Design Principles of Large-Scale Neuromorphic Systems Centered on High Bandwidth Memory

Bruno U. Pedroni[1], Stephen R. Deiss[2], Nishant Mysore[2], and Gert Cauwenberghs[1,2]
[1]Institute for Neural Computation, [2]Department of Bioengineering
UC San Diego, La Jolla, CA, USA – Email: bpedroni@eng.ucsd.edu

2020 International Conference on Rebooting Computing (ICRC) |978-1-6654-1975-8/20/$31.00 ©2020 IEEE | DOI: 10.1109/ICRC2020.2020.00013

***Abstract*—In order for neuromorphic computing to attain full throughput capacity, its hardware design must mitigate any inefficiencies that result from limited bandwidth to neural and synaptic information. In large-scale neuromorphic systems, synaptic memory access is typically the defining bottleneck, demanding that system design closely analyze the interdependence between the functional blocks to keep the memory as active as possible. In this paper, we formulate principles in memory organization of digital spiking neural networks, with a focus on systems with High Bandwidth Memory (HBM) as their bulk memory element. We present some of the fundamental steps and considerations required when designing a highly efficient HBM-centric system, and describe parallelization and pipelining solutions which serve as a foundational architecture for streamlined operation in any multi-port memory system. In our experiments using the Xilinx VU37P FPGA, we demonstrate random, short burst-length memory read bandwidths in excess of 400 GBps (95% relative to sequential-access peak bandwidth), supporting dynamically reconfigurable sparse synaptic connectivity. Therefore, the combination of our proposed network model with practical results suggest a promising path towards implementing highly parallel large-scale neuromorphic systems centered on HBM.**

***Index Terms*—neuromorphic, High Bandwidth Memory, in-memory computing, non-von Neumann architecture, throughput**

I. Introduction

Besides event-driven transmission and processing of binary events (*spikes*), two of the defining characteristics of neuromorphic systems include parallel processing (*neurons*) and in-memory compute (*synapses*) [1], [2]. There is some leeway in how a neuromorphic system is designed, particularly with respect to the analog and/or digital components used for emulating the neural equations and storing the synaptic weights; however, the amount of parallelism and the processing bottleneck will be ultimately governed by the memory throughput (for fetching synaptic data) in the system [3].

In digital neuromorphic systems, synaptic weights are normally stored in DRAM (for compact solutions) or SRAM (for power-efficient solutions) memory elements [4]. The neuron typically presents some reconfigurable parameters (e.g., threshold, membrane potential time constant, synaptic/dendritic equations), requiring a non-negligible amount of digital logic (in relation to an analog counterpart), and, thus, a large number of physical instantiations of the neuron is not possible. This leads to solutions in which the neurons are shared, operating in a time-multiplexed manner. Analog neuromorphic systems, on the other hand, typically present physical instantiations of all the neurons, resulting in a completely non-von Neumann architecture at the processor level [5]. In terms of synapse representation, however, large-scale implementations are still not a feasible and reliable solution, owing greatly to the mismatch effects introduced through imperfections in the production process. Ultimately, large-scale versions of these systems rely on digital memory elements for synaptic connectivity, resulting in mixed-signal solutions.

Alternatively, a reconfigurable digital system using field-programmable gate arrays (FPGAs) is interesting as it can lie anywhere inside the spectrum of physically instantiated neuron cells – naturally, granted sufficient hardware resources. Additionally, hardware reconfigurability has the advantage of tailoring to a wide gamut of use cases and audiences, ranging from neurosciencientific neuron models consisting of multiple state variables with thousands of synapses to simple threshold neurons in machine learning.

To study and define the solutions for mitigating the bottlenecks when designing a large-scale neuromorphic system, we chose the state-of-the-art in terms of digital memory: High Bandwidth Memory (HBM). As compared to off-die memory components, HBM presents lower power consumption, higher bandwidth, less heating, and overall better performance, obtained using silicon 3D stacking technologies which places the DRAM in the same package as the processor via a silicon interposer. In our specific case, the HBM is part of the Xilinx Virtex UltraScale+ FPGA used in our experiments.

