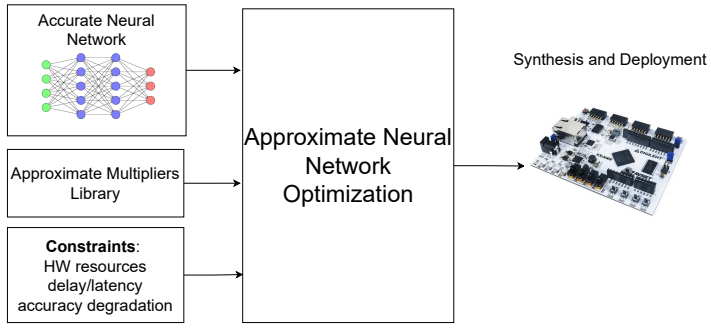
Themen für Projekt-, Bachelor- und Master-Arbeiten

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

Approximate Computing for TinyML Deployment in FPGAs

Embedded Machine Learning (ML) is a fast-growing field incorporating ML algorithms, hardware, and software to be deployed in resource-constrained devices. Approximate computing aims to tradeoff hardware and/or energy resources for inaccurate computations.



Leveraging the inherent error resilience of deep neural networks (DNNs), this work explores the use of approximate computing to enable efficient DNN deployment on small-scale FPGAs. The goal is to investigate approximate multipliers and adders that reduce resource usage and/or critical path delay, in combination with complementary optimization techniques such as mixed-precision quantization and weight pruning.
To mitigate the accuracy degradation introduced by these approximations, the thesis will study DNN adaptation strategies, including approximation-aware weight pruning rules and error correction methods tailored to the applied approximations. The project further involves the design, integration, and evaluation of approximate arithmetic hardware units within a DNN inference accelerator. This requires knowledge of digital hardware design and hardware description languages (HDLs) for FPGA-based deployment, potentially using existing frameworks such as hls4ml and FINN.

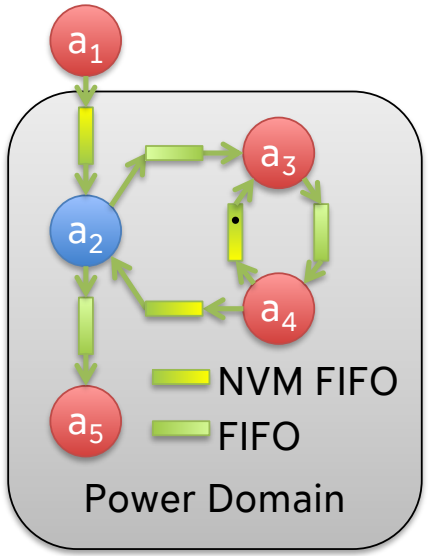
Prerequisites:	Knowledge in HDL (e.g. Verilog or VHDL), Python, and machine learning
Type of Work:	Theory (30%), Conception (20%), Implementation (50%)
Supervisor:	José Juan Hernández Morales (*jose.juan.hernandez*@fau.de)

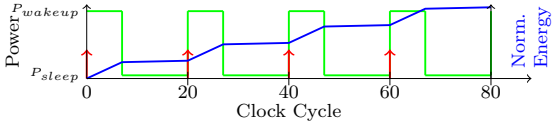
Lehrstuhl für Informatik 12
Hardware-Software-Co-Design
Cauerstraße 11
91058 Erlangen

Friedrich-Alexander-Universität
Technische Fakultät

Performance Optimization of Self-powering DFGs

Dataflow networks are ideally suited to model stream processing applications, and can be used in IoT devices signal for processing and compression. Recent research on *Self-powering Dataflow Networks* have shown that power savings can be achieved on such networks by powering down actors during periods of data unavailability using clock-gating and power-gating techniques. This is done by systematically transforming the FSM of each actor and introducing sleep states. However, this may lead to adverse effects on the throughput and even the total energy consumption of the network due to the additional latency incurred for sleep and wake-up transitions, as well as additonal power consumption due to the overhead circuitry introduced by the transform.





