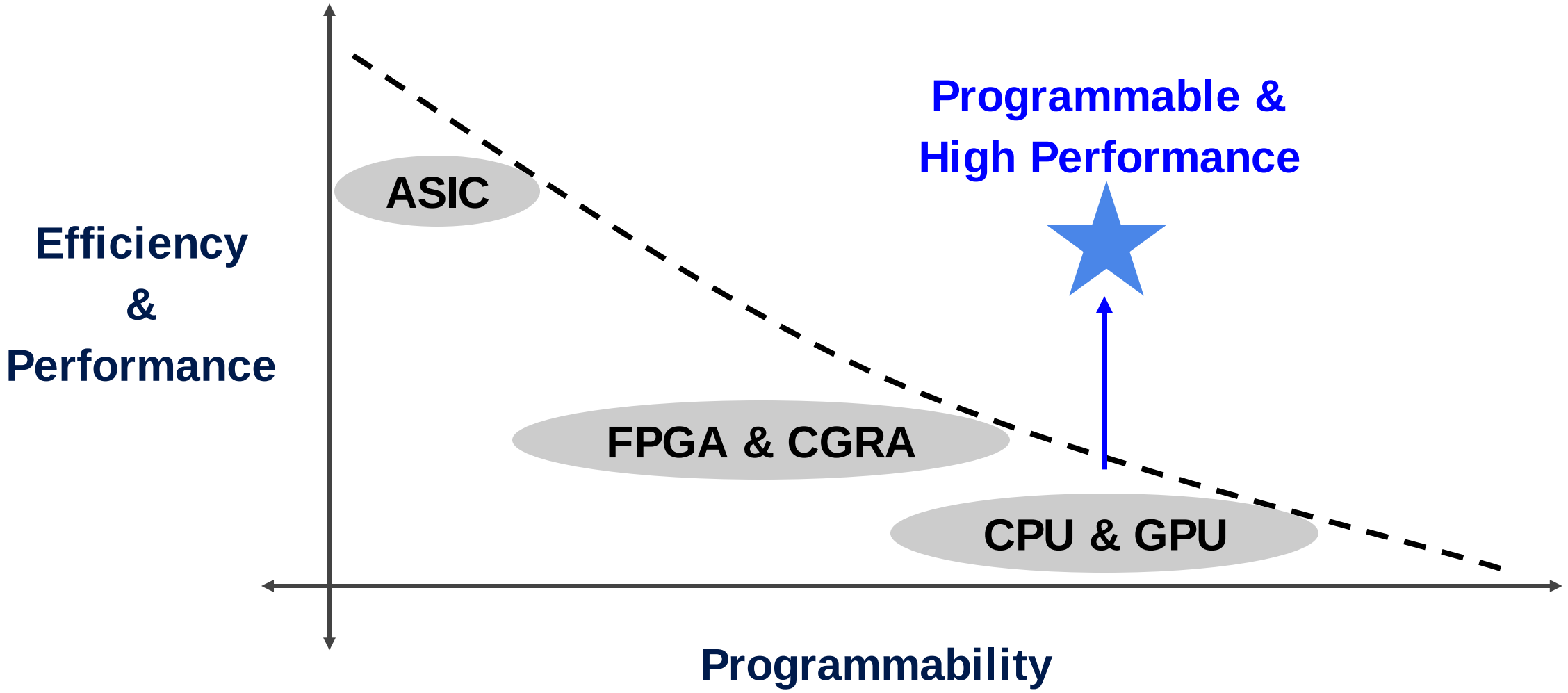
Introspective Computers

Ronald Dreslinski, Trevor Mudge
University of Michigan

Arm Research Summit - September 2019

Specialization Tiers

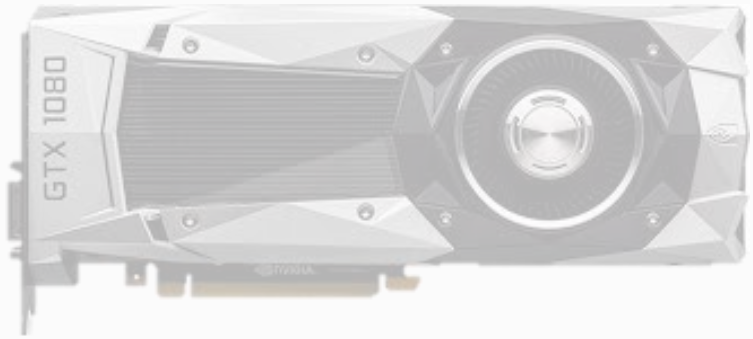


Existing GPPs have limited efficiency and performance

- **Gold standard for parallel workloads**
- **SIMT execution model suffers on control-heavy segments**
- **Requires careful data alignment for high utilization (SIMD hardware)**

- **Gold standard for serial workloads**
- **High overhead for complex OoO cores with coherent caches**
- **Comparatively limited compute resources and parallelism**

Existing GPPs have limited efficiency and performance



- Gold standard for parallel workloads

- SIMT execution model suffers on control-heavy segments
- Requires careful data alignment for high utilization (SIMD hardware)

