

# Uputstvo za korišćenje osnovnih funkcionalnosti Altera Quartus II i ModelSim softvera

Katera za elektroniku  
Elektrotehnički fakultet  
Univerzitet u Beogradu

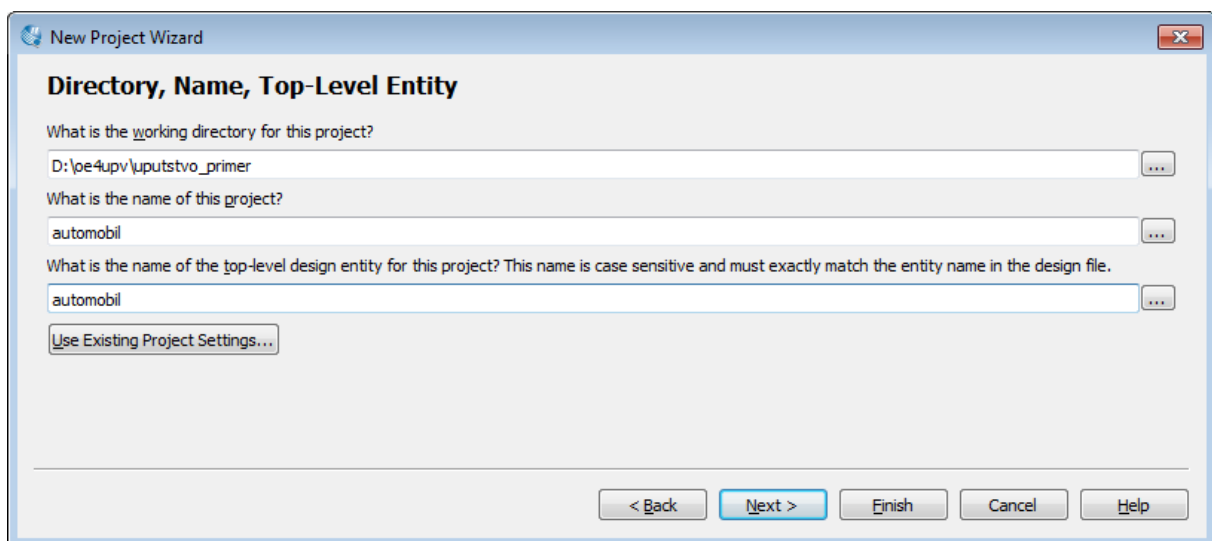
11. oktobar 2015

## Uvod

Ovo je kratko uputstvo za korišćenje Altera Quartus II softvera i Altera ModelSim simulatora. Verzija uputstva: 1.1.

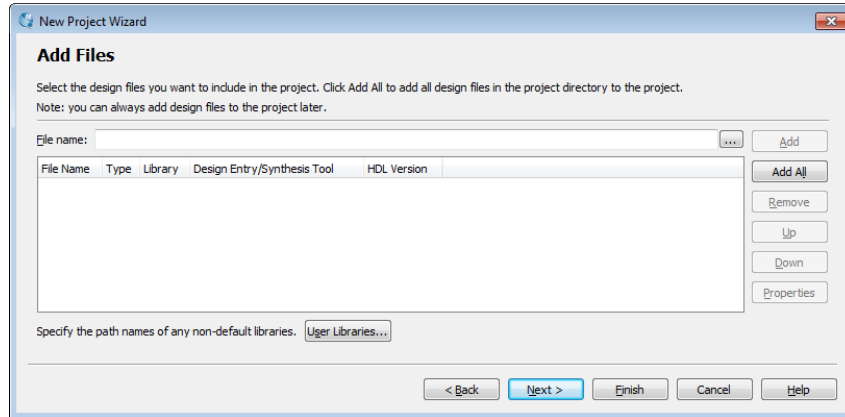
## Kreiranje projekta, analiza i sinteza za FPGA

Pokrenuti Altera Quartus II softver i odabrati File→New Project Wizard. Tada se otvara uvodni prozor na kome je potrebno kliknuti Next. Nakon toga se otvara prozor u kome treba specificirati radni direktorijum projekta (slika 1), ime projekta i ime entitija koji predstavlja najviši modul u hijerarhiji projekta. Poželjno je da putanja direktorijuma projekta bude bez razmaka kako ne bi nastali potencijalni problemi u radu. Primer u ovom uputstvu je zadatak sa časova vežbi (zasnovan na zadatku 5 sa sledećeg linka [tnt.etf.rs/~oe2ode/pdf/vezbe/ode\\_v01\\_osnovna\\_logicka\\_kola.pdf](http://tnt.etf.rs/~oe2ode/pdf/vezbe/ode_v01_osnovna_logicka_kola.pdf)).



