Avtomat za kontrolo križišča (semafor)

30. november 2009

Seminarska naloga pri predmetu Optične in nanotehnologije

Avtorji:
Aljoša Kopina, Mitja Cerkvenik, Bine Gorjanc, Vasja Laharnar, Matija Mazalin, Miha Pinterič

Ljubljana, 30. november 2009

Kazalo

1 UVOD

Naloga seminarske naloge je realizacija semaforja s pomočjo kvantnih celularnih avtomatov (QCA). Na začetku smo se odločali med realizacijo semaforja z implementacijo približka kombinatoričnega vezja, tako da bi se odzival samo na podlagi vhodov v sistem, potem pa smo se odločili za implementacijo z uporabo T pomnilnih celic.

V nadaljevanju so natančneje razložene podrobnosti obeh pristopov, rezultati ter problemi ki smo jih imeli ob načrtovanju.

1.1 KVANTNI CELIČNI AVTOMAT - QCA

QCA je definiran in temelji na nanotehnologiji kot dvo- ali več- stanjski sistem. Osnovni element strukture je celica. Za kvantni celični avtomat je značilna zelo majhna poraba energije, hiter urin takt in je brez notranjih metaliziranih povezav. Njegov največji problem pa je občutljivost na vplive okolja, zaradi majhne razlike v energiji med obema stanjema celice ter tehnološki proces izdelave.

1.2 QCA CELICA

Dinamika v celicah temelji na zakonih kvantne fizike. Celico si lahko interpretiramo kot avtomat kvadratne oblike. QCA celica je sestavljena iz štirih polprevodniških pik, katere so povezane s štirimi tuneli, skozi katere lahko elektrona prehajata med kvantnimi pikami. Ker imata oba elektrona negativni naboj, se medsebojno odbijata, kar pomeni da poskušata biti med seboj kar se da narazen (Coulombov zakon). Razlikujemo dva tipa celic- osnovno in fiksno. V osnovni elektrona neprestano krožita v največji medsebojni oddaljenosti, v fiksni celicah pa sta fiksno polarizirana v enem od dveh stanj (»-1« – logična ničla ali »1« – logična enica). Ko pa postavimo fiksno celico poleg osnovne, se osnovna celica polarizira, torej se postavi v energetsko minimalno stanje. Če se elektrona nahajata v diagonalah, pravimo, da je celica stabilna in da se nahaja v osnovnem stanju (logična ničla ali enica).

1.3 QCA STRUKTURE

Iz posameznih QCA celic lahko tvorimo kompleksnejše elemente. Z majoritetnimi vrati lahko tvorimo logično IN in ALI operacijo. Iz celic postavljenih v vrsto lahko tvorimo »žico«, če pa le te celice zavrtimo za 45°, pa lahko tvorimo alternirajočo linijo. Če dve liniji staknemo tako, da se stikata v vogalih, dobimo logično negacijo. Iz teh preprostih osnovnih struktur je mogoče zgraditi kompleksnejše strukture, vendar se z večanjem števila celic začnejo večati tudi medsebojni vplivi, kar lahko privede do nenavadnih in nepredvidljivih situacij.

1.4 URA

Namen ure je izogibanje polarizacije v napačni smeri, omogoča sekvenčno obdelavo logičnih funkcij, prav tako pa služi za ojačevanje signala. Ura deluje v obliki električnega polja nad množicami celic. Ura nima neposredne povezave s celicami, deluje pa tako, da kontrolira zapiranje in odpiranje tunelov med pikami v posameznih kvantnih celicah. To pomeni, da imamo dejansko več ur s katerimi kontroliramo delovanje v celicah. Ura v QCA ima štiri faze: fazo preklopa, fazo zadrževanja, fazo sproščanja in fazo sproščenosti. V prvi fazi se ob dvigovanju pregrad celice pričnejo polarizirati, glede na vpliv sosednjih celic. V naslednji so pregrade dvignjene, tako da je tuneliranje onemogočeno, povedano drugače, stanje celic se več ne spreminja. V zadnjih dveh fazah se pregrade spustijo in elektroni postanejo prosti.

