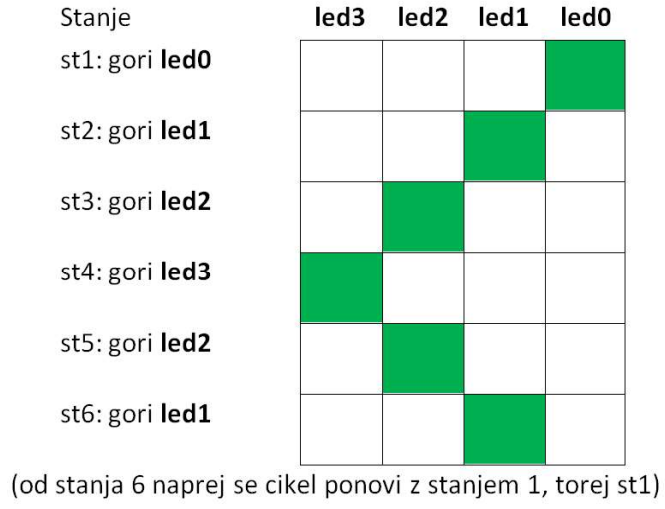
Vaja 6

Potujoča luč

Naredili bomo vezje za učinek potujoče luči, ki se preko štirih LED izmenično pomika levo in desno. Sestavljeno bo iz sekvenčnega stroja in dekodirnika za krmiljenje štirih LED: **led0**, **led1**, **led2** in **led3**.

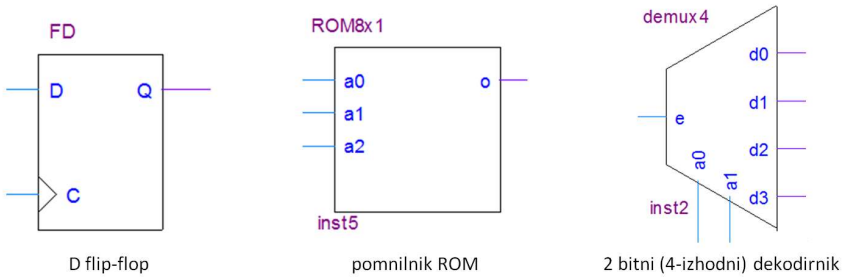


Slika 6.1: *Prikaz delovanja potujoče luči – shema prižiganja LED glede na stanje sekvenčnega vezja.*

6.1 Načrtovanje sekvenčnega vezja

Za zaporedno prižiganje štirih LED diod, kjer se potujoča luč premika v obe smeri, potrebujemo šest stanj. Skupaj z zaporedjem prižiganja diod so prikazana na sliki 6.1. Oglejmo si jih!

Stanje sekvenčnega vezja je shranjeno v flip-flopih. En flip-flop lahko predstavi dve stanji (0 in 1), dva predstavita štiri stanja (00, 01, 10, 11), trije pa osem stanj. Nalogo bomo rešili s končnim strojem stanj, ki shrani tri bite informacije, vendar pa bosta dve stanji od osmih neuporabljeni.



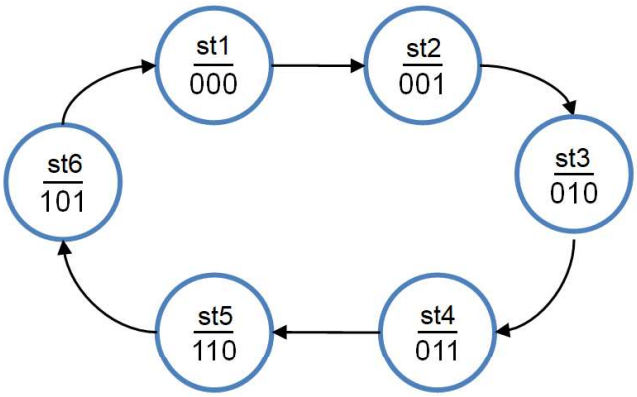
Slika 6.2: *Gradniki vezja za učinek potujoče luči.*

Na shemi vezja bodo poleg D flip-flopov tudi bralni pomnilniki ROM za kombinacijsko logiko sekvenčnega stroja in dekodirnik, kot prikazuje slika 6.2.

