

Poročilo pri predmetu:  
**Računalniške strukture v nanotehnologijah,**  
nosilec predmeta: izr. prof. doc. Miha Mraz,  
maj, 2007

# Iskanje trojiških logičnih funkcij AND, OR in Implication v QCA strukturah

**Marjana Erdelji**

Univerza v Ljubljani, Fakulteta za Računalništvo in Informatiko,  
podiplomski magistrski študij

**Povzetek.** Strukture kvantnih celičnih avtomatov (ang. Quantum-dot Cellular Automata - QCA), katerih temelje je v začetku 80-ih let prejšnjega stoletja postavil C.S. Lent s sodelovci, predstavljajo alternativo za procesno platformo prihodnosti. Prehodu elektronov med kvantnimi točkami v 4-točkovni QCA celici pripišemo pomen logične fronte ali preklopa. Strukture QCA s preklopnimi lastnostmi soupadajo z lastnostmi polnega nabora logičnih operacij klasične dvojiške logike. Z razširitvijo 4-točkovne QCA celice v 6-točkovno EQCA (ang. Extended Quantum-dot Cellular Automata) celico iščemo EQCA strukture, s katero lahko realiziramo trivrednostne logične operacije AND, OR in Implication S.C. Kleena. Z uporabo programskega simulatorja qdCADE-v02.0b, razvitega v Laboratoriju za računalniške strukture in sisteme na Fakulteti za računalništvo in informatiko v Ljubljani, so bile ugotovljene nekatere možne EQCA strukture in podana usmeritev za prihodnjo razširitev raziskav.

## 1 Uvod

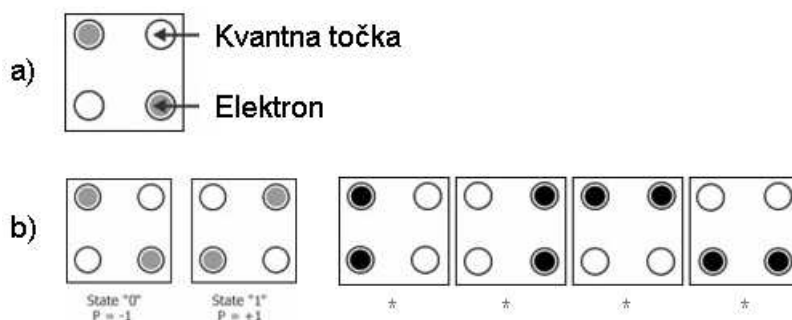
Strukture kvantnih celičnih avtomatov (ang. Quantum-dot Cellular Automata - QCA) so planarne strukture, sestavljene iz kvantnih celic, to je celice reda velikosti nanometra ( $10^{-9}$  m). Vsaka celica vsebuje štiri kvantne pike, med katerimi je mogoč tunelski učinek, ter dva elektrona. Posamezna celica lahko tako zaseda eno od dveh energetsko stabilnih stanj, ki sta pogojeni z lego elektronov. V zadnjih dvajsetih letih se je v strukturah QCA razvil poln dvojiški funkcijski nabor, ki obsega logične funkcije konjunkcije, disjunkcije in negacije, razvijejo pa se tudi orodja za postavljanje kompleksnejših preklopnih funkcij[1].

Z uporabo simulatorja qdCADE-v02.0b, ki z upoštevanjem fizikalnih lastnosti simulira delovanje struktur QCA, predlagam možne strukture razširjene trostanjske 6-točkovne kvantne celice, s katero bi realizirali logične funkcije AND, OR in Implication po S.C. Kleenu.

## 2 4-točkovna in 6-točkovna celica QCA

Celica QCA je po Lentu [2,3] kvadratne oblike, pri čemer se v vsakem vogalu nahaja kvantna pika, ki omogoča zadrževanje elektrona. Elektron se po definiciji v celici lahko nahaja le v eni od štirih pik. Med temi pikami lahko elektron zaradi tunelskega učinka prehaja. Razlog za prehajanje je v dejstvu, da elektron glede na prisotnost odbojnih sil iz svoje okolice teži k prehodu v piko, ki maksimizira njegovo stabilno lego. Povedano drugače, elektron prehaja v piko, v kateri so odbojne sile z okoljem in svojim sosedom v celici minimalne. Okoljske sile so odbojne sile med opazovanim elektronom in elektroni iz sosednjih celic ter sile med opazovanim elektronom in zunanji viri sil. Če v celico postavimo dva elektrona in odbojne sile med njima preglasijo vpliv sil iz okolice (kar je odvisno od oddaljenosti njihovega izvora), se elektrona postavita v eno od dveh diagonalnih leg, v katerih dosežeta maksimalno medsebojno oddaljenost. Z dvema elektronom v celici tako dobimo sistem z dvema stanjema.

