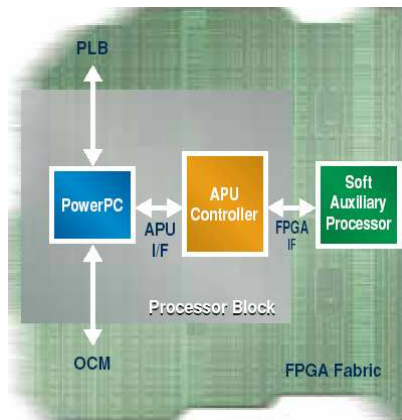


Enhanced Processor Performance through APU Acceleration

Peter Ryser
Xilinx, Inc, San Jose, USA
peter.ryser@xilinx.com

With the increasing computing power of modern microprocessors it becomes feasible to process applications in software that before could only be implemented with dedicated hardware. Using embedded microprocessors in an FPGA and combining them with other advanced features of the fabric through a high-performance and low-latency auxiliary processor unit (APU) controller allows for a powerful solution on a single, reprogrammable chip.

On one side advanced FPGAs embed hard processors directly into the FPGA fabric.



These processors run at frequencies up to 450 MHz and deliver an astounding 700 Dhrystone Mips (DMIPS) at an extreme low power consumption of 0.45 μ W/MHz. On the other side these FPGAs contain a large amount of logic and routing resources as well as specialized DSP functionality.

Other approaches are known to combine the flexibility of hardware in the FPGA fabric with software controlled processors. All of them have disadvantages.

Processors implemented soft in the FPGA fabric provide user defined instructions implemented in FPGA hardware but lack the necessary software

performance. They typically max out at around 200 to 250 MHz and reach about 250 DMIPS. Power consumption to implement the same functionality is up to 120x higher over the solution introduced in this presentation.

Systems accelerating software performance with FPGA fabric attached to the processor buses suffer from high latency, jitter, and problems due to bus arbitration and limited bandwidth. Such an approach is typically used when an FPGA assists a processor chip through a system bus.

Yet other approaches integrate processor and hardware accelerators into ASICs at the cost of impossible design extension, migration, and overall flexibility.

As part of my presentation I will show how the powerful resources of advanced FPGAs can be combined and accessed from the processor through the high-bandwidth and low-latency APU with user-defined instructions at the example of a two-dimensional inverse discrete cosine transform (2D-IDCT). The versatility of this approach will become apparent as part of an accompanying demo.