



Digitalna vezja UL, FRI



Vaja 7 Kombinacijska vezja

N1: Inkrementer in dekrementer

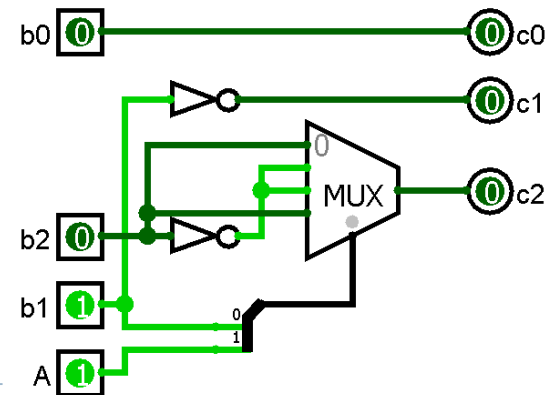
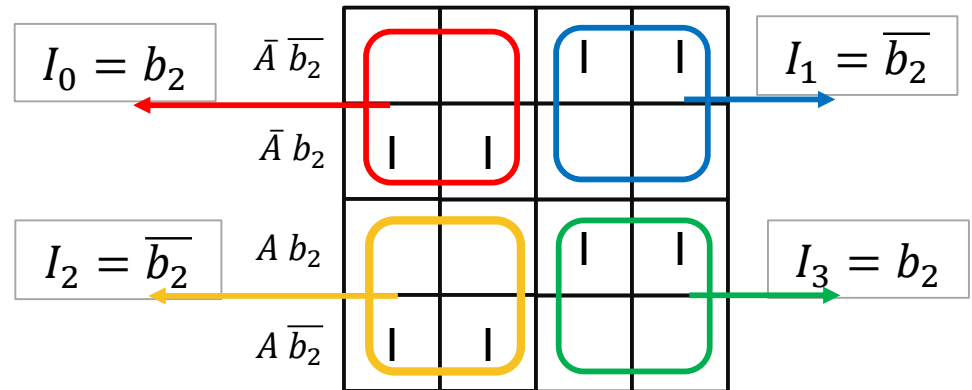
- Vaja 5: 3-bitna števila - $A=0$: $C=B + 2$ (Inkrement), $A=1$: $C=B - 2$ (Dekrement).

A	b_2	b_1	b_0	c_2	c_1	c_0
0	0	0	0	0	1	0
0	0	0	1	0	1	1
0	0	1	0	1	0	0
0	0	1	1	1	0	1
0	1	0	0	1	1	0
0	1	0	1	1	1	1
0	1	1	0	0	0	0
0	1	1	1	0	0	1
1	0	0	0	1	1	0
1	0	0	1	1	1	1
1	0	1	0	0	0	0
1	0	1	1	0	0	1
1	1	0	0	0	1	0
1	1	0	1	0	1	1
1	1	1	0	1	0	0
1	1	1	1	1	0	1

$$c_1 = \overline{b_1}, \quad c_0 = b_0$$

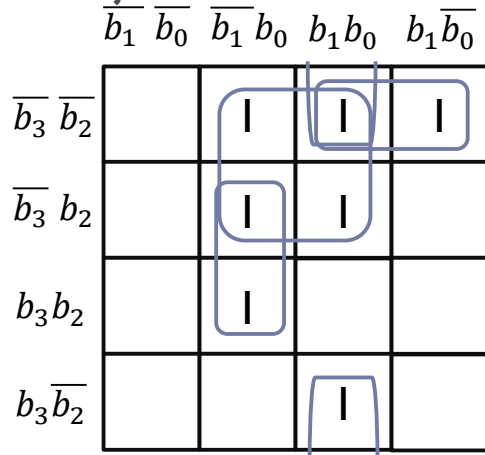
c_2 - 2-nalovni MUX: $A_1 = A, A_0 = b_1$

$$\overline{b_1} \overline{b_0} \quad \overline{b_1} b_0 \quad b_1 b_0 \quad b_1 \overline{b_0}$$

