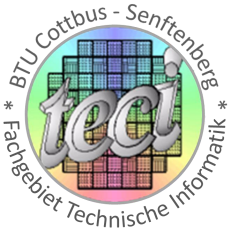
Brandenburg
University of Technology
Cottbus - Senftenberg



Xilinx Open Hardware 2021 design contest

Project name:
FGPU: An Configurable Soft-Core SIMT Accelerator

Team participants:
Hector Gerardo Muñoz Hernandez
Liliia Kudelina
Mitko Veleski

Supervisor:
First: Michael Hübner,
Second: Marcelo Brandalero

University:
Brandenburg University of Technology Cottbus - Senftenberg

1. Introduction:

Machine Learning (ML) is a subtopic of Artificial Intelligence (AI) that significantly grew in importance in recent years [1]. Especially in the domain of embedded computing, ML is of great interest due to low latency, data privacy, and restricted bandwidth concerns. CNNs have proven to be an efficient way of handling ML tasks because they can achieve high accuracy and frequently outperform traditional AI approaches [2]. Due to their flexibility, CNNs have been widely used in object detection and classification, autonomous driving, and drone navigation [2].

CNNs require high computational density and are thus often executed in GPUs and FPGAs, both of them having different benefits like parallel execution and programmable hardware [3]. The GPUs are known for offering high performance for parallel computation at the expense of high energy consumption. On the other hand, FPGAs are known for their increased energy efficiency, but programming them requires hardware knowledge, and it is more time-consuming than its counterpart. Even with tools such as High Level Synthesis (HLS), where higher level programming languages like C or C++ can be used to describe the application and be semi-automatically mapped onto a hardware implementation, the designer still has to bare in mind several information on the hardware structure of the target in order to write an efficient code.

Other than FPGAs and GPUs, a third option for CNN implementations is an overlay processor implemented into an FPGA and programmed via software. The FGPU [4] is such an overlay architecture that implements a microarchitecture similar to that of a GPU. The FGPU allows saving space compared to the FPGAs because it can be reused for more than one application. It is also less time consuming to program than an FPGA, as the hardware doesn't need to be re-programmed for every new application.

1.1 FGPU:

A more detailed description of the FGPU architecture, presented in Figure 1. The FGPU [4] is an open-source 32-bit multi-core GPU-like processor based on the Single Instruction Multiple Thread (SIMT) execution model. The FGPU has its own instruction set architecture, which is composed of 49 MIPS-like instructions inspired by the OpenCL execution model. The translation from a high-level OpenCL code is performed using a dedicated LLVM-based compiler. Another significant feature is that FGPU supports floating-point operations.

The number of Compute Units (CUs), the equivalent of a Streaming Multiprocessor in the other GPU designs, is fully customizable. In the Xilinx ZC706 board, up to 8 CUs can be fit, each consisting of 8 processing elements (PEs). A single CU can run up to 512 work items (threads). Each work item owns a private memory of 32 registers that can be extended using scratchpad memories. Additionally, an off-chip memory size can also be customized and in this work it was limited to 4 GB can be accessed by any working item. The FGPU includes a direct-mapped, multi-ported, and write-back cache system that can simultaneously serve multiple read/write requests. Finally, the FGPU is interfaced and controlled over the AXI4-lite bus.

