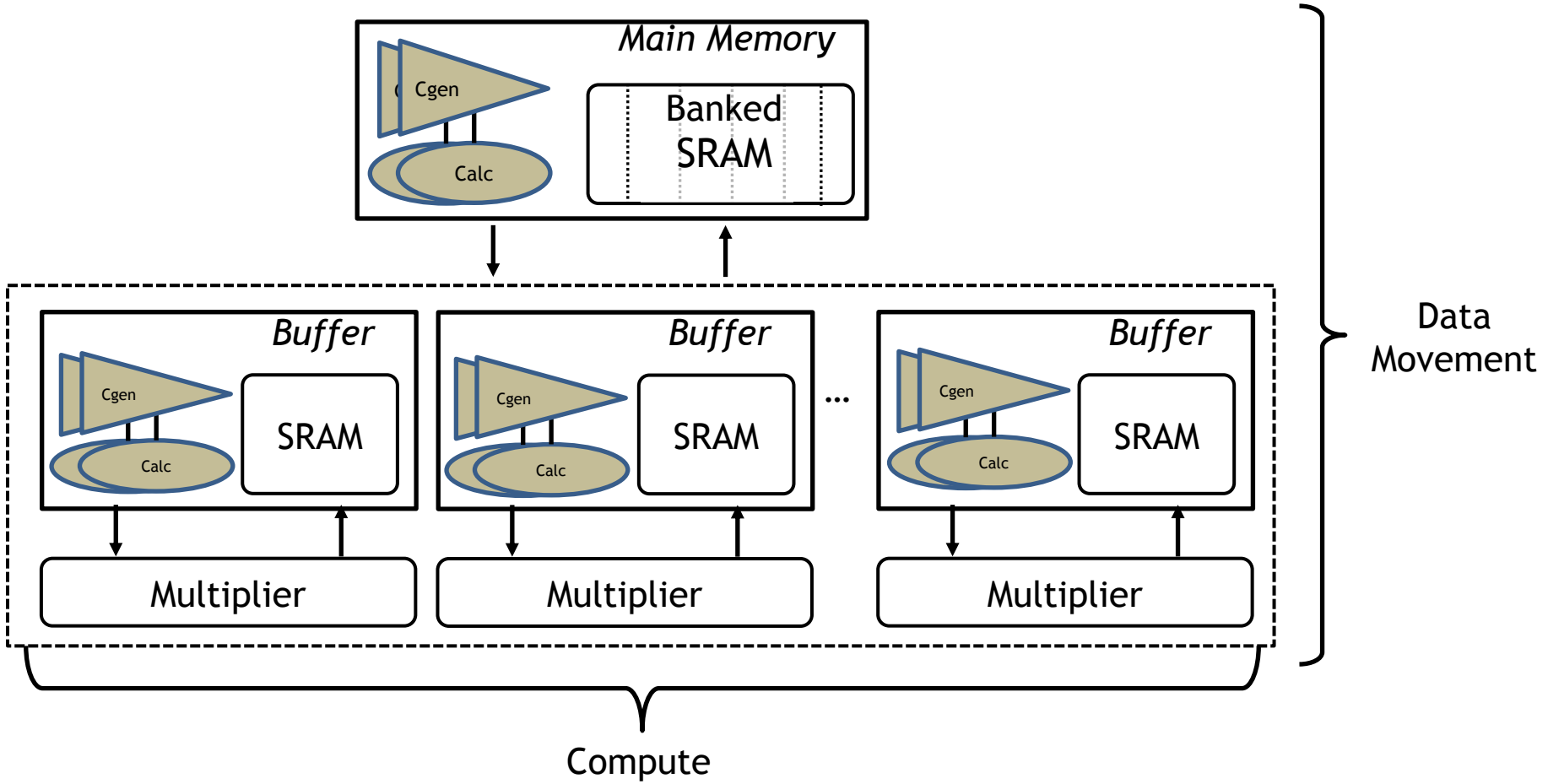
Exploring Energy Efficient Analog Neural Network Accelerator Designs

Joel Emer & Vivienne Sze

October 2022

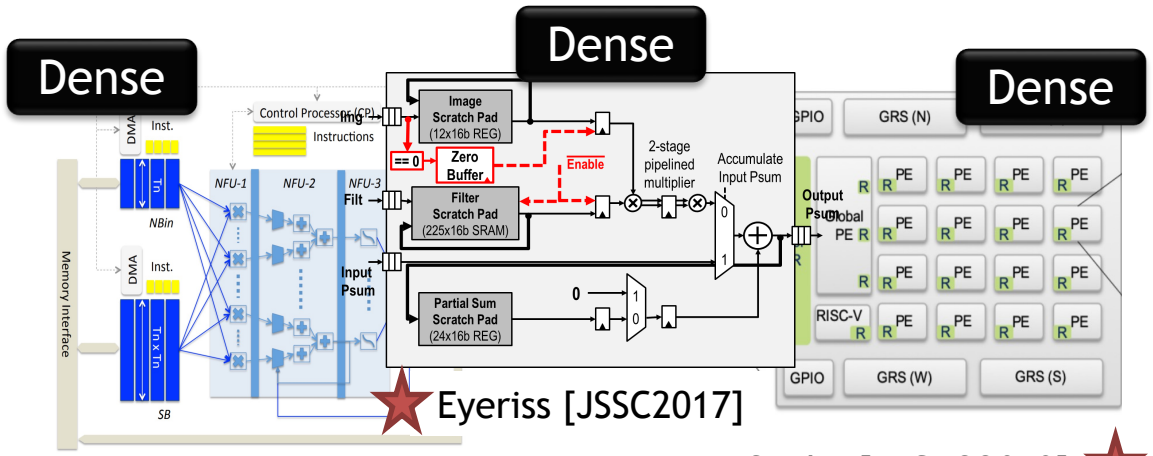
Typical Deep Neural Network Accelerator Organization



Objective: Achieve high energy efficiency and processing speed

Design Space - Digital + Dense

Digital-Compute Accelerator Designs



DianNao [ASPLOS2014]

Eyeriss [JSSC2017]

Simba [MICRO2019]

