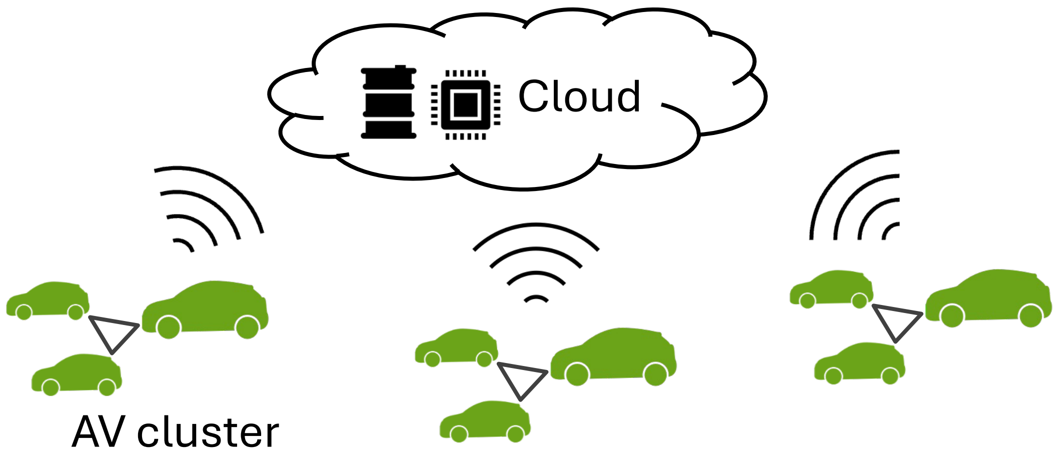
# Accelerating the Discovery of Efficient Real-Time Systems-on-Chips in the Heterogeneous Era

S. Pal*, A. Amarnath*, B. Boroujerdian^, A. Vega*, A. Buyuktosunoglu*, J.-D. Wellman*, V. J. Reddi^, P. Bose*

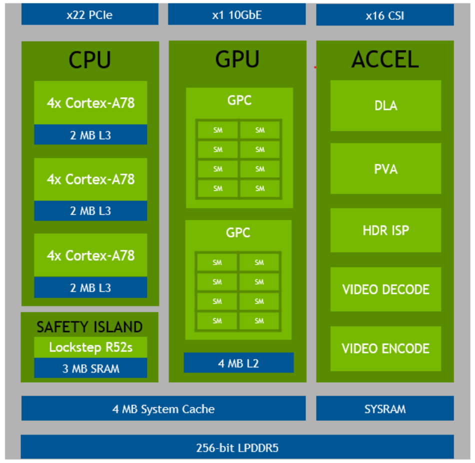
*IBM T.J. Watson Research Center ^Harvard University

Email: Subhankar.Pal@ibm.com

# Heterogeneity in Modern SoCs



Cloud-Backed Connected Vehicles at the Edge



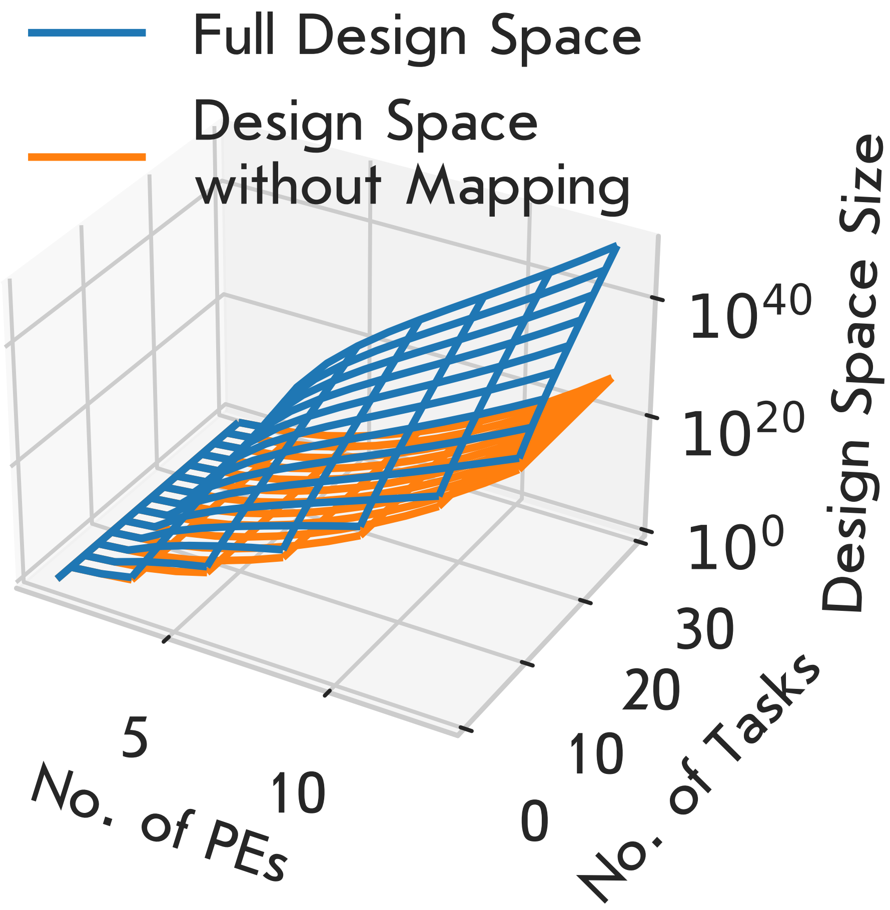
Jetson Orin SoC Block Diagram [1]

- Heterogeneous SoCs are gaining ubiquity for real-time (RT) edge processing, subject to strict deadlines and power/area constraints
- NVIDIA DRIVE platform, e.g., uses 2 Orin SoCs [1] and inputs from 28 sensors
- Agile and efficient design space exploration (DSE) is crucial to make optimal decisions at early stages of design

