



**Kauno technologijos universitetas**  
**Informatikos Fakultetas**

## **3 Laboratorinio Darbo Ataskaita**

P175B100 Skaitmeninės Logikos Pradmenys

---

**Atliko:**

**IFF-1/9 grupės studentas**  
**Nedas Liaudanskis**

---

**Priėmė:**

**Lekt. Jurgita Arnastauskaitė**

---

**Kaunas, 2022**

## **Turinys**

<b>Turinys.....</b>	<b>2</b>
<b>1. Įvadas.....</b>	<b>3</b>
1.1 Tikslas .....	3
1.2 Užduotis .....	3
<b>2. Universalaus registro užduotis .....</b>	<b>4</b>
<b>3. Specializuoto registro realizacija .....</b>	<b>6</b>
<b>4. Universalaus Registro PLIS realizacijos .....</b>	<b>8</b>
<b>5. Išvados .....</b>	<b>13</b>

# 1. Įvadas

## 1.1 Tikslas

Perprasti darbą su postūmio registrais, jų struktūra, veikimo principus, taikymo metodus ir realizavimą naudojant trigerius.

## 1.2 Užduotis

**Užduoties numeris: 130**

**Užduoties registrų sąlygos:**

130	5	LL1, CR2, AL2	0	Sinchroninis	Atvirkštinis
-----	---	---------------	---	--------------	--------------

### Teorija:

- Registrai, tai prietaisai, kurie įrašo informacija ir yra skirti ją saugoti. Registrai dar gali atlikti ir kitas operacijas, tokia kaip postūmius. Registrą sudaro trigeriai ir juos kontroliuojančios schemos. Registrų tipai skirstomi pagal įvedimo ir išvedimo būdą:
  - saugojimo (lygiagretusis);
  - postūmio (nuoseklusis);
  - universalusis.

Registrai taip pat, gali būti vientaktčiai (sudaryti iš dinaminio valdymo trigerių) ir dvitaktčiai, kuriuose informacija įrašoma per du taktus. Visos registro skiltys yra numeruojamos nuo dešinės pusės, pradedant nuo 0. Tokio rašymo pagalba, informaciją yra galima interpretuoti kaip skaičių, kurio reikšmė nustatoma sumuojant atitinkamas reikšmes, gaunamas dvejetą pakėlus indekso laipsniu.

## 2. Universalaus registro užduotis

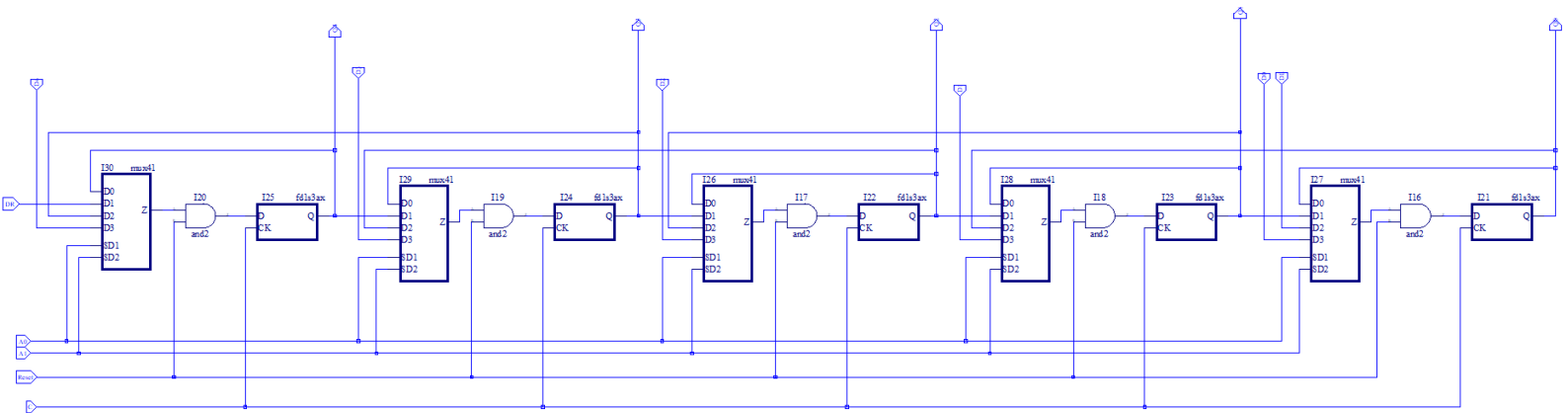
Pirmiausia iš užduoties sąlygos išsiaiškinau, jog mano gautas registras turės 5 trigerius ir gebės saugoti informaciją  $2^4$  dydžiu. Tada pasidariau Universalaus registro veikimo lentelę.