Works we contributed to

Classification of Important Design Aspects

Dataflow

Processing schedule for uncompressed data movement and compute in time and space

Architecture

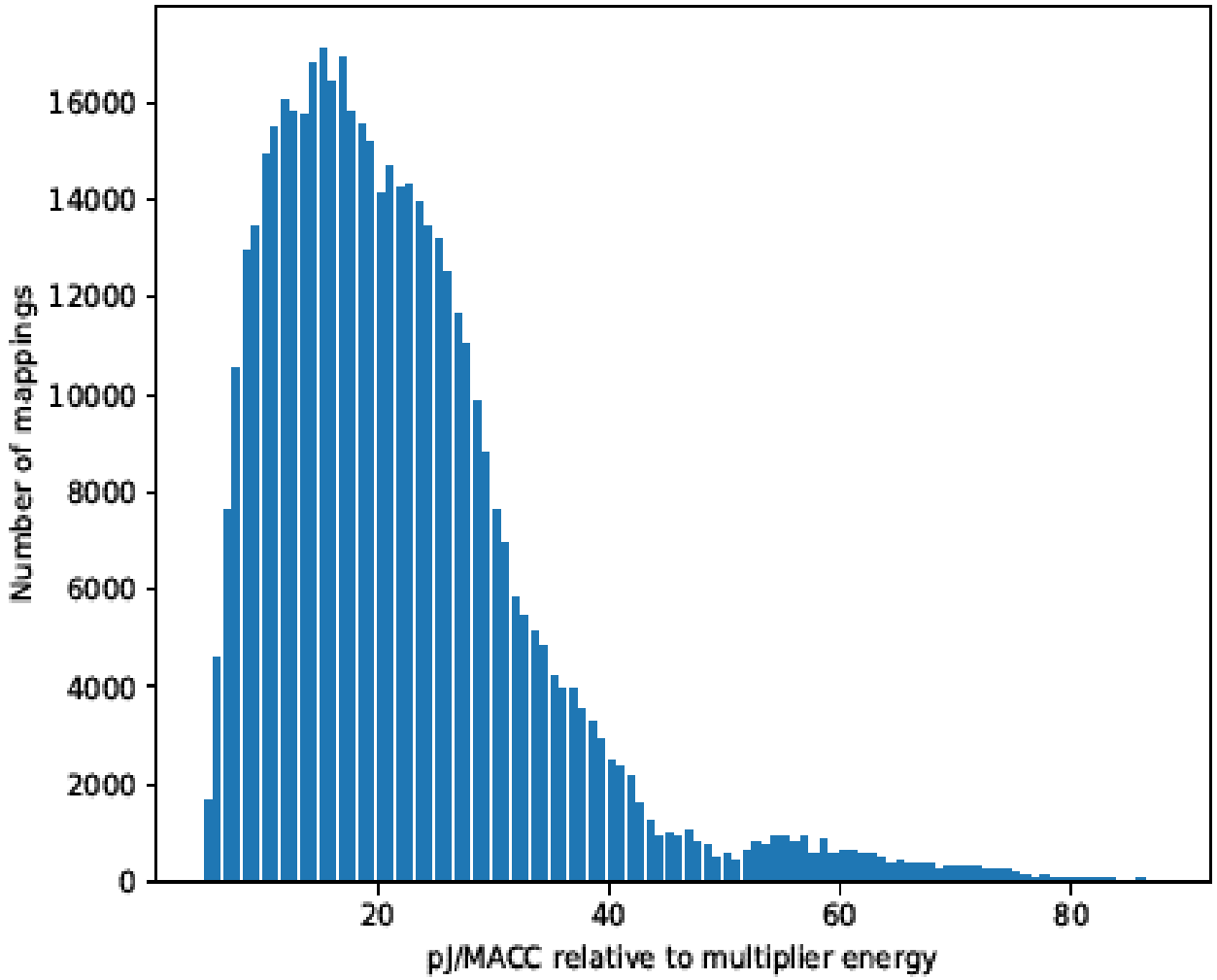
Detailed characterization of hardware, *e.g.*, storage sizes and access and compute widths

Technology

Assessment of costs at device level, *e.g.*, *area*, *energy*, *latency*

Mapping Choices

Energy-efficiency of peak-perf mappings of a single architecture/problem



480,000 mappings shown

Spread: 19x in energy efficiency

Only 1 is optimal, 9 others within 1%

A model needs a mapper to evaluate a DNN workload on an architecture

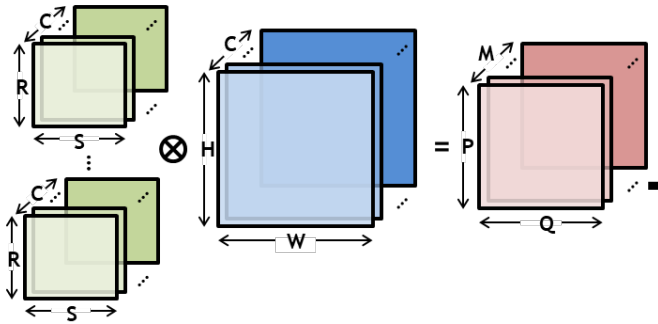
6,582 mappings have min. DRAM accesses but vary 11x in energy efficiency

A mapper needs a good cost model to find an optimal mapping

Source: Parashar, Timeloop

Timeloop: Analytical Modeling for Tensor Accelerators

Workload



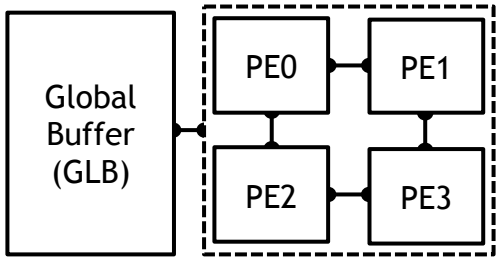
Mapping

```
for (n=0; n<N; n++) {
    for (m=0; m<M; m++) {
        for (q=0; q<Q; q++) {
            for (p=0; p<P; p++) {

                O[n][m][p][q] = B[m];
                for (r=0; r<R; r++) {
                    for (s=0; s<S; s++) {
                        for (c=0; c<C; c++) {
                            O[n][m][p][q] += I[n][c][Up+r][Uq+s] * F[m][c][r][s];
                        }
                    }
                }

                O[n][m][p][q] = Activation(O[n][m][p][q]);
```

Architecture



Timeloop

Energy

Cycles

Accelergy

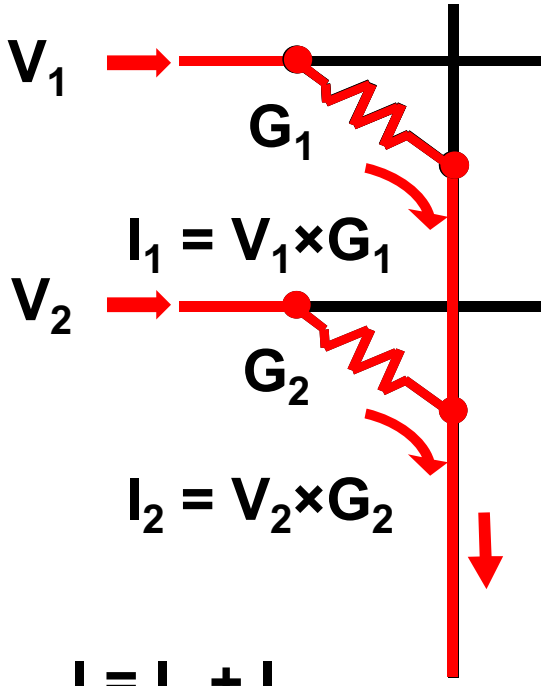
Storage

Compute logic

…

Processing In Memory (PIM)

Activation is input voltage (V_i)
Weight is resistor conductance (G_i)



