BrainWave: an energy-efficient EEG monitoring system - evaluation and trade-offs

BrainWave: an energy-efficient EEG monitoring system – evaluation and trade-offs

Barry de Bruin Kamlesh Singh Jos Huisken Henk Corporaal
Eindhoven University of Technology, The Netherlands
{e.d.bruin,k.k.singh,j.a.huisken,h.corporaal}@tue.nl

ABSTRACT

This paper presents the design and evaluation of an energy-efficient seizure detection system for emerging EEG-based monitoring applications, such as non-convulsive epileptic seizure detection and Freezing-of-Gait (FoG) detection. As part of the BrainWave system, a BrainWave processor for flexible and energy-efficient signal processing is designed. The key system design parameters, including algorithmic optimizations, feature offloading and near-threshold computing are evaluated in this work. The BrainWave processor is evaluated while executing a complex EEG-based epileptic seizure detection algorithm. In a 28-nm FDSOI technology, 325 μJ per classification at 0.9 V and 290 μJ at 0.5 V are achieved using an optimized software-only implementation. By leveraging a Coarse-Grained Reconfigurable Array (CGRA), 160 μJ and 135 μJ are obtained, respectively, while maintaining a high level of flexibility. Near-threshold computing combined with CGRA acceleration leads to an energy reduction of up to 59%, or 55% including idle-time overhead.

KEYWORDS

Wearable EEG monitoring, energy-efficiency, system-level trade-offs, edge processing, reconfigurable accelerators.

1 INTRODUCTION

Brain-related diseases, such as epilepsy and Parkinson's disease (PD), are severely degrading people's quality of life. Approximately 50 to 60 million people worldwide suffer from epilepsy or PD, making them one of the most common neurological diseases globally. Existing diagnosis and treatment methods for these require long-term in-hospital monitoring, which is costly, time-consuming and uncomfortable for the patients. Commercial devices for wearable ambulatory (Electroencephalography) EEG monitoring do exist, but these do generally support only a small number of EEG channels (e.g. EEG patch), have limited battery lifetime (e.g. TMSi Mobita, g.Nautilus-PRO), or use non-EEG sensors that are insufficient to reliably detect more complex brain-related seizure types[2]. This paper aims to address the energy problem for a wearable multi-channel EEG-based monitoring system.

An overview of the BrainWave monitoring system is shown in Fig. 1. To obtain pro-longed battery lifetime (>1 week) without compromising signal quality, different components in an EEG monitoring system need to be carefully tuned. State-of-the-art platforms utilize 10–12 bit ADCs, low noise amplifiers and advanced filtering in the Analog Front-End (AFE) to maximize battery life[3–5]. Biomedical signal processing platforms are commonly designed with multiple processor cores and are typically coupled with hardware accelerators[5–8]. Unfortunately, these architectures either lack energy-efficiency if the architecture is fully programmable, or are specialized towards a limited set of kernels. For emerging and complex monitoring tasks such as non-convulsive epileptic seizure detection and PD FoG prediction, research is ongoing on what algorithms and sensors work best. These applications demand an energy-efficient and flexible platform.

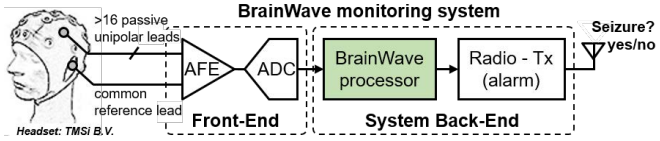

Figure 1: Overview of BrainWave system. BrainWave aims to enable ambulatory EEG monitoring through a portable battery-powered device.

In this work we present the design and evaluation of an energy-efficient and flexible BrainWave seizure detection system, targeted towards wearable 24/7 EEG monitoring. To the best of our knowledge, we are the first to present energy numbers for a signal processing system with CGRA running complex EEG features, including idle-time overhead. The main contributions are:

(1) Design of an efficient EEG monitoring system for a representative seizure detection pipeline (Sections 3.1 and 3.2);
(2) A new BrainWave processor for flexible and energy-efficient signal processing, which includes a CGRA specifically sized for complex EEG features (Section 4);
(3) Evaluation of key system design parameters, including cloud vs edge processing (Section 3.2), algorithmic optimizations (Section 3.1.2), feature offloading and voltage scaling to near-threshold region and duty-cycling (Section 5).

The rest of this paper is organized as follows: Section 2 covers related work. Section 3 introduces the seizure detection algorithm and baseline implementation and discusses energy trade-offs and requirements of wearable EEG monitoring systems. Section 4 introduces the BrainWave processor. Experimental results are provided in Section 5, followed by conclusion remarks in Section 6.

