Balancing Specialized Versus Flexible Computation in Brain–Computer Interfaces

Ioannis Karageorgos and Karthik Sriram, *Yale University, New Haven, CT, 06520, USA*

Ján Veselý, *Rutgers—The State University of New Jersey, New Brunswick, NJ, 08901-8554, USA, and Yale University, New Haven, CT, 06520, USA*

Nick Lindsay and Xiayuan Wen, *Yale University, New Haven, CT, 06520, USA*

Michael Wu, *Rutgers—The State University of New Jersey, New Brunswick, NJ, 08901-8554, USA*

Marc Powell, *University of Pittsburgh, Pittsburgh, PA, USA*

David Borton, *Brown University, Providence, RI, 02912, USA*

Rajit Manohar and Abhishek Bhattacharjee, *Yale University, New Haven, CT, 06520, USA*

We are building HALO, a flexible ultralow-power processing architecture for implantable brain– computer interfaces (BCIs) that directly communicate with biological neurons in real time. This article discusses the rigid power, performance, and flexibility tradeoffs that BCI designers must balance, and how we overcome them via HALO's palette of domain-specific hardware accelerators, general-purpose microcontroller, and configurable interconnect. Our evaluations using neuronal data collected* in vivo *from a nonhuman primate, along with full-stack algorithm to chip codesign, show that HALO achieves flexibility and superior performance per watt versus existing implantable BCIs.

By enabling direct brain-computer communication, brain–computer interfaces (BCIs) can accelerate the process of scientific discovery, restore sensory capabilities, mitigate symptoms of movement disorders like Parkinson's disease, treat pharmacologically resistant depression and anxiety, and even restore motor capabilities for spinal cord injury, brain strokes, and amyotropic lateral sclerosis.[1–3] BCIs interrogate biological neurons and decode pathological behavior or the user's intent, guiding stimulation of the brain to mitigate seizures, control prostheses, actuate assistive devices, and more. BCIs have even been shown to augment human capabilities; e.g., enhancing short-term memory capacity, monitoring attention and mental state to enhance performance, navigating augmented realities via signals from the motor cortex, and reading signals from the visual cortex to infer words, pictures, and videos.[4] Consequently, Facebook and Microsoft are competing with Neuralink, Kernel, Neuropace, and Medtronic to build BCIs that read/stimulate an ever-increasing number of biological neurons with high signal fidelity.[5]

Modern BCIs designs are of two types. While some are noninvasive in the form of headsets or other external devices,[1] invasive BCIs surgically implanted on, around, and in the brain tissue are able to record and stimulate large numbers of neurons with higher signal fidelity, spatial resolution, and tighter real-time characteristics.[6] Low-power hardware for onboard processing is critical to the success of implantable BCIs, especially because elevating tissue temperature by just 1° can damage the brain's cellular structure.[7]

CHALLENGES OF BCI DESIGN

BCI applications must read the electrophysiological activity of as many biological neurons as possible with high spatial and temporal resolution to be useful. Modern BCIs extract neuronal activity at data rates of 10–50 Mbps, with Neuralink demonstrating even orders of magnitude higher data rates,[5] and DARPA's NESD program targeting communication with millions of neurons.[8] These large volumes of data may need to be processed in real time. For

0272-1732 © 2021 IEEE
Digital Object Identifier 10.1109/MM.2021.3065455
Date of publication 11 March 2021; date of current version 25 May 2021.

TABLE 1. Existing commercial and research BCIs meet target power budgets by either restricting their scope to a single use case, or by dropping brain–computer communication bandwidth. HALO is the first flexible implantable BCI architecture to overcome this tradeoff.

