Determination of the Usability of FPGA Technology to Accelerate Option Pricing Algorithms

User Guide

Matthew James Carter
201371920

Table of Contents

1. Introduction

1.1 Overview

This user guide sets out to inform users how to build and execute the code listed in [1]. This collection of code was developed as part of a Master's thesis at the University of Liverpool. The supporting dissertation and results are also listed in [1].

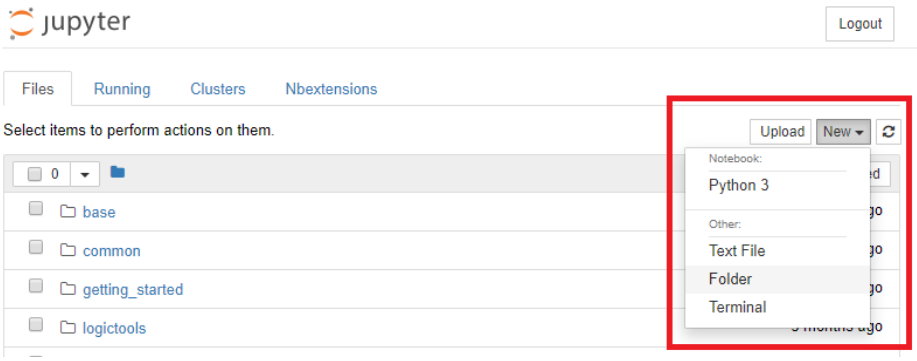
1.2 Requirements

Firstly, users must clone the Github repository listed in [1]. Following this, users must have the following:

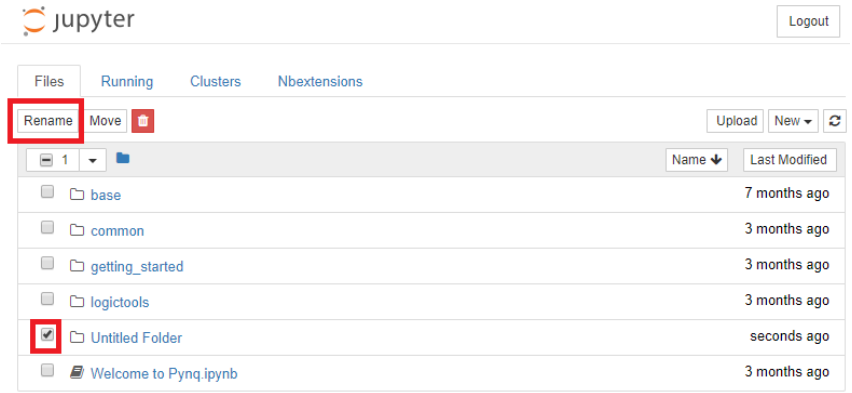
- A Pynq-Z2 board with the relevant board image installed. Instructions on how to install this board image can be found at [2].
- An Alveo u200 mounted and setup for your system. Installation instructions for the Alveo u200 card can be found at [3].

2. Pynq-Z2

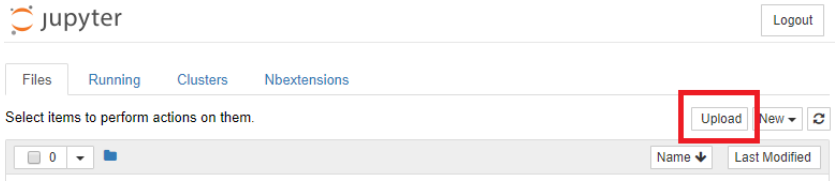
After installing the Pynq-Z2 board image, navigate to http://pynq in your web browser which will open the Jupyter environment. From here, click the “new” button towards the top right-hand side of the environment and then select “folder”.



We can rename this folder by selecting the check box to the left of it and then clicking “rename” towards the top left-hand side of the environment.

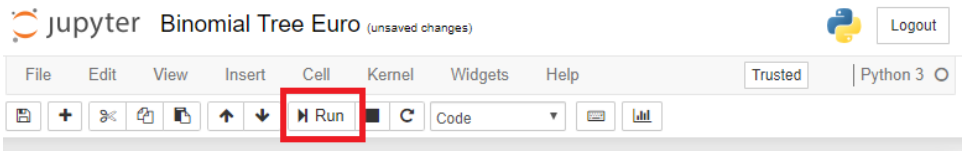


Rename the folder to the name of the algorithm you are going to use, e.g. European Binomial Tree. Navigate into the folder by clicking it. You can then upload the algorithm to the Pynq-Z2 by clicking on the “upload” button towards the top right-hand side of the environment.



