Realizacija dvo-bitnega množilnika v QCA

Seminarska naloga pri predmetu Optične in nano-tehnologije

Ljubljana, 14.11.2008

Seminarsko nalogo izdelali :

Ivo Križman (63050060)

Miha Nagelj (63050062)

Tom Vodopivec (63050129)

Davor Sluga (63050108)

Kazalo

1. Uvod

Naloga seminarske naloge je kot nam že naslov pove realizacija 2-bitnega množilnika s pomočjo kvantnih celularnih avtomatov (QCA). Realizacije smo se lotili na 2 načina :

- Z minimizacijo funkcije množenja za primer dveh dvo-bitnih vhodov
- Z izdelavo univerzalne celice množilnika

V nadaljevanju so podrobnosti obeh pristopov podrobneje razložene.

2. Realizacija množilnika s pomočjo minimizacije

2.1 Razlaga

V naši realizaciji ima 2-bitni množilnik vhoda A (2-bita) in B (2-bita) ter izhod Y (4-bite) ter deluje po naslednji pravilnosti tabeli:

A1	A0	B1	B0	Y3	Y2	Y1	Y0	
0	0	0	0	0	0	0	0	0 x 0 = 0
0	0	0	1	0	0	0	0	0 x 1 = 0
0	0	1	0	0	0	0	0	0 x 2 = 0
0	0	1	1	0	0	0	0	0 x 3 = 0
0	1	0	0	0	0	0	0	1 x 0 = 0
0	1	0	1	0	0	0	1	1 x 1 = 1
0	1	1	0	0	0	1	0	1 x 2 = 2
0	1	1	1	0	0	1	1	1 x 3 = 3
1	0	0	0	0	0	0	0	2 x 0 = 0
1	0	0	1	0	0	1	0	2 x 1 = 2
1	0	1	0	0	1	0	0	2 x 2 = 4
1	0	1	1	0	1	1	0	2 x 3 = 6
1	1	0	0	0	0	0	0	3 x 0 = 0
1	1	0	1	0	0	1	1	3 x 1 = 3
1	1	1	0	0	1	1	0	3 x 2 = 6
1	1	1	1	1	0	0	1	3 x 3 = 9

Z uporabo Veitchevih diagramov smo iz zgornje tabele dobili MDNO za vsak bit izhoda posebej:

$$Y0 = A1 \cdot B1$$
$$Y1 = (A0 \cdot \overline{B0} \cdot B1) + (A0 \cdot \overline{A1} \cdot B1) + (\overline{A0} \cdot A1 \cdot B0) + (A1 \cdot B0 \cdot \overline{B1})$$
$$Y2 = (A0 \cdot B0) \wedge (\overline{A0 \cdot A1 \cdot B0 \cdot B1})$$
$$Y3 = A0 \cdot A1 \cdot B0 \cdot B1$$

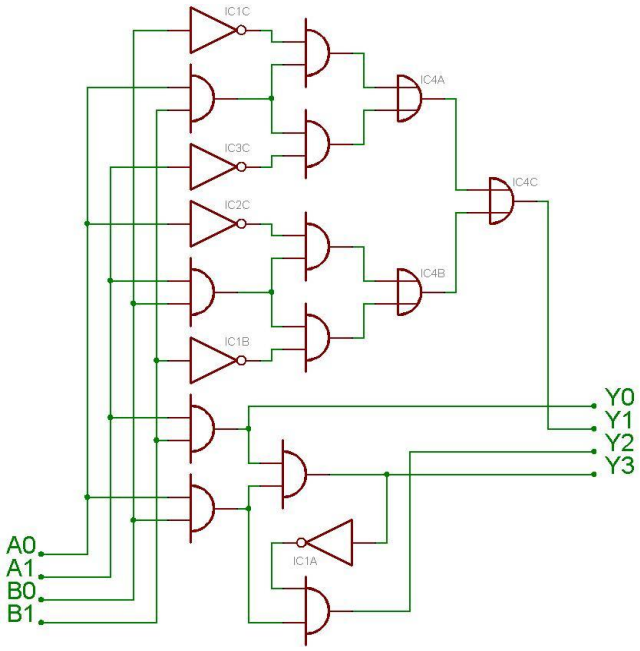
Pri implementaciji vezja v QCA-designerju, smo se odločili, da bomo uporabljali le 3-vhodna majoritetna vrata (za realizacijo funkcij AND in OR) ter negacije, zato smo enačbe za izhod (Y) preoblikovali v ustrezno obliko:

