Halide tutorials summary and lessons learned for IPU

Image processing pipelines usually combine several stages of stencil computations and streaming programs. Their complex structure can have numerous execution schedules, each attaining performance that may differ in orders of magnitude. Efficient scheduling decisions involve trade-offs between parallelism, data locality and recomputations, and their implementations require a considerable and error-prone effort of code rewriting. Halide is a domain specific language for image processing introduced in 2014. It is highly adapted to the description of stencil computations and can also support reductions. Halide offers several levels of simplification for the design of image processing pipelines. First of all it separates the pipeline algorithm from its schedule, enhancing, in this way, the algorithm re-use. Being a functional language, Halide explicitly exposes the algorithm intrinsic parallelism reducing the gap between the algorithm specification and the exploration of its implementations. Halide also provides an intuitive and simple mechanism to specify and explore correct–by-construction code optimizations that lead to different implementations. Halide is released with a compiler targeting several platforms among which CPU, GPU and DSP. Using this compiler in combination with a schedule exploration strategy enables programs to rapidly achieve state-of-the-art performance on a wide range of real image processing pipelines, and across different hardware architectures.
We have explored the possibility to use Halide in several phases of Intel Processing Units design. I this presentation we would like to share the lessons learned on Halide and its application to IPU design.
Interested audience is composed of system architects, algorithm, FW and RTL designers.

Rosilde Corvino is a software engineer and project lead at Intel Benelux B.V., Eindhoven since 2014. She is currently leading an effort to evaluate the use of Halide language for the design of Intel Processing Units. From 2011 to 2014, she worked as scientist and project manager within the ASAM project at Eindhoven Technical University, The Netherlands. In 2010, she was a post-doctoral research fellow in DaRT team at INRIA Lille Nord Europe and was involved in Gaspard2 project. She earned her PhD in 2009, from University Joseph Fourier of Grenoble, in micro and nanoelectronics. Her dissertation focused on using loop transformations to enhance the results of C-to-RTL high level synthesis. She owns a double Italian and French M.Sc in electronic engineer. Her research interests involve design space exploration, parallelization techniques, data transfer and storage mechanisms, high level synthesis, application specific processor design for data intensive applications.