

Sveučilište u Zagrebu
Fakultet elektrotehnike i računarstva

Upute za rad s FPGA razvojnom pločicom ULX2S

Marko Zec

21. listopada 2011.

Sadržaj

1 Uvod.....	2
1.1 Programirljiva polja logičkih blokova (FPGA).....	3
1.2 Postupci opisa, sinteze i programiranja konfiguracije FPGA sklopa.....	5
2 Komponente razvojne pločice.....	7
2.1 Napajanje.....	7
2.2 FPGA sklop.....	8
2.3 JTAG.....	8
2.4 Tipke.....	8
2.5 Prekidači.....	8
2.6 LED indikatori.....	9
2.7 Višenamjenska stereo priključnica.....	9
2.8 RS-232 sučelje.....	9
2.9 Generator takta (oscilator).....	9
2.10 SRAM.....	9
2.11 Flash.....	10
2.12 MicroSD.....	10
2.13 DIL priključnice za proširenje.....	10
3 Instaliranje programske podrške.....	11
3.1 Instaliranje programskog paketa Lattice Diamond.....	11
3.2 Instaliranje programskog alata za programiranje pločice.....	15
3.3 Instaliranje USB pogonskih programa (drivera) za razvojnu pločicu.....	15
3.4 Instaliranje programskih alata na operacijski sustav Linux.....	17
3.5 Instaliranje programskih alata na operacijski sustav FreeBSD 8.2.....	17
4 Ispitivanje rada pločice.....	18
4.1 Ispitivanje funkcionalnosti USB RS-232 sučelja.....	18
4.2 Ispitivanje funkcionalnosti višenamjenske stereo priključnice.....	20
5 Rad s razvojnim alatom Lattice Diamond.....	21
5.1 Kreiranje novog projekta.....	21
5.2 Shematski opis digitalnog sklopa.....	24
5.3 Sinteza konfiguracije FPGA sklopa.....	29
5.4 Programiranje FPGA sklopa.....	30
6 Definicije ulazno / izlaznih signala.....	32

1 Uvod

Pločica ULX2S (USB, Lattice XP2, SRAM) namijenjena je razvoju i ispitivanju rada digitalnih sklopova i sustava koji se mogu sintetizirati na integriranim programirljivim poljima logičkih blokova (eng. *Field Programmable Gate Array – FPGA*). Prvenstveno je zamišljena kao nastavno pomagalo u svladavanju temeljnih načela rada, analize i projektiranja digitalnih sklopova, ali zbog svojih malih dimenzija i mogućnosti povezivanja s vanjskim sklopovima može poslužiti i kao modul širokog spektra primjene za ugradnju u kompleksnije uređaje.

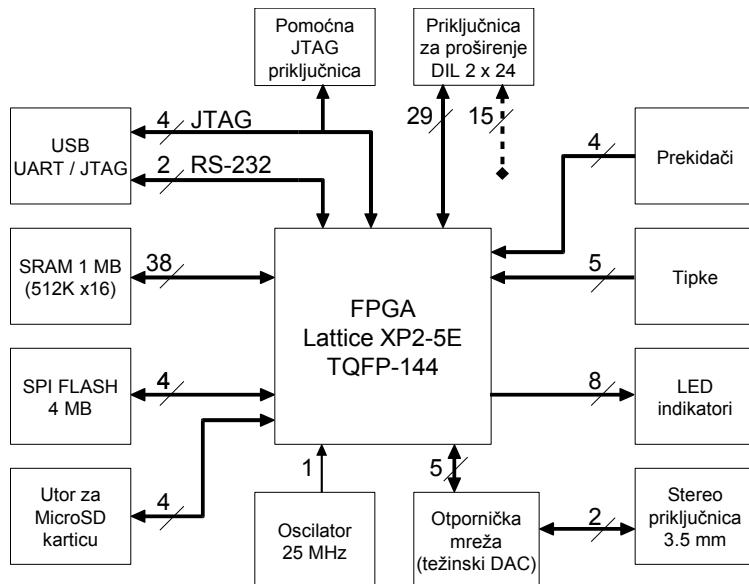
Glavna komponenta na pločici je FPGA sklop XP2-5E proizvođača Lattice Semiconductor, s oko 5000 programirljivih logičkih elemenata temeljenih na preglednim tablicama s 4 ulaza (eng. *4-input lookup table – LUT*). Pri uspostavi napajanja automatski se aktivira konfiguracija FPGA sklopa pohranjenu u integriranu Flash memoriju. FPGA sklop se može neograničeni broj puta rekonfigurirati putem USB sučelja, koje ujedno služi i kao glavni izvor napajanja, te kroz koje se može ostvariti i asinkrona serijska komunikacija s računalom (RS-232). Kako osim USB kabela ne zahtijeva nikakve dodatne komponente za napajanje ili povezivanje s računalom, s pločicom se može raditi kako u laboratoriju tako i kod kuće. Robusni mehanički dizajn i male dimenzije čine pločicu podesnom za učestalo prenošenje.

Ugrađeni FPGA sklop raspolaže s ukupno 100 programirljivih vanjskih ulazno-izlaznih priključaka putem kojih je ostvareno povezivanje sa sljedećim komponentama i priključnicama na razvojnoj pločici:

- LED indikatori (8)
- Tipke (5)
- Prekidači u *dual in-line (DIL)* kućištu (4)
- Generator takta (oscilator) frekvencije 25 Mhz i stabilnosti +/- 25 ppm
- USB - RS-232 asinkrono serijsko sučelje
- Višenamjenska stereo priključnica s 4-bitnom težinskom otporničkom mrežom za D/A pretvorbu
- Utor za memoriske kartice MicroSD
- Flash memorija sa serijskim sučeljem (SPI) kapaciteta 4 MByte
- SRAM memorija kapaciteta 1 MByte (512k * 16 bit)
- Priključnice za proširenje (ukupno 2 * 24 priključka od kojih su 44 signala povezana na FPGA sklop)

Na razvojnoj pločici predviđeno je i mjesto za ugradnju pomoćnog priključka za programiranje (JTAG), koji može poslužiti za povezivanje vanjskog programatora u slučaju potrebe za funkcijama koje trenutno nisu podržane na integriranom USB-JTAG sučelju (npr. *in-system debugging*).

Način povezivanja FPGA sklopa s komponentama i priključnicama na razvojnoj pločici prikazan je blok shemom na slici 1.

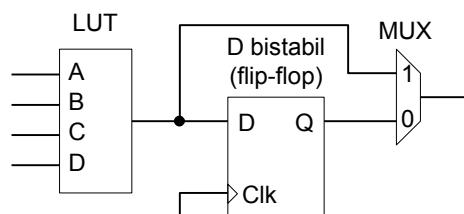


Slika 1: komponente razvojne pločice

1.1 Programirljiva polja logičkih blokova (FPGA)

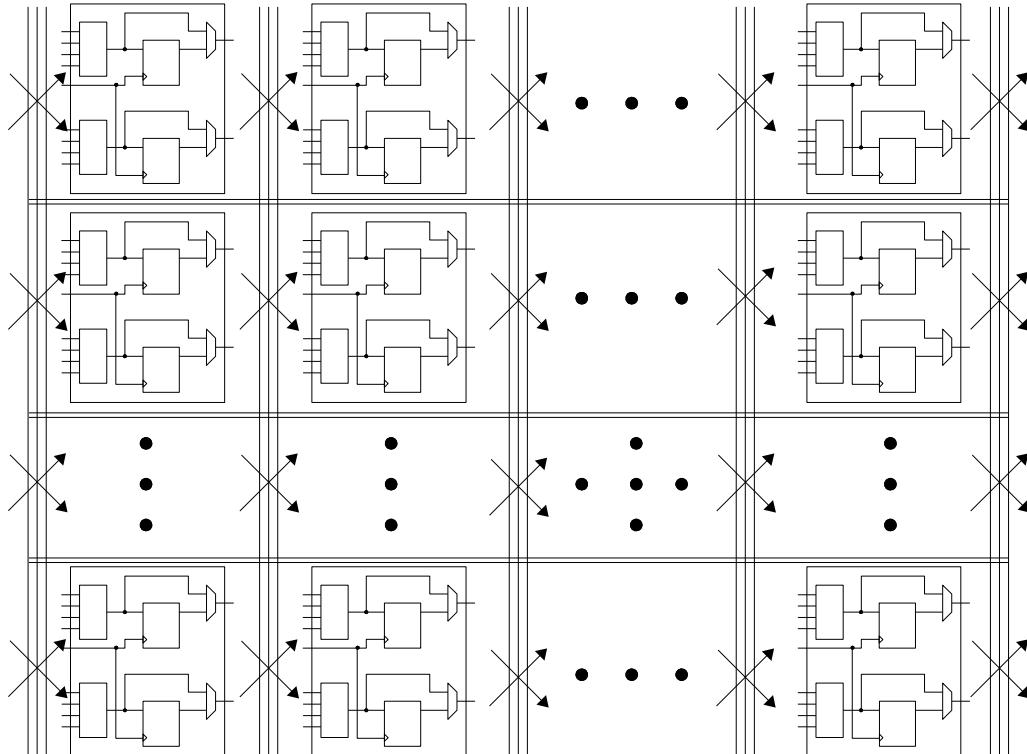
Programirljiva polja logičkih blokova (eng. *Field programmable gate array – FPGA*) su digitalni sklopovi vrlo visokog stupnja integracije u kojima se veliki broj relativno jednostavnih i uniformnih programirljivih logičkih elemenata može povezati u proizvoljni kompleksni sklop, te takav sklop povezati s vanjskim svijetom putem programirljivih ulazno-izlaznih priključaka. FPGA sklopovi danas imaju široku primjenu u industriji (automatsko upravljanje), medicini (ultrazvuk, CT, MR), prometu (sonar, radar), radiokomunikacijama i telekomunikacijama (odašiljači, mrežni prospojnici, sklopovski vatrozidovi), potrošačkoj elektronici (npr. HD video preklopnici), te općenito pri razvoju digitalnih sustava kao pomagala za ispitivanje prototipnih implementacija.

Temeljne komponente programirljivih blokova su **pregledna tablica** (eng. *lookup table – LUT*) i sinkroni bridom okidani **D bistabil** (eng. *flip-flop*). Pregledna tablica omogućuje izvedbu proizvoljne logičke funkcije od N varijabli, gdje je N broj ulaza, koji u današnje vrijeme zavisno od proizvođača i kategorije FPGA sklopa može biti 4, 5 ili 6. Izlaz iz jedne pregledne tablice može se direktno dovesti na ulaz memorijskog elementa (bistabil) ili povezati s ulazima drugih preglednih tablica te takvim kaskadiranjem ostvariti kompleksnije logičke funkcije. Pojednostavljeni model temeljnog programirljivog **logičkog elementa** (eng. *logic element*) modernog FPGA sklopa koji se sastoji od pregledne tablice i D bistabila prikazan je na slici 2.



Slika 2: struktura programirljivog logičkog elementa

Više logičkih elemenata (obično 2 ili 4) grupirani su u tzv. **logičke blokove**, za koje svaki proizvođač ima vlastitu (ponekad nekonzistentnu) terminologiju, npr. *configurable logic block*, *logic slice*, *logic cell* i slično. Za međusobno povezivanje logičkih blokova FPGA sklopovi raspolažu s programirljivom mrežom prospojnih vodova koja omogućuje povezivanje bilo kojeg izlaza bilo kojeg logičkog bloka s bilo kojim ulazom nekog drugog ili istog logičkog bloka odnosno elementa. Posebni dio mreže služi za sinkrono dovođenje signala takta do svakog logičkog bloka. Pojednostavljena struktura tipičnog FPGA sklopa koja se sastoji od polja programirljivih logičkih blokova i prospojne mreže prikazana je na slici 3.



Slika 3: struktura programirljivog polja logičkih blokova

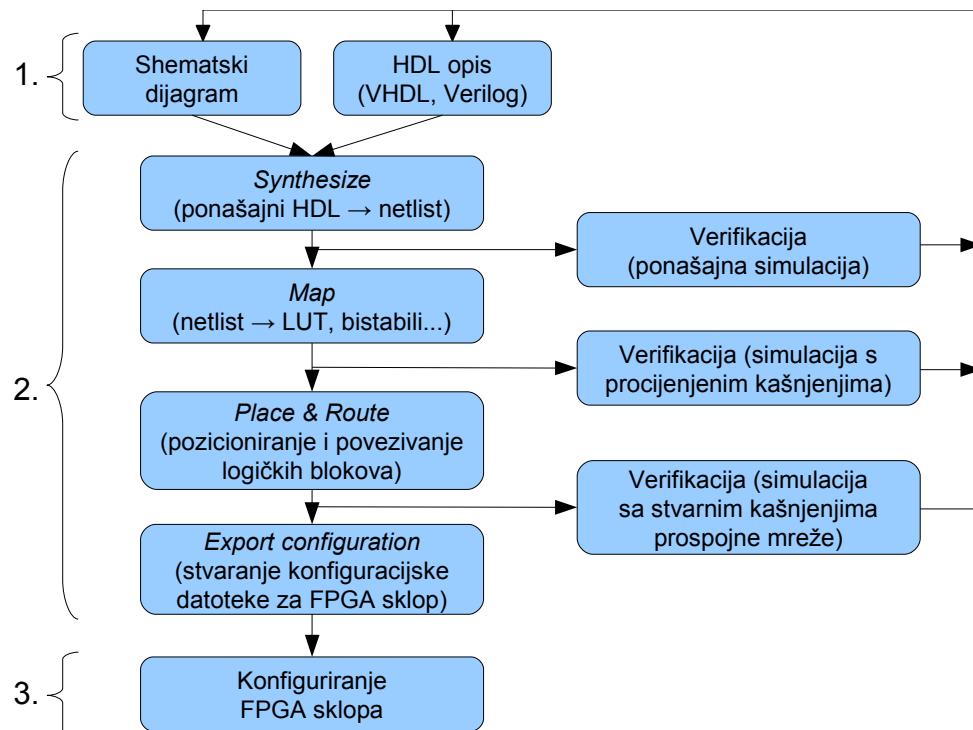
Veličine polja današnjih FPGA sklopova kreću su u rasponu od oko 1.000 do 500.000 preglednih tablica (LUT) odnosno logičkih elemenata. Kašnjenja pojedine pregledne tablice (LUT) tipično su reda veličine od 50 do 100 ps, kašnjenja bistabila su reda veličine od 200 do 500 ps, dok kašnjenja koja unosi prospojna mreža mogu doseći i do nekoliko ns. Bitno je uočiti da svaki logički blok FPGA sklopa radi paralelno i potpuno nezavisno od drugih, za razliku od računala opće namjene u kojima jezgra mikroprocesora slijedno izvršava programske instrukcije jednu za drugom.

