Today, the market demands increasingly sophisticated application-specific digital hardware architectures, but traditional register-transfer level (RTL)-based design methods remain expensive and time-consuming. High-level synthesis (HLS) offers a promising alternative by enabling designers to specify accelerator functionality in high-level languages (e.g., `C/C++`) and automating hardware generation. Although HLS simplifies design-space exploration (DSE) and functional verification, achieving quality of results (QoR) comparable with manually-optimized RTL still requires manual low-level optimizations that involve implementation-specific details, which extend beyond purely functional descriptions originally intended by HLS. Therefore, hardware design predominantly remains at RTL.

This dissertation argues that broader adoption of HLS is hindered by insufficient abstraction, particularly in the management of low-level details. In addition, many optimization opportunities offered by high-level abstractions remain largely underutilized.

To address these challenges, this dissertation introduces novel methodologies aimed at further raising the abstraction level in HLS designs. Key contributions include the introduction of semi-automated memory management for field-programmable gate arrays (FPGAs), via an open-source caching library that enables throughput comparable with manual on-chip buffering, with minimal engineering effort. Designs optimized with the proposed cache are from 8 × to 113 × faster than those accessing the off-chip memory directly. Moreover, this dissertation presents an automatic optimization for digital signal processor (DSP) utilization through open source compiler optimization passes, which identify superword-level parallelism and pack multiple low-precision operations to single high-precision DSPs provided by FPGAs. The methodology matches manual DSP packing, automatically saving on average 70 % and 50 % DSPs in addition-intensive and multiplication-intensive benchmarks, respectively.

This dissertation also proposes new approaches that take advantage of the high abstraction level of HLS for improving the QoR. Specifically, the task-level multi-pumping takes advantage of the HLS pipelining feature to fully exploit the available timing slack and maximize resource sharing, reducing DSP utilization by up to 40 % at equal throughput.