

**KAUNO TECHNOLOGIJOS UNIVERSITETAS**  
**INFORMATIKOS FAKULTETAS**

**SKAITMENINĖS LOGIKOS PRADMENYS(P175B100)**  
***Laboratorinių darbų ataskaita***

Atliko:

IFF-8/11 gr. Studentas

Arnas Švenčionis

2019 m. balandžio 3d.

Priėmė:

dėst. Petras Jakutis

**KAUNAS 2019**

1. Įvadas	2
1.1 Darbo tikslas	2
1.2 Darbo uždutis	2
2. Darbo atlikimas	3
2.1 Pirmojo skaitiklio kodo pakeitimai;	3
2.2 Stendo veikimas;	5
3. Išvados	12

# 1. Įvadas

## 1.1 Darbo tikslas

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

## 1.2 Darbo užduotis

Suprogramuoti skaitliuko schemą Lattice Brevia laboratoriniame stende.

Įsitikinti, ar schema veikia, sukurti ataskaitą.

146 variantas:

$M1 = 12;$

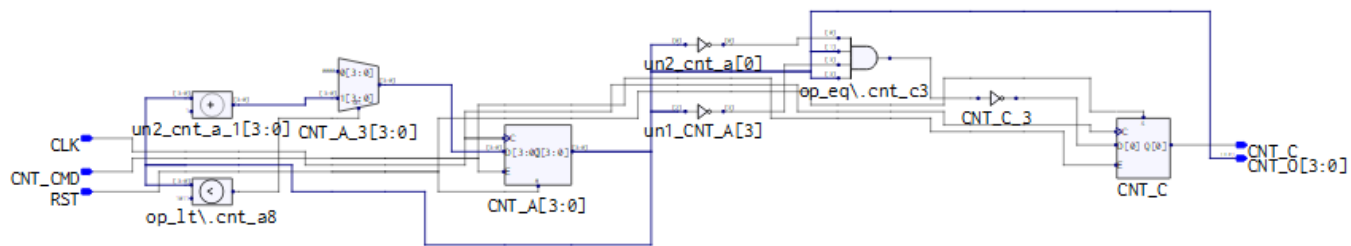
## 2. Darbo atlikimas

### 2.1 Pirmojo skaitiklio kodo pakeitimai;

Darbai naudosis pirmą savo skaitliuką (12). Pasiekus 11 schema įjungs pernašos signalą. Pakeisti reikia reset signalą iš 1 į 0 ir `CNT_O <= std_logic_vector(CNT_A);` į `CNT_O <= not (std_logic_vector(CNT_A));`.

```
7  library ieee;
8  use ieee.std_logic_1164.all;
9  use ieee.numeric_std.all;
10
11  entity CNT12 is port (
12      CLK          : in std_logic; --Sinchro signalas
13      RST          : in std_logic; -- Reset signalas
14      CNT_CMD      : in std_logic; -- Komanda
15      CNT_C        : out std_logic; --Pernasa
16      CNT_O        : out std_logic_vector(3 downto 0)
17  );
18  end CNT12;
19
20
21  architecture rtl of CNT12 is
22      signal CNT_A: unsigned (3 downto 0);
23  begin
24      process(CLK, RST, CNT_CMD)
25      begin
26          if RST = '0' then
27              CNT_A <= "0000";
28              CNT_C <= '1';
29          elsif CLK'event and CLK = '1' and CNT_CMD = '1' then
30              if CNT_A < 11 then
31                  CNT_A <= CNT_A + 1;
32                  if CNT_A = 10 then
33                      CNT_C <= '0';
34                  else
35                      CNT_C <= '1';
36                  end if;
37              else
38                  CNT_C <= '1';
39                  CNT_A <= "0000";
40              end if;
41          end if;
42      end process;
43      CNT_O <= not (std_logic_vector(CNT_A));
44  end rtl;
```

