SAND2022-16846C

ATHENA: Enabling Codesign for Next-Generation AI/ML Architectures

Mark Plagge*‡, Ben Feinberg*, John McFarland†, Fred Rothganger*, Sapan Agarwal*, Amro Awad†, Clayton Hughes*§ and Suma G. Cardwell*¶
*Sandia National Laboratories, Albuquerque, NM, USA
†North Carolina State University, Raleigh, NC, USA

Email: ‡mplagge@sandia.gov, §chughes@sandia.gov, ¶sgcardw@sandia.gov

Abstract—There is a growing market for technologies dedicated to accelerating Artificial Intelligence (AI) workloads. Many of these emerging architectures promise to provide savings in energy efficiency, area, and latency when compared to traditional CPUs for these types of applications. In particular, neuromorphic analog and digital technologies provide both low-power and configurable acceleration of challenging artificial intelligence (AI) algorithms. If designed into a heterogeneous system with other accelerators and conventional compute nodes, these technologies have the potential to augment the capabilities of traditional High Performance Computing (HPC) platforms. We present a codesign ecosystem that leverages an analytical tool, ATHENA, to accelerate design space exploration and evaluation of novel architectures.

Index Terms—Machine Learning, Codesign Tools, Neuromorphic Computing,

I. Introduction

For decades computing relied on the steady growth of performance provided with each new generation of CPUs. As this performance began to taper off, users turned to the wide Single Instruction, Multiple Data (SIMD) capabilities provided by GPUs to supplement the performance of some algorithms. This scaling issue is also driving a significant amount of research in new acceleration hardware designed for high-efficiency and performance [1]. Although the factors that pushed users to GPUs (Moore's Law and Dennard scaling) still exist for the time being [2], [3], advances in fabrication and circuit design are pushing accelerators closer to the CPU [4]–[7].

Moreover, high-performance computing is evolving beyond historically floating-point dense high-fidelity modeling and simulation to one that melds this traditional domain with machine learning models and large, often sparsely connected, volumes of data [8]–[10]. This has brought about a revolution in industry and academia, each proposing new, sometimes exotic, accelerators for these emerging computing domains. Because of these trends, the future likely entails system-on-package designs, blending multiple types of compute in a tightly-coupled package to enable orders of magnitude performance gains [11].

However, there are open questions about what can and should be offloaded to an accelerator and which accelerators make sense to co-package with a CPU. The diversity of workloads from home users, to datacenters, to HPC centers guarantees that there will not be a single solution that fits the needs of all stakeholders. Even looking at the application space of a single site like Argonne National Laboratory [12], Oak Ridge National Laboratory [13], or NERSC [14] makes it difficult to sketch out a possible solution. This vast design space begs for a co-design approach to discover best practices and inform cross-technology standards.

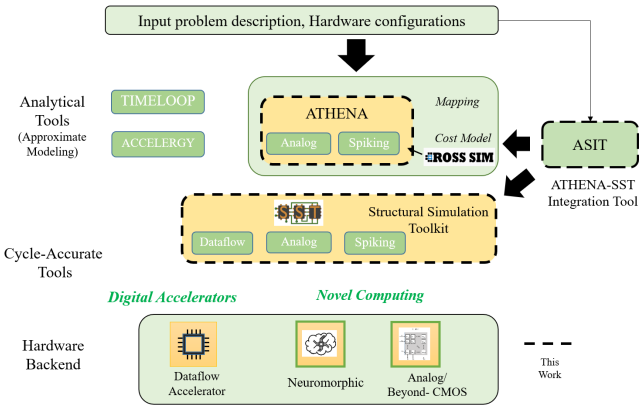

Fig. 1: Codesign Toolflow for this work. Our work enables fast exploration of architectures using ATHENA, and then leverages SST for detailed simulation of the 'best candidate' architecture. ASIT is a tool that enables seamless ATHENA-SST Integration.

This paper leverages ongoing investments in codesign simulation tools such as the Structural Simulation Tool (SST) [15] to provide a flexible cycle-approximate simulation foundation. The problem with these simula-

This work was supported by the DOE Advanced Simulation and Computing program. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. This paper describes technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

tions is that the time-to-solution can be hours or days, which makes identifying an optimal design point a years-long task. To reduce the design space, we developed Analytical Tool for Heterogeneous Neuromorphic Architectures (ATHENA), an analytical performance estimation tool which can be used to gather rapid insights into hardware performance. To help leverage ATHENA's results, we further developed ATHENA-SST Integration Tool (ASIT) to bridge ATHENA with lower-level Structural Simulation Toolkit (SST) simulations [16]. These tools focus on allowing rapid prototyping of emerging analog and neuromorphic architectures.

The following architectures can be evaluated using our codesign tools:

- Dataflow Architectures: Codesigned dataflow accelerator to enable AI technologies.
- Analog Accelerators: Analog neural network inference accelerators that leverage emerging analog devices [17]–[19].
- Spiking Architectures: A highly configurable model of spiking neural network (SNN) hardware, able to model STPU (Spiking Temporal Processing Unit) neuromorphic architecture [20], Intel's Loihi [21], IBM's TrueNorth [22], and future designs.

A. ML Accelerators

Challenges in power scaling of conventional digital computing have ushered in a new 'Golden Age in Computer Architecture' [23]. A wide variety of design tools have emerged to facilitate research into these novel and emerging computational architectures. Design tool support ranges from less precise analytical assessments to high fidelity simulations. Analytical approaches include Modeling Accelerator Efficiency via Spatio-Temporal Resource Occupancy (MAESTRO) and Eyeriss Eyexam [24], [25]. Other analytical tools like Timeloop [26], focus upon assessing properties of a hardware architecture such as the utilization of resources and identifying the optimal dataflow strategy for the architecture. Cycle-accurate tools on the other hand offer more accurate, but slower solutions with increased fidelity. Examples include Systolic CNN AcceLErator Simulator (SCALE Sim) and Nvidia Deep Learning Accelerator (NVDLA) [27], [28]. Scale SIM and NVDLA are largely focused on ML accelerator approaches such as systolic arrays and CNN accelerators. There is also growing interest in emerging neuromorphic architectures. For example, NeMo utilizes the Rensselaer's optimistic simulation system (ROSS) in a discrete event simulation tool to provide a functional simulation of the IBM TrueNorth spiking neuromorphic architecture [29]. There has also been development of additional tools to account for the performance of emerging device technologies such as CrossSim [30] and PUMA [31]. This spectrum of analytical modeling capabilities help enable co-design and the assessment of the impact of incorporating emerging ML accelerator and neuromorphic architectures into truly heterogeneous HPC systems [32].