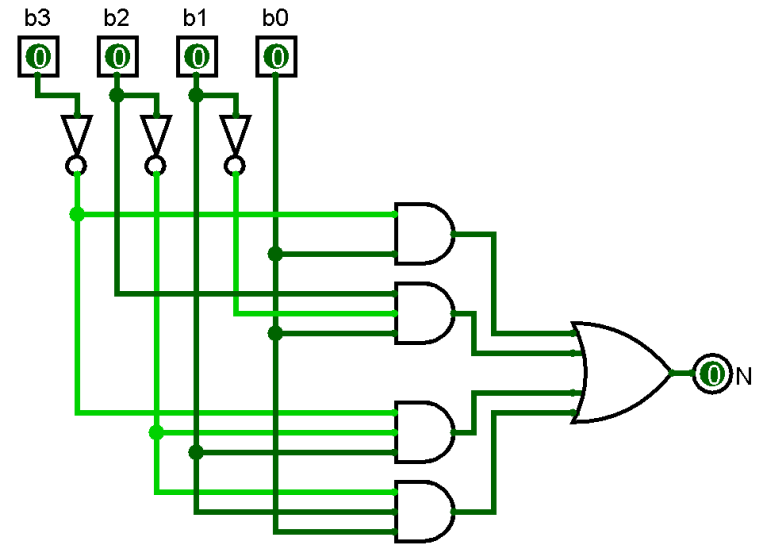


N2: Detekcija števila 1 in praštevil

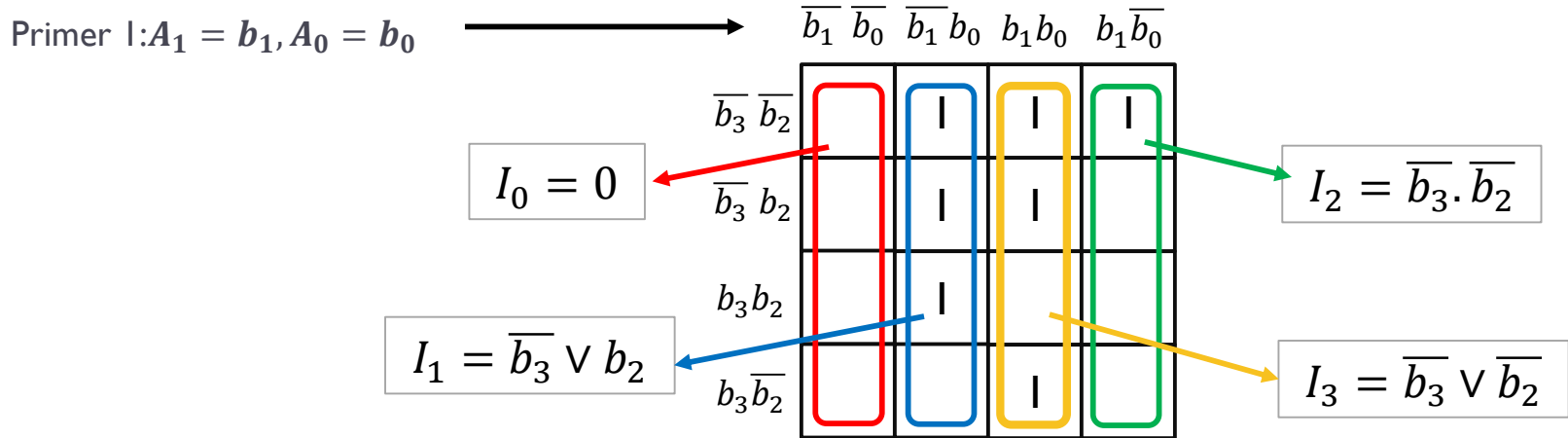
- ❑ Realizirajte vezje za detekcijo števila 1 in praštevil v 4-bitnem dvojiškem zapisu števila $N = (b_3, b_2, b_1, b_0)$. Izhodna funkcija $P = 1$, če je na vходу število 1 ali praštevilo.
- ❑ Naloge:
 - Zapis izhodne funkcije P v Karnaughejev diagram.
 - Minimizacija logične funkcije (minimalna vsota produktov ali MDNO)
 - Realizacija z 2-naslovnimi multiplekserji: $A_1 = b_1, A_0 = b_0$



$$N = \overline{b_3} \cdot b_0 \vee b_2 \cdot \overline{b_1} \cdot b_0 \vee \overline{b_3} \cdot \overline{b_2} \cdot b_1 \vee \overline{b_2} \cdot b_1 \cdot b_0$$



- Realizacija z 2-naslovnimi multiplekserji (Karnaugh):



- Realizacija z 2-naslovnimi multiplekserji – minimalna oblika

Primer 2: $A_1 = b_3, A_0 = b_2$

$$N = \bar{b}_3 \cdot b_0 \vee b_2 \cdot \bar{b}_1 \cdot b_0 \vee \bar{b}_3 \cdot \bar{b}_2 \cdot b_1 \vee \bar{b}_2 \cdot b_1 \cdot b_0 = \bar{b}_3 \cdot \bar{b}_2 \cdot (I_0) \vee \bar{b}_3 \cdot b_2 \cdot (I_1) \vee b_3 \cdot \bar{b}_2 (I_2) \vee b_3 \cdot b_2 \cdot (I_3)$$

Izračun: $I_0 = N(0,0, b_1, b_0), I_1 = N(0,1, b_1, b_0), I_2 = N(1,0, b_1, b_0), I_3 = N(1,1, b_1, b_0)$