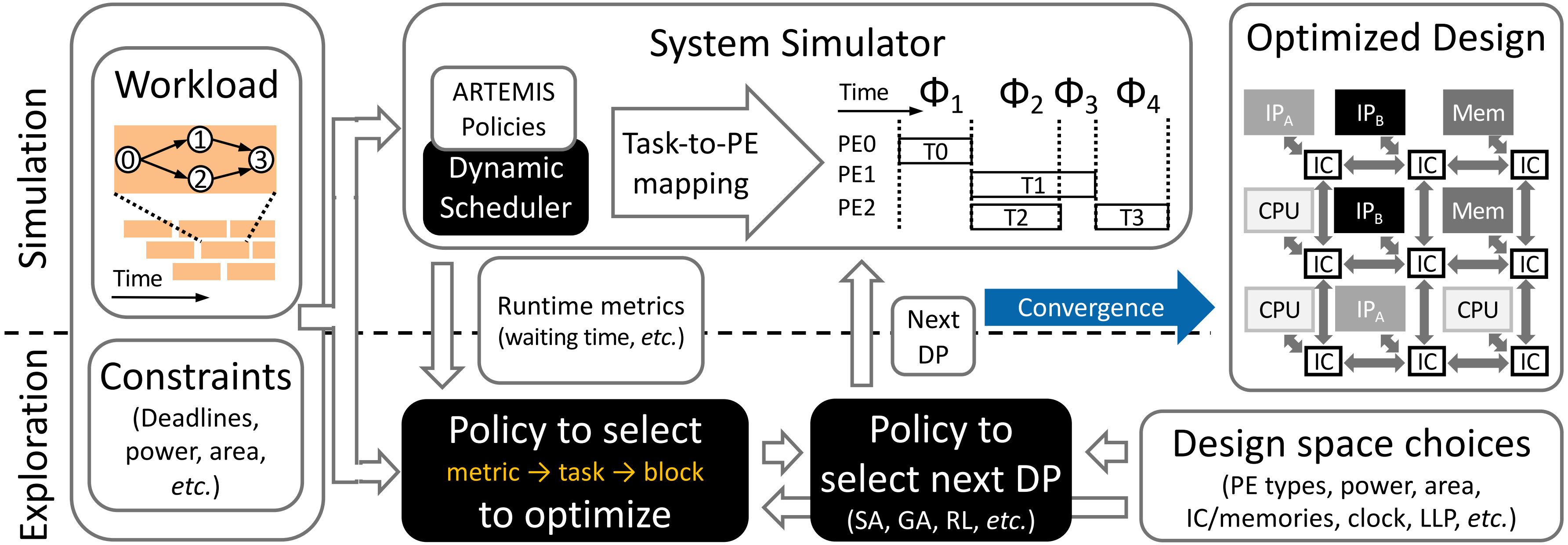
[1] https://www.nvidia.com/content/dam/en-zz/Solutions/gtcf21/jetson-orin/nvidia-jetson-agx-orin-technical-brief.pdf

# Scalability Challenges

- The DSE process, consisting of architectural exploration and task-to-hardware mapping is expensive
- Even a system with a few PE choices and 10s of tasks contains as many design points (DPs) as the number of stars in the Universe!
- **Insight #1**: prior works consider mapping as a static DSE parameter, thereby compounding the design space
- **Insight #2**: prior works do not leverage dynamic run-time insights from the evaluation of DPs for the DSE itself

