OSCAR@ISCA 2026

Early Stage SoC Architectural Concept Definition:

Open-Source Toolset from the EPOCHS Project

June 28, 2026

Pradip Bose

IBM T. J. Watson Research Center

{pbose}@us.ibm.com

Acknowledgement

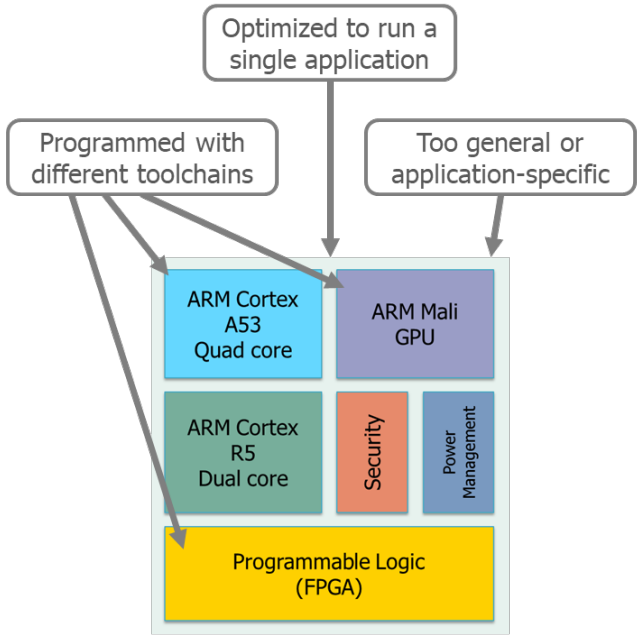
- Thanks to the many IBM colleagues who contribute to and support different aspects of this work + our esteemed university collaborators at **Harvard**, **Columbia**, and **UIUC** (Profs. David Brooks, Vijay Janapa Reddi, Gu-Yeon Wei, Luca Carloni, Ken Shepard, Sarita Adve, Vikram Adve, Sasa Misailovic) + many brilliant graduate students and postdocs!

- Special thanks to **Dr. Thomas Kazior** and **Dr. Thomas Rondeau**, final and first Program Managers of the DARPA MTO DSSoC Program, respectively.
- Also, thanks to John Wohlbier et al. at CMU-SEI, Jesse Jimenez, Gena Osborn et al. at C5ISR, John Marsh, Chris Earl et al. at DARPA for their constructive feedback and support.

This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA). The views, opinions and/or findings expressed are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

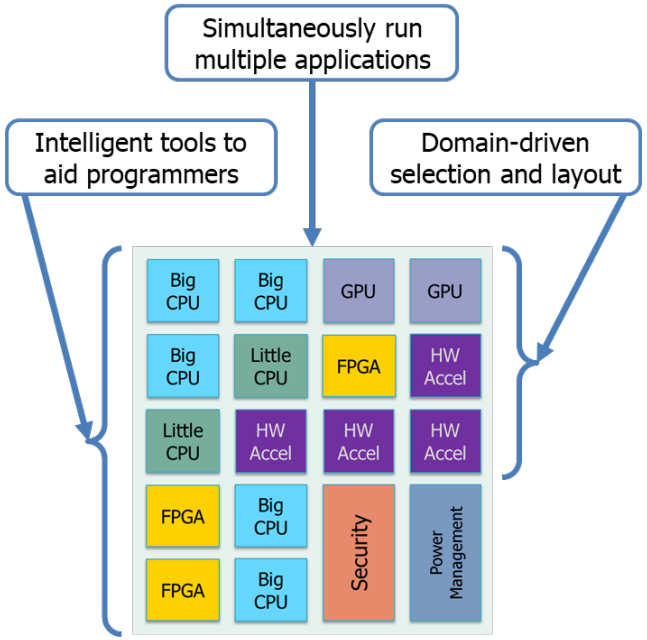
The DSSoC (Domain-Specific SoC) Vision

Status Quo (2018)



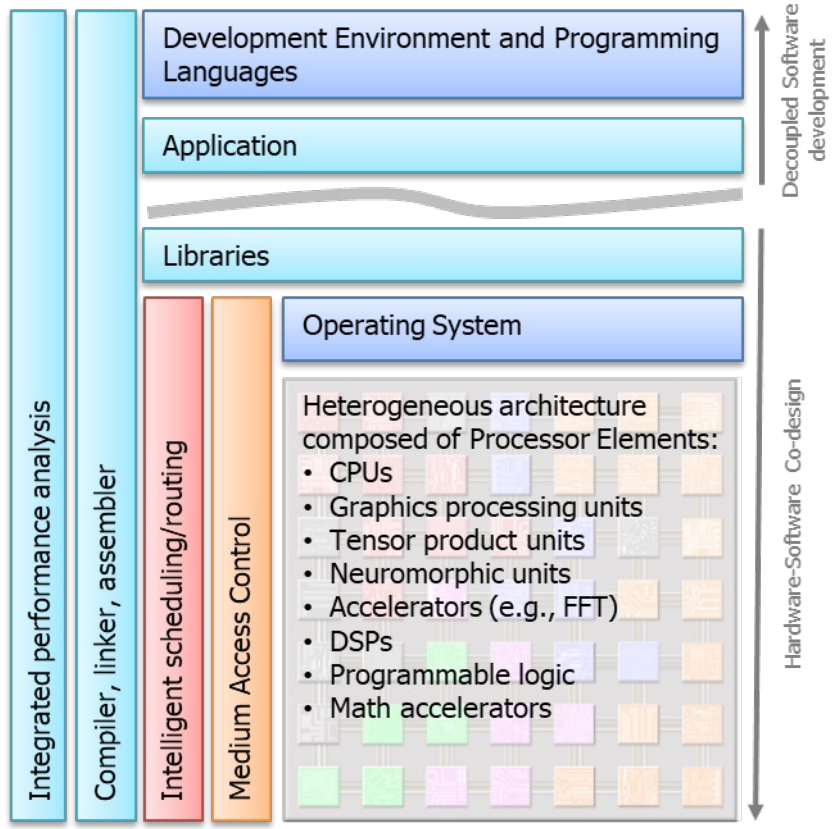
We cannot fully utilize our current multi-processor systems

DSSoC End Goal (2023)



We want to manage an increase of capabilities with integrated specialized accelerators

DSSoC's Full-Stack Integration

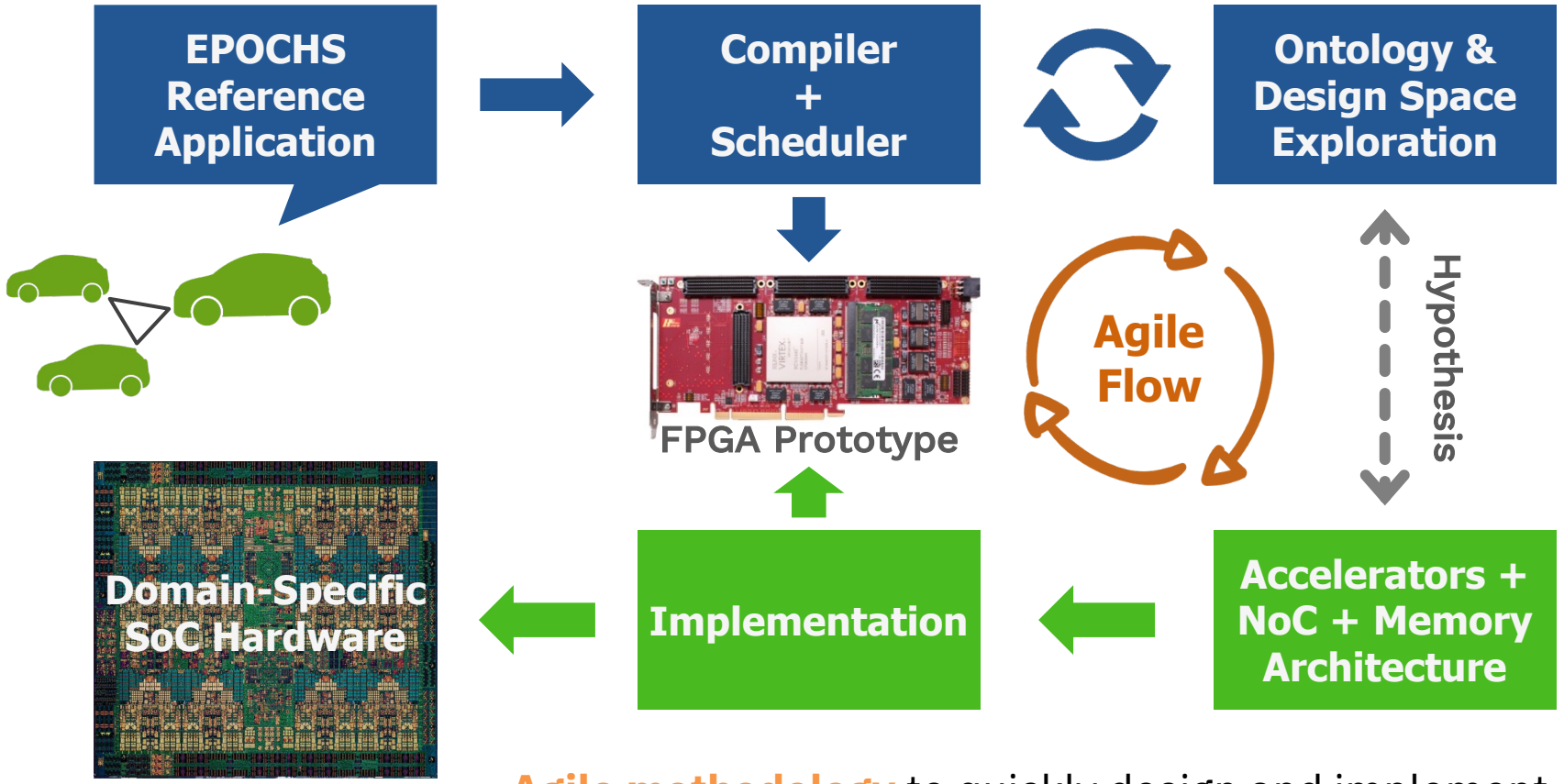


- Software-Hardware Co-Design [Efficiency]
- Agile SoC Design [Design Productivity]
- Easily Programmable [User Friendly]

→ The three driving principles

Major slide content: courtesy Tom Rondeau (DARPA Prog. Mgr)

EPOCHS Agile Flow Methodology



Agile methodology to quickly design and implement an easily programmed domain-specific SoC for real-time cognitive decision engines in connected vehicles

Putting It All Together: EPOCHS Appliance

Programmable! Programmable! Programmable!