Universalaus registro lentelė

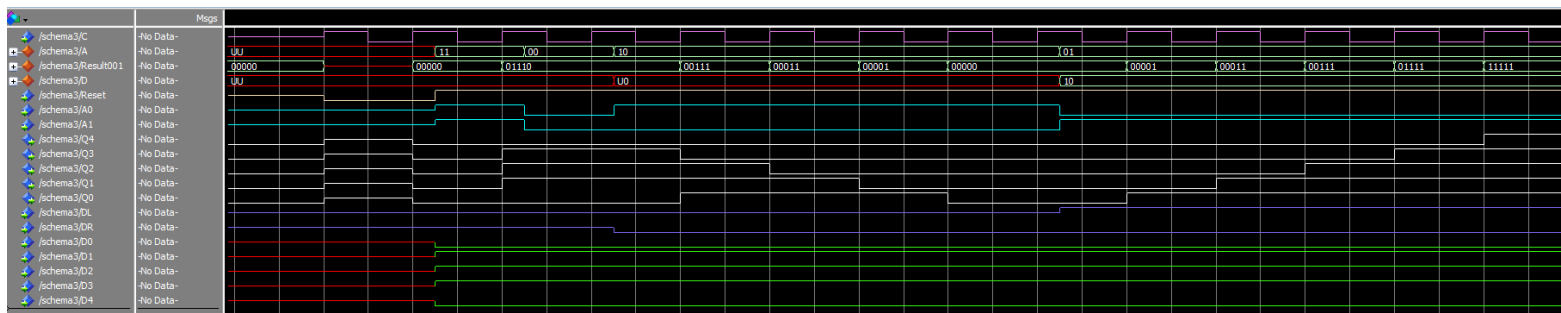
R	A <sub>0</sub>	A <sub>1</sub>	D <sub>R</sub>	D <sub>L</sub>	D <sub>4...D<sub>0</sub></sub>	Q <sub>4</sub>	Q <sub>3</sub>	Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>	Komentaras
0	x	x	x	x	x...x	0	0	0	0	0	Nustatymas į 0
1	0	0	x	x	x...x	Q <sub>4</sub>	Q <sub>3</sub>	Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>	Saugojimas
1	1	0	1	x	x...x	1	Q <sub>4</sub>	Q <sub>3</sub>	Q <sub>2</sub>	Q <sub>1</sub>	Postūmis į dešinę
			0	x		0	Q <sub>4</sub>	Q <sub>3</sub>	Q <sub>2</sub>	Q <sub>1</sub>	
1	0	1	x	1	x...x	Q <sub>3</sub>	Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>	1	Postūmis į kairę
			x	0		Q <sub>3</sub>	Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>	0	
1	1	1	x	x	D <sub>4...D<sub>0</sub></sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	Lygegretus įrašymas

- Stulpelis „R“ – tai registro resetas, kurio dėka bus galima, kontroliuoti ar registras turi vykdyti duota komandą ar ne. Taip pat reseto pagalba galėsime visus dydžius nustatyti į 0, kaip pavaizduota pirmoje eilutėje.
- Stulpelis „A<sub>0</sub> A<sub>1</sub>“ – tai du įvesties punktai, kurių dėka galėsime pasirinkti norimą atlikti komandą.
  - Kai A<sub>0</sub> = 0 ir A<sub>1</sub> = 0 informacija bus saugoma. (antra eilutė)
  - Kai A<sub>0</sub> = 1 ir A<sub>1</sub> = 0 bus vykdomas informacijos postūmis į dešinę. (trečia eilutė)
  - Kai A<sub>0</sub> = 0 ir A<sub>1</sub> = 1 bus vykdomas informacijos postūmis į kairę. (ketvirta eilutė)
  - Kai A<sub>0</sub> = 1 ir A<sub>1</sub> = 1 informacija bus įrašoma lygiagrečiai. (penkta eilutė)
- Stulpelis „D<sub>R</sub> D<sub>L</sub>“ – tai du įvesties punktai, kurių dėka galime pasirinkti, kokį skaitmenį įrašysime vykdant postūmius į atsilaisvintas vietas. Tai gali būti arba 0, arba 1.
- Stulpelis „D<sub>4...D<sub>0</sub></sub>“ – parodo per kokia operacija bus galima įrašyti naujus skaičius lygiagrečiai į registrą.
- Stulpelis „Q<sub>4...Q<sub>0</sub></sub>“ – tai stulpelis, kuris parodo kaip keičiasi informacijos pozicija(skaičiai) ar pati informacijos kiekvienos operacijos metu.

# Universalaus registro schema



## Universalaus registro simuliacijos scenarijus



Simuliacijos direktyvos:

```
force -freeze sim:/schema3/Reset 0 0
force -freeze sim:/schema3/C 1 0, 0 {50 ps} -r 100
run
force -freeze sim:/schema3/Reset 1 0
force -freeze sim:/schema3/A0 1 0
force -freeze sim:/schema3/A1 1 0
force -freeze sim:/schema3/D0 0 0
force -freeze sim:/schema3/D1 1 0
force -freeze sim:/schema3/D2 1 0
force -freeze sim:/schema3/D3 1 0
force -freeze sim:/schema3/D4 0 0
run
force -freeze sim:/schema3/A0 0 0
force -freeze sim:/schema3/A1 0 0
run
force -freeze sim:/schema3/DR 0 0
force -freeze sim:/schema3/A0 1 0
run
force -freeze sim:/schema3/A0 0 0
force -freeze sim:/schema3/A1 1 0
force -freeze sim:/schema3/DL 1 0
run
```

Rezultatai:

Iš simuliacijos galime pamatyti visą registro veikimo principą ir visas jo galimas operacijas.

1. Pirmiausia resetą „Reset“ nustatome į 0 ir uždedame clocką „C“. Tai padarę paleidę matysime kaip visa registro laikoma informacija yra nustatoma į 0.
2. A<sub>0</sub> ir A<sub>1</sub>, nustatome į vienetus ir Resetą nustatome į 1 ir į „D<sub>4</sub>...D<sub>0</sub>“ surašome norimą kombinaciją, kuri bus lygiagrečiai įrašyta į registrą. Paspaudę „run“ informacija surašoma į registrą.

3.  $A_0$  nustatome į 1 ir  $A_1$  nustatome į 0, taip pat į „ $D_R$ “ įrašę norimą skaitmenį (0 arba 1), kuris bus įrašytas į po poslinkio atlaisvintą vietą galime spausti „run“ ir stebėti postūmį į dešinę.
4.  $A_0$  nustatome į 0 ir  $A_1$  nustatome į 1, taip pat į „ $D_L$ “ įrašę norimą skaitmenį (0 arba 1), kuris bus įrašytas į po poslinkio atlaisvintą vietą galime spausti „run“ ir stebėti postūmį į kairę.
5. Jeigu  $A_0$  ir  $A_1$ , nustatysime į nulius, informacija, kuri yra saugoma registre nepasikais.

Visos šios operacijos yra daromos tada, kaip „C“ (clock) yra vienetas 1.