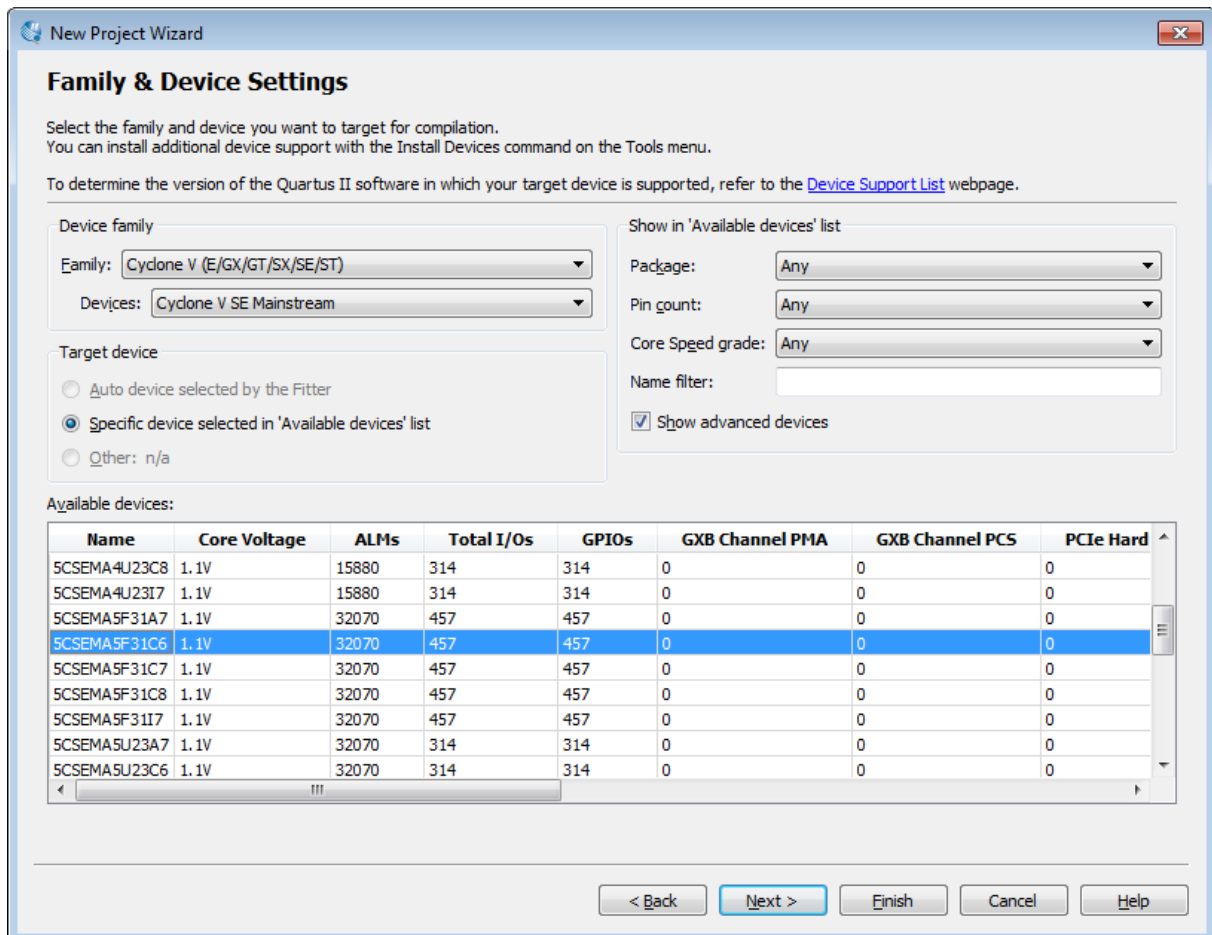
Slika 1

Klikom na Next i odabirom opcije Empty project i ponovnim klikom na Next dolazi se na prozor za dodavanje fajlova u projekat (slika 2). U okviru ovog prozora mogu se dodati već napisani .vhd fajlovi sa VHDL kodom. Klikom na Add All u projekat se dodaju svi fajlovi iz podešenog direktorijuma projekta. Kako trenutno nema napravljenih fajlova, kliknuti Next.



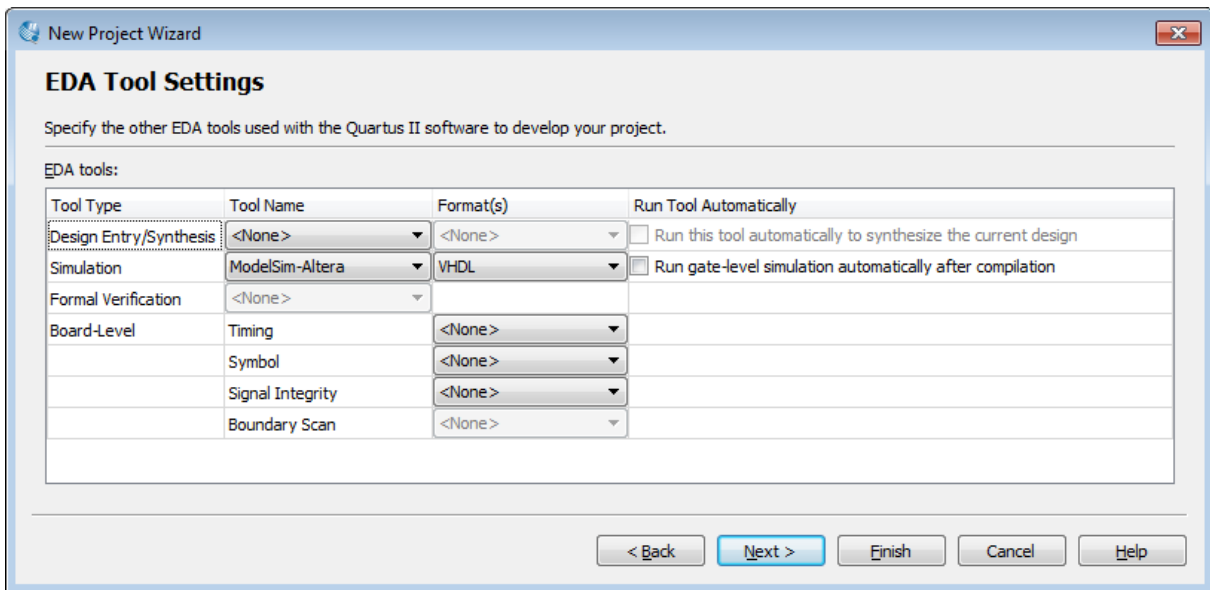
Slika 2

Zatim se otvara prozor za podešavanje FPGA čipa koji će biti programiran nakon što se završi dizajn (slika 3). Odabrati Cyclone V familiju čipova i odabrati čip sa oznakom 5CSEMA5F31C6 kao na slici 3.



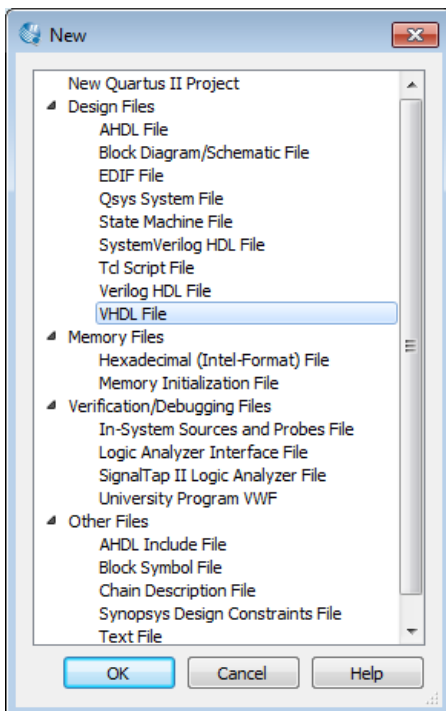
Slika 3

Klikom na **Next** otvara se prozor sa slike 4 u kom samo treba proveriti da li je za simulaciju u padajućem meniju za **Tool Name** podešeno **ModelSim-Altera** i da li je format **VHDL**.



Slika 4

Klikom na **Next** otvara se **Summary** prozor u kom se može proveriti da li je sve dobro podešeno i klikom na **Finish** kreira se projekat u kome se sada mogu kreirati novi VHDL fajlovi, dodavati već postojeći, kompajlirati dizajn itd.




Slika 5

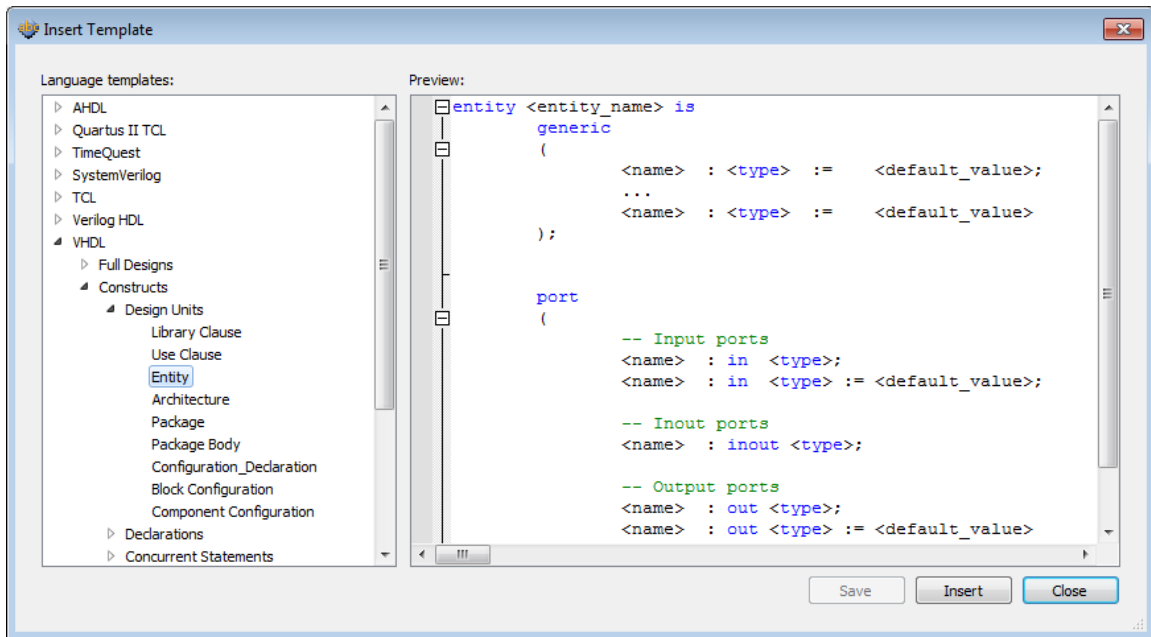
Odabirom opcije **File**→**New** otvara se prozor sa slike 5 gde treba odabrati **VHDL File** i kliknuti na **OK**.

