Learning-Based Near-Optimal Area-Power Trade-offs in Hardware Design for Neural Signal Acquisition

Cosimo Aprile
LIONS and LSM, EPFL
Lausanne, Switzerland
cosimo.aprile@epfl.ch

Luca Baldassarre
LIONS, EPFL
Lausanne, Switzerland

Vipul Gupta
LIONS, EPFL
Lausanne, Switzerland

Juhwan Yoo
Broadcom
United States of America

Mahsa Shoaran
MICS, Caltech
United States of America

Yusuf Leblebici
LSM, EPFL
Lausanne, Switzerland

Volkan Cevher
LIONS, EPFL
Lausanne, Switzerland

ABSTRACT

Wireless implantable devices capable of monitoring the electrical activity of the brain are becoming an important tool for understanding and potentially treating mental diseases such as epilepsy and depression. While such devices exist, it is still necessary to address several challenges to make them more practical in terms of area and power dissipation. In this work, we apply Learning Based Compressive Subsampling (LBCS) to tackle the power and area trade-offs in neural wireless devices. To this end, we propose a low-power and area-efficient system for neural signal acquisition which yields state-of-art compression rates up to $64\times$ with high reconstruction quality, as demonstrated on two human iEEG datasets. This new fully digital architecture handles one neural acquisition channel, with an area of $210 \times 210 \mu m$ in $90nm$ CMOS technology, and a power dissipation of only $1\mu W$.

Keywords

Neural signals, Compressive Sensing, digital signal processing, area-efficient, low-power, signal recovery.

1. INTRODUCTION

The microelectromechanical (MEMS) technology is opening new venues of applications in health-care with significant new possibilities. In recent years MEMS facilitated advances in wireless implantable devices and have enabled monitoring of biological signals in the human body, such as blood pressure, electrical activity of heart and brain, and so on. In particular, the capability of monitoring the brain activity captured the interest of many scientists for decades and from the 90s clinicians have begun to use implantable devices to observe the activity of the neurons [1]. Being able to efficiently interface the electrical system with the biological environment would enable patients with brain diseases (such as epilepsy or Parkinson) to be monitored and treated with minimally invasive systems.

According to the US National Institute of Neurological Disorders and Stroke, more than 50 million people worldwide [2] are affected by epilepsy and 25% of the patients are pharmaco-resistant. Since 1997, the usage of prostheses has been approved to provide medical treatments for some brain diseases, such as Parkinson and epilepsy, and in 2005 also for depression [3]. Data compression is needed for reducing the power consumption of data telemetry and favouring the biocompatibility of a small wireless implantable device.

In order to reduce the power requirements of data transmission, compressive sensing (CS) [4, 5] has been exploited in many recent approaches (e.g., [6, 7, 8] and references therein). In a nutshell, CS consists in taking fewer linear samples than dictated by the Shannon-Nyquist theorem, while still allowing robust off-line signal reconstruction. This is possible by exploiting the fact that the information content of a signal is often much lower than its raw data content.

In this work, we design a digital encoder for neuronal signals based on Learning Based Compressive Subsampling (LBCS) [9], which allows to reduce the chip's power and area requirements, while improving on the reconstruction performance. Such method is based on the simple idea of sampling a fixed set of coefficients that preserve as much of the signal's energy as possible. The set of indices is learnt from a training set of fully sampled signals, by selecting the ones that capture most of the signals' average energy. LBCS offers a pair of highly efficient linear encoder and decoder, thus challenging the conventional recovery approach in CS, where non-linear decoding procedures such as basis pursuit are necessary for reliable signal reconstructions.

In few words, our learning-based digital encoder scheme leverages the benefits of structured linear sampling and linear recovery to yield state-of-the-art compression perfor-

GLSVLSI '16, May 18-20, 2016, Boston, MA, USA

DOI: http://dx.doi.org/10.1145/2902961.2903028

mance, maintaining a high signal reconstruction quality up to 64× compression, as quantitatively demonstrated on two human iEEG datasets.

The paper is organized as follows. We introduce the main concepts of Compressive Sensing and Learning Based Compressive Subsampling in Section 2, and in Section 3 we describe the digital architecture tailored for LBCS. Numerical experiments are reported in Section 4, while in Section 5 we analyse and describe our circuit design. Section 6 concludes the paper.

