overline{R}$ pa par vrat NAND:

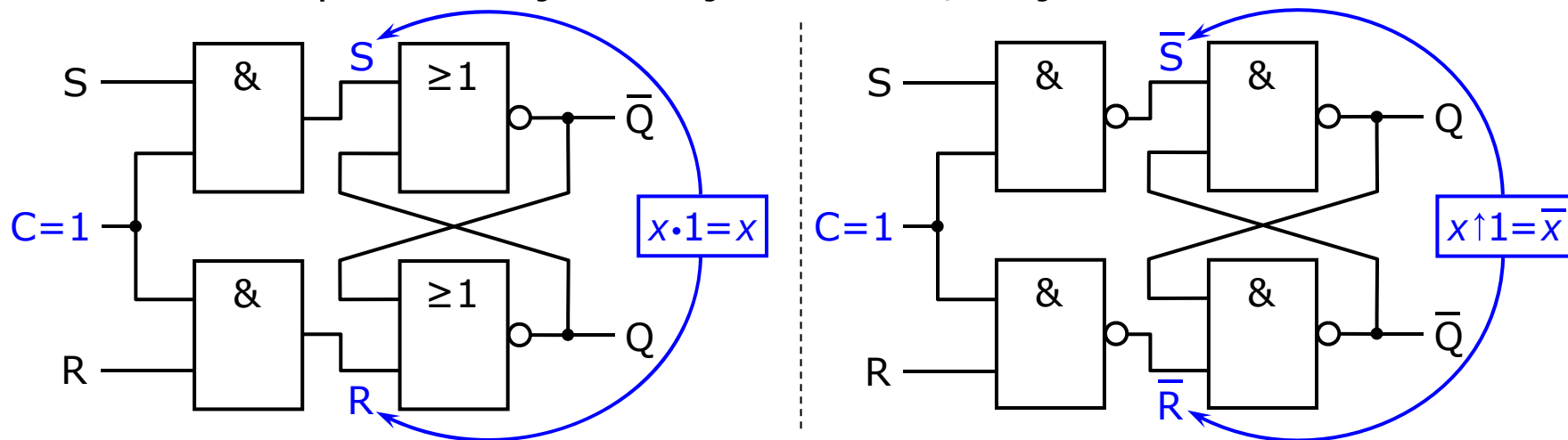




Sekvenčna vezja

Spominske celice: sinhronizirani zapah SR

- delovanje obeh zapahov smo tako **sinhronizirali** s signalom C (*angl. clock* = ura oz. *control* = nadzor, kontrola); vrednosti S in R imata dostop v notranjost vezja le takrat, ko je $C = 1$:

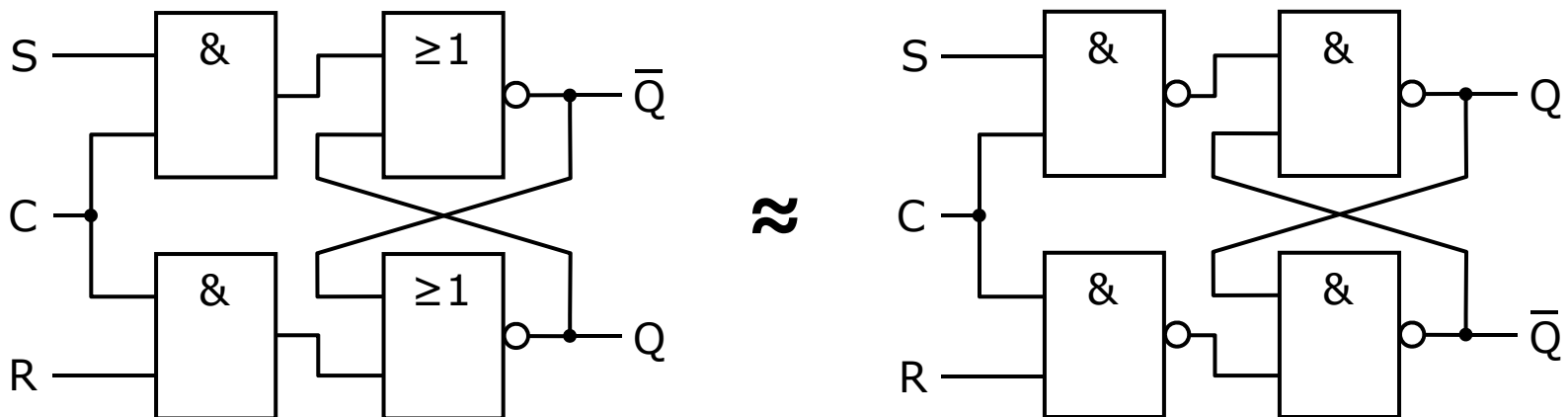


- s sinhronizacijo smo dobili celici, ki imata obe vhoda S in R ter se na kombinacije $\{S = 1, R = 0\}$, $\{S = 0, R = 1\}$ in $\{S = 0, R = 0\}$ odzivata enako, le pri "prepovedani" kombinaciji $\{S = 1, R = 1\}$ zavzameta v izvedbi na levi Q in $\bar{Q}$ vrednost 0, v izvedbi na desni pa vrednost 1

Sekvenčna vezja

Spominske celice: sinhronizirani zapah SR

- ker je pri pravilni uporabi delovanje obeh celic enako, ju pogosto poimenujemo kar z istim imenom – **sinhronizirani zapah SR** (*angl. **SR latch with control** oz. **SR latch with enable***)

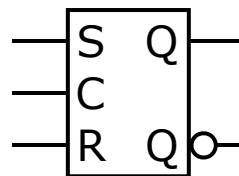
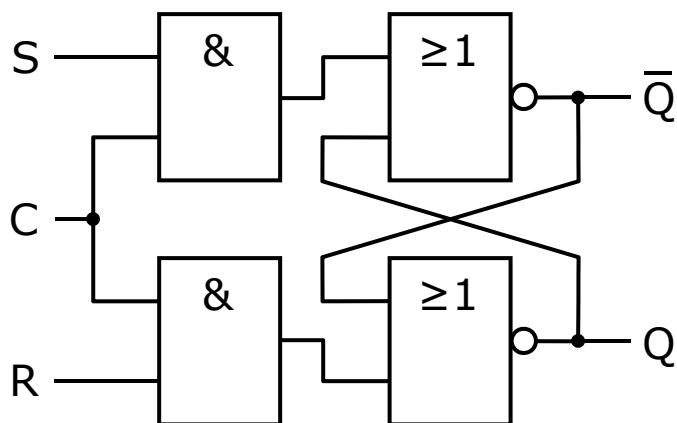


- zaradi tehnoloških prednosti vrat NAND je v uporabi predvsem desna različica – **sinhronizirani zapah SR v izvedbi NAND**



Sekvenčna vezja

Spominske celice: sinhron. zapaha SR v izvedbah NOR in NAND



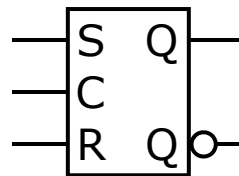
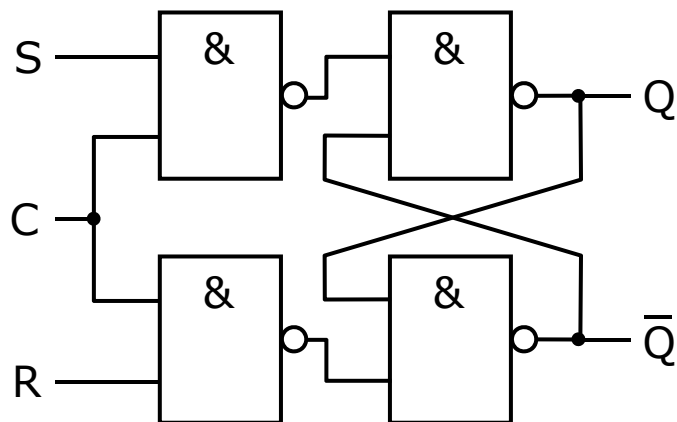
S	R	C	Q^*	$\bar{Q}^*$
0	0	0	Q	$\bar{Q}$
0	0	1	Q	$\bar{Q}$
0	1	1	0	1
1	0	1	1	0
1	1	1	0	0

Q	Q^*	S	R
0	0	0	0
0	1	1	0
1	0	0	1
1	1	0	0

simbol

pravilnostna
tabela

vzbujalna
tabela



S	R	C	Q^*	$\bar{Q}^*$
0	0	0	Q	$\bar{Q}$
0	0	1	Q	$\bar{Q}$
0	1	1	0	1
1	0	1	1	0
1	1	1	1	1

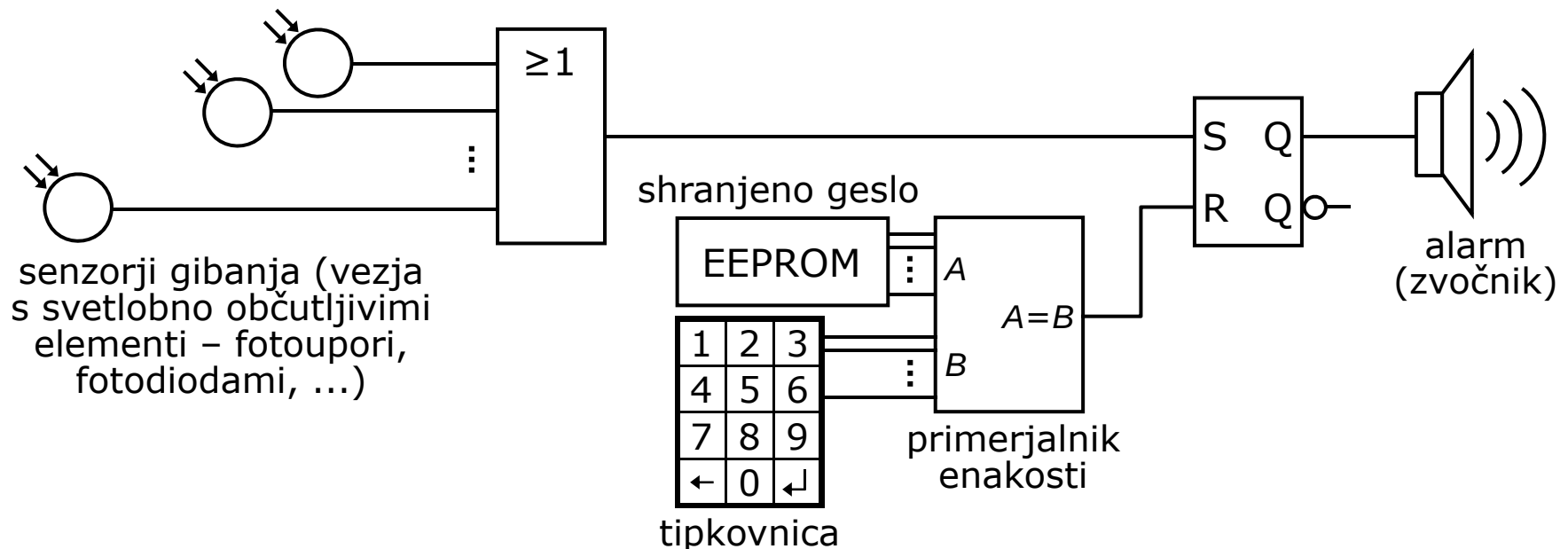
Q	Q^*	S	R
0	0	0	0
0	1	1	0
1	0	0	1
1	1	0	0



Sekvenčna vezja

Spominske celice: zapahi SR in $\bar{S}\bar{R}$

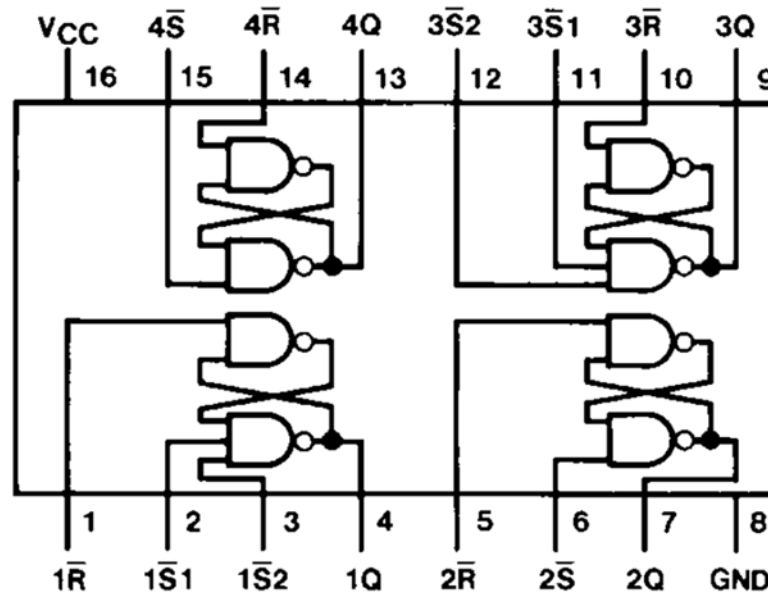
- zapahi SR in $\bar{S}\bar{R}$ so uporabni za izvajanje nalog, pri katerih sta pogoja za zapis enice in za zapis ničle (izbris enice) med seboj neodvisna
- primer: preprosta alarmna naprava – aktivira jo signal s senzorja, deaktivira pa signal z vezja, v katero vtipkamo kodo za deaktivacijo





Sekvenčna vezja

Spominske celice: zapahi SR in $\overline{S}\overline{R}$



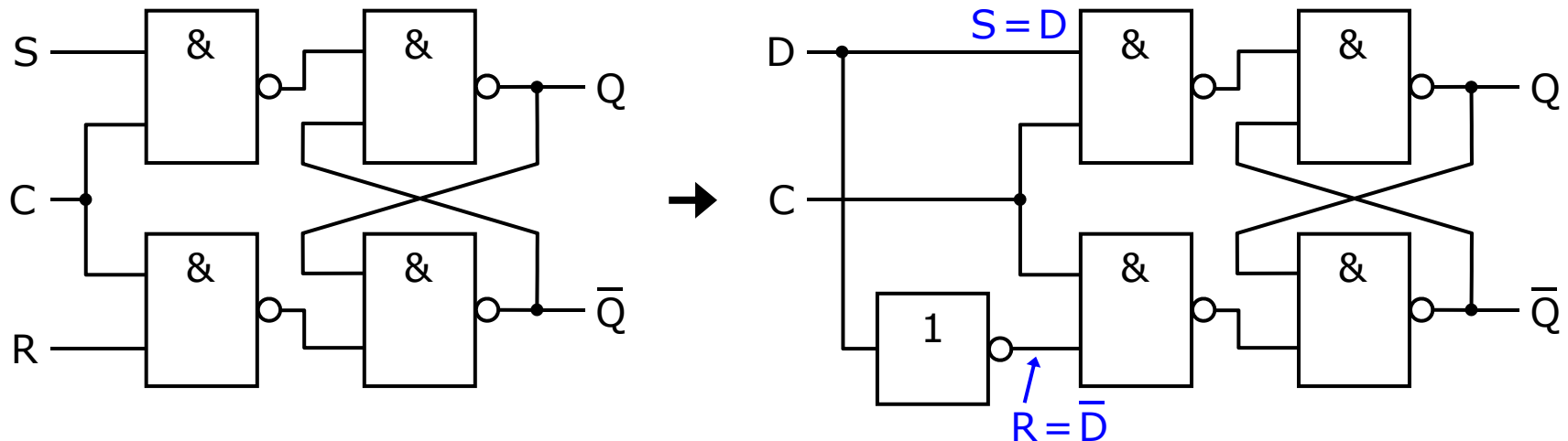
integrirano vezje DM74LS279 (Fairchild) s štirimi nesinhroniziranimi zapahi $\overline{S}\overline{R}$:

- zapaha 1 in 3 imata po dva vhoda $\overline{S}$ in en vhod $\overline{R}$, za vpis enice mora biti ničla vsaj na enem vhodu $\overline{S}$ (t.j. veljati mora $\overline{1S}_1 \cdot \overline{1S}_2 = 0$ oz. $\overline{3S}_1 \cdot \overline{3S}_2 = 0$);
- zapaha 2 in 4 sta običajna zapaha $\overline{S}\overline{R}$ s po enim vhodom $\overline{S}$ in enim vhodom $\overline{R}$

Sekvenčna vezja

Spominske celice: zapah D

- če je namen celice pomnjenje (hranjenje) 1-bitnega podatka, za zapis tega podatka potrebujemo le en vhod (in ne dveh neodvisnih)
- takšno spominsko celico lahko zgradimo tako, da v vhodni del sinhroniziranega zapaha SR dodamo negator, ki zagotavlja $R = \bar{S}$:



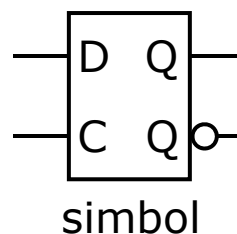
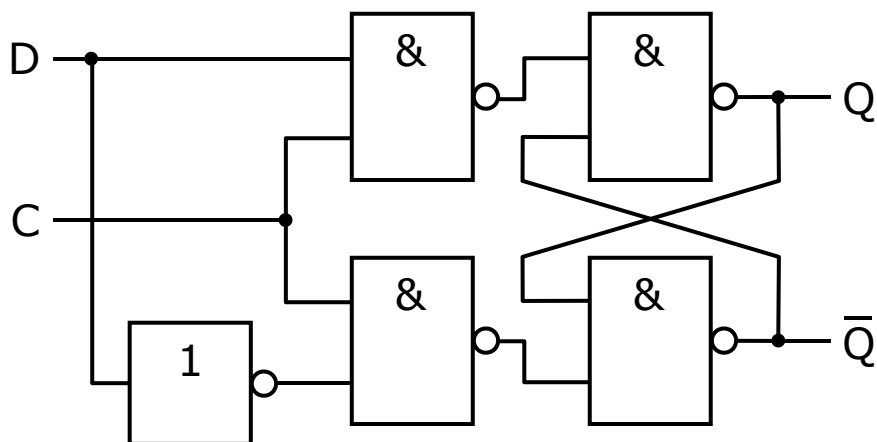
- tako onemogočimo nastop "prepovedane" kombinacije $\{S = 1, R = 1\}$



Sekvenčna vezja

Spominske celice: zapah D

- v takšni celici pri $C=1$ zavzame Q vrednost vhoda D , pri $C=0$ pa vhod D ne vpliva na zapisano vrednost; v celico torej zapisujemo 1-bitni podatek D (*angl. datum = podatek oz. data = podatki*)
- odtod tudi poimenovanje **zapah D** (*angl. **D latch***); ker ta celica pri $C=1$ prepušča vrednost D na izhod Q , jo nekateri imenujejo tudi **prepustni** ali "**prosojni**" **zapah** (*angl. **transparent latch***)



simbol

D	C	Q^*	$\bar{Q}^*$
0	0	Q	$\bar{Q}$
0	1	0	1
1	1	1	0

pravilnostna
tabela

Q	Q^*	D
0	0	0
0	1	1
1	0	0
1	1	1

vzbujalna
tabela



Sekvenčna vezja

Spominske celice: zapahi in flip-flopi

- nesinhronizirani zapahi so **celice brez proženja**, saj se na spremembe vhodov odzivajo vseskozi
- sinhronizirani zapahi se odzivajo le v tistih časovnih **intervalih**, ko je $C = 1$, zato jim pravimo tudi **celice s statičnim proženjem**
- odzivnost lahko še dodatno omejimo le na tiste **trenutke**, ko se vrednost C spremeni; tedaj govorimo o **dinamičnem proženju**, spominskim celicam s takšno lastnostjo pa pravimo **celice z dinamičnim proženjem** oz. krajše **flip-flopi**
- odzivnosti ob spremembi C z vrednosti 0 na 1 pravimo **proženje ob pozitivni fronti**, odzivnosti ob spremembi C z vrednosti 1 na 0 pa **proženje ob negativni fronti**

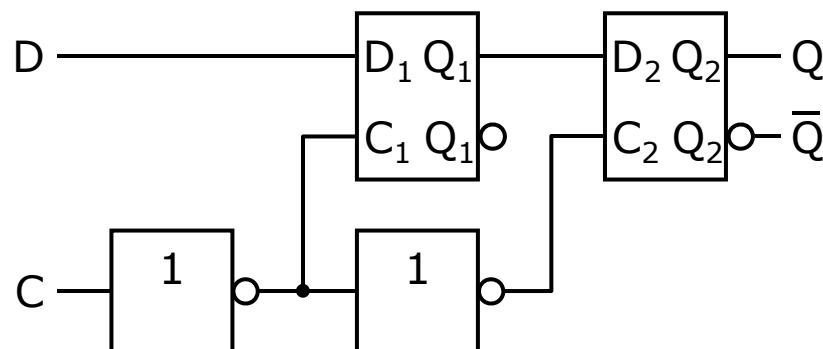
Sekvenčna vezja

Spominske celice: flip-flop D s proženjem ob pozitivni fronti

- **flip-flop D s proženjem ob pozitivni fronti** (*angl. positive-edge-triggered D flip-flop*) tvorita dva zapaha D in dva negatorja:

1) pri $C = 0$ postavimo $D = 1$:

- ker je $C_1 = 1$, levi zapah prepušča $D_1 = 1$ na $Q_1 = D_2 = 1$;
- ker je $C_2 = 0$, se na desnem zapahu ohranja dosedanji $Q_2 = Q$



2) ohranimo $D = 1$, postavimo $C = 1$:

- ker je $C_1 = 0$, se na levem zapahu ohranja $Q_1 = D_2 = 1$;
- ker je $C_2 = 1$, desni zapah prepušča $D_2 = 1$ na Q_2 in dobimo $Q_2 = Q = 1$

3) ohranimo $D = 1$, vrnimo C na vrednost 0:

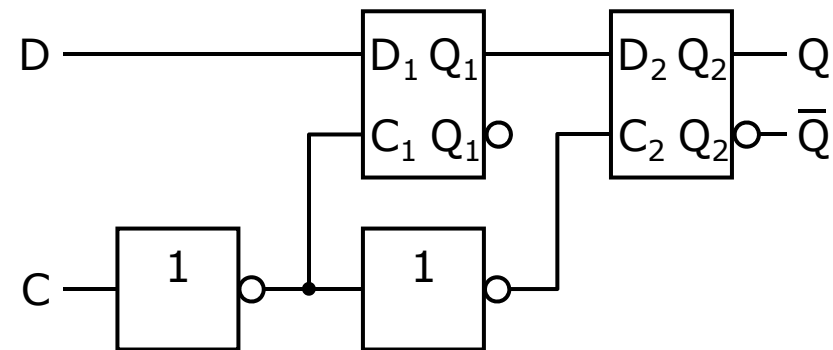
- ker je $C_1 = 1$, levi zapah prepušča $D_1 = 1$ na $Q_1 = D_2 = 1$ (kot prej);
- ker je $C_2 = 0$, se na desnem zapahu ohranja $Q_2 = Q = 1$;
- $D = 1$ se torej zapiše v flip-flop (t.j. v Q) **le ob prehodu C iz 0 na 1**

Sekvenčna vezja

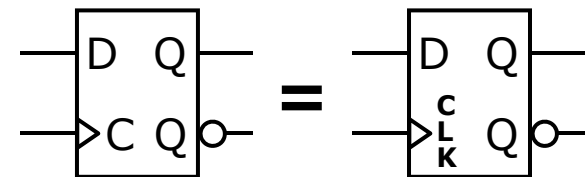
Spominske celice: flip-flop D s proženjem ob pozitivni fronti

- **flip-flop D s proženjem ob pozitivni fronti** (*angl. positive-edge-triggered D flip-flop*) tvorita dva zapaha D in dva negatorja:

4) če korake analize (1)-(3) ponovimo še pri $D=0$, ugotovimo, da se tudi $D=0$ zapiše v flip-flop (t.j. v Q) le ob prehodu C iz 0 na 1



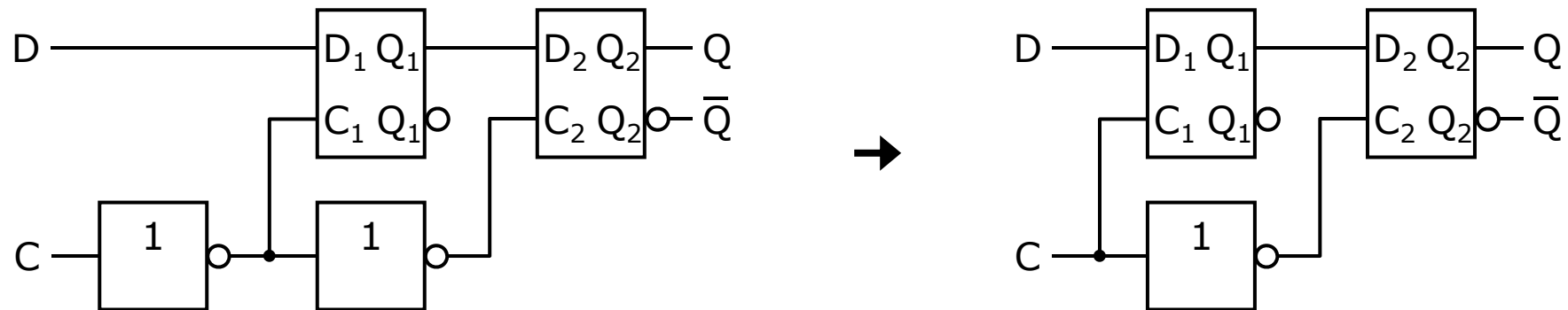
- prikazana spominska celica torej zapisuje vrednost D le ob prehodu C iz 0 na 1 – proženju ob pozitivni fronti signala C ;
- v simbolu flip-flopa takšno proženje odraža trikotnik ob vhodu C ;
- flip-flope pogosto prožimo periodično, t.j. s pulzi ure, zato spremenljivko C in/ali vhod v simbolu označujemo tudi s CLK (*angl. clock*)



Sekvenčna vezja

Spominske celice: flip-flop D s proženjem ob negativni fronti

- flip-flop D s proženjem ob negativni fronti** (*angl. negative-edge-triggered D flip-flop*) dobimo z odstranitvijo prvega negatorja:

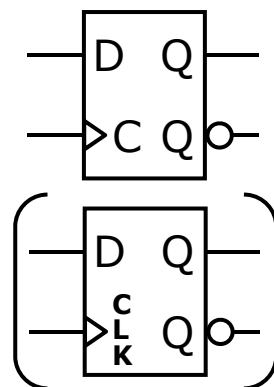
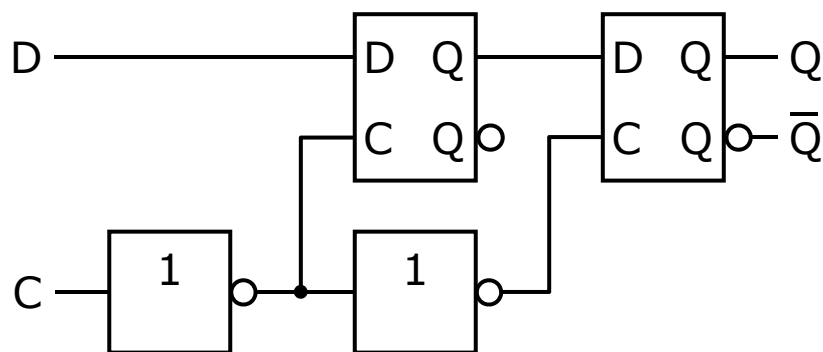


- razlog, da v flip-flopu D s proženjem ob poz. fronti signal C dovajamo na C_2 skozi dva negatorja in ne neposredno, je v zakasnitvah negatorjev; pri izvedbi z dvema negatorjema proženje (dvig C z 0 na 1) najprej zapre prvi zapah D in šele nato odpre drugega; pri neposrednem dovajanju C na C_2 pa bi bil vrstni red nasproten in bi bila nekaj časa odprta (t.j. bi prepuščala) oba zapaha



Sekvenčna vezja

Spominske celice: flip-flopa D s proženjem ob poz. in neg. fronti



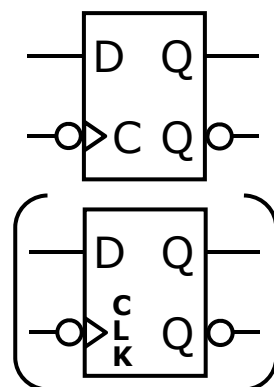
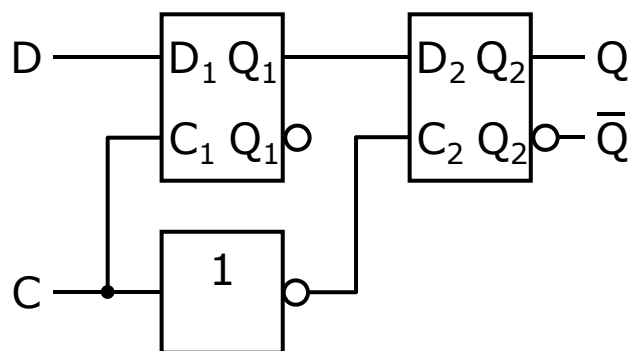
D	C	Q^*	$\overline{Q}^*$
0	0	Q	$\overline{Q}$
0	1	0	1
1	1	1	0

Q	Q^*	D
0	0	0
0	1	1
1	0	0
1	1	1

simbol

pravilnostna
tabela

vzbujalna
tabela



D	C	Q^*	$\overline{Q}^*$
0	0	Q	$\overline{Q}$
0	1	0	1
1	1	1	0

Q	Q^*	D
0	0	0
0	1	1
1	0	0
1	1	1



Sekvenčna vezja

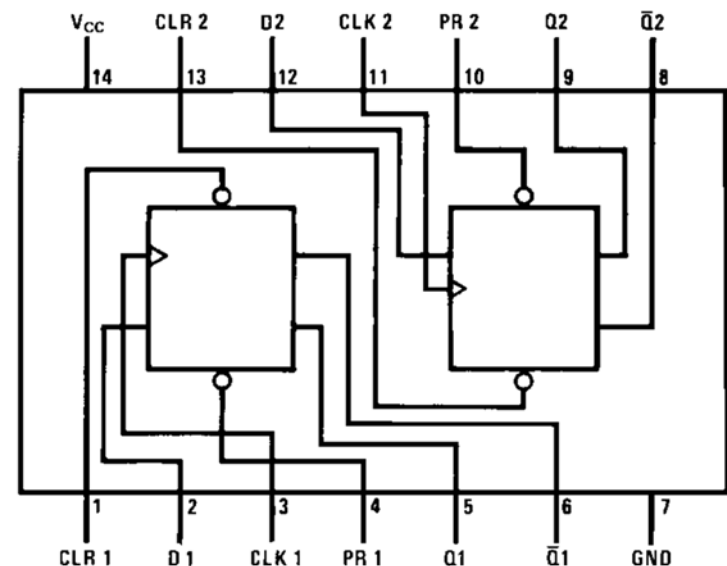
Spominske celice: zapahi D in flip-flopi D

- zapaha D in flip-flope D uporabljamo za izvajanje nalog, ki temeljijo na zapisovanju, pomnjenju (hranjenju) in branju podatkov
- med najpomembnejšimi vezji, ki vsebujejo zapaha in flip-flope D, so **pomikalni registri** in **statični bralno-pisalni pomnilnik (SRAM)**, najdemo pa jih tudi v nekaterih **števcih**; zgradbo in delovanje teh vezij bomo spoznali v nadaljevanju

integrirano vezje DM74LS74A (Fairchild) z dvema flip-flopoma D s proženjem ob pozitivni fronti in z možnostjo prednastavitve pomnjene vrednosti:

- s $PR = 0$ v flip-flop ne glede na C in D zapišemo $Q = 1$ (*angl. preset* = prednastaviti);
- s $CLR = 0$ v flip-flop ne glede na C in D zapišemo $Q = 0$ (*angl. clear* = izbrisati);

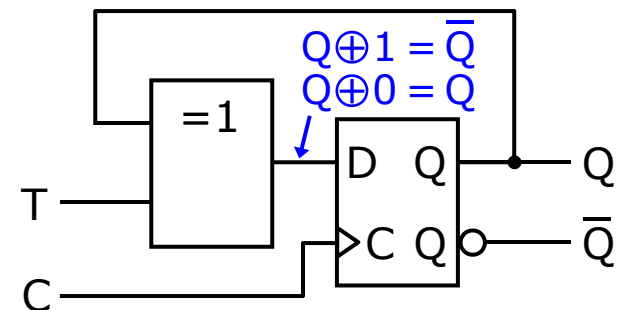
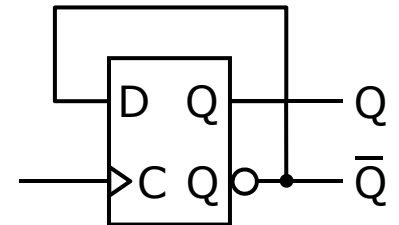
Ker PR in CLR omogočata spreminjanje Q ob poljubnem trenutku, pravimo, da sta to **asinhronska vhoda** flip-flopa



Sekvenčna vezja

Spominske celice: flip-flop T

- v doslej obravnavane spominske celice z ustrezno izbiro vrednosti na vseh vhodih zapisujemo vrednost $Q=0$ ali $Q=1$; včasih pa želimo trenutno stanje Q (za katero ni nujno, da ga poznamo) negirati
- spominsko celico s takšnim delovanjem lahko zgradimo iz flip-flopa D, ki mu izhod $\bar{Q}$ povežemo z vhomom D; stanje Q se tako negira ob vsaki pozitivni fronti signala na vhomu C:
- za nadzor nad odzivnostjo celice izvedemo povratno zvezo z izhoda Q skozi vrata XOR, na drugi vhom teh vrat pa dovedemo signal T (*angl. toggle* = preklapljati); tako dobimo flip-flop T (*angl. **T flip-flop***), ki se pri $T=1$ odziva na proženje (t.j. negira Q ob vsaki poz. fronti na vhomu C), pri $T=0$ pa vzdržuje obstoječe stanje Q

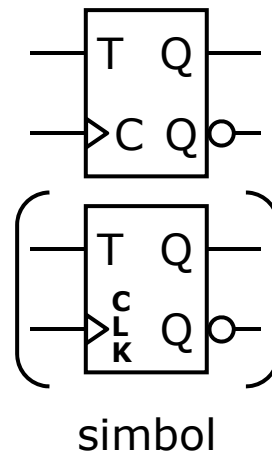
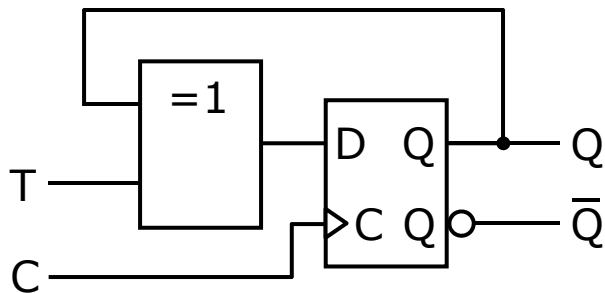




Sekvenčna vezja

Spominske celice: flip-flopi T

- flip-flopi T so najpogostejši gradniki **števc**



T	C	Q^*	$\bar{Q}^*$
0	0	Q	$\bar{Q}$
1	0	Q	$\bar{Q}$
1	1	$\bar{Q}$	Q
1	1	Q	$\bar{Q}$
1	1	Q	$\bar{Q}$

pravilnostna
tabela

Q	Q^*	T
0	0	0
0	1	1
1	0	1
1	1	0

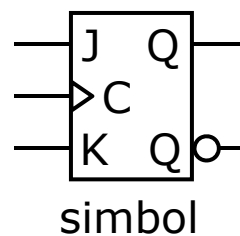
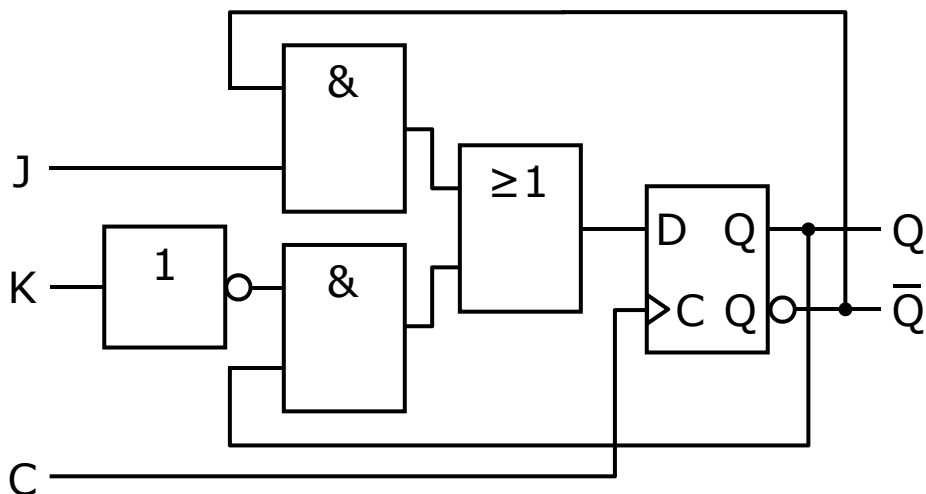
vzbujalna
tabela



Sekvenčna vezja

Spominske celice: flip-flop JK

- flip-flop JK** (*angl. JK flip-flop*, po njegovem izumitelju Jacku Kilbyju) združuje zmožnosti doslej obravnavanih celic, saj omogoča tako zapis vrednosti (stanja) $Q=0$ ali $Q=1$ kot negiranje trenutnega stanja Q



J	K	C	Q^*	$\bar{Q}^*$
0	0	0	Q	$\bar{Q}$
1	1	0	Q	$\bar{Q}$
0	0	1	Q	$\bar{Q}$
0	1	1	0	1
1	0	1	1	0
1	1	1	$\bar{Q}$	Q

pravilnostna
tabela

Q	Q^*	J	K
0	0	0	0
0	1	1	0
1	0	0	1
1	1	0	0

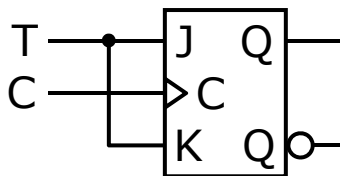
vzbujalna
tabela

- flip-flope JK najdemo v **serijskih seštevalnikih** in nekaterih števcih, a se v števcih danes uporabljajo redkeje od flip-flopov D in T

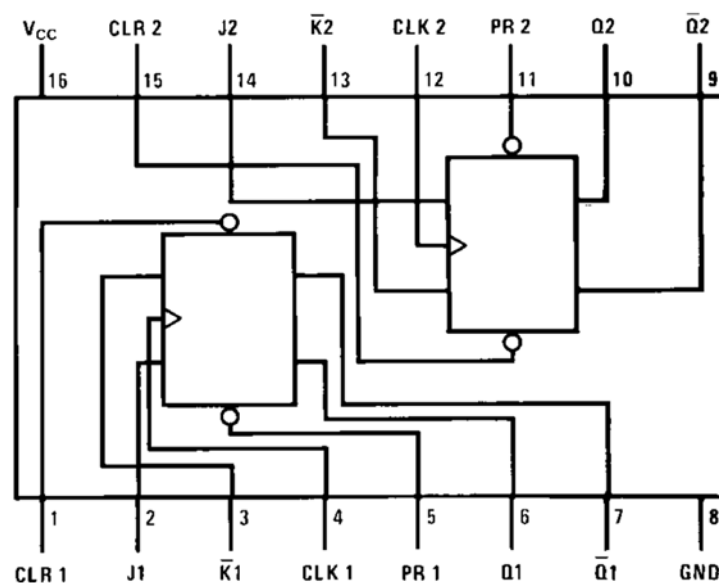
Sekvenčna vezja

Spominske celice: flip-flopi JK

- zaradi svoje prilagodljivosti so flip-flopi JK uporabni v fazi razvoja in testiranja sekvenčnih vezij (npr. na protoboardu), z njimi pa lahko izvedemo tudi druge flip-flope:



flip-flop T, izveden s flip-flopom JK



integrirano vezje DM74LS109A (Fairchild) z dvema flip-flopoma JK (t.j. K v negativni logiki) s proženjem ob poz. fronti (CLK) in možnostjo prednastavitve pomnjene vrednosti Q (PR za zapis 1, CLR za zapis 0)



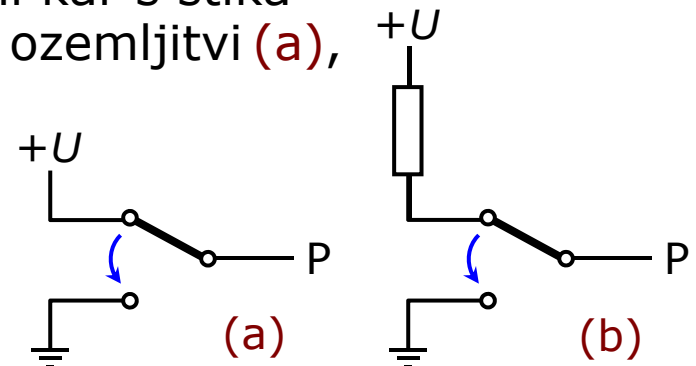
Sekvenčna vezja

Stabilizatorji preklopnikov

- za vnos podatkov v digitalne sisteme, pa tudi za njihov zagon in zaustavitev, pogosto uporabljamo mehanske preklopnike: stikala, tipke in rotacijske preklopnike:



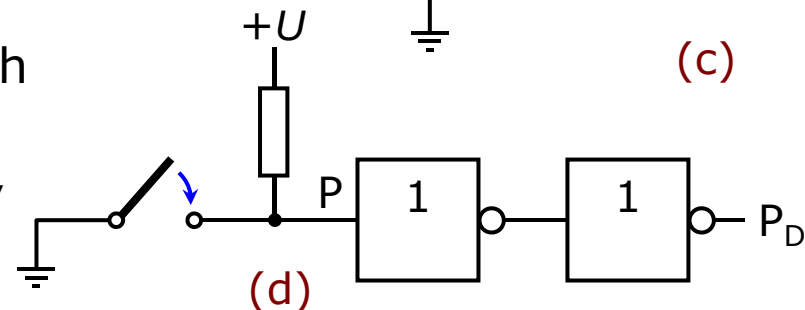
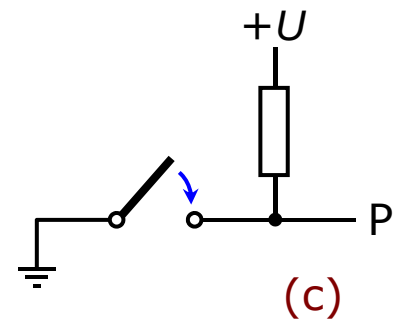
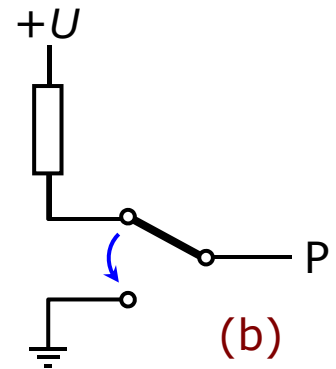
- načeloma bi logično 1 oz. 0 lahko dovedli kar s stikalom s sponkama na ustrezni napetosti in ozemljitvi (a), a priklop neposredno na vir $+U$ lahko privede do prevelikih tokov v digitalne gradnike med njihovim preklapljanjem
- za omejitev teh tokov med vir $+U$ in sponko vgradimo upor (nekaj $k\Omega$) (b)



Sekvenčna vezja

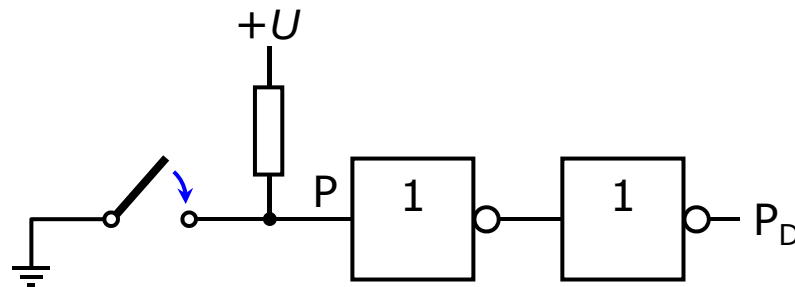
Stabilizatorji preklopnikov

- ker je vhodna upornost digitalnih gradnikov v stacionarnih razmerah zelo visoka (pri CMOS več $G\Omega$), je padec napetosti na uporu zanemarljiv in $P \approx +U$
- v času, ko se takšno stikalo nahaja med sponkama, napetost P ni določena, pri "plavajoči" vhodni napetosti pa digitalni gradniki delujejo nezanesljivo
- nedoločenost lahko odpravimo tako, da uporabimo stikalo z le eno sponko (t.j. enopolno stikalo), vejo z uporom pa prestavimo za stikalo (c); tako dobi upor pri razklenjenem stikalu vlogo "dvigovalca" nivoja napetosti P (angl. **pull-up resistor**)
- za diskretizacijo izhodnih napetostnih nivojev dodamo na izhod stikala še negator za negativno izhodno logiko, ali dva negatorja za pozitivno (d)

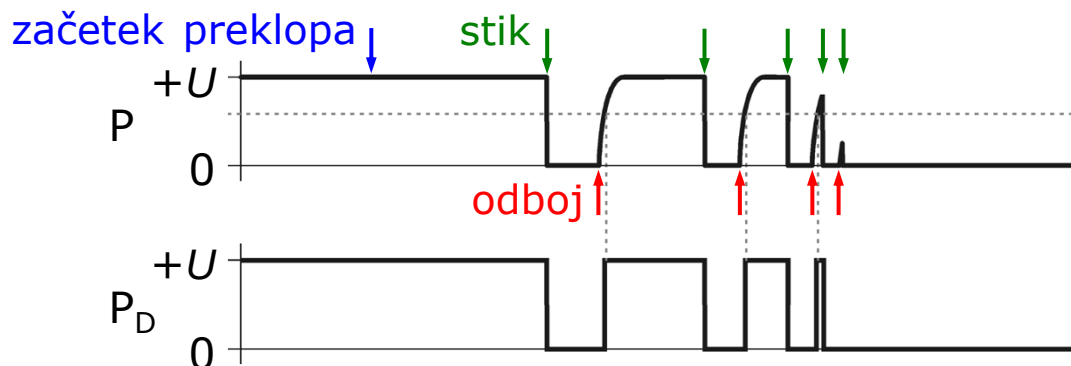


Sekvenčna vezja

Stabilizatorji preklopnikov



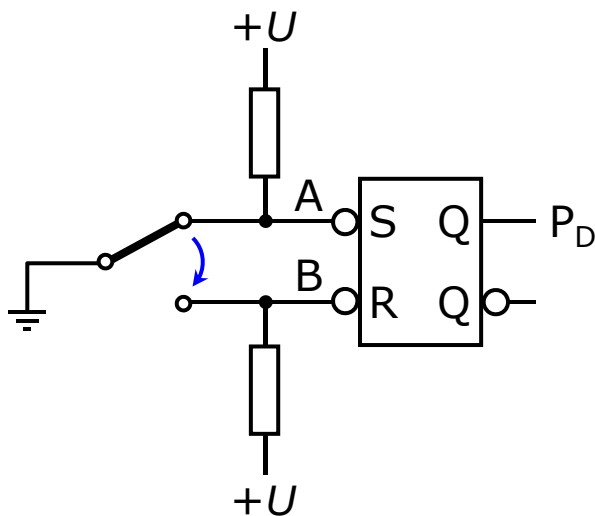
- tudi delovanje takšnega preklopnika pa ni brezhibno, saj lahko zaradi prožnosti stikala ob preklupu pride do kratkotrajnih vibracij, v času katerih ($\sim 5\text{-}30\text{ ms}$) se stik večkrat prekrine in spet vzpostavi:



Sekvenčna vezja

Stabilizatorji preklopnikov

- ker so zakasnitve v digitalnih gradnikih znatno krajše od časa vibriranja preklopnika, se ti gradniki odzovejo na vsak stik in odboj posebej
- to odpravimo s **stabilizatorjem preklopnika** (*angl. switch debouncer* oz. **debounce circuit**), ki ga zgradimo z zapahom $\overline{S}\overline{R}$

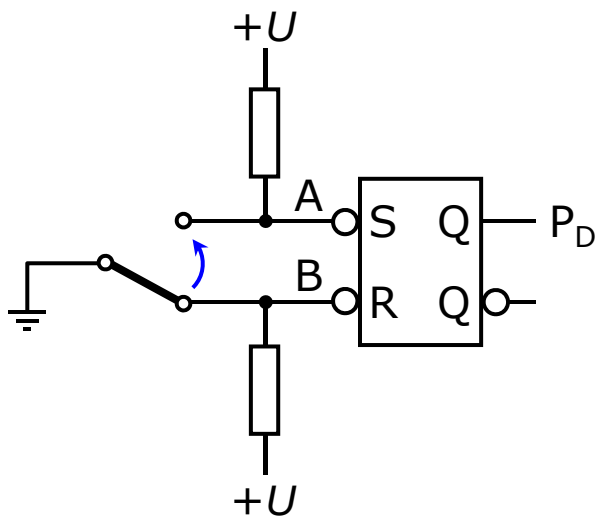


- 1) ko je stikalo v zgornji legi, je $A = 0$ in $B = +U$, zato je $\overline{S} = 0$ in $\overline{R} = 1$, t.j. $S = 1$ in $R = 0$, zato se na izhodu vzdržuje stanje $Q = P_D = 1$
- 2) ko se stikalo giblje navzdol in zgornja sponka izgubi stik z ozemljitvijo, postane $A = +U$ in s tem $\overline{S} = 1$, t.j. $S = 0$; ker je $R = 0$, se na izhodu ohranja stanje $Q = P_D = 1$
- 3) ko se stikalo dotakne spodnje lege, postane $B = 0$ in s tem $\overline{R} = 0$, t.j. $R = 1$, zato $Q = P_D = 0$
- 4) če se stikalo odbije, spodnja sponka začasno izgubi stik z ozemljitvijo, zato postane $B = +U$ in $R = 0$, a ker je $S = 0$, se ohranja $Q = P_D = 0$

Sekvenčna vezja

Stabilizatorji preklopnikov

- ker so zakasnitve v digitalnih gradnikih znatno krajše od časa vibriranja preklopnika, se ti gradniki odzovejo na vsak stik in odboj posebej
- to odpravimo s **stabilizatorjem preklopnika** (*angl. **switch debouncer** oz. **debounce circuit***), ki ga zgradimo z zapahom $\overline{S}\overline{R}$

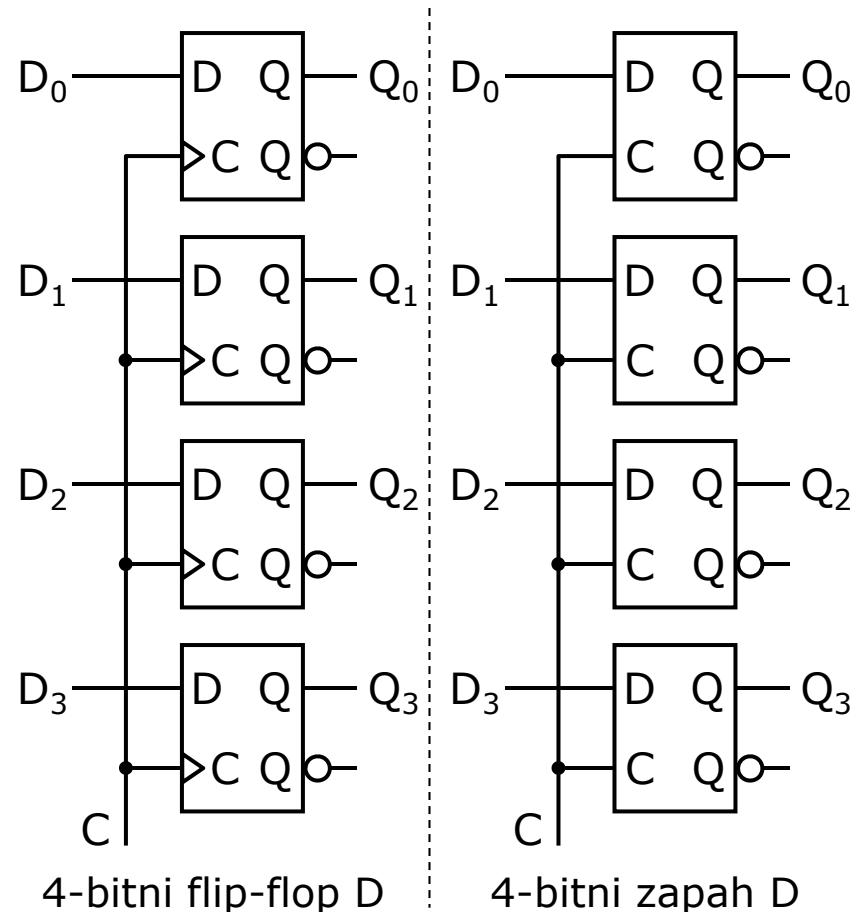


- če korake (1)-(4) ponovimo še za preklon stikala iz spodnje lege v zgornjo, ugotovimo, da se tudi v tem primeru že ob prvem dotiku stikala z zgornjo sponko vzpostavi stanje $Q = P_D = 1$, ki se ohrani tudi ob odbojih stikala
- celotna analiza (1)-(5) seveda velja le, če odboji niso nikoli tako izraziti, da bi se stikalo dotaknilo nasprotne sponke; to zagotovimo z ustreznim dimenzioniranjem stikala, prožnostjo vzmeti v njem ipd.

Sekvenčna vezja

Registri

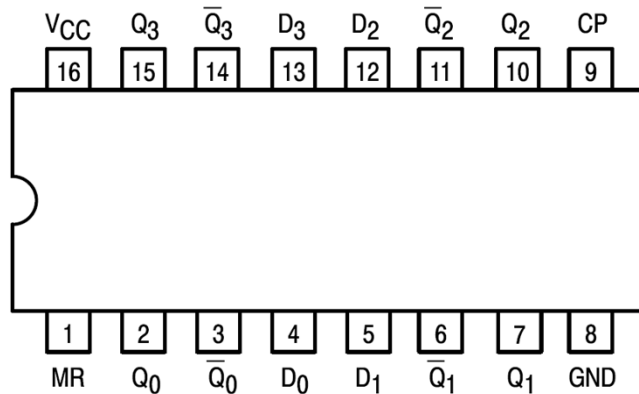
- **register** (*angl. register*) je skupina dveh ali več spominskih celic D
- iz n celic dobimo **n -bitni register**; vanj najpogosteje zapišemo en n -bitni podatek, lahko pa tudi n neodvisnih 1-bitnih podatkov
- ločimo **registre z dinamičnim proženjem** (iz flip-flopov D, zato imenovani tudi **n -bitni flip-flopi D**) in **reg. s statičnim proženjem** (iz zapahov D, **n -bitni zapahi D**)
- prek vhoda C sinhroniziramo zapis podatkov v vse celice (bite) registra



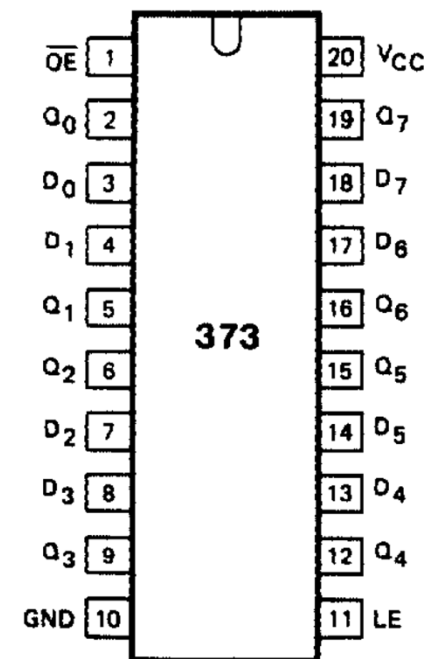
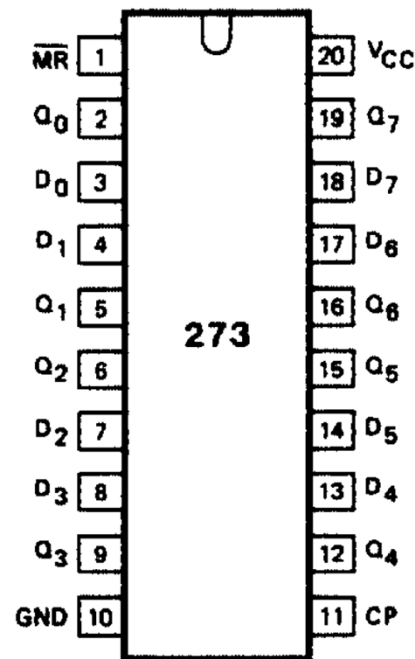


Sekvenčna vezja

Registri



4-bitni flip-flop D s proženjem ob poz. fronti 74HC175N (Motorola);
MR: izbris vseh Q (master reset)
CP: proženje (control/clock pulse)

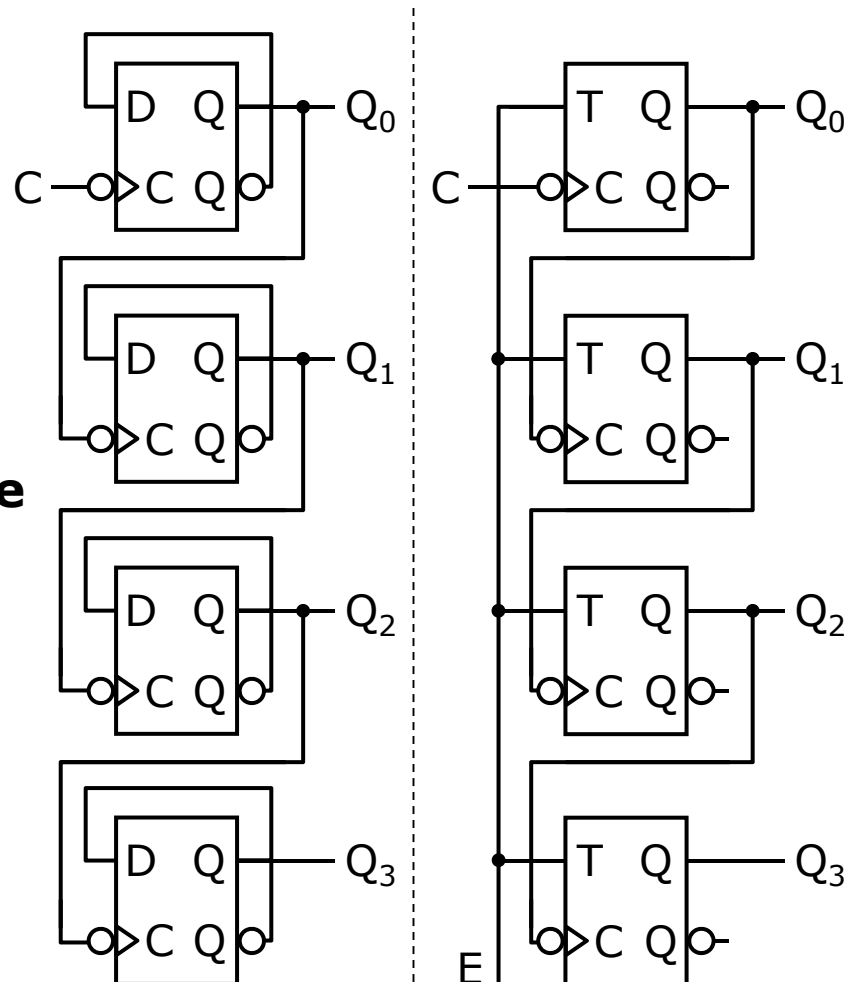


8-bitni flip-flop D 74HC273N in 8-bitni zapah D 74HC373N (oba Philips); zapis v celice zapaha je mogoč pri LE = 1 (latch enable), branje izhodov pa pri OE = 0 (output enable, neg. logika)

Sekvenčna vezja

Števci

- n -bitni **števec** (*angl. counter*) je skupina n spominskih celic, katerih množica stanj $\{Q_0, Q_1, \dots, Q_{n-1}\}$ predstavlja n -bitno število, ki se ob vsakem proženju poveča za 1
- n -bitni **serijski števec** (*angl. ripple counter*) zgradimo iz n flip-flopov D ali T s proženjem ob negativni fronti, v katerih izhode Q kaskadno povežemo s prožilnimi vhodi C
- izvedba s flip-flopi D prišteje 1 ob vsaki neg. fronti C, izvedba s flip-flopi T pa le, če je ob neg. fronti C $E = 1$ (*angl. enable = aktivirati*)



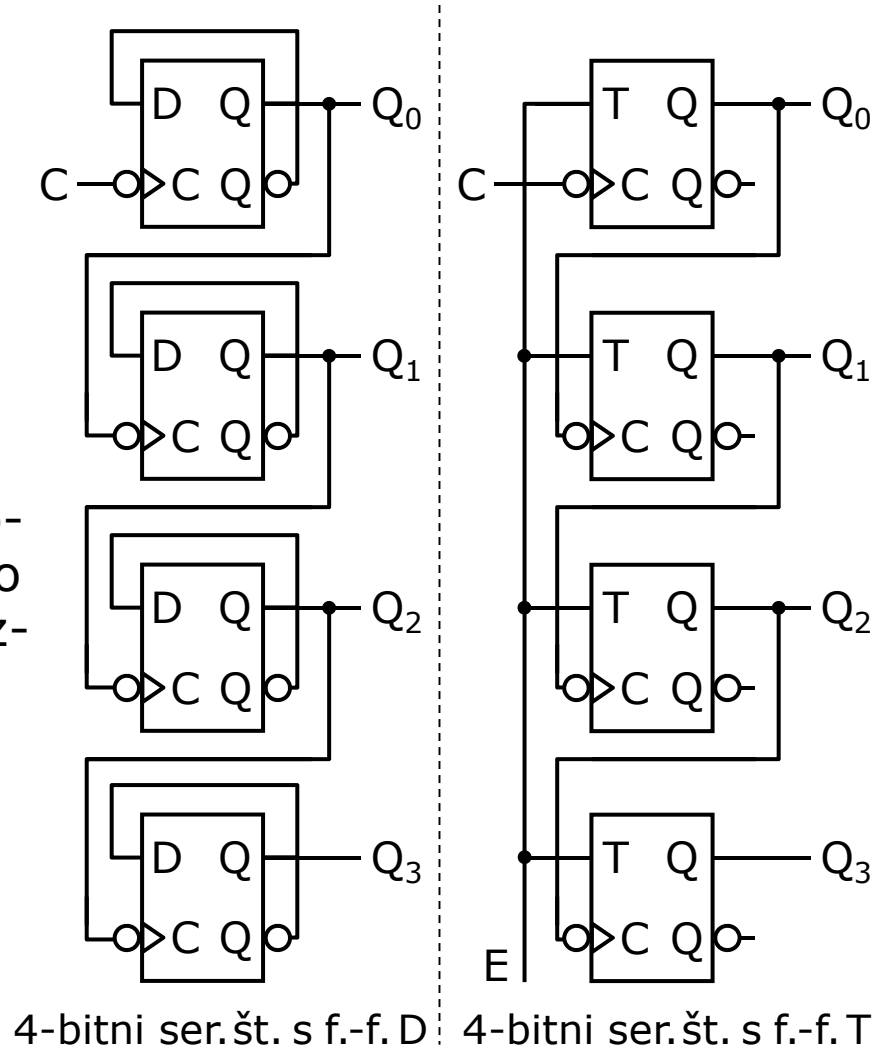
4-bitni ser.št. s f.-f. D

4-bitni ser.št. s f.-f. T

Sekvenčna vezja

Števci

- če izhod Q_3 enega 4-bitnega serijskega števca povežemo z vhodom C drugega takšnega števca, dobimo 8-bitni serijski števec; iz treh 4-bitnih dobimo tako 12-bitnega, iz štirih 16-bitnega itd.
- serijski števci so preprosti, a podobno kot serijski seštevalniki, ki smo jih opisali med kombinacijskimi vezji, so počasni, saj Q_0 vpliva na Q_1 šele po eni zakasnitvi flip-flopa, na Q_2 po še eni zakasnitvi itd.
- serijski števci so **asinhronski**; čeprav ima v števcu vsak flip-flop svoj vhod C, proženje teh vhodov ne poteka sinhrono, t.j. sočasno

