

# Dvobitni delilnik

Seminarska naloga pri predmetu Optične in nanotehnologije

**Tine Mele (63050173), Gašper Bistovič (63050014), Borut Jevševar (63050050), Andrej Veber (63030075)**

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

## 1. Uvod

Pri predmetu Optične in nanotehnologije smo z orodjem QCADesigner implementirali dvobitni delilnik. QCADesigner je odprtokodno simulacijsko orodje za delo s kvantnimi celičnimi avtomati. Dobra lastnost tega orodja je, da uporabniku ni potrebno paziti na vse fizikalne lastnosti kvantnih celic. Vseeno pa je potrebno upoštevati določena pravila načrtovanja (predolge linije, kolena, fan-out, urin zamik, sekanje rotiranih linij, ...).

## 2. Implementacija dvobitnega delilnika

### 2.1. Funkcije dvobitnega delilnika

Pri implementaciji delilnika smo najprej napisali pravilnostno tabelo in določili izhodne funkcije. Na izhodu imamo 4 funkcije, dve za rezultat in dve za ostanek.

X1	X2	Y1	Y2	F1	F2	O1	O2
0	0	0	0	X	X	X	X
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	X	X	X	X
0	1	0	1	0	1	0	0
0	1	1	0	0	0	0	1
0	1	1	1	0	0	0	1
1	0	0	0	X	X	X	X
1	0	0	1	1	0	0	0
1	0	1	0	0	1	0	0
1	0	1	1	0	0	1	0
1	1	0	0	X	X	X	X
1	1	0	1	1	1	0	0
1	1	1	0	0	1	0	1
1	1	1	1	0	1	0	0

**Tabela 1.** Pravilnostna tabela

F1 in F2 predstavljata celoštevilski rezultat, O1 in O2 pa ostanek pri deljenju. To tabelo smo minimizirali in dobili sledeče funkcije:

$$F1 = X1 \wedge \neg Y1$$

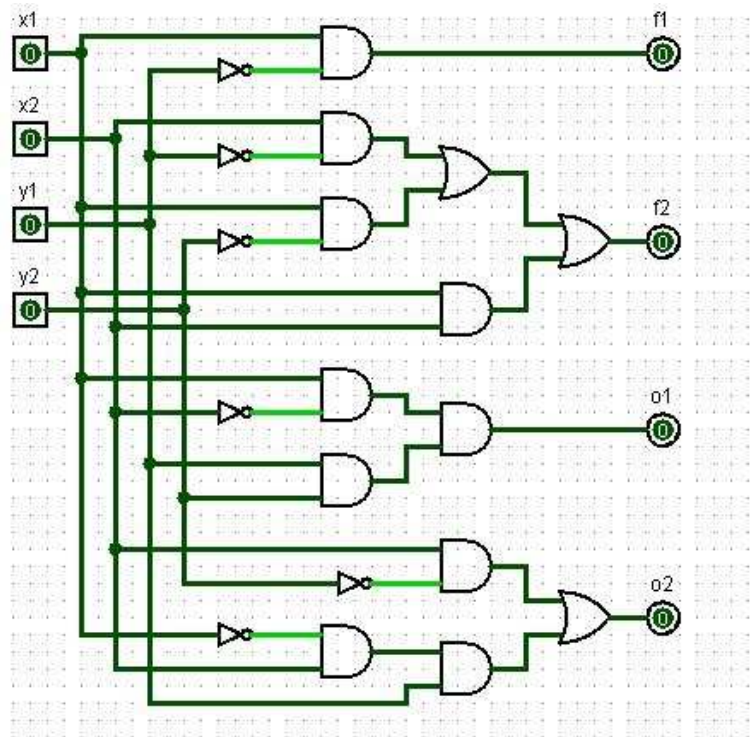
$$F2 = (X2 \wedge \neg Y1) \vee (X1 \wedge \neg Y2) \vee (X1 \wedge X2)$$

$$O1 = X1 \wedge \neg X2 \wedge Y1 \wedge Y2$$

$$O2 = (X2 \wedge \neg Y2) \vee (\neg X1 \wedge X2 \wedge Y1)$$

Deljenje z 0 je nedefinirano, zato je lahko rezultat karkoli. Na to napako mora paziti uporabnik. Pri realizaciji bi lahko dodali dodatna or vrata ( $Y1 \vee Y2$ ), ki nam zaznajo napako pri deljenju z 0.