B. Emerging Analog Accelerators

ATHENA leverages the Silicon-Oxide-Nitride-Oxide-Silicon (SONOS) floating-gate (FG) based accelerator design [17] as an initial exemplar for analog accelerators. These devices are fabricated in the embedded 40 nm process and enable 8-bit in situ matrix multiplications. The SONOS analog memory arrays are optimized for neural network inference and have been shown to achieve 20 TOPS/W on ResNet-50 with a $\gtrapprox 10\times$ gain in energy efficiency over state-of-the-art digital and analog inference accelerators [17].

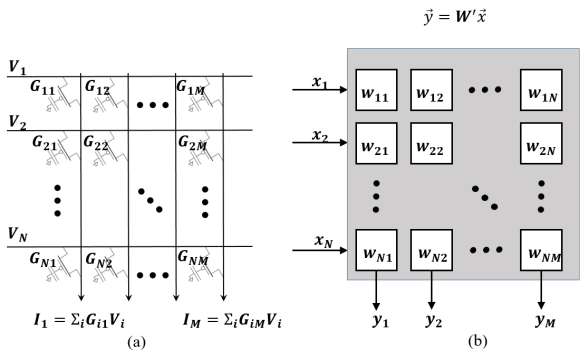

Fig. 2: Analog Crossbar Array (a) Schematic of an analog crossbar using floating-gates. (b) The analog crossbar essentially performs matrix multiplication, where $\vec{V}$ is equivalent to $\vec{x}$, G is equivalent to weight matrix W and output $\vec{I}$ is equivalent to $\vec{y}$.

SONOS floating-gates are a type of non-volatile memory device that enable matrix computation. Typically, digital dataflow accelerators use arrays of Multiply-Accumulate (MAC) units, but are limited by memory read/write and data movement costs. In analog Matrix-Vector Multiplication (MVM) arrays as shown in Figure 2(a), the input vector is encoded in the applied voltage to the rows $\vec{V}$, the weight matrix W is encoded in the memory cell conductance, and the dot product is the output $\vec{I}$ n the column currents. Using Kirchhoff's current law, products accumulate on the bit line. The current is then quantized using an Analog-to-Digital Converter (ADC) and sent to the next layer's array. This is equivalent to Figure 2(b) matrix multiplication, where $\vec{V}$ is equivalent to $\vec{x}$, G is equivalent to weight matrix W and output $\vec{I}$ is equivalent to $\vec{y}$.

Unlike digital dataflow accelerators, the SONOS analog accelerator tile contains multiple MVM arrays as seen in Fig. 6. This posed some challenges adapting to this novel accelerator as discussed in Section III.

II. ATHENA Overview

We intend ATHENA to be an end-to-end tool that enables evaluation of performance across a wide variety

of hardware designs. ATHENA provides the ability to quickly examine the performance in terms of latency, energy requirements, and network traffic limitations of novel analog neuromorphic hardware [33]. In addition, ATHENA will provide the ability to generate estimates of hardware from problems implemented in PyTorch, TensorFlow, and MLIR [34]–[36]. By providing rapid performance estimates, ATHENA will enable us to quickly prototype new hardware designs. In addition to the rapid performance prototyping provided by ATHENA, we will also leverage more traditional simulation tools as the neuromorphic architecture matures [16].

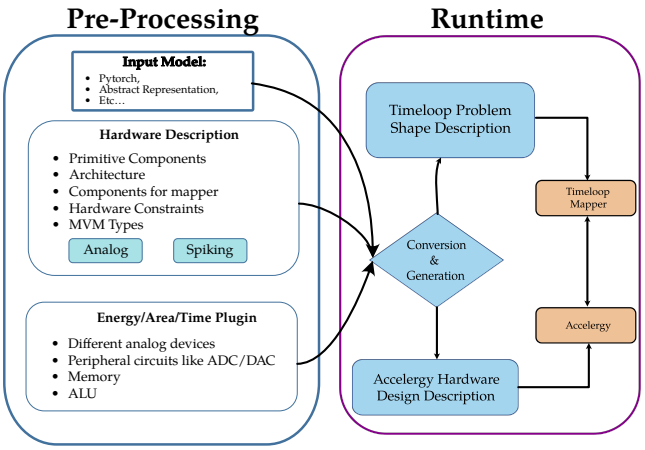

Fig. 3: ATHENA is an analytical tool that can be leveraged for design space exploration of novel architectures that use emerging devices. It describes the hardware specifications and energy/area/timing tables for beyond-CMOS analog accelerators.

Our work leverages the existing work shown in [26] with the Timeloop software tool, a dataflow style mapping tool that can estimate energy for CMOS logic based ML acceleration devices along with Accelergy [37] to assist with energy and area lookup tables. Timeloop supports single layer mapping of a neural network. ATHENA generates multiple hardware/problem inputs for Timeloop while adapting Accelergy hardware design descriptions to generate a hardware layout file for each layer of the run (changing active rows and columns), as shown in Figure 3. This allows for more dynamic energy estimates, enabling support for analog hardware while enabling code re-use. We have adapted Timeloop to provide estimates of a tiled analog ML acceleration device [17] through Accelergy's Energy Reference Table (ERT) and Area Reference Table (ART) integration. This implementation is a proof of concept to demonstrate suitability of analytical methods for estimating performance of analog devices.

III. Modeling the SONOS FG Analog Accelerator in ATHENA

ATHENA models analog MVM array based hardware leveraging the Accelergy+Timeloop software tools. Adaptation of the analog hardware required emulating the MVM array in the context of a CMOS-based hardware acceleration device. To accomplish this, we implemented a plugin and wrapper based system around the Accelergy+Timeloop framework. An overview of the ATHENA system is shown in Figure 5. ATHENA acts primarily as a "wrapper" to Accelergy and Timeloop, providing a user interface entry point as well as analysis tools. Furthermore, ATHENA provides an energy estimate plugin system to Accelergy, providing energy tables that the Timeloop mapper can use to estimate analog hardware performance, shown in a high level in Figure 4. ATHENA is able to coerce TimeLoop into estimating tiled analog hardware through the use of both the wrapper functionality and the Accelergy plugin.

Adapting a dataflow-centric analytical performance estimation tool to enable analog hardware estimation required several design changes. ATHENA works as a wrapper and plugin for Timeloop and Accelergy, allowing for the mapping of multi-component SONOS hardware tiles using Timeloop. A high level overview of ATHENA is shown in Figure 3, detailing the program's flow from input processing, output generation, and wrapping over Timeloop and Accelergy.

Within the SONOS hardware, MVM Arrays are combined into structures called "Tiles" as shown in Figure 6. ATHENA represents tiles as "Fat Processing Element (PE)s", given these tiles are much larger than a typical dataflow accelerator PE, wrapping energy and performance into a single logical cluster of PEs.