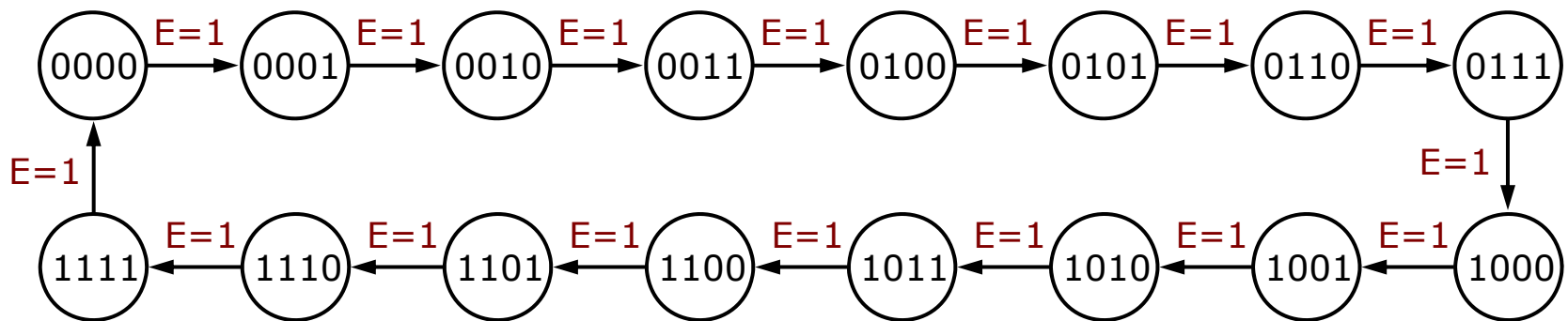




Sekvenčna vezja

Števci in diagrami stanj

- eno od orodij za analizo delovanja sekvenčnih vezij je t.i. **diagram stanj** oz. **diagram prehajanja stanj** (*angl. state diagram*), ki podaja vsa stanja vezja in vse prehode med njimi
- ker si pri števcih stanja vselej sledijo v enakem vrstnem redu, vsi prehodi pa se zgodijo ob enakem pogoju, so diagrami stanj števcov preprosti; diagram stanj 4-bitnega serijskega števca je takšen:

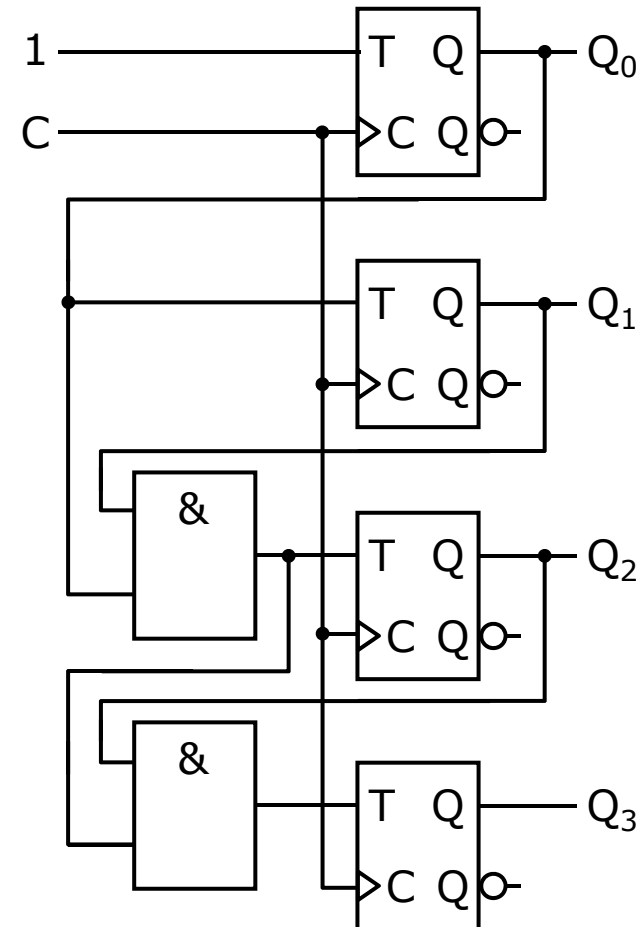


- če so za posamezen prehod poleg proženja potrebni dodatni pogoji, jih dopišemo k puščici; pri ser. števcu s flip-flopi D teh ni, saj šteje ob vsakem proženju, pri ser. števcu s flip-flopi T pa je pogoj **E = 1**

Sekvenčna vezja

Števci

- **sinhronski števec** (*angl. synchronous counter*) zgradimo iz flip-flopov T s proženjem ob pozitivni fronti, njihove vhode C pa povežemo na skupni signal C
- ker pri štetju negacija posameznega bita nastopi le, če imajo vsi nižje ležeči biti vrednost 1, izhode Q vežemo kaskadno na vhode T celic, ki vsebujejo višje ležeče bite; izhod Q_0 vežemo neposredno, vse ostale pa prek vrat AND

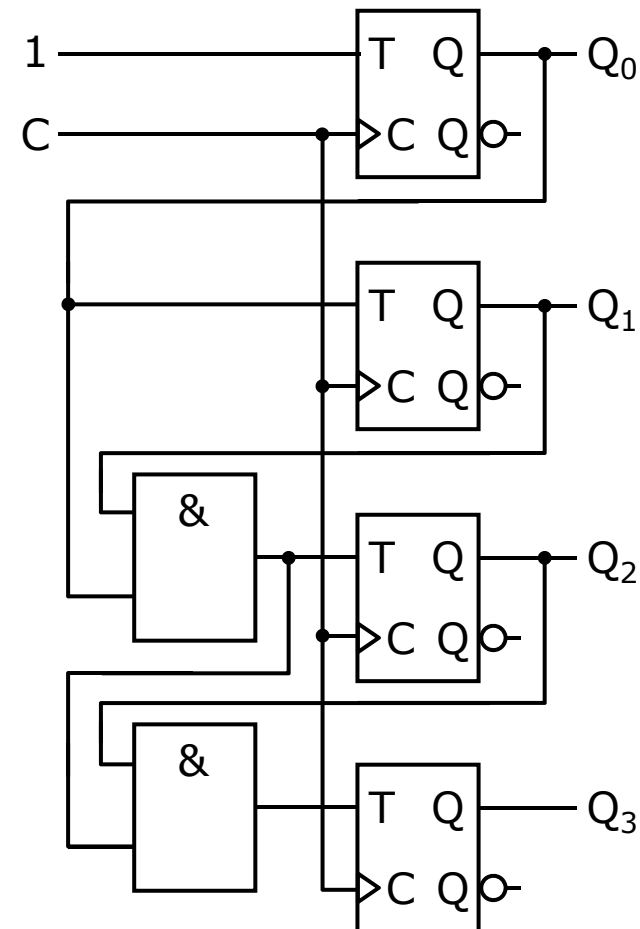


4-bitni sinhronski števec

Sekvenčna vezja

Števci

- proženje flip-flopov in s tem spreminjanje stanj Q je sočasno, zato je tak števec sinhronski, a spremembe na vratih AND potujejo kaskadno; če naslednje proženje nastopi prehitro, vrednosti na vseh T , do katerih se te spremembe še niso razširile, lahko še niso pravilne in pride do napake
- za pravilno delovanje takšnega števca mora biti na vhodu T zgornjega flip-flopa enica, tak števec pa (podobno kot serijski števec, izveden s flip-flopi D) tudi ne omogoča deaktivacije štetja (ni vhoda E)

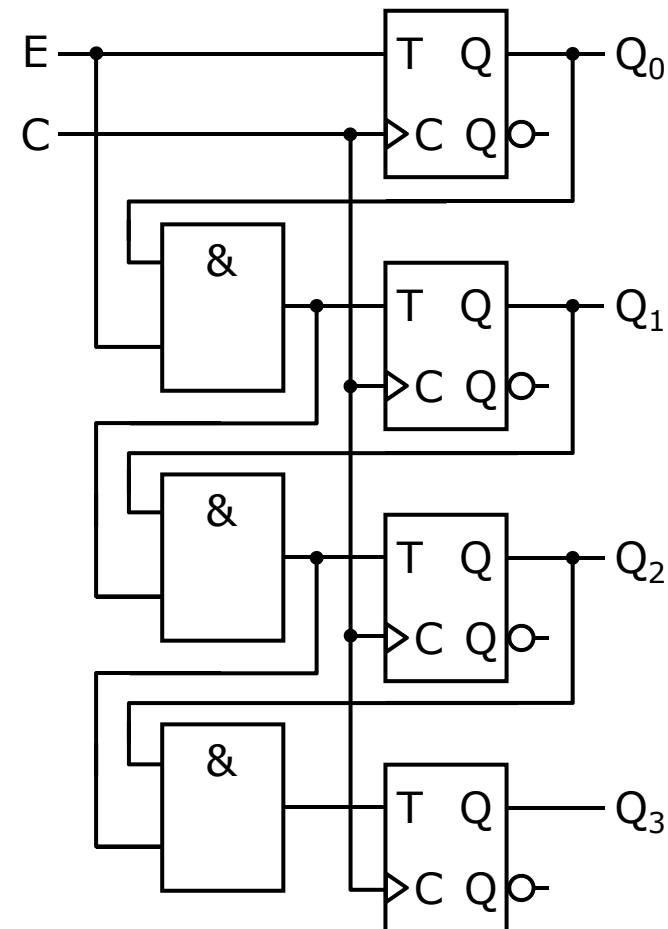


4-bitni sinhronski števec

Sekvenčna vezja

Števci

- deaktivacijo omogočimo tako, da tudi kaskadno vezavo Q_0 izvedemo skozi vrata AND, na vhod zgornjega flip-flopa T in hkrati na drugi vhod dodanih vrat AND pa privedemo signal E
- tako dobimo **sinhronski števec z aktivacijo** (*angl. synchronous counter with enable*), ki šteje le pri $E = 1$
- spremembe se skozi vrata AND tudi tu širijo kaskadno, zato prekratek zamik med zaporednimi proženji (oz. previsoka frekvenca ure pri periodičnem proženju) privede do napak pri štetju

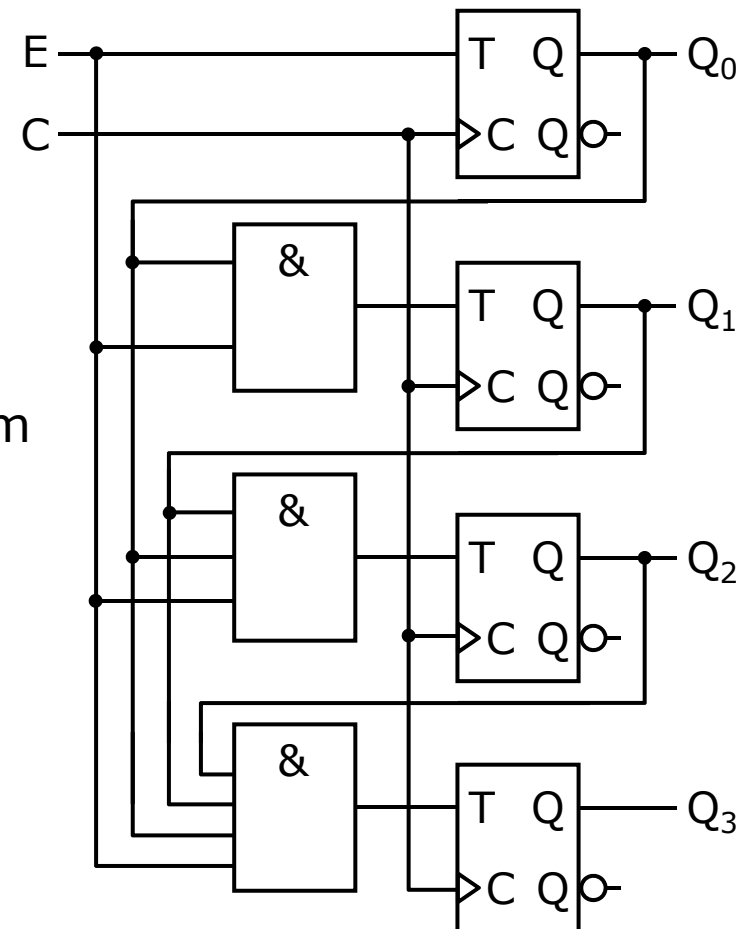


4-bitni sinhr. števec z aktivacijo

Sekvenčna vezja

Števci

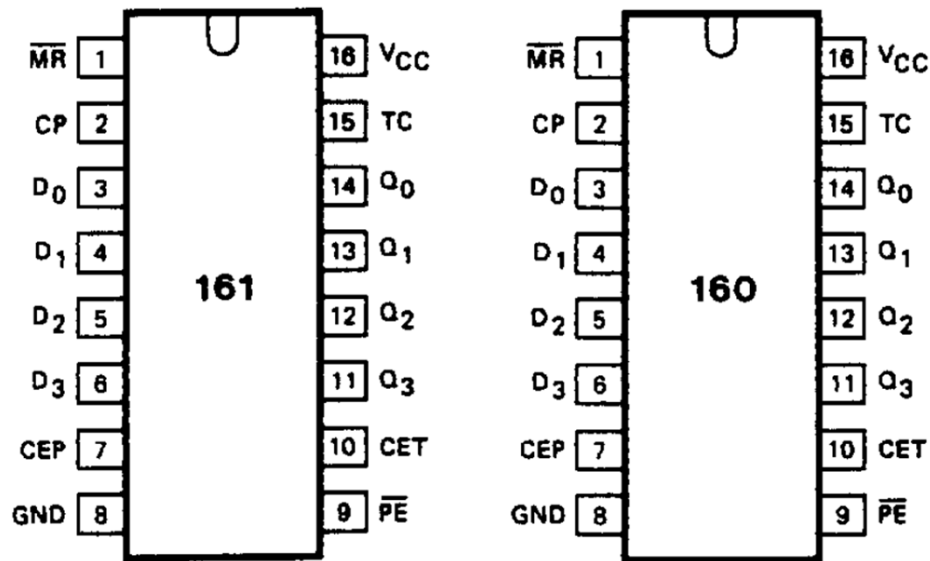
- za povsem paralelno delovanje števca se moramo izogniti kaskadnemu širjenju podatkov o vrednostih nižje ležečih bitov, zato v vsak flip-flop dovedemo podatke o vseh nižje ležečih bitih
- za to potrebujemo vrata AND z ustreznim številom vhodov: zgornja so tudi tukaj 2-vhodna (E in Q_0), sledijo 3-vhodna (E, Q_0, Q_1), nato 4-vhodna itd.
- tako dobimo **sinhronski paralelni števec z aktivacijo** (*angl. synchronous parallel counter with enable*), ki je najhitrejša, a hkrati tudi najobsežnejša izvedba digitalnega števca



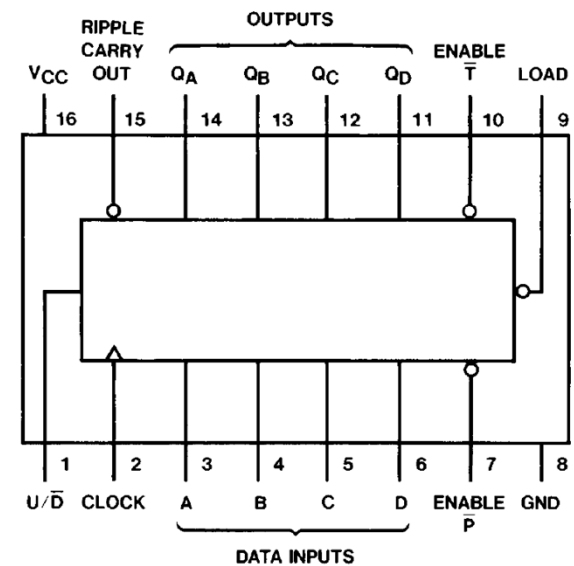
4-bitni sinhr. par. števec z aktivacijo

Sekvenčna vezja

Števci



4-bitna sinhronska paralelna števec: 74HCT161 z binarnim štetjem (0-1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-0) in 74HCT160 z BCD štetjem (0-1-2-3-4-5-6-7-8-9-0) (oba Philips);
CEP: aktivacija štetja (count enable parallel)
PE: asinhronska prednastavitev števca z vhodov D_0 - D_3 (parallel enable, pri $PE=0$ prednastavitev)

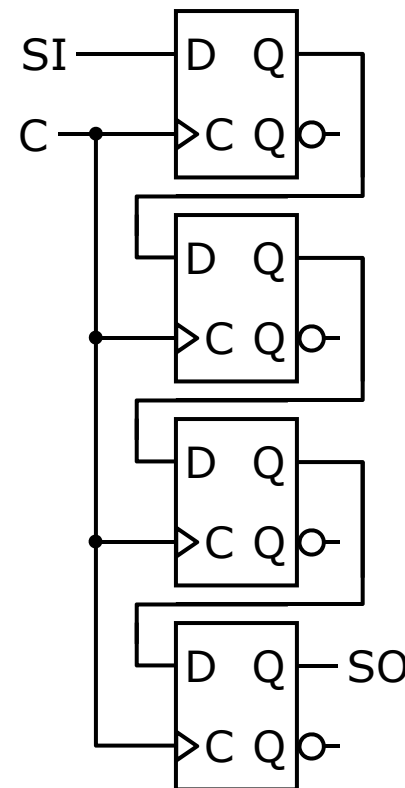


4-bitni sinhronski paralelni števec s štetjem navzgor in navzdol DM-74LS169 (Natl. Semiconductor);
 $U/\overline{D} = 1$: štetje navzgor (0-1-2-...)
 $U/\overline{D} = 0$: navzdol (0-15-14-13-...)

