POKAZNA VEŽBA 1

Osnovi projektovanja digitalnih sistema na nivou logičkih kola

Potrebno predznanje

- Osnovno poznavanje digitalne elektronike
- Bulova (Boolean) algebra

Šta će biti naučeno tokom izrade vežbe?

Nakon urađene vežbe, bićete u mogućnosti da:

- razumete razna digitalna logička kola i njihovu funkciju
- projektujete sistem koji računa proizvoljnu Boolean funkciju pomoću logičkih kola
- opišete sistem sastavljen od digitalnih logičkih kola pomoću logičke šeme
- koristite Intel Quartus za opis i sintezu digitalnih sistema.
- sintetišete i implementirate vaš sistem za FPGA integrisano kolo pomoću Intel Quartus alata

Sa znanjem dobijenim u ovoj vežbi, bićete spremni da projektujete složenije sisteme sastavljene od digitalnih logičkih kola, tzv. kombinacione mreže, o kojima će biti više reči u narednoj vežbi. Znanje dobijeno u ovoj vežbi je važno za pravljenje mosta između digitalnog matematičkog sveta Bulove algebre i fizičkog sveta digitalne elektronike. Osnovno znanje dobijeno u ovoj vežbi omogućiće vam da razumete složene digitalne sisteme koje ćete projektovati tokom predmeta.

Apstrakt i motivacija

U današnje vreme, mnogi uređaji koji nas okružuju su digitalni. Ovi uređaji čitav svoj rad zasnivaju na operacijama unutar skupa od samo dve vrednosti (nule i jedinice), a operacije koje nad njima obavljaju su operacije Bulove algebre nad ovim skupom vrednosti. Računanje tokom rada aviona koje vodi računa da putnici sigurno slete na odredište, Instagram chata na vašem mobilnom telefonu ili puta na Mesec se obavlja isključivo pomoću operacija Bulove algebre nad skupom od dve vrednosti. Nije teško zaključiti da je poznavanje Bulove algebre i njenih operacija ključno znanje prilikom razumevanja rada uređaja modernih tehnologija.

Ova vežba će vas naučiti osnovama operacija Bulove algebre i projektovanja sistema koji računaju vrednosti Bulovih funkcija, koji su osnovna gradivna komponenta digitalnih sistema. Vežba će vas naučiti i kako da opišete te sisteme koristeći logičke šeme. Tokom projektovanja vašeg prvog digitalnog sistema, koristićete Intel Quartus alat koji omogućava ne samo da opišete digitalni sistem nego i da ga simulirate, tj. proverite njegov rad, i implementirate na određenoj fizičkoj platformi. Znanje koje dobijete u ovoj vežbi je prvi korak ka konačnom cilju ovog predmeta – projektovanju složenih digitalnih sistema za računanje, tj. procesora.

TEORIJSKE OSNOVE

1. Digitalna logička kola

Digitalna logička kola su elektronska kola koja imaju mogućnost računanja vrednosti logičkih funkcija Bulove algebre. Bulova algebra je definisana nad skupom od 2 vrednosti:

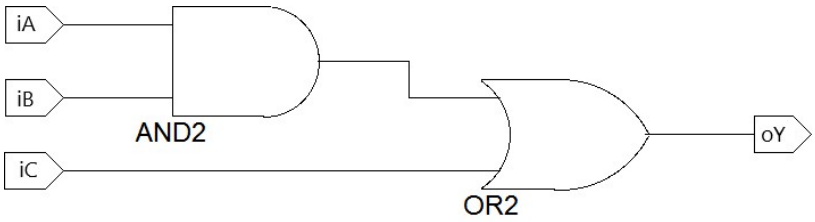
- **Logička nula (0, FALSE)** – predstavljena niskom vrednosti potencijala na žici,
- **Logička jedinica (1, TRUE)** – predstavljena visokom vrednosti potencijala na žici.

Neke od logičkih operacija Bulove algebre su:

- **Negacija (NOT)** – unarna operacija predstavljena znakom komplementa,
- **Konjunkcija (AND)** – binarna operacija predstavljena znakom množenja,
- **Disjunkcija (OR)** – binarna operacija predstavljena znakom sabiranja,
- **Ekskluzivna disjunkcija (XOR)** – binarna operacija predstavljena znakom sabiranja u krugu,
- razne kombinacije ranije navedenih operacija.

Tabela 1.1 na sledećoj strani daje prikaz osnovnih logičkih kola koja se koriste u digitalnoj elektronici, kao i istinitosne tablice koje prikazuju šta će biti izlaz svakog od kola, za svaku kombinaciju ulaza. Svaki od ovih kola se realizuje pomoću elektronskih komponenti – tranzistora. Fizička realizacija logičkih kola je izvan opsega ovog predmeta i neće biti razmatrana. Prilikom projektovanja digitalnih sistema logička kola se koriste kao osnovne komponente, odn. komponente na najnižem nivou hijerarhije.

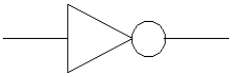
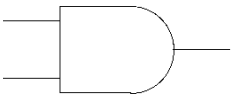
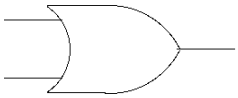
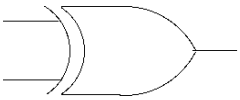
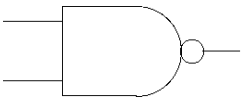
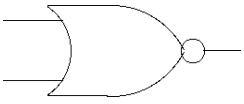
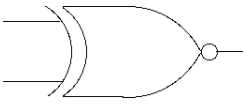
Složene funkcije Bulove algebre se realizuju kombinacijom logičkih kola iz Tabele 1-1. Minimalni skup logičkih kola predstavlja skup pomoću kojeg se može realizovati bilo koja složena funkcija Bulove algebre. Može se dokazati da kolo NAND predstavlja minimalni skup, odn. bilo koja funkcija Bulove algebre može da se realizuje koristeći samo NAND kolo. Slično tako, kolo NOR predstavlja minimalni skup. Zanimljivo je da mnogo korišćenije funkcije u matematičkoj logici - AND, OR i NOT samostalno ne predstavljaju minimalni skup, već minimalni skup sadrži sve tri navedene funkcije. Kao primer realizacije složenih funkcija, posmatrajmo funkciju $Y = AB + C$. Slika 1-1 pokazuje kako se navedena složena funkcija može realizovati pomoću logičkih kola.



Slika 1-1. Primer realizacije složene Bulove funkcije pomoću logičkih kola

Iako izgleda vrlo skromno, sa dve vrednosti i nekoliko operacija, prostor Bulove algebre je matematički izuzetno moćan i dovoljan da se izvrše sva matematička računanja - od jednostavnih, kao što su sabiranje, množenje, do veoma složenih kao što su sistemi odlučivanja u složenim uređajima ili numeričko rešavanje parcijalnih diferencijalnih jednačina. Svi digitalni uređaji su zasnovani na računanju unutar prostora Bulove algebre. Računanje prilikom kontrolisanja leta aviona, rada vašeg mobilnog telefona i kućnog računara, automobila, nečeg složenog kao što je superračunar koji kontroliše let svemirske letelice ili nečeg jednostavnog kao što je budilnik koji vodi računa da ne zakasnite na vežbe iz LPRS1 – sve se to obavlja računanjem u prostoru Bulove algebre i njene dve vrednosti i minimalno jedna operacija su dovoljne za to.