1 pav. Pakoreguotas M1 kodas



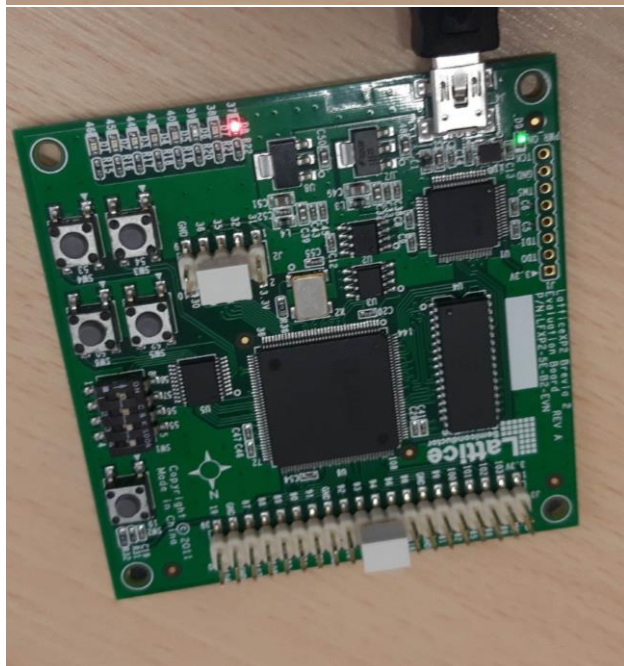
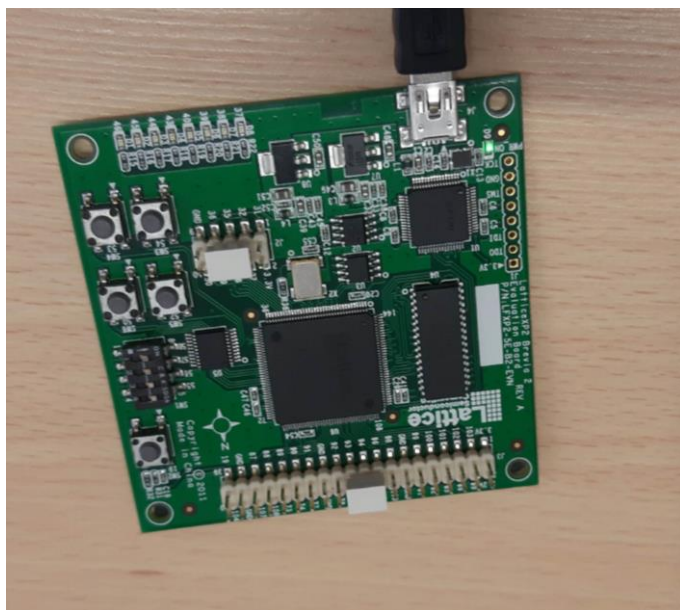
2pav. Symplify pro sudaryta schema

