

## **ELESB16403: Kompiuterių architektūra**

### **1 laboratorinis darbas: Darbo aplinka**

#### **Git + Github užduotys**

Norėdami atlikti 1-3 užduotis užsiregistruokite adresu *GitHub.com*.

##### **1. Vietinės saugyklos sukūrimas ir sukurtos saugyklos kopijavimas**

Kompiuteryje pageidaujamoje vietoje sukurkite aplanką ir pereikite į sukurta katalogą:  
*cd lab1*

Pradėti registruoti failus ir aplankus:

*git init*

Pažiūrėti vietinės saugyklos būseną:

*git status*

Parsisiųskite duomenis į sukurta katalogą – vietinę saugyklą:

*git clone https://github.com/bartnykas/ca.git*

Norint pakeitimus išsaugoti nuotolinėje saugykloje būtina nurodyti savo tapatybę. Tam vietinė saugykla konfigūruojama:

*git config --global user.name "Vardas Pavardė"*

*git config --global user.email vardas.pavarde@stud.vgtu.lt*

Ar pakeitimai įvykdyti patikrinkime komanda:

*git config --list*

##### **2. Darbas su failais**

Failą *xor.vhd* papildykite eilute „-- Compiled by: Vardas Pavardė“. Šią eilutę įterpkite taip, kad ji eitų po eilutės „-- File: xor.vhd“. Tam galite pasinaudoti redaktoriumi *Notepad++*.

Norint nuotolinėje saugykloje išsaugoti failus pirmiausia juos reikia įtraukti. Pakeistą failą *xor.vhd* įtraukiame taip:

*git add xor.vhd*

Patvirtiname įtraukimą:

*git commit -m "xor.vhd failo pakeitimai"*

Nurodyto failo kopijavimas į nuotolinę saugyklą:

*git push*

Ar viskas pavyko peržiūrėkite apsilankę adresu *GitHub.com*.

#### **Modelsim užduotys**

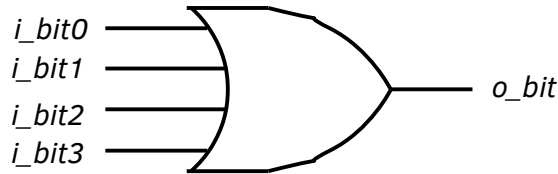
4. Perskaitykite medžiagą pateiktą adresu [nandland.com/vhdl/tutorials/tutorial-modelsimulation-walkthrough.html](http://nandland.com/vhdl/tutorials/tutorial-modelsimulation-walkthrough.html) ir atlikite tai kas nurodyta.

**ELESB16403: Kompiuterių architektūra**  
**2 laboratorinis darbas: Struktūrinis modeliavimas**

**Privalomos užduotys**

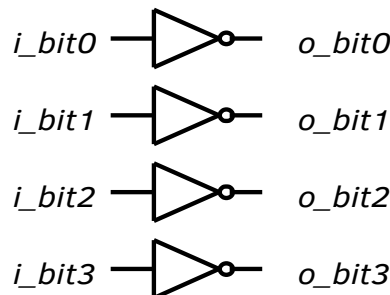
Pavyzdį kaip aprašyti loginę schemą rasite *xor2.vhd* faile<sup>1</sup>. Patikrinti sukurtą *vhd* failą galite adaptuodami failą *xor2.do*. Šis failas įvykdomas *Tools > TCL > Execute Macro*, bet prieš tai reikia nurodyti katalogą kuriame yra darbui skirti failai: *File > Change Directory*.

1. VHDL kalba aprašykite loginę schemą 4ARBA iš 1 pav. ir patikrinkite *ModelSim* programa.



1 pav. Loginio elemento 4ARBA loginė schema

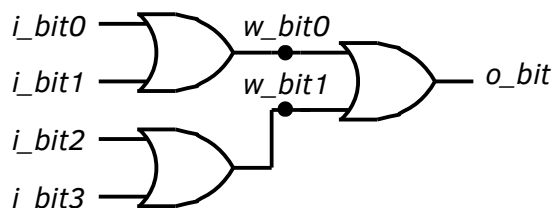
2. VHDL kalba aprašykite neigimą atliekančią keturių bitų loginę schemą iš 2 pav. ir patikrinkite *ModelSim* programa.



2 pav. Loginio elemento 4 bitų NE loginė schema

**Pasirenkamos užduotys**

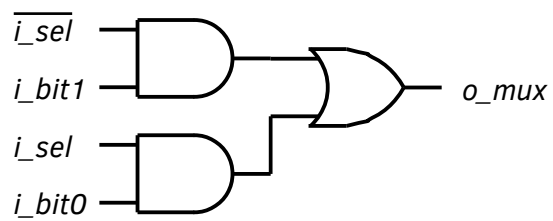
1. Kintamiesiems *w\_bit0* ir *w\_bit1* parinkite duomenų tipą *signal*<sup>2</sup> ir juose išsaugokite schemos iš 1 paveikslo tarpinius rezultatus.



2. VHDL kalba aprašykite multiplekserio iš 2 į 1 loginę schemą iš 3 pav. ir patikrinkite *Modelsim* programa.

<sup>1</sup> Daugiau informacijos apie VHDL rasite adresu [gstitt.ece.ufl.edu/courses/eel4712/labs/free\\_range\\_vhdl.pdf](http://gstitt.ece.ufl.edu/courses/eel4712/labs/free_range_vhdl.pdf)

<sup>2</sup> Pavyzdys adresu <https://www.nandland.com/vhdl/tutorials/tutorial-modelsim-simulation-walkthrough.html>



3 pav. Multiplekserio iš 2 į 1 loginė schema

**ELESB16403: Kompiuterių architektūra**  
**3 laboratorinis darbas: Concurrent statements**

**Privalomos užduotys**

4IR sumodeliuokite trimis skirtingais būdais, t. y. operatoriumiu „<=“ (angl. *concurrent signal assignment*), sakiniais *with/select* (angl. *selected signal assignment*) ir *when/else* (angl. *conditional signal assignment*).

**Pasirenkamos užduotys**

Sumodeliuokite loginę schemą, kuri aprašyta toliau pateikta reikšmių lentelė. Tam pasirinkite sakinį *with/select* arba *when/else*.

<i>in</i>	<i>out</i>
00	011
01	100
10	101
11	110