Scratchy : A Class of Adaptable Architectures with Software-Managed Communication for Edge Streaming Application

Joseph W. Faye[1], Naouel Haggui[1], Florent Kermarrec[2], Kevin J.M Martin[3], Shuvra Bhattacharyya[1,4] , Jean-François Nezan[1], **Maxime Pelcat**[1]

[1] IETR – UMR CNRS 6164, INSA Rennes, France

[2] Enjoy-Digital, Landivisiau, France

[3]Université Bretagne Sud, Lab-STICC UMR 6285, Lorient, France

[4]University of Maryland, College Park, USA

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES RENNES

IETR

EnjoyDigital

Université Bretagne Sud ubs

UNIVERSITY OF MARYLAND 18 56

VAADER IETR research team

Outline

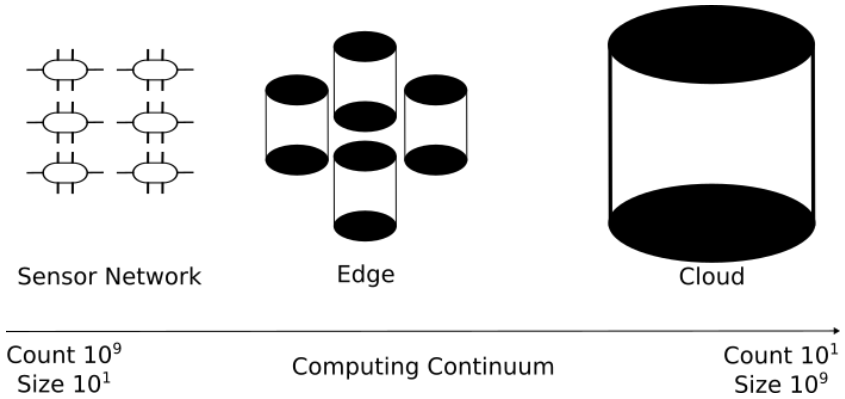
1- Scratchy: building an architecture for dataflow applications

2- Overview and principles of Scratchy

3- Design Space Exploration test case and results

4- Future work

Context

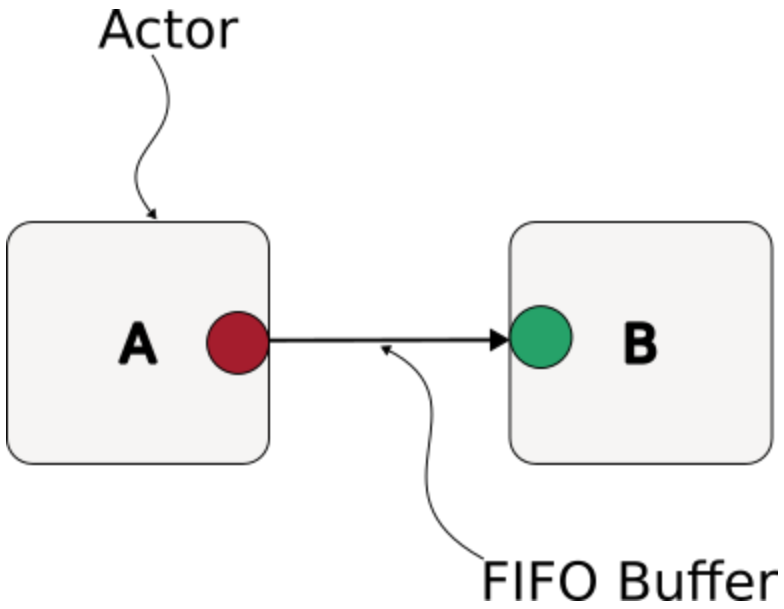


- Many edge devices process **data streams**
- Devices shall be software (re)programmable and upgradeable