Tabela 1-1. Digitalna logička kola

Naziv	Oznaka	Funkcija	Istinitosna tablica
NOT		$Y = \bar{X}$	X | Y 0 | 1 1 | 0
AND		$Y = X_1 X_2$	X_1 | X_2 | Y 0 | 0 | 0 0 | 1 | 0 1 | 0 | 0 1 | 1 | 1
OR		$Y = X_1 + X_2$	X_1 | X_2 | Y 0 | 0 | 0 0 | 1 | 1 1 | 0 | 1 1 | 1 | 1
XOR		$Y = X_1 \oplus X_2$	X_1 | X_2 | Y 0 | 0 | 0 0 | 1 | 1 1 | 0 | 1 1 | 1 | 0
NAND		$Y = \overline{X_1 X_2}$	X_1 | X_2 | Y 0 | 0 | 1 0 | 1 | 1 1 | 0 | 1 1 | 1 | 0
NOR		$Y = \overline{X_1 + X_2}$	X_1 | X_2 | Y 0 | 0 | 1 0 | 1 | 0 1 | 0 | 0 1 | 1 | 0
XNOR		$Y = \overline{X_1 \oplus X_2}$	X_1 | X_2 | Y 0 | 0 | 1 0 | 1 | 0 1 | 0 | 0 1 | 1 | 1

Digitalna kola zasnovana na logičkim kolima iz Tabele 1-1 se nazivaju **kombinacione mreže**. Svaka kombinaciona mreža zahteva određeno vreme računanja koje mora da protekne da bi se na izlazu pojavio validan rezultat. Ovo vreme postoji zbog parazitne kapacitivnosti u elektronskim komponentima pomoću kojih se realizuju logička kola i konačnog vremena koje mora da protekne da bi signal promenio vrednost od visokog ka niskom potencijalu i obrnuto. Zbog toga se za svako logičko kolo definiše **kašnjenje** logičkog kola, a kašnjenje složenog logičkog kola se računa kao kašnjenje na najdužoj putanji od nekog ulaza do nekog izlaza. Za primer sa Slike 1-1, kašnjenje datog kola jednako je zbiru kašnjenja AND i OR kola, pošto je najduža putanja upravo od ulaza A ili B do izlaza Y, koja prolazi kroz oba kola (da bi OR kolo moglo da računa izlaz, prvo mora AND kolo da izračuna vrednost međurezultata). Kašnjenje kola direktno određuje maksimalnu frekvenciju na kojoj digitalni sistem može da funkcioniše, o čemu ćemo više pričati kasnije.

Iako su dovoljna za sva računanja, logička kola iz Tabele 1-1 nisu dovoljna da bi se realizovao kompletan digitalni sistem. Da bi računanja imala smisla, rezultat tih računanja negde treba da se zapamti kako bi bio ponovno iskoristiv. Za tu svrhu se koriste elektronska kola koja imaju dva stanja čime se logička nula i jedinica mogu „upamtiti“ u tim kolima. Uvodeći mogućnost memorisanja vrednosti, omogućava se da digitalni sistem prolazi kroz stanja, čime se uvodi sekvencijalnost u rad digitalnog sistema. Kola koja uz komponente iz Tabele 1-1 sadrže i elektronska kola za memorisanje nazivaju se **sekvencijalne mreže**. O njima će više biti reči kasnije.

2. Opis sistema sa digitalnim logičkim kolima

Za opis dizajna, sintezu i implementaciju digitalnih sistema koristićemo ***Intel Quartus*** alat, koji omogućava projektovanje digitalnih sistema korišćenjem:

- **logičke šeme digitalnog sistema** (*Schematic*) ili
- **HDL** (*Hardware Description Language*) **opisa digitalnog sistema**

Intel Quartus alat podržava opis digitalnih sistema pomoću dva jezika za opis fizičke arhitekture:

1. **VHDL** (*Very high speed integrated circuit Hardware Description Language*)
2. **VERILOG**

Projektovanje digitalnih sistema pomoću logičke šeme postaje mukotrpan posao kada su u pitanju složeni digitalni sistemi. Tokom rada na ovim vežbama, uskoro ćete primetiti da čak i nešto što se smatra jednostavnim komponentama digitalnih sistema (multiplekser ili sabirač, kao primer) zahteva implementaciju pomoću prilično složenih kombinacija logičkih kola, tj. funkcije koju ove komponente računaju su prilično složene Bulove funkcije.

Kako bi se olakšalo i time ubrzalo projektovanje veoma složenih digitalnih sistema, osmišljeni su jezici za opis digitalnih sistema (**HDL** – Hardware Description Language). Ovi jezici liče na programske jezike, kako bi njihovo usvajanje bilo lakše, no njihova namena je drugačija – oni služe da bi se lakše, brže i kraće opisala arhitektura digitalnog sistema. Opis digitalnog sistema u nekom od HDL jezika je ekvivalentan logičkoj šemi i predstavlja samo drugi način predstave istog digitalnog sistema.

U ovom predmetu koristićemo **VHDL**, jezik za opis fizičke arhitekture. VHDL je HDL jezik namenjen za opis digitalnih sistema visoke integracije, iz čega je dobio i svoj naziv (VHDL – *Very high scale integration circuit Hardware Description Language*). Ostali često korišteni jezici za opis fizičke arhitekture su Verilog, SystemVerilog i SystemC, ali njih nećemo koristiti u ovom predmetu.

ZADACI

3. Moj prvi digitalni sistem (Intel Quartus tutorial)

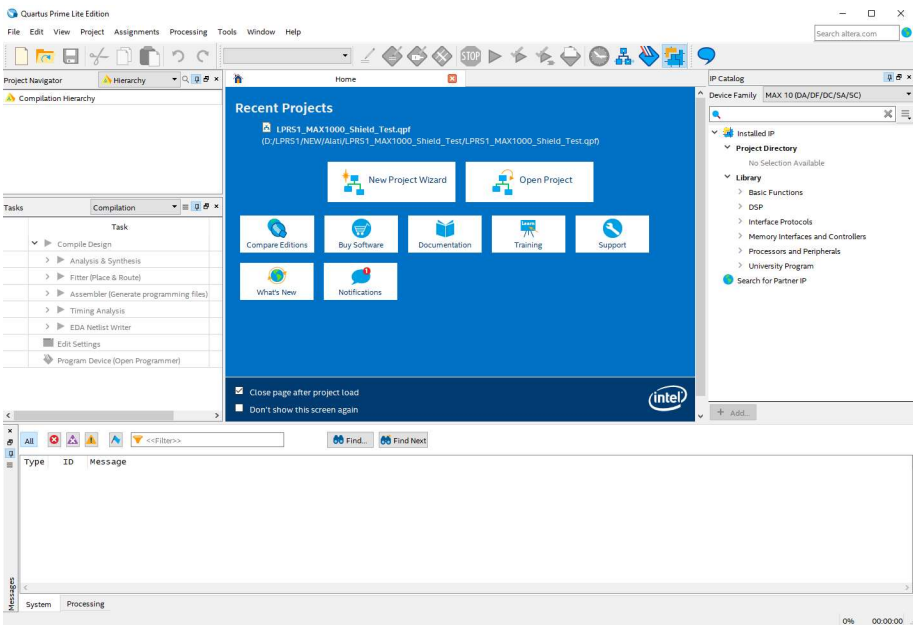
Postupak **implementacije** digitalnog sistema u Intel Quartus alatu se može podeliti u nekoliko koraka:

1. Formiranje projekta
2. Formiranje datoteke u kojoj će biti opis sistema logičkom šemom ili VHDL jezikom,
3. Opis sistema koristeći logičku šemu ili VHDL jezik,
4. Sinteza digitalnog sistema
5. Dodela pinova
6. Implementacija digitalnog sistema na FPGA integrisano kolo

3.1. Formiranje projekta

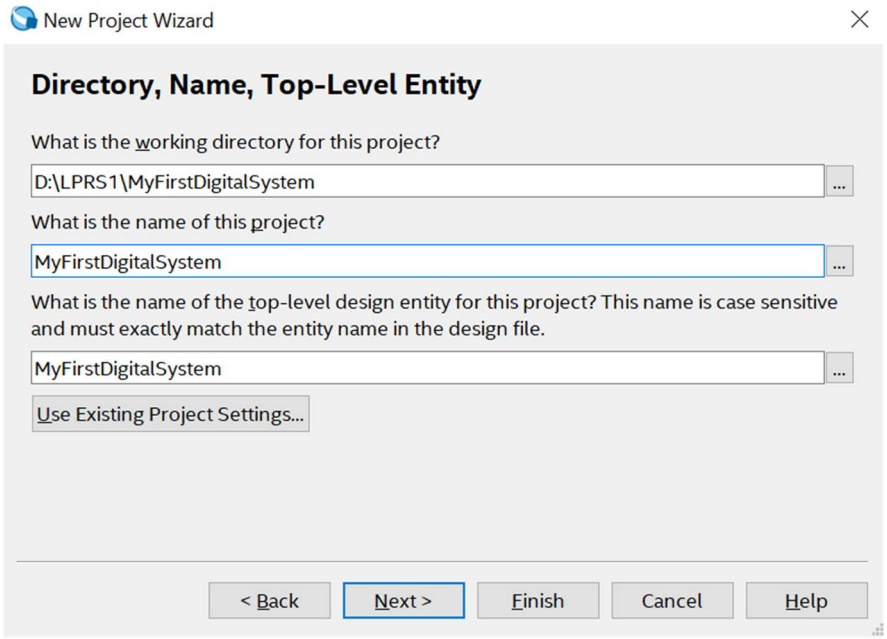
Nakon pokretanja *Intel Quartus* programskog paketa, prikazuje se osnovni prozor (Slika 3-1), koji je podeljen na nekoliko sastavnih prozora:

