

Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Matej Dolenc

Reverzibilni pretvornik QCA med binarno in Grayevo kodo

MAGISTRSKA NALOGA
ŠTUDIJSKI PROGRAM DRUGE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

prof. dr. Miha Mraz
MENTOR

Ljubljana, 2018

© 2018, Matej Dolenc

Rezultati magistrskega dela so intelektualna lastnina avtorja ter Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavljanje ali izkoriščanje rezultatov magistrskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani izjavljam, da sem avtor dela, da slednje ne vsebuje materiala, ki bi ga kdorkoli predhodno že objavil ali oddal v obravnavo za pridobitev naziva na univerzi ali drugem visokošolskem zavodu, razen v primerih kjer so navedeni viri.

S svojim podpisom zagotavljam, da:

- sem delo izdelal samostojno pod mentorstvom prof. dr. Mihe Mraza,
- so elektronska oblika dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko in
- soglašam z javno objavo elektronske oblike dela v zbirki "Dela FRI".

— Matej Dolenc, Ljubljana, november 2018.

Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Matej Dolenc

Reverzibilni pretvornik QCA med binarno in Grayevo kodo

POVZETEK

V pričujočem magistrskem delu predstavimo dve, še relativno nezreli veji računalništva, ki bi v prihodnosti lahko vodili nadaljnji razvoj te stroke. Prva izmed teh dveh vej so kvantni celični avtomati (QCA), ki predstavljajo nov možen način gradnje elektronskih logičnih vezij. Predstavljajo eno izmed alternativ, ki bi v prihodnosti lahko zamenjala trenutno še vedno aktualne tranzistorje v digitalnih vezjih. V magistrskem delu predstavimo kvantno celico, strukturo, ki jih s kvantnimi celicami gradimo, uporabo ure v QCA vezjih in programsko orodje QCADesigner. Druga veja, ki se ji posvetimo, je reverzibilnost v smislu možnega reverzibilnega procesiranja podatkov. Razložimo pojem reverzibilnosti in predstavimo različne reverzibilne logične funkcije.

V nadaljevanju predstavimo Grayevo kodo, postopek pretvorbe binarne kode v Grayevo kodo in različne postopke za generiranje Grayeve kode. Prikažemo tudi pristope k realizaciji pretvorbe iz binarne v Grayevo kodo in sicer na modelu klasičnega elektronskega vezja, na modelu QCA in na primeru elektronsko optične realizacije.

Problem, ki ga rešujemo v magistrski nalogi, je gradnja pretvornika iz binarne v Grayevo kodo. Tega se lotimo s pomočjo QCA struktur in reverzibilnosti. Za začetek v literaturi poiščemo več različnih že obstoječih pretvornikov, ki jih ocenimo z ustreznimi metrikami. Zatem zgradimo naš lastni QCA pretvornik iz binarne v Grayevo kodo, ki ga postopoma nadgrajujemo in sproti primerjamo z že obstoječimi rešitvami na podlagi predhodno omenjenih metrik. Model QCA pretvornika zgradimo s pomočjo referenčnega orodja QCADesigner. Na koncu podamo končni komentar in oceno našega QCA pretvornika glede na obstoječe rešitve iz literature.

Ključne besede: kvantni celični avtomati, reverzibilnost, QCADesigner, Grayeva koda

University of Ljubljana
Faculty of Computer and Information Science

Matej Dolenc

Reversible binary to Gray code QCA converter

ABSTRACT

In this master thesis we introduce two rather unknown branches of computer science which could potentially in the near future lead this field of expertise forward. Firstly, we introduce quantum dot cellular automata (QCA) which represents new approach to implementation of digital circuits. QCA is one of the alternatives to still relevant transistor in digital circuits. Quantum cell, QCA structures, QCA clock and QCADesigner — software for design and simulation of QCA structures, are introduced. Secondly, we introduce reversibility in the sense of reversible data processing. Reversibility and different reversible logic functions are explained.

Next, we shift focus to Gray code, binary to Gray code conversion and different approaches for generating Gray code. In addition, different approaches to binary to Gray code converter implementation are introduced e.g. classic electronic wiring, QCA model and electro-optical implementation.

The main goal of this master thesis is implementing binary to Gray code converter. To achieve this we make use of QCA and reversibility. For starters, we take a look at different existing binary to Gray code converters that are based on QCA technology. We define different metrics for these converters. With the help of QCADesigner we create our own QCA binary to Gray code converter and gradually improve upon it based on previously defined metrics. At the end we compare our converter with existing converters and give our final comments.

Key words: quantum dot cellular automata, reversibility, QCADesigner, Gray code

ZAHVALA

Zahvaljujem se mentorju prof. dr. Mihi Mrazu za nasvete in vodenje pri izdelavi magistrskega dela. Zahvaljujem se tudi vsem mojim najbližjim za podporo in vero v moje sposobnosti.

— Matej Dolenc, Ljubljana, november 2018.

KAZALO

Povzetek	i
Abstract	iii
Zahvala	v
1 Uvod	1
1.1 Motivacija	1
1.2 Pregled dela	2
2 QCA strukture	5
2.1 QCA celica	6
2.2 QCA strukture	7
2.2.1 Linija	7
2.2.2 Negator	7
2.2.3 Majoritetna vrata	8
2.3 Ura v QCA	10
2.4 Programsko orodje QCADesigner	12
2.4.1 Bistabilna aproksimacija	13
2.4.2 Koherenčni vektor	13
3 Reverzibilnost	17
3.1 Osnove reverzibilnosti	17
3.2 Fizična reverzibilnost	18
3.3 Logična reverzibilnost	18
3.4 Reverzibilne logične funkcije	19
3.4.1 Feynmanova funkcija	20

3.4.2	Fredkinova funkcija	21
3.5	Uporabnost reverzibilnosti	22
4	Grayeva koda	25
4.1	Osnove Grayeve kode	26
4.2	Postopek za pretvorbo iz binarne v Grayevo kodo	27
4.3	Postopki za generiranje Grayeve kode	28
4.3.1	Generiranje z reflektiranjem	28
4.3.2	Generiranje z razširjanjem	30
4.4	Pristopi k realizaciji pretvorbe iz binarne v Grayevo kodo	31
4.4.1	Model klasičnega elektronskega vezja	32
4.4.2	Model realizacije s kvantnim celičnim avtomatom	33
4.4.3	Elektronsko optična realizacija	35
5	Pregled QCA pretvornikov iz binarne v Grayevo kodo	37
5.1	Nereverzibilne QCA rešitve	38
5.1.1	Pretvornik po Islamu	38
5.1.2	Pretvornik po Ahmadu	38
5.1.3	Pretvornik po Bhowmiku	40
5.1.4	Pretvornik po Vijayalakshmi	40
5.1.5	Pretvornik po Majeedu	40
5.2	Reverzibilne QCA rešitve	41
5.2.1	Pretvornik po Karkaju	42
5.2.2	Pretvornik po Reshiju	42
6	Predlagana rešitev reverzibilnega QCA pretvornika	45
6.1	Osnovna rešitev	45
6.1.1	Feynmanova vrata kot QCA realizacija	46
6.1.2	Logična in QCA shema pretvornika	47
6.1.3	Analiza delovanja QCA strukture	47
6.1.4	Ocena osnovne rešitve	49
6.2	Izboljšave osnovne rešitve	52
6.3	Lokalna minimizacija	53
6.4	Globalna minimizacija	55

6.4.1	Minimizacija z ohranitvijo ene urine faze	56
6.4.2	Minimizacija z uporabo več urinih faz	56
6.5	Dodatek za reverzibilnost	57
6.6	Simulacija strukture	60
6.7	Ocena rešitve	60
7	Zaključek	65
A	Priloge	73
A.1	Simulacija reverzibilnega QCA pretvornika, ki za svoje delovanje uporablja dve urini fazi	73
A.2	Poizkusi gradnje QCA struktur	75
A.2.1	2-bitni QCA pretvornik iz binarne v Grayevo kodo	75
A.2.2	3-bitni QCA pretvornik iz binarne v Grayevo kodo	77
A.2.3	4-bitni QCA pretvornik iz binarne v Grayevo kodo	79
A.2.4	Poizkusi gradnje alternativnih XOR logičnih vrat	79

1 Uvod

Trenutno prevladujoč trend v mikroelektronski industriji je še vedno tranzistor na poljski pojav (angl. *field effect transistor* - FET) [1]. Predvideva se, da se bo v bližnji prihodnosti pojavila težava pri nadaljnji miniaturizaciji FET tranzistorja. Gordon Moore je v šestdesetih letih prejšnjega stoletja izjavil, da se bo gostota tranzistorjev na mikročipih podvojila na približno vsaki dve leti. V zadnjih nekaj letih se je podvojevanje števila tranzistorjev začelo močno upočasnjevati. Do leta 2020 naj bi že lahko dosegli kritično spodnjo mejo velikosti tranzistorja, po čemer bo nadaljni razvoj tranzistorjev na osnovi obstoječih tehnologij in metod nemogoč [2].

1.1 Motivacija

Ena izmed možnih alternativ tranzistorja za potrebe računalniškega procesiranja je *kvantni celični avtomat* (angl. *quantum dot cellular automata* - QCA). QCA je sestavljen iz množice kvadratnih celic, v katerih so v vogalih štiri pozitivno nabite kvantne pike med seboj povezane s tuneli, v sami celici pa sta ujeta dva elektrona, ki lahko prehajata med pikami v celici po predhodno omenjenih tunelih. Zaradi Coulombovega zakona o silah

med dvema elektronoma se bosta slednja venomer pozicionirala v dve nesosednji kvantni piki. Ti stanji predstavljata dve možni stabilni stanji elektronov v celici, ki ju lahko interpretiramo kot dvostanjski sistem ali kot logično ničlo ali enico.

Na tem mestu vpeljemo še pojem *logične reverzibilnosti* (angl. *logical reversibility*). Reverzibilnost v splošnem omogoča obrnljivost procesov, kar pomeni, da lahko prejšnje stanje procesa vedno rekonstruiramo iz trenutnega stanja procesa [3]. Reverzibilen proces vedno izvaja reverzibilna logična funkcija. Da je logična funkcija reverzibilna, morata biti izpolnjena dva pogoja. Takšna funkcija mora imeti enako število vhodov in izhodov [4] ter mora biti bijektivna. Poznamo več različnih reverzibilnih funkcij. Ena izmed njih je Feynmanova funkcija, ki jo v tem delu uporabimo kot osnovo za izgradnjo reverzibilnega QCA pretvornika iz binarne v Grayevo kodo.

Grayeva koda se je v računalništvu začela uporabljati zaradi nekaterih pomankljivosti binarne kode. Področja uporabe Grayeve kode obsegajo popravljanje napak pri digitalnih televizijskih sistemih, pretvornike med analognim in digitalnim signalom in rotacijske ter optične kodirnike v robotski industriji. Glavna razlika glede na binarno kodo je v tem, da se zapisi sosednjih števil v Grayevi kodi razlikujejo le v enem znaku.

1.2 Pregled dela

Namen magistrskega dela je predstavitev QCA platforme, reverzibilnosti in Grayeve kode, v nadaljevanju pa izgradnja reverzibilnega QCA pretvornika iz binarne v Grayevo kodo. V poglavju 2 najprej predstavimo QCA strukture in referenčno programsko orodje QCA-Designer, ki ga uporabimo za izgradnjo modela našega QCA pretvornika. V poglavju 3 predstavimo pojem reverzibilnosti in sicer na podlagi fizične in logične reverzibilnosti. To razširimo z reverzibilnimi logičnimi funkcijami, na koncu pa podamo tudi prednosti reverzibilnega procesiranja pred klasičnim procesiranjem. V poglavju 4 predstavimo Grayevo kodo, postopke za pridobivanje Grayeve kode iz binarne kode in nekaj različnih pristopov k realizaciji pretvornikov. Sledi pregled že obstoječih QCA pretvornikov v poglavju 5, ki nam v nadaljevanju služijo kot referenca za primerjavo naše rešitve z že obstoječimi rešitvami. V poglavju 6 predstavimo postopek izgradnje naše rešitve reverzibilnega QCA pretvornika iz binarne v Grayevo kodo v programskem orodju QCADesigner. Pri gradnji se osredotočimo na doseg logične reverzibilnosti naše rešitve. Logično reverzibilnost smatramo na nivoju posameznih gradnikov (uporaba Feynmanovih vrat). V sklopu gra-

dnje našega pretvornika zgradimo nova XOR logična vrata in nova Feynmanova vrata, s pomočjo katerih končno rešitev močno izboljšamo v primerjavi z že obstoječimi rešitvami. Na koncu opravimo simulacijo našega modela in na podlagi pridobljenih rezultatov primerjamo našo rešitev z rešitvami iz literature.

2 QCA strukture

Kvantni celični avtomat (angl. *quantum dot cellular automata*) ali krajše QCA je kot prvi leta 1993 predstavil C. S. Lent s sodelavci na univerzi Notre Dame v Indiani, ZDA [5]. Osnova za kvantne celične avtomate je *kvantna celica*, v kateri pride do pojava Coulumbove interakcije med dvema ujetima elektronoma. Kvantna celica v tem primeru predstavlja bistabilni sistem — osnovo za logično ničlo in enico. S pomočjo energetske odbojne interakcije med elektroni večih sosednjih celic so raziskovalci uspeli realizirati v skupkih celic različne logične funkcije. V pričujočem poglavju predstavimo osnove kvantnih celičnih avtomatov, ki so potrebne za nadaljnje razumevanje tega magistrskega dela.

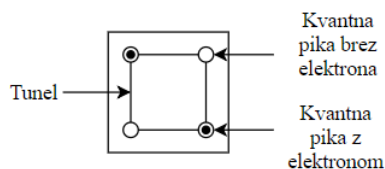
QCA je sestavljen iz množice kvadratnih celic, v katerih so v vogalih štiri pozitivno nabite kvantne pike med seboj povezane s tuneli, v sami celici pa sta ujeta dva elektrona, ki lahko prehajata med pikami v celici po predhodno omenjenih tunelih. Zaradi Coulumbovega zakona o silah med dvema elektronoma se bosta slednja venomer pozicionirala v dve nesosednji kvantni piki. Ti stanji predstavljata dve možni stabilni stanji elektronov v celici, ki ju lahko interpretiramo kot dvostanjski sistem ali kot logično ničlo

ali enico. S pomočjo elektronov, ujetih v kvantni celici, lahko implementiramo različne logične Boolove funkcije [6].

QCA prinaša številne prednosti pred tradicionalnimi tranzistorji — velike hitrosti preklapljanja (prehajanja med stanji), veliko gostoto kvantnih celic na površinsko enoto in majhne toplotne izgube [7]. Na tem mestu omenimo še to, da QCA praktično nima nobene povezave s kvantnim računalništvom (angl. *quantum computing*).

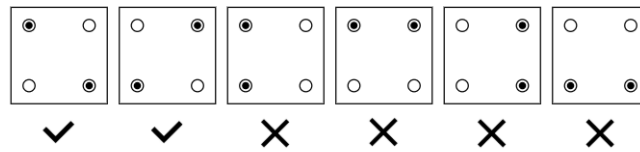
2.1 QCA celica

Osnovo pri gradnji QCA struktur predstavlja posamezna QCA celica. Sestavljena je iz štirih pozitivno nabitih kvantnih pik in štirih navideznih tunelov, ki omenjene kvantne pike povezujejo med seboj. V kvantnih pikah sta ujeta dva elektrona, ki lahko med kvantnimi pikami v celici prosto prehajata s pomočjo tuneliranja, ne moreta pa prehajati iz svoje celice v sosednje celice [7]. Zaradi Coulombovega zakona o silah med dvema elektronom se bosta slednja vedno pozicionirala v stanje medsebojne maksimalne oddaljenosti — v eno izmed dveh možnih polariziranih stanj na diagonalah celice. Ti dve stanji predstavljata najstabilnejši energetska stanja v QCA celici. QCA celica je prikazana na sliki 2.1, na sliki 2.2 pa obe dovoljeni, polarizirani stanji in vsa nedovoljena, nepolarizirana stanja razporeditve elektronov v QCA celici.

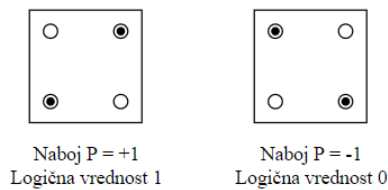


Slika 2.1: Celica kvantnega celičnega avtomata. Sestavljajo jo štiri pozitivno nabite kvantne pike, ki so med seboj povezane z navideznimi tuneli. V kvantnih pikah sta ujeta dva elektrona, ki se zaradi Coulombovega zakona vedno postavita v eno izmed dveh možnih polariziranih stanj na diagonalah celice.

Obe možni stanji v QCA celici predstavljata dve različni polarizaciji QCA celice. Polarizacija $P = -1$ predstavlja logično ničlo, polarizacija $P = +1$ pa logično enico, kar lahko enačimo z logičnim stanjem celice. Tako lahko dve možni stanji QCA celice interpretiramo kot dvostanjski sistem [8] (glej sliko 2.3).



Slika 2.2: Polarizirani stanji in nepolarizirana stanja elektronov v QCA celici.



Slika 2.3: Obe možni polarizirani stanji v QCA celici. Leva celica predstavlja logično enico, desna celica pa logično ničlo.

2.2 QCA strukture

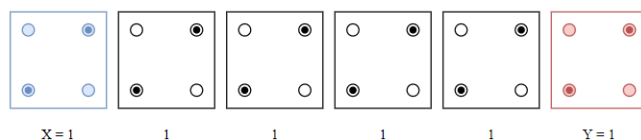
Polarizacija ene QCA celice vedno povzroči polarizacijo tudi v sosednjih celicah na podlagi Coulombove interakcije med elektroni [8]. Na ta način lahko gradimo strukture s postavitvijo celic v različne konfiguracije. V nadaljevanju predstavimo QCA strukture, ki predstavljajo osnovo za gradnjo večjih, bolj kompleksnih struktur.

2.2.1 Linija

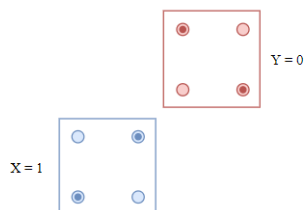
QCA linija (angl. *QCA wire*) je najosnovnejša struktura QCA. Dobimo jo z zaporedno vezavo več celic [9]. Na začetku linije imamo t.i. *gonilno celico* (angl. *driver cell*), katere polarizacija je fiksne narave (logično stanje celice skozi čas je logična ničla ali enica in se ne spreminja). Preko Coulombove interakcije se polarizacija ali stanje te celice propagira v sosednje celice. Na ta način dosežemo širjenje signala preko QCA linije. Primer QCA linije je prikazan na sliki 2.4.

2.2.2 Negator

Negator predstavlja negacijo logičnega stanja vhodne celice. Če dve celici na površini pozicioniramo diagonalno bosta imeli nasprotujoči si polarizaciji (različni logični vrednosti) [10]. Osnovna struktura negatorja je predstavljena na sliki 2.5.



Slika 2.4: QCA linija, kjer se signal propagira od leve vhodne gonilne celice X, proti desni izhodni celici Y.

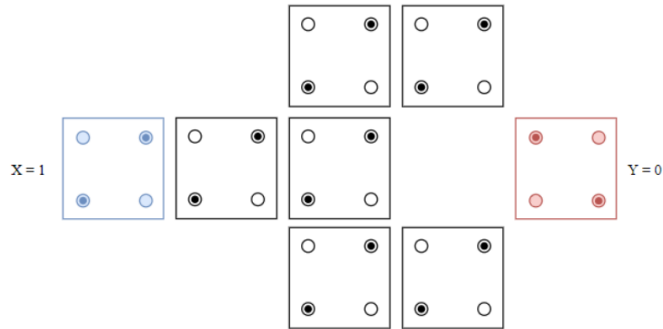


Slika 2.5: Osnovna struktura QCA negatorja. X predstavlja vhodno celico, Y pa izhodno celico.

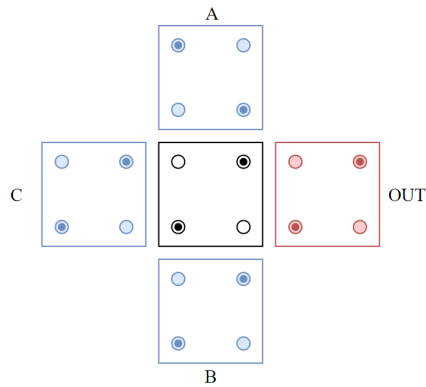
Negator lahko zgradimo tudi drugače. Alternativna struktura negatorja je prikazana na sliki 2.6. Tak negator je bolj robusten in odporen na vplive sosednjih celic. Opazimo, da alternativni negator predstavlja relativno veliko strukturo za zelo preprosto logično operacijo negacije. Negator v CMOS tehnologiji predstavlja najbolj preprost gradnik. Pri QCA tehnologiji opazimo, da negator predstavlja večji gradnik in zato tudi porabi relativno veliko prostora v primerjavi z ostalimi QCA gradniki [11].

2.2.3 Majoritetna vrata

V pričujočem razdelku predstavimo najpomembnejši logični gradnik, ki se uporablja pri gradnji QCA struktur. Omenjeni gradnik predstavlja *majoritetna vrata* (angl. *majority gate* ali *majority voter*), prikazana na sliki 2.7. Majoritetna vrata so trovhodna logična vrata, ki jih sestavlja pet kvantnih celic [12]. Eden izmed vhodov A, B ali C je vedno fiksno polariziran na logično ničlo ali enico, ostala dva vhoda pa sta poljubna. Katerega izmed vhodov izberemo za fiksno polariziranega ni pomembno. Celica v sredini majoritetnih vrat opravi procesiranje vhodnih vrednosti. Celica OUT predstavlja izhodno vrednost majoritetnih vrat.



Slika 2.6: Alternativna arhitektura QCA negatorja, ki je bolj robustna in odporna na zunanje vplive.



Slika 2.7: QCA majoritetna vrata kot osnovni gradnik za gradnjo kompleksnih QCA struktur.