$$L0 = A0 \cdot B0$$
$$L1 = A0 \cdot B1$$
$$L2 = A1 \cdot B0$$

$$Y0 = A1 \cdot B1$$
$$Y3 = L0 \cdot Y0$$
$$Y2 = L0 \cdot \overline{Y3}$$
$$Y1 = ((L1 \cdot \overline{B0}) + (L1 \cdot \overline{A1})) + ((L2 \cdot \overline{A0}) + (L2 \cdot \overline{B1}))$$

L0, L1 in L2 so pomožne spremenljivke.

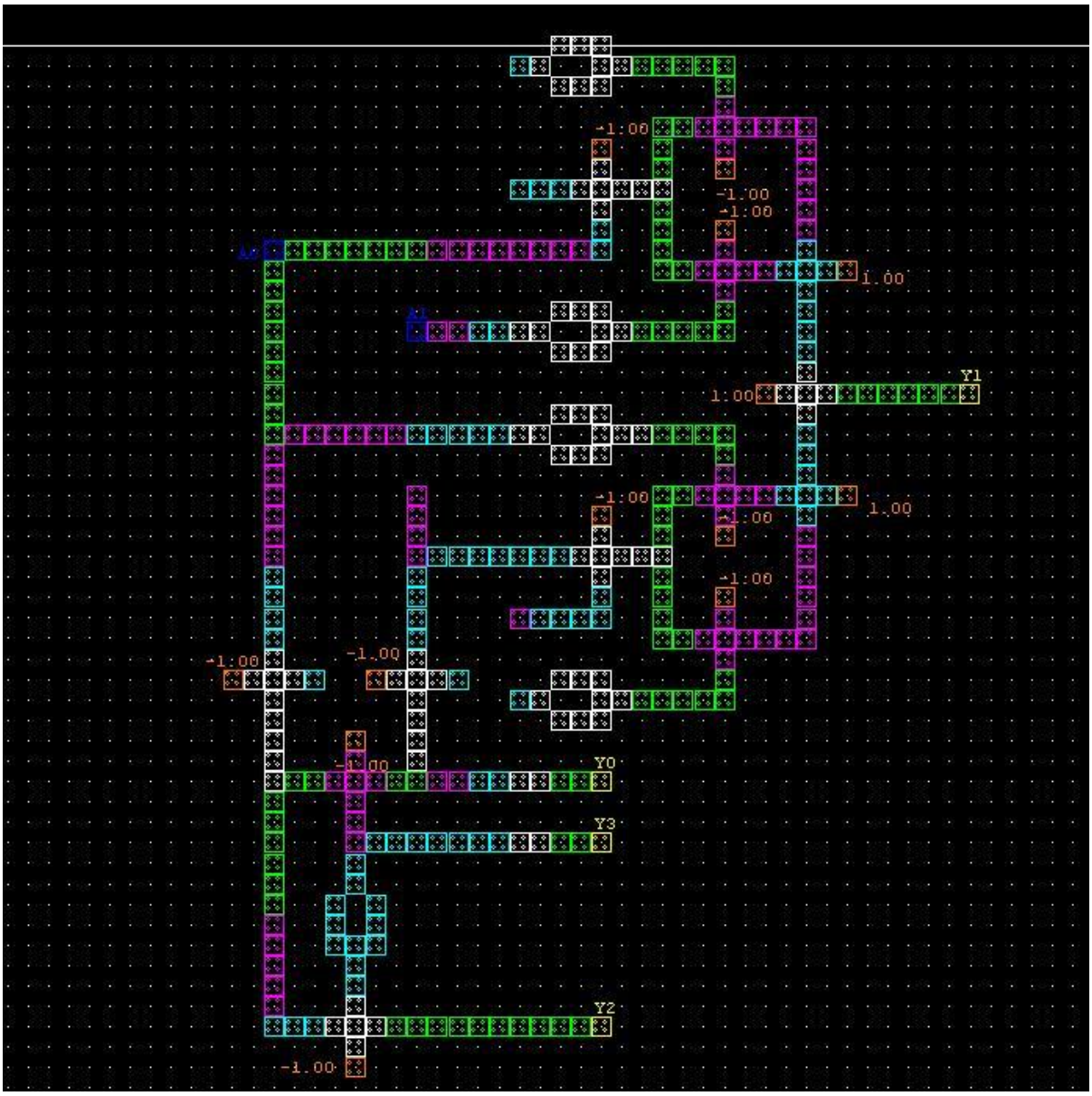
2.2 Logična shema vezja



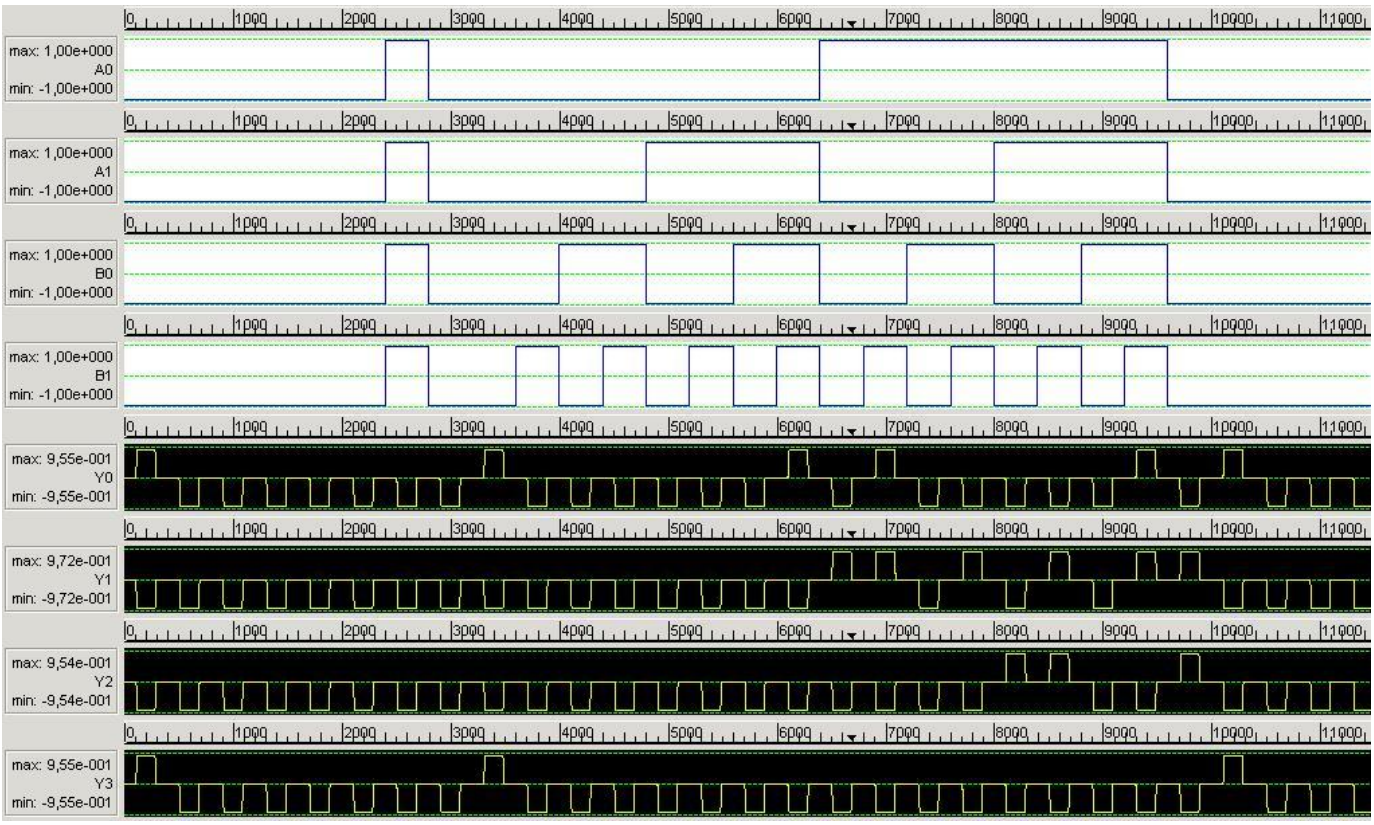
2.3 Realizacija v QCA Designer-ju

Pri realizaciji QCA smo uporabili 3 sloje, ki so nam omogočili križanje povezav. Postopek načrtovanja je oteževalo predvsem nestabilno delovanje samega programa. Celično strukturo avtomata