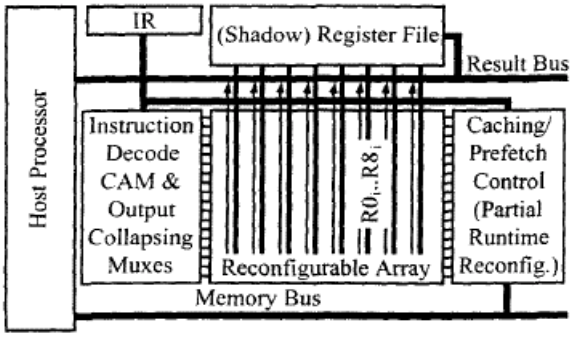
The Chimaera Reconfigurable Functional Unit

Scott Hauck, Thomas Fry,
Matthew Hosler, Jeffrey Kao

Year of publication: *1997*
Area: *Architecture and Technology*



This paper introduced a reconfigurable functional unit that could accelerate critical portions of applications without negatively impacting RISC processor core clock speed or memory bandwidth.

Almost from the introduction of commercial field-programmable gate arrays, it was recognized that reconfigurable logic could be used in conjunction with a standard RISC microprocessor to accelerate certain types of computation. Early FPGA-based coprocessors were generally located on the processor bus, some distance from the processor. These units were often configured for coarse-grained computation due to memory bandwidth constraints between the RISC microprocessor and the coprocessor. The integration of reconfigurable logic as a functional unit inside the RISC processor core met with limited success due to the delay difference associated with reconfigurable logic versus fixed logic. The inclusion of FPGA logic in the processor core limited the clock frequency of the RISC processor, negatively impacting all instructions.

This paper took a fresh approach to addressing both the memory latency and processor performance issues. The Chimaera reconfigurable functional unit (RFU) is located close to the RISC microprocessor, but outside of the primary RISC execution path. The RFU has access to the same registers as the processor via a shadow register file that includes a collection of dedicated, multi-operand ports. This supplemental register file provides fast access to the same data that is available to the processor without restricting the RISC processor operating frequency. The paper describes the possibility of speculatively executing certain reconfigurable operations while the remainder of the RISC code proceeds in a standard fashion. Additionally, unlike earlier FPGA coprocessor offerings, the paper provides a customized LUT-based datapath configuration for the RFU which is optimized for word-based operation. Similar configurations are later seen in other reconfigurable architectures, such as PipeRench.

This paper provided a significant step forward in understanding how to integrate RISC microprocessors and reconfigurable compute units together. Although contemporary commercial efforts in this space (e.g. Stretch) met with limited success, some of the elements of this design style are becoming more popular as compiler technology matures. The ability of a compiler to identify functions that can take advantage of the reconfigurable resource and configure it appropriately is still an active area of research in reconfigurable computing.

Russell Tessier

DOI: http://dx.doi.org/10.1109/FPGA.1997.624608