Navigate to the cloned Github repository on your computer and navigate into the folder of the algorithm you wish to use. Navigate to the “Pynq-Z2” folder and then upload **all** of the contents in this folder.

You can now run the algorithm through the Jupyter environment or through the command line. To run in the Jupyter environment, click on the *.ipynb* file that you uploaded. This will launch the Jupyter notebook, the algorithm can then be run by clicking the "Run" button towards the top of the notebook.



You will have to click run for each cell in the notebook, alternatively you can press *shift + enter* to execute each cell in the notebook. To execute via command line, first navigate to the algorithms folder. From here, you can execute the *.py* file that you uploaded, for example:

sudo python3 eubinomialtree.py

The Python file **must** be executed with the "suffix" prefix. If a number of zeroes appear in the output, simply execute the file again as the cache coherency on the Pynq-Z2 will have failed.

2.1 Binomial Tree

A file called *option_data.txt* is included for the Binomial Trees algorithm. Here, you can enter data for up to 25 options. The price for all options will then be returned at the end in the form of an array.

```
# Option pricing input file
# -------------------------
# You may enter data for up to 25 options, any extra will be ignored
#
# Info
# S - Stock Price       (float) Underlying stock price
# K - Strike Price      (float) Options strike price
# T - Time to Maturity  (float) Time (in years) until the option expires
# D - Dividend Yield    (float) Eg for 5% enter "5" and not "0.05"
# r - Risk-free Rate    (float) Eg for 5% enter "5" and not "0.05"
# v - Volaility         (float) Eg for 20% enter "20" and not "0.2"
# type                  (int)   The type of option, enter "0" for call and "1" for put
# height                (int)   The height of the tree
#
# Structure input data
#  S,K,T,D,r,v,type,height
50,50,1,0,0.05,0.20,1,30000
```

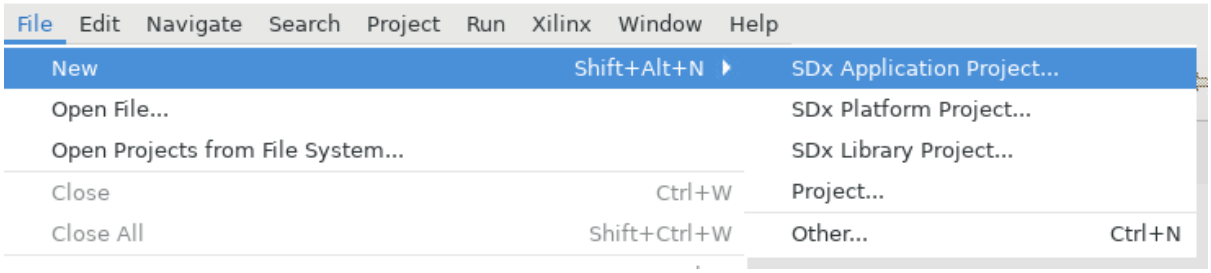
2.2 Monte Carlo

For the Monte Carlo simulation, the stock data and simulation data are set in the notebook/Python file themselves. The data is set in cell 5 of the notebook or between lines 121 and 130 of the Python file. These values must be set before you execute the notebook or Python file.