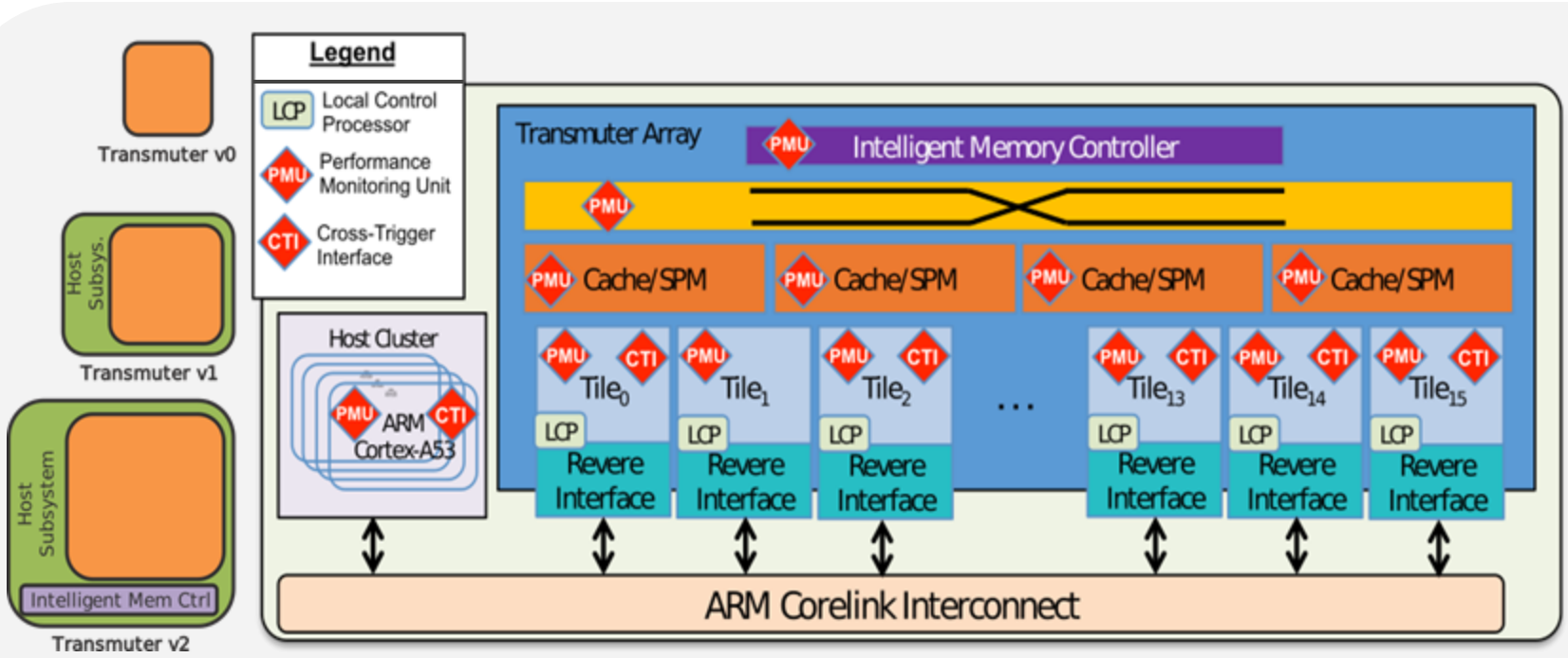
Example from Pal et. al., 2018:

- Outer product sparse matrix multiplication
- GPU threads diverge due to conditionals and uneven distribution of non-zeros

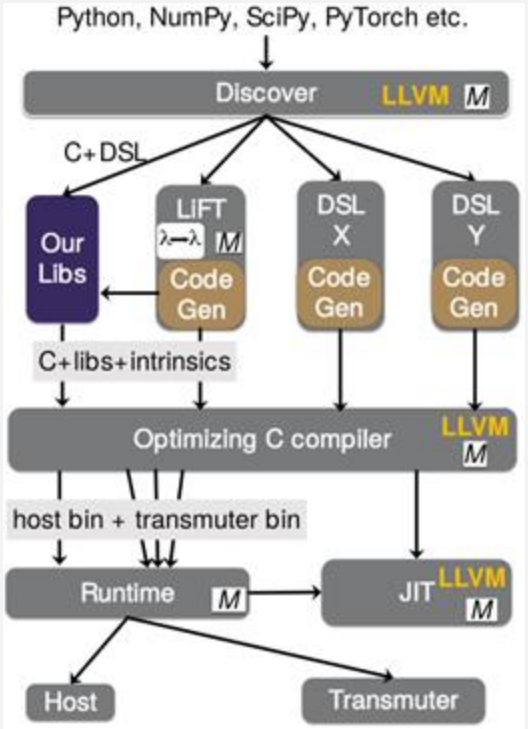
... workloads at this level of sparsity achieve fewer than 1 GFLOPS. L1 cache hit rate and utilized memory bandwidth drop to under 40% and 10 GB/s, from 86% and 30 GB/s at 10% density.

Transmuter: An Introspective Computer Initiative

- Part of a larger, multi-year DARPA program with collaborators at ARM, Arizona State University, and University of Edinburgh
- Led by PI's at UMich, with significant work by many students!



