VTŠ: Osnovi računarske tehnike

Sekvencijalna logika

mr. Veličković Zoran

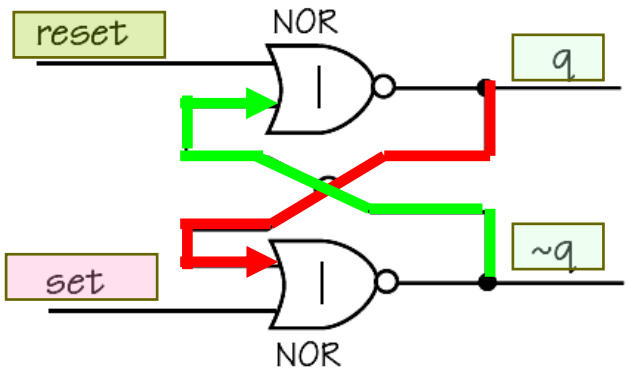
Mart, 2010.

Sekvencijalna logička funkcija

- Logičke funkcije se mogu kategorizirati kao **kombinacione** ili kao **sekvencijalne**.
- U slučaju kombinacionih logičkih funkcija, njihov logički izlaz zavisi samo od **trenutne** kombinacije prisutnih logičkih ulaza (Do sada smo razmatrali samo ove tipove funkcija).
- Kod sekvencijalnih funkcija stanje izlaza takođe zavisi od stanja logičkih ulaza ali i od njihove logičke vrednosti u **prethodnom** stanju.
- Dakle, izlazi kod sekvencijalnih funkcija zavise od vrednosti kompletne ulazne sekvence.
- Sekvencijalne funkcije pamte prethodno stanje ulaza i zato ih nazivamo **memorijskim elementima**.

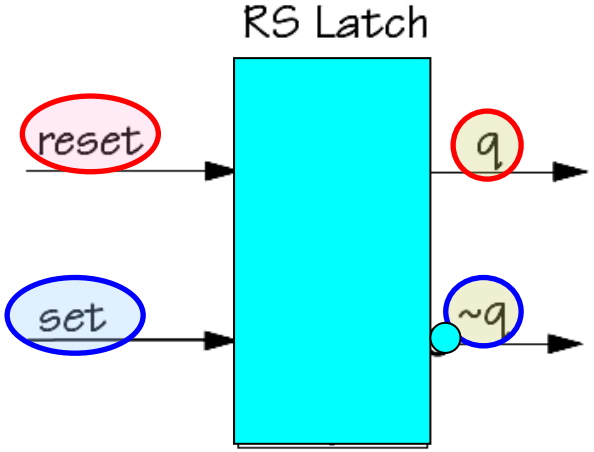
RS latc (leč)

- Jedna od najednostavnijih sekvencijalnih funkcija je **RS leč**, koji se može formirati korišćenjem **dva NILI kola** povezana na specifičan način (**back-to-back**).



- U ovoj NILI (NOR) verziji **RESET** i **SET** ulazi su aktivni visoko.
- Imena ovih ulaza odgovaraju **stanju izlaza Q** na koji deluju.
- Kada je na **RESET** na aktivnom nivou izlaz **q** se postavlja na **0**.
- Kada je **SET** na aktivnom nivou izlaz **q** se postavlja u **1**.
- Izlazi **q** i **~q** se nazivaju **pravi** (*true*) i **komplementarni** izlaz (*commplementary*)

Predstavljanje RS leča



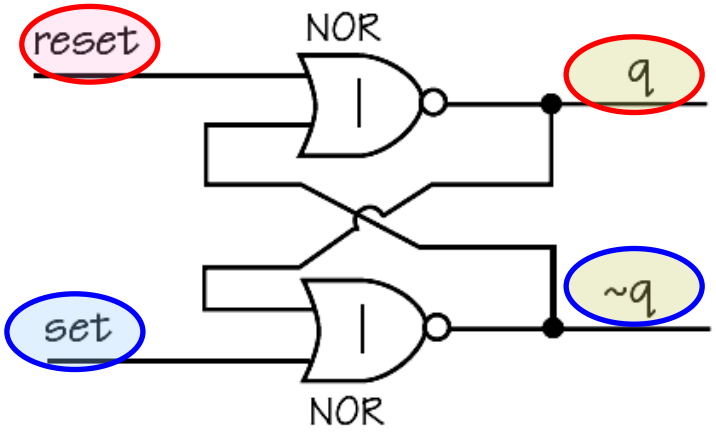


Tabela istinitosti

reset	set	$q_{(n+)}$	$\sim q_{(n+)}$
0	0	$q_{(n)}$	$\sim q_{(n)}$
0	1	1	0
1	0	0	1
1	1	0*	0*

(0* = Unstable State)