2 AVTOMAT ZA KONTROLO KRIŽIŠČA (semafor)

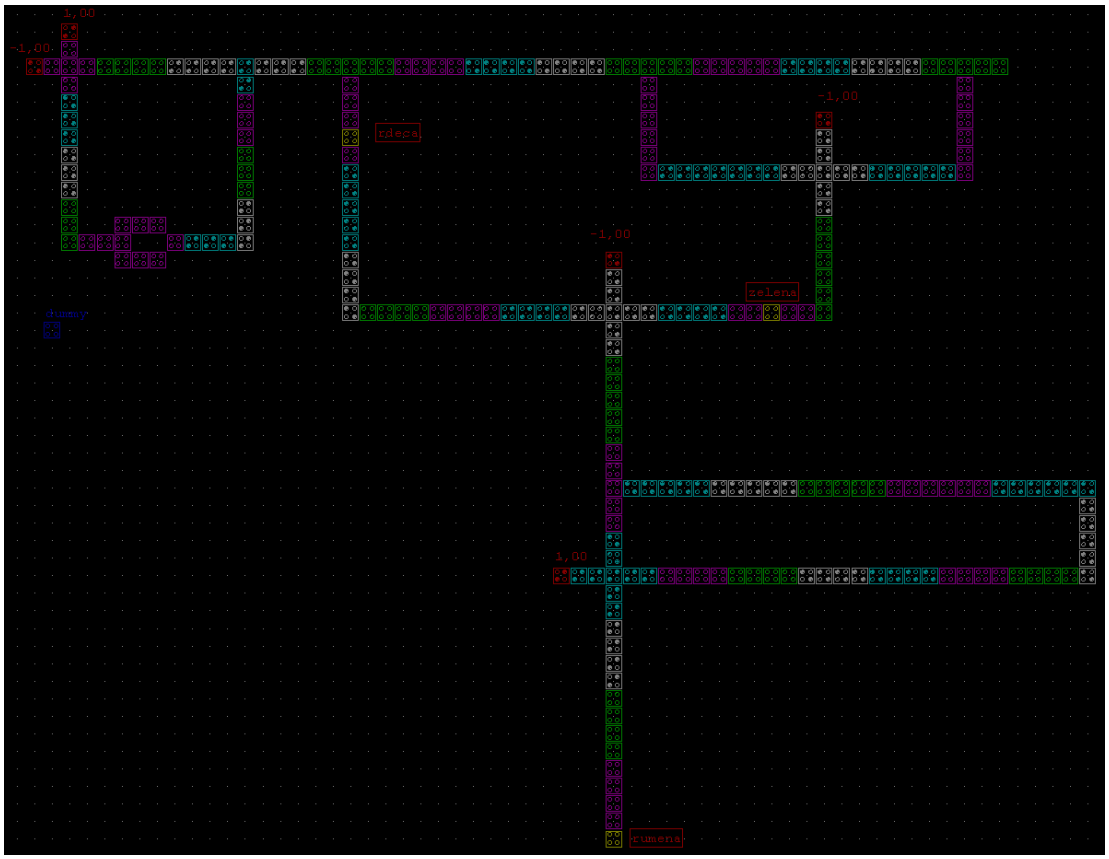
2.1 Implementacija semaforja z uporabo generatorja zaporedja in majoritetnih vrat za nadzor luči

Ideja:
Naša prva ideja je bila narediti vezje, ki bi imelo 3 izhode in bi generiralo 3 alternerajoče signale – enega s periodo 2 UP (0,1,0,1...), enega s periodo 4UP (0,0,1,1,...) in enega s periodo 8 UP (0,0,0,0,1,1,1,1,...), potem pa bi te tri signale uporabili kot vhode v naše kombinatorično vezje, katerega izhodi bi bili signali za krmiljenje luči semaforja. Kasneje smo ugotovili, da lahko rešimo problem semaforja tudi enostavneje, tako da smo pripravili vezje ki na izhod daje periodičen signal s periodo 6UP (0,0,0,1,1,1,0,0,0,1,1,1,...), ta signal pa potem uporabljamo za krmiljenje luči semaforja.