Psum is output current

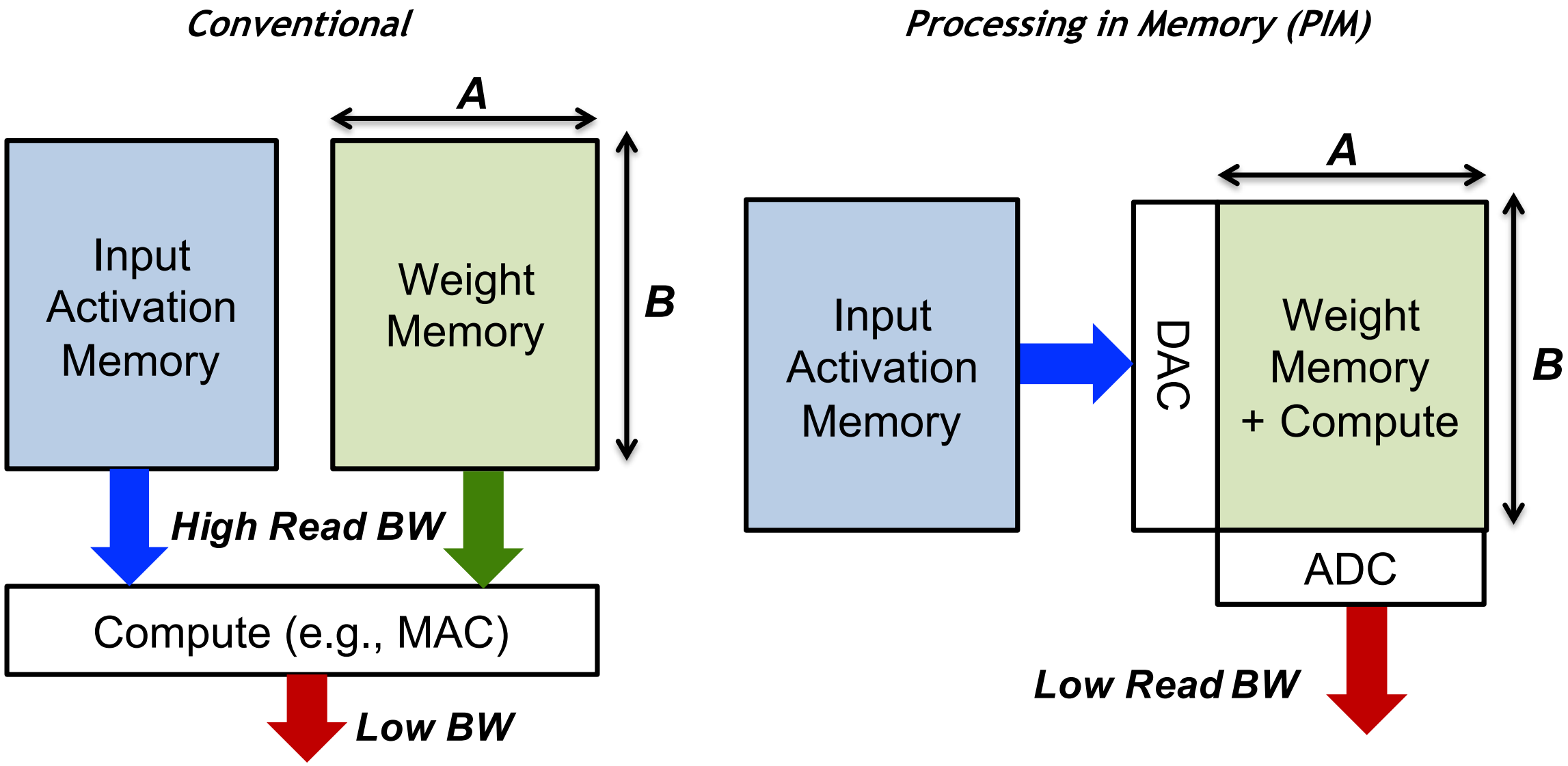
$I = I_1 + I_2$
$= V_1 \times G_1 + V_2 \times G_2$

Image Source: [**Shafiee**, *ISCA* 2016]

- Reduce data movement by **moving compute into memory**
- Compute with memory storage elements
- **Analog Compute**
  - Activations, weights and/or partial sums are encoded with analog voltage, current, or resistance
  - Increased sensitivity to circuit non-idealities
  - A/D and D/A circuits to interface with digital domain
- Leverage **emerging memory device technology**

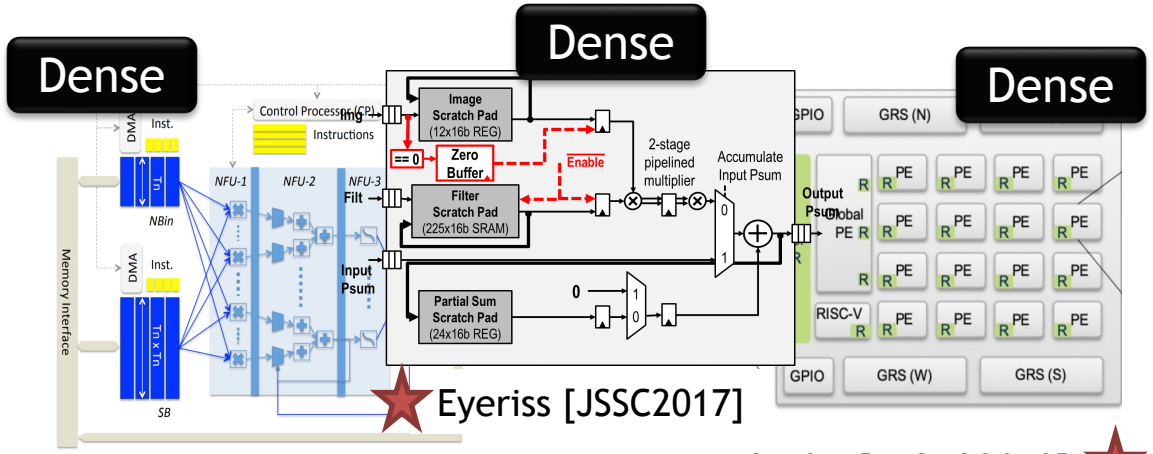
eNVM:[**Yu**, *PIEEE* 2018], SRAM:[**Verma**, *SSCS* 2019]

Conventional vs Processing in Memory (PIM)



Design Space + Analog/Dense

Digital-Compute Accelerator Designs

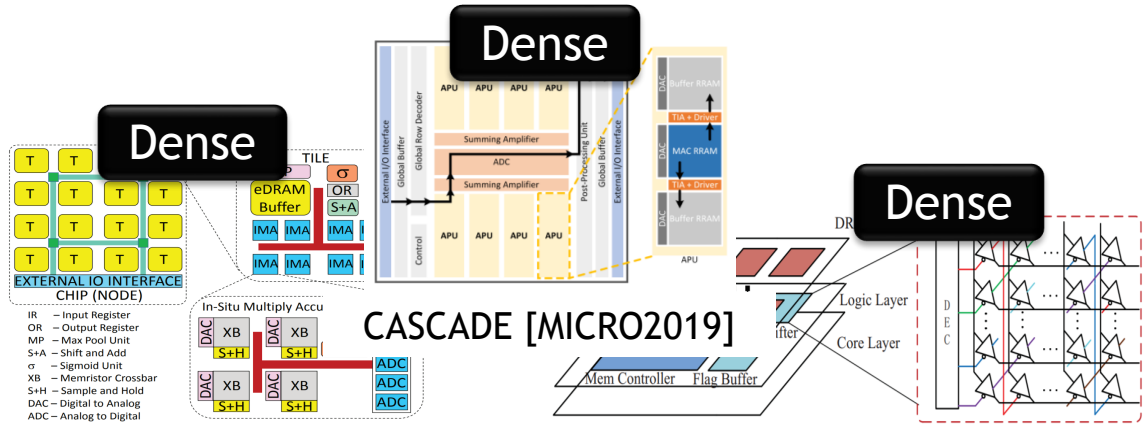


DianNao [ASPLOS2014]

Eyeriss [JSSC2017]

Simba [MICRO2019]

Analog-Compute Accelerator Designs



ISAAC [ISCA2016]