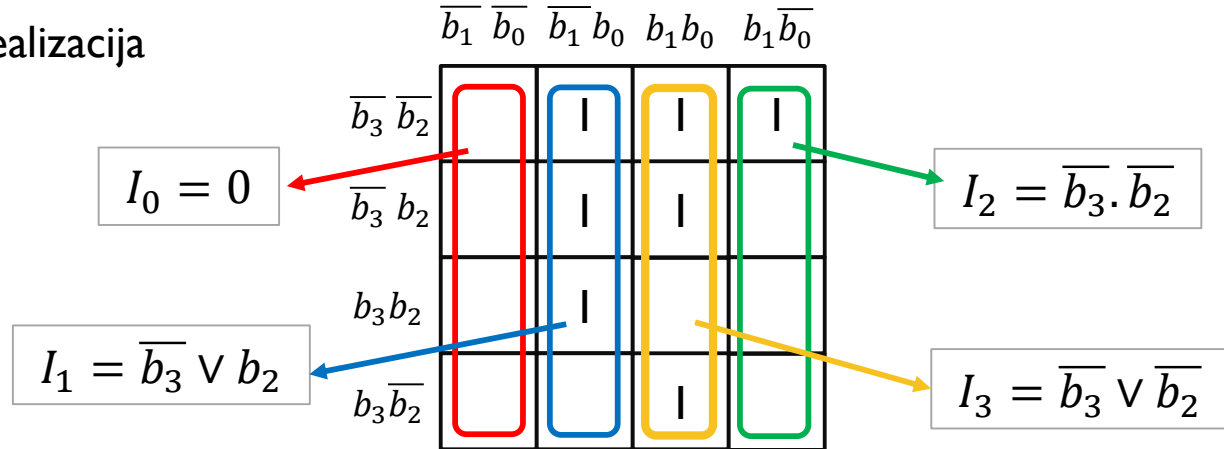
Primer 3: $A_1 = b_3, A_0 = b_1$

$$N = \bar{b}_3 \cdot b_0 \vee b_2 \cdot \bar{b}_1 \cdot b_0 \vee \bar{b}_3 \cdot \bar{b}_2 \cdot b_1 \vee \bar{b}_2 \cdot b_1 \cdot b_0 = \bar{b}_3 \cdot \bar{b}_1 \cdot (I_0) \vee \bar{b}_3 \cdot b_1 \cdot (I_1) \vee b_3 \cdot \bar{b}_1 (I_2) \vee b_3 \cdot b_1 \cdot (I_3)$$

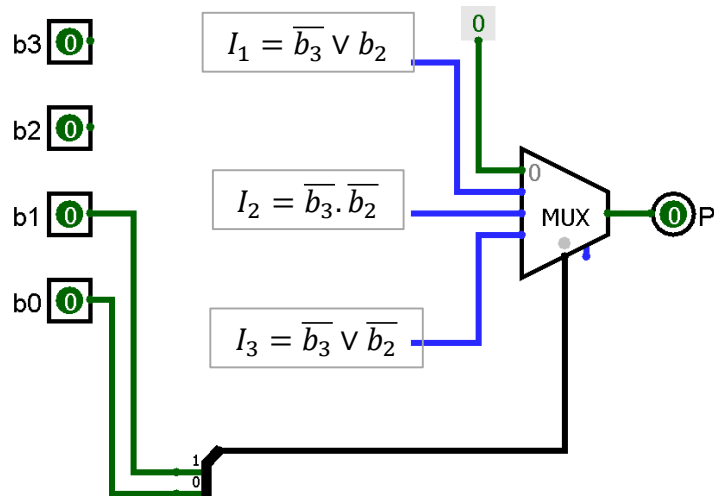
... še ostale možnosti za izbiro naslovnih spremenljivk:

Ugotovitev: Potrebna je večnivojska realizacija z 2-naslovnimi multiplekserji

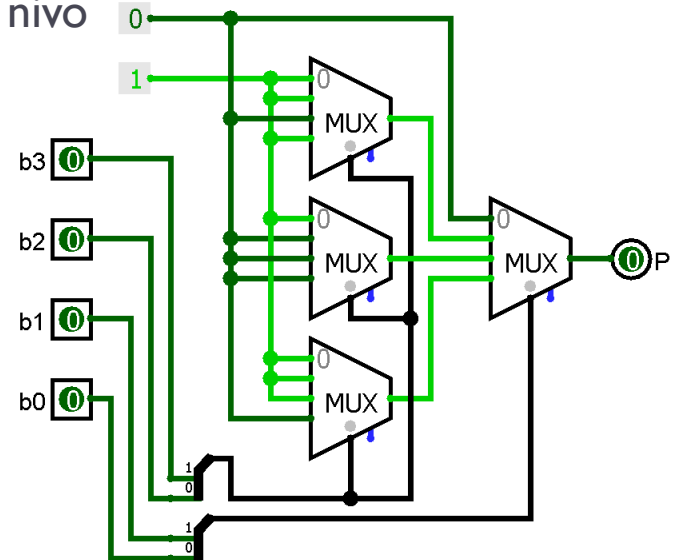
□ Kaskadna realizacija



1-Izhodni nivo

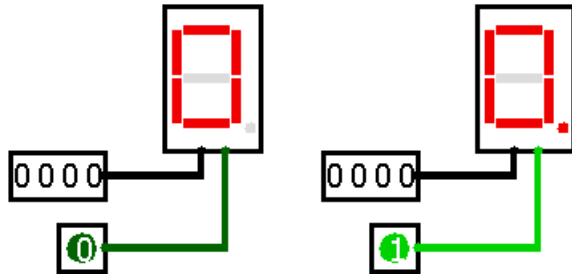


2- vhodni nivo

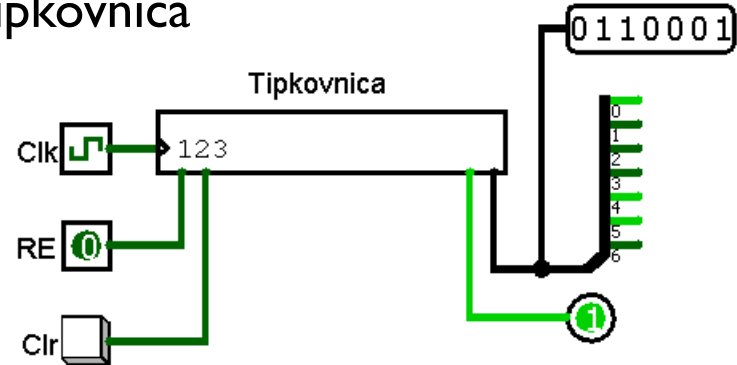


V/I moduli, medpomnilnik, primerjalnik

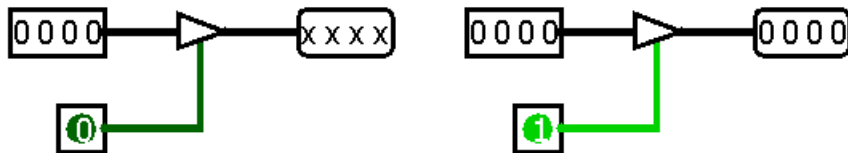
Prikazovalnik



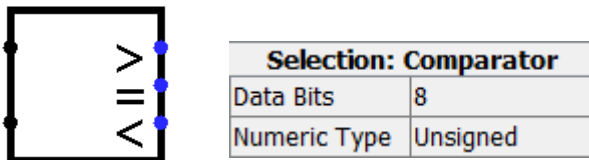
Tipkovnica



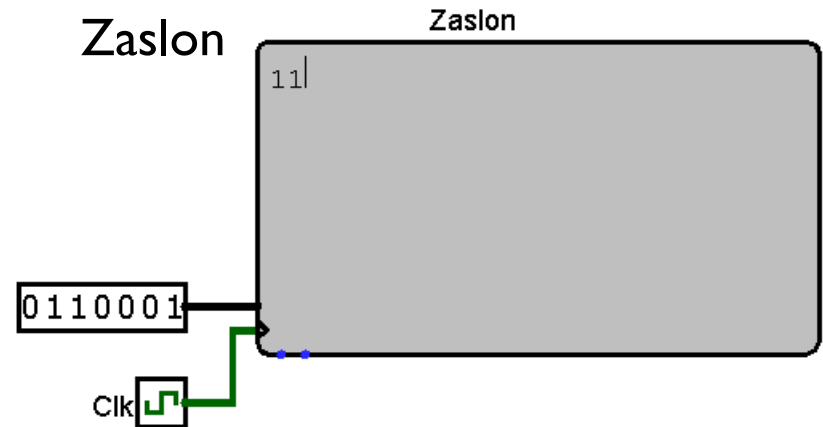
Medpomnilnik



Primerjalnik



Zaslon

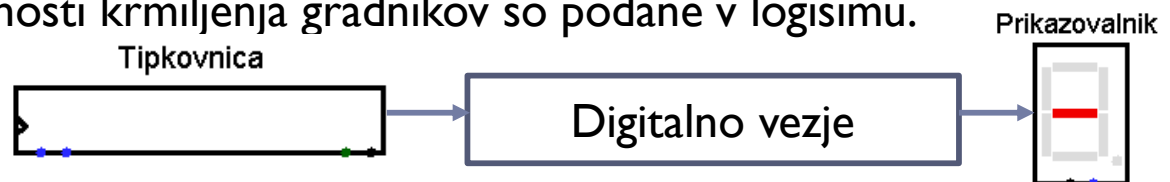


N3: Pretvorba **ASCII** → **bin**

I- Pretvorba cifre **BCD (0,1,...,9)**, ki je podana v 7-bitni kodi ASCII, v 4-bitni dvojiški zapis.

