

Pomnenje s kvantnimi celičnimi avtomati

Jasmina Pegan, Blaž Rojc

November 21, 2019

1 Uvod

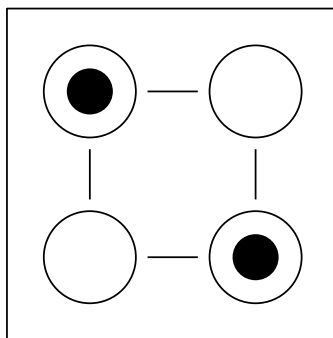
Kvantne celice, osnovni gradniki kvantnih celičnih avtomatov, same po sebi ne omogočajo pomnenja [?]. Podobno kot pri tranzistorjih mora načrtovalec digitalnega vezja sestaviti strukturo iz kvantnih celic, ki pomnenje omogoči. V tem delu bomo predstavili principe pomnenja v kvantnih celičnih avtomatih in nekatere strukture iz kvantnih celic, ki simulirajo delovanje tradicionalnih pomnilnih celic.

2 Pregled področja

2.1 Kvantni celični avtomati

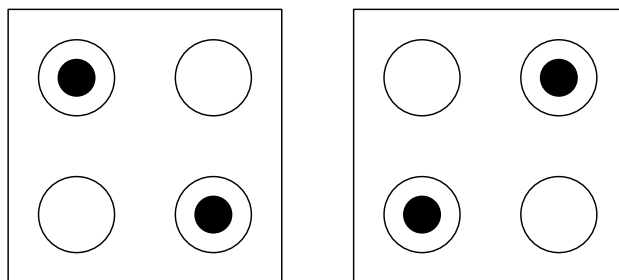
TODO: kaka citacija? šel sem po Janežu + Mrazu

Kvantni celični avtomati predstavljajo izvedbo celičnih avtomatov, v katerih so osnovni gradniki kvantne celice. Kvantna celica je konstrukt kvadratne oblike, ki vsebuje štiri okrogle kvantne pike in štiri tunele. Vsak tunel povezuje dve sosednji kvantni piki. Shema takšne celice je prikazana na sliki 1.



Slika 1: Shema kvantne celice. Poleg štirih kvantnih pik in tunelov sta prikazana še dva elektrona, ki se nahajata v piki levo zgoraj in v piki desno spodaj.

V kvantni celici sta ujeta dva elektrona. Vsak od njiju se lahko nahaja v eni izmed štirih kvantnih pik, med katerimi se lahko pomika prek tunelov, ki jih povezujejo. Elektrona se zaradi odbojnih sil med njima v odsotnosti zunanjih vplivov postavita v eno izmed dveh stabilnih stanj, prikazanih na sliki 2.

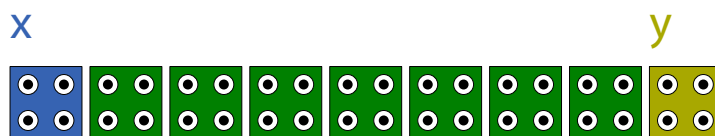


Slika 2: Stabilni stanja kvantne celice. Na levi je prikazano stanje $P = -1$, na desni pa stanje $P = 1$.

Stanje $P = -1$ interpretiramo kot logično vrednost 0, stanje $P = 1$ pa kot logično vrednost 1 [?]. Tuneli med pikami so opremljeni s pregradami. Elektrona lahko prosto prehajata med pikami, dokler pregrade niso vzpostavljene. Vzpostavitev pregrad prehajanje elektronov onemogoči, kar nam omogoča, da stanje celice odčitamo. Take celice označujemo kot izhodne. Celico lahko tudi prisilimo, da zasede določeno stanje. Take celice označujemo kot vhodne.

Kvantni celični avtomat je konstrukt, ki vsebuje eno ali več kvantnih celic, ki zaradi medsebojne bližine interagirajo druga z drugo. Sile med elektroni težijo k vzpostavitvi stanja z najmanjšo skupno energijo. V odsotnosti zunanjih sil se kvantna celica postavi v eno izmed dveh stabilnih stanj z verjetnostjo $\frac{1}{2}$, ampak če so v njeni bližini druge kvantne celice, se ta verjetnost spremeni. S pravilno postavitvijo celic lahko dosežemo, da se avtomat obnaša kot tradicionalna Boolova preklompna funkcija.

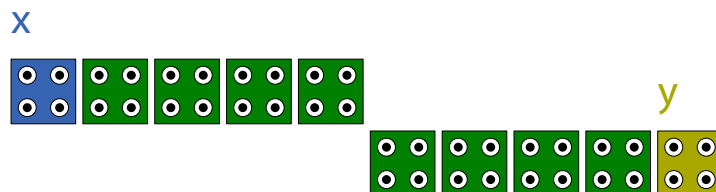
Gradnjo kompleksnejših struktur pričnemo z osnovnimi gradniki. Osnovni gradniki predstavljajo poln funkcijski nabor, s katerim lahko implementiramo poljubno preklompno funkcijo. Najenostavnejši gradnik je vodilo, prikazano na sliki 3.



