INTEGRATING AI HARDWARE IN ACADEMIC TEACHING: EXPERIENCES AND SCOPE FROM BRANDENBURG AND BAVARIA

Zhouyi Xiong[1*], Dirk Stober[1], Miloš Krstić[2, 3], Oliver Korup[2], Maria Isabel Arango[2], Hao Li[1], Martin Werner[1]

[1] Technical University of Munich, Germany - *zhouyi.xiong @tum.de, {dirk.stober, hao_bgd.li, martin.werner}@tum.de
[2] University of Potsdam, Germany - oliver.korup@geo.uni-potsdam.de, maria.arango.carmona@uni-potsdam.de
[3] IHP Microelectronics, Frankfurt (Oder), Germany - krstic@ihp-microelectronics.com

KEY WORDS: Chip Design for AI, AI Detection for Natural Hazards, On Board CNN Detection, Hands on Teaching, University Curriculum, BB-KI.

ABSTRACT:

The field of artificial intelligence (AI) has gained increasing importance in recent years due to its potential to sustain growth and prosperity in a disruptive way. However, the role of special hardware for AI is still underdeveloped, and dedicated AI-capable hardware is crucial for effective and efficient processing. Moreover, hardware aspects are often neglected in university teaching, which emphasizes theoretical foundations and algorithmic implementations. As a result, there is a need for courses that focus on AI hardware development and its diverse applications.
In response to this need, the BB-KI Chips consortium aims to develop a series of hardware-oriented courses with real-world AI applications. This consortium includes the Technical University of Munich (TUM) and the University of Potsdam (UP), which both offer a wide range of courses that focus on AI basics, AI algorithmic development, general computer architectures, chip design, and as well applications of AI. In the BB-KI-CHIPS project, these different capacities are planned to be tightly integrated into a unified curriculum covering knowledge from chip design over AI algorithms and techniques to applications.

1. INTRODUCTION

1.1 Background

Artificial intelligence (AI) has emerged as a vital research field in the era of digitalization, with the potential to facilitate growth and progress in a transformative manner across various domains. Compared to the wealth of algorithms and software solutions, the role of dedicated hardware for AI remains underdeveloped, with limited coverage in academic teaching and university curricula in Germany. Despite efforts by prominent technology firms such as Xilinx, NVIDIA, ARM, and Intel to integrate AI elements into their platforms, university education in AI hardware design and development is often inadequate. In Germany, hardware tailored to AI applications is of particular interest as it obviates data collection in the cloud which raises data protection concerns and high energy cost due to wireless communication. Effective and efficient processing, essential for deep learning-based algorithms, necessitates innovative hardware architecture designed for AI tasks specifically. Nonetheless, current university programs tend to emphasize theoretical foundations and algorithmic implementations over the design, development, and evaluation of new AI architectures for hardware in a hardware software co-design approach.

The need to connect fundamental values such as the right to privacy, data sovereignty, and data federalism with data-driven innovations (such as Industry 4.0, 6G, smart cities, autonomous driving, IoT, remote sensing, and geodata analysis) has made Edge computing a crucial research and education area (Beck et al., 2014). However, the current focus of AI training programs in Germany remains on software development and the application of existing technologies, with inadequate attention given to AI hardware development and its diverse areas of application.

Efforts to bridge this gap in AI hardware education have gained momentum only recently. For instance, Germany's Federal Ministry of Education and Research (BMBF) has launched an initiative that aims to improve AI hardware education and research in universities. Various research institutes and universities have established AI-focused hardware labs, promoting research and development of AI hardware. However, more needs to be done to expand the scope of AI education in universities, particularly in AI hardware development, to enable graduates to meet the current and future needs of the industry.

The significance of AI in shaping the future of various fields is undeniable, and it is essential to promote AI hardware education in universities to ensure graduates are equipped with the necessary skills to develop innovative hardware architecture to support AI applications.