$q = \overline{(reset \mid \sim q)}$

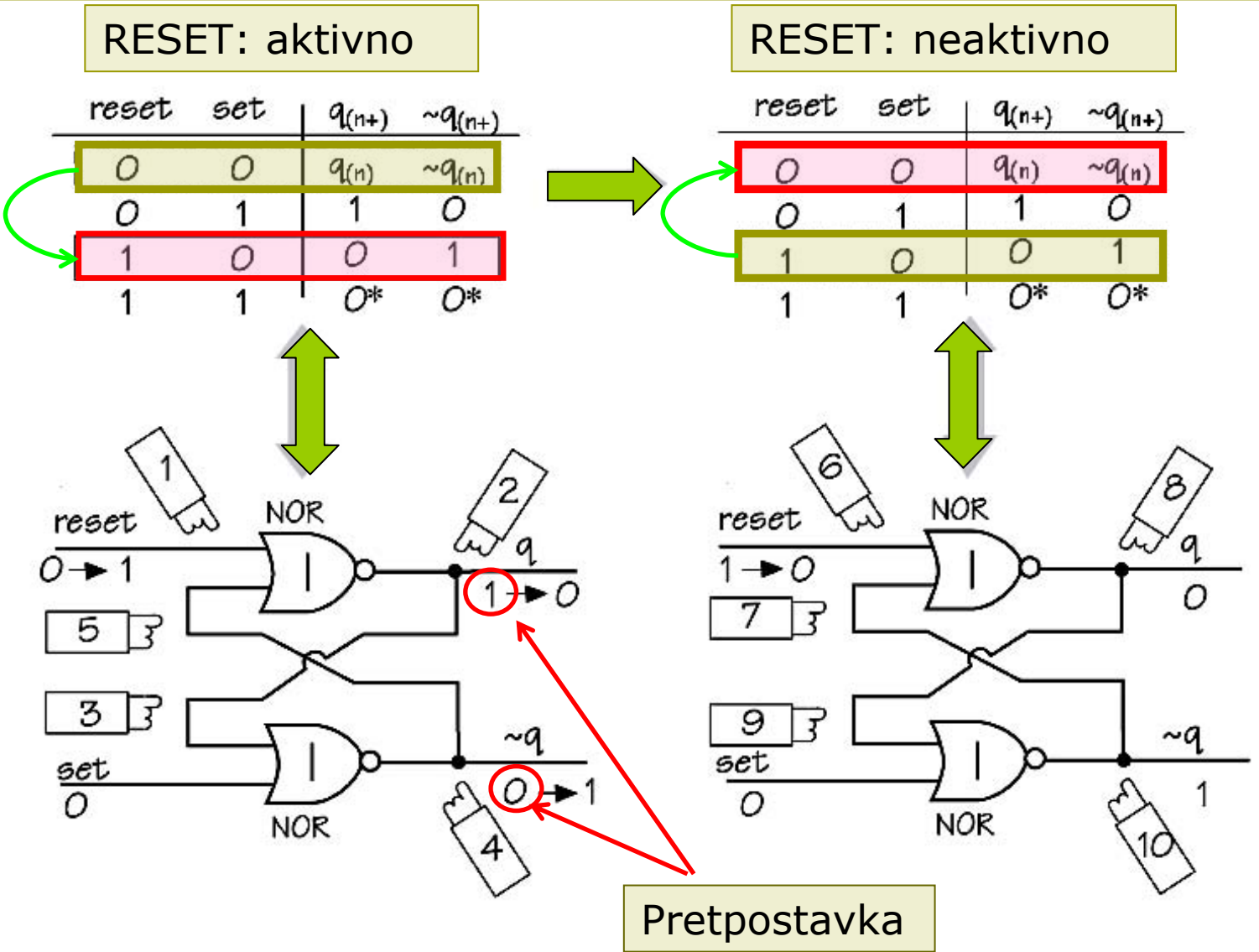
$\sim q = \overline{(set \mid q)}$

Zadržava se prethodno stanje izlaza

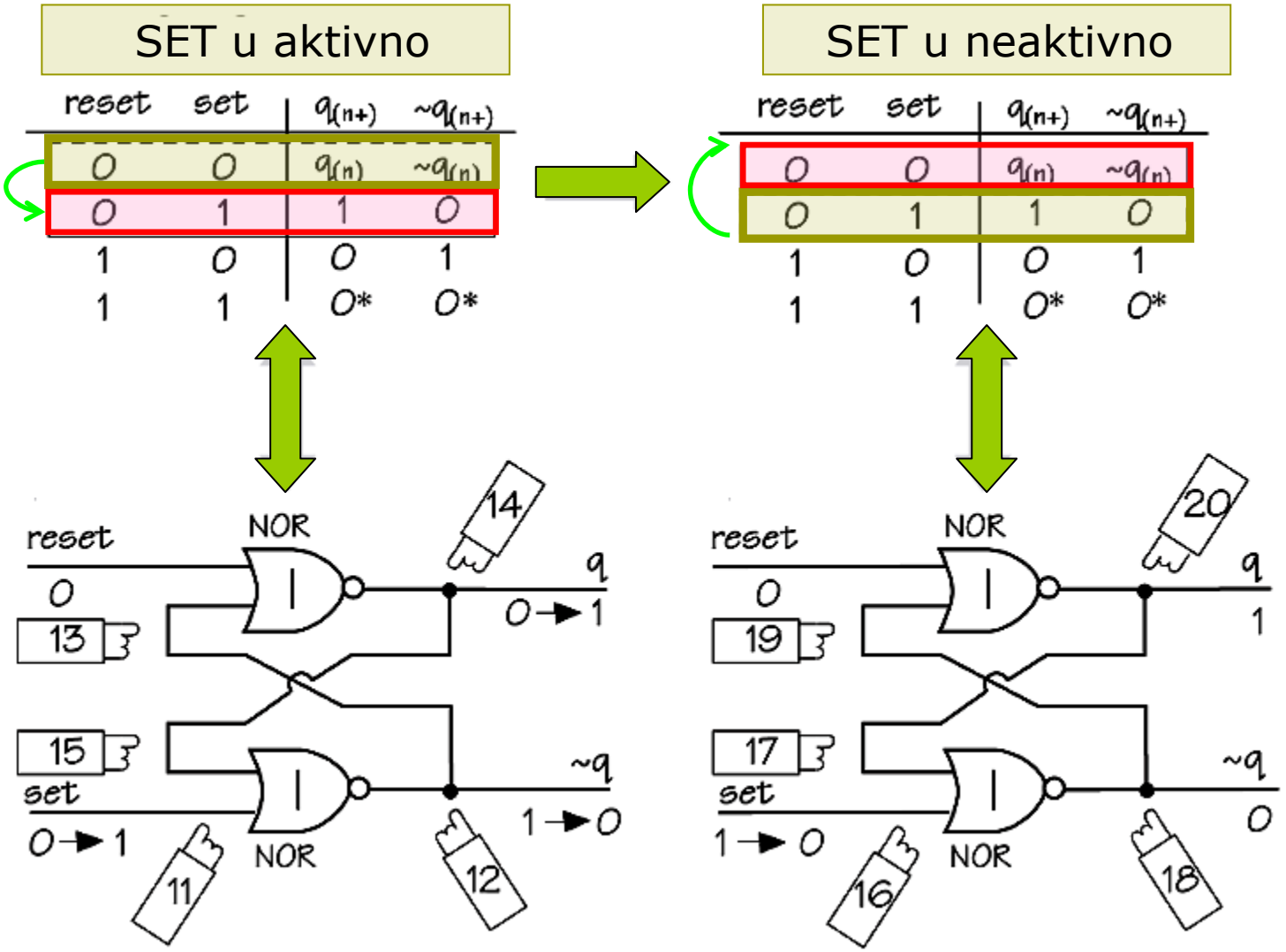
RS leč i feedback

- ~q **nije inverzno** od q samo u slučaju kada su ulazi **SET** i **RESET** istovremeno postavljeni u aktivno stanje (SET=RESET=1)**!**
- Stanja **q(n+)** i **~q(n+)** označavaju **buduće vrednosti** ovih izlaza.
- RS latch **pamti prethodno stanje ulaza** što se bazira na tehnici pod nazivom ***feedback***.
- Ovo je zapravo **vraćanje stanja izlaza** na ulaz (čime se formira **dodatni ulaz** u funkciju).
- Na sledećim slikama ćemo objasniti kako ova tehnika funkcioniše.
- Počnimo sa pretpostavkom da su ulazi (SET=RESET=0) u **neaktivnom stanju**.

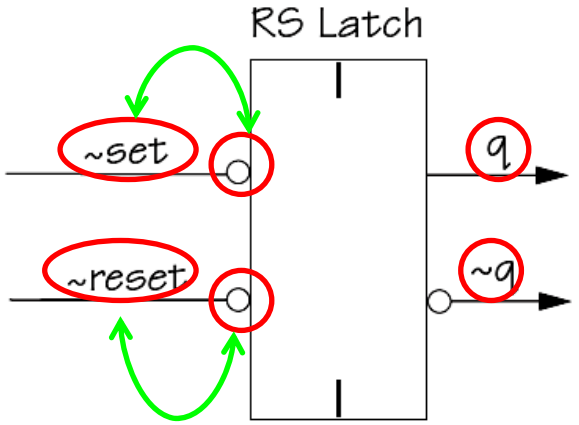
RESET aktivno/neaktivno

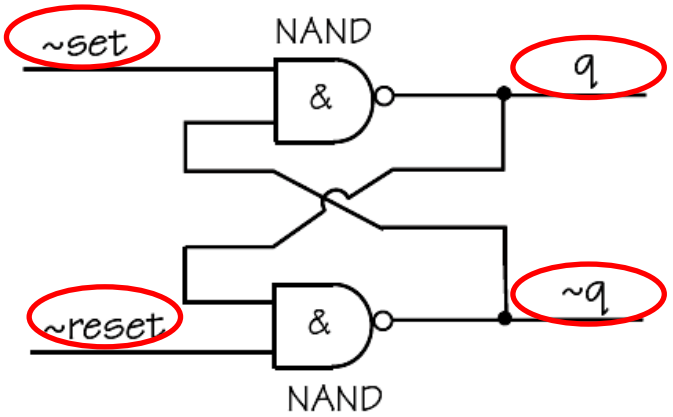


SET aktivno/neaktivno



NI verzija RS leča





~ reset	~ set	$q_{(n+)}$	$\sim q_{(n+)}$
0	0	1*	1*
0	1	0	1
1	0	1	0
1	1	$q_{(n)}$	$\sim q_{(n)}$

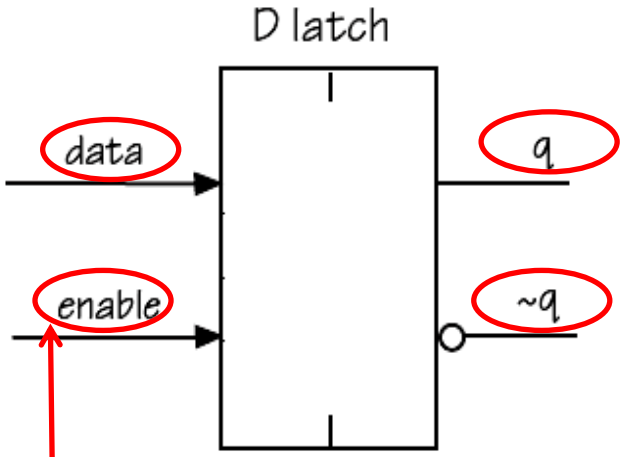
(1* = Unstable State)

$q = \overline{(\sim set \ \& \ \sim q)}$

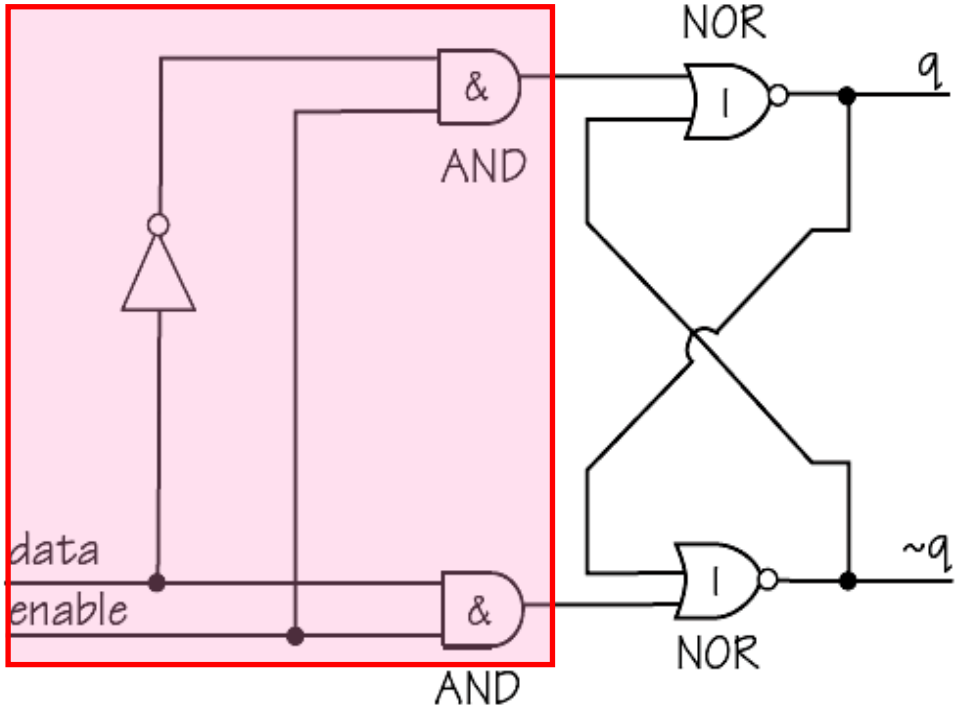
$\sim q = \overline{(\sim reset \ \& \ q)}$