Funkcijo majoritetnih vrat zapišemo z izrazom

$$MV = AB + AC + BC, \quad (2.1)$$

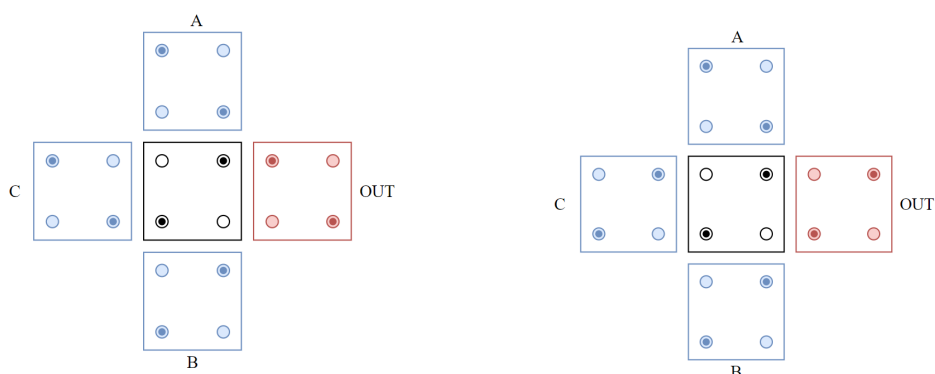
pri čemer MV predstavlja izhodno vrednost majoritetnih vrat, vrednosti A , B in C pa vhodne vrednosti majoritetnih vrat.

Majoritetna vrata lahko opravljajo eno izmed dveh osnovnih Boolovih logičnih funkcij — AND ali OR funkcijo. Kot smo že omenili, predstavlja eden izmed vhodov fiksno polarizirano celico. V primeru, da sta vhodni celici A in B glede na logično vrednost poljubni, C pa fiksno polarizirana celica na logično ničlo ali enico, dobimo naslednji dve možnosti:

$$C = 0 \rightarrow AND : MV = AB + A0 + B0 = AB, \quad (2.2)$$

$$C = 1 \rightarrow OR : MV = AB + A1 + B1 = A + B. \quad (2.3)$$

Majoritetni vrati, ki izvajata AND in OR funkcijo, sta prikazani na sliki 2.8.



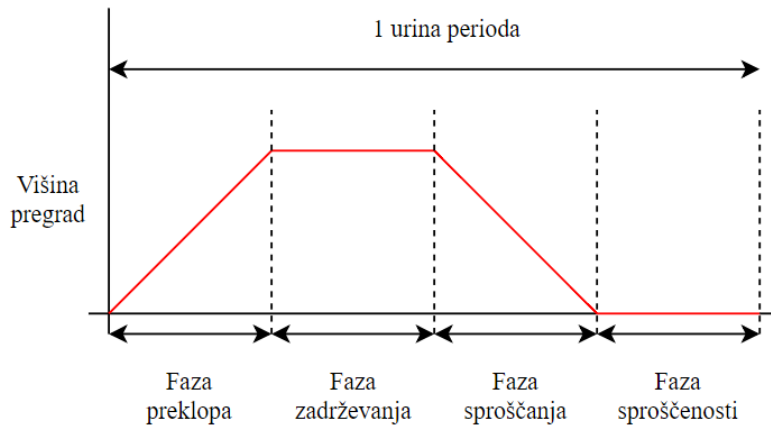
(a) QCA majoritetna vrata, ki opravljajo logično funkcijo AND. Celica C je polarizirana na logično ničlo.

(b) QCA majoritetna vrata, ki opravljajo logično funkcijo OR. Celica C je polarizirana na logično enico.

Slika 2.8: Majoritetni vrati, ki izvajata logični funkciji AND in OR.

2.3 Ura v QCA

Urin signal v QCA strukturo vpeljemo z namenom kontroliranja toka podatkov (angl. *data flow*) skozi strukturo [13] in za regeneriranje moči signala [14]. V primeru dolge linije lahko pride do izgube moči signala, zato je potreben mehanizem, ki lahko signal v QCA strukturi obnavlja oz. ojača. Urin signal v QCA strukturah se od urinega signala v CMOS tehnologiji močno razlikuje. QCA ura je realizirana s pomočjo električnega polja. Ena urina perioda v QCA je sestavljena iz štirih različnih urinih faz, ki jim zaporedno rečemo *faza preklopa*, *faza zadrževanja*, *faza sproščanja* in *faza sproščenosti*. Vse štiri urine faze so prikazane na sliki 2.9. QCA strukturo tako lahko razdelimo v štiri različne skupine celic, vsaka izmed skupin pa je naenkrat lahko v samo eni izmed omenjenih štirih urinih faz.



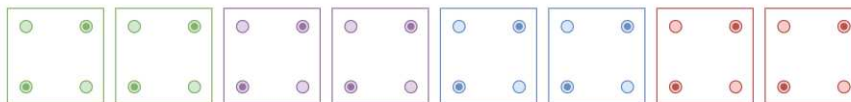
Slika 2.9: Graf, ki ponazarja potek urinih faz znotraj ene urine periode v QCA.

Kaj je pravzaprav namen ure v QCA strukturah? Ura kontrolira pregrade med kvantnimi pikami v QCA celicah. Učinek pregrad v QCA celici dosežemo z uporabo električnega polja. Na ta način se v QCA celicah simulirajo pregrade med kvantnimi pikami, ki omogočijo ali neomogočijo tuneliranje elektronov med kvantnimi pikami. Pregrade se dvigajo ali spuščajo glede na urino fazo, v kateri je v nekem trenutku določeno področje celic strukture. Ko so pregrade spuščene, lahko elektroni prosto prehajajo med kvantnimi pikami v celici, ko so pregrade dvignjene, pa pridemo v eno izmed dveh možnih polariziranih stanj. V tem trenutku je elektronom v celicah gibanje med kvantnimi pikami onemogočeno [7]. Urine faze imajo v QCA naslednji pomen [15]:

- *faza preklopa* — pregrade se v tunelih QCA celice začnejo dvigati; celice prehajajo v polarizirana stanja;
- *faza zadrževanja* — pregrade v tunelih so dvignjene; celice so v fiksni polariziranih stanjih; na ta način delujejo kot gonilniki za sosednje celice;
- *faza sproščanja* — pregrade v tunelih se začnejo spuščati; celice prehajajo v nepolarizirana stanja;
- *faza sproščenosti* — pregrade v tunelih so spuščene; celice so v nepolariziranih stanjih, elektroni pa lahko prosto prehajajo med kvantnimi pikami v celicah;

Za uspešno propagiranje signala skozi QCA strukturo morajo biti fizično sosednje

skupine celic v sosednjih urinih fazah. Primer linije, ki je razdeljena v štiri različne urine faze, je prikazan na sliki 2.10.



Slika 2.10: QCA linija, pri kateri sta po dve sosednji celici v isti urini fazi, kar je ponazorjeno z različnimi barvami celic. Fizično sosednje skupine celic morajo biti v sosednjih urinih fazah.

Velikosti področij celic z isto urino fazo so glede na QCA strukturo lahko zelo različne. V kolikor imamo redko strukturo (npr. dolga linija brez okoliških celic), lahko faze zajemajo večje število celic, saj ne prihaja do motenj iz okolja, ki bi lahko podrle polarizirana stanja celic. V primeru, ko imamo gosto strukturo z veliko linijami, majoritetnimi vrati ali negatorji, mora biti število celic v posamezni urini fazi manjše, saj je struktura zaradi gostote postavitve bolj podvržena k motnjam iz okolice. Napisanega pravila za število celic v posamezni urini fazi ni. Za število celic v posamezni urini fazi se odločimo glede na kompleksnost posameznega dela strukture. Kjer je struktura gosto posejana s celicami in zelo kompleksna, bo število celic v posamezni urini fazi majhno, kjer pa je struktura redka in preprosta, pa je lahko število celic v posamezni fazi večje (npr. tudi deset celic in več) [13].

2.4 Programsko orodje QCADesigner

QCADesigner je referenčno javno dostopno programsko orodje za gradnjo in simulacijo modelov QCA struktur, razvito na univerzi v Calgaryu [16]. Spisano je v programskem jeziku C/C++, licencirano pa s strani GNU javne licence (angl. *GNU public license*) za odprtokodno programsko opremo [17]. Za gradnjo struktur je na voljo grafični vmesnik, ki je bil razvit s pomočjo GTK grafičnih knjižnic.

Simuliranje QCA struktur z orodjem QCADesigner temelji na dveh različnih metodah. Prva je metoda bistabilne aproksimacije, druga pa metoda, ki temelji na uporabi koherentnega vektorja.

2.4.1 Bistabilna aproksimacija

Pri metodi bistabilne aproksimacije je vsaka QCA celica predstavljena kot preprost dvostanjski sistem. Metoda temelji na medsebojni interakciji celic, ki se nahajajo v določenem radiju opazovane celice. Vpliv na polarizacijo opazovane celice i imajo vse sosednje celice j , ki se nahajajo v t.i. radiju efekta (angl. *radius of effect*) R opazovane celice. Za vsako opazovano celico i lahko njen dvostanjski model matematično opišemo s Hamiltonovo matriko [11], ki je prikazana v enačbi (2.4).

$$H_i = \sum_j \begin{bmatrix} -\frac{1}{2}P_j E_{i,j}^k & -\gamma \\ -\gamma & \frac{1}{2}P_j E_{i,j}^k \end{bmatrix} \quad (2.4)$$

V enačbi (2.4) imajo simboli naslednji pomen. $E_{i,j}^k$ predstavlja energijo (angl. *kink energy*) med dvema celicama i in j . Ponazarja ceno v energiji med nasprotnima polarizacijama dveh sosednjih celic. P_j predstavlja polarizirano stanje celice j (logična ničla ali enica), γ pa predstavlja energijo tuneliranja. Za opazovano celico i je vrednost Hamiltonove matrike enaka vsoti matrik sosednjih celic j , ki se nahajajo v radiju efekta R . Z izračunano Hamiltonovo matriko za opazovano celico i lahko nato izračunamo polarizacijo opazovane celice i po časovno neodvisni Schrödingerjevi enačbi (2.5) [17].

$$H_i \psi_i = E_i \psi_i \quad (2.5)$$

S pomočjo Jacobijevega algoritma se iz enačbe (2.5) izračunajo lastne vrednosti E_i in lastni vektorji ψ_i Hamiltonove matrike, pri čemer je H_i Hamiltonova matrika opazovane celice i , ψ_i predstavlja polarizirano stanje celice i , E_i pa je energija, ki se povezuje s stanjem ψ_i . Algoritem uredi stanja ψ_i glede na pripadajočo energijo v naraščajočem zaporedju. Stanje na prvem mestu je tako stanje z najmanjšo energijo. To stanje se zato izbere kot stanje, ki pravilno ponazori polarizacijo celice i .

Na ta način metoda bistabilne aproksimacije izračuna stanje vsake celice v strukturi, dokler celotna struktura ne konvergira v izbrano polarizirano stanje. Ko struktura pride do tega stanja, se zabeleži izhodna vrednost, izberejo se nove vhodne vrednosti in simulacija se ponovi [17].

2.4.2 Koherenčni vektor

Metoda s koherenčnim vektorjem prav tako predstavlja celico kot preprost dvostanjski sistem, ki je predstavljen s Hamiltonovo matriko iz enačbe (2.4). Radij efekta R ponovno

pove obseg vpliva na trenutno opazovano celico. Metoda temelji na matriki gostote (angl. *density matrix*) ρ , ki modelira izgubo moči v QCA strukturah. Pri metodi koherentnega vektorja je vsaka QCA celica predstavljena z dvema različnima stanjema. Prvo stanje predstavlja dejansko stanje celice, ki temelji na matriki gostote, drugo stanje pa predstavlja idealno stabilno stanje (t.j. stanje z najmanjšo energijo), ki naj bi ga QCA celica dosegla. Idealno stabilno stanje je odvisno od polarizacije sosednjih celic in tuneliranja elektronov v opazovani QCA celici [18].

Koherentni vektor $\vec{\lambda}$ je vektorska predstavitev matrike gostote ρ oz. dejansko stanje posamezne celice. Definiran je kot $\vec{\lambda} = (\lambda_x, \lambda_y, \lambda_z)$, pri čemer je vsaka izmed komponent koherentnega vektorja definirana kot $\lambda_i = Tr\{\rho\sigma_i\}$. σ_i predstavlja vsako izmed Paulijevih matrik ($\sigma_x, \sigma_y, \sigma_z$), operacija Tr pa sled matrike. Polarizacija trenutno opazovane celice i je z komponenta koherentnega vektorja [11]. Dobimo jo preko enačbe (2.6).

$$P_i = -Tr\{\rho\sigma_z\} = -\lambda_z \quad (2.6)$$

Faktorji, ki vplivajo na idealno stabilno stanje celice i so opisani s Hamiltonovo matriko, ki je bila definirana že v enačbi (2.4). Koherentni vektor idealnega stabilnega stanja je označen z $\vec{\lambda}_{ss}$. Izračunamo ga s pomočjo energijskega vektorja Γ . Vektor Γ predstavlja energijsko okolje celice i , vključujoč vse vplive celic iz efektivnega radija R , \hbar pa ponazarja reducirano Planckovo konstanto. Vektor $\vec{\Gamma}$ je izračunan na podlagi enačbe (2.7).

$$\vec{\Gamma} = \frac{1}{\hbar} \left[-2\gamma, \quad 0, \quad \sum P_j E_{i,j}^k \right] \quad (2.7)$$

Koherentni vektor idealnega stabilnega stanja je nato podan z enačbo (2.8)

$$\vec{\lambda}_{ss} = -\frac{\vec{\Gamma}}{|\vec{\Gamma}|} \tanh \eta_{th}, \quad (2.8)$$

pri čemer je člen η_{th} definiran po enačbi (2.9). k_B predstavlja Boltzmanovo konstanto, T pa temperaturo.

$$\eta_{th} = \frac{\hbar|\vec{\Gamma}|}{2k_B T} \quad (2.9)$$

Zatem lahko izračunamo dejansko stanje opazovane QCA celice po enačbi (2.10).

$$\frac{\partial}{\partial t} \vec{\lambda} = \vec{\Gamma} \times \vec{\lambda} - \frac{1}{\tau} (\vec{\lambda} - \vec{\lambda}_{ss}) \quad (2.10)$$

Prvi del enačbe ($\vec{\Gamma} \times \vec{\lambda}$) predstavlja idealne razmere za prehod v novo stanje (brez disipacije toplote in energije). V praksi takih razmer ni in zato vedno prihaja do disipacije, kar ponazori drugi del enačbe ($-\frac{1}{\tau}(\vec{\lambda} - \vec{\lambda}_{ss})$). Drugi del enačbe modelira dejanski prehod, pri čemer upošteva tudi disipacijo toplote in energije. τ predstavlja relaksacijski čas, ki je odvisen od izbrane tehnologije in materialov. Nižja kot je vrednost τ , večja je disipacija toplote QCA celice in obratno [18]. Za vsak časovni korak se Γ in λ_{ss} izračunata posebej. Na ta način dobimo vrednost koherentnega vektorja λ in posledično vrednost polarizacije trenutno opazovane celice i v vsakem časovnem koraku.

Metoda s koherentnim vektorjem za svoje izvajanje uporablja natančnejši model kot metoda bistabilne aproksimacije. Zaradi večje natančnosti modela je čas izvajanja simulacije na podlagi koherentnega vektorja daljši od izvajanja bistabilne simulacije, dobimo pa zato natančnejše rezultate [17]. Dodatna razlika je tudi v tem, da metoda s koherentnim vektorjem uporablja časovno odvisno simulacijo, medtem ko je bistabilna simulacije ne uporablja [11].

3 Reverzibilnost

Reverzibilnost oz. reverzibilno računanje (angl. *reversible computing*) je v primerjavi s standardnim računanjem (angl. *forward computing*) relativno ozka veja v računalništvu. Poraja se vprašanje, zakaj bi sploh računali “nazaj”, medtem ko večina računalništva temelji na standardnem računanju, ki ga vsi dobro poznamo. Veliko znanstvenikov je sprva izražalo dvome o reverzibilnem računanju, saj so računske operacije fizikalno gledano nereverzibilne, ker se entropija vedno povečuje [19]. To je eden izmed ključnih razlogov, zakaj je reverzibilnost v domeni računalništva deležna počasnejšega razvoja.

3.1 Osnove reverzibilnosti

Na tem mestu se omejimo na pojem *reverzibilnega procesa*. Reverzibilen proces je proces, ki je časovno obrnljiv. Drugače povedano, reverzibilen proces je proces, čigar predhodno stanje lahko vedno rekonstruiramo iz trenutnega stanja [3]. Za diskretni čas lahko matematično to zapišemo kot

$$S(t - 1) = f^{-1}(E(t), S(t)), \quad (3.1)$$

pri čemer je $S(t)$ trenutno stanje sistema na časovnem koraku t , $E(t)$ dogodek v tem koraku t , f^{-1} pa reverzibilna funkcija, ki nam vrne prejšnje stanje sistema $S(t-1)$ v časovnem koraku $t-1$. Za zvezen reverzibilen proces bi to funkcijo zapisali z izrazom

$$S(t_{i-1}) = f^{-1}(E(t_i), S(t_i)). \quad (3.2)$$

Lahko bi rekli, da imamo pri reverzibilnem računanju možnost hranjenja zgodovine, saj lahko prejšnje stanje sistema vedno dobimo s pomočjo reverzibilne funkcije f^{-1} . To velja izključno za reverzibilne procese.

3.2 Fizična reverzibilnost

Da bomo lahko dobro razumeli pojem logične reverzibilnosti, si najprej pogledajmo pojem fizične reverzibilnosti. Fizična reverzibilnost je eden izmed osnovnih zakonov narave [20]. Fizična reverzibilnost ali boljše rečeno fizična ireverzibilnost se v klasičnih elektronskih vezjih izraža v povečanju entropije, kar se manifestira v obliki izgube energije in oddajanja toplote.

Po Landauerjevem principu se mora vsaka logična sprememba informacije kompenzirati v obliki povečanja fizične entropije v okolju. Primer je dovod energije v računalnik, ki ob računanju spreminja vrednosti bitov, del te energije pa se nato v okolje sprosti kot toplota [21]. Preprost primer je izvajanje dvovhodne logične funkcije, katere izhod je enobiten. Tukaj gre praktično za izbris enega bita (dva bita na vходу in samo en bit na izhodu). Odvečna energija izbrisanega bita se tako manifestira v obliki sproščene toplote v okolje [22].

Fizična reverzibilnost v računalništvu predvideva prenos oz. manipulacijo informacij na bitnem nivoju brez sproščanja toplote v okolje. Takemu procesu rečemo *adiabatni proces*. V kolikor bi reverzibilne računske operacije izvajali na primerni strojni opremi in bi vse korake ireverzibilnega procesa nadomestili z reverzibilnimi koraki, bi lahko dosegli nezvečanje fizične entropije in na ta način reverzibilnost računanja [21].

3.3 Logična reverzibilnost

Pojem *logična reverzibilnost* se v računalništvu nanaša na funkcijo, ki se uporablja za izvajanje reverzibilne računske operacije. Da je funkcija logično reverzibilna, morata biti izpolnjena dva pogoja [23]:

x	$NEG(x)$	x_1	x_2	$AND(x_1, x_2)$	x_1	x_2	$OR(x_1, x_2)$
0	1	0	0	0	0	0	0
1	0	0	1	0	0	1	1
		1	0	0	1	0	1
		1	1	1	1	1	1

Tabela 3.1: Pravilnostne tabele za logične funkcije negacije (NEG), konjunkcije (AND) in disjunkcije (OR).

- funkcija mora biti bijektivna,
- funkcija mora imeti enako število vhodov in izhodov.

Iz bijektivnosti funkcije izpeljemo, da lahko za vsak izhod enolično določimo vhod, ki je bil uporabljen za pridobitev izhoda. Funkciji vedno določimo tip v obliki (n, k) , pri čemer se n glasi na število vhodov, k pa na število izhodov. Za izpolnitev drugega pogoja logične reverzibilnosti mora tako veljati relacija $n = k$.

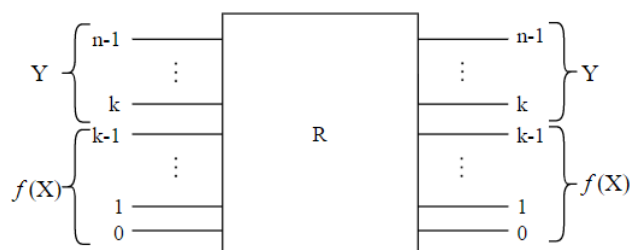
Za osnovo vzemimo preproste Boolove logične funkcije kot so npr. negacija, konjunkcija in disjunkcija. Pravilnostne tabele vseh treh operacij so prikazane v tabelah 3.1. Opazimo, da negacija zadosti obema pogojema za logično reverzibilnost. Ima en vhod in en izhod, prav tako pa opazimo, da je bijektivna. Pri funkcijah konjunkcije in disjunkcije opazimo, da noben izmed pogojev za logično reverzibilnost ni izpolnjen. Iz izhodov ne moremo enolično določiti vhodov, prav tako pa sta obe funkciji tipa $(2, 1)$. Za izpolnitev drugega pogoja logične reverzibilnosti ($n = k$) funkcijam dodamo t.i. *konstante vhode* (angl. *constant inputs*) in *odvečne izhode* (angl. *garbage outputs*) [24]. Ob dodajanju dodatnih vhodov in izhodov moramo paziti, da poleg izpolnitve drugega pogoja logične reverzibilnosti ($n = k$) še vedno zadoščamo izpolnitvi pogoja bijektivnosti funkcije.

3.4 Reverzibilne logične funkcije

V prejšnjem razdelku smo ugotovili, da negacija predstavlja reverzibilno logično funkcijo tipa $(1, 1)$. Poznamo tudi reverzibilne logične funkcije, kjer je število vhodov in izhodov večje. V nadaljevanju bosta predstavljeni Feynmanova in Fredkinova reverzibilna logična funkcija.

Reverzibilne logične funkcije implementiramo s pomočjo *reverzibilnih logičnih vrat* (angl. *reversible logic gates*). Reverzibilna logična vrata sestavljajo *linije*, ki se delijo na

dve skupini in sicer na *procesne linije* in *pomnilne linije*. Pomnilne linije (angl. *temporary storage*) samo ohranjajo vrednost enega izmed vhodov in ga nespremenjenega prenesejo na izhod, na procesnih linijah (angl. *process lines*) pa se izvaja računanje z vrednostmi iz vhoda [25]. Na izhod tako pride nova vrednost, ki je rezultat procesiranja na procesnih linijah. Primer je viden na sliki 3.1, povzeti po viru [25]. Zgornjih $n - k$ linij samo prenaša informacijo o vhodu na izhod, označene pa so z Y . Tem izhodom pravimo odvečni izhodi. Spodnjih k linij, označenih z $f(X)$ predstavlja linije, ki izvajajo logične operacije nad vhodi v reverzibilnih vratih. Na tem mestu omenimo še, da linije v reverzibilnih logičnih funkcijah nimajo nobene zveze s QCA linijami. Gre le za enako poimenovanje dveh različnih pojmov.



Slika 3.1: Shema reverzibilnih logičnih vrat, na kateri so prikazane procesne ($f(X)$) in odvečne (Y) linije.