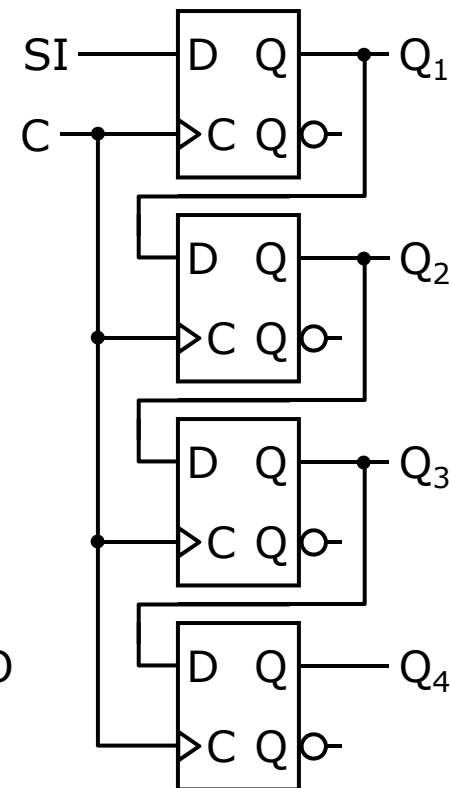
Sekvenčna vezja

Pomikalni registri

- **pomikalni register** (*angl. shift register*) je register, zgrajen tako, da se podatki v njem ob vsakem proženju pomaknejo za en bit
- **serijski pomikalni register** (*angl. serial-in, serial-out shift register, SISO shift register*) in **serijsko-paralelni pomikalni register** (*angl. serial-in, parallel-out shift register, SIPO shift register*) zgradimo iz kaskade flip-flopov D



4-bitni pomikalni register SISO



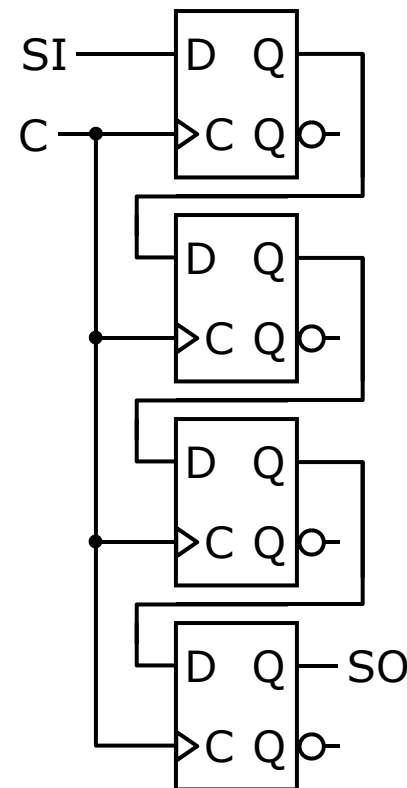
4-bitni pomikalni register SIPO



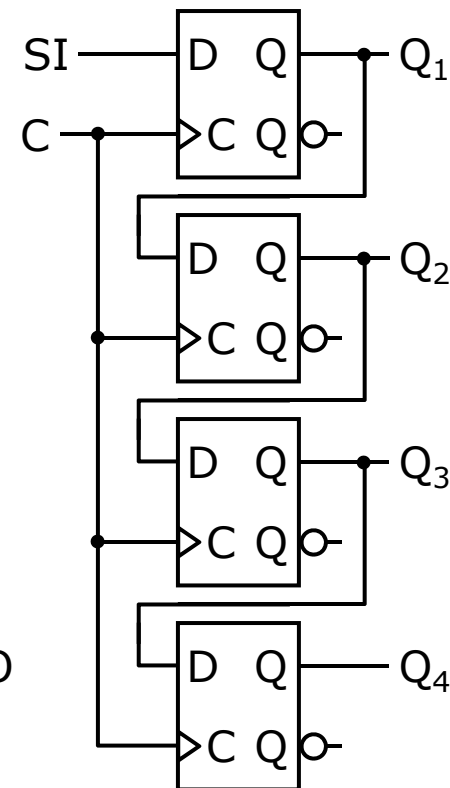
Sekvenčna vezja

Pomikalni registri

- v pomikalnem registru SISO je navzven dosegljiv le en izhod, uporaben pa je za zakasnitev signala; če na C dovedemo periodično proženje, izhod n -bitnega registra zaostaja za vhomom za n pulzov ure
- v pomikalnem registru SIPO so navzven dosegljivi izhodi iz vseh flip-flopov, uporabljamo pa jih predvsem pri pretvorbah podatkov iz serijske v paralelno obliko



4-bitni pomikalni
register SISO

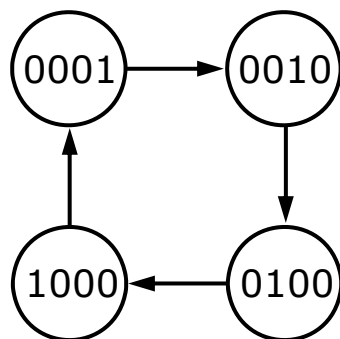


4-bitni pomikalni
register SIPO

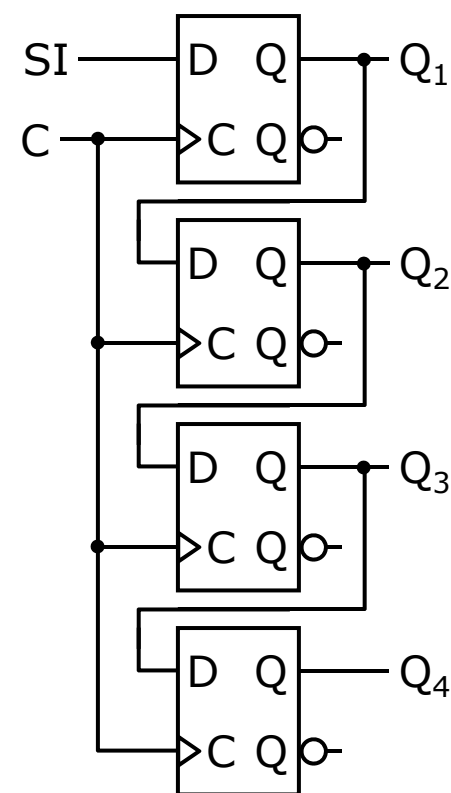
Sekvenčna vezja

Pomikalni registri in krožni števeci

- s pomikalnim registrom lahko zgradimo tudi **krožni števec** (*angl. ring counter*), katerega stanje tvorijo ničle in ena enica, ki se ob vsakem proženju pomakne za eno mesto oz. en bit;
- diagram stanj 4-bitnega krožnega števca torej izgleda takole:



- izvedemo ga s 4-bitnim registrom SIPO, v katerega ob prvih štirih proženjih zapišemo zaporedje $\{1,0,0,0\}$, nato pa SI povežemo z izhodom Q_4

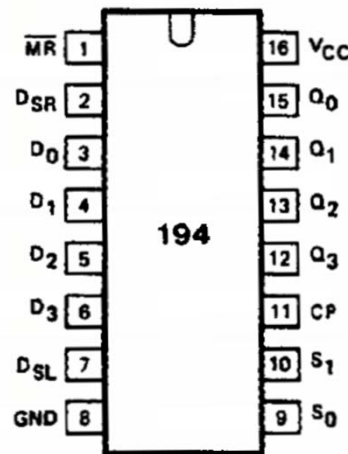


4-bitni pomikalni
register SIPO



Sekvenčna vezja

Pomikalni registri in krožni števc



4-bitni dvosmerni pomikalni register
74HCT194 (Philips);

$S_0 = 0, S_1 = 0$: pomikanje deaktivirano

$S_0 = 0, S_1 = 1$: pomikanje v levo
(branje podatka iz D_{SL})

$S_0 = 1, S_1 = 0$: pomikanje v desno
(branje podatka iz D_{SR})

$S_0 = 1, S_1 = 1$: paralelni zapis vhodov D_0 - D_3



Sekvenčna vezja

Analiza in sinteza sekvenčnih vezij

- **analiza sekvenčnega vezja** je postopek, s katerim pridemo do razumevanja vezja (t.j. iz zgradbe vezja dobimo opis delovanja), **sinteza** pa postopek, s katerim načrtujemo vezje, ki bo izvajalo zastavljene naloge (t.j. iz opisa delovanja dobimo zgradbo vezja)
- oba postopka sta najpreprostejša v primeru **sinhronskih sekvenčnih vezij s periodičnim dinamičnim proženjem**
- takšnim vezjem bomo odslej rekli **sekvenčna vezja z uro** (*angl. clocked sequential circuits*); besedi "sinhronska" in "dinamično" bomo opustili, saj je praviloma ura namenjena sinhronizaciji, proženje s pulzi ure pa je dinamično (t.j. celice v teh vezjih so flip-flopi)
- v shematskih prikazih teh vezij bomo signal, ki dovaja pulze ure, označili s "CLK", v simbolih flip-flopov D, T in JK v vezjih pa bomo obdržali splošnejši "C" (control/clock/count/...), saj so sami flip-flopi enaki ne glede na to, ali je proženje periodično ali aperiodično



Sekvenčna vezja

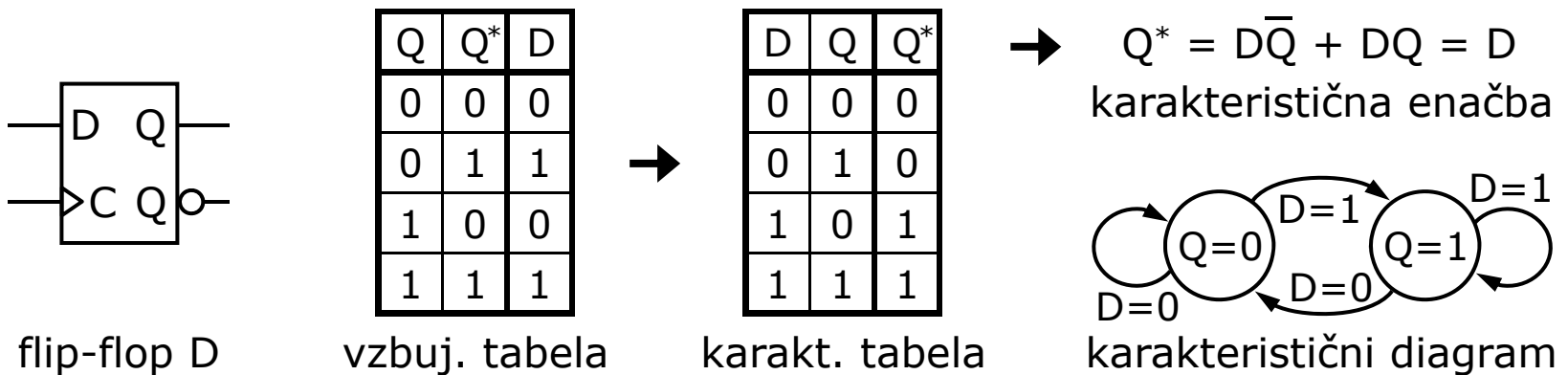
Analiza sekvenčnih vezij

- pri analizi delovanja sekvenčnega vezja so nam v pomoč tabele, enačbe in diagrami, ki opisujejo delovanje vezja *na nivoju posameznih flip-flopov*:
 - **vzbujalna tabela flip-flopa**, ki smo jo že spoznali pri obravnavi posameznih flip-flopov
 - **karakteristična tabela flip-flopa**,
 - **karakteristična enačba flip-flopa** in
 - **karakteristični diagram flip-flopa**
- ... ter *na nivoju vezja kot celote*:
 - **vzbujalne enačbe sekvenčnega vezja**,
 - **enačbe stanj in izhodov sekvenčnega vezja**,
 - **tabela stanj in izhodov sekvenčnega vezja** in
 - **diagram stanj sekvenčnega vezja**, preprosta primera katerega smo spoznali pri opisu delovanja navadnega in krožnega števca

Sekvenčna vezja

Karakteristična tabela, enačba in diagram flip-flopa D

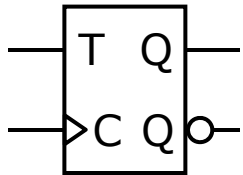
- **karakteristična tabela flip-flopa** podaja njegovo naslednje stanje (Q^*) kot funkcijo vhodov ($D / T / J$ in K) in trenutnega stanja (Q); dobimo jo s predelavo iz vzbujaalne tabele, v kateri prestavimo stolpce tako, da si sledijo vhodi, trenutno stanje in naslednje stanje
- **karakteristično enačbo flip-flopa** dobimo, če iz karakteristične tabele flip-flopa (tako kot iz pravilnostne tabele preklopne funkcije) zapišemo vsoto mintermov in rezultat poenostavimo
- **karakteristični diagram** podaja pravila prehajanja stanj še grafično





Sekvenčna vezja

Karakteristična tabela, enačba in diagram flip-flopa T



flip-flop T

Q	Q*	T
0	0	0
0	1	1
1	0	1
1	1	0

vzbuj. tabela



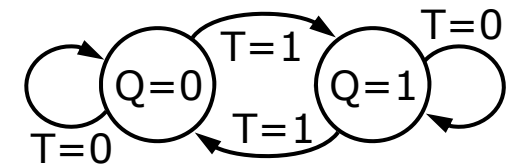
T	Q	Q*
0	0	0
0	1	1
1	0	1
1	1	0

karakt. tabela



$$Q^* = \bar{T}Q + T\bar{Q} = T \oplus Q$$

karakteristična enačba

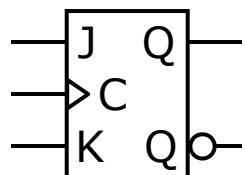


karakteristični diagram



Sekvenčna vezja

Karakteristična tabela, enačba in diagram flip-flopa JK



flip-flop JK

Q	Q*	J	K
0	0	0	0
0	1	1	0
1	0	0	1
1	1	1	1

vzbuj. tabela



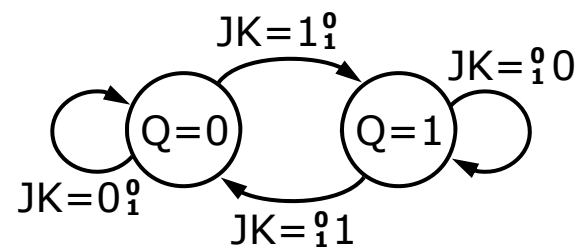
J	K	Q	Q*
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

karakt. tabela



$$Q^* = J\bar{Q} + \bar{K}Q$$

karakteristična enačba



karakteristični diagram



Sekvenčna vezja

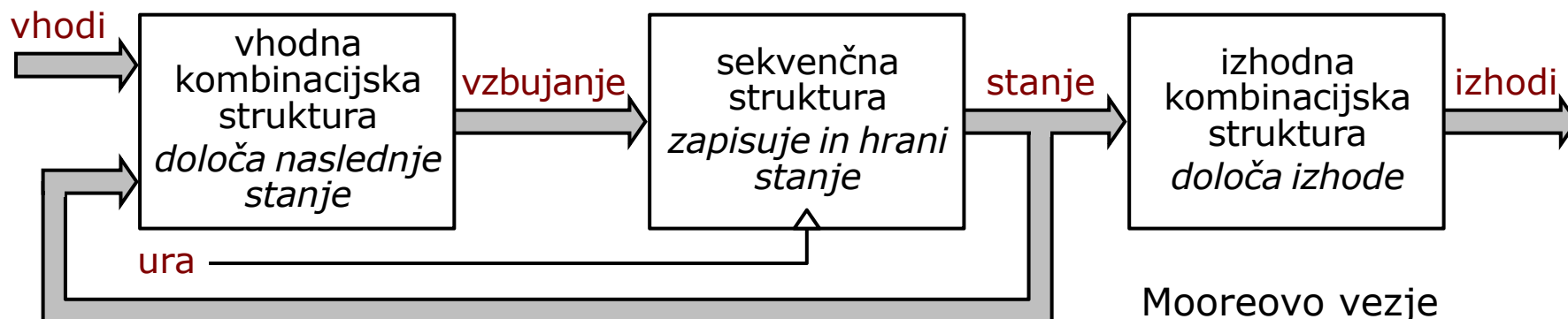
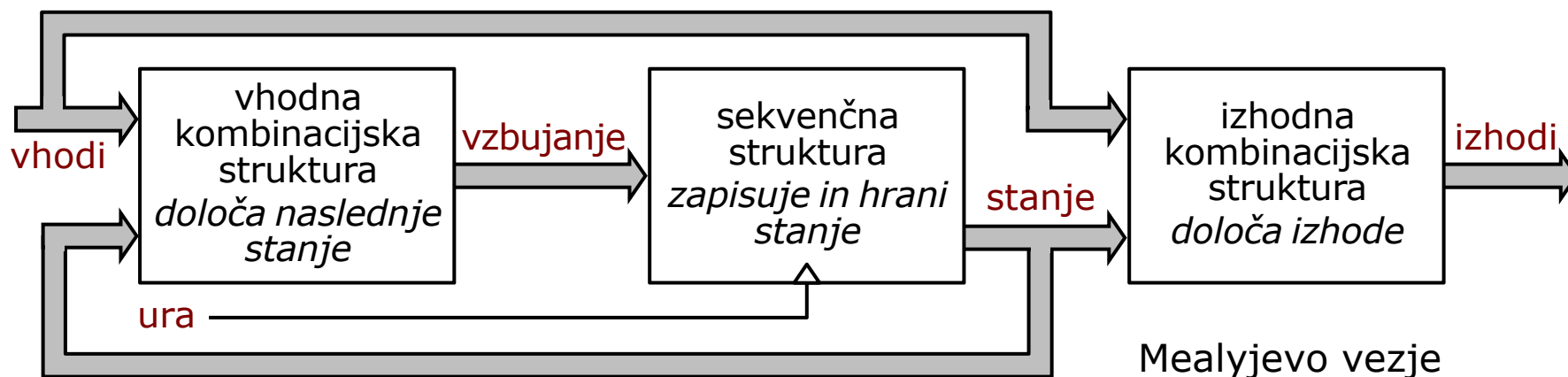
Mealyjeva in Mooreova vezja

- po principu delovanja delimo sekvenčna vezja z uro na:
 - **Mealyjeva vezja**, v katerih so tako naslednje stanje kot izhodi funkcije vhodov in trenutnega stanja,
 - **Mooreova vezja**, v katerih je naslednje stanje funkcija vhodov in trenutnega stanja, izhodi pa so funkcije zgolj trenutnega stanja
- tako Mealyjeva kot Mooreova vezja tvorijo trije deli:
 - **vhodna kombinacijska struktura**, v katero vstopajo vhodi in (prek povratne zveze) stanje vezja ter določa naslednje stanje
 - **sekvenčna struktura** iz flip-flopov, v kateri je zapisano trenutno stanje vezja, vsi flip-flopi pa so proženi z istim signalom (CLK),
 - **izhodna kombinacijska struktura**, v katero vstopajo vhodi in stanje (pri Mealyjevih vezjih) oziroma le stanje (pri Mooreovih) ter določa izhode



Sekvenčna vezja

Mealyjeva in Mooreova vezja





Sekvenčna vezja

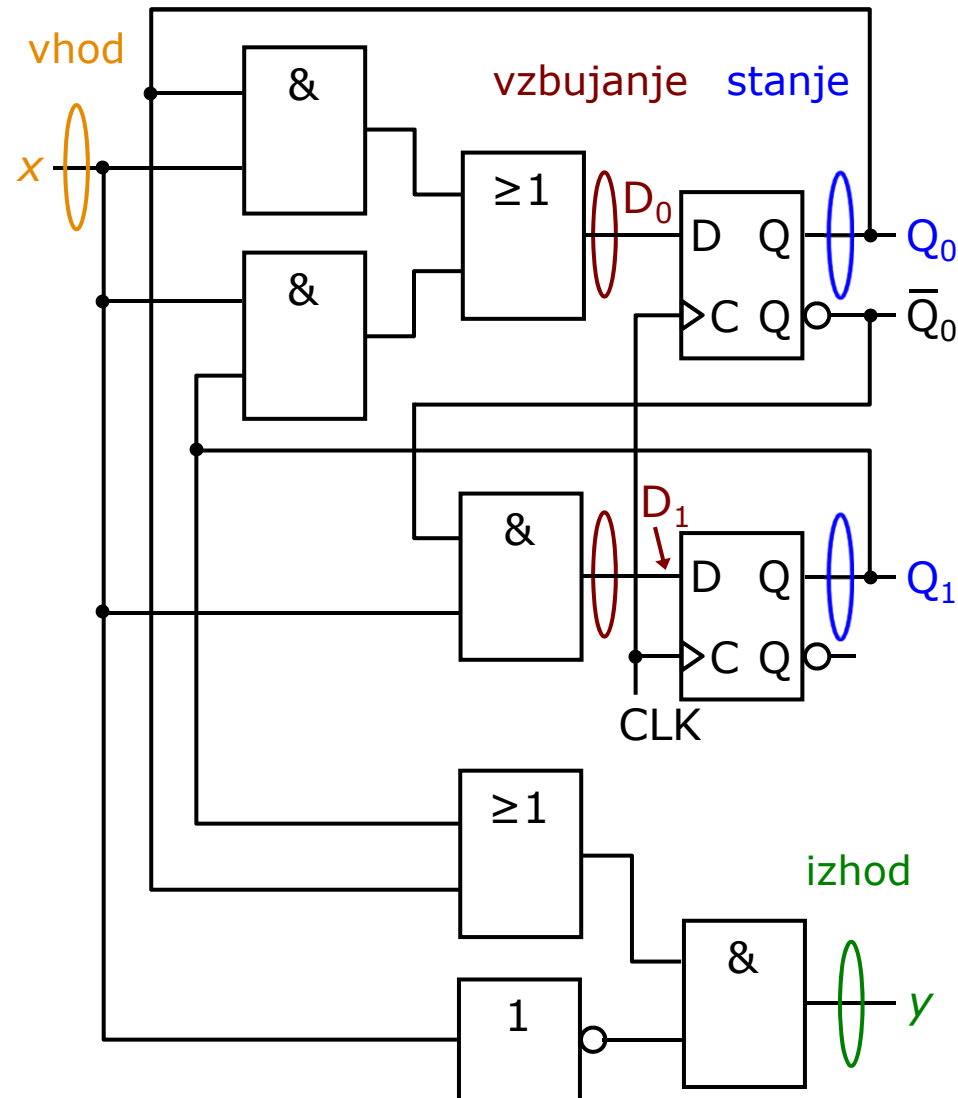
Mealyjeva in Mooreova vezja

- v sekvenčnih vezjih z več izhodi so lahko ti tudi različnih tipov: nekateri Mealyjevi, drugi Mooreovi
- v Mooreovih vezjih so izhodi sinhronski, t.j. spreminjajo se samo ob proženjih, ko se spreminjajo tudi stanja v sekvenčni strukturi vezja
- v Mealyjevih vezjih pa so izhodi asinhronski, saj sprememba vhodov vpliva na izhode tudi mimo sekvenčne strukture, zato se lahko izhodi spremenijo tudi med dvema proženjema (lahko tudi večkrat);
- ker v Mealyjevih vezjih vhodi vplivajo na izhode po dveh poteh z različnima zakasnitvama, lahko začasno pride tudi do nepravilne vrednosti izhodov, ki traja do trenutka, ko se sprememba vhodov odrazi tudi na izhodu sekvenčne strukture
- ti težavi Mealyjevih vezij odpravimo tako, da spreminjanje vhodov omejimo na le eno spremembo na en pulz ure, izhode pa vzorčimo le enkrat na pulz; oboje lahko izvedemo s spominskimi celicami