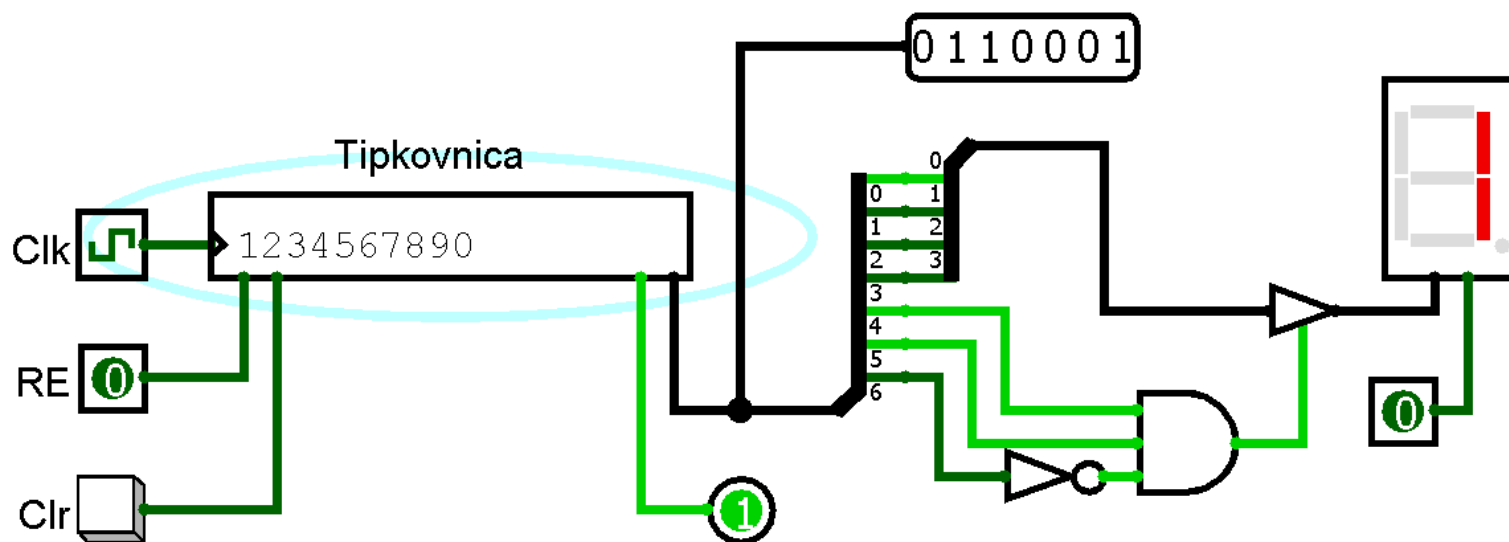
BCD	ASCII (7-bitov)	Bin (4-biti)
0	011 0000	0000
1	011 0001	0001
...
9	011 1001	1001

- V logisimu realizirajte vezje tako, da se ob vpisu cifre na tipkovnici njena vrednost prikaže na prikazovalniku (Hex Digit Display). Če vpisan znak ni število, naj bo izpisan znak minus (-). Uporabite potrebne gradnike, ki so bili predstavljeni oz. smo jih že poznali. Podrobnosti krmiljenja gradnikov so podane v logisimu.



- Vhodi: 7-bitna koda ASCII ($b_6, b_5, b_4, b_3, b_2, b_1, b_0$)
- Izhodi: spodnji 4-bitni kode ASCII: $B=(b_3, b_2, b_1, b_0)$
- Funkcija: če so $b_6, b_5, b_4 = 011$, se prikaže število (0,...,9), sicer se prikaže (-)

- Logično vezje – izpisali se bodo vsi znaki z 011 na zgornjih 3 bitih kode ASCII.
 - Primer: dvopičje (:) – ASCII (3A) - izpiše se A
 - Delovanje vezja, če vpišemo v tipkovnico poljuben znak: :, ,, =,
 - Ali je rezultat pravilen?
 - Če je odgovor NE, potem nadgradite vezje.