3.4.1 Feynmanova funkcija

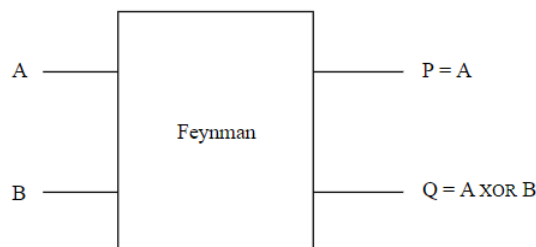
Feynmanova funkcija predstavlja funkcijo tipa $(2, 2)$ [26]. Feynmanovo logično funkcijo implementirajo Feynmanova vrata. Vhoda v Feynmanova vrata predstavljata liniji A in B , izhodni liniji pa sta poimenovani P in Q . Na sliki 3.2 opazimo, da je izhod P odvečen izhod, saj vsebuje le kopijo vhodne vrednosti linije A . Izhodna linija Q predstavlja dejanski procesni del, v katerega vstopata vhoda A in B . Izhod Q predstavlja rezultat logične funkcije XOR nad vhodoma A in B . Iz pravilnostne tabele 3.2 Feynmanove funkcije vidimo, da zadosti obema pogojema za logično reverzibilnost. Število vhodov je enako številu izhodov, prav tako pa lahko vsak vhod enolično preslikamo v izhod.

Drugo ime za Feynmanova vrata je *controlled NOT*. Razlog za to je negacija vhoda B pod pogojem, da je vhodna vrednost $A = 1$. V primeru, da je $A = 0$, se vrednost B ne spreminja. Feynmanova vrata so natančneje razložena v poglavju 6. V nadaljevanju

A	B	P	Q
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

Tabela 3.2: Pravilnostna tabela Feynmanove logične funkcije.

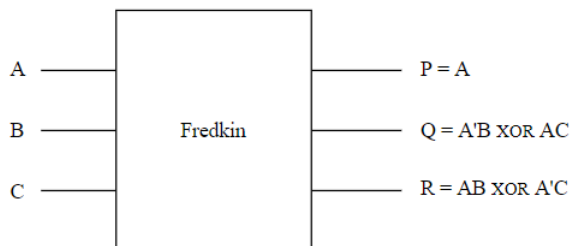
jih uporabimo kot osnovo za naš reverzibilni pretvornik iz binarne v Grayevo kodo.



Slika 3.2: Shema Feynmanovih vrat. Izhodna linija P predstavlja odvečni izhod, procesiranje pa se izvede na izhodni liniji Q .

3.4.2 Fredkinova funkcija

Fredkinova funkcija je primer reverzibilne logične funkcije tipa $(3, 3)$. Funkcijo implementirajo Fredkinova vrata, pri čemer so vhodi A , B in C , izhodi pa so P , Q in R [26]. Fredkinova vrata predstavljajo poln funkcijski nabor, kar pomeni, da lahko s Fredkinovimi vrati realiziramo katerokoli drugo logično funkcijo. Shema Fredkinovih vrat je prikazana na sliki 3.3. Opazimo, da imajo le eno pomnilno linijo $P = A$, ostali dve izhodni liniji Q in R pa sta procesni. Drugo ime za Fredkinova vrata je *controlled SWAP* [24]. Ko je vrednost vhoda $A = 1$ se vrednosti B in C zamenjata, kar je vidno na izhodih Q in R . Pravilnostna tabela Fredkinove funkcije je vidna v tabeli 3.3. Iz tabele je razvidno, da Fredkinova funkcija zadosti obema pogojema za logično reverzibilnost. Število vhodov je enako številu izhodov, prav tako pa se vsak vhod enolično preslika v izhod.



Slika 3.3: Shema Fredkinovih vrat. Izhodna linija P predstavlja odvečen izhod, procesiranje pa se izvede na izhodnih linijah Q in R .

A	B	C	P	Q	R
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	0	0
0	1	1	0	0	1
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	1	0	1
1	1	1	1	1	1

Tabela 3.3: Pravilnostna tabela Fredkinove logične funkcije.

3.5 Uporabnost reverzibilnosti

Na tem mestu se vprašajmo, zakaj je reverzibilnost sploh koristna? Zakaj bi se posvečali reverzibilnosti, če lahko iste računske operacije že izvajamo z obstoječo, nereverzibilno tehnologijo?

Kot smo že omenili, je Landauer dokazal, da računanje z obstoječimi logičnimi vrati neizogibno vodi v izgubo energije v elektronskih vezjih [4], čemur bi se lahko izognili z uporabo računalnikov, ki jih sestavljajo logično in fizično reverzibilna vezja.

Moorov zakon pravi, da se je računska moč v zadnjih petdesetih letih približno podvojila na vsakih 18 mesecev [27], kar je temeljilo predvsem na miniaturizaciji tranzistorjev, ki sestavljajo elektronska vezja. To pomeni povečanje števila tranzistorjev na enoto površine v računalnikih, hitrejša časovna preklapljanja in obenem zvečanje disipa-

cije energije in toplote za vsak tranzistor. Ker so tranzistorji vedno manjši in disipacija energije raste, bo prej ali slej dosežena zgornja meja količine sproščene energije v okolje. Obenem to tudi pomeni, da bo nadaljnja miniaturizacija tranzistorjev praktično onemogočena.

Na tem mestu nastopi reverzibilnost, ki predstavlja način za nadaljnjo izboljšavo elektronskih naprav in nadaljnjo povečanje računske moči. Vir [28] govori o reverzibilnem računanju, ki bo omogočilo povečanje računske moči veliko preko omejitev trenutne tehnologije. V viru je razloženo, da je zmogljivost elektronske naprave omejena s številom operacij, ki se izvedejo na enoto sproščene energije v okolje. Pri procesiranju informacij od nas nič ne zahteva, da bi se energija morala sproščati v okolje. Po Landauerju je v trenutni tehnologiji razlog za to uporaba nereverzibilnih vezij, ki posledično vodijo v fizično nereverzibilnost, ta pa se manifestira v obliki sproščene toplote v okolje [29]. S primerno tehnologijo bi ob procesiranju informacij v elektronskih vezjih lahko omejili sproščanje energije v okolje oz. bi sproščanje energije v okolje ob računanju poljubno minimizirali. Na eni strani tako zmanjšujemo sproščanje energije v okolje, na drugi strani pa to odvečno energijo porabimo za izboljšanje signala v elektronskih vezjih. Na ta način bi lahko izboljšali robustnost signala in povečali odpornost na šume iz okolja in toplotni šum. Teoretično se lahko ob konstantni količini sproščene energije (če količino sproščene energije minimiziramo) število računskih operacij na enoto sproščene toplote približuje k neskončnosti. To dobro ponazori enačba (3.3) povzeta po viru [28]

$$R = F_e \times P_{diss} = \frac{N_{ops}}{E_{diss}} \times \frac{E_{diss}}{t}, \quad (3.3)$$

pri čemer R predstavlja učinkovitost, N_{ops} število opravljenih operacij v sklopu dela, t čas za opravljanje dela, E_{diss} sproščena energija med delom, F_e energijsko učinkovitost, P_{diss} pa povprečno moč, ki se sprosti ob delu. Če se opremo na prejšnje dejstvo, da sproščeno energijo minimiziramo, iz enačbe (3.3) vidimo, da število opravljenih operacij teoretično lahko narašča proti neskončnosti.

S pomočjo navedenih dejstev tako sklepamo, da bo učinkovitost in zmogljivost elektronskih vezij z uporabo reverzibilnosti krepko narasla preko meja zmogljivosti, ki jo predstavlja trenutna nereverzibilna tehnologija.

4 Grayeva koda

Najbolj razširjen številski sistem, ki se uporablja v računalništvu, je dvovrednostni binarni zapis. V njem lahko tako poljubno število, kot tudi poljuben podatek predstavimo z nizom znakov iz množice $\{0, 1\}$. Tako lahko z enim znakom predstavimo desetiški števili $0_{(10)}$ ali $1_{(10)}$, z uporabo dveh znakov pa se nam dolžina zapisa podvoji, predstavimo pa lahko števila od $0_{(10)}$ do $3_{(10)}$ itd. Na tem mestu omenimo, da se bomo v pričujočem delu omejili le na zapis števil in ne na zapis poljubnih podatkov. Prav tako v delu enačimo pojma binarnega zapisa in binarne kode.

Binarni zapis desetiškega števila je definiran z izrazom (4.1)

$$z_{(10)} = \sum_{i=0}^{n-1} z_i 2^i, \quad (4.1)$$

pri čemer $z_{(10)}$ predstavlja število v desetiškem zapisu, n število vseh znakov v njegovem binarnem zapisu, i indeks posameznega znaka v binarnem zapisu, zaporedje $z_{n-1} \dots z_0$ pa binarni zapis števila $z_{(10)}$ [30].

Binarni zapis ne ustreza vsem pogojem in zahtevam, ki se pojavljajo v računalniški industriji. Obstajajo izpeljanke binarnega zapisa, ki so na specifičnih področjih uporabe

veliko bolj uspešne. Eden izmed takih zapisov je *Grayeva koda* ali *reflektirana binarna koda*, ki jo predstavimo v pričujočem poglavju. Tudi tu enačimo pojma Grayeve kode in Grayevega zapisa.

4.1 Osnove Grayeve kode

Grayeva koda je poimenovana po Franku Grayu, raziskovalcu pri Bell Telephone Laboratories (sedaj Nokia Bell Labs) [31]. Patentirana je bila leta 1953 [32], uporabljala pa naj bi se že veliko prej — v 19. stoletju jo je uporabljal že Emile Baudot na področju telegrafije [33].

Grayeva koda je izpeljanka binarne kode. Zaloga vrednosti znakov, ki sestavljajo zapise števil, je enaka binarni zalogi vrednosti znakov $\{0, 1\}$. Pomembna razlika v primerjavi z binarno kodo je v tem, da so si zapisi po vrednosti sosednjih števil različni v samo enem znaku. Ta lastnost je s pridom uporabljena na različnih področjih, kot so npr. rotacijski in optični kodirniki v robotski industriji [34]. V nadaljevanju je za ločevanje obeh kod za binarni zapis uporabljen indeks številskega sistema 2 (npr. $10_{(2)} = 2_{(10)}$), pri Grayevem zapisu pa je indeks številskega sistema enak G (npr. $11_{(G)} = 2_{(10)}$). Indeks številskega sistema $_{(10)}$ označuje desetiški zapis, katerega vsi dobro poznamo.

Težava binarnega zapisa števil, še posebej, ko se ta uporablja v mehanskih napravah, je v tem, da se mora pri prehodih med sosednjimi vrednostmi števil v zapisu spremeniti več kot en znak [35]. Tak primer je npr. prehod iz vrednosti $3_{(10)}$ ($011_{(2)}$) v $4_{(10)}$ ($100_{(2)}$), kjer se spremeni vrednost vseh treh znakov. Problem se pojavi pri branju nove vrednosti, saj lahko pride do branja le-te, še preden so vsi potrebni znaki spremenili svojo vrednost. V omenjenem primeru tako lahko pride do branja katerekoli vrednosti od $000_{(2)}$ do $111_{(2)}$. Z Grayevo kodo, kjer se za vsako spremembo zaporedne vrednosti spremeni le en znak, je možno število napak pri branju podatkov minimizirati. Pri prehodu iz vrednosti $3_{(10)}$ ($010_{(G)}$) v $4_{(10)}$ ($110_{(G)}$) v Grayevi kodi je potrebna sprememba le prvega znaka na levi. Tako je lahko tudi vrednost prebrane vrednosti kvečjemu le $3_{(10)}$ ali $4_{(10)}$. V najslabšem primeru se vrednost prebranega podatka od pravilne vrednosti razlikuje le za 1. V tem pravzaprav tiči najbolj pomembna lastnost Grayeve kode in razlika v primerjavi z binarno kodo — kodni zapisi sosednjih števil so si podobni in se razlikujejo le v enem znaku [31].

4.2 Postopek za pretvorbo iz binarne v Grayevo kodo

Pretvorba iz binarne kode v Grayevo kodo je preprosta, saj z logičnega vidika vključuje ponavljanje le ene operacije — XOR. Postopek pretvorbe poteka v smeri od najbolj pomembnega bita (angl. *most significant bit* — MSB) proti najmanj pomembnemu bitu (angl. *least significant bit* — LSB).

Matematična pretvorba, pri čemer je g_{n-1} MSB v Grayevi kodi, b_{n-1} pa MSB v binarni kodi, je prikazana v izrazih od (4.2) do (4.5). V izrazu (4.2) je prikazan binarni zapis, ki predstavlja izhodišče za pretvorbo. V prvem koraku prepisemo MSB iz binarne kode v MSB Grayeve kode, kar je prikazano v izrazu (4.3). V nadaljevanju pretvorimo preostale znake binarne kode v Grayevo kodo po izrazu (4.4). Na koncu vse izračunane znake g_i zložimo v ciljno Grayevo kodo na način, ki je prikazan v izrazu (4.5).

$$B = b_{n-1}b_{n-2} \dots b_1b_0, \quad (4.2)$$

$$g_{n-1} = b_{n-1}, \quad (4.3)$$

$$g_i = b_{i+1} \oplus b_i, i = n-2, n-3 \dots, 0, \quad (4.4)$$

$$G = g_{n-1}g_{n-2} \dots g_1g_0. \quad (4.5)$$

Z besedami pretvorbo opišemo na sledeči način [36]:

- prvi znak oziroma MSB je pri binarni in Grayevi kodi vedno enak, zato se ga preprosto prepíše;
- preostale znake Grayeve kode (razen prvega znaka) dobimo z XOR operacijo istoletnega znaka v binarni kodi in njegovega levega predhodnika (znaka z večjim indeksom);

Primer pretvorbe je prikazan v tabeli 4.1, kjer pretvorimo število $9_{(10)}$ iz binarne kode v Grayevo. Prvi znak Grayeve kode je enak prvemu znaku binarne kode, drugi znak Grayeve kode je enak rezultatu XOR operacije med prvim in drugim znakom binarne kode, tretji znak Grayeve kode je enak rezultatu XOR operacije med drugim in tretjim znakom binarne kode itd.

b_{n-1}	\dots	\dots	b_0	
1	0	0	1	Binarna koda
\downarrow	\searrow	\downarrow	\searrow	\downarrow
1	$1 \oplus 0$	$0 \oplus 0$	$0 \oplus 1$	
\downarrow	\downarrow	\downarrow	\downarrow	
1	1	0	1	Grayeva koda
g_{n-1}	\dots	\dots	g_0	

Tabela 4.1: Primer pretvorbe iz binarne v Grayevo kodo. Prvi znak se prepíše, preostale znake Grayeve kode pa dobimo kot rezultat XOR operacije sosednjih znakov binarne kode.

4.3 Postopki za generiranje Grayeve kode

V pričujočem razdelku predstavimo dva postopka za generiranje Grayeve kode — postopek z reflektiranjem in postopek z razširjanjem. Postopek z reflektiranjem je definiral Frank Gray že v svojem patentu [32], postopek z razširjanjem pa je povzet po viru [37].

4.3.1 Generiranje z reflektiranjem

Najkrajši zapis binarne kode je sestavljen le iz enega znaka — 0 ali 1. Zaradi tega lahko z enoznakovnim zapisom predstavimo le števili $0_{(10)}$ ali $1_{(10)}$. Enoznakovni zapis je zato v Grayevi kodi enak binarni kodi. V kolikor želimo preiti na predstavitev večjih vrednosti števil, moramo zapis razširiti na več znakov. S povečanjem dolžine zapisa na dva znaka pridobimo možnost predstavitve števil od $0_{(10)}$ do $3_{(10)}$ (binarno: $00_{(2)}$, $01_{(2)}$, $10_{(2)}$, $11_{(2)}$). Cilj Grayeve kode je razlikovanje zapisa sosednjih števil v le enem znaku, čemur se binarna koda ne podreja. Z razširitvijo na dva znaka se tako že pojavi problem binarne kode — razlikovanje nekaterih sosednjih števil v več kot enem znaku. To se vidi v zaporednih številih $1_{(10)}$ in $2_{(10)}$, ki se v binarni kodi $01_{(2)}$ in $10_{(2)}$ razlikujeta v obeh znakih.

Postopek generiranja z reflektiranjem je po viru [31] sledeč. Ničlo in enico zapišemo v skrajno levem stolpcu na sliki 4.1. V naslednjem koraku, ki je viden v drugem stolpcu na sliki 4.1, ničlo in enico najprej prepíšemo v zgornji del drugega stolpca, nato pa ju je potrebno zrcaliti navpično navzdol — ju reflektirati. Da zapis razširimo še na števili $2_{(10)}$ in $3_{(10)}$ v smislu Grayeve kode, pred zgornjo polovico znakov zapišemo ničlo, pred

spodnjo polovico znakov pa enico (vidno v tretjem stolpcu na sliki 4.1 v rdeči barvi).

$$\begin{array}{ccc}
 & 0 & \mathbf{00} \ (0_{(10)}) \\
 0 & \longrightarrow \frac{1}{1} & \longrightarrow \frac{\mathbf{01}}{11} \ (1_{(10)}) \\
 1 & & \mathbf{11} \ (2_{(10)}) \\
 & 0 & \mathbf{10} \ (3_{(10)})
 \end{array}$$

Slika 4.1: Postopek generiranja Grayeve kode za števila od $0_{(10)}$ do $3_{(10)}$. V prvem koraku zapišemo števili 0 in 1 v Grayevi kodi (v tem primeru sta še enaka binarni kodi) ter ju zrcalimo (reflektiramo). Nato pred zgornjo polovico znakov zapišemo ničlo, pred spodnjo polovico znakov pa enico (v rdeči barvi). Tako dobimo Grayevo kodo za števila $0_{(10)} - 00_{(G)}$, $1_{(10)} - 01_{(G)}$, $2_{(10)} - 11_{(G)}$ in $3_{(10)} - 10_{(G)}$, razširjeno na dva znaka.

Na ta način smo dobili zaporeden zapis števil od $0_{(10)}$ do $3_{(10)}$ v Grayevi kodi. Opazimo, da se sosednja števila razlikujejo le v enem znaku. Zapis je prav tako različen od binarnega, saj vrednost $11_{(G)}$ sedaj predstavlja število $2_{(10)}$ (v binarnem zapisu $10_{(2)}$), vrednost $10_{(G)}$ pa število $3_{(10)}$ (v binarnem zapisu $11_{(2)}$).

Enak postopek bi sledil za pridobitev števil od $0_{(10)}$ do $7_{(10)}$. Števila od $0_{(10)}$ do $3_{(10)}$ v Grayevi kodi ponovno zapišemo v skrajno levi stolpec na sliki 4.2, ga prenesemo v zgornji del drugega stolpca in slednjega reflektiramo (zrcalimo v spodnji del stolpca). Nato ponovimo postopek dodajanja ničel in enic. Pred zgornjo polovico števil zapišemo ničlo, pred spodnjo polovico pa enico (prikazano v tretjem stolpcu na sliki 4.2 v rdeči barvi). Tako pridobimo zapis števil od $0_{(10)}$ do $7_{(10)}$ v Grayevi kodi, kjer so si zaporedna števila ponovno različna le v enem znaku.

Na ta način je Gray zadostil pogoju, da so si sosednja števila različna le v enem znaku. Še več, opazimo lahko, da sta si zgornja in zrcaljena spodnja polovica zapisa različni samo v MSB. Pri zgornji polovici je MSB enak 0, pri spodnji polovici pa je MSB enak 1.

		00		000	$(0_{(10)})$
		01		001	$(1_{(10)})$
00		11		011	$(2_{(10)})$
01	→	10	→	010	$(3_{(10)})$
11		10		110	$(4_{(10)})$
10		11		111	$(5_{(10)})$
		01		101	$(6_{(10)})$
		00		100	$(7_{(10)})$

Slika 4.2: Postopek generiranja Grayeve kode za števila od $4_{(10)}$ do $7_{(10)}$. V prvem koraku zapišemo števila od 0 do 3 v Grayevi kodi, ter jih zrcalimo (reflektiramo). Nato pred zgornjo polovico znakov zapišemo ničlo, pred spodnjo polovico znakov pa enico (v rdeči barvi). Tako dobimo Grayevo kodo za števila od $4_{(10)}$ do $7_{(10)}$, razširjeno na 3 znake.

4.3.2 Generiranje z razširjanjem

Bistvo generiranja Grayeve kode z razširjanjem je podvojitve vseh zapisov vhodne Grayeve kode, nato pa razširitev podvojenih zapisov z zaporednimi vrednostmi iz neskončnega vektorja A , kjer se prve štiri vrednosti vektorja ponavljajo. Vektor A ponazorimo z izrazom

$$A = [0 \ 1 \ 1 \ 0 \ 0 \ 1 \ 1 \ 0 \ \dots \ 0 \ 1 \ 1 \ 0]. \quad (4.6)$$

Postopek z razširjanjem Grayeve kode iz $m - 1$ mest na m mest, pri čemer m predstavlja število znakov zapisa števila v izhodni Grayevi kodi, M pa moč množice zaloge vrednosti izhodne Grayeve kode ($M = 2^m$) oz. obseg števil izhodne Grayeve kode, je po viru [37] podan z naslednjimi koraki. Na začetku imamo vhodno množico števil v Grayevi kodi, števila pa so označena od G_0 do $G_{\frac{M}{2}-1}$, pri čemer je vsako število vhodne množice sestavljeno iz $m - 1$ znakov. Prvi korak razširjanja je zapis števil vhodne Grayeve kode po izrazu

$$Gray_{m-1} = [G_0 \ G_1 \ \dots \ G_{\frac{M}{2}-1}]. \quad (4.7)$$

Naslednji korak je podvojitve vseh zapisov vhodnih števil, kar je predstavljeno z izrazom

$$Gray'_m = \left[G_0 \quad G_0 \quad G_1 \quad G_1 \quad \dots \quad G_{\frac{M}{2}-1} \quad G_{\frac{M}{2}-1} \right]. \quad (4.8)$$

Na koncu vse podvojene zapise števil razširimo z zaporednimi vrednostmi iz vektorja A , kar zapišemo z izrazom

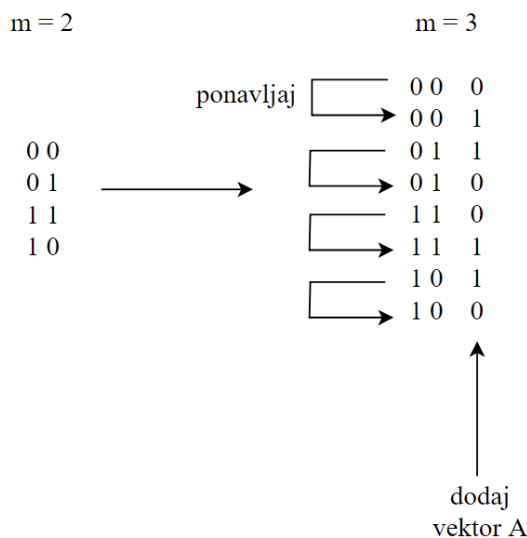
$$Gray_m = \left[G_0 A[0] \quad G_0 A[1] \quad \dots \quad G_{\frac{M}{2}-1} A[M-2] \quad G_{\frac{M}{2}-1} A[M-1] \right]. \quad (4.9)$$

Primer generiranja z razširjanjem je prikazan na sliki 4.3. Gre za razširitev Grayeve kode dolžine $m = 2$ na dolžino $m = 3$, pri čemer m predstavlja število znakov v kodnem zapisu. Kodni zapis dolžine $m = 2$ vsebuje kode $00_{(G)}$, $01_{(G)}$, $11_{(G)}$ in $10_{(G)}$, desetiško gledano pa so to števila od $0_{(10)}$ do $3_{(10)}$. Prikazan je v levem stolpcu na sliki 4.3. Cilj razširitve je podvojitev obsega števil kodnega zapisa oz. pridobitev števil od $0_{(10)}$ do $7_{(10)}$. Kot smo že omenili, je potrebno vse kode podvojiti (glej izraz (4.8)), kar nas privede do osmih kodov ($00_{(G)}$, $00_{(G)}$, $01_{(G)}$, $01_{(G)}$, $11_{(G)}$, $11_{(G)}$, $10_{(G)}$, $10_{(G)}$). Postopek je viden v levem delu desnega stolpca na sliki 4.3. Podvojene kode nato z desne strani razširimo zaporedno z vrednostmi iz vektorja A . Postopek je prikazan v desnem delu desnega stolpca na sliki 4.3. Na ta način smo zapis razširili na dolžino $m = 3$, s čimer smo predstavitev števil razširili od $0_{(10)}$ do $7_{(10)}$ oz. na Grayeve kode $000_{(G)}$, $001_{(G)}$, $011_{(G)}$, $010_{(G)}$, $110_{(G)}$, $111_{(G)}$, $101_{(G)}$, $100_{(G)}$.