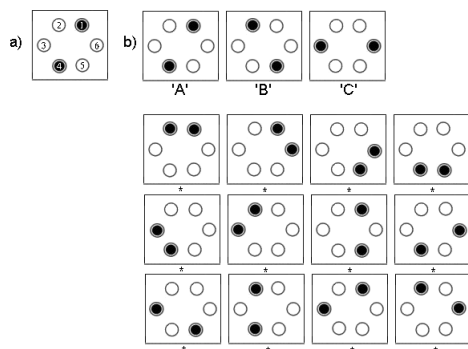
Na sliki 1 je predstavljena osnovna shema kvantne celice z dvema elektronom in štirimi pikami (a) in mogoče lege elektronov (b), pri čemer sta ob homogeni porazdelitvi celic in ustrezni razdalji med njimi legi elektronov označeni s  $P = -1$  in  $P = +1$  mogoči, lege elektronov označene z  $*$  pa nemogoče. Lent mogoči stanji  $P = -1$  in  $P = +1$  interpretira kot logično vrednost '0' in '1', ki smo jih vajeni iz dvojiške logike. Od tod lego elektronov interpretiramo kot *stanje* celice[1].



Slika 1: Celica QCA s štirimi kvadratnimi pikami in dvema elektronom (a) ter vse potencialne lege elektronov (b)

Slabost dvostanjske celice QCA je v omejenosti na dve stanji. Ob hipotezi, da je s tehnološkega vidika na površino celice mogoče postaviti več kot štiri pike lahko definiramo razširjeno celico QCA s šestimi pikami in dvema elektronom predstavljeno na sliki 2. Na sliki so z 'A', 'B' in 'C' označena mogoča stanja, z  $*$  pa stanja, ki jih ni mogoče doseči zaradi prevlade medsebojnih odbojnih sil med elektronom v celici. Celice v stanjih 'A', 'B' in 'C' lahko opišemo s sistemom treh logičnih vrednosti  $\{A, B, C\} = \{0, 1, \frac{1}{2}\}$ . Skupaj je je možnih stanj:

$$V(6, 2) = \frac{6!}{4! * 2!} = 15 \quad (1)$$



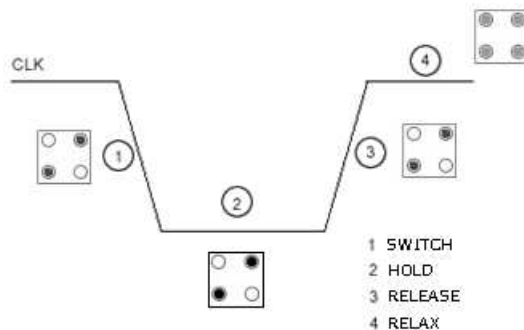
Slika 2: Celica EQCA z šestimi pikami in dvema elektronoma (a) in možne konfiguracije celice (b)

Vse zakonitosti izračunavanja stabilnega stanja celice so povzete po predhodno predstavljenem načinu izračunavanja QCA. Razširjena kvantna celica je povzeta kot osnovna entiteta razširjenih struktur QCA (ang. Extended Quantum-Dot Cellular Automata - EQCA)[1].

### 3 Procesiranje v QCA in EQCA

Za strukturo QCA Lent poimenuje poljubno planarno porazdelitev celic [2,3]. Z ustreznim 'zaklepanjem' posameznih celic, s čimer je onemogočeno prehanje elektronov, se po takšni strukturi doseže poljubna dinamika. Časovni signal v procesne gradnike ne vstopa neposredno, temveč kot regulirni signal. Slednji je realiziran v obliki električnega polja, ki s svojo (ne)prisotnostjo o(ne)mogoča tunelski učinek v obravnavanih celicah. Časovnost prevzame nase 4-fazni urni sistem (Switch, Hold, Release in Relax faza), ki implicitno uvaja princip cevovoda v kakršno koli strukturo. Na tak način kontroliramo pretok informacije med celicami strukture.

Slika 3 prikazuje QCA uro - njene časovne faze in njihov vpliv na energijsko bariero celice. Intenzivnost energijskega stanja celice je prikazana z sivimi odtenci, kjer najtemnejši odtенок pomeni največjo energijo.