The goal of this thesis is to explore sleep and wake-up strategies, to optimize for throughput and energy consumption. Such strategies may include introducing optimal delays before going to sleep based on the dataflow specifications, and/or creating power domains containing multiple actors and channels to reduce additional circuitry overheads.

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

design

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

FAU
Friedrich-Alexander-Universität
Technische Fakultät

Reconfigurable Trusted Platform Modules (TPM) for the IoT

The Internet of Things (IoT) allows electronic products of our daily life to communicate sophistically, e.g., in digital factories or home automation applications. Yet, a big security threat is given by emerging attacks and that many systems not coming with a sophisticated security managing support.

In order to enable a longeveity of many products, it is evident that IoT components need to be upgradable. This not only involves the application software, but also the support of security essentials such as provided by a TPM (Trusted Platform Module).

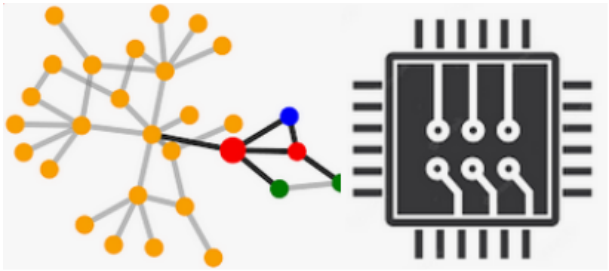
A TPM provides features for random number generation, an engine for encryption and signatures, e.g., RSA, cryptographic hash functions, and memories to store firmware and certificates, and key storage. On computers, TPMs are implemented today by a dedicated chip that is integrated on a main board. The goal of this thesis is to investigate co-design solutions for a reconfigurable TPU on FPGA. It shall be explored which parts of a TPU must be always resident, and which parts might even be only loaded when needed, e.g., during upgrades and updates. Based on the analysis, one suitable design shall be integrated on a real FPGA platform and evaluated for resource cost.

Prerequisites:	Basic knowledge VHDL, Knowledge Co-Design
Type of Work:	Theory (30%), Concepts (40%), Implementation (30%)
Supervisor:	Prof. Teich (juergen.teich@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:

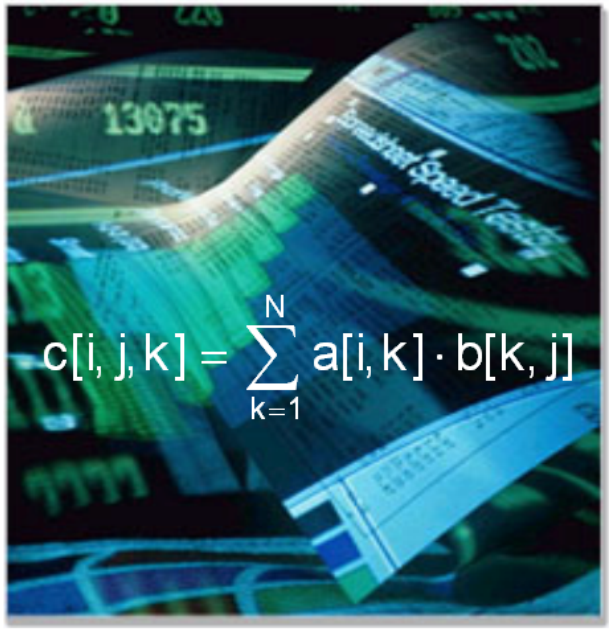
1. Reproduce the existing results.
2. Develop novel models to improve the prediction quality.
3. Propose additional benefits of applying ML for tasks related to IC synthesis.