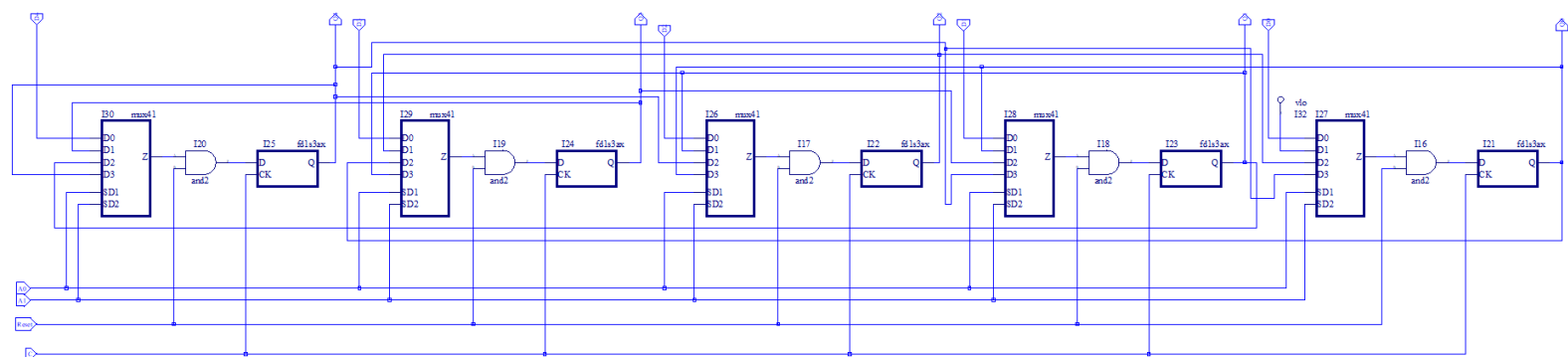
### 3. Specializuoto registro realizacija

#### Specializuoto registro veikimo lentelė

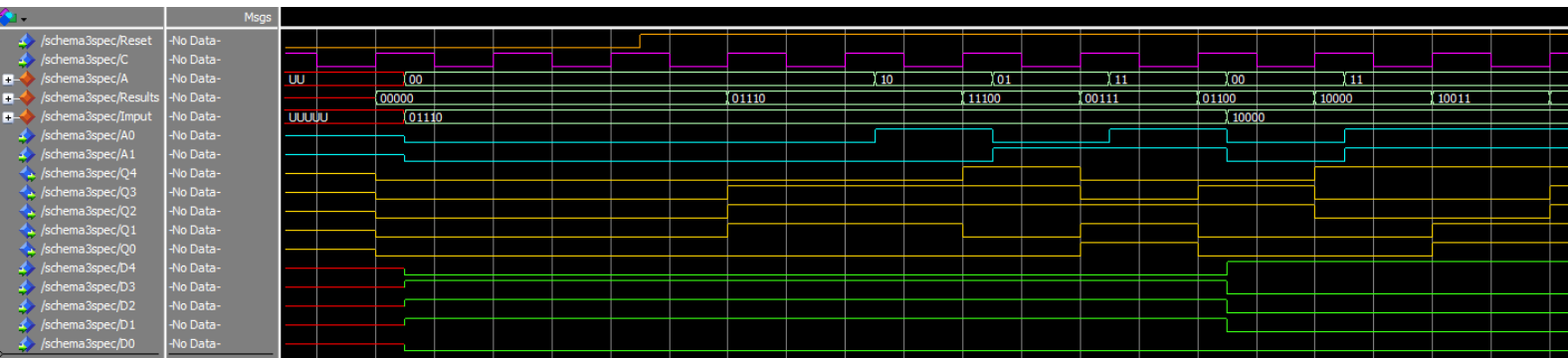
$A_0$	$A_1$	$D_4$	$D_3$	$D_2$	$D_1$	$D_0$	Paaiškinimas
0	0	$Q_4$	$Q_3$	$Q_2$	$Q_1$	$Q_0$	Informacijos įrašymas
1	0	$Q_3$	$Q_2$	$Q_1$	$Q_0$	0	LL1, Loginis p. į kairę
0	1	$Q_1$	$Q_0$	$Q_4$	$Q_3$	$Q_2$	CR2, Ciklinis p. į dešinę
1	1	$Q_4$ (0)	$Q_1$	$Q_0$	0	0	AL2, Aritmetinis atvirkštinis p. į dešinę
		$Q_4$ (1)	$Q_1$	$Q_0$	1	1	

- Stulpelis „ $A_0 A_1$ “ – tai du įvesties punktai, kurių dėka galėsime pasirinkti norimą atlikti komandą.
  - Kai  $A_0 = 0$  ir  $A_1 = 0$  informacija bus įrašoma į registrą. (pirma eilutė)
  - Kai  $A_0 = 1$  ir  $A_1 = 0$  bus vykdomas Loginis postūmis į kairę. (antra eilutė)
  - Kai  $A_0 = 0$  ir  $A_1 = 1$  bus vykdomas Ciklinis postūmis į dešinę. (trečia eilutė)
  - Kai  $A_0 = 1$  ir  $A_1 = 1$  bus vykdomas Aritmetinis atvirkštinis postūmis į dešinę. (ketvirta eilutė)
- Stulpelis „ $D_4...D_0$ “ – tai stulpelis, kuris parodo kaip keičiasi informacijos pozicija(skaiciai) ar pati informacijos kiekvienos operacijos metu.

