



Optične in nanotehnologije  
4.I. UNI-RI  
1. seminarska naloga

# QCA dvo-bitni množilnik

Dejan Štimpfelj, 63040162  
Matjaž Okretič, 63040118  
Miha Gorše, 63020046  
Anže Leban, 63030312

# Kazalo

1. Uvod.....	3
2. Splošno o implementaciji .....	4
3. Implementacija.....	6
4. Posebnosti pri realizaciji.....	10
5. Rezultati simulacije.....	11
6. Zaključek.....	13
7. Viri.....	14

# 1. Uvod

Pri predmetu Optične in nanotehnologije smo se poglobljeje spoznali s kvantnimi celičnimi avtomati. Strukture smo simulirali z orodjem QCA Designer. To orodje nam omogoča snovanje vezij s kvantnimi celičnimi avtomati brez potrebe po poglobljenemu znanju kvantne fizike.

Naša naloga je bila sestaviti dvo-bitni množilnik.

Skozi seminarsko nalogo bomo spoznali tudi nove pristope grajenja vezij, saj kvantni celični avtomati omogočajo brezizgubno križanje linij, kar je v primerjavi s sedanjo tehnologijo izdelovanja vezij velika prednost. Po drugi strani pa na same avtomate lahko zaradi samega principa delovanja vplivajo že malenkosti, kot na primer sosednja celica, če je le-ta preblizu.

## 2. Splošno o implementaciji

Na samem začetku smo morali definirati način, kako zmnožiti dve dvo-bitni števili, da bi bila implementacija kar se da enostavna in hitra. Tukaj se je za najbolj enostaven način množenja izkazalo naslednje pravilo.

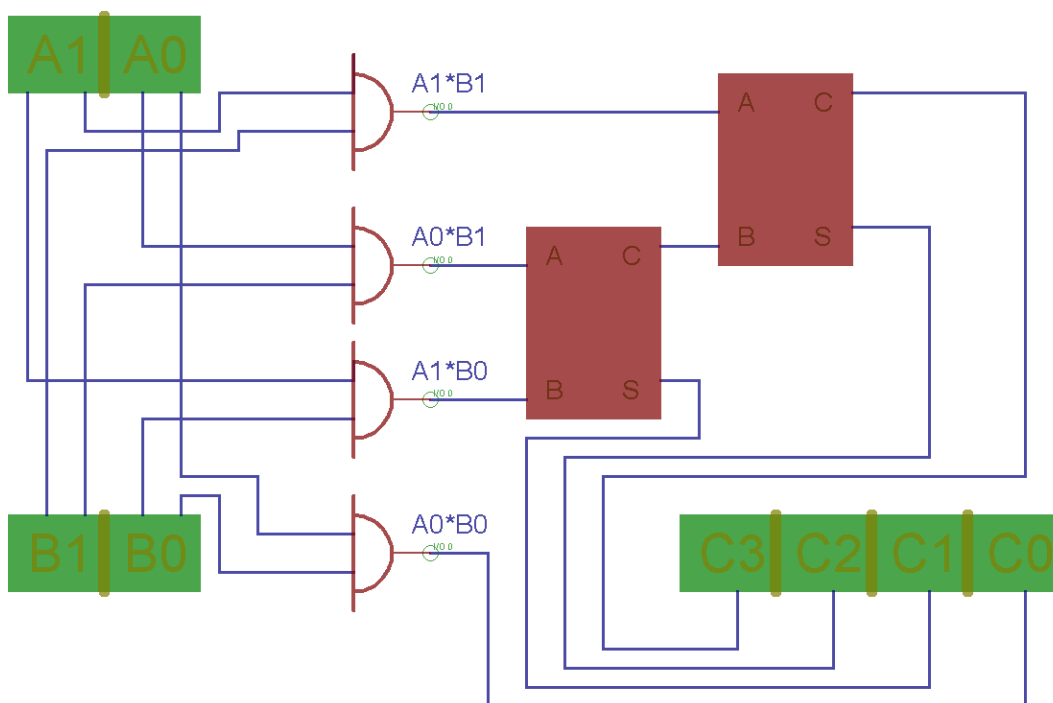
Če upoštevamo, da je prvo dvo-bitno število oblike  $A_1A_0$ , ter da je drugo dvo-bitno število oblike  $B_1B_0$ , potem je njun zmnožek enak:

$$A_1A_0 \times B_1B_0 = A_1 \times B_1 \times 2^2 + A_1 \times B_0 \times 2^1 + A_0 \times B_1 \times 2^1 + A_0 \times B_0 \times 2^0$$

