**KAUNO TECHNOLOGIJOS UNIVERSITETAS**

**INFORMATIKOS FAKULTETAS**

**Modulio P175B125 „Kompiuterių architektūra“**

Laboratorinio darbo aprašas (ataskaita)

**Pirmas laboratorinis darbas (tema 1.3. Procesorius. Valdymo ir operacinis įtaisai)**

|  |  |
| --- | --- |
|  | **Dėstytojas**  Tautvydas Bakšys |
|  |
| **Studentas**  Martynas Kemežys |

**KAUNAS, 2019**

Turinys

[1. Užduoties analizė 3](#_Toc18235180)

[2. Algoritmo medis 4](#_Toc18235181)

[3. simuliacijos rezultatai 5](#_Toc18235182)

[4. Rezultatų analizė 6](#_Toc18235183)

Šis laboratorinis darbas yra skirtas susipažindinti su konceptais reikalingais suprasti žemiausio lygio skaitmeninių įrenginių programavimo pagrindus. Laboratorinis darbas tęsia ir plėtoja Skaitmeninės logikos pradmenyse [P175B100] įgytas žinias ir įgūdžius. Laboratorinis darbas gali būti atliekamas bet kokiu VHDL simuliatoriumi, tačiau buvo pritaikytas darbui su Lattice Diamond Active HDL paketu.

Pagrindiniai laboratorinio darbo fokuso taškai yra:

• Skaitmeninę informaciją apdorojančių įrenginių struktūra

• Informacijos perdavimo tarp įrenginių būdai

• Skaitmeninės informacijos apdorojimo algoritmų kūrimas ir realizacija

• Algoritmų adresavimo tipai

Gauta užduotis:

* Variantas: **188.**
* Adresacija: **F(priverstinė).**
* Kodas: **T(Tiesioginis).**
* Formulė: **N1+N2²/-N3**
* Ženklai(N1,N2,N3): **+-+**

C:\Users\Vartotojas\AppData\Local\Microsoft\Windows\INetCache\Content.Word\uzd.png

# Užduoties analizė

* Formulė: **N1+N2²/-N3**
* Ženklai(N1,N2,N3): **+-+**

Įsivedame į registrus duomenis:

**B = N1;**

**C,E = N2;**

**D = N3;**

Spręsime tokią užduotį: N2**²** (operanda pakelti kvadratiniu laipsniu), mums reikia jį sudauginti su savimi viena kartą. Tam mes turime turėti jo modulį dviejuose registruose (galima tiesiogiai sudauginti du neigiamus skaičius, naudojantis pvz Berkso, Goldsteinoo ir Neimano metodu, tačiau tada operandas turėtų būti pervestas į papildomą kodą). Darant prielaidą, jog N2 operandas yra patalpintas į C ir E registrus, konvertuoti juos į skaičiaus modulį mums užteks 2 mikro komandų:

**LL1(C,E)**

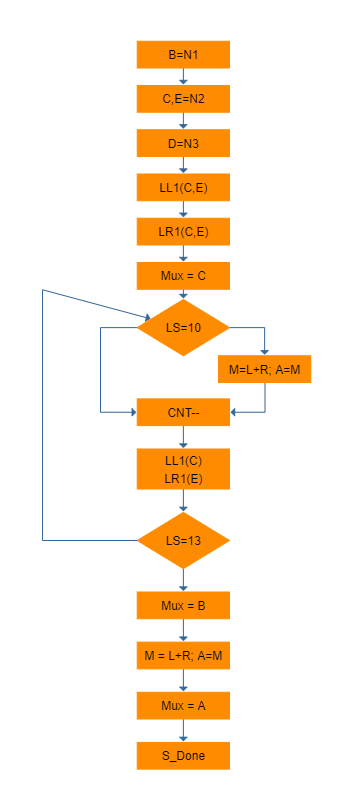
**LR1(C,E)**

Gavę abudu teigiamus operandus per mux‘ą paduodame į ALU C registrą:

**Mux = C;**

Ir tikrindami E registro žemiausią skiltį(**LS=10**) jei ji lygi 1, sumuojam(**M=L+R),** daugiklį C pastumiame į kairę**(LL1(C)),** dauginamajį E į dešine**(LR1(E**)), sumažiname skaitiklio **sk** turinį, kol skaitiklis **sk** nelygus 0, ir prieiname paskutinią skiltį(**LS=13, CNT=0**).