1.2 BB-KI Chips Project

The Technical University of Munich (TUM) and University of Potsdam (UP) offer a wide range of courses that focus on AI basics, AI algorithmic development, general computer architectures, and chip design. Both universities also focus strongly on natural sciences in teaching, in which many applied areas (e.g., computer vision, geodata and natural hazards, big

data) benefit from AI technology. Hardware aspects currently play an underexplored role in teaching, which mostly deals with the development and application of algorithms instead. Hardware should play a more important role. Currently, there is a gap between computer science research on AI and electrical engineering / chip design which we want to close. Similarly, there is a gap between applied AI (e.g., AI4EO or similar approaches) and AI research that we need to close through education. Therefore, in the context of the BB-KI Chips project funded by BMBF, we aim to jointly develop an AI-hardware related curriculum in the German federal states of Brandenburg and Bavaria. The vision of BB-KI Chips is to develop a curriculum of hardware-oriented courses based on and leading to real-world AI applications. Our goal in the long run is to make it easier for TUM and UP students to understand and apply AI technology. Specifically, we aim to simplify the process of connecting AI specifications to the hardware implementation that meets the requirements for performance, safety, and reliability following the vision of hardware software co-design.

The BB-KI Chips project aligns with the three goals of social impact, experience-based learning, and hardware-AI competence. The BB-KI Chips project has a unique opportunity to produce chips in Germany through its affiliation with the Leibniz Institute IHP in Frankfurt/O as well as with Global Foundries (GF). The program seeks to make students fit for work, research, entrepreneurship, and innovation in the field of AI hardware, especially through the practical implementation of student developments as part of the project. By interlinking teaching between application, hardware, and AI, the program will enable relevant hardware competence that integrates the transition between laboratory and practice into teaching for a more sustainable learning outcome.

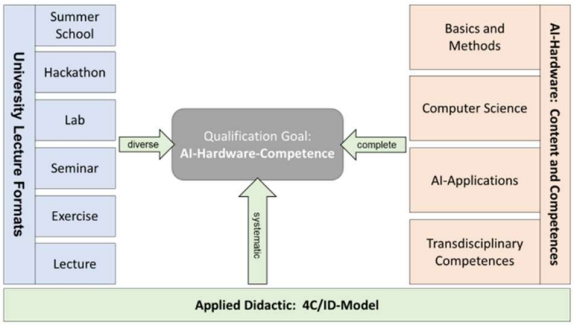

Figure 1. The methodological approach in BB-KI chips.

2. AI DEADWOOD RECOGNITION - A CASE STUDY

Within the scope of the BB-KI-CHIPS project, the project partners implemented a first case study course on machine learning for remote sensing. This seminar course has been offered jointly by TUM and UP in the winter semester of 2022. Master students from different study programs can have selected this course as an elective. The seminar provides hands-on experience to students for dealing with real world remote sensing problems. Students were given real-world drone imagery of an active river channel that had been impacted by a recent landslide-induced flood. The task was to use AI to classify the drone images into those containing deadwood and those containing sediment or river channels only. Deadwood is a major constituent in flood dynamics in forested catchments especially, and a major element compromising the intactness of infrastructure such as bridges or culverts. The images were acquired during a field visit using a consumer-off-the-shelf drone. on the Ecstall River, British Columbia, Canada, several weeks after the flood. The aim of this seminar is to explore the potential of machine learning algorithms in recognizing deadwood in the Ecstall River.

Students were asked to implement the optimal machine learning system for deadwood recognition they can imagine under the constraints that it should be simple and small enough to fit on hardware and real-time implementations. Example Jupyter Notebook for preprocessing the data into valid training samples have been prepared as an entry point to this course. The Notebook showed how to split raw images into small tiles for labeling and basic data preprocessing. Furthermore, the students got access to the computing hardware owned by the Big Geospatial Data Management professorship.