2. COMPRESSION ALGORITHM

In this section, we first introduce the basics of Compressive Sensing, reviewing three recent approaches applied to neuronal signals. We then discuss non-linear structured recovery, before discussing Learning-Based Compressive Subsampling.

2.1 Compressive Sensing

The main tenet of Compressive Sensing states that a signal $\mathbf{x} \in \mathbb{R}^N$ which has K non-zero coefficients can be robustly recovered from only $M = \mathcal{O}(K \log \frac{N}{K})$ samples $\mathbf{y} \in \mathbb{R}^M$,

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{w} \,, \tag{1}$$

where $\mathbf{A}$ is a linear operator that either satisfies the Restricted Isometry Property (RIP) or is incoherent [10], and $\mathbf{w}$ accounts for measurement noise. Obviously, $\mathbf{y}$ offers a compressed version of $\mathbf{x}$. If we are able to directly sample $\mathbf{y}$, we save both on storage and communication power. Recovering $\mathbf{x}$, though, usually requires to solve a non-linear optimization problem. Nonetheless, recent advances in optimization have provided efficient algorithms that can scale to very large signals [11].

Theoretically, i.i.d. sub-Gaussian matrices are incoherent and also satisfy the RIP. Furthermore, they are universal, i.e., given an ortho-normal basis $\mathbf{\Phi}$ which allows for a sparser representation of a signal $\mathbf{x}$, the RIP or the incoherence of $\mathbf{A}\mathbf{\Phi}$ is the same as of the original $\mathbf{A}$ [10]. However, sub-Gaussian matrices are prohibitively expensive to use in practice, since they require $\mathcal{O}(MN)$ space and time.

More efficient types of sampling are being successfully used in real applications, such as subsampled fast transforms, like the Fast Fourier (FFT), the Discrete Cosine (DCT) or the Fast Walsh-Hadamard (FWHT) Transforms, which can be computed in $\mathcal{O}(N \log N)$ time.

The following three randomized sampling approaches, recently proposed for the compression of neural signals are very efficient on the sampling side, but require solving non-linear optimization problems to reconstruct the original signals.

- Bernoulli (BERN): uses a random Bernoulli $\{\pm 1\}$ matrix to sample each channel independently [7].
- Multi-Channel Sampling (MCS): the idea behind MCS [8] consists in taking random Bernoulli $\{0, 1\}$ combinations of the samples across all channels at a given time point i, that is

  $$\mathbf{y}_i = \mathbf{A}\mathbf{x}_i \,, \tag{2}$$

  where $\mathbf{A}$ is a $M \times \#ch$ Bernoulli matrix, with $\#ch$ being the number of channels, and $\mathbf{x}_i$ contains the values at time i for all channels. This allows to design a relatively simple encoder scheme, but requires the channels to be fairly correlated in order to faithfully reconstruct the signals, as further discussed in [12].
- Structured Hadamard Sampling (SHS): the method presented in [12] randomly samples the indices of the Fast Walsh-Hadamard Transform (FWHT) of each channel according to a probability function that favors low frequencies, which seem to carry most of signal's energy.

2.2 Structured recovery

Additional structures in the signal $\mathbf{x}$, such as interdependencies between its non-zero coefficients or constraints on its support, allow to reduce the number of samples required for exact or stable recovery (see [13] and [14]). Many of these structures can be encoded via linear inequalities that admit tight and tractable convex relaxations [15]. Interestingly, natural signals are often characterized by sparse and structured representations in time-frequency (or space-frequency) domains, such as provided by wavelets [16].

In order to reconstruct the original signal $\mathbf{x}$ from its compressive samples $\mathbf{y}$, most structured-sparsity methods resort to solving the following optimization problem on the wavelet coefficients $\boldsymbol{\alpha}$,

$$\begin{aligned} \underset{\boldsymbol{\alpha} \in \mathcal{A}}{\text{minimize}} \quad & f(\boldsymbol{\alpha}) \\ \text{subject to} \quad & \mathbf{A}\mathbf{\Phi}\boldsymbol{\alpha} - \mathbf{y} \in \mathcal{K} \end{aligned} \tag{3}$$

where f is a Gauge function that promotes the structure we expect in $\boldsymbol{\alpha}$, $\mathcal{K}$ encodes our information about the noise and $\mathcal{A}$ is a constraint set that specifies further assumptions about the signal, e.g. boundedness. We reconstruct the signal as $\widetilde{\mathbf{x}} = \mathbf{\Phi}\widetilde{\boldsymbol{\alpha}}$, where $\widetilde{\boldsymbol{\alpha}}$ is the solution to (3) and $\mathbf{\Phi}$ is the wavelet transformation matrix.