#### Specializuoto registro schema



# Specializuoto registro simuliacijos scenarijus



Simuliacijos direktyvos:

```
force -freeze sim:/schema3spec/Reset 0 0
force -freeze sim:/schema3spec/C 1 0, 0 {50 ps} -r 100
run
force -freeze sim:/schema3spec/Reset 1 0
force -freeze sim:/schema3spec/A0 0 0
force -freeze sim:/schema3spec/A1 0 0
force -freeze sim:/schema3spec/D0 0 0
force -freeze sim:/schema3spec/D1 1 0
force -freeze sim:/schema3spec/D2 1 0
force -freeze sim:/schema3spec/D3 1 0
force -freeze sim:/schema3spec/D4 0 0
run
force -freeze sim:/schema3spec/A0 1 0
run
```

```
force -freeze sim:/schema3spec/A0 0 0
force -freeze sim:/schema3spec/A1 1 0
run
force -freeze sim:/schema3spec/A0 1 0
run
force -freeze sim:/schema3spec/A1 0 0
force -freeze sim:/schema3spec/A0 0 0
force -freeze sim:/schema3spec/D4 1 0
force -freeze sim:/schema3spec/D3 0 0
force -freeze sim:/schema3spec/D2 0 0
force -freeze sim:/schema3spec/D1 0 0
force -freeze sim:/schema3spec/D0 0 0
run
```

Rezultatai:

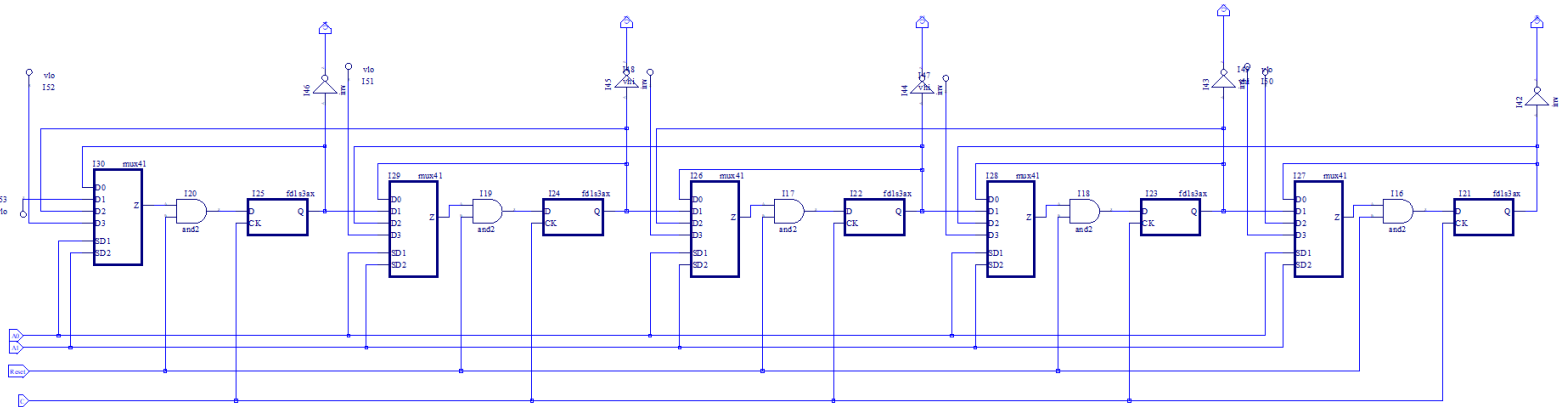
Iš simuliacijos galime pamatyti visą registro veikimo principą ir visas jo galimas operacijas.

1. Pirmiausia resetą „Reset“ nustatome į 0 ir uždedame clocką „C“. Tai padarę paleidę matysime kaip visa registro laikoma informacija yra nustatoma į 0.
2. A<sub>0</sub> ir A<sub>1</sub>, nustatome į nulius ir Resetą nustatome į 1 ir į „D<sub>4</sub>...D<sub>0</sub>“ surašome norimą kombinaciją, kuri bus lygiagrečiai įrašyta į registrą. Paspaudę „run“ informacija surašoma į registrą.
3. A<sub>0</sub> nustatome į 1 ir A<sub>1</sub> nustatome į 0, galime spausti „run“ ir stebėti Loginį postūmį į kairę. (LL1)
4. A<sub>0</sub> nustatome į 0 ir A<sub>1</sub> nustatome į 1, galime spausti „run“ ir stebėti Ciklinį postūmį į kairę. (CR2)
5. Jeigu A<sub>0</sub> ir A<sub>1</sub>, nustatysime į vienetus, ir paspausime „run“, matysime Aritmetinį atvirkštinį postūmį į kairę. (AL2)

Visos šios operacijos yra daromos tada, kaip „C“ (clock) yra vienetas 1.

## 4. Universalus Registro PLIS realizacija

### Universalus Registro schema, pritaikyta FPGA matricai



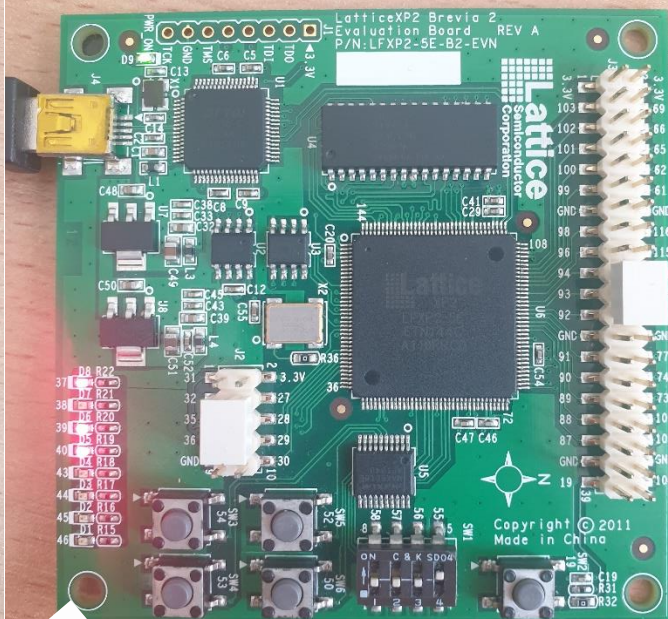
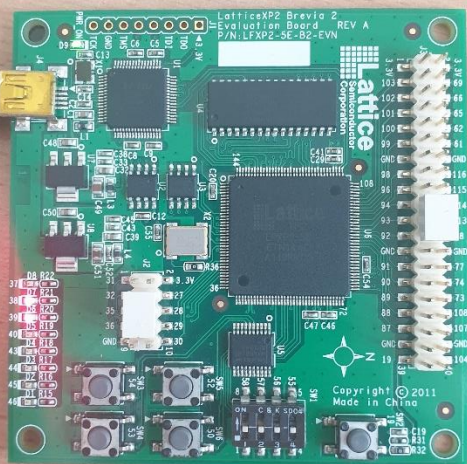
### Loginės schemos įvestimis/išvestimis

