FPGA Proposal: Test Performances for multiple languages on FPGA

The idea is to test different levels of programming languages from high to low level, the goal is python for example which is a high-level language that doesn’t require a lot of memory management. The second language will be C which is mid-level there’s some memory management and the programmer has more control. The lowest level language would be VHDL/Verilog/SystemC which is usually being used to program those kinds of programmable configuration chips.

An Interesting aspect of that would be to see if there’s a lot of overhead with the performance comparing the languages and to compare the time spent on developing such a program. We can discover that VHDL can be very efficient but takes 3 times more than python programming and that has a big effect on the development process. We can find out that high-level languages are more secure or the other way around.

The usages can have an impact on the FPGA society and may change current development processes. if the main open question will show that it’s more beneficial to program in python or any other high-level language, it will open up FPGA’s to a lot of developers and will make it more accessible for them.

Schedule and milestones: Approve the project with professor Kim, read some relevant research paper on the subject to see the latest developments in the field. Decide which ML algorithm will be programmed to run over the FPGA, Getting a working version of the algorithm in each of the three programming languages. Adding performance scales and measurement to validate the assumptions. After we have some preliminary results, the next phase will be to increase the number of algorithms/trying different algorithms. The final step should be adding different FPGAs and testing the testbed on different FPGAs, seeing how the results changes and maybe seeing that a certain company has better performance over the other.

Tips: Synthesis tools for python, fixed point for errors, take a hash, pass data set, time it, static data analysis, highest frequency Fmax, how many LUTs each design used. Timing and space measurement of resources. – It might be hard to learn a hdl language.