Acamar: A Dynamically Reconfigurable Scientific Computing Accelerator for Robust Convergence and Minimal Resource Underutilization

Ubaid Bakhtiar, Helya Hosseini and Bahar Asgari
University of Maryland College Park, Department of Computer Science

Introduction

- Modern supercomputers operate at <5% peak performance on real-world scientific tasks revealing significant underutilization.
- Domain-specific architectures (DSAs) have been explored to boost scientific computing efficiency
- Iterative solvers are used to tackle diverse scientific computing problems with varying structural characteristics in matrices.

$$A\vec{x} = \vec{b}$$

representation

$\vec{x}$: Solution Vector
$\vec{b}$: Constant Vector
A : Coefficient Matrix

Challenges

Solution Divergence: Not every solver guarantee convergence for all types of coefficient matrices. The convergence criteria for the solvers used in Acamar are:

- **JB:** Strictly diagonally dominant coefficient matrix A.
- **CG:** Symmetric and positive-definite coefficient matrix A.
- **BiCG-STAB:** Non-symmetric coefficient matrix A.

Resource Underutilization (R.U): SpMV operations results in uneven resource utilization due to the irregular distribution of non-zero values, leading to sub-optimal performance.

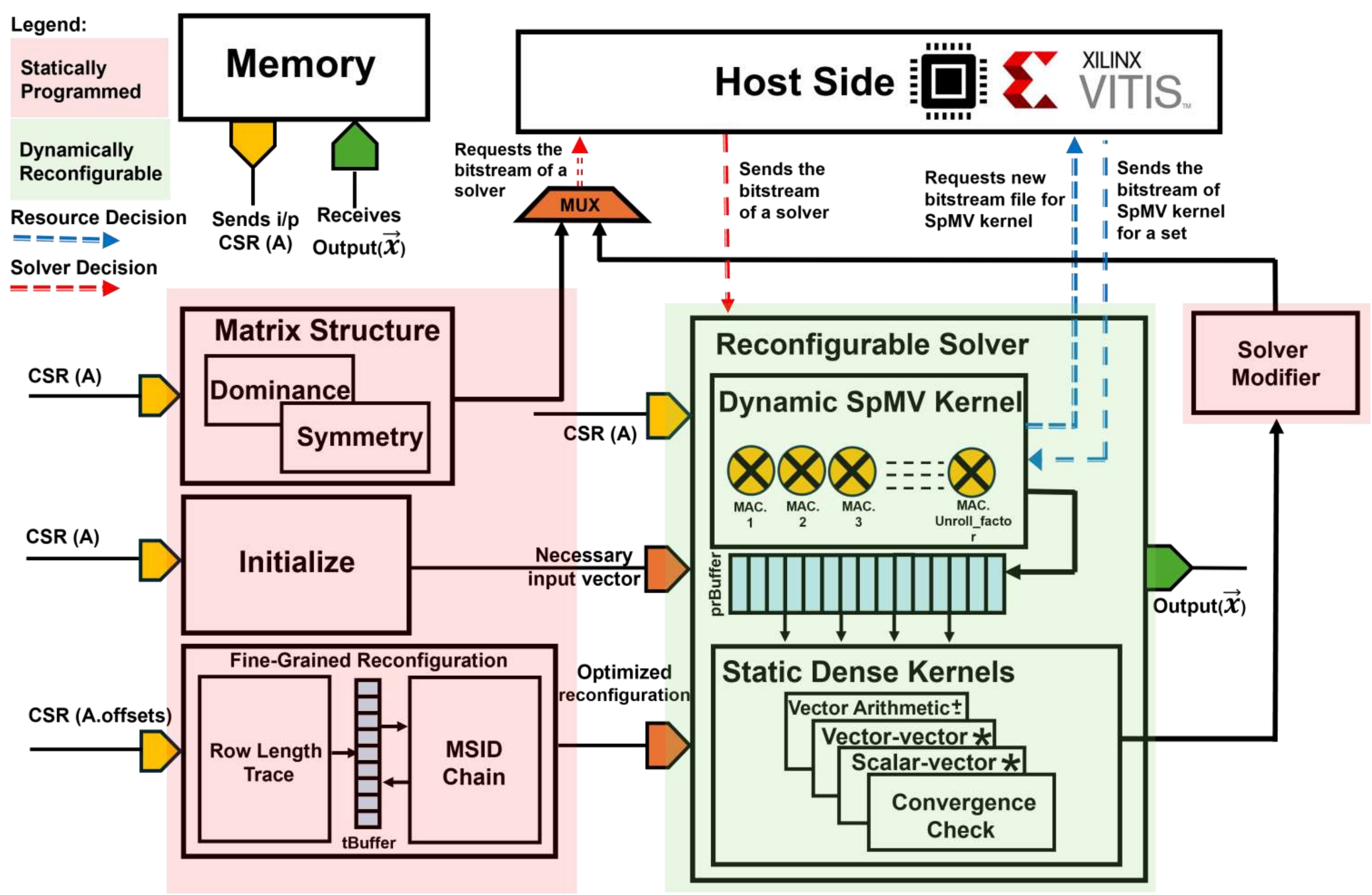
State-of-the-art accelerators assume that
- A given solver is suitable for all types of coefficient matrices.
- SpMV is uniformly efficient.