Lastly, an additional advantage of using FPGAs is that there is basically only a single design focus: optimizing the system architecture in order to obtain the highest throughput. Since the physical resources are already there (independently of their use or not) and power consumption in FPGAs is known to be in the intermediate part of the scale (when compared to low-power ASICs and power-hungry CPUs/GPUs), reduced area and power are not our primary targets. Therefore, in the case of our proposed architecture, the functional bottleneck should be memory access, and so our design must work around this in order to keep the HBM as active as possible.

II. System design

In an ideal neuromorphic system, each neuron would operate as a dedicated processor, meaning that it would be phys-

This research was supported by National Science Foundation CISE-1823366, DARPA HyDDENN, and Western Digital Corporation.

978-1-6654-1975-8/20/$31.00 ©2020 IEEE
DOI 10.1109/ICRC2020.2020.00013

ically instantiated and operate independently of all the other neurons, having exclusive access to its own local memory. Seeing as this is not a reality for large systems, how close can we come to this limit of a completely non-von Neumann architecture? To answer this question, we must first define the individual functional components of our system and their interdependence, to then identify the amount of parallelism available. In the process, we will also identify the bottleneck and design the surrounding components in order to minimize its impact, maximizing the overall system efficiency.

A. Basic components of a neuromorphic system

Motivated by earlier works [4], [6]–[8], we can define the basic components of a neuromorphic system found inside a so-called *neurosynaptic core*. Each core operates in parallel to the other cores, and presents the following functional blocks (refer to Fig. 1): (1) a central processor, responsible for organizing the operation of the other components; (2) presynaptic event memory, which stores spike information coming into the core; (3) postsynaptic neuron equations and memory; (4) pointer memory and FIFO, responsible for decoding spike events into HBM addresses (pointers) for synaptic connection lookup (and possibly temporarily storing the pointer in the FIFO); and (5) synaptic connectivity memory (the bulk of system memory). Communication of spike events between cores is performed via a router. Apart from network and core configuration commands (typically only sent prior to actual network execution), the only information continuously conveyed by the router is spike information.

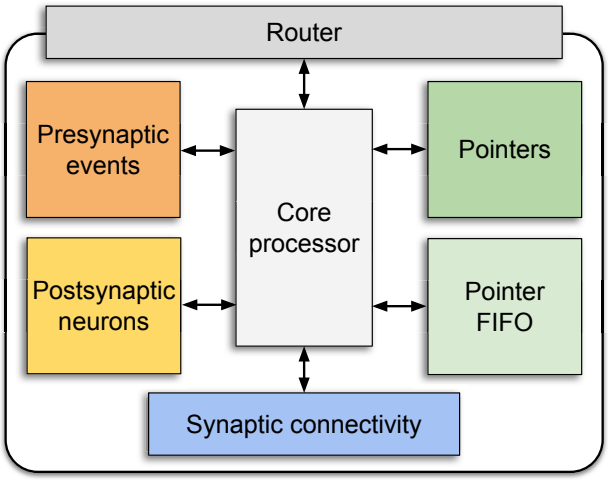

Fig. 1: The functional blocks of a neurosynaptic core.

B. An HBM-centric architecture

As compared to off-die memory components, HBM presents lower power consumption, higher bandwidth, less heating, and overall better performance. This is possible because HBM is a non-planar memory (cube or cuboid) that uses silicon 3D stacking technologies to place the FPGA and the DRAM beside each other in the same package via a silicon interposer. In the Xilinx VU37P FPGA there is a total 8 GB of memory and 32 HBM ports (blue boxes in Fig. 2), accessed via the Advanced eXtensible Interface (AXI) protocol, part of the ARM Advanced Microcontroller Bus Architecture 4 specification [9].

