

PLIS programavimas

Turinys

1	Tikslas	2
2	Teorija	2
3	Laboratoriniai stendai	4
3.1	Lattice Brevia stendas	5
3.2	Lattice MachXO2 stendas	9

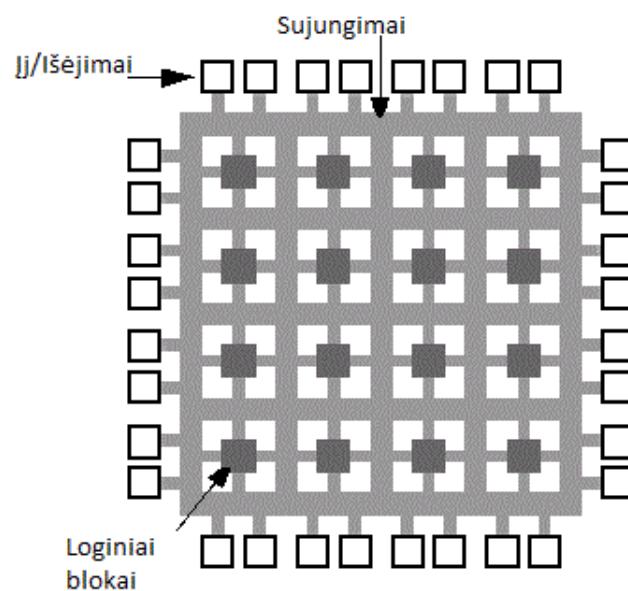
1. Tikslas

Suprasti programuoojamos logikos schemų veikimo principus ir programavimą naudojant programinę įrangą *Lattice Diamond*. Susipažinti su laboratoriniuose darbuose naudojamu PLIS matricos stendu.

2. Teorija

Programuoojamoji loginė integrinė schema (PLIS) (angl. *Field Programmable Gate Array – FPGA*) yra vartotojų elektriškai pagaminta programuoojama integrinė schema. Konfigūruojama dažniausiai aparatūros aprašymo kalba (angl. *Hardware Description language — HDL*), pvz., *VHDL* ar *Verilog* pagalba.

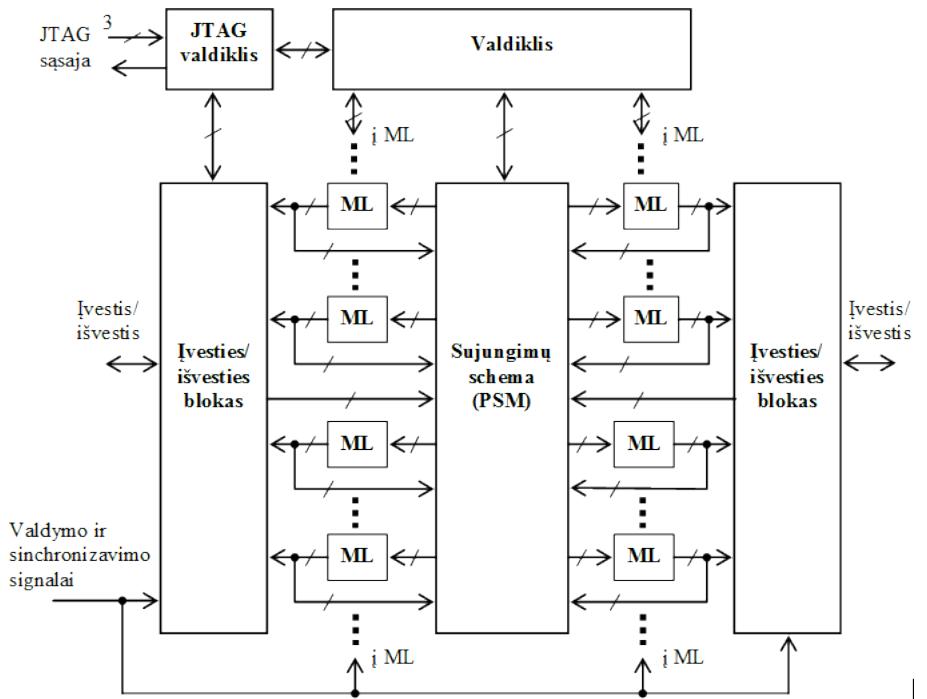
PLIS susideda iš programuoojamos logikos komponentų, vadinamų loginiais blokais arba makrolašteliemis (ML), ir perkonfigūruojamų sujungimų, leidžiančių sujungti ML į reikiama schemą. PLIS struktūra pavaizduota 1 pav. ML gali realizuoti funkcijų (elementarių AND, XOR ir kt., taip pat atminties elementų – trigerių, RAM blokų) rinkinį, sudarantį biblioteką. Ši biblioteka naudojama projektuojant skaitmenines schemas.



