

# Model realizacije funkcionalnosti RS pomnilne celice v biološkem sistemu

Miha Moškon<sup>1</sup>, Monika Ciglič<sup>2</sup>, Roman Jerala<sup>2</sup>, Nikolaj Zimic<sup>1</sup>, Miha Mraz<sup>1</sup>

<sup>1</sup>Laboratorij za računalniške strukture in sisteme, Fakulteta za računalništvo in informatiko, Univerza v Ljubljani, Tržaška 25, SI-1000 Ljubljana, Slovenija

<sup>2</sup>Laboratorij za biotehnologijo, Kemijski inštitut, Hajdrihova 19, SI-1001 Ljubljana, Slovenija

E-pošta: miha.moskon@fri.uni-lj.si

**Povzetek.** Realizacije računalniških struktur, ki predstavljajo temeljne gradnike računalniških sistemov, so v zadnjih 80-ih letih temeljile na elektronskih komponentah. S problemi, ki jih prinašata stalna miniaturizacija osnovnih entitet [1] in z večjo porabo energije računalniških sistemov kot celote, se poraja vprašanje prihodnosti tovrstnega pristopa. Ena od možnih alternativnih platform procesiranja podatkov tako postajajo tudi biološki sistemi [2]. Temelj tovrstnega procesiranja je zapis dinamike v obliki DNK zaporedij in prisotnost specifičnih proteinov kot prožilcev te dinamike. V članku predstavimo osnovne pojme procesiranja v bioloških sistemih na zgledu modela realizacije enostavne RS pomnilne celice. Za omenjeno celico postavimo bistabilni model realizacije in predstavimo simulacijske rezultate obnašanja tako v idealiziranih, kot tudi v neidealiziranih razmerah. Z izgradnjo modela so doseženi pogoji za samo realizacijo celice v biološkem gostitelju eno ali več celične narave.

**Ključne besede:** genetsko regulatorno omrežje, sintezna biologija, RS pomnilna celica, pomnjenje v bioloških sistemih, nekonvencionalno računalništvo

## The model of RS memory cell realization in biological system

**Extended abstract.** Computer structures that represent basic logic primitives for the construction of computer systems have based on electronic components for the past 80 years. Recently we are starting to face many problems with the requirements for their continuous minimization and fast response time [1]. Therefore alternative information processing capable platforms need to be found in order to satisfy the future processing demands. Biological systems are certainly one of the possible alternative processing platforms of the future [2]. Dynamics of these circuits is based on the presence (respectively absence) of specific proteins, i.e. transcriptional factors. We can manipulate these circuits with genetic engineering (i.e. DNA modifications) to achieve their desired behaviour. Here we present the basic concepts of this systems on the example of RS memory cell for which we construct a mathematical model based on Ordinary Differential Equations (ODEs). We also construct a simulation model in Matlab/Simulink environment (see figures 7 and 8). We simulate the behaviour of our circuit in ideal circumstances (see figure 9) and in the circumstances which are the approximation of real environment (see figure 10). Promising simulation results indicate that realization in a living tissue is possible and is therefore included in our future work.

**Key words:** genetic regulatory network, synthetic bio-

logy, RS memory cell, memorizing in biological systems, unconventional computing

---

### 1 Uvod

Z veljavnostjo Moorovega zakona, ki napoveduje podvajanje kompleksnosti in s tem podvajanje hitrosti delovanja mikroprocesorjev na približno vsaki dve leti [1], se velikost osnovnih sestavnih delov digitalnih struktur (tranzistorjev) iz leta v leto zmanjšuje. S takšnim trendom miniaturizacije bodo v približno desetih letih tako velikosti tranzistorjev dosegle nanometrski velikostni nivo, v katerega sodijo atomi in molekule. Ker je elektronsko pogojeno obvladovanje dinamike entitet na tem nivoju težko obvladljivo zaradi vplivov šuma, kvantnih efektov itd., smatramo, da bo računalniška tehnologija soočena z iskanjem alternativnih platform procesiranja.