→Looking for **stream-optimized, programmable architectures**

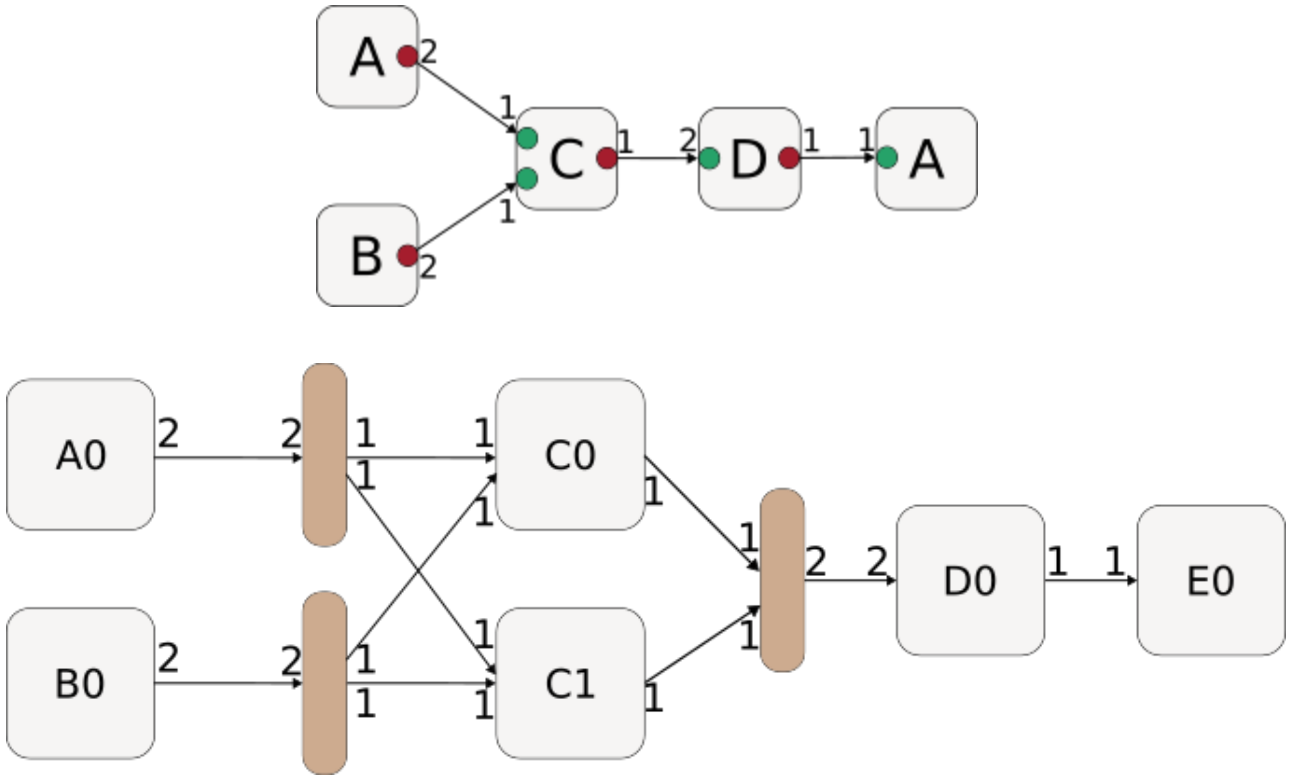
Dataflow Model Of Computation

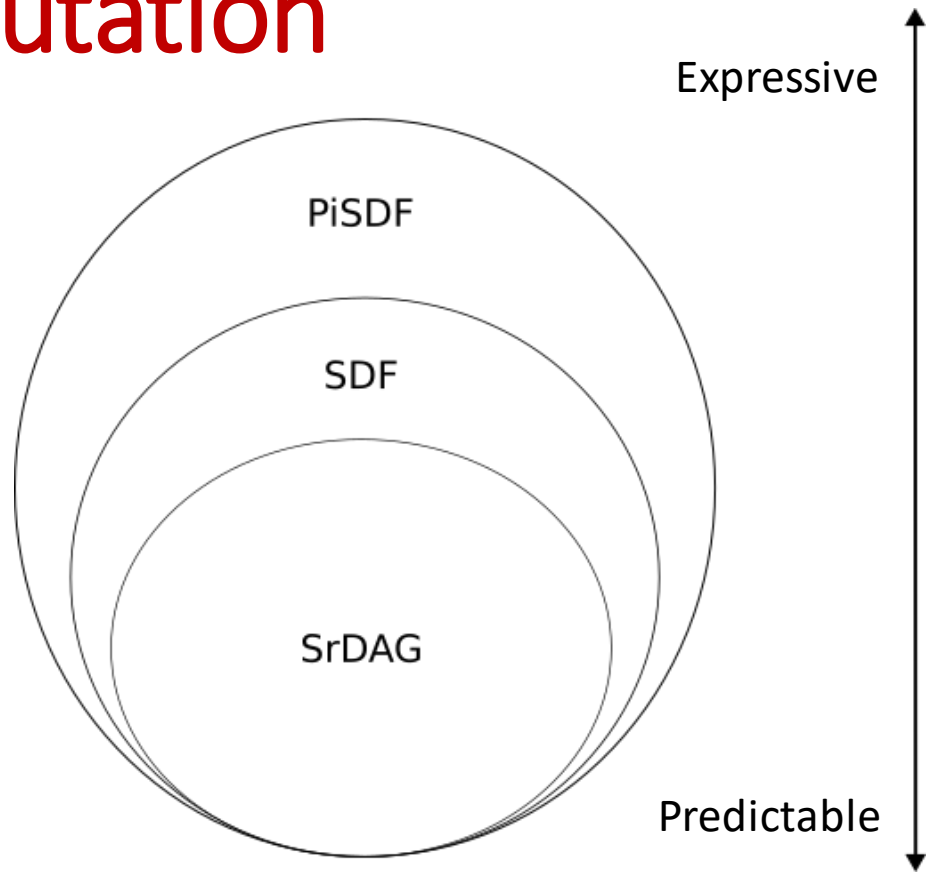
- Data Flow
  - Reveals **dependencies and parallelism** between **actors**
  - **Encapsulate states within actors**
  - Provides application abstraction for **resource allocation**



Lee, E. A., & Parks, T. M. (1995).
Dataflow process networks. Proceedings of the IEEE, 83(5), 773-801.

Model Of Computation





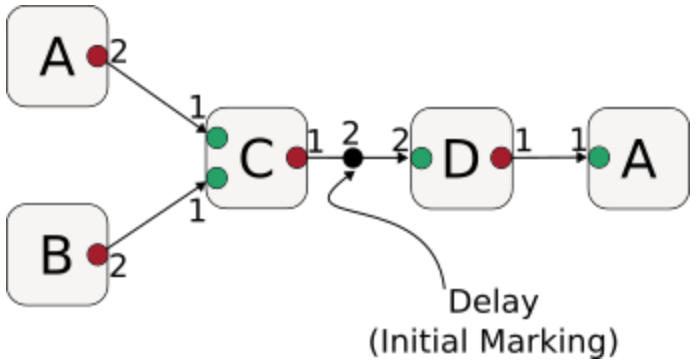
Desnos, K., Pelcat, M., Nezan, J. F., Bhattacharyya, S. S., & Aridhi, S. Pimm: Parameterized and interfaced dataflow meta-model for mpsocs runtime reconfiguration. In SAMOS 2013, IEEE.

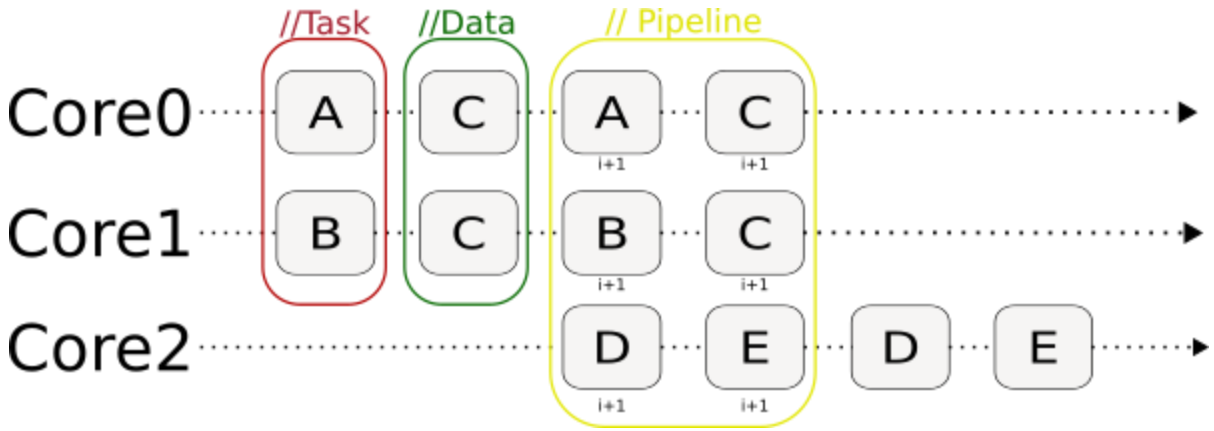
Lee, E. A., & Messerschmitt, D. G. (1987). Synchronous data flow. Proceedings of the IEEE, 75(9), 1235-1245.

Model Of Computation

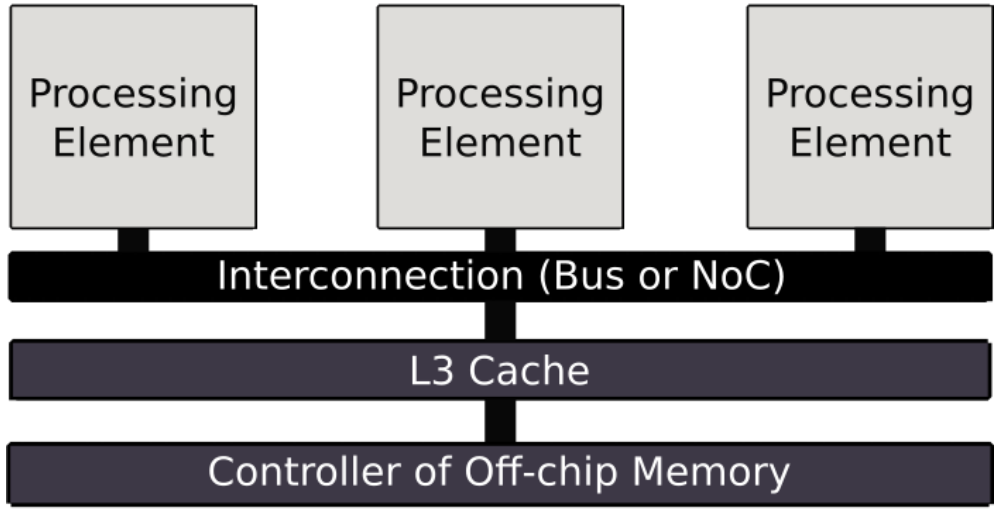
Synchronous Dataflow

- Expresses several types of **parallelism**
- Offers **predictability** of data movements
- Can ensure the **absence of deadlocks**