2 RELATED WORK

This section presents an overview of recent works related to EEG seizure detection algorithms and bio-medical monitoring systems.

1) EEG-based epileptic seizure detection: Acharya *et al.*[9] present an overview of common epileptic seizure detection algorithms.

ISLPED '20, August 10–12, 2020, Boston, MA, USA

ACM ISBN 978-1-4503-7053-0/20/08…$15.00
https://doi.org/10.1145/3370748.3406571

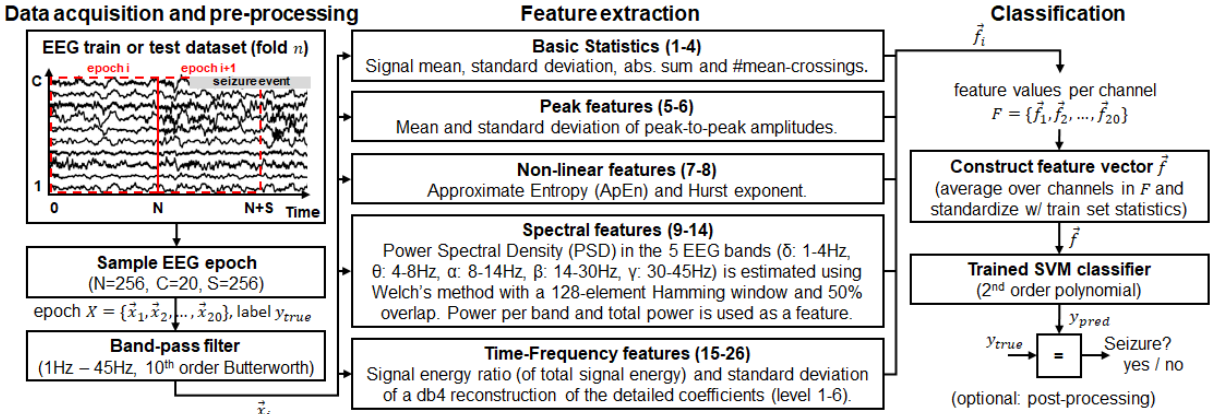

Figure 2: Overview of complete epileptic seizure classification pipeline that is evaluated in this work (based on the work of Wang *et al.*[1]).

The study points out that a wide variety of pre-processing, feature extraction algorithms and classifiers are currently being used for automated seizure detection. A survey and feature importance analysis on 47 common EEG features is conducted by Wang *et al.*[1]. Their findings indicate that the features in the (time-)frequency domain contribute the most to algorithm performance. Also non-linear features, such as entropy, are considered important.

2) Energy-efficient bio-medical processing platforms: Kwong *et al.*[6] employs a micro-processor with hardware accelerators for common bio-medical kernels (FFT, CORDIC, FIR and Median filtering) and reports platform-level energy-savings over 10× on two biomedical applications over a processor-only mapping. Lee *et al.*[7] propose a more flexible approach sharing a CORDIC, specialized data-path unit and a scratch-pad memory between an SVM and active-learning accelerator. This solution results in an 68.3× speedup and 144.7× energy reduction with respect to a processor-only approach. More recently, Coarse-Grained Reconfigurable Architectures (CGRAs) are being advertised as a good compromise between flexibility and energy-efficiency[10–12]. Das *et al.*[12] introduce a CGRA as a co-processor of a multi-core platform targeted towards ultra-low power edge processing. They obtain an energy gain of 6–18× for several common signal processing kernels, compared to a RISC processor. The authors of [11] extend a multi-core system with a CGRA and report 37.2% energy savings over a multi-core only implementation on a complex ECG algorithm. In this work we consider a signal processing system with CGRA running a more complex EEG-based seizure detection algorithm.

3) Power management for duty-cycled applications: In bio-medical monitoring applications the processor is only busy processing for a fraction of the epoch (i.e. the duty cycle). Therefore it is important to minimize system energy consumption during idle time. Power management knobs are also extensively explored[8, 13]. Montagna *et al.*[8] suggest that parallel processing in combination with near-threshold voltage operation leads to the best energy-efficiency for a EEG classification pipeline at different latency constraints. Hulzink *et al.*[13] propose an ECG monitoring system with a low-power sampling domain, which reduces the energy consumption by 3× while sampling. This work exploits duty-cycling to reduce energy consumption in idle mode and near-threshold computing to improve energy-efficiency. Also the idle-time energy overhead is considered.

3 SYSTEM DESIGN AND OPTIMIZATION

3.1 EEG-based epileptic seizure detection

EEG-based seizure detection algorithms typically operate in a periodic or windowed way, as is illustrated in Fig. 3. A time window (or epoch) of EEG samples is collected, the signal is cleaned (i.e. movement artifacts are removed) and important signal characteristics are computed using feature extraction algorithms. Based on the resulting feature values, the likelihood of a seizure being present in the current epoch is predicted by a machine learning classifier. To optimize the ratio between false positive and false negative classifications, a post-processing step is often employed. This step ranges from a patient-specific decision threshold to using a number of previous predictions for the final decision[14].

