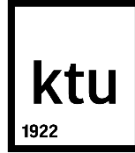


Pagal teisingumo lentelę sumodeliuojame šį registrą Lattice Diamond platformoje. Esminiai skirtumai apjungimai tarp multiplekserių ir Vhigh elemento panaudojimas, vietoj D6-D0 įvedimų.



Kauno technologijos universitetas

Informatikos fakultetas

Skaitmeninės logikos pradmenys P175B100

Trečiojo laboratorinio darbo ataskaita

Studentas

Lukas Kuzmickas

Dėstytoja

Rasa Brūzgienė

Turinys

1. Įvadas	3
1.1 Darbo tikslas	3
1.2 Darbo užduotis	3
2. Pagrindinė dalis	4
2.1 Teorija	4
2.2 Užduoties atlikimo tvarka	7
2.2.1. Universalus registras	7
2.2.2. Specializuotas registras	10
2.2.3. PLIS matricos kūrimas	11
3. Išvados	15

1. Įvadas

1.1 Darbo tikslas

Sumodeliuoti universalų registrą galinti atlikti loginius postūmius (LL1, LR1) bei specializuotą registrą galinti atlikti CL,CR, AR, AL postūmius pagal tam tikrą duotą skaičiaus kodą. Suprasti skirtingų registrų veikimo principą, taikymo galimybes. Išmokti sudarinėti registrus su įvairiomis postūmio operacijomis. Sudaryti tam tikrą registro trigerių darbo schemą.

1.2 Darbo užduotis

121	7	LL2, CL2, AR1	1	Asinchroninis	Papildomas
-----	---	---------------	---	---------------	------------

1 pav. Individuali užduotis

Pagal individualią trečiojo laboratorinio darbo užduotį, mums reikia sumodeliuoti universalų registrą galinti atlikti 2 loginius postūmius – į kairę ir dešinę. Šis registras yra 7 bitų bei asinchroninis. Tuo pačiu pagal užduotį reikia sumodeliuoti specializuotą registrą, kuris gali atlikti loginį postūmį LL2, ciklinį postūmį CL2 ir aritmetinį postūmį AR1. Specializuotas registras privalo būti asinchroninis ir turi būti realizuotas papildomuoju kodu. Mano užduotyje atliekant postūmius buvo prašoma į bitus įrašyti vienetus, visoms operacijoms.

2. Pagrindinė dalis

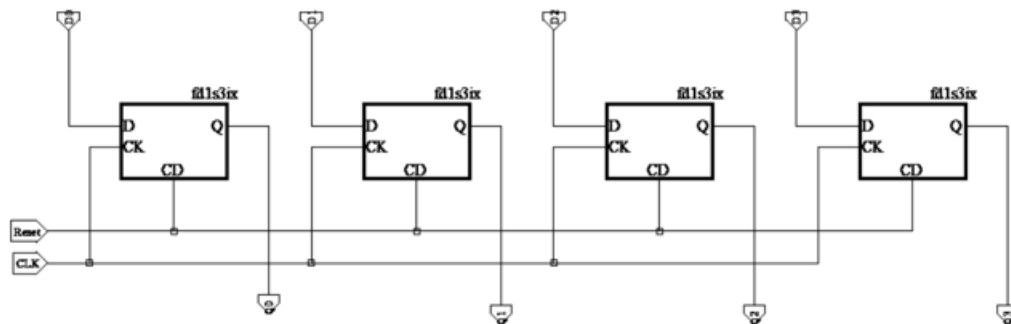
2.1 Teorija

Registrai – atminties elementas, galintis išsaugoti daugiau nei vieną bitą informacijos. Įtaisas skirtas informacijai įrašyti, saugoti, skaityti. Registrai pagal savo atliekamas operacijas gali būti : saugojimo, postūmio, universalūs.