Slika 3: Vodilo. Signal potuje od vhoda x do izhoda y .

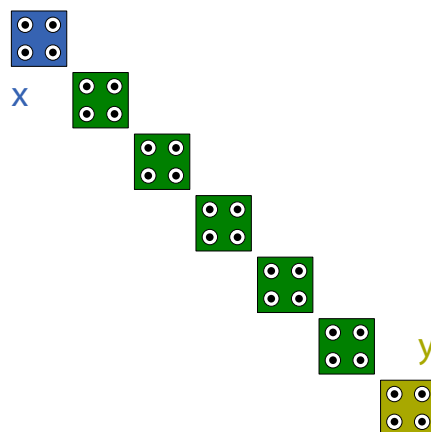
Vodila preslikajo vhodno vrednost na izhod. Uporabljamo jih za pove-

zovanje gradnikov v avtomatu. Drugi gradnik je negator, prikazan na sliki 4.



Slika 4: Negator. Signal potuje od vhoda x do izhoda y in je pri tem negiran.

Negatorje lahko uporabimo tudi kot diagonalno vodilo, kjer je dolžina takega vodila sodo število. Primer take uporabe je prikazan na sliki 5.



Slika 5: Vodilo iz negatorjev. Signal potuje od vhoda x do izhoda y . Pri tem je negiran $2k$ -krat, zato se logična vrednost ne spremeni.

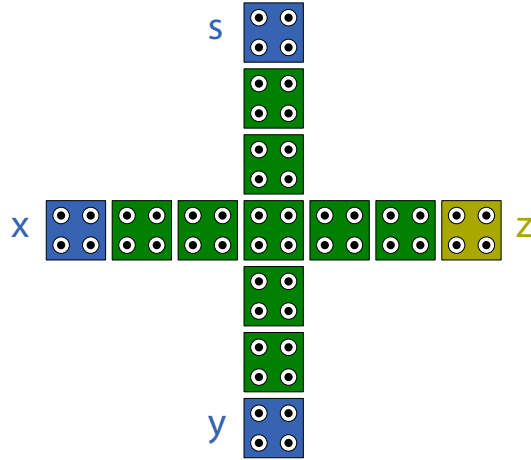
Zadnji osnovni gradnik so večinska oz. majoritetna vrata, prikazana na sliki 6.

Izhod majoritetnih vrat zavzame večinsko vrednost na treh vseh vhodih. Z vezavo konstante na enega izmed vhodov majoritetnih vrat dobimo gradnik, ki predstavlja konjunkcijo ali disjunkcijo, kot je prikazano na sliki 7.

Tok podatkov v avtomatu usmerjamo s pomočjo urinih con. Vsaki celici dodelimo eno izmed štirih urin con, ki se ciklično izmenjujejo. Tako omogočimo logično separacijo delov avtomata in način interakcije med njimi.

2.2 Problematika pomnenja

Ko je ura odsotna, je stanje kvantne celice v odsotnosti zunanjih sil neterministično. Tako kvantnih celic samih ne moremo uporabiti za hrambo podatkov. Uvedba ure omogoča, da kvantne celice spravimo v zaklenjeno stanje, v katerem prehajanje elektronov med pikami ni mogoče, kar pomeni,



Slika 6: Majoritetna vrata. Izhod z zavzame večinsko vrednost na vhodih s , x in y .

s	x	y	z	
0	0	0	0	$AND(x, y)$
0	0	1	0	
0	1	0	0	
0	1	1	1	
1	0	0	0	$OR(x, y)$
1	0	1	1	
1	1	0	1	
1	1	1	1	

Slika 7: Tabela vrednosti izhoda majoritetnih vrat pri različnih vhodnih vrednostih. Majoritetna vrata se glede na x in y pri $s = 0$ obnašajo kot konjunkcija, pri $s = 1$ pa kot disjunkcija.

da je stanje zaklenjene kvantne celice deterministično. Za pomnenje potrebujemo še način za nadzor stanja, ki naj ga zaklenjena celica zavzame. To lahko dosežemo z veriženjem celic, ki ne pripadajo isti urini coni. Takšna veriga je prikazana na sliki **TODO**.

TODO: slika verige s sekvenčnimi urinimi conami

Za samostojno pomnenje to verigo sklenemo v obroč, prikazan na sliki **TODO**.

TODO: slika obroča s sekvenčnimi urinimi conami

V obroču se stanje med potekom urinega cikla ohranja, kot je prikazano na sliki **TODO**.

TODO: shema / tabela s stanji v obroču med urinim ciklom (simulacija v QCA Designer)

Za smiselno uporabo želimo to strukturo prirediti tako, da lahko hranjeno stanje poljubno nastavljamo. Kako to storimo, je stvar implementacije.

