



X-HEEP: An Open-Source, Configurable and Extendible RISC-V Microcontroller

Pasquale Davide Schiavone, Simone Machetti, Miguel Peón-Quirós, Jose Miranda, Benoît Denkingier, Thomas Christoph Müller, Ruben Rodríguez, Saverio Nasturzio, David Atienza Alonso
École Polytechnique Fédérale de Lausanne (EPFL)

Lausanne, Switzerland

(davide.schiavone,simone.machetti,miguel.peon,jose.mirandacalero,benoit.denkingier,christoph.mueller
ruben.rodriguezalvarez,saverio.nasturzio,david.atienza)@epfl.ch

CCS CONCEPTS

• Computer systems organization → Embedded systems.

KEYWORDS

computer architectures, risc-v, open-source, open hardware

ACM Reference Format:

Pasquale Davide Schiavone, Simone Machetti, Miguel Peón-Quirós, Jose Miranda, Benoît Denkingier, Thomas Christoph Müller, Ruben Rodríguez, Saverio Nasturzio, David Atienza Alonso. 2023. *X-HEEP: An Open-Source, Configurable and Extendible RISC-V Microcontroller*. In *20th ACM International Conference on Computing Frontiers (CF '23)*, May 9–11, 2023, Bologna, Italy. ACM, New York, NY, USA, 2 pages. <https://doi.org/10.1145/3587135.3591431>

1 INTRODUCTION

X-HEEP (eXtensible Heterogeneous Energy-Efficient Platform) is an open-source¹, configurable, and extensible single-core RISC-V microcontroller developed at the Embedded Systems Laboratory (ESL) of EPFL for edge-computing platforms. *X-HEEP* can be used standalone as a low-cost microcontroller, or it can be integrated into existing platforms to act like a peripheral subsystem, or it can be extended and customized with external peripherals and accelerators nimbly. The latter is particularly appealing for novel accelerators, memories, or peripherals designers who desire a simple controller to drive their IP and communicate with the external world using software functions. *X-HEEP* is built on top of existing, mature open-source IPs such as CPUs, peripherals, and many other building blocks from the OpenHW Group, the PULP team from ETH Zurich and the University of Bologna, and lowRISC. Its contribution includes its expandability, configurability, and agile use, targetting a large number of users to take one step further towards the democratization of open-source hardware.

X-HEEP is designed to support simulation with open-source and commercial simulators and target both FPGA and ASIC flows. The FPGA support can be useful for rapid prototyping and testing. *X-HEEP* comes with both the standalone implementation and the

¹*X-HEEP* is freely downloadable at <https://github.com/esl-epfl/x-heep> under a permissive license

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

CF '23, May 9–11, 2023, Bologna, Italy

© 2023 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-0140-5/23/05.

<https://doi.org/10.1145/3587135.3591431>

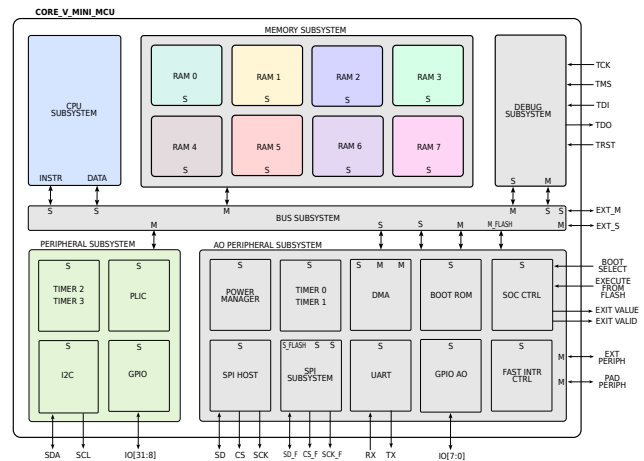


Figure 1: *X-HEEP* architecture.

ARM-hosted implementation. The latter exposes peripherals ports such as GPIOs, UART, JTAG, and SPI to the ARM subsystem, so that users can use the Linux subsystem running on the ARM CPU to emulate peripherals in SW without the need for external physical boards and cables, but still leveraging a large part of the *X-HEEP* SW stack and applications.

