Microwatt End-to-End Digital Neural Signal Processing Systems for Motor Intention Decoding

Zhewei Jiang[1], Chisung Bae[2], Joonseong Kang[2], Sang Joon Kim[2], and Mingoo Seok[1]
[1]Department of Electrical ngineering, Columbia University, New York, US
[2]Samsung Electronics, South Korea
{zj2139, ms4415}@columbia.edu

***Abstract*: This paper presents microwatt end-to-end digital signal processing (DSP) systems for deployment-stage real-time upper-limb movement intent decoding. This brain computer interface (BCI) DSP systems feature intercellular spike detection, sorting, and decoding operations for a 96-channel prosthetic implant. We design the algorithms for those operations to achieve minimal computation complexity while matching or advancing the accuracy of state-of-art BCI sorting and movement decoding. Based on those algorithms, we architect the DSP hardware with the focus on hardware reuse and event-driven operation. The VLSI implementation of the proposed systems in a 65-nm high-V_{TH} shows that it can achieve 4.82 µW at the supply voltage of 300mV in the post-layout simulation. The area is 0.16 mm².**

Keywords: brain machine interface; prosthesis; low power VLSI; Kalman filter;

I. INTRODUCTION

Advances in brain-computer-interface (BCI) research has enabled the development of active prosthesis that can perform intended movement by mapping subject intention to neural activity, providing invaluable rehabilitative services [1-5]. For locked-in patients with no residual muscle activation, active BCI prosthesis is the only alternative to regain a level of motor capabilities. A prosthetic BCI operates by measuring neural activity and inferring the intended movement based on a learned model (cortical map) that relates the neural behavior to the movement. Any neural signals encoding motor intention can support BCI prosthesis if the encoding scheme can be reliably modeled. Extracellular spiking activity from pre-motor or motor cortex is currently the state-of-art for upper limb movement decoding [3], outperforming non-invasive systems based on signals such as EEG [4] or MEG[5].

This paper presents the design and validation of a prosthetic BCI digital signal processing (DSP) back end for real time movement intention decoding from spiking information. As shown in Fig. 1, the full DSP back end is composed of on-chip hardware processing (for detection, feature extraction and clustering, and the first half of intention decoding [ensemble averaging]) at the implant site and off-chip software processing (for the second half of intention decoding [kinematic state prediction]). We partition the task to minimize both the on-chip computation and the transmission to off-chip, which are critical to scale the power dissipation of an implant.

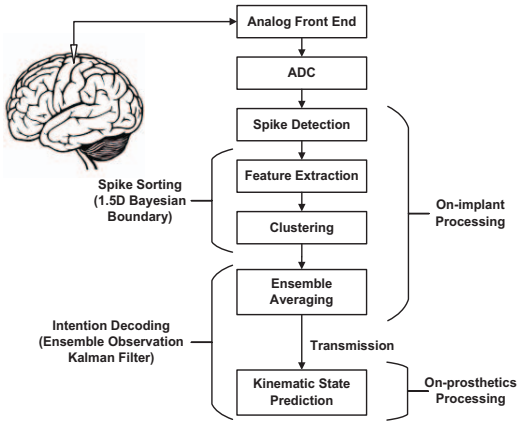

Fig. 1. Proposed prosthetic BCI task flow

The same consideration also motivates the algorithmic and architectural design of the system. Based on the prior works [1,2,6], we upgrade the algorithms for spike sorting and intention decoding to reduce computational complexity but holding accuracy performance comparable with or better than unmodified algorithms.

Also, we create hardware architecture that is event-driven for maximizing resource sharing within the constraints of input rate and characteristics (e.g., the number of spike clusters per channel).

The VLSI implementation of the proposed architecture for 96-channel implants achieves 4.82 µW power dissipation (mostly leakage) and 0.16 mm² silicon area at 0.3V supply voltage (V_{DD}), confirming the feasibility of BCI DSP design that is low power, compact, and accurate enough for miniaturized implants. Particularly the power consumption of our design is very low, comparable to that of Ref. [10] which implements only a single channel sorter in the same 65nm.

