

**Uputstvo za rad sa Xilinx ISE 9.2i programskim  
paketom – projektovanje korišćenjem  
Schematic Editora**

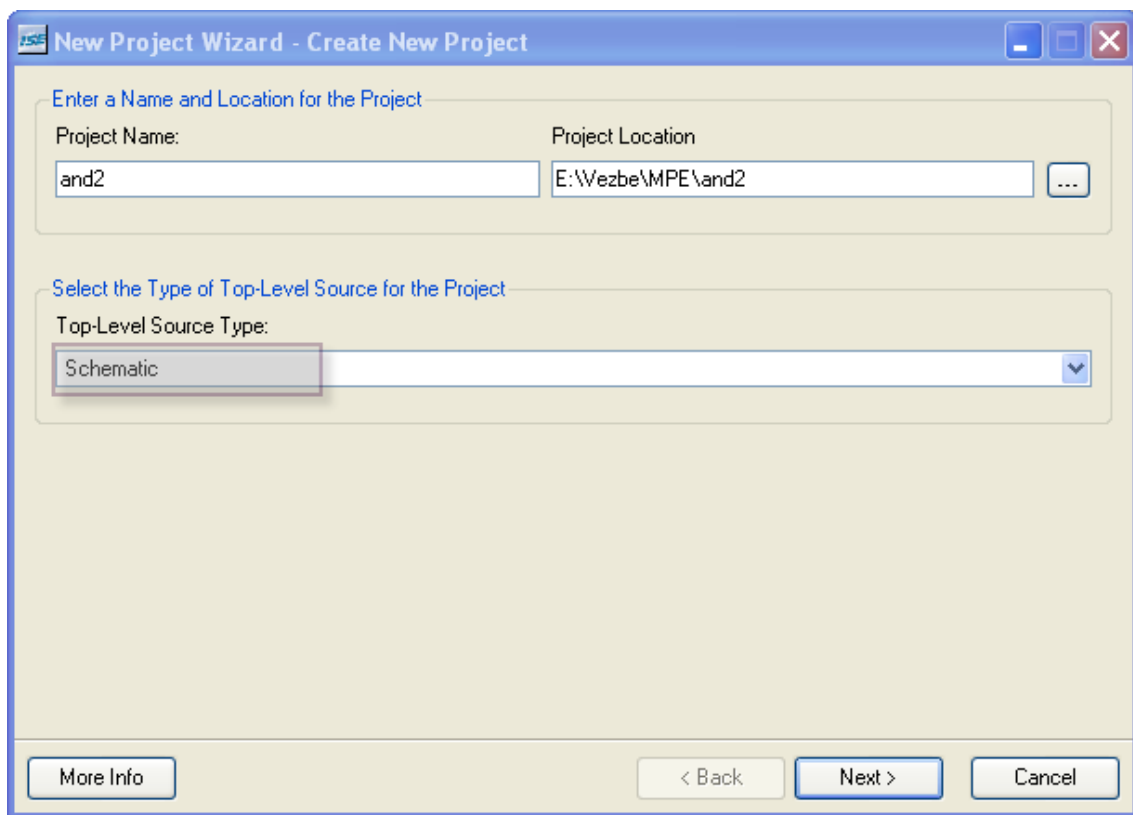
**Novi Sad, novembar 2007.**

## Startovanje programskog paketa

Xilinx Foundation ISE 9.2i programski paket se aktivira dvoklikom na Project navigator ikonicu na desktopu ili aktiviranjem u Start meniju Windowsa iste opcije.

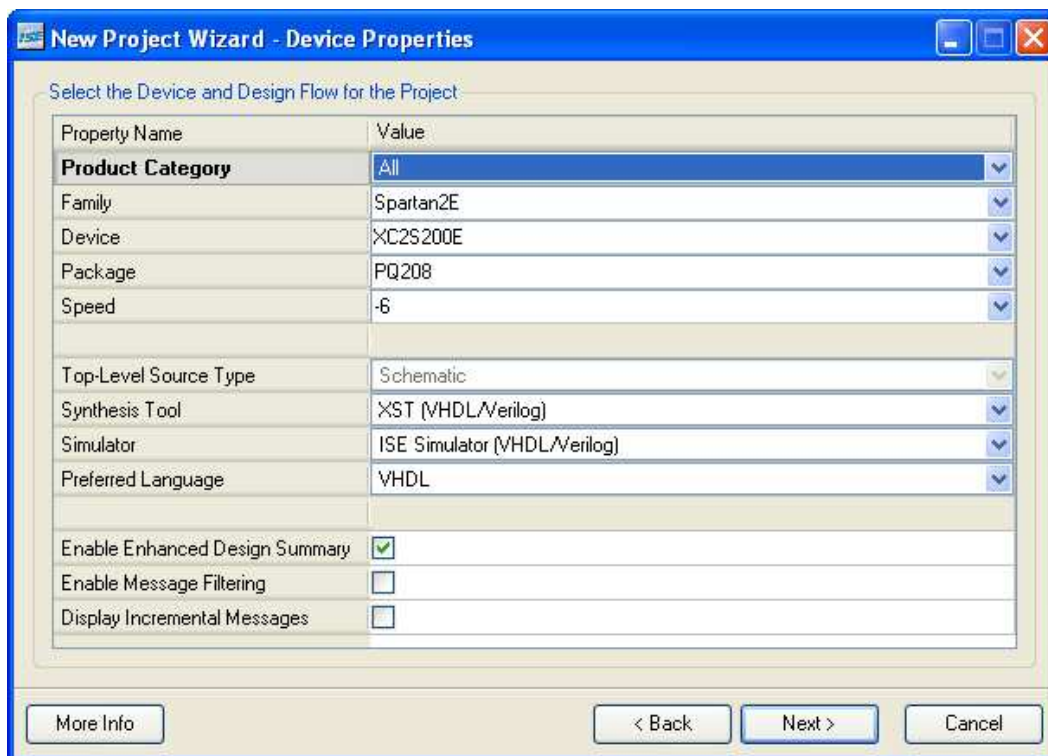
## Kreiranje novog projekta

Kreiranje novog projekta počinje nakon odabira opcije *File* pa *New project* u meniju. Otvara se prozor (slika 1) u kojem treba popuniti naziv novog projekta, lokaciju gde će biti snimljen na disku i tip glavnog modula (mi ćemo raditi samo projektovanje uz pomoć Schematic Editora).



