

# Realizacija 4-bitnega števca na programabilnih rešetkah QCA

Jan Berčič<sup>1</sup>, Žan Palčič<sup>1</sup>, Teja Roštan<sup>1</sup>, and Jasna Urbančič<sup>1</sup>

<sup>1</sup>Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Ljubljana, Slovenija

## POVZETEK

Programabilne rešetke kvantnih celularnih avtomatov so ena izmed mnogih arhitektur za realizacijo kompleksnejših struktur v vezjih QCA. Arhitektura se poskuša približati delovanju vezja FPGA in tako predstavi polje kvantnih celic kot mrežo, ki ima na levi strani vhodne celice, na desni strani izhodne celice, navpično pa vodi linije celic, ki s fiksnimi vrednostmi nastavijo pravilno delovanje vezja. Na križišču navpičnih in vodoravnih linij so realizirana logična vrata AND, OR in negator. Ker je to poln funkcijski nabor, lahko realiziramo poljubno strukturo. Rešetke same po sebi postavljajo dodatne pogoje pri implementaciji vezij, ki v nekaterih primerih izpeljavo malce otežijo, v drugih pa zahtevajo povsem drugačen pristop — pri implementaciji števca in vrat XOR smo naleteli na obe situaciji. Pokazali smo, da s to arhitekturo pri implementaciji števca dobimo dobre rezultate, ki so primerljivi z rezultati v literaturi.

## Uvod

Eden izmed ključnih problemov kvantnih celičnih avtomatov (QCA) je neželena električna interakcija med celicami, ki nam predstavlja veliko oviro pri križanju žic. Sen *et al.*<sup>1</sup> so primerjali koplanarno križanje, ki uporablja dve različni rotaciji celic, z večnivojskim QCA. Križanje v več nivojih eliminira zahtevo po rotaciji celic in zmanjša presluhe. Na primeru nekaterih znanih vezij so pokazali, da je večnivojski QCA v primerjavi z enonivojskim bolj kompakten (uporabi manjše število celic) in ima v vezju manj urinih con. Po drugi strani so Sen *et al.*<sup>2</sup> z namenom zmanjšanja vpliva konstrukcijskih omejitev in napak pri zasnovi predlagali hibridni pristop z dvema različnima rotacijama celic. Na značilnih vezjih, kot sta majoritetna vrata in multiplekser, so pokazali, da lahko dosežejo več kot 97% odpornost na napake, kar pa posledično terja večje število QCA celic. Drugačen pristop k reševanju tega problema so uporabili Tougaw *et al.*<sup>3</sup>, ki so v ta namen uporabili porazdeljeno signalno omrežje. Slednje dovoli, da množica vhodnih signalov vzpostavi poljubne povezave z izhodnimi signali. Omrežje so prav tako zaradi boljše kompaktnosti povečali na dva nivoja.

Medtem so se Ahmad *et al.*<sup>4</sup> osredotočili na zmanjšanje energijske izgube z optimizacijo polnega seštevalnika. Predlagali so zmogljiva tri-vhodna vrata XOR, ki imajo natančno določene interakcije med celicami za dobro delovanje. Predlagana vrata XOR so pokazala veliko izboljšanje glede latence, velikosti območja in števila uporabljenih celic. Problem različnih števil celic znotraj časovnih con in problem pri zagotavljanju povratnih zank so reševali Campos *et al.*<sup>5</sup>, ki so predstavili urino shemo s porazdelitvijo urinih faz med različne časovne cone, znotraj katerih so kvantne celice z isto urino fazo. Celice so si s sosednjimi fazami blizu in prenos informacije med posameznimi ploščicami je natančno določen. Opisana arhitektura omogoča tudi povratne poti poljubnih dolžin in s tem učinkovitejšo realizacijo struktur.

Realizacije kompleksnejših struktur v vezju so se med mnogimi lotili tudi Kalogiton *et al.*<sup>6</sup> z arhitekturo programabilnih rešetk na vezju QCA. Logična vrata so predstavili kot križišča linij kvantnih celic, s katerimi je mogoče implementirati poln funkcijski nabor (AND, OR in negator). Rešetke omogočajo hitrejše računanje, zavzamejo manj površine in imajo nizko časovno zakasnitev. Posledično ne omogočajo vejitev, pri načrtovanju pa je potrebno upoštevati določena priporočila in omejitve za nastavitve ure na posameznih celicah. Kljub temu so prinesle zelo dobre rezultate pri različnih strukturah, posebej pri implementaciji 1-bitnega polnega seštevalnika, ki je ena izmed najbolj pomembnih komponent ALU enote. Arhitektura se poskuša približati vezju FPGA. To je integrirano vezje, ki je zgrajeno iz osnovnih nastavljivih logičnih enot, te pa so povezane v matriko. Prav tako, kot predlaga arhitektura programabilnih rešetk, so na vezju FPGA vhodi definirani na eni strani in izhodi na drugi. Matrična razporeditev celic QCA in dobro definiran postopek za realizacijo poljubnih struktur nam tako omogočata lažji pristop k realizaciji kompleksnejših struktur in prav to je eden ključnih razlogov, da smo se zanj odločili.

