Virtualization of CGRA Based Accelerators

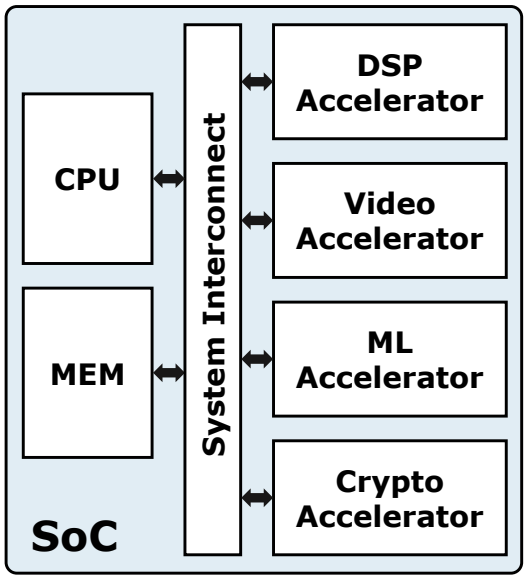
Priyanka Raina
praina@stanford.edu

Work led by Taeyoung Kong, Kalhan Koul and Keyi Zhang

August 26, 2021

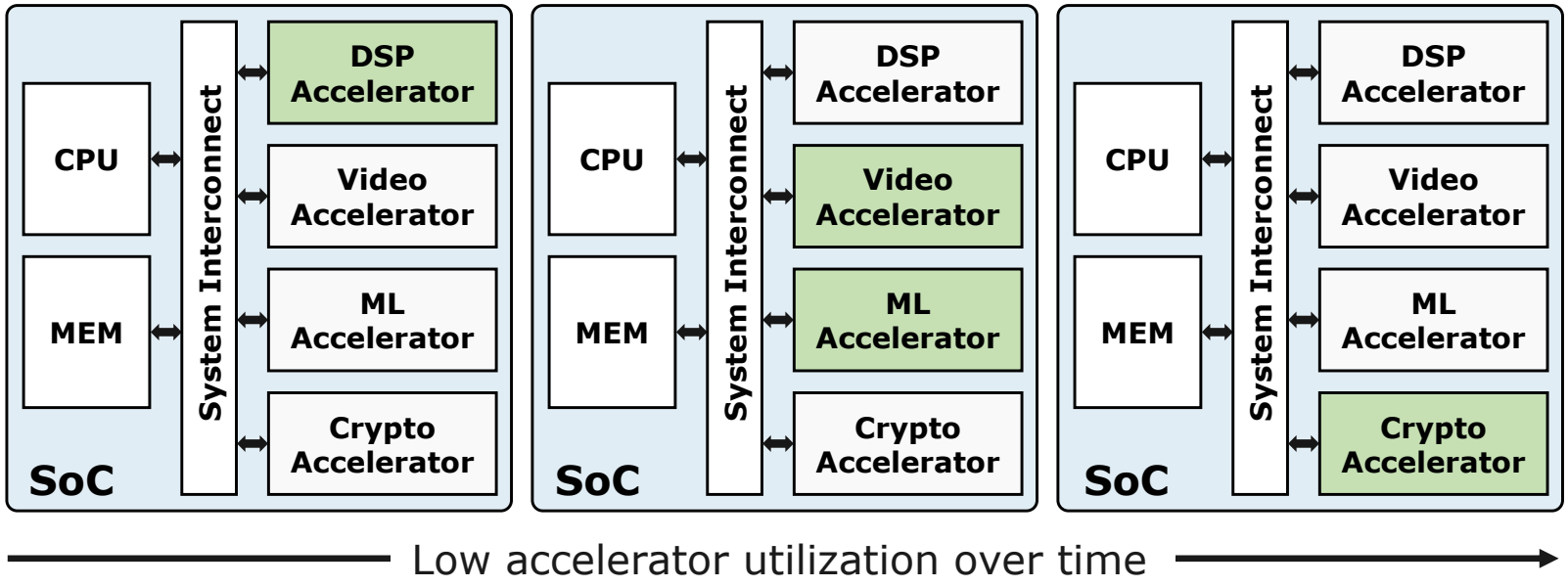
SoCs with Fixed Function Accelerators

- With the slowdown in technology scaling, accelerators are essential for continued performance and energy improvements
- A modern SoC has dozens of semi-fixed-function accelerators
  - True for both edge and cloud SoCs