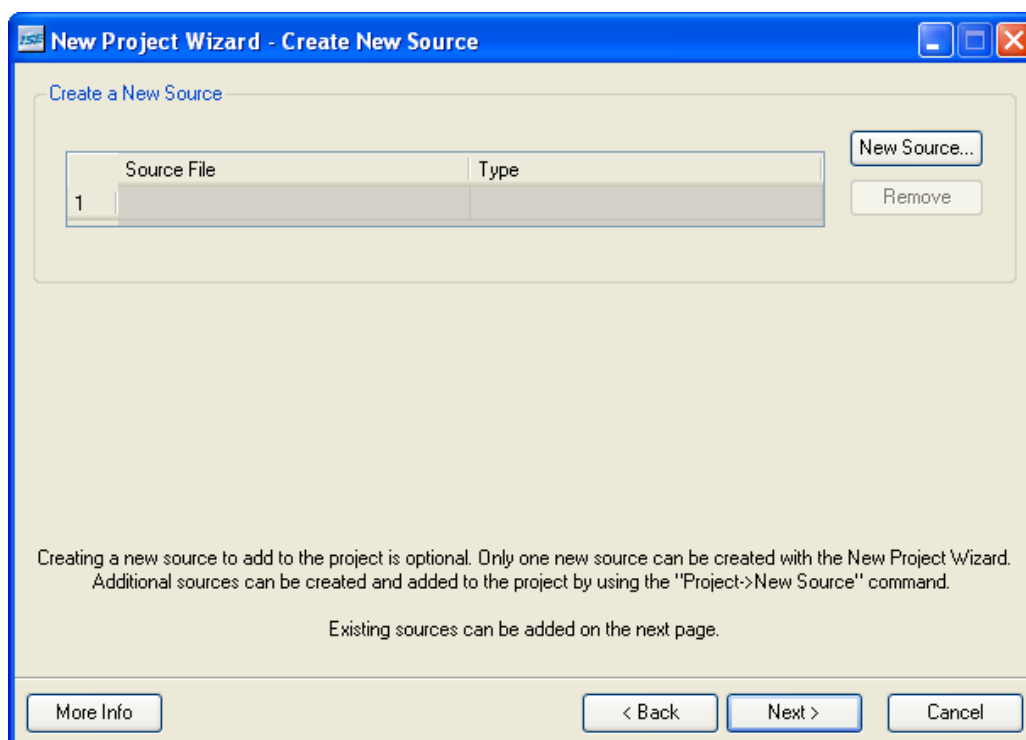
Slika 1: Kreiranje novog projekta

Nakon što smo uneli potrebne podatke (npr. and2), pritiskom na *Next* se otvara novi prozor (slika 2) u kome treba odabrati podatke o Xilinx komponenti koja će se koristiti. Treba podesiti familiju (Spartan2), naziv komponente (xc2s200e), kućište (pq208) i brzinu (-6). Pored toga treba odabrati i softverske alate. Mi ćemo koristiti Xilinxov softver za sintezu i za simulaciju (moguće je korišćenje drugih alata za simulaciju, npr Mentor Graphics-ov ModelSim). Sve će biti rađeno u Schematic Editoru, ali prilikom sinteze Xilinx-ovi alati će to pretvoriti u odgovarajuće VHDL fajlove.



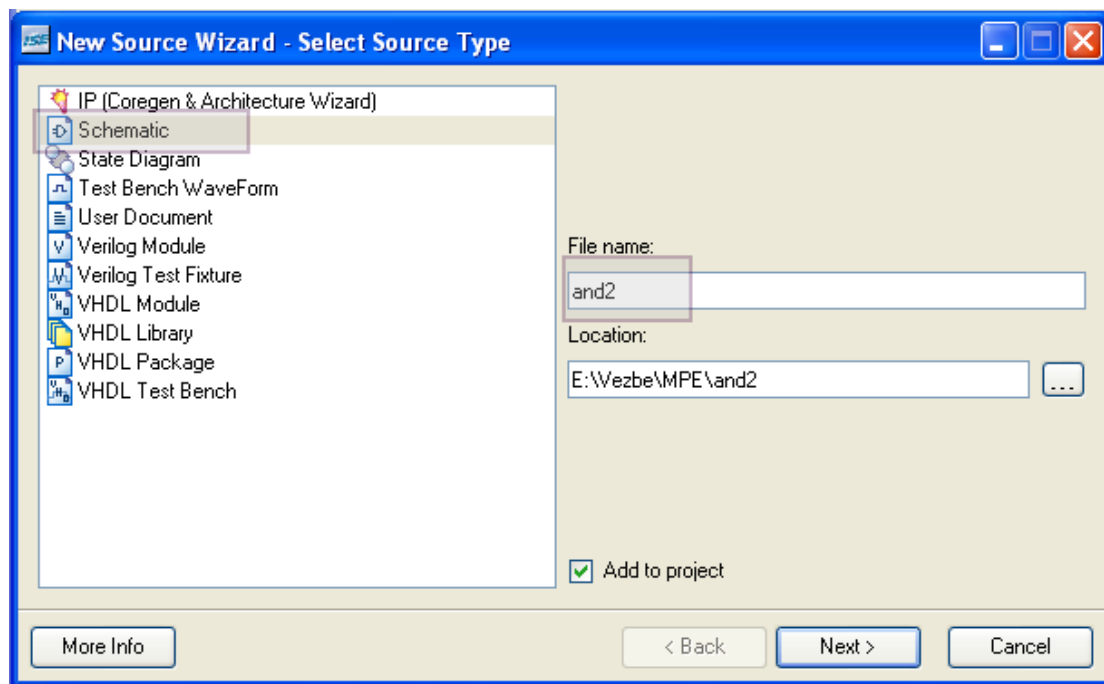
Slika 2: Podešavanje parametara

Nakon pritiska na taster *Next* pojavljuje se prozor (slika 3) koji omogućava kreiranje novog izvornog fajla.



