Stream Processing Architectures for Continuous ECG Monitoring Using Subsampling-Based Classifiers

Johnson Loh and Tobias Gemmeke, *Senior Member, IEEE*

***Abstract*— Monitoring of biomedical data, such as electrocardiogram (ECG) signals, requires accelerators, which can process data streams in a continuous manner. Especially, wearable monitoring systems require both ultralow power consumption and sufficiently complex deep neural network (DNN) classifiers to identify asymptomatic and critical health conditions, such as atrial fibrillation (AF). Such continuous data streams pose unique constraints on the processing pipeline for classification systems, which can be addressed in the design methodology of application-specific integrated circuits (ASICs). In this work, we identify specific constraints to define common operating conditions, which guide the design of ECG accelerators in an algorithm–hardware codesign methodology. In specific, we show that the input frame size and the number of classifications per time frame play a significant role for the computational complexity (CC) of the classifier, as well as the ECG accelerator executing the classifier in a continuous manner. As an example, the constraints are applied in a top-down algorithm–hardware codesign flow. Here, an ECG accelerator is designed starting from an AF classifier, while proposed constraints are considered in an early design stage to estimate costs for the hardware design. In the end, it is essential for future ECG accelerators to adhere to common constraints in the design process to handle increasingly complex DNN classifiers for continuous data streams with ultralow power targets.**

***Index Terms*— Application-specific integrated circuit (ASIC), design constraints, design space exploration, electrocardiogram (ECG) processing, streaming architecture, ultralow power.**

I. Introduction

CONTINUOUS monitoring of electrocardiogram (ECG) data streams enables early detection of anomalies. Regarding the classification of ECG data, the classification accuracy is increasing tremendously in recent years due to significant advancements of machine learning algorithms, such as deep neural networks (DNNs) [2], [3]. Not only do they achieve near-perfect classification accuracy [4] for popular ECG benchmarks, e.g., MIT-BIH arrhythmia database [5], but also they enable automated detection of atrial fibrillation (AF) [2], [3]. As AF is very common among the general population (25% of Europe and USA) and is associated with heart failure and strokes [6], this represents an important milestone in the discipline of automated ECG classification. The ultimate goal is to enable complex ECG anomaly detection, such as AF detection, practically in daily life scenarios through small form factor components integrated in Internet of Things (IoT) devices.

The acceleration of DNN models in application specific integrated circuits (ASICs) is one method for integration in IoT devices. The majority of works for efficient DNN accelerators deal with high-throughput designs for standard networks, e.g., AlexNet [7], targeting established image classification benchmarks [8] (see reviews in [9] and [10]). Considering the used benchmark metrics for those designs, i.e., throughput in terms of giga operations per second (GOPS) or energy efficiency in terms of GOPS/W [10], the emphasis in the design decisions revolves around increasing the amount of operations for as little power as possible. However, in the context of continuous data streams, the amount of operations to be processed in a given time period is simply limited by the input sampling rate and the number of classifications performed on the data stream. One explicit example is that the frame rate of a video, i.e., a sequence of images, is typically 60 frames/s (up to 120 frames/s) [11]. In consequence, an accelerator, e.g., [12], which can perform 279 image classifications per second, would idle more than half of the time waiting for frames to arrive. Furthermore, not every image in a camera stream contains new information for classification, e.g., nonmoving objects in a video, which leads to further questions regarding the necessary number of classifications per second. In the end, it is evident that the efficiency achieved by the accelerator is not evaluated for the continuous monitoring mode. Instead, it is intended for full utilization of the design, while assuming that the input data are readily available and independent of each other (in this work referred to as batch processing). However, accelerators, designed for monitoring of continuous data streams, require a paradigm shift, while still adhering to common application requirements, e.g., input frame size, similar to conventional DNN accelerator design.

In the scope of ECG data, cardiac monitoring systems realized in IoT devices pose hard constraints on accelerator design. For instance, the battery capacity of a common CR2032 cell in IoT devices allows only a current budget of sub-milliampere

Manuscript received 30 March 2023; revised 4 October 2023; accepted 28 October 2023. Date of publication 8 November 2023; date of current version 29 December 2023. This work was supported in part by the German Federal Ministry of Education and Research (Clusters4Future-NeuroSys) under Grant 03ZU1106CA; and in part by the German Federal Ministry for the Environment, Nature Conservation, Nuclear Safety and Consumer Protection (ReSCALE) under Grant 67KI32006A. An earlier version of this paper was presented at the 2022 IEEE Nordic Circuits and Systems Conference (NorCAS) [DOI: 10.1109/NorCAS57515.2022.9934591]. *(Corresponding author: Johnson Loh.)*

The authors are with the Chair of Integrated Digital Systems and Circuit Design, RWTH Aachen University, 52074 Aachen, Germany (e-mail: loh@ids.rwth-aachen.de; gemmeke@ids.rwth-aachen.de).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TVLSI.2023.3329360.

Digital Object Identifier 10.1109/TVLSI.2023.3329360

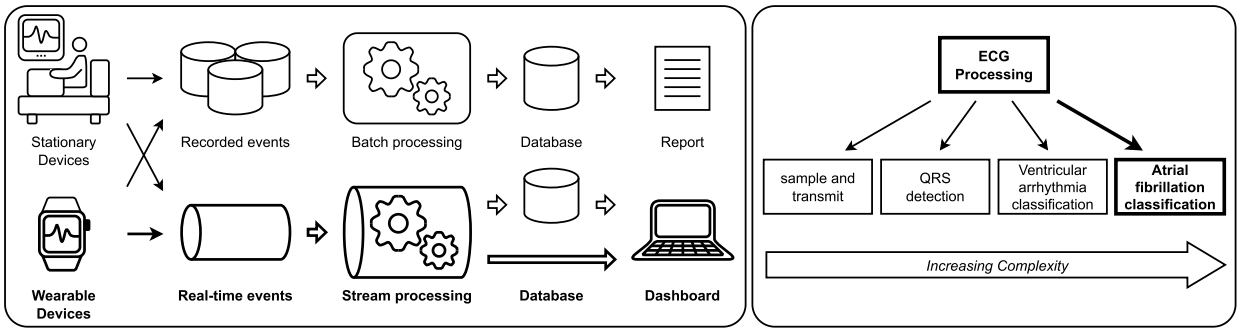


Fig. 1. Overview of ECG processing applications. Data are either recorded for later processing in batches or directly processed with each inserted sample (left). Within ECG processing, the application complexity varies from task to task (right), where complex tasks require high computational complexity (CC).

for a lifetime of more than one week [13]. In the extreme case, implantable cardiac monitoring systems [14] are expected to operate continuously for 2–3 years with a current budget of around 10 μA [15]. These tight power constraints severely limit what function the cardiac monitoring system is capable of realizing. In these cases, it is essential to properly limit the algorithm complexity, i.e., the classifier "size," the frequency, in which the algorithm is executed, and the hardware resources used to execute it. Recent literature targets the acceleration of low-complexity DNNs specifically for ECG data (see Fig. 1 and Section II). Although hardware platforms for ECG processing with machine learning methods have been investigated for nearly a decade [16], only recently do they incorporate AF classifiers with competitive classification accuracy [1], [17], [18]. The challenge in the design process involves the codesign of the algorithm, i.e., the DNN classifier capable of detecting AF, and the hardware, i.e., the ASIC realizing the specialized circuit, such that aforementioned power constraints can be met. In addition, as DNN complexity is growing exponentially [19], the algorithm–hardware codesign of cardiac monitoring systems needs to adapt this trend, while remaining feasible for long-term operation in IoT devices.

Therefore, the guiding question of this work is how complex ECG monitoring tasks, such as AF detection, can be realized in ultralow-power ECG accelerators processing continuous data streams using DNNs. Although algorithm–hardware codesigned ECG accelerators have been investigated extensively in previous literature, many do not consider complex AF classification tasks (see Section II for more details). Our conjecture is that ECG accelerators are able to achieve more complex classification tasks, such as AF detection, with low power by properly scaling the complexity of DNN classifiers to the new application requirements. Within the context of accelerators, which process data streams, the clear definition of a set of operating conditions, such as input frame size and output rate, is paramount to guide the algorithm–hardware codesign to achieve both low power and high classification quality. To the best of our knowledge, the discussion of these operating conditions has not been sufficiently addressed before and would provide tremendous design improvements for ECG monitoring systems. As an example, we showcase the design of an ECG accelerator guided by the presented operating conditions, further referred to as high-level design constraints. We specifically focus on the design of ASICs being the most suitable platform to reach the required ultralow-power operation, e.g., down to sub-microwatt, in this application context [20]. The subsampling-based classifier in [1] is used to exemplify the impact of the high-level design constraints on the design decisions made, e.g., variety of operation and number of processing element (PE) (see Section IV). We observe that well-chosen design decisions strictly adhering to those design constraints lead to an accelerator capable of classifying AF at state-of-the-art accuracy with sub-microwatt power, thus satisfying both high-quality classification and ultralow power requirements for long-term continuous ECG monitoring. In the end, state-of-the-art accelerators are discussed within the context of presented high-level design constraints and their impact on power, ECG classification quality and scalability of presented designs.