# Algoritmo medis



**0=> "10000000000000100000000000000000000000000000000000000000000000000000000000000001", --Ivedimas i B registra**

**1=> "10000000000000000000010000000000000100000000000000000000000000000000000000000010", --Ivedimas i C,E registrus**

**2=> "10000000000000000000000000001000000000000000000000000000000000000000000000000011", --Ivedimas i D registra**

**3=> "00000000000000000000001000000000000010000000000000000000000000000000000000000100", --LL1(C), LL1(E)**

**4=> "00000000000000000000000100000000000001000000000000000000000000000000000000000101", --LR1(C), LR1(E)**

**5=> "00010000000000000000000000000000000000000000000000000000000000000000000000000110", --I MUX'A pateikiamas C**

**6=> "00000000000000000000000000000000000000000000000000000000000000000000101000001000", --Tikrinamas 10 FLAG'AS**

**7=> "00100000000000000000000000000000000000000000000000000000000000000000000000001100", --I MUX'A PATEIKIAMAS B registras**

**8=> "00000000000000000000000000000000000000000000000000000000000000000001000000001010", --Mažinamas ciklas vienu**

**9=> "00000001000000000000000000000000000000000000000001000000000000000000000000001000", --Atliekama sudetis ir A prilyginamas M**

**10=> "00000000000000000000001000000000000001000000000000000000000000000000000000001011", --LL1(C), LR1(E)**

**11=> "00000000000000000000000000000000000000000000000000000000000000000000110100000110", --Tikrinamas 13 FLAG'AS**

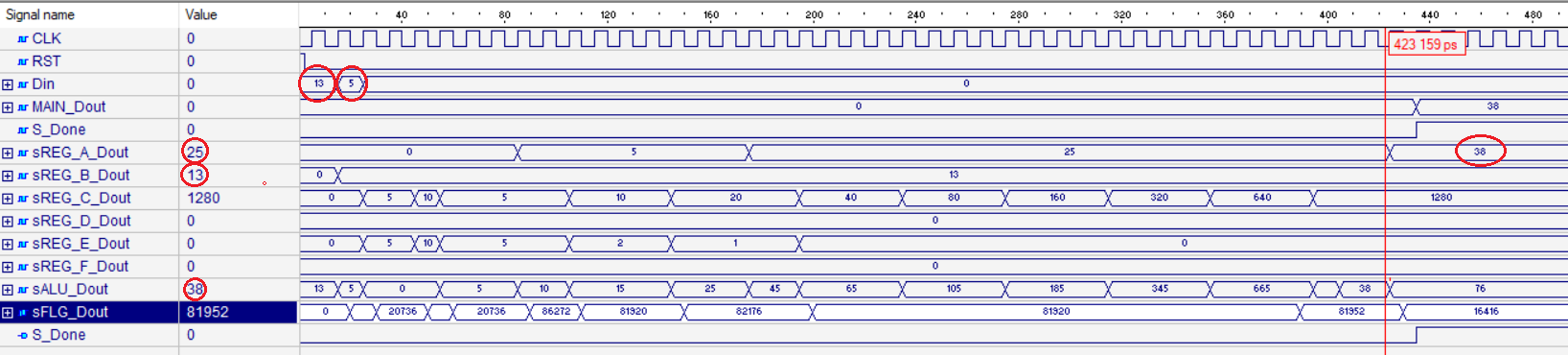
**12=> "00000001000000000000000000000000000000000000000001000000000000000000000000001101", --Atliekama sudetis ir A prilyginamas M**

**13=> "01000000000000000000000000000000000000000000000000000000000000000000000000001110", --I MUX'A PATEIKIAMAS A registras**

**14=> "00000000000000000000000000000000000000000000000000000000000000000010000000001110", --Išvedimas**

# simuliacijos rezultatai

Įvykdę simuliaciją ir gavę rezultatus, iš nuotraukos galime pastebėti kad įvedus **5** ir **13** gaunameteisinga atsakymą: **38**.



# Rezultatų analizė

Darbo metu gautų ir užfiksuotų rezultatų apibendrinimas bei išvadų formulavimas.