Supplementary Material

Comparison of Pareto Frontiers between ZCU104 and Alveo U200

Regarding the comparison of the Pareto frontiers between the ZCU104 and the Alveo U200, we now extend the analysis by presenting the distribution of Pareto-optimal designs across five additional QoR metrics. Specifically, we report (a) latency normalized by the target clock frequency on the left, followed by (b) BRAM utilization percentage, (c) DSP utilization percentage, (d) FF utilization percentage, and (e) LUT utilization percentage toward the right. All resource utilization values are expressed as a percentage of the total resources available on each FPGA, enabling a fair and consistent comparison between the two devices.

Figure 1 presents the distributions of the evaluated QoR metrics, revealing clear differences between the Pareto-optimal designs targeted for the ZCU104 and the Alveo U200 devices. For the ZCU104, latency divided by target clock frequency spans from 1.3×10^{-5} µs² to 0.1 ms², with an average of 288.11 µs². Resource utilization on the ZCU104 is also broadly distributed. BRAM usage ranges from 1% to the full 100% capacity, averaging 21.71%, while DSP utilization varies from 1% to 93% with a mean of 5.86%. Similarly, FF and LUT utilization span from 1%–92% and 1%–94%, with averages of 7.49% and 17.19%, respectively. In contrast, Pareto-optimal designs on the Alveo U200 exhibit more compact and resource-efficient behavior. The latency/frequency metric remains significantly lower, ranging from 8×10^{-6} µs² to 1649 µs², with a mean of only 31.25 µs², almost an order of magnitude smaller than that of the ZCU104. Resource utilization follows a similar trend. BRAM usage is tightly bounded between 1% and 15%, averaging 1.67%, while DSP usage ranges from 1% to 41%, with a mean of 2.97%, roughly half of the ZCU104 average. FF and LUT utilization are also modest, spanning 1%–46% and 1%–31%, with mean values of 2.60% and 4.48%.

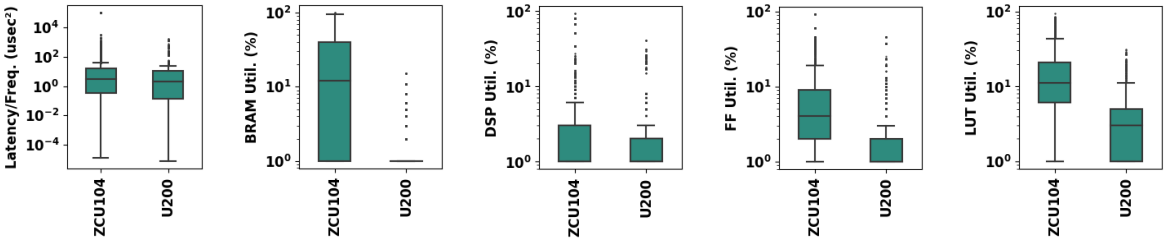

Figure 1. QoR Metrics for Pareto-Optimal Designs on ZCU104 and U200 at 100 MHz

Switching the Optimization Goal: Latency- VS Resource-Efficient Designs

In this part of our analysis, we examine how the optimality target influences the QoR distributions. Specifically, for each Pareto frontier, we identify two key designs: the one with the lowest resource consumption, referred to as Resource-Efficient (RE), and the one with the shortest execution time, referred to as Latency-Efficient (LE). **Figure 2** presents the distribution of the examined QoR metrics for both RE and LE, while also including the

distribution when considering all design points (ALL). This analysis is conducted for the MPSoC UltraScale+ ZCU104.

For the speedup distribution over Vitis, the RE designs exhibit a range from 1.2 x 10^(-4) to 200, with an average speedup of 9.1. In contrast, the LE designs show a distribution, ranging from 0.001 to 54103.2, with an average speedup of 377.6. As expected, the LE designs achieve higher speedup values, with an average speedup that is 41.3x greater than that of RE. The distribution for ALL design points results in an average speedup of 183.9, which falls between the average values of RE and LE, as anticipated. A similar trend is observed in the time required for the HLS process, where LE designs exhibit an average execution time that is 4.4x higher than RE. This increase is attributed to the greater resource consumption of LE designs.

