

# QCA sekvenčna ALE

Miha Zidar, Anže Pečar, Matic Potočnik, Željko Plesac,  
Jan Varljen

Skupina 2 in 4

**Povzetek.** V seminarski nalogi bomo opisali zasnovo sekvenčne ALE s kvantnimi celičnimi avtomati, z uporabo programa QCAdesigner.

*Ključne besede.* kvantni celični avtomati, aritmetično-logična enota, modeliranje in simulacija

## Kazalo.

<b>1</b>	<b>Uvod</b>	<b>2</b>
1.1	Ideja . . . . .	2
1.2	Motivacija . . . . .	2
<b>2</b>	<b>Metode</b>	<b>2</b>
2.1	Opis naloge . . . . .	2
2.2	Ideje za realizacijo . . . . .	3
<b>3</b>	<b>Rezultati</b>	<b>3</b>
3.1	Izbira operacije . . . . .	3
3.2	Negacija . . . . .	5
3.3	Konjunkcija . . . . .	5
3.4	Seštevalnik/odštevalnik . . . . .	6
<b>4</b>	<b>Zaključek</b>	<b>8</b>
	<b>Literatura</b>	<b>8</b>

## 1. Uvod

### 1.1. Ideja

V tej seminarski nalogi bomo opisali zasnovo sekvenčne ALE s kvantnimi celičnimi avtomati (*angl.* quantum cellular automata – QCA). Modelirali jo bomo z uporabo odprtokodnega programa QCADesigner[1], za skice logičnih vezij, pa bomo uporabili TinyCAD.

Sekvenčnost enote tu pomeni, da enota ne izvaja operacij nad vsebino končno dolgih registrov, ampak sprejema tok bitov, nad posameznimi biti izvaja operacije in tudi svoj izhod podaja kot tok bitov. Za določene operacije tak pristop ni praktičen, ali pa je celo nemogoč, je pa mogoče na tak način implementirati poln funkcijski sistem, kar smo v nalogi tudi storili.

### 1.2. Motivacija

Predvideva se, da bo že čez nekaj desetletij minituarizacija in zmogljivost čipov, grajenih na siliciju, dosegla končno stopnjo in bo potrebno za večjo procesno moč preiti na drug osnovni material in najverjetneje tudi spremeniti pristop k modeliranju vezij. Ena izmed obetajočih alternativ so kvantni celični avtomati, ki obljublajo mnoge prednosti pred klasičnimi vezji:

- Možnost večnivojskih vezij
- Možnost križanja vodil
- Enostavna realizacija nekaterih časovnih vezij
- Potencialno nižja poraba in višji takt delovanja
- ...

## 2. Metode

V tem odseku bomo predstavili nekatere osnovne ideje in odločitve, ki smo jih nato uporabili pri realizaciji naše ALE.

### 2.1. Opis naloge

Realizirali bomo sekvenčno ALE, s funkcijsko polnim sistemom operacij. Za izbiro operacije bomo uporabili dva bita, enota pa bo podpirala naslednje operacije:

Operacija	Oznaka	Op. koda
NOT	$\neg$	00
AND	$\wedge$	01
ADD	$\oplus$	10
SUB	$\ominus$	11