V nadaljevanju bomo predstavili nekaj obstoječih implementacij pomnilnih struktur.

3 Pomnilne celice

Za gradnjo sekvenčnih vezij poleg logičnih vrat potrebujemo tudi pomnilne celice. Osnovne pomnilne celice, ki omogočajo hranjenje enega bita informacije, so RS (Reset Set), JK (Jump Kill), T (Trigger) in D (Delay) pomnilna celica.

Pravilnostne tabele:

3.1 Obstoječe implementacije

V članku [?] iz leta 2003 so predstavljene implementacije štirih osnovnih pomnilnih celic. Vse celice porabijo en urin cikel za procesiranje vhodov. Pri postavitvi celic vir uporablja pravilo čimbolj preprostih oblik urinih con in čimmanj le-teh. Po drugi strani pa so nekatere celice zasukane za kot 45° ali zamaknjene za polovico svoje velikosti, kar lahko negativno vpliva na izvedljivost implementacije. Vir uporablja križanje linij, kar je možno doseči z uporabo alternirajočega vodila (rotiranih celic).

Vse implementacije pomnilnih celic v viru [?] iz leta 2018 temeljijo na izpeljavah izraza, ki opisuje RS celico. Te implementacije porabijo manj celic ter ponudijo tako izhodno stanje Q kot njegovo negacijo \bar{Q} , so pa zato urine cone bolj kompleksne in uporabljeni so manj stabilni negatorji kot v prvem viru. Vse celice se izvedejo v 1.5 urinega cikla, le D pomnilna celica ima latenco 1.25 urinega cikla.

3.2 Ideje in popravki (?)

- Uporaba večjih razdalj med gradniki
- Konsistentna uporaba bolj stabilne negacije

4 Implementacija

TODO: primerjaj implementacije: potratnost, izvedljivost, konsistentnost
V tem poglavju bomo primerjali predlagane implementacije pomnilnih celic.

V viru [?] so enačbe posameznih pomnilnih celic izpeljane iz enačb RS celice:

$$Qb = \overline{B \cdot CLK} \cdot \bar{Q} = B \cdot CLK + \bar{Q}, \quad (1)$$

$$Q = \overline{A \cdot CLK} \cdot Qb = A \cdot CLK + \bar{Qb} = A \cdot CLK + \overline{B \cdot CLK} \cdot Q, \quad (2)$$

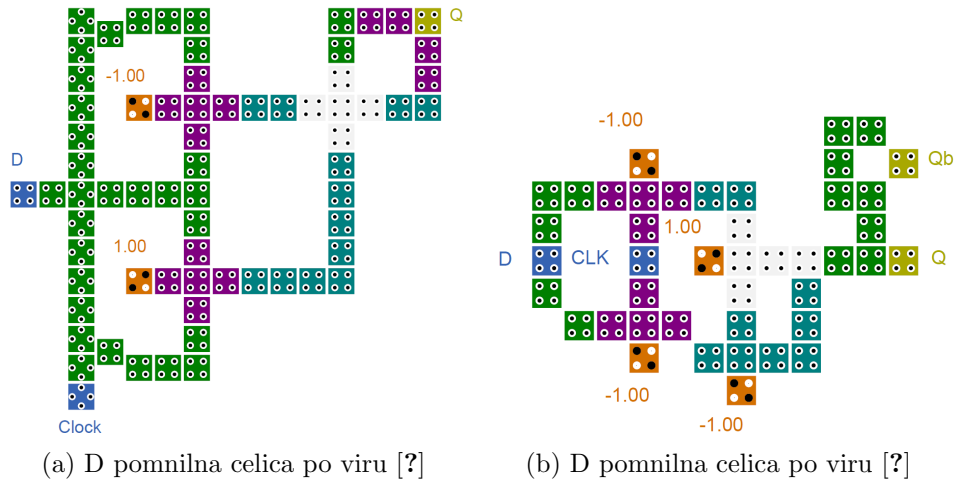
pri čemer je Q izhod celice, Qb njegova negacija, A in B sta vhoda ter CLK je vhodni urin signal.

Pomnilna celica	Vir	Število majoritetnih vrat	Število negatorjev	Število celic
D	[?]	3	2	68
	[?]	4	2	43
JK	[?]	5	2	90
	[?]	6	2	78
RS	[?]	4	1	76
	[?]	4	2	38
T	[?]	5	4	92
	[?]	6	2	81

Table 1: Primerjava števila gradnikov za sestavo pomnilnih celic.

