Demonstrating Analog Inference on the BrainScaleS-2 Mobile System

Yannik Stradmann, Sebastian Billaudelle, Oliver Breitwieser, Falk Leonard Ebert, Arne Emmel, Dan Husmann, Joscha Ilmberger, Eric Müller, Philipp Spilger, Johannes Weis, Johannes Schemmel*, Member, IEEE*

All authors are with the Kirchhoff-Institute for Physics, Heidelberg, Germany (e-mail: yannik.stradmann@kip.uni-heidelberg.de).

All authors contributed equally, a detailed description of individual contributions is given at the end of the manuscript. The development of the presented system has been funded by the German Federal Ministry of Education and Research (BMBF) under grant number 16ES1127 as part of the *Pilotinnovationswettbewerb „Energieeffizientes KI-System"*. It is built upon work funded by the EU (H2020/2014-2020: 720270, 785907, 945539 (HBP)), the Lautenschläger-Forschungspreis 2018 for Karlheinz Meier and the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy EXC 2181/1-390900948 (Heidelberg STRUCTURES Excellence Cluster).

ABSTRACT We present the BrainScaleS-2 mobile system as a compact analog inference engine based on the BrainScaleS-2 ASIC and demonstrate its capabilities at classifying a medical electrocardiogram dataset. The analog network core of the ASIC is utilized to perform the multiply-accumulate operations of a convolutional deep neural network. At a system power consumption of 5.6 W, we measure a total energy consumption of 192 µJ for the ASIC and achieve a classification time of 276 µs per electrocardiographic patient sample. Patients with atrial fibrillation are correctly identified with a detection rate of $(93.7 \pm 0.7)\,\%$ at $(14.0 \pm 1.0)\,\%$ false positives. The system is directly applicable to edge inference applications due to its small size, power envelope, and flexible I/O capabilities. It has enabled the BrainScaleS-2 ASIC to be operated reliably outside a specialized lab setting. In future applications, the system allows for a combination of conventional machine learning layers with online learning in spiking neural networks on a single neuromorphic platform.

INDEX TERMS accelerator, analog computing, convolutional deep neural networks, electrocardiography, inference, low-power, medical, neuromorphic

I. Introduction

ARTIFICIAL neural networks have become an important tool for a broad variety of tasks – from datacenter to edge applications. Striving for energy-efficient and fast computation of these networks, a multitude of novel computing architectures have been developed. Specialized processors either accelerate the processing of artificial convolutional deep neural networks (CDNNs) or – in the field of event-based neuromorphic computing – follow a neuroscience-oriented approach and implement spiking neural networks (SNNs).

Accelerators for vector-matrix multiplication (VMM)-based CDNN models mostly rely on computational units in the digital domain [1, 2, 3, 4], although recent analog approaches show very promising performance [5, 6]. In agreement with their biological example, event-based neuromorphic systems traditionally utilize analog computational paradigms [7, 8, 9], the general availability of modern CMOS process nodes has however boosted the popularity of digital solutions in this field as well [10, 11, 12, 13, 14, 15, 16, 17]. Most recently, research of VMM, as well as SNN accelerators has been augmented by the introduction of post-CMOS technologies based on novel materials [18, 19, 20].

In contrast to aforementioned single-purpose approaches, the BrainScaleS neuromorphic architecture combines analog VMM with the event-based emulation of SNNs. BrainScaleS-2 (BSS-2) therefore provides a highly configurable computational substrate for research in the combined fields of computer- and neuroscience [21, 22] and has been shown to achieve beyond-state-of-the-art energy efficiency and classification latency [23]. Combining potential energy efficiency benefits and online learning capabilities of SNNs with the high computational power of CDNNs on a single application-specific integrated circuit (ASIC) opens up unique opportunities for adaptive inference applications on the edge. The only other neuromorphic architectures simultaneously supporting rate- and spike-based models are the digital Tianjic [24] and MONETA [25] systems, both however do not enable freely programmable on-chip learning rules.

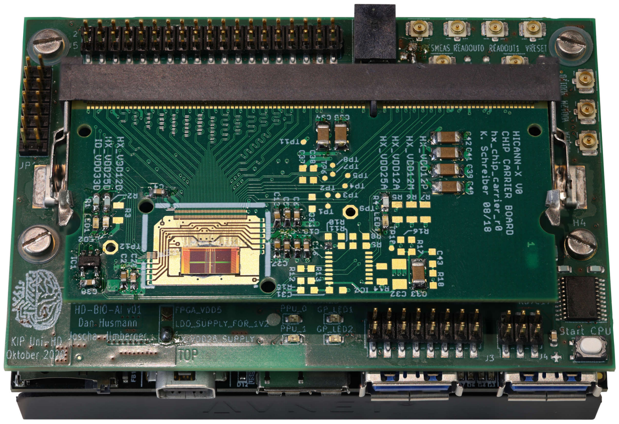

Figure 1. Photo of the BrainScaleS-2 mobile system (from bottom to top): FPGA-based system controller, ASIC adapter PCB, ASIC carrier board with the latest BSS-2 ASIC directly wire-bonded to the PCB. The system has the mechanical footprint of a credit card (84 mm × 55 mm) at a height of approximately 40 mm. It weighs roughly 155 g with and 70 g without the FPGA's heatsink respectively.

We now present a highly integrated mobile demonstrator system for the BSS-2 architecture (Figs. 1 and 2) and showcase the system's capabilities and energy efficiency at the example of electrocardiogram (ECG) anomaly classification. While both, the computation of CDNNs and the emulation of SNNs on BSS-2 have already been shown in controlled lab environments [23, 26], we can now provide a system that is physically small, has a low power envelope and flexible I/O capabilities. These previous experiments designed for BSS-2 are compatible with the presented mobile platform, the herein presented ECG classifier extends the set of applications by a task tailored to edge scenarios.

The design constraints for this system as well as the chosen classification task were motivated by the participation in the independently judged *Pilotinnovationswettbewerb „Energieeffizientes KI-System“* by the German Federal Ministry of Education and Research (BMBF), where it has proven to operate reliably outside controlled lab environments. This competition posed a challenge to classify atrial fibrillation (A-fib) in batches of medical ECG recordings with standalone edge computing accelerators. The provided dataset consists of 16 000 traces from the same patient group and has been recorded with two channels only, mimicking the signal quality to be expected from consumer-grade medical wearables.[1] The classification of anomalies in ECG time series data is an active field of research where both, classical time series analysis and machine learning-based algorithms compete [27].

II. The BrainScaleS-2 Mobile System

The BSS-2 mobile system features a combination of a commercially available FPGA module and the most recent BrainScaleS-2 ASIC. The FPGA contains an embedded CPU which is used for standalone experiment control and I/O. The logic fabric in the FPGA acts as a memory interface and data format converter for the ASIC. Fig. 2 depicts the three main components of the system:

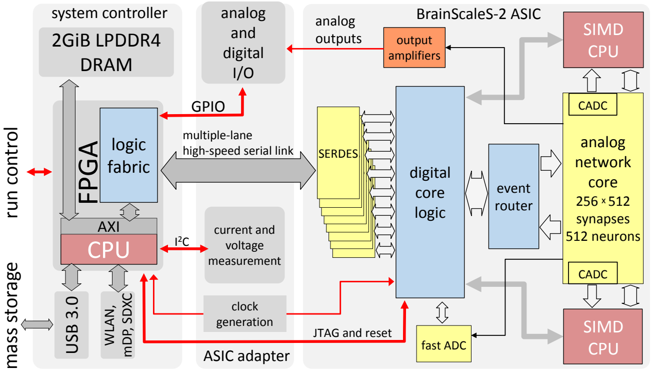

Figure 2. Overview of the BrainScaleS-2 mobile system (from left to right): FPGA-based controller, ASIC adapter PCB and the BSS-2 ASIC.

- the BrainScaleS-2 ASIC directly bonded to a carrier board (right),
- a custom ASIC adapter PCB, interfacing the FPGA board to this ASIC carrier board (center),
- the system controller, consisting of a low-power FPGA with an embedded quad-core microprocessor [28] and 2 GiB of LPDDR4 DRAM, USB 3.0 (device & host), SDXC, 802.11b/g/n Wi-Fi as well as Bluetooth 4.2 (BLE) communication circuits (left).