Annotated ERA

HPVM compiler with interface to Scheduler Library

Scheduler Library and Task Library with flavors of real-time scheduling policies

Hardware libraries for ESP tiles and RISCV cross compiler for the RISCV Ariane core

EPOCHS Reference Application (ERA)

HPVM Compiler

Scheduler Library API

Scheduling Policy

Scheduler Library

Task Library

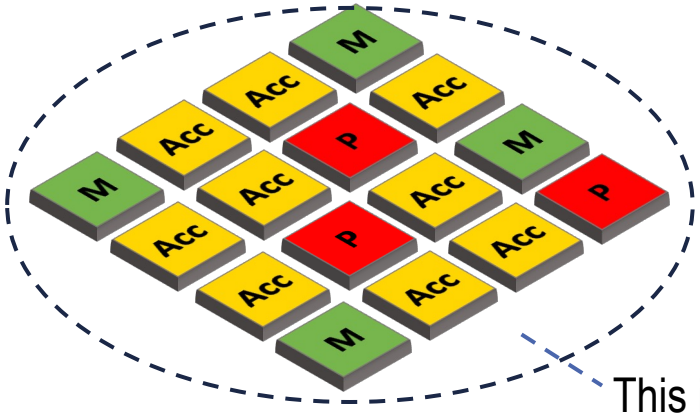
Hardware API

hetero-integration

hpvm-release-epochs

meta

master



SoC-agnostic agile system programmability

(open-source tools)

"Smart" Scheduler

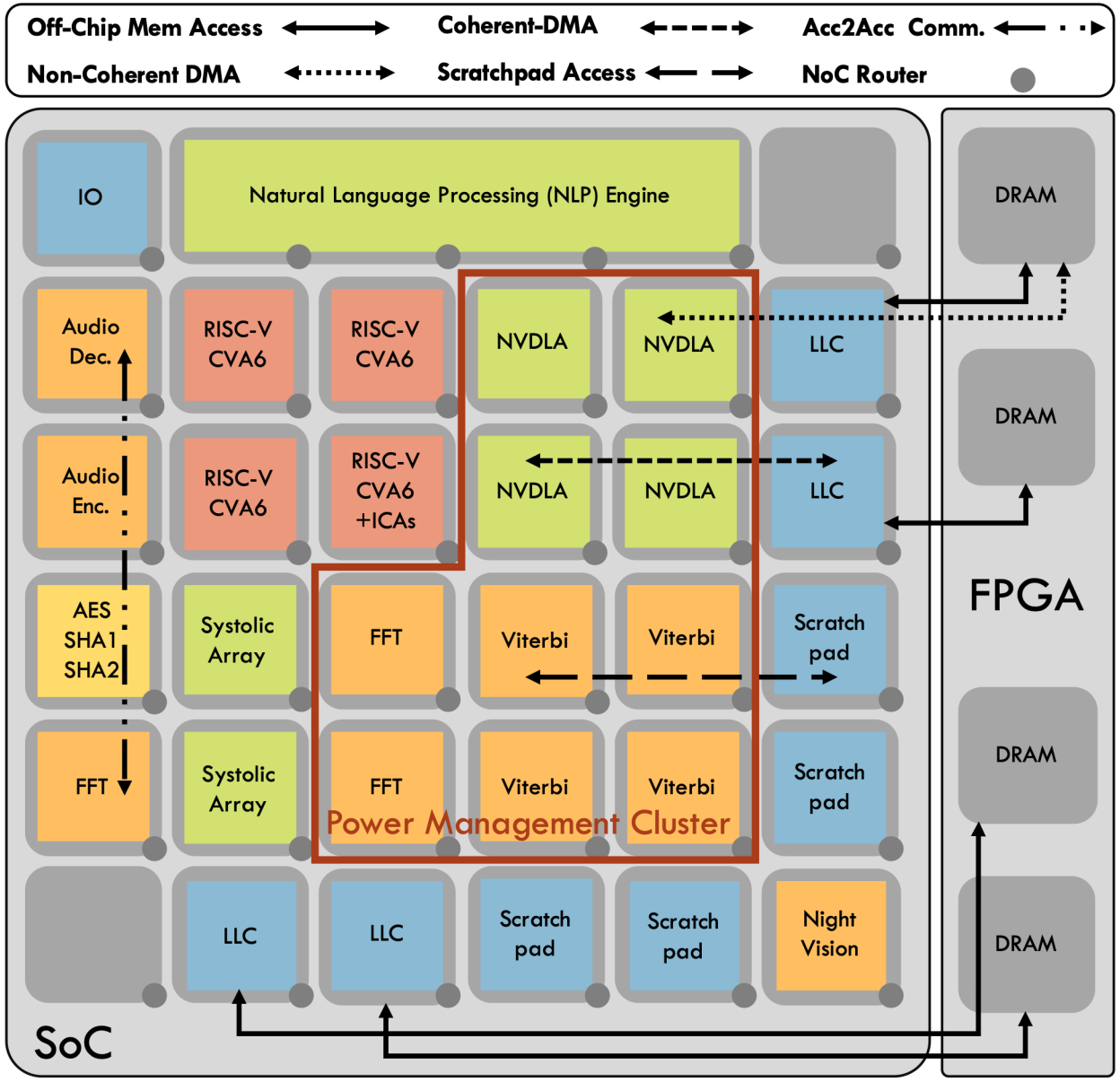
github.com/IBM/scheduler-library

This is a cartoon to help highlight the s/w stack !

The next slide shows the real hardware SoC

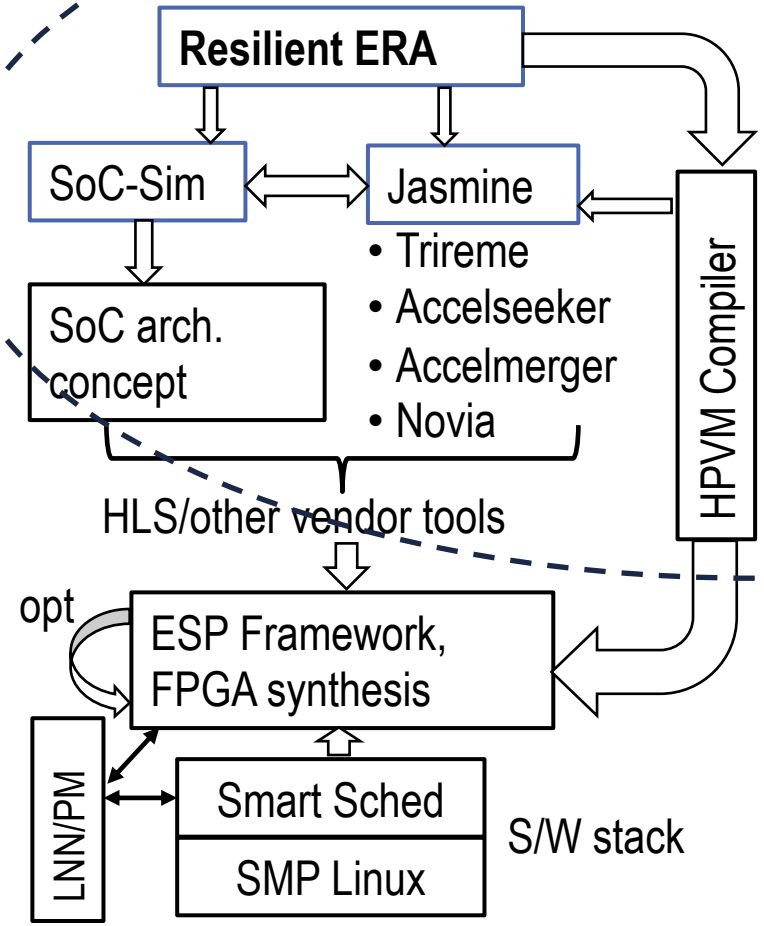
EPOCHS-1 Chip Highlights

- 64 mm² SoC designed in 12 nm FinFET
- 35 clock domains; 23 power domains
- 8.4 MB on-chip SRAM memory
- Tile-based SoC architecture
- 34 tiles connected by a 6-plane 2-D mesh NoC
- The 74 Tbps NoC provides flexible orchestration of data
- 23 accelerators of 14 different types
- 10 accelerators compose a cluster demonstrating a **novel distributed hardware power management scheme**
- Designed by a small team of PhD students, postdocs, and industry researchers in 3 months with ESP, an open-source platform for agile SoC design



6x6 tiled SoC, 2D mesh NoC, using Columbia's open-source ESP as the foundational design enabler: esp.cs.columbia.edu

Documented Toolchain (Open-Source)



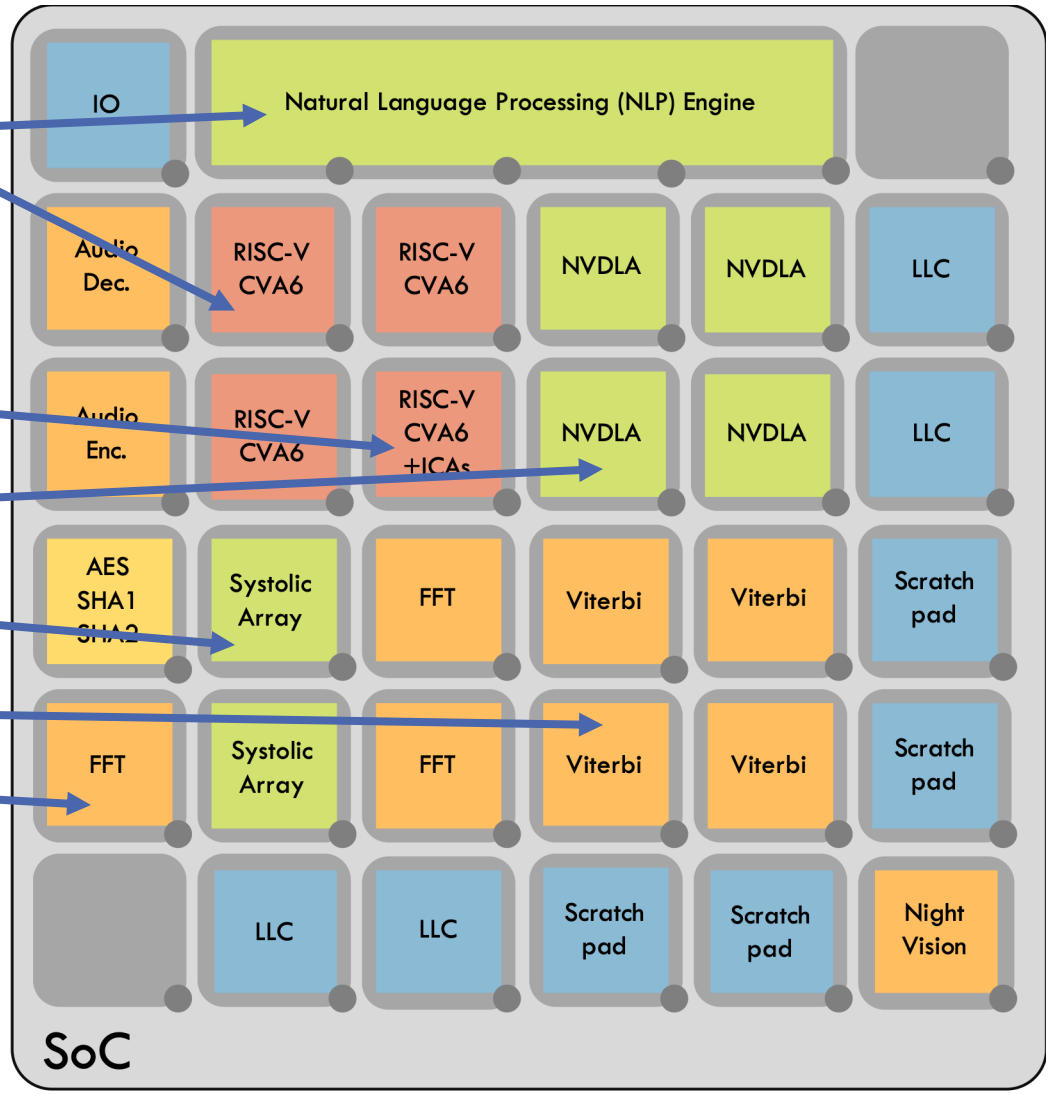
- SoC-Sim (ARTEMIS) development, built around the *FARSI* tool from Harvard and STOMP simulator from IBM
- **Jasmine ontology toolset** – open-source, documented
  - ✓ *Trireme*
  - ✓ *Acclseeker*
  - ✓ *Accelmerger*
  - ✓ *NOVIA*
- Software stack
  - ✓ *Scheduler library (SL)*
  - ✓ *HPVM compiler integration*
  - ✓ *SMP Linux bring-up*
  - ✓ *ESP from Columbia Univ.* (Prof. Luca Carloni's group).