1 pav. PLIS matricos struktūra

Pagrindinis PLIS pranašumas – jos konfigūravimo galimybė, leidžianti greitai realizuoti ir modifikuoti aparatinės schemas, išbandyti schemą prieš lusto gamybą ir taip sumažinti

projektavimo kaštus. Dėl šios priežasties ir nuolat mažėjančios kainos šios matricos, jos naudojamos telekomunikacijos, automatikos, medicinos, karinės pramonės srityse ir kosmoso aparatuose. PLIS architektūra susideda iš programuojamosios sujungimo matricos (PSM) (angl. *Programmable Interconnect Array – PIA*), aibės ML ir įvesties/išvesties blokų. PLIS architektūra parodyta 2 pav.



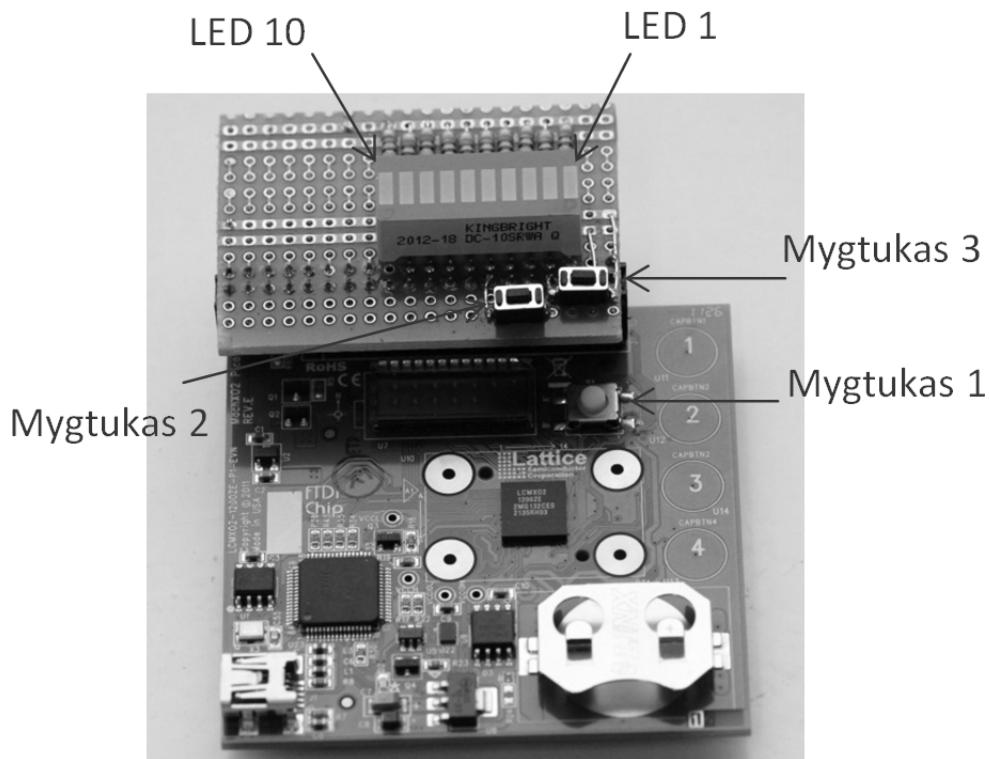
2 pav. PLIS matricos architektūra

PSM realizuota kaip viena matrica, todėl joje būna daug nepanaudotų sujungimų. PLIS matricoje visi tarp ML esantys ryšiai yra identiški. Tai leidžia nustatyti ryšių vėlinimą ir palengvina didelės spartos schemų projektavimą. Tipinė PSM leidžia sujungti bet kurios ML išvestis su kitų ML įvestimis. Kiekviena įvestis gali būti prijungta prie programuoamojo ryšio taško. Kiekvienos ML viduje gali būti savo (lokali) komutavimo sistema, panaši į globaliąją. Beveik visose PLIS naudojama testavimo sistema JTAG. Ši sistema naudoja luste integruotą JTAG valdiklį. Testuojama naudojant standartą IEEE STD 1149.1. Testavimo sistema šiame leidinyje detaliau nenagrinėjama.