	Medtronic	Neuropace	Aziz	Kassiri	Neuralink	NURIP	HALO
	2	2	10	2	5	11	
Tasks supported							
Spike detection	×	×	×	×	×	×	✓
Compression	×	×	✓	×	×	×	✓
Seizure prediction	×	✓	×	✓	×	✓	✓
Movement intent	✓	×	×	×	×	×	✓
Encryption	×	×	×	×	×	×	✓
Technical capabilities							
Programmable	✓	Limited	×	✓	×	Limited	✓
Read channels	4	8	256	24	3072	32	96
Data rate (Mbps)	0.01	0.02	9.76	1.32	545	0.13	46
Safety (<15 mW)	✓	✓	✓	✓	×	✓	✓

example, BCIs that treat seizures must read the activity of biological neurons, process it to detect signs of a seizure or its imminent arrival, predict the movement of the seizure through different brain regions, determine where to apply electrical stimulus (and for how long) to mitigate seizure symptoms, and then stimulate brain tissue.[9] All of this must be done accurately, necessitating significant signal processing of many neuronal channels of data, and quickly, within milliseconds of seizure detection.[9] At the same time, BCIs may not exceed 15 mW for safe chronic implantation, a target that is notoriously challenging given the high data rates that BCIs must support.

BCI APPLICATIONS MUST READ THE ELECTROPHYSIOLOGICAL ACTIVITY OF AS MANY BIOLOGICAL NEURONS AS POSSIBLE WITH HIGH SPATIAL AND TEMPORAL RESOLUTION TO BE USEFUL.

Designers have responded by building BCIs that either achieve power efficiency via specialization for a restricted set of applications/treatments for specific disorders in specific brain regions, or more flexible multiuse designs that achieve power efficiency but only by restricting the number of neurons they read/stimulate. Consequently, the modern BCI ecosystem is fragmented, with many different single-use devices, and lacks standardization of computational capabilities. Table 1 captures this predicament by summarizing the limitations of the current state-of-the-art commercial and research BCIs.

OUR APPROACH: THE HALO PROJECT

An ideal BCI must be flexible as its operation may need to be personalized, there may be multiple neurological conditions to treat, and several brain–computer interactions to support. In response, we are building HALO, a high-performance, ultralow-power, and flexible BCI processing architecture. HALO is a full-stack design effort that uses electrophysiological data collected *in vivo* from a nonhuman primate's motor cortex (specifically, from the regions responsible for arm and leg movement) to evaluate a BCI architecture that balances a palette of power-efficient accelerators with configurable dataflow to support frequently used neural processing kernels. We are realizing HALO via several tape-outs, with Figure 1(a) illustrating a chip diagram of our HALO architecture in a 12-nm technology, after augmenting over the 28-nm technology from the original paper.[3] Furthermore, Figure 1(b) shows how the processing architecture integrates with the remainder of a typical implantable BCI device.

In realizing HALO, we make several research contributions. First, we systematically map the design space of BCI applications to identify a list of target capabilities to support. Because commercial BCIs are generally single-use devices, identification of a canonical set of applications that more flexible BCIs should strive to support has hitherto remained unanswered. This list includes disease

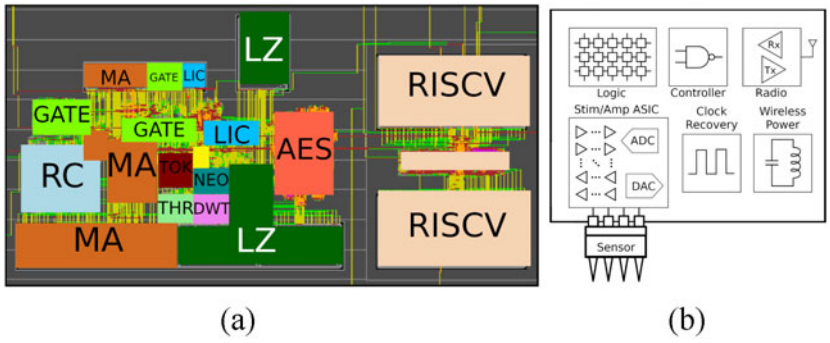

FIGURE 1. Chip diagram on the left shows our HALO tape-out in a 12-nm technology. The block diagram on the right shows other key components of implantable BCIs, including the sensors, which consists of conductive needles that penetrate millimeters of cortical tissue, analog components, a radio, and power sources. Implantable BCIs are packaged in a hermetically fused silica capsule or titanium capsule.

treatment, signal processing, and secure transmission of neuronal data (e.g., compression and encryption of extracellular voltage streams). As BCIs are an active area of research, this list is nonexhaustive. Nevertheless, it identifies a broader set of tasks needed for a flexible BCI platform as a starting point, while also offering a viable path to integrate these tasks.