smo zasnovali precej hitro, večje težave nam je povzročala pravilna postavitev ure. Tu je prišla do izraza uporaba postopka “trial and error”. Ugotovili smo, da je potrebna sprememba ure pri vsakem razcepu ter vratih. Motnje v strukturi povzročajo tudi celice z fiksno polarizacijo, ki jih ravno tako odpravimo s primerno uporabo ure. Časovna zakasnitev skozi vezje znaša dve urini periodi.

Na sliki je zaradi preglednosti prikazan le najvišji sloj.



2.4 Rezultati simulacije

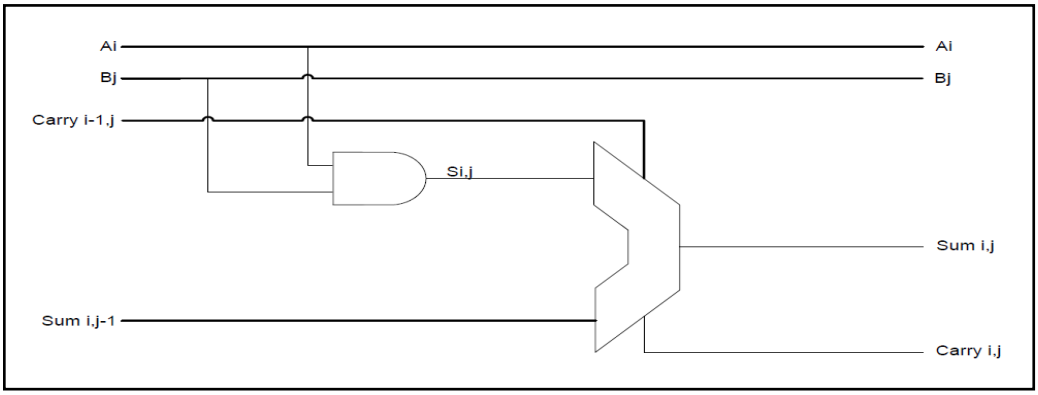


Iz rezultatov simulacije je razvidno pravilno delovanje vezja, kot tudi dejstvo, da so izhodni signali precej izraziti in brez vidnejših anomalij.