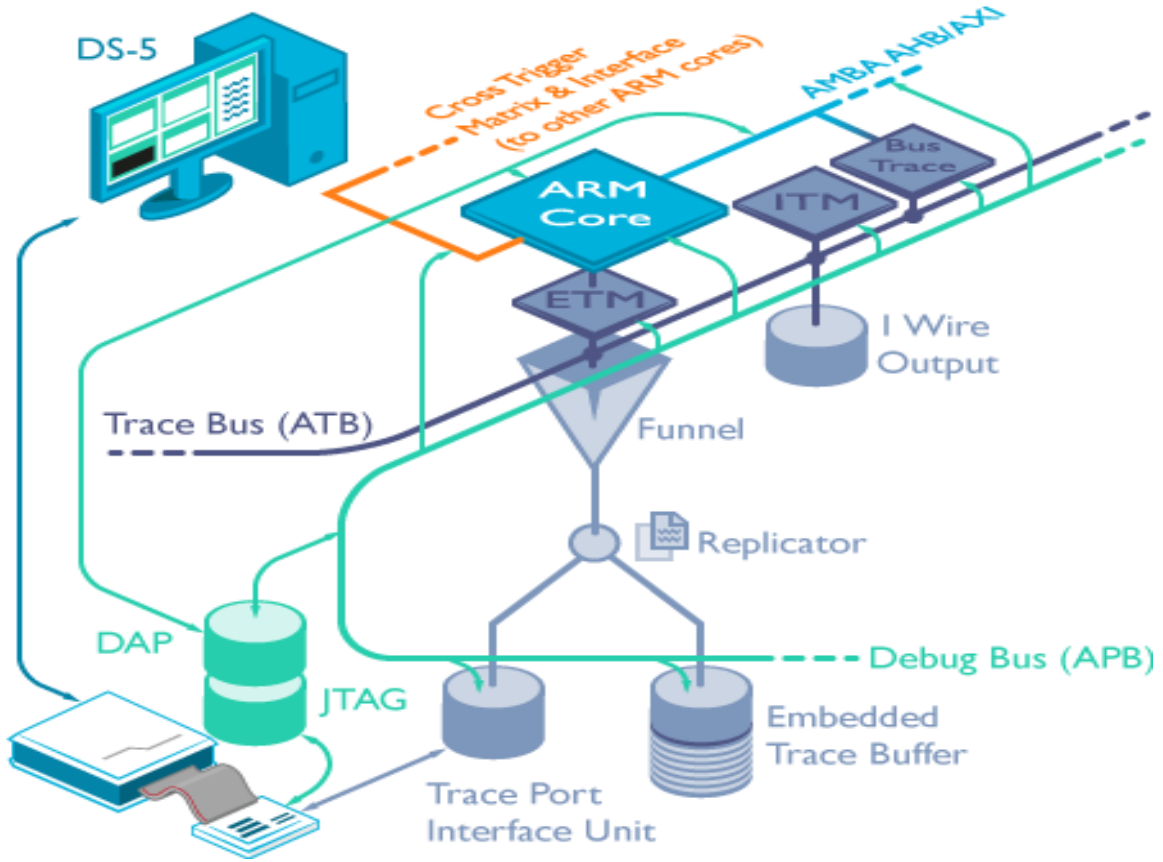
Evolution of transmuter hardware



Software and compiler support by UoE

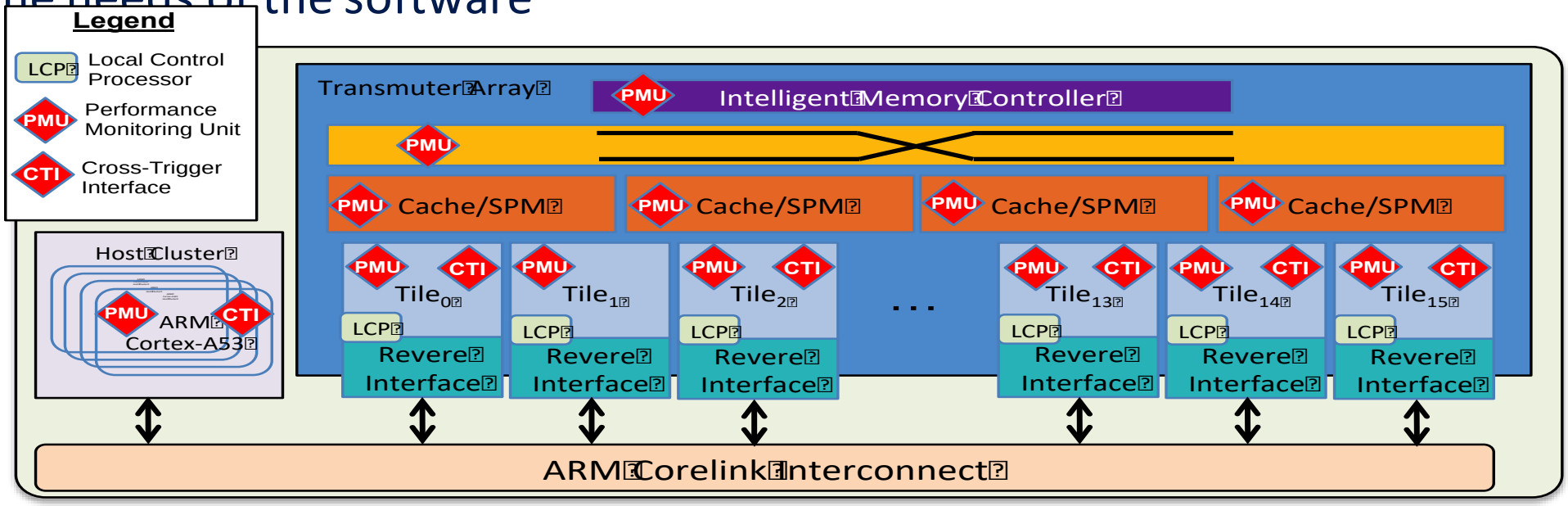
Highly Introspective Design

- Leverage industry standard arm CoreSight technology to provide debug/introspective support
- Each processing element able to be single-stepped for debugging purposes
- All IP extended to support performance monitoring to enable runtime decisions about reconfiguration (TA2)