Second, we navigate a large design space of architecture and integration options to realize this list of BCI capabilities by using principled hardware–software codesign. Standard low-power design dictates that we realize one accelerator per BCI task in the form of a dedicated ASIC. We refer to this as a monolithic ASIC design, and find that they often exceed the 15 mW power budget. In response, we refactor the underlying algorithm of the original BCI tasks into distinct pieces that realize different phases of the algorithm. We refer to these pieces as kernels, and show that they facilitate design of ultralow-power hardware processing elements (PEs) via novel hardware–software codesign approaches. We round out the design with a low-power RISC-V microcontroller to configure PEs into processing pipelines and support computation for which there are currently no PEs. The result is an unconventional style of heterogeneity, where a family of accelerator PEs, each of which is identified in our chip tape-out diagram in Figure 1(a), operates in unrelated clock domains with low-power asynchronous circuit-switched communication.

Third, we devise several hardware–software codesign techniques that raise the level of abstraction of BCI design from "bits and wires" to architectural choices that take inspiration from the world of software engineering. Table 2 summarizes these techniques, which we discuss in the next section. These approaches enable HALO to achieve 4–57× and 2× lower power dissipation than software alternatives and monolithic ASICs, respectively.

TABLE 2. Overview of hardware–software codesign techniques used to realize HALO.

Technique	Direction
Kernel PE decomposition	SW→ HW
PE reuse generalization	SW→ HW
PE locality refactoring	SW← HW
Spatial reprogramming	SW← HW
Counter saturation	SW↔ HW
NoC route selection	SW→ HW

COMPUTATIONAL TASKS SUPPORTED BY HALO

Figure 2 presents an overview of the HALO architecture. The block diagram on the left shows the PEs in our design and the configurable interconnect used to assemble PEs to realize the task pipelines shown on the right. HALO supports BCI tasks ranging from those that require real-time closed-loop support for treatment of neurological disorders to those that exfiltrate neural recordings to external systems for postprocessing and batch analysis. The first category consists of support for seizure treatment and amelioration of movement disorders. Seizure prediction/stimulation pipelines that break neuronal feedback loops are responsible for seizure severity present cutting-edge capabilities of FDA-approved clinical BCIs. So do algorithms to detect/stimulate the brain to counteract movement disorders associated with essential tremor and Parkinson's disease. HALO supports FFTs, cross-correlation, and bandpass filters over linear models to support closed-loop treatment of these neurological disorders.

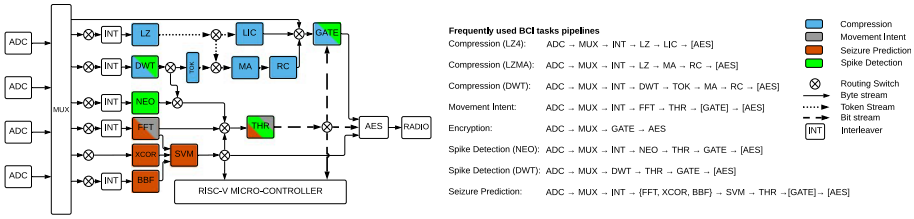

FIGURE 2. HALO consists of low-power hardware PEs and a RISC-V microcontroller. The PEs are configured into pipelines to realize tasks ranging from compression (in blue) to spike detection (in green). Optional PEs (e.g., AES encryption) are shown in square brackets. PEs operating in parallel (e.g., FFT, XCOR, and BBF for seizure prediction) are shown in curly brackets.

