Abstract—Accelerator-coupled systems have been introduced as a promising architectural paradigm that can boost performance and improve power of general-purpose computing platforms. This research focuses on the accelerators' scalability problem due to resource under-utilization in FPGA-based accelerator-coupled platforms. By recognizing that static memory allocation—the de-facto memory management mechanism supported by modern design techniques and synthesis tools forms the main source of memory-induced under-utilization, i.e., leading up to 75% of dark silicon, we propose the development of a) a Single-Chip Many-Accelerator (SCMA) architecture that reduces energy budget by providing high-throughput processing nodes hooked under the same low-latency FPGA die and b) a novel design framework that extends conventional RTL and High Level Synthesis (HLS) design flows with dynamic memory management (DMM) features to leverage scalability by enabling accelerators to dynamically adapt their allocated memory to the runtime memory requirements, thus maximizing the overall accelerator count through effective sharing of FPGA's memories resources. By applying these novel techniques in the state-of-art Vivado-HLS tool, we managed to increase accelerator density up to 3.8× for a Xilinx Ultrascale device and deliver architecture solutions that trade-off per-accelerator latency overhead (1.2× - 19.9×) with overall system's throughput (2.6× - 23.1×) and performance-per-watt (0.69× - 21.7×).

I. MOTIVATION & RELATED ART

The next big challenge in computing systems has been recently identified as the so-called “Breaking of the exascale barrier” [1]. Reaching this goal however, requires a design paradigm shift towards more aggressive hardware/software co-design architecture solutions. Towards this direction, many-accelerator heterogeneous architectures have been proposed to overcome the utilization/power wall [2], [3]. From an electronic design automation (EDA) perspective, high-level synthesis (HLS) tools are expected to play a central role in enabling effective design of many-accelerator computing platforms. Raising the design abstraction layer, designers can now quickly evaluate the performance, power, area and cost requirements of a differing accelerator configurations, thus providing controllable system specifications with reduced effort over traditional HDL-based development flow.

Heterogeneous FPGAs are proven to form an interesting platform solution for many-accelerator architectures. However, in such diverse and large pool of accelerators the memory organization forms a significant performance bottleneck, thus a carefully designed memory subsystem is required in order to keep accelerator datapaths busy [4], [5]. In this research, we show that in modern high-end FPGA devices, the main limiting factor of scaling the number of accelerators is the starvation of the available on-chip memory. This leads in severe resource under-utilization of the FPGA. In fact modern FPGA design tools (both at the RTL or HLS-level) allow only static memory allocation, which dictates the reservation of the maximum memory that an application needs, for the entire execution window. While static allocation works fine for a limited number of accelerators, it does not scale to a many-accelerator design paradigm.

Figure 1 shows a study on the resource demands when scaling the number of parallel accelerators implementing the $K_{\text{means}}$ clustering algorithm on a number of FPGA devices. As shown, the on-chip memory (BRAM type) is the resource type that starves faster, exhibiting this consistent behavior across all the examined FPGA devices. As expected the size of the on-chip memory has a significant impact on the maximum number of allocated accelerators, ranging form 41 for the Ultrascale down to 1 for the Zynq device. A similar behavior, regarding the power budget limitation (characterized as “Dark Silicon”) of modern chips is captured in Figure 2, which is annotated with two threshold values regarding to maximum resource count and maximum power budget (device TDP=125°C), respectively. Considering an ambient temperature of 50°C, the power-induced Dark Silicon manifests itself with an allocation scenario of 105 accelerators consuming around 20 Watts. As shown, memory induced “Dark Silicon” poses a stricter constraint in accelerators count, i.e. up to 2.5× less accelerators [6]. The same behavior on resource demands is evident in all the examined benchmarks. In this research, we propose to alleviate the “resource under-utilization” problem by eliminating the pessimistic memory allocation forced by static approaches. Specifically, we propose the adoption of dynamic memory management (DMM) during HLS for the design of many-accelerator FPGA-based systems, to enable each accelerator to dynamically adapt its allocated memory according to the runtime memory requirements.

II. METHODOLOGY AND ARCHITECTURE FOR SCMA

The proposed design and verification flow supporting Single-Chip Many-Accelerator architectures is based on Xilinx Vivado-HLS, a state-of-art and industrial strength HLS tool. A source-to-source code modification stage is the step where the original code is transformed from statically allocated to dynamically allocated using specific function calls from the proposed DMM-HLS API. For the targeted many-accelerator systems, the static-to-dynamic-allocation code transformation is performed on data structures found within the global scope of the application. The transformed code is augmented by the DMM-HLS function calls and it is synthesized into RTL implementation, via the back-end of Vivado HLS tool.

A key performance factor of the system is the ability to feed accelerators with data so that no processing stalling occurs due to memory read/write latency. Moving from static to dynamic allocation using an aggressive approach could lead to the allocation of all BRAMs under the same memory module so that all allocation/deallocation requests occurs on this unique module. This implies serialization bottleneck in
This research targets the scalability issues of modern many-accelerator FPGA systems by proposing the SCMA architecture. The proposed approach has been extensively evaluated over real-life many-accelerator architectures targeting to emerging applications, showing that its adoption delivers significant gains regarding to accelerators density and throughput/performance-per-watt improvements.

IV. CONCLUSION

This research targets the scalability issues of modern many-accelerator FPGA systems by proposing the SCMA architecture. The proposed approach has been extensively evaluated over real-life many-accelerator architectures targeting to emerging applications, showing that its adoption delivers significant gains regarding to accelerators density and throughput/performance-per-watt improvements.

REFERENCES