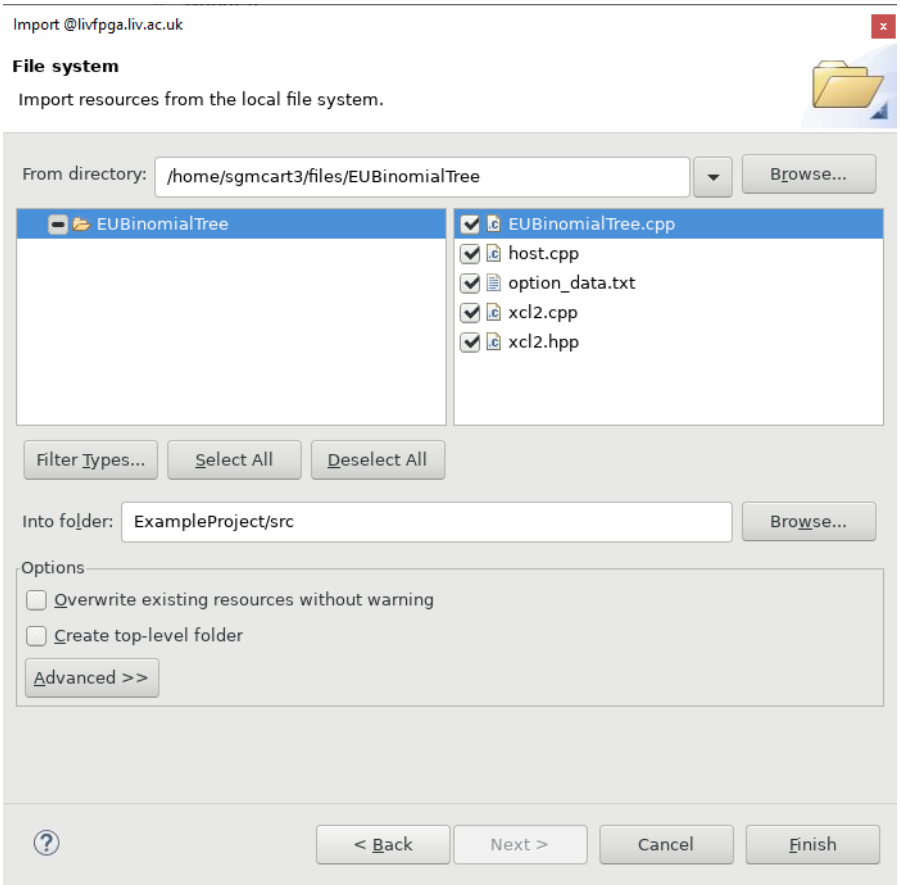
```
S[0] = 50
K[0] = 50
T[0] = 1
D[0] = 0
r[0] = 0.05
v[0] = 0.20
type_r = 1
M = 1000000
N = 100
```

3. Alveo u200

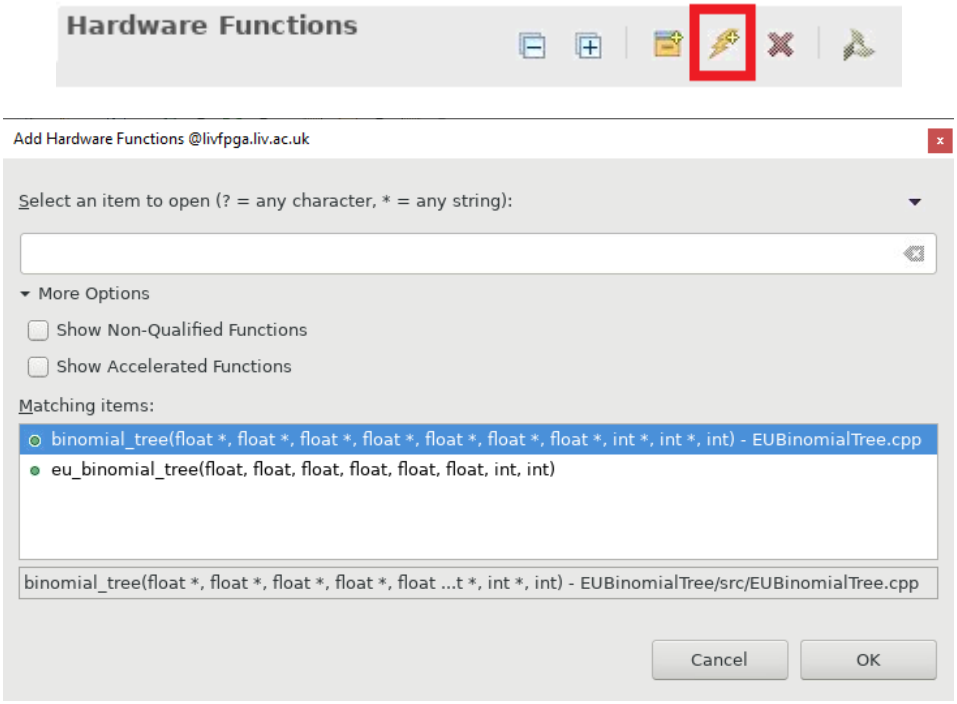
Start by creating a new SDAccel Application Project, this is done by clicking file -> new -> SDAccel Application Project. Ensure that you select the Alveo u200 platform and create an empty application.