- Početni prozor (**Project Navigator**) – levi gornji deo ekrana
  Omogućava pregled i izmenu projekta i njegovih elemenata.
- Izlazni prozor (**Messages**) – donji deo ekrana
  Prikazuje izlazne poruke pokrenutih procesa.
- Prozor za zadatke (**Tasks**) – levi srednji deo ekrana
  Omogućava pokretanje programskog prevodioca, kao i pojedinačnih koraka: analiza, sinteza, asembler, vremenska analiza itd.
- Radni prozor (**Workspace**) – centralni i najveći deo ekrana
  U ovom prozoru se pri samom pokretanju Quartus programa mogu pokrenuti neke početne operacije, kao što su formiranje projekta, otvaranje već postojećeg projekta, kao i linkovi ka nekim dodatnim informacijama. Nakon formiranja ili otvaranja projekta, u ovom prozoru se piše i ispravlja HDL kod ili druge tekstualne datoteke, ali se i formiraju logičke šeme.



Slika 3-1. Osnovni prozor Intel Quartus programskog paketa

Na početku projektovanja digitalnog sistema potrebno je napraviti projektnu datoteku koja će pamtiti sve relevantne podatke o projektu (ime projekta, korišćena komponenta, hijerarhijska struktura, datoteke uključene u projekat). Projektne datoteke se prepoznaju po ekstenziji ***.qpf**. Kreiranje projekta se može započeti odabirom komande **File > New Project** ili pritiskom na **New Project Wizard** u radnom prozoru, čime se otvara prozor za definisanje informacija o projektu kao na Slika 3-2.

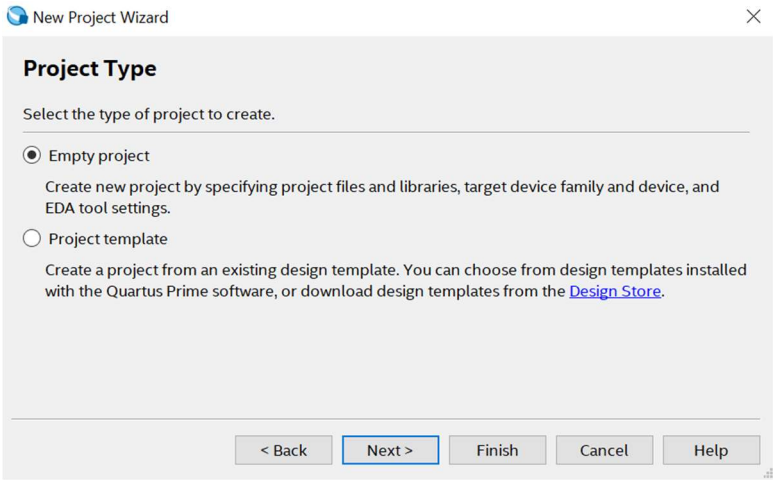


Slika 3-2. Formiranje novog projekta

U prvo polje **Working directory** potrebno je upisati lokaciju projektnog direktorijuma u vidu putanje do odredišnog direktorijuma, dok u drugo polje **Name** treba upisati ime projekta. Treće polje Top-level design entity dobija automatski naziv koje je dato u drugom polju.

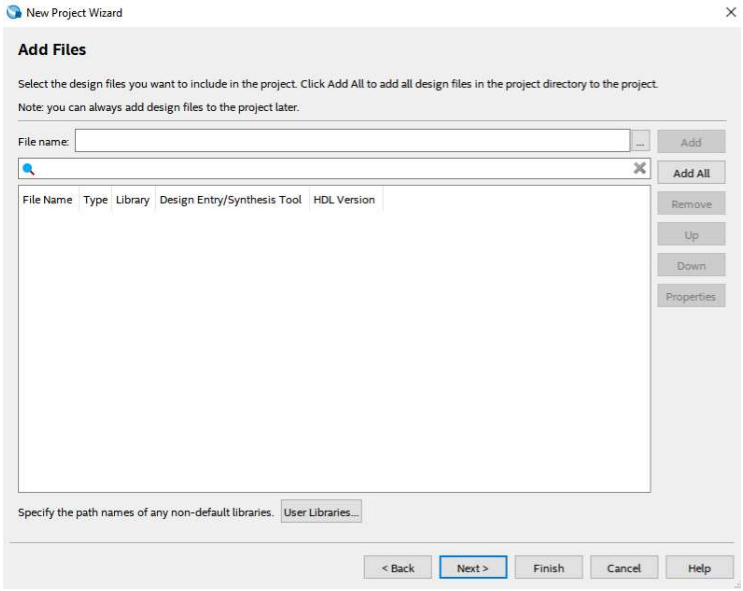
NAPOMENA: Putanja do radnog direktorijuma, kao i nazivi projekta i datoteka **NE SMEJU** imati **RAZMAKE** i **SPECIJALNE KARAKTERE (č, ć, š, đ, ž)** u **NAZIVU**!

Pritiskom na dugme **Next** prelazi se na sledeći prozor **Project Type**, gde je moguće izabrati kreiranje novog projekta (**Empty project**) bez unapred definisanih datoteka ili kreiranje projekta sa unapred definisanim datotekama (**Project template**). Za potrebe ove vežbe izaberite prvu opciju **Empty project** (Slika 3-3).