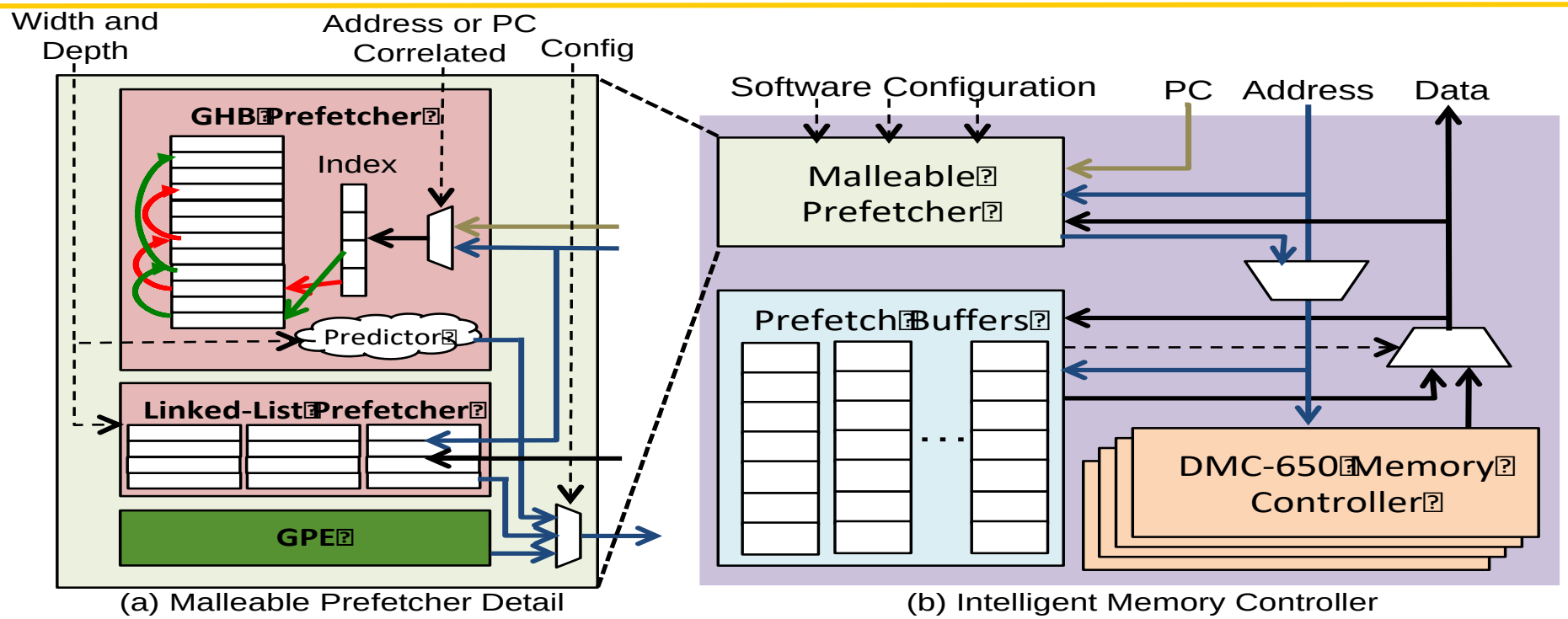


Reconfigurable Scratchpads/Caches

- Clusters of computational elements will share connections to configurable on-chip memory elements
- Reconfiguration can be used to dynamically adjust the ratio of on-chip memory to compute based on performance monitoring of memory access statistics
- Each memory can be configured to be either Cache based or scratchpad depending on the needs of the software