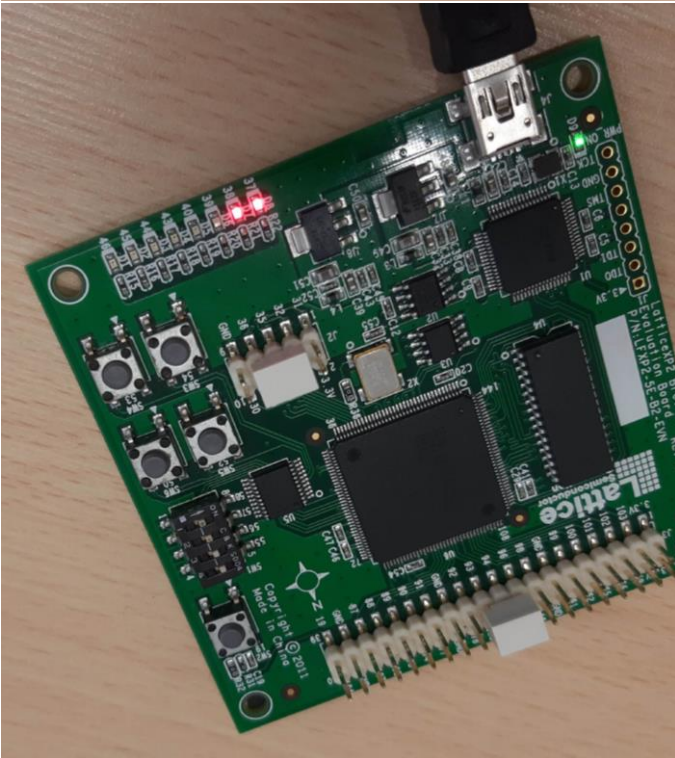
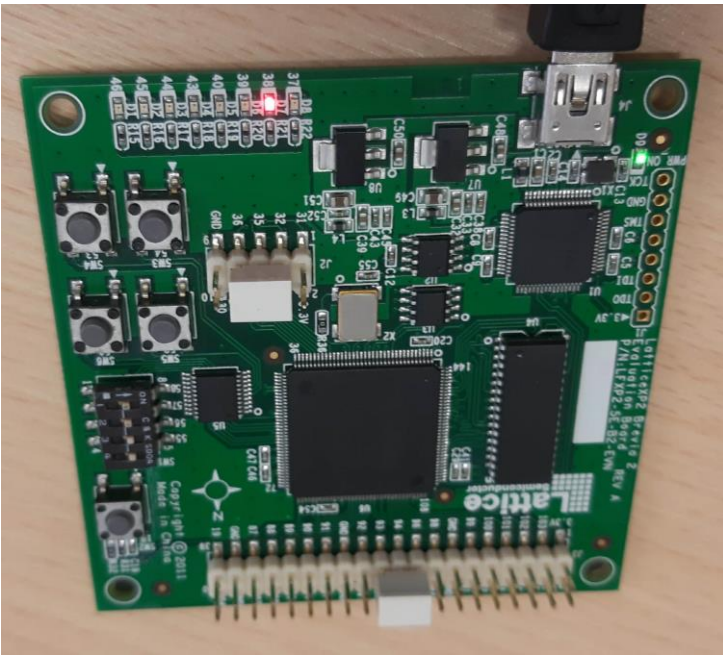
Name	Group Bv	Pin	BANK	VREF	IO TYPE	PULLMODE	DRIVE	SLEWRATE	PCICLAMP	OPENDRAIN	Outload (pF)
▼ All Ports	N/A	N/A	N/A	N/A			N/A				N/A
▼ Input	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
CNT_CMD	N/A			N/A	LVC MOS25	UP	NA	FAST	OFF	OFF	N/A
▼ Clock	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
CLK	N/A	52	5	N/A	LVC MOS25	UP	NA	FAST	OFF	OFF	N/A
RST	N/A	50	5	N/A	LVC MOS25	UP	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	N/A
CNT_C	N/A	46	5	N/A	LVC MOS25	UP	12	FAST	OFF	OFF	0.000
CNT_O[0]	N/A	37	5	N/A	LVC MOS25	UP	12	FAST	OFF	OFF	0.000
CNT_O[1]	N/A	38	5	N/A	LVC MOS25	UP	12	FAST	OFF	OFF	0.000
CNT_O[2]	N/A	39	5	N/A	LVC MOS25	UP	12	FAST	OFF	OFF	0.000
CNT_O[3]	N/A	40	5	N/A	LVC MOS25	UP	12	FAST	OFF	OFF	0.000

