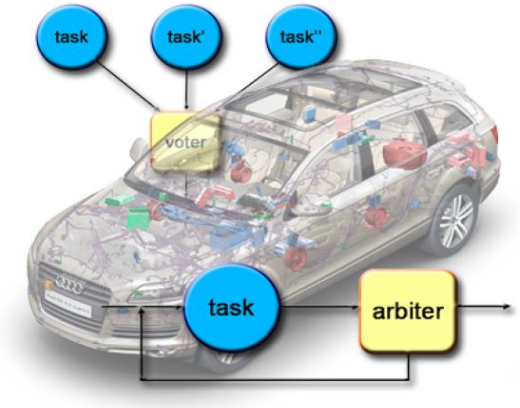
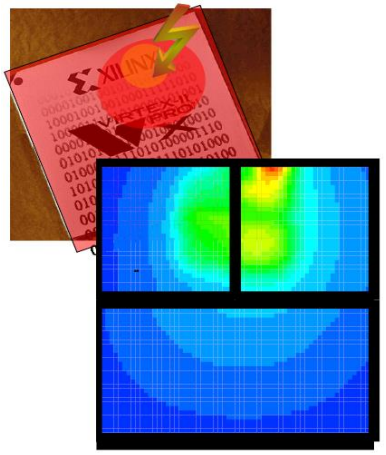
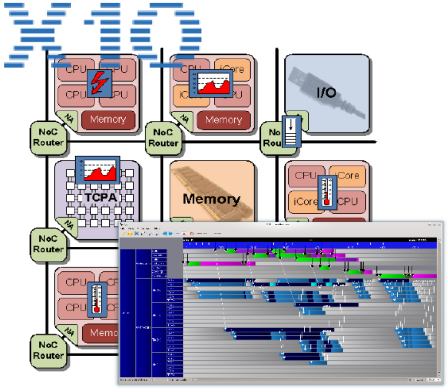
Themen für Projekt-, Bachelor- und Master-Arbeiten



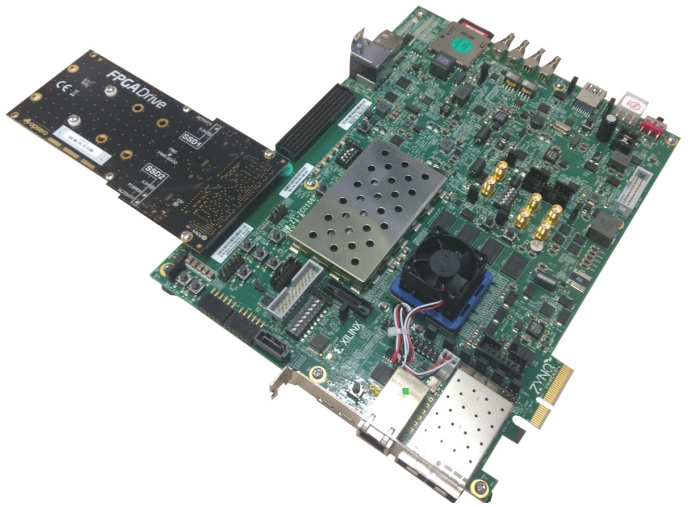




https://www.cs12.tf.fau.de/lehre

Near-data processing using FPGAs

A recent study evaluated four applications (including the Chrome browser, TensorFlow machine learning inference, and video encoding and decoding) and showed that 62% of energy was consumed by data movement. As a remedy, data should be processed as close as possible to its source. One possibility for doing so is using so-called SmartSSDs, which, in addition to the SSD chips, also contain an FPGA to process data directly in the drive.



To make access to databases as efficient as possible, indexes are commonly used, which make it possible to locate individual tuples on the drive quickly. In this work, a management system for such indexes is to be designed directly on the FPGA, with the goal of updating the index on-the-fly when writing to the drive in hardware. If data is later received from the network interface, it can be written directly to the drive without the CPU having to intervene, while still ensuring that the CPU can access data efficiently if necessary.