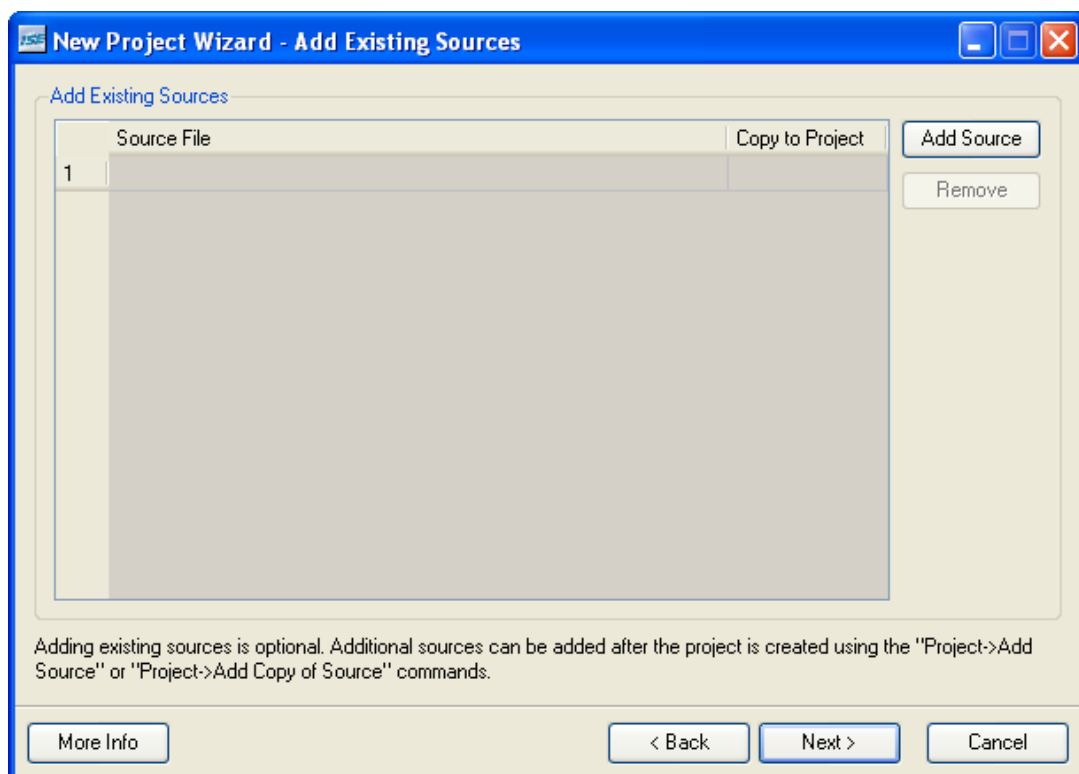
Slika 3: Kreiranje izvornog fajla

Pritiskom na *New Source* se otvara prozor (slika 4) za definisanje vrste novog izvornog programa, kao i dodelu imena. Recimo da ćemo ga nazvati isto primer1 kao i folder u kojem će biti snimljeni svi fajlovi u vezi sa ovim projektom. Za tip izvornog programa treba odabrati *Schematic* opciju.



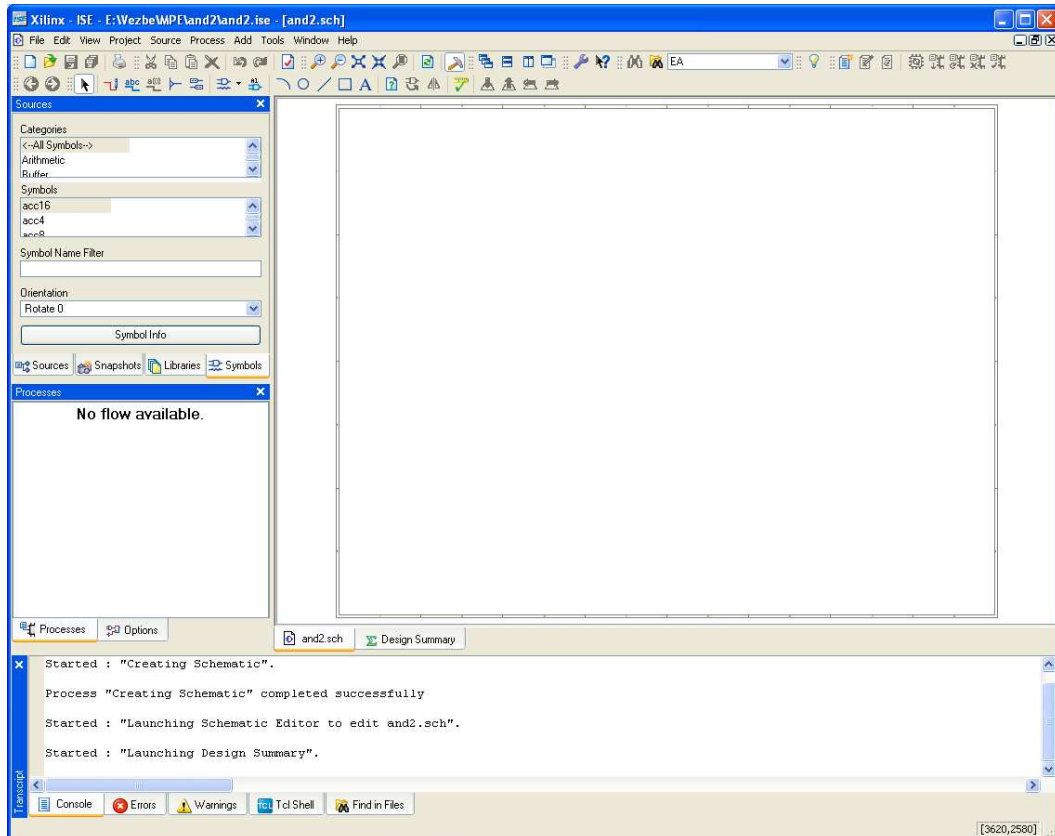
Slika 4: Izbor tipa izvornog programa

Nakon pritiska na *Next* se pojavljuje prozor sa informacijama o novom programu. Pritiskom na *Finish* se kreira novi *schematic* fajl koji se pojavljuje u prozoru odakle smo počeli kreiranje fajla (kao na slici 4). Pritiskom na *Next* se otvara novi prozor (slika 5) u kome možemo da dodamo neke već postojeće izvorne programe.