Ena od obetajočih je procesiranje v bioloških sistemih, kjer je temeljni nosilec programiranja dinamike zaporedje DNK. Glavni del DNK zaporedja je

zapis pogojev za proženje generiranja proteinov in navodila za njihovo izdelavo (t.j. seznam aminokislin, ki sestavljajo posamezen protein). Na tak način si lahko DNK zaporedje interpretiramo kot računalniški program, posamezne proteine pa po eni strani kot vhodne podatke, s pomočjo katerih se samo procesiranje sproži, po drugi strani pa kot izhodne rezultate procesiranja [2].

Z realizacijo pomnjenja lahko v biološke sisteme vpeljemo sekvenčno procesiranje. V pričujočem članku opišemo pristop k realizaciji modela funkcionalnosti RS pomnilne celice v biološkem sistemu. Predstavimo vse potrebne korake za zasnovo takšne strukture, saj je tudi na področju procesiranja v bioloških medijih pomembno, da se že pred samo fizično realizacijo, ki je praviloma dolgotrajna in s tem draga, oborožimo s kopico podatkov o želenem sistemu, ki vsaj hipotetično vodijo k uspešni realizaciji.

## 2 Osnovni gradniki biološkega sistema

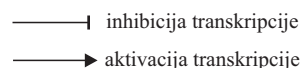
Procesiranje v bioloških sistemih lahko poteka v eni ali več gostiteljskih celicah. Brez izgube na splošnosti se bomo v naši predstavitvi omejili na hipotetično realizacijo v enoceličnem gostitelju prokariontske narave. V takšni zaključeni enoti (celici) se nahajajo naslednji pomembni gradniki, ki vplivajo na *programiranje* dinamike v njej [3]:

- **zapis DNK** je kodiran s štirimi nukleotidnimi bazami (A, G, C in T) in je poljubne dolžine; poleg kontrolnih informacij hrani osnovna navodila za generiranje proteinov;
- **kodon** je osnovni sestavni del zaporedja DNK; sestavljen je iz zaporedja poljubnih treh nukleotidnih baz (treh kodov), tako da lahko predstavlja enega od  $4^3 = 64$  možnih različnih zapisov; posamezni kodon bodisi definira tvorbo eno od dvajsetih možnih ciljnih aminokislin, ali pa ima določa t.i. *stop kodon*;
- **promotor** je del zapisa DNK, na katerega se mora vezati encim *RNK polimeraze*; s tem je izpolnjen potreben pogoj za proženje proizvodnje proteina v celici po *programu* zapisanem na ostalem delu DNK zaporedja;
- **operator** je del promotorja, na katerega se vežejo vnaprej določeni transkripcijski faktorji;
- **transkripcijski faktor** je specifičen protein, ki s svojo vezavo na operator promotorja pospešuje oziroma zavira vezavo RNK polimeraze na promotorski del DNK zapisa in s tem neposredno odloča o intenzivnosti tvorbe izhodnega proteina;

Samo dinamiko v celici, ki se manifestira v obliki tvorbe novih proteinov, delimo na fazi *transkripcije* in *translacije*. Tvorba proteinov se sproži ob vezavi RNK polimeraze na promotor DNK zapisa ob prisotnosti oziroma odsotnosti specifičnih transkripcijskih faktorjev. Po vezavi na promotor RNK polimeraza potuje vzdolž dvojne vijačnice DNK in shranjen genetski zapis prepisuje v mRNK (*sporočilno RNK*). Proces prepisovanja DNK zapisa v mRNK zapis imenujemo transkripcija. Proces transkripcije se ustavi ob stop kodonu. Po procesu transkripcije se na novo nastalo molekulo mRNK vežejo *ribosomi* (natančneje na t.i. *mesto vezave ribosoma* - RBS), s čimer se začne prevajanje mRNK zapisa v zaporedje aminokislin, ki tvorijo ciljni protein. Ta proces imenujemo translacija.