Focus, in this talk, on this pre-RTL SoC architecture definition process

This is the aspect of design that we don't hear about much. Yet, this is where computer architects must deliver so that there are no post-silicon shortfalls in power-performance or other metrics!

Questions that may arise …..

- Why are there **four** RISC-V (CVA-6) Ariane processor cores?
- And, why this 4-tile spanning NLP engine?
- Why is there **one** core with inline compute accelerators **(ICAs)?**
- **Four** NVDLA engines, **plus two** systolic array PEs ?
- **Four** Viterbi and and **three** FFT accelerators?
- And, so on, including questions about the scratch pad and LLC blocks

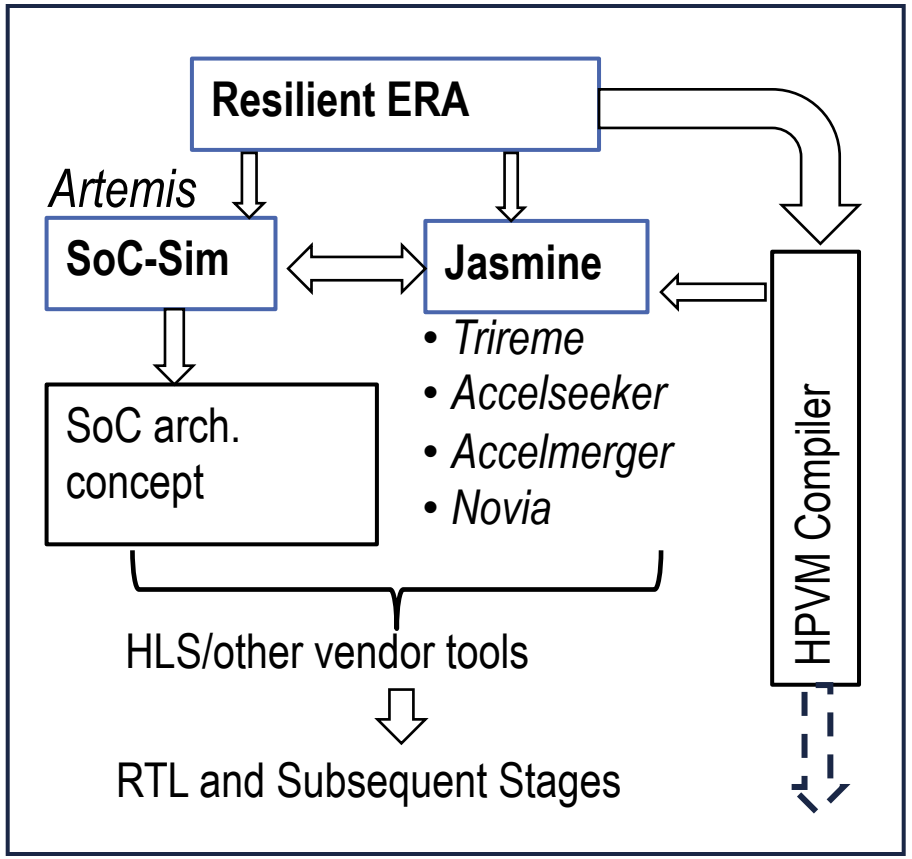


Honest answer: our concept-phase pre-RTL microarchitecture definition toolset was not ready in time! Our choices were *ad hoc!*

But that would not an acceptable answer for a real product chip!

- Every piece of the silicon real-estate must be justified in light of target workloads and metrics!
- So, if one has to design and deploy an edge-AI heterogeneous SoC with tight SWaP-C and real-time performance requirements, one must invest into the pre-RTL SoC microarchitecture definition toolset in advance and use it to ensure a competitive, performant product.
- Therefore, as we attempt to design ***future*** domain-specific SoCs (and SiPs), it is worth examining what we were able to develop in terms of the pre-RTL modeling & microarchitecture optimization tools.

Early-Stage SoC Architecture Definition



Two Key Challenges

a) How does one systematically "discover" the "right" accelerator types and counts?

The Jasmine toolset; the tools listed are:

- ✓ *Trireme*
- ✓ *Acclseeker*
- ✓ *Accelmerger*
- ✓ *NOVIA*

And, once the desired h/w primitives are known,

b) How does one iteratively "optimize" the placement of CPUs and accelerator PE tiles in order to maximize a specified performance or power-performance metric?

SoC-Sim (aka Artemis)

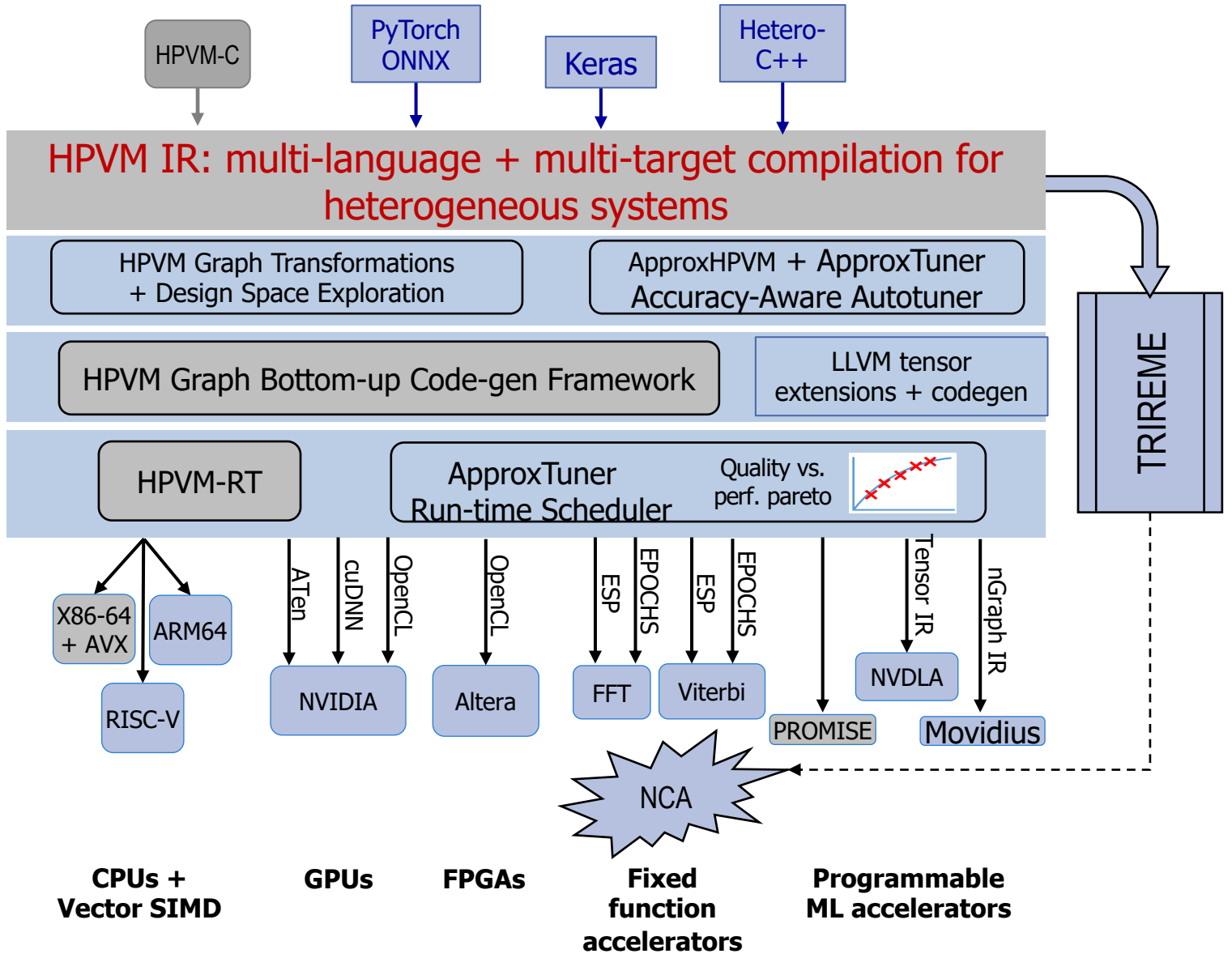
Heterogeneous Parallel Virtual Machine (HPVM) – Compiler Framework

Hashim Sharif, Yifan Zhao, Adel Ejjeh, Akash Kothari, A. Rafae Noor, Leon Medvinsky, Vikram Adve, Sasa Misailovic, Sarita Adve (UIUC)

Goal: Programmability for Heterogeneous Parallel Systems

HPVM enables:

1. Multiple parallel languages, DSLs
2. Retargetable parallel compiler IR, optimizations, codegen
3. Large perf + energy gains for edge ML workloads
4. Automated hardware DSE



Papers

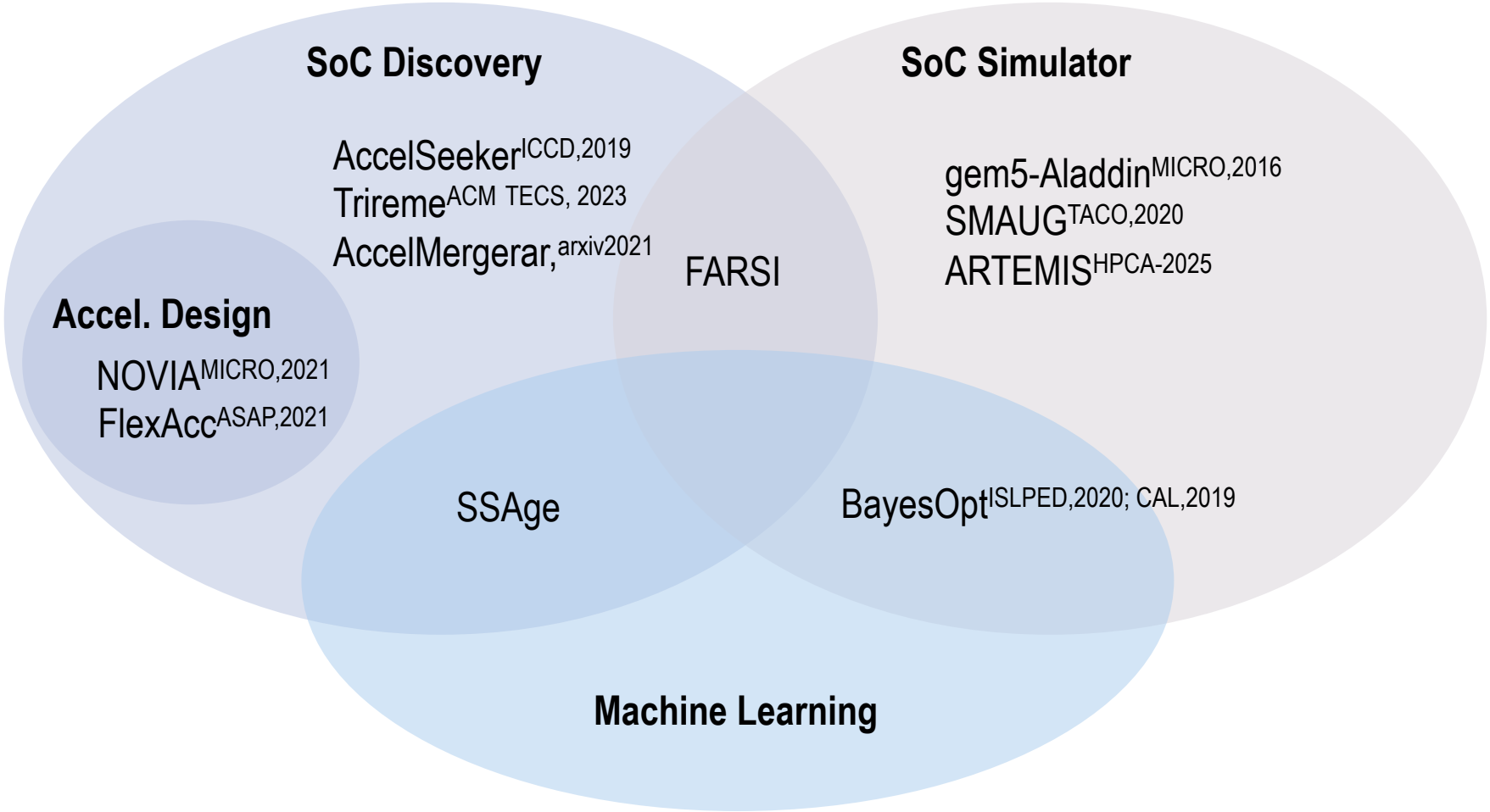
Kotsifakou+, PPOPP 2018
Srivastava+, ISCA 2018
Kang+, MICRO 2019
Sharif+, OOPSLA 2019
Sharif+, PPOPP 2021
Sharif+, *in review, 2021*
Zacharapolous+, *in review. 2021*