As the prime component in our system, the HBM is responsible for storing the bulk of the system memory: the synaptic connectivity information. In Figure 2 we have depicted the HBM in the center of the high-level model representation of our proposed system, with a neurosynaptic core attached to each of the 32 individual HBM ports. In this manner, each core can process and access memory in complete parallel fashion. This parallelism is one way in which our model moves away from the traditional von Neumann architecture. In Section III-C, we will show that inside each core there is yet another level of parallelism, this time based on the amount of data obtained from HBM at each clock cycle. Lastly, in the case of the VU37P, the additional functional blocks of the neurosynaptic core (incoming spikes, neurons, and pointers) are stored in SRAM devices (Block RAM and/or UltraRAM).

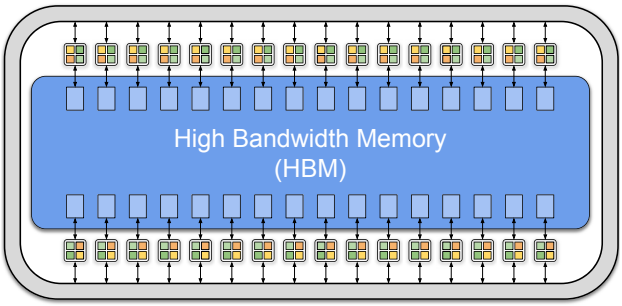

Fig. 2: The HBM as the central component of our system. Each neurosynaptic core stores and accesses synaptic connectivity information via an individual HBM port.

C. System operation

Though our system is comprised of multiple cores operating in parallel, we must define an algorithmic time step to which all the cores can synchronize. This is necessary in order to correctly compute the time-dependent neuron equations and update the state variables accordingly. Additionally, this gives the system a window during which all the events and neurons in a single core are allowed to be processed *sequentially*, producing, nonetheless, seemingly *parallel* computations. The time step is typically defined in a biological scale, usually in the order of 1 millisecond.

The proposed system operates in two phases. First, at the onset of a new time step, the system fetches the active presynaptic events and pushes their associated (start and stop) pointers to the pointer FIFO. Next, the pointers of the active ("spiked") postsynaptic neurons are also pushed into the FIFO. The second phase entails popping the FIFO and using the pointers to access the HBM for synaptic connectivity

information, including weights to on-core neurons and spike events to off-core destinations (conveyed via the router). In order to preserve an accurate algorithmic representation, the two phases (which include spike generation, processing, and routing) should be concluded inside the time step window.

III. High-throughput: keeping the HBM busy

In this section we describe the design principles for maximizing system throughput by optimizing the surrounding components of the synaptic memory. Though these principles are focused on an HBM-centric architecture, they remain valid for other multi-port memory systems.

A. Presynaptic-driven memory organization

Event-driven processing is the essence of neuromorphic systems. In physical realizations of these, memory organization and its impact on overall system operation are critical. Therefore, for a neuromorphic system to be truly event-driven, synaptic connectivity should be organized in a forward fashion; that is, connections should be stored sequentially from the perspective of the presynaptic neuron to all its downstream neurons [10], [11]. With this, synaptic connectivity memory access must only be done for active presynaptic neurons, with the added benefit of sequential burst reads – extremely important for optimizing HBM throughput. Additionally, memory access can be completely avoided for inactive presynaptic neurons; in this case, an entire portion of the memory (corresponding to all of the presynaptic neuron's connections) can be skipped.

B. Presynaptic event storage

There are two main ways of buffering incoming presynaptic events to be used in the next system time step: bitmap and FIFO. For a core with N inputs, a bitmap representation of the events consumes N bits (one per input). In the FIFO representation, for given a core input activity density ρ, a total of $\rho N \log_2 N$ bits are required for storing all the spikes. Based on the number of core inputs and the activity density, we can then compute the bitmap-FIFO equality activity density as $\rho_{eq} = 1/\log_2 N$. The top inset of Fig. 3 shows the curve of ρ_{eq} for a range of N, where storing input events in a bitmap is more advantageous (in terms of memory cost) for densities above the curve, while the opposite is true for a FIFO.

