

ZAVOD ZA ELEKTRONIKU, MIKROELEKTRONIKU, RAČUNALNE I INTELIGENTE SUSTAVE
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA
SVEUČILIŠTE U ZAGREBU

DIGITALNA LOGIKA
UPUTA ZA ŠESTU LABORATORIJSKU VJEŽBU
dr.sc. Marko Čupić

Zagreb, 2013.

6. laboratorijska vježba – Sekvencijski sklopovi II

1. Uvod

Za projektiranje sklopova, ili općenito bilo kojih sustava koji trebaju implementirati složena ponašanja, izuzetno pomoćno sredstvo su konačni automati i pripadajuća teorija. U sklopu predmeta Digitalna logika obrađuju se dvije vrste automata: Mealy-jev i Moore-ov, te se ti automati koriste za projektiranje složenijih digitalnih sklopova. U općem slučaju, prilikom projektiranja nekog digitalnog sklopa kreće se od funkcionalnosti koju treba postići. Ta funkcionalnost prvo se opisuje odgovarajućim konačnim automatom, a potom se taj automat preslikava u željeni sklop. Namjena ove vježbe jest detaljnije upoznavanje s postupcima projektiranja složenijih digitalnih sustava, te s nekim od načina opisivanja automata u VHDL-u.

Prije dolaska na laboratorijsku vježbu preporuča se proučiti gradivo vezano uz projektiranje sekvencijskih sklopova u [1,2]; posebice zadatke 11.1 – 11.3 te 11.19 – 11.22 u [2].

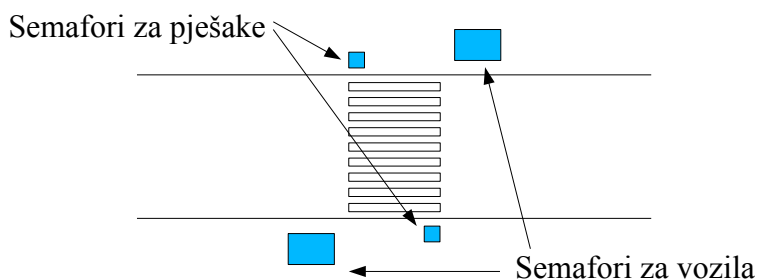
2. Zadatak

U okviru laboratorijske vježbe potrebno je upisati i provjeriti sljedeće modele:

- model komponente BROJILLO: 5-bitno sinkrono binarno brojilo sa sinkronim ulazom za reset, čijim djelovanjem brojilo prelazi u stanje 1,
- model komponente TIMER,
- model komponente AUTOMAT te
- model komponente UPRAVLJAČ SEMAFORA.

Za svaki od napisanih modela podrazumijeva se da će isti prije uporabe u nekom drugom modelu biti testirani u odgovarajućem ispitnom modelu. Detalji i opisi prethodno navedenih sklopova slijede u nastavku ovog dokumenta. Studenti čiji JMBAG završava parnom znamenkom sklop automat u VHDL-u modeliraju kao Mooreov stroj s konačnim brojem stanja; studenti čiji JMBAG završava neparnom znamenkom sklop automat u VHDL-u modeliraju kao Mealyjev stroj s konačnim brojem stanja. Neovisno o JMBAG-u, svi studenti u nastavku ove upute trebaju nacrtati slike obje verzije automata u za to predviđena mjesta.

Tijekom ove laboratorijske vježbe potrebno je projektirati i opisati u VHDL-u jednostavan upravljač semafora. U nastavku teksta opisana je funkcionalnost koju je potrebno postići konačnim sklopom, a detalji realizacije dani su u sljedećem poglavlju.

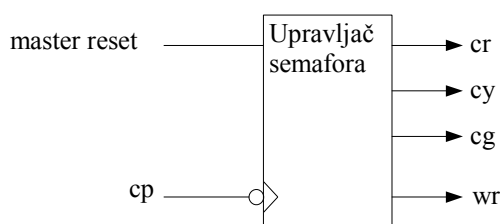


Slika 1. Tlocrt raskršća sa semaforima

Raspored semafora koje treba nadzirati sklop prikazan je na slici 1, dok su u tablici 1 prikazani ciklusi koje semafori trebaju prolaziti. Na slici 2 prikazano je sučelje konačnog sklopa, a u tablici 2 definirana su značenja pojedinih ulaza i izlaza sklopa. Svi izlazni signali aktivni su u visokom stanju, tj. kada im je vrijednost 1, a inače su neaktivni. Sklop sve operacije obavlja na padajući brid signala takta.

