

QCA: Dvobitni delilnik

Rok Česnovar 63060034, Jure Bordon 63060041, Rok Pustoslemšek 63050092, Dunja Rosina 63020136

Povzetek. Poročilo prikazuje delovanje dvobitnega delilnika realiziranega v programu QCA Designer pri katerem so snovni gradniki kvantne celice, te pa s povezovanjem v kompleksnejše strukture povezujemo v logična vrata, ki jih uporabljamo za realizacijo logičnih funkcij, ki so osnova za dvobitni delilnik.

Ključne besede. Dvobitni delilnik, QCA, modeliranje in simulacija

Kazalo.

1	Uvod	2
1.1	Zgodovina QCA strukture	2
1.2	Osnove QCA strukture	2
1.2.1	QCA celice	2
1.2.2	Vrata v QCA strukturi	3
1.3	Žice za prenos podatkov	4
1.4	Ura v QCA	5
2	Metode	5
2.1	Funkcija dvobitnega delilnika in minimizacija	5
2.2	Realizacija	7
3	Rezultati in ugotovitve	10
4	Zaključek	12
5	Viri in literatura	12

1. Uvod

Nekaj besed o QCA oziroma kvantnih celičnih avtomatih

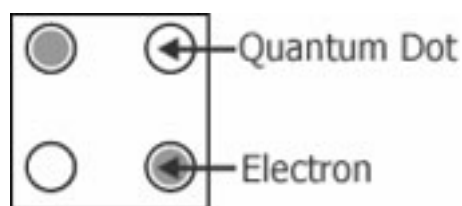
1.1. Zgodovina QCA strukture

Kvantne celične avtomate sta predlagala D. Tougaw in C. Lent kot nadomestilo klasični CMOS tehnologiji. Celični avtomati so ponavadi implementirani kot programi, a leta 1993 je Lent (University of Notre Dame) predlagal fizično implementacijo z uporabo t.i. QCA struktur (angleško quantum-dot-cellular automata). Lent je s svojim delom združil diskretne značilnosti obeh - celičnih avtomatov in kvantne mehanike, da je ustvaril naprave v velikosti nano delcev (razdalje med pikami v celicah so 20nm, velikost pike 10nm), ki so sposobne izredno hitrega preklapljanja med stanji ob izredno majhni porabi energije. Težavno pa je reševanje vplivov okolja na strukturo in tehnološki postopek izdelave.

1.2. Osnove QCA strukture

Glavni temelj QCA strukture je celica, ki si jo predstavljamo kot avtomat četverkotne oblike. QCA struktura je torej sestavljena iz poljubno razporejene množice celic v 2D ali 3D prostoru.

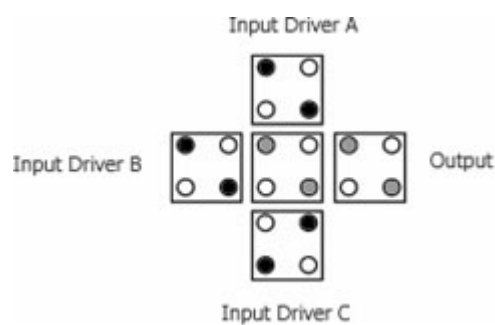
1.2.1. QCA celice V vsaki QCA celici se nahajajo štiri polprevodniške kvantne pike s pozitivnim privlakom, dva elektrona ter štirje tuneli preko katerih elektrona prehajata med različnimi stanji (tuneliranje).



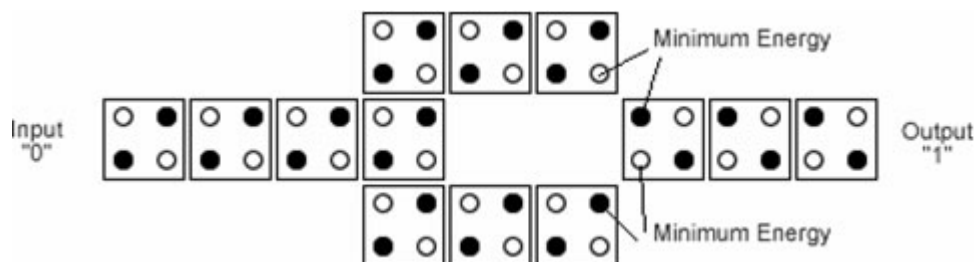
Slika 1: Izgled kvantne celice

Zaradi Coulombovega zakona elektrona težita k stabilnemu stanju; da so odbojne sile med elektronom in elektroni v sosednjih celicah čim manjše. Elektroni se lahko nahajajo le v tunelih ali v pikah. Iz tega torej sledi, da sta v celici dve stabilni stanji (diagonalni stanji).