Na početku svakog VHDL fajla je poželjno napisati zaglavlje sa komentarima koji sadrže naziv projekta, kratak opis sistema, ime autora, datum, verziju koda i sl.

Često se u pisanju koda koriste tipovi podataka ili komponente iz nekih biblioteka. Vrlo česta je **ieee** biblioteka i ona se koristi na ovom kursu. Zbog toga je na početku svakog VHDL fajla, odmah posle zaglavlja potrebno uključiti **ieee** biblioteku:

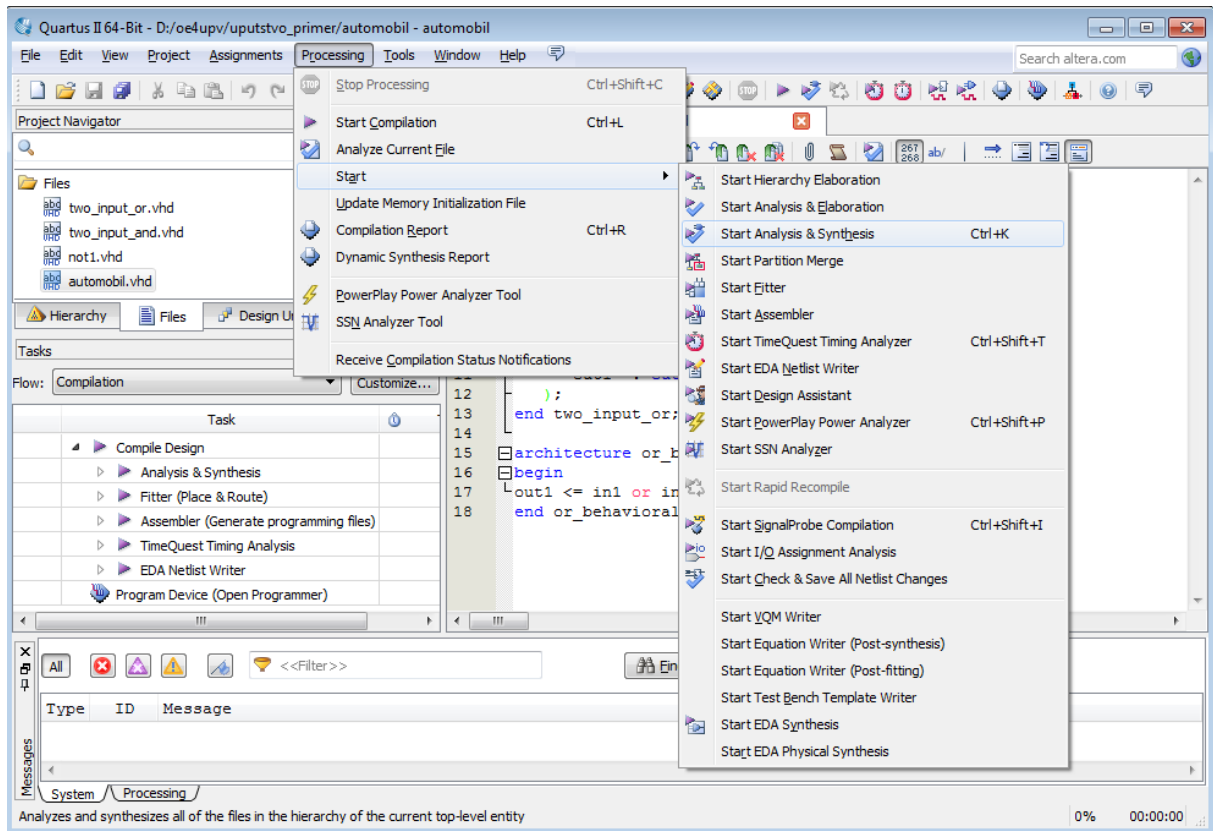
```
library ieee;
use ieee.std_logic_1164.all;
```

Altera Quartus II softver nudi veliki broj šablona za različite VHDL sintaksne konstrukcije, čak i za veliki broj aritmetičkih, logičkih i memorijskih komponenti. Za početak ovo može biti vrlo korisno osobama koje uče jezik i ne snalaze se sa sintaksom. Odabirom opcije **Edit**→**Insert Template** ili klikom na ikonicu sa oznakom  otvara se prozor sa slike 6 u kome se mogu odabrati različiti šabloni VHDL koda. Klikom na **Insert** ceo šablon se upisuje u aktivni fajl i može se editovati.



Slika 6

U ovom uputstvu neće biti detalja o pisanju VHDL koda. Detaljan opis jezika se obrađuje na predavanjima i dostupan je na slajdovima sa predavanja, a može se naći veliki broj uputstava na internetu<sup>1</sup>, ili u knjigama<sup>2</sup>.



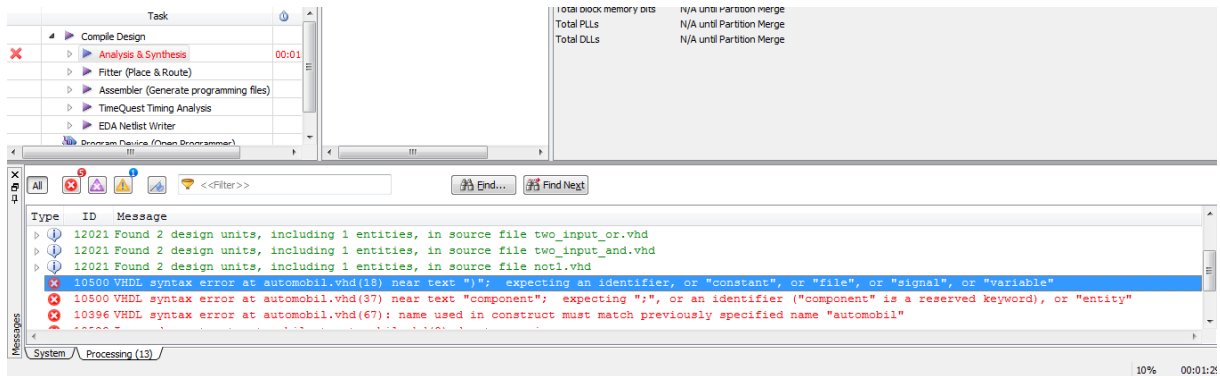
Slika 7

<sup>1</sup>[www.seas.upenn.edu/~ese171/vhdl/vhdl\\_primer.html#\\_Toc526061340](http://www.seas.upenn.edu/~ese171/vhdl/vhdl_primer.html#_Toc526061340)

<sup>2</sup>npr: Dodatak A *Hardware Description Languages* knjige N. H. E. Weste and D. M. Harris, *CMOS VLSI design: a circuits and systems perspective*, 4th ed. Addison-Wesley, 2011

Nakon što se završi sa kucanjem koda, potrebno je kompajlirati ga. To se postiže odabirom opcije Processing→Start Compilation. Za funkcionalnu simulaciju nije potrebno uraditi celu kompilaciju već samo analizu i sintezu što se postiže odabirom opcije Processing→Start→Start Analysis and Synthesis ili duplim klikom na Analysis and Synthesis u prozoru Tasks (slika 7).