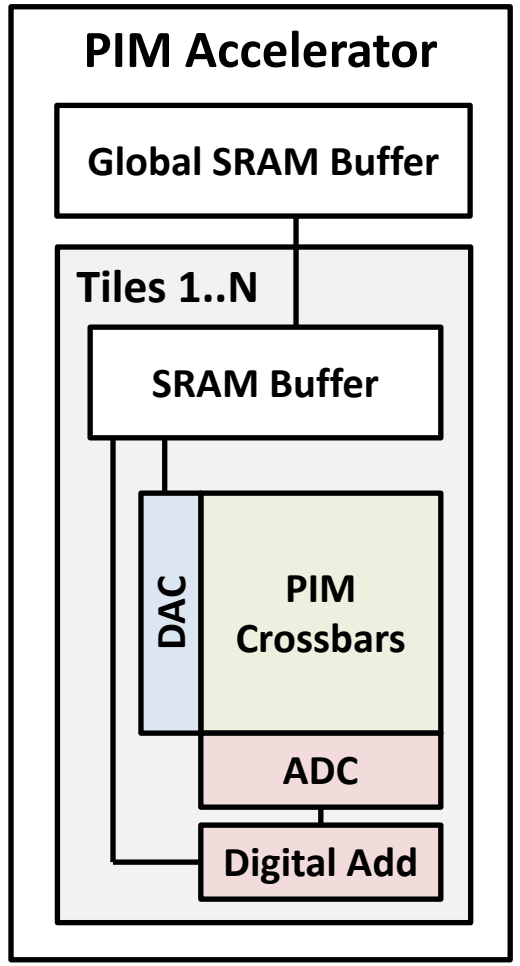
CASCADE [MICRO2019]

DrAcc [DAC2018]

★ *Works we contributed to*

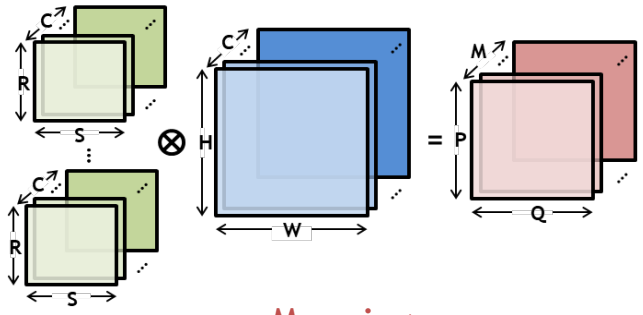
Pieces to Evaluate a PIM Design

- PIM Hardware Components
  - Components determine the basic characteristics of the system
  - Choice of ADC, DAC, memristors
- System Architecture
  - Architecture design sets data movement and reuse opportunities
  - Choices of crossbar dimensions, buffer hierarchy, data movement
- Workload
  - Workload characteristics set interaction with the architecture
- Mapping
  - Workloads can be mapped to architectures in many ways, each with energy, throughput, and utilization tradeoffs.



Timeloop: Analytical Modeling for Tensor Accelerators

Workload



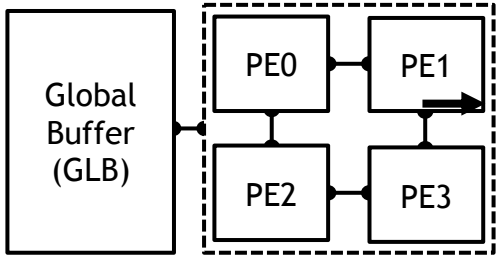
Mapping

```
for (n=0; n<N; n++) {
    for (m=0; m<M; m++) {
        for (q=0; q<Q; q++) {
            for (p=0; p<P; p++) {

                O[n][m][p][q] = B[m];
                for (r=0; r<R; r++) {
                    for (s=0; s<S; s++) {
                        for (c=0; c<C; c++) {
                            O[n][m][p][q] += I[n][c][Up+r][Uq+s] * F[m][c][r][s];
                        }
                    }
                }

                O[n][m][p][q] = Activation(O[n][m][p][q]);
```