4.1 D pomnilna celica

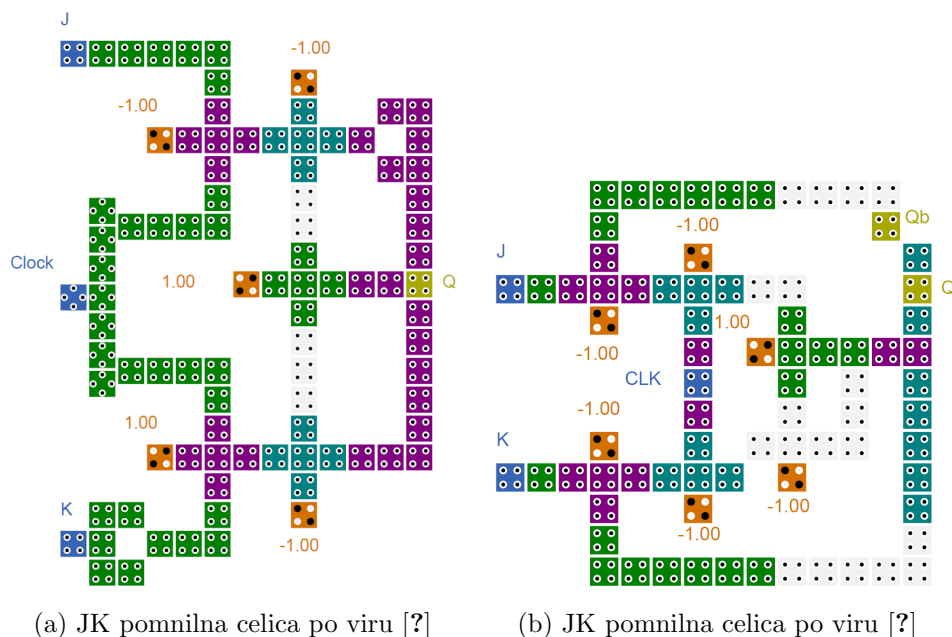
Na sliki 8 vidimo implementaciji D pomnilne celice po obeh virih. Vir [?] uporabi križanje linij urinega signala in vhoda D z uporabo alternirajočega vodila zasukanih celic. Vir [?] v enačbah 1 in 2 zamenja vhoda A in B z D in \overline{D} ter tako dobi enačbo D pomnilne celice.



Slika 8: Implementaciji D pomnilne celice

4.2 JK pomnilna celica

Na sliki 9 vidimo implementaciji JK celice po obeh virih. Vir [?] uporabi alternirajoče vodilo za širjenje urinega signala za bolj stabilen prenos signala naprej, vidimo tudi dva okrepljena negatorja. Da izpelje enačbo JK pomnilne celice, vir [?] strukturi za RS celico doda dve majoritetni vrati.



Slika 9: Implementaciji JK pomnilne celice

4.3 RS pomnilna celica

Na sliki 10 vidimo implementaciji RS celice po obeh virih. Vir [?] kot pri JK celici uporabi alternirajoče vodilo za širjenje urinega signala in en okrepljen negator. Vir [?] v enačbah 1 in 2 zamenja vhoda A in B s S in R ter tako dobi enačbo RS pomnilne celice.

4.4 T pomnilna celica

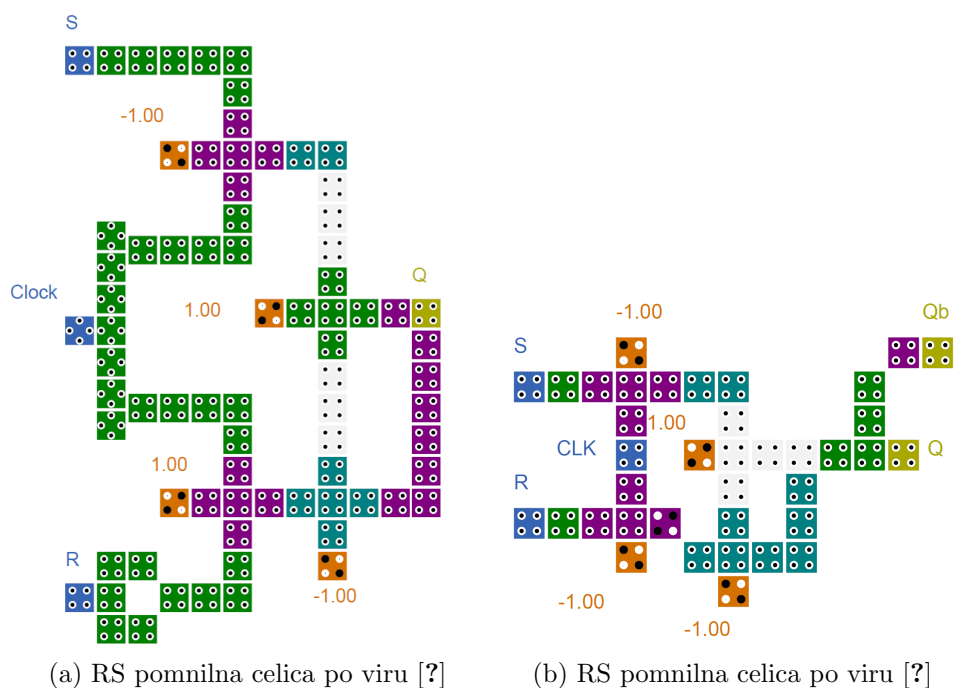
Na sliki 11 vidimo implementaciji T celice po obeh virih. Vir [?] spet uporabi alternirajoče vodilo za širjenje urinega signala in dva okrepljena negatorja. Tudi prenosa z alternirajočega vodila prek polovično zamaknjenih celic učinkujeta kot negatorja. Vir [?] pridobi enačbo za T pomnilno celico iz JK pomnilne celice s povezavo vhodov J in K .