Za samo implementacijo lahko zamenjamo vsa množenja z logičnimi AND-i in tako pridemo do spodnje enačbe:

$$A_1A_0 \times B_1B_0 = A_1 * B_1 \times 2^2 + A_1 * B_0 \times 2^1 + A_0 * B_1 \times 2^1 + A_0 * B_0 \times 2^0$$

Iz enačbe se hitro vidi, da za realizacijo potrebujemo štiri AND vrata ter dva seštevalnika. Logična slika našega dvo-bitnega množilnika je prikazana na spodnji sliki.



Slika 1: Logična shema dvo-bitnega množilnika

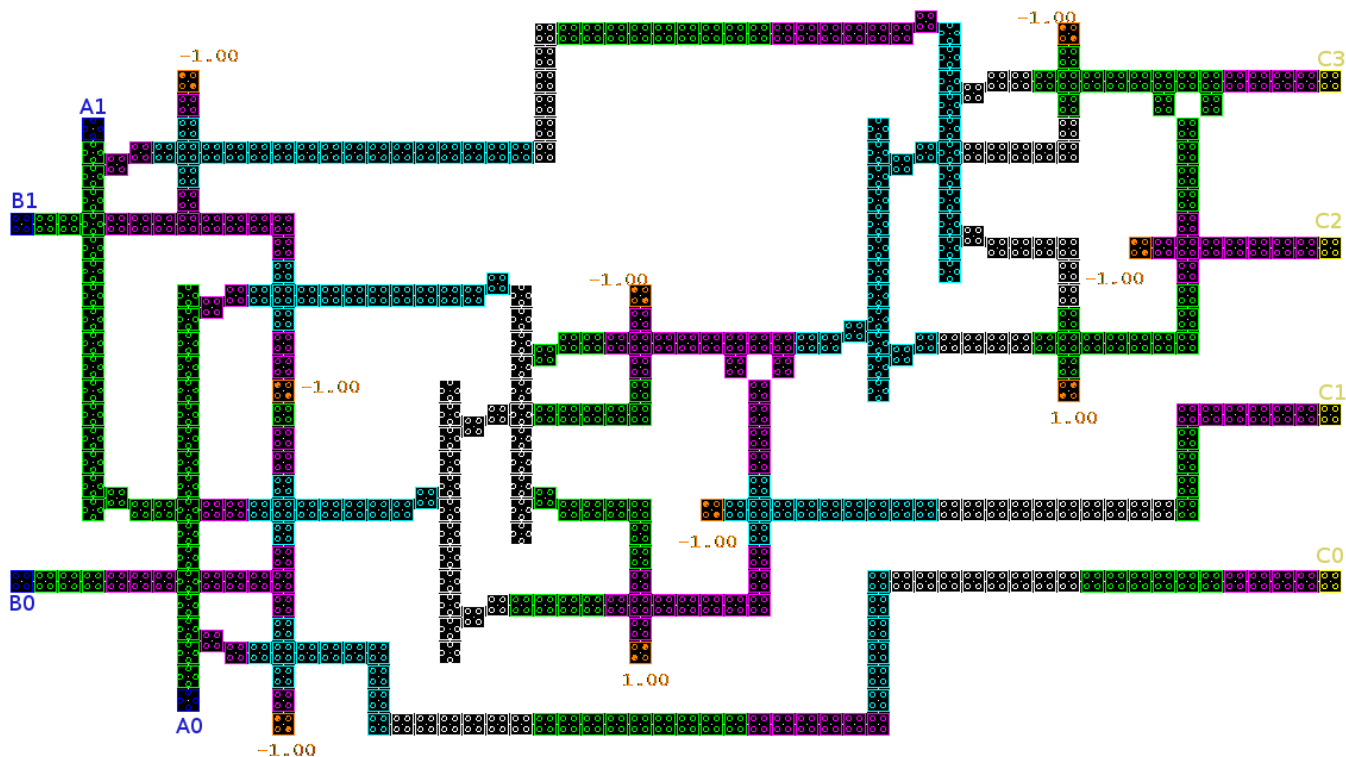
Logični shemi ustreza spodnja pravilnostna tabela.