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

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)

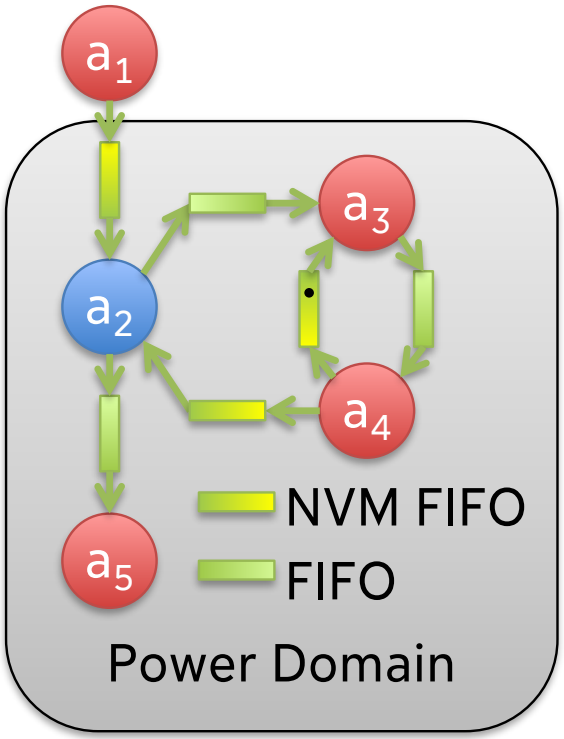
codesign

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.



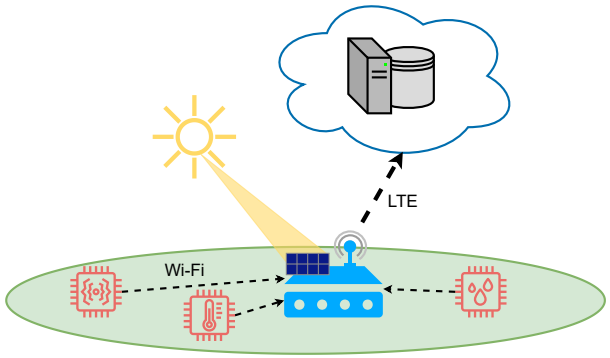
In this thesis, Non-Volatile Memory (NVM) should be exploited to realize dataflow networks in hardware to tackle these intermittency issues, e.g., by investigating and modeling persistable FIFO-based memory units. In particular, dataflow networks operating in mixed volatile/non-volatile operating modes 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

Acceleration of Edge Computing on eFPGAs in Wireless Sensor Networks



In Wireless Sensor Networks (WSNs), sensor nodes often have to perform complex processing tasks on sensed data for, e.g., data reduction, privacy or latency improvements. However, these nodes are usually heavily constrained in terms of available energy, processing power, and memory. To address these challenges, this thesis investigates offloading the processing of sensor data to an embedded FPGA (eFPGA) integrated into the sensor node. An eFPGA provides a reconfigurable, power-efficient, hardware platform that can be tuned to the specific processing tasks, potentially leading to performance and energy efficiency improvements compared to traditional microcontroller-based processing.

This work will involve designing and implementing a hardware accelerator on an eFPGA for a selected application (e.g., compression, audio processing). The thesis will also explore strategies for full dynamic reconfiguration of the eFPGA, enabling it to adapt to scarce energy resources. Hence, the hardware design must be parameterized, to allow trade-offs between energy consumption and accuracy at runtime.

Prerequisites:	Programming Knowledge in C/C++ and VHDL/Verilog
Type of Work:	Theory (20%), Conception (30%), Implementation (50%)
Contact:	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

Runtime Requirement Enforcement of Safety Properties of Human-Computer Interaction

Ensuring human safety is the most important factor within the area of human-robot interaction (HRI). E.g., robotic assistants should safely and flexibly cooperate with human operators. In addition, it must be guaranteed that they do not collide with humans and any obstacles in their shared environment.
Runtime Requirement Enforcement (RRE) has been proposed to ensure the satisfaction of system properties in the presence of uncertainty represented by the varying input from the environment. Finite state machines (FSMs) are used to model the enforcement strategy, which allows to perform formal verification and consequently provide safety guarantees.
In this thesis, an FSM-based RRE should be developed for enforcing a given set of safety requirements on simple HRI use cases such as a handover task. Formal verification (e.g., via PRISM verification tool) should be conducted to provide formal guarantees for the developed enforcement strategy to satisfy the given safety requirements.