Registrai skirstomi į tam tikras kategorijas:

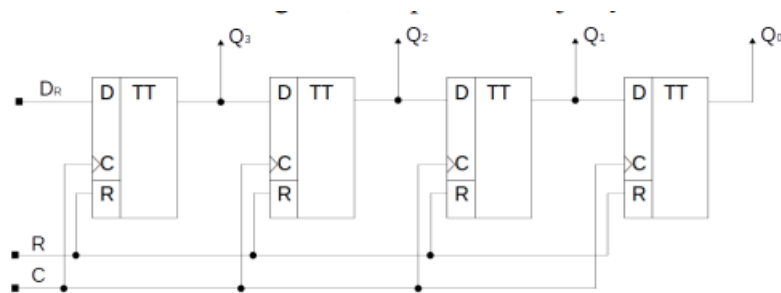
- Saugoti tam tikrą informaciją (Lygiagretusis).
- Stumti informaciją (Nuoseklusis).
- Atlikti įvairius veiksmus (Universalusis).

Lygiagretusis registras – registras sudarytas iš sinchroninių D trigerių su įvestimi ir išvestimi.



2 pav. Lygiagretusis registras

Postūmio registras – registras su postūmio funkcija, šis registras informaciją stumia į dešinę arba kairę (LL ir LR operacijos). Šią informaciją stumiant, sinchronizacijos signalo taktu informacija nuosekliai įvedama stumiant bitus link išėjimo.



3 pav. Registro su postūmio funkcija schema

Postūmiai – šios operacijos reikalingos kompiuteriui atliekant daugyba ir dalyba. Postūmiai skirstomi į loginius – kai kiekviena skiltis perstumiami į gretimos skilties vietą, kairėn arba dešinėn priklausant nuo postūmio tipo. Ciklinis postūmis – į laisvasias skiltis įrašoma informacija iš išstumtų skilčių bei aritmetinis postūmis – atliekamas aukščiausios skilties nestumiant tuo išsaugant skaičiaus ženklą, atlikdami aritmetinį postūmį, į laisvasias skiltis įrašome 0 arba 1, priklausomai nuo skaičiaus kodo ir ženklo.

Postūmio tipai:

- Loginis postūmis.
- Ciklinis postūmis.
- Aritmetinis postūmis.

Neigiami skaičiai dvejetainiame kode išsaugomi kitaip, prie neigiamų skaičių visada dešimtainėje sistemoje visada pridedamas minuso ženklas, o registruose informacija išsaugoma bitais, todėl nėra galimybės pridėti minuso ženklo. Minuso ženklas privalo būti saugomas, kaip bito skaičius.

Pagrindiniai skaičių saugojimo metodai:

- Tiesioginis.
- Atvirkštinis.
- Papildomasis.

Tiesioginis kodas – šiame kode pirmasis bitas nusako, koks yra skaičiaus ženklas. Jei 0 – tai pliusas, jei 1 – tai neigiamas.

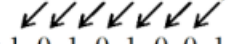
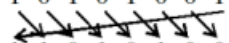
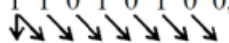
Atvirkštinis kodas – šiame kode visas teigiamo skaičiaus kodas invertuojamas ir taip gaunamas atvirkštinis neigiamas skaičius.

Papildomasis kodas – šiame kode nuliui išreikšti naudojama tik viena bitų kombinacija. Iš teigiamo į neigiamą skaičių šiame kode verčiami:

- Invertuojant visą šio skaičiaus bitų rinkinį.
- Pridedame (+1) prie invertuotos reikšmės.

Ženklas (skaičius)	Kodas	Į laisvą skiltį įrašoma	
		AR stumiant į dešinę	AL stumiant į kairę
+ teigiamas	bet koks	0	0
	tiesioginis	0	0
- neigiamas	atvirkštinis	1	1
	papildomas	1	0

Pavyzdys. Turime

- a) Atlikus postūmį LL1, 1 R = 1 1 0 1 0 1 0 0;

R = 1 0 1 0 1 0 0 1 ← DL;
- b) Atlikus postūmį CR1 R = 1 1 0 1 0 1 0 0;

R = 1 1 0 1 0 1 0 0;
- c) Atlikus postūmį AR1 atv. kode R = 1 1 1 0 1 0 1 0.

