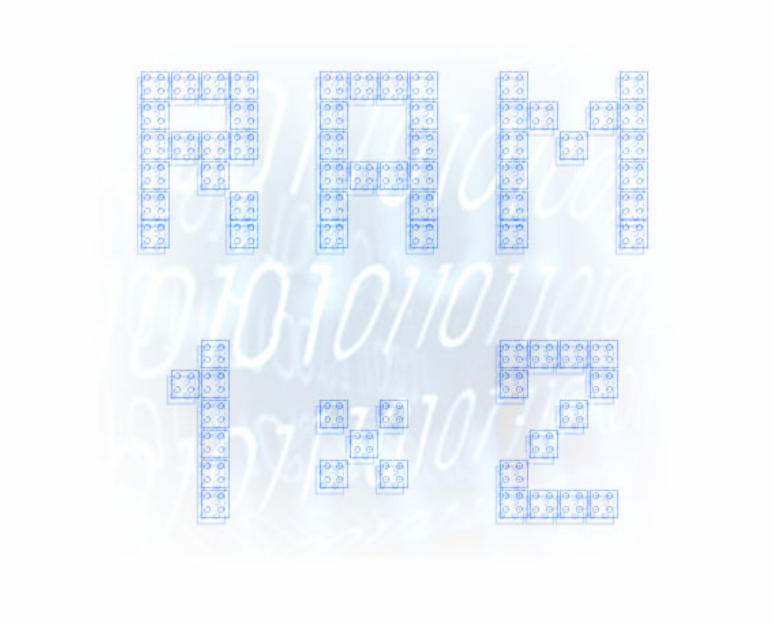
Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Seminarska naloga pri predmetu Optične in nanotehnologije

QCA 1 x 2 RAM



Avtorji:
Erlih Tomi
Ilc Nejc
Tišler Marko
Vidonja Damjan

LJUBLJANA, 17. 11. 2007

1. UVOD

1.1. QCA – kvantni celični avtomat

QCA temelji na nanotehnologiji in je definiran je kot dvo- ali več- stanjski sistem. Osnovni element QCA strukture je celica. Za kvantni celični avtomat je značilna zelo majhna poraba energije. Ima 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 četverokotne 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. Ena večjih prednosti QCA struktur je, da lahko brezizgubno križamo dve liniji. Za križanje potrebujemo eno navadno in eno za 45° stopinj zamaknjeno linijo. 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.