4.5 Analiza delovanja

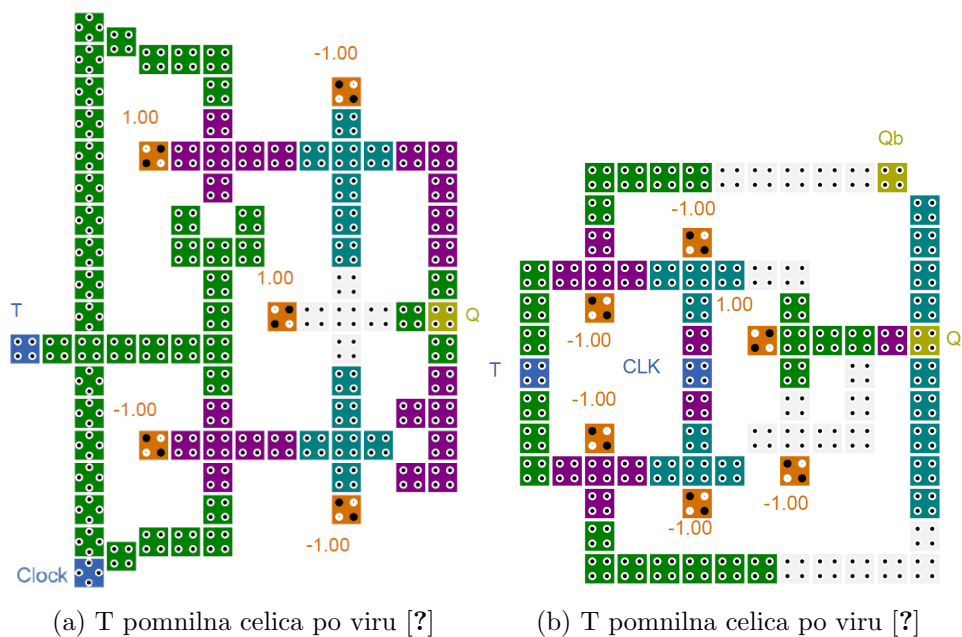
V sledečem poglavju bomo analizirali rezultate simulacij pomnilnih celic in jih primerjali s pričakovanimi pravilnostnimi tabelami.

4.5.1 D pomnilna celica

Pravilnostna tabela D pomnilne celice je predstavljena v tabeli 2.



Slika 10: Implementaciji RS pomnilne celice



Slika 11: Implementaciji T pomnilne celice

Na sliki 12 vidimo obnašanje Q celice v odvisnosti od vhoda D in urinega

d	D^1q
0	0
1	1

Table 2: Pravilnostna tabela D pomnilne celice.

signala *Clock*. Že takoj lahko vidimo, da se rezultata simulacij razlikujeta. Iz razbranega lahko sestavimo pravilnostni tabeli 3.

vir [?]			vir [?]		
D	$Clock$	Q	D	$Clock$	Q
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	1	1	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	1	1	0

Table 3: Rezultat simulacije D pomnilne celice.

4.5.2 JK pomnilna celica

Pravilnostna tabela JK pomnilne celice je predstavljena v tabeli 4.

k	j	D^1q
0	0	q
0	1	1
1	0	0
1	1	\bar{q}

Table 4: Pravilnostna tabela JK pomnilne celice.

Na sliki 13 vidimo obnašanje Q celice v odvisnosti od vhodov J in K ter urinega signala *Clock*. Iz razbranega lahko sestavimo pravilnostni tabeli 5.

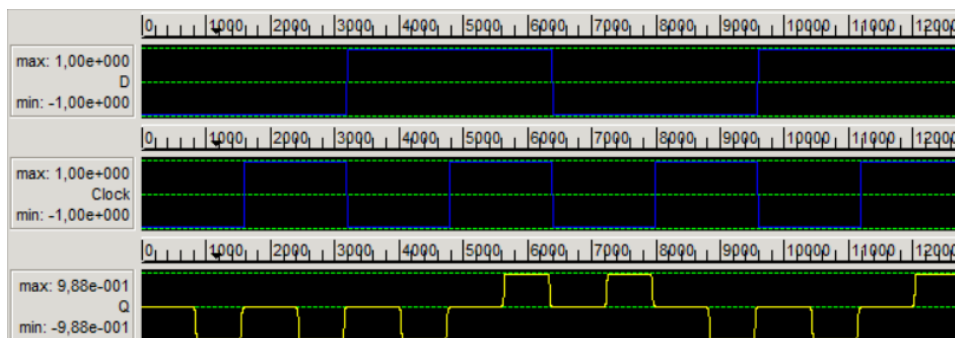
4.5.3 RS pomnilna celica

Pravilnostna tabela RS pomnilne celice je predstavljena v tabeli 6.