$\Big\}$ Unreliable Assumptions

Key insight

Acamar *is an FPGA-based accelerator that is* ***dynamically reconfigured*** *to match the computations required by the solver suitable for a given coefficient matrix offering* ***a robust convergence*** *for diverse datasets.*

Acamar



Acamar has two fundamental architectural categories based on how they map on FPGA:

Statically Programmed:

Matrix Structure Unit: Selects the most suitable solver based on the structural properties of the coefficient matrix.

Initialize Unit: Executes the pre-loop instructions of the solver.

Fine-grained Reconfigurable Unit:

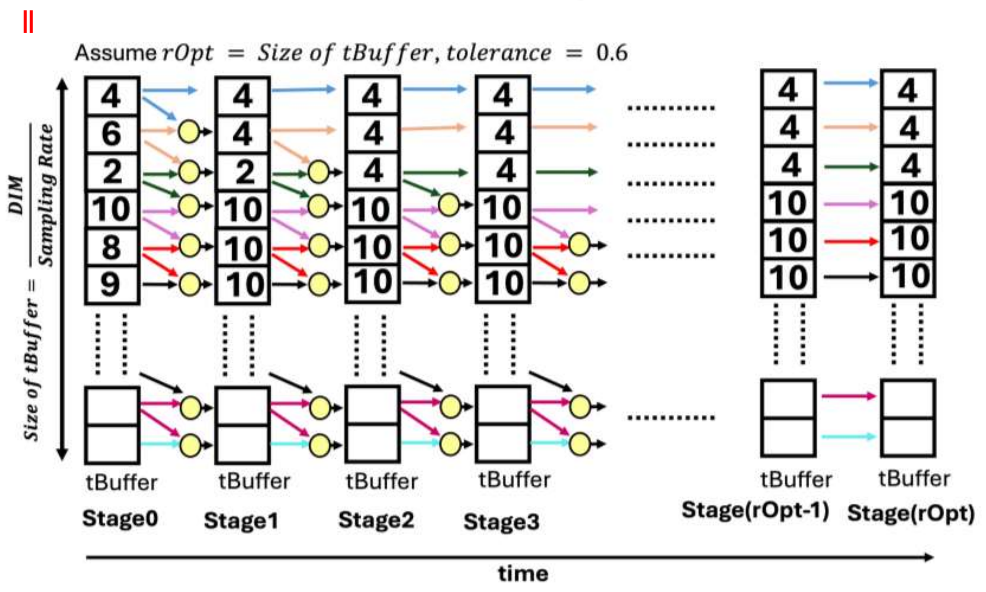
- ***Row-Length Trace Unit:*** Optimizes resource allocation for sparse computation units by adjusting the unroll factor based on average row length
- ***MSID-Chain Unit:*** Updates the unroll factor only when successive values differ beyond a set tolerance

Solver Modifier: In case of divergence, this unit activates and selects the appropriate solver for reconfiguring the Reconfigurable Solver unit.

Dynamically Reconfigurable:

Reconfigurable Solver unit: Configures one of three solvers based on matrix structure analysis, reconfiguring the ***Dynamic SpMV kernel*** as needed to optimize resource utilization for sparse computations.

- Continues processing until convergence, when the solution is stored or triggers the Solver Modifier if divergence occurs.



Experimental Setup

Simulation: Cycle-accurate simulator based on HLS implementation on AMD Xilinx Alveo U55c

