Artificial Intelligence HW: Research Needs
June 2019

SRC's Global Research Collaboration is please to introduce the Artificial Intelligence Hardware (AIHW) program and a new description of research needs in this space. This program is a result of a combination of two previous efforts: System-Level Design (SLD) and Efficiency and Performance for Connectivity Constrained Computing (EP3C). Ongoing work in those programs will transition to AIHW for completion. Incorporated into this document are the needs identified through SRC GRC Executive Technical Advisory Board (ETAB) priorities, the AIHW Technical Advisory Board and other discussions.

The AIHW research needs are described in six major categories:

- Novel Architectures for Accelerating AI Computation
- Modeling and Simulation/Emulation of AI Hardware for Early System Exploration
- Power Efficient AI Hardware System Design
- HW/SW Co-design of AI Compute Systems
- Fairness, Robustness, Privacy, and Explainability of Models and Algorithms for AI Hardware
- Interplay of AI and System Architecture/Microarchitecture Design

Each of these major categories are broken down into several sub-categories which describe the need in more detail. Even so, these are written to be broad in nature to not restrict the investigator's approach. There is no priority order for either the major or minor needs that follow. In each category, there may be applications from large systems to small (datacenter and the edge/end node) and investigators should consider this in their submissions. With the application, appropriate metrics should be used to establish the impact of the advances. For instance, total throughput and throughput per watt might be metrics for datacenter applications while optimal energy usage might be more appropriate for the edge/end node.

The needs in the AIHW space actually cover a broad range of applications, including high performance processors for data centers, automotive, industrial, mobile computing and communication, and healthcare. Investigators are encouraged to link the results of their work with a potential application to help show the relevance of the proposed work.

This needs document is driving the AIHW solicitation. It is issued to universities worldwide, may be addressed by an individual investigator or a research team. Our selection process is divided into two stages. The interested party is requested to submit a brief 1-page white paper. The white paper should clearly identify what can be done in three years, and a successfully selected white paper will result in an invitation to submit a full proposal. These proposals will be further down-selected for research contracts. The number and size of the contracts awarded will be determined by the amount of available funds, and by the number of high-quality proposals.

Investigators who are funded will be expected to publish at top-tier conferences, including but not limited to ISSCC, VLSI, HPC, ISCA (part of Federated Computing Research Conference), MICRO, HPCA, ESSCIRC, and ESWEEK (CASES, CODESISSS, & EMSOFT).

White Papers for all the categories below will be considered for funding. Investigators are limited to participation in two white papers in this solicitation and submissions should highlight which category of need is addressed, such as "A2.3".

<u>CONTRIBUTORS</u>

ARM	Jose Joao, Dam Sunwoo
NXP	Ben Eckermann, Brian Kahne, Adam Fuks
Mubadala	Rafic Makki
IBM	Krishnan Kailas, Matt Ziegler
Intel	Michael Kishinevsky, Greg Chen, Rosario Cammarota
Mentor Graphics	Duaine Pryor
Qualcomm	Rishi Chatuvedi
TI	Steven Bartling, Mahesh Mehendale, Jim Wieser
SRC	Victor Zhirnov, David Yeh