3pav. Spreadsheet view

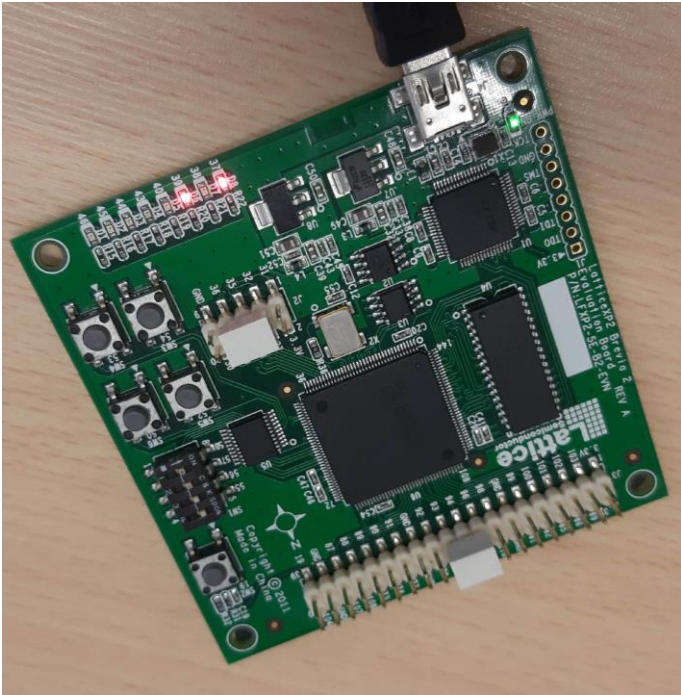
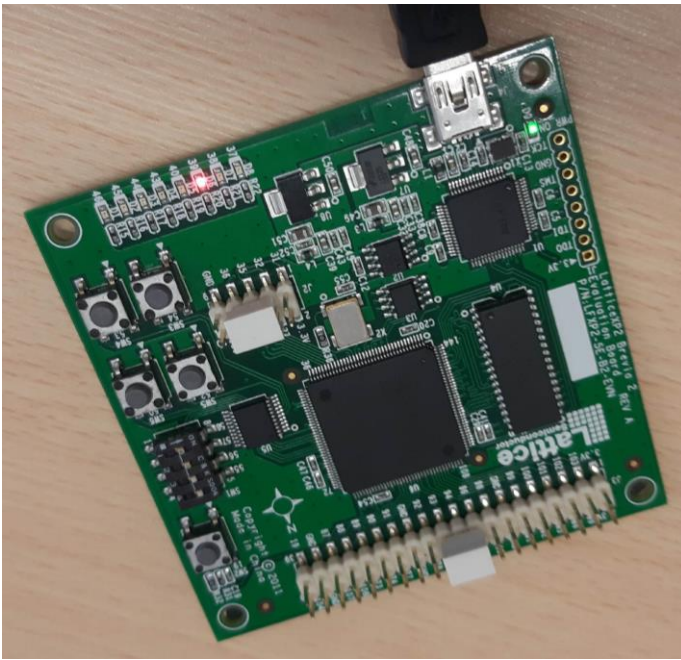
## 2.2 Stendo veikimas;

Kaskart paspaudus sinchrosignalu mygtuką, pridedamas 1 bitas.