Context



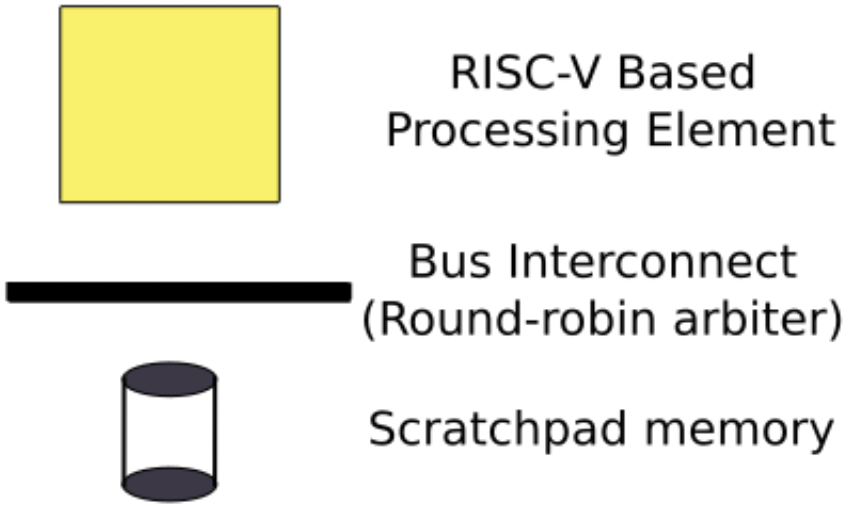
- SMP Multicore architectures show limits on supporting SDF-described applications and keeping the characteristics.
  - **High synchronization overhead**
  - **Bad cache usage**
- New open source SoC design tool LiteX

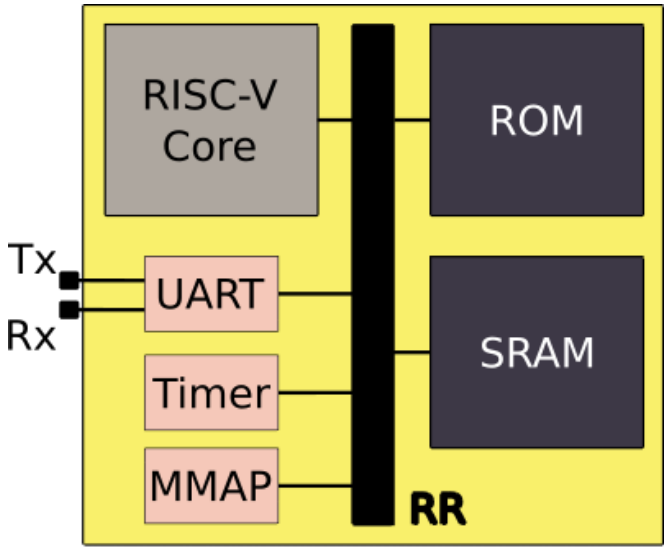
Ghasemi, A., Ruaro, M., Cataldo, R., Diguet, J. P., & Martin, K. J. (2022). The Impact of Cache and Dynamic Memory Management in Static Dataflow Applications. JSPS, 94(7), 721-738, Springer.

Paper Goal

- Propose a simple low-cost architecture with software-managed communication and scratchpads
- Compose RISC-V cores with configurable buses and scratchpads setups
- Can we generate some non-degenerated alternatives for a given dataflow application?