Stanje	Trajanje stanja
Svi semafori pokazuju crveno (inicijalno stanje)	8T
Automobili dobijaju crveno + žuto	2T
Automobili dobijaju zeleno	32T
Automobili dobijaju žuto	4T
Automobili dobijaju crveno	2T
Pješaci dobijaju zeleno	16T
Pješaci dobijaju crveno	4T

Tablica 1. Ciklusi semafora



Slika 2. Sučelje sklopa za upravljanje semaforima

Prvo stanje navedeno u tablici 1 je inicijalno stanje i upravljač semafora više se nikada ne vraća u njega već kruži po ostalim stanjima. Trajanje pojedinih ciklusa semafora zadano je u parametriziranom obliku. To olakšava ispitivanje sklopa, a također omogućava relativno jednostavnu promjenu svih ciklusa promjenom samo jednog parametra. *Master reset* pri tome predstavlja ulaz za početno resetiranje sklopa; ovaj ulaz treba biti aktivan na samom početku rada sklopa u trajanju od jednog takta CP-a, i nakon toga stalno mora biti neaktivan. Djelovanje ulaza *Master reset* nemojte prikazivati niti na dijagramu promjene stanja automata: u obzir ga uzmite samo u VHDL-u koji predstavlja sekvencijski blok (odnosno registar stanja) u modelu automata.

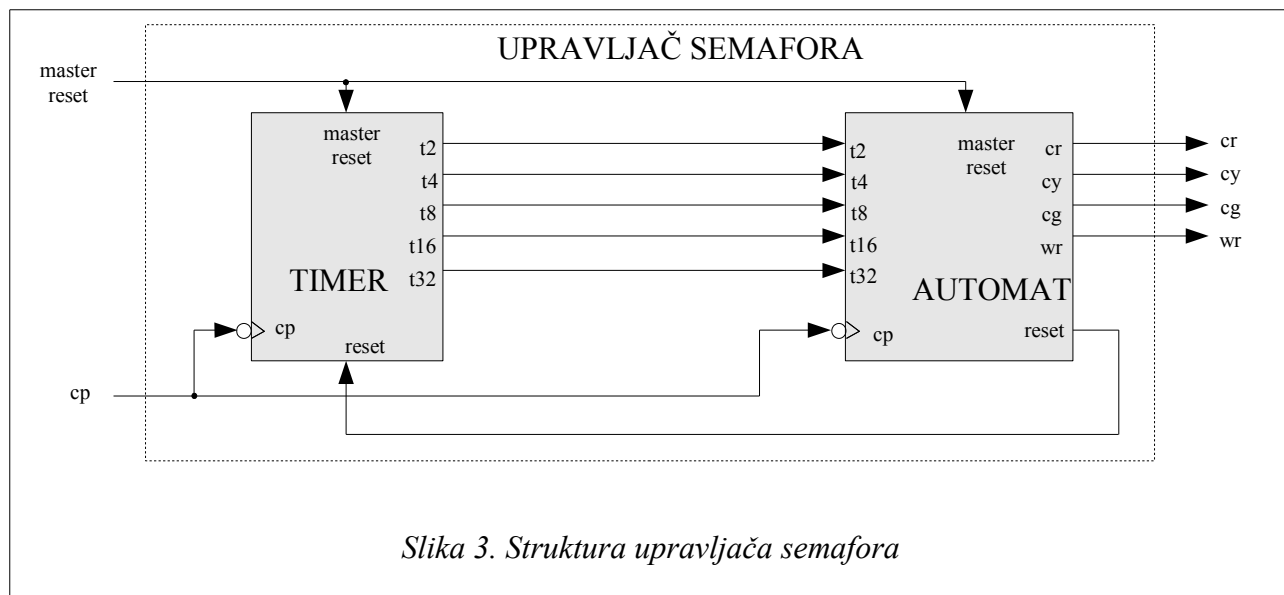
Ulaz/izlaz	Značenje
master reset	Ulaz. Aktiviranjem ovog signala semafor prelazi u početno stanje. Signal djeluje sinkrono.
cp	Ulaz. Dovodi se signal takta.
cr	Izlaz. Kad je ovaj signal aktivan tada je na semaforu za automobile uključeno crveno.
cy	Izlaz. Kad je ovaj signal aktivan tada je na semaforu za automobile uključeno žuto.
cg	Izlaz. Kad je ovaj signal aktivan tada je na semaforu za automobile uključeno zeleno.
wr	Izlaz. Kad je ovaj signal neaktivan na semaforu za pješake prikazano je crveno, a u suprotnom prikazano je zeleno.