Uz velik broj generičkih logičkih blokova temeljenih na preglednim tablicama (LUT) te posebnih elemenata vezanih uz programirljiva vanjska ulazno-izlazna sučelja, moderni FPGA sklopovi obično raspolažu i s manjim brojem dodatnih specijaliziranih elemenata kao što su blokovi staticke RAM memorije, blokovi za množenje, sintetizatori takta, blokovi za izvedbu sučelja prema vanjskim dinamičkim RAM memorijama (DDR), interna Flash memorija i slično. Sve današnje FPGA platforme raspolažu i sa specijaliziranim signalima koji povezuju susjedne logičke blokove, a koji su prvenstveno namijenjeni brzom prijenosu bita preljeva kod implementacije višebitnih blokova za zbrajanje ili oduzimanje (tzv. *fast carry chain*).

1.2 Postupci opisa, sinteze i programiranja konfiguracije FPGA sklopa

Konfiguracija FPGA sklopova, dakle logičkih blokova i prospojne mreže, **u pravilu se generira strojno** uz pomoću specijaliziranih programske alata, tzv. sintetizatora (*eng. synthesis tools*), a na temelju opisa sklopa specificiranog shemom ili korištenjem jezika za opis digitalnih sklopova (*eng. hardware description language - HDL*). Kako svaka kategorija FPGA sklopova svakog proizvođača ima različitu internu strukturu i arhitekturu, tako se razlikuju i alati za sintezu, a vlastite algoritme za sintezu konfiguracija svojih FPGA sklopova svaki proizvođač brižno čuva od konkurenčije. Drugim riječima, za projektiranje digitalnih sklopova i sustava koji će se sintetizirati na FPGA sklopovima **nije potrebno poznavati implementacijske detalje ciljane FPGA platforme**, ali je korisno razumjeti njihovu strukturu kako bi se opis želenog sklopa formulirao na način koji omogućuje sintetizatoru stvaranje optimalne konfiguracije logičkih blokova i prospojne mreže.

Rad s programskim alatima za sintezu konfiguracije FPGA sklopova može se podijeliti u tri glavne cjeline, kao što je prikazano slikom.



Slika 4: koraci pri opisu sklopa, sinteze konfiguracije i programiranja FPGA sklopa

Prvi korak je opis želenog sklopa na način koji je prikladan za daljnju strojnu obradu, najčešće unosom shematskog dijagrama ili korištenjem jezika za opis digitalnih sklopova, od kojih su danas najšire prihvaćeni i podržani VHDL i Verilog.

Sinteza konfiguracije FPGA sklopa na temelju shematskog ili HDL opisa provodi se u više koraka. U koraku koji je u većini programskih alata nazvan "Synthesize" analizira se opis sklopa te se ponašajno opisani blokovi zamjenjuju funkcionalnim ekvivalentnim strukturnim modelima izgrađenim od primitiva specifičnih za ciljanu FPGA platformu, pri čemu se provodi i automatizirana optimizacija kombinacijske i sekvensijske logike. Rezultirajuća mreža primitiva, koja se uobičajeno naziva *netlist*, može se analizirati korištenjem odgovarajućeg simulatora, pri čemu se ne modeliraju kašnjenja primjenjenih primitiva.

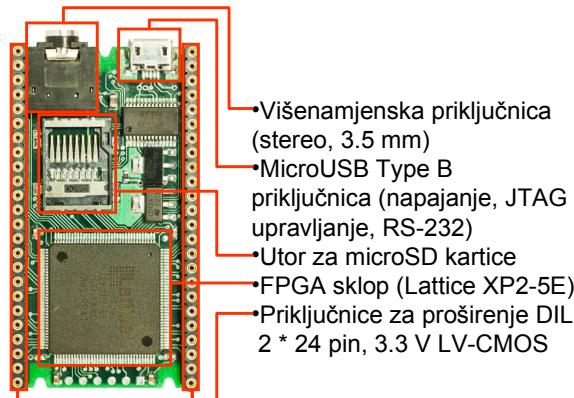
U slijedećem koraku, koji se uobičajeno naziva "*Map*", sintetizator primitive (npr. AND_2, OR_8, MUX_16_1) zamjenjuje mrežom preglednih tablica (LUT), memorijskih elemenata te ostalih specijaliziranih logičkih blokova koji odgovaraju ciljanoj FPGA platformi. U ovom koraku sintetizator može napraviti i simulacijski model sklopa koji uključuje grubu procjenu kašnjenja pri propagaciji signala, budući da u ovom koraku još nisu poznata kašnjenja prospojne mreže.

U koraku nazvanom "*Place and Route*" sintetizator pokušava rasporediti logičke blokove na FPGA sklopu i povezati ih na način koji će omogućiti ispravan rad sklopa na najvišoj mogućoj frekvenciji takta. Zavisno od kompleksnosti opisanog sklopa, ciljane frekvencije takta, postavki *place and route* algoritma te brzine računala na kojem se izvodi algoritam, ovaj dio postupka sinteze može potrajati od nekoliko desetaka sekundi do više sati, pa i dana. Po završetku ovog koraka sintetizator može napraviti simulacijski model koji uključuje vrlo preciznu procjenu kašnjenja svih logičkih elemenata i prospojne mreže. Analizom takvog simulacijskog modela mogu se pronaći kritične točke projektiranog sklopa koje ograničavaju brzinu rada (najslabije karike u lancu), te se na temelju analize može pristupiti preinakama opisa sklopa.

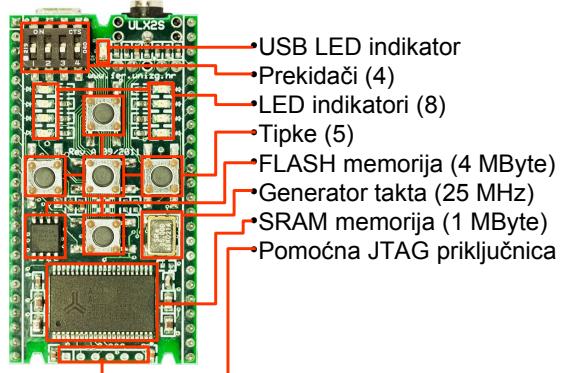
Krajnji rezultat procesa sinteze je konfiguracijska datoteka (*eng. configuration bitstream*) kojom se pomoću odgovarajućeg alata može programirati FPGA sklop, odnosno pohraniti u njegovu trajnu memoriju (Flash) iz koje će sklop automatski učitati konfiguraciju prilikom uspostave napajanja.

2 Komponente razvojne pločice

Na slikama 5 i 6 prikazan je fizički raspored komponenata na pločici.



Slika 5: donja strana pločice



Slika 6: gornja strana pločice

Dimenziije pločice su 64 * 36 * 9 mm.

Pločica se s računalom povezuje putem USB Type A na micro-USB Type B priključnog kabela, koji se **ne isporučuje** zajedno s pločicom.

2.1 Napajanje

Pločica je projektirana za napajanje putem USB priključka, ili uz određena ograničenja putem DIL podnožja za proširenje. Nominalni napon napajanja na USB priključku je 5 V, a pločica će ispravno raditi uz ulazni napon u rasponu od 4.3 V do 6.3 V. Dovođenje napona višeg od 6.3 V na USB priključak može rezultirati oštećenjem komponenti.

Na pločicu su ugrađeni linearni regulatori napajanja za 3.3 V i 1.2 V. Napon od 3.3 V koristi se za napajanje svih ulazno-izlaznih sučelja FPGA sklopa, generatora takta, SRAM i Flash memorije, te MicroSD kartice, dok se 1.2 V koristi isključivo za napajanje jezgre FPGA sklopa. Ukupna potrošnja struje zavisi od kompleksnosti konfiguracije i frekvencije takta FPGA sklopa, te o statičkom i dinamičkom opterećenju izlaznih signala, a može se kretati u rasponu od 55 mA do 300 mA ili više. Kako linearni regulatori razliku između ulaznog i izlaznog napona pretvaraju u toplinsku energiju koja je proporcionalna struci kroz regulator, kod kompleksnijih konfiguracija FPGA sklopa regulatori napajanja, pa i FPGA sklop, mogu se osjetno zagrijati, što međutim ne utječe na njihov rad.

Pločica nema ugrađenu vlastitu zaštitu od eventualnog kratkog spoja, već se oslanja na zaštitne mehanizme ugrađene u USB sučelja na računalima koja ograničavaju potrošnju priključenih uređaja na najviše 500 mA. Ukoliko potrošnja USB uređaja premaši razinu od 500 mA, računalo će automatski privremeno prekinuti napajanje na USB sučelju, bez oštećenja sučelja. Međutim, dodatni oprez je potreban kod priključenja pločice na računalo putem USB *huba* s vlastitim napajanjem, ili na nezavisne izvore napajanja s USB priključkom koji se često koriste za punjenje baterija mobilnih telefona i sličnih uređaja potrošačke elektronike, budući da takvi uređaji ponekad nemaju ugrađene sklopove za zaštitu od kratkog spoja.

2.2 FPGA sklop

(opis XP2, glavna svojstva, I/O konfiguracija, ograničeni broj pisanja po Flash memoriji)

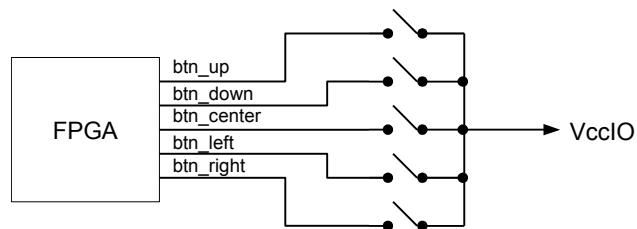
2.3 JTAG

(shema, blinkanje USB LED indikatora)

2.4 Tipke

Pritisak na svaku od pet tipki (btn_up, btn_down, btn_center, btn_left, btn_right) povezuje odgovarajuću priključnicu FPGA sklopa direktno na visoku naponsku razinu (VccIO). Kad tipke nisu pritisnute, signali će biti pritegnuti niskoj naponskoj razini putem internih *pulldown* otpornika, koji su definirani *I/O preference* opisom (poglavlje 6).

Shema povezivanja:



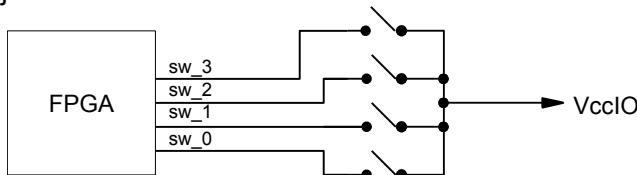
Deklaracija signala u sučelju sklopa (VHDL):

```
port (
    btn_left: in std_logic;
    btn_right: in std_logic;
    btn_up: in std_logic;
    btn_down: in std_logic;
    btn_center: in std_logic
);
```

2.5 Prekidači

U položaju "ON" svaki od četiri prekidača povezuje odgovarajuću priključnicu FPGA sklopa direktno na visoku naponsku razinu (VccIO). U položaju "OFF" signali će biti pritegnuti na nisku naponsku razinu putem internih *pulldown* otpornika, koji su definirani *I/O preference* opisom (poglavlje 6).

Shema povezivanja:



Deklaracija signala u sučelju sklopa (VHDL):

```
port (
    sw: in std_logic_vector(3 downto 0)
);
```

2.6 LED indikatori

(shema)

```
port (
    sw: out std_logic_vector(7 downto 0)
);
```

2.7 Višenamjenska stereo priključnica

(shema otporničke mreže)

```
port (
    p_ring: out std_logic;
    p_tip: out std_logic_vector(3 downto 0)
);
```

2.8 RS-232 sučelje

(shema)

```
port (
    rs232_tx: out std_logic;
    rs232_rx: in std_logic
);
```

2.9 Generator takta (oscilator)

(shema)

```
port (
    clk_25m: in std_logic
);
```

2.10 SRAM

(shema, stanja sabirnice za WEL, UBL, LBL)

```
port (
    sram_a: out std_logic_vector(18 downto 0);
    sram_d: inout std_logic_vector(15 downto 0);
    sram_wel: out std_logic;
    sram_lbl: out std_logic;
    sram_ubl: out std_logic
);
```

2.11 Flash

(shema)

```
port (
    flash_so: in std_logic;
    flash_cen: out std_logic;
    flash_sck: out std_logic;
    flash_si: out std_logic
);
```

2.12 MicroSD

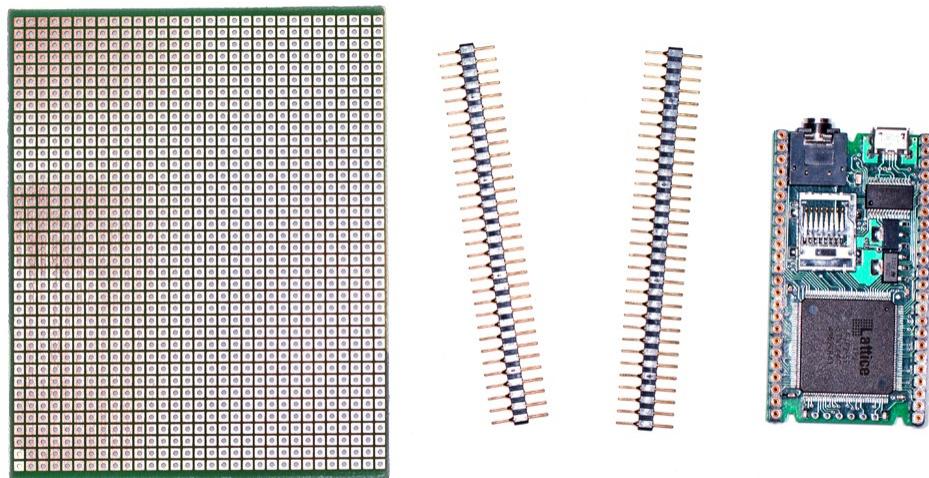
(shema)

```
port (
    sdcard_so: in std_logic;
    sdcard_cen: out std_logic;
    sdcard_sck: out std_logic;
    sdcard_si: out std_logic
);
```

2.13 DIL priključnice za proširenje

(shema, označiti dijeljene signale, slika konektora, ograničenja kod povezivanja s vanjskim sklopovima)

```
port (
    dil: inout std_logic_vector(47 downto 0)
);
```



3 Instaliranje programske podrške

Programsko okruženje za sintezu digitalnih sklopova Lattice Diamond dostupno je za operacijske sustave Microsoft Windows XP i Windows 7, te za operacijski sustav Linux, uz licencu za besplatno korištenje za rad s odabranim kategorijama FPGA i CPLD sklopova proizvođača Lattice Semiconductor. Uz verziju za Microsoft operacijske sustave isporučuje se i alat za simulaciju digitalnih sklopova Aldec Active-HDL. Lattice Diamond za rad zahtijeva računalo s najmanje 1 GByte radne memorije (RAM). U nastavku je dan pregled najvažnijih koraka za instalaciju programskih alata potrebnih za rad s razvojnom pločicom na operacijske sustave Windows XP i Windows 7, te kratke upute za instalaciju na operacijske sustave Linux i FreeBSD.

