Hierarchical Multicast Network-On-Chip for Scalable Reconfigurable Neuromorphic Systems

Gopabandhu Hota[†‡], Nishant Mysore[*‡], Stephen Deiss[*‡], Bruno Pedroni[*‡], and Gert Cauwenberghs[*‡]
Department of Electrical and Computer Engineering[†], Department of Bioengineering[*], and Institute for Neural Computation[‡]
UC San Diego, La Jolla CA 92093
ghota@ucsd.edu, nmysore@ucsd.edu, sdeiss@ucsd.edu, bupedroni@gmail.com, gcauwenberghs@ucsd.edu

***Abstract*—State-of-the-art neuromorphic computing architectures to date suffer from interconnect scalability required for large-scale neural processing. We present a high-performance and low-overhead multicast network-on-chip (NoC) architecture for hierarchical address event routing (Multicast-HiAER) suitable for large-scale reconfigurable neuromorphic systems. Each building block of this efficient NoC architecture consists of several multi-cast advanced high-performance buses (mAHB) running in parallel for high-bandwidth inter-core spike event transmission. This architecture for scalable event routing can help to implement brain-scale sparse neural network connectivity distributed across neuromorphic processing cores, with network constraints typical of locally dense and globally sparse neuron connectivity. For a demonstration using a Xilinx Virtex Ultrascale VU37p FPGA, we have shown an 8×8 grid of mAHBs running at 512MHz clock performing Level-1 and Level-2 inter-core communication at top bandwidth of 420M events per second per 128k neuron node in the hierarchy. This peak absolute bandwidth supports spike event registration with sub-ms latencies under worst-case conditions of all postsynaptic destinations being off-core.**

***Index Terms*—Neuromorphic Computing, Multicasting Network-On-Chip, Advanced High-Performance Bus (AHB), Address-event-representation (AER), Scalable AER**

I. Introduction

Neuromorphic computing has gained tremendous interest in recent years by addressing the computational bottleneck in current high-performance computing systems limited to a small number of cores. Mapping very large-scale models of the biological brain into scalable silicon architectures is a complex challenge. Apart from the scaling challenge in area, power, and throughput, silicon models should also support dynamic reconfigurability of synaptic connectivity in neuronal networks. As the axons in biological neural networks carry action potentials (spikes), similar distributed communication needs to be mimicked in neuromorphic chips. Currently, most of the neuromorphic chips implement this inter-core communication via an Address-event representation (AER) protocol, where each core, containing an array of neurons, send the source or destination neuron address on a shared digital bus when they spike. AER-style communication with synaptic routing tables in source and destination cores provides flexibility and reconfigurability by supporting dynamical assignment of synaptic connections between neurons in different cores. This is extremely crucial for reprogrammable neuromorphic cores as well as adaptive updates of synaptic strength and neuronal connectivity. AER-based inter-core connectivity thus presents a suitable framework for building neuromorphic systems for end-to-end sensory-motor tasks comprised of multichip integration of bio-inspired silicon retinae, silicon cochlea, silicon cortex and connecting them over a unified interface to exchange neural events.

Bandwidth requirements for AER bus to communicate large number of events across core are huge, thus scaling of neuromorphic systems has proven to be a challenge. Several neuromorphic chips have addressed this issue by incorporating differing large-scale network-on-chip (NoC) architectures. Neurogrid [1] [2] has linear grid and tree topologies for point-to-point communication using multiple AER buses based on global addresses of neurons. Certain architectures implementing multicasting mesh AER [3] store router-to-router connectivity in local routing tables and suffer from extreme bandwidth required for a larger number of cores. IBM TrueNorth [4] implements a 2D-mesh of 64x64 cores each with 256 crossbar connected neurons. SpiNNaker [5] [6] has a torus network-on-chip for scalability by incorporating global addressing strategy at the cost of a larger local routing table. However, all these implementations are not very memory-efficient with high cost of routing tables and reduced flexibility and expandability to scale up to brain-scale networks. HiAER [7] addresses this problem by partitioning a network into multiple hierarchies and incorporating a tree-based AER NoC using smaller routing tables and leveraging relay neurons. This is suitable for locally dense and globally sparse connections by keeping bandwidth the same even when having more cores at a higher hierarchy. Usually (Pre-synaptic address, Post-synaptic address, Synaptic weight) is sent over the AER bus for maximum flexibility in connectivity. However, there have been various memory-efficient reconfigurable connectivity strategies [8] that have been developed to store synaptic weights at the destination cores, sending only destination address over AER bus, to gain performance with reduced complexity.