4.4 Pristopi k realizaciji pretvorbe iz binarne v Grayevo kodo

V pričujočem razdelku se posvetimo *logičnim pristopom* k realizaciji pretvornika iz binarne v Grayevo kodo. Logični pristopi so zelo preprosti. Na podlagi logičnih vrat (npr. AND vrat) sestavljamo logične funkcije, te pa nato uporabimo za gradnjo kakršnegakoli elektronskega vezja. Posebej se ozrimo na pojem *polnega funkcijskega nabora*. Poln funkcijski nabor predstavlja nabor logičnih funkcij, s katerim lahko zgradimo vse ostale logične funkcije. Primer polnega funkcijskega nabora je množica logičnih funkcij {AND, OR, NOT}. Polni nabor predstavljata tudi množici {NAND} (Shefferjev operator) in {NOR} (Pierceov operator), kar pomeni, da lahko elektronska vezja gradimo s pomočjo samo enega tipa logičnih vrat.

V nadaljevanju pričujočega razdelka predstavimo tri različne realizacije pretvornika v Grayevo kodo in sicer model klasičnega elektronskega vezja, model realizacije s kvantnim



Slika 4.3: Primer generiranja Grayeve kode s pomočjo razširitve kodnega zapisa velikosti dveh mest na kodni zapis velikosti treh mest. Začetne kode najprej podvojimo, nato pa z desne strani dodajamo po vrsti vrednosti iz vektorja A . Slika je povzeta po viru [37].

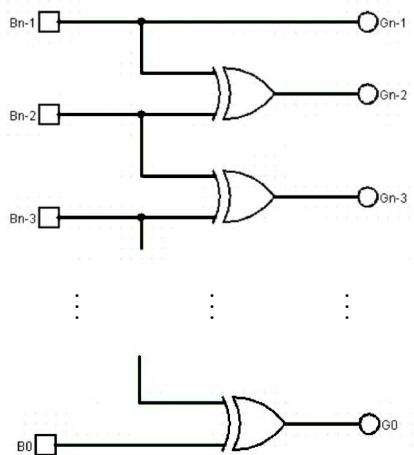
celičnim avtomatom in elektronsko optično realizacijo. Vse tri realizacije temeljijo na uporabi XOR logične funkcije, ki je v binarni logiki inherentno sestavljena iz funkcij polnega funkcijskega nabora {AND, OR, NOT}.

4.4.1 Model klasičnega elektronskega vezja

Model pretvornika binarne kode v Grayevo kodo na osnovi klasičnega elektronskega vezja je zgrajen s pomočjo logičnih vrat XOR. Postavljen je s pomočjo programskega orodja za gradnjo in simulacijo logičnih vezij Logisim [38] in prikazan na sliki 4.4. Na levi strani kot vhodi nastopajo znaki binarne kode, na desni strani pa kot izhode dobimo znake Grayeve kode. Prvi znak binarne kode (MSB) B_{n-1} se prenese v prvi znak Grayeve kode G_{n-1} , preostale znake Grayeve kode pa dobimo kot rezultat XOR funkcije istoležnega znaka v binarni kodi in njegovega predhodnika (znaka z večjim indeksom).

Logična vrata v klasičnih elektronskih vezjih so fizično realizirana s pomočjo tranzistorjev, ki so v uporabi že dolgo let. Tekom uporabe so raziskovalci na tem področju širili meje mogočega — hitrosti preklapljanja so postopoma naraščale, sama velikost tranzistorjev pa je postopoma padala. To je tudi razlog za težavo, ki danes predstavlja največji

izziv za napredek tranzistorske tehnologije. Trenutno so velikosti tranzistorjev dosegle tako majhne velikostne razrede, da je nadaljnja miniaturizacija le-teh močno otežena [1].



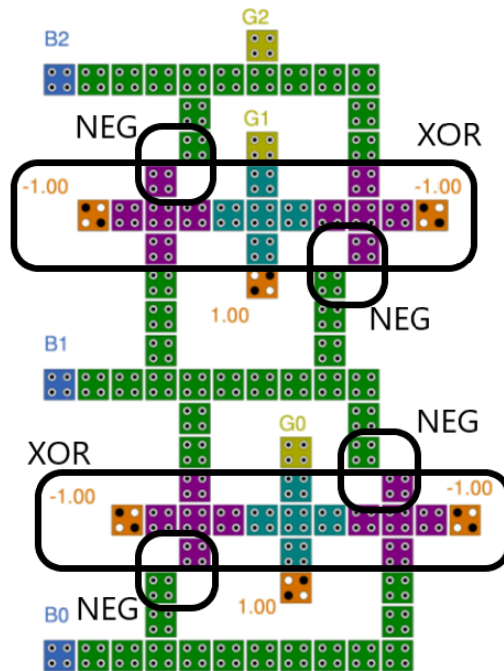
Slika 4.4: Logično vezje pretvornika iz binarne kode v Grayevo kodo. Leva stran predstavlja vhodni binarni zapis, desna stran pa izhodni Grayev zapis kot rezultat. MSB binarne kode B_{n-1} se preprosto prepiše v MSB Grayeve kode G_{n-1} . Preostali znaki Grayeve kode se nato izračunajo kot XOR operacija istoležnega znaka v binarni kodi in njegovega predhodnika z večjo utežjo.

4.4.2 Model realizacije s kvantnim celičnim avtomatom

V pričujočem razdelku predstavimo model pretvornika tri znakovne binarne kode v tri znakovno Grayevo kodo, realiziran s pomočjo kvantnega celičnega avtomata [39]. Vezje, predstavljeno na sliki 4.5, je sestavljeno s pomočjo majoritetnih vrat in negatorjev. Osnove delovanja kvantnih celičnih avtomatov so natančneje predstavljene v poglavju 2.

Iz slike 4.5 je razvidno, da se MSB B_2 iz vhodne celice prenese po vodilu v izhodno celico G_2 . Za pridobitev drugega znaka Grayeve kode (G_1) sta potrebni vrednosti vhodnih celic B_1 in B_2 , ki nastopata kot vhod v majoritetna vrata, ki izvajajo logično funkcijo XOR (črni pravokotnik na sliki 4.5). Predstavljena XOR funkcija je pravzaprav kombinacija treh preprostejših funkcij — AND, OR in NOT. Na začetku se pred vstopom v majoritetna vrata ena izmed dveh vhodnih vrednosti negira. Prvi dve majoritetni vrata (v vijolični barvi) izvršita AND funkcijo, rezultat teh dveh funkcij pa vstopa v tretja

majoritetna vrata (v modri barvi), ki izvršijo funkcijo OR. Kombinacija tega zaporedja opravljanja funkcij nad znaki nam da končni rezultat v izhodni celici G_1 . Enak postopek sledi še za vhodni celici B_0 in B_1 , rezultat pa je v celici G_0 . Rezultat pretvorbe nato preberemo iz izhodnih celic G_2, G_1 in G_0 .



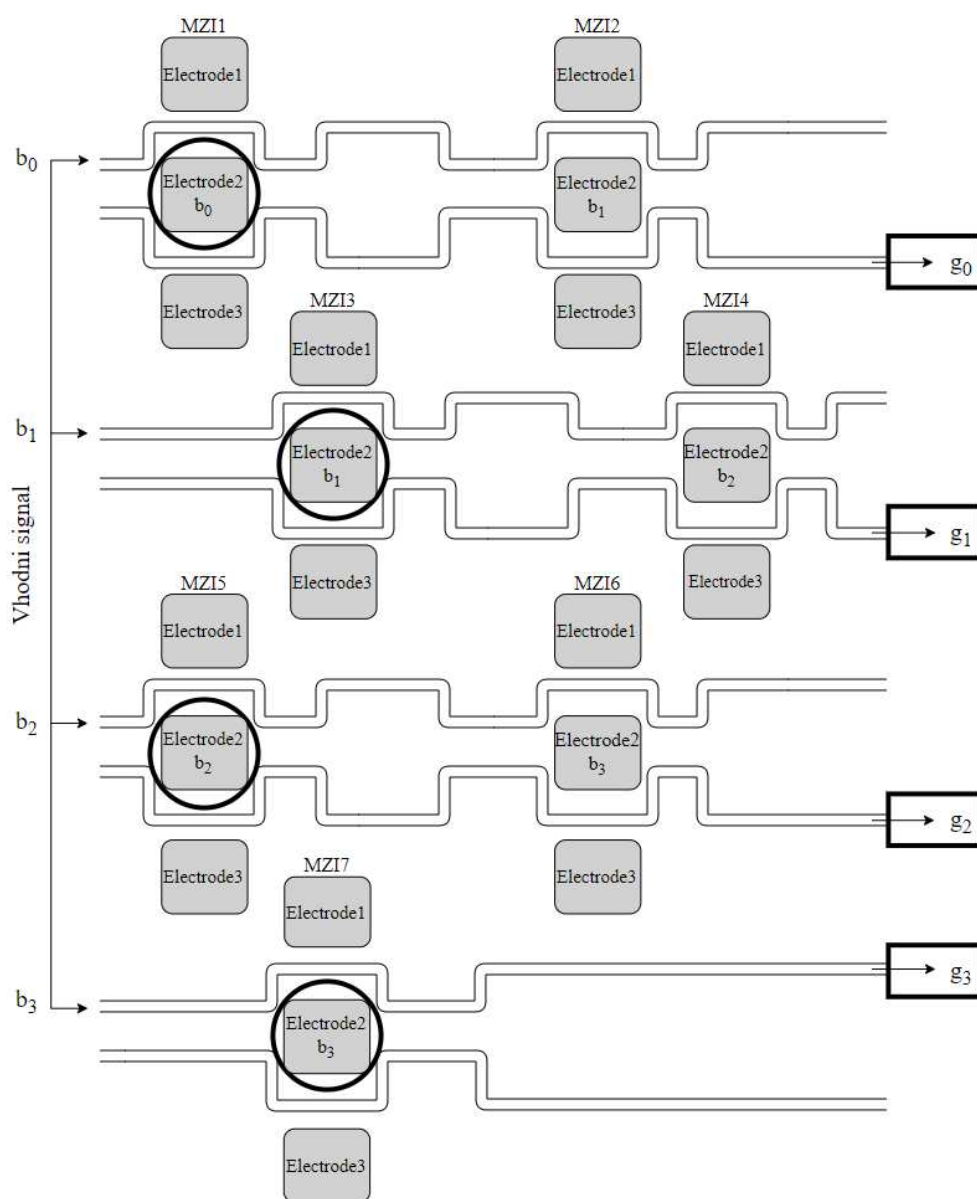
Slika 4.5: Primer pretvornika, ki temelji na kvantnem celičnem avtomatu. Pretvornik vsebuje tri vhodne celice B_2, B_1, B_0 in tri izhodne celice G_2, G_1, G_0 iz katerih preberemo končni rezultat pretvorbe. Slika je povzeta po viru [39].

Prednost uporabe kvantnih celičnih avtomatov pred klasičnimi elektronskimi vezji je predvsem v majhni porabi prostora za fizično realizacijo strukture, manjši energetski potratnosti in višjem taktu ure. Težava, s katero se soočajo kvantni celični avtomati, je izpostavljenost vplivom iz okolja. To se izkaže pri oteženem razlikovanju med polariziranimi in nepolariziranimi stanji elektronov v celicah kvantnega celičnega avtomata [15].

4.4.3 Elektronsko optična realizacija

Elektro optični pretvornik iz vira [40] sta avtorja zgradila na podlagi *Mach-Zehnderjevega interferometra* (MZI). Ta na podlagi elektro optičnega efekta deluje kot optični preklopnik. Kot je prikazano pri realizaciji s klasičnim elektronskim vezjem in kvantnim celičnim avtomatom, je za pretvorbo iz binarne v Grayevo kodo uporabljena logična funkcija XOR. Pri elektro optičnem pristopu XOR funkcijo pridobimo z zaporedno vezavo dveh MZI struktur, pretvornik pa nato zgradimo s paralelno vezavo poljubnega števila dveh zaporednih MZI struktur. Na sliki 4.6 je prikazan 4-bitni elektro optični pretvornik iz binarne v Grayevo kodo. Vhodni znaki binarne kode se aplicirajo na kontrolne elektrode MZI struktur (*Electrode2*), pri čemer se vrednosti znakov vhodne binarne kode $b_i = 1$ tolmačijo kot visoka vhodna napetost, $b_i = 0$ pa kot nizka vhodna napetost. Na podlagi podanih napetosti na kontrolnih elektrodah nato dobimo rezultat XOR funkcije na spodnjem izhodu zaporedne vezave dveh MZI struktur. Rezultat XOR funkcije je prisotnost izhodnega signala (logična enica) oziroma odsotnost le-tega (logična ničla). Izjema pretvorbe je le MSB binarne kode, ki se pri pretvorbi le prepíše v MSB Grayeve kode. Zaradi tega tukaj ni potrebna vezava dveh zaporednih MZI struktur, temveč potrebujemo le eno. Vrednost na kontrolni elektrodi se tako le prenese na višji izhod MZI strukture, kjer odčitamo vrednost g_3 , ki predstavlja MSB Grayeve kode. Na ta način smo pridobili pretvorjene znake binarne kode b_3, b_2, b_1, b_0 v Grayevo kodo iz izhodov g_3, g_2, g_1, g_0 označenih na sliki 4.6.

S pomočjo elektro optičnega pretvornika lahko dosežemo visoke hitrosti preklapljanja. Na podlagi raziskave sta avtorja ugotovila, da lahko predlagana rešitev pripomore pri enkripcijskih tehnikah, ki temeljijo na optičnih mehanizmih.



Slika 4.6: Shema elektro optičnega pretvornika iz 4-bitne binarne kode v Grayevo kodo. Za realizacijo je uporabljenih 7 MZI struktur. S črnimi krogi so označene kontrolne elektrode, na katere se aplicirajo vhodne vrednosti binarne kode, s črnimi pravokotniki pa so označeni izhodi, na katerih se pojavijo izhodne vrednosti, iz katerih sestavimo Grayevo kodo. Slika je povzeta po viru [40].

5 Pregled QCA pretvornikov iz binarne v Grayevo kodo

V pričujočem poglavju opravimo pregled različnih rešitev implementacije QCA pretvornika iz binarne v Grayevo kodo. Predstavimo nereverzibilne rešitve, ki uporabljajo različne pristope pri izgradnji pretvornika (predvsem različne pristope pri gradnji XOR logičnih vrat, potrebnih za pretvorbo) in pa reverzibilne rešitve, ki temeljijo na uporabi Feynmanovih logičnih vrat. V tabeli 5.1 je predstavljenih sedem različnih rešitev QCA pretvornikov, ki nam v nadaljevanju služijo kot izhodišče za primerjavo naše rešitve z obstoječimi implementacijami. Pri vsaki izmed obstoječih rešitev se osredotočimo na število znakov, ki jih pretvornik pretvori iz binarne v Grayevo kodo, število uporabljenih celic v strukturi, površino, ki jo zasede struktura, število uporabljenih urinih faz, število logičnih operacij in na to, ali je struktura reverzibilna. Površina strukture je definirana kot število vseh celic štirikotnika, ki omejuje strukturo. Za vsako izmed rešitev podamo tudi kratek komentar v zadnjem stolpcu tabele.

	Avtor	n	št. celic	S	UF	LO	REV	komentar
1	S. Islam, et al. [39]	4	131	390	3	15	NE	le tri urine faze
2	F. Ahmad, et al. [41]	4	137	306	3	15	NE	kompaktna struktura
3	D. Bhowmik, et al. [42]	4	124	324	4	14	NE	majhna struktura, štiri urine faze
4	P. Vijayalakshmi, et al. [43]	4	34	121	1	6	NE	izredno majhna velikost
5	A. H. Majeed, et al. [44]	4	56	138	2	9	NE	izredno majhna velikost
6	E. T. Karkaj, et al. [45]	4	108	232	4	11	DA	majhna velikost reverzibilne strukture
7	J. I. Reshi, et al. [46]	4	111	252	4	15	DA	kompaktna reverzibilna struktura

Tabela 5.1: Primerjava sedmih različnih rešitev 4-bitnega pretvornika iz binarne v Grayevo kodo. n predstavlja število znakov pretvorbe, stolpec z oznako št. celic pove, koliko kvantnih celic je uporabljenih za rešitev, S označuje površino, ki jo zasede struktura, UF pa ponazarja število urinih faz, uporabljenih v strukturi. Število logičnih operacij je prikazano v stolpcu z oznako LO. Oznaka REV nam pove, ali je struktura reverzibilna ali ne.

5.1 Nereverzibilne QCA rešitve

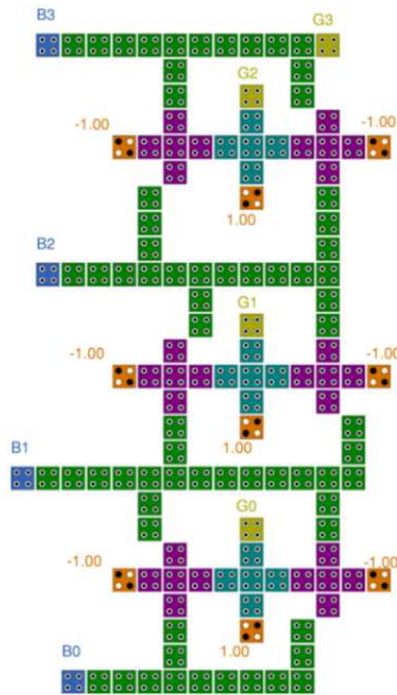
Pričujoči razdelek je namenjen nereverzibilnim rešitvam QCA pretvornika, v katerih so prikazani predvsem različni pristopi gradnje XOR logičnih vrat, ki so potrebna za pravilno delovanje pretvornika.

5.1.1 Pretvornik po Islamu

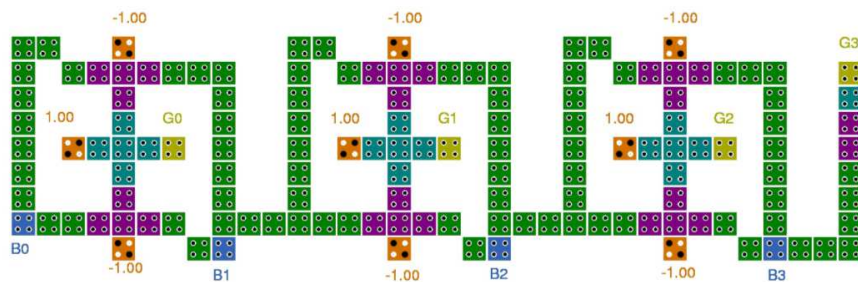
Pri prvi referenčni strukturi [39] opazimo, da za svoje delovanje uporablja samo tri urine faze. V primerjavi z ostalimi strukturami iz tabele zavzema največjo površino. Predstavlja najbolj preprost način izgradnje pretvornika iz binarne v Grayevo kodo. Pretvornik je sestavljen iz treh XOR logičnih vrat. XOR vrata so zgrajena s pomočjo treh majoritetnih vrat (dvoje majoritetnih vrat opravlja logično funkcijo AND, tretja majoritetna vrata pa logično funkcijo OR) in dveh negatorjev. Struktura je predstavljena na sliki 5.1.

5.1.2 Pretvornik po Ahmadu

Druga referenčna struktura iz vira [41] je sestavljena na podoben način kot struktura iz vira [39]. Iz primerjave obeh struktur je razvidno, da je struktura [41] rotirana za 90° . Pri primerjavi s strukturo iz [39] opazimo, da pri izgradnji uporabi večje število celic, kljub temu pa je bolj kompaktna, saj zasede manjšo površino. Prikazana je na sliki 5.2.



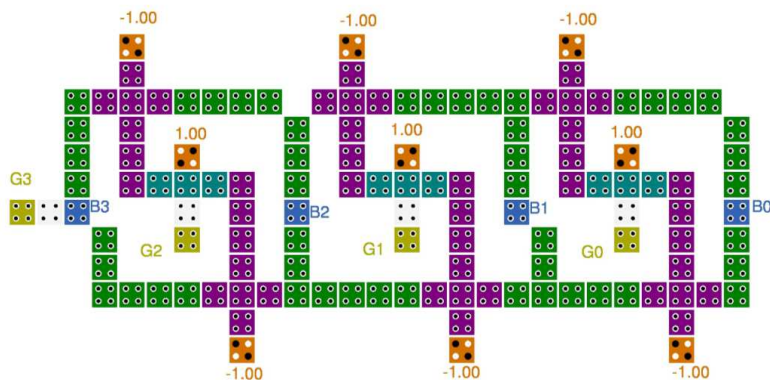
Slika 5.1: Struktura 4-bitnega pretvornika iz vira [39]. Za pretvorbo se uporabljajo tri XOR logična vrata, pri čemer so vsaka od njih sestavljena iz treh majoritetnih vrat in dveh negatorjev.



