

X-HEEP: An Open-Source, Configurable and Extendible RISC-V Microcontroller for the Exploration of Ultra-Low-Power Edge Accelerators

SIMONE MACHETTI, Embedded Systems Laboratory (ESL), EPFL, Switzerland

PASQUALE DAVIDE SCHIAVONE, Embedded Systems Laboratory (ESL), EPFL, Switzerland

THOMAS CHRISTOPH MÜLLER, Embedded Systems Laboratory (ESL), EPFL, Switzerland

MIGUEL PEÓN-QUIRÓS, EcoCloud, EPFL, Switzerland

DAVID ATIENZA, Embedded Systems Laboratory (ESL), EPFL, Switzerland

The field of edge computing has witnessed remarkable growth owing to the increasing demand for real-time processing of data in applications. However, challenges persist due to limitations in the performance and power efficiency of edge-computing devices. To overcome these challenges, heterogeneous architectures have emerged that combine host processors with specialized accelerators tailored to specific applications, leading to improved performance and reduced power consumption. However, most of the existing platforms lack configurability and extendability options, necessitating extensive modifications of the register transfer level (RTL) code for integrating custom accelerators.

To overcome these limitations, we introduce in this paper the eXtendible Heterogeneous Energy-Efficient Platform (X-HEEP). X-HEEP is an open-source platform designed to natively support the integration of ultra-low-power edge accelerators. It provides customization options to match specific application requirements by exploring various core types, bus topologies, and memory addressing modes. It also enables a fine-grained configuration of memory banks to match the constraints of the integrated accelerators. The platform prioritizes energy efficiency by implementing low-power strategies, such as clock-gating and power-gating, and integrating these with connected accelerators through dedicated power control interfaces.

We demonstrate the real-world applicability of X-HEEP by providing an integration example tailored for healthcare applications that includes a coarse-grained reconfigurable array (CGRA) and in-memory computing (IMC) accelerators. The resulting design, called HEEPocrates, has been implemented both in field programmable gate arrays (FPGAs) on multiple Xilinx chips, for prototyping and exploration, and in silicon with TSMC 65 nm low-power CMOS technology. The fabricated chip can operate from 0.8 V to 1.2 V, achieving a maximum frequency of 170 MHz and 470 MHz, respectively. Its power consumption ranges from 270 μ W at 32 kHz and 0.8 V, to 48 mW at 470 MHz and 1.2 V.

We run a set of healthcare applications and measure their energy consumption to demonstrate the alignment of our chip with other state-of-the-art microcontrollers commonly adopted in this domain, showing that HEEPocrates provides a good trade-off between acquisition-dominated and processing-dominated applications for energy efficiency. Moreover, we present the energy benefits of 4.9 \times and 4.8 \times gained by exploiting the integrated CGRA accelerator and IMC accelerator, respectively, compared to running on the host CPU.

Additional Key Words and Phrases: Ultra-Low Power, Energy Efficiency, Microcontroller, Accelerator, Field Programmable Gate Array (FPGA), Course-Grained Reconfigurable Array (CGRA), In-Memory Computing (IMC), Tapeout, Silicon Validation.

Authors' addresses: **Simone Machetti**, Embedded Systems Laboratory (ESL), EPFL, Lausanne, Switzerland; **Pasquale Davide Schiavone**, Embedded Systems Laboratory (ESL), EPFL, Lausanne, Switzerland; **Thomas Christoph Müller**, Embedded Systems Laboratory (ESL), EPFL, Lausanne, Switzerland; **Miguel Peón-Quirós**, EcoCloud, EPFL, Lausanne, Switzerland; **David Atienza**, Embedded Systems Laboratory (ESL), EPFL, Lausanne, Switzerland.

Permission to make digital or hard copies of all or part 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 components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2024 Association for Computing Machinery.

Manuscript submitted to ACM

ACM Reference Format:

Simone Machetti, Pasquale Davide Schiavone, Thomas Christoph Müller, Miguel Peón-Quirós, and David Atienza. 2024. X-HEEP: An Open-Source, Configurable and Extendible RISC-V Microcontroller for the Exploration of Ultra-Low-Power Edge Accelerators. 1, 1 (March 2024), 21 pages. <https://doi.org/10.1145/nnnnnnnn.nnnnnnnn>

1 INTRODUCTION

In recent years, the field of edge computing has witnessed remarkable growth and adoption in commercial products. This process has been driven by the increasing demand for real-time computing solutions, particularly Artificial Intelligence (AI) and Machine Learning (ML) algorithms. As data processing at the edge for new edge AI computing has become more prevalent, the performance and power consumption limitations of edge-computing devices have become increasingly apparent, which has posed significant challenges for researchers and engineers.

Heterogeneous architectures have emerged to overcome these challenges. These architectures offer a promising path toward high energy efficiency while maintaining performance constraints. Heterogeneous architectures rely on a combination of ultra-low-power host processors to run control and communication tasks, and custom accelerators tailored to specific application domains, such as artificial intelligence, image processing, healthcare, and cryptography, to run computationally demanding tasks.

Control and communication tasks include accessing external memories, acquiring data from analog-to-digital converters (ADCs) or sensors, preparing data for computations, and running real-time operating system (RTOS) functions. Meanwhile, computational demanding tasks focus on data processing, for example, convolutional and fully connected layers in neural networks (NNs), fast-Fourier transforms (FFTs) in temporal series, secure-hash algorithms (SHA) in cryptography, etc.

Each accelerator comes with unique requirements, such as memory size, area, performance, and power, to meet the constraints of the target applications. For this reason, proper customization of host platforms is imperative. This may include exploring different CPUs to trade performance and power, bus topologies and memory hierarchy, memory sizes to accommodate the required computational data, peripherals to provide the necessary I/O connectivity, power domains and strategies, etc. However, commercial platforms limit hardware exploration due to their non-open-source nature. They often involve costly licensing models and do not allow for customization. As a result, there is a growing preference for open-source platforms as a more attractive solution that does not limit exploration and customization, and that gives designers digital sovereignty and control over IPs.

Today, there are an increasing number of open-source projects related to heterogeneous systems, thanks to the open RISC-V instruction set architecture (ISA) revolution. However, many of such platforms focus only on the CPU part, whereas microcontroller-based state-of-the-art projects lack the flexibility and customization options needed to fulfill accelerator requirements natively. These limitations include restricted configurability for the internal platform's components (core, memory, bus, etc.) to adapt to the application needs, limited support for external accelerator connectivity to communicate with the host system, and inadequate built-in power management strategies to optimize energy efficiency. Thus, hardware developers need to extensively modify the platform to properly align with the target applications on their own copy of the platform. This includes forking, modifying, and maintaining the forked platform's repository, leading to high maintenance costs. Therefore, addressing the configurability and extendability aspects of these platforms is crucial to lowering the adoption barrier of open-source-based edge solutions.

In this paper, we address the limitations mentioned above by introducing X-HEEP¹ [30], an open-source configurable and extendable platform designed to support the exploration of ultra-low power edge accelerators. X-HEEP is a streamlined configurable host architecture based on RISC-V and built on top of existing IPs from relevant open-source projects, such as the PULP project, the OpenHW Group, and the OpenTitan project, as extensively verified, validated in silicon, and adopted in commercial products. This allows extensive reuse of third-party hardware and software extensions and inheriting verification and silicon validation.

To allow users to explore their custom solutions, X-HEEP can be natively extended via the proposed eXtendible Accelerator InterFace (XAIF), which allows the integration of a wide range of accelerators with different area, power, and performance constraints. Having a complete interface that covers all the edge accelerator performance and power requirements will enable extensive reuse of hardware and software IPs, reducing costs and mitigating fragmentation. To explore the custom hardware design space, users will use X-HEEP as an IP which exposes an interface capable of addressing all the edge-computing state-of-the-art requirements for domain-specific applications. To enable a high degree of versatility, such exploration can be performed both on FPGAs or RTL simulators, as well as SystemC for a mixed high-level and RTL simulation environment. Additionally, to offer high degree of optimizations, X-HEEP offers internal configurability options through the selection of different (1) core types, depending on the target workloads [29]; (2) bus topology and addressing mode, ensuring a perfect match with the bandwidth requirements of the integrated accelerators; (3) memory size, depending on the processing data and application complexity; and finally (4) peripherals, to provide the needed I/O connectivity. This configurability enables designers to tailor the platform to specific application requirements and meet area, power, and performance constraints.

As energy efficiency is a key figure in edge computing devices, X-HEEP implements state-of-the-art fine-grained low-power strategies such as clock-gating, power-gating, and RAM retention, which are integrated into the XAIF interface to be leveraged by the connected accelerators and maximize overall energy efficiency.