Here we demonstrate Multicast-HiAER, which is more scalable and flexible than HiAER to implement sparse neuronal networks hierarchically partitioned into neuromorphic cores. It further boosts the NoC performance by allowing

This research was supported by the National Science Foundation, Office of Naval Research, Western Digital Corporation, and Defense Advanced Research Projects Agency.

simultaneous event transmission across different hierarchies. This also reduces the latency for event propagation between two cores separated in the hierarchy. We also optimize our multicast bus design by incorporating extreme reduction of the routing complexity for AER communication. For each individual multi-cast bus segment that implements AER-based routing protocol for sending spike destination address, we modified the industry-standard Advanced Microcontroller Bus Architecture (AMBA5) Advanced High-performance bus (AHB) [9] protocol, while still adhering to a similar, but simpler handshaking mechanism. We refer to each multicast AHB bus as mAHB.

II. Multicast-HiAER Architecture

Our designed NoC consists of hierarchically arranged parallel multicast buses for connecting cores at different levels. In each level of the hierarchy, total communication bandwidth is the same, as opposed to other NoC architecture where higher-level links are most congested. For illustration, 3 levels of communication with an expanded view of a 2D grid of 8×8 cores interconnected by 8 level1 (L1) bus and 8 level2 (L2) bus as shown in Fig. 1. The 2D grid consists of 8 clusters, where each cluster consists of 8 cores that communicate between each other with the dedicated L1 mAHB. As a means for L2 communication, only the same core IDs in every cluster communicate with each other using similarly dedicated mAHB. In L3 communication, only the same core IDs in each 2D grid are connected over a mAHB. Thus, for an arrangement of 8×8×8 cores, we need 64 L3 buses (each connecting 8 cores vertically) running in parallel. This hierarchically structured communication allows keeping the network bandwidth requirement the same even if we scale the number of cores. As compared to an all-to-all mesh connection where bandwidth NoC requirement increases linearly with the number of cores and design complexity is higher and where simple tree connection often congests the root node routers, Multicast-HiAER offers a scalable design with simpler routing blocks. Our architecture supports simultaneous L1, L2, and L3 event transfer, in contrast to previous HiAER [7] architecture which has higher latency by comparison. For simplicity, henceforth we will only discuss hardware architecture and network connectivity with only 2 levels of hierarchy.

An example of inter-core network connectivity and the mapping into the NoC architecture with the simultaneous orthogonal routing mechanism is shown in Fig. 2. For supporting such hierarchical and orthogonal networking architecture, the actual network compilation to place neurons into this hierarchically structured connectivity plays critical importance. In case there are any diagonal connections even after the compilation optimization, it is realized by a combination of L1 and L2 messaging via a relay neuron [7]. The compiler should handle the network partitioning and placement of neurons in such a way that, only the cores connected locally within their own cluster have maximal local connectivity and inter-cluster diagonal connectivity (via relay neurons) is minimized

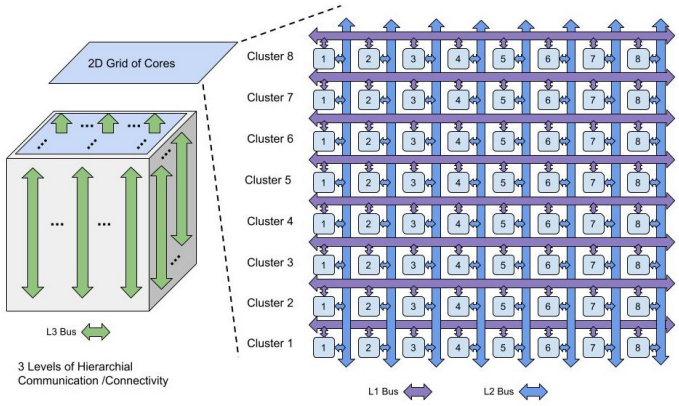

Fig. 1. High-level conceptualized diagram of NoC connecting 8×8×8 cores with multiple mAHBs at L1 (in purple), L2 (in blue) and L3 (in green) communication. This NoC architecture supports concurrent transfer of L1, L2 and L3 messages , thus saving the latency and improving performance through simpler design compared to 2D mesh, torus or 2D tree architectures.