Realizacija:
Vezje smo načrtovali v dveh delih. Prvi del je generator zaporedja: 0,0,0,1,1,1, ki se ponavlja v neskončnost. To dosežemo z zanko ki je dolga 3 UP in vsebuje negator, izhod pa gre preko OR vrat, ki imajo na enem od vhodov vedno 1, kar pomeni da funkcija samo ohranja svoje stanje in ga vsake 3 periode zamenja. Prva implementacija takega generatorja je vsebovala še T-celico, ker pa lahko vezje deluje tudi brez nje, smo jo odstranili. Želeli smo, da semafor deluje po sledeči tabeli (označene so samo vrednosti kjer je vrednost spremenljivke enaka 1, v vseh ostalih primerih pa je 0)

	1	2	3	4	5	6	7	8	9	10	11	12
Rdeča	1	1	1				1	1	1			
Rumena			1			1			1			1
Zelena				1	1					1	1	

Vidimo, da je zaporedje ki ga generiramo s prvim delom vezja kar enako zaporedju ki ga želimo za rdečo luč, tako da lahko izhod poberemo kar enak. Naša naslednja ugotovitev je bila, da lahko krmilimo zeleno luč tako, da gledamo kdaj se dvakrat zapored pojavi rdeča luč (kdaj gori rdeča in je hkrati gorela tudi v prejšnji UP), ter ta signal zakasnimo za 1. Rumena luč pa se pojavlja v našem ciklu dvakrat in sicer na vsake 3 UP tako da smo vzeli kar majoritetna vrata, ki nam opravljalo funkcijo AND in sicer za eno UP zakasnjenega signala za rdečo luč ter zelene luči (kombinacija da sta oba 1 se pojavi natanko enkrat v ciklu). S tem smo dobili signal ki ima vrednost 1 enkrat vsakih 6UP, zato smo dodali še ena majoritetna vrata, v katera pripeljemo originalni signal še zakasnjen za 3UP, ter tako generiramo signal za rumeno luč.



2.2 Implementacija semaforja z uporabo T-pomnilnimi celicami za nadzor luči

Ideja:
Ker smo ob koncu implementacije prve variante ugotovili, da bi v primeru da želimo spremeniti trajanje cikla semaforja ali npr. trajanje posameznih luči, morali ponovno iskati pravilne zamike, ali izvajati minimizacijo za želeno izhodno funkcijo, smo se odločili, da poiščemo še učinkovitejšo realizacijo, ki jo bo enostavneje prilagajati ter bo bolj skalabilna.

Osnovna ideja je bila, da bi vsaka luč imela svojo T- celico, ki bi skrbela, da se stanje celici ohranja, zraven pa bi imeli vezje, ki bi skrbelo za usklajeno in pravočasno proženje signalov za spreminjanja stanja luči.

