

Sveučilište u Zagrebu
Fakultet elektrotehnike i računarstva

Upute za rad s FPGA razvojnou pločicom ULX2S

Marko Zec

listopad 2014.

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.....	6
2.1 Napajanje.....	6
2.2 FPGA sklop.....	7
2.3 Sučelje JTAG.....	8
2.4 Tipke.....	8
2.5 Prekidači.....	9
2.6 LED indikatori.....	9
2.7 Višenamjenska stereo priključnica.....	9
2.8 Sučelje RS-232.....	10
2.9 Generator takta (kvarcni oscilator).....	10
2.10 SRAM.....	11
2.11 Flash memorija.....	11
2.12 MicroSD kartica.....	12
2.13 DIL priključnice za proširenje.....	12
3 Instaliranje programske podrške.....	13
3.1 Instaliranje programskog paketa Lattice Diamond.....	13
3.2 Instaliranje programskog alata za programiranje pločice.....	15
3.3 Instaliranje USB pogonskih programa (drivera) za razvojnu pločicu.....	16
3.4 Instaliranje programskih alata na operacijskom sustavu OS-X.....	16
3.5 Instaliranje programskih alata na operacijskom sustavu Linux.....	17
4 Ispitivanje rada pločice.....	18
4.1 Ispitivanje USB komunikacije.....	18
4.2 Ispitivanje video izlaza.....	19
5 Rad s razvojnim alatom Lattice Diamond.....	20
5.1 Kreiranje novog projekta.....	20
5.2 Shematski opis digitalnog sklopa.....	23
5.3 Sinteza konfiguracije FPGA sklopa.....	27
5.4 Programiranje FPGA sklopa.....	28
5.5 Uobičajeni problemi u radu s FPGA razvojnim alatima.....	29
6 Razvojna pločica kao samostalni mikroprocesorski sustav.....	31
6.1 Instaliranje mikroprocesorske konfiguracije MIPS na FPGA sklop.....	31
6.2 Priprema MIPS programskih alata.....	31
6.3 Programiranje u jeziku C.....	32
7 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. Sintezom odgovarajuće procesorske jezgre na FPGA sklopu razvojna pločica može raditi i kao samostalni programski upravljiv mikroprocesorski sustav.

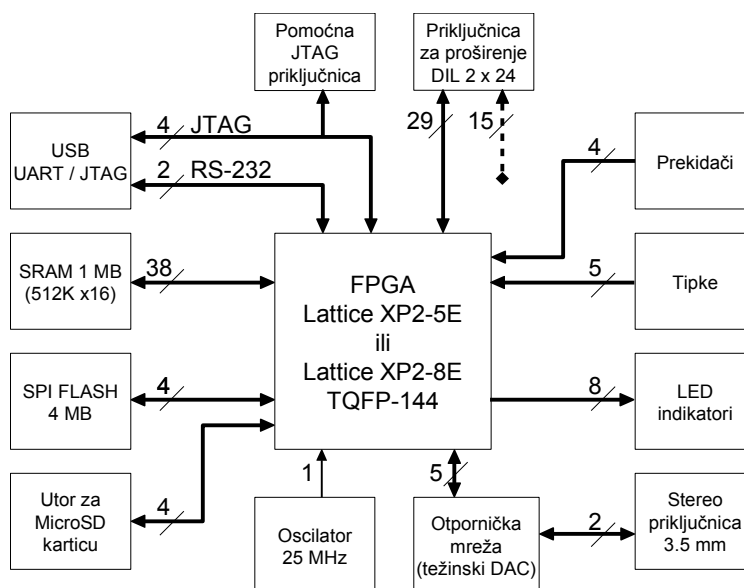
Glavna komponenta na pločici je FPGA sklop XP2 proizvođača Lattice Semiconductor s oko 5000 (varijanta XP2-5E) ili 8000 (varijanta XP2-8E) programirljivih logičkih elemenata temeljenih na preglednim tablicama s 4 ulaza (eng. *4-input lookup table – LUT*). FPGA sklop se može neograničeni broj puta rekonfigurirati 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 i 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 otpornom mrežom za D/A pretvorbu
- Utor za memorijske 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 u standardnom rasteru 2.54 mm, 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.

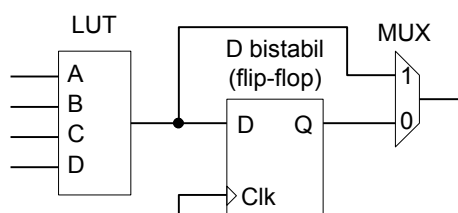


Slika 1: blok-shema povezivanja komponenti razvojne pločice

1.1 Programirljiva polja logičkih blokova (FPGA)

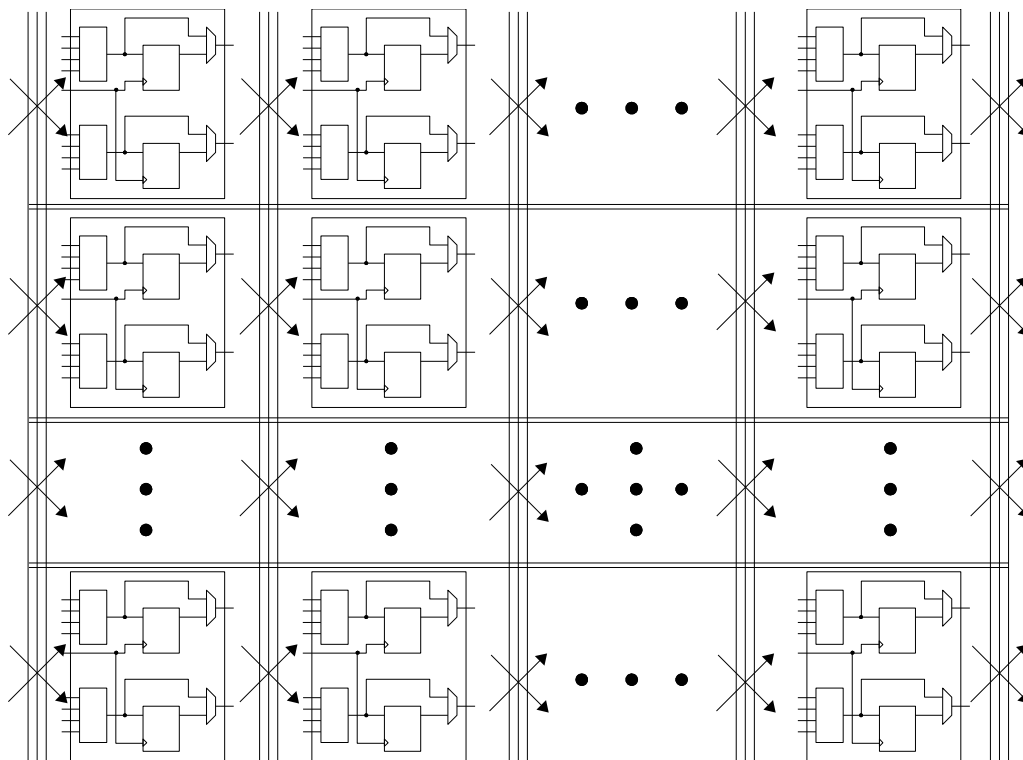
Programirljiva polja logičkih blokova (*eng. Field programmable gate array – FPGA*) su digitalni sklopovi vrlo visokog stupnja integracije u kojima se veliki broj relativno jednostavnih i uniformnih programirljivih logičkih elemenata može povezati u proizvoljni kompleksni sklop, te takav sklop povezati 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 prospojnici), 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

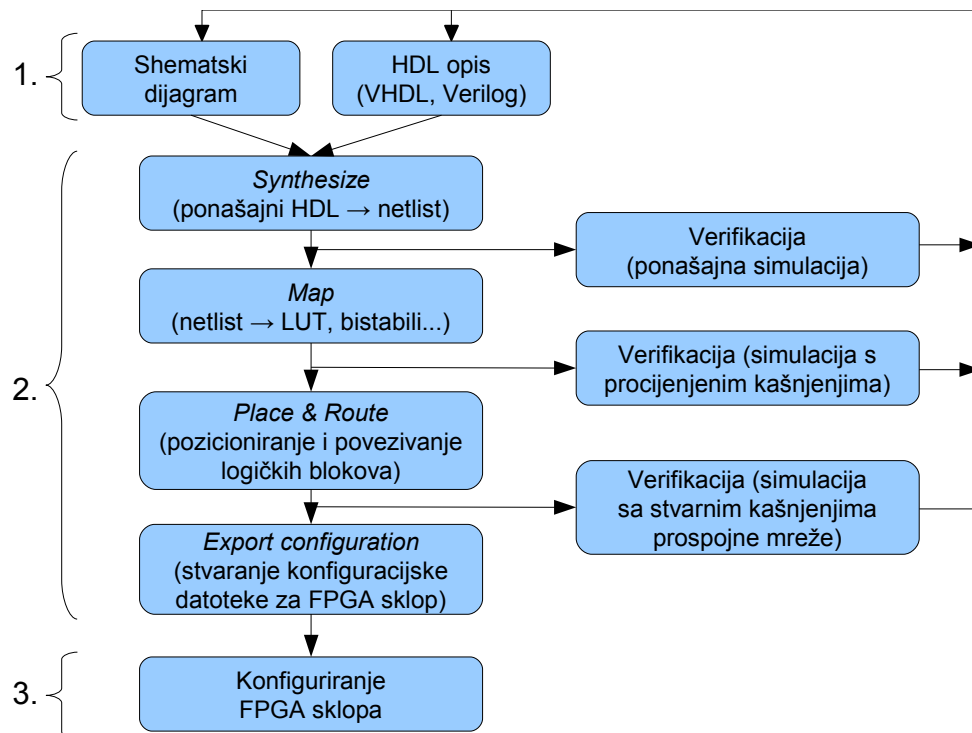
Veličine polja današnjih FPGA sklopova kreću su u rasponu od oko 1.000 do 2.000.000 preglednih tablica (LUT) odnosno logičkih elemenata. Kašnjenja pojedine pregledne tablice (LUT) tipično su reda veličine od 100 do 300 ps, kašnjenja bistabila su reda veličine od 200 do 500 ps, dok kašnjenja koja unosi prospojna mreža mogu doseći i do nekoliko ns. Bitno je uočiti da svaki logički element 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 statičke 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 programskih 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 konkurencije. 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 željenog 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 željenog 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 funkcijski ekvivalentnim strukturnim modelima izgrađenim od primitiva specifičnih za ciljanu FPGA platformu, pri čemu se provodi i automatizirana optimizacija kombinacijske i sekvencijske logike. Rezultirajuća mreža primitiva, koja se uobičajeno naziva *netlist*, može se analizirati korištenjem odgovarajućeg simulatora, pri čemu se ne modeliraju kašnjenja primjenjenih primitiva.

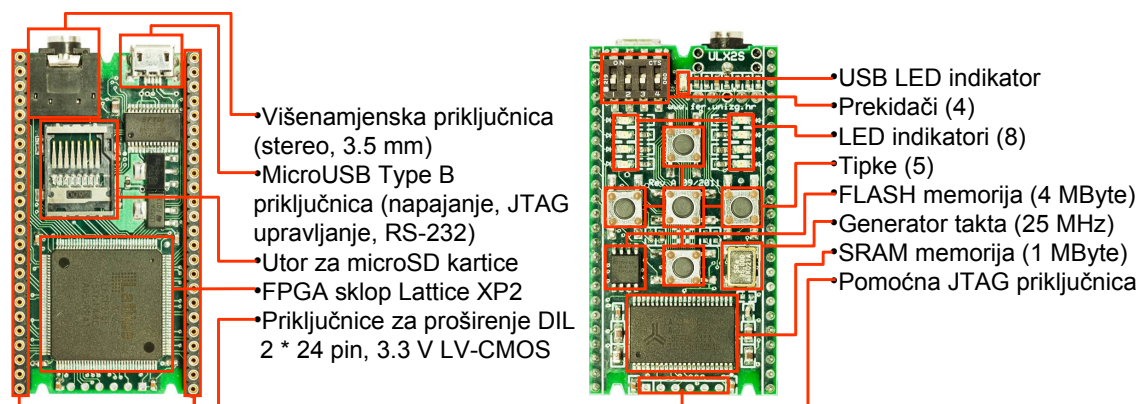
U sljedećem koraku, koji se uobičajeno naziva "*Map*", sintetizator primitive (npr. AND_2, OR_8, MUX_16_1) zamjenjuje mrežom preglednih tablica (LUT), memorijskih elemenata te ostalih specijaliziranih logičkih blokova koji odgovaraju ciljanoj FPGA platformi. U ovom koraku sintetizator može napraviti 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 raspoređuje logičke blokove na FPGA sklopu i povezuje ih na način koji će omogućiti ispravan rad sklopa na najvišoj mogućoj frekvenciji takta. Zavisno od kompleksnosti opisanog sklopa, ciljane frekvencije takta, postavki algoritma *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 modela mogu se pronaći kritični putevi projektiranog sklopa koje ograničuju 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 komponenti na pločici ULX2S.



Slika 5: donja strana pločice

Slika 6: gornja strana pločice

Dimenzije pločice su 64 * 36 * 9 mm.

Pločica se s računalom povezuje putem priključnog kabela tipa USB A na micro-USB tip B, 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 priključku može rezultirati oštećenjem komponenti.