3.1 Instaliranje programskog paketa Lattice Diamond

S web sjedišta tvrtke Lattice Semiconductor (<http://www.latticesemi.com/>) potrebno je dohvatiti instalacijski paket alata Diamond za Microsoft Windows operacijske sustave.

The screenshot shows a Mozilla Firefox window displaying the Lattice Semiconductor website. The URL in the address bar is <http://www.latticesemi.com/products/designsoftware/diamond/downloads.cfm>. The main content area is titled "Lattice Diamond Downloads & Licensing". On the left, there's a sidebar with "In Detail" sections like Features, Key Concepts, Feature List, Videos, and Downloads (which is currently selected). The main content area has a "Windows Software" table with the following rows:

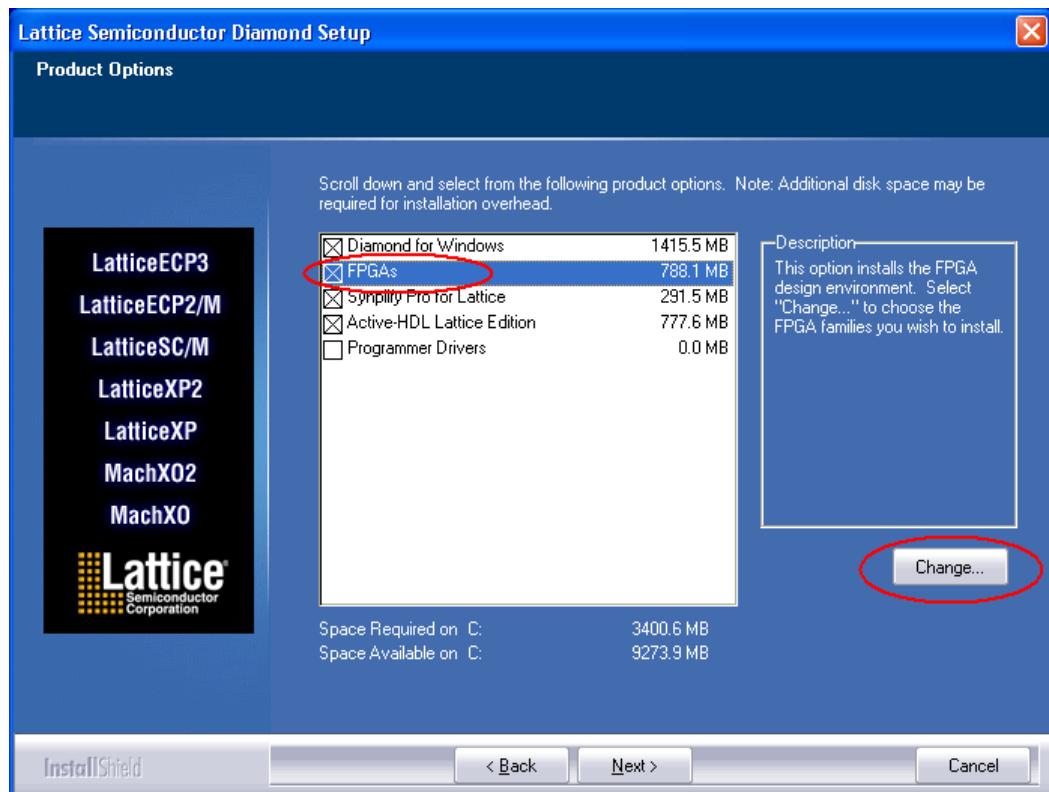
Lattice Diamond Windows Software	
Windows Software	Description
Diamond for Windows	Lattice Diamond installer for Windows (XP, Vista 32-bit, 7 32-bit & 64-bit)
LatticeMico System for Windows	LatticeMico System Installer for use with Diamond for Windows (XP, Vista 32-bit, 7 32-bit & 64-bit)
Standalone Reveal for Windows	Optional Standalone Reveal installer for use with Diamond for Windows (XP, Vista 32-bit, 7 32-bit & 64-bit)
Standalone ORCAstra for Windows	Optional Standalone ORCAstra on Windows (XP, Vista 32-bit, 7 32-bit & 64-bit)
Standalone Programmer for Windows	Optional Standalone Programmer on Windows (Vista 32-bit, 7 32-bit & 64-bit)
Standalone Programmer Encryption Installer for Windows	Optional encryption security installer for use with Standalone Programmer for Windows (XP, Vista 32-bit, 7 32-bit & 64-bit) and secure Lattice devices. Be sure to install the software prior to installing this option. Note: Due to US Export regulations, you must first request a download link by clicking the link above. The download link ar...

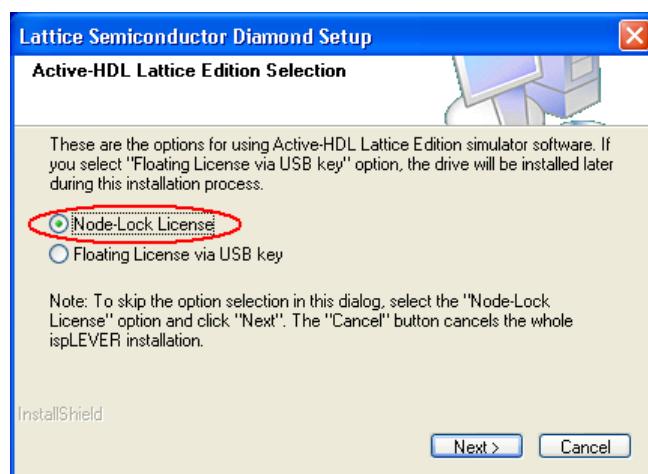
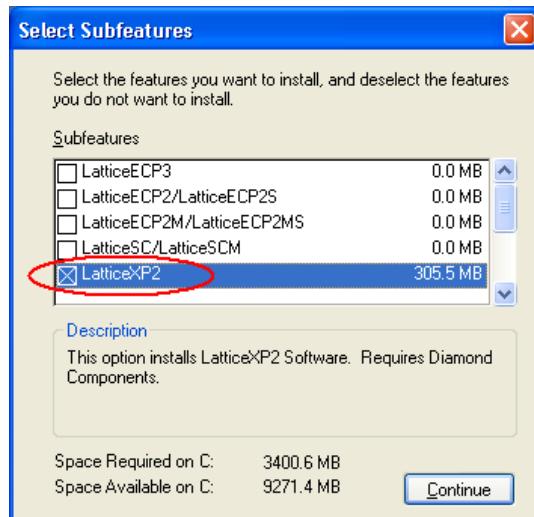
Za pristup dijelu web sjedišta koje omogućuje dohvaćanje instalacijskih paketa, prethodno je potrebno izvršiti *online* registraciju. Veličina instalacijskog paketa je cca. 1.8 GB. Nakon što je instalacijski paket u obliku *zip* arhive dohvaćen na lokalni disk računala, *zip* arhivu je potrebno raspakirati, te pokrenuti instalacijsku .exe datoteku.

Instalacija programskog paketa s pretpostavljenim (*default*) postavkama zahtijeva cca. 6 GB slobodnog prostora na lokalnom disku. Kako instalacijski paket uključuje module za podršku svim FPGA sklopovima proizvođača Lattice Semiconductor, **zauzeće diska može se bitno smanjiti (za oko 3 GB) ako se pri instalaciji odabere samo podrška za Lattice XP2 seriju FPGA sklopova**, koji su ugrađeni u ULX2S razvojne pločice. Također, **nije potrebno instalirati upravljačke programe (drive)** za JTAG programatore tvrtke Lattice Semiconductor, budući da pločica ULX2S koristi vlastiti integrirani USB JTAG sklop za programiranje, koji zahtijeva posebnu programsку podršku.

Prilikom instalacije potrebno je odabrati "**node locked**" model provjere licence za instaliranu programsku podršku. Pritiskom na tipku "Next" završava se postupak instalacije. Nakon uspješno završene instalacije, potrebno je ispuniti *online* zahtjev za dodjelu besplatne licence za korištenje Lattice Diamond programskog paketa. Besplatna licenca vezana je uz MAC adresu Ethernet sučelja računala, pa je prilikom ispunjavanja *online* zahtjeva za licencu potrebno slijediti upute s web sjedišta latticesemi.com o načinu i formatu prijave odgovarajuće MAC adrese. Nakon što električkom poštrom pristigne datoteka s besplatnom licencom, datoteku **license.dat** potrebno je spremiti u direktorij **c:\lsccl\diamond\1.3\license**.

U nastavku su prikazani najvažniji izbornici instalacijskog postupka.





Diamond Software Free Licensing Request - Mozilla Firefox

File Edit View History Bookmarks Tools Help

<http://www.latticesemi.com/licensing/flexlmlicense.cfm?p=diamond>

Lattice Semiconductor Corporation

Welcome Marko Zec ▶ Your Account ▶ Sign Out

Home | Products | Solutions | Support | Documents | Downloads | Sales | Store | About Us |

Home > Products > Design Software > Licensing > Diamond Software Free

Diamond Software Free Licensing Request

Please follow these steps to request your Lattice Diamond Software Free license:

1. Review and update your Web Account information below by clicking on the Update Your Account button.
2. Fill in the Software License Request Form and Submit.

Host NIC: You can find your PC Network Interface Card ID by opening an MS-DOS window, typing "ipconfig /all" and pressing Enter.
The Physical Address is a 12-digit hexadecimal value split into pairs with dashes, like this: 00-01-02-66-1D-E0.
Please enter this information without dashes like this: 000102661DE0.

After you successfully complete and submit this form, a new license file with instructions on how to install it will be e-mailed to you immediately.

For any questions, email lic_admn@latticesemi.com
If you have experienced an Aldec license check out error, please check out a new license.

Software License Request Form

Note: The license file will be sent to the web account email address: zec@fer.hr

Host NIC (physical address): * required field

I verify that I am not an employee of Cadence Design Systems, Mentor Graphics Corporation, or Magma Design Automation

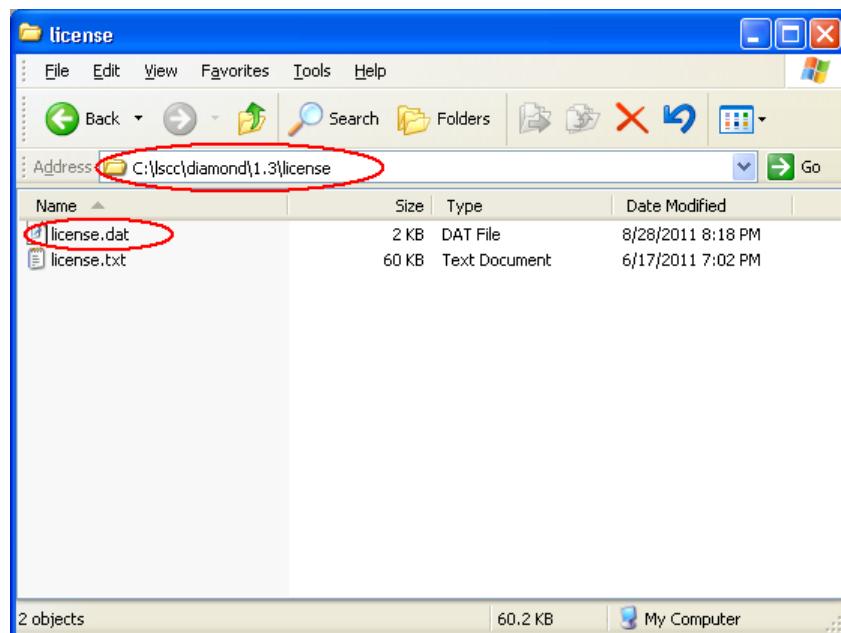
Generate License (click once)

NOTE: This form requires JavaScript to be enabled in your web browser

Your Web Account Info

Name:	Marko Zec
Company:	University of Zagreb
Address 1:	Unska 3
City:	Zagreb
State:	
Zip/Postal Code:	HR10000
Country:	Croatia
Telephone:	38516129737
Email Address:	zec@fer.hr

Update Your Account Info



3.2 Instaliranje programskog alata za programiranje pločice

Razvojna pločica ULX2S ima ugrađeno USB sučelje za programiranje koje nije podržano *driverima* za programatore proizvođača Lattice Semiconductor, zbog čega zahtijeva instalaciju posebnog upravljačkog programskog alata. Program "ujprog.exe" potrebno je dohvatiti s web sjedišta <http://www.nxlab.fer.hr/dl/ujprog.exe> i spremiti ga u direktorij <c:\windows>. Za pisanje po direktorijima operacijskog sustava korisnik treba imati administratorske ovlasti.

