DORA: Dataflow-Instruction Orchestration Architecture for DNN Acceleration

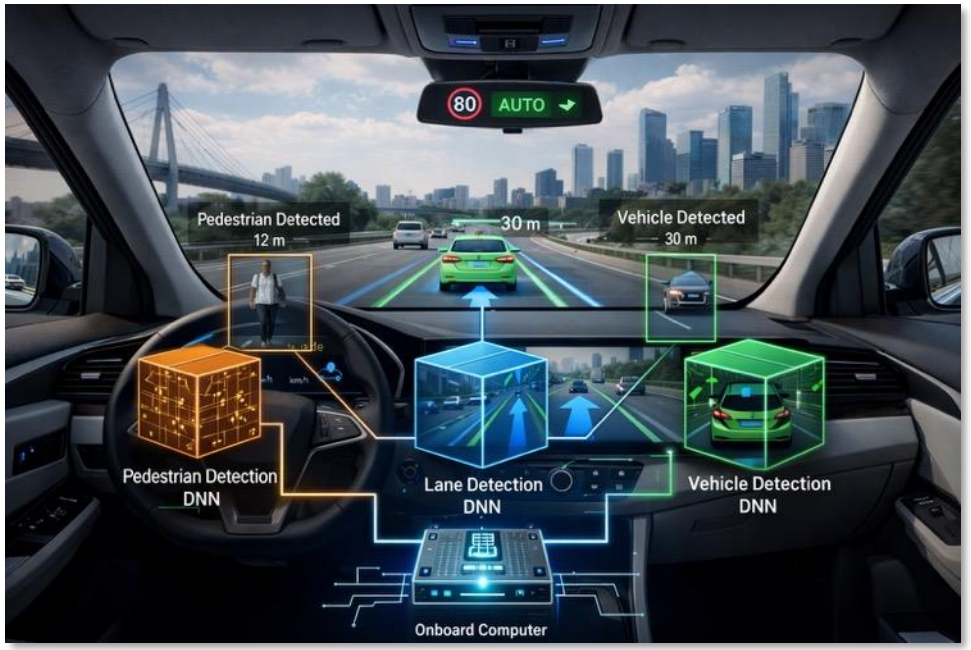
Xingzhen Chen, Zhuoping Yang, Jinming Zhuang, Shixin Ji, Sarah Schultz,
Zheng Dong*, Weisong Shi† and Peipei Zhou

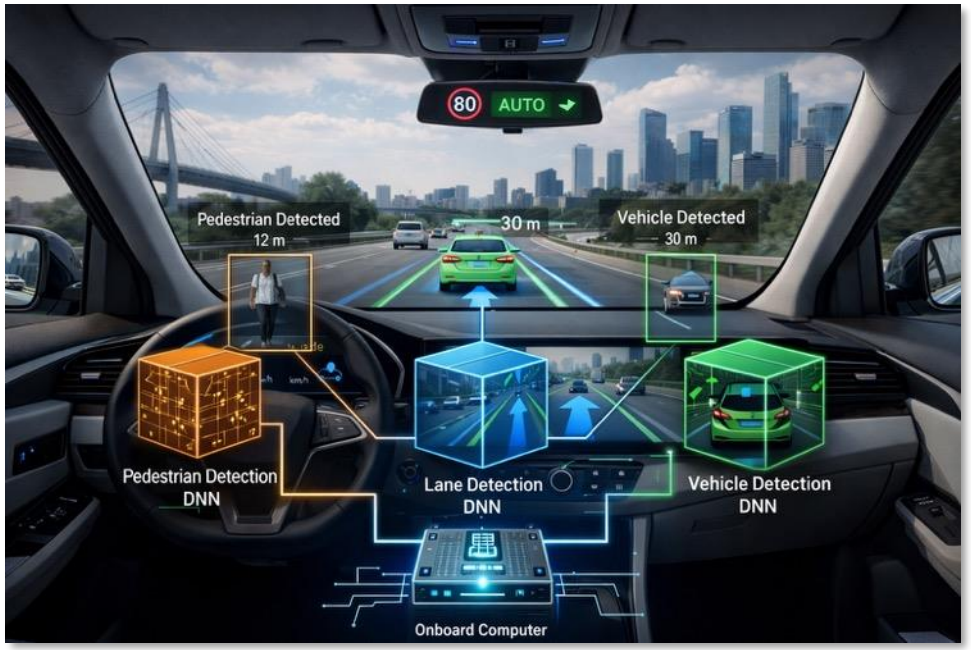
Brown University; Wayne State University*; University of Delaware†

xingzhen_chen@brown.edu
peipei_zhou@brown.edu

https://peipeizhou-eecs.github.io/

https://github.com/arc-research-lab/DORA.git


80
AUTO
Pedestrian Detected
12 m
30 m
Vehicle Detected
30 m
Pedestrian Detection
DNN
Lane Detection
DNN
Vehicle Detection
DNN
Onboard Computer



Tasks behind modern autonomous driving systems.

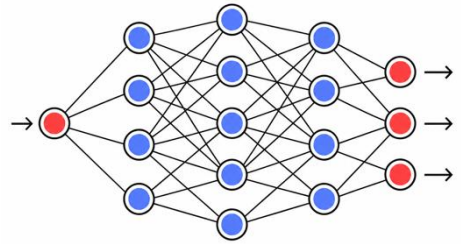
Challenge 1

- Autonomous driving system requires diverse DNN models.

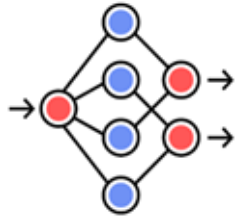
Challenge 1

- Autonomous driving system requires diverse DNN models.

ADS applications



Large DNNs

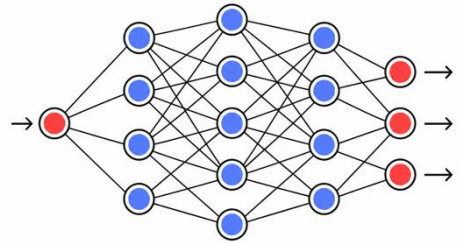


Small DNNs

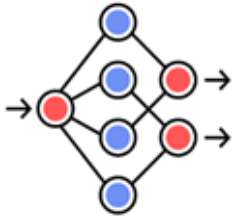
Challenge 1

- Autonomous driving system requires diverse DNN models.

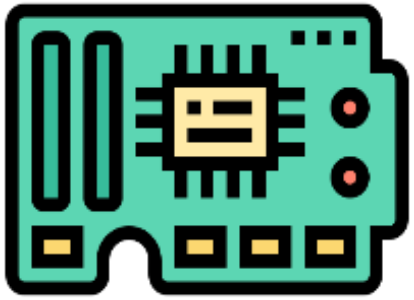
ADS applications



Large DNNs



Small DNNs



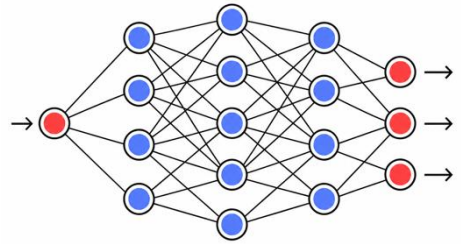
Accelerator for large workloads.

- Good for large DNNs;
- Bad for small DNNs.

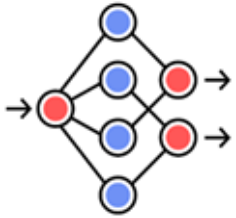
Challenge 1

- Autonomous driving system requires diverse DNN models.

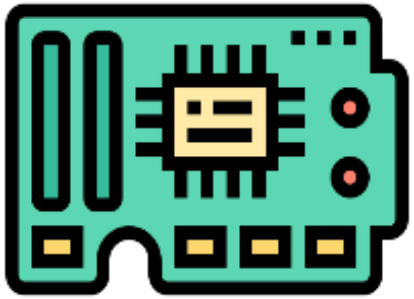
ADS applications



Large DNNs

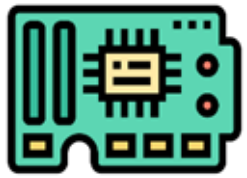


Small DNNs



Accelerator for large workloads.

- Good for large DNNs;
- Bad for small DNNs.



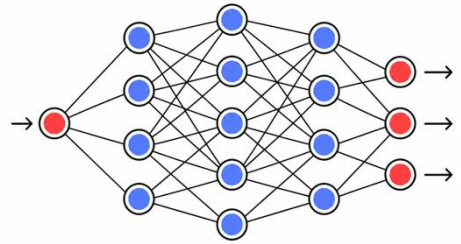
Accelerator for small workloads.

- Bad for large DNNs;
- Good for small DNNs.

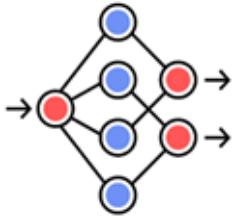
Challenge 1

- Autonomous driving system requires **diverse** DNN models.

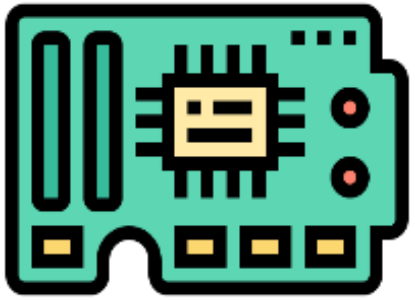
ADS applications



Large DNNs



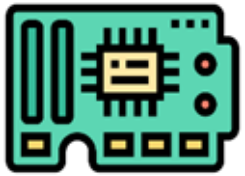
Small DNNs



Accelerator for large workloads.

- Good for large DNNs;
- Bad for small DNNs.

Switching bitstream requires ~20ms.



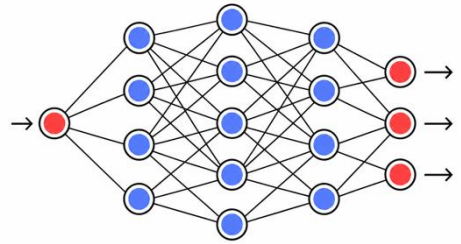
Accelerator for small workloads.

- Bad for large DNNs;
- Good for small DNNs.

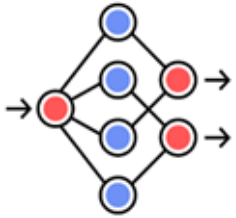
Challenge 1

- Autonomous driving system requires diverse DNN models.

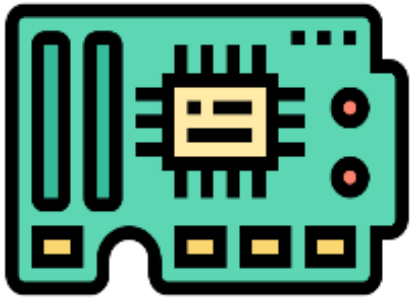
ADS applications



Large DNNs



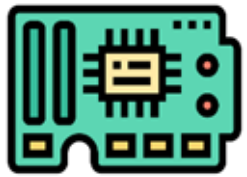
Small DNNs



Accelerator for large workloads.

- Good for large DNNs;
- Bad for small DNNs.

Switching bitstream requires ~20ms.



Accelerator for small workloads.

- Bad for large DNNs;
- Good for small DNNs.

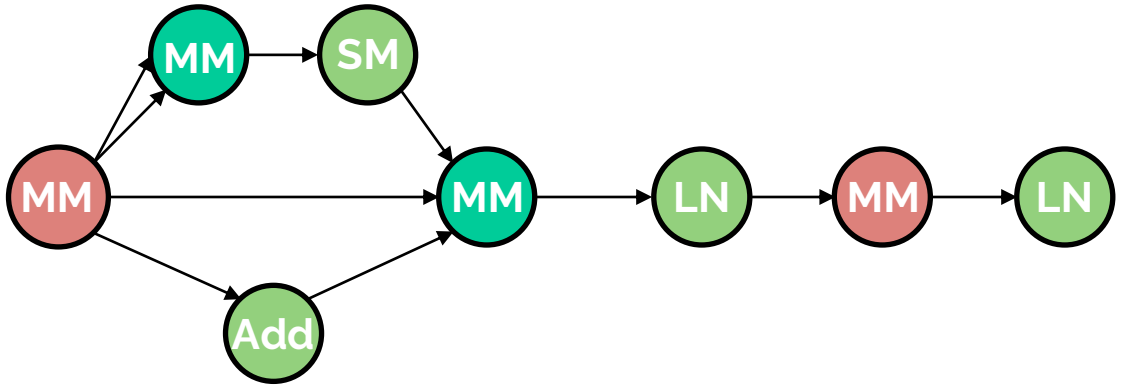
- How to efficiently handle diverse DNN models in one bitstream?

Challenge 2

- Complicated applications lead to a huge design space.

Challenge 2

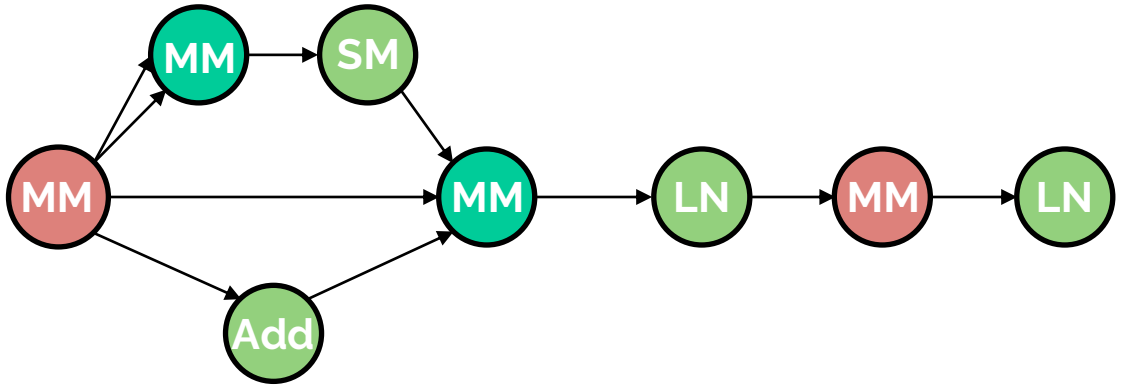
- Complicated applications lead to a **huge design space**.



Challenge 2

- Complicated applications lead to a huge design space.

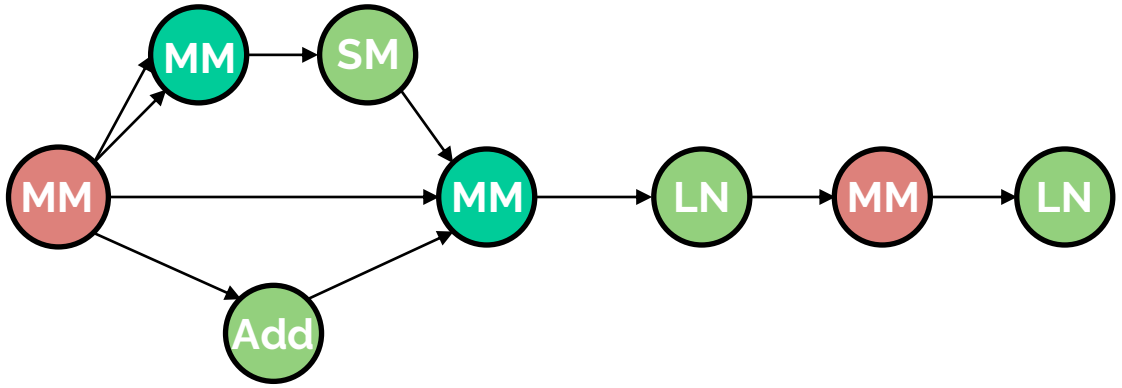
- The design space is exponential to # of layers, which is $O(m^n)$.
  - m is # of mapping solutions for a single layer.
  - n is # of layers.



Challenge 2

- Complicated applications lead to a huge design space.

- The design space is exponential to # of layers, which is $O(m^n)$.
  - m is # of mapping solutions for a single layer.
  - n is # of layers.



950x The U.S. GDP in 2025.

Challenge 3

- Given a task set, automatic framework requires balancing the optimality and scalability.

Challenge 3

- Given a task set, automatic framework requires balancing the optimality and scalability.
  - Brute-force search induces redundant search time.

Challenge 3

- Given a task set, automatic framework requires balancing the optimality and scalability.
  - Brute-force search induces redundant search time.
  - Heuristic search loses many opportunities for better mapping solutions.

Challenge 3

- Given a task set, automatic framework requires balancing the optimality and scalability.
  - Brute-force search induces redundant search time.
  - Heuristic search loses many opportunities for better mapping solutions.
- How to maintain the optimality while preserving the efficiency?

DORA: Dataflow-Instruction Orchestration Architecture for DNN Acceleration

DORA: Dataflow-Instruction Orchestration Architecture for DNN Acceleration

Challenges

- Diverse operations exist in modern DNN models.
- Complicated applications lead to a huge design space.
- Balance between optimality and scalability.

DORA: Dataflow-Instruction Orchestration Architecture for DNN Acceleration

Challenges

- Diverse operations exist in modern DNN models.
- Complicated applications lead to a huge design space.
- Balance between optimality and scalability.

Solutions

- DORA proposes an overlay architecture as well as an ISA abstraction for diverse workloads.
- DORA is equipped with an automatic framework that can analyze the mapping solutions at a fine-grained layer level.
- DORA has a two-stage design space exploration that combines brute-force and heuristic search.

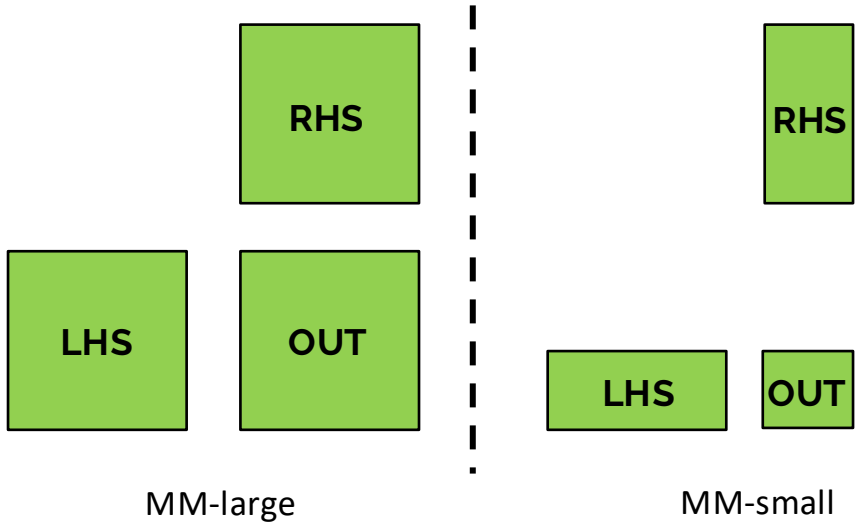
DORA Architecture Design

DORA Architecture Design

- Enable flexible computation tile size by parallelism management.