3.1.1 Epileptic Seizure detection algorithm. An overview of the seizure detection algorithm is depicted in Fig. 2. First an epoch with N samples and C EEG channels is sampled with a length of 2.56 s without overlap (i.e. stride $S = N$). Then each channel is pre-processed using a 10^{th} order 1 Hz–45 Hz Butterworth band-pass filter (BPF). This filter aims to suppress low-frequency movement artifacts and 50 Hz alternating current (AC) mains interference.

Several feature types are calculated from each EEG channel, including basic time-series statistics, traditional Spectral and Time-Frequency features and 2 non-linear features: Approximate Entropy (ApEn) to quantify the irregularity of a time-series and Hurst exponent to quantify the predictability of a time-series. The feature selection is motivated by the feature importance evaluation that is conducted in the work of Wang *et al.*[1]. More information on the calculation of these features can be found in[9]. In the classification stage the feature values of all channels are averaged. To aid Support

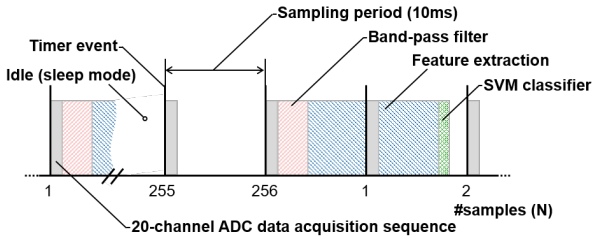

Figure 3: Illustration of duty-cycled seizure detection application.

Vector Machine (SVM) training, the feature vector is standardized using the training set statistics. A 2nd order polynomial kernel is used as a nice compromise between computational efficiency and seizure detection performance[15].

The complete pipeline is implemented in C and functionally verified against a high-precision Matlab reference implementation. For efficient deployment on an embedded platform with a 32-bit integer data path, the pipeline is quantized to fixed-point. Commonly used complex functions such as exp(), log() and sqrt() are implemented using fixed-point lookup tables.

3.1.2 Algorithm bottleneck analysis and optimization. To identify bottlenecks in the baseline mapping, analysis is performed using an open-source RISC-V micro-controller[16]. This micro-controller consists of a RISC-V core, whose performance is comparable to an ARM Cortex-M4 core; a popular low-power embedded signal processing core. The resulting execution time in cycles per classification is shown in Table 1. The results indicate that over 95% of the execution time is spend in the feature extraction stage and that the non-linear features account for 81.20% (ApEn: 79.24%, Hurst: 1.96%). Also the Band-pass filter and the computation of Time-Frequency features consumes a significant portion of the total run-time.

Based on these findings, 3 bottlenecks were further optimized:

1) Band-pass filter: The baseline implementation of the 10th order (21 taps) Butterworth filter takes approximately 16 cycles per MAC when implemented in direct-form II. To maintain filter stability, the coefficients require at least 24 bits of integer precision. As such, 64-bit multiplications and accumulation are emulated, which require 2–4 instructions per operation. Therefore, the filter was implemented using a cascade of 5 second-order sections, which requires only 12 bits of coefficient precision.

2) Non-linear features - ApEn: To compute this feature we consider a time series with N samples: $\vec{x} = \{x_1, x_2, ..., x_N\}$. From this sequence we extract $N - m + 1$ partially overlapping subvectors of length m, where $X_i^m = \{x_i, x_{i+1}, ..., x_{i+m-1}\}$. We define $P_i(m, r)$ as the likelihood of any subvector to be similar to X_i^m:

$$P_i(m, r) = (N - m + 1)^{-1} \sum_{j=1}^{N-m+1} C(X_i^m, X_j^m) \quad (1)$$

where the similarity condition for threshold r is defined as:

$$C(X_i^m, X_j^m) = \begin{cases} 1, & \text{if } \max |X_i^m - X_j^m| \leq r. \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

Approximate Entropy is now computed as follows:

$$ApEn(m, r) = \phi(m, r) - \phi(m + 1, r), \quad (3)$$

$$\text{where } \phi(m, r) = \frac{\sum_{i=1}^{N-m+1} \ln P_i(m, r)}{N - m + 1} \quad (4)$$

The run-time complexity of ApEn is determined by the number of similarity checks. To reduce the execution time several basic optimizations are typically used[17]. First, the computation of $\phi(m, r)$ and $\phi(m + 1, r)$ can be fused. Second, the distance between X_i^m and X_j^m is identical to the distance between X_j^m and X_i^m. A fast algorithm to compute ApEn for small input vectors ($N < 1000$) was proposed by Pan *et al.*[18]. By performing the vector similarity check in Equation 1 in ascending order, i.e. sorted based on the first element of every subvector, an early stopping rule can be constructed. The main idea is that if we iterate j in ascending order until X_j^m becomes dissimilar to X_i^m, then the remaining subvectors will also be dissimilar.

