FPGAs offer high performance and power efficient computation, but are difficult to use. In particular, the effort involved in managing data movements between on-chip computation components and off-chip DRAM has prevented FPGAs from being widely adopted by the computing industry. Recently developed FPGA programming environments layer simplifying abstractions on top of the DRAM interfaces provided by FPGA vendors, but existing programming environments have primarily focused on support for simple, regular data access patterns such as block copy and streaming.

This thesis proposes CoRAM++, an FPGA programming environment that efficiently supports complex data structures such as multi-dimensional arrays and linked lists in addition to simple data access patterns. CoRAM++ application developers manage data movements through an extensible library of data-structure-specific application-level interfaces, which generate specialized soft-logic datapaths between application components and memory. We show that CoRAM++ allows applications to easily and conveniently manage data transfers without sacrificing DRAM access performance: they can saturate the DRAM interface when streaming data, and match the performance of reference applications when performing more complex memory accesses.

Finally, we show that CoRAM++ allows applications to utilize optimized data structure implementations that accelerate pointer-chasing memory accesses through custom logic attached directly to the memory interface, improving data transfer performance up to 5.2 times.

Thesis Committee:
James C. Hoe, Chair
Kayvon Fatahalian
Kenneth Mai
Todd Mowry
Joel S. Emer, MIT/NVidia