Software

Jan. 2020: HPVM 0.5
April 2021: HPVM 1.0

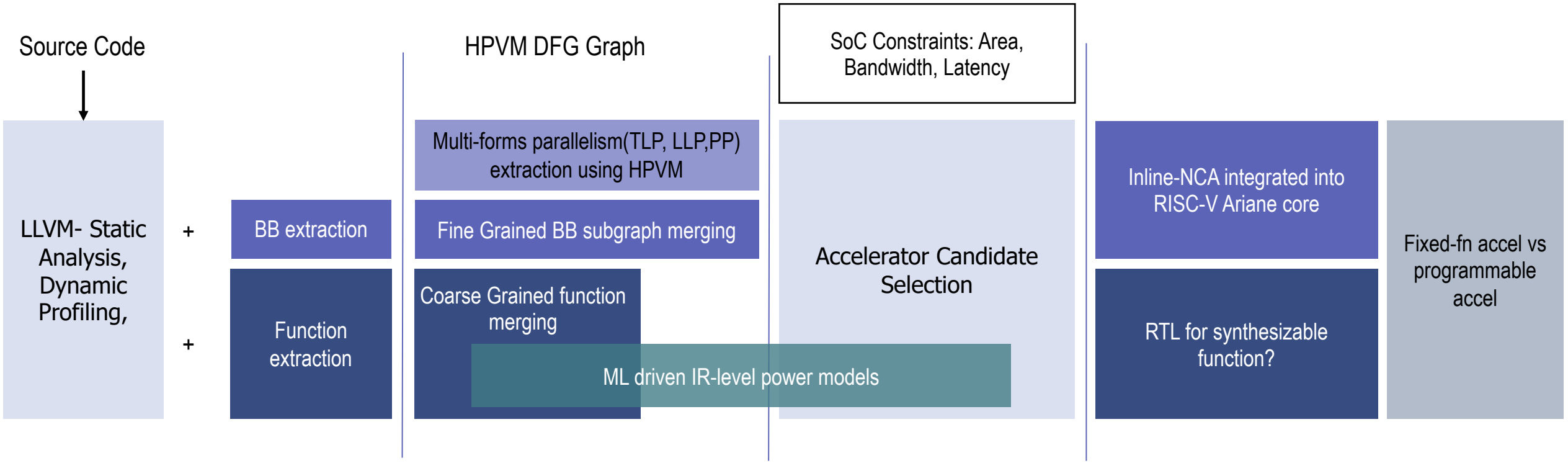
https://publish.illinois.edu/hpvm-project/

Landscape of Ontology Tools: *Jasmine*



We actually attempted a whole lot more than
what finally proved to be appropriate for the problem at hand ☺!

Jasmine Toolflow: *SoC Accelerator Discovery*



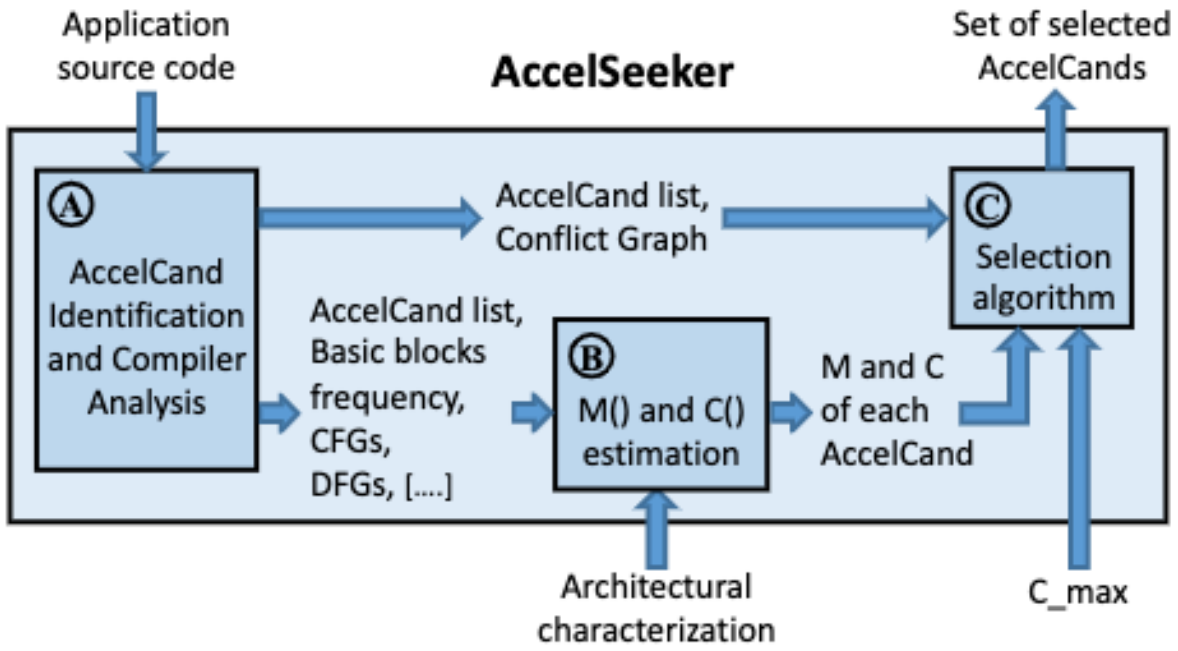
Ontology Tools

AccelSeeker | Trireme | Accelmerger | NOVIA (NCA) | SSAge

- **AccelSeeker**[ICCD,2019]
  - Identify basic-block level parallelism for HW acceleration
- **Trireme**[ACM TECS, 2023]
  - Leverages HPVM data flow graph to identify multiple forms of parallelism for further hardware acceleration
- **NOVIA**[MICRO,2021]
  - Identify and extract NCA (inline fine-grained accelerators integrated to core) for less area overhead and small startup time
- **AccelMerger**[arxiv,2021]
  - Coarse grained, loosely coupled accelerator
  - Identify opportunities to merge kernels for resource-efficient acceleration
- **SSAge**
  - ML driven IR-level power models for merging accelerators

The tools that seemed to make a difference

AccelSeeker* identifies HW acceleration candidates



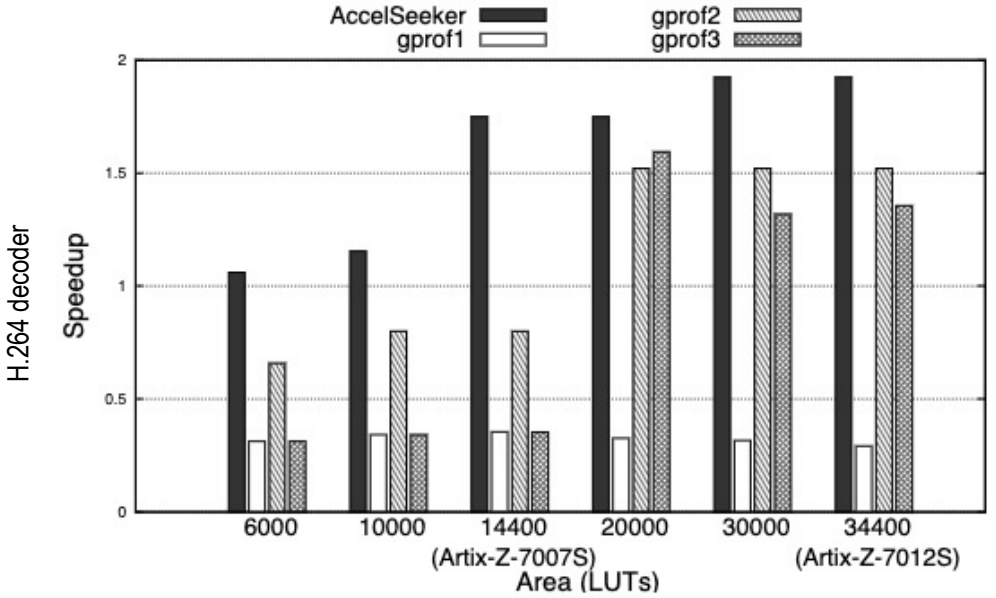
MERIT M()

SW latency
HW Computation Latency
HW Communication Latency
Runtime execution Frequency

COST C()