Sekvenčna vezja

Primer analize vezja z uro

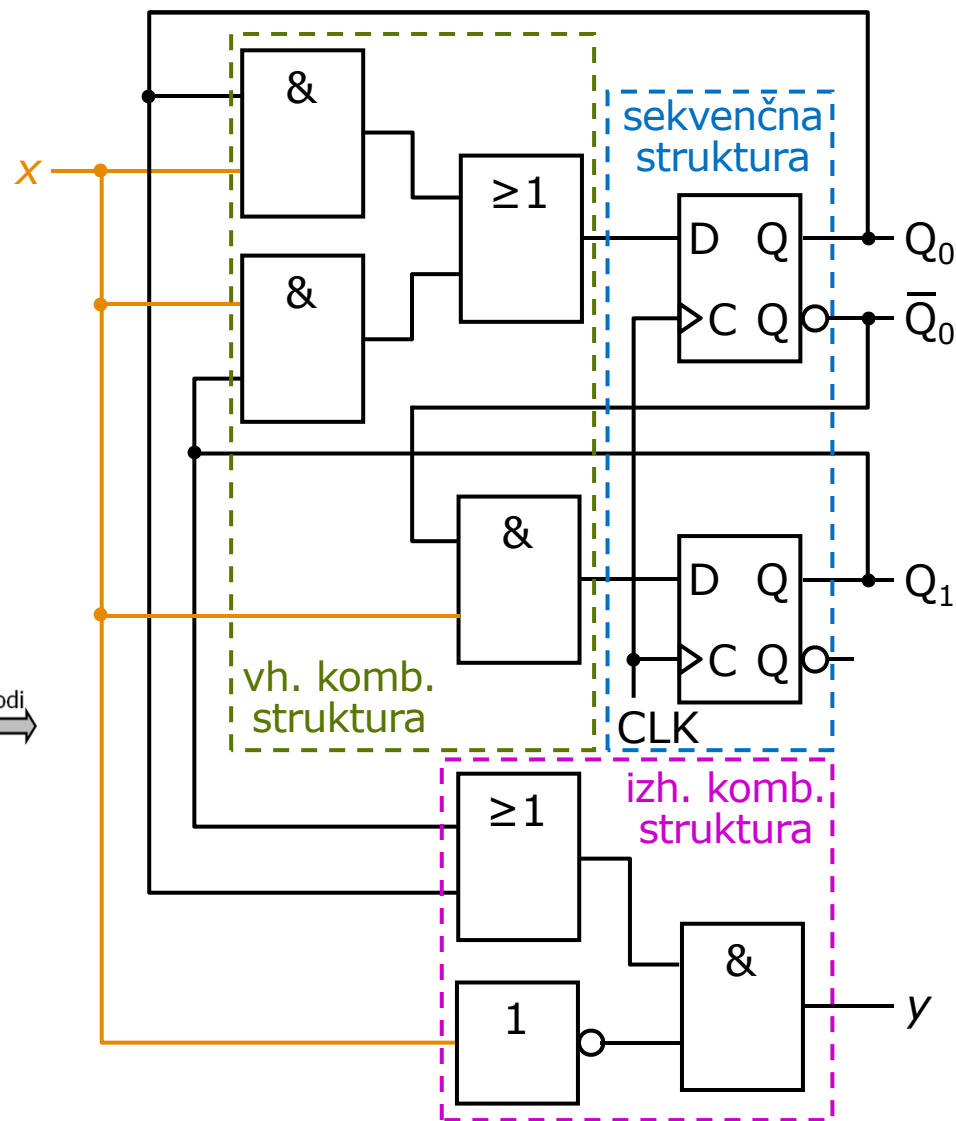
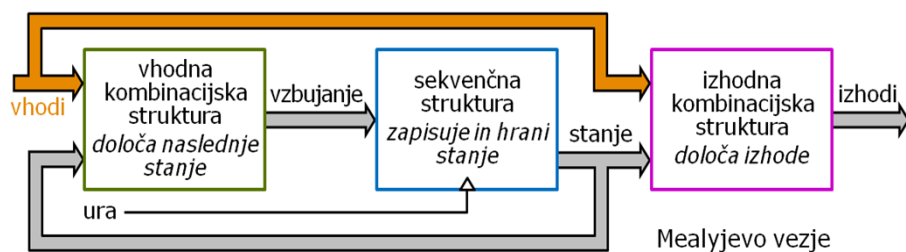
- oglejmo si postopek analize sekvenčnega vezja z uro na primeru vezja na desni
- vezje ima en vhod (x), dve spremenljivki stanja (Q_0 in Q_1 , t.j. 4 možna stanja: $Q_0Q_1 = 00, 01, 10, 11$) in en izhod (y)
- sekvenčno strukturo tvorita dva flip-flopa D s proženjem ob poz. fronti, strukturo pa ob vsakem proženju vzbuja skozi vhoda D_0 in D_1



Sekvenčna vezja

Primer analize vezja z uro

- vezje je Mealyjevo, saj vhod vpliva na izhodno kombina-
cijsko strukturo v vezju tudi
neposredno, ne le skozi
stanje sekvenčne strukture



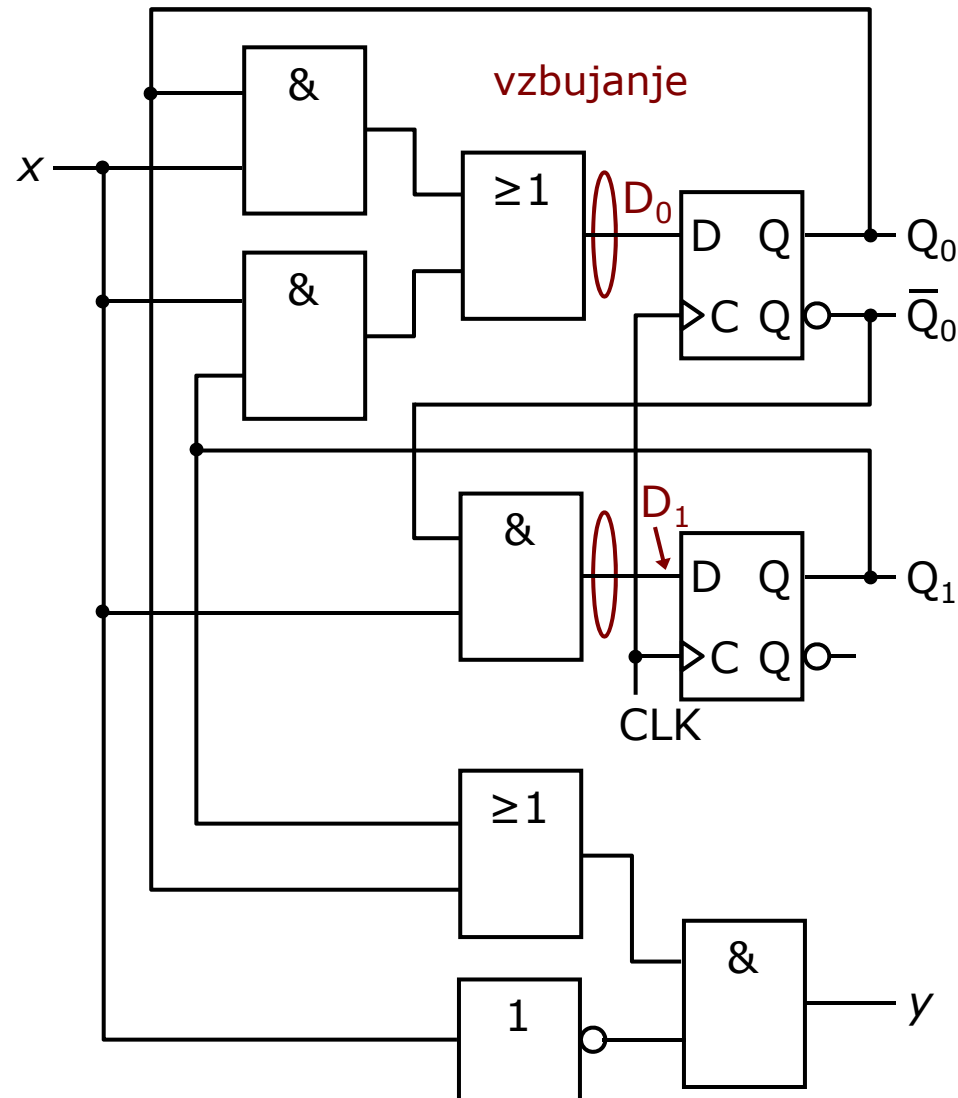
Sekvenčna vezja

Vzbujalne enačbe

- **vzbujalne enačbe** sekvenčnega vezja podajajo vhode v posamezne flip-flope kot funkcije trenutnih stanj teh flip-flopov in vhodov vezja:

$$D_0 = Q_0x + Q_1x = (Q_0+Q_1)x$$

$$D_1 = \bar{Q}_0 x$$





Sekvenčna vezja

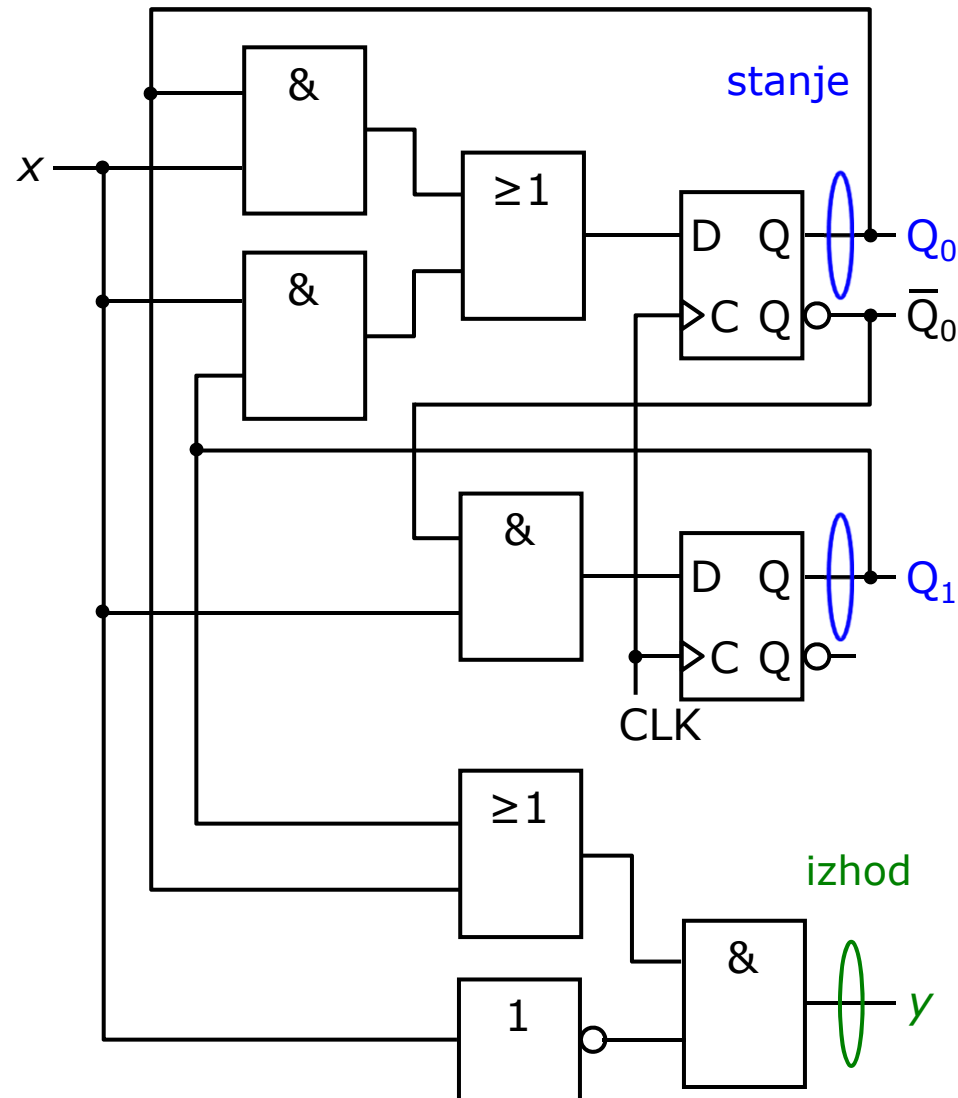
Enačbe stanj in izhodov

- **enačbe stanj** dobimo z vstavitvijo vzbujalnih enačb vezja v karakteristične enačbe ustreznih flip-flopov, **enačbe izhodov** pa razberemo iz zgradbe vezja:

$$Q_0^* = D_0 = (Q_0 + Q_1)x$$

$$Q_1^* = D_1 = \bar{Q}_0 x$$

$$y = (Q_0 + Q_1)\bar{x}$$





Sekvenčna vezja

Tabela stanj in izhodov

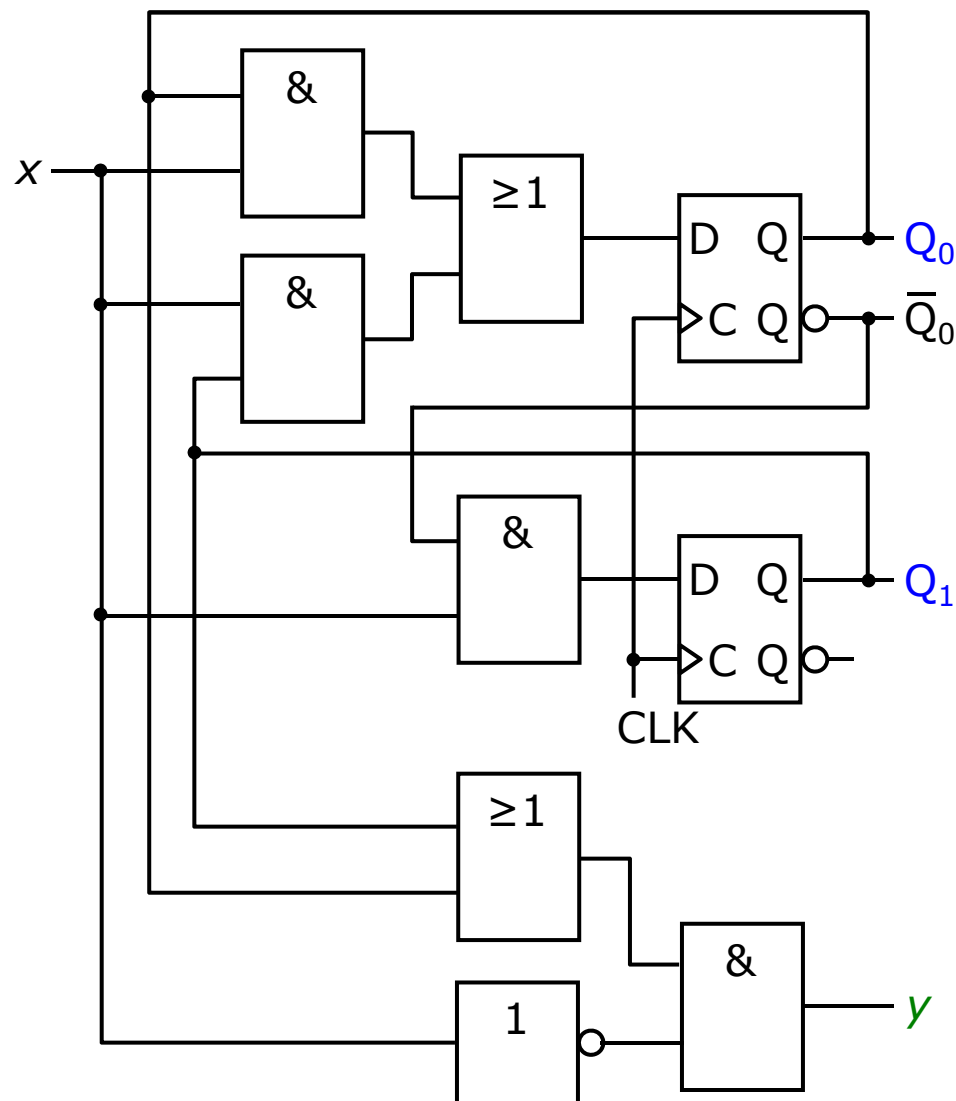
- tabelo stanj in izhodov** dobimo tako, da zapišemo vse kombinacije trenutnega stanja in vhodov, nato pa iz enačb stanj in izhodov določimo naslednje stanje in izhode za vsako kombinacijo:

$$Q_0^* = (Q_0 + Q_1)x$$

$$Q_1^* = \bar{Q}_0 x$$

$$y = (Q_0 + Q_1)\bar{x}$$

Q_0	Q_1	x	Q_0^*	Q_1^*	y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	1	0	0
1	1	0	0	0	1
1	1	1	1	0	0





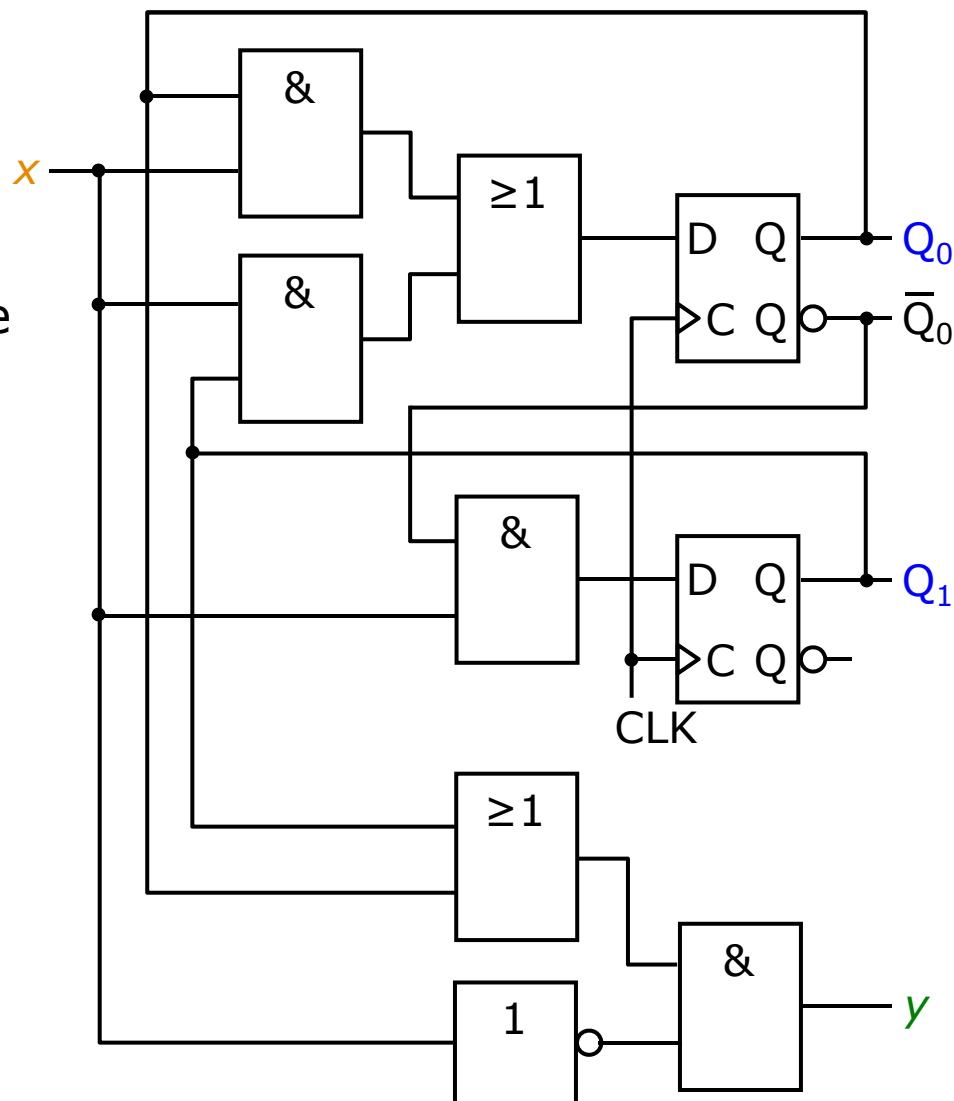
Sekvenčna vezja

Diagram stanj

- diagram stanj** dobimo iz tabele st. in izh.: za vsako stanje narišemo krog in vanj vpišemo vrednosti $Q_0 Q_1 \dots Q_n$, za vsak prehod pa puščico ter ob njej vhod, ki povzroči ta prehod, in izhod, ki nastopi ob prehodu