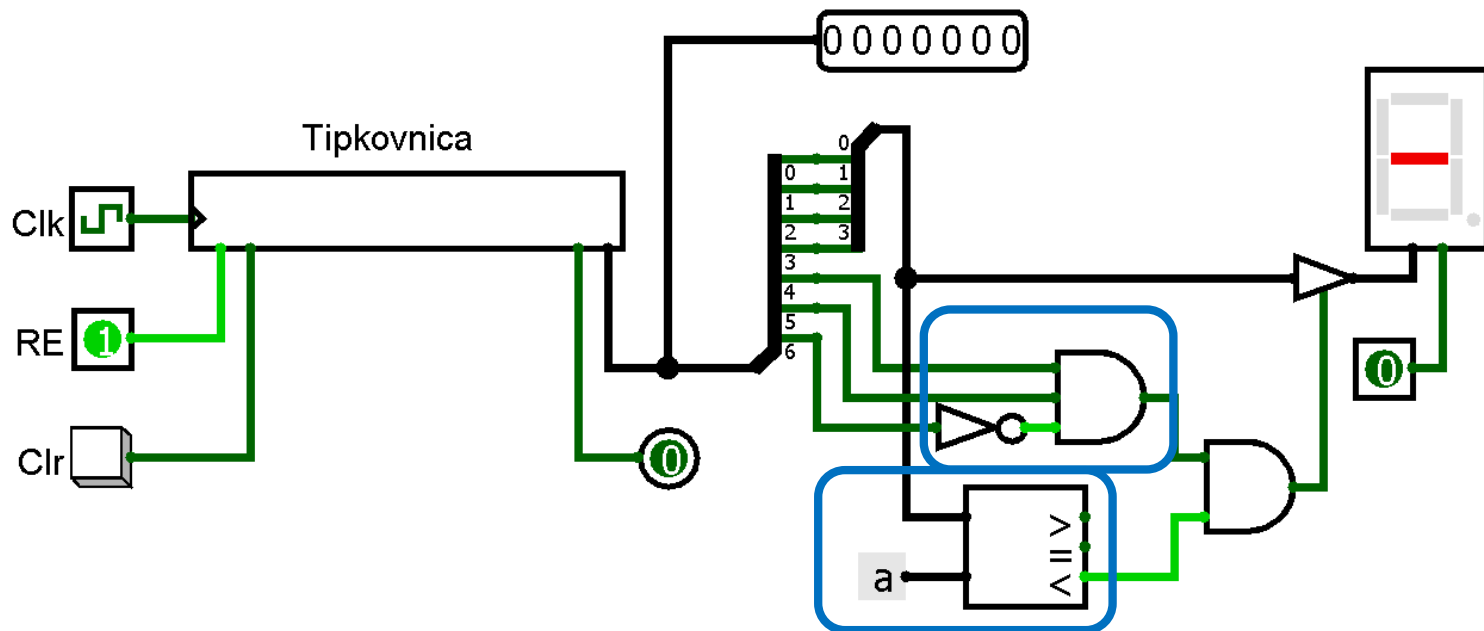


- Logično vezje : ASCII (0,1,...,9) → dvojiško (0000,0001,...,1001)

Primer:

I: 011 0001 → 0001

izpiše se 1, če so zgornji 3 biti kode ASCII 011 in so spodnji 4 biti kode ASCII manjši od 10 (a)



2- Pretvorba znaka (a,b,c,d,e,f), ki je podan v 7-bitni kodi ASCII, v 4-bitni dvojiški zapis.

znak	ASCII (7-bitov)	Bin (4-biti)
a	1100001	1010
b	1100010	1011
...		
f	1100110	1111

- V logisimu realizirajte vezje tako, da se ob vpisu znaka na tipkovnici ta prikaže na prikazovalniku (Hex Digit Display). Če vpisan znak ni črka od a do f, naj bo izpisan znak minus (-). Uporabite potrebne gradnike, ki so bili predstavljeni oz. smo jih že poznali. Podrobnosti krmiljenja gradnikov so podane v logisimu.