To demonstrate the real-world applicability of X-HEEP, we present an integration example specifically tailored for ultra-low-power healthcare applications. These applications typically involve long and slow acquisition periods, where data from external bio-sensors are stored in memory while the rest of the system is in an idle state, followed by intense processing periods to compute pattern extraction algorithms based on digital signal processing algorithms, or machine learning (ML), and deep learning. Therefore, we extended X-HEEP with a CGRA accelerator [9] and an IMC accelerator [31], both of which have been shown to efficiently reduce the overall energy consumption of healthcare applications [7, 24]. We configured X-HEEP with the RISC-V OpenHW Group CV32E20 core [29], 8 banks of 32 KiB on-chip SRAM organized in a contiguous addressing mode, and 11 different power domains (including the external accelerators) that can be individually switched on and off for fine-grained power control.

The resulting design, called HEEPocrates, has been implemented both in FPGAs on the Zynq 7020, Zynq UltraScale+, and Artix 7 chips by Xilinx, for early prototyping, verification, and system exploration, as well as in silicon with TSMC 65 nm low-power CMOS technology, for silicon validation and profiling performance, power, and area figures. The measured performance of the fabricated chip shows that it can operate in a wide voltage range, from 0.8 V to 1.2 V, achieving a maximum frequency of 170 MHz and 470 MHz, respectively. Its power consumption ranges from 270 μ W at 32 kHz and 0.8 V, to 48 mW at 470 MHz and 1.2 V.

To validate our design and compare it with state-of-the-art solutions, we measured the energy consumption of the chip in a set of healthcare applications, showing that it offers a good trade-off between the computationally hungry

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

and acquisition-dominated state-of-the-art microcontrollers commonly adopted in this domain. This demonstrates the flexibility of our host platform in adapting to the specific needs of integrated accelerators and matching the strict requirements of healthcare applications. Furthermore, we present the energy benefits of $4.9 \times$ and $4.8 \times$ gained by exploiting the integrated CGRA accelerator and the IMC accelerator, respectively, compared to running on the host CPU.

Throughout this work, we will dive deeper into the features of the X-HEEP platform, describing its architecture, configurability, and extendability options to build versatile and energy-efficient edge applications.

The following contributions are presented:

- X-HEEP: A configurable and extendible RISC-V microcontroller to support the exploration of ultra-low-power edge accelerators.
- XAIF: A configurable interface that adapts to the different requirements of accelerators in terms of programmability, bandwidth, interrupts, and power modes, and that allows their seamless integration into the X-HEEP architecture.
- HEEPocrates: A real-world integration example, based on TSMC 65 nm low-power CMOS technology, that includes a CGRA and an IMC accelerator.
- An open-source repository with a permissive license with the complete X-HEEP platform code and documentation to allow researchers to explore new custom accelerators and advance research in this field.

The remainder of this paper is structured as follows. In Section 2, we conduct an in-depth analysis of the most relevant state-of-the-art accelerators and host platforms. In Section 3, we provide a qualitative and quantitative description of the configurability and extendability features of X-HEEP. In Section 4, we present a real-world integration example, called HEEPocrates. In Section 5 we describe our experimental setup, while in Section 6 our experimental results. Lastly, in Section 7, we offer a comprehensive summary of the main conclusions of our work.

2 STATE-OF-THE-ART

This section gives an overview of cutting-edge accelerators, analyzing the fundamental requirements necessary for their integration into host platforms. Such requirements are collected into the XAIF to accommodate all the state-of-the-art accelerators. Subsequently, it focuses on host platforms from the state of the art, conducting an evaluation of their strengths and limitations in terms of configurability and extendability.

2.1 Edge-computing accelerators

The extensive array of open-source accelerators includes a diverse range of requirements regarding memory capacity, area, performance, and power efficiency. Therefore, an analysis of their main features becomes imperative for the design of flexible and efficient host platforms.

We can divide accelerators into three main categories: memories, processors (and co-processors), and I/O peripherals. One accelerator can belong to one or more categories.

2.1.1 Memories. Memory accelerators are a class of IPs that feature one or more slave ports to access internal functionality. These IPs require the host CPU, or a DMA, to copy the needed computational data from the main memory of the platform to their internal data memory, before starting the operations. At the end of the computations, an interrupt or status bit could be used to synchronize with the host CPU.

An example is the Keccak accelerator presented in [8]. This accelerator exposes two 32 bit slave ports, one to access the internal register file used for control and status operations and one to access the private data memory, which stores the processing data.

Other examples are in-memory or near-memory macros such as the C-SRAM [15], where the IP is connected via a 32 bit slave port to a host platform that sends commands/instructions to the memory through write operations. C-SRAM decodes the memory instructions, by concatenating the address and the write data transmitted by the CPU, and performs the requested operation.

2.1.2 Processors. To improve performance, many accelerators feature one or multiple master ports to independently read in parallel the processing data and write back the generated results from/to the main memory.

Some examples are domain-specific accelerators, such as DSP engines [16], CGRAs [9], multi-CPU clusters [25], GPUs [33], etc., and application-specific accelerators for neural networks [5], FFT [32], cryptography [8], image processing [20], etc.

An example of a domain-specific accelerator is the CGRA presented in [9], which has two 32 bit slave ports for configuration registers and private instruction memory and four 32 bit master ports for reading and writing data from and to the main memory, reaching a maximum bandwidth of 128 bit per bus cycle.

Another example is the PULP cluster [25], which features four to eight CV32E40P cores [12] connected to a shared instruction cache and a multi-bank scratchpad memory. The cluster exposes one 32 bit slave port for configuration and for pre-loading the memories, and one 64 bit master port shared between the cluster DMA, to transfer data in and out of the scratchpad memory, and the instruction cache, to fetch program code.

Examples of application-specific accelerators are Echoes [32] and Marsellus [13]. The former is used to speed up FFT execution and has eight 32 bit master ports, four allocated for input and four for output, reaching a maximum bandwidth of 256 bit per bus cycle. The latter accelerates convolution layers and offers nine 32 bit master ports, with a maximum bandwidth of 288 bit per bus cycle.

Co-processors are a sub-category of processors used to implement custom ISA extensions. Co-processors are either tightly coupled in the processor pipeline, or integrated via a dedicated interface for reusability.

Their wide application domains include floating-point operations [3], posit arithmetic [18], post-quantum cryptography [10], integer complex arithmetic [34], etc. For example, [10] proposes a post-quantum cryptography ISA extension and interacts with the coupled RISC-V CPU thanks to the CORE-V-XIF [6] interface of OpenHW Group.

All these previous examples illustrate that there is a large choice of possible processors and co-processors for edge AI systems today. Therefore, it is required to have a fast and scalable exploration and prototyping framework to choose the right set of components, co-processors, or domain-specific accelerators that a final implementation should have, and then a well-tuned silicon design flow with a predefined set of open-source hardware components and peripherals.

2.1.3 I/O peripherals. These IPs are meant to implement special interfaces to communicate with off-chip components or to pre/post-process data during such communications.

An example can be found in Arnold [27], where the embedded FPGA (eFPGA) can be used to control an off-chip accelerator, which requires a custom interface, as well as to pre-process data coming from peripherals before being stored in memory.

In this case, there is a clear need to ideally have a framework that can target both on-chip and off-chip accelerator concepts by enabling a flexible set of interconnect standards. This set of standards should be extendable with minimum

effort from the system designer thanks to an interface that enables a superset of interconnection protocols, as we propose in X-HEEP.

2.2 Host platforms

In this subsection, we present a comparison of relevant open-source platforms that can be used to host edge-computing accelerators, focusing on configurability, extendability, and other key features. The limitations of each platform are analyzed in detail to motivate the need for a dedicated solution that can fulfill all the requirements.

2.2.1 *PULPissimo* [28]. A single-core platform within the PULP family, designed to target ultra-low-power edge-computing applications. Depending on performance requirements, designers can configure the platform with the CV32E20 [29] or CV32E40P [12] cores. PULPissimo has been integrated with various accelerators, including the aforementioned multi-CPU cluster [25], neural network accelerators [13], CGRAs [9], eFPGAs [27], etc. Many silicon prototypes have been implemented, which demonstrate best-in-class energy efficiency in a wide range of applications.

However, PULPissimo provides only a generic AXI 32 bit slave and a 64 bit master external interfaces that are used to connect the multi-CPU cluster, while the other accelerators have been integrated by forking and modifying the original RTL code. Such external interfaces may limit accelerators' bandwidth. Moreover, the platform lacks native support for external interrupts/events and power control, which is crucial for efficient power management. Lastly, the platform does not offer configurability options to select memory size, bus topology, and memory addressing mode, or to change the included peripherals, which limit area, bandwidth, and power-space exploration.

2.2.2 *Cheshire* [22]. The limitations mentioned above have been partially addressed by another PULP-based platform, called Cheshire. Cheshire is based on the CVA6 core [35] and allows designers to choose the number of external slave and master ports to connect their custom accelerators. Furthermore, the platform allows for the configuration of the internal last-level cache (LLC) size and of the necessary peripherals, providing the flexibility needed to target specific application requirements.

