

Online Multi-Tau Autocorrelation for XPCS: An End-to-End Heterogeneous CPU+FPGA Pipeline

Authors:

Abdul Rehman Tareen, Tobias Kenter, Philipp Middendorf, Fabian Westermeier, Artem Saleev, Christian Plessl, Hans-Georg Steinrück

X-ray Photon Correlation Spectroscopy (XPCS) is a vital characterization technique used to probe the structural dynamics of complex materials across a vast range of spatial and temporal scales. For analyzing these dynamics, the standard multi-tau algorithm needs to calculate auto-correlation functions for up to millions of pixels per frame. Modern multi-megapixel detectors run at kilohertz frame rates and produce tens of gigabytes data per second. Ideally, the analysis should run online and concurrently to provide live feedback e.g. indicating how long to keep acquiring for statistical quality or when to stop before radiation damage sets in. This workload is a real-time streaming problem: the aggregate pipeline state of these independent correlators is too large to keep on-chip, and offline post-processing forfeits the live feedback needed to steer scarce beam time.

To this end, we previously implemented the multi-tau correlator [1] on an FPGA by storing each pixel's pipeline context in off-chip HBM: a compute unit iteratively loads a pixel's state, advances its time series through a deeply pipelined dataflow, and writes it back, decoupling the problem size from on-chip memory constraints. Based on an Alveo U280, these designs sustained 1,062 FPS for an Eiger 4M and 8,596 FPS for an Eiger2 500K, both exceeding their respective operational detector limits.

Here, we migrate and scale these designs to a V80 FPGA, building a complete, continuously running application deployed at the DESY P10 beamline. We adopt a CPU+FPGA co-design partitioning work by hardware affinity: the FPGA is a deterministic, streaming co-processor at the ingest point for autocorrelation, while the host handles network reception, masking, decompression, pixel-packing, and post-readout processing (e.g. ROI averaging and normalization, and reduction). The pipeline runs end-to-end: frames stream from the detector through CPU pre-processing into the FPGA, and out to an interactive viewer that regenerates the correlation curves every few seconds for live monitoring. To keep the CPU fully overlapped with the FPGA computation, we optimize the host side with hand-written intrinsics and a dynamic thread balancer mediating between pipeline stages. With these optimizations, the pipeline sustains real-time analysis of multi-megapixel detectors at kilohertz frame rates.