The main contributions of this work are summarized as follows.

1) High-level design constraints are presented for accelerators processing continuous data streams, especially ECG data, to guarantee comparable ECG accelerators operating under same conditions (see Section III-A).
2) The presented high-level design constraints are discussed in the context of a top-down algorithm–hardware codesign flow for accelerator design (see Section III-B1) and how the presented high-level design constraints can guide the design decisions in an early design stage (see Section III-B2 and case study in Section IV).
3) An analysis of the state-of-the-art based on the presented high-level design constraints (see Section V).

II. Scope and Related Works

Ultralow-power ECG accelerators target a range of tasks with varying degrees of complexity and processing settings (see Fig. 1). While early works focus on the sample-and-send scenario [21], [22] or simple QRS detection circuits [23], [24], [25], ECG classification started as a seemingly easy task to solve, where classification accuracy reaches up to 99% [26]. Although a variety of hardware architectures exist for ECG processing, such as programmable processor architectures [27], [28] or field-programmable gate arrays [29], [30], within the scope of this work, we focus on the design of ECG-specific digital signal processing back-end (DBE) using ASIC modules, further referred to as ECG accelerator. Nonprogrammable ASICs does not feature the flexibility of

programmable processor cores after fabrication, but can be customized to a more extreme degree to the application [10]. Therefore, it is essential that during the design phase of the ECG accelerator, all the necessary use cases from the application are considered in the design, i.e., through clearly defined design constraints (see Section III). For instance, programmable processor cores can realize, e.g., a variety of ECG delineation methods [31], by changing the software supported by the instructions of the processor cores. ECG accelerators, however, need to carefully select which method and components to use, as it is not reprogrammable during runtime.

For instance, Yin et al. [32] presented an ECG accelerator using multilayer perceptrons (MLPs) for biometric authentication and arrhythmia detection. The design heavily reduces the number of multiplications by exploiting symmetry in filter coefficients and sparsification of DNN weights through Lasso regression to finally achieve a power consumption of 1.06 μW at 0.55 V with a minimum equal error rate at about 1.7%. The specialized preprocessing pipeline, such as R-peak detection, outlier removals, and ECG beat alignment, allows low-power operation, while realizing multiple monitoring applications. Cherupally et al. [33] presented a similar ECG accelerator for biometric authentication, but instead used coarse-grain sparsity to reduce DNNs weights. The design achieves a power consumption of 75.41 μW at 1.2 V with a minimum equal error rate at about 1.0%. Although [32] and [33] target the same application, differences in the authentication algorithm result in key performance indicator (KPI) differences, i.e., different quality of authentication and power consumption.

Liu et al. [34] proposed a reconfigurable biomedical accelerator to classify biosignals, incl. ECG, using MLPs and convolutional neural networks (CNNs), where the number of layers, channels, kernels and kernel sizes, and strides can be adjusted. The design achieves a power consumption of 46.8 μW at 0.75 V classifying ECG signals between two classes with 2.25 μJ per classification. In contrast to [32] and [33], the classification is not performed on a beat-to-beat basis, but much faster than a heart beat, thus providing multiple labels per heart beat. This indicates that further reduction in average power is possible, if the frequency of classification is reduced.

In contrast, event-driven architectures perform computations on-demand. Wang et al. [26] propose a three-stage wake-up system, in which events from a level-crossing analog-to-digital converter (ADC) finally trigger a CNN for ECG classification. Zhao et al. [35] use the R-peak of a heart beat to trigger computations of an MLP in a custom multiply-accumulate (MAC) unit. Liu et al. [36] and Mao et al. [37] propose a spiking neural network to directly classifying inputs from a level-crossing ADC. While their power consumption is extraordinarily low, e.g., down to several hundred nanowatts, it is not clear how well the algorithm models scale from the beat classification task in the MIT-BIH benchmark [5] to more sophisticated classification problems, such as noisy AF classification [38], since state-of-the-art AF classifiers, such as in [39], exhibit model sizes several orders of magnitude larger than shown in these event-driven ECG accelerators.

Recent works show that state-of-the-art AF classifiers can be reduced in complexity such that low-power demands are still met. Parmar et al. [18] classify AF using integer Haar wavelets and an MLP in an end-to-end VLSI architecture to classify the MIT-BIH AF benchmark [40]. Jobst et al. and Loh et al. proposed a recurrent neural network [17], [41] and a CNN accelerator [1], [42] to solve the PhysioNet 2017 challenge [38], respectively. Especially for the noisy data, the number of operations used for classification, e.g., up to 730k MAC operations [1], exceeds previous ECG accelerator significantly. This can be accounted for the larger input frame used for classification, e.g., 18k samples [42], in comparison to previous ECG accelerator classifying on a beat-to-beat basis. As AF classification requires at least 30 s sections for episodes to be sufficiently diagnostic [6], the input frame is at least 30× larger than ECG arrhythmia, which are classified as individual beats[1] [5].

Given a large variety of previous ECG accelerator works, the research question we solve with this work is how the design requirements change, e.g., from conventional beat-based arrhythmia classification to more demanding problems such as AF classification (in our case study, the PhysioNet 2017 challenge [38]). Here, we reduce the problem to a set of operating conditions, which clearly define how much and how often data are s used to perform classifications on the data stream. In combination with the computational complexity (CC) of the DNN classifier, it is possible to estimate, e.g., how many operations are needed per time frame, thus using these estimations as constraints to guide the design process of the ECG accelerator. One explicit cost function is further described in Section III-B2, which is used for optimization.

III. Algorithm–Hardware Codesign Methodology for Streaming ECG Monitoring Accelerators

A. Problem Definition of Continuous ECG Monitoring

In a first step, we define the circumstances under which data streams are processed. We focus on the application of classification, which assigns labels to the input stream. As mentioned earlier, a uniform definition of frame size, output rate, etc. is key for early power estimations for hardware accelerator design.

An overview of design specifications is presented in Fig. 2. In general, state-of-the-art accelerators process the data stream segmented by heart beats [32], [33], [35], feature thresholding [26], [34], or in a full streaming fashion [1], [17], [18]. However, the basic principle can be summarized as follows. First, data are sampled uniformly using rate f_{in}, which is used as the input for processing. For ECG data, the sampling rate is usually in the region of several hundred hertz [5], [38], [43]. Considering, the possible clock frequency in CMOS technology, i.e., up to several gigahertz [44], the discrepancy of multiple orders of magnitude already indicates duty-cycling of the digital backend, when the number of operations for classification is low. Then, frames of size N_{frame} from the

[1]Assuming a regular heart rate at 60 beats per minute.

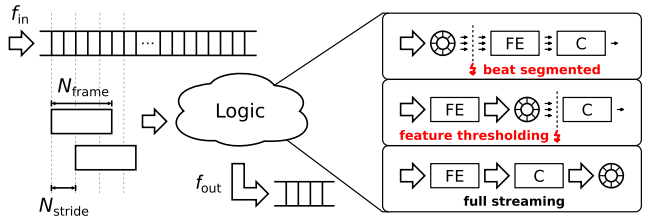


Fig. 2. General design specifications of continuous ECG accelerators. Streams (broad arrow) are processed into samples (small arrow) either directly or buffered. Processing of buffered data in feature extractors (FE) or classifiers (C) is typically triggered by signal features or when samples for one heart beat have been acquired.

resulting data stream are selected as input for processing. After N_{stride} samples, the next frame is processed. The system returns samples with a rate f_{out}, whereas output samples are optionally stored for buffering purposes. Note that implicitly we assume uniform sampling and processing rates. However, the concept is still applicable for mean rates over long periods of time, for instance, to model sparse irregular temporal events [36].