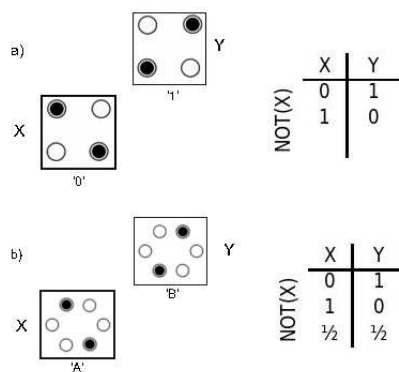
Slika 3: Ura QCA, njene časovne faze in vpliv na energijsko bariero celice

V strukturi QCA in EQCA poznamo tri tipe celic:

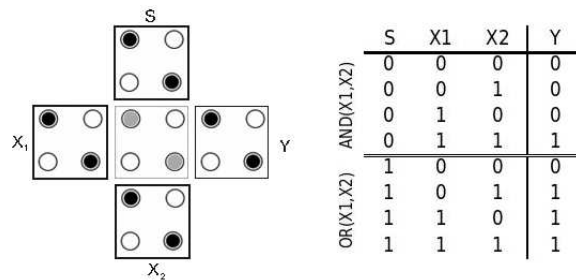
- *vhodne celice*: v njih se pod vplivom zunanjih sil prisilno doseže želena razporeditev elektronov (na nadaljnjih slikah so te celice predstavljene z odebeljenim robom)
- *notranje celice*: v njih se izvede poljubna logična 'obdelava' vhodnih podatkov (na nadaljnjih slikah so te celice predstavljene s sivim robom) in
- *izhodne celice*: njihovo stanje (s časovnim zamikom) obravnavamo kot rezultat procesiranja (na nadaljnjih slikah so te celice predstavljene z navadnim robom)

Strukture, katerih dinamika soupada s polnim naborom logičnih funkcij  $\{AND, OR, NOT\}$  ni zanimiva samo z vidika možnosti realizacije poljubne dvojiške preklopne funkcije, temveč tudi zato, ker funkciji AND in OR realizira v obliki dvojiške *večinske* funkcije (tako funkcijo imenujemo majoritetna vrata)[1] Treba je poudariti, da se pravila, ki veljajo v dvojiškem svetu, podrejo v trojiškem. Zato tudi ne iščemo polnega nabora trojiških funkcij!

Slika 4 prikazuje realizacijo negacije v QCA (a) in EQCA celicah, in slika 5 prikazuje realizacijo večinske funkcije v strukturah QCA. Pri tem so  $X$ ,  $X_1$  in  $X_2$  vhodne celice, preko katerih vstopajo vrednosti vhodnih spremenljivk, celica  $S$  je tudi vhodna celica, ki je 'fiksna' in določa pri logični vrednosti '0' logično funkcijo AND, ter pri logični vrednosti '1' logično funkcijo OR, in  $Y$  je izhodna celica.

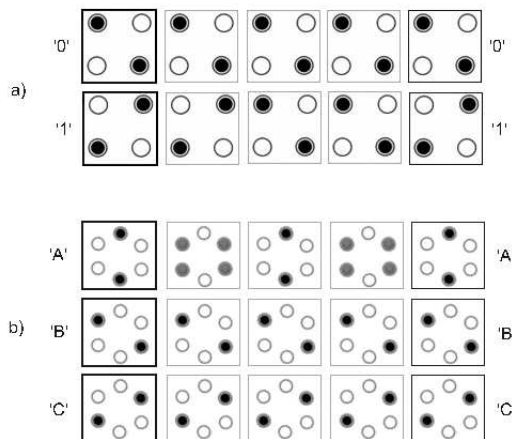


Slika 4: Realizacija dvojiške logične funkcije NOT (a) ter realizacija trojiške logične funkcije NOT za primer ko stanja 'A', 'B' in 'C' interpretiramo kot logično vrednost '0', '1' in  $\frac{1}{2}$  (logično stanje  $\frac{1}{2}$  se ohranja).



Slika 5: Realizacija dvojiških logičnih funkcij AND in OR. Logična vrednost '0' celice S določa logično funkcijo  $AND(X_1, X_2)$ , logična vrednost '1' celice S pa določa logično funkcijo  $OR(X_1, X_2)$ .