However, Cheshire has been designed for high-performance systems and consumes up to 300 mW, making it unsuitable for most ultra-low-power devices, which typically operate in the range of tens of mW. Furthermore, Cheshire lacks support for external interrupts and power control, which has implications for its overall energy efficiency, as the accelerators are usually power-hungry. Lastly, designers do not have the option to select the core type, bus topology, and memory addressing mode.

2.2.3 *BlackParrot* [23]. An open-source Linux-capable platform designed to accommodate one or multiple custom-designed accelerators. The platform showcases a mesh of heterogeneous tiles, offering the flexibility to compose 64 bit BlackParrot cores, L2 cache slices, I/O, DRAM controllers, and accelerators in various configurations.

However, it does not allow for selecting the core type, bus topology, and memory addressing mode. Additionally, the absence of essential peripherals commonly used in edge devices, such as I2Cs, GPIOs, timers, DMAs, interrupt controllers, and a power manager to implement low-power strategies, restricts the usage of the platform for real applications deployed on ultra-low-power edge applications. Moreover, the platform's internal integration of accelerators, as opposed to external plug-ins, involves forking and modifying the original RTL code, leading to greater effort and higher development costs. Lastly, the 64 bit architecture of BlackParrot targets high-performance systems and is unsuitable for ultra-low-power edge devices.

2.2.4 OpenTitan [17]. OpenTitan is designed for ultra-low-power edge-secure applications. It offers a single-core architecture based on the CV32E20 [29] core and an extensive portfolio of peripherals.

Despite these strengths, OpenTitan does not offer external support for accelerator plug-ins, requiring designers to manually modify the RTL code to integrate their custom accelerators. Furthermore, the platform lacks configurability for core type, bus topology, and memory addressing mode and size. Furthermore, OpenTitan does not come equipped with built-in low-power strategies.

2.2.5 Chipyard [1]. On the contrary, the Rocket chip generator [26], which has been subsequently incorporated and expanded into the Chipyard platform, offers extensive configuration options. Using the open-source Chisel hardware description language, designers can craft their system, providing flexibility and customization. The platform offers a wide range of core types, including Ariane, CV32E20, Rocket, and BOOM, allowing designers to tailor the system's performance to meet specific application requirements. Additionally, the memory size and peripherals can be customized, further enhancing its adaptability.

However, even though Chipyard enables accelerators to be integrated into the design using the Chisel language, the platform does not offer external master and slave ports for the connectivity of accelerators. As a result, designers need to invest time in becoming familiar with the Chisel language to successfully configure the architecture and integrate custom accelerators. Furthermore, Chipyard does not provide support for any specific power reduction strategies. Given the critical importance of power efficiency in ultra-low-power applications, designers are forced to implement power-saving techniques manually to achieve the desired energy efficiency level.

2.2.6 LiteX [14] and ESP [19]. Two other notable SoC generators are LiteX and ESP. LiteX serves as a framework thought to explore various FPGA-based architectures. On the other hand, ESP is an open-source platform designed for heterogeneous SoC design and prototyping on FPGAs. Both platforms offer configurable options, allowing designers to customize core type, memory size, peripherals, and the number of external master and slave ports, making them adaptable to various application requirements.

However, LiteX and ESP focus on FPGA development only and do not offer support for ASIC design flow. Such limitations hinder their applicability in projects aimed at silicon implementations and present difficulties in accurately estimating the platform energy consumption, crucial when evaluating the impact of integrated accelerators. Moreover, they lack built-in support for external interrupts and power control, essential for efficient power management.

2.2.7 X-HEEP. To overcome the limitations mentioned above and cater to the unique needs of ultra-low-power edge designers, we present in this paper the X-HEEP platform. The proposed platform features a streamlined architecture that operates in conjunction with dedicated open-source tools. These tools enable developers to easily customize and extend the architecture with their accelerators and interconnection interfaces, thus eliminating the need for manual modification of the RTL code. Using X-HEEP, designers can achieve the desired level of configurability, extendability, and power efficiency, making it an ideal choice for a wide range of ultra-low-power edge applications. In addition, it has been developed using SystemVerilog, to offer high compatibility with most of the available electronic design automation (EDA) tools.

3 X-HEEP

In this section, we present a qualitative and quantitative analysis of the key features of X-HEEP regarding configurability, extendability, and software support. We synthesized X-HEEP with TSMC 65 nm low-power technology and performed our quantitative analysis at the nominal voltage, 1.2 V.

3.1 Architecture

Figure 1 shows the X-HEEP architecture and its essential components. These include a configurable RISC-V CPU, a configurable bus, a configurable memory, two configurable peripheral domains, and a debug unit.

X-HEEP leverages existing widely adopted open-source IPs to maintain compatibility with existing systems and reuse available software routines and hardware extensions. Among the wide portfolio of open-source IPs, we selected those that provide permissive licenses, to ease the X-HEEP adoption to a wide range of users, and written in SystemVerilog, to make the integration in existing systems and EDA tools compatible with industrial standards.

The RISC-V cores have been selected from the OpenHW Group CORE-V family, as extensively verified, mature, and implemented in silicon many times; the bus, the memory models, the debug unit, and a plethora of IPs from the PULP project, as again adopted by several stakeholders and validated in silicon multiple times; and the peripherals from the OpenTitan project as documented, verified, and inclusive of hardware-abstraction-layer (HAL) functions. Moreover, X-HEEP includes home-made IPs such as a boot ROM, a power manager, a fast interrupt controller, and a DMA.

3.1.1 CPU. The user can choose among the CV32E20, CV32E40X, and CV32E40P as core options [29], to trade off power and performance. In particular, the CV32E20 core is optimized for control-oriented tasks, while the CV32E40P core is optimized for processing-oriented tasks. The CV32E40X core offers power consumption and performance similar to the CV32E40P core, without featuring the floating-point RVF and custom Xpulp ISA extensions. Moreover, it provides an external interface, known as CORE-V-XIF [6], that allows for the plug-in of custom co-processors to extend the RISC-V ISA without the need to modify the RTL code of the core.

3.1.2 Memory. The user can select the memory size and number of memory banks to trade off area, power, and storage capacity. Each bank offers a retention state aimed at reducing leakage power, of about 42.5 % compared to active leakage, when the bank is not accessed for some time but the data needs to be preserved.

3.1.3 Bus. To maximize compatibility with the other IPs selected from the OpenHW Group, PULP platforms, and OpenTitan project, the bus is based on the same open-bus interface (OBI) [21] protocol.

The user can choose either a one-at-a-time topology, where only one master at a time can access the bus (one decoder), or a fully connected topology (same number of decoders as simultaneous masters), where multiple masters can access multiple slaves in parallel, to trade off area and bandwidth. When the fully connected option is used, the user can further configure the bus to access a variable number of banks in a contiguous or interleaved addressing mode. The contiguous mode offers limited bandwidth to applications that require multiple masters to access contiguous data stored in memory but allows for power-gating or setting in retention mode the banks that are not actively used. Vice versa, the interleaved mode offers higher bandwidth to applications that access contiguous data in memory, at the cost of keeping all the banks active all the time.

In addition, to connect additional components, the bus also exposes a configurable number of slave and master ports to the external XAIF interface to accommodate one or multiple accelerators with different bandwidth constraints.

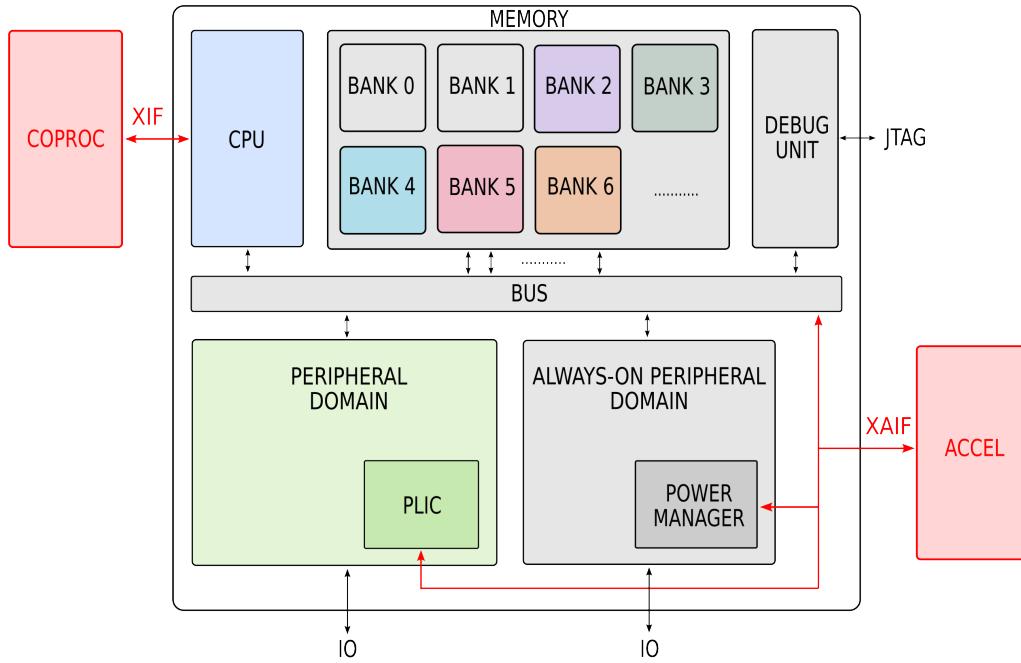


Figure 1. *X-HEEP* architecture. The various power domains are visually marked using different colors. The components in grey are always on. The accelerator and co-processor integration are highlighted in red.