From the perspective of DNN classifiers, it is essential to note that N_{frame} directly affects how many operations and parameters the DNN needs. Event-driven sampling [26], [35], [36], [37] or preliminary feature extraction, such as wavelet transforms [1], [18], is used in previous works to reduce DNN input and, correspondingly, its size. Nevertheless, the DNN input and size still scale, correspondingly, when the effective frame for classification increases from beats to sequences greater than 30 seconds for AF classification [6].

Similarly, N_{stride} and f_{out} determine how often a classifier is executed on the data stream and how much data are reused for the classification. Even though it might seem trivial for beat-labeled data in ECG classification, i.e., one classification per beat [5], the boundaries get more complex, when episodes need to be defined for classification without clear markers in the data stream, e.g., AF classification [6]. Here, the latter case extends well into other data stream classification tasks such as seizure detection [45]. In these cases, the major question is: How often do I need to run the DNN model to detect relevant events? Previous works defined the output rate inconsistently, which, in the end, defines or is defined by the hardware architecture performing the digital processing. For AF classification, it can be determined by the filters with downsampling and pooling used in the processing pipeline resulting in 9.375 Hz for [17], 15.625 Hz for [18], and 4.32 Hz for [1].

To provide a basis for comparison across different N_{frame}, N_{stride}, and f_{out}, part of this work is to propose a normalized operating conditions inspired by the setup in batch processing (see Section I). In batch processing, data are available in distinct frames, e.g., images, which have a label assigned to them. In our case, the data stream is divided into adjacent frames, i.e., $N_{\text{stride,norm}} = N_{\text{frame}}$, and each input frame results in one prediction. The resulting output frequency is given by $f_{\text{out,norm}} = f_{\text{in}}/N_{\text{frame}}$. We introduce a factor $\alpha = N_{\text{stride,norm}}/N_{\text{stride}}$ to convert between the normalized output frequency $f_{\text{out,norm}}$ and an arbitrary f_{out} using the following equation:

$$f_{\text{out}} = \alpha \cdot f_{\text{out,norm}}. \tag{1}$$

Note that the introduced factor α also shows what fraction of the data stream is actually processed by the classifier. For instance, $\alpha > 1$ represents overlapping input frames, whereas $\alpha < 1$ represents sparse distinct frames. In the case of $\alpha = 1$, every sample of the data stream is used exactly once for further processing. This can be used as a good reference case to compare different sets of N_{frame}, N_{stride}, and f_{out}, as classifications are neither performed redundantly nor do they neglect data samples in the stream.

Nevertheless, within this work, we want to show how different sets of N_{frame}, N_{stride}, and f_{out} impact the design of an algorithm–hardware codesigned ECG accelerator. In the following, we implement use these properties as constraints, i.e., high-level design constraints, to guide the design process in different design stages.

B. Design Methodology

Within this work, we focus on the design of ASICs, in which the DNN classifier is mapped to nonprogrammable logic with minimal reconfigurability, e.g., as in [34]. Typically in ECG accelerators, designs are created in a top-down approach, whereas the algorithm is developed before it is mapped on dedicated hardware [1], [34], [35], [46]. In some cases, an algorithm is specifically designed for preexisting accelerators or hardware components [47], [48]. Nevertheless, the overall design process is separable into distinct levels, which focus on algorithm and hardware design separately.

1) General Design Methodology: Fig. 3 visualizes a general design methodology structure. We can see that the development of an algorithm–hardware codesigned architecture consists of four stages, of which the first two, i.e., algorithm design and fitting, concentrate on defining the algorithm to be executed and the latter two, i.e., hardware mapping and digital design flow, focus on the design of the accelerator. The design in all the abstraction levels needs to adhere to the operating conditions, as detailed in Section III-A, in the form of design constraints. In the following, we discuss how these constraints impact both the algorithm and hardware design space exploration.

First, the algorithm, i.e., the DNN classifier, is developed on a functional level. Typically, the initial prototype of the classifier mainly considers different DNN architectures and training configurations to achieve high quality of service (QoS). In this case, QoS represents the quality, which is achieved in the monitoring task. It can be quantified using a diverse set of metrics, such as F1 score [1], [17], [38], accuracy [18], [35], [36], [37], equal error rate [32], [33], and specificity [35], [37]. Nevertheless, the CC of the DNN classifier, typically quantified in terms of MAC operations [9], is dependent on the DNN input size and the complexity of the classification task. Some well-known examples from machine learning benchmarks are MNIST [49] and ImageNet [50]. Although both target the classification of images, best-in-class classifiers differ by multiple orders of magnitude in terms of model parameters, i.e., 1.5M [51] for MNIST and 2440M [52] for ImageNet. We observe similar trends in the domain of ECG classification, e.g., MLP with three MAC operations for premature ventricular contraction [46] and CNN with 730k MAC operations for AF [1].

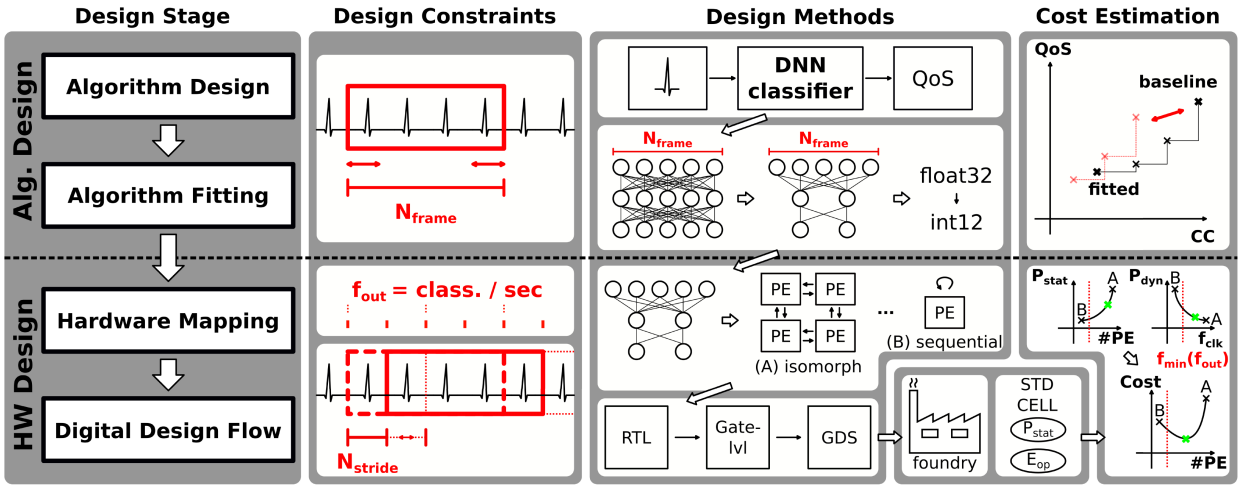


Fig. 3. Top-down design methodology for streaming ECG accelerators. Specifications from application and previous design stages guide the parameterization of used design methods. The result of each design stage is used for cost estimations of model quality in terms of, e.g., QoS, CC, and power.

In a next step, the algorithm is modified to achieve CC reductions typically by sacrificing QoS. This fitting step provides an initial step for a hardware realization by creating a functionally equivalent model as the baseline design, which can be inferred by a low-power ECG accelerator. Although methods such as pruning [53] and quantization [54] effectively reduce the DNN classifier's complexity, the reductions are still relative to the DNN classifier baseline, which in turn is limited by the constraints, i.e., N_{frame} and detectable classes.

From the perspective of ECG accelerator design, the fit DNN classifier needs to be executed in continuous time intervals on the ECG data stream. The mapping process has been studied extensively in previous works, where an algorithm is described using special notations, such as signal flow graphs [55], to be implemented in special purpose hardware. As the computation models [56] are manifold, ranging from isomorphic (or full-flat) mapping [see Fig. 3(a)] to fully sequential mapping [see Fig. 3(b)], the challenge lies in balancing the static and dynamic power of the ECG accelerator components. Furthermore, the mapping is restricted by the required classifications per time interval, i.e., f_{out}. Together with the complexity of the classifier, there is a hard constraint in terms of throughput, i.e., operations per second, the design needs to achieve.

In principle, a variety of mappings can be implemented in a hardware description language, such as Verilog or VHDL, and then simulated using the netlists after synthesis and place-and-route. However, we propose an approach to estimate the tradeoff in terms of static and dynamic power consumption for different hardware mappings.