**Tabela 1.** Seznam operacij

### *2.2. Ideje za realizacijo*

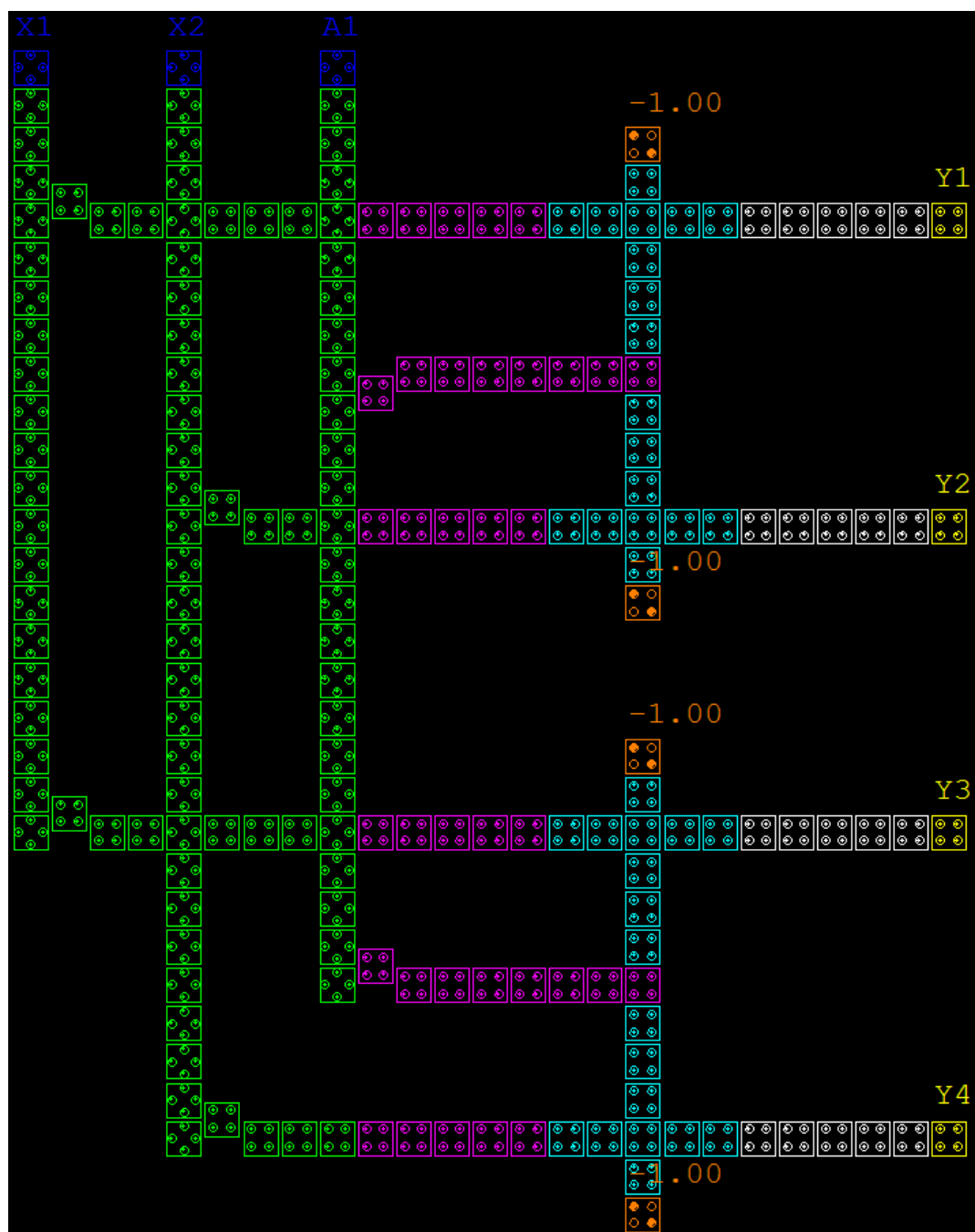
Operaciji NOT in AND sta že v osnovi bitni operaciji in je tako sekvenčna realizacija popolnoma naravna. Pri ADD s stališča sekvenčnosti delovanja tudi ni posebnih zapletov, pri SUB pa smo poskušali operirati, kot bi imeli števili zapisani v predstavitvi z dvojnimi komplementom.

## **3. Rezultati**

V tem delu seminarske naloge bomo predstavili realizacijo posameznih delov ALE. Pri realizaciji smo si pomagali s knjigo[2].

### *3.1. Izbira operacije*

Izbira operacije je realizirana z dvo-bitnim demultiplekserjem, ki izbira med operacijami NOT, AND, ADD in SUB, kot smo to zapisali v [Table 1](#).