## 2.2. Logična shema vezja

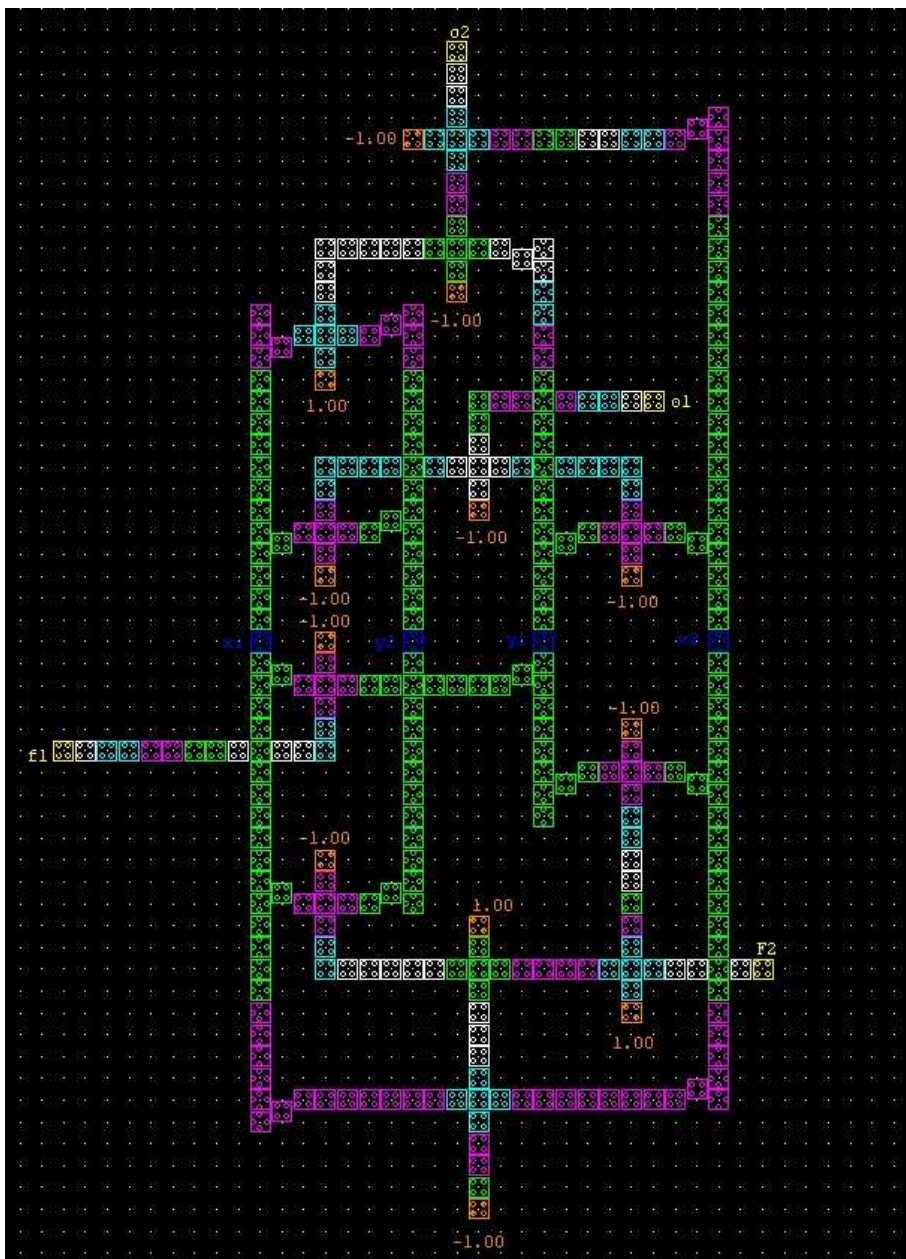


Slika 1. Logična shema vezja

Za realizacijo smo porabili:

- 4 vhode
- 4 izhode
- 12 majoritetnih vrat (9 and in 3 or)