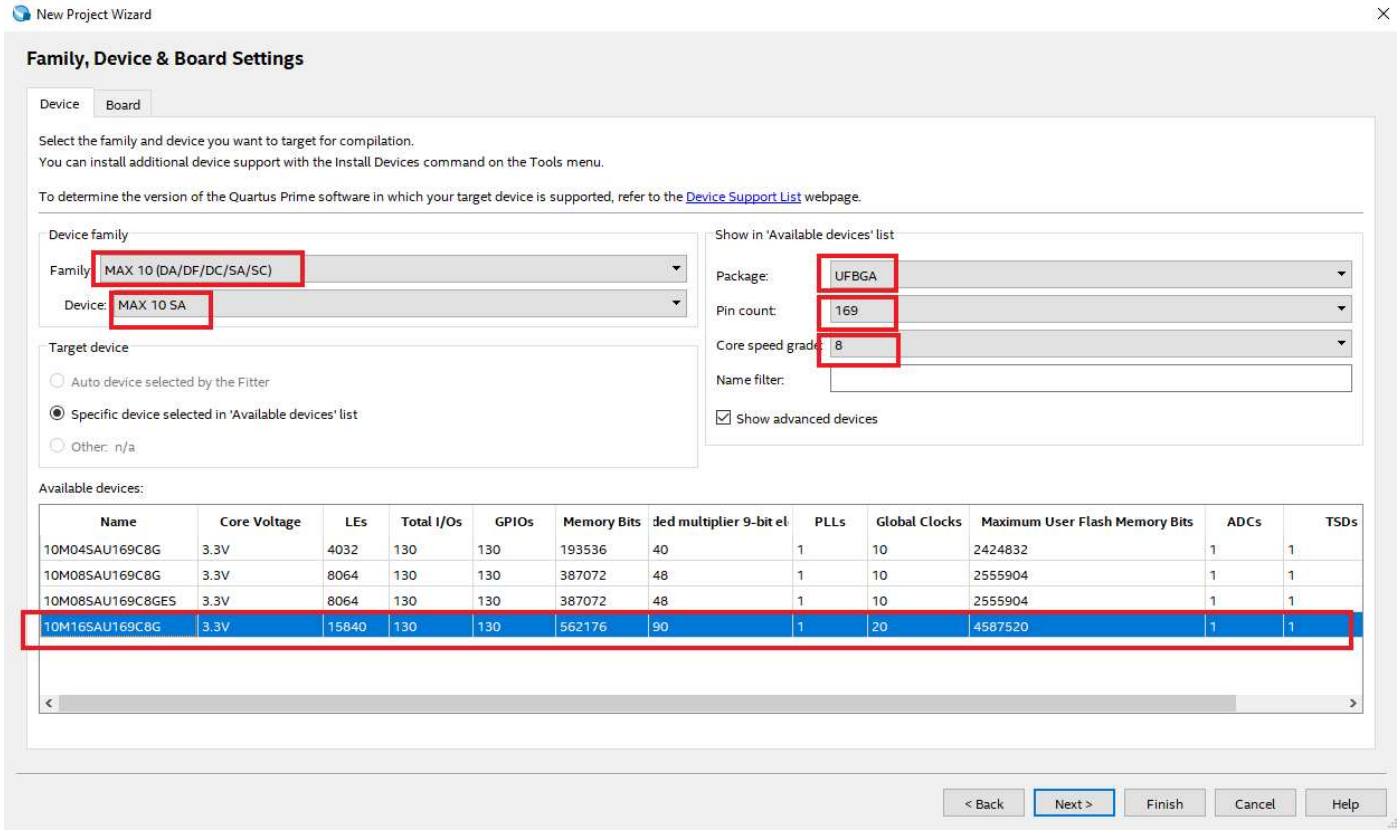
Slika 3-3. Izbor tipa projekta

Sledeći prozor omogućava dodavanje postojećih datoteka što ćemo u ovoj vežbi preskočiti pritiskom na dugme **Next** (Slika 3-4).



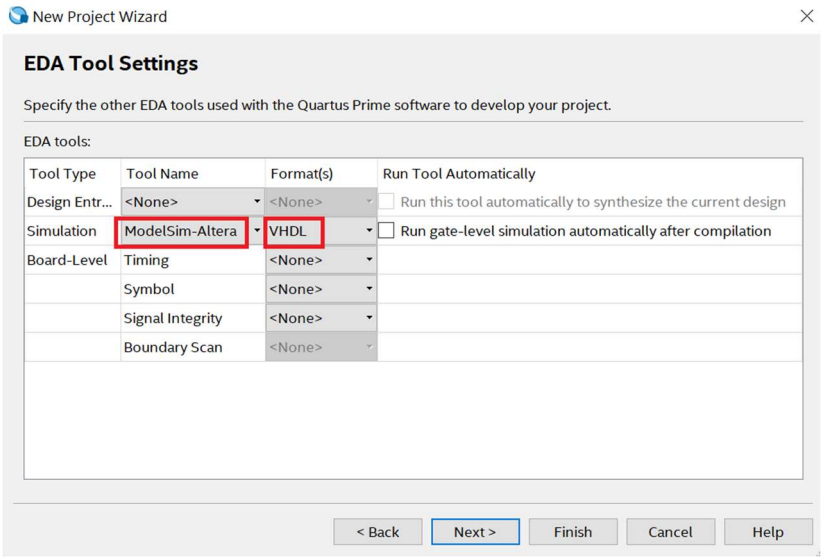
Slika 3-4. Dodavanje postojećih datoteka

U ovom koraku je potrebno podesiti tip uređaja tj. programske platforme za koju se projektuje digitalni sistem (Slika 3-5). U našem slučaju treba odabrati familiju **MAX 10 (DA/DF/DC/SA/SC)** i uređaj **MAX 10 SA**. Zatim u desnom delu prozora treba izabrati paket **UFBGA**, broj pinova **169** i ocenu brzine jezgra **8**. U donjem delu prozora će se pojaviti lista dostupnih uređaja koja obuhvata prethodno zadate parametre i treba izabrati uređaj **10M16SAU169C8G**. Ovaj korak je neophodan ukoliko želimo da naš dizajn/blok šemu spustimo na ploču i probamo je uživo.



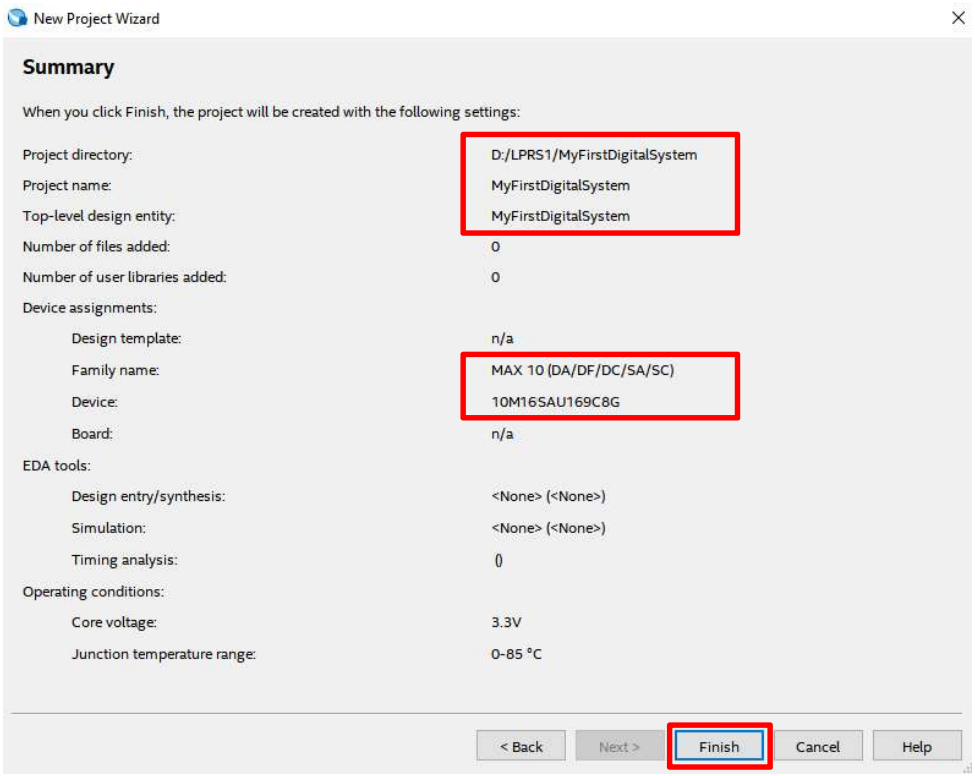
Slika 3-5. Odabir programabilne sekvencijalne mreže

Ukoliko postoji potreba za dodavanjem alata drugih proizvođača (*Third-party tools*), to je moguće uraditi u sledećem koraku podešavanjem EDA alata (*Electronic Design Automation*). Takođe, u ovom koraku je poželjno podesiti polje *Simulation*, ukoliko će se raditi simulacija digitalnog sistema. Na vežbama ćemo koristiti alat **ModelSim-Altera** i format **VHDL** za opis digitalnog sistema i test datoteke, tako da je potrebno izabrati ove dve opcije (Slika 3-6).

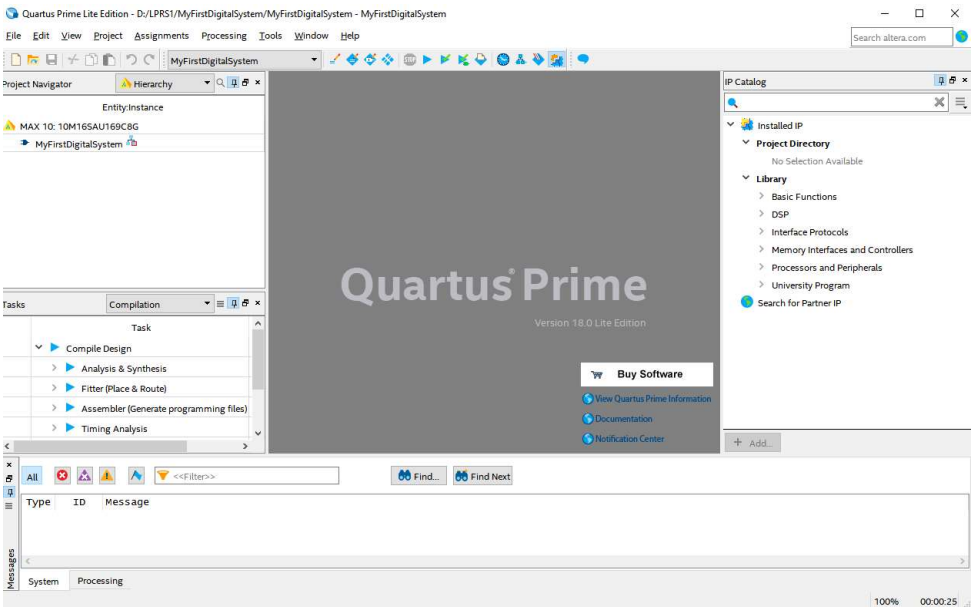


Slika 3-6. Podešavanje EDA alata

Poslednji prozor za formiranje projekta prikazuje sažetak prethodno odabranih postavki (Slika 3-7). Ukoliko je potrebno promeniti neki parametar, klikom na dugme **Back** se može vratiti unatrag kroz opisane prozore i promeniti željena informacija. Klikom na dugme **Finish** se završava formiranje projekta i dobija se osnovni *Intel Quartus* prozor kao na Slici 3-7A.