A. Hardware Description in ATHENA

To provide energy and latency estimates, Timeloop computes data movement across the buffers within a defined hardware device. At the PE level, Timeloop estimates the cycles needed to complete the required computation based on the number of available PEs, the buffer size and width, and the Network-on-Chip (NOC) bandwidth.

First, the tool must be aware of both the number of available compute units and the limits of the MVM array sizes. To represent a tile within these constraints we defined a cluster of PEs within the Timeloop hardware definition system. This cluster contains a set of "dummy" PEs, "dummy" buffers, along with peripheral components that make up the tile. To represent the MVM array within the cluster, a group of PEs coupled to scratchpad memory exist. The scratchpad memory connects to the ATHENA energy estimation tables, which provides data on a SONOS array's performance. The PEs exist to allow Accelergy to successfully map the input problem to hardware, however they report zero energy.

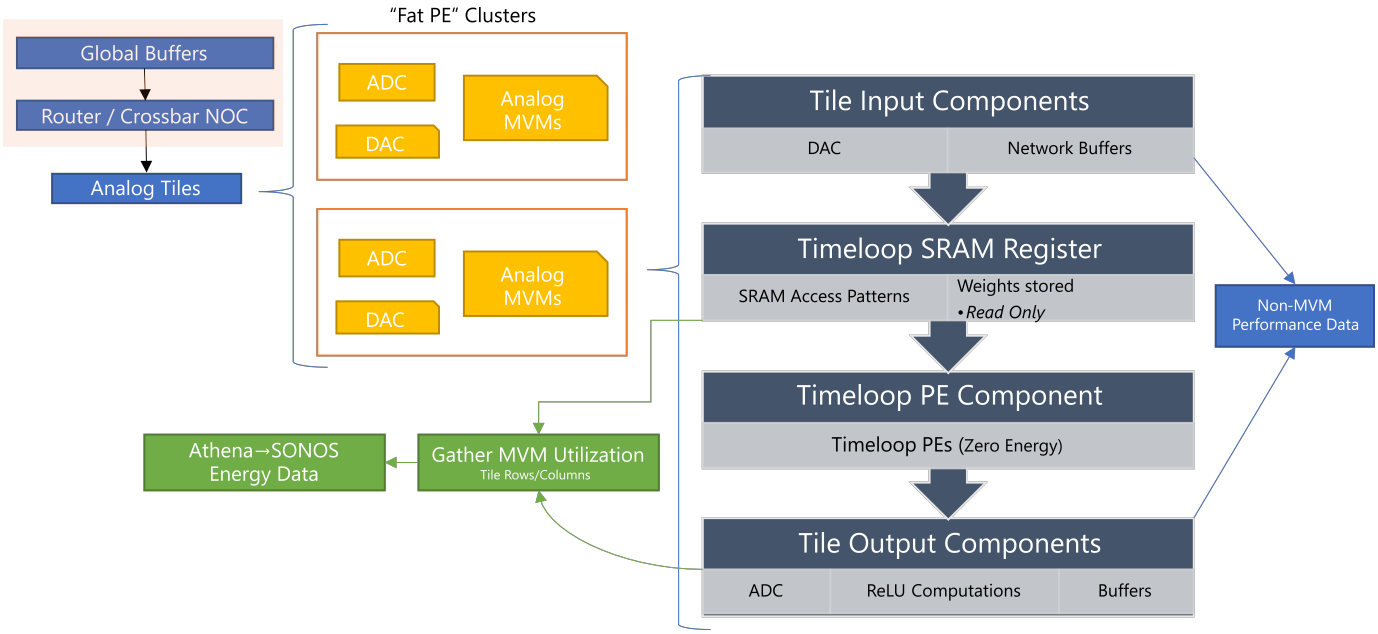

Fig. 4: Overview of ATHENA's hardware mapping and interface to Accelergy and Timeloop.

Each group of MVM cores, in the ATHENA hardware design mapping, contains an extra "scratchpad" memory buffer. This scratchpad memory buffer serves two purposes. First, the memory represents the analog MVM array's stored weight values. The memory is configured as read-only, and can only store the weights of the input problem. This memory layer also represents the intensity of the MVM computation when computing energy values. As Timeloop assumes a linear increase in energy based on the number of arithmetic operations performed, the memory layer's access patternzs are used to infer the total utilization of the MVM array per clock cycle.

One constraint of Accelergy and Timeloop is that they are unable to dynamically change the energy required for a single MAC operation when finding a valid and optimal mapping for a given hardware configuration. However, Accelergy will look up different energy values based on memory access patterns. ATHENA uses these memory layers to identify and compute the active rows and columns in the SONOS array, providing energy values to Timeloop that can be incorporated into the mapping cost model.

Each tile, from the perspective of Accelergy, is a cluster of PEs. This cluster can be mapped similarly to a standard dataflow-centric hardware design. ATHENA's design takes the memory access patterns reported by Accelergy, and uses them to represent tile access energy. This technique enables Timeloop's mapping algorithm to receive more dynamic energy costs when exploring the mapspace. In standard Timeloop, energy costs are fixed at runtime; each MAC operation cost is fixed based on the hardware class and definition within the energy look up tables or calling functions.

```
1  # Fat PE simplified example - MVM array
2  subtree:
3    # Non MVM Components of a tile
4    # Fat PE:
5    subtree: # Virtual cluster of MVM arrays
6      - name: MVMArray[0..4]
7      - local:
8          - name: mvm_in
9            class: SRAM
10           attributes:
11             sizeKB: 8
12         - name: sonos_access
13           class: sonos_array_pattern
14         - name: scratchpad[0..294911] # MVM Array Weights
15           class: sonos_dummy # Scratchpad containing
               ↪ weights
16           attributes:
17             action_name: read
18             network_drain: sonos_output_network
19         - name: MVM[0..294911]
20           class: compute
21           subclass: sonos_array #sonos array
22           attributes:
23             fat: 1
24             action_name: compute
25             n_mvm_rows: 1152
26             n_mvm_cols: 256
27         - name: sonos_output_network
28           class: sonos_tile_network
29         - name: ALUin
30           class: SRAM
31           network_fill: sonos_net_output
32           network_drain: alu_network
```

Fig. 5: Simplified example of a SONOS tile definition using Accelergy's hardware definitions with ATHENA's extensions.

