

Sveučilište u Zagrebu
Fakultet elektrotehnike i računarstva

Digitalna logika

Laboratorijske vježbe korištenjem sklopovskih pomagala

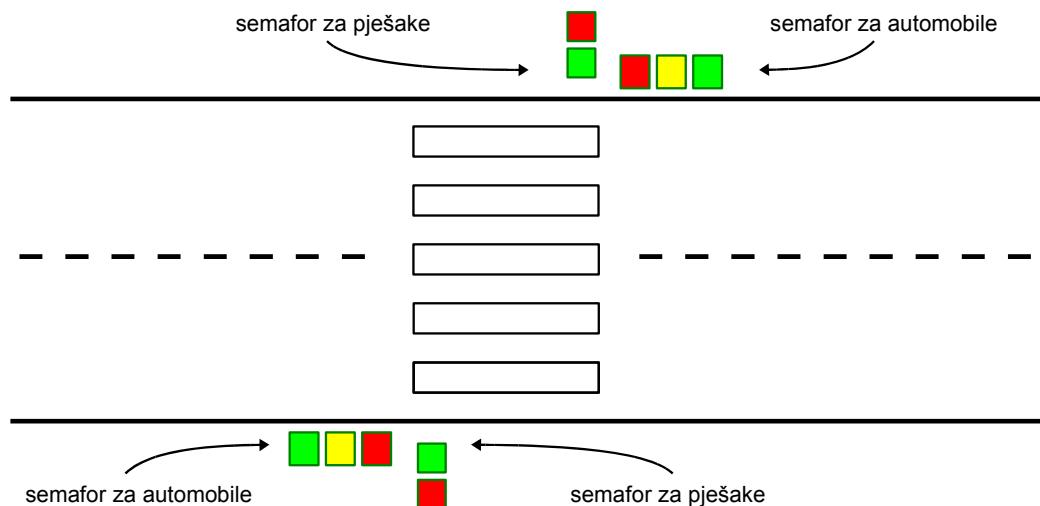
Upute za 6. laboratorijsku vježbu

Marko Zec

Siječanj 2014.

1 Problem: pokvareni semafor

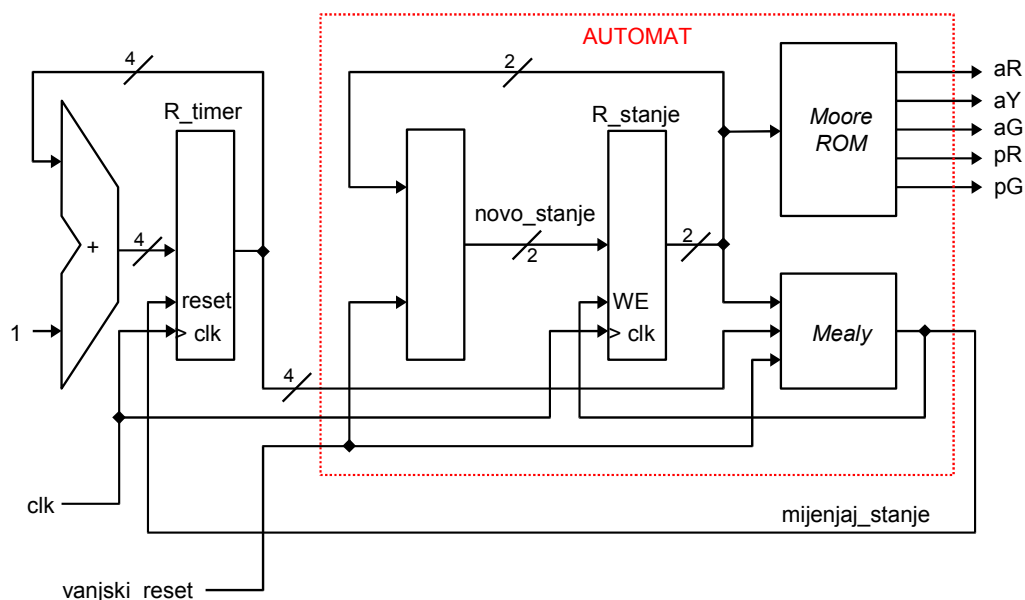
Vaš je zadatak već postojeći, nefunkcionalni prototip upravljačkog modula jednostavnog semafora uz što manje preinaka osposobiti za rad u skladu s funkcijskim specifikacijama.