The remainder of the paper is organized as follows. In Section II, we first briefly present the scope and tasks involved in the BCI system. We then detail the algorithms and evaluate the respective accuracies and cost of implementations. In Section III, we present the architecture of the on-chip DSP component as well as the VLSI implementation. Section IV concludes the paper.

Samsung Global Research Origination (GRO) and Wei Family Private Foundation (WFPF) financially supported this work.

978-3-9815370-8-6/17/$31.00 ©2017 IEEE

II. Systems and Algorithms

A. System Tasks

In a BCI implant, an implanted electrode array senses extracellular voltages from surrounding neurons, potentially with band-pass or low-pass filtering. Then, analog-to-digital converters (ADC) digitize the signals and produce multi-channel digital data streams. Our BCI DSP, as shown in Fig.1, takes those streams and detects neural activations (spikes) via thresholding the action potential.

The DSP then performs spike sorting. Each extracellular electrode measures the activity of neurons in its proximity; hence a sorting process is required to differentiate spikes by their originating neurons. The basis for sorting is the spike shape, under the assumption that each neuron in a channel has various distances (which make filtering effects) to the electrode and unique internal ionic gating states, both of which affect spike shapes. The sorting process identifies the neurons that the electrode array observes.

The neurons identified after sorting need analysis on their spiking behavior and efficacy in kinematic information encoding. This calibration procedure is performed with in-patient experiments, and typically cannot be automated in the deployment system. Those neurons found to encode significant kinematics are used in the following intention decoding steps.

The intention decoding steps mainly use the instantaneous spiking rates (spike count in time bins typically around 100 ms) from the selected neuron population and a cortical mapping that relates the spiking rates and kinematic estimates. We can create the cortical mapping by regressing training data, typically at the same time when we identify the neurons that encode significant kinematics. Finally, we use filtering techniques such as the state-of-the-art Kalman filter to refine the kinematic estimates.

B. Spike Sorting

Transmission is the dominate power consumer for existing BCI implants [7]. Consider a typical 96-electrode array sensing at 8-bit resolution at 30 kHz for prosthetic BCI, the data rate is nearly 3MB/s. If the data are transmitted off-chip before sorting, the required power would make the system unsuitable for long term deployment.

However, for the motor-intention BCI application, only specific neuron spiking rates are relevant. Spike sorting is the process of identifying each spike's origin, encoding constant stream of voltage sample into sparse time stamps and neuron identifiers. This process provides substantial savings in power. Assuming 10-bit identifier for 96 channels (roughly 10 identifiers for each channel), 4 clusters per channel, and an average spiking rate of 50 Hz, spike sorting can achieve ~100X transmission reduction.

A study on feasibility of on-chip spike sorting [8] has shown that the hardware required for real time sorting is feasible and nets positive saving in system-level power. Our goal here is to design algorithm and hardware that consumes minimal energy and area at high accuracy.

1) Boundary-based Spike Sorting

Our sorter is based on our previous single-channel design using Bayesian decision boundaries [6]. It uses two voltage samples from the spike waveform as features, selected for the best

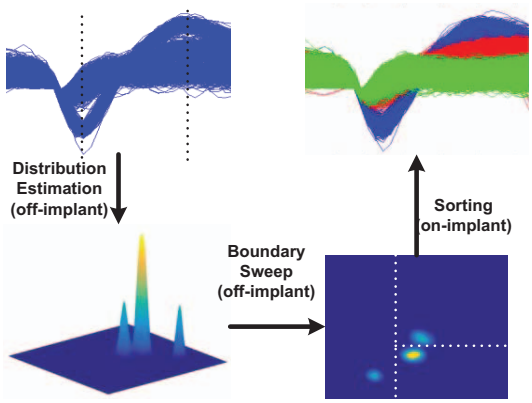

Fig. 2. Visuallization of spike soting via 1.5D Bayesian Boundaries

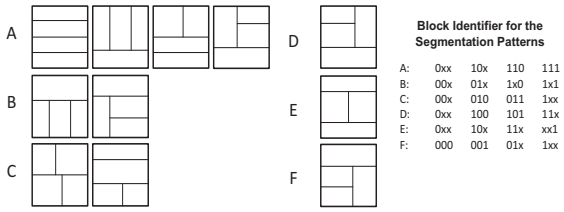

Fig. 3. Feature space segmented block identification coding

sorting performance. The theoretically optimal sorting solution in the 2D feature space is free-form Bayesian boundaries from clusters' distribution intersects which are not constraint to the orthogonal grid. The optimal solution is however too costly as it requires large on-chip memory to store free-form boundaries. Instead, we propose to use the grid-constrained boundary, a 1.5D compromise, for relaxing hardware requirement. Its impact on accuracy can be minimized in feature selection.