1.2.2. *Vrata v QCA strukturi* Osnovna logična vrata v QCA strukturi so majoritetna vrata. Majoritetna vrata lahko vršijo funkciji AND in OR. Na spodnji sliki (2) so prikazana majoritetna vrata s tremi vhodi in enim izhodom. Če sta vhoda A in B v stanju 0, C pa v stanju 1, bo na izhodu stanje 0 - saj je zaradi odbojnih sil (Coulombov zakon) vpliv vhodov A in B močnejši od vhoda C. V kolikor bi stanje celice C nastavili na 0, bi bila edina možnost, da dobimo na izhodu 1 v primeru, ko sta tako A in B v stanju 1. Podobno lahko konstruiramo tudi OR vrata, če stanje celice - recimo C - postavimo na logično 1.



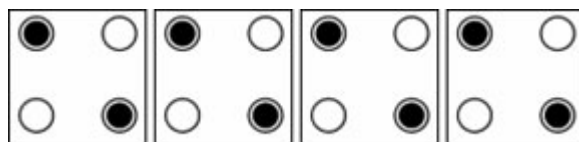
Slika 2: Majoritetna logična vrata



Slika 3: Negator

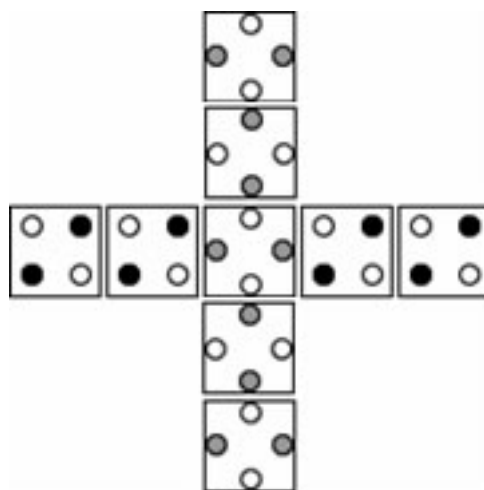
1.3. Žice za prenos podatkov

Žice omogočajo prenos podatkov. Lahko imamo 90° ali 45° žico za prenos podatkov (za dostavo logične vrednosti zahteva liho dolžino žice, saj v nasprotnem primeru dobimo na koncu žice negirano vrednost). Vhodna celica (oziroma driver cell) določa stanje vseh ostalih celic, ki (spet Coulombov zakon) ustrezno spremenijo svoje stanje.



Slika 4: Linija

Kako pa lahko križamo žice, narejene iz celic? Pravzaprav je velika prednost QCA ravno to, da omogočajo (načeloma) brezizgubno križanje žic, kar pa lahko pomeni odpravo večplastnosti povezav, ki so danes v vezjih praktično nujne. Prečkanje žic, zaradi lastnosti celic naredimo z eno 90° in eno 45° žico.



Slika 5: Križanje dveh žic

1.4. Ura v QCA

Zaradi nadzora toka podatkov moramo definirati smer, v kateri se sprememba stanja v celicah zgodi. Ravno zato vpeljemo uro v obliki električnega polja nad množicami celic, ki služi tudi kot napajanje (kot smo že omenili, QCA ne potrebuje veliko energije). Vpeljemo štiri urine faze:

- faza preklopa (switch): celice glede na vpliv okolice phajajo iz nepolariziranih v polarizirana stanja
- faza zadrževanja (hold): pregrade so dvignjene, tuneliranje onemogočeno
- faza sproščanja (release): pregrade se spuščajo, elektroni spet tunelirajo
- faza sproščenosti (relax): pregrade spuščene