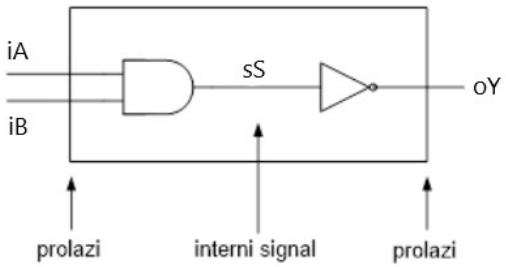
Slika 3-7. Sažetak odabranih postavki za novi projekat



Slika 3-7A. Početni prozor novog projekta u Intel Quartus programskom paketu

3.2. Opis digitalnog sistema logičkom šemom

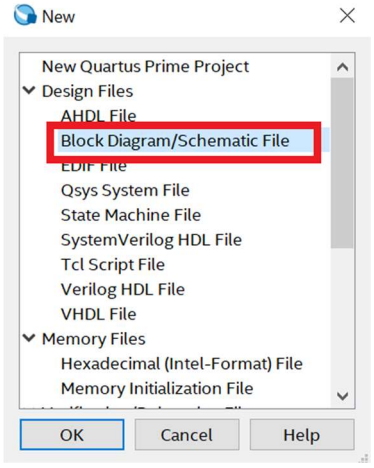
Vreme je da opišete vaš prvi digitalni sistem! Digitalni sistem sa Slike 3-8 ima **2 ulaza** (*iA* i *iB*) i **1 izlaz** *oY* i oni čine **entitet** digitalnog sistema. **Arhitekturu** digitalnog sistema čini jedno **AND** i jedno **NOT** logičko kolo.



Slika 3-8. Naš prvi digitalni sistem

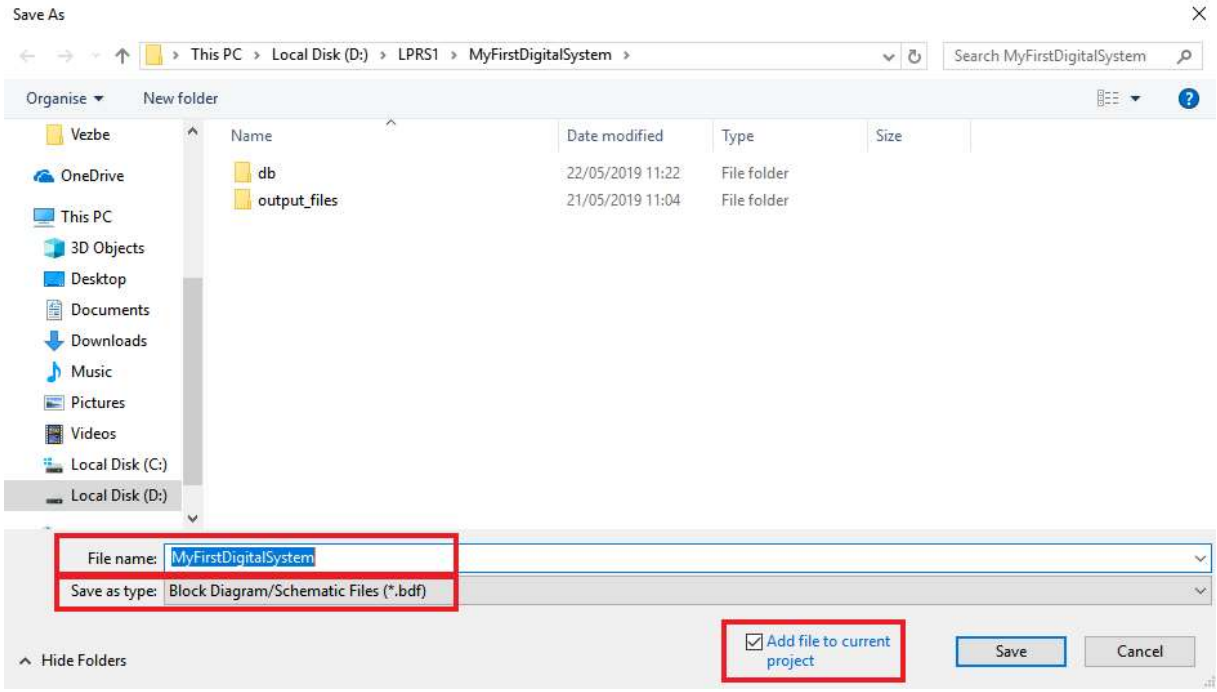
3.3. Formiranje logičke šeme digitalnog sistema

Nova datoteka u koju će biti smeštena logička šema digitalnog sistema se formira izborom **File->New** iz glavnog menija, nakon čega dobijamo prozor kao na Slici 3-9. Potrebno je izabrati **Block Diagram/Schematic File** i pritisnuti **OK**.



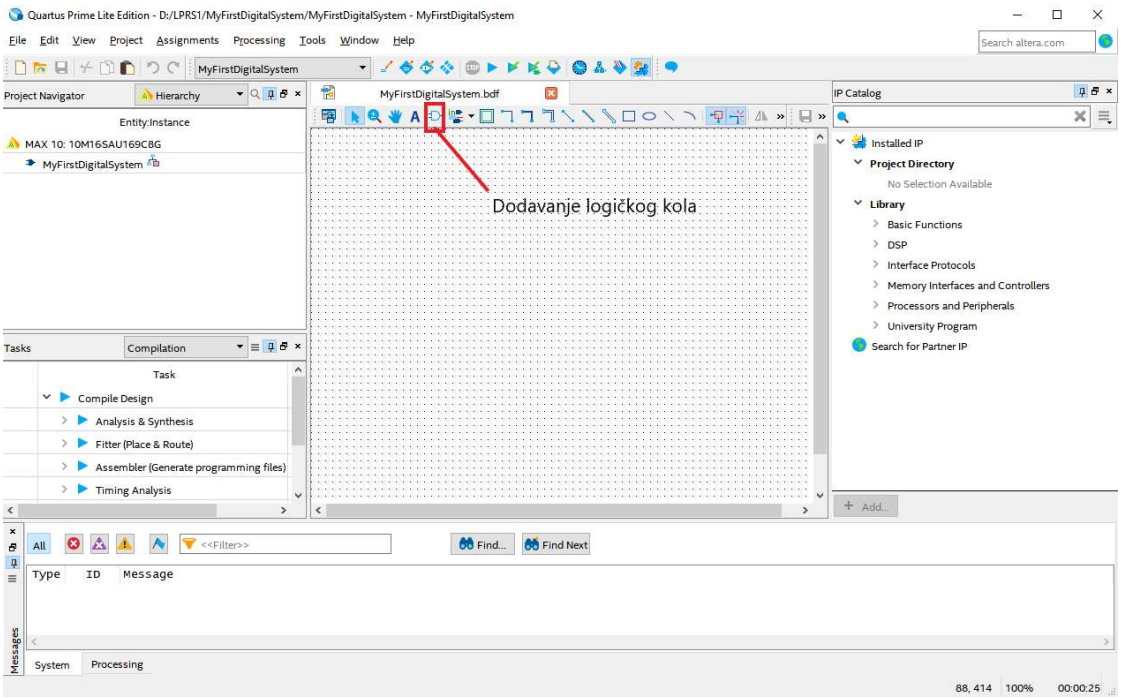
Slika 3-9. Formiranje datoteke sa opisom digitalnog sistema

Nakon ovog koraka, potrebno je dati ime novoj datoteci u koju će se smestiti logička šema izborom **Save As** iz **File** menija. U polje *File name* potrebno je uneti **ime datoteke**, dok u polju *Save as type* treba izabrati **Block Diagram/Schematic File (*.bdf)**. Takođe je potrebno označiti polje **Add file to current project**. Zatim pritisnite dugme **Save**, kako bi sačuvali novoformiranu datoteku u projektni direktorijum (Slika 3-10).