Scratchy Processing Element, based on LiteX

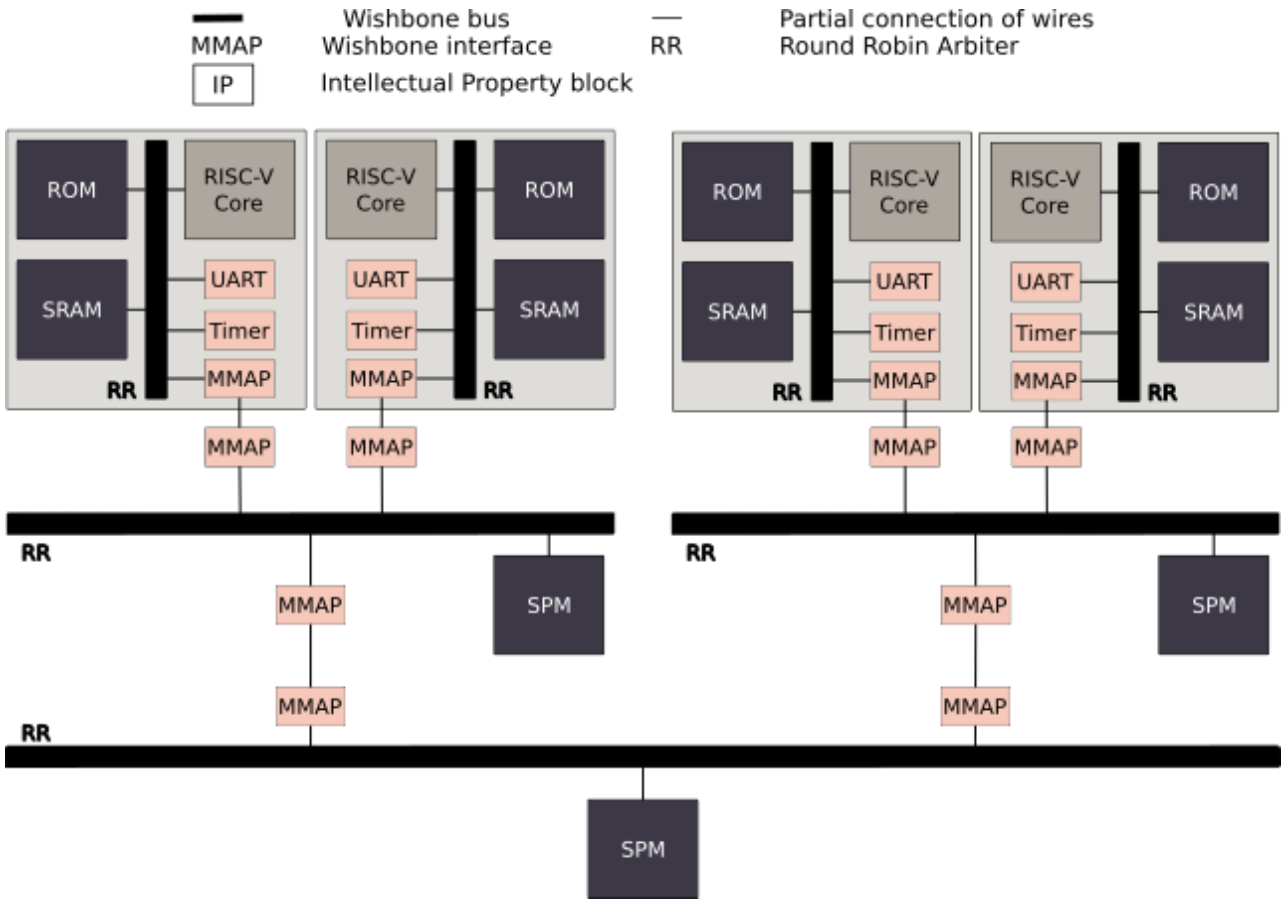




RR Round-Robin Arbiter

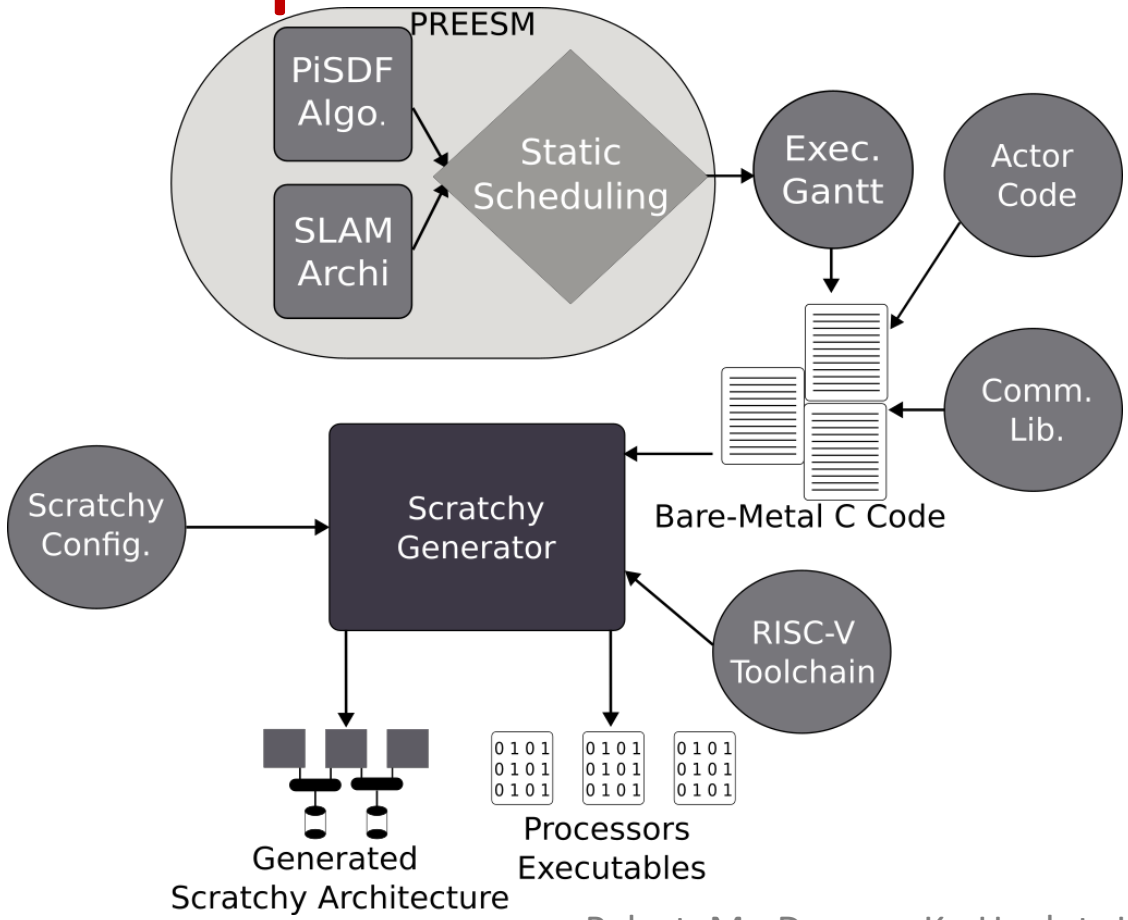
Kermarrec, F., Bourdeauducq, S., Lann, J. C. L., & Badier, H. (2020).
LiteX: an open-source SoC builder and library based on Migen Python DSL. arXiv preprint arXiv:2005.02506.

Overview Of Scratchy Architecture



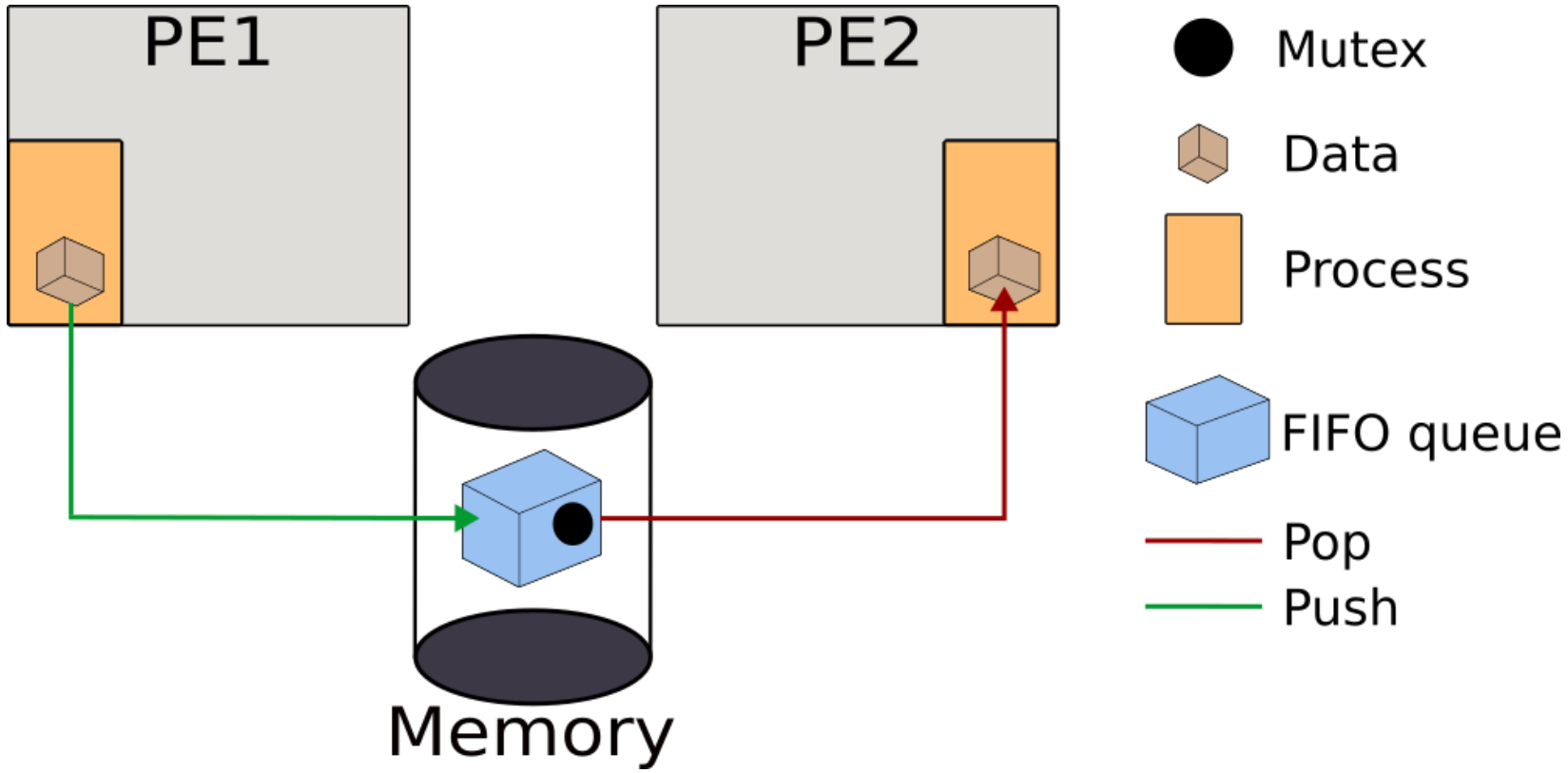
A 4-Core Scratchy Example Architecture

Scratchy Design Flow and Multicore Compilation Framework



Pelcat, M., Desnos, K., Heulot, J., Guy, C., Nezan, J. F., & Aridhi, S. (2014). Preesm: A dataflow-based rapid prototyping framework for simplifying multicore dsp programming. In 2014 EDERC. IEEE.

Scratchy Inter-PE Communication Principle



Design Space Exploration Test-Case

- Find a resource/throughput trade-off

Inputs :