Ukoliko postoje neke sintaksne greške, one će se pojaviti u prozoru sa porukama (Messages) kao crvena obaveštenja (slika 8). Dupli klik na te linije će pokazati na mesto gde je greška nastala. Ako je kod uspešno kompajliran, može se preći na pisanje *test bench* fajla i simulaciju kojom se utvrđuje rad modula koji je napravljen.




Slika 8

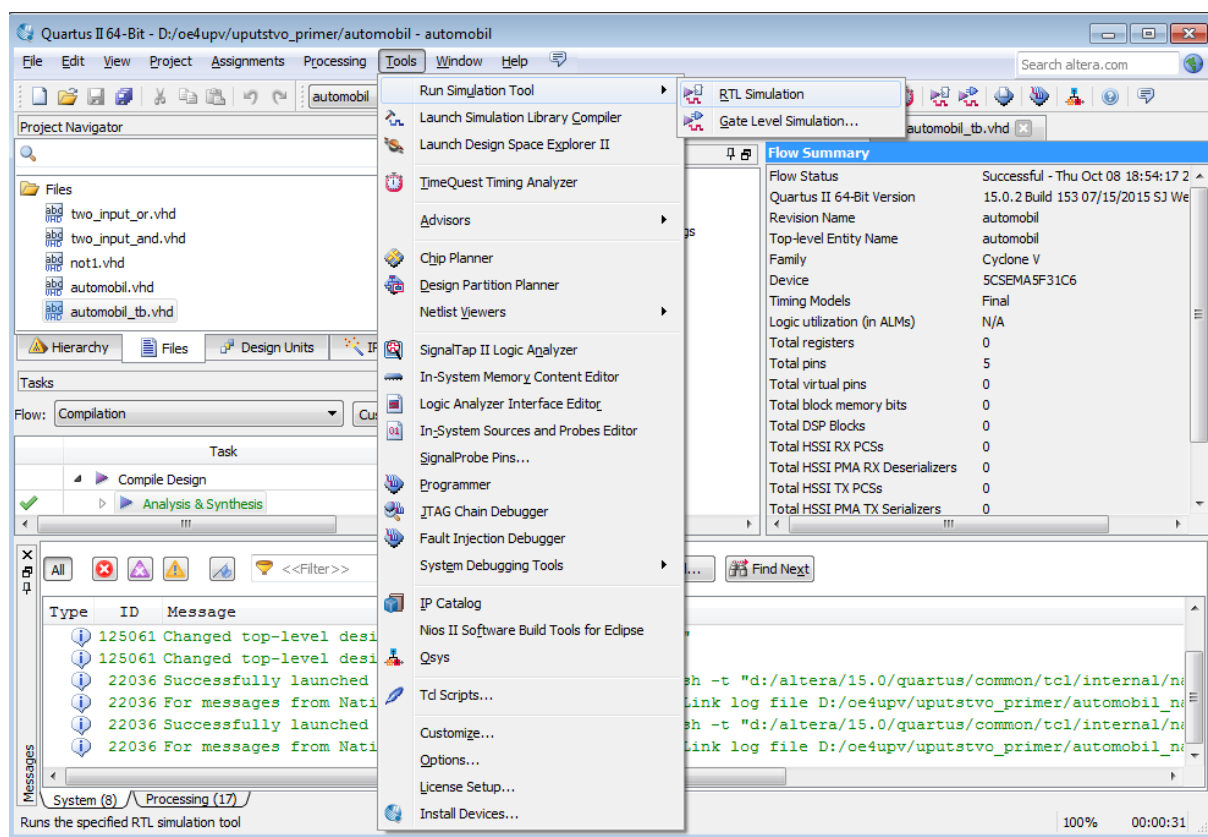
## Test bench testiranje i simulacija

U ovom poglavlju biće opisana dva načina za simulaciju rada modula korišćenjem *test bench* modula. Na korisniku je da odluči koji će od ta dva načina koristiti.

### Ručno pisani *test bench* moduli

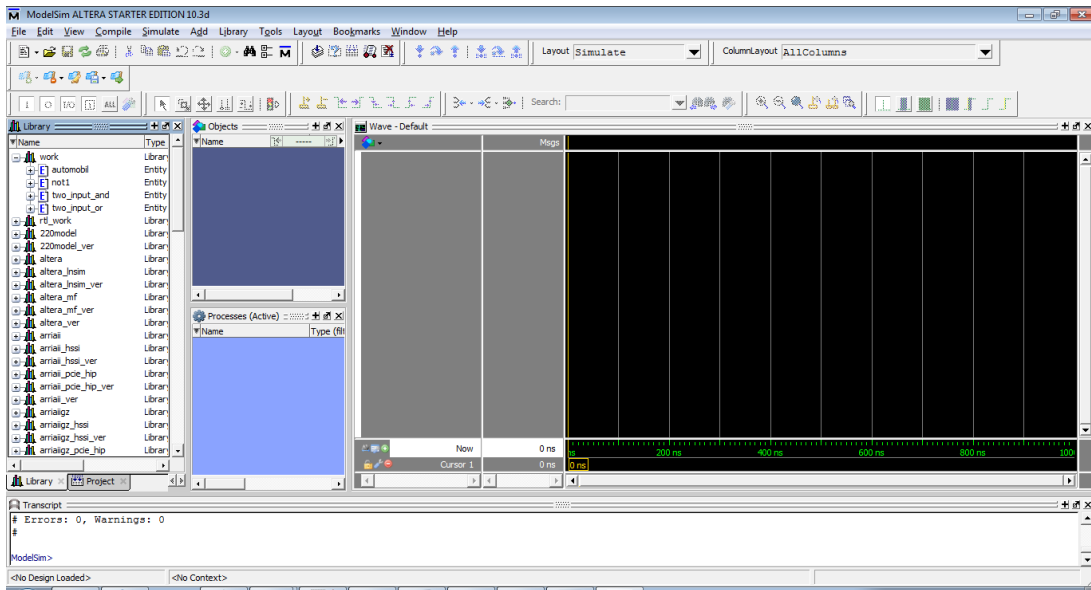
*Test bench* fajlovi se mogu pisati ispočetka, na isti način kao i fajlovi koji opisuju rad modula. Obično se onda fajlu koji služi za testiranje modula ime\_modula dodeli ime ime\_modula\_tb.vhd. Nakon što se napiše ceo *test bench* fajl, uradi se ponovo analiza i sinteza.

Sada je potrebno pokrenuti alat za funkcionalnu simulaciju i to odabirom opcije Tools→Run Simulation Tool→RTL Simulation ili klikom na ikonicu sa oznakom  (slika 9).