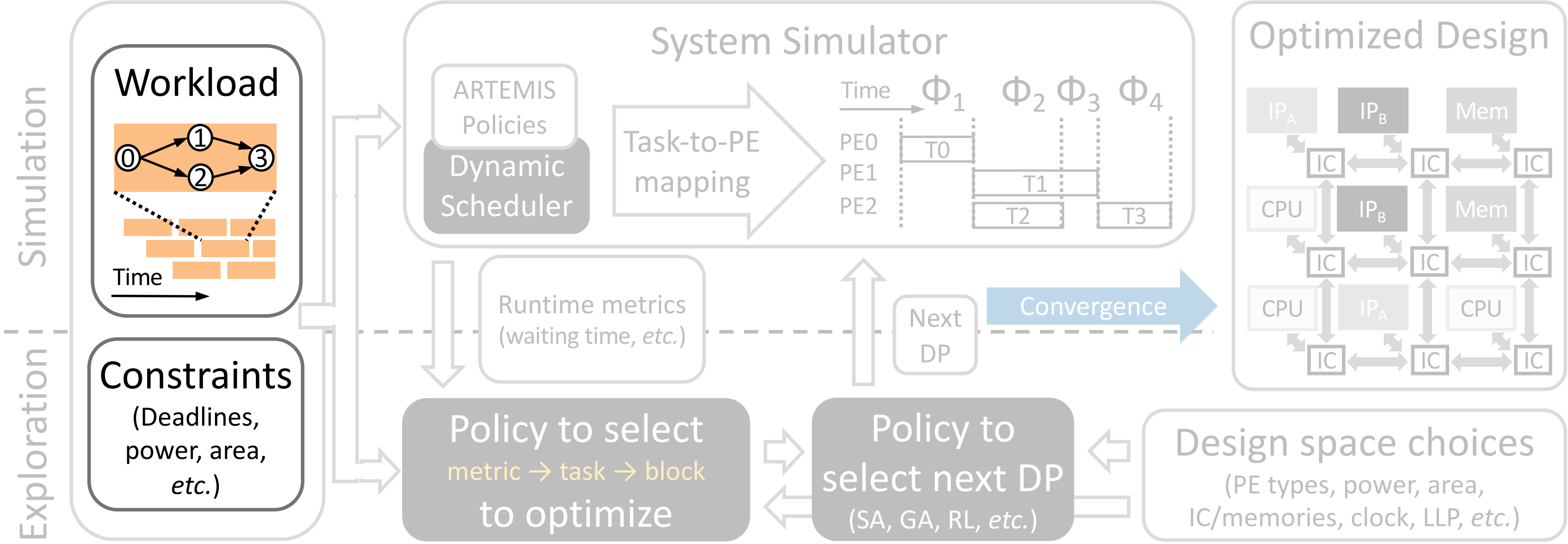


# The ARTEMIS Framework



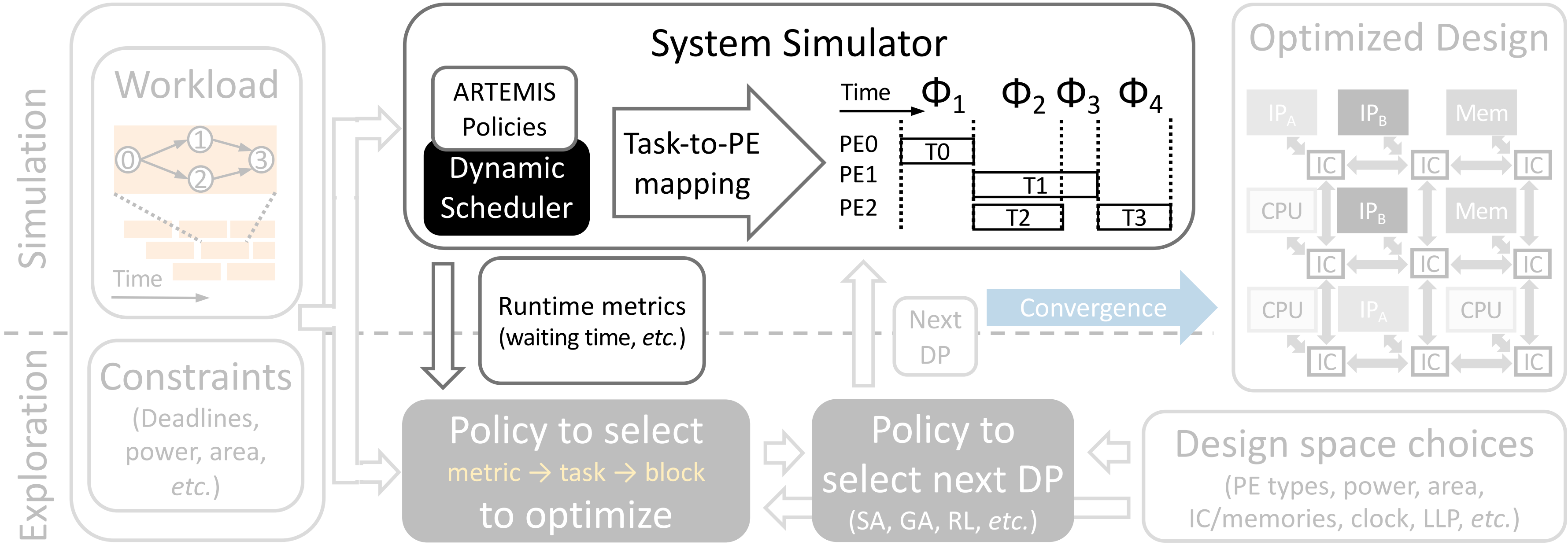
- ARTEMIS consists of an exploration and a simulation framework
  - Begins with a seed SoC DP and incrementally transforms it to an optimized DP that meets the pre-specified constraints

# The ARTEMIS Framework


Simulation
Exploration
Workload
0
1
2
3
Time
Constraints
(Deadlines, power, area, etc.)
System Simulator
ARTEMIS Policies
Dynamic Scheduler
Task-to-PE mapping
Time
$\Phi_1$
$\Phi_2$
$\Phi_3$
$\Phi_4$
PE0
PE1
PE2
T0
T1
T2
T3
Runtime metrics (waiting time, etc.)
Next DP
Convergence
Optimized Design
$IP_A$
$IP_B$
Mem
IC
CPU
Policy to select metric → task → block to optimize
Policy to select next DP (SA, GA, RL, etc.)
Design space choices (PE types, power, area, IC/memories, clock, LLP, etc.)

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

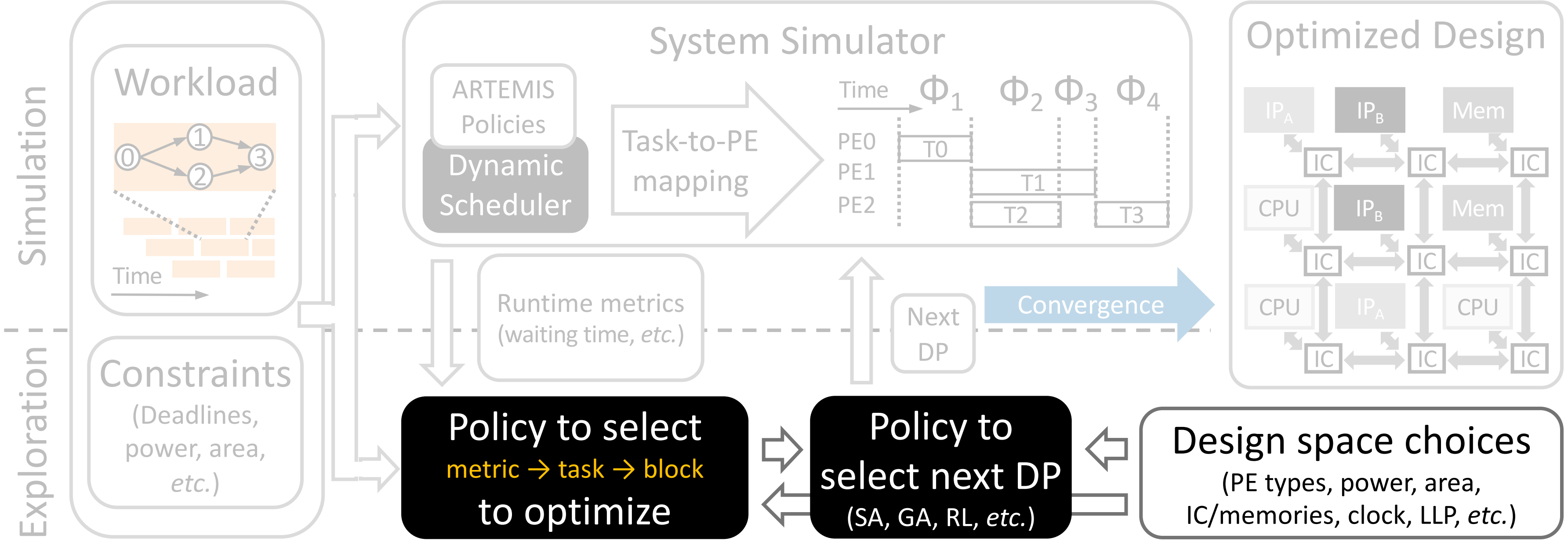
# The ARTEMIS Framework


Simulation
Exploration
Workload
0
1
2
3
Time
Constraints
(Deadlines, power, area, *etc.*)
System Simulator
ARTEMIS Policies
Dynamic Scheduler
Task-to-PE mapping
Time
$\Phi_1$
$\Phi_2$
$\Phi_3$
$\Phi_4$
PE0
PE1
PE2
T0
T1
T2
T3
Runtime metrics (waiting time, *etc.*)
Next DP
Convergence
Optimized Design
$IP_A$
$IP_B$
Mem
IC
CPU
Policy to select metric → task → block to optimize
Policy to select next DP (SA, GA, RL, *etc.*)
Design space choices (PE types, power, area, IC/memories, clock, LLP, *etc.*)