In [12], three different structured-sparsity recovery methods have been compared for reconstructing iEEG signals sampled via the SHS, MCS and BERN approaches. The best performance was obtained using a Gauge function that exploits the natural tree representation of the wavelets coefficients in order to penalize the coefficients closer to the tree leaves. Such an approach is called Hierarchical Group Lasso (HGL). In particular, they considered the above problem with $\mathcal{K} = \{0\}$, $\mathcal{A} = \mathbb{R}^N$ and

$$\|\mathbf{x}\|_{\mathcal{T}} := \sum_{\mathcal{G} \in \mathcal{T}} \|\mathbf{x}_{|_{\mathcal{G}}}\|, \tag{4}$$

where $\mathcal{T} = \{\mathcal{G}_1, \ldots, \mathcal{G}_N\}$ is a collection of index sets, each set $\mathcal{G}_i \subseteq \{1, \ldots, N\}$ containing the node i in the tree and all its descendants, see [17] for more details.

2.3 Learning Based Compressive Subsampling

The compression architecture that we propose in this paper is based on the idea of Learning-Based Compressive Subsampling (LBCS)[9], which consists on *linear encoding* and *linear decoding* with respect to a given orthonormal basis, resulting in a much simpler and faster solution compared to the approaches described in Section 2.1.

LBCS can be summarized as follows. Given a signal $\mathbf{x} \in \mathbb{R}^N$, we consider the compression model

$$\mathbf{y} = \mathbf{P}_\Omega \mathbf{\Psi} \mathbf{x} \,, \tag{5}$$

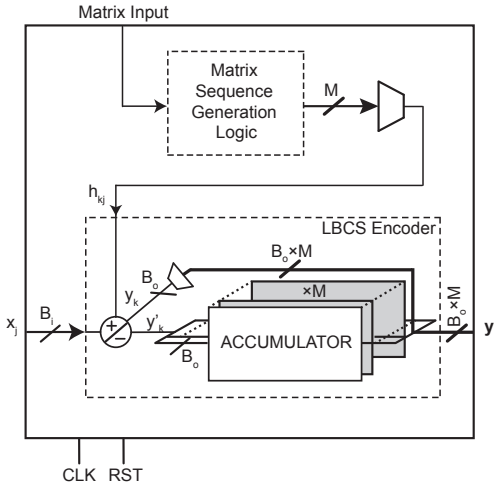

Figure 1: One channel block diagram showing the LBCS encoder and the matrix sequence generation logic.

where $\mathbf{\Psi} \in \mathbb{R}^{N\times N}$ is an orthonormal basis and $\mathbf{P}_\Omega \in \mathbb{R}^{M\times N}$ is a subsampling matrix whose rows are canonical basis vectors. The effect of applying $\mathbf{P}_\Omega$ to $\mathbf{\Psi x}$ is to retain only the coefficients indexed by the set Ω, also known as the *subsampling map*. The vector $\mathbf{y} \in \mathbb{R}^M$ is the compressed version of $\mathbf{x}$, with a nominal compression rate (CR) of $\frac{N}{M}$. The signal $\mathbf{x}$ is then approximately recovered via the fast linear decoder

$$\hat{\mathbf{x}} = \mathbf{\Psi}^* \mathbf{P}_\Omega^T \mathbf{y} \, . \tag{6}$$

Given a training set $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ of m fully sampled signals of unit norm, we learn the optimal subsampling map Ω by choosing the indices that capture most of the average energy in the transform domain:

$$\hat{\Omega} = \underset{\Omega, |\Omega|=M}{\arg\max} \frac{1}{m} \sum_{j=1}^{m} \sum_{i\in\Omega} |\langle \boldsymbol{\psi}_i, \mathbf{x}_j \rangle|^2, \tag{7}$$

where $\boldsymbol{\psi}_i$ is the i-th row of $\mathbf{\Psi}$. $\hat{\Omega}$ can be exactly found by selecting the M indices whose values of $\frac{1}{m}\sum_{j=1}^{m} |\langle \boldsymbol{\psi}_i, \mathbf{x}_j \rangle|^2$ are the largest [9]. The learnt sampling scheme is then used to directly sample only those transform coefficients indexed by $\hat{\Omega}$ for all signals $\mathbf{x}$.