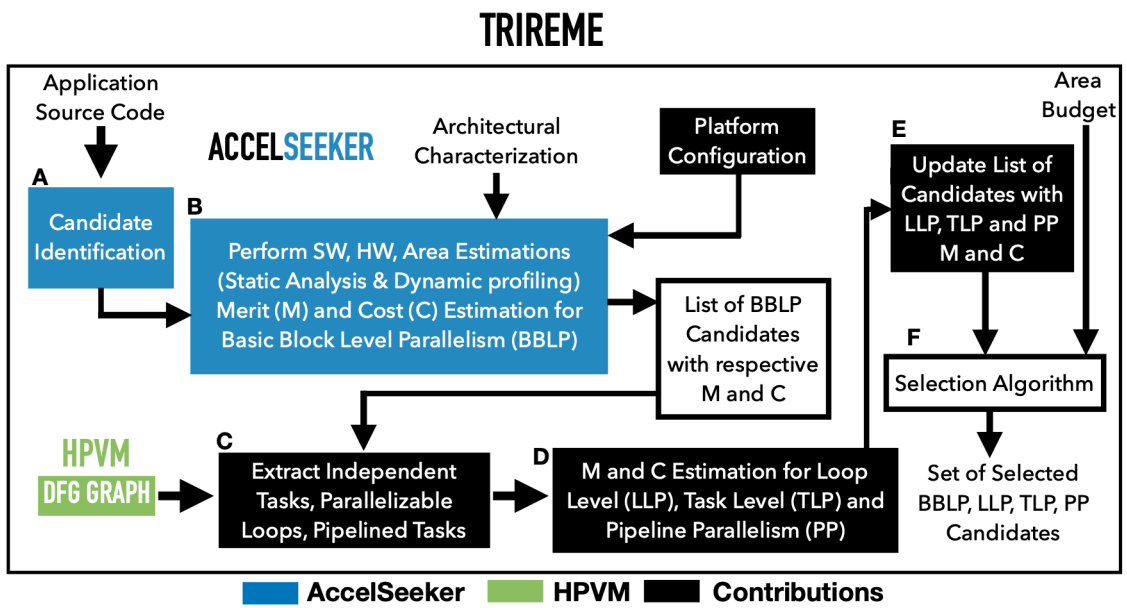
HW Area
Resources: LUTs, DSPs, BRAMs

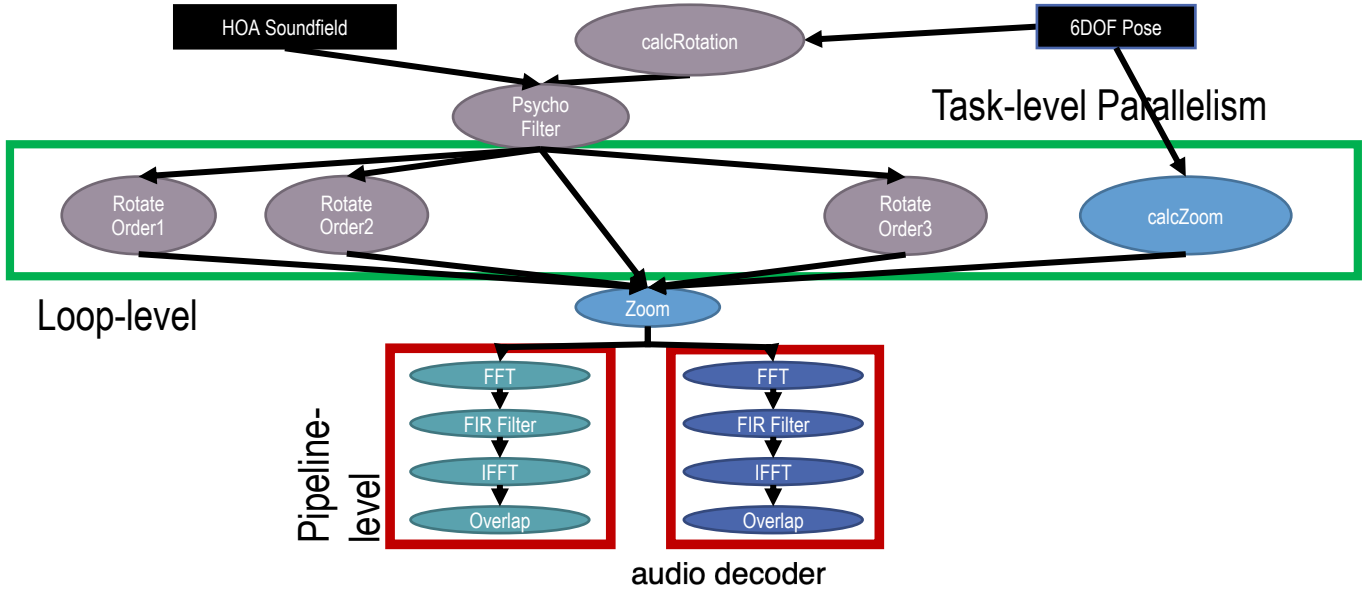
- Automated HW/SW codesign
- What parts of Application should be Accelerated?
- Identify basic-block level parallelism for HW acceleration.

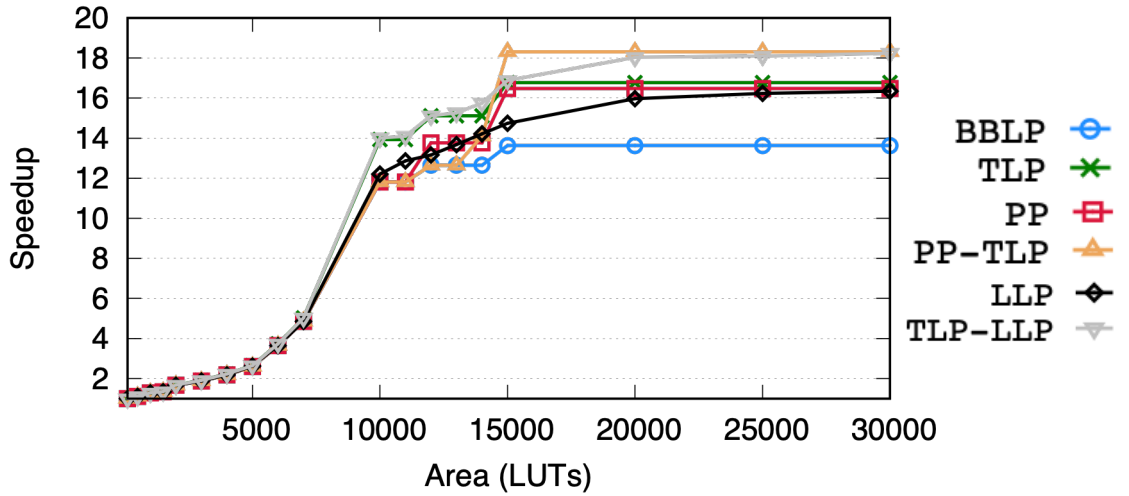


*** Zacharopoulos, Georgios, et al. "Compiler-assisted selection of hardware acceleration candidates from application source code." IEEE International Conference on Computer Design (ICCD), 2019.**

Trireme*: Automatic Identification of Accelerators using HPVM





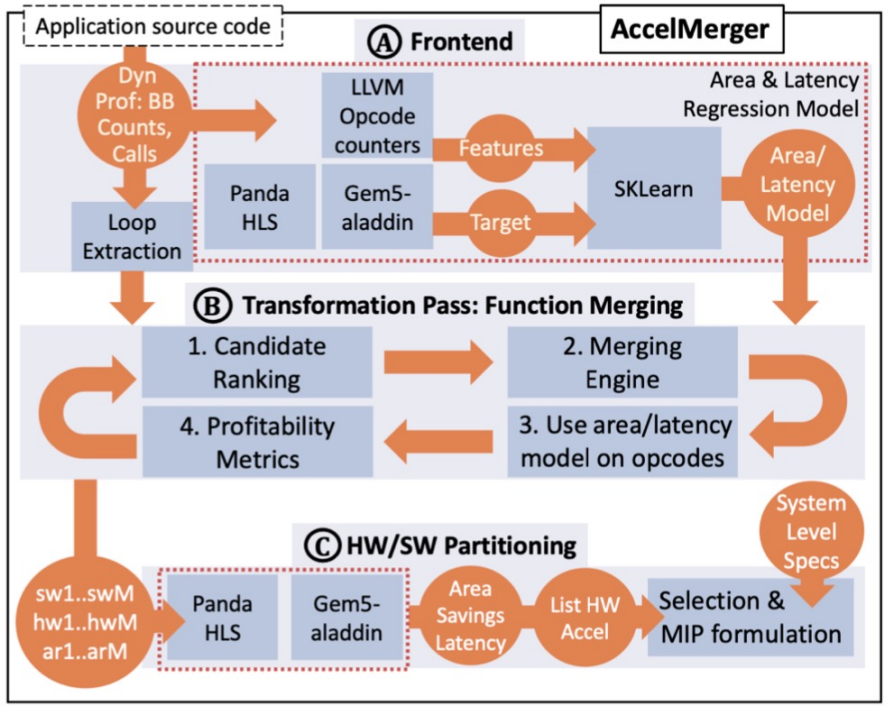


- Leverages HPVM to identify different types of parallelism
  - Task Parallelism (TLP)
  - Loop Parallelism (LLP)
  - Pipeline Parallelism (PP)
- Incorporates them into Metric and Cost estimation

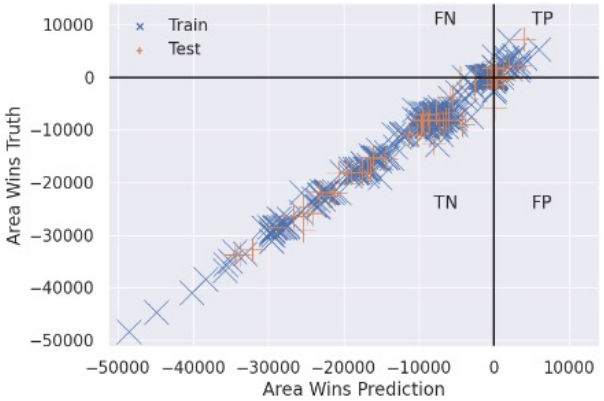
*** Zacharopoulos, Georgios, et al. Trireme: "Exploration of Hierarchical Multi-level Parallelismfor Hardware Acceleration," ACM TECS, vol. 22, issue 3, May 2023.**

AccelMerger*: Automatic Generation of Coarse Grained Merged Accelerators

Methodology



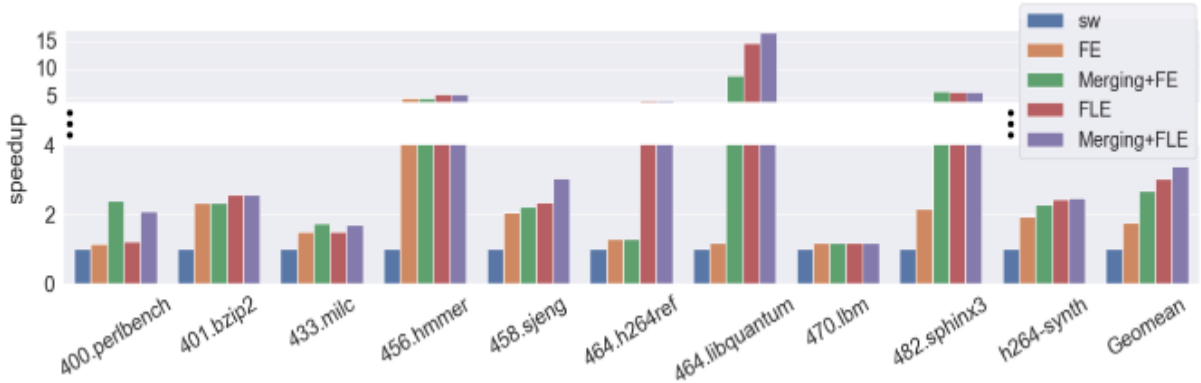

```
int f1(int a, int b, bool sum) {
    int c;
    if (sum)
        c = a + b;
    else
        c = a * b;
    c = f3(c, a);
    return c;
}

int f2(int a, int b, int d, bool mult) {
    int c;
    if (mult)
        c = a * b;
    else
        c = d + b;
    return c;
}

---------- Merged Function ----------

int f12(int a, int b, int d, bool sum,
    bool mult, bool f_sel) {
    int c;
    bool cond = f_sel : sum, not mult;
    if (cond) {
        int op = f_sel : a, d;
        c = op + b;
    } else
        c = a * b;
    if (f_sel)
        c = f3(c, a);
    return c;
}
```



- Functions are merged when the cost model estimates a profitable circuitry reduction.
- HW/SW partitioning enables speedup estimates over SW-only executions.

Identify opportunities to merge kernels for resource-efficient acceleration yielding non-conventional accelerators (NCAs), using an accurate ML resource model.

