[[1]](#footnote-0)

Hardware Acceleration for Hyperdimensional Computing

Professor Sitao Huang, Joseph Prince, Hong Dong, Kyle Van, Brad Lacy

***Abstract*—The goal of this project is to accelerate the calculations of HDC (Hyper-dimensional Computing) using an FPGA. We want to enable embedded systems that have power consumption restraints to run AI applications. To do this, we wrote a classification algorithm, in C code, that heavily uses HDC. After that, we synthesize the code on Vitis HLS so that the code can be run on the Pynq Z2 board using an onboard FPGA. Then, we write the code to run the algorithm on a Jupyter notebook that runs on the Pynq Z2 board. The components of this algorithm that involve HDC are run on the FPGA. All these steps have been completed and will be discussed more in the paper. This project used data from the MNIST dataset to test the speed and accuracy of the algorithm, and we compared the time it took to test with and without the FPGA to see if this project actually sped up HDC.**

# INTRODUCTION

H

ardware Acceleration for Hyper-dimensional Computing is designed to use a simple FPGA on an embedded system to speed up hyper-dimensional computations. A vector is considered hyper-dimensional when it contains a very large amount of data and when it holds values between -1 and 1. It doesn’t necessarily have many dimensions, but rather one dimension is very large. The goal of this project is to run a classification algorithm on a small embedded system with an FPGA that will improve performance time. In doing so, the expensive calculations involved with classification should become viable on the system. To do this, we will use the Pynq Z2 board, shown below in Fig. 1, since it has its own built in FPGA. The processor on this board allows us to write logic and send computations to the FPGA. Other researchers have successfully used hyper-dimensional computing to solve the problem of classification, but no other researchers have tried accelerating it with an FPGA on an embedded system. Therefore, we will be the first group to accomplish our goal.

## Abbreviations and Acronyms

FPGA – Field Programmable Gate Array

HDC – Hyperdimensional Computing

HLS – High-Level Synthesis

RTL – Register Transfer Level

# Main Body

## Materials Used

* Pynq Z2 board: Zynq 7000 development board
* Vitis HLS: Tool for synthesizing C code into programmable logic
* Vivado: Software used to create a block design and for generating a bitstream

A picture containing text

Description automatically generated

Fig. 1. Image of Pynq Z2 Board

## High Level Software/Hardware Systems

The software involved in this project uses an algorithm to classify data. Because computers do not inherently have the ability to learn, this algorithm borrows clues from the way a brain learns for its implementation [1]. This allows computer architecture to learn as if it is a brain. The algorithm is as follows: training data is encoded into a hyperdimensional vector, or a hypervector, and is organized in different classes. The computer should already know what the training data represents. Then, sample data is encoded into a hypervector, and is compared against each class using a cosine similarity [2]. Whichever class the sample hypervector is closest to will then be considered as a member of that class. This algorithm is summarized in Fig. 2 below. On the hardware side, everything is going to run on the Pynq Z2 board. This is a SoC that already has an FPGA built into the system. To avoid directly writing

Diagram

Description automatically generated

Fig. 2. High-level Overview of Classification Algorithm

RTL code, our approach is to utilize Xilinx’s high-level synthesis (HLS) software - Vitis HLS. This software is useful to us because its advanced algorithms allow us to write C code and convert it directly into a Verilog and VHDL project [3]. After synthesis is performed, the resulting RTL project is exported into another piece of software, Vivado, where we create a block design. As shown in Fig. 3 below, this block design defines how the ZYNQ7 Processing System connects to the synthesized design generated from the previous step. With this block design, we are now able to generate the bitstream that tells the FPGA how to interpret data. Essentially, this will allow us to send every matrix multiplication to the FPGA for increased performance. Finally, after writing the logic to handle the algorithm and send data to the FPGA, we are able to use the high computational complexity of the classification algorithm on a real-life system where power is limited [4].

Diagram

Description automatically generated

Fig. 3. Block Design Used to Generate Bitstream

## Methods

The classification algorithm is first coded using the C programming language. This is because HLS is picky about the syntax of the code provided and writing the code in C is much easier for synthesis than even something like C++. After the pure logic is tested, the project is expanded to provide more of an interface for getting data and running the algorithm. This code is then moved over to Vitis HLS where the structure of the code is modified to optimize and improve the generated RTL code. Once our synthesis report gives us a desirable latency and does not have any hardware limitations, we export this design into Vivado to generate the bitstream. This step is fairly straight forward and mainly involves following the steps to connect the processing system to the synthesized code. Finally, the project is set up using a Jupyter Notebook that runs on the board. This notebook will load the FPGA’s BRAM with the appropriate data described by the bitstream and will copy the return data back into the Pynq Z2’s DRAM. After this step, we perform tests to gather metrics on the speed and accuracy of the algorithm both with and without the FPGA. This information lets us determine if the FPGA actually sped up the computations.

## Results and Performance

# Summary

# Conclusion

Acknowledgment

Our team wishes to acknowledge the contributions of Professor Sitao Huang and the department of Electrical Engineering and Computer Science. In providing the group with not only insight, but also the necessary equipment, they have been invaluable in completing this project.

References

[1] P. Kanerva, “Computing with High-Dimensional Vectors,” IEEE

Design & Test, vol. 36, no. 3, pp. 7–14, Jun. 2019, doi:

10.1109/mdat.2018.2890221.

[2] Z. Zou, Y. Kim, F. Imani, H. Alimohamadi, R. Cammarota, and M.

Imani, “Scalable edge-based hyperdimensional learning system with

brain-like neural adaptation,” Proceedings of the International

Conference for High Performance Computing, Networking, Storage and

Analysis, Nov. 2021, doi: 10.1145/3458817.3480958.

[3] J. Cong, Bin Liu, S. Neuendorffer, J. Noguera, K. Vissers, and Zhiru

Zhang, “High-Level Synthesis for FPGAs: From Prototyping to

Deployment,” IEEE Transactions on Computer-Aided Design of

Integrated Circuits and Systems, vol. 30, no. 4, pp. 473–491, Apr. 2011,

doi: 10.1109/tcad.2011.2110592.

[4] M. Imani, J. Messerly, F. Wu, W. Pi, and T. Rosing, “A Binary

Learning Framework for Hyperdimensional Computing,” IEEE Xplore,

Mar. 01, 2019. https://ieeexplore.ieee.org/document/8714821 (accessed

Dec. 02, 2022).

1. This paper was submitted for review on 03/16/2023. Financial support was

   provided by Professor Huang and the department of Electrical Engineering and Computer Science at the University of California, Irvine. All pieces of hardware came from this department.

   Professor S. Huang is with the University of California, Irvine and is an

   Assistant Professor (e-mail: [sitaoh@uci.edu](mailto:sitaoh@uci.edu)).

   J. Prince is a 4th-year student studying Computer Science and Engineering at the University of California, Irvine (e-mail: [jdprince@uci.edu](mailto:jdprince@uci.edu)).

   H. Dong is a 4th-year student studying Computer Science and Engineering at the University of California, Irvine (e-mail: [hsdong1@uci.edu](mailto:hsdong1@uci.edu)).

   K. Van is a 4th-year student studying Computer Engineering at the University of California, Irvine (e-mail: [vankm@uci.edu](mailto:vankm@uci.edu)).

   B. Lacy is a 5th-year student studying Electrical Engineering at the

   University of California, Irvine (e-mail: [lacyb@uci.edu](mailto:lacyb@uci.edu)). [↑](#footnote-ref-0)