Lossless Sparse Temporal Coding for SNN-based Classification of Time-Continuous Signals

Johnson Loh and Tobias Gemmeke
Chair of Integrated Digital Systems and Circuit Design, RWTH Aachen University
Aachen, Germany
Email: {loh,gemmeke}@ids.rwth-aachen.de

***Abstract*—Ultra-low power classification systems using spiking neural networks (SNN) promise efficient processing for mobile devices. Temporal coding represents activations in an artificial neural network (ANN) as binary signaling events in time, thereby minimizing circuit activity. Discrepancies in numeric results are inherent to common conversion schemes, as the atomic computing unit, i.e. the neuron, performs algorithmically different operations and, thus, potentially degrading SNN's quality of service (QoS). In this work, a lossless conversion method is derived in a top-down design approach for continuous time signals using electrocardiogram (ECG) classification as an example. As a result, the converted SNN achieves identical results compared to its fixed-point ANN reference. The computations, implied by proposed method, result in a novel hybrid neuron model located in between the integrate-and-fire (IF) and conventional ANN neuron, which numerical result is equivalent to the latter. Additionally, a dedicated SNN accelerator is implemented in 22 nm FDSOI CMOS suitable for continuous real-time classification. The direct comparison with an equivalent ANN counterpart shows that power reductions of 2.32× and area reductions of 7.22× are achievable without loss in QoS.**

***Index Terms*—ANN-SNN mapping, lossless coding, sparse temporal coding, application-specific integrated circuit (ASIC)**

I. Introduction

The early identification of anomalies in time-continuous signals is the backbone of multiple growing application domains such as predictive maintenance and health monitoring. For instance, atrial fibrillation (AF) is one of the most common heart diseases affecting about 3 % of the general population with a rising tendency [1]. The automated classification of anomalies in indicative data streams, here ECG signals, requires solutions that achieve high QoS and low power at the same time. Although several benchmarks [2], [3] are solvable by ANNs, model complexity increases enormously for increasing data noise and irregularity of less distinctive features as in AF.

To accommodate inference performance, SNNs encode activations in e.g. rates, timing or spatio-temporal features [4]. The sparse dynamic activity of spikes promises less energy while transmitting the same information. Within the diverse landscape of SNN training methods [5], ANN-SNN conversion techniques focus on the direct mapping of pre-trained ANNs to SNNs. They achieve competitive QoS in machine-learning (ML) benchmarks due to the SNN's close numerical relation to their reference [6]. Nevertheless, QoS degradation is still observed resulting from small cumulative deviations in the basic computation unit [7]. The impact is expected to be more dominant for less robust tasks, such as ECG classification [8].

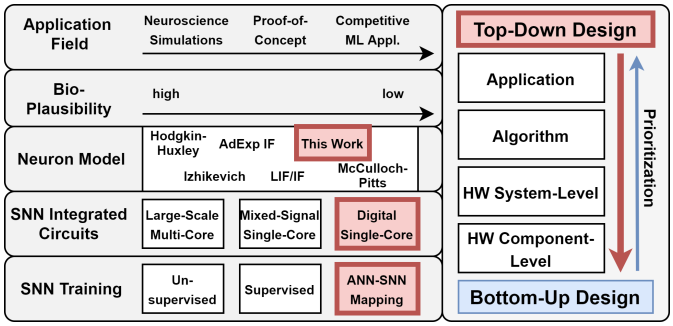


Fig. 1. Simplified overview of SNN design concepts ranging from training to circuit implementations in state-of-the-art. Complexity in all abstraction levels differ greatly dependent on application target and bio-plausibility.

From the perspective of neuromorphic accelerators [9], which support such SNNs, the spectrum of ASIC implementations and design concepts is huge (see Fig. 1). Even the subset, which implements ECG classification, incorporates a wide range of design architectures, i.e. large-scale multi-core [10], [11], mixed-signal single-core [12], [13] and digital single-core solutions [14], [15]. Mostly, SNN classifiers are designed using a bottom-up approach presuming a fixed processing unit behaviour, i.e. IF neurons with optional terms for further biological detail [12]–[15], or reutilizing existing accelerators [10], [11].