PLIS matricos programavimas reikalauja šiokių tokų schemas pakeitimų, norint ją prietaikyti realiam naudojimui, taip pat „apribojimų“ sukūrimo – schemas įejimų ir išėjimų priskyrimo realioms FPGA įvestims ir išvestims, (be to, laikinių apribojimų kūrimo sudėtingesniems projektams, kas šiame leidinyje detaliau nenagrinėjama).

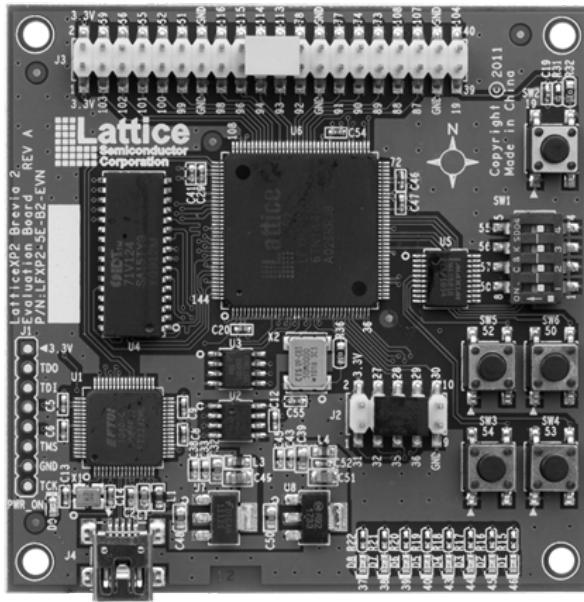
3. Laboratoriniai stendai

Kad studentai praktiškai įvaldytų loginių schemų projektavimą, naudojantis PLIS ir jų programavimu, sukurti laboratoriniai stendai. Juos sudaro firmos *Lattice semiconductor* PLIS su standartinėmis plokštėmis, turinčiomis šviesos diodus (angl. *light emitting diode – LED*), jungiklius bei mygtukus. Dvieju tipų stendai pavaizduoti 3 ir 4 pav.



3 pav. Lattice MachXO2 laboratorinio stendo vaizdas

3.1 Lattice Brevia stendas



4 pav. Lattice Brevia laboratorinio stendo vaizdas

3.1 Lattice Brevia stendas

Prieš programuojant ši stendą, būtina nurodyti renginio tipą *LatticeXP2-5E-6TN144C*. Tam iš pagrindinio meniu pasirenkame punktą **Project → Device** ir atsivérusimė lange nurodome šiuos nustatymus:

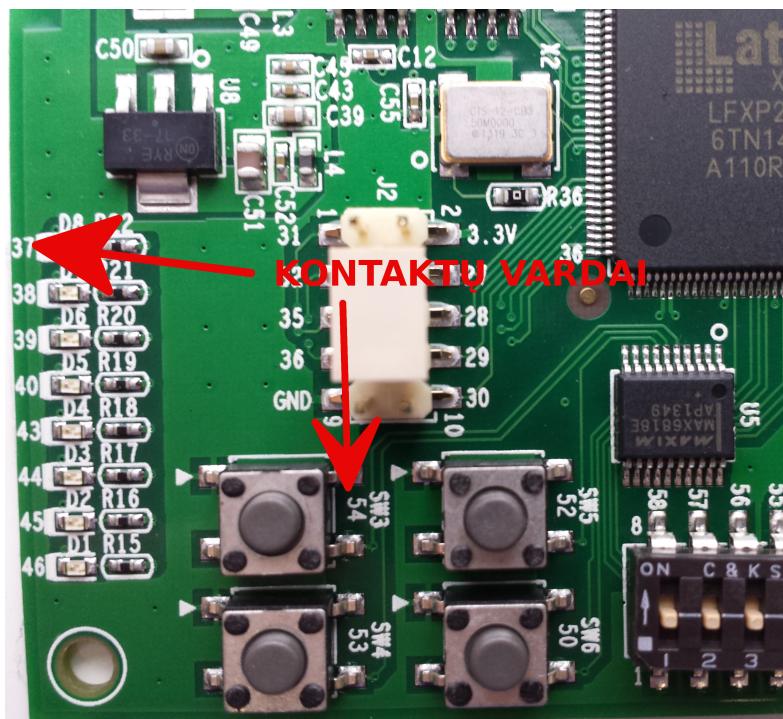
- **Family** – *LatticeXP2*,
- **Device** – *LFXP2-5E*,
- **Package type** – *TQFP144*,
- **Operating conditions** – *Commercial*.
- **Speed Grade** – *6*.