## Implementacija

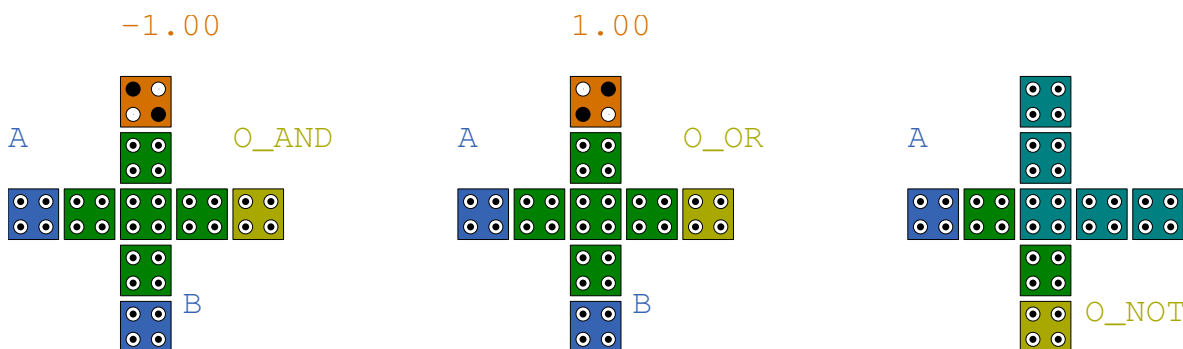
Za cilj smo si zastavili implementacijo stabilnega več-bitnega števca, oziroma natančnejše, 4-bitnega števca na arhitekturi programabilnih rešetk kvantnih celularnih avtomatov. Kot smo omenili, arhitektura postavlja precej omejitev, zato smo se

za začetni korak odločili implementirati manjšo strukturo. Izbrali smo vrata XOR, ker so dovolj enostavna, da jih lahko vzamemo za testno strukturo in dovolj kompleksna, da smo se lahko skozi njihovo programiranje srečali z večino ovir, ki jih z upoštevanjem priporočil in omejitev za programabilne rešetke lahko dobimo pri bolj kompleksnih vezjih.

Ko smo uspešno sestavili vrata XOR, smo se lotili števca. Programiranje števca je možno na dva načina, preko klasičnega flip-flop sistema z JK pomnilno celico ali z uporabo lastnosti samih kvantnih celularnih avtomatov, tj. s pomočjo zank (angl. *loops*).

### Osnovni elementi

Množica z vrati AND in OR ter negatorjem predstavlja poln funkcijski nabor. Vrata AND, na sliki 1 (levo), in vrata OR, na sliki 1 (sredina), so sestavljena iz majoritetnih vrat, ki so dobro prepoznaven model kvantnih celularnih avtomatov. Imajo obliko križa s tremi vhodi in enim izhodom. Eden od treh vhodov predstavlja programsko fiksirano vrednost, ki določi, ali bosta ostala dva vhoda proizvedla izhod AND logične vrednosti ali OR logične vrednosti. Zato, ker tudi negator spada v arhitekturo programabilnih rešetk, so Kalogiton *et al.*<sup>6</sup> poiskali model negatorja, na sliki 1 (desno), ki ravno tako ustreza obliki križa.



Slika 1. Sheme osnovnih vrat v obliki, ki je primerna za rešetko.

### Vrata XOR

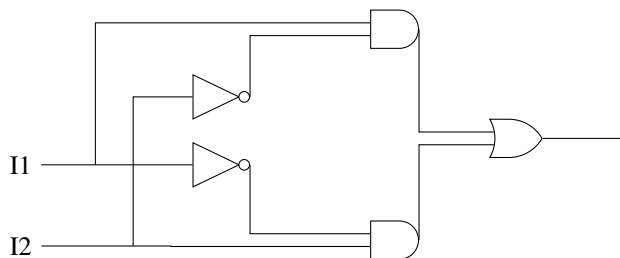
Kot je razvidno iz slike 2, vrata XOR sestavljata dva negatorja, dvojna vrata AND ter končna vrata OR. Vezje vsebuje tudi dve enostavni razvejitvi, t.j. razvejitvi vodil že pri vhodu.

Opazimo lahko, da bo v primerjavi z običajnim polprevodniškim vezjem vezje v obliki rešetkastega QCA malce bolj prostrano, saj vodil ni mogoče poljubno urejati.