$A_1$	$A_0$	$B_1$	$B_0$	$C_3$	$C_2$	$C_1$	$C_0$
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	1
0	1	1	0	0	0	1	0
0	1	1	1	0	0	1	1
1	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0
1	0	1	0	0	1	0	0
1	0	1	1	0	1	1	0
1	1	0	0	0	0	0	0
1	1	0	1	0	0	1	1
1	1	1	0	0	1	1	0
1	1	1	1	1	0	0	1

*Tabela 1: Pravilnostna tabela dvo-bitnega množilnika*

### 3. Implementacija

Pri sami implementaciji smo imeli veliko težav s križanjem vodil, zato smo morali zelo previdno izbrati pozicijo vhodov in vodil. To je seveda zelo vplivalo na samo sliko vezja, ki je prikazana spodaj.



Slika 2: Slika vezja dvo-bitnega množilnika v orodju QCA Designer

Samo vezje implementirano v orodju QCA Designer je popolnoma identično logični shemi. Vhodi  $A_1$ ,  $A_0$ ,  $B_1$  in  $B_0$  se preko štirih AND vrat, ki so realizirana z majoritetnimi vrati, peljejo naprej na dva seštevalnika, od koder dobimo izhod vezja, označen s  $C_3 - C_0$ . Kombinacija vhodov  $A_0 * B_0$  vodi direktno na izhod  $C_0$ , vsi ostali izhodi operacije AND pa se pripeljejo na ustrezne vhode obeh seštevalnikov.

AND vrata smo realizirali z majoritetnimi vrati tako, da smo enemu od vhodov priredili vrednost -1, kar pri kvantnih celičnih avtomatih predstavlja logično vrednost 0. To je standardna realizacija AND vrat. Za realizacijo OR vrat v seštevalnikih pa smo enemu od vhodov majoritetnih vrat priredili vrednost 1, kar pri celičnih kvantnih avtomatih predstavlja logično vrednost 1.

Pri realizaciji vezja smo uporabili polovični seštevalnik, ker polnega nismo potrebovali. Polni seštevalnik ima poleg vhodov, ki jih ima polovični seštevalnik, še vhod za prenos s prejšnjega mesta ( $C_i$ ) in tako sešteje tri eno-bitna števila. Mi tega nismo potrebovali, zato smo vezje poenostavili z uporabo dveh polovičnih seštevalnikov.

Polovični seštevalnik je komponenta, ki sešteje dva vhodna bita. Ima dva vhoda ( $X$  ter  $Y$ ) ter dva izhoda ( $S$  (sum – vsota) in  $C$  (carry – prenos)). Rezultat seštevanja dobimo na izhodu. Izhod  $S$  je v aktivnem stanju v primeru, ko je vsota števil 1, izhod  $C$  pa je v aktivnem stanju v primeru, ko sta aktivna oba vhoda.

Logični enačbi za polovični seštevalnik sta:

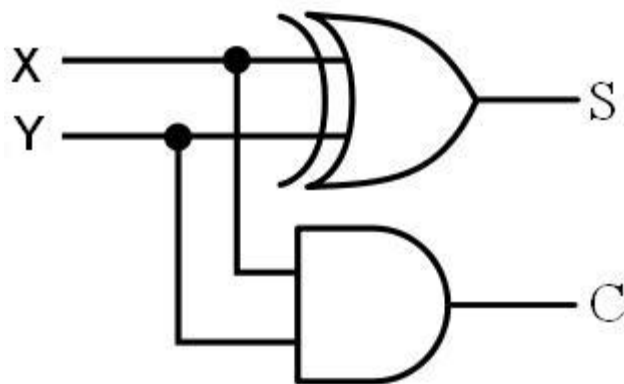
- $C = X \text{ AND } Y$
- $S = X \text{ XOR } Y = [(X \text{ OR } Y) \text{ AND } \overline{(X \text{ AND } Y)}]$

Tema enačbama ustreza spodnja pravilnostna tabela:

X	Y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

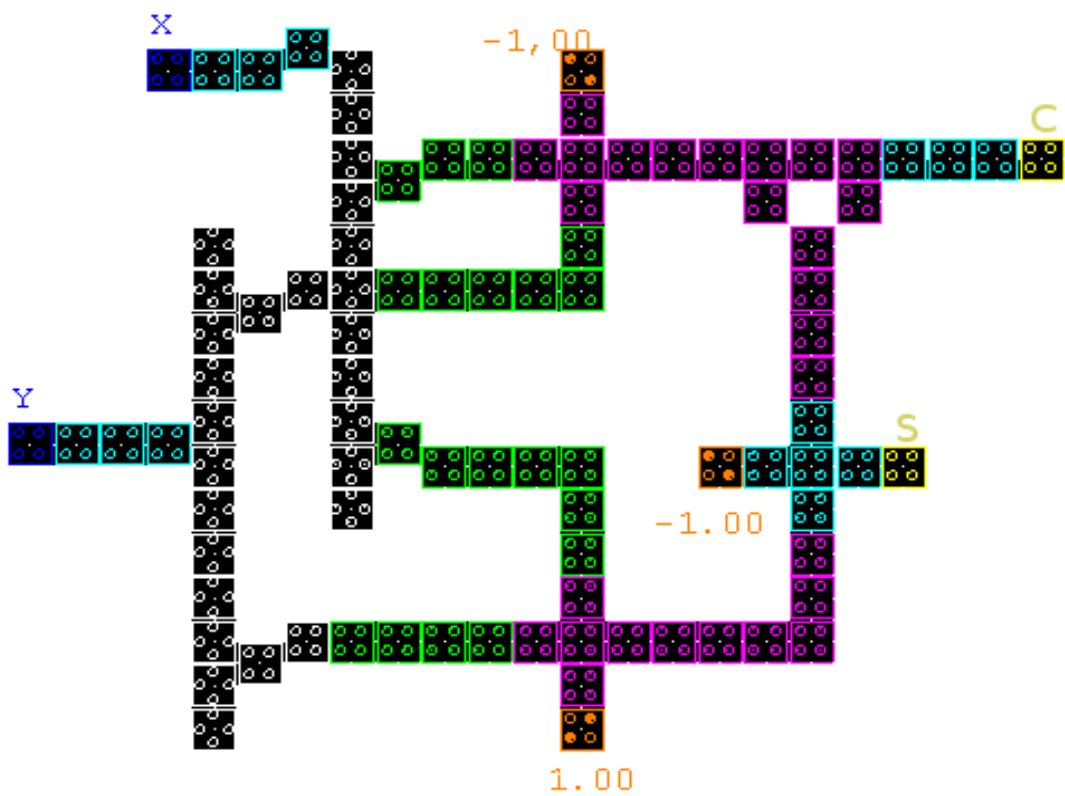
Tabela 2: Pravilnostna tabela seštevalnika

Zgornji pravilnostni tabeli pa ustreza spodnja logična shema:



Slika 3: Logična shema seštevalnika

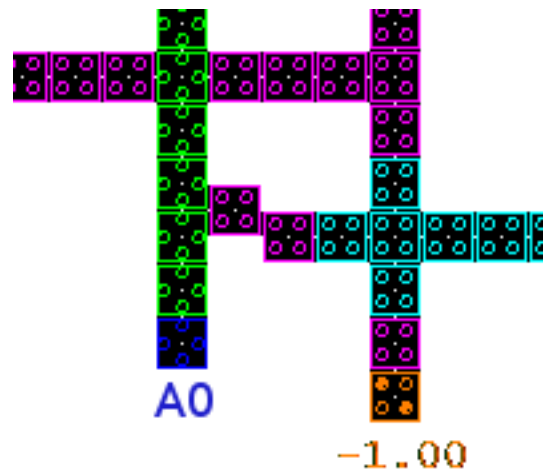
Za razliko od logične sheme seštevalnika, smo pri realizaciji v QCA Designer-ju uporabili dvoje AND in ena OR vrata ter negacijo. Vezje je prikazano na spodnji sliki:



Slika 4: Vezje seštevalnika v orodju QCA Designer



Pri realizaciji celotnega vezja smo morali uporabiti vse štiri ure, ki so na voljo. Razlogov za to je več. Z uporabo večih ur se stanje celic na linijah stabilizira. S tem se poveča zanesljivost in doseže sama pravilnost delovanja vezja. Po drugi strani pa je pri uporabi več kot ene ure zelo pomembno, da so vsi podatki na voljo v istem urinem stanju. Primer za to so AND vrata na spodnji sliki.



Slika 5: AND vrata v orodju QCA Designer

Na sliki 5 lahko vidimo dva primera uporabe različnih ur:

1. Prvi primer je uporaba različnih ur pri križanju. Če sta bili obe liniji, ki se križata, na isti uri, križanje ni delovalo pravilno. Z vpeljavo druge ure smo dosegli, da se liniji med seboj ne motita.
2. Drugi primer pa so AND vrata, ki so implementirana z majoritetnimi vrati. Z uporabo različnih ur smo dosegli, da oba vhoda prideta v vrata ob isti uri in s tem zagotovili pravilno delovanje le teh.