The described system is the result of a tightly coupled interdisciplinary work ranging from chip design to software engineering and machine learning. The following sections describe different aspects of the BSS-2 mobile system from the perspective of the different technological areas.

A. Neuromorphic ASIC

The BSS-2 neuromorphic ASIC[2] [21] is the key component of the presented system. It is a mixed-signal implementation comprised of analog and digital building blocks (Fig. 2) that simultaneously supports the processing of VMM operations and the emulation of SNNs in the analog domain. Embedded single instruction, multiple data central processing units (SIMD CPUs) allow for online on-chip learning.

Analog Network Core

BSS-2 contains a total of 512 analog neuron circuits, each receiving input from 256 synapses. The neurons emulate the Adaptive Exponential Integrate-and-Fire (AdEx) model in 1000-fold accelerated continuous time and can be combined

[1] Since the dataset contains sensitive patient information it is not publicly available.

[2] The ASIC has been manufactured in a standard 65 nm CMOS technology. It was conceived and designed at Heidelberg University. The link layer of the high-speed serial links has been developed in collaboration with the TU Dresden, who also contributed the PLL. The fast ADC is a result of a collaboration with the EPFL Lausanne.

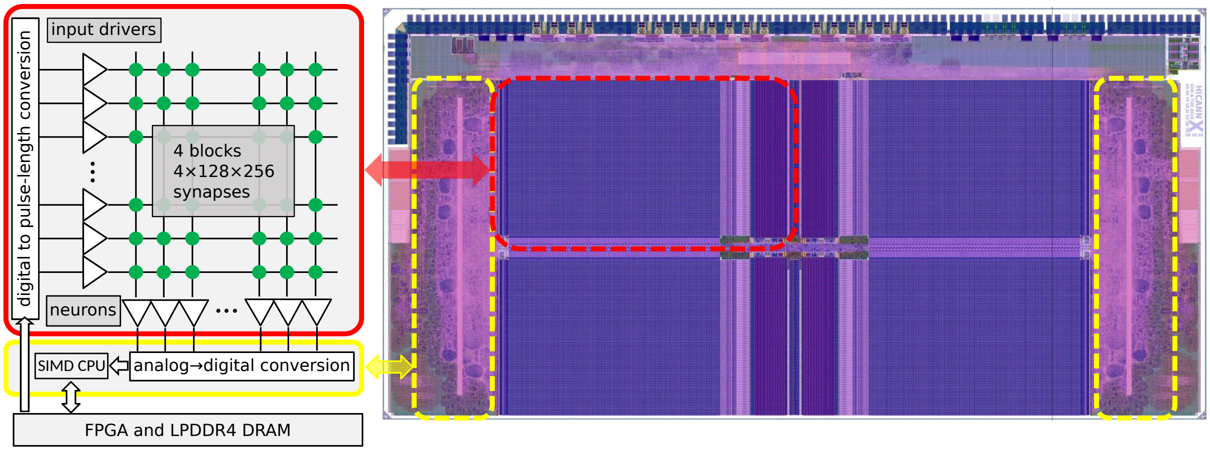

Figure 3. ***Left:*** **internal structure of the BSS-2 ASIC. The analog network core consists of four quadrants, each containing 128 neurons and 128×256 synapses (red). A total of 1024 parallel ADC channels allow for readout of various analog parameters by two embedded SIMD processors (yellow).** ***Right:*** **position of the described functional units on a layout drawing of the BSS-2 ASIC.**

to represent structured neurons with multiple compartments. Each synapse contains correlation sensors enabling spike-timing dependent plasticity (STDP) in SNNs and is modulated by a digital weight with 6 bit resolution. For VMMs, the neuron circuits are configured as analog accumulators, while the synapses perform multiplications. When processing CDNNs and SNNs, the combination of these neurons and the synapse matrix therefore perform all computations in the analog domain.

Event Router

The distribution of the real-time vector inputs or spike events to and from the analog network core is handled by a runtime configurable digital routing crossbar.

Top and Bottom SIMD CPUs

Each chip includes two custom 32 bit CPUs compatible with the embedded PowerPC instruction set architecture (ISA) [29]. They additionally feature SIMD extensions for fast vector operations, which can make use of parallel ADCs (1024 channels, 8 bit resolution) to process analog observables. These embedded cores are primarily intended to support learning and plasticity algorithms in SNNs. They can access most of the internal digital resources of the ASIC and – as described in Section III – serve as experiment controllers.

Digital Core Logic

The core control and network logic handles all off-chip communication from the embedded processors and the event router. In addition, it bidirectionally converts between real-time and time-stamped event packets. The transport layer manages secured memory access operations as well as unsecure, low-latency event streams over high-speed serial links to the FPGA fabric.

The right side of Fig. 3 shows a layout drawing of the ASIC. The embedded processors are highlighted by the yellow rectangles. The red frame depicts one of the four identical quadrants of the analog core. The left side of the figure illustrates the neuromorphic processing loop through the system, together with the arrangement of neurons and synapses within a quadrant.

In CDNN experiments, as used for the ECG classification showcased in Section III, the dataflow is as follows: Initially, the synapse matrix is filled with weight data and the neuron circuits are configured as linear integrators without any long-term internal dynamics. All neurons are reset to an initial membrane value V_{reset} before the arrival of the first component of the input vector. Inference calculation starts when the digital core logic transmits the events it has received from the FPGA to the real-time event router. They are then distributed to synapse drivers, which in turn transmit them into the synapse array.

Fig. 4 illustrates the principle of analog computation used for the VMM: To perform the analog multiplication, the events are converted from 5 bit binary coding to a pulse length representation. Each synapse produces a current proportional to its 6 bit stored weights ω_x for the duration of the input signal they receive from the synapse drivers Δt, thereby performing an analog multiplication. The input line of the neuron subsequently receives the sum of all output currents generated by the synapses within a vertical column. A transconductance amplifier in each neuron generates a current equivalent to the charge received from the synapses. Each column's current is integrated on the membrane capacitance of its associated neuron circuit. Each neuron has two separate inputs for excitatory (A) and inhibitory (B) synaptic inputs. For the inference calculation, they are used to represent positive and negative weight values. For reasons of printing space, the column is shown horizontally in the figure. For up to 65 536 signed matrix elements, this operation is carried out in parallel within the analog core.

After an input vector has been processed in the analog domain, the neuron voltages are digitized by the parallel ADC with 8 bit resolution. The rectified linear unit (ReLU) operation can be performed automatically during this conversion by aligning the ADC offset with the initial membrane

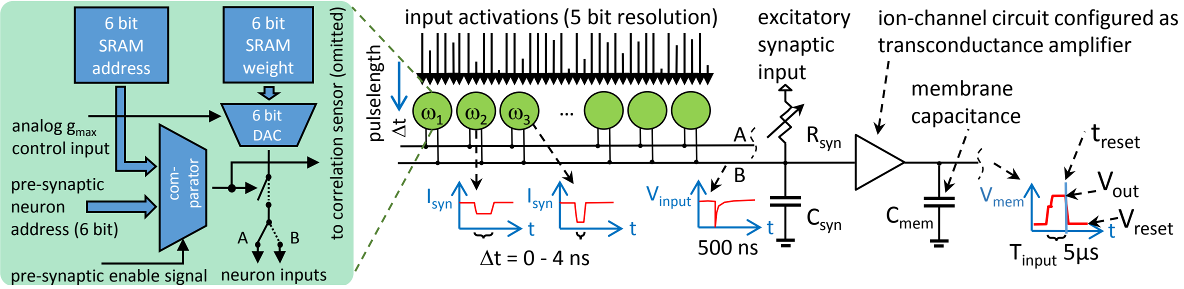
6 bit SRAM address
6 bit SRAM weight
6 bit DAC
analog g$_{max}$ control input
pre-synaptic neuron address (6 bit)
com-parator
pre-synaptic enable signal
neuron inputs
A
B
to correlation sensor (omitted)
pulselength Δt
input activations (5 bit resolution)
ω$_1$
ω$_2$
ω$_3$
...
I$_{syn}$
t
Δt = 0 - 4 ns
V$_{input}$
500 ns
excitatory synaptic input
R$_{syn}$
C$_{syn}$
ion-channel circuit configured as transconductance amplifier
membrane capacitance
C$_{mem}$
V$_{mem}$
t$_{reset}$
V$_{out}$
V$_{reset}$
T$_{input}$
5µs

Figure 4. **Operation principle of CDNN processing: the bottom half depicts the main functional blocks of a synapse circuit. For the VMM calculation only the shaded area is used. The top half shows the analog operations taking place: each synapse generates a current pulse I_{syn} in response to a pre-synaptic input event. During the calculation period T_{input} they are integrated on the membrane capacitance. The final voltage V_{out} of a single neuron represents the result of the analog VMM calculation.**

value V_{reset}. Alternatively, the embedded SIMD CPU can apply an activation function to the digitized analog result, representing the output activations of a network layer. Values that are re-used in a succeeding operation, are then converted to 5 bit input activations by subtracting V_{reset} and applying bitwise right-shifts. The results are passed to the FPGA fabric (Section II-C) and either stored in DRAM or used as inputs for the next layer. This loop is repeatedly executed until all layers have been processed.