- The simulator uses the Gables SoC roofline models and augmented with features such as task-to-task dependency to perform phase-driven simulation, derived from [2]
- During simulation, several statistics, e.g., task waiting times, task deadlines, etc., are computed and fed into the DSE engine

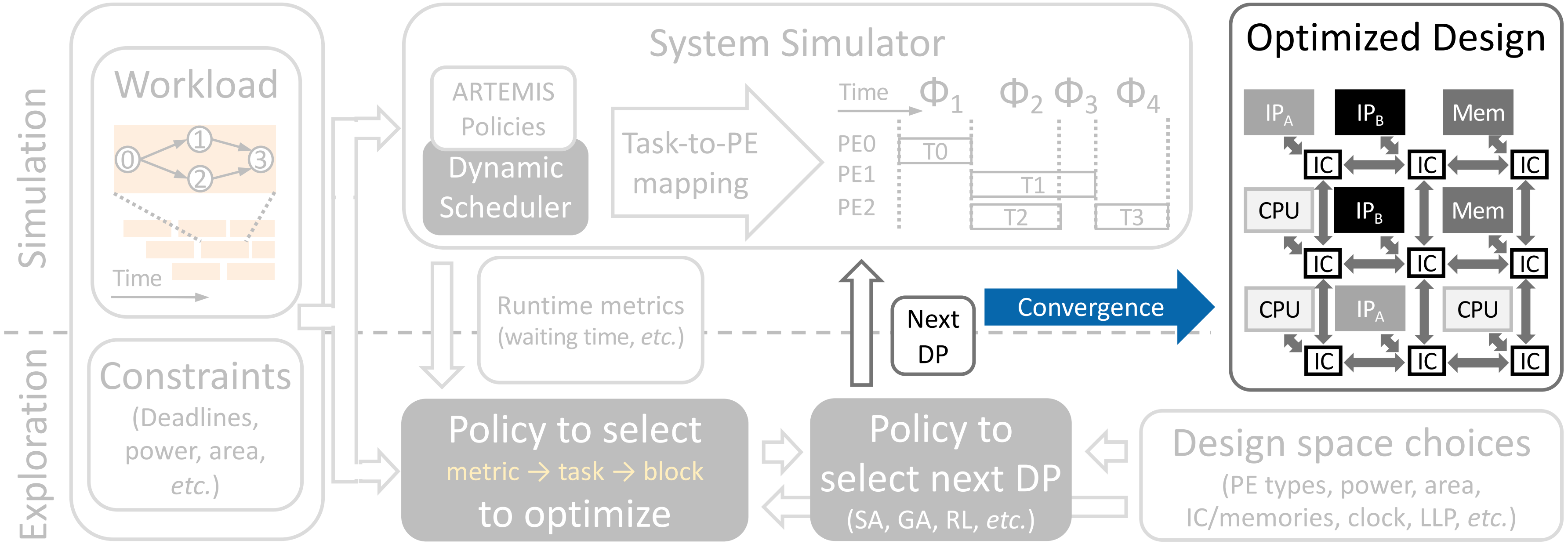
[2] B. Boroujerdian, et al., “FARSI: An Early-stage Design Space Exploration Framework to Tame the Domain-specific System-on-chip Complexity”, TECS ’22.

# 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 architecture-aware simulated annealing

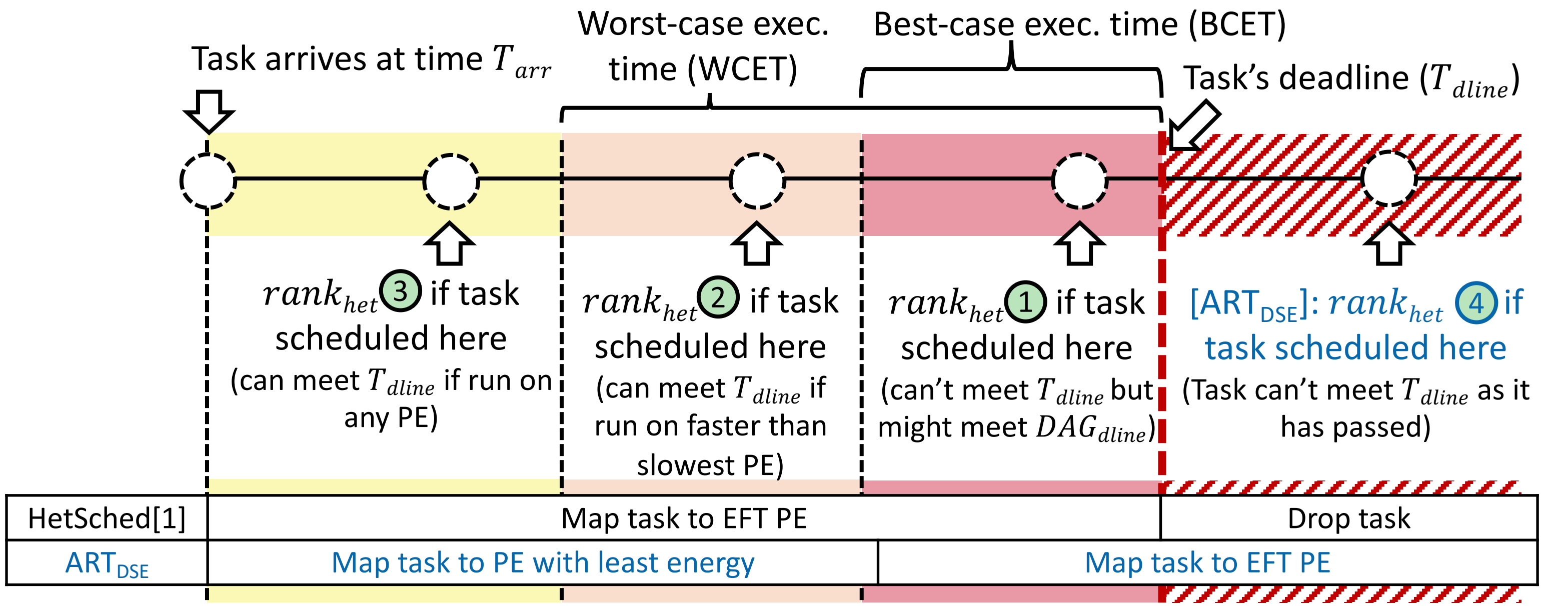
# The ARTEMIS Framework


Simulation
Exploration
Workload
0
1
2
3
Time
Constraints
(Deadlines, power, area, *etc.*)
System Simulator
ARTEMIS Policies
Dynamic Scheduler
Task-to-PE mapping
Time
$\Phi_1$
$\Phi_2$
$\Phi_3$
$\Phi_4$
PE0
PE1
PE2
T0
T1
T2
T3
Runtime metrics (waiting time, *etc.*)
Next DP
Convergence
Optimized Design
$IP_A$
$IP_B$
Mem
IC
CPU
Policy to select metric → task → block to optimize
Policy to select next DP (SA, GA, RL, *etc.*)
Design space choices (PE types, power, area, IC/memories, clock, LLP, *etc.*)