Čas enega cikla semaforja smo določili za 12UP in predvideli sledeče delo-

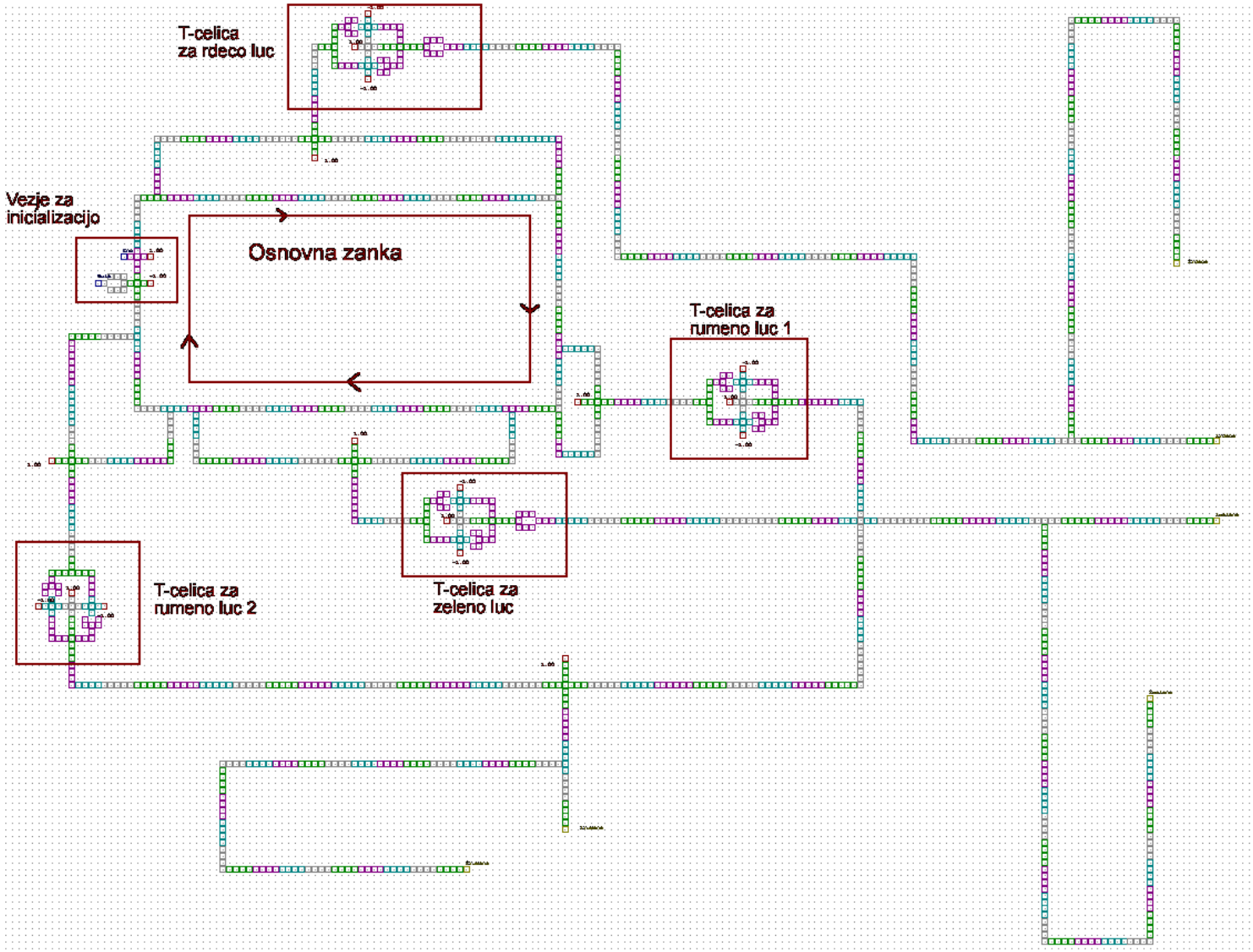
vanje:

SEMAFOR1	1	2	3	4	5	6	7	8	9	10	11	12
Rdeča	1	1	1	1	1	1	1	1				
Rumena								1				1
Zelena									1	1	1	
SEMAFOR2	1	2	3	4	5	6	7	8	9	10	11	12
Rdeča	1	1					1	1	1	1	1	1
Rumena		1				1						
Zelena			1	1	1							

Vidimo, da je takšna sekvenca precej podobna semaforjem v križiščih na naših cestah. Preden se na eni cesti prižge zelena luč se pojavi poleg rdeče še rumena luč, pred tem pa je obdobje, ko gori rdeča na obeh semaforjih. Po izteku zelene luči se prižge rumena, sledi rdeča za vsa vozila (ter da se eventuelno še izprazni križišče), nato pa sledi sekvenca rdeča in hkrati rumena ter potem še zelena na drugem kraku križišča.