Na pločicu su ugrađeni linearni regulatori napajanja s izlazima 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 jezgra FPGA sklopa radi na 1.2 V. Ukupna potrošnja struje zavisi od kompleksnosti konfiguracije i frekvenciji takta unutar FPGA sklopa, te o statičkom i dinamičkom opterećenju izlaznih signala, a uobičajeno se može 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 struji kroz regulator, kod kompleksnijih konfiguracija FPGA sklopa regulatori napajanja pa i FPGA sklop mogu se osjetno zagrijati, što međutim ne utječe na stabilnost njihovog rada i na vijek trajanja. Na primjer, uz ispitnu konfiguraciju FPGA sklopa razvojna pločica tipično troši između 170 i 210 mA, što pri naponu napajanja od 5V odgovara ukupnoj disipaciji snage od otprilike 1 W.

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 bez oštećenja USB sučelja ili priključenog uređaja. Međutim, dodatni oprez potreban je kod priključenja pločice na računalo putem USB *huba* s vlastitim napajanjem ili na nezavisne izvore napajanja s USB priključkom koji se uobičajeno koriste za punjenje baterija mobilnih telefona i sličnih uređaja potrošačke elektronike, budući da takvi uređaji često nemaju ugrađene sklopove za zaštitu od kratkog spoja.

2.2 FPGA sklop

Na pločicu ULX2S ugrađuje se FPGA sklop Lattice XP2 u varijanti 5E ili 8E. Razlike između ove dvije varijante FPGA sklopa prikazane su tablicom:

	XP2-5E	XP2-8E
Preglednih tablica (LUT)	oko 5000	oko 8000
Raspodijeljeni RAM (Kbits)	10	18
Zasebnih RAM blokova (2048x9 bits)	9	12
Blokova za množenje (18x18)	12	16
Blokova za sintezu signala takta (PLL)	2	2

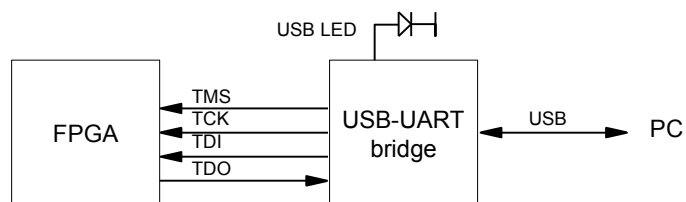
Glavna značajka po kojoj se sklop Lattice XP2 razlikuje od većine FPGA sklopova drugih proizvođača je ugrađena Flash memorija za konfiguraciju. Po uspostavi napajanja FPGA sklop će automatski konfiguracijskim zapisom iz ugrađene Flash memorije programirati vlastiti konfiguracijski RAM, što omogućuje spremnost sklopa za rad samo nekoliko desetaka mikrosekundi nakon uključanja napajanja. Ugrađenu konfiguracijsku Flash memoriju moguće je programirati do 10.000 puta, a deklarirano najkraće vrijeme pamćenja Flash memorije je 20 godina. Za potrebe ispitivanja rada konfiguraciju FPGA sklopa moguće je programirati putem JTAG sučelja i direktno u konfiguracijski RAM bez ograničenja broja ciklusa pisanja.

Napajanje V_{CCIO} svih ulazno-izlaznih sučelja je povezano je na izlaz linearnog regulatora 3.3 V, što znači da su na pločici ULX2S svi priključci FPGA sklopa konfigurirani su za rad s naponskim razinama između 0 i 3.3 V, o čemu treba voditi računa pri eventualnom povezivanju pločice s vanjskim uređajima.

2.3 Sučelje JTAG

FPGA sklop programira se putem sučelja JTAG koje je na pločici ULX2S izvedeno pomoću USB-UART prilagodnika FT-232R. JTAG signalima TMS, TCK, TDI i TDO upravlja se u pravilu pomoću programskog alata `ujprog` kojeg se pokreće na računalu na koje je priključena razvojna pločica (slika 7). Za vrijeme programiranja FPGA sklopa USB LED indikator trepće frekvencijom približno 1 Hz.

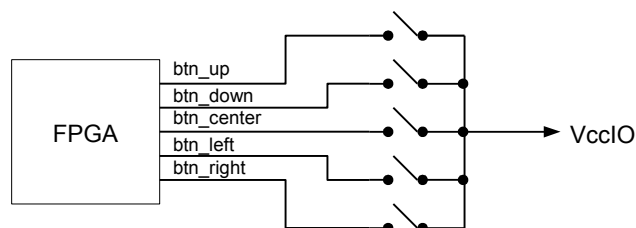
Priključci USB-UART prilagodnika povezani sa signalima TMS, TCK, TDI i TDO su postavljeni u stanje visoke impedancije osim za vrijeme izvršavanja programskog alata `ujprog`, što prema potrebi omogućuje i povezivanje s vanjskim uređajem za programiranje putem pomoćne JTAG priključnice.



Slika 7: Signali sučelja JTAG

2.4 Tipke

Pritisak na svaku od pet tipki na razvojnoj pločici (`btn_up`, `btn_down`, `btn_center`, `btn_left`, `btn_right`) povezuje odgovarajuću priključnicu FPGA sklopa direktno na visoku naponsku razinu (V_{ccIO}). Kad tipke nisu pritisnute, signali će biti pritegnuti niskoj naponskoj razini putem *pull-down* otpornika integriranih u FPGA sklop, a primjena kojih je određena zapisom *I/O preference* (poglavlje 7). Shema povezivanja tipki s FPGA sklopom prikazana je na slici 8:



Slika 8: tipke

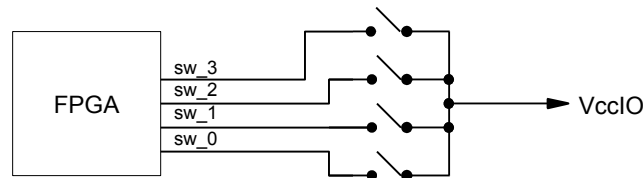
Deklaracija signala u sučelju sklopa (VHDL):

```
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 mikroprekidača povezuje odgovarajuću priključnicu FPGA sklopa direktno na visoku naponsku razinu (V_{CCIO}). U položaju "OFF" signali će biti pritegnuti na nisku naponsku razinu putem *pull-down* otpornika integriranih u FPGA sklop, primjena kojih je određena zapisom *I/O preference* (poglavlje 7).

Gledano s gornje strane pločice, s lijeve na desnu stranu, prekidači su numerirani slijedećim redom: *sw_3*, *sw_2*, *sw_1*, *sw_0*. Shema povezivanja mikroprekidača s FPGA sklopom prikazana je na slici 9:



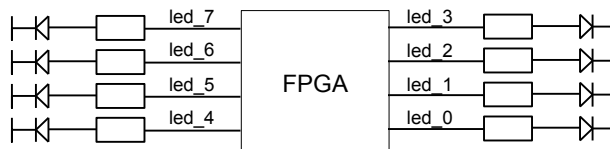
Slika 9: mikroprekidači

Deklaracija signala u sučelju sklopa (VHDL):

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

2.6 LED indikatori

LED indikatori povezani su s FPGA sklopom putem otpornika za ograničavanje struje prema shemi sa slike 10. Gledano s gornje strane pločice, LED indikatori na lijevoj strani odnosno njima pripadajući signali numerirani su slijedećim redom: *led_7*, *led_6*, *led_5* i *led_4*, a indikatori na desnoj strani pločice su *led_3*, *led_2*, *led_1* i *led_0*.



Slika 10: LED indikatori

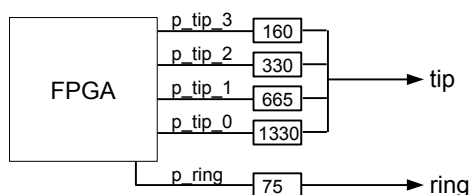
Deklaracija signala u sučelju sklopa (VHDL):

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

2.7 Višenamjenska stereo priključnica

Stereo utičnica prvenstveno je namijenjena za priključivanje niskoomskih slušalica ili ekrana s kompozitnim video ulazom, ali može poslužiti i za druge svrhe. Na kontakt na vrhu priključnice ("tip") dovodi se signal s 4-bitne težinske otporne mreže koja je proračunata tako da uz pretpostavljeni teret od 75 Ohma na izlazu može dati napon između 0 i 1.5 V u 16 diskretnih koraka. Na središnji kontakt priključnice ("ring") dovodi

se samo jedan signal putem otpornika od 75 Ohma. Težinska otporna mreža koja povezuje stereo priključnicu s FPGA sklopom prikazana je slikom 11:



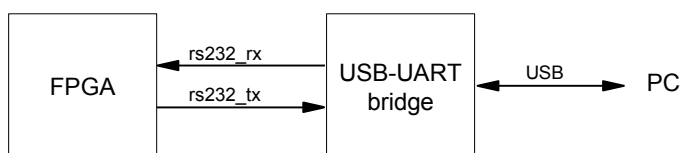
Slika 11: težinska otporna mreža na stereo priključnici

Deklaracija signala u sučelju sklopa (VHDL):

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

2.8 Sučelje RS-232

Komunikacija između sklopa sintetiziranog na FPGA *chipu* i računala najjednostavnije se može ostvariti asinkronim serijskim prijenosom podataka putem USB–UART prenosnika ugrađenog na razvojnu pločicu. USB–UART prenosnik FT-232R podržava brzine prijenosa između 300 i 3000000 bita u sekundi, a na razvojnoj pločici se koriste samo podatkovni signali `rs232_tx` i `rs232_rx`, kao što je prikazano na slici 12:



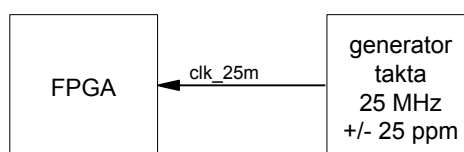
Slika 12: signali za serijsku komunikaciju s računalom

Deklaracija signala u sučelju sklopa (VHDL):

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

2.9 Generator takta (kvarcni oscilator)

Na razvojnu pločicu ugrađen je generator takta frekvencije 25 Mhz i visoke stabilnosti u velikom rasponu temperaturnih, električkih i mehaničkih uvjeta. Izlazni signal generatora takta na sučelju prema FPGA sklopu nazvan je `clk_25m` (slika 13).



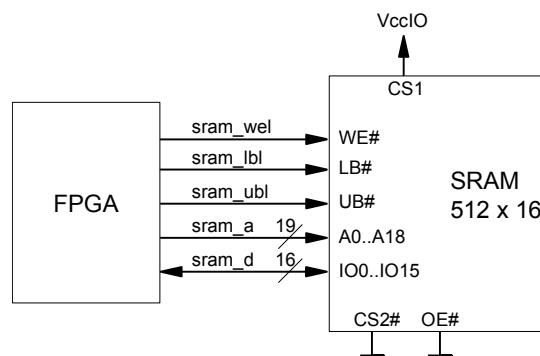
Slika 13: signal takta 25 MHz

Deklaracija signala u sučelju sklopa (VHDL):

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

2.10 SRAM

Na razvojnu pločicu ugrađuje se statička RAM memorija (SRAM) kapaciteta 1 MByte organizirana kao polje od 524288 16-bitnih riječi. Jedan ciklus čitanja ili pisanja mora trajati najmanje 55 ns. Upravljački signali CS1, CS2# i OE# na pločici ULX2S su povezani na način da su stalno aktivni, pa se radom memorije SRAM upravlja isključivo signalima WE# (*write enable low*), LB# (*lower byte enable low*) i UB# (*upper byte enable low*). Za adresiranje podataka služi 19-bitna sabirnica sram_a, a podaci se između FPGA sklopa i memorije prenose 16-bitnom sabirnicom sram_d, pri čemu smjer toka podataka određuje upravljački signal sram_wel. Shema sučelja između FPGA sklopa i memorije SRAM prikazana je slikom 14:



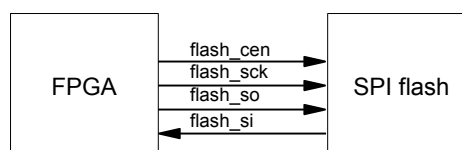
Slika 14: signali sučelja prema vanjskoj memoriji (RAM)

Deklaracija signala u sučelju sklopa (VHDL):

```
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 memorija

Komunikacija s Flash memorijom ugrađenoj na razvojnu pločicu ostvaruje se prema protokolu SPI korištenjem signala flash_cen, flash_sck, flash_so i flash_si (slika 15):



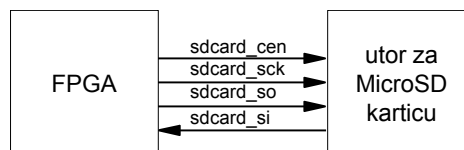
Slika 15: signali sučelja prema vanjskoj Flash memoriji

Deklaracija signala u sučelju sklopa (VHDL):

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

2.12 MicroSD kartica

Komunikacija s MicroSD memorijskom karticom ostvaruje se prema protokolu SPI korištenjem signala `sdcard_cen`, `sdcard_sck`, `sdcard_so` i `sdcard_si` (slika 16):



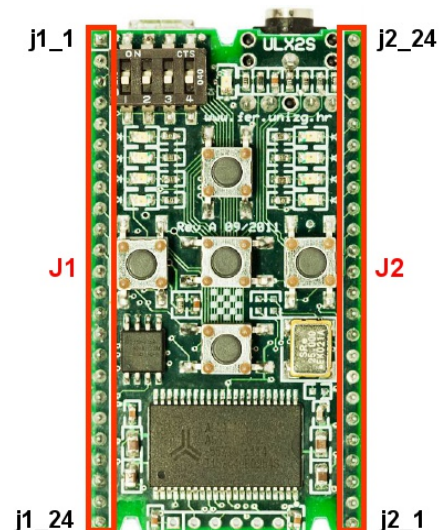
Slika 16: signali sučelja prema MicroSD kartici