Slika 1. Demultiplekser

### 3.2. Negacija

#### 3.2.1. Operacija

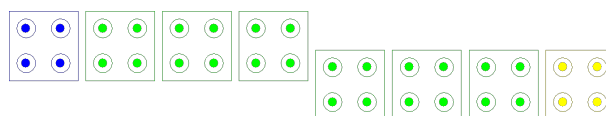
Negacija vrne obratno vrednost vhoda – torej za 0 vrne 1 in obratno.

Enačba:  $out = \neg x$

$x$	out
0	1
1	0

#### 3.2.2. Realizacija

Negacijo (operacijo NOT) smo realizirali s polovičnim zamikom celice.



**Slika 2.** NOT vrata

### 3.3. Konjunkcija

#### 3.3.1. Operacija

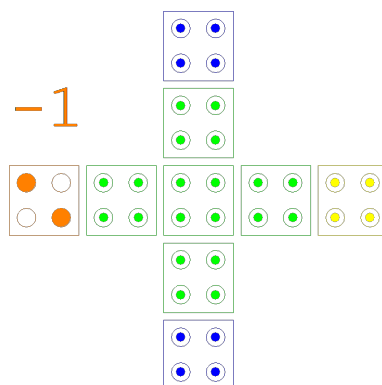
Konjunkcija vrne 1, kadar imata oba vhoda vrednost 1.

Enačba:  $out = x_1 \wedge x_2$

$x_1$	$x_2$	out
0	0	0
0	1	0
1	0	0
1	1	1

#### 3.3.2. Realizacija

Konjunkcijo (operacijo AND) smo realizirali z majoritetnimi vrati, ki imajo enega izmed vhodov nastavljenega na polarizacijo -1.



Slika 3. AND vrata

### 3.4. Seštevalnik/odštevalnik

#### 3.4.1. Operacija

Seštevalnik sešteje bita, hkrati pa upošteva in izračuna še bit za prenos.

Enačba:  $out = (x_1 \oplus x_2) \oplus c$

$c_{out} = (x_1 \wedge x_2) \vee (c \wedge x_2) \vee (c \wedge x_1)$

$x_1$	$x_2$	$c$	out	$c_{out}$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Odštevalnik podobno odšteje vhodna bita in upošteva prenose.

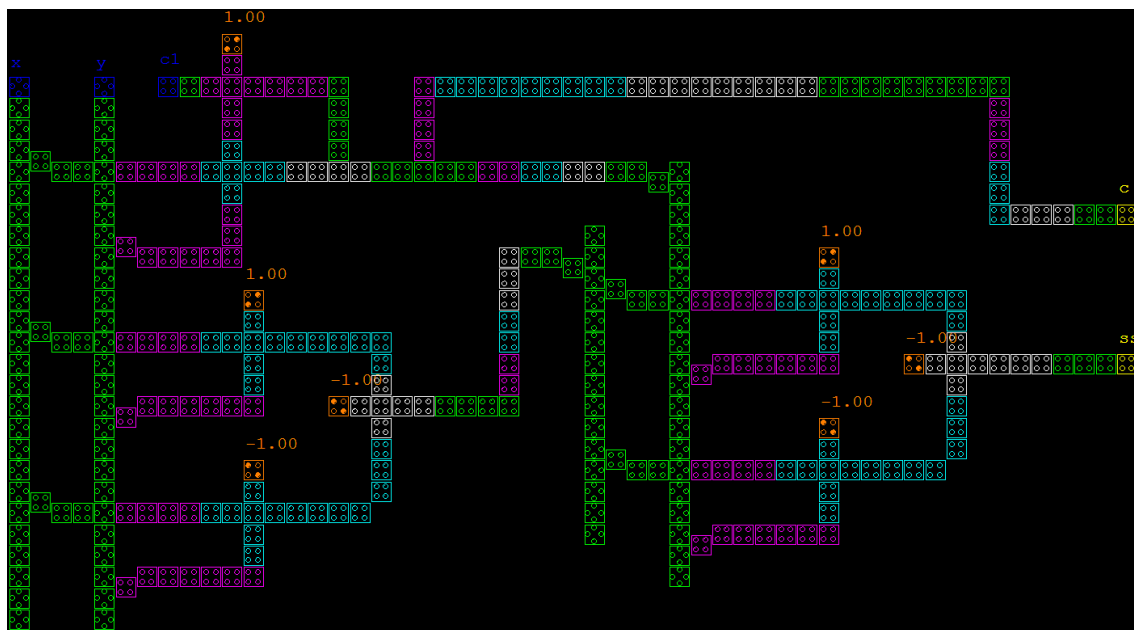
Enačba:  $out = (x_1 \oplus x_2) \oplus c$