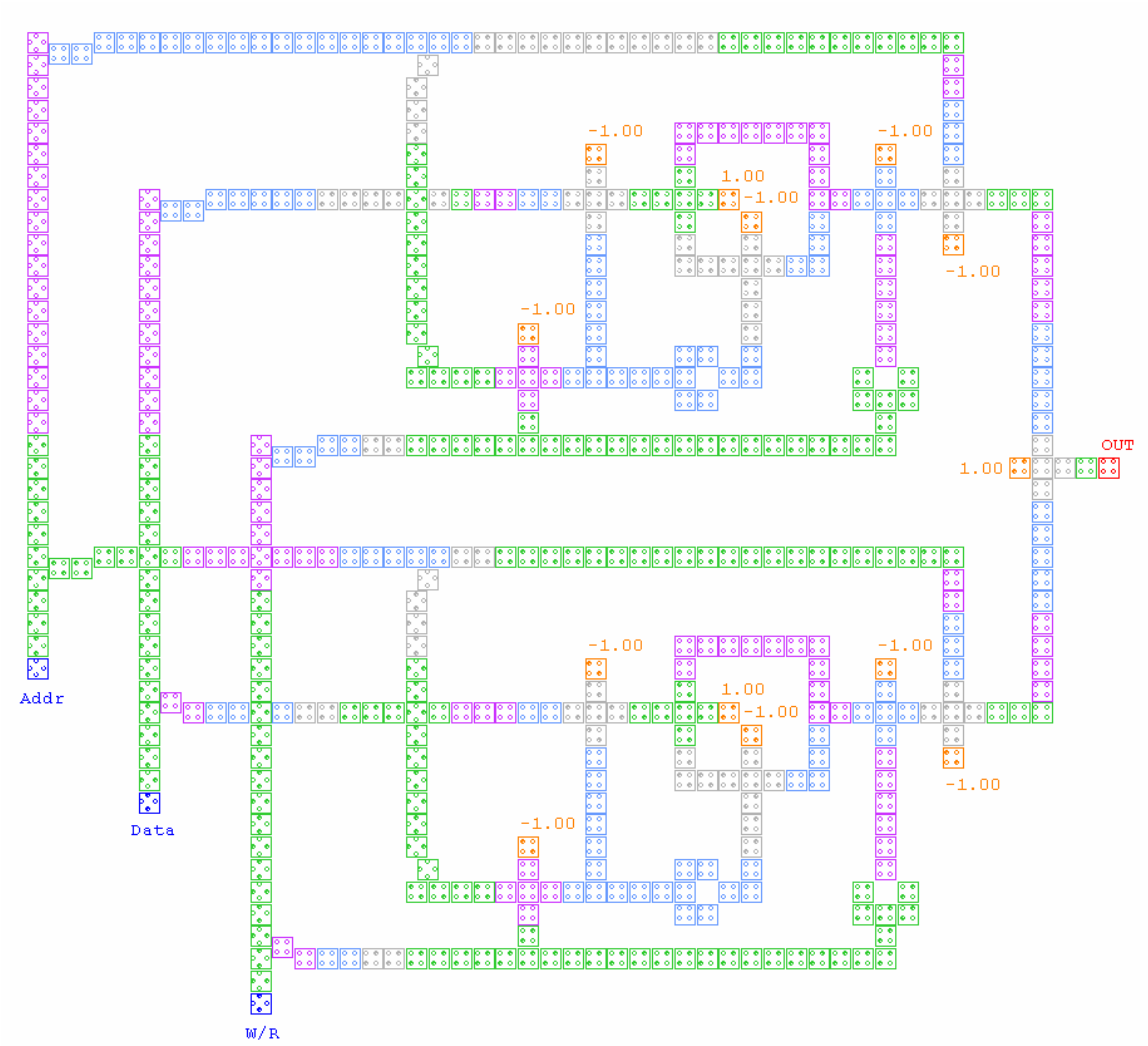
2. NALOGA

V sklopu laboratorijskih vaj pri predmetu Optične in nanotehnologije smo dobili nalogo, da z orodjem QCADesigner načrtujemo strukturo 1x2 RAM, ki mora vsebovati naslovni, podatkovni in W/R signal. Vsi signali so 1-bitni. Torej, gre za pomnilno strukturo, ki je sposobna pomnjenja dveh 1-bitnih vsebin.

3. REALIZACIJA VEZJA

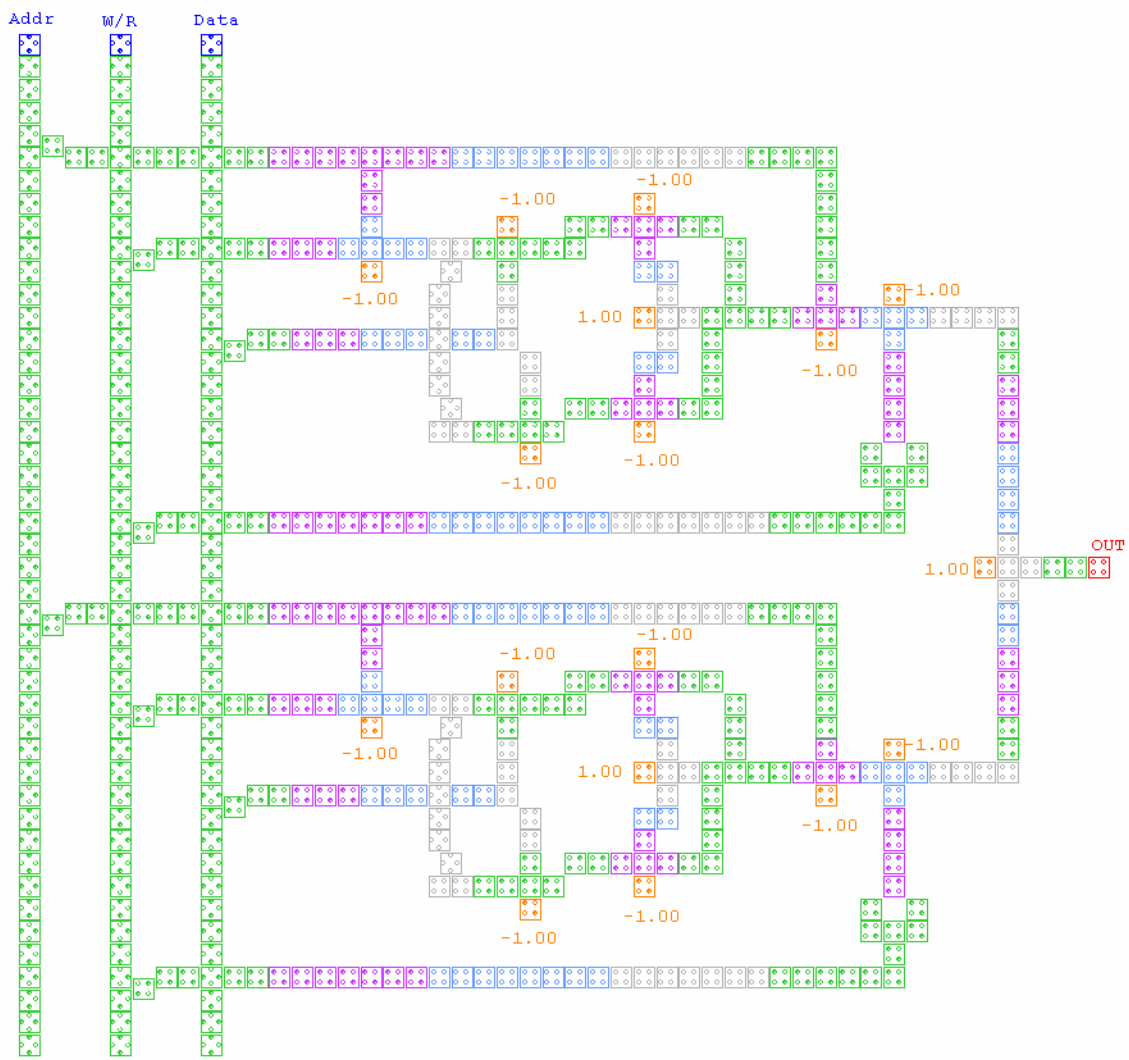
Kvantni celični avtomat s funkcijo pomnjenja dveh enobitnih vsebin (1x2 RAM), smo realizirali z orodjem QCADesigner (http://www.qcadesigner.ca), ki je prosto dostopno na spletu. V tem poglavju prikazujemo dve rešitvi: z uporabo RS in JK pomnilnih celic.

