

Digitalna vezja - Seminar 2

Jernej Habjan 63150106

1) Uvod - Opis problema:

Za nalogo smo morali izdelati kjučavico, ki deluje kot avtomat.

Ključavnica ima uro, gre za sinhronsko vezje. V njo pa lahko vnašamo števila preko tipkovnice.

Naenkrat ključavnica sprejme 4 števila, s katerimi potem določimo akcijo.

Ko so števila vnešena v ključavnico, lahko pritisnemo gumb "Vpiši" ali gumb "Odkleni", ki bosta vpisala novo kombinacijo v ključavnico ali pa jo odklenila.

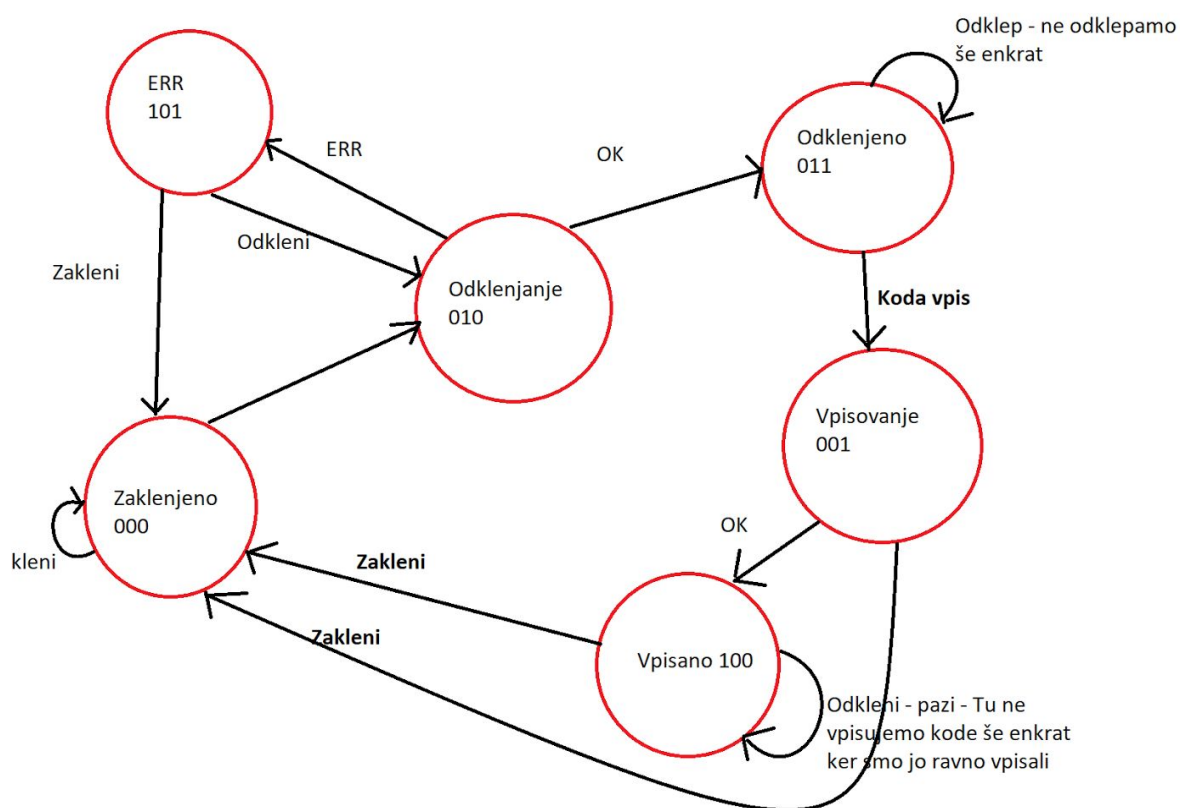
Ni potrebno, da vnesemo vsa 4 števila naenkrat, saj avtomat čaka na nova števila, dokler jih ne bo dobil 4.