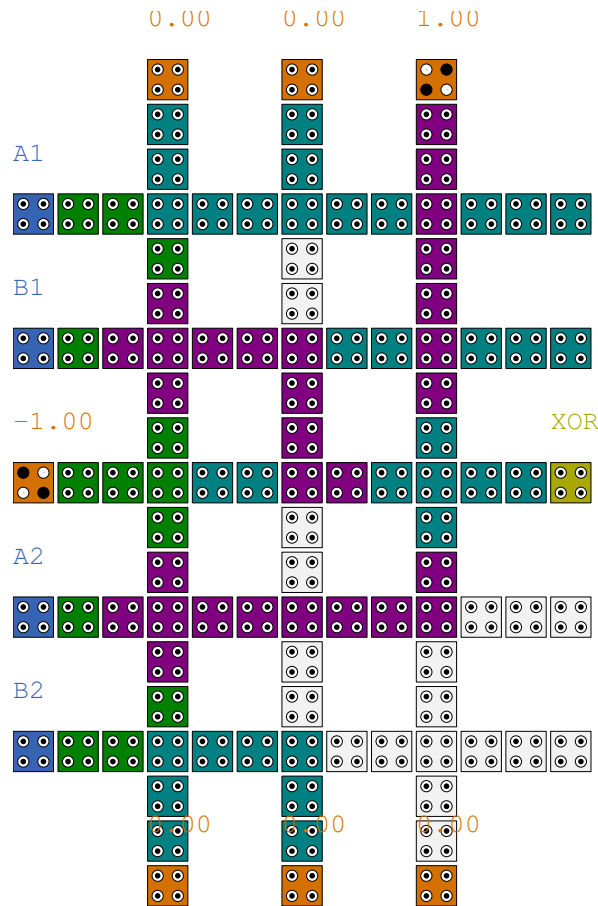
Za prvi poskus smo vrata poskušali oblikovati točno po zgornji shemi, t.j. s podobno fizično razporeditvijo in z obema razvejtvama. Rezultat ni bil zelo uspešen:

- rezultat na izhodu vezja ni ustrežal obnašanju, ki bi ga pričakovali od vrat XOR,
- polarizacija na izhodu ni bila zares stabilna, celica je nihala med vrednostmi zelo majhnih magnitud.

Osnovna izboljšava je preprosto sledenje priporočilom izvirnega članka, zato smo odstranili razvejitvi. Ker sta le-ti enostavni, je to mogoče zgolj s podvajanjem, npr. namesto vhoda A z razvejitvijo vezje nastavimo tako, da vanj privedemo dva vhoda, A1 in A2, od katerih pričakujemo, da imata vedno enako vrednost (to potem postane odgovornost "uporabniškega" vezja, katerega izhodi so kot vhodi priključeni na to rešetko). Skupaj z malce drugačno razporeditvijo potem dobimo stabilno delujoča vrata XOR, ki so prikazana na sliki 3.



Slika 2. Shema vrat XOR.



Slika 3. Končna vrata XOR v mrežni strukturi.

### Števec

Pri implementaciji eno (ali več)-bitnega števca imamo na voljo več različnih pristopov; direktne prevode klasičnih vezij ter pristope, ki bolj upoštevajo lastnosti QCA, tj. vezje števca, sestavljeno iz zank.

- Z direktnim prevodom imamo v mislih implementacijo, kjer z uporabo AND, OR in negatorskih vrat implementiramo tako vezje, kot je najpogosteje v uporabi pri klasičnih silicijevih tiskanih vezjih.
- Alternativno predstavlja pristop modela, sestavljenega iz zank, ki izkorišča eno od osnovnih komponent delovanja kvantnih celularnih avtomatov.

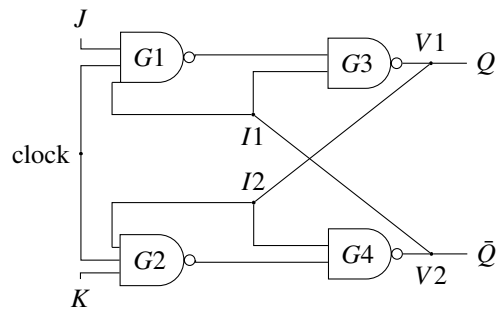
### Klasično vezje

Števci v običajnih polprevodniških vezjih so, na najnižjem nivoju, izvedeni z gručo navzkrižno povezanih vrat NAND: po štiri vrata NAND so povezana v JK multivibrator, dve kopiji le-tega pa skupaj z negatorjem sestavljata en bit števca. Vrata NAND so v QCA rešetki preprosta, saj imamo negator in vrata AND, ki ju potem zaporedno zvežemo skupaj. JK multivibrator je element, pri katerem se poskus implementacije ustavi. Glede na shemo zgoraj imamo znotraj klasičnega multivibratorja štiri razvejitve — povratno zanko, ki v multivibratorju ohranja nastavljeno stanje.

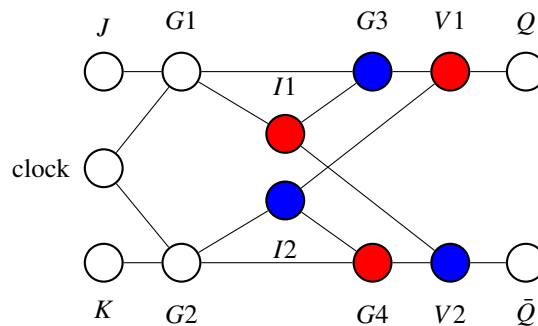
Razvejitve so v tej situaciji problematične zaradi dveh razlogov:

- signala v rešetki ni mogoče stabilno razcepiti v dve vodili,
- graf povezav med elementi sploh ni planaren.