Poleg struktur za implementacijo polnega nabora logičnih funkcij potrebujemo strukture QCA za prenos podatkov, ki jih imenujemo *žica* ali *linija* (ang. *QCA wire*). Linija predstavlja zaporedje celic QCA druga ob drugi. Če je vhodna celica polarizirana, potem se ostale celice verižno sinhronizirajo na njeno polarizacijo zaradi Coulombovih interakcij med njimi. Na sliki 6 je prikazan prenos podatkov po QCA (a) in EQCA (b) linijah.



Slika 6: Prenos binarnih logičnih vrednosti '0' in '1' po liniji QCA (a) ter prenos vrednosti 'A', 'B' in 'C' po liniji EQCA (b). Celice EQCA so rotirane za  $30^\circ$ .

Pri prenosu po liniji QCA in EQCA imamo tri možnosti:

- stanje se pri prenašanju ohranja
- stanje se pri prenašanju ne ohranja
- stanje alternira

Alternirajoče prenašanje stanja lahko opazimo na sliki 6(b), pri prenosu vrednosti 'A'. Za pravilni prenos vrednosti 'A' bi v implementacijskem pogledu

morali pristati na liha števila notranjih celic v strukturi linije EQCA.

## 4 Cilj in obseg simulacije

Cilj naloge je s simulatorjem qdCADE-v02.0b poiskati EQCA strukture s tremi vhodnimi celicami ( $X_1, X_2$  in  $S$ ), eno delovno celico ( $D$ ) in izhodno celico ( $Y$ ), ob različnih kombinacijah postavitve celic v obsegu polja  $3 \times 3$  celic, ki bi realizirale dvovhodno logično funkcijo  $\text{AND}(X_1, X_2)$ ,  $\text{OR}(X_1, X_2)$  in  $\text{Implication}(X_1, X_2)$ , kjer vhod  $S$  predstavlja izbiro logične funkcije.

Pri tem je bilo potrebno dopolniti simulator v naslednjih smereh:

- definiranje možnih pravilnostnih tabel za različne vrednosti stanj 'A', 'B' in 'C'. S tem se izognemo 'zaklepanju' vrednosti stanj  $\{A, B, C\}$ . Možnih je naslednjih 6 konfiguracij:

	'A'	'B'	'C'
1	0	$\frac{1}{2}$	1
2	0	1	$\frac{1}{2}$
3	$\frac{1}{2}$	0	1
4	$\frac{1}{2}$	1	0
5	1	0	$\frac{1}{2}$
6	1	$\frac{1}{2}$	0

Tabela 1: Možne konfiguracije vrednosti stanj  $\{A, B, C\}$  iz množice trojiških vrednosti  $\{0, \frac{1}{2}, 1\}$

- določitev pravilnostne tabele vrednosti logičnih funkcij AND, OR in Implication nad vsemi možnimi konfiguracijami vrednosti  $\{A, B, C\}$ . Slika 7 prikazuje pravilnostno tabelo logičnih funkcij AND, OR in Implication (a) ter NOT (b) po S.C. Kleenu:

a)						b)	
S	X <sub>1</sub>	X <sub>2</sub>	AND(X <sub>1</sub> , X <sub>2</sub> )	OR(X <sub>1</sub> , X <sub>2</sub> )	Implication(X <sub>1</sub> , X <sub>2</sub> )	X	NOT(X)
S = 0	0	0	0	0	0	0	1
	0	0	1/2	0	1/2	1/2	1/2
	0	0	1	0	1	1	0
	0	1/2	0	0	1/2	0	1
	0	1/2	1/2	0	1/2	1/2	1/2
	0	1/2	1	0	1	1	0
	0	1	0	0	1	0	1
	0	1	1/2	0	1	1/2	1/2
	0	1	1	0	1	1	0
	1/2	0	0	0	1/2	1/2	1/2
	1/2	0	1/2	0	1/2	1/2	1/2
	1/2	0	1	0	1	1	0
S = 1/2	1/2	1/2	0	1/2	1/2	1/2	1/2
	1/2	1/2	1/2	1/2	1/2	1/2	1/2
	1/2	1/2	1	1/2	1	1	0
	1/2	1	0	0	1	0	1
	1/2	1	1/2	1/2	1	1/2	1/2
	1/2	1	1	1/2	1	1	0
	1	0	0	0	1	1	0
	1	0	1/2	0	1	1/2	1/2
	1	0	1	0	1	1	0
	1	1/2	0	0	1	1/2	1/2
	1	1/2	1/2	1/2	1	1/2	1/2
	1	1/2	1	1/2	1	1	0
1	1	0	0	1	0	1	
1	1	1/2	1/2	1	1/2	1/2	
1	1	1	1	1	1	0	