### 2.3. Realizacija z QCADesigner-jem

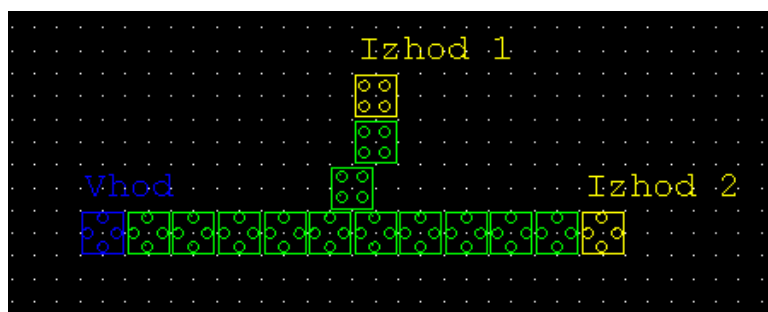


Slika 2. Realizacija v QCADesignerju

### 2.4. Osnovni gradniki vezja

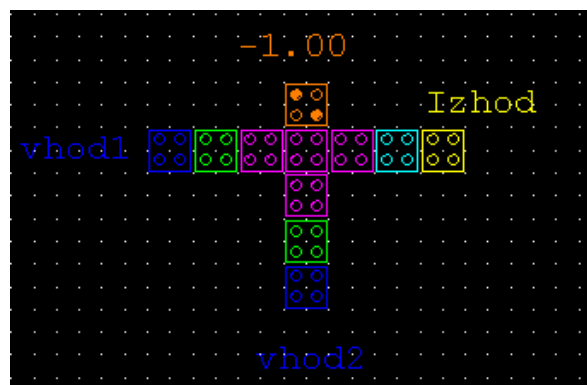
2.4.1. *Rotirana linija* Prednosti rotirane liniji so v tem, da se signal ne izgublja tako hitro kot pri navadni. Tako porabimo manj urinih zamikov. S funkcijo translate

naredimo izhod iz linije kot je prikazan na sliki »Izhod 1«, ki je lahko negiran ali pa ne. Posebnost rotiranih linij je tudi to, da jih lahko sekamo z nerotirano linijo.



Slika 3. Rotirana linija

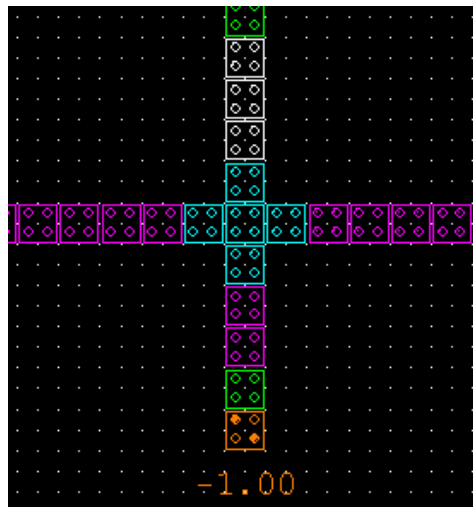
2.4.2. *Majoritetna vrata (and vrata)* Primer na sliki nam prikazuje and vrata. Sestavljena so iz dveh vhodov, konstante in izhoda. Pri tem moramo paziti na pravilno postavitev ure. Vhodi morajo biti enako zakasnjeni. Osrednjih 5 celic mora imeti en urin zamik več kot oba vhoda in enega manj kot izhod iz majoritetnih vrat. Razlika med AND in OR vrata je samo v konstanti, OR ima konstanto 1.



Slika 4. Majoritetna vrata and

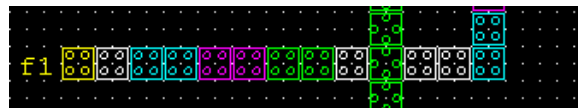
### 2.5. Problemi in posebnosti pri realizaciji

Pri implementaciji v QCADesignerju smo porabili 326 kvantnih celic. Izhod vezja je glede na vhod zamaknjen za 7 urinih zamikov oz. 1,75 urine period. S tem se stanje celic na linijah stabilizira, poveča se zanesljivost in doseže pravilnost delovanja vezja. Prva težava na katero smo naleteli je bila postavitev vhodov. Na koncu smo se odločili, da za vhodne signale uporabimo 4 rotirane linije. Tako smo se izognili težavam s križanjem linij in celotno vezje naredili na eni plasti. Do problema je prišlo tudi pri postavitvi ure na konstantah majoritetnih vrat. V določenih primerih smo morali uro premakniti za več urnih zamikov.