Slika 1: jednostavni semafor za automobile i pješake

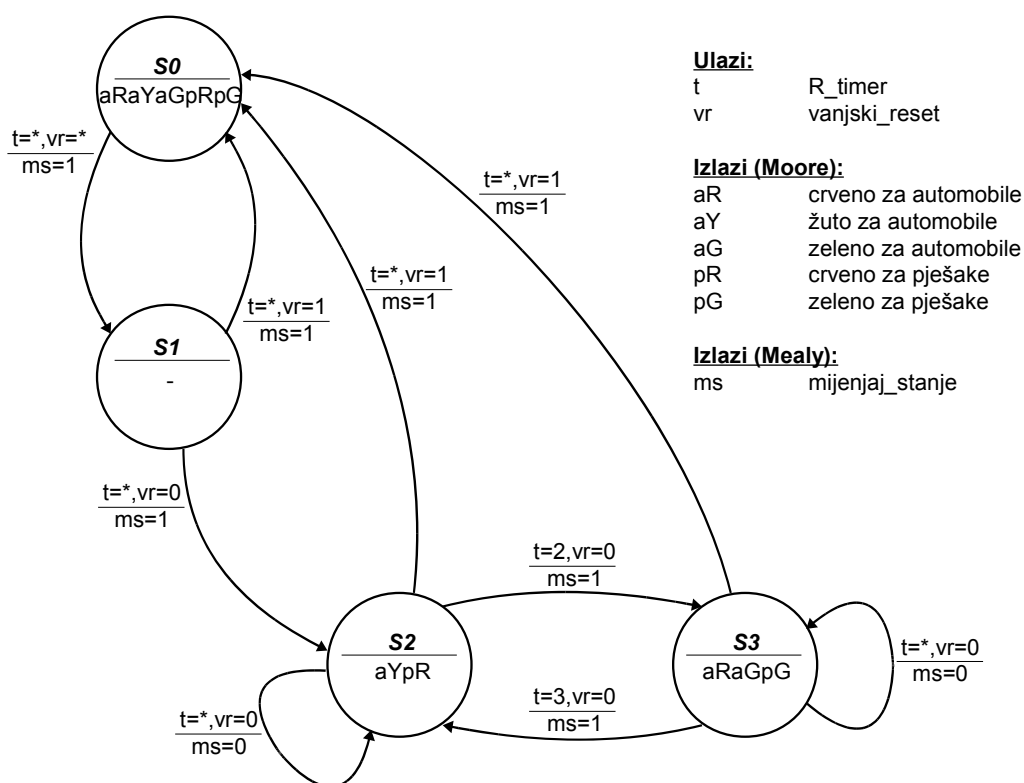
Prototip sklopa koji treba upravljati semaforom sa slike 1 zamišljen je kao automat s konačnim brojem stanja, koji zavisno od ulaza `vanjski_reset` treba raditi u jednom od dva moguća režima rada. Ukoliko je signal `vanjski_reset` postavljen na 1, na semaforu za automobile treba treptati žuto svjetlo, a semafor za pješake treba biti ugašen. Kad je signal `vanjski_reset` postavljen na 0, semafor treba beskonačno ponavljati uobičajeni ciklus paljenja i gašenja signalnih svjetla za automobile i pješake. Za postizanje različitih vremena zadržavanja u pojedinom stanju (npr. u stanju u kojem svijetli zeleno svjetlo automat se treba zadržati dulje nego u stanju u kojem svijetli žuto svjetlo) koristi se vanjski brojač vremenskih impulsa nazvan `timer`, a kojeg se može postaviti na nulu (resetirati) odgovarajućim izlaznim signalom iz automata. Postavljanjem ulaznog signala `vanjski_reset` na 1 automat se u prvom slijedećem ciklusu takta treba prebaciti u režim treptajućeg žutog svjetla bez obzira na trenutno stanje u kojem se nalazi. Rad semafora sinkroniziran je signalom takta frekvencije cca. 1.5 Hz.

Automat za upravljanje radom semafora sastoji se od memorijskog elementa (registra) kojim je određeno trenutno stanje, te kombinacijskih modula kojima se zavisno od ulaza i trenutnog stanja određuju izlazi iz automata, kao i eventualni prijelaz automata u novo stanje. Izlazni signali koji ovise isključivo o trenutnom stanju automata upravljani su kombinacijskim sklopom koji je izveden kao memorija za čitanje (ROM), na čiji se adresni ulaz dovodi kodna riječ trenutnog stanja, a izlazi direktno upravljaju signalnim indikatorima. Signal `mijenjaj_stanje` služi za postavljanje vanjskog brojača ciklusa takta na nulu kod prelaska na novo stanje, a ovisi i o trenutnom stanju i o ulazima u automat. Struktura modula za upravljanje semaforom koji se sastoji od automata i upravljivog brojača ciklusa takta prikazana je blok-shemom na slici 2.