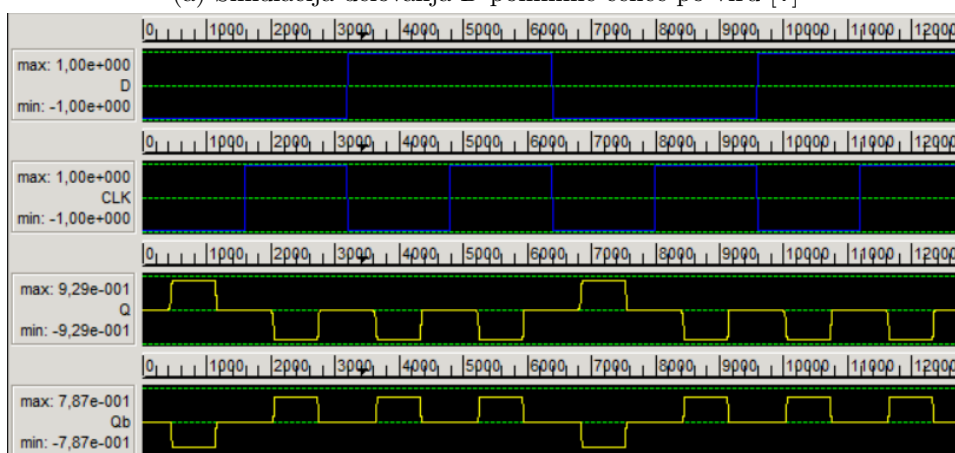
Na sliki 14 vidimo obnašanje Q celice v odvisnosti od vhodov R in S ter urinega signala *Clock*. Iz razbranega lahko sestavimo pravilnostni tabeli 7.

4.5.4 T pomnilna celica

Pravilnostna tabela T pomnilne celice je predstavljena v tabeli 8.



(a) Simulacija delovanja D pomnilne celice po viru [?]



(b) Simulacija delovanja D pomnilne celice po viru [?]

Slika 12: Simulaciji D pomnilne celice

Na sliki 15 vidimo obnašanje Q celice v odvisnosti od vhoda T in urnega signala $Clock$. Že takoj lahko vidimo, da se rezultata simulacij razlikujeta. Iz razbranega lahko sestavimo pravilnostni tabeli 9.

5 Zaključek

TODO: kratek povzetek glavnih ugotovitev + smernice za nadaljnje delo

vir [?]				vir [?]			
J	$Clock$	K	Q	J	$Clock$	K	Q
0	0	0	1	0	0	0	0
0	0	1	1	0	0	1	0
0	1	0	1	0	1	0	0
0	1	1	1	0	1	1	0
1	0	0	1	1	0	0	0
1	0	1	1	1	0	1	0
1	1	0	1	1	1	0	0
1	1	1	0	1	1	1	1
<hr/>				<hr/>			
0	0	0	1	0	0	0	1
0	0	1	1	0	0	1	0
0	1	0	1	0	1	0	0
0	1	1	1	0	1	1	0
1	0	0	1	1	0	0	0
1	0	1	1	1	0	1	0
1	1	0	1	1	1	0	0
1	1	1	0	1	1	1	1

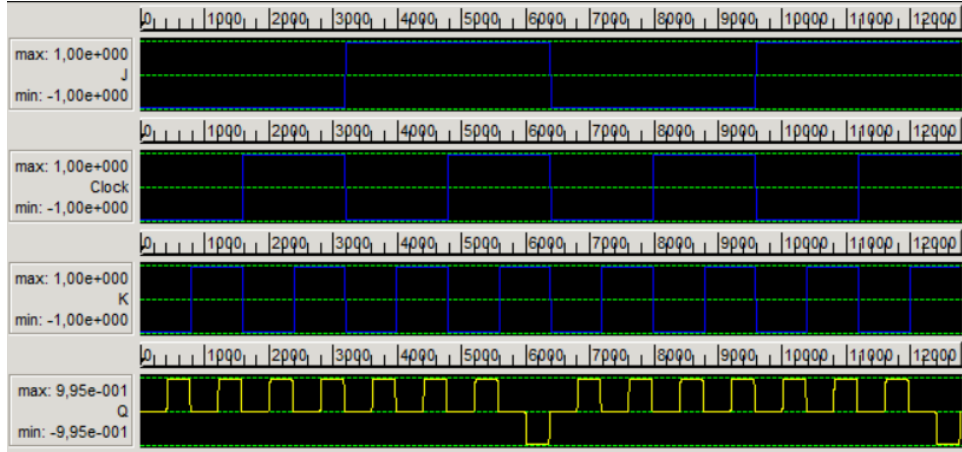
Table 5: Rezultat simulacije JK pomnilne celice.