Tablica 2. Značenje pojedinih ulaza i izlaza sklopa

3. Uputa

Ova vježba može se napraviti na čitav niz različitih načina, a u nastavku je opisana samo jedna moguća varijanta, koje se treba pridržavati. Prvi korak u rješavanju zadanog zadatka jest skica strukture rješenja. Budući da je opisan problem relativno složen u odnosu na dosadašnje zadatke, za njegovo rješavanje iskoristit ćemo znanje o strojevima s konačnim brojem stanja. Upravljač semafora razložit ćemo u dva dijela: timer (tj. vremenski sklop) koji će na svojim izlazima generirati informacije o nastupanju vremena $2T$, $4T$, $8T$, $16T$ i $32T$, te automat koji će na temelju tih informacija upravljati timerom i svjetlima semafora. Skica ove dekompozicije prikazana je u nastavku.

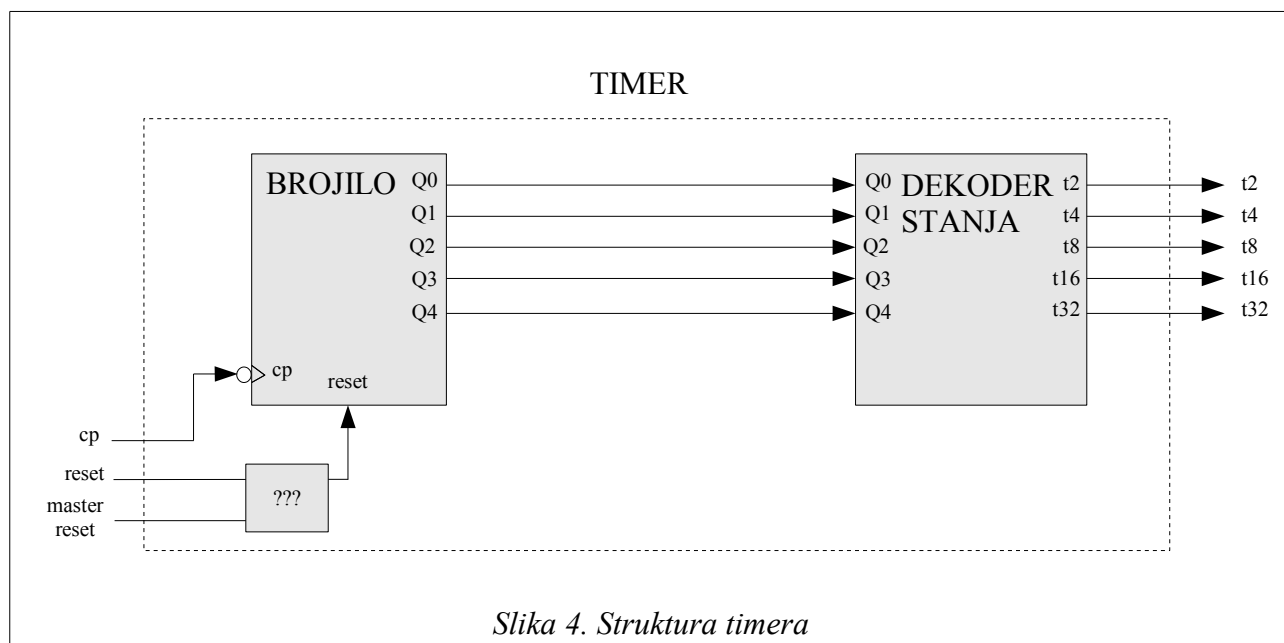
Uočimo na slici što su ulazi i izlazi komponente automat. Ulazi su signali t_2 do t_{32} temeljem kojih automat mijenja svoje stanje i izlaze. Izlazi su podijeljeni u dvije grupe. Prvu grupu čine svjetla semafora, a drugu grupu signal reset kojim automat resetira timer. Signal `master reset` pri tome ne treba tretirati kao ulaz automata (u konceptualnom smislu) – taj ulaz promatrajte kao signal koji će na početku rada resetirati elektroniku automata u početno stanje, i potom nikad više neće biti korišten.



