

Title	[課題研究報告書] 高位合成を用いたQR法による固有値計算のFPGAアクセラレーション
Author(s)	浦本, 真人
Citation	
Issue Date	2022-09
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/18063
Rights	
Description	Supervisor:井口 寧, 先端科学技術研究科, 修士(情報科学)

Abstract

Finding eigenvalues and eigenvectors have a wide range of applications in the natural sciences, engineering, social sciences, and data science. Generally, highly parallel processors such as supercomputers and GPU are used. However, since these architectures also need to ensure the generality of the calculation contents, they cannot necessarily be configured optimally for each process.

FPGA(Field Programmable Gate Array) is characterized by their ability to flexibly configure dedicated circuits and to increase the efficiency of hardware (HW) processing. When the objective is to accelerate specific processing, it is called domain-specific computing, and FPGA are considered to be the most suitable device due to their flexible and re-configurable of circuits.

In general, however, implementation in FPGAs takes a long time to design. One of the reasons for this is that until now, implementation of FPGA acceleration required circuit descriptions in HDL (hardware description language), which made the cost of implementation and verification high.

In recent years, the development of high-level synthesis technology has made it possible to easily convert abstract descriptions such as C/C++ into Hard Ware (HW). In order to use high-level synthesis in this study, we adopted SDAccel, an OpenCL-based platform provided by AMD (Xilinx), a major FPGA vendor.

The results of the evaluation showed that the Jacobi method only achieved a 0.5 to 1.8X speed-up from the baseline circuit, and a 0.1 to 0.6X speed-up from the CPU.

In the QR method, it was confirmed to operate 2 to 180 times faster than the baseline circuit and 0.16 to 12 times faster than the CPU, depending on the matrix size.

As an accelerator, the Jacobi method was inferior to the CPU. The QR method outperformed the general-purpose CPU for limited matrix sizes ($N=16 \sim 64$).