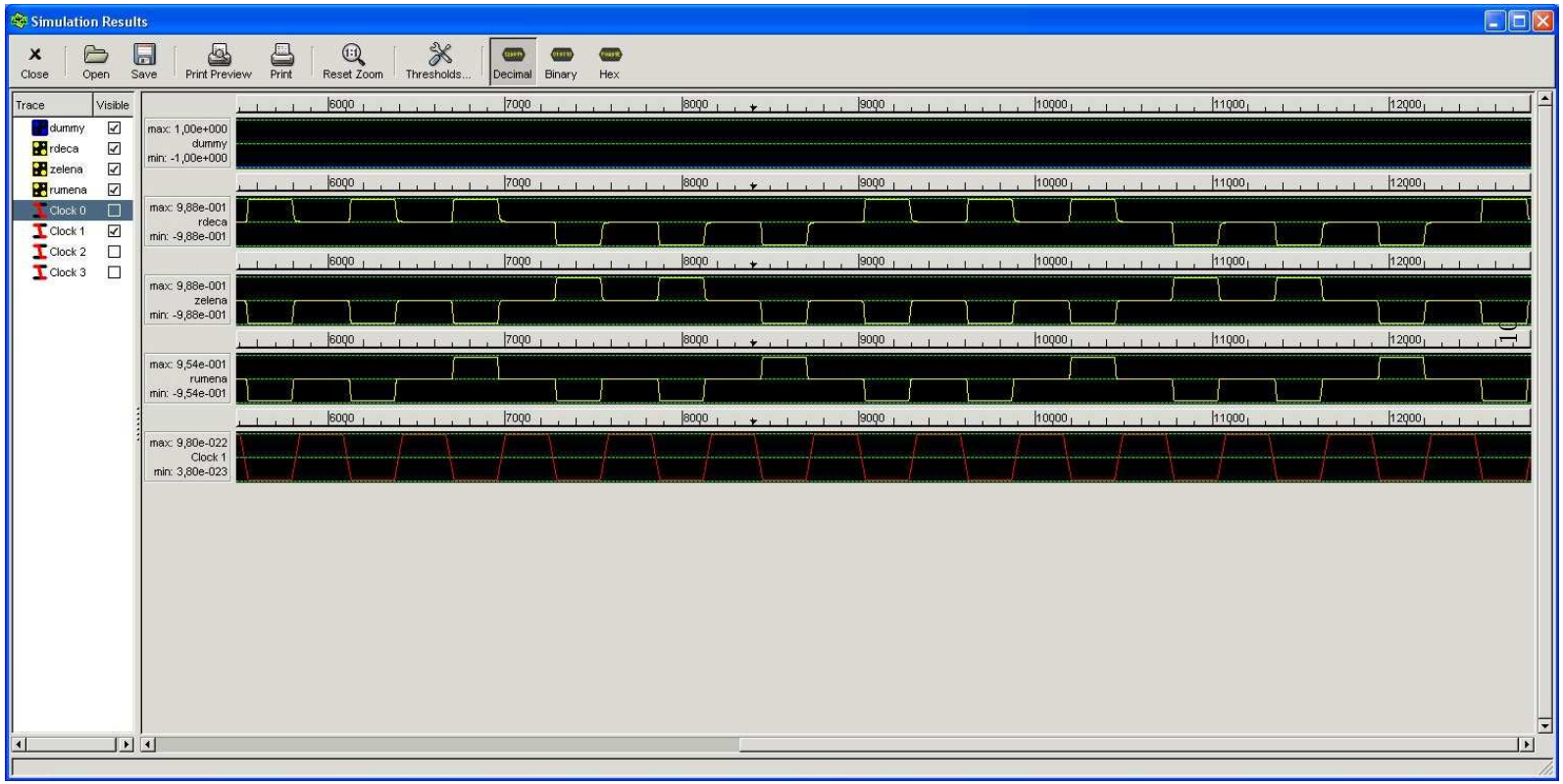
Dolžino trajanja posamezne faze se da enostavno prilagoditi, prav tako dolžino trajanja celotnega cikla semaforja.

Realizacija:
Idejo smo realizirali z uporabo zanke v kateri s periodo 12UP kroži en impulz. To smo dosegli v inicializaciji, tako da smo prvih 12 UP v zanko pošiljali 0, nato pa pošljemo en impulz visokega stanja. Po začetni inicializaciji impulz kroži v neskončnost. Ta pulz pa uporabljamo za proženje T signalov v celicah, ki hranijo stanje posamezne luči, tako da »poberemo« signal v točki A, ter v točki B v glavni zanki (z uporabo fanout-a) ter ga peljemo na majoritetna OR vrata, od tu pa na T-vhod celice. T- celica je v stanju 1 v času ki mine da naš impulz v glavni zanki pride od točke A do točke B. Če želimo spremeniti trajanje posamezne luči, se samo »povežemo« na drug del glavne zanke. V primeru da bi želeli podaljšati trajanje celotnega cikla semaforja, pa bi povečali glavno zanko. Za krmiljenje drugega semaforja uporabimo kar iste signale, le da so vsi za 6UP zakasnjeni.

