This article has been accepted for publication in IEEE Micro. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/MM.2023.3258907

HALO: A Hardware-Software Co-Designed Processor for Brain-Computer Interfaces

Ioannis Karageorgos*, Karthik Sriram*, Xiayuan Wen*, Ján Veselý, Nick Lindsay, Michael Wu, Lenny Khazan, Raghavendra Pradyumna Pothukuchi, Rajit Manohar, Abhishek Bhattacharjee

Yale University

***Abstract*—Brain-computer interfaces (BCIs) enable direct communication with the brain, providing valuable information about brain function and enabling novel treatment of brain disorders. Our group has been building HALO, a flexible and ultra-low-power processing architecture for BCIs. HALO can process up to 46 Mbps of neural data, a significant increase over the interfacing bandwidth achievable by prior BCIs. HALO can also be programmed to support several applications, unlike most prior BCIs. Key to HALO's effectiveness is a hardware accelerator cluster, where each accelerator operates within its own clock domain. A configurable interconnect connects the accelerators to create data flow pipelines that realize neural signal processing algorithms. We have taped out our design in a 12 nm CMOS process. The resulting chip runs at 0.88 V, per-accelerator frequencies of 3–180 MHz, and consumes at most 6.3 mW for each signal processing pipeline. Evaluations using electrophysiological data collected from a non-human primate confirm HALO's flexibility and superior performance per watt.**

I. Introduction

BCIs directly sense and stimulate electrical activity of neurons in the brain, enabling a new approach to increasing our understanding of the brain, treating drug-resistant epilepsy, restoring motor capabilities in individuals suffering from neurological disorders, and more [1–4]. BCIs are also heralding innovation in improving mental focus, short-term memory, mind-controlled assistive devices, and more. Consequently, companies like Meta, Microsoft, Neuralink, Kernel, Neuropace, Synchron, Paradromics and Medtronic are building BCIs that read, process, and stimulate increasingly more neurons with the highest signal fidelity.

BCIs can be realized as non-invasive headsets, or, as invasive devices where the electrodes to sense/stimulate neurons are implanted in or around brain tissue surgically. Our work focuses on the latter, which can record and stimulate a large population of neurons with high fidelity [5], and have important clinical, research and therapeutic uses.

Conflicting constraints make it challenging to design processors for invasive BCIs. On the one hand, BCIs must process increasing volumes of neural data in real-time. For example, BCIs that treat seizures must process neural activity to detect signs of a current or impending seizure, determine where and how to apply electrical stimulus to mitigate the seizure, and apply the stimulus, all within a few milliseconds [6]. Some BCIs can read neuronal activity at 10s of Mbps, recent experimental designs claim even higher rates [7], and DARPA's NESD program targets reading millions of neurons at Gbps data [8]. All this data must be analyzed in real time.

On the other hand, BCIs cannot overheat brain tissue by more than 1 °C. In general, BCI vendors target power consumption under 15 mW for safe permanent implantation.

Current BCIs have adopted the approach of rigid specialization to a particular application, or sacrificing data rates to support multiple applications. Consequently, the BCI landscape is fragmented with many single-use or low capability devices. Table I captures this predicament using a representative list of state-of-art commercial and research BCIs.

	Medtronic [2]	Neuropace [2]	Aziz [9]	Kassiri [2]	Neuralink [7]	NURIP [10]	**HALO**
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 (<15mW)	✓	✓	✓	✓	×	✓	✓

TABLE I: 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.

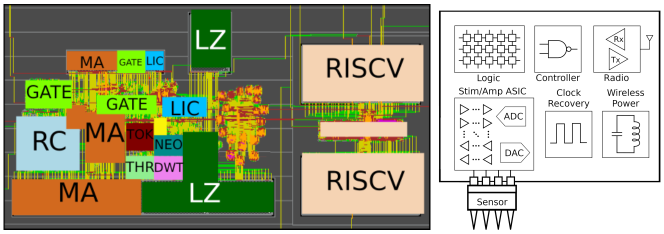
II. The HALO Project

Our goal is to build a BCI processor that can process high neural data rates *and* supports many BCI applications, while meeting the power constraints needed for safe long-term implantation. The outcome of our research is HALO, a BCI processor that has a family of accelerator processing elements (PEs), each operating in separate clock domains with low-power asynchronous circuit-switched communication. Figure 1a shows the chip diagram of a 12 nm CMOS tape out of HALO. Figure 1b shows how HALOintegrates with the remainder of a typical implantable BCI device.