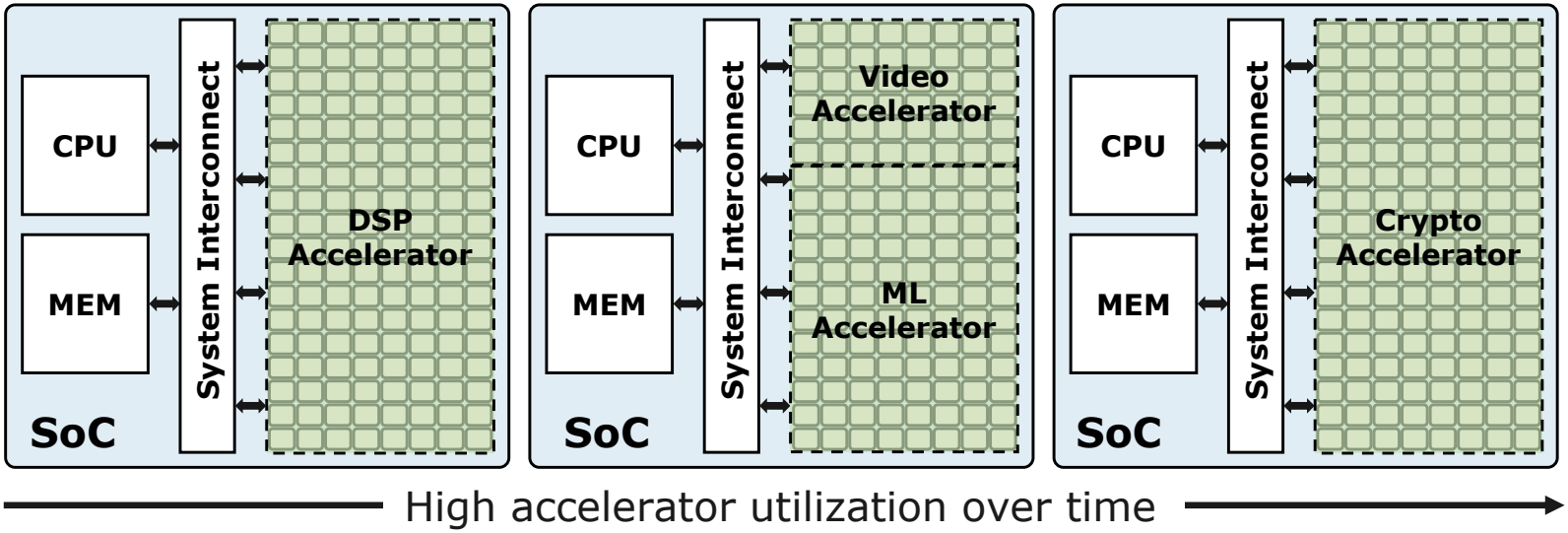
Challenges

- Does not accommodate **evolution and scaling** of applications
- Difficult to keep the accelerators well-utilized



SoCs with Reconfigurable Accelerators

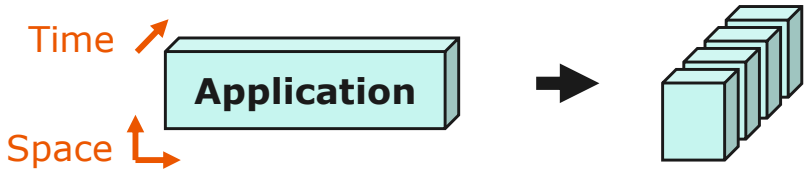
- CGRA-based reconfigurable accelerators
  - Provide programmability for accommodating application evolution
  - Can be dynamically reconfigured to accelerate different application scales and mixes, leading to high accelerator utilization **CGRA Virtualization**