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

# Scheduling Policy Adapted for DSE

- We propose a dynamic scheduling policy, called $ART_{DSE}$, to reduce the design space size; RT-metrics extracted from it are further used to efficiently navigate the design space



Task arrives at time $T_{arr}$

Worst-case exec. time (WCET)

Best-case exec. time (BCET)

Task's deadline ($T_{dline}$)

$rank_{het}$ ③ if task scheduled here (can meet $T_{dline}$ if run on any PE)

$rank_{het}$ ② if task scheduled here (can meet $T_{dline}$ if run on faster than slowest PE)

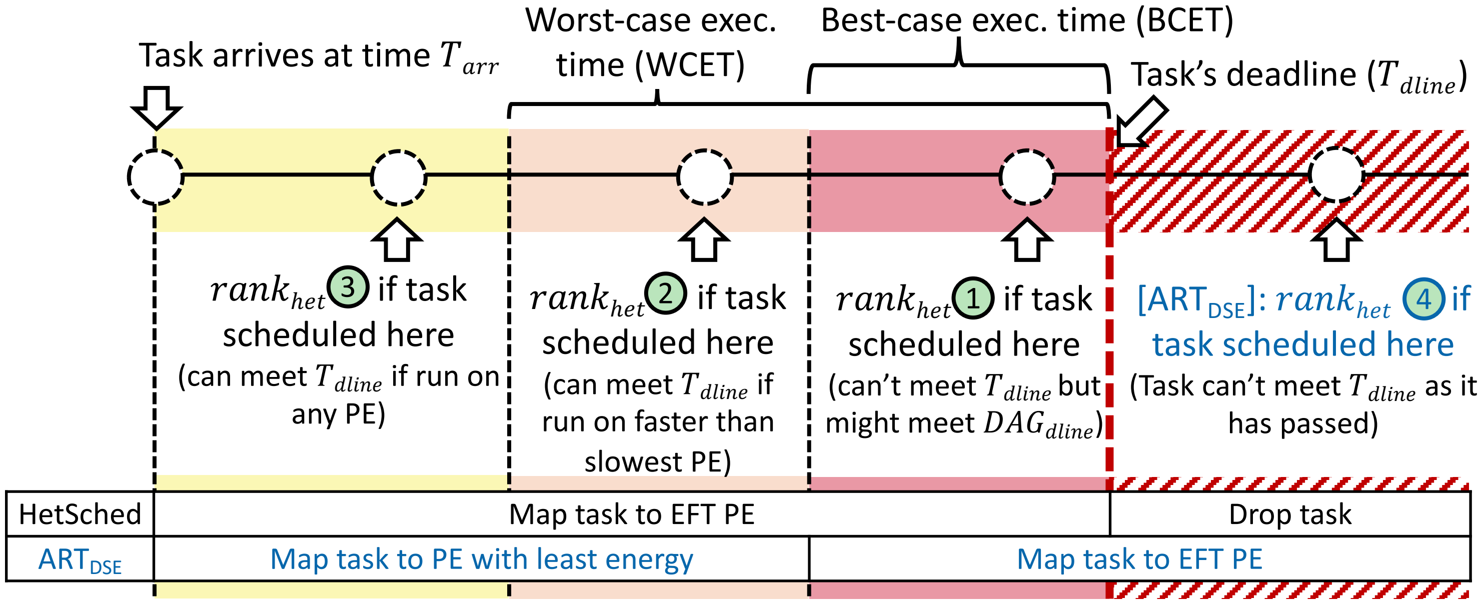
$rank_{het}$ ① if task scheduled here (can't meet $T_{dline}$ but might meet $DAG_{dline}$)

[ART$_{DSE}$]: $rank_{het}$ ④ if task scheduled here (Task can't meet $T_{dline}$ as it has passed)

<table>
<tr><td>HetSched[1]</td><td colspan="2">Map task to EFT PE</td><td>Drop task</td></tr>
<tr><td>ART<sub>DSE</sub></td><td>Map task to PE with least energy</td><td colspan="2">Map task to EFT PE</td></tr>
</table>

[1] A. Amarnath et al., "Heterogeneity-Aware Scheduling on SoCs for Autonomous Vehicles", IEEE CAL, vol. 20, no. 2, pp. 82-85, 1 July-Dec. 2021.

# Scheduling Optimizations to Aid the DSE



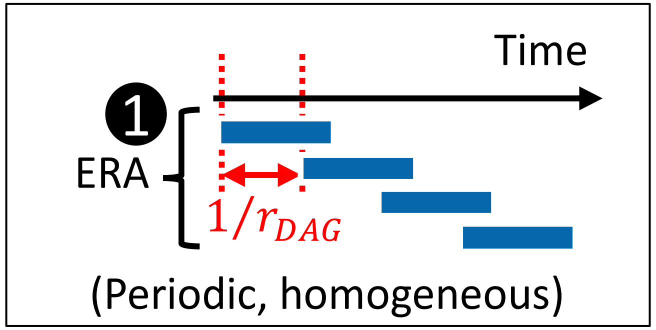
Task arrives at time $T_{arr}$
Worst-case exec. time (WCET)
Best-case exec. time (BCET)
Task's deadline ($T_{dline}$)
$rank_{het}$ ③ if task scheduled here (can meet $T_{dline}$ if run on any PE)
$rank_{het}$ ② if task scheduled here (can meet $T_{dline}$ if run on faster than slowest PE)
$rank_{het}$ ① if task scheduled here (can't meet $T_{dline}$ but might meet $DAG_{dline}$)
[ART$_{DSE}$]: $rank_{het}$ ④ if task scheduled here (Task can't meet $T_{dline}$ as it has passed)