Datasets: SuiteSparse Collection

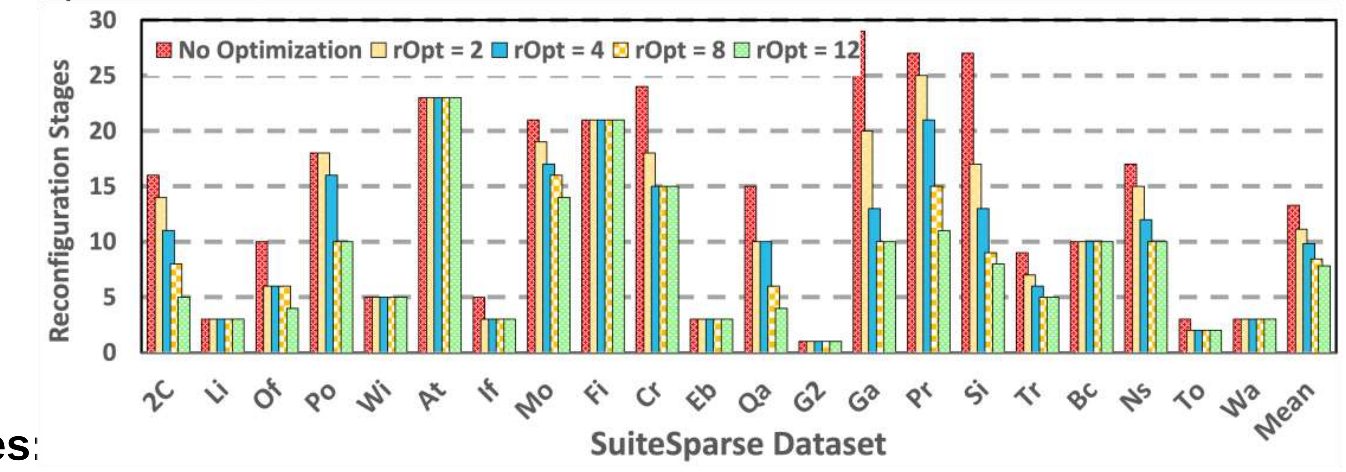
Setup Time: 200 Iterations

Computing Precision: 32bit

Convergence Threshold: 10^{-5}

Fine-Grained Reconfiguration Unit Parameters

- **rOpt:** Number of stages in MSID-chain. As shown in the figure below, the reconfiguration rate becomes constant after *rOpt = 8*
- **Sampling Rate:** Number of sets of rows in coefficient matrix A. It is set to 32 in our experiments
- **Tolerance:** Tunes the tolerance level of the MSID-chain unit (tolerance > 0.5 can result in a lower reconfiguration rate but possibly wasted resources). In our experiments, *tolerance = 0.15*.

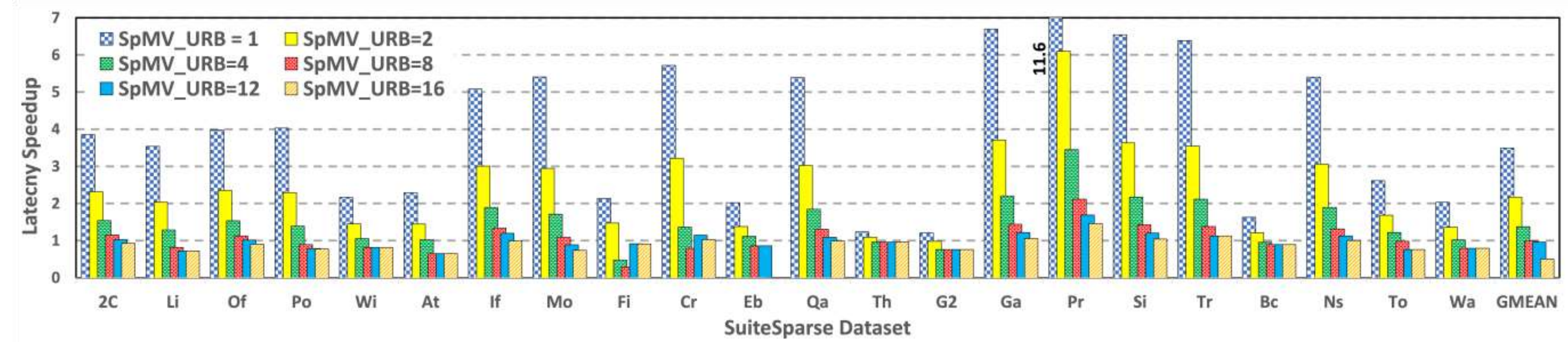


Baselines:

- A static design with fixed number of allocated resources
- SpMV implementation in the cuSparse library on Nvidia GTX 1650 Super running on Cuda v11.6. We used the Nvidia Nsight toolkit to run GPU evaluation.