In contrast, this work approaches the problem of continuous time-series classification using a top-down design principle [16]. First, application constraints are specified to determine upper limits for time-to-solution or throughput. Based on given constraints, we develop a temporal coding scheme for lossless ANN-SNN conversion. Contrary to previous works, the basic processing element (PE) of the SNN is adapted in order to guarantee numerical equivalence with ANNs. In addition to identical SNN QoS, we also show that our proposed temporal coding scheme reduces the number of operations compared to equivalent fixed-point arithmetic for certain resolutions. As a proof-of-concept, multiple hardware designs with identical IO functionality are implemented, whereas the PE and corresponding control logic is interexchanged to quantify component-level impact on system-level performance.

The main contributions are summarized as follows:

978-3-9819263-7-8/DATE23/© 2023 EDAA

- Lossless ANN-SNN mapping method for temporal encoding of a fixed-point quantized reference (Section II-C)
- Corresponding neuron model implemented in a dedicated PE for SNN inference (Section III-C)
- System-level evaluation of SNN accelerator performance compared to an equivalent ANN accelerator implemented in a 22 nm FDSOI technology node (Section IV-A)

II. SNN CLASSIFIER

This section presents the systematic top-down approach for SNN design. Taking continuous ECG classification as an example, the general principles of the proposed methodology are introduced. Note that the baseline ANN, conversion scheme, and corresponding neuron model are transferable to other time-continuous signals without loss of generality.

A. Application Background

In the light of increasing model complexity, we chose a scalable ANN [17], which can process both the de-facto ECG benchmark (MIT-BIH dataset [2]) in hardware domain and a real-world AF classification scenario such as CinC'17 [3]. It comprises discrete wavelet transform (DWT) pre-processing (db2-wavelets) of depth 4, where only low rate coefficients are used for the ANN input. This relates to four subsequent stages of low-pass filters and subsampling components of factor 2, while the last stage contains an additional high-pass filter. The subsequent ANN is composed of 4 blocks with convolution of kernel size 5 and pooling of stride 3 followed by a final fully connected (FC) layer with kernel size 11. Training utilizes batch normalization (BN) layers adjacent to convolution layers. Before conversion, these are integrated into the weights of the convolutional operators by linear scaling [18].

TABLE I
SUMMARY OF BASELINE ANN

Layer	L1	L2	L3	L4	FC
Kernel	5	5	5	5	11
Output Channel	10	13	20	63	4
Subsampl. Factor	3	3	3	3	-

Table I summarizes the baseline ANN architecture. It achieves a competitive F1 score of 79 % (5-fold crossvalidation) in the CinC'17 benchmark similar to [17]. To validate ANN scalability and applicability in MIT-BIH, we trained its reduced version (L1 + FC only) on the corresponding benchmark discriminating abnormal from normal heart beats. Without hyperparameter tuning the reduced model achieves $\sim$94 % accuracy after 60 epochs using $<$2 % of its original's operations. In the following, however, we apply the proposed conversion method on the larger architecture to showcase its applicability in deeper ANN models.

Considering the application context of continuous classification, data is provided in a streaming fashion. Therefore, the ANN needs to process one input sample and resulting intermediate activations before the next sample arrives. Due to preceding DWT, the data rate from initially 300 Hz (ECG sampling rate in [3]) is reduced 4 times by 2$\times$ to 18.75 Hz. Hence, $T = 53.3\,\mathrm{ms}$ are available for the processing of each input sample. Subsampling in the ANN itself, i.e. resulting from pooling, creates sparser computations in deeper layers. This means that L1 is computed for each input, L2 is computed for every third, etc. A prediction is generated after the computation of the FC layer. Hence, the design's critical path $\tau_{crit} = \sum_i \tau_i$ is determined by the computation of all layers. Here, τ_i denotes the time to compute layer i.

B. Conversion Preparation

Weights and biases are normalized according to [18]. Due to model sensitivity for ECG [8], we use the max-norm to capture the full range of activations. Contrary to [18], ANN input samples after DWT contain negative values. Therefore, the scaling parameters in the first layer are modified[1] such that the additional offset of the number range is compensated.

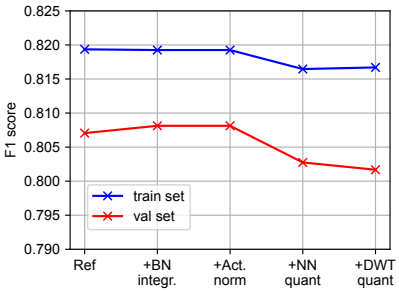


Fig. 2. Impact of normalization and fixed-point quantization on ANN reference. Network modifications are applied consecutively from left to right, while $n = 12$ bits are used for quantization.