Intenzivnosti generiranja določenega proteina lahko spremljamo tako, da vzporedno z njim proizvajamo še proteine zaznavne z danes dosegljivimi tehnologijami (npr. zeleni fluorescirajoči protein ali t.i. *GFP*).

Transkripcijske faktorje glede na njihov učinek delimo na *aktivatorje*, ki vezavo RNK polimeraze pospešujejo (aktivirajo, inducirajo) in *represorje*, ki vezavo zavirajo (represirajo, inhibirajo). Prvi z vezavo na operator večajo frekvenco transkripcij mRNK, oziroma sam proces transkripcije sploh omogočijo, drugi pa proces transkripcije zavirajo. Vplive transkripcijskih faktorjev lahko grafično ponazorimo z notacijo predstavljeno na sliki 1. Ker je razgradnja mRNK molekul v večini primerov izredno hitra, lahko intenzivnost generiranja specifičnega proteina enačimo kar s frekvenco izvedenih transkripcij.



Slika 1. Grafična predstavitev vpliva represorja in aktivatorja.

Figure 1. Graphical presentation of the influence of different transcriptional factors.

Z vidika snovalca klasičnih logičnih digitalnih struktur bi lahko predhodno navedene gradnike razdelili na vhodne, procesne in izhodne entitete po naslednjem principu

- **vhodni segment**: prisotnost oziroma odsotnost proteina s funkcijo transkripcijskega faktorja,
- **decizijski segment**: zapis DNK (*program*), ki vsebuje navodila za tvorbo ciljnega izhodnega proteina,
- **izhodni segment**: novogenerirani protein, ki ima lahko tudi funkcijo transkripcijskega faktorja, ni pa to nujno.

$R$	$S$	$D^1q$	$D^1\bar{q}$
0	0	$q$	$\bar{q}$
0	1	1	0
1	0	0	1
1	1	$X$	$X$

Tabela 1. Prevajalna funkcija RS pomnilne celice.  
Table 1. Lookup table of RS memory cell.

Ostale faktorje, ki vplivajo na frekvenco transkripcij (npr. prisotnost RNK polimeraze) zagotovimo že z umestitvijo molekule DNK v ustrezno okolje.

### 3 Koraki za učinkovito postavitve željenega biološkega sistema

Postavitve zelenega biološkega sistema naj bi potekala v korakih, ki zagotavljajo optimalno izvedbo. Ti so sledeči [4]:

- postavitve modela zelene logične prevajalne funkcije, kot smo je vajeni pri klasičnem načrtovanju logičnih struktur,
- prenos modela logične prevajalne funkcije na model biološkega sistema,
- modeliranje odzivnosti biološkega sistema na osnovi numeričnih atributov hipotetične realizacije,
- fizična realizacija biološkega sistema in njegovo testiranje.

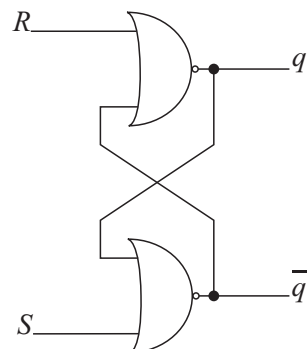
V našem prispevku se omejimo na predstavitev prvih treh korakov.

## 4 Model realizacije RS pomnilne celice v biološkem sistemu

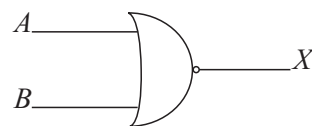
### 4.1 Model zelene logične prevajalne funkcije

Na sliki 2 je prikazana logična shema RS pomnilne celice [5]. Ima dva vhoda in dva izhoda. RESET vhod ( $R$ ) v aktivnem stanju (logična vrednost 1) postavi izhod ( $q$ ) na logično vrednost 0, SET vhod ( $S$ ) v aktivnem stanju pa postavi izhod ( $q$ ) na logično vrednost 1. Če ni aktiven nobeden od vhodov, se stanje izhodov ohranja, kar je zagotovljeno s povratno vezavo. Prehajanje med stanji na osnovi obeh aktivnih vhodov ni definirano. Delovanje RS pomnilne celice ponazarja prevajalna funkcija v tabeli 1.

