D20-022 **DESIGN GROUP**

作品名稱

符合 OpenCL/TensorFlow API 規範的通用繪圖處理器

A GPGPU Which Supports Both OpenCL and TensorFlow Framework

隊伍名稱

風和日瀝 X-FORCE

隊長

王昱翔 成功大學電腦與通信工程研究所

隊員

周沛辰 成功大學電腦與通信工程研究所
張峻豪 成功大學電機工程研究所
林聖堯 成功大學電機工程學系

作品摘要

近年來人工智慧、機器學習領域快速發展，資料處理量大幅提升，通用型繪圖處理器（GPGPU）開始被廣泛運用於需要計算大量資料且可高度平行化處理的應用上。談到 GPU 大家總會想到 NVIDIA、AMD、ARM 等大型國外公司，鮮少聽到有國內廠商自行開發的 GPU，處理器是一種需要長期努力耕耘的重要基礎技術，不做或做不持續就永遠缺乏。本作品 CASLab GPU 是由成功大學電機工程學系 Computer Architecture and System Laboratory 的師生自 2013 年起規畫研製，從軟體端到硬體端完整的系統開發，目標在打造出第一顆國內自己的 SIMT 運算型 GPU。

CASLab GPU 是以 Edge Computing 為目標，並符合 OpenCL/TensorFlow API 規範，建構包含軟體及硬體的整體系統。這包括根據 OpenCL 規範設計 CASLab GPU 的 Runtime，更自行開發了 CASLab GPU 的 OpenCL LLVM Compiler。透過優化的編譯流程，使軟體堆疊更能配合硬體的運作，獲得大幅整體效能之提升，提供開發人員便利的開源執行環境。軟體層無論是 OpenCL Runtime、Compiler 都是由 C 程式開發，CASLab GPU 無論是搭配 ARM、RISC-V CPU 都能夠在其平台上運作與進行應用開發。

CASLab GPU 透過電子系統層級（Electronic System-Level，ESL）的 Full System 設計方案，軟體與硬體設計能在早期開發即進行系統驗證。利用 C 與 C++ 所實作的指令級模擬器（Instruction Set Simulator，ISS）可驗證指令集的功能正確性並提供時間模型（Timing Model）來做效能上的初步評估。而使用 SystemC 高階硬體描述語言則提供彈性的硬體設計方法，因為是 Cycle-Accurate 行為，開發者在早期階段就能夠更準確的分析能達到的效能，也能作為後續 Verilog RTL 的實作範本。CASLab GPU 已在 FPGA 層級完成功能性的驗證，持續的優化將讓 CASLab GPU 成為一顆更高效能的 Edge Computing IP，持續做是我們的信條。

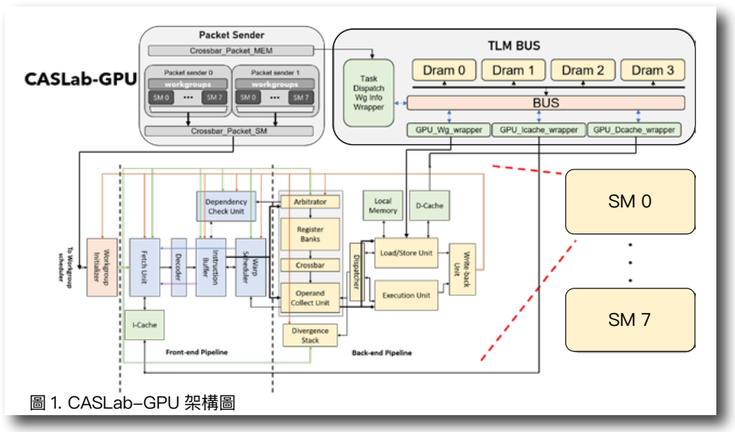

圖 1. CASLab-GPU 架構圖

指導教授

陳中和 成功大學電機工程學系

- 美國西雅圖華盛頓大學電機博士，現為成功大學電機工程學系教授。
- **研究領域：**計算機架構、SOC 整合、VLSI 晶片設計、多重處理器系統、容錯處理系統、AI 加速器系統設計

邱瀝毅 成功大學電機工程學系

- 美國普渡大學電機與計算機工程博士，現為成功大學電機工程學系副教授。
- **研究領域：**超大型積體電路電腦輔助設計、低功率超大型積體電路設計、可重新規劃電路系統

Abstract

For a long time, our local industry didn't make continuous effort on GPU or GPGPU development and they typically out-source the GPU IP from well-known providers. However, GPU processor IP is a critical fundamental industrial technology which needs long-term effort to develop.

The aim of our work is to design and implement a GPGPU which conforms with the APIs of both OpenCL and TensorFlow framework for edge AI computing.

The CASLab GPU IP is developed in ESL (electronic system level) design methodology. GPU software and hardware design can be verified in the early development stage. To enable TensorFlow API in the CASLab GPU, establishing a complete software stack is necessary. Fig. 2 is the software stack of the CASLab GPU. It includes the TensorFlow enabling technology, TensorFlow runtime, OpenCL runtime, LLVM OpenCL compiler, and HSA runtime.

The CASLab GPU uses HSAIL-lite ISA and has its own LLVM OpenCL compiler which is currently implemented in 15,000 lines of C code in the LLVM infrastructure. The original CASLab GPU compiler is AMD CLOC with a finalizer. However, it is not specifically designed for our hardware, which leads to a performance bottleneck. The LLVM OpenCL compiler we build along with the CASLab GPU hardware has effectively increased the obtained performance of the CASLab GPU when compared with state-of-the-art commercial machine of the similar specification.

Fig. 3 illustrates the GPU programming paradigm between the CASLab GPU and Nvidia GPU. The top layer is the input format of data. The second layer is different compiler in each system and the third layer is the ISA that three systems use, respectively.

Our work, CASLab GPU, is now an FPGA functionally verified design for OpenCL data-parallelism applications e.g. Polybench, Search algorithms etc. as well as TensorFlow CNN applications. In the future, we will make continuous effort on optimizing hardware architecture, perfecting OpenCL API, supporting RNN and more AI models.

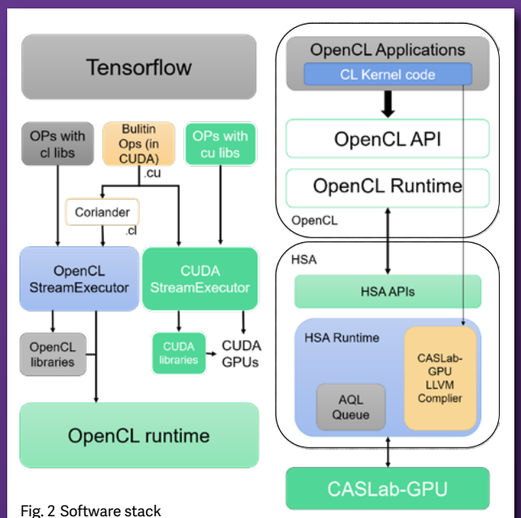

Fig. 2 Software stack

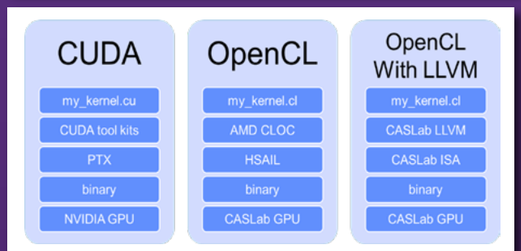

Fig. 3 GPU programming comparison diagram