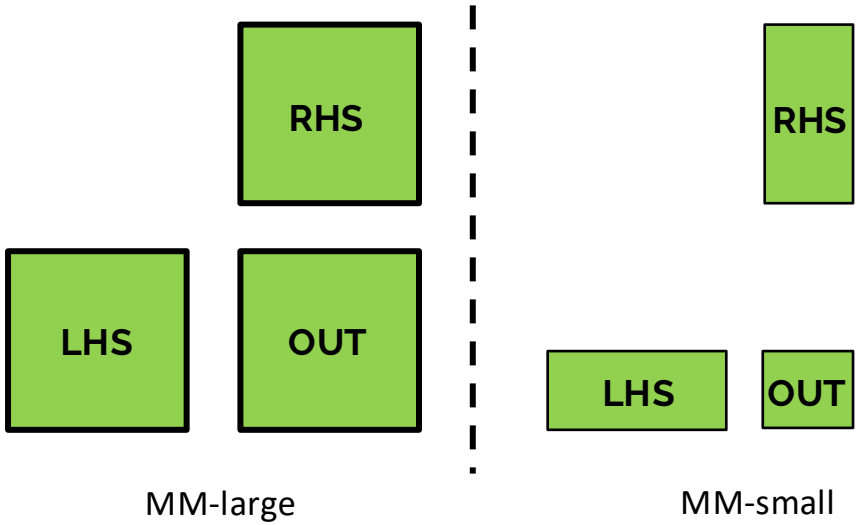
DORA Architecture Design

- Enable flexible computation tile size by parallelism management.



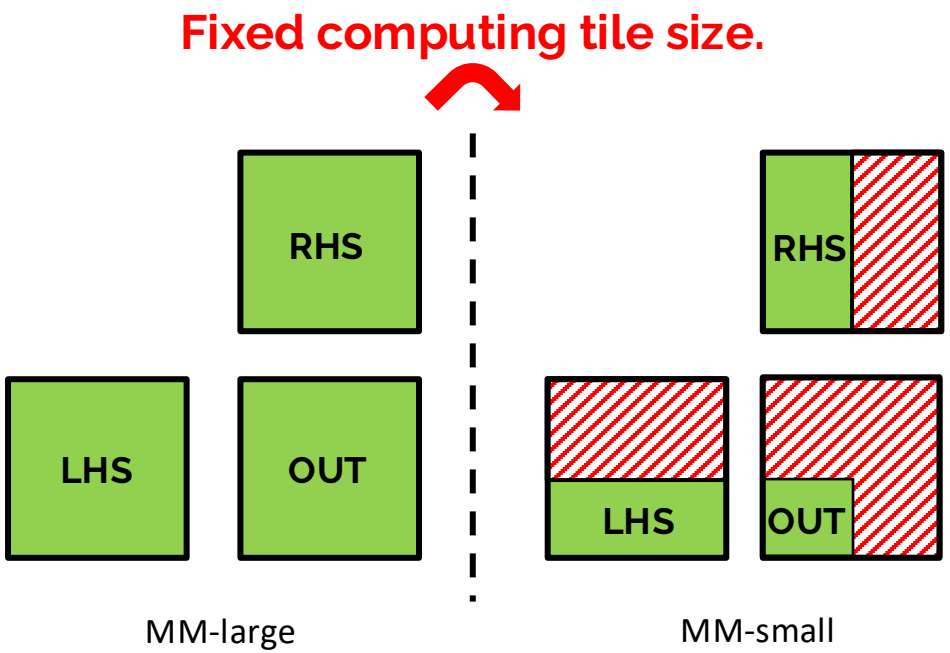
DORA Architecture Design

- Enable flexible computation tile size by parallelism management.



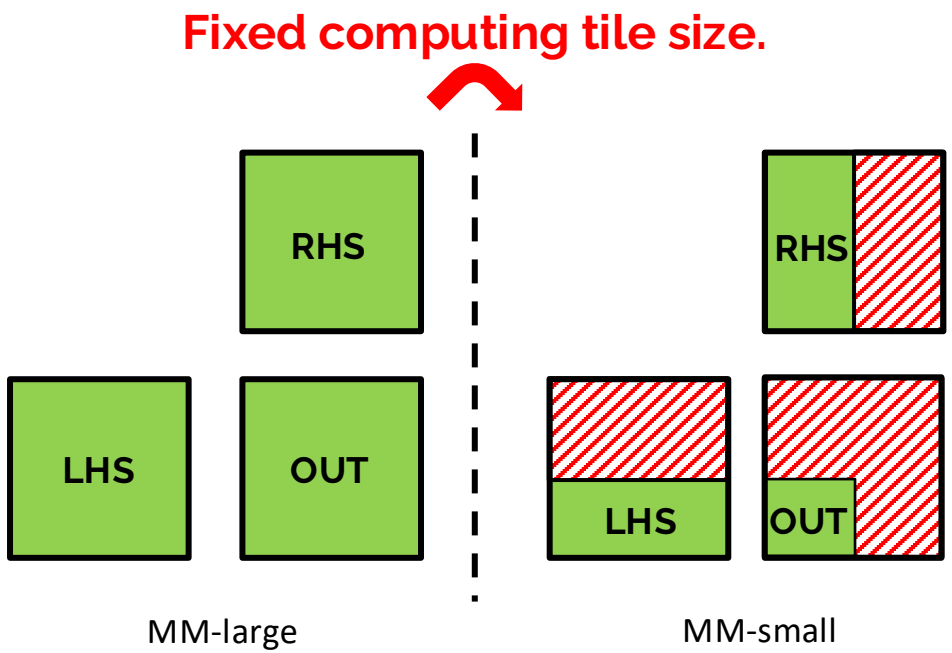
DORA Architecture Design

- Enable flexible computation tile size by parallelism management.



DORA Architecture Design

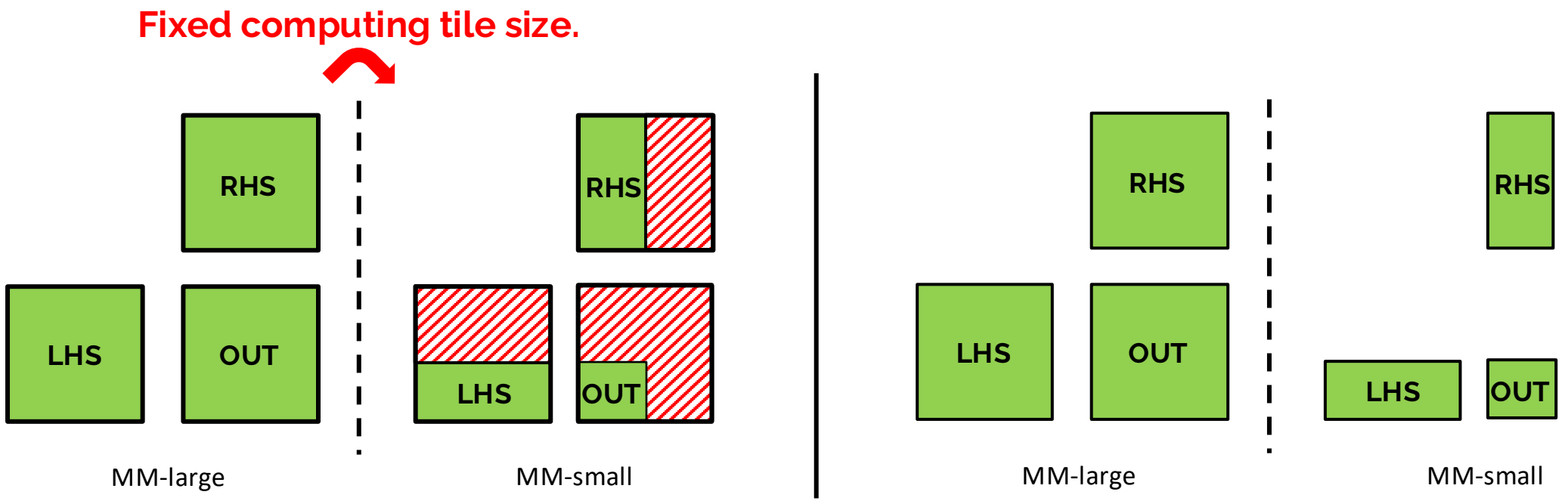
- Enable flexible computation tile size by parallelism management.



Baseline: Induces much computing overhead due to shape mismatch!

DORA Architecture Design

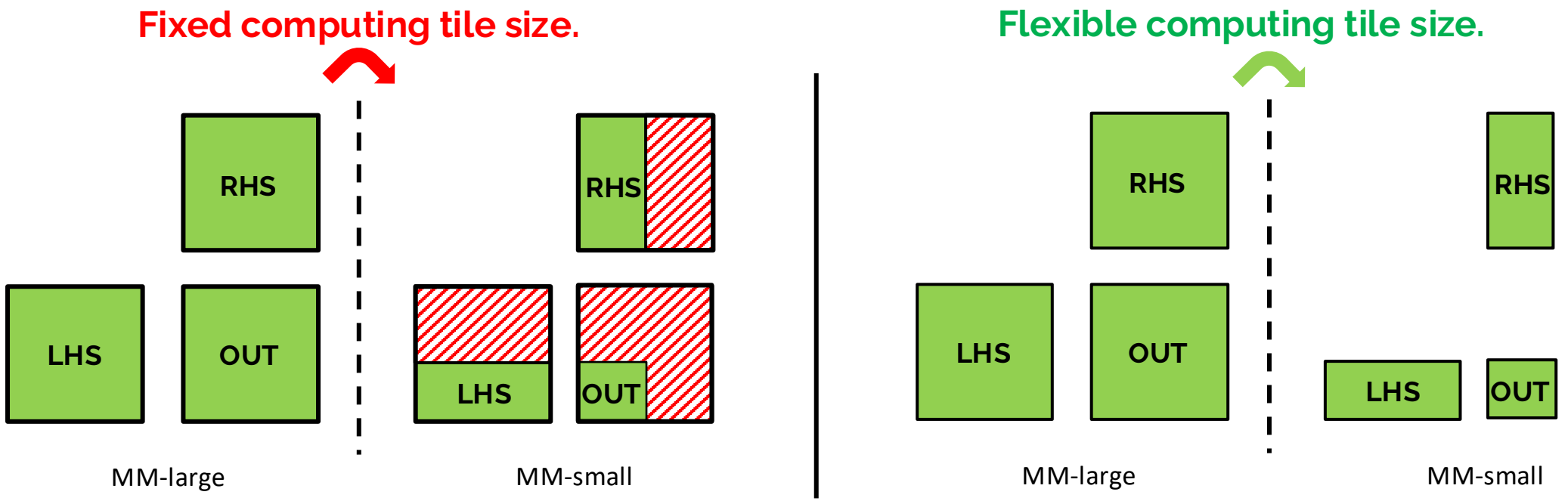
- Enable flexible computation tile size by parallelism management.



Baseline: Induces much computing overhead due to shape mismatch!

DORA Architecture Design

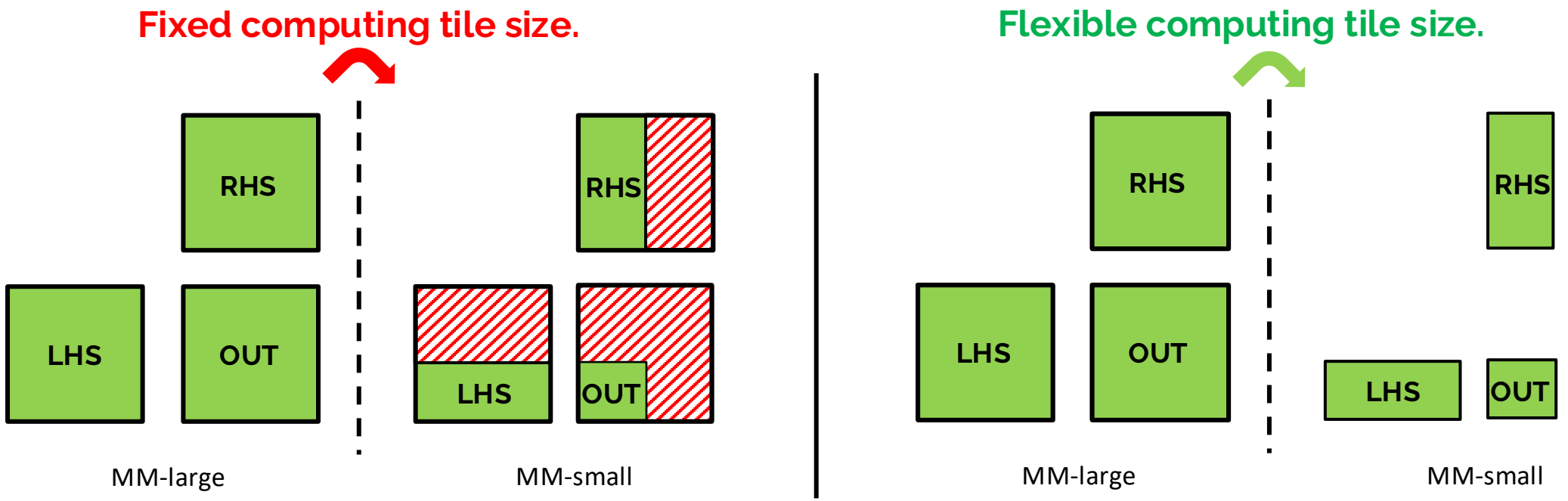
- Enable flexible computation tile size by parallelism management.



Baseline: Induces much computing overhead due to shape mismatch!

DORA Architecture Design

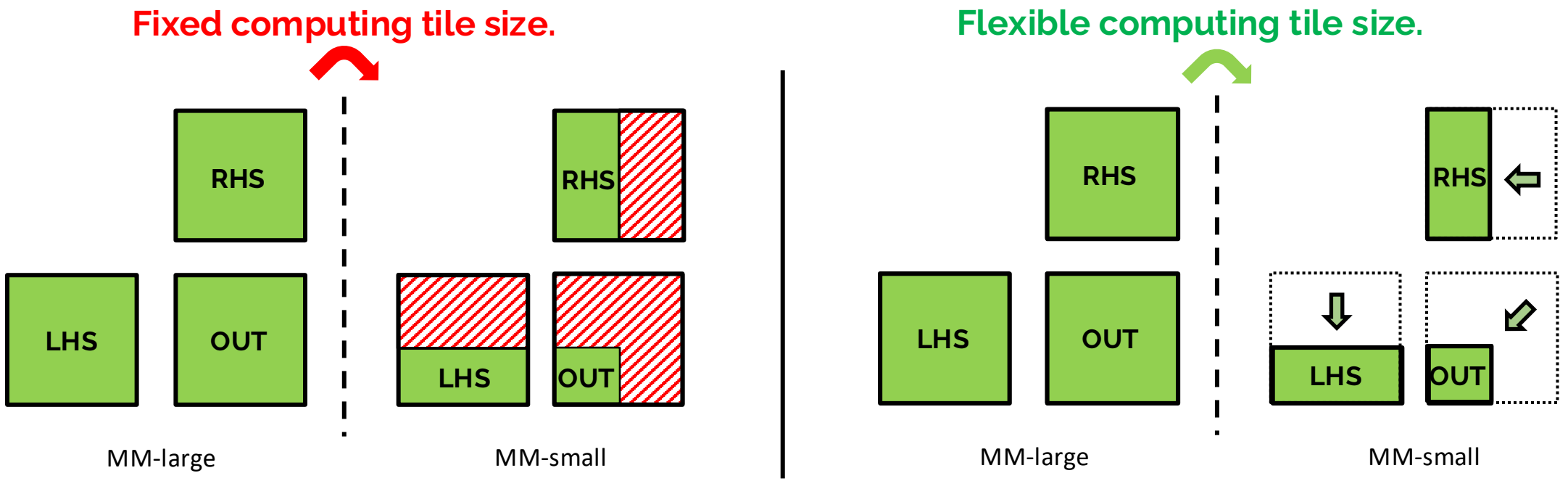
- Enable flexible computation tile size by parallelism management.



Baseline: Induces much computing overhead due to shape mismatch!

DORA Architecture Design

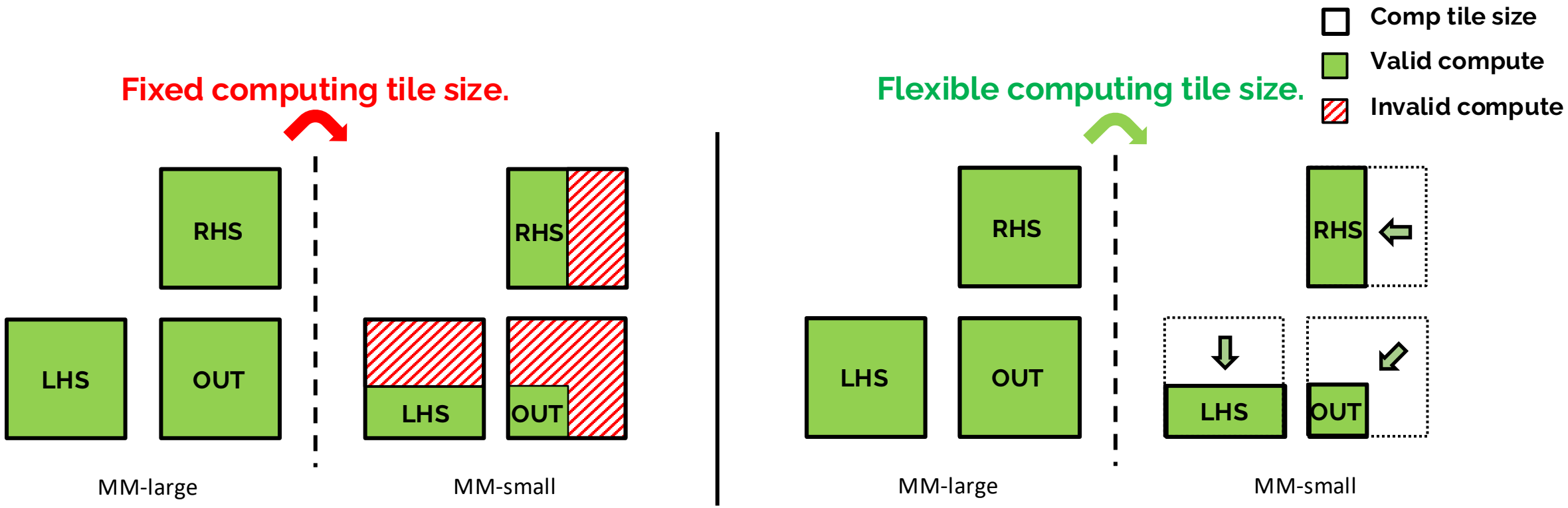
- Enable flexible computation tile size by parallelism management.



Baseline: Induces much computing overhead due to shape mismatch!

DORA Architecture Design

- Enable flexible computation tile size by parallelism management.



Baseline: Induces much computing overhead due to shape mismatch!

DORA: Flexibly adjust computing tile size to accommodate workload.

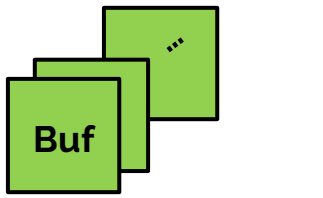
DORA Architecture Design

DORA Architecture Design

- Enable flexible memory view by on-chip buffer management.

DORA Architecture Design

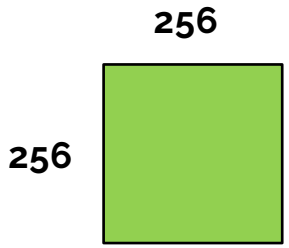
- Enable flexible memory view by on-chip buffer management.



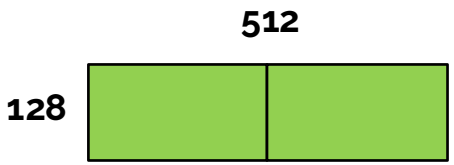
Fixed on-chip capacity

DORA Architecture Design

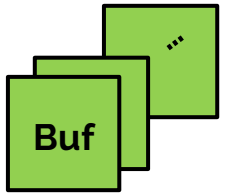
- Enable flexible memory view by on-chip buffer management.



Square matrix



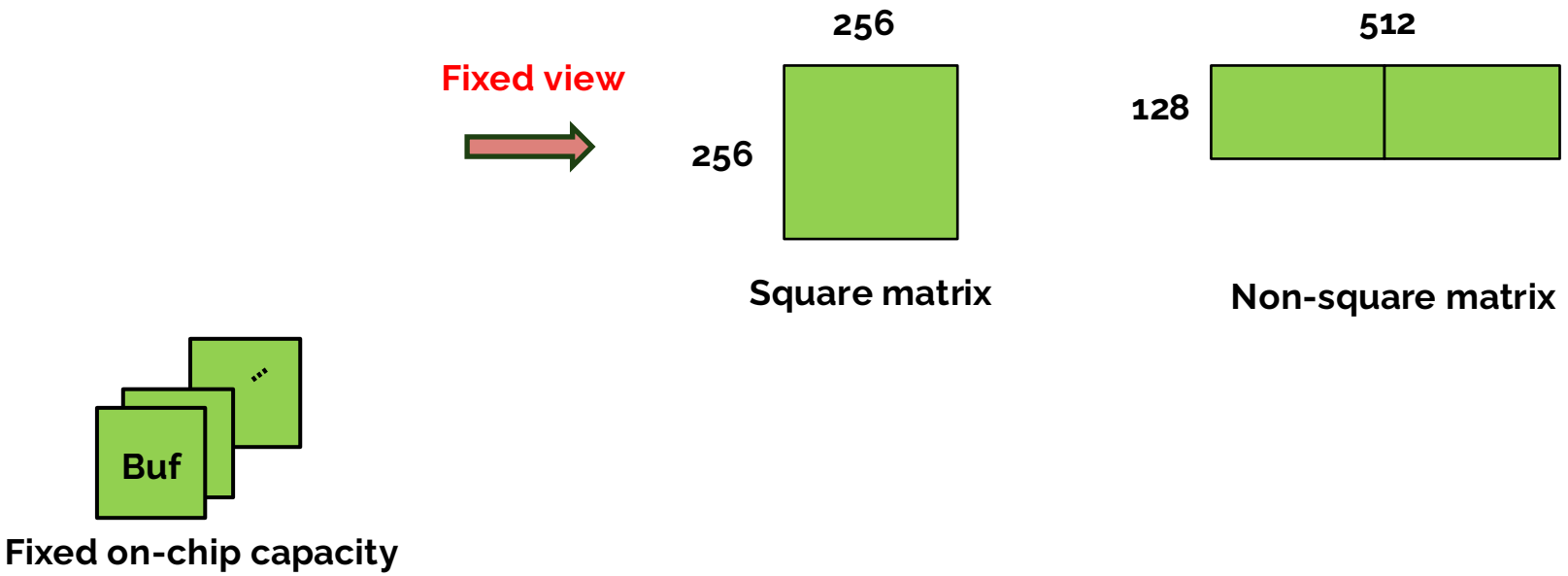
Non-square matrix



Fixed on-chip capacity

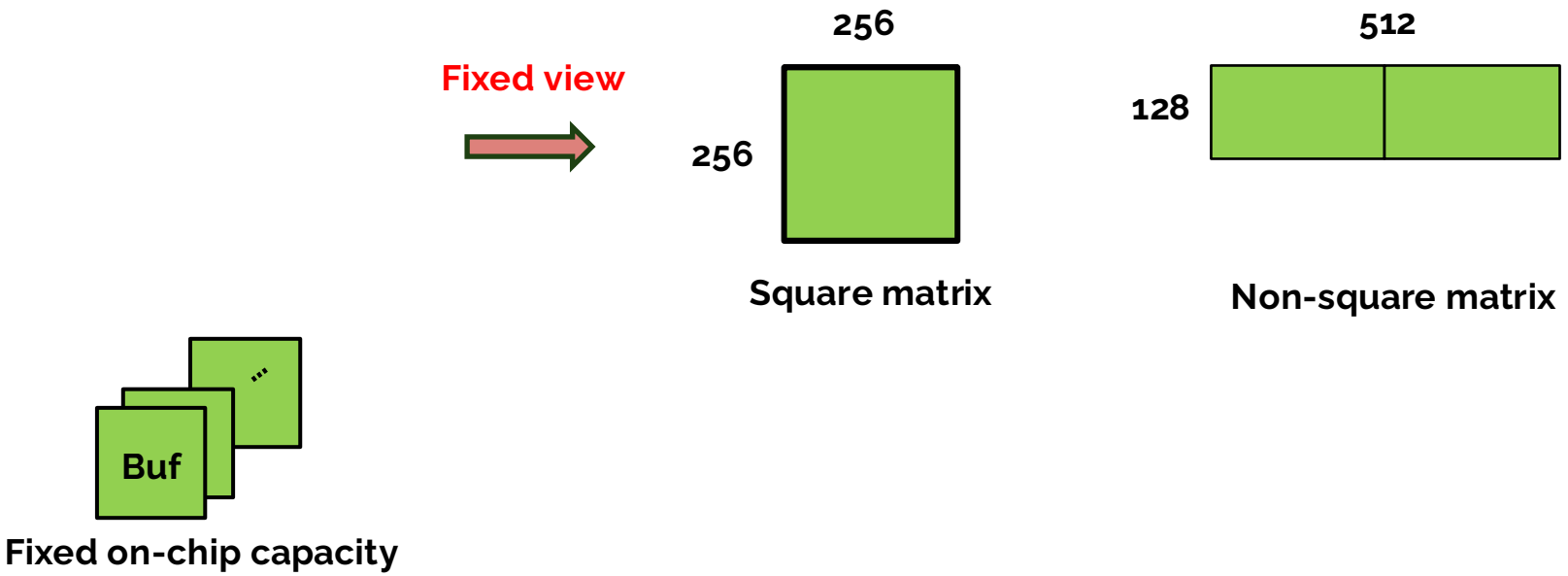
DORA Architecture Design

- Enable flexible memory view by on-chip buffer management.



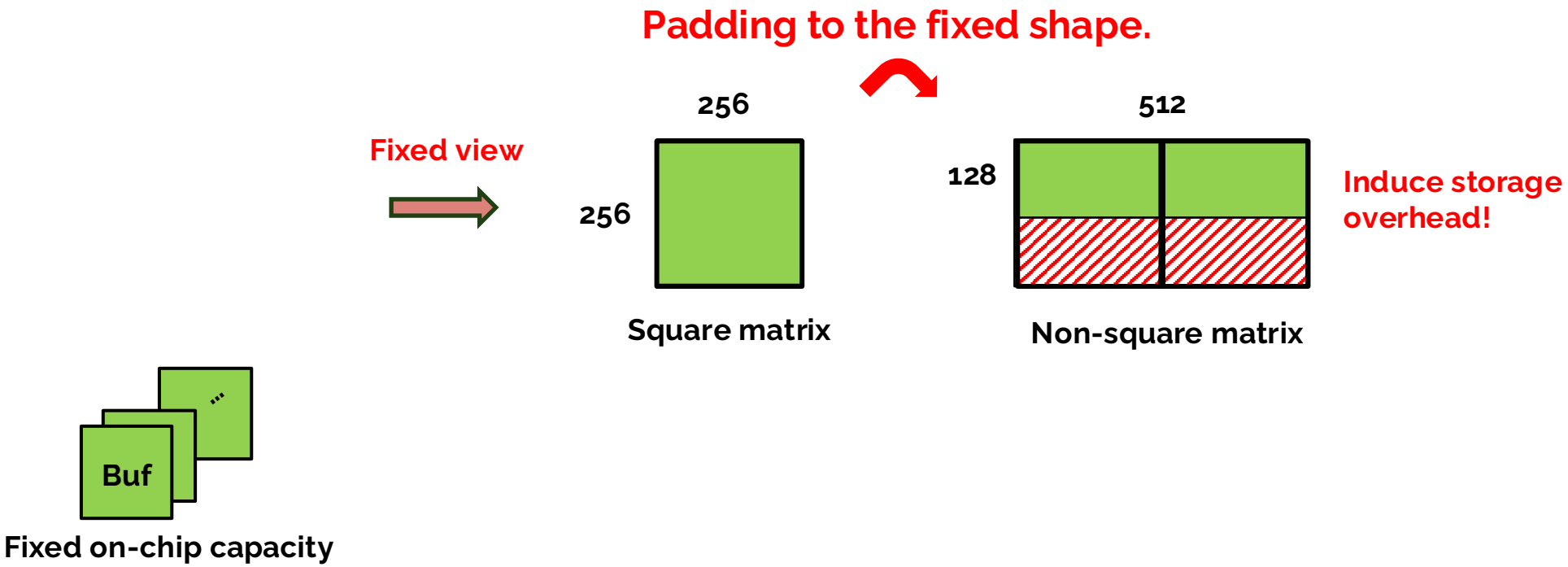
DORA Architecture Design

- Enable flexible memory view by on-chip buffer management.



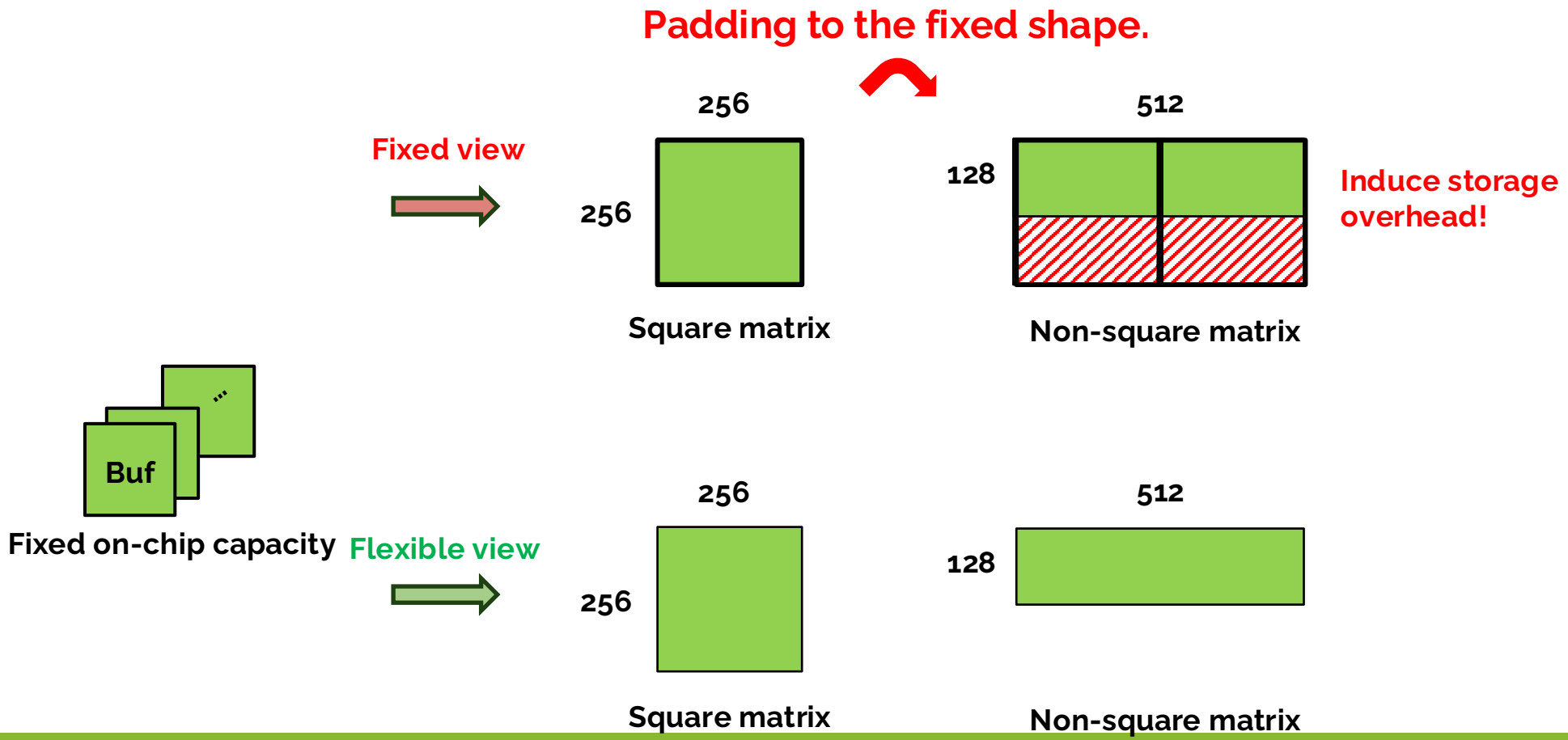
DORA Architecture Design

- Enable flexible memory view by on-chip buffer management.



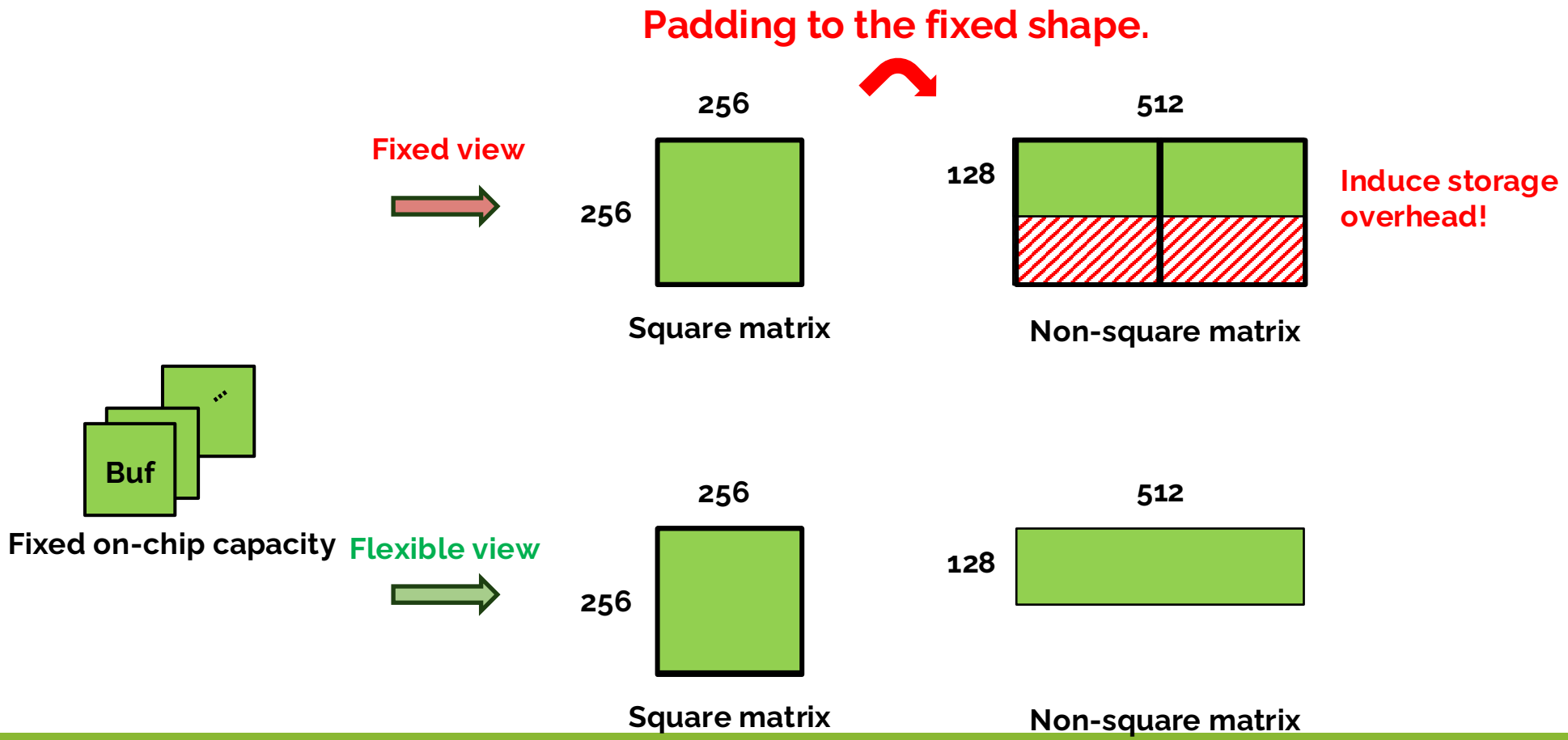
DORA Architecture Design

- Enable flexible memory view by on-chip buffer management.



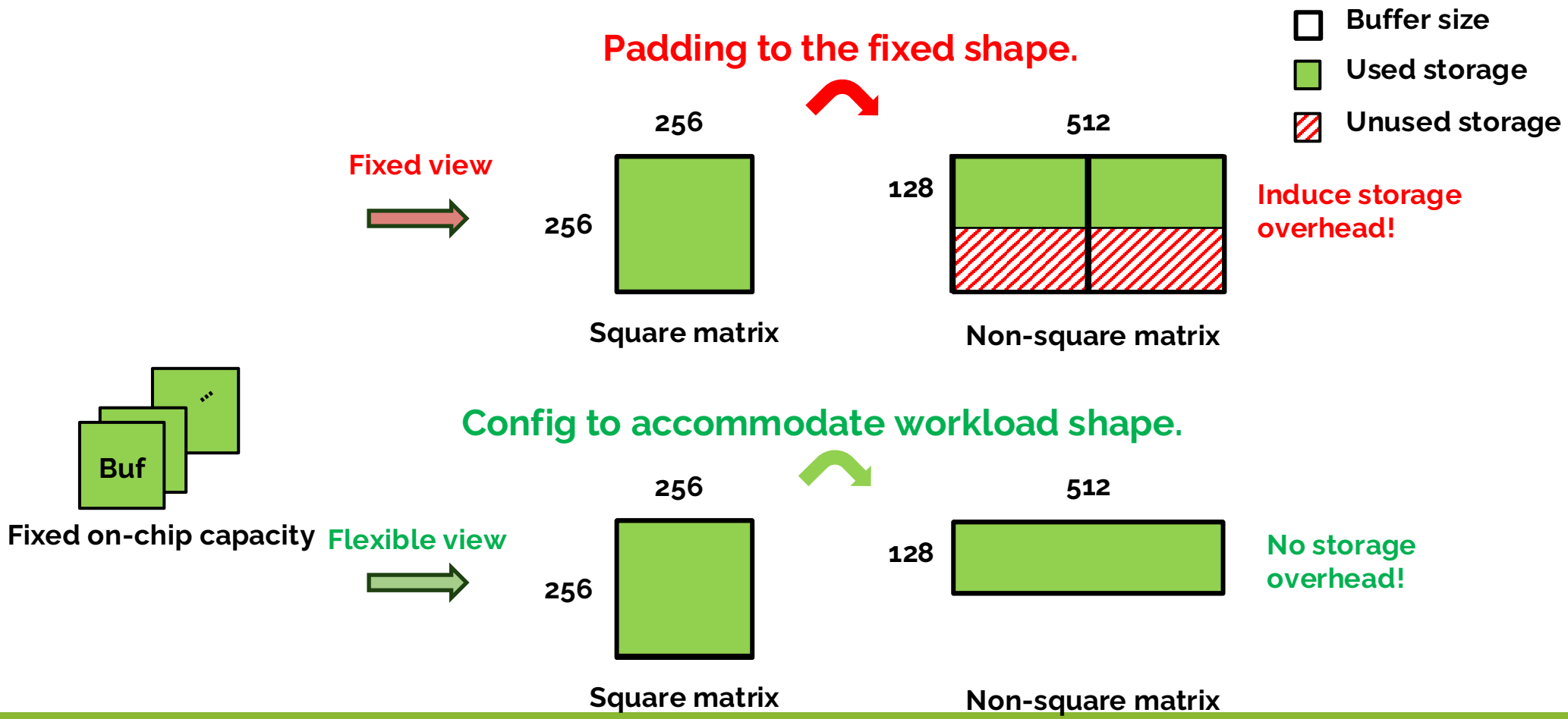
DORA Architecture Design

- Enable flexible memory view by on-chip buffer management.



