FPGA-Based Soft-Core Processors for Image Processing Applications

Paper Link: https://link.springer.com/content/pdf/10.1007/s11265-016-1185-7.pdf

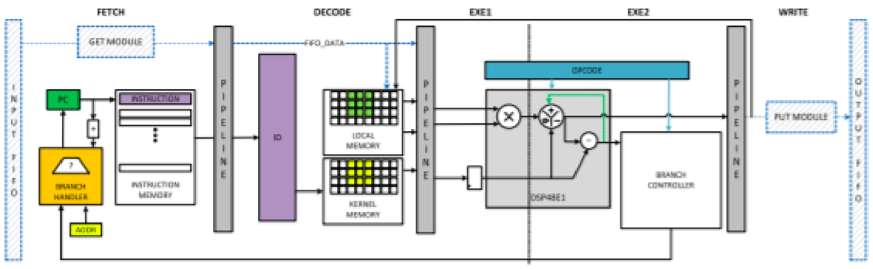
Abstract:

With security and surveillance, there is an increasing need to process image data efficiently and effectively either at source or in a large data network. Whilst a Field-Programmable Gate Array has been seen as a key technology for enabling this, the design process has been viewed as problematic in terms of the time and effort needed for implementation and verification. The work here proposes a different approach of using optimized FPGA-based soft-core processors which allows the user to exploit the task and data level parallelism to achieve the quality of dedicated FPGA implementations whilst reducing design time. The paper also reports some preliminary progress on the design flow to program the structure. An implementation for a Histogram of Gradients algorithm is also reported which shows that a performance of 328 fps can be achieved with this design approach, whilst avoiding the long design time, verification and debugging steps associated with conventional FPGA implementations.

Keywords: Image processing · FPGAs · Heterogeneous multi-core architecture

Summary of Paper:

1) The HDL design flow involves numerous verification and debugging steps, which increases the time to market from weeks to months.
2) Xilinx and Altera have created High Level Synthesis(HLS) tools which allow the designer to use high level languages such as C or OpenCL to create algorithmic representation for FPGA implementations.
3) But still these HLS tools depend on the HDL route to produce bitstream, any changes in the HLS code results in the same long process of verification and validation.
4) In paper they have used a RISC **(Reduced Instruction Set Computing) processor called Image Processing PROcessor (IPPro) [24].**
5) **IPPro architecture:**



6) **IPPro keeps the balance between programmability and the need to maintain the FPGA performance.**
Overall, it has the following addressing modes:
– Local Memory – Local Memory (LM-LM)
– Local Memory – FIFO (LM-FIFO)
– Kernel Memory – FIFO (KM-FIFO)