Figures 2 (a) and (b) show the variation in the area and the bandwidth of the X-HEEP bus by adding slave/master ports to the basic bus configuration, which connects the CV32E20 core, two memory banks, the debug unit, the two peripheral domains, and no external connection. Ports are added in pairs, i.e., for each external master port (M), we add an internal slave port (S) for a memory bank to avoid limiting bandwidth during memory access.

Increasing the number of slave/master ports does not lead to any performance improvement in the one-at-a-time configuration, limited to 32 bit per bus cycle according to its architecture. On the contrary, the fully connected configuration maximizes bandwidth, which increases linearly with the number of bus ports, at the cost of a higher area (and power consumption). The bus in the one-at-a-time configuration occupies about 85 % less silicon space compared to the fully connected configuration, considering the same number of slave/master ports.

In overall performance, a 16×16 matrix multiplication algorithm on X-HEEP takes approximately 34 % fewer clock cycles in the highest performance configuration with the CV32E40P core and fully connected bus compared to the lowest power configuration with the CV32E40P core and one-at-a-time bus. Furthermore, when using the Xpulp extensions and fully connected bus, the CV32E40P can compute matrix multiplication algorithms 4 \times faster with 32 bit data or up to 16 \times faster with 8 bit SIMD extensions for the same CPU without extensions, as shown in [12].

3.1.4 Peripheral domain. Figure 2 (c) shows the area of the IPs located in the peripheral domain. This domain includes peripherals that can be removed from the design or powered off if not needed to trade off area or power and functionality.

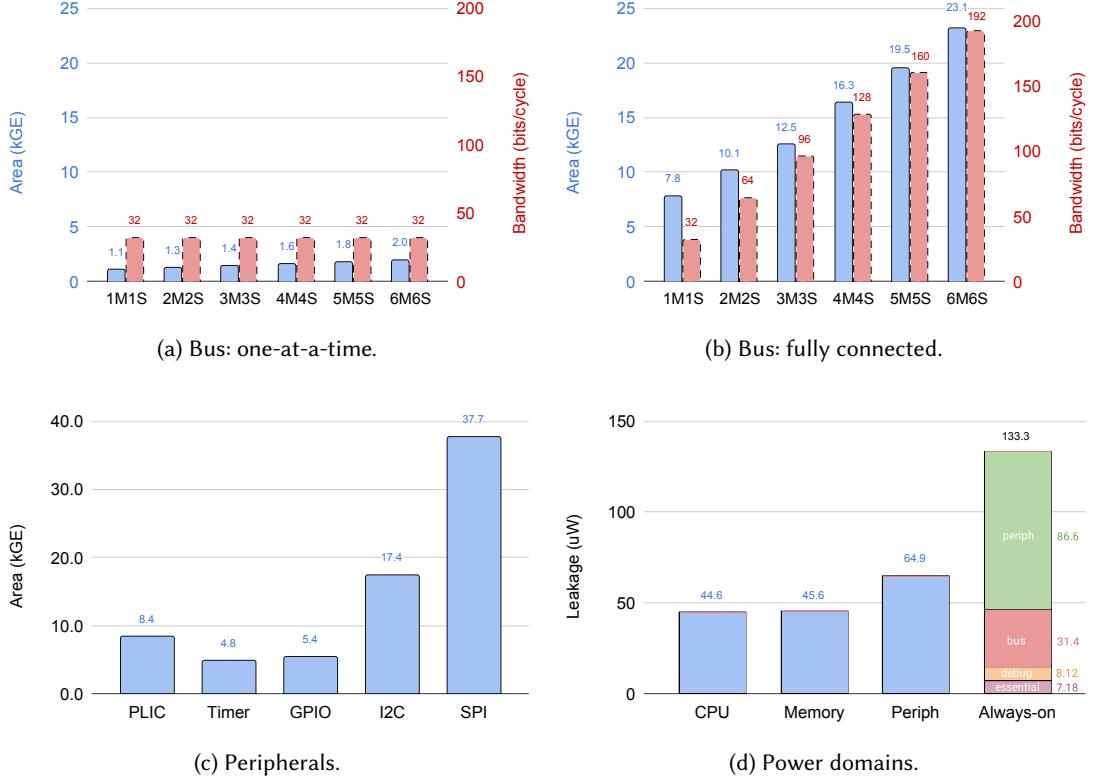


Figure 2. Exploration of different X-HEEP configurations. The used technology is TSMC 65 nm low-power CMOS at 1.2 V.

These include a platform-level interrupt controller (PLIC), a timer, and general-purpose I/O peripherals such as a GPIO, I2C, and SPI.

3.1.5 Always-on peripheral domain. This domain includes IPs that are always powered on. To meet our specific needs and requirements, we custom-designed key components such as an SoC controller, a boot ROM, a power manager, a fast interrupt controller, and a DMA. The domain also includes other peripherals such as a timer, a UART, an SPI, and a GPIO.

The power manager is responsible for implementing low-power strategies, including clock-gating, power-gating, and RAM retention. It features a set of configuration registers that provide the user with real-time control over the available low-power techniques.

The architecture is divided into several power domains, marked with different colors in Figure 1. Clock-gating can be applied to the main CPU, peripheral domain, and each memory bank, while retention can only be applied to memory banks. Additionally, each power domain can be individually power-gated. The leakage power consumption of each domain is reported in Figure 2 (d). The system bus, debug unit, and other essential IPs represent about 35 % of the

leakage power of the always-on domain. The remaining 65 % comes from other general-purpose peripherals added to enhance versatility, such as a GPIO, SPI, UART, etc.

The platform can be extended with additional power domains to include user external accelerators. This is possible thanks to external power ports, part of the XAIF interface, directly connected to the power manager, which can be used to clock-gate, power-gate, or set in retention mode external accelerators.

3.2 Extendible Accelerator InterFace (XAIF)

The extensive array of domain-specific hardware accelerators encompasses a diverse range of requirements, including memory capacity, area, performance, and power efficiency. These varied demands are aggregated into the configurable XAIF interface, facilitating enhanced connectivity to state-of-the-art accelerators, and agile integration into microcontrollers for real-life applications. Such an interface gathers all the requirements to extend X-HEEP with domain-specific customizations. To the best of the authors' knowledge, no other open-source platform for edge-computing applications exists that provides such a complete extension interface to fulfill the requirements of state-of-the-art solutions.

3.2.1 Memory mapped ports. A configurable number of slave and master ports, utilizing the OBI protocol, can be harnessed to connect custom accelerators to the X-HEEP bus. Slave ports provide easy access and configuration for memory-like accelerators, exemplified in Subsection 2.1.1, such as the Keccak [8], which requires two 32 bit slave ports for control and status operations and data memory. In addition, a further peripheral interface connected to the X-HEEP peripheral bus is provided for external custom peripherals. This peripheral interface is further extended by a FIFO interface to allow easy DMA-peripheral connections. This allows the CPU to wait for peripheral transactions to transfer all data to the main memory with the support of the system DMA, as implemented in [27]. On the other hand, master ports accommodate the bandwidth requirements of processor-like accelerators outlined in Subsection 2.1.2. For example, a CGRA [9] leverages the four 32 bit master ports that are used to independently read and write data to and from the main memory.

3.2.2 Interrupt ports. A configurable number of interrupt lines can be used by the custom hardware to rapidly synchronize with the host CPU. Each line is connected to the X-HEEP PLIC interrupt controller, which can be controlled via software. This functionality allows the host CPU to enter a sleep state during active accelerator periods, significantly reducing the overall energy consumption of the running application.

3.2.3 Power control ports. To provide low-power strategy capabilities to custom accelerators, a configurable number of power control interfaces is provided. Each interface is connected to the X-HEEP power manager to implement different power-saving strategies. Each interface includes control signals for power-gating, clock-gating, and RAM retention.

3.3 Tools and software

We present the tools and software provided by X-HEEP to configure, program, and implement user designs.