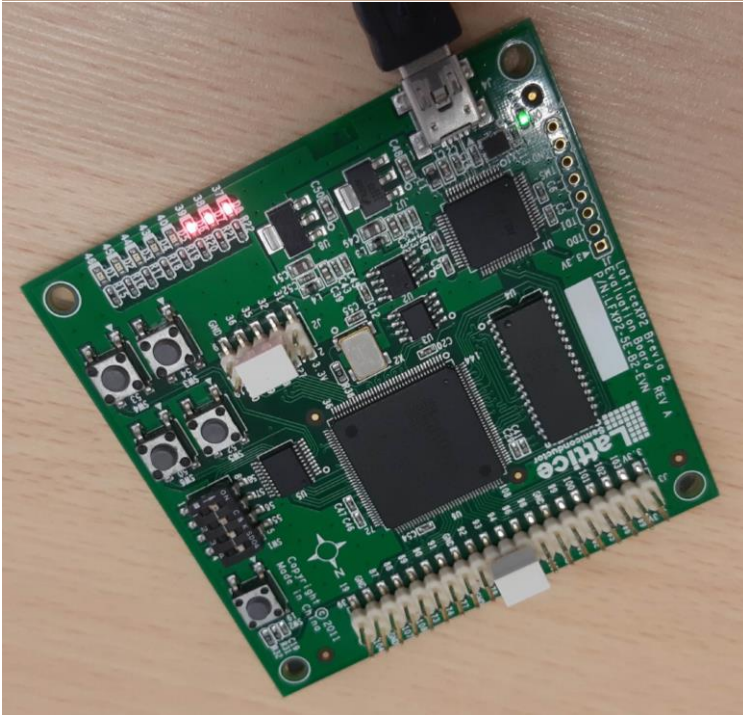
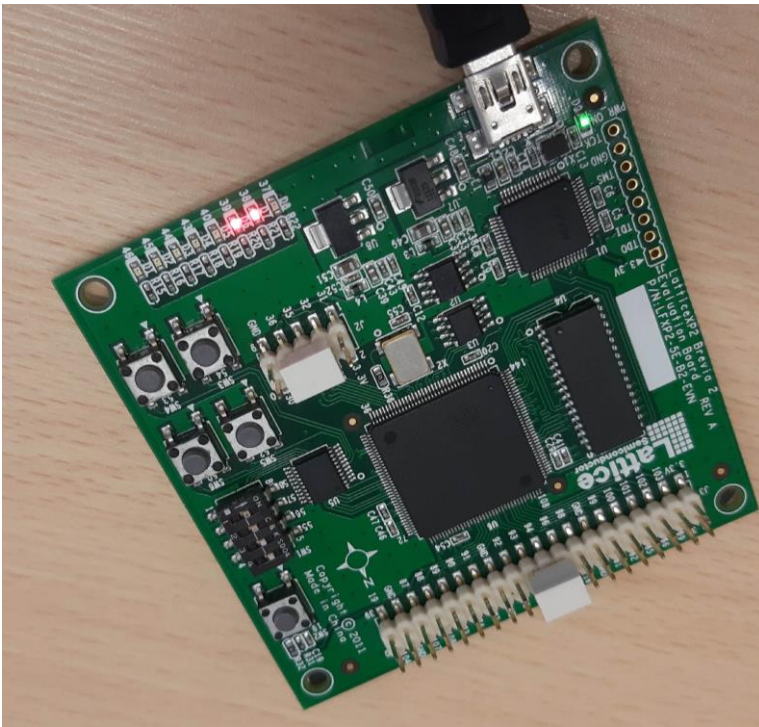