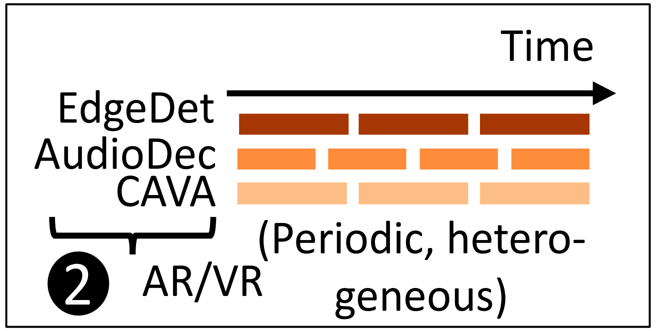
| | | | | |
|---|---|---|---|---|
| HetSched | Map task to EFT PE | | | Drop task |
| ART$_{DSE}$ | Map task to PE with least energy | | Map task to EFT PE | |

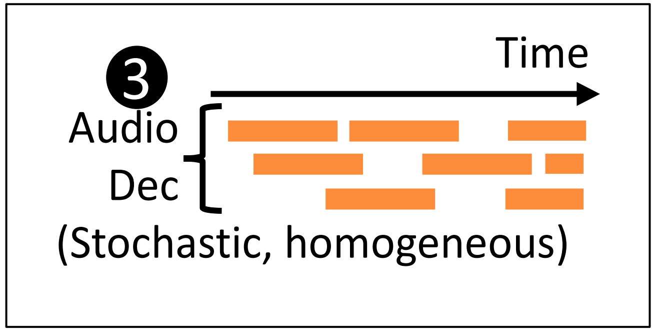
- **Task procrastination**: $ART_{DSE}$ executes tasks that have failed to meet their deadlines, but with the lowest priority; this exposes the task to the DSE engine for acceleration
- **Energy-aware scheduling**: $ART_{DSE}$ identifies tasks that can execute on slower (but lower-power) PEs and still meet their deadlines, thereby optimizing for the overall energy of the system
- **NoC-traffic-aware scheduling**: $ART_{DSE}$ estimates the expected NoC traffic before scheduling a task, based on tasks reading from/writing to memory; this is used to dynamically de-prioritize memory-bound tasks to expose them via task procrastination

# Experiments

- Evaluated against FARSI [2], simulated annealing, MOOS [3] and a two-phase heterogeneous DSE technique [4], on three types of workloads
- **Periodic, homogeneous DAGs**
  - ERA is an AV workload where NCV images, NRad radar inputs, NVit WiFi receiver streams of an AV are processed before a deadline
- **Periodic, heterogeneous DAGs**
  - The AR/VR workload plays back the audio based on the user's pose, pre-processes images to feed to a neural network backend, and finds sharp changes in brightness in input images and detects objects of interest
- **Stochastic, homogeneous DAGs**
  - Represents a signal processing application that, for instance, mixes audio signals from different sources in real-time

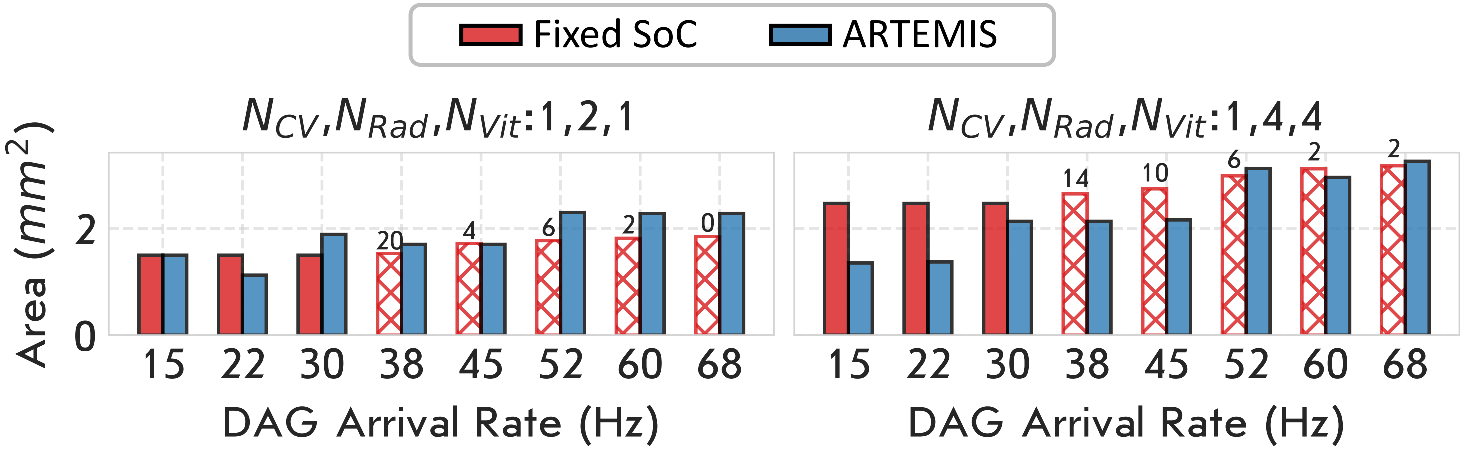






[2] B. Boroujerdian, et al., "FARSI: An Early-stage Design Space Exploration Framework to Tame the Domain-specific System-on-chip Complexity", TECS '22.
[3] A. Deshwal, et al., "MOOS: A Multi-Objective Design Space Exploration and Optimization Framework for NoC Enabled Manycore Systems", TECS '19.
[4] Z. J. Jia, et al., "A Two-Phase Design Space Exploration Strategy for System-Level Real-Time Application Mapping onto MPSoC", Microprocessors & Microsystems 38, 1 (2014), 9–21.