HALO's design is unconventional in many ways. Standard low power design dictates that we realize one accelerator per

*Joint first authors who have contributed to this work equally. Authors are listed in alphabetical order of last name.

This article has been accepted for publication in IEEE Micro. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/MM.2023.3258907



(a) (b)

Fig. 1: The chip diagram on the left shows our HALO tape-out in a 12 nm technology. Per-PE labels show the distinct logic and memory components of the PE that are placed in different physical locations. 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.

BCI application in the form of a dedicated ASIC (which we refer to as a monolithic ASIC). We find that monolithic ASICs exceed the permitted power budget, and do not achieve our desired flexibility in hardware design.

Instead, HALO realizes both flexibility and low power operation. We begin by systematically mapping the design space of BCI applications to identify the target capabilities we wish to support. These include disease treatment, signal processing, and secure transmission of neuronal data (*e.g.,* compression and encryption). While these capabilities are not exhaustive, we identify them to be the broad features required for a flexible multi-use BCI platform.

Next, we refactor the underlying algorithm of the BCI applications into distinct pieces or kernels that realize different phases of the algorithm. The kernels facilitate the design of modular, ultra-low-power hardware processing elements (PEs). By bundling logic with similar complexity within individual PEs, we are able to clock the module at the lowest frequency required to sustain bandwidth and reduce power. We complete the design by including a low-power RISC-V microcontroller to configure PEs into processing pipelines and support computation for which there are no PEs.

Finally, we devise several hardware-software co-design techniques described in Section IV, which optimize the design at the abstraction level of the PEs. These techniques enable HALO to achieve 4-57$\times$ and 2$\times$ lower power dissipation than software and monolithic ASIC implementations, respectively.

HALO's top-down, modular approach provides another important design benefit. It allows us to be agile, and tape out the design with incremental functionality. We evaluate our tape-outs using electrophysiological data collected from a non-human primate's motor cortex. We originally synthesized HALO in 28 nm, and have later synthesized and taped out several modules in 12 nm.

III. 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. The PEs are assembled into the task pipelines shown on the right, by using a configurable interconnect.

HALO supports multiple types of applications. The first category consists of support for seizure treatment and mitigating movement disorders. Seizure prediction/stimulation pipelines are part of the state-of-the-art BCIs approved for clinical use by the U. S. Food and Drug Administration (FDA) [11]. Similarly, algorithms to detect/stimulate the brain to counteract movement disorders associated with essential tremor and Parkinson's disease are under FDA approved trials. HALO supports FFT, cross-correlation, and bandpass filters over linear models to support closed loop treatment of these neurological disorders.

The second category includes compression to reduce radio transmission bandwidth. BCIs generally require *lossless* compression, except in specific scenarios like spike sorting. HALO supports spike detection using the near energy operator (NEO) PE, and implements several lossless compression variants since the best choice of the compression algorithm varies 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, HALO supports encryption with the AES PE.. No existing BCI supports encryption, but we foresee it as becoming necessary in future BCIs for secure data exfiltration. HALO's encryption PE is designed according to standards like HIPAA, NIST, and NSA that require using AES with an encryption key of at least 128 bits.

IV. The 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 clinician. With the conventional monolithic ASIC approach, we would have required eight ASICs. Instead, we decompose the pipelines into reusable PEs, shown in Figure 2. A RISC-V microcontroller is used to configure the PEs into pipelines via programmable switches.

