

Success Story: Software-driven CPU implementation

2022/11/28



Many application domains including automotive, industry automation, IoT, and space, require the usage of well-tailored edge devices capable of processing data from various sensor sources using AI, DSP, and classical software algorithms. Data processing at edge devices must satisfy real-time requirements while consuming as little electric energy and memory footprint as possible. Moreover, for many applications, the aspects of safety, security, and reliability are equally important as performance or electric energy consumption. Therefore, application-specific system-on-chip architectures for edge devices require high customization in terms of the most appropriate performance class of the processor core including necessary custom processor extensions, the memory architecture and capacity, and the design of a parameterizable AI accelerator architecture. The BMBF project Scale4Edge aims at enabling a comprehensive RISC-V-based ecosystem to efficiently assemble optimized edge devices.

By using the Scale4Edge ecosystem, the project partner Bosch developed a neural-network based audio event detection model. This use-case has been ported to a Pulpissimo-based SoC platform[1] using components and software of the Scale4Edge ecosystem.

The SoC platform used one of the MINRES RISC-V standard cores, named TGC_C, aiming for a small silicon area and allowing operating frequencies of between 20 MHz and 700 MHz. To this end, the decision was made to enhance the TGC_C with custom Instruction Set Architecture extensions (ISAX) supporting this type of neural networks. Bosch investigated the required extra functionality and specified a set of five useful instructions to add to the TGC_C ISA. These new instructions have been formulated in CoreDSL[2], a language to describe both base and extended ISAs, which serves as single-source-of-truth for downstream implementations. The language has a C-like syntax familiar to developers, allowing a high-level specification for each instruction's functionality. For a multiply-accumulate ISAX, the following CoreDSL description could be used:

```
MAC {
  encodings: 7'h00000000 :: rs2[4:0] :: rs1[4:0] :: 3'h010 :: rd[4:0] :: 7'h0101011;
  assembly: *(name[rd]), (name[rs1]), (name[rs2])[2];
  behavior: {
    signed<65> result = (signed)X[rs1] * (signed)X[rs2] + (signed)X[rd];
    if(rd != 0) X[rd] = result[31:0];
  }
}
```

Using this description MINRES generated an Instruction Set Simulator (ISS) supporting the custom instructions and delivered it within a day to Bosch to allow early performance validation. The ISS contains a static pipeline model and trace capabilities to allow hotspot analysis, e.g., using kcachegrind. Additionally, a tailored LLVM Clang version was created and delivered alongside a Board Support Package (BSP) and the standard software development toolchain for Bosch to quickly start adapting their algorithms and validate the performance gain using the ISAX extensions.

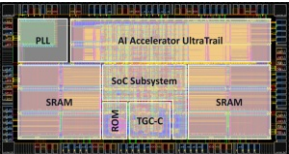
The toolchain and simulator artifacts were distributed for quick installation using the Docker containerization technology and enabled Bosch to begin performance validation with little bring-up effort. Using the MINRES ISS[3] and the included cycle estimation plugin, Bosch was able to quickly identify the acceleration potential of the custom instructions. The ISS trace output combined with the kcachegrind tool provided a holistic view of the executed software and enabled further performance gains by adding additional built-in functions into the tailored LLVM Clang, and rearranging code segments to optimize usage of the custom instructions added to the ISA. In this manner, an overall performance speed-up of around 50% was achieved.

In parallel, the CoreDSL description served as the specification for the RTL implementation of the actual hardware modules underlying the custom instructions. In the future, this hardware design process will be automated by Longnail, a domain-specific high-level synthesis tool developed by the project partner TU Darmstadt. Longnail leverages the state-of-the-art compiler frameworks MLIR and CIRCT to employ CoreDSL to generate hardware blocks and interfaces, all the way down to pipelined functional units described in synthesizable RTL SystemVerilog.

These functional units are automatically added to the TGC_C pipeline using the SCAIE-V scalable interface layer[4] and integration tooling, also developed at TU Darmstadt. MINRES applied extensive cross-level verification against the CoreDSL specification, using simulation and formal methods. The simulation approach was developed at University of Bremen to quickly achieve high coverage. For formal verification, Siemens EDA developed its OneSpin® RISC-V Verification App[5] that automatically detects all functional bugs and thus achieves very high functional quality. These verification innovations were complemented by standard approaches from the OpenHW group and RISC-V international. To quickly and thoroughly ensure functional safety, MinRes applied the OneSpin® functional safety apps[6]. The security of TGC was improved with the UPEC verification approach from the University of Kaiserslautern. After validating the performance achieved by the custom ISA extension by Bosch, and performing design sign-off, the customized CPU was integrated into the SoC by the University of Tübingen.

Especially the cross-level verification approach, developed in cooperation between MINRES and University of Bremen, allowed to achieve a high coverage very quickly. The verification environment uses -amongst other inputs- CoreDSL and the ISS as reference model, which ensures identical behavior for both. Additionally, verification approaches of the OpenHW group as well as compliance tests of the RISC-V International have been deployed in the TGC verification. The simulation-based verification has been augmented with formal verification using SiemensEDA OneSpin® 360 DV RISC-V Verification App⁵. After validating the performance achieved by the custom ISA extension by Bosch, and performing design sign-off, the customized CPU was integrated into the SoC by project partner University of Tübingen.