Architecture



Timeloop

Support for large scale arrays

Energy

Cycles

Accelergy

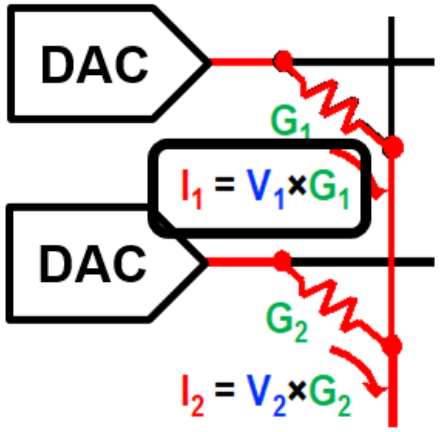
Storage

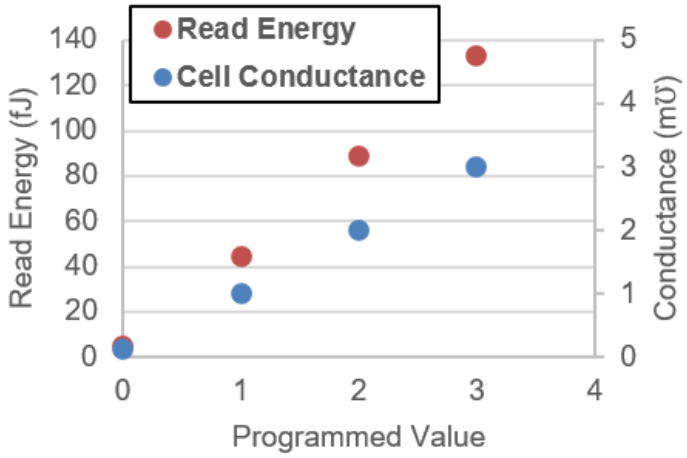
Compute logic

Analog logic-array, ADC/DAC

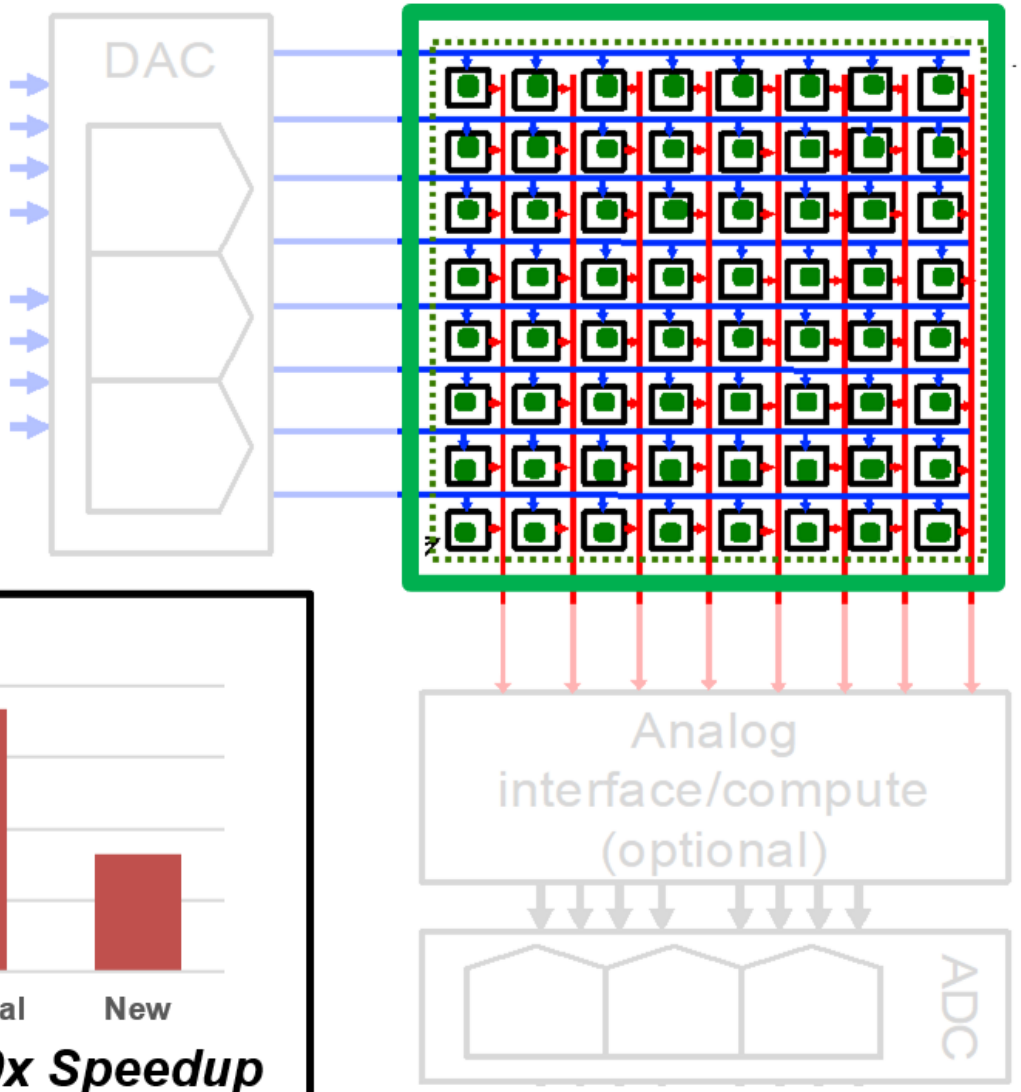
Timeloop + Memristive Crossbars

Data + Device Dependent Memristor Energy

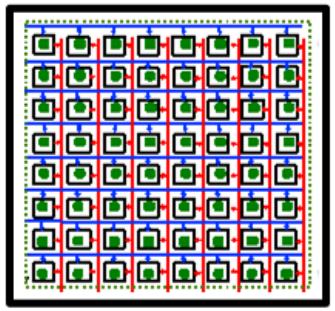




Scaling with Device, Workload

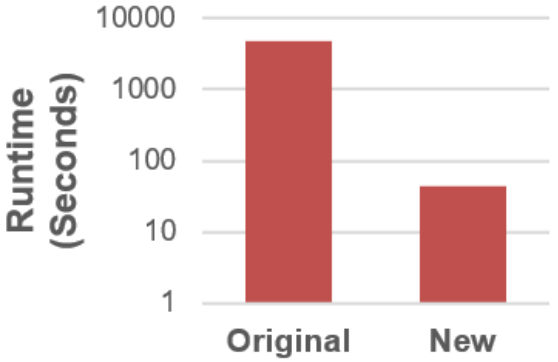


Fast PIM Simulation



PIM systems can have *billions of processing elements*

Modified Timeloop for very-large arrays



100-1000x Speedup

Exploiting Sparsity

Sparse data can be compressed

Can save space and energy

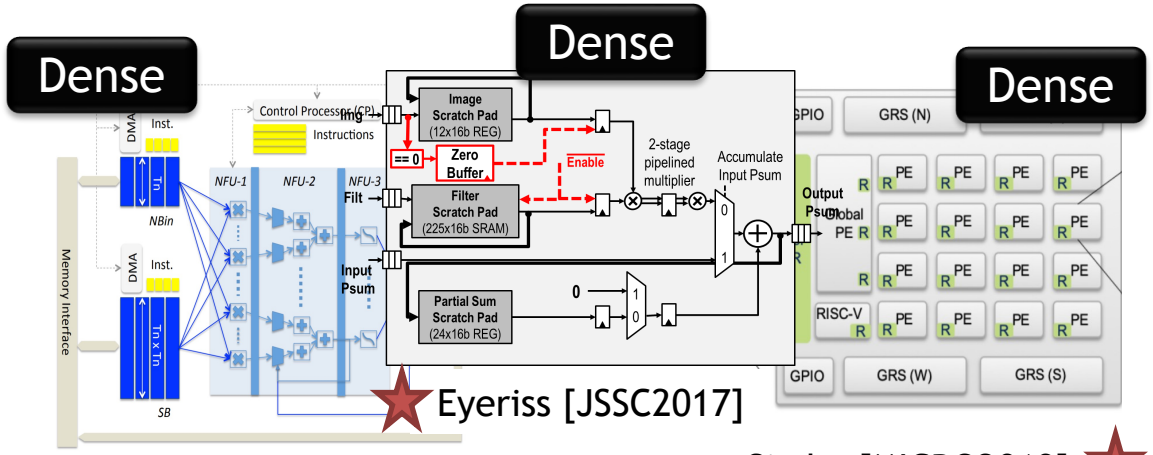
$$anything \times 0 = 0$$

$$anything + 0 = anything$$

Can save time and energy

Design Space + Digital Sparse

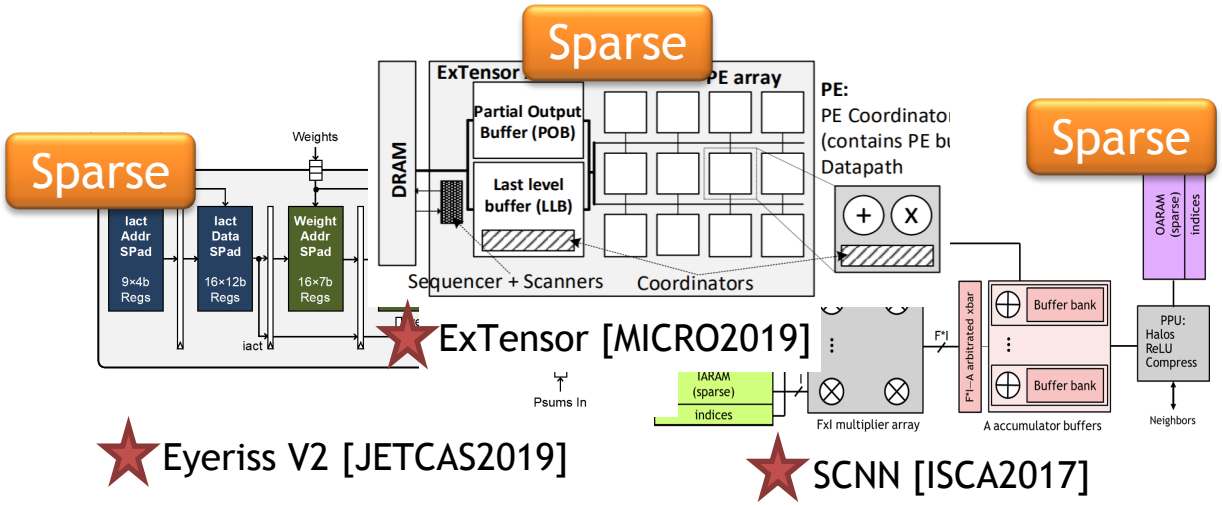
Digital-Compute Accelerator Designs



Eyeriss [JSSC2017]

Simba [MICRO2019]

DianNao [ASPLOS2014]

