ABSTRACT

High Performance Computing (HPC) is suffering from power or performance related issues such as elevated energy consumption, low processing efficiency and memory bottlenecks. This has opened the door to the use of alternate platforms like Field Programmable Gate Arrays (FPGAs) for high-end computing yet also with significant challenges: low operating frequencies, development complexity, low floating point capacity, etc.

Through a thorough review of State of Art works on CPU/GPU/FPGA along with studies on algorithm needs and optimization techniques, this thesis first presents a classification of a widely adopted list of HPC algorithms oriented to their suitability for FPGA implementation. We also include guidelines for better developments on reconfigurable logic in regard with HPC requirements. Secondly, we explore a variety of design methodologies and tools as well as the Dynamic Partial Reconfiguration (DPR) technology. We propose a mixed design approach to conceive efficient FPGA architectures and produce memory oriented enhancements for DPR. Finally, we validate our findings by creating processing systems that concentrate on maximizing scalability for algorithms with irregular memory accesses and low-level arithmetic computation. We use the Knapsack 0-1 and the Flow-Shop Scheduling as case studies and analyse their performance on Xilinx SoC devices composed by ARM cores and FPGA logic.