3. Realizacija z univerzalno celico množilnika

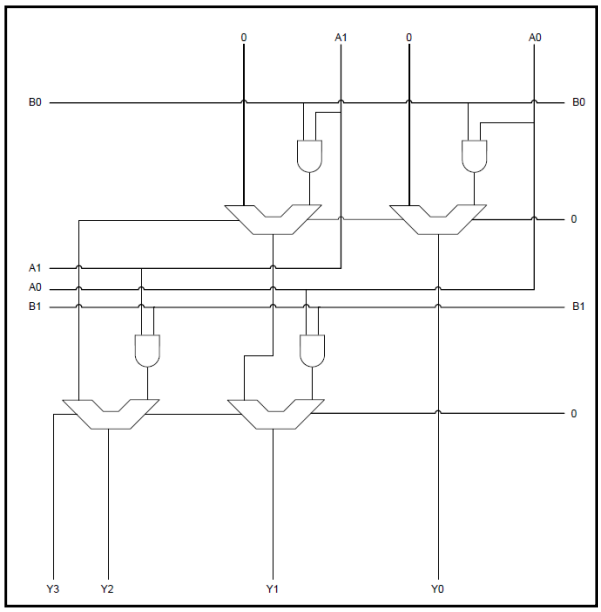
3.1 Univerzalna celica

Takšna realizacija posnema postopek množenja, kot se ga vsi najprej naučimo v šoli – množenje po vrsticah. Za takšno množenje potrebujemo univerzalno celico, ki predstavlja eno izmed števk, ki jo dobimo kot delni rezultat množenja v tisti vrstici. Postopek se izvaja v več vrsticah, v našem primeru za množenje dveh dvobitnih števil potrebujemo 2 vrstici, vsaka pa vsebuje 2 takšni celici. Spodnja slika prikazuje logično shemo univerzalne celice, ki je uporabljena pri množenju.