Each synapse can process back-to-back activations with a period of 8 ns, resulting in a maximum continuous input data rate of 125 MHz (Fig. 4). There are 256×512 synapses in total, which can all simultaneously process input activations at the full data rate. This equals a maximum of

$$125\,\text{MHz} \cdot 256 \cdot 512 \cdot 2\,\text{Op} = 32.8\,\text{TOp/s}, \tag{1}$$

counting multiplication and addition as individual operations.

The full integration cycle, including the necessary time to reset the neuron membrane voltages, takes about 5 µs. This reduces the back-to-back, maximum size VMM rate to 200 kHz and the resulting speed to approximately

$$\frac{1}{5\,\mu\text{s}} \cdot 256 \cdot 512 \cdot 2\,\text{Op} \approx 52\,\text{GOp/s}. \tag{2}$$

For more details on the BrainScaleS-2 architecture, we refer to Pehle et al. [21]; for the rate-based operation mode see Weis et al. [26].

B. ASIC Adapter Board

The ASIC adapter PCB is required to interface an off-the-shelve FPGA board with the BSS-2 ASIC. It provides six power supply rails, three reference voltages, and a reference current to the ASIC, all of which are runtime-adjustable. The individual supply currents of the BrainScaleS ASIC can be monitored by several shunt-based power monitoring integrated circuits (ICs) [30]. The ASIC provides eight independent bidirectional source-synchronous low-voltage differential signaling (LVDS) data channels operated at up to 2 Gbit/s each. Due to I/O limitations of the FPGA board, only five are routed through the ASIC adapter PCB to the FPGA. Micro-SMT coaxial connectors are available for monitoring the analog outputs from the BSS-2 ASIC as well as supplies and reference voltages.

The ASIC itself is directly bonded to a carrier PCB using a zero-insertion force small outline dual in-line memory module (SO-DIMM) board edge connector for an optimal combination of simplicity and reliability. Fig. 1 shows the die bonded to the ASIC carrier PCB.

C. System Controller

The system controller is a low-power FPGA with an embedded quad-core microprocessor [28] coupled with 2 GiB of LPDDR4 DRAM. It features USB 3.0 (device & host), SDXC, 802.11b/g/n Wi-Fi as well as Bluetooth 4.2 (BLE) communication circuits. Further information about the FPGA base board can be found in [31].

Fig. 5 depicts the internal structure of the logic fabric. Main components are the link control and physical layer that implement the high-speed serial links to the ASIC. The playback buffer contains a list of commands to send to the ASIC, while the trace buffer collects events sent back from the ASIC. Memory-mapped write and read commands can also be issued from the ASIC to the FPGA. This allows the SIMD CPUs to access the DRAM memory connected to the FPGA via a memory switch.

A DMA controller reads the input data from memory, converts it into input events, and sends them to the ASIC. For the experiment described in Section III, this DMA controller is programmed by the SIMD CPU on the ASIC to transfer the raw signal data, an ECG trace composed of 12 bit values, from memory. The ASIC requires specially formatted event data packets encoding 5 bit input activations for the vector-matrix multiplication. This demands a preprocessing chain inside the FPGA, which is problem-specific to some extent. Its function will be explained in Section III-A. After the raw signal data is converted into 5 bit values, the vector event generator attaches an event address from a lookup table. This event is sent to the ASIC via the serial links. In the ASIC, the attached addresses are used to forward the events to their target inputs of the analog neuromorphic core. The use of a lookup table inside the FPGA allows arbitrary mapping of input vector elements onto the synapse matrix. During the inference process the

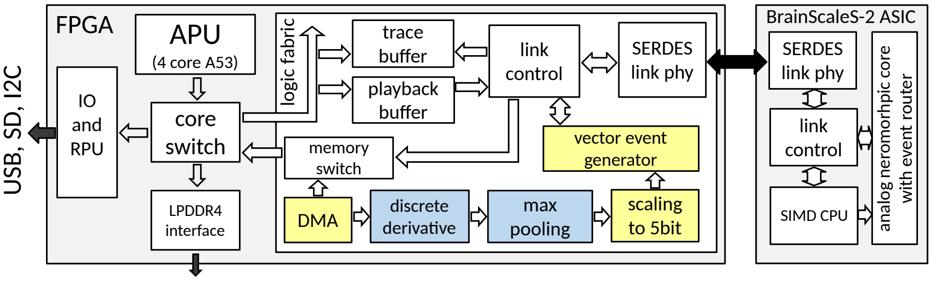

Figure 5. **Block diagram of the major functional units of the FPGA, the part inside the logic fabric has been realized as custom RTL in SystemVerilog. The DMA controller, preprocessing chain elements and vector event generator create the input activation events representing the vector in the vector-matrix multiplication. Some of the preprocessing (blue) is problem-specific for the medical ECG dataset. To the right side the major blocks of the BSS-2 ASIC are shown as well to illustrate the complete communication path from the embedded SIMD CPUs to the DRAM memory. The arrows denote the control flow direction from initiator to follower of the internal (hollow) and external (filled) data buses shown in the figure.**

SIMD CPU inside the ASIC synchronizes the vector event generator inside the FPGA using multiple handshake signals to control the timing of the sent events.

The four 64 bit ARM processor cores contained in the FPGA usually do not participate in the inner loop of the inference calculation and only perform system initialization tasks. Making use of their flexible I/O, they can however be used to form a tight, low-latency coupling between sensors, actors and the neuromorphic ASIC.

D. Software

Similar to other neuromorphic hardware platforms software is an essential component to make complex hardware systems accessible to users, e.g., GraphCore [32], Loihi [33, 34, 35], Neurogrid [36, 37], SpiNNaker [38, 39, 40], Tianjic [41], and TrueNorth [42]. A recent publication covering the older BrainScaleS-1 (BSS-1) platform shortly compares software approaches of multiple neuromorphic systems [43].

In each phase – from hardware commissioning, to model design, to training, to validation – users can take advantage of a software environment that provides appropriate abstraction levels, access to hardware debugging information as well as robust and transparent platform operation. For the BSS-2 architecture, – and, in particular, the mobile system – we provide software support for different system aspects:

User Interface

The PyTorch toolkit [44] is a commonly used workhorse in the field. Particularly, it simplifies many aspects of CDNN modeling. We developed a custom extension for PyTorch, *hxtorch* [45], providing support for the BSS-2 architecture.

Training

Forward propagation is dispatched to the BSS-2 ASIC while backward propagation is performed in software. Hence, *hxtorch* enables using the BrainScaleS-2 system as an inference accelerator in PyTorch while adopting a hardware-in-the-loop-based training approach. The trained model can be serialized, stored to disk, and used in a *standalone inference mode* to increase energy efficiency. In addition, a "mock mode" enables the simulation of certain hardware properties in software. This facilitates migrating from the training of a pure software model to hardware-in-the-loop-based training.