3.1. RAM z RS pomnilnima celicama



Slika 1: Realizacija vezja RAM 2x1 z RS pomnilnima celicama v QCADesigner

3.2. RAM z JK pomnilnima celicama

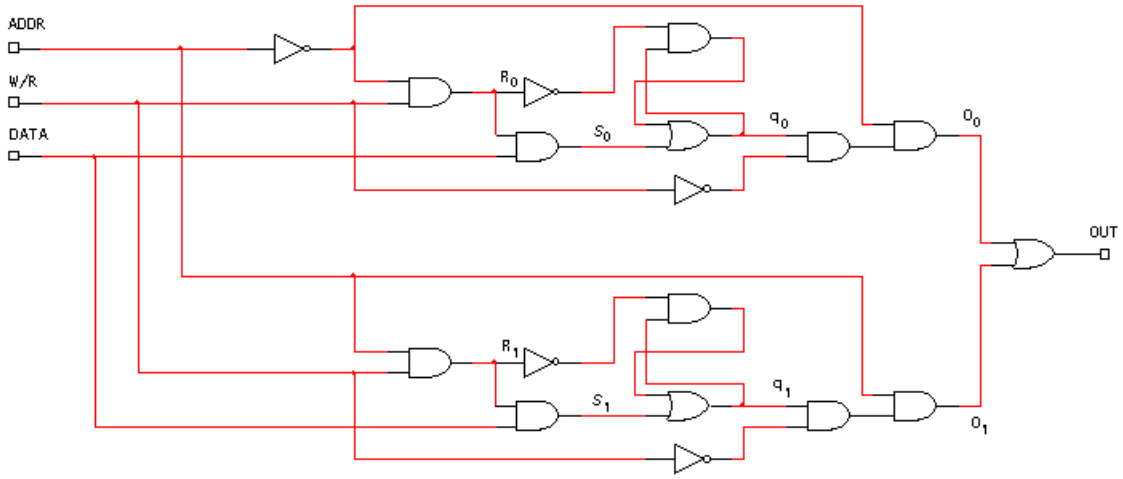


Slika 2: Realizacija vezja RAM 2x1 z JK pomnilnima celicama v QCADesigner

4. DELOVANJE VEZJA

Izdelali smo dve različici vezja: prva kot pomnilna elementa uporablja RS celici, druga pa JK celici. Sam način naslavljanja in podajanja signalov je v obeh primerih enak: na podlagi signala *ADDR* se najprej izbere ena izmed obeh celic (predstavlja »*Chip Select*«), nadaljnji potek pa je odvisen od vrednosti *W/R* signala in vrste pomnilne celice. Glede na logično shemo smo s pomočjo treh vhodnih signalov (*DATA*, *W/R* in *ADDR*) tvorili para signalov R, S in J, K pri ustreznih pomnilnih celicah. Signali *DATA*, *W/R* in *ADDR* so realizirani kot vodilo (za 45° rotirane celice) in predstavljajo vhode v vezje.

4.1. Realizacija z RS pomnilnimi celicami



Slika 3: Logična shema realizacije RAM vezja z RS celicama

Če je signal W/R aktiven (logična 1), to pomeni pisanje v celico in skupaj z *ADDR* se na *AND* vratih to omogoči. Katera vrednost se zapiše, je odvisno od signala *DATA*, ki skupaj s signalom dobljenim z *ADDR* in *W/R* tvori vrednost ki se bo zapisala v celico. V sami celici se dogaja sledeče: trenutno shranjena vrednost kroži po strukturi celice, za en prehod skozi vezje pa potrebuje 4 urine cikle (urin cikel pomeni vse 4 faze). Izvedbo pomnjenja omogočajo *OR* in *AND* vrata v sami strukturi celice. Na *AND* vrata je vezana negirana prej dobljena konjunkcija *ADDR* in *W/R* signala, ki nam v primeru, da gre za operacijo pisanja v celico, vsili vrednost logične 0. Po preklopu tretje ure se ta vrednost pojavi na vhodu *OR* vrat skupaj s konjunkcijo *DATA* in konjunkcijo *ADDR* in *W/R* signalov. Zadnja konjunkcija je potrebna, da pisanje poteka res samo ob ustreznih signalih in ne vsakič, ko je *DATA* različne vrednosti kot pomnjena vrednost v celici. Z *OR* vrati v celici in kroženjem vrednosti po strukturi, ki traja 1 urin cikel, pa je doseženo pomnjenje prejšnje vrednosti, oziroma pisanje nove vrednosti v celico, ta pa se na samem izhodu pojavi ob uri 1.