Accelergy defines hardware as a set of hierarchical levels, with each level containing directly attached components. The topmost level can contain representations of off-chip memory, while the lowest level contains compute

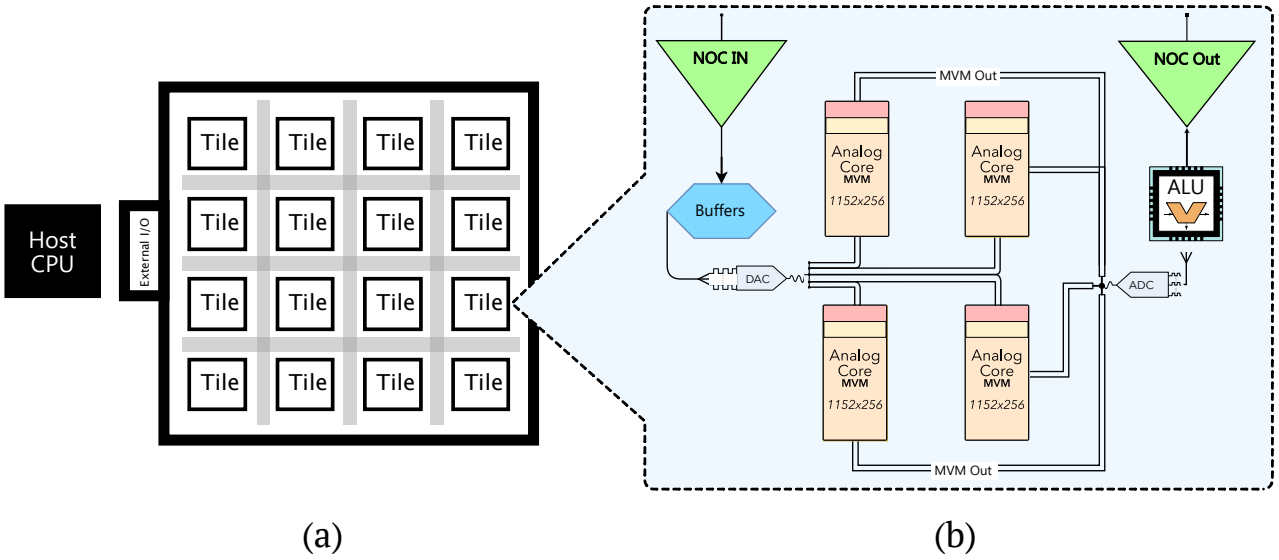

Fig. 6: Overview of SONOS Floating-gate based analog accelerator architecture and tile. (a) Tile architecture for the SONOS analog inference accelerator (b) Detailed diagram of SONOS accelerator tile. Analog accelerator arrays are computationally denser than a conventional digital accelerator PE with four 1152×256 MVM arrays in the same tile along with peripheral circuits, control unit and 64 kB local memory.

elements. Each level can be repeated, to represent multiple groups of connected components. In the case of the SONOS→Accelergy hardware definition, subtrees are defined for the MVM core with attached MVM-in and Arithmetic Logic Unit (ALU)-in buffers and the tile structure. Figure 5 is a simplified example of the hardware definition file used by ATHENA and Accelergy. Each tile has 4 MVM arrays, defined on line 7. Each MVM array has an SRAM buffer component, which is attached to a SONOS_access element. This element is treated as a zero energy memory buffer by Timeloop. However, the memory access patterns are used to find energy used based on the number of active SONOS rows and columns for a particular computation.

To allow Timeloop to map computation, the SONOS access element is attached to a scratchpad memory element which represents the intrinsic memory of the analog array. We restrict the data mapped to this element such that the weights or read-only portion of the input problem are stored within these values. There is one scratchpad entry per MVM cell which provides the weight table for Timeloop's mapping. Next, the MVM array is defined as a set of generic "compute" component classes. The compute class is a standard PE compute element in Timeloop. In this mapping, the ATHENA energy table reports zero energy for using this MVM array, as the inherent cost of computation is already reported by the ATHENA energy tables based on the memory access patterns via the component defined in line 13. The results of these values are sent via the on-tile network to the ALU-in SRAM buffer, which connects to the non-MVM component network. Figure 4 also provides a graphical overview of the structure of the SONOS hardware mapping in the ATHENA system. This diagram illustrates where non-MVM, and thus non-ATHENA components are connected to the ATHENA based hardware.

B. Energy/Area Tables

To enable fast energy and area estimation, ATHENA uses energy and area lookup tables. ATHENA generates these tables before each run, providing a fast way to estimate how much energy a particular input problem will consume. Data from this table was extracted from a simulation of the SONOS hardware, discussed in [17]. Energy provided by this simulation data contains per-array MVM energies based on the number of active rows and columns. Using this data, we generated energy reference tables for use with the ATHENA and Timeloop system.

ERTs consist of a set of values based on the actions performed by a particular hardware component. Each action has a corresponding value in the table, and energies are provided. The end result file is generated by Accelergy, using ATHENA as a plugin. A small sample of this end result data file is shown in Figure 7.

ATHENA uses the number of MAC operations required to compute a problem as the basis for estimating energy required for computation. Within the ERT table, as can be seen in Figure 7, each entry for a particular hardware element has a corresponding activity entry. Since we are adapting Timeloop's energy system to support dynamic MAC energies, MAC compute values are computed and

```
ERT:
  version: 0.3
  tables:
    - name: system_arch.chip.tile[0..255].Core[0..3].
      ↪ sonos_array_pattern
      actions:
        - name: read
          arguments:
            active_cols: 0
            active_rows: 0
          energy: 4.80143808e-07
        - name: read
          arguments:
            active_cols: 1
            active_rows: 0
          energy: 4.92697728e-07
        - name: read
          arguments:
            active_cols: 2
            active_rows: 0
          energy: 5.05251648e-07
        - name: read
          arguments:
            active_cols: 3
            active_rows: 0
          energy: 5.17805568e-07
        - name: read
          arguments:
            active_cols: 4
            active_rows: 0
          energy: 5.30359488e-07
```

Fig. 7: Small selection of an ATHENA+Accelergy ERT, with memory access patterns showing as active_rows and active_columns which provide energy values for the underlying MVM array.

```
ART:
  version: 0.3
  tables:
  - name: system_arch.chip.tile[0..255].D2A_NoC
    area: 84.992
  - name: system_arch.chip.tile[0..255].A2D_NoC
    area: 1972.25
  - name: system_arch.chip.chip_net
    area: 181
```

Fig. 8: Small selection of an ATHENA+Accelergy ART, with various example components and their corresponding areas.

stored as part of the memory read operations. As MAC operations occur while running the mapper, the energy values in the ERT are added to the running total.

The ART is similar to the ERT, in that it is also a generated lookup table. Instead of providing energy-action values it provides area estimates. As an example, Figure 8 shows some components of a ART. This file is generated by using Accelergy, with the ATHENA plugin providing other area estimates. When running ATHENA, the ART is used to provide estimates of the area of the processor. This functionality has the potential to be leveraged for a design space exploration tool. The total area of the processor could be added as a constraint when finding efficient hardware designs. Currently, in ATHENA the ART is an informational tool. Accurate area estimations need to be gathered for specific subcomponents. Furthermore, using ATHENA as a design space exploration tool will be examined as future work.