SET i REST aktivno-nisko

Data tip leča: D-latch



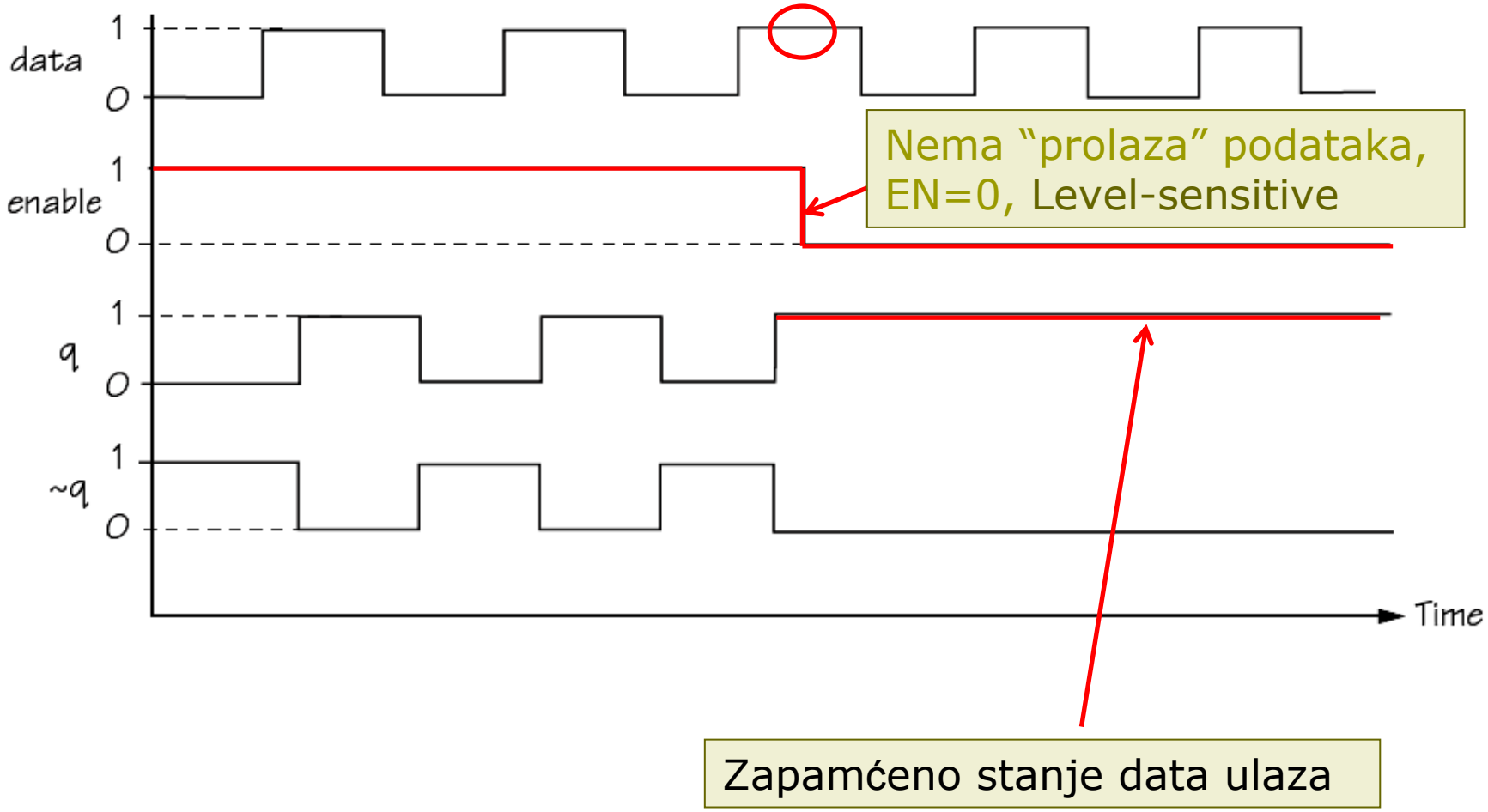
enable	data	$q_{(n+)}$	$\sim q_{(n+)}$
0	?	$q_{(n)}$	$\sim q_{(n)}$
1	0	0	1
1	1	1	0



Aktivno visoko

Leč D tipa sa aktivno-visoko ulazima

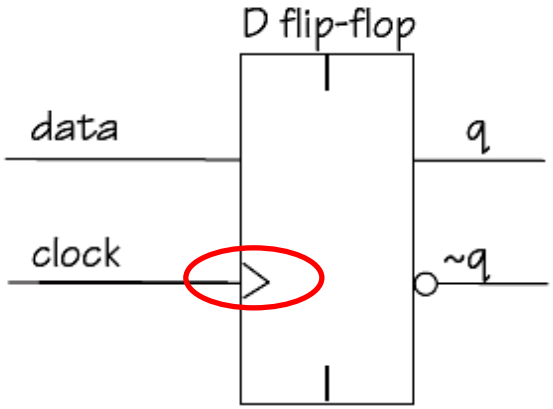
Primena D leča

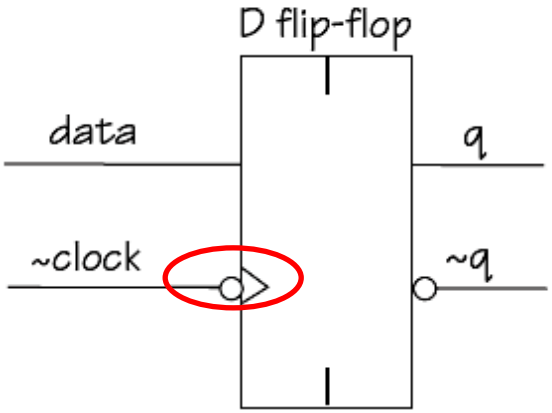


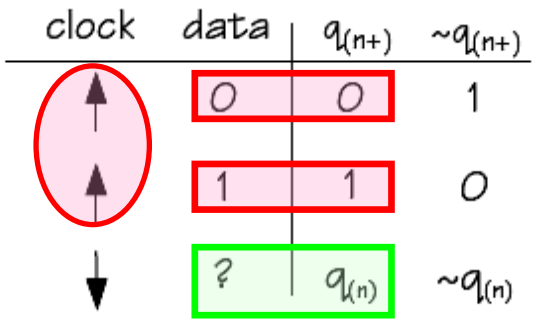
D-latch → D Flip-Flop

- Za razliku od **level-sensitive** logičkih šema, **transition-sensitive** šeme pamte podatake u trenutku promene logičkog nivoa **clock** ulaza.
- Oznaka transition-sensitive ulaza na šemama je sledeća **“>”.**
- Prelazak sa logičkog nivoa **0** na **1** naziva se **rastuća ivica** – pozitivna (*rising-edge), dok se* tranzicija sa 1 na 0 naziva **opadajuća ivica** – negativna (*falling-edge).*
- D-tip flip-flopovi se mogu realizovati i sa **pozitivnim** i **negativnim** *clock* ulazom.