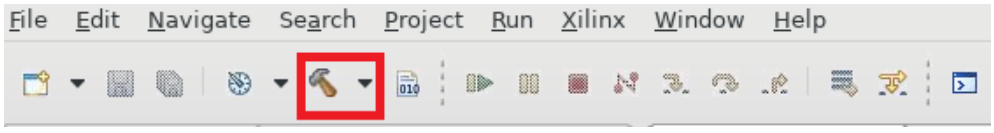
Following this, we are ready to import the source files into the project. If you are working on a remote server, you must first upload the relevant SDAccel and HLS files from the Github repository (NOTE: for the Monte Carlo algorithm there is a separate HLS file for the Pynq-Z2 and Alveo u200 implementation). Following this, right click on the "src" folder of the project and select "import". Then select "General -> File System" and navigate to where the source files are on your system. Highlight them and click "finish".



Now add a hardware function to the project. This is done by clicking the lightning icon next to the “Hardware Functions” title. Select the name of the top level function for the algorithm – this is always the name of the type of algorithm you are running, for example if running a Binomial Tree algorithm, the top-level function is “binomial_tree”.

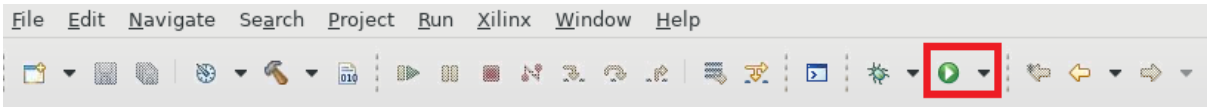


The project is now ready to be built. The C code and hardware design can be verified by selecting either the *sw_emu* or *hw_emu* build configuration. To build for the FPGA, select *System* (NOTE: building for the FPGA takes a minimum of 3 hours for the Binomial Tree algorithms and 10 hours for the Monte Carlo algorithms). After selecting the build configuration, click on the hammer icon in the toolbar.



After the project is built, select the “Run” menu on the toolbar and then select “Run Configurations”. Select the Arguments tab and ensure that “Automatically add binary container(s) to arguments” is selected.

Following this, the build project can be executed by clicking the play button on the menu bar.



The built program can also be executed from the command line. If executing a *sw_emu* build, you must first type the following:

export XCL_EMULATION_MODE=sw_emu

If executing a *hw_emu* build, you must type the following:

export XCL_EMULATION_MODE=hw_emu

The build project can then be executed by typing, for example:

./EUBinomialTree.exe ./binary_container_1.xclbin

3.1 Binomial Tree

A file called *option_data.txt* is included for the Binomial Trees algorithm. Here, you can enter data for up to 25 options. The price for all options will then be returned at the end in the form of an array.

```
# Option pricing input file
# ------------------------
# You may enter data for up to 25 options, any extra will be ignored
#
# Info
# S - Stock Price       (float) Underlying stock price
# K - Strike Price      (float) Options strike price
# T - Time to Maturity  (float) Time (in years) until the option expires
# D - Dividend Yield    (float) Eg for 5% enter "5" and not "0.05"
# r - Risk-free Rate    (float) Eg for 5% enter "5" and not "0.05"
# v - Volaility         (float) Eg for 20% enter "20" and not "0.2"
# type                  (int)   The type of option, enter "0" for call and "1" for put
# height                (int)   The height of the tree
#
# Structure input data
#  S,K,T,D,r,v,type,height
50,50,1,0,0.05,0.20,1,30000
```

NOTE: If executing a Binomial Tree algorithm from the SDAccel environment, the *option_data.txt* file **MUST** be placed in the relevant build folder. For example, if running for *System* build, place the option data file in the "System" folder. If executing via command line the *option_data.txt* file **MUST** be placed in the root of the projects folder.

3.2 Monte Carlo

For the Monte Carlo simulation, the option data is set in the "host.cpp" file. The data is set between lines 31 and 39. If these data are changed, the host file will be recompiled once you run the project. Assuming no changes have been made to the HLS code, only the host file will be recompiled, not the entire project.

4. References

[1] Carter, M. (2019) COMP702 FPGA MSc Project. Available online: https://github.com/mjcarter95/COMP702-FPGA-MSc-Project

[2] Xilinx (2018) PYNQ-Z2 Setup Guide – Python productivity for Zynq (Pynq) v1.0. Available online: https://pynq.readthedocs.io/en/latest/getting_started/pynq_z2_setup.html [Accessed: 05/06/2019].

[3] Xilinx (2019) Getting Started with Alveo Data Center Accelerator Cards. Available online: https://www.xilinx.com/support/documentation/boards_and_kits/accelerator-cards/ug1301-getting-started-guide-alveo-accelerator-cards.pdf [Accessed: 16/09/2019].