Kot je razvidno iz slike 2, lahko RS celico realiziramo na osnovi para logičnih dvovhodnih NOR vrat. Logično shemo NOR vrat prikazuje slika 3, njihovo prevajalno funkcijo pa tabela 2.



Slika 2. Logična shema RS pomnilne celice.  
Figure 2. Logical outline of RS memory cell.



Slika 3. Logična shema NOR logičnih vrat.  
Figure 3. Logical outline of NOR gate.

$A$	$B$	$X$
0	0	1
0	1	0
1	0	0
1	1	0

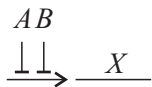
Tabela 2. Prevajalna funkcija NOR logičnih vrat.  
Table 2. Lookup table of NOR gate.

### 4.2 Prenos modela prevajalne funkcije na model biološkega sistema

Model sistema bo temeljil na biološki realizaciji logičnih NOR vrat. Slednjo začnemo snovati pri vhodih ( $A, B$ ), za kar uporabimo par transkripcijskih faktorjev. Če se po vzoru [6] omejimo na uporabo transkripcijskih faktorjev represorske narave, bi glede na prevajalno funkcijo iz tabele 2 lahko posamezna NOR vrata realizirali z *operatorjem*, ki bi prožil postopek transkripcije le ob odsotnosti obeh transkripcijskih faktorjev. Ob prisotnosti vsaj enega od dveh represorjev ( $A$  in/ali  $B$ ) bi se tako postopek transkripcije ustavil.

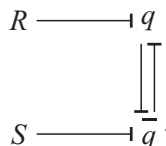
Model biološkega sistema za realizacijo NOR logičnih vrat je predstavljen na sliki 4. Pri tem  $\longrightarrow$  predstavlja promotorski del DNK,  $\underline{X}$  navodilo za izdelavo proteina  $X$  (zaporedje kodirajočega proteina), par ( $A, B$ ) predstavlja represorska proteinska vhoda (signala),  $X$  pa izhodni protein (signal), ki se generira le ob odsotnosti obeh vhodnih proteinov. Notacija je povzeta po [6].

Ob tako pridobljenih hipotetično delujočih bioloških NOR vratih sestavimo tudi model celotne RS



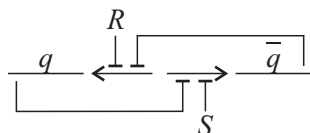
Slika 4. Model biološkega sistema posameznih NOR vrat.  
Figure 4. Model of biological system of NOR logical gate.

celice predstavljen na sliki 5.



Slika 5. Model biološkega sistema celotne RS celice.  
Figure 5. Model of biological system of RS cell.

Kot pri običajni RS celici aktivacija RESET vhoda (prisotnost represorskega proteina z oznako  $R$ ) povzroči deaktivacijo izhodnega signala (ustavi proces transkripcije izhodnega proteina  $q$ ). Prisotnost SET vhoda (prisotnost represorskega proteina z oznako  $S$ ) povzroči deaktivacijo komplementa izhodnega signala ( $\bar{q}$ ) in posledično aktivacijo izhodnega signala. Proteina  $q$  in  $\bar{q}$  se medsebojno represirata, kar nam omogoča ohranjanje stanja RS celice ob odsotnosti obeh vhodnih signalov. Protein  $q$  torej nastopa kot represor proteina  $\bar{q}$ ,  $\bar{q}$  pa kot represor proteina  $q$ . Natančnejši prikaz modela sistema biološke RS celice prikazuje slika 6. Pri tem so oznake skladne z notacijo uporabljeno v prejšnjih razdelkih.