Hardware Resources

hxtorch provides support for the execution of neural network graphs on an arbitrary number of BSS-2 ASICs. Individual layers are partitioned into chip-sized chunks and executed either in parallel, serially, or in the appropriate mixture needed to fit on the available hardware resources. Finally, each ASIC receives and executes a stream of instructions and data.

Data-Flow Graph Execution

Internally, model layers in *hxtorch* build up a data-flow graph. A just-in-time (JIT) compiler traverses the graph and partitions individual layers into chunks fitting onto the available hardware resources. Partitioned layers are converted into configuration data and control flow statements; both of which are transferred to the BSS-2 hardware system and result data is read back. Regarding control flow, the hardware execution engine supports two modes: the first mode uses the FPGA to handle control flow; the second mode, which is also largely used in the standalone inference mode, hands over the control flow to the embedded SIMD CPUs of the ASIC.

Memory Management

Data input, as well as output locations, are precomputed by the BSS-2 software stack allowing for static memory management on the system. The SIMD CPUs use the communication link to the FPGA to program the DMA engine inside the FPGA to automatically deliver the input activations from DRAM to the analog processing cores. Analog operation results are read out by the processors, either held in SRAM for temporary data, or stored back into DRAM for output data.

Standalone Inference Mode

The BSS-2 software layers are written in C++ and provide faster execution speeds compared to an interpreted high-level language such as Python. To create a lightweight inference flow for the energy measurements, a stand-alone version of the *hxtorch* hardware graph executor was developed. This executor is implemented as a standalone binary and builds upon the same internal software layers and data formats as the *hxtorch* extension. In contrast to the JIT-based execution flow, the standalone inference mode requires control flow to be handled by the embedded SIMD CPUs. The processors operate on an instruction stream representing: data load and store operations, trigger operations for delivery of input activations from the FPGA, reading out the neuron membrane values, or performing digital operations that are not supported by the analog substrate.

Embedded System Environment

The BSS-2 mobile system includes a Linux environment[3] running on an embedded ARM64 processor. We take advantage of a fully containerized software environment based on singularity [46] and spack [47] to provide a cross-compiler environment on the host computer as well as on the embedded Linux system. Standard Linux drivers (xHCI, mass storage, FAT32) are used to read out test data from a USB mass storage device; additionally, support for USB-based Ethernet networking hardware is enabled to facilitate remote system usage. An experiment execution service enables users to run Python-based interfaces on host computers that exchange serialized experiment configurations and result data with the mobile system.

Details on *hxtorch* for rate-based hardware operation can be found in Spilger et al. [45]. A general overview of the software stack for BSS-2, including spiking hardware operation, can be found in Müller et al. [48].

III. Showcase: ECG classification

We showcase the BSS-2 mobile system by classifying A-fib in the medical ECG dataset introduced in Section I. This real-world task demonstrates many of the platform's features, such as stand-alone operation, mobility, power efficiency and external connectivity. We deploy a trained model on the system, which then autonomously classifies ECG data supplied via a USB connection.

A. Model

The model design is mainly governed by network size tradeoffs between high accuracy and short runtime. Networks that exceed the size of the compute substrate pose a high runtime and I/O penalty due to frequent reconfiguration. This issue especially becomes relevant for non-batched operation, while it diminishes for large batch sizes. Targeting edge applications, we restrict the inference runs to a batch size of one.

Evaluation of network models showed that a small network that fits on a single chip and does not require reconfiguration can achieve reasonable classification performance. The network used in this showcase is depicted in Fig. 6. It operates on 13.5 s of the 120 s long ECG records, as this has turned out to be sufficient for classification of A-fib. To the left, the graph of the model is shown. It consists of one convolutional and two linear layers. The small size of the network allows it to be completely realized on the ASIC. The calculations in its convolutional first layer can be performed fully in parallel, as well as those in the second and third layers: this mapping to the two halves of a BSS-2 ASIC is shown on the right side of the figure. The ReLU and the final argmax operations are performed in the embedded SIMD CPUs after digital readout of the analog neuron membrane voltages (cf. Section II-A).

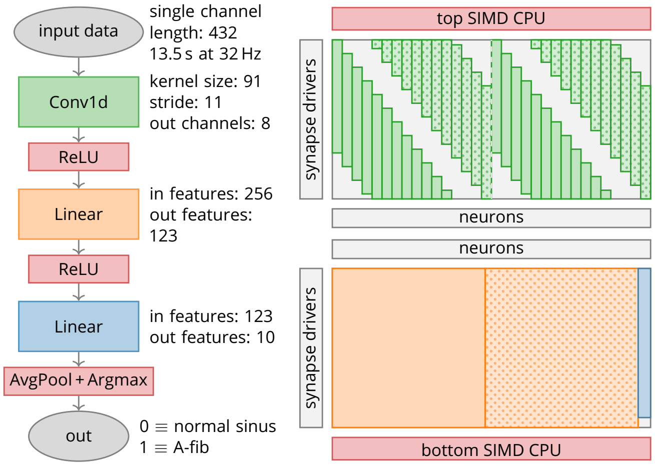

Figure 6. Layer structure (left) and on-chip arrangement (right) of the used deep convolutional neural network model. The convolutional layer (green) is processed in the upper synapse array, the identical weight is arranged 32 times on the substrate to enable parallel processing. All ReLUs (red) are performed in digital logic by the two SIMD CPUs. The further processing takes place on the lower synapse array with a fully connected layer and 123 hidden neurons (orange). To ensure efficient use of the substrate, it is divided into two parts and placed side by side. The dotted part of the layer receives the second half of inputs at the same time and is processed in parallel. The actual classification is then achieved in the last layer (blue) with 10 neurons on the right, which are combined into two logical neurons by average pooling, effectively reducing analog noise.

The ASIC operates on positive activations with 5 bit resolution. Since the raw data samples as input for the inference calculation are provided as 12 bit values with higher dynamic range, some preprocessing is required. Fig. 7 illustrates the performed steps. To avoid unnecessary data movement, the preprocessing is done in the FPGA fabric by a custom processing chain. In the first step of the preprocessing, a discrete derivative of the original signal is calculated to suppress the large baseline fluctuations of the signal. In a second step, the data rate is reduced by calculating the difference between the maximum and the minimum of 32 samples. The resulting samples are quantized to 5 bit and used as inputs to the analog vector-matrix multiplications performed within the ASIC.

B. Training

Training relies on the proven backpropagation algorithm for CDNNs [49]. To facilitate fast prototyping when training the network described in Section III-A, a mathematical abstraction of the hardware operations was implemented on top of PyTorch [44] in *hxtorch* [45]. Incorporating hardware-

[3] Petalinux; the build flow of the embedded Linux distribution is provided by the FPGA manufacturer.

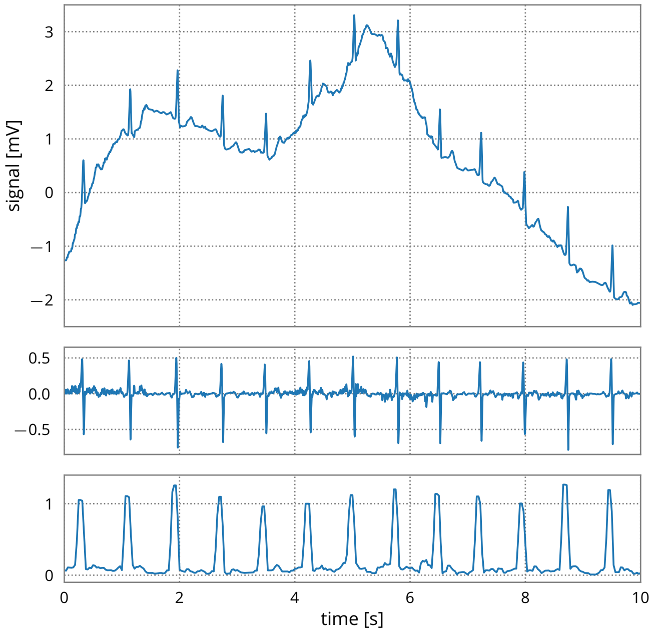

Figure 7. **Preprocessing steps performed in the FPGA fabric (from top to bottom): The raw, i.e. unprocessed, input sample is transformed by taking a discrete derivative to reduce baseline fluctuations. Subsequent maximum-minimum difference pooling reduces the sample rate and provides positive activations, which form the final input signal to the CDNN in the ASIC. Original data taken from Clifford et al. [27].**

related constraints like fixed-pattern noise and limited dynamic range, it enables the training of initial models in software and provides gradient information for the backward-pass when training on hardware. Final model parameters as presented in Section IV, however, were trained on the ASIC following a hardware-in-the-loop approach [50]: The forward pass is evaluated on BSS-2, whereas the backward pass and parameter updates are calculated on the host computer using *hxtorch*. Tensor data structures are seamlessly converted to hardware resolution and back. Data partitioning and experiment control is handled by both on-chip SIMD CPUs (see Section II-D). To the user, the training procedure is completely embedded within PyTorch. To increase robustness and decrease sensitivity to hardware variations, we replace the average pooling in the last layer by a max pooling operation during training. We employ early stopping whenever no substantial improvement is observed between training epochs.