$c_{out} = (c \wedge \overline{(x_1 \oplus x_2)}) \vee (\overline{x_1} \wedge x_2)$

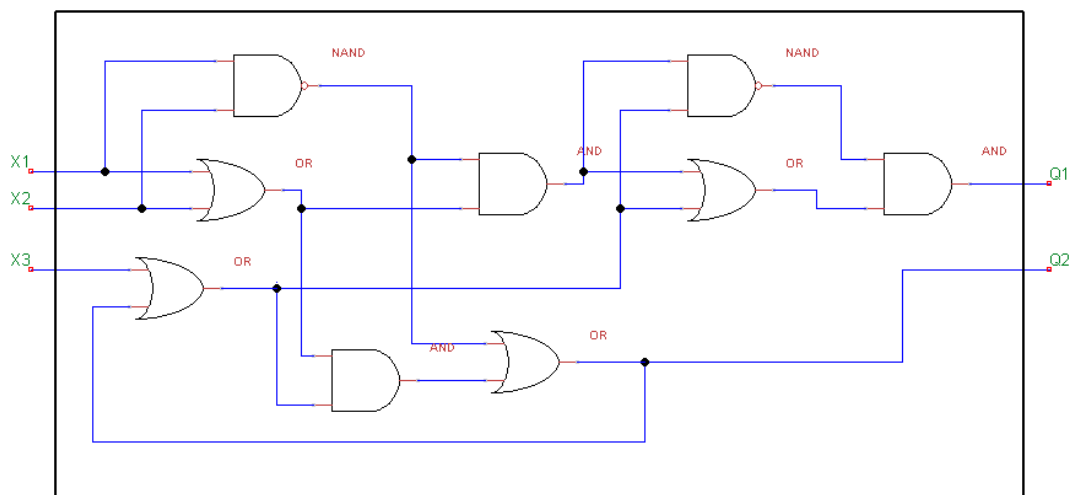
$x_1$	$x_2$	$c$	out	$c_{out}$
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

### 3.4.2. Realizacija

Seštevalnik in odštevalnik (operaciji ADD in SUB), smo želeli realizirati skupaj – pri odštevanju drugo število negiramo in mu pri prvem bitu prištejemo 1, kot da bi bilo zapisano v predstavitvi z dvojnim komplementom. To smo realizirali tako, da je možno določiti carry bit in bi ga pri prvem bitu operacije nastavili na 1.



Slika 4. Seštevalnik



Slika 5. Seštevalnik

#### 4. Zaključek

Z veliko truda in težavami s QCAdesigner-jem, nam je uspelo realizirati posamezne dele ALE, ni pa nam uspelo komponent sestaviti skupaj.

#### Literatura

- [1] K. Walus, T. Dysart, G. Jullien, A. Budiman, Qcadesigner: a rapid design and simulation tool for quantum-dot cellular automata, *IEEE Transactions on Nanotechnology* 3 (1) (2004) 26–31.
- [2] J. Virant, Načrtovanje nanoračunalniških struktur : uvod v nanoračunalniško logiko, Didakta, Radovljica, Slovenia, 2007.