Slika 6. Razširjeni model biološkega sistema celotne RS celice.  
Figure 6. Extended model of biological system of RS cell.

### 4.3 Modeliranje odzivnosti biološkega sistema

Delovanje biološke RS celice smo modelirali z uporabo determinističnega modela, ki določa koncentracije posameznih proteinov v posameznem trenutku z uporabo diferencialnih enačb (angl. *ordinary differential equations* - ODE), natančneje s t.i. *Hillovimi funkcijami* [7]. Koncentracijo proteina  $q$  v danem trenutku lahko izrazimo z izrazom (1), kjer  $\beta_0$  določa puščanje vrat (*leakiness*),  $\beta$  maksimalno stopnjo transkripcije proteina,  $R$  koncentracijo proteina, ki določa RESET vhod,  $\bar{q}$  koncentracijo proteina  $\bar{q}$ ,  $K$  koeficient represiranja (pri  $K$  enotah represorskega proteina, je transkripcija pol manjša od maksimalne),

$n$  koeficient kooperativnosti, ki nam pove koliko enot vhodnega proteina se mora naenkrat vezati na operator, ter  $\delta$  intenzivnost degradacije proteina. Pri tem je puščanje vrat določeno s koncentracijo proteina, ki se generira ne glede na prisotnost oziroma odsotnost transkripcijskih faktorjev.

$$\frac{dq}{dt} = \beta_0 + \frac{\beta}{1 + (\frac{R+\bar{q}}{K})^n} - \delta q. \quad (1)$$

Koncentracijo proteina  $\bar{q}$  lahko zapišemo z izrazom (2), kjer posamezni koeficienti igrajo enake vloge kot pri prvi enačbi,  $S$  pa predstavlja koncentracijo proteina, ki določa SET vhod.

$$\frac{d\bar{q}}{dt} = \beta_0 + \frac{\beta}{1 + (\frac{S+q}{K})^n} - \delta \bar{q}. \quad (2)$$

Po pregledu relevantne literature [8, 9] smo za predhodno navedene parametre izbrali vrednosti

$$\beta_0 = 0.2 \frac{\mu M}{min}, \quad (3)$$

$$\beta_1 = 4 \frac{\mu M}{min}, \quad (4)$$

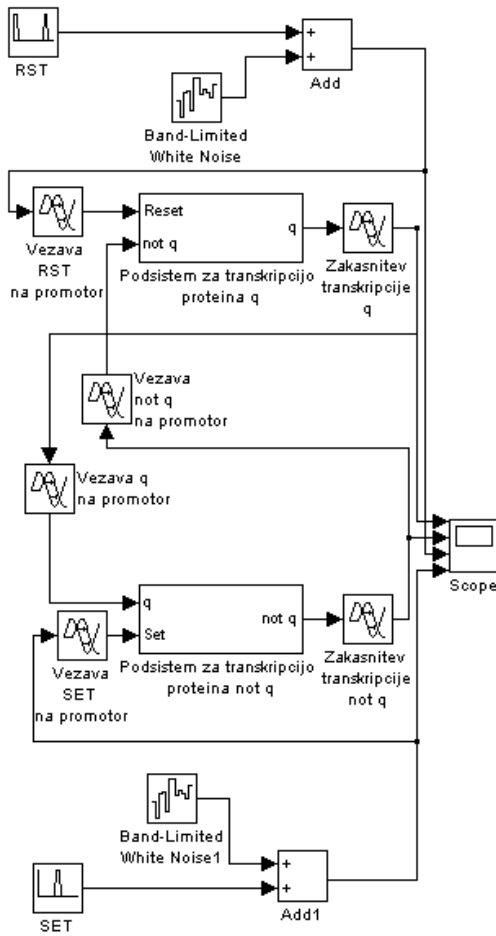
$$K = 1 \mu M, \quad (5)$$

$$n = 3, \quad (6)$$

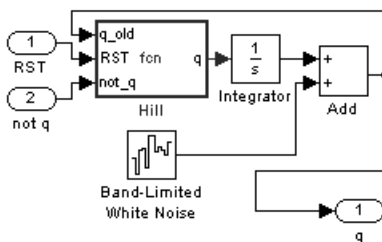
$$\delta = 1 min^{-1}, \quad (7)$$

kjer  $M$  predstavlja enoto  $\frac{mol}{L}$ .