# Evaluation for ERA: vs. Fixed SoC

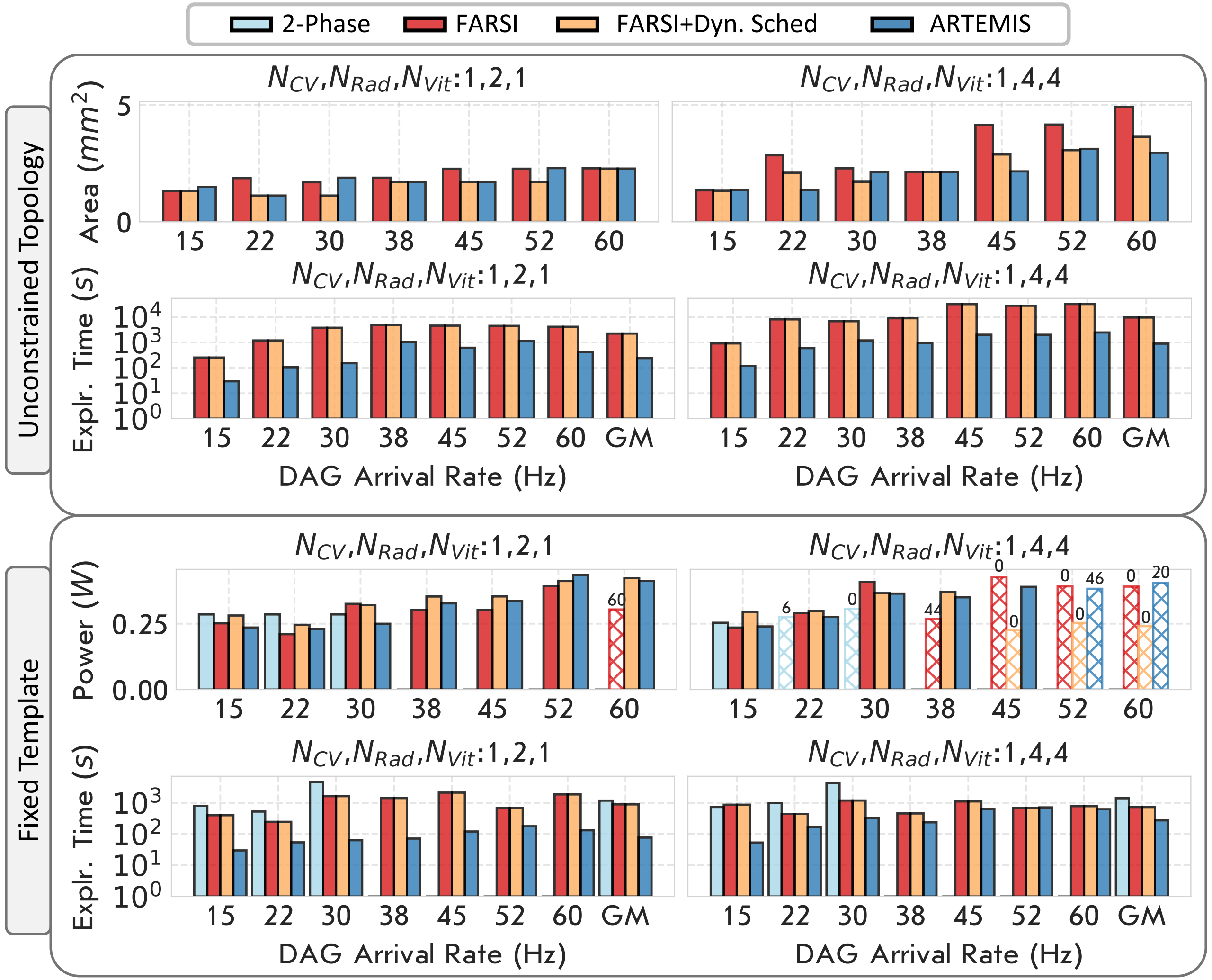


- We compare against $SoC_{fixed}$: 1 CPU, $N_{CV}$ CV IPs, $N_{Rad}$ Radar IPs, $N_{Vit}$ Viterbi IPs
  - The most obvious selection of IPs that the designer may come up with
- ARTEMIS-generated SoCs ($SoC_{ART}$) consume up to 2× lower area compared to $SoC_{fixed}$, which is overprovisioned for slow arrival rates and under-provisioned for faster rates
- $SoC_{ART}$ delivers 1.5× better sustained throughput than $SoC_{fixed}$, at iso-area

# Evaluation for ERA: vs. Prior Work

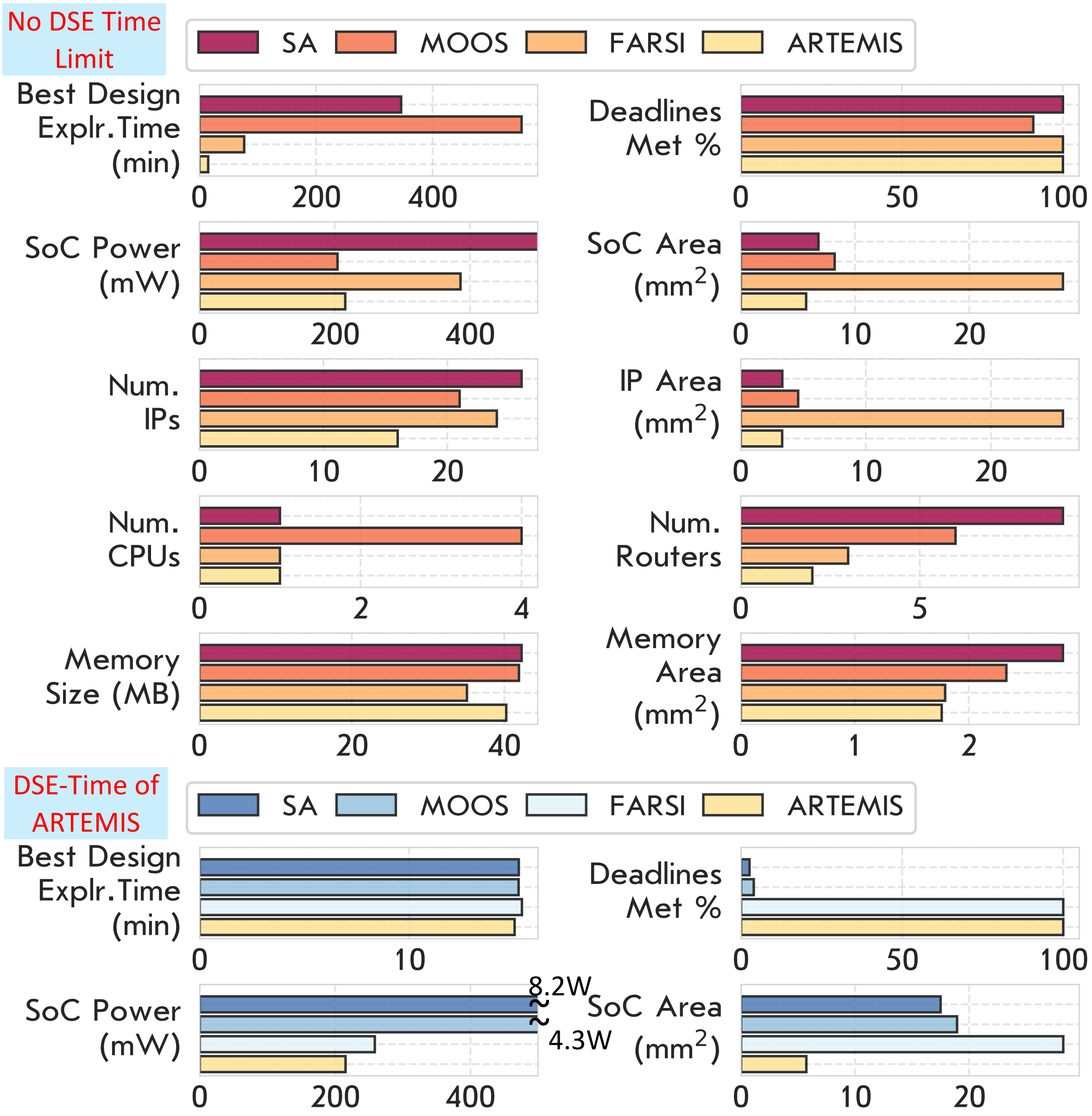
- **Unconstrained topology mode:**
  - $\text{SoC}_{\text{ART}}$ has 2.4× lower area footprint than $\text{SoC}_{\text{FARSI}}$
  - ARTEMIS takes 9.6–13.2× less time for optimal SoC discovery, than FARSI