3.3.1 Configuration. X-HEEP is configured through SystemVerilog templates, which function as a dynamic tool that enables users to automatically customize the RTL code of the platform thanks to customizable parameters. This makes the generated code readable and easy to maintain and debug.

3.3.2 Software. X-HEEP includes a HAL to access peripheral functionalities and supports FreeRTOS for improved development and efficient resource management.

3.3.3 Simulation and implementation. X-HEEP offers support for simulation and implementation, in FPGA and silicon, based on the FuseSoC build system [11]. FuseSoC supports several EDA tools, such as Verilator, Questasim, Design Compiler, Genus, and Vivado, and automatically generates the scripts required to simulate or implement user designs. Thanks to it, the user can explore the design both at a high level by integrating accelerators described in SystemC, at the RTL level, and FPGA, for early prototyping and exploration, as well as in silicon, for final validation.

4 HEEPOCRATES

In this section, we present an integration example to demonstrate the real-world applicability of X-HEEP. This integration effort results in HEEPocrates, a heterogeneous architecture designed for ultra-low-power healthcare applications. These applications typically involve extended data acquisition periods during which data from external biosensors are stored in memory, followed by intensive processing periods to classify such data. Therefore, we exploited the XAIF interface to extend X-HEEP with a CGRA accelerator [9] and an IMC accelerator [31], both of which have been shown to efficiently reduce the overall energy consumption of healthcare applications [7, 24]. Moreover, each accelerator is located in a separate power domain that can be individually switched on and off for fine-grained power control.

4.1 Architecture

Figure 3 shows the HEEPocrates architecture highlighting how the CGRA and IMC accelerators are integrated to minimize power and maximize bandwidth.

4.1.1 X-HEEP configuration. We configured the X-HEEP host platform with (1) the CV32E20 core, which is optimal for running control tasks and offloading performance-intensive computations to the external accelerators while preserving low power consumption; (2) 8 SRAM banks of 32 KiB in contiguous addressing mode to accommodate variable lengths of data acquisitions while power-gating the unused banks on different applications; (3) a fully connected bus to provide high-bandwidth capabilities to the integrated accelerators; (4) all the available peripherals in place to deliver high flexibility; (5) a CGRA and IMC accelerators connected to the external XAIF interface.

4.1.2 CGRA accelerator [9]. This accelerator offers two slave ports, one to access the internal configuration registers, and one for the context memory, plus four master ports used to load and store data. The context memory stores the kernel’s code executed by the four internal processing elements (PEs). Each PE is connected to a dedicated master port to read and write data from/to the X-HEEP main memory, independently. This allows a maximum bandwidth of 128 bit per bus cycle. To synchronize the CGRA and the CPU, the CGRA end-of-computation event is connected to the X-HEEP interrupt controller (PLIC) via the XAIF interface.

The CGRA is divided into two power domains: one for the control logic and the datapaths; and one for the context memory. The control logic and datapaths can be clock-gated or power-gated, while the context memory can be clock-gated, power-gated, or set in retention mode. This dual power domain structure enables clock-gating individual domains during short periods of inactivity and power-gating during extended non-use periods. Additionally, it offers the flexibility of independently setting the context memory in retention mode while clock-gating or power-gating the datapaths and control logic to save CGRA configuration time. The XAIF interface provides control over the various power modes, enabling the system to dynamically adjust its power consumption based on the operational requirements of the CGRA accelerator.

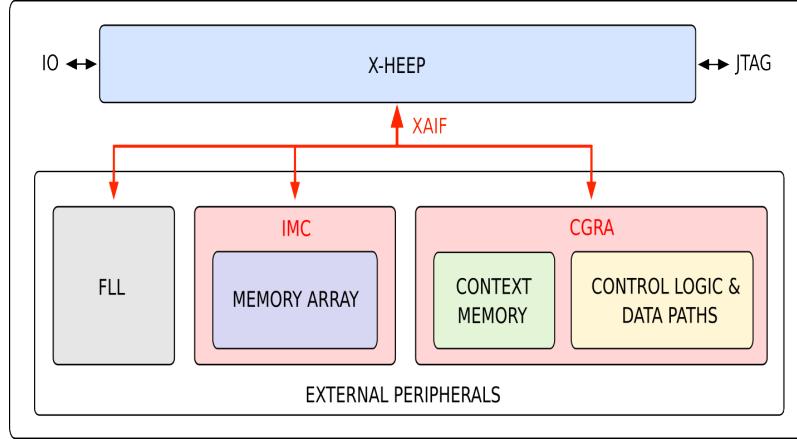


Figure 3. *HEEPocrates* architecture. Power domains are visually marked using different colors. The components highlighted in grey are always on. The accelerator integration is highlighted in red.

4.1.3 IMC accelerator [31]. This accelerator offers one slave port to access its memory array. An internal controller decodes the memory requests and facilitates the transition of the accelerator between two modes: memory mode and computation mode. In memory mode, the memory space functions as a conventional memory bank. In contrast, the computation mode enables the execution of in-memory computations, eliminating the need for additional data transfers between the main memory and the accelerator.

As for the CGRA, the IMC accelerator is placed in a separate power domain to save power when not used.

4.1.4 Frequency-locked loop [2]. We utilized the XAIF interface to connect the frequency-locked loop (FLL) responsible for generating the system clock from a 32 kHz external source. For real-time configurability, the FLL exposes a set of memory-mapped registers that enable the host CPU to adjust the system clock frequency during application execution, dynamically. This feature is precious during extended data acquisition periods in healthcare applications because it allows for reducing the system frequency to the minimum value required for acquiring the necessary biosignals, thereby minimizing dynamic power consumption. Lastly, the FLL can be also bypassed, allowing the external source to serve as the system clock.

4.2 FPGA implementation

We implemented HEEPocrates in FPGAs on the Zynq 7020, Zynq UltraScale+, and Artix 7 chips by Xilinx for early prototyping. This allows for the exploration of different X-HEEP configurations and accelerators to optimally tune the architecture for the healthcare domain.

4.3 Silicon implementation

After FPGA prototyping and exploration, we implemented HEEPocrates in silicon with TSMC 65 nm low-power CMOS technology. Figure 4 shows the 6 mm² layout of HEEPocrates, with the power domains shown in different colors.

For conducting our measurements, we developed a board specifically designed to accommodate our chip. HEEPocrates has been tested from 0.8 V to 1.2 V, achieving a maximum frequency of 170 MHz and 470 MHz, respectively. Power

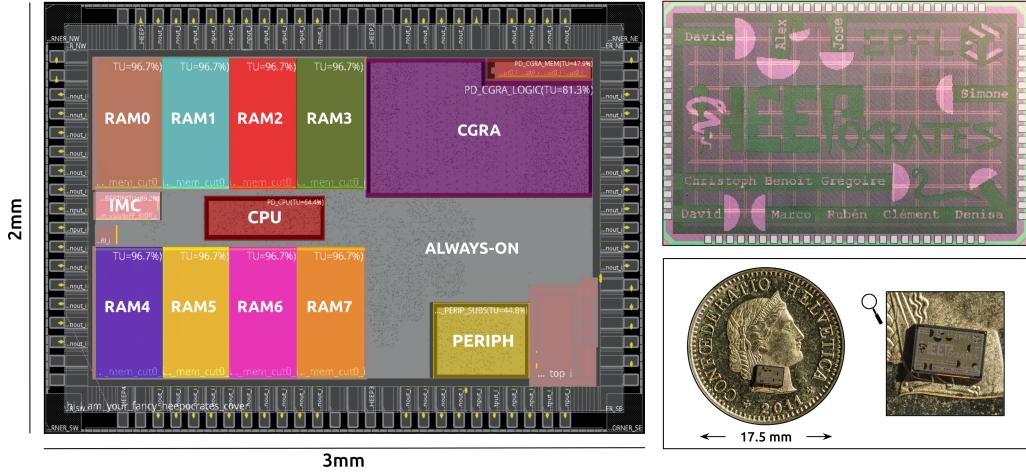


Figure 4. *HEEPocrates* layout, silicon photo, and physical chip (on a Swiss 5-cent franc coin).

consumption ranges from $270\text{ }\mu\text{W}$ at 32 kHz and 0.8 V , to 48 mW at 470 MHz and 1.2 V . Each phase of healthcare applications has been optimized to minimize power consumption.

4.3.1 Acquisition phase. Healthcare applications commonly feature an extended acquisition phase due to the low-bandwidth nature of biosignals and the typical lengthy data windows. During this phase, samples are gathered from external ADCs via SPI, or other I/O peripherals, and stored in memory by the main CPU or the DMA. We run this phase at 1 MHz , 0.8 V to minimize power while offering enough performance for the acquisition of bio-signals in the order of hundreds of Hertz. HEEPocrates consumes $384\text{ }\mu\text{W}$ during acquisition when the complete system is active, and the host CPU is clock-gated when not used. However, power can be further optimized by switching off the unused memory banks, the peripheral domain, and the external accelerators for the entire acquisition period. This enables a reduction in power of 19 %, which leads to $310\text{ }\mu\text{W}$. Furthermore, the CPU can be turned off during idle periods, i.e., when not used actively to acquire ADC samples, reaching the lowest power level of the system at 1 MHz of $286\text{ }\mu\text{W}$, with a further reduction of 8 %.