2. Metode

2.1. Funkcija dvobitnega delilnika in minimizacija

Dvobitni delilnik smo predstavili kot logično funkcijo dveh spremenljivk (deljenec in delitelj) z izhodom prav tako dveh spremenljivk (količnik in ostanek). Za realizacijo potrebujemo minimizirane funkcije za količnik in ostanek. Ker je največji možni količnik 3 in ostanek 2, dobimo v ta namen 2 izhodna signala, ki predstavljata količnik (k_0, k_1) in 2 izhodna signala, ki predstavljata ostanek (O_0, O_1). Minimizirane enačbe funkcije delilnika lahko realiziramo, saj vsebujejo le operacije AND, OR in negacijo za katere uporabimo majoritetna vrata in negator. Ker je deljenje z 0 prepovedana operacija smo realizirali še dodaten signal, "deljenje z 0", ki nam kot kontrolni signal sporoča, kdaj je do te situacije prišlo.

Tabela logične funkcije delilnika v desetiškem zapisu:

y	x	k	O	$error$
0	0	x	x	1
0	1	0	0	0
0	2	0	0	0
0	3	0	0	0
1	0	x	x	1
1	1	1	0	0
1	2	0	1	0
1	3	0	1	0
2	0	x	x	1
2	1	2	0	0
2	2	1	0	0
2	3	0	2	0
3	0	x	x	1
3	1	3	0	0
3	2	1	1	0
3	3	1	0	0

(1)

In še v dvojiškem zapisu:

$y1$	$y0$	$x1$	$x0$	$k1$	$k0$	$O1$	$O2$	$error$
0	0	0	0	x	x	x	x	1
0	0	0	1	0	0	0	0	0
0	0	1	0	0	0	0	0	0
0	0	1	1	0	0	0	0	0
0	1	0	0	x	x	x	x	1
0	1	0	1	0	1	0	0	0
0	1	1	0	0	0	0	1	0
0	1	1	1	0	0	0	1	0
1	0	0	0	x	x	x	x	1
1	0	0	1	1	0	0	0	0
1	0	1	0	0	1	0	0	0
1	0	1	1	0	0	1	0	0
1	1	0	0	x	x	x	x	1
1	1	0	1	1	1	0	0	0
1	1	1	0	0	1	0	1	0
1	1	1	1	0	1	0	0	0

(2)

Minimizirane funkcije, ki smo jih nato realizirali v vezju:

$$k0 = (y1 \wedge y0) \vee (y0 \wedge \overline{x1}) \vee (y1 \wedge \overline{x0})$$

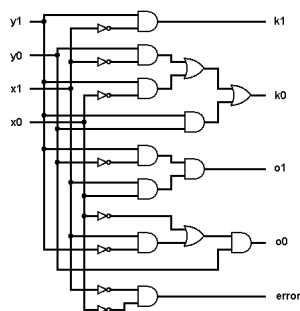
$$k1 = y1 \wedge \overline{x1}$$

$$o0 = y0 \wedge (\overline{x0} \vee (x1 \wedge \overline{y1}))$$

$$o1 = y1 \wedge \overline{y0} \wedge x1 \wedge x0$$

$$error = \overline{x1} \wedge \overline{x0}$$

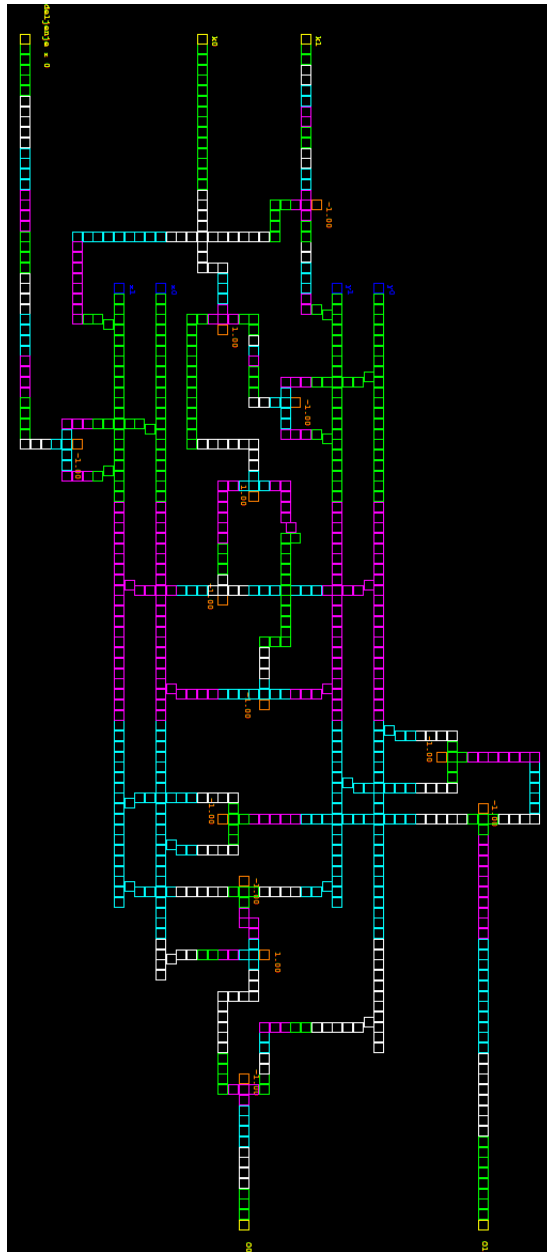
Te enačbe bi ustrezale naslednjemu CMOS vezju:



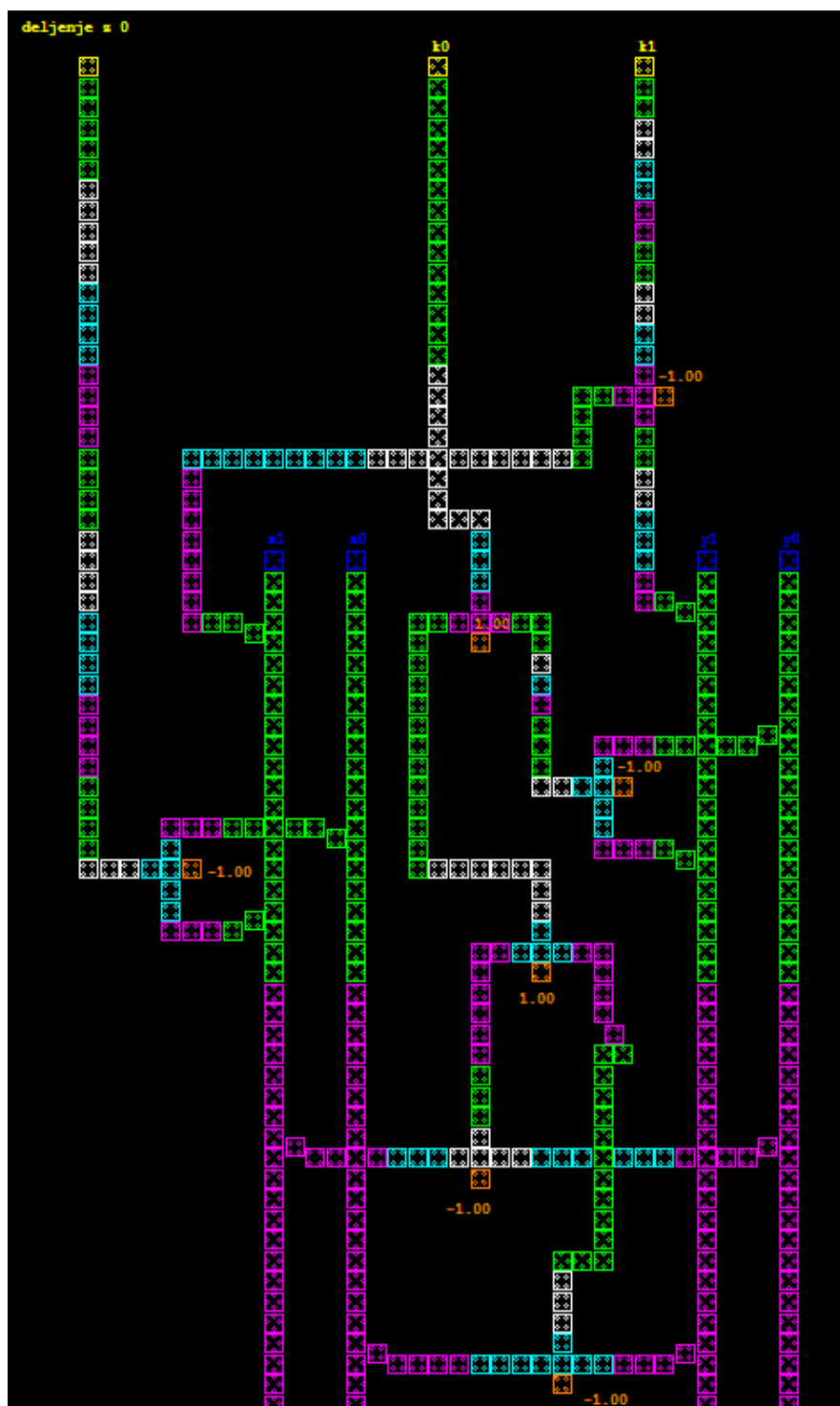
Slika 6: CMOS vezje

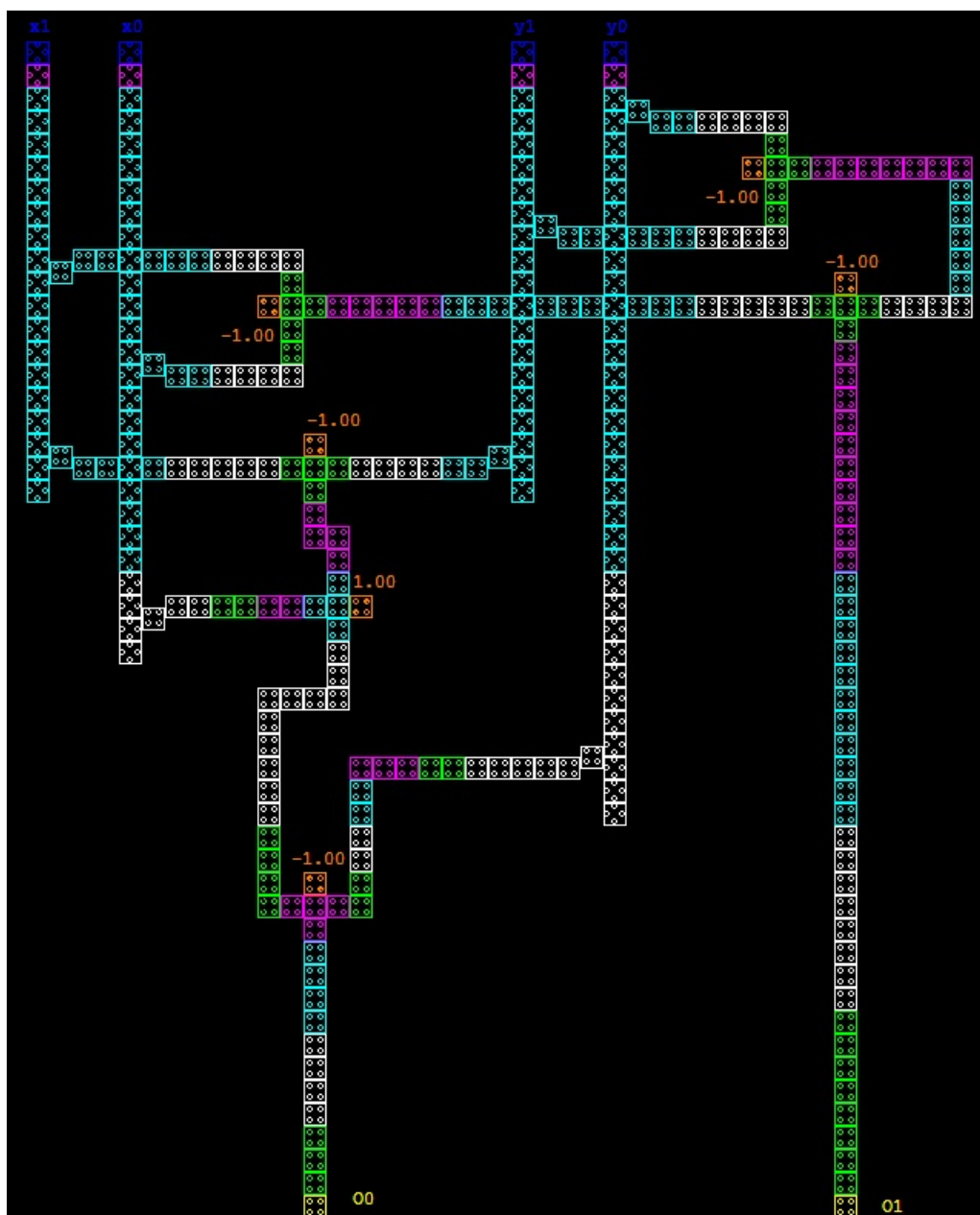
2.2. Realizacija

Na naslednjih slikah je prikazana realizacija celotnega vezja (7), nato pa podrobnejše še realizacija količnika (8) in ostanka (9).