The second workload category includes compression to reduce radio transmission bandwidth. Apart from some specific and well-understood forms of lossy compression—such as spike sorting—BCIs generally require *lossless* compression. HALO supports spike detection via the near-energy operator (NEO) PE, and also implements several lossless compression variants as their effectiveness can change across brain regions and patient activity. We support lossless LZ4 and LZMA compression, as well as discrete wavelet transform (DWT) compression. Compression ratios vary by as much as 40% depending on compression algorithm and target brain region.[3]

Finally, although state-of-the-art BCIs do not currently support encryption, we foresee it as necessary in future BCIs for secure data exfiltration. HIPAA, NIST, and NSA require using AES with an encryption key of at least 128 bits.

HALO ARCHITECTURE

HALO supports five tasks, and can set up two of them in multiple ways, leading to a total of eight distinct pipelines configurable by a doctor or technician. The RISC-V microcontroller is used to configure these pipelines via the programmable switches. With the conventional monolithic ASIC approach, this means that we would implement eight ASICs. However, we decompose these pipelines into the PEs of Figure 2.

Decomposing BCI Tasks Into PEs

Kernel PE Decomposition: Some BCI tasks consist of distinct computational kernels naturally amenable to PE decomposition. For example, seizure prediction combines kernels for FFT, cross-correlation (XCOR), Butterworth bandpass filtering (BBF), and a support vector machine (SVM). We realize each as a PE, as shown in Figure 2. As FFT, XCOR, and BBF have no data dependencies, they can operate in parallel. This approach saves power because XCOR contains complex computation (e.g., divisions and square roots) that scales quadratically with channel count. In contrast, BBF is a simple filter with minimal arithmetic that scales linearly with channel count. Separating XCOR and BBF into separate PEs ensures that BBF's filtering logic is clocked over an order of magnitude slower than the logic for cross-correlation.

PE Reuse Generalization: Many BCI tasks use computational kernels slightly differently. We develop configurable PEs that can be shared among BCI tasks. Consider movement intent, which can be decomposed into FFT, followed by logic that checks whether the FFT output is in a particular spectral range. We create a threshold PE (THR) to determine when a PE's output is within a specified numerical range and enable sharing of the FFT between movement intent and seizure prediction tasks. The FFT PE is configurable because movement intent requires 14–25-point FFTs to detect drops in signal power, while seizure prediction requires 1024-point FFTs.

Algorithm 1. LZMA pseudocode

```
Function LZMA_compress_block(input)
    output = list(lzma_header);
    while data = input.get() do
        best_match = find_best_match(data);
        Prob_match = count(table_match, best_match)
            /count_total(table_match);
        r1 = range_encode(Prob_match);
        output.push_back(r1);
        increment_counter(table_match, best_match);
    end while
    Return output;
end function
```

Major Refactoring: PE decomposition can require significant refactoring of the original algorithm. Consider LZMA and DWTMA compression. Both algorithms use Markov (MA) chains to calculate the probability of the current input value based on observed history, which is used to pick more efficient encoding of the input signal. We found that using the

combined MA PE overshoots the 15-mW power budget. To solve this problem, we refactored the original algorithm to make it more amenable for PE decomposition. To separate algorithmic phases, we realize that data locality (i.e., following routines that manipulate major data structures) is a good indicator of kernel boundaries within programs. This observation is tied to the fact that PEs in HALO have only local memories and cannot share large amounts of data. *Locality refactoring* highlights how design decisions about the architecture (i.e., use of PE-local memories) guided refactoring of our algorithms.

Algorithm 1 demonstrates how we use this insight to change LZMA. The second half of this algorithm can be separated into probability calculations and frequency information updates centered around the maintenance of the core MA data structure, the frequency table (in green), as well as efficient encoding (in blue). This refactoring permits bringing together phases that operate on the same data structures within the hardware, allowing us to separate the PEs since they can now operate independently with minimal data movement. This permits clocking each component at significantly lower frequency, leading to power savings of 2×.