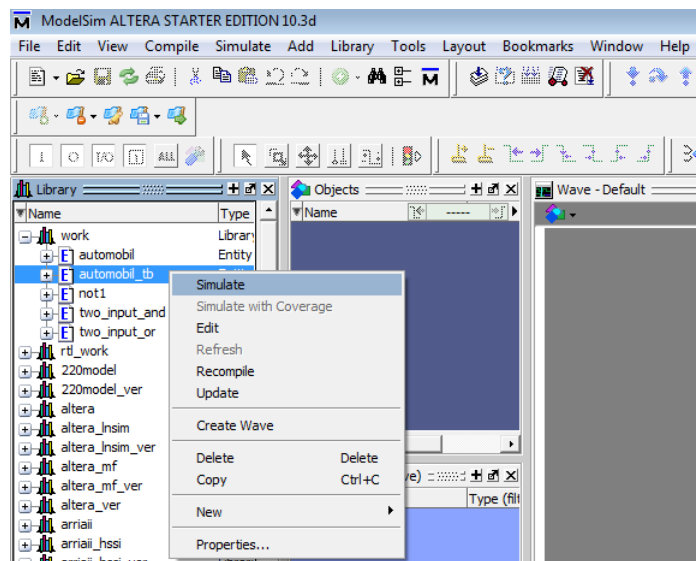
Slika 9

Time se otvara ModelSim simulator prikazan na slici 10. Sa leve strane glavnog prozora simulatora nalazi se prozor Library (ako nije uključen, uključuje se izborom opcije View→Library) gde je prikazana biblioteka work u okviru koje se nalaze svi moduli iz projekta, međutim modul koji odgovara *test bench* entitiju nije u biblioteci, pa je potrebno kompajlirati ga naknadno i to izborom opcije Compile→Compile pri čemu se otvara prozor u kome je potrebno naći *test bench* fajl i kliknuti na Compile, a po završetku kompilacije na Done. Po završetku kompilacije se u biblioteci work pojavljuje modul *test bench* fajla.



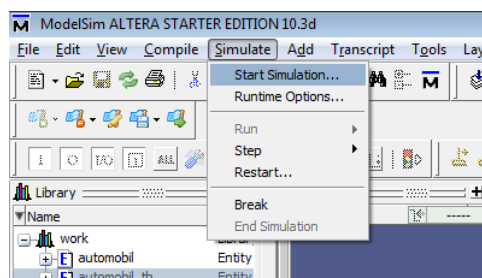
Slika 10

Simulacija se pokreće desnim klikom na modul iz *test bench* fajla u prozoru biblioteke i klikom na Simulate (slika 11).

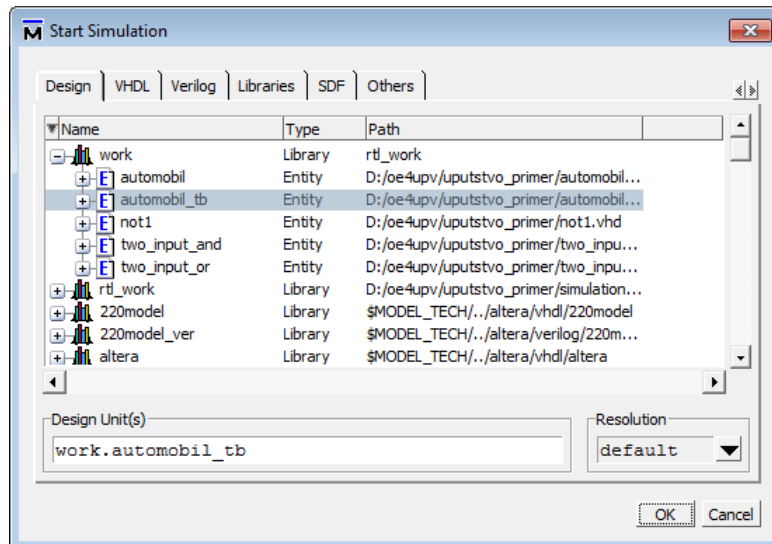


Slika 11

Simulacija se može pokrenuti i izborom opcije Simulate→Start Simulation (slika 12), a zatim odabirom odgovarajućeg modula (slika 13).



Slika 12

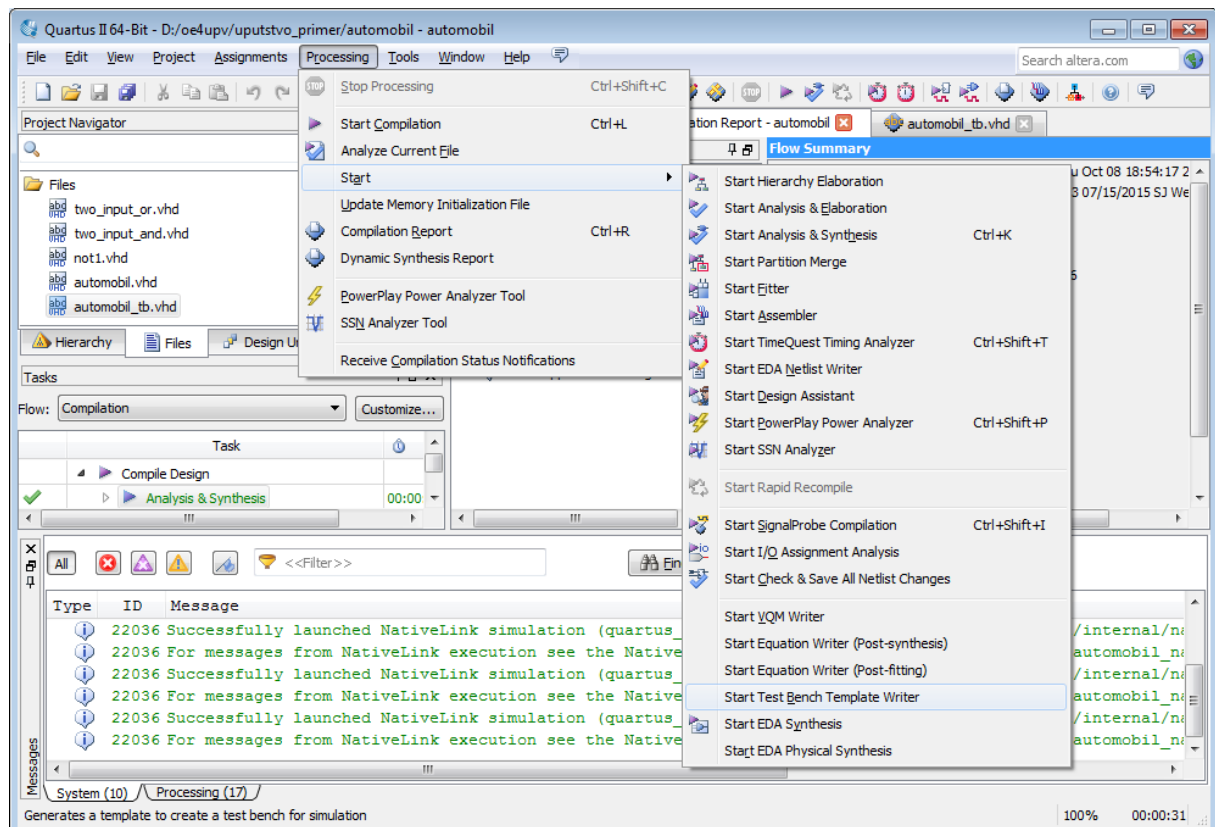


Slika 13

Sada se dolazi u okruženje u kome se mogu posmatrati signali a koje će biti detaljnije opisano nakon objašnjenja automatskog generisanja *test bench* modula.