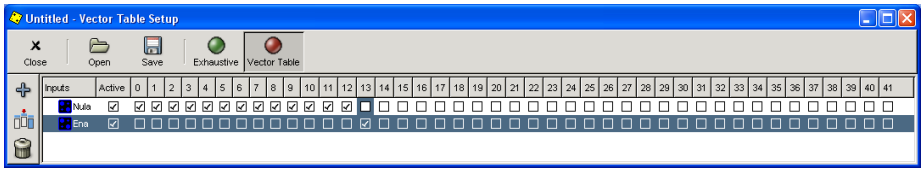

T-celica
za rdeco luc
Vezje za
inicializacijo
Osnovna zanka
T-celica za
rumeno luc 1
T-celica za
rumeno luc 2
T-celica za
zeleno luc

3 REZULTATI IN RAZPRAVA

Rezultati simulacije vezja iz točke 2.1:
Kot lahko vidimo iz rezultatov simulacije, se vezje obnaša kot smo planirali v tabeli. Inicializacija bi pri tem načinu bila potrebna le, če bi želeli, da semafor vedno začne v določenem stanju.

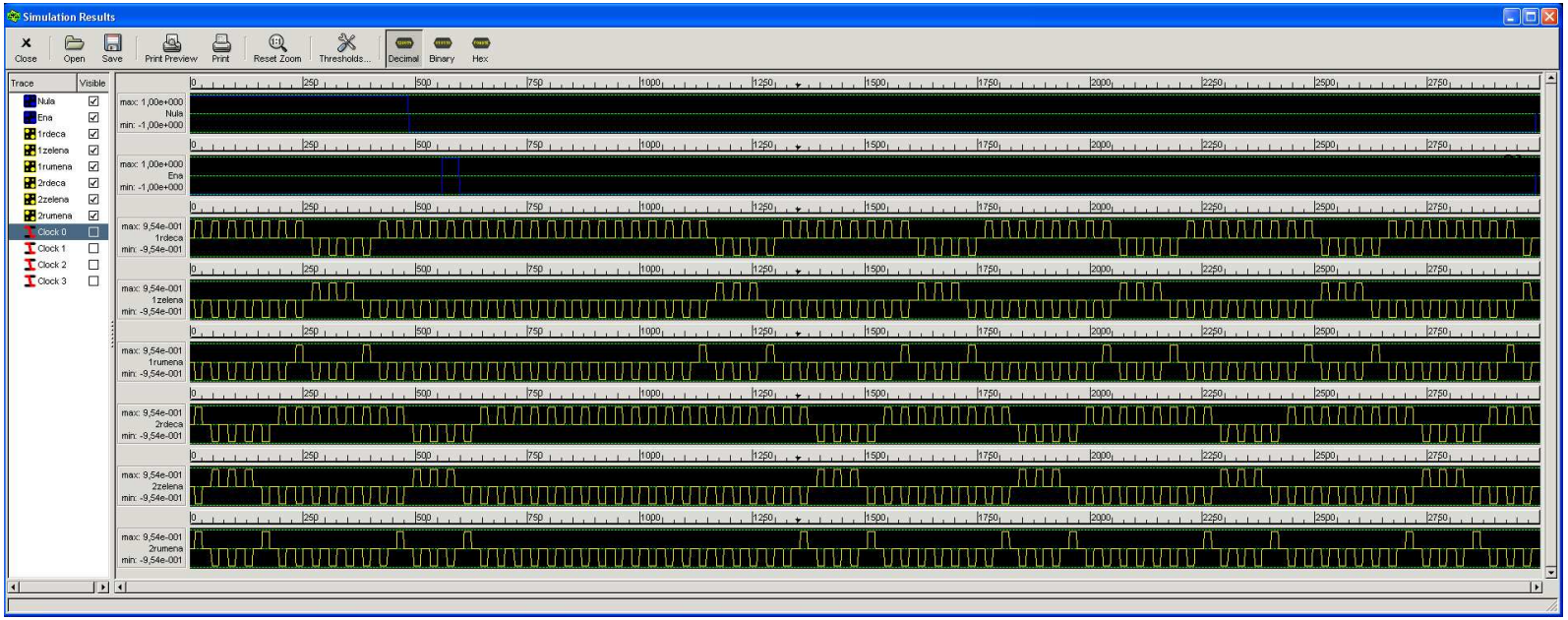

Simulation Results
Close
Open
Save
Print Preview
Print
Reset Zoom
Thresholds...
Decimal
Binary
Hex
Trace
Visible
dummy
rdeca
zelena
rumena
Clock 0
Clock 1
Clock 2
Clock 3
max: 1,00e+000
dummy
min: -1,00e+000
max: 9,88e-001
rdeca
min: -9,88e-001
max: 9,88e-001
zelena
min: -9,88e-001
max: 9,54e-001
rumena
min: -9,54e-001
max: 9,80e-022
Clock 1
min: 3,80e-023
6000
7000
8000
9000
10000
11000
12000