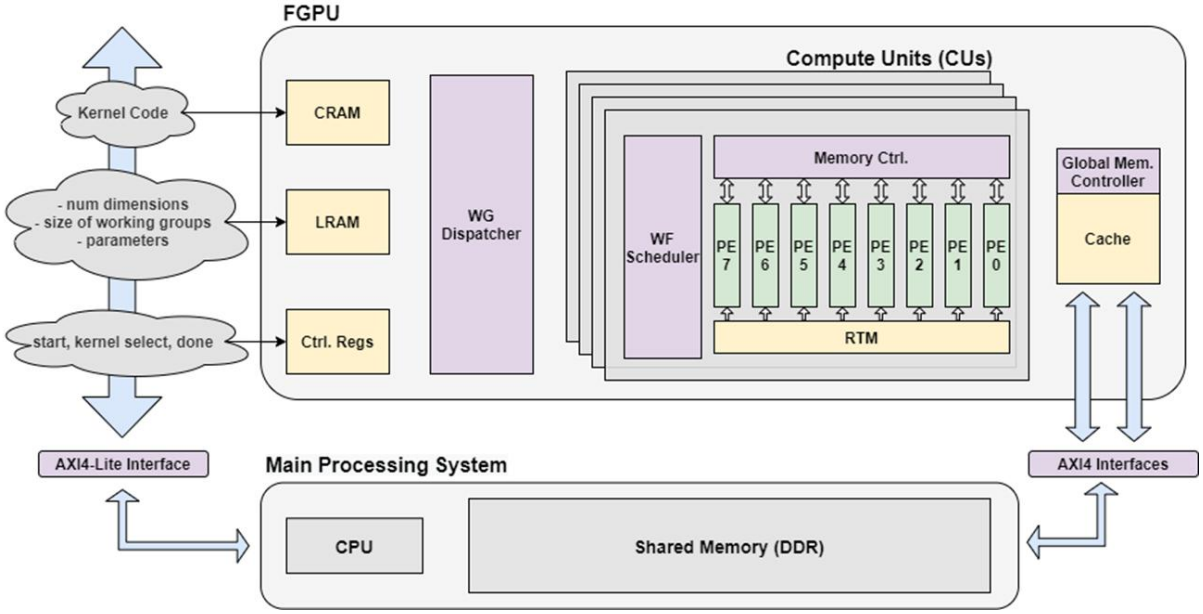


Figure 1. The soft-core FGPU used in this work.

2. Methodology

We showcase a static configuration of the FGPU running in a Zedboard: a standard Xilinx board featuring a ZYNQ-7000 System on Chip (SoC). The FGPU configuration uses 1CU, is running at 100MHz, and has floating point support directly on hardware.

We implemented each layer of the CNN as a separate kernel in OpenCL and compiled them using the LLVM compiler.

On top of this, we created a Vitis project that uses the bitstream containing the FGPU, and ran a simple CNN, capable of recognize handwritten digits, and using the MNIST data-set, it consists of one convolutional layer, one maxpooling layer, one Fully Connected layer (FC), and an output layer.

The input required by this application is a 28x28 grayscale image. Furthermore, our application needs this array of pixels to be flatten into a one dimensional array. We provide a header file (image.h) with the arrays for some digits, (where the user has to comment/uncomment for its use) for testing purposes, but the user can use his/her own images, provided that the pixel values get flatten to one dimensional array.

3. Conclusions

Through the project, our contributions can be listed as:

- A set of scripts that generates a bitstream for the Zedboard using the FGPU overlay.
- A vitis project which includes a convolutional neural network running that identifies handwritten digits.

Furthermore this “base”FGPU version can also be customized for different applications

4. Bibliography

[1] Dean, J., Patterson, D. & Young, C. A New Golden Agein Computer Architecture: Empowering the Machine-Learning Revolution.IEEE Micro38,21–29.ISSN:0272-1732. https : / / ieeexplore . ieee . org / document /8259424/ (2018) (Mar. 2018).

[2] Venieris, S. I., Kouris, A. & Bouganis, C.-S. Toolflowsfor Mapping Convolutional Neural Networks on FPGAs:A Survey and Future Directions.ACM Comput. Surv.51.
ISSN: 0360-0300. https://doi.org/10.1145/3186332 (June2018).

[3] Nurvitadhi, E.et al. Can FPGAs Beat GPUs in Ac-celerating Next-Generation Deep Neural Networks?inProceedings of the 2017

ACM/SIGDA International Sym-posium on Field-Programmable Gate Arrays(ACM,Monterey, California, USA, 2017), 5–14.ISBN: 978-1-4503-4354-1. http : / / doi . acm . org / 10 . 1145 / 3020078 .3021740.

[4] Al Kadi, M., Janssen, B. & Huebner, M.FGPU: AnSIMT-Architecture for FPGAsinProceedings of the2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays(Association for ComputingMachinery, Monterey, California, USA, 2016), 254–263.ISBN: 9781450338561. https://doi.org/10.1145/2847263.2847273.