EECS 151/251A FPGA Lab
Lab 1: Getting Set Up

Prof. John Wawrzynek
TAs: Quincy Huynh, Tan Nguyen
Department of Electrical Engineering and Computer Sciences
College of Engineering, University of California, Berkeley

1 Setting Up Accounts

1.1 Course website and Piazza

The course webpage can be found at http://inst.eecs.berkeley.edu/~eecs151/sp20/ and includes material for lectures, labs, discussion, and homework. You should register for a Piazza account and enroll in the EECS 151/251A class as soon as possible (https://piazza.com/berkeley/spring2020/eecs151251aa). We will be using Piazza to make announcements and as a discussion forum for this class and for the labs.

1.2 Getting an 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://inst.eecs.berkeley.edu/webacct

Once you login using your CalNet ID, you can click on 'Get a new account' in the eecs151 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.

1.4 Submitting Student Information Form

Once you have your accounts, make sure you have submitted this Google Form with your Github account username, email, and your class account login (eecs151-xxx). This will allow us to provide you with lab and project related resources (like github repos).

1.5 How to Login to the Lab Workstations From Your Laptop

The workstations used for this class are `c125m-1.eecs.berkeley.edu` through `c125m-19.eecs.berkeley.edu`, and are physically located in Cory 125. You can access all of these machines remotely through SSH. Others such as `eda-1.eecs.berkeley.edu` through `eda-8.eecs.berkeley.edu` are also available for remote login.

Login to the lab machines by SSHing to them with your class account `eecs151-xxx`.

```
ssh eecs151-xxx@c125m-1.eecs.berkeley.edu
```

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

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).

First, create a new SSH key (do this on the lab computer):

```
ssh-keygen -t rsa -b 4096 -C "your_email@berkeley.edu"
```

Keep hitting enter to use the default settings.

Then, from your terminal run:

```
cat ~/.ssh/id_rsa.pub
```

Copy the public key that's printed out in its entirety. Go here: https://github.com/settings/keys, click on 'New SSH Key', paste your public key into the box, and click 'Add SSH key'.

Finally test your SSH connection: https://help.github.com/articles/testing-your-ssh-connection/#platform-linux.

If you have any issues, ask a TA for help.

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/fpga_labs_sp20.git
```

Whenever a new lab is released, you should only need to `git pull` to retrieve the new files. Furthermore, if there are any updates, `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 References and Resources

We will be using the PYNQ-Z1 platforms offered by Xilinx and Digilent for all the labs and design project. This lab will not ask you to do anything with the boards yet, but you are encouraged

to read the PYNQ-Z1 Reference Manual: https://reference.digilentinc.com/reference/programmable-logic/pynq-z1/start as a starting point.

We will be using Xilinx Vivado Design (version 2019.1) as the primary FPGA software design tool (synthesis, placement, routing, and bitstream generation). Check this out

https://www.xilinx.com/content/dam/xilinx/support/documentation/sw_manuals/xilinx2019_1/ug910-vivado-getting-started.pdf

to get some initial impression on the tool. The lab machines should have Vivado installed. However, in case you would like to do the labs and project on your computers, you can also try installing Vivado locally. The Vivado WebPACK edition is free with limited device support (including PYNQ-Z1, fortunately). Refer to the following Appendix for installation instructions.

That's it! There is no checkoff or report for this lab.

A Installing Vivado Locally

A.1 Linux

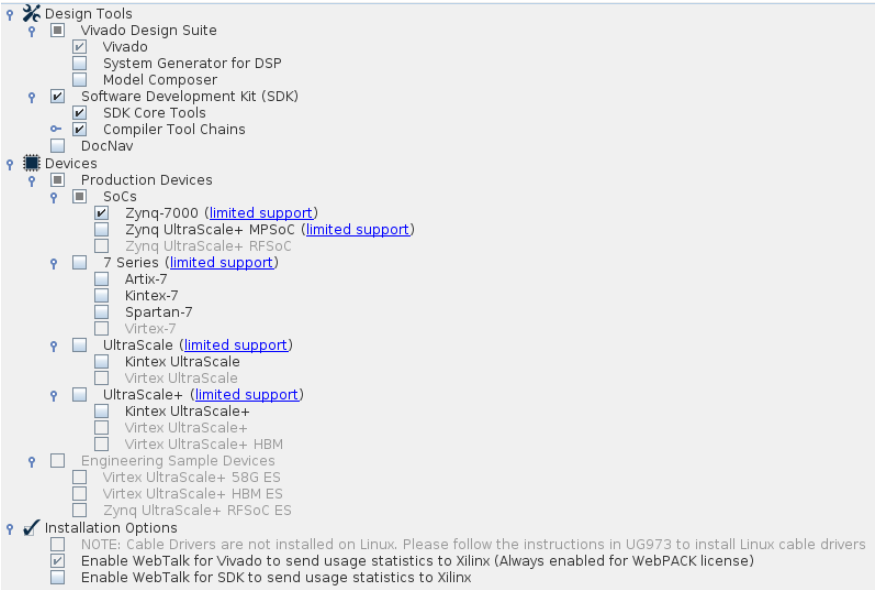
Create an install area for Vivado:

```
sudo mkdir /opt/Xilinx
sudo chmod 775 /opt/Xilinx
sudo chown `whoami`:`whoami` /opt/Xilinx
```

Download the 'Vivado Design Suite - HLx Editions - 2019.1 Full Product Installation' Linux bin from Xilinx. You will need to create a Xilinx account. Execute the downloaded script.

```
chmod +x Xilinx_Vivado_SDK_Web_2019.1_0524_1430_Lin64.bin
./Xilinx_Vivado_SDK_Web_2019.1_0524_1430_Lin64.bin
```

During the install process, specify `/opt/Xilinx` as the install directory. Also, you should only select the install options we're going to use in this class to save disk space:



After it's done, install the Digilent drivers to program the Pynq over USB-JTAG.

```
cd /opt/Xilinx/Vivado/2019.1/data/xicom/cable_drivers/lin64/install_script/install_drivers
sudo ./install_drivers
```

You also need to add PYNQ-Z1 board files to Vivado since it does not have PYNQ support initially. This allows you to set PYNQ-Z1 as the target platform when you create a Vivado project.

Download the following file https://github.com/cathalmccabe/pynq-z1_board_files/raw/master/pynq-z1.zip.

Extract and copy the `pynq-z1` folder to

`<your Vivado installation directory>/Vivado/<version>/data/boards/board_files`

A.2 Windows

Download the 'Vivado Design Suite - HLx Editions - 2019.1 Full Product Installation' Windows exe from Xilinx. You will need to create a Xilinx account. Execute the downloaded exe. Also, you should only select the install options we're going to use in this class to save disk space, as described in the Linux section.

A.3 Mac

Create an Ubuntu 18.04.3 VM using Virtualbox. Use this tutorial. Allocate at least 4GB of RAM and 50GB of disk space for the VM.

Install Vivado inside the VM using the steps in the Linux section.

There's some more stuff to do like tunneling the USB device from OSX to the VM; ask a TA.

Ackowlegement

This lab is the result of the work of many EECS151/251 GSIs over the years including:

- Sp12: James Parker, Daiwei Li, Shaoyi Cheng
- Sp13: Shaoyi Cheng, Vincent Lee
- Fa14: Simon Scott, Ian Juch
- Fa15: James Martin
- Fa16: Vighnesh Iyer
- Fa17: George Alexandrov, Vighnesh Iyer, Nathan Narevsky
- Sp18: Arya Reais-Parsi, Taehwan Kim
- Fa18: Ali Moin, George Alexandrov, Andy Zhou
- Sp19: Christopher Yarp, Arya Reais-Parsi
- Fa19: Cem Yalcin, Rebekah Zhao, Ryan Kaveh, Vighnesh Iyer