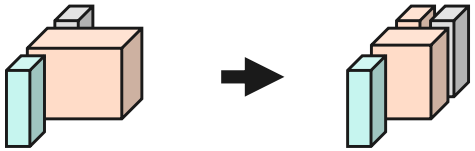
CGRA Virtualization Use Cases

Edge Environment

- Small CGRA, power and area constrained, low latency target, single user
- Partition a single large application to run on the small CGRA temporally



- Map several applications on the CGRA spatially and temporally

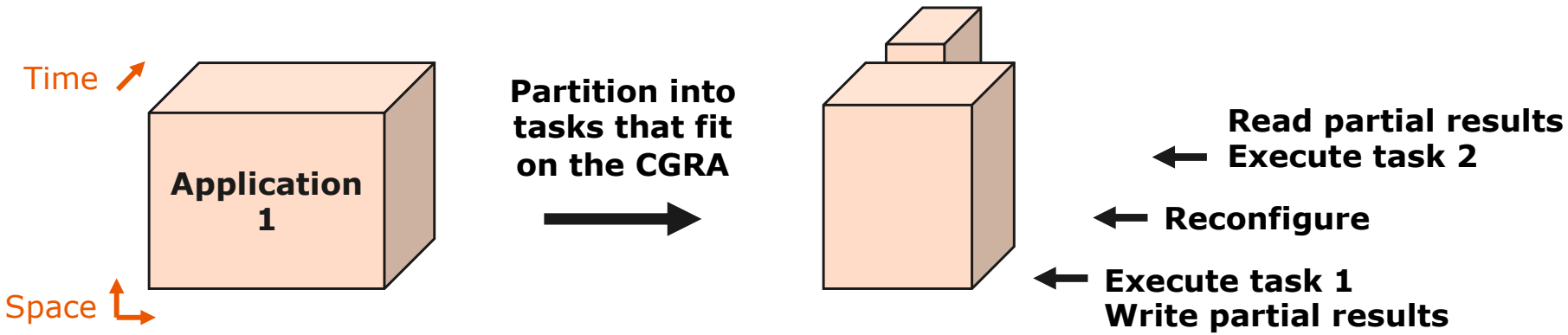


Cloud Environment

- Large CGRA, high throughput target, multiple users/models
- Aggregate applications from multiple users to run spatially and temporally on a large CGRA

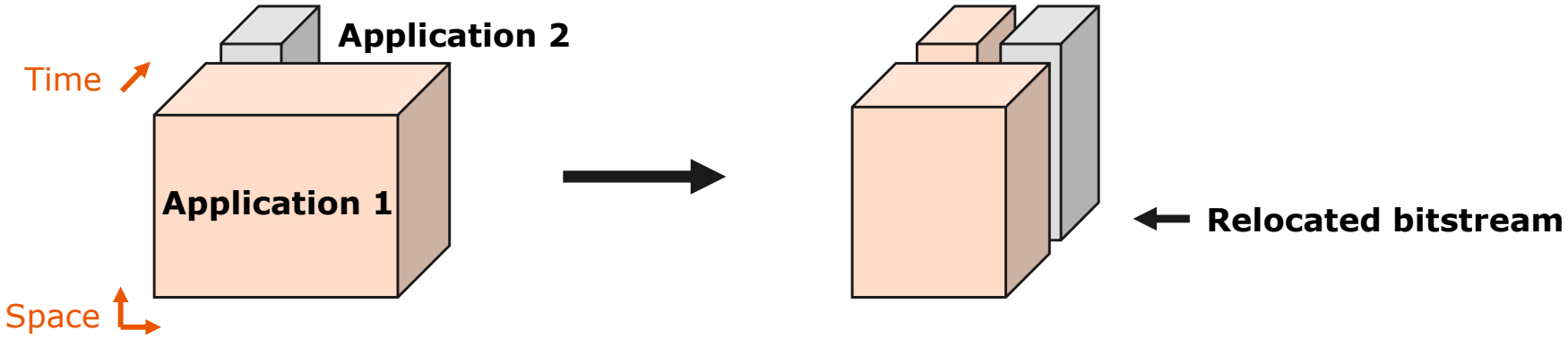
Enabling CGRA Virtualization

- **Application graph partitioning**
  - When an application requires more resources (PEs, MEMs, GLB tiles) than available