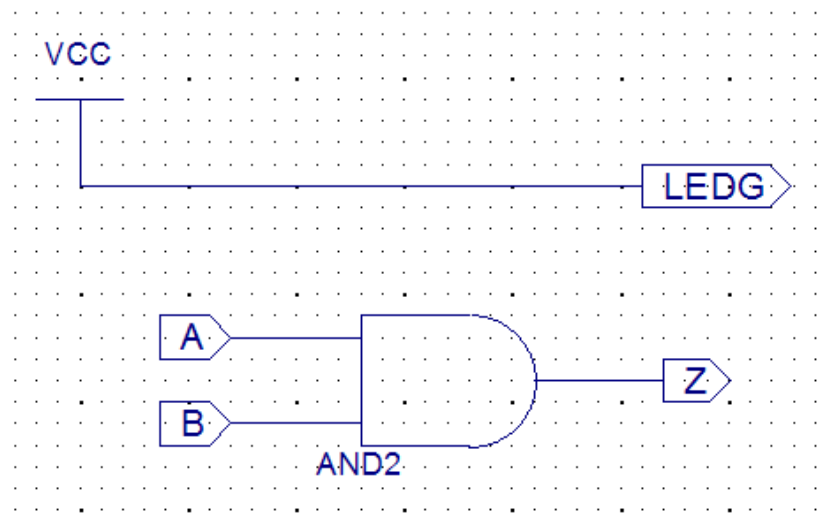
Slika 5: Dodavanje postojećih izvornih programa

Potom pritiskom na *Next* dobijamo prozor sa informacijama o novom projektu. Konačno, pritiskom na *Finish* smo završili postupak kreiranja novog projekta (slika 6).



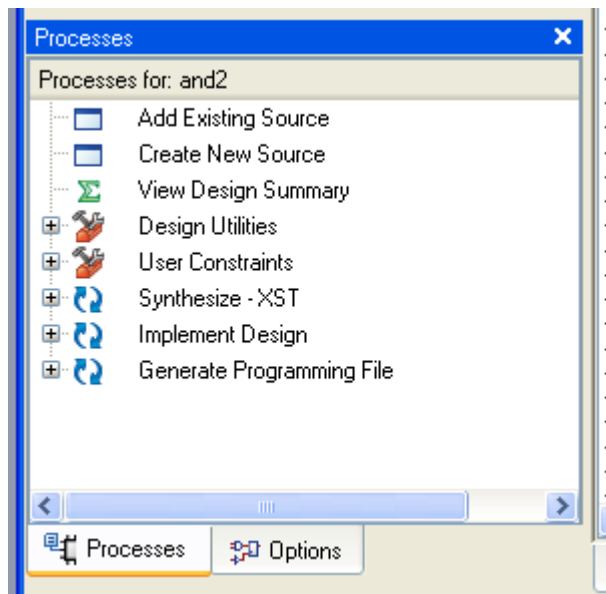
Slika 6: Izgled novog projekta

Pre bilo kakvog daljeg rada treba dopuniti šemu, korišćenjem odgovarajućih komponenti iz biblioteka, alata za crtanje linija i po potrebi drugih alata, tako da radi nešto konkretno. Sve što je potrebno nalazi se ili na paleti sa alatima ili u okviru dela prozora za pretraživanje komponenata. Kao najjednostavniji primer uzećemo da su  $a$  i  $b$  ulazi dvoulaznog I kola, a  $z$  izlaz. Pošto ćemo  $a$  i  $b$  ulaze kasnije vezivati na prekidače, izlaz  $z$  na LED moramo dodati signal dozvole za rad ledovki, LEDG, i postaviti ga na logičku 1 čime dozvoljavamo rad ledovki. Dakle, za ovaj projekat potrebno nam je jedno dvoulazno I kolo (AND2), 4 ulazno/izlazna markera (I/O marker) i jedan priključak za napajanje. Na dva ulazno/izlazna markera ćemo dovoditi ulazne signale, na jednom ulazno/izlaznom markeru će biti izlazni signal I kola i jedan ulazno/izlazni marker ćemo povezati sa signalom za dozvolu rada ledovki. Dopolnjena šema izgleda kao na slici 7.



Slika 7: Šema nacrtana u ISE editoru

U prozoru u kojem se aktiviraju procesi (slika 8) odabrati *Synthesize – XST*, pa opciju *View RTL Schematic* koja vrši sintaksnu proveru koda i kreira elektronsku šemu na osnovu zadate šeme. Ova opcija je veoma zgodna da se proveri da li je Xilinx softver napravio hardver onako kako smo zamislili ili su potrebne korekcije.



Slika 8: Proces u projektovanju

## Simulacija

Da bismo pojednostavili objašnjenje simulacije zamislimo situaciju realnog merenja nekog digitalnog sistema. Ovakav merni sistem bi se sastojao od:

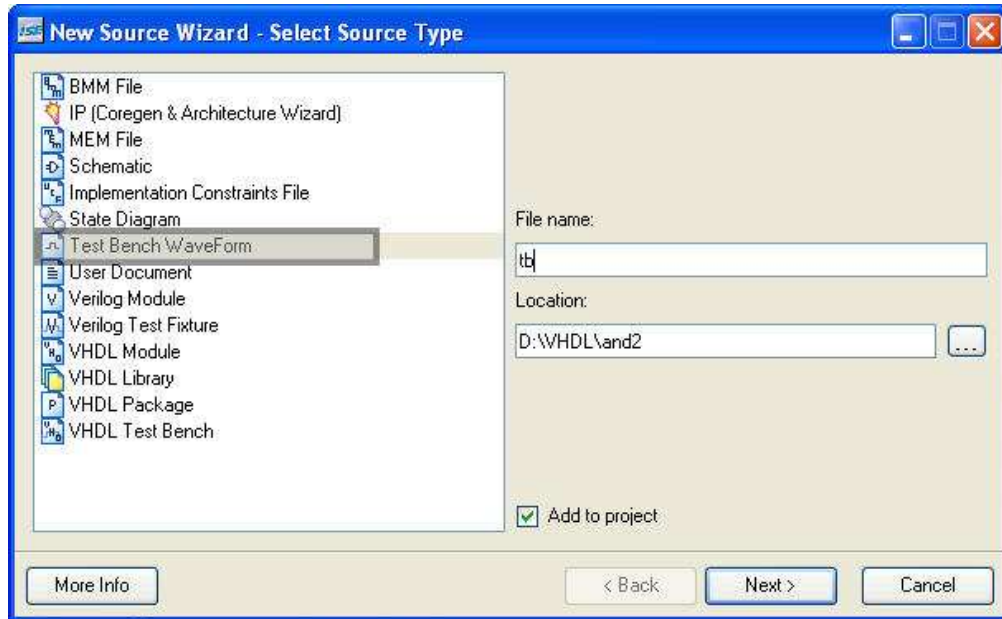
- 1) uređaja koji testiramo
- 2) generatora ulaznih signala
- 3) mernih instrumenata kojima merimo i posmatramo talasne oblike izlaznih signala

U Xilinx ISE-u imamo potpunu analogiju sa gore pomenutim mernim sistemom. Uređaj koji testiramo bila bi šema koju smo nacrtali i koja opisuje funkcionisanje digitalnog sistema koji simuliramo. Generator ulaznih signala se realizuje u posebnom fajlu (*Testbench Wave Form*) gde se definišu i generišu svi potrebni ulazni signali, a merni instrument nam je simulator u kome posmatramo talasne oblike izlaznih signala. Test signali *a* i *b* u test programu se definišu vizuelno, odnosno prilikom dodavanja novog *Testbench Wave Form-a* u projekat. Signale ćemo posmatrati unutar ISE simulatora koji će nam predstavljati neku vrstu osciloskopa.