2.1 Motivation

Deadwood recognition is an important task in terms of flood hazard. Large woody debris such as logs, trees, branches and roots) in rivers may aggravate the consequences of flood events. This debris material may affect infrastructures such as bridges, weirs, etc., especially those intersecting forested mountain rivers. In order to track sediment dynamics and the health of the river ecosystem, the presence of deadwood in rivers must be studied and included in flood hazard and risk analysis (Virginia et.al, 2014). On 1 September 2022, a large ice/rock avalanche occurred on the Ecstall River (Petley, 2022) and dropped into a meltwater lake, thus causing a major displacement wave that triggered a major flood downstream.

A case study like this is ideal for students to learn about the handling and processing of real-world datasets. Computer vision techniques such as CNN are increasingly being used for real-time detection tasks due to their ability to provide high accuracy and fast processing times. In this context, the course aimed to investigate the feasibility of using CNNs for real-time interactive deadwood data acquisition for scenarios in which drone flights are used for rapid assessments of debris and deadwood loads in flooded rivers. Specifically, we seek to determine the extent to which AI hardware requirements can support a system that provides a pilot with immediate feedback on the location and type of deadwood, while simultaneously providing environmental and geoscientists with feedback on the volume or area of deadwood covered, including a precise map of the area.

2.2 Course Design

The first two lectures introduced the deep learning (CNN) model, the training framework, and the learning goal to the students, enabling them to understand the theoretical framework of their active role in the task. The students then formed groups of five, with each group having a PhD student as supervisor. A template notebook was shared with the students, which contained the simplest model and fixed the document structure, dataset loading, model training, model evaluation, and model. This helped the students to familiarize themselves with the system and then iteratively improve it by changing the model and training aspects.

The students were also introduced to the background of why deadwood recognition is essential for hazard appraisals and how deadwood is related to flood risk, given by researchers from UP. This helped the students, who mostly had no previous exposure to environmental or geosciences, to understand the background and motivation of what they are doing in order to better memorize and

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-X-5-W1-2023-75-2023 | © Author(s) 2023. CC BY 4.0 License.

identify with this perspective.

During the group working period, students could show their intermediate steps and ask questions. The tutors provided a high amount of support in the beginning of the course, though gradually lessening as the students' abilities towards the task matured.

At the end of the project, the students presented their work as a talk and submitted slides. The students thus learned how to communicate their results, especially without technical details, and in a frontal talk. The students also submitted their improved Jupyter notebooks together with their datasets that they might have labeled. This helped the students to learn to finalize projects and how to communicate deliverables in a way that is reproducible for others.

Finally, evaluations were given both ways. The final reports from the group work were graded, and the students were asked to provide feedback on what they had to learn from outside the course to build a word cloud. This helped the students to reflect on their own learning behavior and guide the course instructors to extend or change the curriculum.

2.3 Experimental Setup

The dataset consists of two short video streams in MPEG4 format and 16 JPEG images that were captured during a manual drone flight over a river valley affected by deadwood. As this is a single drone survey on a specific day in a particular region, the generalization of the results may be questionable, but served more as a simple proof of concept. Therefore, the findings can be considered initial explorations, and further research is needed to obtain more accurate results.

The primary goal of the project is to classify each image into one of three classes, namely debris, forest, or water. However, the initial problem formulation is biased as most images will not belong to any of these classes, while many tiles may consist of multiple classes. Therefore, data needs to be cut into tiles of adequately defined size to allow a tradeoff between bias, difficulty, computational complexity, and the number of instances available for training. Hence, choosing the tile size is a crucial hyperparameter that must be evaluated carefully. Each of the original images was divided into tiles of 64 pixel by 64 pixel.

In the labeling phase, images are sorted manually into three subfolders, each representing a class (Examples of the labelled tiles are shown in Figure 2-4). However, for evaluation purposes, the dataset had to be split into at least three sets that are statistically independent of each other. This is an important, but challenging task to avoid the effects of spatial autocorrelation. Therefore, it is essential to use appropriate evaluation procedures to ensure the accuracy and reliability of the results. Despite these limitations, the traditional approach in machine learning is to represent image classification tasks using folders of examples.