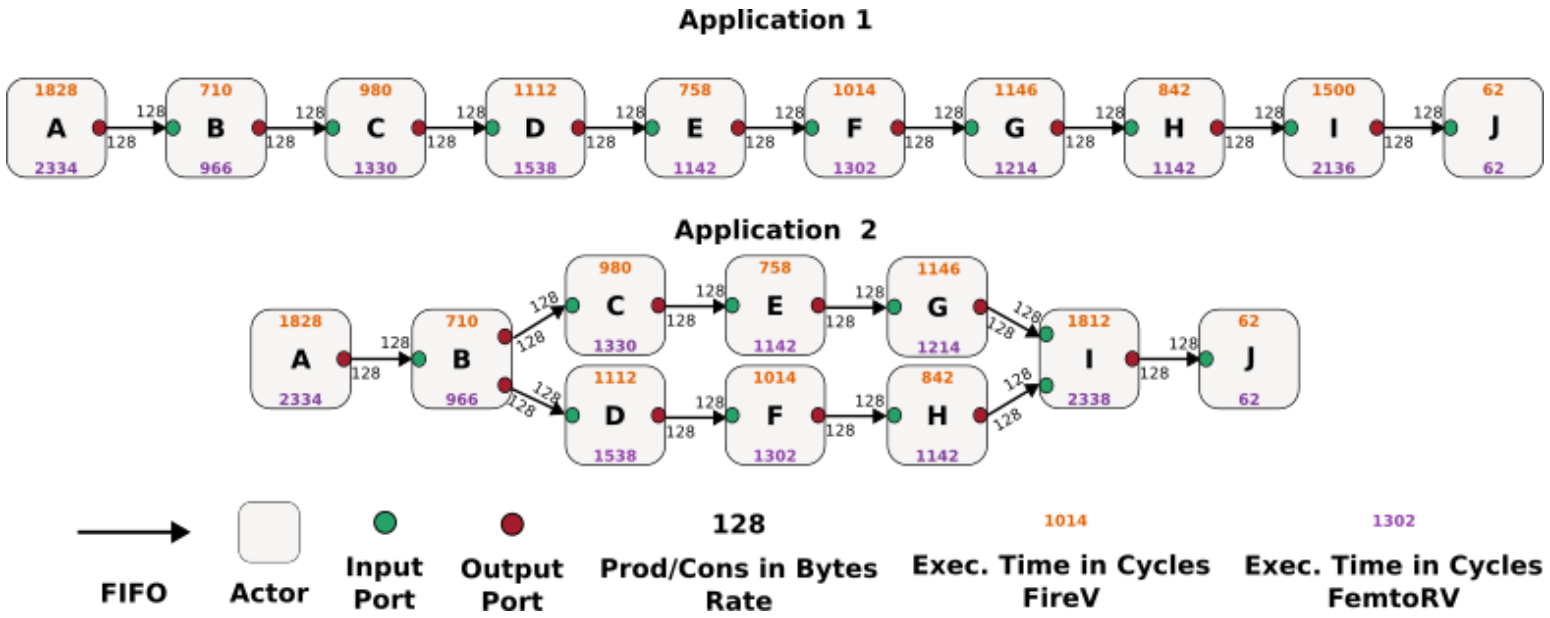
- Application graph
- Set of architecture topologies

Output :