We identify optimal 1.5D boundaries in the offline training (Fig. 2). We first estimate the distribution of data points with Gaussian kernels. We then discover optimal 1.5D boundaries via simple sweeping under the orthogonality constraint. The boundary information (direction, order, and values) is stored in implants. We use off-line training since the tasks in the target application such as neural information encoding analysis and ensemble averaging weight regression anyway involve extensive off-line in-patient calibration. We can simply reuse the data to train the sorter when performing those tasks. Offline training has an accuracy advantage over the on-line one used in Ref. [6] by identifying most suitable features and higher precision boundaries.

We limit the maximum number of neurons per channel to four, which is rarely exceeded. In the case of more than four clusters present in a channel, decoding is not affected with a heuristic that no more than three neurons from each channel are selected for intention decoding.

Under the 4-cluster-per-channel constraint, as shown in Fig. 3, eleven unique segmentation patterns (diagonally symmetric patterns are considered the same) exist in the 2D feature space. Each spike, as a point in 2D feature space, can be located by comparing with the boundaries. Its cluster is identified by the 3

bit comparison results. The 11 segmentation patterns are organized into six groups. For all segmentation patterns in the same group, the four blocks share the same boundary comparison identifier. This encoding scheme reduces memory required to characterize the channel from the on-line trained implementation in [6]. The memory saving is mainly the result of how to handle the worst case, as the 11 patterns are subsets of a 4X4 grid. Online training need to allocate memory for the full grid as the segmentation pattern is unknown. This requires three boundaries along each dimension, and a large hash table to store 16 blocks' information (cluster validness, its associated weight for intention decoding). The proposed off-line variant thus has more than 50% reduction in memory requirement.

2) Sorting Accuracy and Cost Evaluations

We use the data measured from mice [9] to evaluate the sorting accuracy of our proposed algorithm. The dataset (D1, D2, D3, and D4) contains channels having signals from two to four neurons. The sampling rate is 40 kHz, band-pass filtered between 300 Hz and 5 kHz.

We categorize prior works on on-chip spike sorting by their decision metric, distance to template [7,10,11] or boundary [6]. Due to the high area and power cost of a multiplier, distance-based sorters so far all use L1 distance metric (sum of absolute difference) on time-domain features. In Table I, we compare our 1.5D boundary to the L1 norm and find that the 1.5D boundary has a comparable sorting accuracy with L1 norm of two features.

TABLE I. SPIKE SORTING ACCURACY

Decision Metric	Datasets (Number of neurons)			
	D1(2)	*D2 (3)*	*D3 (4)*	*D4 (4)*
L1 Norm	99.97%	99.25%	91.39%	89.00%
1.5D Boundary	99.99%	99.19%	91.61%	89.49%

The decision boundary of L1 norm is orthogonal to the vector linking the point templates in the feature space, which theoretically should outperform the 1.5D boundary. However, as the features are voltage samples taken directly from the spike waveform, its non-idealities account for its sorting performance. Since spike peak, trough, and relaxation slope are primarily driven by different ion pumps, different parts of a spike waveform have different variances. L1 metric does not consider this phenomenon; thus its sorting accuracy is negatively affected.

Our proposed 1.5D boundary metric requires much less on-chip memory and computation. The memory required in the presented sorter per channel is 3 bytes for boundaries, 4 bits for segmentation pattern (total 28 bits) for n-cluster per channel ($n < 5$). By comparison, a 2-feature L1 metric requires $2 \cdot n$ bytes (64 bits when n=4). In term of the computation load, 2-feature L1 requires $3 \cdot n$ 8-b additions/subtractions and n-1 8-b comparisons, while our proposed algorithm needs to perform three 8-b comparisons and 3-b 4-entry memory read.