	Name	Group By	Pin	BANK	VREF	IO_TYPE	PULLMODE	DRIVE	SLEWRATE	PCICLAMP	OPENDRAIN	Outload (pF)	MaxSkew	Clock Load Only	SwitchingID	Ground plane PCB noise (mV)	F
1	All Ports	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.00	-0.0
1.1	Input	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
1.1.1	A0	N/A	56(56)	5(5)	N/A	LVC MOS2...	UP(UP)	NA(NA)	FAST(FAST)	OFF(OFF)	OFF(OFF)	N/A	N/A	N/A	N/A	N/A	N/A
1.1.2	A1	N/A	55(55)	5(5)	N/A	LVC MOS2...	UP(UP)	NA(NA)	FAST(FAST)	OFF(OFF)	OFF(OFF)	N/A	N/A	N/A	N/A	N/A	N/A
1.1.3	Clock	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
1.1.3.1	C	N/A	52(52)	5(5)	N/A	LVC MOS2...	UP(UP)	NA(NA)	FAST(FAST)	OFF(OFF)	OFF(OFF)	N/A	N/A	N/A	N/A	N/A	N/A
1.1.4	D0	N/A	45(45)	5(5)	N/A	LVC MOS2...	UP(UP)	NA(NA)	FAST(FAST)	OFF(OFF)	OFF(OFF)	N/A	N/A	N/A	N/A	N/A	N/A
1.1.5	D1	N/A	48(48)	5(5)	N/A	LVC MOS2...	UP(UP)	NA(NA)	FAST(FAST)	OFF(OFF)	OFF(OFF)	N/A	N/A	N/A	N/A	N/A	N/A
1.1.6	D2	N/A	46(46)	5(5)	N/A	LVC MOS2...	UP(UP)	NA(NA)	FAST(FAST)	OFF(OFF)	OFF(OFF)	N/A	N/A	N/A	N/A	N/A	N/A
1.1.7	D3	N/A	50(50)	5(5)	N/A	LVC MOS2...	UP(UP)	NA(NA)	FAST(FAST)	OFF(OFF)	OFF(OFF)	N/A	N/A	N/A	N/A	N/A	N/A
1.1.8	D4	N/A	53(53)	5(5)	N/A	LVC MOS2...	UP(UP)	NA(NA)	FAST(FAST)	OFF(OFF)	OFF(OFF)	N/A	N/A	N/A	N/A	N/A	N/A
1.1.9	DL	N/A	47(47)	5(5)	N/A	LVC MOS2...	UP(UP)	NA(NA)	FAST(FAST)	OFF(OFF)	OFF(OFF)	N/A	N/A	N/A	N/A	N/A	N/A
1.1.10	DR	N/A	54(54)	5(5)	N/A	LVC MOS2...	UP(UP)	NA(NA)	FAST(FAST)	OFF(OFF)	OFF(OFF)	N/A	N/A	N/A	N/A	N/A	N/A
1.1.11	Reset	N/A	58(58)	5(5)	N/A	LVC MOS2...	UP(UP)	NA(NA)	FAST(FAST)	OFF(OFF)	OFF(OFF)	N/A	N/A	N/A	N/A	N/A	N/A
1.2	Output	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
1.2.1	Q0	N/A	37(37)	5(5)	N/A	LVC MOS2...	UP(UP)	12(12)	FAST(FAST)	OFF(OFF)	OFF(OFF)	0.000	N/A	N/A	N/A	0.00	-0.0
1.2.2	Q1	N/A	38(38)	5(5)	N/A	LVC MOS2...	UP(UP)	12(12)	FAST(FAST)	OFF(OFF)	OFF(OFF)	0.000	N/A	N/A	N/A	0.00	-0.0
1.2.3	Q2	N/A	39(39)	5(5)	N/A	LVC MOS2...	UP(UP)	12(12)	FAST(FAST)	OFF(OFF)	OFF(OFF)	0.000	N/A	N/A	N/A	0.00	-0.0
1.2.4	Q3	N/A	40(40)	5(5)	N/A	LVC MOS2...	UP(UP)	12(12)	FAST(FAST)	OFF(OFF)	OFF(OFF)	0.000	N/A	N/A	N/A	0.00	-0.0
1.2.5	Q4	N/A	43(43)	5(5)	N/A	LVC MOS2...	UP(UP)	12(12)	FAST(FAST)	OFF(OFF)	OFF(OFF)	0.000	N/A	N/A	N/A	0.00	-0.0