PLIS matrica turi įvesties/išvesties kontaktus, žymimus atitinkamais vardais (numeriais), kuriems priskiriamos schemas įvestys ir išvestys. Atliekant priskyrimą, nurodoma, su kuriuo kontaktu bus susietos schemas įvestys ir išvestys. Norint matyti veikiančią schemą, jos išejimams priskiriami kontaktų, prie kurių prijungti LED diodai, vardai. Norint

3.1 Lattice Brevia stendas

valdyti loginę schemą, jos įvestims priskiriami kontaktų, prie kurių prijungti mygtukai, vardai.

Stende esančių LED diodų ir mygtukų kontaktų vardai pavaizduoti prie antitinkamų jungiklių, mygtukų bei šviesos diodų (5 pav.)



5 pav. Lattice Brevia matricos kontaktų vardai, pažymėti prie atitinkamų įrenginių plokštėje

3.1 Lattice Brevia stendas

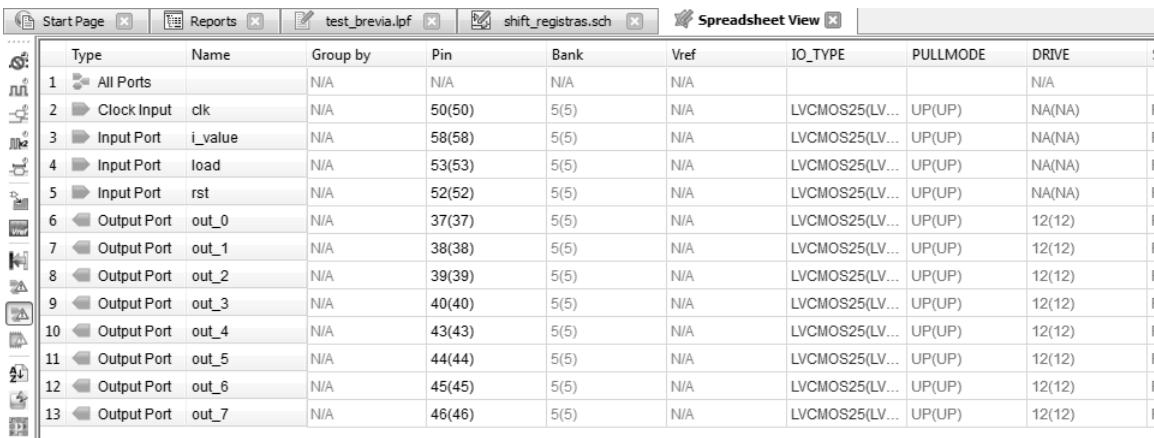
1 lentelė. Brevia stendo kontaktų vardai

Įrenginys	Prievado pavadinimas (Pin name)	Papildomos nuostatos
Mygtukas 1	50	PULLMODE - UP
Mygtukas 2	52	PULLMODE - UP
Mygtukas 3	53	PULLMODE - UP
Mygtukas 4	54	PULLMODE - UP
LED 1	37	PULLMODE - UP
LED 2	38	PULLMODE - UP
LED 3	39	PULLMODE - UP
LED 4	40	PULLMODE - UP
LED 5	43	PULLMODE - UP
LED 6	44	PULLMODE - UP
LED 7	45	PULLMODE - UP
LED 8	46	PULLMODE - UP
Jungiklis 1	55	
Jungiklis 2	56	
Jungiklis 3	57	
Jungiklis 4	58	

Fiziniams kontaktams priskirti naudojamas įrankis **Spreadsheet View**, iškviečiamas pagrindiniame meniu pasirinkus punktą **Tools→Spreadsheet View**. Šio įrankio langas paravizduotas 6 pav. Reikiameiems kontaktams, prijungtiems prie mygtukų bei LED diodų, būtina nustatyti parametrą **PULLMODE** į **UP**, tokiu būdu šie mygtukai generuos žemą loginį lygi paspaudus mygtuką.