The *X-HEEP* architecture is presented in Figure 1. It is composed of RISC-V CPU, a system bus, SRAM memories, and two peripheral domains. It can be configured to select which CPU deploys among three (as of today) OpenHW Group CPUs as the cv32e40p [3], cv32e2 [5], and cv32e40x. The reason why *X-HEEP* employs such CPUs is that they are open-source, mature, verified, implemented in silicon several times, and designed to target edge devices. Depending on the target applications, the users can select the CPU that fits best the energy, area, power, and performance figures. The users can leverage *X-HEEP* to benchmark the core against their application profiles to better select their cores [5]. The CPUs are connected to 19 interrupts, compatible with the RISC-V CLINT spec: the 3 RISC-V machine-mode interrupts, and 16 custom events called FAST. In addition, a RISC-V PLIC peripheral collects a larger number of events which are then sent to a single CLINT line. Those interrupts are called SLOW as the CPU would first jump to the CLINT ISR, then reads the PLIC registers to figure out which peripheral raised the event, and only then serves it. Another *X-HEEP* knob the user can configure is the bus. As of today, *X-HEEP* offers two flavours: a fully-connected crossbar, and a one-at-a-time topology, both compatible with the OpenBus Interface, allowing each

slave to drive its own ready/valid signal to control its latency. The former bus flavour allows all the masters to access in parallel to different slaves, and the latter allows only one master at a time to access the slaves, so whenever two masters access concurrently two different slaves, one is stalled. The former is highly performant and large, as there are as many address decoders as masters, and the latter is low-cost and less performant. One of the most important parts of the microcontroller is the on-chip memory, which can represent up to 91% of the area of a chip [4]. Therefore it is important to let the user tune the memory size. As of today, *X-HEEP* can be configured to have a number of configurable 32kB memory banks, which are instantiated as bus slaves. Thus, the more memory banks, the more slaves, and the higher parallelism. The *X-HEEP* team is working on making the BUS topology and memory subsystem more configurable, allowing for different addressing modes (contiguous or interleaved), and the memory configurations knobs as the bank size. In addition to the CPU, bus, and memories, *X-HEEP* includes a wide range of peripherals, such as timers, PLIC and CLINT, a power manager, bootrom, DMA, JTAG, SPIs, UART, I2C, and GPIOs, mostly taken from the lowRISC OpenTitan and PULP project. The bootrom contains instructions to either wait for the JTAG to load the binary on the chip, or to load instructions from an external flash via SPI. In addition, *X-HEEP* embeds a BUS2SPI IP built on top of the Yosys SPI IP which translates memory read-operations to SPI, e.g. to allow the CPU to fetch instructions or load data from flash transparently, giving SW a lot of flexibility. The CPU can execute the program sitting directly on flash without first loading it on-chip at the cost of slower performance. This mode is particularly appealing for very small controllers with very limited on-chip memory, or to store wake-up routines during deep-sleep operations. As *X-HEEP* targets ultra-low-power applications, it employs a power manager responsible for implementing power-saving strategies such as operand isolation, clock-gating, and power-gating. *X-HEEP* is divided into power domains as always-on, CPU, peripheral, and one for each memory bank. The always-on domain is shown in grey in Figure 1, and includes components that control the chip externally, acquire data from external peripherals such as ADCs, and wake-up events. The peripherals' (green) and memory banks' (multi-colour) power domains can be switched on and off by means of register configurations written by SW, whereas the CPU power domain (blue) can be switched off by register configurations in SW, and switched on by DMA, SPI, TIMER, GPIOs, and SW events.

One of the key advantages of the *X-HEEP* microcontroller is that it is designed to be easily extensible with heterogeneous processing elements. Users can add their own custom peripherals and accelerators to it, allowing them to tailor the platform to their specific needs. For example, developers building a machine learning application may want to add a custom accelerator to the microcontroller to speed up matrix-matrix computations or a RISC-V Vector compatible co-processor to the CPU. To make extensions agile, *X-HEEP* exposes master and slave ports that have access to the main BUS, as well as the core-v-x interface that allows agnostic extensions to the cv32e40x CPU. The advantages are that the users can create their own system where *X-HEEP* is instantiated next to their custom blocks, without the need of forking and modifying *X-HEEP*. This prevents allocating extra human resources to maintain the *X-HEEP* forked version (as bug tracking, new features, etc), and to modify