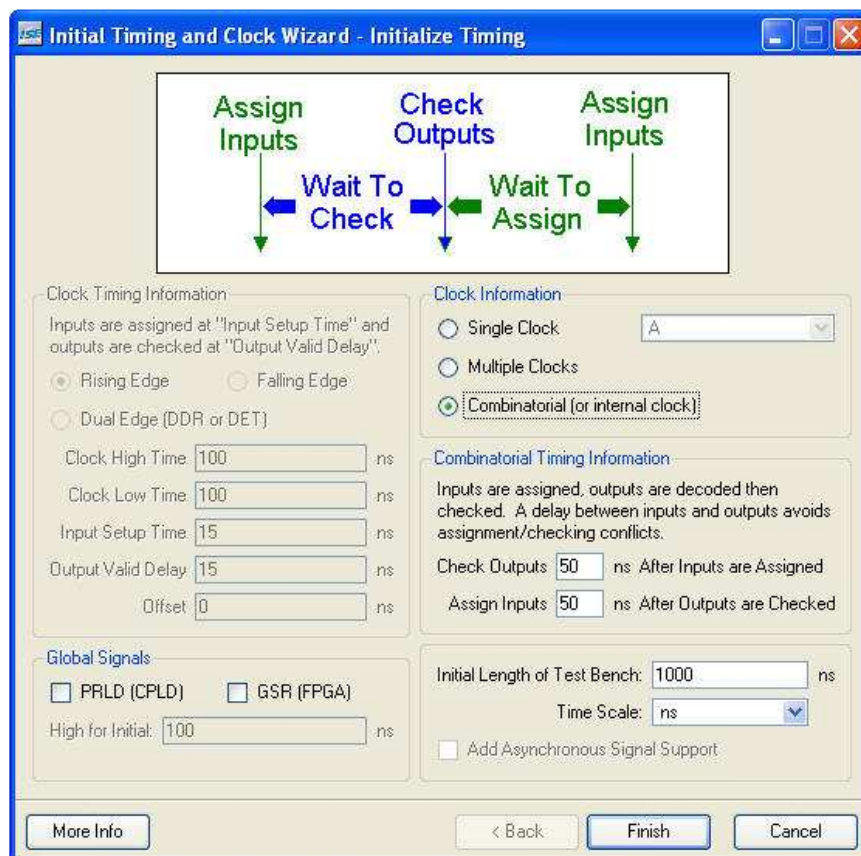
Kako sve to uraditi u Xilinx Foundation ISE-u?

Potrebno je dodati u projekat *Testbench Wave Form* fajl koji ćemo nazvati *tb*. To je program u kome ćemo zadati ulazne test signale – test vektore.

Da bi smo dodali *tb* program, treba odabrati opciju *Project* iz menija i potom *New Source*. Ovim dobijamo prozor koji smo već videli na slici 4, ali ćemo ovog puta za tip fajla odabrati *Test Bench WaveForm* (slika 9) i daćemo ime programu *tb*.

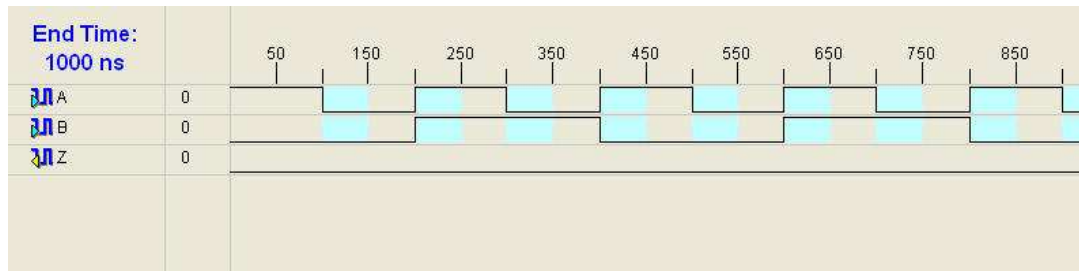
Slika 9: Dodavanje *Test Bench WaveForm* fajla

Nakon prisika na *Next* otvara se prozor za podešavanje takta. Pošto je ovo obično kombinaciono kolo onda se to i odabere u okviru ovog prozora. Ovde se može podesiti i inicijalno trajanje simulacije (ono je postavljeno na 1000 ns).



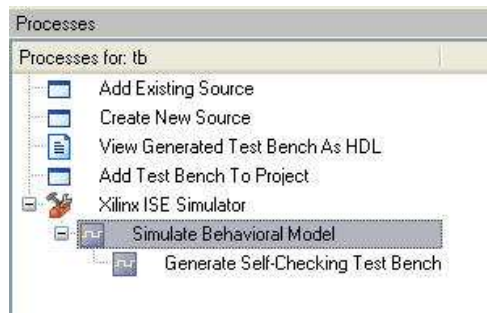
Slika 10: Podešavanje takt signal i trajanja simulacije

Nakon završetka ovih podešavanja, otvara se *tbw* fajl u kojem se mogu grafičkim putem podesiti A i B ulazni signali kao na slici. Podešavanje se vrši na veoma jednostavan način, klikom na mesto gde želimo da imamo tranziciju određenog signala (zelene površine). Izgled dela prozora nakon ove operacije je dat na slici 11.



Slika 11: Deo glavnog prozora u kojem se podešava oblik signala

i nakon toga odabirom *Simulate Behavioral Model* možemo da počnemo simulaciju.