Slika 2: struktura upravljačkog modula `sem_automat`

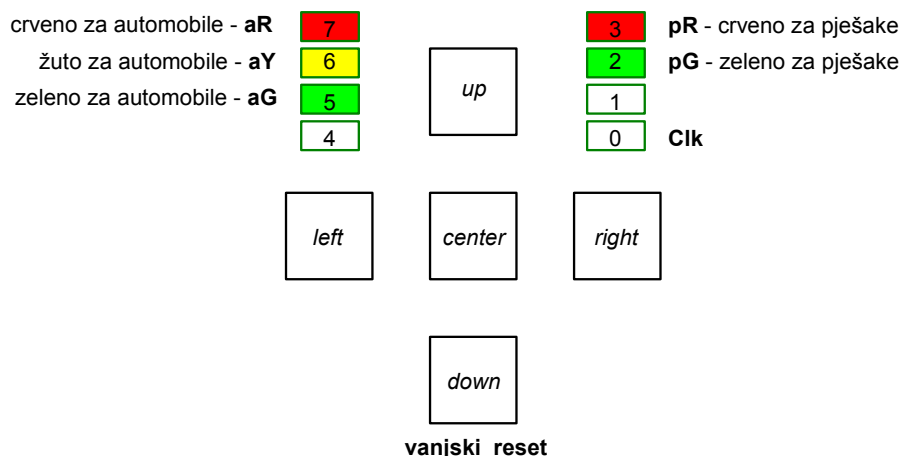
Na slici 3 prikazan je dijagram stanja modula `sem_automat` opisanog datotekom `sem_automat_pokvareni.vhd`, kojeg možete koristiti kao predložak za razvoj vlastitog automata.



Slika 3: dijagram stanja postojećeg predloška sklopa `sem_automat`

Povezivanje ulaznih i izlaznih signala pločice ULX2S i modula `sem_automat` izvedeno je modulom `sem_toplevel`. Modul sadrži i djelitelj takta koji iz ulaznog signala `clk_25m` frekvencije 25 MHz generira signal takta `clk` frekvencije cca. 1.5 Hz. Način

povezivanja izlaznih signala s LED indikatorima, odnosno ulaznog signala `vanjski_reset` s tipkom na FPGA pločici ULX2S ilustriran je slikom 4.



Slika 4: ulazni i izlazni signali modula `sem_toplevel` na pločici ULX2S

2 Priprema

Datoteke `sem_toplevel.vhd` i `sem_automat.vhd` dohvatite s web sjedišta laboratorijskih vježbi, uključite u novi projekt, te sintetizirajte konfiguracijsku datoteku (bitstream) za razvojnu pločicu ULX2S. **Ispitajte rad sklopa i usporedite ga s dijagramom stanja sa slike 3.** Proučite VHDL kod i u njemu identificirajte sve module označene na slici 2: registre, ROM memoriju, ostale kombinacijske module, interne signale koji ih povezuju, te vanjske signale.

U tablici 1 prikazan je željeni raspored stanja kroz koje **ispravni** automat za upravljanje semaforom treba prolaziti:

Stanje	Izlazi	Trajanje
S0	svijetli samo žuto svjetlo za automobile	1 T
S1	ne svijetli ni jedno signalno svjetlo	1 T
S2	svijetli žuto svjetlo za automobile i crveno za pješake	4 T
S3	svijetle crvena svjetla za automobile i za pješake	3 T
S4	svijetli crveno svjetlo za automobile i zeleno svjetlo za pješake	X1 T
S5	svijetle crvena svjetla za automobile i za pješake	4 T
S6	svijetle crveno i žuto svjetlo za automobile i crveno za pješake	3 T
S7	svijetli zeleno svjetlo za automobile i crveno za pješake	X2 T