2.4 Adaptive encoding

Given a basis $\mathbf{\Psi}$ and a desired number of samples M, the optimal linear encoding of each $\mathbf{x}$ is given by retaining only the M largest coefficients of $\mathbf{\Psi x}$ in absolute value. However, this adaptive encoding requires to first compute all the coefficients $\mathbf{\Psi x}$, which is prohibitive with small area and power consumption, as discussed in Section 5.1.

3. SYSTEM ARCHITECTURE

In this section, we propose the architecture to allow an embedded sampling and compression of the neural input signal based on the LBCS approach described in Section 2.3.

In the following, we fix $\mathbf{\Psi}$ equal to the Hadamard matrix $\mathbf{H}$ which has the advantage of only requiring a single bit to represent each matrix entry and also minimizes the matrix multiplication operations. Let $\mathbf{H}_\Omega = \mathbf{P}_\Omega \mathbf{H}$ be the matrix

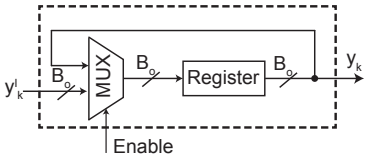

Figure 2: Accumulator block diagram.

composed of the rows of $\mathbf{H}$ indexed by Ω. We sequentially compute $\mathbf{y} = \mathbf{H}_\Omega \mathbf{x}$: looking at each component of $\mathbf{y}$, we have

$$y_k = \sum_{j=1}^{N} h_{kj} x_j, \; k \in \{1, \ldots, M\}, \tag{8}$$

where h_{kj} is the (k, j)-entry of $\mathbf{H}_\Omega$.

3.1 Sampling procedure

Figure 1 shows the block diagram of the LBCS architecture proposed in this work for one-channel sampling. The *Matrix Sequence Generator Logic* is a chip memory that stores the entries of $\mathbf{H}_\Omega$ that are used for the sub-sampling procedure performed by the *LBCS Encoder* block. The entries are stored into the chip memory in a sequential fashion through the *Matrix Input*. The sampling procedure starts once the memory is loaded and a *serializer* is used to sequentially send the h_{kj} weights to the summation node.

The input signal x_j is the digital output of an A/D converter with a resolution of B_i bits. At the beginning of each window of length N, we set $\mathbf{y} = 0$ and then, at each time step j, x_j is summed or subtracted to the B_o-bit accumulator value y_k depending on the one-bit Hadamard entry h_{kj}, updating each component via the rule:

$$y'_k = y_k + h_{kj} x_j, \; k \in \{1, \ldots, M\}, \tag{9}$$

Instead of performing the subtraction through a subtractor, the B_o-bit signal y'_k is formed with a single B_o-bit ripple carry adder, and the h_{kj} input defines the polarity of y_k. This also allows to avoid any multipliers in the weighting phase when y_k is fed-back to the summation node. Each accumulator has to be updated before the next sample x_j arrives, therefore we use an *enable* signal to drive the multiplexer of the accumulator block, shown in Figure 2, in order to update only one register per time. With this design choice, we avoid having one adder per accumulator lane, but require an internal digital clock frequency

$$f_{encoder} = M \times f_s, \tag{10}$$

where f_s is the signal sampling frequency[1] .

When $M = \frac{N}{CR}$ is large, the internal clock frequency may become a limiting factor, requiring additional digital blocks to synchronize the clock. However, as further described in Section 4, the sampling frequency is 5kHz for the considered datasets, choosing $N = 256$ and a hypothetical compression rate of 16×, the LBCS encoder freqency results to be 5kHz×$\frac{256}{16}$ = 80kHz, which is still in a relatively low frequency range.