Če pogledamo logično shemo vezja, lahko ugotovimo, da smo v bistvu s konjunkcijo *DATA* in konjunkcijo *ADDR* in *W/R* signalov tvorili signal Set (S), z negacijo konjunkcije *ADDR* in *W/R* pa signal Reset (R). S tem smo se izognili tudi prepovedani kombinaciji za RS celico (Set=1 in Reset=1), saj jo zadnja negacija onemogoči. Pomembno je, da sta signala na vhodu celice istočasna (ob enaki uri).

Izhod iz pomnilne celice je nato vezan še na konjunkcijo z negiranim *W/R* signalom, kar pomeni operacijo branja, ta pa je nato vezan še na *AND* vrata skupaj z *ADDR* signalom, kar v bistvu tvori enostaven *Output enable* vhod (torej celica je izbrana in poteka branje vsebine). Da imamo samo en izhodni signal iz vezja, sta izhoda iz obeh vezana še na *OR* vrata, s tem pa je dobljen signal *OUT*.

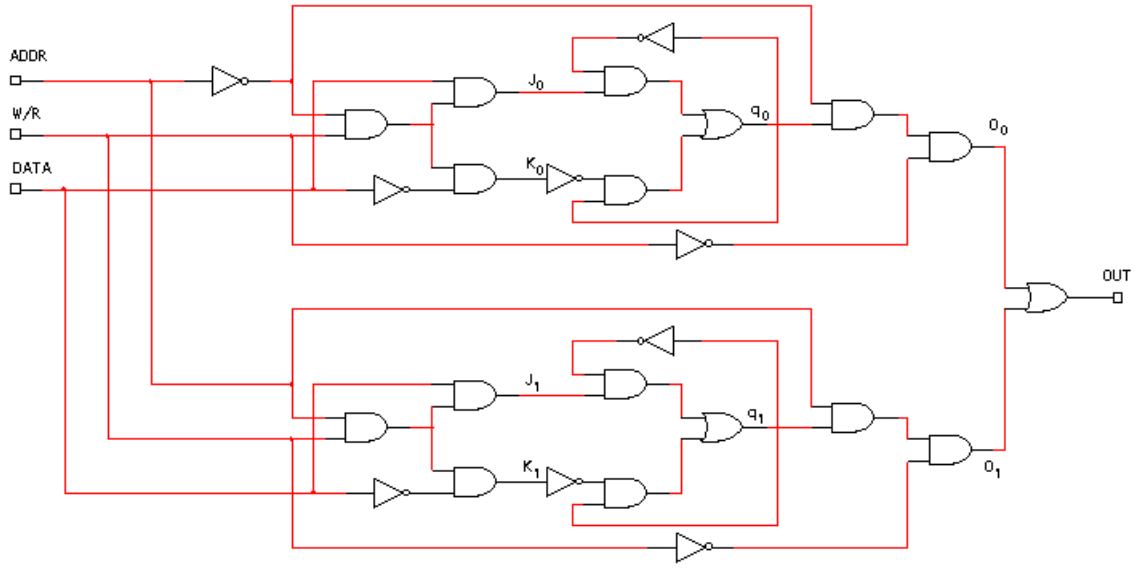
Tabela 1: Pravilnostna tabela RS pomnilne celice

R	***S***	Q_{t+1}	***Operacija***
0	0	Q	Drži stanje
0	1	0	Reset
1	0	1	Set
1	1	X	X

Enačbe 1: RAM z RS pomnilnimi celicami

$$D^1q = s \vee \bar{r}q$$
$$s_0 = Data \cdot \overline{Addr} \cdot W / R$$
$$r_0 = \overline{Addr} \cdot W / R$$
$$s_1 = Data \cdot Addr \cdot W / R$$
$$r_1 = Addr \cdot W / R$$
$$O_0 = \overline{W / R} \cdot \overline{Addr} \cdot q_0$$
$$O_1 = \overline{W / R} \cdot Addr \cdot q_1$$
$$Out = O_0 \vee O_1$$