Išsaugojus pakeitimius lange **Spreadsheet View**, automatiškai sukuriamas projekto apribojimų (constraints) *.lpf failas.

3.1 Lattice Brevia stendas



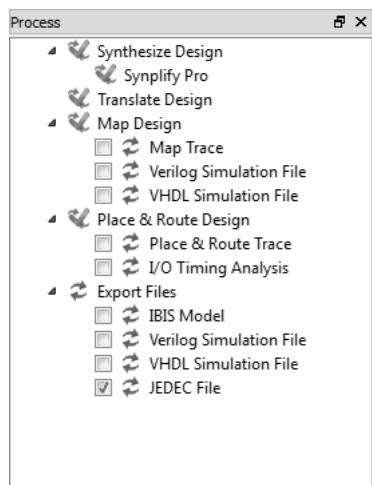
Type	Name	Group by	Pin	Bank	Vref	IO_TYPE	PULLMODE	DRIVE
1	All Ports	N/A	N/A	N/A	N/A			N/A
2	Clock Input	clk	N/A	50(50)	N/A	LVCMS25(LV...)	UP(UP)	NA(NA)
3	Input Port	i_value	N/A	58(58)	5(5)	N/A	LVCMS25(LV...)	UP(UP)
4	Input Port	load	N/A	53(53)	5(5)	N/A	LVCMS25(LV...)	UP(UP)
5	Input Port	rst	N/A	52(52)	5(5)	N/A	LVCMS25(LV...)	UP(UP)
6	Output Port	out_0	N/A	37(37)	5(5)	N/A	LVCMS25(LV...)	UP(UP)
7	Output Port	out_1	N/A	38(38)	5(5)	N/A	LVCMS25(LV...)	12(12)
8	Output Port	out_2	N/A	39(39)	5(5)	N/A	LVCMS25(LV...)	12(12)
9	Output Port	out_3	N/A	40(40)	5(5)	N/A	LVCMS25(LV...)	12(12)
10	Output Port	out_4	N/A	43(43)	5(5)	N/A	LVCMS25(LV...)	12(12)
11	Output Port	out_5	N/A	44(44)	5(5)	N/A	LVCMS25(LV...)	12(12)
12	Output Port	out_6	N/A	45(45)	5(5)	N/A	LVCMS25(LV...)	12(12)
13	Output Port	out_7	N/A	46(46)	5(5)	N/A	LVCMS25(LV...)	12(12)

6 pav. *Spreadsheet View* langas

Atliekant pakeitimius schemaeje šis failas automatiškai neatnaujinamas, todėl jei pakeičiamas įvesčių ar išvesčių skaičius ar pavadinimus, būtina nereikalingas eilutes iš šio failo ištrinti ranka (prievaldų priskyrimą reikės atlikti tik naujoms ar pervadintoms įvestims/išvestims) arba ištrinti visą failą (prievaldų priskyrimą **Spreadsheet view** tokiu atveju reikės atlikti iš naujo).

Schemas fizinis išdėstymas (planavimas) plokštėje atliekamas meniu punktu **Place and Route**. Atliekame žingsnį **Place and Route** iš meniu **Process**.

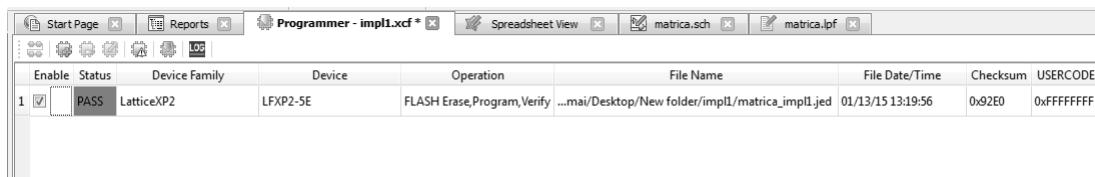
Po žingsnio **Place and Route** būtina generuoti FPGA programavimo failą. Tam varnele pažymimas lange **Process**, skiltyje **Export Files** esantis punktas **JEDEC File**, kaip pavaizduota 7 pav.



