

## LABORATORIJSKI ZADATAK 3

### Automati sa konačnim brojem stanja

#### Potrebno predznanje

- Urađena pokazna vežba 4
- Teorija automata sa konačnim brojem stanja i njihova primena u digitalnim sistemima

#### Šta će biti naučeno tokom izrade vežbe?

U ovoj vežbi:

- Naučićete da implementirate automat sa konačnim brojem stanja kao digitalni sistem,
- Iskoristićete znanje iz merenja vremena u digitalnim sistemima da promenite vremenski interval između prelaza stanja automata,
- Modifikovaćete datu funkciju prelaza automata,
- Implementiraćete vaš automat sa konačnim brojem stanja na E2LP platformi.

#### Apstrakt i motivacija

U ovoj vežbi primenićemo teoriju automata sa konačnim brojem stanja u realizaciji realnog digitalnog sistema koji kontroliše svetleće znakove na automobilima. Sistem će biti zasnovan na automatu koji kontroliše vrednost na LED diodama i simulira automobilski žmigavac – tri diode će označavati skretanje u levo i tri diode skretanje u desno. Poseban ulaz stanja opasnosti, koji treba biti aktiviran kada se automobil nalazi u opasnosti, dovešće do naizmeničnog paljenja i gašenja svih 6 dioda. Tokom izrade vežbe, osim implementacije već definisanog automata, vi ćete i modifikovati sam automat modifikujući mu funkciju prelaza kako bi se ostvario željeni cilj rada sistema.

#### Šta treba doneti na termin laboratorijske vežbe?

- Logičku i/ili blok šemu sistema na papiru ili računaru (za **krajnji izgled sistema**). *Koristite standardne kombinacione i sekvencijalne mreže kao komponente, nemojte crtati blokove koji su suviše apstraktni i nepotpuno definisani.*
- VHDL opis **krajnjeg koraka** u realizaciji sistema (međukoraci nisu potrebni) – krajnji korak je poslednji korak koji uspete da realizujete.
- Testbench za krajnji korak sistema.
- Generisanu *.bit* datoteku za konfigurisanje E2LP platforme za krajnji korak sistema.

## ZADACI

### 1. Žmigavac

Za početak, projektovaćemo automat koji kontroliše žmigavce na automobilu. Neka svaka strana ima po 3 žmigavca koji se pale redom, od unutrašnjeg do spoljašnjeg, kada automobil daje znak skretanja na neku stranu.

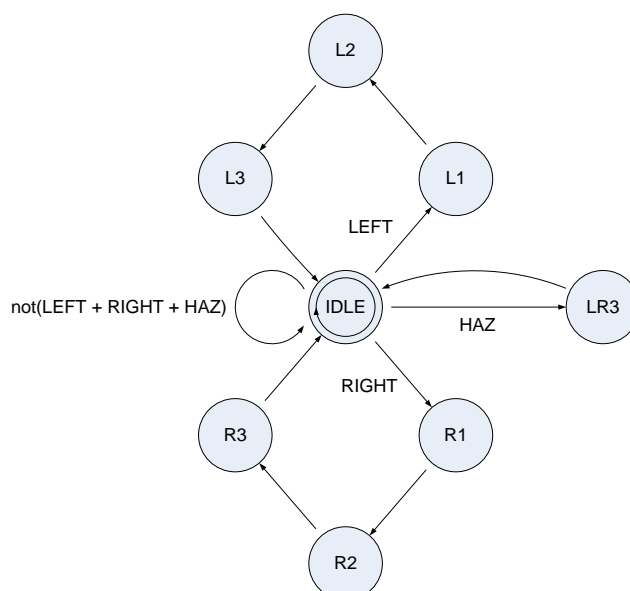
Kontrolni automat prolazi kroz 8 stanja: IDLE, L1, L2, L3, R1, R2, R3, LR3.

Početno stanje je IDLE.

Ulazi automata su:

- inLEFT – jednobitni ulaz, pritisnut zahtev za levi žmigavac,
- inRIGHT – jednobitni ulaz, pritisnut zahtev za desni žmigavac,
- inHAZ – jednobitni ulaz, pritisnut signal za opasnost / nepredviđenu situaciju.

Funkcija prelaza automata je data na Slici 1-1.



Slika 1-1. Graf prelaza stanja automata

U ovom delu zadatka, neka se prelazi između stanja automata dešavaju svakog takta. Ovako rešen zadatak nije moguće proveriti na E2LP platformi zbog brzine promene stanja, pa preporučujemo proveru pomoću simulacije.

Iz početnog stanja, IDLE, prelazi su sledeći:

- ukoliko je pritisnut taster inLEFT, automat prelazi u stanje L1,
- ukoliko je pritisnut taster inRIGHT, automat prelazi u stanje R1,
- ukoliko je pritisnut taster inHAZ, automat prelazi u stanje LR3,
- u suprotnom automat ostaje u stanju IDLE.

Svi prelazi prikazani u grafu prelaza stanja sa strelicom bez pratećeg teksta su *bezuslovni* prelazi, odn. automat iz trenutnog prelazi u sledeće stanje uvek, bez obzira na vrednost ulaza.

Tabela 1-1 prikazuje funkciju izlaza automata. Izlaze realizovati **kao kombinacionu funkciju** trenutnog stanja.

