

# Co-synthesis of FPGA-Based Application-Specific Floating Point SIMD Accelerators

---

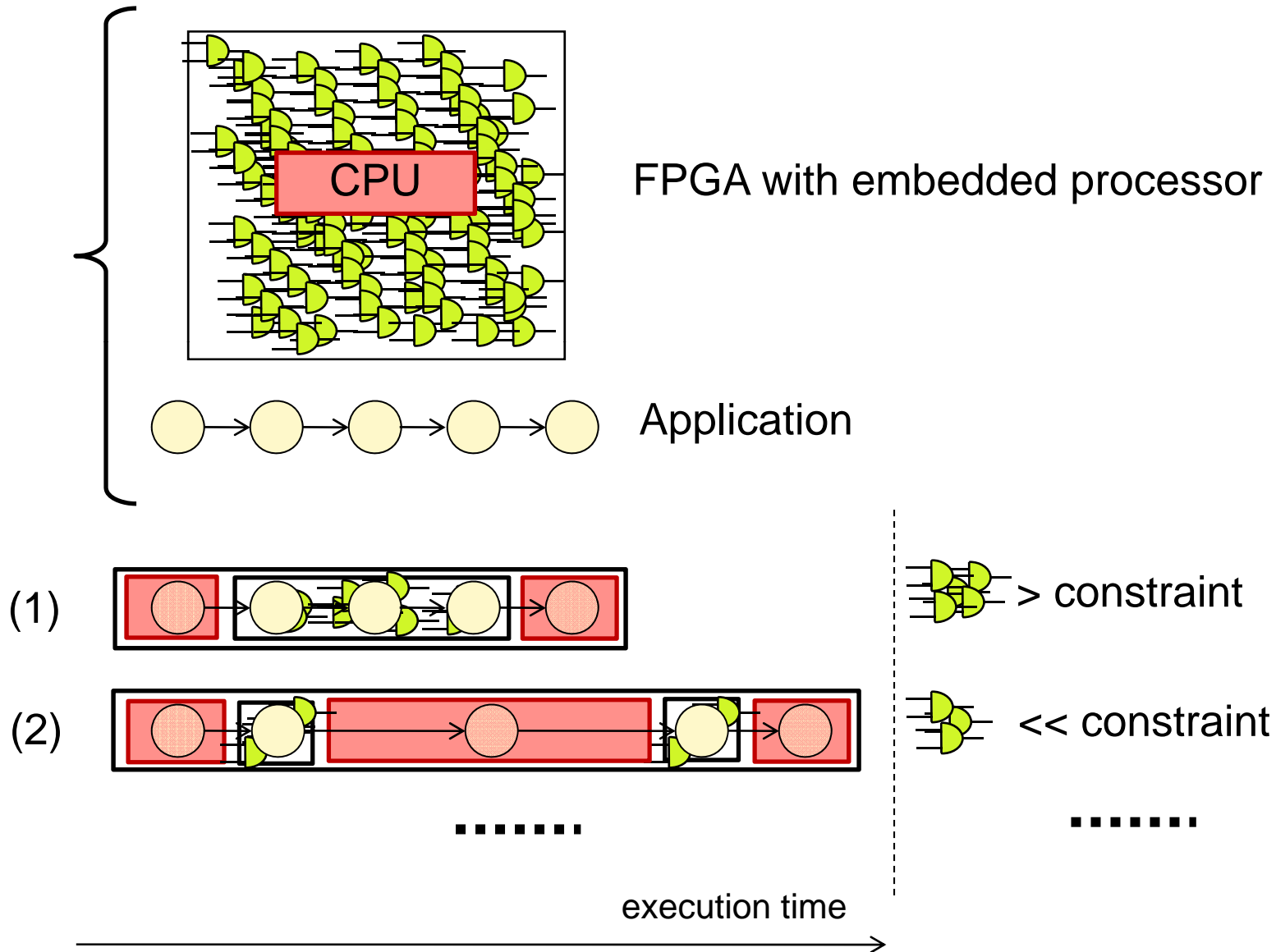
---

---

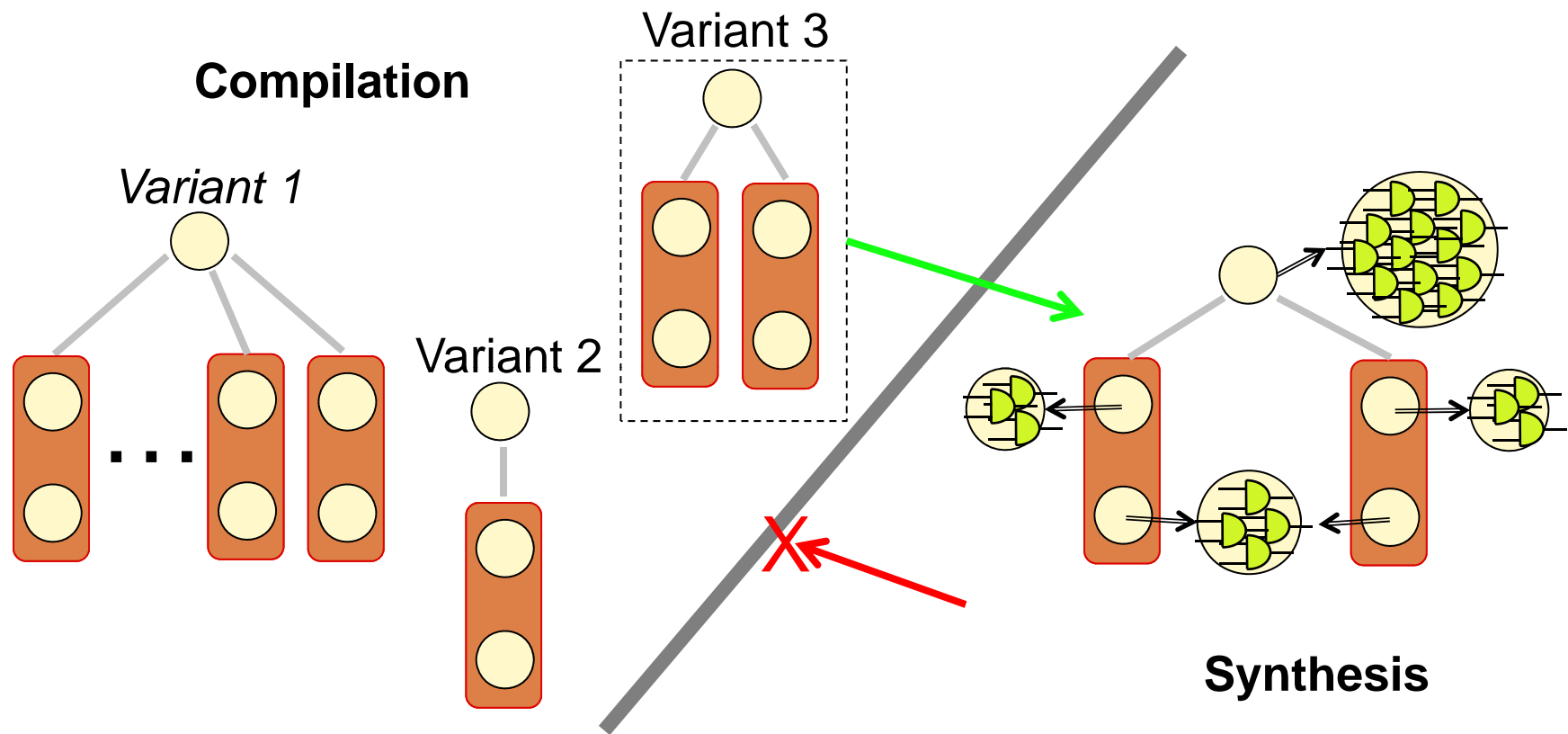