2019 Artificial Intelligence HW Research Needs

<table>
<tr><th>A1</th><th>Novel Architectures for Accelerating AI Computation</th></tr>
<tr><td colspan="2">Accelerating future AI systems may benefit from architectures, circuits, and/or devices beyond today’s conventional computing approaches. New architectures that depart from the deep learning neural network paradigm may provide significant performance and/or power improvements for certain applications. Novel circuits and/or devices may also unlock capabilities unattainable from conventional circuit design and CMOS technology. At the system level, the challenge of integrating multiple chips or approaches to achieve the equivalent of multi-chip systems are of high importance for the future of AI computing.</td></tr>
<tr><td>A1.1</td><td>Architectures using emerging devices and circuits, e.g., NVM devices, near-memory circuits, compute-in-DRAM, compute-in-cache, etc.</td></tr>
<tr><td>A1.2</td><td>System-level integration solutions for emerging architectures, e.g., SoC, 3D, packaging, etc.</td></tr>
<tr><td>A1.3</td><td>Neuromorphic computing: hardware for biologically plausible neuron models and learning rules, such as spiking neural networks, spike timing dependent plasticity, and bio-plausible deep learning</td></tr>
<tr><td>A1.4</td><td>Use of approximate computing (beyond relaxed precision) for AI/Machine Learning architectures</td></tr>
<tr><td>A1.5</td><td>High / Hyper-dimensional computing</td></tr>
<tr><td>A1.6</td><td>AI architectures using quantum computing</td></tr>
<tr><th>A2</th><th>Modeling and Simulation/Emulation of AI Hardware for Early System Exploration</th></tr>
<tr><td colspan="2">End-to-end performance and energy efficiency of AI systems are determined by various components including memory subsystem, I/O, on-chip and off-chip network, in addition to core AI computation. Challenges include, but are not limited to, characterizing and modelling long running AI computations that often take days/weeks to complete. Novel methods for modelling, simulation and emulation are essential for early design-space exploration of next generation AI systems.</td></tr>
<tr><td>A2.1</td><td>AI workload analysis and characterization</td></tr>
<tr><td>A2.2</td><td>Efficient techniques for end-to-end performance/power/reliability modelling (cycle-accurate and analytical), simulation, emulation, and prototyping for exploration of AI systems</td></tr>
<tr><td>A2.3</td><td>Benchmarks for emerging AI applications</td></tr>
<tr><td>A2.4</td><td>Modeling infrastructure and techniques for AI computation at the edge/end node, including sensors</td></tr>
<tr><td>A2.5</td><td>Analysis and comparison of theoretical limits of AI compute efficiency</td></tr>
</table>

<table>
<tr><th>A3</th><th>Power Efficient AI Hardware System Design</th></tr>
<tr><td colspan="2">General challenges include but are not limited to: Energy-efficient end-to-end system architectures and partitioning (cloud to sensor), optimizing energy/bandwidth/latency tradeoffs at all levels within the computational hierarchy (data center, gateway, and edge/end node).

Devices on the edge/end nodes are typically heavily resource constrained with stringent cost, performance, power, communication latency and bandwidth limitations. Also, all edge/end node AI and microcontroller functionality typically resides on a single die, and is implemented on older process nodes to gain access to integrated NVM and high performance analog, creating additional area/power efficiency challenges. At the edge/end node, research is needed to optimize the interplay of on-chip sensing, compute, and off-chip communication requirements.

In the datacenter, high throughput is crucial, but must be balanced by power efficiency. Datacenter computing environments must combine energy efficient processor designs, multi-chip / module communication for data movement and memory access, and the flexibility / programmability to support diverse workloads. Center of cloud AI computation is highly data access limited (bandwidth, latency, storage), data movement limited (I/O bandwidth, power), and often thermally bounded. Novel architectures such as AI compute-in-memory that address fundamental limitations are of interest.</td></tr>
<tr><td>A3.1</td><td>End-to-end optimization schemes that span system-algorithm-architecture-circuit-technology stacks for minimizing energy per decision without compromising accuracy, throughput and cost (power, area, performance) constraints for AI systems consisting of sensors, pre- and post-processors, communication networks, and AI computer hardware</td></tr>
<tr><td>A3.2</td><td>Power efficient scalable AI architectures that can scale 100-1000x across power/performance/silicon area in order to span the middle to edge/end node AI computational needs</td></tr>
<tr><td>A3.3</td><td>Novel circuit/system architectures, e.g. mixed signal techniques combined with in-memory compute and reduced precision/dynamic range have the potential to yield several orders of magnitude improvements in AI energy consumption while maintaining consistent accuracy</td></tr>
<tr><td>A3.4</td><td>Inter-chip / module communication and partitioning for large-scale AI computing tasks, e.g., distributed training</td></tr>
<tr><td>A3.5</td><td>Optimal hardware accelerators which are both flexible and power/energy-efficient for diverse AI tasks across application domains. Example application platforms include cloud, datacenter, or automotive</td></tr>
<tr><th>A4</th><th>HW/SW Co-design of AI Compute Systems</th></tr>
<tr><td colspan="2">Interactions and dependencies between hardware and software are integral for achieving high performance on AI workloads. These two fields of study cannot be decoupled. On the contrary, opportunities abound for hardware/software co-optimization. Topics of interest include compilers that map deep learning models to CPU, GPU, and accelerator hardware with reduced data movement. These compilers may consider persistent memory techniques for near-memory or in-memory computations. AI hardware may require potential augmentation of existing operating systems and firmware to intelligently manage a large number of accelerators, virtualize processes, optimize thread scheduling, power/thermal management, and ensure data security and trust. AI at the edge/end node incorporating real-time sensing creates unique challenges for hardware/software co-design. Edge/end node processors would benefit from automatic dataset generation to enable ubiquitous AI.</td></tr>
<tr><td>A4.1</td><td>Compilers and run-time management that map AI algorithms/computations to homogeneous or heterogeneous hardware accelerators</td></tr>
<tr><td>A4.2</td><td>Compilers and run-time management that optimize data storage in compute in/near memory for reduced data movement</td></tr>
<tr><td>A4.3</td><td>Run-time management of large number of accelerators including virtualization and security of AI computation</td></tr>
<tr><td>A4.4</td><td>Co-design of AI exploration and sensing at the edge/end node</td></tr>
<tr><td>A4.5</td><td>Automated labeling of data sets for self-supervised learning</td></tr>
</table>