Deklaracija signala u sučelju sklopa (VHDL):

```
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

S lijeve i desne strane razvojne pločice nalazi se po jedan stupac slobodnih kontakata u standardnom rasteru 2.54 mm na koje se mogu zalemiti spojnice za povezivanje s vanjskim uređajima ili za ugradnju u složeniji sustav. Svaki stupac sastoji se od 24 kontakta koji su numerirani prema oznakama sa slike 17. Kontakti `j1_1` i `j1_24` povezani su s izlazom iz regulatora napajanja 3.3 V (V_{CCIO}), dok su kontakti `j2_1` i `j2_24` povezani na uzemljenje (GND). Preostala 44 kontakta povezana su s FPGA sklopom, od kojih je 15 signala povezano i s komponentama ugrađenima na razvojnu pločicu (LED indikatori, tipke i težinska otporna mreža), pa je mogućnost njihove primjene ograničena. Popis signala koji su povezani i na ugrađene komponente i na priključnice za proširenje dokumentiran je u poglavlju 7.



Slika 17: priključnice za proširenje

Deklaracija signala u sučelju sklopa (VHDL):

```
port (  
    j1: inout std_logic_vector(23 downto 2);  
    j2: inout std_logic_vector(23 downto 2);  
);
```

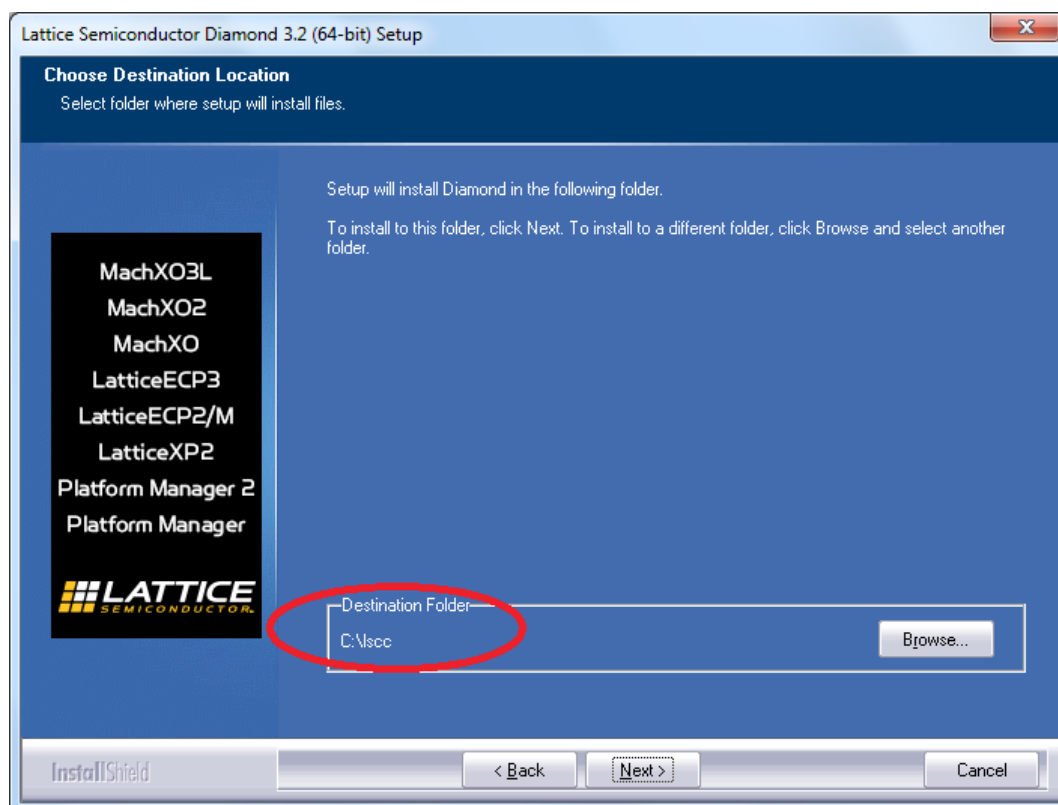
3 Instaliranje programske podrške

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. Uz verziju za Microsoftove 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 OS-X, Linux i FreeBSD.

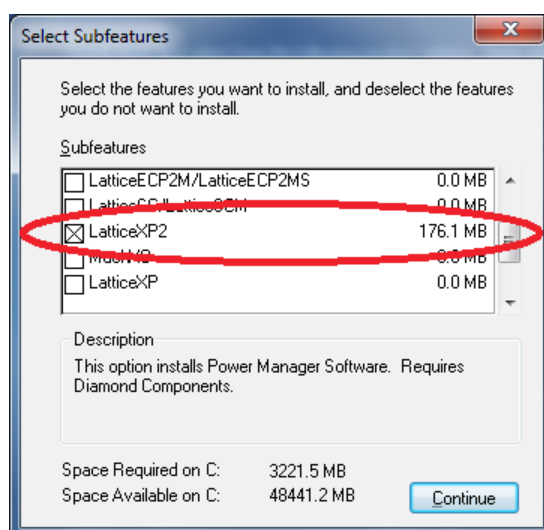
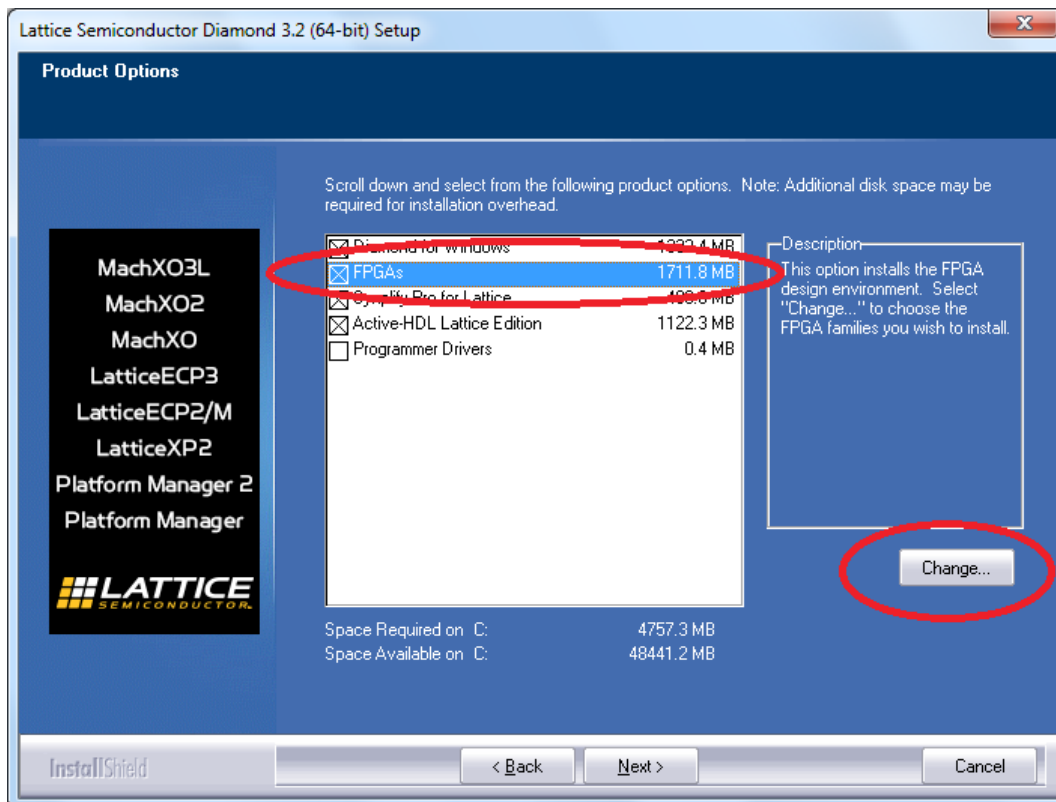
3.1 Instaliranje programskog paketa Lattice Diamond

S web sjedišta tvrtke Lattice Semiconductor (<http://www.latticesemi.com/>) potrebno je dohvatiti instalacijski paket alata Diamond za Microsoft Windows operacijske sustave. Na operacijskim sustavima Microsoft Windows 8 preporuča se koristiti isključivo najnoviju verziju Diamond 3.3. Za pristup dijelu web sjedišta koje omogućuje dohvaćanje instalacijskih paketa prethodno je potrebno izvršiti *online* registraciju. Veličina instalacijskog paketa je cca. 1.6 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.

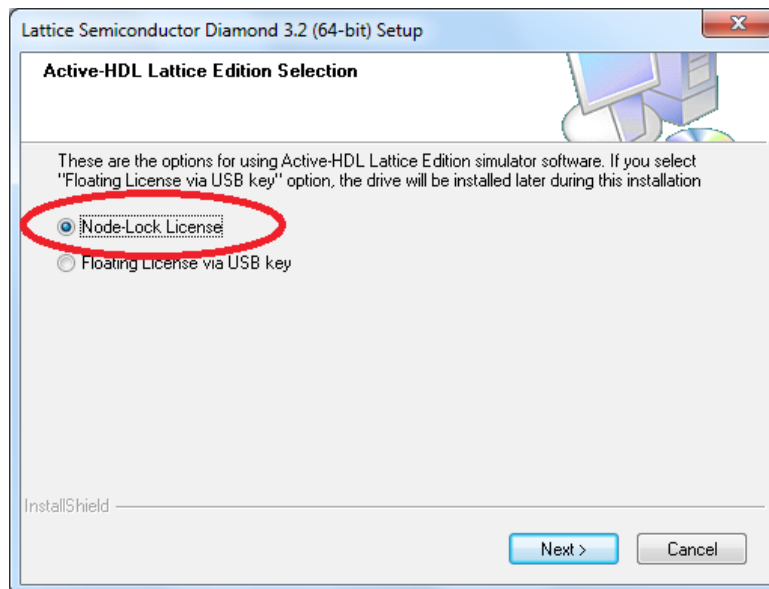
Instalacija programskog paketa s pretpostavljenim (*default*) postavkama zahtijeva cca. 5 GB slobodnog prostora na lokalnom disku. Programski paket se preporuča instalirati u pretpostavljeni (*default*) direktorij `c:\lsc`, jer su u starijim verzijama uočeni problemi u radu u slučaju odabira instalacije na alternativni disk ili direktorij.



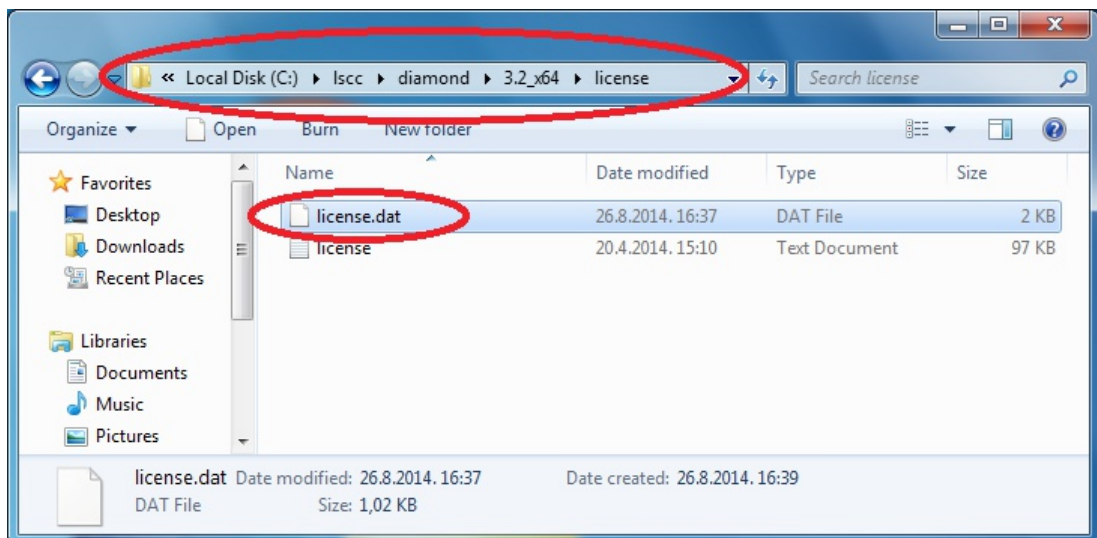
Kako instalacijski paket uključuje module za podršku svim FPGA sklopovima proizvođača Lattice Semiconductor, zauzeće diska može se bitno smanjiti (za oko 3 GB) ako se pri instalaciji odabere samo podrška za Lattice XP2 seriju FPGA sklopova koji su ugrađeni u ULX2S razvojne pločice. Također, nije potrebno instalirati upravljačke programe (drivere) za JTAG programatore tvrtke Lattice Semiconductor, budući da pločica ULX2S koristi vlastiti integrirani USB JTAG sklop za programiranje koji zahtijeva posebnu programsku podršku.



U nastavku instalacije potrebno je odabrati "node locked" model provjere licence za instaliranu programsku podršku.



Pritiskom na tipku "Next" završava se postupak instalacije, nakon čega je potrebno ispuniti *online* zahtjev za dodjelu besplatne licence za korištenje programskog paketa Lattice Diamond. Besplatna licenca vezana je uz MAC adresu Ethernet sučelja računala, pa je prilikom ispunjavanja *online* zahtjeva za licencu potrebno slijediti upute s web sjedišta www.latticesemi.com o načinu i formatu prijave odgovarajuće MAC adrese. Nakon što elektroničkom poštom pristigne datoteka s besplatnom licencom, datoteku `license.dat` potrebno je spremiti u direktorij `c:\lsc\diamond\3.2\license` ili `c:\lsc\diamond\3.2_x64\license`, zavisno od instalirane verzije alata Lattice Diamond.



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 je nužno instalirati posebni upravljački programski alat. Program `ujprog.exe` potrebno je dohvatiti s web sjedišta <http://www.nxlab.fer.hr/dl/ujprog.exe> i spremiti ga u direktorij

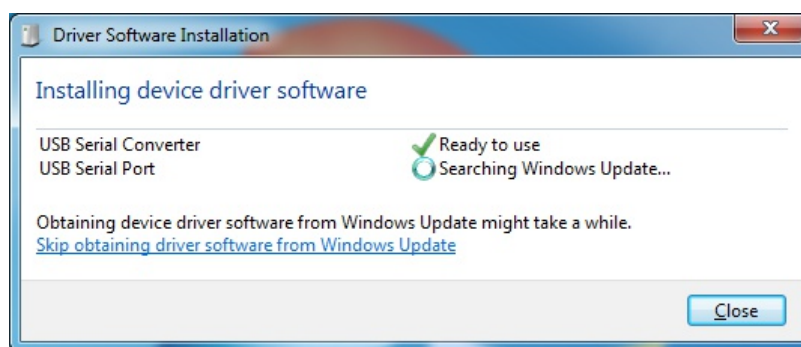
c:\Windows. Za pisanje po direktorijima operacijskog sustava korisnik treba imati administratorske ovlasti.

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 mreži Internet kako bi operacijski sustav mogao dohvatiti *driver*e s odgovarajućih poslužitelja.

Tijek pojavljivanja instalacijskih izbornika razlikuje se između operacijskih sustava Windows XP, Windows 7 i Windows 8, 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 sustavi Windows 7 i 8 bi *driver*e trebao instalirati automatski, bez intervencije korisnika.



Ako operacijski sustav po prvom priključenju pločice na računalo automatski ne pronađe i instalira pogonske programe (*driver*e), iste je potrebno dohvatiti s web sjedišta proizvođača USB sučelja: <http://www.ftdichip.com/FTDrivers.htm>.

3.4 Instaliranje programskih alata na operacijskom sustavu OS-X

Programski alati Lattice Diamond mogu se pokrenuti na operacijom sustavu Apple OS-X pomoću *emulacijske* platforme *Wine*, koja omogućuje izvršavanje 32-bitnih programa za Microsoft operacijske sustave pod *UNIX-like* operacijskim sustavima. Preduvjet za rad platforme *Wine* je instalirano okruženje *Xquartz*, za koje se instalacijski paket može dohvatiti s web sjedišta <http://xquartz.macosforge.org/landing/>.

Platformu *Wine* najjednostavnije je instalirati putem nekog od popularnih *package managera* za OS-X, npr. *Homebrew* (<http://brew.sh/>):