Typically, analytical tools for dataflow accelerators largely ignore the cost of computing activation functions. When examining the energy of dataflow accelerator hardware, the cost of activation functions is relatively small when compared to the cost of the large MAC operation energy cost. This however is important when considering binary neural networks or spiking neural networks. Binary neural networks leverage simplified activation functions which could affect the total energy of an analog accelerator device. In [38], switching from an 8-bit to 1-bit activation function improved overall energy costs of the matrix-vector operation from 2.850nJ to 0.198nJ. This speedup is due to the 1-bit activation functions being computed directly on the MVM array, rather than requiring a separate circuit after the ADC components.

ATHENA has preliminary support for activation function hardware, but this is still a work in progress. To add support for activation functions, we first compute the size of the output dimensions of the running layer. Given a word-size in an ALU and a bit-precision from the network, we can determine the count of activation function operations that need to be completed for a given input layer. This allows ATHENA to compute the energy required to run the activation function circuitry for a given input layer. To further enhance the feature set of ATHENA, we are currently adding support for binary activation functions. Binary activation functions are a way to create neuromorphic spiking hardware using analog devices. This functionality will be fully integrated in a future release of ATHENA.

IV. Results

To examine ATHENA's accuracy for the SONOS accelerator we compared our results with SONOS hardware [17]. Specifically, we compared the energy estimates of the analog MVM tiles. The energy use of these MVM tiles is approximately 1% of the total energy of the accelerator. Examining only the MVM arrays gives a deeper insight into ATHENA's ability to measure analog device energy accurately. These experiments used the convolutional layers in the VGG-16 network with a 224×224 input size.

Table I shows the results of using ATHENA's method of computing values compared against the results from the low-level SONOS simulation. We found that the MVM energy array accuracy ranged from approximately 22 % to 98 % over all layers. A major source of the inaccuracy in these results stems from the more dynamic way that the SONOS simulation engine maps workloads across the available compute resources on-chip. Using an ERT as a lookup method has the potential to lose some dynamic behavior of the underlying hardware.

This is a trade-off between accuracy and modeling speed. Look-up-tables will provide the most performance, but

SONOS Hardware Simulator [17]	ATHENA Result	Accuracy
1.07 pJ	2.14 pJ	32.80%
3.75 pJ	8.52 pJ	22.29%
2.10 pJ	2.13 pJ	98.81%
3.97 pJ	4.02 pJ	98.81%
1.04 pJ	1.06 pJ	97.60%
2.08 pJ	2.13 pJ	97.60%
2.08 pJ	2.13 pJ	97.60%
1.01 pJ	1.06 pJ	95.18%
2.03 pJ	2.13 pJ	95.18%
2.03 pJ	2.13 pJ	95.18%
0.48 pJ	0.53 pJ	90.25%
0.48 pJ	0.53 pJ	90.25%
0.48 pJ	0.53 pJ	90.25%

TABLE I: MVM energy estimates from ATHENA's MAC operation count method versus SONOS simulation against the VGG-16 convolutional neural network. Only convolutional layers are compared. Each energy result is from the MVM arrays in all tiles, and does not contain peripheral circuits.This gives us confidence in ATHENA's modeling of the SONOS FG analog accelerator.

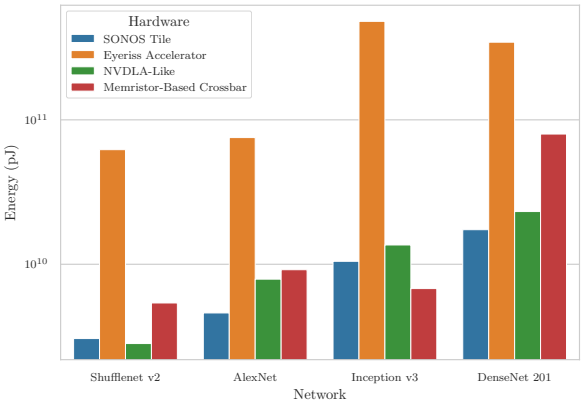

Fig. 9: A comparison of multiple convolutional neural network performances using ATHENA between a single SONOS tile accelerator, a PIM crossbar style analog accelerator, a NVDLA-Like dataflow accelerator, and an Eyeriss-like dataflow accelerator.

with potentially the least accuracy, especially when compared with lower-level simulation tools. This trade-off can be mitigated by leveraging more detailed simulation tools which can both inform the analytical model's accuracy and provide further insights into the overall hardware performance and behavior.

A. Using ATHENA to Compare Hardware Performance

To demonstrate some capabilities of ATHENA, we ran a performance comparison between a virtual Eyeriss-like hardware architecture [25] against the tile-based SONOS analog device with and without a 14-bit ReLU enabled and disabled. In Table II, we show the results of this comparison. In both of these results, the digital component ERTs were generated based on a 32 nm process node. SRAM memory values were previously gathered from CACTI [39], and NOC values were generated using data included in Accelergy. As shown in Table II, for inference over the convolutional layers of VGG-16, the analog hardware was between two and three orders of magnitude more energy efficient, even when accounting for the energy use of the peripheral support components such as DACs and ADCs.

VGG Layer	Eyeriss Energy (pJ)	SONOS Energy (pJ)	
		No ReLU	14-Bit ReLU
Conv. 1	925.06	42.235	42.574
Conv. 2	12196.60	139.119	146.344
Conv. 3	5636.16	75.684	79.296
Conv. 4	11384.15	139.694	146.919
Conv. 5	5524.96	37.442	41.054
Conv. 6	10739.76	74.273	81.498
Conv. 7	10739.76	74.273	81.498
Conv. 8	5174.99	23.995	27.607
Conv. 9	10710.05	71.804	79.029
Conv. 10	10710.05	71.947	79.172
Conv. 11	3015.38	13.763	15.569
Conv. 12	3015.38	13.617	15.423
Conv. 13	3015.38	13.617	15.423

TABLE II: Comparison of energy estimates generated through ATHENA over the convolutional layers of VGG-16 running on an Eyeriss-like hardware device and the SONOS analog accelerator. In addition to the Eyeriss-Like energy values and SONOS tile based architecture, we also show the effect of adding a 14-bit ReLU circuit to the SONOS tiles.

ATHENA's design intent is to allow large scale neuromorphic analog hardware co-design in an efficient and fast manner. To demonstrate the capabilities of ATHENA's modular design, we examined the relative performance of the dataflow architecture used to benchmark the underlying TimeLoop tool [26], a SONOS multi-tile hardware accelerator, a simple 512×512 memristor hardware accelerator, as well as an Eyeriss-like design original described in [25] with 14-bit floating point PEs.

We swept over a selection of common convolutional networks to create a sample of energy performance. We examined DenseNet201, Inception v3, Shufflenet v2, and AlexNet. Figure 9 shows the estimated energy each of these networks would consume based on a $224 \times 224 \times 3$ input image inference. The data provides insights into potential hardware design trade-offs when optimizing for energy.