An ablation study is performed to evaluate ANN QoS before mapping. Fig. 2 shows that the integration of BN as well as normalization of activations have no impact on QoS, while post-training quantization with $n = 12$ results in a minor degradation of less than 1 %. Here, n is chosen the same as in [17].

C. Coding Scheme

Fig. 3 shows a simplified overview of neuron models. Conventional ANN neurons calculate the dot-product of input vector $\mathbf{x}$ and weight vector $\mathbf{w}$ followed by an activation function. On the other hand, IF neurons integrate weighted input spikes in an internal state, i.e. the membrane voltage u. A spike is generated, once u exceeds a threshold θ. In sparse temporal coding methods such as time-to-first-spike (TTFS) the timing information of the first spike represents the neuron's activation [7]. In consequence, all further input spikes are inhibited through a refractory period to guarantee single spikes, but also resulting in a loss in input information.

In our approach, we omit the threshold-based nature of spike generation and split spike integration (decoding) and spike generation (encoding) into separate phases (similar to [19]). In the encoding phase, the vector of quantized activations $\hat{\mathbf{x}}$ is

[1]Linear transformation from arbitrary range $[c_{\min}, c_{\max}] \subset \mathbb{R}$ to $[0, 1]$

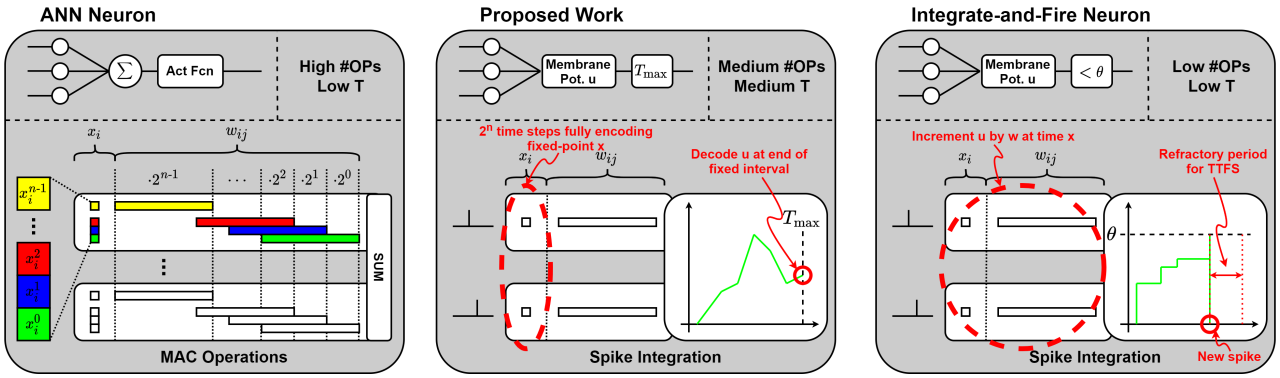


Fig. 3. Conceptual sketch of neuron models used in ANNs (left), proposed model (middle) and IF model (right).

linearly encoded into spike latencies $\hat{\mathbf{t}}$ with $\hat{\mathbf{t}} = \mathbf{1} - \hat{\mathbf{x}}$, where $\mathbf{1}$ is a vector of ones. The inverse linear property in addition to the discrete number representation of $\hat{\mathbf{x}}$ is crucial for the decoding phase. Note that elements in $\hat{\mathbf{x}}$ are representable purely by fractional bits, whereas $\hat{\mathbf{t}}$ is the inversion of $\hat{\mathbf{x}}$.

Considering one output neuron with $0 \leq t < 1$, the membrane potential is modeled, such that

$$u_i^l(t) = \int_{\tau_1=0}^{t} \int_{\tau_2=0}^{\tau_1} \sum_j w_{ij,\mathrm{inc}}^l(\tau_2)\, \mathrm{d}\tau_2\, \mathrm{d}\tau_1, \tag{1}$$

where

$$w_{ij,\mathrm{inc}}^l(\tau) = \begin{cases} w_{ij}^l & \tau = t_j^l \\ 0 & \text{otherwise} \end{cases}. \tag{2}$$

Here, $w_{ij,\mathrm{inc}}^l(\tau)$ denotes the weighted spikes in layer l and output neuron i induced by input neuron j. Eq. (1) and (2) are limited to one decoding phase only, after which the neuron is reset for a following iteration. Hereby, the membrane potential is a piecewise-linear function over time analogous to [7], with the exception of linear encoded inputs. The value at the end of the normalized computation time is corresponding to the convolved value with activation

$$a_i^{l+1} = \max(u_i^l(t=1) + b, 0) \tag{3}$$

after added bias and rectified linear unit (ReLU).