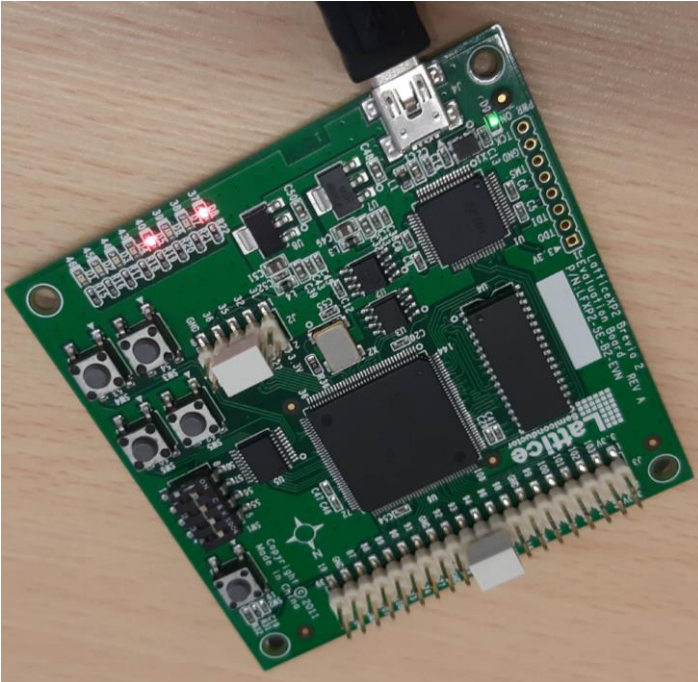
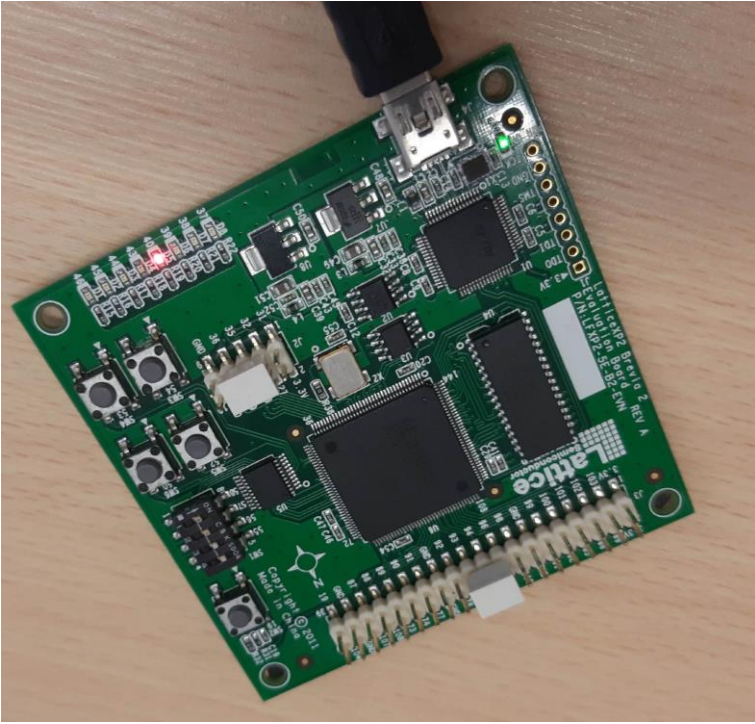