We also examined wall clock time for these runs. Figure 10 shows the average wall clock time for each combination of hardware and network examined by ATHENA. The underlying mapper uses an embarrassingly parallel method of exploring the hardware mapspace, using threading to explore different map space paths. As such, adding cores will potentially improve the total mapspace search size, but will not result in faster wall-clock times. For all

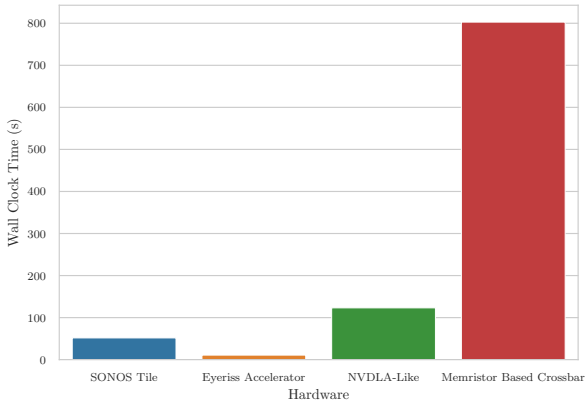

Fig. 10: Mean ATHENA wall-clock time in seconds across multiple hardware and network configurations.

of these runs, we configured the mapper to use 32 cores on a single machine running an Intel® Xeon® Gold 6230R CPU at 2.10GHz. ATHENA is able to produce a mapping using TimeLoop for analog hardware within ≈ 20 minutes when estimating performance against the DenseNet CNN, a network with 201 layers to evaluate. ATHENA's fastest time was on the order of 10 seconds. These times highlight how rapid prototyping can be achieved through the use of analytical methods. ATHENA provides this extremely rapid time to performance estimation, which opens up the potential for extensive design space exploration of analog hardware.

V. ASIT: Athena-SST Integration Tool

ASIT allows for the user to pass in a specific input problem, to be tested against different hardware architecture. ASIT identifies the series of operations needed to run these problems, then builds a set of possible compatible hardware architectures that could theoretically execute the problem set. ASIT then utilizes ATHENA in order to evaluate the performance execution of the input problem set over the specified hardware.

ASIT evaluates performance as the minimum of a selected metric. Currently, energy (as a function based on activity), latency (in clock cycles) and area can be selected individually as optimization targets. ASIT selects the "best" performing hardware design and generates a configuration for an SST component based on this hardware. This is implemented via a set of configuration parameters which define the component's capabilities in terms of compute size, memory capacity, and other values. In the case of the SONOS system the parameters are tile size, number of tiles, and cache sizes. As ATHENA is estimating novel analog hardware performance, the design of SST components to support these devices is critical to increase supported hardware in SST. A supported component must have a set of configurable parameters which ASIT can populate based on the ATHENA hardware design. To increase the number of supported hardware devices, more SST components need to be added, along with a corresponding set of ATHENA hardware and ASIT output configuration parameters.

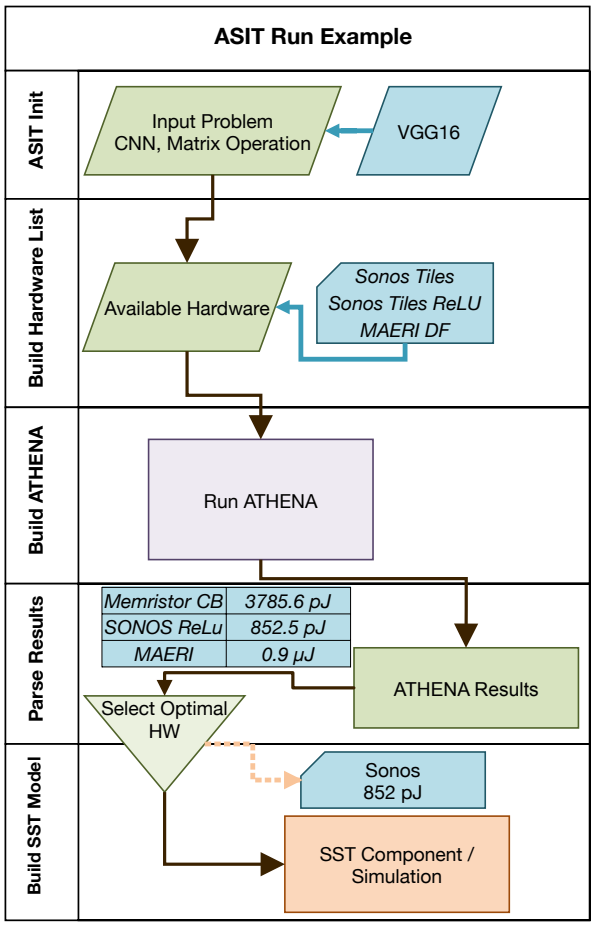

Fig. 11: ASIT Functional Diagram showing the example run using a MAERI style dataflow accelerator, a memristor crossbar based accelerator, and the SONOS with ReLU accelerator.

A. Results

We evaluated ASIT on three different hardware configurations: A SONOS-Tile based analog accelerator, a simple memristor crossbar array design, and one digital hardware configuration, which were then present to ATHENA to evaluate over. In Figure 11, we show where these example inputs are given, and what the resulting data points were. This run attempted to find the least energy use out of the three input hardware designs given. After generating an ATHENA hardware list and running the ATHENA and TimeLoop mapping systems, the results were sent to the ASIT system. In this example, ASIT chose the SONOS tile-based device based on the low reported energy use from the ATHENA run. ASIT then generated a template

file for SST which could be used to generate more accurate results, or run in a larger SST simulation to generate a heterogeneous system result.

B. Extending ASIT

ASIT is intended to provide a link between an analytical performance estimation tool, and the SST simulation engine. In the future, ASIT will be able to generate an SST model based on more primitive SST component templates, allowing for full simulation of a variety of hardware. ASIT will allow for integration of any device supported both by ATHENA and SST template enabled components, specifically targeting architectures including: tile-based digital components, analog accelerators, and neuromorphic accelerators. With the addition of these hardware components to ATHENA and SST, ASIT will allow for the evaluation of efficiency of these hardware designs before implementing them into SST. This would allow for an array of different architectures to be filtered through so the best-fit option is the one chosen to be fully simulated.

Eventually, ASIT will enable rapid prototyping of analog devices for specific applications with a higher fidelity simulation step in the loop. This could pave the way to a true design space exploration software system for analog and analog-based neuromorphic hardware.