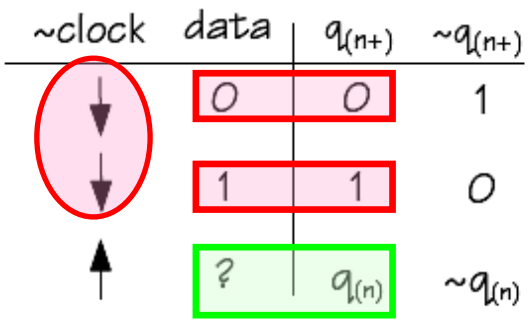
Pozitivno i negativno trigerovani D-FF





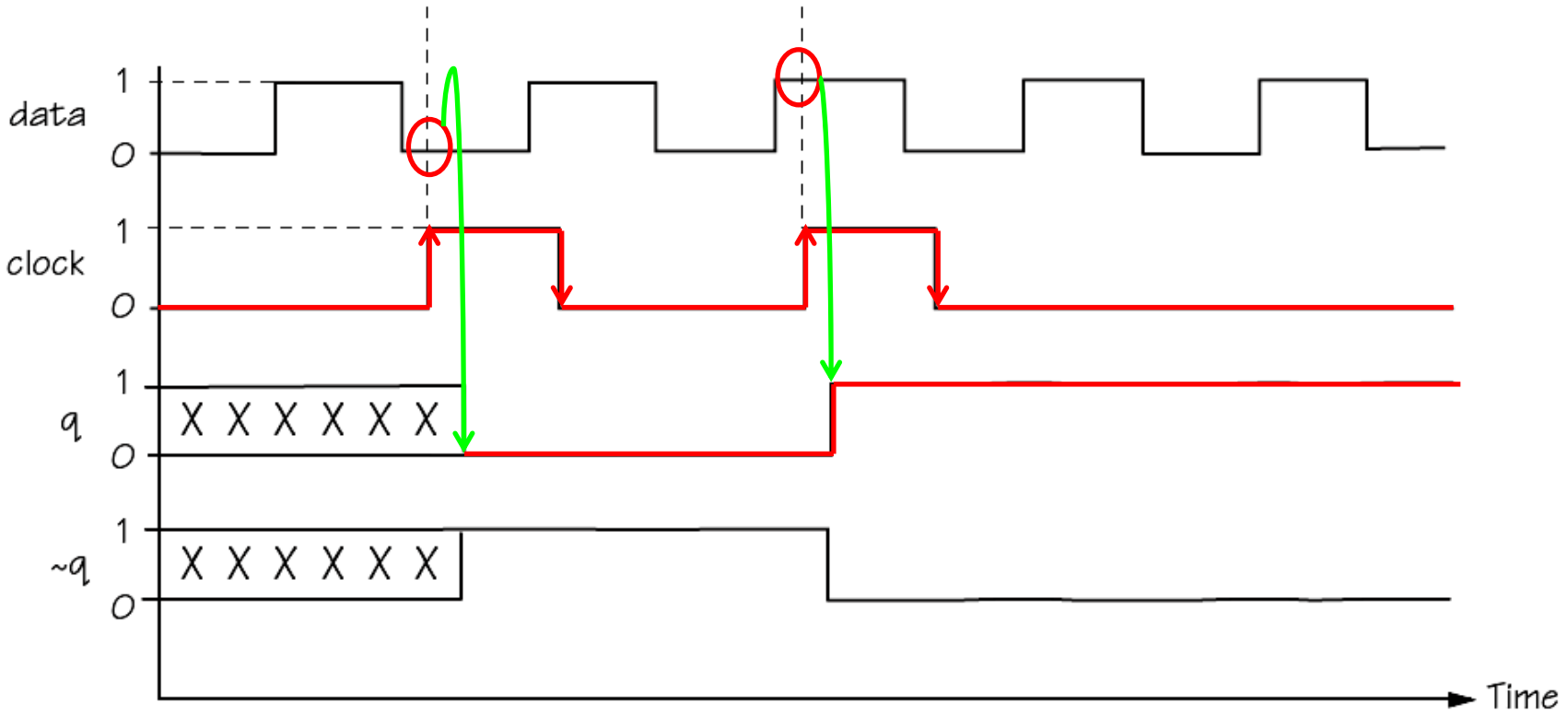


Pozitivno trigerovan



Negativno trigerovan

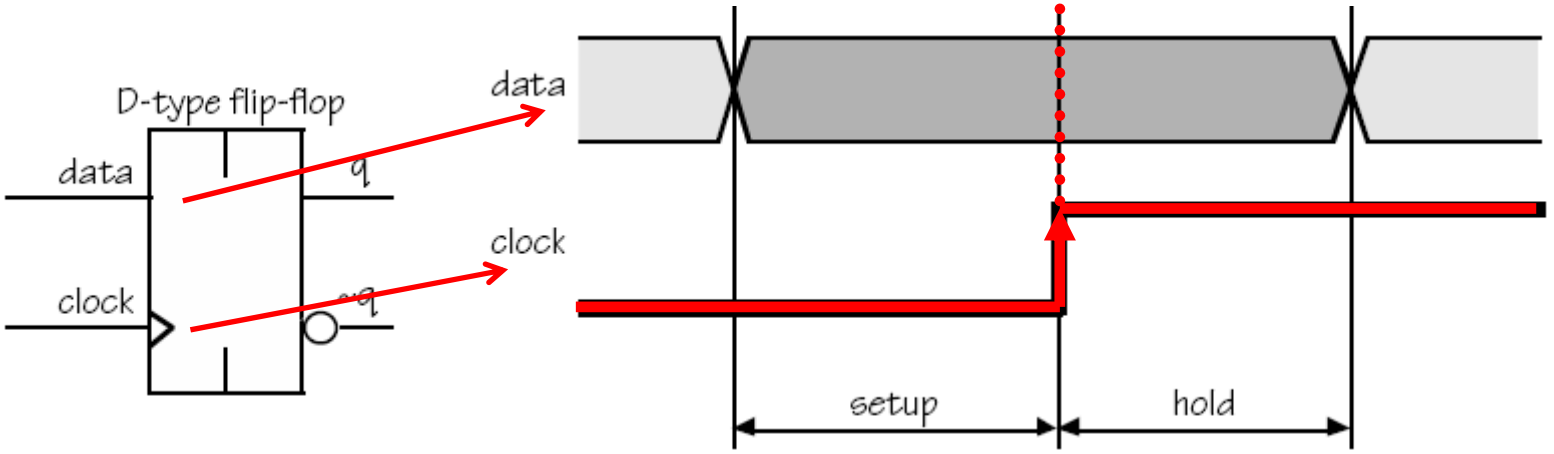
Talasni oblik D-FF



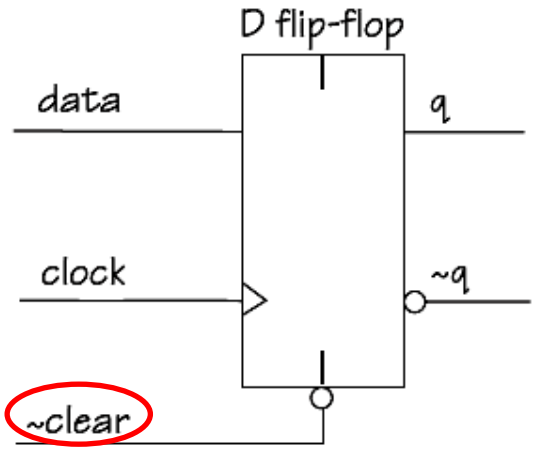
Talasni oblik pozitivno trigerovanog D flip-flopa

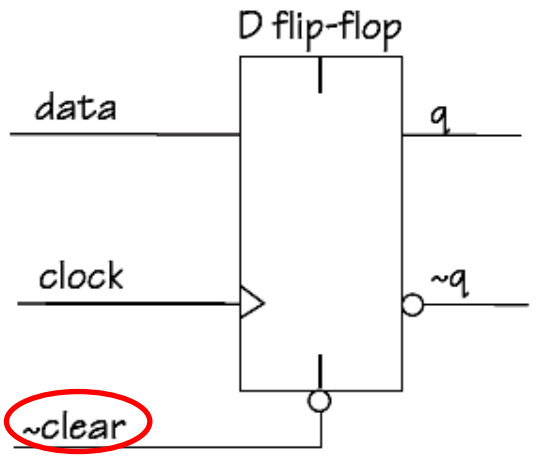
Vreme uspostavljanja signala

- **Setup** (uspostavljanje) and **hold** (držanje) vremena za D-FF su prikazana slici.
- Vremenski period u kome vrednost na data ulazu **mora ostati stabilan** je prikazan tamnijom bojom.
- Ova vremena su posledica **internih kašnjenja** u samom FF.
- Setup i hold definišu **brzinu** same tehnologije!