Pri uporabi večih ur je potrebno paziti, da je na vseh linijah, ki potekajo od vhoda vezja do izhoda, enako število različnih ur. Tudi vsi izhodi morajo biti na isti uri. V nasprotnem primeru vezje ne deluje pravilno.

Zakasnitev vezja našega vezja je  $2\frac{1}{4}$  urine periode. Skupaj smo v vezju porabili 406 celic.

## 4. Posebnosti pri realizaciji

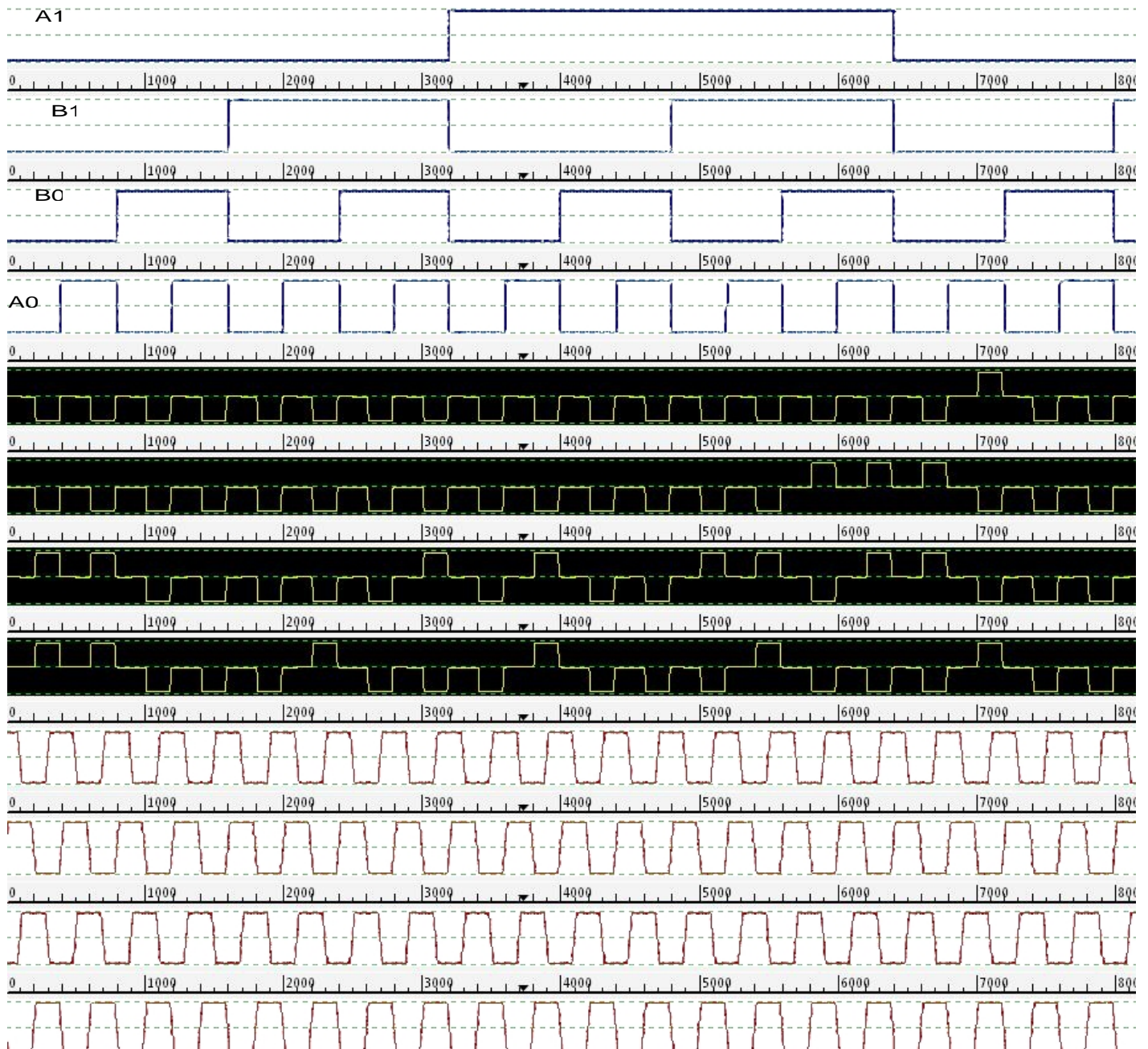
Sprva so nam veliko težav povzročala križanja vodil, ki so bila vsa v celoti na isti uri. Najprej smo poskusili narediti vodila vzporedno enega ob drugem, ampak smo zaradi nedelovanja vezja načrt opustili.