VI. Future HPC Impact

The post-exascale era in computing will require heterogeneous node and system architectures to achieve power, cost, reliability, and usability requirements while maintaining the rate of increase of application performance. As modern High Performance Computing (HPC) systems increase overall compute power, the physical number of compute nodes has been decreasing while the number of compute cores and accelerator cores have been increasing. In the Top 500 list [40], [41], the highest performing systems are no longer those with only the highest node count. The highest performing system are those that leverage smaller counts of more powerful individual nodes with high processor core counts, usually leveraging dedicated acceleration hardware [42]. Previous and current investments develop strategies to target heterogeneous nodes that use well-understood computing components (CPUs alongside GPUs). However, the slowing of Moore's law is driving the computing community toward more specialized forms of compute to achieve performance. This has led to an explosion of different accelerator types actively used in industry. Google's Tensor Processing Unit (TPU)s [43], the Nvidia Deep Learning Accelerator (NVDLA) accelerator, the Cerebras [44] wafer-scale processor, the Mythic analog floating-gate accelerator [45], and a variety of other devices all have unique approaches to improving the performance of specific aspects of computation.

It is imperative that we address the challenges of heterogeneous compute in the post-exascale era in the long-term. Leveraging the power of accelerators to improve the performance of compute-intensive applications is a key component of this effort. Using novel approaches, such as neuromorphic and analog crossbars, that improve performance through the reduction of the von Neumann bottleneck are key components of the future of computing.

These new devices require new co-design approaches which include identification/mapping and architectural exploration. These two approaches are complimentary in our co-design methodology to design heterogeneous architectures that incorporate novel computing paradigms. Future work will include seamlessly integrating ATHENA with SST, to evaluate heterogeneous workloads and dynamically map workloads at run time.

Given an input neural network or other MVM operation, the ATHENA system will generate multiple candidate hardware designs, search through candidates to recommend one or more hardware devices, then through ASIT generate an SST discrete event simulation component. The user is then able to evaluate the original input problem against the selected hardware in either SST or even potentially lower-level simulation tools.

In this work, we showcased the first components of this eventual end-to-end system. ATHENA, leveraging the mapping abilities of the underlying TimeLoop software, generates fast performance estimates of how analog devices would perform against convolutional and matrix operation problems. ASIT provides a way to search across hardware elements, search for an optimal set of hardware designs, and generate a configuration for the SST software. These components form the backbone of a complete analog end-to-end co-design exploration tool.

Current work in progress includes developing analog models in SST and leveraging ATHENA as performance estimator to SST, to enable approximate modeling of performance before detailed cycle-accurate simulations in SST. This work includes simulation models for tile-based analog devices, traditional crossbar devices, and neuromorphic hardware devices. With the inclusion of mapping as an integrated component to SST will enable direct generation of an accelerator component for SST. Since SST provides the ability to map at multiple hardware resolutions including large-scale HPC networks, this integration will provide a full discrete event based simulation of heterogeneous HPC systems, complete with energy and latency estimates based on real-world workloads. This enhanced simulation capability could pave the way for the next generation of efficient hybrid supercomputer designs.

Further work includes allowing ASIT to search across more fine-grained hardware configurations. Currently, ASIT searches against a set of pre-configured hardware designs. Allowing ASIT to vary specific elements of these designs in a limited fashion could further enable rapid hardware design exploration. Adding this capability would require a rapid multi-variable optimization search across not just large hardware configurations but also individual

component sizes. This would make ATHENA capable of large-scale heterogeneous hardware design optimization.

These expanded tool capabilities will identify, evaluate, design, and analyze next-generation architectures specialized for specific workloads. It is a critical step in enabling co-design of next generation heterogeneous computing platforms, from HPC to the edge, with broad impacts in scientific computing, machine learning and performance and SWaP-constrained edge applications.

References

[1] Y. S. Shao and D. Brooks, "Why accelerators, now?" in Research Infrastructures for Hardware Accelerators. Springer, 2016, pp. 1–12.

[2] H. Esmaeilzadeh, E. Blem, R. St. Amant, K. Sankaralingam, and D. Burger, "Dark silicon and the end of multicore scaling," in Proceedings of the 38th annual international symposium on Computer architecture, 2011, pp. 365–376.

[3] W. Liu, F. Lombardi, and M. Shulte, "A retrospective and prospective view of approximate computing [point of view]," Proceedings of the IEEE, vol. 108, no. 3, pp. 394–399, 2020.

[4] Y. Sun, N. B. Agostini, S. Dong, and D. Kaeli, "Summarizing cpu and gpu design trends with product data," arXiv preprint arXiv:1911.11313, 2019.

[5] J. Hennessy and D. Patterson, "A new golden age for computer architecture: domain-specific hardware/software co-design, enhanced," in ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA), 2018.

[6] S. Sarangi and B. Baas, "Deepscaletool: A tool for the accurate estimation of technology scaling in the deep-submicron era," in 2021 IEEE International Symposium on Circuits and Systems (ISCAS), 2021, pp. 1–5.

[7] J. Schemmel, J. Fieres, and K. Meier, "Wafer-scale integration of analog neural networks," in 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence). IEEE, 2008, pp. 431–438.

[8] A. Sriraman and A. Dhanotia, "Understanding acceleration opportunities at hyperscale," IEEE Micro, vol. 41, no. 3, pp. 34–41, 2021.

[9] R. Nair, "Big data needs approximate computing: technical perspective," Communications of the ACM, vol. 58, no. 1, pp. 104–104, 2014.

[10] M. M. S. Aly, T. F. Wu, A. Bartolo, Y. H. Malviya, W. Hwang, G. Hills, I. Markov, M. Wootters, M. M. Shulaker, H.-S. P. Wong et al., "The n3xt approach to energy-efficient abundant-data computing," Proceedings of the IEEE, vol. 107, no. 1, pp. 19–48, 2018.

[11] W. Haensch, "Scaling is over — what now?" in 2017 75th Annual Device Research Conference (DRC), 2017, pp. 1–2.

[12] "Argonne leadership computing facility 2021 annual report," Argonne Leadership Computing Facility, Tech. Rep., 2022.

[13] "US Department of Energy, Office of Science High Performance Computing Facility Operational Assessment 2020 Oak Ridge Leadership Computing Facility," Oak Ridge Leadership Computing Facility, Tech. Rep. ORNL/SPR-2021/1950, 2021.

[14] "National energy research scientific computing center 2020 annual report," National Energy Research Scientific Computing Center, https://www.nersc.gov/assets/Uploads/NERSC-2020-Annual-Report-Final.pdf, Tech. Rep., 2020.

[15] A. F. Rodrigues, K. S. Hemmert, B. W. Barrett, C. Kersey, R. Oldfield, M. Weston, R. Risen, J. Cook, P. Rosenfeld, E. Cooper-Balis et al., "The structural simulation toolkit," ACM SIGMETRICS Performance Evaluation Review, vol. 38, no. 4, pp. 37–42, 2011.