Tablica 1: željena stanja automata i njihovo trajanje izraženo u ciklusima takta

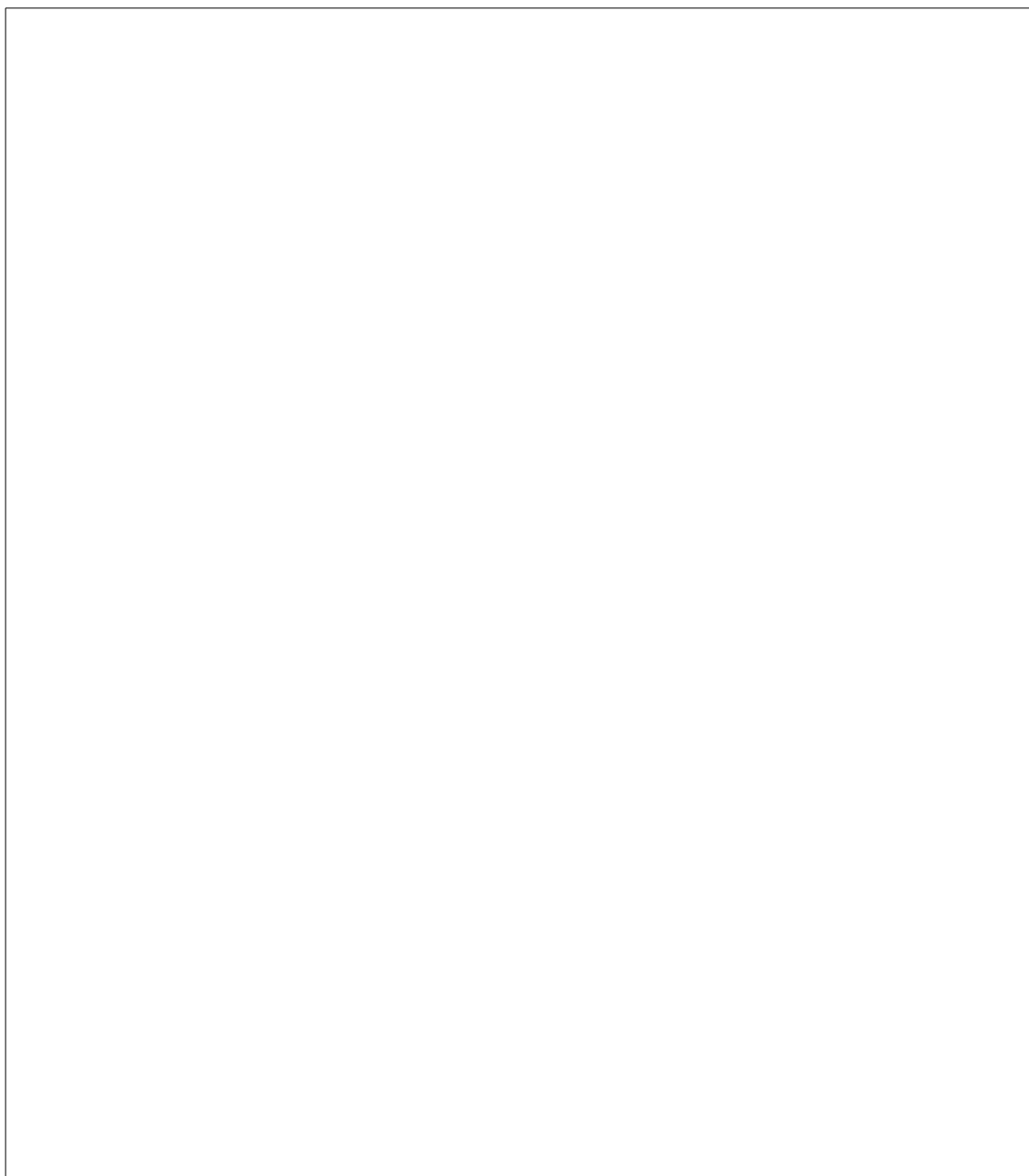
Ukoliko je ulazni signal `vanjski_reset` postavljen na 1, automat se treba prebacivati između prva dva stanja određena tablicom (S0 i S1). Kad je ulazni signal `vanjski_reset` postavljen na 0, automat treba slijedno prolaziti kroz preostalih šest stanja, zadržavajući se u svakom stanju onoliko ciklusa takta koliko je određeno brojem u drugom stupcu tablice. Vremena zadržavanja označena u tablici s **X1** i **X2** određuju se prema formulama:

$$\mathbf{X1 = (JMBAG \bmod 5) + 5}$$

$$\mathbf{X2 = (JMBAG \bmod 5) + 10}$$

Automat treba nakon zadnjeg stanja određenog tablicom (S7) prijeći u stanje S2. Ukoliko se ulazni signal `vanjski_reset` postavi na 1, automat treba prijeći u stanje S0, osim ako se već nalazi u stanju S0. Kod svakog prijelaza u novo stanje automat treba postaviti izlaz `mijenjaj_stanje` na 1, čime će *resetirati* brojač ciklusa takta.

U prazni okvir na ovoj stranici nacrtajte dijagram stanja automata u skladu s tablicom 1, te označite sve prijelaze, ulaze i izlaze automata.