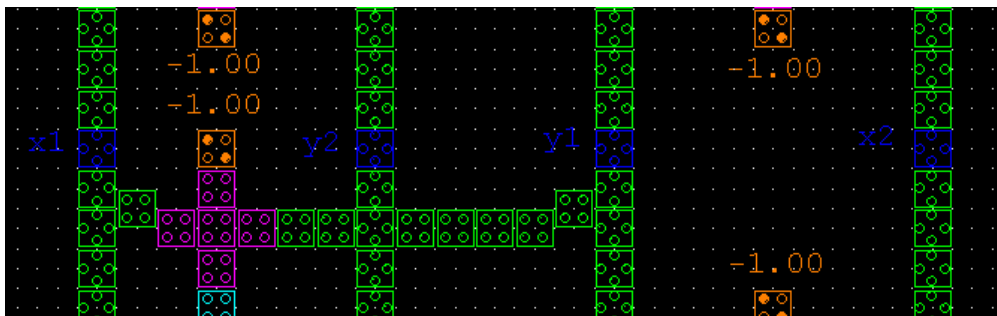
Slika 5. Zamik ure na majoritetnih vratih

Zahtevnost funkcij je različna, zato smo za nekatere porabili več urinih zamikov. Izhodi funkcij pa so sinhronizirani glede na najdaljšo funkcijo.



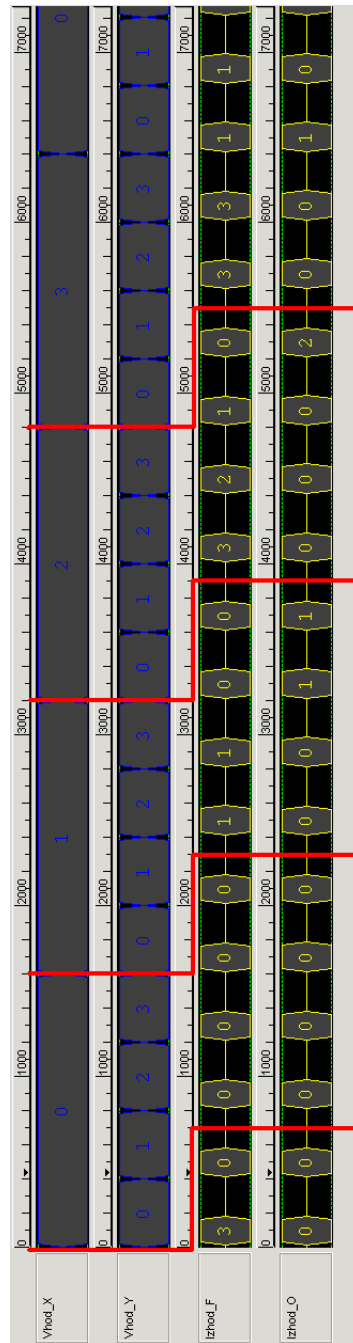
Slika 6. Zamik izhoda

Shema našega vezja se je močno poenostavila s tem, ko smo vhod premaknili na sredino linije. Z uporabo le tega se je zmanjšalo število urinih zamikov. V obratnem primeru smo imeli veliko problemov s predolgimi linijami in križanji. Vezje je postalo nestabilno.