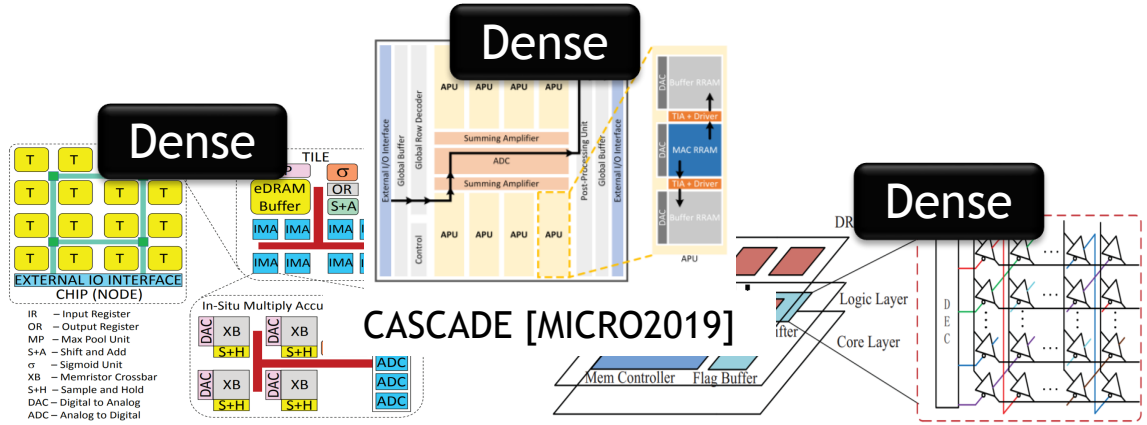


ExTensor [MICRO2019]

Eyeriss V2 [JETCAS2019]

SCNN [ISCA2017]

Analog-Compute Accelerator Designs



ISAAC [ISCA2016]

CASCADE [MICRO2019]

DrAcc [DAC2018]

★ *Works we contributed to*

Classification of Important Design Aspects

Dataflow

Processing schedule for uncompressed data movement and compute in time and space

Sparse Accelerations

Impact of sparse acceleration features on workloads with sparse data

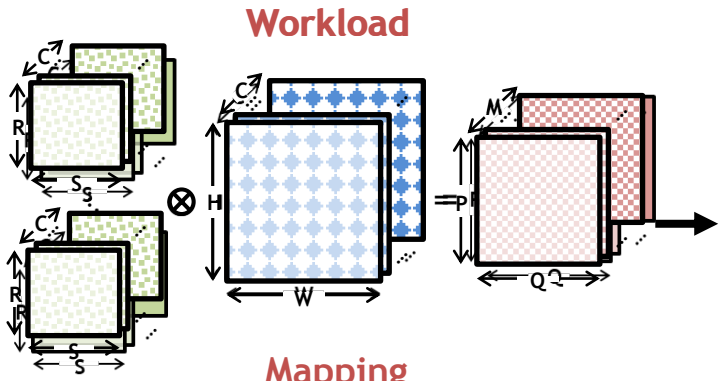
Architecture

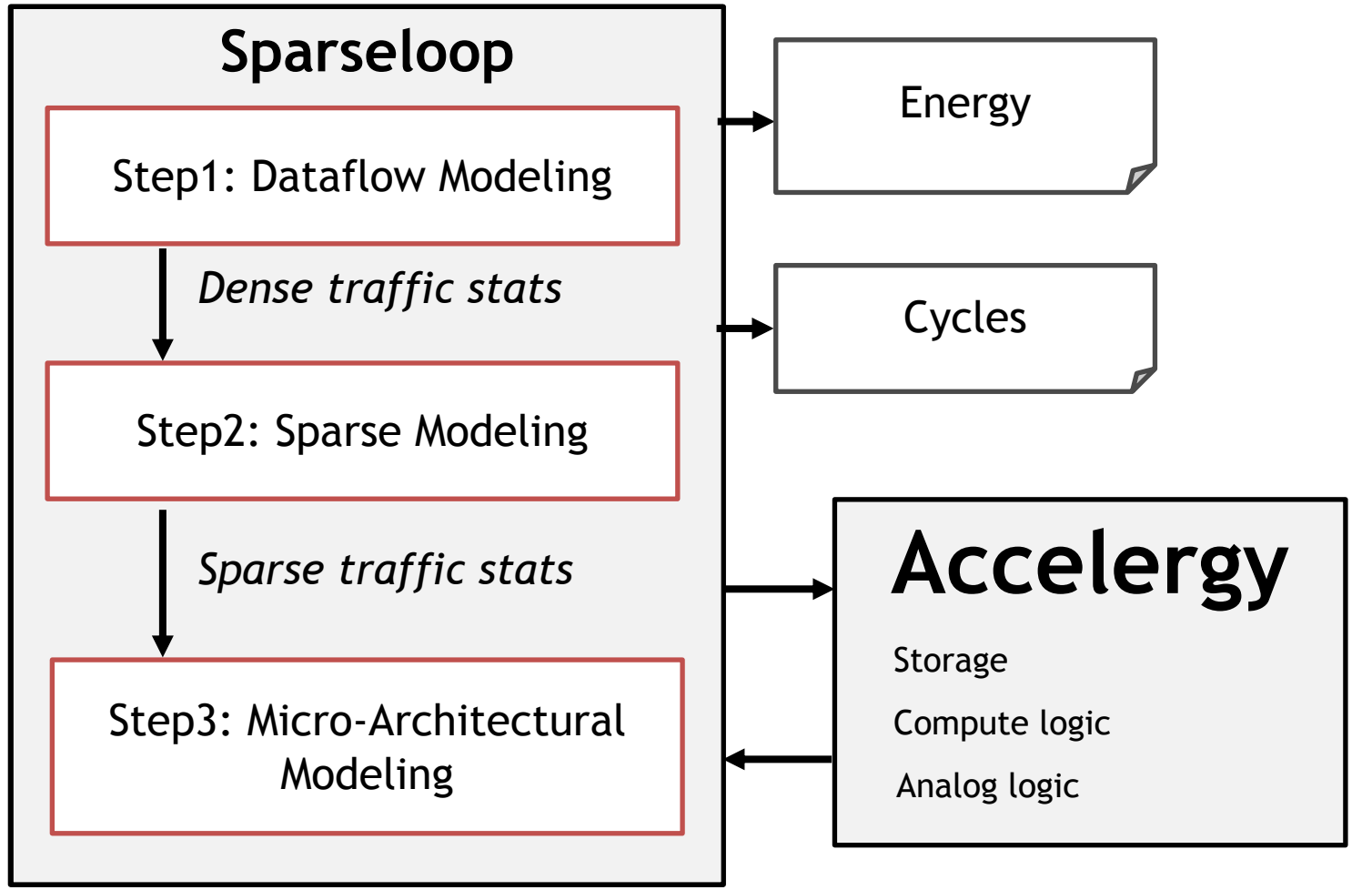
Detailed characterization of hardware, *e.g.*, storage sizes and access and compute widths