- Pareto front

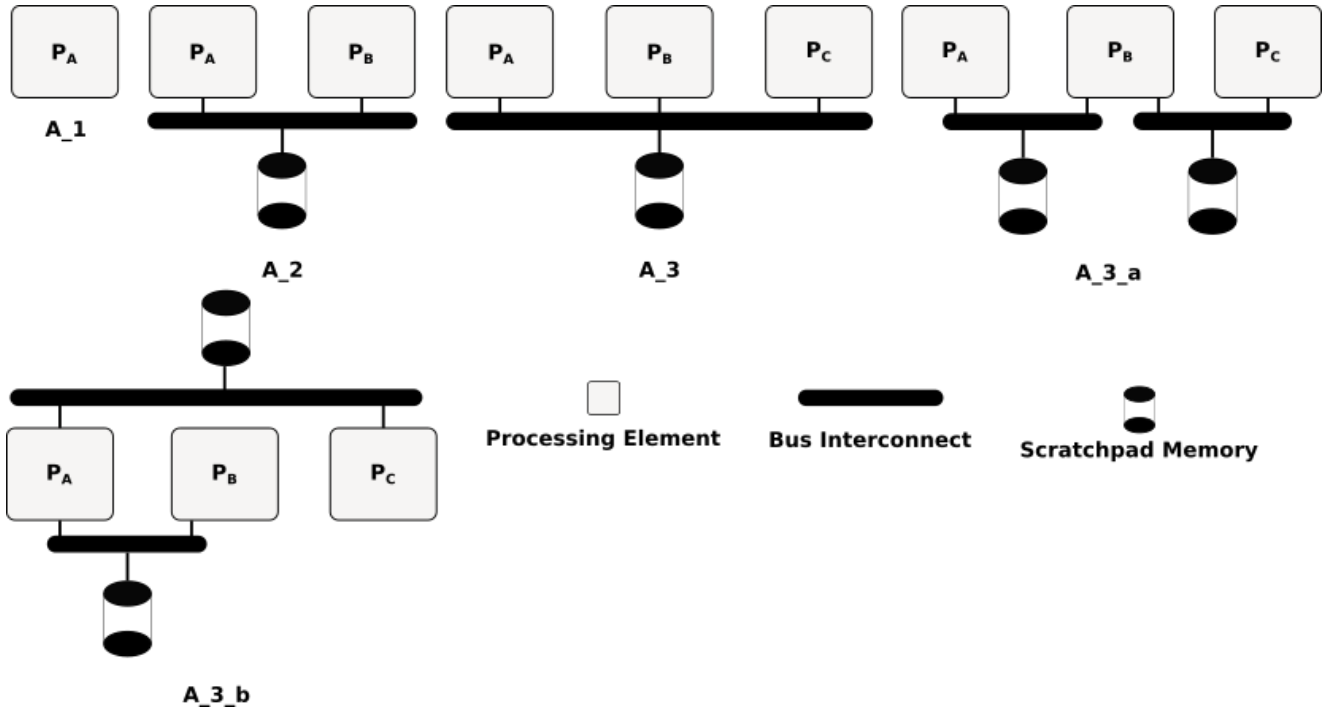
Design Space Exploration Test-Case

- Application Graphs



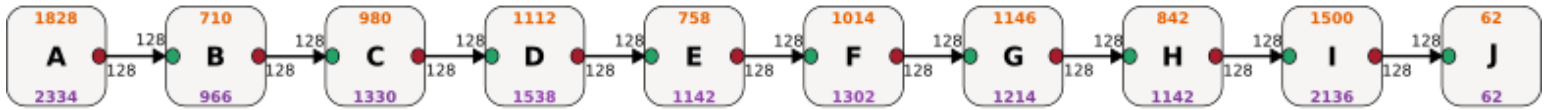
Design Space Exploration Test-Case

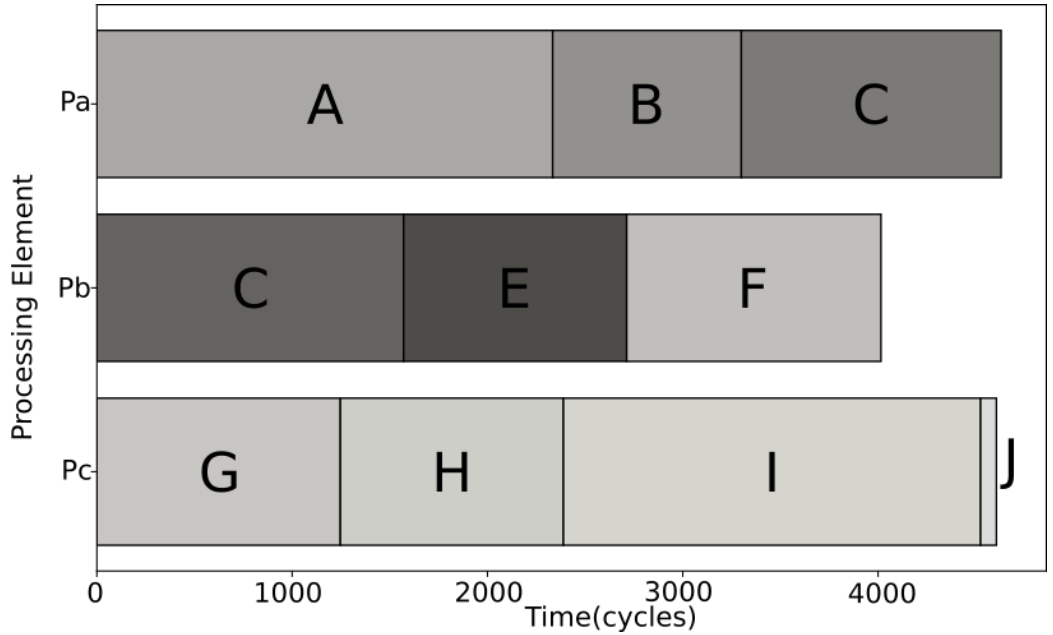
- 5 Architecture Topologies



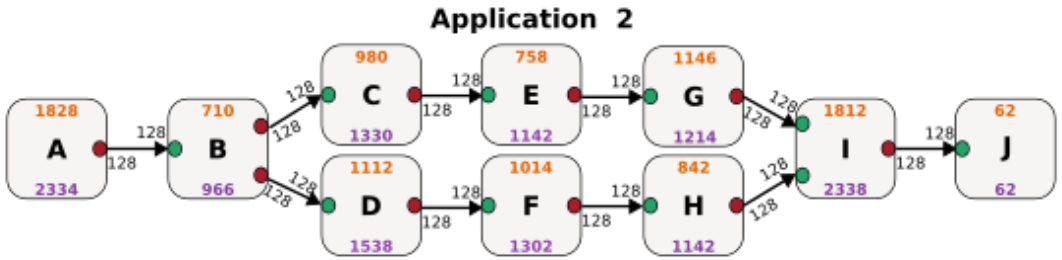
Design Space Exploration Test-Case

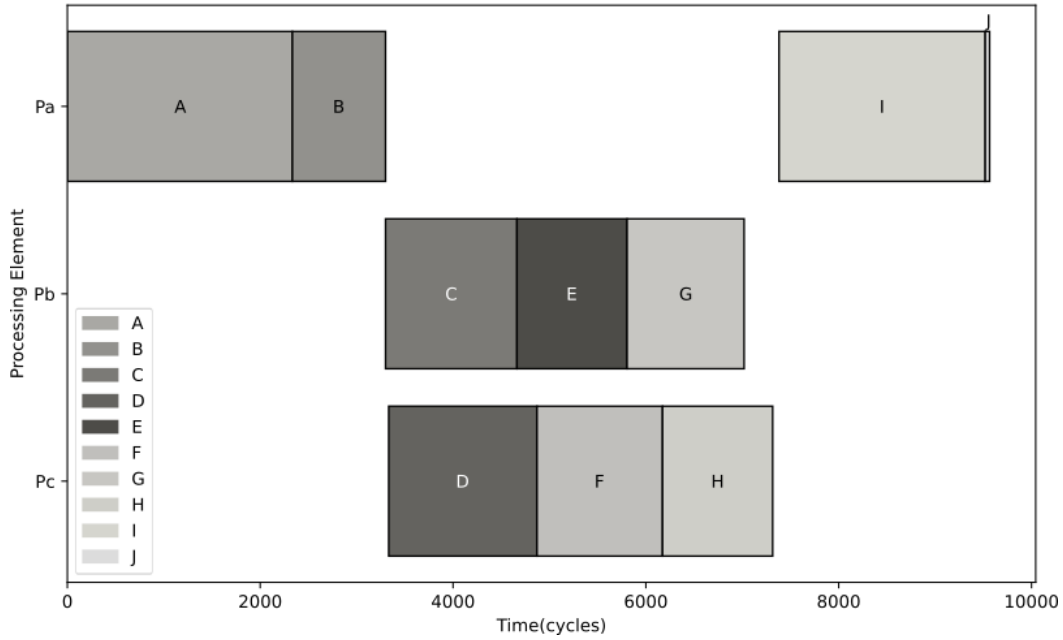
Application 1





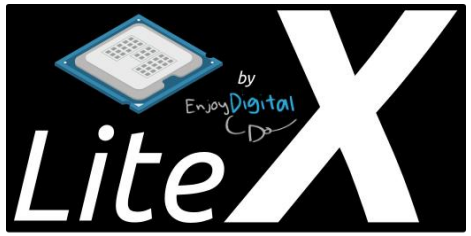
Design Space Exploration Test-Case





Design Space Exploration Test-Case

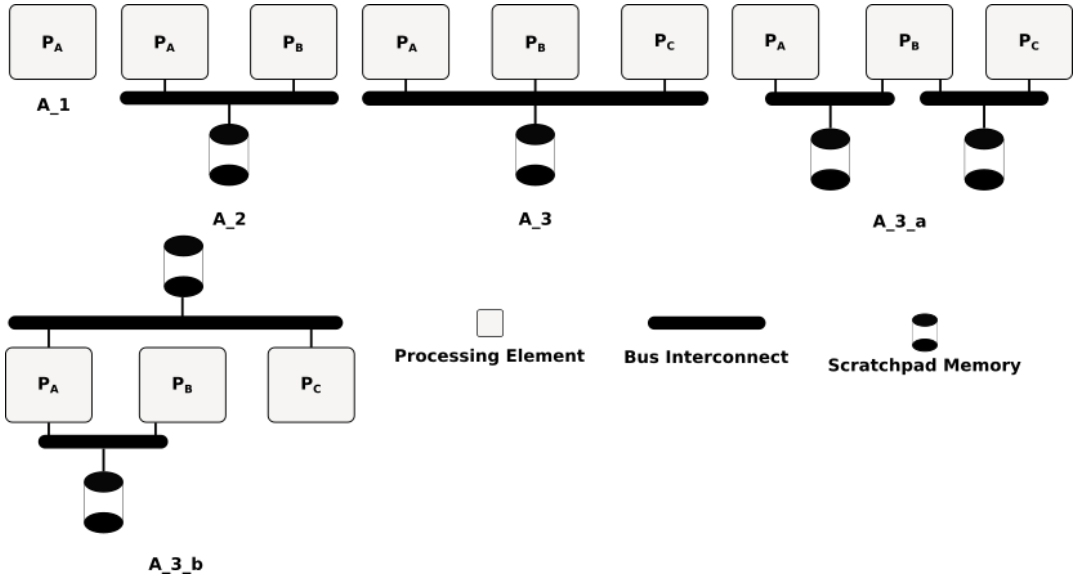
- Technical Setup :
  - Altera De10Lite (3 core maximum)
  - RTL and Software support implementation based on LiteX

