Open the correlation project in WebIDE at https://openspl.doc.ic.ac.uk/

The project contains four different versions of the correlation design:

- **Original:** This contains the original C code of the correlation. The code can be modified and will be recompiled automatically when the application is run.

- **Split Control and Data:** This version contains C code that is consists of control and dataflow parts with data reordering. This project is still in C; it shows how the corresponding part of the code will be transformed into a dataflow implementation.

- **Single Pipe:** A dataflow implementation with the dataflow part written in OpenSPL. It uses a single pipeline. This version can be run in simulation and on the DFE. The project shows the OpenSPL source code for the dataflow engine which can also be modified and tested in simulation; however, the DFE binary is precompiled and will not change.

- **Multi Pipe:** An optimised dataflow implementation with multiple parallel pipes. This version can also be run in simulation and on a DFE. The highly optimised dataflow engine code is not shown, only the C code calling the DFE is available. Both, the simulation and DFE binaries are precompiled and can not be changed.

EXERCISE:

1. Run the Original C code for correlating 256, 512, 1024 and 2048 time series for 10, 100 and 1000 time steps. Record the results in a spreadsheet and analyse the results. What influence do the number of time series and number of time steps have on the observed compute time?

2. Run the Split C code, correlating 256, 512, 1024 and 2048 time series for 1000 time steps. Record the reordering time, compute time and total execution time in a spreadsheet. Explain how these values change with the number of time series. Compare to the results from question 1 and try to explain the observation.

3. Run the Single Pipe and Multi Pipe dataflow designs for correlating 256 time series and 10 time steps in simulation mode (SIM). Record the total execution times in simulation mode. How does the multi pipe version perform in simulation compared to the single pipe version? Explain the observation. Also compare to the software implementations in C (questions 1 and 2 above).

4. Run the Single Pipe and Multi Pipe dataflow designs for correlating 265, 512, 1024, 2048, and 4096 time series for 10, 100 and 100 time steps on the DFE. Record the total execution times in a spreadsheet. Compare the compute times of the single pipe and multi pipe designs when running on the DFE. Compare to the compute times in simulation mode and explain the differences. Also compare to the compute time of

the software implementation in C. What is the maximum speedup and when is acceleration beneficial?

Notes:

In case you observe obvious outliers in your measurements, rerun the experiment and try to explain the reason for the variation.

DFE execution requires an initial setup time of approximately 2 seconds for loading the binary. It will remain loaded for 60 seconds and is available for subsequent runs of the same application. This means that an application will run 2 seconds faster on the second run because the binary is already loaded. Consider this in your experiments (i.e. run the application twice on the first run to avoid measuring the setup overhead).

DFE execution is limited to 60 seconds at a time for this exercise. Longer operation will produce a time-out error. This is required to allow your fellow students gain access to the DFE resources with acceptable waiting time. We allocated enough physical DFEs; however, if all students run their experiments at the same time there is a slight chance that a DFE is not immediately available. In this case the job will be queued and you will experience an unpredictable delay. If this is the case, rerun the experiment.