Slika 7: Pravilnostna tabela logičnih funkcij AND, OR in Implication (a) ter NOT (b) po S.C. Kleenu

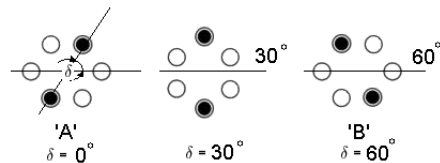
• realizacija funkcije ujemanja, s pomočjo katere ugotavljamo, kako blizu vrednosti iskanih logičnih funkcij je dobljeni rezultat simulacije. Naslednja tabela prikazuje možne razdalje med stanji  $S_1$  in  $S_2$ , kjer je  $\{S_1, S_2\} \in \{ 'A', 'B', 'C' \}$ .

$S_1$	$S_2$	$rd(S_1, S_2)$
'A'	'A'	0
'A'	'B'	1
'A'	'C'	2
'B'	'A'	1
'B'	'B'	0
'B'	'C'	1
'C'	'A'	2
'C'	'B'	1
'C'	'C'	0

Tabela 2: Razdalje med posameznimi stanji  $S_1$  in  $S_2$

Funkcija ujemanja, oziroma minimalna razdalja se izračuna tako, da seštejemo vse razdalje med zelenim izhodom in izhodom dobljenih s simulacijo, za posamezne vrednosti  $S \in \{0, 1, \frac{1}{2}\}$  (glej slika 7).

• upoštevanje vseh možnih kotov celic, ki so definirane s prenosom linije. Možne so vse kombinacije rotacij vhodnih celic od  $0^\circ$  do  $360^\circ$ . Pri tem se lahko omejimo na vrednosti kotov od  $0^\circ$  do  $50^\circ$ . Pri rotaciji za kot  $60^\circ$  se namreč, kot prikazuje slika 8, stanje 'A' zgolj preslika v stanje 'B'. Ker ne 'zaklepamo' stanj, so te vrednosti zaobljete s pravilnostno tabelo (glej slika 7). Pri rotaciji za kot  $120^\circ$  se zaradi simetrije celice stanje 'A' preslika v stanje 'A', preslikano čez vertikalno os. Podobno velja za rotacije za kot od  $180^\circ$  do  $360^\circ$ .

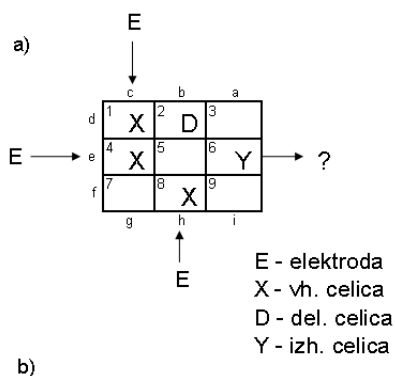


Slika 7: Rotacija celice s stanjem 'A' preko horizontalne osi za kot  $\delta$ . Stanje 'A' se pri rotaciji za kot  $\delta = 60^\circ$  preslika v stanje 'B'.

• definiranje možnih postavitvev celic, ki določajo strukturo. Omejitev je postavitev treh vhodnih celic X, delovne celice D in izhodne celice Y, po polju  $3 \times 3$ , kot je razvidno s slike 8. Vseh možnih postavitvev celic po polju je:

$$V(9, 5, 2) = \frac{9!}{4! * 5! * 3!} = 21 \quad (2)$$

Zaradi dolgotrajnega interpretorskega izvajanja simulacije, ki teče v okolju Matlab, sem se omejila na 10 postavitvev, ki so navedene na sliki 8(b).



b)

	E	X	D	Y
1	d, e, f	1, 4, 7	5	6
2	d, e, f	1, 4, 7	5	3
3	d, e, f	1, 4, 7	5	9
4	b, e, f	2, 4, 7	5	6
5	b, d, e	1, 2, 4	5	6
6	b, d, e	1, 2, 4	5	9
7	b, d, e	1, 2, 4	5	3
8	a, c, f	1, 3, 7	5	6
9	a, c, f	1, 3, 7	5	9
10	b, e, h	2, 4, 8	5	6

Slika 8: Polje postavitvev treh vhodnih celic, označenih z X, delovne celice D in izhodne celice Y (a). Nad vsemi možnimi postavitvami je bilo izbranih 10 postavitvev (b).