C. Intention Decoding

1) Ensemble Observation Kalman Filter

In this section, we propose a modified Kalman filter (KF) using ensemble averaged spiking rates as state observation, which we call ensemble observation Kalman filter (EOKF). This filter can maintain (or improve) decoding accuracy, reduce on-chip computation workload, and minimize the data rate in transmission to off implant.

The standard KF for decoding motor intention has the following form [1]:

$$x_{k+1} = A\, x_k + w_k \tag{1}$$

$$z_k = H\, x_k + q_k \tag{2}$$

where the term x is the state, i.e. position, velocity, etc. A is the state transition matrix; w is the state noise (typically zero mean Gaussian variable); z is observation, i.e., binned spiking rates; H is the observation matrix, i.e. cortical map; q is observation noise (zero mean Gaussian variable); subscript k is time step. Eq. (1) describes the state transition in a Markov chain. It provides a constraint on the estimate made from cortically mapping spiking rates in Eq. (2). Given an estimated state, Eq. (2) reconstructs the expected spiking rates from the selected population.

The computational complexity of the conventional KF is high. It updates its state and parameters iteratively, beginning with an a priori estimate of next kinematic state via the Markov process. A priori estimate is given as:

$$\bar{x}_k = A\, \bar{x}_{k-1} \tag{3}$$

The standard KF's posterior estimate combines the passive estimate from state transition and the weighted error between expected observation and actual observation in the following equation:

$$x_k = \bar{x}_k + K_k (z_k - H\, \bar{x}_k) \tag{4}$$

The error weights are known as Kalman gain K, computed as:

$$K_k = \bar{P}_k H^T (H\, \bar{P}_k H^T + Q)^{-1} \tag{5}$$

The Kalman gain is updated with the error covariance matrix P and measurement error matrix Q. Eq. (5) represents a substantial computation load, since the number of neurons selected for motor intention decoding is relatively large,

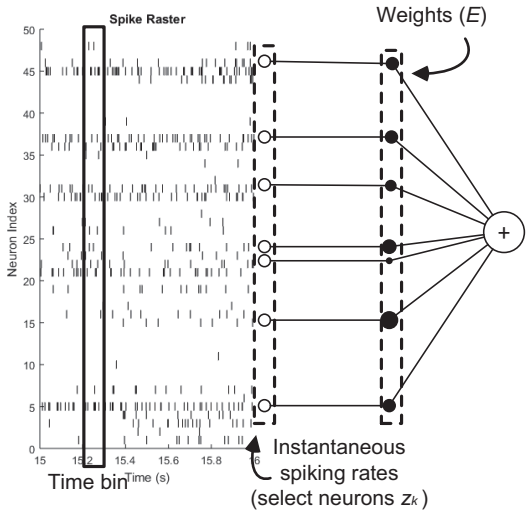

Fig. 4. Instantaneous spiking rates weighted in ensemble average for the observation based estimate of a kinematic state

typically between 20 to 50 [3][12], making $H\,P^-_k\,H^T$ at least a 20x20 matrix. To compute its matrix inverse in implants can cause a considerable amount of power and area overhead.

To reduce this computational complexity, we propose an inverse form of observation-to-state transition as a committee machine, essentially changing (2) to:

$$x_k = E\,z_k + q_k \tag{6}$$

in which the expected state is constructed with ensemble averaging of spiking rates, weighted by E. Fig. 4 illustrates this. Ensemble averaged spiking rate becomes an observation-based estimate of the kinematic state. The cortical map H becomes trivial (identity matrix) in EOKF, and therefore reducing the computational workload henceforth. The proposed filter has the same a priori estimate as the standard KF, as shown in Eq. (4).