r	s	D^1q
0	0	q
0	1	1
1	0	0
1	1	x

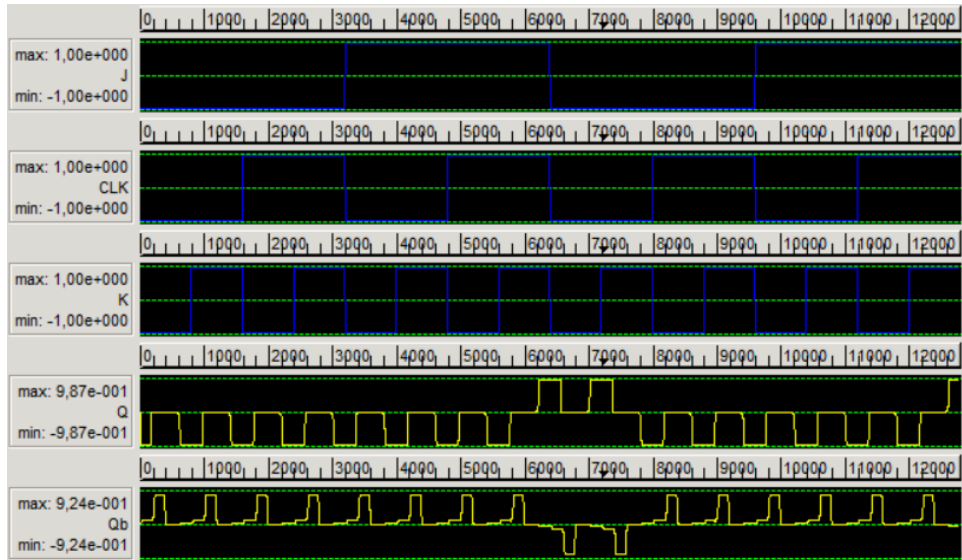
Table 6: Pravilnostna tabela RS pomnilne celice.

vir [?]				vir [?]			
S	$Clock$	R	Q	S	$Clock$	R	Q
0	0	0	0	0	0	0	1
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	1	1	1	1	1
<hr/>				<hr/>			
0	0	0	1	0	0	0	1
0	0	1	1	0	0	1	0
0	1	0	1	0	1	0	0
0	1	1	1	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	1	1	1	1	1

Table 7: Rezultat simulacije RS pomnilne celice.



(a) Simulacija delovanja JK pomnilne celice po viru [?]

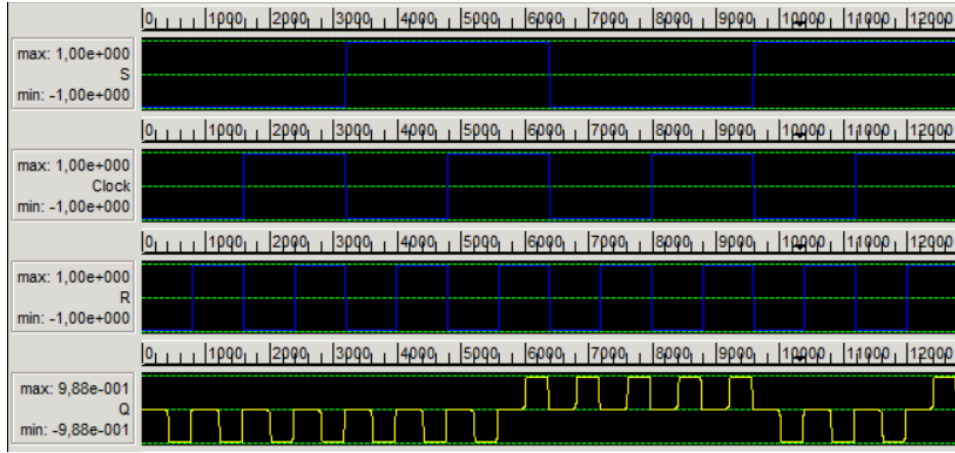


(b) Simulacija delovanja JK pomnilne celice po viru [?]