PE Optimizations

Unchanged PE Output: Some of the PEs (e.g., XCOR and LZ) process data in blocks instead of samples and wait for all inputs in the block to arrive, before computing in a bursty manner. Bursty computation is problematic as it requires either large buffers to sink the bursts or high PE frequency to meet data rates while sustaining periods of bursty activity. Neither is ideal from the perspective of saving power. To achieve power improvements, we *spatially reprogram* the original algorithm and codesign it with the hardware. Consider the XCOR PE. The original algorithm performs computation at the end once all data have been filled into the block. We refactor the algorithm to process inputs as early as they are available. The final form in Algorithm 2 reduces the amount of computation needed in the final step, as well as the number of buffers needed to store the inputs. This translates to a power savings of 2.2× over the original algorithm. This technique also extends to other PEs like LZ to achieve 1.5× power reduction.

Finally, LZ and MA PEs require initialization of data structures at the beginning of every compressed block. We found that dedicated circuits are necessary to meet the 15 mW power budget. These circuits use only combinational logic and reduce PE power consumption by 1.8×.

Algorithm 2. XCOR spatial programming refactoring

```
1:  function XCOR(input, output)
2:    // channel[][] stores input in appropriate channel location
3:    channel[channel_num][sample_num] = input
4:    // data[] stores sums of input received so far
5:    data[count] += input
6:    // data_lag[] stores sums of input till LAG
7:    If count_2 == LAG then
8:      data_lag[count] = data[count]
9:    end if
10:   // Finish correlation computation
11:   if channel.filled() then
12:     for each i, j ∈ channels do
13:       avg_i = data[i]/SIZE
14:       avg_j = (data[j] − data_lag[j])/SIZE
15:       output.push_back(avg_i, avg_j)
16:     end for
17:     return output
18:   end if
19: end function
```

Modified PE Output: Although initialization circuits decrease the direct power/performance cost of starting a new compression block, there is also an indirect cost of using uninitialized internal structures, which leads to lower compression rates. This presents a problem with respect to the choice of block size. Large block sizes lead to better estimates of frequencies, but small block sizes allow the use of smaller data types and reduce the memory footprint and power of the MA PE. One might balance power/compression ratio for an ideal design, but such an approach does not find a design point that fits within the constrained power budget. Instead, we observe that the frequencies of values within a block remain largely unchanged after they have stabilized. Consequently, we allow the frequency counters to saturate and set block size independently of counter bit width. Overall, *counter saturation* modification allows HALO to benefit both from reduced memory footprint of 16 bit counters, and better compression ratio of larger blocks.

On-Chip Network

Each PE operates at the lowest frequency needed for data processing rates, and synthesize with established synchronous design flows. While running PEs in separate clock domains saves power, it can potentially complicate inter-PE communication. Prior work on globally asynchronous locally synchronous (GALS) architectures[12] encountered these issues for packet-switched on-chip networks. Unfortunately, we cannot repurpose their solutions as our analysis with the DSENT tool

estimates that a simple packet-switched mesh network consumes over 50 mW, well over our 15-mW power budget. Instead, we codesign inter-PE communication with the BCI algorithms. The decomposition of BCI tasks into kernels creates static and well-defined data-flows between PEs. *NoC route selection* allows replacement of a packet-switched network to a far lower power circuit-switched network built on an asynchronous communication fabric.