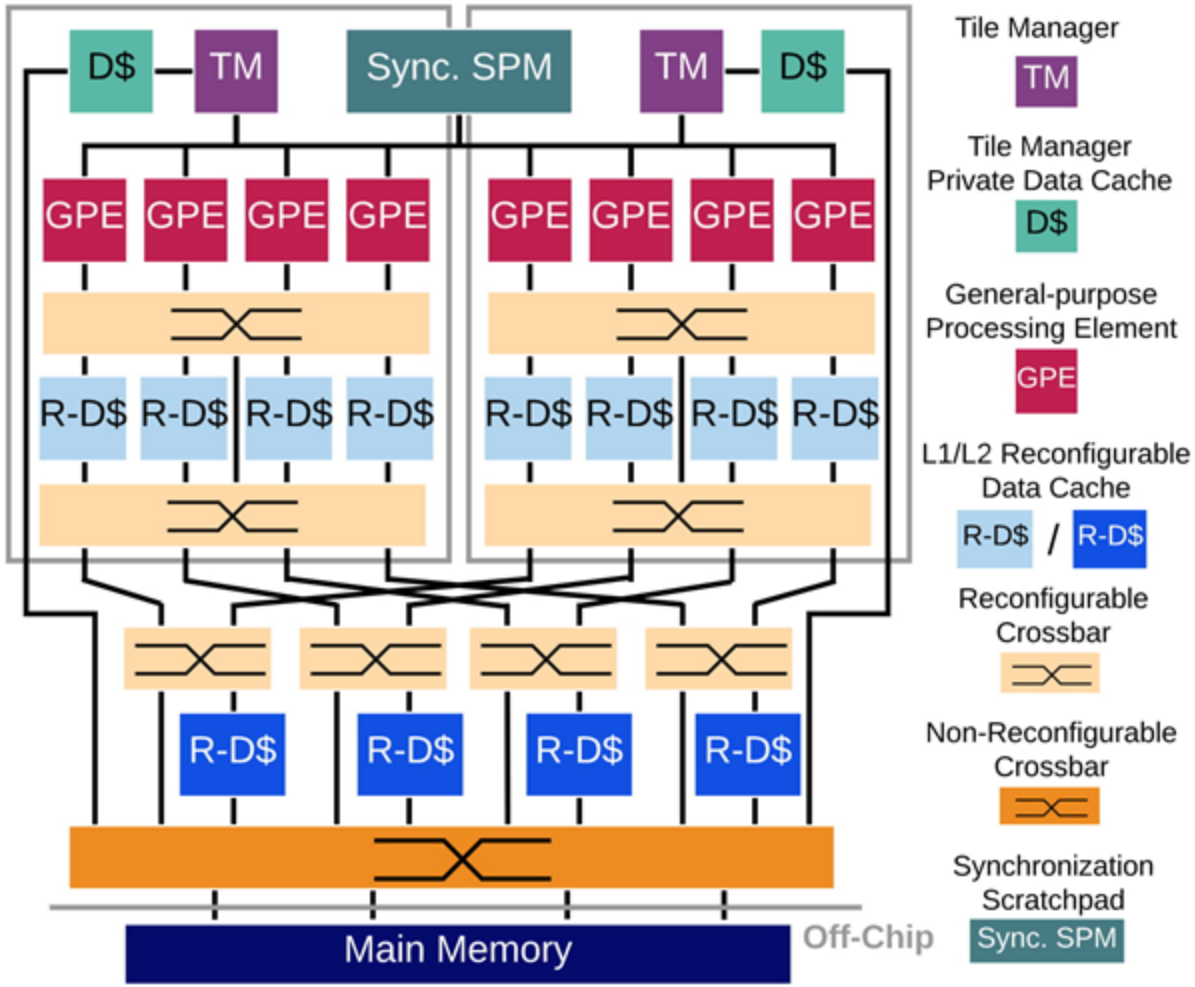


Intelligent Memory Controller



- Memory controller/Prefetcher provides a wide range of prefetching options and configurations
  - GHB Prefetcher for address correlating or PC correlating
  - Linked-list prefetcher
  - General purpose core to support decoupled access/execute style models and data transformations at the memory controller
- Configurable width and depth provides maximum flexibility in the design

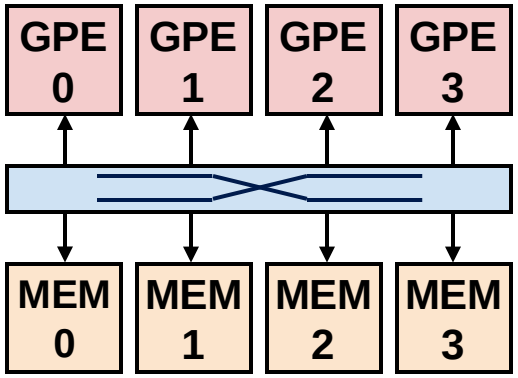
Transmuter



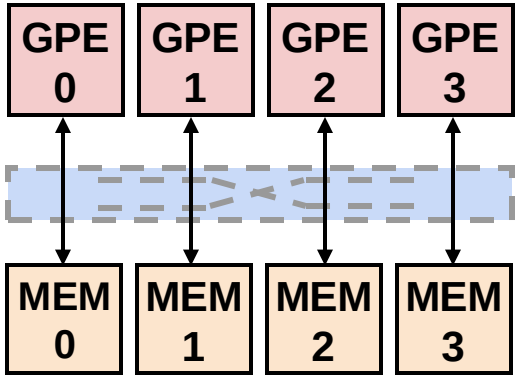
Distinct Features

- **Scalable, low-overhead tiled design**
  - Efficient in-order cores to maximize compute density
- **Reconfigurable caches**
  - Select cache or scratchpad to tailor to workload
- **Software managed scratchpads**
  - Facilitate low-overhead synchronization and storage of persistent shared data
- **Reconfigurable interconnect**
  - Provide shared or private access to R-caches
    - Optimize data reuse and cache line aliasing
    - Eliminate arbitration penalties
  - State machine-driven reconfiguration for *logical* modes, in particular systolic array