Program "ujprog.exe" može ispravno raditi samo ako su na operacijski sustav instalirani i USB pogonski programi (*driveri*) za razvojnu pločicu.

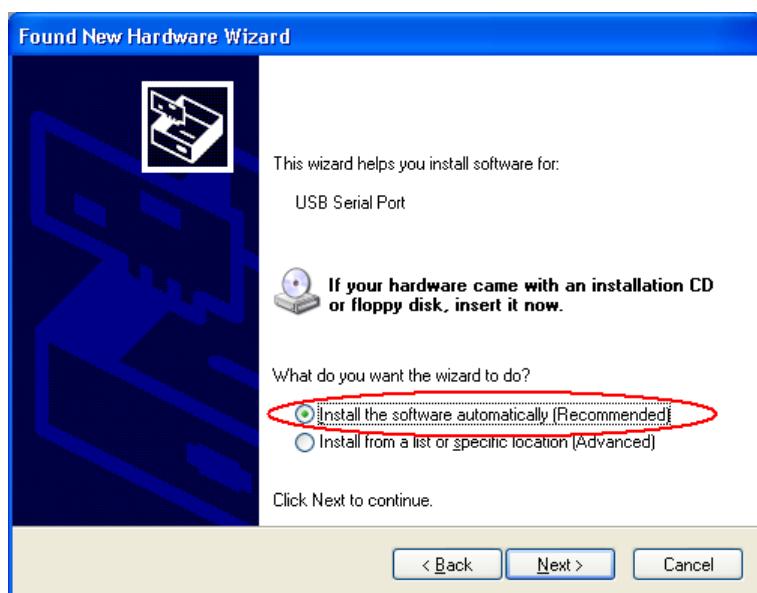
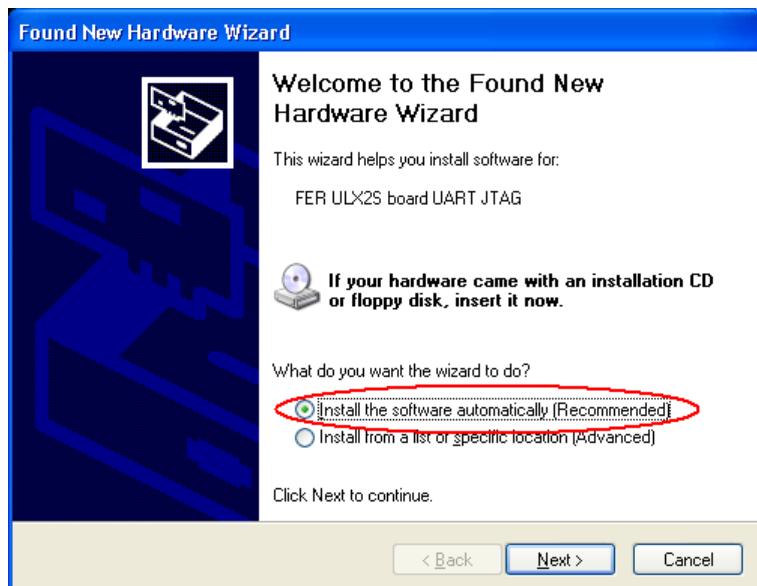
3.3 Instaliranje USB pogonskih programa (*drivera*) za razvojnu pločicu

Instaliranje USB pogonskih programa (*drivera*) za razvojnu pločicu ULX2S odvija se automatski po priključenju pločice na USB priključnicu računala. U trenutku instalacije računalo mora imati omogućen pristup Internet mreži, kako bi operacijski sustav mogao dohvatiti *driveere* s odgovarajućih poslužitelja.

Tijek pojavljivanja instalacijskih izbornika razlikuje se između operacijskih sustava Windows XP i Windows 7, te zavisi od postavki operacijskog sustava. Ukoliko je na operacijski sustav Windows XP već ranije bio instaliran USB uređaj temeljen na sličnim ili istim komponentama koje su primjenjene na pločici ULX2S (USB/RS-232 sučelja tvrtke FTDI), operacijski sustav će registrirati razvojnu pločicu bez pojave izbornika koji zahtijevaju intervenciju korisnika. Ukoliko je korisnik prijavljen na sustav s administratorskim ovlastima, operacijski sustav Windows 7 bi *driveere* trebao instalirati automatski, bez intervencije korisnika.

U nastavku je prikazan tipični tijek pojavljivanja izbornika prilikom instalacije *drivera* za razvojnu pločicu na operacijskom sustavu Windows XP.





3.4 Instaliranje programskih alata na operacijski sustav Linux

S web sjedišta tvrtke Lattice Semiconductor (<http://www.latticesemi.com/>) potrebno je dohvatiti instalacijski paket alata Diamond 1.3 za operacijski sustav Linux. U direktorij s instalacijskim paketom potrebno je spremiti i datoteku s licencem za korištenje paketa Lattice Diamond, te alat "ujprog" za operacijski sustav Linux, kojeg treba dohvatiti s web sjedišta <http://www.nxlab.fer.hr/dl/>.

(XXX daljnji postupak: TBD)

3.5 Instaliranje programskih alata na operacijski sustav FreeBSD 8.2

S web sjedišta tvrtke Lattice Semiconductor (<http://www.latticesemi.com/>) potrebno je dohvatiti instalacijski paket alata Diamond 1.3 za operacijski sustav Linux. U direktorij s instalacijskim paketom potrebno je spremiti i datoteku s licencem za korištenje paketa Lattice Diamond, te alat "ujprog" za operacijski sustav FreeBSD, kojeg treba dohvatiti s web sjedišta <http://www.nxlab.fer.hr/dl/>. Za instaliranje je potrebno izvršiti sljedeći naredbe s administratorskim ovlastima:

```
klldload linux
echo linux_enable="YES" >> /etc/rc.conf

pkg_add -r linux_base-f10 linux-f10-openmotif linux-f10-jpeg linux-f10-png
pkg_add -r bash libftdi

ln -s /usr/local/bin/bash /bin/bash
ln -s /compat/linux/usr/lib/libXm.so /compat/linux/usr/lib/libXm.so.3

rpm -i --nodeps --noscripts --ignoreos --ignorearch --prefix /usr/local \
diamond_1_3-base-92-i386-linux.rpm

cp license.dat /usr/local/diamond/1.3/license/license.dat
chmod +sx ujprog
mv ujprog /usr/local/bin
```

Alat Lattice Diamond pokreće se odabirom punog puta do izvršne datoteke:

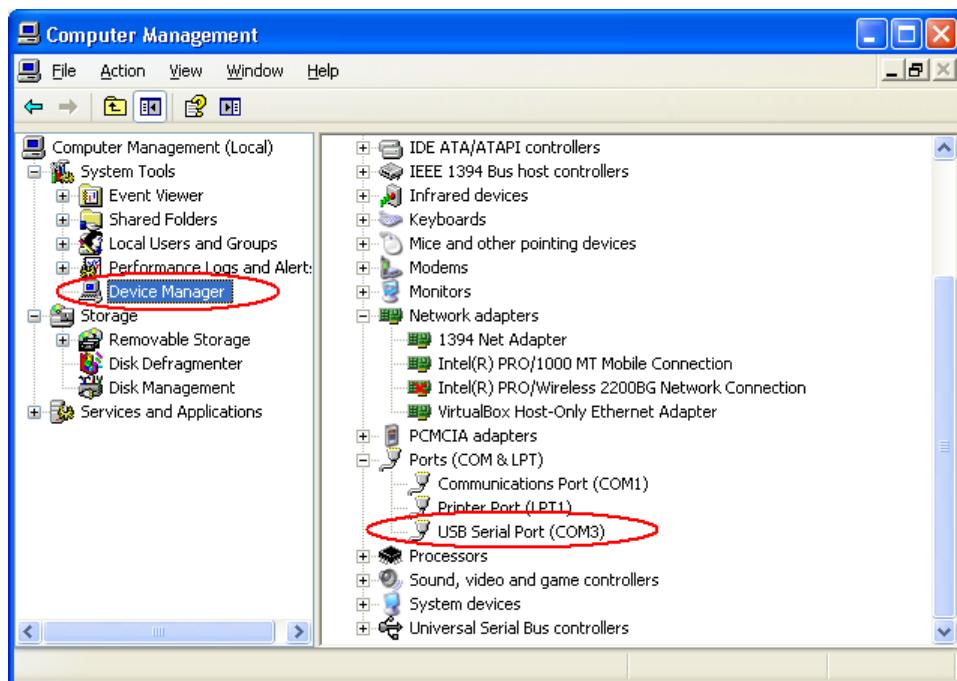
```
% /usr/local/diamond/1.3/bin/lin/diamond &
```

4 Ispitivanje rada pločice

Razvojna pločica isporučuje se programirana dijagnostičko-demonstracijskom konfiguracijom FPGA sklopa, koja omogućuje ispitivanje osnovne funkcionalnosti FPGA sklopa, pomoćnih integriranih sklopova ugrađenih na pločicu (generator takta, Flash i SRAM memorija), te ulazno / izlaznih priključaka i signala (RS-232, LED indikatori, tipke, višenamjenska 3.5 mm stereo priključnica). U nastavku je opisan način ispitavanja osnovnih funkcionalnosti razvojne pločice.

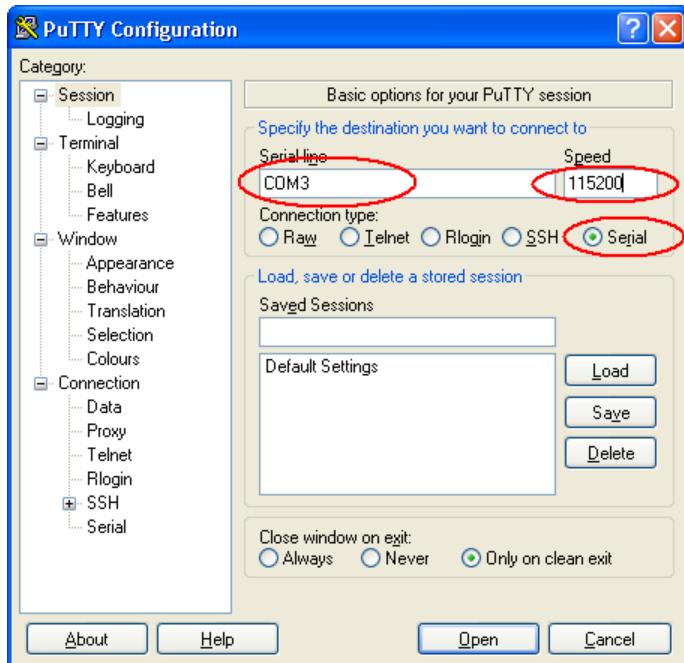
4.1 Ispitivanje funkcionalnosti USB RS-232 sučelja

Po uspješno završenoj proceduri instalacije USB *drivera*, pločica ULX2S bi operacijskom sustavu trebala biti vidljiva kao standardni USB serijski (RS-232) port. Logički broj *COM porta* koji je operacijski sustav dodijelio razvojnoj pločici može se vidjeti kroz *Device Manager* izbornik *Computer Management* prozora. U operacijskom sustavu Windows XP, *Computer Management* prozor može se otvoriti lijevim klikom na glavni izbornik *Start*, te otvaranjem izbornika desnim klikom nad *My Computer*, u kojem se odabere opcija *Manage*.



Za komunikaciju s razvojnom pločicom putem USB-RS-232 sučelja može se koristiti bilo koji program za emulaciju asinkronog terminala. Preporuka je koristiti program **Putty**, koji je dostupan za besplatno dohvaćanje na web sjedištu <http://www.putty.org>. Na računalima s operacijskim sustavom Windows XP možemo koristiti i program **Hyper Terminal**, kojeg se može pozvati kroz izbornike *Start – All Programs – Accessories – Communications – Hyper Terminal*.

Prilikom konfiguracije programa za emulaciju asinkronog terminala bitno je odabrati *COM port* koji je dodijeljen razvojnoj pločici (u ovom primjeru to je COM3), odabrati brzinu prijenosa od 115200 bps, te isključiti sklopošku kontrolu toka (*flow control*) ukoliko je uključena. U nastavku je dan primjer uspostave terminalske sesije s razvojnom pločicom korištenjem programa *Putty*.



Pritiskom na tipku *Enter*, u prozoru terminalskog emulatora trebao bi se pojaviti glavni izbornik demonstracijsko-dijagnostičke konfiguracije FPGA pločice.

```

COM3 - PuTTY
FER - Digitalna logika 2011/2012
ULX2S FPGA pločica - demonstracijsko-dijagnosticki program
v 0.02 27/09/2011

Glavni izbornik:

1: Audio izlaz ukljucen: 1
2: Glasnoca: 8
3: Balans (L/D): 0
4: Brzina reprodukcije: 100%
5: Frekvencija odasiljanja FM signala: 0.0000 MHz
6: LED indikatori (0: VU-metar, 1: byte): 0
7: LED byte: 0
8: USB UART (RS-232) baud rate: 115200 bps
9: SRAM self-test

```

Odabirom opcija iz izbornika moguće je upravljati demonstracijskim funkcijama. Na primjer, odabirom opcije 6 možemo isključiti treptanje LED indikatora, te umjesto toga prikazati vrijednost koju upišemo kroz opciju 7. Odabirom opcije "SRAM self-test" može se ispitati ispravnost rada statičke RAM memorije ugrađene na pločicu.

4.2 Ispitivanje funkcionalnosti višenamjenske stereo priključnice

Zavisno od konfiguracije FPGA sklopa, stereo priključnica ugrađena na pločicu može služiti i kao ulazno i kao izlazno sučelje. Demonstracijsko-dijagnostička konfiguracija omogućuje ispitivanje funkcionalnosti višenamjenske stereo priključnice kao izlaznog sučelja za reprodukciju zvuka, ili kao sučelja za izlaz kompozitnog analognog video signala u formatu PAL.