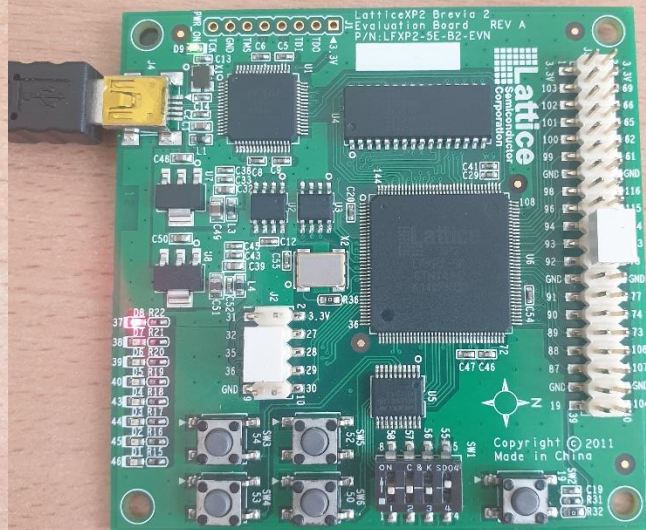
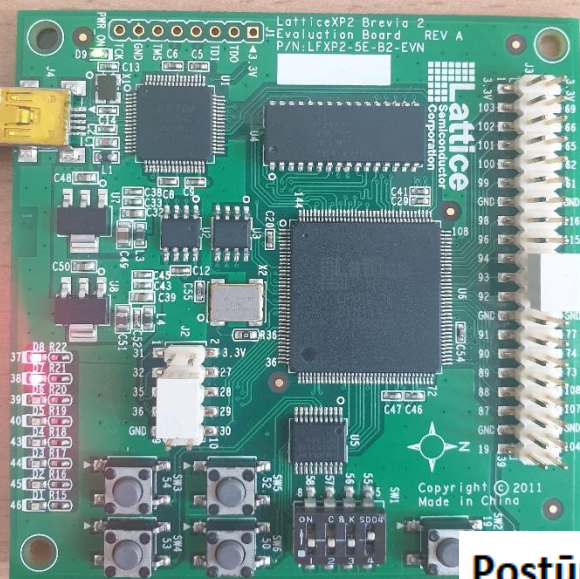
### Universalus Registro Rezultatai panaudojant PLIS matricą



Postūmis į kairę



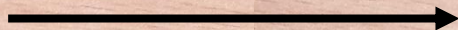
Postūmis į dešinę



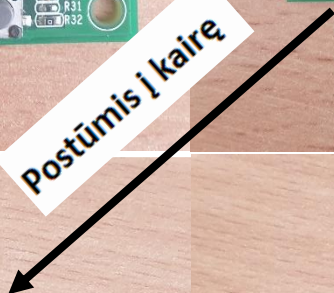
Postūmis į dešinę



Postūmis j kaire



Postūmis j kaire



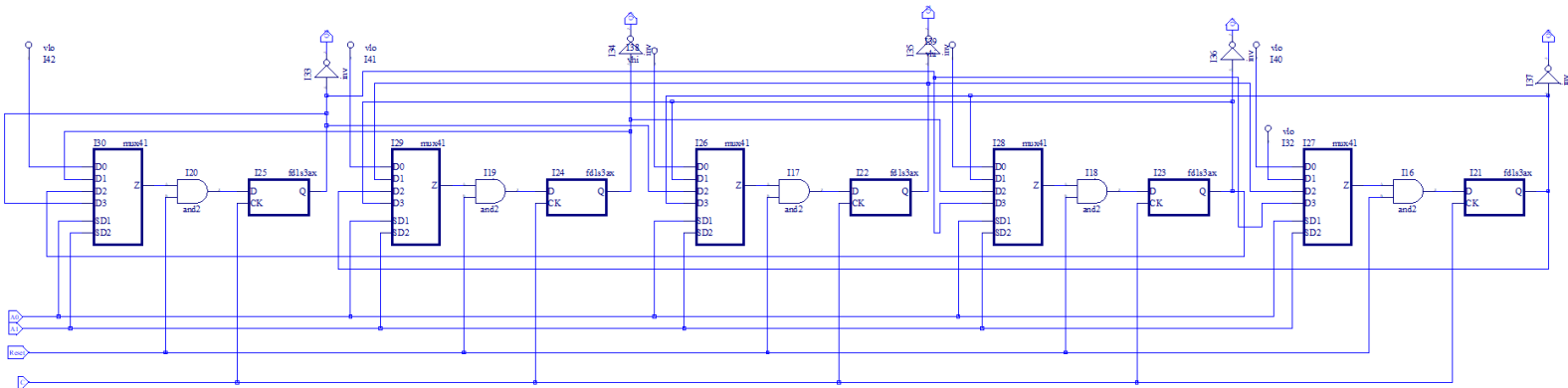
Postūmis j kaire





## 5. Specializuoto Registro PLIS realizacijos

Specializuoto Registro schema, pritaikyta FPGA matricai



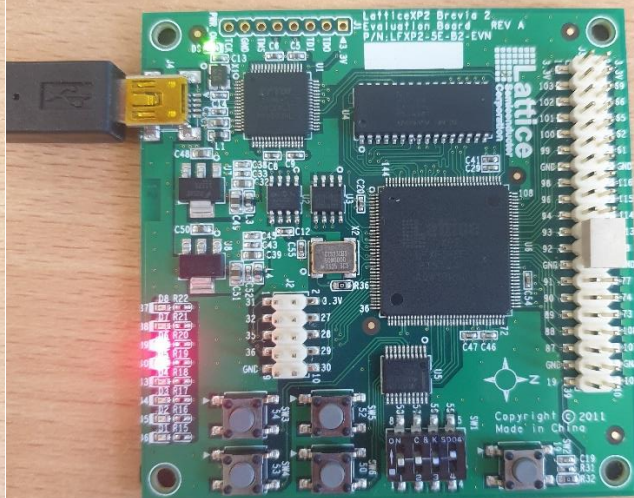
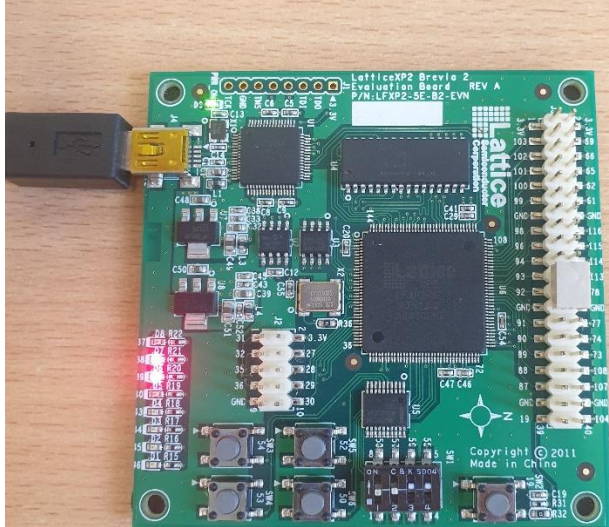
### Loginės schemos įvestimis/išvestimis