FIFO Interfaces

Since the publication of our original paper on HALO,[3] one challenge that we encountered is that our GALS-based approach requires careful data rate matching between PEs in separate clock domains. We use per-PE FIFO buffers to transfer data from the network into the form expected by the PE and to perform this rate matching. Consider PEs *f* and *g*. If their computation is regular—i.e., the functions produce and consume data in a perfectly periodic fashion—then a simple interface between the two for clock domain conversion suffices. However, if *f* produces bursty data or *g* consumes data in bursts, then a FIFO is needed between *f* and *g* to smooth out producer–consumer patterns. The size of this FIFO is determined by the computational properties of *f* and *g*, and the frequencies at which they operate. Increasing the frequency of *g* beyond the minimum operating point to meet data throughput needs would reduce the FIFO size required. We have found that balancing FIFO size with PE frequency is key to meeting the 15-mW power budget.

EVALUATION

Our 15-mW target power budget includes the HALO chip, sensors, ADC, amplifier, and radio technologies. We assume a microelectrode array with 96 channels, each of which records each sample encoded in 16 bits at a frequency of 30 kHz, yielding a data rate of 46 Mbps. After accounting for all analog components, HALO's processing pipelines (including the radio) must consume no more than 12 mW. All results presented use a commercial 28-nm fully depleted silicon-on-insulator (FD-SOI) CMOS process except when noted otherwise. Synthesis and power analysis is performed using Cadence synthesis tools with standard cell libraries from STMicroelectronics.

We use electrophysiological data collected from the brain of a non-human primate. Microelectrode arrays were implanted in two locations in the motor cortex, corresponding to the left upper and lower limbs. We use recordings of brain activity while the animal performed tasks such as walking on a treadmill, reaching for a treat, and overcoming a moving styrofoam obstacle. All research protocols were approved and monitored by Brown University's Institutional Animal Care and Use Committee, and all research was performed in accordance with relevant guidelines and regulations.

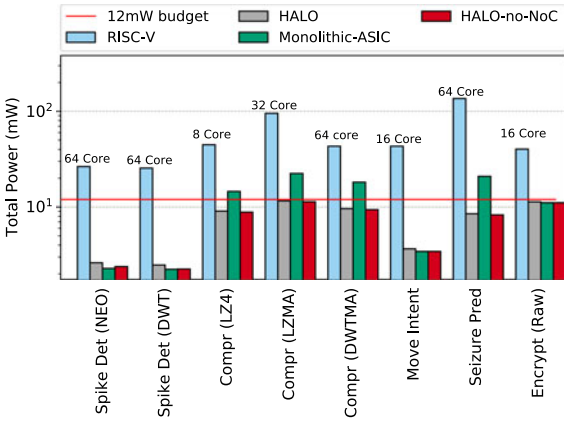

FIGURE 3. Power (in log-scale) of PEs, control logic, and radios for HALO versus RISC-V and monolithic ASICs. To meet the 15-mW device power budget, these components (without ADCs and amplifiers) need to be under 12 mW (the red line). We compare HALO against the lowest power RISC-V and HALO-no-NoC, which shows how much power would be saved if HALO's configurability were sacrificed.

HALO PRESENTS A WET LAB-TO-CHIP DESIGN PROJECT THAT EXPLORES THE QUESTION OF HOW TO BUILD A FLEXIBLE ULTRALOW-POWER PROCESSING ARCHITECTURE FOR NEXT-GENERATION BCIS.

Figure 3 compares HALO's power versus ASICs and software alternatives on RISC-V. Software tasks can execute on microcontroller cores in both single-core and multicore designs, where we divide the 96 channel data streams and operate on them in parallel. We study 1–64 RISC-V core counts and report the best configuration per task. We also show an idealized version of HALO where the on-chip interconnect is removed to quantify the power penalty for the configurability that the network offers. Both HALO variants use the optimizations from the ones described in the "PE Optimization" section. HALO uses less power than monolithic ASICs and RISC-V approaches.

Finally, as we have been extending our chip design efforts, we have discovered the crucial impact of FIFO design on total power. For each PE, we have evaluated the power utilized for various configurations with frequency and input and output FIFO buffers. For each frequency, we select the lowest FIFO size required for the design, and report its power. For example, for the LIC PE, we have found that the lowest power configuration is achieved at 24 MHz, with an 8-entry input FIFO and no output FIFO. We also note that power consumed can vary by as much as 1 mW depending on these configuration options.