Next, we computed the average input activity (i.e. firing rate of presynaptic neurons), based on the system time step duration, required to produce the equality activity density. Using integer $n = \log_2 N$ ranging from 8 to 20, we can observe in the bottom inset of Fig. 3 that for a 1-ms time step the average firing rate which produces equivalent memory costs between bitmap and FIFO lies between 50 and 125 Hz. Therefore, if for a given network the expected activity per core can be estimated, the best input activity storage method can be chosen accordingly based on this graph.

Nonetheless, it is worth mentioning that, even for a sparsely active core, using a bitmap has two advantages over a FIFO. First, a FIFO cannot be allocated dynamically, meaning that it can overflow depending on an unexpected burst of incoming events. Second, and most importantly, in a bitmap the events can be accessed in a consistent order: at the onset of a system time step we can sequentially read each bitmap position to determine if the associated input is active. This is a vital detail for memory throughput optimization given that sequential read accesses in HBM yield higher throughput than random accesses.

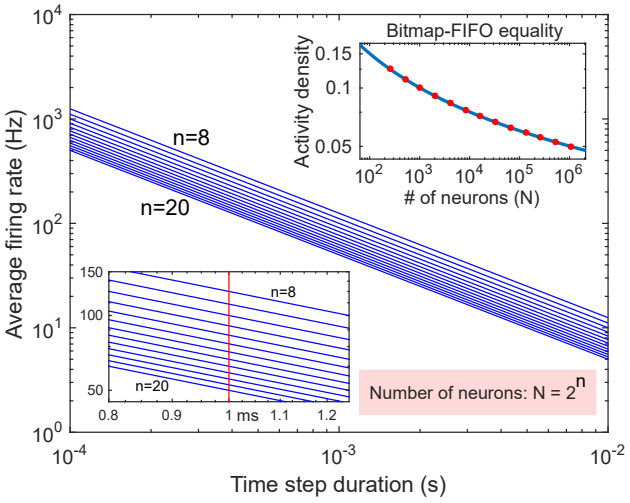

Fig. 3: Presynaptic event storage strategy comparison.

C. Intra-core parallelism and pipelining

The final consideration of an efficient HBM-centric system regards a second layer of parallelism: inside the postsynaptic neurons. Since a truly event-driven neuromorphic system must have synaptic connectivity data organized using a presynaptic perspective (refer to III-A), consecutive weights in HBM will belong to different postsynaptic neurons.

In the HBM, at each read response we receive a 256-bit packet of data (per port), containing multiple synaptic weights. In order to streamline memory access, we must immediately apply all 256 bits of weights to their respective postsynaptic neurons. Therefore, the number of weights that can be read per clock cycle represents the number of neuron processors which must be *physically instantiated* in a core, each with its own SRAM memory block for parallel state variable access. In sum, it is the combined multi-port access to HBM and the multi-weight data packet from each HBM port read response that defines the amount of parallelism in our system – and how far we can distance it from the traditional von Neumann architecture.

Lastly, to reduce the clock frequency required to process and update neuron state variables, we can use the pipelining strategy depicted in Fig. 4. In this specific case, by duplicating the number of neuron groups, we can halve each group's operating frequency in order to reduce the possibility that the combinational logic of the neuron groups be the critical path of the system. In the figure, each neuron group is comprised

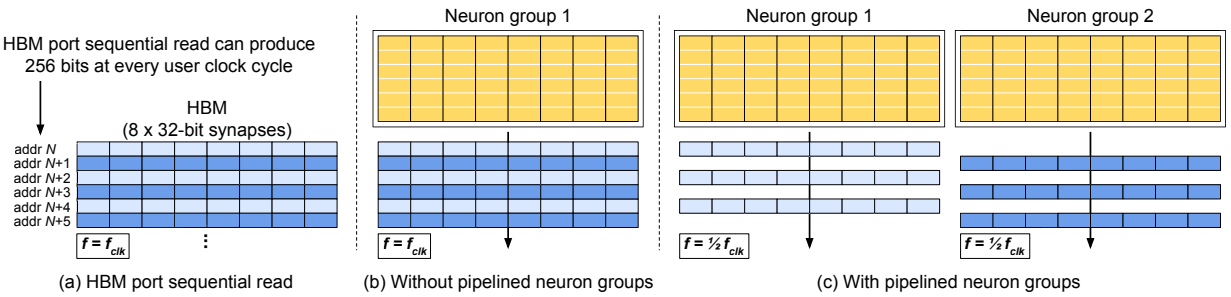

Fig. 4: Pipelining reduces the possibility of the neuron groups being the critical path of the system.

of 8 individual SRAM blocks (columns), and one physical neuron is instantiated per column. By pipelining two neuron groups, each group has to operate only half of the time; the HBM, nonetheless, can remain continuously active.