A. 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. This approach saves power because XCOR contains complex computation (e.g., divisions, square roots) that scales quadratically with the number of inputs. In contrast, BBF is a simple filter with minimal arithmetic that scales linearly with the input count. Separating XCOR and BBF into separate PEs ensures that BBF's filtering

This article has been accepted for publication in IEEE Micro. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/MM.2023.3258907

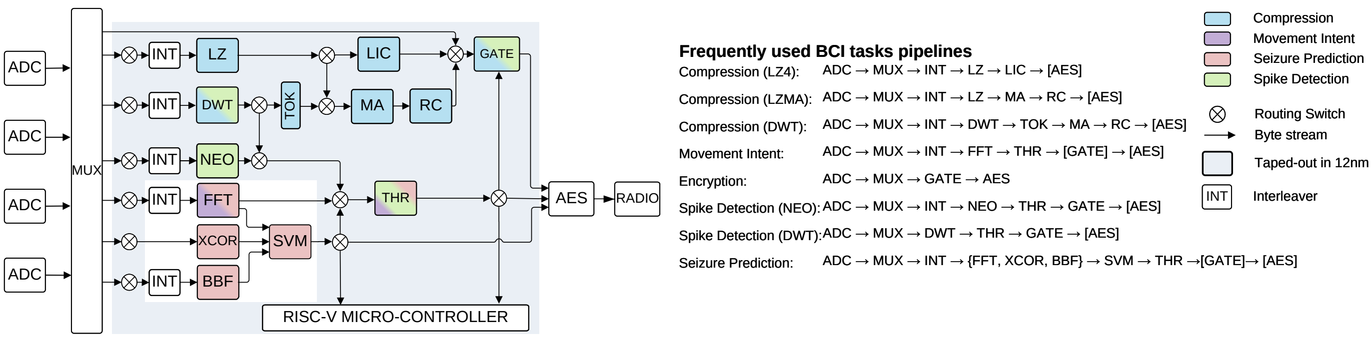


Fig. 2: HALO consists of low-power hardware PEs and a RISC-V micro-controller. The PEs are configured into pipelines to realize tasks ranging from compression (in blue) to spike detection (in green). PEs taped-out in the latest 12nm technology node are shown within the grey background. Optional PEs (e.g., 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.

logic is clocked over an order of magnitude slower than the logic for cross-correlation.

PE reuse generalization: Multiple BCI tasks like movement intent and seizure prediction often share the same computational kernel, such as FFT, but with different configurations (e.g., the FFT resolution). We make our PEs configurable to increase their reuse across applications.

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 is more effective if the original algorithms are refactored. Consider LZMA and DWTMA compression. Both algorithms compute the frequency of data values to encode them efficiently. However, we found that using one PE for all operations overshoots the 15 mW power budget. Therefore, we refactored the original algorithm. We identify that data locality of functions manipulating major data structures is a good indicator of kernel boundaries. This observation is also tied to the fact that PEs in HALO have only local memories and cannot share large amounts of data. We call this approach, *Locality Refactoring*.

Algorithm 1 demonstrates how locality refactoring is used to modify the LZMA application. 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 brings together phases that operate on the same data structures, allowing us to separate distinct such sections into PEs. Each PE can then be clocked at a significantly lower frequency, leading to 2× power savings over a design that combines all operations.

B. Processing Element Optimizations

Unchanged PE output: Some PEs (e.g., XCOR, LZ) process data in blocks instead of samples and wait for all inputs in the block to arrive. This bursty computation is problematic as it requires either large buffers to sink the outputs of computations or running the destination PEs at high frequency to meet data rates. Both approaches waste power. Therefore, when possible, we *spatially reprogram* the original algorithm and co-design it with the hardware. Consider the XCOR PE. The original algorithm waits for all data to arrive before operating on it, but we refactor it to process inputs as they arrive. 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.

Modified PE output: When possible, we modify the PE outputs to save energy without losing accuracy. Consider the data block size used in compression. 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. 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.

C. On-Chip Network

Each PE operates at the lowest frequency needed for data processing rates, and is synthesized with established synchronous design flows. While running PEs in separate clock domains saves power, it can potentially complicate inter-PE

This article has been accepted for publication in IEEE Micro. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/MM.2023.3258907

communication. Prior work on globally asynchronous locally synchronous (GALS) architectures [12] encountered these issues for packet-switched on-chip networks. Unfortunately, we estimate that a simple packet-switched mesh network consumes over 50ṁw, excluding such designs. Instead, we co-design 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.