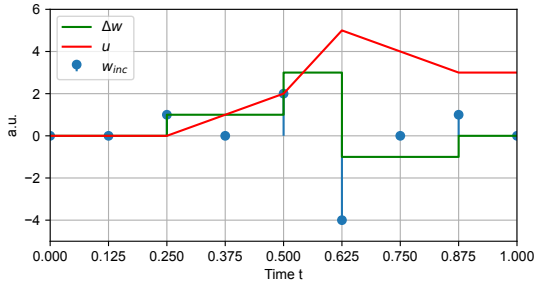


Fig. 4. Minimal example for proposed encoding scheme.

In principle, Eq. (1) and (2) can be interpreted as a decomposed dot-product over time. When the time step is chosen, such that it matches the resolution of a fixed-point number representation, it produces equivalent numerical values. The activations are fully representable by an integer value in the set $\mathbb{D} \in [0, 2^n - 1] \subset \mathbb{N}_0$ scaled by the constant factor 2^{-n}. The scaling of the temporal axis after conversion into spike latencies is a power of 2, which skews the result of the membrane voltage in comparison to a dot-product. These are compensated with constant shift operations after integration without hardware overhead.

We exemplify proposed encoding scheme in a minimal example for an activation word size of $n = 3$ and dot-product of $\hat{\mathbf{x}} = 2^{-3} \cdot [6, 4, 3, 1]^T$ and weights $\hat{\mathbf{w}} = [1, 2, -4, 1]^T$ (see Fig. 4). The corresponding spike latencies $\hat{\mathbf{t}} = 2^{-3} \cdot [2, 4, 5, 7]^T$ are weighted and integrated twice. The first integration results in a sum of weighted step functions representing the derivative Δw of the membrane voltage u. The second integration results in the actual value of u, whereas $u(t = 1) = \hat{\mathbf{x}} \cdot \hat{\mathbf{w}} = 2^{-3} \cdot 3$. Note that this example assumes integer weights. Similar to the activations, arbitrary positions of the radix point are considered with a normalization factor implemented as constant shifts.

III. System Architecture

This section performs a high-level analysis of system performance, similar to previous ANN-SNN mapping literature [7], [18], [19]. However, we additionally aim to compare the impact of the conversion scheme on dedicated logic implementations. Therefore, comparable ANN and SNN designs are implemented in a standard-cell based flow. A generic accelerator framework is designed to incorporate both conventional multiply-and-accumulate (MAC) based PEs and a novel SNN-based PE from neuron model in Section II-C.

A. Number of Operations vs. Precision

To investigate the impact of proposed SNN PE on performed operations, we compare the expected additions for one convolution operation. As mentioned earlier, the resolution of ANN quantization defines the size of the time step. Considering a word size n for all activations $\hat{\mathbf{x}}$ after quantization, the number of additions performed in a dot-product in fixed-point arithmetic is investigated. Here, a 1D-convolution with kernel size k and C input channels plus a bias term, i.e. $\hat{a}_{\mathrm{norm}}^{l+1} = \sum_{\kappa=0}^{k-1} \sum_{c=0}^{C-1} \hat{a}_{\mathrm{norm},\kappa c}^{l} \cdot w_{\kappa c} + b$, results in