R = 1 1 1 0 1 0 1 0.

4 pav. Postūmių teisingumo lentelė ir pavyzdžiai

Universalusis registras – šis registras dažniausiai vykdo dvi paprastas postūmio operacijas, loginį postūmį į dešinę LR (LogicRight) arba loginį postūmį į kairę LL (LogicLeft). Šis registras vadinamas universaliuoju, nes gali realizuoti ir kitas operacijas.

2.2 Užduoties atlikimo tvarka

Pagal užduotį reikia sumodeliuoti universalų, specializuotą registrą, PLIS matricos realizaciją.

121	7	LL2, CL2, AR1	1	Asinchroninis	Papildomas
-----	---	---------------	---	---------------	------------

5 pav. Individuali užduotis

Universalus ir specializuotas registras bus sudarytas iš 7 bitų (7 skilčių) ir jie bus asinchroniniai. Specializuotas registras papildomai realizuos LL2 (Logic Left 2), CL2 postūmis kairėn pusėn, AR1 (Aritmetinis postūmis dešinėn) operacijas. Kodas – papildomasis. Į atsilaisvinusias registro skiltis rašysime 1.

2.2.1. Universalus registras

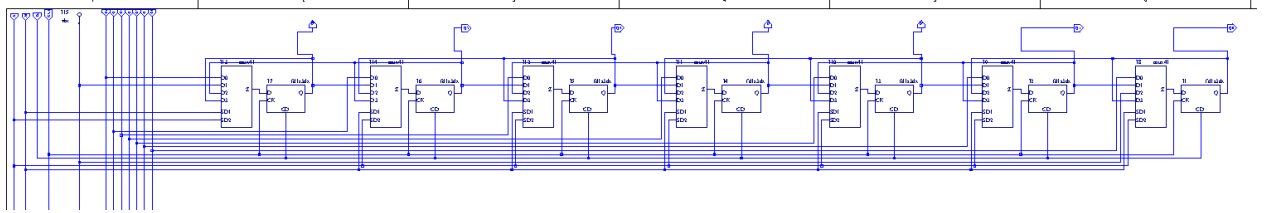
Pradžioje sudarome universalaus registro teisingumo lentelę, su 4 mikrooperacijomis, kuriuos valdysime jungdami tam tikras operacijas prie multiplekserio.

Universalus 7 bitų registras										
R	A0	A1	Q6	Q5	Q4	Q3	Q2	Q1	Q0	Operacija
1	X	X	0	0	0	0	0	0	0	Reset
0	0	0	D6	D5	D4	D3	D2	D1	D0	Lygiagretus įrašymas
0	1	0	DR	Q6	Q5	Q4	Q3	Q2	Q1	Loginis > LR1
0	0	1	Q5	Q4	Q3	Q2	Q1	Q0	DL	Loginis < LL1
0	1	1	Q6	Q5	Q4	Q3	Q2	Q1	Q0	Save

6 pav. Universalaus registro teisingumo lentelė

Iš pradžių sudarome Reset signalą, kai visos išvestys bus nunulinamos, vėliau sudarome kombinacijas 4 mikrooperacijoms, įrašymui, loginiam postūmiui dešinėn, loginiam postūmiui kairėn, informacijos išsaugojimui.

Šį registrą sumodeliuojame Lattice Diamond aplinkoje. Naudodami D trigerius ir multiplekserius.