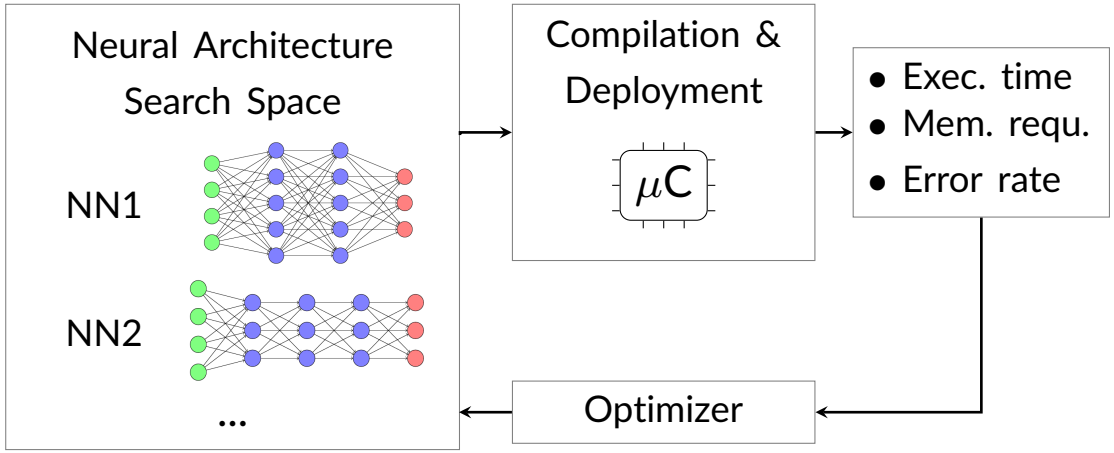
Prerequisites:	Basic knowledge in C++ and VHDL
Type of Work:	Theory (20%), Conception (40%), Implementation (40%)
Supervisors:	Tobias Hahn (tobias.hahn@fau.de)

Lehrstuhl für Informatik 12
Hardware-Software-Co-Design
Cauerstraße 11
91058 Erlangen

Friedrich-Alexander-Universität
Technische Fakultät

Neural Architecture Search for Time Series Prediction on µCs

Deploying neural networks (NNs) on microcontrollers (µCs) allows AI applications to be run close to sensors and increases the scope for future applications. However, designing NNs for resource-constrained µCs requires a delicate balance between maintaining a low prediction error while achieving low memory consumption and execution time.



As this is extremely challenging to achieve manually, platform-aware Neural Architecture Search (NAS) has become a major research topic. Here, the NNs are optimized regarding, e.g., the number of layers or the number of neurons per layer, with the aim of reducing the error rate and the execution time of NNs for a given target platform. In this work, one NAS algorithm is applied to the problem of time series prediction for an automotive µC as target platform. Finally, the NNs found with NAS are compared with existing neural networks and evaluated in terms of NN error rate, as well as memory consumption and execution time when deployed on the target µC.

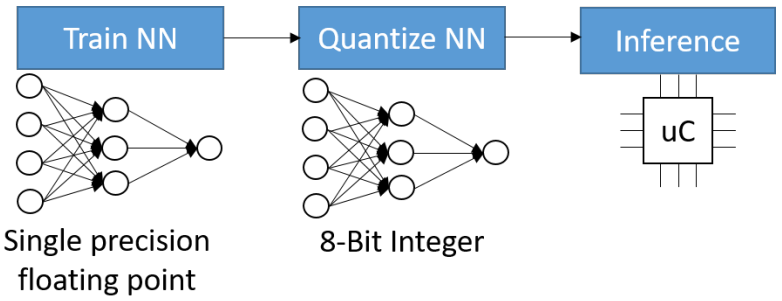
Requirements:	Knowledge in C, Python, and Neural Networks
Type of thesis:	Theory (20%), concept (40%), implementation (40%)
Supervisor:	Christian Heidorn (Christian.Heidorn@fau.de)

Lehrstuhl für Informatik 12
Hardware-Software-Co-Design
Cauerstraße 11
91058 Erlangen

Friedrich-Alexander-Universität
Technische Fakultät

Exploring Quantization of DNNs for Regressions Tasks

Quantization of *Neural Networks (NNs)* is a common way to reduce their computational intensity, which is particularly useful when using resource-constrained microcontrollers (µC). While quantization of NNs trained on classification tasks has been documented to work well, quantization of NNs trained on regression tasks often results in significant degradation of the NN's prediction performance (accuracy).



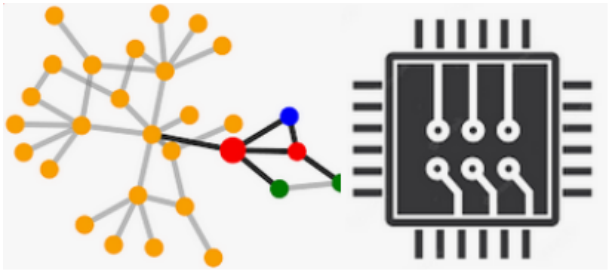
The main reason for this observation is that for classification tasks, small changes in the predicted probability vector across all classes due to quantization usually do not affect the predicted class derived from the distribution. In contrast, for regression tasks, small changes in the prediction can lead to a significant decrease in accuracy. The goal of this thesis is to investigate, implement and compare DNN training and quantization methods that can help improve the accuracy of quantized DNNs in regression tasks. Furthermore, the selected techniques should be evaluated in terms of the accuracy achieved for different regression problems as well as required memory consumption and execution time when deployed on different µCs.