## 5 Rezultati

Z uporabo simulatorja *qdCADE\_v02.0b*, ki simulira delovanje podanih različnih struktur, za katerih izhode ugotavljamo ujemanje z funkcijskimi vrednostmi logičnih funkcij AND, OR in Implication, ob uporabi urnega proženja celic (glej sliko 3) in upoštevanju le tistih izhodnih stanj z verjetnostjo več kot 80%. Dobljeni so bili naslednji rezultati:

- pri simulaciji prenašanja stanj 'A', 'B' in 'C' po liniji, kjer se celice rotirajo za kot od  $0^\circ$  do  $50^\circ$  s korakom  $10^\circ$ , se vrednosti stanj 'A' in 'B' z rotacijo celice za kot  $0^\circ$  ne prenašajo funkcionalno pravilno, pri ostalih rotacijah se nekaj stanj prenaša alternirajoče. Tako lahko omejimo območje na interval od  $10^\circ$  do  $50^\circ$  in privzamemo, da imamo liho število notranjih celic linije, ki vstopajo v strukturo.

- pri simulaciji majoritetnih vrat (postavitev vhodnih celic 2-4-8, glej slika 5 in 8(a)) z rotacijami vhodnih celic za kote od  $0^\circ$  do  $180^\circ$ , kjer so vhodi rotirani za **isti kot**, je bila med rezultati simulacije in željenimi vrednostmi pravilnostne tabele (glej sliko 7) logičnih funkcij dobljena minimalna razdalja 4. Za realizacijo zelenih logičnih funkcij pa želimo minimalno razdaljo 0.

- pri simulaciji majoritetnih vrat (postavitev vhodnih celic 2-4-8, glej slika 5 in 8(a)) z rotacijami vhodnih celic za kote od  $0^\circ$  do  $180^\circ$ , kjer so vhodi rotirani za **različni kot**, je bila med rezultati simulacije in željenimi vrednostmi pravilnostne tabele (glej sliko 7) logičnih funkcij dobljena minimalna razdalja 1.

permutacija	S	$X_1$	$X_2$	$E_1$	$E_2$	$E_3$	D	Y
1	$\delta$	$\delta$	$0^\circ$	$\delta$	$\delta$	$0^\circ$	$0^\circ$	$0^\circ$
2	$0^\circ$	$0^\circ$	$\delta$	$\delta$	$\delta$	$\delta$	$0^\circ$	$0^\circ$
3	$0^\circ$	$\delta$	$\delta$	$0^\circ$	$\delta$	$\delta$	$0^\circ$	$0^\circ$
4	$\delta$	$0^\circ$	$0^\circ$	$\delta$	$0^\circ$	$0^\circ$	$0^\circ$	$0^\circ$
5	$0^\circ$	$\delta$	$0^\circ$	$0^\circ$	$\delta$	$0^\circ$	$0^\circ$	$0^\circ$
6	$0^\circ$	$0^\circ$	$\delta$	$0^\circ$	$0^\circ$	$\delta$	$0^\circ$	$0^\circ$

Tabela 3: Izbrane permutacije rotacij vhodnih celic S,  $X_1$ ,  $X_2$ , elektrod  $E_1$ ,  $E_2$ ,  $E_3$ , delovne celice D in izhodne celice Y za kot  $\delta$  (glej slika 8).

Takoj lahko sklepamo, da rotacije celic strukture pripomorejo k približevanju danega cilja.

- pri simulaciji različnih struktur, to je nad obsegom **različnih postavitvev** celic po polju  $3 \times 3$  (glej sliko 8(b)), z rotacijami vhodnih celic za kote od  $0^\circ$  do  $50^\circ$ , kjer so vhodi rotirani za **različni kot** (glej tabela 3), je bila med rezultati simulacije in željenimi vrednostmi pravilnostne tabele (glej sliko 7) logičnih funkcij dobljena minimalna razdalja 0.

$kot0^\circ$	$kot10^\circ$	$kot20^\circ$	$kot30^\circ$	$kot40^\circ$	$kot50^\circ$
1	1	0	0	1	1

Tabela 4: Minimalne vrednosti funkcije ujemanja med dobljenimi in željenimi vrednostmi nad izbranimi postavitvami (glej sliko 8(b)). Opomba: Rezultatov

za  $kot0^o$  ne upoštevamo zaradi problema nepravilnega prenosa stanj v strukturo.