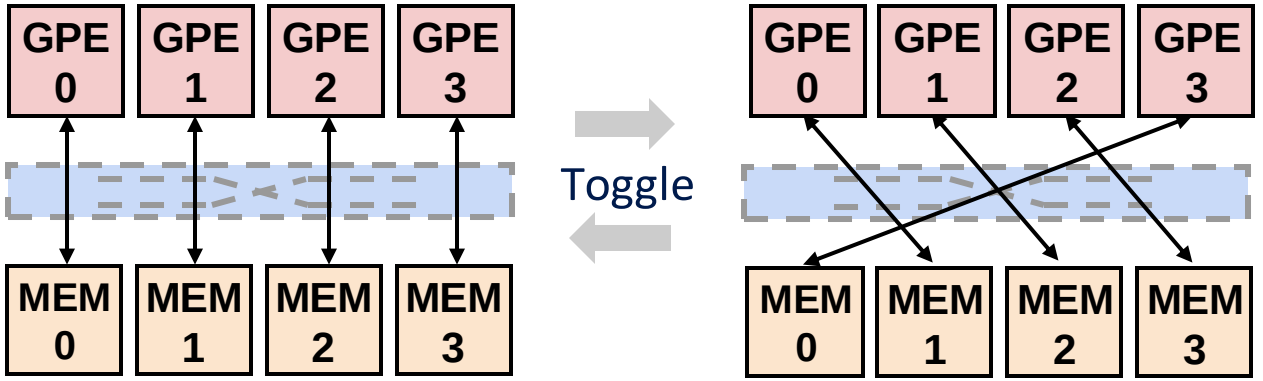
Crossbar and Memory Reconfiguration



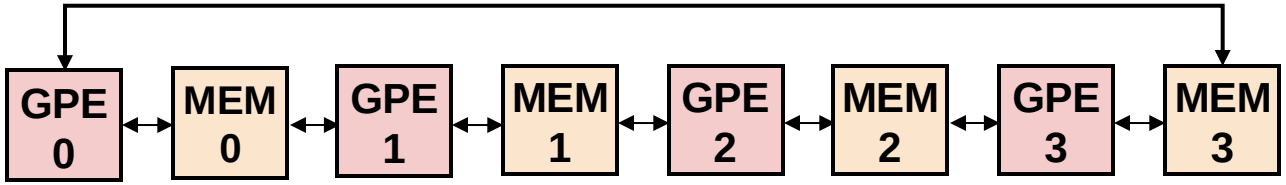
Cache Mode



Scratchpad Mode



Physical Configuration



Logical Configuration

Systolic Array

Programming Support

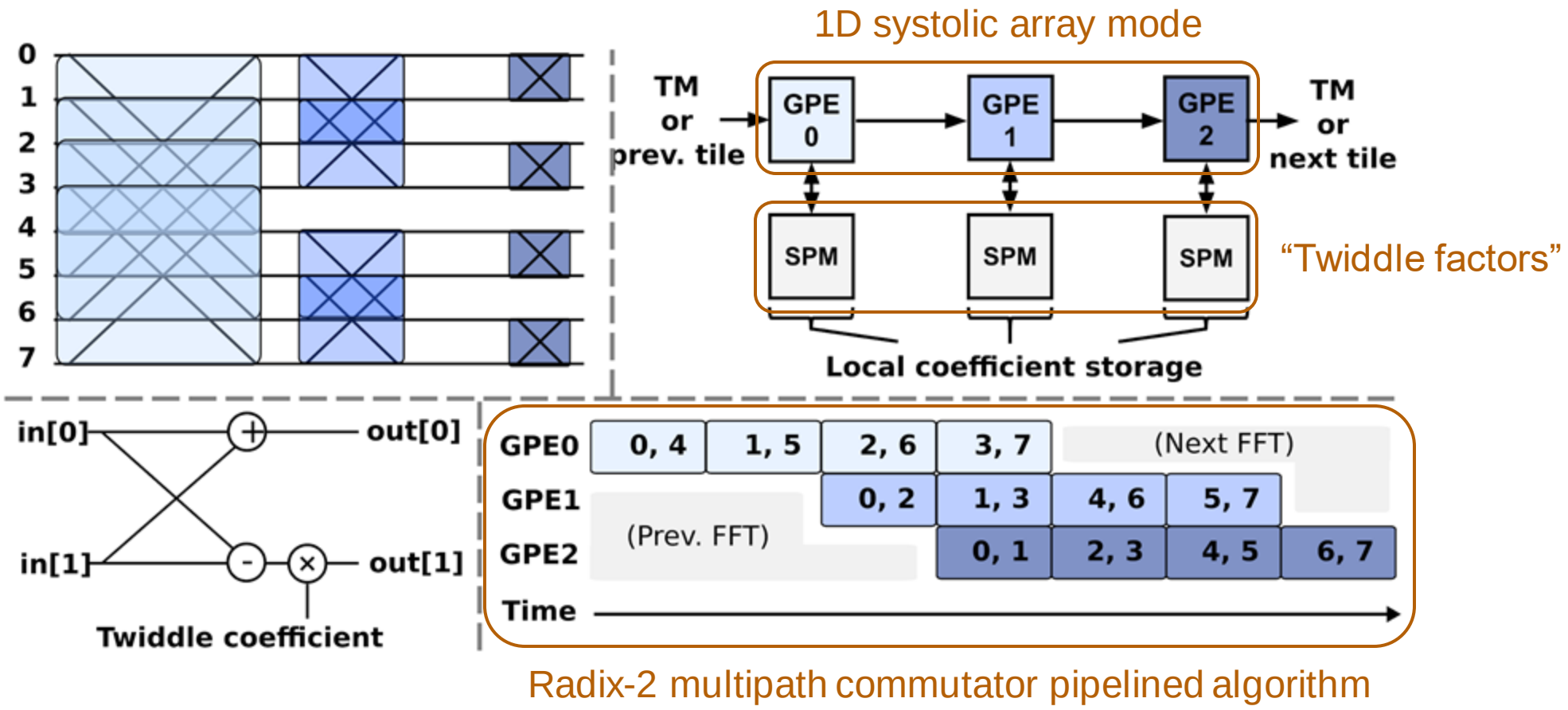
- High-level languages (e.g., C/C++) or assembly
- Custom API's for Transmuter-specific functionality

GPE/TM API Method	Description
`_trans_ld_word(addr)`	Load a word of data from scratchpad memory
`_trans_st_word(addr, val)`	Store a word of data into scratchpad memory
`_trans_sysarr_pop()`	Pop data from neighboring GPE in systolic mode
`_trans_sysarr_push(dir, val)`	Push data to neighboring GPE in systolic mode
`_trans_+q_push(val, [GPEID])`	Push data to work/status queue [blocks if full]
`_trans_+q_full([GPEID])`	Check if the work/status queue is full or not
`_trans_+q_pop([GPEID])`	Pop data from work/status queue [blocks if empty]
`_trans_+q_empty([GPEID])`	Check if the work/status queue is empty or not
`_trans_free_workq_push(val)`	Pushes to the work queue of the next free GPE
`_trans_l1_flush(bank)`	Flushes dirty data from L1 R-DCache[bank] to the L2
`_trans_l2_flush(bank)`	Flushes dirty data from L2 R-DCache[bank] to HBM
`_trans_spm_top(level, bank)`	Get a pointer to top of L[level] R-Dcache[bank] SPM
`_trans_reconfig()`	Wait to synchronize with host and reconfigure