Requirements:	Knowledge in C, Python, and Neural Networks
Type of thesis:	Theory (20%), concept (40%), implementation (40%)
Supervisor:	Christian Heidorn (Christian.Heidorn@fau.de), Mark Deutel (mark.deutel@fau.de)

Lehrstuhl für Informatik 12
Hardware-Software-Co-Design
Cauerstraße 11
91058 Erlangen

Friedrich-Alexander-Universität
Technische Fakultät

Towards ML-Guided Integrated Circuit Synthesis



An important step in Electronic Design Automation (EDA) is the synthesis of the Register Transfer Level (RTL). In this step, a Hardware Description Language (HDL) code is converted into a netlist using various optimizations. These optimizations use various heuristics and minimize the size, power, and speed of the final Integrated Circuit (IC). Graph Neural Networks (GNNs) have been successfully applied to combinatorial optimization problems. Recently, the OpenABC dataset was released to spur research in this area.

The dataset consists of a large set of netlists using various scenarios. As synthesis runs are computationally costly, Machine Learning (ML) can be used to predict the quality of the synthesis recipe without carrying out the synthesis. A set of simple graph models have been benchmarked on the OpenABC dataset.
The following tasks shall be carried out as part of this thesis:

1. Reproduce the existing results.
2. Develop novel models to improve the prediction quality.
3. Propose additional benefits of applying ML for tasks related to IC synthesis.

Prerequisite:	Programing Skills in Python, Understanding of VHDL and Circuit Design
Type of Work:	Theory (20%), Concept (30%), Implementation (50%)
Contact Person:	Muhammad Sabih (muhammad.sabih@fau.de)

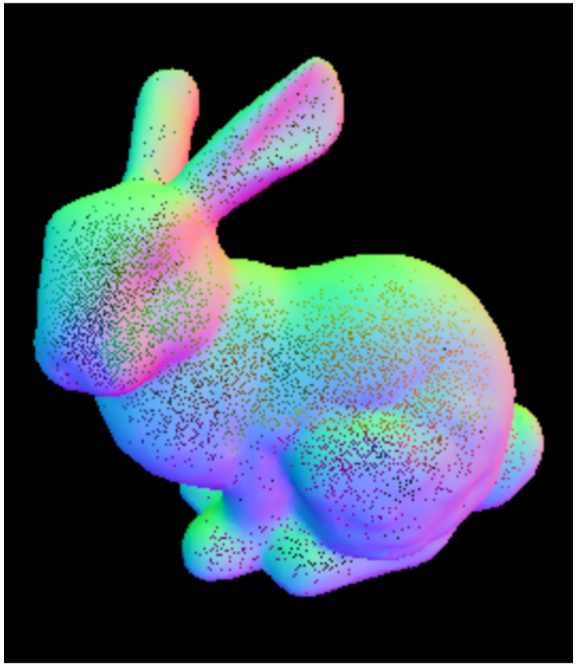
Lehrstuhl für Informatik 12
Hardware-Software-Co-Design
Cauerstraße 11
91058 Erlangen

Friedrich-Alexander-Universität
Technische Fakultät

Point Rendering on FPGAs

In 3D visualization, objects are typically modeled by polygon meshes consisting of triangles. *Point rendering* is an alternative where geometrically complex objects are represented by point clouds. Here, a set of points is drawn, commonly dense enough that no holes are visible in the visualization. Point rendering can significantly increase drawing speed thanks to its simplicity, particularly for complex objects, e.g., captured by high-resolution 3D scanners or lidar sensors.

In this *master's thesis*, a highly parallel hardware design should be developed and prototypically implemented in an FPGA. For this purpose, several transformations (e.g., projection) and visibility tests (backface culling, clipping, Z-buffering) have to be realized as a streaming pipeline. While the computation of the individual points can be embarrassingly parallelized, the final determination of whether a pixel is hidden by another one (Z-buffering) can lead to memory access conflicts. Therefore, one focus of the work should be on an efficient Z-buffer implementation utilizing the rich amount of distributed memory resources of the FPGA.



Demo Point Renderer by Marc Stamminger

Prerequisites:	Good kowledge in hardware and FPGA design, programming knowledge in VHDL, fundamentals in computer graphics
Type of Work:	Theory (10%), Conception (25%), Implementation (65%)
Supervisors:	Patrick Plagwitz, Frank Hannig, Timotei Ardelean*, Tim Weyrich*
Contact:	frank.hannig@fau.de