A5	**Fairness , Robustness, Privacy, and Explainability of Models and Algorithms for AI Hardware**
Machine Learning has made enormous strides in recent years in its ability to train models and infer results with higher degrees of accuracy than many other types of algorithms. However, one of the potential stumbling blocks for machine learning adoption in many applications is the issue of fairness, robustness, privacy, and explainability. Many machine learning algorithms are somewhat of a “black box”, with no easy way to determine why the algorithm produced the specific output. Explainability is key to challenge an AI/ML-based decision, especially in safety-critical applications from a SOTIF (Safety Of The Intended Functionality) perspective. This may be required, for example, to understand whether a correct decision was made in scenarios such as why a loan application was rejected by an AI/ML-based application, or why an autonomous vehicle in an accident decided to drive the route it did. Another important vector is achieving privacy in AI hardware architectures.	
A5.1	Architectures that natively output human-understandable rationales as to why a decision was reached
A5.2	Methods and architectures to add explainability to existing AI/ML-based solutions
A5.3	Architectures and algorithms to add fairness into machine learning algorithms and architectures with minimal accuracy or performance impact, even if the input data used in training may be biased
A5.4	Architectures robust against both natural variations of input data and adversarial attacks to ensure stability of machine learning and AI decisions. Also included under this are architectures capable of uncovering corruption/bias of training phase data and model integrity
A5.5	Enhancing robustness by building prior knowledge about the task to be learned and/or about the training data into the ML solution, e.g. training with a potentially limited set of input data supplemented by rules-based data, and/or pre-wiring the neural network, and/or data synthesis to enlarge training data sets
A5.6	Architectures with the ability to assess the functionality of its AI/ML process, so that a system with functional safety requirements can identify a malfunction and establish appropriate safety actions
A5.7	Privacy and confidentiality preserving AI architectures and systems
A6	**Interplay of AI and System Architecture/Microarchitecture Design**
Advances in AI/ML can significantly impact system design in at least two ways. First, AI/ML-based or AI/ML-inspired components can be directly used in hardware designs. For example, branch predictors, prefetchers, and other hardware predictors can be based on ML models, or can be optimized using ML models; scheduling and resource management at the core, chip, node and data center levels can be based on ML and improve over heuristic-based approaches. Second, AI/ML can be part of the system design process itself, e.g., providing optimizations at the system, architecture and micro-architecture levels that improve over traditional hardware design methods and flows. On the other hand, hardware and systems for AI/ML can benefit from groundbreaking advances in system-level architecture, memory systems and optimizations across multiple levels of the hardware/software stack that can directly impact future AI hardware on different design targets: performance, energy efficiency, security, etc. This interplay of AI and system level design is fundamental for design, construction and management of intelligent self-optimizing systems.	
A6.1	AI-based or AI-inspired components that can be used in hardware designs: e.g., hardware predictors, resource management controllers, etc.
A6.2	AI methods for optimization of hardware designs at the system, architecture and micro-architecture levels, excluding CAD software optimizations (which are part of the CADT thrust)
A6.3	AI-based design and optimization of AI accelerators and their integration in bigger systems
A6.4	Synergistic advances in system design and AI/ML to improve performance, energy-efficiency and security
A6.5	AI-assisted operating system, run-time system, and hardware operation for thread scheduling, DVFS, or power state transitions