Noema: Massive-Scale, Untethered, Real-Time Brain Activity Decoding

Ameer M. S. Abdelhadi, Eugene Sha, and Andreas Moshovos
University of Toronto
{ameer.abdelhadi@utoronto.ca, eugene.sha@mail.utoronto.ca, moshovos@ece.utoronto.ca}

There are approximately 86 billion neurons in the human brain with trillions of connections. These neurons generate and transmit electrophysiological signals to communicate within and between brain regions. In the 1780s, Galvani pioneered the study of bioelectricity using simple electrode technologies [1]. Our understanding of these signals has since evolved, resulting in modern neuroprobes which capture the activity of ever increasing populations of neurons, *e.g.*, [2], [3], [4]. The captured activity is digitized and processed in the digital domain. As advancements in sensing technologies accelerate, the downstream digital processing stack must also grow for the overall *brain-machine interface* system to keep pace and effectively utilize the incoming data. The success of such systems hinges upon meeting latency, throughput, energy, and form-factor constraints when processing the raw neural stream.

We present *Noema One*, an implementation of our *Noema* scalable architecture [5] that meets the requirements of a major component of such brain-machine interfaces: *pattern matching*. The goal of pattern matching is to detect, in real-time, when spatiotemporal neuron activity patterns of interest occur. This processing is purely in the digital domain, and its input are bit streams each representing the activity in time of a neuron. Such spatiotemporal patterns of neuron activity are thought to be key to understanding how the brain represents, reacts, and learns from the external environment [6], as the patterns are curated excerpts of memories, decisions, or perceptions [7], or reliable motor activations [8]. By detecting memories, decisions, emotions, and perceptions in real-time, pattern matching is essential for brain-machine interface applications including driving effectors such as robotic arms, memory retrieval, or even augmenting or "repairing" brain function. As most of these applications need to be *untethered*—where the device can be carried by the subject with a portable power source—a small form factor, low power consumption (*e.g.* < 2W [9]) and meeting a 5ms real-time processing constraint [5], [10] are highly desirable.

Fig. 1 shows a *brain-machine interface* (BMI) with template-matching based pattern detection. Using neuroprobes, neural activity is continuously sampled and processed by the *spike detection and sorting* stage, producing a time-ordered digital stream of *binary indicators*. The typical sampling rate for neural spikes is 30KHz [2] resulting in a 30,000 *bits/sec* stream per neuron. While modern neuroprobes are capable of capturing the activity of several hundreds of neurons [11], [12], the technology is rapidly evolving and neuroprobes capturing thousands and eventually millions of neuron are within sight [13], [14].

Noema One implements a widely used algorithm for pattern matching which computes the Pearson's Correlation Coefficient between the input stream and each of the pre-recorded templates. The memory and computation costs of Template Matching are dependent on the number of sampled neurons, and the number and size of the pre-recorded templates. The cost becomes prohibitive with the rapid increase in the number of neurons simultaneously recorded. Recent estimates range from up to 3K neurons [11] when recording with electrophysiological signals [12] to upwards of a million for optical signals [14].

Prior implementations of template matching implement Pearson's Correlation as originally proposed resulting in prohibitive memory, compute and energy costs. As we have shown, even desktop-class GPUs fail to meet real-time latency for more demanding applications [5]. These implementations first bin the input bit-streams of activity into streams of aggregate integer counts. The core computation initiates only *after* a full time window of relevant samples (equal to the template's time dimension) has been received, increasing memory storage and traffic while delaying response times. This motivated us to develop the *Noema* architecture.

Noema One is a prototype of our family of hardware accelerators that greatly reduce area and energy costs compared to commodity solutions while achieving real-time performance. *Noema* reduces costs and up-time while supporting more intensive workloads to enable further advances in neuroscience. A study of *Noema* justifying the architectural choices has been previously published [5]. This manuscript complements this prior work by presenting, for the first time, a fully-fabricated device of a *Noema* chip. To design a scalable solution we studied representative configurations for a broad spectrum of applications. At the lower-end are applications possible with existing commodity hardware (albeit still not portable), whereas at the high-end are applications that are not practical today but for which the neuroprobe technology is within reach. *Noema One* prototype chip supports a stream of 1K neurons, however, the architecture can be scaled to 30K neurons while still consuming much less than 2W.

At the core of the *Noema* architecture is a decomposition of the template matching algorithm where the bulk of the computations are performed using simple, low-cost, specialized *bit-level* operations. *Noema* performs computations as it receives samples bit-by-bit in sliding time window fashion. This enables *Noema* to produce the final output only a few cycles after the last bit in the relevant window is seen and avoids having to buffer the incoming stream—in contrast with existing implementations—saving memory storage and traffic. In contrast, existing implementations need to buffer a window's worth (template size in time which is in the order of 10s of megabytes for the most demanding configuration) of incoming data and can perform the computation only after the last sample is received failing to meet real-time constraints.