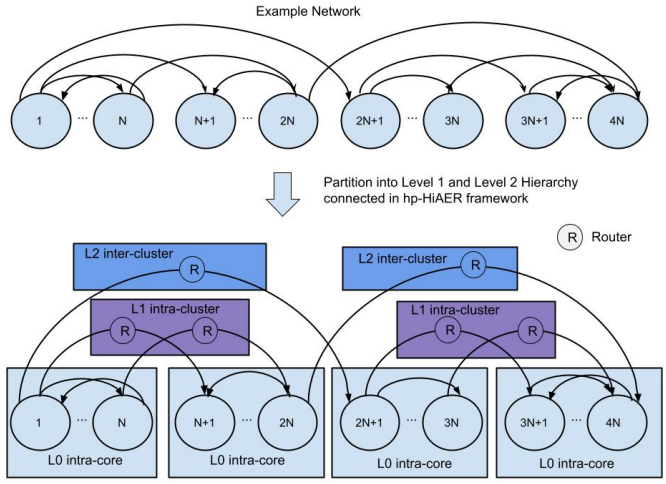

Fig. 2. Example partitioning of a neuron connectivity graph and their mapping into final placement and routing using our Multicast-HiAER approach.

[10]. This neural compiler optimization is not in the scope of discussion in this paper.

Each mAHB is implemented as a multicasting architecture where all cores take their turn to write into the bus in a time-multiplexed manner, while all the other cores listen to the written spike packet. Efficiency in the networking is obtained by using mask bits to select the correct destination cores. Each core stores the routing table containing instruction codes for each neuron. Description of different fields in the instruction code for i.e. an L1 or L2 message is shown in Table. I, which is decoded to route the outgoing event in the appropriate hierarchy. There are 2 router interfaces (RI) in each core dedicated for Level 1 and Level 2 messages. The combined datapath for L1 and L2 bus router interface, spike FIFOs (used to store off-core events), and external event input processor is shown in Fig. 3. Level 1 spike FIFOs and Level 2 spike FIFOs store L1 and L2 outgoing events when any pre-synaptic neuron having off-core destinations fires.

TABLE I
INSTRUCTION CODE FOR EVENT TRANSFER AT DIFFERENT NOC LEVELS

Level-1 off-core event Op-code 2b (10)	Destination Input (17b Address for cores with 128k neurons)	Mask Bits (8b for selecting the targets for multicast)
Level-2 off-core event Op-code 2b (11)	Destination Input (17b Address for cores with 128k neurons)	Mask Bits (8b for selecting the targets for multicast)

III. MODIFIED MULTICAST AHB (MAHB) ARCHITECTURE

For higher performance and lower complexity in the NoC, we implemented an AER multicasting bus architecture mAHB that follows a similar handshaking protocol as AHB, but consists of a reduced number of lines for low-power and low-congestion implementation. As this is a multicasting architecture with mask bits in the data frame, all cores listening to the data on the bus can decode the received mask bits to determine if the spike intended is for them. Thus, we have removed the address lines and address decoders, which is an important and intensive part of the original AMBA AHB architecture. This provides us a very compact data packet and flexibility to design very simple peripheral circuits to perform the routing.

Each RI has a separate initiator and target interface, data and control signals of each are described in detail in table II and III. The initiator interface in each RI sends out the spikes to all peer cores when it gets access to the bus. The target interface in each RI receives the spike packets (destination address and mask bits) and sends out the address into the external event processor if the mask bit is "1".