IV. Results

The performance of the presented system has been evaluated by assigning a set of ECG traces to two classes: patients with sinus rhythm and patients showing atrial fibrillation (Section I). Mimicking the expected workload in a low-energy edge application, all data has been processed with a batch size of one. To increase the accuracy of all measurements, data was processed in blocks of 500 traces. For each block, runtime and energy consumption have been measured using the sensors described in Section II-B and afterwards averaged down to a single inference. The power consumption was measured with a sampling rate of 294 Hz for sensors on the system controller and 4.4 kHz for sensors on the ASIC adapter PCB.

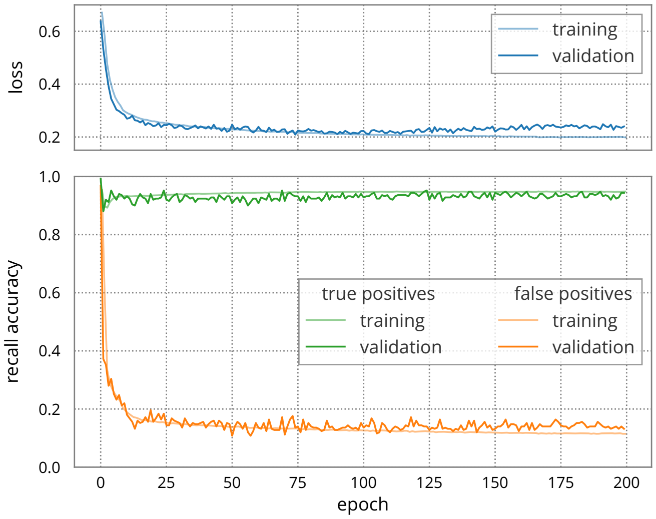

Figure 8. **Training and validation metrics of the model presented in Fig. 6 performed with the BSS-2 ASIC. The test set of 500 records was split from the provided ECG dataset prior to training.**

Classification accuracy has been evaluated by selecting randomized test sets of 500 records prior to training. Metrics of such a training course on the presented system are shown in Fig. 8. With the shown combination of model, software and hardware, this system classified A-fib with a detection rate of $(93.7 \pm 0.7)\,\%$ at $(14.0 \pm 1.0)\,\%$ false positives.

Each block of 500 input traces was found to be processed in 138 ms; starting with raw ECG data in the system controller DRAM and ending with binary classification results ibidem. Table 1 gives an overview over the achieved results: During the inference phase, the system achieved 477 MOp/s with a mean power consumption of 5.6 W. In its current state, classification on the BSS-2 mobile system takes 276 µs and consumes a total of 1.56 mJ per ECG trace, of which 192 µJ were consumed by the BSS-2 ASIC.

V. Discussion & Conclusions

We have presented the BSS-2 mobile system as an analog inference platform and demonstrated medical ECG data classification as one possible application.

The small system is mobile by design and has proven to operate reliably under various environmental conditions. Despite its early prototype stage, it is therefore directly applicable to inference tasks on the edge: The results we have achieved demonstrate that the presented system is sufficiently energy-efficient to run on battery while monitoring the health of a patient. Based on the energy consumption presented in Table 1, a common CR2032 lithium button battery with an approximated energy content of 200 mA h would power the inference calculations for detecting atrial fibrillation in two-minute intervals for five years. At the cost of runtime and thus energy efficiency, we can utilize larger networks

Table 1. Measured results for the classification of a single ECG trace. To increase measurement precision, data has been acquired as a block of 500 traces, which were classified in direct succession with batch size one. Unless noted otherwise, the given number represents the mean value from this set. The tested records have been excluded from training.

quantity	value		unit
time per inference	276	10^{-6}	s
power consumption (system)	5.6		W
power consumption (BSS-2 ASIC)	0.69		W
energy (total)	1.56	10^{-3}	J
energy (system controller, total)	0.7	10^{-3}	J
energy (system controller, ARM CPU)	0.34	10^{-3}	J
energy (system controller, FPGA)	0.21	10^{-3}	J
energy (system controller, DRAM)	0.12	10^{-3}	J
energy (ASIC, total)	0.19	10^{-3}	J
energy (ASIC, IO)	0.07	10^{-3}	J
energy (ASIC, analog)	0.07	10^{-3}	J
energy (ASIC, digital)	0.07	10^{-3}	J
total operations in CDNN	132	10^{3}	Op
BSS-2 ASIC processing speed (mult./acc.)	477	10^{6}	Op/s
BSS-2 ASIC energy efficiency (mult./acc.)	689	10^{6}	Op/J
BSS-2 ASIC energy efficiency (inferences)	5.25	10^{3}	1/J
classification accuracy			
detection rate	93.7 ± 0.7		%
false positives	14.0 ± 1.0		%

to increase the classification accuracy. On the BSS-2 ASIC, we have achieved accuracies of up to 95.5 % for A-fib with 8.0 % false positives.

The achieved detection rates on the BSS-2 mobile system are on par with other state-of-the-art solutions: Rizwan et al. [51] report atrial fibrillation detection rates for machine-learning-based solvers from 80.0 % to 100.0 % with a median of 96.3 % (1.09 % to 26.4 % false positives, median: 6.9 %). Solvers based on classical time series analysis reach 74.2 % to 99.6 % with a median of 97.1 % (1.7 % to 10.2 % false positives, median: 3.2 %), as presented by Marsili et al. [52]. Most of these solutions, however, do not target the low power envelope required for edge applications. In contrast, Azariadi et al. [53], Seitanidis et al. [54] use the off-the-shelve Intel Galileo and Nvidia Jetson Nano platforms to classify ECG anomalies with an energy consumption of 220 mJ and 7.4 mJ per inference.[4] With a similar system controller and power consumption, the presented BSS-2 mobile system only consumes 1.56 mJ per classification. Designed as a generic computational substrate for a multitude of applications, it can however not compete with ASICs specifically built for low-power A-fib classification: Andersson et al. [55] present a classifier that achieves a comparable detection rate of 94.9 % (4.7 % false positives) with a power envelope of only 334 nW.

In addition to the presented multiply-accumulate functionality, BSS-2 is designed to operate as an analog emulator for SNNs. Cramer et al. [23] present classifiers on multiple common datasets that make use of this mode to achieve beyond state-of-the-art classification latency and energy efficiency on BSS-2. To the best of our knowledge, it is the first and only available system to accelerate both, multiply-accumulate operations and SNNs in the analog domain. Due to the stateful nature of the necessary time-continuous operations, multiplexing of analog resources is seldom possible in SNN accelerators, therefore limiting the maximum model size to the available hardware resources. In contrast, rate-based stateless operation using our analog neuromorphic core as a parallel vector-matrix multiplier allows for multiplexing hardware resources in time and therefore has the advantage of supporting arbitrarily large model sizes. Such networks are only limited by the available memory. Most models that are capable of performing real-world tasks, like video analysis or speech translation, need model sizes in the order of 10^7 to 10^9 parameters [56]. These network sizes are feasible with the presented system, as neither the hardware platform nor the *hxtorch* software environment impose size limitations on the model in use.