D-FF sa dodatnim ulazima





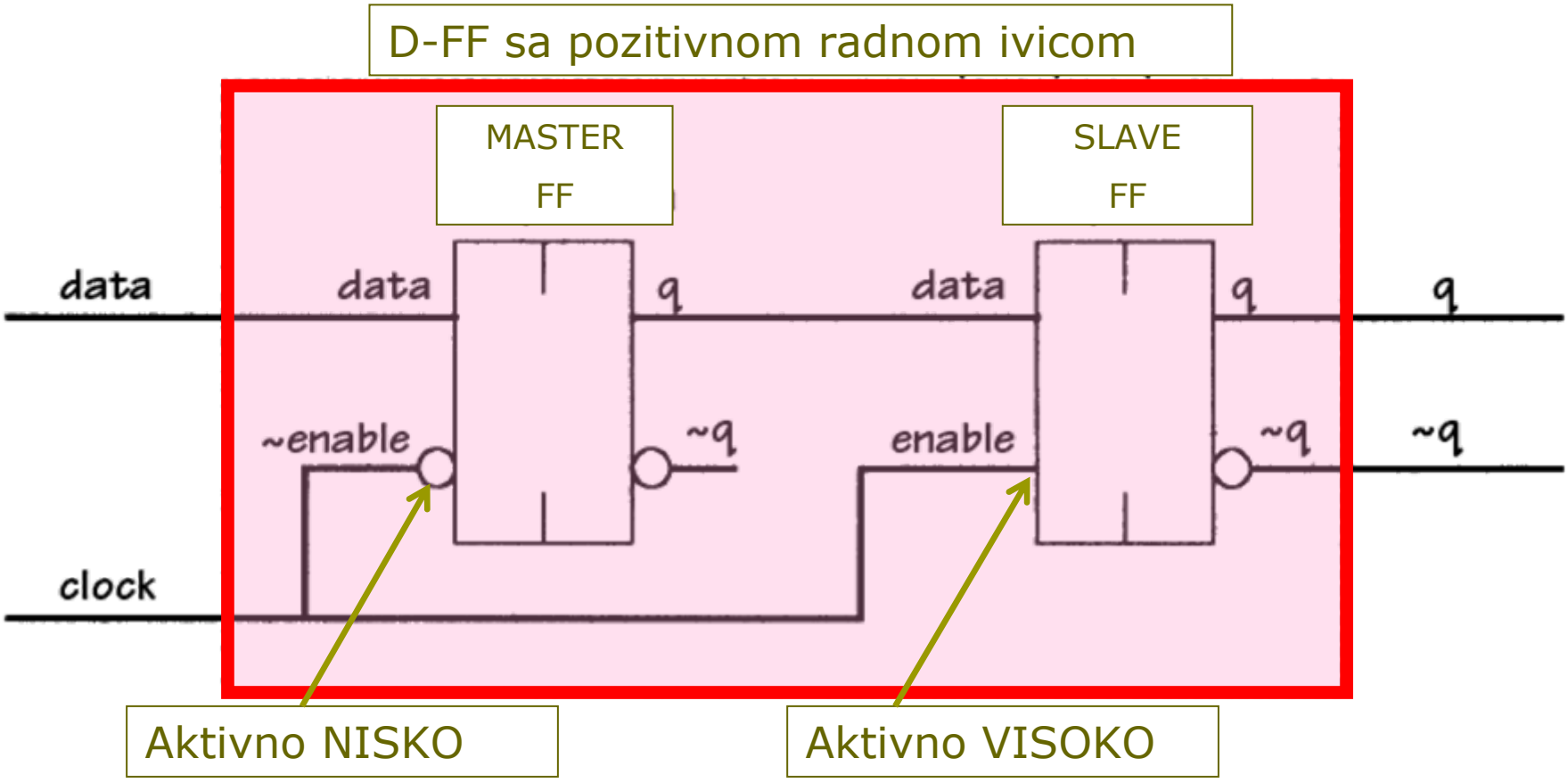
~clear	clock	data	$q_{(n+)}$	$\sim q_{(n+)}$
0	?	?	0	1
1	↑	0	0	1
1	↑	1	1	0