$$N_{\mathrm{ADD,ref}} = n \cdot k \cdot C + 1 \tag{4}$$

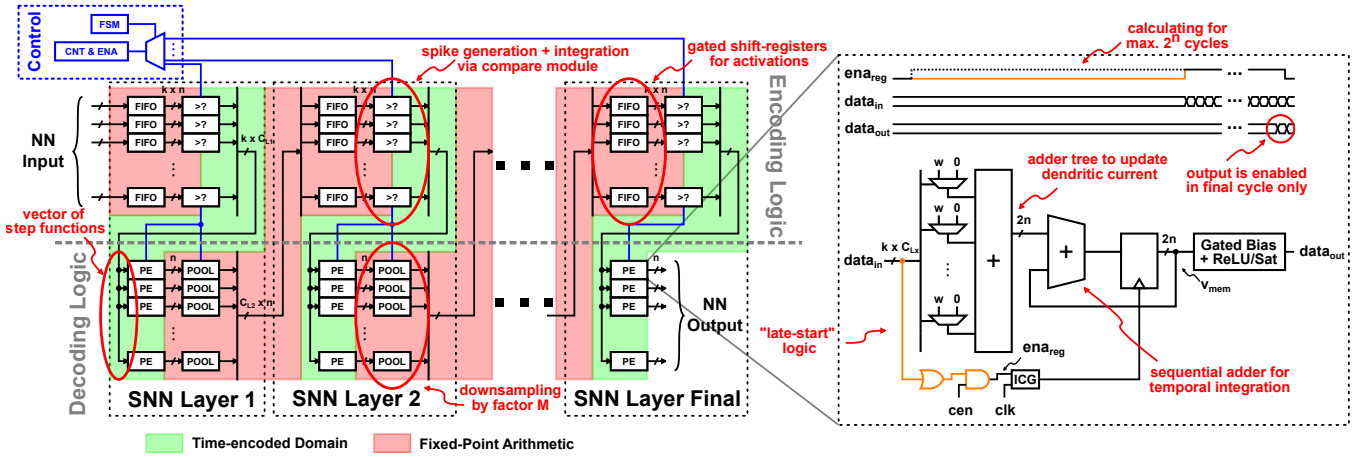


Fig. 5. Concept of proposed system-level architecture (left) and block diagram of a PE unit (right). The accelerator performs the convolution using the time-encoding scheme, while pooling operations and activation memory are calculated using fixed-point arithmetic.

additions. In case of our proposed coding scheme a spike triggers an update of Δw, while each time step accumulates Δw in u for 2^n cycles.

$$N_{\text{ADD,proposed}} = 2^n + k \cdot C + 1 \tag{5}$$

Fig. 6 visualizes the trade-off between both calculation methods. Due to the exponential relation, proposed time-encoding method needs less additions for $n \in [2, 14]$ with declining benefit for fewer input channels. Since chosen reference network is quantized to $n = 12$ bits and channel size is $C \in [2, 63]$, slightly less additions are expected for the reference. As the initial estimation only considers computational complexity, dedicated inference circuits might scale differently based on the mapping approach. Therefore, experimental results are necessary to evaluate the efficiency of circuits proposed in the Section III-C.

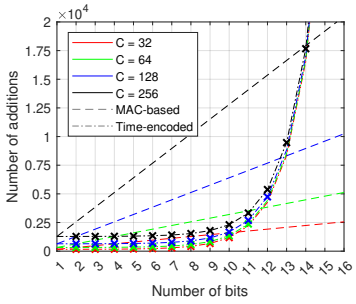


Fig. 6. Number of additions required to calculate a 1D-convolution with conventional and proposed method for kernel size $k = 5$.

B. SNN QoS-Latency Trade-off

We have established in Section II-A that a fixed time period $T = 53.3\,\text{ms} > \tau_{\text{crit}}$ is a real-time constraint on the SNN execution latency. Since the SNN's temporal resolution is directly dependent on the precision of the reference, a trade-off between QoS and latency is observed (see Fig. 7).

The Pareto-optimal front shows that the latency increases exponentially, while QoS converges to an optimum. Since layers are executed in sequence, the critical path results

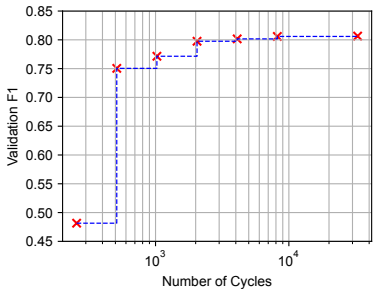


Fig. 7. Pareto-optimal front for number of cycles per layer and SNN QoS at temporal resolutions $n \in [8, 16] \subset \mathbb{N}_0$.

in $\tau_{\text{crit}} \approx 2^n \cdot L / f_{\text{sys}}$. Our proof-of-concept requires approx. 20k cycles for computation ($L = 5$ layers and $n = 12$). In the optimal case, the system frequency f_{sys} is chosen, such that $T = \tau_{\text{crit}}$. This results in a minimum frequency $f_{\text{sys,min}} \approx 384\,\text{kHz}$.

C. PE and Accelerator Design

Fig. 5 shows the overall accelerator design. Analogous to [17], the computations are mapped fully flat into synchronous digital logic. However, the general system architecture needs to be modified to incorporate multiple PE variants, while still providing the full functionality of the reference ANN classifier (see Section II-B).

First, activations are buffered in serial-in parallel-out (SIPO) registers for the calculations in the PEs. Then, after the PE is finished with its computation, the samples are inserted into a separate pooling unit. Since input samples are inserted sequentially, the pooling unit outputs only once every third input (see subsampling factor in Table I).