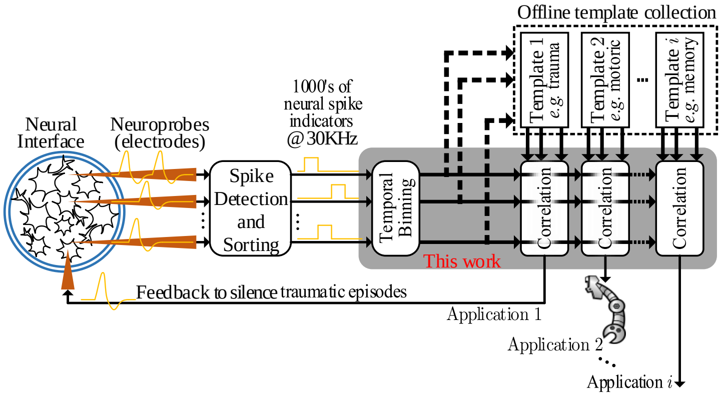
Offline template collection
Template 1 *e.g.* trauma
Template 2 *e.g.* motoric
...
Template *i* *e.g.* memory
Neural Interface
Neuroprobes (electrodes)
1000's of neural spike indicators @30KHz
Spike Detection and Sorting
Temporal Binning
Correlation
Correlation
Correlation
This work
Feedback to silence traumatic episodes
Application 1
Application 2
Application *i*

Fig. 1: Abstraction of template matching applications.

To further reduce memory costs, *Noema* exploits the observation that templates naturally exhibit a geometric-like distribution in their value content which is heavily biased towards very low magnitude values. *Noema* incorporates a hardware-efficient decoder to greatly reduce template storage and traffic with little energy and area cost. The simplicity and tiny area cost of *Noema*'s processing units enables tuning the operating frequency to improve power efficiency via partitioning and replication at a minimal area overhead.

We highlight the following architectural takeaways and innovations the characterize the *Noema* architecture:

- As presented and currently implemented, template matching requires storing large matrices for the templates and binned input streams, which reach 1.24Gb each.
- The computation needs also grow and reach 6.6TOPs (mostly floating-point) for the largest configuration planned (see Table I).
- Computation latency and throughput are the primary challenges. The compute throughput has to keep pace with the incoming stream while latency is restricted to 5 msec.
- *Noema* reformulates the computation so that the input streams are consumed as they are received bit-by-bit, obviating the need for buffering the input. This: 1) greatly reduces memory needs, and 2) allows *Noema* to meet real-time response goals as it leaves very little computation *after* the last piece of input is received.
- *Noema*'s formulation enables the use of tiny bit-serial units for the bulk of the computation. *Noema* replicates and places these units near the template memory banks (near memory compute). This enables highly parallel processing and scaling at low cost.
- *Noema* exploits the sparsity of the template content via per bank, light-weight, hardware friendly decompression units. Templates are compressed in advance in software (offline). *Noema*'s template compression reduces template memory size by at least 2.79× (most demanding configuration).
- *Noema* can greatly reduce power by gating accesses to the template memory, as the input bit stream is sparse and the inputs are processed a single bit at a time.
- An FPGA implementation meets real time requirement only for some of the configurations studied.
- An embedded-class CPU fails real-time constraints for all configurations while a desktop-class GPU fails the most demanding.

Table II shows the physical attributes of the *Noema* device family. *Noema One* (*NOEMA01K1T05S250*) has been fabricated in TSMC 65nm GP technology. The chip micrograph and floorplan are shown in Fig. 2. *Noema One* performs the equivalent of 600MOPs 32bit floating-point operations consuming just 730μW while also meeting the real-time constraint (24μs per template in a sliding window fashion). This is only possible because *Noema* rethinks how to compute the Pearson's Correlation Coefficient over binary indicator streams and in sliding window fashion. As a result, the bulk of the operations *Noema* performs use bit-serial, near memory units without loss of accuracy. By comparison, an Nvidia Jetson Nano requires 10W while it barely meets the real-time constraints for the same configuration and fails to do so for the larger configurations. An Intel i5-7000 also fails to meet real-time latency constraints (63ms). *Noema One* occupies 2.1mm^2 in the 65nm process mode and operates in 30MHz.

The larger *Noema* devices have been fully simulated for functional correctness and are planned for fabrication. The most demanding configuration requires 1.2W, occupies 205mm^2 while performing the equivalent of 6.6TOPs (FP32).

TABLE I: The *Noema* device family

						Requirements†		Implementation	
	Fmax (MHz)	*Neurons (1000's)*	*Templates*	*Duration* (Seconds)*	*Resolution# (mseconds)*	*Compute GOPs*	*Memory (Mb)*	*FPGA‡*	*ASIC§*
NOEMA01K1T05S250	30	1	1	5	250	0.6	0.3	✓	✓
NOEMA10K2T05S005	300	10	2	5	5	628.0	114.4	✓	Planned
NOEMA20K3T09S250	600	20	3	9	250	64.8	33.0	N/A¶	Planned
NOEMA30K4T09S005	900	30	4	9	5	6786.4	1236.0	N/A¶	Planned

* Duration of the decoded experience
# Resolution window of the incoming activities. Activities within this windows are binned (averaged).
† If executed on commodity hardware.
‡ Intels Stratix 10 FPGA
§ TSMC 65nm GP
¶ Not application; device cant meet target frequency.