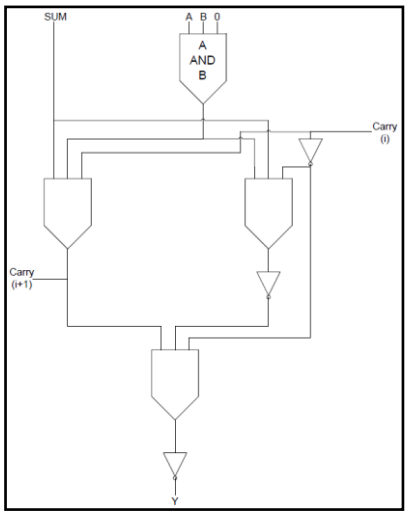
Z eno celico lahko realiziramo eno-bitni množilnik, in sicer, tako da operanda pripeljemo na vhoda Ai in Bj, vhodna Carry in Sum postavimo na '0', izhodna bita pa dobimo na izhodih Sum(LS) in Carry(MS).

Za realizacijo 2-bitnega množilnika potrebujemo 4 takšne celice. Razporedimo jih v 2 vrstici – nivoja, po dve celici na vrstico. Povežemo jih kot prikazuje spodnja shema.

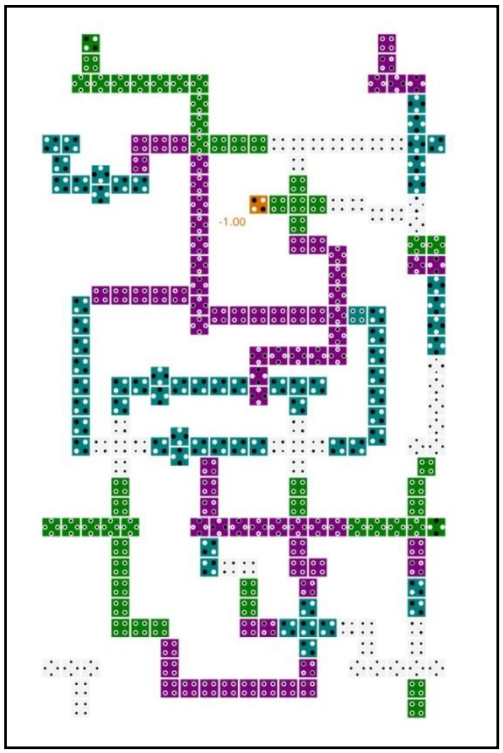