In the proposed EOKF, (5) and (6) are reduced to:

$$x_k = x^-_k + K_k(\,E\,z_k - x^-_k\,) \tag{7}$$

$$K_k = P^-_k(P^-_k + Q\,)^{-1} \tag{8}$$

With the proposed filter (i.e., Eqs. (4), (7), (8)), *the lowest data bandwidth of the entire BCI system is located at computation of* $E{\cdot}z_k$. This term has the equivalent data rate as the final decoder output (x_k). Therefore, we perform only the computation of $E{\cdot}z_k$ and all the other computations are offloaded to prosthetics sites where power and area budget is much greater. This partition also includes the computation of (3) in the prosthetics site since there is no data dependency.

The posterior error covariance matrix updates in the standard KF in the following form.

$$P^-_k = AP_{k\text{-}1}A^T + W \tag{9}$$

$$P_k = (I - K_kH\,)\,P^-_k \tag{10}$$

The error covariance estimate is first updated with state transition and state error variance W in (9). The posterior error covariance incorporates the observation error in the form of Kalman gain in (10). In the proposed filter, (10) is reduced to (11).

$$P_k = (I - K_k\,)\,P^-_k \tag{11}$$

The posterior error covariance matrix width is now the number of state elements instead of the number of neurons.

We compare the number of multiplications, additions, and divisions in the proposed EOKF and the standard KF (Table II) with an assumption that 20 neurons are selected for intention decoding. We can find one or two orders of magnitude reduction in those operations, even for the case to include the computations both on implants and prosthetics site.

TABLE II. COMPARISON OF NUMBER OF CALCULATIONS IN EOKF AND STANDARD KF (20-NEURON CORTICAL MAP)

Equation	Number of Calculations (Mult/Add/Div)	
	Standard Kalman	*Ensemble observation Kalman*
(4)	4/2/~	4/2/~
(5)/(7)	80/80/~	46/46/~
(6)/(8)	32180/32060/1180	10/9/4
(9)	8/8/~	8/8/~
(10)/(11)	88/84/~	8/8/~

2) Evalution of Kinematics Reconstruction

To evaluate the decoding performance of our proposed EOKF, we use an upper-limb reaching data set [13] from the Database for Reaching Experiment and Models (DREAM) [14]. The task is the standard 2D equal-distance 8-target center-out reach-and-return performed by a Rhesus monkey well trained in the experiment. Only the velocity vector of the hand movement in the x, y plane is used as kinematic state, same as the velocity-Kalman study [2]. The data set contains 194 trials of the 196-neuron spiking traces from the motor cortex. Detailed experimental setup can be found in [8].

In this study, the data is used for offline reconstruction of native movements. We train the filter state parameters (transition matrices, error variances) with 80% of the data (randomly selected for each trial). Reconstruction is done on the remaining 20%.

The proposed EOKF outperforms the standard KF with lower trace reconstruction error, by ~5% to ~46% depending on trial selection. This outperformance comes from the advantage that the proposed EOKF always has better observation than the standard KF. In the proposed EOKF, the error variance of the observation-based estimate has an upper bound that is equal to the lowest error variance of its members, in which case, the committee is trivial having the single member determine the output.

The advantage of the committee machine also manifests in error variance consistency. Fig. 5 demonstrates the regression residuals of an ensemble and a single neuron. Each dot represents a kinematic state instance. The residual is color-mapped to show that the accuracy of single neuron model (used in the standard KF) is less consistent across directions and movement speed than the ensemble model used in the proposed EOKF. The error variance matrices are assumed to be invariant in both the standard and ensemble observation KF; hence, the ensemble better fits the variance model due to its evenness.

While EOKF has better observation model, this is not the full story. EOKF is not expected to outperform the standard KF in all conditions. Due to data dependence, optimal observation model can range from single neuron observations (standard) to single committee (EOKF) and in-between (multiple sub-

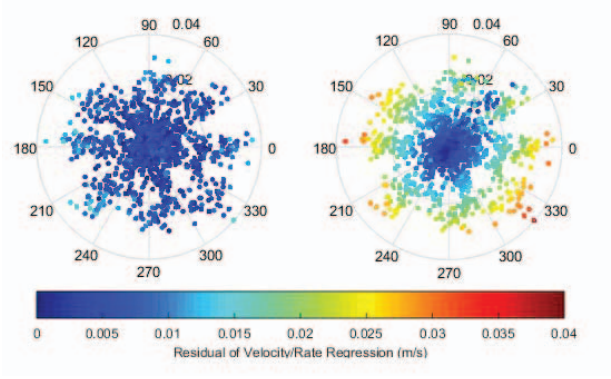

Fig. 5. State prediction error of ensemble prediction (left) and single neuron prediction (right) across angles and speed.