4.3.2 Processing phase. Upon completion of the acquisition phase, we run the processing phase at the maximum speed of 170 MHz , 0.8 V to minimize processing time and race to sleep. HEEPocrates consumes 8.17 mW during the processing phase when the complete system is active and the CPU executes a matrix multiplication. Power can be further optimized 6 % by turning off the unused memory banks, the peripheral domain, and the external accelerators, with a consumption of about 7.68 mW . During the processing phase, the external accelerators can be individually powered on, and computationally intensive tasks can be offloaded by the main CPU to reduce the system's overall energy consumption. HEEPocrates consumes 4.01 mW and 1.65 mW when CNN algorithms are executed on the CGRA accelerator and IMC accelerator, respectively, at their maximum frequency of 60 MHz . The host CPU, the unused memory banks, and the peripheral domain are powered off during accelerator activity.

5 EXPERIMENTAL SETUP

This section introduces a representative set of different families of microcontrollers commonly used in healthcare applications. Subsequently, it describes the biomedical applications that are included in our benchmark.

5.1 Healthcare microcontrollers

Healthcare applications exhibit significant variability in acquisition and processing times, influenced by factors such as the length of sampling windows and the complexity of adopted algorithms. To address this variability, a diverse range of microcontrollers have been designed, each optimized to minimize power consumption during specific phases. The Apollo 3 Blue excels in acquisition phases, prioritizing power efficiency through its deep sleep mode, which ensures remarkably low power consumption when the system is inactive during idle periods. On the other hand, GAP9 takes the lead in processing phases thanks to its higher-performance core, which guarantees substantial reductions in processing time. The analysis of these two microcontrollers enables covering the entire spectrum of ultra-low-power edge devices, ranging from top-tier power efficiency, with Apollo 3 Blue, to top-tier performance, with GAP9. Furthermore, the frequent use of both microcontrollers in this domain demonstrates their capability to meet the rigorous demands of healthcare applications in terms of performance, power, and area. Table 1 reports the features of the selected microcontrollers.

5.1.1 Apollo 3 Blue. This MCU is part of the Ambiq board and features an ARM Cortex-M4 core. The code is stored in the on-chip flash memory with zero overhead in instruction fetching, while the rest of the data resides either entirely in the SRAM when it fits or in both the SRAM and the flash. Unnecessary SRAM banks are turned off for the entire duration of the application. Its optimal processing configuration is 0.7 V, 48 MHz. However, we exploited the TurboSPOT mode to increase the frequency to 96 MHz when required to meet the timing constraints of the benchmark applications. Moreover, during idle periods, the system enters its deep sleep mode, consuming approximately 6 μ A/MHz, where most of the system components are power-gated, with only a few power control modules active.

5.1.2 GAP9. This MCU is part of the GAP9EVK board and features one CV32E40P core, known as the fabric controller (FC), and a cluster (CL) with nine CV32E40P cores, which can be switched on and off. We execute the benchmark applications exclusively on the FC while power-gating the CL and unnecessary SRAM banks for the entire duration of the application. The application code and data are stored in the SRAM for maximum performance. Its optimal processing configuration is 0.65 V, 240 MHz. Furthermore, during idle periods, the system transitions into its sleep mode, where the majority of components are power-gated, except for memory banks, which enter a retention mode.

5.1.3 HEEPocrates. The application code and data are completely stored in the SRAM, when possible, or in a combination of the SRAM and the off-chip flash, connected through the SPI interface. The peripheral domain and the unused memory banks are also powered off throughout the entire duration of the application. We execute all the benchmark applications on the host CPU while power-gating the external accelerators. Moreover, we also accelerate CNN computations on the CGRA and IMC accelerators and showcase the energy improvement compared to running on the host CPU. We performed each measurement under the optimal operating conditions: 170 MHz at 0.8 V, for the host CPU; 60 MHz at 0.8 V, for the CGRA and IMC accelerators. During idle periods, the host CPU and the external accelerators are power-gated, and the system frequency is lowered to 1 MHz to reduce power consumption.

Table 1. Microcontrollers commonly adopted in healthcare applications.

MCU	Board	Processing element	Voltage	Maximum frequency
Apollo 3 Blue	Ambiq	Cortex-M4	0.7 V	48 MHz
GAP9	Gapuino	CV32E40P	0.65 V	240 MHz
HEEPocrates	Testing board	CV32E20	0.8 V	170 MHz

Table 2. Healthcare applications included in our benchmark.

Application	Acquisition window	Input leads	Sampling rate	Bits per sample
Heartbeat classifier	15 s	3	256 Hz	16
Seizure detection CNN	4 s	23	256 Hz	16

5.2 Healthcare applications

Table 2 reports the healthcare applications selected for our benchmark. Our selection ensures that we cover the full spectrum of ultra-low-power healthcare applications, ranging from acquisition-dominated, with the heartbeat classifier, to processing-dominated, with the seizure detection CNN. Moreover, these applications showcase computational algorithms of varying complexity, thereby enhancing the comprehensiveness of our analysis.

5.2.1 Heartbeat classifier [4]. This application is used to detect irregular beat patterns for common heart diseases through the analysis of electrocardiogram (ECG) signals. The most resource-intensive part of this application lies in the initial computation phase, specifically the morphological filtering, which consumes over 80 % of the total processing time. Subsequently, the classification stage employs random projections. Initially, the algorithm processes a single input channel. If an abnormal heartbeat is detected, the analysis extends to the other leads for a more precise determination. Our testing scenarios involve input signals that all contain abnormal beats to evaluate the complete application pipeline. The input signal is derived from three distinct ECG leads, each sampled at 256 Hz with an accuracy of 16 bit. A 15 s acquisition window produces an input signal of 22.5 KiB.

5.2.2 Seizure detection CNN [13]. This application is used to detect seizures in electroencephalography (EEG) signals. It features a CNN with three one-dimensional convolutional layers, each incorporating pooling and ReLU layers. 90 % of the processing time is spent in convolutional computations, which mainly involve multiply and accumulate (MAC) and shift operations. Following each convolution, there is an overflow check and a maximum test for the pooling layer. Two fully connected layers end the network. The signal is sampled from 23 leads at a rate of 256 Hz with 16 bit accuracy and the acquisition phase lasts 4 s, resulting in an input signal size of 46 KiB.

6 EXPERIMENTAL RESULTS

In this section, first, we analyze the energy consumption of the proposed host platform, HEEPocrates (with the accelerators power-gated), in comparison with the selected state-of-the-art microcontrollers that may serve as host platforms. Subsequently, we assess the energy efficiency gained from leveraging the HEEPocrates' accelerators in comparison to execution on the host CPU.

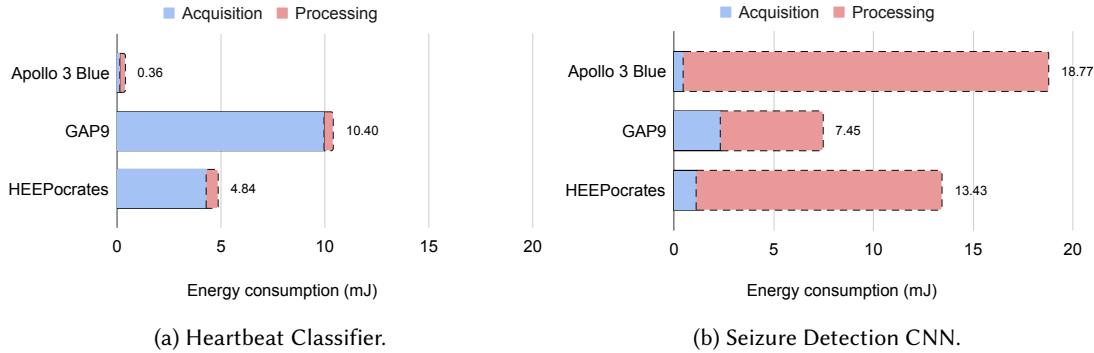


Figure 5. Energy consumption of our benchmark running on common healthcare microcontrollers and on HEEPocrates at 0.8 V.

6.1 Host platforms

Figure 5 illustrates the measured energy values for each healthcare application from our benchmark.