The deadwood recognition project is thus an initial exploration of the potential of machine learning algorithms in detecting deadwood. The limitations of the dataset, including the absence of mission data and the biased initial problem formulation, must be considered when interpreting the results. Therefore, further research is necessary to improve the accuracy and reliability of the findings. However, the project's findings have the potential to contribute to the development of more efficient and effective deadwood detection methods, which can have significant ecological and environmental implications.

Figure 2. Examples of the debris class and one debris tile

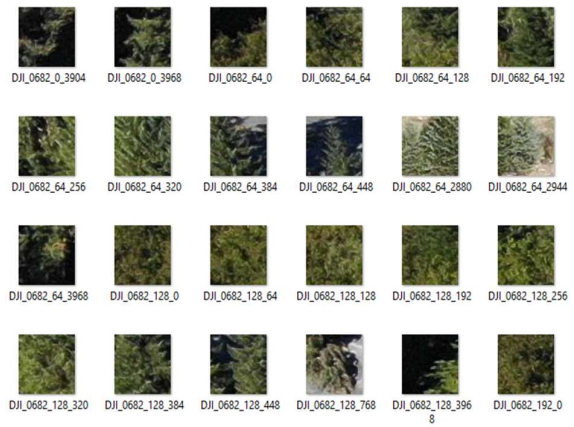

Figure 3. Examples of the forest class

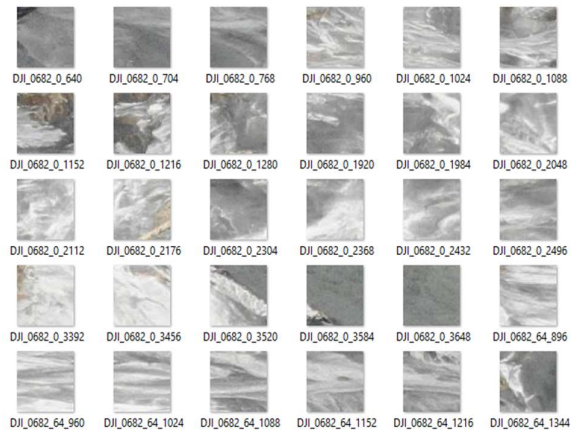

Figure 4. Examples of the water (or active channel) class

2.4 Case Study Results

For the training model selection, different students used various machine learning models including Support Vector Machine (SVM), ResNet, Random Forest (RF), and Mobile Neural Network (Mobilenet).

All the CNN models reached an accuracy above 90%. One CNN architecture utilized consisted of two convolution layers and two max pooling layers reached an accuracy of 98%. The convolution layers are utilized for learning the features of the input images and extracting relevant information for deadwood recognition. Each layer is also composed of a set of filters that slide over the input image, applying a dot product between the filter weights and the input values. The max pooling layers are applied to reduce the spatial dimensionality of the feature maps produced by the convolution layers while preserving the most important information. The final precision value was 94%, recall was 96%, and the F1-score was at 95%, which is highly promising in demonstrating the accuracy of the network while also minimizing the number of false positives.

Each group reported quantitative results on the performance of their models. Notably, one group managed to leverage their trained model to make simple predictions, thereby demonstrating the potential for future practical applications of their approach. Furthermore, another group compared multiple models to identify the most suitable solution. The comparative analysis involved a range of quantitative metrics, including accuracy, precision, recall, and F1-score.