Peripheral circuits needed for simplified multicast AHB consist of an arbiter, data mux, and control mux, as shown in Fig. 4. Arbiter searches for all the initiators requesting the bus with HBUSREQ signals and grants the bus to a single master using an arbitration scheme. After the current initiator is done with the transaction, the arbiter selects the next initiator for access to the bus. Depending on the core ID of the initiator having the bus access, the data multiplexer and control multiplexer selects the appropriate initiator signals to send to all targets. Control signals HREADY and HRESP from all slaves are combined to generate a single control line to inform the initiators about their availability and transaction status. Each initiator, target interface, and arbiter consists of optimized finite-state-machines (FSMs) for very low overheads and minimal hardware complexity.

TABLE II
INITIATOR INTERFACE SIGNALS AND THEIR DIRECTIONS

Signal	Direction	Bitwidth	description
HWDATA	Out	32	Write databus sent
HBUSREQ	Out	1	Initiator requests the bus
HGRANT	In	1	Bus Access granted
HTRANS	Out	2	IDLE/BUSY/Active transaction
HBURST	Out	4	Used for burst transfers
HRESP	In	2	Transfer status received
HREADY	In	1	Targets ready

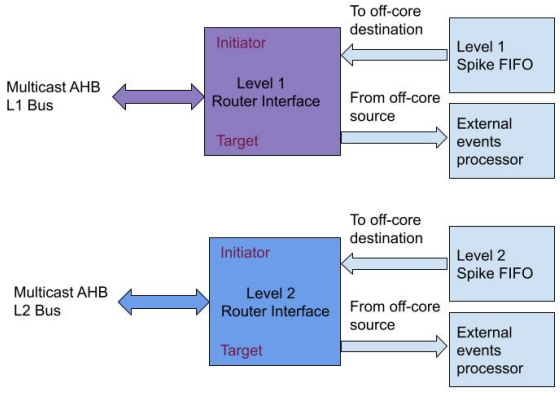

Fig. 3. Dataflow between core and mAHBs for spike transmission after event generation, and spike processing after event reception, both via dedicated mAHBs and router interfaces for each hierarchy. Separate initiator and target interfaces ensure the operation of spike transmission and spike reception within the same control logic and a simplified hardware realization for router.

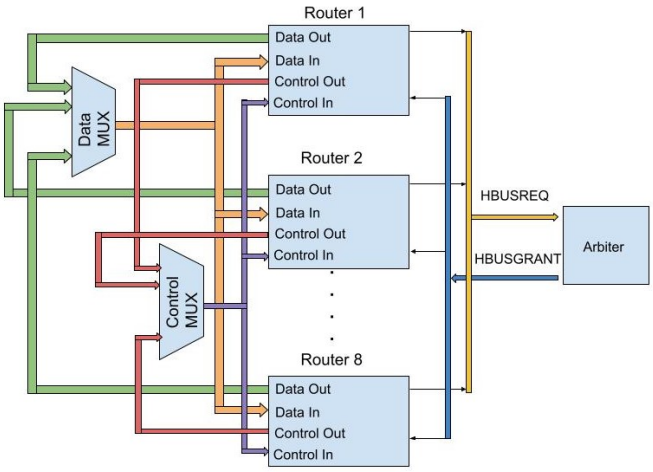

Fig. 4. Routing interfaces and peripheral circuits for supporting multi-cast data transfer over the mAHB protocol. Only a single router has the bus write access at any given time, while all the other cores reading from it.

TABLE III
TARGET INTERFACE SIGNALS AND THEIR DIRECTIONS

Signal	Direction	Bitwidth	description
HWDATA	In	32	Read Databus received
HTRANS	In	2	IDLE/BUSY/Active transaction
HBURST	In	4	Used for burst transfers
HRESP	Out	2	Data Transfer status
HREADY	Out	1	Availability for new packets

IV. PERFORMANCE RESULTS

FPGA resource utilization for the NoC backbone for L1 and L2 communication, containing 16 mAHBs and peripheral circuits is shown in table IV.