**Tabela 1-1. Funkcija izlaza automata**

State	L[2]	L[1]	L[0]	R[2]	R[1]	R[0]
IDLE	0	0	0	0	0	0
L1	0	0	1	0	0	0
L2	0	1	1	0	0	0
L3	1	1	1	0	0	0
R1	0	0	0	1	0	0
R2	0	0	0	1	1	0
R3	0	0	0	1	1	1
LR3	1	1	1	1	1	1

Test bench za ovaj korak sistema treba da proveriti sve prelaze stanja.

*Napomena: ulazi inLEFT, inRIGHT i inHAZ su aktivni na vrednosti 0.*

## 2. Duži vremenski interval između prelaska stanja

Kako bi verifikovali rad sistema na E2LP platformi, prelazi između stanja moraju biti mnogo sporiji od jednog takta koliko sada traje vreme između prelaza. Promenićemo sistem tako da prelazi između stanja traju *pola sekunde*. Kako bi ovo ostvarili, dodaćemo štopericu u naš sistem koja će na svakih pola sekundi javljati automatu da može da promeni stanje pomoću *terminal count* signala, na sličan način na koji smo povećavali brojač sekunde u prethodnoj vežbi. *Terminal count* signal štoperice treba da se iskoristi kao signal dozvole prelaza kod automata. (Nemojte koristiti *terminal count* kao takt automata jer onda pravite više taktnih domena u sistemu što zahteva sinhronizacione korake sa kojima se još nismo upoznali. Čitav sistem treba da radi na istom taktu, a *terminal count* treba iskoristiti kao signal dozvole u ostatku sistema.)

Prolaze sistema povezati sa E2LP platformom prema Tabeli 2-1.

**Tabela 2-1. Povezivanje prolaza na komponente E2LP platforme**

Prolaz	Smer	Komponenta na E2LP platformi
iCLK	in	CLK
inRST	in	RESET
inLEFT	in	JOY1
inRIGHT	in	JOY3
inHAZ	in	JOY2
oLEFT [2:0]	out	LED7 – LED5
oRIGHT [2:0]	out	LED2 – LED0

### 3. Prioritet ulaza

U trenutnoj realizaciji automata, u stanju IDLE, sledeće stanje je definisano ukoliko se pritisne samo jedan od ulaza inLEFT, inRIGHT ili inHAZ. Ukoliko je pritisnuto više tastera, ponašanje nije definisano, odn. automat može da se ponaša na bilo koji način. *U suštini, prioritet ulaza je definisan redosledom kojim ste proveravali ulaze u vašim uslovnim proverama.* U ovom zadatku ćemo ovo promeniti i potpuno definisati ponašanje automata pri svakoj kombinaciji ulaza.

Promeniti funkciju prelaza automata tako da, u stanju IDLE, prioritet ulaza inLEFT, inRIGHT i inHAZ bude sledeći:

- inHAZ je najvećeg prioriteta, odn. kada je pritisnut inHAZ ulaz, prelazi se u stanje LR3 bez obzira na vrednost ulaza inLEFT i inRIGHT,
- Ako inHAZ nije pritisnut:
  - u L1 stanje se prelazi ako je pritisnut inLEFT, a nije pritisnut inRIGHT,
  - u R1 stanje se prelazi ako je pritisnut inRIGHT, a nije pritisnut inLEFT,
  - ako su pritisnuti i inLEFT i inRIGHT, ulazi se u stanje LR3.

Ovaj korak zadatka utiče samo na prelaze iz stanja IDLE. Ostala stanja ne treba menjati.

## OPŠTE NAPOMENE

Prilikom crtanja šeme na papiru ili računaru, koristiti blokove za standardne kombinacione i sekvencijalne mreže – nema potrebe da poznate komponente crtate na nivou logičkih kola. Bitno je da logička šema ispravno opiše logiku sistema. Sve nestandardne komponente koje koristite morate na neki način definisati – bilo istinitosnom tablicom ili opisom na nivou logičke funkcije.

Implementaciju sistema izvršiti za poslednji urađen korak zadatka. Nije neophodno imati urađenu implementaciju za svaki korak. Simulaciju treba raditi posle svakog koraka, jer na sledeći korak treba preći jedino ukoliko je prethodni funkcionalno proveren.

Na termin vežbe doneti **jedan** VHDL opis i **jedan** testbench (nije neophodno donositi ni opis ni testbench međukoraka). Međukoraci su tu da bi vama olakšali put ka kompletnom sistemu i omogućili da vaš sistem projektujete inkrementalno – počev od jednostavnijih komponentata ka složenijim.

Broj poena koje možete osvojiti na vežbama zavisi od koraka do kojeg ste doveli sistem – svaki korak nosi 1 poen (do maksimalno 3), a viši poeni (4, 5) se mogu dobiti izradom modifikacija i malih dopuna vašeg sistema koje ćete dobiti na terminu vežbe. Dodatni zadatak na terminu vežbe vam može doneti i dodatni, šesti poen.

## ZAKLJUČAK

U ovoj vežbi ste primenili vaše znanje iz teorije automata sa konačnim brojem stanja i napravili sistem zasnovan na njima koji vrši jednu funkciju sa kojom se svakodnevno srećemo i koristimo je. Primećujete sada da automati nisu puki bespotrebni matematički koncept, nego veoma korisan alat koji vam omogućuje da napravite sistem koji prolazeći kroz stanja vrši različite operacije u različitim stanjima. Automati su veoma važan deo većine digitalnih sistema i imaju široku primenu, kao što ćete i primetiti u nastavku ovog predmeta.