## Automatski generisani *test bench* moduli

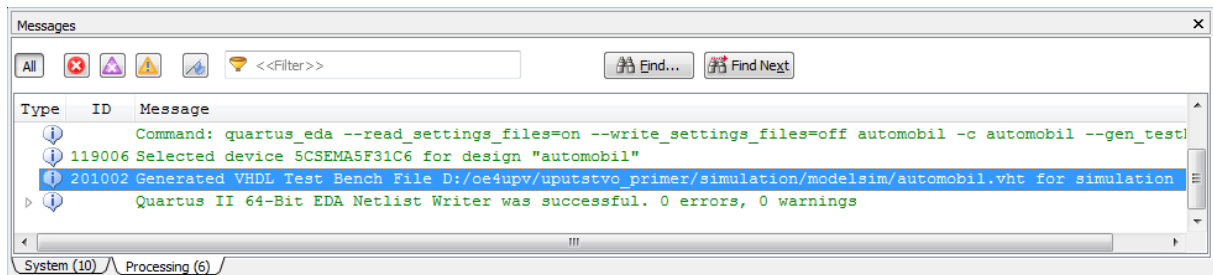
Nakon završene analize i sinteze modula koji se dizajnira, može se pokrenuti automatsko generisanje *test bench* modula izborom opcije Processing→Start→Start Test Bench Template Writer (slika 14).



Slika 14



Nakon toga se u porukama ispiše gde je generisan šablon za *test bench* fajl (slika 15).



Slika 15

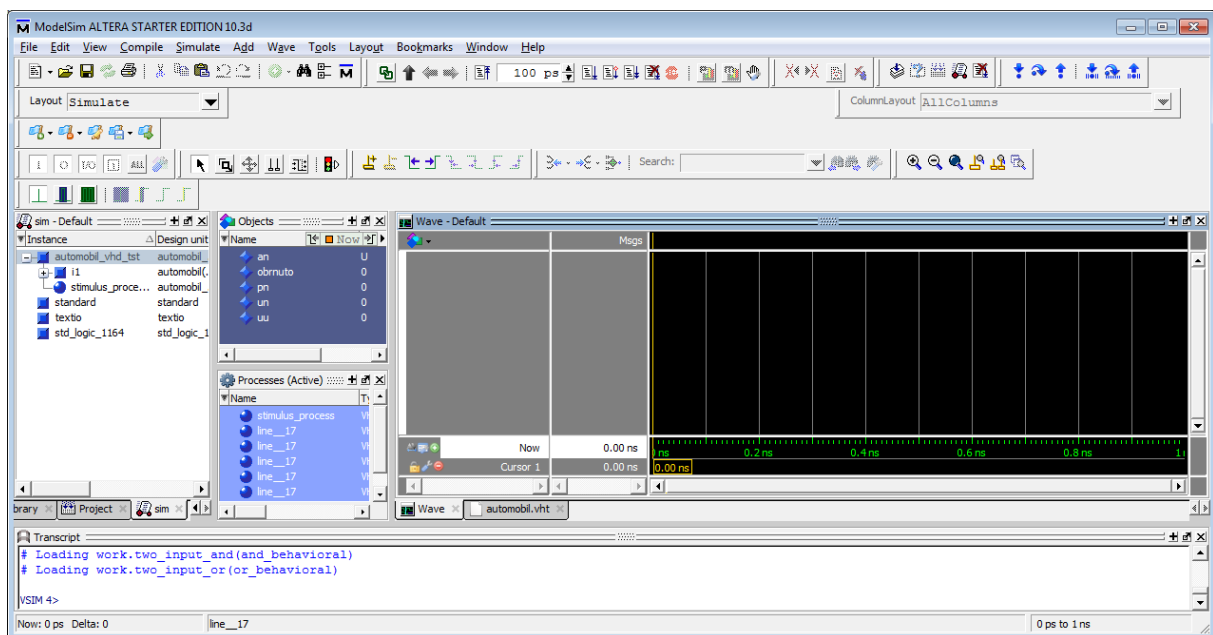
Fajl ima ekstenziju *.vht* što ne treba da zbuni jer se potpuno isto ponaša kao i *.vhd* fajlovi za testiranje. U njemu je već instancirana komponenta koja se testira (*DUT*) i definisani pobudni i izlazni signali. Definisana su i dva procesa koji se koriste za pisanje stimulusa. Na korisniku je da samo definiše kada koji signali menjaju vrednost. Na primer jedan proces se može koristiti za generisanje signala takta, dok se u drugom mogu dodeljivati vrednosti ostalim signalima.

Editovanje se može raditi u bilo kom tekstualnom editoru, ali zbog specifične ekstenzije, najbolje je editovati fajl u okviru ModelSim simulatora, tj. odmah po generisanju *.vht* fajla, pokrenuti ModelSim simulator na isti način kao što je to ranije objašnjeno i otvoriti fajl izborom opcije *File*→*Open*.

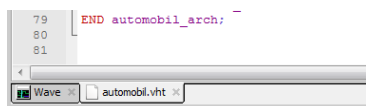
Nakon editovanja fajla, isto kao i u prethodnom slučaju, potrebno je kompajlirati ga, izborom opcije *Compile*→*Compile*. U biblioteci *work* se sada pojavljuje modul *ime\_modula\_koji\_se\_testira\_vhd\_tst* i njegova simulacija se pokreće na isti način kako je i ranije opisano.

## Simulacija u ModelSim okruženju

Nakon pokretanja simulacije, dobija se okruženje kao na slici 16 (kliknuti na karticu *Wave* da se prikaže prozor za prikaz signala (slika 17)).

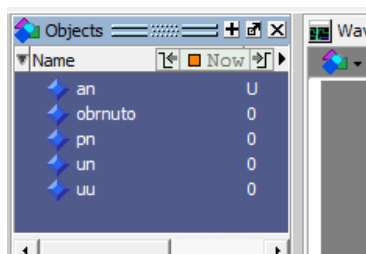


Slika 16



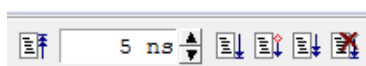
Slika 17

U prozoru **Objects** nalaze se svi signali koji se testiraju (slika 18). Prevlačenjem signala u prozor **Wave** odabira se koji se signali posmatraju.



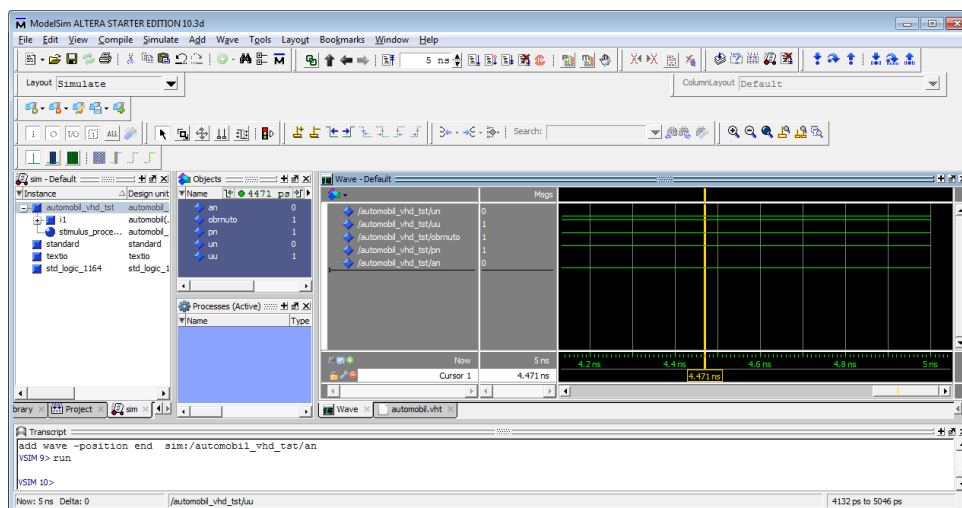
Slika 18

Trajanje simulacije se upisuje u prozor prikazan na slici 19, a simulacija se pokreće klikom na dugme **Run** sa desne strane prozora za zadavanje vremena. Restovanje simulacije se vrši klikom na dugme **Restart** sa leve strane prozora za zadavanje vremena.