3.2 Realizacija množilnika v QCA Designer-ju

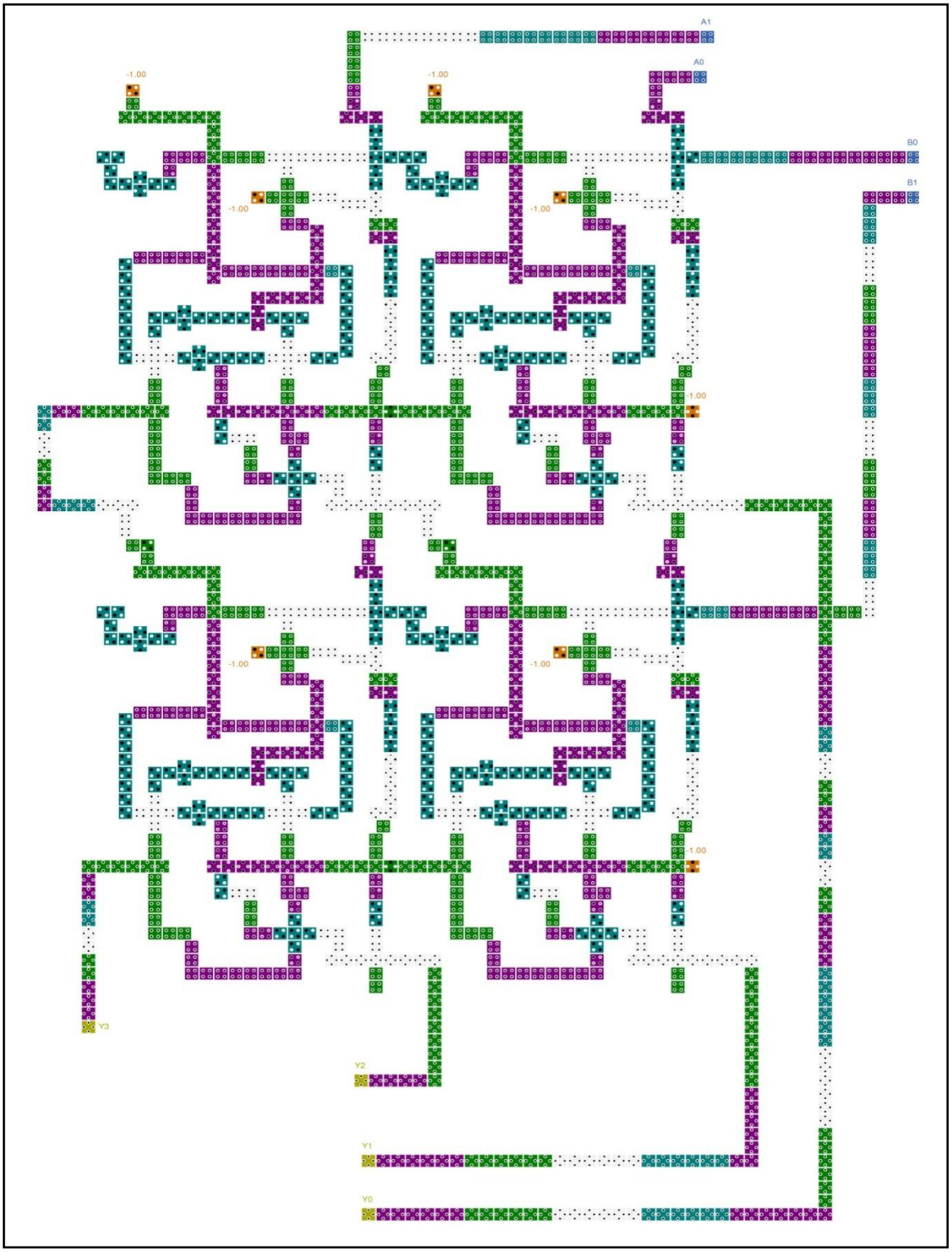
Če želimo množilnik realizirati z QCA, moramo logično shemo celice preurediti, tako da bo vsebovala osnovne elemente, ki jih lahko z QCA realiziramo. Spodnja slika prikazuje logično shemo realiziacije takšne celice z QCA osnovnimi elementi, ki je logični ekvivalent shemi na prejšni strani. Zgornja majoritetna vrata predstavljajo AND vrata, spodnji del sheme pa deluje kot enobitni seštevalnik.