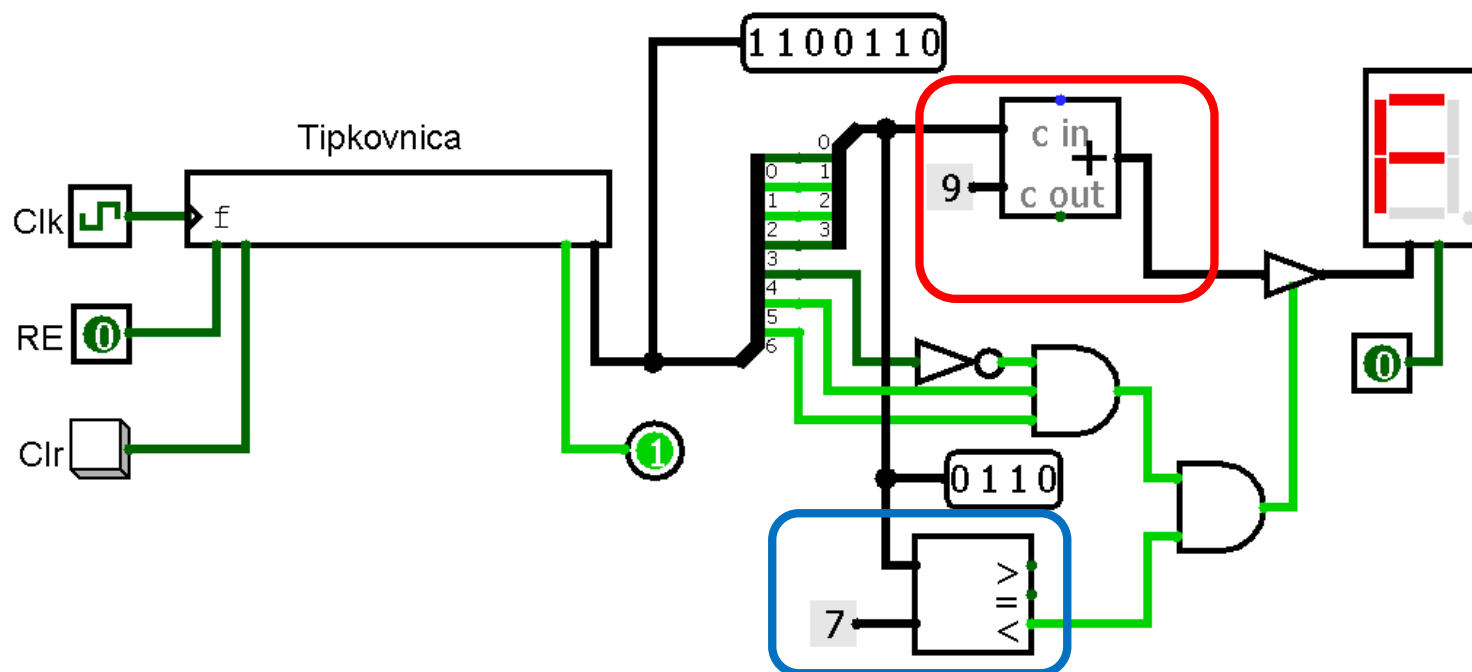
- Vhodi: 7-bitna koda ASCII ($b_6, b_5, b_4, b_3, b_2, b_1, b_0$)
- Izhodi: 4-biti $B=(b_3, b_2, b_1, b_0)$, ki vsebujejo dvojiška števila od 1010 do 1111
- Funkcija: če so $b_6, b_5, b_4 = 110$, se B prišteje 9, prikaže znak (a,...,f), sicer se prikaže (-)

- Logično vezje: ASCII (a,b,c,d,e,f) → dvojiško (1010, 1011, ..., 1111)

Primer:

a: 110 0001 → 0001 + 1001 = 1010

znak a se izpiše, če so zgornji 3 biti kode ASCII 110 in so spodnji 4 biti kode ASCII manjši od 7

