

Pomnenje s kvantnimi celičnimi avtomati

Jasmina Pegan, Blaž Rojc

November 16, 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 tunnel povezuje dve sosednji kvantni piki. Shema takšne celice je prikazana na sliki **TODO**.

TODO: slika QC tukaj

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 **TODO**.

TODO: slika dveh stabilnih stanj

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 preklonna funkcija.

Gradnjo kompleksnejših struktur pričnemo z osnovnimi gradniki. Osnovni gradniki predstavljajo poln funkcijski nabor, s katerim lahko implementiramo poljubno preklonno funkcijo. Najenostavnejši gradnik je vodilo, prikazano na sliki [TODO](#).

[TODO](#): slika vodila

Vodila preslikajo vhodno vrednost na izhod. Uporabljamo jih za povezovanje gradnikov v avtomatu. Drugi gradnik je negator, prikazan na sliki [TODO](#).

[TODO](#): slika negatorja

Negatorje lahko uporabimo tudi kot diagonalno vodilo, kjer je dolžina takega vodila sodo število. Primer take uporabe je prikazano na sliki [TODO](#).

[TODO](#): slika vodila iz negatorjev

Zadnji osnovni gradnik so večinska oz. majoritetna vrata, prikazana na sliki [TODO](#).

[TODO](#): slika majoritetnih vrat

Izhod majoritetnih vrat zavzame večinsko vrednost na treh vhodih. Z vezavo konstante na enega izmed vhodov majoritetnih vrat dobimo gradnik, ki predstavlja konjunkcijo ali disjunkcijo, kot je prikazano na sliki [TODO](#).

[TODO](#): tabela AND, OR iz majoritetnih

Tok podatkov v avtomatu usmerjamo s pomočjo urinih con. Vsaki celici dodelimo eno izmed štirih urinih con, ki se ciklično izmenjujejo. Tako omogočimo logično separacijo delov avtomata in način interakcije med njimi. Primer avtomata z ločenimi urinimi conami je prikazan na sliki [TODO](#).

[TODO](#): slika avtomata z različnimi urinimi conami

2.2 Problematika pomnenja

Ko je ura odsotna, je stanje kvantne celice v odsotnosti zunanjih sil nedeterministično. 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, 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.

2.3 Reverzibilno procesiranje (?)

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.

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).

Drugi vir [?] iz leta 2018 porabi manj prostora in celic ter ponudi tako izhodno stanje Q kot njegovo negacijo \bar{Q} , so pa zato urine cone bolj kompleksne. Po drugi strani ta vir uporablja manj stabilne negatorje kot prvi.

3.2 Reverzibilne implementacije (?)

3.3 Ideje in popravki (?)

4 Implementacija

V tem poglavju bomo primerjali predlagane implementacije pomnilnih celic.

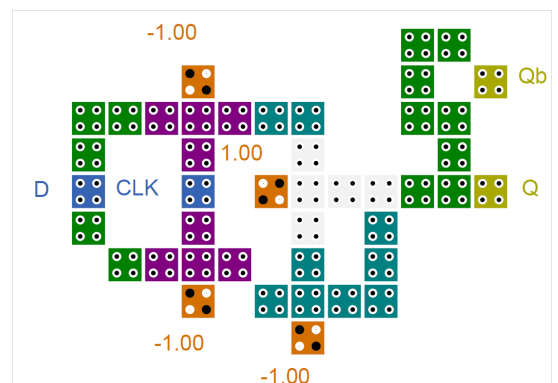
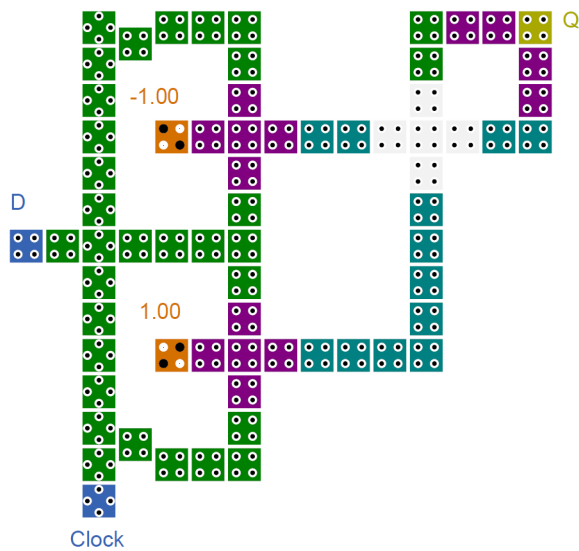
TODO: primerjaj implementacije: potratnost, izvedljivost, konsistentnost