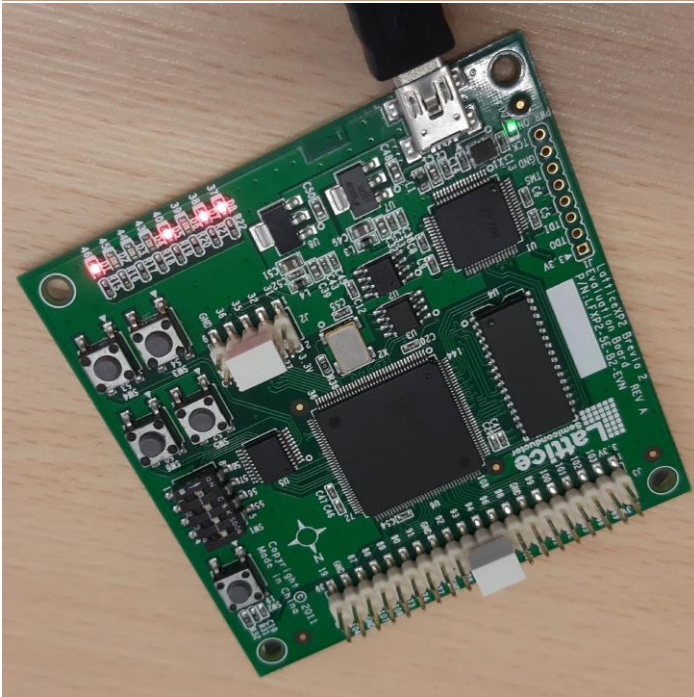
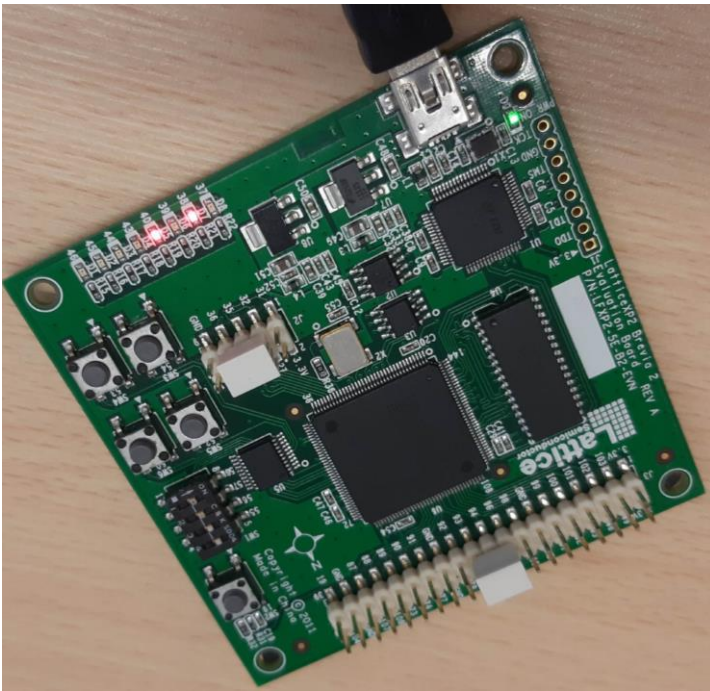