The three different PE variants are implemented for comparison, which provide identical IO functionality:

- **ANN Ref**: MAC-based ANN reference
- **SNN Base**: Proposed time-encoded SNN
- **SNN LS**: SNN Base with "late-start" (LS) logic

While the reference design implements conventional MAC units, the logic for the base SNN variant is shown in Fig. 5 on

the right. First, the PE requires a binary input from the SIPO registers. This is realized by a comparison between the register value and a decrementing (due to inversion of $\hat{\mathbf{x}}$) counter resulting in step functions. Then, those are weighted by multiplexing between 0 and their corresponding weight using the input signal as select signals. The results are accumulated spatially to generate the increment for the membrane voltage Δw, which is then integrated over time with a sequential adder. In the final cycle, the bias is added and the activation function is applied for insertion into pooling or buffer units of the next layer. Note that the only sequential element in the PE, i.e. accumulation register of u, is clock-gated, such that accumulation is only performed while a counter encodes the activations of its layer in a period of 2^n cycles.

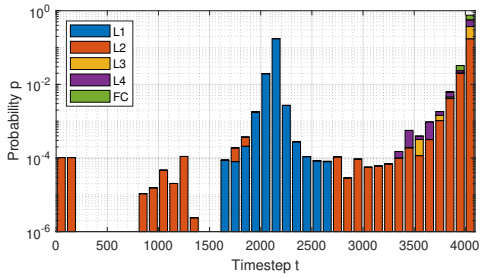


Fig. 8. Statistics of SNN activations after normalization and conversion to the temporal domain. A subset of 86 ECG samples is chosen for quick assessment, while the resulting distribution is similar towards several disjunct sets.

Fig. 8 shows the distribution of activations after normalization and encoding in the temporal domain. It is visible that early spikes are rare, which results in long idle periods at the beginning of the calculation period. Especially, in synchronous designs idle clock pin toggling results in significant overhead. The LS logic exploits that the majority of spikes are generated in later cycles. The goal is to enable the accumulation after the first spike changes the increment Δw to a non-zero value, thus, inducing a non-zero membrane voltage u. This is achieved by combining the ORed input with the global enable signal for input into the integrated clock-gating (ICG) cell.

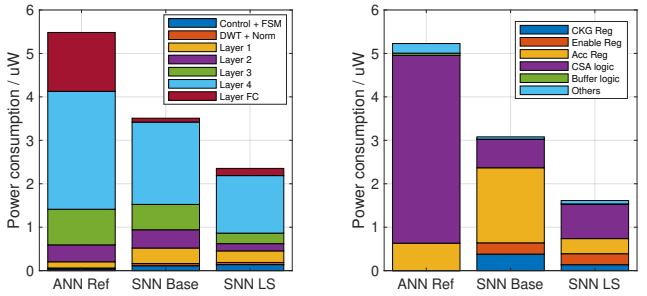
IV. Experimental Results and Discussion

The proposed design is implemented in a 22 nm FDSOI CMOS technology node and a standard digital design flow with industry-standard EDA tools is used for synthesis and simulation. PVT corners are provided as analysis corners to account for variation, whereas post-synthesis simulations and power estimations are performed under nominal conditions (TT process corner, @0.8V, at room temperature). The designs are operated at $f_{\text{sys}} = 1\,\text{MHz}$ satisfying minimum frequency requirement established in Section III-B, while maintaining enough slack to incorporate further cycles for logic-level debugging and analysis purposes.

The designs are tested under real-time conditions, i.e. ECG sequences @300 Hz are streamed into the DUT for 60 s. Traces are captured during worst case conditions, i.e. in the period from the input sample inducing computations in all layers until the result of the prediction is computed.

A. Comparison to the ANN Accelerator Baseline

As shown in Fig. 9a, the power consumption is reduced by 46 % for the base implementation of proposed SNN accelerator and even 57 % for the late-start version compared to the ANN reference. This corresponds to a total system power reduction of up to 2.32×. The reduction can be reasoned by the reduction of addition operations and corresponding adder logic (see Section III-A). Especially in layers with a high number of elements in the convolution operations, e.g. FC layer, the effect is most prominent. A component breakdown reveals, that up to 95.3 % of total power is consumed by the PE, whereas major reductions are achieved by reducing the carry-save adder (CSA) logic in the module (see Fig. 9b). This further supports the findings from the high-level analysis.