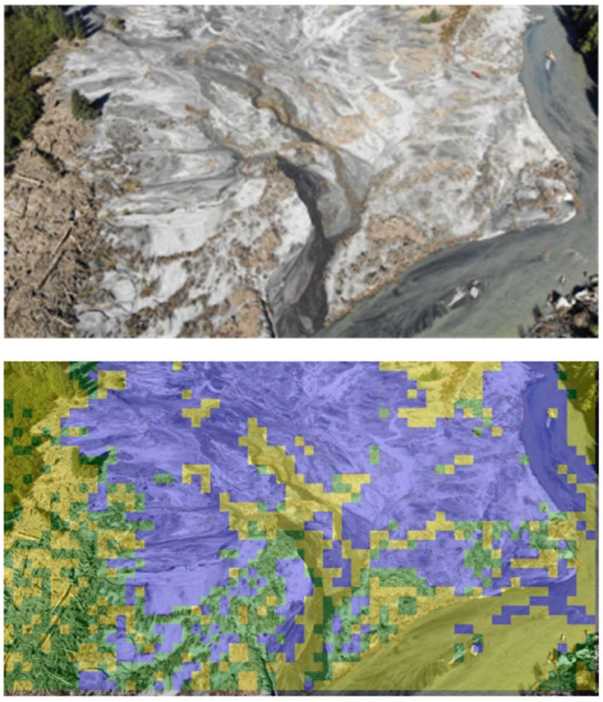

Figure 5. Example of prediction, where color yellow represents the debris green represents forest. We can see some of the forests and water are miss predicted as debris.

The findings of the study provide valuable insights (using other ways to compare different models could lead to more accurate prediction) into the performance of different machine learning models for object detection and classification. These results highlight the potential for future research in the field of machine learning, particularly in the development of more effective and efficient models for deadwood detection and classification.

2.5 Course Summary

A highly successful course on deadwood recognition was conducted, providing students with an opportunity to experience real-world problems in machine learning. Students without much previous programming knowledge were able to handle raw data and train simple models after the course. The course provided a working example to students, offering a high level of support in the initial stages of the learning process.

Unlike preprocessed datasets that are often used in machine learning, the course utilized raw drone images to present real-world challenges to students. This approach fostered self-organized learning and innovation among students, providing them with an opportunity to apply their knowledge and skills to solve complex problems.

The course included strong social support structures, with regular meetings conducted in small groups. This facilitated peer-to-peer learning and collaboration, creating an environment conducive to effective learning and skill development. Overall, the course provided a comprehensive learning experience that prepared students for future challenges in the field of machine learning. The follow-up courses should address the hardware context related to the application of AI models. This will enable students to evaluate different AI models from a hardware implementation perspective, resulting in a set of practical and applicable approaches for future use.

3. ACCELERATING CNNS USING PROGRAMMABLE LOGIC

The course will be offered starting from the summer semester in 2023 as a part of the BB-KI curriculum focus on hardware. Bachelor and master students in computer science and related fields, who have experience in Programming C/C++, can attend the course.

3.1 Course Goals

The objectives of this course are to equip students with a comprehensive understanding of CNNs, familiarize them with common optimization schemes for convolution and provide a basic knowledge of AI accelerators currently in use. Furthermore, they should be able to design simple digital circuits using High-Level Synthesis (HLS) and Hardware Description Language (HDL) and integrate them together with the software on one System-on-Chip (SoC) platform. Finally, students should be able to compare and reason about the performance of different implementations.

3.2 Structure and Resources

The course consists of a lecture and parallel lab exercises that lead up to a final project. To keep students progressing in time there are three mandatory tasks including the report of their optimized Software implementation. After implementing and optimizing the CNN inference in Software, students will present existing AI architectures in a flipped classroom approach. Following the presentations, students will have to suggest their own Hardware design in the form of a short Proposal. Then students will implement the design using HLS and iterate over it with a second implementation in HDL (SystemVerilog).

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-X-5-W1-2023-75-2023 | © Author(s) 2023. CC BY 4.0 License.

The course will start with lectures introducing CNNs for image recognition, which were also used in the deadwood recognition case study. The course will then focus on the convolution kernel, which can take up to 90% of the processing time (Frabet et al., 2010). The programming language used in the course will be C/C++ due to its performance and proximity to hardware.

The Project part will start with the Hardware Proposal and is accompanied by lectures covering Design, Simulation and Implementation of HLS and HDL designs using the Vivado toolchain. Students will work in small groups of up to 3 and use the Pynq Z2 development board. The Pynq Z2 includes a Zynq7000 chip which is a well-documented platform for hardware implementation and experimentation, providing a cost-effective and easily accessible tool.