Nato smo želi zmanjšati število križanj vodil tako, da bi vezje realizirali v več plasteh. Na začetku je kazalo, da bo tako implementirano vezje manj problematično in bolj kompaktno. Ampak na žalost se je tudi ta implementacija ponovno izkazala za slabo, saj vezje ni delovalo pravilno.

Na koncu smo se odločili za implementacijo vezja le na eni plasti, težave s križanjem vodil pa smo rešili tako, da smo strateško razporedili vodila ter elemente. V veliko pomoč pri zagotavljanju stabilnosti delovanja vezja pa je prinesla tudi uporaba vseh štirih ur.

## 5. Rezultati simulacije

Veže smo simulirali tako, da smo za vhod določili vektor s štirimi vhodi, ki je predstavljal dvo-bitni številki A in B. Po simulaciji nam QCA Designer izriše graf, ki ga vidimo na spodnji sliki.



Slika 6: Rezultati simulacije

Pri interpretaciji je potrebno upoštevati, da smo pri implementaciji uporabili več različnih ur, zato moramo izhod gledati zakasnjeno glede na vhode. Vhodnim celicam je določena ura številka 0, izhodnim celicam pa ura številka 1. Zato moramo pri interpretaciji rezultatov gledati za dva in četrtno urinega cikla zakasnjeno uro številka 1.

Za lažje preverjanje pravilnosti delovanja s pomočjo grafov, ki so rezultat simulacije, smo uporabljali spodnjo pravilnostno tabelo.

$A_1$	$B_1$	$B_0$	$A_0$	$O_3$	$O_2$	$O_1$	$O_0$
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	1
0	1	0	0	0	0	0	0
0	1	0	1	0	0	1	0
0	1	1	0	0	0	0	0
0	1	1	1	0	0	1	1
1	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0
1	0	1	0	0	0	1	0
1	0	1	1	0	0	1	1
1	1	0	0	0	1	0	0
1	1	0	1	0	1	1	0
1	1	1	0	0	1	1	0
1	1	1	1	1	0	0	1

*Tabela 3: Pravilnostna tabela dvo-bitnega množilnika uporabljena pri testiranju*

## 6. Zaključek

Skozi seminarsko nalogo smo dodobra spoznali orodje QCA Designer. Res je, da nam omogoča gradnjo in simulacijo vezij sestavljenih iz kvantnih celičnih avtomatov brez predhodnega znanja fizike, vendar pa še zdaleč ni idealno orodje za kakšno resnejše delo. Večkrat smo prišli do spoznanja, da je napaka v QCA Designerju in da z našo shemo praviloma ni bilo nič narobe.

Ko smo rešili vse težave s povezovanjem in smo sestavili celoten množilnik, je prišla na vrsto simulacija. Izkazalo se je, da na vsakih nekaj poskusov pride do motnje, ki je nismo znali obrazložiti. Tukaj pa bi nam verjetno prišlo prav znanje kvantne fizike, saj QCA Designer omogoča nekaj dodatnih nastavitvev parametrov. Tako bi verjetno rešili naključno motnjo.

Naučili smo se tudi uporabljati križanje linij, ki nam je bilo na začetku še španska vas. V današnjih tehnologijah vezij je brezizgubno križanje linij nemogoče, saj moramo za ta namen uporabiti večplastna vezja. Najprej smo se lotili podobnega pristopa tudi pri našem množilniku. Vendar se stvar kaj hitro zakomplicira po nepotrebnem. Na koncu smo križanje linij dobro izkoristili.

Kolikor teorija kvantnih celičnih avtomatov na videz deluje zelo obetavno, je treba priznati, da je še kar nekaj ovir, preden bo nadomestila klasična elektronska vezja. Zagotovo pa prinaša nov pristop k oblikovanju sistemov za procesiranje.

## 7. Viri

- QCA Designer  
<http://www.qcadesigner.ca/>
- Domača stran predmeta ONT  
<http://lrs.fri.uni-lj.si/sl/teaching/ont/default.asp?lc=sl>