TABLE II: *Noema* ASIC devices

	Silicon Area (mm^2)			Power (mW)			Latency (μsec)	Chip Status
	Memory	Logic	Total	Memory	Logic	Total		
NOEMA01K1T05S250	0.36	0.07	*0.43	0.30	0.43	0.73	23.9	In lab
NOEMA10K2T05S005	28.46	1.35	29.81	89.78	84.28	174.06	2.8	Simulated† ‡
NOEMA20K3T09S250	6.26	0.09	6.25	18.55	9.68	28.23	1.5	Simulated†
NOEMA30K4T09S005	202.00	3.42	205.42	682.70	522.76	1205.46	1.0	Simulated

* Core only; Total die size is 2.1 mm^2
† Fabricated with TSMC 65nm GP
‡ Also tested on Intels Stratix 10 FPGA

References

[1] L. Galvani, *Aloysii Galvani De viribus electricitatis in motu musculari commentarius*, 1791.

[2] J. Navajas, D. Barsakcioglu, A. Eftekhar, A. Jackson, T. Constandinou, and R. Q. Quiroga, "Minimum requirements for accurate and efficient real-time on-chip spike sorting," *J. of neuroscience methods*, vol. 230, pp. 51–64, 2014.

[3] S. Wang *et al.*, "A compact quad-shank cmos neural probe with 5,120 addressable recording sites and 384 fully differential parallel channels," *IEEE Trans. on Biomedical Circuits and Systems*, vol. 13, no. 6, pp. 1625–1634, 2019.

[4] K. Sahasrabuddhe *et al.*, "The argo: A 65,536 channel recording system for high density neural recording in vivo," *bioRxiv*, 2020.

[5] A. M. S. Abdelhadi, E. Sha, C. Bannon, H. Steenland, and A. Moshovos, "Noema: Hardware-efficient template matching for neural population pattern detection," in *MICRO-54: 54th Annual IEEE/ACM Int. Symp. on Microarchitecture*, 2021, pp. 522–534.

[6] S. Panzeri, J. H. Macke, J. Gross, and C. Kayser, "Neural population coding: Combining insights from microscopic and mass signals," *Trends in Cognitive Sciences*, 2015.

[7] H. F. Ólafsdóttir, F. Carpenter, and C. Barry, "Coordinated grid and place cell replay during rest," *Nature Neuroscience*, 2016.

[8] M. A. Lebedev, J. M. Carmena, and M. A. Nicolelis, "Directional tuning of frontal and parietal neurons during operation of brain - machine interface." *Society for Neuroscience Abstract Viewer and Itinerary Planner*, 2003.

[9] R. K. Shepherd, *Neurobionics: The biomedical engineering of neural prostheses*. John Wiley & Sons, 2016.

[10] D. Ciliberti, F. Michon, and F. Kloosterman, "Real-time classification of experience-related ensemble spiking patterns for closed-loop applications," *eLife*, 2018.

[11] C. Stringer, M. Pachitariu, N. Steinmetz, C. B. Reddy, M. Carandini, and K. D. Harris, "Spontaneous behaviors drive multidimensional, brainwide activity," *Science*, 2019.

[12] J. Jun *et al.*, "Fully integrated silicon probes for high-density recording of neural activity," *Nature*, vol. 551, no. 7679, pp. 232–236, 2017.

[13] I. Stevenson and K. Kording, "How advances in neural recording affect data analysis," *Nature neuroscience*, vol. 14, pp. 139–42, 02 2011.

[14] T. Kim *et al.*, "Long-Term Optical Access to an Estimated One Million Neurons in the Live Mouse Cortex," *Cell Rep.*, vol. 17, no. 12, pp. 3385–3394, Dec 2016.

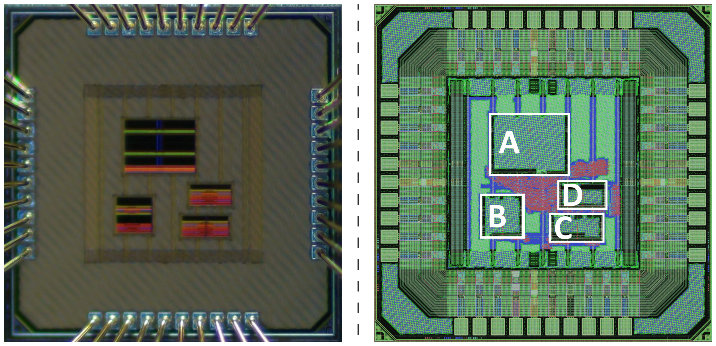

Fig. 2: NOEMA01K1T05S250 (left) die micrograph, and (right) floorplan. Distributed RAM blocks are highlighted, (A) template, (B, C, and D) compute RAM.