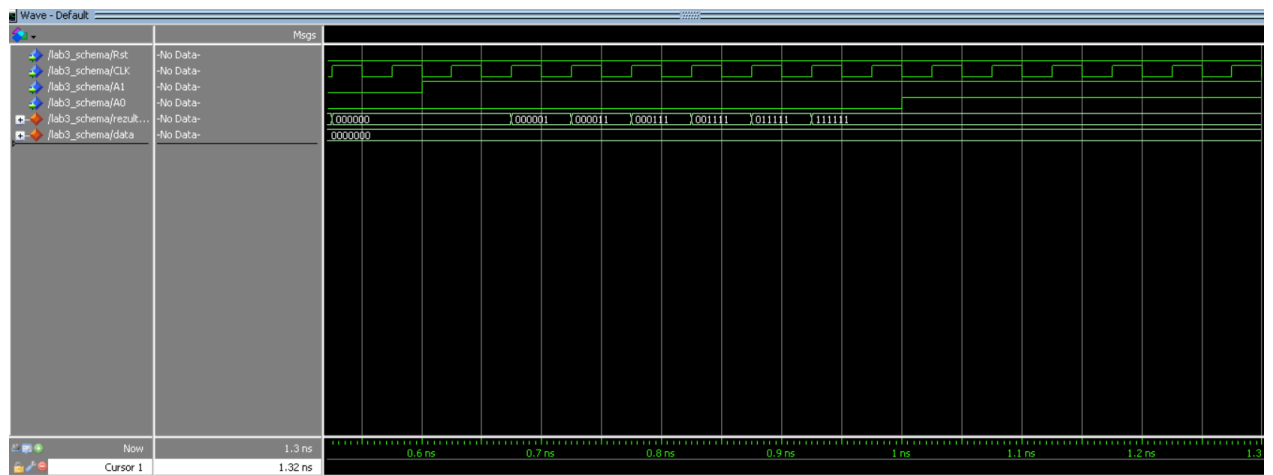
7 pav. Universalaus registro schema

Patikriname universalaus registro veikimą, pradžioje susidarome šiam registrui reikalingas direktyvas.

```
// universalus
restart -f
force -freeze sim:/lab3_schema/Rst 1 0, 0 {10 ps}
force -freeze sim:/lab3_schema/Clk 0 0, 1 {25 ps} -r 50
force -freeze sim:/lab3_schema/A0 0 0, 1 {100 ps}
force -freeze sim:/lab3_schema/A0 0 500, 1 {1000 ps}
force -freeze sim:/lab3_schema/A1 0 0, 1 {600 ps}
force -freeze sim:/lab3_schema/D6 0 0
force -freeze sim:/lab3_schema/D5 0 0
force -freeze sim:/lab3_schema/D4 0 0
force -freeze sim:/lab3_schema/D3 0 0
force -freeze sim:/lab3_schema/D2 0 0
force -freeze sim:/lab3_schema/D1 0 0
force -freeze sim:/lab3_schema/D0 0 0
run 1300
```