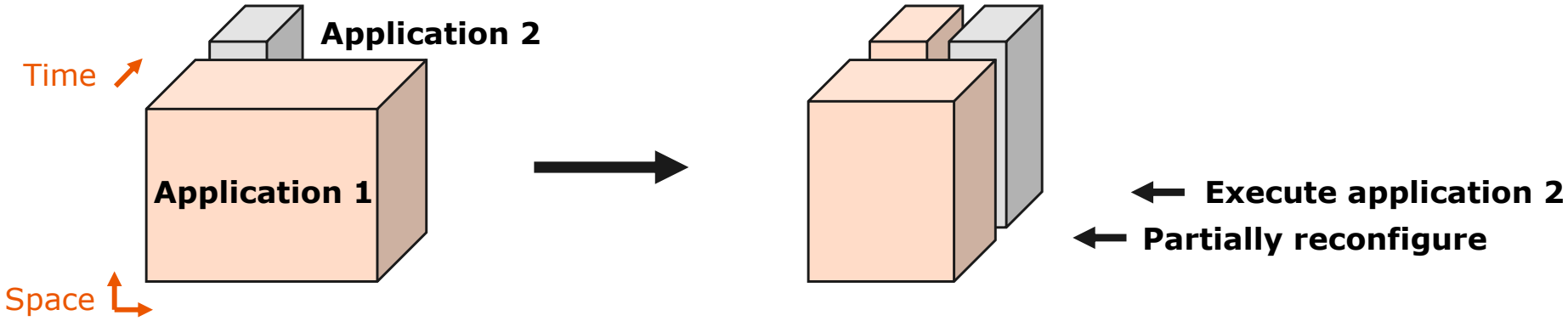
Enabling CGRA Virtualization

- Application graph partitioning
- **Application bitstream relocation**
  - Move pre-compiled bitstream to an available region on the CGRA
  - Requires the CGRA to be somewhat homogeneous



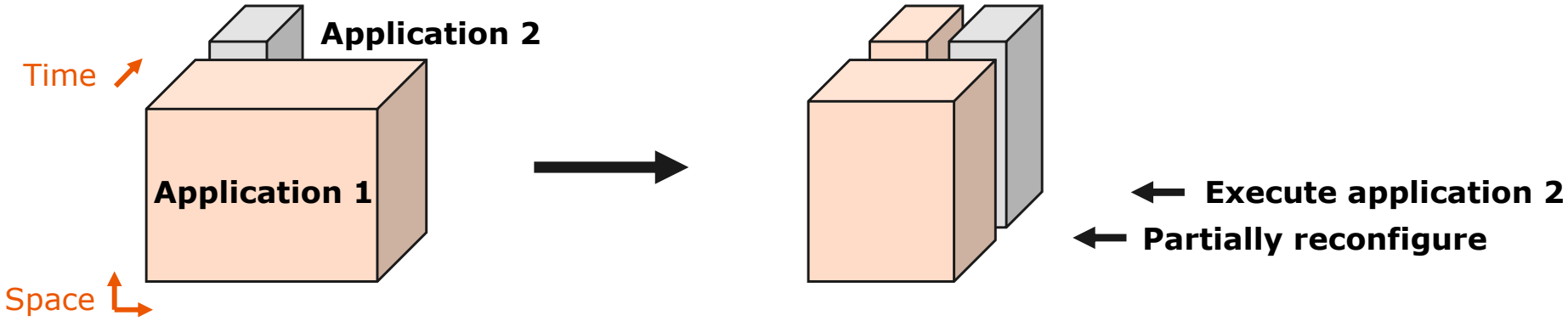
Enabling CGRA Virtualization

- Application graph partitioning
- Application bitstream relocation
- **Dynamic partial reconfiguration (DPR)**
  - While another application may be concurrently running