3 Projektiranje automata i ispitivanje rada

3.1 Simulacija u okruženju VHDLLab

Na vlastito računalo instalirajte klijentski dio aplikacije VHDLLab prema uputama sa stranice <http://ferko.fer.hr/vhdlab2/>. Studentima koji još nisu koristili VHDLLab mogu pomoći i animirane upute za rad ilustrirane na jednostavnom primjeru: <http://ferko.fer.hr/upute/vhdlab/vhdlab-lab1.htm>.

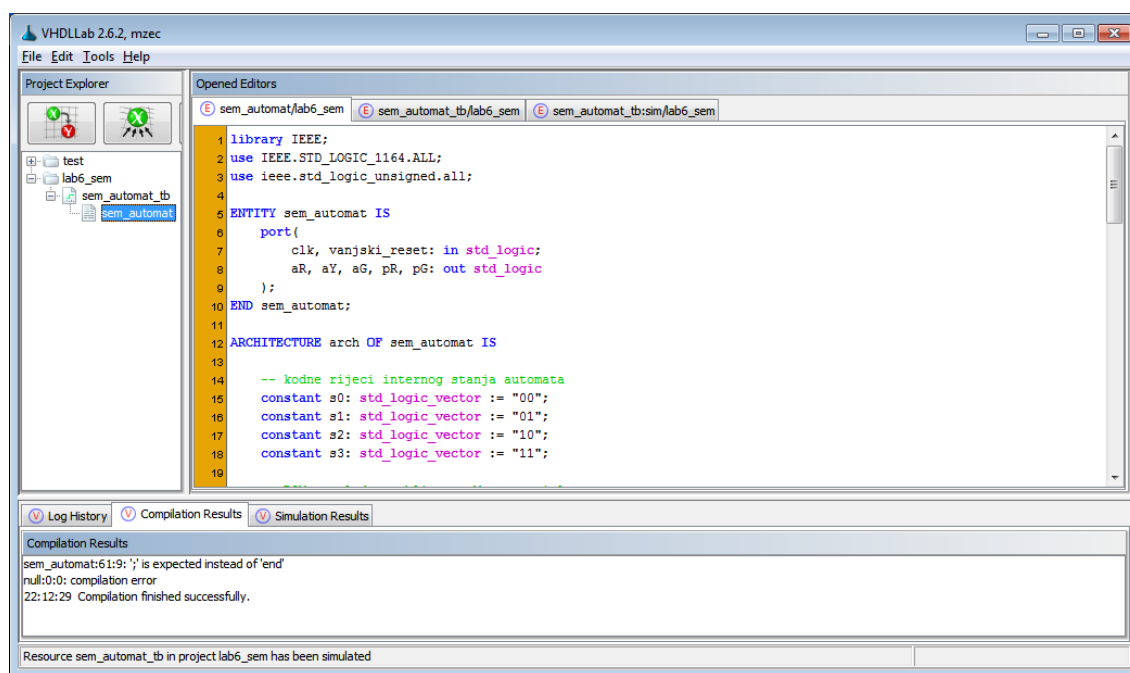
Pokrenite aplikaciju VHDLLab i prijavite se u sustav korisničkim imenom i zaporkom kojima inače pristupate sustavu Ferko.

Stvorite novi projekt: *File – New – Project*.

Unutar novog projekta stvorite novu VHDL datoteku koju treba nazvati `sem_automat`: *File – New – VHDL Source*.

Otvorite VHDL datoteku, te u zaglavlju uključite biblioteku `ieee.std_logic_unsigned.all`.

Prema dijagramu stanja kojeg ste nacrtali na prethodnoj stranici VHDLom opišite odgovarajući automat. Ulazni signali modula trebaju biti nazvani točno `clk` i `vanjski_reset`, a izlazi `aR`, `aY`, `aG`, `pR` i `pG`. Svi signali u sučelju modula trebaju biti tipa `std_logic`. Pri izradi modula možete koristiti predložak `sem_automat_pokvareni.vhd`, ali automat možete modelirati i prema vlastitim zamislima, uz uvjet da funkcijski odgovara zadanim specifikacijama, te da je izveden u samo jednom VHDL modulu odnosno datoteci.

