CornellEngineering
Electrical and Computer Engineering

Translation of CNN Model for Hardware Acceleration

Authors: Nikhil Mhatre, Devin Singh, Junze Zhou Advisor: Hunter Adams

Software Implementation of CNN

Convolutional Neural Networks (CNNs) allow us to process raw digital information and transform it into actionable knowledge. Currently, general-purpose CPUs are commonly used as a software platform for running inference with CNNs due to the simplicity of development with common programming languages such as C++ and Python.

CNNs perform highly repetitive and computationally intensive calculations, which specialized hardware can take advantage of for better performance.

We believe that an implementation of CNNs using a hardware description language (HDL) on a Field Programmable Gate Array (FPGA) can result in faster prediction times while maintaining accurate results.

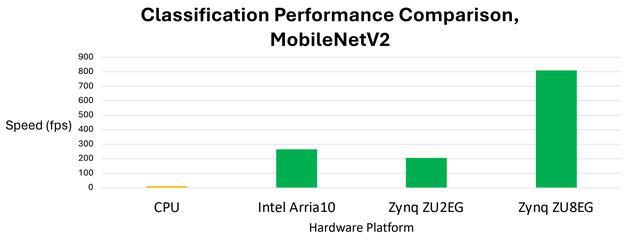
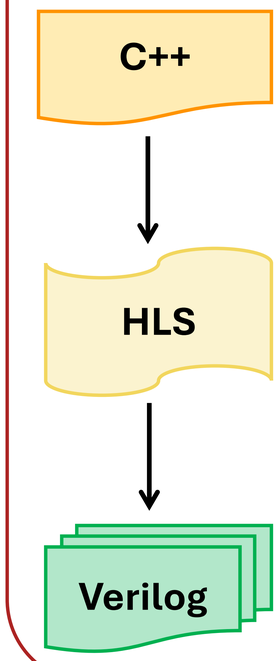


Figure 1. CPU and FPGA FPS comparison [1]

High-Level Synthesis: Translating C++ to Verilog



High-Level Synthesis (HLS) provides a mechanism for automating the translation from C-level programs to hardware description languages.

Optimization directives like Pipelining, Unrolling, and Array Partitioning can be included within the C-level program to improve FPGA performance.

HLS will then generate synthesizable RTL that takes advantage of the FPGA's parallel architecture based on the implemented optimization directive.

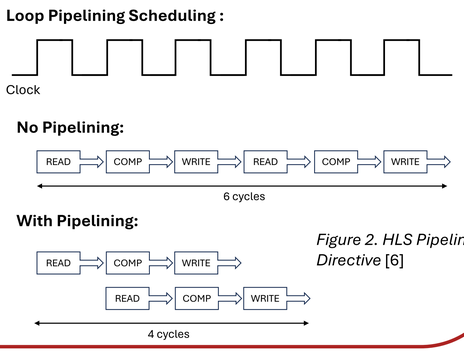


Figure 2. HLS Pipelining Directive [6]

Translation Infrastructure

Translating C++ programs to an equivalent Verilog representation with HLS is not easily done. Certain memory, loop, and data communication optimizations must be made to ensure that the translated program efficiently utilizes resources on the destination FPGA board.

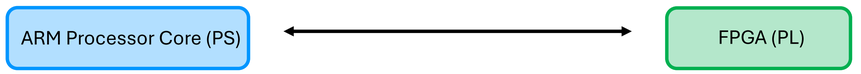
Translations Solutions:

Analyzing the C-level code for parallelization opportunities allowed us to exploit pipelining directives to reduce latency within loops.

Parallelization comes at the cost of increased resource utilization. The limited resources on a FPGA needs to be taken into consideration when increasing the degree of parallelism.

Converting floating to fixed-point representation allows for higher efficiency hardware inference because less hardware resources are needed to represent data.

Organizing the data flow from the ARM to the FPGA by packing four 8-bit data for transmission optimizes communication for better performance.



CNN Platform Translation

The **FPGA** hardware is programmed with a bitstream file generated from the HLS translation. Applications hosted on the FPGA hardware writes and reads data from the application FIFOs.

The host **ARM** processing system, makes Direct Memory Access (DMA) requests via AXI communication to the Xillybus IP core. Upon receipt of requests, the IP core will write or read to the respective application FIFO.

The **Xillybus IP core** creates a seamless communication between the ARM processing system and the FPGA programmable logic. This allows for computations to be isolated, performed on the FPGA fabric, and the output results to be displayed using a Linux interface hosted on the ARM.