For a small-world graph with significant randomness where on-core and off-core connections are equally likely, we compare the synaptic event throughput and average latency per

TABLE IV
FPGA RESOURCE UTILIZATION SUMMARY FOR 16 MAHBS
(SYNTHESIZED ON A XILINX VIRTEX ULTRASCALE VU37P FPGA)

Resources	Usage
CLB LUTs	3120
CLB Registers	4032
F7 Muxes	608

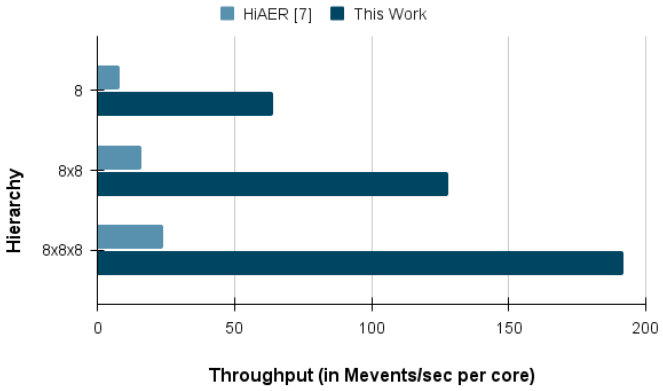

Fig. 5. Synaptic event throughput (per sec per 128-k neuron node) comparison between HiAER and Multicast-HiAER for various communication hierarchies.

spike metrics with previously benchmarks in HiAER [7]. Fig. 5 represents the event throughput per 128k-neuron node comparison between HiAER and Multicast-HiAER. Here, we observe that Multicast-HiAER provides a linear scaling in total event throughput, as we increase the number of hierarchies. The highest possible bandwidth out of 2-level 8×8 grid is 128M events per sec per 128k Neuron mode (when burst length=1). Therefore, it can handle the worst-case spike rate of 1000 Hz with 100% off-core messages. As compared to HiAER, where NoCs are arranged in a tree fashion and upper hierarchy messages still have to cross through the lower hierarchy routers, Multicast-HiAER provides a 10x more event throughput. This improvement is also due to the multicast routing topology which drastically decreases the communication traffic. Due to the orthogonal arrangement of mAHBs, we also obtain an almost linear scaling of throughput with the number of hierarchies.

We also compare the average latency per spike between HiAER and Multicast-HiAER in Fig. 6 in order to calculate the total communication overhead due to the shared mAHBs where each initiator interface takes its turn in a time-multiplexed manner using arbitration logic. Because of simultaneous event transfers and separate event FIFOs, the average latency for this NoC is drastically reduced from the previous work. All the eight cores connected to each mAHB have equal chances to get access to mAHB if they have any pending outgoing events. Also, there is some control overhead due to the initiator and target handshaking via READY and RESP signals. As the average latency is highly dependent on spike rate as well, we show the average latency numbers at different spiking rates. We see an overall reduction of 100% in average latency. This improvement is because of simultaneous multi-level event transfers possible by orthogonal mAHBs.

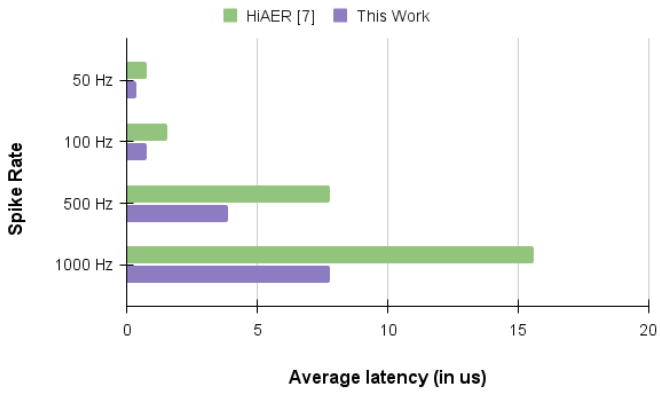

Fig. 6. Average spike latency (in us) comparison between HiAER and Multicast-HiAER for different spiking rates (all neuron spiking at same rate).