IV. Experimental results: HBM throughput

In the Virtex UltraScale+ VU37P FPGA by Xilinx, the 32-port HBM operates at a maximum frequency of 900 MHz (double data rate). At each rising and falling HBM clock edge, a port can access 64 bits of data, resulting in a maximum theoretical bandwidth of $32 \times (2 \times 900\text{M}) \times 64 \approx 460$ GBps. The HBM can be addressed at a granularity of 256 bits (32 bytes), meaning that 8 GB of HBM requires 28 bits per address. Additionally, read and write commands can be performed in bursts, ranging from 1 to 16 data points.

To verify the actual HBM throughput, representing the capacity of our proposed system, we designed an HBM tester block in Verilog (Fig. 5). The tester can perform sequential and (pseudo-)random memory accesses (*read_mode*), with the latter being a more faithful representation of actual system operation. By partitioning the HBM equally among the 32 ports, each port uses a 23-bit address to access its part of the memory. For generating the pseudo-random addresses, we created a 23-bit maximal linear-feedback shift register (LFSR). The tester block outputs the memory read duration and bit-error count. To analyze the entire HBM, an individual tester block was instantiated for each of the 32 ports.

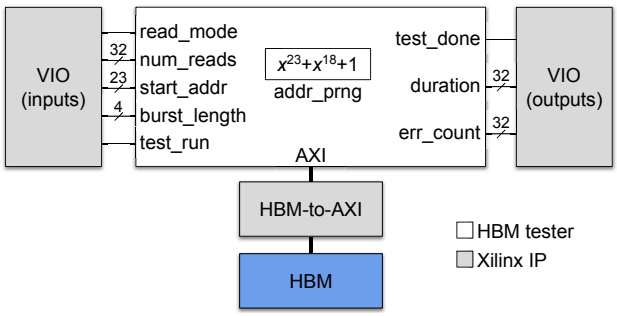

Fig. 5: The HBM port tester block design.

A benchmark maximum HBM throughput of ≈ 425 GBps was obtained by sequentially sweeping the memory using the maximal burst length, requiring one read command for every 16 read responses. This represents 92.4% of the theoretical maximum throughput. Next, we assigned distinct seeds (*start_addr*) to each port's LFSR (*addr_prng*) and swept the entire memory (all ports simultaneously) in a pseudo-random fashion for a range of burst lengths. This method of memory access is a more accurate representation of the random reads which are performed independently by each HBM port during actual synaptic connectivity lookup. HBM throughput results for all 32 ports performing random read accesses in parallel are presented in Fig. 6. For burst lengths above 3, a throughput above 350 GBps is already obtained; for burst lengths above 6 the throughput ranges around 390 and 405 GBps, with the maximum recorded throughput of 406.6 GBps for burst length 15 (95% of the benchmark sequential read throughput value).

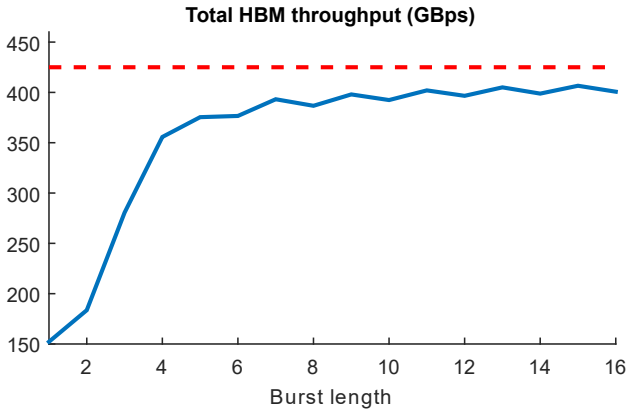

Fig. 6: Total HBM throughput for the 32 ports performing random read access independently and simultaneously.