(a) System-level power consumption (b) Power consumption of PEs only

Fig. 9. Power breakdown of HW accelerators for ANN reference, SNN base implementation and SNN with late-start logic.

However, control logic consumes more power in the SNN designs (ANN ref: 0.66 %, SNN base: 3.23 %, SNN LS: 5.91 %), due to the additional counter logic for temporal encoding. However, compared to the PE units they are negligible.

Due to sequential accumulation, register activity is high even when Δw is zero given toggling of the clock pin. Although this results in an increase of register power (2.73×), the LS logic reduces this by 4.98× resulting in an overall PE power reduction of 3.24× compared to the ANN PEs.

TABLE II
Cell Area of Implemented Logic Designs

	ANN ref	SNN base	SNN LS
Area (μm^2)	843.2k	73.4k	116.8k

Table II summarizes the cell area of synthesized designs. SNN implementations occupy less, i.e. 11.49× for SNN base and 7.22× for SNN LS, when mapped fully flat into logic.

B. State-of-the-Art Discussion

As mentioned in Section I, this work approaches SNN acceleration from a radically different perspective than state-of-the-art. The direct comparison proves especially difficult, if the application context and classifier architecture deviates too much. Many works evaluate SNN inference efficiency without hardware implementations [7], [18], [19] or extrapolate energy consumption from spike events, thereby neglecting system periphery [12]. Further, we focus on neuromorphic designs

implementing ECG processing, which report system power as an optimization metric for our target application.

This work reimplements the architecture from [17] with a generic control architecture for better comparability of proposed PE implementation with its MAC-based reference. Furthermore, worst case power is compared at maximal load. Unfortunately, previous work only reports average power under power gating assumptions.

Other works design classifiers for an existing neuromorphic system, e.g. DYNAP-SE [10] or Loihi [11], or reconfigurable FPGA architectures [15]. Typically, they consume multiple orders of magnitude more power ($>500\,\mu W$) than the proposed system, due to system flexibility and scalability through sophisticated communication fabrics. Other works achieve remarkable power records ($<1\,\mu W$) through level-crossing analog-to-digital converters [14] and/or compute-in-memory (CIM) integration plus mixed-signal design [13]. Hereby, the SNN architecture and its training technique is optimized, such that good benchmark performance is achievable with minimal number of computations. Our applied top-down approach mainly intends to preserve full QoS of its reference. Therefore, performance gains in terms of power are relative to the ANN reference, while QoS can be optimized with quantization related methods [20]. Naturally, same device-level techniques are applicable for proposed architecture. Especially spike generation benefit from CIM structures with e.g. standard-cell based memory [21], due to its highly regular structure and thin computation near the activation memory.

Looking generally at digital neuromorphic circuits, which employ ANN-SNN mapped classifiers, temporal coding is rarely supported (exception: e.g. [22]). Instead, rate coding is more prominent, e.g. [22], [23], due to native support of IF neuron variants for such classifiers. However, recent works question the efficiency of corresponding digital implementations compared to conventional ANNs [24]. Our direct comparison shows that sparse temporal coding does enable more efficient neural computation than an ANN equivalent.

V. Conclusion

This work introduces a novel lossless ANN-SNN mapping method, which encodes an ANN reference with spike latencies. In a top-down approach a neuron model is derived, which produces numerically equivalent values compared to fixed-point ANN neurons. A system-level comparison reveals that the temporal decomposition of performed dot-product results in complexity reductions for a range of activation word lengths. These complexity reductions are seen in power estimations of hardware implementations with identical IO, which can classify ECG in real-time. Experimental results from a $22\,\mathrm{nm}$ FDSOI technology node reveal that proposed SNN solutions achieve power reductions of $2.32\times$ and area reductions of $7.22\times$ compared to its ANN reference.

References

[1] P. Kirchhof *et al.*, "2016 ESC guidelines for the management of atrial fibrillation developed in collaboration with EACTS," *European Heart Journal*, vol. 37, no. 38, pp. 2893–2962, Aug. 2016.

[2] G. Moody and R. Mark, "The impact of the MIT-BIH arrhythmia database," *IEEE Engineering in Medicine and Biology Magazine*, vol. 20, no. 3, pp. 45–50, 2001.

[3] G. Clifford *et al.*, "AF classification from a short single lead ECG recording: the physionet computing in cardiology challenge 2017," in *2017 Computing in Cardiology Conference (CinC)*. Computing in Cardiology, Sep. 2017.