*Opomba.* Izrek Kuratowskega nam pravi, da je graf planaren natanko tedaj, ko ne vsebuje subdivizij grafov  $K_{3,3}$  ali  $K_5$ . Na pogled vidimo, da je slika 4 ekvivalentna grafu na sliki 5. Ta graf pa kot podgraf vsebuje  $K_{3,3}$ : vozlišča  $\{I1, I2, G3, G4, V1, V2\}$  lahko razdelimo na množici  $\{I1, G4, V1\}$  in  $\{I2, G3, V2\}$ .  $\square$



**Slika 4.** Shema JK flip-flop vrat.



**Slika 5.** Shema JK flip-flop vrat prevedena v graf. Z modro in rdečo so označene točke, ki predstavljajo subdivizijo grafa  $K_{3,3}$ .

Pristop z rešetkami zahteva, da so kvantne pike urejene v linije enotske mreže, kar izdelovalcu onemogoča kakršnakoli križanja. Ker JK multivibrator, kot smo omenili zgoraj, ni ravninski, to pomeni, da ga v okviru enega nivoja QCA rešetke ni mogoče narediti.

#### **Vezje iz zank**

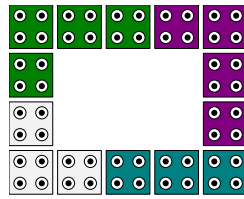
Vezje števca iz zank je bilo že večkrat testirano na kvantnih celularnih avtomatih, saj izkorišča eno od osnovnih komponent delovanja kvantnih celičnih avtomatov. Izkorišča lastnost adiabatske urine sheme, katere namen je, da sistem za določen trenutek ohrani elektrone v celicah v energetskem stabilnem stanju. Kvantni celularni avtomati uporabljajo urino shemo zaporedja štirih periodičnih faz:

- *Switch* faza ali faza preklopa
- *Hold* faza ali faza zadrževanja
- *Release* faza ali faza sproščanja
- *Relax* faza ali faza sproščenosti

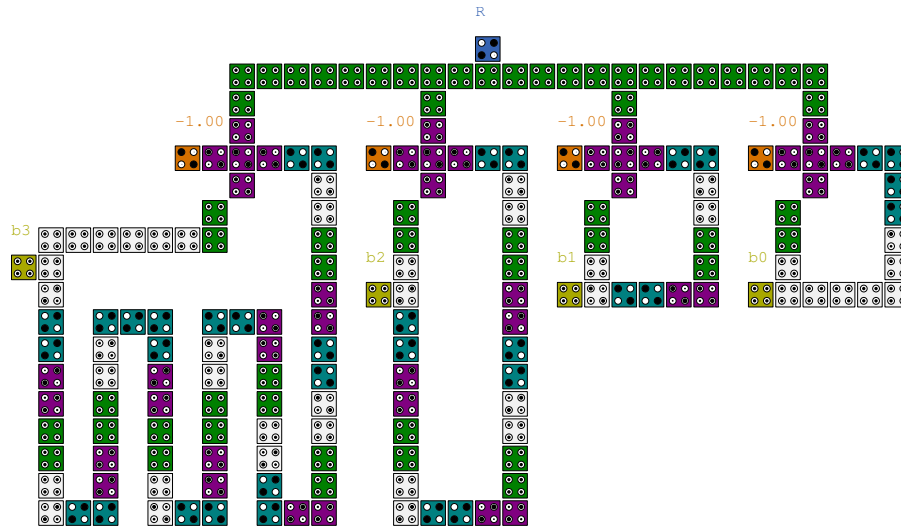
Zato, da kvantni celularni avtomati delujejo s pomočjo omenjenih faz, mora biti vezje implementirano tako, da vse celice v isti fazi delujejo z isto uro. To je lastnost, ki jo je nujno potrebno upoštevati pri izdelavi vezja, saj omenjene štiri faze določajo tok smeri signalov v vezju.

Zanka, ki je sestavljena iz štirih zaporednih faz, je predstavljena na sliki 6. Lahko bi jo smatrali kot spominsko celico za shranjevanje enega bita podatkov. Slednji način uporabe zank lahko uporabimo kot zadrževanje stanja za določeno število ciklov, kar je ključna lastnost, ki je potrebna za izdelavo števca. Eno-bitni števec je sestavljen iz treh komponent. AND vrat, negatorja in zanke. Dolžina zanke določa število urinih period, ko se trenutna logična vrednost obrne.