Odabir izlaznog signala na stereo priključku obavlja se putem mikroprekidača iznad USB utičnice označenog brojem 1. Ako je prekidač u položaju "isključeno" (poluga prekidača pomakнута je prema dolje), na stereo utičnicu dovoditi će se zvučni signal, odnosno reprodukcija kratkog tonskog zapisa pohranjenog na Flash memoriju. Zvučni signal moguće je reproducirati priključenjem standardnih slušalica, ili priključenjem na aktivni zvučnički sustav. Glasnoća signala može se podešavati tipkama gore odnosno dolje, a balans tipkama lijevo odnosno desno. Pritisom na srednju tipku prekida se odnosno omogućuje izlaz zvuka. Parametre reprodukcije zvučnog zapisa također je moguće podešavati putem izbornika dostupnog na USB – RS-232 sučelju.

Ukoliko je prekidač br. 1. postavljen u položaj "uključeno" (poluga prekidača pomakuta prema gore), na jednom kanalu višenamjenske stereo priključnice generirati će se kompozitni analogni video signal u formatu PAL, a na drugom kanalu kratki tonski signali. Za reprodukciju video signala pločicu je potrebno povezati s ekranom ili TV prijemnikom s ulazom za kompozitni analogni PAL video signal. Za povezivanje se može koristiti 3.5 mm stereo na 2 * RCA (cinch / phono) adapter / kabel, koji se često isporučuje u kompletu s digitalnim fotoaparatima, kamerama i / ili mobilnim telefonima, a može se nabaviti i zasebno u većini trgovina električkom opremom.

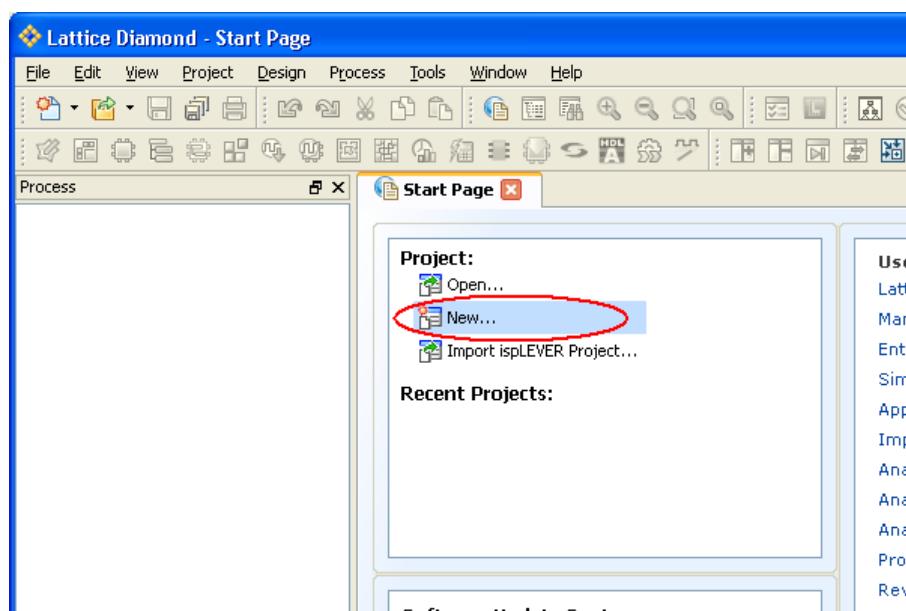


5 Rad s razvojnim alatom Lattice Diamond

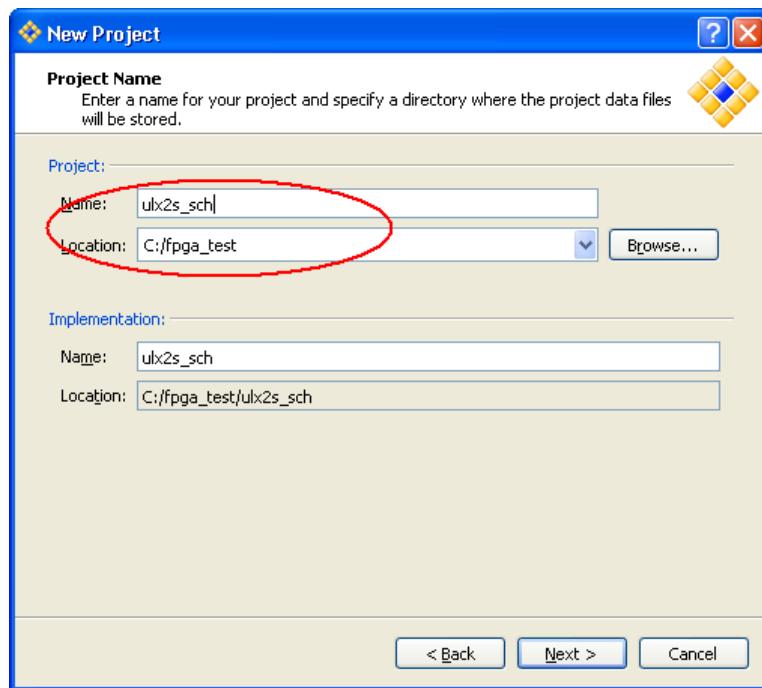
Glavna namjena programskog alata Lattice Diamond je sinteza konfiguracije FPGA sklopova iz odgovarajućeg opisa odnosno specifikacije digitalnog sklopa. Alat omogućuje opis digitalnog sklopa unosom shematskih dijagrama, ili korištenjem jezika za opis sklopova VHDL ili Verilog. U nastavku je dan primjer korištenja alata kroz nekoliko faza, od kreiranja projekta, unosa odnosno uređivanja shematskog opisa sklopa, sinteze konfiguracije FPGA sklopa, te programiranja konfiguracijskog *bitstreama* na razvojnu pločicu ULX2S, za što se koristi dodatni programski alat.

5.1 Kreiranje novog projekta

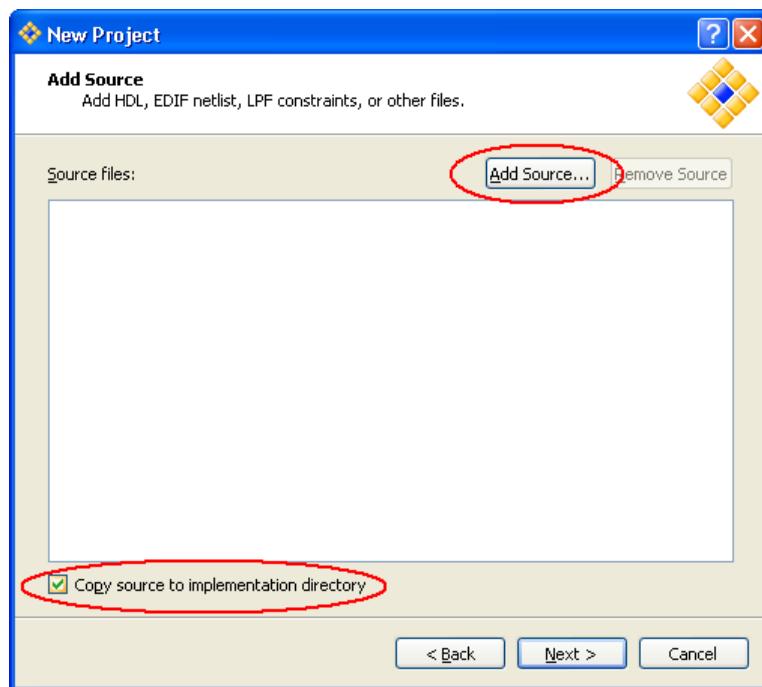
Pokrenite alat Lattice Diamond. Odaberite kreiranje novog projekta: *Project -> New*



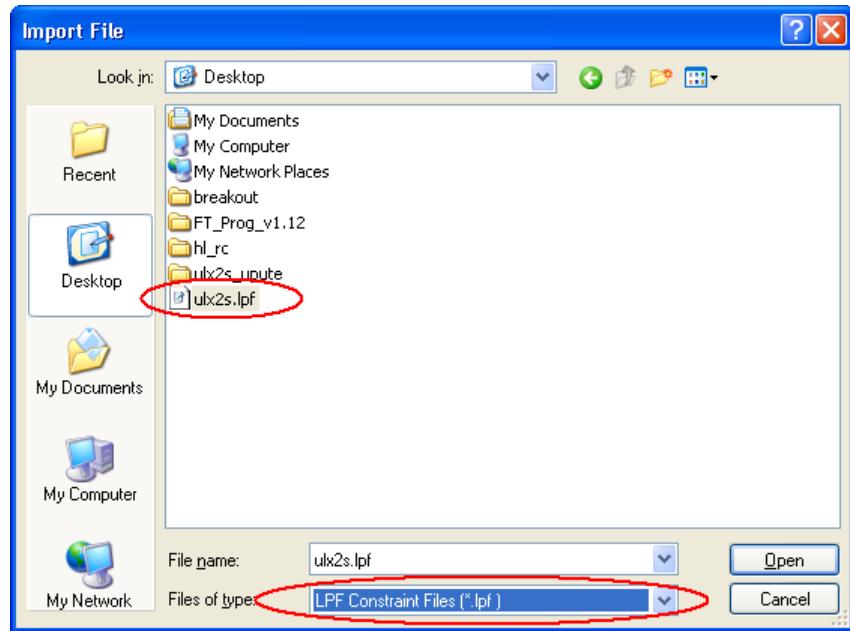
Odaberite ime novog projekta, te radni direktorij u kojem će biti pohranjene ulazne datoteke vezane uz projekt. U ovom primjeru projekt je nazvan "**ulx2s_sch**", a koristi se direktorij "**c:\fpga_test**". Većina pomoćnih datoteka koje će programski alat kreirati tijekom svog rada, kao i izlazna *bitstream* konfiguracijska datoteka FPGA sklopa, biti će spremljene u poddirektoriju "**c:\fpga_test\ulx2s_sch**".



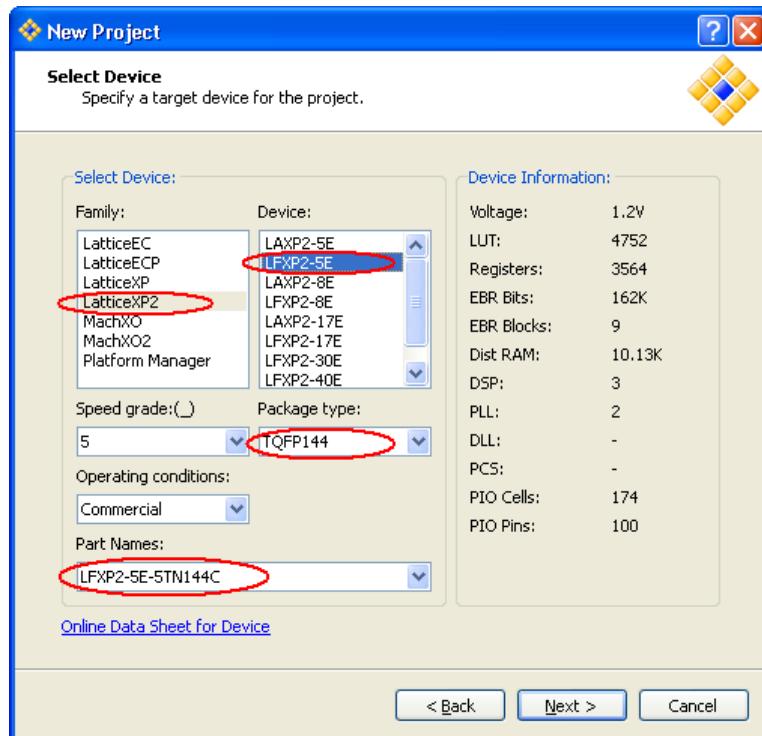
U sljedećem koraku alat nudi izbornik za dodavanje već postojećih (unaprijed pripremljenih) datoteka u novi projekt.



U ovom koraku potrebno je u projekt uključiti datoteku koja određuje povezivanje između logičkih naziva ulazno-izlaznih signala (npr. LED_0) i fizičkih priključaka FPGA sklopa (npr. pin broj 29). Datoteku s definicijom svih logičkih signala može se dohvatiti s web sjedišta <http://www.nxlab.fer.hr/dl/ulx2s.lpf>. Ispis sadržaja datoteke ulx2s.lpf nalazi se u poglavljiju 6 ovog dokumenta.



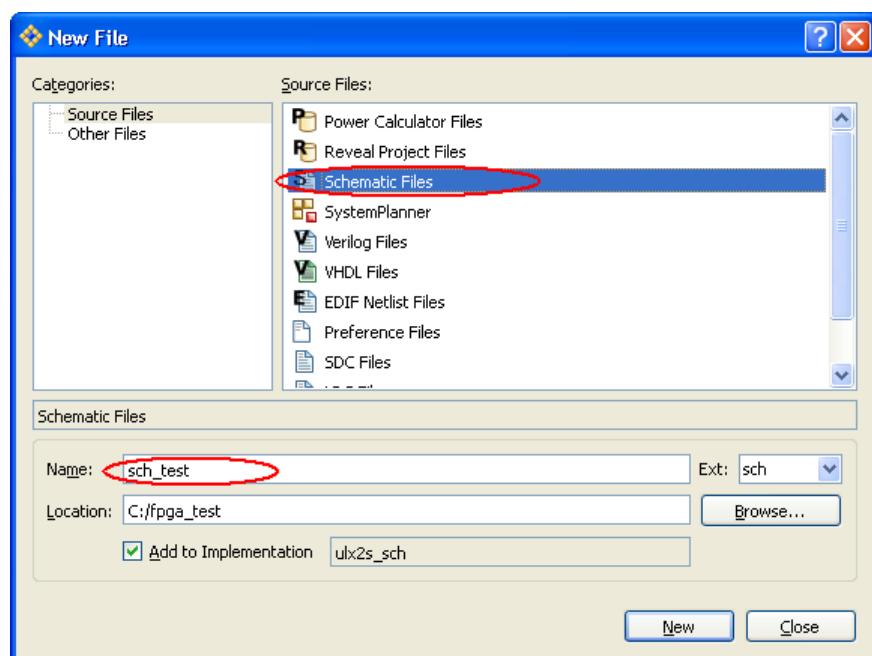
U slijedećem koraku potrebno je odabratи točan tip FPGA sklopa. Na pločicu ULX2S ugrađen je FPGA sklop Lattice XP2-5E, brzine rada (*speed grade*) 5, u kućištu TQFP144.