Slika 5.2: Struktura 4-bitnega pretvornika iz vira [41]. Struktura se glede na strukturo iz [39] razlikuje v rotaciji za 90° in v kompaktnosti.

5.1.3 Pretvornik po Bhowmiku

Tretja referenčna struktura [42] predstavlja še enega izmed bolj klasičnih pristopov gradnje. Struktura je zelo podobna predhodno opisanima strukturama. Predstavljena je na sliki 5.3. Za svoje delovanje uporablja en negator manj, prav tako pa za svoje delovanje porabi manj celic. Slabše se odreže pri urinih fazah, saj uporablja štiri.



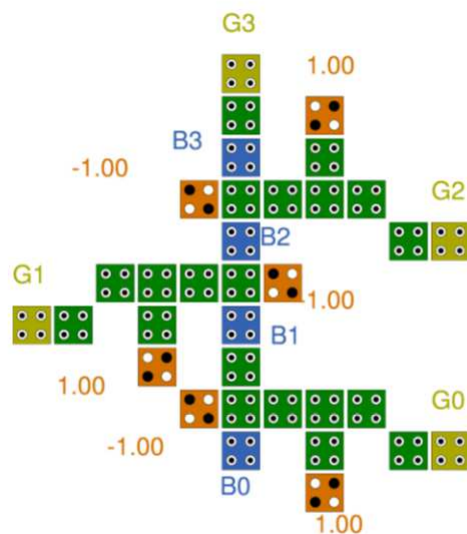
Slika 5.3: Struktura 4-bitnega pretvornika iz vira [42]. Za svoje delovanje uporablja štiri urine faze.

5.1.4 Pretvornik po Vijayalakshmi

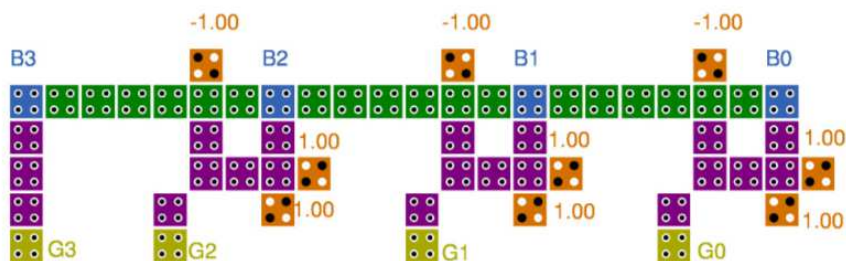
Četrta referenčna struktura [43] predstavlja po kriteriju površine najmanjšo izmed vseh analiziranih. Uporablja nov pristop pri gradnji XOR logičnih vrat. XOR vrata, predstavljena v tem članku, za svoje delovanje uporabljajo samo ena majoritetna vrata. Zaradi manjšega števila uporabljenih majoritetnih vrat pridobijo tako na prostoru, kot tudi številu uporabljenih celic. Še več, za svoje delovanje struktura uporablja samo eno urino fazo. Shema strukture je prikazana na sliki 5.4.

5.1.5 Pretvornik po Majeedu

Peta referenčna struktura prav tako uporablja nov pristop pri gradnji XOR logičnih vrat [44]. Avtor za izgradnjo XOR vrat uporabi le dvoje majoritetnih vrat in en negator. Število uporabljenih celic je močno zmanjšano glede na strukture, ki uporabljajo XOR logična vrata, ki temeljijo na treh majoritetnih vratih in dveh negatorjih. Struktura za delovanje prav tako uporablja le dve urini fazi. Struktura je predstavljena na sliki 5.5.



Slika 5.4: Struktura 4-bitnega pretvornika iz vira [43] uporablja drugačen pristop pri gradnji XOR logičnih vrat. XOR vrata uporabljajo samo ena majoritetna vrata in en negator.



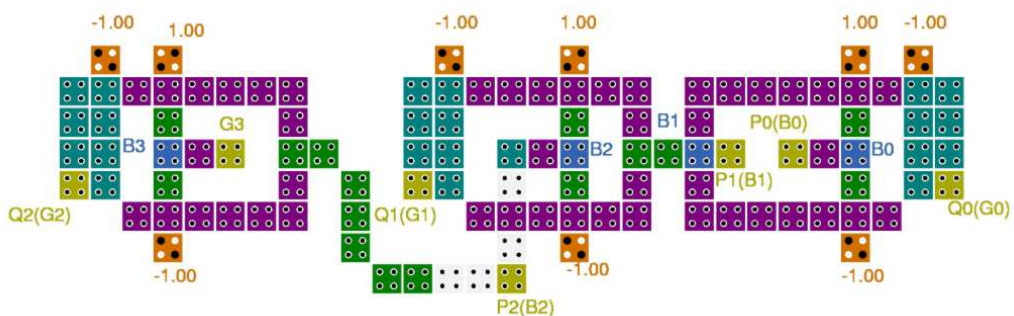
Slika 5.5: Struktura 4-bitnega pretvornika iz vira [44] prav tako uporablja drugačen pristop pri gradnji XOR logičnih vrat. XOR vrata so sestavljena iz dvojih majoritetnih vrat in enega negatorja.

5.2 Reverzibilne QCA rešitve

V nadaljevanju predstavimo še dve strukturi QCA pretvornika, ki se od predhodno opisanih struktur razlikujeta v tem, da sta logično reverzibilni.

5.2.1 Pretvornik po Karkaju

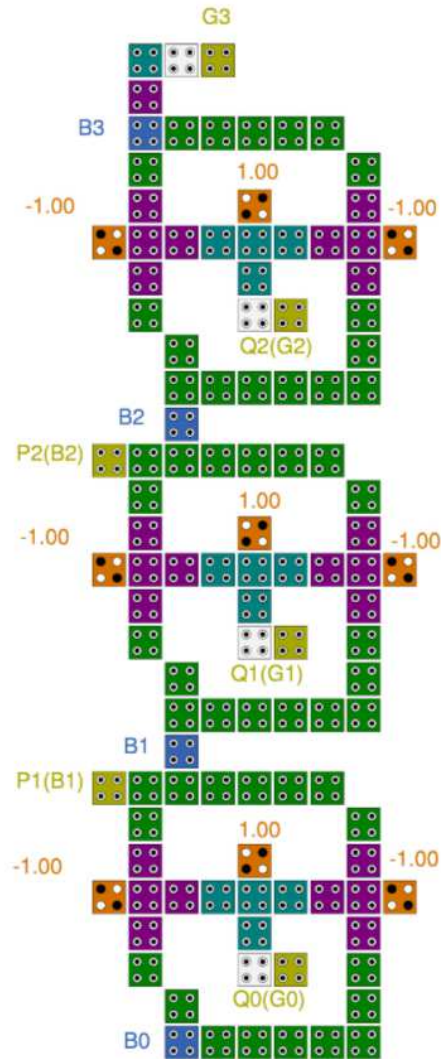
Prva referenčna reverzibilna struktura [45] uporablja Feynmanova logična vrata, ki predstavljajo reverzibilna vrata tipa (2,2). Eden izmed izhodov služi kot odvečni izhod, na drugem izhodu pa dobimo rešitev XOR logične funkcije. Vhod v to funkcijo predstavljata oba vhoda Feynmanovih vrat. Navkljub reverzibilnosti ta struktura še vedno ohranja majhno velikost, uporablja pa vse štiri urine faze. Predstavljena je na sliki 5.6.



Slika 5.6: Struktura reverzibilnega 4-bitnega pretvornika iz vira [45], ki temelji na Feynmanovih logičnih vratih.

5.2.2 Pretvornik po Reshiju

Druga referenčna reverzibilna struktura prav tako temelji na uporabi Feynmanovih vrat [46]. Ta struktura je pravzaprav zelo podobna strukturi iz vira [39]. Avtorji so v članku odvečne izhode sicer odstranili, kar daje občutek, da struktura ni reverzibilna. Sliko smo na podlagi članka prerisali tako, da smo dodali tudi odvečne izhode, kar nakazuje na to, da je struktura reverzibilna. V primerjavi s prejšnjo reverzibilno strukturo je ta nekoliko večja, prav tako uporablja nekaj več celic, še vedno pa je rešitev zelo kompaktna. Za svoje delovanje prav tako uporablja štiri urine faze. Struktura je prikazana na sliki 5.7.



Slika 5.7: Struktura reverzibilnega 4-bitnega pretvornika iz vira [46], ki temelji na Feynmanovih logičnih vratih.

6 Predlagana rešitev reverzibilnega QCA pretvornika

S pregledom obstoječih rešitev QCA pretvornika iz binarne v Grayevo kodo v poglavju 5 pridobimo dovolj znanja, da identificiramo slabosti obstoječih rešitev, ki jih bomo z našo rešitvijo skušali odpraviti. Naša rešitev je izdelana postopno po korakih. V prvem koraku postavimo osnoven model nereverzibilnega QCA pretvornika s pomočjo Feynmanovih vrat, ki jih sestavljajo tri majoritetna vrata in dva negatorja. Zatem našo osnovno rešitev postopno minimiziramo na lokalnem nivoju (na nivoju enih Feynmanovih vrat) in globalnem nivoju (na nivoju celotne strukture QCA pretvornika), prav tako pa dodamo še dodatek za reverzibilnost. Na koncu podamo končno oceno in komentar naše rešitve.

6.1 Osnovna rešitev

V pričujočem poglavju se posvetimo Feynmanovim vratom, ki predstavljajo osnovo pri gradnji pretvornika iz binarne v Grayevo kodo. Prav tako predstavimo tudi osnovno rešitev, jo ocenimo in navedemo nadaljnje možnosti za izboljšavo.

6.1.1 Feynmanova vrata kot QCA realizacija

Feynmanova vrata so primer reverzibilnih vrat tipa $(2, 2)$, kar pomeni, da imajo dva vhoda in dva izhoda. Vhoda v Feynmanova vrata sta poimenovana A in B , izhoda pa P in Q . Izhod P je definiran kot $P = A$. Izhod Q je definiran kot $Q = A \oplus B$, torej je rezultat XOR operacija nad vodomoma A in B . Logična shema Feynmanovih vrat je prikazana na sliki 6.1.



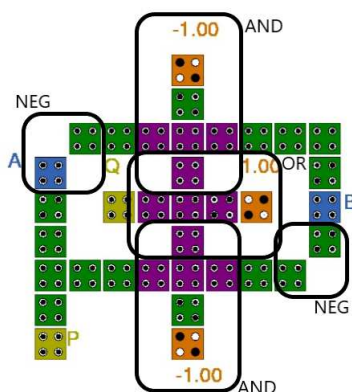
Slika 6.1: Logična shema Feynmanovih vrat.

Za izhodišče QCA realizacije smo uporabili Feynmanova vrata iz vira [47]. Njihova QCA implementacija je vidna na sliki 6.2. QCA realizacija Feynmanovih vrat sestoji iz petih logičnih primitivov in sicer iz dveh negatorjev in treh majoritetnih vrat. Majoritetna vrata dvakrat vršijo logično funkcijo AND, enkrat pa logično funkcijo OR. Modro obarvani celici na sliki 6.2 predstavljajta vhoda v Feynmanova vrata. Vhod A se, kot smo že omenili, samo prenese v izhodno celico P , izhodna celica Q pa potrebuje nekaj več predhodnega procesiranja, da lahko iz nje preberemo končni rezultat. Izhodna celica Q je definirana kot XOR operacija nad vodomoma A in B ($Q = A \oplus B$). Za lažjo predstavo lahko XOR operacijo zapišemo kot kombinacijo negacije, konjunkcije in disjunkcije — $Q = \bar{A}B \vee A\bar{B}$. V zgornjem delu vrat na sliki 6.2 vidimo, da se vhod A najprej negira, nato pa vstopa v majoritetna vrata, ki vršijo funkcijo AND. Na drugi strani vhod B nespremenjen vstopa v ista majoritetna vrata. Na ta način se izvrši logična funkcija AND, definirana kot $MV(\bar{A}, B, 0)$, pri čemer fiksno polarizirana oranžna celica -1.00 predstavlja logično vrednost 0, ki nastopa kot tretji parameter majoritetnih vrat. Na ta način tudi nakazuje, da majoritetna vrata vršijo funkcijo AND.

V spodnjem delu strukture na sliki 6.2 je postopek podoben. Razlika je v tem, da se negira vhod B , vhod A pa ostane nespremenjen. Oba nato potujeta do spodnjih majoritetnih vrat, v katera vstopata skupaj s fiksno polarizirano oranžno celico -1.00 . Funkcija, ki se izvrši je $MV(A, \bar{B}, 0)$, torej ponovno AND funkcija.

Rezultata obeh majoritetnih vrat se nato preneseta kot vhodna parametra v tretja majoritetna vrata, kot tretji vhod pa nastopa fiksno polarizirana oranžna celica 1.00 , kar

predstavlja logično vrednost 1. To nam pove, da ta majoritetna vrata vršijo logično funkcijo OR, saj je tretji parameter enak 1. Majoritetna vrata tako vršijo funkcijo $MV(MV(\bar{A}, B, 0), MV(A, \bar{B}, 0), 1)$. Rezultat se prenese v izhodno celico Q , ki tako hrani rezultat XOR operacije nad vhodoma A in B .



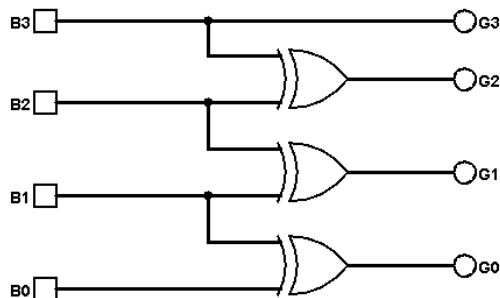
Slika 6.2: QCA realizacija Feynmanovih vrat, ki smo jo uporabili za izhodišče pri gradnji 4-bitnega pretvornika iz binarne v Grayevo kodo, povzeta po viru [47]. Označene so tudi vse logične operacije, ki se izvedejo ob procesiranju obeh vhodov.

6.1.2 Logična in QCA shema pretvornika

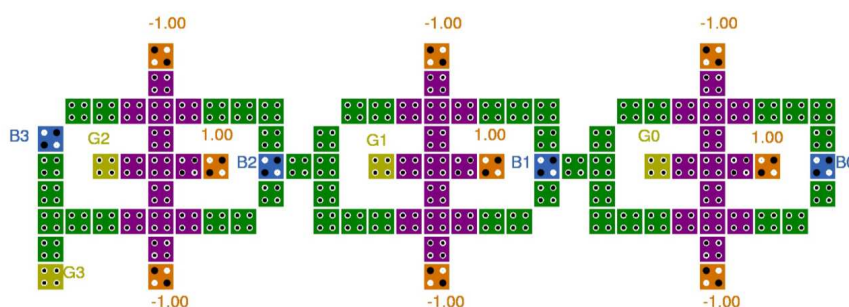
V pričujočem razdelku je prikazana logična in QCA shema 4-bitnega pretvornika iz binarne v Grayevo kodo. Za izhodišče gradnje smo vzeli Feynmanova vrata, ki so bila opisana v prejšnjem razdelku. Logična shema pretvornika je predstavljena na sliki 6.3, QCA struktura pa na sliki 6.4. Iz logične sheme je razvidno katere funkcije se v vezju vršijo, zato služi kot referenca ob analizi QCA sheme. Na QCA shemi opazimo, da struktura pretvornika deluje v dveh urinih fazah. Zelena barva predstavlja prvo urino fazo, vijolična barva pa drugo urino fazo. Urine faze so definirane na sliki 6.5.

6.1.3 Analiza delovanja QCA strukture

Delovanje 4-bitnega pretvornika s slike 6.4 smo preverili s pomočjo simulacije delovanja v programskem orodju QCADesigner. Za simulacijo vezja sta na voljo dva pristopa — bistabilna aproksimacija in simulacija s koherentnim vektorjem. Za našo simulacijo smo izbrali pristop s koherentnim vektorjem, saj je ta bolj robusten. Pred samim izvajanjem simulacije v orodju QCADesigner je možno nastavljanje različne parametre, ki vplivajo

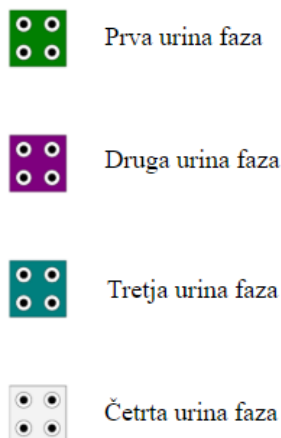


Slika 6.3: Logična shema 4-bitnega pretvornika iz binarne v Grayevo kodo.



Slika 6.4: QCA shema 4-bitnega pretvornika iz binarne v Grayevo kodo, ki za osnovo uporablja Feynmanova vrata iz vira [47].

na simulacijo. Le-te smo pustili na privzetih vrednostih. Za lažje preverjanje rezultatov delovanja strukture smo vključili tabelo, ki prikazuje vse 4-bitne binarne vrednosti in njihove pripadajoče vrednosti v Grayevem zapisu (glej tabelo 6.1). Rezultati simulacije so predstavljeni na sliki 6.6. Modri grafi predstavljajo vhode B_3 , B_2 , B_1 in B_0 , rumeni grafi pa pripadajoče pretvorjene izhode G_3 , G_2 , G_1 in G_0 v Grayevem zapisu. Rdeči grafi ponazarjajo potek urinega signala v QCA vezju. Ker vezje deluje le v dveh urinih fazah, sta za nas relevantna le urina signala $CLOCK0$ in $CLOCK1$. Na tem mestu bi opomnili na to, da se izhod G_3 bere, ko je signal $CLOCK0$ nizek, izhodi G_2 , G_1 in G_0 pa ko je signal $CLOCK1$ nizek. Razlog za to je viden že iz same sheme QCA pretvornika (slika 6.4), saj je izhod G_3 vezan na celico v prvi urini fazi (signal $CLOCK0$), preostali izhodi pa na drugo urino fazo (signal $CLOCK1$). Iz grafov lahko razberemo, da vezje deluje pravilno, saj rezultati sovpadajo z vrednostmi iz tabele 6.1.



Slika 6.5: Barvno kodirane faze ure, kakor jih za svoje delovanje uporablja programsko orodje QCADesigner.

6.1.4 Ocena osnovne rešitve

Sledi primerjava osnovne rešitve s predhodno analiziranimi rešitvami drugih avtorjev. V ta namen smo v to pod poglavje ponovno vključili tabelo rešitev (glej tabelo 6.2), v katero je dodana tudi naša osnovna rešitev. Na ta način imamo lažji in hitrejši vpogled v primerjavo.

Primerjava sledi po stolpcih tabele. Kot vse obstoječe rešitve, smo tudi mi naredili 4-bitni pretvornik iz binarne v Grayevo kodo. Naslednja metrika je število uporabljenih celic v vezju. Naša shema uporablja 103 celice. V tej metriki smo slabši le od rešitev [43] in [44], ki pa obe uporabljata bolj inovativen pristop pri gradnji XOR vrat z manj celicami.

Naslednja metrika je površina (S), pri čemer naša površina vsebuje 261 celic. V tej metriki nas prekašata prej omenjeni rešitvi z inovativnim pristopom gradnje XOR vrat, kot tudi obe reverzibilni rešitvi [45] in [46].

Naslednja metrika je število uporabljenih urinih faz. Naša osnovna rešitev uporablja le dve urini fazi. S tem rezultatom smo slabši le od rešitve [43], izenačimo se z rešitvijo [44], ostale pa prekašamo.

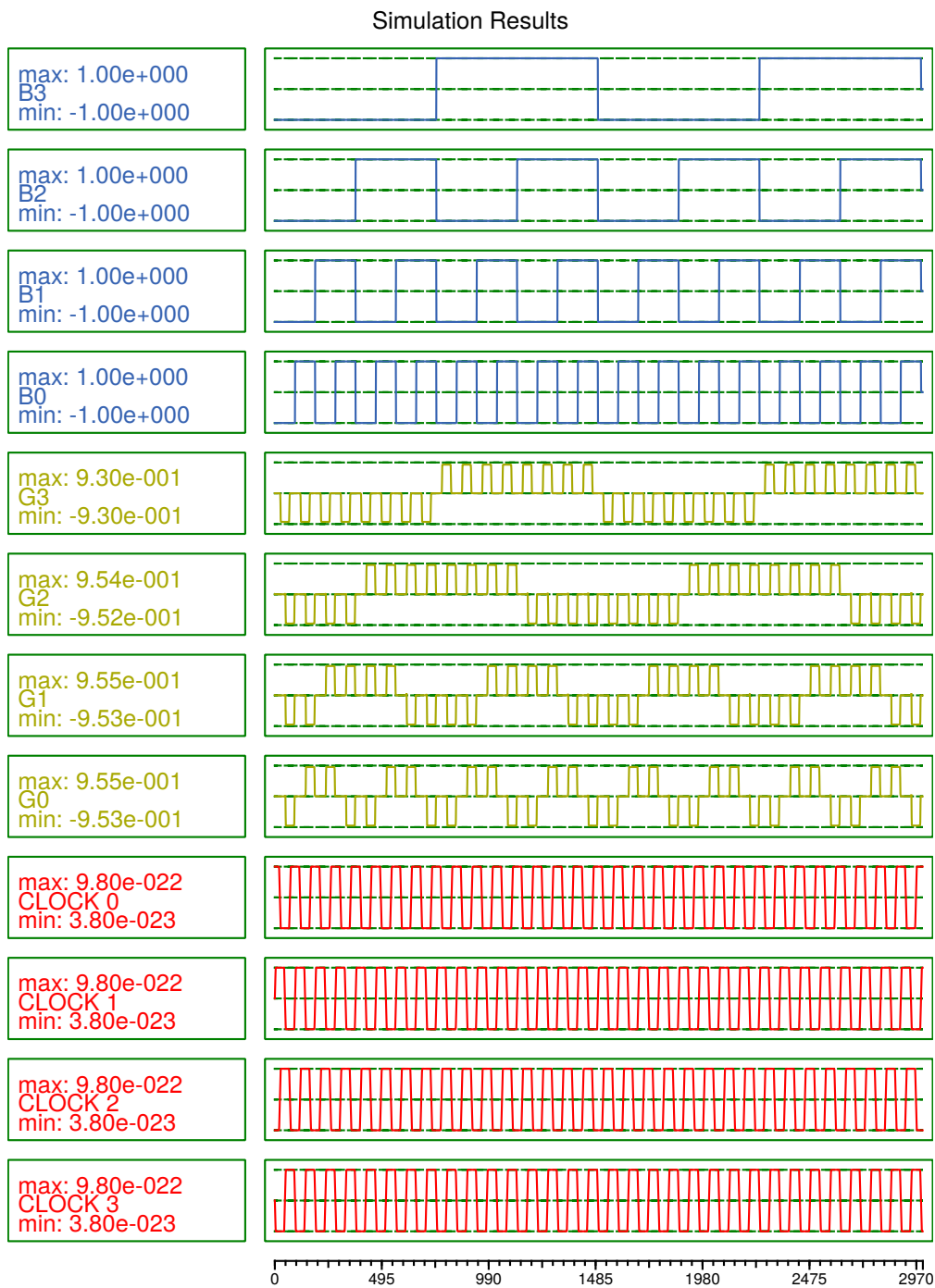
Sledi še število logičnih operacij. Pri gradnji naše osnovne rešitve uporabljamo klasičen pristop gradnje s pomočjo majoritetnih vrat in negatorjev. Tako uporabljamo

b_3	b_2	b_1	b_0	g_3	g_2	g_1	g_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

Tabela 6.1: Vse 4-bitne vrednosti za binarno kodo (levi del tabele) in za Grayevo kodo (desni del tabele).