TODO: vsem slikam dodati naslov in se sklicati v besedilu

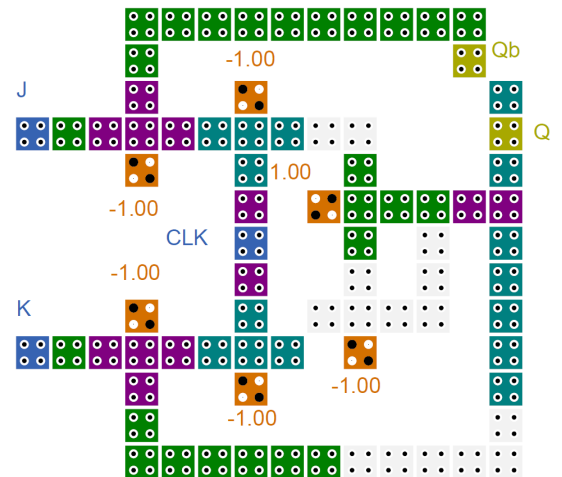
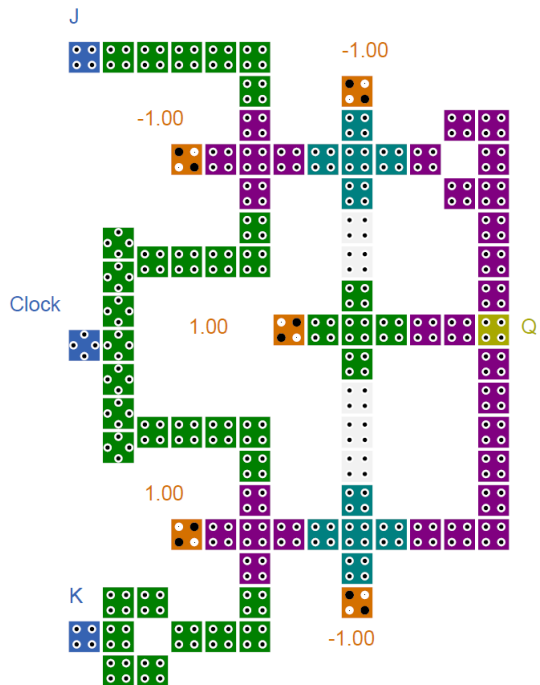
TODO: ali štejemo kot negator tudi določene povezave, ki niso standardna negatorja (navaden in dvojni), ampak opravljajo isto funkcijo?

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

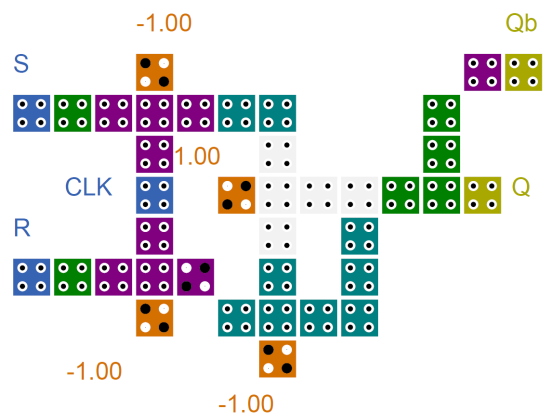
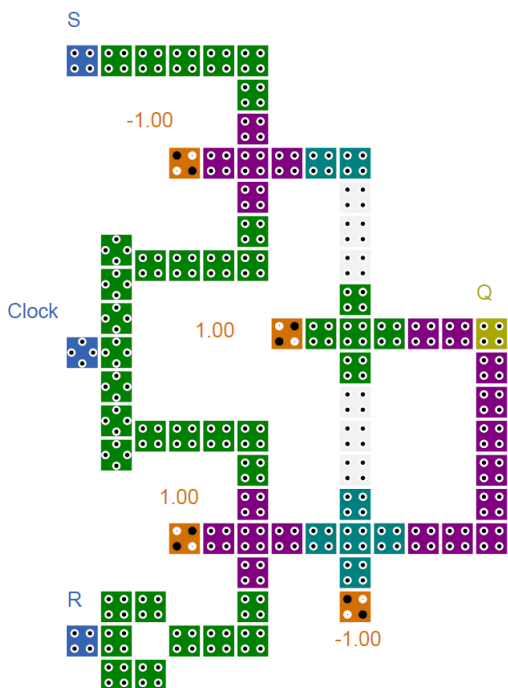
4.1 D pomnilna celica