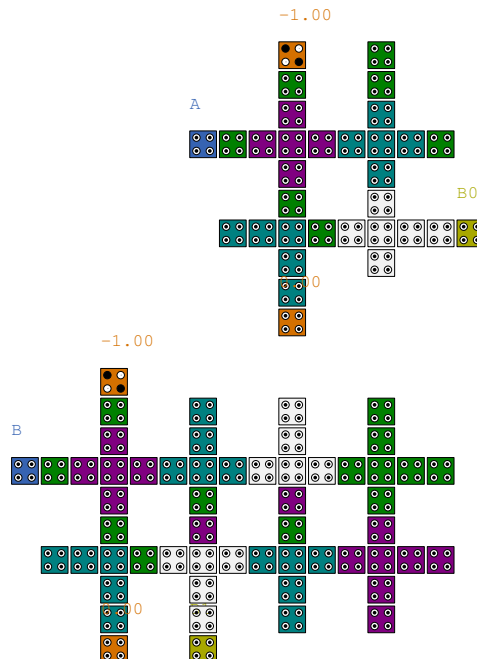
Za izdelavo 4-bitnega števca smo se z gledovali po načrtu, ki so ga zasnovali Sarmadi *et al.*<sup>7</sup>. Njihovo shemo na sliki 7 smo prilagodili zahtevam arhitekture programabilnih rešetk. Najprej smo implementirali preprost 1-bitni števec oz. 1-bitno zanko in po enakem principu razširili zanko v 2-bitno, 3-bitno in na koncu tudi 4-bitno zanko. Primer 1-bitne in 2-bitne zanke lahko pogledamo na sliki 8. Pri povezovanju zank različnih velikosti smo pri arhitekturi z rešetkami imeli veliko težav, saj nas ta



Slika 6. Spominska celica za en bit.



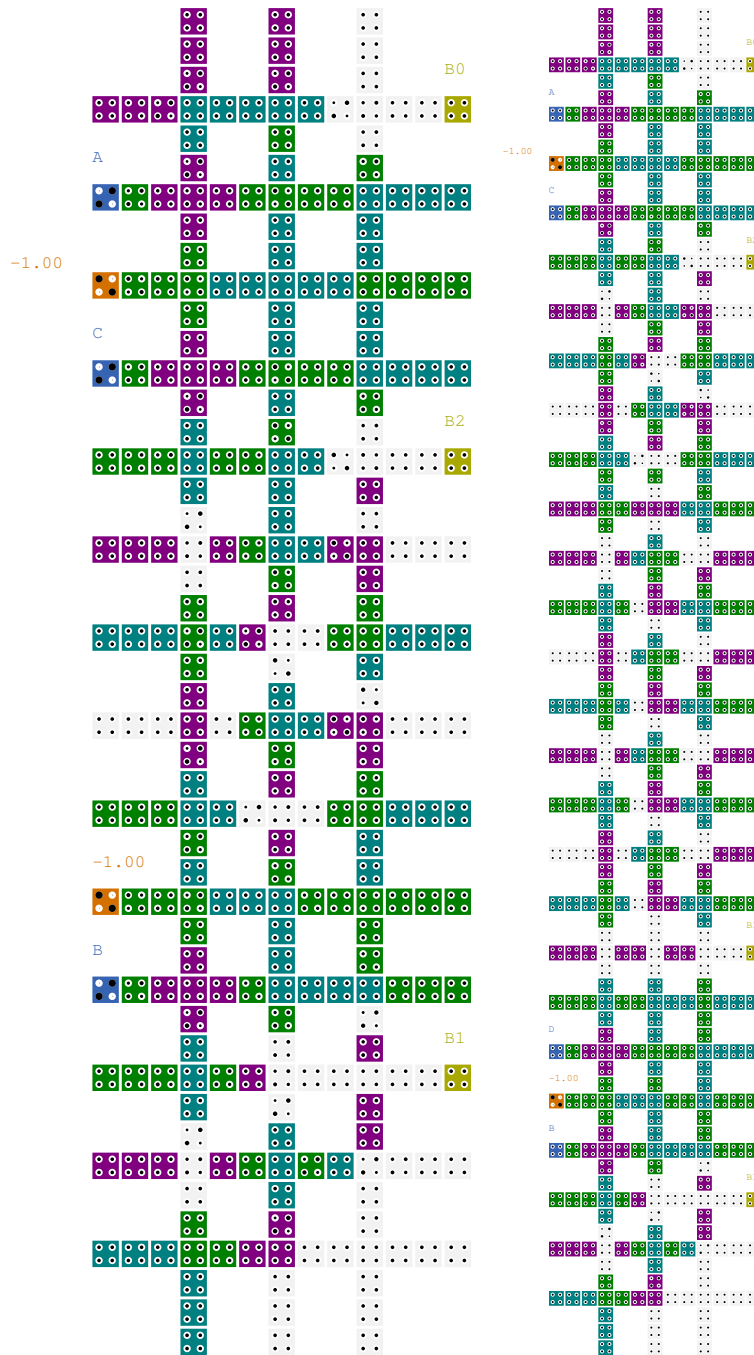
Slika 7. Shema 4-bitnega števca z zankami po zasnovi Sarmadiet *al.*<sup>7</sup>



Slika 8. Prototipa 1-bitnega in 2-bitne zanke.

omejuje pri poziciji vhodov in izhodov. Strukturo 4-bitnega števca smo morali prilagoditi arhitekturi rešetk. Implementirali smo ožjo in višjo strukturo, saj nam je ta omogočala enostaven izhod na desni strani in vhod na levi strani strukture. Prikaz strukture si lahko ogledamo na sliki 9. V primeru, da bi se odločili za širšo strukturo, bi potrebovali zakasnitve pri posameznih

zankah, da bi izhod izpeljali na skrajno desno stran rešetk. Poleg nujno potrebnih celic, ki dejansko nosijo informacijo za posamezno zanko, smo uporabili tudi dodatne celice za dopolnitev strukture do popolne oblike rešetk.