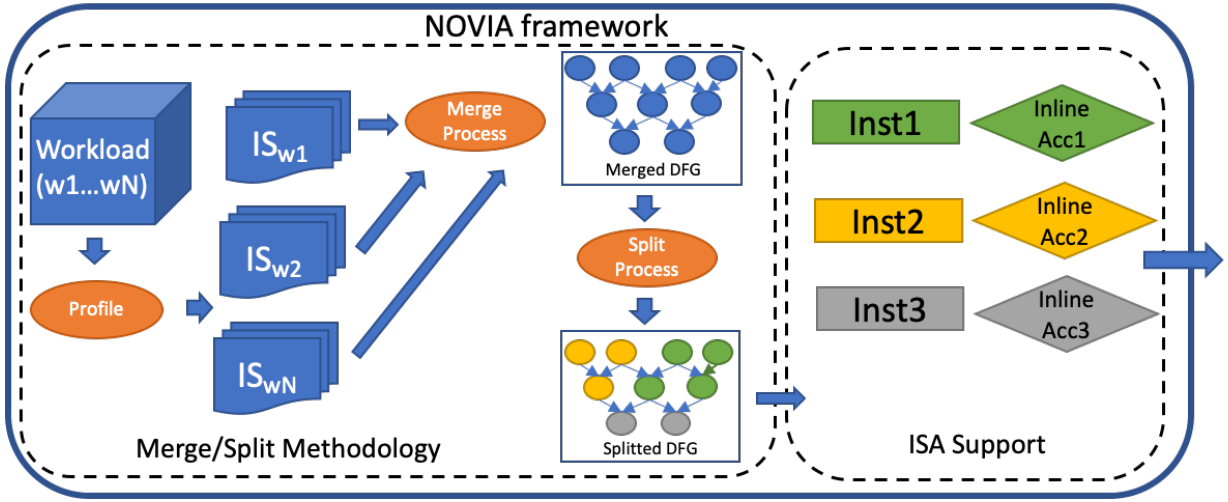
Results



* Brumar, Iulian, et al. "Early DSE and Automatic Generation of Coarse Grained Merged Accelerators." *arXiv preprint arXiv:2111.09222* (2021)

NOVIA*: A Framework for Discovering Non-Conventional Inline Accelerators

Systematic Methodology

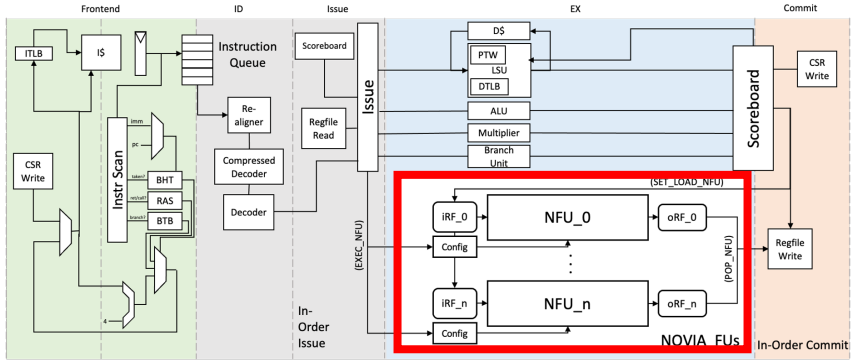


- Offload accelerators:
  - Large enough computation to compensate for invocation overheads
  - Offload accelerator start-up can restrict the functions that can be speed up
- **Inline accelerators:**
  - Faster start-up times
    - allowing them to provide speedup on smaller pieces of the workload
  - Close to core pipeline
  - Customized functional units
  - Minimal integration effort

Integration with Ariane Cores

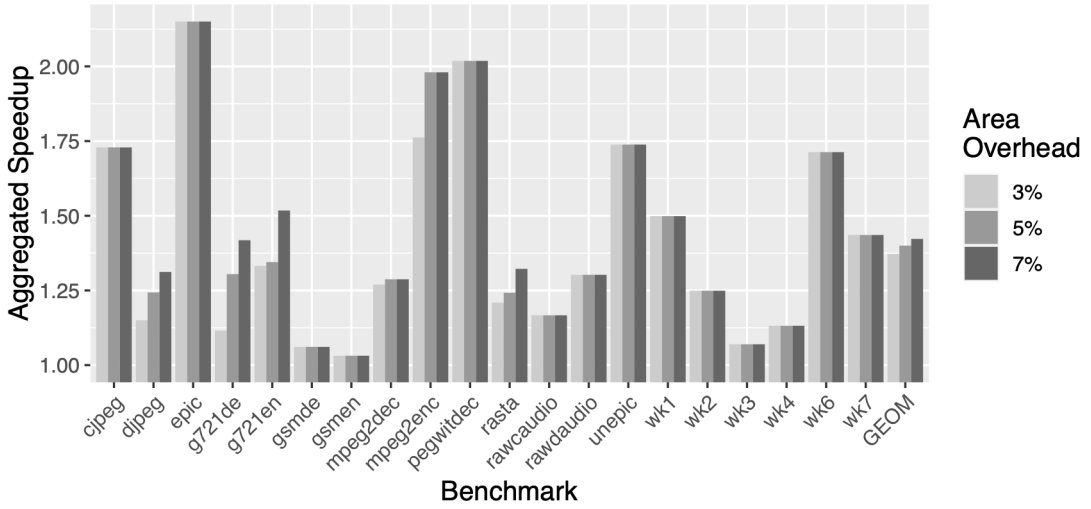
- A modular interface for any proposed inline accelerators

Ariane core pipeline



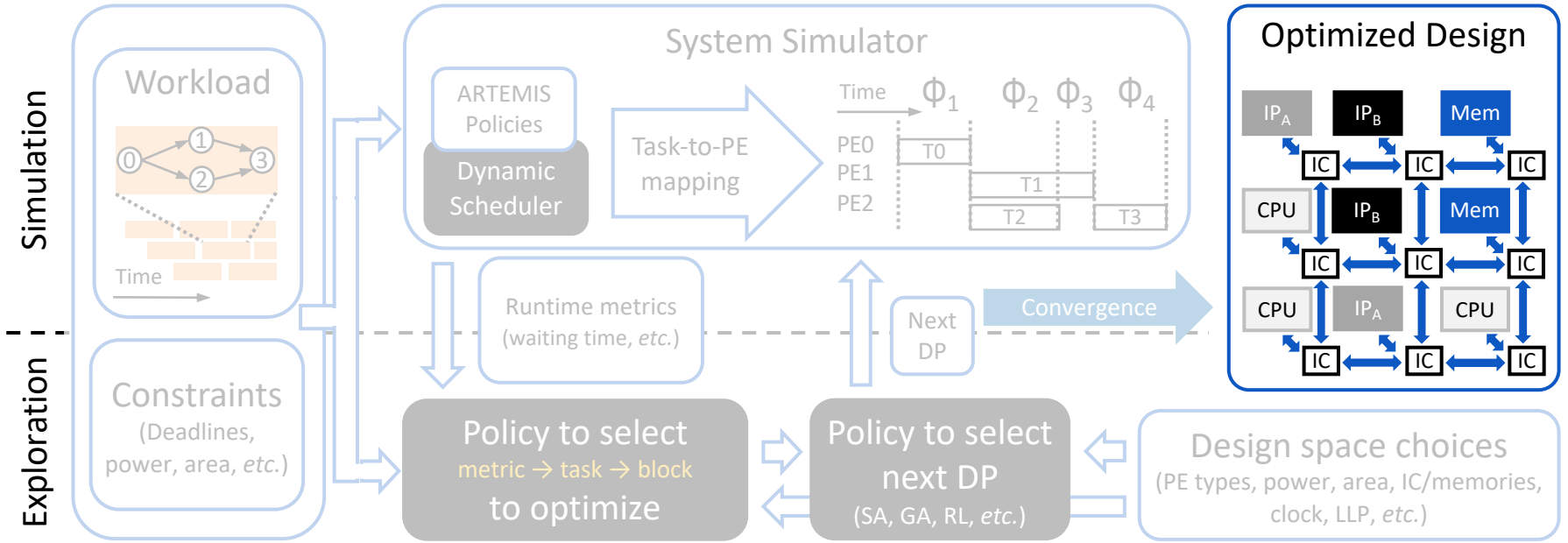
Integrated Inline accelerator unit (NFU)

Results



* Trilla, David, et al. "NOVIA: A Framework for Discovering Non-Conventional Inline Accelerators." *MICRO-54: 54th Annual IEEE/ACM International Symposium on Microarchitecture*, 2021.

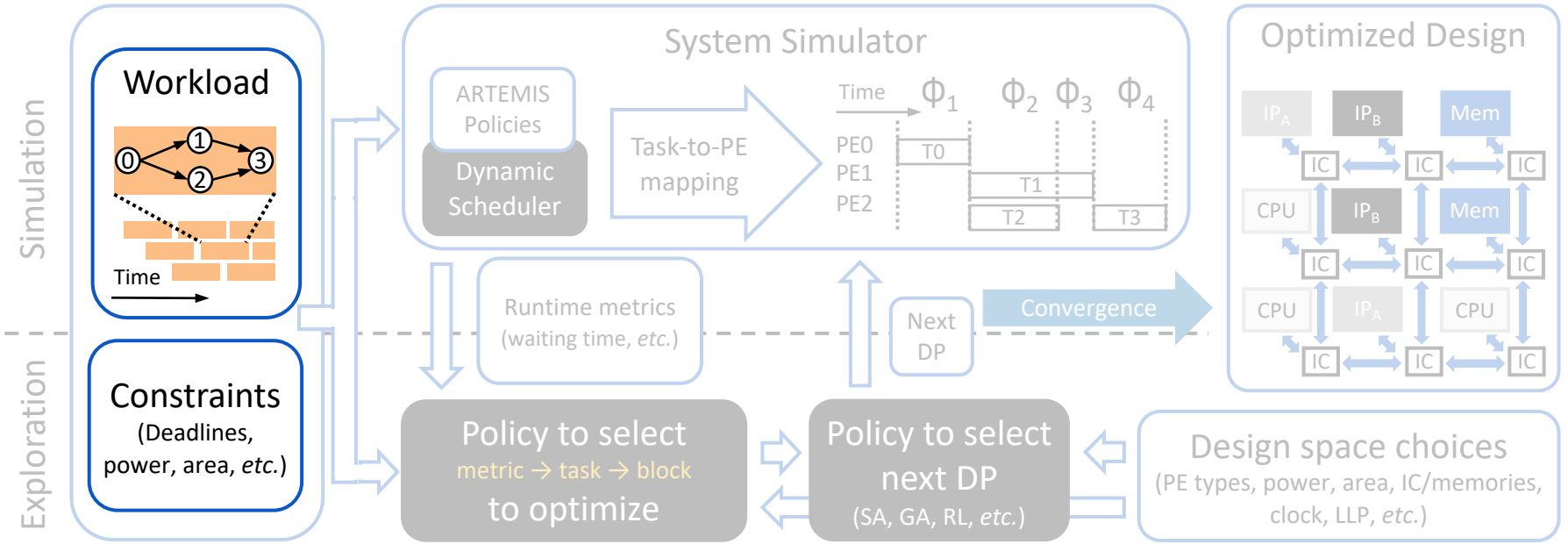
The ARTEMIS* Framework



- ARTEMIS consists of an exploration and a simulation framework
  - Begins with a seed SoC DP and incrementally transforms it to an optimized DP that meets the pre-specified constraints

***Subhankar Pal, Aporva Amarnath, Behzad Boroujerdian, Augusto Vega, Alper Buyuktosunoglu, John-David Wellman, Vijay Janapa Reddi, Pradip Bose, "ARTEMIS: Agile Discovery of Efficient Real-Time Systems-on-Chips in the Heterogeneous Era," HPCA 2025.**