2) Abstract Cost Modeling for Low-Power Designs: In the following, we extend the concept of the abstract cost model, as detailed in [1]. The basic concept is to integrate hardware domain knowledge, such as the static power P_{stat} and energy per operation E_{op} for individual logic gates, in an early design stage of development, especially to guide design choices made in the mapping process. The method itself is mainly inspired by cost modeling approaches to derive consumed energy of an arbitrary hardware platform based on the CC of an algorithm, e.g., using instruction counters [57], and methods to extract early information from a design, e.g., in data path placement [58]. The cost model is defined as follows:

$$\text{Cost}_E = \sum_i \kappa_{\text{op},i} \cdot E_{\text{op},i} + \kappa_{\text{leak},i} \cdot T_{\text{cyc}} \cdot P_{\text{leak},i}. \tag{2}$$

The cost Cost_E consists of scalable terms for dynamic energy $E_{\text{op},i}$ and static power $P_{\text{leak},i}$ normalized to a predefined time period, e.g., cycle time T_{cyc}. Conceptually, we scale known HW metrics from component to system level using information from the algorithm model, i.e., incorporated as $\kappa_{\text{op},i}$ and $\kappa_{\text{leak},i}$. Here, $\kappa_{\text{op},i}$ and $\kappa_{\text{leak},i}$ indicate scaling parameters for the energy of an operation and the leakage for the implemented components, respectively. Dependent on the context, (2) can be used to compare memory implementations using different devices or provide a means to optimize over Pareto-optimal design points of different degrees of PE concurrence. These two specific examples are shown in Section IV-B. In the case of ECG accelerators for continuous monitoring, the cost should directly relate to system power consumption for minimization.

Although this method enables cost estimations without an actual implementation on register transfer level (RTL) or gate level, it is imperative to distinguish $\text{Cost}_E / T_{\text{cyc}}$ from actual power dissipation of the CMOS circuit. While leakage may be composed of the sum of leakage of its individual components, dynamic power comprises short-circuit and switching power dissipation. These require knowledge about load capacitance or rise/fall time, which are not available in this design stage.

3) Exploration Vehicle—Subsampling-Based Classifier: To showcase our guided methodology, we consider a layered classifier architecture consisting of stages with computations and a (subsequent) subsampling component. We opted for this architecture for exploration due to several reasons.

First, it consists of convolutions and subsampling components, as found in short-time fourier transforms [39] and discrete wavelet transform (DWT) [1], [18], or neural network layers [1], [39]. Second, the raw input data are often not used "as is" for classification, but reduced to most prominent

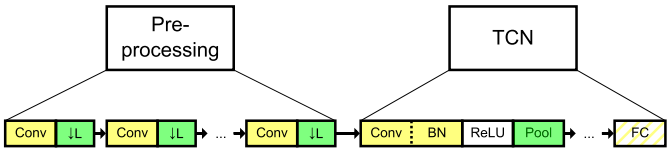


Fig. 4. Subsampling-based classifier introduced in [1]. Yellow and green blocks indicate convolutions and subsampling components, respectively.

features (e.g., R-R peak intervals [46], [59]) and/or events (e.g., level-crossing ADCs [36], delta-encoding [17]). Hence, the rate of features for classification is reduced, whereas it is interpretable as subsampling components. Although these types of event- and data-based methods are not as deterministic as fixed subsampling components, e.g., pooling layers, the mean rate of events in the raw data is predictable by a priori information (e.g., heart rate ranges, lead signal morphology).

An example for a subsampling-based classifier used in our case study (see Section IV) is shown in Fig. 4. It consists of DWT as preprocessing and a subsequent temporal convolutional network, i.e., a temporal 1D-CNN, as classifier. The basic building block for this algorithm is a convolution operation and a subsampling component (either by a constant factor or through pooling). As convolutions primarily consist of MAC operations, it is well-suited to showcase our exploration, since: 1) operations can be executed in PE arrays with various degrees of concurrence and 2) CC is well quantified by the number of MAC operations in the classifier [9].

IV. Case Study: ECG Accelerator

In this section, an ECG accelerator is designed from scratch to exemplify each design stage of a guided top-down algorithm–hardware codesign flow.

A. Algorithm Design Space Exploration

In a first step, the algorithm is designed to perform the task of ECG classification. In specific, we chose the classification of AF in single-lead ECG signals, i.e., the PhysionNet 2017 challenge [38], due to the following aspects. Single-lead ECG data provide a realistic assumption for the input in mobile ECG sensor devices, e.g., smartwatches or implantable cardiac monitors [14]. Here, the attachment of leads (and even the lead positions) is determined by convenience and device form factor, which is viable for daily human activity with minimal compromises. The problem class of AF is chosen due to its complexity and prominence in the general world population [6]. The former reason is crucial, since it eliminates trivial classifier solutions and represents the requirements of CC in a real-world scenario. High-end classifiers are able to solve this with high precision, but also high CC [38]. Using these algorithms as a baseline, the algorithm can be fit to hardware using a systematic approach. In our case, the fitting stage is performed through DNN training sweeps across DNN layer depth, width, number of bits for weights, and activation and channel pruning using PyTorch as a framework [60]. Our previous work details the design process of the algorithm fitting stage [42].

B. Hardware Design Space Exploration

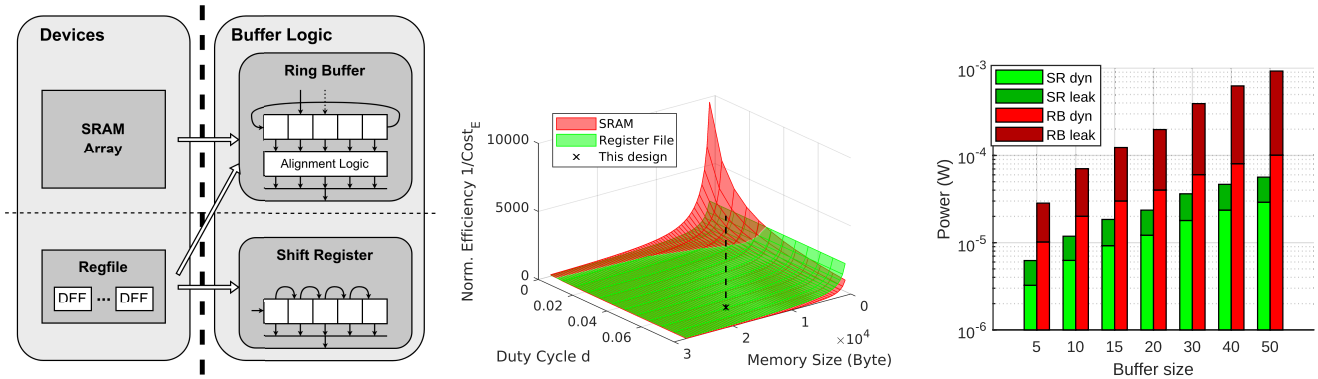
In the next step, a hardware architecture is designed to accelerate the fit model. In the mapping stage, the design of the system architecture and its individual components are important for the overall performance. Both are mainly constrained by the choice of design specifications N_{skip}, f_{in}, and f_{out}, as described in Section III-A, while their optimization is guided separately using the abstract cost model.

1) Activation Memory: In a first step, the buffering logic for intermediate results is investigated. Here, the buffer needs to receive data in a serial fashion and output data in a parallel fashion, i.e., serial-in parallel-out (SIPO) buffers [61]. Although this functionality can be implemented using multiple memory hierarchies, i.e., one global memory and a PE-level memory [17], [34], it is possible to use single-level hierarchies, i.e., no prebuffering at PE level [1], [35]. This mitigates memory transactions in between memory hierarchies. Even within single-level buffers, it is not obvious how to realize the SIPO functionality. On one hand, the memory can be implemented with either SRAMs or register files. On the other hand, the functionality is realizable with ring buffers or shift registers [see Fig. 5(a)].

In the following exploration, we focus on finding a design decision for single-level memory hierarchies using the abstract cost model. The constants for $P_{\text{leak,mem}}$ and $E_{\text{op,mem}}$ are acquired from provided datasheets of a D-Flip-Flop and an SRAM compiler, which depend on technology node and threshold voltage. Those are scaled with the number of used components $\kappa_{\text{leak,mem}}$ and number of memory transactions $\kappa_{\text{op,mem}}$ to acquire equivalent memory sizes and utilization. Note that depending on memory implementation, the scaling factors deviate significantly and need to be estimated for the specific use case and available devices. Fig. 5(b) shows how efficiency, i.e., inverse cost, relates to component specifications and utilization, i.e., duty cycle and memory size. Here, duty cycle d is zero, when no memory transactions are performed, while 1 indicates transactions in every cycle. It is evident that depending on the required memory size and memory usage either one provides a better cost.