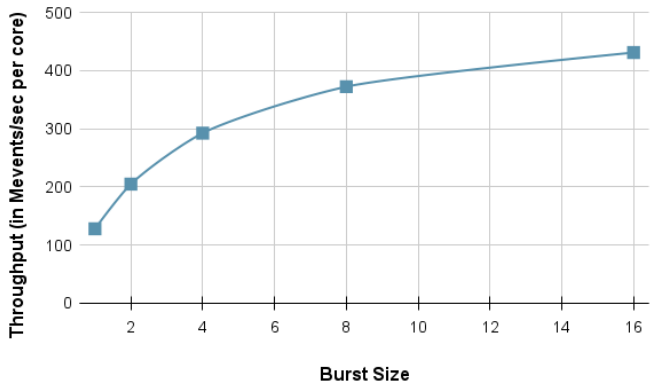

Fig. 7. Synaptic event throughput (per sec per 128-k neuron node) with Multicast-HiAER for messaging with different burst lengths.

While Fig. 5 and Fig. 6 show the number with spikes transaction burst length of 1, the synaptic event throughput is further enhanced if we utilize the higher burst lengths available using HBURST lines. As shown in Fig. 7, total synaptic event throughput per sec per 128k-neuron core improves from 128M to 420M events/sec with a burst size of 16×. Higher burst size hides the overall control overhead for HREADY/HRESP handshaking between initiators and targets, thus offering low latency in large-scale sparse connections. This bandwidth enhancement helps us to scale the number of neurons per core while ensuring communication of all events on time.

V. CONCLUSION

We presented a very high-performance network-on-chip that provides massive bandwidth for interconnecting neuromorphic cores with a large number of neurons and reconfigurable connectivity between them through AER. Our multicast AHB (mAHB) architecture handles spike transmission at high efficiency allowing increased bandwidth and reduced spike transmission latency beyond what's observed from prior state-of-the-art neuromorphic chips. This NoC architecture can further scale to interconnect extremely large neuromorphic compute clusters while maintaining the reduced hardware cost and complexity to realize that.

REFERENCES

[1] B. V. Benjamin, P. Gao, E. McQuinn, S. Choudhary, A. R. Chandrasekaran, J.-M. Bussat, R. Alvarez-Icaza, J. V. Arthur, P. A. Merolla, and K. Boahen, "Neurogrid: A mixed-analog-digital multichip system for large-scale neural simulations," *Proceedings of the IEEE*, vol. 102, no. 5, pp. 699–716, 2014.

[2] P. Merolla, J. Arthur, R. Alvarez, J.-M. Bussat, and K. Boahen, "A multicast tree router for multichip neuromorphic systems," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 61, no. 3, pp. 820–833, 2013.

[3] C. Zamarreño-Ramos, A. Linares-Barranco, T. Serrano-Gotarredona, and B. Linares-Barranco, "Multicasting mesh AER: A scalable assembly approach for reconfigurable neuromorphic structured AER systems. application to ConvNets," *IEEE transactions on biomedical circuits and systems*, vol. 7, no. 1, pp. 82–102, 2012.

[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] M. M. Khan, D. R. Lester, L. A. Plana, A. Rast, X. Jin, E. Painkras, and S. B. Furber, "SpiNNaker: mapping neural networks onto a massively-parallel chip multiprocessor," in *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pp. 2849–2856, Ieee, 2008.

[6] E. Painkras, L. A. Plana, J. Garside, S. Temple, F. Galluppi, C. Patterson, D. R. Lester, A. D. Brown, and S. B. Furber, "SpiNNaker: A 1-W 18-core system-on-chip for massively-parallel neural network simulation," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 8, pp. 1943–1953, 2013.

[7] J. Park, T. Yu, S. Joshi, C. Maier, and G. Cauwenberghs, "Hierarchical address event routing for reconfigurable large-scale neuromorphic systems.," *IEEE Trans. Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2408–2422, 2017.

[8] 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.

[9] "ARM AMBA5 AHB protocol specification," *ARM IHI 0033B.b (ID102715)*, 2015.

[10] M. et. al, "Hierarchical network partitioning for reconfigurable large-scale neuromorphic systems," p. in press.