3.3 Software Implementation

The course will focus on studying how to implement inference of CNNs, like those in the case study of deadwood recognition project. Implementing CNNs can be overwhelming at the start, as different layers (Fully Connected, Pooling, Activation Function, Convolution) have to be implemented and connected which can be difficult to debug. To address this problem and simplify the task for students, they will start with implementing the different layers separately and are provided with a 3D data structure (Tensor) and python scripts to generate test cases to verify correctness, before connecting the individual layers.

Integrating the layers poses more challenges including loading of weights, setting up the different Network architectures and reading the inputs from a file and outputting the classification, this will allow them to gain a deeper understanding of CNN inference. By the end of the course, students will have a solid grasp of the primitives inside a CNN and be able to implement them for other Neural Networks architectures as well.

3.4 Optimizations

The convolution kernel inside a CNN can be optimized in Software leading to a considerable speedup depending on the input and kernel size (Lavin and Gray, 2016). Three popular optimization schemes will have to be implemented by the students. First, Convolution can be formulated as a Matrix-Multiplication with use of the Toeplitz Matrix, which allows for the use of already optimized Matrix-Vector subroutines (Sze et al., 2017).

Second, convolution in the time domain can be performed in the frequency domain as a complex element-wise multiplication. To transform the Input, Weight and Output maps the FFT in combination with an Overlap-save scheme can be used (Harris and Elliot, 1987). During Inference the Weights of the Network do not change, amortizing their transformation as they can be computed once and stored in the Frequency domain.

To accelerate smaller convolutional kernels, the Winograd Transform can be used (Winograd, 1980). The Transform uses the spatial locality of the inputs to compute multiple outputs at once with a minimal amount of multiplications required at the cost of more additions. These speeds up the kernel, as additions are generally faster than multiplications.

The algorithm transforms Inputs, Weights and Outputs to and from the Winograd domain using the Matrices G, A, and B (Fig. 6). The matrices G, A, and B are derived using the Chinese Remainder Theorem, which can be used for different input tiles and kernels (Winograd, 1980).

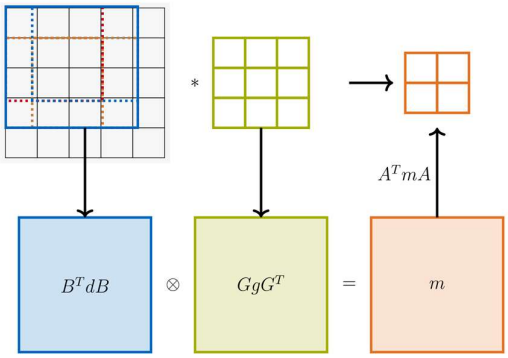

Figure 6.: Convolution using Winograd for a 3x3 kernel and 2x2 output tile.

For small input tiles and weight kernels the matrices mostly include small integers (1, 2, ...), but for bigger tiles the matrices get more complex, leaving the algorithm less effective for larger kernels. Both the FFT and the Winograd follow a similar pattern and can be formulated as a Matrix-Multiplication. ((Lavin and Gray, 2016)

3.5 Accelerator Architecture

The students propose architecture as mentioned previously; however, this is not the final version and can be changed later on if desired. The students can decide for themselves how general the architecture should be, and the final design is not solely evaluated on performance.

For example, the simplest approach would be to implement a specific accelerator that only works with a specific sequence of layers. This would probably be the most efficient design but could not be used for any other networks. A more general design would include support for a different sequence of layers with a set of implemented layers.

Another approach is the implementation of a systolic array that can support matrix multiplication like Google's TPU (Jouppi et al., 2017). Finally, an even more general architecture would be an array of processing elements with individual control and register files like the Eyeriss architecture (Chen et al., 2019). The more general the architecture the less performance is to be expected, however as the final goal is to eventually design an ASIC more general designs must be explored as well.