Table 1: Average run-time of fixed-point classification pipeline on RISC-V processor (20 channels × 256 samples epoch size).

Stage	Calls (#)	Cycles ($\times 10^6$)	Total (%)
Data acquisition	256	0.27	0.63
Band-pass filter	20	1.73	4.10
Basic statistics	20	0.11	0.27
Peak features[1]	20	0.58	1.36
Non-linear features[1]	20	34.35	81.20
Spectral features	20	1.34	3.18
Time-Frequency features	20	3.90	9.23
Construct feature vector	1	0.01	0.02
Trained SVM classifier	1	0.01	0.01
Total		42.31	100.00

[1] The run-time of the peak and non-linear features is data-dependent, and will therefore have a varying run-time for different input vectors.

3) Time-Frequency features: Time-Frequency features are computed using a multi-level wavelet decomposition followed by a reconstruction of the detailed coefficients of the first 6 levels. For each of these reconstructed levels, the standard deviation and energy ratio is used as a feature value. The decomposition and reconstruction operations are implemented using convolution-based discrete wavelet transforms (DWT) and inverse DWT (IDWT). The filters are derived from the Daubechies 4 (db4) wavelet, and consist of 8 taps per filter. The feature computation consists of one full multi-level signal decomposition, and 6 partial signal reconstructions.

A commonly used approach to speed up DWT/IDWT kernels is to implement a lifting scheme. For the db4 wavelet the filter length is reduced from 16 taps per DWT/IDWT operation to 10. However, the computational structure becomes more irregular due to the multiple smaller filter stages. Additionally, the boundary handling overhead for small input vectors is significant.

3.2 Wearable EEG monitoring systems

A system for wearable EEG monitoring is presented in Fig. 1. The system is divided in Front-End (FE) and Back-End (BE). The FE is responsible for data acquisition and analog-to-digital conversion. The BE performs the signal conditioning and seizure classification and optionally utilizes a wireless link to notify medical experts or to store data in the cloud for post-analysis. An important system design decision is whether the seizure detection pipeline is executed at the edge (on-chip) or in the cloud. In the former case seizure detection is performed on-chip, and only an alarm is send to a wireless end-point. In the latter case there is no on-chip signal analysis; the raw EEG data is transmitted to the end-point where the seizure detection is performed.

Table 2 lists an estimated energy breakdown based on a representative system. The system consists of a power-optimized EEG front-end, as proposed by Joo *et al.*[4], with an energy efficiency of up to 2 nJ/sample. For the wireless communication the Dialog DA14580 SoC[19] with integrated Bluetooth Low Energy (BLE) transceiver is chosen, which consumes 4.7 mA · 3 V / 128 kbit/s = 110 nJ/bit (payload) in transmission mode. The energy consumption of the seizure detection pipeline on a RISC-V micro-controller, as depicted in Table 2, is approximately 1.18 mJ/epoch (post-synthesis simulation;

Table 2: Energy breakdown between cloud and edge seizure detection system (20 channels × 256 samples epoch size).

Component	Cloud processing	Edge processing
AFE + 10-bit ADC[4]	0.01 mJ/epoch	0.01 mJ/epoch
RISC-V micro-controller[16]	not used	1.18 mJ/epoch
Radio - Tx[19]	5.63 mJ/epoch	≈ 0 mJ/epoch
Total	5.64 mJ/epoch	≈ 1.19 mJ/epoch

0.9 V/100 MHz). It can be observed that on-chip processing is more attractive due to reduction in wireless traffic. This is in line with other research, who generally perform digital signal processing and data reduction on-chip to minimize wireless communication[5, 20].

We conclude that the FE is not the primary energy bottleneck, when the ADCs are properly sized[21]. Therefore the main emphasis of this work focuses on optimization of the algorithm and digital BE for on-chip processing.