[1]The h_{kj}-serializer works at frequency $f_{encoder}$ too.

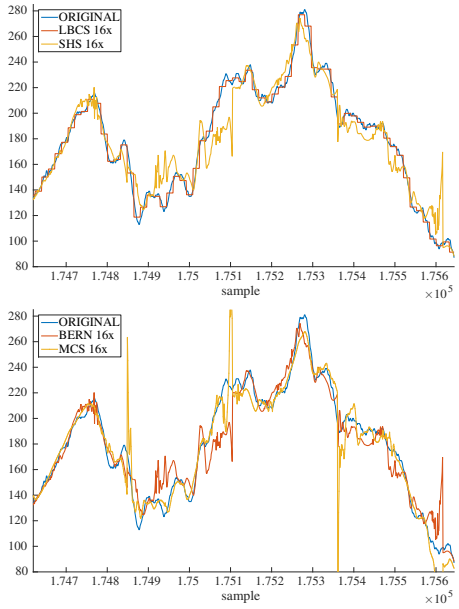

Figure 3: `I001-P034-D01` Reconstruction example for channel `Grid28` on four windows of length 256 each.

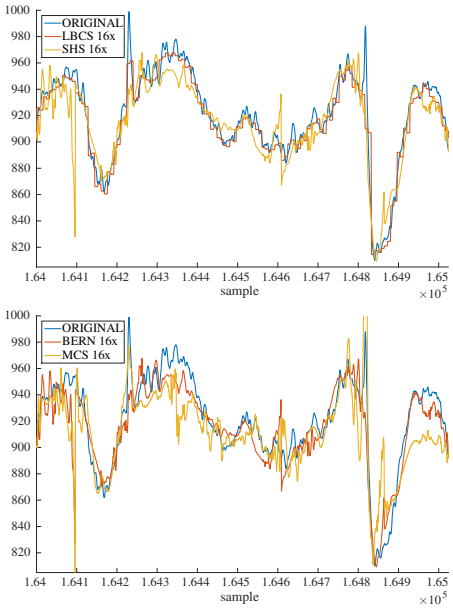

Figure 4: `Study 040` Reconstruction example for channel `LG50` on four windows of length 256 each.

4. SIMULATIONS

In this section, we first give the details related to the human iEEG datasets used in the experiments and then we compare the numerical results obtained applying the LBCS encoder against the other approaches described in Section 2.1.

4.1 Dataset details

The `iEEG.org` portal contains several datasets of EEG and iEEG data which are manually annotated by expert clinicians. We focus on the following two datasets.

4.1.1 `I001-P034-D01`

The `I001-P034-D01` dataset consists of approximately 1 day, 8 hours and 10 minutes of recordings at 5kHz, or approximately $6 \cdot 10^8$ samples. In order to reduce the dataset size, we use samples only from the 12th and 13th seizure, and an equal number of samples before the seizure onset, for training and testing respectively.

We consider the 32 active grid electrodes which, from a first visual inspection, more clearly show significant changes between the samples annotated as seizures from the rest. In order to better compare to the sampling strategy that combines samples across the channels (MCS), we consider only a sub-grid of 4×4 electrodes.

4.1.2 `Study 040`

The `Study 040` dataset consists of approximately 2 days, 23 hours and 50 minutes of recordings at 5kHz, or approximately $1.3 \cdot 10^9$ samples. In order to reduce the dataset size, we use samples only from the 1st and the 3rd seizure and an equal number of samples before the seizure onset, for training and testing respectively. We consider all the 64 active grid electrodes.

4.2 Experimental protocol

The training set of both datasets are used to learn the sampling pattern for the LBCS approach and also to tune the variable density parameters for the SHS method. Once the sampling pattern is fixed, LBCS uses it to compress all the signal windows in the test set. The reconstruction is then performed with the linear decoder (6). For the randomized methods, MCS, BERN and SHS, we draw 20 different sampling patterns from the relative distributions for each signal window in the test and reconstruct using the tree-based HGL norm (4), which was shown in [12] to yield the best results.