[16] B. Feinberg, S. Agarwal, M. Plagge, F. Rothganger, S. Cardwell, and C. Hughes, "Modeling analog tile-based accelerators using sst." Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), Tech. Rep., 2022.

[17] T. P. Xiao, B. Feinberg, C. H. Bennett, V. Agrawal, P. Saxena, V. Prabhakar, K. Ramkumar, H. Medu, V. Raghavan, R. Chettuvetty et al., "An accurate, error-tolerant, and energy-efficient neural network inference engine based on sonos analog memory," IEEE Transactions on Circuits and Systems I: Regular Papers, 2022.

[18] S. Agarwal, A. Hsia, R. Jacobs-Gedrim, D. R. Hughart, S. J. Plimpton, C. D. James, and M. J. Marinella, "Designing an analog crossbar based neuromorphic accelerator," in 2017 Fifth Berkeley Symposium on Energy Efficient Electronic Systems & Steep Transistors Workshop (E3S). IEEE, 2017, pp. 1–3.

[19] T. P. Xiao, C. H. Bennett, B. Feinberg, S. Agarwal, and M. J. Marinella, "Analog architectures for neural network acceleration based on non-volatile memory," Applied Physics Reviews, vol. 7, no. 3, p. 031301, 2020. [Online]. Available: https://doi.org/10.1063/1.5143815

[20] A. J. Hill, J. W. Donaldson, F. H. Rothganger, C. M. Vineyard, D. R. Follett, P. L. Follett, M. R. Smith, S. J. Verzi, W. Severa, F. Wang et al., "A spike-timing neuromorphic architecture," in 2017 IEEE International Conference on Rebooting Computing (ICRC). IEEE, 2017, pp. 1–8.

[21] 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.

[22] 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.

[23] J. Dean, D. Patterson, and C. Young, "A new golden age in computer architecture: Empowering the machine-learning revolution," IEEE Micro, vol. 38, no. 2, pp. 21–29, 2018.

[24] H. Kwon, M. Pellauer, and T. Krishna, "Maestro: an open-source infrastructure for modeling dataflows within deep learning accelerators," arXiv preprint arXiv:1805.02566, 2018.

[25] Y.-H. Chen, T.-J. Yang, J. Emer, and V. Sze, "Eyeriss v2: A flexible accelerator for emerging deep neural networks on mobile devices," IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol. 9, no. 2, pp. 292–308, 2019.

[26] A. Parashar, P. Raina, Y. S. Shao, Y.-H. Chen, V. A. Ying, A. Mukkara, R. Venkatesan, B. Khailany, S. W. Keckler, and J. Emer, "Timeloop: A systematic approach to dnn accelerator evaluation," in 2019 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS). IEEE, 2019, pp. 304–315.

[27] A. Samajdar, Y. Zhu, P. Whatmough, M. Mattina, and T. Krishna, "Scale-sim: Systolic cnn accelerator simulator," arXiv preprint arXiv:1811.02883, 2018.

[28] NVIDIA Inc. Nvdla.org. Accessed: 2022, October 20. [Online]. Available: http://nvdla.org/index.html

[29] M. Plagge, C. D. Carothers, E. Gonsiorowski, and N. Mcglohon, "Nemo: A massively parallel discrete-event simulation model for neuromorphic architectures," ACM Transactions on Modeling and Computer Simulation (TOMACS), vol. 28, no. 4, pp. 1–25, 2018.

[30] S. J. Plimpton, S. Agarwal, R. Schiek, and I. Richter, "Crosssim," Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), Tech. Rep., 2016.

[31] A. Ankit, I. E. Hajj, S. R. Chalamalasetti, G. Ndu, M. Foltin, R. S. Williams, P. Faraboschi, W.-m. W. Hwu, J. P. Strachan, K. Roy et al., "Puma: A programmable ultra-efficient memristor-based accelerator for machine learning inference," in Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems, 2019, pp. 715–731.

[32] S. G. Cardwell, C. Vineyard, W. Severa, F. S. Chance, F. Rothganger, F. Wang, S. Musuvathy, C. Teeter, and J. B. Aimone, "Truly heterogeneous hpc: Co-design to achieve what science needs from hpc," in Smoky Mountains Computational Sciences and Engineering Conference. Springer, 2020, pp. 349–365.

[33] S. Cardwell, M. Plagge, C. Hughes, F. Rothganger, S. Agarwal, B. Feinberg, A. Awad, J. McFarland, and L. Parker, "Athena: Analytical tool for heterogeneous neuromorphic architectures."

Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), Tech. Rep., 2022.
[34] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga et al., "Pytorch: An imperative style, high-performance deep learning library," vol. 32. Curran Associates, Inc., 2019, pp. 8024–8035.
[35] C. Lattner, M. Amini, U. Bondhugula, A. Cohen, A. Davis, J. Pienaar, R. Riddle, T. Shpeisman, N. Vasilache, and O. Zinenko, "MLIR: Scaling compiler infrastructure for domain specific computation," in 2021 IEEE/ACM International Symposium on Code Generation and Optimization (CGO), 2021, pp. 2–14.
[36] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/
[37] Y. N. Wu, J. S. Emer, and V. Sze, "Accelergy: An architecture-level energy estimation methodology for accelerator designs," in 2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD). IEEE, 2019, pp. 1–8.
[38] J. B. Aimone, C. H. Bennett, S. G. Cardwell, R. A. Dellana, and P. Xiao, "Mosaic, the best of both worlds: Analog devices with digital spiking communication to build a hybrid neural network accelerator," Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), Tech. Rep., 2020.
[39] G. Reinman and N. P. Jouppi, "Cacti 2.0: An integrated cache timing and power model," Tech. Rep., 2000.
[40] E. Strohmaier, "Top500 supercomputer," in Proceedings of the 2006 ACM/IEEE Conference on Supercomputing, ser. SC '06. New York, NY, USA: Association for Computing Machinery, 2006, p. 18–es. [Online]. Available: https://doi.org/10.1145/1188455.1188474
[41] J. Dongarra and P. Luszczek, TOP500. Boston, MA: Springer US, 2011, pp. 2055–2057.
[42] W. Zheng, "Research trend of large-scale supercomputers and applications from the top500 and gordon bell prize," Science China Information Sciences, vol. 63, no. 7, pp. 1–14, 2020.
[43] N. P. Jouppi, C. Young, N. Patil, D. Patterson, G. Agrawal, R. Bajwa, S. Bates, S. Bhatia, N. Boden, A. Borchers et al., "In-datacenter performance analysis of a tensor processing unit," in Proceedings of the 44th annual international symposium on computer architecture, 2017, pp. 1–12.
[44] Cerebras Inc. Cerebras AI Homepage. Accessed: 2022, October 29. [Online]. Available: https://www.cerebras.net/
[45] M. Demler, "Mythic multiplies in a flash," Tech. Rep., 2018.