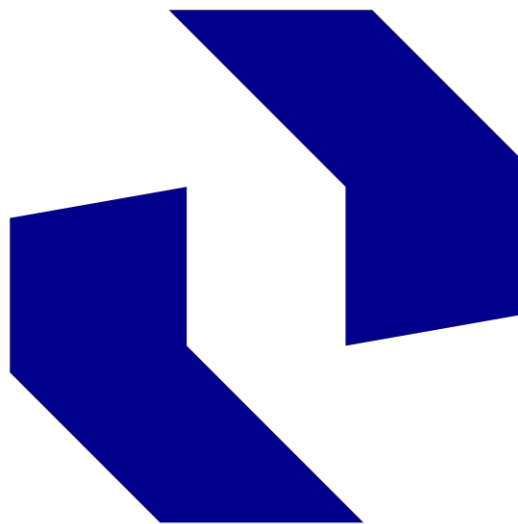
To demonstrate the audio event detection use-case, the platform was taped out by project partners University of Tübingen and Paderborn University using the 22 nm technology 22FDX by GlobalFoundries.



The platform integrates the ISA-extended TGC_C core by MINRES[7], the AI hardware accelerator UltraTrail by University of Tübingen, and a custom PLL by Paderborn University in a SoC tailored to the application requirements. The 2.5 mm x 1.35 mm large chip was designed for operating frequencies from 20 MHz to 700 MHz with support for dynamic frequency scaling, as well as clock and power gating methods. This enables real-time operation of the audio event detection model in a low-power domain. To achieve timing closure during implementation, some microarchitectural changes in the TGC_C were required. Thanks to the high level of design automation and the cross-level verification approach, those changes and their impact on the overall performance were quickly realized, allowing a timely tape-out. Both the chip and an evaluation board are currently in production and are expected to be operational in the first quarter of 2023.

Cont@ct: MINRES Technologies GmbH | Eyck Jentzsch | [eyck@minres\[dot\]com](mailto:eyck@minres[dot]com) | <https://www.edacentrum.de/scale4edge/> [8]

Further Scale4Edge partners and sub-contractors



Technische Universität Dresden

R
P
TU Rheinland-Pfälzische
Technische Universität
Kaiserslautern
Landau



[1] <https://github.com/pulp-platform/pulpissimo> [24]

[2] <https://github.com/Minres/CoreDSL> [25]

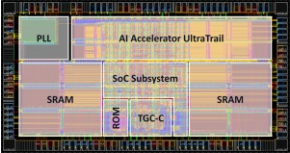
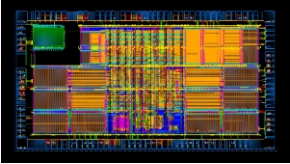
[3] <https://github.com/Minres/TGC-VP> [26]

[4] <https://github.com/esa-tu-darmstadt/SCAIE-V> [27]

[5] <https://www.onespin.com/risc-v/> [28]


[6] <https://www.onespin.com/fmeda> [29]

[7] <https://www.minres.com/products/the-good-folk-series/> [30]



```
MMC {
  encoding: 7'b0000000 :: ra2[4:0] :: ra1[4:0] :: 3'b010 :: rd[4:0] :: 7'b0101011;
  assembly: "{name(rd)}, {name(ra1)}, {name(ra2)}";
  behavior: {
    signed465> result = (signed)X[ra1] * (signed)X[ra2] + (signed)X[rd];
    if(rd != 0) X[rd] = result[31:0];
  }
}
```

Herunterladen:

 [Scale4Edge Success Story: Software-driven CPU implementation](#) ^[31]

Das Projekt Scale4Edge wird unter den Förderkennzeichen 16ME0122K-140, 16ME0465, 16ME0900, 16ME0901 im Förderprogramm ZuSE durch das deutsche Bundesministerium für Forschung, Technologie und Raumfahrt (BMFTR) gefördert.

Quell-URL: <https://project.edacentrum.de/scale4edge/success-story-software-driven-cpu-implementation>

Links:

- [1] <https://www.bosch.de>
- [2] <https://www.minres.com>
- [3] <https://eda.sw.siemens.com/en-US/>
- [4] <https://www.uni-bremen.de>
- [5] <https://www.tu-darmstadt.de>
- [6] <https://www.hni.uni-paderborn.de/sct/>
- [7] <https://uni-tuebingen.de/fakultaeten/mathematisch-naturwissenschaftliche-fakultaet/fachbereiche/informatik/fachbereich/>
- [8] <https://www.edacentrum.de/scale4edge/>
- [9] <https://www.AbsInt.com>
- [10] <https://uni-freiburg.de>
- [11] <https://www.arquimea.com/>
- [12] <https://www.concept.de>
- [13] <https://www.dlr.de>
- [14] <https://www.epos-d.com>
- [15] <https://www.fzi.de>
- [16] <https://www.ihp-microelectronics.com>
- [17] <https://www.infineon.com>
- [18] <https://www.sysgo.com>
- [19] <https://tu-dresden.de/>
- [20] <https://rptu.de/>
- [21] <https://www.ei.tum.de/eda/>
- [22] <https://www.lauterbach.com>
- [23] <https://www.edacentrum.de>
- [24] <https://github.com/pulp-platform/pulpassimo>
- [25] <https://github.com/Minres/CoreDSL>
- [26] <https://github.com/Minres/TGC-VP>
- [27] <https://github.com/esa-tu-darmstadt/SCAIE-V>
- [28] <https://www.onespin.com/risc-v/>
- [29] <https://www.onespin.com/fmeda>
- [30] <https://www.minres.com/products/the-good-folk-series/>
- [31] https://project.edacentrum.de/scale4edge/system/files/ct_project_news/scale4edge-success-story-audio-event-detection.pdf