**Slika 9.** Levo prikazana realizacija 3-bitnega števec in desno realizacija 4-bitnega števec na arhitekturi programabilnih rešetk.

## Rezultati

Na slikah 10 in 11 smo prikazali rezultate simulacij za 3- in 4-bitni števec. Za izračun končne simulacije smo uporabili simulacijski model *Coherence vector*. Uporabili smo enake vrednosti parametrov simulacijskega modela kakor Sarmadi *et al.*<sup>7</sup> pri svojem števcu iz zank. Za pridobitev željenih rezultatov simulacije smo si pomagali s sestavljanjem vektorske tabele. Logična vrednost vhoda 0 predstavlja ponastavitev števec. Če želimo, da model deluje kot več-bitni števec, morajo vsi vhodi istočasno predstavljati enako logično vrednost. Rezultati prikazujejo robustno delovanje.

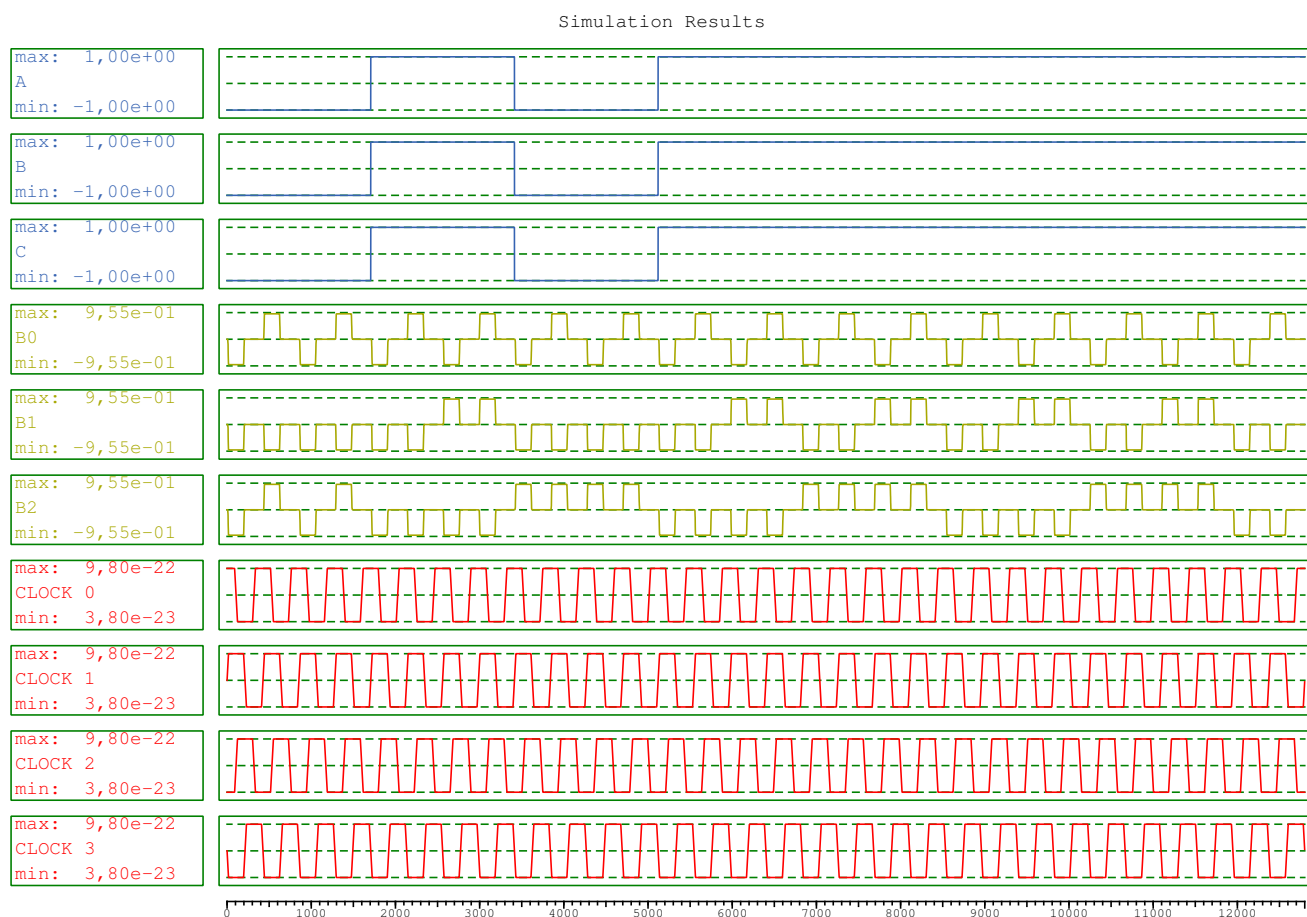
**Tabela 1.** Primerjava različnih metrik z drugimi implementacijami števca