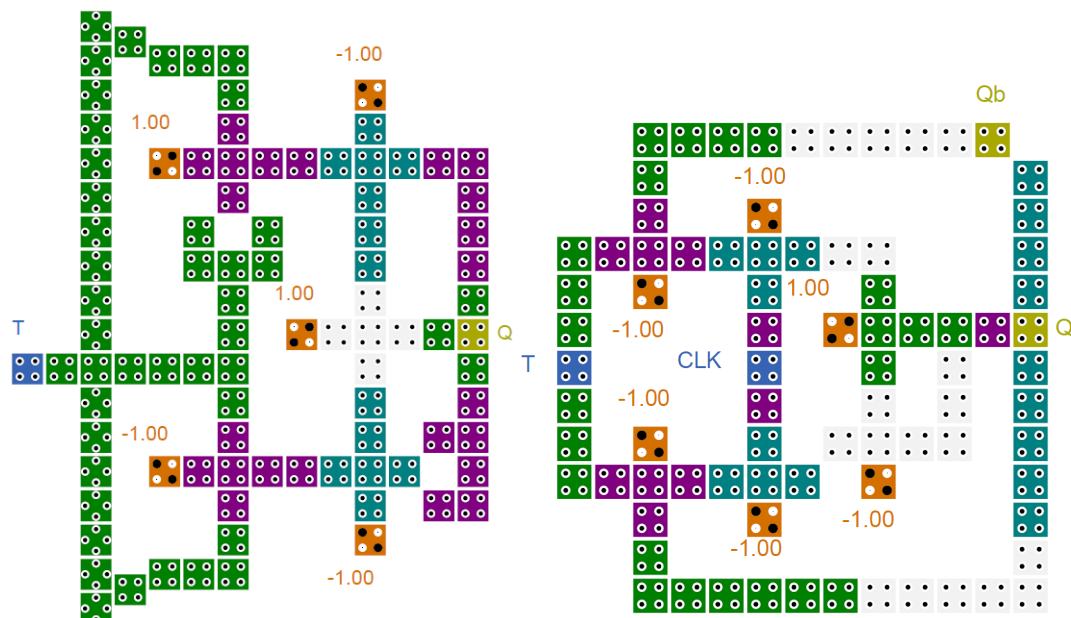
4.2 JK pomnilna celica



4.3 RS pomnilna celica

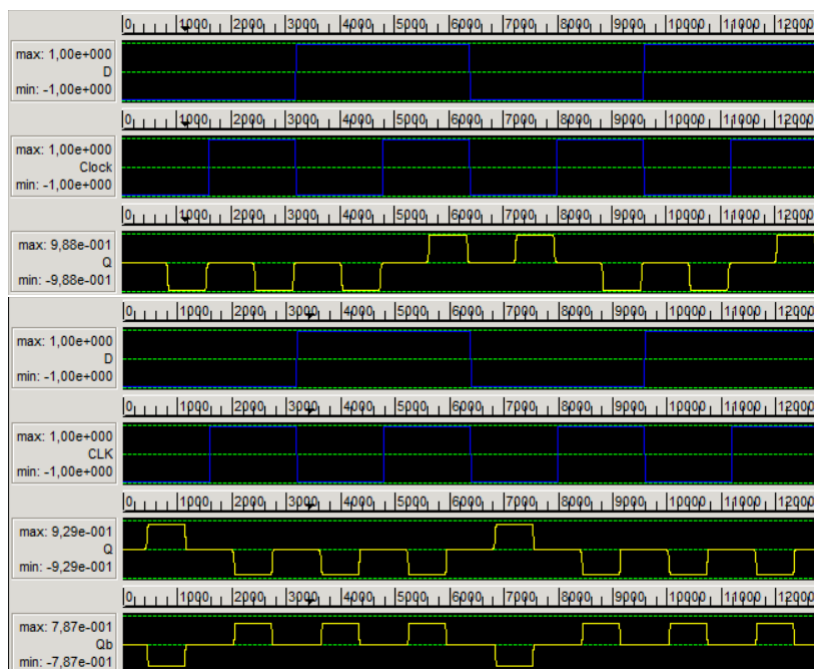


4.4 T pomnilna celica



4.5 Analiza delovanja

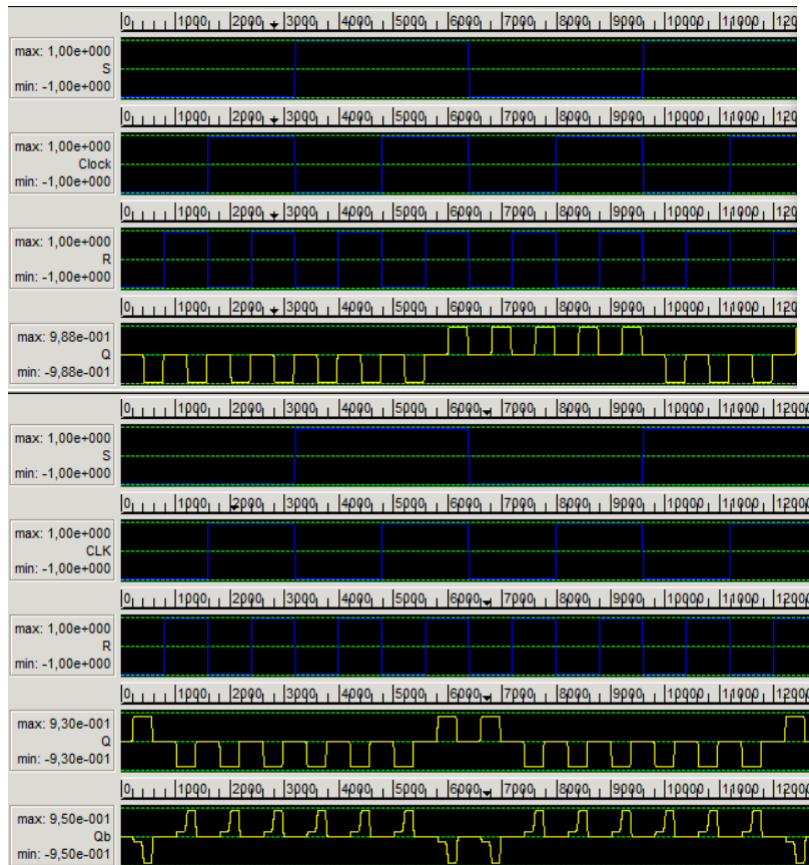