Analizirajmo najprije izvedbu komponente TIMER. Ova komponenta zadužena je za signaliziranje vremenskih intervala. Pritom izlaz t_2 treba generirati '1' ako su prošla dva impulsa CP-a, t_4 ako su prošla 4 impulsa CP-a itd. Jedan od jednostavnijih načina da se napravi ovaj sklop jest uporabom sinkronog 5-bitnog binarnog brojila. Za potrebe ove vježbe taj sklop treba imati sinkroni ulaz za reset, i djelovanjem tog signala brojilo se treba postaviti u stanje 1. U ovoj vježbi sinkrono binarno brojilo izvedite strukturno, uporabom sklopa `sekvsklop` koji ste napisali i provjerili na prošloj laboratorijskoj vježbi. Da biste to mogli, u projekt dodajte modele bistabila `sintff` i potom sklopa `sekvsklop`. Primijetite da je novu komponentu potrebno napraviti zbog usklađivanja sučelja (sklopovi `sekvsklop` i `brojilo` imaju različita sučelja).

Struktura sklopa `timer` prikazana je u nastavku.

Iako je na ovoj slici dekodera stanja nacrtan kao posebna komponenta, prilikom modeliranja ne treba generirati poseban sklop. Dovoljno je svaki od izlaza t_i opisati jednim logičkim AND sklopom (odnosno uporabom operatora AND), iz čega slijedi da se jedino `brojilo` mora posebno modelirati i koristiti u sklopu `timer` kao komponenta. Također obratite pažnju da `timer` ima dva ulaza za reset (oba sinkrona) a `brojilo` samo jedan, pa je te ulaze potrebno na neki način stopiti u jedan (*kojim sklopom?*). Prilikom izvedbe dekodera stanja, izlaz t_2 treba izvesti tako da dekodira stanje brojila 1, izlaz t_4 tako da dekodira stanje brojila 3, izlaz t_8 tako da dekodira stanje brojila 7,



itd. Po završetku pripreme razmislite zašto dekodiramo baš ta stanja? Svaki izlaz dekodera treba kasniti 10 ns.

Jednom kad je timer modeliran, možemo se posvetiti modeliranju samog automata. Automat mora imati toliko različitih stanja koliko mu treba da bi upravljao semaforima prema zadanoj tablici. Koliko će to točno biti, ovisi o nekoliko faktora, između kojih i o vrsti samog automata: Mooreov automat ili Mealyjev automat. No neovisno o vrsti automata, ideja rješenja je ista: automat se nalazi u nekom stabilnom stanju, i tu čeka signal sklopa timer. Jednom kad timer signalizira da je vrijeme za promjenu stanja, automat prelazi u novo stanje, i resetira timer (međutim, uočite da se zbog sinkronog djelovanja reseta timer ne resetira odmah, nego tek na sljedeći brid CP-a čime brojilo/timer efektivno propušta jedan impuls takta; rješenje ovog problema je postavljanje brojila u stanje 1 a ne u 0 na signal reseta).

Ovdje opisan automat u pripremi **treba riješiti na dva načina**: najprije treba **nacrtati Mooreov automat** koji rješava zadani problem, a potom **Mealyjev automat**.

Prilikom modeliranja Mooreovog automata problem se sigurno može riješiti s 13 stanja. Kod izrade automata sjetite se da timer nakon svake promjene na semaforu mora ponovno početi mjeriti vrijeme.

Mealyjev automat može riješiti sa samo 7 stanja (*može li možda i manje? Razmislite!*). Naime, Mealyjev automat je “*stroj prijelaza*” pa različit izlaz možete definirati za prijelaz kojim automat dolazi u novo stanje, a različit za prijelaz kojim automat ostaje u istom stanju.

Više primjera kako se konstruiraju sami automati koji rješavaju određene probleme može se pogledati u [2] (posebno zadaci 11.1 do 11.5).