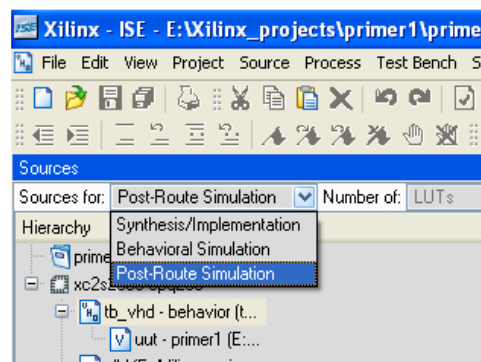
Slika 12: Pokretanje simulacije

Nakon aktiviranja, skoro sva podešavanja urađena su automatski. Jedino po potrebi treba promeniti vreme trajanja simulacije (ono je postavljeno na 1000 ns). Rezultate simulacije vidimo na slici 13.



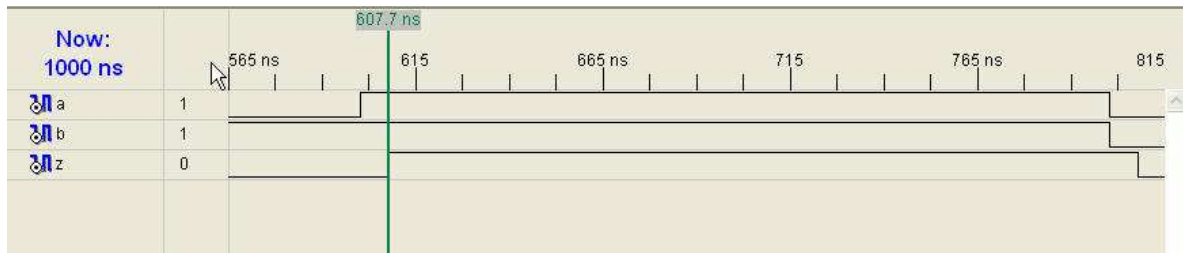
Slika 13: Rezultati simulacije

Ovim smo dobili rezultate osnovne funkcionalne ili tzv. *bihevijalne* simulacije. Ukoliko želimo da dobijemo preciznije rezultate onda ćemo raditi *Post Route* simulaciju. Odabir sa kojim izvornim fajlovima radimo se vrši u combo box-u prikazanom na slici 14. Ona se aktivira isto kao i prethodna simulacija (s tim da je opcija *Simulate Post Place and Route Model*). Uvećani rezultat na slici 15 nam pokazuje da je kašnjenje izlaza za ulazima oko 7 ns što se u osnovnoj simulaciji ne vidi.



Slika 14: Izbor *Post-Route* simulacije

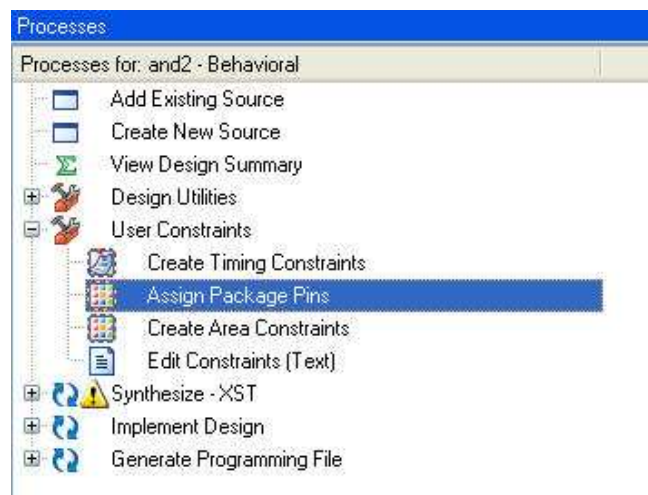




Slika 15: Uvećan rezultat *post-route* simulacije

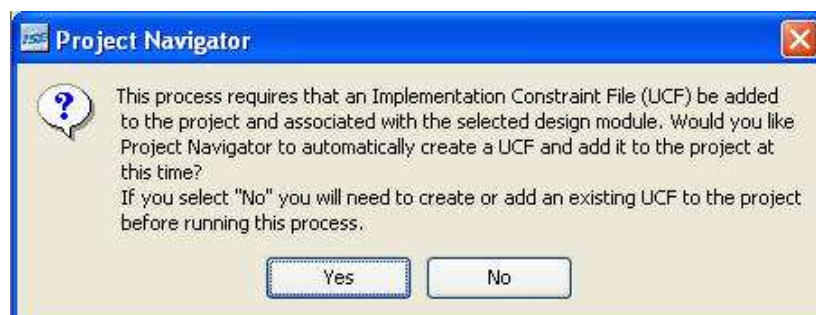
## Povezivanje dizajna sa pinovima na FPGA kolu

Da bismo mogli da proverimo da li dizajnirani sistem zaista radi na FPGA kolu (pored simulacije) potrebno je da odgovarajuće portove našeg dizajna povežemo sa pinovima na FPGA kolu. Taj proces započinjemo pozivom User Constraints Editora, dvostrukim klikom na natpis Assign Package Pins



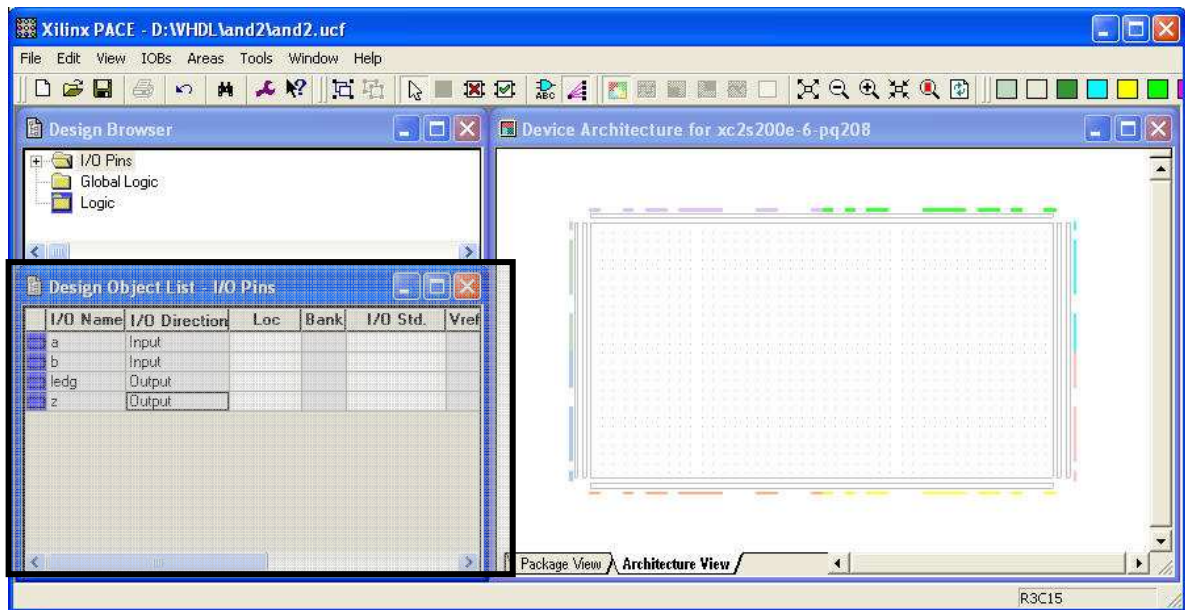
Slika 16: Proces kojim se započinje dodela pinova dizajnu