```
brew install wine
```

S web sjedišta <http://www.nxlab.fer.hr/dl/> potrebno je preuzeti alat `ujprog` za operacijski sustav OS-X. Datoteku `ujprog` treba učiniti izvršnom i instalirati naredbama:

```
sudo cp ujprog /usr/local/bin
sudo chmod +sx /usr/local/bin/ujprog
```

Pod okruženjem Wine provjereno radi **samo Diamond verzija 3.2, dok starije verzije, kao ni najnovija verzija 3.3, ne rade ispravno ili se uopće ne mogu instalirati**. S web sjedišta tvrtke Lattice Semiconductor (<http://www.latticesemi.com/>) dohvatite instalacijski paket alata Diamond za **32-bitne** Microsoft Windows operacijske sustave. Dohvaćenu *zip* arhivu raspakirajte.

Pokrenite instalacijski program unutar okruženja *Wine*:

```
wine 3.2.0.134_Diamond.exe
```

Slijedite korake odabira instalacijskih opcija opisane u poglavlju 3.1. Nakon završetka instalacijskog postupka datoteku s licencom za rad s paketom Lattice Diamond `license.dat` potrebno je spremiti u direktorij `~/.wine/drive_c/lssc/diamond/3.2/license/`.

Alat Lattice Diamond pokreće se naredbom:

```
wine "C:\lssc\diamond\3.2\bin\nt\pnmain.exe"
```

Prije prvog pozivanja alata `ujprog` potrebno je deaktivirati pogonske programe (drivere) operacijskog sustava za FTDI USB sučelja:

```
sudo kextunload /System/Library/Extensions/FTDIUSBSerialDriver.kext
```

Ovu naredbu potrebno je ponoviti nakon svakog *restarta* operacijskog sustava.

3.5 Instaliranje programskih alata na operacijskom sustavu Linux

Prije ostalih programskih paketa potrebno je instalirati paket `libusb-1.0-0-dev`, što se na operacijskim sustavima Ubuntu/Debian to se postiže naredbom:

```
sudo apt-get install libusb-1.0-0-dev
```

S web sjedišta <http://www.nxlab.fer.hr/dl/> potrebno je preuzeti alat `ujprog` za operacijski sustav Linux (32 ili 64-bitni). Datoteku `ujprog` treba učiniti izvršnom i instalirati naredbama:

```
sudo cp ujprog /usr/local/bin
sudo chmod +sx /usr/local/bin/ujprog
```

S web sjedišta tvrtke Lattice Semiconductor (<http://www.latticesemi.com/>) potrebno je dohvatiti instalacijski paket alata Diamond 3.2 za operacijski sustav Linux. U direktorij s instalacijskim paketom potrebno je spremiti i datoteku s licencom za korištenje paketa Lattice Diamond. Ukoliko koristite RPM package manager, paket Diamond može se instalirati pomoću ove naredbe:

```
rpm -i diamond_3_2-base_x64-134-x86_64-linux.rpm
```

Ukoliko **ne** koristite RPM package manager, paket Diamond moguće je instalirati alatom Alien:

```
sudo apt-get install alien
```

```
sudo alien -i diamond_3_2-base_x64-134-x86_64-linux.rpm
```

Postupak instalacije može potrajati oko desetak minuta ovisno o brzini računala, a posebno diskovnog podsustava. Nakon instalacije alata Diamond potrebno je još premjestiti datoteku s licencom za rad u za to predviđeni direktorij:

```
cp license.dat /usr/local/diamond/3.2/license/license.dat
```

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