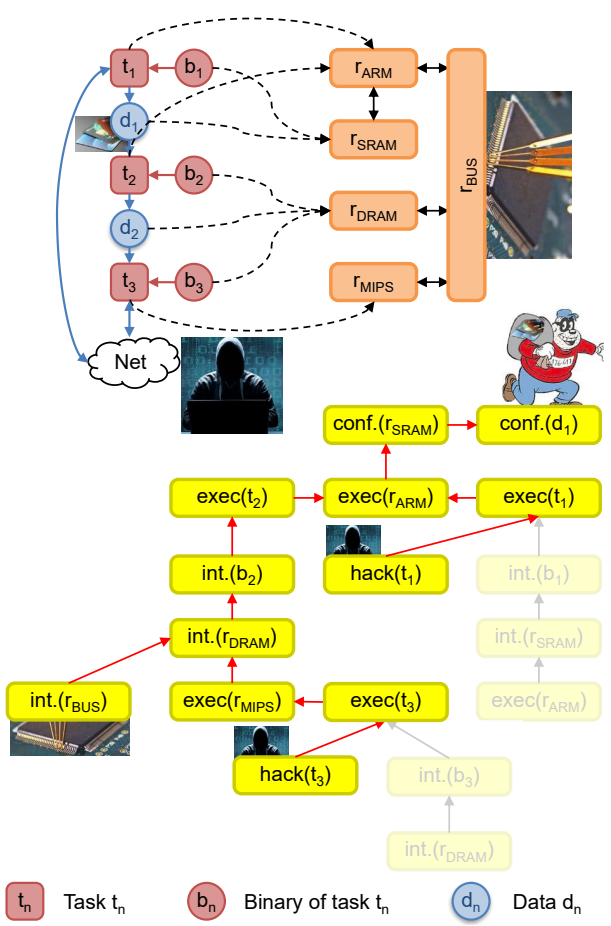
*Chair of Digital Reality

Lehrstuhl für Informatik 12
Hardware-Software-Co-Design
Cauerstraße 11
91058 Erlangen

Friedrich-Alexander-Universität
Technische Fakultät

Sicherheitsmodellierung auf der elektronischen Systemebene

Eingebettete Systeme (ES) beeinflussen viele Aspekte unseres Alltags. Sie befinden sich z. B. in Autos, medizinischen Geräten und dem Internet der Dinge. Daher ist es wichtig zu verstehen, welche Angriffe auf solche Systeme möglich sind und wie man sich davor schützen kann.
In dieser Masterarbeit werden Angriffe auf ES auf der elektronischen Systemebene untersucht. In einem ersten Schritt sollen mögliche Angriffe auf architektonische Hardwarekomponenten (Prozessoren, Speicher, Busse etc.) und die Anwendungssoftware (bestehend aus Tasks, Daten etc.) identifiziert werden. Dann sollten Regeln basierend auf diesen Angriffen formuliert werden, um *Angriffsziele* zu modellieren, die auf die *Vertraulichkeit*, *Integrität* und *Authentizität* eines bestimmten eingebetteten Systems abzielen. Angriffsbäume sind eine formale Darstellung, wie Angriffsziele erreicht werden können. Das Ziel dieser Arbeit ist die Bereitstellung einer Methodik zur automatischen Generierung solcher *Angriffsbäume* basierend auf einer gegebenen Systemspezifikation durch Anwendung der gefundenen Regeln. Diese Darstellung ermöglicht die Analyse von Sicherheitsrisiken auf Systemebene und kann sogar dabei helfen, Gegenmaßnahmen mit Hilfe von Entwurfsraum-Explorations-Techniken zu finden.



Voraussetzungen: Programmierkenntnisse in Java

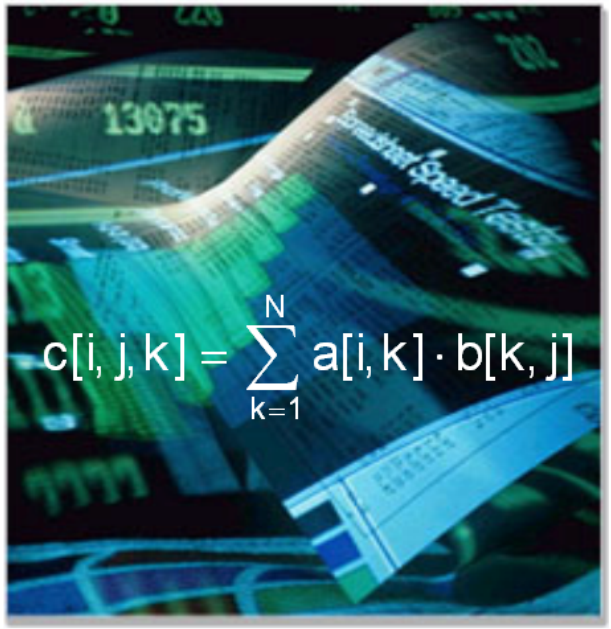
Art der Arbeit: Theorie (30%), Konzeption (30%), Implementierung (40%)

Ansprechpartner: Joachim Falk und Stefan Wildermann (vorname.nachname@fau.de)

Lehrstuhl für Informatik 12
Hardware-Software-Co-Design
Cauerstraße 11
91058 Erlangen