Odabirom tipa FPGA sklopa završava postupak stvaranja novog Lattice Diamond projekta (tipka "Finish").

5.2 Shematski opis digitalnog sklopa

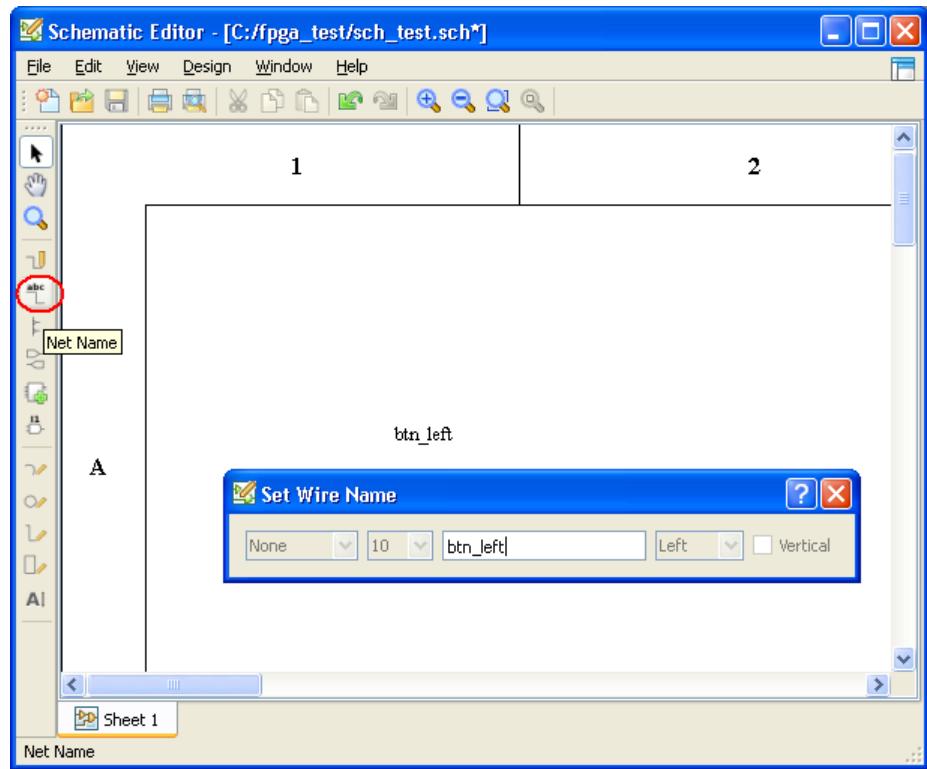
Kako bi mogli shematski opisati željeni digitalni sklop, potrebno je prvo stvoriti novu praznu *schematic file* datoteku i uključiti ju u trenutni projekt. Postupak za stvaranje novih VDHL ili Verilog datoteka i njihovo uključivanje u projekt je identičan, međutim zbog preglednosti i jednostavnosti u ovom primjeru prikazan je isključivo shematski način unosa odnosno opisa digitalnog sklopa. Otvaranje izbornika za dodavanje novih datoteka u projekt može se pokrenuti kroz izbornik *File – New – File*, ili pritiskom na tipke CTRL+N. U izborniku je potrebno odabrati "Source files" / "Schematic files", te datoteci dodijeliti proizvoljno ime (u ovom primjeru ime datoteke je "**sch_test**").



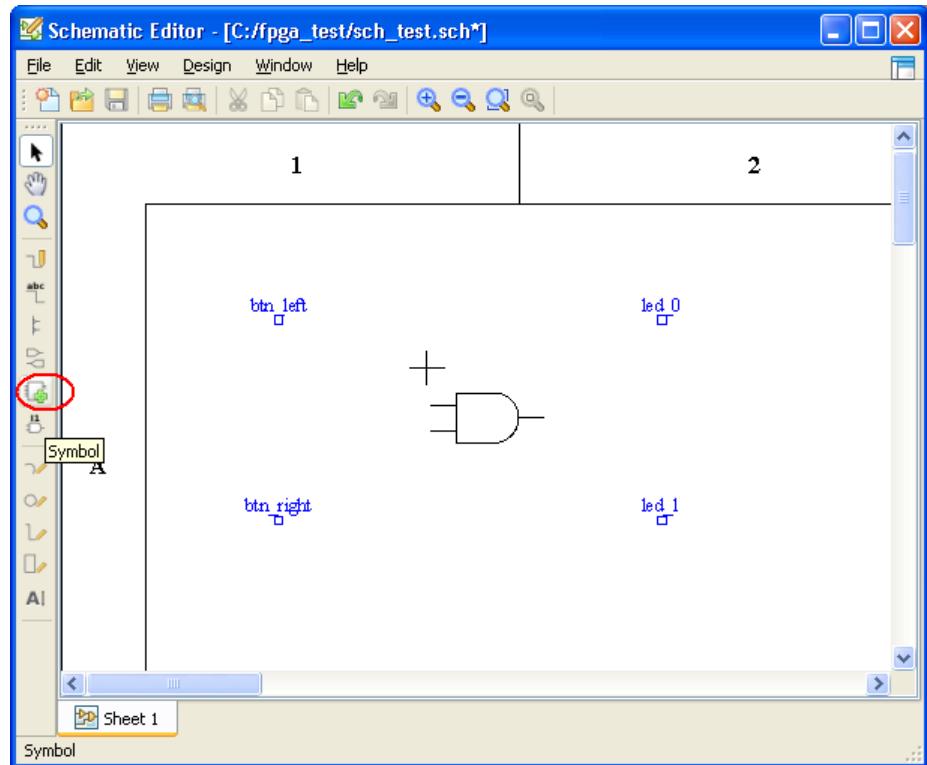
U ovom primjeru prikazan je postupak specificiranja sklopa s dva ulaza (tipke "**btn_left**" i "**btn_right**") i dva izlaza (LED indikatori "**led_0**" i "**led_1**"), na koje se povezuju izlazi iz sklopova "**I**" (*and*) i "**ILI**" (*or*).

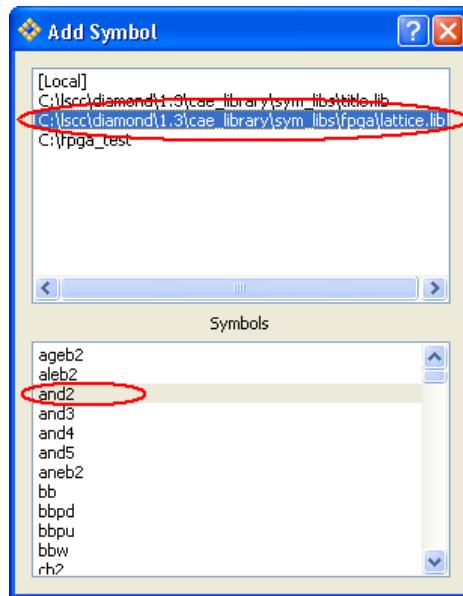
Za lakši rad sa shematskim uređivačem (editorom) moguće je prozor uređivača izdvojiti iz integrirane radne površine programa Lattice Diamond desnim klikom miša na *tab sch_test.sch*, te odabirom opcije "*Detach tool*". Na taj način i na računalima s manjim zaslonom može se dobiti dovoljno velika radna površina za raspoređivanje elemenata u shematskom opisu sklopa. Prilikom uređivanja shematskog dijagrama korisno je poslužiti se opcijama "*Zoom in*" odnosno "*Zoom out*", kako bi komponente sklopa s kojim radimo bile prikazane uvećane na radnoj površini. Upravljanje povećanjem odnosno smanjenjem prikaza može se postići i pomoću *scroll* tipke miša uz istodobno pritisnutu tipku "CTRL".

U prvom koraku potreno je stvoriti priključne točke za ulazne i izlazne signale, i dodjeliti im odgovarajuća imena. Dodjeljivanje imena signalima moguće je izvršiti odabirom alata "*Net name*" na izborničkoj traci s lijeve strane shematskog uređivača.

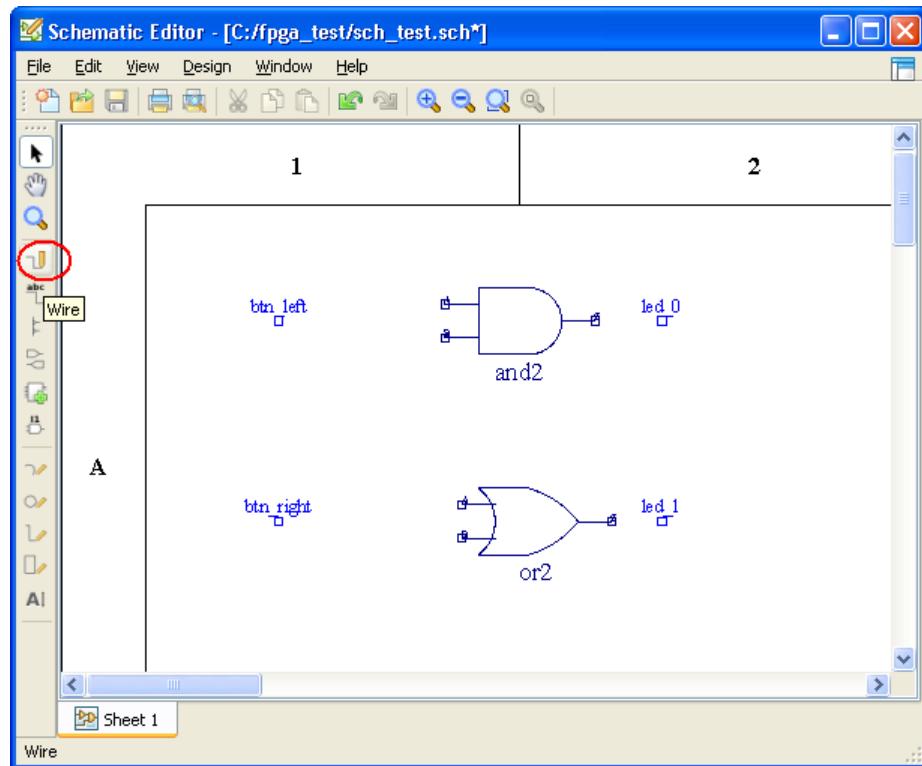


Nakon što su stvorene i ispravno imenovane oznake za sva četiri željena ulazno / izlazna signala, odabirom alata "Symbol" potrebno je otvoriti izbornik za postavljanje komponenti iz neke od dostupnih biblioteka u shematski dijagram. Treba odabratи biblioteku "lattice.lib", iz koje se može u shematski dijagram uključiti po jednu instancu sklopova "and2" (dvoulazni sklop "I") i "or2" (dvoulazni sklop "ILI").

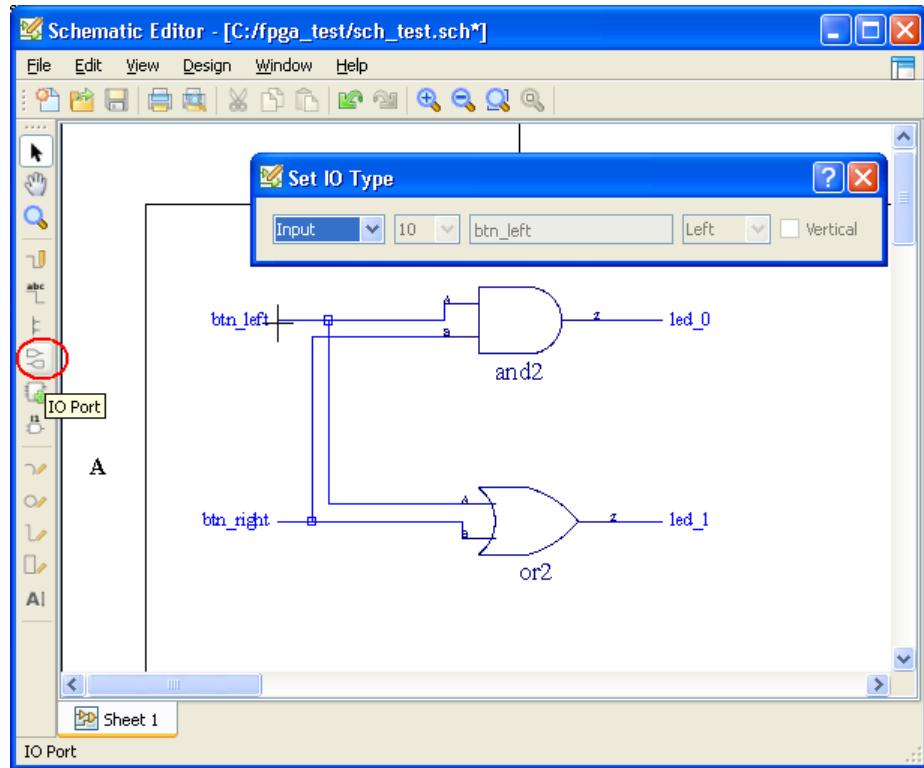




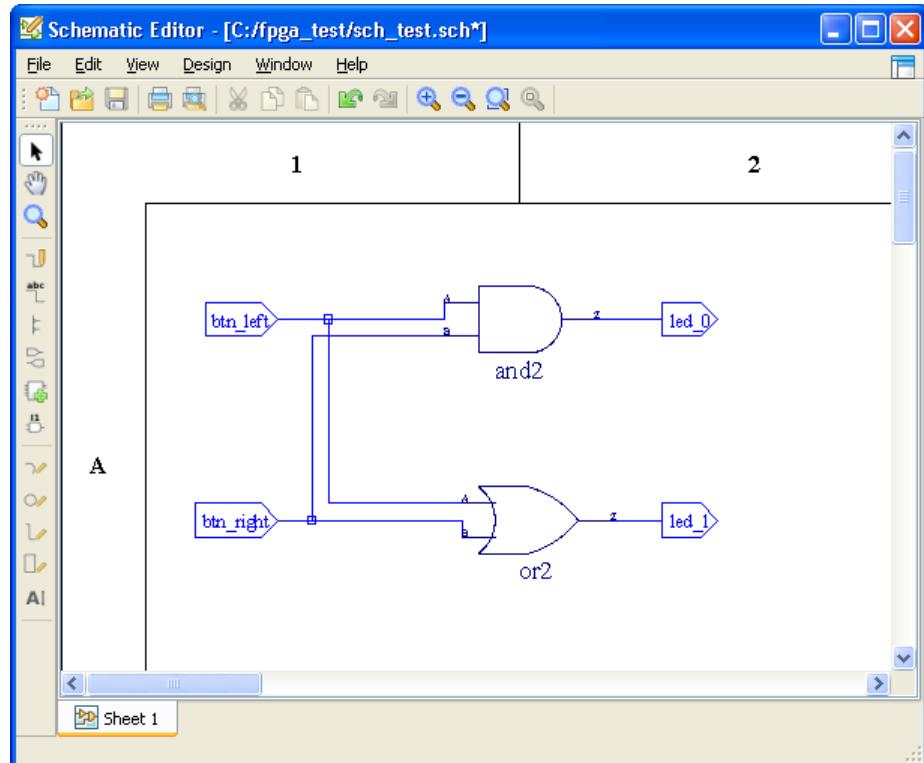
Nakon što su instance sklopova "and2" i "or2" postavljene na radnu plohu, pomoći prozor "Add symbol" može se zatvoriti pritiskom na tipku "Esc". Instance sklopova "and2" i "or2" može se povezati s ulaznim odnosno izlaznim signalima odabirom alata "Wire" na izborničkoj traci s lijeve strane prozora.