6.2 Diagram prehajanja stanj

Delovanje sekvenčnega stroja predstavimo z diagramom prehajanja stanj, ki je prikazan na sliki 6.3. Vsakemu od stanj pripišemo unikaten bitni vzorec, s katerim bo to stanje zapisano (kodirano) v vezju. Bitni vzorci so naprimer zaporedne binarne kombinacije, včasih pa nam bolj ustreza drugačno zaporedje.

V našem primeru je bitni vzorec določen tako, da bomo lahko spodnja dva bita neposredno povezali na dekodirnik za LED. Če ponovno preučimo prikaz delovanja, ugotovimo, da morata imeti stanji **st2** in **st6** enako kombinacijo spodnjih bitov. Podobno velja za stanji **st3** in **st5**.



Slika 6.3: *Diagram prehajanja stanj za sekvenčno vezje potujoče luči.*

Tabela prehajanja stanj

Iz diagrama prehajanja stanj zapišemo tabelo prehajanja stanj, ki določa, katero stanje sledi prejšnjemu. Posamezne bite stanja bomo označili s **s0**, **s1** in **s2**. (primer: za stanje **st5** velja **s0** = 0, **s1** = 1, **s2** = 1). Tabelo 6.1 do konca izpolnite sami, pri tem pa si pomagajte z diagramom prehajanja stanj. Manjkajoče kombinacije iz diagrama naj gredo v stanje 000.

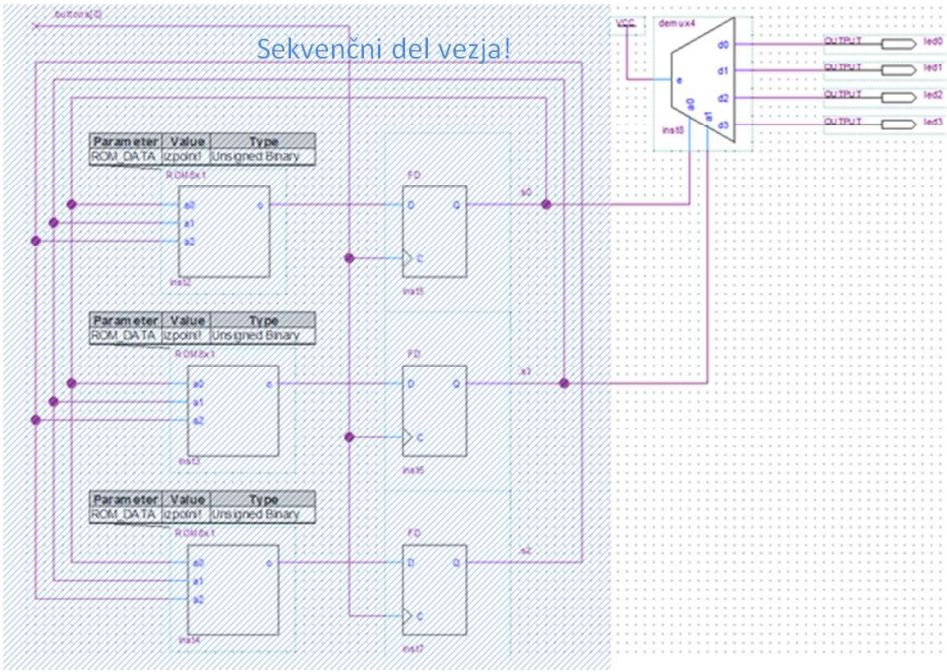
	prejšnje stanje			novo stanje		
	s2	s1	s0	s2	s1	s0
	0	0	0			
	0	0	1			
	0	1	0			
	0	1	1			
X	1	0	0			
	1	0	1			
	1	1	0	**1**	**0**	**1**
X	1	1	1	**0**	**0**	**0**

X označuje neuporabljena stanja!

Tabela 6.1: *Tabela prehajanja stanj. Preostanek tabele izpolnite sami.*

6.3 Shema sekvenčnega vezja

V vezju so trije D flip-flopi z izhodi **s0**, **s1** in **s2**, ki predstavljajo posamezen bit stanja sekvenčnega vezja. Flip-flopi prenašajo vrednost vhoda **D** na izhod **Q** ob vsakem ciklu ure **C**. Stanja flip-flopov preko vhodne logike preslikamo v nova stanja, ki jih peljemo na vhode flip-flopov. To preslikavo izvedemo s pomnilniki ROM v povratni vezavi, kot prikazuje slika 6.4.