4 BRAINWAVE PROCESSING PLATFORM

This section introduces the BrainWave processor: a processor platform that is flexible and capable of performing energy-efficient signal processing. The BrainWave processor offloads complex EEG features to a CGRA and exploits duty-cycling with near-threshold computing to improve energy-efficiency. The processor architecture is depicted in Fig. 4a. The seizure detection algorithm runs on the single-issue RISC-V core with tightly-coupled program (PMEM) and data memories (DMEM). Periodic sampling from an (external) ADC is implemented using a timer. Every sampling period the RISC-V core will be waken up to issue an SPI read request to sample all EEG channels, as is illustrated in Fig. 3. When the transfer is completed, the core will copy the data from FIFO to the DMEM. The DMEM is sized to store up to 20 channels × 256 samples/epoch × 2 byte/sample elements (× 2 for double-buffering) and some scratch-pad memory to perform the feature computations. An UART is included to interface with an external radio module to notify a medical expert in case of emergency.

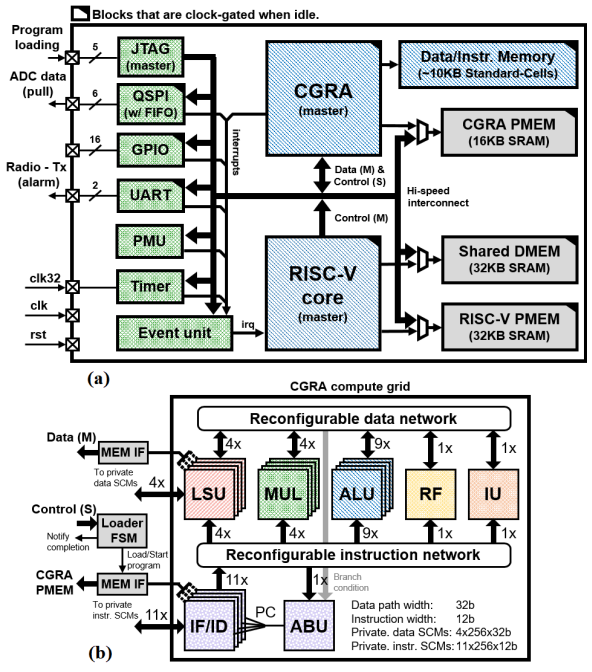

Figure 4: (a) BrainWave processor with CGRA (b) instantiated CGRA with memory and bus interfaces.

Most blocks can be clock-gated or disabled when not used. The Power Management Unit (PMU) supports explicit clock-gating of most peripherals, cores and accelerators. All SRAM memories have internal clock-gates and support for low-leakage retention modes. Computationally-intensive features are offloaded to a CGRA accelerator, as illustrated in Fig. 4b. This CGRA enables flexible and energy-efficient processing by providing programmable function units (FUs) and a reconfigurable data-path to bypass the register file[22]. These FUs operate in lock-step and act as a (Very long instruction word) VLIW processor.vector-processing (SIMD) is naturally supported since multiple FUs can share the same instructions and data via the reconfigurable data and instruction network. The CGRA has a private memory where its network configurations and programs are stored. CGRA programs and configurations can be reused by consecutive acceleration requests to reduce reconfiguration overhead. Typically the RISC-V core will issue a new acceleration request. After this request, the CGRA will start executing the preloaded program. Important parameters such as which kernel it should execute and where the data is stored, are read from a fixed location in the (shared) DMEM. A CGRA acceleration request takes 80–90 cycles.

The internal structure of the instantiated CGRA is illustrated in Fig. 4b. The CGRA consists of 6 different types of FUs, which can perform RISC-like instructions. More information on the instruction set can be found in[22]. Small local standard-cell memories (SCM) are used for local processing. Every Load-Store Unit (LSU) can access the shared data memory to access the EEG data. The currently loaded program is also stored in SCMs. The instantiated CGRA contains 20 FUs and 11 instruction decoders (IF/ID) that can be connected to one or more FUs. It supports up to 4 multiply-accumulations per cycle using 4-wide SIMD operation.

5 EXPERIMENTAL EVALUATION

5.1 Experimental Setup

5.1.1 System implementation. The complete design, as introduced in the previous section, is synthesized (using Cadence Genus) and evaluated in a commercial 28-nm FDSOI technology (SS corner), 12-track RVT standard cell library and Foundry SRAM memories. Evaluation of software optimizations is performed using RTL simulations. Power analysis is performed using post-synthesis netlist simulations for the operation conditions listed in Table 3. In all experiments the SRAM memories operate at 0.9 V and the IO pin power dissipation is ignored.

Table 3: BrainWave processor operation points (ASIC synthesis).

$V_{dd,logic}$	0.5 V	0.6 V	0.7 V	0.8 V	0.9 V
f_{max}	12.5 MHz	25 MHz	50 MHz	72.5 MHz	100 MHz

* Simulations were performed under typical conditions (TT corner, 25 °C).
* SRAM voltage $V_{dd,mem}$ is fixed to 0.9 V under all operations points.