The combination of spiking and convolutional neural networks on a single substrate therefore greatly widens the application of SNNs in edge applications: it allows features to be extracted by conventional high dimensional CDNN layers on multiplexed hardware resources, while sparse spiking layers can simultaneously be used for their final classification. Using the embedded SIMD CPUs, BSS-2 can utilize online learning for the SNN layers [21] and thereby improve classification performance and adapt to environmental changes in the field.

Given its early prototyping stage, the system as well as the BSS-2 chip itself contain a large potential for optimization. Currently, the FPGA is primarily used as a memory controller for the ASIC – functionality that could be incorporated into the chip's digital core. This would remove the power consumption of the FPGA from the system's energy balance and would increase the bandwidth between memory and analog core.

The main motivation during the development of the BSS-2 ASIC was to enable flexible on-chip online learning in SNNs. Thus, the speed of the analog CDNN calculation has not yet been optimized. While the synapse arrays that perform the multiply-accumulate operation already support 32.8 TOp/s, see (1), the usage of the spike-based neurons for the integration of the summation currents limits the actual speed to approximately 52 GOp/s, see (2).

The current area efficiency of the analog MAC in the synapse arrays can be calculated as

$$\frac{32.8\,\mathrm{TOp/s}}{256 \cdot 512 \cdot 8\,\mu\mathrm{m} \cdot 12\,\mu\mathrm{m}} = 2.6\,\mathrm{TOp/(s\,mm^2)}. \qquad (3)$$

[4] We assume a power consumption of 2.2 W for the Intel Galileo and 5.0 W for the Nvidia Jetson Nano system and use the published inference runtimes to estimate the energy per inference.

As a conservative approximation based on the current die size of 32 mm^2, we target an area efficiency above 1 TOp/(s mm^2) for the full chip. State-of-the-art implementations using similar technologies and architectures reach up to 0.32 TOp/(s mm^2) based on full die size [6, 25].

Multiple approaches have to be taken to make use of the aforementioned processing speed of the synapse array: First, specialized circuits for the integration of the synapses' output currents in the non-spiking operation mode of the ASIC have to be integrated. These specialized accumulators could be combined with revised parallel ADCs that are – in contrast to the currently implemented design – capable of sufficient conversion speed. The increased data rate will require higher I/O bandwidth that could be achieved by the aforementioned integration of an on-chip memory controller.

In its current state, the BrainScaleS-2 system is available to the scientific community via the EBRAINS project[5]. Example applications using SNNs as well as the built-in multiply-accumulate functionality are available and can be executed online through a browser-based interface. Hardware access to the BSS-2 (mobile) system is available upon request.

Contributions

Yannik Stradmann directed the development and modeling efforts for the presented experiment and hardware setup. He contributed to all components. Sebastian Billaudelle contributed to the chip design, chip commissioning and implementation of the experiment. Oliver Breitwieser contributed to the software stack, is the main architect of the preemptive experiment scheduling service and contributed to modeling and model verification. Falk Ebert is a main contributor to the energy measurement system. Arne Emmel developed and implemented the model, designed the preprocessing, adapted the training to the hardware platform and contributed to the software integration. Dan Husmann developed the ASIC adapter PCB. Joscha Ilmberger is the main system developer contributing to PCB design, porting of the FPGA design to the new platform and adding functionality such as preprocessing and the vector event generator. Eric Müller is the lead developer and architect of the BSS-2 software stack; he commissioned the embedded platform, ported the software development environment as well as the BSS-2 software stack to the embedded FPGA platform. Philipp Spilger is the main developer of the software for the non-spiking operation mode of the BSS-2 ASIC and a contributor to the software stack. Johannes Weis is the main developer of calibration routines for the analog network core, commissioned the first non-spiking experiments on the hardware platform and contributed to the model. Johannes Schemmel is the lead designer and architect of the BSS-2 neuromorphic system. He wrote the initial version of the paper. All authors contributed to and edited the final manuscript.

Acknowledgments

The authors wish to thank all present and former members of the Electronic Vision(s) research group contributing to the BrainScaleS-2 neuromorphic platform. For the publication fee the authors acknowledge financial support by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) within the funding programme „Open Access Publikationskosten“ as well as by Heidelberg University.

References

[1] Coral. (2020, Aug.) Edge TPU performance benchmarks. [Online]. Available: https://coral.ai/docs/edgetpu/benchmarks/

[2] D. Moloney, B. Barry, R. Richmond, F. Connor, C. Brick, and D. Donohoe, “Myriad 2: Eye of the computational vision storm,” in *2014 IEEE Hot Chips 26 Symposium (HCS)*, 2014, pp. 1–18.

[3] B. Hickmann, J. Chen, M. Rotzin, A. Yang, M. Urbanski, and S. Avancha, “Intel nervana neural network processor-t (NNP-T) fused floating point many-term dot product,” in *2020 IEEE 27th Symposium on Computer Arithmetic (ARITH)*, 2020, pp. 133–136.

[4] A. Reuther, P. Michaleas, M. Jones, V. Gadepally, S. Samsi, and J. Kepner, “Survey and benchmarking of machine learning accelerators,” in *2019 IEEE High Performance Extreme Computing Conference (HPEC)*, 2019, pp. 1–9.

[5] L. Fick, D. Blaauw, D. Sylvester, S. Skrzyniarz, M. Parikh, and D. Fick, “Analog in-memory subthreshold deep neural network accelerator,” in *2017 IEEE Custom Integrated Circuits Conference (CICC)*, 2017, pp. 1–4.

[6] L. Fick, S. Skrzyniarz, M. Parikh, M. B. Henry, and D. Fick, “Analog matrix processor for edge ai real-time video analytics,” in *2022 IEEE International Solid- State Circuits Conference (ISSCC)*, vol. 65, 2022, pp. 260–262.

[7] C. A. Mead and M. A. Mahowald, “A silicon model of early visual processing,” *Neural Networks*, vol. 1, no. 1, pp. 91–97, 1988.

[8] S. Moradi, N. Qiao, F. Stefanini, and G. Indiveri, “A scalable multicore architecture with heterogeneous memory structures for dynamic neuromorphic asynchronous processors (DYNAPs),” *IEEE Trans. Biomed. Circuits Syst.*, vol. 12, no. 1, pp. 106–122, 2018.

[9] B. V. Benjamin, P. Gao, E. McQuinn, S. Choudhary, A. R. Chandrasekaran, J.-M. Bussat, R. Alvarez-Icaza, J. V. Arthur, P. A. Merolla, and K. Boahen, “Neurogrid: A mixed-analog-digital multichip system for large-scale neural simulations,” *Proceedings of the IEEE*, vol. 102, no. 5, pp. 699–716, 2014.

[10] M. Khan, D. Lester, L. A. Plana, A. Rast, X. Jin, E. Painkras, and S. B. Furber, “Spinnaker: mapping neural networks onto a massively-parallel chip multiprocessor,” in *Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*. IEEE, 2008, pp. 2849–2856.

[11] P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza, A. S. Cassidy, J. Sawada, F. Akopyan, B. L. Jackson, N. Imam, C. Guo, Y. Nakamura *et al.*, “A million spiking-neuron integrated circuit with a scalable communication network and interface,” *Science*, vol. 345, no. 6197, pp. 668–673, 2014.

[12] C. Frenkel, M. Lefebvre, J.-D. Legat, and D. Bol, “A 0.086-mm^{2}12.7-pj/sop 64k-synapse 256-neuron online-learning digital spiking neuromorphic processor in 28-nm cmos,” *IEEE transactions on biomedical circuits and systems*, vol. 13, no. 1, pp. 145–158, 2018.

[5] https://ebrains.eu/register

[13] C. Frenkel, J.-D. Legat, and D. Bol, "Morphic: A 65-nm 738k-synapse/mm^2 quad-core binary-weight digital neuromorphic processor with stochastic spike-driven online learning," *IEEE transactions on biomedical circuits and systems*, vol. 13, no. 5, pp. 999–1010, 2019.

[14] ——, "A 28-nm convolutional neuromorphic processor enabling online learning with spike-based retinas," in *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2020, pp. 1–5.

[15] C. Frenkel and G. Indiveri, "ReckOn: A 28nm sub-mm2 task-agnostic spiking recurrent neural network processor enabling on-chip learning over second-long timescales," in *2022 IEEE International Solid- State Circuits Conference (ISSCC)*, vol. 65, 2022, pp. 1–3.