```
% /usr/local/diamond/3.2/bin/lin/diamond & # za 32-bitni operacijski sustav
```

odnosno

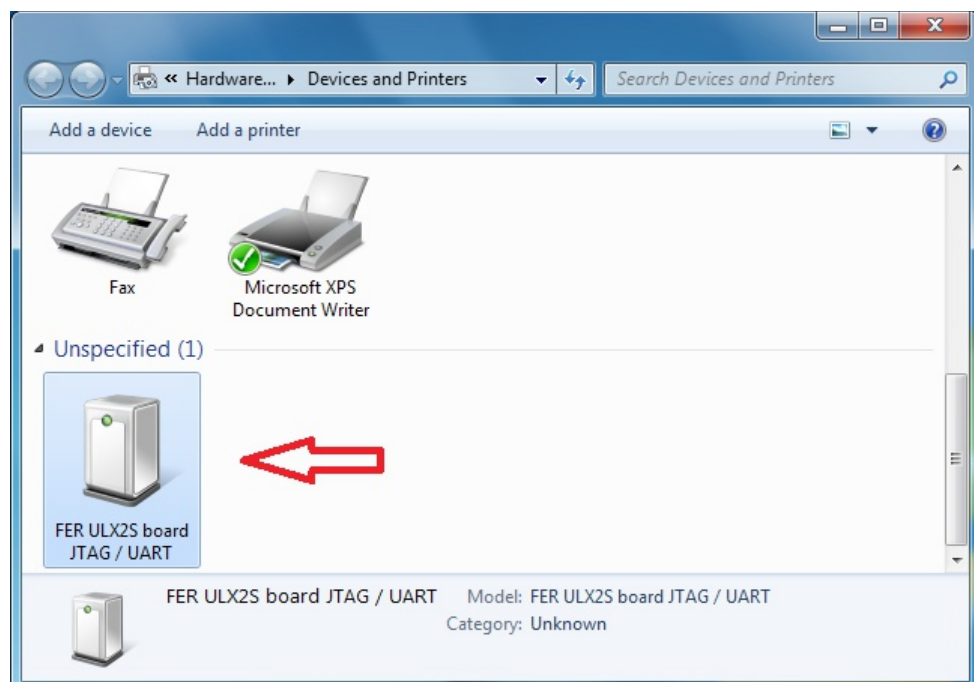
```
% /usr/local/diamond/3.2_x64/bin/lin64/diamond & # za 64-bitni operacijski sustav
```

4 Ispitivanje rada pločice

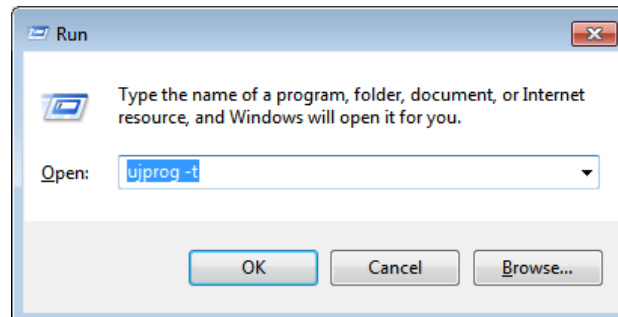
Razvojna pločica isporučuje se programirana dijagnostičko-demonstracijskom konfiguracijom FPGA sklopa koja omogućuje ispitivanje osnovnih funkcionalnosti pomoćnih integriranih sklopova ugrađenih na pločicu (generator takta, Flash i SRAM memorija) te ulazno / izlaznih priključaka i signala (RS-232, LED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

4.1 Ispitivanje USB komunikacije

Po uspješno završenoj proceduri instalacije USB *drivera*, pločica ULX2S bi operacijskom sustavu trebala biti istodobno vidljiva kao *Unspecified* uređaj u kategoriji *Devices and Printers*, te kao standardni USB serijski (RS-232) port. Dok je pločica priključena na računalo uz pokrenut operacijski sustav kontrolni LED indikator USB sučelja na pločici bi trebao biti stalno upaljen.



Serijsku komunikaciju s razvojnom pločicom najjednostavnije je ostvariti korištenjem alata `ujprog` kojeg se poziva iz komandne ljuške (*command shell*) operacijskog sustava. Na operacijskom sustavu Microsoft Windows 7 komandnu ljušku može se otvoriti tako da se u pregledniku datoteka (explorer) desnim klikom miša **uz pritisak na tipku *shift*** u padajućem izborniku odabere akcija "Open command window here", ili kroz izbornik *Run program* kojeg se otvara kombinacijom tipki "Windows" i "R":



Ukoliko je FPGA sklop programiran demonstracijskom konfiguracijom, na zaslonu računala bi se po pokretanju alata `ujprog` uz opciju `-t` trebala ispisivati poruka demonstracijskog programa, kojeg se može prekinuti pritiskom na kombinaciju tipki CTRL+C. Iz programa `ujprog` izlazi se utipkavanjem slijeda tipki "ENTER", "~", ".".

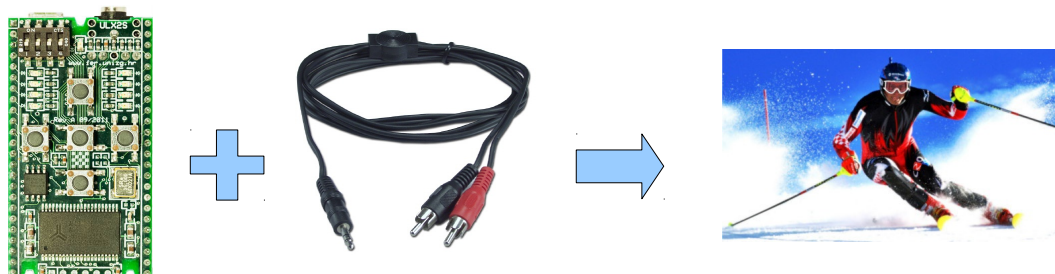
```
C:\Windows\system32\cmd.exe - ujprog -t
ULX2S JTAG programmer v 1.12 $Id: ujprog.c 2242 2014-09-28 16:44:07Z marko $
Using USB cable: FER ULX2S board JTAG / UART
Terminal emulation mode, using 115200 bauds
Press ENTER, ~, ? for help
Ucitavam sliku iz demo/pics/zastava.jpg
Ucitavam sliku iz demo/pics/biokovo.jpg
Ucitavam sliku iz demo/pics/bjelovar.jpg
breaking at line 200
Ready
>dir
Directory for C:/
<DIR>      boot
<DIR>      demo
<DIR>      bench
          1300  copyright.txt
          38616  readme.txt
           79   autoexec.bas
39 Kbytes in 6 files, 1312 Kbytes free.
Ready
>
```

Za komunikaciju s razvojnom pločicom putem USB–RS-232 sučelja može se koristiti i bilo koji drugi program za emulaciju asinkronog terminala. Preporuka je koristiti program **Putty**, koji je dostupan za besplatno dohvaćanje na web sjedištu <http://www.putty.org>. Odabrani program za emulaciju asinkronog terminala treba podesiti za rad pri brzini od 115200 bps, 8 bitova po znaku, bez pariteta i bez korištenja dodatnih upravljačkih signala (RTS/CTS ili DSR/DTR).

4.2 Ispitivanje video izlaza

U sklopu demonstracijske konfiguracije razvojne pločice sintetiziran je i modul za generiranje video signala. Za reprodukciju ispitnog video signala pločicu je potrebno povezati s ekranom ili TV prijemnikom s ulazom za kompozitni analogni PAL video signal. Za povezivanje se može koristiti kabel tipa 3.5 mm stereo na *RCA* (*cinch* /

phono), koji se uobičajeno isporučuje kao standardna oprema digitalnih fotoaparata i kamera, a može se nabaviti i zasebno u većini trgovina elektroničkom opremom. U kompozitni video ulaz TV prijemnika ili monitora treba utaknuti onaj RCA konektor koji je povezan s vrškom (*tip*) 3.5 mm stereo utikača kojeg se umeće u stereo utičnicu na razvojnoj pločici, a preostali RCA konektori mogu ostati odspojeni ili se mogu utaknuti u audio ulaze ekrana ili TV prijemnika.



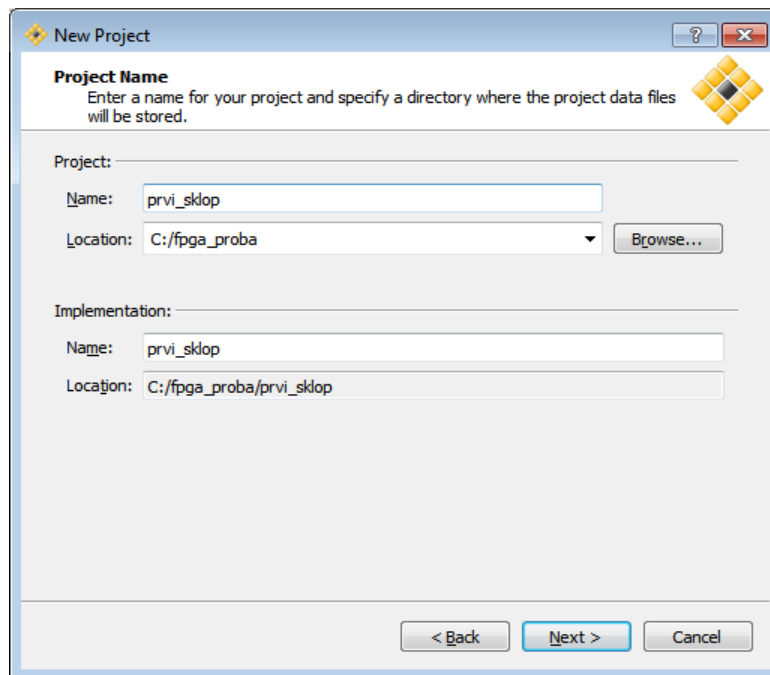
Demonstracijska konfiguracija razvojne pločice na ekranu će ciklički prikazivati nekoliko slika pohranjenih na SPI Flash memoriju. Prikazom slika upravlja jednostavan program napisan u jeziku BASIC, o čemu je više detalja izloženo u poglavlju 6.

5 Rad s razvojnim alatom Lattice Diamond

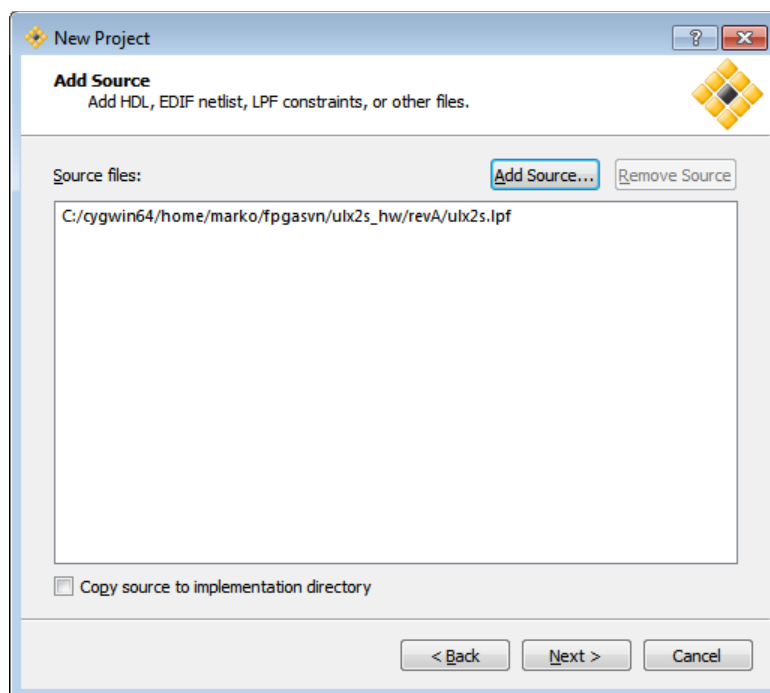