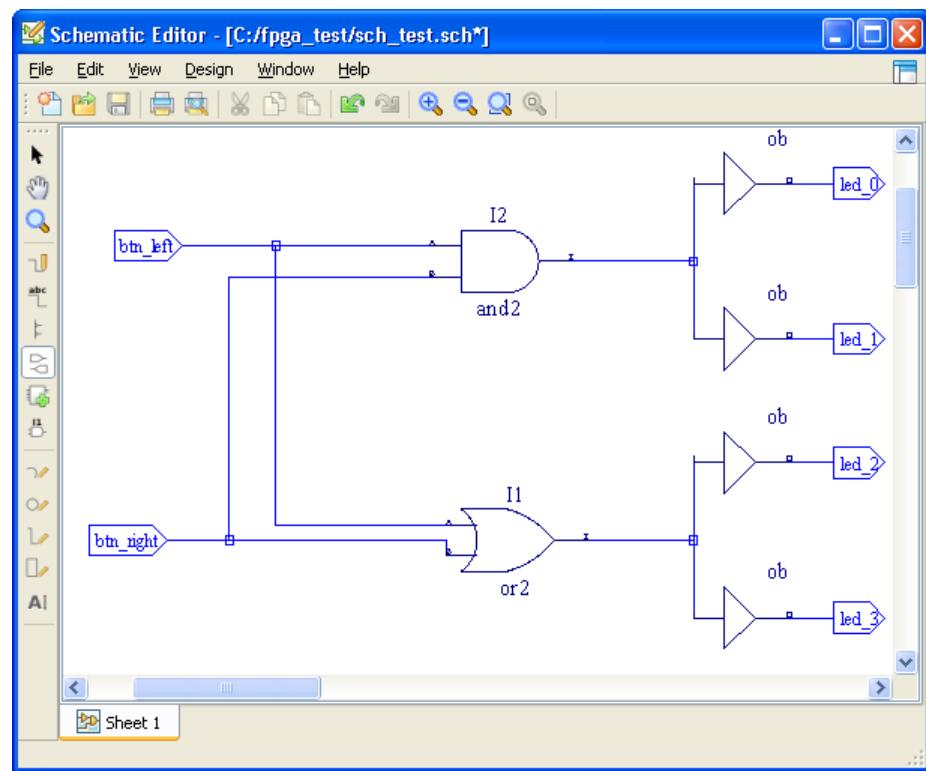
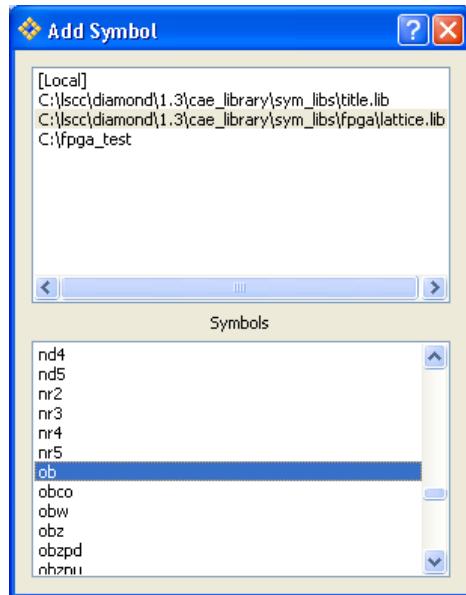
Nakon što su svi skloovi i signali međusobno povezani na željeni način, potrebno je signale deklarirati kao ulazne, odnosno izlazne, korištenjem alata "IO Port" iz izborničke trake s lijeve strane prozora. U pomoćnom prozoru "Set IO Type" potrebno je odabrati ulazni (input) odnosno izlazni (output) tip signala, te na radnoj površini lijevim klikom miša pokazivačem odabrati (*select*) oznake imena željenih signala.



Ogledni sklop sad je potpuno specificiran i spreman za sintezu.

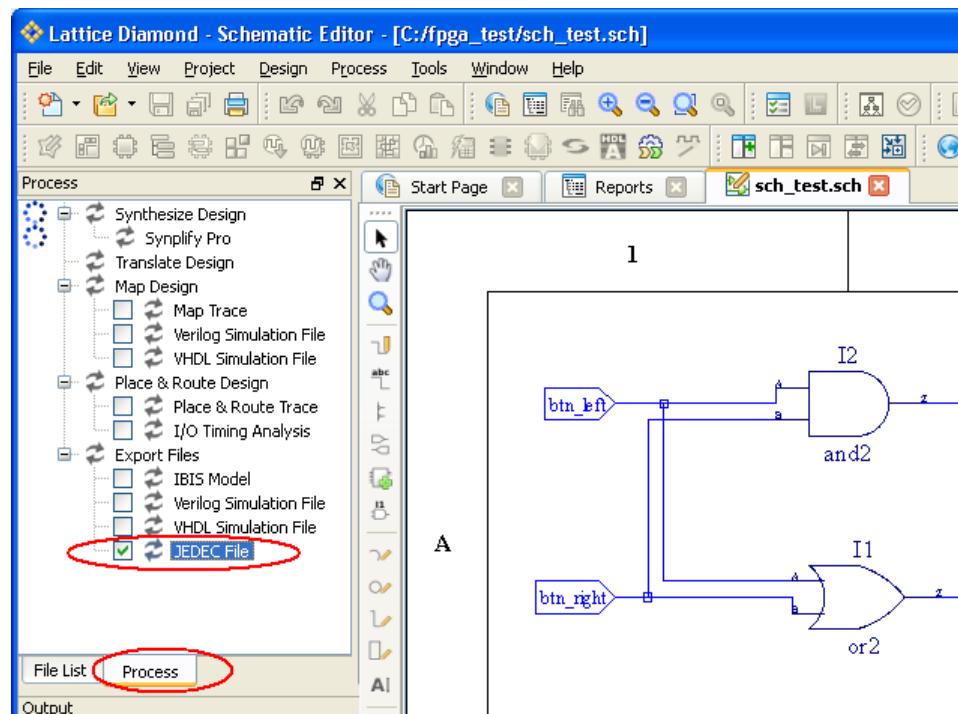


U slučaju da je potrebno određeni signal dovesti na više od jednog izlaznog sučelja, svakom izlazu treba dodijeliti zasebno izlazno pojačalo (eng. *output buffer*). U slijedećem primjeru svaki od izlaznih signala povezan je na zasebno izlazno pojačalo, koje se u biblioteci shematskih simbola krije pod skraćenicom "ob".



5.3 Sinteza konfiguracije FPGA sklopa

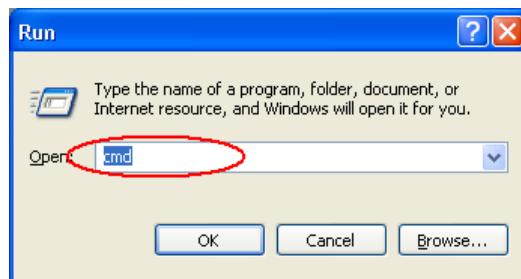
Za pokretanje postupka sinteze konfiguracijskog *bitstreama* za odabrani FPGA sklop dovoljno je odabrati opciju "Export files" -> "JEDEC File" na kartici (tabu) "Process" na lijevoj strani prozora, te dvaput kliknuti mišem na opciju "Export files". Zavisno od brzine i opterećenja računala na kojem je pokrenuto programsko okruženje Lattice Diamond, proces sinteze konfiguracijskog *bitstreama* može potrajati od nekoliko desetaka sekundi do nekoliko minuta.



Uspješno proveden postupak sinteze alat Lattice Diamond označiti će "kvačicom" zelene boje uz svaki od provedenih koraka (*Synthesize Design*, *Map Design*, *Place & Route Design*, *Export Files*). Korak "*Map Design*" u ovom primjeru biti će označen znakom upozorenja (žuti trokut), zbog toga jer veći broj ulazno / izlaznih signala koji su definirani u datoteci "ulx2s.lpf" nije upotrebljen u oglednom sklopu. Ovo upozorenje može se u cijelosti zanemariti. Alternativno, datoteku "ulx2s.lpf" može se urediti tako da sadrži samo definicije za ulazno / izlazne signale korištene u ovom primjeru (btn_left, btn_right, led_0, led_1), u kojem slučaju će i korak "*Map Design*" biti proveden bez poruka upozorenja.

5.4 Programiranje FPGA sklopa

Za programiranje konfiguracijskog *bitstreama* na FPGA sklop razvojne pločice koristi se alat "ujprog", koji je nezavisan od programskog okruženja Lattice Diamond, a poziva se iz komandne ljeske (*command shell*). Na operacijskom sustavu Windows XP komandnu ljesku može se otvoriti kroz glavni izbornik *Start – Run*, te upisati ime programa "cmd".



Konfiguracijsku datoteku za FPGA sklop treba potražiti u radnom direktoriju koji je korišten prilikom sinteze pomoću alata Lattice Diamond. U primjeru iz prethodnog poglavlja radni direktorij projekta je "c:\fpga_test\ulx2s_sch". Konfiguracijske datoteke FPGA sklopova koje stvori alat Lattice Diamond imaju ekstenziju ".jed". U ovom primjeru alat je stvorio konfiguracijsku datoteku pod imenom "ulx2s_sch.ulx2s_sch.jed".

FPGA sklop može se direktno konfigurirati novom konfiguracijskom datotekom, čime konfiguracija FPGA sklopa postaje odmah aktivna, ali se gubi prekidom napajanja. Alternativno, konfiguracijska datoteka može se pohraniti u internu Flash memoriju FPGA sklopa, u kojem slučaju će ta konfiguracija automatski postati aktivna kod slijedećeg *reset-a* FPGA sklopa.

Postupak direktnog konfiguiranja FPGA sklopa programom "ujprog" prikazan je slijedećom slikom. Za ispitivanje rada vlastitih sklopova preporuča se koristiti ovu metodu programiranja FPGA sklopa. Tijekom programiranja LED indikator USB priključka na razvojnoj pločici treptati će u kratkim vremenskim intervalima.

A screenshot of a Microsoft Windows XP command prompt window. The title bar says 'C:\WINDOWS\system32\cmd.exe'. The window content shows:

```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\marko>cd c:\fpga_test\ulx2s_sch
C:\fpga_test\ulx2s_sch>ujprog ulx2s_sch.ulx2s_sch.jed
ULX2S JTAG programmer v 0.04 09/2011
Using USB cable: FER ULX2S board JTAG / UART
Programming: 100%
Completed in 3.19 seconds.

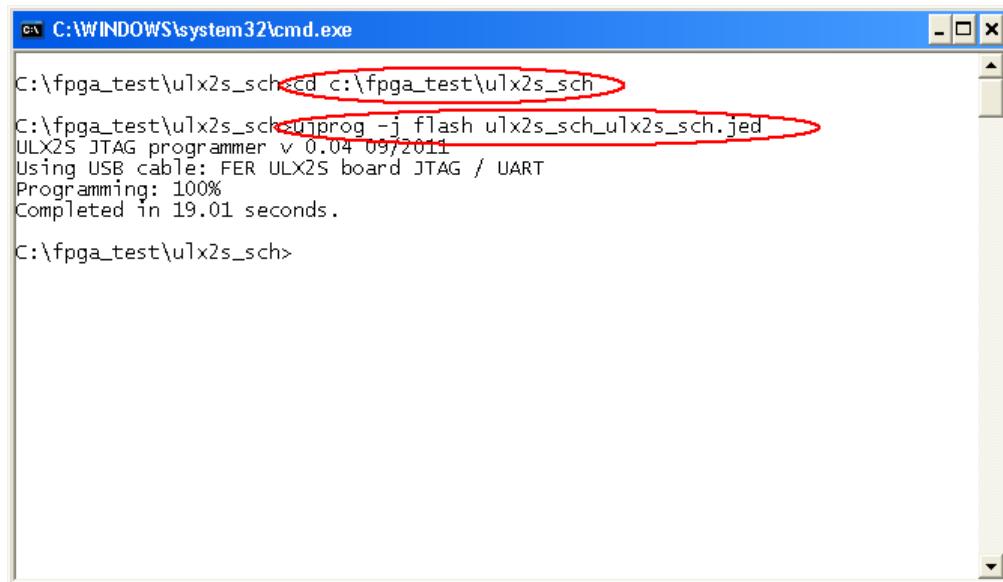
C:\fpga_test\ulx2s_sch>
```

The command 'ujprog ulx2s_sch.ulx2s_sch.jed' and its execution output are circled in red.

Prije početka programiranja FPGA sklopa potrebno je zaustaviti sve programe koji komuniciraju s pločicom putem USB – RS-232 sučelja, npr. *Putty* ili *Hyper Terminal*.