4.2. Realizacija z JK pomnilnimi celicami



Slika 4: Logična shema realizacije RAM vezja z JK celicama

Ta realizacija zahteva nekaj več logičnih elementov, kar je vidno iz logične sheme vezja, posledično pa tudi več celic v sami rešitvi, kar bi lahko hitreje pripeljalo do nestabilnosti. Signala J in K v tem primeru tvorimo podobno kot *R* in *S* v prejšnjem. Najprej se izvede konjunkcija med *ADDR* in *W/R*, dobljeni signal pa je vezan na *AND* vrata, enkrat skupaj z *DATA* signalom, drugič pa z negiranim DATA signalom. Na ta način smo dobili v prvem primeru signal *J* in v drugem signal *K*. Ker imamo pri drugi konjunkciji enkrat *DATA* in drugič negiran *DATA* signal, kombinacija *J=1* in *K=1* tudi v tem primeru ni mogoča, kar pomeni, da smo zgolj povečali kompleksnost samega vezja, pridobili pa nismo nobene dodatne funkcionalnosti. Pomnjenje se v tem primeru izvaja s pomočjo dveh *AND* vrat in *OR* vrat in kroženja vrednosti po strukturi. *AND* vrata, katerih izhod vodi v *OR* vrata, služijo za signal *K* (postavimo vsebino celice na 0), vrata ki pa so nad *OR* vrati pa služijo za *J* signal (postavimo vsebino celice na 1). Vsebina potuje krožno po celici tudi v tem primeru 1 cikel ure, na izhodu iz celice pa se pojavi ob uri 0. Prav tako je pomembno, da sta veljavna vhoda *J* in *K* v celico aktivna ob isti uri! Izhodna logika je podobna kot v prejšnjem primeru, le da sta zamenjana signala *W/R* in *ADDR* na obeh *AND* vratih.

Kar se tiče poteka ur 0, 1, 2 in 3, sta operaciji pisanja in branja v obeh vezjih enako dolga: 4 urine cikle. Če pa se osredotočimo zgolj na pomnilni celici, pri realizaciji z RS celico zaradi enostavnejšega vezja dobimo na izhodu RS celice podatek 2 preklopa ure prej kot pri drugi. Zaradi enostavne realizacije križanj

med signali smo v primerih ko je bilo križanje potrebno uporabili križanje s pomočjo rotacije celic za 45°.

Tabela 2: Pravilnostna tabela JK pomnilne celice

J	***K***	Q_{t+1}	***Operacija***
0	0	Q	Drži stanje
0	1	0	Kill(Reset)
1	0	1	Jump(Set)
1	1	$\overline{Q}$	Negiraj vsebino

Enačbe 2: RAM z JK pomnilnimi celicami

$$D^1q = j\overline{q} \vee \overline{k}q$$
$$j_0 = Data \cdot \overline{Addr} \cdot W / R$$
$$k_0 = \overline{Data} \cdot \overline{Addr} \cdot W / R$$
$$j_1 = Data \cdot Addr \cdot W / R$$
$$k_1 = \overline{Data} \cdot Addr \cdot W / R$$
$$O_0 = \overline{W / R} \cdot \overline{Addr} \cdot q_0$$
$$O_1 = \overline{W / R} \cdot Addr \cdot q_1$$
$$Out = O_0 \vee O_1$$

Tabela 3: Interpretacija vhodnih signalov kot operacije

Addr	***W/R***	***Data***	***Operacija***
0	0	X	Beri iz celice A0
0	1	X	Piši Data c celico A0
1	0	X	Beri iz celice A1
1	1	X	Piši v celico A1

5. REZULTATI

Delovanje izdelanega RAM vezja je bilo testirano z različnimi vhodnimi vektorji ("vector table" v QCADesignerju). Ker so lastnosti RAM vezja, izdelanega s pomočjo dveh tipov spominskih celic – JK in RS, skoraj do potankost enake, velja spodaj napisano tako za prvi, kot za drugi tip RAM-a. Tako lahko učinkovito primerjavo delovanje obeh tipov pomnilnih celic. Primerjavo olajšuje dejstvo, da imata oba tipa celic enako zakasnitev, tj. 3 urine periode.
Vsak korak testiranja posamezne operacije predstavlja dogodek, ki se zgodi v eni urini periodi. Kateri korak se kje zgodi, je tudi označeno na sliki grafa delovanja pomnilne celice. Pri tem je zmeraj potrebno upoštevati dejstvo, da je opazovana vrednost na izhodu zakasnjena za tri urine periode. Ravno iz tega razloga so na vseh grafih vrednosti izhoda za prve tri periode delovanja zamegljene (slabše vidne).

Slika delovanja pomnilne celice je sestavljena iz naslednjih grafov:

- Address – graf prikazuje vrednost na liniji, s katero izbiramo lokacijo v RAM-u (ADDR)
- Write/Read – prikazuje vrednost na liniji s katero določamo vrsto operacije: pisanje (logična vrednost 1) ali branje (logična vrednost 0)
- Data – podatek, ki ga želimo vpisat v pomnilno celico
- Output – prikazuje potek vrednosti, ki jih dobimo na izhodu RAM pomnilne celice
- Clock – prikazuje stanje ure; ob vsakem nizkem nivoju ure se postavijo nove vrednosti celic avtomata

Različne barve grafov pomenijo različne tipe vrednosti:
- Modra prikazuje vhodne vrednosti
- Zelena prikazuje grafe za izhodne vrednosti
- Rdeča predstavlja potek urinega signala