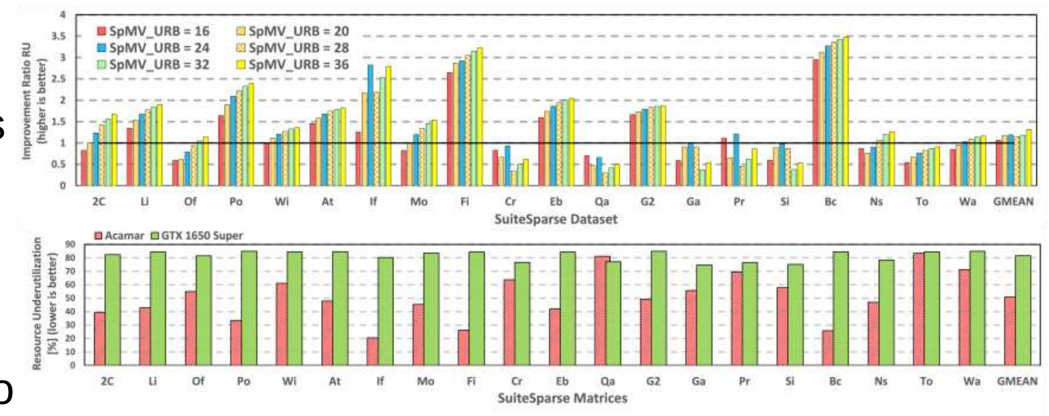
Evaluation

Speedup: 11.61× compared to a baseline implementation of SpMV with a single MAC unit. Improvements diminish as we allocate more resources in the baseline

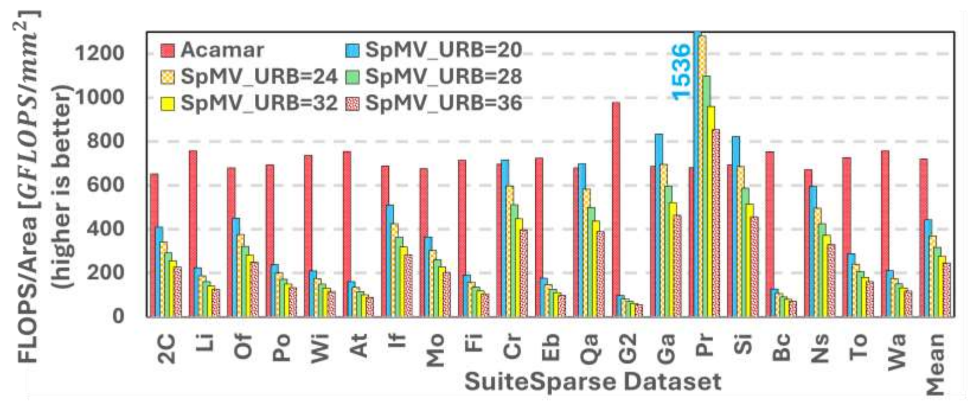


Resource Underutilization:

- Improve up to 3× in Acamar as compared to the static design of the baseline.
- On average Acamar is underutilized 50% compared to 81% underutilized GPU



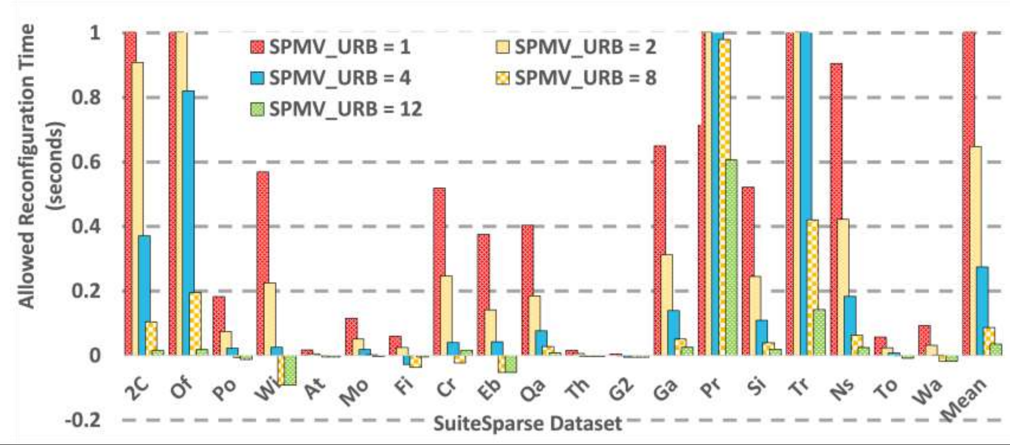
Performance Efficiency:



- Measured as number of floating point operations (FLOPS) per square millimeter area of FPGA fabric
- On average , Acamar achieves 720 $GFLOPS/mm^2$ performance efficiency.

Allowed Reconfiguration Time:

- This figure shows the bounds within which the reconfiguration must be done to ensure that Acamar incurs the same or less latency as the baseline.



Conclusions

- This paper introduces Acamar, a dynamically reconfigurable FPGA-based accelerator designed for various scientific computing workloads, which overcomes the limitations of static designs by adapting to different coefficient matrix structures.
- By enabling seamless transitions between solvers like JB, CG, and BiCG-STAB, along with the optimized SpMV unit and an efficient MSID chain to reduce reconfiguration overhead, Acamar enhances resource utilization and represents a significant advancement in adaptive design space architectures for real-time scientific problem-solving.