D. Choice of FIFO Buffer Design

Despite optimizations, FIFO buffers are necessary at the output of some of HALO's bursty PEs, *e.g.,* PEs in the compression pipeline. Reducing the buffer sizes is important to reduce power, especially for our 12 nm tape-outs. We achieve this by first increasing the frequency of the PE that reads from the buffer, beyond the rate necessary to sustain the data-processing rate. We select the optimal frequency and FIFO buffer size by studying the power tradeoff between the higher frequency of the PE and the lower size of the FIFO. We show this tradeoff in Section VI-B. Next, for FIFOs larger than 1 KB, we use SRAM instead of registers, since SRAM consumes less power than registers.

V. Synthesis

Our 15 mW target power budget includes the HALO chip, sensors, ADC, amplifier, and radio. 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. We present results for our original evaluation at 28 nm Fully-Depleted Silicon-On-Insulator (FD-SOI) CMOS process as well as our augmented evaluation for our tape-out at 12 nm (which includes accurate estimates for the interconnect). Synthesis and power analysis is performed using the latest generation of Cadence® synthesis tools with standard cell libraries from STMicroelectronics.

Table II shows the synthesis results from 28 nm, and 12 nm. Typically, a lower process node facilitates using a lower frequency to sustain a given data processing rate, since the gate delays are lower. However, Table II shows that several PEs have a higher frequency at 12 nm. This was necessary to optimize the FIFO buffer size (Section IV-D), and is especially noticeable for the inherently bursty PEs (LZ, DWT, MA, RC).

Algorithm 2 XCOR spatial programming refactoring

function XCOR($input, output$)
  // channel[][] stores input in appropriate channel location
  $channel[channel\_num][sample\_num] = input$
  // data[] stores sums of input received so far
  $data[count] += input$
  // data_lag[] stores sums of input till LAG
  if $count\_2 == LAG$ **then**
    $data\_lag[count] = data[count]$
  end if
  // Finish correlation computation
  if $channel.filled()$ **then**
    for each $i, j \in channels$ **do**
      $avg\_i = data[i]/SIZE$
      $avg\_j = (data[j] - data\_lag[j])/SIZE$
      $output.push\_back(avg\_i, avg\_j)$
    end for
    return output
  end if
end function

PE	28 nm				12 nm			
	Freq (MHz)	Logic (mW)	Mem (mW)	Area (KGE)	Freq (MHz)	Logic (mW)	Mem (mW)	Area (KGE)
LZ	129	1.51	1.56	55	155	1.17	0.59	158
LIC	22.5	0.32	0.05	25	25	0.16	0.01	20
MA	92	2.28	1.06	66	180	1.87	0.62	274
RC	90	0.79	0	12	60	2.08	0	40
DWT	3	0.01	0	2	36	0.07	0.00	3
TOK	6	0.01	0	1	8	0.06	0.00	4
NEO	3	0.02	0	5	14	0.04	0.00	3
THR	16	0.01	0	1	3.5	0.04	0.00	4
GATE	5	0.01	0.12	17	42	0.34	0.04	28
RISCV	25	0.48	1.38	70	25	0.26	0.28	297
FFT	15.7	0.57	0.44	22	-	-	-	-
XCOR	85	4.25	0.36	81	-	-	-	-
BBF	6	0.10	0	23	-	-	-	-
SVM	3	0.04	0.11	8	-	-	-	-
AES	5	0.11	0	34	-	-	-	-

TABLE II: Frequency, power, and area characteristics of our 28nm and 12nm HALO variants.

VI. Evaluation

We use a physical synthesis flow for 28nm and 12nm technology nodes. A subset of our evaluations (*i.e.,* compression analysis) use brain data from a non-human primate collected by the Borton Lab at Brown University as per our original HALO paper [3].

A. Power Consumption

Figure 3 compares HALO's power at 28 nm and 12 nm with the monolithic ASIC approach, and another approach that runs the applications on a RISC-V processor. Software tasks on RISC-V can execute sequentially or in parallel, where the 96 electrode data streams are split between the multiple cores. We study core counts from 1 to 64 and report the outcome of the best configuration per task. HALO uses less power than monolithic ASICs and RISC-V approaches, and is the only design within the power limit of 15 mW for all applications.

