



**FER**

SVEUČILIŠTE U ZAGREBU  
F A K U L T E T  
ELEKTROTEHNIKE  
I RAČUNARSTVA

## FORMALNE METODE U OBLIKOVANJU SUSTAVA

### 1. DOMAĆA ZADAĆA

ZAGREB, 2009.

## **TRAFFIC LIGHT CONTROLLER (upoznavanje sa Vis sustavom)**

Potrebno je proučiti dokumentaciju za VIS sustav (vis\_user.pdf) i nakon toga napraviti slijedeće:

1. Proučiti datoteku tlc.v u kojoj se nalazi Verilog primjer Traffic Light Controller-a (TLC) opisanog u vis\_user.pdf.
2. Proučiti prikaz konačnih automata na slici 2.2 (Figure 2.2 u vis\_user.pdf) i prikazane automate presložiti kao Mealyjeve i Moorove konačne automate (Neki od modula su Mealyjevi, a neki Mooreovi automati što je potrebno prepoznati). Potrebno je analizirati verilog primjer TLC-a, i onda iznova izgraditi automate (treba ispraviti pogreške u automatima iz vis\_user.pdf).

- Izgraditi tablicu u kojoj je opisano međudjelovanje modula. Potrebno je opisati koji moduli utječu na rad ostalih modula i kojim varijablama. Neka reci i stupci označavaju module, a element polja tablice neka budu varijable kojima moduli (iz redaka) utječu na druge module (iz stupaca) kao što je prikazano:

	Modul 1	Modul 2
Modul 1		Req
Modul 2	Ack	

{Modul 1 postavlja zahtjev pomoću varijable Req i upućuje ga na Modul 2. Modul 2 poduzima akciju obrade zahtjeva i varijablom Ack utječe na Modul 1}

	Timer	Sensor	Hwy_control	Farm_control
Timer				
Sensor				
Hwy_control				
Farm_control				

- Pomoću programa vl2mv potrebno je pretvoriti Verilog opis TLC-a (tlc.v) u BLIF-MV opis (tlc.mv) i ukratko objasniti što je BLIF-MV i što se s takvim opisom može prikazati:
- Pokrenuti VIS i upisati «help» te proučiti koje su komande dostupne te pročitati njihovo objašnjenje u vis\_user.pdf
- Učitati BLIF-MV opis TLC-a u VIS pomoću naredbe «read\_blif\_mv» (ili alias-a «rlmv», popis svih alias-a može se dobiti pomoću komande «alias»)
- Isprobati komande za šetanje kroz i prikaz hijerarhije modela («ls», «cd», «pwd»)
- Isprobati komandu «print\_models» (alias «pm») i prikazati te komentirati rezultate:
- Isprobati komandu «print\_hierarchy\_stats» (alias «phs») i prikazati te komentirati rezultate:

10. Unutar svih modula izvesti «flatten\_hierarchy» naredbu i pogledati statistiku svakog modula zasebno «print\_network\_statistics». Objasniti što se dešava prilikom «flatten\_hierarchy» i komentirati dobivene rezultate:

### **BUS ARBITER (provjera modela)**

Cilj ovog dijela domaće zadaće je provjera modela arbitraže sabirnice (koji je objašnjen na predavanju) pomoću CTL formula. Sintaksa CTL logike koju prihvaća VIS sustav objašnjena je u vis\_user.pdf i vis-ctl-pdf. Za provjeru svojstva sustava u CTL notaciji potrebno je nakon učitavanja primjera u sustav pokrenuti komandu init (koja obuhvaća sve radnje potrebne za izgradnju stroja s konačnim brojem stanja nad kojim se vrši provjera CTL formula) i nakon toga komandu model\_check uz koju se upiše ime datoteke u kojoj se nalaze CTL formule.

Potrebno je napraviti slijedeće:

1. Proučiti verilog opis iz datoteke ex1.v (u datoteci arbiter.pdf nalazi se hijerarhijski prikaz sustava modeliranog u ex1.v) i za sve module nacrtati FSM i tablicu u kojoj je prikazano međudjelovanje modula kao što je navedeno u prethodnoj vježbi.

	Arbiter	Controller	Client
Arbiter			
Controller			
Client			

2. Upiši primjer ex1.v u VIS sustav.
3. Specificiraj i napiši u CTL notaciji obilježje ***sigurnosti***:  
"Dva procesa ne mogu dobiti (upravljati) sabirnicom u isto vrijeme"
  
4. Provjeri i eventualno ispravi primjer ex1.v (iskoristi "debug trace") kako bi zadovoljio obilježje sigurnosti.
5. Specificiraj i napiši u CTL notaciji obilježje ***stalnog zahtjeva***:  
"Ako klijent A traži sabirnicu, on neće poništiti svoj zahtjev sve dok ne dobije potvrdu zahtjeva".
  
6. Provjeri i eventualno ispravi primjer ex1.v (iskoristi "debug trace") kako bi zadovoljio obilježje stalnog zahtjeva.
7. Specificiraj i napiši u CTL notaciji obilježje ***jednakih šansi***:  
"Ako A drži sabirnicu, a B ju zahtijeva, tada nije moguće da A otpusti sabirnicu i dobije ju natrag prije nego B upravlja sabirnicom".
  
8. Provjeri i eventualno ispravi primjer ex1.v (iskoristi "debug trace") kako bi zadovoljio obilježje jednakih šansi.
9. Specificiraj i napiši u CTL notaciji obilježje ***životnosti*** (engl. liveness):  
"Uvijek vrijedi: ako klijent A zahtijeva sabirnicu, konačno će je i dobiti".
  
10. Provjeri ovo obilježje i komentiraj rezultat.
  
11. Ako ne zadovoljava, predloži u CTL notaciji ograničenja klijentima kako bi zadovoljili obilježje. Upiši ograničenja u datoteku ex3.fair, te provjeri obilježje životnosti. Navedi upisana ograničenja i komentiraj rezultate: