

# LIM-GEN: A Data-guided Framework for Automated Generation of Heterogeneous Logic-in-Memory Architecture

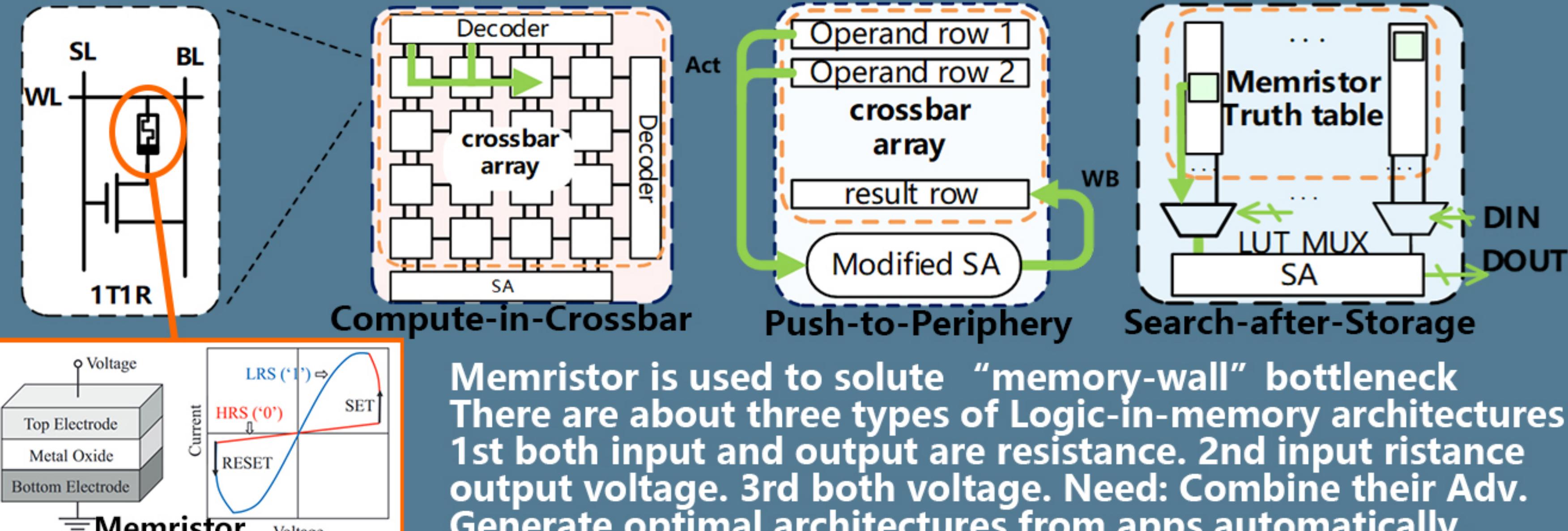
Libo Shen<sup>✉†</sup>, Boyu Long<sup>✉†</sup>, Rui Liu<sup>✉‡</sup>, Xiaoyu Zhang<sup>✉†</sup>, Yinhe Han<sup>✉</sup>, Xiaoming Chen<sup>✉</sup>

<sup>✉</sup>Institute of Computing Technology, CAS, <sup>†</sup>University of Chinese Academy of Sciences, <sup>‡</sup>Xiangtan University  
ContactMe! [shenlibo21s@ict.ac.cn](mailto:shenlibo21s@ict.ac.cn) | <https://github.com/BUPTslb> | [juankingdon\(IG\)](#) | [shen0605Don\(wechat\)](#)

## Main Idea

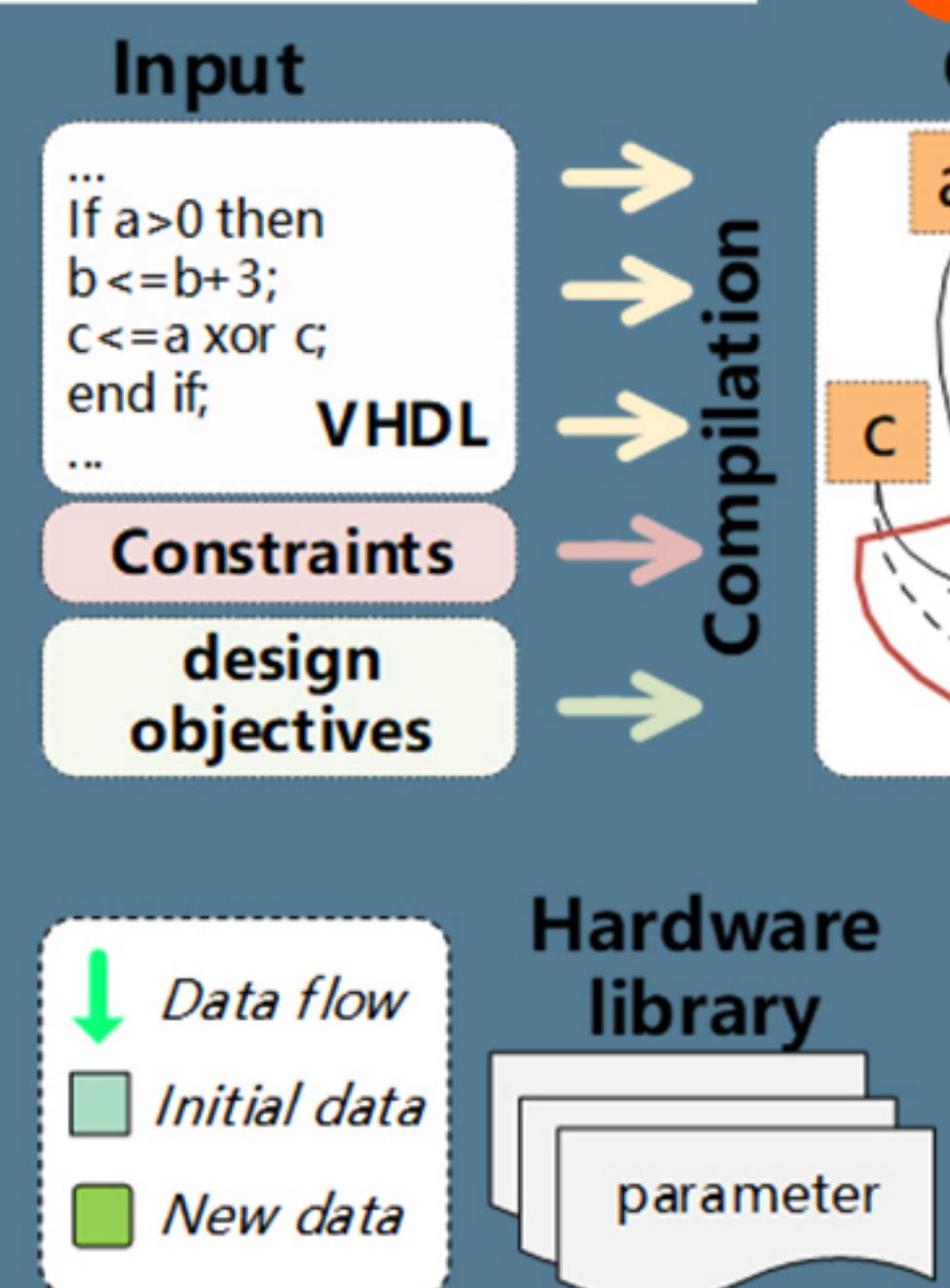
1. We propose combining different Logic-in-memory(LIM) modes to create heterogeneous LIM architecture.
2. We optimize the existing LIM modes and modularize them to build a rich design space.
3. To generate heterogeneous LIM architectures from application behavior descriptions, a synthesis flow is proposed. It takes VHDL as input and performs a series of optimizations on data distribution, task allocation and crossbar mapping, to generate PPA-optimized LIM architectures

## Background: LIM architectures

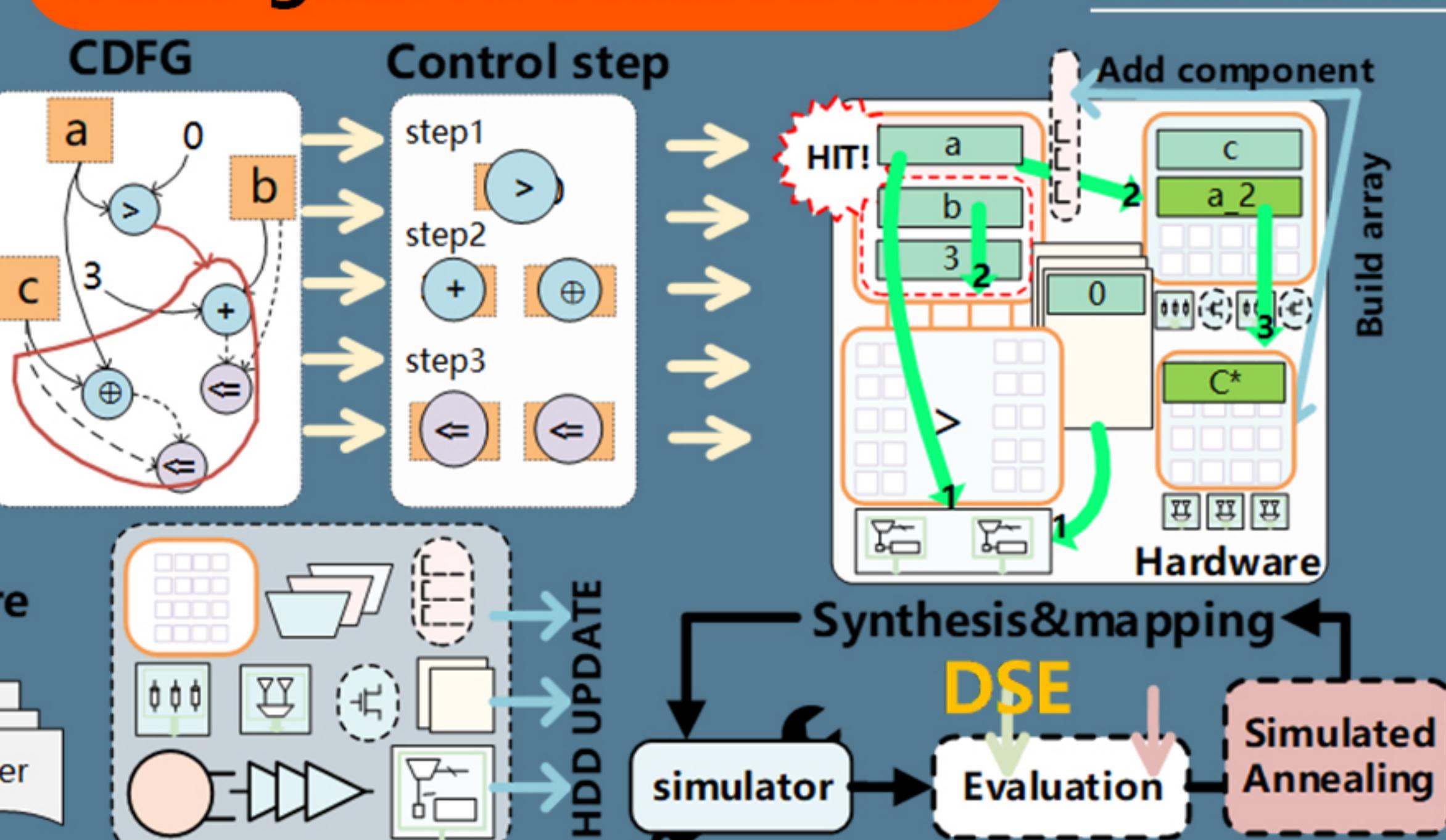


## GOAL: Get General Architecture by High Level Synthesis (software-hardware codesign)

### From behavioral expression



### Data-guided Framework



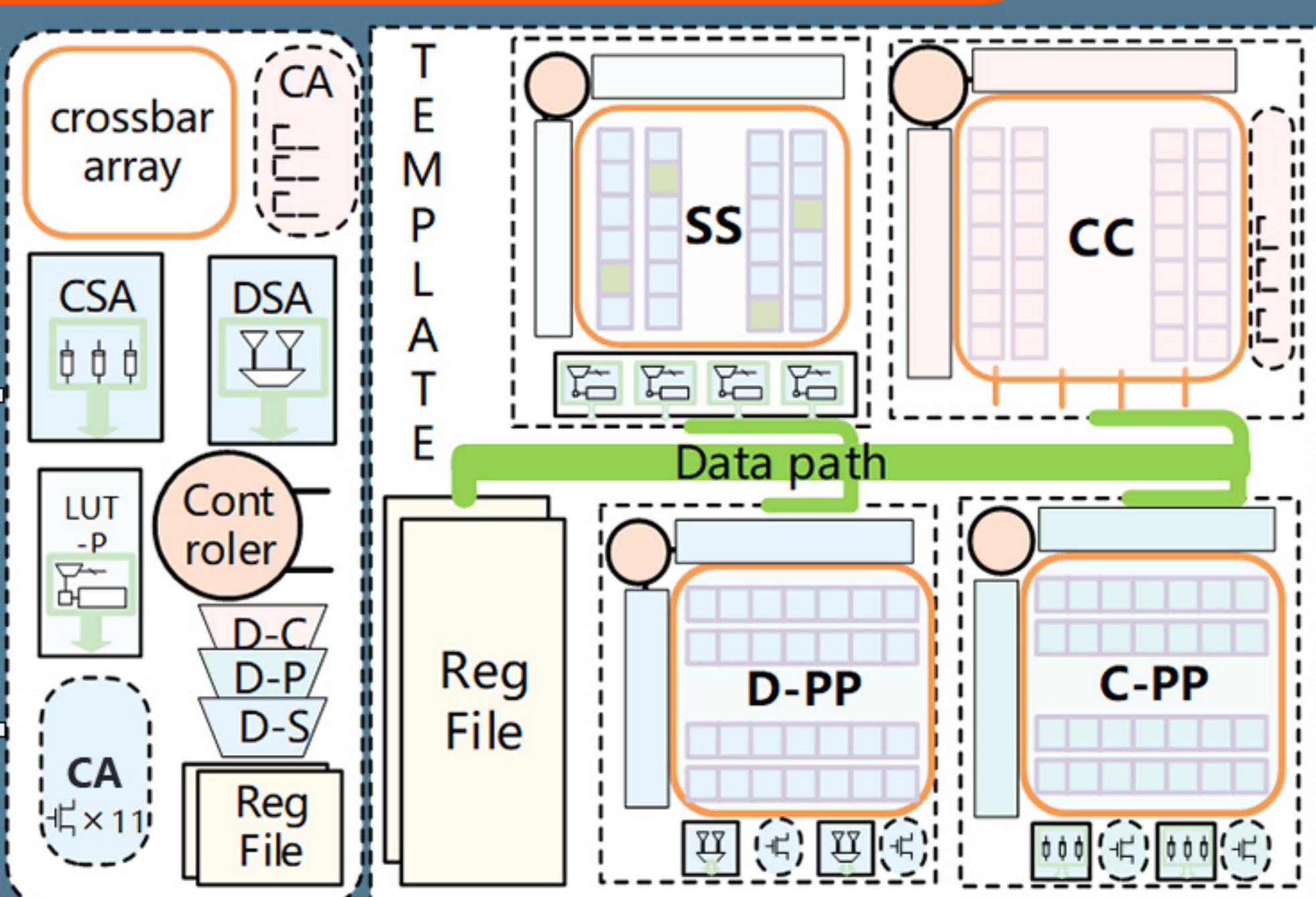
### To Architecture

We optimize the generality of each LIM architecture and get Hardware library. Final Architecture is composed of these components.

SA :sense amplifier  
CSA:capacitor SA  
DSA:double threshold SA  
CA :carry align  
D-X:decoder

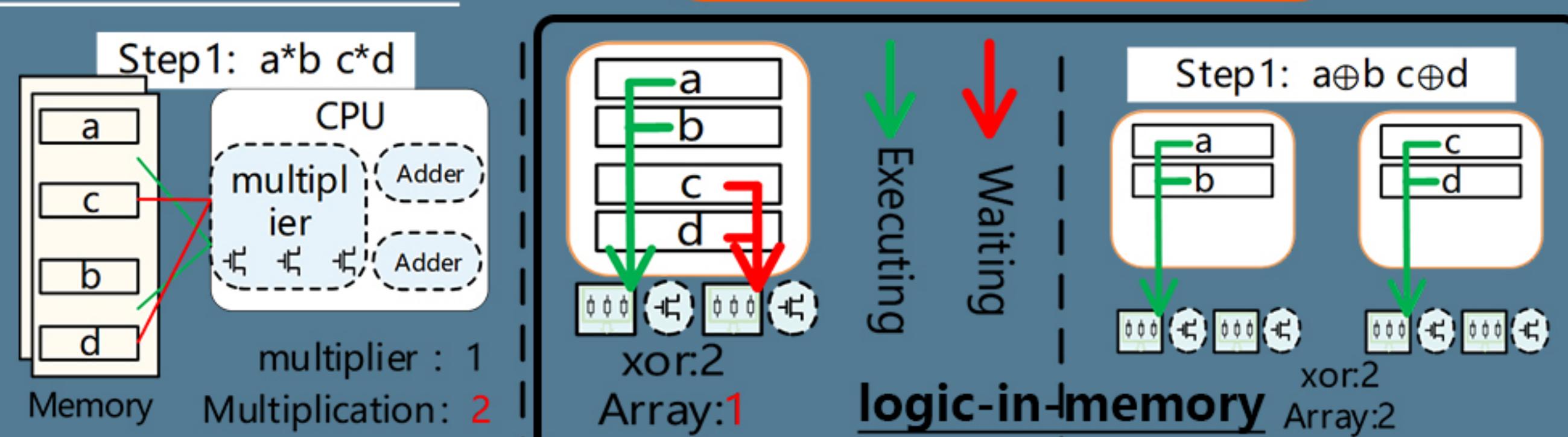
RegFile for immediate data intermediate data

### Heterogeneous LIM Architecture



Not only decides the architecture design, but also decides the location of data (Put data in right place)

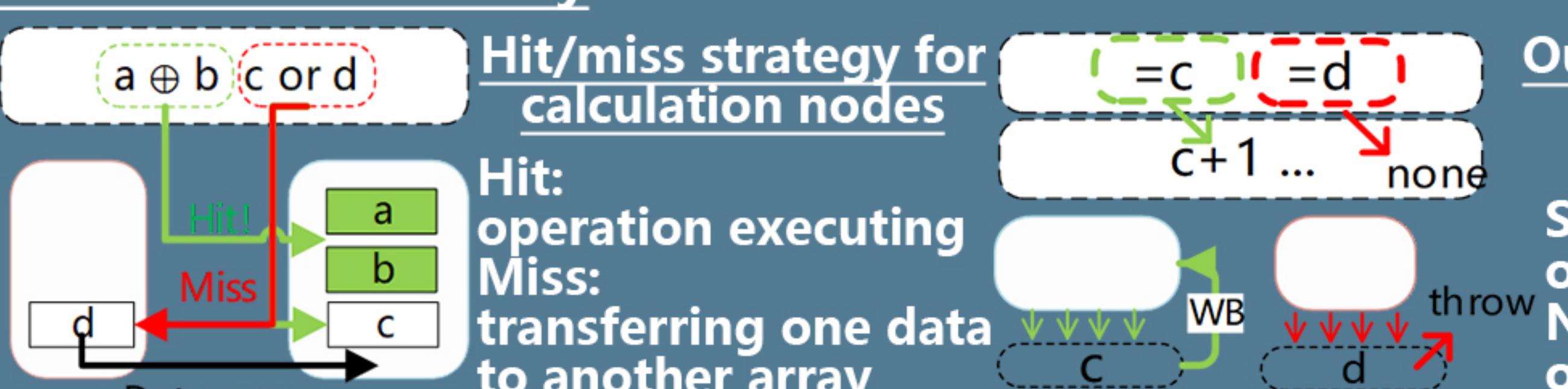
### Structural conflict



### Schedule Strategy

Conventional Arc: computing resource amount  
LIM Arc: Data Location

### Select Execution Array

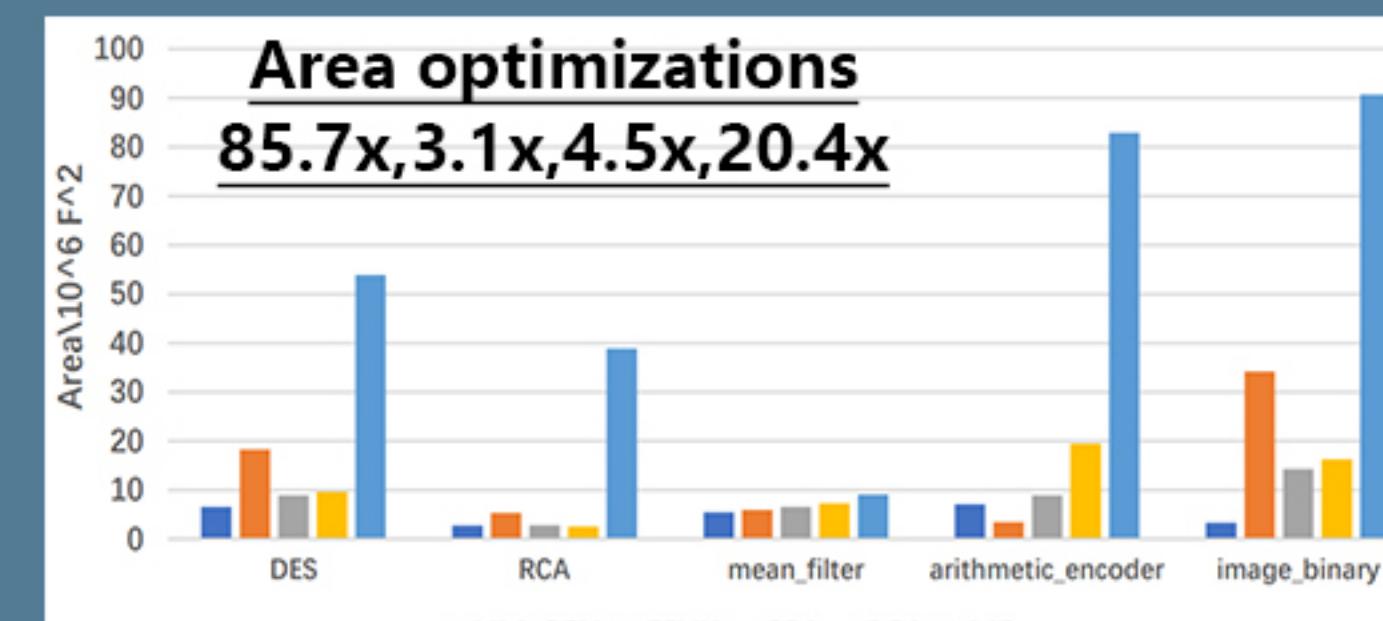
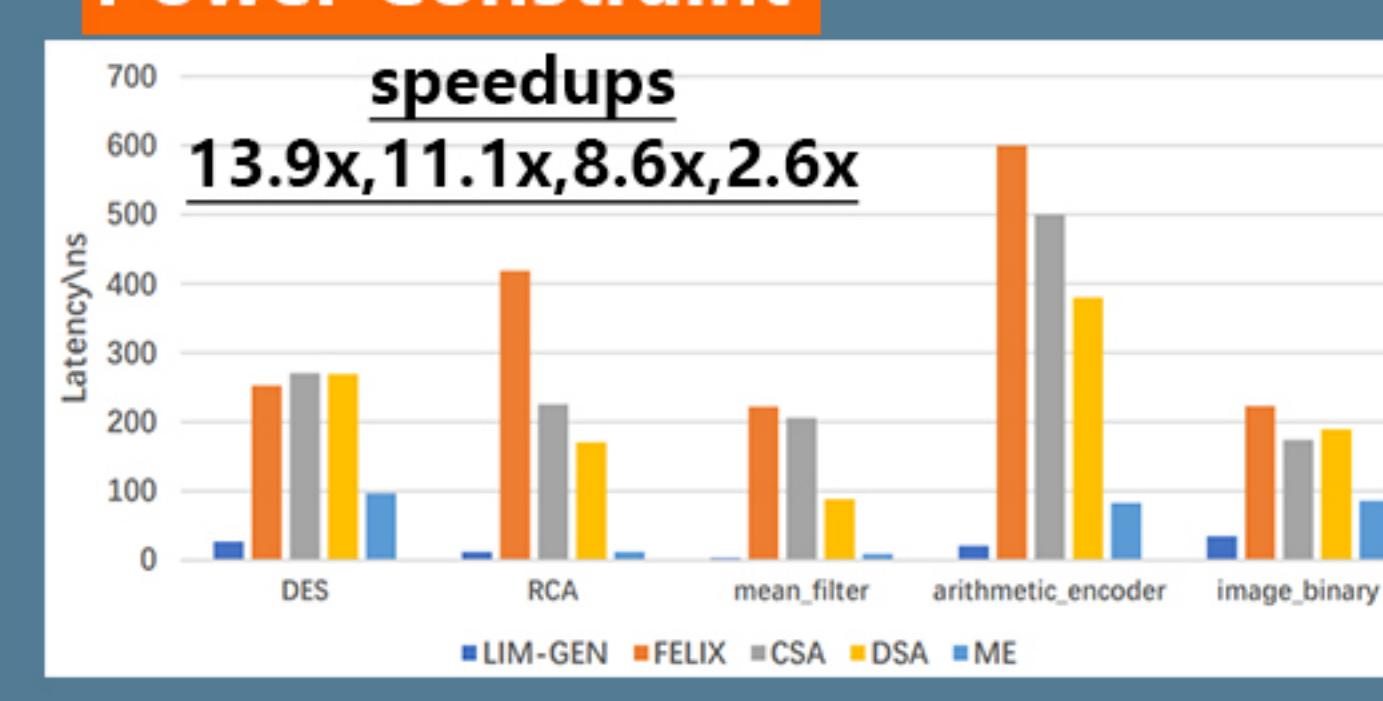


### Out-degree strategy for assignment nodes

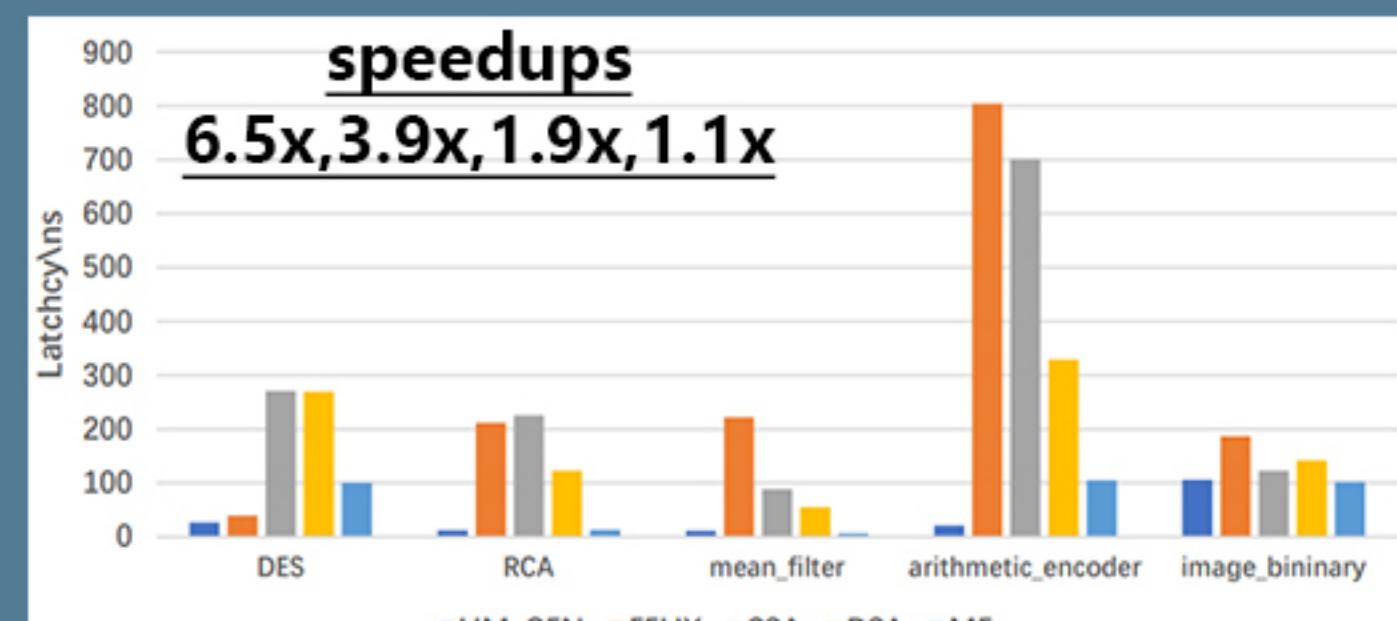
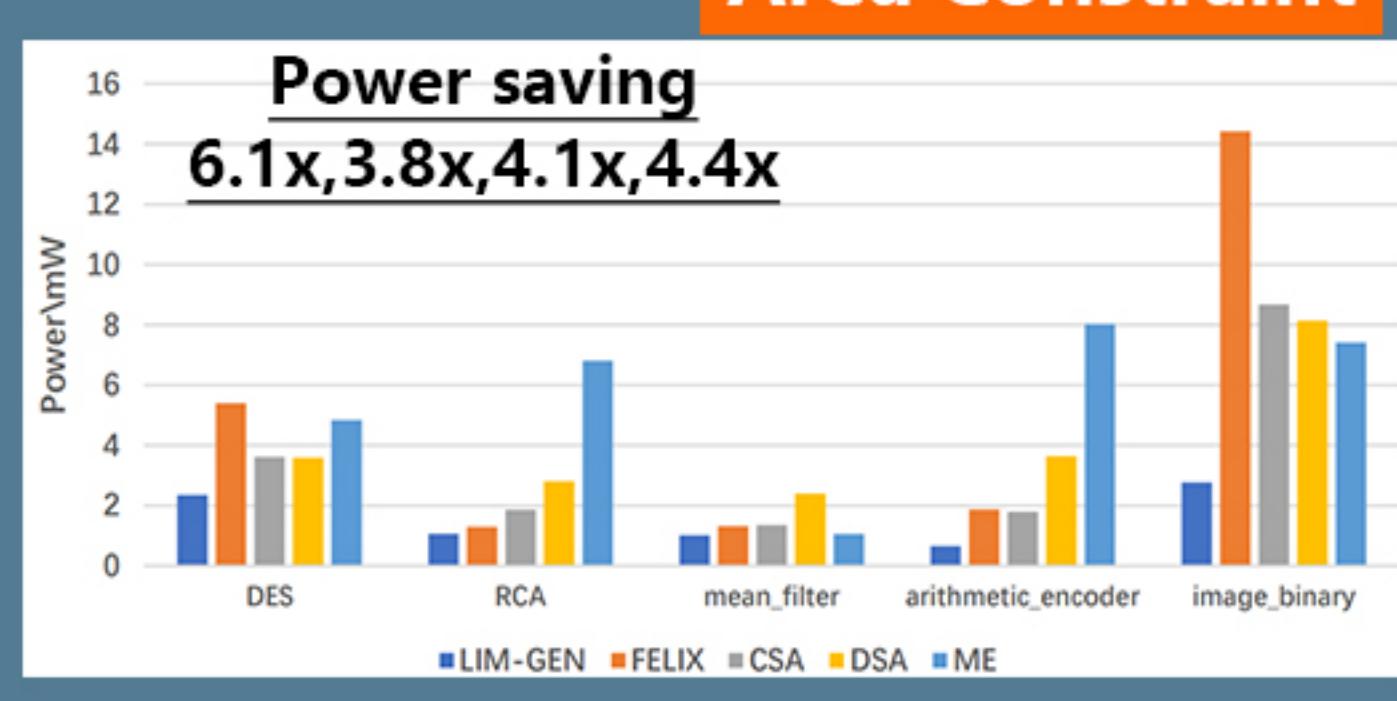
Subsequent using: outdegree>0, write back  
No sub-using: outdegree==0, throw

### Experiment result

#### Power Constraint



#### Area Constraint



Each design is exclusively optimized for only one specific metric, while the other metrics are required to remain within reasonable ranges.

### PRIOR DECISION

- Use Compute-in-crossbar for Fine-grained task
- Use Push-to-Periphery for Parallel task
- Use Search-after-storage for Complex task

Determination process of the array type requires some randomness, because the impact of the execution of an operation on the overall performance is unknown and hard to evaluate

### RUN TIME

The crossbar types will be randomly changed during DSE

### Conclusion

By comparing heterogeneous LIM architectures designed by LIM-GEN with four individual LIM architectures, we observe significant improvements in latency, area, and power consumption. It demonstrates that a single LIM mode may not be the best, and instead, different LIM modes should be combined for different logic operations.