Technology

Assessment of costs at device level, *e.g.*, *area*, *energy*, *latency*

Sparseloop: Analytical Modeling for Sparse Tensor Accelerators

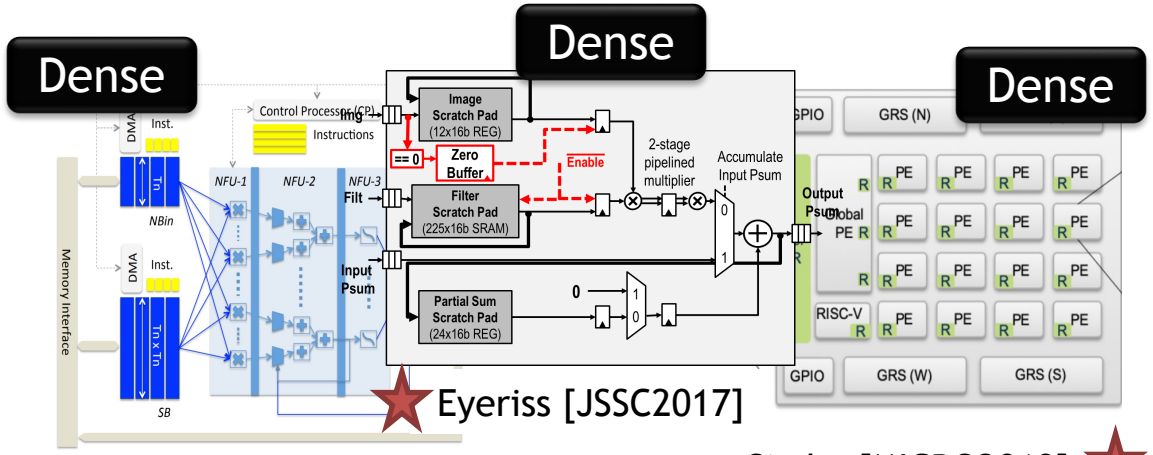




Available at sparseloop.mit.edu (MICRO 2022 Distinguished Artifact Award: **Nellie Wu**)

Design Space + Analog/Sparse

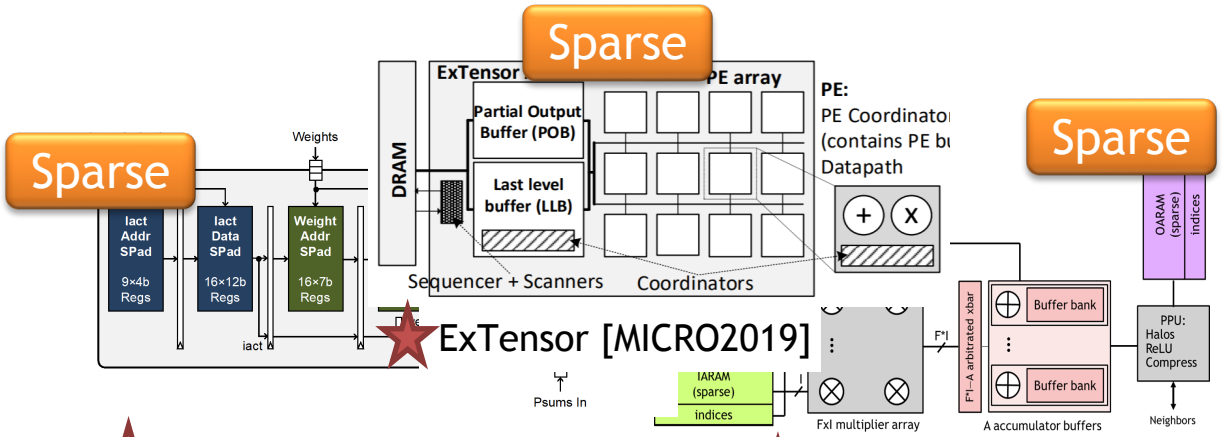
Digital-Compute Accelerator Designs



Eyeriss [JSSC2017]

DianNao [ASPLOS2014]

Simba [MICRO2019]

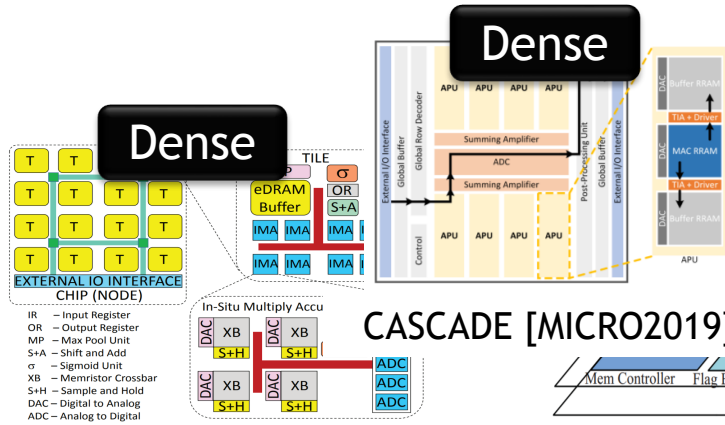


Eyeriss V2 [JETCAS2019]

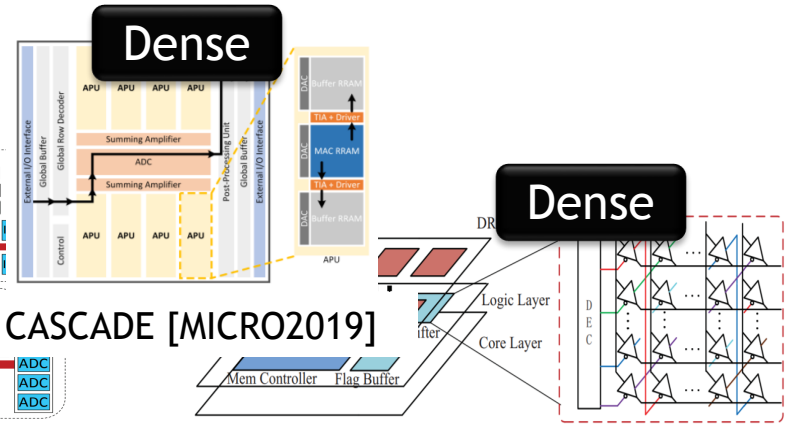
ExTensor [MICRO2019]

SCNN [ISCA2017]

Analog-Compute Accelerator Designs

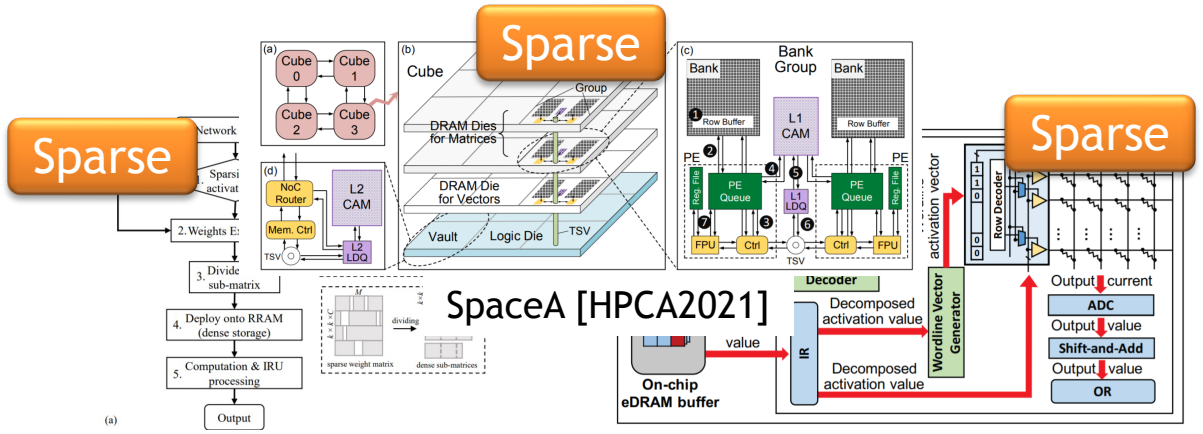


ISAAC [ISCA2016]