B. Power Trade-off in the FIFO Buffer Design

Bursty PEs require large FIFOs to buffer data till the PE can accept it. We show this trade-off between using a large FIFO buffer versus increasing the frequency of the PE using the MA module in the LZ-MA-RC pipeline.

Figure 4 shows the total power consumed by the MA-RC segment of the compression pipeline, split into the power consumed by the FIFO buffer, and the PE compute, as the

This article has been accepted for publication in IEEE Micro. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/MM.2023.3258907

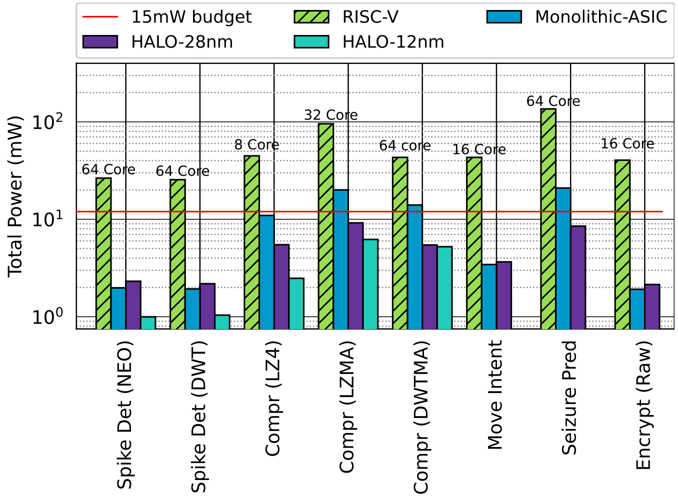


Fig. 3: Power (in log-scale) of PEs, control logic and radios for HALO versus RISC-V and monolithic ASICs. To meet the 15 mW device power budget, these components (without ADCs and amplifiers) need to be under 12 mW (the red line). We compare HALO against the lowest-power RISC-V and Monolithic-ASIC, the standard approach to low power design. HALO-28 nm shows our original evaluation for 28 nm process node. HALO-12 nm shows the evaluation for 12 nm process node with accurate power analysis for interconnect.

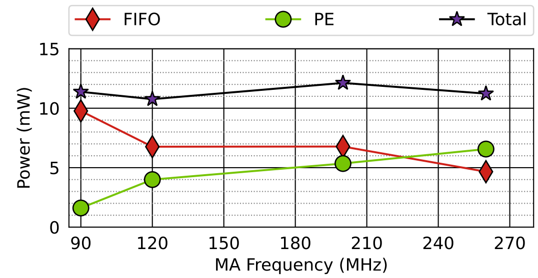


Fig. 4: Power of MA-RC components divided into FIFO and PE power for the LZMA pipeline. As MA frequency increases, FIFO size and power decreases. Correspondingly, PE power increases. The total power is minimized at 120 MHz.

frequency of MA is varied. MA must run at 90 MHz to process the input rate of 46 Mbps. Figure 4 shows that the power consumed by the FIFO buffer decreases as frequency is increased. With a higher frequency, the PE can process inputs faster, reducing the buffer time, and consequently, the size of the buffer. However, a higher frequency increases the dynamic power of not only MA but also for the subsequent PEs, *i.e.* RC, to sustain the increased dataflow rate. The figure shows that the overall power is lowest when MA operates at 120 MHz, which is 33% higher than the minimum frequency required to sustain the input datarate.

We perform a similar analysis for all PEs with bursty datarates, considering all pipelines they are part of. For example, from Figure 2, MA is in another compression pipeline with DWT, and optimizing the power of that pipeline yields a frequency of 180 MHz, which we finally use for MA.