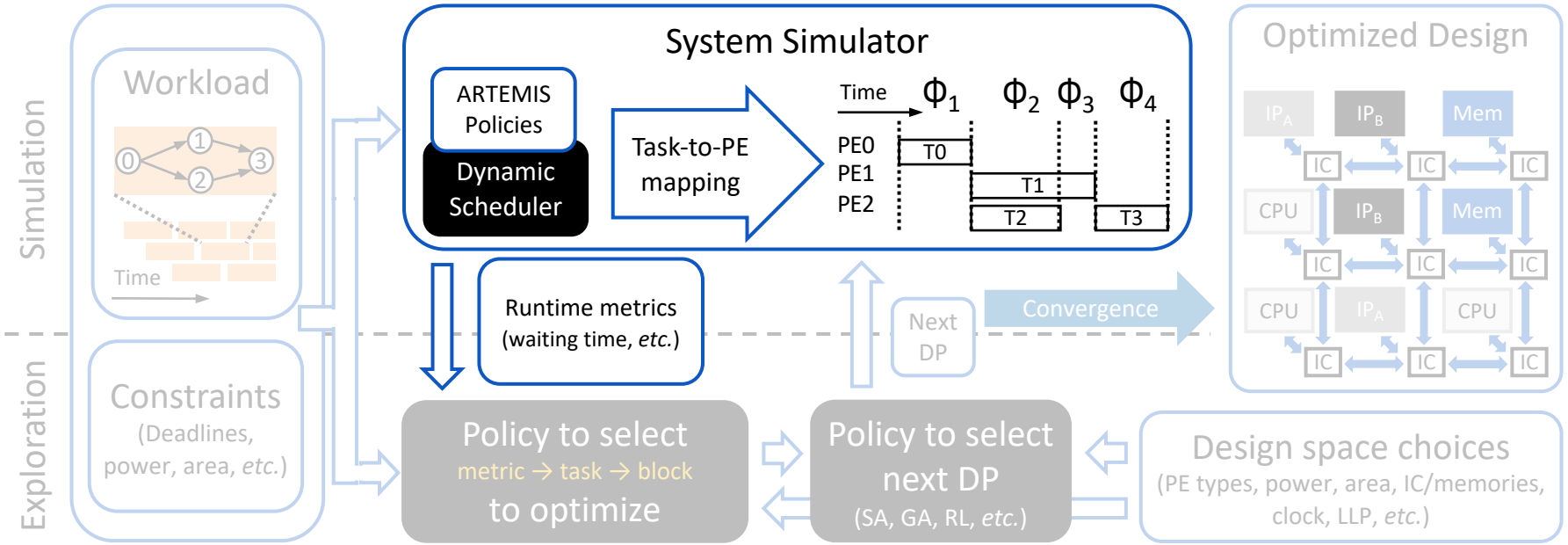
The ARTEMIS* Framework



- DAG representation of the workload and application and system constraints are sent as inputs to the framework

*Subhankar Pal, Aporva Amarnath, Behzad Boroujerdian, Augusto Vega, Alper Buyuktosunoglu, John-David Wellman, Vijay Janapa Reddi, Pradip Bose, "ARTEMIS: Agile Discovery of Efficient Real-Time Systems-on-Chips in the Heterogeneous Era," HPCA 2025.

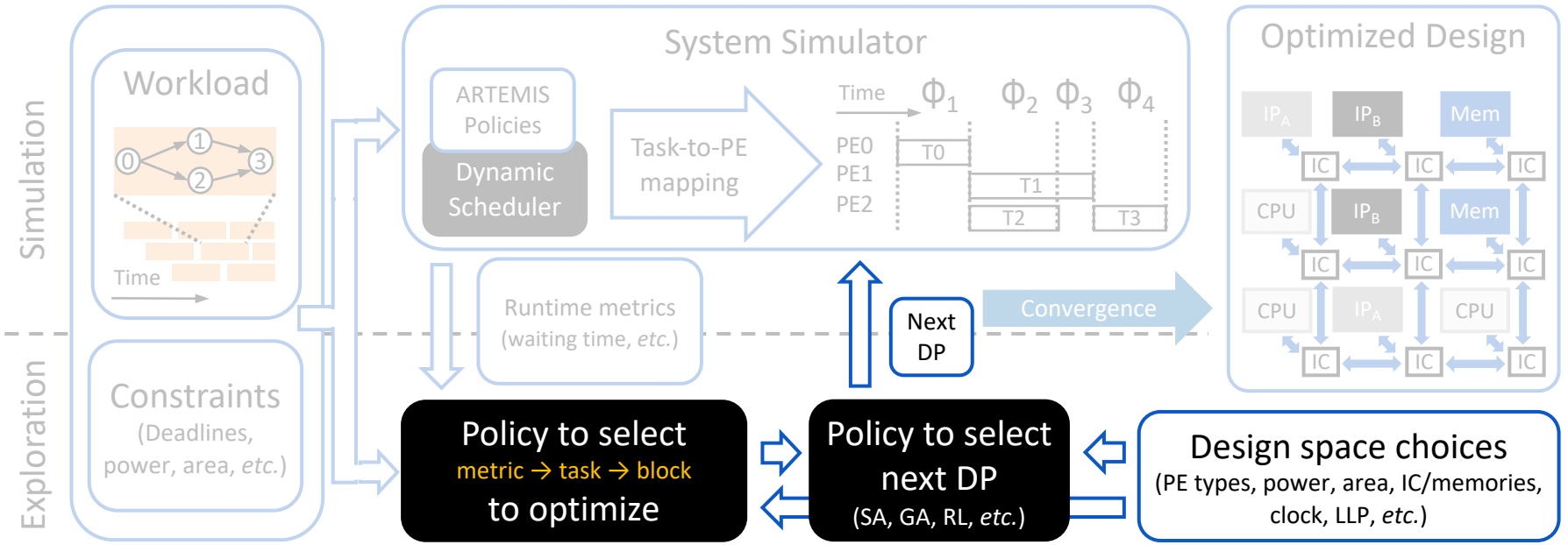
The ARTEMIS* Framework



- The simulator uses an SoC roofline model to perform phase-driven simulation
  - We use a dynamic scheduler in the simulator to perform fast and efficient task-to-PE mapping and identify bottlenecks in the system
- Several runtime statistics (e.g., task waiting times and task deadlines) are collected and fed into the explorer

***Subhankar Pal, Aporva Amarnath, Behzad Boroujerdian, Augusto Vega, Alper Buyuktosunoglu, John-David Wellman, Vijay Janapa Reddi, Pradip Bose, "ARTEMIS: Agile Discovery of Efficient Real-Time Systems-on-Chips in the Heterogeneous Era," HPCA 2025.**

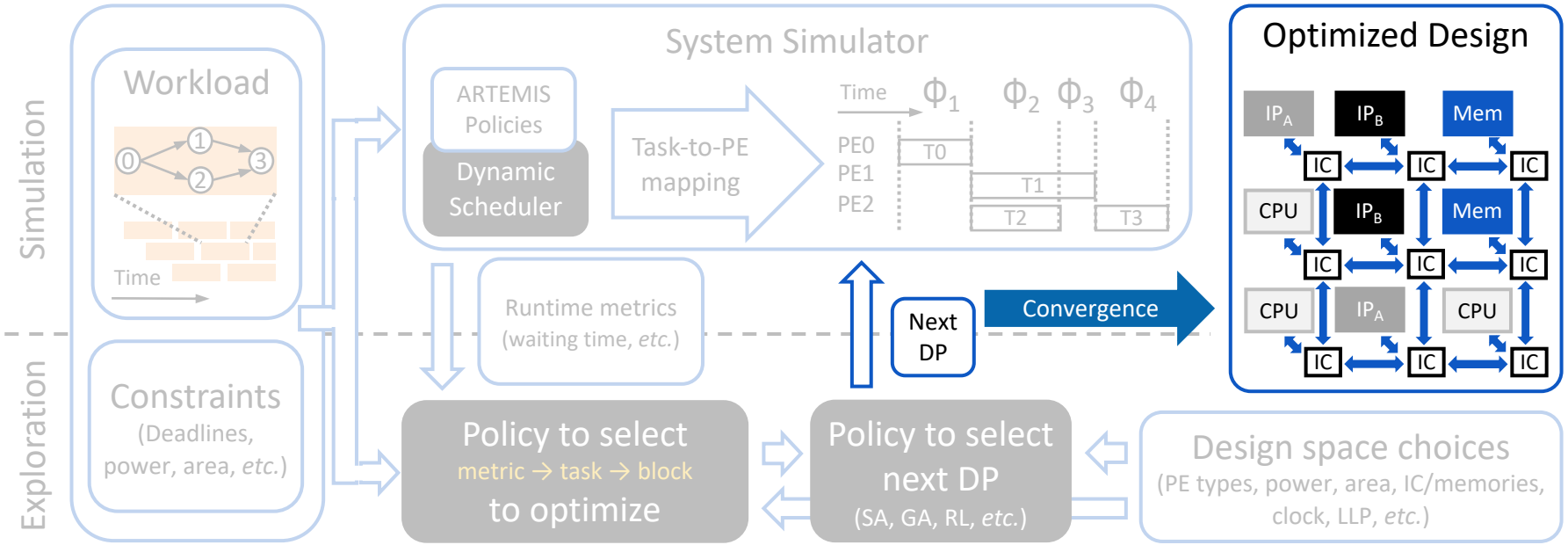
The ARTEMIS* Framework



- The explorer uses RT-aware policies to iteratively select ❶ the metric to optimize (latency/power/area), ❷ the task to optimize to improve this metric, and ❸ the hardware block to improve upon
- A library of pre-characterized PE/IC/memory blocks are fed in as inputs
- The next design point is selected based on an architecture-aware simulated annealing process

***Subhankar Pal, Aporva Amarnath, Behzad Boroujerdian, Augusto Vega, Alper Buyuktosunoglu, John-David Wellman, Vijay Janapa Reddi, Pradip Bose, "ARTEMIS: Agile Discovery of Efficient Real-Time Systems-on-Chips in the Heterogeneous Era," HPCA 2025.**

The ARTEMIS Framework



- DSE continues until ARTEMIS encounters a DP that meets deadlines for all DAGs, and has power and area within the constraints

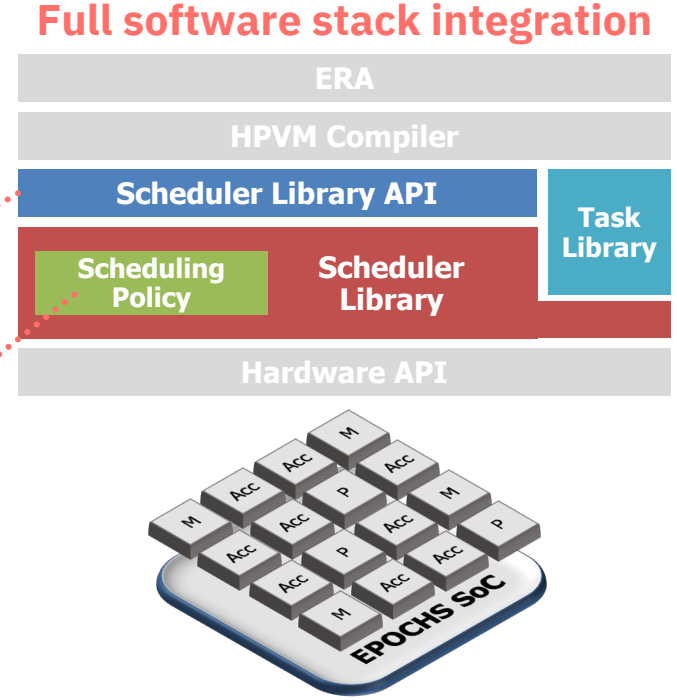
***Subhankar Pal, Aporva Amarnath, Behzad Boroujerdian, Augusto Vega, Alper Buyuktosunoglu, John-David Wellman, Vijay Janapa Reddi, Pradip Bose, "ARTEMIS: Agile Discovery of Efficient Real-Time Systems-on-Chips in the Heterogeneous Era," HPCA 2025.**