the microcontroller HDL and framework infrastructure. We tested the agility of the extendability feature in the first ASIC implementation (in tsmc 65nm) deploying *X-HEEP* configured with the cv32e2 core and with 256kB of memory, called *HEEPocrates*. *HEEPocrates* instantiates *X-HEEP* as the main microcontroller driving a *CGRA* [2], an In-Memory Computing macro [6], and it is clocked by an FLL [1]. There was no need to modify *X-HEEP* for this specific tapeout, and the developers of the custom blocks inherit a whole microcontroller, saving time and resources in building one. The *X-HEEP* chip occupies an area of 2.3mm², has a maximum frequency of 250MHz, and consumes 28μW/MHz post-place-and-route.

X-HEEP flow is based on the lowRISC OpenTitan, using *vendor* to handle 3rd-party IPs, *regtool* to create peripherals, *mako templates* for highly configurable HDL files generation, and *FuseSoc* for describing the manifest file and generating tool-dependent scripts (e.g. Verilator, Questasim, Vivado, Design Compiler, etc.).

X-HEEP is fully compliant with RISC-V, thus the standard RISC-V GCC or LLVM can be used for compiling applications. It includes several application examples, a *HAL*, and *FreeRTOS* support, which is the most used open-source real-time operating system.

X-HEEP is a work in progress, and we expect to expand the number of configuration knobs, as well as expand the peripheral and CPUs supported.

In conclusion, the *X-HEEP* microcontroller is a powerful and flexible platform for building embedded systems. Its open-source design, easy configurability, and ability to support custom extensions make it an attractive option for SoC designers. With its set of peripherals and expansion headers, *X-HEEP* is well-suited for use in a wide range of applications, from simple sensors to complex control systems.

2 ACKNOWLEDGMENTS

The development of the *X-HEEP* framework has been supported in part by Ecocloud, the EPFL research center on sustainable computing, and by the Swiss NSF ML-Edge Project (GA No.200020 182009).

REFERENCES

- [1] David E Bellasi and Luca Benini. 2017. Smart energy-efficient clock synthesizer for duty-cycled sensor soics in 65 nm/28nm cmos. *IEEE Transactions on Circuits and Systems I: Regular Papers* 64, 9 (2017), 2322–2333.
- [2] Loris Duch, Soumya Basu, Rubén Braojos, David Atienza, Giovanni Ansaloni, and Laura Pozzi. 2016. A multi-core reconfigurable architecture for ultra-low power bio-signal analysis. In *2016 IEEE Biomedical Circuits and Systems Conference (BioCAS)*. IEEE, 416–419.
- [3] Michael Gautschi, Pasquale Davide Schiavone, Andreas Traber, Igor Loi, Antonio Pullini, Davide Rossi, Eric Flamand, Frank K Gürkaynak, and Luca Benini. 2017. Near-threshold RISC-V core with DSP extensions for scalable IoT endpoint devices. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 25, 10 (2017), 2700–2713.
- [4] Alfio Di Mauro, Francesco Conti, Pasquale Davide Schiavone, Davide Rossi, and Luca Benini. 2020. Always-On 674u W@4GOP/s Error Resilient Binary Neural Networks With Aggressive SRAM Voltage Scaling on a 22-nm IoT End-Node. *IEEE Transactions on Circuits and Systems I: Regular Papers* 67, 11 (2020), 3905–3918. <https://doi.org/10.1109/TCSI.2020.3012576>
- [5] Pasquale Davide Schiavone, Francesco Conti, Davide Rossi, Michael Gautschi, Antonio Pullini, Eric Flamand, and Luca Benini. 2017. Slow and steady wins the race? A comparison of ultra-low-power RISC-V cores for Internet-of-Things applications. In *2017 27th International Symposium on Power and Timing Modeling, Optimization and Simulation (PATMOS)*. IEEE, 1–8.
- [6] William Andrew Simon, Yasir Mahmood Qureshi, Marco Rios, Alexandre Levisse, Marina Zapater, and David Atienza. 2020. BLADE: An in-cache computing architecture for edge devices. *IEEE Trans. Comput.* 69, 9 (2020), 1349–1363.