Slika 7: Celotno vezje

Slika 8: Vezje za tvorbo količnika (k_0k_1) ter signala za deljenje z 0



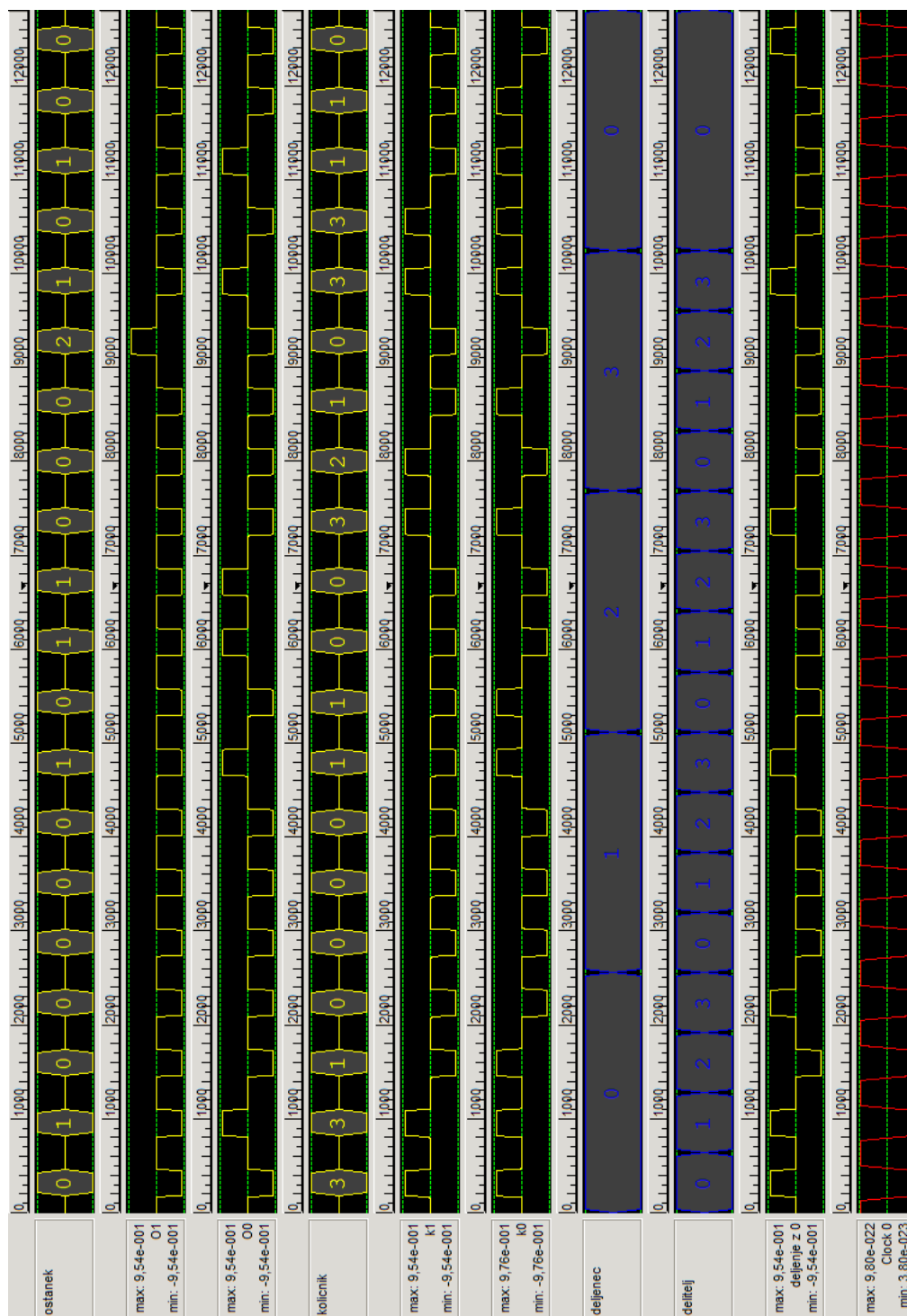
Slika 9: Vezje za tvorbo ostanka (0001)