Friedrich-Alexander-Universität
Technische Fakultät

Data Integrity Modeling and Assurance in Co-Design



The assurance of security requirements, i.e., the integrity of data exchanged between resources in an embedded system, has not yet been adequately considered in design automation.
Data integrity can be impaired by either faults, aging, but also by security attacks. Examples of countermeasures include techniques for fault detection and fault correction, e.g., error-correcting codes, ECCs, redundancy, or through strong isolation of application data from other applications, e.g., by memory access control.
In this Master thesis, constraints on the integrity of data should be incorporated into an established co-design flow by attributing communications between tasks with integrity attributes. Based on these attributes, mapping constraints shall be generated to establish and thus assure integrity constraints to hold in any feasible mapping, including the allocation of resources and binding of tasks to resources.
The constraints to be generated thereby restrict mapping options, e.g., that data items of different applications must not be mapped to the same memory (strong isolation) or that memory protection is enabled through memory mapping units (MMUs). The techniques shall be implemented in an existing co-design framework and tested for sample applications.

Prerequisites:	Java knowledge required
Type of Work:	Theory (40%), Conception (20%), Implementation (40%)
Supervisor:	Joachim Falk (joachim.falk@fau.de) and Jürgen Teich (juergen.teich@fau.de)

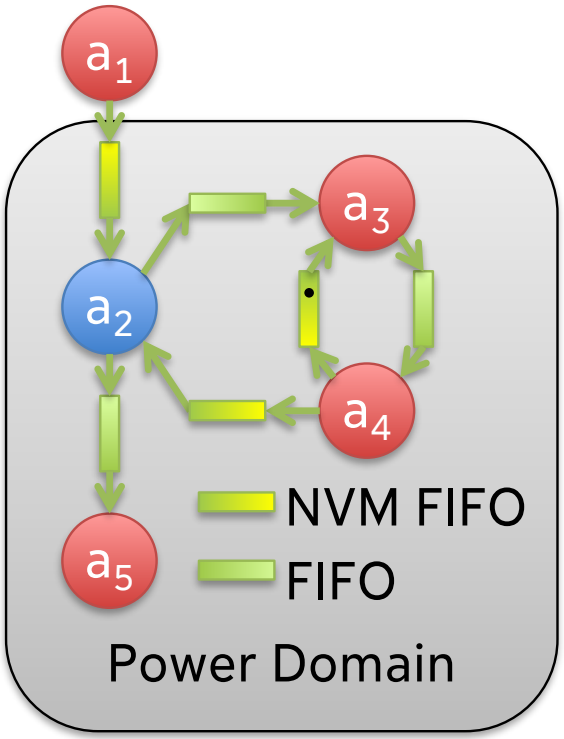
codesign

Lehrstuhl für Informatik 12
Hardware-Software-Co-Design
Cauerstraße 11
91058 Erlangen

FAU
Friedrich-Alexander-Universität
Technische Fakultät

Exploiting Non-Volatile Memory for Dataflow Computation

IoT sensor platforms need to be cheap, (ultra-)low power, long-running, and maintenance-free. Thus, battery-less IoT devices utilizing energy harvesting are more and more deployed. Often, such systems execute signal processing applications, which can be specified preferably by dataflow networks. These naturally allow the exploitation of concurrency by implementing each actor as a hardware circuit, all running in parallel. However, programming such sensor platforms offers unique challenges due to their intermittent power supply.



In this thesis, Non-Volatile Memory (NVM) should be exploited to realize dataflow networks in hardware to tackle these intermittency issues, e.g., by investigating and modeling persistable FIFO-based memory units. In particular, dataflow networks operating in mixed volatile/non-volatile operating modes shall be modeled by combining the system-level concept of dataflow, which is based on self-scheduled activations of computations, with NVM-based FIFOs. Inactive actors or even subnets should power down and reactivate upon the arrival of more data to be processed. In addition, for a continuously safe mode of operation, a powering down must also be triggered upon any intermittent shortage of power supply. Analogously, actors shall perform an auto-wakeup after recovery from the power shortage.

