Lab 1: Introduction to Git and VHDL

EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research,
Development and Education

MINISTRY OF EDUCATION,
YOUTH AND SPORTS

The Study of Modern and Developing Engineering BUT
CZ.02.2.69/0.0/0.0/18_056/0013325

Learning objectives

After completing this lab you will be able to:

- Use markdown README files
- Create git repository
- Understand basic structure of VHDL files

In this laboratory exercise, you will learn how to use the git versioning system, write the markdown readme file, use the Windows/Linux console terminal to work in the lab, and then how to compose a basic VHDL code using the online development tool.

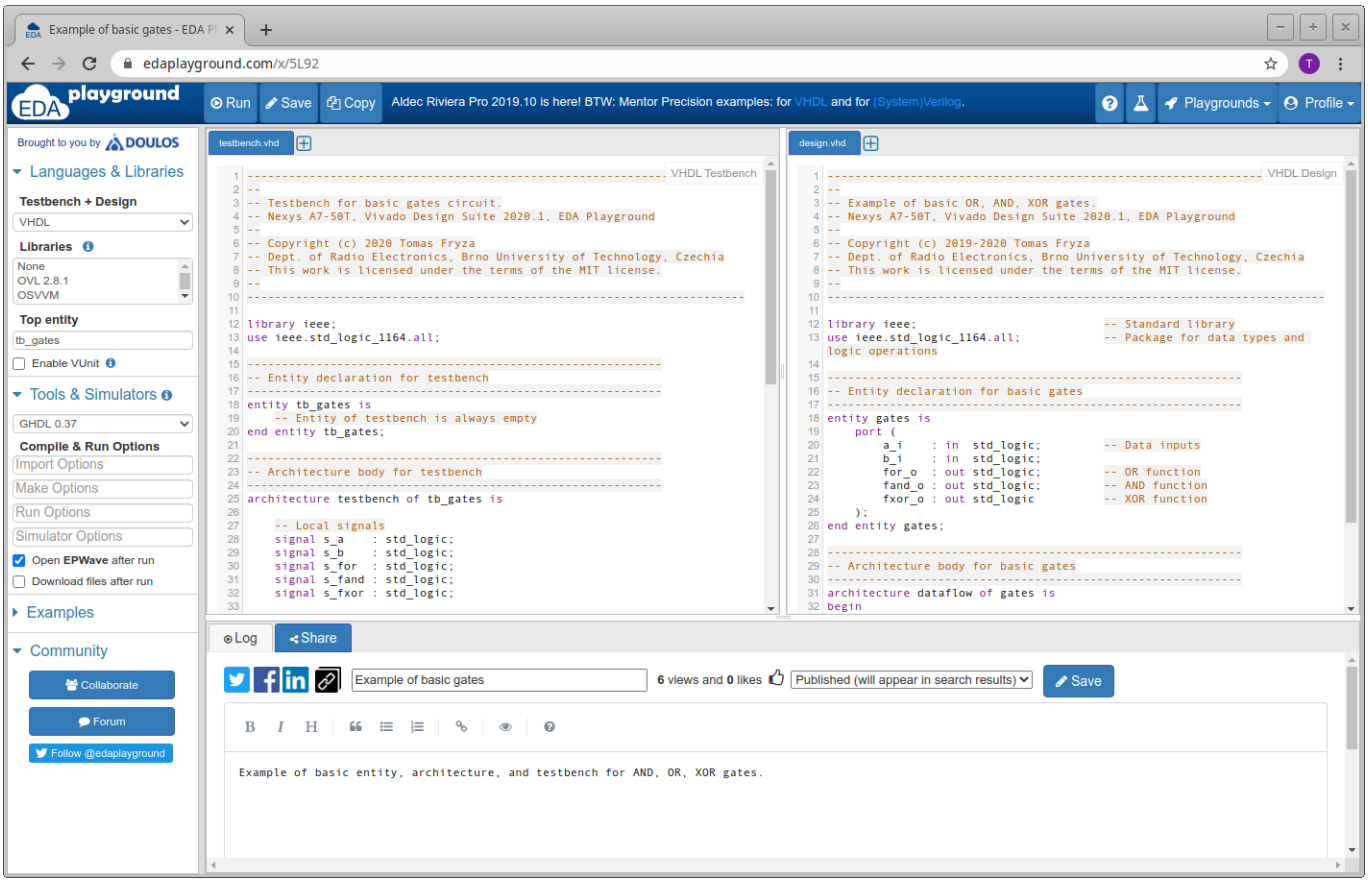

Table of contents

Preparation tasks (done before the lab at home)

Create an account on GitHub.

If you don't have Google or Facebook account, register your account on EDA Playground.

Download and install git.

Part 1: GitHub

GitHub is a code hosting platform for collaboration and version control. GitHub lets you (and others) work together on projects.

In GitHub, create a new public repository titled **Digital-electronics-1**. Initialize a README and MIT license.

Use one of the available git manuals, such as 1, 2, or 3, and add the following sections to your README file.