3.6 Hardware Implementation

High-level synthesis (HLS) and Hardware Description Languages (HDL) are two popular methods to design FPGAs. HLS allows students to rapidly prototype their design from their already existing C/C++ implementation. To also understand the benefits and challenges of using an HDL, they will implement a second iteration using SystemVerilog and will aim to get a better performance than the HLS implementation.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-X-5-W1-2023-75-2023 | © Author(s) 2023. CC BY 4.0 License.

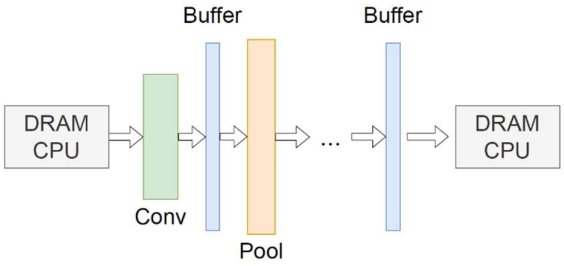

Figure 7. Layer-Specific Architecture

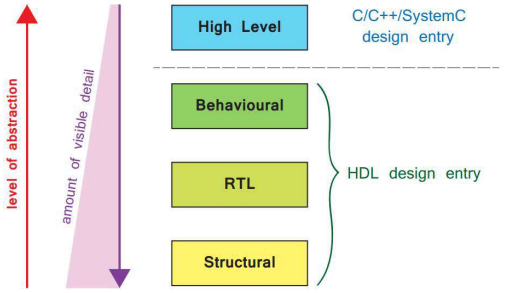

Figure 8. Levels of abstraction in FPGA designs (Crockett et al., 2014.)

When implementing a design on a SoC, it is necessary to communicate between the Central Processing Unit (CPU) and the PL/FPGA. One common method for connecting these components is through the Advanced eXtensible Interface (AXI) protocol. The AXI protocol is available in different variants for different use cases. The Vivado Design Suite offers existing IPs to connect CPU and PL using AXI and automatically generates drivers. This feature saves time and simplifies the development process by automatically generating code that can be used to program the CPU and control the Accelerator. Finally, students should balance the workload between accelerator and CPU to try to extract optimal performance. The results should then be presented in the form of a presentation and a report.

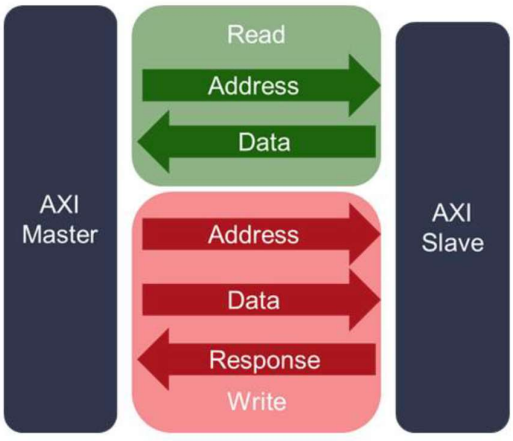

Figure 9. Connection via AXI-Interface

3.7 Course Overview

This course design aims to equip students with a comprehensive understanding of Convolutional Neural Networks (CNNs) and their computational requirements. Additionally, the course covers common optimization schemes for convolution and provides hands-on experience with implementation in software and optimization techniques. Students will also gain basic knowledge of existing AI accelerators and be able to design simple digital circuits using high-level synthesis (HLS) and SystemVerilog from concept to implementation. The course also emphasizes the integration of both software and programmable logic (PL) on a system-on-chip (SoC) platform, culminating in a project that involves designing and integrating an accelerator. Finally, students will develop the ability to reason about the performance of different implementations through reading papers, their own presentations, and written reports.