[kot, perm., post.] <sup>1</sup>	konfig. <sup>2</sup>	S <sup>3</sup>	Impl.(X <sub>1</sub> ,X <sub>2</sub> )	AND(X <sub>1</sub> ,X <sub>2</sub> )	OR(X <sub>1</sub> ,X <sub>2</sub> )
[20, 1, 2]	1	'C'		✓	
[20, 1, 2]	3	'C'			✓
[20, 1, 2]	4	'C'		✓	
[20, 1, 2]	6	'C'		✓	
[30, 1, 2]	1	'C'			✓
[30, 2, 6]	1	'C'			✓
[30, 3, 2]	1	'C'			✓
[30, 3, 5]	1	'C'			✓
[30, 3, 6]	1	'C'			✓
[30, 5, 2]	1	'C'			✓
[30, 6, 6]	1	'C'			✓
[30, 1, 2]	3	'C'			✓
[30, 2, 6]	3	'C'			✓
[30, 3, 5]	3	'C'			✓
[30, 3, 6]	3	'C'			✓
[30, 5, 2]	3	'C'			✓
[30, 6, 6]	3	'C'			✓
[30, 1, 2]	4	'C'		✓	
[30, 2, 6]	4	'C'		✓	
[30, 3, 2]	4	'C'		✓	
[30, 3, 5]	4	'C'		✓	
[30, 3, 6]	4	'C'		✓	
[30, 5, 2]	4	'C'		✓	
[30, 6, 6]	4	'C'		✓	
[30, 1, 2]	6	'C'		✓	
[30, 2, 6]	6	'C'		✓	
[30, 3, 2]	6	'C'		✓	
[30, 3, 5]	6	'C'		✓	
[30, 3, 6]	6	'C'		✓	
[30, 5, 2]	6	'C'		✓	
[30, 6, 6]	6	'C'		✓	

1: Stolpec [kot, permutacija, postavitev] označuje strukturo z velikostjo kota rotacije celice (glej sliko 7), zaporedno številko permutacije kotov (glej tabelo 3) in zaporedno številko postavitve celic v strukturo (glej sliko 8).

2: Stolpec konfiguracije označuje zaporedno številko konfiguracije trojiških vrednosti za stanja {A, B, C} (glej tabelo 1).

3: Stolpec S označuje celico S (glej sliko 7), s katero želimo izbirati logično funkcijo Implementation, AND in OR dane strukture.

Tabela 5: Rezultati simulacije iskanja struktur za realizacijo logičnih funkcij Implication, AND in OR nad danim obsegom.

S tabele 5 je razvidno, da realizacija logične funkcije Implication nad danim obsegom, ki ima minimalno razdaljo 0, ne obstaja. Prav tako vidimo, da z

vhodno celico S ne bo mogoče izbirati logično funkcijo, sej ima v tem primeru isto stanje tako za OR kot AND. Na tabeli 6 vidimo možne strukture za realizacijo Implication funkcije, z minimalno razdaljo 1, ki bi jih bilo potrebno nadgraditi.

[kot, perm., post.] <sup>1</sup>	konfig. <sup>2</sup>	S <sup>3</sup>
[0, 1, 9]	1	'B'
[0, 2, 9]	1	'B'
[0, 4, 9]	1	'B'
[0, 5, 9]	1	'B'
[0, 6, 9]	1	'B'
[10, 4, 9]	1	'B'
[40, 4, 9]	1	'B'
[40, 1, 7]	3	'A'
[40, 2, 9]	5	'C'
[50, 1, 7]	3	'A'

1: Stolpec [kot, permutacija, postavitev] označuje strukturo z velikostjo kota rotacije celice (glej sliko 7), zaporedno številko permutacije kotov (glej tabelo 3) in zaporedno številko postavitve celic v strukturo (glej sliko 8).

2: Stolpec konfiguracije označuje zaporedno številko konfiguracije trojiških vrednosti za stanja  $\{A, B, C\}$  (glej tabelo 1).

3: Stolpec S označuje celico S (glej sliko 7), s katero želimo izbirati logično funkcijo AND, OR in Implication dane strukture.

Tabela 6: Rezultati simulacije iskanja struktur za realizacijo logične funkcije Implication, z minimalno razdaljo 1, nad danim obsegom.

