

LABORATORIJSKI ZADATAK 4

Automati sa konačnim brojem stanja

Potrebno predznanje

- Urađena pokazna vežba 6
- 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

Kao što je napomenuto u prethodnoj vežbi, veoma često postoji potreba da digitalni sistem izvršava neku sekvencu unapred definisanih zadataka. Jedan tipičan primer takvog sistema jeste semafor. U ovoj vežbi primenićemo teoriju automata sa konačnim brojem stanja u realizaciji realnog digitalnog sistema koji modeluje semafor gde ćemo umesto crvenog, žutog i zelenog svetla, koristiti LED diode na E2LP ploči. Uz pomoć jednog tastera ćemo omogućiti normalan režim rada semafora sa poznatom sekvencom stanja, a uz pomoć drugog ćemo modelovati režim kada je semafor u kvaru i kada dolazi do naizmeničnog paljenja i gašenja LED dioda.

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. Semafor

Za početak, projektovaćemo automat koji kontroliše rad jednog jednostavnog semafora. Svetla semafora ćemo predstaviti sa parovima dioda – 2 diode na bitima najveće težine će predstavljati crveno svetlo, 2 diode će predstavljati žuto i 2 zeleno svetlo.

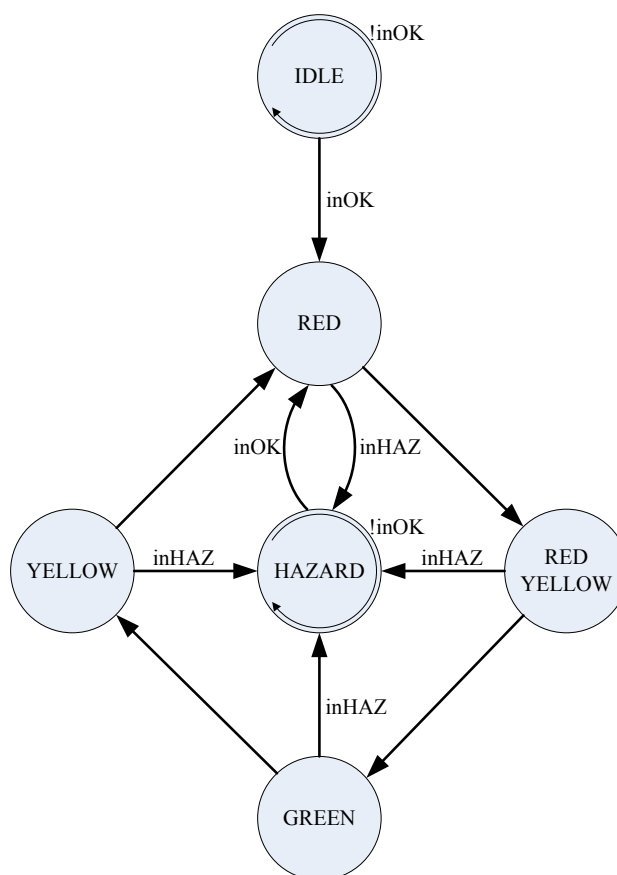
Automat prolazi kroz 6 stanja: IDLE, RED, RED_YELLOW, GREEN, YELLOW i HAZARD.

Početno stanje automata je IDLE.

Ulazi automata su:

- inOK – jednobitni ulaz pokreće sekvencu stanja ukoliko je semafor u stanju IDLE i nastavlja sa normalnim radom semafora ukoliko je bio u stanju HAZARD
- inHAZ – jednobitni ulaz koji iz bilo kog radnog stanja prebacuje semafor u stanje HAZARD.

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.

Funkcija prelaza je sledeća:

- Iz početnog stanja IDLE automat prelazi u stanje RED kada se pritisne taster inOK. Dok se ne pritisne inOK, automat ostaje u stanju IDLE.
- Kada pređe u stanje RED, bezuslovno prelazi u stanje RED_YELLOW, potom u stanje GREEN, zatim u stanje YELLOW i na kraju se vraća u stanje RED i tako nastavlja u krug.
- Iz bilo kog od ova pomenuta 4 stanja automat može preći u stanje HAZARD ukoliko je pritisnut taster in HAZ. Kada automat pređe u stanje HAZARD, on ostaje u tom stanju sve dok taster inOK ne bude pritisnut, nakon čega prelazi u stanje RED i započinje novi ciklus.

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 (sa izuzetkom, naravno inRST signala koji uvek automat prebacuje u stanje IDLE).

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

Tabela 1-1. Funkcija izlaza automata

State	R[1]	R[0]	Y[1]	Y[0]	G[1]	G[0]
IDLE	0	0	0	0	0	0
RED	1	1	0	0	0	0
RED_YELLOW	1	1	1	1	0	0
GREEN	0	0	0	0	1	1
YELLOW	0	0	1	1	0	0
HAZARD	1	1	1	1	1	1

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

Napomena: ulazi inOK 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 sinhronizaciju 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
inOK	in	JOY1
inHAZ	in	JOY2
oRED [1:0]	out	LED7-LED6
oYELLOW [1:0]	out	LED4-LED3
oGREEN [1:0]	out	LED1-LED0

3. Naizmenično paljenje i gašenje LED dioda u stanju HAZARD

Kako bi stanje HAZARD bilo malo realnije modelovano, uvešćemo naizmenično paljenje i gašenje dioda u ovom stanju. Obično kada dođe do kvara semafora prikazano je samo treptajuće žuto svetlo (ili ne svetli ništa) umesto da bude upaljeno istovremeno i crveno i žuto i zeleno svetlo. Kada se automat nađe u ovom stanju, *pola sekunde* treba da budu upaljene samo diode LED4 i LED3 a zatim drugu *polovinu sekunde* ne treba da svetli ni jedna diode. Ovaj proces treba ponavljati sve dok se taster inOK ne bude pritisnut.

Postoji nekoliko načina da realizujete ovu funkcionalnost. Jedno rešenje je da ubacite novo pomoćno stanje i odgovarajući izlaz za njega. Primer drugog rešenja bi bio ne menjate broj stanja u automatu već da ubacite dodatnu kombinacionu logiku u funkciji izlaza za stanje HAZARD. Ukoliko Vam padne na pamet nešto treće, slobodno implementirajte na taj način.

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 komponenata ka složenijim.

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 koristan 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.