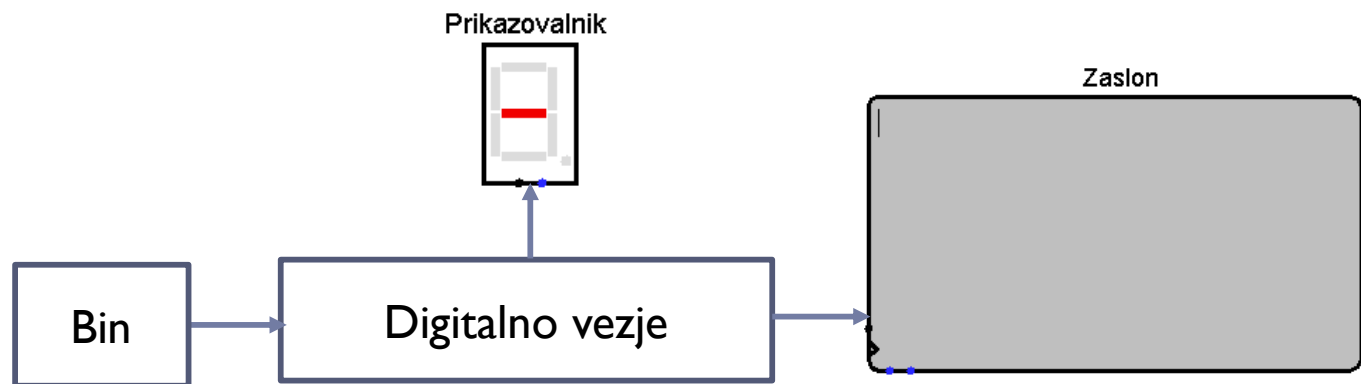


N4: Pretvorba **bin** → **ASCII**

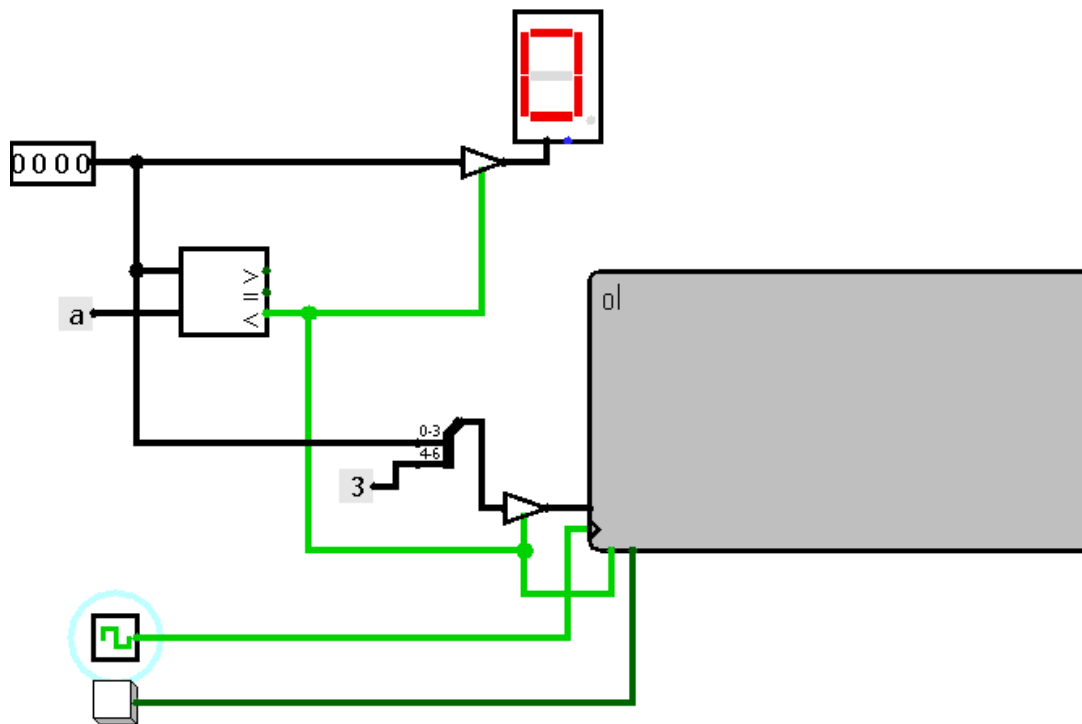
- ❑ Pretvorba 4-bitnega dvojiškega števila v **BCD zapis števila (0, 1, ..., 9)** v 7-bitno kodo ASCII.

Bin (4-biti)	ASCII (7-bitov)	BCD
0000	011 0000	0
0001	011 0001	1
...
1001	011 1001	9

- ❑ V logisimu realizirajte vezje tako, da na tipkovnici vpišemo število, ki se prikaže na prikazovalniku (Hex Digit Display) in na zaslonu (TTY).



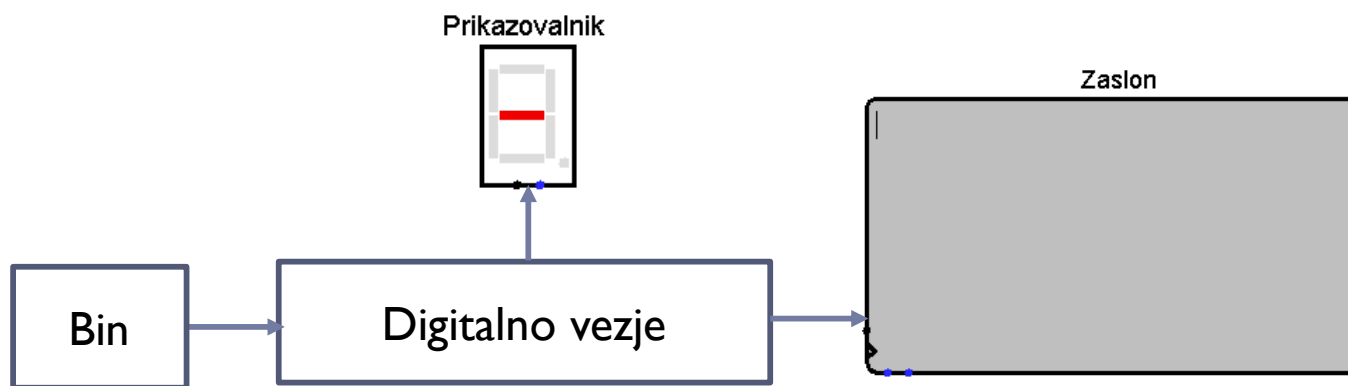
- Logično vezje – Na zaslonu so prikazane cifre BCD, ki smo jih podali kot 4-bitno dvojiško vrednost in pretvorili v ASCII. Če na vhodu zaslona ni cifra, se nič ne izpiše.



- ❑ Pretvorba 4-bitnega dvojiškega števila v **zapis znaka (a,b,c,d,e, f)** v 7-bitno kodo ASCII.

Bin (4-biti)	ASCII (7-bitov)	Bin (4-biti)
1010	1100001	a
1011	1100010	b
1111	1100110	f

- ❑ V logisimu realizirajte vezje tako, da na tipkovnici vpišemo znak, ki se prikaže na prikazovalniku (Hex Digit Display) in na zaslonu (TTY).



- Logično vezje – Na zaslonu so prikazane črka a,...f, ki smo jih podali kot 4-bitno dvojiško vrednost in pretvorili v ASCII. Če na vhodu zaslona ni ena od pričakovanih črk, se nič ne izpiše.