Slika 19

Nakon pokretanja simulacije dobija se prozor kao na slici 20.



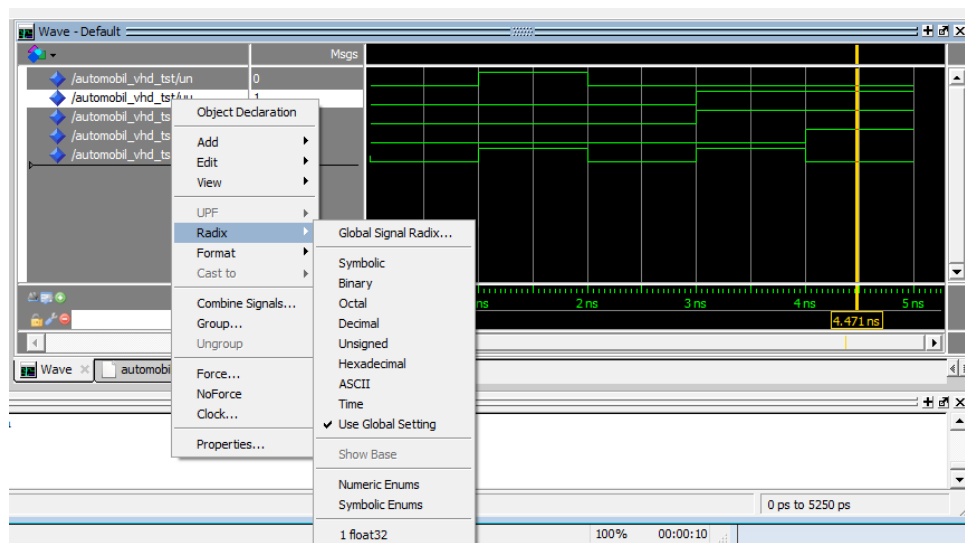
Slika 20

Prikaz vremenskog opsega signala se menja klikom na neko od dugmića sa slike 21.



Slika 21

U ovom primeru, to nije slučaj, ali ako se posmatraju vektori signala, onda je zgodno odabrati format u kome se vektori prikazuju: decimalni, heksadecimalni, binarni... To je moguće uraditi odabirom opcija kao na slici 22.



Slika 22

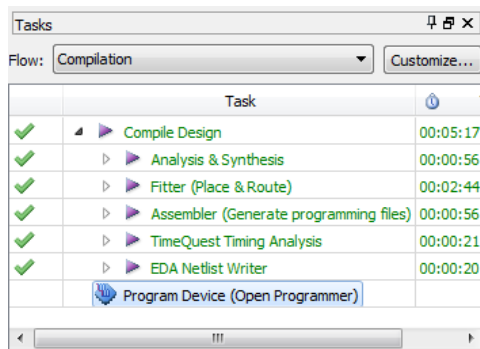
## Testiranje na *DE1-Soc* razvojnoj ploči

Nakon uspešnog kreiranja i testiranja celog dizajna potrebno je povezati pinove koji se nalaze u interfejsu projektovanog finalnog sistema sa pinovima na FPGA čipu. Ovo se najlakše može uraditi pomocu TCL skripte.

Na stranici [tnt.etf.rs/~oe4upv/lab/pin\\_assign\\_de1\\_soc.tcl](http://tnt.etf.rs/~oe4upv/lab/pin_assign_de1_soc.tcl) je data kompletna TCL skripta sa svim izdefinisanim pinovima na *DE1-Soc* razvojnoj ploči. TCL skripta se može otvoriti pomoću bilo kog tekstualnog editora. Potrebno je proveriti da li se imena signala u skripti slažu sa imenima signala u projektovanom finalnom sistemu, a ako to nije slučaj potrebno ih je uskladiti. Kada su nazivi signala u skripti i interfejsu sistema uskladeni može se pristupiti povezivanju pinova. Skripta se pokrece tako što se u glavnom Quartus II prozoru pokrene komanda **Tools**→**Tcl Scripts...** i odabere odgovarajući .tcl fajl, a sama skripta se pokrece komandom **Run**.

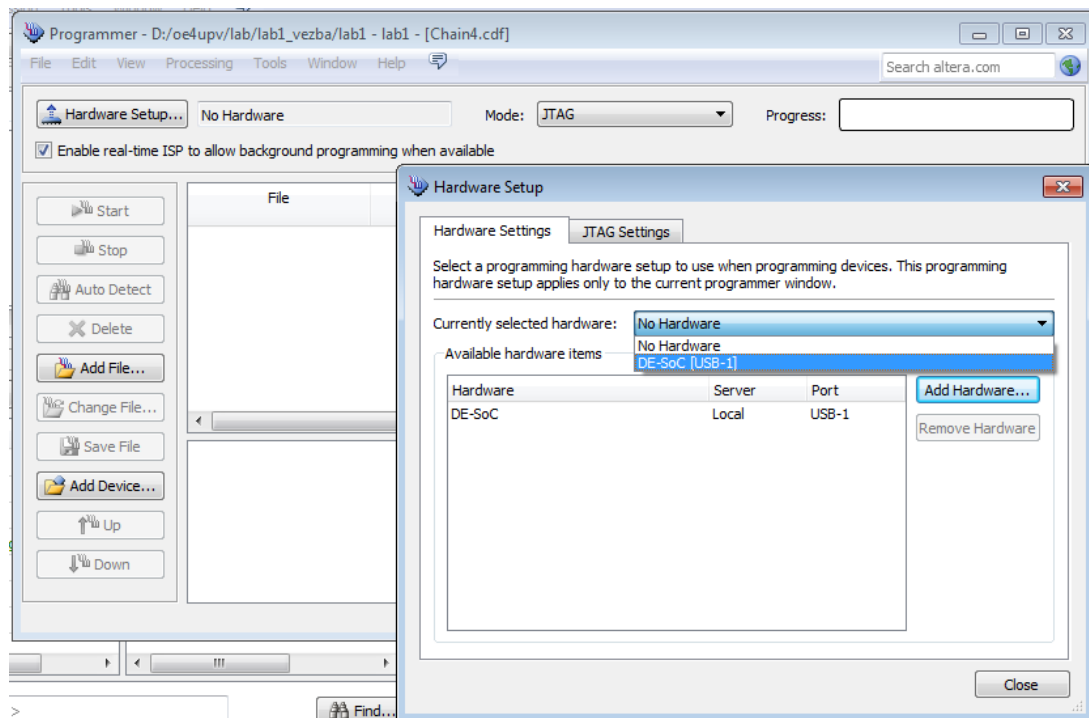
Nakon izvršenja TCL skripte, potrebno je kompajlirati sistem u potpunosti, a ne samo izvršiti korak analize i sinteze. Kompajliranje se pokrece komandom **Processing**→**Start Compilation** i posle nekog vremena i uz malo sreće sve se završava bez grešaka. Osnovni proizvod kompajliranja je .sof fajl koji služi za programiranje FPGA čipa i nalazi se u direktorijumu **output\_files**.

Na kraju je potrebno pokrenuti alat za programiranje FPGA čipa klikom na prečicu **Program Device (Open Programmer)** u odeljku za kompajliranje prikazanom na slici 23. Ovim se otvara prozor programatora.