4.3 Performance Evaluation

We concatenate all reconstructed windows for each channel j together, forming the entire reconstructed signal, $\hat{\mathbf{x}}_j$ for the test seizure. We then compute the SNR for each channel as $\text{SNR}_j = 20 \log_{10} \left(\frac{\|\mathbf{x}_j\|_2}{\|\mathbf{x}_j - \hat{\mathbf{x}}_j\|_2} \right)$, where $\mathbf{x}_j$ is the recorded signal for channel j, and average these SNRs to obtain our final measure of performance, $\text{SNR} = \frac{1}{\#ch} \sum_{i=1}^{\#ch} \text{SNR}_j$. For the randomized methods, we also average over the 20 draws.

4.4 Numerical results

We conducted numerical experiments with all the methods described in this paper on both datasets. We varied the length of the signal window N, the number of bits, B_i, of the input A/D converter and the compression rate CR. We observed that the LBCS approach is not very sensitive to the window length N, therefore, for the sake of space, we present only results for $N = 256$ and $B_i = 10$ bits, which seemed to offer the best trade-off between reconstruction quality and area-power consumption, as further discussed in Section 5.1.

Tables 1 and 2 report the reconstruction quality, in dB, obtained on the `I001-P034-D01` and the `Study 040` datasets respectively. As expected, adaptive compression sets the upper limit on the achievable performance. LBCS offers the best reconstruction quality at any compression rate, with an

Table 1: `I001-P034-D01` $\mathbf{N = 256}$, $\mathrm{B_i} = \mathbf{10}$

Method	Compression rate					
	2	4	8	16	32	64
Adaptive	41.60	39.86	36.38	31.40	25.42	19.43
LBCS	**40.79**	**37.64**	**33.27**	**28.48**	**23.27**	**18.06**
SHS HGL	36.92	27.96	23.89	20.26	18.53	14.49
BERN HGL	37.48	26.69	20.49	16.87	13.53	11.15
MCS HGL	28.96	24.40	20.92	17.48	n.a.	n.a.

Table 2: `Study 040` $\mathbf{N = 256}$, $\mathrm{B_i} = \mathbf{10}$

Method	Compression rate					
	2	4	8	16	32	64
Adaptive	40.79	40.05	38.11	35.28	32.07	28.61
LBCS	**40.55**	**38.90**	**35.77**	**33.09**	**30.28**	**27.28**
SHS HGL	37.58	33.67	31.75	29.21	27.73	24.75
BERN HGL	38.23	33.57	29.59	26.62	24.03	22.08
MCS HGL	37.20	34.22	30.82	27.03	23.00	18.45

increase in the SNR of several dBs compared to the other methods. The SHS approach offers the second best performance, as its variable density is adapted to the signals, but still fails at capturing as much structure as LBCS. The BERN and MCS methods offer a much inferior performance at high compression rates, because imposing structure only during reconstruction does not fully compensate the limitations of their structure-unaware sampling mechanisms. Figures 3 and 4 show some reconstructions obtained with each method on both datasets. The LBCS reconstructions are much smoother and better follow the original signal.

The linear decoder (6) yields reconstructions at a fraction of the computational cost of the other methods. Indeed, solving a single optimization problem with the HGL norm, using `DecOpt` [11], requires on average approximately 0.1s, while the linear decoder requires only approximately 10^{-5}s for a 256 samples signal.

5. CIRCUIT DESIGN AND VALIDATION

In this section, we first analyze the difference in terms of area and power consumption between Hadamard-based adaptive and LBCS encoding. Afterwards we describe the implemented circuit.

5.1 Adaptive vs LBCS encoders

Section 2.4 describes that the best linear encoder, for a fixed compression rate, is given by adaptively sampling the coefficients that capture most of the energy of each signal. We now analyze the power and area costs for LBCS and adaptive encoding respectively.

5.1.1 LBCS Power and Area analysis

- *Power cost*: as shown in Figure 1, M B_o-bit accumulators are used to store the Hadamard coefficients. This leads to a dynamic power consumption of:

$$P_{LBCS} \propto M \cdot B_o \cdot f_s \cdot V_{DD}^2 \cdot C_{ref}, \qquad (11)$$

  where V_{DD} is the operating voltage of the digital block and C_{ref} is the reference capacitance defined by the technology.