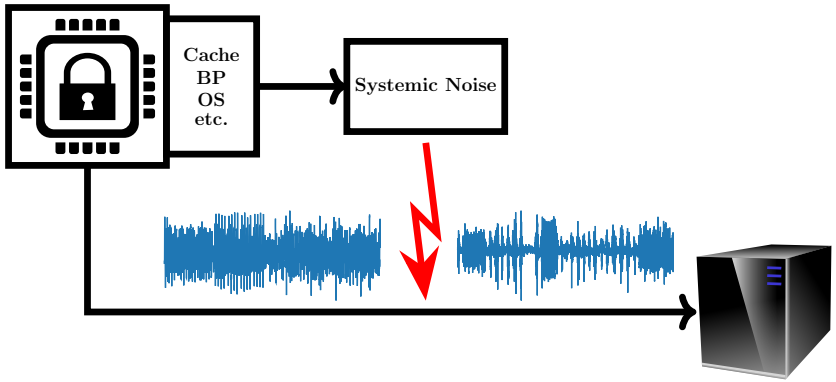
Prerequisites:	C++ and VHDL or Verilog knowledge required
Type of Work:	Theory (30%), Conception (30%), Implementation (40%)
Supervisor:	Joachim Falk (joachim.falk@fau.de)

Lehrstuhl für Informatik 12
Hardware-Software-Co-Design
Cauerstraße 11
91058 Erlangen

Friedrich-Alexander-Universität
Technische Fakultät

Investigating the Effects of Systemic Noise in physical Side-Channel Analysis on High-Performance Targets

Side-channel attacks are a powerful tool to extract secret information from cryptographic algorithms otherwise considered secure. However, side-channel analysis research is often performed on rather simple hardware, like microcontrollers, making the resulting attack techniques not necessarily applicable on more complex platforms.



This work focuses on the *systemic noise* inherent to complex platforms, as they contain performance-enhancing features, like Caches or Branch Prediction, which may lead to less deterministic execution behavior and as such less consistent side-channel measurements, impeding attacks.

In this work, a physical side-channel analysis of current cryptographic algorithm standards implemented on a high-performance platform containing multiple sources of *systemic noise* is performed in order to research the impact of the different *systemic noise* components and the applicability of previously developed side-channel attacks. Furthermore the extraction of internal data from associated *systemic noise* and its usage as another exploitable side-channel for attacks is investigated.

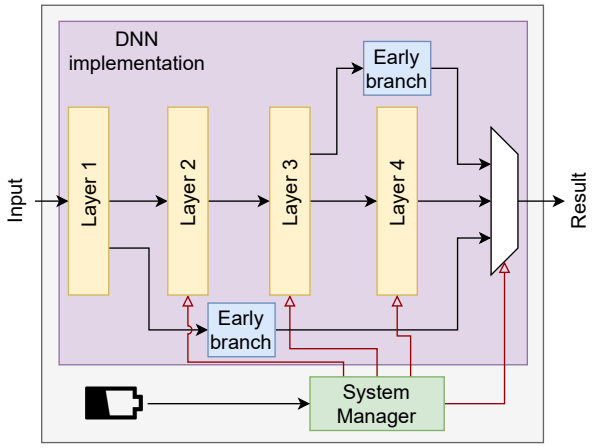
Prerequisites:	Basic knowledge in C/C++ and Python
Type of Work:	Theory (30%), Conception (20%), Implementation (50%)
Supervisor:	Paul Krüger (paul.krueger@fau.de)

design

Lehrstuhl für Informatik 12
Hardware-Software-Co-Design
Cauerstraße 11
91058 Erlangen

FAU
Friedrich-Alexander-Universität
Technische Fakultät

Energy-aware early-exit DNN Inference for Edge AI



Deploying DNNs on the edge, closer to where the data is collected, can help to reduce the energy consumed for data transmission to the cloud. When executing DNNs on these platforms, satisfying energy constraints due to an unreliable energy source (e.g., photovoltaic panels) while maintaining a good accuracy is challenging and requires adaptive DNN inference. One type of such adaptive inference, called Early Exit (EE), assumes that for a given classification problem, some data points are easier to classify than others, and as a result, high confidence can be achieved for such points even if only a subset of the DNN's layers, followed by an early-exit branch, are executed. In this thesis, the student investigates the use of EE for an audio processing application deployed on an energy-harvesting embedded device. The decision to use the output of an EE branch or not must be based on the accuracy and energy consumption of the branch, and on the remaining energy on the device. The student will have to design and train a classifier which leverages EE, evaluate its energy consumption and accuracy, and deploy it on an embedded target.

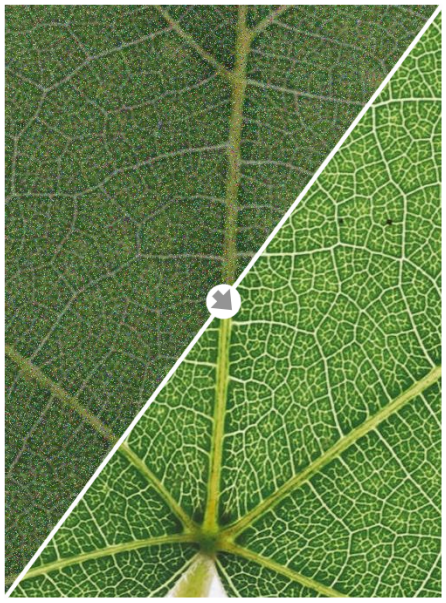
Prerequisites:	Basic knowledge of C, Python, and Deep Learning
Type of Work:	Theory (30%), Design (40%), Implementation (30%)
Supervisors:	Mark Deutel (mark.deutel@fau.de), Pierre-Louis Sixdenier (pierre-louis.e.sixdenier@fau.de)