	Name	Group By	Pin	BANK	VREF	IO_TYPE	PULLMODE	DRIVE	SLEWRATE	PCICLAMP	OPENDRAIN	Outload (pF)	MaxSkew	Clock Load Only	SwitchingI/D	Ground plane PCB noise (mV)	F
1	▼ All Ports	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.00	-0.0
1.1	▼ Input	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
1.1.1	► A0	N/A	56(56)	5(5)	N/A	LVC MOS2...	UP(UP)	NA(NA)	FAST(FAST)	OFF(OFF)	OFF(OFF)	N/A	N/A	N/A	N/A	N/A	N/A
1.1.2	► A1	N/A	55(55)	5(5)	N/A	LVC MOS2...	UP(UP)	NA(NA)	FAST(FAST)	OFF(OFF)	OFF(OFF)	N/A	N/A	N/A	N/A	N/A	N/A
1.1.3	▼ Clock	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
1.1.3.1	► C	N/A	52(52)	5(5)	N/A	LVC MOS2...	UP(UP)	NA(NA)	FAST(FAST)	OFF(OFF)	OFF(OFF)	N/A	N/A	N/A	N/A	N/A	N/A
1.1.4	► D0	N/A	(45)	(5)	N/A	LVC MOS2...	UP(UP)	NA(NA)	FAST(FAST)	OFF(OFF)	OFF(OFF)	N/A	N/A	N/A	N/A	N/A	N/A
1.1.5	► D1	N/A	(48)	(5)	N/A	LVC MOS2...	UP(UP)	NA(NA)	FAST(FAST)	OFF(OFF)	OFF(OFF)	N/A	N/A	N/A	N/A	N/A	N/A
1.1.6	► D2	N/A	(46)	(5)	N/A	LVC MOS2...	UP(UP)	NA(NA)	FAST(FAST)	OFF(OFF)	OFF(OFF)	N/A	N/A	N/A	N/A	N/A	N/A
1.1.7	► D3	N/A	(50)	(5)	N/A	LVC MOS2...	UP(UP)	NA(NA)	FAST(FAST)	OFF(OFF)	OFF(OFF)	N/A	N/A	N/A	N/A	N/A	N/A
1.1.8	► D4	N/A	(53)	(5)	N/A	LVC MOS2...	UP(UP)	NA(NA)	FAST(FAST)	OFF(OFF)	OFF(OFF)	N/A	N/A	N/A	N/A	N/A	N/A
1.1.9	► DL	N/A	(47)	(5)	N/A	LVC MOS2...	UP(UP)	NA(NA)	FAST(FAST)	OFF(OFF)	OFF(OFF)	N/A	N/A	N/A	N/A	N/A	N/A
1.1.10	► DR	N/A	(54)	(5)	N/A	LVC MOS2...	UP(UP)	NA(NA)	FAST(FAST)	OFF(OFF)	OFF(OFF)	N/A	N/A	N/A	N/A	N/A	N/A
1.1.11	► Reset	N/A	58(58)	5(5)	N/A	LVC MOS2...	UP(UP)	NA(NA)	FAST(FAST)	OFF(OFF)	OFF(OFF)	N/A	N/A	N/A	N/A	N/A	N/A
1.2	▼ Output	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
1.2.1	► Q0	N/A	37(37)	5(5)	N/A	LVC MOS2...	UP(UP)	12(12)	FAST(FAST)	OFF(OFF)	OFF(OFF)	0.000	N/A	N/A	N/A	0.00	-0.0
1.2.2	► Q1	N/A	38(38)	5(5)	N/A	LVC MOS2...	UP(UP)	12(12)	FAST(FAST)	OFF(OFF)	OFF(OFF)	0.000	N/A	N/A	N/A	0.00	-0.0
1.2.3	► Q2	N/A	39(39)	5(5)	N/A	LVC MOS2...	UP(UP)	12(12)	FAST(FAST)	OFF(OFF)	OFF(OFF)	0.000	N/A	N/A	N/A	0.00	-0.0
1.2.4	► Q3	N/A	40(40)	5(5)	N/A	LVC MOS2...	UP(UP)	12(12)	FAST(FAST)	OFF(OFF)	OFF(OFF)	0.000	N/A	N/A	N/A	0.00	-0.0
1.2.5	► Q4	N/A	43(43)	5(5)	N/A	LVC MOS2...	UP(UP)	12(12)	FAST(FAST)	OFF(OFF)	OFF(OFF)	0.000	N/A	N/A	N/A	0.00	-0.0