An alternative method to the abstract cost model is the implementation and simulation of each design point. Fig. 5(c) shows the comparison between shift registers and ring buffers as activation memory. In this example, ring buffer solutions show higher power consumption in simulation. This is reasoned by the required overhead resulting from additional alignment logic for input activations. Obviously, simulative approaches are more precise in terms of power estimation, as it uses characterized components evaluated by an EDA tool chain. However, the main downsides are twofold: the engineering time required to implement the components and the time required for simulation and testing. This method is less suitable, the more complex the component is and the larger the design space gets.

2) Degree of Concurrence: Regarding the PEs, previous ECG accelerator research mainly focused on the incorporation of novel features, such as event-driven sampling [35] or delta-GRU acceleration [17], instead of the dataflow and utilization of PEs, as in the image processing domain [9]. With the trend of rising CC for more complex ECG processing tasks, both reuse and concurrence are important in the mapping process.



(a) (b) (c)

Fig. 5. Activation memory exploration by defining. (a) Design space dimensions and cost evaluation through, (b) abstract cost model or (c) post-synthesis simulations performed in [1].

For the special purpose hardware, it is common in the state-of-the-art ECG accelerators to separate the components for preprocessing and classification stages. However, the degree of parallel PEs is diverse even considering only for the classification stage. It can range from single MAC units [17], [35] to flat mapping of the whole neural network [36], [42], [47], [48], where hybrid mappings on vector MAC units also exist [1], [34]. Note that the computations in the classifier are highly uniform for either conventional DNNs or spiking neural networks [36], [47], [48]. Nevertheless, the main question is how many PEs are required to perform the whole algorithm.

While the extreme cases, i.e., sequential or isomorphic mapping [56], do minimize static power or maximize throughput respectively, the overall system power needs to be minimized for ultralow-power operation. To find an informed solution to the problem of concurrence, the abstract cost model is used to analyze the cost required for each design point. However, in a first step it needs to be clarified how different dataflow methodologies[2] are applicable in our context [9]. For stream processing, we find that a weight and input stationary approach are less suitable, since the input feature maps are not completely available at one time step for a prediction, such that they can be reused. In contrast, an output sample can stay stationary for temporal multiplexing over input channels at one time step. In case of an output stationary dataflow, the number of cycles required for one output in each layer can be estimated using the PE size and loop tiling.

In our exploration, we use the number of PEs N_{pe} and the minimum operating frequency $f_{\text{sys,min}}$ to scale leakage and dynamic energy, since $P_{\text{leak}} \propto N_{\text{pe}}$ and $P_{\text{dyn}} \propto f_{\text{sys,min}}$. For the former, PEs are inserted for each activation buffer per parallelized layer L starting from the initial DWT layers, whereas $N_{\text{pe}} = \sum_i^L k_i \cdot C_i$ with k_i and C_i being the kernel size and number of output channels of layer i, respectively. For the latter, the algorithm summarized in Table I is used to determine the minimum frequency, i.e., $f_{\text{sys,min}} = f_{\text{in}} \cdot \prod_{i=L}^{L_{\max}} d_i$ with d_i being the subsampling factor of layer i. To model the relationship between leakage and energy per operation of a technology node, a characterized template cell, e.g., DFF, from the standard library is used for P_{leak} and E_{op}. Obviously, the main drawback is that the used template cell does not fully represent the diversity of used logic in the circuit. However, it serves as a first-order approximation and can be improved by a more fine-grained split of cost terms.

[2]In this context, dataflow refers to the term used in DNN accelerator design as in [9].

TABLE I
ALGORITHMIC MODEL OF USED TCN

Layer[1]	Kernel Size k	# Output Channels C	Subsampl. Factor d
DWT L1	4	1	2
DWT L2	4	1	2
DWT L3	4	1	2
DWT L4	4	2	2
TCN L1	5	10	3
TCN L2	5	13	3
TCN L3	5	20	3
TCN L4	5	63	3
TCN FC	11	4	-

[1] Each layer Lx consists of one convolution (incl. rectified linear unit for TCN layers) and one subsampling block.

Fig. 6 shows the results of the exploration. It is evident that all the design points are Pareto-optimal considering N_{pe} and $f_{\text{sys,min}}$, only. However, the cost shows an optimum for the case, in which the first four DWT layers have independent PEs and a vector MAC unit is used for the DNN part. This is inline with the fact that it is beneficial to have dedicated components for the preprocessing part, which is done intuitively by previous works. This is reasoned by small PEs required to implement feature extraction in contrast to high CC for classification. Nevertheless, as the complexity of features grows, the balance of CC might shift and the feasibility of separate components needs to be reevaluated.

3) Processing Scheme in Streaming ECG Accelerators: Given previous exploration, the overall system architecture is realized, where the preprocessing is separated from the classifier logic. While the DWT components are cascaded transversal filters with a subsampled output, the execution of the classifier on a single vector PE is more complex in terms of dataflow.

As briefly mentioned in Section IV-B2, the temporal nature of input feature maps restricts the reuse of weights and input

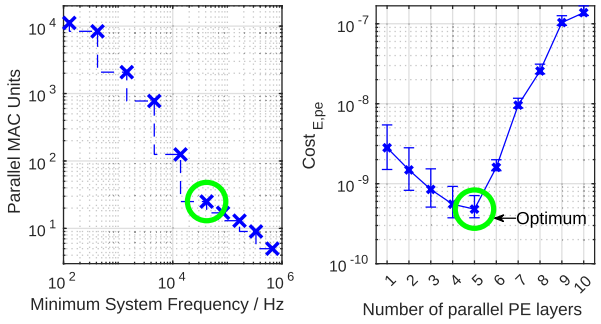


Fig. 6. Pareto-optimal front (left) and estimated cost (right) for parallelized layers. Variations in $P_{\text{leak,cell}}$ and $E_{\text{cyc,cell}}$ for rise/fall transitions are resulting in cost uncertainties.

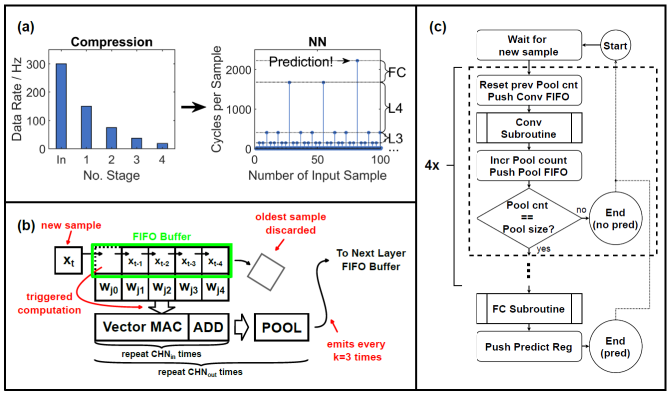


Fig. 7. (a) Data rate reduction and number of sparsely triggered computation cycles in NN accelerator. (b) Triggered data movement in SIPO register with inherent alignment to weight kernel for convolution subroutine. (c) Flowchart of control logic function realizing data-driven processing pattern for each new input sample.

feature maps at a specific point of time. Hence, a data-driven processing scheme is used with an output stationary dataflow. Here, the different layers of the CNN are executed conditionally as function of the subsampling factors in its architecture (see Table I). Fig. 7(a) shows that the first CNN layer is calculated for every input sample, whereas the second CNN layer is triggered for every third output sample of the first CNN layer and so on. Only the samples within a filter kernel need to be stored, whereas spatial reuse of the input is achieved by shifting the samples within the SIPO register [see Fig. 7(b)]. In between new input samples, however, the vector MAC unit performs subconvolutions per input channel and accumulates for one output channel. Multiple output channels are calculated in a time-multiplexed manner. In the end, the processing scheme is summarized in Fig. 7(c).

It is notable that in contrast to the proposed processing scheme, it is possible to buffer all the samples of an input frame for conventional frame-based processing, as in [9]. As it is feasible for small N_{frame}, such as a heart beat [35], [61], the overhead for buffering large ECG traces, e.g., 60 s sections as required for AF [6], increases the static power of memory units significantly. This impact remains to be evaluated in future work.