Glavna namjena programskog alata Lattice Diamond je sinteza konfiguracije FPGA sklopova iz odgovarajućeg opisa odnosno specifikacije digitalnog sklopa. Alat omogućuje opis digitalnog sklopa unosom shematskih dijagrama, ili korištenjem jezika za opis sklopova VHDL ili Verilog. U nastavku je dan primjer 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

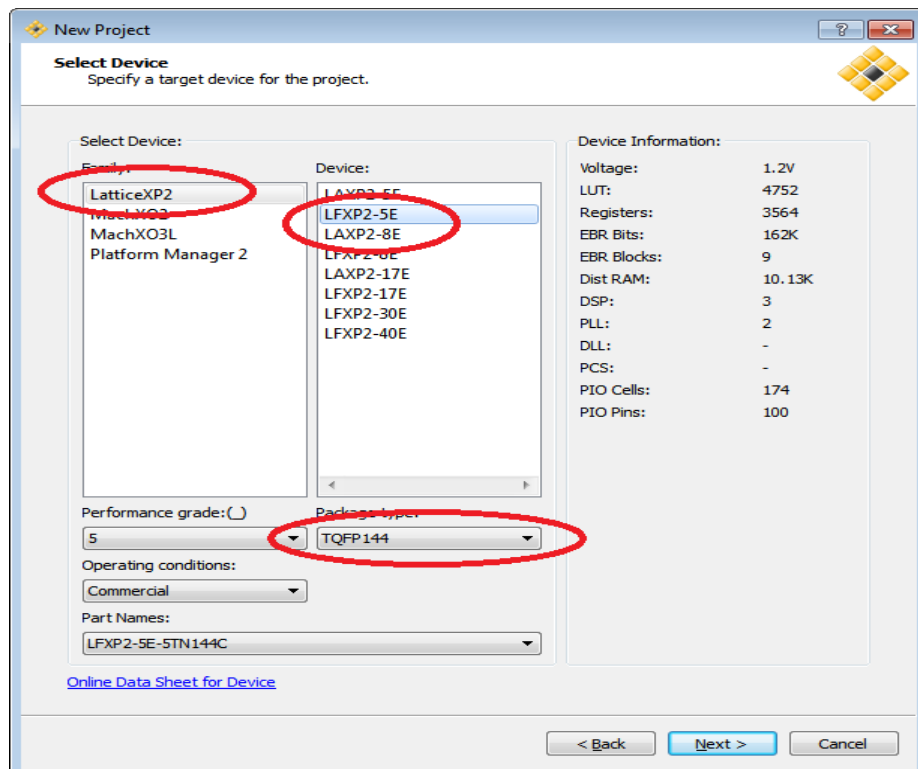
Pokrenite alat Lattice Diamond. Odaberite kreiranje novog projekta: *Project -> New*. Odaberite ime novog projekta, te radni direktorij u kojem će biti pohranjene ulazne datoteke vezane uz projekt.



U slijedećem koraku alat nudi izbornik za dodavanje već postojećih (unaprijed pripremljenih) datoteka u novi projekt. U projekt treba uključiti datoteku koja određuje poveznice 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 poglavlju 7.



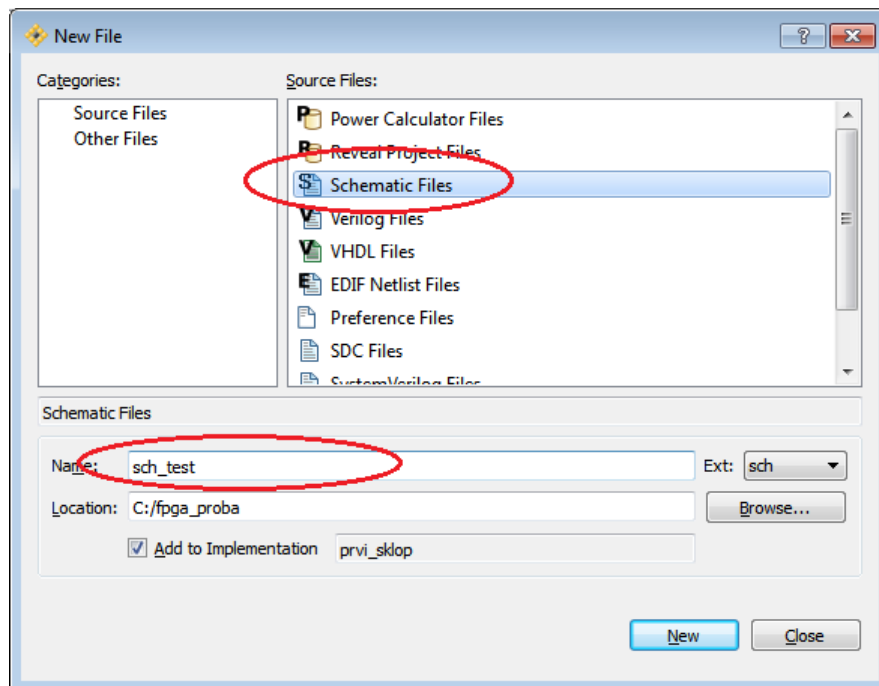
U slijedećem koraku potrebno je odabrati točan tip FPGA sklopa. Na pločicu ULX2S ugrađuje se FPGA sklop **LFXP2-5E** ili **LFXP2-8E**, brzine rada (*speed grade*) 5, u kućištu **TQFP144**.



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

5.2 Shematski opis digitalnog sklopa

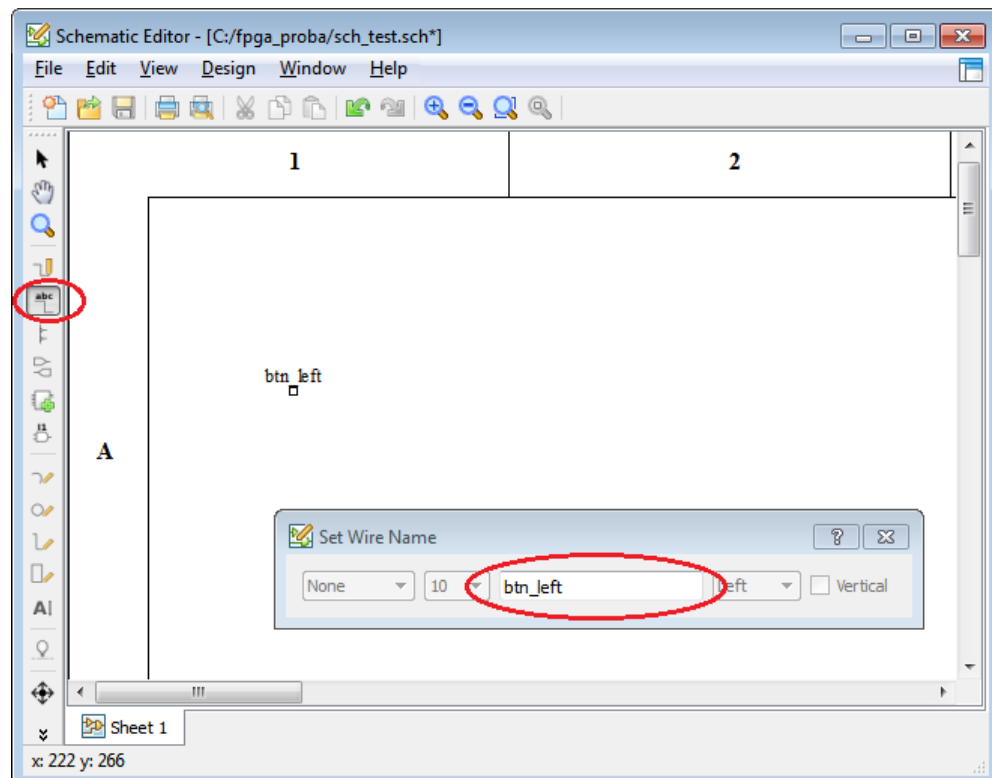
Kako bi mogli shematski opisati željeni digitalni sklop, potrebno je prvo stvoriti novu praznu *schematic file* datoteku i uključiti ju u trenutni projekt. Postupak za stvaranje novih VHDL ili Verilog datoteka i njihovo uključivanje u projekt je identičan, međutim zbog preglednosti i jednostavnosti u 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`).



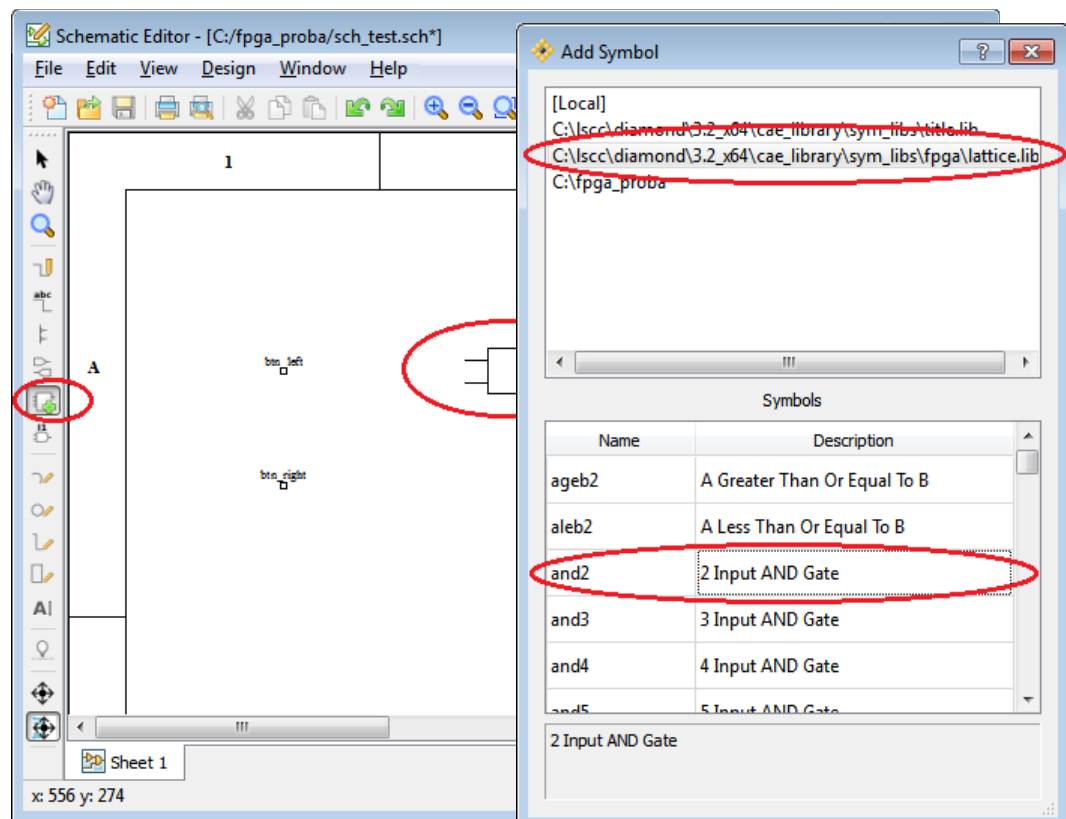
U ovom primjeru prikazan je postupak specificiranja sklopa s dva ulaza (tipke `btn_left` i `btn_right`) i dva izlaza (LED indikatori `led_0` i `led_1`), na koje se povezuju izlazi iz sklopova "I" (*and*) i "ILI" (*or*).

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 površinu 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".

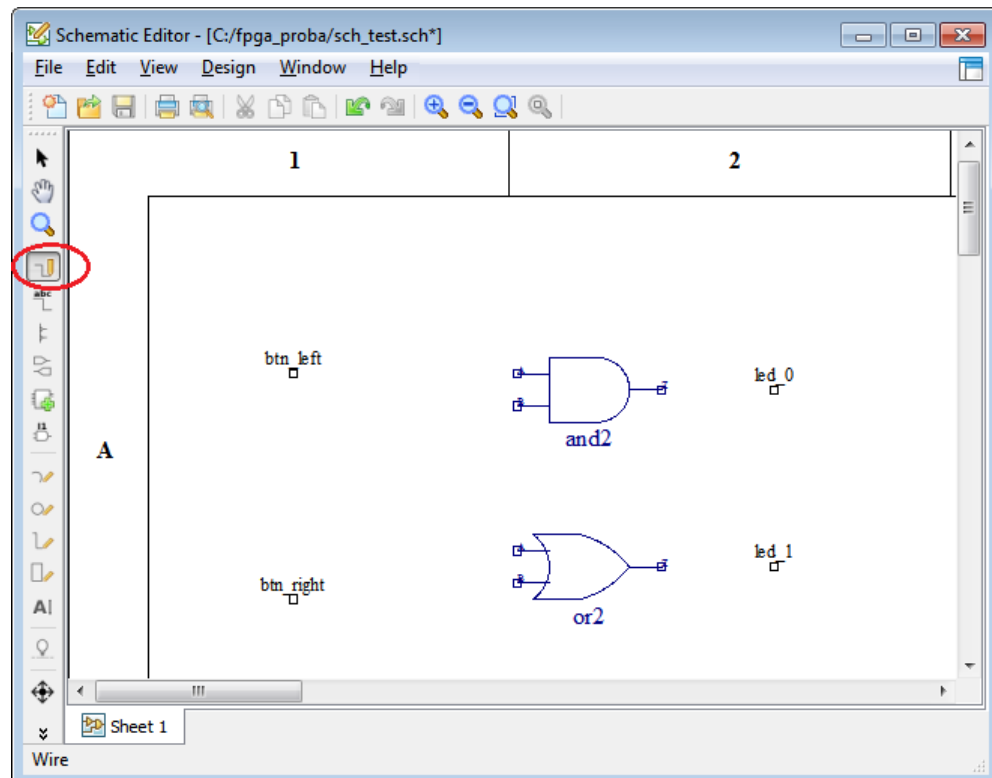
U prvom koraku treba stvoriti priključne točke za ulazne i izlazne signale, i dodjeliti im odgovarajuća imena. Dodjeljivanje imena signalima moguće je izvršiti odabirom alata "*Net name*" na izborničkoj traci s lijeve strane shematskog uređivača.



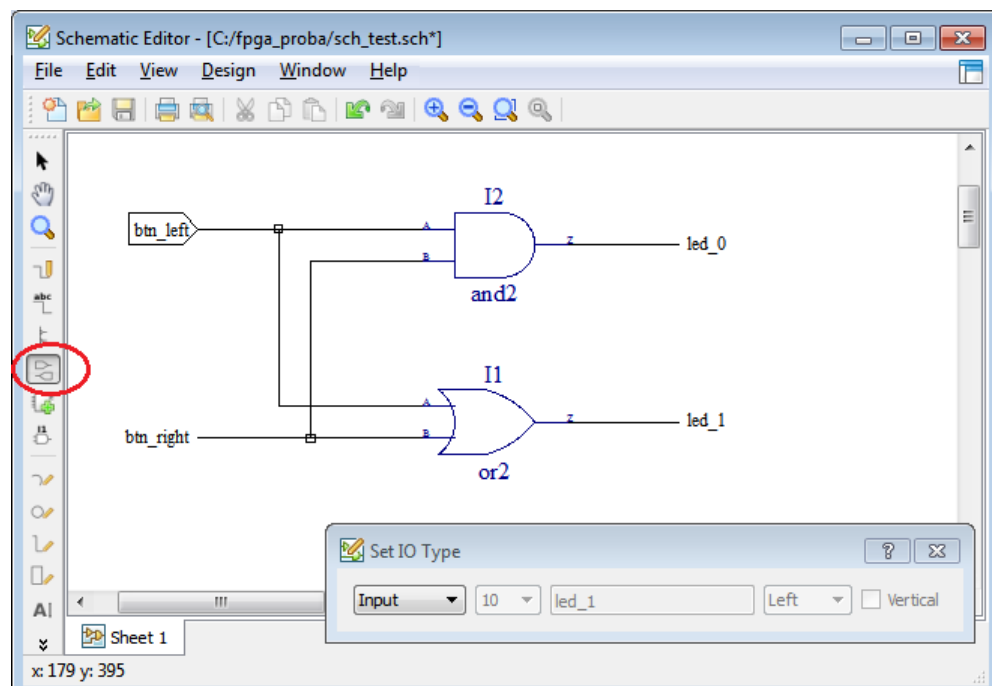
Nakon što su stvorene i ispravno imenovane oznake za sva četiri željena ulazno / izlazna signala, odabirom alata "Symbol" potrebno je otvoriti izbornik za postavljanje komponenti iz neke od dostupnih biblioteka u shematski dijagram. Treba odabrati biblioteku `lattice.lib`, iz koje se može u shematski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



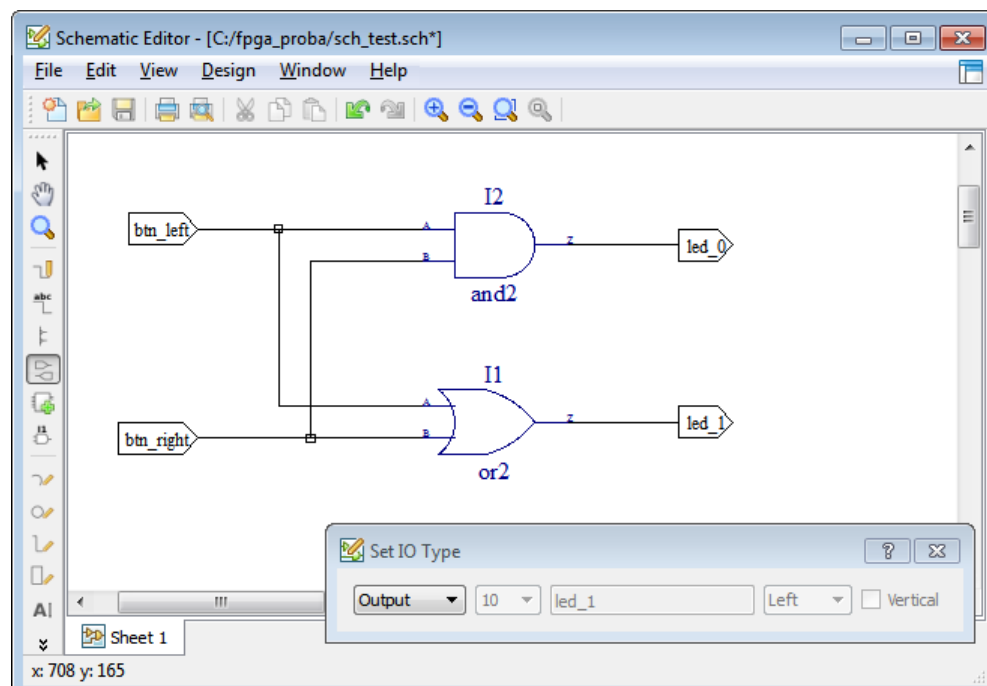
Nakon što su instance sklopova `and2` i `or2` postavljene na radnu plohu, instance sklopova `and2` i `or2` može se povezati s ulaznim odnosno izlaznim signalima odabirom alata "Wire" na izborničkoj traci s lijeve strane prozora.



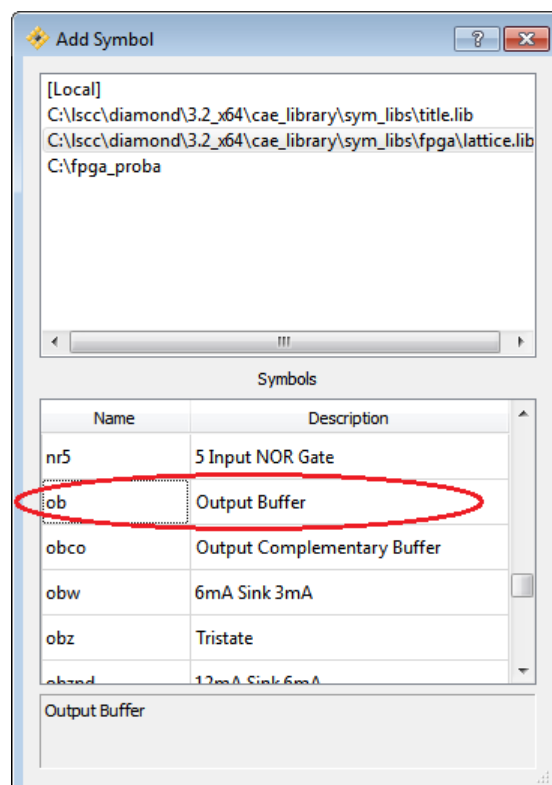
Nakon što su svi sklopovi i signali međusobno povezani na željeni način, potrebno je signale deklarirati kao ulazne, odnosno izlazne, korištenjem alata "IO Port" iz izborničke trake s lijeve strane prozora. U pomoćnom prozoru "Set IO Type" potrebno je odabrati ulazni (input) odnosno izlazni (output)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

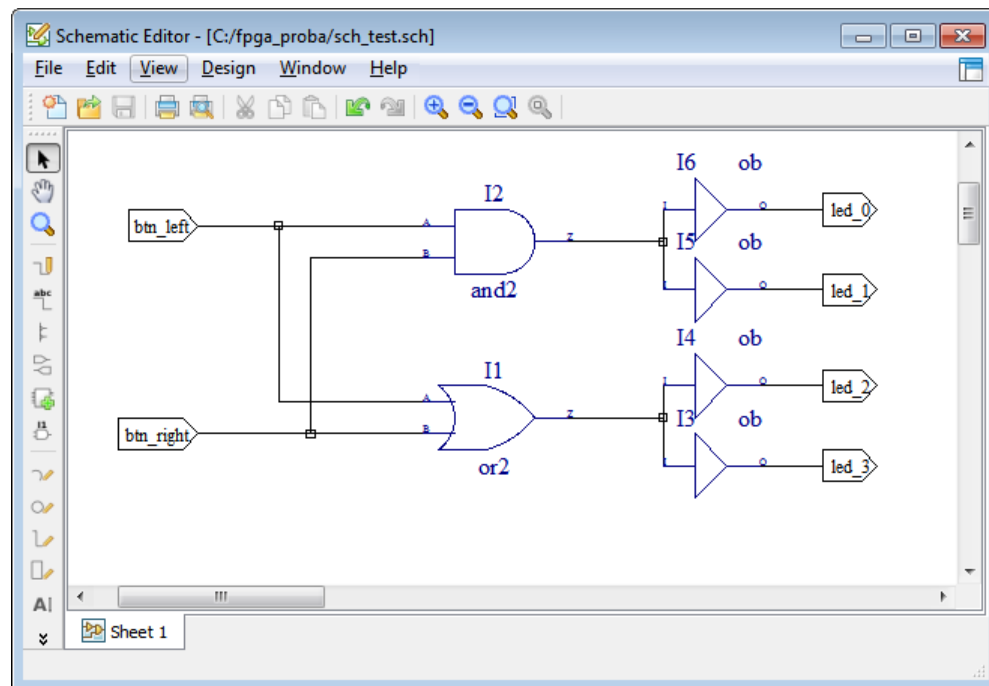


Ogledni sklop sad je potpuno specificiran i spreman za sintezu:



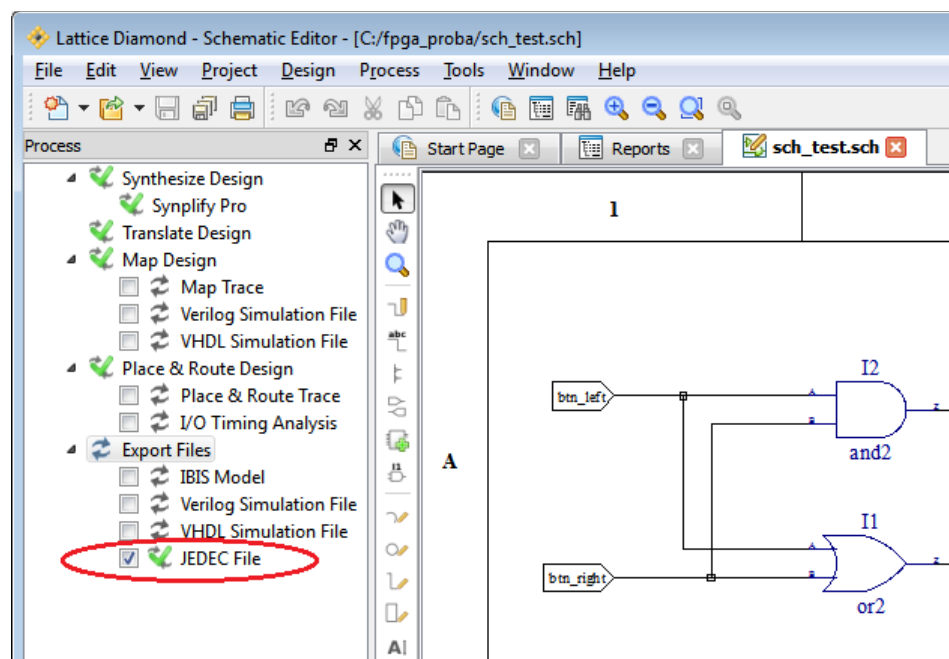
U slučaju da je određeni signal potrebno dovesti na više od jednog izlaznog sučelja, svakom izlazu treba dodijeliti zasebno izlazno pojačalo (*eng. output buffer*). U slijedećem primjeru svaki od izlaznih signala povezan je na zasebno izlazno pojačalo, koje se u biblioteci shematskih simbola krije pod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

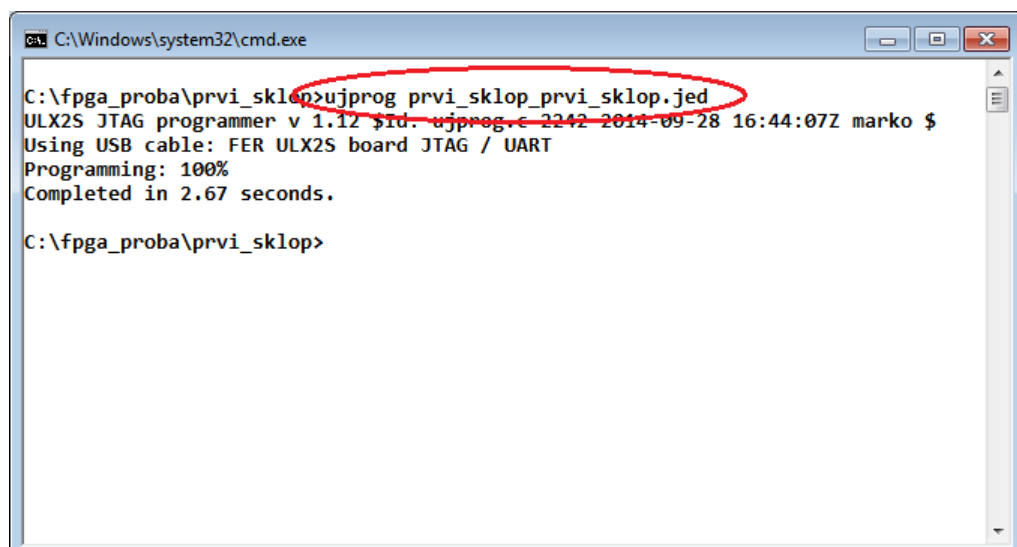
5.4 Programiranje FPGA sklopa

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 ljuške (*command shell*). Ljuskicu operacijskog sustava može se otvoriti prema uputama iz poglavlja 4.1.

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_proba\prvi_sklop`. Konfiguracijske datoteke FPGA sklopova koje stvori alat Lattice Diamond imaju ekstenziju `.jed`. U ovom primjeru alat je stvorio konfiguracijsku datoteku pod imenom `prvi_sklop_prvi_sklop.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 konfiguriranja 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.