Glede na enostavnost diferencialnih enačb smo računski model postavili v okolju Matlab/Simulink (slika 7), pri čemer gradnika  $RST$  in  $SET$  predstavljata generatorja vhodnih signalov. Vsakemu je dodan 10% šum z gradnikom tipa *Band Limited White Noise*. Modeliranje zakasnitev pri vezavi specifičnih proteinov na promotorske regije in zakasnitev pri transkripciji specifičnih proteinov je bilo implementirano s pomočjo gradnika *Transport Delay*. Podsystem, ki služi modeliranju samega procesa transkripcije proteina  $q$  je predstavljen na sliki 8. Pri tem gradnik *Hill* vsebuje Hillovo funkcijo transkripcije proteina  $q$  (glej enačbo 1), ki jo je potrebno še integrirati (gradnik *Integrator*). Rezultatu transkripcije prav tako kot vhodom dodajamo šum. Modeliranje transkripcije proteina  $\bar{q}$  je analogno modeliranju transkripcije proteina  $q$ .



Slika 7. Model sistema postavljen v okolju Matlab/Simulink.  
Figure 7. Matlab/Simulink model of the biological system.

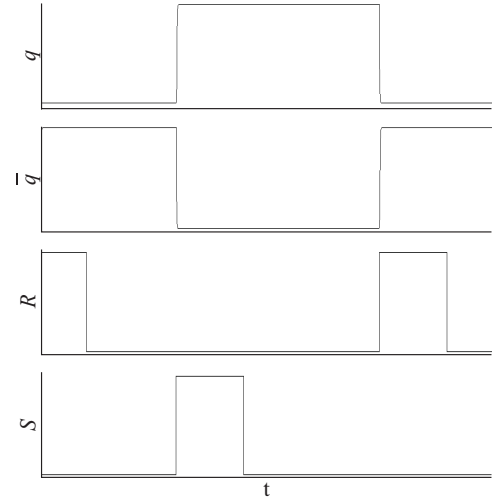


Slika 8. Podsystem za modeliranje transkripcije proteina  $q$  postavljen v okolju Matlab/Simulink.  
Figure 8. Matlab/Simulink subsystem that models the transcription process of protein  $q$ .

### 4.3.1 Simulacijski rezultati v idealnem okolju

Za idealno simulacijsko okolje smatramo razmere, pri katerih je hipotetično možen hipen vnos ustreznih koncentracij transkripcijskih faktorjev. Simulacija dinamike je prikazana na sliki 9. Pri tem  $x$  os predstavlja simulacijski čas ene ure,  $y$  osi pa po vrsti kon-

centracije izhodnega proteina  $q$ , izhodnega proteina  $\bar{q}$ , ter koncentracije vhodnih proteinov  $R$  in  $S$ .



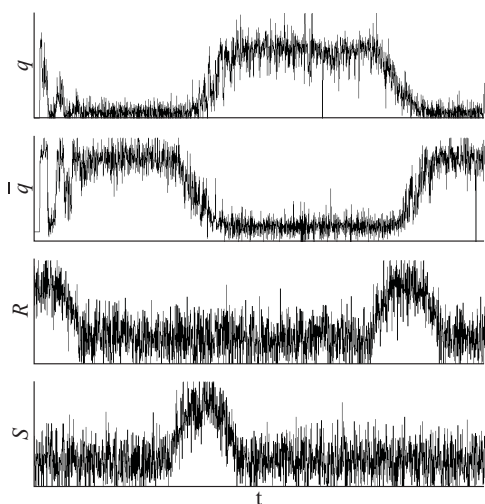
Slika 9. Simulacijski rezultati dinamike generiranja izhodnega proteina  $q$  oziroma  $\bar{q}$  v odvisnosti od koncentracije vhodnih represorjev.  
Figure 9. Simulation results that indicate the dependence of output protein  $q$  (respectively  $\bar{q}$ ) regarding to input repressors concentration.