Q_0	Q_1	x	Q_0^*	Q_1^*	y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	1	0	0
1	1	0	0	0	1
1	1	1	1	0	0

00



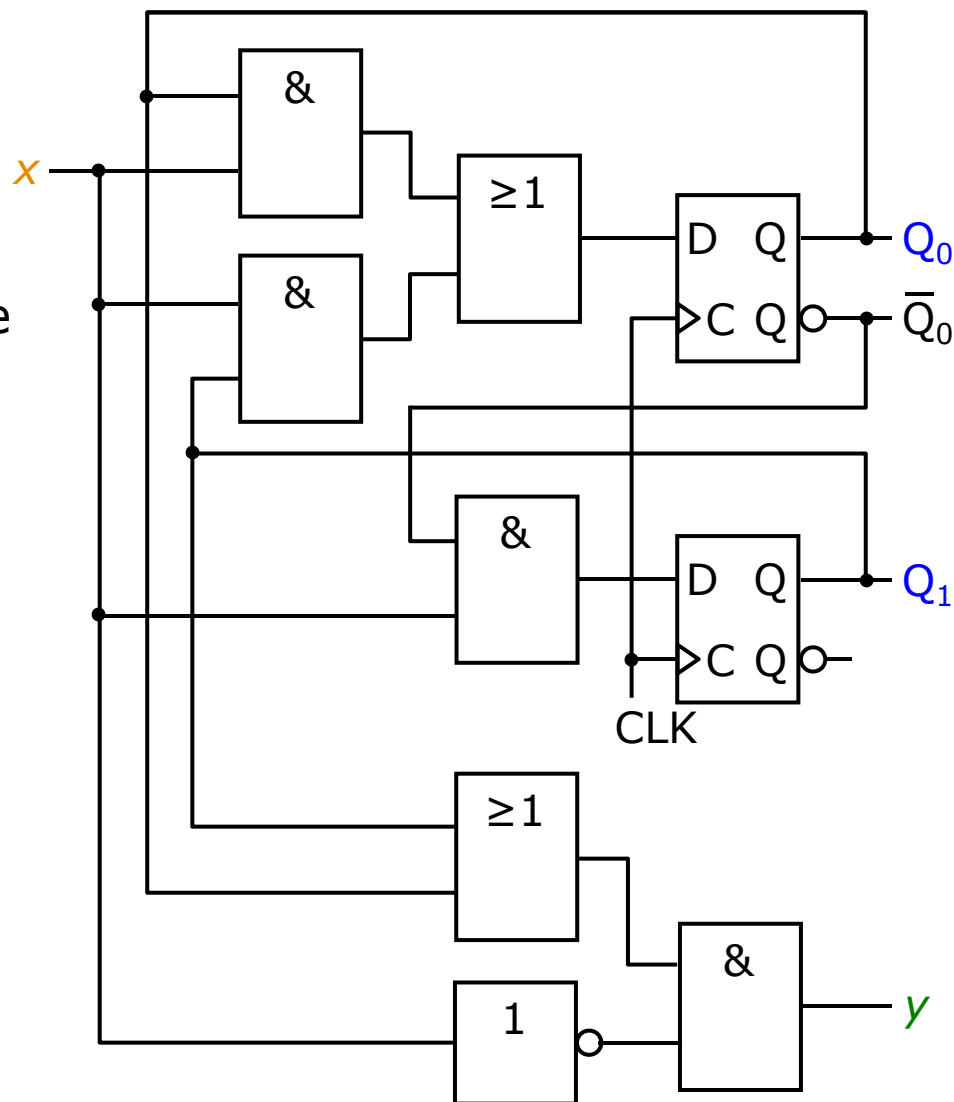
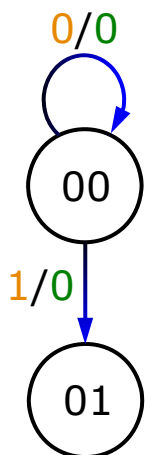


Sekvenčna vezja

Diagram stanj

- diagram stanj** dobimo iz tabele st. in izh.: za vsako stanje narišemo krog in vanj vpišemo vrednosti $Q_0 Q_1 \dots Q_n$, za vsak prehod pa puščico ter ob njej vhod, ki povzroči ta prehod, in izhod, ki nastopi ob prehodu

Q_0	Q_1	x	Q_0^*	Q_1^*	y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	1	0	0
1	1	0	0	0	1
1	1	1	1	0	0



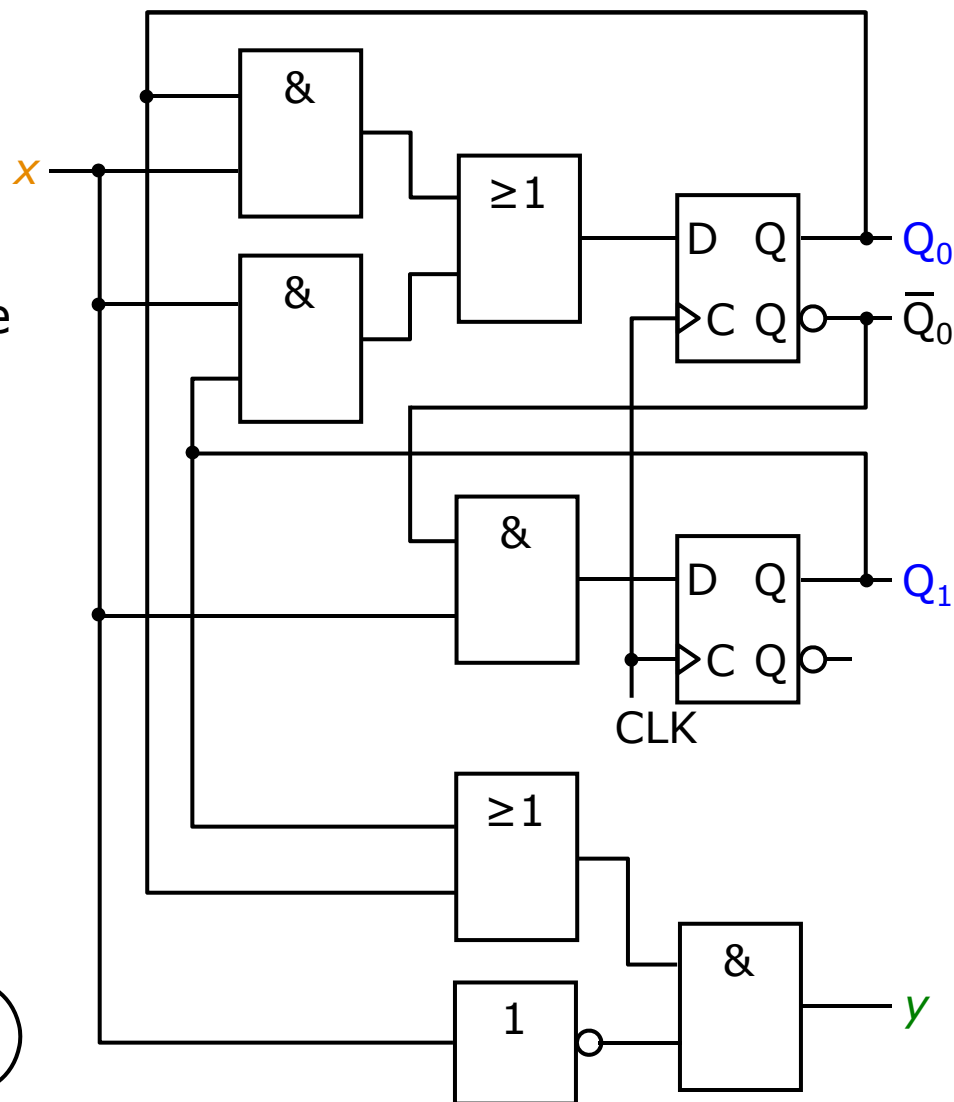
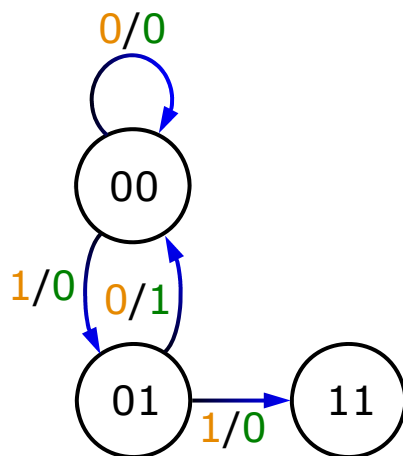


Sekvenčna vezja

Diagram stanj

- diagram stanj** dobimo iz tabele st. in izh.: za vsako stanje narišemo krog in vanj vpišemo vrednosti $Q_0 Q_1 \dots Q_n$, za vsak prehod pa puščico ter ob njej vhod, ki povzroči ta prehod, in izhod, ki nastopi ob prehodu

Q_0	Q_1	x	Q_0^*	Q_1^*	y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	1	0	0
1	1	0	0	0	1
1	1	1	1	0	0



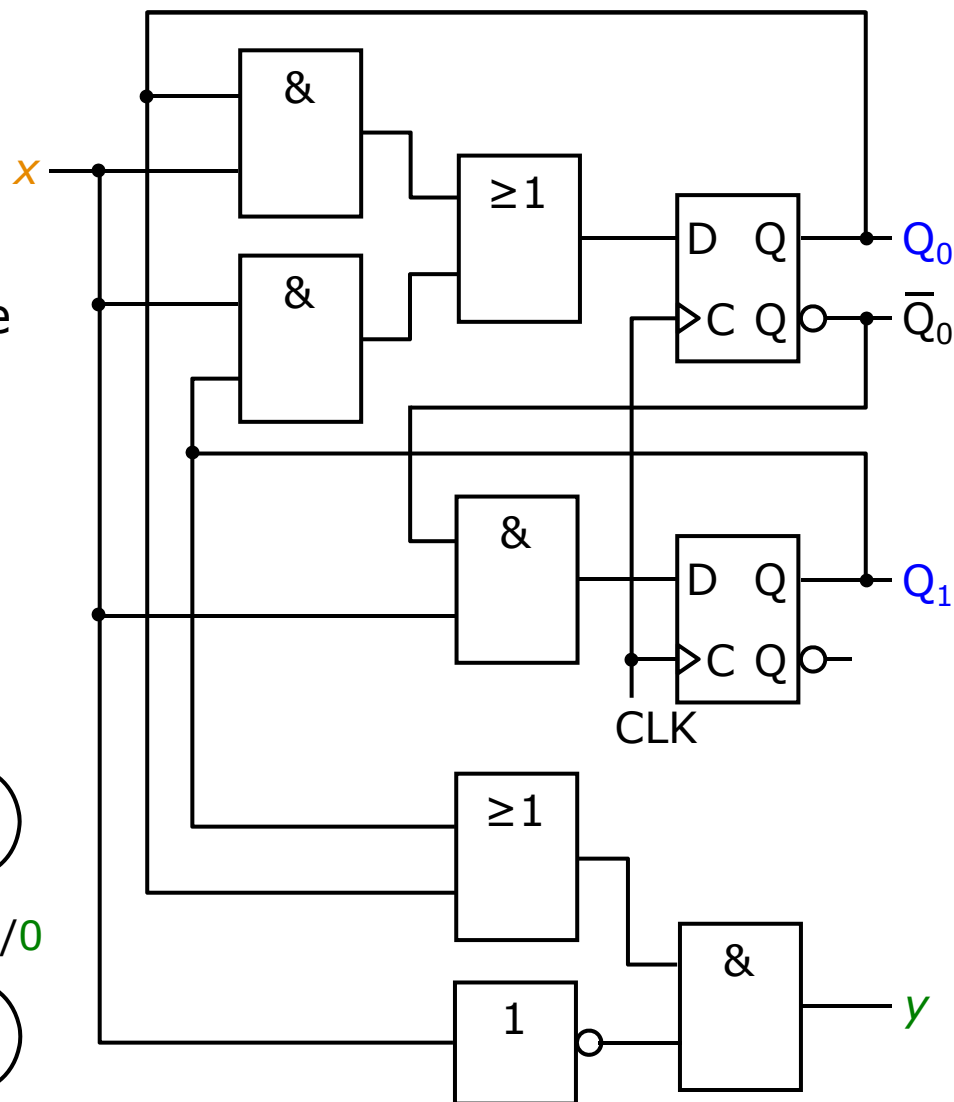
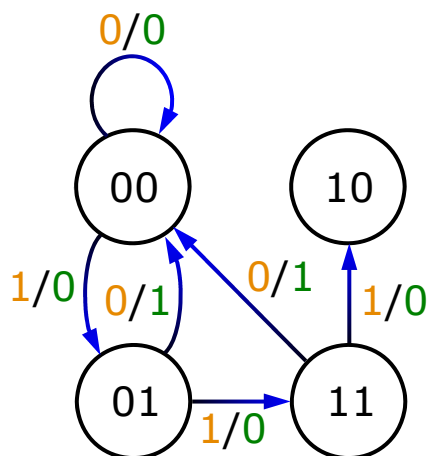


Sekvenčna vezja

Diagram stanj

- diagram stanj** dobimo iz tabele st. in izh.: za vsako stanje narišemo krog in vanj vpišemo vrednosti $Q_0 Q_1 \dots Q_n$, za vsak prehod pa puščico ter ob njej vhod, ki povzroči ta prehod, in izhod, ki nastopi ob prehodu

Q_0	Q_1	x	Q_0^*	Q_1^*	y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	1	0	0
1	1	0	0	0	1
1	1	1	1	0	0



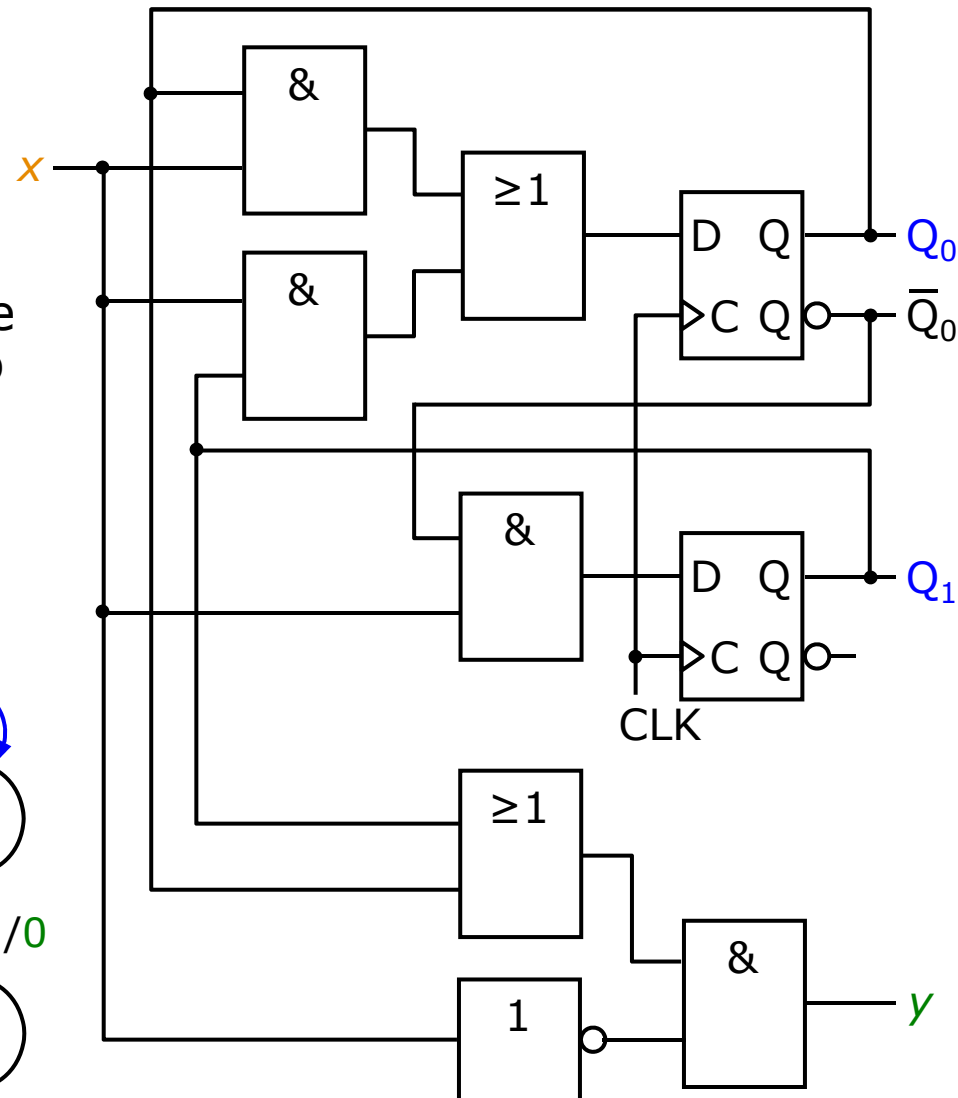
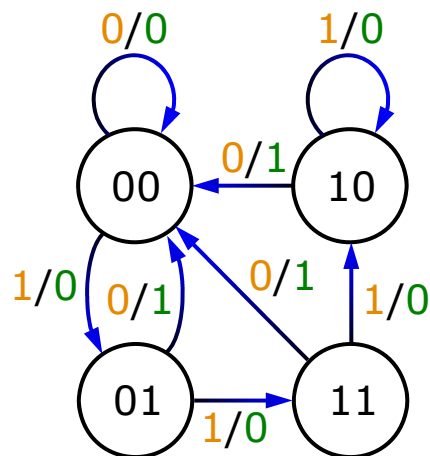


Sekvenčna vezja

Diagram stanj

- diagram stanj** dobimo iz tabele st. in izh.: za vsako stanje narišemo krog in vanj vpišemo vrednosti $Q_0 Q_1 \dots Q_n$, za vsak prehod pa puščico ter ob njej vhod, ki povzroči ta prehod, in izhod, ki nastopi ob prehodu

Q_0	Q_1	x	Q_0^*	Q_1^*	y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	1	0	0
1	1	0	0	0	1
1	1	1	1	0	0





Sekvenčna vezja

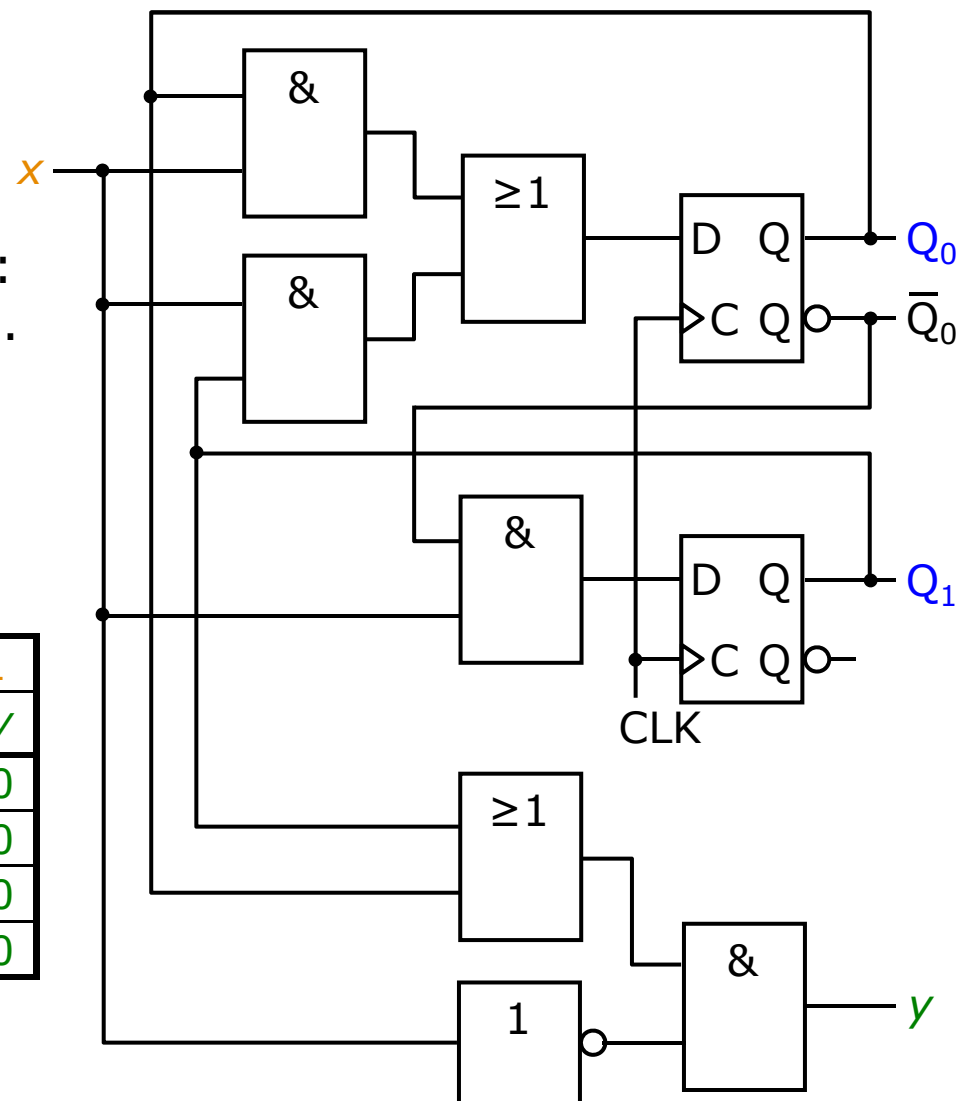
Poimenovanje stanj

- stanja lahko tudi poimenujemo:
 $Q_0Q_1 = 00 = S_0$, $Q_0Q_1 = 01 = S_1$, ...
ter temu prilagodimo tabelo stanj in izhodov ter diagram stanj