Slika 7. Vhodi na sredini vezja

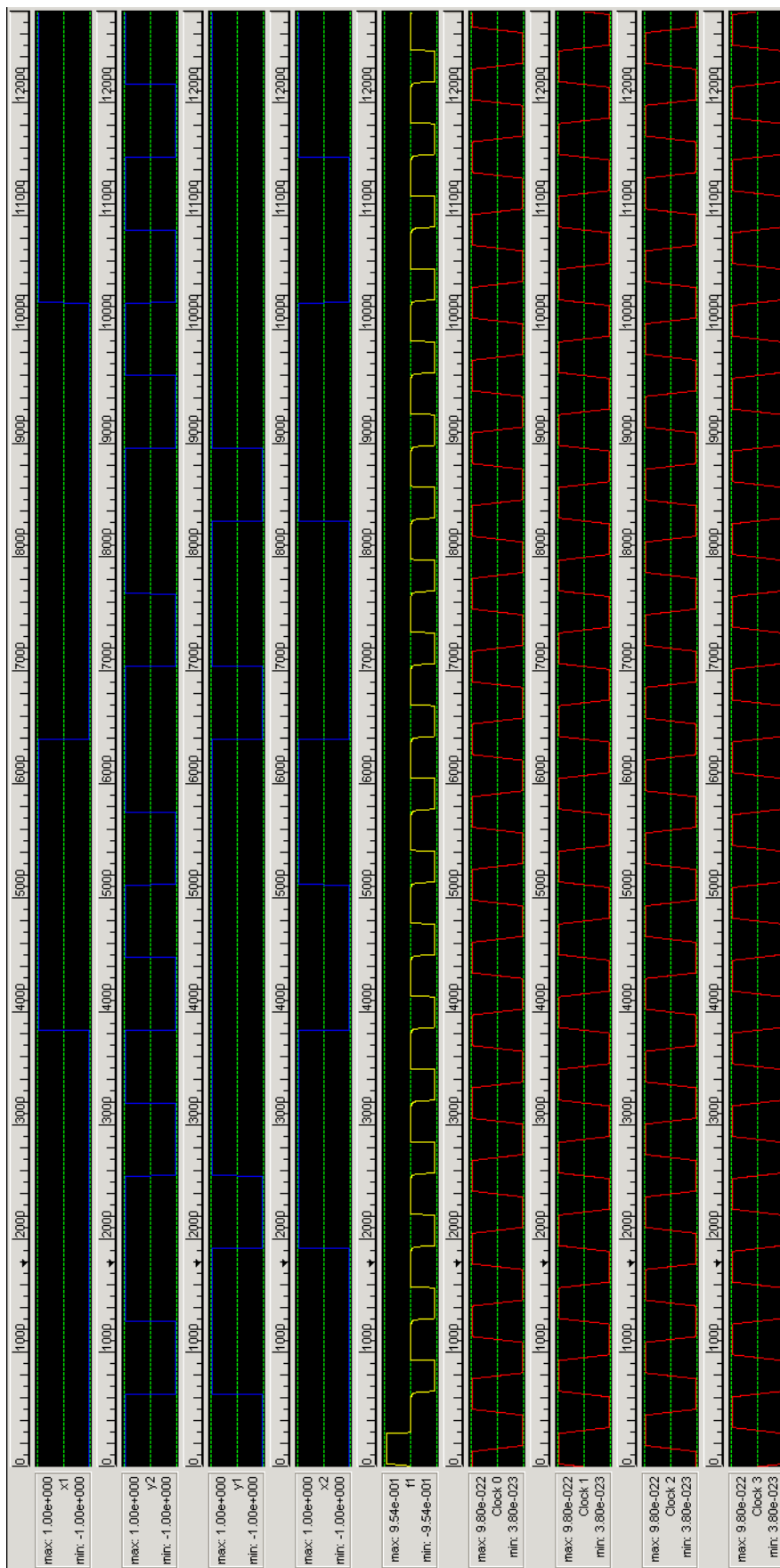
## 3. Rezultati testiranja



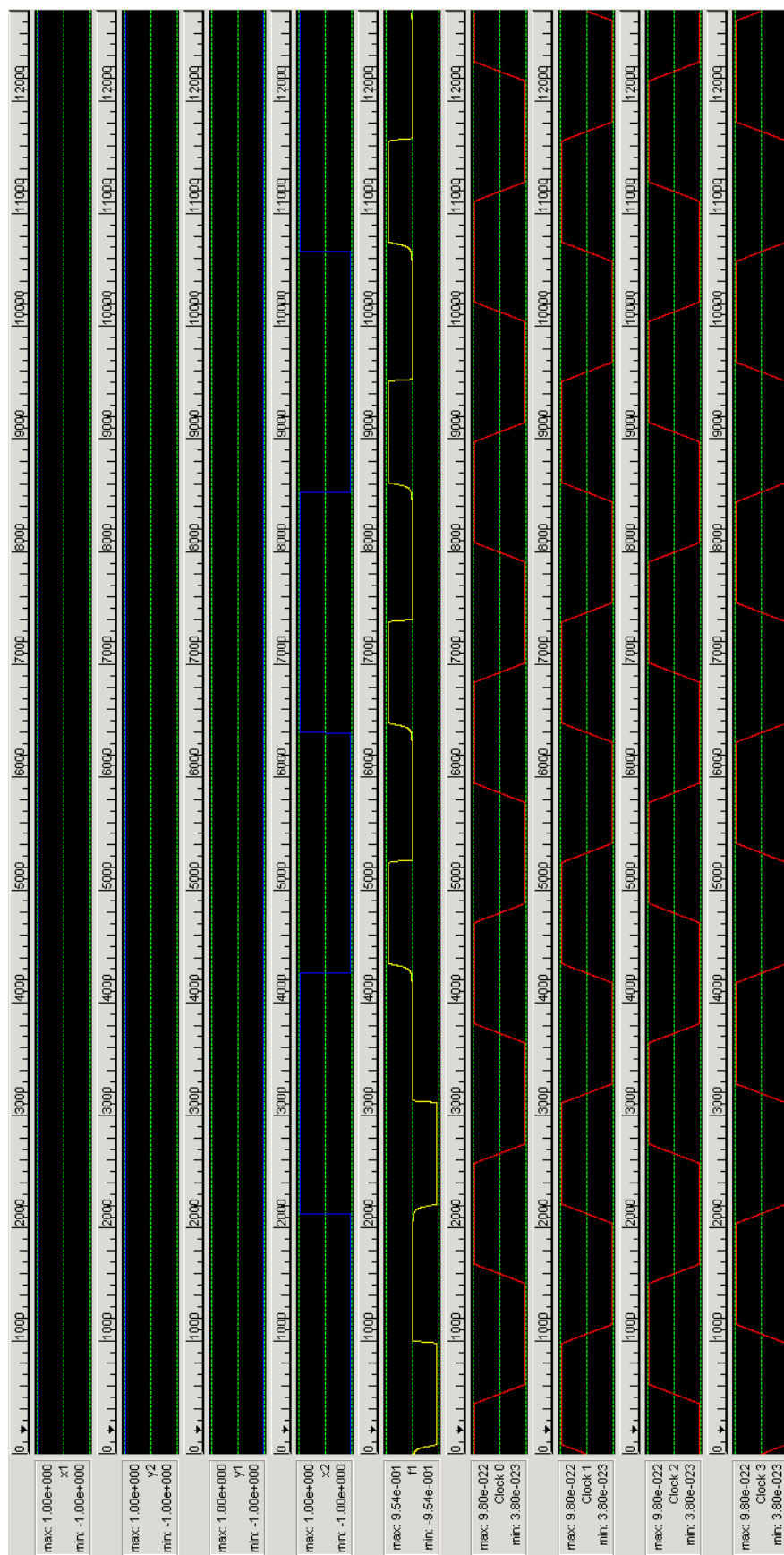
Slika 8. Rezultat simulacije

Na sliki vidimo rezultat simulacije, ki smo jo pognali v QCA Designerju. Vhod\_X delimo z Vhod\_Y. Rezultat nam predstavlja Izhod\_F, ostanek pa Izhod\_O. Izhod se pojavi zakasnen, kar nam označujejo rdeče črte. Paziti je potrebno na nedefinirane rezultate pri deljenju z nič. Pravilnost delovanja vezja smo preverjali z vhodnimi vektorji, ker je bilo tako najlažje odkriti napake. Na sliki je primer testiranja z vhodnimi vektorji za funkcijo F1. Pravilni rezultati na izhodu se pojavijo zakasneni za sedem urinih zamikov. Zaradi tega se vsi vektorji ponovijo dvakrat, tako dobimo pravi rezultat tudi za začetne vhode.





Slika 9. Rezultati testiranja za  $F1=0$



Slika 10. Rezultati testiranja za F1=1

Na tak način smo testirali tudi ostale izhodne funkcije ter popravljali manjše napake, dokler ni bil izhod vedno pravilen.

#### 4. Zaključek

S QCADesignerjem smo se soočili prvič, zato smo imeli na začetku kar nekaj težav. Napak pri simuliranju si velikokrat nismo znali razložiti. Ponavadi smo jih rešili z zakasnitvijo ure, določeni primeri pa so ostali nepojasneni. Zato smo imeli tudi pri enostavnem dvobitnem delilniku kar nekaj problemov. To nas je tudi pripeljalo do takega načina realizacije. Na koncu smo razmišljali tudi o realizaciji 4-bitnega delilnika. Vendar pa je bilo časa za implementacijo le tega premalo. Imeli bi več vhodov in izhodov, predvsem pa bi bile funkcije precej kompleksnejše.

#### 5. Literatura

- Spletna stran orodja QCADesigner:  
<http://www.qcadesigner.ca/>
- Domača stran predmeta ONT:  
<http://lrs.fri.uni-lj.si/sl/teaching/ont/>