7 pav. *JEDEC* pasirinkimas

3.2 Lattice MachXO2 stendas

Pasirinkus **JEDEC** įvykdomas punktas **Export Files**. Projekte turėtų atsirasti *projekto_vardas.jed* failas. Šis failas ir naudojamas matricai programuoti. Tam tikslui matricą prijungiamo prie USB prievedo ir pagrindiniame meniu pasirenkame **Tools→Programmer**. Atsidariusiame programuotuvo lange patikriname, ar **Device Family** bei **Device** atitinka turimą matricą (šiame laboratoriiniame stende – tai *LatticeXP2-5E-6TN144C*, jei neatitinka, projekto kūrimo metu pasirinkta ne ta biblioteka), be to, įsitikiname, kad skiltyje **File Name** nurodytas reikiamas (šio projekto) **.jed* failas. Programuotuvo langas pavaizduotas 8 pav.



8 pav. Lattice Brevia programavimo langas

Suprogramuoti matricai naudojame pagrindinio meniu komandą **Design→Program** arba mygtuką **Program** programuotuvo lange.

Detalesni FPGA matricos programavimo pavyzdžiai pateikiami atitinkamų laboratoriinių darbų aprašymų medžiagoje.

3.2 Lattice MachXO2 stendas

Prieš programuojant šį stendą, būtina nurodyti įrenginio tipą *MachXO2-1200ZE-2MG132C*. Tam iš pagrindinio meniu pasirenkame punktą **Project → Device** ir atsivérusiamė lange nurodome šiuos nustatymus:

- **Family** – *MachXO2*,
- **Device** – *LCMXO2-1200ZE*,
- **Package type** – *CSBGA132*,
- **Performance Grade** – *2*,
- **Operating conditions** – *Commercial*.

PLIS matrica turi įvesties/išvesties kontaktus, kuriems priskiriamos schemas įvestys ir išvestys. Kiekvienas matricos kontaktas žymimas atitinkamu kontakto vardu (raide ir

3.2 Lattice MachXO2 stendas

numeriu). Atliekant priskyrimą, nurodoma, su kuriuo kontaktu bus susietos schemas įvestys ir išvestys. Norint matyti veikiančią schemą, jos išėjimams priskiriami kontaktų, prie kurių prijungti LED diodai, vardai. Norint valdyti loginę schemą, jos įvestims priskiriami kontaktų, prie kurių prijungti mygtukai, vardai.

Stende esančių LED diodų ir mygtukų kontaktų vardai pateikiami **2** lentelėje. Čia pateikiamas ir papildomas nuostatos, reikalingos, kad mygtukai veiktų tinkamai.

2 lentelė. Stendo kontaktų vardai

Irenginys Mygtukas 1 (pagrindini nėje FPGA plokštėje)	Prievado pavadinimas (Pin name) N3	Papildomos nuostatos PULLMODE - UP
Mygtukas 2 (kairysis)	B1	
Mygtukas 3 (dešinysis)	C3	PULLMODE - UP
LED_1 (dešinysis)	P13	
LED_2	M4	
LED_3	P3	
LED_4	N12	
LED_5	C10	
LED_6	A13	
LED_7	B13	
LED_8	C8	
LED_9	B8	
LED_10 (kairysis)	D1	

Fiziniam kontaktams priskirti naudojamas įrankis *Spreadsheet View*, iškviečiamas pagrindiniame meniu pasirinkus punktą **Tools→Spreadsheet View**. Šio įrankio langas paraižduotas **9** pav. Kontaktams B1 ir C3, prijungtiems prie mygtukų, būtina nustatyti parametrą **PULLMODE** į parametrą **UP**, kitaip šie mygtukai generuos aukštą loginį lygi. Paspaudus mygtuką, lygis taps žemas.

Išsaugojus pakeitimų lange **Spreadsheet View**, automatiškai sukuriamas projekto apribojimų (constraints) *.lpf failas.