- *Area cost*: since a single adder is used for sampling, the area of the digital encoder block is proportional to the number M of accumulators:

$$Area_{LBCS} \propto M. \qquad (12)$$

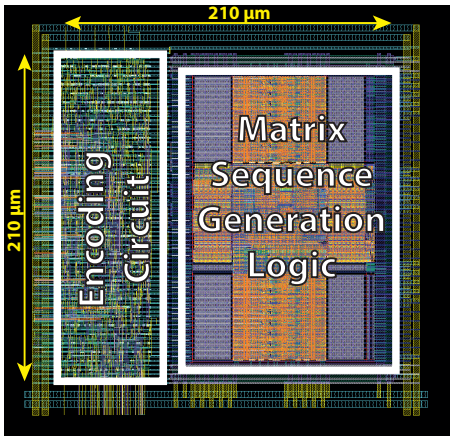

Figure 5: One channel encoder layout showing the LBCS encoding circuit and the matrix sequence generation logic for $N = 256$ and $CR = 16$.

5.1.2 Adaptive Power and Area analysis

- *Power cost*: considering a similar architecture, the adaptive encoder requires N accumulators, leading to a dynamic power consumption:

$$P_{Adaptive} \propto N \cdot B_o \cdot f_s \cdot V_{DD}^2 \cdot C_{ref}. \qquad (13)$$

- *Area cost*: the area cost is proportional to the number of accumulators used to store all the the Hadamard coefficients:

$$Area_{Adaptive} \propto N. \qquad (14)$$

5.1.3 Comparison

Comparing the area-power costs of the two approaches, we obtain

$$\frac{P_{Adaptive}}{P_{LBCS}} \geq \frac{N}{M} = CR,$$
$$\frac{Area_{Adaptive}}{Area_{LBCS}} \geq \frac{N}{M} = CR.$$

Combining these observations with Tables 1 and 2, we conclude that LBCS yields reconstructions almost as good as the ones obtained with the adaptive encoder, but at a fraction of its power and area cost. The advantage is more significant the higher the compression ratio.

5.2 Circuit implementation

To implement the proposed architecture, we have defined our target signal quality close to 30dB. Then, considering a sampling time window of 256 samples and assuming an ADC resolution $B_i = 10$ bits, we have set the compression ratio $CR = 16$ following the numerical results reported in Tables 1 and 2. The internal encoder core clock frequency is then $f_{encoder} = M \times f_s = 80$kHz and the accumulator resolution is set as $B_o = B_i + \log_2(N)$ to avoid overflow.

The architecture shown in Figure 1 has been implemented in a 1P9M 90 nm CMOS technology. The design is fully

Table 3: Comparison With Published Work

Parameter	[7]	[8]	[12]	This Work
Compression Method	BERN	MCS	SHS	LBCS
Compression Rate	10	16	16	16
Technology [μm CMOS]	0.09	0.18	-	0.09
Compression Power [μW]	1.9	17.83[a]	-	1.0
Compression Area [mm^2]	0.090	0.090	-	0.044
Recovered Signal [dB][b]	21.7	22.2	24.7	30.8

[a] Compression power cost over 16 channels.
[b] Average SNR calculated from Tables 1 and 2, considering CR=16 for all the compression methods.

digital and the layout of a one-channel encoder is shown in Figure 5. To verify the functionality of the digital encoder, the digitized neuronal data is directly given as input to the LBCS block. A post place-and-route simulation has verified that the M outputs given by the encoder are equal to the expected values computed in Matlab. The simulation has been run considering a worst case scenario with slow-slow process corner operating at 0.9V, which results in an estimated power consumption of the LBCS encoder around $1\mu W$. The silicon area of the encoder block is $210 \times 210\mu m$. Considering the fact that the electrode pitch in a typical Utah-MEA is $400\mu m$, the resulting size of the encoder is fully suitable for such embedded applications.

6. CONCLUSIONS

This work shows the application of LBCS to address the reduction of area and power costs for neural signal encoding and data telemetry in wireless implantable devices. The proposed scheme enables *on-the-fly* data compression with faster off-line recovery and higher reconstruction quality than random Bernoulli [7], multi-channel [8] or Structured Hadamard Sampling [12]. In Table 3 there is the performance summary and comparison with published works.