Prije dolaska na laboratorijsku vježbu, nacrtajte slike automata (stanja + prijelaze) u nastavku. Budući da u svakom stanju može djelovati puno kombinacija pobuda, poslužite se jednostavnom pokratom. Za pobude koje su “interesantne”, eksplicitno ih navedite na luku. Sve ostale pobude koje ne želite eksplicitno nabrajati označite sa *. Primjerice, neka neki automat ostaje u stanju S40 za sve pobude osim t8, a za t8 prelazi u stanje S41. Nacrtat ćete stanja S40 i S41, luk od S40 u S41 i označiti ga pobudom t8 te luk koji ide iz S40 u S40 i označiti ga sa *.

Izvedba sklopa automat kao Mooreov stroj s konačnim brojem stanja – nacrtajte!

Izvedba sklopa automat kao Mealyjev stroj s konačnim brojem stanja – nacrtajte!

Upute kako jezikom VHDL opisati pojedine vrste automata mogu se pogledati u [2] (zadatak 11.19 – Moore, te zadatak 11.20 – Mealy). Kod Mealyjevog automata napraviti onu varijantu koja izlaze čuva u izlaznom registru. Za modeliranje stanja u obje varijante koristite konstante; nemojte koristiti korisničke tipove podataka (ključna riječ `TYPE`). Također oba automata modelirati tako da signal `reset` na njih djeluje sinkrono (a ne asinkrono kao u primjerima u [2]). Izlaz automata mora imati vrijeme kašnjenja od 10 ns.

Na laboratoriju, nakon modeliranja svake komponente potrebno je napraviti i ispitni modul, te ispitati rad sklopa.

Prilikom modeliranja i ispitivanja brojila, a općenito i cijelokupnog sklopa, zgodno je pretpostaviti da T traje jednu periodu signala vremenskog vođenja. Kasnije se relativno jednostavnim modifikacijama brojila ta vrijednost može povećati na proizvoljnu vrijednost.

Kao dio pripreme također razmislite što bi se dogodilo kada bi signal `reset` brojilo postavljao u stanje 0, te što bi se dogodilo kada bi signal `reset` za brojilo bio asinkron? Kako bi se to odrazilo na rad sklopa? Vaš zaključak možete provjeriti na laboratorijskoj vježbi.

4. Priprema

Na samu vježbu potrebno je donijeti ovu uputu u kojoj sva polja za unos rješenja trebaju biti popunjena. VHDL kod treba biti unesen u sustav VHDLLab2. Svi modeli moraju biti prikladno testirani.

Automat modelirajte ponašajno, pisanjem VHDL koda. Pri tome za čuvanje stanja koristite interni signal tipa `std_logic_vector` potrebne širine i konstante kojima ćete definirati kako izgleda tablica kodiranja stanja (nemojte koristiti ključnu riječ `type`).

5. Pitanja za vježbu

- Koja je razlika između Mooreovog i Mealyjevog stroja s konačnim brojem stanja?
- O čemu ovisi sljedeće stanje Mooreovog automata?
- O čemu ovisi izlaz Mooreovog automata?
- O čemu ovisi sljedeće stanje Mealyjevog automata?
- O čemu ovisi izlaz Mealyjevog automata?
- Općenito govoreći, za modeliranje nekog sklopa uporabom automata potrošit ćemo više stanja kod koje vrste automata – Mooreovog ili Mealyjevog?
- Kako bi se promijenilo ponašanje sklopa kada bi reset prebacio brojilo u stanje 0?
- Kako bi se promijenilo ponašanje sklopa kada bi reset djelovao asinkrono?
- Nacrtajte strukturni model 5-bitnog sinkronog binarnog brojila izvedenog T bistabilima.
- Napišite ponašajni model 5-bitnog sinkronog binarnog brojila. Pristupite ovom problemu kao da modelirate višebitni registar s paralelnim ulazima i paralelnim izlazima, pri čemu se izlazi vode na zbrajalo koje trenutnom stanju pridodaje 1 i taj rezultat vraća na paralelni ulaz registra. Ovo se sve može riješiti unutar jednog bloka procesa i uporabom nekoliko varijabli.

Literatura:

- [1] Peruško, Glavinić: *Digitalni sustavi*. Školska knjiga, 2005.
- [2] Čupić, *Digitalna elektronika i Digitalna logika. Zbirka riješenih zadataka*. Kigen, 2006.