Enabling CGRA Virtualization

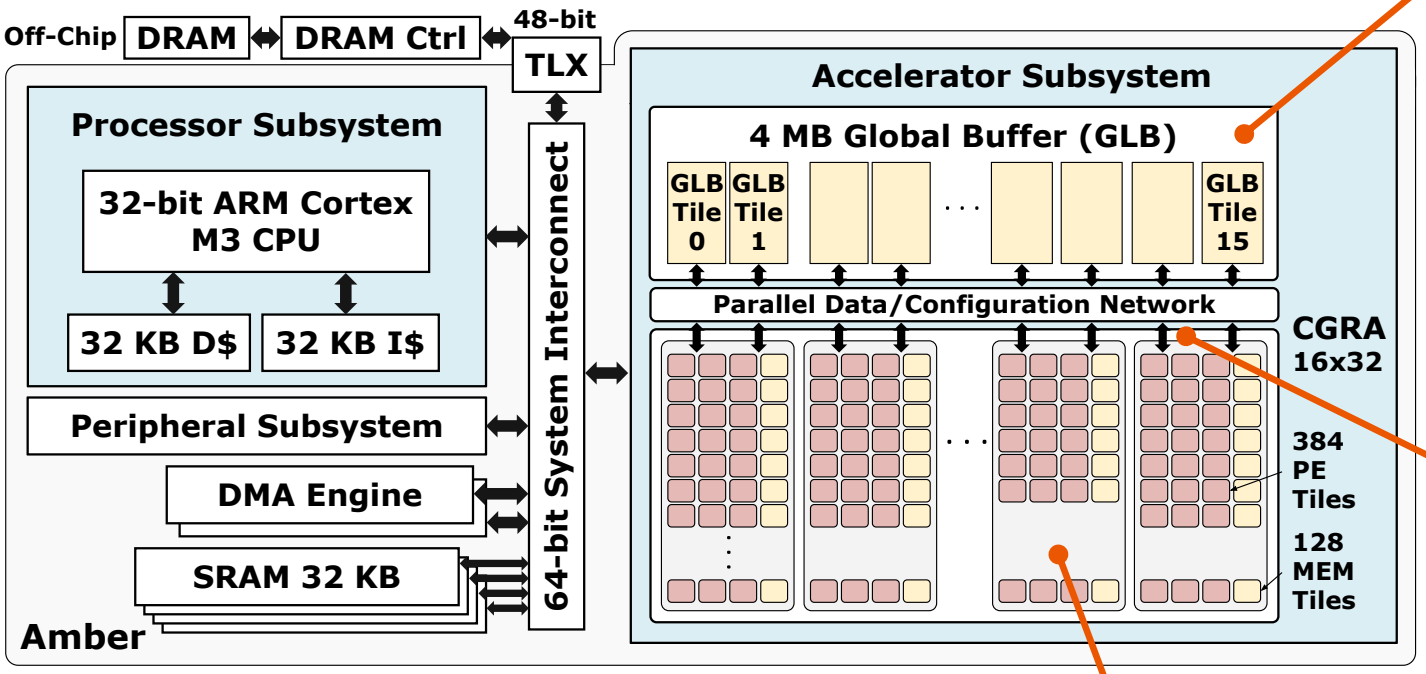
- Application graph partitioning
- Application bitstream relocation
- Dynamic partial reconfiguration (DPR)
- **Memory for staging configuration bitstreams and application data**



Creating a CGRA with Virtualization Support

Following our agile design methodology, we make these architectural modifications on top of a baseline CGRA

Virtualization in Amber SoC



Multi-banked global buffer with software-managed address generators

- Stages both application bitstreams and data
- Provides high-bandwidth access

Parallel dynamic partial reconfiguration (DPR)

- For full CGRA configuration (75,520 registers), parallel DPR is **128x faster** than AXI4-Lite

Homogeneous 4-column partial reconfiguration (PR) regions

Virtualization in Next-Generation Onyx SoC

- Adding bitstream relocation logic into the GLB load DMA instead of using the control processor
- Design space exploration of PR region shape and CGRA interconnection network
- Faster DPR by double buffering configuration in the CGRA tiles
  - Explore performance vs. area tradeoff

- Evaluation for different application scales and mixes in edge and cloud use cases