3.2 Lattice MachXO2 stendas

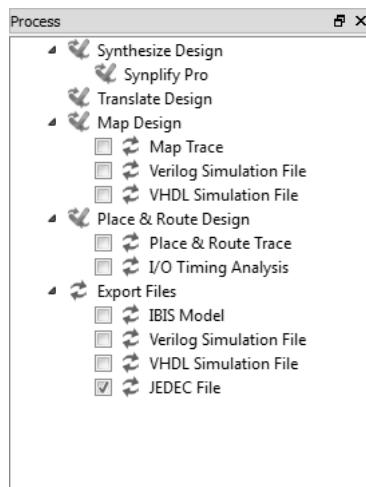
Type	Name	Group by	Pin	Bank	Bank_VCCIO	Vref	IO_TYPE	PULLMODE
1	All Ports	N/A	N/A	N/A	N/A	N/A		
2	Input Port R	N/A			N/A	N/A	LVCMS25	UP
3	Clock Input C	N/A			N/A	N/A	LVCMS25	DOWN
4	Input Port PR2	N/A			N/A	N/A	LVCMS25	DOWN
5	Input Port PR1	N/A			N/A	N/A	LVCMS25	DOWN
6	Input Port PR0	N/A			N/A	N/A	LVCMS25	DOWN
7	Output Port C_out	N/A			N/A	N/A	LVCMS25	DOWN

9 pav. *Spreadsheet View* langas

Atliekant pakeitimius schemaeje šis failas automatiškai neatnaujinamas, todėl jei pakeičiamas įvesčių ar išvesčių skaičius ar pavadinimus, būtina nereikalingas eilutes iš šio failo ištrinti ranka (prievaldų priskyrimą reikės atlikti tik naujoms ar pervadintoms įvestims/išvestims) arba ištrinti visą failą (prievaldų priskyrimą **Spreadsheet view** tokiu atveju reikės atlikti iš naujo).

Schemas fizinis išdėstymas (planavimas) plokštėje atliekamas meniu punktu **Place and Route**. Atliekame žingsnį **Place and Route** iš meniu **Process**.

Po žingsnio **Place and Route** būtina generuoti FPGA programavimo failą. Tam varnele pažymimas lange **Process**, skiltyje **Export Files** esantis punktas **JEDEC File**, kaip pavaizduota 10 pav.

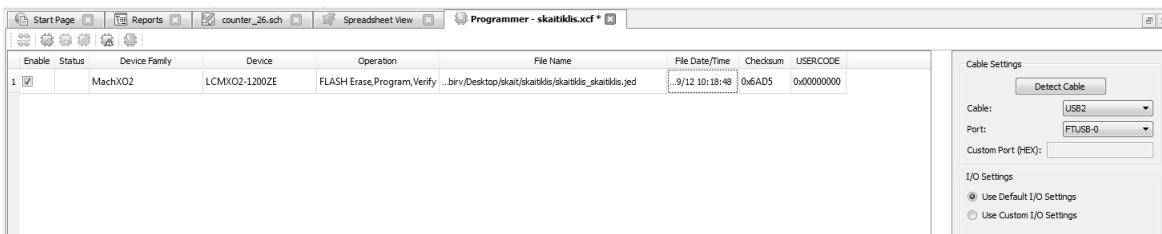


10 pav. *JEDEC* pasirinkimas

Pasirinkus **JEDEC** įvykdomas punktas **Export Files**. Projekte turėtų atsirasti *projekto_vardas.jed* failas. Šis failas ir naudojamas matricai programuoti. Tam tikslui matricą prijungiamo prie USB prievaldo ir pagrindiniame meniu pasirenkame **Tools→Programmer**. Atsidariusiame programuotuvu lange patikriname, ar **Device Family** bei **Device**

3.2 Lattice MachXO2 stendas

atitinka turimą matricą (šiame laboratoriniame stende – tai *MachXO2-LCMXO2-1200ZE*, jei neatitinka, projekto kūrimo metu pasirinkta ne ta biblioteka), be to, įsitikiname, kad skiltyje **File Name** nurodytas reikiamas (šio projekto) *.jed failas. Programuotuvo langas pavaizduotas 11 pav.



11 pav. Lattice MachXO2 programavimo langas

Suprogramuoti matricai naudojame pagrindinio meniu komandą **Design→Program** arba mygtuką **Program** programuotuvo lange.

Detalesni FPGA matricos programavimo pavyzdžiai pateikiami atitinkamų laboratorių darbų aprašymų medžiagoje.