Fig. 8 shows the block diagram for the neural network engine. It is evident that the accelerator mainly consists of buffering logic for intermediate activations and SRAM for CNN weights and biases. The pooling units serve as buffering of the maximum value but also trigger the computation of the

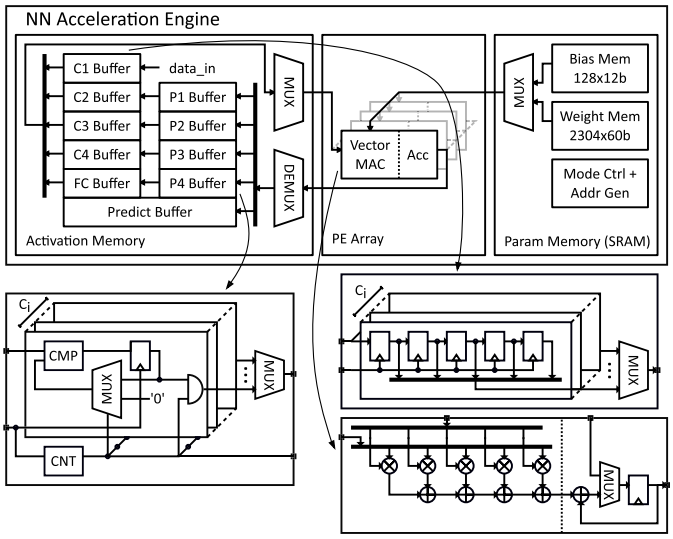


Fig. 8. Hardware architecture of the neural network acceleration engine with detailed depiction of the PE and the convolution/pooling buffers.

next layer. The vector MAC unit is purely combinatorial with an accumulation register for the output sample.

While the above-mentioned steps comprise the HW mapping stage of the design methodology in Fig. 3, the next design stage simply follows best practice. Here, the buffers are clock-gated and only enabled for one cycle, when triggered by the pooling layer. Assuming a low system frequency, ultralow leakage cells and SRAM are used to implement the whole design. Logic described above is implemented in Verilog, whereas channel sizes, the number of vector MAC units, the number of CNN layers, and the word-length of fixed-point numbers are parameterized for flexibility. Synthesis and Place-and-Route are performed using commercial EDA tools from Cadence, such as Genus and Innovus, following the reference flow provided by the foundry, i.e., in our case Globalfoundries. After signoff, the design is exported in the graphic design system (GDS) format for fabrication in the foundry.

4) Simulation and Measurement Results: The resulting design is validated in postlayout simulations under real-time conditions. The following experiments are performed for 60-s ECG traces sampled at 300 Hz. Simulations are performed with back-annotated postlayout netlists under nominal conditions (TT process cornerat 0.8 V at room temperature). The intermediate activations and predictions showed identical numerical values to the software reference from the algorithm exploration phase.

The fabricated design is used for power measurements. Here, it is possible to sweep the supply voltage down to 0.5 V for further power savings (see Fig. 9). In the end, the design in this case study consumes down to 525 nW, while the total power is scaling approx. with $P \propto V_{\text{dd}}^2$ in this case (see dashed line in Fig. 9). This trend is later used to remove the impact of voltage scaling, when comparing the state-of-the-art.

V. Discussion of the State-of-the-Art

As indicated in the introduction, the application task for ECG monitoring varies strongly among previous works. This includes especially differences in the benchmarks with varying operating conditions, such as different numbers of classes

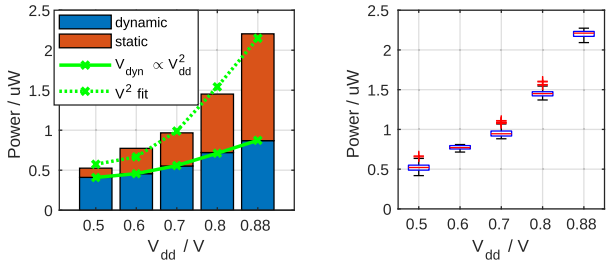


Fig. 9. Measurement results of design operating at $f_{sys} = 500$ kHz (mean: left, distribution: right).

for classification (e.g., two [18], [32], four [1], [17], [26], [36], five [30], [34], [35], [47], [48]) or different ECG datasets (e.g., MIT-BIH arrhythmia [5], MIT-BIH AF dataset [43], various PhysioNet challenges [38], [62], [63], or even in-house datasets [41]). In consequence, the interpretation of the presented KPIs should be seen in relation to the experimental setup. As mentioned in Section II, our focus is on DNN classifiers, which show the capability to scale to complex classification tasks, i.e., in our case the classification of AF (PhysioNet 2017 challenge) [38]. In the following, we discuss how the presented design constraints impact design choices in previous works. Note that we select representative works for discussion to show trends in literature regarding the constraints for continuous data stream processing in ECG accelerator designs.

As mentioned before, the majority of ECG accelerators focus on small input frames in the region of single or few beats. Yin et al. [32] and Cherupally et al. [33] limit the input for their MLP to single beat sample windows from different finite-impulse response (FIR) filters aligned around the R-peak or Q-wave. Here, the number of input is limited to max. 160 samples and, thus, limiting the CC of the baseline classifier to a few hundred neurons per MLP. In the compressed model of [33], it even requires only 32 MAC operations per neuron. Since detections are performed for every heartbeat, the number of MAC operations per second is limited in the region of kOPS and, thus, resulting in accelerator designs in the region of microwatt.

Zhao et al. [35] and Liu et al. [36] use level-crossing ADCs to generate sparse input feeding into a small classifier, i.e., a fully connected DNN and SNN, respectively. Very low power (down to 100 nW range) is achieved through very small DNN architectures (53 [35] or 68 neurons [36]), which can be inferred with low costs regardless of underlying hardware accelerator. Here, the algorithm design is geared toward the event-based input, allowing an increased potential in the algorithm fitting stage. Although hardware-specific components do contribute to final power reductions, a major prerequisite is the low CC algorithm for inference. This is achieved by a condensed representation of the input. Similarly, Liu et al. [34] define simple features, e.g., zero-crossings and line length, to both trigger the classification and use as features for it. Janveja et al. [61] hand-tune the number of samples required according to morphological features and, thus, decreasing the input frame size significantly. Here again, the tight selection of input features enables the training of small DNN architectures for inference, which ultimately results in overall lower power.

The problem of AF classification, however, is not discussed in earlier DNN accelerator works and only specifically targeted recently. Sadasivuni et al. [64] used the clinical MIT-BIH AF benchmark [40] to classify between AF and normal ECG signals. Here, 63 ECG time-domain features, e.g., R-peaks and QRS-complexes, are used for classification in an MLP with 26 neurons. Similarly, Parmar et al. [18] used an MLP to classify extracted features. A data frame of only 500 samples is downsampled by a preliminary DWT stage and then fed into an MLP with 92 neurons. Although both the works report high classification accuracy, i.e., >90%, with a low power consumption in the region of microwatt, the low complexity classifiers are not tested against noisy AF from commercial devices, hence validated against practical benchmarks scenarios for IoT devices.

In contrast, Jobst et al. [17] and Loh and Gemmeke [1] use the CinC 2017 challenge as a benchmark to develop the DNN classifier as a baseline for algorithm–hardware co-optimization. In these cases, it is critical that the maximum QoS of the application, i.e., AF classification of noisy ECG data for a commercial ECG recording devices, is only achievable by large complex classifiers [38], whereas the input frame used for classification is either huge (approx. 17k samples [1]) or is fully represented in the neural networks' memory units [17]. The miniaturization of the classifier algorithm in a top-down design approach (see Fig. 3) decreases both CC and QoS, such that an ultralow-power ECG accelerator is designed, while performing the same task as large SW models, e.g., million parameter DNNs [39]. Note that both the works follow the presented design methodology to systematically achieve a tradeoff between CC and QoS in the algorithm level, in which the resulting fit model serves as a compact baseline for hardware mapping. Although they do not feature any special hardware features, e.g., asynchronous design or level-crossing ADCs, power consumption in the hundreds of nanowatt is achieved simply by instantiating well-established hardware modules, such as FIR filter components, in fixed-point arithmetic combined with voltage scaling.