Realizacije	Število celic	Porabljena površina ( $\mu m^2$ )	Časovna zakasnitev	Število nivojev	Število vhodov
Yang* <i>et al.</i> <sup>8</sup>	1130	2,20	7	1	1
Sheikhfaal <i>et al.</i> <sup>9</sup>	652	0,74	2	1	1
Aghababa <i>et al.</i> <sup>10</sup>	232	0,20	8	1	4
Sarmadi <i>et al.</i> <sup>7</sup>	183	0,24	8	1	1
Predstavljeni števec 9	496	0,57	8	1	4

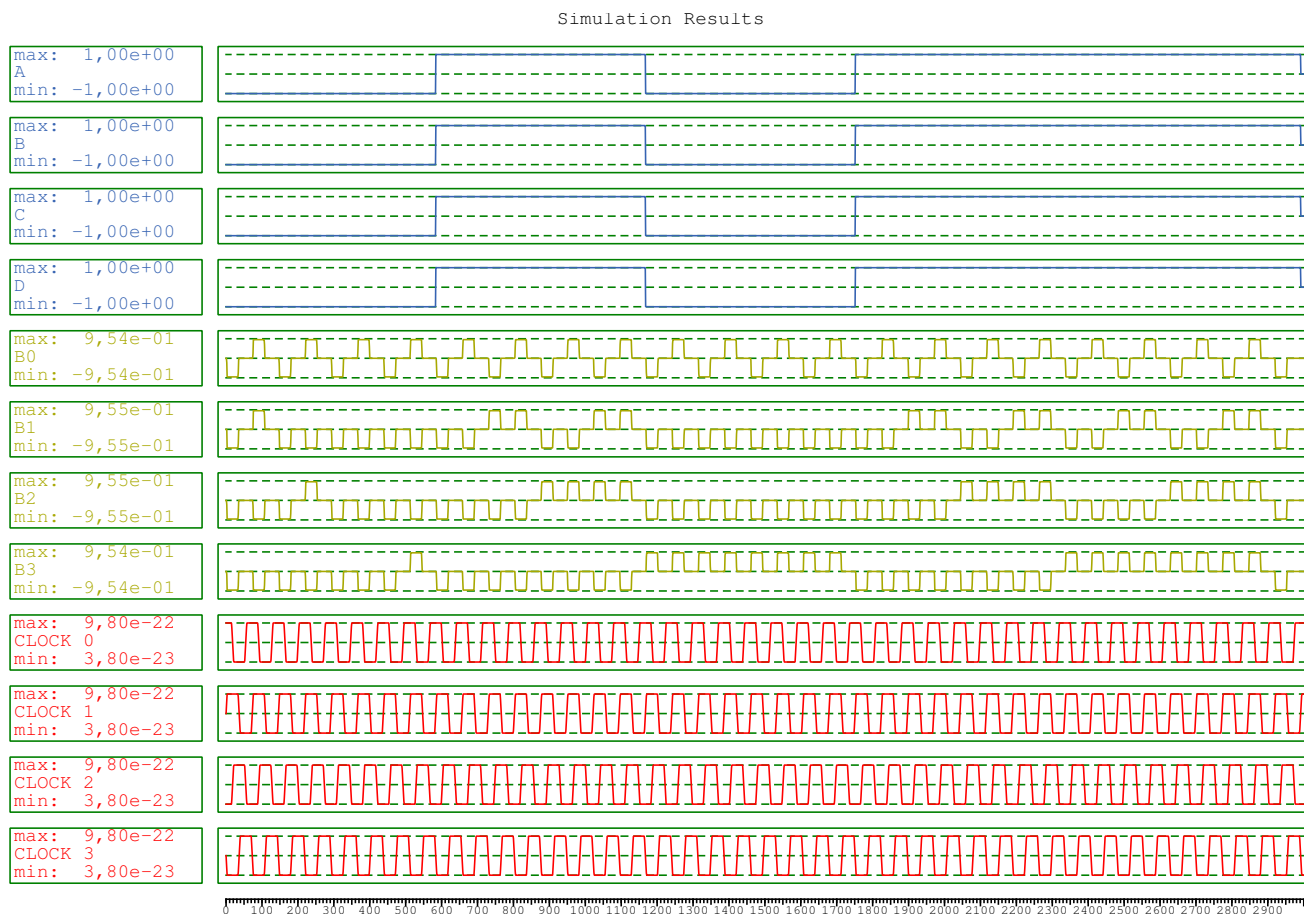
\* Fizično realizirali le JK multivibrator in 2-biten števec. Ocena porabe celic in prostora je ekstrapolacija na podlagi teh dveh eksperimentov in analize prostorske strukture vezja.