***: GPE/TM +: work/status []: required when called by GPE**

Transmuter API's for cache control, reconfiguration, and custom data movement

Benefits of Reconfiguration - FFT Case Study



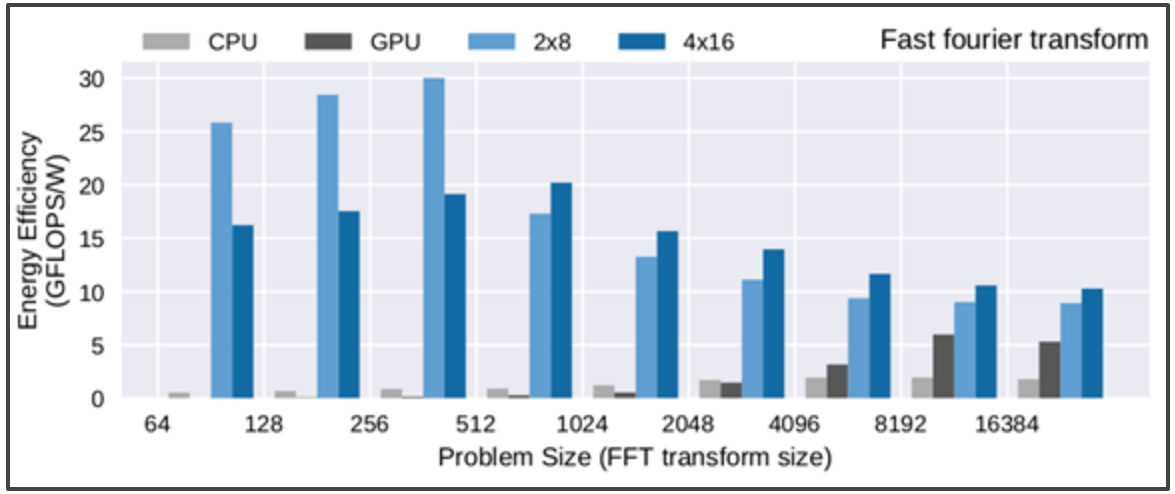
Radix-2 multipath commutator pipelined algorithm

Benefits of Reconfiguration - FFT Case Study

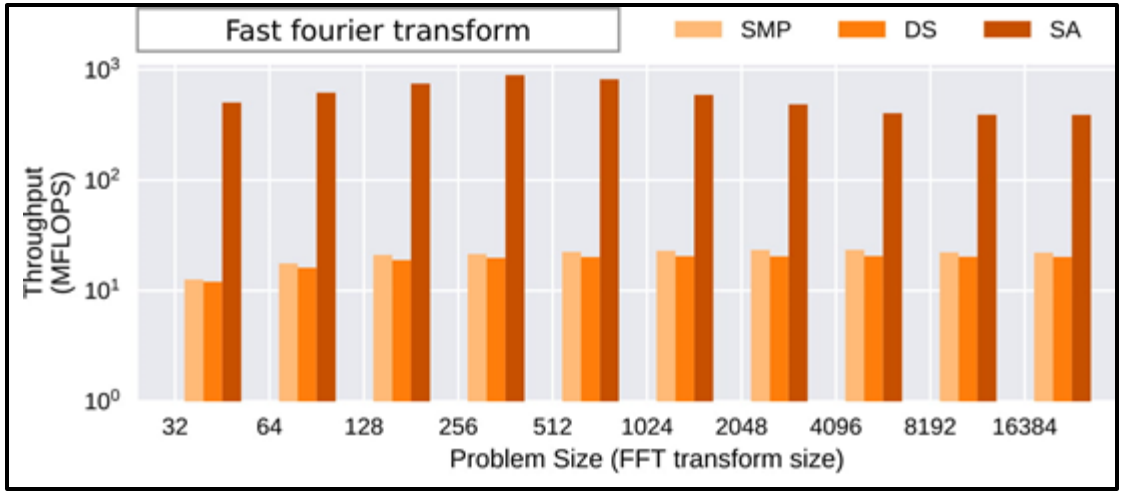
Energy (left fig.) and mode comparison (right fig.)

- Systolic array: SA
- L1 private-cache + L2 private SPM: DS *(data-streaming)*
- L1 private cache + L2 shared cache: SMP *(symmetric multi-processor)*

Energy Efficiency v.s. CPU/GPU: 12x/26x (geomean)



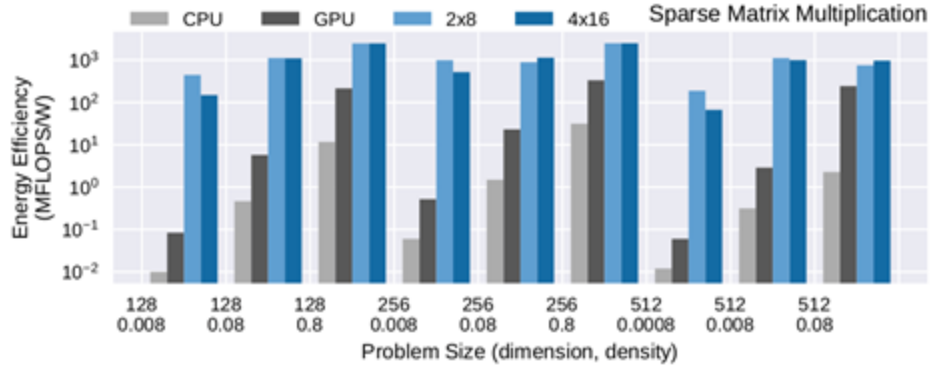
Throughput Improvement of SA: 29x (geomean)