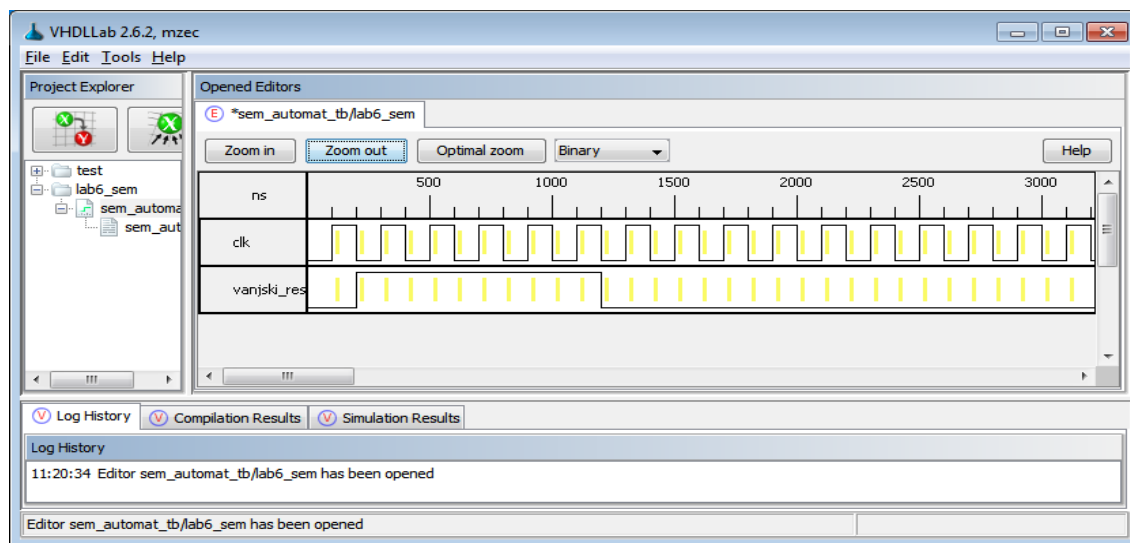


Slika 5: unos VHDL opisa modula `sem_automat` u sustavu VHDLLab

Nakon što se uredili opis modula (slika 5), pripremite ga za simuliranje: *Tools – Compile*.

Unutar postojećeg projekta stvorite novu ispitnu pobudu vezanu uz modul

sem_automat: File – New – Testbench. Ispitna pobuda treba sadržavati bar tridesetak pravilnih impulsa na ulazu `clk`, što se najbrže može urediti kroz izbornik "Apply pattern" kojeg se može otvoriti pritiskom na desnu tipku miša iznad željenog signala. U početnom dijelu ispitne pobude signal `vanjski_reset` treba biti aktivan bar pet ciklusa takta, kao što je prikazano na slici 6.



Slika 6: ispitna pobuda

Pokrenite simulaciju, te po potrebi korigirajte opis automata uz ponavljanje postupka sve dok simulirani odziv sklopa ne bude odgovarao zadanom.

3.2 Sinteza za rad na FPGA pločici

Automat razvijen i simuliran u okruženju VHDL Lab uključite u projekt u okruženju Lattice Diamond na način da u VHDL Labu obilježite cijeli kod (sekvenca `ctrl+A`, `ctrl+C`) te ga kopirajte (`ctrl+V`) u praznu datoteku u okruženju Diamond.

Nakon što ste sintetizirali i ispitati rad sklopa na FPGA pločici, u sustav Ferko prebacite (uploadajte) datoteke `sem_automat.vhd` i `lab7.jed`.