committees). Under the circumstance that each neuron provide consistent and comparable error variance in training, the advantage of the committee machine is not as pronounced, and the fewer co-variance distributions (one observation per state element in EOKF) may provide inferior estimates depending on data. The EOKF and standard KF have no inherent accuracy advantage over one another.

III. Haredware Architecture

A. Resource Optimization

The DSP architectural design focuses on resource sharing to minimize area and power. We exploit data sparsity inherent in neural spike activities to implement event driven hardware for resource sharing. All stages after parallel spike detections can share hardware without a separate fast clock domain, data stream mux, or additional controller for time multiplexing.

The on-chip hardware architecture of the proposed DSP system includes modules for spike detection, spike sorting, and the ensemble averaging part of the EOKF (Fig. 4).

We design 96 non-overlapping spike detectors and integrate within them simple feature extractors to deliver spike features directly to the sorter modules. We grouped 32 of the spike detectors to share a single set of sorter hardware. Each set of hardware still has own memory entries of boundaries and ensemble neuron identifiers. The spike waveform length, 32 samples in our datasets, determines the group size (32) because in the non-overlapping detection scheme, no channel can generate more than one spike event within 32 cycles.

At the event of spike, the detector enters an event token (i.e., spike features) onto a conveyer style queue (Fig. 6). The queue entry points from detectors have a simple stalling rule that gives priority to the token already on the conveyer, the stalled token then attempts to enter the queue in the subsequent clock cycles until a free spot is available. With the non-overlapping detection and the defined samples per spike, token stalling does not generate backpressure to earlier stages. The order of spike time is not preserved in the conveyer queue, as the order is irrelevant downstream where only spiking rate is considered. No spike is stalled beyond the time bin edges (nor would it be problematic as spike rate has relatively high Poisson noise) to affect the following rate computation.

The sorter modules, each responsible for 32 channels, consume event tokens in the order of channel array if no collision is present in the data streams, but in cases of token stalling, channel order is compromised. As a result, an address must be included in the token to retrieve correct boundary information used in sorting. The sorter checks the incoming features against boundaries stored in the Bayesian boundary memory. Note that we use the coding based on fixed segmentation patterns in the feature space (Sec. II). Thus, the outputs of the sorter are the addresses for the memory storing weights for decoding.

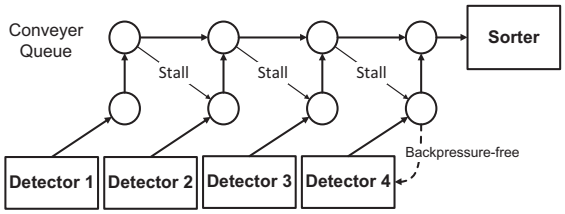

Fig. 6. Conveyer style queue of a 4-detector example

Finally, the architecture has a single ensemble-averaging module (memory and accumulator) does not require resource sharing since all data paths converge to a single register per state element in Ez_k. Instead of calculating spike rates and multiply them with weights (E), upon every spike event, we perform memory read for retrieving a coefficient and then accumulate it using a single adder. This is equivalent to multiply-accumulate since the instantaneous spiking rate is a count of spikes in a fixed-duration time bin. In place of the multiplier, each spike event immediately triggers an addition of its weight in the ensemble observation registers. This architecture can reduce silicon area and thus leakage power.