Xilinx ISE nas tada obaveštava da će pinove FPGA kola koje budemo dodelili našem dizajnu smestiti u poseban fajl koji bi trebalo da dodamo odgovarajućem modulu u dizajnu. Pitanje koje nam postavlja je da li želimo da on to automatski doda projektu nakon kreiranja.



Slika 17: Da li želimo da se neki procesi automatski obave?

Potom se otvara program Xilinx PACE programa kojim se vrši dodela pinova FPGA kola.



Slika 18: Xilinx PACE

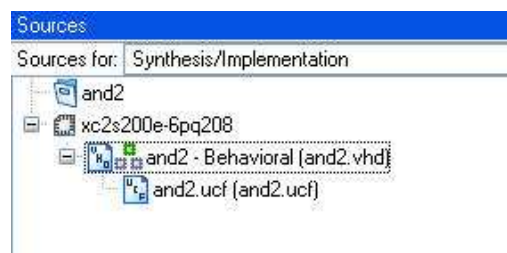
Deo koji je nama od interesa je *Design Object List – I/O Pins* (označen kvadratom na slici 18). U ovom delu vidimo portove našeg dizajna koje je potrebno povezati sa odgovarajućim pinovima. Povezivanje se vrši tako što u koloni *Loc* kod odgovarajućeg porta upišemo broj pina. Popunjena tabela je prikazana slici 19.

I/O Name	I/O Direction	Loc	Bank	I/O Std.	Vref
a	Input	p71	BANK		
b	Input	p69	BANK		
ledg	Output	p88	BANK		
z	Output	p111	BANK		

Slika 19: Pinovi dodeljeni dizajnu

Tabela na slici 19 je popunjena korišćenjem tabele u prilogu koja nam govori kojem pinu FPGA kola odgovara koja periferija na kartici DIO4 ako se kartica priključi na A1 – A2 konektore. Konkretno u ovom slučaju imamo dvoulazno I kolo koje je na ulazu povezano sa dva prekidača (SW1 i SW2), a na izlazu sa LED1, LEDG signal dozvoljava rad svetlećih dioda i on mora biti na logičkoj 1 da bi diode radile (zbog toga je i dodat u dizajn).

Ukoliko je proces dodele pinova uspešno završen odgovarajućem modulu (and2) će biti dodeljen odgovarajući fajl (kako je to prikazano na slici 20)



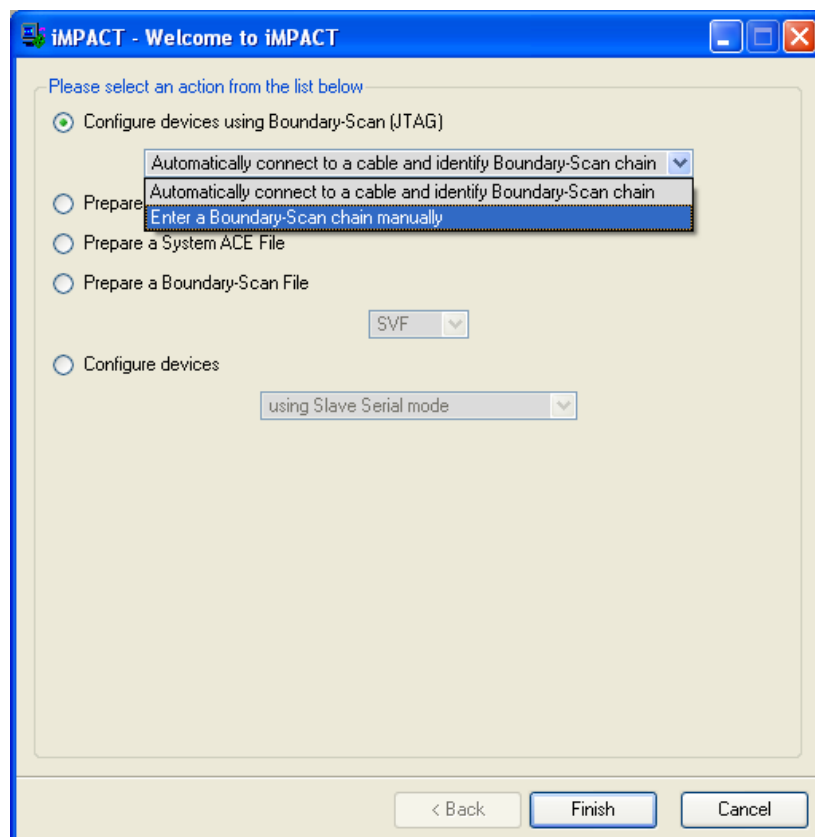
Slika 20: Proces dodele pinova dizajnu uspešan

## Programiranje FPGA

Nakon uspešno urađene simulacije i implementacije, poslednji korak u projektovanju je implementacija dizajna na realnom hardveru. U našem slučaju, reč je o Xilinx razvojnom sistemu (Digilab).