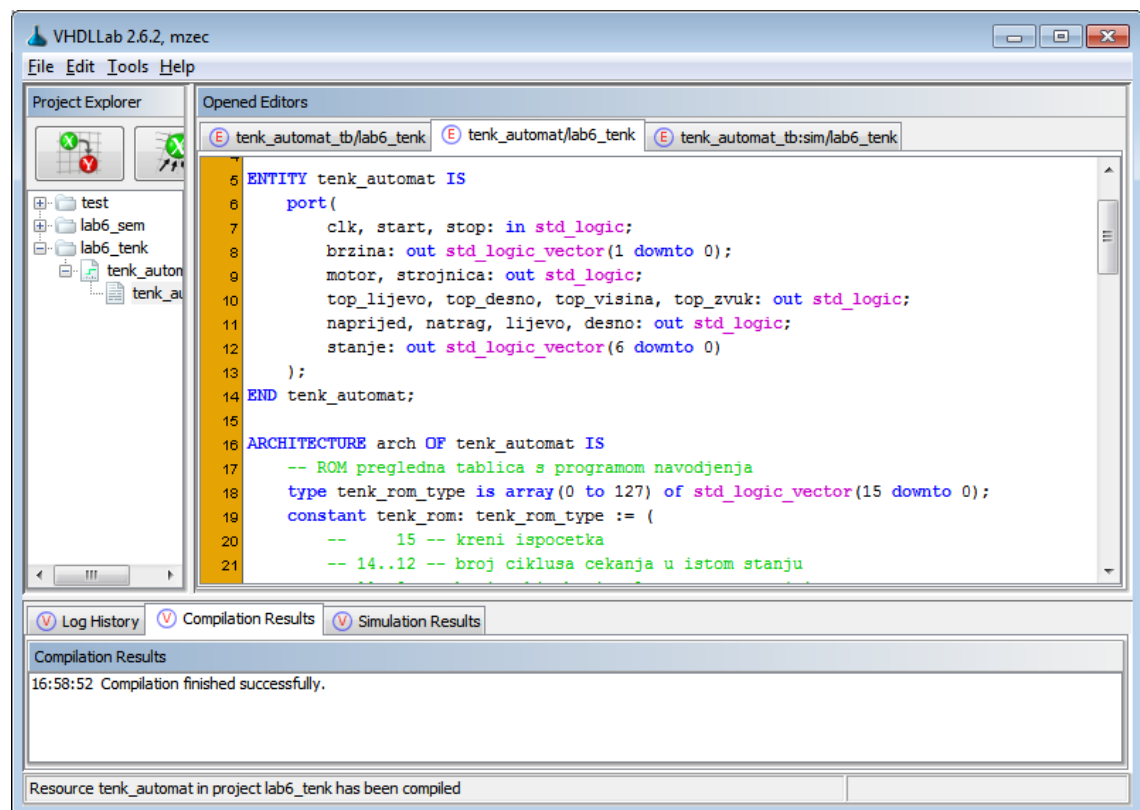
4 Automat za vođenje daljinski upravljane makete

Vaš je zadatak koristeći se već gotovim predloškom konstruirati automat za vođenje radioupravljive makete po unaprijed zadanom putu.

Automat treba imati točno tri ulaza: `clk`, `start` i `stop`. Na ulaz `clk` dovest će se signal takta frekvencije cca. 1.5 Hz. Automat treba čekati u stanju 0 sve dok se na ulazu `start` ne pojavi impuls u trajanju najmanje jednog ciklusa takta, nakon čega treba slijedno prolaziti kroz više stanja kojima će se postavljati signali daljinskog upravljanja makete. Impuls u trajanju najmanje jednog ciklusa takta na ulazu `stop` treba odmah prekinuti rad automata vraćanjem u stanje 0.

Većina izlaznih signala automata povezat će se s ulazima već gotovog modula `rf_modulator` za radiofrekvencijsko moduliranje upravljačkog signala kojeg ste koristili u 1. laboratorijskoj vježbi, dok će 7-bitni izlaz `stanje` biti povezan s LED indikatorima razvojne pločice, kako bi se mogučilo vizualno praćenje rada automata.

Zadano sučelje automata sa svim ulaznim i izlaznim signalima prikazano je na slici 7.



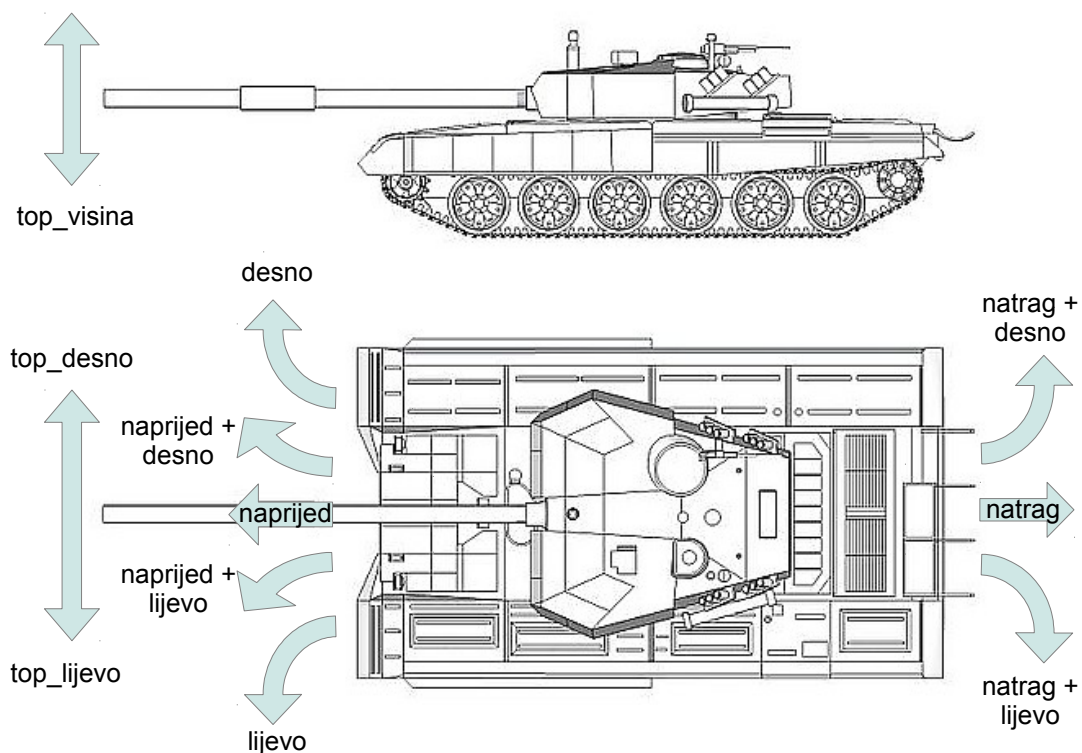
Slika 7: sučelje modula `tenk_automat` u editoru VHDLLab