5.1.2 Algorithm validation setup. In order to evaluate the performance degradation in epileptic seizure detection performance after fixed-point quantization, the SVM classifier is trained on a representative epileptic seizure dataset[1]. We report the SVM classification

Table 4: SVM classification performance summary on test folds.

Version	Precision	Recall	Specificity
Baseline - float32	0.94	0.57	1.0000
Quantized - int32[1]	0.68	0.57	0.9996
Quantized - int32 with retraining[1,2]	0.89	0.53	0.9999

[1] Complete application mapped onto 32-bit integer data path.
[2] SVM is retrained on quantized feature vectors.

Table 5: Application run-time (in cycles $\times 10^6$) breakdown on BrainWave processor (2.56s epoch with sampling rate Fs = 100Hz).

	Baseline	Optimized	
Stage	SW	SW	SW+CGRA
Band-pass filter	1.73	0.77	0.09
Non-linear features - ApEn[1]	33.53	4.13	1.03
Index sort	0.00	1.12	0.26
Similarity checking	32.93	2.42	0.70
Feature computation	0.59	0.59	0.07
Time-Frequency features	3.90	3.90	0.52
Db4 decomposition	0.46	0.46	0.08
Db4 reconstruction[2]	2.97	2.97	0.44
Feature computation	0.47	0.47	0.01
Remaining stages (Table 1)	3.14	3.14	3.14
Total	42.31	11.95	4.78

[1] Subvector length $m = 3$ and threshold $r = 0.125 \cdot \sigma(\vec{x})$ was used.
[2] Reconstruction of detail coefficients using multiple IDWT calls with zero-ed LPF at lowest level and zero-ed HPF at other levels.

accuracy on the test splits in terms of Precision (PPV), Recall (Sensitivity) and Specificity, which are defined as follows (given *t*rue/*f*alse *p*ositive/*n*egative classifications):

$$Prec. = \frac{tp}{tp + fp} \quad Rec. = \frac{tp}{tp + fn} \quad Spec. = \frac{tn}{tn + fp}$$

The dataset contains 24.7 h of continuous scalp EEG recording. 3 seizures are measured with an average duration of 28.3 s. All seizure events were contaminated with seizure-related muscle (EMG) artifacts in all channels. The dataset is captured and annotated by experts in a clinical environment, and is digitized using a 24-bit ADC at 100 samples per second. The recording consists of 20 unipolar EEG leads (referenced against ear lead) that were positioned following the international 10-20 electrode placement system.

5-fold cross validation (CV) is performed to evaluate the SVM performance. The subject data is split in 5 equal folds, of which one fold is used as test set and the remaining folds are used as training set. Test sets without seizures are excluded from the performance evaluation. As shown in Fig. 2, both sets are segmented in non-overlapping 2.56 s epochs. Epochs on the seizure onset/offset boundary are excluded. Finally, the resulting feature vectors are computed and standardized on the training split statistics.

The SVM classifier is optimized on the training set using a grid search on margin parameter $\gamma = 2^i$ for $i \in \{-1, 0, 1, 2, 3, 4\}$ and error penalty term $C = 10^j$ for $j \in \{-5, -4, -3\}$. C is kept small to reduce training time and to prevent overfitting on the training set. To account for the major class imbalance between seizure/non-seizure epochs, the SVMs are optimized using the F1-score, which is defined as the harmonic mean of Precision and Recall.

5.2 Results and Evaluation

5.2.1 Algorithm performance validation. The resulting training and quantization results are summarized in Table 4. The reference implementation is able to detect all 3 seizures. Quantizing the whole pipeline increases the number of false positives. However, retraining the SVM on the quantized feature vectors recovers most of the performance loss. It should be emphasized that the quantized pipeline still detects all seizures. The increased number of false-positives can likely be reduced using a post-processing stage[14], where only one alarm is generated for every detected seizure.

5.2.2 Algorithmic optimizations. The impact of the algorithmic optimizations from Section 3.1.2 are summarized in Table 5. The throughput of BPF is increased by approximately 2.6× on the RISC-V core (SW). The ApEn algorithm with basic optimizations improves the throughput by approximately 2.5×. When combined with the algorithm of Pan *et al.*[18] we report a final speedup 8.1× on the RISC-V core. Using the above-mentioned algorithmic optimizations, we improve the baseline throughput of the classification pipeline by 3.5×, without compromising on feature accuracy.

5.2.3 Energy efficient acceleration using the CGRA. Table 6 depicts a performance evaluation of the CGRA while executing the main bottleneck kernels. The BPF and DWT/IDWT kernels are computed using the same CGRA configuration, which is optimized to perform filter computations. The ApEn kernels (index sort and similarity checking) use a different configuration, tailored towards efficient sorting and similarity checking. Both configurations are able to exploit data-level parallelism by computing two EEG channels in parallel. As discussed in Section 3.1.2, the similarity checking kernel has a data-dependent early stopping condition, which in SIMD-mode has to wait until both channels are done. This approach enables computing ApEn with multiple EEG channels in parallel.