15 različnih logičnih operacij (šest negatorjev, šest AND operacij in tri OR operacije). V tej metriki smo najslabši. Izenačimo se z rešitvama [39] in [41], ostale rešitve pa so v tej metriki boljše.

Zadnja metrika pove ali je rešitev reverzibilna ali ne. Naša osnovna rešitev pravzaprav še ni reverzibilna. Da bo rešitev reverzibilna, morajo vsaka izmed Feynmanovih vrat v strukturi vsebovati dva vhoda in dva izhoda, pri čemer je eden izmed izhodov odveč in služi kot pogoj za reverzibilnost. Izjema je le odvečni izhod pri prvih Feynmanovih vratih, ki predstavlja MSB Grayeve kode in je na ta način del končne rešitve. Če se sklicujemo na sliko 6.3, vhoda v prva Feynmanova vrata predstavljata vrednosti B_3 in B_2 , na izhodu pa dobimo vrednosti G_3 (kot odvečni izhod) in G_2 . Vhoda v druga Feynmanova vrata predstavljata vrednosti B_2 in B_1 , kot izhod dobimo G_1 , manjka pa še drugi, odvečni izhod, ki za nas sicer ne nosi nobenega pomena. Enako sledi še pri tretjih Feynmanovih vratih. Vhoda predstavljata vrednosti B_1 in B_0 , izhod je G_0 , prav tako pa manjka še odvečni izhod. Iz tega lahko ugotovimo, da nam za reverzibilnost naše osnovne rešitve manjkata še dva odvečna izhoda. Če se vrnemo k primerjavi osnovne rešitve pretvornika z rešitvami iz literature ugotovimo, da sta zato obe obstoječi reverzibilni rešitvi [45] in



Slika 6.6: Rezultat 4-bitnega pretvornika iz binarne v Grayevo kodo. Pretvorjene vrednosti Grayeve kode so predstavljene z rumenimi grafi.

Avtor	n	št. celic	S	UF	LO	REV	komentar
1 S. Islam, et al. [39]	4	131	390	3	15	NE	le tri urine faze
2 F. Ahmad, et al. [41]	4	137	306	3	15	NE	kompaktna struktura
3 D. Bhowmik, et al. [42]	4	124	324	4	14	NE	majhna struktura, štiri urine faze
4 P. Vijayalakshmi, et al. [43]	4	34	121	1	6	NE	izredno majhna velikost
5 A. H. Majeed, et al. [44]	4	56	138	2	9	NE	izredno majhna velikost
6 E. T. Karkaj, et al. [45]	4	108	232	4	11	DA	majhna velikost reverzibilne strukture
7 J. I. Reshi, et al. [46]	4	111	252	4	15	DA	kompaktna reverzibilna struktura
8 Osnovna rešitev	4	103	261	2	15	NE	samo dve urini fazi, relativno majhna struktura

Tabela 6.2: Primerjava naše osnovne rešitve z ostalimi rešitvami. n predstavlja število znakov pretvorbe, stolpec z oznako št. celic pove, koliko kvantnih celic je uporabljenih za rešitev, S označuje površino, ki jo zasede struktura, UF pa ponazarja število urinih faz, uporabljenih v strukturi. Število logičnih operacij je prikazano v stolpcu z oznako LO. Oznaka REV nam pove, ali je struktura reverzibilna ali ne.

[46] v tem smislu boljši.

Na tem mestu bi lahko rekli, da je naša osnovna rešitev v določenih pogledih že kar dobra, lahko pa jo izboljšamo še na nekaj mestih. V nadaljevanju dela bomo stremeli k zmanjšanju vrednosti predstavljenih metrik. Prav tako bomo naš pretvornik spremenili v reverzibilno obliko.

6.2 Izboljšave osnovne rešitve

V nadaljevanju definiramo načrtovalske cilje, h katerim bomo stremeli z namenom izboljšanja naše osnovne rešitve v smislu predhodno definiranih metrik. Ker so QCA majoritetna vrata prostorsko zelo potratna, bomo strukturo najprej minimizirali na lokalnem nivoju (na nivoju XOR oz. Feynmanovih vrat) s pomočjo medsebojne interakcije kvantnih celic. Na ta način bomo drastično zmanjšali število uporabljenih celic v strukturi. Nato se bomo posvetili minimizaciji strukture na globalnem nivoju (na nivoju strukture). S tem bomo zmanjšali število celic in posledično tudi površino, ki jo zaseda pravokotnik, v katerega je vpeta struktura pretvornika. Za konec bomo dodali tudi potrebne manjkajoče odvečne izhode, da bomo zadostili reverzibilnosti strukture. Obenem bomo tekom preizkušanja omenjenih izboljšav stremeli k ohranjanju ali zmanjšanju števila uporabljenih urinih faz. Načrtovalske cilje, namenjene izboljšavi naše strukture, lahko tako strnemo v tri točke:

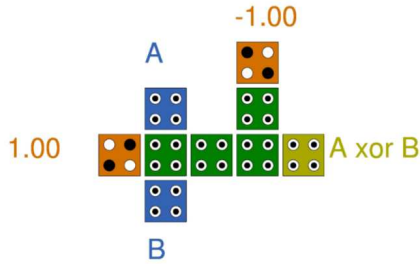
- minimizacija na lokalnem nivoju (nov pristop h gradnji XOR logičnih vrat),
- minimizacija na globalnem nivoju (na nivoju celotne strukture pretvornika) in
- dodatek za reverzibilnost.

6.3 Lokalna minimizacija

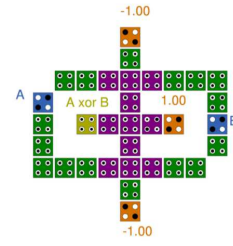
Prva izboljšava, ki jo preizkusimo, je minimizacija na lokalnem nivoju. Cilj te izboljšave je predelava logičnih vrat XOR na ta način, da uporabljajo čim manj majoritetnih vrat in negatorjev. Za doseg tega cilja se opremo na lastnost interakcije kvantnih celic med seboj. Vsaka celica ima v svojem okolju določen vpliv na ostale celice in to s pridom izkoristimo. Pravzaprav iščemo tako postavitev celic, kjer bodo celice kot struktura izvedle XOR logično funkcijo. V literaturi obstaja več različnih pristopov za doseg tega cilja [43, 44, 48–50]. Za izdelavo novih XOR logičnih vrat uporabimo celice fiksne polarizacije, s katerimi vplivamo na vhodne in prenosne celice tako, da dosežemo željeno funkcionalnost — v našem primeru je to pravilno izvajanje XOR logične funkcije. Za osnovo vzamemo majoritetna vrata, ki vršijo logično funkcijo OR, nato pa v okolici iščemo tako razporeditev celic, ki bi na izhod majoritetnih vrat vplivale tako, da bi v primeru, ko sta oba vhoda v majoritetna vrata enaka logični enici, na izhodu dobili logično ničlo, namesto enice.

Takšna postavitev celic je predstavljena na sliki 6.7a. S poizkušanjem različnih konfiguracij postavitve celic nam je uspelo zgraditi nova XOR logična vrata. Na levem delu slike 6.7 so predstavljena naša nova logična vrata XOR, na desni pa logična vrata XOR, ki so uporabljena v osnovni rešitvi pretvornika. Logična vrata XOR iz slike 6.7a bi lahko razdelili na tri dele. Levi del predstavlja majoritetna vrata, desni del pa predstavlja le izhodno celico v rumeni barvi. Ostane nam še sredinski del s fiksno polarizirano celico na -1.00 (logična ničla), ki predstavlja razporeditev celic, ki na izhod majoritetnih vrat vplivajo na tak način, da struktura kot celota opravlja XOR logično funkcijo.

Iz te strukture sedaj preprosto naredimo Feynmanova vrata, ki jih bomo kasneje uporabili za postavitev 4-bitnega pretvornika iz binarne v Grayevo kodo. Nad modro vhodno celico z oznako A dodamo še t.i. odvečni izhod, s čimer zadostimo pogoju za Feynmanova vrata (dva vhoda in dva izhoda). Na sliki 6.8 so predstavljena naša nova Feynmanova vrata, poleg njih pa je še shema Feynmanovih vrat, ki smo jih uporabili v osnovni rešitvi pretvornika.

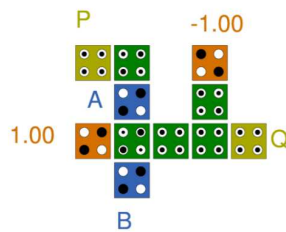


(a) Nova logična vrata XOR.

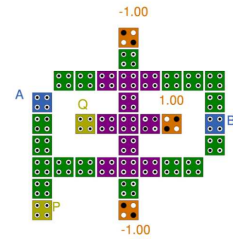


(b) Predhodna logična vrata XOR.

Slika 6.7: Nov pristop h gradnji logičnih vrat XOR (slika 6.7a), pri katerem so uporabljena samo ena majoritetna vrata. Število celic se glede na prejšnja logična vrata XOR (slika 6.7b) drastično zmanjša.



(a) Nova Feynmanova vrata.



(b) Predhodna Feynmanova vrata.

Slika 6.8: Primerjava novih in predhodnih Feynmanovih vrat. Na sliki 6.8a so prikazana izboljšana Feynmanova vrata (v smislu števila celic, površine, števila majoritetnih vrat in negatorjev), na sliki 6.8b pa Feynmanova vrata, ki smo jih vzeli kot izhodišče za osnovno rešitev.

Opazimo, da smo strukturo Feynmanovih vrat občutno izboljšali. Predhodno smo v strukturi Feynmanovih vrat uporabili tri majoritetna vrata in dva negatorja. Trenutna postavitev za svoje izvajanje uporablja samo ena majoritetna vrata. Prav tako trenutna struktura za svoje izvajanje uporablja le eno urino fazo. Pravilno delovanje novih Feynmanovih vrat smo preverili s pomočjo orodja QCADesigner, kjer se rezultati simulacije delovanja izkažejo za pravilne. Rezultati so vidni na sliki 6.9.



Slika 6.9: Rezultat simulacije novih Feynmanovih vrat. Izhod P je enak vhodu A, izhod Q pa predstavlja rezultat logične funkcije XOR nad vhodoma A in B.

6.4 Globalna minimizacija

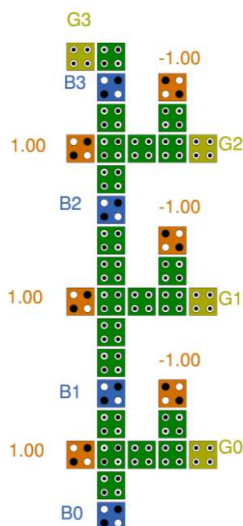
Po uspešno končani lokalni minimizaciji se posvetimo minimizaciji na globalnem nivoju oz. minimizaciji na nivoju strukture celotnega pretvornika. Minimizacija na globalnem nivoju se osredotoča na zmanjšanje števila celic in zmanjšanju površine pravokotnika, v katerega je vpeta struktura. S pomočjo novih Feynmanovih vrat lahko občutno zmanjšamo strukturo pretvornika iz prejšnjega poglavja (glej sliko 6.4). Za postavitev nove strukture pretvornika iz binarne v Grayevo kodo paralelno vezemo tri Feynmanova vrata. Tu se nam pojavita dve možnosti izboljšave na globalnem nivoju in sicer

- minimizacija z ohranitvijo ene urine faze in
- minimizacija s povečanjem števila uporabljenih urinih faz.

6.4.1 Minimizacija z ohranitvijo ene urine faze

Struktura, ki jo minimiziramo z ohranitvijo ene urine faze je vidna na sliki 6.10. Pri minimizaciji smo prišli do dveh ugotovitev. Ti sta sledeči:

- Med Feynmanovimi vrati smo morali vpeljati razdaljo vsaj dveh celic, sicer sta se izhoda G_2 in G_1 pokvarila. To smo dosegli z vpeljavo praznega prostora velikosti dveh QCA celic nad fiksno polariziranimi celicami -1.00 .
- Izhod G_1 je najbolj občutljiv, ker se nahaja na sredinskih Feynmanovih vratih (vpliv prihaja tako od zgornjih kot tudi spodnjih Feynmanovih vrat). Zaradi tega sta vhoda B_2 in B_1 od srednjih Feynmanovih vrat oddaljena za eno celico več, sicer se je vhod G_1 pokvaril.



Slika 6.10: 4-bitni pretvornik iz binarne v Grayevo kodo, zgrajen s pomočjo nadgrajenih Feynmanovih vrat. Struktura je globalno minimizirana, pri čemer ohranjamo eno urino fazo.

Z upoštevanjem omenjenih ugotovitev smo obdržali pravilno delovanje pretvornika.

6.4.2 Minimizacija z uporabo več urinih faz

Strukturo pretvornika lahko še dodatno minimiziramo, mu pa moramo zato pripisati drugo slabost in sicer je to uporaba več urinih faz, kar se manifestira v zakasnitvi. S pre-

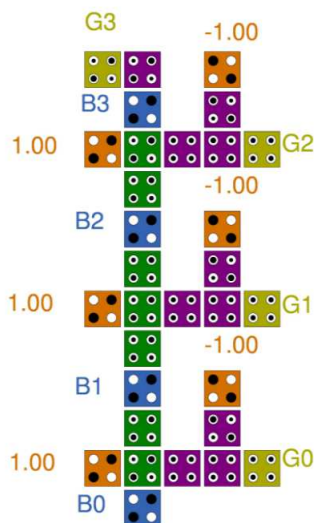
stavitvijo nekaterih celic v drugo urino fazo izničimo vpliv, ki ga imajo celice v različnih urinih fazah med seboj. Pri tem izključimo fiksno polarizirane celice, saj te svoje stanje ohranjajo in posledično tudi vplivajo na ostale celice ves čas. Na ta način smo še dodatno minimizirali strukturo pretvornika, ki je predstavljena na sliki 6.11. V drugo urino fazo smo prestavili vse izhodne celice v rumeni barvi (od G_3 do G_0) in vse celice, ki so na sliki obarvane vijolično. Prišli smo do sledečih ugotovitev:

- Če uporabljamo dve urini fazi in izhode G_3 , G_2 , G_1 in G_0 postavimo v drugo urino fazo, potem fiksno polarizirane celice -1.00 (logična ničla), ki ostanejo v prvi urini fazi, nimajo na izhode več nikakršnega vpliva. Tako smo lahko razdaljo med Feynmanovimi vrati iz predhodnih dveh celic zmanjšali na zgolj eno celico. Nad fiksno polariziranimi celicami -1.00 imamo tako prazen prostor velikosti le ene QCA celice.
- Vhoda B_2 in B_1 prav tako več ne motita izhoda G_1 in sta tako od sredinskih Feynmanovih vrat lahko oddaljena le še za eno celico.
- Izhodne celice G_3 ne moremo vezati direktno na vhodno celico B_3 , saj ima fiksno polarizirana celica -1.00 negativen vpliv na izhodno vrednost celice G_3 .

Z upoštevanjem omenjenih ugotovitev še dodatno zmanjšamo strukturo, obenem pa tudi ohranimo pravilno delovanje pretvornika.

6.5 Dodatek za reverzibilnost

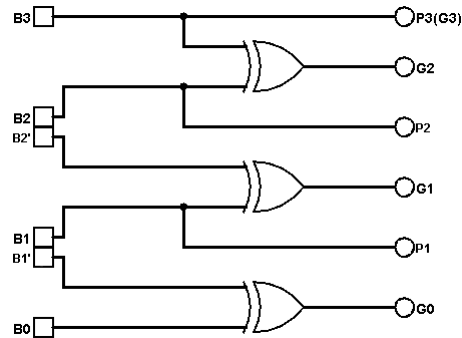
Zadnja izboljšava, ki se je lotimo, je sprememba obstoječe strukture v reverzibilno. Reverzibilni QCA pretvornik iz binarne v Grayevo kodo je sestavljen iz treh Feynmanovih vrat. Logična struktura pretvornika je prikazana na sliki 6.12. Če logično shemo primerjamo z našo QCA shemo pretvornika (sliki 6.10 in 6.11) opazimo, da prva (zgornja) Feynmanova vrata v pretvorniku upoštevajo pravilo dveh vhodov in dveh izhodov, pri čemer je odvečni izhod pravzaprav izhod G_3 , drugi izhod pa rezultat XOR logične funkcije nad vhodoma B_3 in B_2 , ki nam da rezultat G_2 . Pri naslednjih dveh Feynmanovih vratih, kjer dobimo rezultata G_1 in G_0 opazimo, da nam za upoštevanje pogoja dveh vhodov in dveh izhodov manjkata še dva odvečna izhoda, ki sta pri Feynmanovih vratih nujno potrebna. Obe strukturi pretvornika tako razširimo še z odvečnima izhodoma P_2 in P_1 . S tem, ko obe strukturi razširimo z odvečnima izhodoma P_2 in P_1 , prekršimo



Slika 6.11: 4-bitni pretvornik iz binarne v Grayevo kodo, zgrajen s pomočjo nadgrajenih Feynmanovih vrat. Struktura je globalno minimizirana, pri čemer smo število celic glede na strukturo na sliki 6.10 še zmanjšali s pomočjo vpeljave dodatne urine faze.

pravilo, ki pravi, da mora biti število vhodov in izhodov enako. Zato moramo v spodnji dve Feynmanovi vrati vpeljati še dva nova vhoda, ki pa sta pravzaprav kopiji vhodov B_2 in B_1 , saj ta dva vhoda v prejšnji strukturi pretvornika vstopata v dvojne Feynmanovih vrat. S pomočjo orodja QCADesigner lahko definiramo vhodni vektor, ki sestoji iz dveh vhodnih celic, ki bosta vedno imeli enako vrednost. Na ta način vpeljemo vhodni vektor \mathbf{B}_2 , ki vsebuje vhodni celici B_2 in B'_2 ter vhodni vektor \mathbf{B}_1 , ki vsebuje vhodni celici B_1 in B'_1 . Na sliki 6.12 je sedaj razvidno, da vsaka izmed treh Feynmanovih vrat upoštevajo pravilo dveh vhodov in dveh izhodov. Zgornja Feynmanova vrata vsebujejo vhoda B_3 in B_2 ter izhoda G_3 in G_2 . Srednja Feynmanova vrata vsebujejo vhoda B'_2 (kopija vhoda B_2) in B_1 ter izhoda P_2 in G_1 . Spodnja Feynmanova vrata vsebujejo vhoda B'_1 (kopija vhoda B_1) in B_0 ter izhoda P_1 in G_0 . Spremembi v strukturah sta sledeči:

- Pri obeh strukturah uvedemo dva dodatna vhoda B'_2 in B'_1 , ki predstavljata kopiji obstoječih vhodov B_2 in B_1 in dva dodatna izhoda P_2 in P_1 .
- Pri strukturi z dvema urinima fazama je pomembno, da je med vhodom in izhodom vsaj ena celica. Odvečna izhoda sta prav tako postavljena v drugo urino fazo, da sovpadata z ostalimi izhodi v strukturi pretvornika.



Slika 6.12: Logična shema reverzibilnega 4-bitnega pretvornika iz binarne v Grayovo kodo.

Na ta način smo obe strukturi napravili še reverzibilni v smislu uporabe reverzibilnih gradnikov. To je sicer prineslo nekaj dodatnih celic in povečanje površine v primeru pretvornika z dvema urinima fazama. Kljub temu nam reverzibilnost obeh rešitev omenjeni slabosti odtehtata. Reverzibilna QCA pretvornika iz binarne v Grayovo kodo sta vidna na sliki 6.13.



(a) Reverzibilni pretvornik z eno urino fazo. (b) Reverzibilni pretvornik z dvema urinima fazama.

Slika 6.13: Dve različni strukturi reverzibilnega 4-bitnega pretvornika iz binarne v Grayovo kodo. Na sliki 6.13a je prikazan reverzibilni pretvornik, ki za svoje delovanje uporablja le eno urino fazo, na sliki 6.13b pa reverzibilni pretvornik, ki je še malo manjši, uporablja pa dve urini fazi.

V nadaljevanju dela se posvetimo še analizi delovanja, kjer pokažemo, da struktura resnično deluje pravilno. Nato sledi še zadnja primerjava z obstoječimi rešitvami iz literature in končna ocena pretvornika.

6.6 Simulacija strukture

Za začetek si pogledjmo delovanje 4-bitnega reverzibilnega pretvornika (glej sliko 6.13a), ki za svoje delovanje uporablja samo eno urino fazo. Signal se skozi strukturo pretvornika razširja iz modro obarvanih vhodnih celic B_3 , B_2 , B'_2 , B_1 , B'_1 in B_0 in napreduje do rumeno obarvanih izhodnih celic G_3 , G_2 , G_1 in G_0 , iz katerih razberemo končni rezultat, ter dveh odvečnih izhodov P_2 in P_1 . Na poti signal spreminja svojo vrednost glede na majoritetna vrata in postavitev fiksno polariziranih celic v oranžni barvi. Ker smo predhodno že definirali delujoča Feynmanova vrata predvidevamo, da naš pretvornik, ki sestoji iz treh paralelno vezanih Feynmanovih vrat, deluje pravilno.