- Zaklepanje in Error:
Če med odklepanjem ključavnice vnesemo napačno število, bo ključavnica prešla v Err stanje, prav tako se bo pa zaklenila in pobrisala ostala števila na vhodu. Posvetili se bosta rdeči lučki za Err in Zaklenjenost.
- Zamenjava kode:
Ko zamenjamo kodo ključavnici, se nam hkrati še odklene in se posvetita Rumena lučka, ki je indikator zamenjane kode in Zelena lučka, ki nam pove, da je ključavnica odklenjena.
- Odklepanje:
Pri odklepanju ključavnice, vsak urin signal vzamemo iz tipkovnice levo črko in jo primerjamo z vpisanim številom v pomnilniku (RAM). Če sta številki enaki, preidemo v novo stanje, drugače pa vrnemo Err. Če uspešno vpišemo vsa 4 števila, preidemo v stanje odklepa, kjer posije Zelena lučka.

2) Teoretična rešitev:

Imamo 6 stanj:

- Zaklenjeno
- Vpisovanje
- Odklepanje
- Odklenjeno
- Vpisano
- neveljavno



Slika 1: Načrtovanje prehajanja stanj

Kodirna tabela:

	S2	S1	S0
Zaklenjeno	0	0	0
Vpisovanje	0	0	1
Odklepanje	0	1	0
Odklenjeno	0	1	1
Vpisano	1	0	0
Error	1	0	1

Definicija naslednjega stanja:

Gumbi			Spremenl		Stanje T			Stanje T+1			Obrazložitev
V P I S	O D K L E N I	Z A K L E N I	N A P R E J	ERR	S 2	S1	S0	S2	S1	S0	
0	0	0	0	0	0	0	0	0	0	0	ohranimo
0	0	1	0	0	0	0	0	0	0	0	zaklenemo
0	1	0	0	0	0	0	0	0	1	0	odklenjanje
0	0	0	0	0	0	1	0	0	1	0	ohranimo
0	0	0	0	1	0	1	0	1	0	1	odklenemo
0	0	0	0	1	0	1	0	1	0	1	err
0	0	0	0	0	1	0	1	1	0	1	ohranimo
0	1	0	0	0	1	0	1	0	1	0	Nazaj z err na odklepanje
0	0	1	0	0	1	0	1	0	0	0	zaklenemo
0	0	0	0	0	0	1	1	0	1	1	ohranimo
0	1	0	0	0	0	1	1	0	1	1	Se enkrat odklenemo
1	0	0	0	0	0	1	1	0	0	1	Lahko odklenemo
0	0	1	0	0	0	1	1	0	0	0	zaklenemo
0	0	0	0	0	0	0	1	0	0	1	ohranimo
0	0	0	1	0	0	0	1	1	0	0	Nadaljujemo na vpisano
0	0	0	0	0	1	0	0	1	0	0	ohranimo
0	0	1	0	0	1	0	0	0	0	0	zaklenemo
0	1	0	0	0	1	0	0	1	0	0	Poskusimo odklenit - je že

Enačbe:

Zelo dolge enačbe - poenostavil mi jih je logisim - zaradi $2^{**}8$ kombinacij - 256 kombinacij

Vezje za prižig luči:

S2	S1	S0	Odklenjeno	Vpisano	Err
0	0	0	0	0	0
0	0	1	1	0	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	1	1	0
1	0	1	0	0	1

Enačbe:

Odklenjeno = $\sim S2 * S0 + S2 * \sim S0 = S2 \text{ XOR } S0$

Vpisano = $S2 * \sim S0$

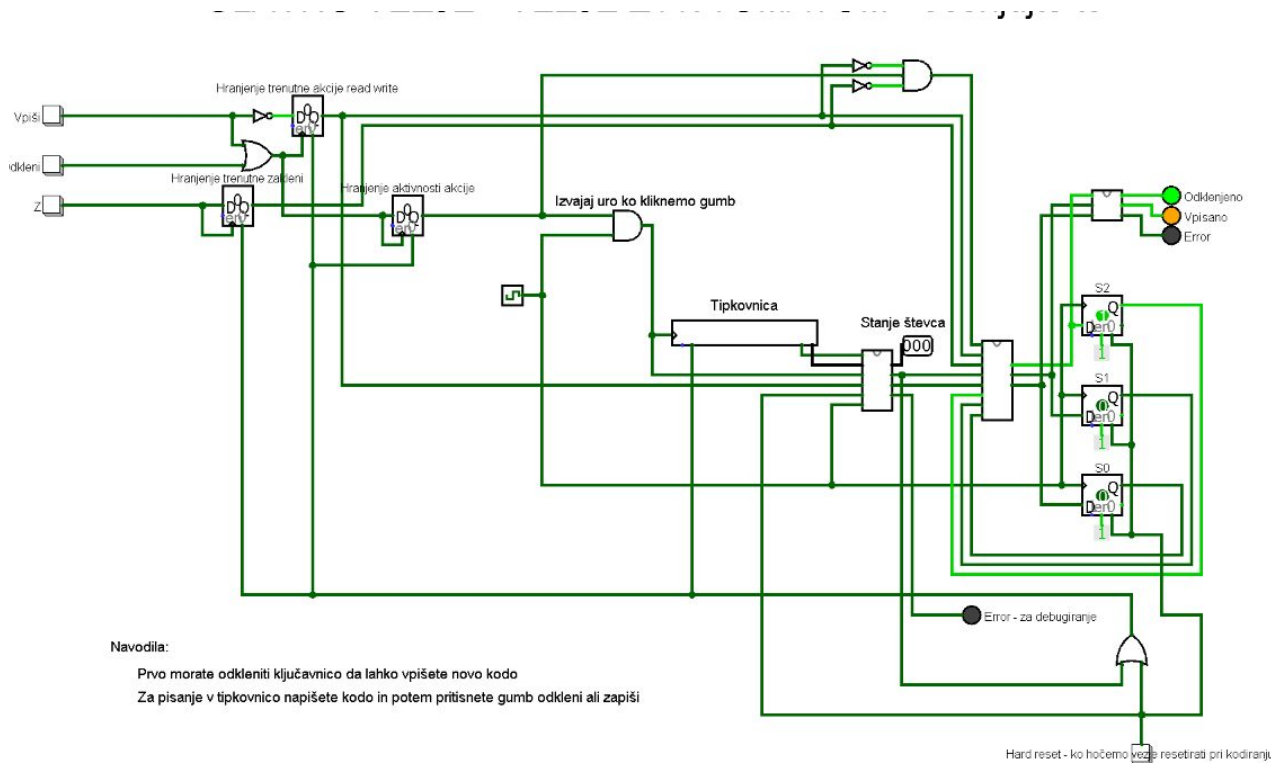
Err = $S2 * S0$

Gradniki:

- **Comparator:**
Za primerjavo dveh števil - če sta števili enaki, vrne komparator == in takrat povečamo števec.
Ko pa vrne < ali >, pa javimo napako. Za javljanje napake pogledamo še če je trenutno kakšen niz v tipkovnici in če izvajamo katero funkcij od vpisov - da ne javljamo napake ko je tipkovnica prazna.
- **Registri:**
Uporabljam registre za pomnjenje trenutnega stanja akcije. Ko kliknemo gumb "Vpiši" ali "Odkleni" se v register shrani ta opcija, ki se uporabi še naslednje 4 cikle, ko primeramo niz ali pa ga vpisujemo.
- **RAM (4x4):**
Ram lahko naslavljamo z 2 biti ($2^2 = 4$ pomnilniške besede), kjer so besede dolge 4 bite. Za branje, na pin Load pripeljem signal 1, za pisanje pa signal 0
- **3 stanjski kontroler:**
Omogoča mi branje in pisanje v RAM
- **Tipkovnica:**
Iz tipkovnice vsako urino periodo (če se izvaja akcija in ima vnešen niz) poberem levo črko in jo pošljem avtomatu. Prav tako ima tipkovnica izhodni pin "Avaliable", ki pove, če ima tipkovnica vpisan kakšen niz, s katerim si pomagam pri urinem signalu. Prav tako iz tipkovnice pride 7 bitov (ASCII), jaz pa potrebujem le 4, zato ostale 3 izpustim.
- **D pomnilna celica:**
Uporabljam jo za beleženje napake pri vnosu niza, saj bi se mi vezje drugače takoj resetiralo, ko bi prišlo do napake, tako da bi vedno ostalo v začetnem stanju.
- **Vrata Not, And, Or, Ura, gumbi, LED diode.**