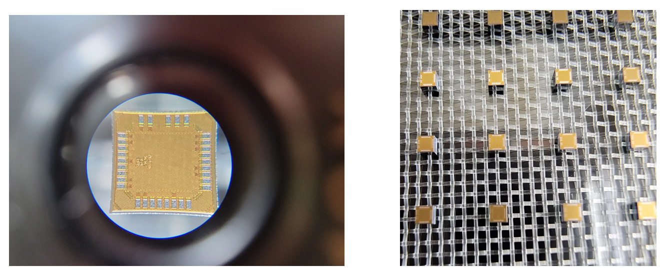
VII. Agile Prototyping

HALO is an unconventional BCI processor, and we follow an agile approach to tape out and verify it incrementally. Our first tape out only includes the RISC-V processor that we develop entirely in-house. This chip has an area of about 297 KGE (kilo gate equivalent), normalized to the cell area of a 2-input NAND gate.

Next, we add some pipelines that are relatively easy to verify because their signal processing is simpler (*i.e.,* spike detection) and some, which are more complex (*i.e.,* compression). Figure 1a showed this layout, and has an area of of 832 KGE. We tape out two versions of this design. One exposes the RISC-V interface, the PE interfaces that carry reconfiguration commands, and PE internal memory, externally for testing and debugging. The other variant has these connections internal, as they would be in the final system.

HALO has been designed in a modular manner from the beginning to support such an agile workflow (Section II).

Figure 5 shows the initial dies we received from the foundry. They operate at 0.88 V with overall dimensions $< 1\,\text{mm}^2$.



(a) Single die. (b) Multiple dies from a wafer.

Fig. 5: Chips from our first tape-out in 12 nm technology node.

We will complete additional tape outs to include all our PEs, and then package the chips with the remaining components of the BCI: sensors and stimulation units, ADC, radio, and a power source (Figure 1b). Along with neuroscientists, we plan on evaluating the performance and safety of the final package in vivo using animal studies.

VIII. 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, but 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.

This article has been accepted for publication in IEEE Micro. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/MM.2023.3258907

IX. Acknowledgements

This work was partially supported by the National Science Foundation (awards 2019529, 2118851, 1815718, and 2040682). This work was also supported by a grant from the Swebilius Foundation. The authors also gratefully acknowledge Muhammed Ugur, Gabriel Petrov, Oliver Ye, Michal Gerasimiuk, Hitten Zaveri, Dennis Spencer, Nick Turk-Browne, and Imran Quraishi for their feedback on this work.

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 Journal on 3D Interactive Systems*, vol. 4, no. 1, 2013.

[2] Hossein Kassiri, Sana Tonekaboni, M. Tariqus Salam, Nima Soltani, karim Abdelhalim, Jose Luis Perez Velasquez, Roman Genov, "Closed-Loop Neurostimulators: A Survey and A Seizure-Predicting Design Example for Intractable Epilepsy Treatment," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 11, no. 5, pp. 1026–1040, 2017.

[3] I. Karageorgos, K. Sriram, J. Vesely, M. Powell, D. Borton, R. Manohar, and A. Bhattacharjee, "Hardware-Software Co-Design for Brain-Computer Interfaces," *International Symposium on Computer Architecture (ISCA)*, 2020.

[4] I. Karageorgos, K. Sriram, J. Veselý, N. Lindsay, X. Wen, M. Wu, M. Powell, D. Borton, R. Manohar, and A. Bhattacharjee, "Balancing specialized versus flexible computation in brain–computer interfaces," *IEEE Micro*, vol. 41, no. 3, pp. 87–94, 2021.

[5] I. Stevenson and K. Kording, "How Advances in Neural Recording Affect Data Analysis," *Nature neuroscience*, vol. 14, pp. 139–42, 02 2011.

[6] S. Li, W. Zhou, Q. Yuan, and Y. Liu, "Seizure Prediction Using Spike Rate of Intracranial EEG," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 21, pp. 880–886, Nov 2013.

[7] E. Musk and Neuralink, "An Integrated Brain-Machine Interface Platform with Thousands of Channels," *bioRxiv*, 2019.