Asinhroni CLEAR

clock	~clear	data	$q_{(n+)}$	$\sim q_{(n+)}$
↑	0	?	0	1
↑	1	0	0	1
↑	1	1	1	0

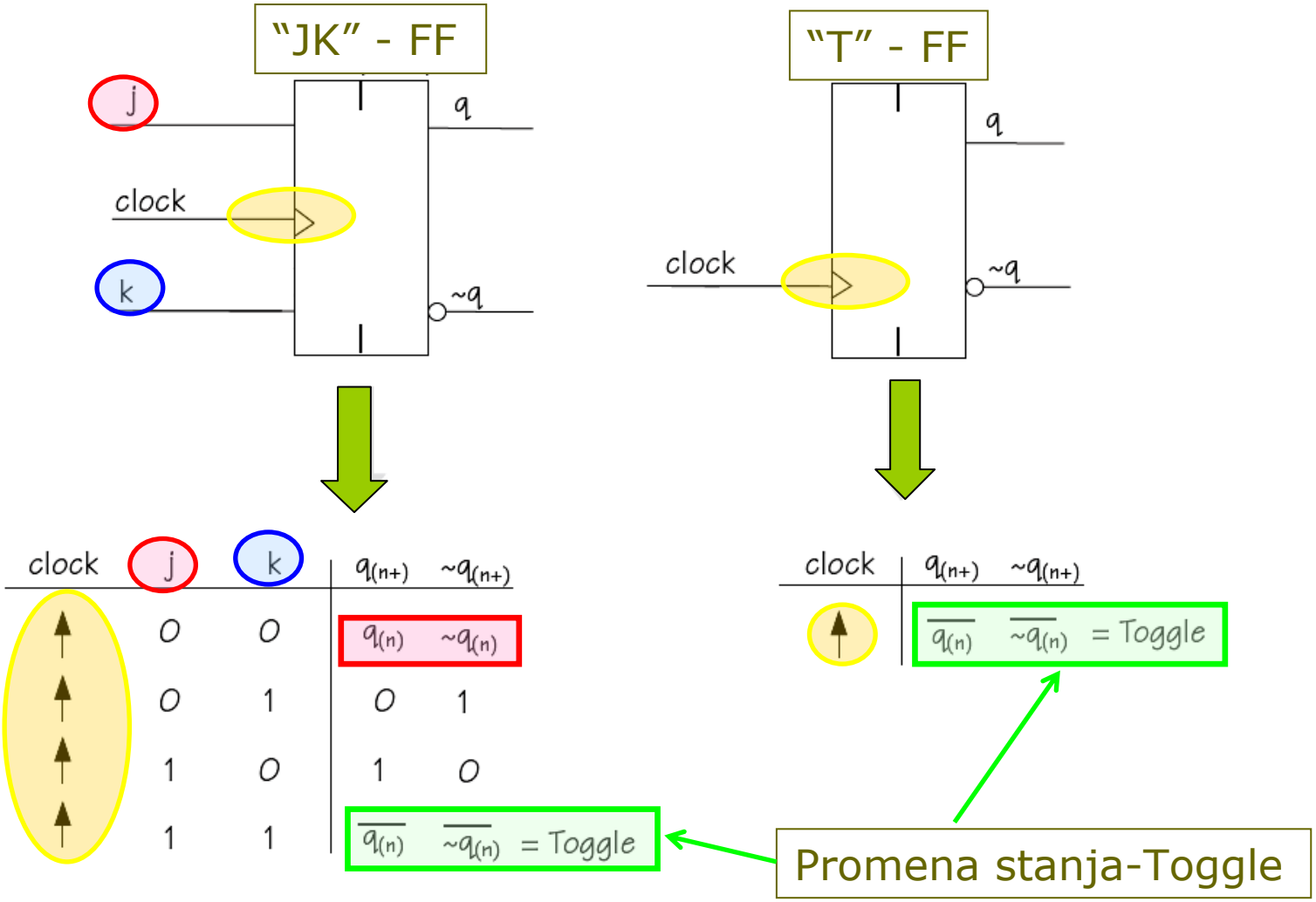
Sinhroni CLEAR

MASTER-SLAVE D-FF

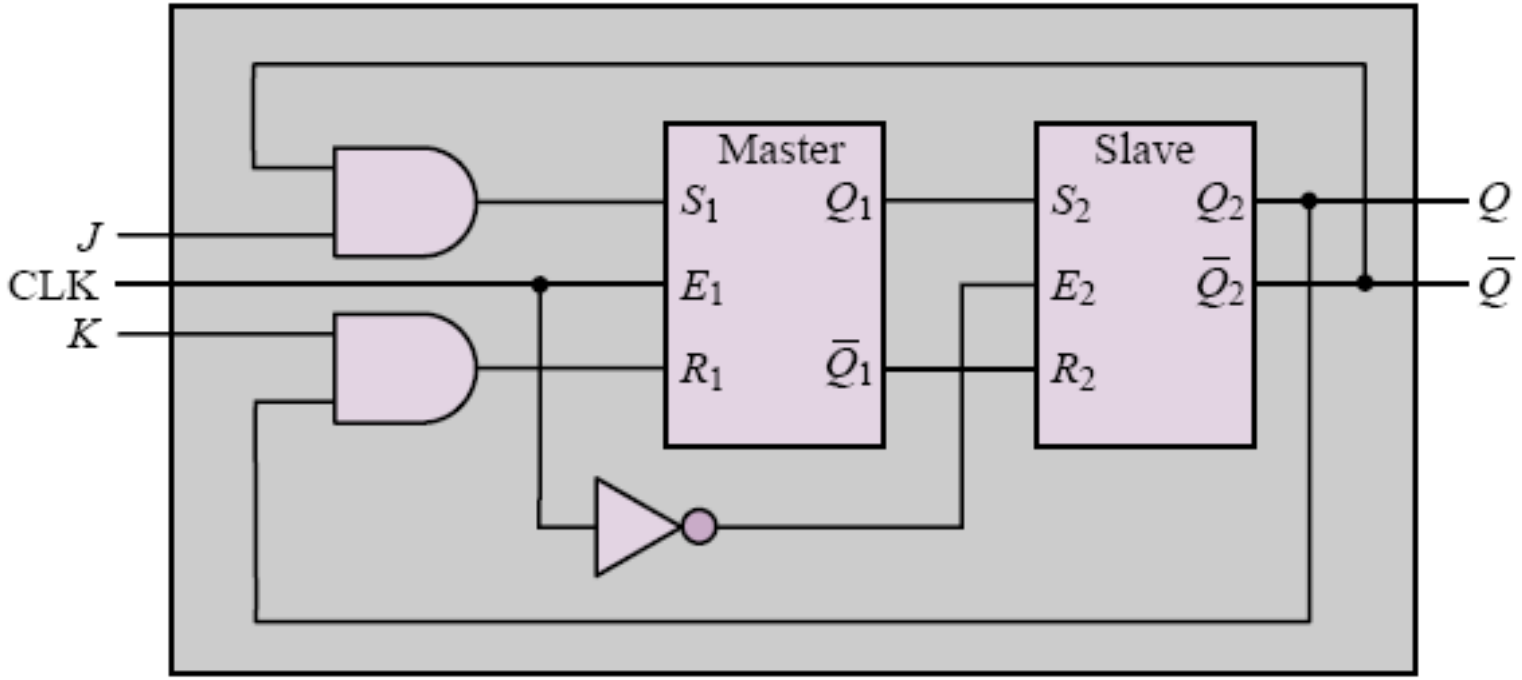


- Iako je sve realiuzovano preko logičkih NIVOA, spolja gledano sve je sinhronizovano sa **rastućom ivicom** CLOCK-a.

JK & T - FF

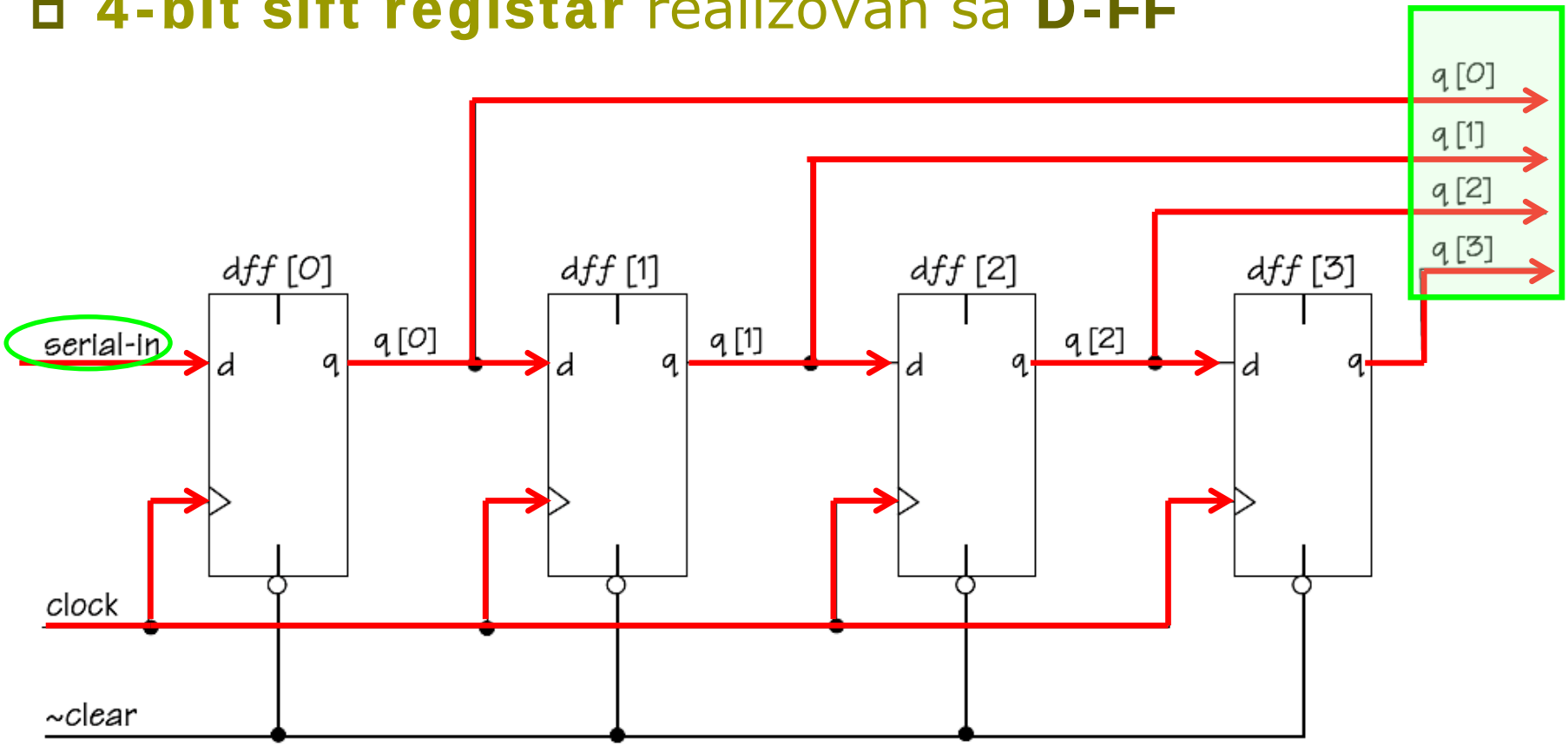

"JK" - FF
j
clock
k
q
~q
"T" - FF
clock
q
~q
clock
j
k
$q_{(n+)}$
$\sim q_{(n+)}$
0 0 $q_{(n)}$ $\sim q_{(n)}$
0 1 0 1
1 0 1 0
1 1 $\overline{q_{(n)}}$ $\overline{\sim q_{(n)}}$ = Toggle
clock
$q_{(n+)}$
$\sim q_{(n+)}$
$\overline{q_{(n)}}$ $\overline{\sim q_{(n)}}$ = Toggle
Promena stanja-Toggle