Da bismo mogli da programiramo FPGA čipa na ovom razvojnom sistemu, potrebno je aktivirati proces koji to radi. U prozoru gde se pokreću procesi otvoriti *Generate Programming File* (videti sliku 9) pa potom *Configure Device* koji će aktivirati deo programskom paketa koji je zadužen za programiranje FPGA čipa. U prvom prozoru koji se otvori (slika 21) odabrati *Enter a Boundary-Scan chain manually* i pritisnuti *Finish*. U prozoru desnim klikom odabrati opciju *Add Xilinx Device* nakon koje se otvara fajl dijalog prozor gde bi trebalo da se nalazi fajl sa ekstenzijom *.bit* i imenom pod kojim je i projekat snimljen. Otvoriti taj fajl dvoklikom ili selekcijom uz pritisak na *Open*. (slika 22)

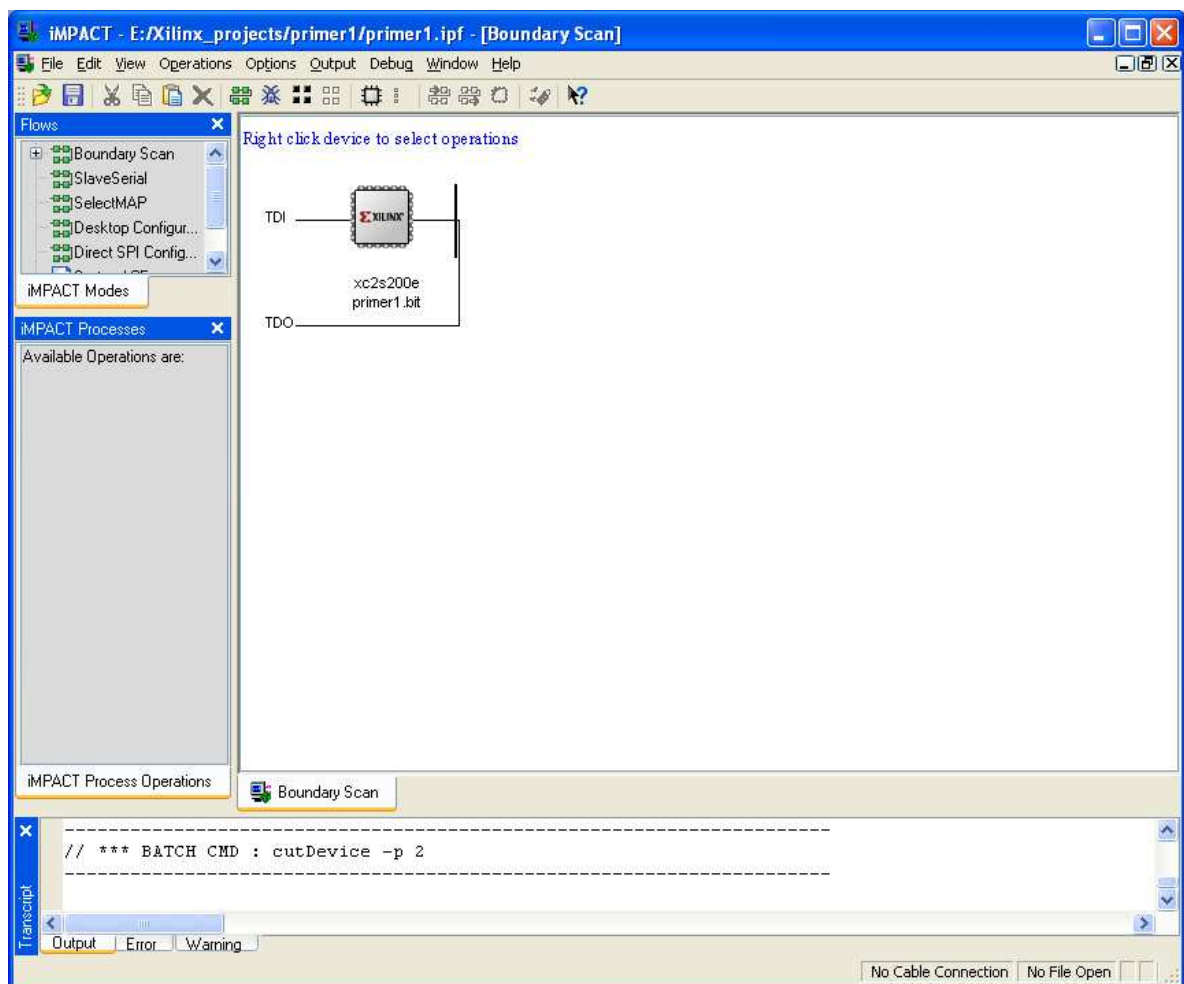
Upozorenje koje se pojavi zatvoriti pritiskom na *OK*. Nakon ovoga je urađena inicijalizacija Boundary-Scan Moda i sada desnim klikom na kvadrat koji simboliše FPGA čip (slika 23) odabrati Program ociju iz pop-up menija. Pritiskom na *OK*, počinje proces prvo povezivanja programa sa programatorom, a potom i samo programiranje čipa. Uspešan proces programiranja je označen plavom porukom *Programming Succeeded*. U suprotnom pojavljuje se crvena poruka *Programming Failed* i tada je potrebno ponoviti ceo postupak ponovo.



Slika 21: Početak programiranja FPGA čipa



Slika 22: Otvaranje bit fajla



Slika 23: Prozor nakon inicijalizacije Boundary-Scan Moda

**Prilog 1: Veza pinova na ekspanzionim karticama sa sa pinovima Spartan2E FPGA kola u slučaju kada se povezuju na B1-B2 ekspanzioni konektor D2-SB kartice i A-B ekspanzioni konektor D2E kartice**

Pin na ekspanzionoj kartici	FPGA pin na D2E kartici	FPGA pin na D2-SB kartici	Opis funkcije
BTN1	p40	p47	Tasteri
BTN2	p41	p46	
BTN3	p42	p86	
BTN4	p43	p84	
BTN5	p64	p83	
SW1	p16	p71	Prekidači
SW2	p18	p69	
SW3	p21	p64	
SW4	p23	p62	
SW5	p27	p60	
SW6	p30	p58	
SW7	p33	p56	
SW8	p35	p49	
LED1	p44	p111	Ledovke
LED2	p46	p109	
LED3	p48	p102	
LED4	p55	p100	
LED5	p57	p98	
LED6	p59	p96	
LED7	p61	p94	
LED8	p63	p89	
LEDG	p68	p88	Dozvola ledovki
SDP	p36	p48	Decimalna tačka
S0	p17	p70	Segmenti
S1	p20	p68	
S2	p22	p63	
S3	p24	p61	
S4	p29	p59	
S5	p31	p57	
S6	p34	p55	
AN1	p45	p82	Selekcija cifre na 7. segmentnom displeju
AN2	p47	p81	
AN3	p49	p75	
AN4	p56	p74	

## **Prilog 2: Veza pinova Spartan2E kola sa perifernim uređajima na razvojnoj kartici**

Periferija na razvojnoj kartici	D2E kartica	D2-SB kartica	Opis funkcije
CLK	p80	p182	Glavni takt
LED1	p69	p154	Kontrolna LED
BTN1	p77	p187	Taster