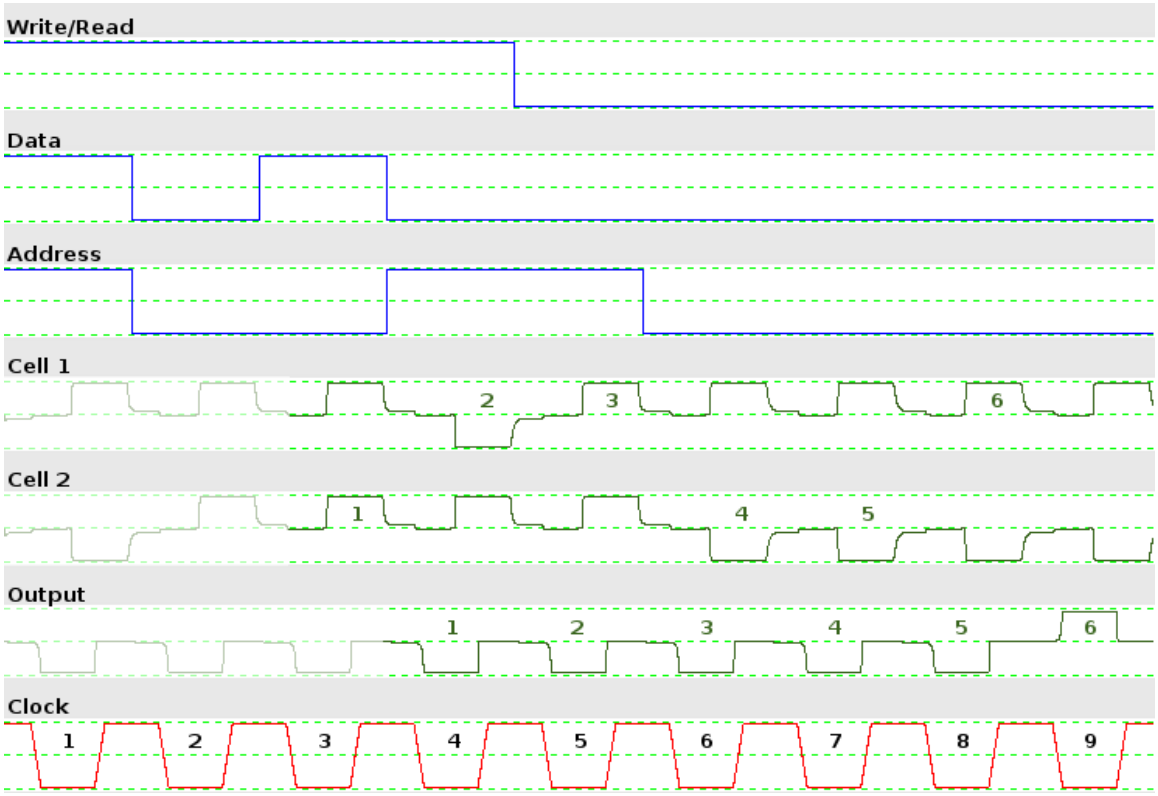
5.1. Pisanje

Pri pisanju v spominsko celico se mora podatek iz podatkovne linije prenesti v spominsko celico in se ohraniti v njej, dokler ne vpišemo nove vrednosti. Ob vpisu v pomnilno celico dobimo na izhodu potrditev – logično vrednost nič.

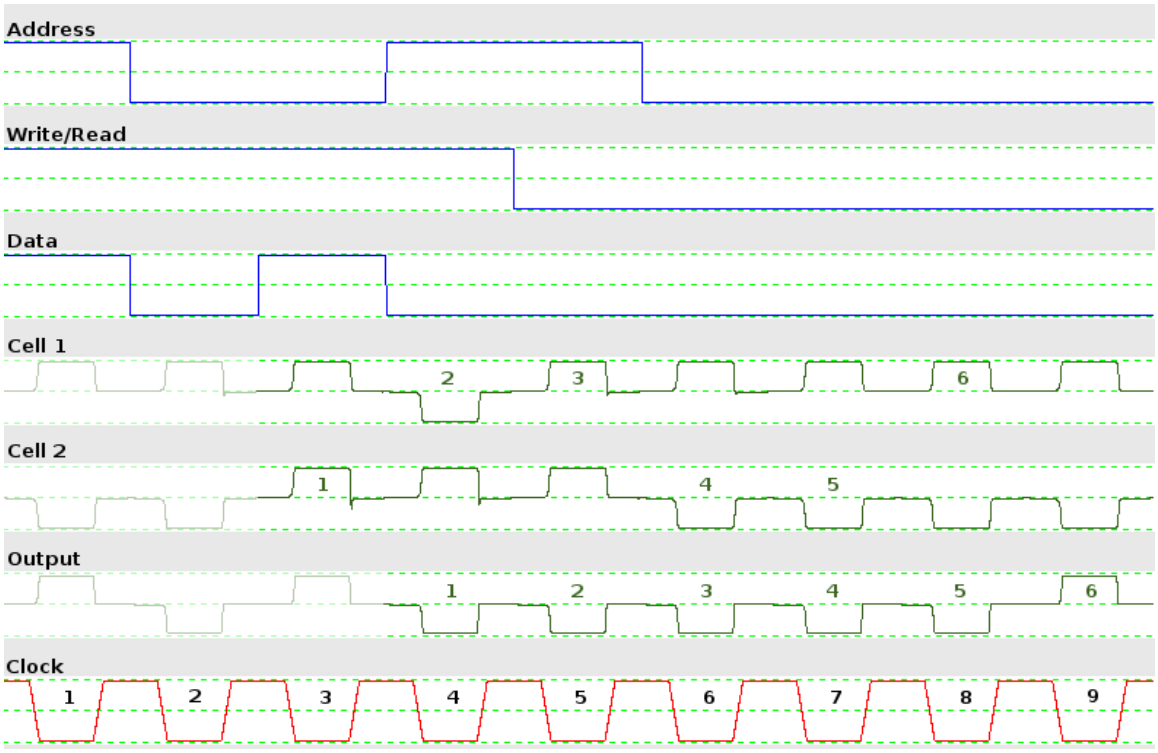
Zgoraj opisani sliki delovanja celice smo dodali še dva dodatna grafa („Cell 1" in „Cell 2"), ki prikazujeta trenutno vrednost, shranjeno v pomnilni celici. Pri tem je potrebno poudariti, da so tu vrednosti zakasnjene samo za dve urini periodi, saj se sam vpis zgodi že v drugi urini periodi.