Andrei Hagiescu and Weng-Fai Wong

*National University of Singapore*

# Iterative fitting



# The Great Divide



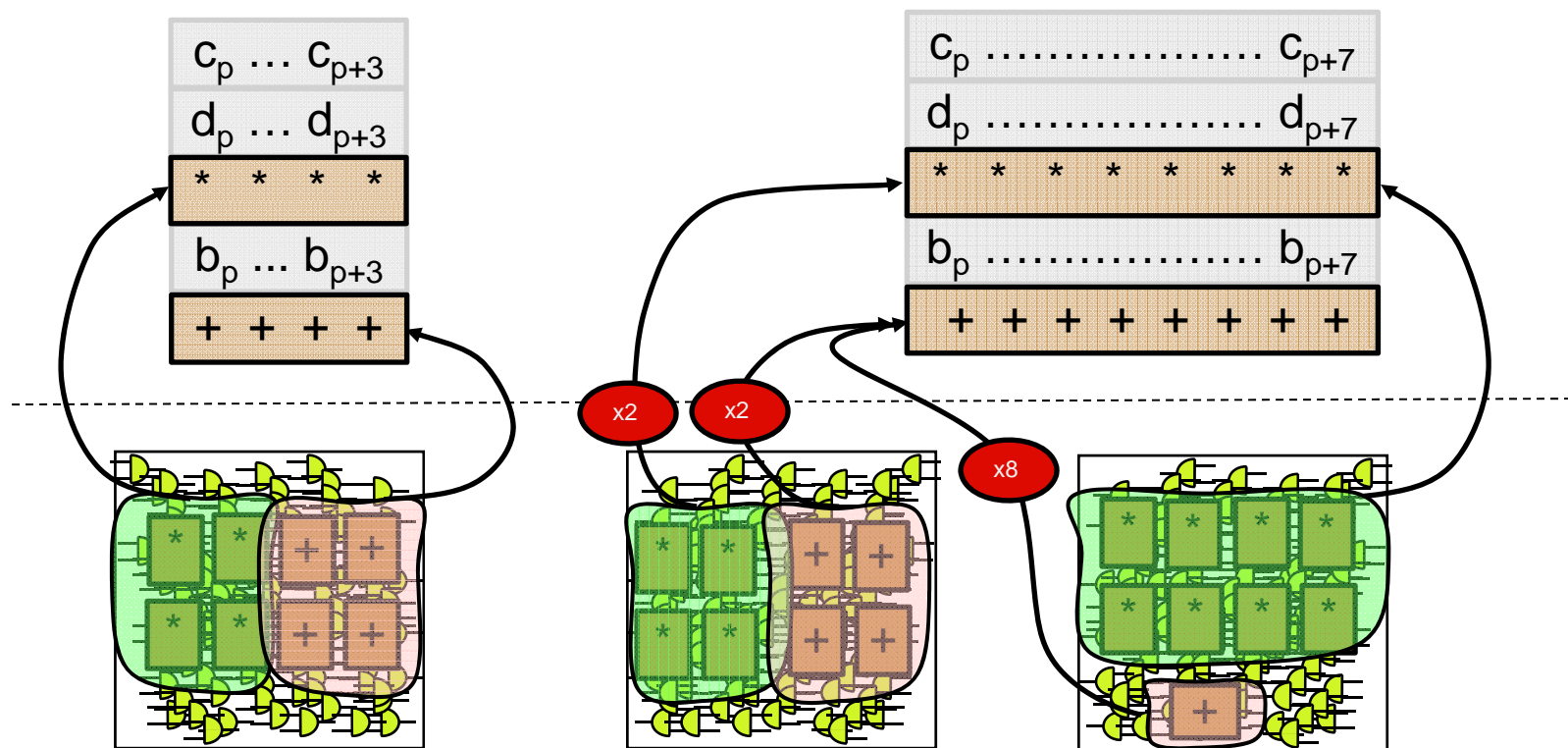
What IF some compilation decisions are postponed?

What IF we carry semantic information?

# Virtualized FP SIMD

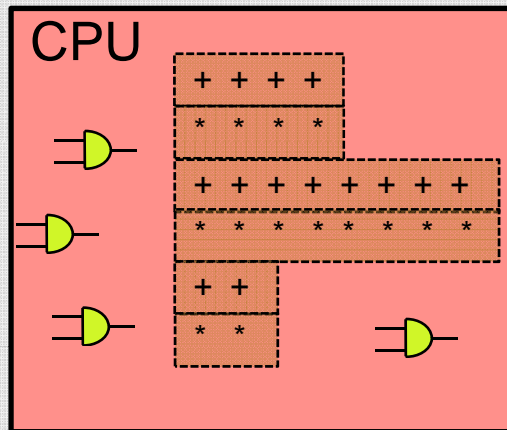
- SIMD floating point instructions
- *Disconnect* semantics from implementation → folding