[4] D. Auge *et al.*, "A survey of encoding techniques for signal processing in spiking neural networks," *Neural Processing Letters*, vol. 53, no. 6, pp. 4693–4710, Jul. 2021.

[5] M. Pfeiffer and T. Pfeil, "Deep learning with spiking neurons: Opportunities and challenges," *Frontiers in Neuroscience*, vol. 12, Oct. 2018.

[6] A. Tavanaei *et al.*, "Deep learning in spiking neural networks," *Neural Networks*, vol. 111, pp. 47–63, Mar. 2019.

[7] B. Rueckauer and S.-C. Liu, "Conversion of analog to spiking neural networks using sparse temporal coding," in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2018, pp. 1–5.

[8] J. Venton *et al.*, "Robustness of convolutional neural networks to physiological electrocardiogram noise," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 379, no. 2212, Oct. 2021.

[9] A. Basu *et al.*, "Spiking neural network integrated circuits: A review of trends and future directions," in *2022 IEEE Custom Integrated Circuits Conference (CICC)*. IEEE, Apr. 2022.

[10] 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. on Biomed. Circ. and Syst.*, vol. 13, no. 6, pp. 1575–1582, Dec. 2019.

[11] K. Buettner and A. D. George, "Heartbeat classification with spiking neural networks on the loihi neuromorphic processor," in *IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*. IEEE, Jul. 2021.

[12] A. Amirshahi and M. Hashemi, "ECG classification algorithm based on STDP and r-STDP neural networks for real-time monitoring on ultra low-power personal wearable devices," *IEEE Trans. on Biomed. Circ. and Syst.*, vol. 13, no. 6, pp. 1483–1493, Dec. 2019.

[13] Y. Liu *et al.*, "An 82nw 0.53pj/SOP clock-free spiking neural network with $40\mu s$ latency for AIoT wake-up functions using ultimate-event-driven bionic architecture and computing-in-memory technique," in *IEEE Int. Solid- State Circ. Conf. (ISSCC)*. IEEE, Feb. 2022.

[14] H. Chu *et al.*, "A neuromorphic processing system with spike-driven SNN processor for wearable ECG classification," *IEEE Trans. on Biomed. Circ. and Syst.*, pp. 1–12, 2022.

[15] Y. Xing *et al.*, "Accurate ECG classification based on spiking neural network and attentional mechanism for real-time implementation on personal portable devices," *Electronics*, Jun. 2022.

[16] C. Frenkel, D. Bol, and G. Indiveri, "Bottom-up and top-down neural processing systems design: Neuromorphic intelligence as the convergence of natural and artificial intelligence," *arXiv preprint*, Jun. 2021.

[17] J. Loh, J. Wen, and T. Gemmeke, "Low-cost DNN hardware accelerator for wearable, high-quality cardiac arrythmia detection," in *2020 IEEE 31st International Conference on Application-specific Systems, Architectures and Processors (ASAP)*, Jul. 2020.

[18] B. Rueckauer *et al.*, "Conversion of continuous-valued deep networks to efficient event-driven networks for image classification," *Frontiers in Neuroscience*, vol. 11, Dec. 2017.

[19] S. Park *et al.*, "T2FSNN: Deep spiking neural networks with time-to-first-spike coding," in *Proc. Design Automation Conf.*, ser. DAC '20, no. 25. IEEE Press, 2020, pp. 1–6.

[20] A. Gholami *et al.*, "A survey of quantization methods for efficient neural network inference," in *Low-Power Computer Vision*. Chapman and Hall/CRC, Jan. 2022, pp. 291–326.

[21] X. Fan *et al.*, "Synthesizable memory arrays based on logic gates for subthreshold operation in IoT," *IEEE Trans. on Circ. and Syst. I*, vol. 66, no. 3, pp. 941–954, Mar. 2019.

[22] J. Stuijt *et al.*, "µBrain: An event-driven and fully synthesizable architecture for spiking neural networks," *Frontiers in Neuroscience*, vol. 15, May 2021.

[23] D. Wang *et al.*, "Always-on, sub-300-nW, event-driven spiking neural network based on spike-driven clock-generation and clock- and power-gating for an ultra-low-power intelligent device," in *2020 IEEE Asian Solid-State Circuits Conference (A-SSCC)*. IEEE, Nov. 2020.

[24] S. Davidson and S. B. Furber, "Comparison of artificial and spiking neural networks on digital hardware," *Frontiers in Neuroscience*, vol. 15, Apr. 2021.