A typical challenge of event-driven implementation in place of a scheduled one is potential data collision hazards. In our architecture, we can have collisions among the three spike sorting modules when they try to access the ensemble-averaging module at the same time. Unlike spike detection, the sorter has no hardware constraint for token generation. With a finite amount of buffer memory, therefore, token loss is possible. Practically, however, token loss is improbable since only a small subset of all sorted neuron channels (e.g., 20 ~ 50 in typical experiments) is selected for intention decoding. In our test, no token loss or even collision occurs even with all sorted neuron channels considered valid. In the unlikely event

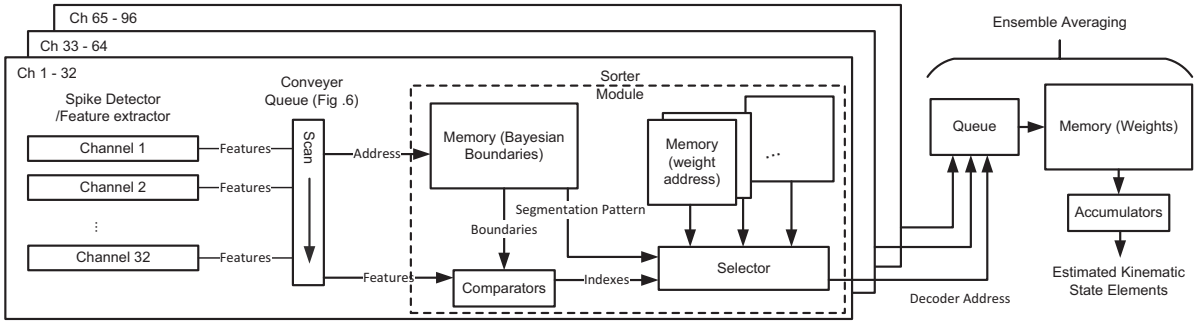

Fig. 7. Full architecture of the proposed BCI system DSP back-end

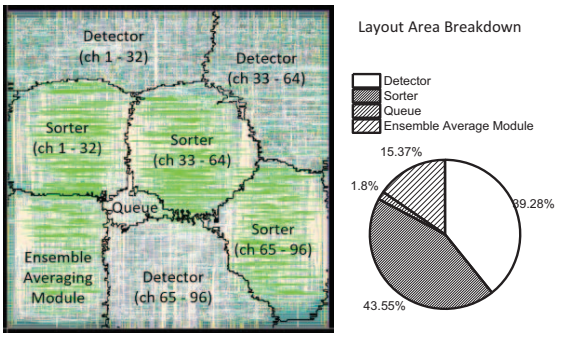

Fig. 8. On-chip DSP acclerator layout

that a token is lost, its effect is minimal since the ensemble-averaging module can easily tolerate small loss (see Sec. II).

B. VLSI Implementation

Our proposed DSP architecture (Fig. 7) is implemented in a 65nm. The previous work [7] identified the dominant power expenditure in intercellular spike decoding chips is leakage due to the low input rates. The DSP is, therefore, synthesized with high-V_t devices for minimizing leakage. The DSP has a single clock domain, its frequency matching that of the front-end ADC output (30kHz). The slow clock allows the use of near/sub-threshold V_{DD} for low power. To ensure the robustness of gates at the aggressively-scaled V_{DD}, only a subset of the industrial standard cell library that are found to be robust at near/sub-threshold V_{DD} are used for synthesis. We also perform physical design of the synthesized netlist using an APR tool. The layout photo is shown in Fig. 8. The area of the design is 0.16 mm^2 at the utilization ratio of 85.5%.

We perform the static timing analysis using the libraries characterized at multiple V_{DD}s and the interconnect parasitic information generated by the APR tool. The accuracy of the timing and power characterization flow is calibrated and confirmed by comparing its result to the results of SPICE simulation for benchmark circuits. At 0.3V and 30kHz clock frequency, the DSP consumes 4.82µW.

IV. Conclusion

In this work, we propose an algorithm-hardware co-design approach for the DSP back end for BCI prosthesis, optimizing for hardware and energy cost. Our design provides substantial resource savings from prior arts through algorithmic modifications. We verified our algorithm using data driven testing for spike sorting and intention decoding, and with additional boundedness analysis for the proposed ensemble observation model. We also devise event-driven hardware architecture, which enables to share resources and thus, substantially reduce area. This reduces leakage power dissipation, a dominant power expenditure in this kind of DSP due to low input rates. We implement the architecture in 65nm and perform post-layout delay and power analysis. Our DSP consumes single-digit microwatt power dissipation, which is very low and comparable to the power dissipation of some recent hardware that implements only a single channel sorter in the same technology [10].