Q_0	Q_1	x	Q_0^*	Q_1^*	y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	1	0	0
1	1	0	0	0	1
1	1	1	1	0	0



S	x=0		x=1	
	S*	y	S*	y
S ₀	S ₀	0	S ₁	0
S ₁	S ₀	1	S ₃	0
S ₂	S ₀	1	S ₂	0
S ₃	S ₀	1	S ₂	0



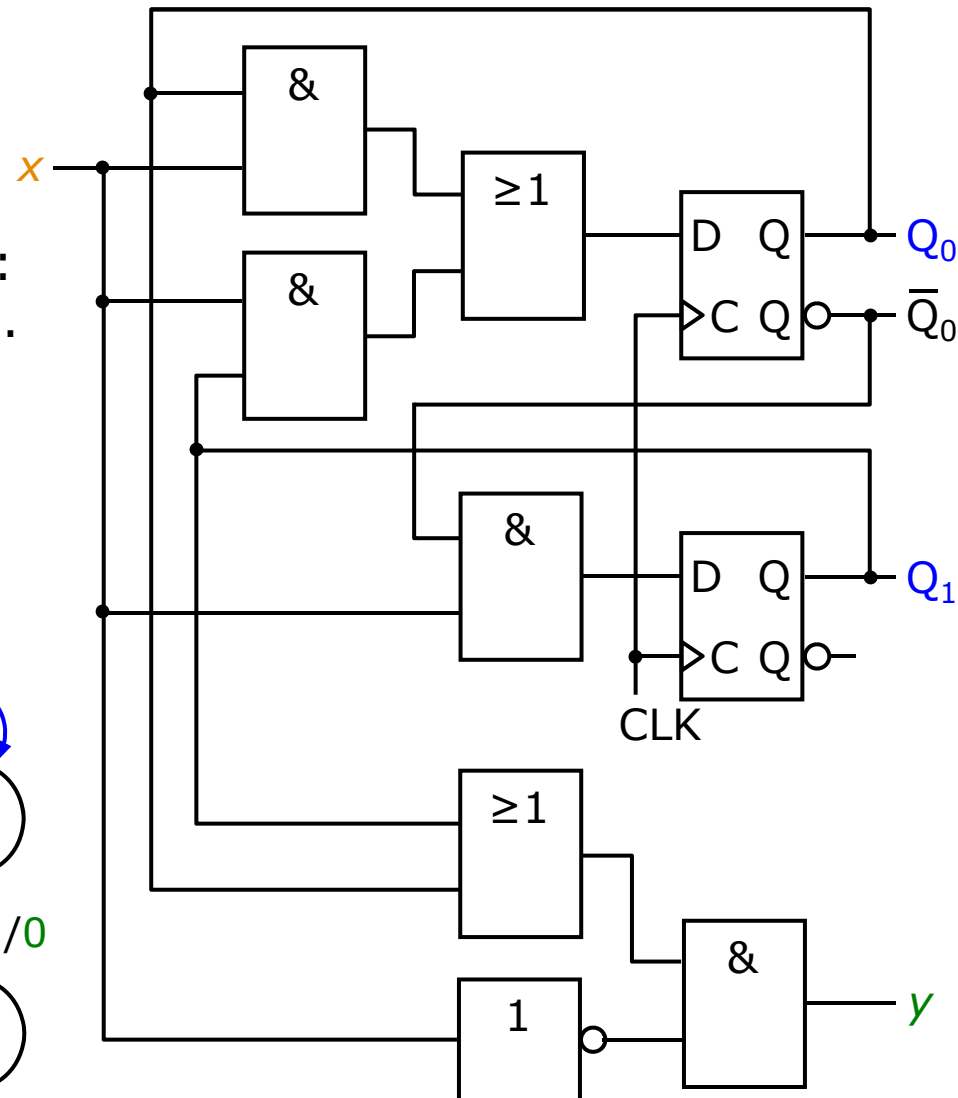
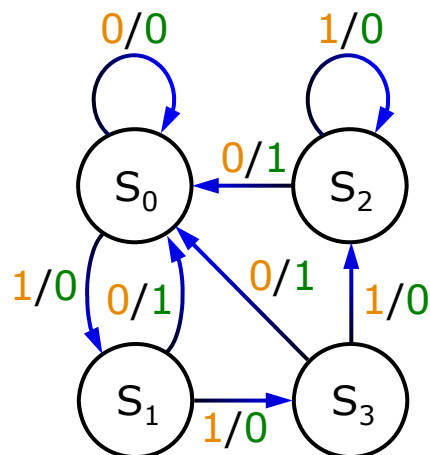


Sekvenčna vezja

Poimenovanje stanj

- stanja lahko tudi poimenujemo:
 $Q_0Q_1 = 00 = S_0, Q_0Q_1 = 01 = S_1, \dots$
ter temu prilagodimo tabelo stanj in izhodov ter diagram stanj

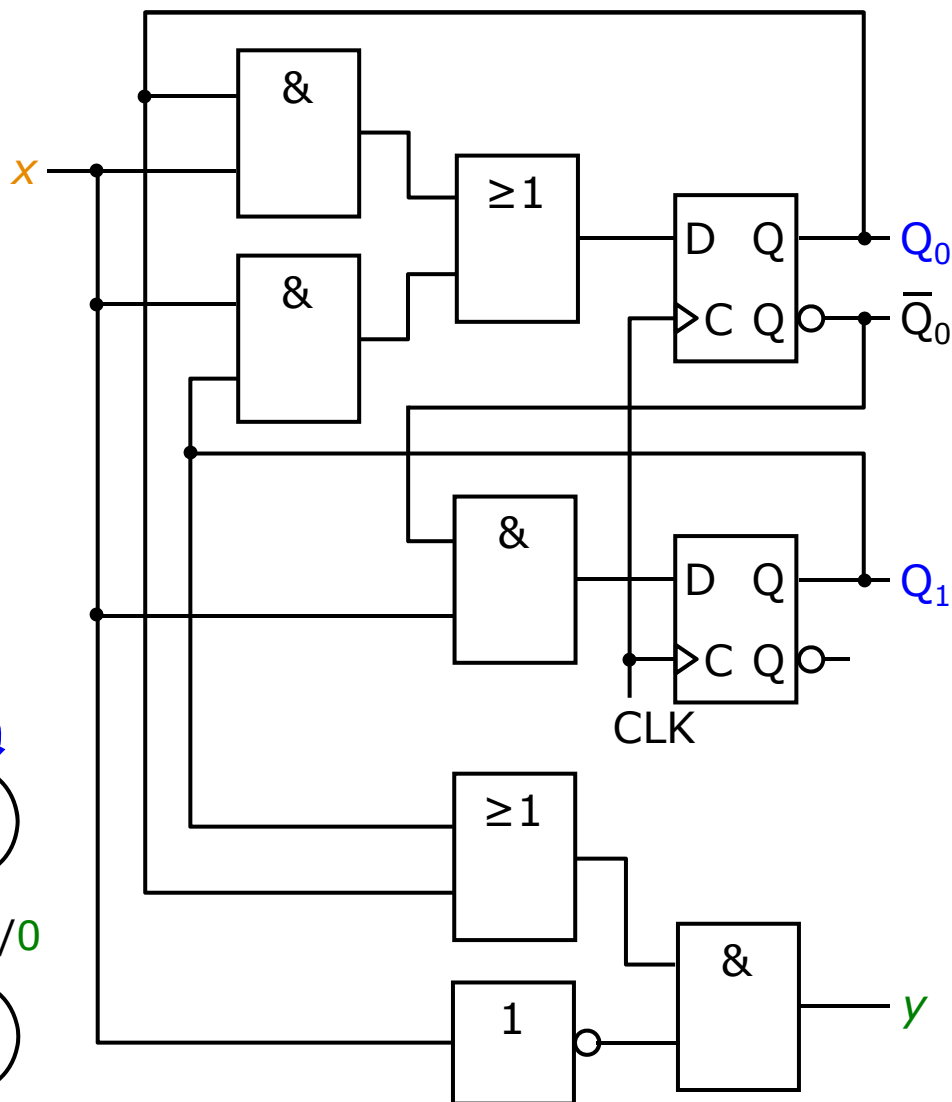
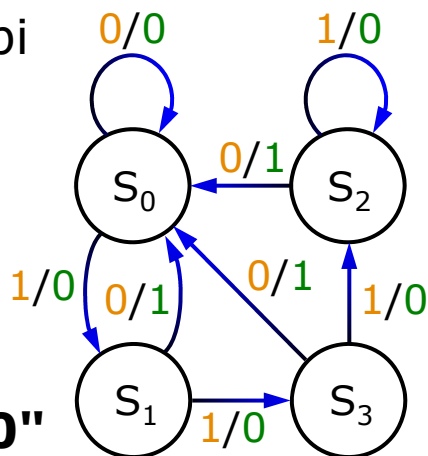
S	x=0		x=1	
	S^*	y	S^*	y
S_0	S_0	0	S_1	0
S_1	S_0	1	S_3	0
S_2	S_0	1	S_2	0
S_3	S_0	1	S_2	0



Sekvenčna vezja

Besedni opis delovanja vezja

- iz diagrama stanj sekvenčnega vezja je najlažje razumeti tudi delovanje vezja:
- če izhajamo iz stanja S_0 , ima izhod vrednost 0 tako dolgo, dokler ima vhod vrednost 1
- ob spremembi vhoda na 0 dobi izhod vrednost 1, vezje pa se vrne v S_0
- vezje je torej **detektor zaporedja "10"**



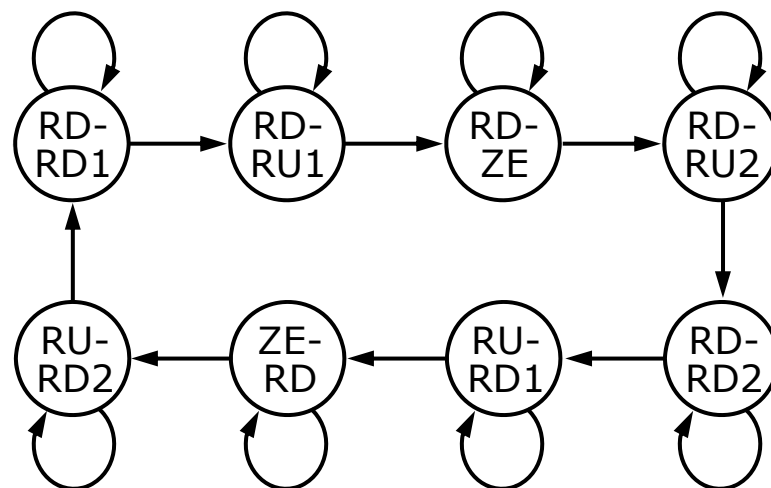
Sekvenčna vezja

Poimenovanje stanj in kodirna tabela

- za boljše razumevanje delovanja vezja so lahko poimenovanja stanj tudi opisna – npr. pri vezjih, ki upravljajo delovanje semaforjev na križišču: RD(eča), RU(mena), ZE(lena)
- če stanja poimenujemo opisno, diagram stanj več ne vsebuje podatkov o binarnih vrednostih spremenljivk stanja, zato moramo za popoln opis delovanja vezja priložiti še **kodirno tabelo**, ki podaja dogovor o **kodiranju stanj**:



S	Q ₀	Q ₁	Q ₂
RD-RD1	0	0	0
RD-RU1	0	0	1
RD-ZE	0	1	0
RD-RU2	0	1	1
RD-RD2	1	0	0
RU-RD1	1	0	1
ZE-RD	1	1	0
RU-RD2	1	1	1





Sekvenčna vezja

Diagrami stanj Mealyjevih in Mooreovih vezij

- tako v Mealyjevih kot v Mooreovih vezjih vhodi vplivajo na sekvenčno strukturo in s tem na stanje vezja; vsak prehod med stanjema je tako odvisen od vhodov, zato v diagramu stanj vrednosti vhodov pišemo ob ustrezni puščici
- v Mealyjevih vezjih so izhodi odvisni tako od stanja kot neposredno od vhodov, zato v diagramu stanj vrednosti izhodov pišemo ob puščici, ki podaja prehod iz ustreznega stanja ob ustrezni vrednosti vhodov
- v Mooreovih vezjih pa so izhodi odvisni samo od stanja, zato v diagramu stanj njihove vrednosti navajamo kar ob ustreznih stanjih

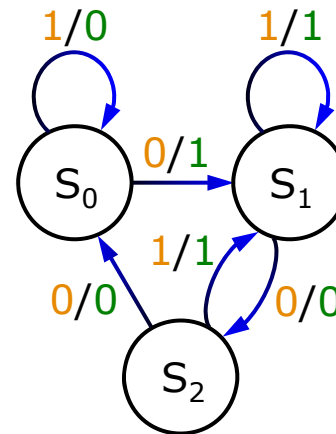


diagram stanj
Mealyjevega vezja

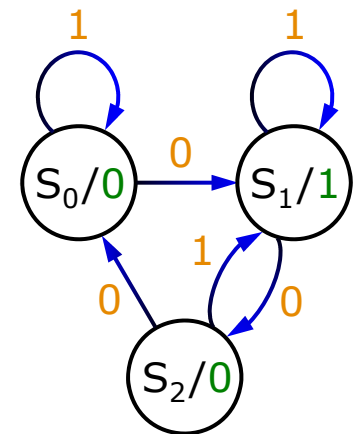


diagram stanj
Mooreovega vezja



Sekvenčna vezja

Analiza in sinteza sekvenčnih vezij

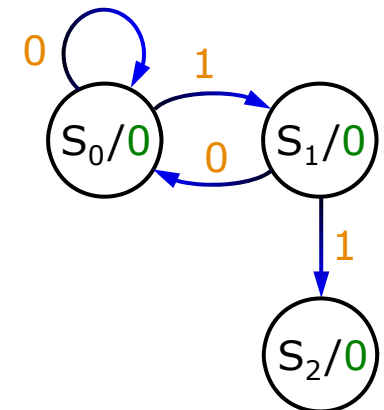
- postopek analize sekvenčnega vezja z uro, ki smo ga spoznali na primeru detektorja zaporedja "10", je sestavljen iz naslednjih korakov:
 - 1) določimo vzbujaalne enačbe flip-flopov
 - 2) določimo enačbe stanj in izhodov
 - 3) zapišemo tabelo stanj in izhodov (stanja lahko tudi poimenujemo)
 - 4) narišemo diagram stanj (pri opisnih imenih dodamo kodirno tabelo)
 - 5) podamo besedni opis delovanja vezja
- pri sintezi vezja sledimo tem korakom približno v nasprotnem redu:
 - 1) podamo besedni opis delovanja vezja
 - 2) narišemo diagram stanj (število stanj lahko nato še minimiziramo)
 - 3) zapišemo kodirno tabelo in tabelo stanj
 - 4) določimo enačbe stanj in izhodov ter jih minimiziramo
 - 5) izberemo vrsto flip-flopov in določimo vzbujaalne enačbe
 - 6) narišemo simbolno shemo vezja



Sekvenčna vezja

Primer sinteze vezja z uro

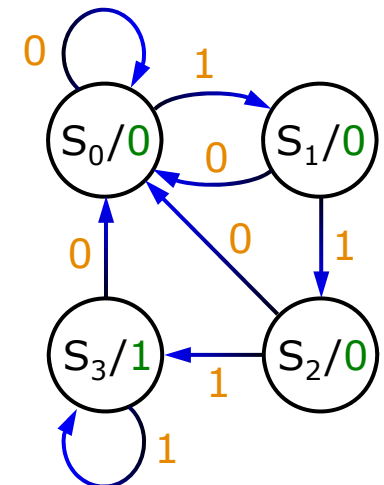
- tudi postopek sinteze sekvenčnega vezja z uro si oglejmo na primeru vezja, ki deluje kot detektor zaporedja
- besedni opis delovanja:** sekvenčno vezje naj bo Mooreovo, deluje pa naj kot detektor vhodnega zaporedja $x="111"$; ob takem zaporedju naj se na izhodu pojavi $y=1$, sicer pa naj bo na izhodu $y=0$
- diagram stanj:**
 - v izhodiščnem stanju S_0 bo $y=0$
 - pri $x=0$ bo vezje ostalo v S_0
 - pri $x=1$ bo prešlo v S_1 (detekcija "1"), kjer bo še vedno $y=0$
 - pri $x=0$ se bo vezje iz S_1 vrnilo v S_0 , pri $x=1$ pa bo prešlo v S_2 (detekcija "11"), še vedno z $y=0$



Sekvenčna vezja

Primer sinteze vezja z uro

- tudi postopek sinteze sekvenčnega vezja z uro si oglejmo na primeru vezja, ki deluje kot detektor zaporedja
- besedni opis delovanja:** sekvenčno vezje naj bo Mooreovo, deluje pa naj kot detektor vhodnega zaporedja $x="111"$; ob takem zaporedju naj se na izhodu pojavi $y=1$, sicer pa naj bo na izhodu $y=0$
- diagram stanj:**
... (nadaljevanje)
 - pri $x=0$ se bo vezje iz S_2 vrnilo v S_0 ,
pri $x=1$ pa prešlo v S_3 (detekcija "111"),
kjer bo $y=1$, saj je prišlo do detekcije "111"
 - pri $x=0$ se bo vezje tudi iz S_3 vrnilo v S_0 ,
pri $x=1$ pa bo ostalo v S_3 , saj tudi zaporedja štirih in več enic vsebujejo "111"

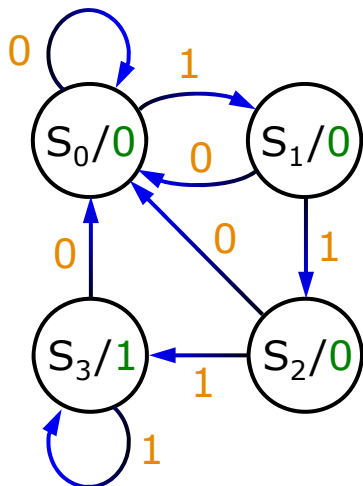




Sekvenčna vezja

Primer sinteze vezja z uro

- za štiri različna stanja, S_0 - S_3 , bomo potrebovali dve spremenljivki stanja; kot običajno ju poimenujmo Q_0 in Q_1 , izberimo pa tudi običajni način kodiranja, ki ga podaja spodnja **kodirna tabela**
- **tabelo stanj in izhodov** dobimo iz diagrama stanj: iz vsakega od osmih prehodov med stanji razberemo ustrezne vrednosti Q_0^* , Q_1^* in y



S	Q_0	Q_1
S_0	0	0
S_1	0	1
S_2	1	0
S_3	1	1

Q_0	Q_1	x	Q_0^*	Q_1^*	y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	1	1	0
1	1	0	0	0	1
1	1	1	1	1	1



Sekvenčna vezja

Primer sinteze vezja z uro

- **enačbe stanj in izhodov** zapišemo iz tabele stanj in izhodov kot vsote mintermov, ki jih nato minimiziramo (pri obsežnejših izrazih si pomagamo s K-diagramom in/ali drugimi postopki poenostavljanja)
- za Q_0 in Q_1 potrebujemo dve spominski celici; če izberemo flip-flopa D, velja $Q^* = D$, zato sta **vzbujalni enačbi** kar enaki enačbama stanj

Q_0	Q_1	x	Q_0^*	Q_1^*	y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	1	1	0
1	1	0	0	0	1
1	1	1	1	1	1

$$Q_0^* = \bar{Q}_0 Q_1 x + Q_0 \bar{Q}_1 x + Q_0 Q_1 x = (Q_0 + Q_1) x$$

$$Q_1^* = \bar{Q}_0 \bar{Q}_1 x + Q_0 \bar{Q}_1 x + Q_0 Q_1 x = (Q_0 + \bar{Q}_1) x$$

$$y = Q_0 Q_1 \bar{x} + Q_0 Q_1 x = Q_0 Q_1$$

$$D_0 = (Q_0 + Q_1) x$$

$$D_1 = (Q_0 + \bar{Q}_1) x$$



Sekvenčna vezja

Primer sinteze vezja z uro

- iz vzbujalnih enačb obeh flip-flopov in enačbe izhoda sledi še zgradba vezja, ki jo podaja **simbolna shema** na desni

$$y = Q_0 Q_1 \bar{x} + Q_0 Q_1 x = Q_0 Q_1$$

$$D_0 = (Q_0 + Q_1)x$$

$$D_1 = (Q_0 + \bar{Q}_1)x$$

