Title: Hough transform Software, Firmware, and Hardware investigation for fast tracking in Phase-II LHC upgrades

100 word Abstract: This study shows a pattern recognition system based on Hough transform implemented on last generations of FPGA families, as the Xilinx UltraScale+. This investigation started from the ATLAS HTT project, and now it is proposed in general for the LHC Phase-II trigger upgrades HEP experiments, especially to those which aim at fast tracking capabilities. We have designed a Hough transform software emulator first, then a firmware synthesizable design has been developed using a low-level HDL description, and finally we have implemented the project into a large FPGA. The software, firmware, and hardware simulations and tests show compatible results.

500 word Summary for TWEPP 2022: We investigated the ability of the Hough transform to recognize input tracks within a large bundle of unrelated inputs by emulating a series of real input patterns, superimposed on unnecessary data, pretending that they originated from a physical tracker detector. In more detail, the study follows a big investigation we recently made to detect tracks of ionizing particles in high-energy physics. Furthermore, we completed the study by implementing the Hough algorithm in frontier Xilinx FPGAs. For this, the design was not only simulated by means of a software (python) emulator but was also validated by an optimized VHDL firmware design, possibly suitable for a large UltraScale+ FPGA. In particular we used the ALVEO U250, the VCU1525, and the FELIX-II (featuring VU9P) development cards. The main advantage of using FPGAs over any other component is their speed and low latency for studying low-resolution and / or incomplete pattern recognition problems in a high-density data environment. In fact, FPGAs guarantee a latency that increases linearly with the incoming data, while other solutions increase latency times more quickly.
The figures and the table below show some results in terms of pattern recognition efficiency. Figure 1 shows the cross-comparisons between the different software-firmware and hardware tools. Figure 2 is a photograph of the test stand, and Figure 3 shows a plot obtained by extracting the candidate tracks from the input clusters that we supplied first to the emulator, then to the VHDL simulator, and finally to the FPGA system. This number of candidate tracks increases as a combination of unwanted input data (pilup and noisy clusters) and physically interesting data. When these two latter parameters are simultaneously high, the number of candidate tracks extracted increases significantly, increasing processing time and latency. The table shows some example numbers for the simulations we ran, with estimates of the total processing time, which is very small compared to software approaches. We use this type of simulation to limit the percentage of acceptable noise and undesired data, depending on the application, to not excessively increase the total latency of the process. This investigation, from the software emulator to the FPGA hardware implementation, has so far permitted us to reduce the FPGA resources in terms of LUT and FF by about a factor of 2. The use of FPGAs that implement the Hough transform in trajectory recognition problems can extend the tracking and triggering capabilities for Phase II upgrade of all LHC experiments. We started this study a few years ago as part of the ATLAS hard tracking trigger project, now closed for reasons independent of the performance of the Hough transform, and now we propose it as a general approach for HEP experiments on LHC Phase-II trigger updates.
In this study we also present the considerations we made to optimize the Hough algorithm to best fit a hardware device, considering the parameters of the ATLAS internal tracker as a realistic example. As a practical conclusion, the algorithm implemented on high-performance FPGA can speed up the experiment tracking apparatus.

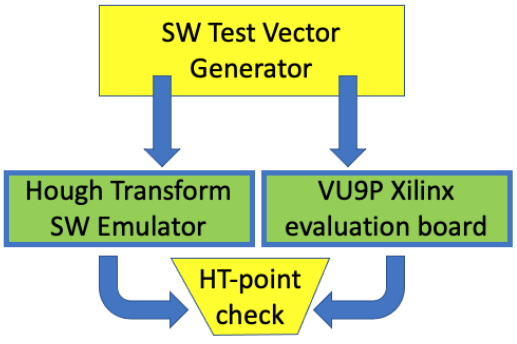

Figure 1

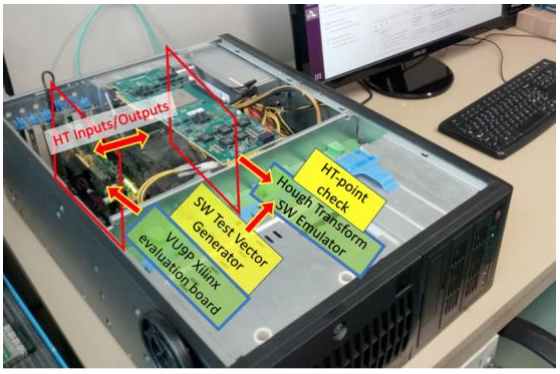

Figure 2

Input Set A	Input Data Pairs	Candidate Tracks	Extracted Candidate Tracks	Processing Time (ns)
A1	928	50	168	2940
A2	1008	55	220	3548
A3	768	40	93	1932
A4	848	45	118	2244
A5	608	30	55	1444
A6	688	35	73	1652
A7	448	20	32	1052
A8	528	25	42	1220

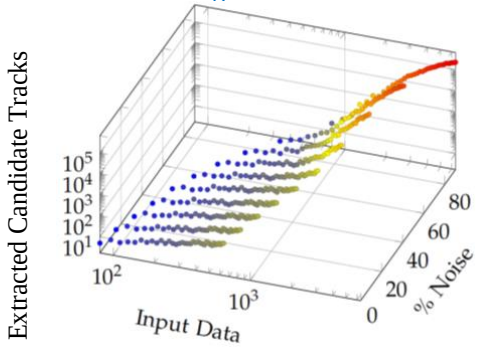

Figure 3