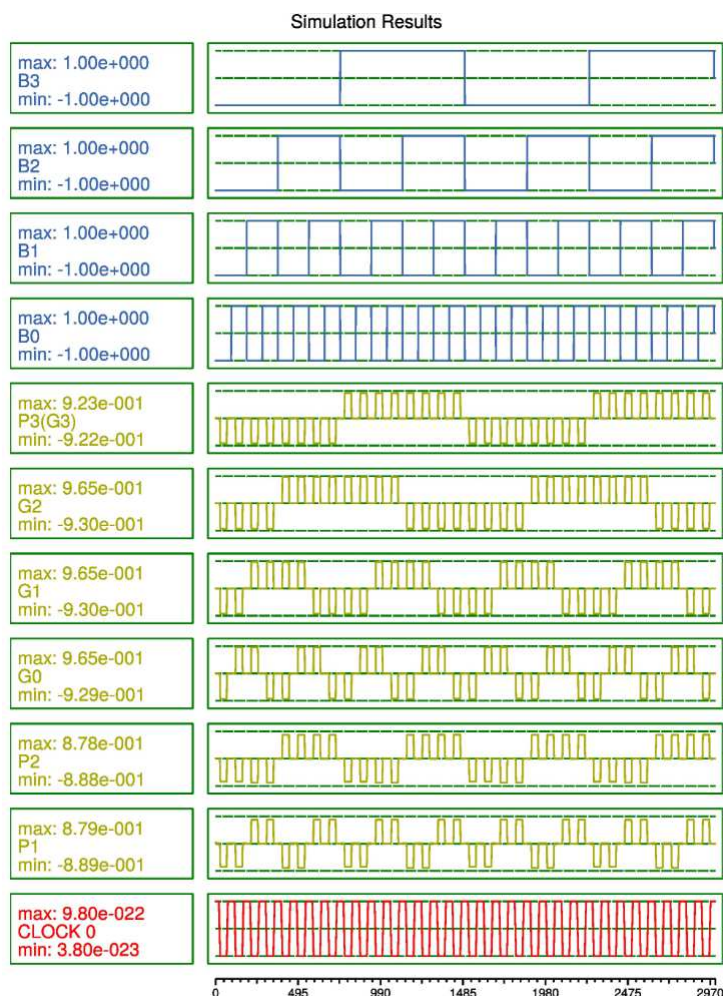
Simulacijo pretvornika izvedemo s pomočjo programskega orodja QCADesigner. Za simulacijo izberemo pristop s koherentnim vektorjem — enak pristop, kot je uporabljen pri simulaciji osnovne rešitve. Parametre, ki vplivajo na potek simulacije, prav tako ponovno pustimo na privzetih vrednostih. Rezultati simulacije so predstavljeni na sliki 6.14. Modri grafi predstavljajo vhode B_3 , B_2 , B_1 in B_0 (vhoda B'_2 in B'_1 sta enaka vhodoma B_2 in B_1 , zato ju izpustimo), rumeni grafi pa izhode G_3 , G_2 , G_1 in G_0 in oba odvečna izhoda P_2 in P_1 , ki za končni rezultat sicer nista potrebna. Rdeč graf prikazuje potek urinega signala v strukturi pretvornika. Ker pretvornik za svoje delovanje uporablja samo eno urino fazo, smo v rezultate vključili le prvi urin signal *CLOCK0*.

V kolikor primerjamo grafe G_3 , G_2 , G_1 in G_0 s tabelo 6.1 vidimo, da so rezultati enaki in tako tudi pravilni. Potrdimo lahko, da naš 4-bitni reverzibilni pretvornik iz binarne v Grayevo kodo deluje pravilno.

6.7 Ocena rešitve

Za lažjo predstavitev naše rešitve ponovno vpeljemo tabelo z ovrednotenimi rešitvami iz literature in našo osnovno rešitvijo. V tabelo dodamo tudi naši končni rešitvi z izpolnjenimi metrikami (glej tabelo 6.3).

Na tem mestu nadaljujemo s primerjavo našega 4-bitnega reverzibilnega pretvornika, ki za svoje delovanje uporablja le eno urino fazo, z rešitvami iz literature. Rezultati



Slika 6.14: Rezultati delovanja 4-bitnega reverzibilnega pretvornika iz binarne v Grayevo kodo, ki uporablja le eno urino fazo. Rezultati Grayeve kode so predstavljeni z rumenimi grafi G_3 , G_2 , G_1 in G_0 .

pretvornika so prikazani v tabeli 6.3 v 9. vrstici z oznako Naša rešitev (1). Pri izboljšanju naše rešitve smo ohranili 4-bitno pretvorbo, tako kot vse ostale rešitve. Prva metrika, ki jo primerjamo je število uporabljenih celic za izgradnjo pretvornika. Naš pretvornik je sestavljen iz 37 celic, s čimer smo boljši od vseh rešitev razen od [43], kateri pa se približamo na le tri celice. S trenutno postavitvijo našega pretvornika števila celic ne moremo več zmanjšati, smo pa s to postavitvijo veliko pridobili na naslednji metriki, ki predstavlja ploščino pravokotnika, v katerega je vpeta struktura pretvornika. V tej

	Avtor	n	št. celic	S	UF	LO	REV	komentar
1	S. Islam, et al. [39]	4	131	390	3	15	NE	le tri urine faze
2	F. Ahmad, et al. [41]	4	137	306	3	15	NE	kompaktna struktura
3	D. Bhowmik, et al. [42]	4	124	324	4	14	NE	majhna struktura, štiri urine faze
4	P. Vijayalakshmi, et al. [43]	4	34	121	1	6	NE	izredno majhna velikost
5	A. H. Majeed, et al. [44]	4	56	138	2	9	NE	izredno majhna velikost
6	E. T. Karkaj, et al. [45]	4	108	232	4	11	DA	majhna velikost reverzibilne strukture
7	J. I. Reshi, et al. [46]	4	111	252	4	15	DA	kompaktna reverzibilna struktura
8	Osnovna rešitev	4	103	261	2	15	NE	samo dve urini fazi, relativno majhna struktura
9	Naša rešitev (1)	4	37	80	1	3	DA	majhna struktura, ena urina faza, rev., malo LO
10	Naša rešitev (2)	4	35	72	2	3	DA	še manjša struktura, dve urini fazi, rev., malo LO

Tabela 6.3: Primerjava naših končnih rešitev z ostalimi rešitvami. n predstavlja število znakov pretvorbe, stolpec z oznako št. celic pove, koliko kvantnih celic je uporabljenih za rešitev, S označuje površino, ki jo zasede struktura, UF pa ponazarja število urinih faz, uporabljenih v strukturi. Število logičnih operacij je prikazano v stolpcu z oznako LO. Oznaka REV nam pove, ali je struktura reverzibilna ali ne.

metriki smo občutno boljši od vseh analiziranih rešitev iz literature. Od drugouvrščene rešitve [43] smo boljši za kar tretjino velikosti ploščine. Opazimo tudi, da smo našo osnovno rešitev večkratno izboljšali glede na število celic in površino strukture.

Naslednja metrika je število uporabljenih urinih faz. Naša rešitev število uporabljenih urinih faz minimizira na le eno uporabljeno, s čimer smo boljši od vseh rešitev razen od [43], s katero se izenačimo.

Sledi metrika, ki pove število logičnih operacij v strukturi pretvornika. V naši strukturi uporabimo le tri logične operacije (troje majoritetnih vrat). V tej metriki smo boljši od vseh analiziranih rešitev. Naslednja najboljša rešitev je ponovno [43], ki v svoji strukturi uporablja šest logičnih operacij.

Zadnja metrika pove ali je rešitev reverzibilna, ali ne. Cilj tega magistrskega dela je izgradnja reverzibilnega pretvornika iz binarne v Grayevo kodo, kar smo tudi dosegli s pomočjo uporabe reverzibilnih Feynmanovih vrat, ki predstavljajo osnovo našega pretvornika. Na tem mestu tako potrdimo, da je naša rešitev boljša od vseh analiziranih nereverzibilnih rešitev.

Na podlagi pridobljenih rezultatov ocenimo kvaliteto naše rešitve. Vse metrike razen števila celic izenačimo ali izboljšamo v primerjavi z rezultati iz literature. V številu uporabljenih celic smo slabši le od rešitve [43], ampak še to le za tri celice. Prav tako kljub

večjemu številu uporabljenih celic občutno pridobimo na površini, ki jo naš pretvornik zasede v primerjavi z ostalimi rešitvami. Uspelo nam je postaviti robusten pretvornik, ki potrebuje majhno število celic, zasede majhno površino, hitro procesira zaradi uporabe le ene urine faze (ni daljših zakasnitev) in majhne količine logičnih operacij, prav tako pa je reverzibilen in na ta način zmanjša izgubo energije in disipacijo toplote v okolje.

Omenimo še, da smo izvedli simulacijo tudi s pretvornikom, ki za delovanje uporablja dve urini fazi (glej sliko 6.13b). Rezultati pretvornika so prikazani v tabeli 6.3 v 10. vrstici z oznako Naša rešitev (2). Pri tem pretvorniku število uporabljenih celic in površino še zmanjšamo, vpeljemo pa eno urino fazo več. Rezultate simulacije in strnjen opis primerjave z ostalimi rešitvami iz literature je dostopen v prilogi magistrskega dela.

7 Zaključek

Tekom magistrskega dela smo predstavili kvantne celične avtomate, reverzibilnost in Grayevo kodo. Na podlagi kvantnih celičnih avtomatov smo nato zgradili pretvornik iz binarne v Grayevo kodo s pomočjo programskega orodja QCADesigner. Zatem je sledila simulacija modela, da smo se prepričali o pravilnem delovanju pretvornika. Prvo instanco pretvornika smo nato primerjali z nekaj že obstoječimi rešitvami iz literature na podlagi predhodno definiranih metrik (površina strukture, število uporabljenih celic v strukturi, število uporabljenih urinih faz, reverzibilnost strukture in število logičnih operacij v strukturi). Sledile so postopne nadgradnje strukture našega pretvornika, dokler nismo vseh ocen metrik izboljšali ali vsaj približno izenačili (v primeru, da se metrike ni dalo izboljšati) z ocenami obstoječih pretvornikov. V sklopu teh nadgradenj smo definirali novo strukturo XOR logičnih vrat (slika 6.7a) in Feynmanovih vrat (slika 6.8a).

Končna rešitev sta dva različna QCA pretvornika iz binarne v Grayevo kodo, ki zavzameta zelo majhno površino, sta zgrajena iz majhnega števila celic, za svoje delovanje uporabljata eno ali dve urini fazi (v tem je razlika med obema pretvornikoma), sta logično reverzibilna na nivoju gradnikov in v svoji strukturi uporabljata le tri logične

operacije. Za oba pretvornika smo opravili tudi simulacije in z rezultati simulacij dokazali, da delujeta pravilno. Na podlagi končnih rezultatov lahko rečemo, da smo boljši od obstoječih QCA pretvornikov iz binarne v Grayevo kodo, in da smo na ta način tudi uspešno rešili zadani problem magistrskega dela.

LITERATURA

- [1] GL Snider, AO Orlov, I Amlani, X Zuo, GH Bernstein, CS Lent, JL Merz, and W Porod. Quantum-dot cellular automata: Review and recent experiments. *Journal of Applied Physics*, 85(8):4283–4285, 1999.
- [2] M Mitchell Waldrop. More than Moore. *Nature*, 530(7589):144–148, 2016.
- [3] University of Florida. The reversible and quantum computing group (Revcomp). <https://www.cise.ufl.edu/research/revcomp/>. Zadnji dostop: 4.8.2018.
- [4] Marek Perkowski, Lech Jozwiak, Pawel Kerntopf, Alan Mishchenko, Anas Al-Rabadi, Alan Coppola, Andrzej Buller, Xiaoyu Song, Svetlana Yanushkevich, Vlad P Shmerko, et al. A general decomposition for reversible logic. *PDXScholar, Portland State University*, 2001.
- [5] Craig S. Lent, G. L. Snider, G. Bernstein, W. Porod, A. Orlov, M. Lieberman, T. Fehlner, M. Niemier, and P. Kogge. Quantum-dot cellular automata. In *Electron Transport in Quantum Dots*, pages 397–431. Springer, 2003.
- [6] Gregory L. Snider, Alexei O. Orlov, Ravi K. Kummamuru, Rajagopal Ramasubramaniam, Islamshah Amlani, Gary H. Bernstein, and Craig S. Lent. Quantum-dot cellular automata. *MRS Online Proceedings Library Archive*, 696, 2001.
- [7] Weiqiang Liu, Liang Lu, Máire O’Neill, and Earl E. Swartzlander. Design rules for quantum-dot cellular automata. *2011 IEEE International Symposium of Circuits and Systems (ISCAS)*, pages 2361–2364, 2011.
- [8] C. S. Lent. The concept of quantum-dot cellular automata. In Massimo Macucci, editor, *Quantum Cellular Automata: Theory, Experimentation and Prospects*, pages 1–15. Imperial College Press, 2006.

- [9] Wei Wang, Konrad Walus, and Graham A Jullien. Quantum-dot cellular automata adders. In *Nanotechnology, 2003. IEEE-NANO 2003. 2003 Third IEEE Conference on*, volume 1, pages 461–464. IEEE, 2003.
- [10] Somaye Mohammadyan, Shaahin Angizi, and Keivan Navi. New fully single layer QCA full-adder cell based on feedback model. *International Journal of High Performance Systems Architecture*, 5(4):202–208, 2015.
- [11] M. Momenzadeh, J. Huang, and F. Lombardi. QCA. In *Design and Test of Digital Circuits by Quantum-Dot Cellular Automata*, pages 37–67. Artech House, 2008.
- [12] Whitney J. Townsend and Jacob A. Abraham. Complex gate implementations for quantum dot cellular automata. In *Nanotechnology, 2004. 4th IEEE Conference on*, pages 625–627. IEEE, 2004.
- [13] University of Erlangen-Nürnberg. Quantum-dot cellular automata (QCA). https://www3.informatik.uni-erlangen.de/Research/KOMINA/QCA_script.pdf. Zadnji dostop: 30.7.2018.
- [14] M. T. Niemier and P. M. Kogge. Origins and motivations for design rules in QCA. In S. K. Shukla and R. Iris Bahar, editors, *Nano, Quantum and Molecular Computing*, pages 267–293. Kluwer Academic Publishers, 2004.
- [15] Miha Mraz. Kvantni celični avtomati. http://lrss.fri.uni-lj.si/sl/teaching/npmp/lectures/2_QCA.pdf, 2017. Zadnji dostop: 27.10.2018.
- [16] K. Walus, G. A. Jullien, and V. S. Dimitrov. Computer arithmetic structures for quantum cellular automata. In *Signals, Systems and Computers, 2004. Conference Record of the Thirty-Seventh Asilomar Conference on*, volume 2, pages 1435–1439. IEEE, 2003.
- [17] Konrad Walus, Timothy J. Dysart, Graham A. Jullien, and Arief R. Budiman. QCA-Designer: A rapid design and simulation tool for quantum-dot cellular automata. *IEEE transactions on nanotechnology*, 3(1):26–31, 2004.
- [18] Frank Sill Torres, Robert Wille, Philipp Niemann, and Rolf Drechsler. An energy-aware model for the logic synthesis of quantum-dot cellular automata. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2018.

- [19] Kalyan S. Perumalla. *Introduction to reversible computing*. Chapman and Hall/CRC, 2013.
- [20] Kenichi Morita. Reversible computing and cellular automata—a survey. *Theoretical Computer Science*, 395(1):101–131, 2008.
- [21] Charles H. Bennett. Notes on Landauer’s principle, reversible computation, and Maxwell’s demon. *Studies In History and Philosophy of Science Part B: Studies In History and Philosophy of Modern Physics*, 34(3):501–510, 2003.
- [22] Ralph C. Merkle. Two types of mechanical reversible logic. *Nanotechnology*, 4(2):114, 1999.
- [23] Saleem Mohammed Ridha Taha. *Reversible Logic Synthesis Methodologies with Application to Quantum Computing*. Springer, Cham, Switzerland, 2016.
- [24] Miha Mraz. Reverzibilnost. http://lrss.fri.uni-lj.si/sl/teaching/npmp/lectures/3_Reversible.pdf, 2017. Zadnji dostop: 27.10.2018.
- [25] Vivek V. Shende, Aditya K. Prasad, Igor L. Markov, and John P. Hayes. Synthesis of reversible logic circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 22(6):710–722, 2003.
- [26] Papiya Biswas, Namit Gupta, and Nilesh Patidar. Basic reversible logic gates and it’s QCA implementation. *Int. Journal of Engineering Research and Applications*, 4(6):12–16, 2014.
- [27] Paul Vitányi. Time, space, and energy in reversible computing. In *Proceedings of the 2nd conference on Computing frontiers*, pages 435–444. ACM, 2005.
- [28] Michael P. Frank. Introduction to reversible computing: motivation, progress, and challenges. In *Proceedings of the 2nd Conference on Computing Frontiers*, pages 385–390. ACM, 2005.
- [29] Rolf Landauer. Irreversibility and heat generation in the computing process. *IBM journal of research and development*, 5(3):183–191, 1961.
- [30] Horst Stöcker. *Matematični priročnik z osnovami računalništva*. Tehniška založba Slovenije, Ljubljana, 2006.

- [31] Robert W. Doran. The Gray code. *Journal of Universal Computer Science*, 13(11):1573–1597, 2007.
- [32] Frank Gray. Pulse code communication, March 17 1953. US Patent 2,632,058.
- [33] F. G. Heath. Origins of the binary code. *Scientific American*, 227(2):76–83, 1972.
- [34] Zijian Zhang, Yangyang Dong, Fenglei Ni, Minghe Jin, and Hong Liu. A method for measurement of absolute angular position and application in a novel electromagnetic encoder system. *Hindawi, Journal of Sensors*, 2015(1):1–10, 2015.
- [35] Gray code basics. <https://www.allaboutcircuits.com/technical-articles/gray-code-basics/>. Zadnji dostop: 26.2.2018.
- [36] Gray code — binary to Gray code and that to binary conversion. <https://www.electrical4u.com/gray-code-binary-to-gray-code-and-that-to-binary-conversion/>. Zadnji dostop: 8.2.2018.
- [37] Erik Agrell, Johan Lassing, Erik G. Strom, and Tony Ottosson. On the optimality of the binary reflected Gray code. *IEEE Transactions on Information Theory*, 50(12):3170–3182, 2004.
- [38] Logisim. <http://www.cburch.com/logisim/>. Zadnji dostop: 11.3.2018.
- [39] Shifatul Islam, Mohammad Abdullah Al Shafi, and Ali Newaz Bahar. Implementation of binary to Gray code converters in quantum dot cellular automata. *Chitakra University Publications*, 3(2):145–160, 2015.
- [40] Ajay Kumar and Sanjeev Kumar Raghuvanshi. Implementation of optical Gray code converter and even parity checker using the electro-optic effect in the Mach–Zehnder interferometer. *Optical and Quantum Electronics*, 47(7):2117–2140, 2015.
- [41] Firdous Ahmad and G. Bhat. Novel code converters based on quantum-dot cellular automata (QCA). *International Journal of Science and Research*, 3(5):364–371, 2012.
- [42] Dhurbajyoti Bhowmik, Apu Kr Saha, and Paramartha Dutta. A novel design and implementation of binary to Gray code converters up to 4-bit by quantum dot cellular

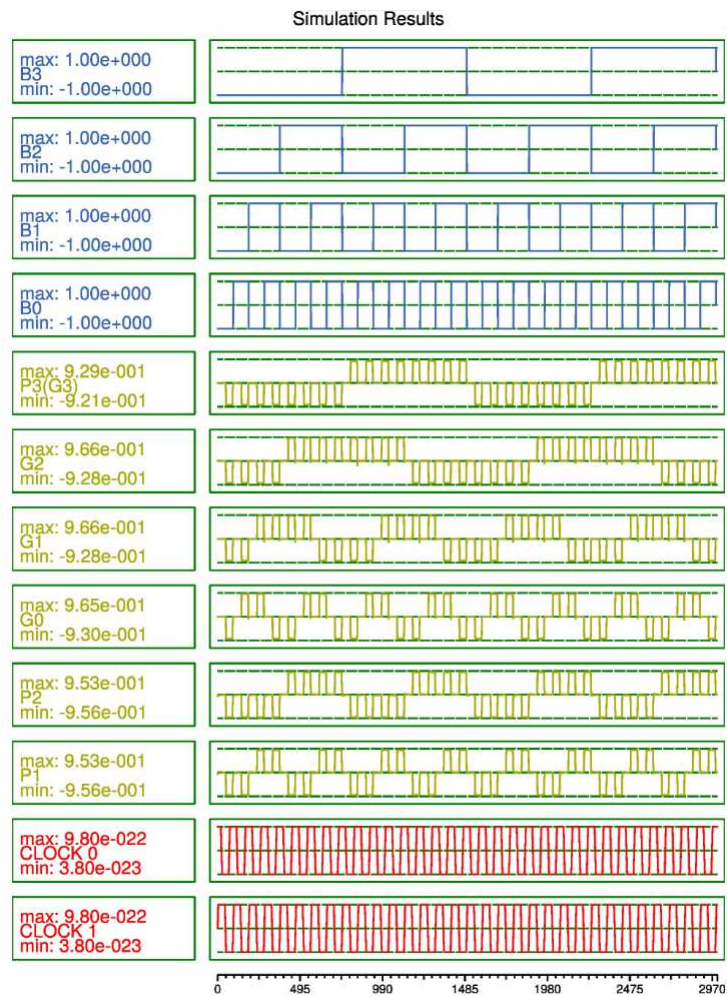
- automata. *International Journal of Control Theory and Applications*, 9(41):697–707, 2016.
- [43] P. Vijayalakshmi and V. Monisha. Design of efficient novel XOR and a code converter using QCA with minimum number of cells. *International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering*, 5(5), 2017.
- [44] Ali H. Majeed. A novel design binary to Gray converter with QCA nanotechnology. *International Journal of Advance Engineering and Research Development*, 4(9), 2017.
- [45] Ehsan Taher Karkaj and Saeed Rasouli Heikalabad. Binary to Gray and Gray to binary converter in quantum-dot cellular automata. *Optik-International Journal for Light and Electron Optics*, 130:981–989, 2017.
- [46] J. Iqbal Reshi and M. Tariq Banday. Efficient design of reversible code converters using quantum dot cellular automata. *Journal of Nano-and Electronic Physics*, 8(2):2042–1, 2016.
- [47] Ali Newaz Bahar, Sajjad Waheed, and Md Ahsan Habib. A novel presentation of reversible logic gate in quantum-dot cellular automata (QCA). In *International Conference on Electrical Engineering and Information & Communication Technology (ICEEICT)*, pages 1–6. IEEE, 2014.
- [48] Jin-Seong Lee, Young-Won You, and Jun-Cheol Jeon. Implementation of QCA XOR gate based on interaction between QCA cells. In *Manila International Conference on Trends in Engineering and Technology (MTET-17)*, Philippine, pages 81–84, 2017.
- [49] Radhouane Laajimi, Ali Ajimi, Lamjed Touil, and Ali Newaz Bahar. A novel design for XOR gate used for quantum-dot cellular automata (QCA) to create a revolution in nanotechnology structure. *International Journal of Advanced Computer Science and Applications*, 8(10):279–287, 2017.
- [50] Ali Newaz Bahar, Sajjad Waheed, Nazir Hossain, and Md Asaduzzaman. A novel 3-input XOR function implementation in quantum dot-cellular automata with energy dissipation analysis. *Alexandria Engineering Journal*, 57(2):729–738, 2017.