[16] C. Mayr, S. Hoeppner, and S. Furber, "Spinnaker 2: A 10 million core processor system for brain simulation and machine learning," *arXiv preprint arXiv:1911.02385*, 2019.

[17] M. Davies, N. Srinivasa, T.-H. Lin, G. Chinya, Y. Cao, S. H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain *et al.*, "Loihi: A neuromorphic manycore processor with on-chip learning," *IEEE Micro*, vol. 38, no. 1, pp. 82–99, 2018.

[18] V. Joshi, M. L. Gallo, S. Haefeli, I. Boybat, S. R. Nandakumar, C. Piveteau, M. Dazzi, B. Rajendran, A. Sebastian, and E. Eleftheriou, "Accurate deep neural network inference using computational phase-change memory," *Nature Communications*, vol. 11, no. 1, May 2020.

[19] L. Chua, V. Sbitnev, and H. Kim, "Hodgkin–huxley axon is made of memristors," *International Journal of Bifurcation and Chaos*, vol. 22, no. 03, p. 1230011, Mar. 2012.

[20] B. J. Shastri, A. N. Tait, T. Ferreira de Lima, W. H. P. Pernice, H. Bhaskaran, C. D. Wright, and P. R. Prucnal, "Photonics for artificial intelligence and neuromorphic computing," *Nature Photonics*, vol. 15, no. 2, pp. 102–114, 2021.

[21] C. Pehle, S. Billaudelle, B. Cramer, J. Kaiser, K. Schreiber, Y. Stradmann, J. Weis, A. Leibfried, E. Müller, and J. Schemmel, "The BrainScaleS-2 accelerated neuromorphic system with hybrid plasticity," *Frontiers in Neuroscience*, vol. 16, 2022.

[22] B. Klein, L. Kuhn, J. Weis, A. Emmel, Y. Stradmann, J. Schemmel, and H. Fröning, "Towards addressing noise and static variations of analog computations using efficient retraining," in *Machine Learning and Principles and Practice of Knowledge Discovery in Databases*. Cham: Springer International Publishing, 2021, pp. 409–420.

[23] B. Cramer, S. Billaudelle, S. Kanya, A. Leibfried, A. Grübl, V. Karasenko, C. Pehle, K. Schreiber, Y. Stradmann, J. Weis *et al.*, "Surrogate gradients for analog neuromorphic computing," *Proceedings of the National Academy of Sciences*, vol. 119, no. 4, 2022.

[24] J. Pei, L. Deng, S. Song, M. Zhao, Y. Zhang, S. Wu, G. Wang, Z. Zou, Z. Wu, W. He, F. Chen, N. Deng, S. Wu, Y. Wang, Y. Wu, Z. Yang, C. Ma, G. Li, W. Han, H. Li, H. Wu, R. Zhao, Y. Xie, and L. Shi, "Towards artificial general intelligence with hybrid tianjic chip architecture," *Nature*, vol. 572, no. 7767, pp. 106–111, Aug. 2019.

[25] D. Kim, B. Chakraborty, X. She, E. Lee, B. Kang, and S. Mukhopadhyay, "MONETA: A processing-in-memory-based hardware platform for the hybrid convolutional spiking neural network with online learning," *Frontiers in Neuroscience*, vol. 16, Apr. 2022.

[26] J. Weis, P. Spilger, S. Billaudelle, Y. Stradmann, A. Emmel, E. Müller, O. Breitwieser, A. Grübl, J. Ilmberger, V. Karasenko, M. Kleider, C. Mauch, K. Schreiber, and J. Schemmel, "Inference with artificial neural networks on analog neuromorphic hardware," in *IoT Streams for Data-Driven Predictive Maintenance and IoT, Edge, and Mobile for Embedded Machine Learning*. Cham: Springer International Publishing, 2020, pp. 201–212.

[27] G. D. Clifford, C. Liu, B. Moody, L.-W. H. Lehman, I. Silva, Q. Li, A. E. Johnson, and R. G. Mark, "Af classification from a short single lead ecg recording: the physionet/computing in cardiology challenge 2017," *Computing in Cardiology*, vol. 44, 2017.

[28] Xilinx, *Zync UltraScale+ MPSoC Data Sheet*, 2019. [Online]. Available: https://www.xilinx.com/support/documentation/data_sheets/ds891-zynq-ultrascale-plus-overview.pdf

[29] PowerISA, "PowerISA version 2.06 revision b," Power.org, Specification, Jul. 2010. [Online]. Available: http://www.power.org/resources/reading/

[30] Texas Instruments, *INA219 Zerø-Drift, Bidirectional Current/Power Monitor With I2C Interface*, 2020. [Online]. Available: https://www.ti.com/lit/ds/symlink/ina219.pdf

[31] AVNET, *Ultra96-V2*, 2020. [Online]. Available: http://zedboard.org/sites/default/files/product_briefs/5365-pb-ultra96-v2-v10b.pdf

[32] I. Kacher, M. Portaz, H. Randrianarivo, and S. Peyronnet, "Graphcore c2 card performance for image-based deep learning application: A report," *arXiv preprint*, Feb. 2020.

[33] B. Rueckauer, C. Bybee, R. Goettsche, Y. Singh, J. Mishra, and A. Wild, "NxTF: An api and compiler for deep spiking neural networks on intel loihi," *arXiv preprint*, Jan. 2021.

[34] T. DeWolf, P. Jaworski, and C. Eliasmith, "Nengo and low-power ai hardware for robust, embedded neurorobotics," *Frontiers in Neurorobotics*, vol. 14, 2020.

[35] C.-K. Lin, A. Wild, G. N. Chinya, Y. Cao, M. Davies, D. M. Lavery, and H. Wang, "Programming spiking neural networks on intel's loihi," *Computer*, vol. 51, no. 3, pp. 52–61, 2018.

[36] B. V. Benjamin, N. A. Steinmetz, N. N. Oza, J. J. Aguayo, and K. Boahen, "Neurogrid simulates cortical cell-types, active dendrites, and top-down attention," *Neuromorphic Computing and Engineering*, vol. 1, no. 1, p. 013001, 2021.

[37] A. R. Voelker, B. V. Benjamin, T. C. Stewart, K. Boahen, and C. Eliasmith, "Extending the neural engineering framework for nonideal silicon synapses," in *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2017, pp. 1–4.

[38] O. Rhodes, P. A. Bogdan, C. Brenninkmeijer, S. Davidson, D. Fellows, A. Gait, D. R. Lester, M. Mikaitis, L. A. Plana, A. G. D. Rowley, A. B. Stokes, and S. B. Furber, "spynnaker: A software package for running pynn simulations on spinnaker," *Frontiers in Neuroscience*, vol. 12, p. 816, 2018.

[39] A. G. D. Rowley, C. Brenninkmeijer, S. Davidson, D. Fellows, A. Gait, D. R. Lester, L. A. Plana, O. Rhodes, A. B. Stokes, and S. B. Furber, "Spinntools: The execution engine for the spinnaker platform," *Frontiers in Neuroscience*, vol. 13, p. 231, 2019.

[40] F. Galluppi, X. Lagorce, E. Stromatias, M. Pfeiffer, L. A. Plana, S. B. Furber, and R. B. Benosman, "A framework for plasticity implementation on the spinnaker neural architecture," *Frontiers in Neuroscience*, vol. 8, no. 429, 2015.

[41] Y. Ji, Y. Zhang, S. Li, P. Chi, C. Jiang, P. Qu, Y. Xie, and W. Chen, "Neutrams: Neural network transformation and co-design under neuromorphic hardware constraints," in *2016 49th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*. IEEE, 2016,

pp. 1–13.

[42] A. Amir, P. Datta, W. P. Risk, A. S. Cassidy, J. A. Kusnitz, S. K. Esser, A. Andreopoulos, T. M. Wong, M. Flickner, R. Alvarez-Icaza, E. McQuinn, B. Shaw, N. Pass, and D. S. Modha, "Cognitive computing programming paradigm: A corelet language for composing networks of neurosynaptic cores," in *The 2013 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2013, pp. 1–10.