JK & T – FF, realizacija

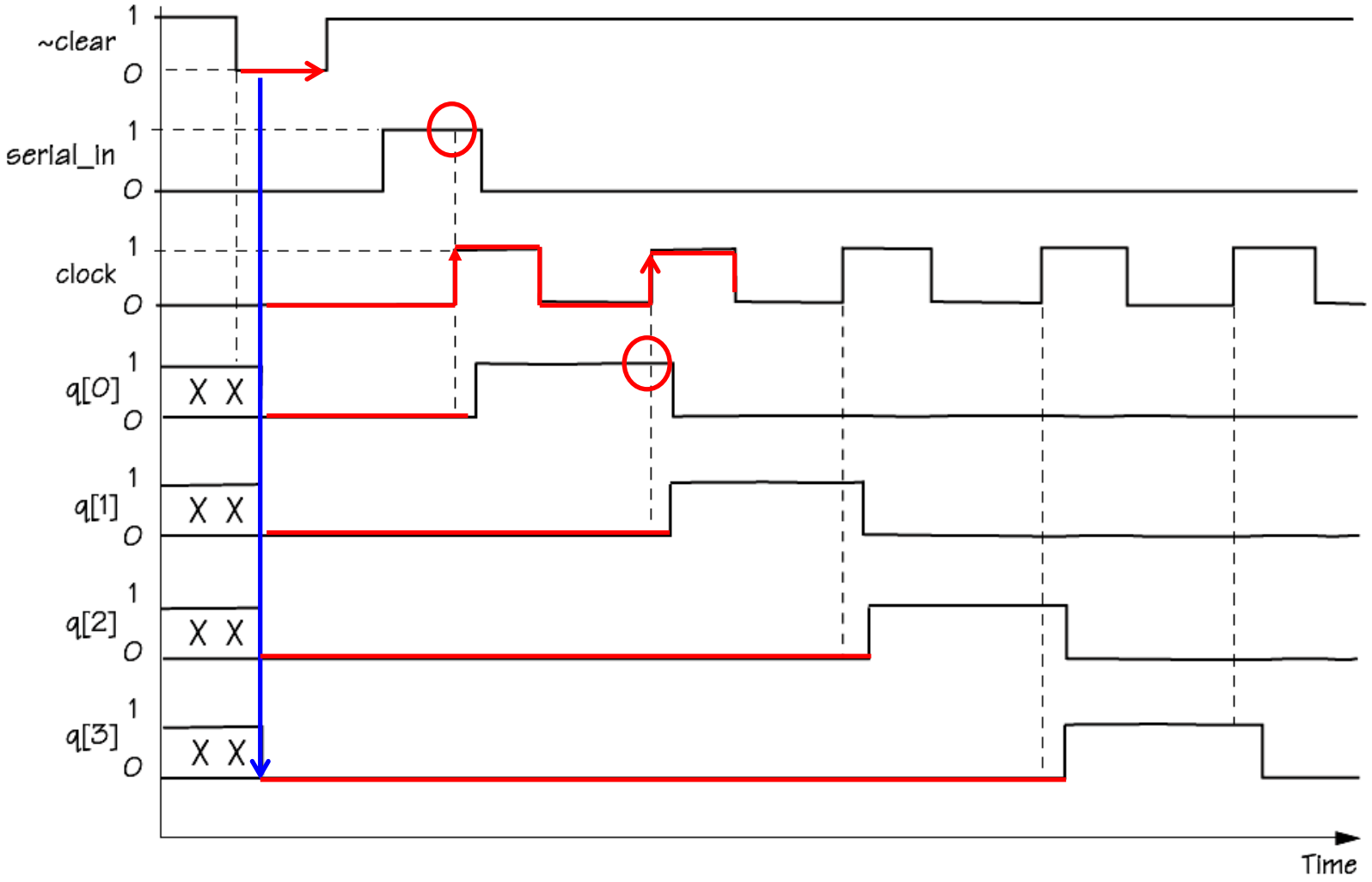


Šift registri - SIPO

- Posebna vrsta registara koja **pomera** (**shift**) binarne podatke za jedan bit.
- **4-bit šift registar** realizovan sa **D-FF**