Nakon što je FPGA sklop uspješno konfiguriran možemo ispitati njegov rad pritiskom na tipke "btn_left" i "btn_right", što bi trebalo rezultirati paljenjem LED indikatora "led_0" i "led_1", u skladu sa shematskom specifikacijom sklopa.

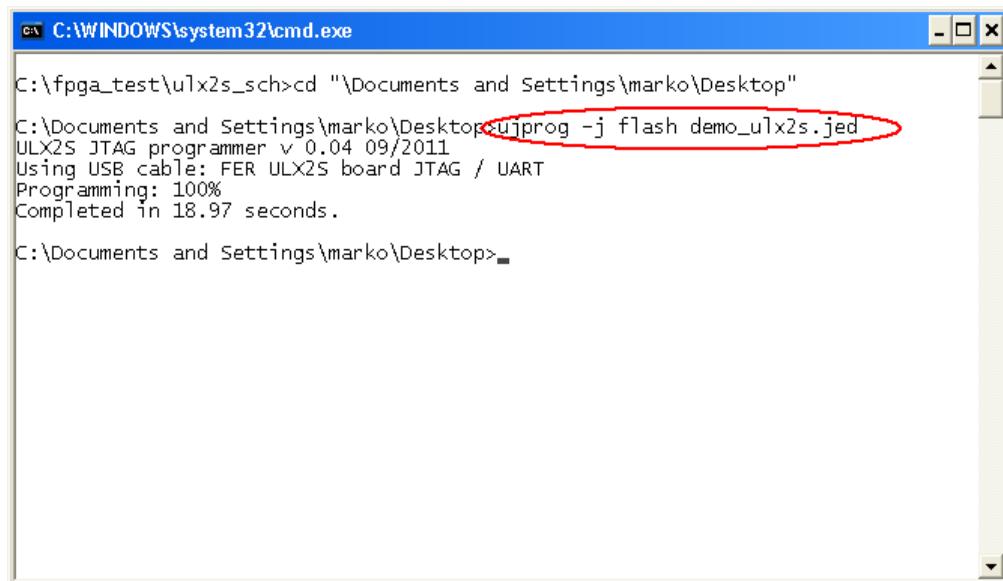
Za programiranje interne konfiguracijske Flash memorije FPGA sklopa program "**ujprog**" je potrebno pozvati s uključenom opcijom "-j flash", kao što je prikazano na slici.



```
C:\fpga_test\ulx2s_sch>cd c:\fpga_test\ulx2s_sch
C:\fpga_test\ulx2s_sch>ujprog -j flash ulx2s_sch.ulx2s_sch.jed
ULX2S JTAG programmer v 0.04 09/2011
Using USB cable: FER ULX2S board JTAG / UART
Programming: 100%
Completed in 19.01 seconds.

C:\fpga_test\ulx2s_sch>
```

Ukoliko želimo konfigurirati pločicu da se po priključenju napajanja automatski pokrene dijagnostičko-demonstracijski konfiguracijski *bitstream*, istu možemo dohvatiti s web sjedišta http://www.nxlab.fer.hr/dl/demo_ulx2s.jed, te konfiguracijski *bitstream* programirati u internu Flash memoriju FPGA sklopa, kao što je prikazano slijedećom slikom.



```
C:\fpga_test\ulx2s_sch>cd "\Documents and Settings\marko\Desktop"
C:\Documents and Settings\marko\Desktop>ujprog -j flash demo_ulx2s.jed
ULX2S JTAG programmer v 0.04 09/2011
Using USB cable: FER ULX2S board JTAG / UART
Programming: 100%
Completed in 18.97 seconds.

C:\Documents and Settings\marko\Desktop>
```

6 Definicije ulazno / izlaznih signala

```
#  
# FER ULX2S board I/O constraints  
#  
# Board revision A, Sep 2011  
# Constraints file revision 0.92, 2011/09/08  
#  
  
COMMERCIAL;  
BLOCK RESETPATHS;  
BLOCK ASYNCPATHS;  
BLOCK JTAGPATHS;  
  
# Main 25 MHz on-board clock generator  
LOCATE COMP "clk_25m" SITE "30"; # IN  
FREQUENCY PORT "clk_25m" 25 MHz;  
  
# USB UART receive and transmit data  
LOCATE COMP "rs232_rx" SITE "10"; # IN  
LOCATE COMP "rs232_tx" SITE "137"; # OUT  
  
# LED indicators  
LOCATE COMP "led_0" SITE "29"; # OUT (shared with dil_41)  
LOCATE COMP "led_1" SITE "37"; # OUT (shared with dil_42)  
LOCATE COMP "led_2" SITE "36"; # OUT (shared with dil_43)  
LOCATE COMP "led_3" SITE "35"; # OUT (shared with dil_44)  
LOCATE COMP "led_4" SITE "144"; # OUT (shared with dil_12)  
LOCATE COMP "led_5" SITE "5"; # OUT (shared with dil_7)  
LOCATE COMP "led_6" SITE "6"; # OUT (shared with dil_6)  
LOCATE COMP "led_7" SITE "138"; # OUT (shared with dil_5)  
  
# DIP switches  
LOCATE COMP "sw_0" SITE "45"; # IN  
LOCATE COMP "sw_1" SITE "46"; # IN  
LOCATE COMP "sw_2" SITE "129"; # IN  
LOCATE COMP "sw_3" SITE "130"; # IN  
  
# Pushbuttons  
LOCATE COMP "btn_up" SITE "11"; # IN  
LOCATE COMP "btn_down" SITE "1"; # IN (shared with dil_10)  
LOCATE COMP "btn_left" SITE "2"; # IN (shared with dil_11)  
LOCATE COMP "btn_right" SITE "38"; # IN (shared with dil_38)  
LOCATE COMP "btn_center" SITE "22"; # IN (shared with dil_39)  
  
# Phone jack - resistor network only on tip!  
LOCATE COMP "p_ring" SITE "44"; # INOUT (75 Ohm)  
LOCATE COMP "p_tip_0" SITE "32"; # INOUT (1330 Ohm) (shared with dil_45)  
LOCATE COMP "p_tip_1" SITE "27"; # INOUT (665 Ohm) (shared with dil_46)  
LOCATE COMP "p_tip_2" SITE "28"; # INOUT (330 Ohm) (shared with dil_47)  
LOCATE COMP "p_tip_3" SITE "43"; # INOUT (160 Ohm)  
  
# Serial flash (SPI)  
LOCATE COMP "flash_cen" SITE "13"; # OUT  
LOCATE COMP "flash_sck" SITE "16"; # OUT  
LOCATE COMP "flash_si" SITE "19"; # OUT  
LOCATE COMP "flash_so" SITE "18"; # IN  
  
# MicroSD card (SPI)  
LOCATE COMP "sdcard_cen" SITE "15"; # OUT  
LOCATE COMP "sdcard_sck" SITE "20"; # OUT  
LOCATE COMP "sdcard_si" SITE "17"; # OUT  
LOCATE COMP "sdcard_so" SITE "21"; # IN  
  
# 2 x 24 pin DIL header  
# "dil_1" -> VCCIO (3.3 V)  
LOCATE COMP "dil_2" SITE "131"; # INOUT  
LOCATE COMP "dil_3" SITE "134"; # INOUT  
LOCATE COMP "dil_4" SITE "9"; # INOUT  
LOCATE COMP "dil_5" SITE "138"; # INOUT (shared with led_7)  
LOCATE COMP "dil_6" SITE "6"; # INOUT (shared with led_6)  
LOCATE COMP "dil_7" SITE "5"; # INOUT (shared with led_5)  
LOCATE COMP "dil_8" SITE "8"; # INOUT
```

```

LOCATE COMP "dil_9" SITE "7"; # INOUT
LOCATE COMP "dil_10" SITE "1"; # INOUT (shared with btn_down)
LOCATE COMP "dil_11" SITE "2"; # INOUT (shared with btn_left)
LOCATE COMP "dil_12" SITE "144"; # INOUT (shared with led_4)
LOCATE COMP "dil_13" SITE "143"; # INOUT
LOCATE COMP "dil_14" SITE "142"; # INOUT
LOCATE COMP "dil_15" SITE "141"; # INOUT
LOCATE COMP "dil_16" SITE "133"; # INOUT
LOCATE COMP "dil_17" SITE "132"; # INOUT
LOCATE COMP "dil_18" SITE "123"; # INOUT
LOCATE COMP "dil_19" SITE "124"; # INOUT
LOCATE COMP "dil_20" SITE "115"; # INOUT
LOCATE COMP "dil_21" SITE "114"; # INOUT
LOCATE COMP "dil_22" SITE "110"; # INOUT
LOCATE COMP "dil_23" SITE "109"; # INOUT
# "dil_24" -> VCCIO (3.3 V)
# "dil_25" -> GND
LOCATE COMP "dil_26" SITE "72"; # INOUT
LOCATE COMP "dil_27" SITE "71"; # INOUT
LOCATE COMP "dil_28" SITE "70"; # INOUT
LOCATE COMP "dil_29" SITE "69"; # INOUT
LOCATE COMP "dil_30" SITE "66"; # INOUT
LOCATE COMP "dil_31" SITE "65"; # INOUT
LOCATE COMP "dil_32" SITE "52"; # INOUT
LOCATE COMP "dil_33" SITE "50"; # INOUT
LOCATE COMP "dil_34" SITE "48"; # INOUT
LOCATE COMP "dil_35" SITE "47"; # INOUT
LOCATE COMP "dil_36" SITE "40"; # INOUT
LOCATE COMP "dil_37" SITE "39"; # INOUT
LOCATE COMP "dil_38" SITE "38"; # INOUT (shared with btn_right)
LOCATE COMP "dil_39" SITE "22"; # INOUT (shared with btn_center)
LOCATE COMP "dil_40" SITE "31"; # INOUT
LOCATE COMP "dil_41" SITE "29"; # INOUT (shared with led_0)
LOCATE COMP "dil_42" SITE "37"; # INOUT (shared with led_1)
LOCATE COMP "dil_43" SITE "36"; # INOUT (shared with led_2)
LOCATE COMP "dil_44" SITE "35"; # INOUT (shared with led_3)
LOCATE COMP "dil_45" SITE "32"; # INOUT (shared with p_tip_0)
LOCATE COMP "dil_46" SITE "27"; # INOUT (shared with p_tip_1)
LOCATE COMP "dil_47" SITE "28"; # INOUT (shared with p_tip_2)
# "dil_48" -> GND

# 512K x 16 bit SRAM
LOCATE COMP "sram_wel" SITE "93"; # OUT
LOCATE COMP "sram_lbl" SITE "120"; # OUT
LOCATE COMP "sram_ubl" SITE "121"; # OUT
LOCATE COMP "sram_a_0" SITE "101"; # OUT
LOCATE COMP "sram_a_1" SITE "103"; # OUT
LOCATE COMP "sram_a_2" SITE "104"; # OUT
LOCATE COMP "sram_a_3" SITE "107"; # OUT
LOCATE COMP "sram_a_4" SITE "108"; # OUT
LOCATE COMP "sram_a_5" SITE "125"; # OUT
LOCATE COMP "sram_a_6" SITE "127"; # OUT
LOCATE COMP "sram_a_7" SITE "122"; # OUT
LOCATE COMP "sram_a_8" SITE "58"; # OUT
LOCATE COMP "sram_a_9" SITE "53"; # OUT
LOCATE COMP "sram_a_10" SITE "54"; # OUT
LOCATE COMP "sram_a_11" SITE "55"; # OUT
LOCATE COMP "sram_a_12" SITE "56"; # OUT
LOCATE COMP "sram_a_13" SITE "57"; # OUT
LOCATE COMP "sram_a_14" SITE "73"; # OUT
LOCATE COMP "sram_a_15" SITE "74"; # OUT
LOCATE COMP "sram_a_16" SITE "77"; # OUT
LOCATE COMP "sram_a_17" SITE "78"; # OUT
LOCATE COMP "sram_a_18" SITE "94"; # OUT
LOCATE COMP "sram_d_0" SITE "98"; # INOUT
LOCATE COMP "sram_d_1" SITE "96"; # INOUT
LOCATE COMP "sram_d_2" SITE "92"; # INOUT
LOCATE COMP "sram_d_3" SITE "91"; # INOUT
LOCATE COMP "sram_d_4" SITE "89"; # INOUT
LOCATE COMP "sram_d_5" SITE "88"; # INOUT
LOCATE COMP "sram_d_6" SITE "87"; # INOUT
LOCATE COMP "sram_d_7" SITE "90"; # INOUT
LOCATE COMP "sram_d_8" SITE "61"; # INOUT
LOCATE COMP "sram_d_9" SITE "62"; # INOUT
LOCATE COMP "sram_d_10" SITE "99"; # INOUT
LOCATE COMP "sram_d_11" SITE "100"; # INOUT
LOCATE COMP "sram_d_12" SITE "102"; # INOUT

```

```

LOCATE COMP "sram_d_13" SITE "113";          # INOUT
LOCATE COMP "sram_d_14" SITE "116";          # INOUT
LOCATE COMP "sram_d_15" SITE "119";          # INOUT

#
# Special I/O properties should be declared here
#

# All I/O ports are tied to 3.3 V
IOBUF ALLPORTS IO_TYPE=LVCMOS33 PULLMODE=NONE;

# Phone jack may drain more current than usual
IOBUF PORT "p_ring" DRIVE=20;
IOBUF PORT "p_tip_0" DRIVE=20;
IOBUF PORT "p_tip_1" DRIVE=20;
IOBUF PORT "p_tip_2" DRIVE=20;
IOBUF PORT "p_tip_3" DRIVE=20;

# DIP switches and pushbuttons need a pulldown resistor
IOBUF PORT "sw_0" PULLMODE=DOWN;
IOBUF PORT "sw_1" PULLMODE=DOWN;
IOBUF PORT "sw_2" PULLMODE=DOWN;
IOBUF PORT "sw_3" PULLMODE=DOWN;
IOBUF PORT "btn_up" PULLMODE=DOWN;
IOBUF PORT "btn_down" PULLMODE=DOWN;
IOBUF PORT "btn_left" PULLMODE=DOWN;
IOBUF PORT "btn_right" PULLMODE=DOWN;
IOBUF PORT "btn_center" PULLMODE=DOWN;

```