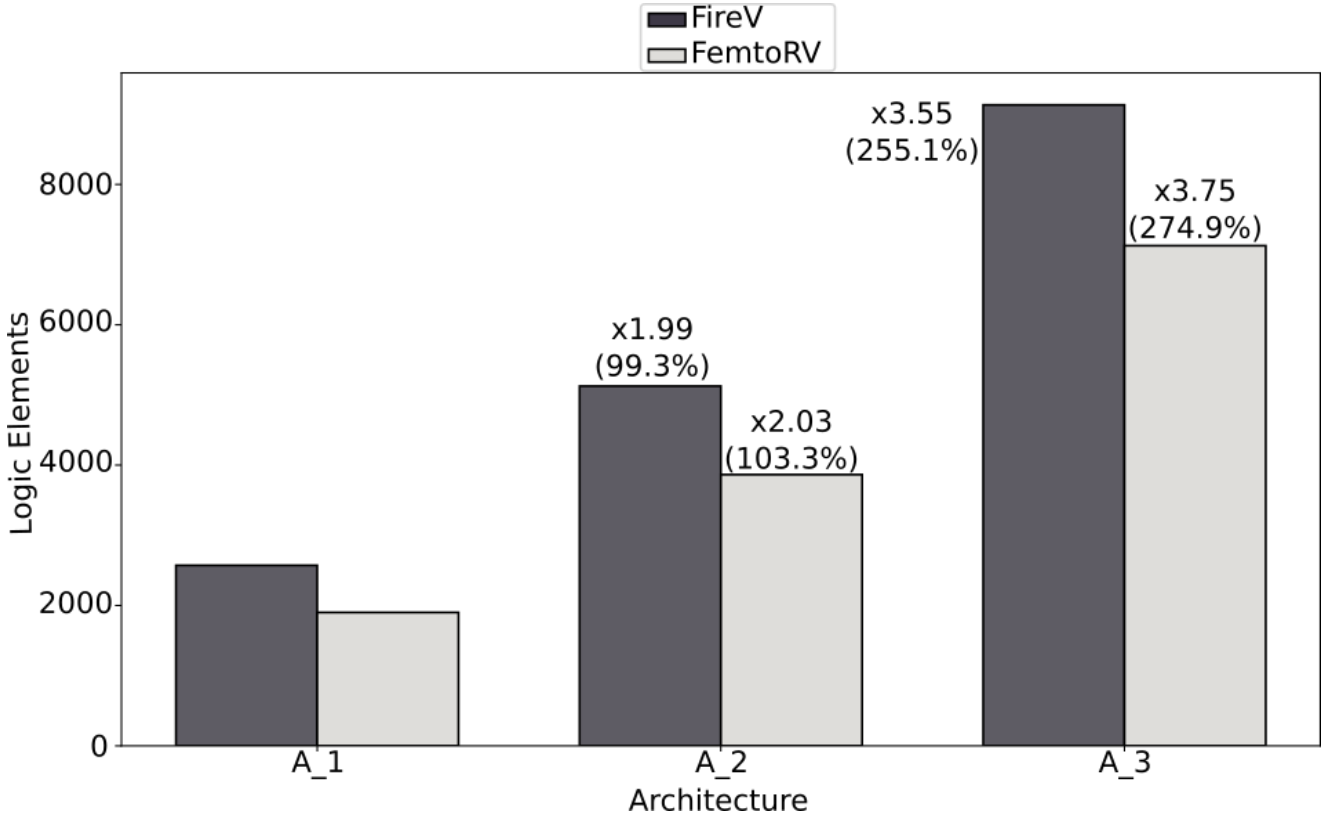


Build your hardware, easily!

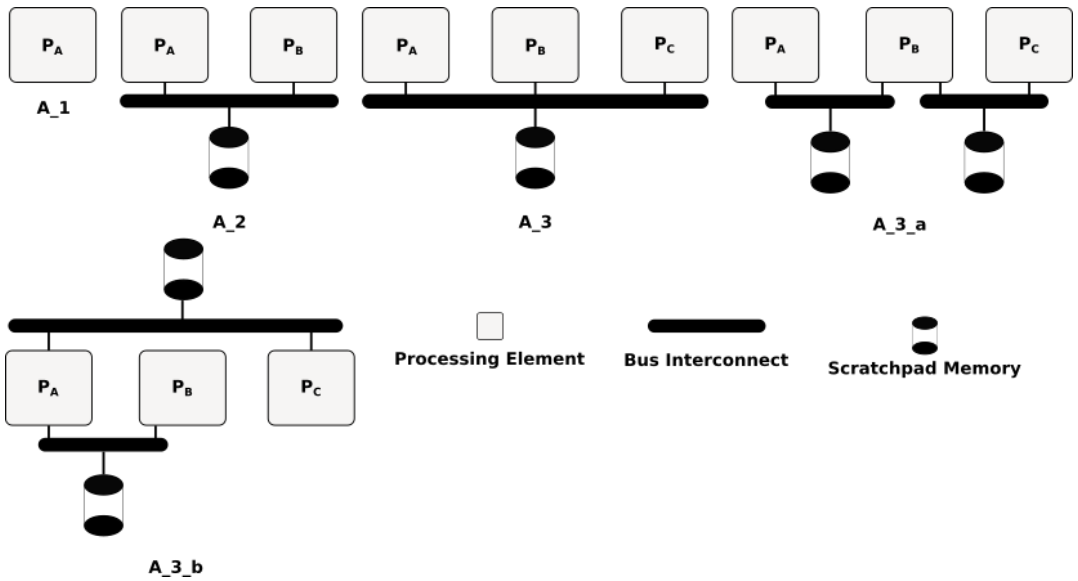
- **RISC-V Compile**r
- provides all the common components required to easily create an FPGA Core/SoC