V. Conclusions and considerations

Though there are many moving parts when designing a large-scale neuromorphic system, practically speaking, the actual parallelism and overall system throughput will be defined by synaptic connectivity memory throughput. In this paper, we presented some of the fundamental steps and considerations required when designing a highly efficient system centered around High Bandwidth Memory. In order to operate as

close as possible to full throughput capacity, the system should be designed considering a presynaptic-driven memory organization, with bitmap representations enabling sequential memory access, and neuron groups providing additional intra-core parallelism and pipelining. Experimental results showed that throughput of over 400 GBps can be obtained even when performing completely random read access simultaneously over all 32 HBM ports. Therefore, the combination of our proposed network model with practical HBM results provide a promising path towards implementing highly parallel large-scale digital neuromorphic systems.

REFERENCES

[1] C. Mead, "Neuromorphic electronic systems," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1629–1636, 1990.

[2] G. Indiveri, B. Linares-Barranco, T. J. Hamilton, A. Van Schaik, R. Etienne-Cummings, T. Delbruck, S.-C. Liu, P. Dudek, P. Häfliger, S. Renaud, *et al.*, "Neuromorphic silicon neuron circuits," *Frontiers in Neuroscience*, vol. 5, p. 73, 2011.

[3] I. K. Schuller, R. Stevens, R. Pino, and M. Pechan, "Neuromorphic computing – from materials research to systems architecture roundtable," tech. rep., USDOE Office of Science (SC)(United States), 2015.

[4] P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza, A. S. Cassidy, J. Sawada, F. Akopyan, B. L. Jackson, N. Imam, C. Guo, Y. Nakamura, *et al.*, "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science*, vol. 345, no. 6197, pp. 668–673, 2014.

[5] T. Yu, J. Park, S. Joshi, C. Maier, and G. Cauwenberghs, "65k-neuron integrate-and-fire array transceiver with address-event reconfigurable synaptic routing," in *2012 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pp. 21–24, IEEE, 2012.

[6] S. Joshi, B. U. Pedroni, and G. Cauwenberghs, "Neuromorphic event-driven multi-scale synaptic connectivity and plasticity," in *Signals, Systems, and Computers, 2017 51st Asilomar Conference on*, pp. 1–5, IEEE, 2017.

[7] G. Detorakis, C. A. Sadique Sheik, S. Paul, B. U. Pedroni, N. Dutt, J. Krichmar, G. Cauwenberghs, and E. Neftci, "Neural and Synaptic Array Transceiver: A Brain-Inspired Computing Framework for Embedded Learning," *Frontiers in neuroscience*, vol. 12, 2018.

[8] M. Davies, N. Srinivasa, T.-H. Lin, G. Chinya, Y. Cao, S. H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain, *et al.*, "Loihi: A neuromorphic manycore processor with on-chip learning," *IEEE Micro*, vol. 38, no. 1, pp. 82–99, 2018.

[9] D. Flynn, "AMBA: enabling reusable on-chip designs," *IEEE micro*, vol. 17, no. 4, pp. 20–27, 1997.

[10] B. U. Pedroni, S. Sheik, S. Joshi, G. Detorakis, S. Paul, C. Augustine, E. Neftci, and G. Cauwenberghs, "Forward table-based presynaptic event-triggered spike-timing-dependent plasticity," in *Biomedical Circuits and Systems Conference (BioCAS)*, pp. 580–583, IEEE, 2016.

[11] B. U. Pedroni, S. Joshi, S. R. Deiss, S. Sheik, G. Detorakis, S. Paul, C. Augustine, E. O. Neftci, and G. Cauwenberghs, "Memory-efficient synaptic connectivity for spike-timing-dependent plasticity," *Frontiers in Neuroscience*, vol. 13, p. 357, 2019.