DORA Architecture Design

- Enable flexible memory view by on-chip buffer management.



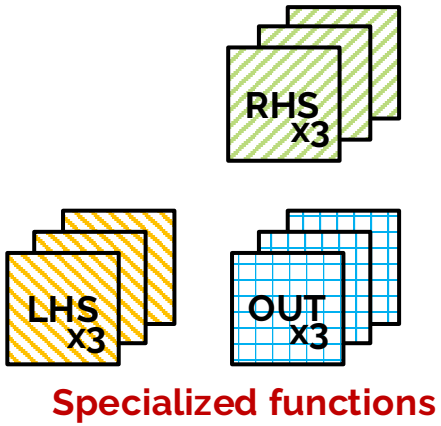
DORA Architecture Design

DORA Architecture Design

- Enable flexible memory functionalities by topology management.

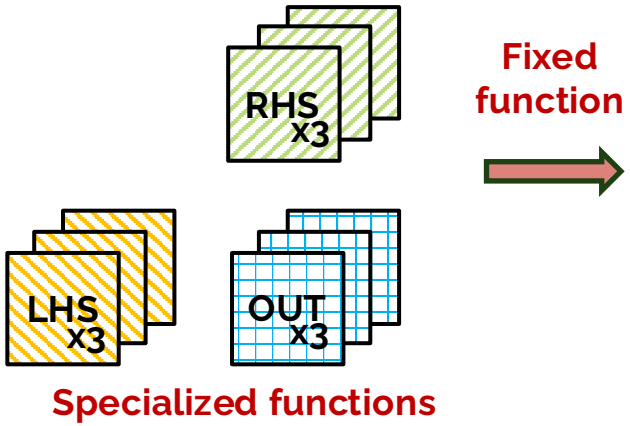
DORA Architecture Design

- Enable flexible memory functionalities by topology management.



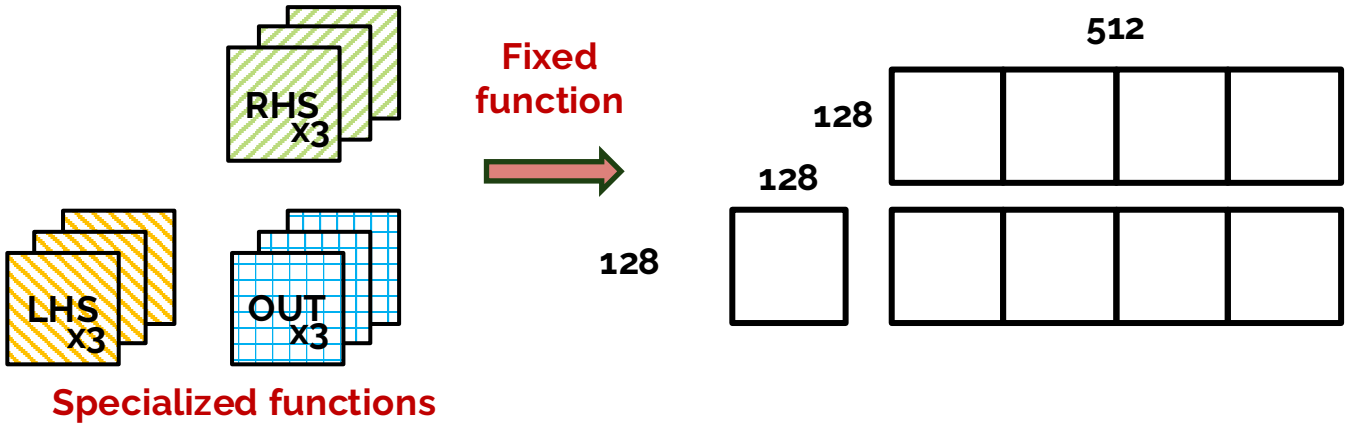
DORA Architecture Design

- Enable flexible memory functionalities by topology management.



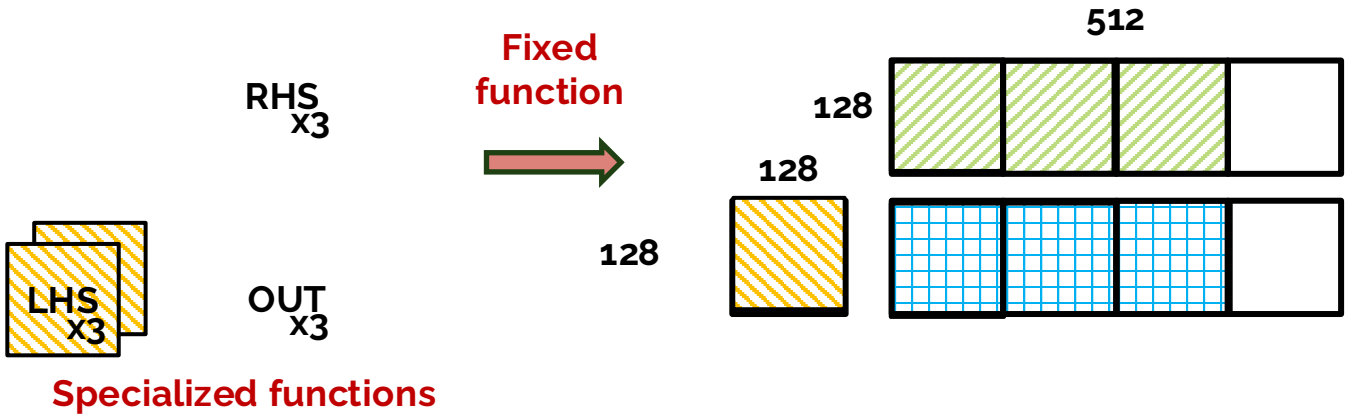
DORA Architecture Design

- Enable flexible memory functionalities by topology management.



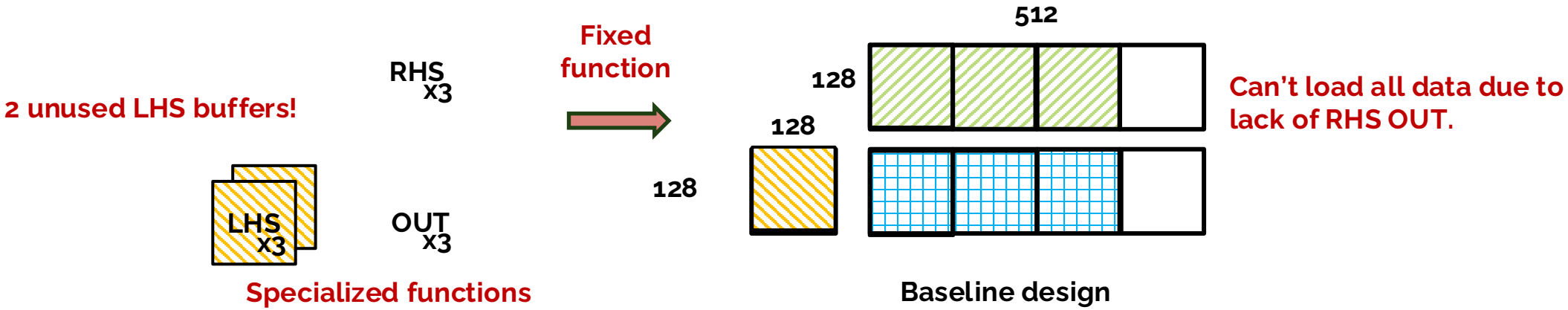
DORA Architecture Design

- Enable flexible memory functionalities by topology management.



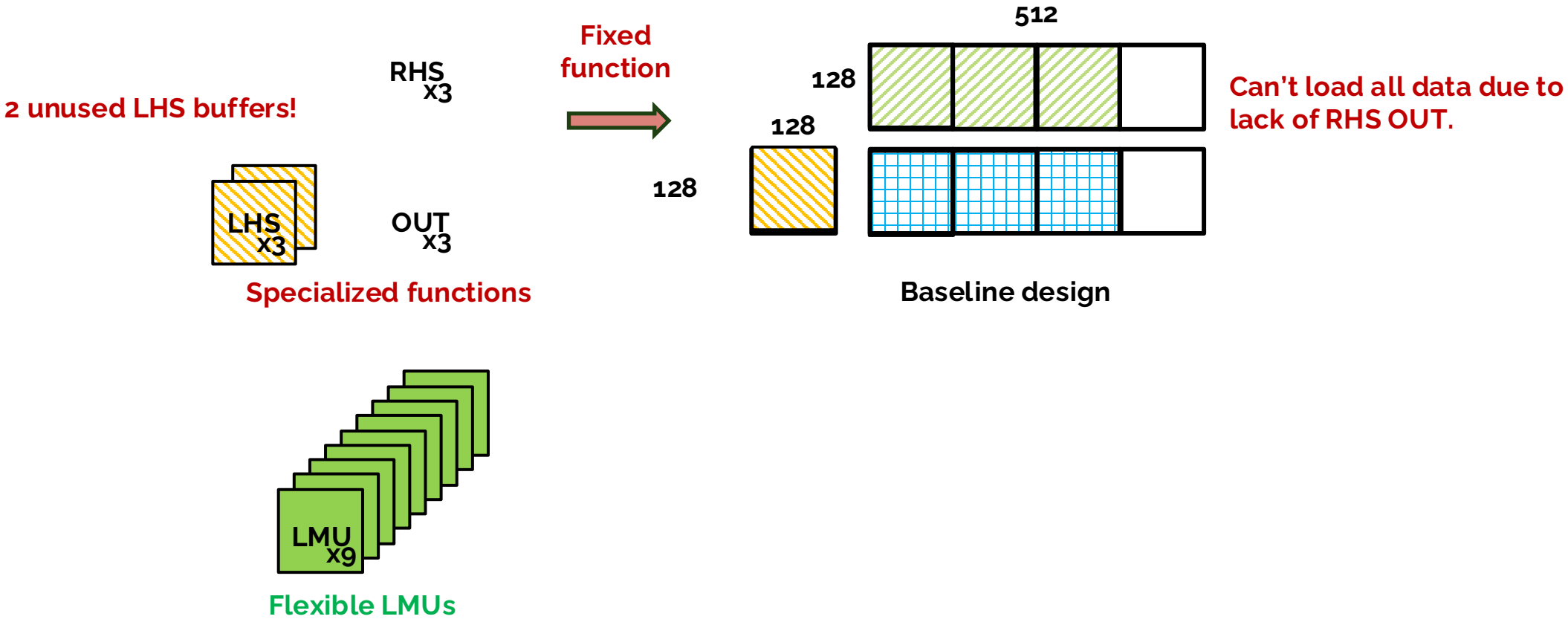
DORA Architecture Design

- Enable flexible memory functionalities by topology management.



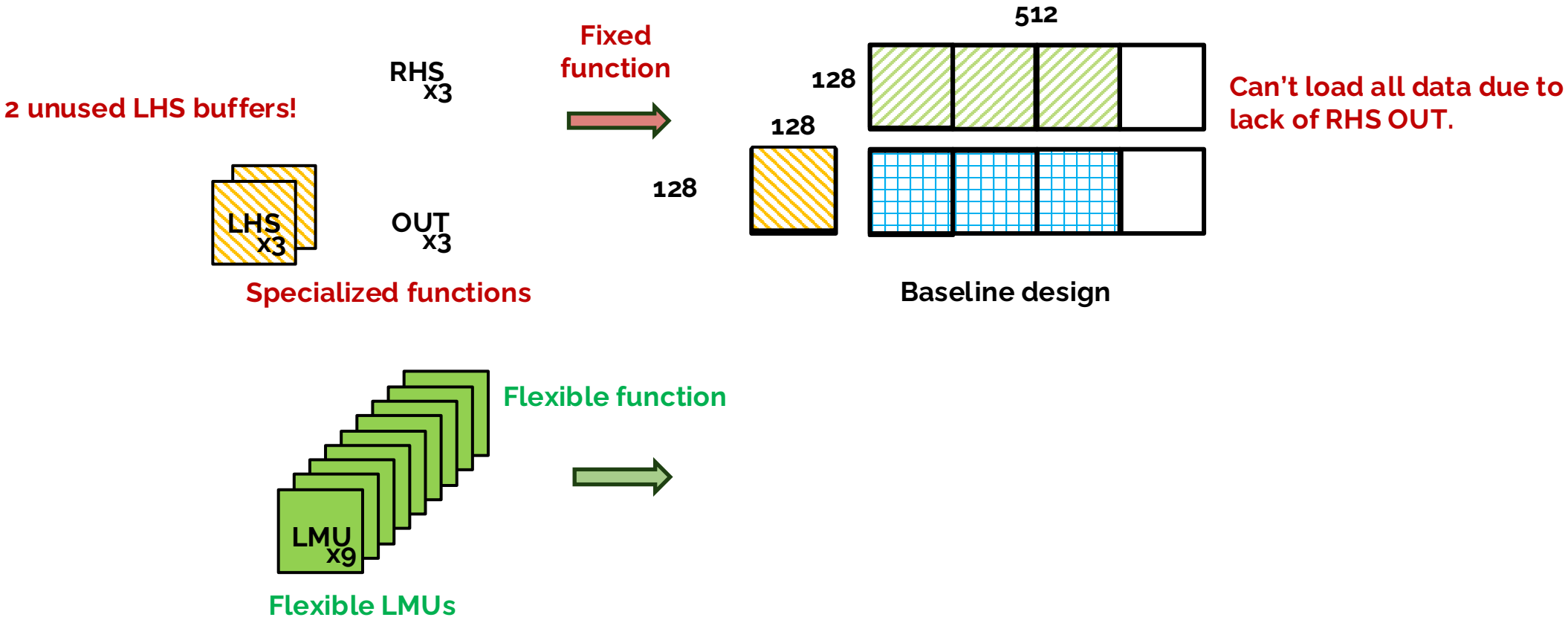
DORA Architecture Design

- Enable flexible memory functionalities by topology management.



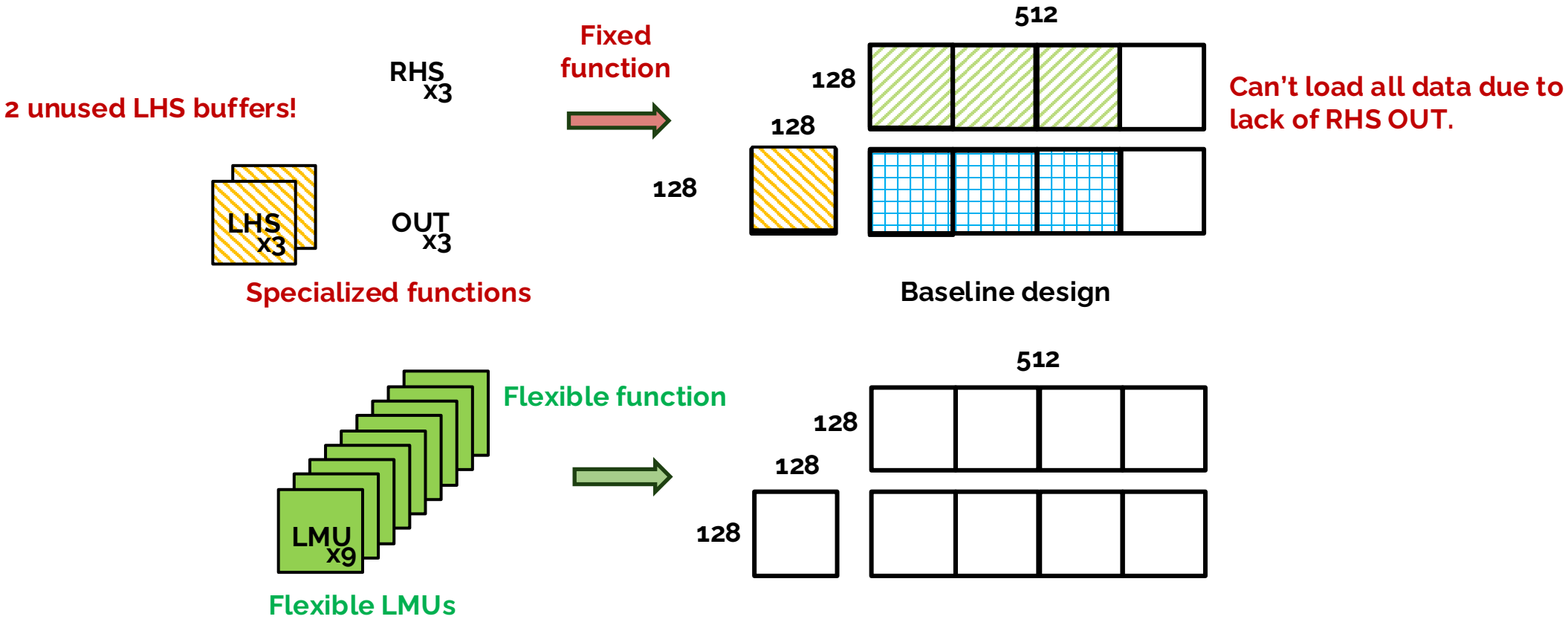
DORA Architecture Design

- Enable flexible memory functionalities by topology management.