In the proposed design, the memory that stores the subsampled Hadamard matrix entries occupies a relative large area. In a multichannel implementation, the memory content is shared among all the channels, reducing the impact of the storage area over the overall chip area. Furthermore, we are currently studying how to generate the desired matrix entries directly in the chip.

LBCS is a general approach applicable to any sparse data acquisition system for which fully sampled signals are available. Our future work will focus on designing digital encoders for other applications like image processing.

Acknowledgment

This work was supported in part by the European Commission under grant ERC Future Proof and by the Swiss Science Foundation under grants SNF 200021-146750 and SNF CRSII2-147633. The authors would like to thank Jonathan Narinx for useful discussions on the system design.

7. REFERENCES

[1] A. C. Hoogerwerf and K. D. Wise, "A three-dimensional microelectrode array for chronic neural recording," *Biomedical Engineering, IEEE Transactions on*, vol. 41, no. 12, pp. 1136–1146, 1994.

[2] M. Leonardi and T. B. Ustun, "The global burden of epilepsy," *Epilepsia*, vol. 43, no. s6, pp. 21–25, 2002.

[3] C. B. Nemeroff, H. S. Mayberg, S. E. Krahl, J. McNamara, A. Frazer, T. R. Henry, M. S. George, D. S. Charney, and S. K. Brannan, "Vns therapy in treatment-resistant depression: clinical evidence and putative neurobiological mechanisms," *Neuropsychopharmacology*, vol. 31, no. 7, pp. 1345–1355, 2006.

[4] E. J. Candès, "Compressive sampling," in *Proceedings oh the International Congress of Mathematicians: Madrid, August 22-30, 2006: invited lectures*, 2006, pp. 1433–1452.

[5] D. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[6] J. N. Laska, S. Kirolos, M. F. Duarte, T. S. Ragheb, R. G. Baraniuk, and Y. Massoud, "Theory and implementation of an analog-to-information converter using random demodulation," in *IEEE International Symposium on Circuits and Systems*, 2007, pp. 1959–1962.

[7] F. Chen, A. P. Chandrakasan, and V. M. Stojanovic, "Design and analysis of a hardware-efficient compressed sensing architecture for data compression in wireless sensors," *IEEE Journal of Solid-State Circuits*, vol. 47, no. 3, pp. 744–756, 2012.

[8] M. Shoaran, M. H. Kamal, C. Pollo, P. Vandergheynst, and A. Schmid, "Compact low-power cortical recording architecture for compressive multichannel data acquisition," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 8, no. 6, pp. 857–870, December 2014.

[9] L. Baldassarre, Y.-H. Li, J. Scarlett, B. Gözcü, I. Bogunovic, and V. Cevher, "Learning-based compressive subsampling," *arXiv preprint arXiv:1510.06188*, 2015.

[10] S. Foucart and H. Rauhut, *A mathematical introduction to compressive sensing*. Springer, 2013.

[11] Q. Tran-Dinh and V. Cevher, "A primal-dual algorithmic framework for constrained convex minimization," *arXiv preprint arXiv:1406.5403*, 2014.

[12] L. Baldassarre, C. Aprile, M. Shoaran, Y. Leblebici, and V. Cevher, "Structured sampling and recovery of ieeg signals," in *6th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, 2015.

[13] R. Baraniuk, V. Cevher, M. Duarte, , and C. Hegde, "Model-based compressive sensing," *IEEE Transactions on Information Theory*, vol. 56, no. 4, pp. 1982–2001, 2010.

[14] A. Kyrillidis, L. Baldassarre, M. El Halabi, Q. Tran-Dinh, and V. Cevher, "Structured sparsity: Discrete and convex approaches," in *Compressed Sensing and its Applications*. Springer, 2015, pp. 341–387.

[15] M. E. Halabi and V. Cevher, "A totally unimodular view of structured sparsity," in *AISTATS*, 2015.

[16] S. Mallat, *A wavelet tour of signal processing*. Academic press, 1999.

[17] R. Jenatton, J. Mairal, G. Obozinski, and F. Bach, "Proximal methods for hierarchical sparse coding," *Journal of Machine Learning Reasearch*, vol. 12, pp. 2297–2334, 2011.