$$\forall i = [1, n] \quad a_i = b_i + c_i d_i$$



# Flexible SIMD instructions

## Integer SIMD



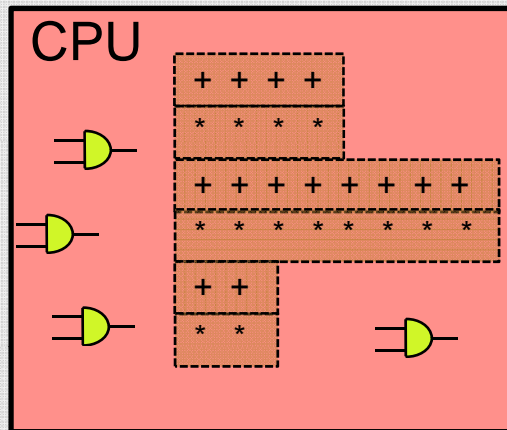
Slow CPU



Tightly connected instructions

# Flexible SIMD instructions

Integer SIMD



Slow CPU

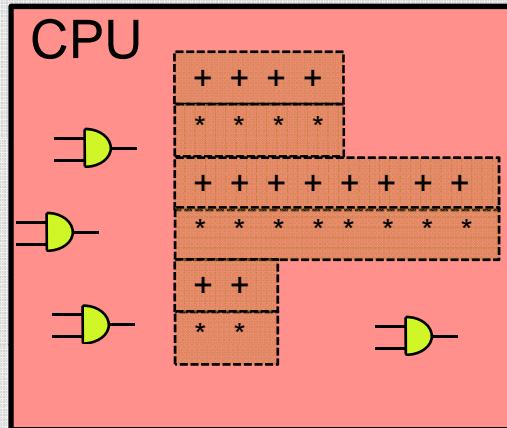


Tightly connected instructions

Tensilica Xtensa  
VESPA

# Flexible SIMD instructions

## Integer SIMD



Slow CPU

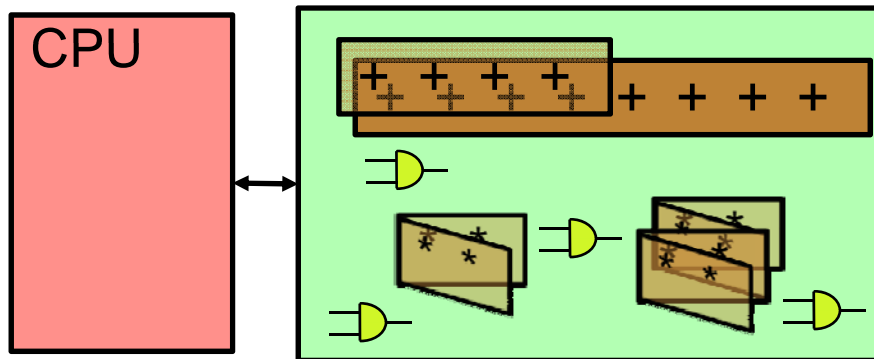


Tightly connected instructions