Na začetku simulacije smo aktivirali  $R$  (RESET) vhod in deaktivirali  $S$  (SET) vhod. Kombinacija je represirala generiranje proteina  $q$ . Zaradi odsotnosti represorskih proteinov  $q$  in  $S$ , je bilo izražanje proteina  $\bar{q}$  maksimalno. Čez nekaj časa smo v simulaciji vhodni protein  $R$  odstranili, kar je povzročilo ohranjanje stanja. Represorski protein  $\bar{q}$  je preprečeval izražanje (tvorbo) proteina  $q$ . Represorska proteina, ki vplivata na izražanje proteina  $\bar{q}$  nista bila prisotna, zato je bilo izražanje le-tega še vedno maksimalno. Aktivacija vhodnega proteina  $S$  je vodila v represiranje izražanja proteina  $\bar{q}$ . Njegova transkripcija se je ustavila. Po določenem času se je protein  $\bar{q}$  razgradil do koncentracije, ki je bila premajhna za represiranje transkripcije proteina  $q$ , kar je vodilo v njegovo maksimalno izražanje. Stanje se je po odstranitvi vhodnega proteina  $S$  ohranjalo, saj je bila v danem trenutku koncentracija proteina  $q$  tako velika, da je represirala izražanje proteina  $\bar{q}$ .

### 4.3.2 Simulacijski rezultati v realnem neidealnem okolju

Za realno neidealno simulacijsko okolje smatramo razmere, pri katerih je vnos ustreznih koncentracij transkripcijskih faktorjev časovno pogojen v obliki rastoče ali padajoče funkcije koncentracije, poleg tega pa transkripcija proteina in njegova vezava na promotorska področja ni hipna, kot je bilo predpostavljeno pri idealizirani simulaciji. Koncentracijo vhodnih proteinov smo zato predstavili s funkcijama, ki

postopno prehajata iz aktivnega v neaktivno stanje in obratno, poleg tega pa smo v simulacijo vpeljali časovne zakasnitve in šum. Rezultati simulacije v realnem neidealnem okolju so predstavljeni na sliki 10, iz katerih je razvidno pravilno delovanje obravnavanega vezja. Pri tem  $x$  os predstavlja simulacijski čas ene ure,  $y$  osi pa po vrsti koncentracije izhodnega proteina  $q$ , njegovega komplementa  $\bar{q}$  ter koncentracije vhodnih proteinov  $R$  in  $S$ .



Slika 10. Simulacijski rezultati dinamike generiranja izhodnega proteina  $q$  oziroma  $\bar{q}$  v odvisnosti od koncentracije vhodnih represorjev v realnem neidealnem okolju. Figure 10. Simulation results that indicate the dependence of output protein  $q$  (respectively  $\bar{q}$ ) regarding to input repressors concentration in unideal environment.

## 5 Zaključek

V pričujočem prispevku smo prikazali osnovne korake za alternativni pristop snovanja procesno pomnilnih struktur v bioloških sistemih. Sami rezultati zasnove in modela so obetajoči in so hipotetično dobra osnova za prehod na zadnjo fazo same fizične izvedbe v gojenih celičnih strukturah. Počasnost obravnavanega vezja na prvi pogled predstavlja oviro pri postavljanju procesno zahtevnejših platform, po drugi strani pa to smatramo zgolj kot kompromis ob veliki stopnji paralelnosti, ki nam zagotavlja visoko zmogljivost tovrstnih sistemov. Avtorji pričujočega članka smo v fazi priprave same realizacije, ki bo izvedena na Kemijskem inštitutu v Ljubljani. Predpostavljamo, da bo fizična realizacija uspešna, s čimer bomo tudi v praksi potrdili našo hipotezo o možnosti realizacije procesnih primitivov v bioloških sistemih.