Lehrstuhl für Informatik 12
Hardware-Software-Co-Design
Cauerstraße 11
91058 Erlangen

Friedrich-Alexander-Universität
Technische Fakultät

Latency-Constrained, Structure-Preserving Image Denoising

Denoising images is a crucial task in the domain of real-time medical image processing. Here, it is especially important that the fine-grained structures contained in the image, such as blood vessels, are not erased during the denoising process. Finding and choosing a neural network that efficiently enhances image quality while adhering to latency constraints and preserving the images' fine-grained structures can be achieved through *Neural Architecture Search* (NAS).



In this work, your task is to employ NAS to find suitable neural networks for denoising images that adhere to latency constraints and preserve image structure. You will define appropriate metrics for these objectives and employ methods to measure them and assess their quality. In addition, you will explore and extend an existing search space to find pareto-optimal neural networks for the given task.

This work is suited for a Master's thesis.

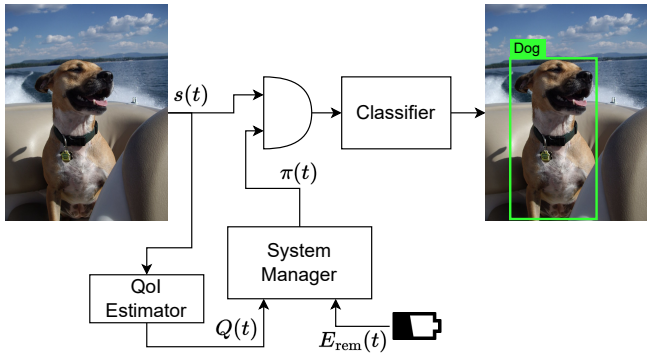
Prerequisites:	Python
Type of Work:	Theory (20%), Conception (20%), Implementation (60%)
Supervisor:	Stefan Groth (stefan.groth@fau.de)

Lehrstuhl für Informatik 12
Hardware-Software-Co-Design
Cauerstraße 11
91058 Erlangen

Friedrich-Alexander-Universität
Technische Fakultät

QoI Estimator Design Methodology for energy-efficient Image Processing on the Edge

In IoT, Quality of Information (QoI) is defined as 'an objective measure of the information utility which can be determined from the information object only'. On a sensor, the decision to transmit or process a collected sample can be made with regards to an application-specific QoI. Online evaluation of QoI can be a complex task, and, when performed on an energy-constrained device, must not consume more energy than how much can be saved based on QoI-aware data transmission/processing.



The goal of this thesis is to develop a QoI estimator design methodology for an image classification application. An estimator can be composed of different primitives, ranging from *Image Quality Assessment* methods (e.g., sharpness, dynamic range) to *explainable AI* (e.g., integrated gradient). The assembly of an estimator out of such primitives has an impact on its of latency, energy consumption and accuracy. As part of this thesis, design space exploration shall be investigated using multi-objective optimisation techniques to find Pareto-optimal estimators.

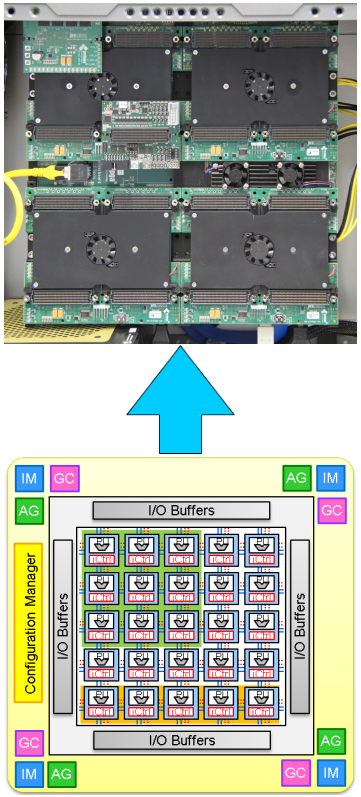
Prerequisites:	Basic knowledge in C++, Python, interest in Deep Learning and image processing
Type of work:	Theory (30%), Design (30%), Implementation (40%)
Supervisor:	Pierre-Louis Sixdenier (pierre-louis.e.sixdenier@fau.de)

Lehrstuhl für Informatik 12
Hardware-Software-Co-Design
Cauerstraße 11
91058 Erlangen

Friedrich-Alexander-Universität
Technische Fakultät

Automatic Partitioning of Large Processor Arrays onto a Multi-FPGA Platform

Tightly Coupled Processor Arrays (TCPAs) are a class of highly customizable hardware accelerators that are perfectly suited to accelerate loop applications like FIR filters, matrix-matrix-multiplications, and also of the popular Convolutional Neural Network (CNN) applications used in machine learning. A TCPA consists of an array of processors, while each processor can communicate data directly to its neighbor.
The goal of this work is to investigate and implement the segmentation of TCPA architectures that do not fit on a single FPGA (for example, with 100 or more processing elements) onto several FPGAs without violating timing properties between neighboring processing elements. The performance and energy consumption of multiple applications, e.g., from the digital signal or image processing domains, shall be investigated and compared between partitioned and non-partitioned TCPA implementations.