Tensilica Xtensa  
VESPA

*versus*

## Floating point SIMD



Fast CPU

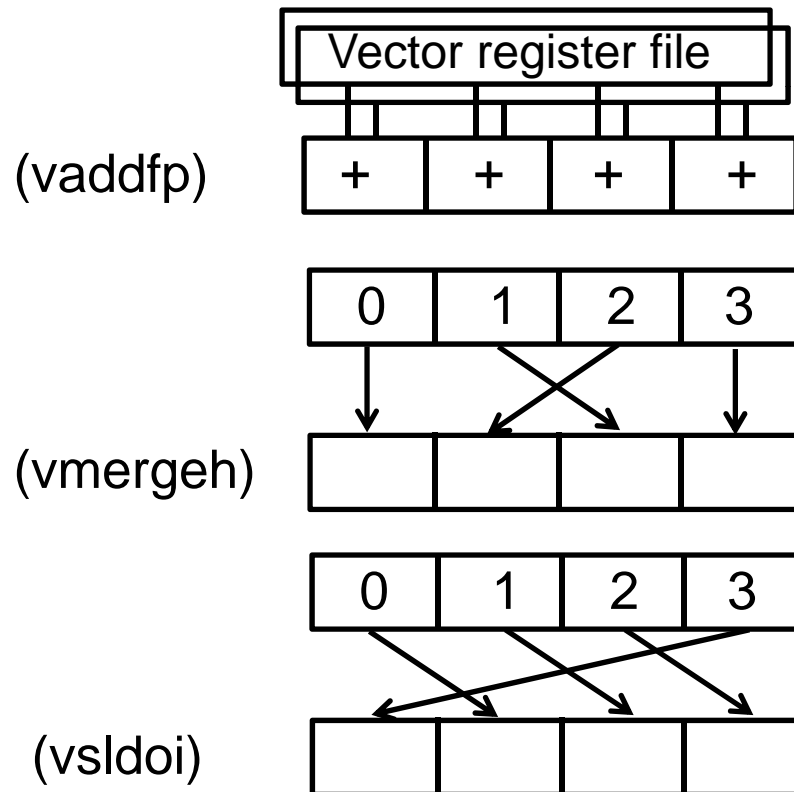


Interface overhead

# Extending AltiVec

---

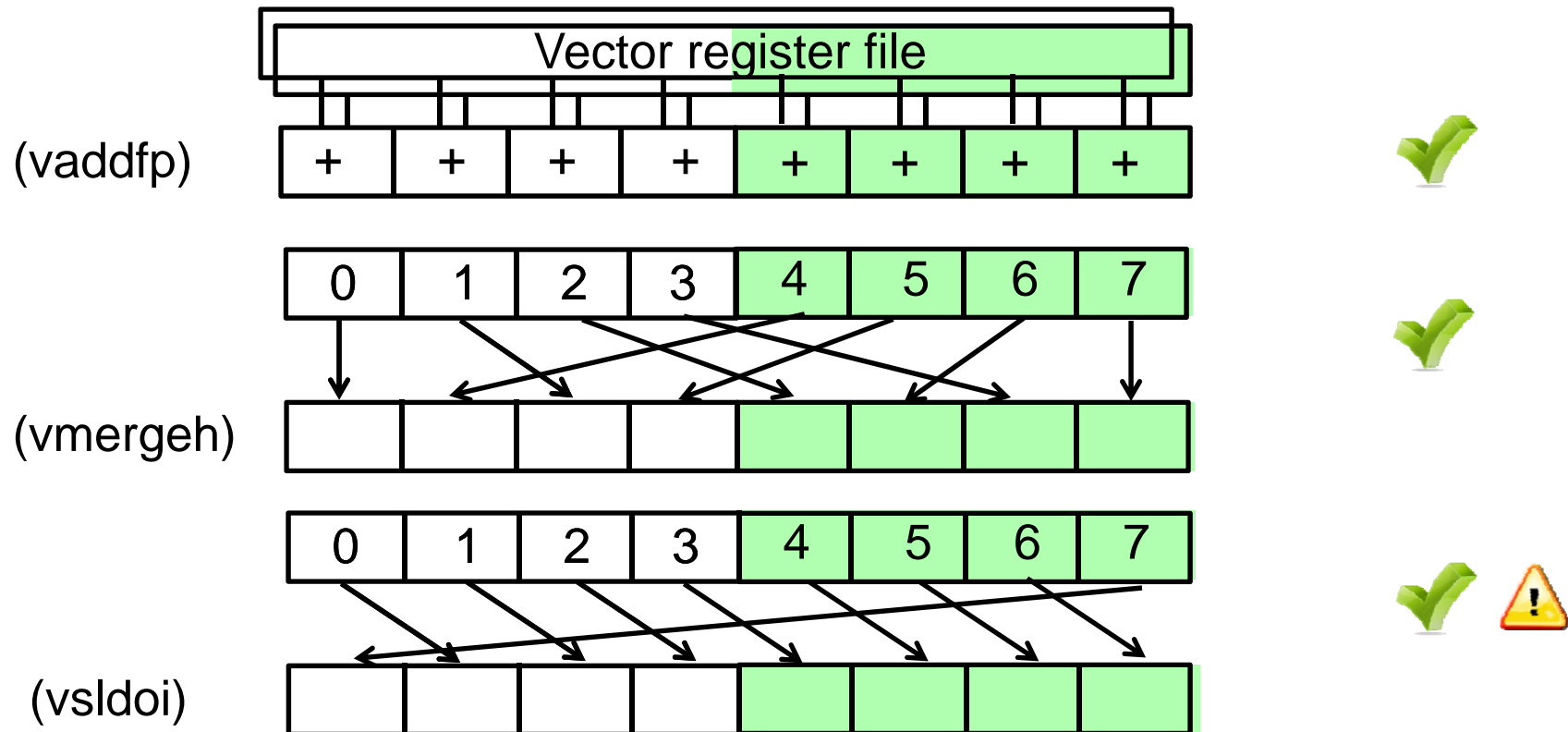
- More ILP encapsulated in each vector instruction
- Extension relies on the generalizing the patterns





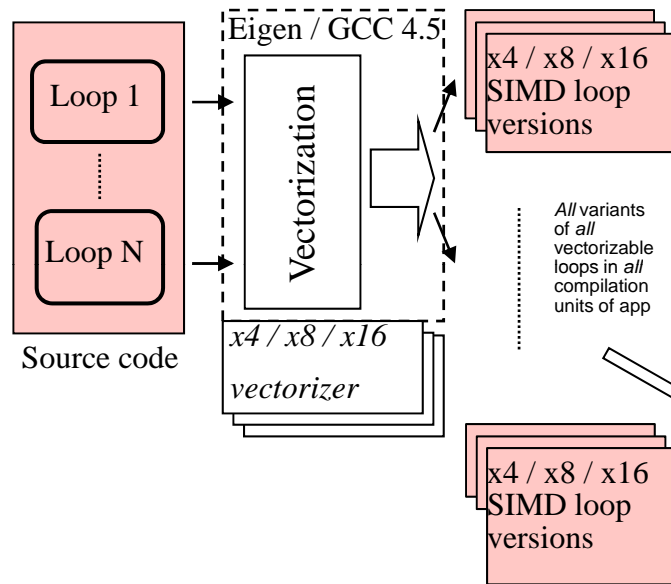
# Extending Altivec

- More ILP encapsulated in each vector instruction
- Extension relies on the generalizing the patterns

