The density advantage of configurable computing

André DeHon

Published: *IEEE Computer, April 2000, Pages 41–49.*

This is a classic paper in Reconfigurable Computing that provides an analytical model to explain the advantages of FPGA hardware. For a community driven by empirical validation and extensive benchmarking, this paper provides deeper insights into the reasons for these advantages. The paper explains how the fixed resource (silicon) budget is allocated to instructions, computation, and data movement functions. It then shows how the FPGA devotes a large fraction of its resources for computation and data movement while extensively exploiting instruction reuse and locality of spatial data flow dependencies. This is in stark contrast to CPUs and other devices that may have large instruction memories and caches that steal resource away from the ALUs (compute).

For a graduate student in the early 2000s like myself and countless industry practitioners, this paper was a validation of the enthusiasm many felt over the promise of reconfigurable technology. The paper articulated with models what many knew through empirical studies. As FPGAs now enjoy wider acceptance, here are some key tradeoffs foreseen by this paper:

(1) *Precision evaluation*: A significant saving in FPGA cost, and a corresponding increase in density is a result of precision customization of the FPGA datapath. Modern FPGA programming tools like Xilinx Vivado HLS, now provide a C++ template library that you can use to easily customize precision of your computation.

(2) *Specialized resources*: The paper discusses embedded specialized function units in FPGAs intermingled with LUTs through the example of a hardware multiplier of DSP block. Modern FPGAs now provide hard DSP and also RAM blocks to further the density advantage of pure LUT-based architectures. The latest Xilinx Versal FPGAs also integrate specialized AI compute engines that take specialization to the extreme. This paper also cautions against overgeneralization and underutilization of such blocks, which may be useful reminders in designing next-generation FPGA architectures.

(3) *Hybrid Compute*: Quoting the 90/10 rule in computer architecture, the paper argues in favor of hybrid processor-FPGA systems that are able to offload the performance-limited 10% portions of the code the FPGA while the rest of the application resides on the CPU. FPGA vendors have an "SoC" offering targeting the embedded domain that fits this hybrid worldview.

Going forward, the key takeaways from this paper will guide FPGA design for the foreseeable future. The need for specialization of architecture to better support ML (Machine Learning) workloads, and addition of hardened NoC (Network-on-Chip) communication blocks for rapid data movement are two recent examples where we can clearly see glimpses of this paper's predictions coming true.

Endorsement by: Nachiket Kapre, University of Waterloo

DOI: https://doi.org/10.1109/2.839320