The heartbeat classifier application exhibits an acquisition-driven nature, characterized by extended acquisition windows and a low sampling rate of 256 Hz. This forces microcontrollers to spend a significant amount of time in idle states during acquisition. In particular, the Apollo 3 Blue stands out for its energy efficiency, attributed to its remarkably low sleep mode of only 6 μ A/MHz, where most of the system is power-gated, with only a few control modules active. On the contrary, GAP9 lacks aggressive sleep modes and keeps more modules always on, resulting in considerably higher energy consumption. Even during the processing phase, Apollo 3 Blue maintains a slight energy advantage over GAP9. This can be attributed to the optimized design of its CPU, the ARM Cortex-M4, which is more efficient for the specific operations required by this application, including logical and comparison operations, branches, as well as load and store instructions [4].

HEEPocrates positions itself in a middle ground during acquisition, offering a more robust sleep mode compared to GAP9. However, it does not reach the exceptionally low power consumption levels of Apollo 3 Blue due to the absence of aggressive sleep strategies for faster wake-up times, which includes in the always-on IPs more peripherals as an FLL, a pad controller, bus, a debug unit, and more general-purpose peripherals added for enhanced versatility (e.g. SPI, UART, etc.). However, HEEPocrates' energy efficiency can be improved by removing the general-purpose peripherals, resulting in a 27 % reduction in overall energy consumption. During processing, HEEPocrates consumes slightly higher energy compared to the other microcontrollers, due to its ultra-low-power CV32E20 core [29] that is not optimized for performance like GAP9, and due to the higher-power consumption of the active part of the chip compared to Apollo-3, sitting HEEPocrates in the middle between the two.

The seizure detection CNN application is processing-dominated due to its computationally intense convolutional network, leading microcontrollers to spend the majority of their time in the processing phase. GAP9 emerges as the dominant contender in this phase, leveraging its high-performance core to achieve reduced processing times and efficient transitions to sleep. In contrast, the core of Apollo 3 Blue lacks sufficient computational power, resulting in an extended processing phase and increased energy consumption. However, during the acquisition phase, Apollo 3 Blue maintains dominance over GAP9 due to its more efficient sleep mode, resulting in lower energy consumption.

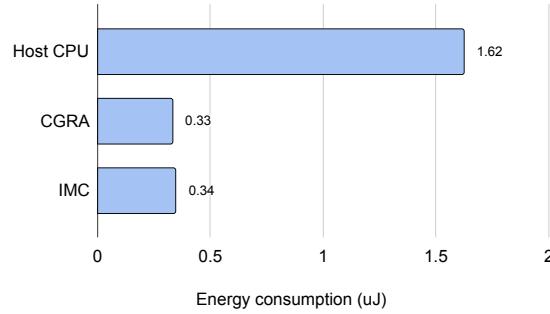


Figure 6. Energy consumption of HEEPocrates at 0.8 V running a 16x16 convolution (3x3 filter) on the host CPU (at 170 MHz) and the CGRA and IMC accelerators (at 60 MHz).

HEEPocrates finds itself positioned between Apollo 3 Blue and GAP9 in both the processing and acquisition phases. During acquisition, it offers a more efficient low-power mode than GAP9 but does not reach the efficiency levels of Apollo 3 Blue, for the reasons explained earlier. In the processing phase, the higher performance of HEEPocrates allows for faster entry into the sleep state than Apollo 3 Blue but lags behind GAP9 due to its higher-frequency core. Notably, similar to the previous application, HEEPocrates’ energy efficiency may be enhanced by removing general-purpose peripherals from the always-on domain, resulting in an overall energy reduction of about 3 %.

In conclusion, our analysis reveals the energy consumption alignment of HEEPocrates with state-of-the-art microcontrollers commonly adopted in healthcare applications. The performance and power efficiency of our platform falls between the top-tier power efficiency of Apollo 3 Blue and the top-tier performance of GAP9. This underscores that HEEPocrates achieves state-of-the-art energy efficiency figures across a wide range of real-world application profiles typical of the healthcare domain, ranging from acquisition-dominated to processing-dominated scenarios.

6.2 Accelerators

In Figure 6, we compare the energy consumption of HEEPocrates while running a 16×16 convolutional layer with a 3×3 filter on the host CPU, the CGRA and IMC accelerators. Our results demonstrate an improvement in energy efficiency of approximately 4.9 × and 4.8 × achieved by exploiting the integrated CGRA accelerator and the IMC accelerator, respectively, compared to running on the host CPU. This improvement is attributed to the higher parallelism of the proposed accelerators, which compensates for the increased power consumption resulting from the more intense computation.

7 CONCLUSIONS

In this paper, we have explored the growth and increasing demand for efficient processing solutions in the field of edge computing, particularly in the context of new AI/ML applications. Persistent challenges arise from the limitations in performance and power consumption of edge devices, which impact overall energy efficiency.

To address these challenges, heterogeneous architectures have emerged, presenting a promising solution by combining ultra-low-power host processors with specialized accelerators tailored to specific applications or domains.

However, we have shown the limitations of existing host platforms in exploring the design space of accelerator-based ultra-low power edge AI platforms, as well as in providing the configurability and extendability options needed

to integrate the large variety of custom accelerators and interfaces that exist nowadays. Consequently, extensive modifications to the RTL code are often required to integrate accelerators effectively, leading to high maintenance costs.

To overcome these limitations, we introduced X-HEEP, an open-source solution designed specifically to support the integration and exploration of ultra-low-power edge AI/ML accelerators. The platform offers comprehensive customizability and extendability options via the proposed XAIF, which gathers all the requirements of state-of-the-art domain-specific solutions, as memory-mapped accelerators, including memory, processors, and peripherals with DMA-support, custom ISA co-processor, interrupts, and power saving strategies interface, enabling designers to tailor the platform to meet the unique requirements of the target applications in performance, power, and area.

X-HEEP provides configuration options to match specific application requirements by exploring various core types, bus topologies, and memory addressing modes. It also enables a fine-grained configuration of memory banks to match the constraints of the integrated accelerators. The platform prioritizes energy efficiency by implementing low-power strategies and integrating them with accelerators through dedicated power control interfaces. This cohesive integration ensures that all system components work together to maximize energy savings.

To illustrate the practical benefits of X-HEEP, in this work, we presented a real-world integration example tailored for healthcare applications, which shows high variability among acquisition and processing-dominated application profiles. This example featured a CGRA accelerator and an IMC accelerator, both of which have proved to effectively reduce the overall energy consumption for this application domain. The resulting design, called HEEPocrates, has been implemented both in FPGAs on the Zynq 7020, Zynq UltraScale+, and Artix 7 chips by Xilinx, for early prototyping and exploration, and in silicon with TSMC 65 nm low-power CMOS technology, for silicon validation. The fabricated chip can operate from 0.8 V to 1.2 V, achieving a maximum frequency of 170 MHz and 470 MHz, respectively. Its power consumption ranges from 270 μ W at 32 kHz and 0.8 V, to 48 mW at 470 MHz and 1.2 V.

To measure the performance and versatility of the proposed design, we analyze the execution of an illustrative real-life set of edge AI/ML benchmarks that combines ultra-low power healthcare applications from the latest advances in the field, showing high variability in the execution profile. Through the execution of our benchmark and the measurement of the energy consumption of the chip, we demonstrated HEEPocrates' alignment with other state-of-the-art microcontrollers that are frequently employed in healthcare applications. This is achieved thanks to a balanced trade-off between fine-grain power domains, to reduce power consumption during acquisition phases, and on-demand accelerator capabilities, to speed up the execution of processing phases, resulting in a good trade-off between acquisition-dominated and processing-dominated applications. These results also showcase the representativeness of the experiments that other researchers could perform after integrating their accelerators with X-HEEP. Lastly, we proved the energy benefit of 4.9 \times and 4.8 \times gained by exploiting the integrated CGRA accelerator and IMC accelerator, respectively, compared to running on the host CPU.

In conclusion, the introduction of the X-HEEP platform leads to a significant step forward in overcoming the challenges faced in the field of edge computing. By providing extensive options for customizability and extendability, prioritizing energy efficiency, and presenting a practical real-world integration example, X-HEEP presents itself as an innovative platform, empowering designers and researchers to create efficient heterogeneous edge AI/ML computing systems.

8 ACKNOWLEDGEMENTS

We would like to thank the entire X-HEEP team for their great contribution to the platform.