Testirali smo sledeči primer:

1. Vpis logične vrednosti 1 v drugo lokacijo
2. Vpis logične vrednosti 0 v prvo lokacijo
3. Vpis logične vrednosti 1 v prvo lokacijo
4. Vpis logične vrednosti 0 v drugo lokacijo
5. Branje vrednosti druge lokacije
6. Branje vrednosti prve lokacije



Slika 5: Pisanje v RS pomnilni celici



Slika 6: Pisanje v JK pomnilni celici

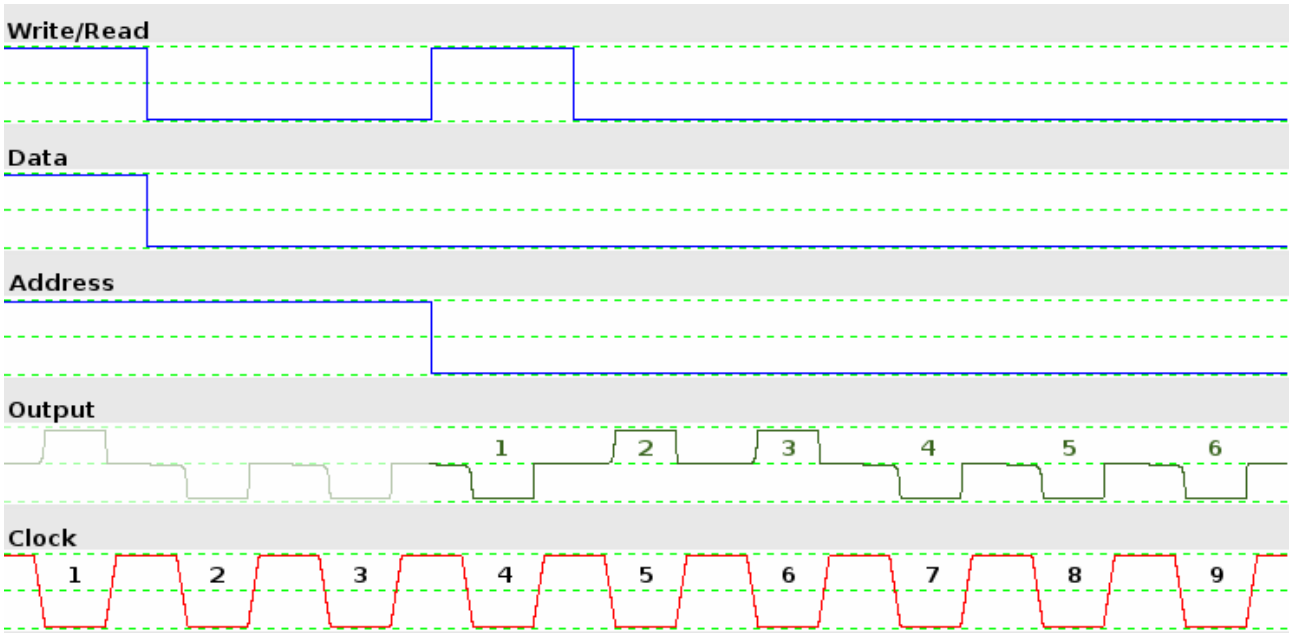
Iz grafa izhodnih vrednosti je moč videti, da dobimo na izhodu pričakovane vrednosti: za prve štiri periode, ko podatke le vpisujemo v pomnilne celice, dobimo na izhodu ves čas logično ničlo, za zadnji dve periodi pa trenutno hranjeni vrednosti obeh lokacij (logično ničlo za lokacijo 2 ter logično enico za lokacijo 1). Na grafih „Cell 1" in „Cell 2" lahko ob tem spremljamo spreminjanje vrednosti shranjenih na obeh lokacijah.