- Števec:
Uporabljam 5 stanjski števec, saj imamo 4 števila in začetno stanje 0.
Števec se povečuje za 1 vsakič ko mu primerjalnik vrne, da sta števili enaki.
Ob dosegu maksimalnega stanja, števec prižge luč za odklep ali za uspešen vnos kode.

3) Izvedba in rešitev v logisimu:



Slika 2: Rešitev v logisimu - glavno vezje

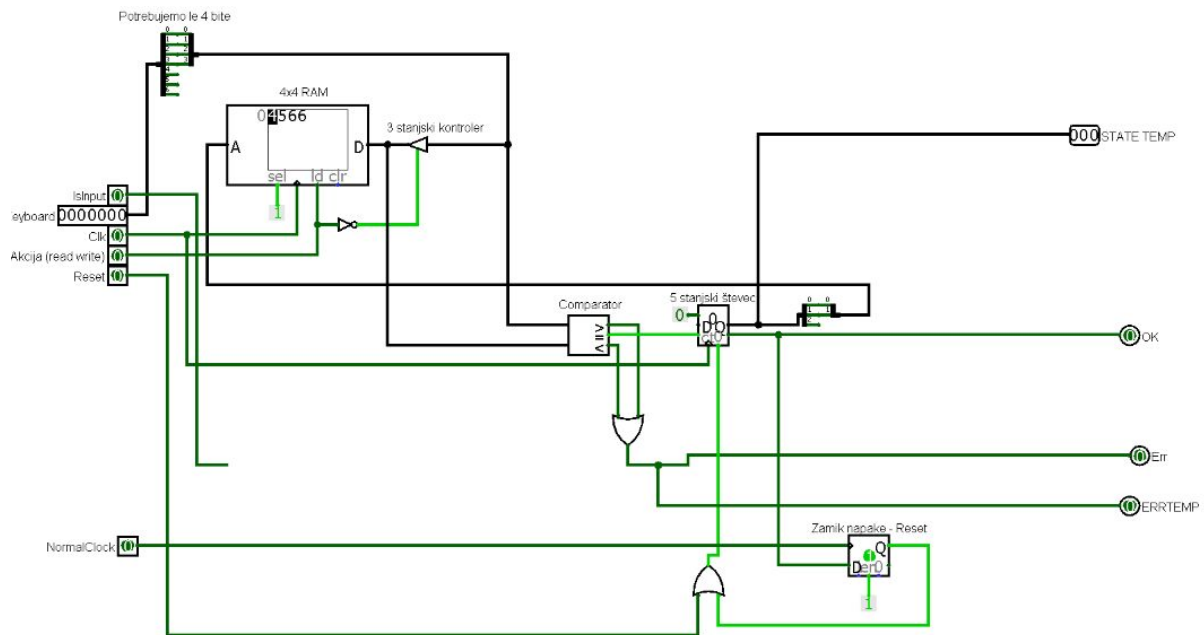
Na shemi so podane labele nad gradniki, ki pojasnijo delovanje gradnika.

Delovanje vezja sem že predstavil zgoraj.