```
C:\Windows\system32\cmd.exe

C:\fpga_proba\prvi_sklop>ujprog prvi_sklop_prvi_sklop.jed
ULX2S JTAG programmer v 1.12 $Id: ujprog.c 2242 2014-09-28 16:44:07Z marko $
Using USB cable: FER ULX2S board JTAG / UART
Programming: 100%
Completed in 2.67 seconds.

C:\fpga_proba\prvi_sklop>
```

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`, kako je prikazano na slici.

```
C:\Windows\system32\cmd.exe

C:\fpga_proba\prvi_sklop>ujprog prvi_sklop_prvi_sklop.jed
ULX2S JTAG programmer v 1.12 $Id: ujprog.c 2242 2014-09-28 16:44:07Z marko $
Using USB cable: FER ULX2S board JTAG / UART
Programming: 100%
Completed in 2.67 seconds.

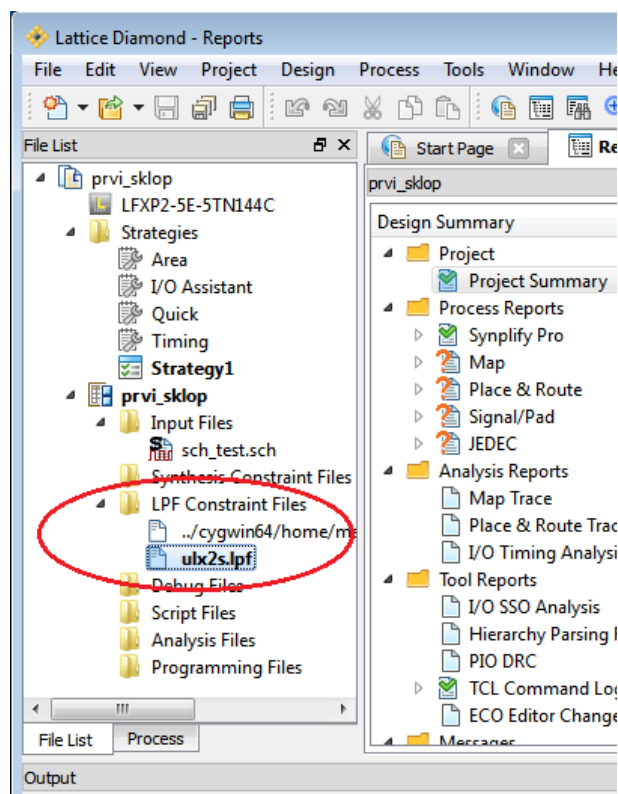
C:\fpga_proba\prvi_sklop>ujprog -j flash prvi_sklop_prvi_sklop.jed
ULX2S JTAG programmer v 1.12 $Id: ujprog.c 2242 2014-09-28 16:44:07Z marko $
Using USB cable: FER ULX2S board JTAG / UART
Programming: 100%
Completed in 23.04 seconds.

C:\fpga_proba\prvi_sklop>
```

5.5 Uobičajeni problemi u radu s FPGA razvojnim alatima

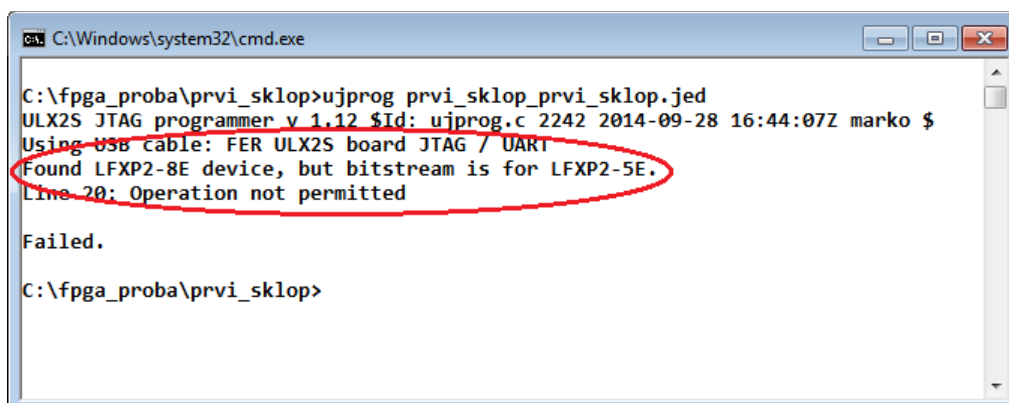
5.5.1 Definicije logičkih naziva vanjskih priključaka

Ako u projekt nije uključena datoteka s definicijama vanjskih priključaka FPGA sklopa `ulx2s.lpf`, ili je ta datoteka prazna, sintetizator ne može povezati logičke nazive ulaznih i izlaznih signala s fizičkim pinovima. Tipični simptom je da sintetizirani sklop ne reagira na pobudu, a svi LED indikatori tinjaju slabim intenzitetom. U ovakvom slučaju u izborniku "File list" na lijevoj strani radne površine treba desnim klikom na ikonu "LPF Constraint Files" odabrati "Add existing file" te u projekt dodati datoteku `ulx2s.lpf`, nakon čega istu treba obavezno označiti desnim klikom miša odabirom opcije "Set as Active Preference File".



5.5.2 Odabir pogrešnog FPGA sklopa

Na pločicu ULX2S ugrađen je FPGA sklop **LFXP2-8E** ili **LFXP2-5E** u kućištu tipa **TQFP-144**. U slučaju pogrešnog odabira prilikom konfiguriranja FPGA sklopa javlja se poruka "Bitstream for unsupported target" ili "Found xxx device, but bitstream is for yyy". U izborniku "File list" potrebno je dvostrukim klikom na tip FPGA sklopa otvoriti odgovarajući prozor i odabrati ispravan tip sklopa.



```
C:\Windows\system32\cmd.exe
C:\fpga_proba\prvi_sklop>ujprog prvi_sklop_prvi_sklop.jed
ULX2S JTAG programmer v 1.12 $Id: ujprog.c 2242 2014-09-28 16:44:07Z marko $
Using USB cable: FER ULX2S board JTAG / UART
Found LFXP2-8E device, but bitstream is for LFXP2-5E.
Line 20: Operation not permitted

Failed.