4. CONCLUSION

In summary, the collaborative effort between TUM and UP aims to facilitate the connection between AI specification and the implementation of hardware to meet the complex requirements for performance, safety, and reliability. This effort includes the integration of trained models from seminar courses into the hardware design course, learning application specific on-board chip design and providing a dedicated solution for on-board model inferencing. Additionally, the universities plan to offer courses like AI in Ethics and Field Programmable Gate Arrays (FPGAs) design workshops/courses regularly to equip students with the necessary knowledge and skills to navigate the constantly evolving field of AI. The program may also offer opportunities for students to participate in summer schools or visits to fabs to gain practical experience and deepen their understanding of hardware implementation.

Furthermore, the program includes a student demonstrator role, which aims to provide hands-on experience with the implementation of AI models on hardware. The first student demonstrator project, which involved the development of an ML chip for digit recognition, has already been completed. This joint effort represents a significant contribution to the development and growth of the AI industry, and it is expected to have a positive impact on the future of AI research and education.

ACKNOWLEDGEMENTS

This work was supported by the Federal Ministry of Education and Research of Germany under the grant number 16DHBKI020.

I want to thank the professors and colleagues for their valuable help in writing this paper. I also would like to thank my family Philipp and Aaron for your support

REFERENCES

Beck, M. T., Werner, M., Feld, S., & Schimper, T. (2014). Mobile Edge Computing: A Taxonomy. *Proceedings of the Sixth International Conference on Advances in Future Internet* (AFIN 2014).

Crockett, L., Fathi, M., & Boxall, J. (2014). The Zynq Book: Embedded Processing with the Arm Cortex-A9 on the Xilinx Zynq-7000 All Programmable Soc. Strathclyde Academic Media.

Farabet, C., Martini, B., Akselrod, P., Talay, S., LeCun, Y., & Culurciello, E. (2010). Hardware accelerated convolutional neural networks for synthetic vision systems. In *Proceedings of 2010 IEEE International Symposium on Circuits and Systems* 257-260. Paris, France. doi: 10.1109/ISCAS.2010.5537908

Harris, F. J., & Elliot, D. F. (1987). *Handbook of Digital Signal Processing*. Academic Press. ISBN: 9780080507804.

Jouppi, N. P., Young, C., Patil, N., Patterson, D., Agrawal, G., Bajwa, R., ... & Suresh, B. (2017). In-datacenter performance analysis of a tensor processing unit. In *Proceedings of the 44th Annual International Symposium on Computer Architecture* 1-12. Toronto, ON, Canada. doi: 10.1145/3079856.3080246

Lavin, A., & Gray, S. (2016). Fast algorithms for convolutional neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition* (CVPR) (pp. 4013-4021). Las Vegas, NV, USA. doi: 10.1109/CVPR.2016.435

Petley, D. (2022, September 21). Landslide Blog: Ecstall River 1. American Geophysical Union. Retrieved from https://blogs.agu.org/landslideblog/2022/09/21/ecstall-river-1/

Ruiz-Villanueva, V., Díez-Herrero, A., Bodoque, J., & Bladé Castellet, E. (2014). Large wood in rivers and its influence on flood hazard. Cuadernos de Investigación Geográfica, 40, 229-241. https://doi.org/10.18172/cig.2523

Sze, V., Chen, Y. -H., Yang, T. -J., & Emer, J. S. (2017). Efficient processing of deep neural networks: A tutorial and survey. *Proceedings of the IEEE*, 105(12), 2295-2329. doi: 10.1109/JPROC.2017.2761740

Vasilache, N., Johnson, J., Mathieu, M., Chintala, S., Piantino, S., & LeCun, Y. (2014). Fast convolutional nets with fbfft: A GPU performance evaluation. CoRR, abs/1412.7580.

Winograd, S. (1980). Arithmetic complexity of computations. Society for Industrial and Applied Mathematics (SIAM). ISBN: 9781611970364. URL: https://books.google.de/books?id=wANiW8bGQpEC

Yang, T. -J., Chen, Y. -H., Emer, J., & Sze, V. (2019). Eyeriss v2: A flexible accelerator for emerging deep neural networks on mobile devices. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 9(2), 292-308. doi: 10.1109/JET