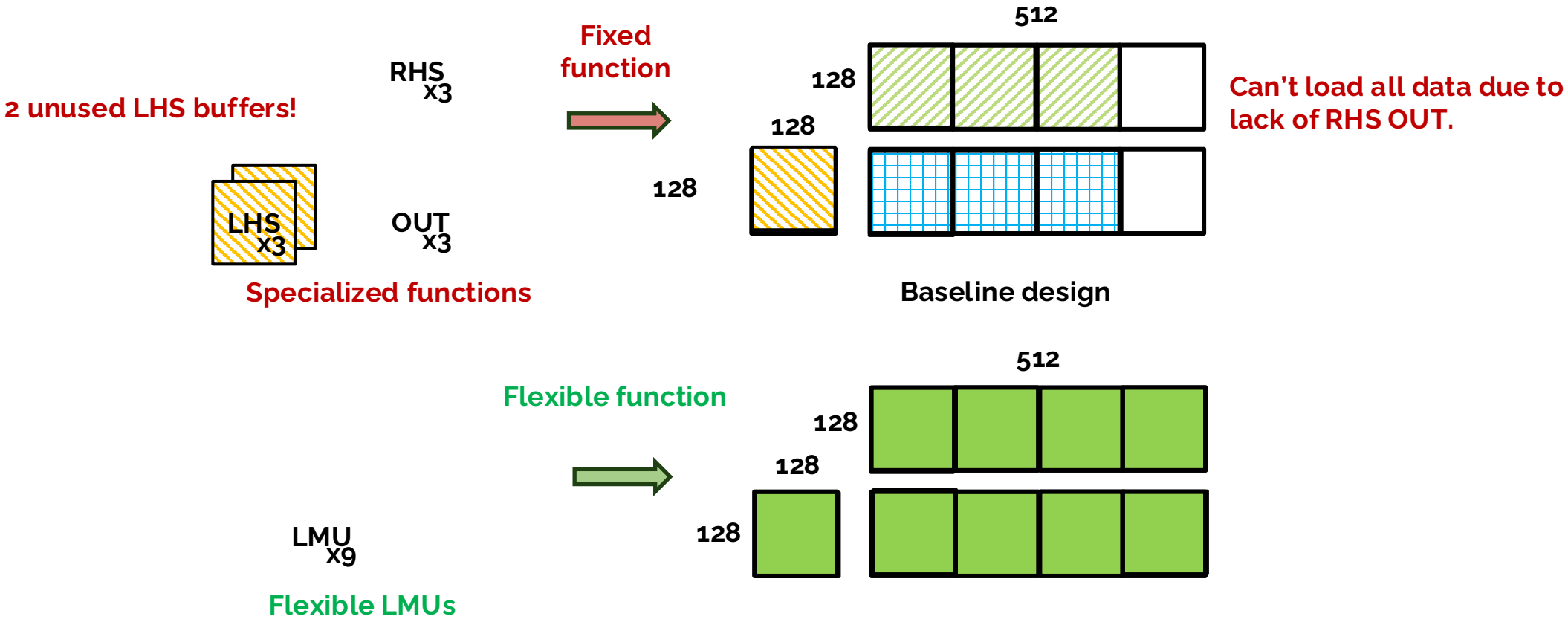
DORA Architecture Design

- Enable flexible memory functionalities by topology management.



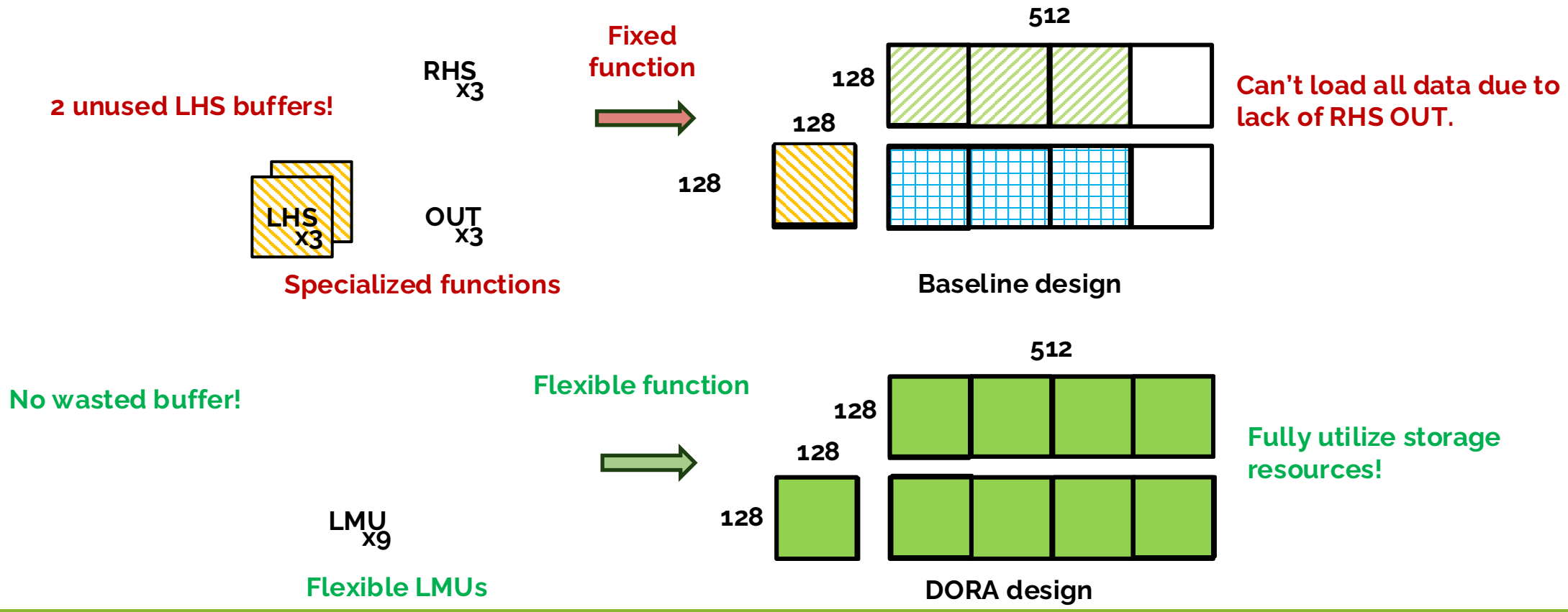
DORA Architecture Design

- Enable flexible memory functionalities by topology management.



DORA Architecture Design

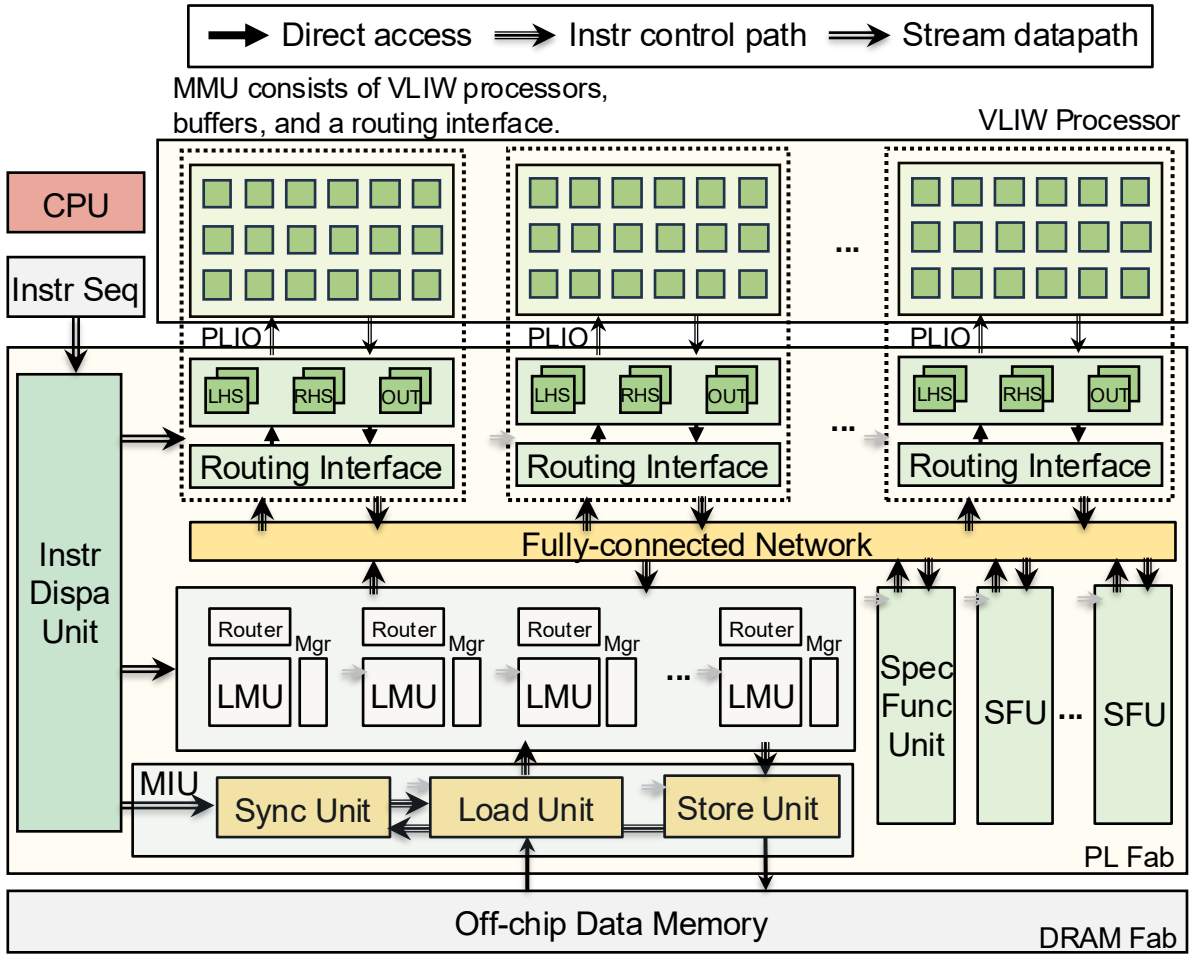
- Enable flexible memory functionalities by topology management.



DORA Architecture Design

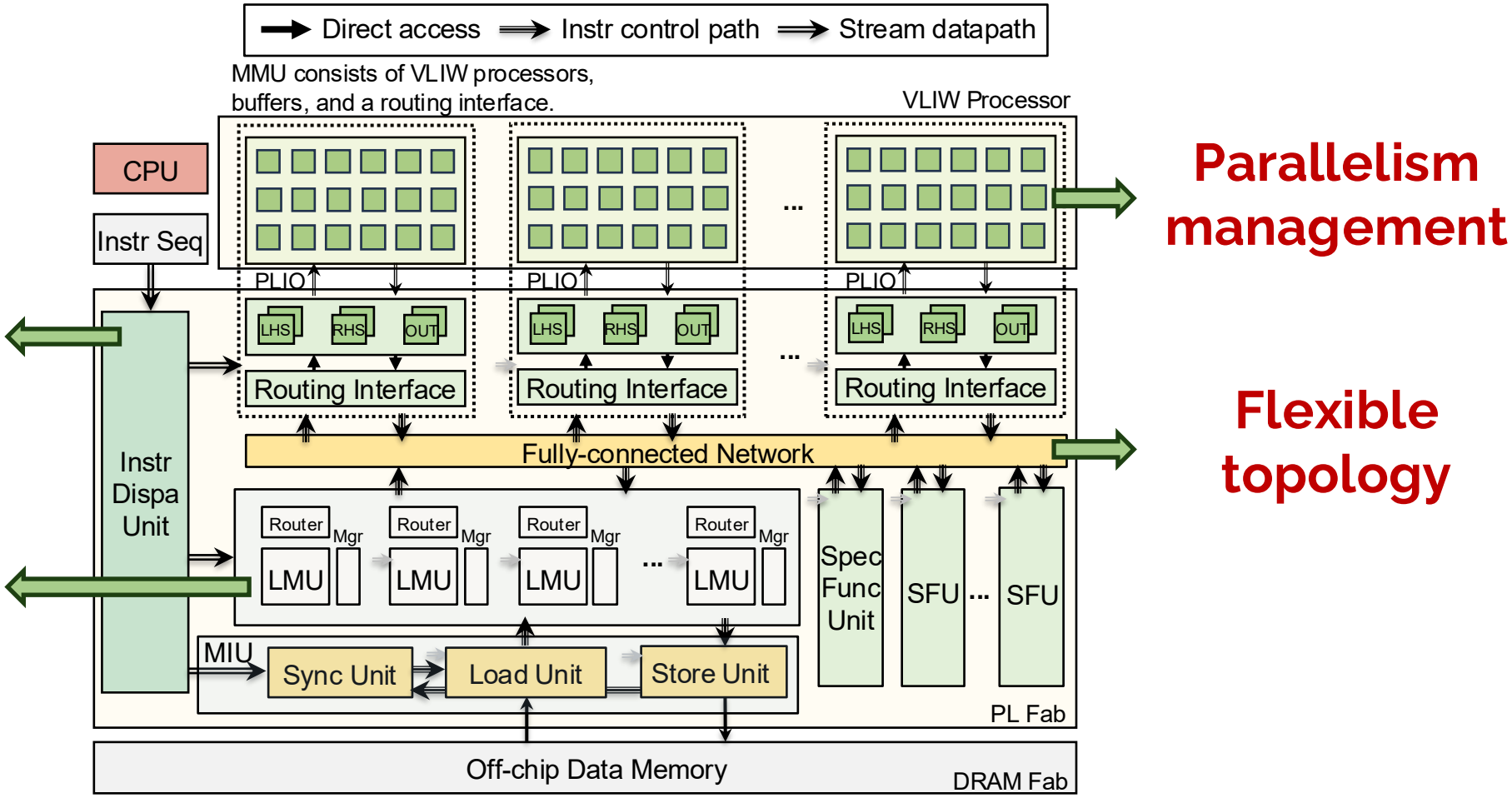
DORA Architecture Design

- DORA overlay can orchestrate **dataflow** with **instructions** to enable flexible execution patterns for **diverse workloads**.



DORA Architecture Design

- DORA overlay can orchestrate **dataflow** with **instructions** to enable flexible execution patterns for **diverse workloads**.



Instruction control

Flexible memory management

Parallelism management

Flexible topology

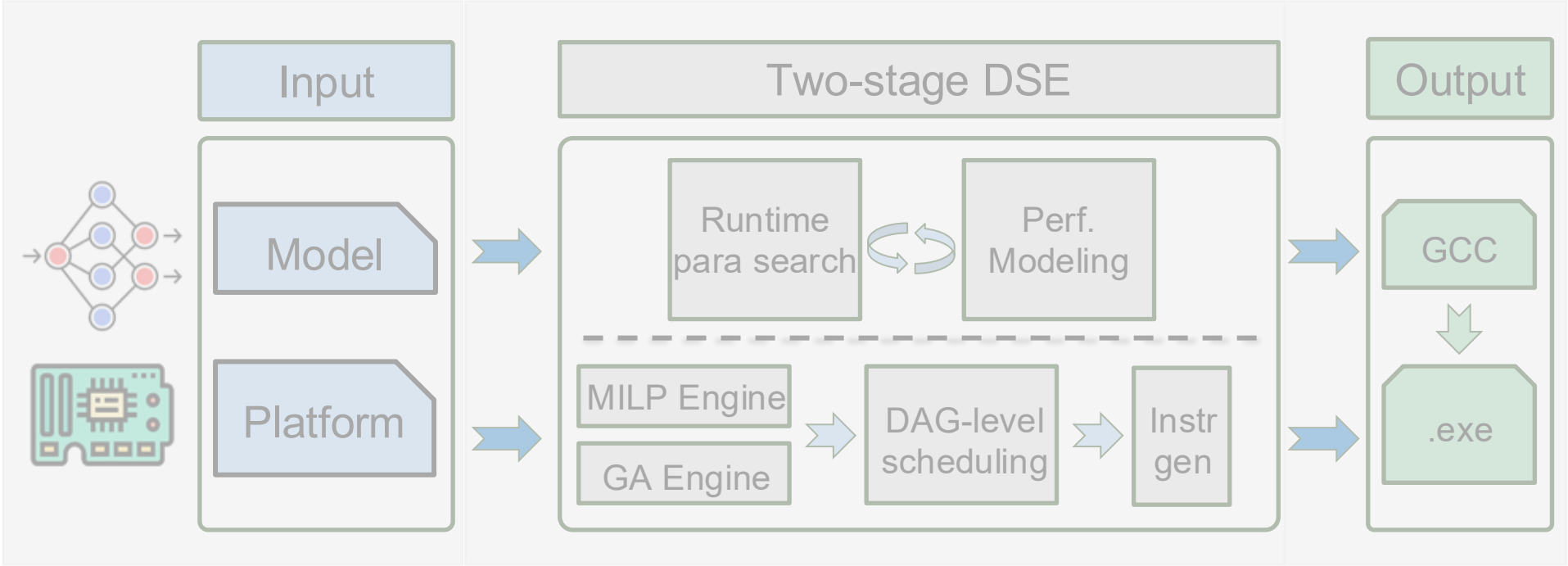
DORA Architecture Design

- DORA overlay can orchestrate dataflow with instructions to enable flexible execution patterns for diverse workloads.

How to efficiently program DORA overlay for software developers?

