CoRAM: FPGA Architecture for Computing


Eric S. Chung


James C. Hoe


Ken Mai

Despite their promise in both processing performance and efficiency, today's FPGAs are inadequate as computing devices, particularly due to their lack of a standard native memory architecture. When attempting to develop an application for the FPGA—besides creating the core computation kernels, a designer is burdened by the responsibility of crafting from scratch the application-specific infrastructure logic to transport data between the kernels and external main memory. This infrastructure development not only increases design time and effort but also locks a design to a particular FPGA and platform environment, severely restricting scalability and portability.

Figure 1 offers a conceptual view of the proposed CoRAM FPGA architecture for reconfigurable computing. On one hand, the CoRAM architecture preserves the conventional reconfigurable fabric technology to support highly tuned hardware processing kernels; on the other hand, the mechanisms for data transport between kernels and external memory are supported by native, hardwired datapaths as shown in **Figure 2**, whose details are however hidden from the developer through the CoRAM programming abstraction. Instead, the kernels view external memory only by proxy through local distributed SRAMs. The application developer separately prescribes, using a C-based language, a set of control threads to dynamically manage the data contents of the SRAMs in coordination with the execution of the processing kernels. In addition to improving performance and efficiency, this separation of concerns between processing and data movement allows for a virtualized application environment that simplifies an application's development and improves its portability and scalability.

To demonstrate CoRAM, a functioning prototype generator called CORCC (CoRAM Compiler) has been developed, which automatically generates synthesizable RTL from the CoRAM language into existing FPGA platforms. Application-driven evaluations have been conducted to show the programmability and performance benefits of CoRAM. CORCC will be released at the end of 2011 as an open-sourced tool for evaluation and research.

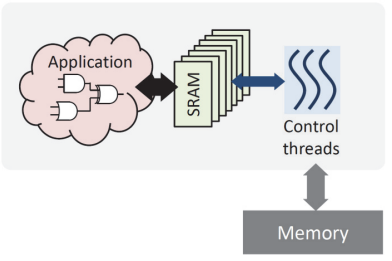

Fig. 1: Application Writer's View of CoRAM.

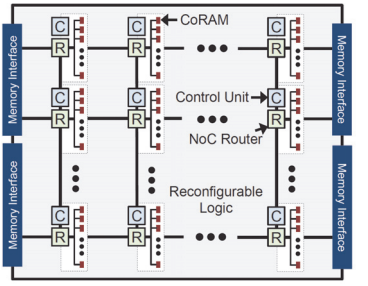

Fig 2: FPGA with dedicated CoRAM support.