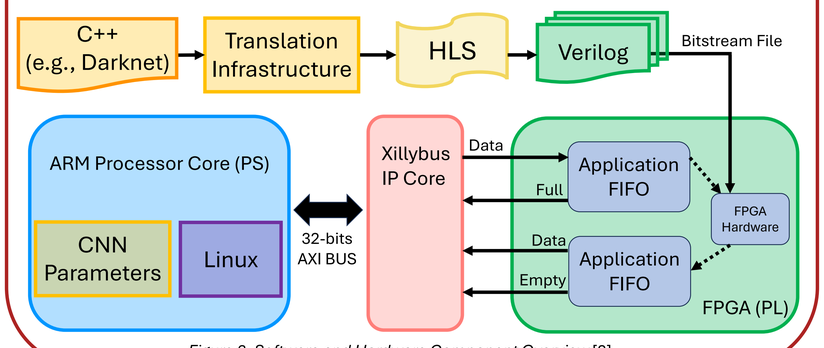


Figure 3. Software and Hardware Component Overview [2]

Hardware Acceleration Case Study: Darknet

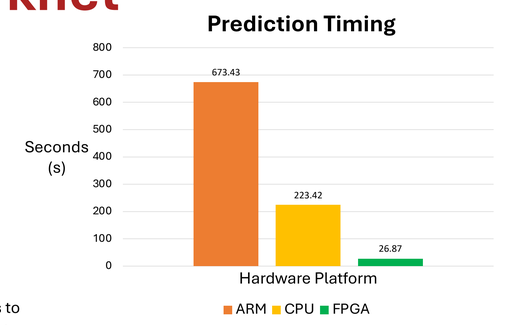
Figure 4. Input Test Image [3]

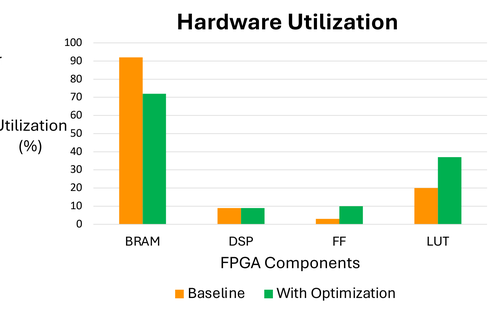
The Tiny Darknet network for image classification highlights the benefits of running a CNN on a FPGA platform.

The network is originally written in C++ and uses a series of layers to process the input image, perform computation, and feed the next layer with a feature map. [5]

Each layer has been pretrained offline with set parameters used for processing the image and providing an output prediction.

Inferencing Accuracy		
	Baseline (confidence %)	With Optimization (confidence %)
Hummingbird	**62.18**	**50.20**
Banded gecko	3.23	2.75
Vase	2.66	2.59
Dragonfly	1.97	2.36
Hair slide	1.46	1.87





CNN Acceleration Achievement

- We successfully built a process that uses High-Level Synthesis to translate a CNN model written in C++ to Verilog.
- Our work demonstrates the advantages of implementing computationally intensive algorithms on a FPGA.
- Looking ahead, CNN deployment on specialized hardware such as GPUs is a viable solution for further enhancing the performance of CNNs while keeping development time low.

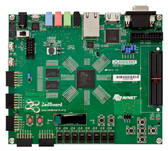


Figure 5. Zedboard FPGA [4]

References

[1] D. Wu et al., "A High-Performance CNN Processor Based on FPGA for MobileNets," 2019 29th International Conference on Field Programmable Logic and Applications (FPL), Barcelona, Spain, 2019, pp. 136-143, doi: 10.1109/FPL.2019.00030. keywords: {Engines;Convolution;Standards;Field programmable gate arrays;Schedules;Acceleration;Computational modeling;convolution neural network;FPGA;hardware accelerator;MobileNet},

[2] "Getting Started with Xillinux for Zynq-7000 v2.0." Documentation, xillybus.com/downloads/doc/xillybus_getting_started_zynq.pdf. Accessed 22 Apr. 2024.

[3] Kulvete, Brian. Ruby-throated Hummingbird Archilochus colubris. 3 May 2018. https://macaulaylibrary.org/asset/97947051. Accessed 22 Apr. 2024.

[4] Martha. "Zedboard." ZedBoard - Digilent Reference, digilent.com/reference/programmable-logic/zedboard/start. Accessed 24 Apr. 2024.

[5] Redmon, Joseph. Tiny Darknet, pjreddie.com/darknet/tiny-darknet/. Accessed 22 Apr. 2024.

[6] "Vivado Design Suite User Guide: High-Level Synthesis." AMD Technical Information Portal, 4 May 2021, docs.amd.com/v/u/en-US/ug902-vivado-high-level-synthesis.

Thank you to our advisor, Hunter Adams, for offering his guidance and support to us this past year