VI. Conclusion

This work proposed identified key constraints to guide the design process of ECG accelerators for data stream processing. In the context of growing DNN complexity for more complex ECG classification tasks, proper co-optimization of algorithm and hardware needs common operating conditions to systematically reach ultralow power requirements, while keeping high classification quality. First, the operating conditions for continuous stream processing are formalized in terms of input–output frequency, frame size, and overlap. These conditions are used as application constraints in early stages of algorithm–hardware codesign of ECG accelerators. In a top-down algorithm–hardware codesign methodology, the constraints are shown to influence the design of DNN classifier in terms of achievable CC and QoS. Furthermore, the abstract cost model is elaborated in detail, which uses component-level hardware metrics, i.e., leakage and energy

per operation, to scale system-level costs. These costs are used to guide design decisions ranging from component selection to hardware mapping. The concept is elaborated using the example of an AF classifier, which is capable of classifying ECG data from commercial recording devices with the state-of-the-art QoS and ultralow power. Finally, a comparison to the state-of-the-art reveals the breadth of experimental setups and benchmark settings across ECG accelerators. The ultralow-power operation can be traced back to a multitude of possible reasons, e.g., sparse activation, small input frames, and low complexity classifier. In the end, the evolving nature of DNN classifier complexity on the algorithm-level enables more complex ECG classification tasks, but also needs to be supported by ultralow-power ECG accelerators. The presented work strives to met the scaling demands on increasing classifier complexity for algorithm–hardware codesign ECG accelerators by providing normalized specifications for continuous ECG monitoring.

REFERENCES

[1] J. Loh and T. Gemmeke, "Dataflow optimizations in a sub-μW data-driven TCN accelerator for continuous ECG monitoring," in *Proc. IEEE Nordic Circuits Syst. Conf. (NorCAS)*, Oct. 2022, pp. 1–7.

[2] Z. Ebrahimi, M. Loni, M. Daneshtalab, and A. Gharehbaghi, "A review on deep learning methods for ECG arrhythmia classification," *Expert Syst. Appl., X*, vol. 7, Sep. 2020, Art. no. 100033.

[3] S. Somani et al., "Deep learning and the electrocardiogram: Review of the current state-of-the-art," *EP Europace*, vol. 23, no. 8, pp. 1179–1191, Feb. 2021.

[4] S. Mousavi and F. Afghah, "Inter- and intra-patient ECG heartbeat classification for arrhythmia detection: A sequence to sequence deep learning approach," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 1308–1312.

[5] G. B. Moody and R. G. Mark, "The impact of the MIT-BIH arrhythmia database," *IEEE Eng. Med. Biol. Mag.*, vol. 20, no. 3, pp. 45–50, May/Jun. 2001.

[6] P. Kirchhof et al., "2016 ESC guidelines for the management of atrial fibrillation developed in collaboration with EACTS," *Kardiologia Polska*, vol. 74, no. 12, pp. 1359–1469, Dec. 2016.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. 25th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, vol. 1. Red Hook, NY, USA: Curran Associates, 2012, pp. 1097–1105.

[8] O. Russakovsky et al., "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Apr. 2015.

[9] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, "Efficient processing of deep neural networks: A tutorial and survey," *Proc. IEEE*, vol. 105, no. 12, pp. 2295–2329, Dec. 2017.

[10] P. Dhilleswararao, S. Boppu, M. S. Manikandan, and L. R. Cenkeramaddi, "Efficient hardware architectures for accelerating deep neural networks: Survey," *IEEE Access*, vol. 10, pp. 131788–131828, 2022.

[11] A. Mackin, F. Zhang, and D. R. Bull, "A study of high frame rate video formats," *IEEE Trans. Multimedia*, vol. 21, no. 6, pp. 1499–1512, Jun. 2019.

[12] Y.-H. Chen, T.-J. Yang, J. Emer, and V. Sze, "Eyeriss v2: A flexible accelerator for emerging deep neural networks on mobile devices," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 9, no. 2, pp. 292–308, Jun. 2019.

[13] D. Griffith, "Toward zero: Power consumption trends in low data rate wireless connectivity," *IEEE Solid State Circuits Mag.*, vol. 14, no. 4, pp. 51–60, Fall. 2022.

[14] C. Sandesara, R. Gopinathannair, and B. Olshansky, "Implantable cardiac monitors: Evolution through disruption," *J. Innov. Cardiac Rhythm Manage.*, vol. 8, no. 9, pp. 2824–2834, Sep. 2017.

[15] Y. Yin et al., "A 2.63 μW ECG processor with adaptive arrhythmia detection and data compression for implantable cardiac monitoring device," *IEEE Trans. Biomed. Circuits Syst.*, vol. 15, no. 4, pp. 777–790, Aug. 2021.

[16] Y. Wei et al., "A review of algorithm-hardware design for AI-based biomedical applications," *IEEE Trans. Biomed. Circuits Syst.*, vol. 14, no. 2, pp. 145–163, Apr. 2020.

[17] M. Jobst et al., "ZEN: A flexible energy-efficient hardware classifier exploiting temporal sparsity in ECG data," in *Proc. IEEE 4th Int. Conf. Artif. Intell. Circuits Syst. (AICAS)*, Jun. 2022, pp. 214–217.

[18] R. Parmar, M. Janveja, J. Pidanic, and G. Trivedi, "Design of DNN-based low-power VLSI architecture to classify atrial fibrillation for wearable devices," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 31, no. 3, pp. 320–330, Mar. 2023.

[19] X. Xu et al., "Scaling for edge inference of deep neural networks," *Nature Electron.*, vol. 1, no. 4, pp. 216–222, Apr. 2018.

[20] K. Guo et al. *Neural Network Accelerator Comparison*. Accessed: Mar. 27, 2023. [Online]. Available: https://nicsefc.ee.tsinghua.edu.cn/projects/neural-network-accelerator/

[21] T. H. Teo et al., "A 700-μW wireless sensor node SoC for continuous real-time health monitoring," *IEEE J. Solid-State Circuits*, vol. 45, no. 11, pp. 2292–2299, Nov. 2010.

[22] X. Zhang, Z. Zhang, Y. Li, C. Liu, Y. X. Guo, and Y. Lian, "A 2.89 μW dry-electrode enabled clockless wireless ECG SoC for wearable applications," *IEEE J. Solid-State Circuits*, vol. 51, no. 10, pp. 2287–2298, Oct. 2016.

[23] R. F. Yazicioglu et al., "A 30 μW analog signal processor ASIC for portable biopotential signal monitoring," *IEEE J. Solid-State Circuits*, vol. 46, no. 1, pp. 209–223, Jan. 2011.

[24] R. A. Abdallah and N. R. Shanbhag, "An energy-efficient ECG processor in 45-nm CMOS using statistical error compensation," *IEEE J. Solid-State Circuits*, vol. 48, no. 11, pp. 2882–2893, Nov. 2013.

[25] X. Liu et al., "A 457 nW near-threshold cognitive multi-functional ECG processor for long-term cardiac monitoring," *IEEE J. Solid-State Circuits*, vol. 49, no. 11, pp. 2422–2434, Nov. 2014.

[26] Z. Wang et al., "A 148-nW reconfigurable event-driven intelligent wake-up system for AIoT nodes using an asynchronous pulse-based feature extractor and a convolutional neural network," *IEEE J. Solid-State Circuits*, vol. 56, no. 11, pp. 3274–3288, Nov. 2021.

[27] H. Kim et al., "A configurable and low-power mixed signal SoC for portable ECG monitoring applications," *IEEE Trans. Biomed. Circuits Syst.*, vol. 8, no. 2, pp. 257–267, Apr. 2014.

[28] G. Sivapalan, K. K. Nundy, S. Dev, B. Cardiff, and D. John, "ANNet: A lightweight neural network for ECG anomaly detection in IoT edge sensors," *IEEE Trans. Biomed. Circuits Syst.*, vol. 16, no. 1, pp. 24–35, Feb. 2022.

[29] H. Chu et al., "A neuromorphic processing system with spike-driven SNN processor for wearable ECG classification," *IEEE Trans. Biomed. Circuits Syst.*, vol. 16, no. 4, pp. 511–523, Aug. 2022.

[30] J. Lu, D. Liu, X. Cheng, L. Wei, A. Hu, and X. Zou, "An efficient unstructured sparse convolutional neural network accelerator for wearable ECG classification device," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 69, no. 11, pp. 4572–4582, Nov. 2022.

[31] F. Rincón, J. Recas, N. Khaled, and D. Atienza, "Development and evaluation of multilead wavelet-based ECG delineation algorithms for embedded wireless sensor nodes," *IEEE Trans. Inf. Technol. Biomed.*, vol. 15, no. 6, pp. 854–863, Nov. 2011.