Iz tabele 7 je razvidno, pri kateri vhodni kombinaciji stanj izhodno stanje  $Y_s$ , dobljenega s simulacijo, odstopa od želenega rezultata Y (podčrtana vrstica):

S	$X_1$	$X_2$	S	$X_1$	$X_2$	$Y_s$	Y
$\frac{1}{2}$	0	0	C	A	A	A	0
$\frac{1}{2}$	0	$\frac{1}{2}$	C	A	C	C	$\frac{1}{2}$
$\frac{1}{2}$	0	1	C	A	B	C	1
$\frac{1}{2}$	$\frac{1}{2}$	0	C	C	A	A	0
$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	C	C	C	C	$\frac{1}{2}$
$\frac{1}{2}$	$\frac{1}{2}$	1	C	C	B	C	$\frac{1}{2}$
$\frac{1}{2}$	1	0	C	B	A	A	0
$\frac{1}{2}$	1	$\frac{1}{2}$	C	B	B	C	$\frac{1}{2}$
$\frac{1}{2}$	1	1	C	B	C	C	$\frac{1}{2}$

Tabela 7: Pravilnostna tabela simulirane in pravilne vrednosti logične funkcije Implication

Nadgradnja strukture, z sestavljanjem struktur v smislu  $\text{Implication}(X_1, X_2) \otimes f(X_1, X_2)$ , s katero bi dosegli pravilno vrednost logične funkcije  $\text{Implication}(X_1, X_2)$  na enostaven in učinkovit način ni bilo mogoče doseči.

## 6 Sklep

Cilj naloge, poiskati strukturo EQCA (ang. Extended Quantum-dot Cellular Automata), ki bi nad danim obsegom simulacije realizirala dvovhodno logično funkcijo  $\text{AND}(X_1, X_2)$ ,  $\text{OR}(X_1, X_2)$  in  $\text{Implication}(X_1, X_2)$ , kjer stanje vhodne celice  $S$  predstavlja izbiro logične funkcije, ni bil dosežen.

Na izhodno stanje strukture lahko predvsem vplivamo z različno rotacijo in postavitvijo celic. V rezultatih so bile najdene možne strukture za realizacijo AND in OR (glej tabelo 5). V vseh primerih je rešitev podana z vrednostjo vhoda  $S = 'C'$ , kar pomeni, da lahko logično funkcijo AND ali OR določimo le z izbiro rotacij celic, s permutacijo rotacij, s postavitvijo in s konfiguracijo vrednosti za stanja A, B in C. Ko sestavljamo različne strukture med seboj, je priporočljivo in bolj enostavno, da so vse strukture iste konfiguracije.

Z dodajanjem struktur k strukturi za funkcijo Implementation minimalne razdalje 1 (tabela 6) ni bilo najti neke enostavne rešitve, ki bi popravila izhodno stanje na točno vrednost.

Zato za prihodnje delo predlagam predvsem razširitev raziskanega obsega možnosti:

- uporaba preostalih možnih permutacij nad vsemi celicami (glej tabela 3)
- uporaba preostalih možnih postavitvev po polju (glej slika 8(b))
- povečevanja števila vhodnih celic, pri tej hitrosti delovanja simulatorja ne priporočam (delna simulacija enega kota  $[0,1:6,1:10]$  nad tremi vhodi, je tekla okoli dve uri. Za štiri vhode bi simulacija tekla 3-krat več). Priporočam pa povečanje polja za postavitev celic, vendar tu ne bi pričakovala nekih bistvenih izboljšav, zaradi oddaljenosti med celicami.

Smisel iskanja struktur za logične funkcije naj bi bil v enostavnosti rešitve, tako kot je realiziran enostaven primer z uporabo večinske funkcije pri celicah QCA. Možno je, da enostavna rešitev ne obstaja. Predvsem bi bilo potrebno preiskati vse možnosti, pohitriti izvanjanje simulatorja s prehodom na novo platformo in najti enostavnejši način pregledovanja in interpretacije rezultatov simulacije.

## 7 Literatura

- [1] I. Lebar Bajec. M. Mraz, **Večstanjsko procesiranje v strukturah kvantnih celičnih avtomatov**, Elektrotehniški vestnik XX(Y): 1-6, YEAR, Slovenija, November 1997.
- [2] S.C. Lent. P.D. Tougaw. W. Porod. G.H. Bernstein, **Quantum cellular automata**, *Nanotechnology*, Vol. 4, pp. 49-57, 1993.
- [3] C.S. Lent. P.D. Tougaw, **Lines of interacting quantum-dot cells: A binary wire**, *Journal of Applied Physics*, Vol. 74, pp. 6227-6233, 1993.
- [4] Klir G. Bo Yuan, **Fuzzy Sets and Fuzzy Logic**, pp. 217-220 ,1995.