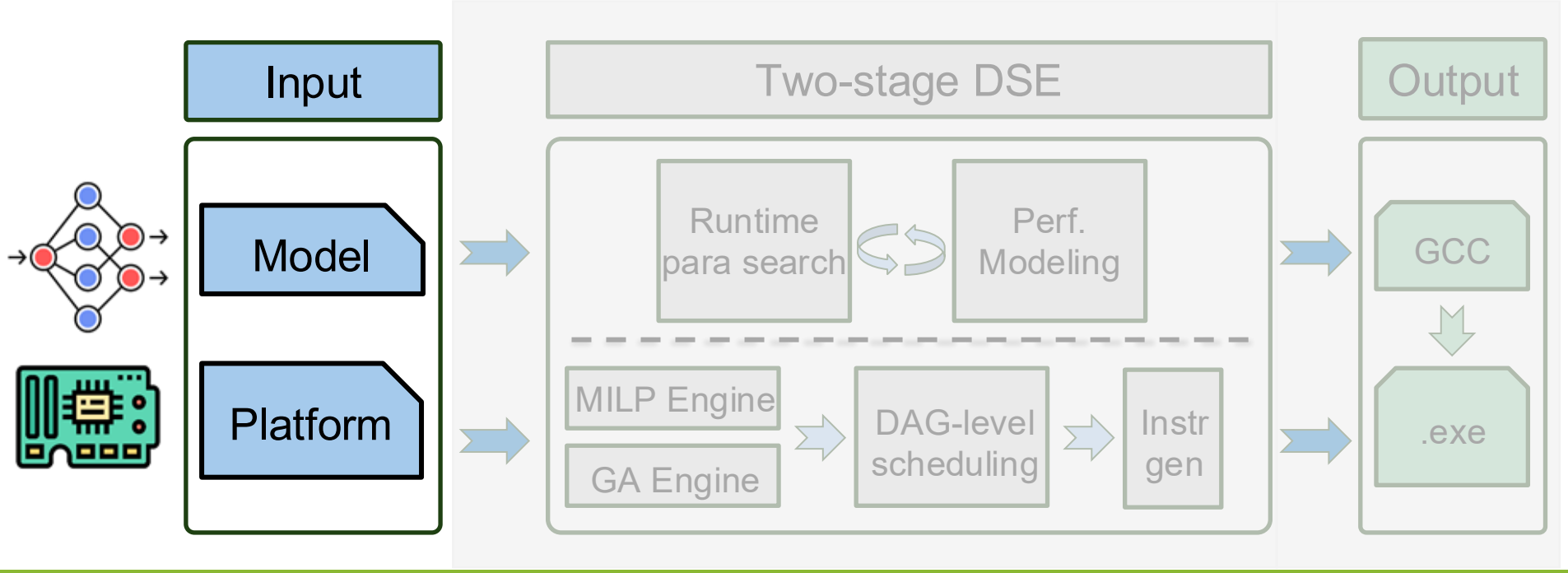
DORA Framework

DORA Framework



DORA Framework

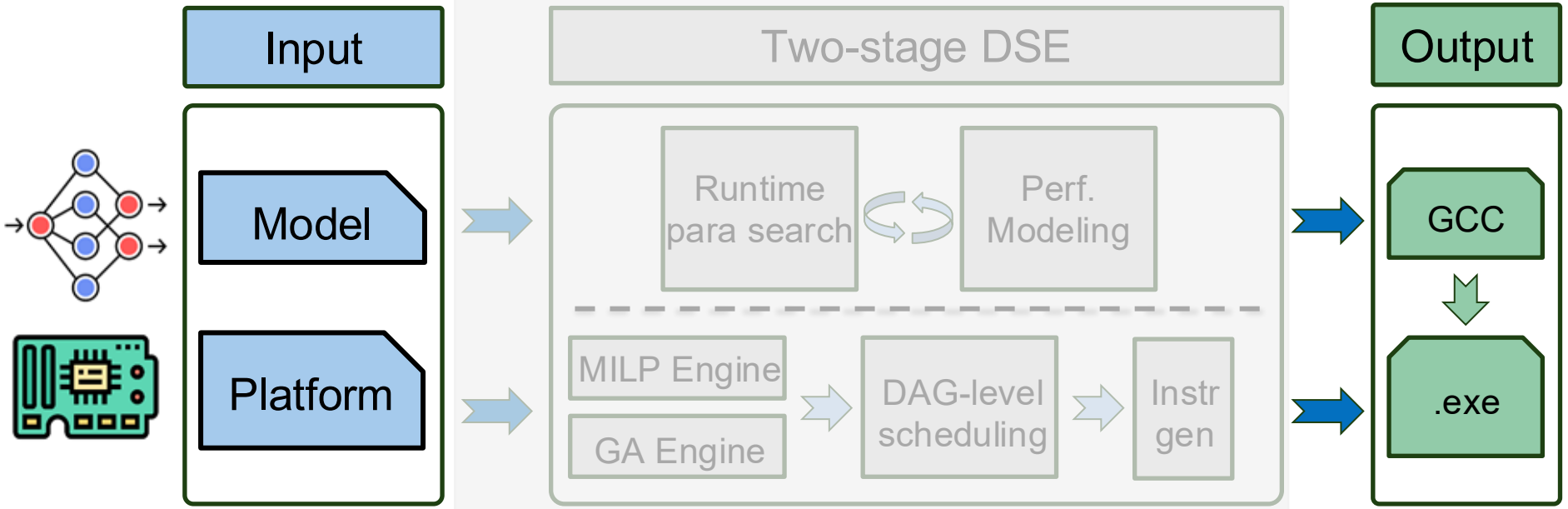
Input:
1) MM-based AI Model;
2) Hardware platforms information.



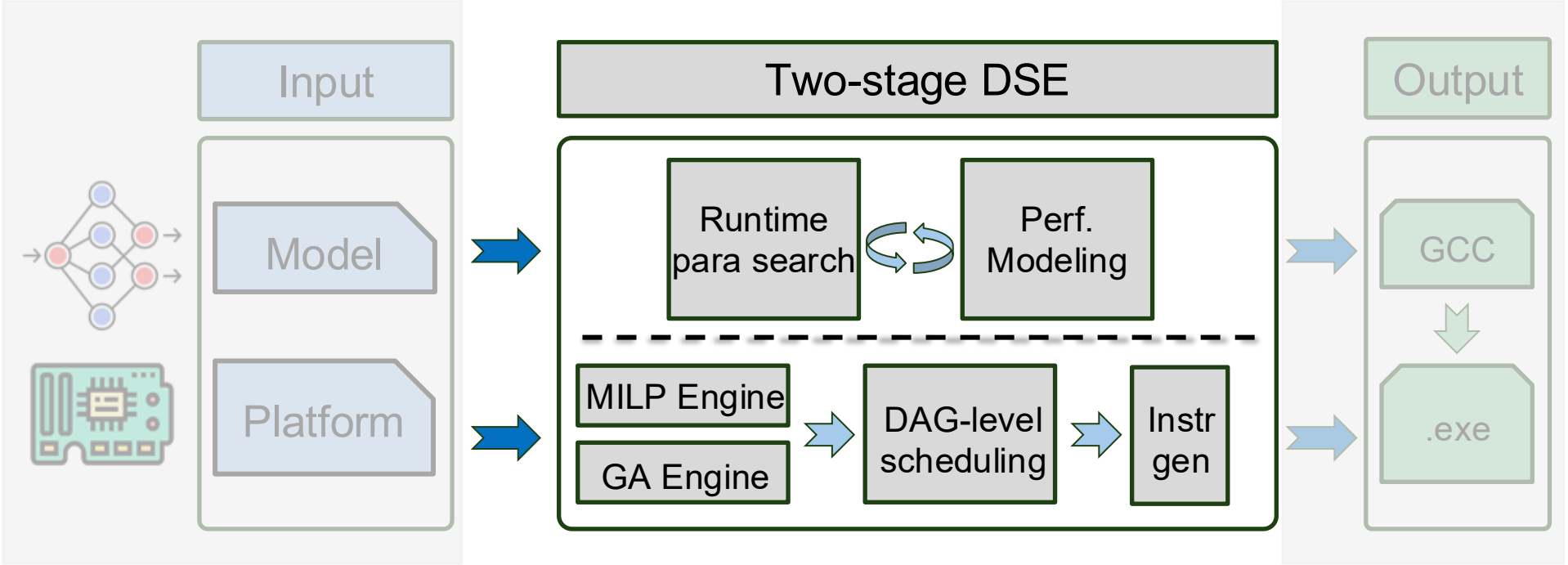
DORA Framework

Input:
1) MM-based AI Model;
2) Hardware platforms information.

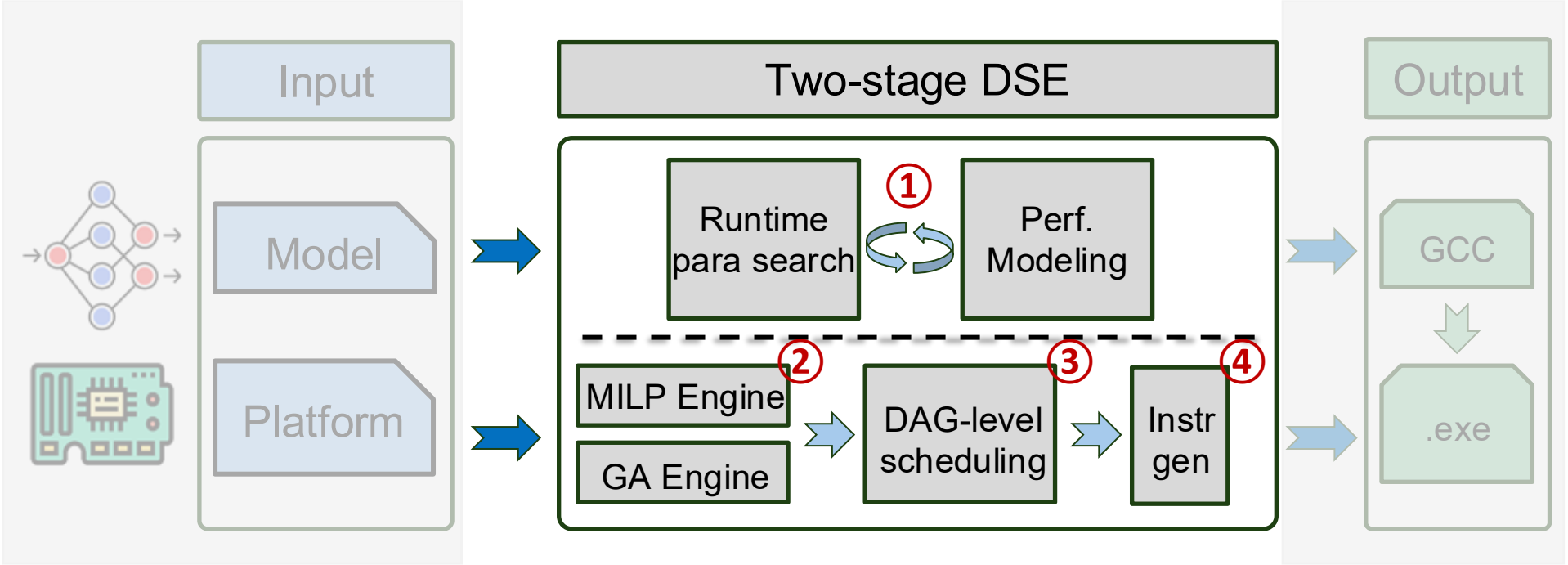
Output:
1) Bitstream running on the AIE and PL;
2) Executable file running on ARM CPU.



DORA Framework

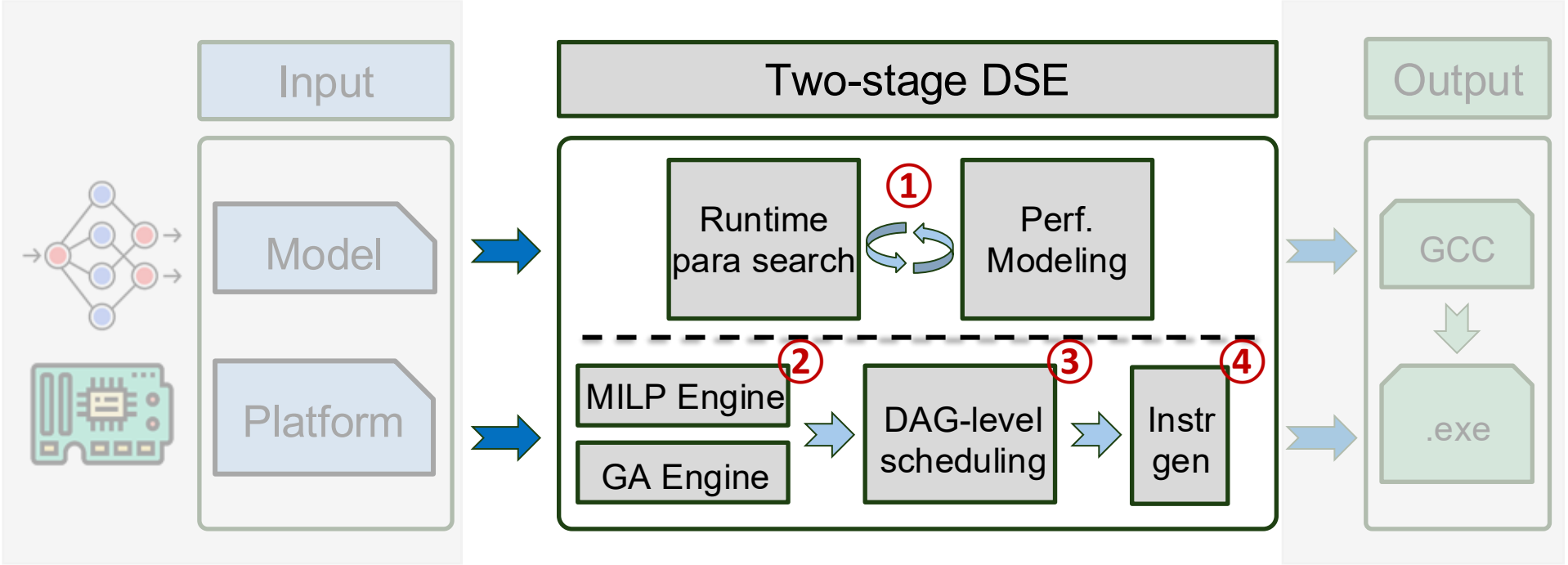


DORA Framework



DORA Framework

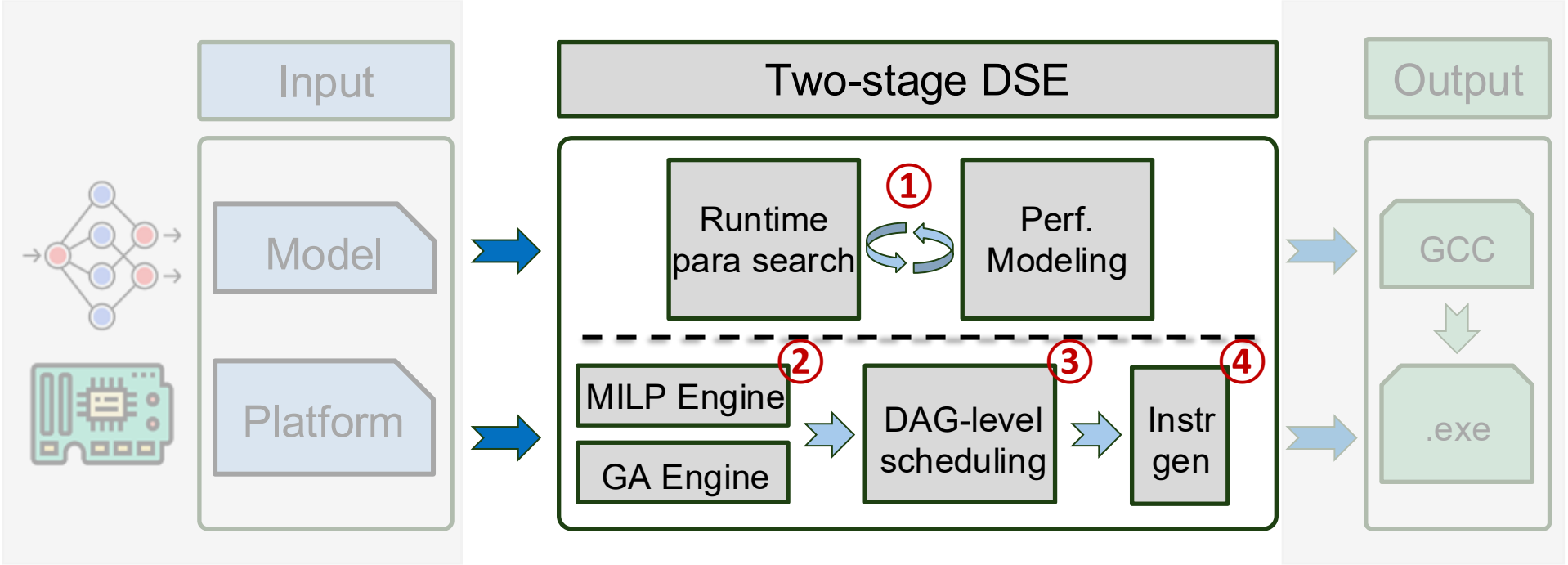
①Single-layer brute-force design space exploration.



DORA Framework

①Single-layer brute-force design space exploration.

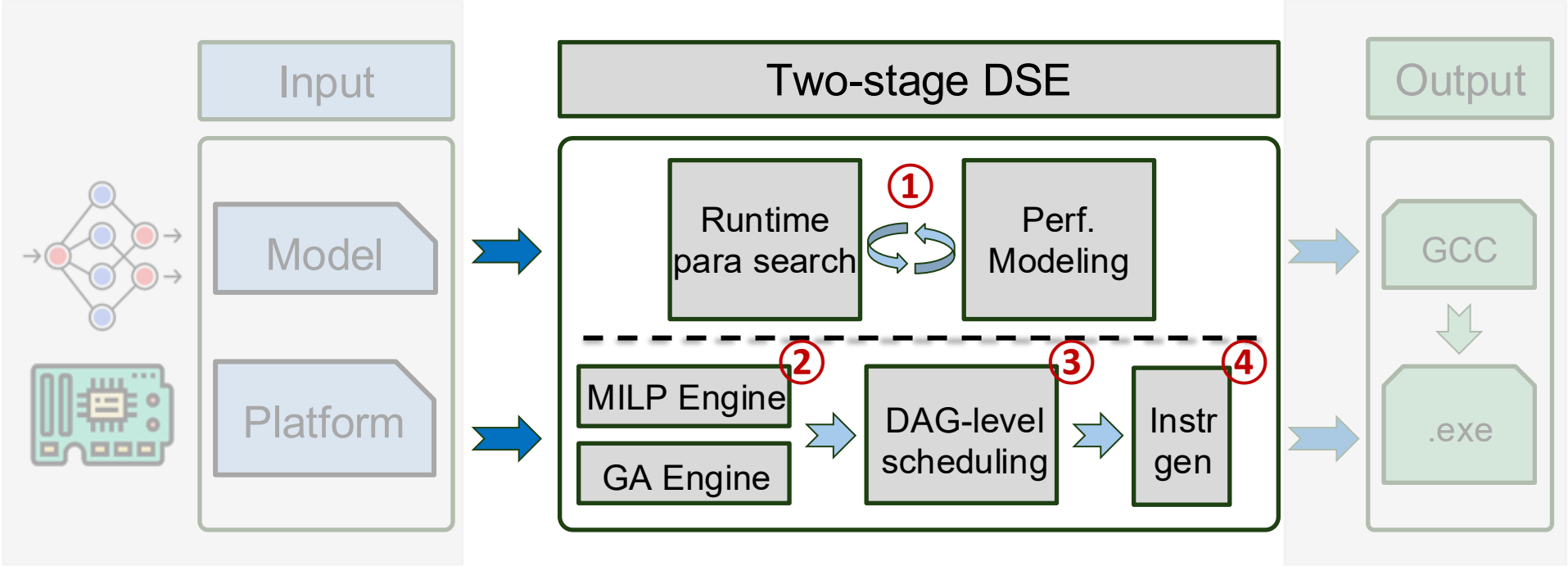
- **Populate all possible mapping solutions with #memory and #compute resources for a single layer.**



DORA Framework

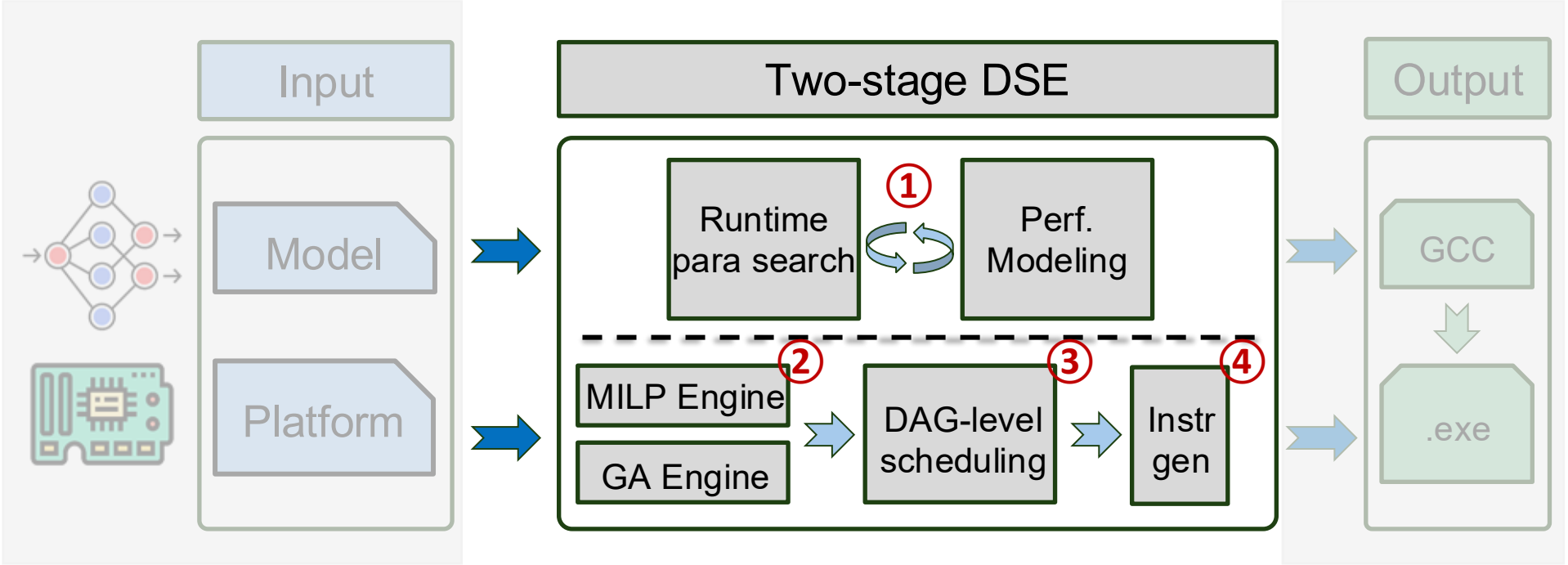
①Single-layer brute-force design space exploration.