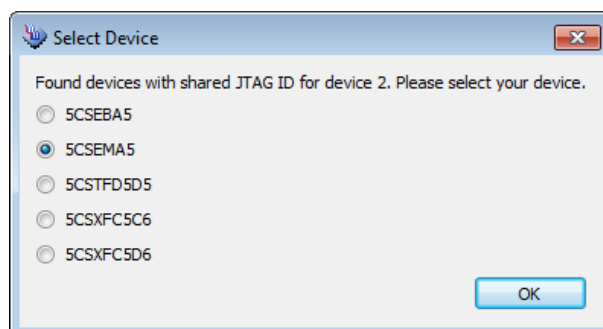
Slika 23

Potrebno je podesiti port na koji je povezan uređaj koji se programira klikom na dugme Hardware Setup. Otvara se prozor kao sa slike 24 u kome za hardver treba odabrati DE1-SoC [USB-1].



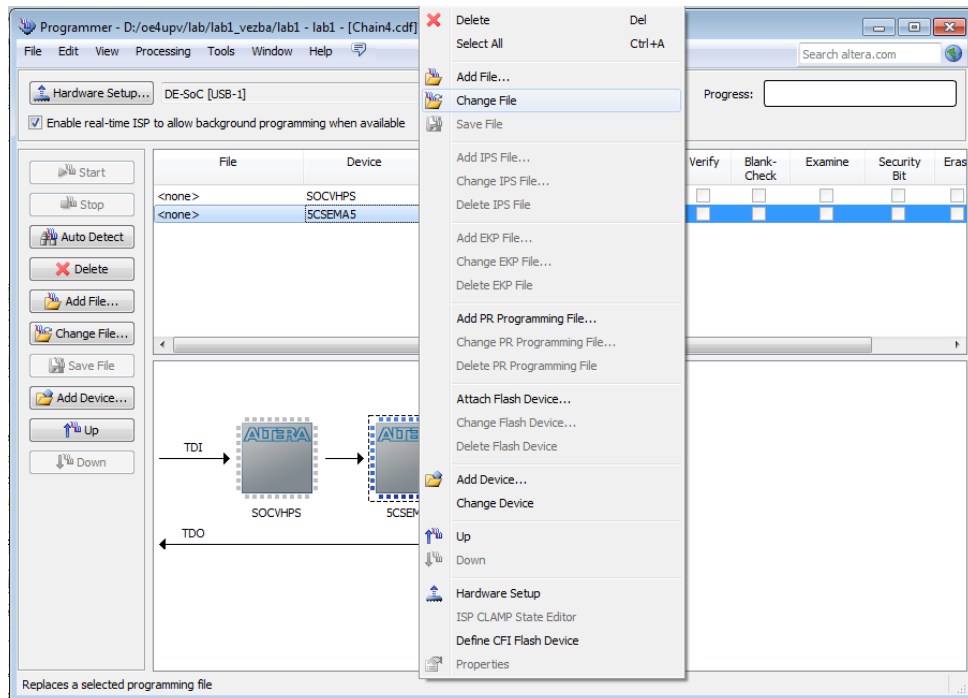
Slika 24

Nakon odabira uređaja koji se programira, potrebno je kliknuti na dugme Auto Detect kojim se otvara prozor sa slike 25 u kome treba odabrati 5CSEMA5.

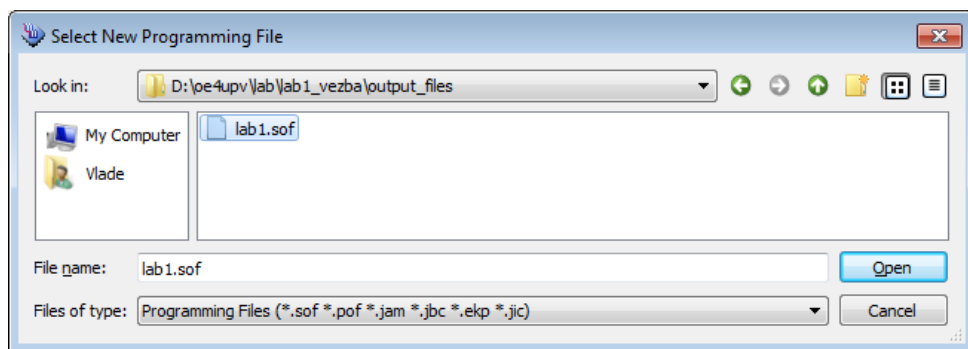


Slika 25

Sledeća stvar koju treba podesiti je učitavanje odgovarajućeg .sof fajla. Desnim klikom na automatski detektovani fajl koji odgovara uređaju 5CSEMA5 i odabirom opcije **Change File** (slika 26) otvara se prozor u kome treba odabrati .sof fajl kreiran kompilacijom dizajna koji se želi testirati na ploči (slika 27).

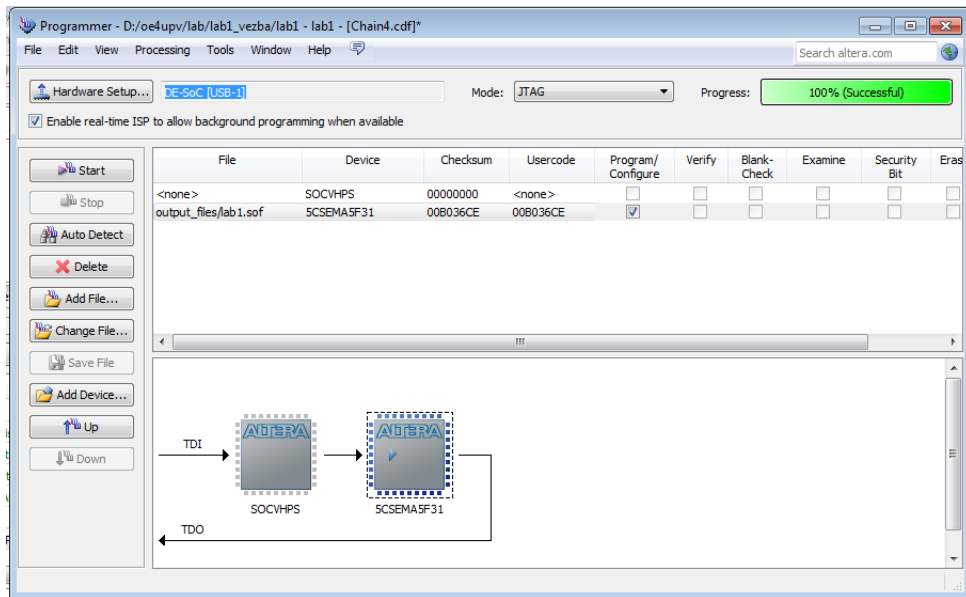


Slika 26



Slika 27

Važno je da opcija **Program/Configure** bude izabrana za .sof fajl koji služi za programiranje FPGA (slika 28). Programiranje se zapocinje klikom na taster **Start**. U gornjem desnom uglu se prikazuje progres procesa programiranja i kada dođe do 100% FPGA čip je u potpunosti isprogramiran.



Slika 28

## Dodatna uputstva

Ovim su pokrivena neke osnovne funkcionalnosti Altera Quartus II i Altera ModelSim alata. Ostatak će biti objašnjen na laboratorijskim vežbama, a moguće i u narednim verzijama uputstva.

Ukoliko postoji neki problem, ili su potrebna dodatna uputstva, javiti se na [oe4upv@el.etf.rs](mailto:oe4upv@el.etf.rs).