Rezultati simulacije vezja iz točke 2.2:
Delovanje semaforja smo simulirali z uporabo sledečega testnega vektorja:



(Slika rezultatov je na naslednji strani)

Obe implementaciji semaforja delujeta, kot smo želeli in predvideli. Rešitev z uporabo T-celic ki hranijo stanja je definitivno boljša izbira, če želimo imeti odprto možnost enostavnega prilagajanja sistema, ter tudi nekoliko bolj realna od naše prve rešitve, ker pri prvi izvedbi ob uporabi dveh semaforjev dobimo situacijo ko se takoj za rumeno lučjo na enem semaforju prižge zelena na drugem. Tak semafor v realnosti, še posebej sodeč po razmerah na naših cestah ne bi bil primeren, ker mu manjkata stanja ko sta oba semaforja rdeča. To bi lahko odpravili tako, da bi glavno zanko ki generira osnovno zaporedje podaljšali vsaj za 2UP in prilagodili obstoječa, ter dodali nova majoritetna vrata, tako da bi bili izhodi ustrezni naši novi predvideni tabeli delovanja.


Simulation Results
Close
Open
Save
Print Preview
Print
Reset Zoom
Thresholds...
Decimal
Binary
Hex
Trace
Visible
Nula
Ena
1rdeca
1zelena
1rumena
2rdeca
2zelena
2rumena
Clock 0
Clock 1
Clock 2
Clock 3
max: 1,00e+000
Nula
min: -1,00e+000
max: 1,00e+000
Ena
min: -1,00e+000
max: 9,54e-001
1rdeca
min: -9,54e-001
max: 9,54e-001
1zelena
min: -9,54e-001
max: 9,54e-001
1rumena
min: -9,54e-001
max: 9,54e-001
2rdeca
min: -9,54e-001
max: 9,54e-001
2zelena
min: -9,54e-001
max: 9,54e-001
2rumena
min: -9,54e-001

4 ZAKLJUČEK

Ob načrtovanju naših idej smo sproti dobivali kar nekaj idej za izboljšave, večino smo jih tudi uresničili, zato so končna verzija realizirana z več celicami kot bi lahko bila, ker smo jih stalno popravljali, dopolnjevali. Za izboljšavo bi se lahko najprej lotili optimizacije števila celic, ter morda tudi zakasnitev do časa ko začne semafor pravilno delovati. Težava pri optimizaciji števila celic in zakasnitev je predvsem v tem da se zelo hitro sistem začne obnašati nestabilno, zaradi prevelike občutljivosti celice na vpliv iz sosednjih celic. Druga možnost za optimizacijo bi morda bila realizacija semaforja iz točke 2.1 tako, da bi imel osnovno periodo 8UP, ter bi s tem rešili pomanjkanje situacije, ko je na obeh semaforjih rdeča luč.

Pri delu z QCADesignerjem smo naleteli na nemalo težav, sčasoma pa smo se navadili obvladovati njegove muhe ter se pri tem veliko naučili.

5 VIRI

- Tomaž Orač - Realizacija aritmetično-logičnih primitivov s strukturami kvantnih celičnih avtomatov, diplomska naloga na univerzitetnem študiju
- http://tams-www.informatik.uni-hamburg.de/applets/hades/webdemos/18-fsm/10-trafficlight/ampel_44.html
- QUANTUM-DOT CELLULAR AUTOMATA OF FLIP-FLOPS,Anoop Vetteth, Konrad Walus, Vassil S. Dimitrov, Graham A. Jullien
- Černe, J., Vesel, J., Živec, M., Janša, J., Svetek, J., 2008. QCA serijski seštevalnik I. seminarska naloga, Ljubljana.