Tabela 1 predstavlja primerjavo različnih metrik z drugimi implementacijami števca. Opazimo lahko, da naš 4-bitni števec uporabi večje število celic in več površine v primerjavi z najboljšima primeroma. Z njima si deli enako časovno zakasnitev, ki pa je večja od ostalih primerov. Poleg tega uporabi štiri vhode, večina pa uporablja le enega. Predstavljeni primer ne presega ostalih implementacij po nobenem od kriterijev, ki so navedeni v tabeli 1. Ne smemo pa pozabiti omeniti, da naš števec predstavlja izboljšave, ki niso omenjene v predstavljeni metriki in tudi niso nezanemarljive.

Arhitektura programirljivih rešetak se poskuša približati delovanju vezja FPGA, zato predstavi polje kvantnih celic kot mrežo, ki ima na levi strani vhodne celice, na desni strani izhodne celice, navpično pa vodijo linije celic, ki s fiksnimi vrednostmi nastavijo pravilno delovanje vezja. Omenjene lastnosti ponujajo boljše možnosti za industrijsko proizvodnjo vezja kvantnih celularnih avtomatov in posledično tudi serijsko proizvodnjo.



**Slika 10.** Rezultati simulacije 3-bitnega števca. Logične vrednosti 0 vhodov A, B in C predstavljajo ponastavitev števca.



**Slika 11.** Rezultati simulacije 4-bitnega števca. Logične vrednosti 0 vhodov A, B, C in D predstavljajo ponastavitev števca.

## Zaključek

Pokazali smo, da lahko z uporabo rešetk kljub nekaterim omejitvam in prilagoditvam, implementiramo kompleksne strukture in vezja, ki v simulacijah dajejo dobre rezultate. Obdržali smo dobre lastnosti programirljivih rešetk. Za kriterije iz metrike v tabeli 1 menimo, da je na arhitekturi programirljivih rešetk odprtih še veliko možnosti za izboljšave. V procesu implementacije smo odkrili dokaj neugodno pomanjkljivost rešetk – na njih lahko izvedemo le planarna vezja, kar pa v nekaterih primerih ni dovolj. V prihodnosti bi nas zanimala razširitev rešetke v več plasti, kar bi morda rešilo problem, na katerega smo naleteli. Kljub nekaterim težavam in omejitvam pa menimo, da je raziskovanje te arhitekture zelo perspektivno zaradi njenega industrijskega potenciala.

## Literatura

1. Sen, B., Nag, A., De, A. & Sikdar, B. K. Towards the hierarchical design of multilayer qca logic circuit. *Journal of Computational Science* **11**, 233–244 (2015).
2. Sen, B. *et al.* Towards the design of hybrid qca tiles targeting high fault tolerance. *Journal of Computational Electronics* **15**, 429–445 (2016).
3. Tougaw, D., Szaday, J. & Will, J. D. A signal distribution grid for quantum-dot cellular automata. *J. Comput. Electron.* **15**, 446–454 (2016).
4. Ahmad, F. *et al.* Towards single layer quantum-dot cellular automata adders based on explicit interaction of cells. *Journal of Computational Science* **16**, 8 – 15 (2016).
5. Campos, C. A. T., Marciano, A. L., Neto, O. P. V. & Torres, F. S. Use: A universal, scalable, and efficient clocking scheme for qca. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **35**, 513–517 (2016).



6. Kalogeiton, V. S. *et al.* Programmable crossbar quantum-dot cellular automata circuits. *arXiv preprint arXiv:1604.07803* (2016).
7. Sarmadi, S., Azimi, S., Sheikhaal, S. & Angizi, S. Designing counter using inherent capability of quantum-dot cellular automata loops. *International Journal of Modern Education and Computer Science* **7**, 22 (2015).
8. Yang, X., Cai, L., Zhao, X. & Zhang, N. Design and simulation of sequential circuits in quantum-dot cellular automata: Falling edge-triggered flip-flop and counter study. *Microelectron. J.* **41**, 56–63 (2010).
9. Sheikhaal, S., Navi, K., Angizi, S. & Navin, A. H. Designing high speed sequential circuits by quantum-dot cellular automata: Memory cell and counter study. *Quantum Matter* **4**, 190–197 (2015).
10. Aghababa, H., Yazdinejad, M. H., Afzali, A. & Forouzandeh, B. Simplified quantum-dot cellular automata implementation of counters. In *2008 7th International Caribbean Conference on Devices, Circuits and Systems*, 1–4 (2008).

## Doprinos avtorjev

T. R. je izvedla pregled člankov<sup>3,4</sup>. J. U. je pregledala članke prvopodpisanega avtorja Sen<sup>1,2</sup>. Ž. P. je pregledal članka<sup>5,6</sup>. T. R. in Ž. P. sta implementirala števec z zankami, med tem ko sta J. U. in J. B. poskušala sestaviti vezje števca z JK multivibratorji.