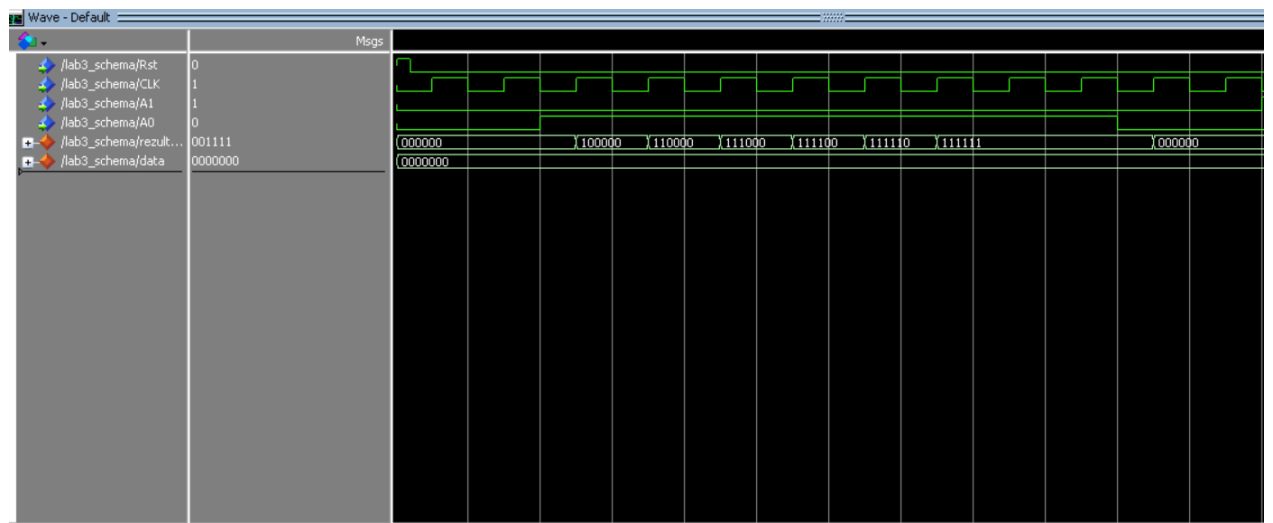
8 pav. Universalaus registro direktyvos

Iš simuliacijos matome, kad saugojimo, įrašymo, LL1, LR1 operacijos sėkmingai realizuojamos.



9 pav. Universalaus registro postūmio į kairę operacija LL1 (LogicLeft1), Save operacija

Iš simuliacijos matome, kaip skaičius sėkmingai stumiamas kairėn pusėn, įrašomas vienetas, atliekama loginė postūmio operacija LL1, po to tas skaičius išsaugomas Save operacijos realizacija.



10 pav. Universalaus registro postūmio į dešinę operacija LR1 (LogicRight1)

Dar kartą sėkmingai matome, kaip realizuojama loginio postūmio operacija LR1, atliekamas loginis postūmis dešinėn, po to tas skaičius nuresetinamas – Reset operacijos realizavimas.

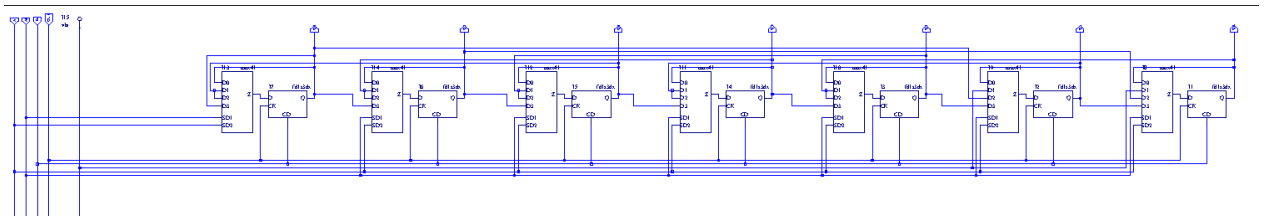
2.2.2. Specializuotas registras

Antrajam specializuotam registrui sudarome teisingumo lentelę, saugojimo, LL2, CL2, AR1 mikrooperacijoms.

Specializuotas 7 bitų registras										
R	A0	A1	Q6	Q5	Q4	Q3	Q2	Q1	Q0	Operacija
1	X	X	0	0	0	0	0	0	0	Reset
0	0	0	Q6	Q5	Q4	Q3	Q2	Q1	Q0	Save
0	1	0	Q4	Q3	Q2	Q1	Q0	1	1	LL2
0	0	1	Q4	Q3	Q2	Q1	Q0	Q6	Q5	CL2
0	1	1	Q6	Q6	Q5	Q4	Q3	Q2	Q1	AR1

11 pav. Specializuoto registro teisingumo lentelė

Pagal teisingumo lentelę sumodeliuojame šį registrą Lattice Diamond platformoje. Esminiai skirtumai apjungimai tarp multiplekserių ir Vhigh elemento panaudojimas, vietoj D6-D0 įvedimų.



12 pav. Specializuoto registro schema

Susidarome šio specializuoto registro direktyvas visoms operacijoms

Specializuoto registro simuliacijos rezultatai

Iš simuliacijos pastebime, kad išsaugojimo, LL2, CL2, AR1 operacijos atliekamos teisingai.