- **Populate all possible mapping solutions with #memory and #compute resources for a single layer.**
- **Store the candidate execution table for the second stage.**



DORA Framework

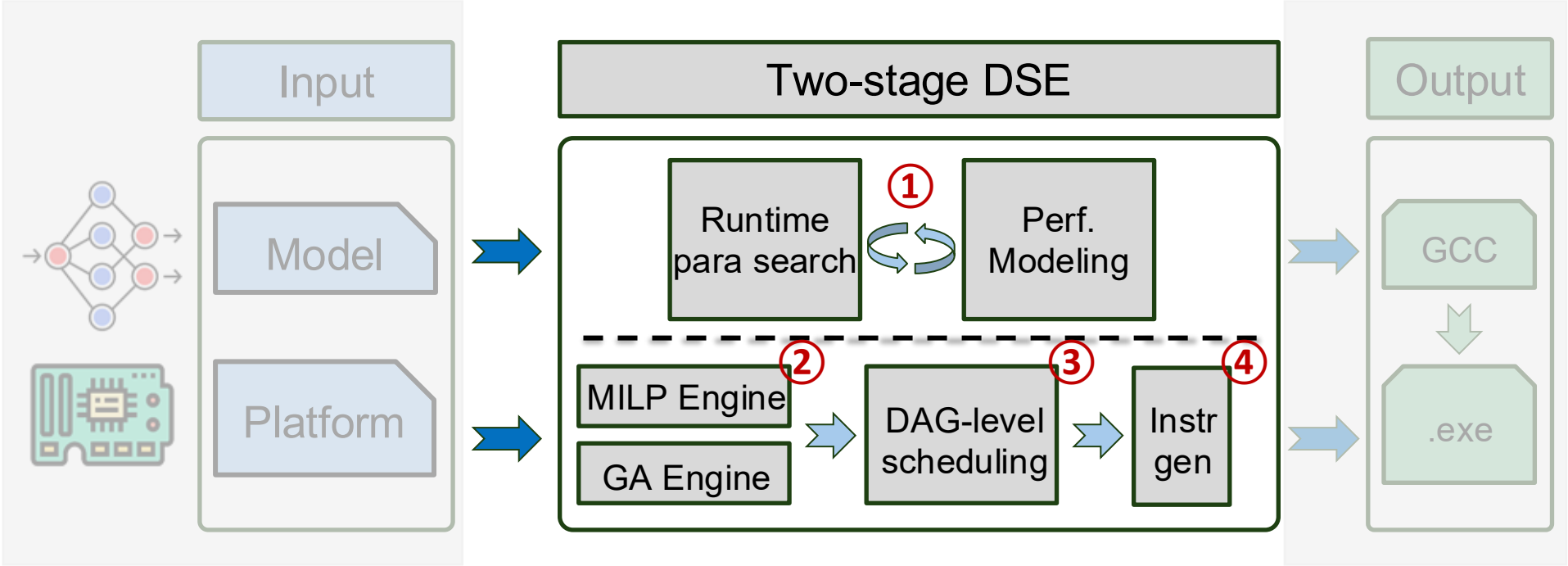
②Scheduling engines at the graph level for whole workloads.



DORA Framework

②Scheduling engines at the graph level for whole workloads.

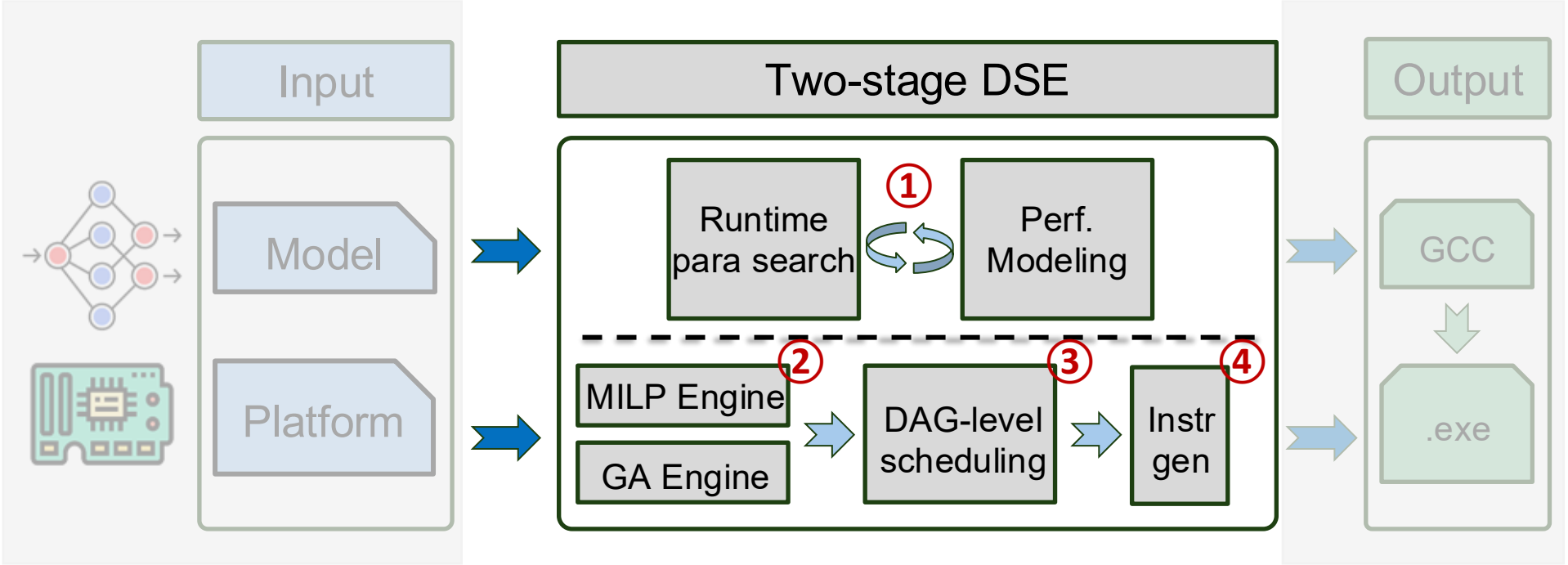
- **Mixed Integer Linear Programming (MILP) engine can guarantee the optimal scheduling.**



DORA Framework

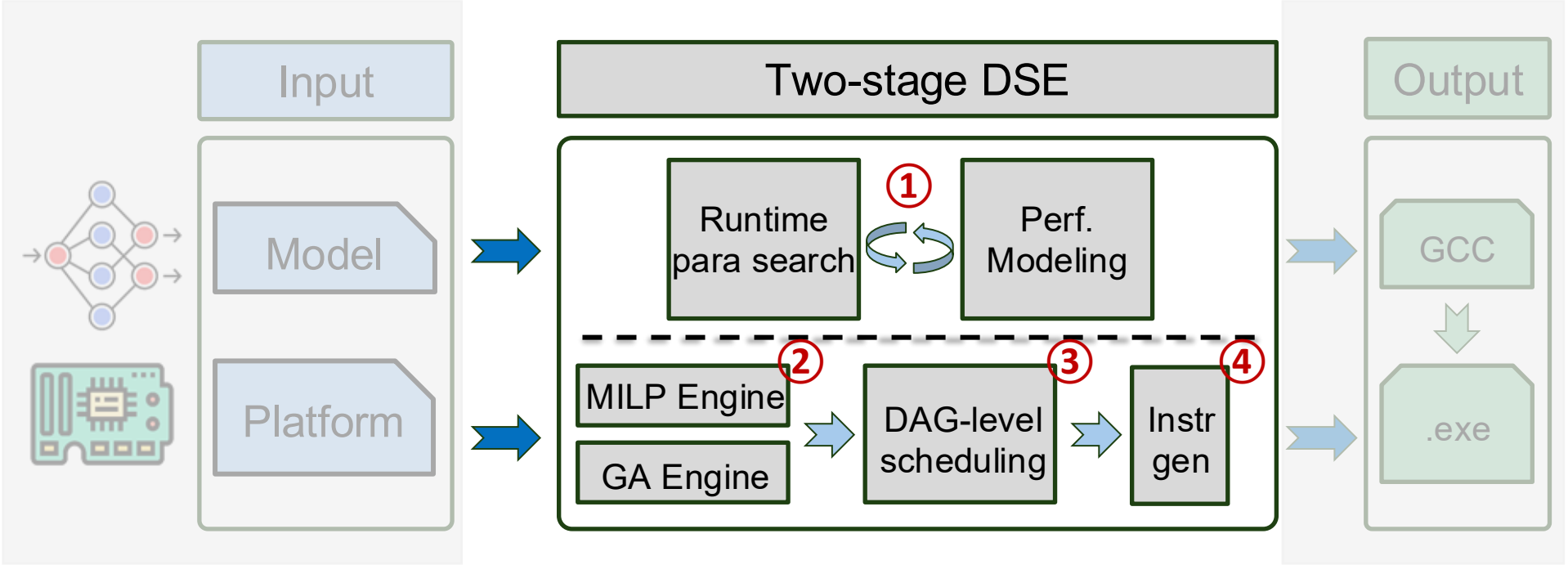
②Scheduling engines at the graph level for whole workloads.

- **Mixed Integer Linear Programming (MILP) engine can guarantee the optimal scheduling.**
- **Genetic Algorithm (GA) engine can achieve high efficiency.**



DORA Framework

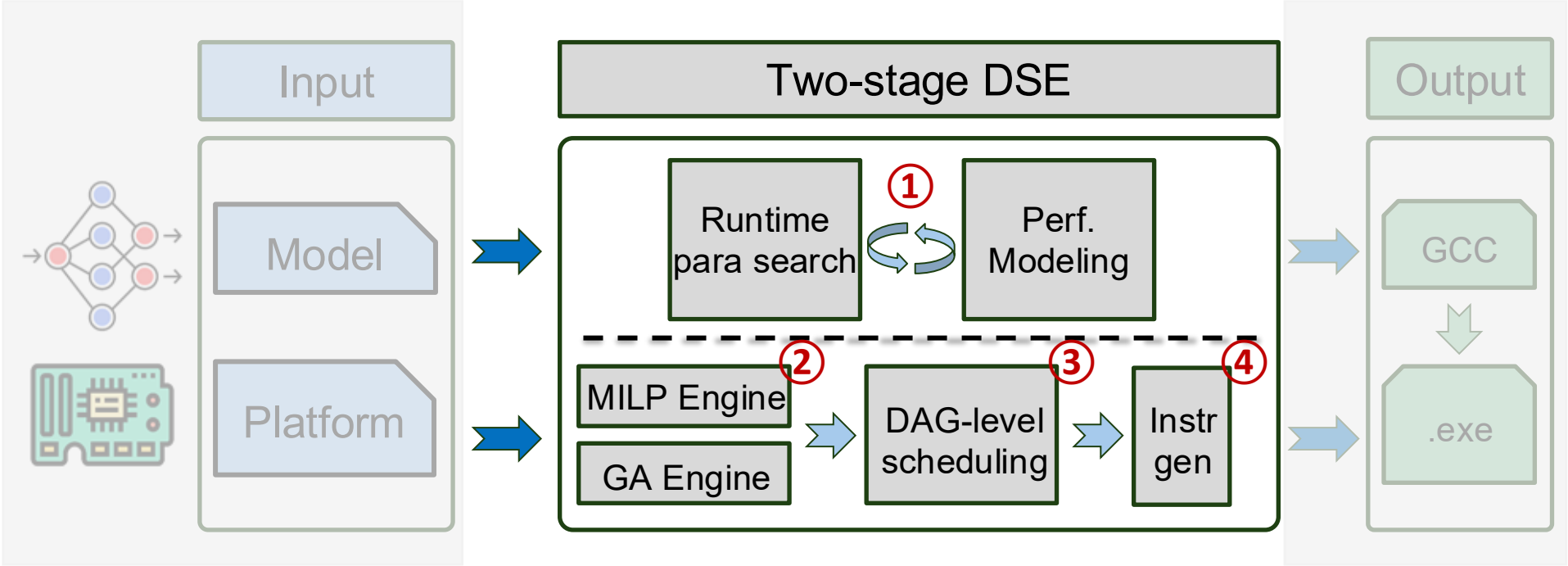
③Generate execution flow for each functional unit.



DORA Framework

③Generate execution flow for each functional unit.

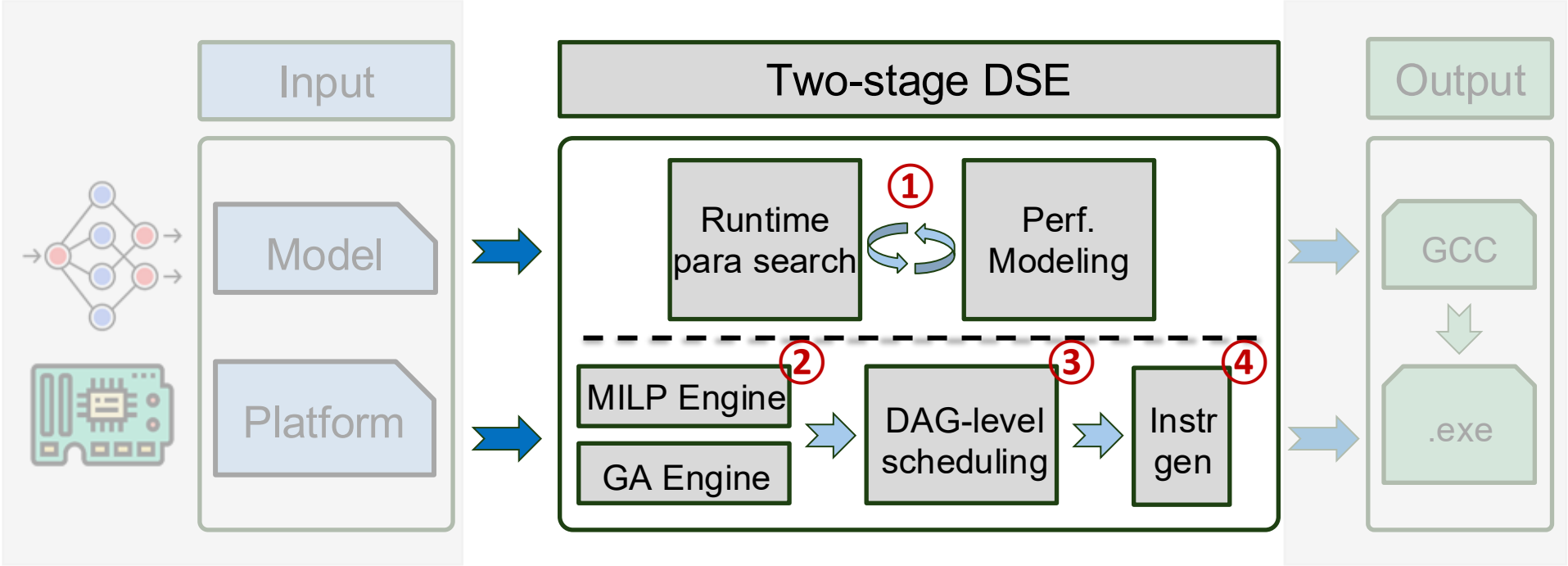
- **Analyze the dataflow for each functional unit;**



DORA Framework

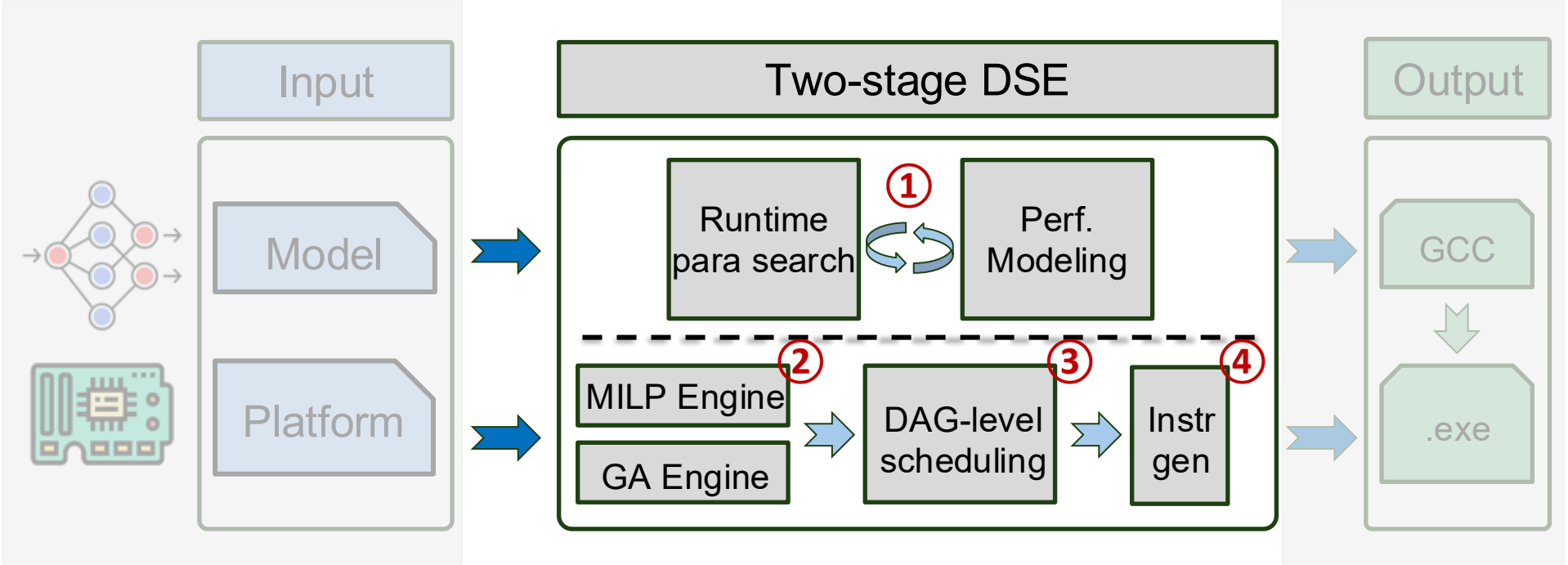
③Generate execution flow for each functional unit.