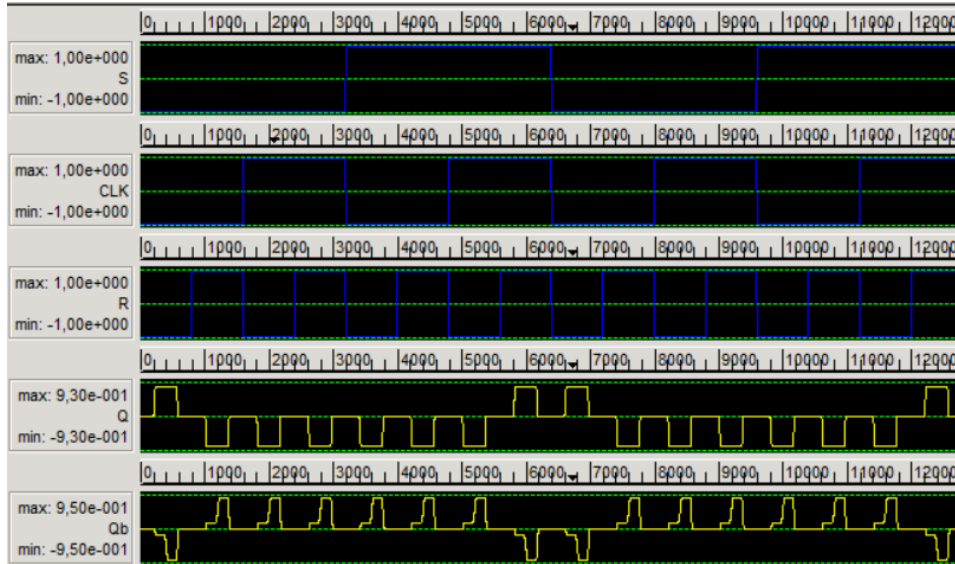
Slika 13: Simulaciji JK pomnilne celice

t	D^1q
0	q
1	\bar{q}

Table 8: Pravilnostna tabela T pomnilne celice.



(a) Simulacija delovanja RS pomnilne celice po viru [?]

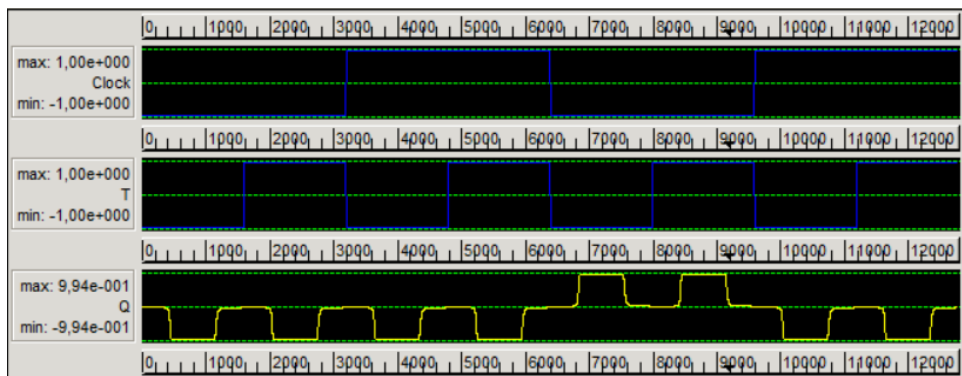


(b) Simulacija delovanja RS pomnilne celice po viru [?]

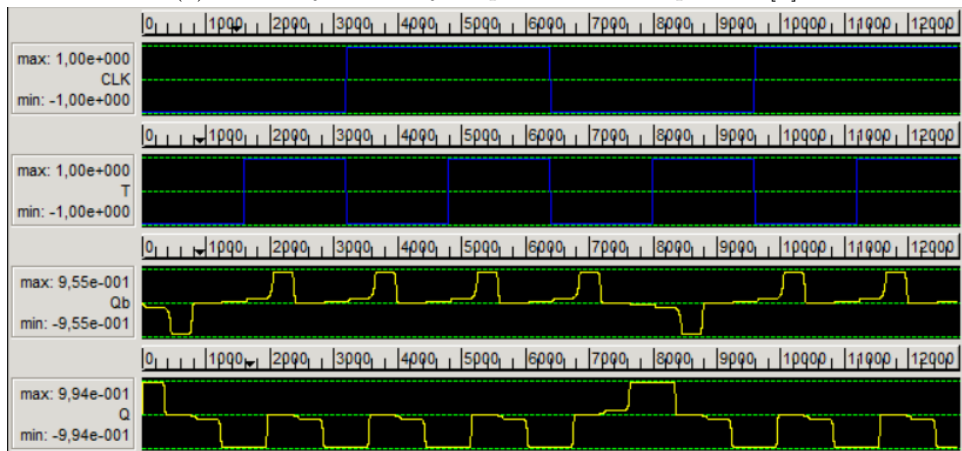
Slika 14: Simulaciji RS pomnilne celice

vir [?]			vir [?]		
<i>Clock</i>	<i>T</i>	<i>Q</i>	<i>Clock</i>	<i>T</i>	<i>Q</i>
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	0	1	1	0
0	0	1	0	0	0
0	1	1	0	1	1
1	0	0	1	0	0
1	1	0	1	1	0

Table 9: Rezultat simulacije T pomnilne celice.



(a) Simulacija delovanja T pomnilne celice po viru [?]



(b) Simulacija delovanja T pomnilne celice po viru [?]

Slika 15: Simulaciji T pomnilne celice