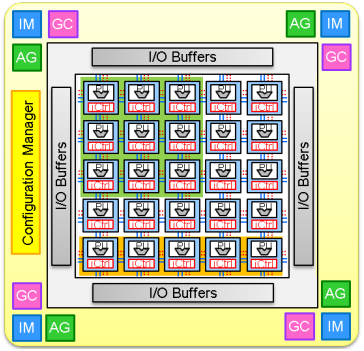
Requirements:	Knowledge in HDL programming mandatory
Type of thesis:	Theory (30%), concept (40%), implementation (30%)
Supervisor:	Marcel Brand (marcel.brand@fau.de)

Lehrstuhl für Informatik 12
Hardware-Software-Co-Design
Cauerstraße 11
91058 Erlangen

Friedrich-Alexander-Universität
Technische Fakultät

Latency and energy reduction of CNN inference by utilising multiply-accumulate functional units in a loop accelerator

Tightly Coupled Processor Arrays (TCPAs) are a class of highly customisable hardware accelerators that are perfectly suited to accelerate loop applications like FIR filters, matrix-matrix-multiplications, and also of the popular Convolutional Neural Network (CNN) applications used in machine learning. A TCPA consists of an array of processors, while each processor can communicate data directly to its neighbor. This design enables fast and low-power execution of such loop applications.



The goal of this work is to investigate how the acceleration of a CNN using a TCPA architecture may benefit from *vectorized multiply-accumulate floating-point functional units*, especially when they are run on low bit precisions. For CNN inference, a precision of 16 bits or less often suffices for computing a satisfactory result. This provides the chance for our TCPAs to be even more efficient by utilizing multiply-accumulate operations and vector processing. In this work, such a functional unit has to be designed, implemented, and integrated into a TCPA to investigate possible speed-ups compared to, e.g., implementations on a general purpose processor or TCPA architectures that utilise regular floating-point ALUs.

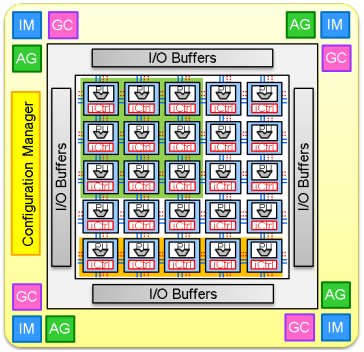
Requirements:	Knowledge in VHDL programming mandatory
Type of thesis:	Theory (30%), concept (30%), implementation (40%)
Supervisor:	Marcel Brand (marcel.brand@fau.de)

Lehrstuhl für Informatik 12
Hardware-Software-Co-Design
Cauerstraße 11
91058 Erlangen

Friedrich-Alexander-Universität
Technische Fakultät

Energy and performance improvements of CNNs on anytime processing elements by exploiting SIMD

Tightly Coupled Processor Arrays (TCPAs) are a class of highly customizable hardware accelerators that are perfectly suited to accelerate loop applications like FIR filters, matrix-matrix-multiplications, and popular Convolutional Neural Network (CNN) applications used in machine learning. A TCPA consists of an array of processors, while each processor can communicate data directly to its neighbor. This design enables fast and low-power execution of such loop applications.



The goal of this work is to analyse the benefits of combining the novel anytime processing elements with SIMD vector processing for floating-point computations. Anytime processing is a type of approximate computing that trades off the computation accuracy with performance and energy efficiency. In anytime processing, one can even control the accuracy of the instructions on bit level. SIMD vector processing, on the other hand, enables high parallelism with reduced control flow while executing applications.

We want to analyse whether the floating-point anytime functional units that are already integrated in the TCPA architecture can leverage the hardware structures of integrated vector processing and may further reduce the run-time of, e.g., the convolutions of a CNN application when computing at low accuracies.

Requirements:	Basic knowledge in hardware description languages (e.g., VHDL)
Type of thesis:	Theory (30%), concept (30%), implementation (40%)
Supervisor:	Marcel Brand (marcel.brand@fau.de)

Lehrstuhl für Informatik 12
Hardware-Software-Co-Design
Cauerstraße 11
91058 Erlangen

Friedrich-Alexander-Universität
Technische Fakultät