

# EECS 151/251A FPGA Lab

## Lab 0: Getting Set Up and Familiarizing Yourself with Tools

Prof. Borivoje Nikolic

TA: Vighnesh Iyer

Department of Electrical Engineering and Computer Sciences  
College of Engineering, University of California, Berkeley

## 1 Setting Up Accounts

### 1.1 bCourses and Piazza

If you are enrolled in this class, you should have access to the EECS 151/251A bCourses page (<https://bcourses.berkeley.edu/courses/1454619>) already. If you don't have access, please let a TA or the instructor know immediately so we can resolve the issue. We will be using bCourses to publish documents and make announcements for this course.

You should also register for a Piazza account and enroll in the EECS 151/251A class as soon as possible (<https://piazza.com/class/is8gjgauucn5rz>). On the class page, please submit a response for the GSI Office Hours Poll so that the TAs can set up office hours soon. We will be using Piazza as a discussion forum for this class and the labs. We will also send announcements through Piazza so it is vital that everyone is signed up.

### 1.2 Getting a EECS 151 Account

All students enrolled in the FPGA lab are required to get a EECS 151 class account to login to the workstations in lab. This semester, you can get a class account by using the webapp here:

<https://acropolis.cs.berkeley.edu/~account/webacct/>

Once you login using your CalNet ID, you can click on 'Get a new account' in the eeecs151 row. Once the account has been created, you can email your class account form to yourself to have a record of your account information.

Now you should be able to login to the workstations we have available in the lab. Enter your login and initial password in the login screen. Let the lab TA know if you have any problems setting up your class account.

#### 1.2.1 Changing your password

To change your default password, click on Applications on the top left toolbar on your workstation desktop, then hover over System, then click on Terminal. In the terminal type and execute the

command: `ssh update.cs.berkeley.edu`

You can then follow the prompts to set up a new password. You can always use the same webapp that you used to create your account to reset your password if you forget it.

### 1.3 Getting a Github Account

If you haven't done so previously, sign up for a Github account at <https://github.com/> with your berkeley.edu email address.

If you already have a Github account that's registered with your personal email address, don't create a new account. Instead, login to Github, go here <https://github.com/settings/emails>, and add your berkeley.edu email address to your Github account.

Once you have an account, make a private (but not anonymous) post on Piazza to the instructors with your Github account username/email and your class account login so we can give you access to the repositories you will need.

### 1.4 How to Login to the Lab Workstations From Your Laptop

## 2 Getting Familiar with our Development Environment

### 2.1 Linux Basics

In this class, we will be using a Linux development environment. We will be using CentOS as our Linux distro, which is a free version of Red Hat Linux. If you are unfamiliar or uncomfortable with Linux, and in particular, using the bash terminal, you should definitely check out this tutorial:

[https://www.digitalocean.com/community/tutorial\\_series/getting-started-with-linux](https://www.digitalocean.com/community/tutorial_series/getting-started-with-linux)

It is highly recommended to go through all four parts of the tutorial above, even if you already are familiar with the content. To complete the labs and projects for this course, you will find it helpful to have good command line skills.

One of the best ways to expand your working knowledge of bash is to watch others who are more experienced. Pay attention when you are watching someone else's screen and ask questions when you see something you don't understand. You will quickly learn many new commands and shortcuts.

### 2.2 Git Basics

Version control systems help track how files change over time and make it easier for collaborators to work on the same files and share their changes. For projects of any reasonable complexity, some sort of version control is an absolute necessity. There are tons of version control systems out there, each with some pros and cons. In this class, we will be using Git, one of the most popular version control systems. It is highly recommended that you make the effort to really understand how Git works, as it will make understanding how to actually use it much easier. Please check out the

following link, which provides a good high level overview:

<http://git-scm.com/book/en/Getting-Started-Git-Basics>

Once you think you understand the material above, please complete the following tutorial:

<http://try.github.com>

Git is a very powerful tool, but it can be a bit overwhelming at first. If you don't know what you are doing, you can really cause lots of headaches for yourself and those around you, so please be careful. If you are ever doubtful about how to do something with Git ask a TA or an experienced classmate.

For the purposes of this class you will probably only need to be proficient with the following commands:

- `git status`
- `git add`
- `git commit`
- `git pull`
- `git push`
- `git clone`

However, if you put in the effort to learn how to use some of the more powerful features (diff, blame, branch, log, mergetool, rebase, and many others), they can really increase your productivity.

Git has a huge feature set which is well documented on the internet. If there is something you think Git should be able to do, chances are the command already exists. We highly encourage you to explore and discuss with fellow classmates and TA's.

*Optional:* If you would like to explore further, check out the slightly more advanced tutorial written for CS250:

<http://inst.eecs.berkeley.edu/~cs250/fa13/handouts/tut1-git.pdf>

## 3 Setting Up Github Access

We will be using Github as our remote Git server for this class. Github is a popular Git hosting service which is home to many private and public (open-source) projects.

### 3.1 SSH Keys

Github authenticates you for access to your repository using ssh keys. Follow this tutorial to get SSH keys set up (this should be done on a lab workstation when you are logged in with your eecs151

```
class account):
```

```
https://help.github.com/articles/generating-an-ssh-key/
```

### 3.2 Acquiring Lab Files

The lab files, and eventually the project files, will be made available through a git repository provided by the staff. The suggested way to obtain these files is as follows. First, set up your ssh keys as described above. Then run the command below in your home directory,

```
git clone git@github.com:EECS150/labs_fa16.git
```

Whenever a new lab is released, you should only need to do a `git pull` to retrieve the new files. Furthermore, if there are any updates to the labs, `git pull` will fetch the changes and merge them in.

For now, you will only have pull access to this repository. If you make any local commits, you will not be able to push them to the remote server. Later on, each team will receive their own private repo for the project, and you will be able to push and pull from that.

## 4 Your First FPGA Design