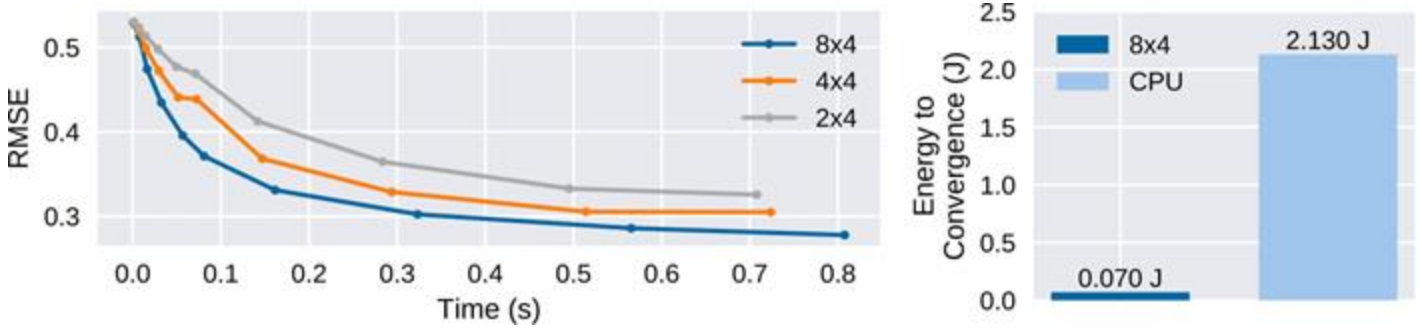
Wins from:
deterministic data reuse (e.g., twiddles), elimination of traditional multithreaded synchronization, and high utilization from the pipelined FFT algorithm.

Glimpse of Other Experiments

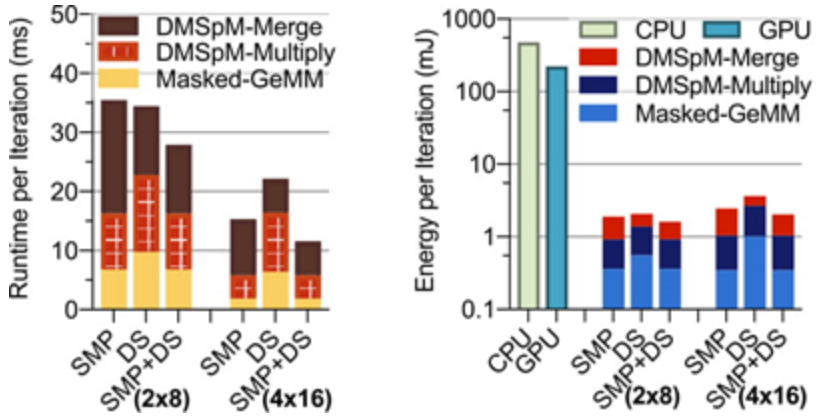
Sparse Matrix Multiplication (SpGeMM)



NOMAD [*]
(matrix completion algorithm for recommender systems)



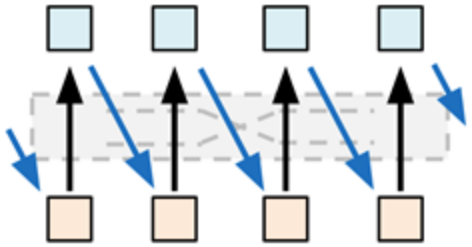
Sinkhorn [**]
(machine learning algorithm for document similarity analysis)



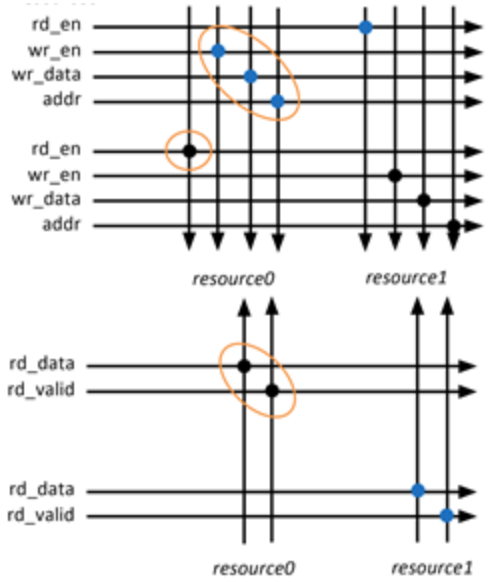
[*] H. Yun et. al, "NOMAD: non-locking, stochastic multi-machine algorithm for asynchronous and decentralized matrix completion," arXiv, 1312.0193, 2013
[**]: M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," NIPS, 2013

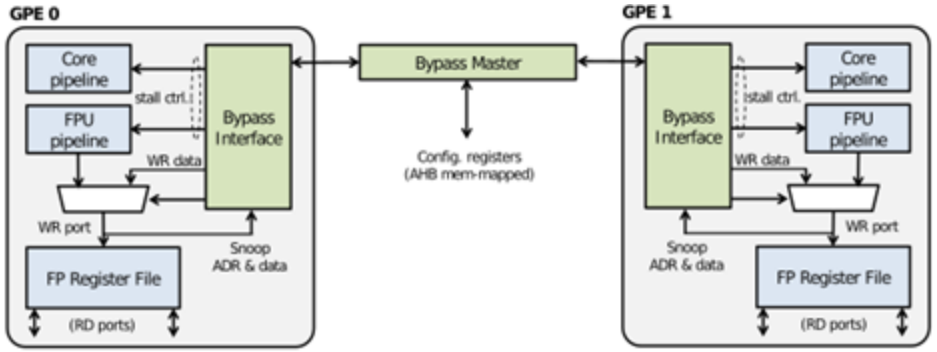
Ongoing Research

1. **Directly coupled core pipelines**
   - Augmenting general purpose cores with true systolic capability
2. **Streaming crossbars**
   - High bandwidth low latency data movement between buffers
3. **Chip development and fabrication**
   - Implemented a 32 core prototype in TSMC 28 nm process



Reorganizing crosspoints for streaming crossbars





Register-to-register tunneling between FPU's

Conclusion

Transmuter is a **programmable many-core accelerator** that achieves performance and energy efficiency through **introspection** and **reconfiguration**

Performance gains are primarily achieved with **reconfigurable caches and crossbars**, which facilitate **optimized data reuse**, and **high throughput data movement**

We demonstrate **energy and throughput improvements over modern CPU and GPUs up to 290x**, on a variety of signal processing and machine learning benchmarks

Thanks!