CONCLUSION

HALO presents a wet lab-to-chip design project that explores the question of how to build a flexible ultra-low-power processing architecture for next-generation BCIs. While this work performs an initial exploration of workloads that are important for neuroscience, the list of tasks can be expanded. Future BCIs will implement other workloads, with different pipelines targeting different research and medical objectives. Because of its modular design, HALO will be able to support such workloads seamlessly.

REFERENCES

1. A. L. S. Ferreira, L. C. D. Miranda, and E. E. Cunha de Miranda, "A survey of interactive systems based on brain-computer interfaces," *SBC J. Interactive Syst.*, vol. 4, no. 1, pp. 3–13, 2013, doi: 10.5753/jis.2013.623.
2. H. Kassiri *et al.*, "Closed-loop neurostimulators: A survey and a seizure-predicting design example for intractable epilepsy treatment," *IEEE Trans. Biomed. Circuits Syst.*, vol. 11, no. 5, pp. 1026–1040, Oct. 2017, doi: 10.1109/TBCAS.2017.2694638.
3. I. K. *et al.*, "Hardware-software codesign for brain-computer interfaces," in *Proc. 47th Annu. Int. Symp. Comput. Archit.*, 2020, pp. 391–404, doi: 10.1109/ISCA45697.2020.00041.
4. C. Cinel, D. Valeriani, and R. Poli, "Neurotechnologies for human cognitive augmentation: Current state of the art and future prospects," *Front., Hum. Neurosci.*, vol. 13, 2019, Art. no. 13, doi: 10.3389/fnhum.2019.00013.
5. E. Musk and Neuralink, "An integrated brain-machine interface platform with thousands of channels," *J. Med. Internet Res.*, vol. 21, no. 10, 2019, Paper e16194, doi: 10.1101/703801.
6. I. Stevenson and K. Kording, "How advances in neural recording affect data analysis," *Nature Neurosci.*, vol. 14, pp. 139–42, 2011, doi: 10.1038/nn.2731.
7. P. D. Wolf, "Thermal considerations for the design of an implanted cortical brain-machine interface (BMI)," *Indwelling Neural Implants: Strategies for Contending With the in Vivo Environment*, W. M. Reichert, Eds. Boca Raton (FL): CRC Press/Taylor & Francis, 2008, ch. 3, doi: 10.1201/9781420009309-11.
8. DARPA, Bridging the Bio-Electronic Divide Accessed: Aug. 10, 2019. [Online]. Available: https://www.darpa.mil/news-events/2015-01-19.
9. S. Li, W. Zhou, Q. Yuan, and Y. Liu, "Seizure prediction using spike rate of intracranial EEG," *IEEE Trans. Neural Syst.*, vol. 21, no. 6, pp. 880–886, Nov. 2013, doi: 10.1109/TNSRE.2013.2282153.
10. J. N. Y. Aziz *et al.*, "256-Channel neural recording and delta compression microsystem with 3D electrodes," *IEEE J. Solid-State Circuits*, vol. 44, no. 3, pp. 995–1005, Mar. 2009, doi: 10.1109/JSSC.2008.2010997.
11. G. O'Leary *et al.*, "NURIP: Neural interface processor for brain-state classification and programmable-waveform neurostimulation," *IEEE J. Solid-State Circuits*, vol. 53, no. 11, pp. 3150–3162, Nov. 2018, doi: 10.1109/JSSC.2018.2869579.
12. M. Krstic and E. Grass, "New GALS technique for datapath architectures," in *Proc. Int. Workshop Power Timing Modeling, Optim. Simulation*, 2003, pp. 161–170, doi: 10.1007/978-3-540-39762-5_18.