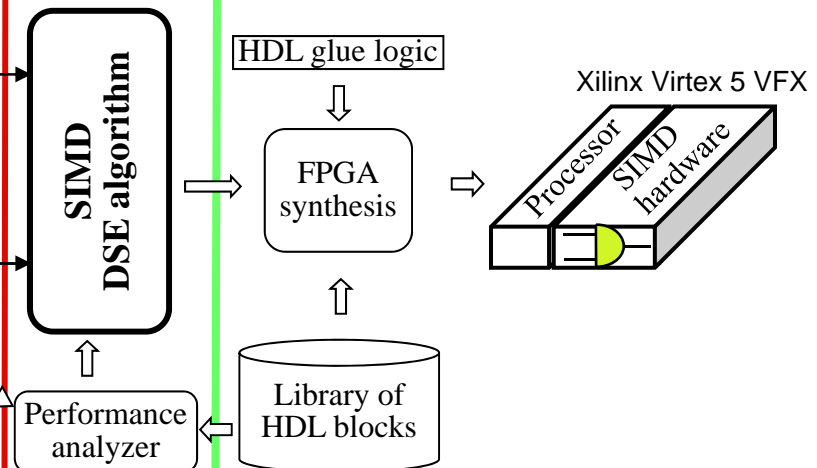


# Compilation flow

## Program compilation & vectorization



## Synthesis



Generate hardware:

- ✓ Global optimization
- ✓ Pareto-optimal performance
- ✓ Non-iterative algorithm

Include correct loop versions (LTO)

Vector length undecided.

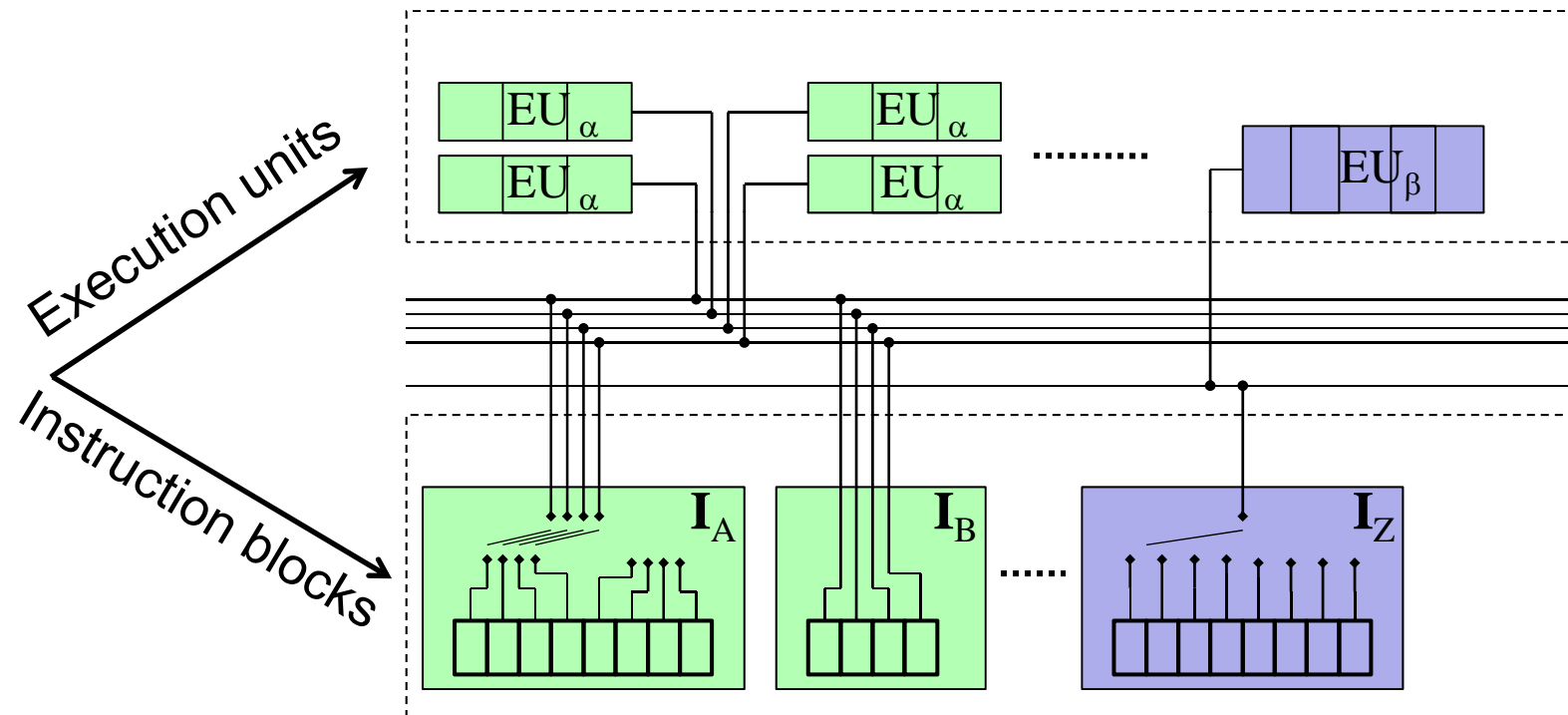
Types and number of execution units undecided.