- Headers
- Emphasis (italics, bold)
- Lists (ordered, unordered)
- Links
- Table
- Listing of VHDL source code (with syntax highlighting)

Part 2: Local repository

Run Git Bash (Windows) of Terminal (Linux) and create your own home folder inside `Documents`.

```
## Windows Git Bash:
$ cd d:/Documents/
$ mkdir your-name
$ cd your-name/

## Linux:
$ cd
$ cd Documents/
$ mkdir your-name
$ cd your-name/
```

With help of `git` command, clone a local copy of your public repository.

```
## Windows Git Bash or Linux:
$ git clone https://github.com/your-github-account/Digital-electronics-1
$ cd Digital-electronics-1/
$ ls
LICENSE  README.md
```

Download `Docs` folder from this repository and copy it to your `Digital-electronics-1` local repository.

```
## Windows Git Bash or Linux:
$ ls
Docs  LICENSE  README.md
```

Create a new working folder `Labs/01-gates` for this exercise.

```
## Windows Git Bash or Linux:
$ mkdir Labs
$ cd Labs/
$ mkdir 01-gates
```

Part 3: EDA Playground

Open the Example of basic gates. Take a look at the basic parts of the VHDL source code, such as entity, architecture, and testbench. Use button **Run** to run the simulation and log in to your account using **Log In (save edits)**.

Most common VHDL operators are shown in the table.

Operator	Description
`<=`	Value assignment
`and`	Logical AND
`nand`	Logical AND with negated output
`or`	Logical OR
`nor`	Logical OR with negated output
`not`	Negation
`xor`	Exclusive OR
`xnor`	Exclusive OR with negated output
`-- comment`	Comments

Use De Morgan's laws and modify the following logic function to the form with NAND and NOR gates only. Verify all three functions in EDA Playground tool.

$$f(c,b,a) = \overline{b}\,a + \overline{c}\,\overline{b}$$
$$f(c,b,a)_{\mathrm{NAND}} =$$
$$f(c,b,a)_{\mathrm{NOR}} =$$

> Note that, equations were generated by Online LaTeX Equation Editor using the following code.

```
\begin{align*}
    f(c,b,a) =&~ \overline{b}\,a + \overline{c}\,\overline{b}\\
    f(c,b,a)_{\textup{NAND}} =&\\
    f(c,b,a)_{\textup{NOR}} =&\\
\end{align*}
```

Run any text editor, such as *Visual Studio Code* or *Atom*, open/create your `Digital-electronics-1/Labs/01-gates/README.md` local file (not on GitHub), complete tables with logical values, add link to your Playground and a screenshot with time waveforms from the simulator.

c	b	a	f(c,b,a)	f_NAND(c,b,a)	f_NOR(c,b,a)
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

Synchronize git

When you finish working, always synchronize the contents of your working folder with the local and remote versions of your repository. This way you are sure that you will not lose any of your changes.

Use git commands to add, commit, and push all local changes to your remote repository. Note that, a detailed description of all git commands can be found here. Check the repository at GitHub web page for changes.

```
## Windows Git Bash or Linux:
$ git status
$ git add <your-modified-files>
$ git status
$ git commit -m "[LAB] Creating 01-gates lab"
$ git status
```

```
$ git push
$ git status
```

Experiments on your own

1. In EDA Playground, verify Distributive laws:

$$x \cdot y + x \cdot z = x \cdot (y + z)$$
$$f1_{left} = x \cdot y + x \cdot z$$
$$f1_{right} = x \cdot (y + z)$$

$$(x + y) \cdot (x + z) = x + (y \cdot z)$$
$$f2_{left} = (x + y) \cdot (x + z)$$
$$f2_{right} = x + (y \cdot z)$$

2. Try several online graphics simulators, such as CircuitVerse, Logicly, CircuitLab, simulatorIO, LogicEmu, and compare their options.

Lab assignment

1. Submit the link to your `Digital-electronics-1` GitHub repository in the form `https://github.com/...`

2. Verification of De Morgan's laws of function f(c,b,a). Submit:

   - Equations of all three versions of logic function f(c,b,a),

$$f(c,b,a) = \overline{b}\,a + \overline{c}\,\overline{b}$$
$$f(c,b,a)_{\mathrm{NAND}} =$$
$$f(c,b,a)_{\mathrm{NOR}} =$$

   - Completed table with logic functions' values,
   - Listing of VHDL architecture from design file (`design.vhd`) with syntax highlighting,
   - Screenshot with simulated time waveforms of all three functions,
   - Link to your public EDA Playground example in the form `https://www.edaplayground.com/...`

3. Verification of Distributive laws. Submit:

   - Completed table with logic functions' values,
   - Listing of VHDL architecture from design file (`design.vhd`) with syntax highlighting,
   - Screenshot with simulated time waveforms of all four functions,
   - Link to your public EDA Playground example in the form `https://www.edaplayground.com/...`

Prepare all parts of the assignment on a computer (not by hand), insert them in your README file `Digital-electronics-1/Labs/01-gates/README.md`, export the formated output (not the listing in markdown

language) from HTML to PDF, use BUT e-learning web page and submit a single PDF file. The deadline for submitting the task is the day before the next laboratory exercise.