[8] DARPA, "Bridging the Bio-Electronic Divide (January 1, 2016) Retrieved August 10, 2019 from https://www.darpa.mil/news-events/2015-01-19,"

[9] J. N. Y. Aziz, K. Abdelhalim, R. Shulyzki, R. Genov, B. L. Bardakjian, M. Derchansky, D. Serletis, and P. L. Carlen, "256-Channel Neural Recording and Delta Compression Microsystem With 3D Electrodes," *IEEE Journal of Solid-State Circuits*, vol. 44, pp. 995–1005, March 2009.

[10] G. O'Leary and D. M. Groppe and T. A. Valiante and N. Verma and R. Genov, "Nurip: Neural interface processor for brain-state classification and programmable-waveform neurostimulation," *IEEE Journal of Solid State Circuits*, vol. 53, 2018.

[11] F. T. Sun and M. J. Morrell, "Closed-loop neurostimulation: the clinical experience," *Neurotherapeutics*, vol. 11, no. 3, pp. 553–563, 2014.

[12] M. Krstic and E. Grass, "New GALS Technique for Datapath Architectures," in *PATMOS*, 2003.

PLACE PHOTO HERE

Ioannis Karageorgos is an R&D Engineer at Blue Cheetah and a Research Associate at Yale University. His primary research interests are in the general area of VLSI, including GALS architectures, logical and physical SoC/ASIC design, and DTCO. Ioannis received the Ph.D. degree in Electrical Engineering from KU Leuven and IMEC, Belgium. Contact him at ikarageo@aya.yale.edu.

PLACE PHOTO HERE

Karthik Sriram is currently a Ph.D. student at Yale University, New Haven. He received his B.S. degree in Computer Science from Rutgers University. His research interests are in Computer Systems and Architecture, Hardware-Software co-design, specially in the design of Brain-Computer Interfaces. Contact him at karthik.sriram@yale.edu.

PLACE PHOTO HERE

Xiayuan Wen is a PhD student at Yale University. Her research interests include computer architecture, and circuit design. She received a BS degree from Nanjing University and an MS degree from Yale University. Her contact email address is xiayuan.wen@yale.edu.

PLACE PHOTO HERE

Ján Veselý is a software engineer at NVIDIA. He graduated in 2021 from Rutgers University, his thesis focused on hardware and software methods of integrating accelerators into heterogeneous systems. Ján's interests span the areas of architecture, operating systems, and compiler techniques for accelerators

PLACE PHOTO HERE

Nick Lindsay Ṅick Lindsay is a graduate student at Yale. His interests lie in building secure, safe and high performance heterogeneous systems. Nick received his BEng Electrical Engineering degree from the University of Glasgow.

PLACE PHOTO HERE

Michael Wu is a PhD student at Yale University. He is interested in applications of machine learning in computer systems. He received his Bachelor's in Computer Science from Rutgers University, New Brunswick.

This article has been accepted for publication in IEEE Micro. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/MM.2023.3258907

PLACE
PHOTO
HERE

Lenny Khazan is a software engineer at Instabase. He is interested in software engineering and machine learning challenges involved in building computer systems. He received his Bachelor's in Computer Science from Yale University.

PLACE
PHOTO
HERE

Raghavendra Pradyumna Pothukuchi is an Associate Research Scientist and a CRA/NSF Computing Innovation Fellow at Yale University. His research area is computer architecture and systems, and he has interdisciplinary interest in cognitive science, quantum computing, brain-computer interfaces, formal control, energy efficiency, security, machine learning and compilers. He received his Ph.D. from the University of Illinois at Urbana-Champaign. Contact him at raghav.pothukuchi@yale.edu.

PLACE
PHOTO
HERE

Rajit Manohar Ṙajit Manohar is the John C. Malone Professor of Electrical Engineering and Professor of Computer Science at Yale University, New Haven, CT, USA. His research interests are in the design and implementation of asynchronous circuits and systems. He has a Ph.D. in Computer Science from Caltech. Contact him at rajit.manohar@yale.edu.

PLACE
PHOTO
HERE

Abhishek Bhattacharjee Åbhishek Bhattacharjee is an Associate Professor of Computer Science at Yale University. His research interests are in 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. Abhishek received his PhD from Princeton University in 2010, and his Bachelor's in Engineering from McGill University in 2005. Contact him at abhishek@cs.yale.edu.