[43] E. Müller, S. Schmitt, C. Mauch, S. Billaudelle, A. Grübl, M. Güttler, D. Husmann, J. Ilmberger, S. Jeltsch, J. Kaiser, J. Klähn, M. Kleider, C. Koke, J. Montes, P. Müller, J. Partzsch, F. Passenberg, H. Schmidt, B. Vogginger, J. Weidner, C. Mayr, and J. Schemmel, "The operating system of the neuromorphic BrainScaleS-1 system," *Neurocomputing*, vol. 501, pp. 790–810, 2022.

[44] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035.

[45] P. Spilger, E. Müller, A. Emmel, A. Leibfried, C. Mauch, C. Pehle, J. Weis, O. Breitwieser, S. Billaudelle, S. Schmitt, T. C. Wunderlich, Y. Stradmann, and J. Schemmel, "hxtorch: PyTorch for BrainScaleS-2 — perceptrons on analog neuromorphic hardware," in *IoT Streams for Data-Driven Predictive Maintenance and IoT, Edge, and Mobile for Embedded Machine Learning*. Cham: Springer International Publishing, 2020, pp. 189–200.

[46] G. M. Kurtzer, V. Sochat, and M. W. Bauer, "Singularity: Scientific containers for mobility of compute," *PLOS ONE*, vol. 12, no. 5, pp. 1–20, 05 2017.

[47] T. Gamblin, M. LeGendre, M. R. Collette, G. L. Lee, A. Moody, B. R. de Supinski, and S. Futral, "The spack package manager: Bringing order to hpc software chaos," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, ser. SC '15. New York, NY, USA: ACM, 2015, pp. 40:1–40:12.

[48] E. Müller, E. Arnold, O. Breitwieser, M. Czierlinski, A. Emmel, J. Kaiser, C. Mauch, S. Schmitt, P. Spilger, R. Stock, Y. Stradmann, J. Weis, A. Baumbach, S. Billaudelle, B. Cramer, F. Ebert, J. Göltz, J. Ilmberger, V. Karasenko, M. Kleider, A. Leibfried, C. Pehle, and J. Schemmel, "A scalable approach to modeling on accelerated neuromorphic hardware," *Front. Neurosci.*, vol. 16, 2022.

[49] D. E. Rumelhart, G. E. Hinton, and W. R.J., "Learning internal representations by error propagation," *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*, vol. I, pp. 318–362, 1986.

[50] S. Schmitt, J. Klähn, G. Bellec, A. Grübl, M. Güttler, A. Hartel, S. Hartmann, D. Husmann, K. Husmann, S. Jeltsch, V. Karasenko, M. Kleider, C. Koke, A. Kononov, C. Mauch, E. Müller, P. Müller, J. Partzsch, M. A. Petrovici, B. Vogginger, S. Schiefer, S. Scholze, V. Thanasoulis, J. Schemmel, R. Legenstein, W. Maass, C. Mayr, and K. Meier, "Neuromorphic hardware in the loop: Training a deep spiking network on the brainscales wafer-scale system," *Proceedings of the 2017 IEEE International Joint Conference on Neural Networks*, 2017.

[51] A. Rizwan, A. Zoha, I. B. Mabrouk, H. M. Sabbour, A. S. Al-Sumaiti, A. Alomainy, M. A. Imran, and Q. H. Abbasi, "A review on the state of the art in atrial fibrillation detection enabled by machine learning," *IEEE Reviews in Biomedical Engineering*, vol. 14, pp. 219–239, 2021.

[52] I. A. Marsili, L. Biasiolli, M. Masè, A. Adami, A. O. Andrighetti, F. Ravelli, and G. Nollo, "Implementation and validation of real-time algorithms for atrial fibrillation detection on a wearable ECG device," *Computers in Biology and Medicine*, vol. 116, p. 103540, 2020.

[53] D. Azariadi, V. Tsoutsouras, S. Xydis, and D. Soudris, "ECG signal analysis and arrhythmia detection on iot wearable medical devices," in *2016 5th International Conference on Modern Circuits and Systems Technologies (MOCAST)*, 2016, pp. 1–4.

[54] P. Seitanidis, J. Gialelis, and G. Papaconstantinou, "Identifying heart arrhythmias through multi-level algorithmic processing of ECG on edge devices," *Procedia Computer Science*, vol. 203, pp. 699–706, 2022.

[55] O. Andersson, K. H. Chon, L. Sörnmo, and J. N. Rodrigues, "A 290 mv sub-v_{t} asic for real-time atrial fibrillation detection," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 9, no. 3, pp. 377–386, 2015.

[56] R. Aharoni, M. Johnson, and O. Firat, "Massively multilingual neural machine translation," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 3874–3884.

Yannik Stradmann received the M.Sc. degree in Physics from Heidelberg University, Germany, in 2019. Currently, he is a Ph.D. student in the Electronic Vision(s) group at Heidelberg University. His research focuses on the development and characterization of mixed-signal VLSI circuits for neuromorphic hardware and their application for real-time control tasks.

Sebastian Billaudelle received his Ph.D. degree in physics from Heidelberg University, Germany in 2022. As a postdoc in the Electronic Vision(s) group at Heidelberg University, he contributes to the analog neuromorphic circuits of BrainScaleS-2 and devises training and learning algorithms for the neuromorphic system.

Oliver Breitwieser received the Ph.D. degree in Physics from Heidelberg University, Germany, in 2021. His research interests include neuroscience, machine learning, particularly applied to neuromorphic hardware, as well as sustainable distributed large-scale computing. He is a core developer of the BrainScaleS operating system and responsible for operations of the BrainScaleS compute infrastructure.

Falk Leonard Ebert is studying physics at Heidelberg University, Germany and currently working on his B.Sc. thesis in the Electronic Vision(s) group at the Kirchhoff-Institute for Physics.

Arne Emmel received the M.Sc. degree in physics from Heidelberg University, Germany, in 2020. Currently, he is a researcher in the Electronic Vision(s) group at Kirchhoff-Institute for Physics, Heidelberg University. His interests include modeling and the exploration and optimization of training algorithms for the specific requirements of analog neuromorphic hardware.

Dan Husmann received the diploma degree in physics from Heidelberg University, Germany, in 2000. Currently, he is a researcher in the Electronic Vision(s) group at Kirchhoff-Institute for Physics, Heidelberg University. His research interest are wafer-scale integration techniques and building of neuromorphic hardware.

Joscha Ilmberger received his M.Sc. degree in physics from Heidelberg University, Germany in 2017. As a Ph.D. student in the Electronic Vision(s) group at Heidelberg University, his research interests are scaling and digital architecture design of novel analog neuromorphic hardware.

Eric Müller received the Ph.D. degree in physics from Heidelberg University, Germany, in 2014. Currently, he is a researcher in the Electronic Vision(s) group at Kirchhoff-Institute for Physics, Heidelberg University. His research interests are large-scale computing, information processing in closed-loop environments, and non-von-Neumann computing paradigms. He is the architect of the BrainScaleS operating system and leads BrainScaleS software development.

Philipp Spilger received the M.Sc. degree in physics from Heidelberg University, Germany, in 2021. Currently, he is a Ph.D. student in the Electronic Vision(s) group at Kirchhoff-Institute for Physics, Heidelberg University. His research interests are software abstraction for control and configuration of neuromorphic hardware based on optimization and compilation techniques.

Johannes Weis received the M.Sc. degree in physics from Heidelberg University, Germany, in 2020. Currently, he is a researcher in the Electronic Vision(s) group at Kirchhoff-Institute for Physics, Heidelberg University. His research interests are characterization and calibration of analog VSLI systems for emulation of biologically inspred neural networks, including hardware-specific model development and optimization.

Johannes Schemmel (M'08) received the Ph.D. degree in physics from Heidelberg University, Germany, in 1999. Currently, he is 'Akademischer Oberrat' in the Kirchhoff Institute of Physics, Heidelberg, where he is head of the ASIC lab and the Electronic Vision(s) group. His research interests are mixed-mode VLSI systems for information processing, especially the analog implementation of biologically realistic neural network models. He is the architect of the Spikey and BrainScaleS accelerated Neuromorphic hardware systems.