References

[1] Serruya, M., Shaikhouni, A. and Donoghue, J.P., 2003. Neural decoding of cursor motion using a Kalman filter. In *Advances in Neural Information Processing Systems 15: Proceedings of the 2002 Conference* (Vol. 15, p. 133). MIT Press.

[2] Kim, S.P., Simeral, J.D., Hochberg, L.R., Donoghue, J.P. and Black, M.J., 2008. Neural control of computer cursor velocity by decoding motor cortical spiking activity in humans with tetraplegia. *Journal of neural engineering*, 5(4), p.455.

[3] Gilja, V., Nuyujukian, P., Chestek, C.A., Cunningham, J.P., Byron, M.Y., Fan, J.M., Churchland, M.M., Kaufman, M.T., Kao, J.C., Ryu, S.I. and Shenoy, K.V., 2012. A high-performance neural prosthesis enabled by control algorithm design. *Nature neuroscience*, 15(12), pp.1752-1757.

[4] Pineda, J.A., Allison, B.Z. and Vankov, A., 2000. The effects of self-movement, observation, and imagination on µ rhythms and readiness potentials (RP's): toward a brain-computer interface (BCI). *IEEE Transactions on Rehabilitation Engineering*, 8(2), pp.219-222.

[5] Mellinger, J., Schalk, G., Braun, C., Preissl, H., Rosenstiel, W., Birbaumer, N. and Kübler, A., 2007. An MEG-based brain–computer interface (BCI). *Neuroimage*, 36(3), pp.581-593.

[6] Jiang, Z., Cerqueira, J., Kim, J., Wang, Q., Seok, M., 2016. 1.74 - μ W/ch, 95.3% - accurate Spike sorting hardware based on Bayesian decision. *Symposium on VLSI Circuits* (SOVC), pp. 34-35.

[7] Karkare, V., Gibson, S. and Marković, D., 2013. A 75-µw, 16-channel neural spike-sorting processor with unsupervised clustering. *IEEE Journal of Solid-State Circuits*, 48(9), pp.2230-2238.

[8] Zumsteg, Z.S., Kemere, C., O'Driscoll, S., Santhanam, G., Ahmed, R.E., Shenoy, K.V. and Meng, T.H., 2005. Power feasibility of implantable digital spike sorting circuits for neural prosthetic systems. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 13(3), pp.272-279.

[9] Wang, Q., Webber, R.M. and Stanley, G.B., 2010. Thalamic synchrony and the adaptive gating of information flow to cortex. *Nature neuroscience*, 13(12), pp.1534-1541.

[10] Jiang, Z., Wang, Q. and Seok, M., 2015, June. A low power unsupervised spike sorting accelerator insensitive to clustering initialization in sub-optimal feature space. In *Proceedings of the 52nd Annual Design Automation Conference* (p. 174). ACM.

[11] Zeinolabedin, SMA., Do, AT., Jeon, D., Sylvester, D., and Kim, T., 2016. A 128-Channel Spike Sorting Processor Featuring 0.175 µW and 0.0033 mm2 per Channel in 65-nm CMOS, *Symposium on VLSI Circuits* (SOVC), pp. 32-33.

[12] Hauschild, M., Mulliken, G.H., Fineman, I., Loeb, G.E. and Andersen, R.A., 2012. Cognitive signals for brain–machine interfaces in posterior parietal cortex include continuous 3D trajectory commands. *Proceedings of the National Academy of Sciences*, 109(42), pp.17075-17080.

[13] Stevenson, I.H., Cherian, A., London, B.M., Sachs, N.A., Lindberg, E., Reimer, J., Slutzky, M.W., Hatsopoulos, N.G., Miller, L.E. and Kording, K.P., 2011. Statistical assessment of the stability of neural movement representations. *Journal of neurophysiology*, 106(2), pp.764-774.

[14] Walker, B. and Kording, K., 2013. The database for reaching experiments and models. *PloS one*, 8(11), p.e78747.