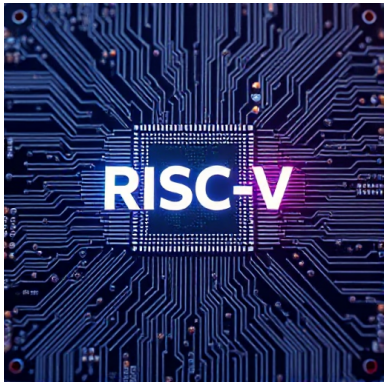
Voraussetzungen:	Programming knowledge in C++ or Java, knowledge in formal verification and temporal logic, basic knowledge in Robotics, basic knowledge in control
Art der Arbeit:	Theory (30%), Conception (35%), Implementation (35%)
Ansprechpartner:	Khalil Esper (khalil.esper@fau.de)

design

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

FAU
Friedrich-Alexander-Universität
Technische Fakultät

Efficient Hardware/Software Co-Design for Deploying Neural Networks on FPGAs



Different neural networks and machine learning algorithms have varying complexity, characterized by number of computations, memory requirements for storing model and intermediate data. To optimize efficiency, hardware configurations may be tailored to such specific workloads. The growing interest in open-source hardware and toolchains has led to the development of customizable RISC-V cores, which can be implemented on FPGAs as "soft-cores". These cores offer flexible configuration features like cache sizes, bus widths, and hardware extensions. However, the complex design space makes optimization challenging and requires a careful hardware/software co-design of neural network models and hardware architectures.

The primary goal is to develop a systematic design exploration strategy that effectively navigates the trade-offs between area, latency, and performance. The methodology involves deploying and profiling neural network workloads on soft-core RISC-V implementations to establish baseline performance metrics. Subsequently, the research will explore dedicated hardware accelerators and various hardware configurations, aiming to optimize the balance between resource utilization and computational efficiency.

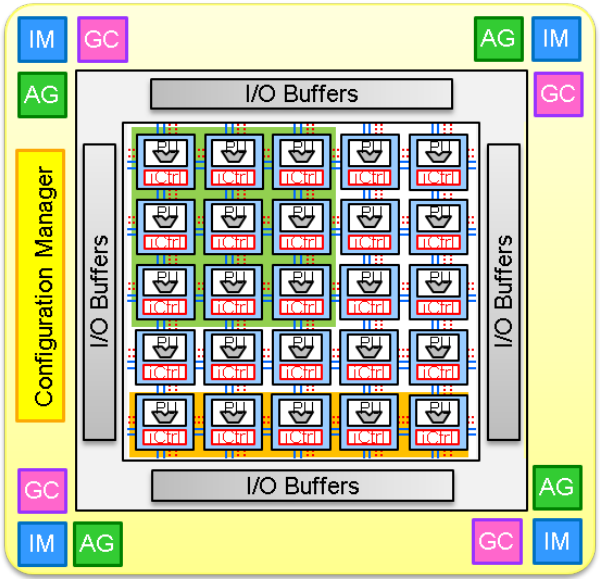
Prerequisites:	Knowledge in Python, C/C++, and hardware design
Type of Work:	Theory (10%), Conception (30%), Implementation (60%)
Supervisors:	Batuhan Sesli (batuhan.sesli@fau.de)

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

Friedrich-Alexander-Universität
Technische Fakultät

Automatische Exploration von Varianten eng gekoppelter Prozessorfelder

Tightly Coupled Processor Arrays (TCPAs) bestehen aus einem Feld von leichtgewichtigen Rechenelementen (PEs), die über ein rekonfigurierbares Netzwerk eng miteinander gekoppelt sind. TCPAs stellen eine ganze Klasse unterschiedlicher Beschleunigerarchitekturen dar, die über eine Menge von Parametern (Anzahl PEs, Speichergröße, usw.) definiert ist. Dabei hängen wichtige Architektureigenschaften wie die elektrische Leistungsaufnahme, Chipfläche und Rechenleistung oft direkt von den gewählten Parametern ab. Die manuelle Auswahl geeigneter Parameter ist jedoch sehr aufwendig und erfordert ein hohes Maß an Expertenwissen.