Slika 6.4: *Shema vezja potujoče luči. Prehajanje stanj v sekvenčnem vezju določimo z vsebino pomnilnikov ROM!*

Vsebino pomnilnikov ROM določimo glede na tabelo prehajanja stanj. Naslovni signali **s0**, **s1** in **s2** predstavljajo prejšnje stanje, ROM pa na podlagi tega določi novo stanje. Vsak ROM predstavlja en izhodni stolpec v tabeli prehajanja stanj.

Dekodirnik na koncu vezja preslika spodnja dva bita stanj **s0** in **s1** v štiri pozicije luči. Kodiranje stanj smo izbrali tako, da se ta dva bita spreminjata v zaporedju: 00, 01, 10, 11, 10, 01, 00, itd., ki generira želeni učinek.

6.4 Načrt vezja v strojno-opisnem jeziku

Sekvenčno vezje potujoče luči lahko učinkovito opišemo v strojno-opisnem jeziku. Uporabite poenostavljen jezik SHDL in spletno orodje za opis vezja na naslovu: https://lniv.fe.uni-lj.si/shdl/ [4].

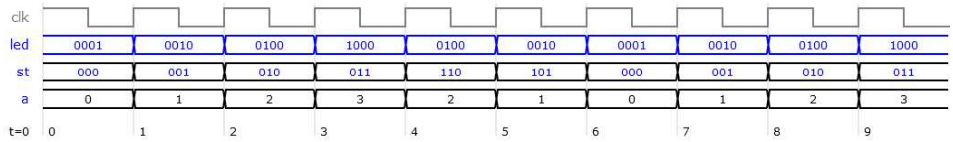
Tabela prehajanja stanj predstavlja navodila za opis sekvenčnega vezja. Definirajmo notranji signal, ki bo shranjeval 3-bitno stanje in zapišimo pogoje za prehod iz enega v drugo stanje:

```
if st="000" then st<="001";
elsif st="001" then st<="010";
...
end
```

Vhod v dekodirnik predstavljata spodnja dva bita registra stanj, ki ju prenesemo v dvobitni notranji signal **a**:

```
a = st(1 downto 0);
```

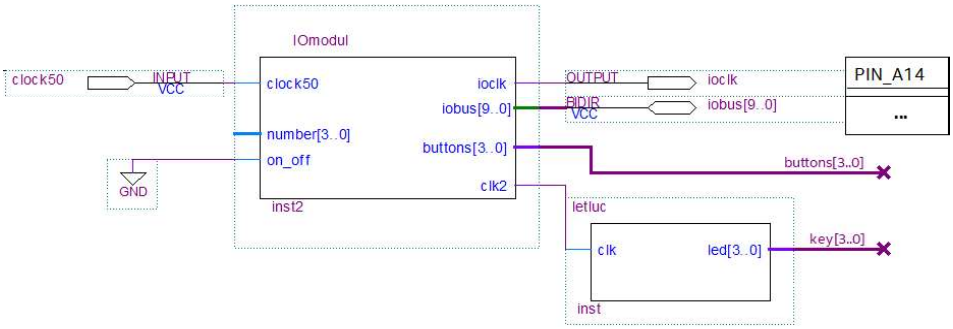
Dodati je potrebno le še opis dekodirnika, ki kombinacije signala **a** preslika v 4-bitni izhod **led**. Dekodirnik je kombinacijsko vezje, zato uporabljamo navaden prireditveni operator =. Slika 6.5 predstavlja simulacijo vezja v orodju SHDL, kjer smo prikazali stanje (st) in izhod (led) v binarni obliki. Način prikaza signala menjamo z desnim klikom na signal.



Slika 6.5: *Simulacija vezja za učinek potujoče luči.*

6.5 Kaj morate narediti vi?

- Izpolnite tabelo prehajanja stanj glede na diagram prehajanja stanj
- Odprite vzorec projekta v orodju **Quartus** in pobrišite simbol RND.
- Narišite shemo vezja pri kateri ne pozabite določiti vsebine ROM, ali pa naredite model vezja v orodju SHDL in prenesite izhodno datoteko v Quartus.
- Vezje pretvorite v nov simbol, ki ga dodajte na glavno shemo in povežite z LED. Uro povežite na počasen signal clk2, da boste lahko opazovali delovanje.
- Prevedite vezje in ga preizkusite na razvojni plošči



Slika 6.6: *Glavna shema s komponento vezja za učinek potujoče luči.*