REFERENCES

- [1] Alon Amid et al. “Chipyard: Integrated Design, Simulation, and Implementation Framework for Custom SoCs”. In: *IEEE Micro* 40.4 (2020), pp. 10–21. doi: [10.1109/MM.2020.2996616](https://doi.org/10.1109/MM.2020.2996616).
- [2] David E Bellasi and Luca Benini. “Smart energy-efficient clock synthesizer for duty-cycled sensor socs in 65 nm/28nm cmos”. In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 64.9 (2017), pp. 2322–2333.
- [3] Andrea Bocco, Yves Durand, and Florent De Dinechin. “SMURF: Scalar Multiple-Precision Unum Risc-V Floating-Point Accelerator for Scientific Computing”. In: *Proc. of the ACM Conference for Next Generation Arithmetic. CoNGA’19*. 2019. isbn: 9781450371391. doi: [10.1145/3316279.3316280](https://doi.org/10.1145/3316279.3316280).
- [4] Rubén Braojos, Giovanni Ansaloni, and David Atienza. “A Methodology for Embedded Classification of Heartbeats Using Random Projections”. In: *DATE*. IEEE, May 2013, pp. 899–904. isbn: 9781467350716. doi: [10.7873/DATE.2013.189](https://doi.org/10.7873/DATE.2013.189).
- [5] Francesco Conti et al. “A 12.4TOPS/W @ 136GOPS AI-IoT System-on-Chip with 16 RISC-V, 2-to-8b Precision-Scalable DNN Acceleration and 30%-Boost Adaptive Body Biasing”. In: *IEEE ISSCC. 2023*, pp. 21–23. doi: [10.1109/ISSCC42615.2023.10067643](https://doi.org/10.1109/ISSCC42615.2023.10067643).
- [6] CORE-V X-Interface. URL: <https://github.com/openhwgroup/core-v-xif>.
- [7] Elisabetta De Giovanni et al. “Modular Design and Optimization of Biomedical Applications for Ultralow Power Heterogeneous Platforms”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 39.11 (2020), pp. 3821–3832. doi: [10.1109/TCAD.2020.3012652](https://doi.org/10.1109/TCAD.2020.3012652).
- [8] Alessandra Dolmeta et al. “Implementation and Integration of Keccak Accelerator on RISC-V for CRYSTALS-Kyber”. In: *Proc. of the 20th ACM Int. Conf. on Computing Frontiers. CF ’23*. Bologna, Italy, 2023, pp. 381–382. doi: [10.1145/3587135.3591432](https://doi.org/10.1145/3587135.3591432).
- [9] Loris Duch et al. “A multi-core reconfigurable architecture for ultra-low power bio-signal analysis”. In: *IEEE BioCAS*. 2016, pp. 416–419. doi: [10.1109/BioCAS.2016.7833820](https://doi.org/10.1109/BioCAS.2016.7833820).
- [10] Tim Fritzmann, Georg Sigl, and Johanna Sepúlveda. “RISQ-V: Tightly Coupled RISC-V Accelerators for Post-Quantum Cryptography”. In: *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2020.4 (Aug. 2020), pp. 239–280. doi: [10.13154/tches.v2020.i4.239-280](https://doi.org/10.13154/tches.v2020.i4.239-280).
- [11] FuseSoC. URL: <https://github.com/olofk/fusesoc>.
- [12] Michael Gautschi et al. “Near-threshold RISC-V core with DSP extensions for scalable IoT endpoint devices”. In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 25.10 (2017), pp. 2700–2713.
- [13] Catalina Gómez et al. “Automatic seizure detection based on imaged-EEG signals through fully convolutional networks”. In: *Scientific reports* 10.1 (2020), pp. 1–13.
- [14] Florent Kermarrec et al. *LiteX: an open-source SoC builder and library based on Migen Python DSL*. 2020. arXiv: [2005.02506 \[cs.AR\]](https://arxiv.org/abs/2005.02506).
- [15] Maha Kooli et al. “Towards a Truly Integrated Vector Processing Unit for Memory-Bound Applications Based on a Cost-Competitive Computational SRAM Design Solution”. In: *J. Emerg. Technol. Comput. Syst.* 18.2 (2022). issn: 1550-4832. doi: [10.1145/3485823](https://doi.org/10.1145/3485823).
- [16] Kai Li, Wei Yin, and Qiang Liu. “A Portable DSP Coprocessor Design Using RISC-V Packed-SIMD Instructions”. In: *IEEE ISCAS. 2023*, pp. 1–5. doi: [10.1109/ISCAS46773.2023.10181681](https://doi.org/10.1109/ISCAS46773.2023.10181681).
- [17] LowRISC. OpenTitan. URL: <https://github.com/lowRISC/opentitan>.

- [18] David Mallasén, Alberto A. del Barrio, and Manuel Prieto-Matias. *Big-PERCIVAL: Exploring the Native Use of 64-Bit Posit Arithmetic in Scientific Computing*. 2023. arXiv: [2305.06946](https://arxiv.org/abs/2305.06946).
- [19] Paolo Mantovani et al. “Agile SoC Development with Open ESP : Invited Paper”. In: *2020 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*. 2020, pp. 1–9.
- [20] Katayoun Neshatpour et al. “Big biomedical image processing hardware acceleration: A case study for K-means and image filtering”. In: *IEEE ISCAS*. 2016, pp. 1134–1137. doi: [10.1109/ISCAS.2016.7527445](https://doi.org/10.1109/ISCAS.2016.7527445).
- [21] *Open Bus Interface Protocol*. URL: <https://github.com/openhwgroup/obi>.
- [22] Alessandro Ottaviano et al. “Cheshire: A Lightweight, Linux-Capable RISC-V Host Platform for Domain-Specific Accelerator Plug-In”. In: *IEEE Transactions on Circuits and Systems II: Express Briefs* (2023), pp. 1–1. doi: [10.1109/TCSII.2023.3289186](https://doi.org/10.1109/TCSII.2023.3289186).
- [23] Daniel Petrisko et al. “BlackParrot: An Agile Open-Source RISC-V Multicore for Accelerator SoCs”. In: *IEEE Micro* 40.4 (2020), pp. 93–102. doi: [10.1109/MM.2020.2996145](https://doi.org/10.1109/MM.2020.2996145).
- [24] Flavio Ponzina et al. “A Hardware/Software Co-Design Vision for Deep Learning at the Edge”. In: *IEEE Micro* 42.6 (July 2022), pp. 48–54. doi: [10.1109/MM.2022.3195617](https://doi.org/10.1109/MM.2022.3195617).
- [25] Antonio Pullini et al. “Mr. Wolf: An energy-precision scalable parallel ultra low power SoC for IoT edge processing”. In: *IEEE Journal of Solid-State Circuits* 54.7 (2019), pp. 1970–1981.
- [26] *Rocket*. URL: <https://github.com/chipsalliance/rocket-chip>.
- [27] Pasquale Davide Schiavone et al. “Arnold: An eFPGA-augmented RISC-V SoC for flexible and low-power IoT end nodes”. In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 29.4 (2021), pp. 677–690.
- [28] Pasquale Davide Schiavone et al. “Quentin: an Ultra-Low-Power PULPissimo SoC in 22nm FDX”. In: (2018), pp. 1–3. doi: [10.1109/S3S.2018.8640145](https://doi.org/10.1109/S3S.2018.8640145).
- [29] Pasquale Davide Schiavone et al. “Slow and steady wins the race? A comparison of ultra-low-power RISC-V cores for Internet-of-Things applications”. In: *Int. Symp. on Power and Timing Modeling, Optimization and Simulation (PATMOS)*. IEEE. 2017, pp. 1–8.
- [30] Pasquale Davide Schiavone et al. “X-HEEP: An Open-Source, Configurable and Extendible RISC-V Microcontroller”. In: *Proc. of Int. Conf. on Computing Frontiers*. CF ’23. New York, NY, USA: ACM, 2023, pp. 379–380. ISBN: 9798400701405. doi: [10.1145/3587135.3591431](https://doi.org/10.1145/3587135.3591431).
- [31] William Andrew Simon et al. “BLADE: An in-cache computing architecture for edge devices”. In: *IEEE Transactions on Computers* 69.9 (2020), pp. 1349–1363.
- [32] Mattia Sinigaglia et al. *Echoes: a 200 GOPS/W Frequency Domain SoC with FFT Processor and I2S DSP for Flexible Data Acquisition from Microphone Arrays*. 2023. arXiv: [2305.07325](https://arxiv.org/abs/2305.07325).
- [33] Blaise Tine et al. “Vortex: Extending the RISC-V ISA for GPGPU and 3D-Graphics”. In: *IEEE/ACM Int. Symp. on Microarchitecture (MICRO)*. 2021, pp. 754–766. ISBN: 9781450385572. doi: [10.1145/3466752.3480128](https://doi.org/10.1145/3466752.3480128).
- [34] Y. Varma and M.P. Tull. “Architectural design of a complex arithmetic signal processor (CASP)”. In: *Region 5 Conference: Annual Technical and Leadership Workshop*. 2004, pp. 69–76. doi: [10.1109/REG5.2004.1300163](https://doi.org/10.1109/REG5.2004.1300163).
- [35] Florian Zaruba and Luca Benini. “The cost of application-class processing: Energy and performance analysis of a Linux-ready 1.7-GHz 64-bit RISC-V core in 22-nm FDSOI technology”. In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 27.11 (2019), pp. 2629–2640.