From one to 3 cores

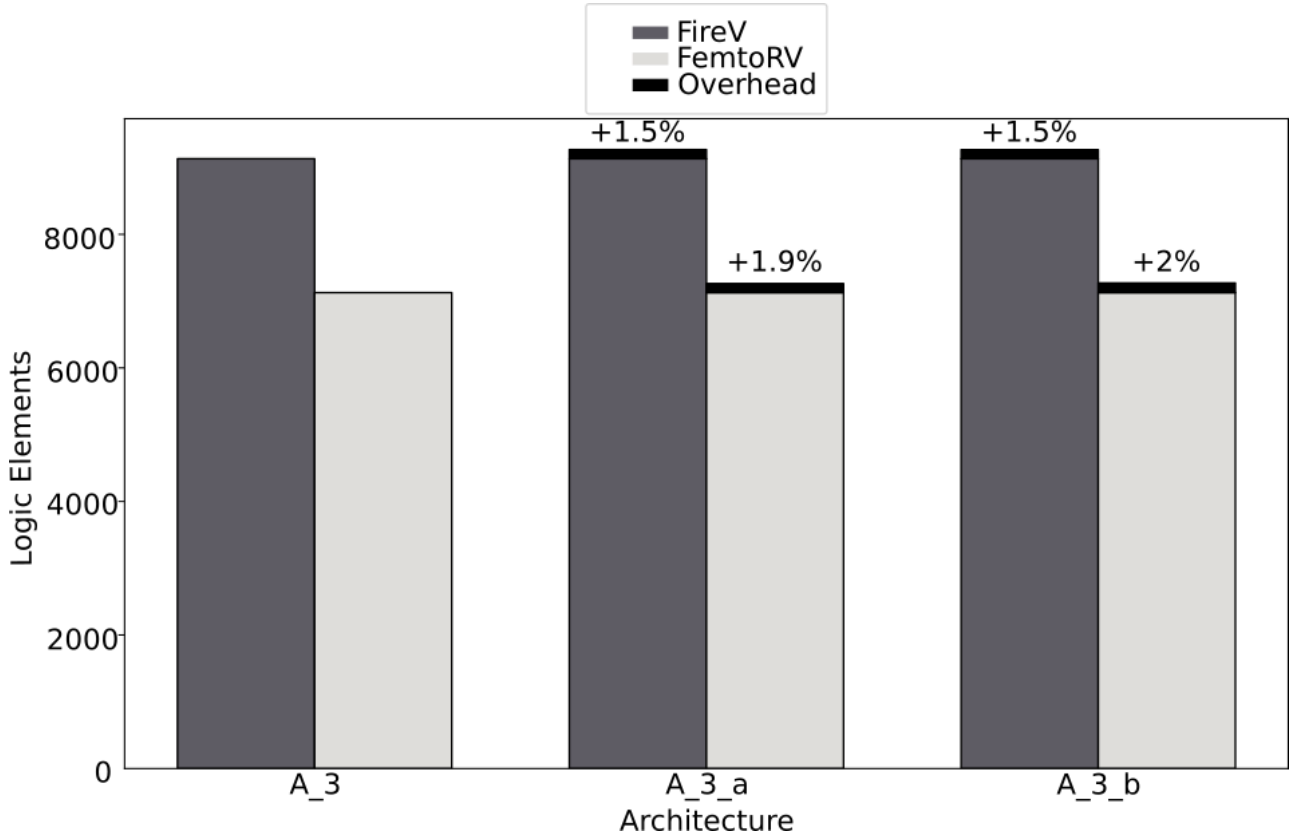




From one to multiple buses and scratchpads

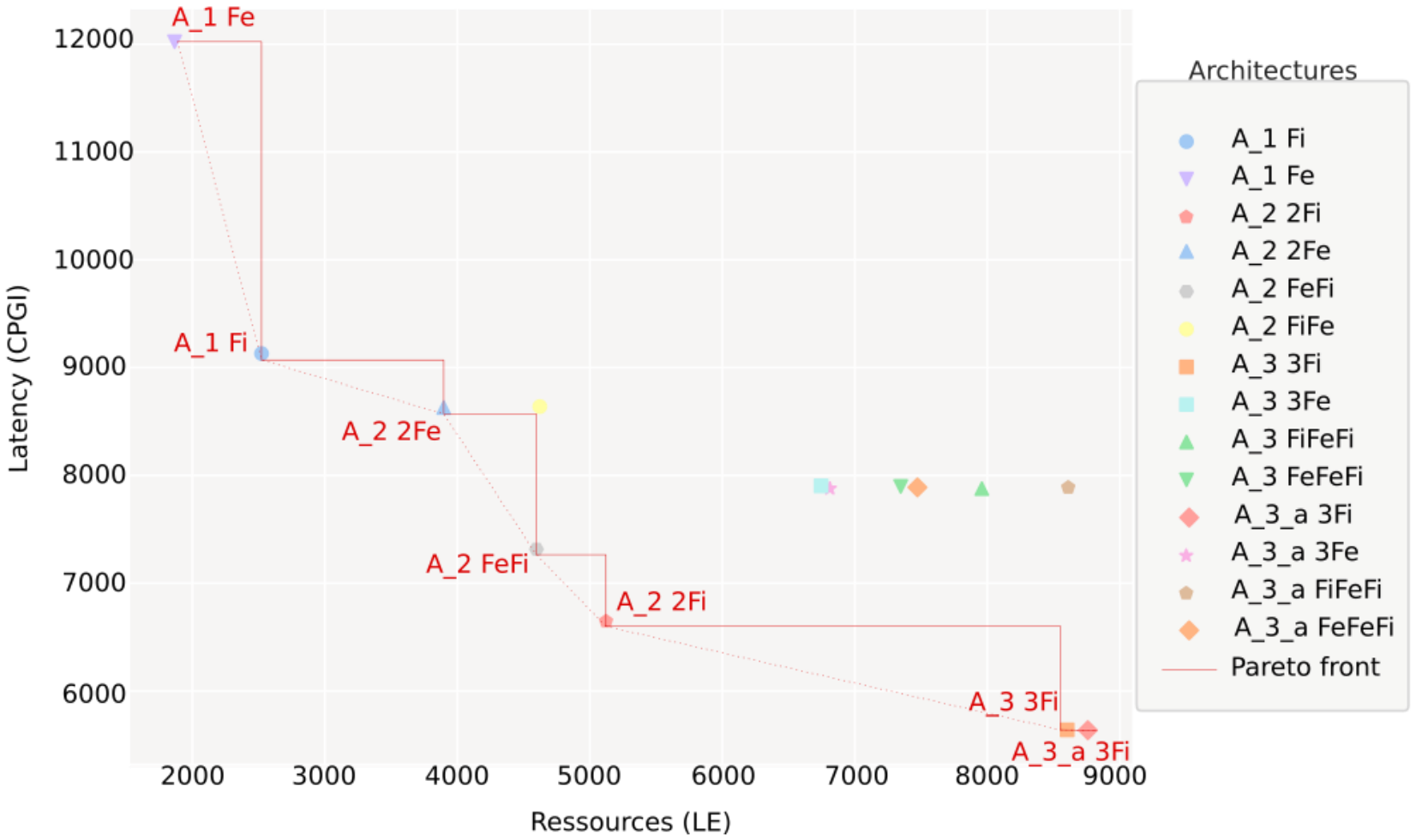


Communication resource overhead



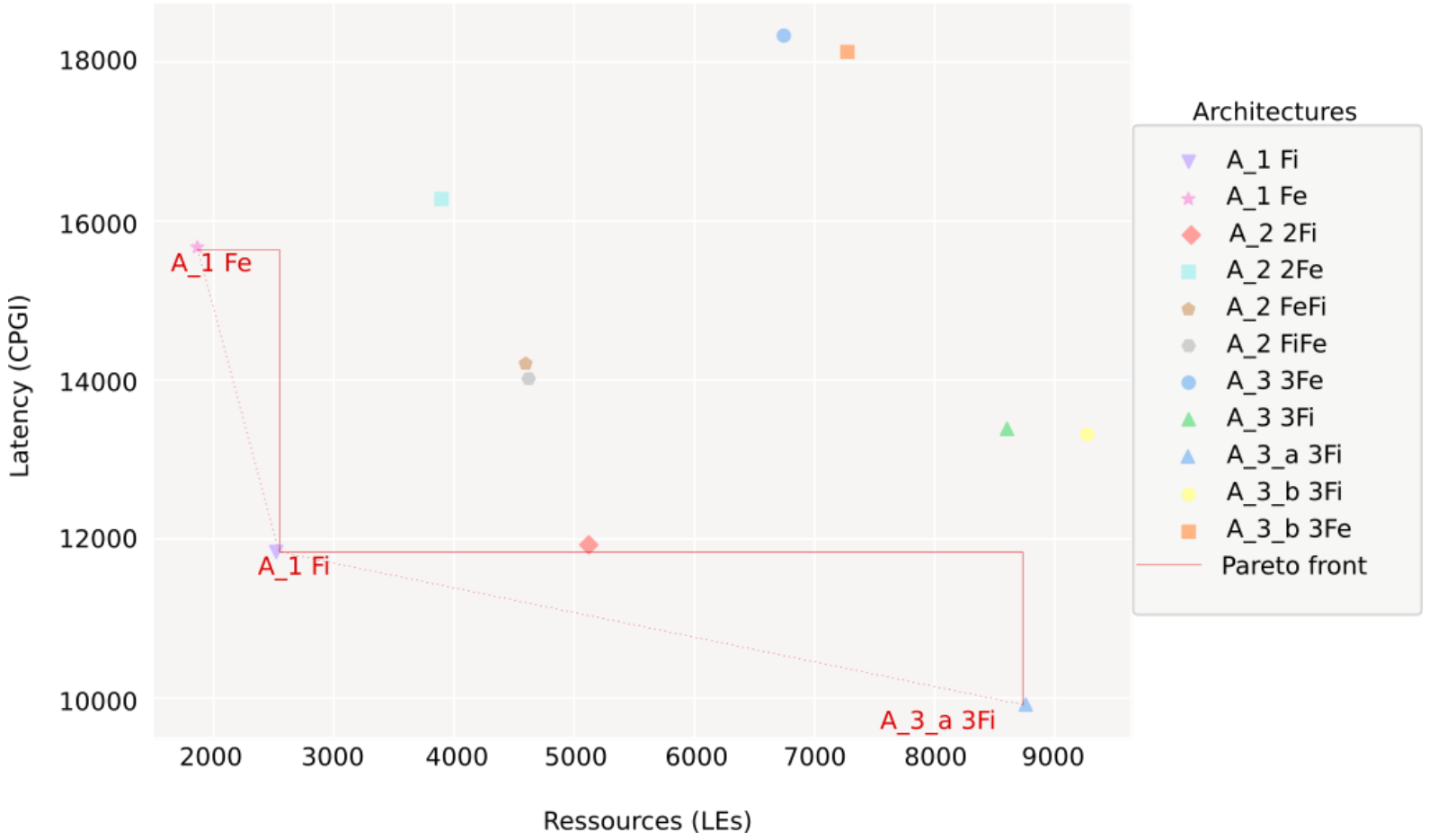
Application latency on hardware versions

Application 1



Application latency on hardware versions

Application 2



Summary and Future Work

Wrapping up

- Scratchy: an adaptable scratchpad-based multi-core architecture for dataflow-modeled applications.
- Demonstrate scratchy through a DSE and Pareto-front generation.

Future Work

- Raise the number of cores
- Automate architecture generation from application requirements
- Support workloads with dynamism, with minimal additional overhead

"What an exciting time to be a computer architect!"

John Hennesy and David Patterson

Thank you for you attention !

Any Question ?

Additional slides

Y-Chart System Design