C:\fpga_proba\prvi_sklop>
```

5.5.3 Specijalni znakovi u imenu projekta i datoteka

Put do kazala (*foldera*) u kojeg su smještene datoteke projekta, kao i ime projekta i svih datoteka koje su u njega uključene, smiju sadržavati samo velika i mala slova engleske abecede, brojeke, te znak "_". Korištenje ostalih znakova (posebno razmaka i zagrada) dovodi do pogrešaka u radu alata Lattice Diamond.

5.5.4 Projekt bez datoteka

Datoteke koje nisu uključene u projekt mogu se uređivati (editirati) ali se ne mogu sintetizirati, pa se preporuča uređivati isključivo datoteke koje su vidljive u izborniku "File list" u odjeljku "Input files", na način da se u tom izborniku klikne na željenu datoteku.

5.5.5 Nepotrebne datoteke u projektu

Sličan problem predstavlja i uključivanje nepotrebnih datoteka u projekt, uslijed čega sintetizator ne može ispravno odabrati glavni (*top-level*) modul, ili prijavljuje različite poruke o grešci prilikom sinteze, ili sinteza rezultira nefunkcionalnim sklopom. U izborniku "File list" potrebno je iz popisa datoteka "Input files" izbrisati sve datoteke koje su eventualno greškom uključene u projekt.

5.5.6 Odabir glavnog (*top-level*) modula

Kod shematskog opisa sintetizator ponekad ne odabere ispravno glavni modul sklopa, što se događa u pravilu kad se prije prvog pokretanja postupka sinteze propusti u projekt uključiti sve potrebne datoteke, a tipično se radi o datotekama s definicijama shematskih simbola (.sym) koje povezuju implementaciju nekog od *black-box* modula s njegovim grafičkim prikazom u shematskom editoru. Problem se nakon dodavanja potrebnih odnosno uklanjanja suvišnih datoteka iz projekta može riješiti uključivanjem opcije "View – Show Views – Hierarchy" koja će indirektno rezultirati ispravnim odabirom glavnog (*top-level*) modula.

6 Razvojna pločica kao samostalni mikroprocesorski sustav

Sintezom odgovarajuće procesorske jezgre na FPGA sklopu razvojna pločica može raditi i kao samostalni mikroprocesorski sustav. Velik broj različitih procesorskih jezgri opisanih jezicima VHDL ili Verilog dostupan je besplatno na <http://www.opencores.org/>. Tvrtka Lattice Semiconductor razvila je i vlastite, također besplatne 8-bitne i 32-bitne procesorske jezgre (<http://www.latticesemi.com/mico32>) koje su prilagođene njihovim FPGA sklopovima i za koje su dostupni i integrirani razvojni alati.

Prilagodba i sinteza za rad na FPGA sklopu mikroprocesorskog sustava temeljenog na *soft-core* procesorskim jezgrama, a posebno povezivanje s vanjskim memorijama i uređajima, te razvoj odgovarajuće programske podrške može biti kompleksan i vremenski zahtjevan zadatak. Zbog toga je za pločicu ULX2S pripremljena već gotova (sintetizirana) mikroprocesorska konfiguracija "MIPS" koja omogućuje komunikaciju i programsko upravljanje sa svim sklopovima ugrađenima na razvojnu pločicu, a temelji se na 32-bitnoj RISC procesorskoj jezgri koja radi na taktu frekvencije 81.25 Mhz.

6.1 Instaliranje mikroprocesorske konfiguracije MIPS na FPGA sklop

Nove pločice isporučuju se već programirane spomenutom mikroprocesorskom konfiguracijom koja se koristi u demonstracijske i ispitne svrhe. Upute za programiranje pločica starije generacije, kao i novijih pločica na kojima je demonstracijsko-ispitna konfiguracija izbrisana, dostupne su na <http://www.nxlab.fer.hr/dl/firmware.html>.

6.2 Priprema MIPS programskih alata

Demonstracijsko-ispitna mikroprocesorska konfiguracija može se programirati u jezicima C i BASIC. Upute za rad s BASIC interpreterom na pločici ULX2S dostupne su na web sjedištu <http://www.nxlab.fer.hr/dl/basic.pdf>.

Za razvoj programa u jeziku C potrebno je pripremiti i instalirati odgovarajuće razvojne alate temeljene na GNU *toolchainu*. Na računalima s Microsoftovim operacijskim sustavima GNU razvojni alati moraju se instalirati unutar okruženja *Cygwin*, za koje se instalacijski paket može dohvatiti sa stranice <http://cygwin.org>. Prilikom instaliranja potrebno je odabrati opcionalne pakete `devel/gcc-core`, `devel/make` i `utils/diffutils`.

Nakon što je na računalo uspješno instalirana *Cygwin* radna okolina, potrebno je otvoriti interaktivnu *cygwin* ljusku unutar koje se provodi postupak instaliranja GNU razvojnih alata. U *home* direktorij potrebno je dohvatiti te raspakirati paket biblioteka i programa http://www.nxlab.fer.hr/dl/f32c_src.tgz. Slijedite upute za instaliranje GNU alata iz datoteke `f32c/src/patches/instalacija_gnu_toolova.txt`.

```
$ tar xf f32c_src.tgz
$ ls
f32c          f32c_src.tgz
$ less f32c/src/patches/instalacija_gnu_toolova.txt
```

Nakon što su GNU alati uspješno instalirani potrebno je podesiti *environment* varijablu `MAKEFILES` tako da pokazuje na datoteku `~/f32c/src/conf/f32c.mk`. U Bourne ljusci (*bash*) koja je pretpostavljena ljuska okruženja *Cygwin* to se postiže naredbom:

```
$ export MAKEFILES=~/f32c/src/conf/f32c.mk
```

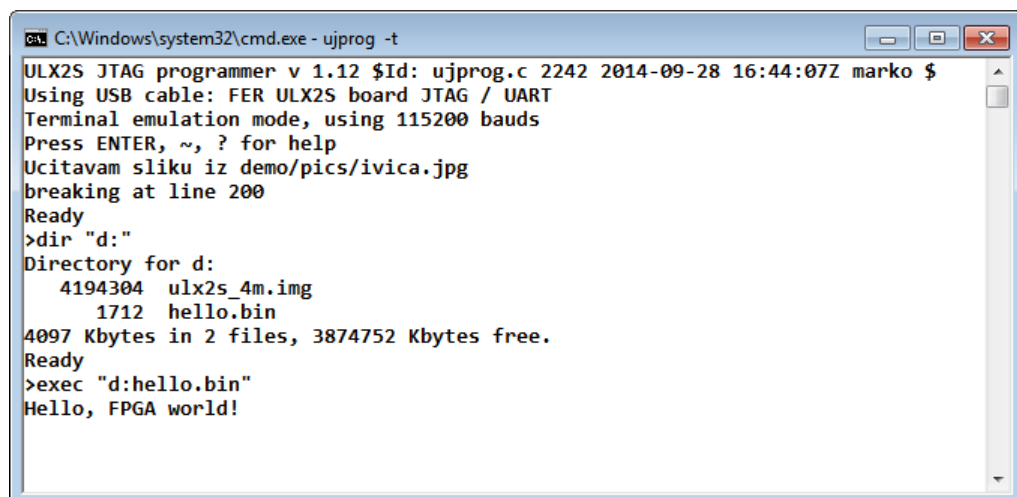

Postavljanje varijable `MAKEFILES` može se automatizirati na način da se prethodna naredba doda u datoteku `~/.bash_profile`.

6.3 Programiranje u jeziku C

Uz ispravno pripremljene GNU razvojne alate, prevođenjem programa u jeziku C najjednostavnije je upravljati putem već postojećih make pravila. U nastavku je prikazan primjer prevođenja oglednog C programa:

```
$ cd f32c/src/examples/
$ ls
hello_asm      hello_c      video_test
$ cd hello_c
$ ls
Makefile       hello.c
$ make
...
mips-elf-objcopy -O binary hello hello.bin
mips-elf-objcopy -O srec hello hello.hex
```

Prevedeni izvršni kod s ekstenzijom `.bin` može se kopirati na MicroSD karticu, te na FPGA pločici pokrenuti iz BASIC interpretera naredbom `exec`:



```
C:\Windows\system32\cmd.exe - ujprog -t
ULX2S JTAG programmer v 1.12 $Id: ujprog.c 2242 2014-09-28 16:44:07Z marko $
Using USB cable: FER ULX2S board JTAG / UART
Terminal emulation mode, using 115200 bauds
Press ENTER, ~, ? for help
Ucitavam sliku iz demo/pics/ivica.jpg
breaking at line 200
Ready
>dir "d:"
Directory for d:
  4194304  ulx2s_4m.img
    1712   hello.bin
4097 Kbytes in 2 files, 3874752 Kbytes free.
Ready
>exec "d:hello.bin"
Hello, FPGA world!
```

Ako se izvršna datoteka na MicroSD karticu pohrani pod imenom `bootme.bin`, kod uključnja pločice program će se učitati i pokrenuti automatski.

7 Definicije ulazno / izlaznih signala

```
#
# FER ULX2S board I/O constraints
#
# Board revision A, Sep 2011
# Constraints file revision 1.03, 2013/12/08
#

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 j2_17)
LOCATE COMP "led_1" SITE "37"; # OUT (shared with j2_18)
LOCATE COMP "led_2" SITE "36"; # OUT (shared with j2_19)
LOCATE COMP "led_3" SITE "35"; # OUT (shared with j2_20)
LOCATE COMP "led_4" SITE "144"; # OUT (shared with j1_12)
LOCATE COMP "led_5" SITE "5"; # OUT (shared with j1_7)
LOCATE COMP "led_6" SITE "6"; # OUT (shared with j1_6)
LOCATE COMP "led_7" SITE "138"; # OUT (shared with j1_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 j1_10)
LOCATE COMP "btn_left" SITE "2"; # IN (shared with j1_11)
LOCATE COMP "btn_right" SITE "38"; # IN (shared with j2_14)
LOCATE COMP "btn_center" SITE "22"; # IN (shared with j2_15)

# 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 j2_21)
LOCATE COMP "p_tip_1" SITE "27"; # INOUT (665 Ohm) (shared with j2_22)
LOCATE COMP "p_tip_2" SITE "28"; # INOUT (330 Ohm) (shared with j2_23)
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

# 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

# 2 x 24 pin DIL header: j1 (left side)
# "j1_1" -> VCCIO (3.3 V)
LOCATE COMP "j1_2" SITE "131"; # INOUT
LOCATE COMP "j1_3" SITE "134"; # INOUT
LOCATE COMP "j1_4" SITE "9"; # INOUT
LOCATE COMP "j1_5" SITE "138"; # INOUT (shared with led_7)
LOCATE COMP "j1_6" SITE "6"; # INOUT (shared with led_6)
LOCATE COMP "j1_7" SITE "5"; # INOUT (shared with led_5)
LOCATE COMP "j1_8" SITE "8"; # INOUT
LOCATE COMP "j1_9" SITE "7"; # INOUT
LOCATE COMP "j1_10" SITE "1"; # INOUT (shared with btn_down)
LOCATE COMP "j1_11" SITE "2"; # INOUT (shared with btn_left)
LOCATE COMP "j1_12" SITE "144"; # INOUT (shared with led_4)
LOCATE COMP "j1_13" SITE "143"; # INOUT
LOCATE COMP "j1_14" SITE "142"; # INOUT
LOCATE COMP "j1_15" SITE "141"; # INOUT
LOCATE COMP "j1_16" SITE "133"; # INOUT
LOCATE COMP "j1_17" SITE "132"; # INOUT
LOCATE COMP "j1_19" SITE "123"; # INOUT
LOCATE COMP "j1_18" SITE "124"; # INOUT
LOCATE COMP "j1_20" SITE "115"; # INOUT
LOCATE COMP "j1_21" SITE "114"; # INOUT
LOCATE COMP "j1_22" SITE "110"; # INOUT
LOCATE COMP "j1_23" SITE "109"; # INOUT
# "j1_24" -> VCCIO (3.3 V)

# 2 x 24 pin DIL header: j2 (right side)
# "j2_1" -> GND
LOCATE COMP "j2_2" SITE "72"; # INOUT
LOCATE COMP "j2_3" SITE "71"; # INOUT
LOCATE COMP "j2_4" SITE "70"; # INOUT
LOCATE COMP "j2_5" SITE "69"; # INOUT
LOCATE COMP "j2_6" SITE "66"; # INOUT
LOCATE COMP "j2_7" SITE "65"; # INOUT
LOCATE COMP "j2_8" SITE "52"; # INOUT
LOCATE COMP "j2_9" SITE "50"; # INOUT
LOCATE COMP "j2_10" SITE "48"; # INOUT
LOCATE COMP "j2_11" SITE "47"; # INOUT
LOCATE COMP "j2_12" SITE "40"; # INOUT
LOCATE COMP "j2_13" SITE "39"; # INOUT
LOCATE COMP "j2_14" SITE "38"; # INOUT (shared with btn_right)
LOCATE COMP "j2_15" SITE "22"; # INOUT (shared with btn_center)
LOCATE COMP "j2_16" SITE "31"; # INOUT
LOCATE COMP "j2_17" SITE "29"; # INOUT (shared with led_0)
LOCATE COMP "j2_18" SITE "37"; # INOUT (shared with led_1)
LOCATE COMP "j2_19" SITE "36"; # INOUT (shared with led_2)
LOCATE COMP "j2_20" SITE "35"; # INOUT (shared with led_3)
LOCATE COMP "j2_21" SITE "32"; # INOUT (shared with p_tip_0)
LOCATE COMP "j2_22" SITE "27"; # INOUT (shared with p_tip_1)
LOCATE COMP "j2_23" SITE "28"; # INOUT (shared with p_tip_2)
# "j2_24" -> GND

#
# Breadboard signal aliases
#
LOCATE COMP "lcd_data_0" SITE "8"; # OUT (shared with j1_8)
LOCATE COMP "lcd_data_1" SITE "9"; # OUT (shared with j1_4)
LOCATE COMP "lcd_data_2" SITE "134"; # OUT (shared with j1_3)
LOCATE COMP "lcd_data_3" SITE "131"; # OUT (shared with j1_2)
LOCATE COMP "lcd_rs" SITE "143"; # OUT (shared with j1_13)
LOCATE COMP "lcd_e" SITE "7"; # OUT (shared with j1_9)

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

```

```
# All I/O ports are tied to 3.3 V
IOBUF ALLPORTS IO_TYPE=LVC MOS33 PULLMODE=NONE;

# DIP switches and pushbuttons need a pulldown resistor, overriding defaults
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;

# Dampen noise on dangling MicroSD input pin, overriding defaults
IOBUF PORT "sdcard_so" PULLMODE=UP;

# Phone jack signals may drain a significant amount of current
IOBUF PORT "p_ring" DRIVE=20 PULLMODE=NONE;
IOBUF PORT "p_tip_0" DRIVE=20 PULLMODE=NONE;
IOBUF PORT "p_tip_1" DRIVE=20 PULLMODE=NONE;
IOBUF PORT "p_tip_2" DRIVE=20 PULLMODE=NONE;
IOBUF PORT "p_tip_3" DRIVE=20 PULLMODE=NONE;
```