IOANNIS KARAGEORGOS is currently an associate research scientist at Yale University, New Haven, CT, USA. His primary research interests focuses on the general area of VLSI, including GALS architectures, logical and physical SoC/ASIC design, and DTCO. Karageorgos received a Ph.D. degree in electrical engineering from KU Leuven and IMEC, Belgium. Contact him at ikarageo@aya.yale.edu.

KARTHIK SRIRAM is currently a Ph.D. student at Yale University, New Haven, CT, USA. His research interests include computer systems and architecture, hardware-software codesign, specially in the design of brain–computer interfaces. Sriram received a B.S. degree in computer science from Rutgers University. He is the corresponding author of this article. Contact him at karthik.sriram@yale.edu.

JÁN VESELÝ is currently a software engineer with Nvidia. His interests span the areas of architecture, operating systems, and compiler techniques for accelerators. Veselý contributed to this work while he was a visiting student at Yale. Veselý graduated in 2021 from Rutgers University. His thesis focused on hardware and software methods of integrating accelerators into heterogeneous systems. Contact him at jan.vesely@rutgers.edu.

NICK LINDSAY is currently a graduate student at Yale University, New Haven, CT, USA. His interests include building secure, safe, and high-performance heterogeneous systems. Lindsay received a B.Eng. degree in electrical engineering degree from the University of Glasgow. Contact him at Nick.Lindsay@yale.edu.

XIAYUAN WEN is currently a Ph.D. student at Yale University, New Haven, CT, USA. Her research interests include computer architecture and circuit design. Wen received the B.S. degree from Nanjing University and an M.S. degree from Yale University. Contact her at xiayuan.wen@yale.edu.

MICHAEL WU is an incoming Ph.D. student at Yale University, New Haven, CT, USA. His research focuses on the applications of machine learning in computer systems. Wu received a bachelor's degree in computer science from Rutgers University, New Brunswick, NJ, USA. Contact him at mw811@cs.rutgers.edu.

MARC POWELL is currently a postdoctoral associate with the Department of Neurological Surgery, University of Pittsburgh. His research focuses on the development of advanced neural technologies designed to provide unprecedented access to the nervous system and apply these tools to the treatment of neurological dysfunction caused by injury or disease. A major goal of his work is to facilitate the clinical translation of these devices and ensure that they are implemented safely and reliably. Powell received a bachelor's degree in biomedical engineering from Georgia Institute of Technology in 2014 and a Ph.D. degree in biomedical engineering from Brown University in 2021. Contact him at marc_powell@pitt.edu.

DAVID BORTON is currently an assistant professor of biomedical engineering at the Brown University School of Engineering, the Carney Institute for Brain Science, and is also a biomedical engineer at the Providence Veterans Affairs Center for Neurorestoration and Neurotechnology, New Haven, CT, USA. He leads an interdisciplinary team of researchers focused on the design, development, and deployment of novel neural recording and stimulation technologies. His team leverages engineering principles to untangle the underpinnings of sensorimotor and neuropsychiatric disease and injury. Borton received a B.S. degree in biomedical engineering from Washington University in St. Louis in 2006 and a Ph.D. degree in bioengineering from Brown University in 2012. He was a Marie Curie Postdoctoral Fellow at the Ecole Polytechnique Frale de Lausanne. Contact him at david_borton@brown.edu.

RAJIT MANOHAR is currently a John C. Malone Professor of Electrical Engineering and a professor of computer science at Yale University, New Haven, CT, USA. His research focuses on the design and implementation of asynchronous circuits and systems. Manohar received a Ph.D. degree in computer science from Caltech. Contact him at rajit.manohar@yale.edu.

ABHISHEK BHATTACHARJEE is currently an associate professor of computer science at Yale University, New Haven, CT, USA. His research focuses on computer architecture and systems at all scales of computing, ranging from server systems for large-scale data centers to embedded systems for implantable brain– computer interfaces. Bhattacharjee received a bachelor's degree in engineering from McGill University in 2005 and a Ph.D. from Princeton University in 2010. Contact him at abhishek@cs.yale.edu.