CASCADE [MICRO2019]

DrAcc [DAC2018]



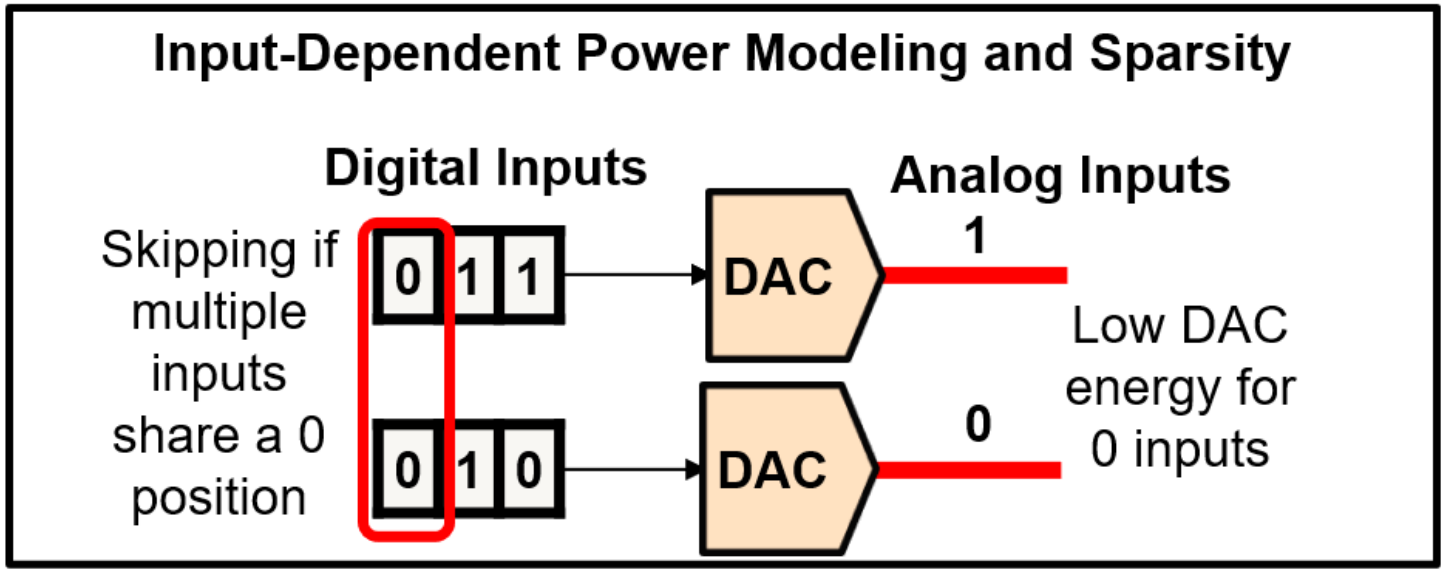
SpaceA [HPCA2021]

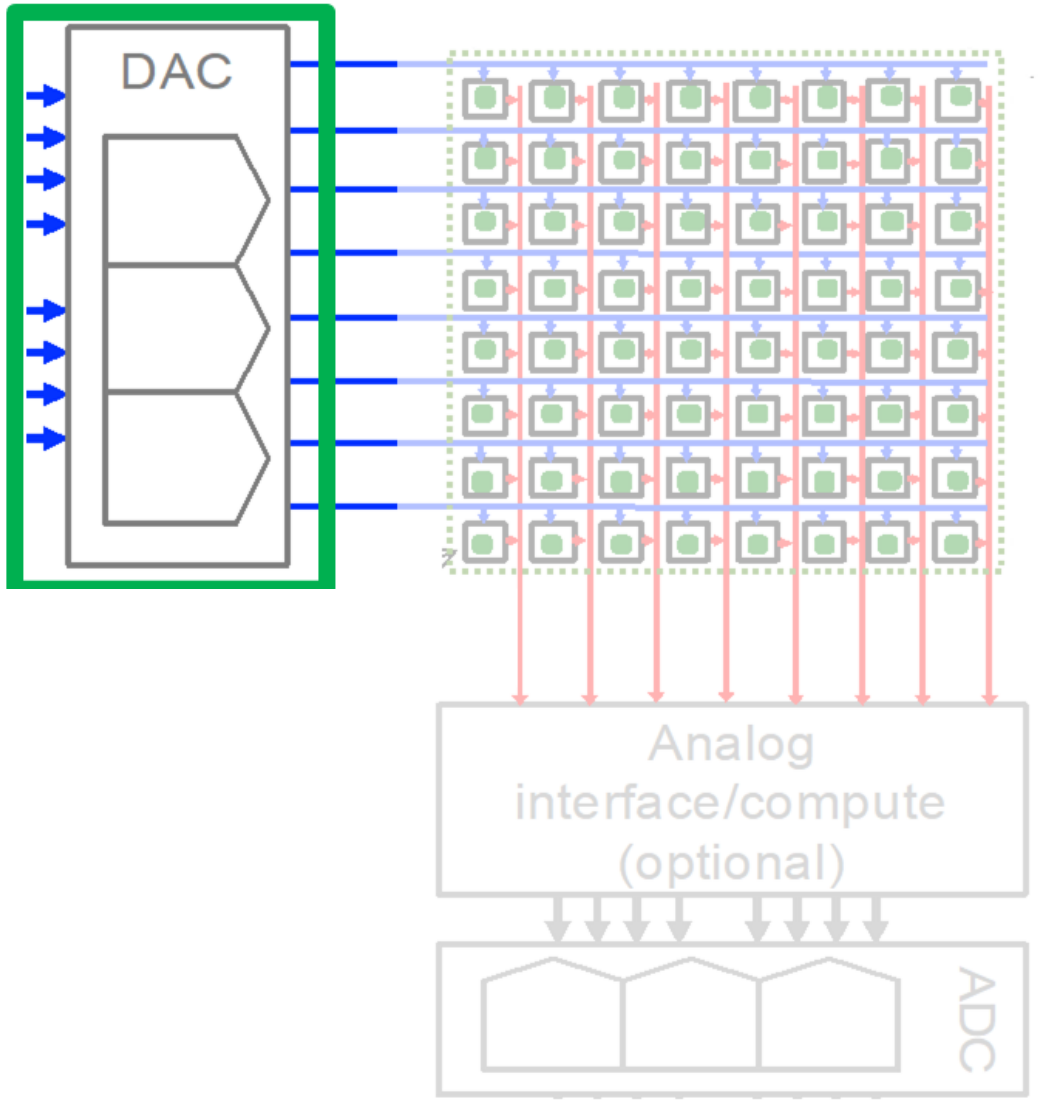
SNrram [DAC2018]

Sparse-ReRAM [ISCA2019]

★ *Works we contributed to*

Timeloop + Memristive Crossbars





Conclusion

- Timeloop+Accelergy is a model:
  - that generates performance and energy projections…
  - allowing analysis of design-space tradeoffs…
  - for a wide range of architectures, including…
  - for designs with digital and analog components…
  - as well as for designs that exploit sparsity…
  - that is fast enough to search for optimal mappings…
  - and should facilitate design-space searches…

Questions