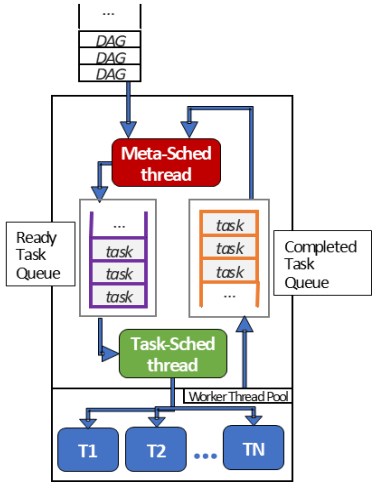
EPOCHS Appliance: The Smart Scheduler

EPOCHS Scheduler

github.com/IBM/scheduler-library

- Provides "smart" scheduling capabilities to applications in a hardware-independent manner
  - Schedule tasks across heterogeneous processing elements
  - Improve metrics of interest (throughput, criticality, efficiency)
  - Implemented as a user-level library (Scheduler Library)
- Exposes an API that abstracts applications from the scheduler
- User-specified scheduling policies can be easily "plug" into the scheduler





Two-level scheduling: **META policies (level 1)**

- Assign rank based on our HetSched paper [1]

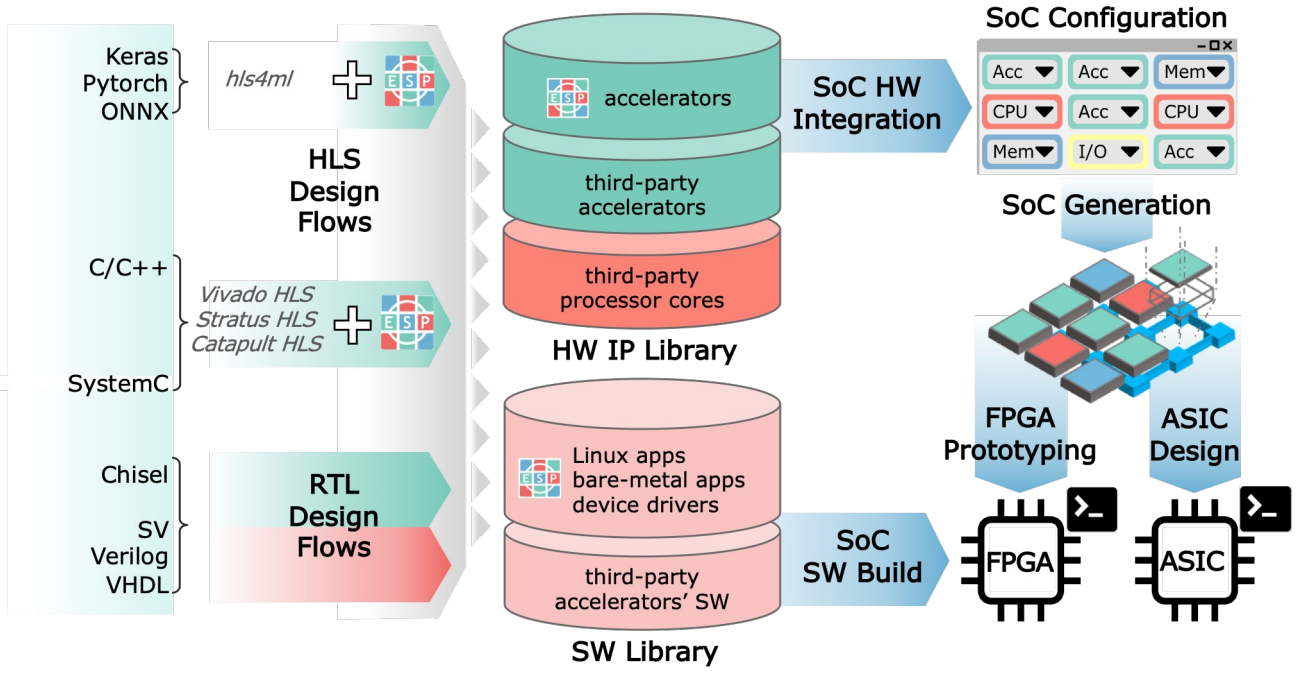
Two-level scheduling: **TASK policies (level 2)**
(choose the PE on HW where task should be executed)

[1] Amarnath, Aporva, et al. "Heterogeneity-Aware Scheduling on SoCs for Autonomous Vehicles", IEEE CAL (2021).

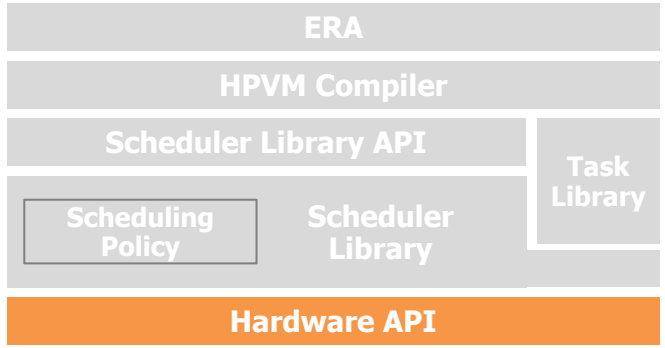
EPOCHS Appliance: The Agile SoC Flow

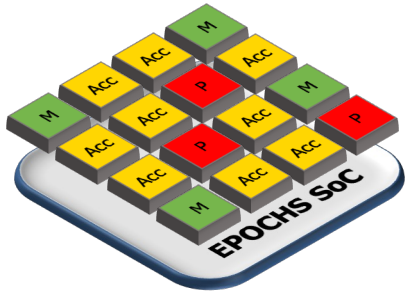
ESP: the open-source agile flow for system-on-chip (SoC) design

- *Seamless* integration of SoC components with NoC and platform services
- *Push-button* generation of complete SoC RTL for ASIC physical design
- **Two EPOCHS full SoC ASIC chip tapeouts to date**
- Rapid FPGA prototyping with software build for early application development



Full software stack integration





www.esp.cs.columbia.edu

Accomplishments Summary

EPOCHS-0 SoC tapeout

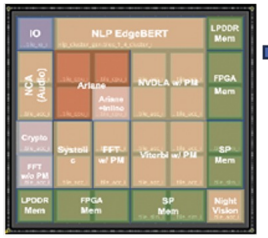
– 4×4 SoC fabricated

1.52 GHz peak @ 1.0 V
2.43mW @ 0.5V
1.83W @ 1.0 V
ESSCIRC-2022

Scaled-out EPOCHS-1 SoC tapeout

– 6×6 SoC with new accelerators

– Novel distributed h/w power manager



Significant design cost mitigation

– 10×–100× reduction in person-years

Hardware-agnostic programming of heterogeneous SoCs

– HPVM compiler, smart scheduler…

Open-source ecosystem for collaboration

ERI Summit 2023

SCAN ME

ERA: github.com/IBM/era	**HPVM:** gitlab.engr.illinois.edu/llvm/hpvm-release
Mini-ERA: github.com/IBM/mini-era	**STOMP:** github.com/IBM/stomp
ESP: www.esp.cs.columbia.edu	**Scheduler:** github.com/IBM/scheduler-library
Spandex: github.com/sld-columbia/esp/tree/master/rtl/caches	

Benefits of acceleration

	FFT	Viterbi
Performance	71×	20×
Energy	233×	56×

Tape-out: Sept. 2021, respin: Nov. 2022

ISSCC-2024

A 12nm Linux-SMP-Capable RISC-V SoC with 14 Accelerator Types, Distributed Hardware Power Management and NoC-Based Data Orchestration

Maico Cassel dos Santos[1]*, Tianyu Jia[2]*, Joseph Zuckerman[1]*, Martin Cochet[3]*, Davide Giri[1], Erik Loscalzo[1], Karthik Swaminathan[3], Thierry Tambe[2], Jeff Jun Zhang[2], Alper Buyuktosunoglu[3], Kuan-Lin Chiu[1], Giuseppe Di Guglielmo[1], Paolo Mantovani[1], Luca Piccolboni[1], Gabriele Tombesi[1], David Trilla[3], John-David Wellman[3], En-Yu Yang[2], Aporva Amarnath[3], Ying Jing[4], Bakshree Mishra[4], Joshua Park[2], Vignesh Suresh[4], Sarita Adve[4], Pradip Bose[3], David Brooks[2], Luca P. Carloni[1], Kenneth L. Shepard[1], Gu-Yeon Wei[2]
* These authors have equal contributions.

[1] Columbia University in the City of New York [2] Harvard University [3] IBM Research [4] University of Illinois

© 2024 IEEE International Solid-State Circuits Conference
14.5: A 12nm Linux-SMP-Capable RISC-V SoC with 14 Accelerator Types, Distributed Hardware Power Management and NoC-Based Data Orchestration
1 of 21

2024 ACM/IEEE 51st Annual International Symposium on Computer Architecture (ISCA)

ISCA-2024

BlitzCoin: Fully Decentralized Hardware Power Management for Accelerator-Rich SoCs

Martin Cochet[1], Karthik Swaminathan[1], Erik Loscalzo[2], Joseph Zuckerman[2], Maico Cassel dos Santos[2], Davide Giri[2], Alper Buyuktosunoglu[1], Tianyu Jia[3], David Brooks[3], Gu-Yeon Wei[3], Kenneth Shepard[2], Luca P. Carloni[2], and Pradip Bose[1]
[1]IBM Research, Yorktown Heights, NY [2]Columbia University, New York, NY [3]Harvard University, Cambridge, MA

Davide Giri (1990-2021): Key Contributor

Thank you for everything, Davide!

Karthik Swaminathan (1984 – 2026): KEY CONTRIBUTOR

Thank you from colleagues and friends 🙏

- **Star performer** in Pradip's team at IBM ("Efficient & Resilient Systems")
  - ✓ At IBM Research (fulltime) since 2014.
  - ✓ Summer intern before that.

- **Most recent key accomplishments:**
  - ✓ Key contributor to the DoW RAMP-C funded (SARA-1) AI/FHE hardware accelerator SoC design and implementation
    - ➢ IBM Research Accomplishment award: 2025
  - ✓ Novel decentralized hardware power management (DHPM) architecture and design – part of EPOCHS-1 SoC (DARPA)
    - ➢ **BlitzCoin** paper at ISCA-2024 ("Top Pick", IEEE Micro, 2025)
    - ➢ **EPOCHS-1 SoC** at ISSCC-2024; journal paper, JSSCC, 2026)
    - ➢ IBM Research accomplishment award (EPOCHS/DSSoC): 2024
  - ✓ Novel energy-efficient AI hardware research and development
    - ➢ Bit-error robustness for energy-efficient AI/ML systems:
      BERRY: DAC 2023, MulBERRY: ASPLOS 2024 (joint work, GaTech/IBM)
  - ✓ SERMiner and other methodology innovations in support of IBM's POWER and mainframe processor products: DSN-2021.

- **Ph.D in Computer Science and Engineering**, Penn State University (2014)
- **B.Tech + M.Tech in Electrical Engineering**, IIT Madras (2008)

ERI Summit 2023

SCAN ME

Thank You!

Pradip Bose **+ many colleagues** at IBM T. J. Watson Research Center
Sarita Adve, Vikram Adve, Sasa Misailovic, University of Illinois at Urbana-Champaign
Luca Carloni, Ken Shepard, Columbia University
David Brooks, Gu-Yeon Wei, Vijay Janapa Reddi, Harvard University
plus many talented students and postdoctoral fellows