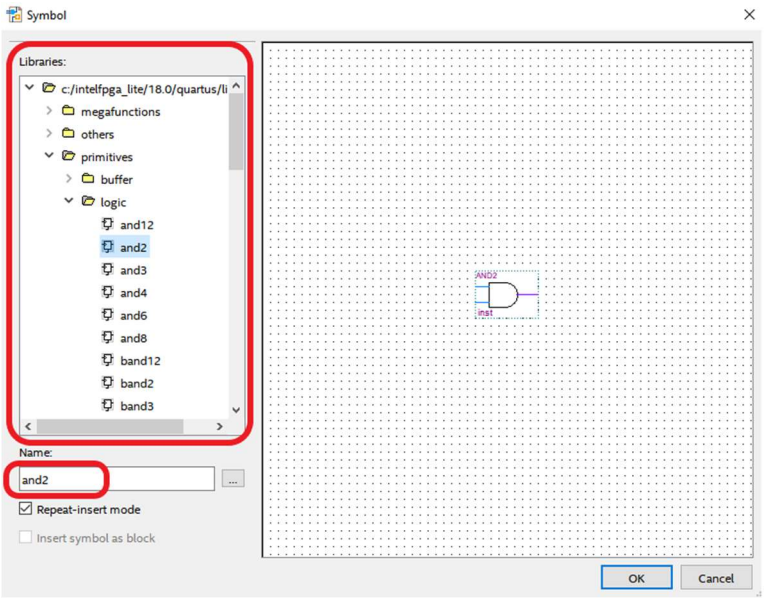


Slika 3-10. Dodavanje nove datoteke u Quartus projekat

Konačni izgled prozora sa grafičkim editorom za pravljenje logičke šeme prikazan je na Slici 3-11. Logička šema se formira dodavanjem elemenata logičkih kola pritiskom na dugme koje izgleda kao logičko AND kolo , koje je obeleženo na Slici 3-11 ili duplim pritiskom na grafički prozor. Tada se otvara novi prozor prikazan na Slici 3-12, koji omogućava veliki izbor logičkih kola, ulaza, izlaza itd.



Slika 3-11. Grafički prozor za pravljenje logičke šeme



Slika 3-12. Dodavanje elemenata logičke šeme i izgled konačne šeme

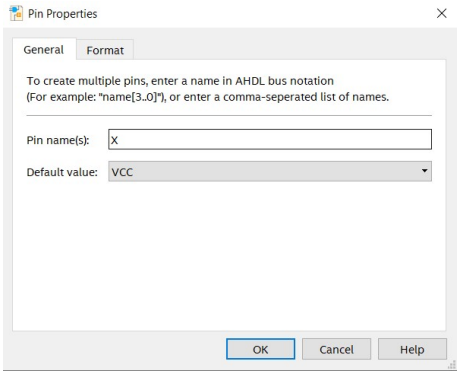
Simboli koji odgovaraju logičkim kolima iz Tabele 1-1 su:

- **not** – invertor (logičko kolo NOT),
- **and2** – dvoulazno AND kolo,
- **or2** – dvoulazno OR kolo,
- **xor2** – dvoulazno XOR kolo,
- **nand2** – dvoulazno NAND kolo,
- **nor2** – dvoulazno NOR kolo,
- **xnor2** – dvoulazno XNOR kolo.

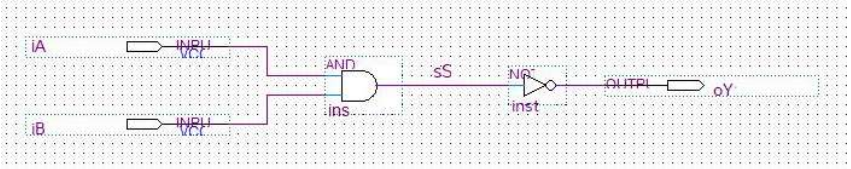
Veze između kola se definišu spajanjem njihovih ulaza/izlaza pomoću žica: *Orthogonal Node Tool.*

Ulazi i izlazi sistema se definišu pomoću ulazno/izlaznih markera: *Pin Tool (Input, Output, Bidir)*.

Nakon što je završeno pravljenje logičke šeme, potrebno je dati imena ulazima i izlazima. Duplim klikom na ulazni signal, otvara se prozor kao na Slici 3-13 i potrebno je dodeliti imena ulazima i izlazima i podesiti podrazumevanu vrednost na VCC ili GND. Izgled konačne šeme je prikazan na Slici 3-14.



Slika 3-13. Dodela imena ulazima i izlazima



Slika 3-14. Završena Block Diagram šema

3.4. Sinteza digitalnog sistema

Napravljena datoteka sa logičkom šemom ***MyFirstDigitalSystem.bdf*** se obrađuje pomoću Quartus Prime alata za analizu i sintezu i generiše implementaciju za ciljnu platformu.

Prevođenje se pokreće izborom **Processing->Start Compilation** iz glavnog menija ili pritiskom na dugme u gornjem delu ekrana koje izgleda kao plavi trougao . U toku prevođenja prikazuju se poruke u donjem prozoru, dok se nakon završetka prikazuje prozor sa statusom, greškama i upozorenjima.

Ukoliko je digitalni sistem ispravan, jedna od poslednjih poruka će biti da je prevođenje uspešno završeno i da nema grešaka. Ukoliko ipak greške postoje, duplim klikom na poruku sa greškom, Intel Quartus će prikazati deo na šemi koji ima problem. Ispravite grešku i ponovo pokrenite prevođenje.

3.5. Dodela pinova

Ukoliko nema grešaka prilikom prevođenja, sledeći korak je dodela pinova FPGA ulazima i izlazima.

Zbog čega ovo radimo? FPGA integrisano kolo je povezano sa ostatkom MAX 10 platforme preko svojih nožica, odn. **pinova**, koje možemo posmatrati kao ulaze i izlaze samog FPGA integrisanog kola. Svaki pin je povezan na neku od komponenti MAX 10 platforme, npr. na prekidače, tastere, LED-ove, LCD, itd. Kada želimo da implementiramo sistem za FPGA, mi moramo alatu naglasiti na koji pin na poveže koji ulaz i izlaz našeg sistema. Pinove biramo tako da naš sistem bude proverljiv na platformi, odn. ulaze ćemo povezati na neki od prekidača ili tastera, a izlaze najčešće na neke od dioda. Ukoliko naš sistem radi sa LCD ekranom ili nekim drugom složenijom periferijom koja postoji na platformi (npr. sa HDMI izlazom ili audio sistemom), onda ulaze i izlaze sistema treba povezati na odgovarajuće pinove koji su na platformi fizički povezani sa odgovarajućom periferijom.

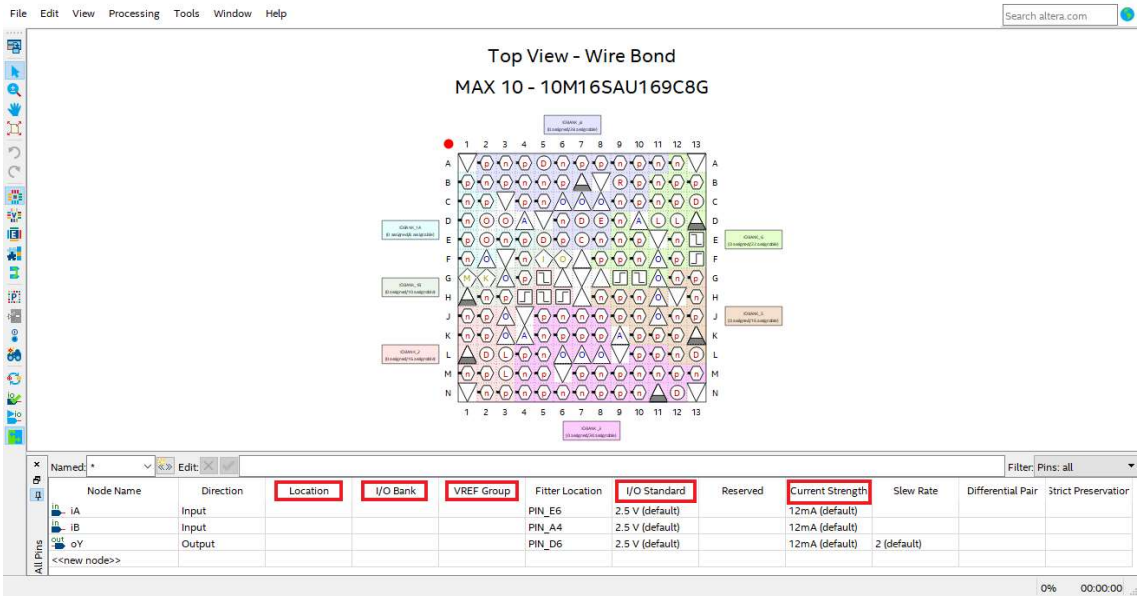
Za primer našeg prvog digitalnog sistema, ulaze ćemo povezati na levi i desni taster, a izlaz na LED diodu, prema sledećoj Tabeli 3-1:

Tabela 3-1. Dodela pinova za ulaze/izlaze našeg prvog digitalnog sistema

Port	Smer	Komponenta	FPGA pin
iA	input	PB_LEFT	PIN_D1
iB	input	PB_RIGHT	PIN_E3
oY	output	LED 0	PIN_A8

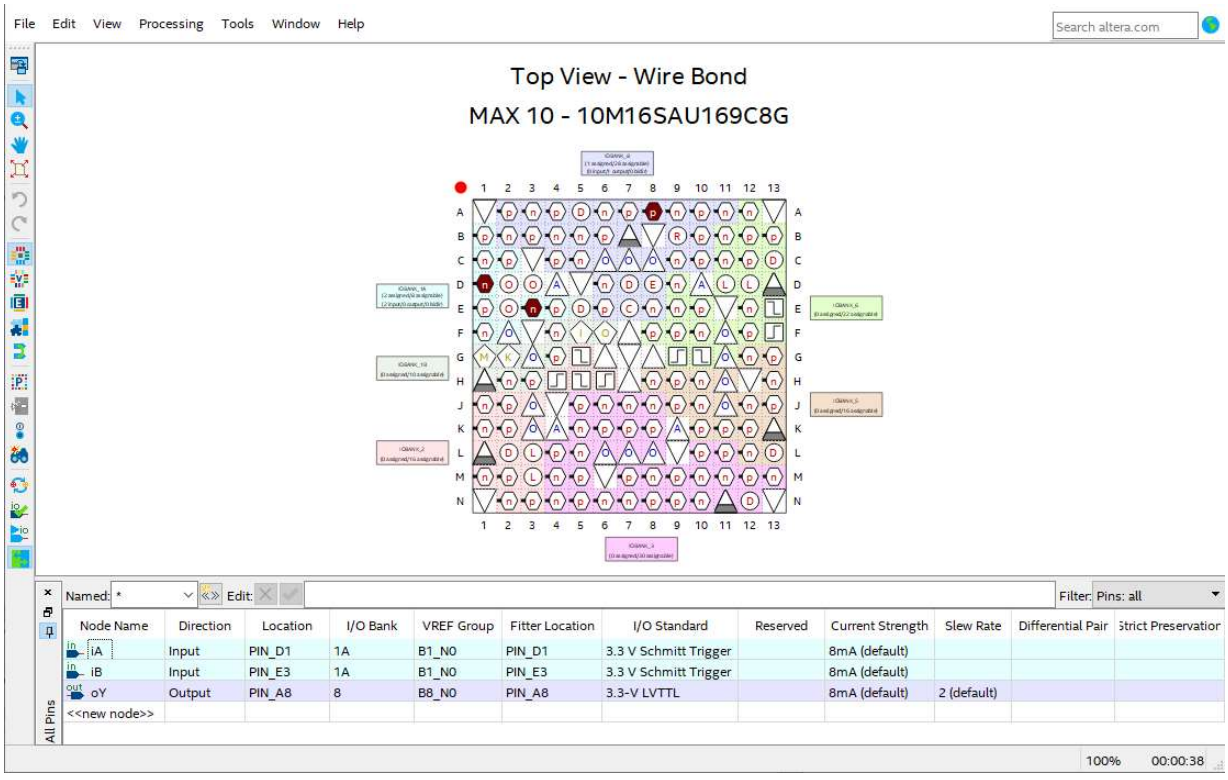
Spisak svih pinova i podešavanja se nalazi u dokumentu ***LPRS1_FPGA_pins.pdf***.

Dodela pinova ulazima i izlazima sistema se obavlja u alatu Intel Quartus korišćenjem ***Pin Planner***-*a*, koji omogućava grafički prikaz ulazno-izlaznih resursa na ciljnoj platformi. Izaberite opciju menija **Assignment->Pin Planner** ili pritisnite ikonicu iznad radnog prozora i otvoriće se prozor kao na Slici 3-15. Podrazumeva se da je prevođenje sistema urađeno pre ovog koraka, kako bi bili prikazani svi postojeći ulazi i izlazi u prozoru za dodelu pinova. Intel Quartus će automatski dodeliti proizvoljne pinove u **Fitter Location** polju, a potrebno je podesiti sledeća polja za svaki ulaz i izlaz: **Location**, **I/O Bank**, **VREF Group**, **I/O Standard** i **Current Strength**.



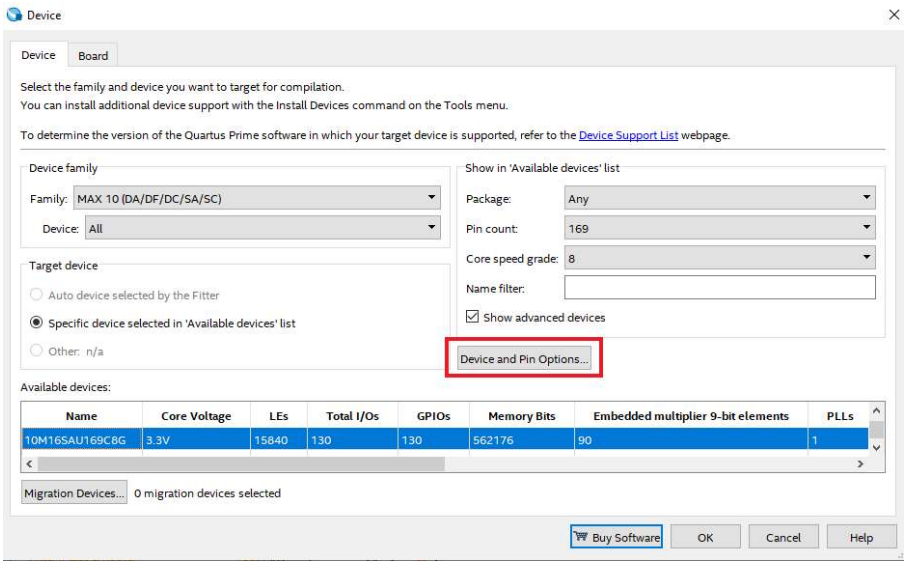
Slika 3-15. Dodela pinova u Intel Quartus-u

Unesite podešavanja kao na Slici 3-16, a zatim ponovo **prevedite ceo projekat**. Tek nakon ponovnog prevođenja sistema, polje **Fitter Location** će promeniti proizvoljno dodeljene u zadate pinove.

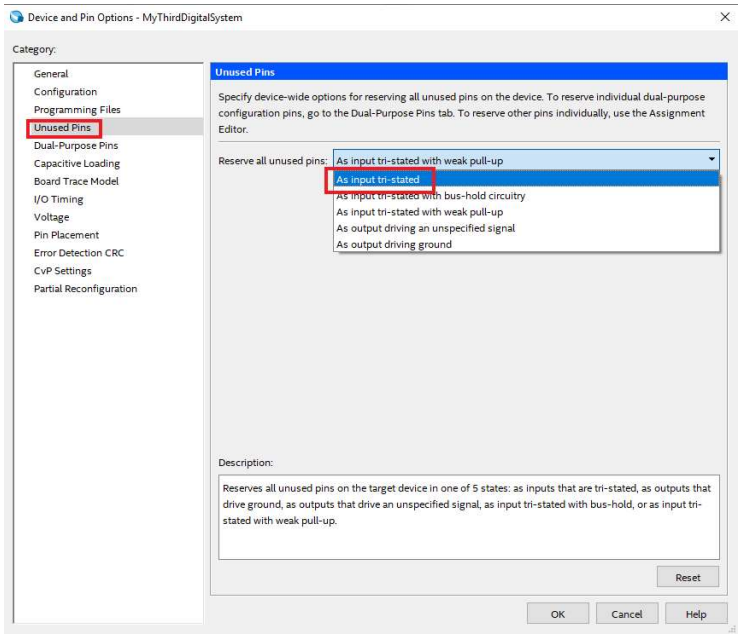


Slika 3-16. Izgled Pin planer-a nakon dodele pinova

Kako bi isključili nepotrebne PIN-ove, potrebno je u glavnom meniju **Assignments/Device** pritisnuti taster **Device and Pin Options...** kao na Slici 3-17. U novom prozoru sa leve strane izaberite podešavanje za nekorišćene pinove - **Unused Pins**, a zatim u opcijama **Reserve all unused pins** podesite **As input tri-stated**, kao na Slici 3-18.