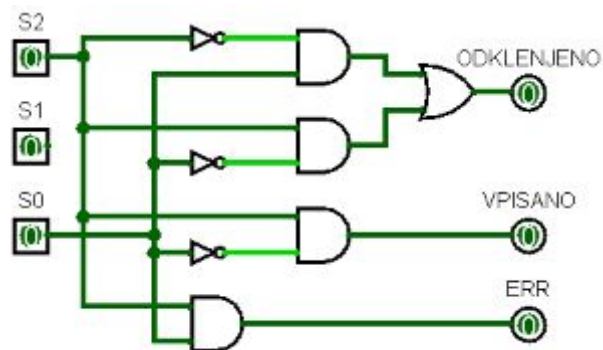
Priložil sem še starejše vezje, ki sem ga naredil prvo preden sem izdelal novo vezje na podlagi avtomata. Prilagam ga zato, ker nočem izbrisati dela.

Glavno vezje vsebuje še 3 module:

Števec, avtomat za prehajanje stanj in vezje za prižiganje luči

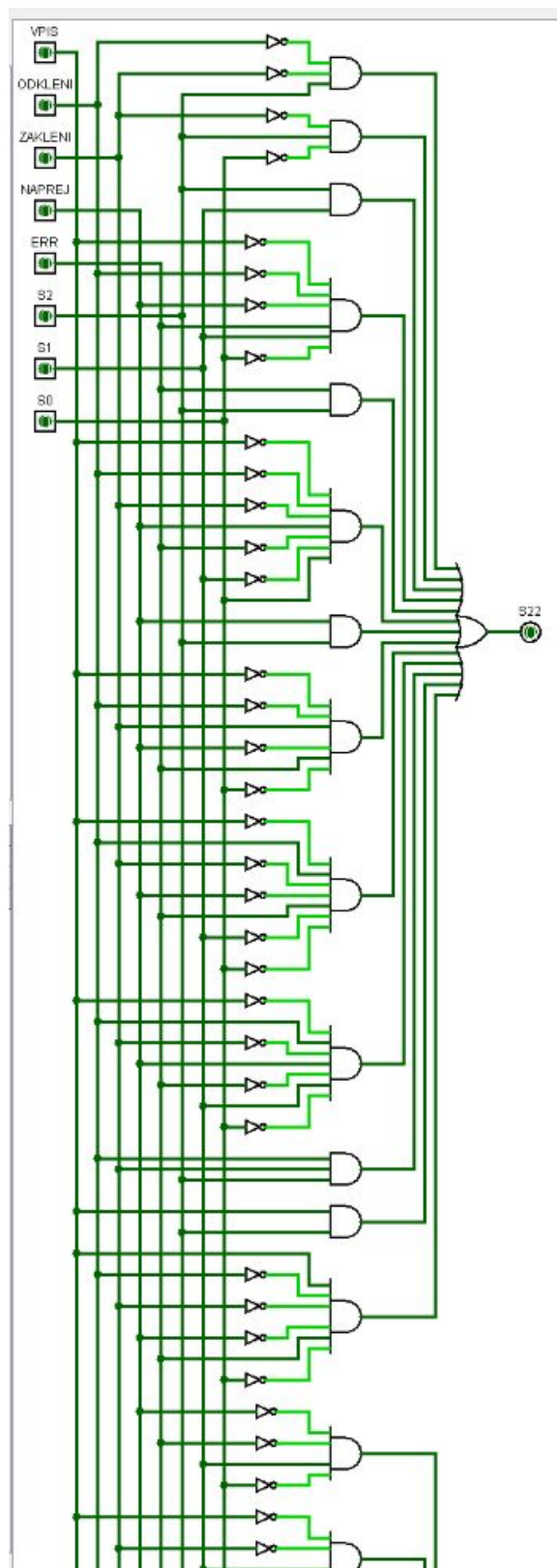


Slika 3 - vezje števca uporaba primerjalnika in števca s petimi stanji - vrne OK, ki pomeni prehod v novo stanje in ERR ki pomeni napako



Slika 4 - Vezja za prižiganje luči - 3 luči zato 3 izhodi

V stanju vpisane kode - ko smo vpisali novo kodo gorita lučki Vpisano in Odklenjeno, ker je takrat ključavnica še vedno odklenjena.



Slika 5 - vezje za definicijo naslednjega stanja (ni cela slika) in trije izhodi (S2,S1,S0)