Vector length of each loop determined.

Types and number of execution units determined.


# Implementing SIMD Instructions

- Instruction blocks fold the execution units as needed



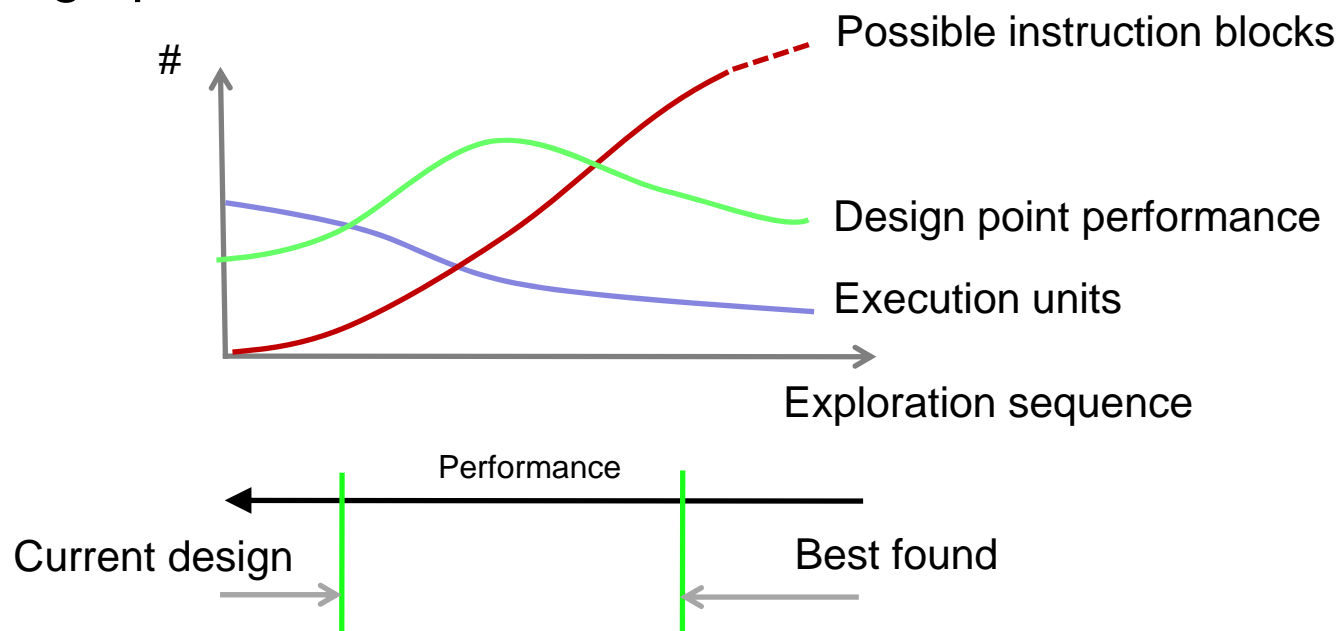
# Hardware

---

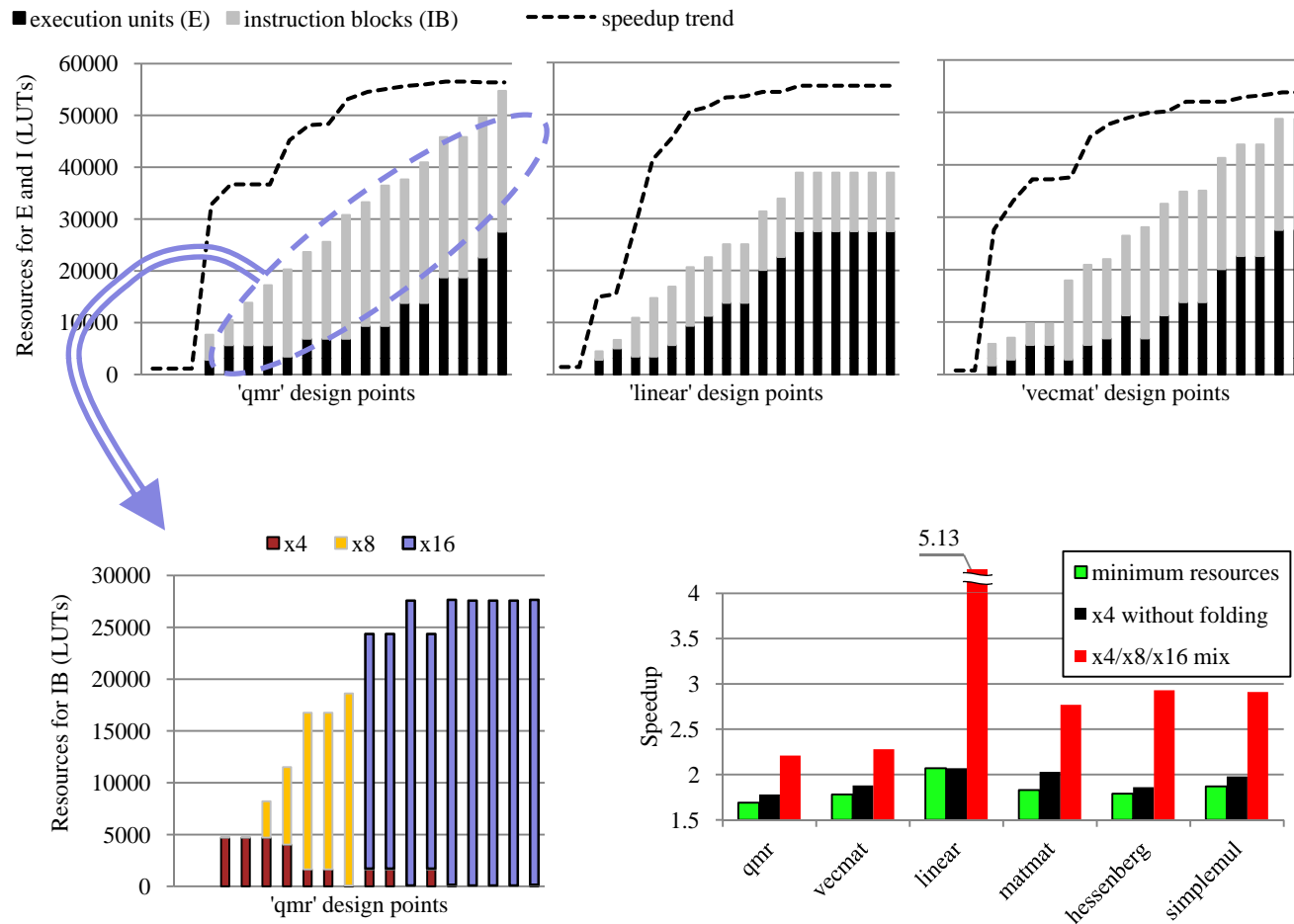
- Cohabitation of Xilinx scalar FP and our SIMD extension
- Shared register file between all SIMD vector lengths
  - ⦿ Multiport: multiple shadow copies
- Folding the bulky SIMD instructions
  - ⦿ Arithmetic
  - ⦿ Permutation
  - ⦿ Loads / stores 
- Folding parameters:
  - ⦿ Instruction blocks
  - ⦿ Execution units

# Hardware DSE algorithm

- Intuition:
  - ⊙ Reduce execution units → free area → evaluate new designs
  - ⊙ Max achievable performance changes monotonically
- Leverage nature of exploration to evaluate a small set of design points



# Results



# Conclusions

---

- Fully automated non-iterative toolchain
- Folding to match resource constraints
- Improved energy consumption
  - ⊙ 22% – 57% of scalar Xilinx FP - measured
- Future directions:
  - ⊙ Partial reconfiguration
  - ⊙ Operator fusing

---

# Questions?