Ziel dieser Arbeit ist es daher, diesen Entwurfsraum so zu explorieren, dass für eine gegebene Anwendungsmenge eine maßgeschneiderte Zielarchitektur automatisch generiert werden kann.

Voraussetzungen:	Sehr gute Kenntnisse in C++- und Python-Programmierung
Art der Arbeit:	Theorie (10%), Konzeption (30%), Implementierung (60%)
Ansprechpartner:	Avinash Mahesh Nirmala (avinash.avi.mahesh@fau.de)
	Dominik Walter (dominik.l.walter@fau.de)
	Frank Hannig (frank.hannig@fau.de)

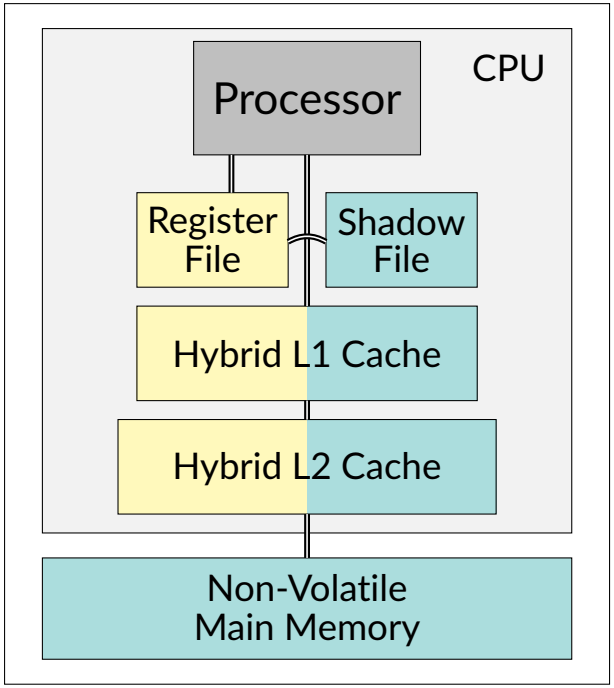
Lehrstuhl für Informatik 12
Hardware-Software-Co-Design
Cauerstraße 11
91058 Erlangen

Friedrich-Alexander-Universität
Technische Fakultät

Laufzeitanalyse hybrider Speicherzugriffsmuster

Computerspeicher teilt man schon lange in flüchtigen Hauptspeicher und persistente Festplatte. Softwarelösungen wie Betriebs- oder Datenbankverwaltungssysteme (OS bzw. DBMS) müssen die Grenze zwischen den beiden umsichtig behandeln, um Verlässlichkeit zu gewährleisten, besonders im Falle eines Stromausfalls. Hohe Schreibkosten beschränken nicht-flüchtigen Speicher (NVM) bisher auf Nischenanwendungen.

Diese Arbeit wird zur Entwicklung einer hybriden Architektur beitragen, die sowohl flüchtige als auch nicht-flüchtige Speichertechnologien in eine Cache-Hierarchie aufnimmt. Damit Software dieses Design effizient verwenden kann sollen Speicherregionen entsprechend der Art des Zugriffs gecached werden.



Methoden der Software-Laufzeitanalyse (Profiling) sollen identifiziert, evaluiert und erweitert werden, um bestimmte Speicherzugriffsmuster zu erkennen. Darüber hinaus sollen ein oder mehrere Profiler ausgewählt, auf eine Reihe an Benchmark-Programmen angewandt und ihre Ausgabe hinsichtlich ihrer Nutzbarkeit für einen Compiler analysiert werden.

Voraussetzungen:	Erfahrung mit C/C++ und Linux, Grundkenntnisse in Python empfohlen
Art der Arbeit:	Theorie (30%), Konzeption (50%), Implementierung (20%)
Betreuung:	Stefan Meißner (stefan.meissner@fau.de), Stefan Wildermann (stefan.wildermann@fau.de)

design

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

FAU
Friedrich-Alexander-Universität
Technische Fakultät