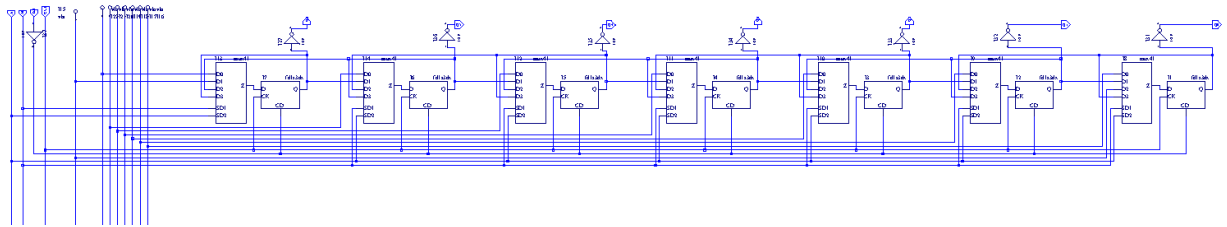
2.2.3. PLIS matricos kūrimas

PLIS matricos realizavimui reikia pasirinkti matricos kontaktų kodus, modifikuoti schemas (invertuojant, pridėdant VHigh, VLow elementus).

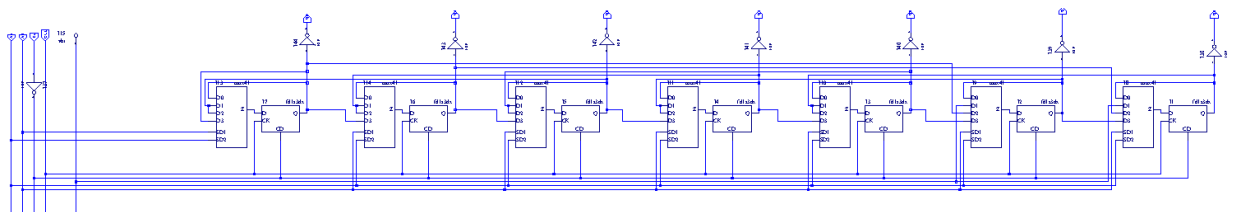
- Pertvarkyti išvedimus, juos invertuoti.
- Invertuoti Reset signalą.
- 0000000 kombinacija, kad būtų įmanoma pastebėti vieneto įrašymo postūmius.

Matricos modeliavimui buvo pasirinkta Lattice Brevia stendas. Buvo pertvarkytos dvi registro schemos, šiai matricai sudaryti.

Pagal šiuos reikalavimus buvo pertvarkytos dvi schemos, kad juos būtų realizuotos Lattice Brevia stendui.



13 pav. Universalus registras pertvarkytas PLIS matricos realizacijai



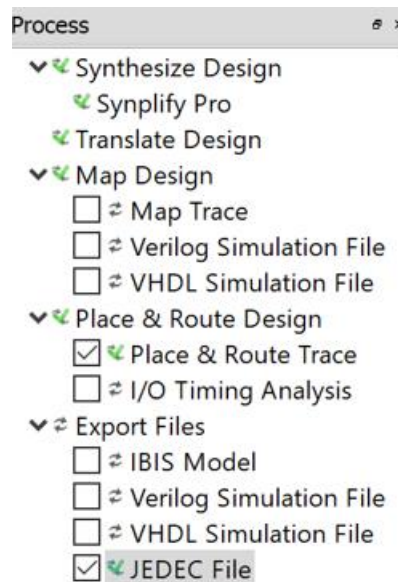
14 pav. Specializuotas registras pertvarkytas PLIS matricos realizacijai

Toliau reikia pritaikyti šių schemų įvedimus ir išvedimus tam tikriems Lattice Brevia stendo kontaktams, pasinaudodami Tools => Spreadsheet view įrankiais. Šiame lape atitinkamai priskiriame kiekvienam įvedimui ir išvedimui stendo kontaktų numerius. Visus įvedimus ir išvedimus, išskyrus Reset signalą nustatome į PULLMODE – UP, Reset – DOWN, nes tai asinchroninė schema.