- **Fixed-template mode:**
  - $\text{SoC}_{\text{ART}}$ has 1.2×, 1.5× & 3× better sustained throughput over $\text{SoC}_{\text{FARSI}}$ (w/ dynamic scheduling), $\text{SoC}_{\text{FARSI}}$, $\text{SoC}_{\text{2\_phase}}$, respectively
  - Converges the DSE loop with 8.4–12.8× faster wall time



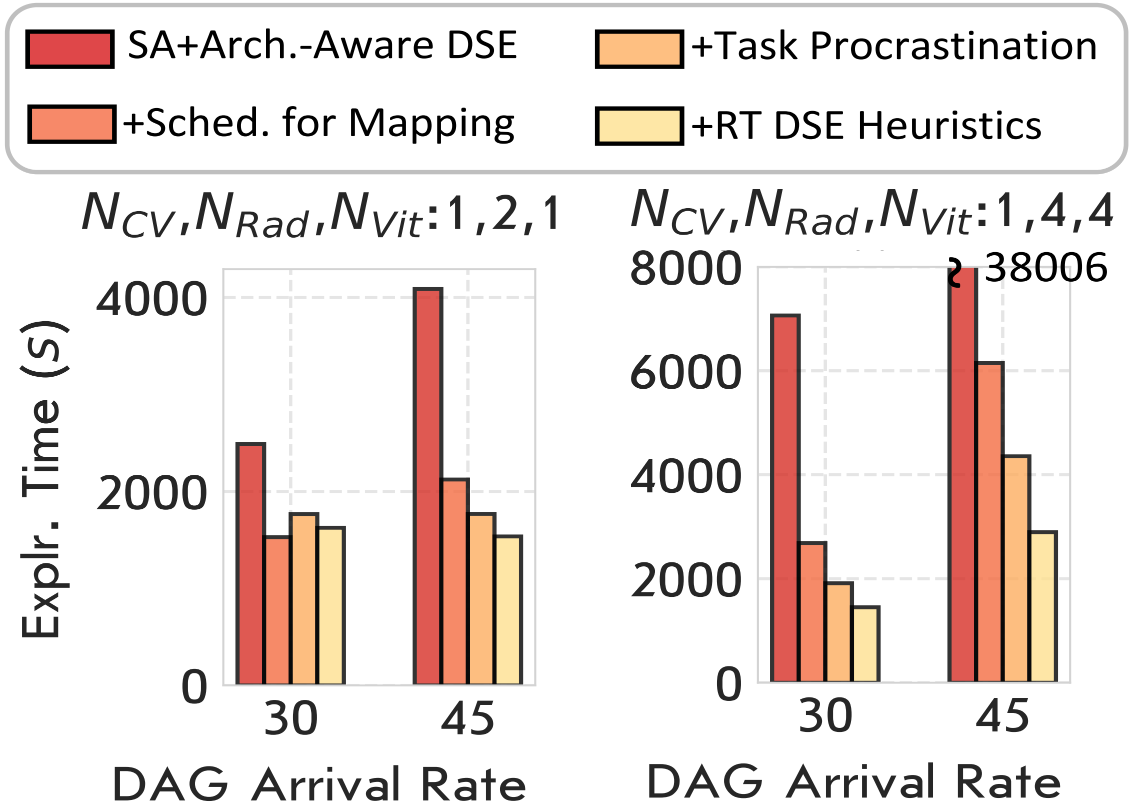
# Evaluation for AR/VR

- **5.1–36.8×** times faster DSE vs. the baseline frameworks
- 100% deadlines met with a more compact SoC design than prior work
- When DSE time is restricted to ARTEMIS’s convergence time, FARSI over-provisions, and MOOS/SA produce bad DPs that neither meet deadlines nor the power budget



# Breakdown of Benefits with Each Feature

- Using the scheduler for mapping offers 6.2× DSE speedup
- Task procrastination provides an additional 1.4× speedup for the high congestion case
- RT-aware metric, task and block selection provides an added benefit of up to 1.5×
- **1.5–13.1× net speedup**



# Conclusions and Future Work

- We propose ARTEMIS, a framework for agile DSE of real-time heterogeneous SoCs
- Uses scheduler for mapping, task+ procrastination, energy-aware, NoC traffic-aware scheduling for DSE
- Demonstrated AV and AR/VR SoCs with better PPA metrics than prior work, with 5.1–12.8× reduction in DSE time
- Future work would implement support for DAG-to-DAG dependencies and chiplet-level hierarchy support for DSE
- Open sourcing the framework on GitHub in progress

# Acknowledgments

This research was sponsored by DARPA. The views and findings should not be interpreted as representing the official views or policies of DoD or the U.S. Government

# Thank You

Questions?