- **Analyze the dataflow for each functional unit;**
- **Prepare instruction sequence at FU level.**



DORA Framework

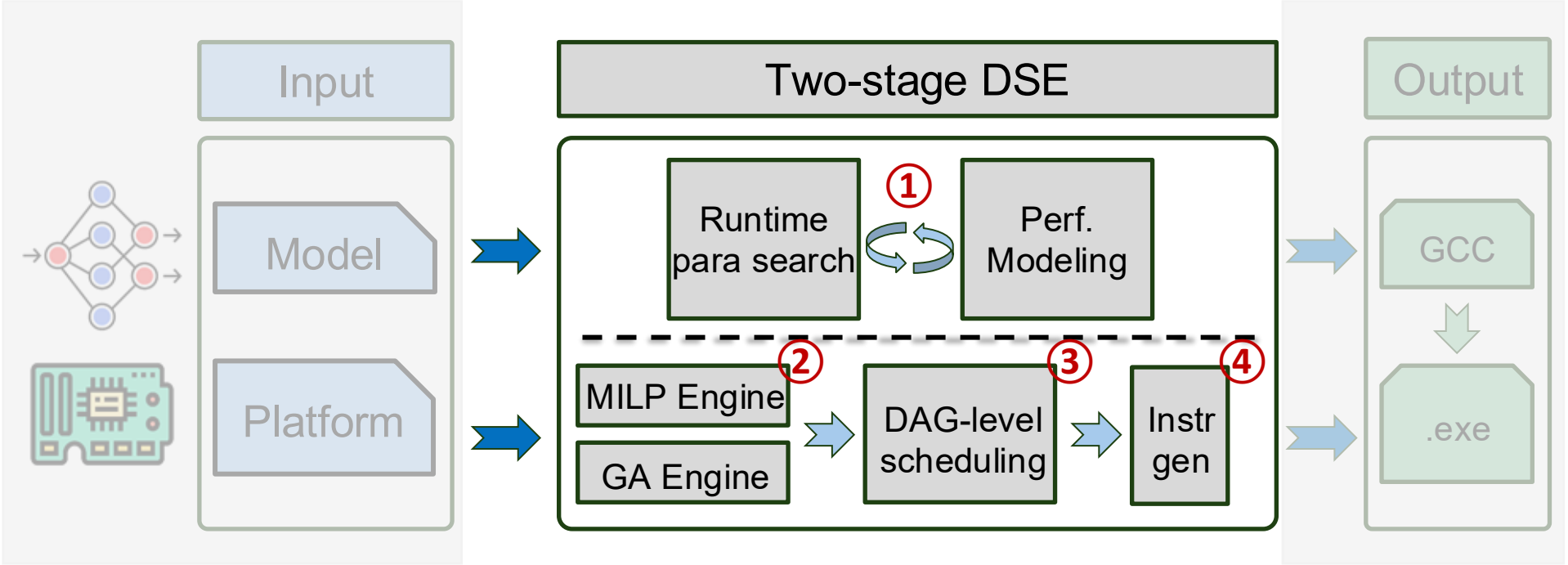
④Instruction sequence generation.



DORA Framework

④Instruction sequence generation.

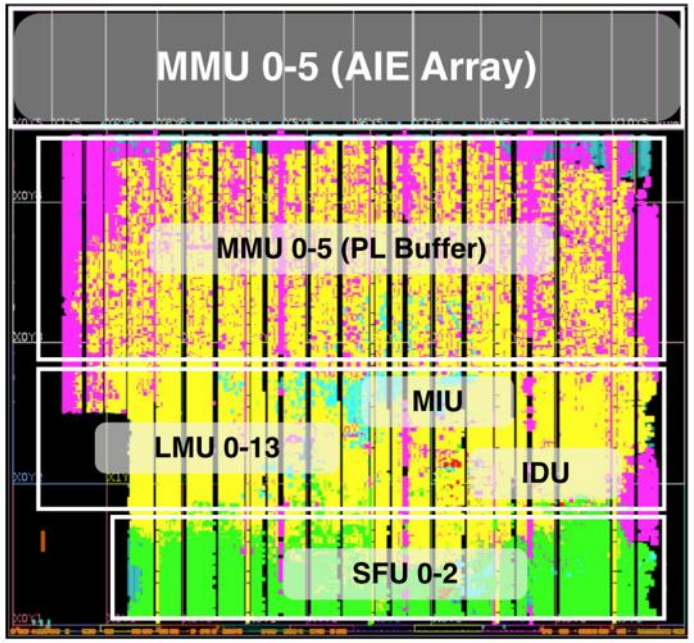
- Generate the instruction sequence and store them into host DRAM.



Experiment Setup

Experiment Setup

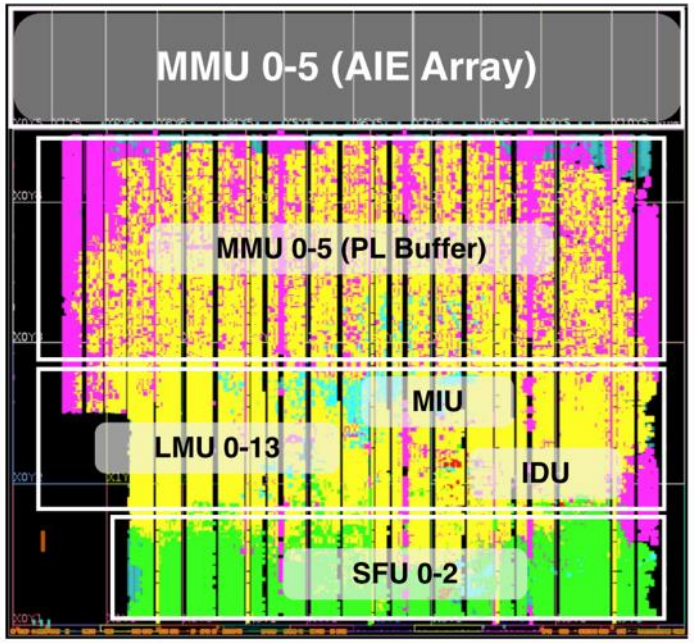
- We prototype DORA architecture on AMD Versal VCK190 platform.



DORA implementation layout

Experiment Setup

- We prototype DORA architecture on AMD Versal VCK190 platform.
- Instruction-related modules only consume REG and LUT, which are usually not bottlenecks in accelerator design.



DORA implementation layout

Unit	REG	LUTLogic	LUTMem	BRAM	URAM	DSP	AIE
MMU	110598 (6.12%)	103890 (11.52%)	0	576 (60%)	0	6 (0.3%)	384 (96%)
LMU	64358 (3.64%)	174538 (19.46%)	43358 (9.66%)	0	448 (96%)	280 (14.28%)	0
SFU	72000 (4%)	100974 (11.22%)	2328 (0.5%)	48 (4.95%)	0	96 (4.89%)	0
IDU	613 (0.03%)	415 (0.05%)	0	0	0	0	0
MIU	1810 (0.09%)	3913 (0.44%)	28 (<0.1%)	0	0	6 (0.3%)	0
Fully-con Network	264768 (14.7%)	68208 (7.58%)	0	0	0	0	0
Others	55111 (3%)	65504 (7.28%)	1419 (0.3%)	23 (2.4%)	0	0	0
Total	569258 (31.5%)	517442 (57.4%)	47133 (10.5%)	647 (67.4%)	448 (96%)	388 (19.8%)	384 (96%)

Resource utilization on VCK190

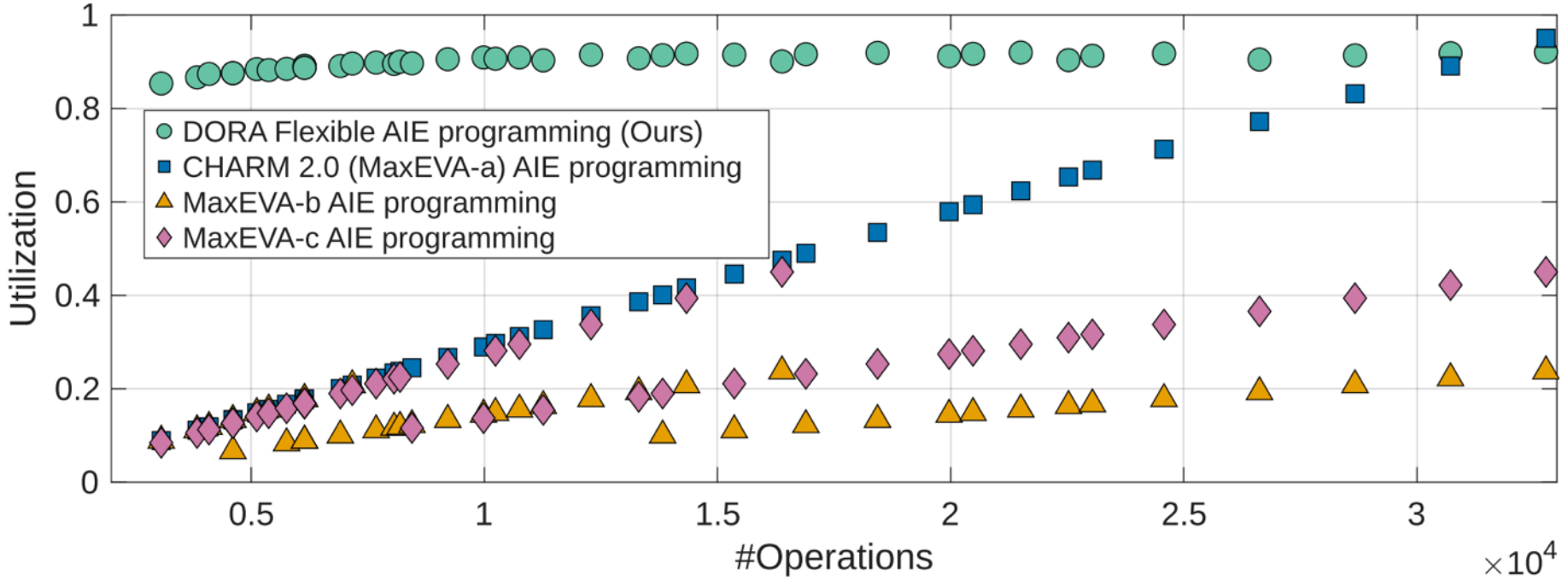
Experiment Results

Experiment Results

- **Flexibility on a single AI Engine core for diverse matrix multiplication.**

Experiment Results

- **Flexibility on a single AI Engine core for diverse matrix multiplication.**
- **Populate MM operations with different sizes, and DORA parallelism management efficiently handles diverse MMs.**



Experiments on a single AIE tile for diverse matrix multiplications.

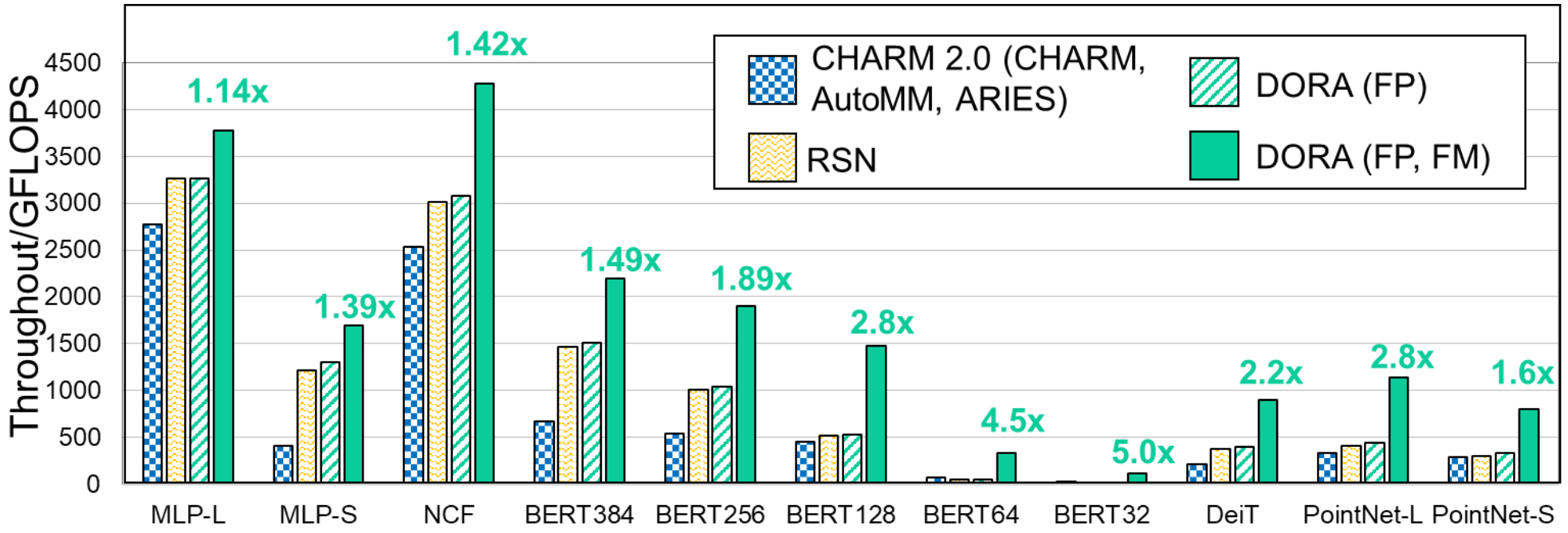
Experiment Results

Experiment Results

- **Efficiency on end-to-end DNN inference.**

Experiment Results

- **Efficiency on end-to-end DNN inference.**
- **DORA always outperforms customized architecture design and existing overlay design.**



Experiments on end-to-end DNN inference for diverse models.

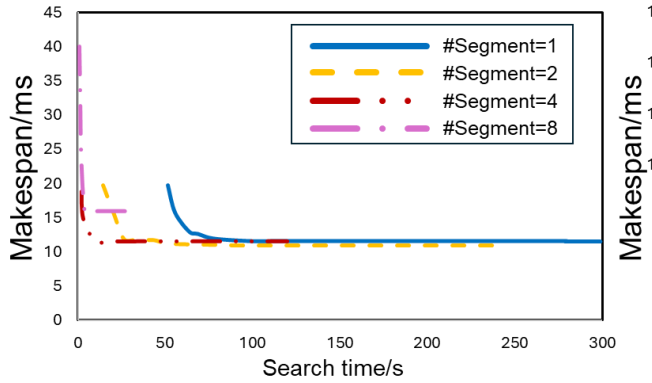
Experiment Results

Experiment Results

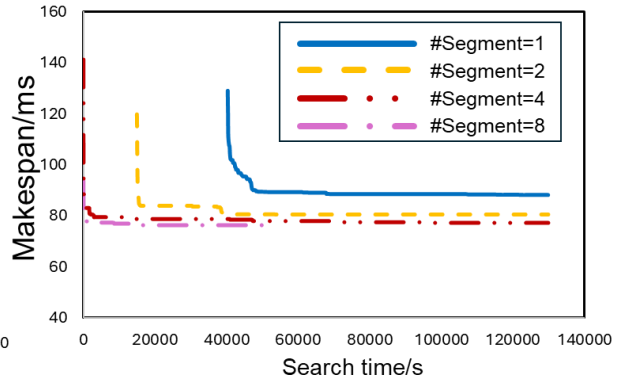
- Optimality and efficiency on MILP and GA formulations.

Experiment Results

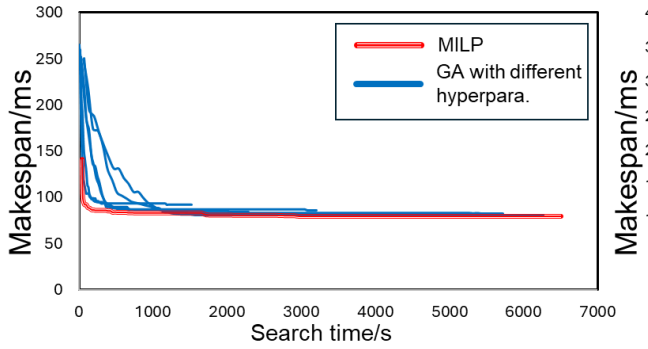
- Optimality and efficiency on MILP and GA formulations.
  - MILP guarantees the optimality of medium problem size.
  - GA provides the scalability and efficiency for large problem size.



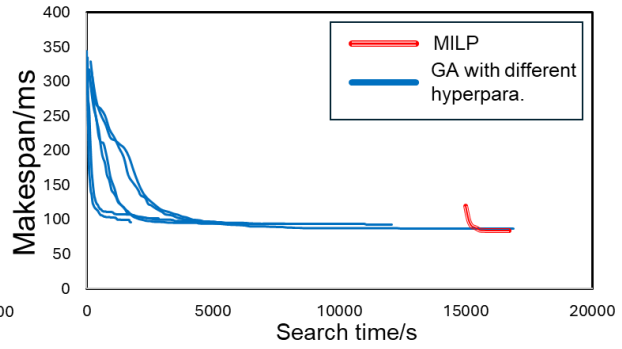
(a) Different DSE strategies on small model.



(b) Different DSE strategies on large model.



(c) MILP vs. GA on small model.



(d) MILP vs. GA on large model.

Key Takeaways

Key Takeaways

- DORA proposes a flexible overlay architecture that orchestrates dataflow with instructions for diverse workloads.
- DORA features an automatic framework with 2-stage DSE that can take applications and platform information to generate the ready-to-run executable files and bitstreams.
- DORA has a good generality and scalability, especially on large problem sizes.
- DORA open-source GitHub repo:

  https://github.com/arc-research-lab/DORA.git

Thank You & Welcome to questions!

DORA: Dataflow-Instruction Orchestration Architecture for DNN Acceleration

Xingzhen Chen, Zhuoping Yang, Jinming Zhuang, Shixin Ji, Sarah Schultz,
Zheng Dong*, Weisong Shi† and Peipei Zhou

Brown University; Wayne State University*; University of Delaware†

xingzhen_chen@brown.edu
peipei_zhou@brown.edu

https://peipeizhou-eecs.github.io/

https://github.com/arc-research-lab/DORA.git

DORA: Dataflow-Instruction Orchestration Architecture for DNN Acceleration

Xingzhen Chen, Zhuoping Yang, Jinming Zhuang, Shixin Ji, Sarah Schultz,
Zheng Dong*, Weisong Shi† and Peipei Zhou

Brown University; Wayne State University*; University of Delaware†

xingzhen_chen@brown.edu
peipei_zhou@brown.edu

https://peipeizhou-eecs.github.io/

https://github.com/arc-research-lab/DORA.git