▼ All Ports	N/A	N/A	N/A			N/A				N/A
▼ Input	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
A0	55	5	N/A	LVC MOS25	UP	NA	FAST	OFF	OFF	N/A
A1	56	5	N/A	LVC MOS25	UP	NA	FAST	OFF	OFF	N/A
▼ Clock	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
CLK	52	6	N/A	LVC MOS25	UP	NA	FAST	OFF	OFF	N/A
Rst	▲ 57		N/A	LVC MOS25	DOWN	NA	FAST	OFF	OFF	N/A
▼ Output	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Q0	▲ 37		N/A	LVC MOS25	UP	12	FAST	OFF	OFF	0.000
Q1	▲ 38		N/A	LVC MOS25	UP	12	FAST	OFF	OFF	0.000
Q2	39	6	N/A	LVC MOS25	UP	12	FAST	OFF	OFF	0.000
Q3	40	6	N/A	LVC MOS25	UP	12	FAST	OFF	OFF	0.000
Q4	43	6	N/A	LVC MOS25	UP	12	FAST	OFF	OFF	0.000
Q5	44	6	N/A	LVC MOS25	UP	12	FAST	OFF	OFF	0.000
Q6	45	6	N/A	LVC MOS25	UP	12	FAST	OFF	OFF	0.000

15 pav. Matricos sujungimų modeliavimas

Paruošus šios schemos įvedimo ir išvedimo kontaktų išdėstymą Lattice Brevia stendui, pasinaudodami Place & Route Design ir Export Files – JEDEC File procesais šį mūsų sujungimų realizavimą konvertuojame į .jed* formato failą, kuriuo suprogramuosime matricą.



16 pav. *.jed failo kūrimas

Tada per Tools => Programmer pasirenkame matricos USB sujungimus ir patį schemos sugeneruotą .jed formato failą, matricos realizacijai.

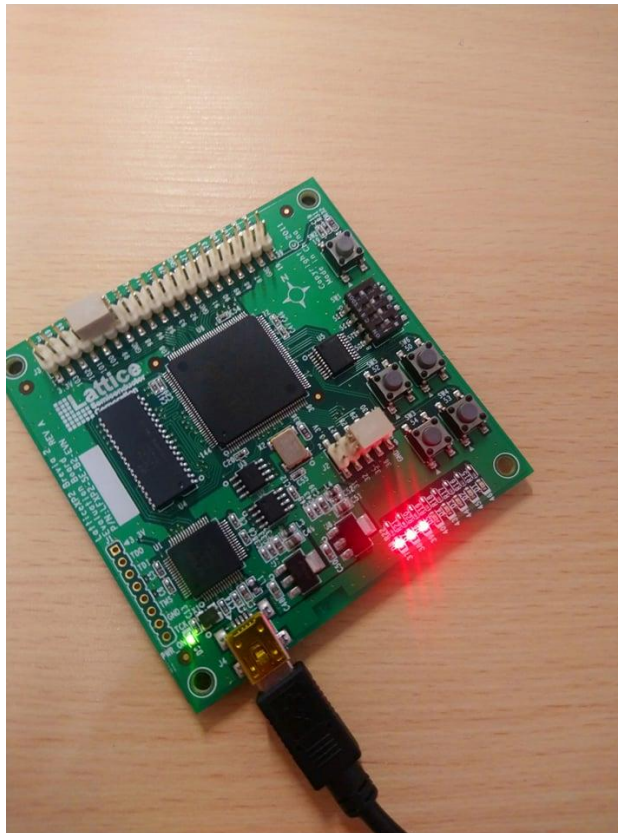
Enable	Status	Device Family	Device	Operation	File Name	File Date/Time	Checksum	USERCODE
1	<input checked="" type="checkbox"/>	LatticeXP2	LFXP2-5E	...am,Verify	...VERSALUS/Registrai/impl1/Registrai_impl1.jed	...13/22 15:07:14	0x7ABB	0xFFFFFFFF

17 pav. Matricos programavimas

Dabar beliko pratestuoti matricą su Design – Program įrankių. Užkrauname matricą ir naudodami teisingumo lentelės kombinacijas pratestuojame visas šios matricos postūmio operacijas.