It follows from Table 6 that the CGRA is able to obtain kernel-level energy savings of 3.3×–4.6× over the RISC-V core. The ApEn kernels consume significantly less energy per operation, as the multipliers are not used. Also, the first 3 kernels have a significant number of stall cycles (≈ 20%), which are caused by the lack of latency hiding for global data memory accesses. For more complex kernels this issue is less significant, as most accesses are on the local data SCMs. Overall the results indicate that the CGRA is well-tailored for processing a variety of EEG kernels efficiently.

Table 6: Platform performance evaluation for different kernels at 0.9 V/100 MHz (for 2 channels × 256 samples).

		Execute on CGRA				Execute on RISC-V core	
Kernel	Description	Ops	Cycles	Utilization[1]	Energy (μJ)	Cycles	Energy (μJ)
BPF	5-stage biquad bandpass filter	76400	9149	42%	0.63	77100	2.10
DWT	full db4 wavelet decomposition	56869	7788	37%	0.37	46360	1.40
IDWT	full db4 wavelet reconstruction	58257	7820	37%	0.38	60148	1.74
Index sort	mergesort that sorts array indices	166144	26433	31%	0.70	112302	2.54
Similarity checking	2-dim. vector comp. loop with early exit	546941	62336	39%	1.97	241570	6.69

[1] Utilization = Ops / (FUs · Cycles). FUs = 20 for the instantiated CGRA.

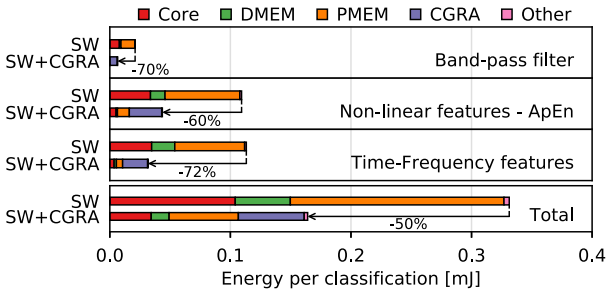

Figure 5: Energy consumption for different applications with (SW+CGRA) and without (SW) CGRA offloading at 0.9 V/100 MHz.

5.2.4 Feature offloading using the CGRA. In the previous paragraph the bottleneck kernels were offloaded to the CGRA to improve energy-efficiency. Parts of the feature computations that are not offloaded are computed on the RISC-V core. The resulting speedup is denoted in Table 5. It follows that the CGRA is able to obtain a speedup of 4.0×–8.6× for all offloaded kernels. It follows that CGRA offloading leads to a final speedup of 8.9×. The RISC-V core was able to perform work in parallel to the CGRA during the computation of both features. Fig. 5 depicts the energy breakdown for a single execution of the seizure detection pipeline at 0.9 V/100 MHz. From the results it follows that the SW+CGRA is able to save 70%–72% energy for the BPF and features, compared to a SW-only approach. A large fraction of the energy-savings is due the reduced number of processor PMEM and DMEM accesses. The execution time of the classification pipeline is reduced by 73.7%. The average energy savings for all three offloaded features combined is 66%. At 0.9 V/100 MHz, a single classification takes 325 μJ on the RISC-V core. In combination with the CGRA, 160 μJ is obtained. This results in an energy reduction of 50%.

5.2.5 Energy impact of near-threshold computing and duty-cycling. We conclude by investigating the energy-efficiency of the proposed platform and mapping while considering voltage scaling to near-threshold and duty-cycling. To calculate the energy consumption for a complete epoch, the following energy model is considered. The sampling and processing energy is computed using post-synthesis netlist simulations. The sampling energy is based on an epoch-length of 2.56 s. The CGRA is power-gated when the system is idle (modelled by assuming a 100× leakage reduction, see e.g. [10]). The resulting energy consumption for a complete epoch is depicted in Fig. 6. It follows that voltage scaling to near-threshold improves the energy-efficiency by up to 33% at 0.5 V/12.5 MHz, or up to 55% including CGRA acceleration. It should be noted that the sampling energy is still very significant, even at low voltages. This can be primarily attributed to the SRAM memory leakage, despite being in low-leakage mode.