A closer examination of resource utilization reveals the expected trend, where the BRAM, DSP, FF, and LUT usage in the RE designs is, on average, lower than that of the LE designs. Specifically, for BRAM utilization in the RE designs, the distribution ranges from 6 to 599, with an average of 79 units. For DSPs, the utilization ranges from 17 to 69, averaging 19. FF utilization varies between 4.6K and 28K, with an average of 7.2K. Similarly, LUT utilization spans from 2.3K to 78K, with an average of 9.8K. On average, LE designs exhibit higher resource utilization compared to RE designs, with increases of 1.47x for BRAMs, 9.1x for DSPs, 9.2x for FFs, and 6.3x for LUTs. An interesting observation is that while the average values for DSPs, FFs, and LUTs fall between those of RE and LE as expected, the ALL distribution shows a 1.19x higher speedup for BRAM utilization compared to LE. This behavior is primarily attributed to the use of complete partitioning directives, which increase bandwidth and often result in highly performance-efficient designs when they fit within the FPGA. As a result, memory arrays are implemented using FFs, leading to lower BRAM utilization compared to the ALL distribution.

Empirical Summary: Latency-efficient designs deliver much higher speedup than resource-efficient ones, but this improvement comes with a significant increase in resource consumption.

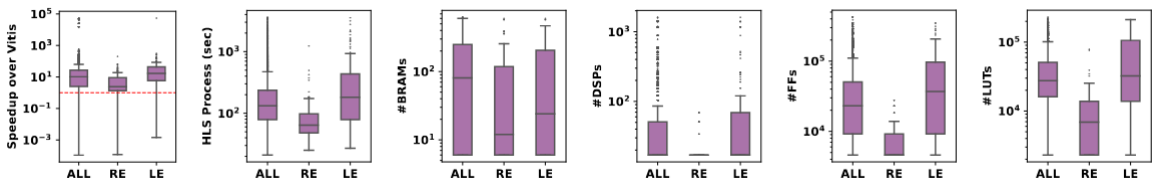

Figure 2. QoR Metrics for Full Pareto Front (ALL) Resource-Efficient (RE), and Latency-Efficient (LE) Designs on ZCU104

Analyzing Feature Importance on Performance

To address the reviewer's comment, we performed an additional analysis on the Pareto-optimal designs targeting the UltraScale+ ZCU104 FPGA at 100 MHz. Our goal was to investigate how different action points in the source code influence performance, providing users with guidance on which source code structures deserve priority during exploration. The results are presented in **Figure 3**. The horizontal axis lists all action points identified across both loop and array, while the color intensity reflects each action point's impact on speedup, with darker shades indicating a stronger influence. To properly interpret **Figure 3**, it should be examined alongside Figures 6A and 6B from the manuscript, which provide further insights into the characteristics of the action points. Figure 6A illustrates the distribution of array sizes associated with each array-related action point, while Figure 6B presents the distribution of loop tripcounts for the loop-related action points. As a reminder, the array action points are displayed in descending order of array size, whereas the loop action points follow the order in which they appear in the source code.

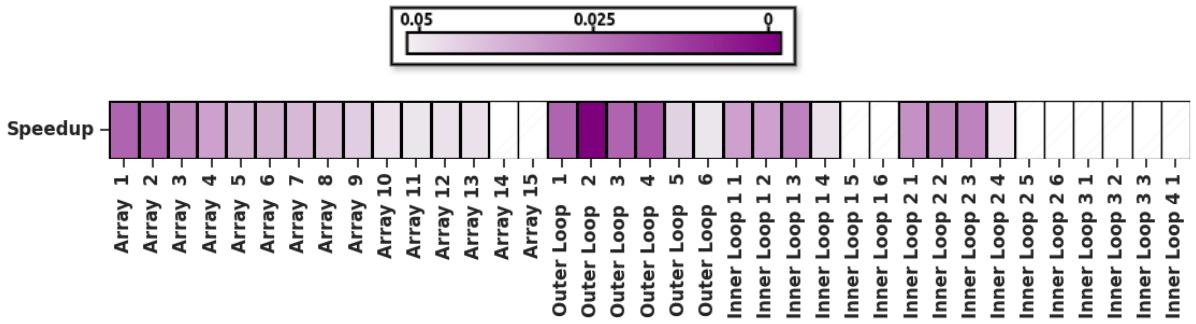

Figure 3. Impact of Action Points on Performance

Focusing on the array action points, we observe that Array 1 through Array 4 exhibit the strongest influence on performance because they correspond to the largest data structures in the design, and applying directives to these arrays directly affects memory bandwidth and data-access parallelism, which are key determinants of overall performance in HLS-generated hardware. As we move toward action points associated with smaller arrays, the impact on performance diminishes, indicating that optimizations applied to less frequently accessed or smaller data structures provide limited benefit. A similar trend is observed in the loop action points, where directives applied to outer loops with high trip counts lead to substantial performance gains since they dominate the execution time, while loops with lower trip counts exhibit progressively smaller influence because accelerating them affects only a small portion of the total latency. Additionally, for the two outer loops with the highest impact, their corresponding inner loops also show notable influence, suggesting that optimizing both levels of the loop hierarchy can jointly produce meaningful performance improvement when appropriate HLS directives are applied.