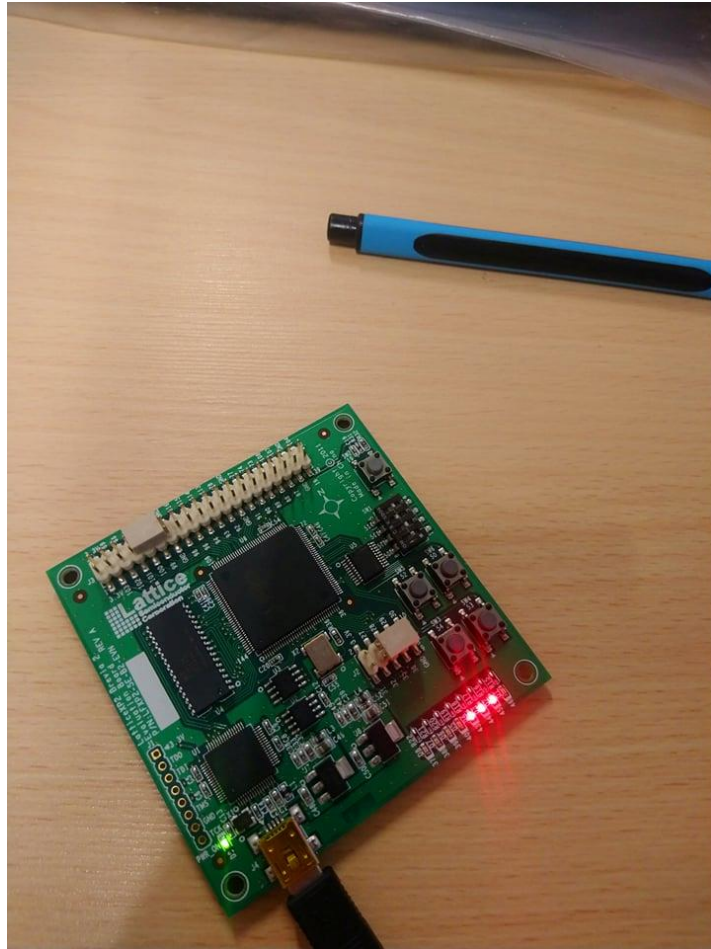
Užsidegęs kontakto LED lemputė indikuoja Vhigh signalą – 1 įrašymas.

Nustatydami tam tikras A0,A1, Reset signalo reikšmes, keisdami Clock signalo reikšmes gauname Universalaus registro LR1 operaciją (Loginio postūmio į dešinę).



18 pav. Universalaus registro PLIS loginis postūmis į dešinę LR

Nustatydami tam tikras A0,A1, Reset signalo reikšmes, keisdami Clock signalo reikšmes gauname Universalaus registro LL1 operaciją (Loginio postūmio į kairę).



19 pav. Universalaus registro PLIS loginis postūmis į kairę LL

3. Išvados

Apibendrinant, kuriant registrus – elementus, kurie gali išsaugoti daugiau negu vieną bitą informacijos, buvo plačiau suprasta pačio registro veikimas, jo pagrindinės struktūros dalys. Plačiau panagrinėtos registro postūmio operacijos – loginės, ciklinės, aritmetinės. Aiškiai suprastas skirtumas tarp šių postūmio operacijų ir jų realizavimas registruose. Realizavimas schemomis, simuliacijomis, PLIS matricos modeliavimu – suteikė realios darbinės patirties su registrais.