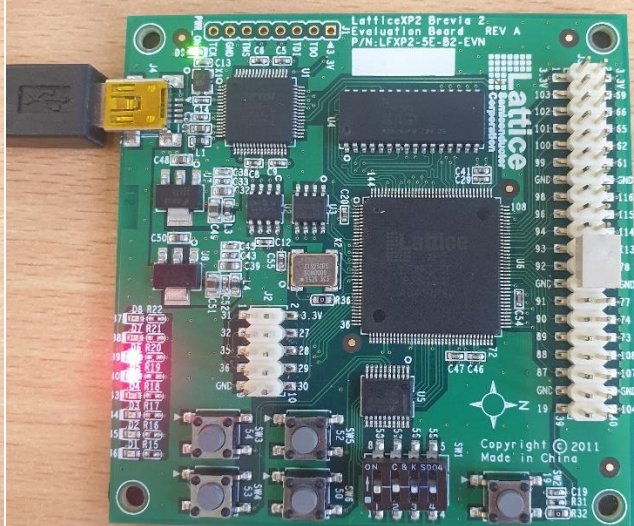
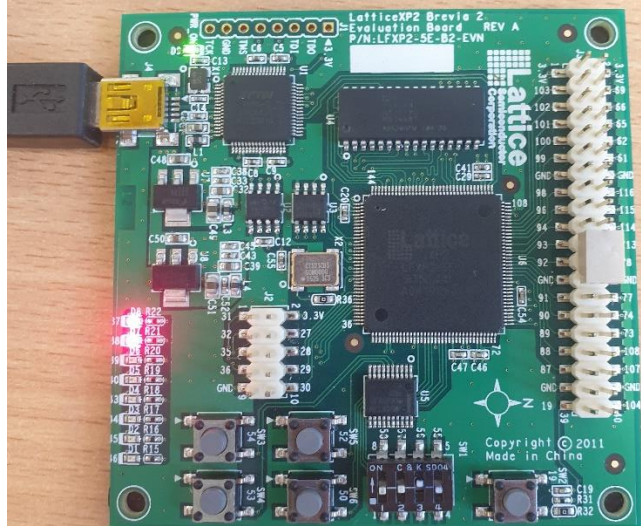
Universalaus Registro Rezultatai panaudojant PLIS matricą

Kombinacija (0 0 1 1 0)      Aritmetinis atvirkštinis p. į kairę (AL2)

Loginis p. į kairę (LL1)



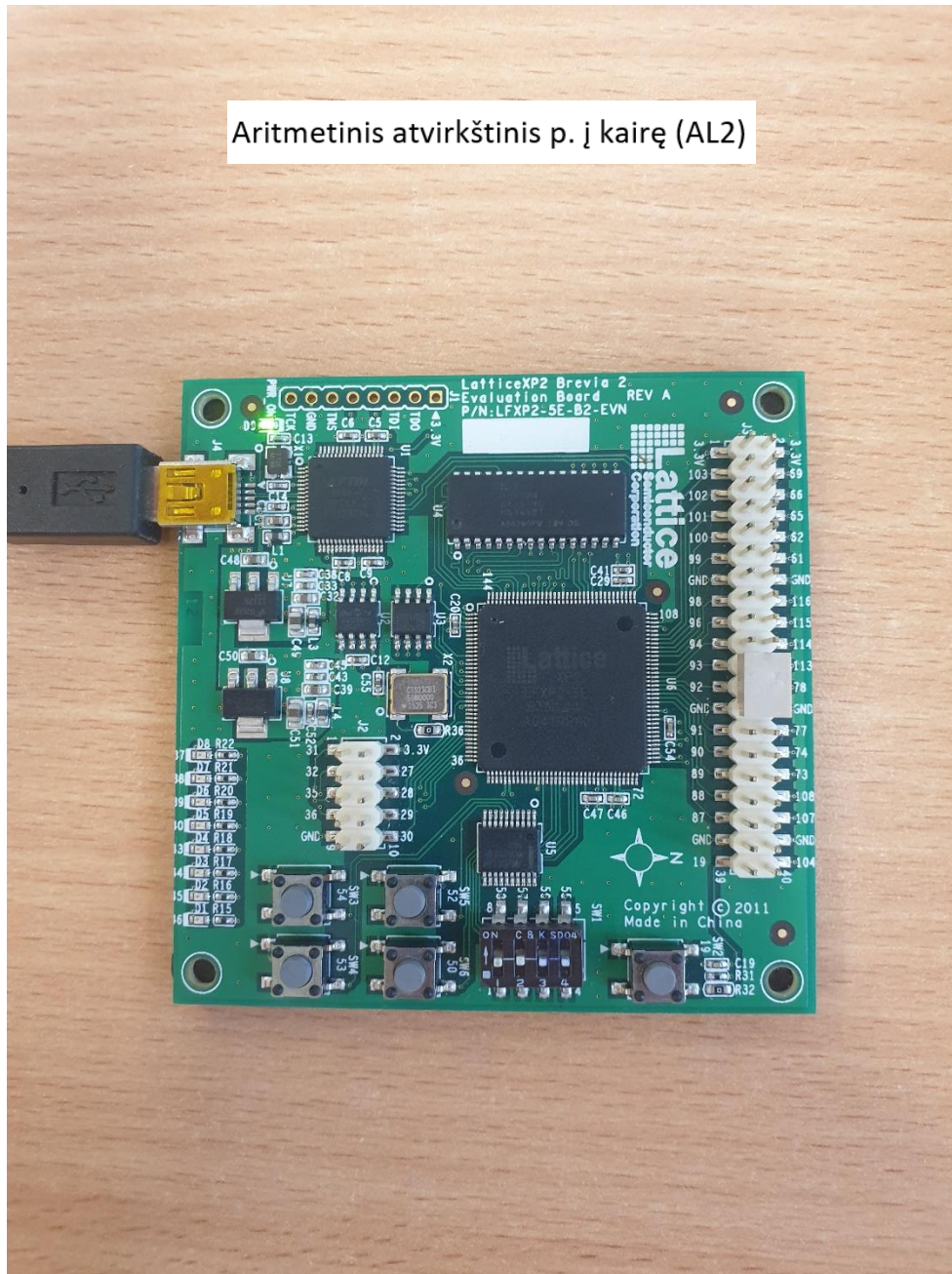
Ciklinis p. į dešinę (CR2)



Aritmetinis atvirkštinis p. į kairę (AL2)



Aritmetinis atvirkštinis p. į kairę (AL2)



## 6. Išvados

- Išmokau dirbti su Universaliais ir Specializuotais registrais.
- Išmokau sudaryti Registrų veikimo lenteles ir jas naudojant nubraižyti schemas, per Lattice Diamond.
- Išmokau sujungti matricą su mano sukurta schema per Lattice Diamond.