## 6 Zahvala

Pričujoča raziskava je bila izvedena v okviru znanstveno raziskovalnega programa *Vseprisotno računal-*

*ništvo* (P2-0359), ki ga financira Agencija za raziskave in razvoj Republike Slovenije.

## 7 Literatura

- [1] G. Moore, "Cramming more components onto integrated circuits," *Electronics*, vol. 38, 1965.
- [2] N. Ramakrishnan, U. S. Bhalla, and J. J. Tyson, "Computing with proteins," *Computer*, vol. 42, pp. 47–56, 2009.
- [3] B. R. Glick and J. Pasternack, *Molecular biotechnology: principles and applications of recombinant DNA*, 3rd Edition. ASM Press, 2003.
- [4] B. Cvetkovič, "Analiza možnosti realizacije primitivnih računalniških struktur na osnovi DNK gradnikov-diplomska naloga, univerza v Ljubljani, fakulteta za računalništvo in informatiko," 2008.
- [5] Z. Kohavi, *Switching and Finite Automata Theory*. McGraw-Hill College, 1978.
- [6] T. S. Gardner, C. R. Cantor, and J. J. Collins, "Construction of a genetic toggle switch in *Escherichia coli*," *Nature*, vol. 403, pp. 339–342, 2000.
- [7] U. Alon, *An Introduction to Systems Biology*, A. M. Etheridge, Ed. Chapman & Hall/CRC, 2007.
- [8] H. Kobayashi, M. Kaern, M. Araki, K. Chung, T. S. Gardner, C. R. Cantor, and J. J. Collins, "Programmable cells: Interfacing natural and engineered gene networks," *Proceedings of the National Academy of Sciences*, vol. 101, pp. 8414–8419, 2004.
- [9] T. Tian and K. Burrage, "Stochastic models for regulatory networks of the genetic toggle switch," *Proceedings of the National Academy of Sciences*, vol. 103, pp. 8372–8377, 2006.

**Miha Moškon** je asistent na Fakulteti za računalništvo in informatiko v Ljubljani. Raziskovalno se ukvarja s področjem alternativnih procesnih metod in struktur bodočnosti. Je (so)avtor 6 objavljenih referatov in dobitnik Prešernove nagrade FRI v letu 2006.

**Monika Ciglič** je mlada raziskovalka na Kemijskem inštitutu. Raziskovalno se ukvarja z mikrobiologijo in imunologijo. Že dvakrat je bila članica zmagovalne ekipe na mednarodnem tekmovanju iGEM (International Genetically Engineered Machine Competition), in sicer leta 2006 kot študentka in leta 2008 kot mentorica.

**Roman Jerala** je vodja Laboratorija za biotehnologijo na Kemijskem inštitutu. Raziskovalno se ukvarja z medicinsko pomembnimi procesi, predvsem z imunskim odzivom na bakterijske infekcije. Poleg tega predava Bioinformatiko na Univerzi v Ljubljani.

**Nikolaj Zimic** je redni profesor na Fakulteti za računalništvo in informatiko v Ljubljani. Raziskovalno se ukvarja s področjem alternativnih procesnih metod in struktur bodočnosti. Je (so)avtor dveh učbenikov ter 110 objavljenih člankov in referatov (80 v tujini).

**Miha Mraz** je izredni profesor na Fakulteti za računalništvo in informatiko v Ljubljani. Raziskovalno se ukvarja s področjem alternativnih procesnih metod in struktur bodočnosti. Je (so)avtor enega učbenika ter 90 objavljenih člankov in referatov (70 v tujini).