3. Rezultati in ugotovitve

V našem vezju je 766 celic. Menimo, da bi število celic lahko še zmanjšali, a ne bistveno, saj ožanje razdalj med posameznimi linijami (in s tem zmanjševanje števila celic) pogosto spremeni potek delovanja vezja. Največ problemov nam je povzročalo ravno to. Pri premiku celic bližje določenim linijam je popolnoma spremenilo delovanje vezja na zelo oddaljenem mestu (ker smo na primer z prečkanjem oslabili signal v vertikalni liniji v zgornjem delu, nam je prenehal delovati izhod iz te linije v nižjem delu vezja). Imeli smo torej opcijo krajšanja razdalje med linijami in višanja števila ur (kjer se signal preveč oslabi ga lahko ojačamo s spremembo ure) ali pa obdržati dovolj velike razdalje med linijami in s tem manjšati število urinih period. Ker smo mnenja, da je hitrost delovanja samega vezja bolj pomembna, smo se odločili za slednje.

Naše vezje za izračun enega deljenja potrebuje 12 urinih period, rezultat posamezne operacije je torej zamaknjen za 3 cele urine periode. V tabeli rezultatov imamo 16 vhodnih kombinacij za 4 spremenljivke. Y1 predstavlja bit1 deljenca, Y0 pa bit0. Podobno velja za vhodna signala x1 in x0, ki predstavljata delitelja. Pri prvi enici na izhodu "deljenje z 0" imamo torej odziv na začetno stanje vhodnih signalov (0,0,0,0). Z nastavljanjem drugih parametrov testiranja se nismo ukvarjali, našo simulacijo smo izvajali na privzetih vrednostih, stanja celic pa smo računali s postopkom bistabilne aproksimacije.

Simulacijo je pokazala pravilne rezultate tudi če smo jo pognali večkrat zapored ter tudi z več različnimi vektorji zapored. Slika (10) na naslednji strani prikazuje rezultate simulacije enega izmed vektorjev, ki vsebuje vse kombinacije deljenj.



Slika 10: Rezultati simulacije našega vezja v QCADesigner-ju

4. Zaključek

Kvantni celični avtomati so strukture, ki so šele v fazi razvijanja, njihovo delovanje pa je pogojeno z poglobljenim znanjem kvantne fizike. QCADesigner nam omogoča, da brez poglobljanja v fizikalno ozadje načrtujemo kompleksna logična vezja in s pomočjo simulacije določimo pravilnost delovanja našega vezja. Pri naši realizaciji smo imeli ogromno problemov s križanji linij in nepredvidljivimi odzivi vezja. Veliko težav smo rešili s testiranjem in poizkušanjem, saj smo hitro ugotovili, da nekatere kombinacije poteka linij in pozicij vrat preprosto ne delujejo. Tako smo posledično uvedli urine periode že na sama vodila, med posameznimi deli vezja pa smo pustili več prostora. Na koncu nam je delilnik uspelo realizirati enoplastno, za kar pa smo porabili 12 urinih period (oz 3 cele). V kolikor bi želeli realizirati večje, kompleksnejše sisteme s kvantnimi celičnimi avtomati, bi po našem mnenju potrebovali boljši simulator, kot je QCADesigner. Orodju manjka neke vrste razhroščevalnik, saj ob dodajanju novih elementov vezju hitro porušimo delovanje drugih delov, ki smo jih predhodno že testirali. Ob načrtovanju večjih sistemov je posledično iskanje vzroka zaka kaj neka logično pravilna realizacija ne deluje izjemno časovno potratno. Področje kvantnih celičnih avtomatov je predvsem zaradi visokih frekvenc ure ter nizke porabe izjemno privlačno in je zato sigurno ena od potencialno uporabnih nanotehnologij v prihodnosti.

5. Viri in literatura

- (i) Mraz, M., 2009. Zapiski s predavanj: 1. Uvod v nanoračunalništvo
- (ii) <http://www.mina.ubc.ca/qcadesigner>