5.2. Branje

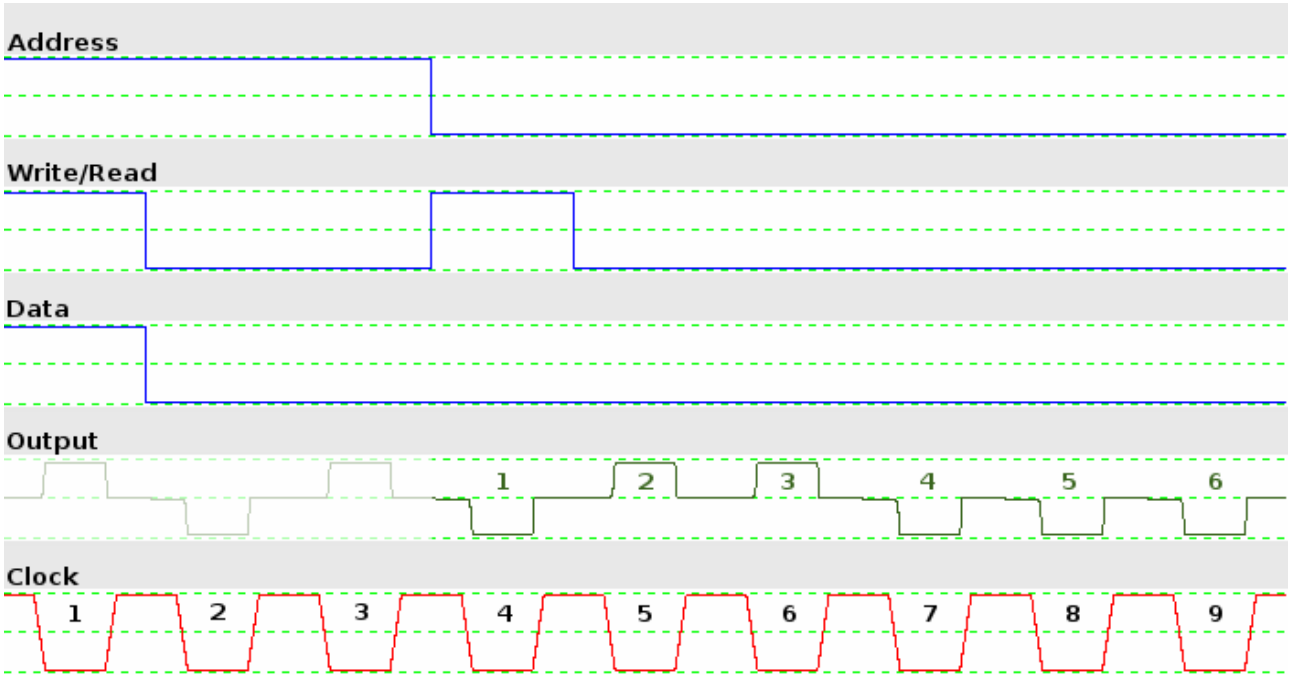
Branje prenese shranjene vrednosti v pomnilni celici iz želene lokacije na izhod, ne da bi shranjeno vrednost ob tem „poškodoval" - branje ni destruktivno.

Testirali smo naslednji primer:

1. Zapis logične vrednosti 1 na drugo lokacijo
2. Branje vrednosti, shranjene na drugi lokaciji
3. Branje vrednosti, shranjene na drugi lokaciji
4. Zapis logične vrednosti 0 na prvo lokacijo
5. Branje vrednosti, shranjene na prvi lokaciji
6. Branje vrednosti, shranjene na prvi lokaciji



Slika 7: Branje iz RS celic



Slika 8: Branje iz JK celic

Iz grafa izhodnih vrednosti vidimo, da se v prvi periodi (po zakasnitvi treh period) na izhodu najprej pojavi logična ničla, ki jo dobimo zaradi operacije pisanja, nato pa sledita dve logični enici, ki ju beremo iz druge lokacije. Spet sledi operacija pisanja, ter še dve logični ničli na izhodu, kar je skladno z dvojnih branjem vrednosti (logične ničle) na prvi lokaciji.

6. ZAKLJUČEK

Delovanje obeh tipov celic v testnih primerih je potrdilo, da celice logično pravilno delujejo. Iz oblik izhodnih vrednosti obeh tipov celic vidimo, da so signali zelo stabilni in močni. Če primerjamo oba tipa celic, so razlike na izhodu minimalne, skoraj neopazne. Določene manjše razlike med njima pa izvirajo bolj iz samega delovanja simulatorja, ki za vsako simulacijo ne "resetira" začetnih pogojev in vrednosti v celicah.

Pri testiranju RAM pomnilne celice, ki je bila realizirana s pomočjo JK pomnilne celice je prišlo do anomalij, ki so pod istimi pogoji, nad isto strukturo dale zelo različne rezultate. Le-te imajo lahko enega od dveh vzrokov:

- Simuliranje avtomata z več kot 500 celicami in na tako velikem vhodnem vektorju (več kot 25 vhodnih vrednosti) zmede simulator
- Branje iz celice le ni tako nedestruktivno, kot je bilo predvideno in tako privede do napačnega delovanja

7. VIRI

- http://www.nsti.org/publ/Nanotech2003v2/M7003.pdf
- prof. Miha Mraz: predloge za predavanja pri predmetu ONT
- mag. Primož Pečar: predloge za vaje pri predmetu ONT