4.1 Priprema

S web sjedišta laboratorijskih vježbi dohvatite datoteke `tenk_toplevel.vhd`, `tenk_automat.vhd` i `rf_modulator.vhd`, te ih uključite u novi projekt u okruženju Lattice Diamond. Proučite strukturu modula `tenk_toplevel` i `tenk_automat`, sintetizirajte sklop i dobivenom konfiguracijom programirajte FPGA pločicu, te praćenjem odziva na tipke `start` i `stop` na LED indikatorima ispitajte rad sklopa. Uočite način kodiranja vremena zadržavanja u pojedinom stanju te direktnog upravljanja izlaznim signalima koji je ostvaren 16-bitnim kodnim riječima pohranjenim u ROM tablicu `tenk_rom`.

Vaš je zadatak modificirati postojeći predložak modula `tenk_automat`, ili po Vašem izboru konstruirati potpuno novi automat, kojim će se moći ispitati sve funkcije daljinskog upravljanja makete tenka. Imate potpunu slobodu u osmišljavanju puta makete te slijeda ispitivanja ostalih funkcija upravljanja, pri čemu ipak vodite računa o ograničenom prostoru po kojem se maketa može kretati u laboratoriju. Iz istog razloga, iako je dopušteno ispitati sve brzine kretanja makete, preporuča se ograničiti vođenje makete većim brzinama na kratke vremenske intervale.

Funkcije glavnih upravljačkih signala makete ilustrirane su na slici 8, a detaljnije su opisane u uputama za 1. laboratorijsku vježbu. Kod projektiranja automata preporuča se modul `tenk_automat` prvo modelirati i njegovo ponašanje simulirati u sustavu VHDLLab, a nakon dobivanja zadovoljavajućeg odziva VHDL opis prebaciti u okruženje Lattice Diamond i pripremiti konfiguracijsku datoteku za FPGA sklop.



Slika 8: funkcije glavnih upravljačkih signala modula *rf_modulator*

4.2 Rad u laboratoriju

- Uključite napajanje makete pomicanjem **većeg** prekidača s donje strane tenka u položaj "ON". Signalne bijele LED lampice s prednje strane tenka bi trebale početi treptati u pravilnom ritmu;
- U laboratoriju će biti raspoložive dvije makete tenka, s prijemnicima podešenima za rad na različitim kanalima. Uskladite radiofrekvencijski kanal odašiljača i prijemnika na način da mikroprekidače `sw_1` i `sw_0` postavite u isti položaj koji je označen na anteni makete tenka kojom ćete upravljati (npr. "10" ili "11");
- Programirajte FPGA pločicu putem USB sučelja, ili je priključite na baterijsko napajanje, ukoliko ste konfiguraciju FPGA sklopa programirali u *flash* memoriju. U višenamjensku stereo-utičnicu razvojne pločice utaknite žičanu antenu;
- Pokrenite automat pritiskom na tipku `btn_up` generirajte kratki impuls na ulazu automata `start`. U svakom trenutku trebate biti spremni zaustaviti rad automata pritiskom na tipku `btn_down`, koja je povezana s ulazom automata `stop`.
- Nemojte ometati kolege koje trenutno ispituju rad svog sklopa generiranjem radiofrekvencijskog signala na istom kanalu;
- Makete je dozvoljeno voziti isključivo po podu;
- Vodite računa da ne dolazi do sudara dvaju maketa;