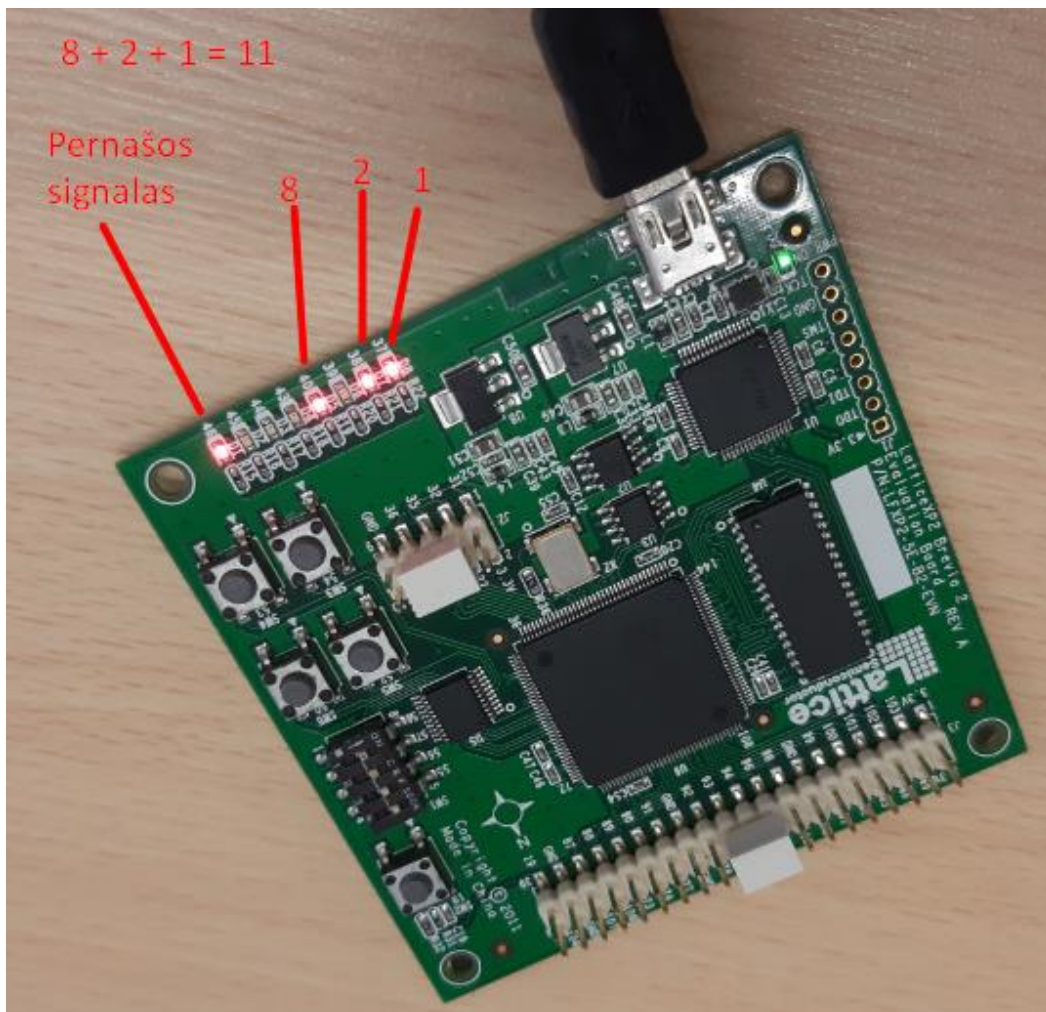






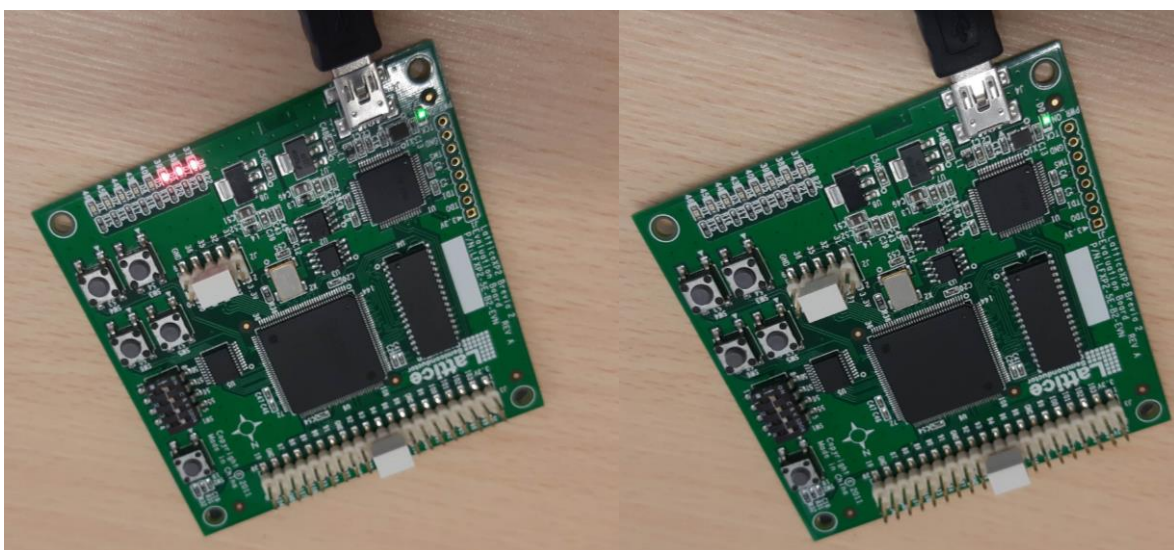






Su kiekvieno sinchrosignalų mygtuko paspaudimu, galima matyti bitų pridėjimą. Įjungta lemputė reiškia vienetą, o išjungta – nulį. Paskutinė lemputė (46) – pernašos signalas, kuris įsijungia dvejetainiai sistemai pasiekus 11 reikšmę (įsiskaito ir 0).

Esant betkokiai būsenai paspaudus reset mygtuką, schema resetinasi – grįžta į nulį.



### **3. Išvados**

Susipažinau su Lattice diamond galimybėmis perkelti programoje sudarytą schemą ar kodą į realią schemą, Kaip tai padaryti, kaip priskirti michroschemos mygtukus, lemputes prie inputų, outputų.