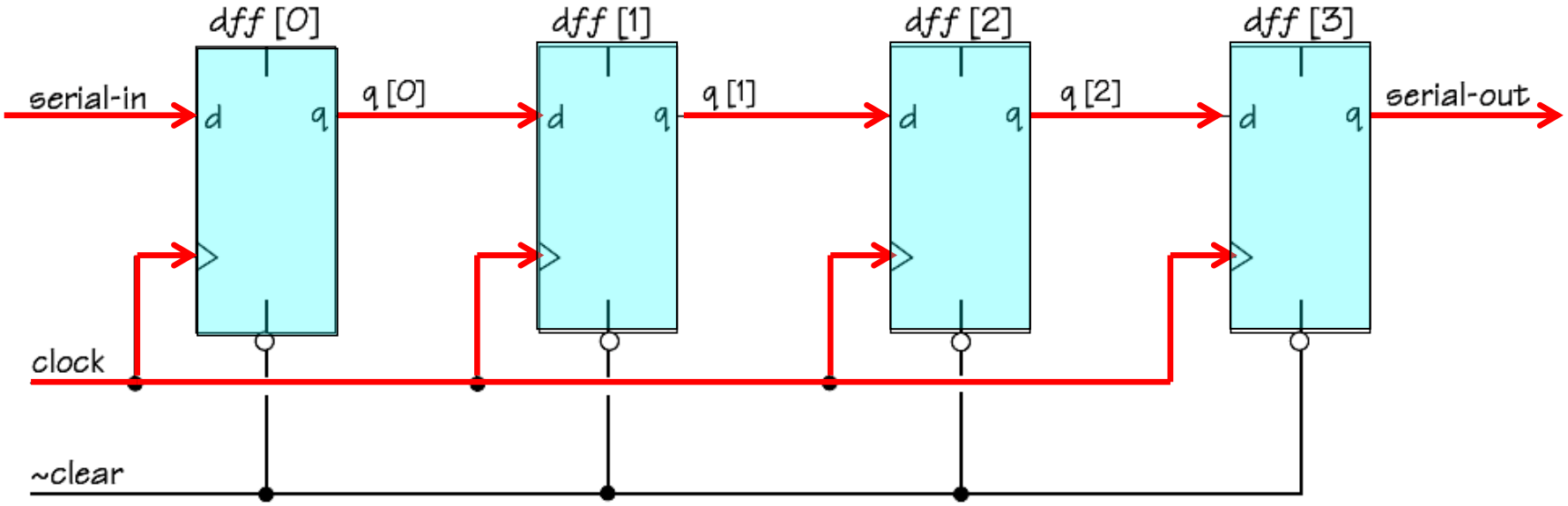


Talasni oblik SIPO shift registra



SISO šift registar

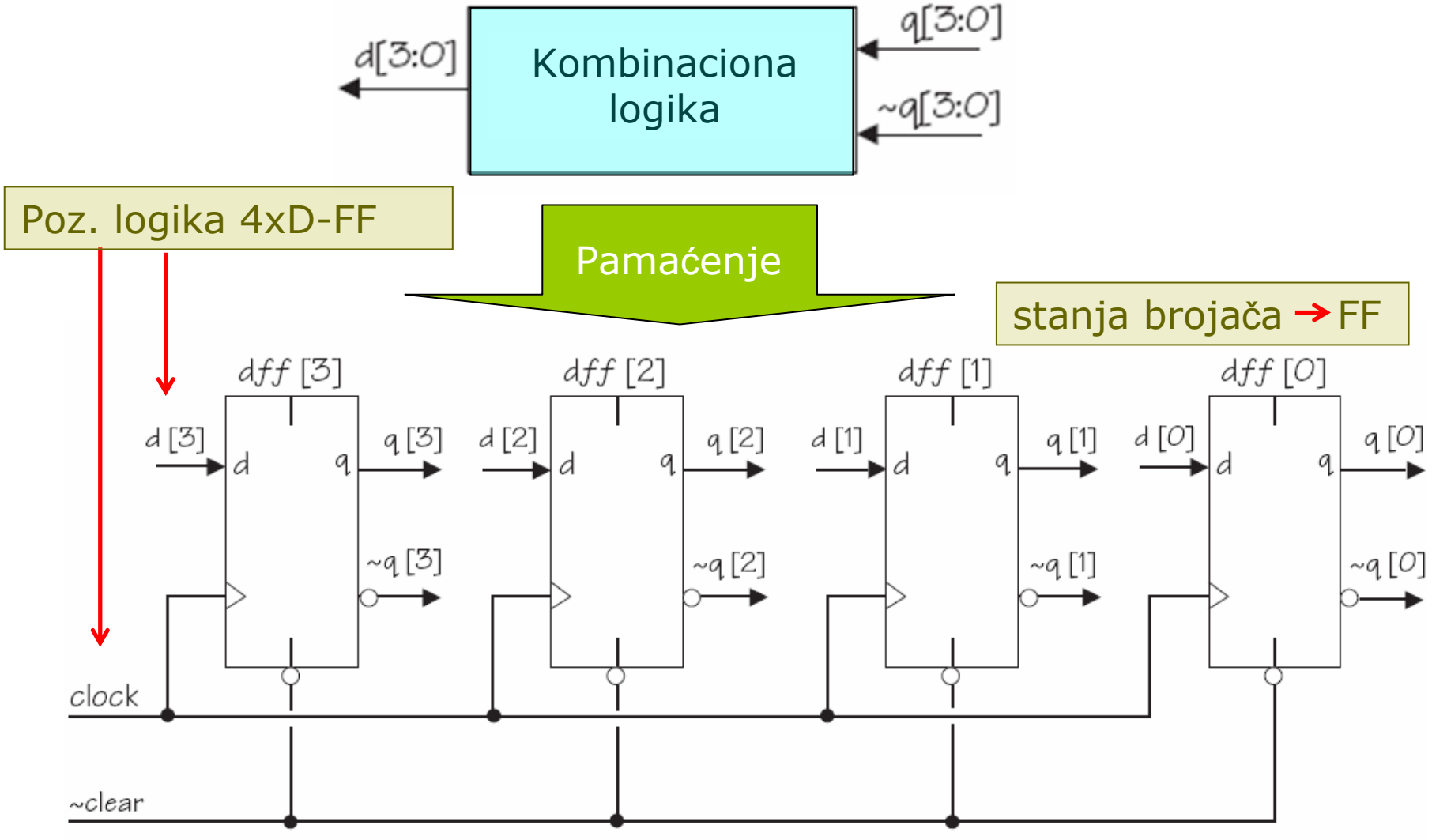
- Serijski ulaz serijski izlaz



Brojači

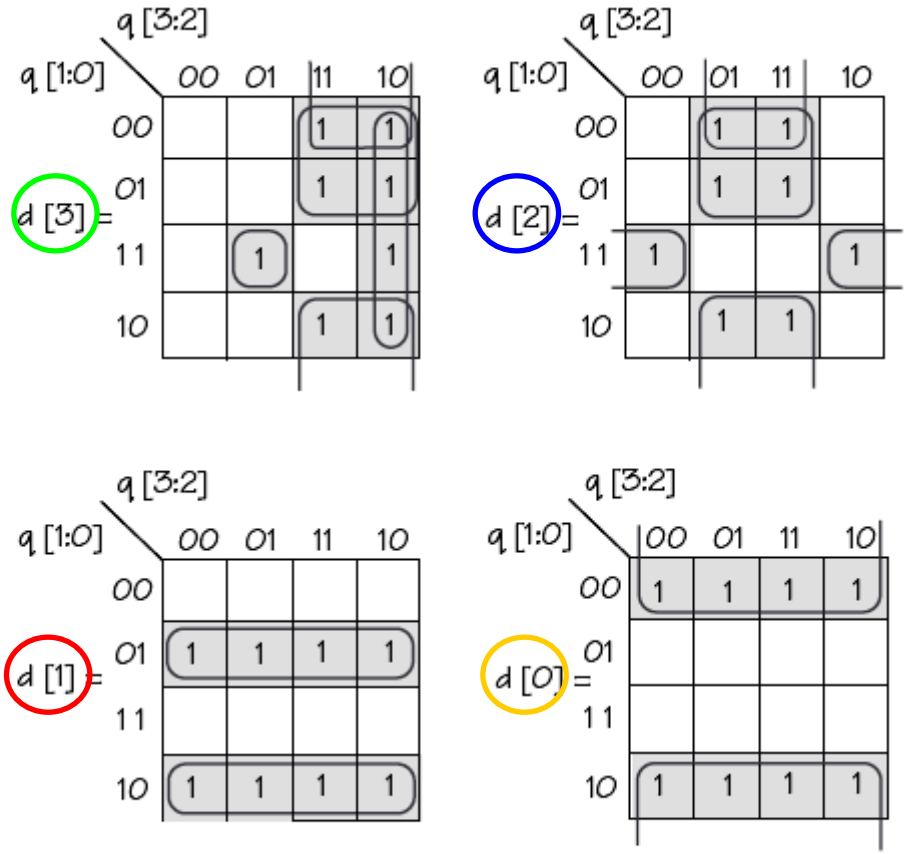
- Brojačke funkcije se veoma često koriste u digitalnim sistemima.
- Moduo brojača je broj stanja kroz koje prolazi brojač pre povratka na početno stanje.
- Primer: brojač koji broji od 0000_2 do 1111_2 u binarnom sistemu (ili od 0 to 15 u decimalnom) ima moduo šesnaest (16)

Brojač: moduo 16 sa D-FF



Brojač u kombinacionoj tehn.

Current Value q [3:0]	Next Value d [3:0]
0 0 0 0	0 0 0 1
0 0 0 1	0 0 1 0
0 0 1 0	0 0 1 1
0 0 1 1	0 1 0 0
0 1 0 0	0 1 0 1
0 1 0 1	0 1 1 0
0 1 1 0	0 1 1 1
0 1 1 1	1 0 0 0
1 0 0 0	1 0 0 1
1 0 0 1	1 0 1 0
1 0 1 0	1 0 1 1
1 0 1 1	1 1 0 0
1 1 0 0	1 1 0 1
1 1 0 1	1 1 1 0
1 1 1 0	1 1 1 1
1 1 1 1	0 0 0 0



$d[3] = (q[3] \,\&\, \overline{q[1]}) \mid (q[3] \,\&\, \overline{q[2]}) \mid (q[3] \,\&\, \overline{q[0]}) \mid (\overline{q[3]} \,\&\, q[2] \,\&\, q[1] \,\&\, q[0])$

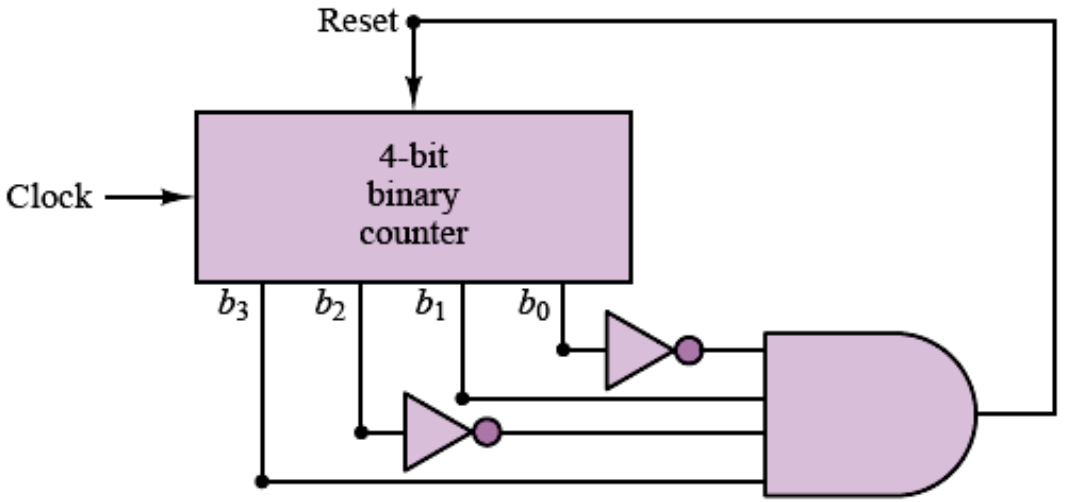
$d[2] = (q[2] \,\&\, \overline{q[1]}) \mid (q[2] \,\&\, \overline{q[0]}) \mid (\overline{q[2]} \,\&\, q[1] \,\&\, q[0])$

$d[1] = (q[1] \,\&\, \overline{q[0]}) \mid (\overline{q[1]} \,\&\, q[0])$

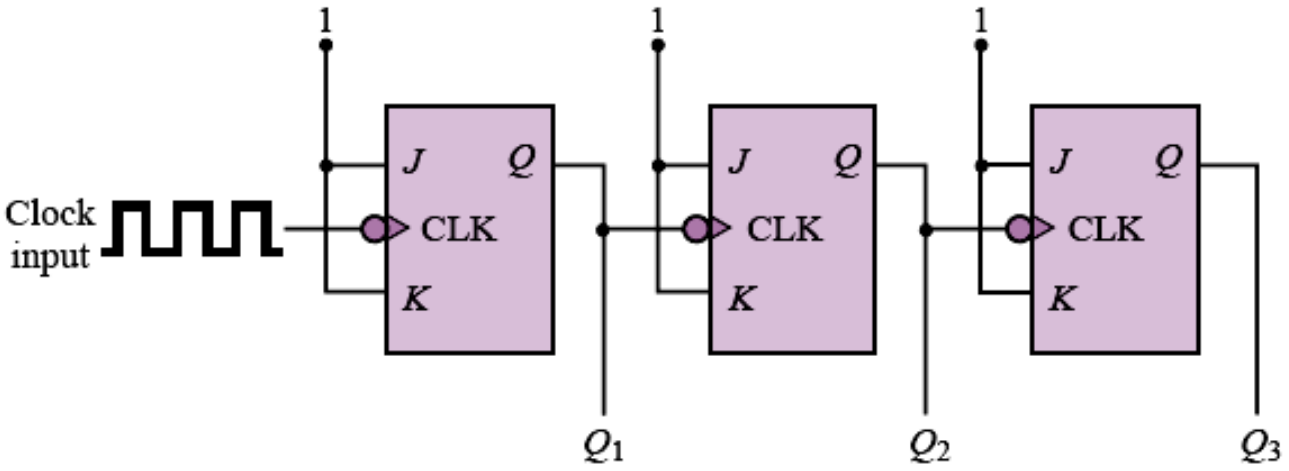
$d[0] = (\overline{q[0]})$

Dekadni Brojač

Input pulses	b_3	b_2	b_1	b_0	
0	0	0	0	0	
1	0	0	0	1	
2	0	0	1	0	
3	0	0	1	1	
4	0	1	0	0	
5	0	1	0	1	
6	0	1	1	0	
7	0	1	1	1	
8	1	0	0	0	
9	1	0	0	1	
10	1	0	1	0	Reset



Ripple - brojač



Input	Q_3	Q_2	Q_1
⎍	0	0	0
⎍	0	0	1
⎍	0	1	0
⎍	0	1	1
⎍	1	0	0
⎍	1	0	1
⎍	1	1	0
⎍	1	1	1

Sedmosegmentni dekoder