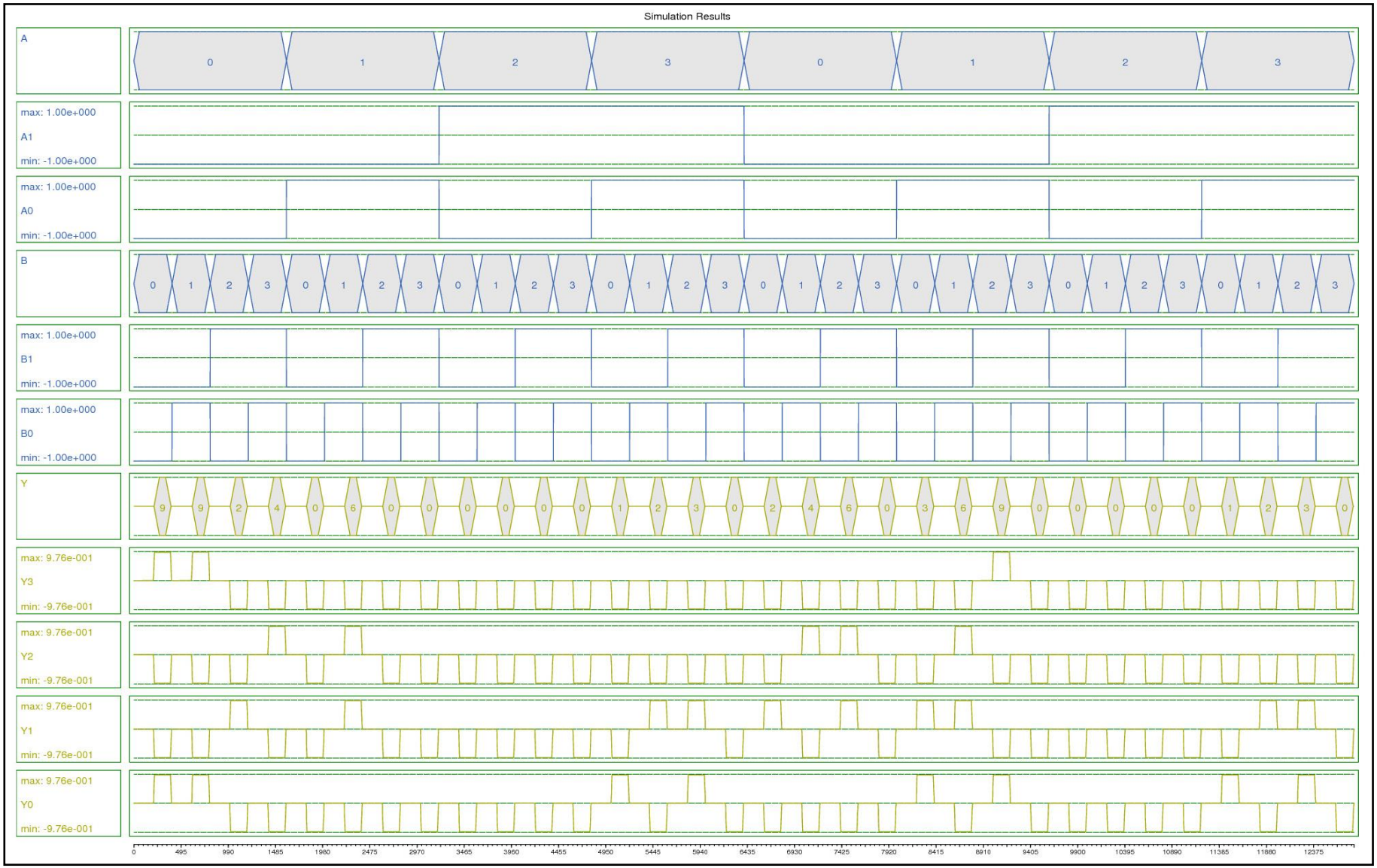
Realizacija celice v QCA Designer-ju je manj pregledna in enostavna, saj imamo veliko težav pri »kotih«, križanjih in realizacijah vrat. Spodnja slika prikazuje realizacijo celice množilnika v QCA Designer-ju.



Za sintezo dvobitnega množilnika moramo povezati 4 takšne celice, tako kot je prikazano v logični shemi. Tudi tukaj moramo upoštevati posebnosti QCA arhitekture, poskrbeti moramo predvsem za pravilne zakasnitve med univerzalnimi celicami.



3.3 Rezultati



Iz rezultatov je razvidno, da množilnik pravilno deluje. Rezultat se na izhodu pojavi z zakasnitvijo sedmih urinih period. Kot je značilno za QCA arhitekturo množilnik deluje cevovodno. To pomeni, da lahko vhod nastavimo vsako periodo, in tudi rezultati bodo izstavljeni vsako urino periodo, z zakasnitvijo sedmih.

4. Zaključek

V seminarski nalogi smo pokazali, da obe realizaciji dvo-bitnega množilnika lahko izvedemo s QCA. Če potrebujemo samo dvo-bitni množilnik je bolje, da izberemo realizacijo s pomočjo minimizacije funkcije, saj s tem močno zmanjšamo število uporabljenih celic in pohitrimo delovanje. Druga realizacija je namenjena predvsem posnemanju človeku bolj naravnega pristopa k množenju. Omogoča enostavno razširitev množilnika na več bitna števila, pri čemer se na račun tega povečuje zakasnitev izstavitve rezultata na izhodu.