4.5.1 D pomnilna celica



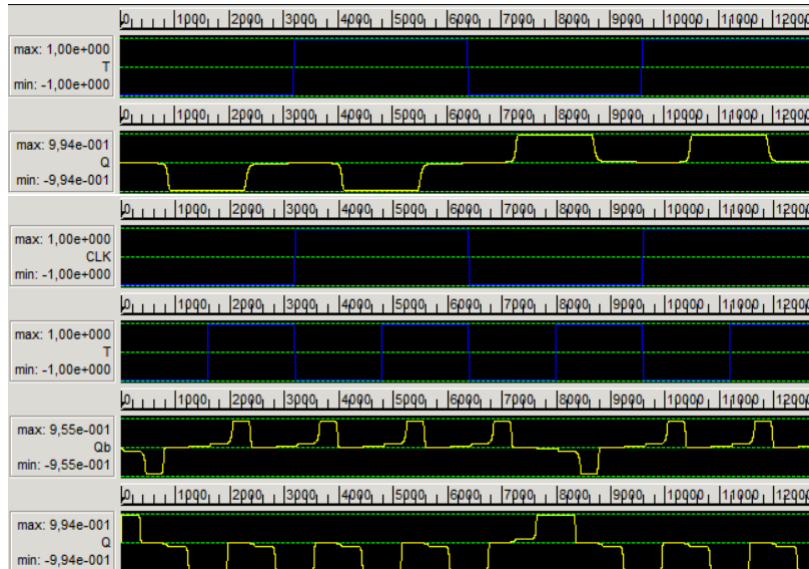
4.5.2 JK pomnilna celica



4.5.3 RS pomnilna celica



4.5.4 T pomnilna celica



5 Zaključek

TODO: kratek povzetek glavnih ugotovitev + smernice za nadaljnje delo