A Priloge

V prilogi se nahaja simulacija in analiza druge končne rešitve — reverzibilnega QCA pretvornika iz binarne v Grayevo kodo, ki za svoje delovanje uporablja dve urini fazi. Prav tako smo v priloge uvrstili vse poizkuse kreiranja XOR logičnih vrat in QCA pretvornika iz binarne v Grayevo kodo, ki smo jih testirali tekom izdelave magistrskega dela.

A.1 Simulacija reverzibilnega QCA pretvornika, ki za svoje delovanje uporablja dve urini fazi

Rezultati 4-bitnega reverzibilnega pretvornika iz binarne v Grayevo kodo, ki uporablja dve urini fazi so vidni na sliki A.1. Shema pretvornika je prikazana na sliki 6.13b. Vhoda B_3 , B_2 , B_1 in B_0 ponovno predstavljajo modro obarvani grafi (vhoda B'_2 in B'_1 sta enaka vhodoma B_2 in B_1 , zato ju izpustimo), rezultate G_3 , G_2 , G_1 in G_0 pa rumeno obarvani grafi. Prav tako sta z rumenim grafom predstavljena odvečna izhoda P_2 in P_1 . Poleg urinega signala $CLOCK0$ je dodan tudi urin signal $CLOCK1$, saj se v tem primeru rešitve vsi izhodi berejo v urinem signalu $CLOCK1$ (izhodi so v tej rešitvi postavljeni v drugo urino fazo).



Slika A.1: Rezultati 4-bitnega reverzibilnega pretvornika iz binarne v Grayevo kodo. Za svoje delovanje uporablja dve urini fazi. Število celic in površina je še dodatno zmanjšana glede na 6.14. Rezultati Grayeve kode so predstavljeni z rumenimi grafi G_3 , G_2 , G_1 in G_0 .

Če primerjamo dobljene rezultate iz slike A.1 s tistimi iz tabele 6.1 vidimo, da so si enaki. Iz tega lahko izpeljemo, da tudi pretvornik deluje pravilno. Nekatere metrike glede na prvotno rešitev (pretvornik z eno urino fazo) še izboljšamo. Število uporabljenih celic iz 37 pade na 35, prav tako dodatno zmanjšamo površino, ki iz 80 celic pade na 72 celic. Tu še vedno ne izenačimo ali izboljšamo število uporabljenih celic z rešitvijo [43], ki

uporablja le 34 celic, smo pa zato dodatno zmanjšali površino (skoraj dvakratno). Prav tako poslabšamo metriko urinih faz, saj sedaj namesto ene uporabljamo dve urini fazi — cena, ki jo plačamo z nadaljno minimizacijo strukture pretvornika. Število logičnih operacij obdržimo enako, kot tudi samo reverzibilnost rešitve.

Pojavi se nam vprašanje — katera izmed obeh naših rešitev je boljša? Menimo, da je to stvar debate. Odločiti se je potrebno ali želimo prihraniti na prostoru (uporabimo rešitev z dvema urinima fazama), ali želimo strukturo, ki bo hitreje procesirala (uporabimo rešitev, ki uporablja le eno urino fazo).

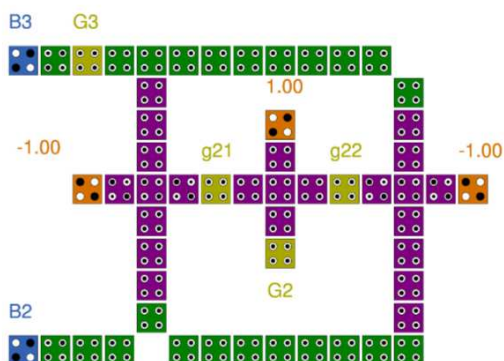
A.2 Poizkusi gradnje QCA struktur

V pričujočem razdelku je prikazanih več različnih struktur, ki smo jih zgradili tekom izdelave magistrskega dela. Predstavljajo neuspele teste, poizkuse gradnje, zavrnjene hipoteze ali pa preprosto ne dovolj dobre rešitve s katerimi smo se srečali ob delu.

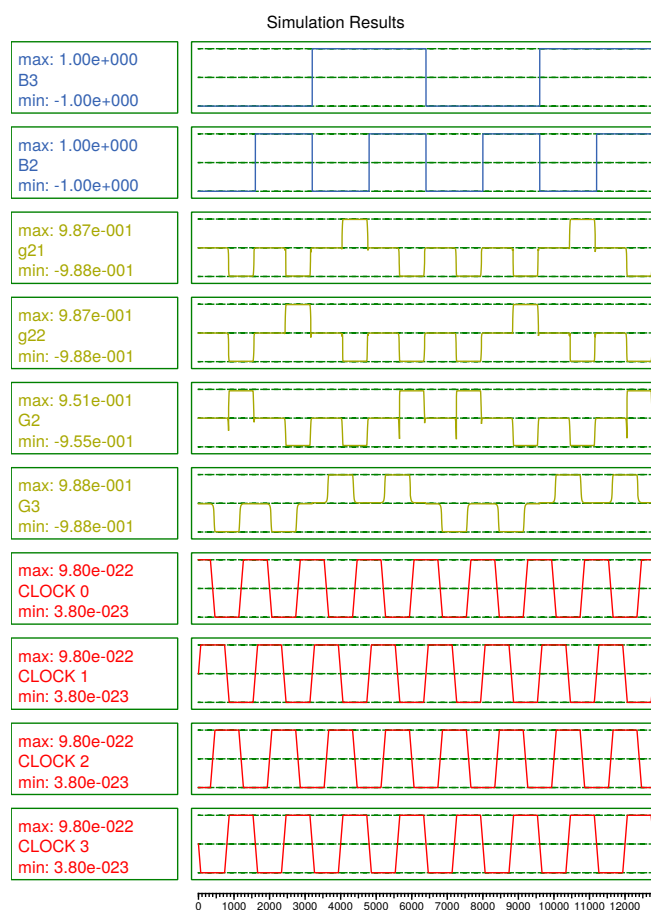
A.2.1 2-bitni QCA pretvornik iz binarne v Grayevo kodo

Na sliki A.2 je prikazan neuspeh poizkusa kreiranja 2-bitnega pretvornika iz binarne v Grayevo kodo. Rezultati simulacije so prikazani na sliki A.3. Nadaljnji komentarji za to strukturo so:

- Simulacija je izvedena s koherentnim vektorjem.
- Rezultat G_3 je pravilen, saj se samo prepíše iz vhoda B_3 (MSB). Rezultat G_2 pa je ravno invertiran (glej rezultate na sliki A.3). Kjer je signal enak 1, bi moral biti enak 0 in obratno. Vmesna izhoda g_{21} in g_{22} nakazujeta, da je signal do zadnjih majoritetnih vrat pravilen, nato pa se izkaže, da zadnja majoritetna vrata izvedejo namesto OR funkcije kar NOR funkcijo. Kjer sta vhoda enaka 0 bo rezultat enak 1, v kolikor sta različna pa je rezultat enak 0.
- Težava se izkaže v zadnjih majoritetnih vratih, ki bi morala izvajati OR funkcijo nad g_{21} in g_{22} .
- Prednosti: uporablja samo dve urini fazi.
- Slabosti: deluje samo na nivoju dveh bitov, napaka pri zadnjih majoritetnih vratih.



Slika A.2: 2-bitni QCA pretvornik iz binarne v Grayevo kodo. Uporablja dve urini fazi.

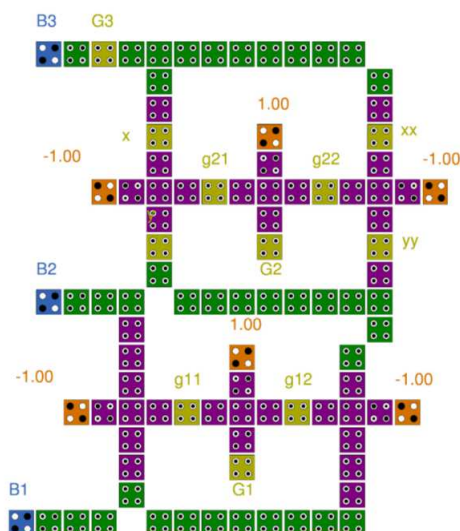


Slika A.3: Rezultati 2-bitnega QCA pretvornika iz binarne v Grayevo kodo.

A.2.2 3-bitni QCA pretvornik iz binarne v Grayevo kodo

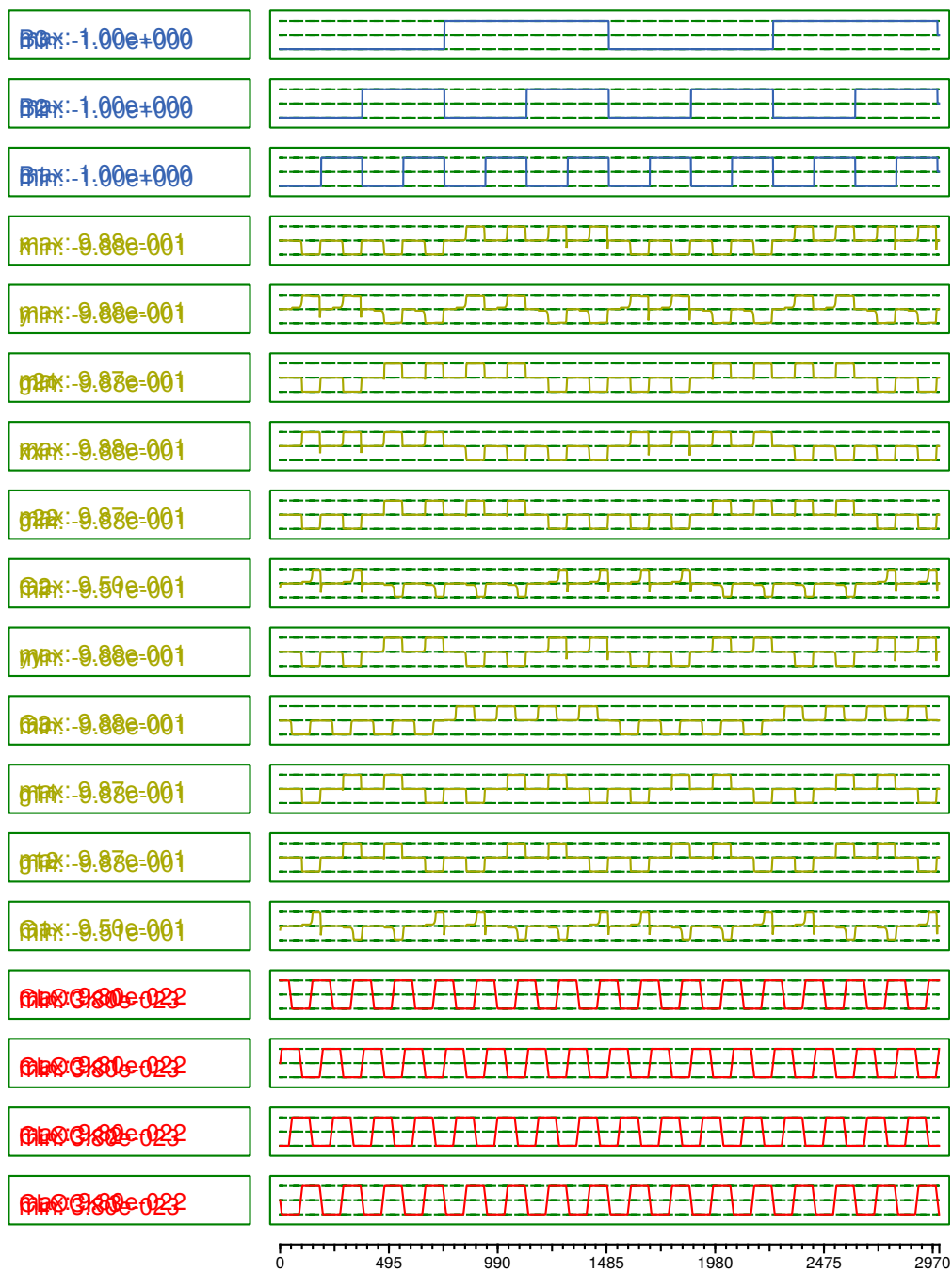
Testirali smo tudi gradnjo 3-bitnega QCA pretvornika, ki je viden na sliki A.4. Rezultati so prikazani na sliki A.5. Nadaljnji komentarji za to strukturo so:

- Simulacija je izvedena s koherentnim vektorjem.
- Struktura deluje v treh urinih fazah. V tretji urini fazi se nahajata izhoda G_1 in G_2 .
- Pri testiranju smo opazili težavo. Vhod B_3 se je pred vstopom v prva majoritetna vrata negiral pri določenih stanjih vhoda (vhod 110 se je spremenil v 010 oz. vhod 111 se je spremenil v 011).
- Pojavil se je podoben problem kot pri 2-bitnem QCA pretvorniku. Izhoda G_2 in G_1 sta bila invertirana, G_3 pa je bil pravilen.
- Prednosti: Namesto vseh štirih urinih faz uporablja samo tri.
- Slabosti: Deluje le na nivoju treh bitov, pojavila se je težava z invertiranjem izhoda.



Slika A.4: 3-bitni QCA pretvornik iz binarne v Grayevo kodo. Uporablja tri urine faze.

Simulation Results

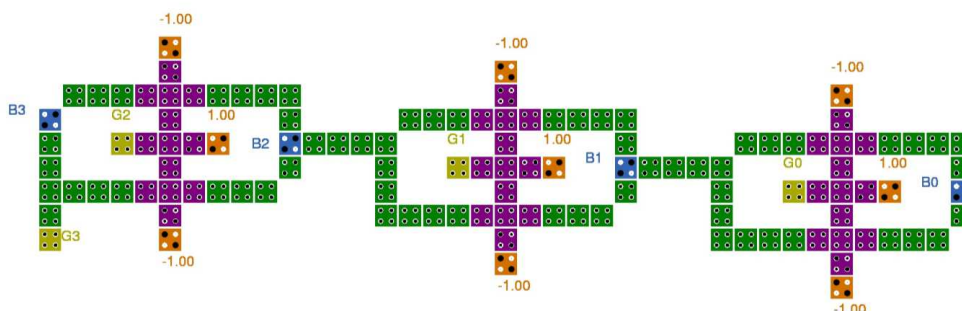


Slika A.5: Rezultati 3-bitnega QCA pretvornika iz binarne v Grayevo kodo.

A.2.3 4-bitni QCA pretvornik iz binarne v Grayevo kodo

To je primer prvega 4-bitnega pretvornika iz binarne v Grayevo kodo, ki je deloval pravilno. Temelji na uporabi Feynmanovih vrat. Prikazan je na sliki A.6, rezultati so vidni na sliki A.7. Nadaljnji komentarji za to strukturo so:

- Simulacija je izvedena s koherentnim vektorjem.
- Pristop gradnje Feynmanovih vrat je klasičen — dva negatorja, dve AND vrati in ena OR vrata.
- Vhodi B_2 , B_1 , B_0 se uporabljajo kot vez med preostalimi Feynmanovimi vrati (glej sliko A.6).
- Prednosti: uporablja samo dve urini fazi, sestavljen je iz samo 119 celic.
- Slabosti: zavzame precej veliko površino (429 celic).
- Izboljšave: zmanjšaj površino in število celic, kjer je to možno.

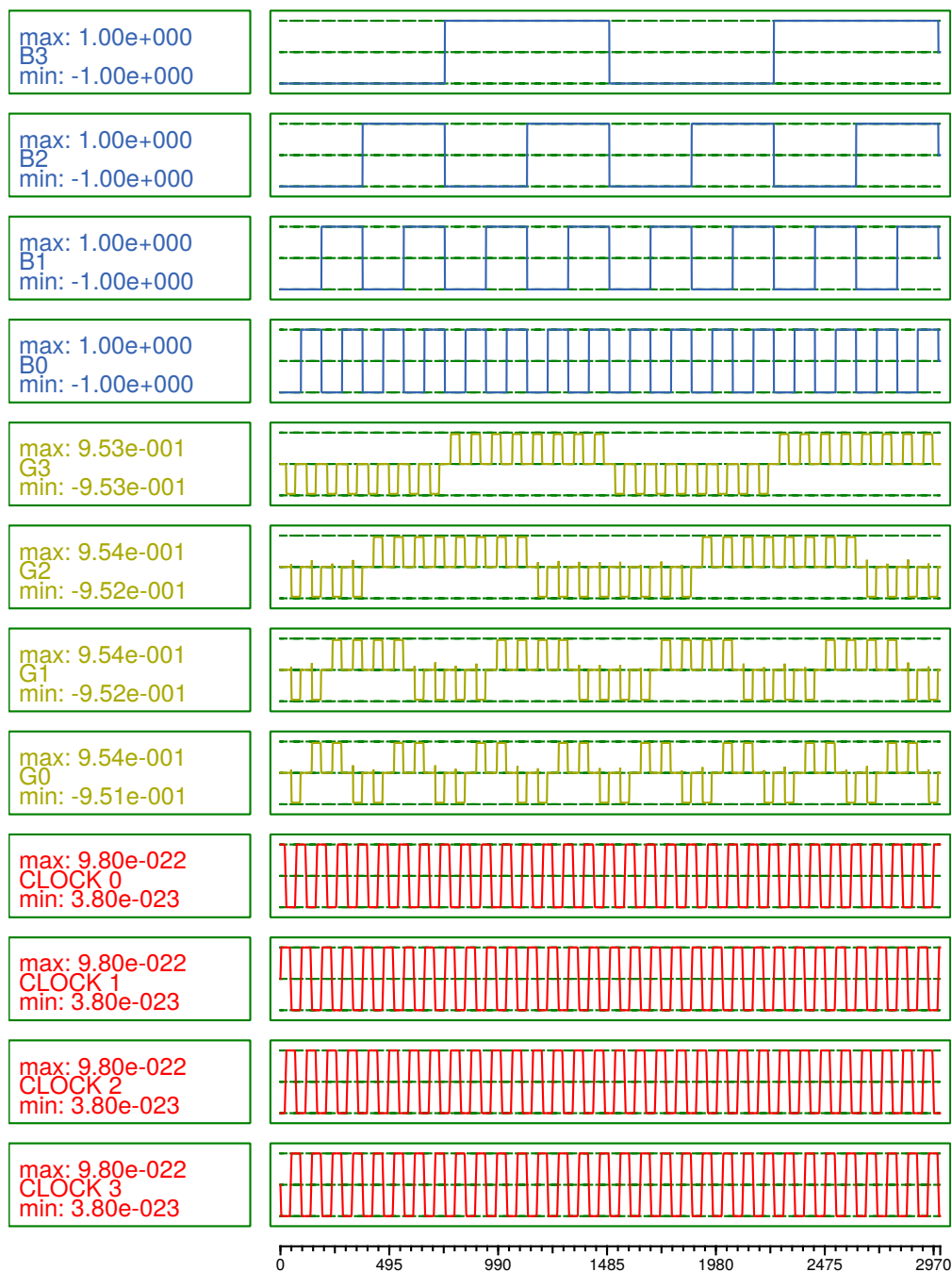


Slika A.6: 4-bitni pretvornik iz binarne v Grayevo kodo na podlagi Feynmanovih vrat. Uporablja dve urini fazi.

A.2.4 Poizkusi gradnje alternativnih XOR logičnih vrat

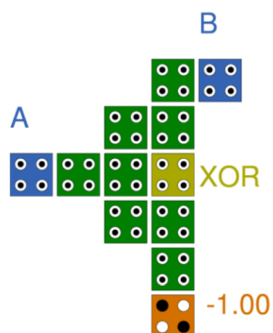
V pričujočem razdelku je prikazanih nekaj poizkusov gradnje XOR logičnih vrat, ki za svoje delovanje ne predvidevajo uporabe dveh negatorjev, dveh AND vrat in enih OR vrat. Na ta način smo skušali priti do novih XOR logičnih vrat, ki bi zavzemale manjšo površino in bi bila sestavljena iz manjšega števila celic. Nadaljnji komentarji pri teh poizkusih so:

Simulation Results

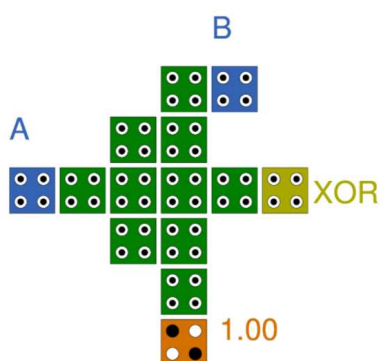


Slika A.7: Rezultati 4-bitnega pretvornika iz binarne v Grayevo kodo na podlagi Feynmanovih vrat.

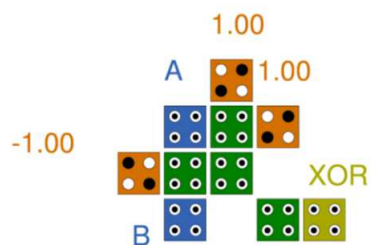
- Na slikah A.8 in A.9 smo poizkusili zgraditi delujoča XOR logična vrata s pomočjo fizične polarizacije 0 in 1, vendar smo vedno dobili končne rezultate enake AND ali OR funkciji. Poizkusili smo več različnih razporeditev celic.
- Na sliki A.10 smo poizkusili zgraditi vrata XOR s pomočjo več fiksno polariziranih celic. Dobili smo zanimive rezultate (npr. izhod je bil 1 samo v primeru ko sta bila $A = 1$ in $B = 0$), ni pa nam s to konfiguracijo celic uspelo zagotoviti pravilnega izvajanja logične funkcije XOR.
- Na sliki A.11 nam je uspelo zgraditi delujoča logična vrata XOR. Pomankljivost, ki jih imajo ta vrata, je uporaba dveh urinih faz. Želeli smo zgraditi logična vrata XOR, ki za svoje delovanje uporabljajo le eno urino fazo.



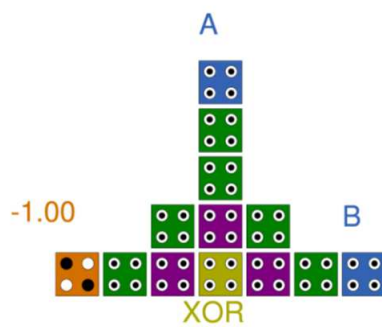
Slika A.8: Poizkus 1 — nedelujoča alternativna logična vrata XOR.



Slika A.9: Poizkus 2 — nedelujoča alternativna logična vrata XOR.



Slika A.10: Poizkus 3 — nedelujoča alternativna logična vrata XOR.



Slika A.11: Poizkus 4 — delujoča alternativna logična vrata XOR. Za svoje delovanje uporablja dve urini periodi.