[32] S. Yin et al., "A 1.06-μW smart ECG processor in 65-nm CMOS for real-time biometric authentication and personal cardiac monitoring," *IEEE J. Solid-State Circuits*, vol. 54, no. 8, pp. 2316–2326, Aug. 2019.

[33] S. K. Cherupally et al., "ECG authentication hardware design with low-power signal processing and neural network optimization with low precision and structured compression," *IEEE Trans. Biomed. Circuits Syst.*, vol. 14, no. 2, pp. 198–208, Apr. 2020.

[34] J. Liu et al., "4.5 BioAIP: A reconfigurable biomedical AI processor with adaptive learning for versatile intelligent health monitoring," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, vol. 64, Feb. 2021, pp. 62–64.

[35] Y. Zhao, Z. Shang, and Y. Lian, "A 13.34 μW event-driven patient-specific ANN cardiac arrhythmia classifier for wearable ECG sensors," *IEEE Trans. Biomed. Circuits Syst.*, vol. 14, no. 2, pp. 186–197, Apr. 2020.

[36] Y. Liu et al., "An 82 nW 0.53 pj/SOP clock-free spiking neural network with 40 μs latency for AIoT wake-up functions using ultimate-event-driven bionic architecture and computing-in-memory technique," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2022, pp. 372–374.

[37] R. Mao et al., "An ultra-energy-efficient and high accuracy ECG classification processor with SNN inference assisted by on-chip ANN learning," *IEEE Trans. Biomed. Circuits Syst.*, vol. 16, no. 5, pp. 832–841, Oct. 2022.

[38] G. Clifford et al., "AF classification from a short single lead ECG recording: The PhysioNet/computing in cardiology challenge 2017," in *Proc. Comput. Cardiol. Conf. (CinC)*, Sep. 2017, pp. 1–4.

[39] M. Zihlmann, D. Perekrestenko, and M. Tschannen, "Convolutional recurrent neural networks for electrocardiogram classification," in *Proc. Comput. Cardiol. (CinC)*, Sep. 2017, pp. 1–4.

[40] G. B. Moody and R. G. Mark, "A new method for detecting atrial fibrillation using RR intervals," *Comput. Cardiol.*, vol. 10, pp. 227–230, Jan. 1983.

[41] M. Jobst et al., "Event-based neural network for ECG classification with delta encoding and early stopping," in *Proc. 6th Int. Conf. Event-Based Control, Commun., Signal Process. (EBCCSP)*, Sep. 2020, pp. 1–4.

[42] J. Loh, J. Wen, and T. Gemmeke, "Low-cost DNN hardware accelerator for wearable, high-quality cardiac arrythmia detection," in *Proc. IEEE 31st Int. Conf. Appl.-Specific Syst., Archit. Processors (ASAP)*, Jul. 2020, pp. 213–216.

[43] A. L. Goldberger et al., "PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. E215–E220, Jun. 2000.

[44] K. Gavaskar, R. Dhivya, and R. D. Dayana, "Low power CMOS design of phase locked loop for fastest frequency acquisition at various nanometer technologies," *Wireless Pers. Commun.*, vol. 125, no. 3, pp. 2239–2251, Mar. 2022.

[45] W. Zeng et al., "Epileptic seizure detection with deep EEG features by convolutional neural network and shallow classifiers," *Frontiers Neurosci.*, vol. 17, May 2023, Art. no. 1145526.

[46] Z. Chen, H. Xu, J. Luo, T. Zhu, and J. Meng, "Low-power perceptron model based ECG processor for premature ventricular contraction detection," *Microprocessors Microsyst.*, vol. 59, pp. 29–36, Jun. 2018.

[47] F. C. Bauer, D. R. Muir, and G. Indiveri, "Real-time ultra-low power ECG anomaly detection using an event-driven neuromorphic processor," *IEEE Trans. Biomed. Circuits Syst.*, vol. 13, no. 6, pp. 1575–1582, Dec. 2019.

[48] K. Buettner and A. D. George, "Heartbeat classification with spiking neural networks on the Loihi neuromorphic processor," in *Proc. IEEE Comput. Soc. Annu. Symp. VLSI (ISVLSI)*, Jul. 2021, pp. 138–143.

[49] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[50] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[51] A. Byerly, T. Kalganova, and I. Dear, "No routing needed between capsules," *Neurocomputing*, vol. 463, pp. 545–553, Nov. 2021.

[52] X. Chen et al., "Symbolic discovery of optimization algorithms," 2023, *arXiv:2302.06675*.

[53] S. A. Janowsky, "Pruning versus clipping in neural networks," *Phys. Rev. A, Gen. Phys.*, vol. 39, no. 12, pp. 6600–6603, Jun. 1989.

[54] S. Gupta et al., "Deep learning with limited numerical precision," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2015, pp. 1737–1746.

[55] R. E. Crochiere and A. V. Oppenheim, "Analysis of linear digital networks," *Proc. IEEE*, vol. 63, no. 4, pp. 581–595, Apr. 1975.

[56] E. A. Lee and D. G. Messerschmitt, "Synchronous data flow," *Proc. IEEE*, vol. 75, no. 9, pp. 1235–1245, Sep. 1987.

[57] E. García-Martín, C. F. Rodrigues, G. Riley, and H. Grahn, "Estimation of energy consumption in machine learning," *J. Parallel Distrib. Comput.*, vol. 134, pp. 75–88, Dec. 2019.

[58] T. T. Ye and G. D. Micheli, "Data path placement with regularity," in *IEEE/ACM Int. Conf. Comput. Aided Design. (ICCAD), IEEE/ACM Dig. Tech. Papers*, Nov. 2000, pp. 264–270.

[59] J. Pan and W. J. Tompkins, "A real-time QRS detection algorithm," *IEEE Trans. Biomed. Eng.*, vol. BME-32, no. 3, pp. 230–236, Mar. 1985.

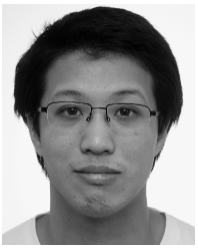
[60] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, H. Wallach et al., Eds. Red Hook, NY, USA: Curran Associates, 2019, pp. 1–12.

[61] M. Janveja, M. Tantuway, K. Chaudhari, and G. Trivedi, "Design of low power VLSI architecture for classification of arrhythmic beats using DNN for wearable device applications," in *Proc. IEEE Nordic Circuits Syst. Conf. (NorCAS)*, Oct. 2021, pp. 1–6.

[62] E. A. Perez Alday et al., "Classification of 12-lead ECGs: The PhysioNet/computing in cardiology challenge 2020," *Physiol. Meas.*, vol. 41, no. 12, Dec. 2020, Art. no. 124003.

[63] M. A. Reyna et al., "Issues in the automated classification of multilead ECGs using heterogeneous labels and populations," *Physiol. Meas.*, vol. 43, no. 8, Aug. 2022, Art. no. 084001.

[64] S. Sadasivuni, S. P. Bhanushali, S. S. Singamsetti, I. Banerjee, and A. Sanyal, "Multi-task learning mixed-signal classifier for in-situ detection of atrial fibrillation and sepsis," in *Proc. IEEE Biomed. Circuits Syst. Conf. (BioCAS)*, Oct. 2021, pp. 1–4.



Johnson Loh received the B.S. and M.S. degrees in electrical engineering from RWTH Aachen University, Aachen, Germany, in 2015 and 2018, respectively, where he is currently working toward the Ph.D. degree.

His current research interests include ultralow-power neural network and neuromorphic accelerators for biomedical applications.

Tobias Gemmeke (Senior Member, IEEE) received the M.S. and Ph.D. degrees in electrical engineering from RWTH Aachen University, Aachen, Germany, in 1998 and 2006, respectively.

In 2004, he transitioned to IBM's Research and Development Organization, Böblingen, Germany, targeting high-performance processors. In 2007, he joined former startup Aquantia Corporation (now Marvell) conceiving energy-efficient PHY solutions for the 10GBase-T over copper standard. In 2011, he became the Technical Lead of the Digital Design Team, Holst Centre/IMEC, Eindhoven, The Netherlands, focusing on ultralow-power design of wearable sensor nodes. Since 2017, he has been a Full Professor with the Chair of Integrated Digital System and Circuit Design (IDS), RWTH Aachen University. His research interests include the design of digital systems considering all entry levels from algorithmic optimization down to the physically oriented design.