Slika 3-17. Dodatno podešavanje pinova

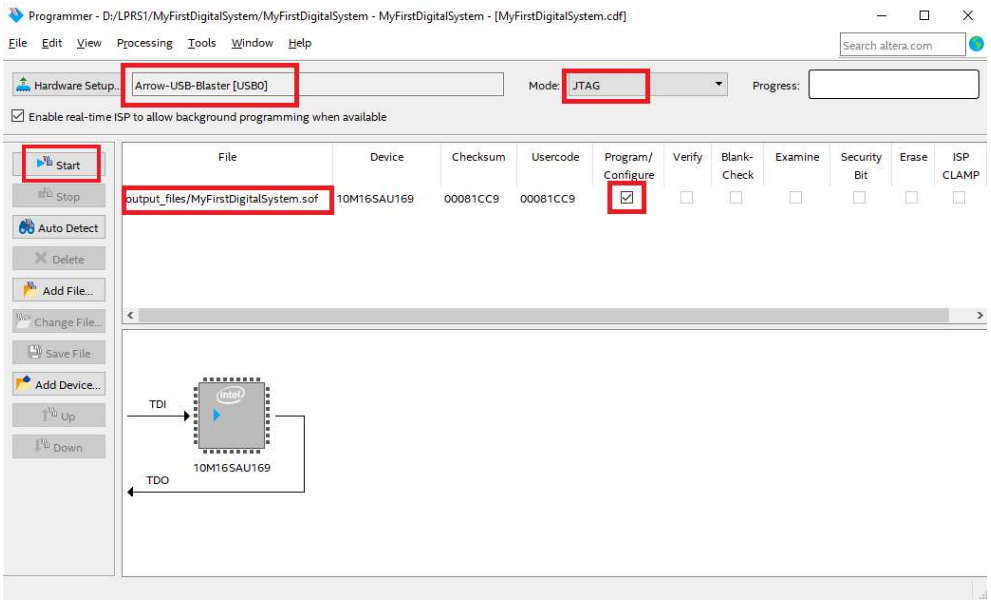


Slika 3-18. Podešavanje nekorišćenih pinova

3.6. Implementacija digitalnog sistema

MAX10 platforma, koju ćemo koristiti na ovim vežbama, sadrži FPGA integrisano kolo koje se može konfigurisati da se ponaša kao proizvoljan digitalni sistem. Intel Quartus alat omogućava da se digitalni sistem opisan VHDL jezikom ili logičkom šemom, implementira za određeno FPGA integrisano kolo. To znači da se generiše konfiguraciona datoteka sa kojom može da se izvrši konfiguracija FPGA integrisanog kola kako bi se on ponašao kao željeni projektovani digitalni sistem.

Prvi korak za implementaciju digitalnog sistema je povezivanje MAX 10 platforme preko USB Blaster-a i USB porta na računaru. Intel Quartus alat prilikom prevođenja generiše **.sof** datoteku, pomoću koje se obavlja JTAG configuracija za MAX 10 platformu. Nakon prevođenja, potrebno je uneti odgovarajuća podešavanja za programiranje platforme. Izborom opcije glavnog menija **Tools->Programmer** ili pritiskom na ikonicu , otvara se prozor kao na Slici 3-19.



Slika 3-19. Implementacija digitalnog sistema

U Hardware Setup prozoru izaberite **Arrow-USB-Blaster[USB0]**, Mode: **JTAG** i u donjem prozoru putanju do **MyFirstDigitalSystem.sof** datoteke. Obeležite opciju **Program/Configure** i nakon toga pritisnite dugme **Start** u levom delu prozora. Pokrenuće se implementacija digitalnog sistema i probajte da testirate napravljeni sistem.

3.7. Provera ispravnosti sistema – testiranje

Nakon uspešne sinteze sistema, sledeći korak je provera ispravnosti. To je moguće uraditi na dva načina:

- Testiranje na razvojnoj platformi
- Testiranje uz pomoć simulatora.

Generalno je praksa da se u ranim fazama razvoja oslanjamo prvenstveno na simulator, a kasnije na razvojnu platformu. Za potrebe simulacije je potrebno kreirati posebnu VHDL datoteku koja se naziva **Test Bench** u kojoj se zadaju **SVI ulazi** sistema koji želimo da testiramo, dok se izlazi sistema posmatraju u simulaciji preko talasića.

Testiranje na razvojnoj platformi podrazumeva da smo pre provere ispravnosti dodelili pinove i implementirali sistem u Intel Quartus alatu, nakon čega možemo fizički proveriti ponašanje sistema za zadate ulaze (npr. pritiskom određenih tastera ili prekidača, očekujemo da će tačno određena dioda biti uključena).

U toku testiranja treba voditi računa da se pokriju svi ili što više testnih slučajeva (zadavanjem svih kombinacija ulaza), a zatim treba proveriti ispravnost izlaza sistema. U nastavku sledi istinitosna tablica sistema koji smo dizajnirali i na osnovu nje treba proveriti ispravnost sistema. Istinitosna tablica pokazuje da na izlazu treba da bude uključena dioda (Y = 1) za sve slučajeve, osim kada su oba tastera pritisnuta tj. kada su A = 1 i B = 1.

iA	iB	sS	oY
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

4. Moj (već) drugi digitalni sistem

Čestitamo na vašem prvom uspešno projektovanom, opisanom i proverenom digitalnom sistemu! Sada je vreme da sami implementirate nešto složeniji sistem (kombinacionu mrežu) sa 3 izlaza i 3 ulaza. Pretpostavimo da je naš novi digitalni sistem opisan sledećom istinitisnom tablicom, tj. sledećim trima Bulovim funkcijama od 3 promenljive:

iX2	iX1	iX0	oY2	oY1	oY0
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	0

Sledeće jednačine predstavljaju Bulove funkcije koje sistem računa:

$$oY2 = iX2 * \overline{iX0} + iX2 * \overline{iX1} + \overline{iX2} * iX1 * iX0$$

$$oY1 = iX1 * \overline{iX0} + \overline{iX1} * iX0$$

$$oY0 = \overline{iX0}$$

Vaš zadatak je da prođete kroz sve korake projektovanja sistema i opišete i implementirate ovaj sistem koristeći Intel Quartus alat. Napravite novi projekat i nazovite ga **MySecondDigitalSystem**. Ovaj naziv će biti zgodan jer ćete moći da iskoristite dobar deo test bench-a koji smo koristili za zadatak 1 i samo promenite imena gde je potrebno.

I za kraj jedno

> **Pitanje:** Ukoliko ulaz i izlaz sistema posmatramo kao 3-bitne brojeve (gde je indeks bita jednak stepenu mesne vrednosti broja), koju matematičku funkciju ovaj sistem računa?

5. Zaključak

Ova vežba vas je upoznala sa pojmom digitalnih logičkih kola i kombinacionih mreža kao fizičkim realizacijama Bulovih funkcija. Videli ste kako se digitalni sistem projektuje, od momenta ideje na papiru, preko opisa sistema do simulacije. Znanje koje ste dobili u ovoj vežbi je dovoljno da ste u stanju samostalno da projektujete bilo koju kombinacionu mrežu zadatu Bulovim funkcijama. U narednoj vežbi ćemo raditi nešto složenije digitalne sisteme uz primenu Karnoovih mapa i VHDL jezika za opis fizičke arhitekture.