6 CONCLUSIONS

This paper presents the methodology of design and evaluation of the energy efficient BrainWave system for wearable EEG-based monitoring. The system combines a RISC-V core with a CGRA and optimized mappings of several complex and common seizure detection features. Utilizing algorithmic optimizations leads to throughput improvements up to 8.1× for the ApEn feature on the RISC-V core. Feature offloading using the CGRA improves the throughput

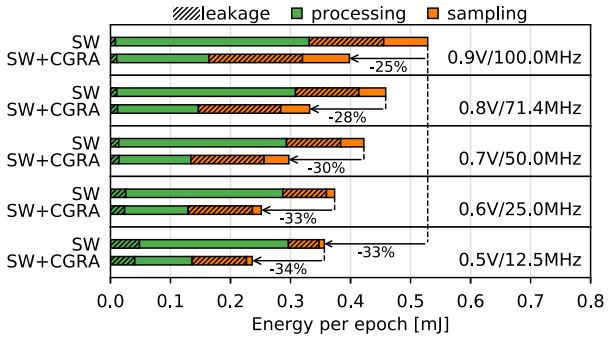

Figure 6: Energy consumption with duty-cycling (processing energy of 0.9 V/100 MHz corresponds to total energy in Fig. 5).

by 4× and energy by 60% over an optimized RISC-V implementation. The system-level efficiency using a complex and representative epileptic seizure detection pipeline is validated. The results indicate that combining near-threshold computing with CGRA acceleration leads to an energy reduction of up to 55%, including idle-time overhead. Future directions include improving the memory subsystem and further specialization of the CGRA.

ACKNOWLEDGMENTS

This work is funded by the Dutch NWO project 14714 BrainWave.

REFERENCES

[1] L. Wang *et al.*, "Seizure pattern-specific epileptic epoch detection in patients with intellectual disability," *Elsevier BSPC*, 2017.
[2] A. Ulate-Campos *et al.*, "Automated seizure detection systems and their effectiveness for each type of seizure," *Seizure*, vol. 40, 2016.
[3] N. Verma *et al.*, "A micro-power EEG acquisition SoC with integrated seizure detection processor for continuous patient monitoring," in *IEEE VLSIC*, 2009.
[4] J. Yoo *et al.*, "An 8-channel scalable EEG acquisition SoC with patient-specific seizure classification and recording processor," *IEEE JSSC*, 2013.
[5] M. A. B. Altaf *et al.*, "Design of energy-efficient on-chip EEG classification and recording processors for wearable environments," in *IEEE ISCAS*, 2016.
[6] J. Kwong *et al.*, "An energy-efficient biomedical signal processing platform," *IEEE JSSC*, 2011.
[7] K. H. Lee *et al.*, "A low-power processor with configurable embedded machine-learning accelerators for high-order and adaptive analysis of medical-sensor signals," *IEEE JSSC*, 2013.
[8] F. Montagna *et al.*, "Flexible, scalable and energy efficient bio-signals processing on the pulp platform: A case study on seizure detection," *MDPI JLPEA*, 2017.
[9] U. R. Acharya *et al.*, "Automated EEG analysis of epilepsy: A review," *Knowledge-Based Systems*, 2013.
[10] C. Kim *et al.*, "Ulp-srp: Ultra low-power samsung reconfigurable processor for biomedical applications," *ACM TRETS*, 2014.
[11] L. Duch *et al.*, "HEAL-WEAR: An ultra-low power heterogeneous system for bio-signal analysis," *IEEE TCSI*, 2017.
[12] S. Das *et al.*, "An energy-efficient integrated programmable array accelerator and compilation flow for near-sensor ultralow power processing," *IEEE TCAD*, 2019.
[13] J. Hulzink *et al.*, "An ultra low energy biomedical signal processing system operating at near-threshold," *IEEE TBioCAS*, 2011.
[14] L. Chisci *et al.*, "Real-time epileptic seizure prediction using ar models and support vector machines," *IEEE TBME*, 2010.
[15] K. H. Lee *et al.*, "Improving kernel-energy trade-offs for machine learning in implantable and wearable biomedical applications," in *IEEE ICASSP*, 2011.
[16] A. Traber *et al.*, "PULPino: datasheet," *ETH Zurich*, 2017.
[17] G. Manis, "Fast computation of approximate entropy," *Elsevier CMPB*, 2008.
[18] Y.-H. Pan *et al.*, "Fast computation of sample entropy and approximate entropy in biomedicine," *Elsevier CMPB*, 2011.
[19] *DA14580 Datasheet*, Dialog Semiconductor, 11 2016, rev. 3.4.
[20] S. Iranmanesh *et al.*, "A 950 nw analog-based data reduction chip for wearable EEG systems in epilepsy," *IEEE JSSC*, 2017.
[21] B. Murmann, "ADC performance survey 1997-2018," http://web.stanford.edu/~murmann/adcsurvey.html, 2018, [Online].
[22] M. Wijtvliet *et al.*, "Blocks: Redesigning coarse grained reconfigurable architectures for energy efficiency," in *FPL*, 2019.